

AOSONG

数字温湿度传感器

AM2320 产品手册

AM2320数字温湿度传感器

超小体积最优性价比
IIC+单总线双输出
小体积：12x15x4.5mm



AOSONG 奥松电子
最专业的温湿度传感器制造商

产品特性

- 超小体积
- 超性价比
- 超低电压工作
- 卓越的长期稳定性
- 标准 I²C 及单总线输出

更多详情请登陆：www.aosong.com

一、产品概述

AM2320 数字温湿度传感器是一款含有已校准数字信号输出的温湿度复合型传感器。采用专用的温湿度采集技术,确保产品具有极高的可靠性与卓越的长期稳定性。传感器包括一个电容式感湿元件和一个高精度集成测温元件,并与一个高性能微处理器相连接。该产品具有品质卓越、超快响应、抗干扰能力强、性价比极高等优点。

AM2320 通信方式采用单总线、标准 I²C 两种通信方式。标准单总线接口,使系统集成变得简易快捷。超小的体积、极低的功耗,信号传输距离可达 20 米以上,使其成为各类应用甚至最为苛刻的应用场合的最佳选择。I²C 通信方式采用标准的通信时序,用户可直接挂在 I²C 通信总线上,无需额外布线,使用简单。两种通信方式都采用直接输出经温度补偿后的湿度、温度及校验 CRC 等数字信息,用户无需对数字输出进行二次计算,也无需要对湿度进行温度补偿,便可得到准确的温湿度信息。两种通信方式可自由切换,用户可自由选择,使用方便,应该领域广泛。产品为 4 引线,连接方便,特殊封装形式可根据用户需求而提供。

AM2320数字温湿度传感器

超小体积最优性价比
IIC+单总线双输出
小体积: 12x15x4.5mm



AOSONG 奥松电子
最专业的温湿度传感器制造商

二、应用范围

暖通空调、除湿器、测试及检测设备、消费品、汽车、自动控制、数据记录器、气象站、家电、湿度调节、医疗、及其他相关湿度检测控制。

三、产品亮点

完全互换、成本低、长期稳定、相对湿度和温度测量、超长的信号传输距离、数字信号输出、精确校准、功耗极低、标准单总线数字接口、标准 I²C 总线数字接口、通信方式可自由选择。

四、外形尺寸 (单位: mm)

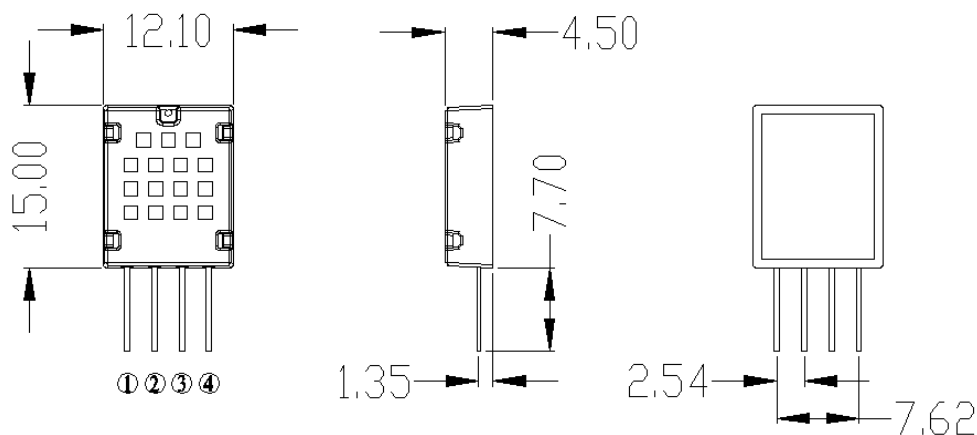


图1 AM2320 外形尺寸

外部接口: 1:VDD 2:SDA 3:GND 4:SCL

五、传感器性能

5.1 相对湿度

表 1: AM2320 相对湿度性能表

| 参数 | 条件 | min | typ | max | 单位 |
|-------------------|----------|------|------|------|--------|
| 分辨率 | | | 0.1 | | %RH |
| 量程范围 | | 0 | | 99.9 | %RH |
| 精度 ¹ | 25°C | | ±3 | | %RH |
| 重复性 | | | ±0.1 | | %RH |
| 互换性 | | 完全互换 | | | |
| 响应时间 ² | 1/e(63%) | | <5 | | S |
| 迟滞 | | | ±0.3 | | %RH |
| 漂移 ³ | 典型值 | | <0.5 | | %RH/yr |

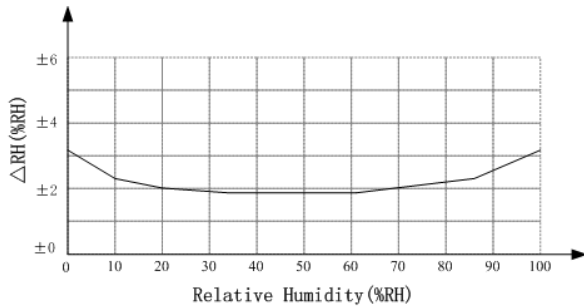


图 2: 25°C时 AM2320 的相对湿度最大误差

5.2 温度

表 2: AM2320 相对温度性能表

| 参数 | 条件 | min | typ | max | 单位 |
|------|----------|------|------|-----|-------|
| 分辨率 | | | 0.1 | | °C |
| | | | 16 | | bit |
| 精度 | | | ±0.5 | | °C |
| 量程范围 | | -40 | | 80 | °C |
| 重复性 | | | ±0.2 | | °C |
| 互换性 | | 完全互换 | | | |
| 响应时间 | 1/e(63%) | | <5 | | S |
| 漂移 | | | ±0.1 | | °C/yr |

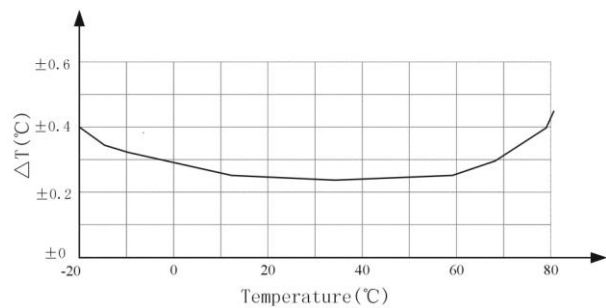


图 3: 温度传感器的温度最大误差

六、电气特性

电气特性,如能耗,高、低电平,输入、输出电压等,都取决于电源。表 3 详细说明了 AM2320 的电气特性,若没有标明,则表示供电电压为 5V。若想与传感器获得最佳效果,请设计时严格遵照表 3 与图 14 及图 27 的条件设计。

表 3: AM2320 直流特性。

| 参数 | 条件 | min | typ | max | 单位 |
|------------------------------|------------------------------|-----|-----|------|-----|
| 供电电压 | | 3.1 | 5 | 5.5 | V |
| 功耗 ⁴ | 休眠 | 8 | 10 | | μA |
| | 测量 | | 950 | | μA |
| | 平均 | | 350 | | μA |
| 低电平输出电压 | I _{OL} ⁵ | 0 | | 300 | mV |
| 高电平输出电压 | R _P <25 kΩ | 90% | | 100% | VDD |
| 低电平输入电压 | 下降 | 0 | | 30% | VDD |
| 高电平输入电压 | 上升 | 70% | | 100% | VDD |
| R _{pu} ⁶ | VDD = 5V VIN = VSS | 30 | 45 | 60 | kΩ |
| 输出电流 | 开 | | 8 | | mA |
| | 三态(关) | 10 | 20 | | μA |
| 采样周期 | | 2.0 | | | S |

1 此精度为出厂时检验时,传感器在 25°C和 5V,条件下测试的精度指标,其不包括迟滞和非线性,且只适合非冷凝环境。

2 在 25°C和 1m/s 气流的条件下,达到一阶响应 63%所需要的时间。

3 在挥发性有机混合物中数值可能会高一些。见说明书应用储存信息。

4 此数值为 VDD =5.0V 在温度为 25°C时,2S/次,条件下的平均值。

5 低电平输出电流。

6 表示上接电阻。

七、接口定义

7.1 AM2320 引脚分配

表 4: AM2320 引脚分配

| 引脚 | 名称 | 描述 |
|----|-----|-------------------|
| 1 | VDD | 电源 (3.1V~5.5V) |
| 2 | SDA | 串行数据, 双向口 |
| 3 | GND | 地 |
| 4 | SCL | 串行时钟, 输入口(单总线时接地) |

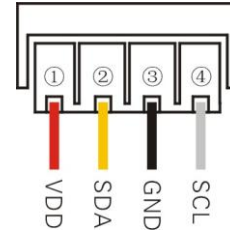


图 4: AM2320 接线图

7.2 电源引脚 (VDD GND)

AM2320 的供电电压范围为 3.1V – 5.5V。

7.3 串行时钟输入 (SCL)

SCL 引脚用于传感器通信方式的选择及 I²C 通信时钟线。当上电时 SCL 保持低电平 500ms 甚至一直保持低电平时, 表示用户选择单总线方式通信, 否则为 I²C 通信; 选择通式方式后, 在通电期间, 传感器通信方式保持不变; 如要更改通信方式, 请重新上电, 并按操作要求选择通信方式。

当选择 I2C 通信时, SCL 用于微处理器与 AM2320 之间的通讯同步。

7.4 串行数据 (SDA)

SDA 引脚为三态结构, 用于读、写传感器数据。具体的通信时序, 见通信方式的详细说明。

八、I²C 及单总线通信协议

AM2320 的串行接口, 在传感器信号的读取及电源损耗方面, 都做了优化处理。传感器采用单总线、I²C 两种通信方式输出, 通信方式可自由切换, 使用方便。总线以 SCL 信号线上电时的电平决定其总线通信方式: 上电时 SCL 保持低电平为单总线通信模式; 上电时 SCL 保持高电平则为 I²C 通信模式。单总线完全兼容本公司其他产品的单总线通信; I²C 通信则按照 I²C 标准协议编址, 可直接挂在 I²C 总线上(总线上只允许挂一个产品), 无需额外布线, 操作简单。读 AM2320 传感器时, 请严格按照两种通信方式的协议及时序进行。具体通信协议详细见单总线及 I²C 通信协议详解。

8.1 标准 I²C 通信协议介绍

8.1.1 I²C 总线简介

AM2320 与微控处理器的接口的另一种形式是 I²C 串行总线, 在此简要地介绍一下 I²C 总线协议标准。限于篇幅, 无法列出协议的全部内容, 更深层次的问题, 请查阅相关的信息(可参考飞利浦公司的网站进行查阅)。

8.1.2 I²C 总线概述

飞利浦 (Philips) 于 20 多年前发明了一种简单的双向二线制串行通信总线, 总线被称为 Inter-I²C 总线。目前 I²C 总线已经成为业界嵌入式应用的标准解决方案, 被广泛应用在各式各样基于微控器的专业、消费与电信产品中, 作为控制、诊断与电源管理总线。多个符合 I²C 总线标准的器件都可以通过同一条 I²C 总线进行通信, 而不需要地址译码器。

I²C 总线只需要由两根信号线组成, 一根是串行数据线 SDA, 另一根是串行时钟线 SCL。一般具有 I²C 总线的器件其 SDA 和 SCL 引脚都是漏极开路 (或集电极开路) 输出结构。因此实际使用时, SDA

和SCL信号线都必须加上拉电阻（ R_p , Pull-Up Resistor）。上拉电阻一般取值3~10 k Ω 。因此，当总线空闲时，这两条信号线都保持高电平，几乎不消耗电流；电气兼容性好，支持各种不同电压的逻辑器件接口；不同的两根总线之间可以直接相连，不需要额外的转换电路，支持多种通信方式—主多从是最常见的通信方式。此外还支持双主机通信、多主机通信以及广播模式等等。

I²C典型配置如图5所示。

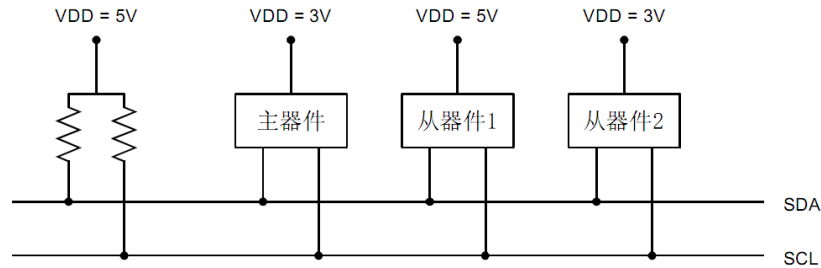


图5: I²C典型配置

8.1.3 I²C 总线协议规范

◎ I²C 总线术语定义

I²C总线通过串行数据 SDA 和串行时钟 SCL 线连接到总线上，使器件间传递信息。每个器件都有一个唯一的地址识别，而且都可以作为一个发送器或接收器（由器件的功能决定），器件在执行数据传输时也可以被看作是主机或从机，主机是初始化总线的数据传输并产生允许传输的时钟信号的器件。此时，任何被寻址的器件都被认为是从机。I²C总线术语定义详见表5。

◎ I²C 总线传送速率

I²C总线的通信速率受主机控制，能快能慢。但是最高速率是有限制的，I²C总线上数据的传输速率在标准模式下最快可达 100Kb/s。

◎ I²C 总线位传输

I²C总线的位传输是通过数据线SDA 和时钟线SCL 两线共同完成的。在时钟线SCL 的高电平期间，数据线SDA 为低电平表示当前传输逻辑电平“0”；在时钟线SCL 的高电平期间，数据线SDA 为高电平表示当前传输逻辑电平“1”。逻辑“0”（低）和“1”（高）的电平，是由 VDD 的相关电平决定（详见表3 AM2320 直流特性表）。同时每传输一个数据位就产生一个时钟脉冲。

表 5: I²C 总线术语的定义

| 术 语 | 描 述 |
|-----|--|
| 发送器 | 发送数据到总线的器件 |
| 接收器 | 从总线接收数据的器件 |
| 主机 | 初始化发送 产生时钟信号和终止发送的器件 |
| 从机 | 被主机寻址的器件 |
| 多主机 | 同时有多于一个主机尝试控制总线 但不破坏报文 |
| 仲裁 | 是一个在有多多个主机同时尝试控制总线 但只允许其中一个控制总线并使报文不被破坏的过程 |
| 同步 | 两个或多个器件同步时钟信号的过程 |

◎ 数据的有效性

数据线SDA 的数据必须在时钟的高电平周期保持稳定不变。数据线SDA 的高或低电平状态只有在时钟线SCL 处于低电平期间才允许改变。但是在I²C总线的起始和结束时例外（详见起始条件和停止条件）。某些其它的串行总线可能规定数据在时钟信号的边沿（上升沿或下降沿）有效，但I²C总线则是电平有效。具体时序图见图6所示。

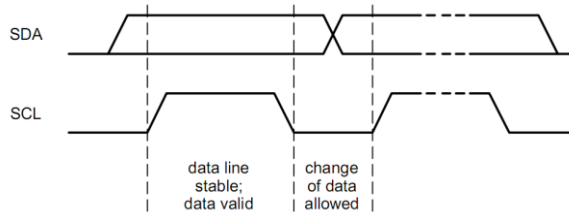


图 6: I²C 总线的位传输

◎ 起始和停止条件

起始条件: 当SCL处于高电平期间时, SDA从高电平向低电平跳变时产生起始条件。总线在起始条件产生后便处于忙的状态。起始条件常常简记为S。

停止条件: 当SCL处于高电平期间时, SDA从低电平向高电平跳变时产生停止条件。总线在停止条件产生后处于空闲状态。停止条件简记为P。

起始和停止条件示意图如图7所示。

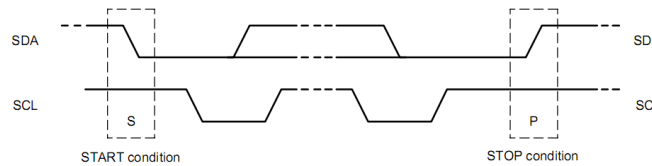


图 7: 起始和停止条件示意图

◎ 字节传输格式

I²C 总线以字节为单位收发数据。传输到SDA线上的每个字节必须为8位。每次传输的字节数量不受限制。首先传输的是数据的最高位 (MSB 第7位), 最后传输的是最低位 (LSB, 第0位)。另外每个字节后必须跟一个响应位 (ACK)。I²C传输数据如图8所示。

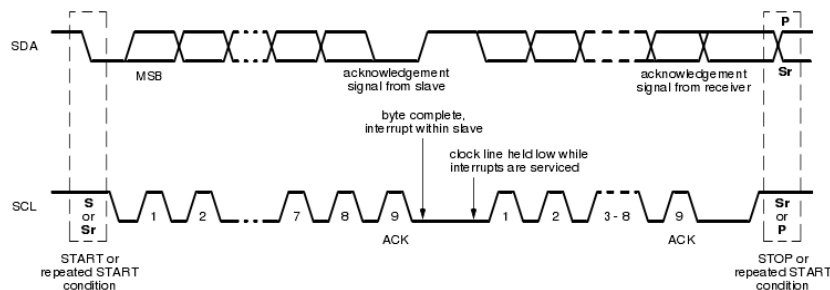


图 8: I²C 总线的数据传输

◎ I²C 总线响应

在I²C总线传输数据过程中, 每传输一个字节, 都要跟一个应答状态位。接收器接收数据的情况可以通过应答位来告知发送器。应答位的时钟脉冲仍由主机产生, 而应答位的数据状态则遵循“谁

接收谁产生”的原则，即总是由接收器产生应答位，在响应的时钟脉冲期间接收器必须将SDA 线拉低，使它在这个时钟脉冲的高电平期间保持稳定的低电平（见图9），当然必须考虑建立和保持时间（详细请查阅表6）。主机向从机发送数据时，应答位由从机产生；主机从从机接收数据时，应答位由主机产生。

I2C总线标准规定：应答位为0表示接收器应答(ACK)，常常简记为A；为1则表示非应答(NACK)，常简记为NA。发送器发送LSB之后，应当释放SDA线（拉高SDA），以等待接收器产生应答位。

如果接收器在接收完最后一个字节的数据，或者不能再接收更多的数据时，应当产生非应答信号来通知发送器。发送器如果发现接收器产生了非应答状态，则应当终止发送。

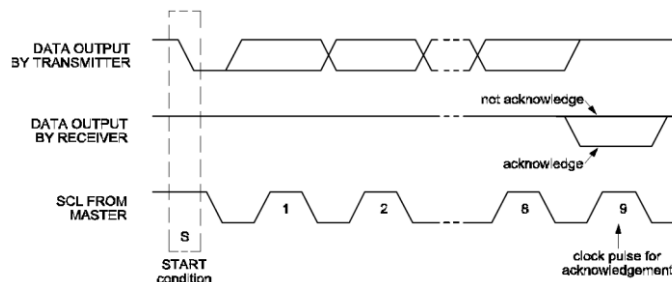


图9：I²C 总线的响应

◎ 从机地址

I²C总线不需要额外的地址译码器和片选信号。多个具有I²C总线接口的器件都可以连接到同一条I²C总线上，它们之间通过器件地址来区分。I²C 总线的寻址过程是通常在起始条件后的第一个字节决定了主机选择哪一个从机，即7位寻址地址（另一种是10位寻址地址，有所不同，本传感器采用7位寻址地址）。第一个字节的位定义如图10所示，第一个字节的头7位组成了从机地址，最低位（LSB）是第8位。它决定了报文的方向，第一个字节的最低位（LSB）是“0”：表示主机会写信息到被选中的从机；“1”表示主机会向从机读信息。

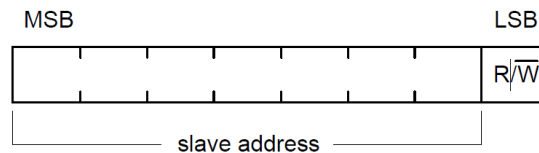


图10：起始条件后的第一个字节

发送了一个地址后，系统中的每个器件都在起始条件后，将头7位与它自己的地址比较如果一样，器件会认为它被主机寻址，至于是从机接收器还是从机发送器都由R/W 位决定。主机是主控制器

件，它不需要器件器件地址，其它器件都属于从机，要有器件地址。必须保证同一条I²C总线上所有从机的地址都是唯一确定的，不能有重复，否则I²C总线将不能正常工作。

◎ 基本的数据传输格式示意图

图11、图12分别给出了I²C的发送和接收数据的基本格式。应当注意的是，图11和图12情况不同的是，在图11中，主机在向从机发送最后一个字节的数据时，从机可能应答也可能非应答，但不管怎样主机都可以产生停止条件。如果主机在向从机发送数据（甚至包括从机地址在内）时检测到从机非应答，则应当及时停止传输。

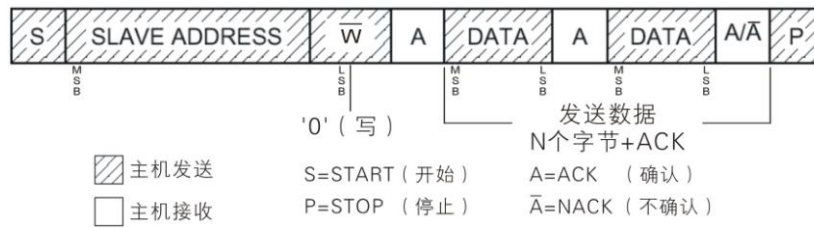


图 11: I²C 总线主机向从机发送数据基本格式

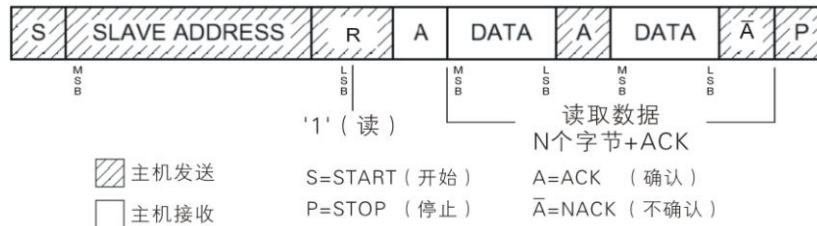


图 12: I²C 总线主机从从机接收数据的基本格式

8.2 AM2320 传感器 I²C 通信协议

AM2320 的串行接口为 I²C 总线，完全按照 I²C 标准协议编址，可直接挂在 I²C 总线上。AM2320 传感器的 I²C 的地址 (SLAVE ADDRESS) 为 0xB8，在 I²C 标准总线的协议基础上，基于 ModBus 协议，制定了独有的通信协议，降低了传输误码率。微控器读 AM2320 传感器时，请严格按照 AM2320 传感器的 I²C_ModBus 通信协议及时序进行设计。

8.2.1 I²C 接口说明

AM2320 数字温湿度传感器作为从机，与主机（用户微处理器）之间通讯方式采用标准 I²C 总线方式。为精确测量环境湿度，减少温度对湿度的影响，AM2320 传感器在非工作期间，自动转为休眠状态，以降低功耗，达到降低传感器自身发热对周围环境湿度值的影响。AM2320 工作模式采用被动式，即主机通过指令唤醒传感器后，再发送相应指令，读取相应的温湿度值；通信结束后，传感器触发一次温湿度采集；因此若长时间没有读传感器，请连续读两次传感器（两次读取最小间隔为 2 S），以第二次值为最新测量值；采集结束后传感器自动转为休眠状态。下次主机需再读传感器时，需重新唤醒传感器。需注意的是主机通信从开始到结束，最长时间为 3 S。若 3 S 内通信未完成，传感器自动结束通信，自动转为休眠状态，主机如要再读传感器，需重新发送唤醒指令。

8.2.2 I²C 接口特性

本小节介绍了 AM2320 传感器的 I²C 接口特性，若想与传感器获得最佳通讯效果，请设计时严格遵照图 13 与表 6 的条件设计。

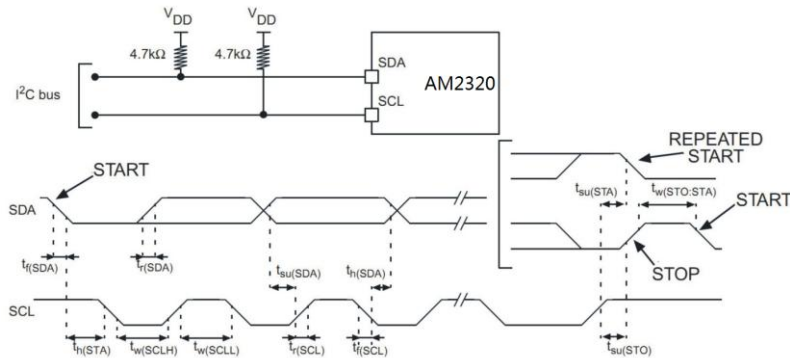


图 13: AM2320 典型的 I2C 总线应用电路和时序图

表 6: AM2320 传感器 I²C 接口特性

| 符号 | 参数 | 标准 I2C 模式 | | 单位 |
|-------------|-----------------|------------------|------|-----|
| | | min | max | |
| SCL 时钟频率 | | | 100 | kHz |
| tw(SCLL) | SCL 时钟低时间 | 4.7 | | μs |
| tw(SCLH) | SCL 时钟高时间 | 4.0 | | |
| tsu(SDA) | SDA 建立时间 | 250 | | ns |
| th(SDA) | SDA 数据保持时间 | 0 ^[1] | | |
| tr(SDA) | SDA 和 SCL 上升时间 | | 1000 | |
| tr(SCL) | | | | |
| tf(SDA) | SDA 和 SCL 下降时间 | | 300 | |
| tf(SCL) | | | | |
| th(STA) | 开始条件保持时间 | 4.0 | | μs |
| tsu(STA) | 重复的开始条件建立时间 | 4.7 | | |
| tsu(STO) | 停止条件建立时间 | 4.0 | | μs |
| tw(STO:STA) | 停止至开始条件时间(总线空闲) | 4.7 | | μs |
| Cb | 每条总线的容性负载 | | 400 | pF |

8.2.3 通信协议

AM2320 传感器的 I²C 通信协议，是在标准 I²C 总线协议的基础上，参照 ModBus 协议，根据 AM2320 传感器自身的特点，组合而成的 I²C_ModBus 协议。具体格式如下：

◎ 通讯数据（信息帧）格式

| | | | | |
|--------------|----------------------|-------|-------|-----------------------|
| 数据格式： | I2C 地址+R/W | 功能码 | 数据区 | CRC 校验 ^[3] |
| 数据长度： | 1 个字节 ^[2] | 1 个字节 | N 个字节 | 16 位 CRC 码（冗余循环码） |

[1] 如果接口不允许延长低电平时间，则只需要遵守开始条件的最长保持时间。

[2] 1 个字节由 8 位二进制数组成（既 8 bit）。

[3] CRC 校验算法，详细见：CRC 码的计算方法；详细见下文。

◎ **通讯信息传输过程**

当通讯命令由发送设备（主机）发送至传感器时，符合传感器 I²C 地址的命令，传感器才进行接收，并根据功能码及相关要求读取信息；然后把执行结果（数据）返送给主机。返回的信息中包括功能码、执行后的数据以及 CRC 校验码（用户可不读取其 CRC，可直接发送停止条件）。

◎ **通讯 I²C 从机地址**

AM2320 传感器每个 I²C 地址都相同，且为 0xB8。因此在同一条总线上只能挂一个 AM2320 传感器，传感器只有在收到起始信号及与本身 I²C 地址相同时才会响应主机。

◎ **通讯 I²C 的功能码**

功能码是每次通讯信息帧传送的第一个字节。I²C_ModBus 通讯规则，可定义的功能码为 1 到 127。作为主机请求发送，通过功能码告诉从机应执行什么动作。作为从机响应，从机返回的功能码与从主机发送来的功能码一样，则表明从机已响应主机并且已进行相关的操作。I²C_ModBus 部分功能码说明见表 7。

表 7: I²C_ModBus 部分功能码

| 功能码 | 定义 | 操作（二进制） |
|------|--------|-----------------|
| 0x03 | 读寄存器数据 | 读取一个或多个寄存器的数据 |
| 0x10 | 写多路寄存器 | 把多组二进制数据写入多个寄存器 |

◎ **通讯 I²C 的数据区**

数据区包括需要由传感器返回何种信息或执行什么动作。这些信息可以是数据（如：温度值、湿度值、传感器设备信息，用户写入数据等等）、参考地址等。例如，主机通过功能码 03 告诉传感器返回寄存器的值（包含要读取寄存器的起始地址及读取寄存器的长度），则返回的数据包括寄存器的数据长度及寄存器的数据内容。

传感器采用自定义的 I²C_Modbus 通讯协议，主机利用通讯命令（功能码 03），可以任意读取其数据寄存器，其数据寄存器表详见表 8。传感器的数据寄存器存储了温湿度值及传感器相应设备信息及其他相关信号；每个数据寄存器是单字节（8 个位）的二进制数据；一次最多可读取传感器的 10 个寄存器的数据，超过读取长度，传感器会返回相应错误代码。错误代码信息见附表 1。

表 8: AM2320 数据寄存器表

| 寄存器信息 | 地址 | 寄存器信息 | 地址 | 寄存器信息 | 地址 | 寄存器信息 | 地址 |
|-------|------|-------------------|------|------------|------|-------|------|
| 湿度高位 | 0x00 | 设备型号高位 | 0x08 | 用户寄存器 1 高位 | 0x10 | 保留 | 0x18 |
| 湿度低位 | 0x01 | 设备型号低位 | 0x09 | 用户寄存器 1 低位 | 0x11 | 保留 | 0x19 |
| 温度高位 | 0x02 | 版本号 | 0x0A | 用户寄存器 2 高位 | 0x12 | 保留 | 0x1A |
| 温度低位 | 0x03 | 设备 ID(24-31) Bit | 0x0B | 用户寄存器 2 低位 | 0x13 | 保留 | 0x1B |
| 保留 | 0x04 | 设备 ID(16-23) Bit | 0x0C | 保留 | 0x14 | 保留 | 0x1C |
| 保留 | 0x05 | 设备 ID(8 - 15) Bit | 0x0D | 保留 | 0x15 | 保留 | 0x1D |
| 保留 | 0x06 | 设备 ID(0 - 7) Bit | 0x0E | 保留 | 0x16 | 保留 | 0x1E |
| 保留 | 0x07 | 状态寄存器 | 0x0F | 保留 | 0x17 | 保留 | 0x1F |

◎ 温度输出格式

温度分辨率是 16Bit，温度最高位（Bit15）等于 1 表示负温度，温度最高位（Bit15）等于 0 表示正温度；温度除了最高位（Bit14~Bit0）表示传感器串出的温度值。传感器串出的温度值是实际温度值的 10 倍；

◎ 状态寄存器

状态寄存器，Bit7~Bit0 位，暂时保留

| 状态寄存器位 | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|--------|------|------|------|------|------|------|------|------|
| 功能介绍 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 |

◎ I²C_ModBus 功能码简介

一、功能码“03”：读传感器的多路寄存器

主机发送读帧格式：

START+ (I²C 地址+W)+功能码 (0x03) +起始地址+寄存器个数+STOP

主机读取返回数据：

START+ (I²C 地址+R)+连续读取传感器返回的数据+STOP

传感器响应帧格式：

功能码 (0x03) +寄存器个数+数据+CRC^[1]

例如：主机连续读取传感器的数据：起始地址为0x00的4个传感器的寄存器数据。

传感器的数据寄存器的地址和数据为：

| 寄存器地址 | 寄存器数据 | 数据说明 | 寄存器地址 | 寄存器数据 | 数据说明 |
|-------|-------|------|-------|-------|------|
| 0x00 | 0x01 | 湿度高位 | 0x02 | 0x00 | 温度高位 |
| 0x01 | 0xF4 | 湿度低位 | 0x03 | 0xFA | 温度低位 |

主机发送的报文格式：

| 主机发送 | 字节数 | 发送的信息 | 备注 |
|-------|-----|-------|---|
| 传感器地址 | 1 | 0xB8 | 传感器 I ² C 地址 (0xB8) +W (0) |
| 功能码 | 1 | 0x03 | 读取寄存器 |
| 起始地址 | 1 | 0x00 | 寄存器起始地址为 0x00 |
| 寄存器个数 | 1 | 0x04 | 读取寄存器个数 |

传感器响应返回的报文格式：

| 从机响应 | 字节数 | 发送的信息 | 备注 |
|-------|-----|-------|------------------------|
| 功能码 | 1 | 0x03 | 读取寄存器 |
| 返回字节数 | 1 | 0x04 | 返回 4 个寄存器共 4 个字节 |
| 寄存器 1 | 1 | 0x01 | 地址为 0x00 的内容 (湿度高字节) |
| 寄存器 2 | 1 | 0xF4 | 地址为 0x01 的内容 (湿度低字节) |
| 寄存器 3 | 1 | 0x00 | 地址为 0x02 的内容 (温度高字节) |
| 寄存器 4 | 1 | 0xFA | 地址为 0x03 的内容 (温度低字节) |
| CRC 码 | 2 | 31A5 | 传感器计算返回的CRC码，低字节在前； |

[1]： CRC 校验的计算详见后面 CRC 校验介绍，传感器返回所有数据均有 CRC 校验，用户可选择读取或不读取。

数值计算:

从传感器读回的温湿度值,只要把数值转化成十进制数再除以 10 便是对应的温湿度值,对应的温度单位为℃,湿度单位为%RH。例如上面读回的数据:

湿度: 01F4 = 1 × 256 + 15 × 16 + 4 = 500 => 湿度 = 500 ÷ 10 = 50.0%RH;

温度: 00FA = 15 × 16 + 10 = 250 => 温度 = 250 ÷ 10 = 25.0℃

注意: CRC 验证码可由 CRC 码的计算代码得出,然后与传感器传的 CRC 码进行对比;相同,则认为数据接收正确,否则,认为数据有误码。

二、功能码“10”：向传感器写多路寄存器

主机利用这个功能码可以把多个数据保存到传感器的寄存器中。AM2320 传感器的单个寄存器为 1 字节,即 8 个位。传感器一次最多允许保存 10 个数据寄存器。因此,主机单次最多向传感器保存 10 个寄存器。超过 10 个,传感器会返回相应错误代码。

主机发送写帧格式:

START+(I²C 地址+W)+功能码(0x10)+起始地址+寄存器个数+保存数据
+CRC+STOP

主机读取确认指令:

START+(I²C 地址+R)+读取传感器返回的数据+STOP

传感器响应帧格式:

功能码 (0x10) + 起始地址 + 寄存器个数 + CRC

例如: 主机要把 01, 02 保存到地址为 10、11 的传感器寄存器中去。

主机发送的报文格式:

| 主机发送 | 字节数 | 发送的信息 | 备注 |
|--------|-----|-------|--|
| 传感器地址 | 1 | 0xB8 | 传感器 I ² C 地址 (0xB8) + W (0) |
| 功能码 | 1 | 0x10 | 写多路寄存器 |
| 起始地址 | 1 | 0x10 | 要写入的寄存器的起始地址 |
| 保存字长度 | 1 | 0x02 | 保存数据的字长度 (共2字) |
| 保存数据 1 | 1 | 0x01 | 保存数据 (地址: 10) |
| 保存数据 2 | 1 | 0x02 | 保存数据 (地址: 11) |
| CRC 码 | 2 | C092 | 主机计算得到的CRC码,低字节在前 (I ² C地址不纳入CRC计算) |

传感器响应返回的报文格式:

| 从机响应 | 字节数 | 发送的信息 | 备注 |
|--------|-----|-------|--------------------|
| 功能码 | 1 | 0x10 | 写多路寄存器 |
| 起始地址 | 1 | 0x10 | 保存的起始地址 |
| 保存数据长度 | 1 | 0x02 | 传感器保存数据长度 |
| CRC 码 | 2 | FC04 | 传感器计算返回CRC码,低字节在前; |

◎ CRC 校验

主机或传感器可用校验码进行判别接收信息是否正确。由于电子噪声或一些其它干扰,信息在传输过程中有时会发生错误,错误校验码 (CRC) 可以检验主机或传感器在通讯数据传送过程中的信息是否有误,错误的信息可以放弃 (无论是发送还是接收),这样增加了系统的安全和效率。

I²C_ModBus通讯协议的CRC（冗余循环码）包含2个字节，即16位二进制数。CRC码由发送设备（主机）计算，放置于发送信息帧的尾部，I²C地址不纳入CRC计算。CRC不管是发送还是接收都是按低字节在前，高字节在后的格式进行发送。接收信息的设备（传感器）再重新计算接收到信息的CRC，比较计算得到的CRC是否与接收到的相符，如果两者不相符，则表明出错。用户需特别注意的是，读传感器指令无需加CRC校验；写传感器时，必需加CRC校验；且所有返回数据均有CRC。

◎ CRC 码的计算方法

1. 预置1个16位的寄存器为十六进制FFFF（即全为1）；称此寄存器为CRC寄存器；
2. 把第一个8位二进制数据（既通讯信息帧的第一个字节）与16位的CRC寄存器的低8位相异或，把结果放于CRC寄存器；
3. 把CRC寄存器的内容右移一位（朝低位）用0填补最高位，并检查右移后的移出位；
4. 如果移出位为0：重复第3步（再次右移一位）；如果移出位为1：CRC寄存器与多项式A001（1010 0000 0000 0001）进行异或；
5. 重复步骤3和4，直到右移8次，这样整个8位数据全部进行了处理；
6. 重复步骤2到步骤5，进行通讯信息帧下一个字节的处理；
7. 将该通讯信息帧所有字节按上述步骤计算完成后，得到的16位CRC寄存器的高、低字节进行交换；
8. 最后得到的CRC寄存器内容即为：CRC码。

◎ CRC 码的 C 语言计算代码

说明：此程序计算*ptr内前len长度个字节的CRC码。

```
unsigned short crc16(unsigned char *ptr, unsigned char len)
```

```
{
    unsigned short crc=0xFFFF;
    unsigned char i;
    while(len--)
    {
        crc ^=*ptr++;
        for(i=0;i<8;i++)
        {
            if(crc & 0x01)
            {
                crc>>=1;
                crc^=0xA001;
            }else
            {
                crc>>=1;
            }
        }
    }
    return crc;
}
```

8.2.4 I²C 通信时序

AM2320 传感器 I²C 通信方式，虽然是按照标准的 I²C 通信时序，但必需按照我们的通信协议及通信时序要求，才能准确的读写传感器。请设计时严格按照通信协议及时序进行读取。

◎ I²C读写完整时序示例

图14给出了完整的读写传感器的示例及读写时特殊时间要求，请严格按照特殊时间要求来读写，否则将出现无法读取传感器或数据不正确等情况。图中的几个时序特别需注意，详细见图中

的时间要求；主机通信从开始到结束，最长时间为3 S。



图 14: I²C 读写传感器完整示例图

◎ I²C 读写时序分解

读或写传感器，必需按以下三个步骤进行，否则将无法通信或无法读出正确的数据：

步骤一：唤醒传感器

为了降低传感器自身发热带来的湿度误差，传感器在非工作状态，处于休眠状态，因此要读传感器必须先唤醒传感器，才能进行发送读写指令，否则传感器将不作响应。须注意的是，在唤醒传感器时，发送的 I²C 地址，传感器不会回应 ACK，但主机必须发送确认是否回 ACK 的时钟，即第九个 SCL 时钟信号。唤醒传感器的操作为如下指令：

主机发送起始信号加起始地址后，等待一段时间（等待时间至少 800μs，最大 3ms；如主机是硬件 I²C，则无需等待，硬件 I²C 会自动等待），然后发送停止信号。

即：起始信号+0xB8+等待(>800us)+停止信号 时序图如图 15 所示。

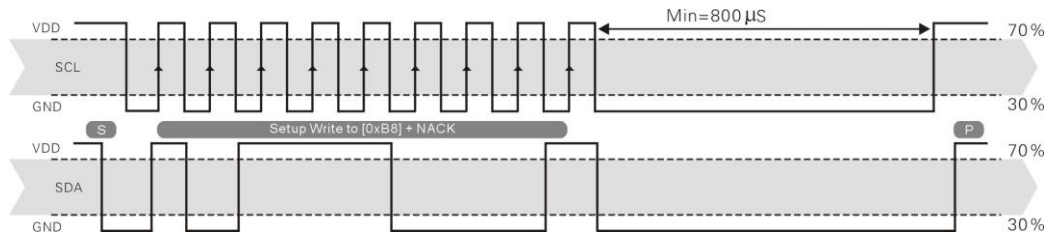


图 15: 唤醒传感器

步骤二 发送读指令或发送写指令

唤醒 AM2320 传感器后，就可完全按照标准的 I²C 读写时序了，最大速度支持 100Kb/s。读取温湿度示例，如图 16 所示。

主机发送指令为：START+0xB8 (SLA) +0x03(功能码)+0x00(起始地址)
+0x04(寄存器长度)+STOP

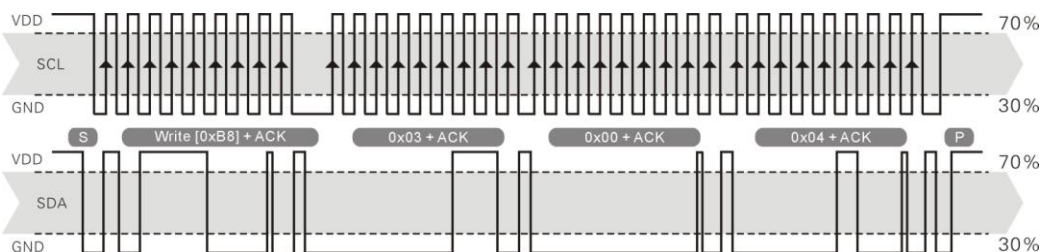


图 16: 发送读温湿度指令示例

步骤三 读返回数据或确认信号

发送读/写指令后，主机需等待至少 1.5ms，然后再发送读取时序，读取返回数据示例如图 17 所示；需注意的是，你读取数据时，发完 I²C 地址后，需至少等待 30μs 以上再发送下一个串行时钟，进行读取数据，否则通信将出现错误。

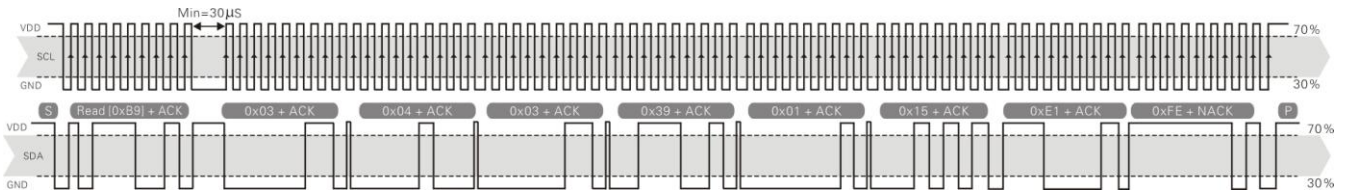


图 17: 读取温湿度值示例

主机读回数据为:

0x03(功能码)+0x04(数据长度)+0x03(湿度高位)+0x39(湿度低位)+

0x01(温度高位)+0x15(温度低位)+0xE1(CRC 校验码低字节)+ 0xFE(CRC 校验码高字节);

因此: 0339H = 3 × 256 + 3 × 16 + 9 = 825 =>湿度 = 825 ÷ 10 = 82.5%RH;

0115H = 1 × 256 + 1 × 16 + 5 = 277 =>温度 = 277 ÷ 10 = 27.7°C

通过以上三个步骤可完成对传感器所有寄存器的读、写操作（用户可写寄存器，只有5个，即状态寄存器，四个用户寄存器，同时，状态寄存器，只能单独写入，否则会报错）；用户在设计时，必须完全按照以上三个步骤进行读写。

传感器发送完数据后，触发一次温湿度测量；测量完成后，记录温湿度值，此时一次通信全部完成，传感器自动转入休眠状态；因此如长时间未读传感器，连续读取二次传感器，以第二次读回的温湿度为最新的值（连续读取最小间隔为2S）。

8.2.5 外设读取流程图

AM2320 传感器读取 I²C 流程图示意图如图 18 所示，同时我们公司还提供了 C51 的读取代码示例，需下载的客户，请登录我公司的网站（www.aosong.com）进行相关下载，此说明书不提供代码说明。

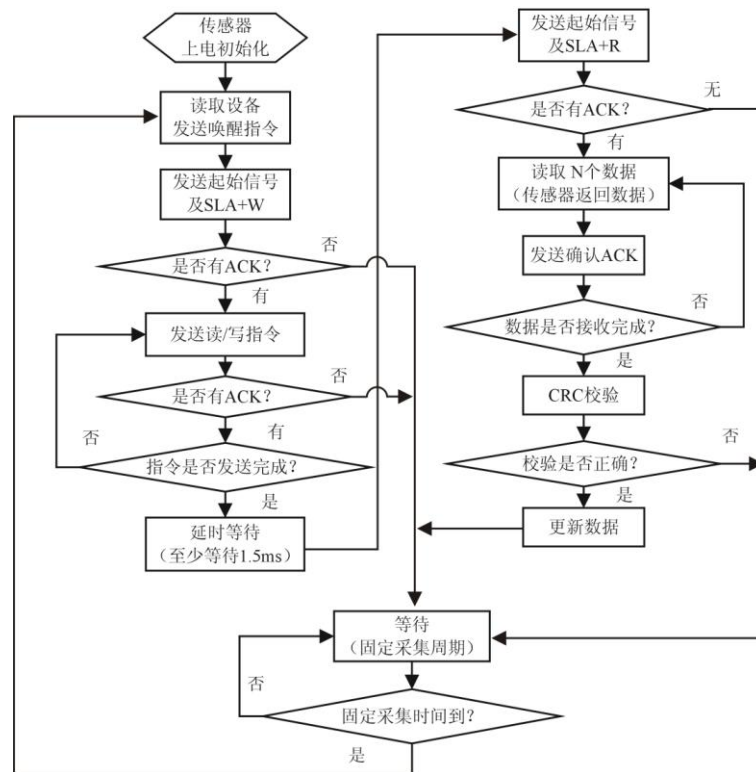


图 18: 用 I²C 读传感器流程图

附表 1: I²C_MODBUS 通信协议总表

| 读取总线说明: I ² C 的地址为 0xB8; 一次最多访问 10 个寄存器; 读取总线一次通信时间最长时间为 3 S。 传感器每次返回数据都加 CRC 校验, 用户可选择不读 CRC 校验。 | | | | | | | |
|--|------|-------------------|---|------------|------|-------|------|
| 读传感器帧格式: 主机帧格式: (SLA+W)+功能码(0x03)+起始地址+寄存器个数 从机帧格式: 功能码(0x03)+数据长度+返回数据+CRC | | | | | | | |
| 写传感器帧格式: 主机帧格式: (SLA+W)+功能码(0x10)+起始地址+寄存器个数+保存数据+CRC 从机帧格式: 功能码(0x03)+起始地址+寄存器个数+CRC | | | | | | | |
| AM2320 传感器寄存器列表: | | | | | | | |
| 寄存器信息 | 地址 | 寄存器信息 | 地址 | 寄存器信息 | 地址 | 寄存器信息 | 地址 |
| 湿度高位 | 0x00 | 设备型号高位 | 0x08 | 用户寄存器 1 高位 | 0x10 | 保留 | 0x18 |
| 湿度低位 | 0x01 | 设备型号低位 | 0x09 | 用户寄存器 1 低位 | 0x11 | 保留 | 0x19 |
| 温度高位 | 0x02 | 版本号 | 0x0A | 用户寄存器 2 高位 | 0x12 | 保留 | 0x1A |
| 温度低位 | 0x03 | 设备 ID(24-31) Bit | 0x0B | 用户寄存器 2 低位 | 0x13 | 保留 | 0x1B |
| 保留 | 0x04 | 设备 ID(16-23) Bit | 0x0C | 保留 | 0x14 | 保留 | 0x1C |
| 保留 | 0x05 | 设备 ID(8 - 15) Bit | 0x0D | 保留 | 0x15 | 保留 | 0x1D |
| 保留 | 0x06 | 设备 ID(0 - 7) Bit | 0x0E | 保留 | 0x16 | 保留 | 0x1E |
| 保留 | 0x07 | 状态寄存器 | 0x0F | 保留 | 0x17 | 保留 | 0x1F |
| 状态寄存器定义: Bit7-Bit0 位保留; | | | | | | | |
| 温度格式: 温度最高位 (Bit15) 等于 1 表示负温度, 温度最高位 (Bit15) 等于 0 表示正温度; 温度除了最高位 (Bit14~Bit0) 表示传感器串出的温度值。传感器串出的温度值是实际湿度值的 10 倍; | | | | | | | |
| 写传感器: 可供用户写的寄存器为(0x0F~0x13); 其他寄存器禁止写, 且状态寄存器只能单独写。 | | | | | | | |
| 读写示例: | | | | | | | |
| 功 能 | 功能码 | 始地址 | 帧数据内容 | | | | |
| 读取温湿度 | 0x03 | 0x00 | 发送: (SLA+W)+0x03+0x00+0x04 | | | | |
| | | | 返回: 0x03+0x04+湿度高位+湿度低位+温度高位+温度低位+CRC | | | | |
| 读取温度 | 0x03 | 0x02 | 发送: (SLA+W)+0x03+0x02+0x02 | | | | |
| | | | 返回: 0x03+0x02+温度高位+温度低位+CRC | | | | |
| 读取湿度 | 0x03 | 0x00 | 发送: (SLA+W)+0x03+0x00+0x02 | | | | |
| | | | 返回: 0x03+0x02+湿度高位+湿度低位+ CRC | | | | |
| 读设备信息 | 0x03 | 0x08 | 发送: (SLA+W)+0x03+0x08+0x07 | | | | |
| | | | 返回: 0x03+0x07+设备型号(16 位)+版本号(8 位)+ ID(32 位)+CRC | | | | |
| 写状态寄存器 | 0x10 | 0x0F | 发送: (SLA+W)+0x10+0x0F+0x01+0x01+0xF4 (低位)+0xB7 (高位) 注释: 功能码+寄存器起始地址+寄存器个数+保存内容+CRC | | | | |
| | | | 返回: 0x10+0x0F+0x01+0xB4 (低字节)+0x35 (高字节) 注释: 功能码+寄存器起始地址+寄存器个数+ CRC | | | | |
| 写用户寄存器 1 | 0x10 | 0x10 | 发送: (SLA+W)+0x10+0x10+0x02+0x01+0x02+0xC0+0x92 | | | | |
| | | | 返回: 0x10+0x10+0x02+0xFC+0x04 | | | | |

注: SLA= I²C 地址 0xB8。表中 CRC 为校验位, CRC 为 16 位, 低字节在前, 高字节在后。

返回错误代码: 0x80: 不支持功能码 0x81: 读取非法地址 0x82: 超出写数据范围
 0x83: CRC 校验错误 0x84: 禁止写。

8.3 单总线通信 (ONE-WIRE)

8.3.1 单总线典型电路

微处理器与 AM2320 的连接典型应用电路如图 19 所示。单总线通信模式时，SDA 上拉后与微处理器的 I/O 端口相连。

单总线通信特殊说明：

1. 典型应用电路中建议连接线长度短于 30 米时用 5.1K 上拉电阻，大于 30 米时根据实际情况降低上拉电阻的阻值。
2. 建议 3.3V 电压供电时连接线长度不得大于 1m。否则线路压降有可能会造成传感器通信异常。
3. 读取传感器最小间隔时间为 2S；读取间隔时间小于 2S，可能导致温湿度不准或通信不成功等情况。
4. 每次读出的温湿度数值是上一次测量的结果，欲获取实时数据，需连续读取两次，建议连续多次读取传感器，且每次读取传感器间隔大于 2 秒即可获得准确的数据。

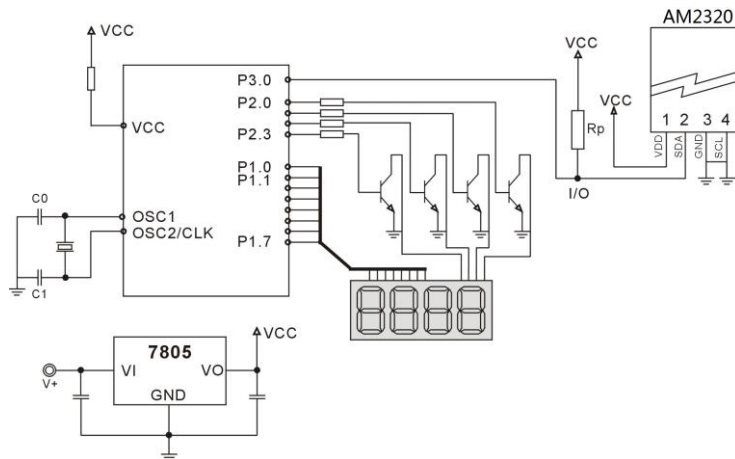


图 19： AM2320 单总线典型电路

8.3.2、单总线通信协议

◎单总线说明

AM2320 器件采用简化的单总线通信。单总线即只有一根数据线，系统中的数据交换、控制均由数据线完成。设备（微处理器）通过一个漏极开路或三态端口连至该数据线，以允许设备在不发送数据时能够释放总线，而让其它设备使用总线；单总线通常要求外接一个约 5.1kΩ 的上拉电阻，这样，当总线闲置时，其状态为高电平。由于它们是主从结构，只有主机呼叫传感器时，传感器才会应答，因此主机访问传感器都必须严格遵循单总线序列，如果出现序列混乱，传感器将不响应主机。

◎单总线传送数据定义

SDA 用于微处理器与 AM2320 之间的通讯和同步,采用单总线数据格式，一次传送 40 位数据，高位先出。具体通信时序如图 20 所示，通信格式说明见表 9。

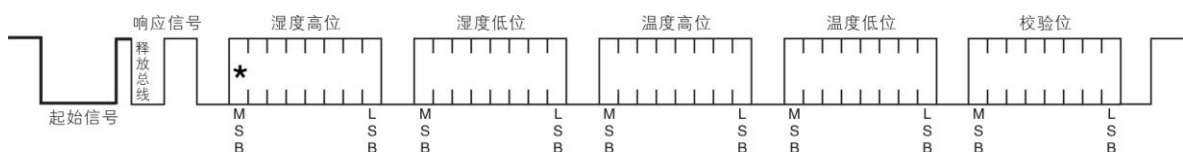


图 20： AM2320 单总线通信协议

表 9: AM2320 通信格式说明

| 名称 | 单总线格式定义 |
|------|--|
| 起始信号 | 微处理器把数据总线 (SDA) 拉低一段时间(至少 800μs) ^[1] , 通知传感器准备数据。 |
| 响应信号 | 传感器把数据总线 (SDA) 拉低 80μs, 再接高 80μs 以响应主机的起始信号。 |
| 数据格式 | 收到主机起始信号后, 传感器一次性从数据总线 (SDA) 串出 40 位数据, 高位先出 |
| 湿度 | 湿度分辨率是 16Bit, 高位在前; 传感器串出的湿度值是实际湿度值的 10 倍。 |
| 温度 | 温度分辨率是 16Bit, 高位在前; 传感器串出的温度值是实际温度值的 10 倍; 温度最高位 (Bit15) 等于 1 表示负温度, 温度最高位 (Bit15) 等于 0 表示正温度; 温度除了最高位 (Bit14~Bit0) 表示温度值。 |
| 校验位 | 校验位 = 湿度高位 + 湿度低位 + 温度高位 + 温度低位 |

[1] 详细说明见 7.3

◎单总线数据计算示例

示例一: 接收到的 40 位数据为:

0000 0010 1001 0010 0000 0001 0000 1101 1010 0010
 湿度高 8 位 湿度低 8 位 温度高 8 位 温度低 8 位 校验位

计算:

$0000\ 0010 + 1001\ 0010 + 0000\ 0001 + 0000\ 1101 = 1010\ 0010$ (校验位)

接收数据正确:

湿度: $0000\ 0010\ 1001\ 0010 = 0292H$ (十六进制) = $2 \times 256 + 9 \times 16 + 2 = 658$
 \Rightarrow 湿度 = 65.8%RH

温度: $0000\ 0001\ 0000\ 1101 = 10DH$ (十六进制) = $1 \times 256 + 0 \times 16 + 13 = 269$
 \Rightarrow 温度 = 26.9℃

◎特殊说明:

当温度低于 0℃ 时温度数据的最高位置 1。

示例: -10.1℃ 表示为 1 000 0000 0110 0101

温度: $0000\ 0000\ 0110\ 0101 = 0065H$ (十六进制) = $6 \times 16 + 5 = 101$
 \Rightarrow 温度 = -10.1℃

示例二: 接收到的 40 位数据为:

0000 0010 1001 0010 0000 0001 0000 1101 1011 0010
 湿度高 8 位 湿度低 8 位 温度高 8 位 温度低 8 位 校验位

计算:

$0000\ 0010 + 1001\ 0010 + 0000\ 0001 + 0000\ 1101 = 1010\ 0010 \neq 1011\ 0010$ (校验错误)

本次接收的数据不正确, 放弃, 重新接收数据。

8.3.3 单总线通信时序

用户主机 (MCU) 发送一次起始信号 (把数据总线 SDA 拉低至少 800μs) 后, AM2320 从休眠模式转换到高速模式。待主机开始信号结束后, AM2320 发送响应信号, 从数据总线 SDA 串行送出 40Bit

的数据, 先发送字节的高位; 发送的数据依次为湿度高位、湿度低位、温度高位、温度低位、校验位, 发送数据结束触发一次信息采集, 采集结束传感器自动转入休眠模式, 直到下一次通信来临。

详细时序信号特性见表 10，单总线通信时序图如图 21 所示。

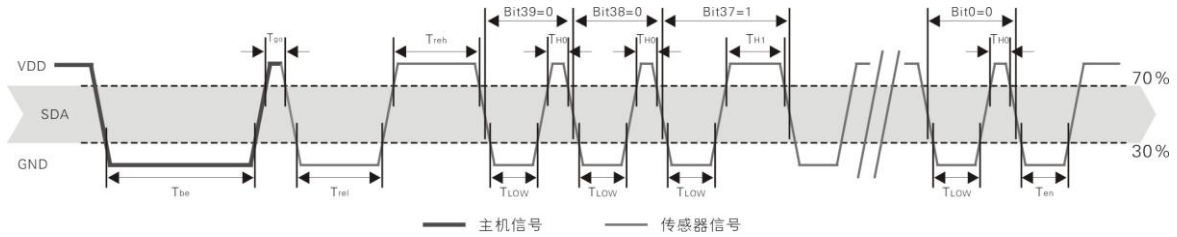


图 21: AM2320 单总线通信时序

注：主机从 AM2320 读取的温湿度数据总是前一次的测量值，如两次测量间隔时间很长，请连续读两次以第二次获得的值为实时温湿度值，同时两次读取间隔时间最小为 2S。

表 10: 单总线信号特性

| 符号 | 参数 | min | typ | max | 单位 |
|-----------|----------------|-----|-----|-----|---------|
| T_{be} | 主机起始信号拉低时间 | 0.8 | 1 | 20 | mS |
| T_{go} | 主机释放总线时间 | 20 | 30 | 200 | μ S |
| T_{rel} | 响应低电平时间 | 75 | 80 | 85 | μ S |
| T_{reh} | 响应高电平时间 | 75 | 80 | 85 | μ S |
| T_{LOW} | 信号“0”、“1”低电平时间 | 48 | 50 | 55 | μ S |
| T_{H0} | 信号“0”高电平时间 | 22 | 26 | 30 | μ S |
| T_{H1} | 信号“1”高电平时间 | 68 | 70 | 75 | μ S |
| T_{en} | 传感器释放总线时间 | 45 | 50 | 55 | μ S |

注：为保证传感器的准确通讯，用户在读取信号时，请严格按照表 10 和图 21 中的参数和时序进行设计

8.3.4 外设读取步骤示例

主机和传感器之间的通信可通过如下三个步骤完成读取数据。

步骤一

AM2320 上电后（AM2320 上电后要等待 2S 以越过不稳定状态，在此期间读取设备不能发送任何指令），测试环境温度湿度数据，并记录数据，此后传感器自动转入休眠状态。AM2320 的 SDA 数据线由上拉电阻拉高一直保持高电平，此时 AM2320 的 SDA 引脚处于输入状态，时刻检测外部信号。

步骤二：

微处理器的 I/O 设置为输出，同时输出低电平，且低电平保持时间不能小于 800 μ s，典型值是拉低 1MS，然后微处理器的 I/O 设置为输入状态，释放总线，由于上拉电阻，微处理器的 I/O 即 AM2320 的 SDA 数据线也随之变高，等主机释放总线后，AM2320 发送响应信号，即输出 80 微秒的低电平作为应答信号，紧接着输出 80 微秒的高电平通知外设准备接收数据，信号传输如图 22 所示：

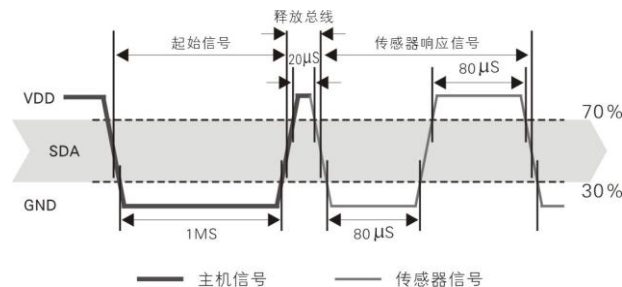


图 22: 单总线分解时序图

步骤三:

AM2320 发送完响应后，随后由数据总线 SDA 连续串行输出 40 位数据，微处理器根据 I/O 电平的变化接收 40 位数据。

位数据“0”的格式为：50 微秒的低电平加 26~28 微秒的高电平；

位数据“1”的格式为：50 微秒的低电平加 70 微秒的高电平；

位数据“0”、位数据“1”格式信号如图 23 所示：

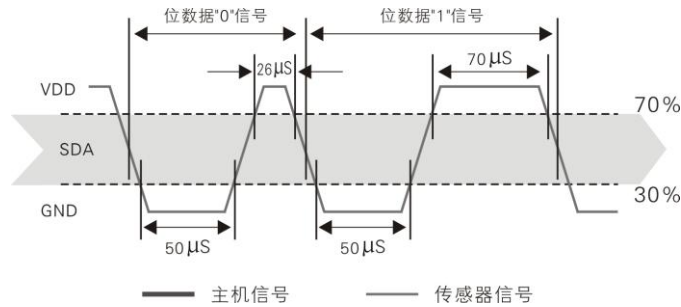


图 23：单总线分解时序图

AM2320 的数据总线 SDA 输出 40 位数据后，继续输出低电平 50 微秒后转为输入状态，由于上拉电阻随之变为高电平。同时 AM2320 内部重测环境温湿度数据，并记录数据，测试记录结束，单片机自动进入休眠状态。单片机只有收到主机的起始信号后，才重新唤醒传感器，进入工作状态。

8.3.5 外设读取流程图

AM2320 传感器读单总线的流程图示意图如图 24 所示，同时我们还提供了 C51 的读取代码示例，需下载的客户，请登录我公司的网站 (www.aosong.com) 进行相关下载，此说明书不提供代码说明。

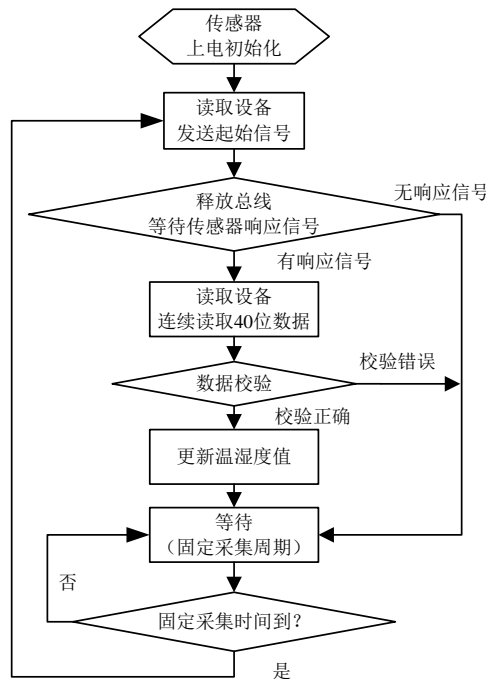


图 24：单总线读取流程图

九、应用信息

1、工作与贮存条件

出建议的工作范围可能导致高达3%RH的临时性漂移信号。返回正常工作条后，传感器会缓慢地向校准状态恢复。要加速恢复进程可参阅“恢复处理”。在非正常工作条件下长时间使用会加速产品的老化过程。

避免将元件长期放在结露和干燥的环境中以及以下环境。

A、盐雾

B、酸性或氧化气体，例如二氧化硫，盐酸

推荐的存储环境

温度：10~40℃ 湿度： 60%RH 以下

2、暴露在化学物质中的影响

电容式湿度传感器的感应层会受到化学蒸汽的干扰，化学物质在感应层中的扩散可能导致测量值漂移和灵敏度下降。在一个纯净的环境中，污染物质会缓慢地释放出去。下文所述的恢复处理将加速实现这一过程。高浓度的化学污染会导致传感器感应层的彻底损坏。

3、温度影响

气体的相对湿度，在很大程度上依赖于温度。因此在测量湿度时，应尽可能保证湿度传感器在同一温度下工作。如果与释放热量的电子元件共用一个印刷电路板，在安装时应尽可能将传感器远离电子元件，并安装在热源下方，同时保持外壳的良好通风。为降低热传导，传感器与印刷电路板其它部分的铜镀层应尽可能最小，并在两者之间留出一道缝隙。

4、光线影响

长时间暴露在太阳光下或强烈的紫外线辐射中，会使性能降低。

5、恢复处理

置于极限工作条件下或化学蒸汽中的传感器，通过如下处理程序，可使其恢复到校准时的状态。在 45℃ 和 < 10%RH 的湿度条件下保持 2 小时（烘干）；随后在 20-30℃ 和 >70%RH 的湿度条件下保持 5 小时以上。

6、配线注意事项

信号线材质量会影响通讯距离和通讯质量,推荐使用高质量屏蔽线。

7、焊接信息

手动焊接，在最高 300℃ 的温度条件下接触时间须少于 10 秒。

8、产品升级

具体请咨询我公司技术部门。

十、许可证协议

未经版权持有人的事先书面许可，不得以任何形式或者任何手段，无论是电子的还是机械的（其中包括影印），对本手册任何部分进行复制，也不得将其内容传达给第三方。本说明手册内容如有变更，恕不另行通知。

奥松电子有限公司和第三方拥有软件的所有权，用户只有在签订了合同或软件使用许可证后方可使用。

十一、警告及人身伤害

勿将本产品应用于安全保护装置或急停设备上，以及由于该产品故障可能导致人身伤害的任何其它应用中。不得应用本产品除非有特别的目的或有使用授权。在安装、处理、使用或维护该产品前要参考产品数据表及应用指南。如不遵从此建议，对可能导致的死亡或严重的人身伤害，本公司将不承担由此产生的人身伤害及死亡的所有赔偿，并且免除由此对公司管理者和雇员以及附属代理商、分销商等可能产生的任何索赔要求，包括：各种成本费用、赔偿费用、律师费用等等。

十二、品质保证

本公司对其产品的直接购买者提供为期 12 个月(一年)的质量保证(自发货之日起计算)。以公司出版的该产品的数据手册的技术规格为准。如果在保质期内，产品被证质量实有缺陷，公司将提供免费的维修或更换。用户需满足下述条件：

- ① 该产品在发现缺陷 14 天内书面通知公司；
- ② 该产品应由购买者付费寄回到公司；
- ③ 该产品应在保质期内。

本公司只对那些应用在符合该产品技术条件的场合而产生缺陷的产品负责。公司对其产品应用在那些特殊的应用场合不做任何的保证、担保或是书面陈述。同时公司对其产品应用到产品或是电路中的可靠性也不做任何承诺。