

# Zadaní závěrečného projektu do MIT

## Název: Řízení větrání a závlahy

Ovládání dvou ventilátorů 1) odtah

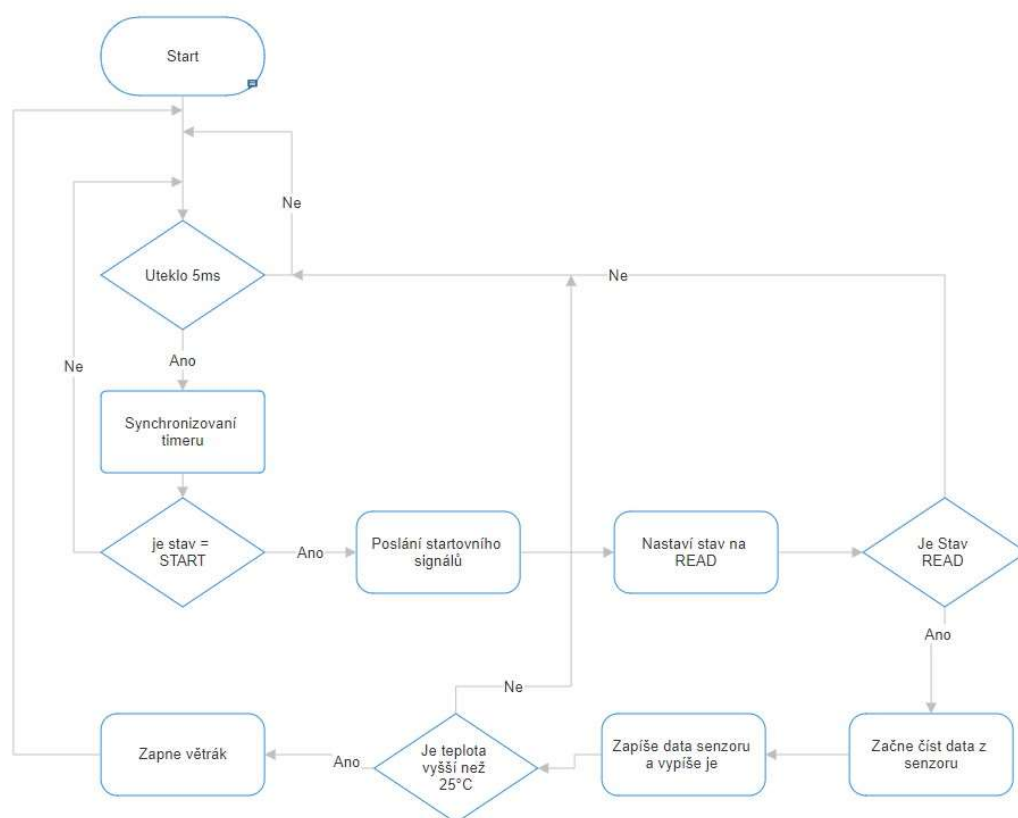
2) přisávání

Ovládání systému závlahy kombinací informací z teplotního čidla 2 a vlhkoměru

- a) Zapnutí ventilátorů
- b) Zapnutí čerpadla

Potřeby: 2x ventilátor, 1x čerpadlo, 2x teplotní čidlo ,1x vlhkoměr, 1x relé

Vývojový Diagram:



## Inicializace vstupů, Zapnutí a nastavení interruptů ,Nastavení časovače

Trigger pin nastavený na OD (Open Drain), HIZ (High Impedance)

HIZ – toto nastavení umožňuje, aby pin byl „odpojen“ a tím pádem nemá téměř žádný vliv

OD – je nastaven kvůli ochraně zařízení

```
void init(void) {
    CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1); // taktovani MCU na
16MHz

    // Piny
    GPIO_Init(GPIOE, GPIO_PIN_0, GPIO_MODE_IN_PU_IT); // DATA in PIN
    GPIO_Init(GPIOG, GPIO_PIN_0, GPIO_MODE_OUT_OD_HIZ_SLOW); // Trigger
    GPIO_Init(BTN_PORT, BTN_PIN, GPIO_MODE_IN_FL_NO_IT); // Tlačítko
    GPIO_Init(GPIOD,GPIO_PIN_3,GPIO_MODE_OUT_PP_LOW_SLOW); //Signalizace

    // Interrupty
    EXTI_SetExtIntSensitivity(EXTI_PORT_GPIOE,
EXTI_SENSITIVITY_RISE_FALL);
    ITC_SetSoftwarePriority(ITC_IRQ_PORTE, ITC_PRIORITYLEVEL_1);
    enableInterrupts();

    // Časovač
    TIM4_TimeBaseInit(TIM4_PRESCALER_16, 0xffff);
    TIM4_Cmd(ENABLE);

    // UART
    init_uart1();
    init_milis();
}
```

## Hlavní část kódu

```
int main(void) {
    uint32_t time = 0;
    uint32_t poslední = 0;
    uint64_t j = 1LL << 39;
    uint8_t i = 0;

    init();
    printf("UART test ");
    while (1) {
        if ((milis() - time) > 5000 ) {
            time = milis();
            TIM4_GetCounter();
            stav = START;
        }

        if (stav == NONE) {
            poslední = milis();
        } else if (stav == START) {
            if (milis() - poslední < 19) {
                GPIO_WriteLow(GPIOG, GPIO_PIN_0);
            } else {
                poslední = milis();
                TIM4_SetCounter(0);
                lcounter = 0;
                data = 0LL;
                GPIO_WriteHigh(GPIOG, GPIO_PIN_0);
                stav = READ;
            }
        } else if (stav == READ) {
            while (j) {
                if (data & j) {
                    printf("1");
                } else {
                    printf("0");
                }
                if (++i % 8 == 0) {
                    printf(" ");
                }
                j >>= 1;
            }
            printf("\n");
        }
    }
}
```

```

uint8_t temp_L = data >> 8;
uint8_t temp_H = data >> 16;
uint8_t RH_L = data >> 24;
uint8_t RH_H = data >> 32;

uint8_t kontrola = data;

printf("data: 0x %02X %02X %02X %02X\n", RH_H, RH_L,
temp_H, temp_L);
printf("vlhkost: %d %, teplota: %d.%d °C\n", RH_H,
temp_H, temp_L);
stav = NONE;
if (temp_H >25){
    GPIO_WriteHigh(GPIOD,GPIO_PIN_3);
} else {
    GPIO_WriteLow(GPIOD,GPIO_PIN_3);
}
}
return 0;
}

```

Na začátku máme nastavení proměnných

- Co 5ms se pošle signál který zapne senzor
- Stav == NONE =>
  
- Stav == START
  - Pošle startovní signál (LOW)
  - Po uplynutí potřebného času se vrátí na HIGH
- STAV READ

Načtou se data(binární) následně se rozdělí na teplotu a vlhkost(8/8 bitů), nakonec se vytisknou, + podmínka pro spuštění.

Schéma:

