**数字钟大作业介绍**

**强陶 522031910206**

**一、数字钟功能详细介绍**

**开机状态**：显示学号后8位31910206，显示姓名的拼音qiangtao。LED全部闪烁3次。

**sw1**:状态机，按下后可以切换不同的功能。

**状态1**（LED1亮起）：

以流水灯形式展示当前日期。分为2档速度。按下sw3进入最慢速模式，按下sw4进入快速模式。按下sw8进入左流水灯模式，按下sw7进入右流水灯模式，按下sw6正常显示。

**状态2**（LED2亮起）：

对当前日期进行修改。按下sw3对增加day,按下sw4减小day。按下sw8增加month,按下sw7减小month。按下sw6增大year,按下sw5减小year。

**状态3**（LED3亮起）：

以流水灯形式展示当前时间。分为2档速度。按下sw3进入最慢速模式，按下sw4进入快速模式。按下sw8进入左流水灯模式，按下sw7进入右流水灯模式，按下sw6正常显示。

默认从12:00:00开始计时，秒数的走动与真实时间的流速对齐。

**状态4**（LED4亮起）：

对当前时间进行修改。按下sw3对增加second,按下sw4减小second。按下sw8增加minute,按下sw7减小minute。按下sw6增大hour,按下sw5减小hour。

在修改时间的时候，计时停止。

**状态5**（LED5亮起）：

以流水灯形式展示当前时间。分为3档速度。按下sw3进入最慢速模式，按下sw4进入快速模式，按下sw5推出流水灯模式。

默认闹钟时间为11:59:59，计时的时间走到闹钟时间时，会在串口软件发送提示，并且播放音乐。

**状态6**（LED6亮起）：

对当前闹钟进行修改。按下sw3对增加alarm\_second,按下sw4减小alarm\_second。按下sw8增加alarm\_minute,按下sw7减小alarm\_minute。按下sw6增大alarm\_hour,按下sw5减小alarm\_hour。

在修改闹钟时间的时候，闹钟不行使对应功能。

**串口通信指令**：以上操作都可以在串口通信软件中进行，具体指令详见readme.txt。

1. 项目思路介绍

数字钟是一个功能齐全，较为复杂的项目。与先前4个Arm项目不同，前4个Arm项目的主要目的是通过修改已有的项目，学习如具体部件控制，串口通信等相关知识。数字钟需要从头开始搭建一个完整的大项目，具有很大的挑战性。

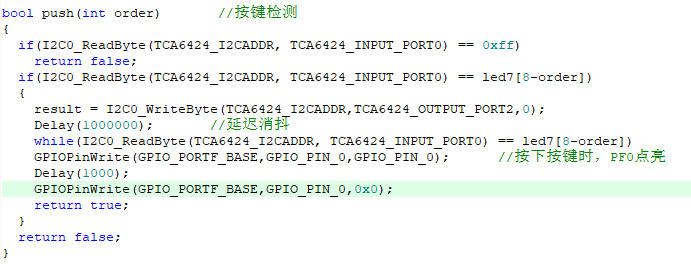
1、项目架构

开发项目的核心是：架构与封装。我从二者分别入手对于项目进行推进。在架构层面，我使用变量func来控制此刻为何种功能，并且通过按动板子上的按钮sw1来对func进行改变，从而完成对于状态的控制。在开发前，我将自己将要实现的功能进行罗列，然后通过if else语句划分不同功能对应的代码区域，从而分模块对大任务进行推进，使得我的开发思路极为明确。



2、功能的封装

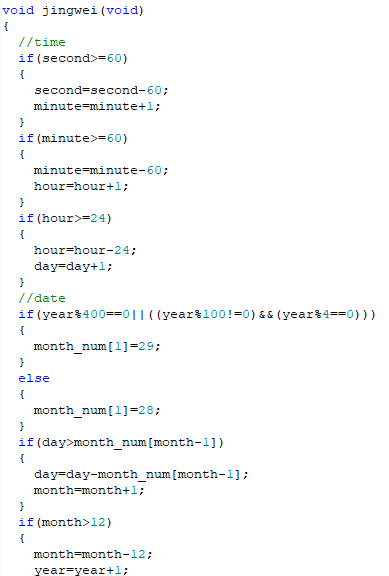
从封装的层面上，封装的目的是为了将所需使用的底层功能抽象出来，能够在更高层进行开发时直接调用，不考虑其内部实现方式，也避免代码过于冗长。

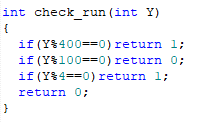


比如对于按键的检测，直接抽象出函数push(x)，将按下直接使用push(x)函数来判断是否按下了sw-x按键。极大减少了代码量。

3、细节与代码合理性的维护

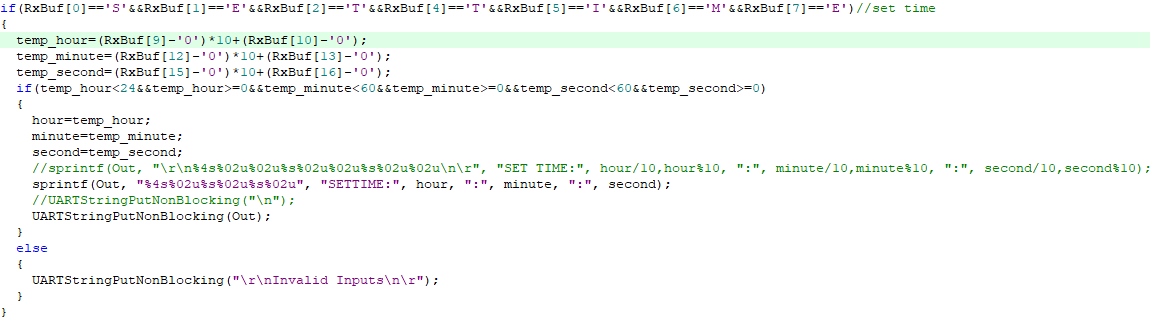
对于当前时间、日期、闹钟的维护，重点在于其判断的精准性与合理性。比如对于闰年的判断，对于不同月份日期的判断，都需要遵循客观规律。尽量避免不符合事实的数据呈现到产品上。



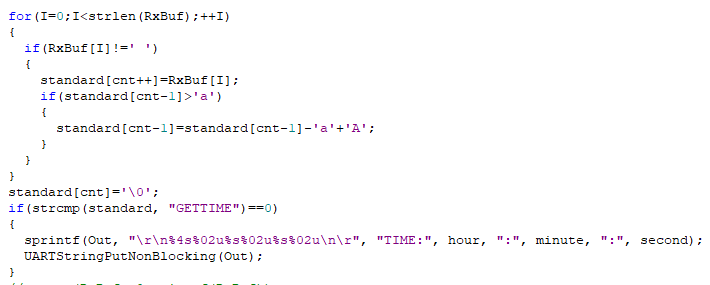


4、串行通信与字符串处理

在串行通信的维护上，字符串的处理是至关重要的部分。



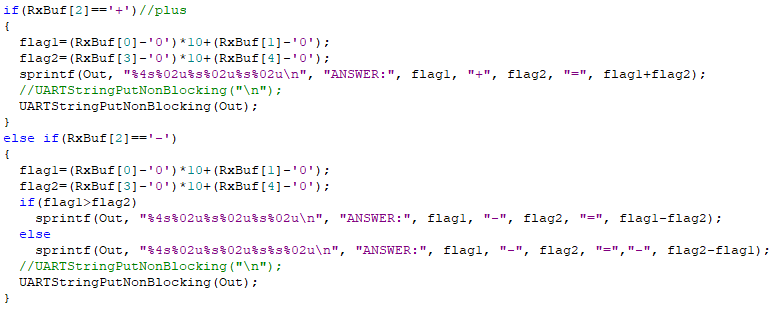
一方面，我们要从字符串中提取到所需要的数字，可以直接规范好输入与输出的形式，这样就可以在特定位置上取到我们需要的数据。



另一方面，对于用户输入的容错处理。我们可以把用户输入的不规范字符串处理为规范格式：去除所有的空格，将所有的字母修改为大写字母。处理完后再去和标准模板进行比较，选择需要进行的操作。

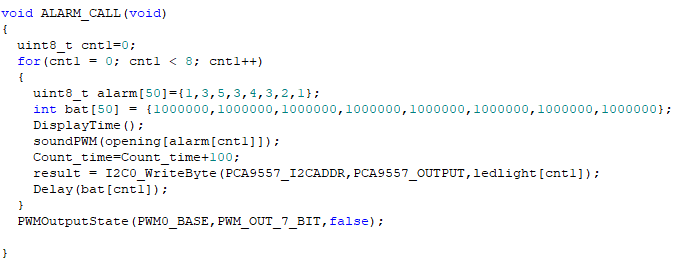
1. 项目亮点介绍

1、计算器模块



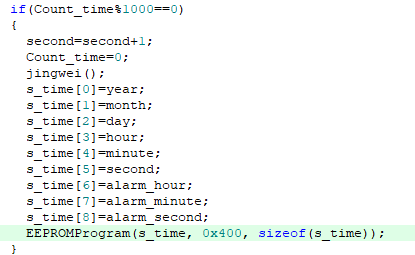
项目由C语言进行维护，由于C语言是一门较为底层的语言，可用于数值分析相关的工作。于是我将简单计算器的模块加入到了数字钟里。通过串口通信软件，可以完成加法与减法的交互。

2、闹钟模块



板子中集成了蜂鸣器的模块，可以通过代码调节占空比，生成旋律并发出声音。这个模块的添加与闹钟模块结合密切，从而除了在串口通信软件中进行相关文字的提示外，还可以直接发出声音来完成闹钟的功能。并且考虑到蜂鸣器发声时使用了Delay函数，会影响到计时器的正常使用，所以在每一次Delay后都对Count\_time变量进行操作，保证蜂鸣器发声不影响时间的变化。

1. EEROM的使用



使用了EEROM来对DATE,TIME和ALARM三类变量进行维护，即使断电也可以保存这三种变量，从而在reset后继续使用上次保存的变量。在每次更新时间时就进行存储，最大限度提升保存的精度。

1. 心得与体会

前四个Arm实验主要以修改已有代码，实现某种特定的功能作为目的来学习。但是数字钟大作业是一个从无到有，自己搭建整个体系的过程。整个写作的过程，对我来说也是一个整合的过程。就像搭积木一样，把之前几次实验的学习成果有机结合，最终组成了这样一个大的项目。

在整个过程中，我也获得了一些经验教训。我在整个项目的推进过程中，没有对代码进行严格的版本管理。全程将代码存储在移动硬盘而非github上，导致在遇到无法解决的代码问题时无法通过回溯进行解决。此外我的开发环境是装载了keil的windows虚拟机。相较于宿主机而言，虚拟机的反应速度和开发体验都较差，我应该在一开始就对宿主机进行环境配置。最后是在审题方面，我对需求的把握不算精准，所以在满足需求这个层面踩了一些坑，花费了多余的精力去完成所有任务。

在整个学习过程中，感谢何老师与助教学长对我的问题进行及时解答。在整个实操的过程中，我对于封装与架构的体会愈发深刻，这将会对我未来的发展产生巨大的帮助。