Hinweise:

Nutzen Sie für die Umsetzung die bestehenden Programm und Dateistrukturen.

- Timer-Initialisierungen in Datei "Timer.c "
- "Timer.h" um weitere Funktionsdeklarationen ergänzen
 - z.B. void timer0_init(void) void timer1_init(void) usw.
- Interrupt-Service-Routinen in Datei "isr.c"

Interrupt-Priorität =0 hat die höchste Priorität

Bearbeiten Sie folgende Aufgaben selbständig:

- 1. LED1 *Chip1768-Modul* blinkt mit 0,5 Hz
- -> Timer 1 Interrupt gesteuert, Interrupt Priorität 14
- LED4 *Chip1768-Modul* blinkt mit 4 Hz
- -> Timer 3 Interrupt gesteuert, Interrupt Priorität 13
- LED5 Experimentierbord TestBed blinkt mit 0,5 Hz
 LED4 Experimentierbord TestBed blinkt mit 4 Hz
- -> Timer ohne Interrupt gesteuert
- -> Timer 0 Interrupt gesteuert, Interrupt
 Priorität 15
- 3. **OPTION**: 4 LEDs blinken mit unterschiedlicher Frequenz; Timer Interrupt gesteuert
 - LED1 *Chip1768-Modul* blinkt mit 0,5 Hz
- -> Timer 0 Interrupt gesteuert, Interrupt Priorität 14
- LED2 *Chip1768-Modul* blinkt mit 1 Hz
- -> Timer 1 Interrupt gesteuert, Interrupt Priorität 13
- LED1 *Chip1768-Modul* blinkt mit 2 Hz
- -> Timer 2 Interrupt gesteuert, Interrupt Priorität 12
- LED4 *Chip1768-Modul* blinkt mit 4 Hz
- -> Timer 3 Interrupt gesteuert, Interrupt Priorität 11