



Díleňská praxe

A4	04 Maticový displej		
Kubanek Tobias		1/6	Známka:
21.12.2023	Datum odevzdání:	4.1.2024	Odevzdáno:



Zadání:

Zpracujte program v programovacím jazyce C ovládající určený připojený maticový displej a klávesnici tak, aby obsahoval nejméně tyto funkce:

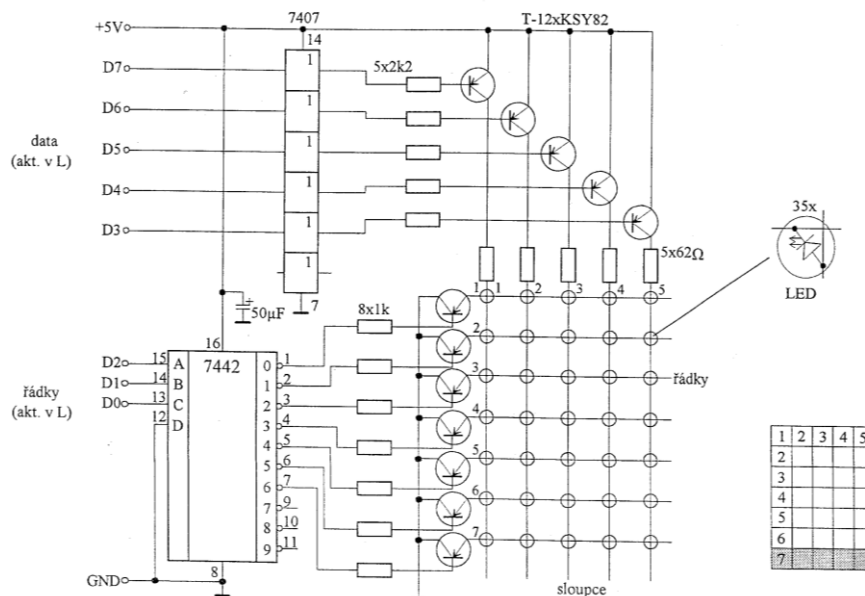
- 1) stisknuté tlačítko klávesnice se uloží do paměti modulu MB-ATmega128, minimálně 5 kláves, maximálně 15 kláves
- 2) každé klávese bude přiřazen vhodný zobrazovaný symbol
- 3) rozpoznání stavu vkládání znaků a stavu přehrávání vložených znaků ovládaných pomocí maticové klávesnice klávesnice
- 4) přepínání mezi těmito režimy
- 5) využití všech vhodných HW možností přípravku MB-ATmega128

Schema zapojení (situační schema):

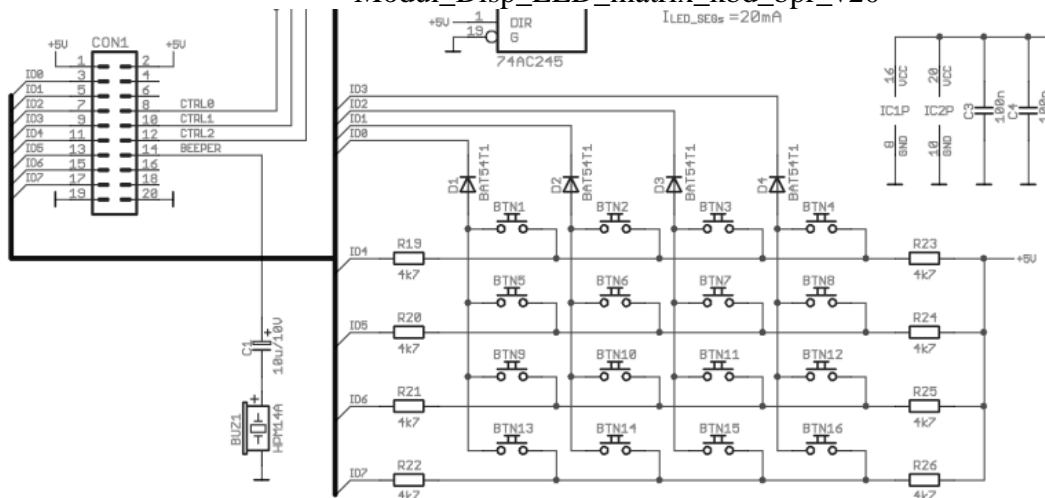
LABORATOŘE MIKROPROCESOROVÉ TECHNIKY

název úlohy: Maticový displej

Maticový displej 5x7 bodů s Hi-bright LED:



Modul_Displ_LED_matrix_kbd_bpr_v20





Postup řešení:

1. Vytvoříme program v jazyce C pro ovládání maticového displeje 5x7 LED displej pomocí maticové klávesnice (Modul_Displ_LED_matrix_kbd_bpr_v20).
2. Využijeme základní nastavení, nastavení přerušení, vytvoření tabulky hodnot v programu.
3. Vytvoříme znakové generátory pro 5x7 LED displej.
4. Vytvoříme program pro zapisování a rozpoznávání znaků z maticové klávesnice a odesílání na daný maticový displej 5x7 LED.
5. Poté program zapneme a plně otestujeme, zda všechno funguje tak jak má.

Komentář k programu:

Výpis programu:

Viz. příloha.

Přílohy:

Vývojový diagram. (1 strana)

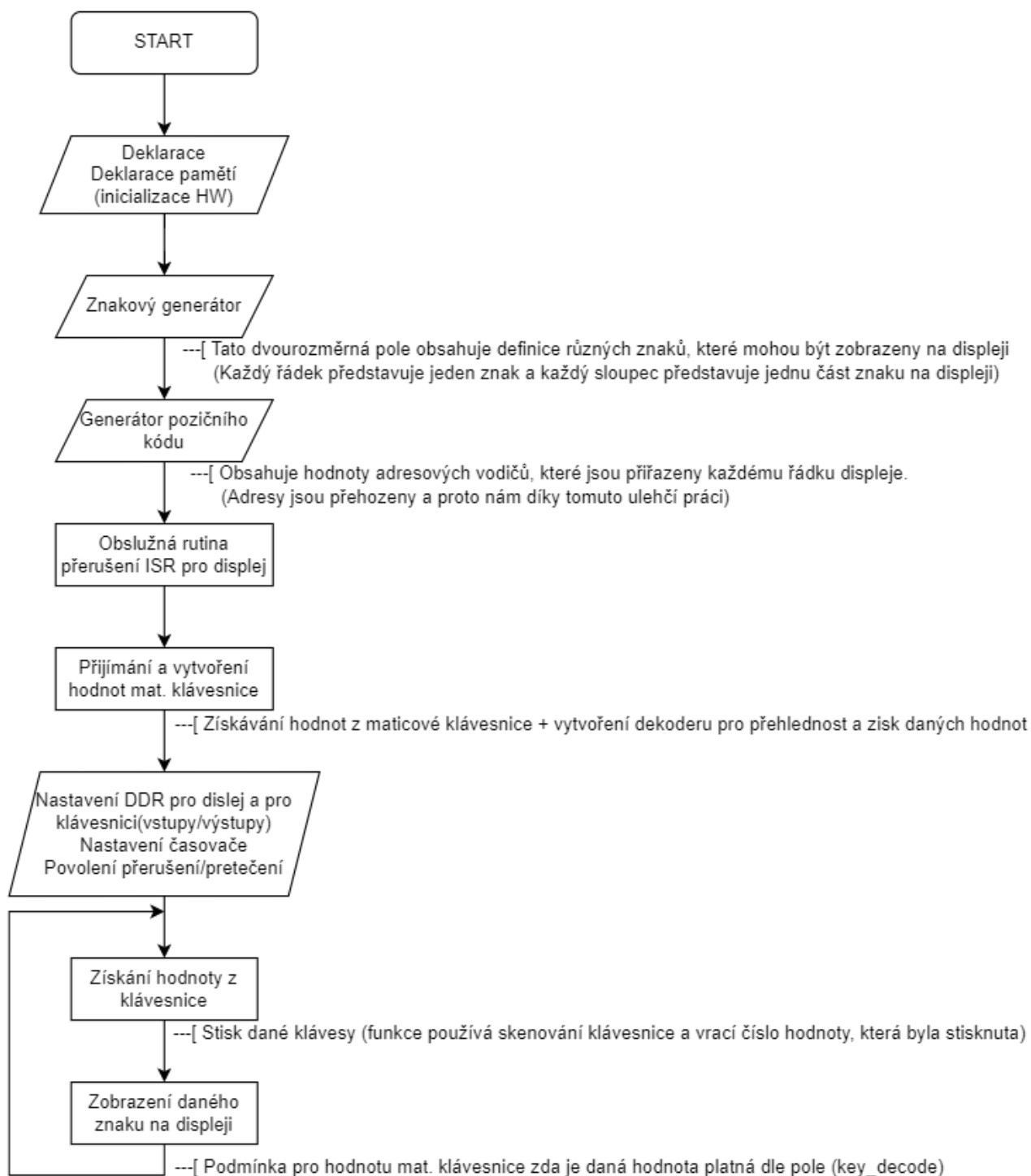
Výpis programu. (2 strany)

Závěr:

Výsledkem mé práce je program na obsluhu maticové klávesnice společně s maticovým displejem. Celý program jsem si nestihl odzkoušet, ale během cvičení jsem si ověřoval funkci jednotlivých kroků zvlášť a zbytek jsem dodělal doma, všechny tyto kroky jsem následně složil do jednoho a měli by plně fungovat



Vývojový diagram:





Výpis programu:

```
#define F_CPU 14745600UL // Krystal PK-Design desky

#include<stdio.h>
#include<avr/io.h>
#include<avr/interrupt.h>
#include<util/delay.h>

#define DDR_DSP DDRE // Smerovy registr displeje
#define PORT_DSP PORTE // Port displeje
#define DDR_KBD DDRB // Smerovy registr klavesnice
#define PORT_KBD PORTB // Port klavesnice vystupni
#define PIN_KBD PINB // Port klavesnice vstupni
#define KBD_INTERCOLUMN_DELAY 500 // Cas na zotaveni diod klavesnice (mikrosekundy)

volatile uint8_t idx=0; // Index radku, který se prave kresli na displeji
volatile uint8_t num=0; // Cislo, které se prave zobrazuje

// Znakovy generator
volatile uint8_t c_gen[12][7]={ 0x0E,0x11,0x11,0x11,0x11,0x11,0x0E, // 0
                                0x04,0x0C,0x14,0x04,0x04,0x04,0x1F, // 1
                                0x0E,0x11,0x01,0x02,0x04,0x08,0x1F, // 2
                                0x0E,0x11,0x01,0x06,0x01,0x11,0x0E, // 3
                                0x02,0x06,0x0A,0x1F,0x02,0x02,0x02, // 4
                                0x1F,0x10,0x1E,0x01,0x01,0x11,0x0E, // 5
                                0x0E,0x11,0x10,0x0E,0x11,0x11,0x0E, // 6
                                0x1F,0x01,0x02,0x04,0x04,0x04,0x04, // 7
                                0x0E,0x11,0x11,0x0E,0x11,0x11,0x0E, // 8
                                0x0E,0x11,0x11,0x0E,0x01,0x11,0x0E, // 9
                                0x15,0x0A,0x15,0x0A,0x15,0x0A,0x15, // Sachovnice
                                0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x1F // Lamp test
                                };

// Adresove vodice jsou prehazene proto je lepsi je adresovat pomoci pole
volatile uint8_t p_gen[7]={0x00,0x04,0x02,0x06,0x01,0x05,0x03}; // Generator pozicniho kodu

ISR (TIMER0_OVF_vect)
{
    // ~ Bitovy negator
    uint8_t out_val= (~(c_gen[num][idx]) << 3) + p_gen[idx]; // obsah displeje + adresa dekodery
    PORT_DSP=out_val; // aktualizuje obsah displeje
    idx++;
    if (idx>6) idx=0; // Citac modulo 7 (0 az 6 a potom reset -> 0)
}

uint8_t get_key (void)
{
    uint8_t col=0;
    uint8_t row=0;
    // Esc F1 F2 * # CR
    uint8_t key_decode[16]={1,2,3,20,4,5,6,21,7,8,9,22,10,0,11,23};
    while (1)
    {
        for (col=0;col<4;col++)
        {
            PORT_KBD = ~(1 << col) & 0x0F;
            _delay_us (KBD_INTERCOLUMN_DELAY);
            for (row=4;row<8;row++)
            {
                if ((PIN_KBD & (1 << row)) == 0) return (key_decode[((row-4) * 4)+col]);
            }
        }
    }
}
```



```
    }  
  }  
}  
  
void hw_init(void)  
{  
    DDR_DSP=0xFF; // Displej  
    DDR_KBD=0x0F; // Dolní polovina vystup (sloupce vodice klavesnice), horní vstupy (radky)  
    TCCR0=0x03;   // Prescaler 3 = F_CPU/8 (tik), /256 (8-bit counter)  
    TIMSK=0x01;   // Prerušeni povoleno od pretečení timeru 0  
    sei();  
}  
  
int main (void)  
{  
    uint8_t key=0;  
    hw_init();  
    while (1)  
    {  
        key=get_key();  
        if (key<12) num=key; // Zobrazujeme jen znaky, které máme ve znakovém generatoru  
                             // Je možné to rozšířit  
    }  
}
```