

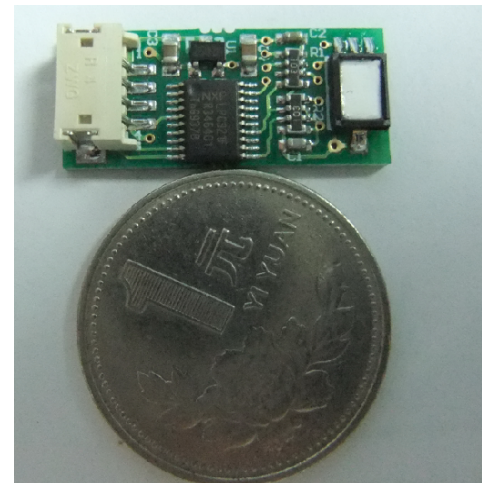
HTD2230-I2C 数字温湿度传感器

【突出优点】

- 全量程温湿度数字化直接输出，提供开发软件包；
- 专利的 solid polymer 结构；
- 突出的防尘性能；
- 高湿防结露加热控制；
- 优于±3%RH和±0.25℃的完全互换；
- 0.03%RH和0.01℃的分辨率；
- 快速的温湿度响应、高湿后迅速恢复；
- 卓越的长期稳定性；
- 低功耗、小尺寸（11mm×27mm）；
- 极高的性能价格比；
- 对光线不敏感，无需避光；
- 每只传感器单独标定数据存于传感器中，断电不丢失，准确度不受传输距离影响。

【典型应用】

- 空调、加湿器、除湿机
- 湿度控制、湿度变送器
- 湿度计、湿度记录仪
- 数据记录
- 复印机
- 时钟、天气预报气压计



【产品概述】

HTD2230-I2C 为 OEM 客户提供了精确、可靠的温湿度数字测量方式。

HTD2230-I2C 自身含有一个 I2C 数字接口，按照 I2C 指令协议，直接输出经温度补偿后的湿度、温度、露点及校验 CRC 信息。用户无需对数字输出进行二次计算，也不需要湿度进行温度补偿，便可得到准确的温湿度信息；免费提供的开发包可嵌入用户代码中，使用户对传感器的掌握和使用变得轻松容易。

对传感器的特殊设计，确保在高湿、灰尘、脏污等恶劣环境湿度测量，较同类产品具有明显的优势。

在雾、霪、雨及结露等环境，因湿度接近饱和，探头被打湿后变得对周围环境不敏感。因此在湿度接近饱和的环境中，要可靠地进行湿度测量变得非常困难。针对这类环境需要去露加热降湿，为此专门设计了传感器的加热功能。在正常的测湿环境，缺省的温湿度检测速度为每秒一次，每次持续时间约 125ms，其它时间传感器处于极低功耗的待机状态。在雾、霪、雨及结露等环境，选择更快的温湿度刷新速度如全速（每秒八次），此时传感器微处理器也全速运行，消耗电流增加至 4-5mA，相当于对传感器周围环境连续加热，结果降低了传感器周围的湿度，亦即降低了传感器探头结露的风险。

对于大多数湿度传感器，会要求应用在洁净的环境，或为传感器安装特殊的防尘罩，这无形中增加了传感器的体积和成本。HTD2230-I2C 电路采取了特殊软硬件设计，将灰尘及脏污的影响较同类产品降低至十分之一；同时，传感器自身带防尘膜，从而最大限度地降低了灰尘及脏污对湿度精确测量的影响。

HTD2230-I2C 数字标定信息储存于 Flash 芯片中。对于批量用户，开放传感器的标定与校准功能，用户使用特殊指令便可对传感器进行批量、快速的二次标定或校准。

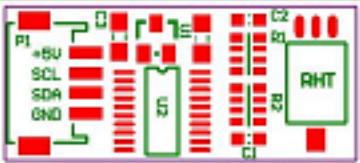
【接口说明】

HTD2230-I2C 数字温湿度传感器作为从机，与主机（用户机）之间通讯采取 I2C 总线方式。

由于湿度对温度特别敏感，如果不是针对持续易结露的高湿环境，必须严格控制传感器的平均功耗，降低传感器自身发热对周围环境湿度值的影响，这就要求传感器在非温湿度测量时，自动转为休眠状态，以降低功耗。通过选择传感器的温湿度输出刷新速率，亦即选择传感器自动唤醒时间间隔，便可控制传感器的功耗。

HTD2230-I2C 工作模式为主动请求式，即传感器自动按照用户选定的速度进行温湿度刷新。温湿度刷新后，其就绪状态在 SDA 信号线上输出低电平。主机读取从机（传感器）信息时，必须先拉低 SCL，给从机一个应答，从机收到应答信号后，释放 SDA 总线使其变高电平。此后主（用户机）、从机（传感器）间便可按标准的 I2C 规约进行通讯。需要注意的是自 SDA 输出温湿度就绪状态的时刻起，用户必须在 125ms 内完成对温湿度信息帧的读取，否则会造成数据的丢失。

引脚	名称	说明
1	VDD	电源, 典型为5.0VDC或3.3VDC
2	SCL	串行时钟线, 也是主机响应传感器要进行数据读取的应答信号线
3	SDA	串行数据线, 也是温湿度数字转换就绪(低电平)信号线
4	GND	地



【技术指标】

参 数	符号	最小	典型	最大	单位
湿 度	测量范围	RH	0	100	%RH
	测量精度		±3	±5	
	分辨率		0.03	0.05	
温 度	测量范围	Tcc	-45	85	℃
	测量精度		±0.25	±0.5	
	分辨率		0.01	0.02	
供电电源	Vs	3.0	3.3/5.0	5.5	VDC
恢复时间(在100%RH存放150小时后)	tr		10		s
湿度迟滞			±1		%RH
长期稳定性			±0.5		%RH/年
平均耗电流	1次/2秒	Is		0.25	mA
	1次/秒			0.5	
	8次/秒			4	
峰值耗电流		Ipeak		4	mA
储存环境	温度	Tstg	-40	85	℃

【可靠性测试】

HTD2230-I2C 经过以下测试: 震动、冲击、高温、高湿、ESD 等。

【输出格式】

主机发出读取传感器(从机)温湿度等信息命令时, 传感器共输出6个字节的数据帧。

温湿度输出数据帧:

湿度	温度	保留	CRC
2 字节	2 字节	2 字节	1 字节

湿度(2字节)输出格式:

高字节

符号位	bit6	bit5	bit4	bit3	bit2	bit1	bit0
S	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰

低字节

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴	2 ⁻⁵	2 ⁻⁶	2 ⁻⁷	2 ⁻⁸

湿度值(%RH)	二进制输出(Binary)	十六进制输出(Hex)
100.0	01100100 00000000	6400
97.3	01100001 01001100	614C
75.3	01001011 01001100	4B4C
55.0	00110111 00000000	3700
32.8	00100000 11001100	20CC
11.3	00001011 01001100	0B4C
0.0	00000000 00000000	0000

温度 (2 字节) 输出格式:

高字节

符号位	bit6	bit5	bit4	bit3	bit2	bit1	bit0
S	2^6	2^5	2^4	2^3	2^2	2^1	2^0

低字节

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}

温度值 (°C)	二进制输出(Binary)	十六进制输出(Hex)
+85	01010101 00000000	5500
+55.5	00110111 10000000	3780
+25.0625	00011001 00010000	1910
+10.125	00001010 00100000	0A20
+0.5	00000000 00001000	0080
0	00000000 00000000	0000
-0.5	11111111 10000000	FF80
-10.125	11110101 11100000	F5E0
-25.0625	11100110 11110000	E6F0
-45.0	11010011 00000000	D300

CRC 校验 (1 字节):

在温湿度数据输出帧的最后一个字节为前 6 个字节的 8bit CRC 校验码, 用于验证收到的信息是否正确。当主机与传感器间引线较长时, 使用 CRC 校验码, 可增强测量的抗干扰能力。CRC 算法参见提供的 C51 软件包。

传感器地址 (1 字节):

传感器的 I2C 地址为 56h(十六进制)

发送命令:

用户针对传感器所使用的命令有 2 个:

64h(十六进制) —— 读温湿度数据帧

82h(十六进制) —— 调整温湿度输出刷新频率 (0: 8 次/秒 1: 4 次/秒 2: 2 次/秒 3: 1 次/秒 4: 二秒一次)

缺省值为 1 次/秒

发送命令方式参见提供的 C51 软件包。

HTD2230-I2C 用户指南

1. I2C 总线简介

HTD2230-I2C 与微控制器的接口形式是 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，请参考飞利浦公司的网站 (<http://www.semiconductors.philips.com/>)，或者是何立民教授编著的《I2C 总线应用系统设计》。下一节还给出了用 C51 编写的 I2C 总线软件包，以供参考学习。

1.1 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，这个总线被称为 Inter-IC 或者 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，被广泛地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，而不需要额外的地址译码器。由于 I2C 是一种二线式串行总线，因此简单的操作特性成为它快速崛起成为业界标准的关键因素。

1.2 I2C 总线的信号线

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 $3\sim 10K\Omega$ 。开漏结构的好处是：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，上拉电阻接 5V 电源就能与 5V 逻辑器件接口，上拉电阻接 3V 电源又能与 3V 逻辑器件接口；因为是开漏结构，所以不同器件的 SDA 与 SDA 之间、SCL 与 SCL 之间可以直接相连，不需要额外的转换电路。

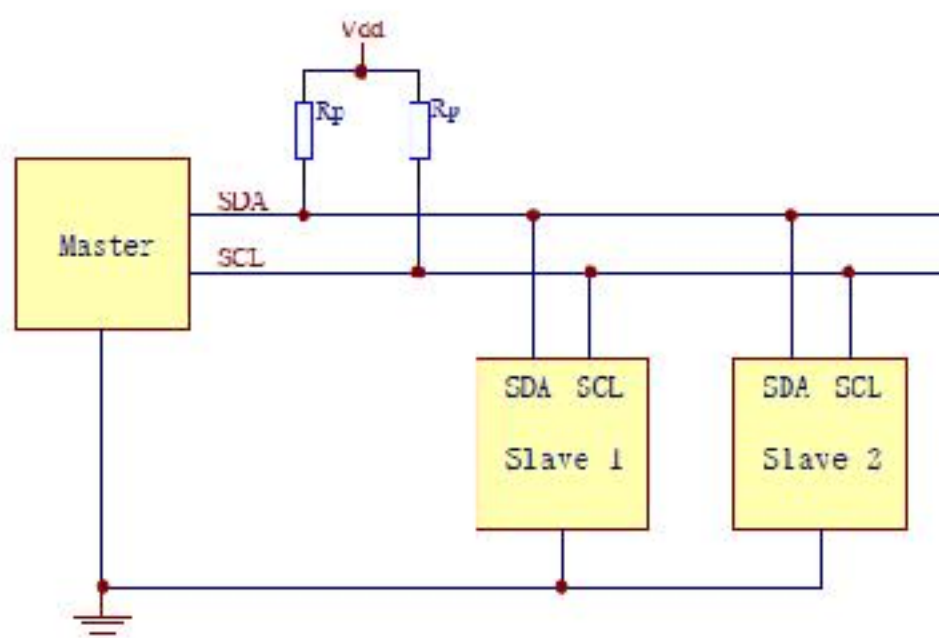


图 1.1 I2C 总线信号连接示意图

1.3 I2C 总线基本概念

- 发送器 (Transmitter)：发送数据到总线的器件；
- 接收器 (Receiver)：从总线接收数据的器件；
- 主机 (Master)：初始化发送、产生时钟信号和终止发送的器件；
- 从机 (Slave)：被主机寻址的器件。

I2C 总线是双向传输的总线，因此主机和从机都可能成为发送器和接收器。如果主机向从机发送数据，则主机是发送器，而从机是接收器；如果主机从从机读取数据，则主机是接收器，而从机是发送器。

1.4 I2C 总线数据传送速率

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 (Standard-mode) 下最快可达 100Kb/s。

1.5 I2C 总线上数据的有效性 (Data validity)

数据线 SDA 的电平状态必须在时钟线 SCL 处于高电平期间保持稳定不变。SDA 的电平状态只有在 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。某些其它的串行总线协议可能规定数据在时钟信号的边沿（上升沿或下降沿）有效，而 I2C 总线则是电平有效。

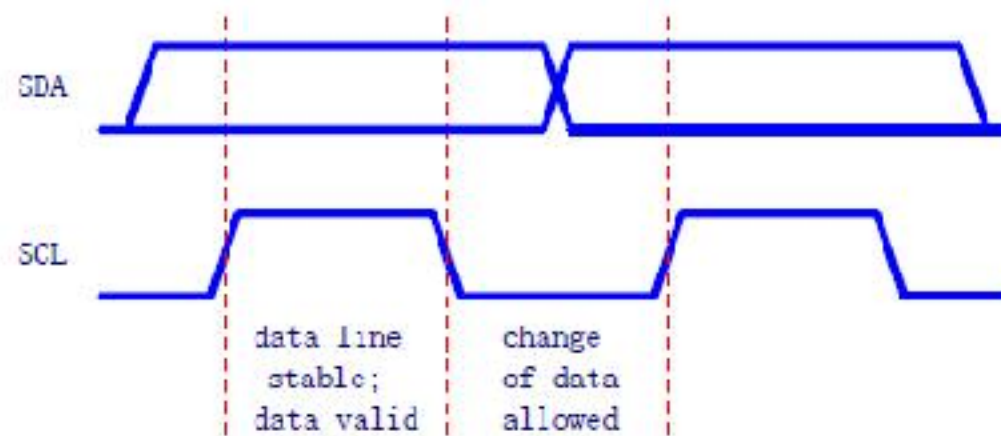


图 1.2 I2C 总线上数据有效性的示意图

1.6 起始条件和停止条件 (START and STOP conditions)

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。

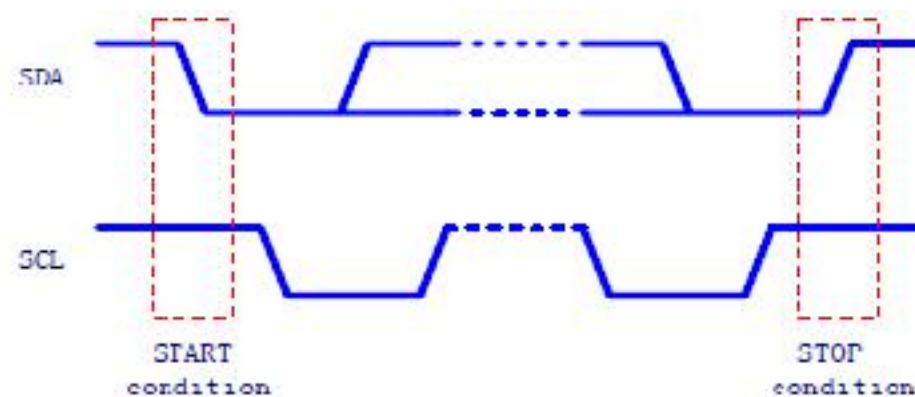


图 1.3 I2C 起始条件和停止条件示意图

1.7 从机地址 (Slave Address)

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。主机是主控制器件, 它不需要器件地址, 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。一般从机地址由 7 位地址位和一位读写标志 (R/W) 组成, 7 位地址占据高 7 位, 读写位在最后。读写位是 0, 表示主机将要向从机写入数据; 读写位是 1, 则表示主机将从从机读取数据。

特别声明: 由于 HTD2230-I2C 数字温湿度传感器为主动输出式, SDA 和 SCL 线分别兼具就绪和应答功能。因此在该 SCL、SDA 总线上不能再接其它 I2C 器件。

1.8 数据传输的基本格式

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, 每个字节之后还要跟一个响应位, 称为应答。

1.9 应答 (Acknowledge)

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。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), 常常简记为 A。发送器发送完 LSB 之后, 应当释放 SDA 线 (拉高 SDA, 输出晶体

管截止)，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。

1.10 基本的数据传输格式示意图



图 1.4 主机向从机发送数据的基本格式



图 1.5 主机从从机接收数据的基本格式

在图 1.4 和图 1.5 中，各种符号的意义为：

- S: 起始位 (START)；
- SA: 从机地址 (Slave Address)，7 位从机地址；
- W: 写标志位 (Write)，1 位写标志；
- R: 读标志位 (Read)，1 位读标志；
- A: 应答位 (Acknowledge)，1 位应答；
- A: 非应答位 (Not Acknowledge)，1 位非应答；
- D: 数据 (Data)，每个数据都必须是 8 位；
- P: 停止位 (STOP)；
- 阴影: 主机产生的信号；
- 无阴影: 从机产生的信号。

应当注意的是，与图 1.5 中的情况不同的是，在图 1.4 中，主机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。

1.11 传输一个字节数据的时序图

为了更清楚地了解 I2C 总线的基本数据传输过程，下面画出了只传输 1 个字节的时序图，这是最基本的传输方式。在图 1.6 和图 1.7 中，SDA 信号线被画成了两个，一个是主机产生的，另一个是从机产生的。实际上主机和从机的 SDA 信号线总是连接在一起的，是同一根 SDA。画成两个 SDA 有助于进一步理解在 I2C 总线上主机和从机的不同行为。

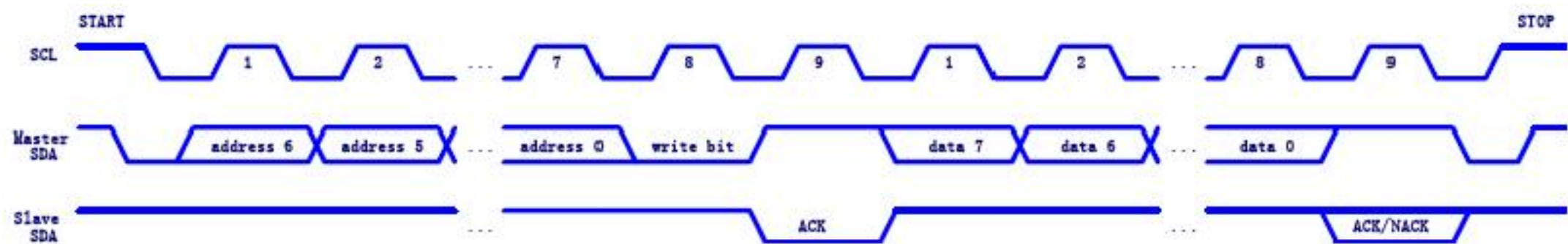


图 1.6 主机向从机发送 1 个字节数据的时序图

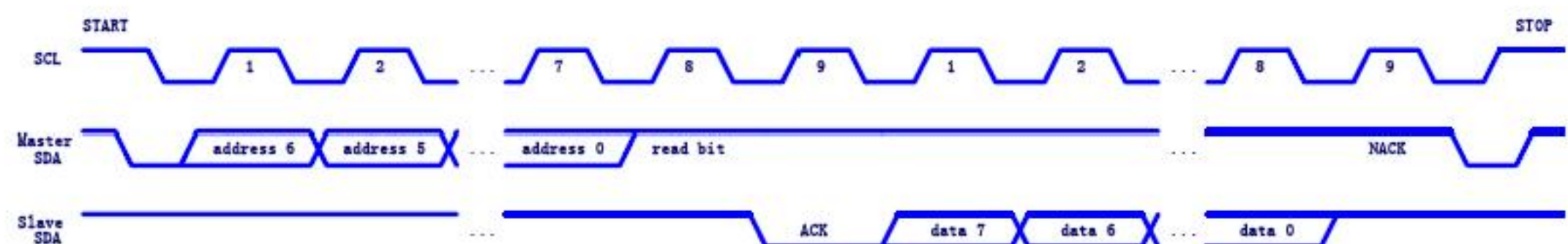


图 1.7 主机从从机接收 1 个字节数据的时序图

1.12 传输多个字节数据的时序图

主机连续向从机发送或从从机接收多个字节数据的情况也很容易理解，下面直接给出相关时序图。

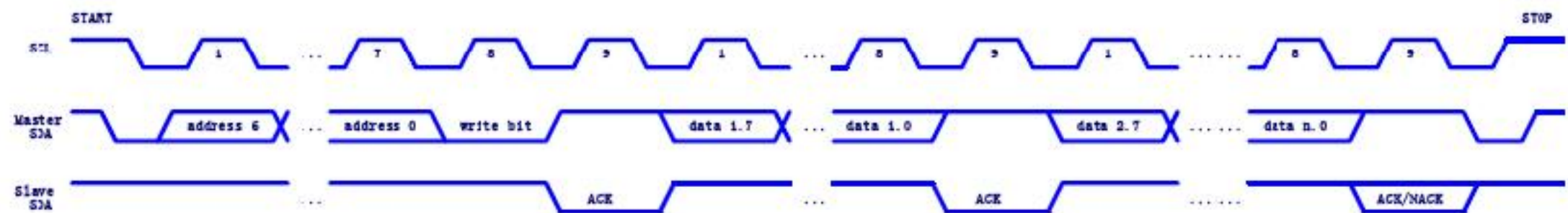


图 1.8 主机向从机连续发送多个字节数据的时序图

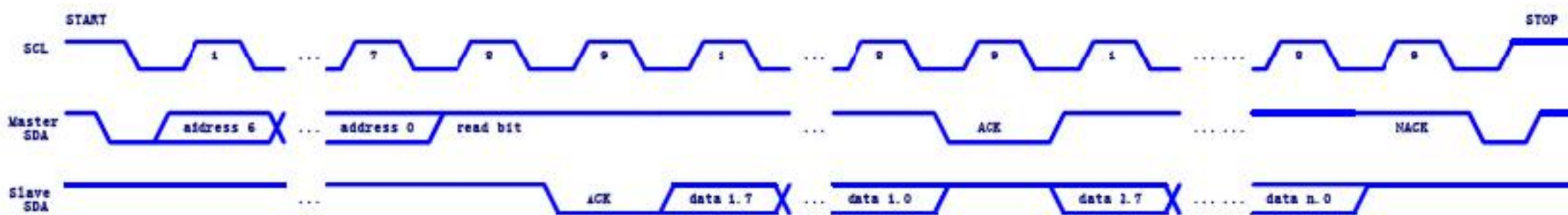


图 1.9 主机从从机连续接收多个字节数据的时序图

1.13 重复起始条件 (Repeated START condition)

主机与从机进行通信时，有时需要切换数据的收发方向，例如访问某一具有 I2C 总线接口的 E2PROM 存储器时，主机先向存储器输入存储单元的地址信息（发送数据），然后再读取其中的存储内容（接收数据）。在切换数据的传输方向时，可以不必先产生停止条件再开始下次传输，而是直接再一次产生开始条件。I2C 总线在已经处于忙的状态下，再一次直接产生起始条件的情况被称为重复起始条件。重复起始条件常常简记为 S_r 。正常的起始条件和重复起始条件在物理波形上并没有什么不同，区别仅仅是在逻辑方面。在进行多字节数据传输过程中，只要数据的收发方向发生了切换，就要用到重复起始条件。

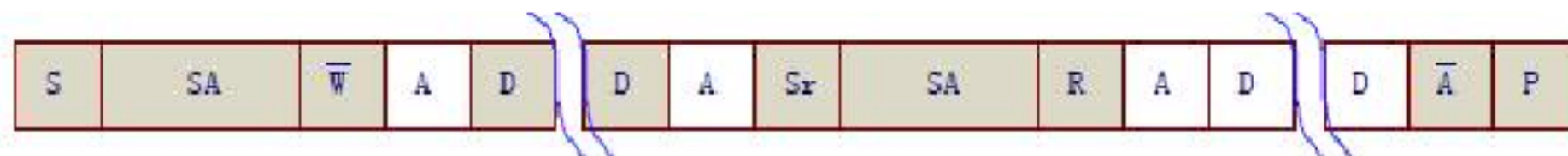


图 1.10 带有重复起始条件的多字节数据传输格式示意图

图 1.10 给出了带有重复起始条件的多字节数据传输格式示意图，图中的各种符号的意义与第 1.10 节中的相同。要特别注意图中重复起始条件 S_r 的用法。如果读者有兴趣的话，可以自行画出其对应的时序图。

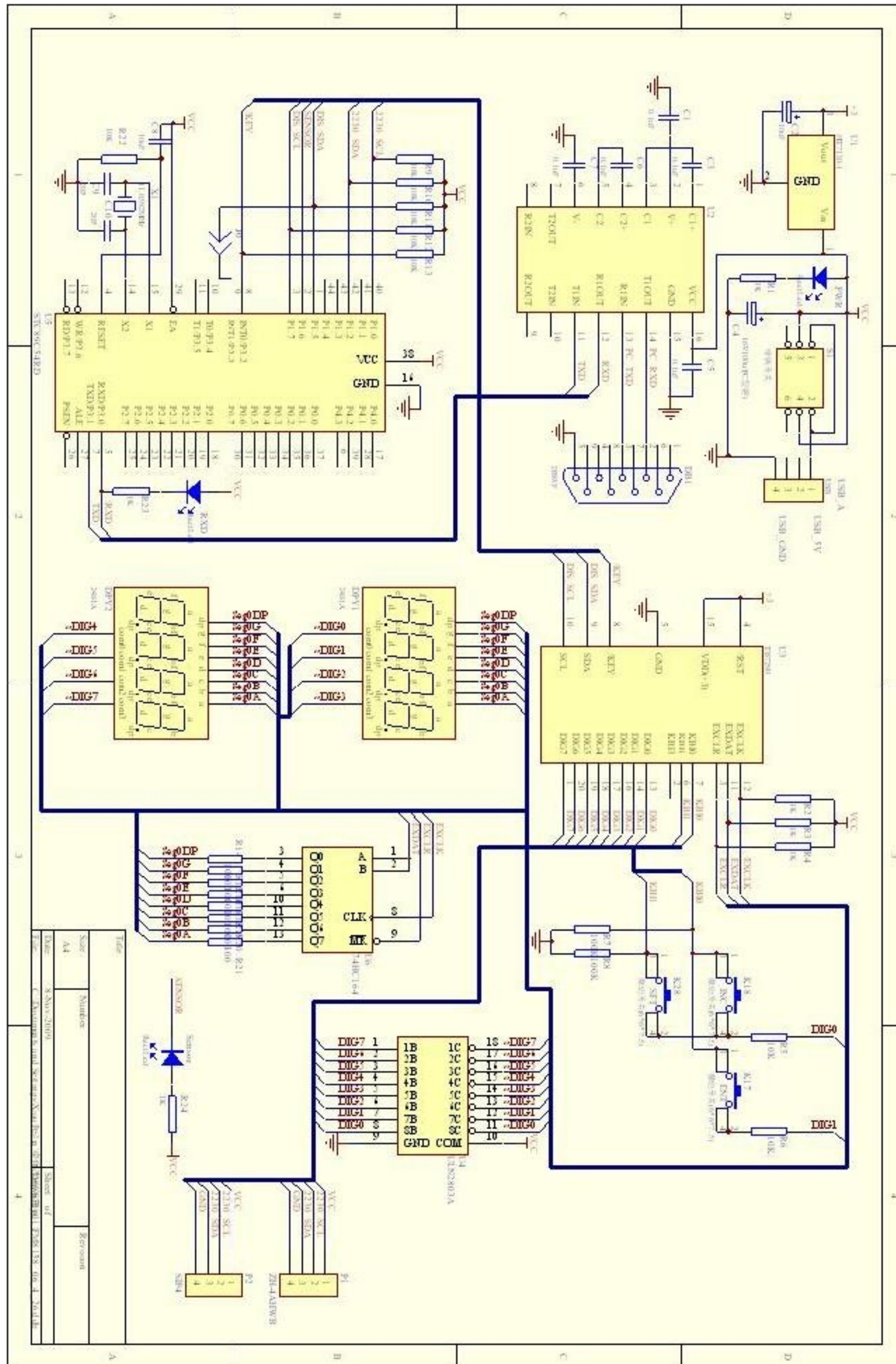
1.14 无子地址器件与有子地址器件

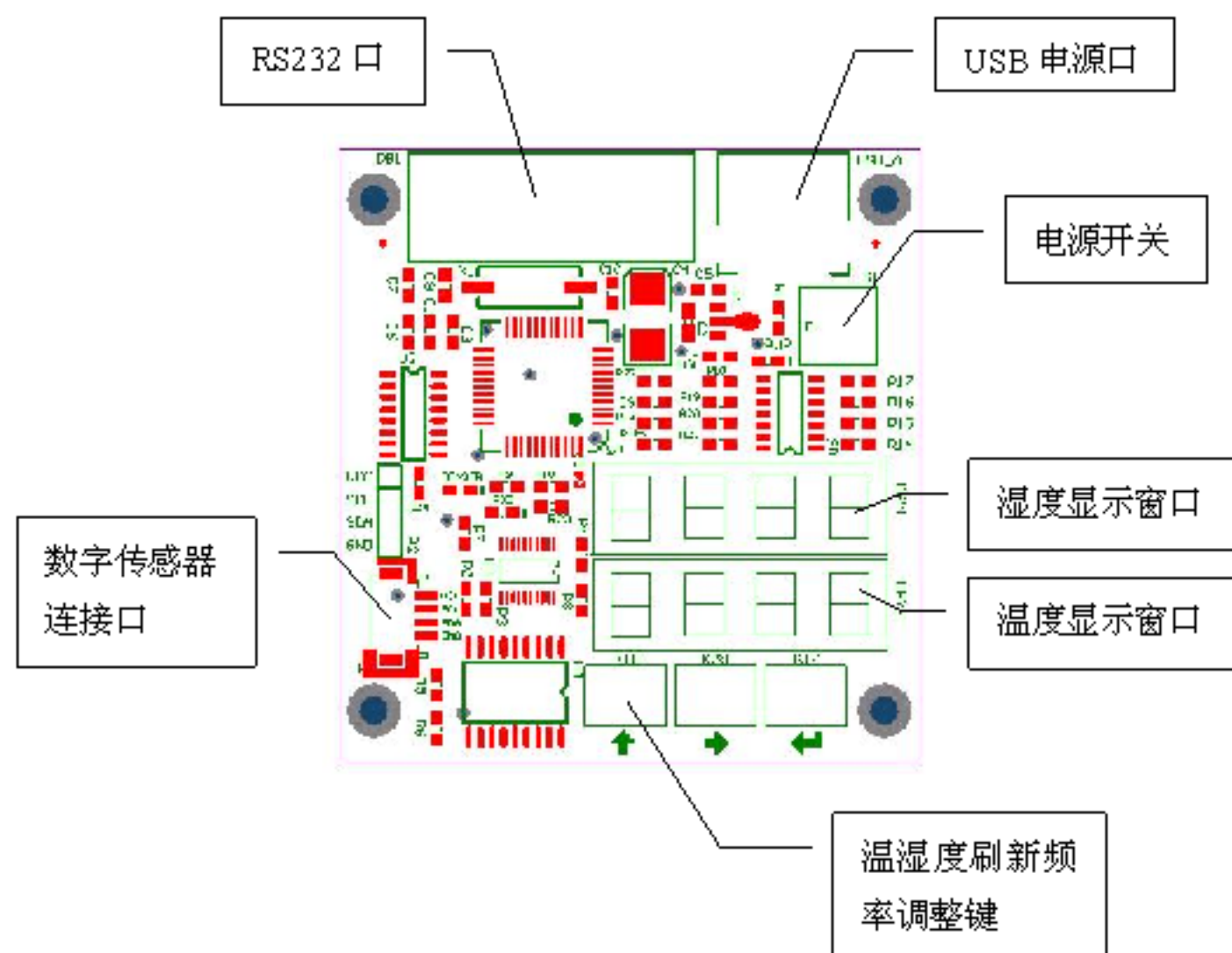
带有 I2C 总线的器件除了有从机地址 (Slave Address) 外，还可能有子地址 (Sub-Address)。从机地址是指该器件在 I2C 总线上被主机寻址的地址，而子地址是指该器件内部不同部件或存储单元的编址。例如，带 I2C 总线接口的 E2PROM 就是拥有子地址器件的典型代表。另外一些器件（只占少数）内部结构比较简单，可能没有子地址，只有必须的从机地址。与从机地址一样，子地址实际上也是像普通数据那样进行传输的，传输格式仍然是与数据相统一的，区分传输的到底是地址还是数据要靠收发双方具体的逻辑约定。子地址的长度必须由整数个字节组成，可能是单字节（8 位子地址），也可能是双字节（16 位子地址），还可能是 3 字节以上，这要看具体器件的规定。

在第 3 章的 I2C 总线软件包中，已经同时考虑到了无子地址器件和有子地址器件的情况。

2. HTD2230-I2C Demo 演示板

HTD2230-I2C Demo 演示板使用的单片机是 STC89C54RD+, 与 89C52 兼容, 无需专用编程器可 ISP 编程, 编程软件包可从 www.MCU-Memory.com 下载。





3. HTD2230-I2C 数字温湿度传感器 Keil C51 开发软件包

3.1 驱动程序软件包

由于 HTD2230-I2C 数字温湿度传感器和用于显示和键盘管理的芯片均为 I2C 器件，因此 89C52 兼容单片机 C51 驱动程序 I2C 模拟软件包分别由 I2C_HTD_DRIVER.H、I2C_HTD_DRIVER.C 和 I2C_DIS_DRIVER.H、I2C_DIS_DRIVER.C 组成。头文件 I2C_HTD_DRIVER.H、I2C_DIS_DRIVER.H 包括了信号线 SDA 和 SCL 的 I/O 接口定义和用户函数的声明，C 语言文件 I2C_HTD_DRIVER.C、I2C_DIS_DRIVER.C 是这些函数的具体实现。这是一个用 C51 模拟 I2C 总线协议的精简版本，只考虑一主多从模式，不考虑多主模式，也不考虑时钟同步等问题。要想更清楚地了解本程序的细节，请参考 Philips 公司的相关协议标准。

3.1.1 HTD2230_I2C 驱动

- 把文件“I2C_HTD_DRIVER.H”和“I2C_HTD_DRIVER.C”一起复制到您的工程文件夹下；
- 根据实际电路，修改 I2C 总线两根信号线 SCL 和 SDA 的定义；
- 通过宏定义 I2C_HTD_DELAY_VALUE 调整 I2C 总线的速度使其符合实际需要；
- 把文件“I2C_HTD_DRIVER.C”添加进工程中，在需要的地方包含头文件“I2C_DRIVER.H”；
- 在 main()函数的开始处，应当调用一次初始化函数 I2C_HTD_Init();
- I2C_HTD_Puts()和 I2C_HTD_Gets()是 I2C 总线综合读写函数，请看清注释后再使用；
- 所有全局性质的标识符都以“I2C_HTD_”开头，可有效避免命名冲突问题；
- 注意：从机地址采用 7 位纯地址表示，不含读写位，即第 0~6 位是地址，第 7 位无效。这可能与其它程序的规定不同。

3.1.2 TB7290 显示及键盘管理驱动

- 把文件“I2C_DIS_DRIVER.H”和“I2C_DIS_DRIVER.C”一起复制到您的工程文件夹下；
- 根据实际电路，修改 I2C 总线两根信号线 SCL 和 SDA 的定义；
- 通过宏定义 I2C_DIS_DELAY_VALUE 调整 I2C 总线的速度使其符合实际需要；
- 把文件“I2C_DIS_DRIVER.C”添加进工程中，在需要的地方包含头文件“I2C_DRIVER.H”；
- 在 main()函数的开始处，应当调用一次初始化函数 I2C_DIS_Init();
- I2C_DIS_Puts()和 I2C_DIS_Gets()是 I2C 总线综合读写函数，请看清注释后再使用；

- 所有全局性质的标识符都以“I2C_DIS_”开头，可有效避免命名冲突问题；
- 注意：从机地址采用 7 位纯地址表示，不含读写位，即第 0~6 位是地址，第 7 位无效。这可能与其它程序的规定不同。具体源代码及说明参见随机提供的文档注释。

3.2 HTD2230-I2C Demo 演示板 Keil C51 应用程序软件包

文件 HTD2230.H 定义了读取数字温湿度信息的命令、结构体和函数声明。HTD2230.C 相应给出了执行有关命令的函数。具体源代码及说明参见随机提供的文档注释。

将 PC 机超级终端通讯波特率设置为 9600，数据位(D): 8，奇偶校验(P): 无，停止位(S): 1，数据流控制(F): 无；Demo 演示板由 PC 机 USB 口供电，与 PC 机 RS232 连接，便可与 PC 机通讯，温湿度数据在 PC 机屏上显示。

可直接运行已定义好设置的超级终端文件：9600



4. 文档信息

4.1 参考文献

- [1] I2C 总线规范 Philips 半导体公司. 2000
- [2] 马忠梅, 戚军, 刘滨, 马岩. 《单片机 C 语言 Windows 环境编程宝典》. 北京航空航天大学出版社. 2003
- [3] 何立民. 《I2C 总线应用系统设计》. 北京航空航天大学出版社. 1995