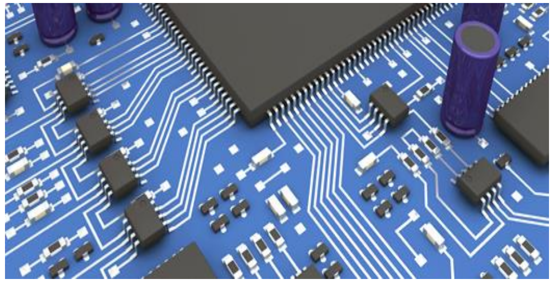
Aplikace Embedded systémů v Mechatronice



Michal Bastl
A2/713a

Aplikace Embedded systémů v Mechatronice

Obsah přednášky:

- Opakování
- PWM
- PWM PIC18
- Nastavení PWM
- Ukázky použití
- Hardware poznámky

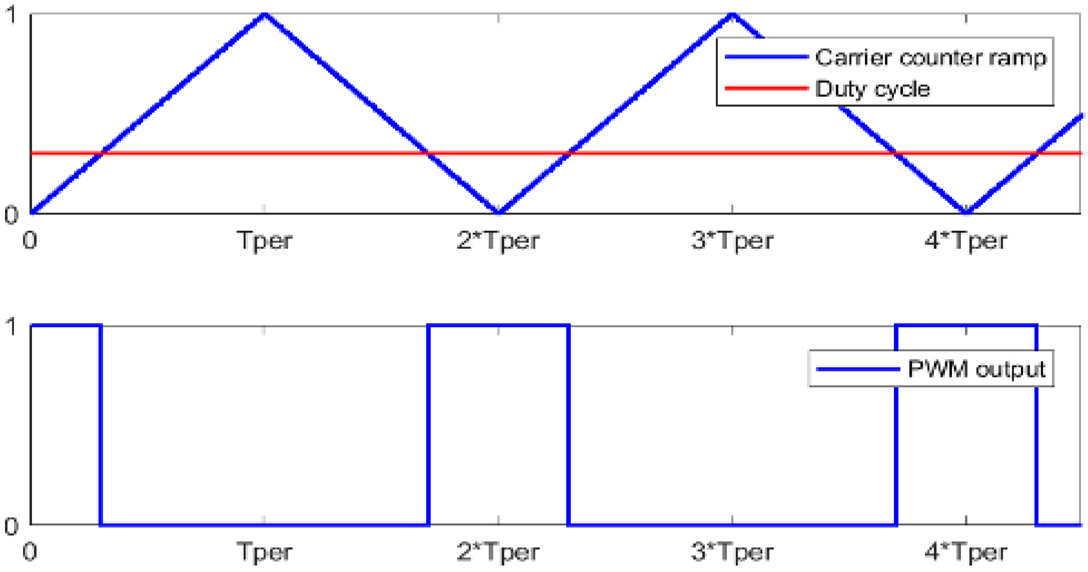
Opakování

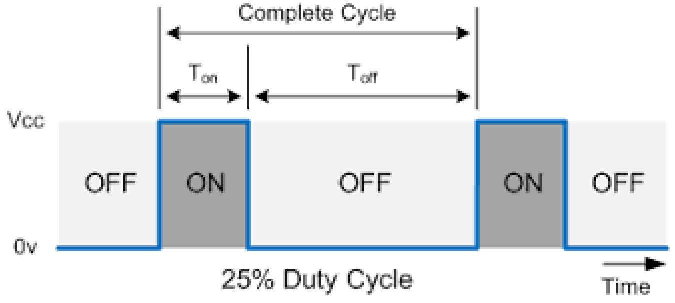
K čemu slouží ADC převodník?
Jak přepočítám výstup převodníku na napětí?
Jaký druh ADC obsahuje PIC18?

PWM

PWM – pulsně šířková modulace

- používá se k regulaci výkonu
- spínací součástky mají jen stavy zapnuto/vypnuto
- minimalizace ztrát
- je umožněna díky dynamickým vlastnostem (setrvačnost dějů)





$$D = \frac{PW}{T}$$

PWM PIC18

FIGURE 14-3: CCP PWM OUTPUT SIGNAL

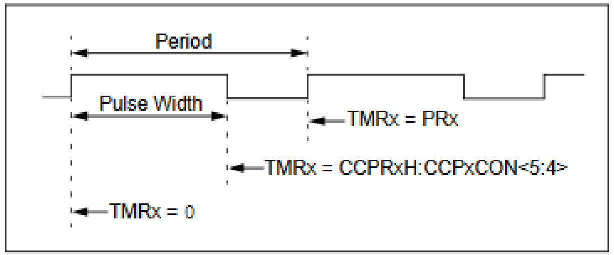
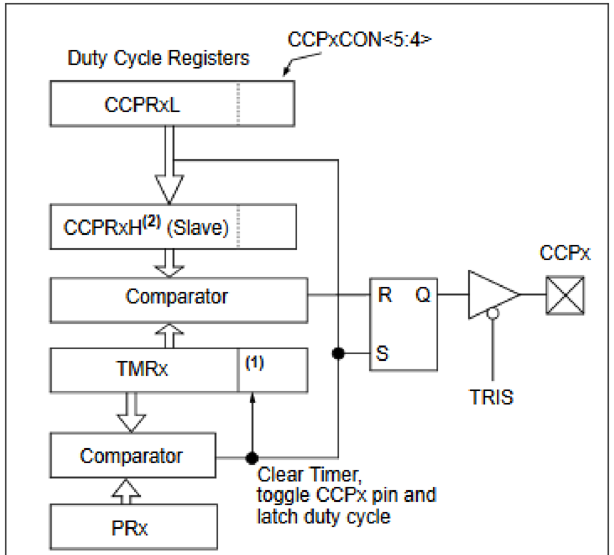
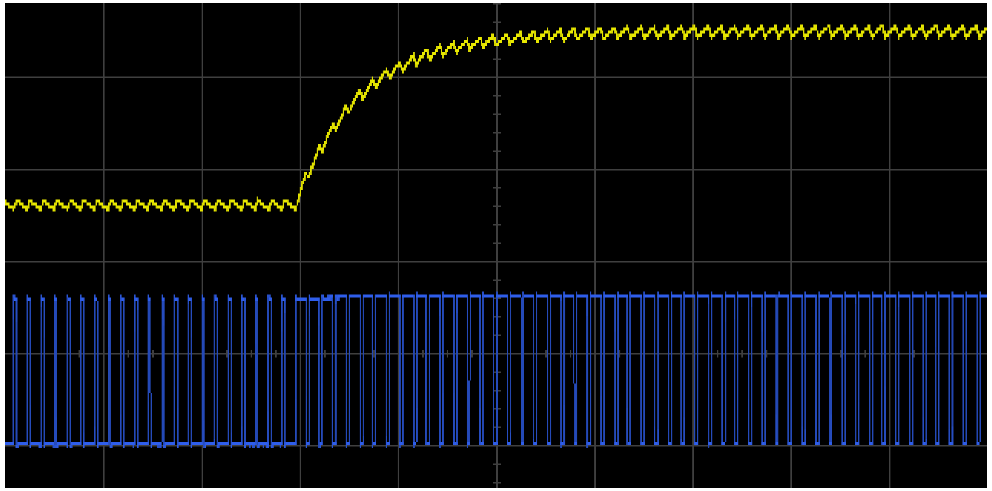


FIGURE 14-4: SIMPLIFIED PWM BLOCK DIAGRAM

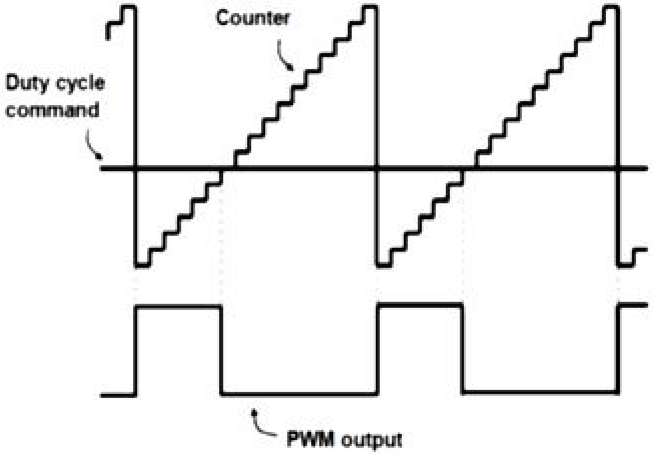


Note 1: The 8-bit timer TMRx register is concatenated with the 2-bit internal system clock (FOSC), or 2 bits of the prescaler, to create the 10-bit time base.

2: In PWM mode, CCPRxH is a read-only register.



- Přiřazení Timeru
- Zvolení periody
- Zápis střídy do registru

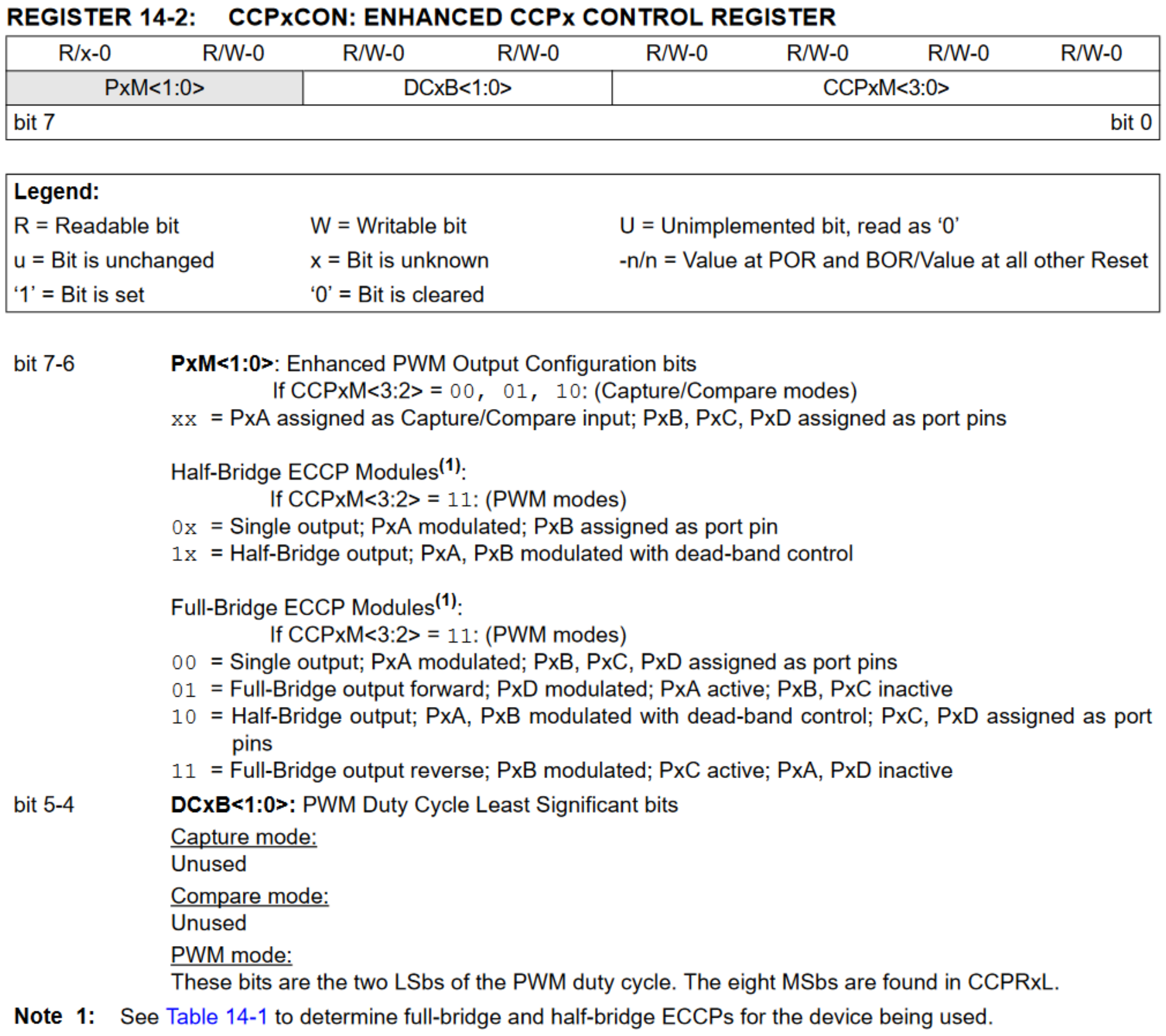


EQUATION 14-1: PWM PERIOD

$$PWM\ Period = [(PRx) + 1] \bullet 4 \bullet TOSC \bullet (TMRx\ Prescale\ Value)$$

Note 1: TOSC = 1/FOSC

PWM PIC18 registry



REGISTER 14-2: CCPxCON: ENHANCED CCPx CONTROL REGISTER

R/x-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PxM<1:0>		DCxB<1:0>		CCPxM<3:0>			
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Reset
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **PxM<1:0>**: Enhanced PWM Output Configuration bits

If CCPxM<3:2> = `00`, `01`, `10`: (Capture/Compare modes)

`xx` = PxA assigned as Capture/Compare input; PxB, PxC, PxD assigned as port pins

Half-Bridge ECCP Modules[1]:

If CCPxM<3:2> = `11`: (PWM modes)

`0x` = Single output; PxA modulated; PxB assigned as port pin

`1x` = Half-Bridge output; PxA, PxB modulated with dead-band control

Full-Bridge ECCP Modules[1]:

If CCPxM<3:2> = `11`: (PWM modes)

`00` = Single output; PxA modulated; PxB, PxC, PxD assigned as port pins

`01` = Full-Bridge output forward; PxD modulated; PxA active; PxB, PxC inactive

`10` = Half-Bridge output; PxA, PxB modulated with dead-band control; PxC, PxD assigned as port pins

`11` = Full-Bridge output reverse; PxB modulated; PxC active; PxA, PxD inactive

bit 5-4 **DCxB<1:0>:** PWM Duty Cycle Least Significant bits

Capture mode:

Unused

Compare mode:

Unused

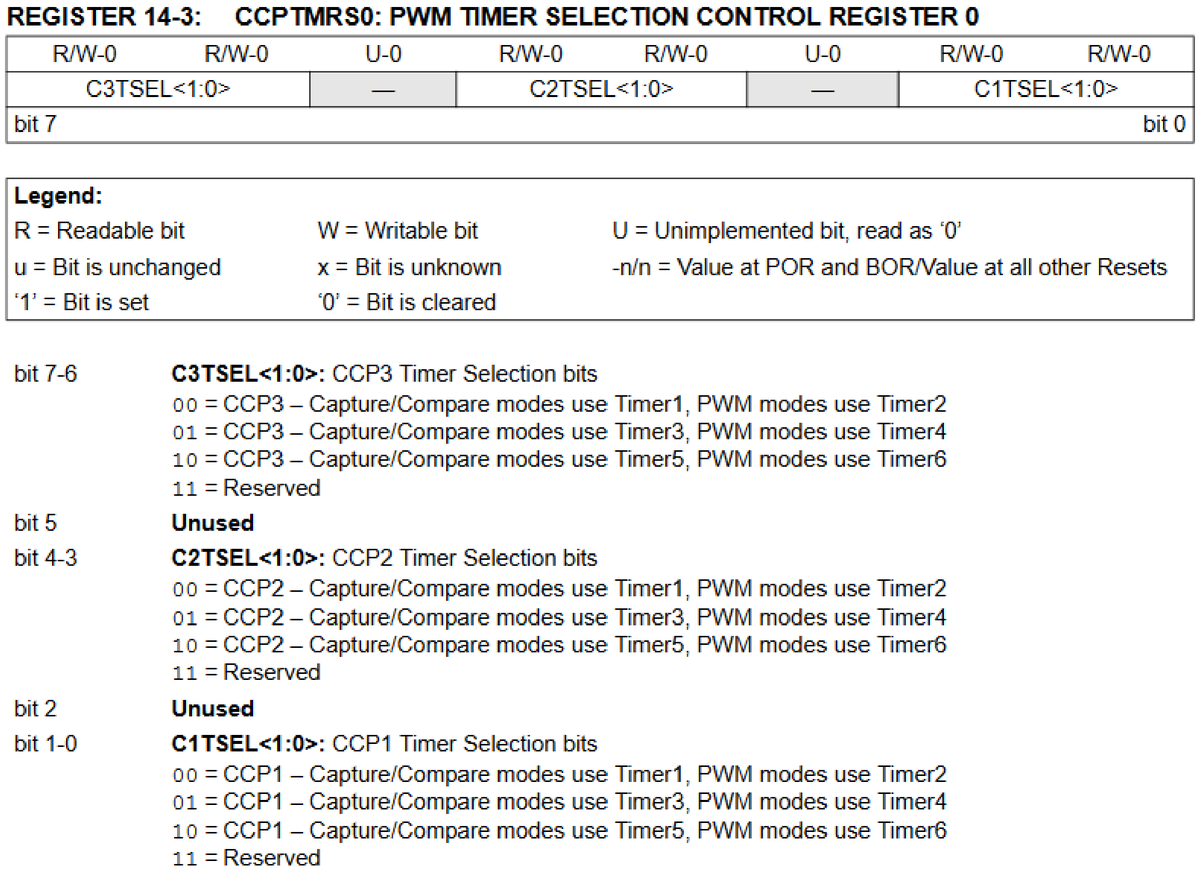
PWM mode:

These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPRxL.

Note 1: See Table 14-1 to determine full-bridge and half-bridge ECCPs for the device being used.

```
//init – PWM
PSTR1CON = 0b11;                        //P1A PWM
CCPTMRS0bits.C1TSEL = 0b00;             //timer2 will be used
PR2 = 200;                              //period
CCP1CON = 0b00001100;                   //enable PWM
CCPR1L = 200;                           //duty cycle

//init - timer2
T2CON = 0b00111101;
```

PWM PIC18 registry



REGISTER 14-3: CCPTMRS0: PWM TIMER SELECTION CONTROL REGISTER 0

R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
C3TSEL<1:0>		—	C2TSEL<1:0>		—	C1TSEL<1:0>	
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-6 **C3TSEL<1:0>:** CCP3 Timer Selection bits
00 = CCP3 – Capture/Compare modes use Timer1, PWM modes use Timer2
01 = CCP3 – Capture/Compare modes use Timer3, PWM modes use Timer4
10 = CCP3 – Capture/Compare modes use Timer5, PWM modes use Timer6
11 = Reserved

bit 5 **Unused**

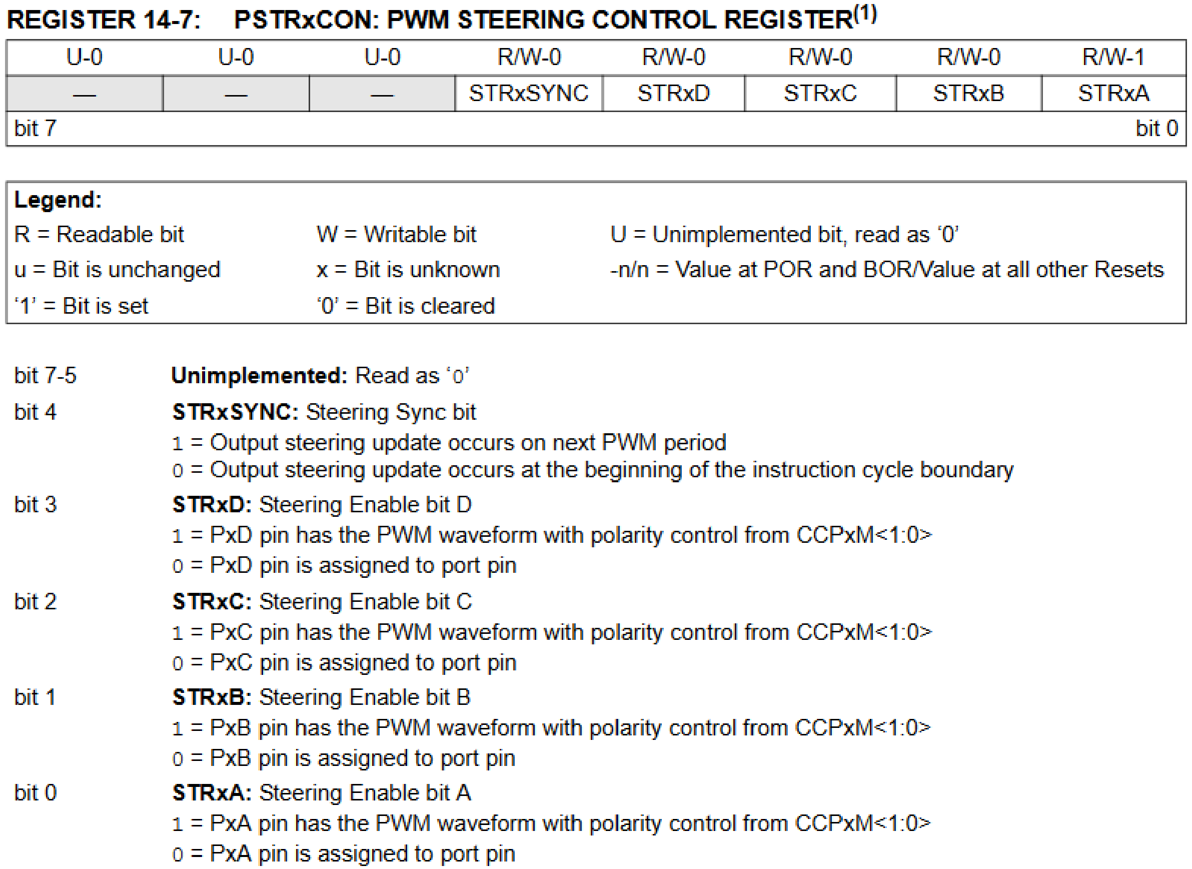
bit 4-3 **C2TSEL<1:0>:** CCP2 Timer Selection bits
00 = CCP2 – Capture/Compare modes use Timer1, PWM modes use Timer2
01 = CCP2 – Capture/Compare modes use Timer3, PWM modes use Timer4
10 = CCP2 – Capture/Compare modes use Timer5, PWM modes use Timer6
11 = Reserved

bit 2 **Unused**

bit 1-0 **C1TSEL<1:0>:** CCP1 Timer Selection bits
00 = CCP1 – Capture/Compare modes use Timer1, PWM modes use Timer2
01 = CCP1 – Capture/Compare modes use Timer3, PWM modes use Timer4
10 = CCP1 – Capture/Compare modes use Timer5, PWM modes use Timer6
11 = Reserved

```
 //init – PWM
PSTR1CON = 0b11;                          //P1A PWM
CCPTMRS0bits.C1TSEL = 0b00;               //timer2 will be used
PR2 = 200;                                //period
CCP1CON = 0b00001100;                     //enable PWM
CCPR1L = 200;                             //duty cycle

//init - timer2
T2CON = 0b00111101;
```

PWM PIC18 registry



REGISTER 14-7: PSTRxCON: PWM STEERING CONTROL REGISTER(1)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
—	—	—	STRxSYNC	STRxD	STRxC	STRxB	STRxA
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4 **STRxSYNC:** Steering Sync bit
1 = Output steering update occurs on next PWM period
0 = Output steering update occurs at the beginning of the instruction cycle boundary

bit 3 **STRxD:** Steering Enable bit D
1 = PxD pin has the PWM waveform with polarity control from CCPxM<1:0>
0 = PxD pin is assigned to port pin

bit 2 **STRxC:** Steering Enable bit C
1 = PxC pin has the PWM waveform with polarity control from CCPxM<1:0>
0 = PxC pin is assigned to port pin

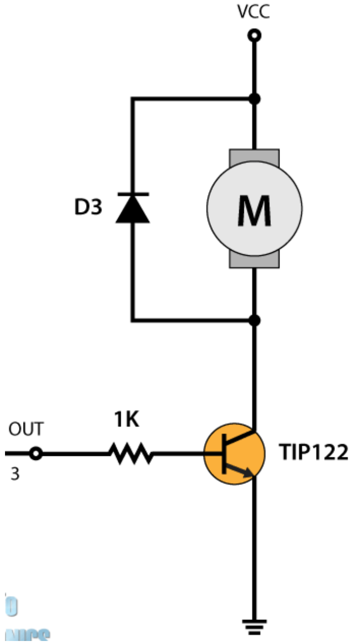
bit 1 **STRxB:** Steering Enable bit B
1 = PxB pin has the PWM waveform with polarity control from CCPxM<1:0>
0 = PxB pin is assigned to port pin

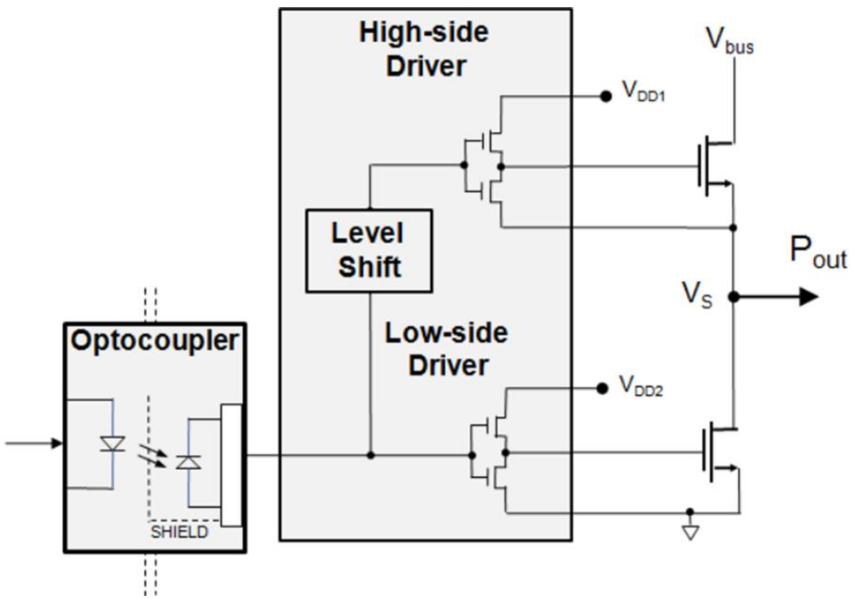
bit 0 **STRxA:** Steering Enable bit A
1 = PxA pin has the PWM waveform with polarity control from CCPxM<1:0>
0 = PxA pin is assigned to port pin

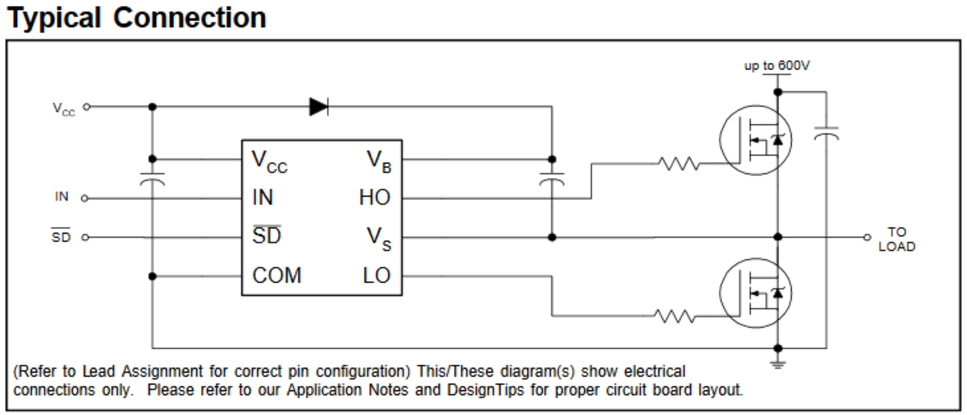
```
//init – PWM
PSTR1CON = 0b11;                          //P1A PWM
CCPTMRS0bits.C1TSEL = 0b00;               //timer2 will be used
PR2 = 200;                                //period
CCP1CON = 0b00001100;                     //enable PWM
CCPR1L = 200;                             //duty cycle

//init - timer2
T2CON = 0b00111101;
```

Regulace výkonu

- Při použití N-MOS nastává problém se spínáním horního tranzistoru
- Použití speciálního zákaznického obvodu tzv. driveru
- Infineon IR2104 (vpravo dole)
- Obvod řeší i dead-time, tedy ochranu proti sepnutí obou tranzistoru nad sebou do zkratu







(Refer to Lead Assignment for correct pin configuration) This/These diagram(s) show electrical connections only. Please refer to our Application Notes and DesignTips for proper circuit board layout.

PWM PIC18 módy

- Kromě jednoduchého single modů umožňuje PWM periferie další specializované konfigurace
- Half-bridge
- Full-bridge

Standard Half-Bridge Circuit (“Push-Pull”)

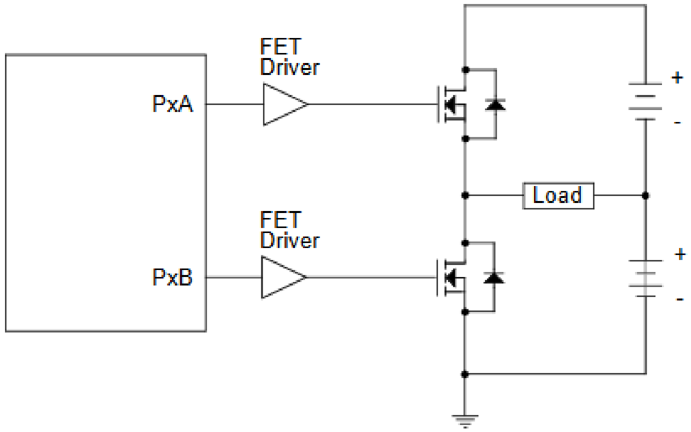
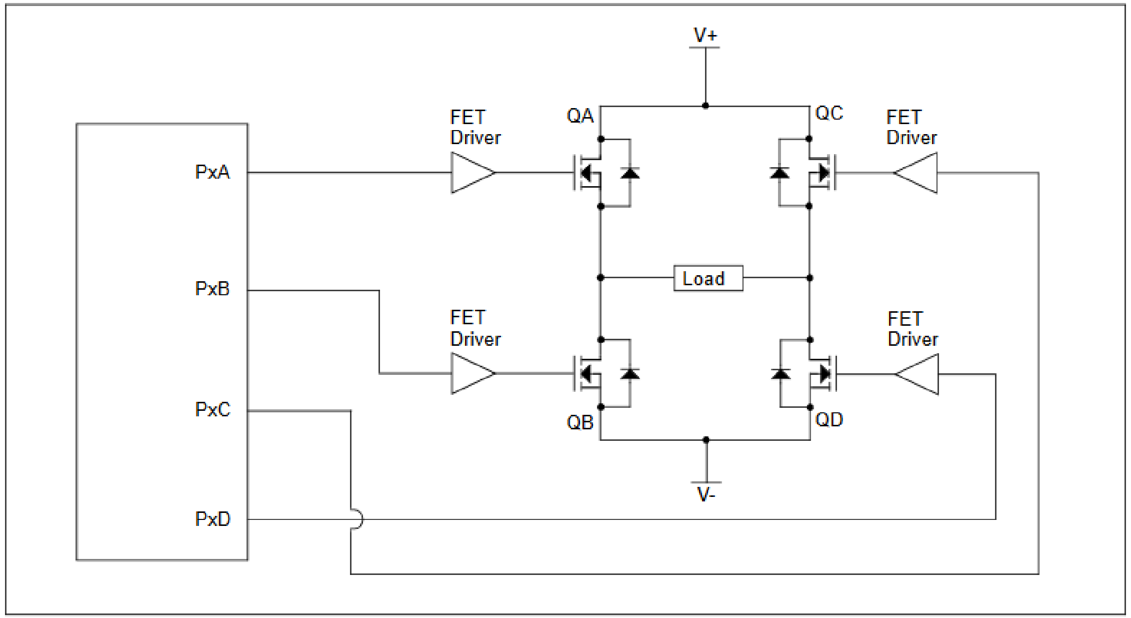
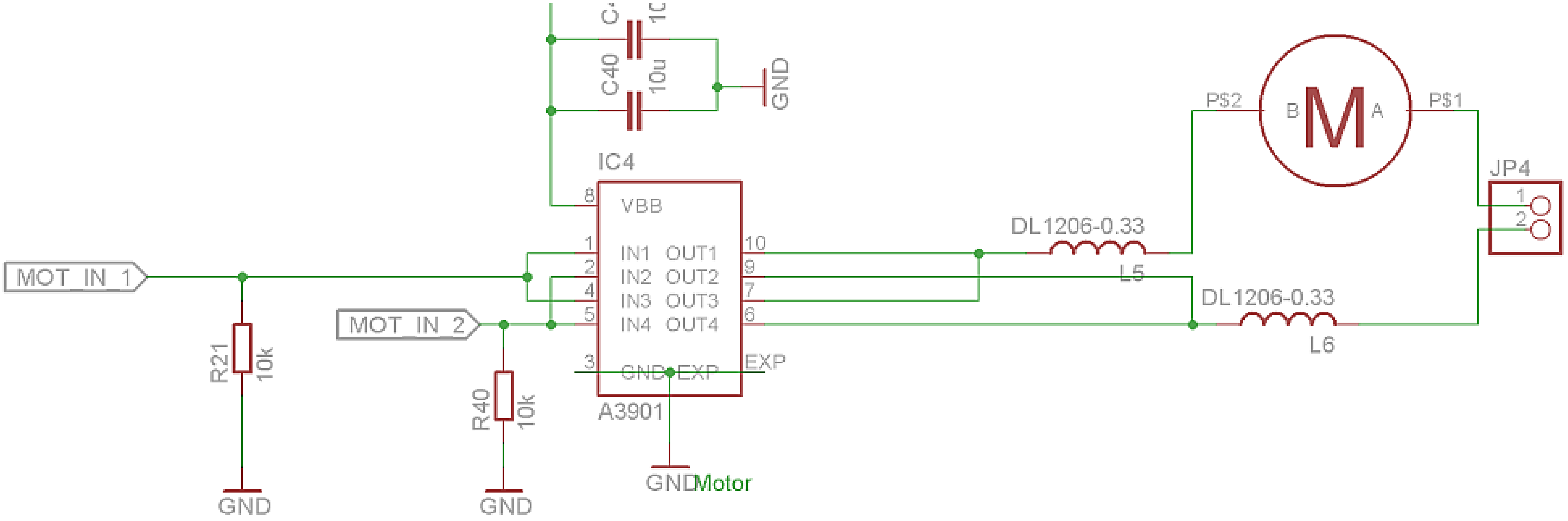


FIGURE 14-10: EXAMPLE OF FULL-BRIDGE APPLICATION



PWM PIC18 módy

- Motor lze ovládat jedním PWM MOT IN 1 Pin P1A
- MOT IN 2 pak pomocí běžného GPIO pinu volí DIR (směr motoru)
- Invertuje se i PWM!!!



Domácí úkol

1. Ovládejte rychlost motoru potenciometrem
   - V polovině rozsahu motor stojí
   - na každou stranu nastavuji rychlost motoru vpravo/vlevo
2. Aktuální rychlost(střídu) a směr otáčení zobrazujte na displeji