# Lab 5: Martin Kousal

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

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

7-segment library

1. In your words, describe the difference between Common Cathode and Common Anode 7-segment display.

   - **CC SSD** - displays with common cathode have shared cathode between all segments, single segments are turned on with logical 1 (positive voltage). If we wants to turn one digit on, we must ground this cathode.
   - **CA SSD** - displays with common anode have shared anode between all segments, single segments are turned on with logical 0 (grounded). If we wants to turn digit on, we must provide supply voltage to this anode.

2. Code listing with syntax highlighting of two interrupt service routines (`TIMER0_OVF_vect`, `TIMER0_OVF_vect`) from counter application with at least two digits, ie. values from 00 to 59:

```c
volatile uint8_t cnt[] = {0,0,0,0};

/* Interrupt service routines ---------------------------------------*/
/***********************************************************************
 * Function: Timer/Counter1 overflow interrupt
 * Purpose:  Increment decimal counter value and display it on SSD.
 **********************************************************************/
ISR(TIMER1_OVF_vect)
{
    cnt[0]++;

    if (cnt[0] == 10)
    {
        cnt[0] = 0;
        cnt[1]++;
    }
    if (cnt[1] == 6)
    {
        cnt[1] = 0;
        cnt[2]++;
    }
    if (cnt[2] == 10)
    {
        cnt[2] = 0;
        cnt[3]++;
    }
    if (cnt[3] == 6)
    {
```
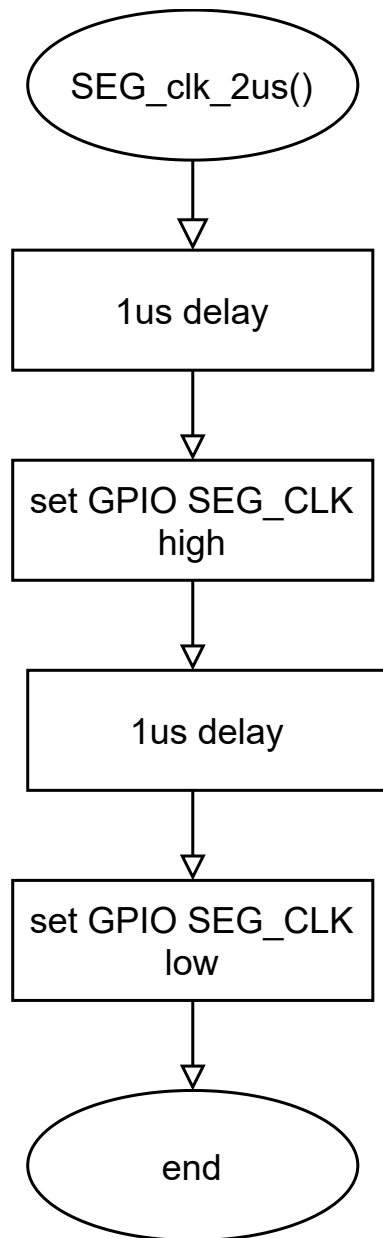
```
            cnt[3] = 0;
        }
    }
```

```c
/*********************************************************************
 * Function: Timer/Counter0 overflow interrupt
 * Purpose:  Display tens and units of a counter at SSD.
 *********************************************************************/
ISR(TIMER0_OVF_vect)
{
    static uint8_t pos = 0;
    static int8_t dir = 1;
    if (pos != 2)
        SEG_update_shift_regs(cnt[pos], pos, 0);
    else
        SEG_update_shift_regs(cnt[pos], pos, 1);
    pos += dir;
    if (pos == 3)
        dir = -1;
    if (pos == 0)
        dir = 1;
}
```

3. Flowchart figure for function `SEG_clk_2us()` which generates one clock period on `SEG_CLK` pin with a duration of 2 us. The image can be drawn on a computer or by hand. Use clear descriptions of the individual steps of the algorithms.

```
        SEG_clk_2us()
              │
              ▼
        ┌──────────────┐
        │   1us delay  │
        └──────────────┘
              │
              ▼
        ┌──────────────┐
        │ set GPIO SEG_CLK │
        │      high        │
        └──────────────┘
              │
              ▼
        ┌──────────────┐
        │   1us delay  │
        └──────────────┘
              │
              ▼
        ┌──────────────┐
        │ set GPIO SEG_CLK │
        │       low        │
        └──────────────┘
              │
              ▼
            ( end )
```

## Kitchen alarm

Consider a kitchen alarm with a 7-segment display, one LED and three push buttons: start, +1 minute, -1 minute. Use the +1/-1 minute buttons to increment/decrement the timer value. After pressing the Start button, the countdown starts. The countdown value is shown on the display in the form of mm.ss (minutes.seconds). At the end of the countdown, the LED will start blinking.

1. Scheme of kitchen alarm; do not forget the supply voltage. The image can be drawn on a computer or by hand. Always name all components and their values.

+5V

Arduino_UNO_R3

VIN
3V3
+5V
A1

D0/RX
D1/TX
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13

RESET
IOREF
AREF

A0
A1
A2
A3
SDA/A4
SCL/A5
SDA/A4
SCL/A5

GND
GND
GND

GND

+5V

D1

R4
1k

SRCLK
RCLK
SER

R1
10k

R2
10k

R3
10k SW1

S1-A1
SW2

S1-A1
SW3

S1-A1

GND

JP1
Jumper

+5V

SER

SRCLK
RCLK

+5V

U2
74HC595

SER
VCC
SRCLK
SRCLR
RCLK
OE
GND

QA
QB
QC
QD
QE
QF
QG
QH
QH'

GND

U1
CA56-12EWA

a
b
c
d
e
f
g
DPX

f  a  b
g
e  d  c

DP1   DP2   DP3   DP4

CA1
CA2
CA3
CA4

+5V

U3
74HC595

SER
VCC
SRCLK
SRCLR
RCLK
OE
GND

QA
QB
QC
QD
QE
QF
QG
QH
QH'

GND

SRCLK

RCLK

GND