

# Lab 4: François Soulié

---

Link to your Digital-electronics-2 GitHub repository:

<https://github.com/francois07/digital-electronics-2>

## Overflow times

1. Complete table with overflow times.

Module	Number of bits	1	8	32	64	128	256	1024
Timer/Counter0	8	16u	128u	--	1m	--	4m	16m
Timer/Counter1	16	4m	32m	--	262m	--	1s	4s
Timer/Counter2	8	16u	128u	512u	1m	2m	4m	16m

## Timer library

1. In your words, describe the difference between common C function and interrupt service routine.

- Function
- Interrupt service routine

The common C function runs at the start of the program while the interrupt service routine waits for a specific event to run.

2. Part of the header file listing with syntax highlighting, which defines settings for Timer/Counter0:

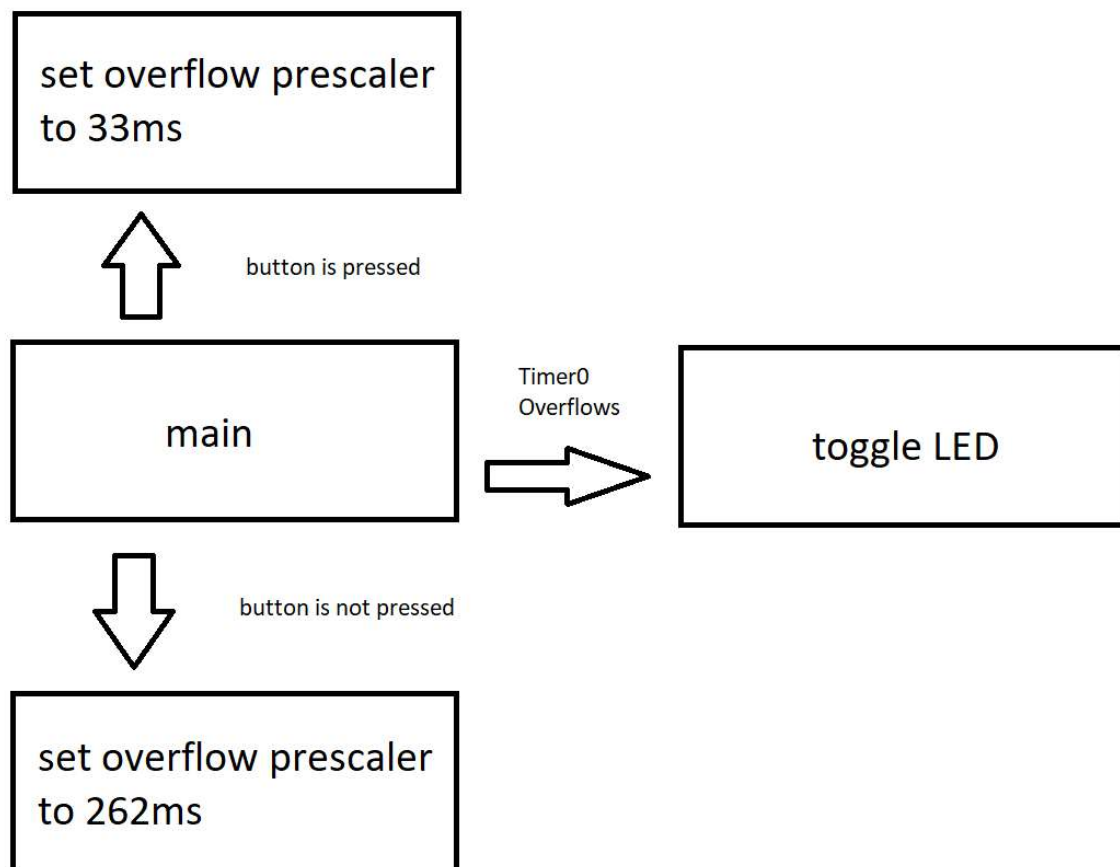
```
/**
 * @name Definitions of Timer/Counter0
 * @note F_CPU = 16 MHz
 */
/** @brief Stop timer, prescaler 000 --> STOP */
#define TIM0_stop() TCCR0B &= ~((1<<CS02) | (1<<CS01) | (1<<CS00));
/** @brief Set overflow 16us, prescaler 001 --> 1 */
#define TIM0_overflow_16us() TCCR0B &= ~((1<<CS02) | (1<<CS01)); TCCR0B |= (1<<CS00)
/** @brief Set overflow 128us, prescaler 010 --> 8 */
#define TIM0_overflow_128us() TCCR0B &= ~((1<<CS02) | (1<<CS00)); TCCR0B |= (1<<CS01)
/** @brief Set overflow 1ms, prescaler 011 --> 64 */
#define TIM0_overflow_1ms() TCCR0B &= ~(1<<CS02); TCCR0B |= (1<<CS01) | (1<<CS10);
/** @brief Set overflow 4ms, prescaler 100 --> 256 */
#define TIM0_overflow_4ms() TCCR0B &= ~((1<<CS01) | (1<<CS00)); TCCR0B |= (1<<CS02);
/** @brief Set overflow 16ms, prescaler // 101 --> 1024 */
#define TIM0_overflow_16ms() TCCR0B &= ~(1<<CS01); TCCR0B |= (1<<CS02) | (1<<CS00);
/** @brief Enable overflow interrupt, 1 --> enable */
#define TIM0_overflow_interrupt_enable() TIMSK0 |= (1<<TOIE0);
/** @brief Disable overflow interrupt, 0 --> disable */
```

```

/ enable disable overflow interrupt, 0 / enable /
#define TIM0_overflow_interrupt_disable() TMSK0 &= ~(1<<TOIE0);

```

3. Flowchart figure for function `main()` and interrupt service routine `ISR(TIMER1_OVF_vect)` of application that ensures the flashing of one LED in the timer interruption. When the button is pressed, the blinking is faster, when the button is released, it is slower. Use only a timer overflow and not a delay library. The image can be drawn on a computer or by hand. Use clear descriptions of the individual steps of the algorithms.



## Knight Rider

1. Scheme of Knight Rider application with four LEDs and a push button, connected according to Multi-function shield. Connect AVR device, LEDs, resistors, push button, and supply voltage. The image can be drawn on a computer or by hand. Always name all components and their values.

