UTAN25 / Digital-electronics-2 Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main

Digital-electronics-2 / Labs / 04-interrupts / Assignment.md

UTAN25 Update Assignment.md

1 contributor

Lab 4: Unai Telletxea

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

https://github.com/UTAN25/Digital-electronics-2

Overflow times

1. Complete table with overflow times.

Module	Number of bits	1	8	32	64	128	256
Timer/Counter0	8	16us	128us	--	1ms	--	4ms
Timer/Counter1	16	4ms	33ms	--	262ms	--	1s
Timer/Counter2	8	16us	128us	512us	1ms	2ms	4ms

Timer library

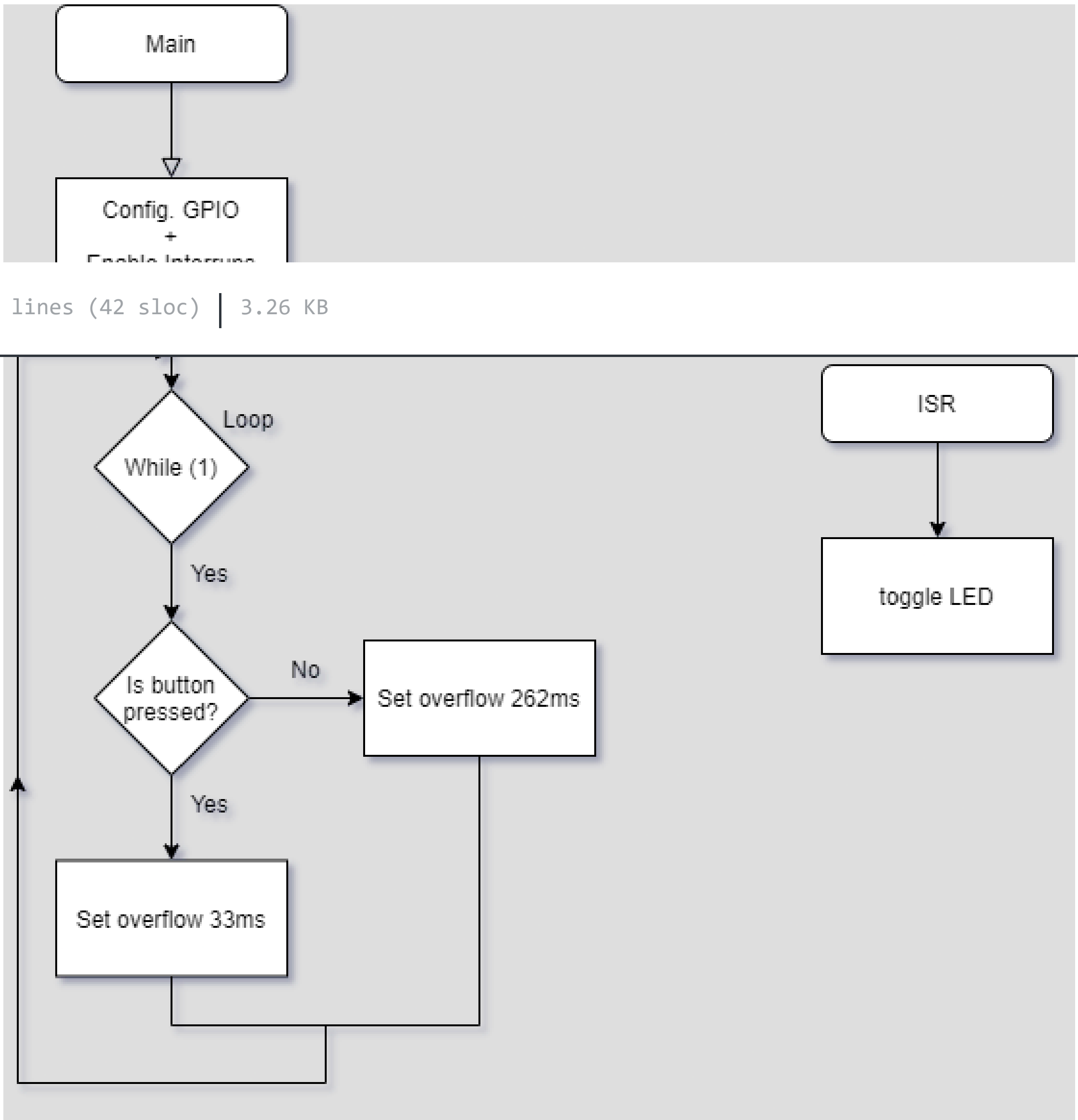
1. In your words, describe the difference between common C function and interrupt service routine.

   - Functions have to be called every time we need to use them in the program. In a secuential way, They have no special priority and some cases like checking a botton, depending on the timing the information might be lost (if the button is pressed when the program is not looking for it).
   - Interrupt service routine, when activated, they can interrupt the execution of the code and start processing the routine. In this way It is possible to give priority to certain tasks and leave the others for later.

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

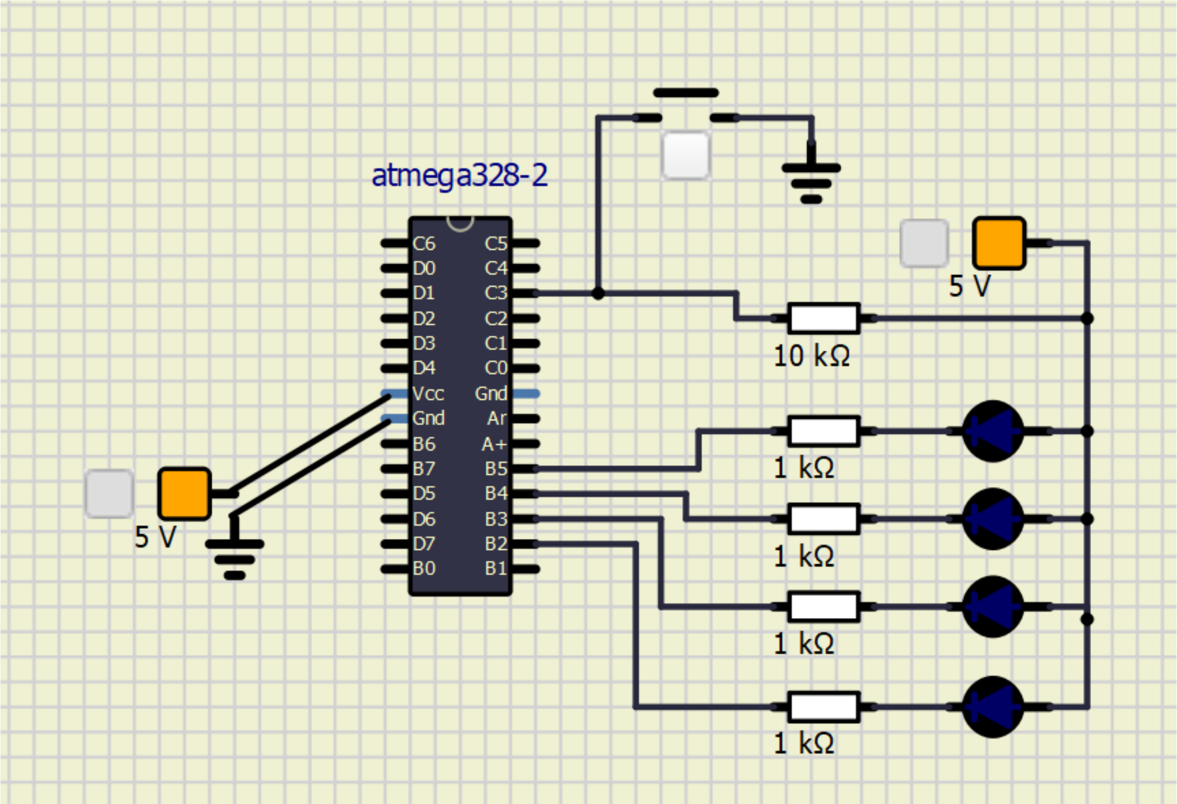
```
/**
 * @name  Definitions of Timer/Counter0
 * @note  F_CPU = 16 MHz
 */
/** @brief Stop timer, prescaler 000 --> STOP */
#define TIM0_stop()           TCCR0B &= ~((1<<CS02) | (1<<CS01) | (1<<CS00));
/** @brief Set overflow, prescaler 001 --> 1 */
#define TIM0_overflow_16us()   TCCR0B &= ~((1<<CS02) | (1<<CS01)); TCCR0B |= (1<<
/** @brief Set overflow, prescaler 010 --> 8 */
#define TIM0_overflow_128us()  TCCR0B &= ~((1<<CS02) | (1<<CS00)); TCCR0B |= (1<<
/** @brief Set overflow, prescaler 011 --> 64 */
#define TIM0_overflow_1ms() TCCR0B &= ~(1<<CS02); TCCR0B |= (1<<CS01) | (1<<CS00)
/** @brief Set overflow, prescaler 100 --> 256 */
#define TIM0_overflow_4ms()    TCCR0B &= ~((1<<CS01) | (1<<CS00)); TCCR0B |= (1<<
/** @brief Set overflow, prescaler // 101 --> 1024 */
#define TIM0_overflow_16ms()    TCCR0B &= ~(1<<CS01); TCCR0B |= (1<<CS02) | (1<<C
/** @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!


atmega328-2
C6
D0
D1
D2
D3
D4
Vcc
Gnd
B6
B7
D5
D6
D7
B0
C5
C4
C3
C2
C1
C0
Gnd
Ar
A+
B5
B4
B3
B2
B1
5 V
5 V
10 kΩ
1 kΩ
1 kΩ
1 kΩ
1 kΩ