基于MSP430下A5191HRT和AD421的HART协议的设计

任红星
(陕西国防工业职业技术学院 陕西西安 710300)

摘 要:叙述HART协议的通信结构和层次结构,主要完成以MSP430为主控制器,基于HART调制解调器A5191HRT和电流环数模转换器AD421的HART协议通信物理层设计。

关键词:HART协议 串行通信 频移键控
中图分类号:TN914.3 文献标识码:A 文章编号:1674-098X(2009)06(a)-0026-02

1 引言

HART(Highway AddressableRemote Transducer,可寻址远程传感器高速通道)通信协议由美国Rosemount公司于20世纪80年代中期推出,之后又进行了修订和增补,作为开放标准由HART 通信基金会HCF向全世界发布,主要用于现场智能仪表和控制室系统间进行数字通信。由于兼容传统的4~20mA模拟系统的HART 通信协议具有诸多优点,发展至今,它已被世界上三分之二以上的现场设备所采用,成为智能工业控制领域事实上的国际标准。

2 HART通信协议简介

HART协议参考ISO/OSI模型(开放系统互联模型),采用简化的3层模型结构,即第一层物理层、第二层数据链路层和第七层应用层。物理层规定信号的传输方法,采用基于Bell202标准使用频移键控FSK(Frequency Shift Keying)技术将数字信号变换为音频信号叠加在低频4~20 mA模拟信号上叠加音频数字信号进行双向通信,数字信号幅度为0.5 mA,协议规定的信号频率(1200 Hz代表1,2200 Hz代表0)和传输速率(1200 bit/s)。如图1所示,图2为HART协议的数字信号和模拟信号叠加后传输图。这些音频正弦波的平均值为零,所以在现存的模拟信号中不增加直流成分,因此在二根线上可以同时传送互不影响的模拟和数字信号。正是由于HART协议的这种优点,使它成为工业现场广泛应用的、事实上的工业标准。

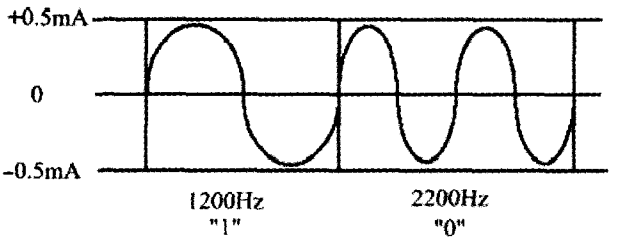

图1 HART协议通信的FSK信号

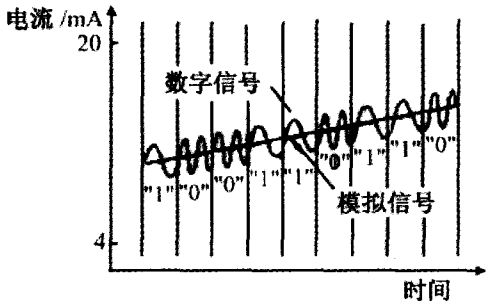

图2 HART数字通信信号叠加在电流环上

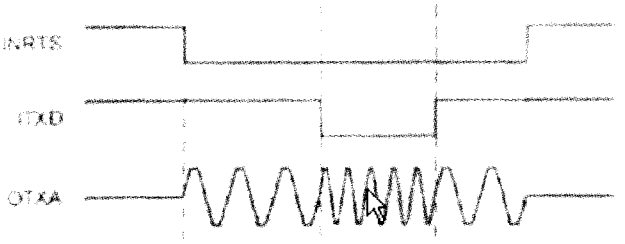

图3 调制过程波形

3 HART协议物理层设计

根据HART协议物理层的设计要求,设计了物理层电路。其中MSP430为控制器(CPU),A5191HRT是美国AMI公司专为实现HART协议而设计的调制解调器芯片它内部包括发送数据调制器与波形整形电路、载波检测电路、接收滤波器与解调电路、控制逻辑和时钟振荡器电路.A5191HRT的ORXD和ITXD分别和MSP430的异步串行通信口的接收端P3.5、P3.4口连接,INRTS和OCD分别接P2.6、P2.7口,当主设备从串行通信口发送命令时电流环上的HART信号送到A5191的接收滤波器进行滤波和解调为"0""1"数字信号,MSP430接收到有效的HART通信信号后进行命令分析,然后将返回相应的通信信号给A5191进行调制和波形整形后,通过Ch19加到AD421的C3引脚。OCD端为载波检测输出,当IRXAC检测到有效的输入时OCD端变为高,所以CPU可以通过检测OCD脚来进入接收状态。INRST为发送请求引脚,当为低电平时,调制器工作,解调器关闭。调制器模块接收由ITXD引脚输入的不归零制(NRZ)数字信号,生成FSK调制信号由OTXA引脚输出。

AD421是美国模拟器件公司生产的16位数模转换器,可将输入锁存器的数字信号转换为4~20mA的电流信号。AD421和MSP430的串行SPI口相连,即数据线(DATA)--P5.1(MISO)、时钟线(CLK)--P5.3(UCLK1)、锁存线(LATCH)—P5.2(MOSI),AD421在时钟的控制下,输入移位寄存器的把DATA引脚上的信号按位依次读入,LATCH锁存脉冲把数据锁存到DAC中,转换位相应的电流然后经过三级阻容电路进行滤波。(三个电阻在AD421的芯片内,三只电容必须外接在C1、C2、C3引脚上),为满足HART的信号频率Ch4、Ch5、Ch6分别取0.01uF、0.5uF、0.16uF。

HART输出在电流环路上的电流峰峰值应为1mA可通过Ch4上的施加电压来实现,为了实现HART的模拟和数字通信而且要确保注入电流环路上不干扰模拟电流值,Ch4和Ch19将A5191HRT送来的信号进行衰减以确保HART传来的信号为1mA的峰峰值。根据AD421的内部电路C3引脚上的信号为20mA才能在环路上产生1mA的电流。根据A5191的调制输出幅度可的Ch19为6.2nF。

4 调试过程

当INRTS引脚为低电平时,调制器工

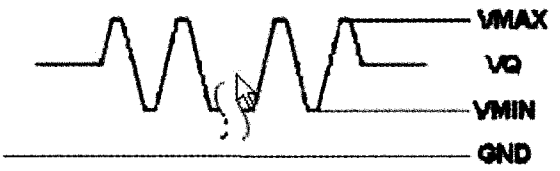

图4 发送数据波形

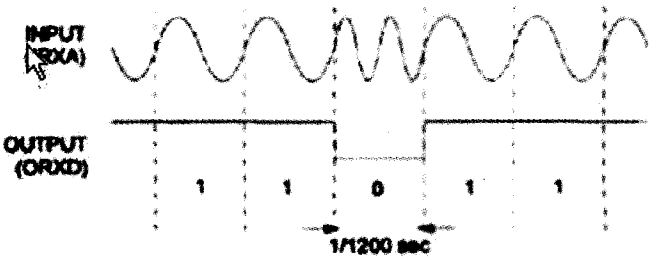

图5 为解调过程波形

作，解调器关闭。调制器模块接收由ITXD引脚输入的不归零制(NRZ)数字信号，生成FSK调制信号由OTXA引脚输出。波形如图3所示。

当ITXD引脚为高电平时，OTXA引脚输出1200Hz的梯形波；ITXD引脚为低电平时，OTXA引脚输出2200Hz的梯形波，OTXA输出通常需要和放大器或缓冲器进行交流耦合。OTXA的输出电压幅度由IAREF引脚上的参考电压决定，其波形示意图见图4。图4中VQ为OTXA上的静态电压(非调制状态)，VREF为IAREF引脚上的参考电压。VREF为1.235V时，VQ为0.5V，信号电压为0.25V到0.75V，这正好满足HART主设备连接到网络上的电压要求。但由于OTXA没有足够的驱动能力直接接入HART网络，所以需要连接一个缓冲放大器。在工业现场仪表的应用中，通常将OTXA上的0.5V峰峰电压输出转换为1mA峰峰电流输出。

图5为解调过程波形当INRTS引脚为高电平时，解调器工作，调制器关闭。接收到的信号需经过一个带通滤波器。这个滤波器的一部分被集成到A5191HRT内部，其余元件外接用以降低电源变化带来的影响。整个带通滤波器由一个单极点低通滤波器和一个四极点高通滤波器构成。A5191HRT需要IAREF和ICDREF两个电压参考源，IAREF为芯片内部的放大器和比较器提供电压基准，通常为1.235V。ICDREF用于载波检测，应比IAREF低0.08V。

载波检测：

当IRXAC引脚上的电压小于ICDREF引脚上的电压时，AD421中的比较器输出逻辑低电平。这个输出被引入一个载波检测模块，当INRTS为高且有四个连续脉冲到达时，OCD置高，下一个有效脉冲必须在2.5ms内被接收才能使OCD保持高电平。

HART时钟模块：

HART芯片正常工作需要460.8 KHz的时钟信号，可在引脚OXTL和IXTL间连接一个晶体或陶瓷谐振器，或在引脚OXTL上连接外部时钟同时将引脚IXTL接地。本电路中主要由MSP430产生一个460.8 KHz的时钟信号。

5 结语

本文HART协议通信协议物理层设计主要用于流量计、变送器等设备。经过实验和调试以及现场运行证明是可行的。而且方便可靠实用性好。

(上接25页)

由于页面缓存的原因而仅读取本地内容)，并将其赋值给url变量，这是一个普通的URL串，同时传递了两个变量，分别是timestamp和xd。其中后面的xd值是我们所需要的。

```
xmlHttp.onreadystatechange =handleSelNj;
```

的含义是当调用返回时(即xmlHttp对象状态改变时)执行handleSelNj函数过程。

```
xmlHttp.open("get", url, false);
xmlHttp.send(null);
```

该两行的做用是具体的执行动作，即根据url所指定的地址，进行url访问，并传递相关的参数，其传递方法是get方法(相应地数据获取要用request.querystring方法)，false值指示当url中的代码文件指行时，可以进行异步处理(即产生零等待的效果)。

url中所指示的文件selnj.asp，其代码如下：

```
<!--#include file="Read_Xd.asp" -->
<%
xd=request.querystring("xd")
AdoOpen cnr,rst,"select distinct nj from ClassInfo where xd='"&xd&"'"
Do While not(rst.eof or rst.bof)
nj=rst("nj")
StrOptions=StrOptions & "<option value=" & nj& ">" & nj&"年</option>"
rst.movenext
if rst.eof then exit do
Loop
AdoClose cnr,rst
        response.write escape(StrOptions)
%>
```

第一行是数据库读写文件，此处略。第三行是获取前面传递过来的变量xd，即获取要查询年级所在的学段，然后根据xd值来读取数据库中的相应的年级，即第4-11行，然后在第12行，进行输出如下的结果(Select控件中的Option选项)：

```
<option value=二>二年</option>
<option value=三>三年</option><option value=一>一年</option>
```

第12行是一个很关键的行，这里是调用response对象的write方法，其中比较特别的是escape函数，该函数的作用是对其中的字符串进行编码，防止其中含有中文时response.write输出会产生乱码的问题。

接下来会触发

```
xmlHttp.onreadystatechange =handleSelNj;
```

，该行含义是：当XMLHTTP对象的状态被改变时，调用handleSelnj函数，如下：

```
26 function handleSelNj() {
27     if (xmlHttp.readyState == 4) {
28         if(xmlHttp.status == 200) {
29         SetNj(unescape(xmlHttp.responseText));
30         }
31         else{
32             alert("异常错误! ");
33         }
34     }
35 }
```

该部分大意是指：如果返回结果正常，则调用SetNj过程(第27-30行)，否则显示“异常错误”的提示(第31-33行)。其中Unescape?函数是对前面进行了escape编码的字符串进行反编，恢复为正常输出。

setNj()是一段Vbscript代码，如下：

```
58 sub setNj(v)
59     alert(v)
60     v="<option value=''>请选择</option>"&v
61     document.getElementById("tdnj").innerHTML=
"<select size=1 name=nj id=nj class=cssSelect onchange="
62 &chr(34)&"vbscript:selbj"&chr(34)&">"&V&"</select>"
63 end sub
```

该段代码是首先接受由setnj传递过来的参数，然后在第60行中，增加一个“请选择”提示项。然后在第61行中将该内容连同构造的Select控件，一同放到id=”tdnj”的单元格中。该单元格的定义如下：

```
<td id="tdnj"><select size="1" name="xd"
onchange="vbscript:selbj"></select></td>
```

运行后即可发现，级联菜单如桌面应用一样，没有等待延时。

以上为选择年级的过程，班级部分的选择过程与年级大同小异，大家可以参考年级部队分自行分析。

通过以上讲解，总结一下Ajax技术的应用步骤如下：

先由一个事件来触发一个过程，该过程中首先生成xmlHttp对象，然后构造可传递参数的字符串，并根据字符串内容来异步(同步)调用服务端相应的代码，经过在服务端处理后，生成可以在客户端直接显示的字符串，然后返回给客户端来，由客户端的相应过程来决定字符串的最终的显示方式与位置，以完成整个过程的调用。

这里比较重要的部分一个是xmlHttp对象，一个是对ID参数的定义与调用。

通过以上应用可以看出，Ajax方法编程，可以解决因Submit动作所产生的页面刷新与等待，给人以桌面程序的感受。

参考文献

[1] Ajax基础教程.人民邮电出版社.
[2] Ajax实战：实例详解.人民邮电出版社.,