

Modul řízení otáček stejnosměrného motorku

Výukový modul pro předmět Programování Mikropočítačů

Popis modulu – úvod

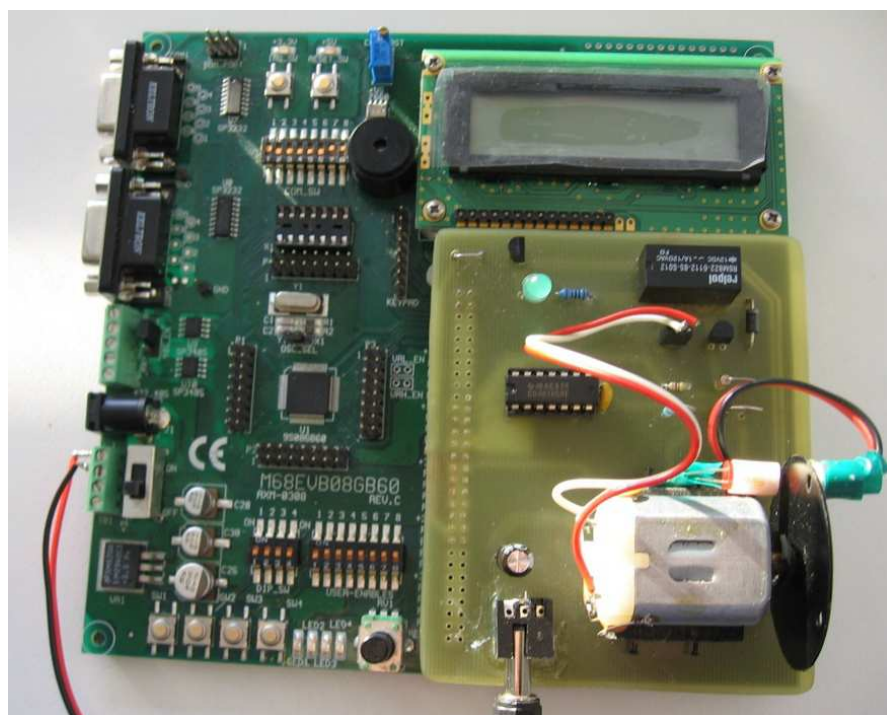
Jedná se o rozšiřující modul pro výukový kit. Modul obsahuje stejnosměrný motorek osazený zařízením pro optické snímání otáček, tj. LED a snímacím fototranzistorem. Modul umožňuje řídit otáčky motoru pomocí PWM modulace (využití PWM funkce časovače), přepínání směru otáčení (diskrétní výstup) a snímání skutečných otáček motorku (využití input capture funkce časovače)

Pro přepínání směru otáčení motorku je použito relé. Relé je ovládáno softwarově pomocí výstupu z mikropočítače. Pro indikaci směru otáček slouží dioda LED 2 (zelená).

Napětí na motorku se spíná pomocí tranzistoru. Na bázi tohoto tranzistoru jsou z výstupu mikropočítače vysílány impulsy PWM modulace, čímž je možno řídit rychlost otáčení.

Měření otáček elektromotorku zajišťuje optický snímač, který je složen z diody LED 1 (červená) a foto-tranzistoru T1. Hřídel stejnosměrného elektromotorku je opatřena černým kotoučem, ve kterém jsou navrtány otvory tak, aby úhlová vzdálenost mezi nimi nebyla stejná. Dioda LED svítí přes otvor v kotouči na foto-tranzistor. Ten pak na základě osvětlení své báze spojuje kladné napětí na kolektoru s GND na svém emitoru a tím vytváří pulzní výstup na svém kolektoru.

Tento výstup má nedokonalý obdélníkový charakter, který vytvaruje Schmittův klopný obvod na obdélníkový impuls o správných napěťových úrovních pro logické hodnoty 1 a 0.



Propojení modulu s mikropočítačem - ovládané funkce

1. Relé přepínání směru otáček: PTD6 (na konektoru pin 43)
2. Otáčky motoru (PWM): PTD4 = TPM0, CH4 (na konektoru pin 26)
3. Snímač otáček (impulsy od optického snímače): PTD2 = TPM0, CH2 (na konektoru pin 23)
 - o LED zelená indikuje směr. Svítí při nastaveném směru vpravo. LED není ovládána z mikropočítače.

Modul se připojuje na konektor MCU_PORT. Odpovídající kontakty na spodní straně desky modulu se zasunou do odpovídajících zdírek tohoto konektoru. Typický modul využívá pouze tři až čtyři vývody. Schéma celého konektoru je na následujícím obrázku.

	MCU port				
SW3	PTA16	1	2	PTA17	SW4
SW1	PTA4	3	4	PTA5	SW2
	PTE30	5	6	PTE20	
	PTE21	7	8	PTB0	
+3V	3V	9	10	GND	GND
	PTE5	11	12	PTC1	Sdíleno s analog. port
	PTE4	13	14	PTE3	
temp. senzor I2C_SDA	PTE0 (I2C1_SDA)	15	16	(I2C1_SCL) PTE1	temp. senzor I2C_SCL
	PTE2	17	18	PTB11	
+3V	3V	19	20	GND	GND
Sdíleno s analog. port	PTB1	21	22	(FTM0_CH0) PTD0	
	PTD2 (FTM0_CH2)	23	24	(FTM0_CH3) PTD3	
	PTD5	25	26	(FTM0_CH4) PTD4	
		27	28	PTC16	
+3V	3V	29	30	GND	GND
		31	32	PTC12	LCD_E
EEPROM_MOSI	PTC6 (SPI0_MOSI)	33	34	(SPI0_SCK) PTC5	EEPROM_SCK
		35	36	(SPI0_MISO) PTC7	EEPROM_MISO
		37	38		
+3V	3V	39	40	GND	GND
	PTC17	41	42	PTE31	
	PTD6	43	44	PTD7	
		45	46		
		47	48		
	3V	49	50	GND	GND
LCD_DB6	PTC10	51	52	PTC11	LCD_DB7
LCD_DB4	PTC8	53	54	PTC9	LCD_DB5
		55	56	PTA12	LCD_RW
		57	58		
RESET	RESET/PTA20	59	60		

Postup připojení modulu ke kitu

- Na napájecí svorky kitu připojíme napájecí konektor modulů
- Modul zasuneme do konektoru MCU_PORT na kitu. Pozor na správné zarovnání s okrajem konektoru tak, aby odpovídající kontakty modulu byly zasunuty ve správných zdírkách konektoru! Poslední řada kontaktů v horní části modulu motorku musí být zasunuta v posledních zdírkách konektoru na kitu (zdičky 1 a 2) viz obr. výše.
- Zapojíme napájecí konektor do modulu

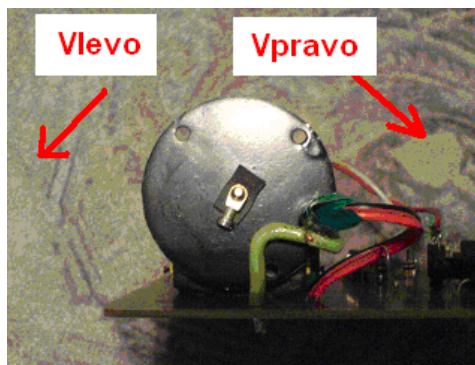
Programová obsluha – ovladač

Následující informace se týkají ovladače pro starší vývojový kit s mikropočítačem HCS08GB60. Pro nový vývojový kit s mikropočítačem Kinetis KL25Z není ovladač k dispozici. Původní ovladač ovšem může posloužit jako pomůcka.

K ovládaní modulu je naprogramována knihovna podprogramů (ovladač), která uživateli umožňuje měnit směr a rychlost otáčení stejnosměrného elektromotorku. Dále pak zjišťuje aktuální rychlost a směr otáček.

Pro práci s modulem je využit modul časovače TPM2, který má celkem 5 kanálů. Knihovna využívá dva kanály, jeden v režimu PWM (kanál 0, tj. TPM2 CH0) a druhý v režimu input capture (kanál 1, tj. TPM2 CH0).

Označení směru otáčení elektromotorku je určeno podle následujícího obrázku, tedy pohledem na motorek zepředu.



Ovladač v jazyku C

Ovladač se skládá ze souborů **motor_gb60.c** a **motor_gb60.h**.

Ovladač nevyžaduje, aby v projektu byla povolena podpora „floating point“.

Přehled funkcí ovladače C

void Minit(void)

Inicializace ovladače. Nutno volat před prvním použitím funkcí ovladače (na začátku vašeho programu). Funkce nastavuje časovač pro PWM a porty.

void SetPwm(char pwm)

Funkce slouží k nastavení rychlosti otáček a má jeden vstupní parametr – pwm. Parametr se zadává v hodnotách od 0 do 100 (tj. v procentech) a udává požadovaný příkon do motorku, resp. šířku aktivního pulzu v PWM modulaci.

int GetOtacky(void)

Funkce se používá k získání aktuálních otáček elektromotorku a nemá žádné vstupní parametry. Funkce vrací aktuální otáčky elektromotorku za minutu.

void SetSmer(char smer)

Funkce slouží k nastavení směru otáčení elektromotorku a má jeden vstupní parametr - smer. K nastavení směru otáčení elektromotorku doleva se předává hodnota 0 a k nastaven směru doprava se předává hodnota 1.

char GetSmer(void)

Funkce se používá k získání aktuálního směru otáček elektromotorku. Vrací aktuální směr otáčení motorku. Hodnota 0 znamená, že se motorek se otáčí doleva, 1 = doprava. Směr je zjištěn z optického snímače.

Ukázkový program

Uveden je pouze ukázkový program pro jazyk C. Ukázkový program pro assembler je možno najít v archivu ZIP obsahujícím soubory ovladače.

Ukázkový program periodicky mění směr a rychlost otáčení motorku a zobrazuje aktuální otáčky na displeji.

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

#include "main_asm.h" /* interface to the assembly module */
#include "stdio.h"      // sprintf
#include "disp_gb60.h"  // displej
#include "motor_gb60.h" // motor

void MCU_init(void); /* Device initialization function declaration */
void cekej(void);

char smer;
int otac;
char text[10];
void main(void) {
```

```

/* Uncomment this function call after using Device Initialization
   to use the generated code */
/* MCU_init(); */

EnableInterrupts; /* enable interrupts */

/* include your code here */
asm_main(); /* call the assembly function */

Minit();           // inicializace modulu Motorek
dinit();
dcls();
smer = 0;
while(1)
{
    SetPwm(0);      // zastav motorek
    cekej();
    dcls();
    SetSmer(smer);  // nastav smer
    cekej();
    SetPwm(90);     // nastav otacky
    cekej();

    otac = GetOtacky(); // zjistí otacky
    sprintf(text, "%d", otac);
    dcls();
    dtext(text);

    cekej();
    otac = GetOtacky();
    sprintf(text, "%d", otac);
    dcls();
    dtext(text);

    cekej();
    cekej();
    otac = GetOtacky();
    sprintf(text, "%d", otac);
    dcls();
    dtext(text);

    smer = GetSmer(); // zjistí smer
    smer = !smer;
    cekej();

    __RESET_WATCHDOG();
}

//for(;;) {
//    __RESET_WATCHDOG(); /* feeds the dog */
//} /* loop forever */
/* please make sure that you never leave this function */
}

void cekej(void)
{
    ulong i;
    for (i=0; i<100000; i++ )
        __RESET_WATCHDOG();
}

```

Literatura

Korčák Petr: Výukový modul pro předmět mikropočítače: řízení a měření otáček stejnosměrného motoru, Bakalářská práce, UTB Zlín, 2007.

Příloha: Schéma zapojení modulu

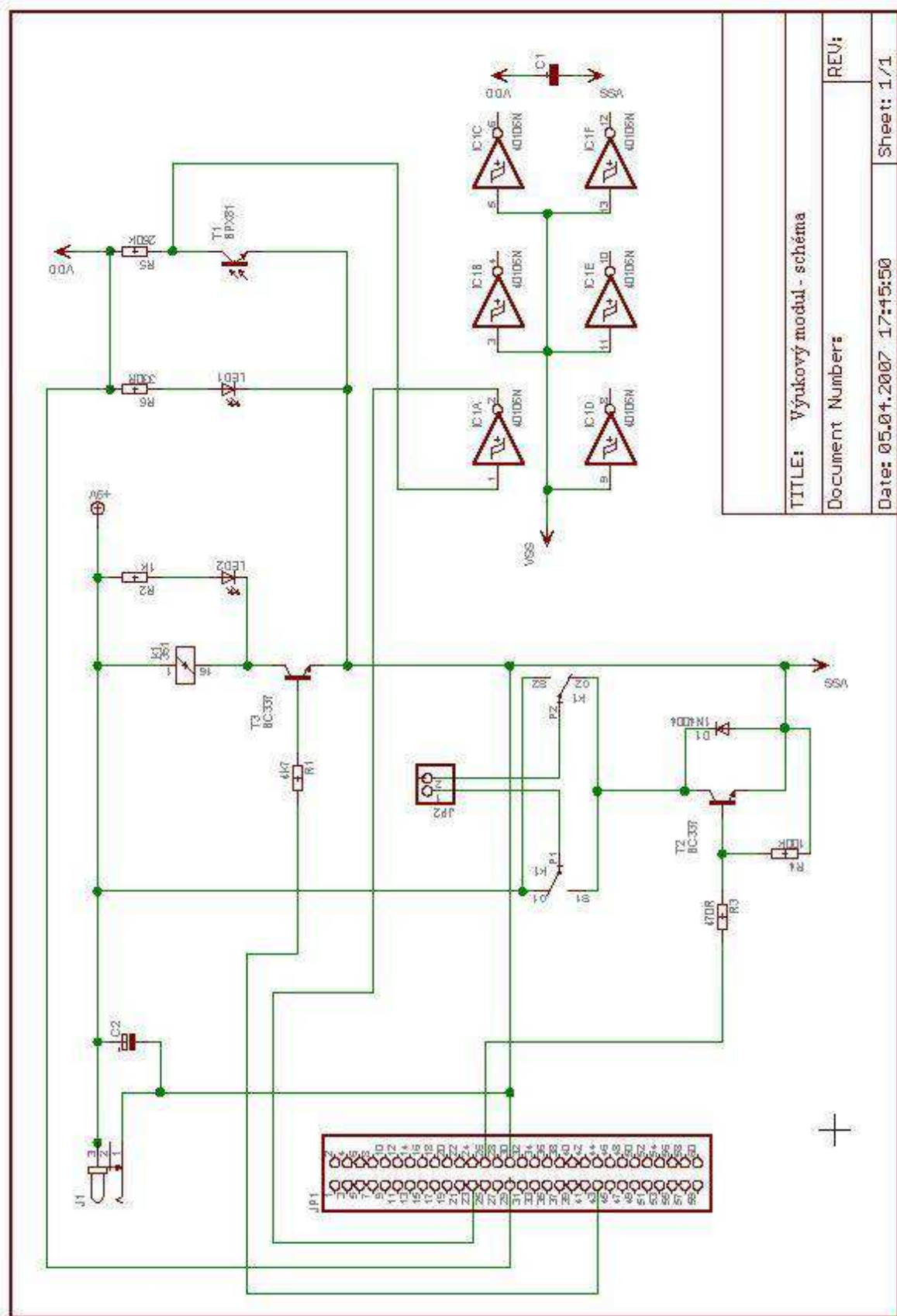


Schéma zapojení modulu.