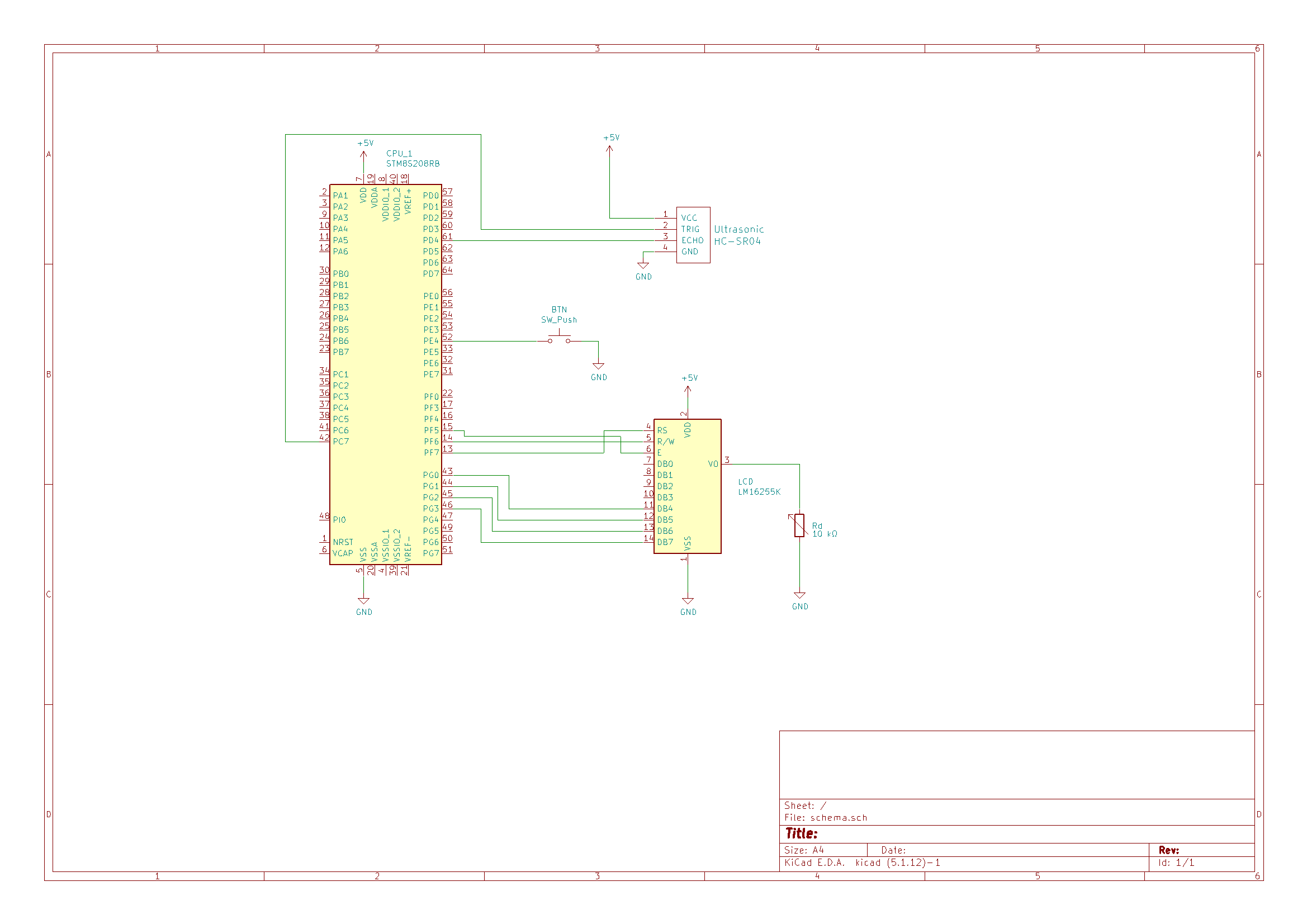
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Vyšší odborná škola a Střední průmyslová škola elektrotechnickáBožetěchova 3, OlomoucMikroprocesorová technika | | | | | | | | | | | | | | | | | |
| **ZÁVĚREČNÝ PROJEKT MIT** | | | | | | | | | | | | | | | | | |
| **Obsah**  [**Schéma ZAPOJENÍ** 2](#_Toc93237847)  [**SLOVNÍ POPIS ZAPOJENÍ** 2](#_Toc93237848)  [**BLOKOVÉ SCHÉMA ZAPOJENÍ** 2](#_Toc93237849)  [**SLOVNÍ POPIS PŘÍPRAVKŮ** 3](#_Toc93237851)  [**SLOVNÍ POPIS FUNKCE PROGRAMU** 3](#_Toc93237852)  [**VÝVOJOVÝ DIAGRAM PROGRAMU** 4](#_Toc93237853)  [**ZÁVĚR** 5](#_Toc93237855)  [**UKÁZKA PROGRAMU** 6](#_Toc93237857) | | | | | | | | | | | | | | | | | |
| Poř. č. | Příjmení a jméno | | |  | | | | | | Třída | | Skupina | | Školní rok | |  | |
| 14 | PŘIKRYL Jan | | | | | | | | | 4B | | 1 | | 2021/22 | | | |
| Datum zpracování | |  | Datum odevzdání | |  | Počet listů | |  | Klasifikace | | | | | | | |
| 14. 1. 2022 | | | 17. 1. 2022 | | | 7 | | | Textová část | | Obhajoba | | | | Funkčnost | |
| Protokol o měření obsahuje: | | | | | | | teoretický úvod | | | | | | tabulky | | | | |
|  | | | | | | | schéma | | | | | | příklad výpočtu | | | | |
|  | | | | | | | použité přístroje | | | | | | grafy | | | | |
|  | | | | | | | postup měření | | | | | | závěr | | | | |

**Schéma 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 SCHÉMA ZAPOJENÍ**

Displej

Tlačítko

STM 8

Ultrasonic

(ultrazvukový snímač)

UART 1

(komunikace s PC)

**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 10µs. Šířka impulzu 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 µs* \* 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, kopírkách 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, milis.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 milis, 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 něž 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 impulzu 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).

**VÝVOJOVÝ DIAGRAM PROGRAMU**

START

Importuj knihovny pro práci s displejem, časovači a poly.

Vytvoř makra pro piny a porty snímače a displeje a pro tlačítko.

Vytvoř funkce pro komunikaci s PC, pro nastavení a povolení všech komponent.

Je aktuální stav

TRGG\_START

pocet\_mereni = 0

mtime\_ultrasonic = 0

diff = 0

STATE\_TypeDef state = RGG\_START

text[16], mereni\_text[16]

Pravda

KONEC

-

+

Vypiš vzdálenost na

displeji a PC

-

+

Spustit náběžnou hranu TRIG a stav = TRGG\_WAIT

Konec Trig signálu, smaž všechny vlajky a stav = MEASURMENT\_WAIT

Vynuluj měření a vypiš 0

Vypiš počet pokusů zvětšený o 1

Stav = TRGG\_START

Definuj enum pro aktuální stav

Nastav všechno potřebné, inicializuj displej a vypiš uvítací hlášku na displej a do PC

-

Je aktuální stav

TRGG\_WAIT

Je

aktuální stav

MEASURMENT\_WAIT

Je

Stisknuto tlačítko

-

+

-

+

-

+

milis() - mtime

\_ultrasonic >

MASURMENT

\_PERON

milis() - mtime\_ultrasonic

> 1

-

+

-

+

Sestupná hrana ECHO signálu

-

+

Vzdálenost >=400 cm

**ZÁVĚR**

Shrnutí a zhodnocení

* Vytvořil jsem program na měření vzdálenosti s STM8. Vzdálenost je měřena snímačem Ultrasonic HC-SR04. Výsledek je zobrazen na lcd disleji 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.

**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 GPIOD

#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 merení (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(prijímá data) vstup z UART*

{

int c = 0;

while (UART1\_GetFlagStatus(UART1\_FLAG\_RXNE) == RESET);

c = UART1\_ReceiveData8();

return (c);

}

void init\_uart1(void) *//Povoleni UART1 (Vyuzivane 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(); *//Rozbehnuti casovace milis*

init\_uart1(); *//Povoleni komunikace s PC*

GPIO\_Init(BTN\_PORT, BTN\_PIN,GPIO\_MODE\_IN\_FL\_NO\_IT); *// Tlačítko jako vstup (vynulovani)*

*//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 (CaptureRegistr1)*

TIM2\_ICPOLARITY\_RISING, *// nábežná hrana*

TIM2\_ICSELECTION\_DIRECTTI, *// CaptureRegistr1 bude ovládán z CH1*

TIM2\_ICPSC\_DIV1, *// delicka je vypnutá*

0 *// vstupní filter je vypnutý*

);

TIM2\_ICInit(TIM2\_CHANNEL\_2, *// nastavuji CH2 (CaptureRegistr2)*

TIM2\_ICPOLARITY\_FALLING, *// sestupná hrana*

TIM2\_ICSELECTION\_INDIRECTTI, *// CaptureRegistr2 bude ovládán z CH1*

TIM2\_ICPSC\_DIV1, *// delicka je vypnutá*

0 *// vstupní filter je vypnutý*

);

}

typedef **enum** *//Enum pro stavy snimace vzdalenosti*

{

TRGG\_START, *// zahájení trigger impoulzu*

TRGG\_WAIT, *// cekání na konec trrigger impoulzu*

MEASURMENT\_WAIT *// ckání na dokoncení merení*

} 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 (milis() - mtime\_ultrasonic > MASURMENT\_PERON) {

mtime\_ultrasonic = milis();

TRGG\_ON;

state = TRGG\_WAIT;

}

**break**;

case TRGG\_WAIT:

if (milis() - 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 = MEASURMENT\_WAIT;

}

**break**;

case MEASURMENT\_WAIT:

*// detekuji 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 vypsani 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 talčítko - vynulovat pokusy a napsat nulu*

pocet\_mereni = 0;

lcd\_gotoxy(8,1);

lcd\_puts("0 ");

}

}

}