

ZÁVĚREČNÝ PROJEKT MIT

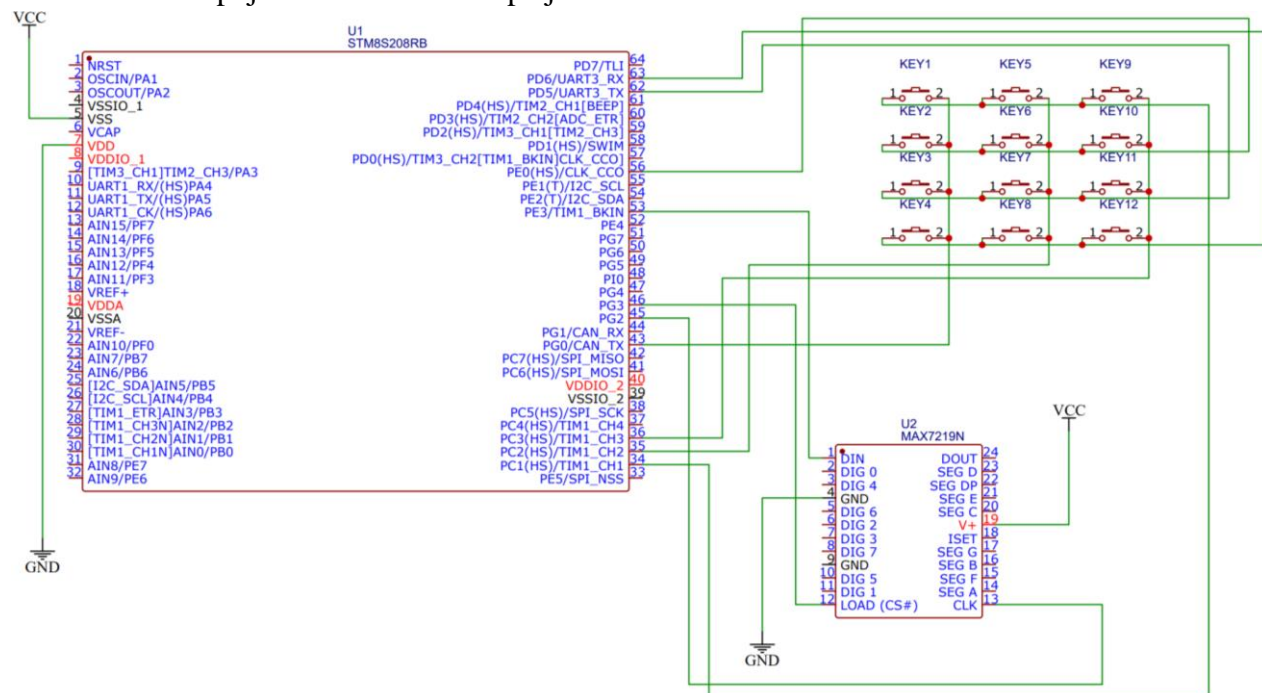
Obsah

SCHÉMA ZAPOJENÍ	2
SLOVNÍ POPIS ZAPOJENÍ	2
BLOKOVÉ SCHÉMA ZAPOJENÍ	2
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 9. 4. 2022		Datum odevzdání 12. 4. 2022	Počet listů 8	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 klávesnice a displeje na STM8



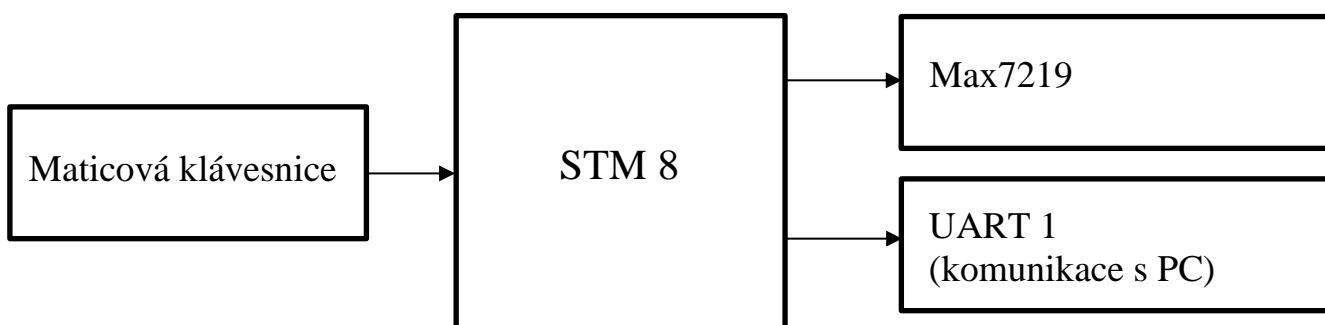
SLOVNÍ POPIS ZAPOJENÍ

Maticová klávesnice je připojena přes tyto piny k STM8: PG0, PC2, PC3 – jako sloupce tlačítek. Řádky jsou připojeny přes tyto piny: PC1, PE0, PD5, PD6.

Jako výstup je mimo komunikaci s PC (UART) zapojen sedmi segmentový displej max7219. Ten je připojen přes piny PG2, PG3 a PE3.

Displej je napájen pěti volty (pin VCC), uzemněn GND.

BLOKOVÉ SCHEMA ZAPOJENÍ



SLOVNÍ POPIS FUNKCE PROGRAMU

Nejprve se importují všechny potřebné knihovny (to jsou `milis.h`, `stdio.h`, `stm8s.h`, `assert.h`, `keypad.h`, `delay.h`). Následuje založení maker pro piny, porty a stavy displeje a všechny potřebné funkce (makra a funkce pro fungování klávesnice jsou v knihovně `keypad.h`).

Poté se vytvoří funkce pro UART komunikaci, inicializaci (taktování procesoru na 16 MHz, rozběhnutí časovače milis, povolení komunikace s počítačem a nastavení displeje).

Ve funkci `main` (hlavní funkce každého programu v jazyce C) jsou založeny proměnné pro minulé a aktuální zmáčkuté tlačítko a pozici.

Poté se zavolá inicializační funkce a vypíše se uvítací hláška.

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

Opakovat se bude následující: Každých 55 ms se uloží aktuální čas milis.

Zkontroluje se stisknuté tlačítko. Nerovná se minulému stisknutému tlačítku, bylo tedy stisknuto, vypíše se do PC stisknutá klávesa.

Byla-li stisknuta klávesa `*` (netisknutelný znak), na displej se vypíše prázdné místo.

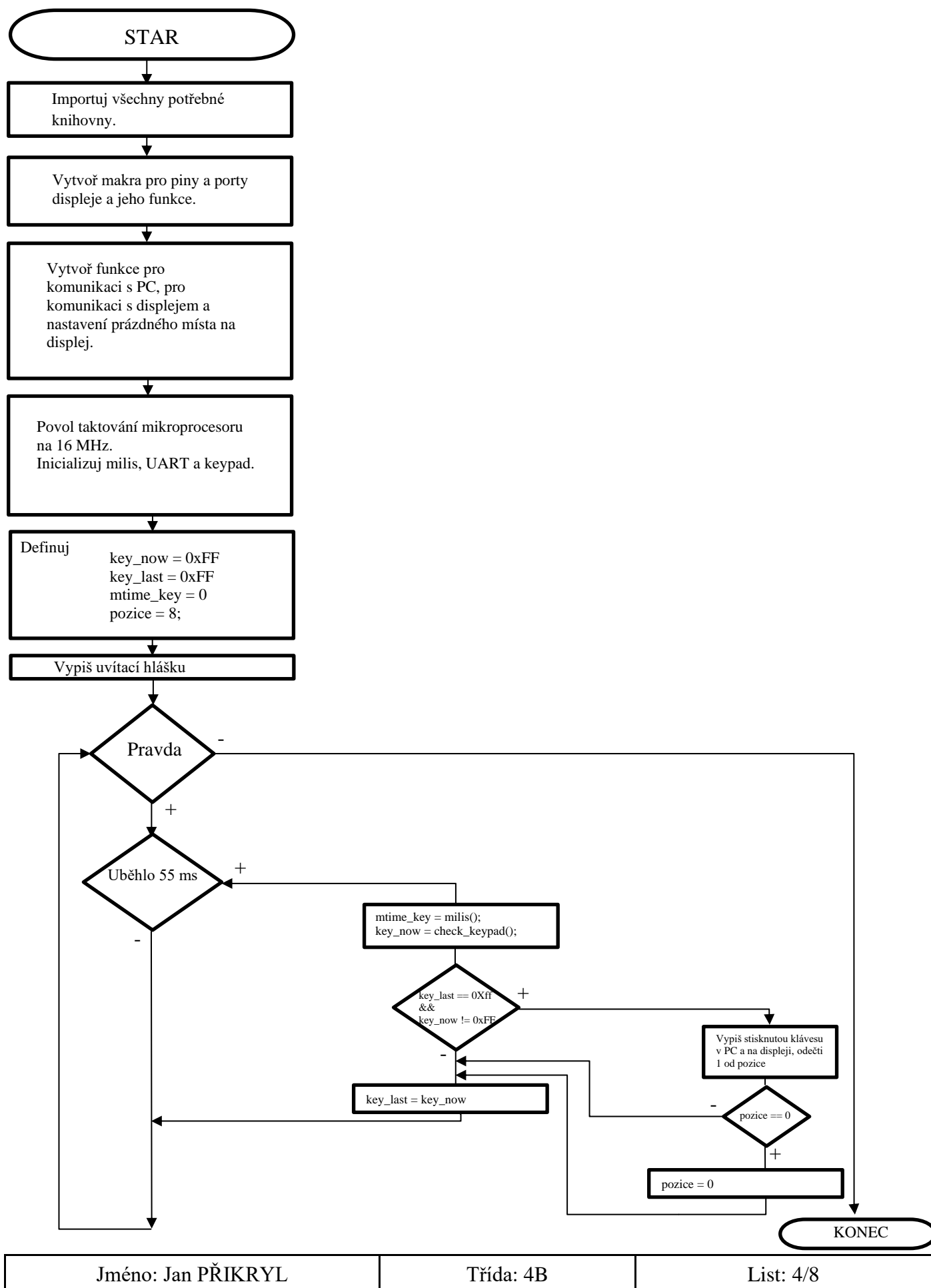
Byla-li stisknuta klávesa `#` (opět netisknutelný znak), všechny pozice na displeji se nastaví na prázdné místo a pozice se nastaví na 8 (první místo displeje).

Byla-li stisknuta jakákoliv jiná klávesa, vypíše se na pozici na místě pozice.

Poté se do proměnné pozice přičte 1. Je-li toto číslo 0, nastaví se opět na 8.

Nakonec se minule stisknutá klávesa bude rovnat aktuálně stisknuté klávese.

VÝVOJOVÝ DIAGRAM PROGRAMU



ZÁVĚR

Shrnutí a zhodnocení

- Vytvořil jsem program na zaznamenávání čísel (poznámkový blok) s STM8. Čísla jsou zadávána na maticové klávesnici. Čísla jsou zobrazována na displeji max7219 a pro přehlednost 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 se dá celý vymazat.

Nevýhody

- Nelze vkládat a zobrazovat písmena.

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

- Naučil jsem se používat maticovou klávesnici.

UKÁZKA PROGRAMU

```
#include "stm8s.h"
#include <stdio.h>
#include "assert.h"
#include "milis.h"
#include "keypad.h"
#include "delay.h"

//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
#define INTENSITY          10
#define SCAN_LIMIT         11
#define SHUTDOWN            12
#define DISPLAY_TEST       15

// 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
#define DECODE_NONE        0

//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) //funkce čte vstup z UART
{
    int c = 0;
    while (UART1_GetFlagStatus(UART1_FLAG_RXNE) == RESET);
        c = UART1_ReceiveData8();
    return (c);
}
//Povolení UART1
void init_uart1(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);
}
void max7219(uint8_t address, uint8_t data)
{
    uint16_t mask;
    CLR(CS);
    mask = 1<<7;
    while(mask){
        CLR(CLK);
        if(address & mask){
            SET(DIN);
        } else {
            CLR(DIN);
        }
        SET(CLK);
        mask >>=1;
        CLR(CLK);
    }
    mask = 1<<7;
    while(mask){
        CLR(CLK);
        if(data & mask){
            SET(DIN);
        } else {
            CLR(DIN);
        }
        SET(CLK);
        mask >>=1;
        CLR(CLK);
    }
    SET(CS);
}
void nic(void) { //Vypsání prázdných míst
    max7219(1, 15);
    max7219(2, 15);
    max7219(3, 15);
    max7219(4, 15);
    max7219(5, 15);
    max7219(6, 15);
    max7219(7, 15);
    max7219(8, 15);
}

```

```

void setup(void)
{
    CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1); // taktovat MCU na 16MHz
    init_milis(); //Inicializace milis
    init_uart1(); //Povoleni komunikace s PC
    init_keypad(); //Inicializace klávesnice

    //max7219
    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);
    max7219(DECODE_MODE, DECODE_ALL);
    max7219(SCAN_LIMIT, 7);
    max7219(INTENSITY, 3);
    max7219(DISPLAY_TEST, DISPLAY_TEST_OFF);
    max7219(SHUTDOWN, DISPLAY_ON); // zapneme displej
    nic();
}

int main(void)
{
    uint32_t mtime_led = 0;
    uint8_t key_now = 0xFF;
    uint8_t key_last = 0xFF;
    uint32_t mtime_key = 0;
    uint8_t pozice = 8;
    setup();
    printf("Dobry den, \n\r");
    printf("vitejte v programu poznamkovy blok. \n\r");
    printf("Tlacitky klavesnice muzete vypisovat postupne na jednotlivé  
segmenty cisla. \n\r");
    printf("Tlacitkem * napisete prazdne misto, tlacitkem # napisete vsude  
prazdna mista a kurzor posunete na zacatek. \r\n");
    while (1) {
        if (milis() - mtime_key > 55) { // detekce stisknuté klávesy
            mtime_key = milis();
            key_now = check_keypad();
            if (key_last == 0xFF && key_now != 0xFF) {
                char x[2];
                sprintf(x, "%x", key_now);
                printf("Klavesa: %c \n\r", x[0]);
                if (x[0] == 'a') {
                    max7219(pozice, 15);
                } else if (x[0] == 'b') {
                    nic();
                    pozice = 9;
                } else {
                    max7219(pozice, x[0]);
                }
                pozice = pozice - 1;
                if (pozice == 0) {
                    pozice = 8;
                }
            }
            key_last = key_now;
        }
    }
}

```