|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 小区安防，烟雾，红外，温度，煤气。 **引言**  中国安居工程需要大力发展安全文明小区建设，严峻的社会治安形势，需要建立并完善安全文明小区防盗报警网络系统体系。如何解决在当前每个家庭经济承受能力有限的情况下，建设满足防范功能及可靠性需求的安全文明小区防范防盗报警网络系统，是当前急需解决的课题。本方案是在考虑以上因素的基础上，所设计的可行实施方案。  小区安全防范报警系统是智能小区实现安全管理的重要系统,主要包括防盗报警、煤气泄漏报警、消防报警等。小区管理极为重要的内容是确保住宅­,住户安全,生活中,人人都可能出现一些意想不到的求助情况,现代居住的格局,邻里常年不来往已是常事,家庭生活稳密性、封闭,性越来越强。因此,小区安全防范­及报警系统是具有先进的设计和设备,并为小区住户的安全提供保障的必要系统。  我国的安防自动报警控制系统经历了从无到有、从简单到复杂的发展过程，其智能化程度也越来越高。在小区内的每个住户单元安装一台报警主机，住户可选择安装在住户门口、窗户处安装声检、紧急求助，烟雾/煤气探头、瓦斯探头，等报警感知设备，报警主机通过总线与管理中心的电脑想连接，进行安防信息管理。如果发生盗贼闯入、抢劫、烟雾、燃汽泄露，声音过高等紧急事故，传感器就会立即获知并由报警系统即刻触发声光警报以有效恫吓企图行窃的盗贼；系统还会迅速向报警中心传送报警信息；报警中心接到警情后立即自动进行分辨处理，迅速识别判定警报类型、地点、用户，中心据此派出机动力量采取相应解救措施；系统具备24小时防破坏等并自我监视，一旦有任何被破坏的迹象也会即刻报警。总之，无论白天黑夜，你离家在外还是在家休息，电子保安时时刻刻保护的安全，这正是能为家人、家庭、财产所做的最有效的安全防盗保护措施。  **1  总体设计1.1 设计背景**  在当今高速发展的社会中，人们对自身所处的环境越来越关心，居家安全已成为当今小康之家优先考虑的问题。当上班家中无人，或者仅有老人孩子在家，或晚上在家熟睡，都必须确保家庭成员和财产的绝对安全。  中国安居工程需要大力发展安全文明小区建设，严峻的社会治安形势，需要建立并完善安全文明小区防盗报警网络系统体系。如何解决在当前每个家庭经济承受能力有限的情况下，建设满足防范功能及可靠性需求的安全文明小区防范防盗报警网络系统，是当前急需解决的课题。本方案是在考虑以上因素的基础上，所设计的可行实施方案。一、住宅小区防盗报警系统要求 当前，随着经济的发展，人民的生活日益改善，人们对家庭生命财产安全越来越重视，采取了许多措施来保护家庭的安全。以往的做法是安装防盗门、防盗网，但也存在有碍美观，不符合防火要求，不能有效地防止坏人的入侵。现在，全国各地都在如火如荼地开展建设安全文明小区的活动，而且很多地方都提出取消防盗网的口号，家庭电子防盗报警系统也就应运而生。因为大多数家庭都是双职工，白天家里通常没有人，发生报警后，必须要有专人来处理，因此，必须设立报警中心。而且因为国内住宅区大多数是密集型分布，一个住宅区往往有几百上千户，并且都有自身的保安队伍，因此当用户防盗报警系统报警时，除了在现场报警外，还需要向当地派出所或公安分局进行报警联网外，也需要向住宅小区的保安中心进行联网报警，以便警情得到迅速处理。另外，考虑到国内普遍收入水平较低，对于每一户家庭的防盗报警系统成本不可太高，但因为用户数量多，也不能采用质量差的产品，以免误报频繁造成不良影响。根据以上分析，住宅小区对防盗报警系统的要求如下：1.广泛性——即要求小区内每个家庭都能得到保护。2.实用性——即要求每个家庭的防范系统能在实际可能发生受侵害的情况下及时报警。并要求操作简便，环节少，易学。3.系统性——即要求每个家庭的防范系统在案情发生时，除能自身报警外，必须及时传到保卫部门，并同时上报当地公安报警中心。4.可靠性——即要求系统所设计的结构合理产品经久耐用、系统是可靠。5.投资可行性------即要求系统投资或造价能控制在小区家庭能承受的范围之内。  **1.2 设计思想**  在小区内的每个住户单元安装一台报警主机，住户可选择安装在住户门口、窗户处安装声检、紧急求助，烟雾/煤气探头、瓦斯探头，病毒检测器，防盗报警器等报警感知设备，报警主机通过总线与管理中心的电脑想连接，进行安防信息管理，本系统具有远程报警功能。如果发生盗贼闯入、抢劫、烟雾、燃汽泄露，声音过高等紧急事故，传感器就会立即获知并由报警系统即刻触发声光警报以有效恫吓企图行窃的盗贼；系统还会迅速向报警中心传送报警信息；报警中心接到警情后立即自动进行分辨处理，迅速识别判定警报类型、地点、用户，中心据此派出机动力量采取相应解救措施；系统具备24小时防破坏等并自我监视，一旦有任何被破坏的迹象也会即刻报警。总之，无论白天黑夜，你离家在外还是在家休息，电子保安时时刻刻保护的安全，这正是能为家人、家庭、财产所做的最有效的安全防盗保护措施。  **1.3 系统功能**  设计具有防盗并且兼有防火，防煤气，防盗毒等功能的小区防盗报警系统，无疑可以使家庭保安自动化。系统采用体积小巧，功能强大，价格便宜的单片微型计算机作为居民家庭第一监视端，与单片机连接各种用于家庭安保的传感器作为收集信号并送给单片机初步处理，单片机实时与远方主监控计算机通信，将从传感器接受来的信号实时传给主机，主机端就可以知道小区各个居民家庭的异常情况，从而进行实时处理。系统利用CAN总线传输信息，提高了可靠性，操作更方便，在工程实际中有广泛的应用，因而具有良好的应用前景和工程推广价值。  该系统运用了了单片机AT89C51， CAN总线的主要特性和功能，将主从式微机通信运用于今天的生活，体现了自动化的发展前景一片光明。对于应用区域：  每个家庭可实现：家中无人时，如上班了，可把家庭报警系统设置在外出布防状态，使所有的探测器都工作起来。当窃贼试图破门而入或从阳台闯入，被动红外探测器探测到动作，警号发声，并且保安中心立刻接收到警情，接着在数秒后公安局报警中心也会收到报警信号。家中有人时，如睡觉时，把系统设置在留守布防状态，当窃贼企图从大门闯入时，立刻动作发出警报；如果主人有紧急情况，如急病或受到挟持时，可按动键盘上的紧急按钮发出警报。  **1.4 系统设计框图**  图1.1 系统组成结构图  **1.5 系统实现方法**      (1) 利用各种传感器接收需要防备外界的异常信号，信号的检测与转换传输。  (2) 单片机接收传感器检测信号，进行初步处理，如记录发生情况的位置，何种报警等等，也可以单片机进行现场报警。  (3) 利用CAN总线实现单片机与主控计算机的串口通讯。  (4) 在主控计算机上接收单片机送来的报警信号，发出报警信息。  **2  系统可行性分析2.1 硬件模块**  报警器硬件由温度烟雾信号采集模块、煤气信号采集模块、防盗报警模块、光报警模块，键盘显示模块以及单片机与CAN总线模块组成。  **2.1.1单片机控制模块**  经过比较，作者决定使用常用、廉价的单片机AT89C51作为主控制器。单片机AT89C51是由ATMEL公司生产的51单片机。简单介绍如下：  AT89C51是一个低电压，高性能CMOS 8位单片机，片内含4k bytes的可反复擦写的只读程序存储器（PEROM）和128 bytes的随机存取数据存储器（RAM），器件采用ATMEL公司的高密度、非易失性存储技术生产，兼容标准MCS-51指令系统，片内置通用8位中央处理器和Flash存储单元，内置功能强大的微型计算机的AT89C51提供了高性价比的解决方案。     AT89C51是一个低功耗高性能单片机，40个引脚，32个外部双向输入/输出（I/O）端口，同时内含2个外中断口，2个16位可编程定时计数器,2个全双工串行通信口，         AT89C51可以按照常规方法进行编程，也可以在线编程。其将通用的微处理器和Flash存储器结合在一起，特别是可反复擦写的Flash存储器可有效地降低开发成本。  AT89C51具有PDIP、PQFP/TQFP及PLCC等三种封装形式，以适应不同产品的需求。  AT89C51的引脚图如下图所示：  图2.1AT89C51芯片的引脚图  VCC：供电电压。  GND：接地。  P0口：P0口为一个8位漏级开路双向I/O口，每脚可吸收8TTL门电流。当P1口的管脚第一次写1时，被定义为高阻输入。P0能够用于外部程序数据存储器，它可以被定义为数据/地址的第八位。在FIASH编程时，P0 口作为原码输入口，当FIASH进行校验时，P0输出原码，此时P0外部必须被拉高。  P1口：P1口是一个内部提供上拉电阻的8位双向I/O口，P1口缓冲器能接收输出4TTL门电流。P1口管脚写入1后，被内部上拉为高，可用作输入，P1口被外部下拉为低电平时，将输出电流，这是由于内部上拉的缘故。在FLASH编程和校验时，P1口作为第八位地址接收。  P2口：P2口为一个内部上拉电阻的8位双向I/O口，P2口缓冲器可接收，输出4个TTL门电流，当P2口被写“1”时，其管脚被内部上拉电阻拉高，且作为输入。并因此作为输入时，P2口的管脚被外部拉低，将输出电流。这是由于内部上拉的缘故。P2口当用于外部程序存储器或16位地址外部数据存储器进行存取时，P2口输出地址的高八位。在给出地址“1”时，它利用内部上拉优势，当对外部八位地址数据存储器进行读写时，P2口输出其特殊功能寄存器的内容。P2口在FLASH编程和校验时接收高八位地址信号和控制信号。  P3口：P3口管脚是8个带内部上拉电阻的双向I/O口，可接收输出4个TTL门电流。当P3口写入“1”后，它们被内部上拉为高电平，并用作输入。作为输入，由于外部下拉为低电平，P3口将输出电流（ILL）这是由于上拉的缘故。  P3口也可作为AT89C51的一些特殊功能口，如下所示：  P3.0 RXD（串行输入口）  P3.1 TXD（串行输出口）  P3.2 /INT0（外部中断0）  P3.3 /INT1（外部中断1）  P3.4 T0（记时器0外部输入）  P3.5 T1（记时器1外部输入）  P3.6 /WR（外部数据存储器写选通）  P3.7 /RD（外部数据存储器读选通）  P3口同时为闪烁编程和编程校验接收一些控制信号。  RST：复位输入。当振荡器复位器件时，要保持RST脚两个机器周期的高电平时间。  ALE/PROG：当访问外部存储器时，地址锁存允许的输出电平用于锁存地址的地位字节。在FLASH编程期间，此引脚用于输入编程脉冲。在平时，ALE端以不变的频率周期输出正脉冲信号，此频率为振荡器频率的1/6。因此它可用作对外部输出的脉冲或用于定时目的。然而要注意的是：每当用作外部数据存储器时，将跳过一个ALE脉冲。如想禁止ALE的输出可在SFR8EH地址上置0。此时， ALE只有在执行MOVX，MOVC指令是ALE才起作用。另外，该引脚被略微拉高。如果微处理器在外部执行状态ALE禁止，置位无效。  /PSEN：外部程序存储器的选通信号。在由外部程序存储器取指期间，每个机器周期两次/PSEN有效。但在访问外部数据存储器时，这两次有效的/PSEN信号将不出现。  /EA/VPP：当/EA保持低电平时，则在此期间外部程序存储器（0000H-FFFFH），不管是否有内部程序存储器。注意加密方式1时，/EA将内部锁定为RESET；当/EA端保持高电平时，此间内部程序存储器。在FLASH编程期间，此引脚也用于施加12V编程电源（VPP）。  XTAL1：反向振荡放大器的输入及内部时钟工作电路的输入。  XTAL2：来自反向振荡器的输出。  此单片机主要用于控制，包括响应中断、延时、判断、发送对方号码等等。  **2.1.2温度烟雾信号采集模块**  要准确的进行火灾报警，选择合适的温度传感器和烟雾传感器是准确报警的前提。综合考虑各种因素，作者选择集成数字温度传感器DS18B20和烟雾传感器NIS-09C用作采集系统的敏感元件。  **2.1.3煤气信号采集模块**  煤气泄露报警:一当室内煤气超过正常标准时,它将通过传感器向家庭控制器发出报警信号。对于密度大于空气的气体，感应器放在气体源的下方。对于密度小于空气的气体，感应器放在气体源的上方，当煤气泄露报警启动,就通知管理中心。综合考虑各种因素，作者选用气敏传感器TP-1.1A，用作煤气报警。  **2.1.4防盗报警模块**  防盗报警:在小区每一住户内安装防盗报警装置。当住户家中无人时,可把家庭内的防盗报警系统设置为布防状态,当窃贼闯入时,报警系统自动发出警报并向小区安保中­心报警。综合考虑各种因素，选用热电释红外传感器RE200B来进行防盗报警。  **2.1.5光报警模块**      由AT89C51的P1口的P1．4～P1．7分别控制4个发光二极管，予以光报警。P1．4～P1．7控制的灯依次为正常信号灯、煤气信号灯、火灾信号灯和防盗信号灯。当这些输出端输出低电平时，对应的信号灯便会发光报警。  图2.2 光报警电路图  **2.1.6数据采集模块**  A/D转换是本系统比较关键的一步，通过A/D转换，成功的把所要测量和控制的数据采集过来，是保证系统功能可以实现的第一步。为了设计好A/D转换模块，使它达到预想的A/D转换效果，必须选定合适的A/D转换器。  A/D转换电路采用了常用的8位8通道数模转换专用芯片ADC0809，ADC0809由8路模拟开头、地址锁存与译码器、8位A/D转换器和三态输出锁存缓冲器组成。  (1) 其主要性能指标为:  ①分辨率为8位。  ②最大不可调误差：0808为1／2LSB，0809为1LSB。  ③单电源+5v供电，基准电压由外部提供，典型值为+5v，此时允许输入模拟电压为0—5V。  ④具有锁存控制的8路模拟选通开关。  ⑤可锁存三态输出，输出电平与TTL电平兼容。  功耗15mW。  ⑥转换速度取决于芯片的时钟频率。时钟频率范围500KHz时，转换时间为128μs。  (2)ADC0809引脚功能  图2.3为ADC0809引脚图,说明如下；  图2.3 ADC0809芯片引脚  ①IN0—IN7——8路模拟信号输入端。  ②ADDA、ADDB、ADDC——3位地址码输入端。8路模拟信号转换选择由A、B、C决定。A为低位，C为高位。与低8位地址中A0—A2连接。由A0—A2地址000－111选择INO－IN7八路A／D通道。其中模拟开关与输入通道的关系见表3.1。  ③CLK——外部时钟输入端。时钟频率高，A／D转换速度快。允许范围为10—1280KHz，典型值为640KHz，此时A／D转换时间为100μs通常由MCS－51型单片机ALE端直接或分频后与0809CLK端相连接。当MCS－51型单片机无读写外RAM操作时，ALE信号固定为CPU时钟频率的1／6。若晶振为6MHz，则1／6为1MHz时．A／D转换时间为64μs。  表2.1路模拟开关与输入通道的关系表   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 通入通道 | IN0 | IN1 | IN2 | IN3 | IN4 | IN5 | IN6 | IN7 | | A | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | | B | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | | C | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |   ④D0—D7——数字量输出端。  ⑤ ——A／D转换结果输出允许控制端。当面端为高电平时，允许将A／D转换结果从Do—D7端输出。通常由MCS－ 51型单片机的端与0809片选端(例如P2.0)通过或非门与0809端相连接。当DPTR为FEFFH，且执行MOVX  A，＠DPTR指令  后，RD和P2.0均有效，或非后产生高电平，使0809端有效，0809将A／D转换结果送入数据总线Po口，CPU再读入A中。  ⑥ALE——地址锁存允许信号。8路模拟通道地址由A、B、C输入，在0809的ALE信号有效时将该8路地址锁存(注意0809ALE与803lALE的区别)。  ⑦START——启动A／D转换信号。当START端输入一个正脉冲时，立即启动0809进行A／D转换。START端与ALE端连在一起，由MCS－51型单片机WR与0809片选端(例如P2.0)通过或非门相连，当DPTR为EFF8H，执行“MOVX@DPTR，A”指令后，将启动0809模拟通道0的A／D转换。FEF8H—FEFFH分别为8路模拟输入通道的地址。执行MOVX写指令，井非真的将A中内容写进0809，而是产生WR信号和P2.o有效，从而使0809的START和ALE有效，且输出A／D通道0地址A0—A2。事实上也无法将A内容写进0809，0809中没有一个寄存器能容纳A中内容，0809的输入通道是IN0—IN7，输出通道是D0—D7，因此，执行“MOVX＠DPTR，A”指令与A中内容无关，但DPTR地址应指向当前A／D的通道地址。  ⑧EOC——A／D转换结束信号。当启动0809A／D转换后；EOC输出低电平；转换结束后，EOC输出高电平，表示可以读取A／D转换结果。该信号取反后，若与MCS－51型单片机引脚INT0或连接，可引发CPU中断，在中断服务程序中读A／D转换的数字信号。若MC5—51型单片机两个中断源已用完，则EOC也可与P1口或四口的一条端线相连，不采用中断方式，采用查询方式，查得EOC为高电平后，再读A／D转换值。  ⑨VREF+、VREF- ——正负基准电压输入端。基准电压的典型值为+5v，可与电源电压(+5v)相连，但电源电压往往有一定波动，将影响A／D精度。因此，精度要求较高时，可用高稳定度基准电源输入。当模拟信号电压较低时，基准电压也可取低于5v的数值。  ⑩Vcc——正电源电压(+5v)。GND——接地端。  ADC0809的A/D转换过程是在时钟信号的协调下进行的。ADC0809的时钟信号由CLOCK端送入,其最高频率为640MHz,在这个最高频率下ADC0809的A/D转换时间为100uS左右。  A/D转换结束后,A/D转换的结果(8位数字量)送到三态锁存输出缓冲器,此时A/D转换结果还没有现在DB0—DB7八条数字量输出线上,单片机不能获取之。单片机要想读到A/D转换结果,必须使ADC0809的允许输出控制端OE为高电平,打开三态输出锁存器,A/D转换结果出现在DB0—DB7上。  图2.4 ADC0809的读、写、启动以及A/D转换时序图  图2.4中EOC为转换结束输出信号。在A/D转换期间,EOC维持高电平,当A/D转换结束时,EOC变成高电平。ADC0809的START端收到下降沿后,并不立即进行A/D转换,EOC=1,而是延迟10uS后,才开始A/D转换,EOC变为低电平。单片机读取A/D转换结果的方法有三种(1)延迟法:单片机启动ADC0809后,延时130uS以上,可以读到正确的A/D转换结果。(2)查询法:EOC必须接到AT89C51的一条I/O线上。单片机启动ADC0809后,延迟10uS,检测EOC,若EOC=0则A/D转换没有结束,继续检测EOC直到EOC=1。当EOC=1时,A/D转换已经结束,单片机读取A/D转换结果。(3)中断法:EOC必须经过非门接到AT89C51的中断请求输入线INT0或INT1上,AT89C51的中断触发方式为下降沿触发。单片机启动A/D转换后可以做其它工作,当A/D转换结束时,EOC由0—1经过非门传到INT端,AT89C51收到中断请求信号,若AT89C51开中断,则进入中断服务程序,在中断服务程序中单片机读取A/D转换的结果。  数据采集的电路如图2.5所示。温度、烟雾、煤气、防盗传感器的输出分别接到ADC0809的IN4、IN5、IN6和IN7。ADC0809的通道选择地址A，B，C分别由AT89C51的P0．0～P0．2经地址锁存器74LS373输出提供。当P2.0=0时，与写信号WR共同选通ADC0809。图中ALE信号与ST信号连在一起，在WR信号的前沿写入地址信号，在其后沿启动转换。例如，输出地址7FF8H可选通通道IN4，实现对温度传感器输出的模拟量进行转换；输出地址7FF9H可选通通道IN1，实现对烟雾传感器输出的模拟量进行转换。图中ADC0809的转换结束状态信号EOC接到89C51的INT1引脚，当A/D转换完成后，EOC变为高电平，表示转换结束，产生中断。在中断服务程序中，将转换好的数据送到指定的存储单元。  图2.5 数据采集模块  **2.1.7键盘显示模块**  单片机系统所用的键盘有编码键盘和非编码键盘两种。  编码键盘是只要按下它的某一个键，就能产生这个键的代码，与此同时还产生一个脉冲信号，以通知CPU接收键码。编码键盘使用方便，也不用编写太复杂的程序。但硬件电路复杂，比非编码简易键盘成本要高。  非编码键盘也称简易键盘，它的按键是排列成行、列矩阵形式的。按键的作用只是简单地实现接点的接通或断开，因此必须有一套相应的程序与之配合，才能产生相应的键码，它基本上不需要附加什么硬件电路。但需要通过软件来解决按的识别、防抖动以及如何产生键码的问题。  为了节约成本，本系统选用简易键盘作为拨号输入。  单片机所用的显示有LED和LCD两种，从设计的难度和成本造价来说LCD都要高于LED。本作品要显示的是要拨的号码和来电显示对方机子的号码，为数字显示。因此，本作品选用了由LED组成的7段发光显示器，它有简单、经济、易于与单片机接口等优点。7段LED选用的是共阳极的。  **2.1.8CAN总线协议**  CAN（Controller Area Network）总线，又称控制器局域网，是Bosch公司在现代汽车技术中领先推出的一种多主机局域网。由于其卓越的性能，极高的可靠性，独特灵活的设计和低廉的价格，现已广泛应用于工业现场控制、智能大厦、小区安防、交通工具、医疗仪器、环境监控等众多领域。  CAN的通信协议主要由CAN控制器完成。CAN控制器主要由实现CAN总线协议的部分和实现与微处理器接口部分的电路组成。对于不同型号的CAN总线通信控制器，实现CAN协议部分电路的结构和功能大多相同，而与微处理器接口部分的结构和方式存在一些差异。  CAN控制器选用SJA1000作为控制器。SJA1000是高集成度CAN控制器。具有多主结构、总线访问优先权、成组与广播报文功能及硬件滤波功能。输入时钟频率为16MHZ时钟,输出可编程控制。由以下几部分构成：接口管理逻辑、发送缓存器、接收缓存器、位流处理器、位定时逻辑、收发逻辑、错误管理逻辑、控制器接口逻辑等。  SJA1000在电路中是一个总线接口芯片，通过它实现上位机与现场微处理器之间的数据通信。该电路的主要功能是通过CAN总线接收来自上位机的数据进行分析组态然后下传给下位机的控制电路实现控制功能，当CAN总线接口接收到下位机的上传数据，SJA1000就产生一个中断，引发微处理器产生中断，通过中断处理程序接收每一帧信息并通过CAN总线上传给上位机进行分析。AT89C51是CAN总线接口电路的核心，其承担CAN控制器的初始化、CAN的收发控制等任务。  CAN总线收发器选用PCA82C250作为总线收发器。 PCA82C250是CAN 协议控制器和物理总线之间的接口。82C250 可以为总线提供不同的发送性能，为CAN 控制器提供不同的接收性能。而且它与“ISO 11898”标准完全兼容。PCA82C250的目的是为了增大通信距离，提高系统的瞬间抗干扰能力，保护总线，降低射频干扰（RFI）实现热防护等。  接口主要元件电路原理图见图2.6，在进行电路设计时应注意以下几点，否则达不到预期的效果。  （1）总线两端必须接两个终端匹配电阻RT，忽略掉它们，会使数据通信的抗干扰性及可靠性大大降低。  （2）PCA82C250为CAN控制器和物理总线之间的接口，它可以提供向总线的  差动发送能力和CAN控制器的差动接收能力，TXD和RXD引脚分别发送经过驱动后的发送和接收信号。其引脚8（RS）可以选择2种不同的工作方式：把该引脚直接与地相连，系统将处于高速工作方式，在这种方式下，为避免射频干扰，建议使用屏蔽电缆作总线；而在波特率较低，总线较短时，一般采用斜率控制方式，上升及下降的斜率取决于RS的阻值，实践表明15-200k为RS较理想的取值范围，在这种方式下，可以使用双绞线作总线（本系统采用该工作方式）。  （3）SJA1000的TX1脚悬空，RX1引脚的电位必须维持在约0.5VCC上，否则，  将不能形成CAN协议要求的电平逻辑。因本系统传输距离近，环境干扰小，可以不用电流隔离，这样可以直接把82C250的VREF端（约为0.5VCC）与SJA1000的RX1相连，从而简化了电路。  （4）设计时将SJA1000的CLOCKOUT的时钟信号接至AT89C51的时钟电路输  入端，作为AT89C51的外部时钟输入，解决了时钟同步问题；SJA1000中断输出信号/INT接至AT89C51的/INT0端，通过中断方式与AT89C51通信。      AT89C51                 SJA1000     图2.6 接口主要元器件电路图  **2.1.9传感器模块**  上述所说的各种信号采集模块都是经过传感器来进行信号采集，再送到ADC0809。  传感器是一个从系统接收功率，通常以另一种形式将功率送到第二个系统中的器件。按照这一定义，可以说传感器是一种吸收(提取)、转换和传送能量的器件。严格地说，应该区别检测和变换这两个概念。前者只是将被测量转移到系统中去，而后者从字面上说是将被测量的种类加以改变。有时，对被测量只进行一次变换还不够，必须进行第二次、第三次变换，一直到所希望的能量形式为止。传感器转换能量的理论基础都是利用物理学、化学等各种现象和效应来进行能量形式的变换。随着微电子和计算机技术的发展，传感器输出信号的形式应尽可能是电量。  传感器的组成按其定义一般是由敏感元件、变换元件和测量电路三部分组成。除自源型传感器外，还需外加辅助电源，用框图表示，如图2.7所示。由图可知，传感器由以下几个部分组成：  图2.7 传感器的组成框图  （1）敏感元件(预变换器)  直接感受被测量(一般为非电量)并将其转换为与被测量有确定关系的易变成电量(包括电量)的其它量的元件。  （2）转换元件(变换器)  它能将其它物理量直接转换为有确定关系的电量的元件。  （3）测量电路(变换电路)  把转换元件输出的电信号变为便于处理、显示、记录、控制的可用电信号的电路。测量电路的类型视转换元件的不同而定．经常采用的有电桥电路和其它特殊电路，加高阻抗输入电路、脉冲电路、振荡电路等。  （4）辅助电源  供给转换能量。  **2.2 软件模块**  通过上面分析，决定选用C51系列单片机作为控制器，那么它的软件编写就要按照单片机语言：C语言或汇编语言来编写，这里选用执行效率高的汇编语言编写。 开发工具为keil c51，它的功能强大，支持c语言和汇编语言的编写与调试，其软件模拟I/O口也会给软件调试带来极大的方便。  **2.2.1温度烟雾信号采集模块**  进行火灾报警，需要温度和烟雾两个传感器，当温度达到一定的度数和烟雾的浓度达到一定的百分比时，就进行火灾报警。在本设计中，作者选用了数字温度传感器AS18B20和烟雾传感器NIS-09C来进行报警。  把数字温度传感器AS18B20和烟雾传感器NIS-09C分别接到ADC0809IN4口和IN5口。当进行火灾报警时，ADC0809就将采集到的数据信号存储在数据区，进行A/D转换，并将指针指向下一个通道。  **2.2.2煤气信号采集模块**  进行煤气报警，作者选用了常用的家庭煤气报警传感器TP-1.1A。将煤气传感器TP-1.1A接到ADC0809的IN6口。当进行煤气报警时，ADC0809就将采集到的数据信号存储在数据区，进行A/D转换，并将指针指向下一个通道。  **2.2.3防盗报警信号采集模块**  人体有恒定的体温，一般在37摄氏度，所以会发出特定波长的红外线。热电释红外传感器探头就是靠探测人体发射的红外线而进行工作。人体发出的红外线，通过菲涅尔滤光片增强后，聚集到红外感应源上。热电式元件在接收到人体红外辐射温度发生变化时将会失去电荷平衡，向外释放电荷，后续电路经检测处理后产生报警信号。进行防盗报警时，作者选用了价格比较合适的热电释红外传感器RE200B。当人体靠近热电释红外传感器RE200B到一定距离时，热电释红外传感器就会报警，ADC0809就会将采集的信号存储在数据区，进行A/D转换。  **2.2.4光报警**  把四个发光二极管分别接在AT89C51的P1．4～P1．7口。P1．4～P1．7口控制的灯依次为正常信号灯、煤气信号灯、火灾信号灯和防盗信号灯。当这些输出端输出低电平时，对应的信号灯便会发光报警。  **2.2.5信号采集模块**  数据采集部分的程序设计包括：驱动ADC0809的IN4，IN5，IN6和IN7进行A/D转换，分别由子程序ADC1(温度转换)，ADC2(烟雾浓度转换)，ADC3（煤气浓度转换）和ADC4（防盗转换）完成；单片机接收转换好的数据，存入指定内存单元，由INT1中断服务程序完成。每次驱动A/D转换后等待外部中断1，中断到来说明A/D转换已经完成，通过中断服务程序读取转换得到的数据。  数据采集模块的工作原理：各类传感器采集到数据后模拟量传送到ADC0809，0809将转换成的数字量传给89C51，最后单片机将采集到的数据送到SJA1000通过CAN总线收发器82C250传上总线，完成数据采集工作，并由CAN 总线通过RS232传给上位机。  **2.2.6键盘显示模块**      本单元需要两个并行口：一个作为行信号输出，它与显示器的位选码共享一个口，这里作者选用P2口的P2.2和P2.3；另一个作为列信号的读入口，这里选用P0口的前四位P2.4至P2.7。  **2.2.7CAN总线通讯模块**  CAN总线测控系统的通信软件分为3部分：CAN初始化、数据发送和数据接收。  CAN初始化，其主要是设置CAN的通信参数。需要初始化的寄存器有：模式寄存器（Peli CAN模式）、时分寄存器、接收代码寄存器、屏蔽寄存器、总线定时寄存器、输出控制寄存器等。需要注意的是，这些寄存器仅能在复位期间可写访向，因此,在对这些寄存器初始化前，必须确保系统进入了复位状态，并且系统中各CAN控制器的总线定时寄存器的初始化字必须相同。  数据发送，现场的各传感器把环境多参数的检测信号（数字量、模拟量、开关量）进行转换处理后，发向CAN控制器的发送缓冲区，然后启动CAN控制器的发送命令，此时CAN控制器将自动向总线发送数据，不再需传感器的微控制器进行干预。  数据接收，整个微机测控系统中的CAN控制器检测到总线上有数据时会自动接收总线上的数据，存入其接收缓冲区，并向AT89C51微控制器发送接收中断，启动中断接收服务程序，AT89C51通过执行中断接收服务程序，从CAN控制器的接收缓冲区读取数据，并对其进行进一步处理工作。  **3  硬件单元电路设计3.1 ADC0809和四个报警传感器的接口电路设计**  ADC0809和四个报警传感器的接口电路设计图如图3.1所示。  图3.1 ADC0809和传感器的接口电路图  上图所示的四个插针分别代表温度传感器，烟雾传感器，煤气传感器和防盗传感器。因为传感器进行报警时，要将传感器放于不同的位子，所以在设计电路图时，作者就选用插针，这样，用软线把插针与传感器连接起来，显得更加方便，可以将传感器放于需要的位子。  元器件的选取：  ADC0809芯片一块  插针四个（三排插针两个，两排插折两个）  47K的电阻一个，51欧姆的电阻一个，50K的电阻两个  **3.2 AT89C51与ADC0809接口电路**  AT89C51与ADC0809接口电路如3.2所示。  图3.2 AT89C51与ADC0809接口电路图  ADC0809的通道选择地址A，B，C分别由AT89C51的P0．0～P0．2经地址锁存器74LS373输出提供。当P2.7=0时，与写信号WR共同选通ADC0809。图中ALE信号与ST信号连在一起，在WR信 号的前沿写入地址信号，在其后沿启动转换。ADC0809的转换结束状态信号EOC接到AT89C51的INT1引脚，当A/D转换完成后，EOC变为高电平，表示转换结束，产生中断。在中断服务程序中，将转换好的数据送到指定的存储单元。  AT89C51的频率跟ADC0809的频率不一致，所以选用74LS74进行二分频。  元器件的选取：  AT89C51单片机一块  ADC0809芯片一块  74LS74芯片一块  74LS02芯片一块  74LS04芯片两块  **3.3 AT89C51与光报警接口电路**  AT89C51与光报警接口电路如图3.3所示  图3.3 AT89C51与光报警接口电路图  把四个发光二极管分别接在AT89C51的P1．4～P1．7口。P1．4～P1．7口控制的灯依次为正常信号灯、煤气信号灯、火灾信号灯和防盗信号灯。当进行一种报警时，所对应的发光二极管就会亮。例如，当进行火灾报警时，P1.6口的发光二极管就会亮。  元器件的选取：  四个340欧姆的电阻  四个发光二极管  **3.4 键盘显示接口电路**  键盘显示接口电路如图3.4所示。  图3.4 键盘显示接口电路图  具体扫描法：（1）将全部行线Y0-Y3（行）置低电平，然后检测列的状态。只要有一列的电平为低，则表示按键中有键被按下，而且闭合的键位于低电平与4根行线相交叉的4个按键之中。若所有列线均为高电平，则键盘中没有键按下。（2）判断闭合键所在的位置。在确定有按键按下以后，就可以进入确定具体闭合键的过程。其方法是：依次将行线置低电平，即在置某根行线为低电平时，其线为高电平。在确定某根行线位置为低电平后，再将行检测各列的电平状态。若为低，则该列线与置为低电平的行线交叉处的按键就是闭合的按键。  元器件的选取：  按键选用我们熟悉的四脚矩形按键8个  9012PNP放大三极管4个  四位共阳LED数码管一块  4.7k的限流电阻4个  **3.5 AT89C51与CAN总线接口电路**  AT89C51与CAN总线接口电路图如图3.5所示。  图3.5 AT89C51与CAN总线接口电路图  在系统中，SJA1000的片选信号一般由地址总线经译码获得，并由此决定出CAN控制器各寄存器的地址。在本设计中，作者用单片机AT89C51的P2.7为片选信号。所以，SJA1000的地址为：7F00-7F32H。  SJA1000的TX1脚悬空，RX1引脚的电位必须维持在约0.5Vcc上，否则，将不能形成CAN协议所要求的电平逻辑。如果系统传输距离近，环境干扰小，可以不用电流隔离，这样可直接把82C250的VREF端（约为0.5 Vcc）与RX1脚相连，从而简化了电路。  当上电复位时，AT89C52的上电复位，需要从低到高的电平变化来激活，而SJA1000的17脚RST被激活，需要出现一个由高电平到低电平的跳变，因此，这必须加一个反相器。  82C50第8脚与地之间的电阻Rs称为斜率电阻，它的取值决定了系统处于高速工作方式还是斜率控制方式。而在波特率较低、总线较短时，一般采用斜率控制方式，上升及下降的斜率取决于民的阻值，实验数据表明15~200kΩ为Rs较理想的取值范围，在该方式下，可以使用平行线或双绞线作总线。在本设计中，Rs选用150K。  元器件的选取：  SJA1000芯片一块  PCA82C250芯片一块  74LS04芯片一块  16MHZ晶振一个  22PF的二极管两个  60欧姆的电阻一个  150K的电阻一个  **4  软件设计**  本系统的软件设计主要是采集四个报警传感器的报警信号，再通过发光二极管显示，每一个发光二极管亮就表示有一种报警传感器报警。用户可以根据自己家庭的具体情况来设置报警的最高限度，用键盘显示来实现。单片机还要把采集的信号通过CAN总线传输到上位机，在上位机上显示出来。  **4.1 CAN总线程序**  CAN总线测控系统的通信软件分为3部分：CAN初始化、数据发送和数据接收。  **4.1.1CAN初始化**  CAN初始化，其主要是设置CAN的通信参数。需要初始化的寄存器有：模式寄存器（Peli CAN模式）、时分寄存器、接收代码寄存器、屏蔽寄存器、总线定时寄存器、输出控制寄存器。  CAN初始化程序代码为：   1. #include <reg52.h> 2. #include <stdio.h> 3. #include <string.h> 4. #include <absacc.h> 5. #define byte unsigned char 6. #define word unsigned int 7. #define time0  65536-2000 8. #define BAUD  -13 9. #define MAIN    1  ///定义主机，从机地址 10. #define ADDR    2 11. #define READ   'R' 12. #define WRIT   'W' 13. #define CAN\_PORT P0 14. #define CAN\_CR   0    //控制 15. #define CAN\_CMR  1    //命令 16. #define CAN\_SR   2    //状态 17. #define CAN\_IR   3    //中断 18. #define CAN\_ACR  4    //验收 19. #define CAN\_AMR  5    //屏蔽 20. #define CAN\_BTR0 6    //时序0 21. #define CAN\_BTR1 7    //时序1 22. #define CAN\_OCR  8    //输出 23. #define CAN\_TXB  10   //发送缓冲 24. #define CAN\_RXB  20   //接收缓冲 25. #define CAN\_CDR  31   //分频 26. sbit CAN\_WR  = P2^5; 27. sbit CAN\_RD  = P2^6; 28. sbit CAN\_ALE = P2^7; 29. sbit CAN\_RST = P1^0; 30. sbit da\_in   = P3^7; 31. sbit da\_clk  = P3^6; 32. sbit da\_cs   = P3^5; 33. sbit KEY1    = P2^0; 34. sbit KEY2    = P2^1; 35. sbit LED1    = P1^3; 36. sbit LED2    = P1^4; 37. sbit LED7    = P1^7; 38. sbit beer    = P1^1; 39. bit CANBUS; 40. byte bdata Acc ; 41. sbit Acc7=Acc^7; 42. byte idata TXB[10]={1,2,3,4,5,6,7,8,9,10}; //第一个为报文标志码 44. byte idata RXB[10]; 45. void CAN\_INI()  /\*初始化部分\*/ 46. { 47. CANW(CAN\_CR,0x01); 48. CANW(CAN\_ACR,MAIN); //验收滤波器需与报文标志码(从机的发送的第一个字节)一致 49. CANW(CAN\_AMR,0x00); 50. CANW(CAN\_BTR0,0x42);   //125k-BPS 51. CANW(CAN\_BTR1,0x1c); 52. CANW(CAN\_OCR,0x1A);   //0xaa,0xd2,0xfa 53. CANW(CAN\_CR,0x02);    //\*\*中断使能 54. } 55. 4.1.2CAN数据发送 56. 数据发送，现场的各传感器把环境多参数的检测信号（数字量、模拟量、开关量）进行转换处理后，发向CAN控制器的发送缓冲区，然后启动CAN控制器的发送命令，此时CAN控制器将自动向总线发送数据，不再需传感器的微控制器进行干预。 57. CAN数据发送的程序代码为： 58. void CANW(byte adr,byte dat) //写CAN数 59. { 60. CAN\_ALE=1; 61. CAN\_PORT=adr; 62. CAN\_ALE=0; 63. CAN\_WR=0; 64. CAN\_PORT=dat; 65. CAN\_WR=1; 66. } 67. byte CANR(byte adr)  //读CAN数 68. { 69. byte dat; 70. CAN\_ALE=1; 71. CAN\_PORT=adr; 72. CAN\_ALE=0; 73. CAN\_PORT=0xff; 74. CAN\_RD=0; 75. dat=CAN\_PORT; 76. CAN\_RD=1; 77. return dat; 78. } 79. void CAN\_SEND() /\*发送数据部分\*/ 80. { 81. byte i,p; 82. p=CANR(CAN\_SR); 83. if (p&0x04) 84. { 85. p=CAN\_TXB;        /\*p指向发送缓存首址\*/ 86. for(i=0;i<10;i++) CANW(p++,TXB[i]); 87. CANW(CAN\_CMR,0x01);    /\*请求发送\*/ 88. } 89. } 90. 4.1.3CAN接收数据 91. 数据接收，整个微机测控系统中的CAN控制器检测到总线上有数据时会自动接收总线上的数据，存入其接收缓冲区，并向AT89C51微控制器发送接收中断，启动中断接收服务程序，AT89C51通过执行中断接收服务程序，从CAN控制器的接收缓冲区读取数据，并对其进行进一步处理工作。 92. CAN接收数据程序为： 93. void CAN\_INT() interrupt 0 using 1/\*接收中断\*/ 94. { 95. byte i,p; 96. p=CANR(CAN\_IR); 97. if(p&0x01) 98. { 99. p=CAN\_RXB; 100. for(i=0;i<10;i++) 101. { 102. RXB[i]=CANR(p++); 103. } 104. CANW(CAN\_CMR,0x04);     /\*释放接收缓存\*/ 105. CANBUS=1;// LED2=1; 106. } 107. } 108. void CAN\_SCAN() 109. { 110. TXB[0]=ADDR; 111. TXB[1]=8; 112. TXB[2]=ADDR; 113. TXB[3]=WRIT; 114. TXB[4]=2; 115. if(KEY1==0) TXB[5]=0; else TXB[5]=1; 116. if(KEY2==0) TXB[6]=0; else TXB[6]=1; 117. CAN\_SEND(); 118. //  while(!CANBUS); //等待接收完标志 119. CANBUS=0; 120. TXB[0]=ADDR;  //总线地址 121. TXB[1]=8;     //发送数据字节个数! 122. TXB[2]=ADDR; 123. TXB[3]=READ; 124. TXB[4]=2; 125. CAN\_SEND(); 126. //  while(!CANBUS); 127. CANBUS=0; 128. if(RXB[5]==0) LED1=0; else LED1=1; 129. if(RXB[6]==0) LED2=0; else LED2=1; 130. } 131. 4.2 温度传感器18B20 132. DS18B20的程序见附录三。 133. 4.3 ADC0809转换 134. ADC0809的转换程序为： 135. #include <reg51.h> 136. #define uchar unsigned char 137. #define uint unsigned int 138. //#define t0 65536-50000 139. //#define t1 65536-30000 140. sbit key3 =P2^3; 141. sbit key2 =P2^2; 142. sbit key1 =P2^1; 143. sbit key0 =P2^0; 144. sbit P30= P3^0; 145. sbit P31= P3^1; 146. sbit P32= P3^2; 147. sbit P33= P3^3; 148. sbit P34= P3^4; 149. //uchar scan=0,dis=0,keyx=1,key; 150. uchar now; 151. uchar code              scantab[4]              ={0x80,0x40,0x20,0x10}; 152. uchar code              disptab[4]              ={0x01,0x80,0x40,0x20}; 153. uchar                            n[4]                            ={0x00,0x00,0x00,0x00}; 154. uchar code              trantab[18]              = 155. {              0x28,0xEE,0x32,0xA2,0xE4,0xA1,0x21, 156. 0xEA,0x20,0xA0,0x60,0x25,0x39,0x26, 157. 0x31,0x71,0x29,0xff 158. }; 159. int              dispnum; 160. sbit start=P3^1; 161. sbit oe=P3^2; 162. #define t0 65536-1000 163. bit flag; 164. unsigned long uin,udis; 165. /\* 166. int temp; 167. int  get\_p1()                            {P1=0xff;return P1;} 168. int  get\_p3()                            {P3=0xff;return P3;} 169. void put\_p1(int h1)              {P1=h1;} 170. void put\_p3(int h2)              {P3=h2;} 171. \*/ 172. void delay(uint t) 173. {              uint i; 174. for(i=0;i<t;i++); 175. } 176. /\* 177. void dealkey() 178. {              if(key>0) 179. {              switch(key) 180. {              case              1: P3^0=0;                                          break; 181. case              5: P3^0=1;                                          break; 182. case              2: P3^1=0;                                          break; 183. case              6: P3^1=1;                                          break; 184. case              3: clock=0;                                          break; 185. case              7: clock=1;                                          break; 186. case              4: P3^3=0xff;dispnum=P3^3;              break; 187. case              8: P3^3=temp;                                                        break; 188. case              9: P31=1;                                                                      break; 189. case   13: P31=1;                                                                      break; 190. case   10: P33=1;dispnum=P33;                                                        break; 191. case   14: temp=get\_p1(); dispnum=temp;                            break; 192. case   11: temp=get\_p1(); dispnum=5\*P1/255; 193. break; 194. case   15:              break; 195. case   12: P32=0;              break; 196. case   16: P32=!P32;              break; 197. } 198. if(dispnum<0)              dispnum=0; 199. if(dispnum>9999)              dispnum=9999; 200. distran(dispnum); 201. } 202. key=0; 203. } 204. \*/ 205. void distran(int  dispnum) 206. {              n[0]              =trantab[dispnum                            %10]; 207. n[1]              =trantab[(dispnum/10)              %10]; 208. n[2]              =trantab[(dispnum/100)              %10]; 209. n[3]              =trantab[(dispnum/1000)              %10]; 210. } 211. void display() 212. {              P2=~disptab[now]; 213. P0=n[now]; 214. }/\* 215. void scankey() 216. {              P0=trantab[17]; 217. now=(now+1)%4; 218. P2=~scantab[now]; 219. key0=key1=key2=key3=1; 220. if(!key0)keyx=now+1; 221. if(!key1)keyx=now+5; 222. if(!key2)keyx=now+9; 223. if(!key3)keyx=now+13; 224. } 225. \*/ 226. void t0\_sever() interrupt 1 227. {              TH0=t0>>8;TL0=t0; 228. //              scankey(); 229. //              dealkey(); 230. display(); 231. flag=0; 232. } 233. /\* 234. void t1\_sever() interrupt 2 235. {              TH1=t1>>8;TL1=t1; 236. //scankey(); 237. //dealkey(); 238. display(); 239. } 240. \*/ 241. void deal0809() 242. {              unsigned long u; 243. u=(unsigned long)uin; 244. udis=((u\*5\*1000)/256); 245. distran(udis); 246. } 247. void get0809() 248. {              //delay(80); 249. start=1;//f=(double)((fxx\*200000000)/(fcc\*16)); 250. delay(8); 251. start=0; 252. delay(50); 253. oe=1; 254. P1=1; 255. uin=P1; 256. oe=0; 257. } 258. void main() 259. { 260. start=0; 261. oe=0; 262. flag=0; 263. EA=1; 264. ET0=1;              TH0=t0>>8;              TL0=t0;              TR0=1; 265. now=0; 266. while(1) 267. { 268. if(flag)display(); 269. else 270. {              get0809(); 271. deal0809(); 272. //display(); 273. } 274. } 275. } 276. /\* 277. if(keyx>0) 278. {              key=keyx; 279. while(keyx!=0){keyx=0;time(10000);} 280. dealkey(); 281. } 282. time(10000); 283. \*/ 284. 4.4 数字显示 285. 本模块功能主要为小区家庭用户根据自家的情况显示不同的家庭门牌号码。具体程序代码为： 286. #include <reg51.h> 287. #define uint unsigned int 288. #define uchar unsigned char 289. #define tim 65536-3000 290. sbit dis0 =P2^7; 291. sbit dis1 =P2^6; 292. sbit dis2 =P2^5; 293. sbit dis3 =P2^4; 294. sbit key1 =P2^3; 295. sbit key0 =P2^2; 296. int i; 297. uchar disp[4],dis; 298. uchar key,keyx,ts; 299. uint para[8],ptr; 300. uchar code numtab[4]={0x80,0x40,0x20,0x10}; 301. uchar code distab[16]={0x28,0xee,0x32,0xa2,0xe4,0xa1,0x21,0xea, 302. 0x20,0xa0,0x60,0x25,0x39,0x26,0x31,0x71}; 303. int  code param[8]={300,255,300,255,255,255,255,8}; 304. void time(uint t) 305. { 306. uint j; 307. for(j=0;j<t;j++); 308. } 309. void distran() 310. { 311. disp[0]=distab[para[ptr]%10]; 312. disp[1]=distab[(para[ptr]/10)%10]; 313. disp[2]=distab[(para[ptr]/100)%10]; 314. disp[3]=distab[(para[ptr]/1000)%10]; 315. } 316. void display() 317. { 318. dis=(dis+1)%4; 319. P2=0xff; 320. P0=disp[dis]; 321. P2=~numtab[dis]; 322. key0=1; key1=1; 323. if(!key0) keyx=dis+1;//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*键盘扫描 324. if(!key1) keyx=dis+5; 325. } 326. void keyproc() 327. { 328. if(key>0) 329. { 330. switch(key) 331. { 332. case 1: para[ptr]--;      break; 333. case 5: para[ptr]++;      break;  //\*\*\*\*\*\*\*\*\*\*\*\*个位 334. case 2: para[ptr]-=10;     break; 335. case 6: para[ptr]+=10;     break;  //\*\*\*\*\*\*\*\*\*\*\*\*\*十 336. case 3: para[ptr]-=100;    break; 337. case 7: para[ptr]+=100;    break;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*百 338. case 4: ptr--;             break; 339. case 8: ptr++;             break;      //\*\*\*\*\*\*\*\*\*千位 340. } 341. if(ptr>7) ptr=0; 342. if(ptr<0) ptr=7; 343. if(para[ptr]<0)   para[ptr]=0; 344. if(para[ptr]>param[ptr])   para[ptr]=param[ptr]; 345. distran(); 346. key=0; 347. } 348. } 349. void timer0() interrupt 1 350. { 351. TH0=tim>>8; TL0=tim; display(); 352. } 353. void main() 354. { 355. TMOD=0x11;IE=0x82; 356. dis=0; 357. ptr=0; 358. distran(); 359. TH0=tim>>8;  TL0=tim; TR0=1; 360. for(i=0;i<7;i++)   para[i]=i; 361. for(;;) 362. { 363. if(keyx>0)                 //有键按下 364. { 365. key=keyx; 366. while(keyx!=0) {keyx=0; time(1000);} 367. if(keyx!=0) 368. { 369. /// if(!key0) {keyx=dis+1;}//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*键盘扫描 370. ////  if(!key1) {keyx=dis+5;} 371. } 372. keyproc(); 373. } 374. time(1000); 375. } 376. }   复制代码  **5  系统可靠性设计**  小区安防系统的报警要求准确无误，必须是可靠的，因此可靠性是一个很重要的指标，因此，系统的硬件电路、软件系统的设计均应围绕这个中心进行。  **5.1 系统抗干扰性设计**  小区安防系统主要进行小区内各个住户的各种突发事件的报警。信号会受到各种干扰，使系统不能正常工作。  在抗干扰方面，有硬件措施和软件措施，也有软硬件结合的措施。硬件抗干扰有效率高的有优点，但是增加系统的投资和设备的体积；软件抗干扰有投资低的优点，但会降低系统的工作效率。  **5.1.1硬件抗干扰措施**  干扰的渠道有：供电系统的干扰，过程通道的干扰，空间的干扰等。在印刷电路板设计时，考虑到电路板过大时，印刷线条长，抗噪声能力下降；过小时，散热不好，容易受邻近线干扰。所有本系统设计电路板时，把相关的器件尽量放靠近些，容易产生噪声的器件、电流电路、大电流电路尽量远离单片机。同时整个系统用一个电源，并保证其良好的共地。  **5.1.2软件抗干扰措施**  软件抗干扰属于单片机系统自身防御行为，其前提条件是：系统中抗干扰软件不会因干扰而损坏。  （1）程序结构化，功能模块化  编程时不能让程序任意跳转，否则整个程序会纠缠在一起，便于程序维护，而且对软件抗干扰方案的实施有着十分重要的意义。在系统中把整个程序分解为若干个功能模块，每个功能模块是一个正规程序。  （2）状态信号的抗干扰输入方法  干扰信号多呈毛刺状，作用时间短，在采集某一状态信号时，可以多次重复采集，只有连续两次采集结果一致时才视为有效；在各次采集状态信号之间增加一段延时，能对抗较宽的干扰。  （3）CPU抗干扰技术  CPU属于高速数字器件，易受干扰的有运算器、控制器以及控制寄存器，当外界干扰串入时，很可能通过三总线改写CPU寄存器的内容，导致初始化错误。寻址失败乃至系统瘫痪；当程序受到干扰后，和可能打乱CPU的正常执行程序。如果拦截失去控制的程序流程，如果使系统的损失减少，尽可能恢复系统的正常状态是本系统需要考虑的一项重要内容。  系统采取的措施是在各段程序中，对单片机及片外扩展器件的各种功能、断口、方式、状态等采集永久性或临时性的设置。系统不仅要保证上电后系统要初始化，而且在程序中每次使用某种功能，都要再一次对相应的控制寄存器设定动作模式。  **5.2 提高元器件的可靠性**  提高单片机应用系统中所有元件的质量，以提高系统内在的可靠性，这也是关键性的措施。采取的措施有：  （1）选用质量好的接插件，并设计好工艺结构，如选用带屏蔽的接插件；  （2）选用合格的电子元件，并进行严格的测试、筛选；  （3）设计时，技术参数留有一定的余量；  （4）提高印刷板的组装质量。  **5.3 容错技术**  在提高系统可靠性的同时，采取一些容错技术，当系统工作出错时，系统可自动纠正错误。在系统中主要应用了信息冗余技术。  （1）奇偶校验。系统通信时采用偶校验，收方根据校验结果，决定是否要求对方重发。     （2）循环码校验。在发送数据时按一定的规则产生循环冗余码，并附加在数据后一起发送；接收方按同样的规则根据接收的数据产生循环冗余码，并和接收到的循环冗余码进行必比较，校验数据传送是否正确。  **6  系统调试6.1 整体调试分析**  整个调试所使用的测试仪器仪表和工具：   1. 微机一台； 2. 示波器； 3. 万用表； 4. 直流稳压电源； 5. 单片机开发系统一套；   本系统的调试主要分为硬件调试、软件调试和联机调试三大部分。  经过初步的分析设计后，在制作硬件电路的同时，软件调试也在穿插进行。这样有利于问题的分析和解决，不会造成问题的积累，而且不会因为一个小问题而需要检查整体电路，从而可以节约大量的调试时间。软件编程中，作者首先完成的是单元功能模块的调试，然后进行系统调试，整体上与硬件调试的方法差不多。联机调试是最重要的一部分，同时也是本装置成功的关键，有许多问题都比较棘手。  **6.2 硬件单元电路调试6.2.1最小系统板调试**  首先检测所制作的电路板是否有虚焊，短路等现象。然后接上+5V电源，检测各个芯片的电源是否为+5V，如果是+5V的话，说明供电正确。接上电压，看单片机是否发热，如不发热说明电源连接正确。测单片机30脚（ALE）是否有脉冲波形输出，如果有说明单片机正常工作。  **6.2.2A/D转换电路调试**  先对A/D进行编程调试，通过后才能继续完成其他模块的调试，这一步很关键，也是最难调试的一步。  给AT89C51烧入A/D转换程序先调试A/D转换电路。A/D转换电路调试步骤如下：  1）、给四路模拟电压输入信号和一路数字电压输入信号分别输入相应的直流电压，用万用表检测电压是否采集过来，也就是输入端电压与电源所给电压是否一致。  2）检测ADC0809的启动信号和输出使能信号输出是否正确。  3）测得输入端电压与电源所给电压一致后，再检测ADC0809的通道选择是否正确：如：给IN5输入0~5V的电压，ADC0809的通道选择端，如果addrc=5V,addrb=0V,addra=5V, 也即：addrc=1,addrb=0,addra=1;表示该通道选择正确，否则不正确。如果不正确，检查程序是否正确，如果程序没有问题，再检查电路是否接对了。  4）测得通道选择正确后，再测量数字量输出，也即ADC0809的D0~D7输出端的数字量是否与输入模拟量相对应。改变模拟电压量输入，看数字信号变化是否和模拟量变化对应，如果输出数字量与输入模拟量相对应说明A/D转换正确。  **6.2.3光报警电路调试**  首先先检查发光二极管的正负极有没有接错，发光二极管有缺口的是负极。当出现低电平时，发光二极管亮。  当接收到传感器所检测的信号有异样时，所相应的发光二极管是否亮。例如，当温度传感器所检测到的温度高于所设定的最高温度时，所对应的发光二极管就发亮。  **6.2.4键盘显示电路调试**  在调试前，值得提到的是三极管的几个极的判断和数码管的a至g所对应的管脚的测试，这是十分重要的，这直接关系到此部分的成败。1、判断基极和管子类型（PNP或NPN）：由于基极与发射极、基极与集电极，分别是两上PN结，它们之间的反向电阻值都很小，所以用万用表欧姆档（R×100或R×1K）测量时，先将任一表黑笔接到某一个认定的管脚上，另一表笔先后接到其余两个管脚上。如果测量得到的阻值都很大（或都很小），然后对换表笔，重复上述测量时，阻值恰好与上述相反。都很小（或都很大）。则可断定所认定的管脚为基极。若不符合上述结果，应另换一个认定管脚重新测量，直至符合上述结果为止。测量时注意管脚和表笔的极性。当黑笔接在基极，红表笔分别接在其它两极时，测得的电阻值都较小，则可判定该三极管为NPN型，反之即为PNP型。2、判断集电极和发射极：判断集电极和发射极的基本原理是把三极管接成基本单管放大电路。利用测量管子的电流放大系数β的大小来判断集电极和发射极。对于常用的NPN型小功率硅管。若集电极接电源正极，发射极接电源负极，这时表针偏较大。如果电压极性反接，则表针偏转就比较小。由此即可判断出集电极和发射极。另一种更可靠的办法是，当肯定被测为NPN型硅管后，将黑表笔接于一个待测的管脚，红表笔接另一个管脚，基极悬空，观察表针偏转情况。然后将黑表笔所接管脚与已断定的基极用手捏住（注意不能使其相碰，这时的人体电阻相当于电阻Rb，比较测量出的阻值变化。然后更换黑、红表笔，再观察阻值变化。如前者的变化比较大，则前者黑表笔所接管脚就是集电极。红表笔所接管脚为发射极。如为PNP管子与上述情况相反。根据上面的方法，测得所使用的三极管是PNP管，其管脚平面对向自己，管脚从左到右为e，b，c。数码管脚的测试就显得比较简单：如果是共阴极的数码管就让位选接地，然后点亮其它的管脚，通过显示结果来断定引脚是a，b，c等，位选也是拿万用表的黑表笔去探试的。如果是共阳极的话就用红表笔来探试位选，其它的与共阴极的一致。经过以上的判断，得到正确的管脚接入电路中。  具体调试步骤为：  （1）键盘是否工作正常  （2）当按下键盘时，数码管显示的数字是否与软件设计的一样。  **6.2.5传感器电路调试**  在进行传感器调试时，首先检查各个传感器的接线电路是否正确，特别是有三个脚的传感器。传感器接到ADC0809时，要接负载电阻，负载电阻的阻值大小有传感器本身决定。  当四个传感器接收到各种异常信号时，如，温度过高时，温度传感器就接收到信号了，通过ADC0809将温度转换成电压，送单片机，与设定的温度进行比较，高于设定温度时，给发光二极管一个低电平，发光二极管就亮，显示报警。  其他传感器的调试也是如此。当四个传感器都能接收异常信号时，说明传感器电路是正确的，调试以后就没问题了。  **6.3 软件程序调试6.3.1软件系统设置**  对于本系统而言，软件程序所实现的功能比较少，程序的调试显得比较简单。整个程序使用的是汇编语言，在keil c51下编写调试完成的。Keil C51是美国Keil Software公司出品的51系列兼容单片机的软件开发系统，其中uVision与Ishell分别是C51 for Windows和for Dos的集成开发环境(IDE)，可以完成编辑、编译、连接、调试、仿真等整个开发流程。开发人员可用IDE本身或其它编辑器编辑C或汇编源文件。然后分别由C51及A51编译器编译生成目标文件(.OBJ)。目标文件可由LIB51创建生成库文件，也可以与库文件一起经L51连接定位生成绝对目标文件(.ABS)。ABS文件由OH51转换成标准的Hex文件，以供调试器dScope51或tScope51使用进行源代码级调试，也可由仿真器使用直接对目标板进行调试，也可以直接写入程序存贮器如EPROM中。  程序要通过烧写器烧写到芯片里。烧写的过程为：假定要写入的文件是“test.hex”。首先，连接编程器主板到计算机并口。然后连接电源。编程器指示灯亮，将89C51插入ZIF插座，锁紧手柄。(请注意芯片的插入方向：缺口向上，朝扳手方向！ 否则会损坏芯片)然后启动软件，鼠标单击在右边的“芯片”下拉框，选择好芯片类型，这里选“89C51”。在“文件”菜单选择“打开Hex文件”，找到“test.hex”文件，单击“打开”。窗口中显示文件内容，右下角显示校验和。单击“写芯片”按钮，编程器的烧写指示灯亮，烧写完毕后报告结果，写入之前会自动擦除芯片内容，写入完毕会自动校验。如果是需要加密，可以选择加密级别。到此为止整个软件程序调试环境就已经配好了。  **6.3.2单片机系统软件调试**  虽然在前面的硬件电路调试过程中每个模块的调试已经穿插了部分模块的软件调试。但是整体单片机系统的调试还是有一定的困难的。例如，在对A/D转换电路的硬件进行测试时，是分别对单通道采样测试。而整体系统要求的是分时显示三个参数以及他们的上下限。因为要显示的数据比较多，所以比较麻烦。刚开始时由于编程时没有设置好各通道显示的标志位，实现不了数码管分时显示的功能。于是从主程序开始检查，发现A/D转换子程序所用的标志位与主程序对子程序调用的标志位不一致。  **6.3.3CAN总线软件调试**      在调试CAN总线的程序时，刚开始是在keil c51下进行调试，编译老是同不过，后来，改了一些程序后，在CAN总线的发送程序和接收程序的开头都加上初始化程序，这样程序就没有错了。  **6.4 调试过程中发现的问题及解决方法**  （1）硬件电路的详细检测问题。硬件电路的正确接线是整个系统能够工作的最基本的前提。在调试前，一定要详细检查电路的焊接和接线。杜绝虚焊、短路和断路等现象。使硬件调试达到事半功倍的效果。  在检查过程中，由于有几跟焊线比较细，焊盘比较小，在焊的过程中，跟其他线接触到了，引起了短路。在发现问题后，又重新焊了一遍，最后检查无误。  （2）接地问题。作者原先就由于没有处理好接地问题而严重影响了调试的进程。地没有接好，影响了数据的正确输出和显示。  所有的地线都应该接在一起，这样就不会引起干扰。  （3）ADC0809通道选择问题。通道选择不正确，就不能把数据采集过来。在调A/D转换的过程中，开始不能把数据采集过来，通过检查发现通道选择不对应。重新修改程序，改正通道选择标记，使数据得以采集过来。  （4）延时问题。本系统为了串口通信的方便采用12M的晶振，延时要比用6M的晶振时要长。如果延时不够，有可能使得系统显示不稳定。  （5）键盘显示程序问题。在第一次把程序烧入到芯片时，接上5V电源后，发现什么都没有，检查硬件电路，是四个三极管的E极没有接上电源。把四个三极管的E极接上5V电源后，数码管有显示。但是不是原先设计的那样。后来，作者检查软件，改了一些程序后，就显示正确了。  （6）CAN总线通信问题。CAN控制器SJA1000的RESET脚是低电平的，AT89C51的RESET是高电平，因此，在连接时要接一个非门。作者在设计电路的时候，忽略了这一点，接上去以后，就正确了。  **6.5 联机调试**  下位机调试完成后，就进行联机调试。  联机调试所需要的设备：  直流稳压电源两台；  单机两件；  排线若干；  联机调试是本系统比较关键的部分。在整个联机调试过程中也遇到了不少问题。例如，没注意将程序与上位机的接收程序相联系，导致数据不能通过CAN总线传给上位机。  在进行联机调试时，把所有的程序结合起来，烧到芯片上。在进行编译时，有一些程序不能匹配，作者就另外修改了一些程序，这样编译就通过了。在本系统中，作者所设计的键盘用来显示小区住户的门牌房间号，在单独调试键盘时，键盘显示正常。但是，联机调试时，发现键盘显示不正常，确定硬件电路没问题时，作者就看程序，发现键盘的程序里，存储器跟ADC0809的存储器用了同一个。发现问题后，作者就另外设了一个变量，用来存储键盘的变量。重新编译后，烧到芯片里，上电后，就显示正常了。  **7  系统使用说明**  本系统使用起来非常方便，下面作者就系统的使用方法作一个简单的介绍。首先  小区的使用用户将各个传感器接在自家的合适位置，例如，将煤气报警传感器接在厨  房，将温度传感器和烟雾传感器接在客厅，将红外传感器接在门口。然后接上电源就能工作了。  当有报警时，小区控制中心的报警灯就会亮。例如当温度传感器检测到室内温度过高时，就会将这个异常信号由ADC0809采集，送单片机处理，此时，与温度传感器相对应的发光二极管就会亮。其他情况也是如此。  当有报警信号时，小区的住户可以按下装在家里的按键，按下自己家里的房间号，这样，房间号就会显示在小区控制中心，小区控制中心就可以知道发生的具体情况和具体房间号。  本装置使用起来简单易学，而且功能比较强大。  **8  结论**  本文设计和实现了一种基于单片机AT89C51的小区安防报警系统。该系统主要以方便、廉价、可靠和实用等为目的，主要以小区各个住户为主要对象，采用了ADC0809数据采集，利用AT89C51作为控制器，通过简易键盘设置报警上限，共阳极四位数码显示，由二极管发光进行报警。同时，此系统与另外一个同学的小区网站组合成一个完整的系统，能在小区网站上实现安防实时监控。  本文的重点是ADC0809的四路模拟电压采集以及单片机键盘显示最小系统。在实际的实现过程中，遵循从易到难，从硬件到软件的过程。首先理解设计任务以及系统要求实现的功能，有针对性地查阅相关的资料，熟悉所需芯片以及元器件的工作原理，再着手设计硬件电路并根据硬件电路的设计特点，有步骤地对硬件电路进行分模块调试。  本系统有四个传感器来监测家庭异常情况，如有异常情况发生时，传感器就将检测到的信号经过ADC0809采集后，送到AT89C51进行处理，同时，报警灯亮，显示报警。在此系统中，键盘用来设置小区家庭报警的门牌房间号，数码管用来显示。  由于本人的知识和能力有限，对其中的某些问题，如对本系统对讲的保密性等，未能进行深入分析研究，此系统也只达到了一部分的实际效果。  文中定有许多不足之处，敬请各位老师和读者不吝赐教。  **谢  辞**  挥手间四年已逝，回首近四年的学习生活，其间有失落也有收获，有艰辛也有快乐，有迷茫也有方向。在桂林电子科技大学这最美好、最宝贵的四年学习生活中，我由一个当初少不更事的孩子成长为一个对社会、对人生充满自信的青年。在这里需要感谢的有很多。  真心地感谢各位老师四年来的传道、授业和解惑，给了我人生一笔永恒的财富。  本次毕业设计，是大学生涯结束最后一堂结业课，有幸龙超老师作为指导老师，整个做毕业设计阶段，由于平时多理论，少实践的学习，导致做毕业设计过程中，常常遇到这样那样的问题，当然除了自己看书，查资料解决问题外，也常常求助龙老师，龙老师在帮助学生解决问题的过程中，态度亲和，并且循循善诱的提示去启发学生自己思考，老师“授之于鱼不若授之于渔”的良苦用心学生体会很深，故一直心存感激。  感谢毕业设计过程中所有给我真诚帮助的老师和同学们。  最后感谢我的母校桂林电子科技大学，给我提供学习和生活的空间，真心地希望它的明天更加美好！  **参考文献**  [1] 尹建华 张惠群 刘鲁源 刘迎澍编著.微型计算机原理与接口技术[M].北京：高等教育出版社，2002.  [2] 范逸之编著.Visual Basic与RS-232串行通信控制[M].北京：中国青年出版社, 2002.  [3] 江思敏编著.Protel电路设计教程[M].北京：清华出版社，2002.  [4] 余海生等编著.微型计算机控制技术[M].北京：清华大学出版社，2003.  [5] 靳达编著.单片机应用系统开发实例导航[M].北京：人民邮电出版社,  2003.  [6] 谢自美编著.电子线路设计.实验.测试[M].武汉：华中科技大学出版社，2000.  [7] 闫玉德 俞虹编著.MCS-51系列单片机原理与应用（C语言版）[M].北京：  北京机械工业出版社，2002.  [8] 李朝青编著.PC机及单片机数据通信技术[M].北京：北京航空航天大学出版社, 2000.  [9] 张增强等编著.突破Delphi 7.0 编程百例[M].北京：中国水利水电出版社 ，2003.  [10] 石东海编著.单片机应用系统设计从入门到精通[M].西安：西安电子科技大学出版社, 2002.  [11] 崔建华 郭瑞军编著.Delphi串口通信工程开发实例导航[M].北京：人民邮  电出版社，2004.  [12] 李学海编著.EM78单片机实用教程—基础篇[M].北京：电子工业出版社，2003.  [13] M . Herry. Analog Integrate Circuits WILEY.2001.  **附  录**  附录一  系统原理图  附录二     系统PCB图 |