**目 录**

**[一、 系统功能、性能与使用说明 4](#_Toc14807)**

**[（一）系统功能 4](#_Toc11545)**

**[1. 时钟功能 4](#_Toc15277)**

**[2. 日历功能 4](#_Toc12755)**

**[3. 世界时钟功能 4](#_Toc6380)**

**[4. 闹钟功能 4](#_Toc14708)**

**[5. 秒表功能 5](#_Toc32531)**

**[6. 计时器功能 5](#_Toc32380)**

**[7. 串口时间校准功能 5](#_Toc32159)**

**[（二） 系统性能 5](#_Toc32138)**

**[（三）使用说明 6](#_Toc25981)**

**[二、 系统硬件设计 11](#_Toc6011)**

**[（一）硬件总体设计 11](#_Toc8496)**

**[（二）硬件选择 12](#_Toc19827)**

**[1.主控模块选择 12](#_Toc13110)**

**[2.显示模块选择 12](#_Toc11837)**

**[3.时钟模块选择 12](#_Toc13412)**

**[4.按键模块 12](#_Toc1963)**

**[5.串口校时模块 13](#_Toc14932)**

**[（三）硬件功能 13](#_Toc17362)**

**[1.主控模块 13](#_Toc3613)**

**[2. 显示模块 13](#_Toc22876)**

**[3. 时钟模块 15](#_Toc7658)**

**[4. 串口校时模块 17](#_Toc5435)**

**[5. 按键模块 18](#_Toc14633)**

**[三、 系统软件设计 18](#_Toc8767)**

**[（一）软件总体设计 19](#_Toc28045)**

**[（二）软件功能模块实现 20](#_Toc10335)**

**[1.显示功能实现 20](#_Toc20583)**

**[2.主界面实现 21](#_Toc712)**

**[3.菜单界面实现 23](#_Toc4197)**

**[4.秒表功能实现 25](#_Toc19652)**

**[5.闹铃功能实现 27](#_Toc11700)**

**[6.世界时钟功能实现 29](#_Toc14249)**

**[7.计时器功能实现 30](#_Toc1713)**

**[8. 串口校时功能实现 32](#_Toc14676)**

**[四、系统整机测试 33](#_Toc20200)**

**[（一） 显示模块测试 33](#_Toc8100)**

**[（二）功能模块测试 37](#_Toc27450)**

**[1.秒表功能 37](#_Toc14468)**

**[2.闹铃功能 38](#_Toc17455)**

**[3.世界时钟功能 39](#_Toc31622)**

**[4.计时器功能 40](#_Toc26542)**

**[5.串口校时测试 41](#_Toc18719)**

**[五、分析总结与体会 44](#_Toc23996)**

1. **系统功能、性能与使用说明**

**（一）系统功能**

本智能钟控系统共实现了以下7个功能：时钟功能、日历功能、世界时钟功能、闹钟功能、秒表功能、计时器功能和串口时间校准功能，下面对这些功能一一介绍。

1. **时钟功能**

* 在OLED显示屏上显示出当前时间，时间格式为----XX : XX : XX；
* 自动走时，并且可以按照日常生活中的时间格式正确地进位；
* 12/24小时可通过按键切换，12小时制时显示模式会在时间的下一行显示AM/PM。

1. **日历功能**

* 在OLED显示屏上显示出当前日期，日期格式为----XXXX年 XX月XX日 周X；
* 星期自动校正，输入年月日后能够自动地校准当前日期是星期几；
* 闰年闰月自动校准。

1. **世界时钟功能**

* 在OLED屏显示出当前时区地点，默认为北京时间，在时间前面显示“Beijing:”;
* 在功能界面同时显示出三个时区的时间，分别为北京时间、华盛顿时间和伦敦时间；

1. **闹钟功能**

* 在OLED显示屏显示出闹铃时间，闹铃格式为----闹铃：周X XX时XX分；
* 时间到达设定的闹铃时间后，闹钟响铃一分钟，期间可以通过按键来取消铃响；
* 闹铃时间可通过按键来手动设置，且也可以通过按键控制闹铃的打开或关闭。

1. **秒表功能**

* 在OLED显示屏上显示出时间计次，时间计次格式为----XX : XX : XX，分辨率为0.01s；
* 按照0.01s的分辨率自动走时；
* 通过按键控制时间计次，共可计次4次。

1. **计时器功能**

* 在OLED显示屏上显示出倒计时的时间，时间格式为----XX : XX : XX，分辨率为0.01s；
* 通过按键手动设置倒计时时间，可设置的最小时间单位为秒；
* 通过按键控制倒计时的开始，并在倒计时结束时响铃5s。

1. **串口时间校准功能**

* 通过电脑终端发送时间自动校准显示屏上的时间，发送的数据格式为----XXXX（前两位代表时，后两位代表分）；
* 通过电脑终端发送日期自动校准显示屏上的日期，发送的数据分两种格式：第一种为带年的日期校准，格式为----XXXXXXXX（前四位代表年，后四位代表月和日）；第二种为不带年的日期校准，格式为----XXXX（前两位代表月，后两位代表日）。星期可以自动根据输入的日期来校准。

1. **系统性能**

**硬件条件：**系统以MSP432为主控板，基于实时时钟芯片DS1302构建数字时钟与万年历，并通过OLED实时显示，通过串口有线通信实现授时功能。因此硬件条件是MSP432主控板、DS1302时钟芯片、OLED显示屏。

**供电条件：**通过USB接口将设备与PC端相连供电。系统初始上电状态，按下home键，系统显示正常的万年历、闹铃等；若显示屏处于无序状，可通过主控板的复位键恢复，再按home键。

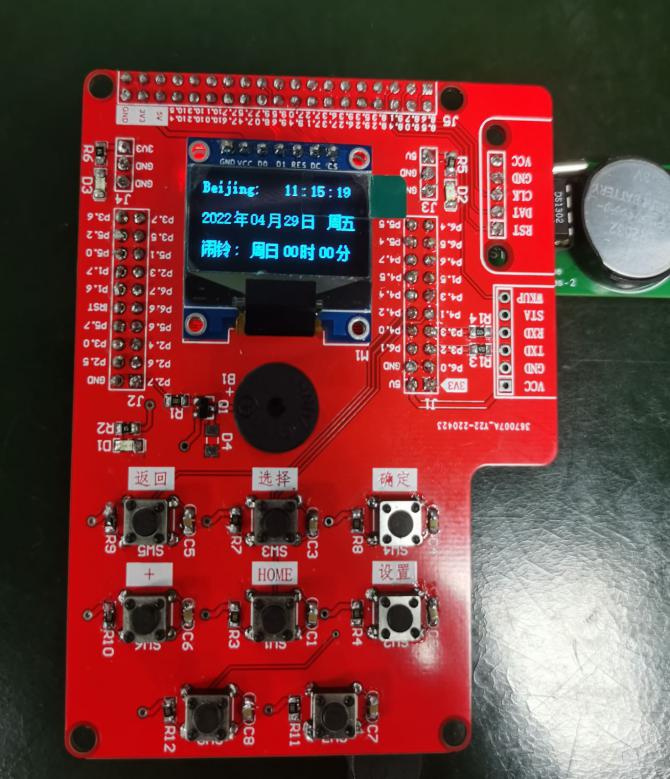
**响应时长：**响应时长（RT）是从请求发出，经过系统处理到系统返回结果，整个过程的时间。响应时长直接影响用户的使用感受。因此，在性能测试中响应时间是一个非常重要的指标。本系统存在的从按键按下到OLED显示屏显示这一响应时间。按键按下表示系统输入从低电平到高电平，此后经过主控板运行到OLED屏的显示，系统的响应时长在微秒级，几乎是瞬时完成。

**并发用户数：**系统能同时处理的请求数，该指标也反映了系统的处理能力。在本系统中，系统不能同时按下多个键，一次只能按一个键有效。如同时按下两个键，当这两个按键只有一个在此界面有效时，显示屏按此键发生改变；若这两个按键在此界面都有对应的操作，显示屏只发生一次闪烁，并无实际显示变化。

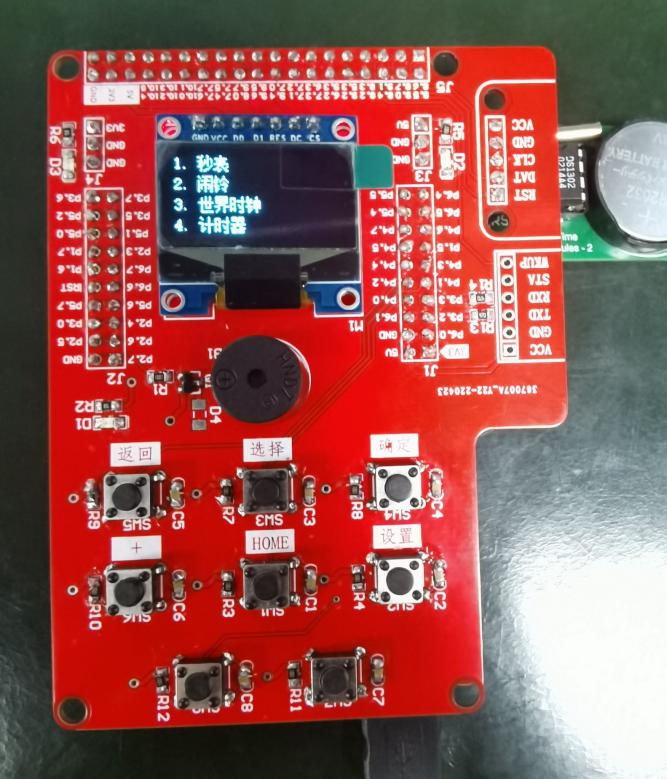
**应用场景：**在干燥环境下使用，禁止将产品浸入水中。

**（三）使用说明**

1. 接上电源后，按下Home键（P4.0），主界面显示，如下所示：

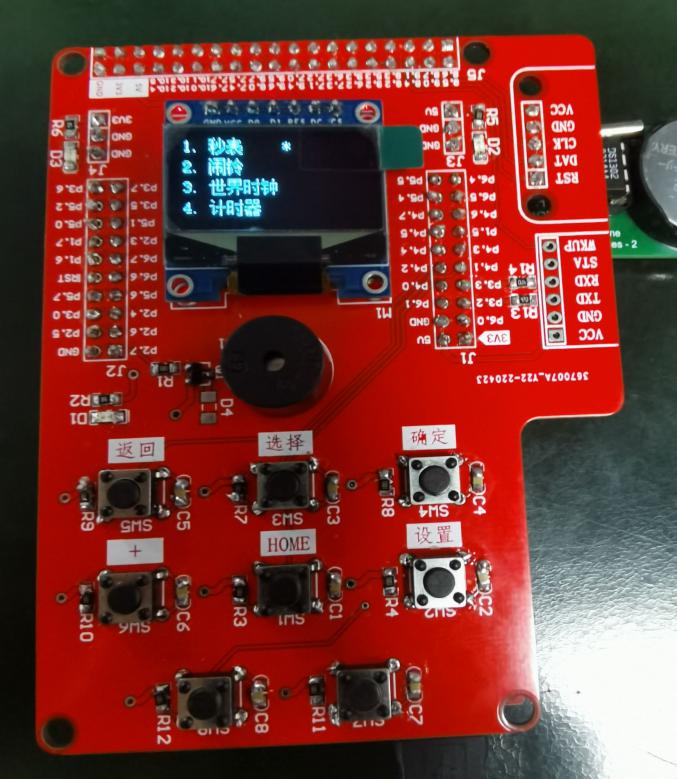


1. 按下设置键（P4.4），进入菜单界面，如下所示：

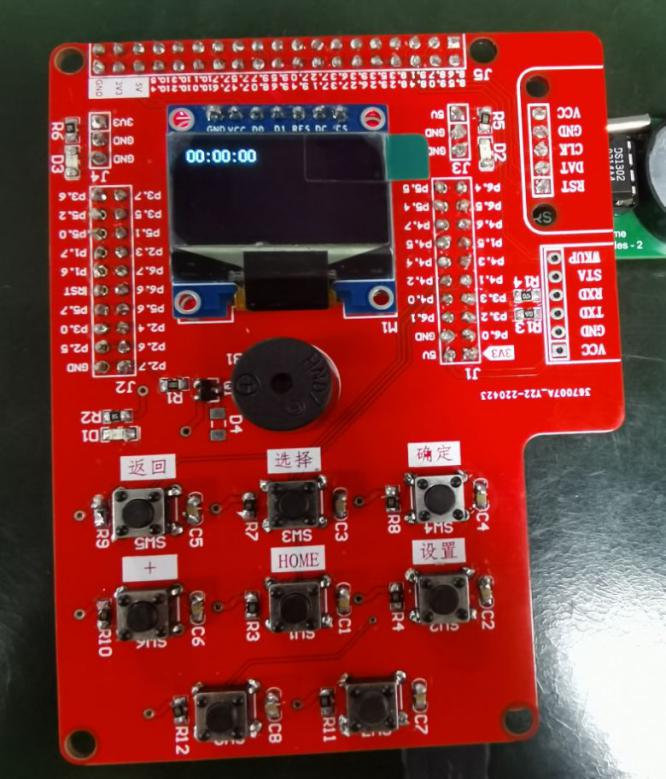


进入菜单界面后，若要返回主界面，需按下Home键；

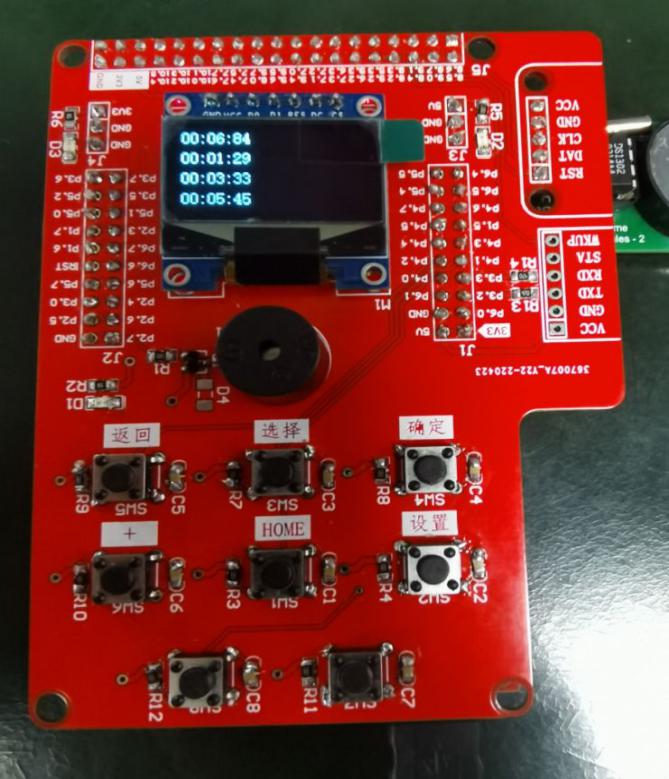
1. 进入菜单界面后，继续按下设置键，选择光标“\*”在四个功能后移动，以此来表示对不同功能的选择，如下所示：



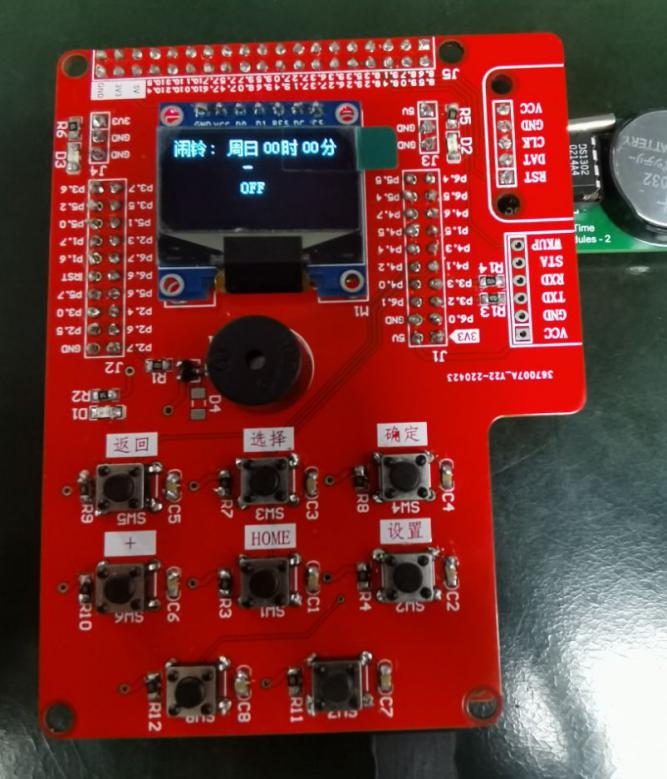
1. 按下选择键（P4.1），则可进入相应的功能界面；进入每个功能界面之后，按下返回键（P4.2）都可返回至菜单界面；
2. 进入秒表功能界面，如下所示：



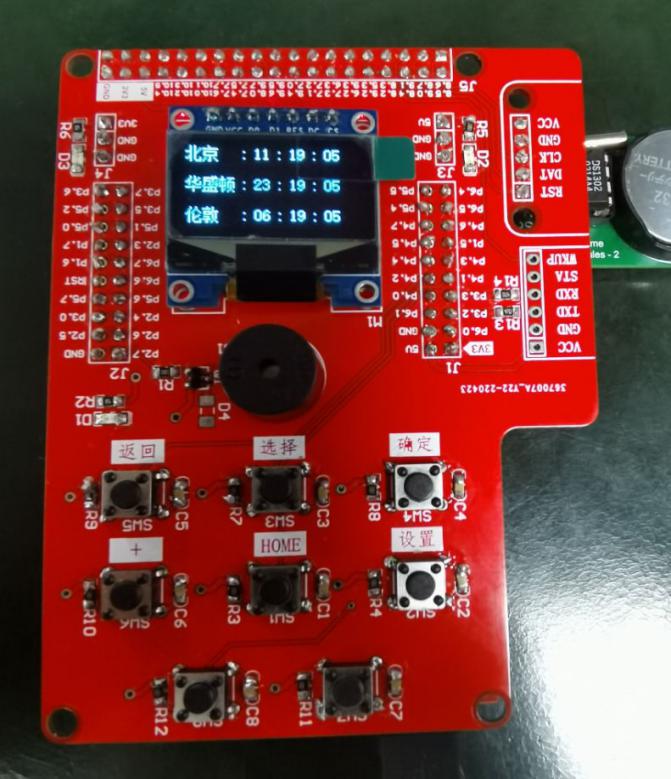
按下确定键（P4.5），开始走时，再次按下确定键，记录一次当前的时间，一 共可记录四次，如下所示：



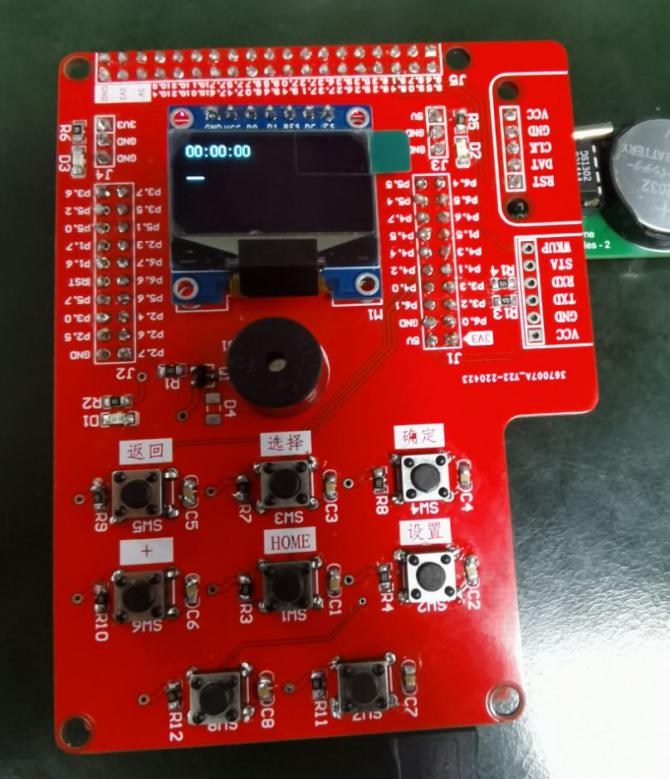
1. 进入闹铃功能界面，按下确定键可设置闹铃的打开和关闭，分别显示为“ON”和“OFF”；按下“+”键（P4.6）对闹铃的时间进行更改，按下备用1键（P4.7）可以在星期、时和分之间转换，实现时间不同部分的更改，闹铃界面如下所示：



1. 进入世界时钟界面，显示三个时区的时间，如下所示：



1. 进入计时器界面，显示如下所示：



按下“+”键，更改倒计时的时间；按下备用1键，更改需要改动的时间（分 或者秒）；按下确定键，开始倒计时，倒计时完毕后响铃。

1. 串口时间校准，需要用USB线连接电脑，在键盘输入时间数据。按下P1.1键，在CCS里的终端输入时间数据，OLED显示屏上的时间就会更正为输入的时间；按下P1.4键，在CCS里的终端输入日期数据，OLED显示屏上的日期就会更正为输入的日期。
2. 按下备用2键（P4.3），OLED显示屏上的时间会在12/24小时制之间切换，12小时制的界面如下所示：

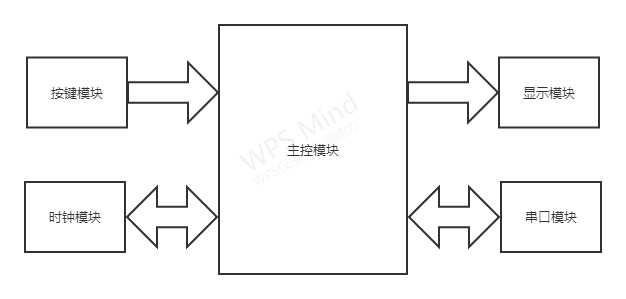


1. **系统硬件设计**

**（一）硬件总体设计**

为实现系统功能，硬件上应当包括至少四个模块：主控模块、显示模块、时钟模块、按键模块；大体上系统的工作流程为：输入初始数据进主控板，主控板运行相应程序，时钟模块在接收到主控板传出的数据后会设置一个内部初始时间并开始走时，在时钟模块走时的同时传输数据回到主控板，主控板接收数据并经过程序处理后控制显示模块显示出当前时间。并且通过按键模块选择不同的功能后也会运行不同程序并在显示模块上显示不同的界面。在实现这些功能之后还可以加入串口校时模块，主控模块与校时模块通过UART通信。

硬件结构框图如下：



**（二）硬件选择**

**1.主控模块选择**

根据现有条件以及系统所需实现功能综合考虑，选择使用MSP432P401R单片机作为系统的主控板。

**2.显示模块选择**

目前市面上主流的显示屏有两种：LCD显示屏和OLED显示屏，LCD的发光原理主要是依靠背光层，通常由大量的LED背光灯造成，彩色通过背光层上再加上一层颜色的薄膜，最后通过液晶层，由改变两级电压的大小来控制液晶分子的排布，最后调整红绿蓝的配比；OLED不需要LCD屏幕那样的背光层，也不需要控制出光量的液晶层，OLED可以自发光，所以OLED就像一个有着无数个小的彩色灯泡组合的屏幕。由于在显示原理上的差异导致LCD屏幕比OLED更厚，重量上也更重，在对比度上，得益于OLED自发光的特性，对比度比LCD更出色，因为背光层的存在使LCD很难表现出纯正的黑色；同样是自发光的特性，OLED耗电量更低，能够独立控制发光和关闭；综合考虑我们选择市面上常用的一种OLED显示屏——0.96寸OLED显示屏作为本智能钟控系统的显示模块。

**3.时钟模块选择**

在进行市场调研以及多方查询资料后，发现DS1302实时时钟芯片广泛应用于电话、传真、便携式仪器等产品领域，因此我们选择其作为系统的时钟模块。

**4.按键模块**

市面上的按键种类繁多，我们选择其中一种四脚轻触开关作按键模块。

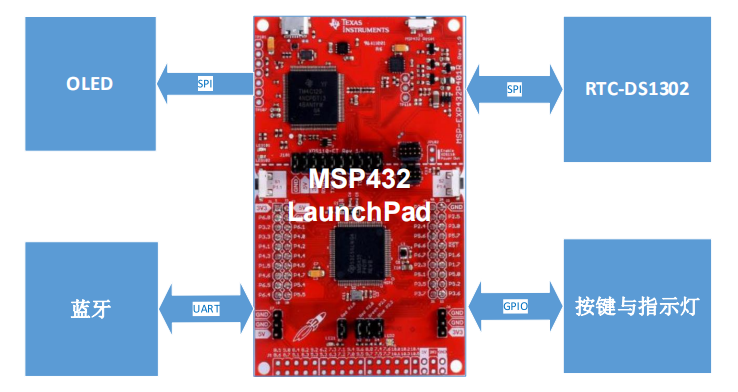
**5.串口校时模块**

MSP432 单片机内置了 UART 功能，它可以将外部设备串行数据转换为并行数据接收。对于UART通信的硬件实现也很简单，只需将UART0通过USB线与电脑端连接即可。因此，我们选择通过UART来实现串口校时功能。

**（三）硬件功能**

**1.主控模块**

系统以 MSP432 为主控器，基于实时时钟芯片 DS1302 构建数字时钟与万年历，并通过OLED 实时显示，通过UART实现远程授时功能。



1. **显示模块**

（1）功能介绍

本系统采用的显示屏是中景园电子生产的0.96寸OLED显示屏。其显示原理和普通的OLED显示屏没有很大差异，OLED（Organic Light-Emitting Diode）即有机发光二极管。OLED显示技术无需背光灯，采用非常薄的有机材料涂层和玻璃基板(或柔性有机基板)，当有电流通过时，这些有机材料就会发光。其分辨率为128x64,每一个像素都是一个LED。

其产品参数如下表；

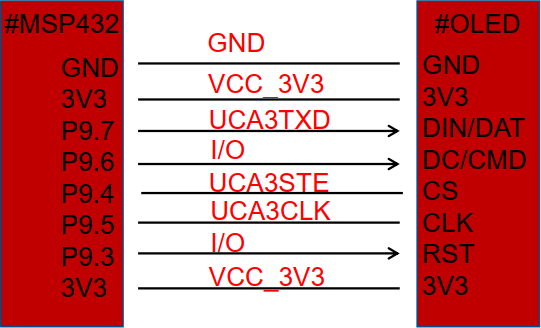
|  |  |  |
| --- | --- | --- |
| 尺寸 | 0.96 英寸 |  |
| 分辨率 | 128x64 |
| 接口类型 | SPI 接口 |
| 工作电压 | 3.3V |
| 驱动芯片 | SSD1306 |
| 显示颜色 | 蓝色 |
| 工作温度 | -40~+70℃ |

推荐使用的接口方式为MSP432的SPI接口控制0.96OLED显示屏，下面简略介绍SPI通信。

SPI（Serial Peripheral Interface）为串行外设接口的简称，它是一种同步全双工通信协议，其硬件功能很强，与 SPI 有关的软件相当简单，使 CPU 有更多的时间处理其他事务。MSP432单片机的eUSCI\_A和eUSCI\_B模块都支持 SPI 通信模式。SPI总线上可以连接多个可作为主机的MCU和装有SPI接口的输入/输出设备，如液晶驱动、片外ADC等外设。SPI通信模块通过3线（SOMI、SIMO、CLK）或者4线（SOMI、SIMO、CLK 及STE）同外界进行通信。其中SOMI、SIMO、CLK在主机模式和从机模式下存在差别，如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **引脚** | **含义** | **主机模式** | **从机模式** |
| SOMI | 从出主入 | 数据输入引脚 | 数据输出引脚 |
| SIMO | 从入主出 | 数据输出引脚 | 数据输入引脚 |
| CLK | UART 时钟 | 输出时钟 | 输入时钟 |
| STE | 从机模式发送/接收控制 | 从机选择 | 片选信号 |

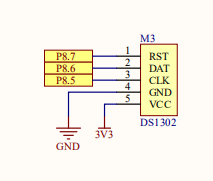
（2）接口方式

MSP432上的多组IO口均可设置为SPI接口，但是大部IO口被RSLK 小车其他组件占据（参考 RSLK 基础版原理图），推荐使用 P9.4~P9.5作为OLED的通信接口，连接方式如图：

1. **时钟模块**

（1）功能介绍

DS1302 是涓流充电时钟芯片，内含有一个实时时钟/日历和 31 字节静态 RAM，通过简单的串行接口与单片机进行通信。实时时钟日历电路提供秒、分、时、日、日期、月、年的信息，每月的天数和闰年的天数可自动调整，时钟操作可通过 AM/PM 指示决定采用24 或 12 小时格式。DS1302 与单片机之间能简单地采用同步串行的方式进行通信，仅需用到三个口线: (1)RES(复位)，(2)I/O (数据线)，(3)SCLK(串行时钟)。时钟/RAM 的读/写数据以一个字节或多达 31 个字节的字符组方式通信。它广泛应用于电话、传真、便携式仪器以及电池供电的仪器仪表等产品领域。



其内部结构为：

（1）SCLK：串行时钟输入端，控制数据输入与输出。  
 （2）I/O：双向输入线。  
 （3）CE：使能端，CE为高时允许DS1302读写数据，CE端为低时DS1302数据不可读写。  
 （4）X1与X2：外接32.768的圆形晶振，给时钟芯片提供晶振频率。

对DS1302的操作就是对其内部寄存器的操作，DS1302内部共有12个寄存器，其中有：7 个寄存器与日历、时钟相关，存放的数据位为 BCD 码形式。

此外，DS1302 还有年份寄存器、控制寄存器、充电寄存器、时钟突发寄存器及与 RAM 相关的寄存器等。如下图所示，时钟日历包含在 7 个读/写寄存器内，读/写寄存器中的数据是BCD 码。

秒寄存器的BIT7定义为时间暂停位，当BIT1为1时，时钟振荡器停止工作，DS1302进入低功耗模式，电源消耗小于100微安，当 BIT1 为 0 时，时钟振荡器启动，DS1302 正常工作。

小时寄存器的BIT7定义为12或24小时工作模式选择位，当BIT7为高时，为12小时工作模式，此时BIT5为AM/PM位，低电平标示AM，高电平标示PM，在24小时模式下，BIT5为第二个10小时位标示(20~23 时)。

写保护寄存器的 BIT7：WP 是写保护位，工作时，除WP外的其他位都置为0，对时钟/日历寄存器或RAM进行写操作之前，WP必须为0，当WP为高电平的时候，不能对任何时钟/日历寄存器或 RAM 进行写操作。

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| READ | WRITE | BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 | RANGE |
| 81h | 80h | CH | 10 Seconds | | | Second | | | | 00-59 |
| 83h | 82h |  | 10 Minute | | | Minute | | | | 00-59 |
| 85h | 84h | 12/ | 0 | 10 | Hour | Hour | | | | 1-12/  0-23 |
|  |
| 87h | 86h | 0 | 0 | 10 Date | | Date | | | | 1-31 |
| 89h | 88h | 0 | 0 | 0 | 10  Month | Month | | | | 1-12 |
| 8Bh | 8Ah | 0 | 0 | 0 | 0 | 0 | Day | | | 1-7 |
| 8Dh | 8Ch | 10year | | | |  |  |  |  | 00-99 |
| 8Fh | 8Eh | WP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - |
| 91h | 90h | TCS | TCS | TCS | TCS | TCS | DS | RS | RS | - |

此外，DS1302还有突发模式（burst mode或称多字节传输模式）可以指定任何的时钟/日历或者 RAM 寄存器为突发模式，此处不再赘述。

（2）接口方式

DS1302时钟芯片有五个引脚，其中我们将GND接地，VCC接3.3V，CLK接P8.5，DAT接P8.6，RST接P8.7。

1. **串口校时模块**

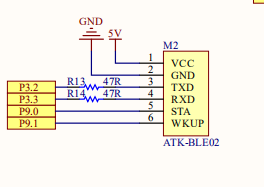
（1）功能介绍

客户端通过UART通信与主控板连接。下面简略介绍UART通信：

UART即异步串行通信，可设置成全双工异步通信方式，与 PC（个人计算机）等通信； 或设置成半双工同步模式，与其他外设通信，如 ADC 或 DAC。MSP432 单片机内置了 UART 功能，它的作用是将外部设备串行数据转换为并行数据接收；将内部并行数据转换为串行数据发送。在通用异步收发模式下，eUSCI\_Ax 模块通过两个外部收发引脚 UCAxRXD 和UCAxTXD 把 MSP432单片机与外界连接起来。当寄存器 UCAxCLT0 的UCSYNC 控制位被清零，UCMODEx控制位被配置为00时，eUSCI\_A 模块被配置为UART异步通信模式。在UART模式下，eUSCI\_A 模块由串行数据接收逻辑波特率发生器和串行数据发送逻辑3个部分组成。串行数据接收逻辑用于接收串行数据，包含接收移位寄存器、接收缓冲寄存器和接收状态机以及接收标志位设置逻辑。波特率发生器用于产生接收和发送的时钟信号， 其参考时钟可以来源于 ACLK或 SMCLK，也可以来自于外部时钟信号输入UCLK，通过整数或小数分频得到特定的数据传输波特率。传输数据发送逻辑用于发送串行数据，包含发送移位寄存器、发送缓冲寄存器和发送状态机以及发送标志位设置逻辑。在 UART 模式下，eUSCI\_A异步以一定速率向另一个设备发送和接收字符，每个字符的传输时钟基于软件对波特率的设定，发送和接收操作使用相同的波特率频率。

（2）接口方式

MSP432共有4组IO口可以设置为UART,其中UART0被下载器占据，UART1IO口被三色灯占据，UART3IO口被OLED占据，这里使用UART0与电脑相连接，在电脑控制台上输入不同时间可以改变数字时钟的时间。



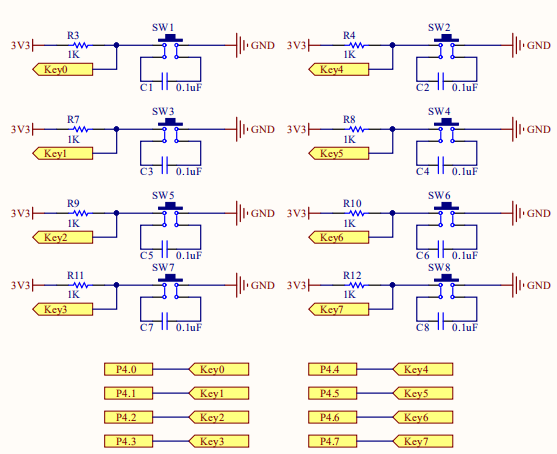
1. **按键模块**

（1）功能介绍

按键接入电路后，通过调用按键扫描程序和中断函数判断哪一个按键被按下，并在主函数执行相应的程序以达到所需实现的功能。

（2）接口方式

根据系统需求选择P4.0~4.7作为外部按键接口。设计‘Home’键、‘设置’键、‘选择’键、‘确定’键、‘返回’键、‘+’键与两个备用键，接口方式如下图所示：

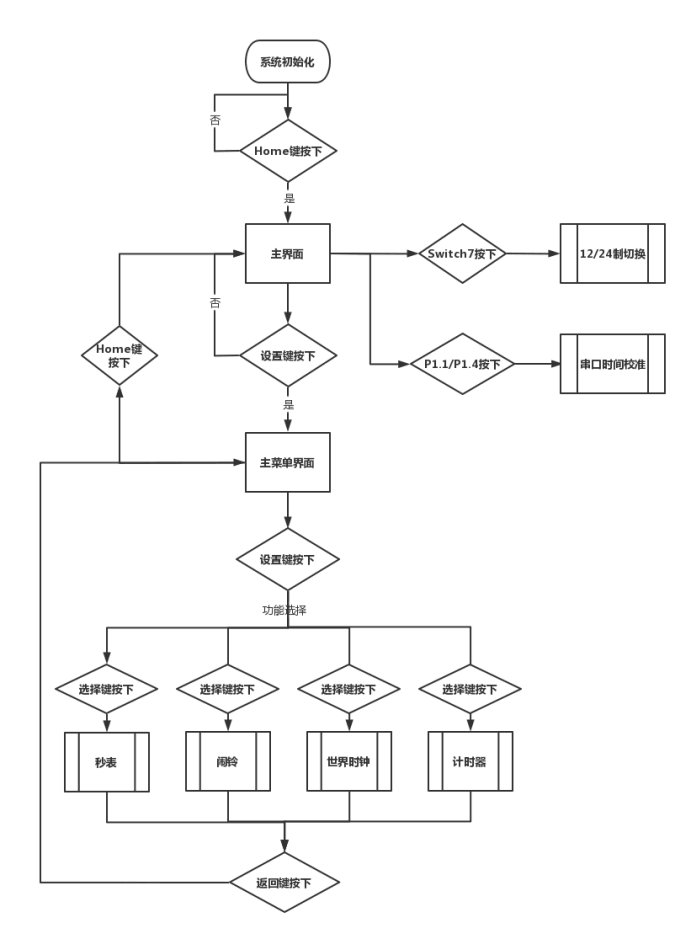


其中key0-key7分别对应home键、选择键、返回键、switch7、设置键、确定键、‘+’键、swith8。

1. **系统软件设计**

**（一）软件总体设计**

为了使数字时钟能够达到相应的功能，需要用到的模块有：单片机、时钟芯片DS1302模块、OLED显示模块、串口模块、按键模块。因此软件系统应由以下几个部分组成：主界面、主菜单界面、模块子程序、按键判断程序。具体的流程图如图所示：



根据软件总体流程图指导，自顶向下的完成系统程序框架构建。将所有使用到的外部引用、全局变量声明写在程序最上方，将按键中断函数写在main函数外面，按键判断程序中加入变量以达到按键按下时变量做出改变。根据按键判断函数中的变量执行相应程序。

**（二）软件功能模块实现**

**1.显示功能实现**

基于中景园电子提供的oled.h与oled.c构建0.96寸OLED显示界面,oled.c库函数里有数字显示函数OLED\_ShowNum,字符显示函数OLED\_ShowString以及汉字显示函数OLED\_ShowChinese,分别介绍三个函数的具体用法:

**OLED\_ShowNum(u8 x,u8 y,u32 num,u8 len,u8 sizey)**----显示数字

x,y :起点坐标；num:要显示的数字；len :数字的位数；sizey:字体大小;

其作用是在屏幕的x列y行显示一个位数为len的数字,其字体大小为sizey(0.96寸OLED 尺寸为128\*64)。

**OLED\_ShowString(u8 x,u8 y,u8 \*chr,u8 sizey)**----显示一个字符串

x,y :起点坐标；chr:要显示的字符串；sizey:字体大小;

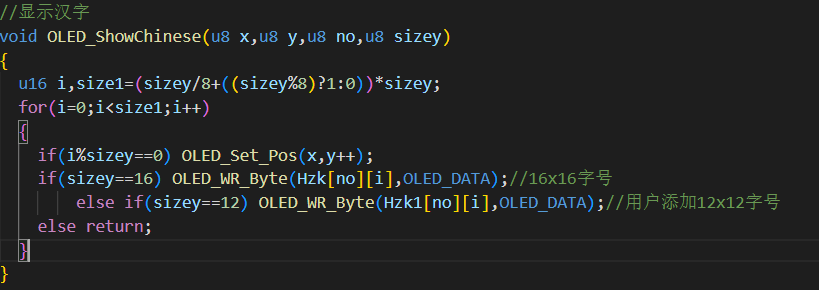
其作用是在屏幕的x列y行显示一个字符串chr,其字体大小为sizey。

**OLED\_ShowChinese(u8 x,u8 y,u8 no,u8 sizey)**----显示汉字

x,y :起点坐标；no:汉字字库数组号；sizey:字体大小；

其作用是在屏幕的x列y行显示汉字字库中数组号为no的汉字,其字体大小为sizey。

但为了使屏幕更加美观,我们没有采用oeld.c中使用的8\*8和16\*16号字体,而是另外在取模软件PCtoLCD2002.exe上将12\*12号字体的数字、字符以及汉字字库数组取了一遍,并将其加入到了oledfont.h(oled.c字库数组存储的外部文件)中。并将三个显示函数中分别加上了12\*12号字体,以汉字显示函数为例,改动后的汉字显示函数如下:

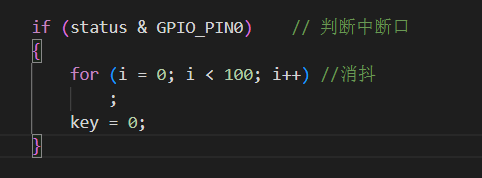
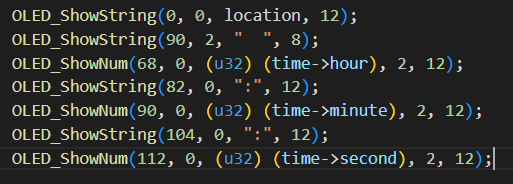


加入了12\*12号字体后,每次都只需调用显示函数即可在屏幕上得到所需的显示效果。

**2.主界面实现**

**显示功能：**

按下home键后中断函数判断中断口为P4.0,将中断函数中的变量key赋值为0,main函数中if(key==0)语句条件成立,进入if语句执行相应的显示程序,显示出主界面。具体的中断函数判断以及主界面显示函数如下:



Ps:由于篇幅原因这里只展示了第一行所显示北京时间的代码,具体的代码在附件中的main.c文件中的Home函数中。

出现在程序中的time是我们定义的一个结构体，类型为指针变量，在给它赋初值后，调用ds1302.h中的DS1302\_write\_time(time)函数，将其写入到时钟芯片DS1302中，时钟芯片的初始时间数据就会变成time的初值，之后在while循环中再调用ds1302.h中的DS1302\_read\_time(time)函数，时钟芯片的时间数据就会实时更新，再调用数字显示函数将当前时间显示到屏幕上。

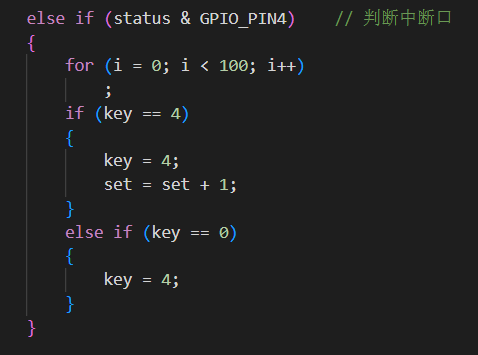
**12/24小时制切换功能：**

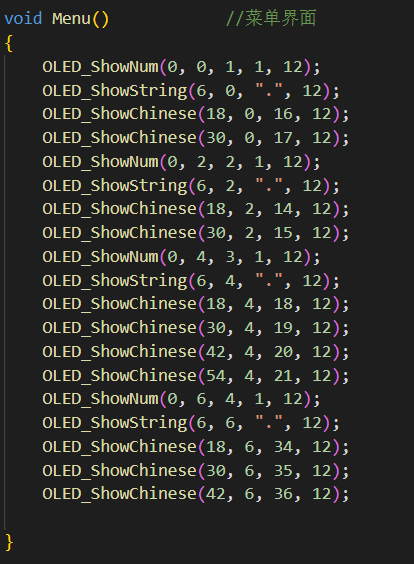
在主界面时，按下按键switch8中断函数判断中断口为P4.3，变量convert++，convert初值为0，当convert%2==0时系统为24小时制，主界面显示24小时制时间；当convert%2==1时系统为12小时制，主界面显示12小时制时间，并在时间下方多出AM/PM的字样以表明当前为上午/下午。12/24小时制的区别只体现在时针以及有无AM/PM。其中12小时制的显示代码如下：



**3.菜单界面实现**

按下设置键后中断函数判断中断口为P4.4,将中断函数中的变量key赋值为4，main函数中if(key==0)语句条件成立,进入if语句执行相应的显示程序,显示出主菜单界面。进入主菜单界面后,再次按下设置键执行中断函数使变量set=set+1,同时主菜单界面会在秒表后多出一个‘\*’以表示选中秒表功能;同理,再次按下设置键set++,主菜单界面上的‘\*’出现在闹铃后以表示选中闹铃功能。同时还在中断函数中设置了一个限定条件使只有在进入主菜单界面后才能切换具体子功能。中断函数以及主菜单界面显示程序如下:





在key==4时会显示菜单界面,子功能选中显示程序在main.c文件中。

假设已经选中并进入了某一个子功能，按下返回键后中断函数判断中断口为P4.2，变量back++，back初值为0，back！=0时会退出子功能返回到主菜单界面，具体的程序如下：

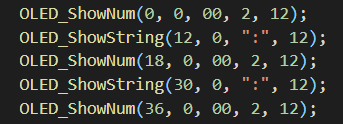
这里是进入了闹铃功能，按下返回键back++，满足back!=0的条件，使得key=4,即返回到了主菜单界面，并且set=2,仍然选中闹铃功能。

**4.秒表功能实现**

根据上述系统功能中对秒表功能的描述，此功能需分为三个模块：显示模块，走时模块（分辨率0.01s）和计次模块（共可计次四次），下面分别对三个模块进行设计。

**显示模块：**

按照秒表显示格式：XX : XX : XX，需要显示数字和字符“:”。根据上述说明的OLED显示屏原理，直接调用库函数编写显示格式：



以上代码为在第一行显示“00:00:00”，随后的时间计次显示与此原理一致，只需调整函数中的参数使之显示在不同行即可。

**走时模块：**

走时模块可以用软件和硬件两种方法来实现。 软件走时一般是根据所需要的时间常数来设计一个延时子程序来实现。 硬件走时一般是利用专门的定时器件作为主要实现器件，在简单的软件控制下，产生延时时间。由于本模块是在已有系统上做软件开发，且延时时间较小而重复次数又有限，我们采用了编写延时子程序的方法来实现走时。

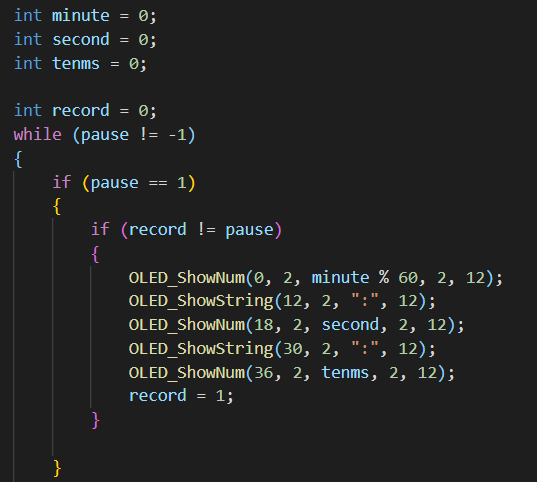
在初始化时钟后，通过调用 Clock\_Delay1ms() 或Clock\_Delay1us() 可以产生准确的延时。本模块的时间走时以0.01s，即10ms为单位，但是不能直接使用 Clock\_Delay1ms(10) 来产生10ms的延时，因为其他程序也要花费一定的时间。经过对延时程序之外程序的源码分析与实际测试，延时程序中设置大致7.2ms的延时时走时最为准确。同时，还需考虑单位间的进位问题。0.01s位在满99后，需要在下一个延时过后变为00且使1s位加一；同理，1s位在满59后也要变为0且向1min位进一。延时和进位程序如下：



接下来只需把上述程序放入条件循环中，即可实现自动走时功能。

**计次模块：**

计次模块的基本要求是按下按键记录一次当前的时间，我们通过按键中断来实现此模块，因为这样在计次时走时模块也能运行。计次共需完成四次，为此功能我们加入一个参数 pause来判断计次的数量，在每按一次计次按键后使pause加一，使之可以进入下一次计时。当前时间的记录我们引入一个参数record来实现，下面是第一次计次的程序：

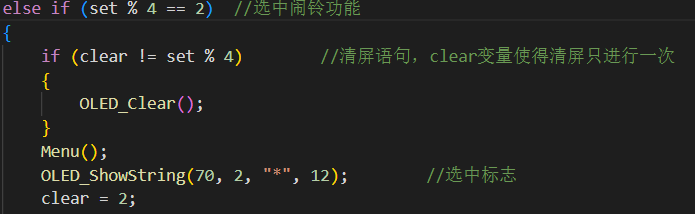


剩下3次计次之一致，只需改动相关参数即可。

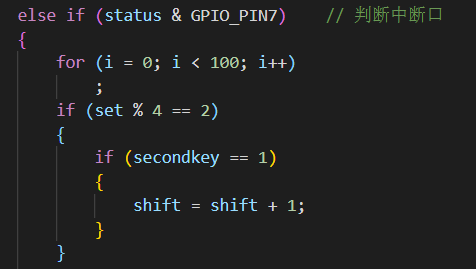
**5.闹铃功能实现**

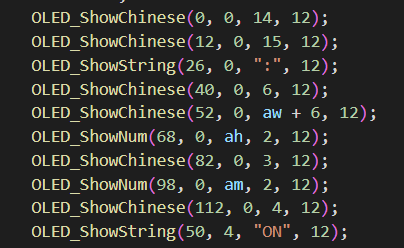
**显示模块:**

在主菜单中选中闹铃功能，按下选择键中断函数判断中断口为P4.1,将中断函数中的变量secondkey赋值为1，while(secondkey==1)语句条件成立，进入while循环，先清屏，再显示闹铃界面。菜单中选中闹铃功能程序如下:

**设定闹铃时间：**

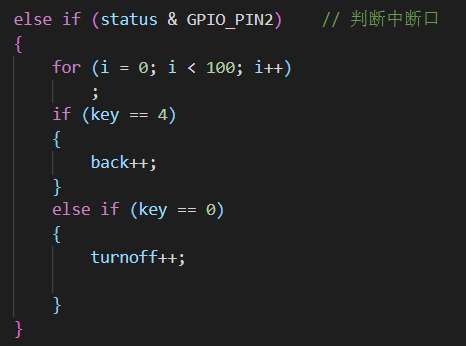
进入闹铃界面后,我们设置了三个变量aw，ah，am分别为闹铃周位,闹铃时位以及闹铃分位。按下switch7时中断函数判断中断口为P4.7，每按一次switch7中断函数中的变量shift加一；按下‘+’键后中断函数判断中断口为P4.6，每按一次中断函数中的变量plus加一；当shift%3的结果分别为0、1、2时，控制当前能改变的变量分别为aw、ah、am，且每一次切换变量时plus都会重新从0开始改变。令aw、ah、am分别为plus%7、plus%24、plus%60就能达到通过控制switch7和‘+’键来实现更改闹铃时间的功能。并且在确定键按下时中断函数判断中断口为P4.5，中断函数中的变量sw (闹钟开关)++，sw%2==0时表示闹钟关闭并在闹铃功能界面显示‘OFF’，sw%2==1时表示闹钟打开并在闹铃功能界面中显示‘ON’。我们还设置了限定条件使得只有在sw%==0也就是闹钟关闭使才能够调节闹铃时。中断函数以及显示界面代码如下，更改闹铃时间程序在main.c文件中。

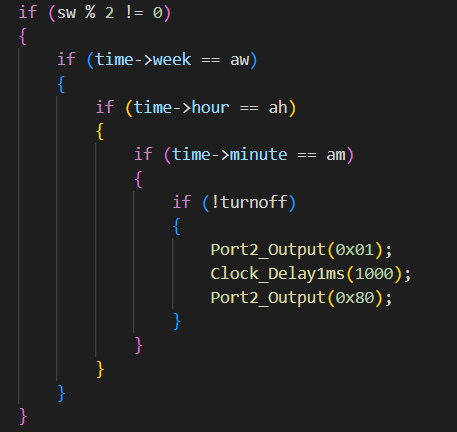




**响铃功能：**

在while循环中的时间数据会随着时钟芯片不断变化，当当前时间与设定的闹铃时间一致后，蜂鸣器输出低电平开始工作一分钟，在蜂鸣器响时若按下返回键，中断函数判断中断口为P4.2使得变量turnoff++，turnoff初值为0，turnoff变为1时蜂鸣器输出高电平关闭。具体的程序为：



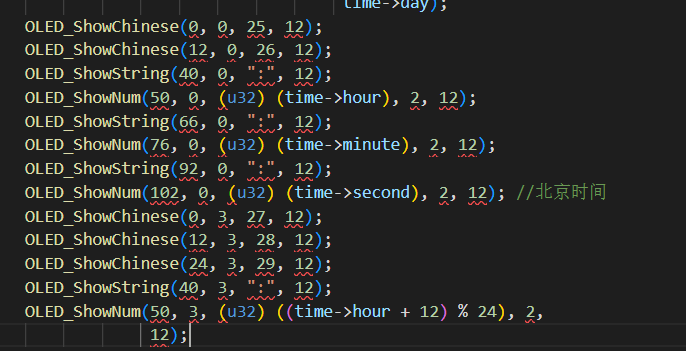


将显示、设定、响铃三个模块连接在一起，就能够实现闹铃功能。

**6.世界时钟功能实现**

**显示模块：**

在主菜单中选中世界时钟功能（set==3），按下选择键进入世界时钟功能，清屏一次后显示出三个时区的时间，以显示北京时间为例程序如下：



这里的time与主界面中的time一致，同样的依次调用DS1302\_write\_time(time)以及DS1302\_read\_time(time)函数可以实时更新时间；想要显示华盛顿时间与伦敦时间只需将它们与北京时间的时差计算出来，在显示函数中更改time->hour即可。

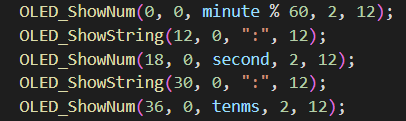
因为世界时钟功能只是简单的显示三个时区的时间，所以只需要一个显示模块即可实现全部功能。

**7.计时器功能实现**

根据上述系统功能中对计时器功能的描述，此功能需分为四个模块：显示模块，走时模块（分辨率0.01s），时间设置模块和响铃模块。下面分别介绍一下四个模块的实现：

**显示模块：**

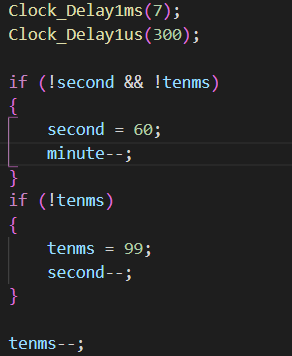
由于为倒计时功能，因此只需显示一行即可，时间格式为：XX:XX:XX，调用OLED库函数在显示屏上显示：



在后续程序中，用上面的minute、second和tenms三个变量接收当时的时间，并放在循环里走时，即可在显示屏上显示出倒计时。

**走时模块：**

原理与秒表功能的走时模块一致，区别是一个从0开始增加，一个从设定时间开始减少。因此程序的设计大致一致，只是需要更改进位逻辑。

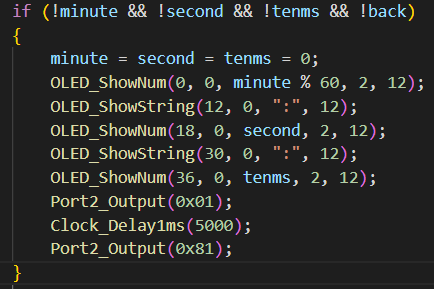


**时间设置模块：**

与上面闹钟功能的时间设置模块原理一样且改变时间的两个按键也一致，可代码复用。

**响铃模块：**

当时间减至“00:00:00”时，需要判定为倒计时结束，使其不再减少，并且还需要闹铃响起以提示倒计时已经结束。倒计时结束的判定通过简单的与关系便可实现，响铃响起需要将P2.7端口配为低电平，其主要原理参考上述闹铃功能处，在这里我们设置了铃响时长为5s（不至于响铃时间太长又能起到提醒的效果）。实现程序如下：



1. **串口校时功能实现**

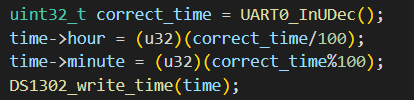
该功能是通过串口在客户端输入时间数据，发送到MSP432主控板上，以此更正自己的时间。MSP432 单片机内置了 UART 功能，它可以将外部设备串行数据转换为并行数据接收。因此，客户端与主板间的通信，若使用UART的话直接调用库函数即可实现；而且对于UART通信的硬件实现也很简单，只需将UART0通过USB线与电脑端连接即可。基于以上两点，我们选择通过UART来实现串口校时功能。

代码端的编写，通过阅读UART0库，主要用到三个函数。首先是对UART0进行初始化：

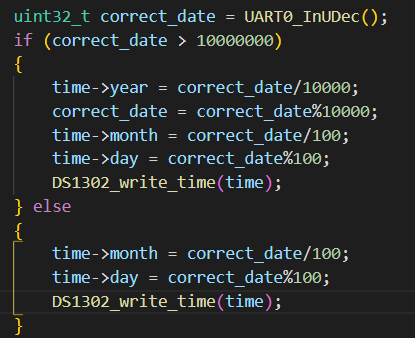


然后是调用 UART0\_InUDec() 通过电脑终端发送数据。UART0\_InUDec() 函数返回我们输入是数据并发送给MSP432，根据函数源码的设定，发送数据的范围为 。若要校准时间，需发送4位数据，分别代表时和分；若要校准日期，需发送8位数据或4位数据，8位时前四位代表年，后四位代表月和日，4位时仅代表月和日。以上的数据大小均小于函数可发送数据的最大值，因此数据位的设定可行。在电脑端输入数据后，需要对其进行取位操作以区分不同的时间和日期。具体代码如下：

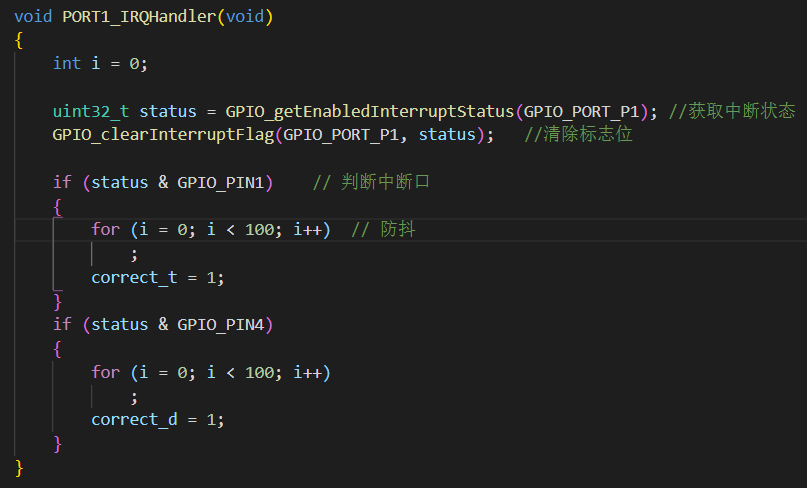
时间取位：



日期取位（分为了两种数据情况）：



同时为了能够开启串口校时功能，需要设置按键中断来表示进入校时功能，我们分别设置了P1.1/P1.4中断表示时间校准和日期校准：

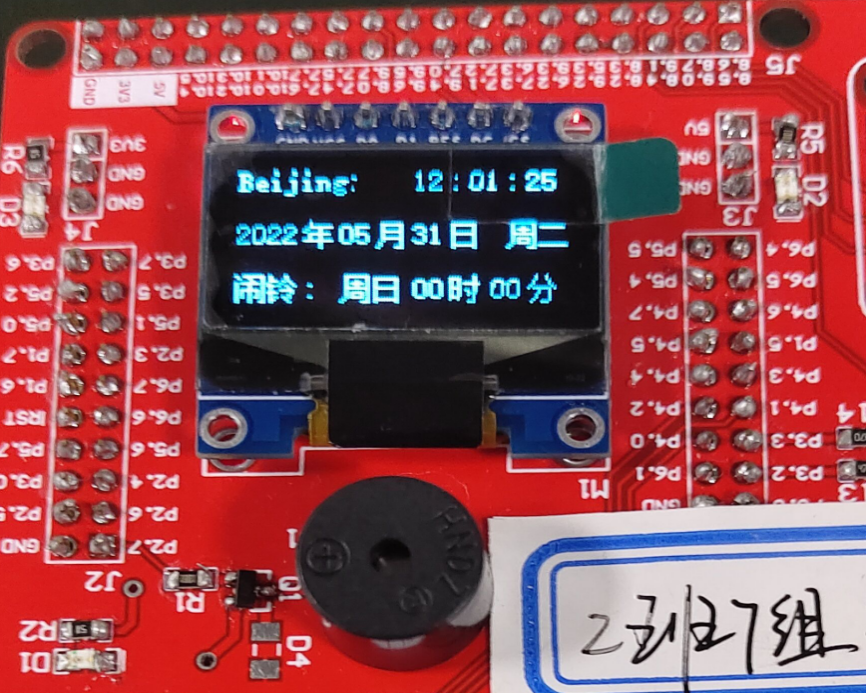


**四、系统整机测试**

在硬件组装完毕和软件程序调试完毕后，对系统进行整机测试。可分成显示模块测试与功能模块测试，首先检测系统的各个界面能否正常显示，再对每一个具体的功能进行测试。

1. **显示模块测试**

按下home键，观察主界面显示情况：



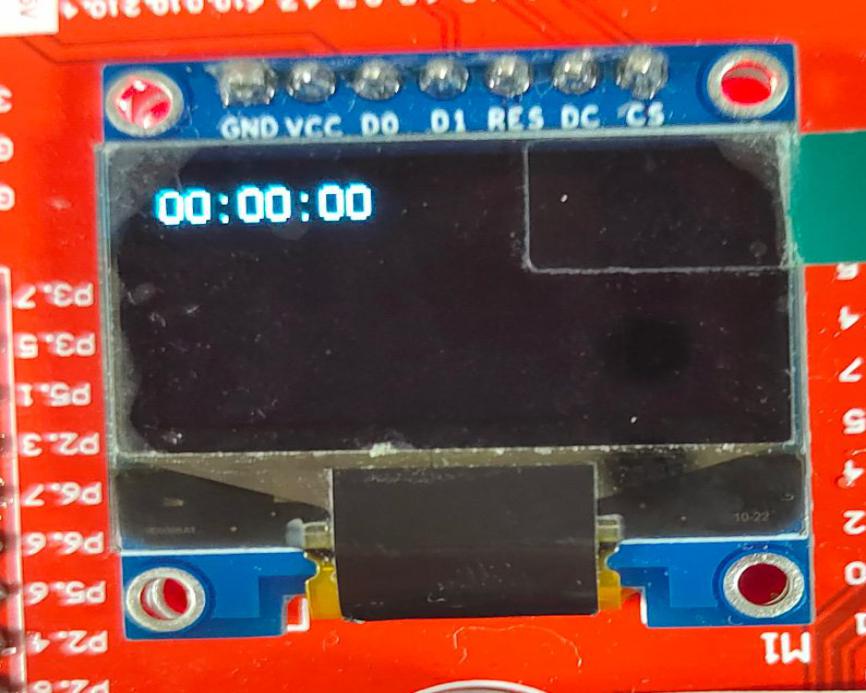
按下switch8，观察主界面是否切换小时制：



按下设置键，观察主菜单界面显示情况：



选中秒表功能，按下选择键进入秒表界面：



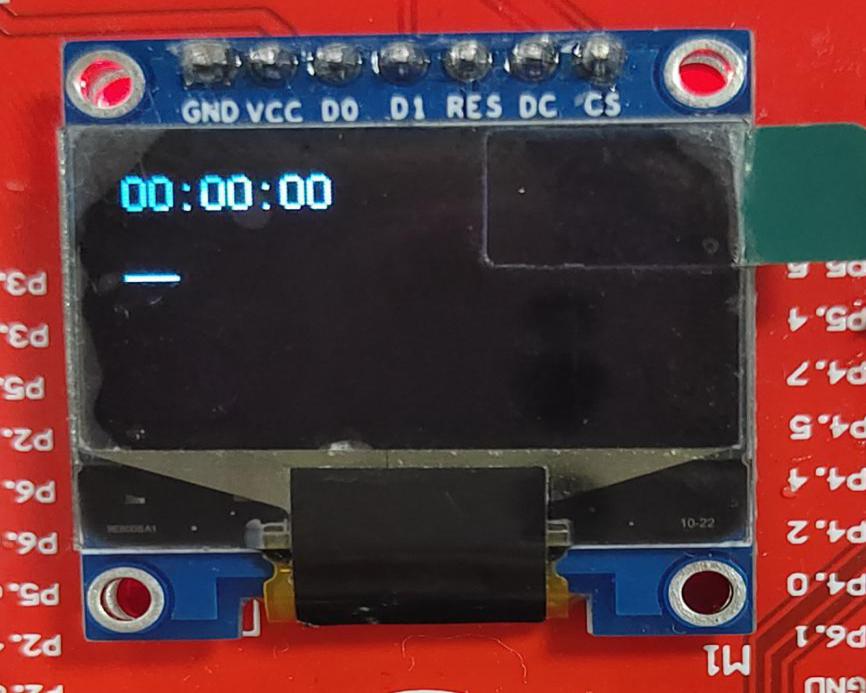
选中闹铃功能，按下选择键进入闹铃界面：



选中世界时钟功能，按下选择键进入世界时钟界面：



选中计时器功能，按下选择键进入计时器界面：

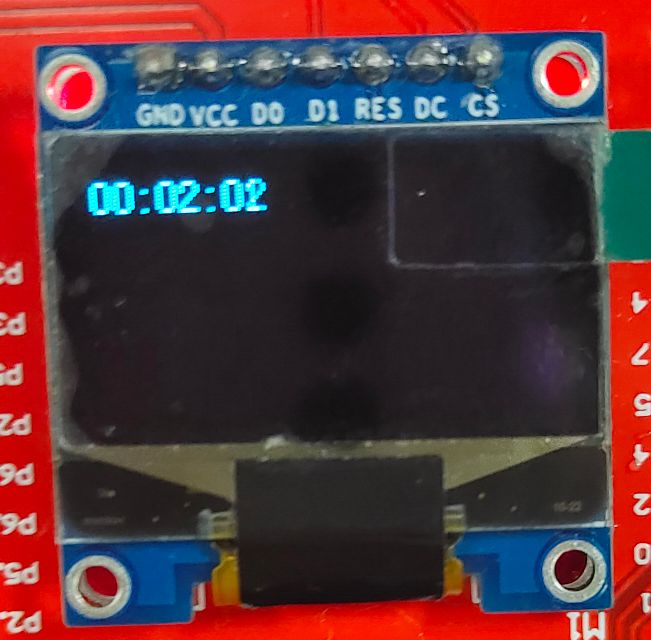


显示功能均正常。

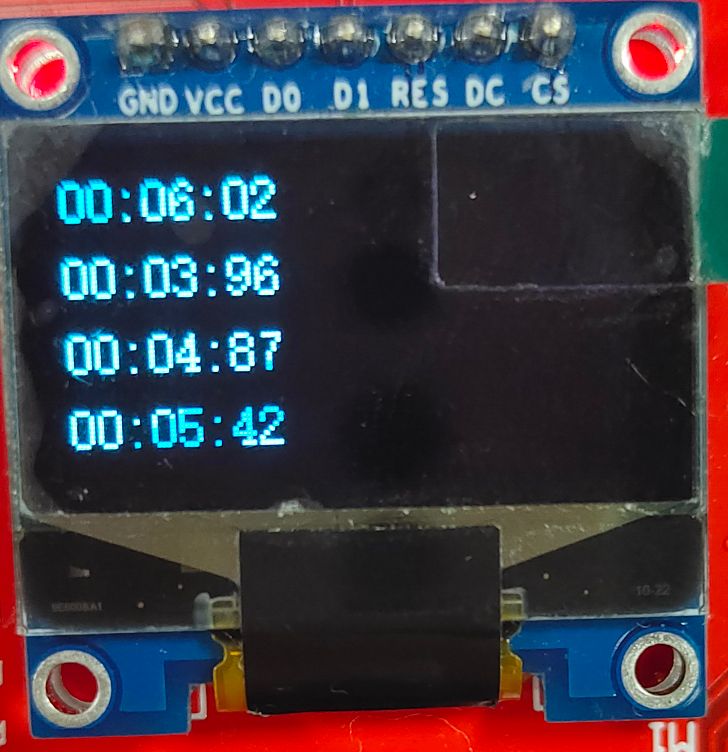
**（二）功能模块测试**

**1.秒表功能**

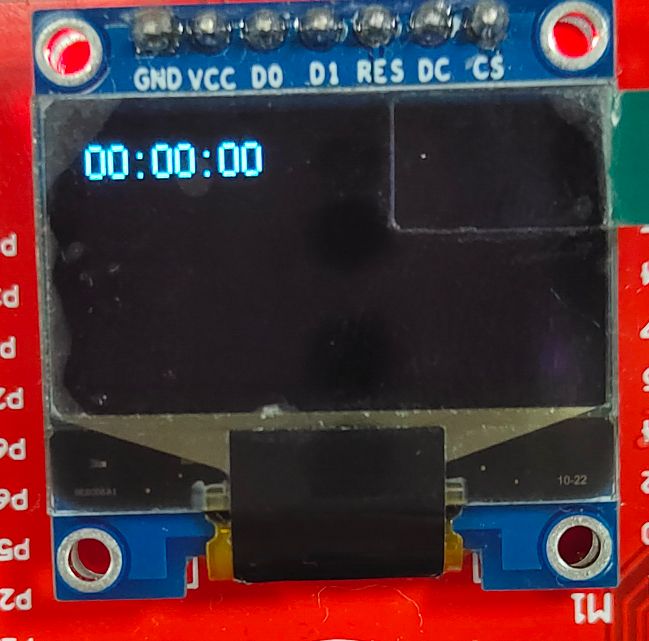
进入秒表界面后，按下确定键开始走时：



再次按下确定键，记录下当前时间，一共可以记四次，第一行为最后一次：



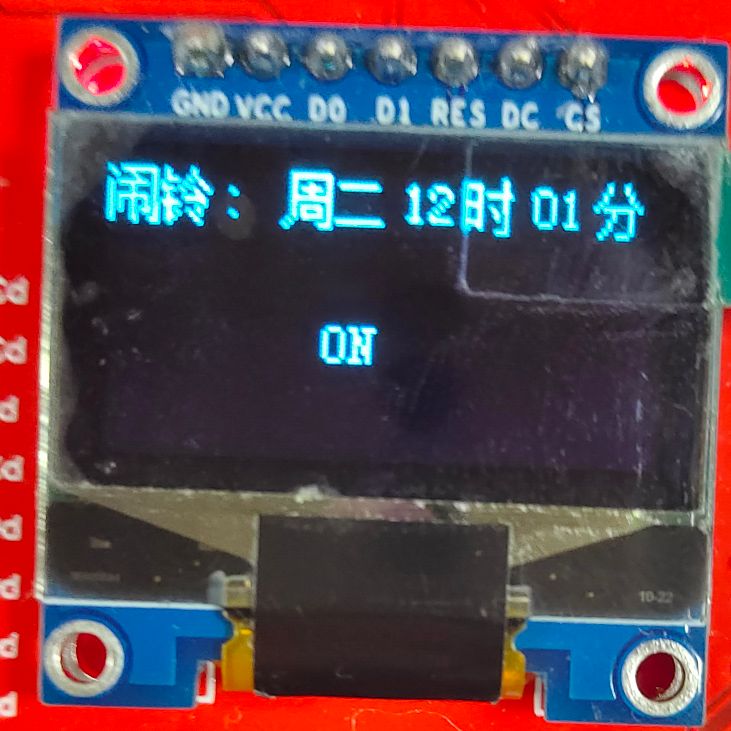
记完四次时后，再次按下确定键，秒表将会清零，回到初始状态：



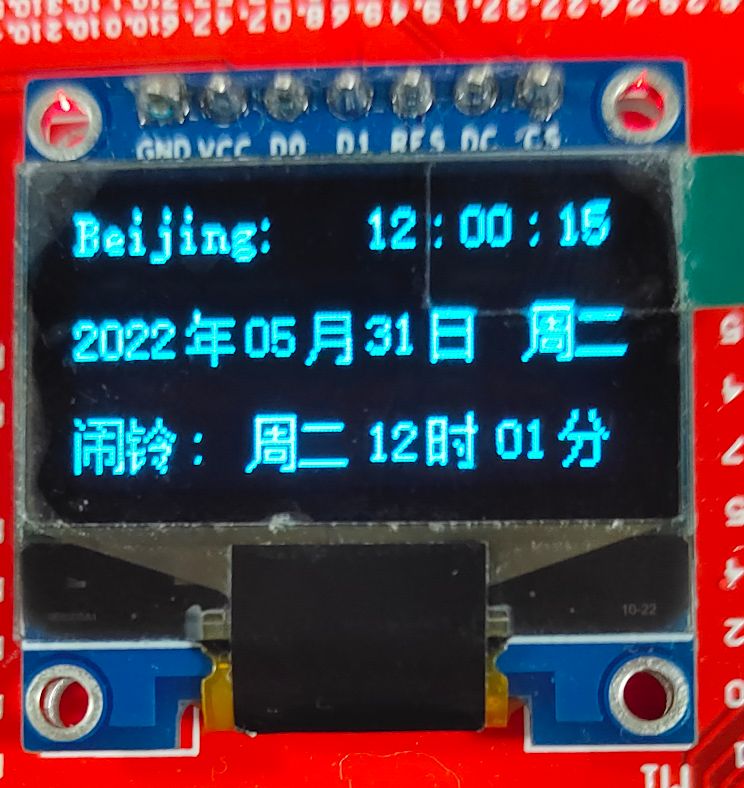
测试结果与设计的功能一致，秒表功能正常。

**2.闹铃功能**

进入闹铃界面后，先设置一个闹铃时间，设置完后按确定键打开闹铃，操作完后的秒表界面为：



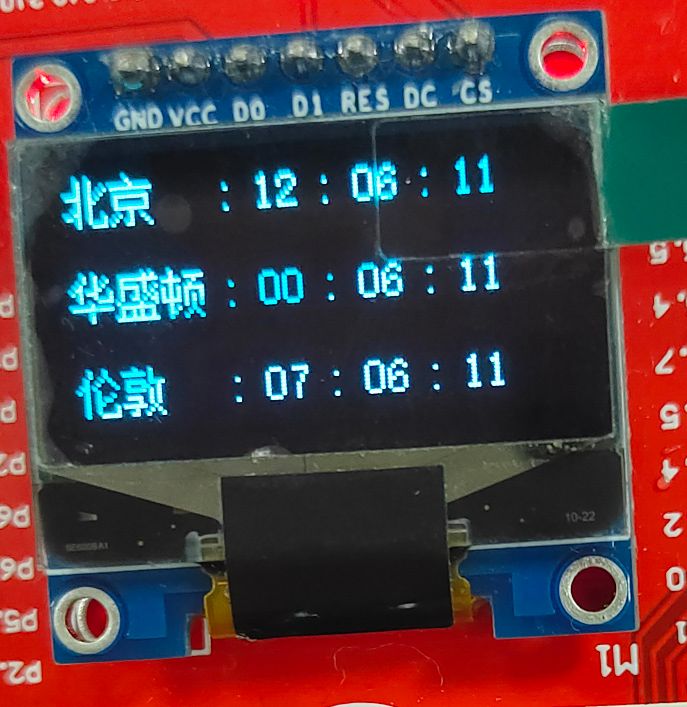
设置完闹铃时间后，返回home界面，home界面显示了设置的闹铃时间：



当第一行的时间与闹铃时间一致时，蜂鸣器开始工作，按下返回键后蜂鸣器停止工作。闹铃功能正常。

**3.世界时钟功能**

进入世界时钟功能后，显示三个时区的时间：



世界时钟功能正常。

**4.计时器功能**

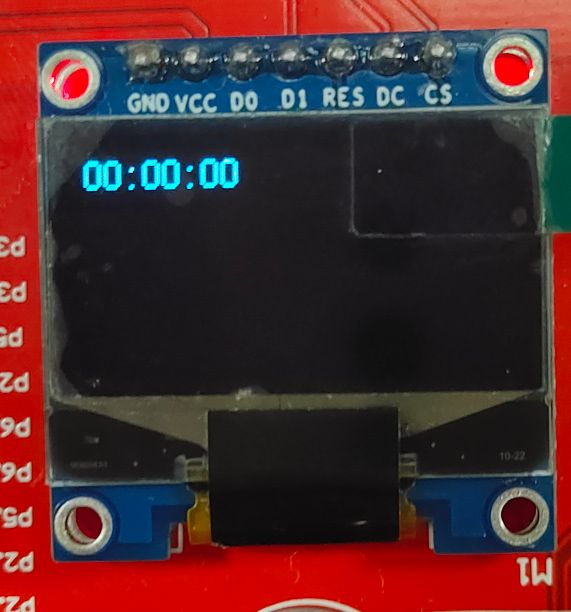
进入计时器功能后，先设置一个倒计时时间：



设置完倒计时时间后，按下确定键，开始倒计时：



倒计时结束后，蜂鸣器工作五秒以提示倒计时结束：



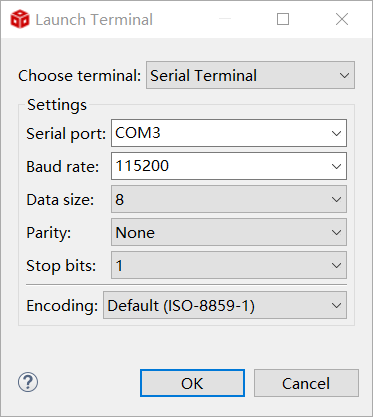
计时器功能正常。

**5.串口校时测试**

首先用USB 线将MSP432主控板与电脑端口相连：

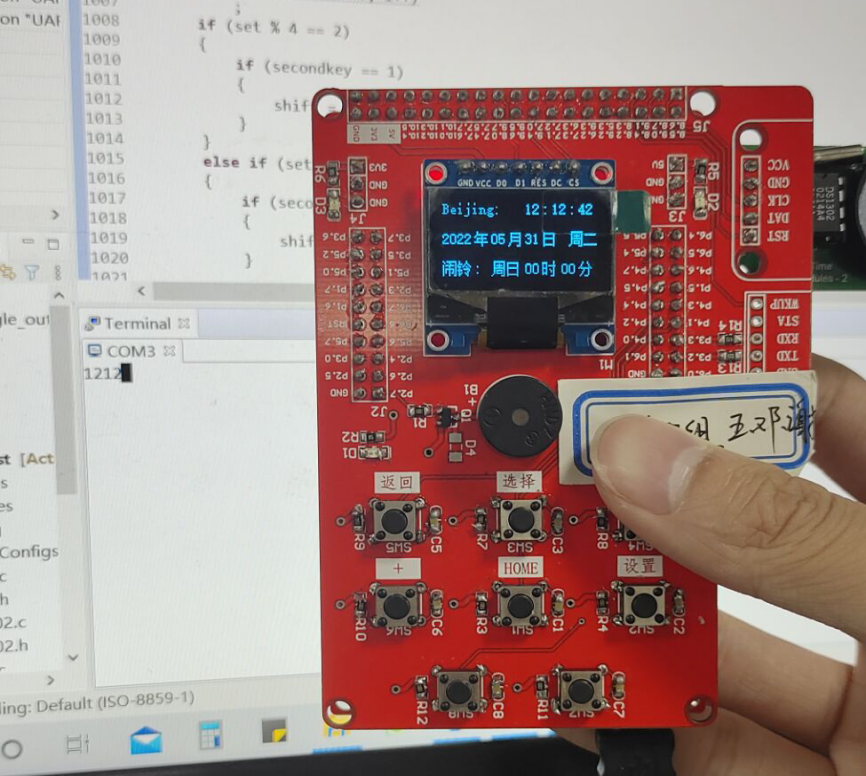


接着在CCS打开串口：



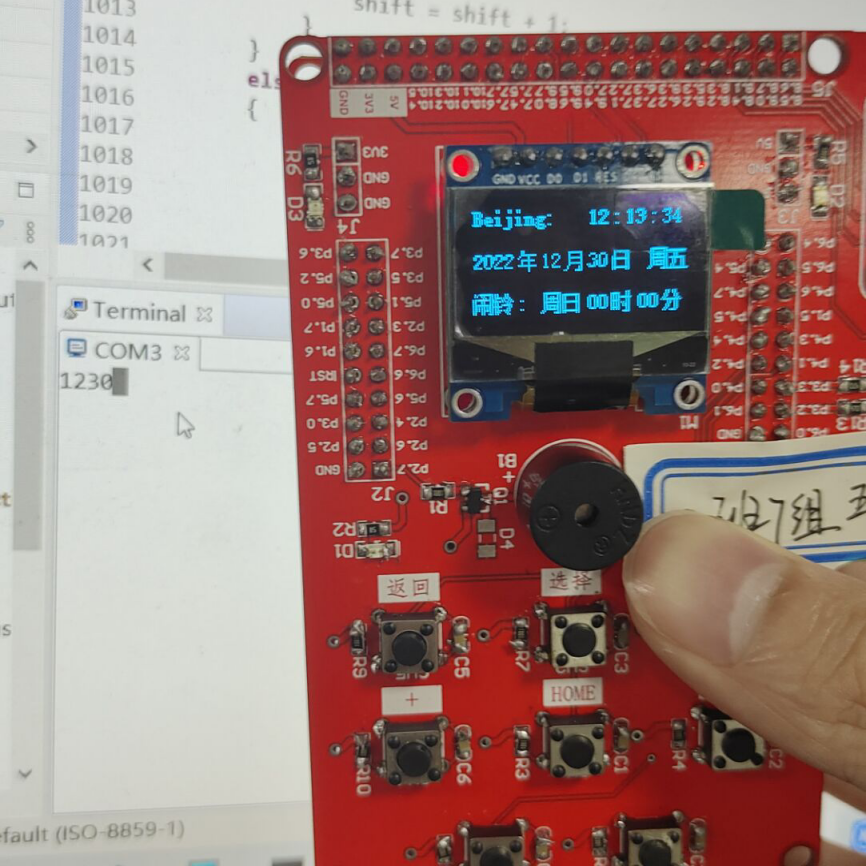
这里的Serial port需要根据实际连接的端口选择，波特率需要选择115200；

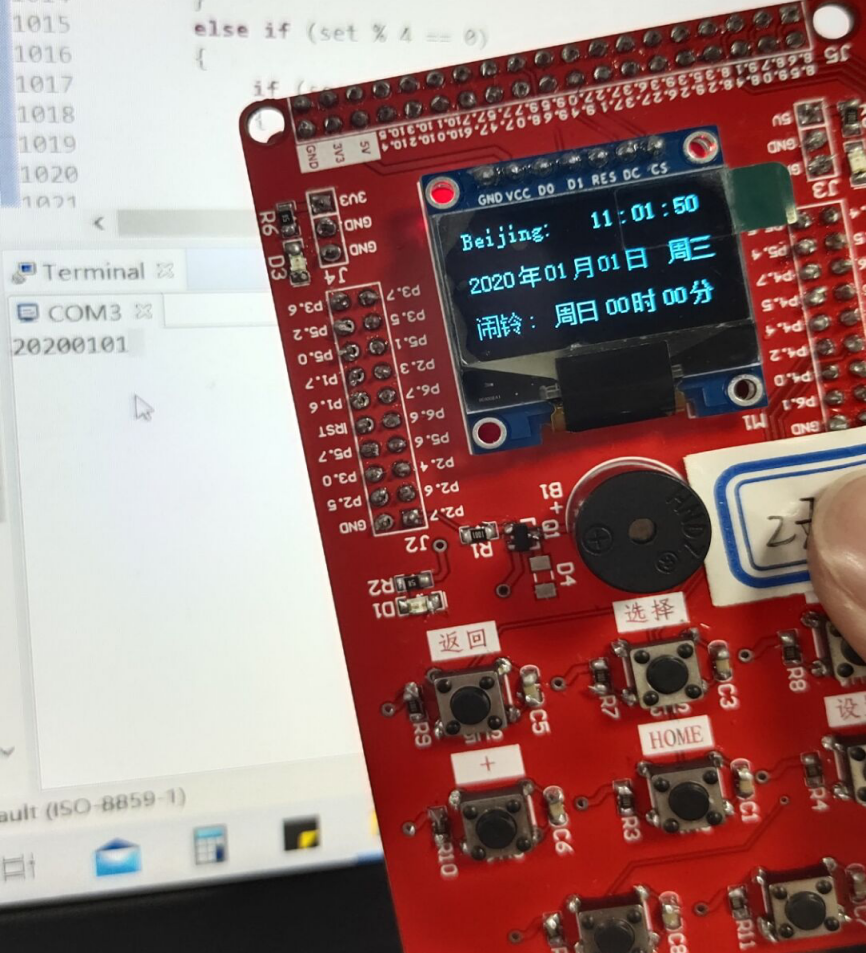
然后按下P1.1键，进入时间校准功能，在电脑终端输入一个四位数据，按下回车键发送，观察OLED上的时间：



可以看到OLED上的时间变为了后面终端上的时间，时间校准正常；

再按下P1.4键进入日期校准功能，与上述操作一致，测试图如下：





两种数据位情况下OLED上的日期均显示为电脑终端上的日期，功能正常。