**实验8 单片机异步串行通信**

**一、实验目的**

1. 了解异步串行通信原理；

2. 掌握MSP430异步串行通信模块及其编程控制方法。

**二、实验任务**

**1. 单片机串口与计算机USB虚拟串口连接后，计算机串口的检测**

因外部32.768KHz晶振在扩展电路板上，需把扩展板接到单片机板上，才能使用该晶振，注意把JP8处晶振相关的跳线接至晶振侧，参见实验5的图5-1。

如图8-1，先将USB转串口模块的发送TxD、接收信号RxD、GND分别与扩展板上msp430G2553单片机串口的接收引脚P1.1(RxD)、发送引脚P1.2(TxD)、GND相连，**注意不要接错了**！，再将USB转串口模块插入计算机侧的USB接口中，构成计算机利用USB虚拟串口与单片机串口进行通信的硬件接线。

由于串口助手软件有时会在运行中出现异常，造成不正常工作情况。故在做与单片机串口通信的实验中，还会需时不时通过自发自收检测计算机的USB转串口，在串口助手的控制下能否正确工作。

麻烦的是测试计算机USB转串口的自发自收，需要断开与单片机的连线，然后将USB转串口的TxD、RxD信号短接，测试完在断开短接线，然后再和单片机的串口接线。

此时可以用下面方法来测试计算机侧USB转串口的自发自收：

如图8-2，在扩展板上，单片机的引脚P1.1/RxD和P1.2/TxD共有两处引出。其中一处以在图8-1中用于USB转串口发收信号TxD和RxD对接。如图8-3，只要将另一处中的P1.1和P1.2短接，则在硬件连线上，相当于USB转串口发收信号TxD和RxD短接了。接好线后，在CCS下进入DEBUG后，此时处于暂停状态，由于单片机上电或复位时P1.1和P1.2的初始设置为基本输入功能，这样在图8-3中，只有USB转串口模块的TxD向连接点发送数据(输出)，其他的引脚均从连接点为接收数据(输入)，满足电气上的信号方向不冲突。这样任何时刻，在DEBUG下，只要按菜单栏上的复位按钮，短接P1.1和P1.2，就可以按照任务1中给出方法测试计算机串口自发自收，且单片机与USB转串口的线接上后也能随时检测。

当计算机侧USB转串口检测成功后，断开图8-3中P1.1和P1.2间的短接线，回到图8-1单片机串口与计算机USB虚拟串口收发对接状态，进行计算机与单片机两者之间的串行通信。注意，在DEBUG下如果运行了的自己编写的串口控制程序，会将P1.1、P1.2设置为单片机的串口引脚，此时即使短接了P1.1和P1.2，当前信号关系参看图8-4，不满足电气上的信号关系，不能做计算机侧串口自发自收的检测接线。如果想重新检测计算机串口的自发自收，只需在CCS下按复位restart按钮 ，将P1.1、P1.2恢复到复位时的基本输入状态，然后可以检测。改线期间以及之后，不要断开USB数据线与单片机的连线，否则都需重测计算机USB虚拟串口的自发自收，确保计算机侧串口正常受控串口助手。

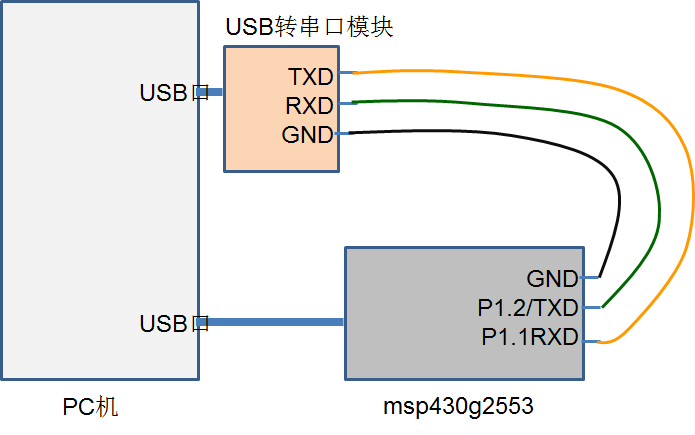
****

图8-1 单片机与计算机利用USB转串口进行通信

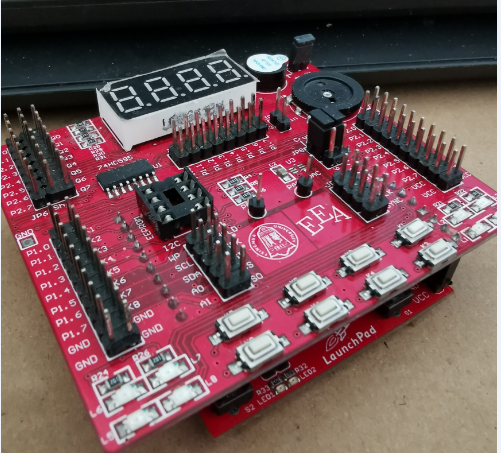


图8-2 扩展板上两处P1.1/RxD、P1.2/TxD

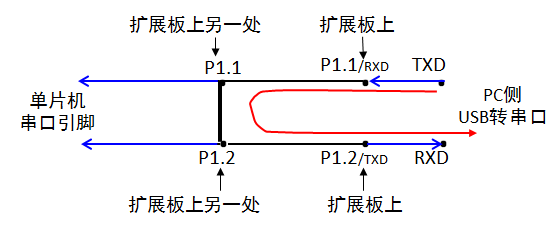


图8-3 短接设为基本输入的单片机P1.1、P1.2引脚

与USB虚拟串口的TxD、RxD对接时的信号关系

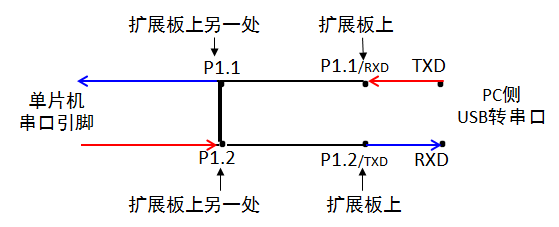


图8-4 短接设为串行接口的单片机P1.1、P1.2引脚

与USB虚拟串口的TxD、RxD对接时的信号关系

1. **控制单片机与计算机实现异步串行通信**

掌握了计算机侧USB转串口模块是否正堂工作的检测方法后，按图8-1接好单片机与计算机的串口通信连线，卸下引脚P1.1与P1.2间的短线块。**完成下面内容：**

1. 阅读程序L8\_TestSCI.c(提供电子版)，调试完成程序功能：设置数据格式为波特率9600bps、无校验、8位数据、先低后高、1个停止位，利于用单片机异步串行接口，**采用查询方式，**实现单片机与计算机之间的数据通信，掌握单片机串口通信编程和调试的基本方法。

***步骤：***

在计算机上运行串口助手，打开USB转串口的端口号，在CCS下将程序L8\_TestSCI.c编译连接，进入到DEBUG下，按任务2的方法检测计算机侧USB虚拟串口的自发自收成功后，运行程序。在串口助手的发送窗口输入5个字符，比如“12345”，**并按回车**，然后点击串口助手上的“手动发送”，发送区输入的数据经单片机串口接收后，又发送至计算机，并在串口助手中显示出来。注意加上回车操作中包含的回车和换行两个字符（\r\n），实际输入了7个字符。串口助手发送的是这些字符对应的ASCII，即发送的是 0x31，0x32，0x33，0x34，0x35，0x0d，0x0a，其中0x0d、0x0a分别是回车符和换行符的ASCII码。如果在串口助手上，选择用十六进制发送，则应输入 31 32 33 34 35 0d 0a，中间需用空格分开，然后点击手动发送。选择用十六进制显示的话，看到接收到的数据应是 31 32 33 34 35 0d 0a，有时会看到个别误码的情况，即收到的数据与发送的数据不一致，这要细查原因。

***注意：***

实验中单片机与计算机通信不成功，有时可能是计算机侧的串口助手未能正常工作。可通过检测计算机的自发自收成功，确认计算机的串口受串口助手控制。计算机侧正常工作，再细查单片机侧的问题。

1. 在任务1)接线基础上，添加按键K1、发光二极管L1和L2与单片机的接线，编程完成：

在实验板上按下按键K1，单片机向计算机发送字符串“MCU get ready！”；然后，单片机等待计算机侧发送的一个字符信息，单片机收到字符后，根据收到的信息，控制L1和L2做出相应变化。当计算机向单片机分别发送单个字符“F”(即Forward首字母)、“B” (Backward)、“L” (Left)、“R” (Right)、“S” (Stop)，实验板上的L1、L2两个发光二极管分别全亮、全闪、L1单个闪、L2单个闪、全灭。例如，计算机发送单个字符“F”，单片机上的L1、L2全亮；计算机发送单个字符“B”，L1、L2全闪，如此类推。

1. 按照实验7-5的课程定义通信协议，完成计算机与单片机之间的通信。

在任务2)的基础上添加按键K2、K3、K4、L3、L4 与单片机的连线，编程完成：

当计算机向单片机发送遥信指令时，单片机按照约定的协议，上传相应按键的状态；

***比如*** 在计算机串口助手用十六进制方式发送 01 02 03 04，单片机发送 01 02 K3状态 K4状态。

当计算机向单片机发送遥控指令时，单片机按照约定的协议，执行指令，控制LED的状态。

***比如*** 在计算机串口助手用十六进制方式发送 03 02 01 00，

单片机收到指令后，熄灭L1，并向计算机发送 01 01 CC，表示执行成功；

如果发送03 01 FF，则表示执行失败。

说明：与任务2)只接收单个字符相比，任务3)增加了单片机侧数据串的接收，以及利用数据串中的长度段控制接收数据的个数，并利用接收到的功能码和数据段内容作出相应的响应。

1. **利用蓝牙模块实现单片机与手机蓝牙通信**

在任务2的基础上，如图8-5-1或图8-5-2，将单片机的串口引脚与蓝牙模块连接，改用手机上的蓝牙串口助手，利用蓝牙技术，完成手机与单片机的数据传送，将任务2的有线通信变为无线通信。

***步骤：***

1)先断开msp430G2553实验小板与计算机的USB数据线；

2)断开USB转串口模块与MSP430G2553单片机的连接，改用HC-05蓝牙模块的收/发信号与实验板上单片机的异步串口发/收信号对接，连线如图8-5-1。如果接线后，蓝牙模块上的灯不闪，说明蓝牙供电不足，改用图8-5-2接线，用USB转串口模块输出的5V给蓝牙模块供电，此时USB转串口模块只是一个提供电源的模块；

3)连接msp430G2553实验板与计算机的USB数据线；

4)打开手机蓝牙，运行手机蓝牙串口助手，并与HC-05蓝牙模块配对；

5)在单片机上分别运行任务2的1)、2)、3)的程序，实现单片机与手机的蓝牙通信。

***调试说明：***

单片机用HC-05蓝牙模块进行蓝牙通信时，可先按任务2的方法，调试完成单片机与计算机的异步串行通信，然后再接上蓝牙模块，利用蓝牙模块的透传功能，完成将有线的异步串行通信，改为无线的蓝牙通信。通信出现问题时，按照实验7的任务1-3和本次实验1-2介绍的调试方法，查找各环节可能存在的问题。

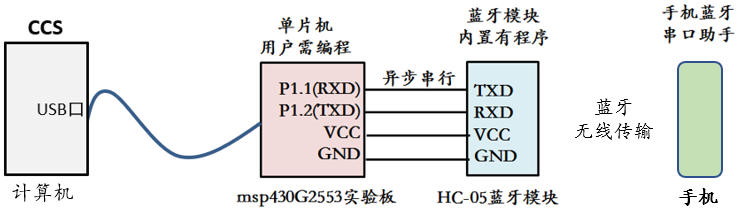


图8-5-1 单片机与手机蓝牙通信

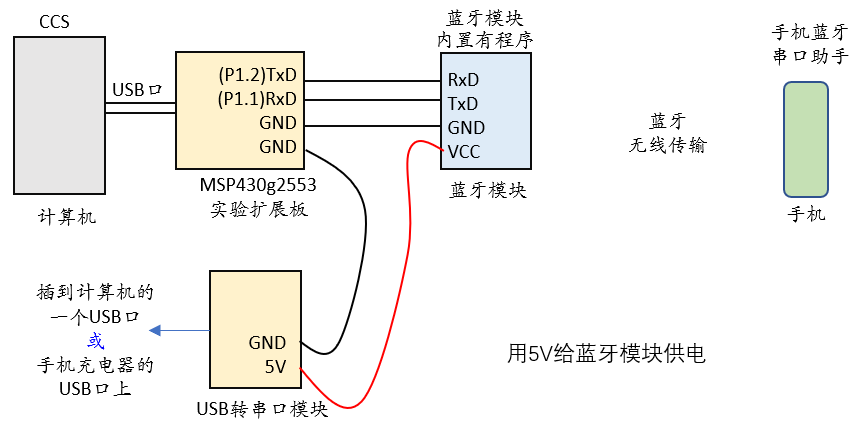


图8-5-2 单片机与手机蓝牙通信

1. **(选做) 利用蓝牙模块实现单片机与电脑蓝牙通信**

如果手机蓝牙无法与蓝牙模块通信，电脑上有蓝牙的话，如实验7的选做任务6，可以用电脑上的蓝牙代替手机上的蓝牙做任务3。接线如图8-6-1, 如果接线后，蓝牙模块上的灯不闪，说明蓝牙供电不足，改用图8-6-2接线，用USB转串口模块输出的5V给蓝牙模块供电，此时USB转串口模块只是一个提供电源的模块。步骤与上面的任务3相似，不同的只是用电脑蓝牙与HC-05蓝牙模块配对，用单片机串口程序控制蓝牙模块与电脑蓝牙通信。

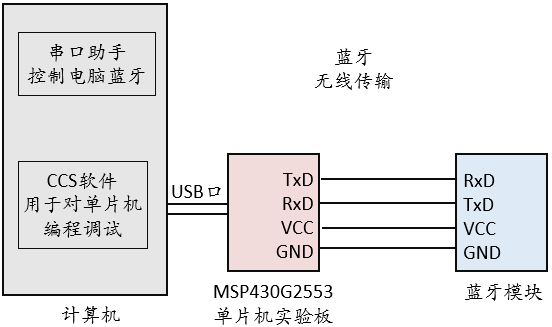


图8-6-1 单片机与电脑蓝牙通信

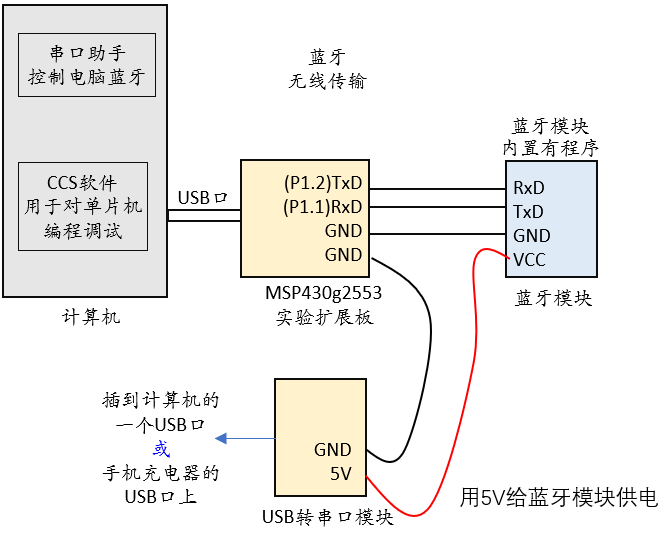


图8-6-2 单片机与电脑蓝牙通信

1. **(提高) 时钟信号对通信的影响**
2. 按图8-1接好单片机与计算机间的连线。假设SMCLK上电复位频率为1MHz（DCO上电复位时钟值），改写L8\_testSCI.c，选用SMCLK作为串口波特率时钟源，波特率为115200，运行L8\_testSCI.c。若计算机串口助手正确显示字符串，表示程序改写成功；若未正确收到字符串，表示单片机程序改写尚有问题。

若是实验板单片机上电复位SMCLK频率与1MHz偏差较大引起：

* **线下同学**可用实验5测量得到的上电SMCLK频率，或示波器测量得到的上电SMCLK频率，修改串口初始化程序中的设置，使单片机可按115200波特单片机率与计算机进行通信；
* **线上同学**没有示波器，可采用逐步尝试法，因已知上电复位SMCLK大约在1MHz，只是受实际环境温度、工作电压的大小影响，略有偏差。可分别假设SMCLK为1MHz、1.05MHz、1.1MHz，计算UCA0BR0 的值；并修改UCA0MCTL设置，尝试顺序用UCBRS\_0到UCBRS\_7这八个值一一去修改“UCA0MCTL=UCBRF\_0+UCBRS\_x”中的UCBRS\_x，直至单片机发送的字符串在计算机串口助手下，正确接收，说明当前的初始化115200波特率与计算机侧的115200波特率接近。

1. 在任务1)单片机能完成以115200波特率与计算机串行通信的基础上，掌握了“实验7-4 MIDI协议和弹奏控制”, 编程让单片机向计算机发送3字节的MIDI信息，控制计算机上自由钢琴的各琴键反复从低音向高音顺序弹奏。
2. **(提高) 中断方式控制串行通信的接收**

编程：采用中断方式接收、查询方式发送完成任务2中2)模拟小车的LED控制。(可参看讲义中断例子)

L8\_TestSCI.c 见下页

L8\_TestSCI.c程序：

**#include** "msp430.h"

**void** **UARTA0\_init**( );

**char** buffer[20],string[30]="Please input 7 characters:\r\n\0";

**unsigned** **char** j;

**int** **main** ( **void** )

{ WDTCTL = WDTPW + WDTHOLD; //关闭看门狗

//做实验时，如果有示波器

//可在P1.0、P1.4引脚输出时钟ACLK、SMCLK，以便观察时钟频率,修改波特率寄存器设置

// P1SEL |= BIT0+BIT4;

// P1SEL2 |=~( BIT0+BIT4);

// P1DIR |= BIT0+BIT4;

UARTA0\_init( ); //初始化串口

**while**(1)

{ j=0;

**while**(string[j]!='\0') //输出提示信息

{ **while**((IFG2&UCA0TXIFG)==0); //检测发送缓冲是否空

UCA0TXBUF=string[j]; //取一个数据发送

j++;

};

**for** (j=0; j<7; j++) //接收字符串

{ **while**((IFG2&UCA0RXIFG)==0); //检测接收缓冲器是否满

buffer[j]= UCA0RXBUF; //接收一个数据并保存

};

**for** (j=0;j<7;j++) //发送字符串

{ **while**((IFG2&UCA0TXIFG)==0); //检测发送缓冲是否空

UCA0TXBUF=buffer[j]; //取一个数据发送

};

};

}

**void** **UARTA0\_init**( )

{ UCA0CTL1|= UCSWRST; //置软件复位位swrst为1

P1SEL |= BIT1+BIT2; //置P1.1、P1.2为串行接口收、发引脚功能

P1SEL2 |= BIT1+BIT2;

//数据格式选用上电复位设置：无校验，8位数据，1个停止位，异步串行通信

UCA0CTL1|=UCSSEL0+UCRXEIE; //波特率时钟选择上电复位时的ACLK，32.768KHz，对错均收

UCA0BR0 =3; //波特率9600

UCA0BR1 = 0;

UCA0MCTL=UCBRF\_0+UCBRS\_3;

UCA0CTL1 &=~UCSWRST; //置软件复位位swrst为0，串口设置完毕

}