Vyšší odborná škola a Střední průmyslová škola elektrotechnická
Božetěchova 3, Olomouc
Laboratoře elektrotechnických měření

MIT-Projekt zimní

Název úlohy	Číslo úlohy
Měření vzdálenosti	1-4R

Zadání:
Zjistěte jestli trasa dokáže přenést 10Gbit/s.

Poř. č.	Příjmení a jméno	Třída	Skupina	Školní rok
21	ŠLEHOFER Dominik	4.B	2	2021/22

Datum měření	Datum odevzdání	Počet listů	Klasifikace			
			příprava	měření	protokol	obhajoba
2022	2022	4				

Protokol o měření obsahuje:

- teoretický úvod
- schéma
- použité přístroje
- postup měření
- tabulky
- příklad výpočtu
- grafy
- závěr

Vývojový diagram:

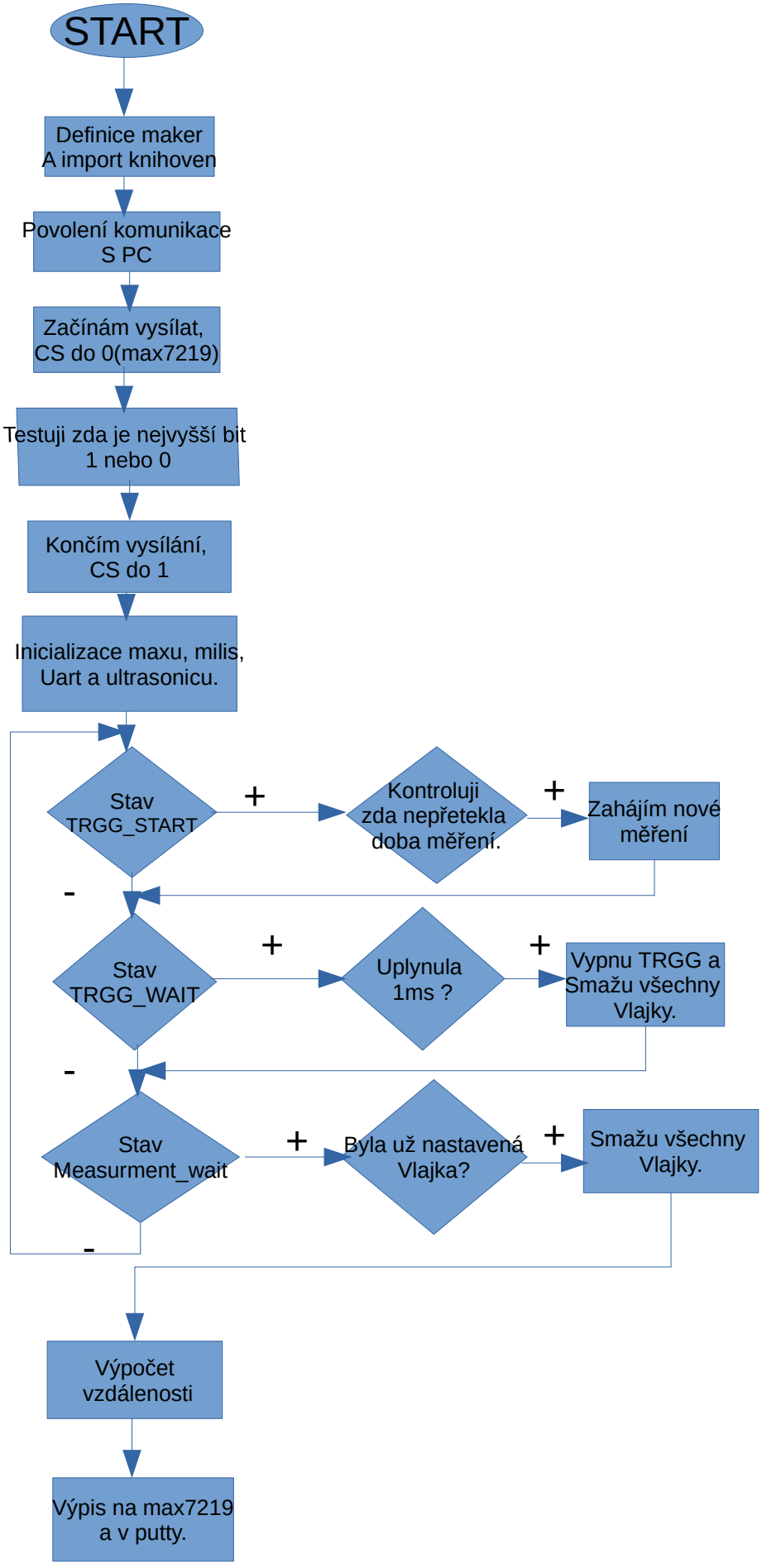
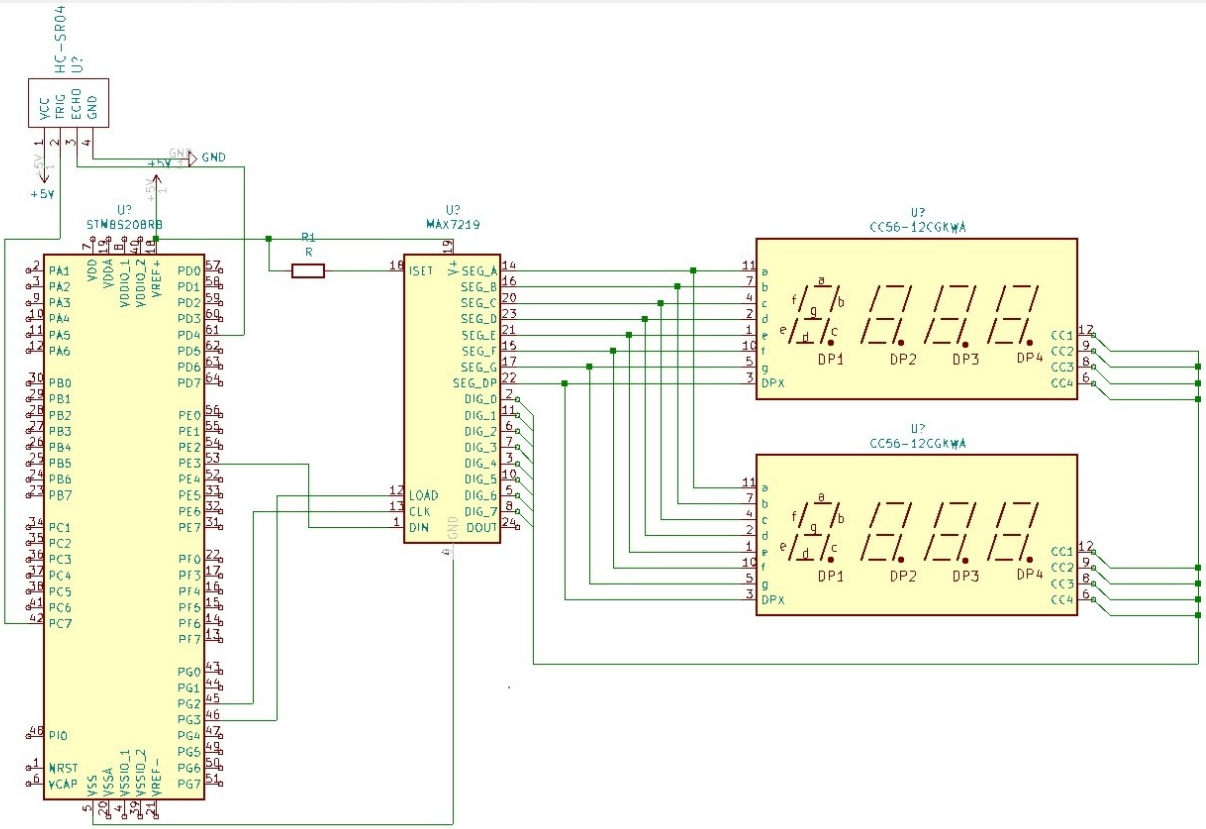
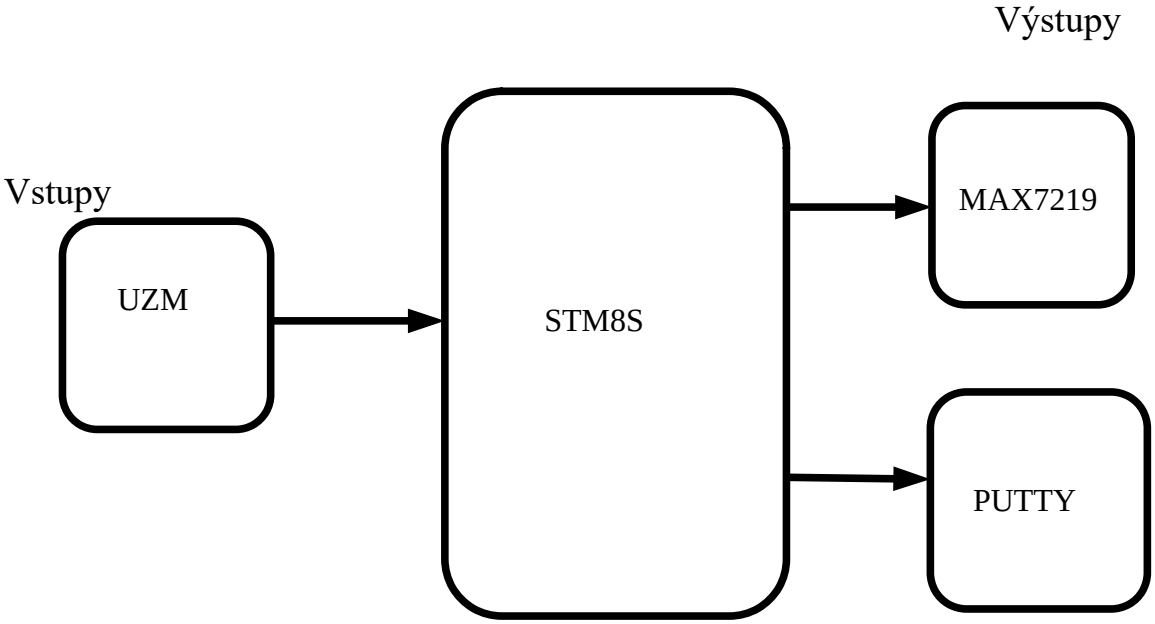


Schéma:



Blokové schéma



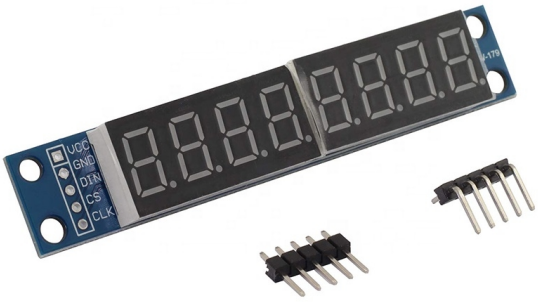
Jméno: Dominik ŠLEHOFER	Třída: 4.B	Číslo protokolu: 1-4R	List: 3/4

Slovní popis

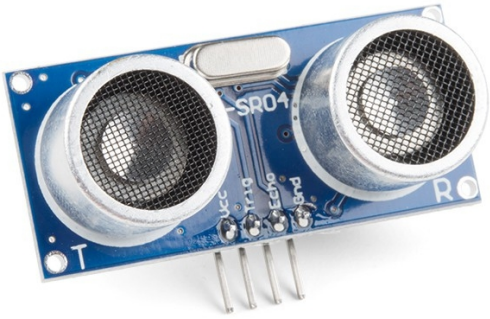
Na začátku si importujeme knihovny a definujeme makra. Následně si vytvoříme funkce pro komunikaci UART, abychom mohli zobrazovat vzdálenost v PUTTY na PC.
Dále máme funkci max7219, pro komunikaci s touto součástkou. V této funkci zjišťujeme zda je nejvyšší bit 1 nebo 0.
Ve funkci init se nastaví jas displeje maxu7219, taktujeme MCU na 16MHz, inicializuje se milis a uart a ultrasonic. Ve funkci nuly se pří nahrání programu do stm8 na každém digitu max7219 vypíšou nuly.
Následně si definujeme datový typ enum pro stavy snímače vzdálenosti a v něm si nadefinujeme stavy.
Dále následuje funkce main, ve které si definujeme proměnné pro vzdálenost a čas měření. Dále proměnnou, která mi bude říkat v jakém stavu se nacházím. Inicializuje se funkce init( ) a nuly( ). Dále je cyklus while(1), který je nekonečný. Ve while (1) se zjistí aktuální stav snímače. V TRGG_START kontrolujeme, zda přetekla doba měření. Pokud ano tak zahájím nové měření. Jakmile se nastavil impuls, tak přepínám do stavu TRGG_WAIT. Pokud jsem v tomto čekacím stavu a už uplynula 1ms, tak se vypne a udělá se sestupná hrana a smažou všechny vlajky. Ve stavu MEASURMENT_WAIT detekuji sestupnou hranou Echo. Ptám se jestli byla už nastavená vlajka. Jestli ano tak smažu vlajky CC1 a CC2. Vypočítá se délka impulzu a převede se na cm.
Výsledek v cm se pak zobrazí na displeji max7219 a v PUTTY.

Popis zapojení a použité součástky:

SPI , max7219



Ultrazvukový měřič vzdálenosti.



Kit STM8S máme propojený pomocí kabelu USB. K STM8 máme připojený ultrasonic. Na 5V a zem. Trigger na na na PC7 a echo na PD4. Není jedno kam ho zapojím. Max7219 je připojený také k 5V a GND. DIN je na PE3, CS na PG3 a CLK na PG2.

Jméno: Dominik ŠLEHOFER	Třída: 4.B	Číslo protokolu: 1-4R	List: 4/4

Závěr:

Program funguje a dělal jsem ho v STVD. Původně jsem chtěl ještě přidat generování zvuku, nicméně to se mi nepovedlo rozchodit.
Osobně si myslím, že teorie MIT není zase tak složitá, ale programování STM8 je velmi obtížné a těžko pochopitelné. Nepomáhá tomu ani to, že jediná vývojová prostředí o kterých vím pro stm8 (codium a stvd) mají pořád nějaké problémy a většinou nefungují (správně).

Program

```
#include "stm8s.h"
#include "assert.h"
#include "delay.h"
#include "milis.h"
#include "stdio.h"
#include "spse_stm8.h"

//bzučák
#define BZ_PORT GPIOB
#define BZ_PIN GPIO_PIN_0

//Ultrasonic
#define TI1_PORT GPIOD//nelze připojit kamkoliv!!!
#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 MEASURMENT_PERON 444    // perioda měření (ms). Doba po kterou čekám a pokud se nic nestane tak je měření nedoměřené

//Max7219
#define CLK_PORT GPIOG
#define CLK_PIN GPIO_PIN_2

#define CS_PORT GPIOG
```

```
#define CS_PIN GPIO_PIN_3

#define DIN_PORT GPIOE
#define DIN_PIN GPIO_PIN_3



#define SET(BAGR) GPIO_WriteHigh(BAGR##_PORT, BAGR##_PIN)
#define CLR(BAGR) GPIO_WriteLow(BAGR##_PORT, BAGR##_PIN)



#define NOOP             0
#define DIGIT0          1
#define DIGIT1          2
#define DIGIT2          3
#define DIGIT3          4
#define DIGIT4          5
#define DIGIT5          6
#define DIGIT6          7
#define DIGIT7          8
#define DECODE_MODE 9   // Aktivace/Deaktivace znakové sady (my volíme v?dy hodnotu DECODE_ALL)
#define INTENSITY       10      // Nastavení jasu - argument je číslo 0 až 15 (vetší číslo vetší jas)
#define SCAN_LIMIT      11      // Volba poctu cifer (velikosti displeje)
#define SHUTDOWN        12      // Aktivace/Deaktivace displeje (ON / OFF)
#define DISPLAY_TEST    15      // Aktivace/Deaktivace "testu" (rozsvítí všechny segmenty)

// makra argumentu
// argumenty pro SHUTDOWN
#define DISPLAY_ON              1       // zapne displej
#define DISPLAY_OFF             0       // vypne displej
// argumenty pro DISPLAY_TEST
#define DISPLAY_TEST_ON         1       // zapne test displeje
#define DISPLAY_TEST_OFF        0       // vypne test displeje
// argumenty pro DECODE_MOD
#define DECODE_ALL              0b11111111 // zapína znakovou sadu pro vsechny cifry
#define DECODE_NONE             0 // vypíná znakovou sadu pro vsechny cifry




//UART komunikace
char putchar (char c)
{
```

```
  /* Write a character to the UART1 */
  UART1_SendData8(c);
  /* Loop until the end of transmission */
  while (UART1_GetFlagStatus(UART1_FLAG_TXE) == RESET);

  return (c);
}

char getchar (void) //čte vstup z UART
{

  int c = 0;
  /* Loop until the Read data register flag is SET */
  while (UART1_GetFlagStatus(UART1_FLAG_RXNE) == RESET);
         c = UART1_ReceiveData8();
  return (c);
}



//Povoleni UART1 (Vyuzivane na komunikaci s PC)
void init_uart(void)
{
    UART1_DeInit();         // smazat starou konfiguraci
                  UART1_Init((uint32_t)115200, //Nova konfigurace

UART1_WORDLENGTH_8D,

UART1_STOPBITS_1,

UART1_PARITY_NO,

UART1_SYNCMODE_CLOCK_DISABLE,

UART1_MODE_TXRX_ENABLE);
    //UART1_Cmd(ENABLE);  // povolí UART1

    //UART1_ITConfig(UART1_IT_RXNE_OR, ENABLE);   // povolí přerušení UART1 Rx

}
```

```
void max7219(uint8_t address, uint8_t data)
{
uint16_t mask;
CLR(CS);//začínám vysílat, Chip Select(CS) dám do 0(CLEAR)

//testuji jestli nejvyšší bit je 1 nebo 0


//adresa
mask = 1<<7;//posunutí 1 o 7 bitu do leva

while(mask){
 CLR(CLK);
 if(address & mask){//logický součin adresy a masky

        SET(DIN);//když je nejvyšší bit 1 zavolám SET
                } else {
                        CLR(DIN);//když je nejvyšší bit 0 zavolám CLR
           }
                SET(CLK);//tikání nahoru
                mask >>=1;//posunutí masky o 1 bit do prava

                CLR(CLK);//tikání dolů
          }


//data
mask = 1<<7;
while(mask){
 CLR(CLK);
 if(data & mask){//logický součin dat a masky
        SET(DIN);
                } else {
                        CLR(DIN);
           }
                SET(CLK);
                mask >>=1;
                CLR(CLK);
          }

          SET(CS);//končí vysílání chip select dám do 1
```

```
}



void init(void){
        //Inicializace maxu7219
  GPIO_Init(CLK_PORT, CLK_PIN, GPIO_MODE_OUT_PP_LOW_SLOW);
        GPIO_Init(CS_PORT, CS_PIN, GPIO_MODE_OUT_PP_LOW_SLOW);
        GPIO_Init(DIN_PORT, DIN_PIN, GPIO_MODE_OUT_PP_LOW_SLOW);
        GPIO_Init (BZ_PORT, BZ_PIN, GPIO_MODE_OUT_PP_LOW_SLOW);//buzzer
        max7219(DECODE_MODE, DECODE_ALL); // zapnout znakovou sadu na vsech cifrach
  max7219(SCAN_LIMIT, 7); // velikost displeje 8 cifer (pocítano od nuly, proto je argument cislo 7)
  max7219(INTENSITY, 3);
  max7219(DISPLAY_TEST, DISPLAY_TEST_OFF); // Funkci "test" nechceme m?t zapnutou
  max7219(SHUTDOWN, DISPLAY_ON); // zapneme displej

        CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1); // taktovat MCU na 16MHz



        init_milis();
        init_uart(); //Povoleni komunikace s PC




        //Inicializace ultrasonicu
    //Trigger setup
    GPIO_Init(TRGG_PORT, TRGG_PIN, GPIO_MODE_OUT_PP_LOW_SLOW); //režim push pull


                //TIM2 setup
    GPIO_Init(TI1_PORT, TI1_PIN, GPIO_MODE_IN_FL_NO_IT);  // kanál 1 jako vstupní kanál
                // potřebuji nastavit předděličku a strop časovače


    TIM2_TimeBaseInit(TIM2_PRESCALER_16, 0xFFFF );//16 bitový, nechci ho omezovat >strop nastavím na
ffff, 0x>hex soustava ffff=65535
                //Prescaler dělí frekvenci z master. Mám 16MHz a dělím 16 takže mám 1 MHz.Perioda
1 mikro sek.
                //čítač je 16 bitový, každou mikro sekundu přijde impuls.Takže napočíta max 65536
mikro sekund.



    /*TIM2_ITConfig(TIM2_IT_UPDATE, ENABLE);*/
```

```
    TIM2_Cmd(ENABLE);
    TIM2_ICInit(TIM2_CHANNEL_1,        // nastavuji CH1 (CaptureRegistr1)
              TIM2_ICPOLARITY_RISING,    // zachycení se spouští pomocí naběžné hrany
              TIM2_ICSELECTION_DIRECTTI, // CaptureRegistr1 bude ovládán přímo (DIRECT)z CH1
              TIM2_ICPSC_DIV1,           // delicka je vypnuta
              0                          // vstupní filter je vypnuty
         );
    TIM2_ICInit(TIM2_CHANNEL_2,        // nastavuji CH2 (CaptureRegistr2)
              TIM2_ICPOLARITY_FALLING,   //zachytávat se  sestupnou hranou
              TIM2_ICSELECTION_INDIRECTTI, // CaptureRegistr2 bude ovládán  (nepřímo) z CH1
              TIM2_ICPSC_DIV1,           // delicka je vypnuta
              0                          // vstupni filter je vypnuty
         );
}


//Na maxu budou všude nuly
void nuly(void)
{
          max7219(1,0);
          max7219(2,0);
          max7219(3,0);
          max7219(4,0);
          max7219(5,0);
          max7219(6,0);
          max7219(7,0);
          max7219(8,0);
}




typedef enum //datový typ Enum pro stavy snimace vzdalenosti
{//definuju si stavy
    TRGG_START,       // zahájení trigger impulzu
    TRGG_WAIT,        // cekání na konec trrigger impulzu
    MEASURMENT_WAIT   // čekání na dokončení meření
} STATE_TypeDef;


void main(void)
{
```

```
    uint32_t mtime_ultrasonic = 0;
    uint32_t delka;
    STATE_TypeDef state = TRGG_START;//říká mi v jakém stavu se nachází. Nejdřiv udělám trigger impuls a budu čekat než uplyne doba trrg impulsu.



    init();
                nuly();







    while (1) {//nekonečná smyčka  co běha dokola
        switch (state) { //Stav snimace
        case TRGG_START:
            if (milis() - mtime_ultrasonic > MEASURMENT_PERON) {// kontroluji zda nepřetekla doba měření
                mtime_ultrasonic = milis();// pokud ano zahájí nové měření
                TRGG_ON;//zahájím měření
                state = TRGG_WAIT;//jakmile zjistím, že jsem nastavil impuls, tak přepínám do stavu trigger wait.
            }
            break;

        case TRGG_WAIT:
            if (milis() - mtime_ultrasonic > 1) {//pokud jsem ve stavu trigger wait a už uplynula 1ms
                TRGG_OFF;//Tak to vypnu a udělám sestupnou hranu.
                // smažu všechny vlajky
                TIM2_ClearFlag(TIM2_FLAG_CC1);
                TIM2_ClearFlag(TIM2_FLAG_CC2);
                TIM2_ClearFlag(TIM2_FLAG_CC1OF);
                TIM2_ClearFlag(TIM2_FLAG_CC2OF);
                state = MEASURMENT_WAIT;
            }
            break;

        case MEASURMENT_WAIT:
             /* detekuji sestupnou hranu ECHO signálu; vzestupnou hranu
              * detekovat nemusím, zachyceni CC1 i CC2 probehne automaticky  */
            if (TIM2_GetFlagStatus(TIM2_FLAG_CC2) == RESET) {//byla už nastavená vlajka ???
                TIM2_ClearFlag(TIM2_FLAG_CC1);  // smažu vlajku CC1
                TIM2_ClearFlag(TIM2_FLAG_CC2);  // smažu vlajku CC2
```

```
                    // délka impulzu
                    delka = (TIM2_GetCapture2() - TIM2_GetCapture1());

                                                                        //Vypocet a vypsani
vzdalenosti na PC a displej
                    delka = (delka * 340)/ 20000; // FixPoint prepocet na cm
                    printf("Vzdalenost: %lu cm\r\n", delka);


                                            max7219(1,delka-(delka/10*10));//jednotky
                                            max7219(2,delka/10); //Desítky
                                            max7219(3,delka/100); //Stovky

                                if (delka > 10){
GPIO_WriteHigh(BZ_PORT,BZ_PIN);
                                                                        }




                    state = TRGG_START;
                }
                break;
            default:
                state = TRGG_START;
            }

        }


}

void assert_failed(uint8_t* file, uint32_t line)// stvd házelo nějakou chybu s assert a tohle pomohlo


{

  /* User can add his own implementation to report the file name and line number,
```

```
    ex: printf(''Wrong parameters value: file %s on line %d\r\n'', file, line) */


  /* Infinite loop */

  while (1)

  {

  }

}
```