

Vysoká škola chemicko-technologická v Praze
Fakulta chemicko-inženýrská
Ústav počítačové a řídicí techniky

Aplikace mikroprocesorů



KROKOVÝ MOTOREK

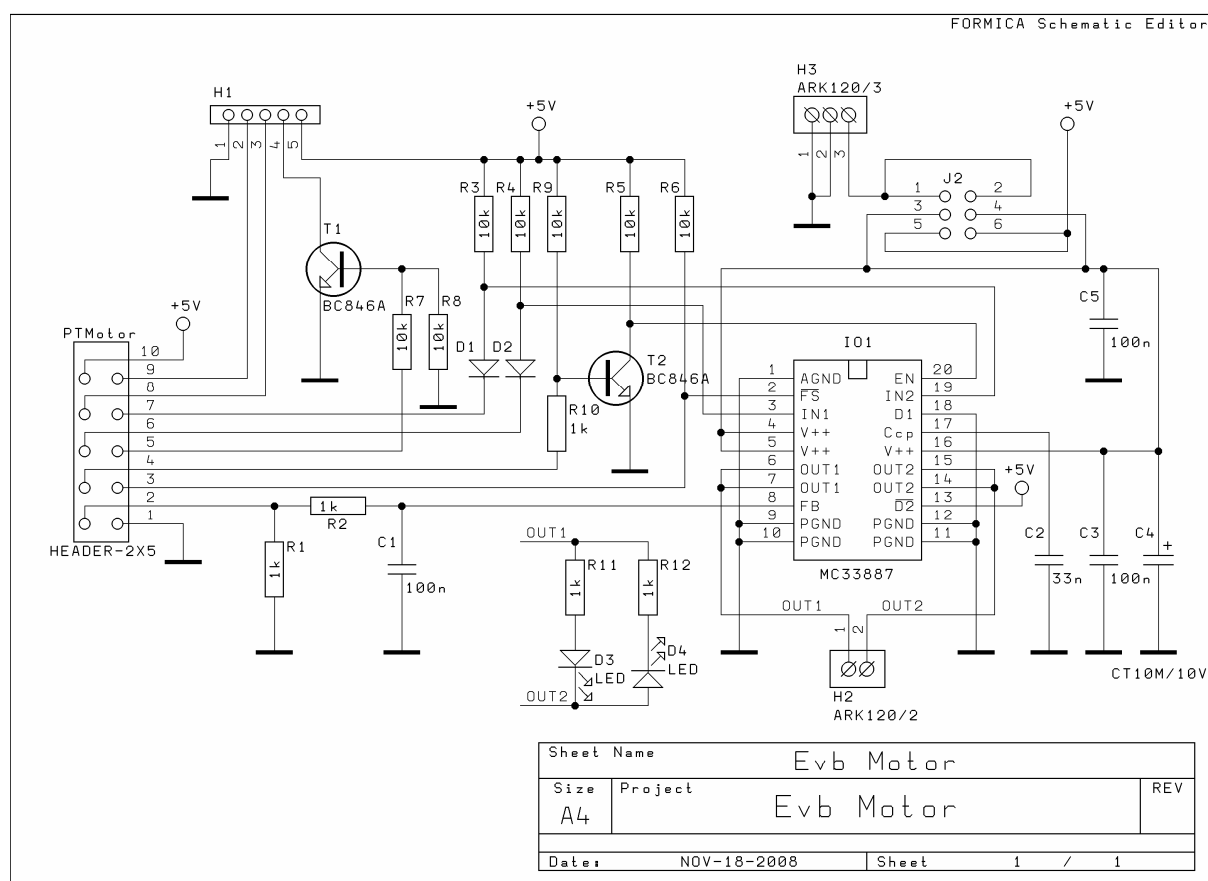
Návod k použití

Obsah

1	DESKA EVB MOTOR.....	2
1.1	PROPOJENÍ DESKY EVB MOTOR S DESKOU EVBHCS08	2
1.2	NAPÁJENÍ DESKY	3
1.3	OBVOD OPTICKÉ BRÁNY EVB MOTOR INKREMENTAL.....	4
1.4	INTEGROVANÝ OBVOD IO1 (H-BRIDGE WITH LOAD CURRENT FEEDBACK).....	4
2	VZOROVÉ PROGRAMY	7
2.1	KROKOVÁNÍ MOTORKU.....	7
2.2	REGULACE RYCHLOSTI OTÁČENÍ.....	9
3	LITERATURA	11

1 Deska Evb Motor

Deska Evb Motor je základní deska sloužící k ovládání krokového motorku, který je na ní umístěn. Nejedná se o klasický krokový motor ovládaný řízením magnetického pole statoru, nýbrž o obyčejný servomotor, u něhož je funkce krokování realizována pomocí optické brány složené z vysílače (infračervená LED dioda) a přijímače (fotodioda) optického signálu. Na ose rotoru motorku, kolmo na spojnici mezi vysílačem a přijímačem optické brány, je umístěn plastový terč, na němž se střídají průhledné a neprůhledné pruhy. Pokud motorek rotuje, způsobuje rotace terče střídavé otevírání a zavírání optické brány, čímž dochází ke generování výstupního signálu, který je možné využít k řízení či měření otáček motorku. Na *obr. 1* je znázorněno schéma propojení jednotlivých součástí desky, které si blíže popíšeme. V dalším textu budeme předpokládat, že čtenář je již blíže seznámen s deskou EvbHCS08 a programováním procesoru MC9S08GT32A v prostředí CodeWarrior.



Obr. 1. Schéma desky Evb Motor

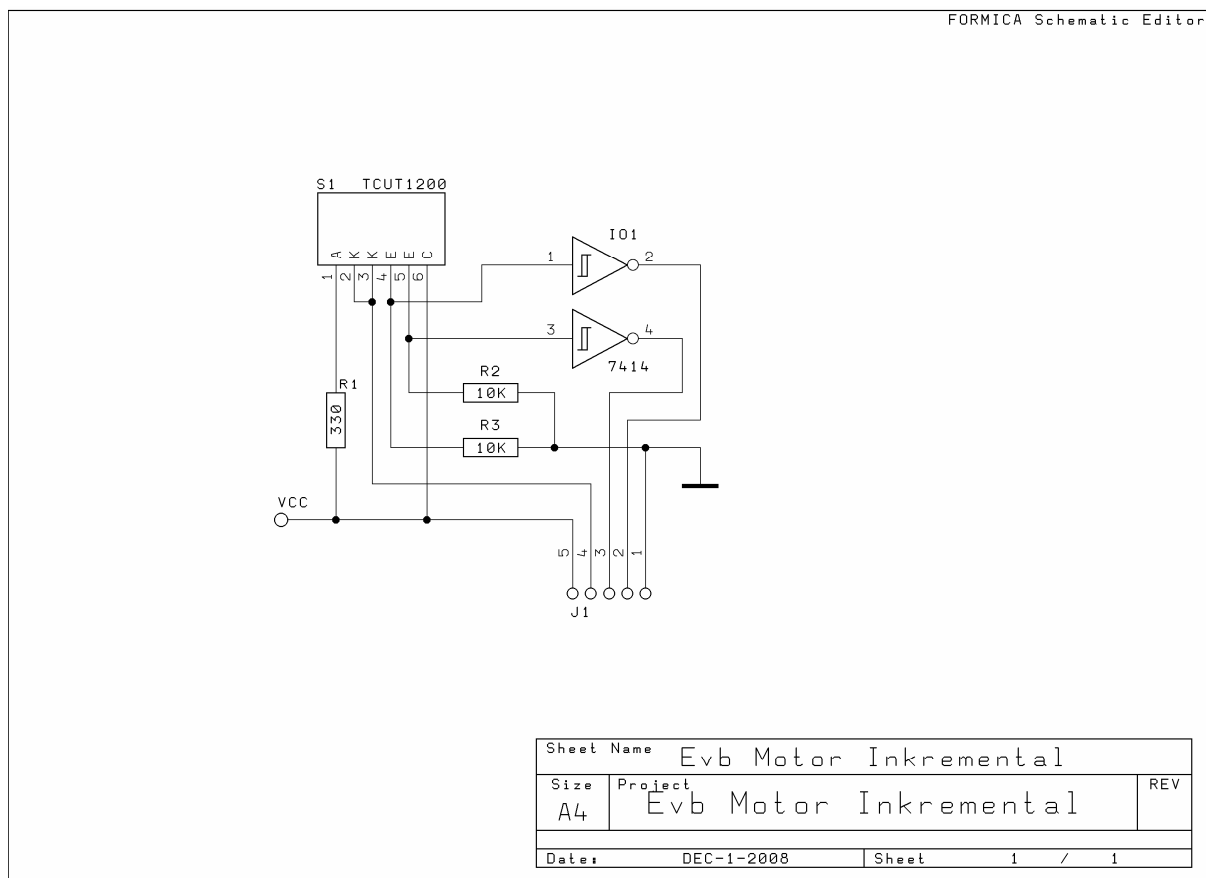
1.1 Propojení desky Evb Motor s deskou EvbHCS08

Desku Evb Motor je nutné ovládat přes desku EvbHCS08. Tyto desky je možné vzájemně propojit, a to propojením portu **PTMotor** desky Evb Motor (*obr. 1* vlevo) s některým z vstupně/výstupních paralelních portů PTA, PTB, PTC nebo PTD+E desky

EvbHCS08. Je dobré mít na vědomí, že některé piny paralelních portů desky EvbHCS08 jsou sdíleny s vstupy/výstupy vestavěných periférií této desky. Například pokud bychom chtěli využívat k ovládní motorku tlačítko TL2, není možné desku propojit přes port PTD+E, neboť na pin PTD3 by kromě tlačítka byl zároveň přiveden jeden z výstupů optobrány motorku, což by vedlo ke generování falešných stisknutí tlačítka. Podobně připojení přes port PTDB by mohlo vést k znemožnění použití AD převodníku, jehož vstupy jsou sdíleny s piny PTB0 a PTB1 (této skutečnosti lze však i využít, jak se dozvíme dále). U portu PTC je pin PTC7 nahrazen pinem IRQ. Nejvhodnější volnou je proto propojit desku přes port PTA, jehož piny nejsou sdíleny s žádnou z periférií vhodnou k ovládní motorku.

1.2 Napájení desky

Desku Evb Motor je nutné napájet samostatným zdrojem proudu připojeným na svorkovnici **H3** (obr. 1 vpravo nahoře). K tomu je možné využít adaptér dostupný v laboratoři nastavený na napětí 4,5 V, jehož plusový kabel (označený bílým pruhem) se připojí na svorku V+ a minusový kabel (černý) na jednu ze svorek GND. Napětí přivedené z desky EvbHCS08 (konektor **PTMotor**, pin 10) je pro provoz motoru nedostačující a slouží pouze jako logický signál, který využívají ostatní součásti desky Evb Motor. Důležitým krokem je správné nastavení jumperu **J2**, který umožňuje výběr mezi dvěma zmíněnými zdroji. Pro výběr externího zdroje proudu je nutné propojit piny 1-3 a 2-4.



Obr. 2. Schéma obvodu zajišťujícího funkci optické brány

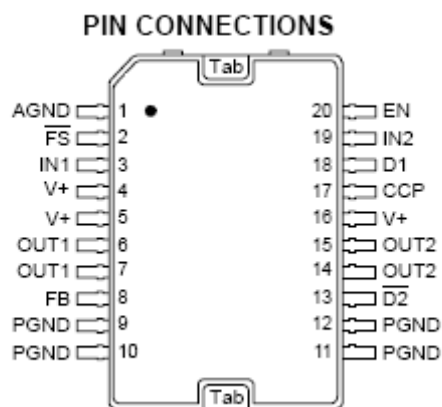
1.3 Obvod optické brány Evb Motor Inkremental

Tento obvod (*obr. 2*) zahrnuje optickou bránu a je k desce Evb Motor připojen přes konektor **H1** (*obr. 1* vlevo nahoře). Na pin 5 je přivedeno napětí +5V, pin 1 je přiveden na zem. Pin 4 slouží zapnutí/vypnutí brány. Zapnutí brány je realizováno propojením pinu 4 se zemí přes tranzistor **T1**. Aby tranzistorem mohl procházet proud, je potřeba přivést kladné napětí na jeho bázi. Jelikož báze tranzistoru je propojena s pinem 5 konektoru **PTMotor**, je pro zapnutí brány nutné zapsat logickou 1 na odpovídající pin paralelního portu desky EvbHCS08 (v případě použití portu PTA se jedná o pin PTA3).

Piny 2 a 3 konektoru **H1** slouží jako výstupy optické brány a čtením jejich hodnoty je možné určit, zda je brána v otevřeném, nebo zavřeném stavu. Pokud se motorek otáčí, jsou přes tyto piny vysílány pulzy, které je možno využít např. k určování otáček motorku nebo k jeho krokování. Piny 2 a 3 jsou propojeny s piny 9 a 8 konektoru **PTMotor**. To znamená, že v případě propojení desek přes port PTA lze jejich hodnoty číst přes piny PTA7 a PTA6. Z uvedeného vyplývá zajímavá možnost, a tou je užití signálu generovaného optickou bránou jako externích hodin časovače TPM2 procesoru HCS08 propojením desek přes port PTD+E, jehož piny PTD4 a PTD3 jsou sdíleny s kanály TPM2CH1 a TPM2CH0 časovače TPM2.

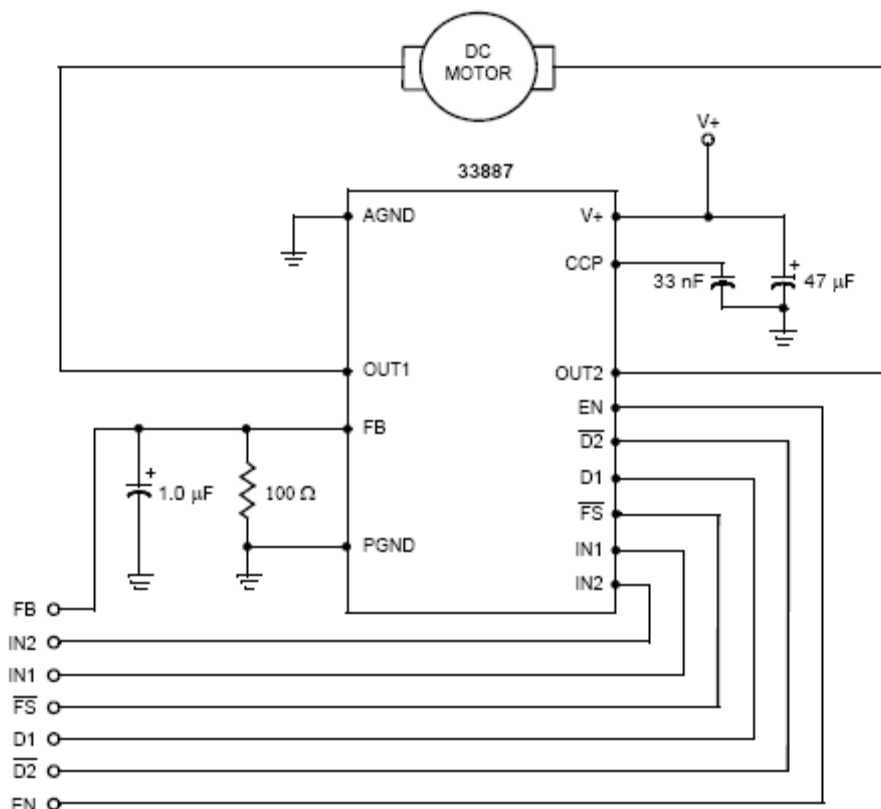
1.4 Integrovaný obvod IO1 (H-Bridge with Load Current Feedback)

Tento obvod tvoří jádro desky Evb Motor, které umožňuje ovládat motorek. Jeho piny jsou znázorněny na *obr. 3*, typické aplikační schéma na *obr. 4*.



Obr. 3. Piny integrovaného obvodu IO1

Pro komunikaci s obvodem slouží vstupy EN, IN1, IN2, D1 a $\overline{D2}$ a výstupy FB a \overline{FS} . Funkce těchto pinů je popsána v dalším textu. Na základě hodnot vstupů nastavuje obvod napětí na výstupních pinech OUT1 a OUT2, které jsou přes konektor **H2** desky Evb Motor připojeny na póly servomotoru. V případě, že je na piny OUT1 a OUT2 vystaveno rozdílné napětí, motorek se začne otáčet. Na desce je zároveň realizovaná indikace směru otáčení LED diodami **D3** a **D4** (*obr. 1* dole uprostřed). Způsob, jakým obvod převádí hodnoty vstupů na výstupy, vyplývá s tabulky pravdivostních hodnot *tab. 1*.



Obr. 4. Typické aplikační schéma integrovaného obvodu IO1

Pin EN (Enable)

Tento pin slouží k zapnutí/vypnutí zařízení. Pokud je na tento pin přivedeno napětí + 5V (logická 1), přepne se zařízení do aktivního módu a začne převádět vstupní signály na výstupní. Na desce je tento pin připojen na kladné napětí +5V přes rezistor **R5** (obr. 1 nahoře uprostřed), ovšem zároveň je propojen se zemí přes tranzistor **T2**, jehož báze je přivedena na kladné napětí +5V přes rezistor **R9**, což znamená, že z kolektoru na emitor tranzistoru může procházet proud. V tomto výchozím stavu je tedy na pinu EN logická 0, neboť průchodem proudu přes rezistor **R5** dochází k úbytku veškerého napětí.

Abychom na pin EN přivedli signál logická 1, musíme zajistit, aby přes tranzistor **T2** neprocházel proud (tj. odpojit pin EN od země). Tento stav nastane tehdy, pokud na bázi tranzistoru bude nulové napětí. Báze tranzistoru je vyvedena na pin 3 konektoru **PTMotor**, ke kterému můžeme přistupovat přes pin PTA2 (v případě propojení desek Evb Motor a EvbHCS08 přes port PTA). Zapišeme-li na pin PTA2 hodnotu logická 1, bude na bázi tranzistoru setrvávat kladné napětí a na pinu EN zůstane logická 0. Zapišeme-li na pin PTA2 hodnotu logická 0, dojde k propojení tohoto pinu se zemí a přes rezistor **R9** začne procházet proud, který na průchod přes tento rezistor spotřebuje veškeré napětí. Napětí na bázi tranzistoru **T2** poklesne na nulu a dojde k odpojení pinu EN od země.

Z uvedeného vyplývá, že chceme-li zařízení aktivovat, musíme povolit pin PTA2 pro zápis a zapsat na něj hodnotu logická 0.

Piny IN1 a IN2 (Logic input control)

Za předpokladu, že se zařízení nachází v aktivním režimu, slouží tyto vstupní piny k zastavování a spouštění rotace motorku zvoleným směrem. Logická hodnota těchto vstupních pinů přímo určuje logickou hodnotu výstupních pinů OUT1 a OUT2 (viz tab. 1). Standardně je na oba piny přes rezistory **R4** a **R5** přivedeno kladné napětí +5V (logická 1) a motorek stojí. Aby bylo možné tuto hodnotu ovládat, jsou piny IN1 a IN2 vyvedeny přes diody **D2** a **D1** na piny 6 a 7 konektoru **PTMotor**, což odpovídá pinům PTA4 a PTA5 portu PTA desky EvbHCS08. Zápis hodnoty logická 1 na pin PTA4 nebo PTA5 nemá žádný efekt, neboť tento signál neprojde závěrným směr diod. Naopak zapíšeme-li na pin PTA4 nebo PTA5 hodnotu logická 0, propojí se tento pin se zemí a přes rezistor **R3** nebo **R4** začne procházet proud, čímž dojde k poklesu napětí na pinu IN1 nebo IN2 na nulu.

Shrneme-li předchozí odstavec, tak logická 1 na pinu PTA4, resp. PTA5 znamená logickou 1 na pinu IN1, resp. IN2 a naopak.

Piny D1 a $\overline{D2}$ (Disable)

Tyto piny jsou navrženy taktéž k ovládání výstupů OUT1 a OUT2, nicméně konstrukce desky Evb Motor znemožňuje měnit jejich hodnotu, neboť pin D1 je napevno propojen se zemí (logická 0) a na pin $\overline{D2}$ je napevno přivedeno napětí +5V (logická 1).

Pin \overline{FS} (Fault status)

Jedná se o výstupní pin, jehož hodnota logická 0 signalizuje chybový stav zařízení. Je vyveden na pin 3 konektoru PTMotor, jemuž odpovídá pin PTA1 portu PTA.

Pin FB (Feedback for H-Bridge)

Výstupní pin, který umožňuje sledovat proud procházející mezi piny OUT1 a OUT2 (tzn. přes motorek). Napětí vystavované na tento pin má hodnotu 1/375 zmíněného proudu. Pin FB je propojen s pinem 2 portu PTMotor, jemuž odpovídá pin PTA0 portu PTA. Propojíme-li desky Evb Motor a EvbHCS08 přes port PTB, můžeme využít skutečnosti, že piny tohoto portu jsou sdíleny s kanály A/D převodníku procesoru MCS9S08GT32A a převádět tak analogovou hodnotu pinu FB na digitální signál a monitorovat otáčky motorku.

Tab. 1. Tabulka pravdivostních hodnot pro vybrané hodnoty vstupů integrovaného obvodu IO1

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

2 Vzorové programy

2.1 Krokování motorku

Zadání

Úkolem je krokovat motorek zvoleným směrem. Stiskem tlačítka TL1 dojde k otočení motorku o zvolený počet kroků dopředu, stiskem tlačítka TL2 to samé opačným směrem. Počet kroků se bude nastavovat pomocí trimru TR1.

Řešení

Propojíme port PTMotor desky Evb Motor s portem PTA desky EvbHCS08 pomocí přiloženého kabelu. Na svorkovnici H3 desky Evb Motor připojíme externí zdroj proudu dle odst. 1.2. Sestavíme program v prostředí Freescale Code Warrior. Program provede nastavení registrů procesoru a poté probíhá v nekonečné smyčce, ve které čte pomocí AD převodníku aktuální polohu trimru a ověřuje stav tlačítka TL2. Zároveň jsou zachytávána přerušení od tlačítka TL1. Pokud dojde ke stisku některého z tlačítek, otočí se motorek správným směrem o k kroků, přičemž k je vypočítáno na základě polohy trimru získané AD převodníkem.

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

int steps=1; //globální proměnná určující počet kroků
int max_steps=100; //rozsah AD převodníku (0 až 255) se přečte na
                    //(0 až max_steps) kroků motorku

void Move(int number_of_steps) {
//funkce Move provede zadaný počet kroků motorku směrem určeným znaménkem

    int i;
    int backward=0;
    int init_gate_state;

    if (number_of_steps<0) {
        number_of_steps=-number_of_steps;
        backward=1;
    }

    for(i=1;i<=number_of_steps;i++) {

        init_gate_state=PTAD_PTAD6; //poč.stav optobrány:
                                    //0=brána otevřena; 1=brána uzavřena
        while (PTAD_PTAD6==init_gate_state) { //dokud nedošlo ke změně stavu brány

            PTAD_PTAD5=!backward; //spustě otáčení vzad/vpřed
            PTAD_PTAD4=backward;

        }

        PTAD_PTAD5=1; //zastav otáčení
        PTAD_PTAD4=1;

    }
}
```



```

interrupt 2 void IRQpin (void) {
//přerušeni od IRQ pinu

    asm {
        bset 2,IRQSC //IRQ acknowledge - potvrzení příjmu přerušeni
    }

    Move(-steps);
}

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

    steps=ATD1RH; //čtení výsledku
    steps=max_steps*steps/255+1; //výsledek AD převodu se pře počte na počet kroků
    ATD1SC=0b00000000; //nulování příznaku přerušeni, povolení dalšího převodu
}

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 přerušeni (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_RES8=1; //ATD Result: výsledek převodu bude 8bitový
    ATD1SC=0b00000000; //ADT Status and Control: povolení převodu a výběr kanálu 0

    EnableInterrupts; //povolení přerušeni

    for(;;) {
        //nekonečná smyčka, zachytává se přerušeni 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
        }
    }
}

```

2.2 Regulace rychlosti otáčení

Zadání

Úkolem je regulovat rychlost otáčení pomocí trimru TR1. V krajní poloze trimru odpovídající napětí 0V se motorek bude točit maximální rychlostí vpřed, v krajní poloze trimru odpovídající napětí 5V maximální rychlostí vzad. Pokud se hodnota trimru bude nacházet uprostřed (s jistou tolerancí), motorek bude stát. Mezi těmito třemi polohami se bude rychlost otáčení lineárně měnit.

Řešení

Propojíme port PTMotor desky Evb Motor s portem PTA desky EvbHCS08 pomocí přiloženého kabelu. Na svorkovnici H3 desky Evb Motor připojíme externí zdroj proudu dle odst. 1.2. Sestavíme program v prostředí Freescale Code Warrior. Rychlost otáčení motorku nemůžeme ovládat regulací napětí, protože jsou k dispozici pouze 2 napěťové stavy – vypnuto/zapnuto. Abychom docílili požadovaného efektu, musíme motorek střídavě zapínat a vypínat, přičemž rychlost otáčení bude úměrná době trvání zapnutého stavu. K tomu využijeme časovou funkci PWM časovače TPMx – pulzně šířkovou modulaci se zarovnáním na hranu. Tato funkce umožňuje generovat pulzy s pevnou periodou a proměnnou šířkou pulzu. Do časovacího registru časovače (TPMxMOD) uložíme pevnou délku periody a do kanálového časovacího registru zvoleného kanálu (TPMxCnV) šířku pulzu odpovídajícího sepnutému stavu motorku (tu budeme poté měnit pomocí trimru). Na začátku každé periody vyvolá časovač přerušení „overflow“ a v rámci obsluhy tohoto přerušení spustíme motorek. Po uplynutí doby pulzu je generováno přerušení „output compare“, při kterém motorek vypneme a v tomto stavu ho ponecháme až do skončení periody. Čím delší bude šířka pulzu, tím rychleji se bude motorek otáčet a naopak. Už zbývá jen převést výsledek AD převodu (tzn. polohu trimru) na šířku pulzu PWM cyklu. Při tom musíme mít na paměti, že šířka pulzu musí být násobkem 1/256 periody. Pokud zvolíme periodu např. 4096, šířka pulzu musí být násobkem $4096/256 = 16$ apod.

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

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

interrupt 8 void tpmlovf (void) {
//přerušení při skončení periody PWM

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

    //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 tpm1ch0 (void) {
//přerušeni při skončení pulzu PWM

    asm {
        bclr 7,TPM1C0SC //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) {
        TPM1C0V=16*2*(-speed+127); // nastavení délky pulzu PWM, točení vzad
    }
    else {
        TPM1C0V=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:
    TPM1C0SC=0b01101000; //interrupt enable, mode: edge-aligned PWM
    TPM1C0V=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 přerušeni,převodu a výběr kanálu 0

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

    EnableInterrupts; //povolení přerušeni

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

```

3 Literatura

- [1] Freescale Semiconductor Technical Data – A H-Bridge with Load Current Feedback
- [2] Freescale Semiconductor MC9S08GB/GT Data Sheet