Lab 4: Pavel Vaněk

Link to my `Digital-electronics-2` GitHub repository:

https://github.com/Bobik77/Digital-electronics-2

Overflow times

1. Complete table with overflow times.

Module	Number of bits	1	8	32	64	128	256	102
Timer/Counter0	8	16u	128u	--	1024u	--	4096u	1638
Timer/Counter1	16	4096u	32,768m	--	262,14m	--	1,049s	4,19
Timer/Counter2	8	16u	128u	512u	1024u	2048u	4096u	1638

Timer library

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

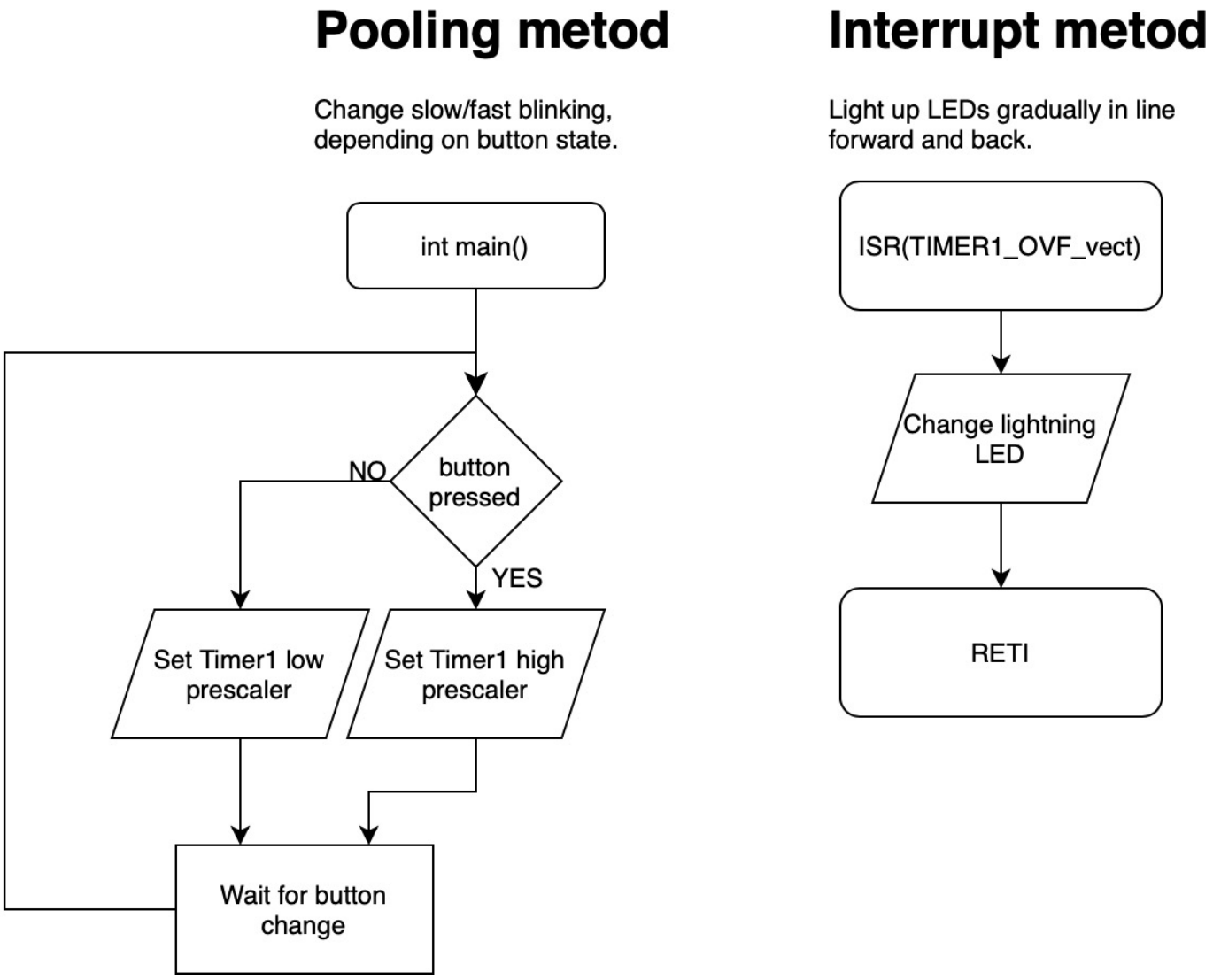
     Je volána softwearově, můžeme jí předat určitý argument může vracet výstupní hodnotu.
   - Interrupt service routine:

     Je volána hardwarově při vzniku přerušení (např přetečení timeru). Nemá žádné vstupní argumenty výstupní hodnoty.

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

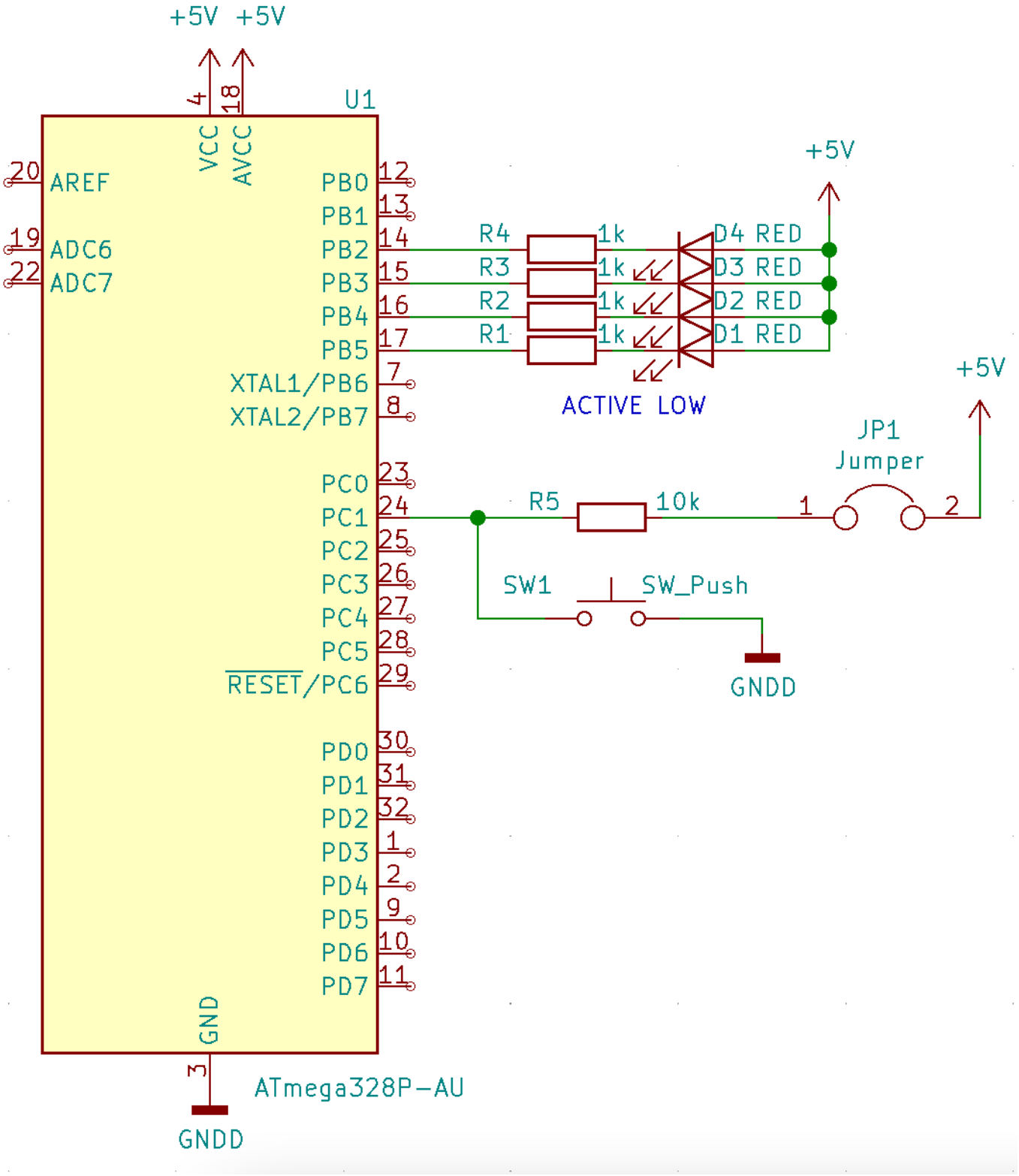
```
/**
 * @name  Definitions for 8-bit Timer/Counter0
 * @note  t_OVF = 1/F_CPU * prescaler * 2^n where n = 8, F_CPU = 16 MHz
 */
/** @brief Stop timer, prescaler 000 --> STOP */
#define TIM0_stop()           TCCR0B &= ~((1<<CS02) | (1<<CS01) | (1<<CS00));
/** @brief Set overflow 16us, prescaler 001 --> 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<<CS00);
/** @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 */
#define TIM0_overflow_interrupt_disable() TIMSK0 &= ~(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.



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!


+5V
+5V
4
18
U1
VCC
AVCC
20
AREF
19
ADC6
22
ADC7
PB0
PB1
PB2
PB3
PB4
PB5
XTAL1/PB6
XTAL2/PB7
PC0
PC1
PC2
PC3
PC4
PC5
RESET/PC6
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
12
13
14
15
16
17
7
8
23
24
25
26
27
28
29
30
31
32
1
2
9
10
11
GND
3
ATmega328P-AU
GNDD
+5V
R4
R3
R2
R1
1k
1k
1k
1k
D4 RED
D3 RED
D2 RED
D1 RED
ACTIVE LOW
+5V
JP1
Jumper
R5
10k
1
2
SW1
SW_Push
GNDD