

基于可编程数字电位器在 AVR 单片机中的应用 I2C

AVR 嵌入式单片机具有丰富的硬件、软件资源，其中的串行 I2C 接口能满足很多应用场合的要求，两个 AVR 单片机通过 I2C 总线直接连接就可实现单片机相互通信；AVR 单片机还可以和任何具有 I2C 总线接口的外设直接连接而无须其它硬件电路支持。而 X9221 系列可编程数字电位器在智能测试设备上应用非常广泛，通过 I2C 总线可以简单地构成单片机与各种外设之间乃至与计算机之间的通信，建立友好的人机界面联系。硬件设计简单、灵活，只需要将所有设备的 SDA 和 SCL 信号线分别并联在一起并加上拉电阻即可，有助于提高设备的自动化水平、可靠性、稳定性及电气装配的工艺性。AVR 单片机和 X9221 系列可编程数字电位器都有内置的 E2ROM 单元，可以非常方便地为用户保留一些工艺参数；X9221 系列电位器 0~63 级的变化可以将电位器调节到手动无法实现的平滑级别，调节过程中不会产生噪声且寿命长、不受机械振动污染潮湿影响等。

2 X9221 系列数字电位器介绍

2.1 电气特性及硬件结构原理

X9221 系列电位器共有 15 种规格型号，有双列直插 DIP20 和表面贴装 SOIC20 两种封装。电源电压民品级、工业级为 4.5~5.5V，军品级为 2.7~5.5V；一组封装芯片内有两个数字电位器，参见图 1，X9221 系列电位器封装及引脚功能。内置的 E2ROM 单元，可以在掉电时将数据很好地保存，上电时自动加载到自己的 RAM 单元，电位器接口是标准的 I2C 总线，其中的数据寄存器可通过 I2C 总线进行读写操作；电位器滑动端 (VW0 / RW0, VW1 / RW1) 相当于普通电位器中间抽头，是无摩擦“触点”。

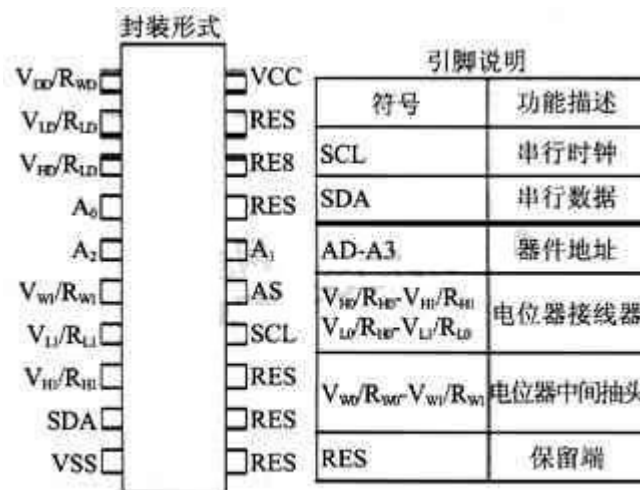


图 1 X9221 系列数字电位器封装及引脚功能

X9221 系列电位器总电阻配置有 3 种阻值 2 kΩ、10kΩ、50kΩ，用户可根据自己的设计需求来选型。每一个芯片有 A0~A3 四位二进制编程的器件地址以

区分 I2C 总线上接入的不同外设，因此，一条总线上最多可接入 16 个 X9221 器件。器件内部有两组类型的寄存器“DATA”和“WCR”，他们的访问则由串行数据线上的命令字来确定。电阻阵列通过内部编码可以有 64 种状态，0~63 种状态的连续变化，相当于电位器中心抽头从普通电位器的一端滑动到另一端；所有的寄存器都可以通过 I2C 总线进行双向操作，即可读可写。

“在线”实时的调节电位器“中间抽头位置”可以有 3 种方法：通过 I2C 总线向 WCR 寄存器写数据(串行加载)、通过对相应的 DATA 类寄存器直接写数据(并行加载)、以及增量减量命令(下面会讨论到)直接写数据；概念上电位器“中间抽头位置”可以由“WCR”寄存器来替代，其另一类寄存器可以由“DATA”寄存器来替代。

2.2 数字电位器 I2C 总线时序及指令

2.2.1 一般 I2C 总线通信时序

X9221 系列数字电位器其接口是按标准的 I2C 总线设计的，因此，硬件连接非常简单，只需把串行时钟线 SCL、串行数据线 SDA 与之对应相连即可。其通信完全符合 I2C 总线协议要求，串行时钟线 SCL、串行数据线 SDA 按照规定的协议产生一序列脉冲串，进而完成传输一组数据的任务。参见图 2 I2C 总线通信时序。

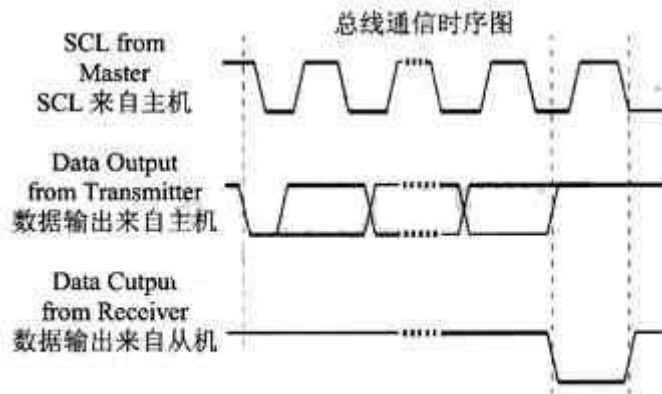


图 2 I2C 总线通信时序

X9221 系列数字电位器完全以从机的身份出现在 I2C 总线上，它不可以主机的身份出现在 I2C 总线上，也就是说对其寄存器的读写操作都受控于主机，这一点一定要在硬件和软件设计上留意。当主机发出“START”信号后，从机(X9221A)芯片即将拉高 SDA 线，表明放弃数据线权限由主机控制，主机发送一个字节后再由从机(X9221A)芯片自动拉低 SDA 线，表明数据收到，如果此时从机将 SDA 线拉不低，表明从机没有正常接受数据，主机必须启动下一个写周期。可以理解为主机每发送一个字节后都会等待一个 ACK 回答响应信号，否则，主机认为从机(X9221A)没有正常接收数据。

2.2.2 X9221 指令表

表 1 是 X9221 所有指令说明,其中前 4 种指令的正常执行需要在串行数据线上至少有 3 个步骤:①通过数据线写从机地址;②写指令;③写数据。才能完成一个寄存器的读或写,这 4 种指令适用于 X9221 三字节时序格式操作。如果是写在 WCR 类型的寄存器中,掉电后数据丢失,写在 DATA 类型的寄存器中,掉电后数据存在其中;全局转换的 4 种指令的正常执行需要在串行数据线上至少有 2 个步骤:①通过数据线写从机地址、②写指令,才能完成所有寄存器之间的数据交换,适用于 X9221 两字节时序格式操作;最后一种“+/-”指令相当于“在线”上下调节电位器中间抽头,比较直观,很好理解。“+/-”指令仅访问 WCR 类型寄存器,且只是写操作,如果数据线 SDA 保持高电平,下一个时钟信号 SCL 周期到来,WCR 寄存器数据“+1”。如果数据线 SDA 保持低电平,下一个时钟信号 SCL 周期到来,WCR 寄存器数据“-1”。然而,不管哪种命令,只要写 DATA 类型的寄存器(芯片内的 E2ROM 存储单元),写操作要花费大约 10 ms 时间才能完成,在设计通信软件时一定要特别留意。

表 1 X9221 指令表

指令	指令格式									操作说明
	I ₂	I ₁	I ₀	0	P ₀	R ₁	R ₀			
读 WCR Register	1	0	0	1	0	1/0	N/A	N/A	N/A	WCR Register 到 SDA
写 WCR Register	1	0	1	0	0	1/0	N/A	N/A	N/A	SDA 到 WCR Register
读 Data Register	1	0	1	1	0	1/0	1/0	1/0		Data Register 到 SDA
写 Data Register	1	1	0	0	0	1/0	1/0	1/0		SDA 到 Data Register
转换 Data Register 到 WCR Register	1	1	0	1	0	1/0	1/0	1/0		Data Register 到 WCR Register
转换 WCR Register 到 Data Register	1	1	1	0	0	1/0	1/0	1/0		WCR Register 到 Data Register
全局转换 Data Register 到 WCR Register	0	0	0	1	N/A	N/A	1/0	1/0		P0 和 P1 Data Register 到对应 WCR Register
全局转换 WCR Register 到 Data Register	1	0	0	0	N/A	N/A	1/0	1/0		P0 和 P1 WCR Register 到对应 Data Register
+/-WCR Register	0	0	1	0	0	1/0	N/A	N/A	N/A	WCR Register 加/减 1

3 AVR 系列单片机与数字电位器硬件和软件设计

3.1 X9221 数字电位器与 AVR 系列单片机硬件接口

X9221 数字电位器与 AVR 系列单片机硬件接口如图 3 所示。硬件连接非常简单,只需将所有设备串行数据线 SDA、串行时钟线 SCL 相连接并接 10 kΩ 左右的上拉电阻即可。连接在 I2C 总线上的逻辑电平为“线与”逻辑关系,只要有一个设备将其拉低,总线上即出现低电平,当所有设备总线都悬挂起时呈现高阻状态。主机发送模式就是向其中一个外设写入数据,首先产生一个启动信号“START”,“START”发送成功后才向外设发送设备地址,地址发送成功后,再发送数据,数据可以是 1~N 个字节,所有数据发送完成,最后发送“STOP”完成主机写的过程。主机接收模式也是一样的,过程类似。对 X9221 来说地址发送完成后,必须发送

指令模式，告诉 X9221 目前将做什么，指令的含义上面已提到，指令模式后是一组数据，其它过程一样。

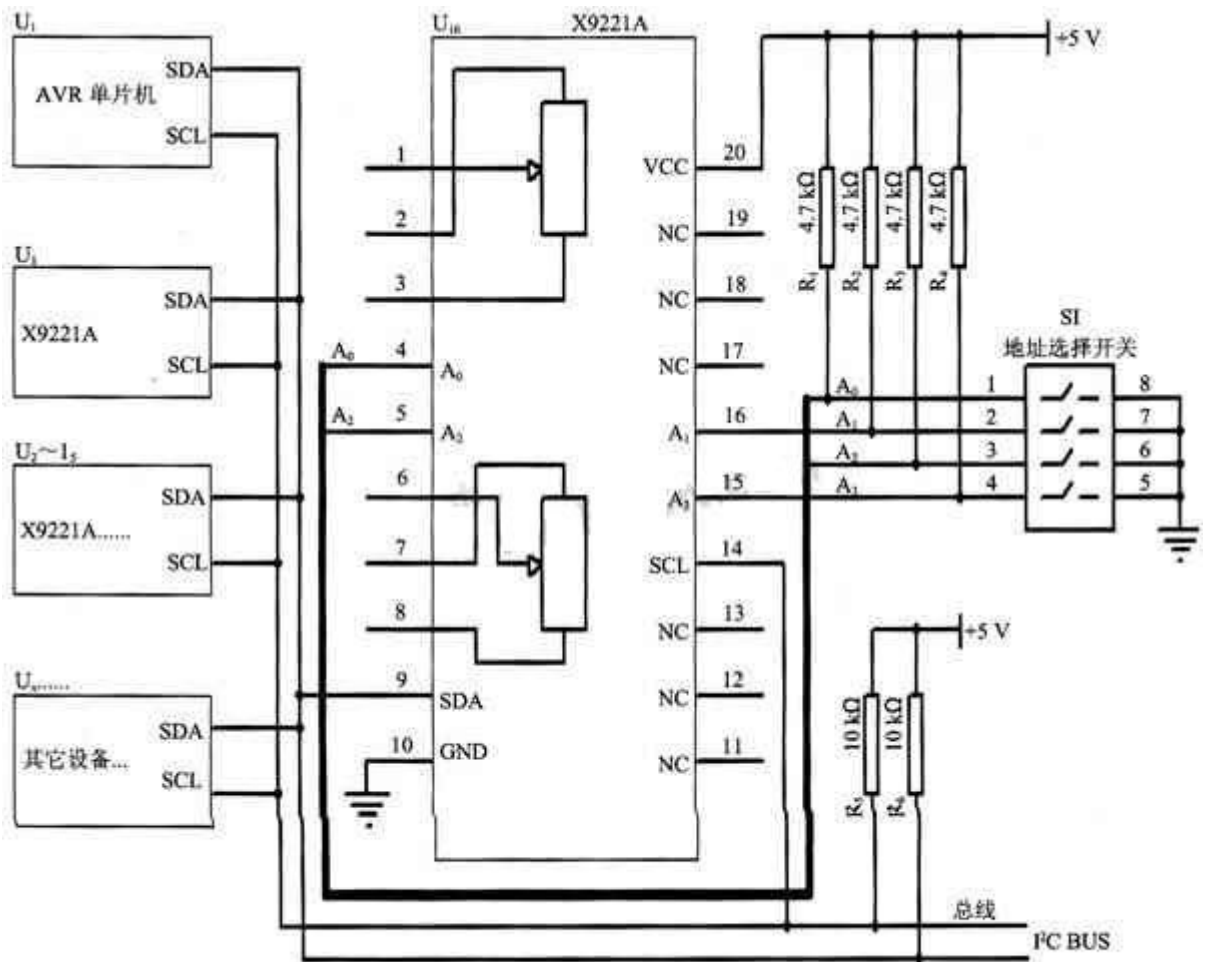


图 3 X9221 数字电位器与 AVR 系列单片机硬件接口图

WeeQoo 维库

为了使 I2C 总线处于正常状态，所有总线上的设备必须加电，如果有一个设备没有加电，总线将处于不正常工作状态。在总线上有几个设备同时欲将发送数据时，竞争中某主机发现仲裁失败后，应立即回到从机状态或放弃总线请求，保证获取总线控制权的主机正常发送数据；不同的主机有不同的总线时钟频率，通过 SCL 信号“线与”来保证，即高电平为高电平中最短的，低电平为低电平中最长的，也就是说协议“照顾”了速度最慢的设备；集结在总线上所有的传送必须包含相同数目的数据包，否则，多主机系统中仲裁结果无法定义，这在设计上要特别注意。

3.2 X9221 数字电位器与 AVR 系列单片机通信软件设计

下面是一个通过查询由 [ATMEGA16](#) 单片机向 X9221 发送数据的例程，在这个例程中需要注意的是 TWINT 标志利用软件写“1”进行清零(清除硬件置位标志)，而当前发送完成后由硬件自动将 TWINT 标志置“1”，TWINT 标志是不能通过硬件

清零的；每次通过检查 TWINT 标志判断当前发送是否完成，通过读取状态寄存器 TWSR 的值判断发送的数据是否正确。

START:

LDI R16, \$A4; 设置 AVR 单片机 I2C 总线控制寄存器

OUT TWCR, R16; 软件清除 TWINT 标志 WAIT1:

IN R16, TWCR; 读取控制寄存器 TWCR

SBRS R16, 7; 等待硬件置位 TWINT 标志

RJMP WAIT1; TWINT 标志没置位，循环检查等待

IN R16, TWSR; 读取状态寄存器 TWSR

ANDI R16, \$F8; 屏蔽无效位

CPI R16, \$08; 检查“START”信号是否发送成功

BRNE ERROR; “START”信号发送未成功，转入出错处理

LDI R16, \$56; 发送 X9221 器件地址 56=“5”器件标识，“6”器件地址

OUT TWDR, R16

LDI R16, \$84

OUT TWCR, R16; 软件清除 TWINT 标志

IN R16, TWSR

ANDI R16, \$F8

CPI R16, \$18; 检查 X9221 器件地址是否发送成功

BRNE ERROR; X9221 器件地址发送未成功，转入出错处理

LDI R16, \$2F; 设置所发送的数据(0~63 有效，例中为 47)

OUT TWDR, R16; 数据输出到数据寄存器 TWDR

LDI R16, \$84

OUT TWCR, R16; 软件清除 TWINT 标志; 检查数据“2F”是否发送完成, 类似于上述循环继续

LDI R16, \$94; 发送“STOP”信号, 一次‘\$2F’数据发送完成

OUT TWCR, R16

ERROR: . . . ; 出错处理过程

END

在此例程中, 器件的地址对 X9221 来说为 56, 其中“5”是器件本身固有的以区别于其它类型的设备, 而“6”是设计硬件时规定的, 可以是“0~F”任何十六进制数值; 例程中 AVR 单片机仅作为主机使用且向 X9221 发送数据, 其实从 X9221 中读取当前数据也是一样的。本例程采用查询方式, 程序显得繁琐, 但是如果采用中断方式的话, 程序就会显得简单多, 只要标志 SREG 寄存器中“I”位和 I2C 总线控制寄存器中的“TEWIE”置“1”, 即中断使能有效, 当“TWINT”标志置“1”立即产生中断请求, 表明完成当前数据发送完成, 可以准备下一个字节数据发送或停止发送等操作。

4 X9221 可编程数字电位器在半导体专用设备一些应用研究

4.1 用于测量微小电压变化

图 4 是微小电压测量电路模型, 用于芯片键合设备中去检测吸头上是否有芯片吸附以及吸头是否堵塞、或者芯片是否丢失等现象。当内径为大约 0.1~0.15 mm 的吸头去吸附一个小芯片时, 如果芯片较透明, 光敏传感器检测出来的电压变化较小, 一般在 10~50 mV 间; 同样当吸头吸附一个小芯片时, 气路真空的压力也会产生变化, 这种压力的变化以传感器电压变化输出。为了改变电位器“中间抽头”以便与传感器检测输入电压相匹配, 计算机通过 [RS232](#) 接口向单片机发送数据, 单片机收到数据后转发给 X9221 可编程数字电位器以改变基准电压值。比较基准电压 U_{Σ} +按下式确定:

$$U_{\Sigma+} = (3.5/63)N \text{ (V)} \quad (1)$$

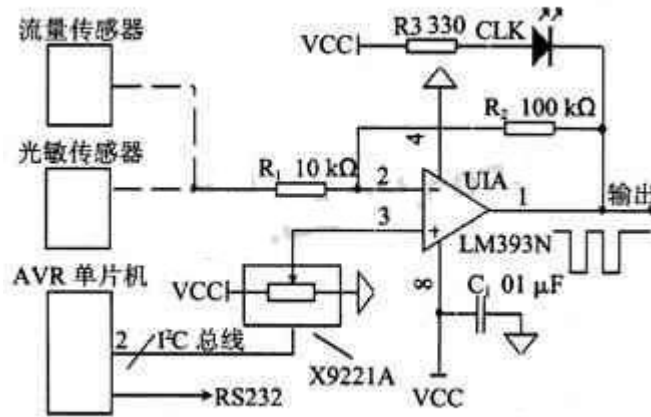


图 4 微小电压测量电路检测芯片丢失状态

$U_{\Sigma+}$ 是 LM393 运放同相输入端电压(在这种状态下, 考虑到前级传感器输出基本上处于放大状态, 所用传感器电源电压为+5 V, 最高输出电压按 3.5 V 计算)

N 是 0~63 共 64 种状态变化值, 那么, 当 $U_{\Sigma+}$ 从 0~3.5 V 变化时, 最小分辨率可以达到 55 mV 左右, 用手工进行一般电位器调节达到这样的分辨率是难以掌控的。

基准电压(即 $U_{\Sigma+}$)通过机器的人机界面可以“直接调节”, 如果将电位器数值进行标定, 随时还可以看到当前的基准电压大小。传感器输出电压与基准电压比较, 使电压比较器输出反应当前的状态, 不同的时刻高低电平代表不同的意义, 如“吸头阻塞”、“芯片丢片”、“真空不足”等。

4.2 测量微小电流变化

图 5 是微小电流测量电路模型, 用于引线键合设备中去检测断线、连线、短路等情况, 被测器件是一个半导体元件, 当在焊盘上键合上一根金线时, 通过检测漏电流来判断这条金线与芯片键合过程中是否存在“断线”、“连线”、“短路”等情况。计算机通过界面操作发送指令以改变 X9221 可编程数字电位器的阻值, 进而改变电压源的放大倍数以改变加在被测元件的电压, 从而达到适应不同品种的半导体器件性能要求。

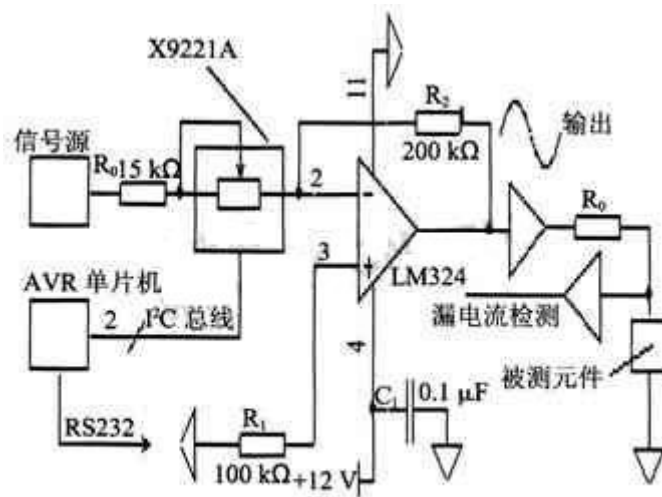


图 5 微小电流测量电路检测芯片断线

设：信号源的输出为 U_s ，运放 LM324 输出为 U_0 ，X9221 电阻为 R_x ，被测元件阻抗为 R_z ，被测元件流过的漏电流为 I_0 ，则用以下两个表达即可表达它们之间的关系：

$$U_0 = -R_z / (R_0 + R_x) U_s \quad (2)$$

$$I_0 = U_0 / (R_0 + R_z) \quad (3)$$

通过主机界面改变 R_x 可编程电阻值，即可以改变 U_0 ， U_0 的改变等于改变了 I_0 ，而 I_0 的改变等于改变了流过被测器件的最大允许电流，从而保护了被测器件不会因为检测漏电流而损坏，通过检测被测器件上施加的电压和 U_0 之差值即可判断漏电流大小，从而检测金线是否与被测器件焊盘点键合上。 I_0 是根据不同器件在工艺参数上需要经常调节的量，以适应不同场合的要求。

5 结束语

可编程数字电位器的最大优点在于直接可以和带有 I2C 总线单片机相连而无需特殊设计，上位机可以随机读取电位器当前设定值。利用它的这些优点，可以提高设备仪器的智能化水平，特别是在带有 I2C 总线的嵌入式单片机中应用十分灵活简单。随着电子技术的飞速发展，人们对设备、仪器以及家用电器的追求已不再仅仅满足功能使用上，而是在产品应用的人性化上要求越来越高，由于大多数自动化设备上，都具有友好的人机界面，人们通过计算机界面想完成所有操作，比如调节一个电位器以调节电流、电压或者电机速度、转矩、频率等物理量，利用可编程数字电位器完全可以通过界面完成。适时采用可编程数字电位器不但可以降低成本、简化电路设计、提高可靠性，而且可以使设计更加人性化。另外，AVR 单片机可以通过 JTAG 接口完成仿真调试、下载程序；片内有 FLASH 和 E2ROM 存储单元，有标准的串行接口、I2C 总线接口、SPI 接口增强了其硬件功能；支持 C 语言编程，便于掌握 C 语言者无需太多地了解硬件就能进行一些编程。本文探讨了 AVR 单片机、X9221 可编程数字电位器及其接口和软件编程的一些实际应用，许多问题是笔者在应用中的经验，可能会有些错误，希望读者批*指正。