

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechniky

BAKALÁŘSKÁ PRÁCE



Laboratorní přípravek s pásovým dopravníkem

Vedoucí bakalářské práce: Doc.Ing. Kateřina Hyniová, CSc.

Praha, 2007

Martin HOLEC

Prohlášení:

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, SW, atd.) uvedené v příloženém seznamu.

V Praze dne:

Podpis:

Poděkování:

Rád bych touto cestou srdečně poděkoval především Doc. Ing. Kateřině Hyniové, CSc. za vstřícnost, ochotu a pomoc podávanou po celou dobu zpracovávání mé bakalářské práce. Dále bych rád poděkoval Jiřímu Kubiasovi za pomoc při konzultování technických problémů a technickou podporu.

Anotace:

Cílem práce bylo vytvořit model pásového dopravníku pro potřeby studentů ČVUT Praha, a to jak z hlediska návrhu elektronické části, tak samotné fyzické realizace.

Hlavními částmi úlohy jsou pásový dopravník, poháněný stejnosměrným motorkem, na který je připevněno inkrementální čidlo pro detekci rychlosti, dále pak optická závora pro detekci předmětu a řídicí jednotka. Ta pomocí procesoru vyhodnocuje rychlost posuvu pásu a signálu z optické závory a tudíž je schopna rozpoznat, pokud se snímané objekty nacházejí těsně za sebou.

Zhotovený přípravek pomůže studentům při získávání znalostí a zkušeností s principy inkrementálního senzoru otáček a optického čidla pro detekci předmětů, a to především pomocí možnosti si dané přístroje prakticky vyzkoušet a sledovat jejich chování.

Annotation:

The goal of my bachelor work was to build a transporter for usage of students of CTU Prague. Both parts, electrical and mechanical were meant to be realized.

Main sections of the instrument are belt transporter ran by DC motor equipped by an incremental sensor for speed detection, followed by optical gate for detection of objects' presentation and control unit. The control unit uses a processor to evaluate signals from incremental sensor and optical gate to recognize, how many objects passed through the gate, even though they are stacked together.

This instrument will be helpful for students to get knowledge and experiences with used sensors, mostly because of the opportunity to see them working practically in use.

Obsah

1. Úvod	6
2. Úvodní úvaha	6
3. Softwarová část	7
3.1. Výběr procesoru	7
3.2. Hlavní program	7
3.3. Shrnutí	13
3.4. Obsluha LCD displeje	13
4. Elektronická část	14
4.1. Úvodní myšlenka	14
4.2. Testovací obvod na nepájivém poli	14
4.3. Zhotovení plošného spoje	15
4.4. Osazení a oživení finálního obvodu	16
5. Senzory	18
5.1. Senzor rychlosti motorku	18
5.2. Senzor přítomnosti objektu	19
6. Mechanická část	20
6.1. Konstrukce	20
7. Závěr	23
8. Přílohy	24
8.1. Příloha 1 – Katalogový list Atmega88	24
8.2. Příloha 2 – Katalogový list Omron	25
8.3. Příloha 3 – Katalogový list Sick	29
8.4. Příloha 4 – Zdrojový kód programu	32
9. Použitá literatura	36

1. Úvod

Studium nejen technických oborů je efektivnější a pro studenty příjemnější, pokud mají možnost si studovanou látku v praxi ověřit a vyzkoušet. Díky těmto metodám je snazší lépe a hlouběji porozumět teoreticky přednášeným faktům.

Laboratoře vysokých škol jsou vybaveny řadou simulačních úloh pro tyto účely. Přípravky jsou prospěšné jak studujícím ve fázi studia, tak i v případě pokud absolvent hodlá své poznatky předat mladším ročníkům a své zkušenosti a vědomosti nabyté během let na vysoké škole shrnout do závěrečné bakalářské či diplomové práce.

Již od střední školy se zajímám o automatizační techniku, proto mě velmi zaujalo zadání s tímto odvětvím úzce spojené, a to zhotovit model dopravníku osazeného senzory pro vyhodnocování a zpracování signálů dle požadavku zadání. K výběru této práce mě vedl mimo sympatií k zadání a zájmu o jeho řešení i fakt, že mohu přispět svým nástupcům k rozšiřování obzorů v těchto sférách, dále pak že díky řešení problémů spojených se zpracováváním bakalářské práce mohu prohloubit své znalosti týkající se problematiky v několika odlišných odvětvích, a to nejen v návrhu elektronického obvodu a programování řídicího mikročítače, ale i fyzická konstrukce celé úlohy.

2. Úvodní úvaha

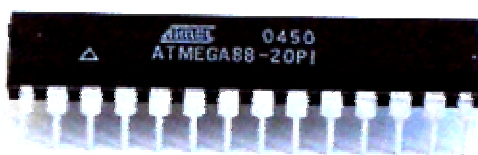
Na počátku veškeré snahy o realizaci bylo třeba se zamyslet nad činností, kterou má finální přípravek zastávat a dle těchto kritérií zvolit příslušné komponenty. Prvním krokem bylo stanovení signálů, které má řídicí jednotka zpracovávat a jaké akční zásahy se od ní čekají. Dle požadavků zadání bylo třeba splnit i možnost, při které jsou objekty přítomné na dopravníku umístěny těsně za sebou a rychlost pohybu může být regulovatelná. Tento sled faktů vyžadoval nutnost opatřit úlohu nejen jednoduchou elektronikou pro ovládání, ale také mikroprocesorem, který bude veškeré vstupní informace zpracovávat. Jako vstupy byly stanoveny signály z inkrementálního senzoru otáček motoru a optického senzoru pro detekci přítomnosti objektů. Jako akční člen byla regulace rychlosti pohybu pásu dopravníku.

Pro fyzickou realizaci bylo nutné si uvědomit, že se jedná o produkt určený studentům a student je, jak je všeobecně známo, tvor zvědavý. Proto celý panel bylo potřeba zhotovit z odolného materiálu, bez přebytečných viditelných kabelů. Celý řídicí systém je tedy vložen ve spodní části panelu, v horní části je přístupný pouze ovládací panel a dopravník se senzory.

3. Softwarová část

3.1 Výběr mikroprocesoru

Pro realizaci řídicí jednotky byl zvolen mikroprocesor od společnosti Atmel, a to model ATmega88 (obr. 1). Jedná se o osmibitový mikrokontrolér s 8 KB programovatelné paměti FLASH. Detailní katalogový list (stažen z oficiálních stránek Atmelu) je obsažen v příloze (příloha 1). Pro danou aplikaci jsem dále z výbavy tohoto modelu využil jeden ze dvou osmibitových čítačů pro čítání přerušovacích vektorů a generaci PWM signálu.



Obrázek 1 – mikrokontrolér Atmega88

Právě schopnost generovat PWM signál mě navedla k myšlence zrealizovat regulaci rychlosti posuvu dopravního pásu touto cestou, nikoli užitím potenciometru, který by byl v krajní poloze vzhledem k odběru motoru při plném napětí značně zatěžován a hrozilo by zničení.

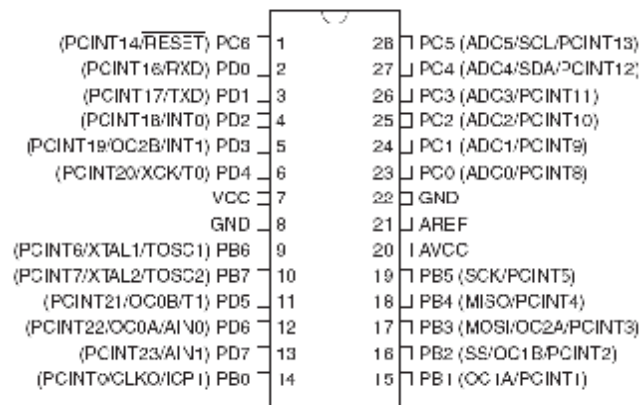
Další výhodou a ukazatelem pro volbu tohoto modelu byla skutečnost, že procesory Atmel lze programovat ve velmi příjemném prostředí AVRStudia, využívající pro psaní kódu běžnou syntaxi jazyka “C”, která je daleko přehlednější a uživatelsky příjemnější, než klasické programování assemblerovským způsobem. AVRStudio je volně stažitelné ze stránek společnosti Atmel, www.atmel.com. Jedinou podmínkou je jednoduchá registrace.

3.2 Hlavní program

Při tvorbě hlavního programu jsem s ohledem na mé zkušenosti s tímto procesorem postupoval od začátku. Zhotovil jsem jednoduchý program pro blikání LED diodou na definovaném výstupním portu. Tato aplikace nejlépe uživatele seznámí s vlastnostmi daného procesoru. Následovalo studium literatury, ze které jsem zvolil knihu “Práce s mikrokontroléry ATMEL AVR – Atmega 16” napsanou panem Davidem Matouškem, vydanou nakladatelstvím BEN. Publikace je psaná pro modelovou řadu Atmega 16, tudíž byla nutná modifikace označování registrů s využitím katalogového listu Atmega 88 pro zajištění správné funkce.

V počátku programu je nutné importovat několik knihoven, jejichž definice jsou v kódu využívány. Jedná se především o knihovny **io.h** a **iom88.h** pro práci s vstupně – výstupními porty procesoru, dále pak obsluhu LCD displeje nazvanou **lcd.h**. Následuje definice proměnných. Mezi nejdůležitější, napevno nadefinované, patří proměnná *KOSTKA*, která obsahuje komparační hodnotu ve formě počtu pulzů inkrementálního snímače, které jsou načítány pro průběh jednoho předmětu optickou závorou. Dále jsou užity proměnné *pulzy*, která slouží jako zásobník pro počítání pulzů z inkrementálního čidla, následně proměnné pro záznam hodnot kostek, a to *pocetkostekISR* a *pocetkostekCOMPL*. První ze zmíněných je víceméně pomocná proměnná, sloužící právě v případě, pokud jsou předměty na pásu umístěny těsně za sebou. Jakmile se sled kostek na dopravníku přeruší, proměnná se nuluje. Díky přítomnosti této proměnné se počet na displeji aktualizuje i během spojitého uspořádání předmětů, ne až po skončení řetězce. *PocetkostekCOMPL* slouží pro ukládání finální hodnoty. Poslední proměnnou je *upgLCD*, což je pomocná proměnná určená pro práci s LCD displejem, zejména pro jeho aktualizaci v případě nutnosti.

Následuje řada funkcí, jejichž úlohu bych rád více přiblížil. Nejprve je ale třeba seznámit se s rozložením pinů procesoru pro lepší porozumění užitých výrazů. K nahlédnutí na obr. 2 je popis jednotlivých vývodů mikropočítače Atmega88.



Obrázek 2 – popis pinů Atmega88

Než přistoupím k popisu jednotlivých funkcí, je třeba stručně nastínit principy přerušení. Při zaznamenání příslušného přerušení program vyvolá obslužnou rutinu po jejímž dokončení se vrací zpět na místo, kde přerušení běhu programu vykonal. Této vlastnosti využívám jak pro čítání pulzů z inkrementálního čidla, tak pro zajištění čítání objektů pouze za podmínky, že je před optickou závorou detekován předmět. Rutina *ISR (Int1_vect)* slouží pro čítání pulzů inkrementálního čítače. Výstup čidla je přiveden na pin 5 procesoru, kde se nachází detekce

přerušení INT1. Tedy pro každý pulz snímače rychlosti se vyvolá obsluha přerušení, která způsobí inkrementaci proměnné *pulzy*. Následuje porovnávání a vyhodnocování hodnot proměnných. Uvažuje se počet načítaných pulzů v porovnání s referenční hodnotou pro jednu kostku. Tímto způsobem je ošetřena detekce natěsno naskládaných objektů. Tedy pokud je detekován objekt, zjišťuje se jeho délka. Pokud je větší než referenční hodnota pro jeden objekt, referenční hodnota se odečte od načítané a inkrementuje se počet objektů zobrazovaných na displeji. Procedura se opakuje po celou dobu, kdy je platný signál z optické závory o přítomnosti objektu.

Rutina *ISR (Int1_vect)* zajišťuje rozpoznávání signálu z optické závory a nastavuje podmínky, které jejím stavům odpovídají. V zásadě je touto rutinou realizováno překlápění stavů, kdy je povoleno počítat pulzy z inkrementálního čítače (a tedy i počet objektů), a kdy ne, neboli jde o povolování a zakazování detekce přerušení *INT1* ze senzoru rychlosti motorku. V případě, že čekáme na příchod předmětu před optickou závoru, nastaví se podmínky tak, aby bylo možné sledovat stav z inkrementálního čítače (přerušení *INT1* povoleno, nastavení bitů registru *EICRA* pro tento stav je *ISC00=1* a *ISC01=1*). Pokud přijde náběžná hrana, povolí se přerušení *INT1* a nastaví se podmínky pro reakci na sestupnou hranu, jakmile přijde, zakáže se přerušení *INT1* a obnoví se podmínky pro detekci náběžné hrany. Jednoduše řečeno, smyslem této rutiny je zabezpečit, abychom zpracovávali počet pulzů z inkrementálního čidla, ze kterého je odvozen počet objektů pouze tehdy, pokud máme objekty detekovány. Jinými slovy, cílem je zabezpečit, abychom zpracovávali počet pulzů z inkrementálního čidla, ze kterého je odvozen počet objektů pouze tehdy, pokud máme-li objekty detekovány.

Další funkcí po výše zmíněných je *preruseni*. Tato funkce je volána z hlavního programu při startu a její hlavní činnost je nastavení bitu *SREG_I* do jedničky, což způsobí globální povolení přerušení. Bez tohoto bitu bychom nebyli schopni detekovat jakákoli přerušení. Je to vlastně jakýsi hlavní klíč k přístupu k přerušením. Doplňující funkcí je nastavování bitů registrů *EICRA* a *EMSK*, které způsobí stav, kdy čekáme na příchod vzestupné hrany z optické závory, a tedy přítomnost prvního předmětu.

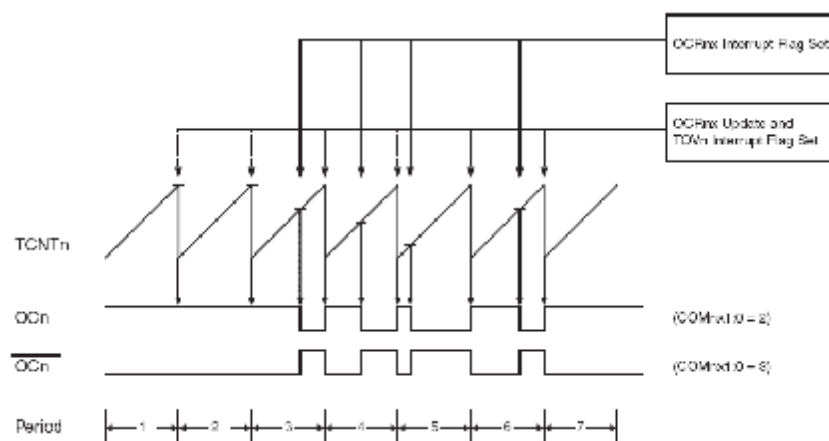
Funkce *cekej* je triviální funkcí známou již ze základů programovací techniky, která v závislosti na parametru *x* generuje dekrementaci s každým hodinovým cyklem zpoždovací smyčku o dané délce. Tato funkce je v programu využívána například jako zpoždění v úvodu startu programu pro správnou inicializaci LCD displeje, dále pro vyčkání při zobrazení úvodního uvítacího loga “ČVUT Praha”, nebo při stisku tlačítek, jako ošetření, pokud by došlo k odskočení kontaktů, aby jedno stisknutí nebylo vyhodnoceno jako vícenásobné.

Ovládací panel přípravku je realizována třemi tlačítky. Resetem počtu načítaných předmětů, tlačítky pro inkrementaci a dekrementaci rychlosti posuvu dopravníku. Z pohledu procesoru je nutné tyto ovládací prvky nadefinovat jako vstupy, o což je postaráno v deklarační funkci *tlacitka*, a to nastavením bitů *DDxy* registru *DDRx* na logickou nulu. Tím procesor vnímá daný pin jako vstupní. Dalším krokem je aktivování pull-up rezistorů pro dané piny nastavením bitů *Pxy* registrů *PORTx* do logické jedničky. Pro identifikaci procedury, při které se vyhodnocuje, které tlačítko bylo zmáčknuto byl použit podmíněný příkaz *if*, ve kterém se sledují stavy logických úrovní na jednotlivých vstupních pinech. Přičemž při detekci zmáčknutí tlačítka (tedy logické jedničky) zapisují do pomocné proměnné *pom* hodnotu, kterou dále zpracují ve funkci pro obsluhu tlačítek. Pokud je hodnota proměnné *pom* 1, pak se provede reset displeje, pokud je přiřazená hodnota rovna 2, provede se zrychlení posuvu dopravníku, v případě hodnoty 3 naopak zpomalení, a to následovně. Pro reset se jednoduše nuluje proměnná *pocetkostekCOMPL*, která uchovává informaci o celkovém počtu napočítaných předmětů a provede se upgrade displeje, tedy zápis nové hodnoty – nuly. V případě zrychlování a zpomalování motoru bylo potřeba se více zamyslet, jelikož v reálné situaci se dopravník choval odlišně, než v teoreticky navrženém případě. Pro vysvětlení této části je nutno lehce nastínit princip.

Jak již bylo dříve zmíněno, procesor sám o sobě umí generovat PWM signál, který je vhodný pro ovládání rychlosti. O generaci PWM signálu budu hovořit ve zvláštním odstavci, proto zde použiji obecnější popis pro pochopení problematiky. Klíčovou roli zde hraje osmi bitový registr *OCROB*, který obsahuje komparační hodnotu pro určování střídy PWM signálu. Rozsah tohoto registru je 255 hodnot, přičemž hodnota 255 symbolizuje v tomto případě zastavený dopravník, hodnota 0 pak plnou rychlost, tedy přivedení plného napájecího napětí 9V. Původní myšlenka byla umožnit uživateli volbu pěti stupňů rychlosti, včetně zastavení. Číslo pět bylo voleno z důvodu ideální dělitelnosti čísla 255 pětkou. Pro každý krok by se hodnota v registru zvětšila (pro zpomalování), či zmenšila (pro zrychlování). Realita však ukázala, že při tomto způsobu řešení se dopravník na první rychlost viditelně vůbec nerozeběhne. Proto následovala řada výpočtů a pokusů na optimalizaci běhu aplikace. Výsledkem je nelineární řízení. Z klidu se při přidání rychlosti provede skok v registru *OCROB* z hodnoty 255 na hodnotu 155, tedy o sto (což je dvojnásobná hodnota, než v případě původní myšlenky). Při tomto stavu se již dopravník viditelně pohybuje. Následuje 155 hodnot (155 až 0), které jsou rozděleny opět celočíselným dělením na pět kroků o totožných hodnotách velikosti 31.

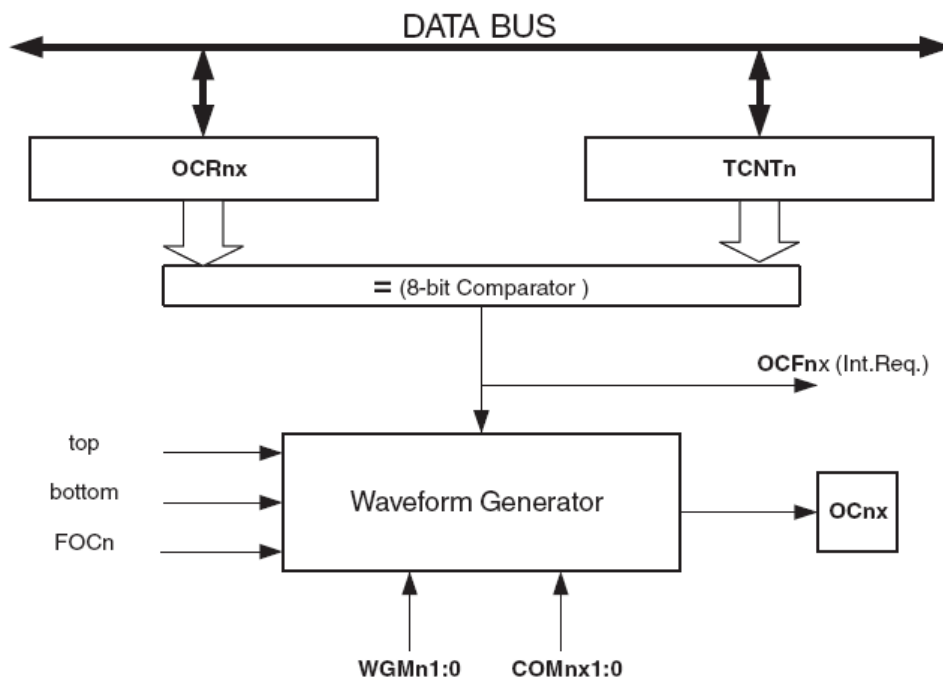
Pokud je tedy dopravník zastaven, pak $OCROB = 255$. Přidáním rychlosti na první stupeň se $OCROB = 155$. Pro zrychlování se vždy dekrementuje o hodnotu 31 až do nuly. Pro zpomalování je postup alternativní, pouze jakmile je zaznamenána hodnota $OCROB = 155$, tedy nejpomalejší rychlost, při dalším stisku tlačítka pro zpomalení nedojde k inkrementaci o 31, ale k přímému skoku na hodnotu 255, a tím k zastavení pohonu.

Nyní k hlubšímu popisu generace PWM procesorem Atmega88. Pro generaci tohoto řídicího signálu je použit osmibitový čítač (který periodicky čítá na námi zvolené frekvenci od nuly až do maximální hodnoty, při jejímž dosažení dojde s dalším hodinovým cyklem k nulování) jehož aktuální hodnota je ukládána v registru $TCNT0$, která se neustále porovnává s komparační hodnotou uloženou ve výše zmiňovaném registru $OCROB$. Pochopení tohoto principu je zřejmé z diagramu průběhů znázorněném na obrázku 3, importovaném z katalogového listu.



Obrázek 3 – princip generování PWM signálu – rychlý mód

Pokud se obsahy obou zmiňovaných registrů rovnají, komparátor zaznamená shodu a nastaví při následujícím hodinovém cyklu příznakový bit $OCFOB$. Na obrázku 4 je znázorněno blokové schéma celého systému.



Obrázek 4 – blokové schéma systému pro generování PWM signálu

Pro zabezpečení správné funkce není doporučeno manuálně zapisovat hodnoty do registru *TCNT0*. Mohl by nastat případ, kdy v běhu programu proběhne zápis nekorektní hodnoty, při kterém by došlo k přeskočení hodnoty, která by znamenala shodu s komparační hodnotou, a tudíž by došlo k vygenerování nesprávného signálu.

V případě daného programu byl zvolen tak zvaný rychlý PWM režim, zvolen nastavením bitů *WGM00* a *WGM01* do jedničky, který je charakteristický následujícími vlastnostmi: těmito hodnotami je zvolen právě rychlý PWM mód, kdy horní hodnota čítače je 0xFF a čítání probíhá od nuly k této hodnotě; aktualizace hodnoty komparačního registru probíhá v hodnotě čítače rovné nule příznak přetečení je nastaven po dosažení maximální hodnoty, tedy 0xFF. Uživatel je dále schopen ovlivňovat frekvenci, a to použitím jak děličky hodinové frekvence procesoru, tak i externího zdroje hodinového signálu. Toto nastavení je realizováno přepínáním bitů *CS00*, *CS01* a *CS02* registru *TCCR0A* podle tabulky 13-9 v katalogovém listu. Pro daný případ je pomocí pokusů zvolena hodinová frekvenci dělená 64mi, při které se regulace chová nejlépe. Procesor umožňuje nastavení řady dalších módů pro generování PWM signálu. Pro navrženou aplikaci byl však nejvhodnější právě režim rychlého PWM.

3.3 Shrnutí

Předchozí výklad byl převážně technického charakteru zaměřen na popis použitých registrů, odůvodnění nastavování příslušných bitů a snahou o přiblížení čtenáři vlastností procesoru Atmega88. Rád bych nyní shrnul funkci programu, vytvořeného pro řízení dopravníku, bez přemíry technických detailů.

Výchozí hodnoty jsou nastaveny jako nulová rychlost posuvu a nulový počet detekovaných objektů. Po zapnutí napájecího napětí proběhne reset procesoru, následně se nastaví povolení přerušování a rozběhne procedura pro generování PWM signálu. Zároveň dojde k inicializaci LCD displeje, na který se následně vypíše uvítací text a spustí se čekací smyčka několikrát za sebou, aby bylo dosaženo potřebného zpoždění. Po úvodní obrazovce následuje na prvním řádku displeje text "Rychlost: 0" a na druhém text "Pocet: 0000", kam se budou během programu zapisovat uživatelem nastavené hodnoty rychlosti a množství napočítaných předmětů. Uživatel může ve vymezeném rozsahu tlačítky regulovat rychlost a umisťovat na pás dopravníku předměty a sledovat správnost jejich počítání i v případě, že jsou těsně za sebou. Tlačítkem reset je realizováno vynulování počtu předmětů na displeji. Díky vlastnostem LCD displeje se provádí nulování pouze této hodnoty, nikoli smazání a následné přepsání celé obrazovky. To je umožněno nastavením kurzoru na libovolné místo na displeji a zápis od této pozice.

3.4 Obsluha LCD displeje

Pro zobrazování dat byl zvolen místo osmi segmentového zobrazovače LCD displej. Jedním z důvodů byl elegantnější vzhled, z hlediska technického směru je považován za jeho výhodu fakt, že data zaslaná na LCD displej není potřeba neustále udržovat (není potřeba použití obnovovacích procesů). To usnadňuje programování a předchází zbytečným chybám.

Pro obsluhu displeje je v programu importován podprogram, ve kterém jsou definovány veškeré funkce spojené s požadavky na zobrazování (jakási knihovna pro zobrazování na LCD). Tyto funkce jsou všeobecně platné pro LCD displeje, volně dostupné na Internetu a v literatuře. Osobně jsem čerpal ze zdrojů získaných v předmětu *Elektronické počítačové systémy*, vyučovaném na Střední průmyslové škole strojní a elektrotechnické v Liberci. Bylo potřeba však změnit zpožďovací hodnoty a přizpůsobit je aplikaci.

Na displej je možné vysílat samostatný znak či text, a to v mezích limitovaných rozměrem (v tomto případě 2 řádky, 16 znaků na řádek). Přístupná je kterákoli pozice zavoláním příslušné funkce s parametry čísla řádku a místa, na který se má kurzor umístit. Smazání lze provést buďto globálně, nebo jednotlivě vysláním prázdného znaku na zvolenou pozici. Kurzor lze zviditelnit, nebo nastavit jeho skrytí.

4. Elektronická část

4.1 Úvodní myšlenka

Při návrhu ovládací jednotky není možné naprogramovat mikroprocesor bez jakéhokoli průběžného testování. Proto od počátku zrodu softwaru bylo potřeba postupně jednotlivé kroky zkoušet na konkrétní aplikaci. Realizace testovacího provozu proběhla na nepájivém poli. Prvním krokem bylo nasimulovat čítání a zobrazovat a zaznamenávání přerušení LED diodou, následovalo doplnění o LCD displej, generování PWM signálu a doplnění tlačítek.

4.2 Testovací obvod na nepájivém poli

Simulace čítání a praktické vyzkoušení přerušovacích procedur nevyžadovalo žádné přídavné součástky a vystačil mi pouze mikrokontrolér a LED dioda. Šlo pouze o aplikaci softwaru na reálný systém, nikoli pouze simulace v debugovacím režimu AVR Studia. Cílem bylo rozsvítit LED diodu a tím signalizovat, že došlo k přerušení v procesoru, podmíněné přetečením čítače. Po softwarovém doplnění LCD displeje bylo potřeba otestovat chování zobrazovací jednotky v praxi. Následovalo zapojení veškerých napájecí a uzemňovací propojek včetně datových vodičů, připravený software byl nahrán do procesoru a odstartován. Bylo potřeba vyřešit řadu problémů se kterými je provoz LCD spojený. Po úvodním odstartování displej nejevil jakékoli známky spolupráce. Podsvětlení se podařilo zprovoznit po odhalení špatného kontaktu v nepájivém poli, nicméně nebyl viditelný kurzor, ani jednotlivé masky znaků. Jako příčina se ukázal špatně zvolený odpor, vedoucí do pinu 12 na LCD zobrazovači (označený jako *RS*), kterým se nastavuje jas. Po zařazení odpovídající odporové kombinace (ta je na závěrečném plošném spoji nahrazena potenciometrem, tudíž jas je plynule regulovatelný) byl displej zprovozněn a připraven k použití.

Následovalo postupné přidávání veškerých komponent nutných k odsimulování správné činnosti aplikace. Dalším problémem bylo řešení tlačítek pomocí vyvedených drátků z nepájivého pole, jimiž se spojením simulovalo stisknutí tlačítka. Při testování touto cestou se aplikace chovala velice chaoticky, při inkrementaci jednoho pulzu (jednoho stisknutí tlačítka) displej zobrazoval například 100 inkrementovaných pulzů, přičemž pokaždé se toto číslo nevypočítavě měnilo. Jako důvod se ukázal právě nevhodně zvolený systém simulace tlačítek. Vyvedené drátky nemají dokonale rovný povrch, proto přestože pro lidské oko je registrovatelné jedno stisknutí, z hlediska průběhu signálu takto zvolené řešení generuje pulzů několikanásobně více.

Posledním krokem k odzkoušení na nepájivém poli byla aplikace spínání motoru pomocí naprogramované PWM modulace, tedy výkonové části. Vzhledem k jednoduchosti tohoto kroku nebyly očekávané žádné problémy a po odstartování aplikace se zapojení chovalo přesně podle očekávání.

Po úspěšném odzkoušení celého obvodu bylo možné přistoupit k nakreslení finálního schématu a následnému vygenerování podkladů pro zhotovení desky plošných spojů.

4.3. Zhotovení desky plošných spojů

Pro náčrt schématu byl využit speciální program pro tyto účely určený, a to OrCAD Capture. Vzhledem k velikosti celého schématu zde není uvedeno jako obrázek (k nahlédnutí je obsažen v příloze). Pro srozumitelnost veškerých prvků je k dispozici tabulka 1 s vysvětlivkami.

Po revizi zapojení proběhl export dat pro OrCAD Layout sloužící již k samotnému rozmístění součástek na desku plošných spojů a propojení příslušných bodů. Tato operace je z hlediska náročnosti velmi relativní. Je důležité součástky korektně rozložit, pak není kreslení samotných cestiček náročnou operací. Pokud ovšem dojde již v počátku k chybnému umístění součástky, může se chyba projevit až ke konci fáze propojování a mnohdy je třeba začít znovu od začátku. Pro danou aplikaci byl zvolen oboustranný plošný spoj a po okraji desky umístěny veškeré konektory, svorkovnice a výkonové prvky (tranzistor a stabilizátory). Díky volbě oboustranné varianty a nahrazením některých součástek jejich SMD variantou, proběhlo zhotovení propojovacích cestiček bez větších problémů. Pro dobrou orientaci při osazování je celá deska doplněna o popisky jednotlivých komponent.

Číslo	Označení	Popis
J5	Napájení 12V	Svorkovnice pro přívodní napájecí kabely
J6	Motor	Svorkovnice pro kabely motoru
J7	IRC	Konektor zapojení inkrementálního čidla
J9	OPTO	Konektor zapojení optické závory
J10	Prog	Konektor pro připojení programátoru procesoru v případě nutnosti úpravy softwaru
J11	RS232	Konektor pro sériovou komunikaci, pokud by byl v budoucnu pro rozšíření aplikace potřeba
J12	Tlacítko	Konektor pro připojení tlačítek
J13	LED	Konektor pro připojení napájení LED diod tlačítek
U3	Atmega88	Patice procesoru Atmega88
U5	78L09/TO92	Stabilizátor 9V
U6	78L05/TO92	Stabilizátor 5V
U7	LCD	Konektor LCD displeje
U8A	7404	Patice logického obvodu 7404

Tabulka 1 – popis schématu

4.4. Osazení a oživení finálního obvodu

Jakmile byl zhotoven konečný tištěný spoj, nastal okamžik osazení jednotlivými součástkami a následné oživení celého obvodu. V tabulce 2 je k dispozici seznam použitých součástek a komponent:

Množství	Reference	Hodnota	Pouzdro
2	C1,C3	0.33uF	C1206
2	C2,C4	0.1uF	C1206
5	C5,C6,C7,C8,C9	100n	C1206
5	C10,C11,C12,C13,C14	470u	E470M/10v
4	D4,D5,D6,D7	1N4007	DIODE
1	Q2	IRF510	TO220
1	RP1	PK50HK005	PK50HK005
2	R17,R18	1k0	R1206
1	R19	RR 3X1K	Odporová síť
1	R20	RR 3X1K	Odporová síť
1	R21	RR 3X330R	Odporová síť
1	U3	Precizní patice	DIP28
1	U5	L78L09	TO220
1	U6	L78L05	TO220
1	U8	Precizní patice	DIP14
3	Chladič	v7110	V7110

Tabulka 2 – seznam součástek

Oživení neprobíhá kompletním osazením celého kompletu, nýbrž postupným přidáváním rizikových prvků. Proto byl obvod nejprve osazen součástkami zajišťující elektronickou roli, tedy vynechán procesor a logický obvod. Důvod tohoto kroku byl, aby mohla být nejprve prověřena správnost napájení na jednotlivých pinech a nedopatřením tyto obvody nebyly nezničeny. Hned při úvodním zapnutí a ověřování přítomnosti správných hodnot napětí se kázalo, že stabilizátory mají opačné číslování vývodů, tudíž byly zapojeny obráceně a nevykonávali požadovanou funkci. Po napravení této závady následovalo doplnění o logický obvod 7404, které také neproběhlo zcela bez problémů. Při přeměřování odběru proudu se hodnota zdála příliš vysoká a IO se začal nezvykle ohřívat. Po řadě krocích vyšlo najevo, že jeden z pinů patice pro tento obvod je nekorektně zapájen a kontakt není stoprocentní. Jednoduchým přepájením byla chyba odstraněna a bylo možno přejít k dalšímu kroku, implementaci samotného procesoru. Po finálním přeměření se osazená deska (obrázek 5) chovala podle očekávání a stačilo již doplnit displej, tlačítka a senzory. Jakmile byly všechny sběrnice a kabely nakonektorovány, následovalo kompletní zapojení a odzkoušení celého přípravku.



Obrázek 5 – osazená DPS

Vzhledem k průběžnému testování a řešení veškerých vzniklých problémů se celá sestava v závěrečné fázi chovala zcela podle předpokladů.

5. Senzory

Aby bylo možné splnit zadání, byla třeba uvažovat nejen nutnost přítomnosti prvku realizujícího optickou závoru pro detekci přítomnosti objektu, ale také snímat rychlost pohybu motorku.

5.1 Senzor rychlosti motorku

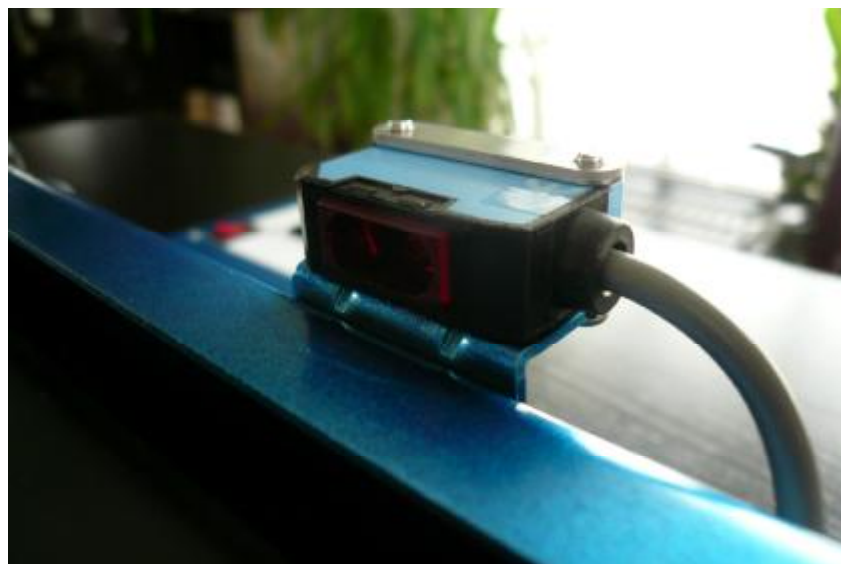
Snímání rychlosti otáčení motorku bylo nutné k zajištění splnění požadavku proměnné rychlosti posuvu dopravníku. Použit bylo možné inkrementálního optického snímače OMRON E6A2-CWZ3 (obrázek 6) s rozlišovací schopností 200 pulzů na otáčku. Jeho katalogový list je rovněž obsažen v příloze (příloha 2) pro detailnější informace, co ze základních týče, jedná se o prvek napájený stejnosměrným napětím o hodnotě 5 - 12V. Disponuje dvěma napájecími vodiči (+5V až +12V - hnědá, GND - modrá), dále pak třemi datovými vývody. Dva z vodičů reprezentují přesně počet pulzů načítaných během otáčení, navzájem oproti sobě fázově posunutých z důvodu rozpoznání směru otáčení v případě nutnosti, dále vodič, na kterém je generován pulz vždy při dokončení jedné celé otáčky. Tato varianta by pro naši úlohu byla plně dostačující, jelikož se nejedná o aplikaci vyžadující dokonalou přesnost, nicméně je použita přesnější varianta pro lepší demonstraci vlastností inkrementálního čidla.



Obrázek 6 – uchycení inkrementálního čidla na přípravku

5.2 Senzor přítomnosti objektu

Detekce přítomnosti předmětu v určeném místě na dopravníku se dala zrealizovat několika způsoby. Nejjednodušším řešením by pravděpodobně bylo použití foto prvku. Na jednu ze stran dopravníku umístit například svítící LED diodu, na druhou stranu foto diodu. Tato metoda byla na poskytnutém přípravku původně použita, nicméně pro účely bakalářské práce jsem zvolil elegantnější řešení, a to použití optického čidla SICK Optex WT150-P132 (obrázek 7). Katalogový list pro konkrétní model je přílohou práce (Příloha 3), nicméně jelikož dohledání kvalitní dokumentace v anglickém nebo českém jazyce nebylo úspěšné, je k dispozici pouze originál v jazyce Francouzském. Senzor je napájen stejnosměrným napětím 10 – 30V, jehož výstupní signál dosahuje také úrovně deseti voltů, což působilo v mé úloze drobné komplikace, jelikož veškeré další zařízení pracovalo užívalo napětí 5V. Minimální spínací vzdálenost odpovídá 1cm a čidlo je schopno detekovat objekty až do vzdálenosti 20cm, záleží pochopitelně na odrazivosti povrchu snímaného předmětu. Dosah čidla je plynule regulovatelný potenciometrem umístěným na vrcholu snímače.



Obrázek 7 – uchycení optické závory na dopravníku

Vzhledem k opotřebení a původnímu stavu dopravníku, který jsem měl v úloze použít bylo třeba provést drobnou renovaci. Dopravníkový pás byl zpuchřelý a vytahaný. Po několika otáčkách motoru se začal navíjet po stranách na hnací soukolí. Nahradil jsem ho tedy pásem novým, pro jehož použití jsem zvolil gumovou upevňovací lištu pro usazování linolea. Kritickým bodem je slepení obou konců. Jako nejlepší řešení se ukázalo použít speciální vulkanizační lepidlo, které se standardně prodává v sadách pro opravu duší kol. Pro celkový dobrý dojem z dopravníkového systému byla celá souprava přelakována a usazena na připravený podstavec. Následovalo vrtání děr pro různou řadu úchytů, otvorů pro průsmyk kabeláže a podobně. Před samotným zabudováním řídicí jednotky do útrobu úlohy bylo třeba nejprve osadit ovládací panel. Bylo nutné na speciální fólii vytisknout popisky tlačítek a tu následně nalepit na předem připravený hliníkový plech. Montáž hlavního vypínače a LCD displeje nebyla komplikovaná, problémy začaly s ovládacími tlačítky pro rychlost a reset počtu načítaných objektů. Tyto tlačítka totiž nemají žádné ukotvovací body, jsou určena na montáž přímo do desek plošných spojů. Zvolil jsem je ze dvou důvodů. Mají dobré vlastnosti odezvy, nevznikají na nich falešné zákmity a také díky modrému podsvícení dobře vypadají. Pro jejich umístění byla zhotovena plastová lišta, uchycená šrouby k panelu, na kterou dobře přilne tavné lepidlo, a díky jeho nevodivé vlastnosti bylo možné tlačítka přilepit vysoce odolným lepidlem, aby nedošlo k jejich promáčknutí skrz panel. V případě nutnosti výměny tlačítka se dá každé zvlášť jednoduše odříznout a vyměnit. Výsledný design ovládacího panelu je zobrazen na obrázku 9.



Obrázek 9 – konečná podoba ovládacího panelu

Úchyt optického senzoru jsem jemně modifikoval, nicméně v zásadě byl použit držák dodávaný se senzorem. Pro inkrementální čidlo otáček bylo nutné realizovat řadu vedlejších prací. V první řadě jsem si na katedře nechal vyhotovit novou osičku motorku, kterou bylo nutné vyvést skrze stěnu dopravníku ven, aby ji bylo možné přes spojku upevnit k čidlu. Dále byl vyroben držák na samotný senzor, který zajišťoval perfektní polohovou stabilitu čidla, nutnou pro správnou funkci.

Posledním krokem bylo ukotvení řídicí jednotky na distanční sloupky do nitra podstavce. Nakonec se ukázalo jako nezbytné vybavit celý obvod chladícím ventilátorem. Výkonové prvky vykazovaly sice přiměřený ohřev, nicméně v uzavřeném prostoru, ve kterém se celý obvod nachází, jsem volil přídatné chlazení z důvodu zajištění stoprocentní spolehlivosti a tepelné nezávislosti.

Po řadě drobných dokončovací a začistovací prací je celý výsledek vyobrazen na obrázku 10, kde je k dispozici náhled na uspořádání úlohy, a na obrázku 11, kde je k nahlédnutí spodní strana podstavce, kde se nachází veškerá elektronika a propojovací kabely a sběrnice.



Obrázek 10 – závěrečná podoba přípravku



Obrázek 11 – pohled zespodu

7. Závěr

Problém, který se může při užívání úlohy objevit, je zmatené chování LCD displeje po zapnutí. Tato situace může nastat při vypnutí a urychleném zapnutí. Pro inicializaci LCD displeje je potřeba, aby všechny kondenzátory byly vybity (to plyne z konstrukčních vlastností LCD zobrazovačů a samozřejmě na řídicím softwaru). V případě, pokud se nestačí vybit, můžou nastat komplikace se zobrazovací jednotkou. Čas potřebný k vybití se pohybuje v desetinách sekundy, proto tato situace může nastat v opravdu rychlém přepnutí napájecího spínače. Při běžném užívání nikoli. V této situaci stačí přípravek vypnout, chvíli počkat a opět spustit. Vše bude v pořádku.

Cílem bylo sestrojít úlohu vhodnou pro studenty, na které by si byli schopni prakticky vyzkoušet funkce jednotlivých senzorů, kterými je přípravek vybaven. Ovládání je velmi intuitivní a celá úloha je vybavena veškerými potřebnými ochranami proti přepětí, díky stabilizátorům napětí, a přepólováním pomocí ochranných diod. Z důvodu negativního vlivu a vzniku problémů při inicializaci LCD displeje není do ochranných prvků zařazena proudová pojistka. Z tohoto důvodu je nutné dbát na dodržení napájecího napětí 12 V.

Studenti si mohou vyzkoušet chování především optické závory, a to například studovat spínací vzdálenost předmětů s různě odrazivým povrchem, nebo nastudovat rozptyl snímání z katalogového listu a prakticky jej ověřit. V případě potřeby se dají velmi jednoduše vyvést na horní část přípravku piny nesoucí signál z jednotlivých senzorů a pozorovat je na osciloskopu.

Doufám, že tento laboratorní přípravek přinese cílové skupině lepší pochopení v oblasti automatizační techniky, a díky názornému vyzkoušení bude všem uživatelům tato problematika bližší. Osobně jsem se díky bakalářské práci hlouběji seznámil především s programováním mikroprocesorů a návrhem elektronických obvodů. Největším přínosem však hodnotím nutnost flexibilního řešení nečekaných problémů, které člověka provázejí celý život.

8. Přílohy

8.1 Příloha 1

Celkový katalogový list o rozsahu 374 stránek je obsažen na přiloženém CD.

Features

- High Performance, Low Power AVR[®] 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- Non-volatile Program and Data Memories
 - 4/8/16K Bytes of In-System Self-Programmable Flash (ATmega48/88/168)
Endurance: 10,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
In-System Programming by On-chip Boot Program
True Read-While-Write Operation
 - 256/512/512 Bytes EEPROM (ATmega48/88/168)
Endurance: 100,000 Write/Erase Cycles
 - 512/1K/1K Byte Internal SRAM (ATmega48/88/168)
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TOFP and QFN/MLF package
 - 6-channel 10-bit ADC in PDIP Package
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TOFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.6 - 5.5V for ATmega48V/68V/168V
 - 2.7 - 5.5V for ATmega48/88/168
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - ATmega48V/88V/168V: 0 - 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V
 - ATmega48/88/168: 0 - 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
- Low Power Consumption
 - Active Mode:
 - 250 µA at 1 MHz, 1.8V
 - 15 µA at 32 kHz, 1.8V (including Oscillator)
 - Power-down Mode:
 - 0.1µA at 1.8V



8-bit AVR[®]
Microcontroller
with 8K Bytes
In-System
Programmable
Flash

ATmega48/V
ATmega88/V *
ATmega168/V *

* Preliminary

Rev. 2045R-AVR-04/07



8.2 Příloha 2

OMRON

Incremental Rotary Encoder

E6A2

Miniature Rotary Encoder for Positioning in Space-Confined Areas

- Wide variety of supply voltages and output forms to match input devices
- Models with zero index function ideal for positioning applications
- High resolution models (300 or 360 pulses per revolution) substantially improve measuring accuracy
- High response frequency and noise immunity make encoders ideal for factory automation applications



CE

Ordering Information

ENCODERS

When ordering, add the resolution (pulses per revolution) between the part number and cable length. For example, E6A2-CWZ3E 200 P/R 0.5M.

Resolution (pulses per revolution)	Output phases	Output form	Supply voltage	Part number
10, 60, 100, 200, 300, 360	A	Voltage	5 to 12 VDC	E6A2-CS3E □□□P/R 0.5M
		Open collector	5 to 12 VDC	E6A2-CS3C □□□P/R 0.5M
		Open collector	12 to 24 VDC	E6A2-CS5C □□□P/R 0.5M
100, 200	A, B	Voltage	5 to 12 VDC	E6A2-CW3E □□□P/R 0.5M
		Open collector	5 to 12 VDC	E6A2-CW3C □□□P/R 0.5M
		Open collector	12 to 24 VDC	E6A2-CW5C □□□P/R 0.5M
100, 200	A, B, Z (zero)	Voltage	5 to 12 VDC	E6A2-CWZ3E □□□P/R 0.5M
		Open collector	5 to 12 VDC	E6A2-CWZ3C □□□P/R 0.5M

REPLACEMENT PARTS

Description	Part number
Shaft coupler (supplied with each encoder)	E69-C04B
Mounting bracket (supplied with E6A2-CWZ encoders)	E69-1

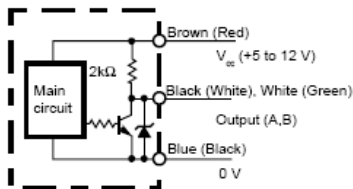
Specifications

Part number	E6A2- CS3E	E6A2- CW3E	E6A2- CWZ3E	E6A2- CS3C	E6A2- CW3C	E6A2- CWZ3C	E6A2- CS5C	E6A2- CW5C
Supply voltage	5 VDC -5% to 12 VDC +10%; max. 5% ripple peak-to-peak						12 VDC -10% to 24 VDC +15%; max. 5% ripple	
Current consumption	30 mA max.		50 mA max.	20 mA max.		30 mA max.	20 mA max.	
Resolution (pulses per revolution)	10, 60, 100, 200, 300, 360	100, 200	100, 200	10, 60, 100, 200, 300, 360	100, 200	100, 200	10, 60, 100, 200, 300, 360	100, 200
Output phases	A	A, B	A, B, Z	A	A, B	A, B, Z	A	A, B
Output form	Voltage output			Open collector output			Open collector output	
Output capacity	Output resistance: 2 k Ω Residual voltage: 0.4 V max. Sink current: 20 mA max.			Applied voltage: 30 VDC max. Residual voltage: 0.4 V max. Sink current: 30 mA max.			Applied voltage: 30 VDC Residual voltage: 0.4 V Sink current: 30 mA max.	
Maximum response frequency	30 kHz	20 kHz	20 kHz	30 kHz	20 kHz	20 kHz	30 kHz	20 kHz
Rotation direction	Reversible, CW + CCW			Reversible, CW + CCW			Reversible, CW + CCW	
Phase difference of output	—	90° \pm 45°	90° \pm 45°	—	90° \pm 45°	90° \pm 45°	—	90° \pm 45°
Output rise and fall times	1.0 μ s max. (at sink current of 10 mA with 2 m cable)			1.0 μ s max. (at control output voltage of 5 V and load resistance of 1 k Ω with 2 m cable)				
Starting torque	10 g-cm (0.14 oz.-inch) max.							
Shaft loading	Radial	1 kgf (7.2 ft•lbs)						
	Axial	0.5 kgf (3.6 ft•lbs)						
Moment of inertia	1 g-cm ² (0.0055 oz-inch ²)							
Maximum rpm	5,000 rpm							
Electrical connection	Prewired with 0.5 m (1.64 ft) length cable							
Weight	Approx. 35 g (1.2 oz)							
Enclosure rating	IEC: IP50							
Ambient temperature	Operating Storage	-10°C to 55°C (14°F to 131°F)						
		-25°C to 80°C (-13°F to 176°F)						
Ambient humidity	35% to 85% RH							
Vibration resistance	Mechanical durability: 10 to 55 Hz, 1.5 mm double amplitude, in X, Y, and Z directions for 2 hours each							
Shock resistance	Mechanical durability: 500 m/s ² (approx. 50 G) in X, Y, and Z directions, 3 times each							
Insulation resistance	10 M Ω minimum at 500 VDC between current-carrying part and housing							
Dielectric strength	500 VAC, 50/60 Hz for 1 minute between current-carrying part and housing							

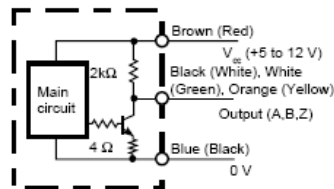
Operation

■ OUTPUT CIRCUIT DIAGRAMS

Voltage Output
E6A2-CS3E, E6A2-CW3E



E6A2-CWZ3E

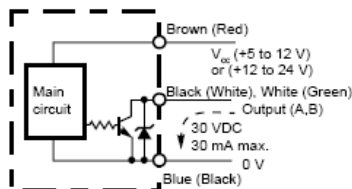


Wire Color Code

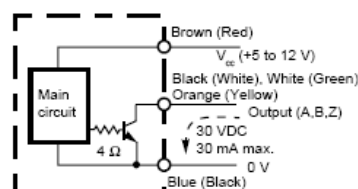
Note: IEC colors shown first.

Wire color	Signal
Brown (Red)	V_{CC}
Black (White)	A
White (Green)	B
Orange (Yellow)	Z
Blue (Black)	0 V (common)

Open Collector Output
E6A2-CS□□C, E6A2-CW□□C



E6A2-CWZ3C

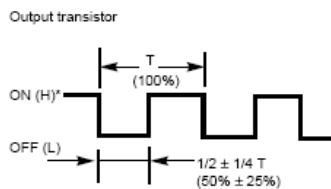


Note:

1. The white (green) and orange (yellow) lines of the single type (E6A2-CS) do not output signals (no connection).
2. The orange (yellow) line of the reversible type (E6A2-CW) does not output signal (no connection).
3. The voltage output type is capable of sinking a maximum of 20 mA.

■ TIMING CHARTS

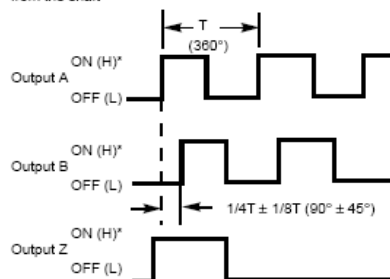
E6A2-CS



E6A2-CW, E6A2-CWZ

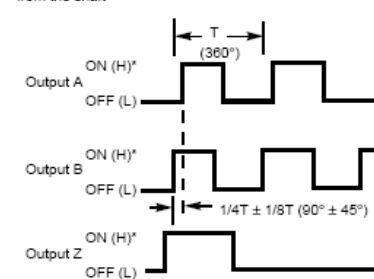
Direction of rotation: CW
Clockwise as viewed
from the shaft

→ CW direction



Direction of rotation: CCW
Counterclockwise as viewed
from the shaft

→ CCW direction



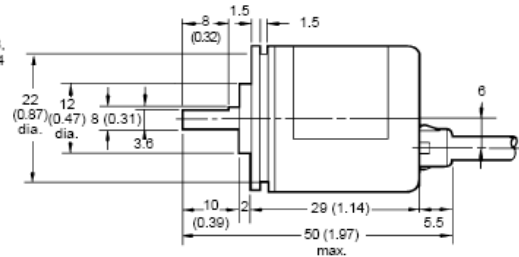
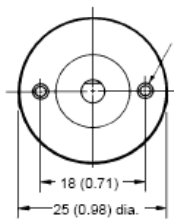
Note:

1. *(H) and (L) indicate the output levels of the voltage output type.
2. Output A leads B by $1/4T \pm 1/8T$ when the shaft revolves clockwise. Output A lags behind B by $1/4T \pm 1/8T$ when the shaft revolves counterclockwise.

Dimensions

Unit: mm (inch)

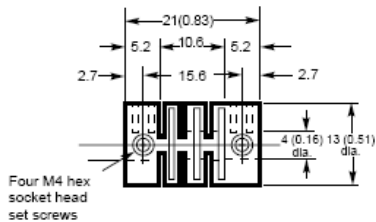
ENCODERS



*Output cable (shielded) O.D.: 4 dia.
Standard length: 50 cm (1.64 ft)

ACCESSORIES

Shaft Coupler E69-C04B

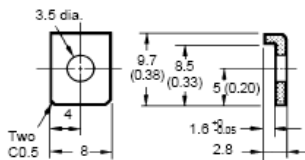


Note:

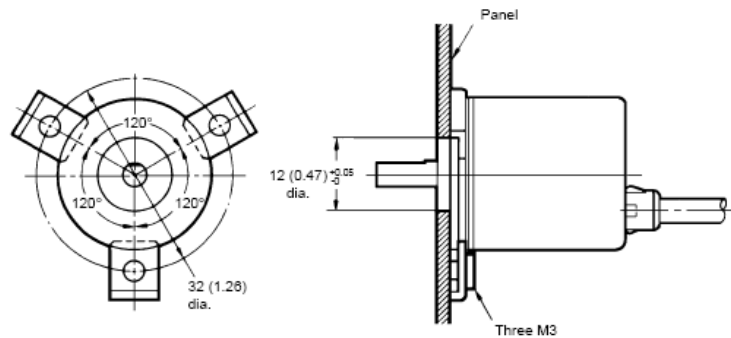
1. Material: Glass-filled polybutadiene terephthalate (PBT).
2. A coupler is supplied with each E6A2 encoder.
3. Each set screw must be tightened to 2.5 kg-cm (2.17 in-lbs)

Mounting Bracket E69-1

supplied with E6A2-CWZ encoders



Dimensions with Encoder



OMRON

OMRON ELECTRONICS LLC

One East Commerce Drive
Schaumburg, IL 60173
1-800-55-OMRON

Cat. No. CEDSAX4

11/01

OMRON ON-LINE

Global - <http://www.omron.com>
USA - <http://www.omron.com/oei>
Canada - <http://www.omron.com/oci>

Specifications subject to change without notice.

OMRON CANADA, INC.

885 Milner Avenue
Scarborough, Ontario M1B 5V8
416-286-6465

Printed in the U.S.A.

8.3 Příloha 3



Portée

10 ... 200 mm
Type : énergétique

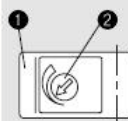
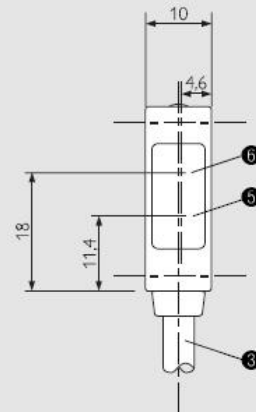
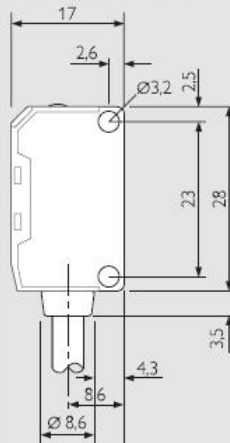


Caractéristiques:

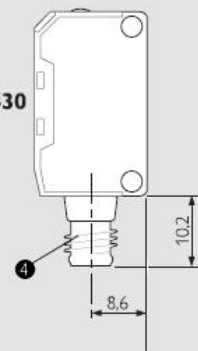
- Capteur miniature avec ampli intégré
- Forme miniature idéale lors d'un problème de place
- Détecteur énergétique
 - pour applications standard
 - pour détection de contrastes simples
- Point de détection réglable par potentiomètre
- Alimentation CC 10 ... 30 V
- Câble de raccordement 2 m ou Connecteur M 8-4 pol.
- /

WT 150

WT150-P/-N132



-P/-N 430

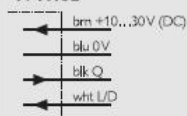


- 1 Diode témoin rouge : lumière reçue \geq seuil de détection
- 2 Potentiomètre réglage sensibilité 270°
- 3 Câble de raccordement
- 4 Connecteur M 8-4 pôles
- 5 Axe optique émetteur
- 6 Axe optique récepteur

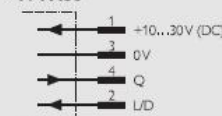
Câble surmoulé avec prise M 8-4 pol. voir accessoires
Équerre BEF-W 150-A comprise dans livraison,
Option : équerre BEF-W 150-B voir accessoires

Schéma de raccordement

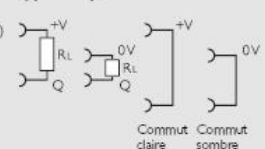
WT150 -P/-N132



-P/-N430



-N -P



L/D=Entréeclair/sombre

brn	blu	blk	wht
brown	blue	black	white
brun	bleu	noir	blanc

Réserve de fonctionnement

WT 150 énergétique

Seuil de commutation

Portée (mm)

SICK OPTIC-ELECTRONIC

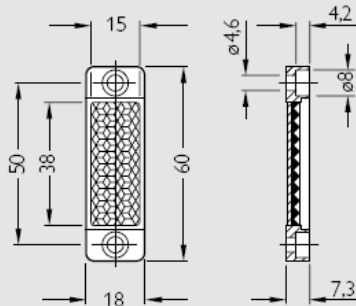
Détecteur réflex WT 150 Type : énergétique

WT 150	-P	-N
Connectique	Câble de raccordement 2 m	
Type	WT 150-P 132	WT 150-N 132
Référence	6 011 042	6 011 039
Connectique	Connecteur M 8-4 pol.	
Type	WT 150-P 430	WT 150-N 430
Référence	6 011 044	6 011 041
Distance de détection (réglable)	10 ... 200 mm, objet de réémission 90% (d'après DIN 50330)	
Diamètre spot	env. 20 mm à 200 mm de distance	
Angle d'ouverture	env. 6°	
Réglage de sensibilité	Potentiomètre, 270°	
Émetteur	LED, lumière rouge visible	
Alimentation U_a	CC 10 ... 30 V ¹⁾	
Consommation (sans charge)	max. 20 mA	
Type de commutation	Clair/Sombre par entrée extérieure ²⁾	
Sorties collecteur ouvert	PNP	NPN
Tension HIGH	U _a - (≤ 1,5 V)	env. U _a
Tension LOW	env. 0 V	≤ 1,5 V
Courant de sortie max.	100 mA	
Temps de réponse (max.); fréquence (min.)	0,5 ms; 1000/s	
Classe de protection VDE	III	
Indice de protection(IEC 144)	IP 67 (IEC 144)	
Protections électriques ³⁾	A, B, C, D	
Température d'utilisation T _U	- 25 ... + 55 °C	
Température de stockage T _L	- 40 ... + 70 °C	
Connecteur	M 8-4 pol.	
Câble de raccordement	PVC, 2 m, 4 × 0,18 mm ² , ø 3,5 mm	
Poids	avec connecteur avec câble 2 m	
	7 g	44 g

1) Valeurs limites, Ondulation max. ± 10 %
2) Entrée ouverte
NPN: commutation claire
PNP: commutation sombre
Entré au + V: commutation claire
Entrée au 0 V: commutation sombre

3) A = inversions polarité alimentation
B = inversions polarité entrées sorties
C = insensibilité impulsions parasites
D = sorties protégées contre courts-circuits et surintensités

Accessoires



Réflecteur PL 20 A
Référence 1 012 719
Livré avec la WL 150 .

Équerre BEF-W 150-A
Référence 5 306 217
Face de montage horizontale, pour W 150, incluse dans livraison, avec contre plaque et 2 vis de fixation M 3 x 15 mm

Équerre BEF-W 150-B
Référence. 5 306 279
Montage vertical, pour W 150, comprenant : contre plaque et 2 vis de fixation M 3 x 15 mm.
A commander séparément

Diaphragmes BL-150-10 Référence 5 306 216

Contenu : 3 paires avec largeurs de fente A, B, C.
Montage par autocollant au dos du diaphragme sur les optiques des VS150 et WE150. Utilisés pour détecter des objets fins ou augmenter la précision de positionnement.

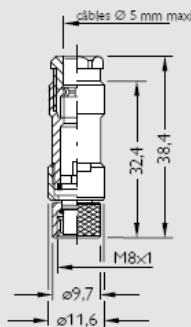
Modifications des portées:

- A) Diaphragme 0,5 mm: Portée= 0,5 m
- B) Diaphragme 1,0 mm: Portée= 1,0 m
- C) Diaphragme 2,0 mm: Portée= 2,0 m

Prises et connecteurs: M 8-4 pol. (suivant EN 50044)
à souder pour câbles de \varnothing 3,5 mm ... 5 mm.

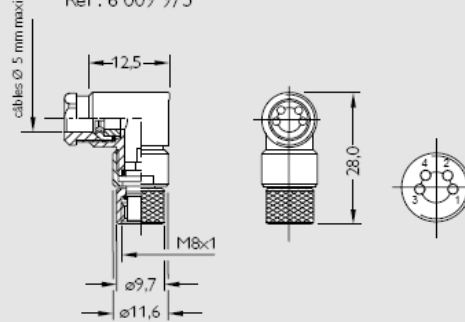
Câbles surmoulés avec prise M8-4pôles de longueur 2 m, 5 m et 10 m

M 8-4 pol. droit
Ref. 6 009 974



câble avec prise M 8-4 pol. droite,
longueur :
2 m, Ref N° 6 009 870
5 m, Ref N° 6 009 872
10 m, Ref N° 6 010 754

M 8-4 pol. coudé
Ref. 6 009 975



câble avec prise M 8-4 pol. coudée,
longueur
2 m, Ref N° 6 009 871
5 m, Ref N° 6 009 873
10 m, Ref N° 6 010 755

Brochage:
Pin 1 = brun
Pin 2 = blanc
Pin 3 = bleu
Pin 4 = noir

GXF • 7 008 512 / . 0697 • Nous nous réservons le droit de procéder à des modifications

SICK

optic
electronic

SIÈGE ET SAV

BP 42
F-77312 Mame la vallée
Cedex 2
☎ : 01 64 62 35 00
Fax : 01 64 62 35 77
www.sick.de

LYON

Parc Club du Moulin à vent
33 Rue Georges Levy
F-69693 Venissieux Cedex
☎ : 04 78 00 06 15
Fax: 04 78 00 47 37

NANTES

Parc Club du Perray
4 Rue de la Rainière
BP 93 901
F-44 439 Nantes cedex 03
☎ : 02 40 50 00 55
Fax: 02 40 52 13 88

PARIS

BP 42
F - 77 312 Mame la vallée
Cedex 2
☎ : 01 64 62 35 00
Fax: 01 64 62 35 77

SICK OPTIC-ELECTRONIC

8.4 Příloha 4

```
// seznam definic jednotlivych registru
```

```
#include <avr/io.h>
```

```
#include <avr/iom88.h>
```

```
#include <avr/interrupt.h>
```

```
#include "lcd.h"
```

```
#define KOSTKA 13000 // pocet pulsu (delka impulsu) jednoho predmetu  
// napevno definovana promenna, pri zmene predmetu  
// nutno zmenit
```

```
const char InfoText[] = "CVUT Praha"; // uvitaci text na LCD displeji  
volatile unsigned int pulzy = 0; // definovani promenne pro citani pulzu IRC  
unsigned int pocetkostekCOMPL = 0; // definovani promenne pro celkovy pocet kostek  
unsigned char upgLCD = 0; // pomocna promenna pro upgrade displeje
```

```
ISR (INT1_vect) //odchyceni vektoru preruseni inkrementalniho cidla  
{  
    pulzy +=1; // inkrementace pulzu z inkrementalniho cidla  
    if (pulzy >= KOSTKA) // porovnavani, v pripade ze je nacistano stejne pulzu  
    { // jako je v promenne KOSTKA, dojde k odedcteni,  
        // inkrementaci  
        upgLCD = 1; // poctu kostek a upgradu displeje  
        pulzy -= KOSTKA;  
        ++pocetkostekCOMPL;  
    }  
}
```

```
ISR (INT0_vect) //odchytavani vektoru preruseni pro optickou zavoru  
{  
    if ( (EICRA & ((1<<ISC00) | (1<<ISC01))) == ((1<<ISC01)) // kdyz je sestupna  
    {  
        EICRA &= ~((1<<ISC00) | (1<<ISC01));  
        EICRA |= (1<<ISC00) | (1<<ISC01); // nabezna hrana  
        EIMSK |= (1 << INT1); // povoleni preruseni od INT1  
        pulzy = 0; // nulovani promenne  
    }  
    else // reakze na nabeznu hranu  
    {  
        EICRA &= ~((1<<ISC00) | (1<<ISC01));  
        EICRA |= (1<<ISC01); // nastaveni reakce na sestupnou hranu  
        EIMSK &= ~(1 << INT1); // zakazani preruseni INT1  
    }  
}
```

```
void preruseni(void) // uvodni nastaveni podminek pro povoleni preruseni  
{
```



```

    EICRA = (1<<ISC10) | (1<<ISC01);
    EIMSK = (1 << INT0);
    SREG = (1<<SREG_I);
}

void cekej(int x)      // cekaci funkce pro zpozdeni pri aktivaci tlacitek
{                    // a pro inicializaci LCD displeje

    while (--x);

}

void prepocet( unsigned int val)      // funkce slouzici k prepocetu hodnoty zasobniku
{                                    // na hodnotu zobrazitelnou na displeji
    unsigned int pom = 10000;        // pracuje na principu postupneho deleni
    unsigned char i,j;               // a vyslani prislusneho znaku na LCD

    for( i=0 ; i < 5 ; i++)
    {
        j = val/pom;
        if (j == 0) LCD_znak('0');
        else LCD_znak('0'+j);
        val -= j*pom;
        pom /= 10;
    }

}

void obsluha1reset(void)      //funkce slouzici pro resetovani poctu kostek na LCD
{

    LCDupdate();
    pocetkostekCOMPL = 0;
}

void obsluha2plus(void)      // funkce pro zrychlovani dopravniku
{                            // pracuje s registrem OCR0B v rozsahu 0 - 255
                            // kdy 0 = max; 255 = min rychlost
    if (OCR0B < 155) OCR0B += 31;
    else if (OCR0B == 155) OCR0B = 255;
}

void obsluha3minus(void)    // funkce pro zpomalovani dopravniku
{
    if (OCR0B == 255) OCR0B = 155;
    else if (OCR0B >= 31) OCR0B -= 31;
}

void pwm(void)              // funkce slouzici pro generovani PWM signalu
{                            // pro pochopeni nutno nastudovat kapitolu 13 z katalogoveho listu
}

```

```

OCR0B = 0xFF;
PORTD |= (1<<PD5);
DDRD |= (1<<DDD5);          // OC0B

TCCR0A = (1<<COM0B1) | (1<<WGM01) | (1<<WGM00);
TCCR0B = (1<<CS01) | (1<<CS00);

}

void LCDupdate()           // upgrade hodnot na displeji
{

/*
Tato funkce slouzi k prepsavani hodnot na LCD. Jelikož při rozdělení
zasobníku OCR0B na pět rovnoměrných dílů motorek pro první rychlost nezvládal
dopravník rozpořehovat, bylo nutné funkci modifikovat a pro první rychlost
zvolit nelineární skok.
*/

    unsigned char pom = OCR0B;
    if(pom == 255) pom = 6;
    else pom /= 31;

    LCD_row1(10);
    LCD_znak('6' - pom);
    LCD_row2(7);
    prepocet(pocetkostekCOMPL);
    upgLCD = 0;
}

unsigned char tlacitka()
{
/*
Definice tlačítek. Každému tlačítku je přiřazen určitý výstupní pin.
Tlačítka jsou tedy definována na piny PB0, PD6 a PD7.
Pro zajištění správné funkce jsou navíc aktivní pull-up rezistory
na všech zvolených pinech.
*/

    unsigned char pom =0;

    DDRB &= ~(1<<DDB0);          //nastavení vstupních pinů - tlačítek
    DDRD &= ~(1<<DDD6);
    DDRD &= ~(1<<DDD7);

    PORTB |= (1<<PB0);           // nastavení pull-upu
    PORTD |= (1<<PD6);
    PORTD |= (1<<PD7);
}

```

```

/*
Tato cast zjistuje konfiguraci prave aktivniho pinu. Podle toho, se kterym je
pracovano (ktere tlacitko bylo zmacknuto), nastavi pomocnou promennou pom a
vykona se prislusna rutina.
*/
    if ((PINB & (1<<PB0)) != (1<<PB0)) pom = 1;
    else if ((PIND & (1<<PD6)) != (1<<PD6)) pom = 2;
    else if ((PIND & (1<<PD7)) != (1<<PD7)) pom = 3;

    return pom;
}

int main(void)
{
    unsigned char pom = 0;

    preruseni();           // volani funkce preruseni
    pwm();                 // volani funkce pwm
    LCD_ini();             // inicializace LCD displeje
    cekej(50000);          // cekaci cyklus (s parametrem 50000) pro spravnou inicializaci LCD
    LCD_row1(3);           // pozice kurzoru na LCD
    LCD_data(&InfoText);  // vypis info textu "CVUT Praha"
    cekej(65000);          // cekaci cyklus s parametrem 65000
    cekej(65000);          // nutno nekolikrat za sebou pro korektni
    cekej(65000);          // zobrazeni a citelnosti napisu
    cekej(65000);

    LCD_clr ();            // mazani LCD
    LCD_data("Rychlost: 0"); // zobrazeni rychlosti na radku 1
    LCD_row2(0);           // nastaveni kurzoru na druhy radek
    LCD_data("Pocet: ");  // zobrazeni poctu kosticek na druhem radku
    prepopet(0);           // volani funkce prepopet

    while(1)
    {
        pom = tlacitka();
        switch (pom)
        {
            case 1: obsluha1reset(); // obsluha pri zmacknuti RESET
                    LCDupdate();
                    cekej(60000);
                    break;

            case 2: obsluha2plus(); // obsluha pri zmacknuti SPEED+
                    LCDupdate();
                    cekej(60000);
                    break;

            case 3: obsluha3minus(); // obsluha pri zmacknuti SPEED-
                    LCDupdate();
                    cekej(60000);
                    break;

            default: break;
        }
        if(upgLCD == 1)LCDupdate(); // pokud je nastavena promenna pro obnoveni
        // LCD, obmena se provede.
    };
}

```

9. Použitá literatura

- [1] BURKHARD, Mann: *C pro mikrokontroléry*; vydané nakladatelstvím BEN v Praze, 2004

- [2] MATOUŠEK, David: *Práce s mikrokontroléry Atmel AVR – Atmega16*; vydané nakladatelstvím BEN v Praze, 2006

- [3] Katalogový list *Atmega 88*, dostupný z WWW: www.atmel.com

- [4] Katalogový list senzoru *SICK Optex WT150-P132*, dostupný z WWW:
<http://docs-europe.electrocomponents.com/>

- [5] Katalogový list senzoru *OMRON E6A2-CWZ3*, dostupný z WWW:
<http://oeiwcsnts1.omron.com>

- [6] Katalogové listy součástek dostupné po přihlášení z WWW, na stránkách GM Electronic, Křižíkova 77, Praha: www.gme.cz