Lab 4:Dogancan Gurbuz

https://github.com/DogancanG/Digital-electronics-2/tree/main/Labs/04-interrupts/timer

Overflow times

Module	Number of bits	1	8	32	64	128	256	1024
Timer/Counter0	8	16u	128u	--	1ms	--	4.1ms	16ms
Timer/Counter1	16	4.1ms	33ms	--	262ms	--	1.04s	4.2s
Timer/Counter2	8	16u	128u	512u	1ms	2ms	4.1ms	16ms

Timer library

1. An Interrupt is a feature of the processor hardware; eg, on an 8051, and interrupt can occur when the UART receives a character. A Function is a construct of the 'C' programming language - it represents a piece of executable code that can be "called" by other parts of the program. (other languages have similar constructs) A special case of a function is when it is used to service an interrupt - commonly known as an Interrupt Service Routine or "ISR"

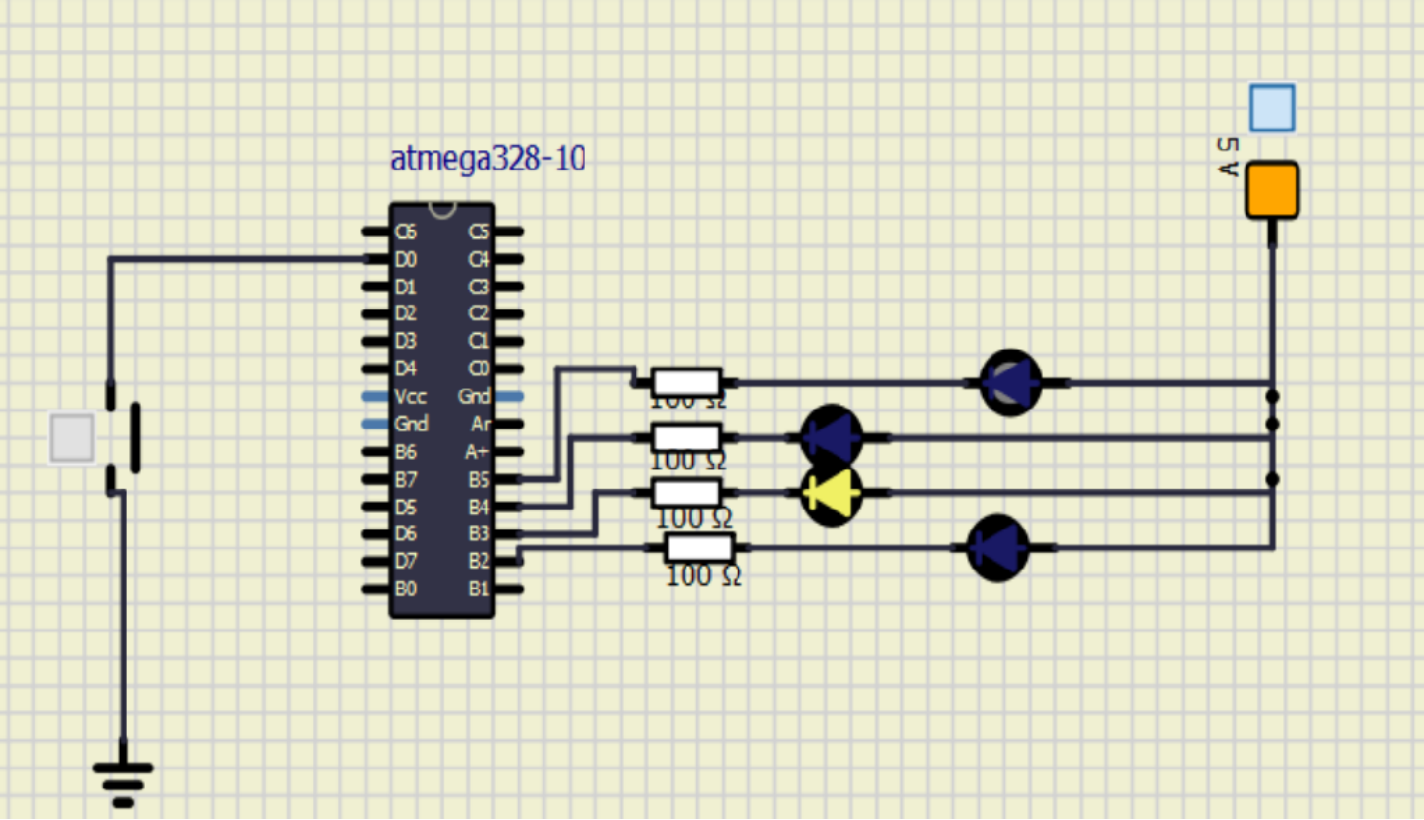
2.

```
/**
 * @name  Definitions of Timer/Counter0
 * @note  F_CPU = 16 MHz
 */
// WRITE YOUR CODE HERE
#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.

```
 #define LED_D1  PB5
#define LED_D2  PB4
#define LED_D3  PB3
#define LED_D4  PB2
#define BUTTON_S1 PC1

/* Includes ------------------------------------------------------------*/
#include <avr/io.h>         // AVR device-specific IO definitions
#include <avr/interrupt.h>  // Interrupts standard C library for AVR-GCC
#include "gpio.h"           // GPIO library for AVR-GCC
#include "timer.h"          // Timer library for AVR-GCC

/* Function definitions ----------------------------------------------*/
/**********************************************************************
 * Function: Main function where the program execution begins
 * Purpose:  Toggle one LED on the Multi-function shield using
             the internal 8- or 16-bit Timer/Counter.
 * Returns:  none
 **********************************************************************/
int main(void)
{
    // Configuration of LED(s) at port B
    GPIO_config_output(&DDRB, LED_D1);
    GPIO_write_low(&PORTB, LED_D1);

    // Configuration of 16-bit Timer/Counter1 for LED blinking
    // Set the overflow prescaler to 262 ms and enable interrupt
    TIM1_overflow_262ms();
    TIM1_overflow_interrupt_enable();

    // push button at port C
    GPIO_config_input_nopull(&DDRC, BUTTON_S1);


    // Enables interrupts by setting the global interrupt mask
    sei();

    // Infinite loop
    while (1)
    {
        /* Empty loop. All subsequent operations are performed exclusively
         * inside interrupt service routines ISRs */
        // check push button on state
        if(GPIO_read(&PINC, BUTTON_S1) == 0) // push button pressed
        {
            TIM1_overflow_33ms();

        }
        else                                    // push button is released
        {
            TIM1_overflow_262ms();
        }
    }

    // Will never reach this
    return 0;
}

/* Interrupt service routines ----------------------------------------*/
/**********************************************************************
 * Function: Timer/Counter1 overflow interrupt
 * Purpose:  Toggle D1 LED on Multi-function shield.
 **********************************************************************/
ISR(TIMER1_OVF_vect)
{

    // WRITE YOUR CODE HERE
    GPIO_toggle(&PORTB, LED_D1);

}
```

Knight Rider


atmega328-10
5 V
100 Ω
100 Ω
100 Ω