零级标题

科技入门一本通之 430 系列单片机

写这篇 430 的介绍的前提是大家已经对 51 有了比较深入的了解，于很多单片机通用的知识这里就不在加以介绍了，而且在已经接触过 51 单片机之后，430 就更需要大家的自主学习了，所以对于 430，这里的讲解就不会像 51 单片机一般详细，而是更注重实例的列举，同时讲一下 430 和 51 单片机的一些差距，方便大家比较学习，然后给大家分享几个写 430 常用的库函数。

下面我们就进入正题。

一、430 系列单片机简介

msp430 系列单片机是 TI 公司生产的单片机，如果要参加 TI 杯（也就是小电赛），msp430 是必须要会用，而且对它的时钟中断这些都需要了解得比较透彻，它包含一个十六位精简指令集（RISC）CPU，单个时钟周期就可以执行一条指令，使用相同晶振，它的速度是 51 单片机的 12 倍，其另一大特点就是它的超低功耗，一个水果电源就可以驱动它工作了。msp430 系列单片机有很多，这里给大家介绍的是 G2553 系列单片机。一般都会选用 LaunchPad 开发板进行学习，编写 430 系列单片机的软件有两个，IAR 和 DDS，这里给大家介绍的是 IAR 软件，关于 iar 软件的下载，我这里没有安装包，大家去 TI 公司官网上可以找到安装包，百度上有详细的安装教程，或者大家可以关注软件安装管家公众号，在这里可以找到很多软件资源和安装教程，下面给大家分享一个 PPT，主要是介绍这款 LacnchPad 以及 iar 软件的使用，还有少许 C 语言重点知识。

链接：*https://wk.baidu.com/view/2256e748ba1aa8114431d9cd?pcf=2*

https://wk.baidu.com/view/20b4f57a524de518974b7d02?pcf=2

相信看完再经过一番实战，大家对于这款开发板已经有了一定的了解，关于 iar 软件的使用也没有问题了，虽然这个过程可能花了大家一定的时间，但是请记住，万事开头难嘛，其实学习单片机之前的软件安装和各种配置就是相对很复杂的一环了，所以后续一定要有信心，关于建立工程什么的这里就不在说了，msp430 单片机相对 51 单片机而言内部资源会多不少，这里不可能给大家一一讲解，或者说我有心一一讲解相信在没有实战的前提下大家也没有耐心看完，所以 430 单片机推荐学法就是用什么学什么，网上有丰富的教程和经验之谈，文末也会给大家分享 msp430 的中文资料，为了用而学比没有目的的学会高效很多，下面主要就 IO 口配置，时钟系统以及中断来讲述一下 430 单片机和 51 单片机的区别所在，讲完就开始给大家分享实用库函数。

二、430 系列单片机的 IO 口

相较于 51 单片机，430 系列单片机 IO 口的配置相对麻烦，因为不能使用位操作，而要对寄存器进行配置，所以首先我们要了解 IO 口涉及的寄存器。

例：利用 launchpadP1.3 口连接的按键控制 P1.0 口连接的 LED，实现按键按下灯亮，按键弹起灯灭。

```
#include<msp430.h>
int main(void)
{
    WDTCTL = WDTHOLD + WDTPW;  // 关闭看门狗(当不用看门狗定时器时,这句话一定需要,不然就会一直复位)
    P1SEL &= ~(BIT3+BIT0);  // 设置 P1.,P1.3 为 IO 口
```

```
    P1DIR |= BIT0;  //P1.0 口为输出
    P1DIR &= ~BIT3;  //P1.3 口为输入
    P1REN |= BIT3;  // 使能 sw2 为上下拉(P1.3)
    P1OUT |= BIT3;  // 使能 sw2 上拉
    While(1)
    {
        If((BIT3&P1OUT))
            P1OUT |= BIT0;  // 如果按键按下为高电平即灯亮
        else
            P1OUT &=~BIT0;  // 否则灯灭
    }
}
```

三、msp430 的时钟系统

1、msp430 基本时钟模块 + 简介

基本时钟模块 + 支持低系统成本和超低功耗。 采用三种内部时钟信号，用户可以选择性能和低功耗的最佳平衡。 为了实现无任何外部元件操作，可在全软件控制下，用一个外部电阻、一个或两个外部晶振、或用振荡器来配置基本时钟模块 +。

基本时钟模块 + 有 2 个，3 个或 4 个时钟源：

• LFXT1CLK：低频 / 高频振荡器可以与低频时钟晶振或外接 32768Hz 时钟源，或与标准晶振、振荡器，外部 400KHz ～ 16MHz 的外部时钟源一起使用。

• XT2CLK：可以与标准晶振、振荡器，或外部 400KHz ～ 16MHz 的外部时钟源一起使用的可供选择的高频振荡器。

• DCOCLK：内部数控振荡器 (DCO)。

• VLOCLK：内部超低功耗、12KHz 典型频率的低频振荡器。

2、msp430 的基本时钟模块 + 可提供的三种时钟信号：

• ACLK：辅助时钟。 ACLK 是由软件选择来作为 LFXT1CLK 或 VLOCLK。 ACLK 经 1，2，4，8 分频后得到。ACLK 可由软件选作各个外围模块。

• MCLK：主机时钟。 MCLK 由软件选择作 LFXT1CLK，VLOCLK，XT2CLK（如果片上提供），或 DCOCLK。 MCLK 由 1，2，4，8 分频得到。MCLK 用于 CPU 和系统。

• SMCLK：系统子时钟。 SMCLK 由软件选作 LFXT1CLK，VLOCLK，XT2CLK（如果片上提供），或 DCOCLK。 SMCLK 由 1，2，4，8 分频得到。SMCLK 可由软件选作各个外围模块。

3、三种时钟信号的区别

• 辅助时钟 ACLK 主要用于低速外设的，可以选作外围模块的时钟信号。

• 主时钟 MCLK 用于 CPU 和系统。

• 系统子时钟 SMCLK 用于高速外围模块。

4、msp430 基本时钟模块 + 寄存器

寄存器	简表	寄存器类型	地址	初始化状态
DCO 控制寄存器	DCOCTL1	读取/写入	056h	060h 与 PUC
基本时钟系统控制 1	BCSCTL1	读取/写入	057h	087h 与 POR
基本时钟系统控制 2	BCSCTL2	读取/写入	058h	用 PUC 复位
基本时钟系统控制 3	BCSCTL3	读取/写入	053h	005h 与 PUC
SFR 中断使能寄存器 1	IE1	读取/写入	000h	用 PUC 复位
SFR 中断标志寄存器 1	IFG1	读取/写入	002h	用 PUC 复位

(P 2-2-9) msp430 基本时钟模块 + 寄存器

对于时钟模块详细的寄存器名称以及操作，我想了很久，还是决定不写在此书凑字数了，因为当你们不需要运用的时候，多半是没有耐心看的，或者说很难形成记忆，但是在这里要提醒一下，等到要用时钟的时候，大家一定要记得好好研究寄存器，我记得我的学长说过，单片机的不同其实就在于寄存器的不同，了解不同的单片机主要就是了解它的寄存器。

四、msp430 的中断

关于中断的概念前面讲解 51 单片机的时候已经给大家详细介绍过了，这里就不重复介绍了，但是每一种单片机对于开启中断进入中断的格式时不一样的，，430 系列单片机有很多中断源，定时器计数器看门狗 AD 等在配置正确的情况下都可以触发中断，这里也不一一给大家介绍了，同样是需要用什么学什么，毕竟这本书是为了带大家入门，后面更深入的部分大家一定要自己去钻研，我们灌输的始终是我们知识，只有你们开始自主学习和练习了，才能转换成你们知识。下面就以一个简单的实例简单的讲解一下 msp430 的中断。

例：在 launchpad 上实现按键按下 LED 灯的状态翻转，要求用中断进行按键识别。（这个例子同时可以带大家复习 IO 口的知识）。

```
#include<msp430.h>
int main(void)
{
    WDTCTL = WDTHOLD + WDTPW;  //关闭看门狗
    P1SEL  &= ~(BIT3+BIT0); //设  置 P1.,P1.3 为 IO 口
    P1DIR |= BIT0;  //P1.0 口为输出
    P1DIR &= ~BIT3;  //P1.3 口为输入 ‘
    P1REN |= BIT3;   //使能 sw2 为上下拉（P1.3）
    P1OUT |= BIT3;  //使能 sw2 上拉
    P1IFG &=~BIT3;      //P1.3 口中断标志位清零
    _BIS_SR(GIE);      //*打开总中断（至关重要的一步！！我曾经就因为忘记这句话调代码调了好久）
    While(1)
    {
        }
}
    #pragma vector=PORT1_VECTOR  //这句话是根据中断向量表来的，一个字母都不能错，对于不同中断，一定要好好对照中断向量表，这句话一错根本进不了中断
    __interrupt void PORT1(void)   //这是中断的格式，函数名可以自己取，但是 __interrupt 是必须的
{
    P1OUT^=BIT0;  //按键按下一次 P1.0 口状态翻转一次
    P1IFG &=~BIT3;  //清除标志位
}
```

五、常用函数分享

1、长短按键检测函数

长短按键检测在 430 单片机实用时非常常用，因为 launchpad 上只有一个按键，多数情况下我们需要识别的状态却不止一种，为了提高按键的利用率，引入长短按键检测函数是很有必要的。下面这个工程就是一个完整的长短按键函数，大家用的时候只需要把它包含在 main.c 文件里再调用相应的子函数即可。

链　接：https://pan.baidu.com/s/1UrJm0sxrA3uCTnaY2RCPZw。

2、PWM 函数

首先给大家介绍一下 PWM，这个也是以后写代码很常用的。

PWM 是 Pulse Width Modulation 的缩写，中文意思就是脉冲宽度调制，简称脉宽调制。它是利用微处理器的数字输出来对模拟电路进

行控制的一种非常有效的技术。

PWM 是一种对模拟信号电平进行数字编码的方法。通过高分辨率计数器的使用，方波的占空比被调制用来对一个具体模拟信号的电平进行编码。PWM 信号仍然是数字的，因为在给定的任何时刻，满幅值的直流供电要么完全有(ON)，要么完全无(OFF)。电压或电流源是以一种通(ON)或断(OFF)的重复脉冲序列被加到模拟负载上去的。通的时候即是直流供电被加到负载上的时候，断的时候即是供电被断开的时候。只要带宽足够，任何模拟值都可以使用 PWM 进行编码。

这段话理解起来可能还有点吃力，其实 PWM 就是用来产生占空比（高电平所占比例）不同的方波信号，它只有两种信号即高和低，方波高处即为高，低处即为低。

这里给大家用一个简单的实例帮助理解 PWM，假如我们要做一个呼吸灯，何为呼吸灯呢，就是亮度随时间周期性变化的 LED 灯，那么我们怎么实现呢，这里主要需要的就是 PWM 了，我们只需要周期性的对方波占空比进行改变，即可实现亮度的周期性变化了，这是 PWM 最简单的运用实例，也是很多人理解 PWM 的第一步。下面就是库函数了，同样只需要包含进 main.c 文件调用子函数即可。

链　接：https://pan.baidu.com/s/1UrJm0sxrA3uCTnaY2RCPZw。

那对于 430 的简介也就到这里了，可能讲的确实粗略，但是希望大家看完能够有所了解，然后根据自己的所需以及兴趣去深入研究和学习，最后附上 430 的中文资料，也是我写 430 简介的参考资料之一，讲解相当详细，很适合大家学习。

链　接：https://www.docin.com/p-69128956.html。