

9. Rozšiřující desky Evb_Display a Evb_keyboard



Čas ke studiu: 2-3 hodiny



Cíl Po prostudování tohoto odstavce budete něco vědět o

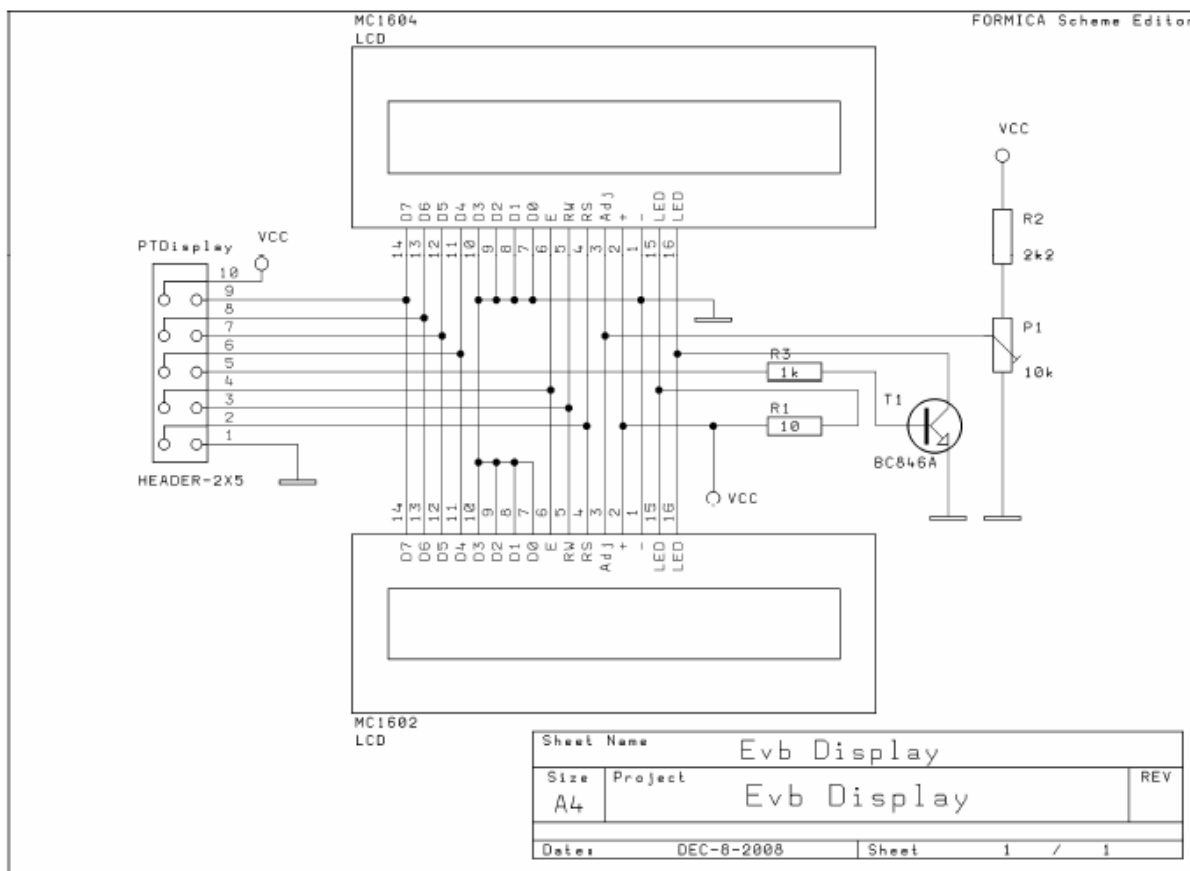
- Zobrazovacích displejích
- Principu činnosti a programování čtyřřádkového displeje
- Maticových klávesnicích
- Principu činnosti a programování maticové



Výklad

9.1 Rozšiřující deska Evb_Display

Rozšiřující deska Evb_Display je určena k připojení se základní deskou Evb_HCS08. Elektrické schéma desky je uvedeno na obrázku 9.1.1. Deska se připojuje pomocí plochého desetizilového kabelu k některému z vstupně/výstupních paralelních portů PTA, PTB, PTC nebo PTD+E. Je osazena čtyřřádkovým LCD displejem, délka každého řádku je 16 znaků.



Obr. 9.1.1: Elektrické schéma desky Evb_Display

□ Komunikace s displejem

Komunikace s displejem je řízena bity RS, RW a E a probíhá pomocí bitů D4 – D7, přes které můžeme odesílat řídicí příkazy nebo znaky. O tom, zda se jedná o příkaz nebo znak rozhoduje hodnota bitu RS.

Tab. 1: Význam jednotlivých bitů pro komunikaci s displejem.

Přiřazení portpinu	Název bitu	Význam
0	RS	Určuje, zda je zadáván znak nebo příkaz
1	RW	Určuje, zda se bude číst nebo zapisovat
2	E	Povoluje přístup k bitům D4 – D7
3	LED	Podsvícení displeje
4	D4	Příjem 0. a 4. bitu znaku
5	D5	Příjem 1. a 5. bitu znaku
6	D6	Příjem 2. a 6. bitu znaku
7	D7	Příjem 3. a 7. bitu znaku

□ Inicializace displeje

Abychom mohli zahájit komunikaci s displejem, je zapotřebí na začátku programu odeslat sérii příkazu, které jsou definovány výrobcem. Inicializace probíhá v následujících krocích:

- Nastavení všech pinů na připojeném portu na základní desce pro zápis
- Nastavení bitu RC na 0, ostatní zbývající bity na 1
- Nastavit bit RS na 1 a zbývající piny na 0
- Odeslat na displej příkazy v pořadí dle tabulky 2

Tab. 2. Inicializační příkazy displeje

pořadí	hodnota bin	hodnota hex
1	0b00000011	0x03
2	0b00000011	0x03
3	0b00000011	0x03
4	0b00000010	0x02
5	0b00101000	0x28
6	0b00001000	0x08
7	0b00000001	0x01
8	0b00000110	0x06
9	0b00101000	0x28
10	0b00001100	0x0C
11	0b00000110	0x06
12	0b00001100	0x0C

□ Odesílání znaků a příkazů na displej

Pro odeslání znaku nebo příkazu na displej je nutné dodržet následující postup:

1. Bit RW nastavit na 0 (zápis)
2. Bit RS nastavit na 0(následuje příkaz) nebo na 1 (následuje znak)
3. Bit E nastavit na 1
4. Zápis horních 4 bitů bajtu
5. Bit E nastavit na 0
6. Bit E nastavit opět na 1
7. Zápis dolních 4 bitů bajtu
8. Bit E nastavit na 0

Na displeji je možno zobrazit znaky ASCII kódu 32-127.

□ Dostupné příkazy

V tabulce 3 jsou uvedeny dostupné příkazy, které je možno při práci s displejem použít:

Tab. 3. Dostupné příkazy pro displej

hodnota	význam
0b00000001	zapne displej
0b00001000	vypne displej
0b00000001	smaže obsah displeje
0b10000000	nastaví kurzor na začátek prvního řádku
0b11000000	nastaví kurzor na začátek druhého řádku
0b10000001	nastaví kurzor na 2.pozici prvního řádku (přičtením požadovaného čísla k příkazu 0b10000000 můžete kurzor nastavit na libovolnou pozici)
0b10010000	nastaví kurzor na začátek třetího řádku (1. řádek + 16 pozic)
0b11010000	nastaví kurzor na začátek čtvrtého řádku (2. řádek + 16 pozic)
0b00001100	kurzor neviditelný
0b00001101	kurzor podtržítka bliká
0b00001110	kurzor plný
0b00001111	kurzor plný bliká

□ Programový pracovní rámec

V hlavním programu si pro usnadnění práce nadefinuju konstanty představující bity displeje, počet řádků a počet sloupců:

```
#define RS      PTBD_PTBD0
#define RW      PTBD_PTBD1
#define E       PTBD_PTBD2
#define LED     PTBD_PTBD3
#define D4      PTBD_PTBD4
#define D5      PTBD_PTBD5
#define D6      PTBD_PTBD6
#define D7      PTBD_PTBD7

#define dispRowCount  4
#define dispColCount  16
```

Mezi jednotlivými příkazy je vždy nutné ponechat displeji čas na jejich zpracování. Příklad funkce, která realizuje časovou prodlevu volitelné delky :

```
void wait() {
    unsigned int j;
    for (j=0;j<200;j++) {
        __RESET_WATCHDOG();
    }
}

void waitMore(unsigned int times) {
    unsigned int i;
    for (i=0;i<times;i++) {
        wait();
    }
}
```

□ **Knihovna funkcí pro práci s displejem**

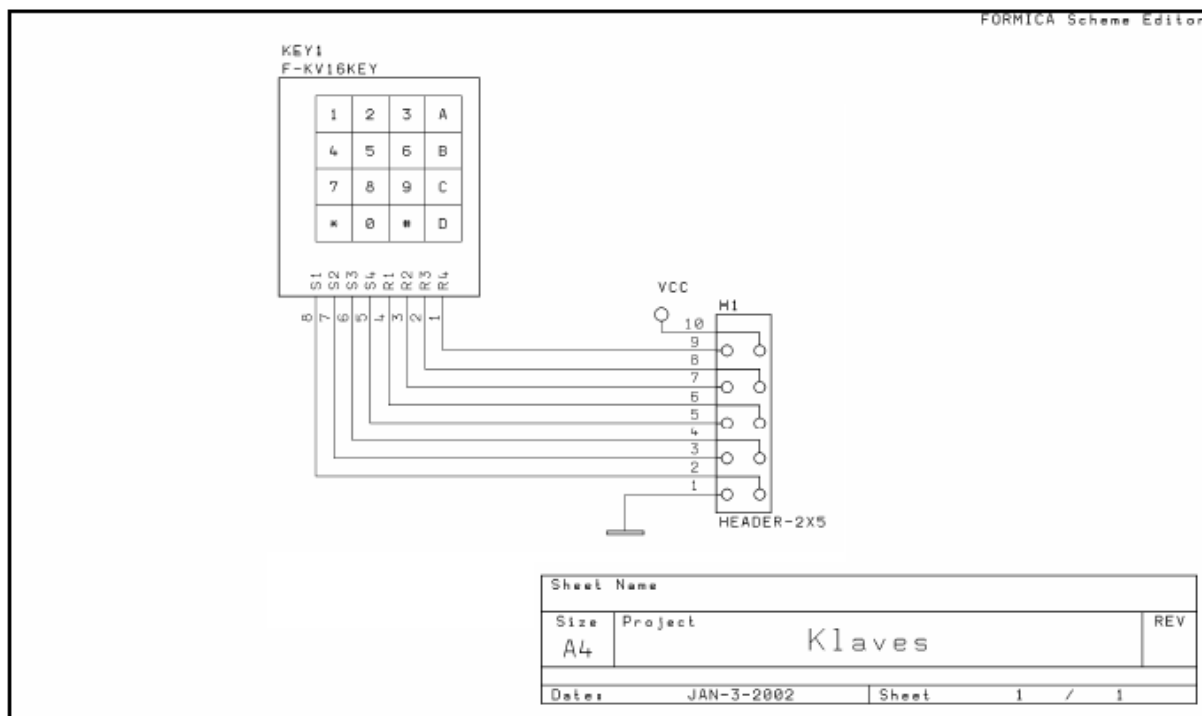
Pro usnadnění práce s displejem byla vytvořena řada knihovních funkcí, které je možno vložit do vlastního programu pomocí příkazu *include*. Popis funkcí, okomentovaný výpis kódu je možno nastudovat v literatuře: Lahoda L.: Návod k použití desky Evb_Keyboard a Evb_Display.

Jedná se o tyto funkce:

- funkce pro inicializaci displeje
- funkce čekání
- knihovna definující konstanty
- funkce pro odeslání bajtu
- funkce pro odeslání příkazu/znaku
- funkce pro výběr řádku
- funkce pro odeslání řetězce nebo matice znaků

9.2 Deska Evb_Keyboard

Rozšiřující deska Evb_Keyboard je určená k připojení se základní deskou Evb_HCS08. Elektrické schéma desky je uvedeno na obr. 9.2.1. Deska se připojuje pomocí plochého desetižilového kabelu k některému z vstupně/výstupních paralelních portů PTA, PTB, PTC nebo PTD+E. Deska je osazena maticovou klávesnicí o čtyřech sloupcích a čtyřech řádcích.



Obr. 9.2.1: Elektrické zapojení desky Evb_Keyboard

□ Ovládání klávesnice

K ovládání klávesnice je k dispozici osm bitů indikující stisk tlačítka v příslušném řádku a příslušném sloupci. Bity S1 – S4 indikují stisk tlačítka ve sloupci 1 – 4 a bity R1 – R4 indikují stisk tlačítka na řádku 1 – 4. Tyto bity jsou určeny výhradně pro čtení a jejich propojení s bity portu A uvádí tabulka 4. Pozor – řádky jsou číslovány zdola nahoru a sloupce zprava doleva.

Tab. 4: Význam jednotlivých bitů pro komunikaci s klávesnicí.

Piny portu PTA	Název bitu	Význam
PTA0	S1	Indikace stisku tlačítka v 1. řádku
PTA1	S2	Indikace stisku tlačítka v 2. řádku
PTA2	S3	Indikace stisku tlačítka v 3. řádku
PTA3	S4	Indikace stisku tlačítka v 4. řádku
PTA4	R1	Indikace stisku tlačítka v 1. sloupci
PTA5	R2	Indikace stisku tlačítka v 2. sloupci
PTA6	R3	Indikace stisku tlačítka v 3. sloupci
PTA7	R4	Indikace stisku tlačítka v 4. sloupci

Pro čtení z klávesnice je zapotřebí nastavit všechny piny portu A pro čtení a následně číst nejdříve řádek a poté soupec. Pokud odečteme hodnotu 1, znamená to, že příslušná klávesa nebyla stisknuta. 0 znamená, že v daném řádku /sloupci byly klávesa stisknuta.

□ Pracovní rámec programu

Pro usnadnění práce si vytvořte definici potřebných konstant a přilinkujte funkci wait, realizující časové zpoždění viz předchozí kapitola.

```
#include <hidef.h> /* for EnableInterrupts macro */
#include "derivative.h" /* include peripheral declarations */

//konstanty pro klávesnici:
#define S1      PTAD_PTAD0
#define S2      PTAD_PTAD1
#define S3      PTAD_PTAD2
#define S4      PTAD_PTAD3
#define R1      PTAD_PTAD4
#define R2      PTAD_PTAD5
#define R3      PTAD_PTAD6
#define R4      PTAD_PTAD7

//další konstanty, např. pro displej

#include "wait.c"
#include "kb.c"
//další include

//další kód
```

Pro detekci, která klávesa byla stisknuta, jsou k dispozici tři funkce:

1. Funkce pro zjištění řádku

```
unsigned char kbGetCol() {

    PTADD= 0b00001111;    //sloupce
    PTAPE= 0b11110000;

    waitMore(5);

    if (!R1) return 4;
    else if (!R2) return 3;
    else if (!R3) return 2;
    else if (!R4) return 1;
    return 0;

}
```

2. Funkce pro zjištění sloupce

```
unsigned char kbPressed=0;

unsigned char kbGetRow() {

    PTADD= 0b11110000;    //řádky
    PTAPE= 0b00001111;

    waitMore(5);

    if (!S1) return 4;
    else if (!S2) return 3;
    else if (!S3) return 2;
    else if (!S4) return 1;
    return 0;

}
```

2. Funkce pro zjištění, který znak odpovídá dané kombinace řádek – sloupec.

```
unsigned char kbGetChar() {  
  
    unsigned char row;  
    unsigned char col;  
  
    if(!(row=kbGetRow()) || !(col=kbGetCol())) {  
        return (kbPressed=0);  
    }  
  
    if (!kbPressed) {  
        kbPressed=1;  
        switch (row) {  
            case 1:  
                switch(col) {  
                    case 1: return '1';  
                    case 2: return '2';  
                    case 3: return '3';  
                    case 4: return 'A';  
                }  
            case 2:  
                switch(col) {  
                    case 1: return '4';  
                    case 2: return '5';  
                    case 3: return '6';  
                    case 4: return 'B';  
                }  
            case 3:  
                switch(col) {  
                    case 1: return '7';  
                    case 2: return '8';  
                    case 3: return '9';  
                    case 4: return 'C';  
                }  
  
            case 4:  
                switch(col) {  
                    case 1: return '*';  
                    case 2: return '0';  
                    case 3: return '#';  
                    case 4: return 'D';  
                }  
        }  
    }  
  
    return 0;  
}
```

Čtení sloupce a řádku se děje sekvenčně protože současné čtení není možné. Čtené bity je nutné nastavit pro čtení a ostatní bity pro zápis. Pro čtené bity se navíc musí nastavit polup enable na 1. Čteme nejdříve horní pak, dolní čtyři bity portu. Mezi nastavením registrů portu a samotným čtením hodnot pinů je nutné ponechat určitou časovou rezervu.

Ukázkový program *snake* spojuje možnosti displeje a klávesnice. Jedná se o zjednodušenou verzi hry „Had“, známou z počátků mobilních telefonů. Hada pevné délky se pohybuje v polích displeje a jehož směr pohybu můžeme měnit stiskem čtyř kláves na klávesnici, která je implementována na rozšiřující desce *Evb_Keyboard*. Na začátku programu je opět nutno definovat konstanty pro klávesnici. Program využívá tři funkce,

teré byly vytvořeny pro detekci stisku klávesy. Funkce kbGetCol zjišťuje sloupec stisknuté klávesy, funkce kbGetRow řádek a funkce kbGetChar je pro určení, jaký znak odpovídá dané kombinaci. V programu bylo zapotřebí ošetřit, aby nedocházelo k opakovanému čtení stisknuté klávesy. Čtení více kláves současně není implementováno. Všechny výše zmíněné programy a funkce se nacházejí v literatuře: Lahoda L.: Návod k použití desky Evb_Keyboard a Evb_Display.



Shrnutí pojmů 9

Princip LCD displeje, řídicí příkazy, inicializace, zápis znaku, definice polohy kurzoru
Maticová klávesnice, čtení pozice zmáčknuté klávesy, čtení logické hodnoty klávesy.



Otázky 9

1. Princip LCD displeje.
2. Připojení desky Evb_Display k základní desce
3. Inicializace displeje.
4. Vyslání řídicího signálu, vyslání znaku.
5. Nastavení kurzoru na definovanou pozici.
6. Princip maticové klávesnice.
7. Propojení desky Evb_Keyboard se základní deskou.
8. Čtení pozice zmáčknuté klávesy.
9. Proč je nutno používat funkci časové prodlevy u obou zařízení.

Úkol 5.1.

Prostudujte si demonstrační programy k desce Evb_Displej a Evb_Keyboard v laboratorních návodech: Lahoda L.: Návod k použití desky Evb_Keyboard a Evb_Display.