

Vážení zákazníci,

dovolujeme si Vás upozornit, že na tuto ukázkou knihy se vztahují autorská práva, tzv. copyright.

To znamená, že ukáзка má sloužit výhradně pro osobní potřebu potenciálního kupujícího (aby čtenář viděl, jakým způsobem je titul zpracován a mohl se také podle tohoto, jako jednoho z parametrů, rozhodnout, zda titul koupí či ne).

Z toho vyplývá, že není dovoleno tuto ukázkou jakýmkoliv způsobem dále šířit, veřejně či neveřejně např. umístováním na datová média, na jiné internetové stránky (ani prostřednictvím odkazů) apod.

redakce nakladatelství BEN – technická literatura
redakce@ben.cz



18

JEDNOTKA ECCP U PIC18F1220

Mikrokontrolér PIC18F1220 disponuje jednotkou ECCP (Enhanced Capture/Compare/PWM). V této kapitole se seznámíme s jejími možnostmi a použitím.

18.1 ZÁKLADNÍ VLASTNOSTI

ECCP jednotka je implementována jako klasická jednotka CCP (podobně jako v kapitole 17) s rozšířenými schopnostmi v režimu PWM. Jedná se o podporu 2 nebo 4 kanálů, uživatelsky nastavitelné polarity, řízení mrtvého pásma a automatické vypnutí a restartu.

7	6	5	4	3	2	1	0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

P1M1:P1M0 – PWM Output Configuration bits (konfigurační bity PWM výstupu)

pro CCP1M3:CCP1M0 = 00xx, 01xx, 10xx

xx – P1A je použit jako CCP1 vývod; P1B, P1C, P1D jsou obyčejné vývody portu

pro CCP1M3:CCP1M0 = 11xx

00 – jednoduchý výstup; P1A je modulován; P1B, P1C, P1D jsou obyčejné vývody portu

01 – plný most; P1D je modulován; P1A aktivní; P1B, P1C, neaktivní

10 – poloviční most; P1A, P1B jsou modulovány včetně řízení mrtvého pásma; P1C, P1D jsou obyčejné vývody portu

11 – plný most, reverzní výstup; P1B je modulován; P1C aktivní; P1A, P1D, neaktivní

DC1B1:DC1B0 – PWM Duty Cycle bit1 and bit0 (spodní bity střídý pro PWM)

tyto bity se používají pouze v případě, že jednotka je konfigurována do režimu PWM

jedná se o dva nejnižší bity (bit1 a bit0) 10bitové PWM střídý, horních 8 bitů je uloženo v registru **CCPRxL**

CCP1M3:CCP1M0 – ECCP1 Mode Select bits (výběr režimu jednotky ECCP1)

0000 – ECCP jednotka odstavena, reset jednotky ECCP

0001 – vyhrazeno

0010 – režim Compare, negace výstupu při shodě (toggle funkce, bit **ECCP1IF** se nastaví)

0011 – vyhrazeno

0100 – režim Capture, aktivace sestupnou hranou

0101 – režim Capture, aktivace náběžnou hranou

0110 – režim Capture, aktivace každou čtvrtou náběžnou hranou (dělička 1:4)

0111 – režim Capture, aktivace každou šestnáctou náběžnou hranou (dělička 1:16)

1000 – režim Compare, výchozí stav vývodu CCP1 je log. 0, po shodě log. 1 (bit **ECCP1IF** se nastaví)

1001 – režim Compare, výchozí stav vývodu CCP1 je log. 1, po shodě log. 0 (bit **ECCP1IF** se nastaví)

1010 – režim Compare, při shodě generuje přerušování (bit **ECCP1IF** se nastaví, stav vývodu ECCP1 se nemění)

1011 – režim Compare, speciální událostní spouštěč (bit **ECCP1IF** se nastaví, ECCP nuluje TMR1 nebo TMR3 a spouští A/D převod, pokud je A/D povolen)

1100 – režim PWM, P1A a P1C aktivní v log. 1, P1B a P1D aktivní v log. 1

1101 – režim PWM, P1A a P1C aktivní v log. 1, P1B a P1D aktivní v log. 0

1110 – režim PWM, P1A a P1C aktivní v log. 0, P1B a P1D aktivní v log. 1

1111 – režim PWM, P1A a P1C aktivní v log. 0, P1B a P1D aktivní v log. 0

Obr. 18.1 Registr **CCP1CON**

Jednotka obsahuje 16bitový registr, který může pracovat buď jako 16bitový záchytný registr (capture), 16bitový porovnávací registr (compare) nebo PWM registr. Jedná se o registrový pár **CCPR1H:CCPR1L**.

Obr. 18.1 uvádí řídicí registr **CCP1CON**. Pro rozšířené PWM operace a automatické vypnutí se používají registry **PWM1CON** a **ECCPAS**.

ECCP výstupy

Jednotka ECCP může mít (v závislosti na zvoleném režimu) až čtyři výstupy. Tyto výstupy označené jako **P1A** až **P1D** jsou multiplexovány s vývody portu B. Přiřazení vývodů je zřejmé z tab. 18.1. Šedě vybarvené buňky indikují přiřazení vývodů, které není použito pro jednotku ECCP.

Pro konfiguraci vývodů portu B jako PWM výstupů je třeba správně nastavit bity **CCP1M3:CCP1M0** z registru **CCP1CON**. Rovněž je třeba konfigurovat odpovídající vývody jako výstupní pomocí registru **TRISB**.

Tab. 18.1 Přiřazení vývodů v různých režimech jednotky ECCP

ECCP režim	CCP1CON konfigurace	RB3	RB2	RB6	RB7
kompatibilní CCP	00xx 11xx	CCP1	RB2/INT2	RB6/PGC/ T1OSO/ T13CKI/ KBI2	RB7/PGD/ T1OSI/KBI3
dva PWM	10xx 11xx	P1A	P1B	RB6/PGC/ T1OSO/ T13CKI/ KBI2	RB7/PGD/ T1OSI/KBI3
čtyři PWM	x1xx 11xx	P1A	P1B	P1C	P1D

Tab. 18.2 Zdroje hodin pro CCP režim

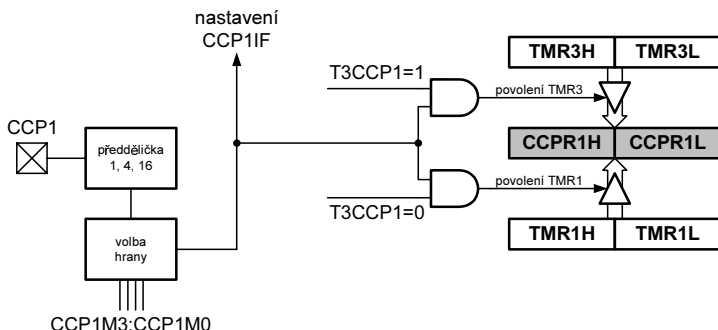
ECCP režim	Zdroj hodin
Capture	Timer1 nebo Timer3
Compare	Timer1 nebo Timer3
PWM	Timer2

18.2 REŽIM CAPTURE

Režim Capture (též Input Capture, česky často překládáno jako záchytný registr) zachytí aktuální obsah registru **TMR1/TMR3** ve chvíli, kdy nastane tzv. vstupní událost. Jako vstupní událost lze konfigurovat (pomocí bitů **CCP1M3:CCP1M0** z registru **CCP1CON**):

- sestupná hrana na vstupu **CCP1**,
- náběžná hrana na vstupu **CCP1**,
- každá čtvrtá náběžná hrana na vstupu **CCP1**,
- každá šestnáctá náběžná hrana na vstupu **CCP1**.

Dojde-li ke spouštěcí události, je jednak nastaven příznak přerušení **CCP1IF** z registru **PIR1** a dále dojde (dle aktivity řídicího bitu **T3CCP1**) k uložení aktuálního stavu registrového páru **TMR1H:TMR1L/TMR3H:TMR3L** do páru **CCPR1H:CCPR1L**.



Obr. 18.2 Blokové schéma pro režim **Capture**

Pokud si obsah časovače Timer1/Timer3 představíme jako časovou osu, je tedy uložen okamžik aktivace vstupní události.

Pokud nastane nová vstupní událost před čtením obsahu **CCPR1H:CCPR1L**, dojde k přepisu nově zachycenou hodnotou.

Konfigurace vývodu **CCP1**

Vývod **CCP1** by měl být příslušným registrem **TRISB** nastaven jako vstupní. Pokud vývod konfigurujeme jako výstupní, může zápis na port způsobit vstupní událost.

Volba **Timer1** nebo **Timer3** jako zdroje hodin

Jako zdroj hodin pro jednotku Capture lze použít Timer1 nebo Timer3 pracující v režimu časovače nebo jako synchronizovaný čítač (asynchronní čítač nelze použít). Výběr mezi Timer1 a Timer3 provádíme pomocí registru **T3CON** bitem **T3CCP1**.

Přerušení

V případě, že změním režim jednotky CCP1, může dojít k falešné generaci přerušení. Proto je dobré před změnou režimu vynulovat bit **CCP1IE** z registru **PIE1** a následně vynulovat bit **CCP1IF** z registru **PIR1**.

Předdělička **CCP**

Předdělička je nulována vždy, když je jednotka CCP vypnuta nebo když se nenačítá v režimu Capture. K vynulování dojde i po libovolném resetu mikrokontroléru.

Při přepínání mezi jednotlivými režimy Capture nedochází k automatickému nulování předděličky. Proto je třeba přepnutí přes režim **CCP1M3:CCP1M0 = 0000**.

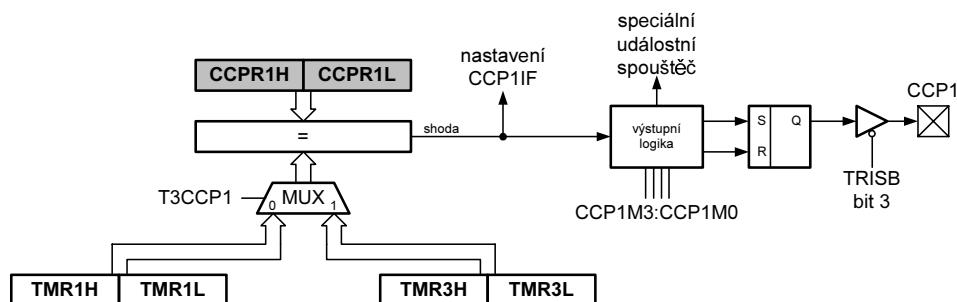
18.3 REŽIM COMPARE

V režimu Compare (též Output Compare) je obsah registru **CCPR1H:CCPR1L** neustále porovnáván s obsahem registru **TMR1/TMR3**. Při shodě se nastaví příznak **CCP1IF** a vývod **CCP1** může být:

- uveden do log. 1,
- uveden do log. 0,
- znegovat svůj stav (tzv. toggle),
- zůstat beze změny.

Akce na vývodu **CCP1** je závislá na nastavení bitů **CCP1M3:CCP1M0**. Viz obr. 18.1.

Princip činnosti v režimu Compare je zřejmý z obr. 18.3.



Obr. 18.3 Blokové schéma pro režim Compare

Dle stavu bitu **T3CCP1** se testuje shoda obsahu registrového páru **CCPR1H:CCPR1L** s obsahem páru **TMR1H:TMR1L/TMR3H:TMR3L**.

Je-li dosaženo shody, je jednak nastaven příznak přerušení **CCP1IF** z registru **PIR1** a dále dojde k aktivaci bloku výstupní logiky.

Dle zvoleného režimu pak výstupní logika nulování, nastavení nebo negaci pomocí klopného obvodu RS. Signál z klopného obvodu pak pokračuje výstupním budičem (cesta musí být aktivována vynulováním příslušného bitu z registru **TRISB**) na vývod **CCP1**.

Tab. 18.3 Souhrn registrů spojených s režimy Capture, Compare a Timer1, Timer3

Registr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hodnota po POR, BOR	Hodnota po jiných resetech
TMR1L	spodních 8 bitů Timer1								xxxx xxxx	uuuu uuuu
TMR1H	horních 8bitů Timer1								xxxx xxxx	uuuu uuuu
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	-000 -000	-000 -000
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 -000	-000 -000
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	-111 -111	-111 -111
T1CON	RD16	T1RUN	T1CKPS1:0	T1OSCN	T1SYNC	TMR1CS	TMR1ON		0000 0000	uuuu uuuu
TMR3L	spodních 8 bitů Timer3								xxxx xxxx	uuuu uuuu
TMR3H	horních 8bitů Timer3								xxxx xxxx	uuuu uuuu
T3CON	RD16	—	T3CKPS1:0	T3CCP1	T3SYNC	TMR3CS	TMR3ON		0-00 0000	u-uu uuuu
TRISB	registr směru dat portu B								1111 1111	1111 1111
CCPR1L	registr CCP1 (dolní bajt)								xxxx xxxx	uuuu uuuu
CCPR1H	registr CCP1 (horní bajt)								xxxx xxxx	uuuu uuuu
CCP1CON	P1M1	P1M0	DC1B1:DC1B0		CCP1M3:CCP1M0				0000 0000	0000 0000
ADCON0	VCFG1:VCFG0	—		CHS2:CHS0	GO	ADON			00-0 0000	00-0 0000

Legenda:

- x = neznámá hodnota,
- u = hodnota se nezmění,
- šedě vybarvené buňky nejsou v režimech Capture nebo jednotkou Timer1 využívány.

Konfigurace vývodu CCP1

Vývod **CCP1** musí být příslušným registrem **TRISB** nastaven jako výstupní. Vynulování registru **CCP1CON** nastaví vývod do log. 0.

Volba Timer1 nebo Timer3 jako zdroje hodin

Jednotky Timer1 nebo Timer3 musí pracovat jako časovače nebo synchronizované čítače. Při použití čítače v asynchronním režimu nebude režim Compare pracovat.

Speciální událostní spouštěč

Výstup speciálního událostního spouštěče jednotky **CCP1** nuluje obsah **TMR1/TMR3** registru. Tím **CCPR1H:CCPR1L** registr efektivně zajišťuje 16bitovou programovatelnou periodu pro jednotku Timer1/Timer3.

Jednotku **CCP1** lze použít rovněž pro spuštění A/D převodu.

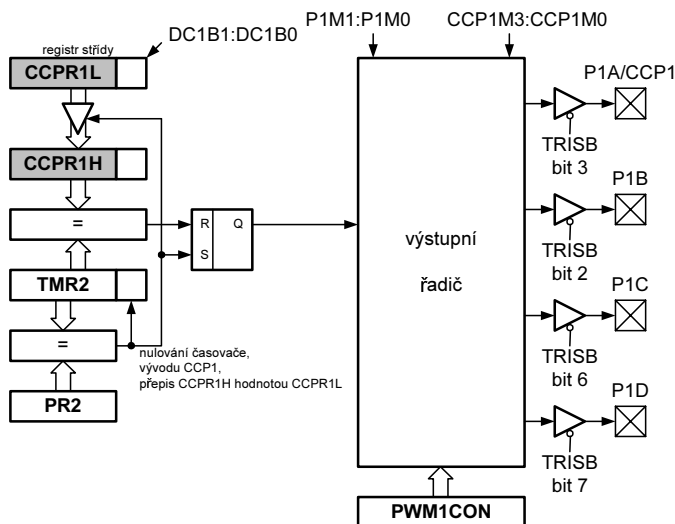
Vynulování Timer1/Timer3 nevede nikdy k nastavení příznaku přetečení těchto jednotek.

18.4 ROZŠÍŘENÝ REŽIM PWM

Rozšířený PWM režim poskytuje širší možnosti generování PWM signálu než klasická jednotka CCP. Je možné zvolit polaritu signálu (zda je aktivní v log. 1 nebo v log. 0) pomocí bitů **P1M1:P1M0** a **CCP1M3:CCP1M0** z registru **CCP1CON**.

Obr. 18.4 ukazuje zjednodušené blokové schéma PWM jednotky. Všechny řídicí registry jsou dvojitě bufferované a k aktualizaci dochází vždy na začátku PWM cyklu aby se zabránilo zákmitům výstupů. Výjimkou je registr zpoždění **PWM1CON**, který se nahrává buď na konci aktivní fáze nebo na konci periody. Bufferování vede k tomu, že průběhy PWM výstupů nejsou stejné jako u běžné PWM jednotku (například u PIC18F452), ale vzniká zpoždění v délce jednoho instrukčního cyklu.

Pro povolení funkce výstupů je třeba správně konfigurovat registr **TRISB**.



Obr. 18.4 Blokové schéma pro režim PWM

Perioda PWM

Perioda PWM signálu je určena hodnotou uloženou v registru **PR2**. Periodu PWM lze vypočítat takto: $T_{\text{PWM}} = (\text{PR2} + 1) \cdot 4 \cdot T_{\text{OSC}}$ (předdělička TMR2), kmitočet je tedy:

$$f_{\text{PWM}} = \frac{1}{T_{\text{PWM}}}$$

V okamžiku hodinového cyklu následujícího po shodě TMR2 = PR2 budou provedeny tyto akce:

- **TMR2** bude vynulován,
- vývod **CCP1** bude uveden do log. 1 (vyjma případu, že střída je 0; tedy je **CCP1** stále v log. 0),
- hodnota střídý je přepsána z **CCPR1L** do **CCPR1H**.

Střída PWM

Střída je určena 10bitovou hodnotou **CCPR1L:DC1B1:DC1B0**. Registr **CCPR1L** obsahuje nejvyšších 8 bitů, **DC1B1:DC1B0** obsahují nejnižší 2 bity.

Pro výpočet doby trvání aktivní části periody (střídý v časovém měřítku) lze použít vzorec: doba aktivní části (CCPR1L:DC1B1:DC1B0) $\cdot 4 \cdot T_{\text{OSC}}$ (předdělička TMR2).

Pokud bude aktivní část delší než perioda PWM, nedojde k vynulování vývodu **CCP1**. Zůstane stále v log. 1.

Hodnotu do **CCPR1L:DC1B1:DC1B0** lze zapsat v libovolném časovém okamžiku, k přepisu do **CCPR1H** dojde vždy na konci periody. V PWM režimu lze registr **CCPR1H** pouze číst. Této technice se říká **dvojitě bufferování**, je nezbytná pro zabránění vzniku zákmitů při nesynchronizované změně parametrů PWM.

Maximální rozlišení PWM v bitech pro daný kmitočet PWM lze stanovit pomocí vztahu:

$$\text{rozlišení PWM} = \frac{\log\left(\frac{f_{\text{OSC}}}{f_{\text{PWM}}}\right)}{\log(2)} \text{ bitů.}$$

Tab. 18.4 Příklad kmitočtů PWM a rozlišení při hodinovém kmitočtu 40 MHz

f_{PWM} [kHz]	2,44	9,77	39,06	156,25	312,50	416,67
Předdělička časovače	16	4	1	1	1	1
PR2 hodnota	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximální rozlišení (bity)	14	12	10	8	7	6,58

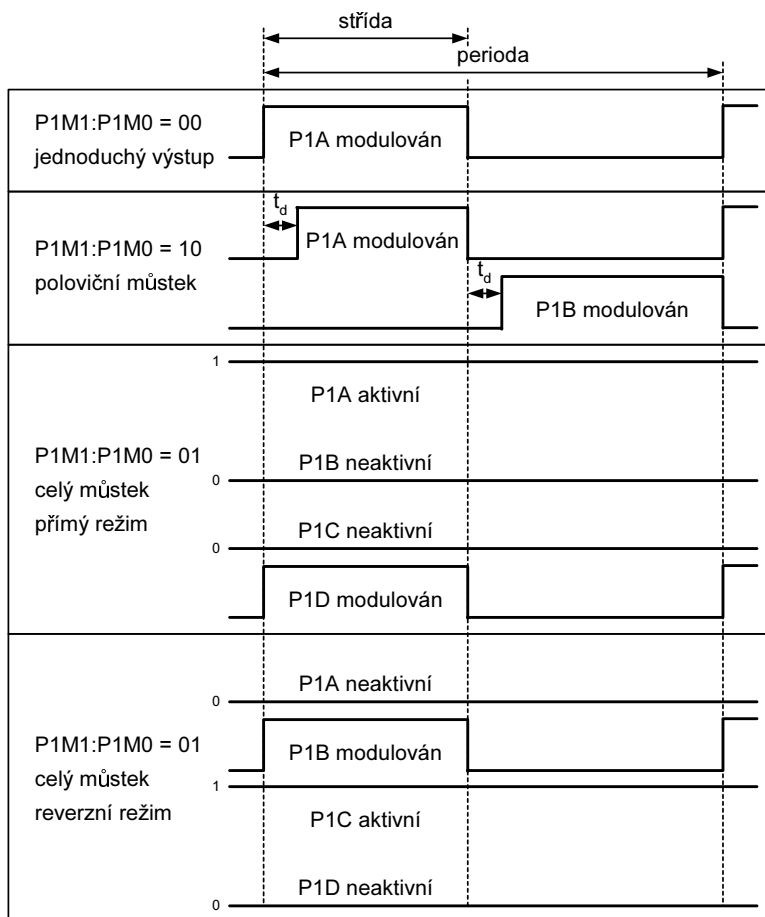
Konfigurace PWM výstupu

Bity **P1M1:P1M0** registru **CCP1CON** umožňují konfigurovat výstup rozšířené jednotky PWM do těchto čtyř variant (viz obr. 18.5):

- jednoduchý výstup (kompatibilní se standardním PWM výstupem),
- výstup typu poloviční můstek,

- výstup typu plný můstek, přímý režim,
- výstup typu plný můstek, reverzní režim.

Obr. 18.5 platí pro režim aktivní v log. 1 (**CCP1M3:CCP1M0 = 1100**). Interval t_d zařazený v režimu polovičního můstku odpovídá intervalu mrtvého pásma nastaveného pomocí registru **PWM1CON**.



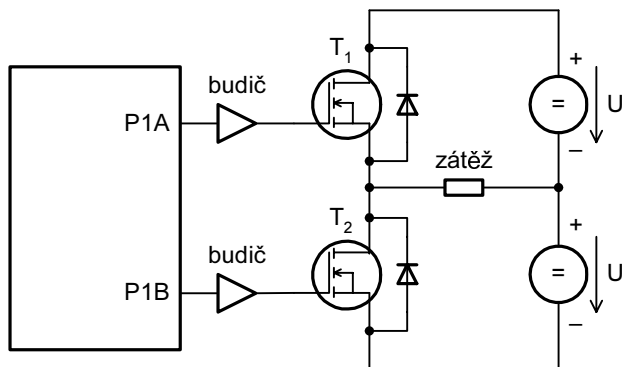
Obr. 18.5 Časové průběhy PWM výstupů

Poloviční můstek (H můstek)

V režimu polovičního můstku se používají dva výstupní signály: **P1A** (klasický PWM výstup) a **P1B** (komplementární výstup).

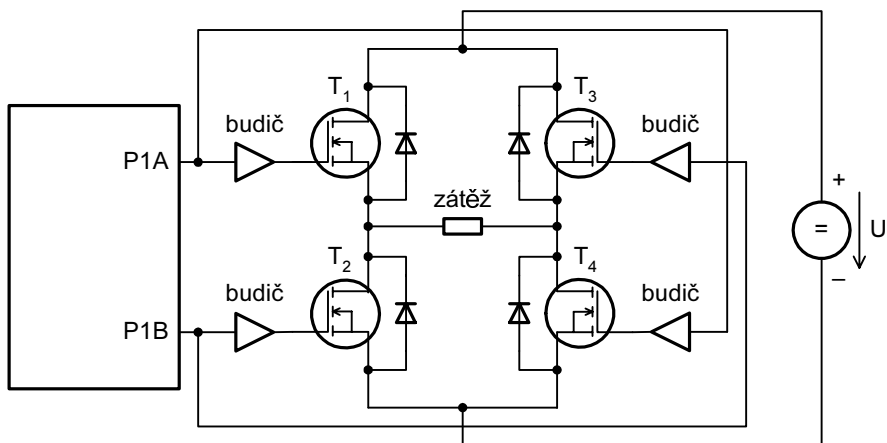
Zátěž je možno budít pomocí dvou spínacích tranzistorů v podobě polovičního můstku (viz obr. 18.6) nebo pomocí čtyř spínacích tranzistorů v podobě plného můstku (viz obr. 18.7).

Obvod dle obr. 18.6 nechá protékat proud zátěži ve směru zleva doprava při sepnutí tranzistoru T_1 (aktivaci výstupu P1A). Naopak při sepnutí tranzistoru T_2 (aktivaci výstupu P1B) bude proud zátěži protékat ve směru zprava doleva. Budiče tranzistorů zajistí plný rozkmit signálu dle napájecího zdroje zátěže.



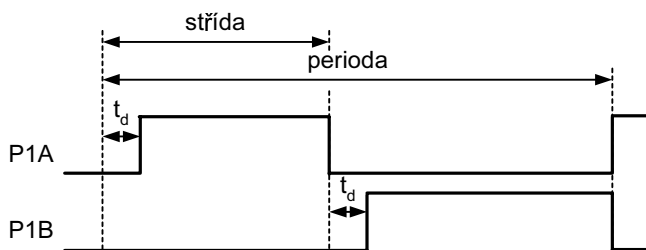
Obr. 18.6 Příklad zapojení standardního polovičního můstku (nutné dva zdroje)

Obvod dle obr. 18.7 vystačí s jedním napájecím zdrojem. Proud protéká zátěží ve směru zleva doprava při současném sepnutí tranzistorů T_1 a T_4 (aktivaci výstupu P1A). Naopak při současném sepnutí tranzistorů T_2 a T_3 (aktivaci výstupu P1B) bude proud zátěží protékat ve směru zprava doleva. Budiče tranzistorů zajistí plný rozkmit signálu dle napájecího zdroje zátěže.



Obr. 18.7 Příklad zapojení celého můstku řízeného signály polovičního můstku (jeden zdroj)

V režimu polovičního můstku je možné nastavit mrtvý interval t_d , aby se vyloučilo současné sepnutí tranzistorů v totemu (T_1 a T_2 případně i T_3 a T_4 a tedy i zkrat napájecího zdroje). Tento interval se nastavuje pomocí registru **PWM1CON**. Tak je možno stanovit, kolik instrukčních cyklů má uplynout mezi vypnutím jednoho tranzistoru a zapnutím druhého tranzistoru v totemu. Situace je zřejmá z obr. 18.8.



Obr. 18.8 Výstup PWM v režimu polovičního můstku (signály aktivní v log. 1)

Plný můstek

V režimu plného můstku se používají čtyři výstupní signály, aktuálně jsou však vždy aktivní pouze dva z nich.

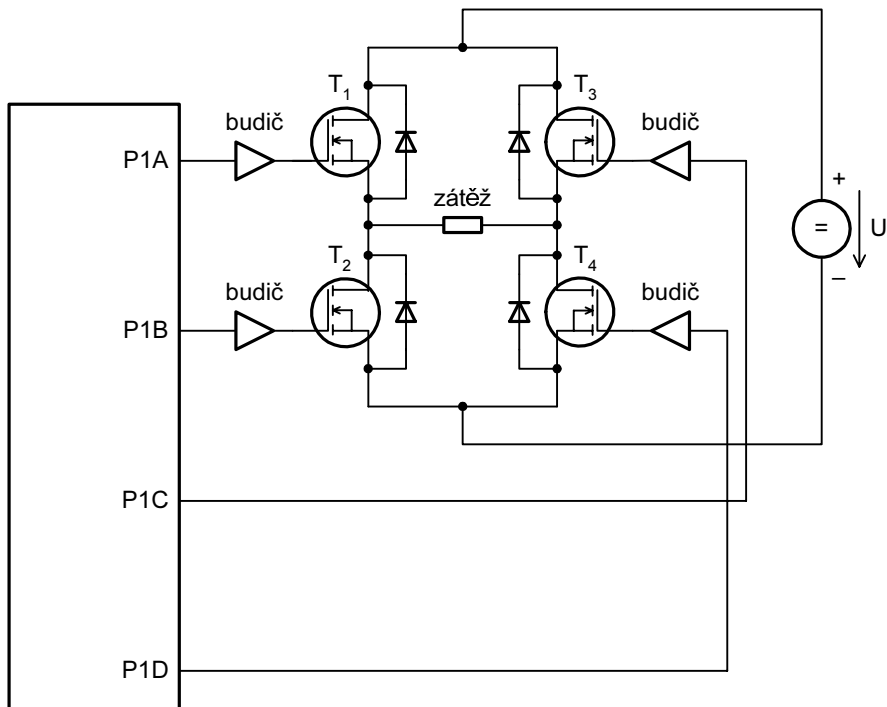
V přímém režimu je **P1A** stále aktivní a **P1D** je modulován.

V reverzním režimu je **P1C** stále aktivní a **P1B** je modulován.

Situace je zřejmá z obr. 18.5.

V zapojení dle obr. 18.9 protéká proud zátěží ve směru zleva doprava při sepnutí tranzistorů T_1 a T_4 (přímý směr). Proud zprava doleva protéká při sepnutí tranzistorů T_3 a T_2 (reverzní směr).

Řízení směru mezi přímým a reverzním zajišťuje v režimu plného můstku bit **P1M1** z registru **CCP1CON**. Po změně tohoto bitu dojde k přepnutí směru v následujícím



Obr. 18.9 Příklad zapojení plného můstku

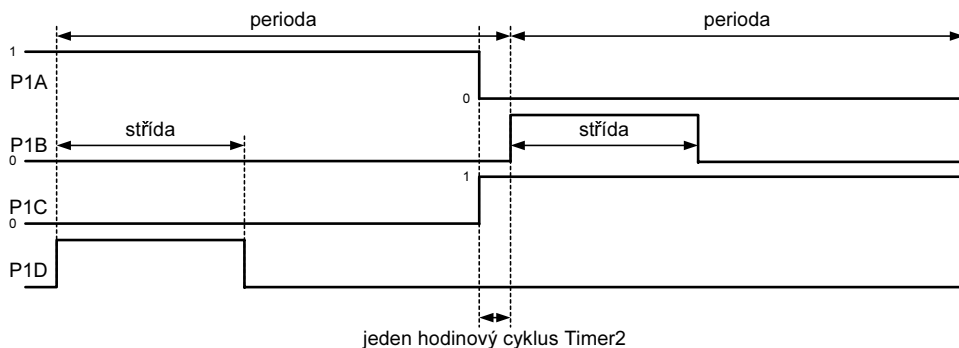
PWM cyklu. Průběh přepnutí ukazuje *obr. 18.10*. K přepnutí směru dojde jeden hodinový cyklus periody Timer2 před koncem periody, signál střídá bude spuštěn až se začátkem nové periody.

Režim plného můstku neposkytuje mrtvý interval, v řadě případů to není třeba. Situace, kdy by byl mrtvý interval vhodný jsou:

- při změně směru PWM výstupu v případě, že střída je blízko 100 % (dle *obr. 18.9* a *obr. 18.10* dojdeme ke zjištění, že se mohou současně sepnout například tranzistory T_3 a T_4 a totemem vpravo poteče příčný proud),
- vypínací čas tranzistorů (včetně budiče) je větší než zapínací čas (opět může dojít k nežádoucímu souběhu).

Uvedené problémy lze řešit buď pomocnými obvody nebo správným návrhem programu resp. zapojení:

- před přepnutím směru je dobré snížit hodnotu střídá,
- použít tranzistory a budiče, které zajistí krátké doby vypnutí (kratší než doby zapnutí).



Obr. 18.10 Přepnutí směru (režim aktivní v log. 1)

Programovatelný mrtvý interval

Poloviční můstek přináší potenciální nebezpečí souběžného sepnutí horního i dolního tranzistoru v totemu (viz *obr. 18.6* a *obr. 18.7*). Tento jev lze eliminovat vzájemným zpožděním aktivace pomocí programovatelného mrtvého intervalu (viz *obr. 18.8*).

Pro nastavení mrtvého intervalu slouží spodních 7 bitů registru **PWM1CON** dle *obr. 18.11*. Interval je nastavitelný v rozsahu 0 až 127 strojních cyklů mikrokontroléru (připomeňme, že strojní cyklus trvá čtyřnásobek hodinového cyklu).

PWM auto-shutdown

Výstupní vývody ECCP lze v režimu PWM konfigurovat jako auto-shutdown. Označení Auto-shutdown znamená automatické odpojení vývodů. Po aktivaci vnější události přejdou výstupy do předdefinovaných úrovní.

Pro řízení tohoto režimu slouží registr **ECCPAS** dle *obr. 18.12*.

Pro volbu spouštěcího zdroje slouží bity **ECCPAS2:ECCPAS0**, jako zdroj lze zvolit vývody **INT0**, **INT2** nebo **INT1**.

Po aktivaci vybraného vstupu jsou dané výstupy okamžitě uvedeny do přednastavených hodnot dle stavu bitů **PSSAC1:PSSAC0** a **PSSBD1:PSSBD0**. Po aktivaci auto-shutdown režimu mohou výstupu přejít do log. 0, log. 1 nebo do třetího stavu.

7	6	5	4	3	2	1	0
PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0

PRSEN – PWM Restart Enable bit (povolení restartu PWM)

při režimu automatického vypnutí

PRSEN=1 bity ECCPASE se nulují automaticky jakmile nastane shutdown, PWM se restartuje automaticky

PRSEN=0 bity ECCPASE se musí nulovat programově

PDC6:PDC0 – PWM Delay Count bits (zpoždění v počtu cyklů)

počet strojních cyklů mezi plánovaným okamžikem aktivace a skutečným okamžikem aktivace

Obr. 18.11 Registr **PWM1CON**

7	6	5	4	3	2	1	0
ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0
R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0	R/W 0

ECCPASE – ECCP Auto-Shutdown Event Statuts bit (statový bit auto-shutdown)

1 – ECCP výstupy pracují

0 – ECCP výstupy jsou odstaveny

ECCPAS2 – ECCP Auto-Shutdown bit 2 (odstavovací bit 2)

1 – vývod INT0 nemá vliv

0 – log. 0 na INT0 způsobí odstavení

ECCPAS1 – ECCP Auto-Shutdown bit 1 (odstavovací bit 1)

1 – vývod INT2 nemá vliv

0 – log. 0 na INT2 způsobí odstavení

ECCPAS0 – ECCP Auto-Shutdown bit 0 (odstavovací bit 0)

1 – vývod INT1 nemá vliv

0 – log. 0 na INT1 způsobí odstavení

PSSACn – Pins A and C Shutdown State Control bits (stav P1A a P1C po odstavení)

00 – vývody P1A a P1C budou v log. 0

01 – vývody P1A a P1C budou v log. 1

1x – vývody P1A a P1C budou ve třetím stavu

PSSBDn – Pins A and C Shutdown State Control bits (stav P1B a P1D po odstavení)

00 – vývody P1B a P1D budou v log. 0

01 – vývody P1B a P1D budou v log. 1

1x – vývody P1B a P1D budou ve třetím stavu

Obr. 18.12 Registr **ECCPAS**

Po aktivaci režimu auto-shutdown je nastaven bit **ECCPASE** a tím indikuje aktivaci tohoto režimu. Pokud není povolen automatický restart (**PRSEN = 0** z registru **PWM1CON**), lze bit **ECCPASE** nulovat pouze programově. Pokud je povolen automatický restart (**PRSEN = 1** z registru **PWM1CON**), dojde k vynulování bitu **ECCPASE** automaticky po tom, co příčina aktivace zanikne.

Pokud je bit **ECCPASE** nastaven v okamžiku, když PWM perioda začíná, zůstanou výstupy na přednastavených hodnotách i v dalších periodách. Při vynulování bitu **ECCPASE** se výstupy vrátí do normální funkce na začátku nové periody.

Zápis do bitu **ECCPASE** je odstaven, pokud je podmínka aktivace aktivní.

Režim auto-shutdown lze konfigurovat s automatickým restartem. Restart provede další spouštěcí událost. Tuto možnost povolíme nastavením bitu **PRSEN** z registru **PWM1CON** (viz obr. 18.11).

Je-li **PRSEN = 1**, zůstane bit **ECCPASE** nastaven, dokud příčina aktivace trvá. Když příčina zanikne, je bit vynulován.

Je-li **PRSEN = 0**, způsobí první aktivace nastavení bitu **ECCPASE**. Bit lze nulovat pouze programově.

Vynulování bitu **ECCPASE** vede k novému rozběhu v sledující periodě PWM.

Režim auto-shutdown lze vyvolat programově nastavení bitu **ECCPASE**.

Nastavení PWM režimu

Jelikož mikrokontrolér ve stavu resetu uvádí vývody do třetího stavu, musí vnější obvod při použití PWM režimu zajistit připojení vnějších pull-up nebo pull-down rezistorů tak, aby nedošlo k nežádoucímu sepnutí spínacích tranzistorů. Rezistor pull-up vytáhne linku do log. 1, rezistor pull-down stáhne linku do log. 0.

Bity **CCP1M1:CCP1M0** z registru **CCP1CON** umožňují volit aktivní úroveň pro spínání vnějších tranzistorů. Tato polarita výstupu musí být nastavena před konfigurováním vývodů jako výstupů. Změna polarity v průběhu PWM cyklu může poškodit vnější obvod.

Pro úspěšnou konfiguraci modulu ECCP jako PWM je třeba provést tyto operace:

1. Konfigurace vývodů **P1A** až **P1D** jako vstupů nastavením odpovídajících bitů **TRISB**.
2. Nastavení periody PWM pomocí registru **PR2**.
3. Konfigurace režimu pomocí registru **CCP1CON** (výběr režimu výstupů pomocí bitů **P1M1:P1M0**, výběr polarity pomocí bitů **CCP1M3:CCP1M0**).
4. Nastavení střídý pomocí **CCP1RL:DC1B1:DC1B0**.
5. Pro poloviční můstek je třeba nastavit interval t_d pomocí registru **PWM1CON**.
6. Pro auto-shutdown operaci je třeba nastavit registr **ECCPAS** (výběr zdroje pro aktivaci pomocí bitů **ECCPAS2:ECCPAS0**, volba přednastavených hodnot výstupů pomocí bitů **PSSAC1:PSSAC0** a **PSSBD1:PSSBD0**, nastavení bitu **ECCPASE**).
7. Pro auto-restart operaci je třeba nastavit bit **PRSEN** z registru **PWM1CON**.
8. Konfigurace jednotky Timer2 pro spuštění PWM (vynulování bitu **TMR2IF** z registru **PIR1**, nastavení předděličky pomocí bitů **T2CKPS2:0** z registru **T2CON**, spouštění časovače pomocí bitu **TMR2ON** z registru **T2CON**).

9. Povolení PWM výstupů po startu nového PWM cyklu (počkáme na přetečení jednotky Timer2 – bit **TMR2IF** z registru **PIR1** bude nastaven, povolení příslušných výstupů pomocí registru **TRISB**, vynulování bitu **ECCPASE** z registru **ECCPAS**).

Práce v režimech snížené spotřeby

V režimu Sleep jsou všechny zdroje hodin odstaveny, takže ani Timer2 nemůže měnit svůj obsah a generovat PWM signál. Hodnoty vybavené na výstupy zůstanou zachovány.

V režimu **PRI_IDLE** je hlavní zdroj v činnosti a jednotka ECCP pokračuje v činnosti.

V ostatních režimech snížené spotřeby lze vybrat hodinový signál pro řízení Timer2.

Je-li povolena jednotka Fail-Safe Clock Monitor a dojde k selhání hodin, přepne se mikrokontrolér do režimu **RC_RUN** a nastaví se bit **OSCFIF** z registru **PIR2**. Jednotka ECCP je pak taktována ze zdroje hodin **INTRC** s velmi nízkým kmitočtem. Pomocí bitů **IRCF2:IRCF0** při resetu lze povolit oscilátor **INTOSC** s vyšším kmitočtem.

Tab. 18.5 Souhrn registrů spojených s režimem PWM a Timer2

Registr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Hodnota po POR, BOR	Hodnota po jiných resetech		
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u		
RCON	IPEN	—	—	/RI	/TI	/PD	/POR	/BOR	0- - 1 11qq	0- - q qquu		
PIR1	—	ADIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	-000 -000	-000 -000		
PIE1	—	ADIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 -000	-000 -000		
IPR1	—	ADIP	RCIP	TXIP	—	CCP1IP	TMR2IP	TMR1IP	-111 -111	-111 -111		
TMR2	obsah Timer2								0000 0000	0000 0000		
PR2	perioda Timer2								1111 1111	1111 1111		
T2CON	—	TOUTPS3:TOUTPS0				TM2RON	T2CKPS1:T2CKPS0			-000 0000	-000 0000	
TRISB	registr směru dat portu B								1111 1111	1111 1111		
CCPR1L	registr CCP1 (dolní bajt)								xxxx xxxx	uuuu uuuu		
CCPR1H	registr CCP1 (horní bajt)								xxxx xxxx	uuuu uuuu		
CCP1CON	P1M1:P1M0		DC1B1:DC1B0		CCP1M3:CCP1M0					0000 0000	0000 0000	
ECCPAS	ECCPASE		ECCPAS2:ECCPAS0		PSSAC1:PSSAC0		PSSBD1:PSSBD0			0000 0000	0000 0000	
PWM1CON	PRSEN		PDC6:PDC0								0000 0000	uuuu uuuu
OSCCON	IDLEN		IRCF2:IRCF0		OSTS	IOFS	SCS1	SCS0	0000 0000	uuuu uuuu		
ADCON0	VCFG1:VCFG0		—		CHS2:CHS0		GO	ADON	0000 qq00	0000 qq00		

Legenda:

- x = neznámá hodnota,
- u = hodnota se nezmění,
- šedě vybarvené buňky nejsou v režimech rozšířené jednotky PWM použity.

18.5 PŘÍPRAVEK MBRIDGE

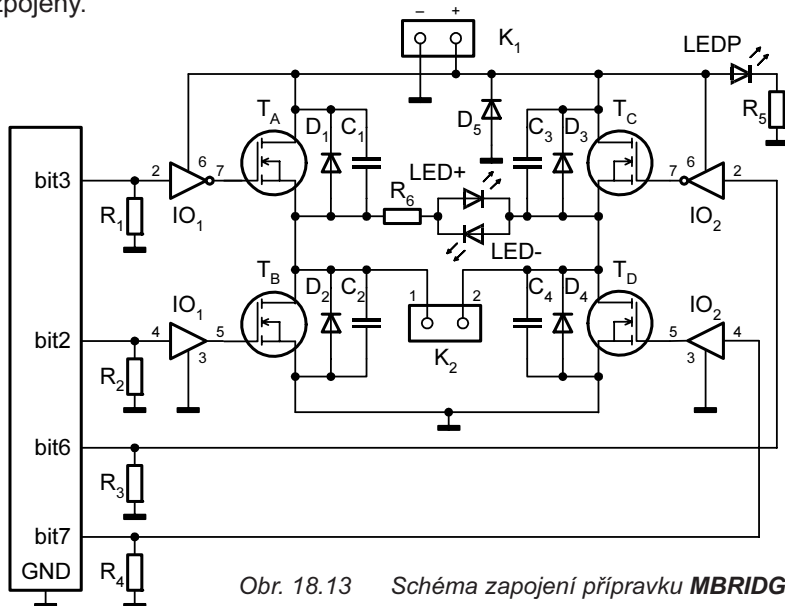
Přípravek **MBRIDGE** umožňuje provést test PWM režimu plný můstek. Schéma zapojení je uvedeno na obr. 18.13.

Jádrem přípravku jsou budiče **IO₁**, **IO₂** typu **TC428**. Jedná se o budiče vyráběné firmou Microchip s pracovním napětím 4,5 až 18 V. Pomocí těchto budičů lze snadno řídit unipolární tranzistory označené **T_A** až **T_D**.

Vzhledem k tomu, že při spínání směrem k napájecímu napětí je lepší používat tranzistor s kanálem P a při spínání směrem k zemi je lepší používat tranzistor s kanálem N, byl vybrán právě dvojitý budič **TC428**. Horní budič je totiž invertující (při log. 1 sepne tranzistor s P kanálem) a dolní budič je neinvertující (při log. 1 sepne tranzistor s N kanálem). Pokud bychom neměli tento typ budiče, nedal by

se použít režim **CCP1M3:CCP1M0 = 1100**. Ale problém by se dal vyřešit volbou jiného režimu.

Rezistory R_1 až R_4 pracují jako pull-down rezistory pro jednotlivé budiče. Je-li řídicí vývod ve třetím stavu, definují na vstupu log. 0. Takže jsou všechny tranzistory rozpojeny.



Obr. 18.13 Schéma zapojení přípravku **MBRIDGE**

Kondenzátory C_5 až C_8 zajišťují blokování napájecího napětí (ve schématu nejsou uvedeny), jsou důležité vzhledem k vysokým hodnotám špičkových odběrů při přepínání budičů.

Napájecí napětí je třeba připojit na svorkovnici K_1 . Dioda D_5 zajišťuje ochranu přípravku proti prepólování. LED označená **LEDP** indikuje připojení napájecího zdroje ve správné polaritě.

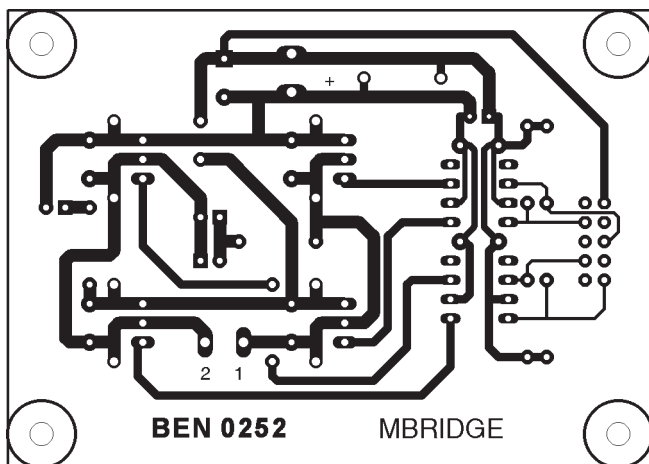
Tranzistory T_A a T_C jsou s kanálem P typu **IRF9530**, tranzistory T_B a T_D jsou s kanálem N typu **IRF530**. Diody D_1 až D_4 jsou reverzační. Oba typy tranzistorů je mají zintegrovány, ale pro případný jiný typ tranzistorů jsou přidány i jako vnější součástky. Kondenzátory C_1 až C_4 potlačují vysokofrekvenční rušení.

Výstup můstku je vyveden na konektor K_2 . Diody označené **LED+** a **LED-** indikují polaritu výstupu (jsou přidány spíše pro informaci o aktuálním stavu). **LED+** svítí, pokud jsou sepnuty T_A a T_D . **LED-** svítí, pokud jsou sepnuty T_C a T_B .

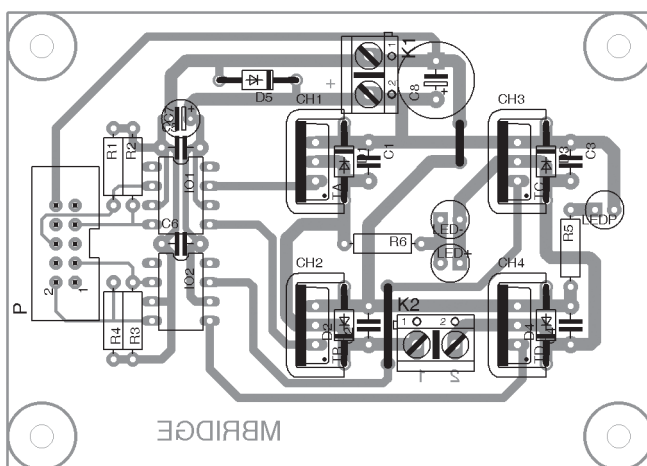
Rozpis součástek pro přípravek MBRIDGE (cena asi 120 Kč):

C_1 až C_4	CK 10P	4 ks
C_5 , C_6	CK 100N	2 ks
C_7	E2M2/50V	1 ks
C_8	E470M/25V	1 ks
D_1 až D_5	1N4004	5 ks
IO_1 , IO_2	TC428CPA+patice DIL08	2 ks

K_1, K_2	ARK500/2	2 ks
LED+, LED-, LEDP	LED 5MM	3 ks
P	MLW10G	1 ks
R_1 až R_4	RR 10K	4 ks
R_5, R_6	RR 1K	2 ks
T_A, T_C	IRF9530PBF+chladič DO1A	2 ks
T_B, T_D	IRF530PBF+chladič DO1A	2 ks



Obr. 18.14 Výkres desky plošných spojů přípravku **MBRIDGE (BEN 0252)**



Obr. 18.15 Osazovací plánec (strana běžných součástek, dvě drátové propojky)

18.6 PŘÍKLAD Č. 22

V tomto příkladu si ukážeme použití přípravku **MBRIDGE** pro regulaci otáček motorku oběma směry s použitím PWM režimu plného můstku. Příklad je konstruován podobně jako příklad č. 21.

Příklad je určen pro mikrokontrolér **PIC18F1220**. Na port B připojíme přípravek **MBRIDGE** (na svorky **K₁** jsme přivedli napájecí napětí 6 V a na svorky **K₂** jsme připojili malý motorek na 6 V). Na port A připojíme přípravek **MLEDSW**.

Nejdříve je zvolen zdroj hodin. Použijeme režim **INTIO2** a pomocí registru **OSCCON** zvolíme taktovací kmitočet 8 MHz.

Pro řízení změny otáček oběma směry použijeme proměnné **hodnota** (velikost střídy) a **režim** (0 – přímý směr, 1 – reverzní směr). Pro realizaci prodlevy čtení stavu tlačítek používáme proměnnou **d**.

Při inicializaci PWM postupujeme dle dříve uvedeného návodu. Nejdříve přepneme port B do vstupního režimu, nastavíme plnou 8bitovou periodu, PWM konfiguruje jako plný můstek v přímém režimu, vypneme generování nastavením bitu **ECCPASE**, vynulujeme příznak **TMR2IF**, nastavíme výchozí hodnotu střídy a jednotku Timer2 konfiguruje s děličkou 1:4. Kmitočet PWM je pak zhruba 1953 Hz.

Programová smyčka vyčkává na přetečení Timer2, poté aktivujeme port B jako výstupní a vynulujeme bit **ECCPASE**. Tím je PWM jednotka inicializována.

Pro správnou konfiguraci portu A musíme použít nejen registr **TRISA**, ale současně musíme přepnout vývody portu A z analogového do digitálního režimu pomocí registru **ADCON1** (viz kapitulu 4.3 a kapitulu 12.1).

Následně se rozbíhá cyklus, který sleduje stav tlačítek. Realizace je velmi podobná jako v příkladu č. 21. Rozdíl však spočívá v přepínání režimu mezi přímým a reverzním směrem.

Při přímém režimu (**režim = 0**) vede stisk tlačítka na bitu **RA0** ke zvýšení střídy a stisk tlačítka na bitu **RA1** ke snížení střídy. Při reverzním režim (**režim = 1**) je funkce tlačítek obrácená.

Tento postup je zvolen proto, aby při snižování střídy došlo postupně k přechodu z přímého do reverzního režimu (od maxima v přímém směru přes nulu do maxima v reverzním směru). A podobně při zvyšování střídy byl opět hladký přechod mezi reverzním a přímým režimem.

Režim se přepíná při dosažení nulové střídy.

Tlačítky bit0 a bit1 na přípravku **MLEDSW** lze tedy regulovat otáčky motorku oběma směry.

PROG_22.C:

```
#include <p18cxxx.h>
```

```
#pragma config OSC = INTIO2
```

```
#pragma config WDT = OFF
```

```
void main()
```

```
{
```

```

unsigned d;
unsigned char rezim=0,hodnota=50;
OSCCON=0b01110011;           //8 MHz

TRISB=255;                    //vstupy
PR2=0xFF;                     //perioda
CCP1CON=0b01001100;          //konfigurace PWM
ECCPAS=0b10000000;           //ECCPASE=1
PIR1bits.TMR2IF=0;           //nulovani TMR2IF
CCPR1L=hodnota;              //strida
T2CON=0b00000101;           //konfigurace Timer2

while (PIR1bits.TMR2IF==0);
TRISB=0;                      //vystupy
ECCPAS=0b00000000;           //ECCPASE=0

TRISA=255;
ADCON1=0b01111111;

while (1)
{
    if (PORTAbits.RA0==0 && rezim==0 && hodnota<255)
        hodnota++;
    if (PORTAbits.RA0==0 && rezim==1 && hodnota>0)
        hodnota--;
    if (PORTAbits.RA1==0 && rezim==0 && hodnota>0)
        hodnota--;
    if (PORTAbits.RA1==0 && rezim==1 && hodnota<255)
        hodnota++;
    if ((PORTAbits.RA0==0 || PORTAbits.RA1==0) && hodnota==0)
        rezim=!rezim;

    CCPR1L=hodnota;
    CCP1CONbits.P1M1=rezim;

    for (d=0;d<1000U;d++);
}
}

```