

## 8. Rozšiřující deska Evb\_IO a Evb\_Motor



**Čas ke studiu:** 2-3 hodiny



**Cíl** Po prostudování tohoto odstavce budete něco vědět o

- Rozšiřující desce s dalšími paralelními porty
- Rozšiřující desce s motorkem
- Elektrickém zapojení
- Principu činnosti
- Způsobu programování



**Výklad**

### 8.1 Rozšiřující deska Evb\_IO

Rozšiřující deska Evb\_IO je univerzální vstupně-výstupní deska určená k propojení se základní deskou Evb\_HCS08. Deska se připojuje pomocí plochého desetižilového kabelu k některému z vstupně/výstupních paralelních portů PTA, PTB, PTC nebo PTD+E. Piny číslo 10 všech těchto portů slouží jako zdroj napětí, které je využito rovněž jako logický signál pro tlačítka a LED diody. Elektrické schéma desky Evb\_IO je uvedena na obrázku č. 8.1. Deska obsahuje osm tlačítek, osm LED diod, čtyři osmibitové přepínače typu switch a tři konektory pro připojení dalších externích zařízení.

Práci s rozšiřující deskou Evb\_IO byly vysvětlují čtyři dva vzorové programy, které demonstrují správné nastavení paralelních portů, využívají funkce čítačů pro generování časového intervalu a testují schopnost orientace ve dvojkové soustavě.

- Had

Program generuje hada o délce tří diod, který se při každém přerušení od časovače přesune ze své počáteční polohy o jednu pozici zvoleným směrem. Program je založen na instrukci rotace a využití CARRY bitu.

```
interrupt 8 void tpmlovf (void) {  
  
    asm {  
  
        rol PTAD //rotace  
        bcc next //branch if carry bit clear  
  
        lda PTAD  
        ora #0b00000001  
        sta PTAD  
  
        next:  
    }  
}
```

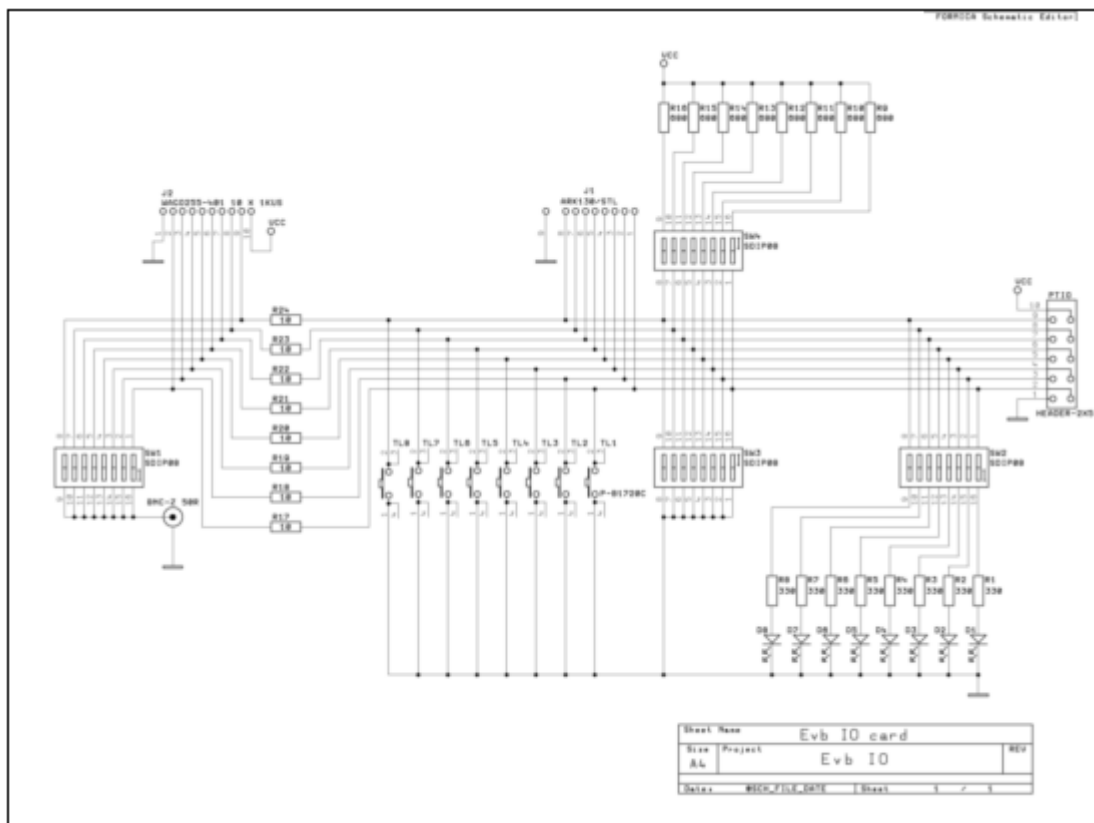
- Tlačítka

Tento jednoduchý program slouží k demonstraci čtení logické hodnoty po stisku a uvolnění tlačítka. Současně se stiskem tlačítka dojde k rozsvícení odpovídající LED diody.

```
void main(void) {
    SOPT_COPE=0; //zákaz watchdog
    PTADD=0b00000000; //nastavení PTA pro čtení

    for(;;) {

        if (PTAD PTAD0==0) {
            asm {
                lda 0b00001111;
            }
        }
        else {
            asm {
                lda 0b11110000;
            }
        }
    }
}
```



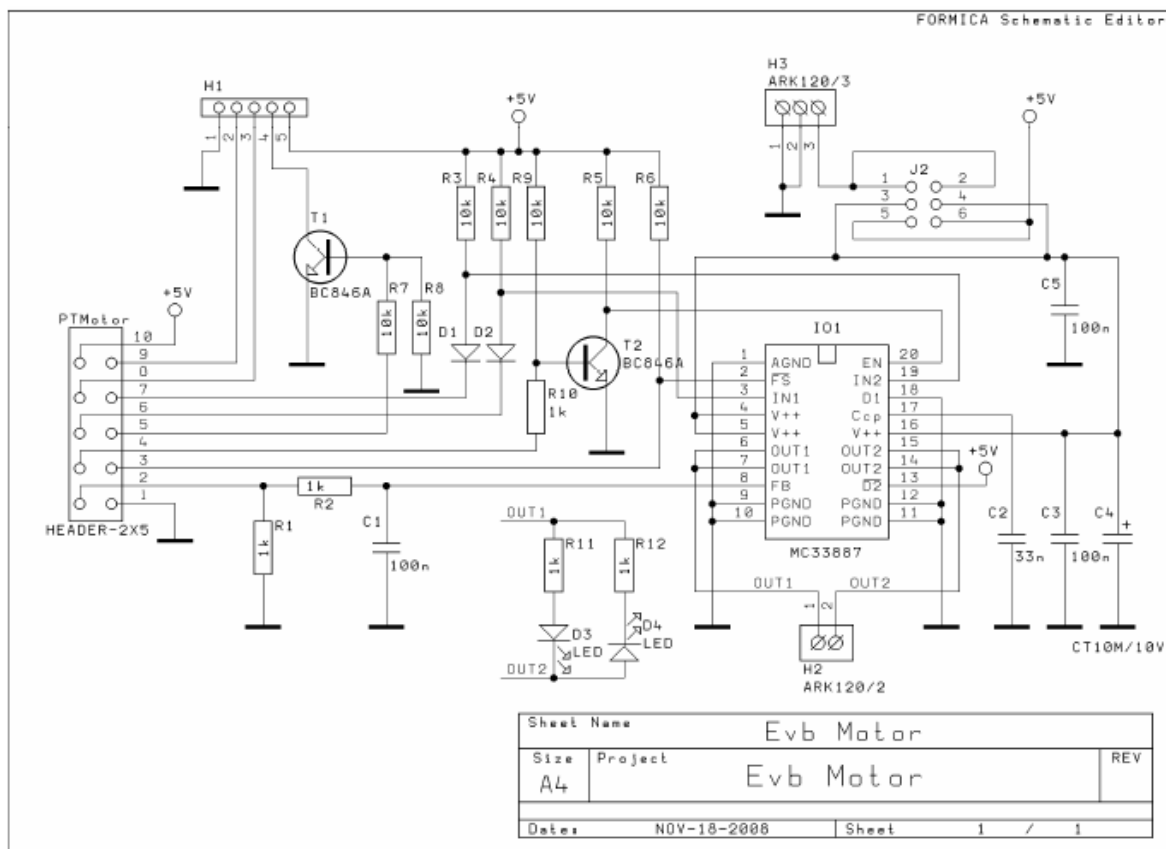
Obr. 8.1:Elektrické schéma rozšiřující desky Evb\_IO

## 8.2 Rozšiřující deska Evb\_MotorI

Tato deska je osazena stejnosměrným motorkem, inkrementálním čidlem a podpůrnými obvody umožňujícími plynulou regulaci rychlosti otáček, směru otáček a měření otáček. Blokové schéma je znázorněno na obrázku 8.2.1. Motor se ovládá pomocí základní desky, ke které je připojen propojením portu PTMotor na straně motoru k některému z vstupně/ výstupních paralelních portů PTA, PTB, PTC nebo PTD+E základní desky. Nejvhodnější volbou je propojení s portem PTA, jehož piny nesdílí žádnou periférii implementovanou na základní desce.

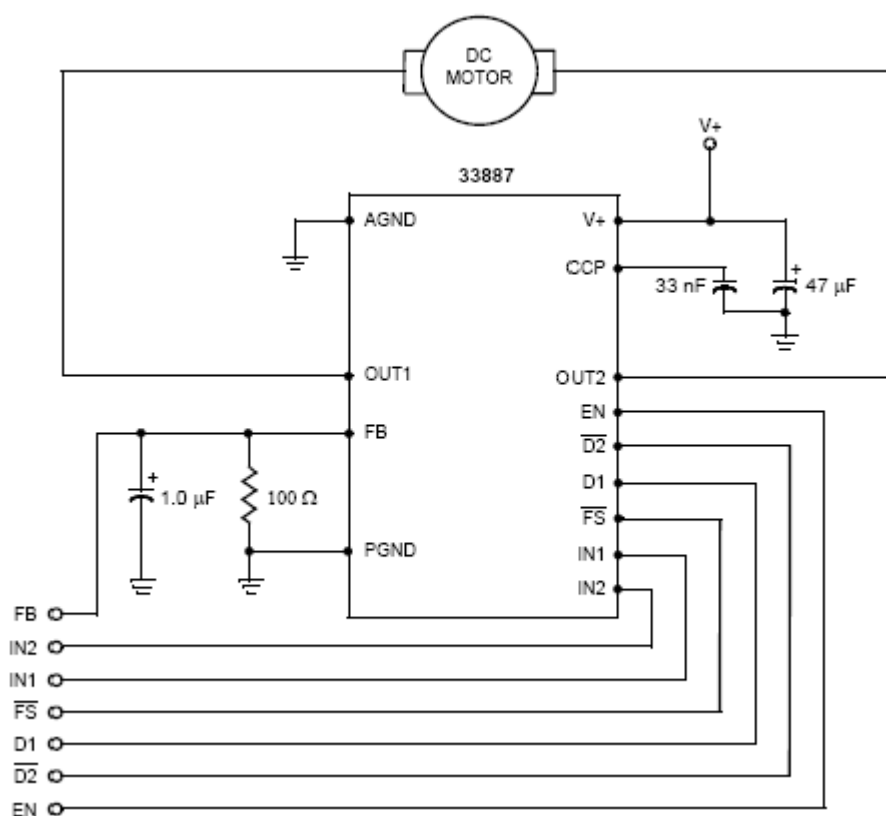
Napájení motorku je zajištěno v propojovacím kabelu přivedením napětí 5V ze základní desky EvbHCS08. Důležité je správné nastavení přepínače, který umožňuje výběr mezi dvěma napěťovými zdroji 3,7V a 5V.

Funkce krokování motorku je realizována pomocí optické brány složené z vysílače realizovaného infračervenou LED diodou a přijímače optického signálu realizovaného fotodiodou. Na ose motorku, kolno na spojnici mezi vysílačem a přijímačem optické brány je umístěn plastový terč, na němž se střídají průhledné a neprůhledné pruhy. Rotace terče způsobuje střídavé otevírání a zavírání optické brány, čímž dochází ke generování výstupního signálu, který je možno využít k měření otáček motorku. Tyto pulsy je možno rovněž využít jako externí hodiny časovače.



Obr. 8.2.1: Elektrické schéma rozšiřující desky Evb\_Motor

Logické jádro desky tvoří integrovaný obvod IO1. Tento obvod po uvedení do aktivního módu začne převádět vstupní signály na signály výstupní, kterými se řídí spouštění a zastavování otáček motorku, směr otáčení, případně signalizuje chybový stav zařízení. Následující obrázek č. 8.2.2 demonstruje typické aplikační zapojení integrovaného obvodu IO1.



Obr. 8.2.2: Aplikační schéma integrovaného obvodu IO1.

Tab.1: Tabulka pravdivostních hodnot pro vybrané hodnoty vstupů integrovaného obvodu IO1. [2]

Stav motorku	Vstupní piny					Výstupní piny		
	EN	D1	$\overline{D2}$	IN1	IN2	$\overline{FS}$	OUT1	OUT2
Točí vpřed	1	0	1	1	0	1	1	0
Točí vzad	1	0	1	0	1	1	0	1
Stojí	1	0	1	0	0	1	0	0
Stojí	1	0	1	1	1	1	1	1
Režim spánku	0	x	x	x	x	1	0	0

pozn.: x = nezáleží na stavu

V rámci semestrální práce byly pro rozšiřující desku Evb\_MOTOR vytvořeny dva demonstrační programy:

### **1. Krokování motorku**

Program je ovládán pomocí tlačítek a trimru ze základní desky. Stiskem tlačítka TL1 dojde k otočení motorku o zvolený počet kroků vpřed, stiskem tlačítka TL2 dojde k otočení o zvolený počet kroků vzad. Počet kroků pro otočení se počítá na základě polohy trimru, jehož aktuální poloha se získává čtením vstupu A/D převodníku. Zároveň je nastaveno přerušení od obou tlačítek. V programu pro obsluhu přerušení se pak nastaví požadovaný směr otočení. Tento program demonstruje především postup pro ovládání otáček krokového motorku, rovněž se v něm uplatňují techniky programování periférií základní desky. Protože Evb\_Motor je pouze rozšiřující modul, nelze jej od základní desky nikterak oddělit, naopak je vhodné a žádoucí propojování možností obou desek.

### **2. Regulace rychlosti otáčení**

Program umožňuje spojitou regulaci otáček motorku. Tomuto účelu je využit trimr na základní desce. V krajní poloze trimru, která odpovídá napětí 0V, se motor točí maximální rychlostí vpřed, v druhé krajní poloze, která odpovídá napětí 5V, se točí maximální rychlostí vzad. Pokud se trimr nachází uprostřed krajních poloh, motor stojí.

Okomentované kódy programu jsou uvedeny níže. Slouží k pochopení principu programování

- směru otáček motorku
- krokování otáčení o nejmenší možný krok pootočení, nebo o jeho násobky
- spojitou regulaci rychlosti otáčení s využitím A/D převodníku
- měření rychlosti otáčení inkrementálním čidlem

```

void main(void) {

    int step_made=0;

    /*** vypnutí watchdog ***/
    SOPT_COPE=0; //SOPT=System Options register, COPE=Watchdog Enable

    /*** PORT PTA ***/
    //ovládá desku Evb Motor
    PTAD=0b00111000;
    PTADD=0b00111100;

    /*** PORT PTD ***/
    //zachytává stisk tlačítka TL2
    PTDPE_PTDPE3=1; //pullup enable na PTD3 - tlačítko TL2

    /*** PIN IRQ ***/
    //zachytává stisk tlačítka TL1
    IRQSC_IRQPE=1; //povolí IRQ pin (IRQ Pin Enable)
    IRQSC_IRQEDG=0; //bude detekovat sestupnou hranu (IRQ Edge Select)
    IRQSC_IRQIE=1; //povolí přerušení (IRQ Interrupt Enable)

    /*** AD PŘEVODNÍK ***/
    ATD1PE_ATDPE0=1; //ATD Pin Enable: povolení pinu 0 jako vstupu do AD
    ATD1C_ATDPU=1; //ATD Power Up: zapne AD převodník
    ATD1C_RESS=1; //ATD Result: výsledek převodu bude 8bitový
    ATD1SC=0b00000000; //ADT Status and Control: povolení převodu a výběr kanálu 0

    EnableInterrupts; //povolení přerušení

    for(;;) {
        //nekonečná smyčka, zachytává se přerušení od pinu IRQ a AD převodníku

        /*** OBSLUHA TLAČÍTKA TL2 ***/
        if(!PTDD_PTDD3 && !step_made) { //stisk tlačítka TL1 vyvolá krok motorku

            Move(steps); //motorek provede nastavený počet kroků
            step_made=1; //příznak, že motorek už provedl krok

        }

        if(PTDD_PTDD3) {
            step_made=0; //po uvolnění tlačítka TL2 se nuluje příznak provedeného kroku
        }
    }
}

```

```

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

int speed=10; //globální proměnná určující rychlost otáčení

interrupt 8 void tpm1ovf (void) {
//přerušeni při skončení periody PWM

    asm {
        bclr 7,TPM1SC //nulování příznaku přerušeni
    }

    //spuštění motorku
    if (speed>=122 && speed<=132) {
        //pokud trimr uprostřed (s jistou tolerancí), motor stojí
        PTAD_PTAD5=1;
        PTAD_PTAD4=1;
    }
    else if (speed<122) {
        //toč vzad
        PTAD_PTAD4=0;
        PTAD_PTAD5=1;
    }
    else if (speed>132) {
        //toč vpřed
        PTAD_PTAD4=1;
        PTAD_PTAD5=0;
    }
}

interrupt 5 void tpm1cho (void) {
//přerušeni při skončení pulzu PWM

    asm {
        bclr 7,TPM1COSC //nulování příznaku přerušeni
    }

    //zastavení motorku
    PTAD_PTAD4=1;
    PTAD_PTAD5=1;
}

interrupt 23 void atdcoco (void) {
//přerušeni při skončení AD převodu

speed=ATD1RH; //čtení výsledku

    if (speed<127) {
        TPM1COV=16*2*(-speed+127); // nastavení délky pulzu PWM, točení vzad
    }
    else {
        TPM1COV=16*2*(speed-127); //nastavení délky pulzu PWM, točení vpřed
    }

    ATD1SC=0b01000000; //povolení dalšího převodu
}

```

```

void main(void) {

    /*** vypnutí watchdogu ***/
    SOPT_COPE=0;

    /*** ČASOVAČ ***/
    TPM1SC=0b01001110; //interrupt enable, clock: busclk, prescaler: 64
    TPM1MOD=0x1000; //modulo regist určuje periodu cyklu PWM (0x1000=#4096)
    //kanál 0:
    TPM1COSC=0b01101000; //interrupt enable, mode: edge-aligned PWM
    TPM1COV=0x0010;

    /*** AD PŘEVODNÍK ***/
    ATD1PE_ATDPE0=1; //ATD Pin Enable: povolení pinu 0 jako vstupu do AD
    ATD1C_ATDPU=1; //ATD Power Up: zapne AD převodník
    ATD1C_RES8=1; //ATD Result: výsledek převodu bude 8bitový
    ATD1SC=0b01000000; //ADTSC: povolení přerušení,převodu a výběr kanálu 0

    /*** PORT PTA ***/
    //ovládá desku Evb Motor
    PTAD=0b00110000; //povolení motoru, nastavení IN1 a IN2 na 1
    PTADD=0b00111100; //nastavení pinů 2 až 5 pro zápis

    EnableInterrupts; //povolení přerušení

    for(;;) {
        //nekonečná smyčka, zachytávají se přerušení od časovače a AD převodníku
    }
}

```



### Shrnutí pojmů 3

Paralelní porty – vstupy (tlačítka), výstupy (LED diody), binární čísla. Stejnosměrný motor, iiniciace, směr otáčení, krokování, spojitá regulace. Měření otáček, inkrementální čidlo, optická brána.



### Otázky 3

1. Jak se pracuje s paralelními porty
2. Propojení rozšiřující desky Evb\_IO se základní deskou Evb\_HCS08.
3. Jaký je rozdíl mezi paralelním vstupem a výstupem. Jak jsou technicky reprezentovány.
4. Propojení rozšiřující desky Evb\_Motor se základní deskou Evb\_HCS08.
5. Princip regulace otáček u motoru.
6. Nastavení směru otáčení.
7. Co to je inkrementální čidlo, jak pracuje.



## Úkol 5.1.

1. Pro Rozšiřující desku Evb\_IO vytvořte program „Binární čísla“

Program postupně zobrazuje ve zvoleném časovém okamžiku čísla od nuly do 255 v binárním kódu s využitím jedné sady osmi LED diod na rozšiřující desce. Jedná se o jednoduchou funkci binárního přičítání jedničky. Po dosažení maximální hodnoty se číslo nuluje a program se cyklicky opakuje.

2. Pro Rozšiřující desku Evb\_IO vytvořte program „Binární čísla“

Program pomocí LED diod na desce Evb\_IO vizualizuje padání cihel na hromadu. Na začátku programu jsou všechny diody zhasnuté. Poté se postupně padající „cihly“ vrství na hromadu až dojde k rozsvícení všech diod, hromada je plná a program se cyklicky opakuje. Logická stavba tohoto programu je složitější než u předchozích dvou. Program pracuje s binárními instrukcemi logických šiftů, logickým součtem a s řadou binárních testovacích instrukcí.

3. Pro Rozšiřující desku Evb\_IO vytvořte program „Had“

Program generuje jedno rozsvícenou diodu, která se v určitých časových okamžicích generovaných přerušením od časovače přesune ze své počáteční polohy o jednu pozici zvoleným směrem.

Tento program modifikujte inverzním rozsvícením diod. Pohybující had bude tvořen zhaslou diodou v poli rozsvícených diod.