

ZÁVĚREČNÝ PROJEKT MIT

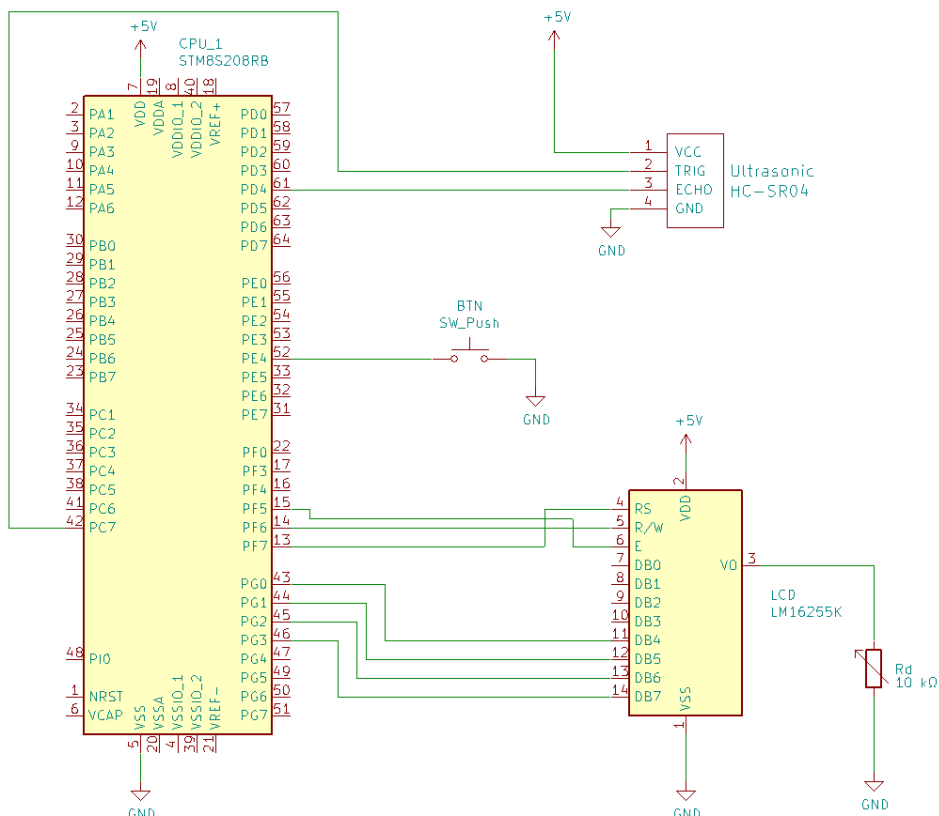
Obsah

SCHÉMA ZAPOJENÍ	2
SLOVNÍ POPIS ZAPOJENÍ	2
BLOKOVÉ SCHÉMA ZAPOJENÍ	2
SLOVNÍ POPIS PŘÍPRAVKŮ	3
SLOVNÍ POPIS FUNKCE PROGRAMU	3
VÝVOJOVÝ DIAGRAM PROGRAMU	4
ZÁVĚR	5
UKÁZKA PROGRAMU	6

Poř. č. 14	Příjmení a jméno PŘIKRYL Jan		Třída 4B	Skupina 1	Školní rok 2021/22
Datum zpracování 14. 1. 2022		Datum odevzdání 17. 1. 2022	Počet listů 7	Textová část	Klasifikace Obhajoba Funkčnost
Protokol o měření obsahuje:			teoretický úvod schéma použité přístroje postup měření	tabulky příklad výpočtu grafy závěr	

SCHEMA ZAPOJENÍ

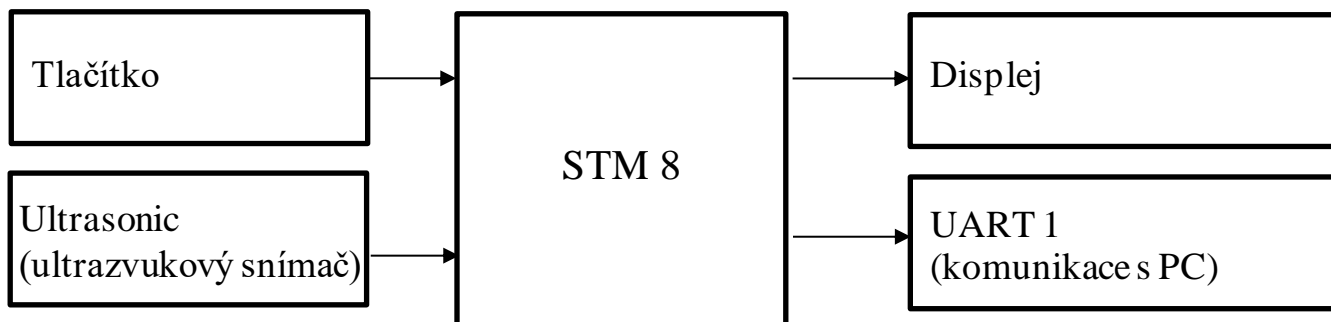
Schéma č.1: Zapojení snímače a displeje na STM8



SLOVNÍ POPIS ZAPOJENÍ

Snímač polohy je připojen k STM8 přes piny PD4 (ECHO - vstupní kanál), pin PC7 (TRIG – výstup) a napájení (+5V a GND). Displej je napájen pěti volty (pin VDD), uzemněn VSS. Data jsou přenášena čtyř vodičově (což je úspornější varianta). V0 přes potenciometr na zem (slouží k regulaci jasu, pokud bude přímo uzemněn, displej bude fungovat, ale bude špatně čitelný). RS na pin PF7, RW na PF6, E na PF5, D4 na PG0, D5 na PG1, D6 na PG2, D7 na PG3. Jako tlačítko je použito integrované tlačítko na desce.

BLOKOVÉ SCHEMA ZAPOJENÍ



SLOVNÍ POPIS PŘÍPRAVKŮ

Ultrasonic je modul ultrazvukového senzoru vzdálenosti pro mikrokontroléry (měří vzdálenost pomocí zvukové vlny). Jeho rozsah se pohybuje ve vzdálenosti od 2 cm do 400 cm, přesnost měření je 3 mm. Provozní napětí je 5 V, proud 2 mA. Princip činnosti spočívá v tom, že na pin TRIG je přiveden napěťový impuls (vysoký 5 V) po dobu 10 μs. Šířka impulsu jeho výstupu je závislá na vzdálenosti překážky od snímače. Z této hodnoty je možno vypočítat dobu mezi náběžnou a sestupnou hranou. Vzdálenost v centimetrech je dána vzorcem: $(čas\ v\ \mu s * 340) / 20000$.

LCD displej (neboli displej z tekutých krystalů) je zobrazovací zařízení s nízkou spotřebou a proto se hodí do zařízení na naše aplikace. Piny jsou u většiny displejů stejné. Displej má celkem 16 pinů, lze jej ale zapojit úsporněji. Displej je napájen 5 V. Má 2 řádky o 16 znacích. Běžně se používá na automatech, kopírákách či průmyslových zařízeních.

UART komunikace (Universal Synchronous / Asynchronous Receiver and Transmitter - Synchronní / asynchronní sériové rozhraní) je sériová komunikace využívaná pro komunikaci řadiče s počítačem.

SLOVNÍ POPIS FUNKCE PROGRAMU

Nejprve se importují všechny potřebné knihovny (to jsou: assert.h, delay.h, millis.h, stdio.h, stm8_hd44780.h). Následuje založení maker pro piny, porty a stavy tlačítka a snímače (makra pro displej jsou v knihovně stm8_hd44780.h).

Poté se vytvoří funkce pro UART komunikaci, inicializaci (taktování procesoru na 16 MHz, rozběhnutí časovače millis, povolení komunikace s počítačem a nastavení časovače TIM2 pro snímání polohy). Následuje založení enumu pro aktuální stav snímače.

Ve funkci main (hlavní funkce každého programu v jazyce C) jsou založeny proměnné pro počet měření, čas měření, vzdálenost, pro vypsání měření vzdálenosti a počtu pokusů. Zavolá se inicializační funkce a vypíše se uvítací hláška.

Poté následuje funkce while (1), což způsobí nekonečný cyklus opakování.

Opakovat se bude následující: Zjistí se aktuální stav snímače.

Pokud se stav bude rovnat startu a čas časovač bude větší než 444 ms (definovaná hodnota makrem), tak se spustí náběžná hrana TRIG a stav snímače bude čekej.

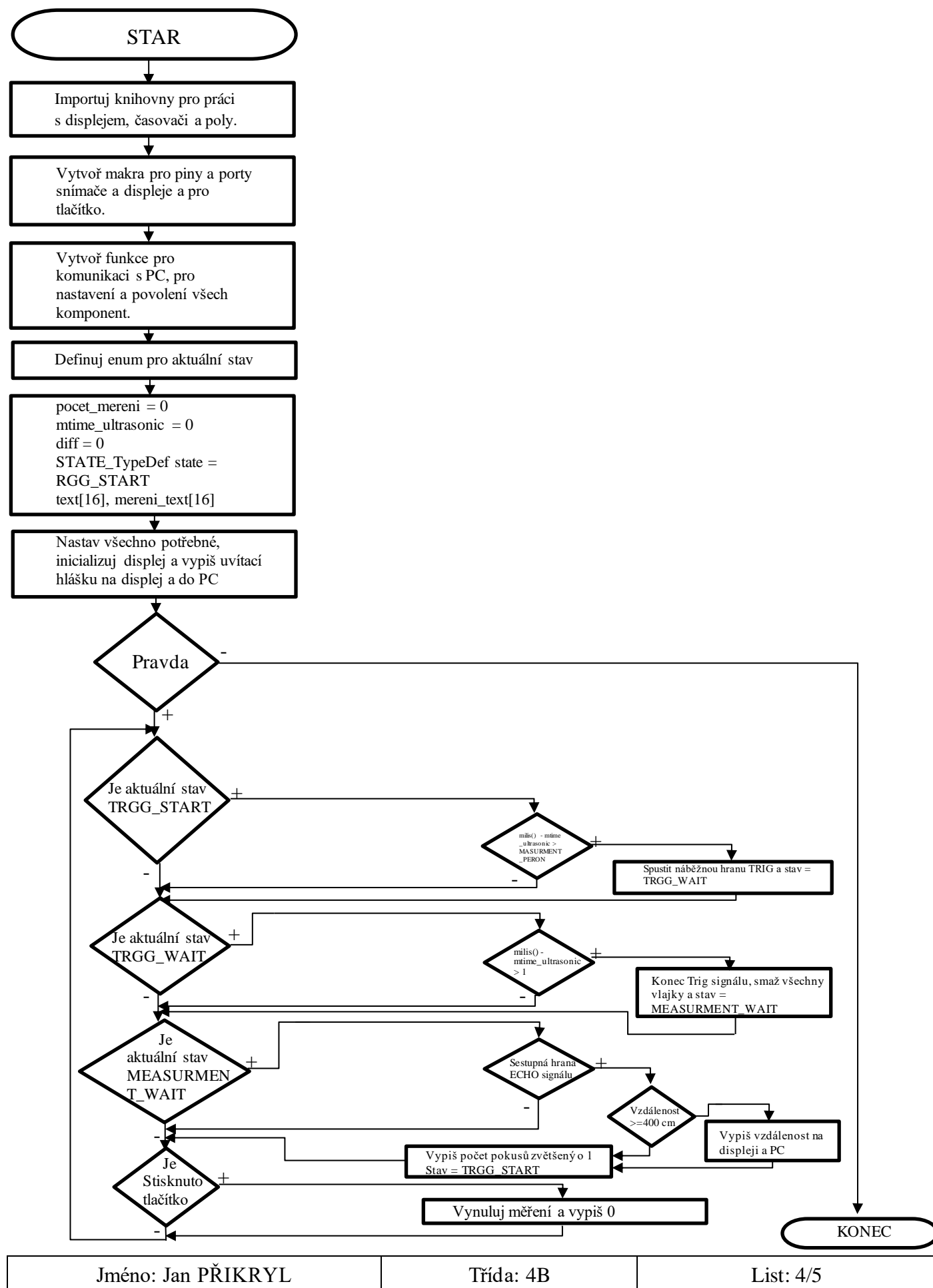
Pokud stav snímače bude čekej a čas snímače bude větší, než 1 ms, tak nastane konec Trig signálu, smažou se všechny vlajky a stav bude čekej na měření.

Pokud bude stav čekej na měření, detekuje se sestupná hrana signálu ECHO a v kladném případě se smažou vlajky CC1 a CC2. Délka impulsu se vypočítá odečtením TIM2 capture registru 2 od TIM2 capture registru 1. Z toho je možné vypočítat vzdálenost v centimetrech ($vzdalenost = (vzdalenost * 340) / 20000$). Pokud je vzdálenost menší, než maximální vzdálenost, tak ji vypíše na displej a pošle do počítače. Každopádně se vypíše počet pokusů zvýšený o 1. Stav se nastaví na start.

Nakonec se zkontroluje, jestli nebylo stisknuto tlačítko. V tomto případě se vynuluje počet pokusů a vypíše se 0 a prázdné místa (aby se nestalo, že na displeji zůstane nějaké číslo z minulého měření, a tak by za počtem měření zavazely další čísla).

Jméno: Jan PŘIKRYL	Třída: 4B	List: 3/5
--------------------	-----------	-----------

VÝVOJOVÝ DIAGRAM PROGRAMU



ZÁVĚR

Shrnutí a zhodnocení

- Vytvořil jsem program na měření vzdálenosti s STM8. Vzdálenost je měřena snímačem Ultrasonic HC-SR04. Výsledek je zobrazen na lcd displeji a může být zobrazen i v počítači, což se hodí v případě absence či poškození displeje).
- Program jsem psal v prostředí ST Visual Develop.

Výhody

- Výstup je zobrazen na dvou místech.
- Displej zobrazuje i počet pokusů měření.

Nevýhody

- Snímač vzdálenosti může změřit neplatnou hodnotu (hodnotu nad rozsah by ale nevypsal).

Co jsem se naučil a v čem to pro mě mělo přínos?

- Naučil jsem se používat alfanumerický displej, snímač (Ultrasonic HC-SR04) a nahrávání na GitHub.

Jméno: Jan PŘIKRYL	Třída: 4B	List: 5/5
--------------------	-----------	-----------

UKÁZKA PROGRAMU

```
#include "stm8s.h"
#include "assert.h"
#include "delay.h"
#include "milis.h"
#include "stdio.h"
#include "stm8_hd44780.h"
//Ultrasonic
#define TI1_PORT GPIOC
#define TI1_PIN GPIO_PIN_4
#define TRGG_PORT GPIOC
#define TRGG_PIN GPIO_PIN_7
#define TRGG_ON GPIO_WriteHigh(TRGG_PORT, TRGG_PIN);
#define TRGG_OFF GPIO_WriteLow(TRGG_PORT, TRGG_PIN);
#define TRGG_REVERSE GPIO_WriteReverse(TRGG_PORT, TRGG_PIN);
#define MASURMENT_PERON 444 // maximální celkový čas měření (ms)
#define MAXIMALNI_VZDALENOST 400
//Tlačítko
#define BTN_PORT GPIOE
#define BTN_PIN GPIO_PIN_4
#define BTN_PUSH (GPIO_ReadInputPin(BTN_PORT, BTN_PIN)==RESET)
//UART komunikace
char putchar (char c)
{
    UART1_SendData8(c);
    while (UART1_GetFlagStatus(UART1_FLAG_TXE) == RESET);
    return (c);
}
char getchar (void) //funkce čte(přijímá data) vstup z UART
{
    int c = 0;
    while (UART1_GetFlagStatus(UART1_FLAG_RXNE) == RESET);
    c = UART1_ReceiveData8();
    return (c);
}
void init_uart1(void) //Povolení UART1 (Využívane na komunikaci s PC)
{
    UART1_DeInit(); // smazat starou konfiguraci
    UART1_Init((uint32_t)115200, //Nová konfigurace
        UART1_WORDLENGTH_8D,
        UART1_STOPBITS_1,
        UART1_PARITY_NO,
        UART1_SYNCMODE_CLOCK_DISABLE,
        UART1_MODE_TXRX_ENABLE);
}
void setup(void)
{
    CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1); // taktovat MCU na 16MHz
    init_milis(); //Rozběhnutí časovace milis
    init_uart1(); //Povolení komunikace s PC
    GPIO_Init(BTN_PORT, BTN_PIN, GPIO_MODE_IN_FL_NO_IT); // Tlačítko jako vstup (vynulování)
    //Ultrasonic
    GPIO_Init(TRGG_PORT, TRGG_PIN, GPIO_MODE_OUT_PP_LOW_SLOW);
    GPIO_Init(TI1_PORT, TI1_PIN, GPIO_MODE_IN_FL_NO_IT); // kanál 1 jako vstup
    TIM2_TimeBaseInit(TIM2_PRESCALER_16, 0xFFFF);
    TIM2_Cmd(ENABLE);
    TIM2_ICInit(TIM2_CHANNEL_1, // nastavuji CH1 (CaptureRegister1)
        TIM2_ICPOLARITY_RISING, // náběžná hrana
        TIM2_ICSELECTION_DIRECTTI, // CaptureRegister1 bude ovládán z CH1
        TIM2_ICPSC_DIV1, // delička je vypnutá
        0 // vstupní filter je vypnutý
    );
    TIM2_ICInit(TIM2_CHANNEL_2, // nastavuji CH2 (CaptureRegister2)
        TIM2_ICPOLARITY_FALLING, // sestupná hrana
        TIM2_ICSELECTION_INDIRECTTI, // CaptureRegister2 bude ovládán z CH1
        TIM2_ICPSC_DIV1, // delička je vypnutá
        0 // vstupní filter je vypnutý
    );
}
```

```

typedef enum //Enum pro stavy snimace vzdalenosti
{
    TRGG_START,           // zahájení trigger impoulzu
    TRGG_WAIT,            // čekání na konec trigger impoulzu
    MEASUREMENT_WAIT      // čekání na dokončení měření
} STATE_TypeDef;
void main(void)
{
    uint32_t pocet_mereni = 0;
    uint32_t mtime_ultrasonic = 0;
    uint32_t vzdalenost;
    STATE_TypeDef state = TRGG_START;
    char text[16];
    char mereni_text[16];
    setup();
    printf("Start programu\r\n"); //Uvítací hláška
    lcd_init(); //Inicializace lcd displeje
    lcd_gotoxy(0,0);
    lcd_puts("Start programu"); //Uvítací hláška
    while (1) {
        switch (state) { //Stav snímače
            case TRGG_START:
                if (millis() - mtime_ultrasonic > MEASUREMENT_PERIOD) {
                    mtime_ultrasonic = millis();
                    TRGG_ON;
                    state = TRGG_WAIT;
                }
                break;
            case TRGG_WAIT:
                if (millis() - mtime_ultrasonic > 1) {
                    TRGG_OFF;
                    // smažu všechny vlajky
                    TIM2_ClearFlag(TIM2_FLAG_CC1);
                    TIM2_ClearFlag(TIM2_FLAG_CC2);
                    TIM2_ClearFlag(TIM2_FLAG_CC1OF);
                    TIM2_ClearFlag(TIM2_FLAG_CC2OF);
                    state = MEASUREMENT_WAIT;
                }
                break;
            case MEASUREMENT_WAIT:
                // detekují sestupnou hranu ECHO signálu
                if (TIM2_GetFlagStatus(TIM2_FLAG_CC2) == RESET) {
                    TIM2_ClearFlag(TIM2_FLAG_CC1); // smažu vlajku CC1
                    TIM2_ClearFlag(TIM2_FLAG_CC2); // smažu vlajku CC2
                    // délka impulzu v µs
                    vzdalenost = (TIM2_GetCapture2() - TIM2_GetCapture1());
                    //Vypocet a vypsání vzdalenosti na PC a displej
                    vzdalenost = (vzdalenost * 340) / 20000; // FixPoint prepocet na cm
                    if (vzdalenost <= 400) {
                        printf("Vzdalenost: %lu cm\r\n", vzdalenost);
                        sprintf(text, "Vzdalenost:%lu cm", vzdalenost);
                        lcd_gotoxy(0,0);
                        lcd_puts(text);
                    }
                    //Vypsání počtu pokusů
                    sprintf(mereni_text, "Zmereno:%lu", pocet_mereni++);
                    lcd_gotoxy(0,1);
                    lcd_puts(mereni_text);
                    state = TRGG_START;
                }
                break;
            default:
                state = TRGG_START;
        }
        if (BTN_PUSH) { //Pokud je stisknuto tlačítko - vynulovat pokusy a napsat nulu
            pocet_mereni = 0;
            lcd_gotoxy(8,1);
            lcd_puts("0");
        }
    }
}

```