```c
/*
 * ArbkravEkstra.c
 *
 * Created: 22/04/2023 14:36:52
 * Author : NTNU
 */
#define F_CPU 4000000UL
#define PERIOD_VALUE (0x01a0)
#define DUTY_CYCLE (0x00D0)

#include "omsfah_usart.h"
#include <stdbool.h>
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

void ADC0_init(void);
uint16_t ADC0_read(void);

volatile bool state=1; //Volatile due to being modified by an interrupt
uint8_t dutyCycle = 0;
uint8_t dutyCycleStep = 1;
uint16_t adcVal;
uint16_t servoVal;

ISR(PORTB_PORT_vect){
    _delay_ms(10); //debounce
    state = !state;
    if (state)
    {
        printf("Systemet er i LED modus\r\n");
    }
    else
    {
        printf("Systemet er i servo modus\r\n");
    }
    PORTB.INTFLAGS = 0xff;
}

void TCA0_init(void)
{
    PORTMUX.TCAROUTEA = PORTMUX_TCA0_PORTB_gc;
    TCA0.SPLIT.CTRLD = TCA_SPLIT_SPLITM_bm;
    //TCA.SPLIT.CTRLB =
    // enable W00 and W03
    TCA0.SPLIT.CTRLB = TCA_SPLIT_HCMP0EN_bm | TCA_SPLIT_LCMP0EN_bm;
    TCA0.SPLIT.HPER = DUTY_CYCLE;
    TCA0.SPLIT.CTRLA = TCA_SPLIT_CLKSEL_DIV4_gc | TCA_SPLIT_ENABLE_bm;
}

void ADC0_init(void)
{
    /* Disable digital input buffer */
    PORTD.PIN6CTRL &= ~PORT_ISC_gm;
    PORTD.PIN6CTRL |= PORT_ISC_INPUT_DISABLE_gc;
    /* Disable pull-up resistor */
```

```
57      PORTD.PIN6CTRL &= ~PORT_PULLUPEN_bm;
58      ADC0.CTRLC = ADC_PRESC_DIV4_gc; /* CLK_PER divided by 4 */
59      VREF.ADC0REF = VREF_REFSEL_VDD_gc; /* Internal reference */
60      ADC0.CTRLA = ADC_ENABLE_bm /* ADC Enable: enabled */
61      | ADC_RESSEL_10BIT_gc; /* 10-bit mode */
62      /* Select ADC channel */
63      ADC0.MUXPOS = ADC_MUXPOS_AIN6_gc;
64  }
65  uint16_t ADC0_read(void)
66  {
67      /* Start ADC conversion */
68      ADC0.COMMAND = ADC_STCONV_bm;
69      /* Wait until ADC conversion done */
70      while ( !(ADC0.INTFLAGS & ADC_RESRDY_bm) )
71      {
72          ;
73      }
74      /* Clear the interrupt flag by writing 1: */
75      ADC0.INTFLAGS = ADC_RESRDY_bm;
76      return ADC0.RES;
77  }
78
79  int main(void)
80  {
81      PORTB.DIR |= PIN0_bm;
82      PORTB.DIR |= PIN3_bm;
83      PORTB.PIN2CTRL = PORT_PULLUPEN_bm | PORT_ISC_FALLING_gc; //Enable pins
84      sei();
85      USART3_init();
86      TCA0_init();
87      ADC0_init();
88      _delay_ms(10);
89      stdout = &USART_stream;
90      while (1)
91      {
92          if (state)
93          {
94              TCA0.SPLIT.HCMP0=dutyCycle;
95              _delay_ms(5);
96              dutyCycle = adcVal/4; //Worked until SERVO was connected, fear
                  that IC is fried, lost also USART later on, tried to reflash
                  chip directly
97          }
98          else
99          {
100
101             TCA0.SPLIT.LCMP0=servoVal; //For some reason theres interference
                  between ports
102             _delay_ms(5);
103             servoVal=adcVal+750; //Stolen from SERVO guide from Carl Richard

104         }
105
106     }
107 }
```