学会了配置系统时钟和设置定时器中断，配置时钟的频率，配置时钟中断信号，设置定时器中断处理程序。
以下在实验报告中也有体现。
实验代码分析：

SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);

该句代码用配置时钟，SYSCTL_SYSDIV_5 代表 5 分频，SYSCTL_USE_PLL 代表使用 400MHZ 的 PLL 振荡器做系统时钟的发生器，SYSCTL_XTAL_16MHZ 代表使用 16MHZ 的晶振作为驱动。

SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
这两句代码是打开 GPIO 使能信号，配置 led 输出引脚

SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER0);
TimerConfigure(TIMER0_BASE, TIMER_CFG_PERIODIC)
打开 TIMER0 使能信号,配置 TIMER0 为 32 位周期性时钟

ui32Period = (SysCtlClockGet() / 10) / 2;
TimerLoadSet(TIMER0_BASE, TIMER_A, ui32Period -1);
设置时钟中断，中断时间 40M/(40M/10/2)=0.05s

IntEnable(INT_TIMER0A);允许 TIMER0A 连接一个中断处理
TimerIntEnable(TIMER0_BASE, TIMER_TIMA_TIMEOUT);允许 TIMER0A 在 TIMEOUT 时产生一个中断。
IntMasterEnable();允许总中断

TimerEnable(TIMER0_BASE, TIMER_A);
允许 TIMER0A 中断。

void Timer0IntHandler(void)
TIMER0 中断处理程序
当中断发生时，改变 LED 灯的状态

异常产生和分析
当注释掉主函数中一行：

SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);

时，debug 程序进板子，发现 led 灯并没有像正常的闪蓝灯，而是一直暗着，此时挂起程序发现，程序停在一个 FAULTISR 错误中断处理函数中，此处理函数有个 while（1）的死循环且拥有最高的 priority，所以程序一直停留在这。原因是程序禁用了 GPIO 的使能信号。