

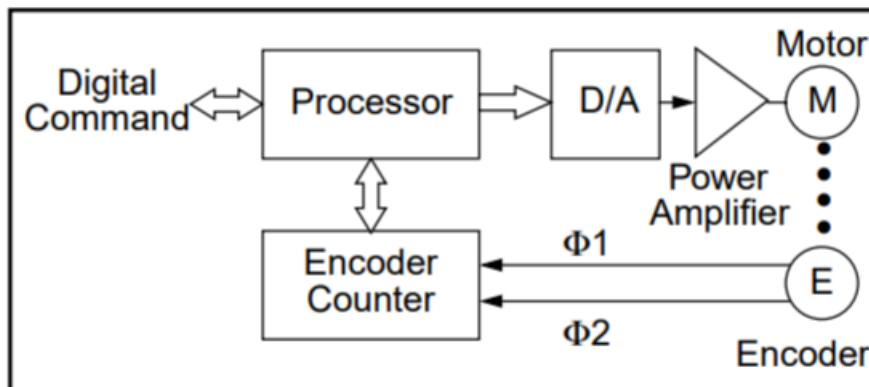
## Servo riadenie jednosmerného kefkového motora

Mikrokontrolér PIC17C42 je vynikajúcou voľbou pre vstavané aplikácie. Je postavený na Harvardskej architektúre s RISC vďaka čomu ponúka vysokú výpočtovú rýchlosť na riadenie v uzavretej slučke v reálnom čase. Výpočet PID (proporcionálny, integrálny, diferenciálny) riadenia sa môže vykonávať za menej ako 200 us (@ 16 MHz), čo umožňuje vzorkovacie časy riadiacich slučiek v rozsahu 2 kHz. Enkodér do 3 MHz dokáže ľahko ovládať PIC17C42 periférie. Ovládanie servomotora v uzavretej slučke sa zvyčajne praktizuje 16-bitovými mikrokontrolérmi a externou logikou. V snahe zvýšiť výkon sa veľa aplikácií inovuje na DSP (Digital Signal Processors). Avšak veľmi vysoký výkon PIC17C42 umožňuje implementáciu týchto aplikácií na riadenie servopohonov pri výraznom znížení celkových systémových nákladov. Servosystém zahrňujúci mikrokontrolér PIC17C42, programovateľné logické zariadenie (PLD) a jednočipový H-mostík by sa mohol použiť ako regulátor polohy v tlačiarňach alebo skeneroch. Nízke náklady na implementáciu systému riadenia serva pomocou PIC17C42 umožňujú tomuto systému priaznivú konkurenciu so systémami krokových motorov tým, že ponúkajú množstvo výhod:

- Zvýšené zrýchlenie, rýchlosť
- Zvýšená účinnosť
- Znížený hluk
- Skutočné odmietnutie poruchy

Moderné digitálne servosystémy sú tvorené tak, ako je znázornené na obrázku 1

Obrázok 1



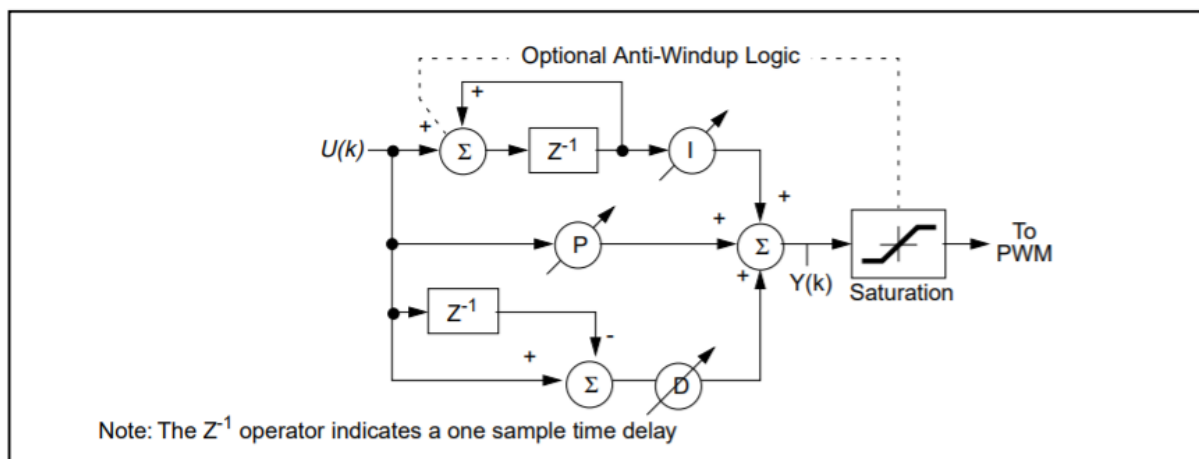
Tieto systémy riadia motor pomocou zariadenia s inkrementálnou spätnou väzbou, známeho ako sekvenčný enkodér. Pozostávajú z počítača, procesora, nejakej formy D / A (digitálno-analógového) prevodníka a výkonového zosilňovača, ktorý dodáva do motora prúd alebo napätie.

PIC17C42 implementuje algoritmus servo kompenzátora a generovanie trajektórie (lichobežník). Algoritmus generovania trajektórie je potrebný pre optimálny pohyb a jeho implementácia je rovnako dôležitá ako samotný servo kompenzátor. Kompenzátor serva môže byť implementovaný ako tradičný digitálny filter, fuzzy logický algoritmus alebo jednoduchý PID algoritmus (ako je implementovaný v tejto aplikačnej poznámke). Kombinácia výpočtov servo kompenzátora a trajektórie môže klást značné nároky na procesor. Konverzia D / A sa dá zvládnuť pomocou konvenčného DAC alebo pomocou pulznej šírkového modulácie PIC17C42 (PWM). V

obidvoch prípadoch sa výstupný signál privádza do výkonovej fázy, ktorá prevádza analógový signál (signály) na použiteľné napätie a prúdy na poháňanie motora. Výstupom PWM môže byť signál prevádzkového cyklu v kombinácii so smerovým signálom alebo jediný signál, ktorý prenáša obe informácie. V druhom prípade povoľuje 50% pracovný cyklus nulový výstup, 0% pracovný cyklus povoľuje maximálny negatívny výstup a 100% maximálny pozitívny výstup. Zosilňovač môže byť nakonfigurovaný na dodávanie riadeného napätia alebo prúdu do motora. Väčšina vstavaných systémov využíva výstup napätia, pretože je jednoduchší a lacnejší. Sekvenčné kódovače produkujú kvadratúrne impulzné vlaky, z ktorých je možné odvodiť polohu, rýchlosť a smer otáčania motora. Frekvencia je úmerná rýchlosti a každý prechod F1 a F2 predstavuje prírastok polohy. Fáza signálov sa používa na určenie smeru otáčania. Tieto signály kódovača sú obvykle dekodované do impulzov Count Up a Count Down pomocou malého stavového stroja. Tieto impulzy sú potom smerované do N-bitového čítača nahor / nadol, ktorého hodnota zodpovedá polohe hriadeľa motora. Dekodér / počítadlo sa môže implementovať v hardvéri, softvéri alebo ich kombinácii.

Program PID je najpoužívanejší algoritmus na riadenie servomotora. Aj keď to nemusí byť optimálny radič pre všetky aplikácie, je ľahké ho pochopiť a vyladiť. Forma štandardného algoritmu digitálneho PID je znázornená na obrázku 2. U(k) je chyba polohy alebo rýchlosti a na výstupe je Y(k). Tento algoritmus bol implementovaný pomocou matematickej knižnice PIC17C42. Vyžaduje sa iba 800 inštrukčných cyklov, čo vedie k času vykonávania PID 0,2 ms pri 16 MHz. Windup integrátora je stav, ktorý sa vyskytuje v PID regulátoroch, keď je v systéme prítomná veľká nasledujúca chyba, napríklad keď sa vyskytne rušenie veľkým krokom. Integrátor sa počas tohto nasledujúceho chybového stavu neustále vytvára, aj keď je výstup nasýtený. Integrátor sa potom „uvoľní“, keď servosystém dosiahne svoje konečné miesto určenia, čo spôsobí nadmerné kmitanie. Implementácia PID uvedená na obrázku 2 sa tomuto problému vyhýba zastavením činnosti integrátora počas saturácie výstupu.

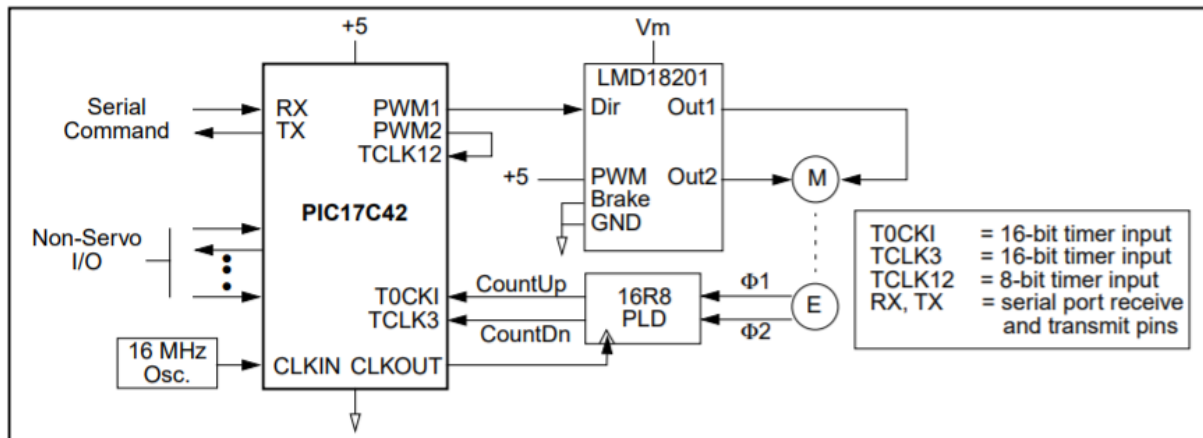
Obrázok 2



PIC17C42 obsahuje subsystém impulznej šírkovj modulačie s vysokým rozlíšením (PWM). Toto vytvára veľmi efektívny výkonový D / A prevodník, keď je pripojený k jednoduchému stupňu spínania. Rozlíšenie subsystému PIC17C42 PWM je 62,5 ns (pri 16 MHz). To sa premieťa do 10-bitového rozlíšenia rýchlosťou 15,6 kHz alebo 1 časť v rozlíšení 800 (9 1/2 bit) pri 20 kHz. To umožňuje účinnú reguláciu napätia pri súčasnom udržiavaní modulačnej frekvencie na alebo nad hranicou ľudského sluchu. Toto je obzvlášť dôležité v kancelárskych automatizačných zariadeniach, kde je minimalizácia hluku cieľom dizajnu. Motor reaguje na výstupnú fázu PWM časom a spriemeruje pracovný cyklus výstupu. Väčšina motorov reaguje pomaly a má elektrickú časovú

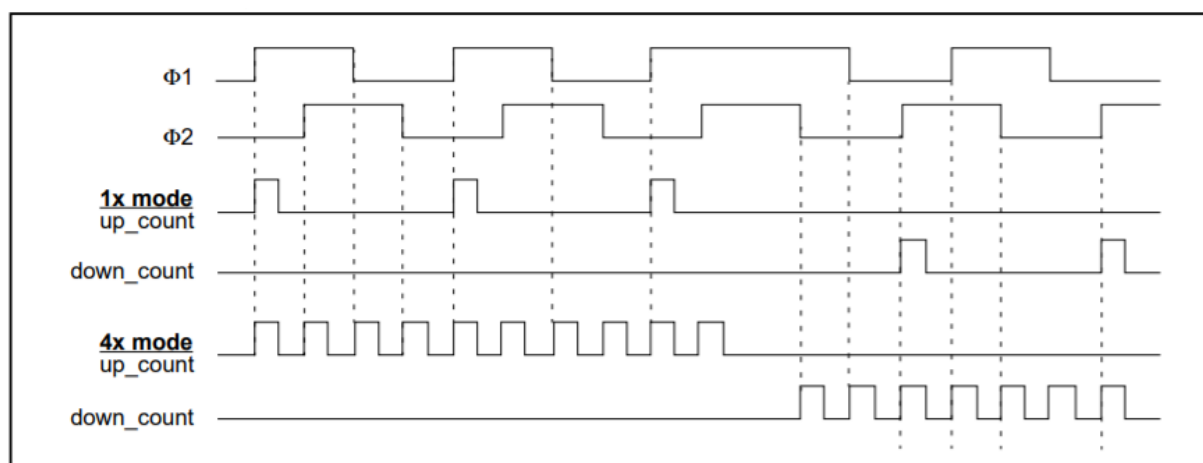
konštantu 0,5 ms alebo viac a mechanickú časovú konštantu 20,0 ms alebo viac. PWM výstup 15 kHz je účinne ekvivalentný výstupu lineárneho zosilňovača. V systéme znázornenom na obrázku 3 je smerový vstup mostíka H zapojený priamo na výstup PIC17C42 PWM. H-most je napájaný jednosmerným napájacím napätím  $V_m$ . V tejto konfigurácii je 0 V dodávaný motoru, keď je signál PWM v 50% pracovnom cykle,  $-V_m$  pri 0% pracovnom cykle a  $+V_m$  pri 100% pracovnom cykle.

Obrázok 3



Spätná väzba polohy pre príklad systému je odvodená od kvadratúrneho kódovača namontovaného na hriadeli motora. Z tohto lacného zariadenia je možné odvodiť inkrementálnu polohu aj smer. Signály kvadratúrneho kódéra sa spracúvajú pomocou PLD zariadenia typu 16R8, ako je znázornené na obrázku 3. PLD prevádza kvadratúrne impulzy na dva pulzné toky: Count Up a Count Down (Obrázok 4). Tieto signály sa potom privádzajú do dvoch 16-bitových časovačov PIC17C42 (Timer3 a Timer0). PIC17C42 sleduje prírastkovú polohu hriadeľa motora tým, že odlišuje tieto dva 16-bitové časovače. Táto operácia sa vykonáva pri každej vzorke serva a aktuálna poloha sa vypočíta pripočítaním prírastkovej polohy k predchádzajúcej polohe. Pretože oba časovače sú 16-bitové, nie je potrebné sledovanie pretečenia, pokiaľ nie je frekvencia signálov kódovača väčšia ako 32767-násobok vzorkovacej frekvencie. Napríklad v čase vzorkovania serva 1 ms by maximálna rýchlosť kódéra bola 3,2767 MHz. Obojstranné vyrovnanie nie je problémom, pretože sa používa iba rozdiel medzi týmito dvoma počítadlami. Odčítanie dvoch doplnkov sa o to automaticky postará. Veľkosť vnútorného registra polôh však nie je nijako obmedzená. Pridaním 16-bitovej inkrementálnej polohy každej vzorky do N-bajtového softvérového registra sa môže udržiavať N-bajtová pozícia.

Obrázok 4



Algoritmus generovania trajektórie je nevyhnutný pre optimálne riadenie pohybu. V tejto aplikácii je implementovaná lineárna trajektória po častiach. Pri polohovom pohybe sa rýchlosť zvyšuje o hodnotu konštantného zrýchlenia, až kým sa nedosiahne stanovená maximálna rýchlosť. Maximálna rýchlosť sa udržiava po požadovanú dobu a potom sa znižuje o rovnakú hodnotu zrýchlenia (spomalenia), až kým sa nedosiahne nulová rýchlosť. Trajektória rýchlosti je preto lichobežníková pre dlhý pohyb a trojuholníková pre krátky pohyb, keď sa nedosiahla maximálna rýchlosť (obrázok 5). Podprogram doPreMove sa vyvolá raz na začiatku pohybu, aby sa vypočítali limity trajektórie. Program doMove je potom vyvolaná pri každej vzorke na výpočet nových „požadovaných“ hodnôt rýchlosti a polohy nasledovne:

$$VK = VK-1 A \quad (A = \text{zrýchlenie})$$

$$PK = PK-1 VK-1 A / 2$$

Obrázok 5

