**项目文档详述**

实训课题： 嵌入式智慧农场环境监测系统

专 业： 计算机科学与技术 班 级： 20级2班

姓 名： 王陈友 学 号： 200102122

指导老师：顾勤凤，刘益

报告撰写日期： 2022 年 4 月 22 日

|  |
| --- |
| **项目背景：**  物联网环境监测系统是一个新生产物，在延续传统环境监测监控系统优势基础上研发而成，可以实现自动监控系统中安全可靠的数据采集、处理和传输。数据采集终端设备纳入物联网系统，并且它们可以直接互联互通、实现自组局域网，相互协作完成特定的业务（比如目前人们日常生活最敏感的温度湿度、光线和 PM2.5 等）。通过主数据采集终端设备节点，实现远程数据采集终端的智能软件；然后通过无线传输至中心服务器处理存储；最后用户通过手持设备（比如手机、平板电脑等）实时查看当前环境数据。这些产品应用范围广阔，可应用于环保、农业、工矿、电信、市政、交通等工业和公用事业中的各种类型自动监测站的数据采集与传输领域。  NB-IoT 通信采用超窄带、重复传输、精简网络协议等设计，以牺牲一定速率、时延、移动性性能，获取面向 LWPA 物联网的承载能力。NB-IoT 作为一种新的窄带蜂窝通信 LPWAN（低功耗广域网）解决方案，将给物联网行业带来巨大的变革与创新。  本项目传统环境监测监控系统统建设的基础上，结合物联网的技术和 NB-IoT 窄带物联网通信技术，根据环境监测监控系统建设的新形势和新要求而提出来的一个全新的嵌入式项目。  **实训内容：**  本项目主要分为六大流程：  **一：开发板数据采集**  通过湿温度一体化的数字传感器DHT11进行环境温湿度数据采集，该传感器包括一个电阻式测湿元件和一个 NTC测温元件，并与一个高性能 8 位单片机相连接。通过单片机等微处理器简单的电路连接就能够实时的采集本地湿度和温度。其管脚排列图如下：  DHT11数据发送，首先首先主机发送开始信号，即：拉低数据线，保持 t1（至少 18ms）时间，然后拉高数据线 t2（20~40us）时间，然后读取 DHT11 的响应，正常的话，DHT11 会拉低数据线，保持 t3（40~50us）时间，作为响应信号，然后 DHT11 拉高数据线，保持 t4（40~50us）时间后，开始输出数据，输出数据主要包括0和1两个数字，下面是**读取的单个位数据：**  // 读取单个位数据  uint8\_t DHT11\_Read\_Bit(void)  {  uint8\_t retry = 0;  while (DHT11\_DQ\_IN && retry < 100) // DHT11延时40~80us  {  retry++;  delay\_us(3);  }  retry = 0;  while (!DHT11\_DQ\_IN && retry < 100) // DHT11延时40~80us  {  retry++;  delay\_us(3);  }  delay\_us(50); //??40us  if (DHT11\_DQ\_IN)  return 1;  else  return 0;  }  **二、数据的收发**  对于采集到的温湿度数据，我们需要通过串口发送到我们对应的UDP服务器上，所以先在KILL项目工程中编写好数据的读取程序，将采集到的数据存入到自定义的字符数组中备用。  // 读取字节数据值  uint8\_t DHT11\_Read\_Byte(void)  {  uint8\_t i, dat;  dat = 0;  for (i = 0; i < 8; i++)  {  dat <<= 1;  dat |= DHT11\_Read\_Bit();  }  return dat;  }  **三、UDP服务器搭建**  在服务器的搭建中，我们选择UDP服务器，通过串口助手进行数据的收发，将从开发板传送过来的数据通过UDP协议发送到我们的服务器，再通过服务器将数据存储到数据库中。这里的数据库，我们采用SQLITE3，在嵌入式开发中更为方便，有效。  在搭建服务器时，要注意绑定的IP地址以及所使用的端口号，在对应的串口助手中一定一一对应。  不仅如此，我们在接收到来自串口的数据后，需要对数据进行处理，将多余的字符筛除出去，所以这里有一个字符串的裁剪，我的方法同老师的方法略有不同，**代码如下：**  int mySubstr(char \*buf, char \*s, int l)  {  int len = strlen(buf), idx = 0;  int i = 0;  for (i = l; i <= len; i++)  {  if (buf[i] >= '0' && buf[i] <= '9' && i < len)  {  s[idx++] = buf[i];  }  else if (idx > 0)  {  break;  }  }  return i;  }  于此同时，我们还需要做到对数据库的插入操作，为了避免像数据库中，我们在每一次打开服务器的时候，可以先将数据库中的数据清除掉，这样可以节约资源。只要是涉及到关于数据库的操作时，我们都需要找到数据库的地址，其次打开数据库，向我们的数据表中插入或删除数据。  **数据清除代码如下：**  void deltable()  {  sqlite3 \*db;  char \*err = 0;  int re = 0;  // 开数据库  re = sqlite3\_open("/var/www/xuedao", &db);  // 清表  re = sqlite3\_exec(db, "delete from wcy",NULL,NULL, &err);  if (re != SQLITE\_OK)  {  printf("sqlite3 delete error\n");  exit(1);  sqlite3\_close(db);  }  sqlite3\_close(db);  }  **数据插入代码：**  void sqlinsert(int idx, char \*temp, char \*humi)  {  sqlite3 \*db;  char \*err = 0;  int re = 0, empty = 1;  // 开数据库  re = sqlite3\_open("/var/www/xuedao", &db);  if (re != SQLITE\_OK)  {  printf("sqlite3 open error\n");  exit(1);  sqlite3\_close(db);  }  // 插数据  char sql[1024] = {0};  re = sprintf(sql, "insert into wcy values(%d,%s,%s)", idx, temp, humi);  re = sqlite3\_exec(db, sql, NULL, NULL, &err);  if (re != SQLITE\_OK)  {  printf("sqlite3 insert error\n");  exit(1);  sqlite3\_close(db);  }  sqlite3\_close(db);  }  当然，在后期的开发中，我们还需要通过按钮来控制我们的LED灯，所以在这里我们还需要将数据从数据库中读取出来，再发送到我们的开发板上，点亮LED灯。  void sqlselect()  {  sqlite3 \*db;  char \*err = 0;  int ret = 0;  int empty = 1;  // 打开数据库  ret = sqlite3\_open("/var/www/xuedao", &db);  if (ret != SQLITE\_OK)  {  printf("sqlite3 open error \n");  exit(1);  }  // 查询  char str[100] = "SELECT \* FROM wcyled";  ret = sqlite3\_exec(db, str, myfunc, &empty, &err);  if (ret != SQLITE\_OK)  {  printf("sqlite3 selsect error \n");  sqlite3\_close(db);  exit(1);  }  sqlite3\_close(db);  }  在进行数据库的查询中，所涉及到了一个回调函数，这里并没有进行什么特殊的处理，只是将数据库中读取出来的标志，给存储下来，在通过sendto函数发送到开发板上即可。  char flg;  int myfunc(void \*p, int argc, char \*\*argv, char \*\*argv\_name)  {  flg = \*argv[0];  return 0;  }  sendto(sock, &flg, 1, 0, (struct sockaddr \*)&dst, rcv\_len);  **四、HTML界面搭建**  HTMLWeb界面的设计与搭建，在本次项目实训中选择了cgi方式进行HTML网页搭建，只需要将原本的HTML代码放进C语言的pritf()中即可，最后编译成CGI文件即可。  void show\_head()  {  printf("Content-type: text/html\n\n");  printf("<html>\n");  printf("<head>\n");  printf("<meta http-equiv=\"refresh\" content=\"1\">\n");  printf("<title>Group2</title>\n");  printf("<style type=\"text/css\">\n");  printf("tr {font-size: 85px; font-weight: bold;font-family: 微软雅黑;text-align: center}\n");  printf("button { font-size: 85px; font-weight: bold;font-family: 微软雅黑; background-color:#F5DEB3;\n");  printf("display: inline-block; cursor: pointer; transition-duration: 0.5s; border-style:none}\n");  printf(".button2:hover{box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24),0 17px 50px 0 rgba(0,0,0,0.19);}\n");  printf("div {position: relative; margin:auto; text-align:cneter;}\n");  printf("</style>\n");  printf("</head>\n");  printf("<body bgcolor=\"#888888\">\n");  printf("<div>\n");  printf("<center>\n");  //printf("<h1>智慧农场监测系统</h1>\n");  }  **五、CGI数据展示**  在前面HTML的基础上，我们需要将数据库中的数据读取出来，并显示到我们的HTML网页上，所以这里也是同之前的步骤一样，打开数据进行数据查询即可，在查询数据时，我们需要着重运用到回调函数，将回调函数中的各项值读取出来再放入到HTML中进行数据的展示。  int myfunc(void \*p, int argc, char \*\*argv, char \*\*argv\_name)  {  show\_head();  // 表格详细数据  printf("<table border=\"0\" cellspacing=\"0\" width=\"100%\" height=\"100%\">\n");  printf("<tr style=\"background-color:#FFCC33;\">\n");  printf("<td colspan = '2'>智慧农场环境监测系统</td>\n");  printf("</tr>\n");  printf("<tr style=\"background-color:#CCFF66;\">\n");  printf("<td>温&nbsp;&nbsp;度：</td>\n");  printf("<td>%s</td>\n",argv[1]);  printf("</tr>\n");  printf("<tr style=\"background-color:#CCFF99;\">\n");  printf("<td>湿&nbsp;&nbsp;度：</td>\n");  printf("<td>%s</td>\n",argv[2]);  printf("</tr>\n");  char str[100] = {0};  show\_time(str);  printf("<tr style=\"background-color:#FFFF99;\">\n");  printf("<td>时&nbsp;&nbsp;间：</td>\n");  printf("<td>%s</td>\n",str);  printf("</tr>\n");  printf("<tr style=\"background-color:#FFF8DC;\">");  printf("<td colspan=\"2\">\n");  printf("<button class=\"button2\" onclick=\"window.location.href='group2\_wcy\_open.cgi'\">开&nbsp;&nbsp;灯</button>\n");  printf("<button class=\"button2\" onclick=\"window.location.href='group2\_wcy\_close.cgi'\">关&nbsp;&nbsp;灯</button>\n");  printf("</td>\n");  printf("</tr>\n");  printf("</table>\n");  // 表格数据结束  show\_end();  return 0;  }  其中包含了当前的实时时间的显示，这里只是调用了C语言内置的时间库进行了简单的处理，代码如下：  void show\_time(char \*str)  {  time\_t timer0;  timer0= time(NULL);  struct tm\* tt = localtime(&timer0);  int y = tt->tm\_year + 1900,m = tt->tm\_mon + 1, d = tt->tm\_mday;  int h = tt->tm\_hour, mi = tt->tm\_min, s = tt->tm\_sec;  sprintf(str,"%02d-%02d-%02d %02d:%02d:%02d",y,m,d,h,mi,s);  }  **六、Web端LED控制**  在Web页面想要对开发板的LED进行控制，那么我们便需要一个控制信号，随后对开发板发出控制指令，完成操作。在这里我们编写了额外的两个网页用于跳转时向数据库中插入控制数据，最后通过服务器查询数据达到控制LED灯的效果。在每一个操作执行前，首先应清空数据库中的数据，然后再插入数据，这里没有用到查询数据，所以没有回调函数，但是为了方便调试，查看数据库中的值是否被修改，所以这里运用了一个数据库查询操作，我将网页代码封装在了回调函数中。  开灯主要代码展示：  void insTable()  {  sqlite3 \*db;  char \*err = 0;  int re = 0, empty = 1;  // 开数据库  re = sqlite3\_open("/var/www/xuedao", &db);  if (re != SQLITE\_OK)  {  printf("sqlite3 open error\n");  sqlite3\_close(db);  exit(1);  }  int idx = 1;  // 插数据  char sql[1024] = {0};  sprintf(sql, "insert into wcyled values(%d)",idx);  re = sqlite3\_exec(db, sql, NULL, NULL, &err);  if (re != SQLITE\_OK)  {  printf("sqlite3 insert error\n");  sqlite3\_close(db);  exit(1);  }  sqlite3\_close(db);  }  关闭LED灯，同理，只需要在插入数据时将数据改为0即可。  下面时完成的项目成果展示：        **实训心得：**  这次的嵌入式实训让我受益匪浅，让我更深入地理解了嵌入式系统的设计和实现。在这次实训中，我们通过STM32上的DHT11模块读取温湿度，然后撰写了一个UDP服务器，将读取的数据发送到服务器，服务器将数据存储到数据库，最后通过CGI编写的HTML文件将数据展示在网页上面。  首先，我们学习了如何在STM32上使用DHT11模块读取温湿度数据。通过这个模块，我们可以实时获取室内环境的温度和湿度，这个实现对于很多实际应用场景都是非常有用的，比如智能家居系统、空调温度自动控制系统等等。在实现过程中，我们了解了STM32的GPIO和定时器等外设的使用方法，也学习了如何通过中断的方式读取传感器的数据。通过这些学习，我更加深入地理解了嵌入式系统的硬件设计和实现。  接下来，我们实现了一个UDP服务器。UDP是一种无连接的网络协议，它的主要特点是传输速度快，但是传输可靠性不高。在实现UDP服务器的过程中，我们了解了UDP协议的基本原理，如何建立UDP连接、如何发送和接收数据等等。同时，我们也学习了Linux系统下的socket编程，掌握了如何在Linux系统下实现UDP协议的编程实现。  在实现UDP服务器的过程中，我们还需要将STM32与服务器进行串口通信，将读取的温湿度数据发送到服务器上。这个过程中，我们也学习了如何使用UART串口通信协议，以及STM32的串口外设的使用方法。通过这个实现，我更加深入地理解了嵌入式系统和计算机系统的通信原理和实现方法。  在本次实训中，我们使用了sqlite3作为数据库存储的解决方案。sqlite3是一种轻量级的关系型数据库管理系统，它支持SQL语言标准，并提供了方便的API接口，使得在嵌入式系统中使用sqlite3非常方便。在实现过程中，我们首先需要创建一个数据库，并定义数据表结构。在本次实训中，我们定义了一个名为“wcy”的数据表，用于存储从STM32读取到的温湿度数据。该数据表包含3个字段，分别是“id”（数据记录编号）、“temperature”（温度值）、“humidity”（湿度值）。  在程序中，我们使用sqlite3提供的API接口，实现了从UDP服务器读取数据并存储到sqlite3数据库的功能。具体来说，我们使用sqlite3\_open()函数打开数据库连接，使用sqlite3\_exec()函数执行SQL语句，创建数据表和插入数据，使用sqlite3\_close()函数关闭数据库连接。  通过使用sqlite3，我们可以方便地实现数据的存储和查询功能，使得我们可以将实时获取的温湿度数据进行持久化存储，方便后续的数据分析和应用。同时，sqlite3还具有轻量级、高效、可移植等特点，非常适合在嵌入式系统中使用。  最后，我们使用CGI编写了一个HTML页面，将从服务器上获取的温湿度数据展示在网页上面。通过这个实现，我了解了如何使用CGI和HTML编写Web应用程序，并且了解了如何从数据库中获取数据并展示在网页上面。这个实现也为我打开了Web应用程序开发的大门，让我更加深入地了解了Web开发的实现原理。  在使用CGI编程的时候，遇到了数据错误,同样的代码在别的地方能够正常运行，可是在打开LED灯的时候，就会出现问题，最后经过老师的讲解，才发现数据库权限的问题，最后才得以成功。  虽然在本次实训的项目开发中遇到了很多BUG，同时也因为服务器总是崩溃导致心态炸裂，但最终我都克服了这些困难，在老师的带领下，独自一人完成了整个项目，总的来说这次的嵌入式实训让我深入地了解了嵌入式系统的设计和实现方法，同时也让我学习了Linux系统下的网络编程和Web开发。通过这次实训，我不仅掌握了更多的技术知识，也提高了我的动手实践能力和解决问题的能力。 |