

DSP 实践课程

上海交通大学

可参考网站: <https://github.com/kangzhiheng/TMS320C6748/>

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。

参考链接:

<https://github.com/kangzhiheng/TMS320C6748/tree/master/%E5%AE%9E%E9%AA%8C%E5%8F%82%E8%80%83%E4%BB%A3%E7%A0%81/%E5%AE%9E%E9%AA%8C%E4%B8%80>

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。

参考链接：

<https://github.com/kangzhiheng/TMS320C6748/tree/master/%E5%AE%9E%E9%AA%8C%E5%8F%82%E8%80%83%E4%BB%A3%E7%A0%81/%E5%AE%9E%E9%AA%8C%E4%BA%8C>

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 1 μ s，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。

参考链接：

<https://github.com/kangzhiheng/TMS320C6748/tree/master/%E5%AE%9E%E9%AA%8C%E5%8F%82%E8%80%83%E4%BB%A3%E7%A0%81/%E5%AE%9E%E9%AA%8C%E4%B8%89>

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, $n \geq 7$) 点的数据, 并进行 FFT 分析
4. 输出结果显示在图像窗口