



東南大學

《计算机应用系统课程设计》 实验报告

姓名： 白丰硕 学号： 71116233

姓名： 马欣宇 学号： 71116302

东南大学计算机科学与工程学院、软件学院

二〇一九年六月十二日

一、课程设计题目

基于单片机 MSP430 的多功能应用设计

二、课程设计的主要功能描述

利用单片机设计了一个多功能小应用，包括时钟、计时器、时间设置三大模块。通过选择对应的功能模式，进行简单设置，最后由 LED 进行显示。

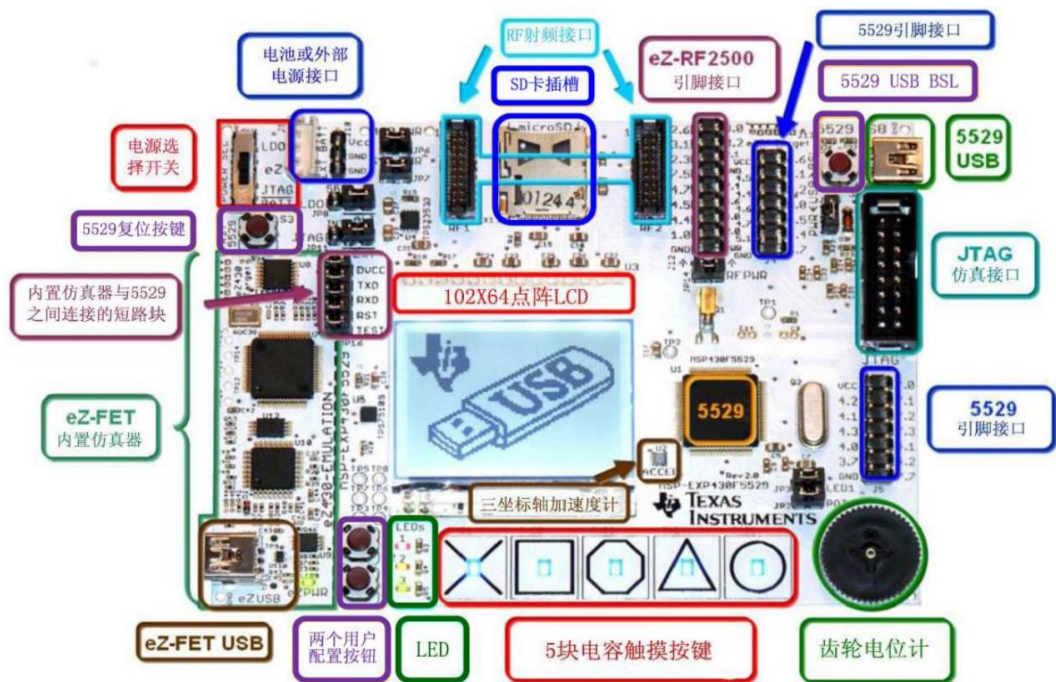
三、课程设计的目的

1. 掌握单片机 MSP430 的工作原理
2. 能够进行单片机简单系统设计，包括电源模块、复位模块、及相应的控制模块的设计
3. 掌握单片机的指令系统能够针对具体的设计要求来编写相应的控制程序

四、开发工具描述

MSP430F5529 开发板(MSP-EXP430F5529)是 MSP430F5529 器件的开发平台，为最新一代的具有集成 USB 的 MSP430 器件。该开发板与 CC2520EMK 等众多 TI 低功耗射频 无线评估模块兼容。开发板能帮助设计者快速使用 F5529 MCU 进行学习和开发，其中 F5529 MCU 为能量收集、无线传感以及自动抄表基础设施 (AMI) 等应用，提供了业界最低工作功耗的集成 USB、更大的内存和领先的集成技术。MSP430F5529 开发板的结构组成如图 1.3 所示。我们从左上角开始，按顺时针方向介绍：MSP-EXP430F5529 集成了电源选择开关（4 种电源选择方式）、电池或外部电源接口、1 个 RF 射频接口、Micro SD Card 插槽（附 1G 内存卡）、MSP430F5529 引脚接口、5529USB 接口、JTAG 仿真接口、1 个齿轮电位计、5 块电容触摸按键、9 个 LED、4 个按钮（2 个 用户配置按钮、1 个复位按钮、1 个 BSL 按钮）、eZ-FET 内置仿真器、1 块 102x64 点阵 LCD、1 个三坐标轴加速度计以及 MSP430F5529 芯片。该开发板将 F5529 部分引脚接出来，方便用户进行实验操作，既适合科研开发，又适合实验教学、课程设计、毕业设计等，为广大高校师生提供了良好的实验开发环境，同时也是广大电子爱好者学习、开发 MSP430 系列单片机的良好平台。

实物硬件资源图



结构框图

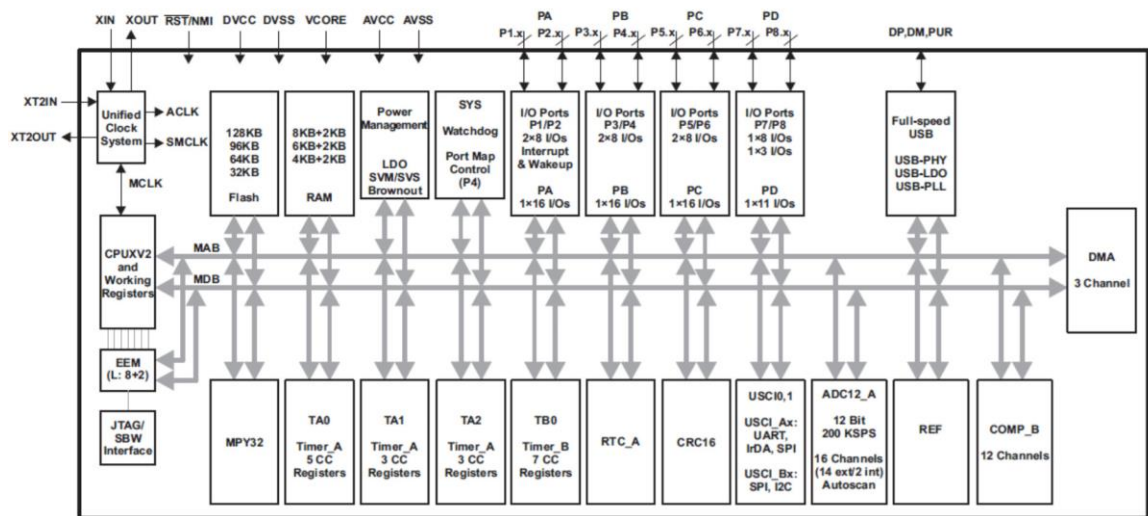


图 1.2 MSP430F5529 结构框图

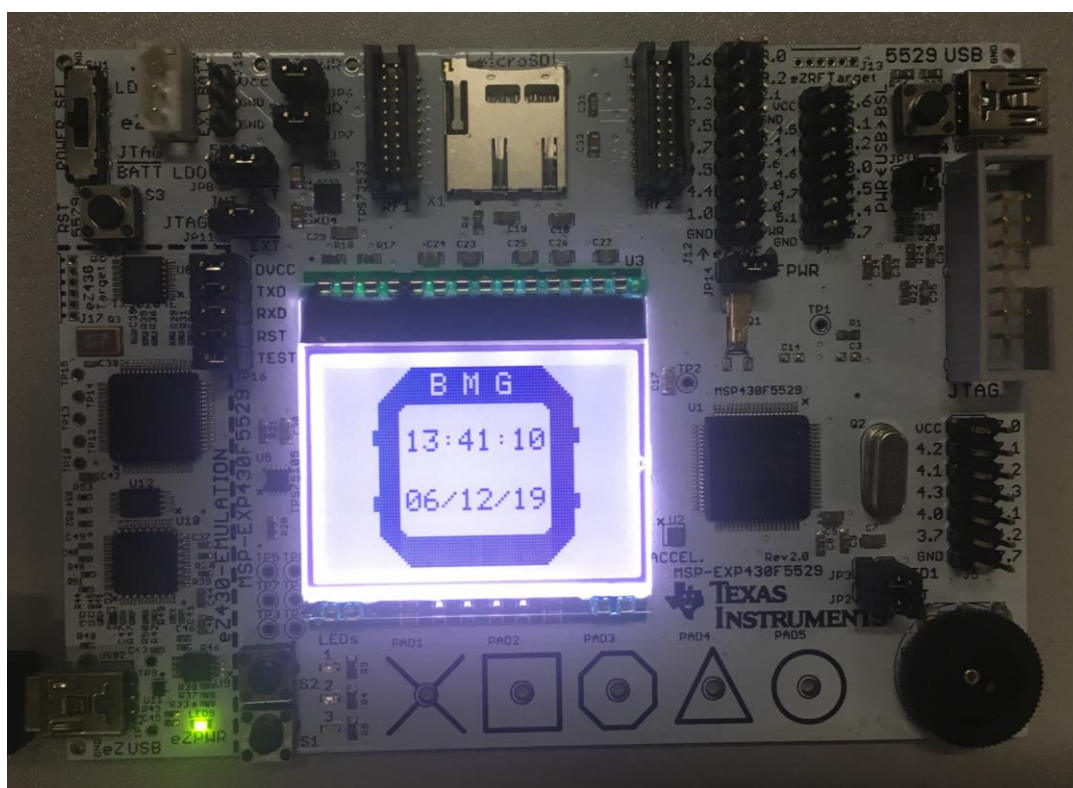
五、课程设计的主要内容和要求

1. 设计单片机工作电源模块及其复位电路
2. 设计四个显示模块：由 LCD 屏幕显示当前模式内容
3. 选择传感器，通过单片机采集其数据
4. 使用汇编语言编制相应的控制程序，并对软件进行调试

六、功能模块介绍与说明

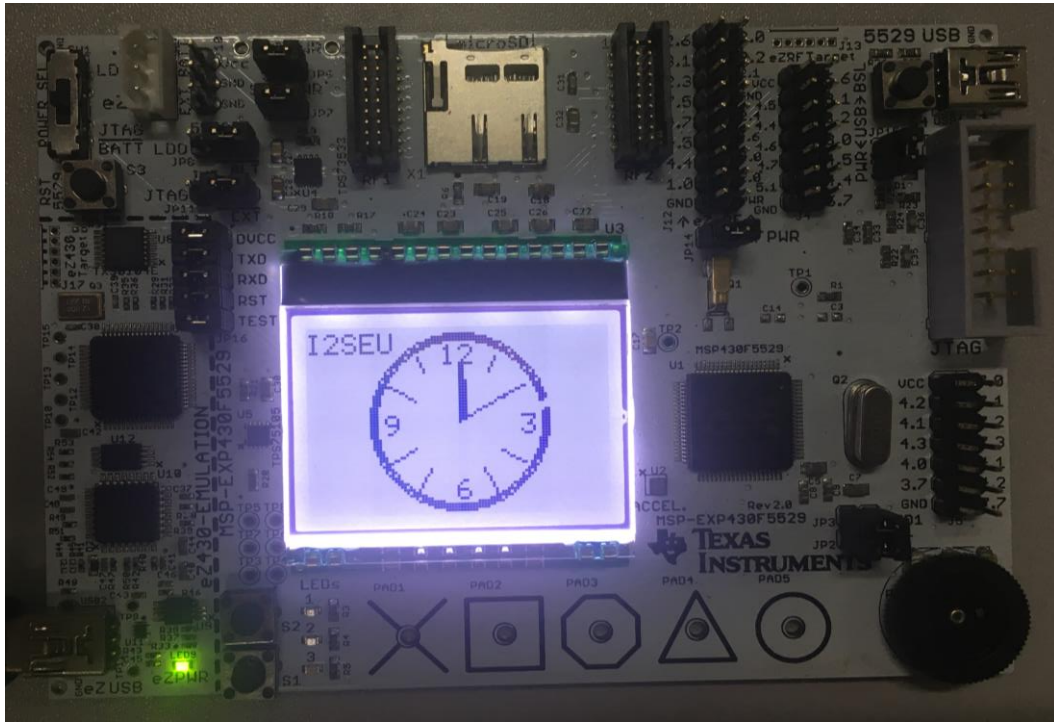
数字时钟

通过调用 `Display()` 函数，来显示其中表盘和表盘的数字，上排分别是时分秒，下排是日月年。表盘上显示的是我们的小组名称“BMG”。在 `Display()` 中分别调用清屏函数 `Dogs102x6_clearScreen()` 清除上一页所留下来的全部像素。再调用 `Dogs102x6_imageDraw()` 构建表盘轮廓。在没有按下任一个按键时，一直保持更新数字，如果发生数字发生改变，就调用 `DigitalClockDraw()` 函数将数值进行更新。



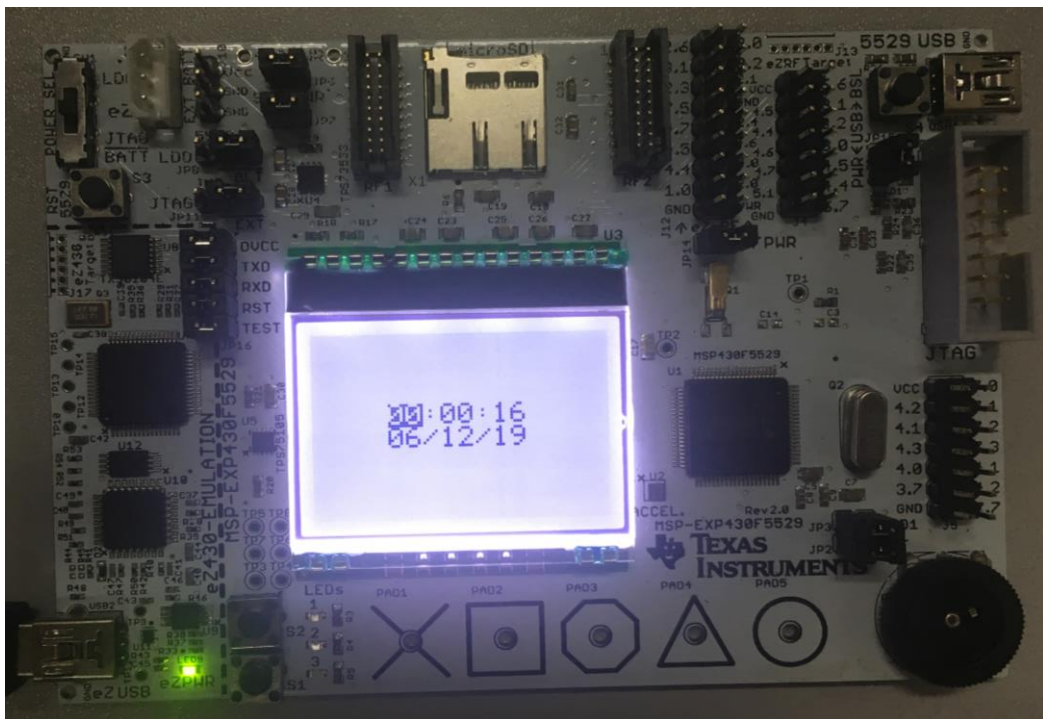
计时器

通过调用 `AnalogClock()` 函数，来启动这个计时器，进入模拟时钟之前会显示 **Pokemon** 精灵球 logo，这是我们小组的队伍 logo。持续一段时间后，就会显示计时器。这个计时器是由一个圆形表盘的模拟时钟来展示。三根针分别表示时分秒针，最小分辨率是一秒，秒针随着时间的变化不断的转动。该计时器还有清零的功能，如果需要重新计时，只需要按一次 `S1` 按键，即可完成时间的重置，且不会显示进入的画面。任意时间按 `S2` 键即可退出到主菜单界面。



时间设置

时间设置功能模块，主要是调用 `SetTime()`，来设置年月日时分秒。上排分别是时分秒，下排是日月年，首先调用 `Dogs102x6_clearScreen()` 函数进行一次清屏操作，然后调用 `DigitalClockUpdate()` 函数来更新时间，通过按键调节不同的修改位置来改变时间。内部逻辑主要是调用 `Dogs102x6_charDraw()` 来改变现实的数字，通过转动齿轮来修改时间表示的值。最后按键表示退回到主菜单。



七、实验主要代码

```
void DisplayDigitalClock(void)
{
    buttonsPressed = 0;
    Dogs102x6_clearScreen();
    Dogs102x6_imageDraw(digitalWatch, 0, 0);
    while (!buttonsPressed)
    {
        hourBCD = RTCHOUR;
        minuteBCD = RTCMIN;
        secondBCD = RTCSEC;
        if (hourBCD != hourOldBCD || minuteBCD != minuteOldBCD ||
            secondBCD != secondOldBCD)
        {
            DigitalClockDraw();
        }
        hourOldBCD = hourBCD;
        minuteOldBCD = minuteBCD;
        secondOldBCD = secondBCD;
    }
    buttonsPressed = 0;
    Dogs102x6_clearScreen();
}

void DigitalClockDraw(void)
{
    DigitalClockUpdate();
    Dogs102x6_clearScreen();
    Dogs102x6_imageDraw(digitalWatch, 0, 0);
    Dogs102x6_charDrawXY(36, 2, 'B', DOGS102x6_DRAW_INVERT);
    Dogs102x6_charDrawXY(47, 2, 'M', DOGS102x6_DRAW_INVERT);
    Dogs102x6_charDrawXY(58, 2, 'G', DOGS102x6_DRAW_INVERT);
    Dogs102x6_lineDraw(32,0,65,0,DOGS102x6_DRAW_NORMAL);
    Dogs102x6_lineDraw(32,1,65,1,DOGS102x6_DRAW_NORMAL);
    Dogs102x6_lineDraw(32,10,65,10,DOGS102x6_DRAW_NORMAL);
    Dogs102x6_lineDraw(32,11,65,11,DOGS102x6_DRAW_NORMAL);
    Dogs102x6_stringDrawXY(27,20,&timeVector[ HOUR10],DOGS102x6_DRAW_NORMAL);
    Dogs102x6_stringDrawXY(27,39,&date[MONTH10],DOGS102x6_DRAW_NORMAL);
}

void Pokeman(){
    Dogs102x6_clearScreen();
```

```
Dogs102x6_circleDraw(51, 32,30, DOGS102x6_DRAW_NORMAL);
Dogs102x6_circleDraw(51, 32,29, DOGS102x6_DRAW_NORMAL);
int j = 4;
while(j--){Dogs102x6_circleDraw(47,32,10+j,DOGS102x6_DRAW_NORMAL);}
Dogs102x6_circleDraw(47, 32,5, DOGS102x6_DRAW_NORMAL);
Dogs102x6_circleDraw(47, 32,6, DOGS102x6_DRAW_NORMAL);
int i = 6;
while(i--){
    Dogs102x6_lineDraw(21,29+i,36,29+i,DOGS102x6_DRAW_NORMAL);
    Dogs102x6_lineDraw(81,29+i,61,29+i,DOGS102x6_DRAW_NORMAL);
}
__delay_cycles(100000000);
}
```

八、实验问题

在本次实验室中遇到了很多的问题。在最开始的时候，遇到很多编译的错误，主要是由于运行环境和编译工具引起的，导致很长时间都花费在查资料解决编译的问题。经过晚上查询的资料，环境和编译的问题被解决了，开始构建主要的业务逻辑。这个板子的显示屏的显示机制是由 204 个 `uint8_t` 来表示各个位置的显示像素，如果这个 `bit` 为 1 则为黑色，如果为 0 则为白色。但是它和传统的图片显示不是完全一致的，这也引起了很多想打印出想打印的图形的问题，需要深入的了解其图像打印机制，来在合适的位置构建合适的绘图元素。

九、实验心得

在本次实验中，我学到很多单片机开发的知识，主要是了解到了单片机 MSP430F5529 的原理和结构，以及上面的所属模块，但是这次的实验中的，开发板的功能不是很完整，故有很多想尝试的小的功能模块没有尝试。再次之前参见过智能车比赛，所以对于嵌入式开发板是有一定的了解的，所以对于嵌入式开发还算是有一点经验可言。在这次实验过程中，遇到了很多的问题，比如说代码编译出现错误，其中可能会引起编译工具 `make` 的错误。还有导致文件链接失败，无法正确将代码烧录进开发板中。但是这些问题通过网上查阅资料，自行解决，后来开发的过程中便逐渐熟悉了开发流程，修改函数以及 `debug` 也变得轻车熟路。在设计启动动画时，原本打算是展现一个完整的精灵球出场的动画，但是由于时间的问题，没能完全的做完。但是根据之前的实验示例的基础之上进行自我变化和改变得到现在的成果。整体来说这一次开发过程中，还算是比较顺利，我认为自我探索式的学习方式是很有趣味性的，在课下时我们也可以多进行自我探索，学习更多更有趣的新知识和新技能。