**实验一：GPIO驱动LED与定时器中断**

基本要求：使用GPIO控制四个LED灯，并按照下列顺序循环重复（各状态间的时间间隔大约为500ms）：

（1）LED0 亮（其他全灭）；

（2）LED1 亮（其他全灭）；

（3）LED2 亮（其他全灭）；

（4）LED3 亮（其他全灭）；

（5）LED2 亮（其他全灭）；

（6）LED1 亮（其他全灭）；

（7）LED0 亮（其他全灭）；

拓展1：在基本要求的基础上，加入以下顺序（各状态间的时间间隔大约为500ms）：

（8）LED0 LED1 亮（其他全灭）；

（9）LED0 LED1 LED2 亮（其他全灭）；

（10）全亮；

（11）LED0 灭（其他全亮）；

（12）LED1 灭（其他全亮）；

（13）LED2 灭（其他全亮）；

（14）LED3 灭（其他全亮）；

（15）LED2 灭（其他全亮）；

（16）LED1 灭（其他全亮）；

（17）LED0 灭（其他全亮）；

（18）全灭

拓展2：在基本要求和拓展1的基础上，要求各状态间的时间间隔利用定时器中断实现500ms的延时。

**实验二：UART中断接收字符按位显示**

基本要求：

配置 UART 为输入中断，通过上位机 sscom 接收用户输入的 LED 状态字（只需要一次接受一个char 类型的数据即可），发送到 UART，触发 UART 中断，用户读取状态字，实时将状态字转换为 LED3，LED2，LED1，LED0 的闪烁变化。LED 的闪烁根据以下描述：

假设 sscom 上用户输入了‘1’字符，UART 接受到的该字符为 char 类型，ASCII 码对应的 16 进制数为 0x31，即 0b00110001，占据了一个 BYTE 的内存空间。我们只有 4 个 LED 灯，无法一次显示 8bit，因此分两次显示：

第一次显示低 4bit，即 0b0001，此时 LED3，LED2，LED1 灭，LED0 亮；

第二次显示高 4bit，即 0b0011，此时 LED3，LED2 灭，LED1，LED0 亮。

两次显示切换间隔使用 delay 函数，延迟大概 0.5s 即可。

拓展1：

在基本要求的基础上使用 Timer 进行精确延迟。

拓展2：

在扩展要求一的基础上，修改程序，使得程序可以一次接受最多 100 个字符，并以此使用 LED显示其状态。此时 n 个字符的 LED 显示需要 n 秒。

**实验三：EPWM基础实验**

基本要求

EPWM配制：输出两路EPWM1A和EPWM1B，频率为15KHz，EPWM1A占空比为25%；EPWM1B与EPWM1A互补；设置死区为1us，以EPWM1A为上升沿、下降沿参考源；死区上升沿、下降沿均有输出，使用Active Low Complementary极性模式输出。

其他要求：TZ禁止，斩波模式禁止，中断源选择为EHRPWM\_ETSEL\_INTSEL\_TBCTREQUPRD，两次时间发生后触发一次EPWM中断。

**大作业 FFT变换**

实验要求

对一段离散数据进行FFT分析，并将其输出结果呈现在输出窗口中，以增加对算法的认识。

其具体要求如下：

1. 无示例程序

2. 熟悉并编写FFT算法

3. 自行生成至少2^n（n为自然数，n>=7）点的数据，并进行FFT分析

4. 输出结果显示在图像窗口