



## **Středoškolská technika 2010**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

### **Rotující LED zobrazovač**

**Jiří Slíž**

**Střední průmyslová škola elektrotechniky a informatiky, Ostrava, příspěvková organizace  
Kratochvílova 7/1490, Ostrava - Moravská Ostrava**

# STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

---

Obor SOČ: 10. Elektrotechnika, elektronika a telekomunikace

**Rotující LED zobrazovač**

**Rotating LED display**

Autor: Jiří Sliž

Škola: Střední průmyslová škola elektrotechniky a  
informatiky,  
Ostrava, Kratochvílova 7/1490,  
702 00 Ostrava - Moravská Ostrava

**Opava 2010**

Prohlašuji tímto, že jsem soutěžní práci vypracoval samostatně a uvedl v seznamu literatury veškerou použitou literaturu a další informační zdroje včetně internetu.

Ve Opavě dne 10.3.09

---

vlastnoruční podpis autora

### Obsah

Anotace .....	4
1. Princip.....	5
1.1. Princip setrvačnosti oka .....	5
1.2. Princip kruhového LED zobrazovače .....	5
1.3. Snímání polohy.....	5
2. Kruhový rotační LED zobrazovač .....	6
2.1. Technická konstrukce .....	6
2.2. Napájení a regulace otáček motoru .....	7
2.3. Schéma a DPS .....	8
2.4. Popis mikropočítače PIC16F874A.....	9
2.4.1. Blokové schéma mikropočítače PIC16F874A .....	9
2.4.2. Rozvržení paměti programu .....	10
2.4.3. Speciální funkční registry (SFR) .....	11
2.4.4. I/O porty .....	12
2.4.5. Přerušování .....	12
3. IR senzor a enkodér.....	13
3.1. Monochromatické IR čtení barvy .....	13
3.2. Enkodér .....	13
4. Kulový rotační LED zobrazovač.....	14
4.1. Schémata zapojení .....	15
4.1.1. Blokové schéma.....	15
4.1.2. Schéma zapojení řídicí jednotky.....	16
4.1.3. Schéma zapojení serio-paralelního převodníku.....	17
4.1.4. IR snímače.....	17
4.2. Konstrukce.....	18
4.2.1. Technická konstrukce .....	18
4.2.2. DPS.....	21
4.3. PIC18F25J11.....	22
4.4. Program .....	23
5. Závěr.....	24
Seznam použité literatury .....	24

### Anotace

Cílem této práce je návrh univerzálního rotujícího LED zobrazovače, který je použitelný v mnoha aplikacích. Důraz je kladen na větší rozlišení a zároveň nižší cenu. Využití by toto zařízení mělo najít v reklamě, informačních panelech, ale také v samostatných aplikacích jako hodiny, či teploměr, nebo dokonce využití umělecké, jako dekorace. V práci jsou popsány dvě možnosti řešení této problematiky.

# 1. Princip

## 1.1. Princip setrvačnosti oka

Je to vlastnost lidského oka, která způsobuje, že dokážeme rozeznávat dva obrazy od sebe pouze pokud se mění pomaleji než 25Hz. Při rychlejším střídání obou obrazů se nám spojí v jeden. A tohoto principu využívám i já ve své práci.

## 1.2. Princip kruhového LED zobrazovače

U většiny dnešních zobrazovačů se používá matice, to znamená mít tolik fyzických bodů, kolik je požadované rozlišení. U LED zobrazovačů je poměrně nákladné a taky pracné vytvořit zobrazovací jednotku s potřebným rozlišením. Proto jsem se rozhodl řešit tento problém pohybem LED diod do kruhu či koule. Pro ilustraci, kdyby bylo potřeba LED pole o velikosti 100 x 100 LED diod, muselo by se použít 10000 LED diod. Při použití rotujícího zobrazovače stačí pouze 100 LED diod. Samozřejmě je nutné dobře zpracovat software pro řízení, ten ovšem není nějak složitý.

Funkci popíšu na kruhovém rotačním zobrazovači. V mém případě pás 16 LED diod se točí rychlostí aspoň 25Hz a tím tvoří kruh o poloměru tedy 16 LED diod. Jedna otočka je rozdělena do  $x$  kroků, pro můj kruhový zobrazovač je to 188. V každém tomto kroku se na LED diodách rozsvítí příslušná kombinace pro danou část právě zobrazovaného obrazu. Při pootočení o další krok se zobrazí následující kombinace a tak dál dokud neskončí otáčka a pak se proces opakuje, buď ze stejným, nebo z následujícím obrázkem.

U kulového rotačního LED zobrazovače je to podobné, jen se jinak čte obrázek. U kruhového se musel z obrázku vytvořit kruh, u kulového je čtení hodnot o mnoho jednodušší, stačí číst po sloupcích a obdélníkový obraz se zobrazí na povrchu koule.

## 1.3. Snímání polohy

Pro snímání polohy LED pásu je více možností. První a asi nejjednodušší, ale také nákladnější je použití krokového motoru, která má odpovídající úhel na jeden krok. V tomto případě by se při každém kroku motoru změnila LED, jak je popsáno výše. Další možností je použití IR závoje. Pro tuto metodu je možné použít stejnosměrný motor. Použití IR se dá rozdělit ještě na dvě možnosti. V první možnosti IR snímá jen jednou za otáčku. Tím můžeme změřit rychlost otáčení a v řídicí jednotce čas periody otáčení rozdělit na příslušný počet dílků. Tento způsob ale vyžaduje přesné konstantní otáčky a toho je těžké dosáhnout. Při nedodržení konstantních otáček se bude obraz klepat. Druhá možnost použití IR je tedy použít v mojí práci. V této variantě IR senzor snímá určitý kotouč, který je rozdělen a požadovaný počet dílků. Tak při každém dílku je do řídicí jednotky poslán impuls. Tento způsob nevyžaduje přesně konstantní otáčky, je ale nutné dobře nastavit IR čidlo. Při použití pouze jednoho IR čidla záleží na startovní poloze snímače, proto jsem u druhého prototypu zavedl ještě jedno IR čidlo, které snímá pouze jeden nulovací impuls. Čítač čítající jednotlivé impulsy z prvního IR je při nulovacím impulsu vynulován.

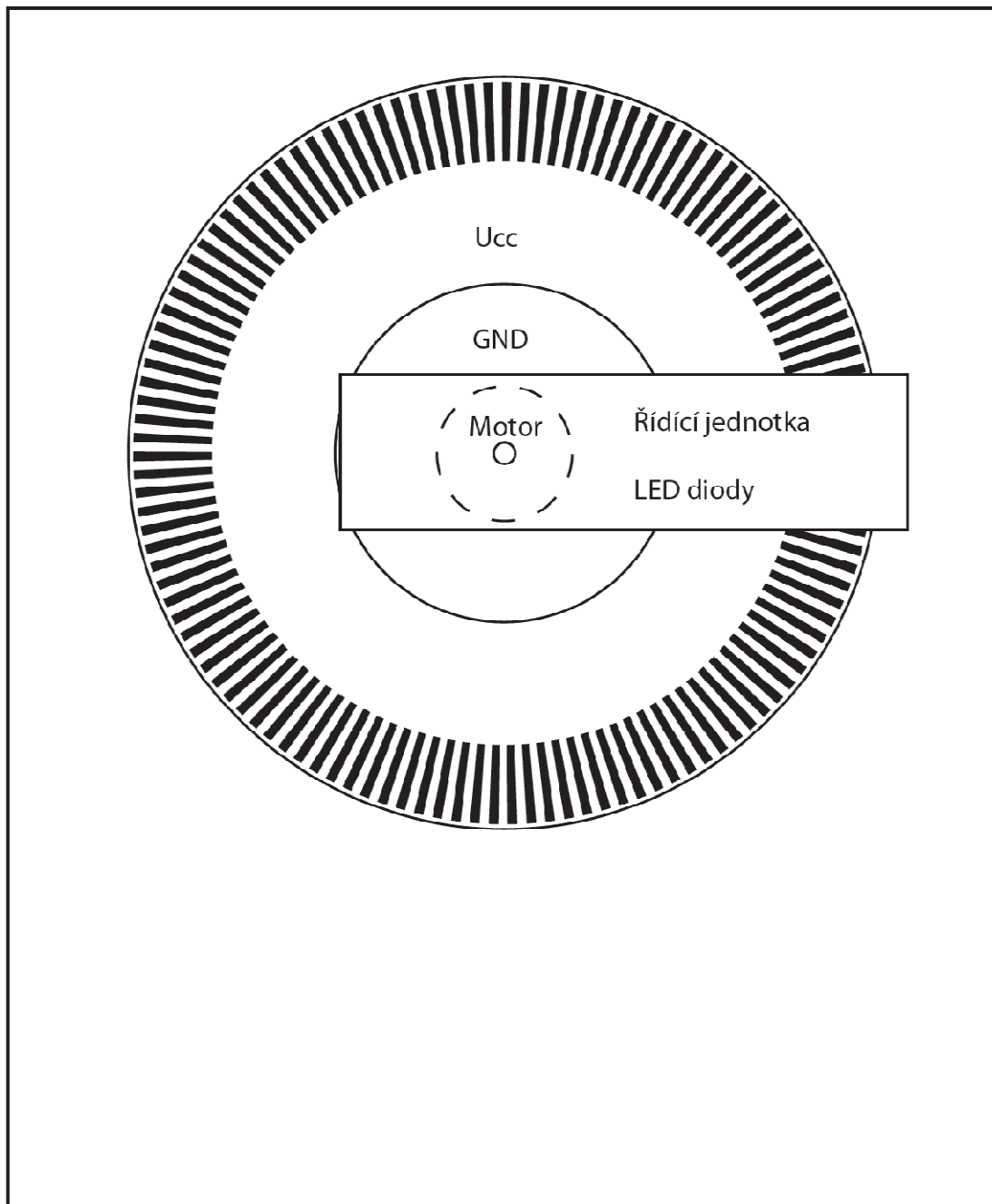
V řídicí jednotce, v mém případě je to mikropočítač PIC od firmy Microchip, se musí čítat impulsy a to do hodnoty souhlasící se skutečným počtem dílků kotouče. Při nesouhlasícím počtu dílků by se obraz roztočil.

## 2. Kruhový rotační LED zobrazovač

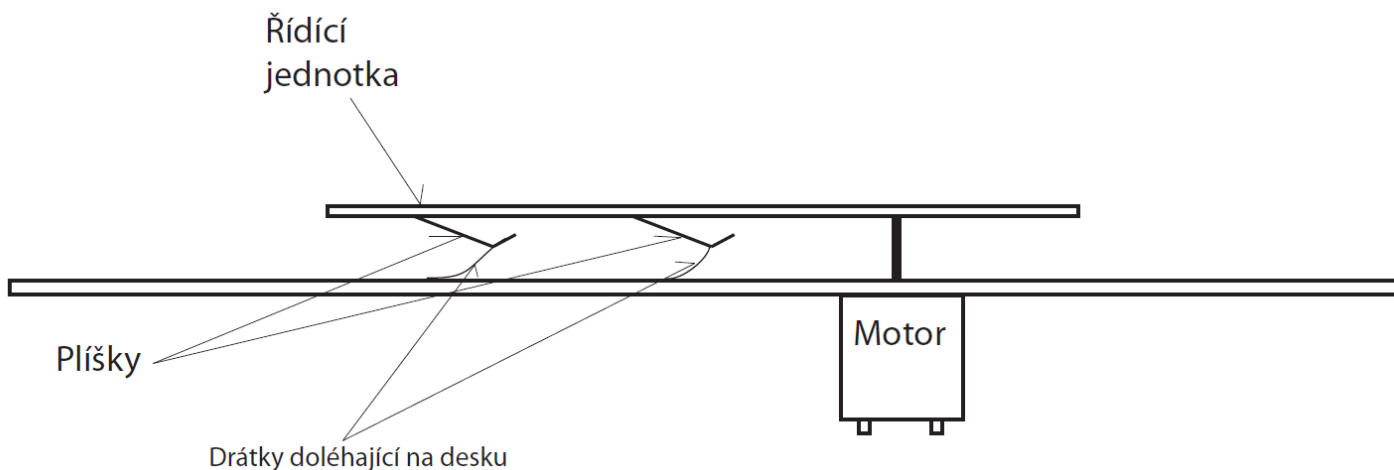
Jako první prototyp jsem vytvořil právě tento kruhový zobrazovač. Technická konstrukce je jednodušší, protože můžeme roztočit přímo desku plošných spojů i se součástkami. U kulového zobrazovače je to složitější.

### 2.1. Technická konstrukce

Na obrázku je vidět náčrt kruhového rotačního zobrazovače ze předu. Na takřka svislou desku je připevněný stejnosměrný motorek. Tato deska je připevněna na dřevěný podstavec. Na této části je také zdroj a regulace otáček motorku. Na osu motorku je přímo připevněna deska plošných spojů s řídicím obvodem, LED diodami a tlačítkem. Na obrázku je také vidět kotouč určující rozlišení.

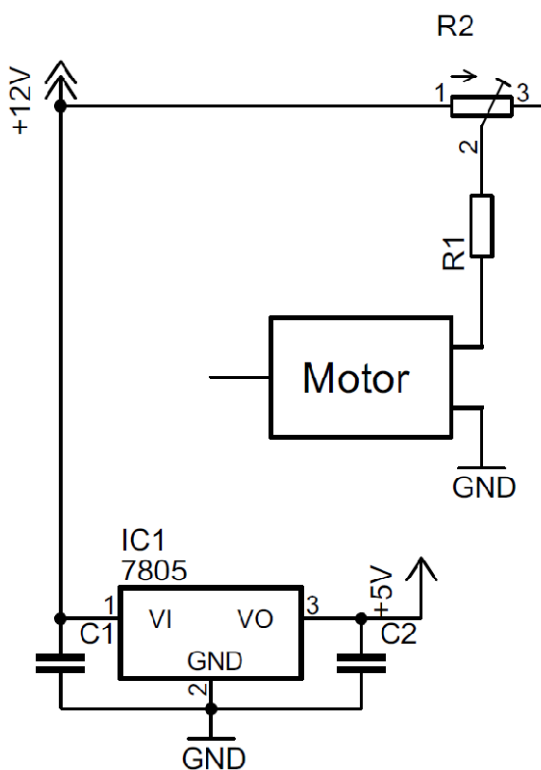


Napájení řídicí jednotky je realizováno pomocí dvou snímacích kartáčků. Na nákrese lze vidět plochu s kladným napájecím napětím ( $U_{cc}$ ) a plochu se záporným napájecím napětím (GND). Na tyto plochy doléhají dva kartáčky, připevněné na řídicí jednotku. Kartáčky jsou tvořeny z pevnějšího plíšku a drátku, který přiléhá na desku. Tento systém se ale bohužel projevil jako nespolehlivý a poruchový, plíšky vydávají při pohybu zvuk a to je velmi nepříjemné. U další verze budu problém napájení řešit pomocí ložisek.



U konstrukce je velmi důležité, aby byla točivá část správně vyvážená. U tohoto prototypu tomu tak bohužel není, proto musí být celé zařízení dostatečně zatíženo, aby neposkakovalo. Uchycení řídicí jednotky na osu motoru je provedeno pomocí kovové svorkovnice.

### 2.2. Napájení a regulace otáček motoru



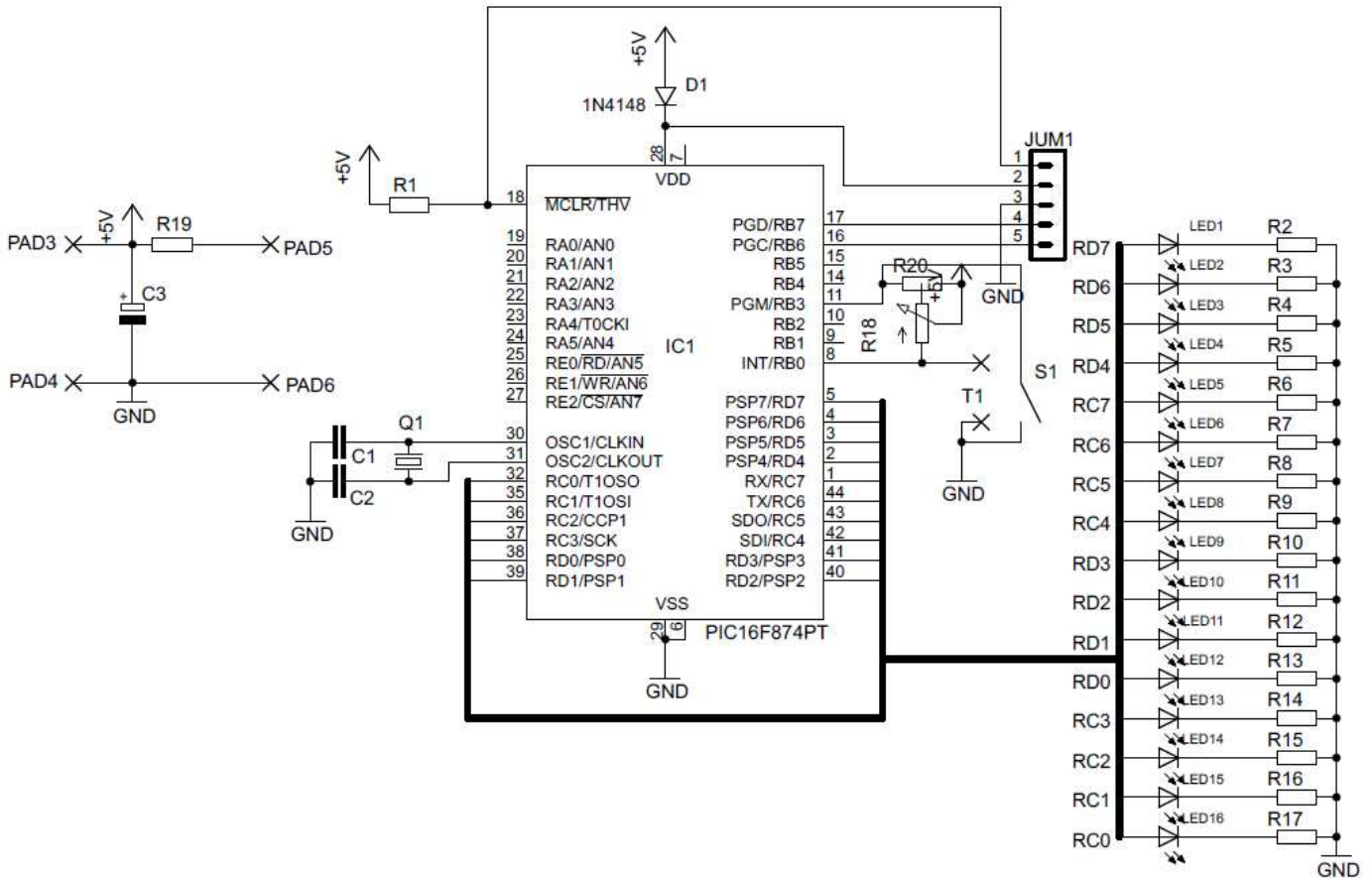
Na obrázku je jednoduché schéma. Pro stabilizaci napětí je použit stabilizátor 7805. Na jeho vstup je připojeno napájení 12V z externího adaptéru. Jeho výstup je přiveden přes kartáčky na řídicí pohyblivou desku.

Otáčky motoru jsou velmi jednoduše regulovány. Napětí je přivedeno přes sériově zapojené odpor a trimr. Jak jsem již zmínil, proud protékající tímto motorem není tak velký, aby bylo nutno použít aktivní regulaci. Trimrem tedy můžeme upravit otáčky.



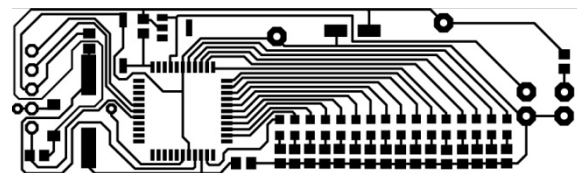
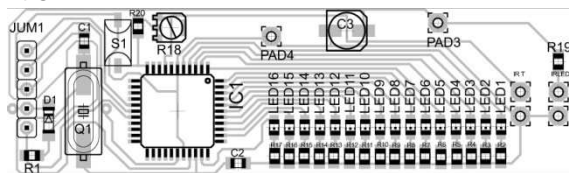
### 2.3. Schéma a DPS

Schéma zapojení:



Řídícím obvodem je PIC16F874A, na který jsou připojeny všechny periferie. Na PAD3 a PAD4 je přes kartáčky přivedeno napájecí napětí. Na PAD3 kladné napětí +5V a na PAD4 zem. Je zde vyhlazovací kondenzátor s vysokou kapacitou, který má za úkol také podržet napětí při chvilkovém přerušení napájení v důsledku nedokonalosti napájecích kartáčků. Přes odpor R19 je na svorkách připojena IR LED dioda, která je namířena nad enkodérem (kotoučem). K obvodu PIC16F874A je připojen krystalový oscilátor 20MHz. Na vstup RB0 je připojen fototranzistor, který je umístěn vedle IR LED diody a snímá infračervený signál vyslaný LED diodou odražený od enkodéru. K RB0 je také připojen trimr, kterým se nastavuje citlivost fototranzistoru. Na vstup RB3 je připojeno tlačítko S1. Na PORTC a PORTD jsou připojeny LED diody, které zobrazují požadovaný obraz. K mikropočítači jsou také připojeny jumpery pro programování ICSP (In-Circuit Serial Programming).

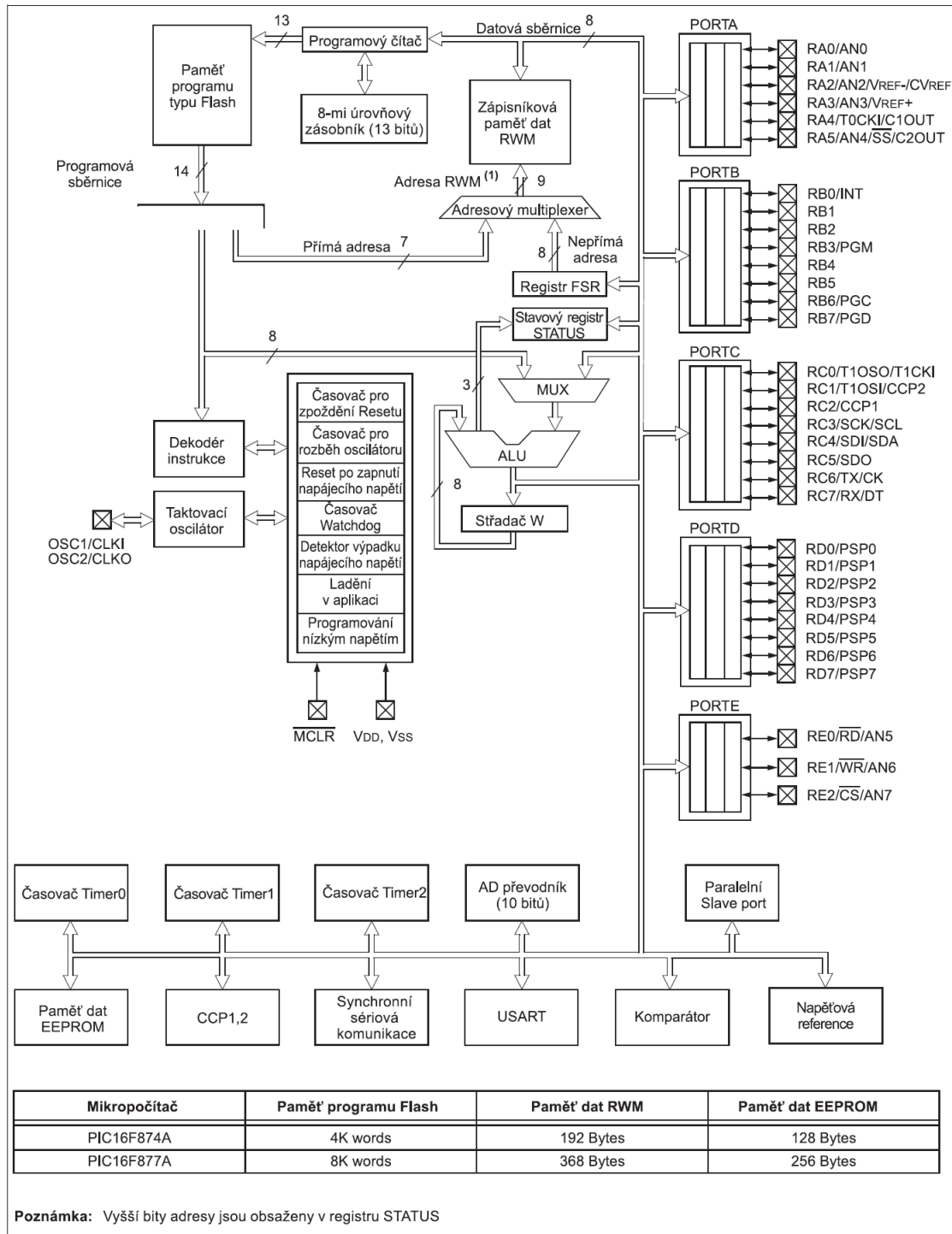
#### DPS



## 2.4. Popis mikropočítače PIC16F874A

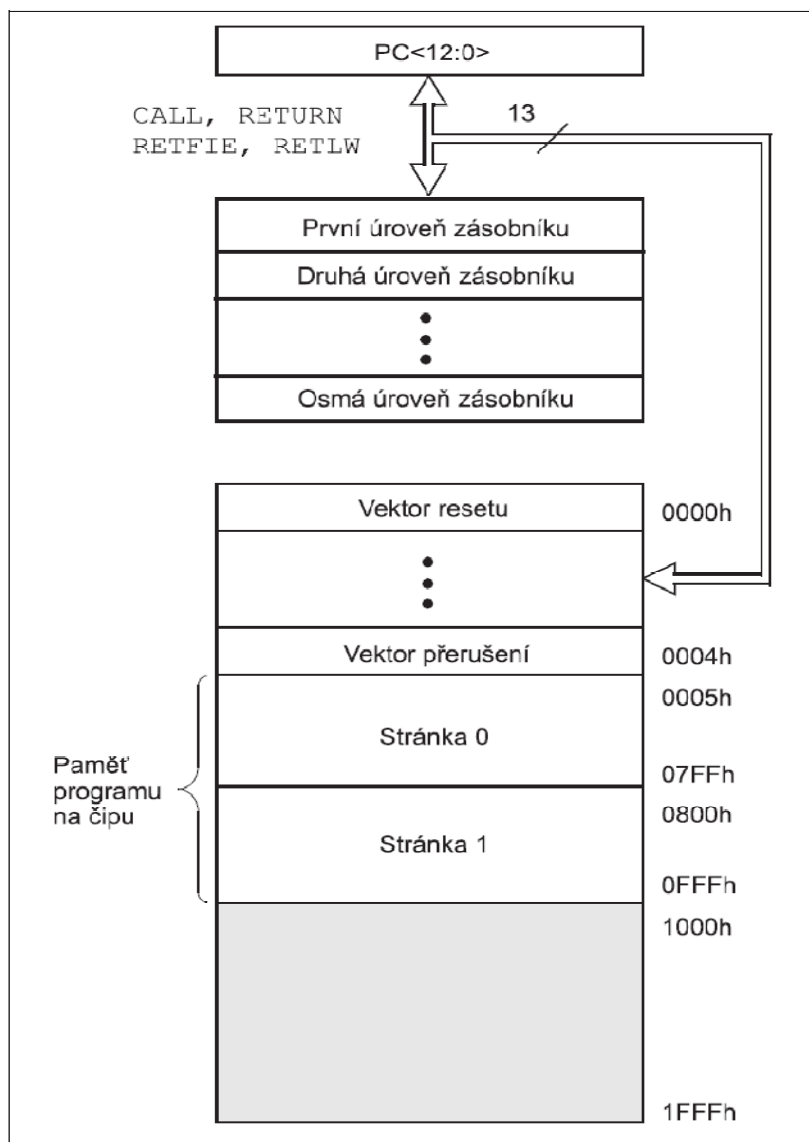
Je to jednočipový RISC mikropočítač od firmy Microchip. Instrukční sada se skládá z 35 instrukcí. Maximální taktovací frekvence je 20MHz. Paměť programu je 7,2KB tj. 4096 instrukcí. Paměť dat RWM je 192B a paměť EEPROM je 128B. Mikropočítač obsahuje 5 vstupně výstupní porty, neboli 33 vstupně výstupních pinů. 10-bitový A/D převodník s nejméně osmi vstupy. Dále obsahuje 3 čítače/časovače. Modul komparátoru a 2 výstupy PWM. Synchronní sériový port s SPI nebo I2C. Universální asynchronní port USART/SCI a paralelní port.

### 2.4.1. Blokové schéma mikropočítače PIC16F874A



Blokové schéma zobrazuje jednotlivé části vnitřní struktury mikropočítače.

## 2.4.2. Rozvržení paměti programu



Na obrázku vlevo je vidět organizace paměti programu. Jak je vidět je zde Vektor resetu, zde po resetu začíná program. Dále je zde jeden vektor přerušení. Zdrojů přerušení je více, pokud používáme více přerušení najednou, musíme před vykonáním podprogramu přerušení rozhodnout co bylo právě zdrojem tohoto přerušení a také nastavit priority jednotlivých přerušení.

Pro paměť programu jsou k dispozici 2 stránky. Na obrázku lze také vidět zásobník pro návrat z podprogramu. Ten umožňuje až 8 zanoření do podprogramů současně. Při devátém volání podprogramu a po jeho ukončení již dojde ke kolizi programu. Na horní části obrázku je zobrazen 12-ti bitový čítač PC a ten udává aktuální místo programu. Je pomocí něho možno provádět nepřímé skoky v programu.

### 2.4.3. Speciální funkční registry (SFR)

Adresa registru		Adresa registru		Adresa registru		Adresa registru	
Indirect addr. (*)	00h	Indirect addr. (*)	80h	Indirect addr. (*)	100h	Indirect addr. (*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD <sup>(1)</sup>	08h	TRISD <sup>(1)</sup>	88h		108h		188h
PORTE <sup>(1)</sup>	09h	TRISE <sup>(1)</sup>	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Rezervováno <sup>(2)</sup>	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Rezervováno <sup>(2)</sup>	18Fh
T1CON	10h		90h		110h		190h
TMR2	11h	SSPCON2	91h				
T2CON	12h	PR2	92h				
SSPBUF	13h	SSPADD	93h				
SSPCON	14h	SSPSTAT	94h				
CCPR1L	15h		95h				
CCPR1H	16h		96h				
CCP1CON	17h		97h				
RCSTA	18h	TXSTA	98h				
TXREG	19h	SPBRG	99h				
RCREG	1Ah		9Ah				
CCPR2L	1Bh		9Bh				
CCPR2H	1Ch	CMCON	9Ch				
CCP2CON	1Dh	CVRCON	9Dh				
ADRESH	1Eh	ADRESL	9Eh				
ADCON0	1Fh	ADCON1	9Fh				
	20h		A0h		120h		1A0h
Registry pro všeobecné použití (Zápisníkova paměť RWM) 96 Bajtů		Registry pro všeobecné použití (Zápisníkova paměť RWM) 96 Bajtů		Mapováno do 20h-7Fh		Mapováno do A0h - FFh	
	7Fh		FFh		16Fh		1EFh
					170h		1F0h
					17Fh		1FFh

Bank 0      Bank 1      Bank 2      Bank 3

■ Nepoužité buňky, při čtení "0"  
\* Nejedná se o fyzický registr.

**Poznámka 1:** Tyto registry nejsou použity u mikropočítače PIC16F873A.  
**2:** Tyto registry jsou rezervované, ponechte jejich obsah roven 00h.

Na obrázku vlevo jsou zobrazeny SFR seřazené podle jednotlivých stránek, neboli bank, na kterých se nacházejí. Pokud chceme pracovat, to znamená číst, nebo zapisovat, s SFR, tak je nutné předem nastavit banku. To se provádí v registru STATUS bity RPO a RP1. Do dvou bitů lze zapsat 4 kombinace, tedy binární čísla 0 – 3 a ty odpovídají jednotlivým bankám. Jak je vidět na obrázku, registr STATUS je přítomen ve všech bankách.

Každý SFR má určitou funkci, která ovládá vnitřní periferie, nastavují se pomocí nich vstupně-výstupní porty, čítače/časovače, přerušení, AD převod, komparátor, PWM výstupy a samozřejmě implementované komunikační rozhraní.

Na všech čtyřech stránkách je také kromě registru STATUS také registr FSR a na úplně první pozici je registr INDF

(Indirect address) a Zapsáním do tohoto registru se námi vložená hodnota uloží na pozici určenou registrem FSR, tomuto způsobu se říká nepřímé adresování.

Je také nutno si uvědomit, že většina operací se provádí přes pracovní registr W.

### 2.4.4. I/O porty

Na blokovém schématu mikropočítače je dobře vidět všech 5 portů (PORTA, PORTB, PORTC, PORTD, PORTE, PORTF) a také kolik má který pinů. Každý pin portu lze nakonfigurovat jako digitální vstup či výstup. Většina portů má ještě další možnosti nastavení, jako analogový vstup, některé komunikační rozhraní, PWM výstup, vstup vnějšího oscilátoru a podobně.

Digitální vstupně výstupní piny u všech portů nastavují registry tris (TRISA, TRISB, TRISC, TRISD, TRISE, TRISF). Jestliže bude v registru TRISA hodnota 00000000b, tak všechny piny PORTA budou výstupní, jestliže budou v TRISA 11111111b tak piny PORTA budou vstupní. Konfigurace dalších možností nastavení portů, jako AD vstupy, komunikační rozhraní, se nastavují pomocí dalších SFR a ty jsou přehledně popsány v datasheetu, proto je zde nebudu uvádět.

### 2.4.5. Přerušování

PIC 16F874 má možnost vyvolání přerušování od 15-ti zdrojů.

#### **Princip přerušování**

Po povolení přerušování od požadovaného zdroje a vyvolání přerušování od tohoto zdroje způsobí skok na vektor přerušování, to je adresa v programu (0004h) na které je uložen podprogram přerušování. Tento podprogram může být vyvolán kdykoli a proto, aby se neovlivnil běh hlavního programu, tak se musí na začátku podprogramu přerušování uložit registr W a registr STATUS. Tyto dva registry budou před návratem z přerušování obnoveny na původní hodnotu. Každý zdroj přerušování má také příznakový bit tohoto přerušování. Příznakový bit musí být v podprogramu přerušování vynulován. Při použití více přerušování je nutno nastavit prioritu přerušování a také v podprogramu přerušování rozhodnout, z jakého zdroje bylo přerušování vyvoláno.

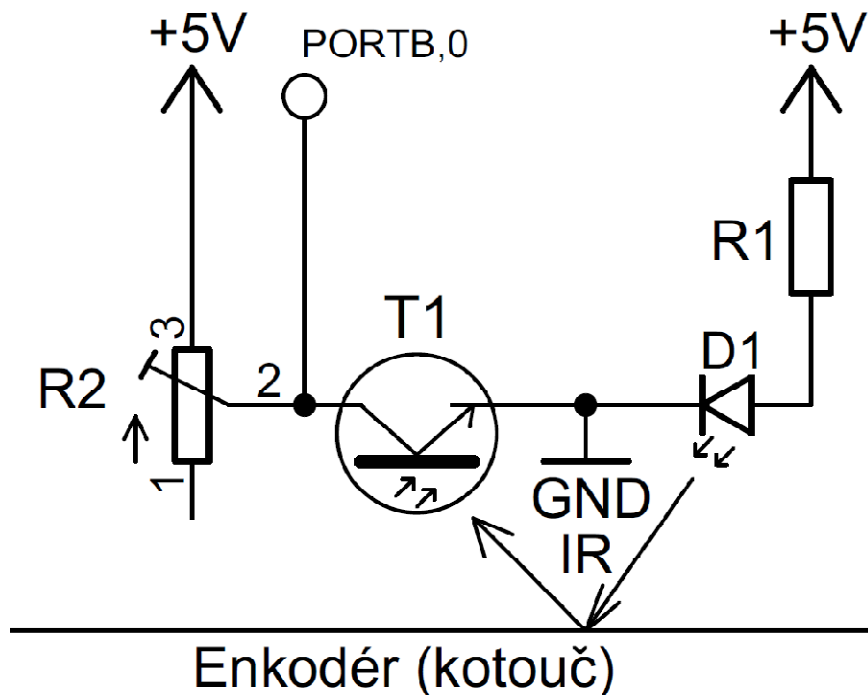
Nastavení přerušování se provádí pomocí registrů INTCON, PIE1, PIR1, PIE2, PIR2.

### 3. IR senzor a enkodér

Pro snímání aktuální polohy řídicí jednotky jsem použil IR senzor a enkodér.

#### 3.1. Monochromatické IR čtení barvy

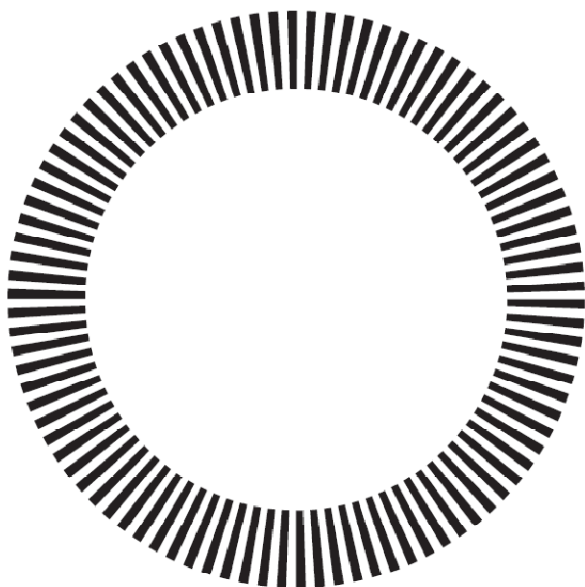
Monochromatickým snímáním barvy je myšleno snímání černé a bílé a to pomocí IR senzoru. K tomu je použita dvojice IR LED dioda a IR foto-tranzistor. Ty jsou namířené na čtený povrch tak, aby se IR signál z LED odrazil od čteného povrchu směrem k foto-tranzistoru.



Výpočet R1: 
$$R1 = \frac{U_{cc} - U_D}{I_D} = \frac{5 - 1,2}{20 \times 10^{-3}} = 190 \Omega$$

Pomocí odporového trimru R2 lze regulovat citlivost tranzistoru T1. Nastavení musí být takové, aby bílá barva povrchu znamenala logickou 1 a černá barva logickou 0.

#### 3.2. Enkodér



Jak je vidět na obrázku, je to kotouč, na jehož obvodu jsou proužky, ty jsou snímány IR senzorem, jenž byl popsán výše. Těchto proužků je zde 188 a toto číslo určuje z kolika polopřímek se bude výsledný obraz skládat. Jednu tuto polopřímku tvoří tedy v tomto případě 16 LED. Enkodér tedy určuje rozlišení.

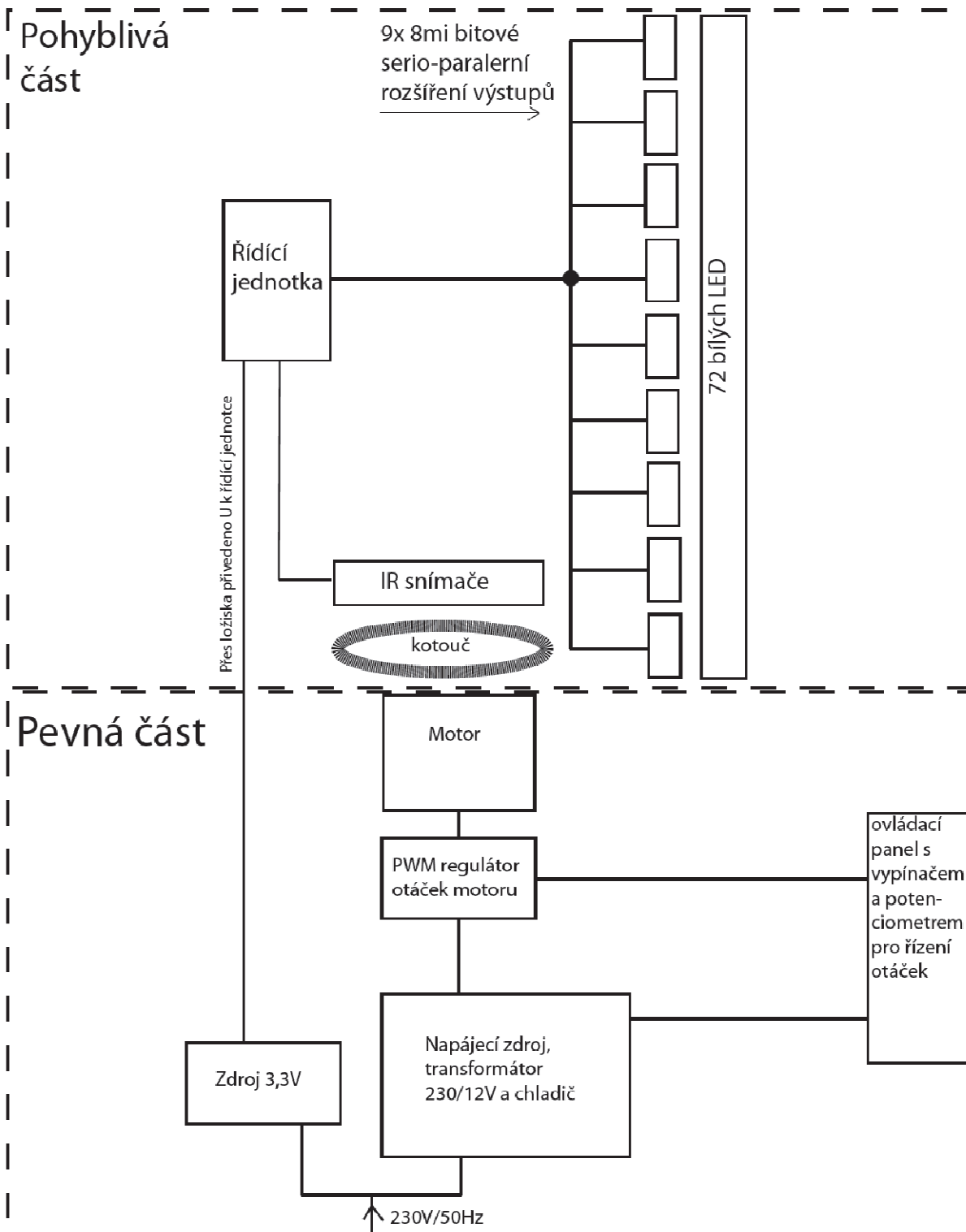
### 4. Kulový rotační LED zobrazovač

Oproti předchozímu prototypu je tato verze větší a celkově vylepšena. Odstraňuje použitím ložisek problém s kartáčky, dále je zde pro větší rozlišení použito více LED a také neméně důležitým faktorem je použití lepšího řídicího obvodu.

Tato verze, jak je zřejmé z názvu, zobrazuje požadované informace na povrchu koule, kterou vytvoří rotací a multiplexním řízením LED stejně jak tomu bylo u předchozí verze, jen s kruhem. Řídicím obvodem je PIC18F25J11. Tento obvod jsem vybral, pro jeho rychlost, dostatečnou kapacitu a také pro to, že může pracovat na nejvyšší možné rychlosti již od 2,35V. To, že může být použito nižší napájecí napětí umožňuje připojit LED bez jakýchkoli předřadných odporů.

## 4.1. Schémata zapojení

### 4.1.1. Blokové schéma



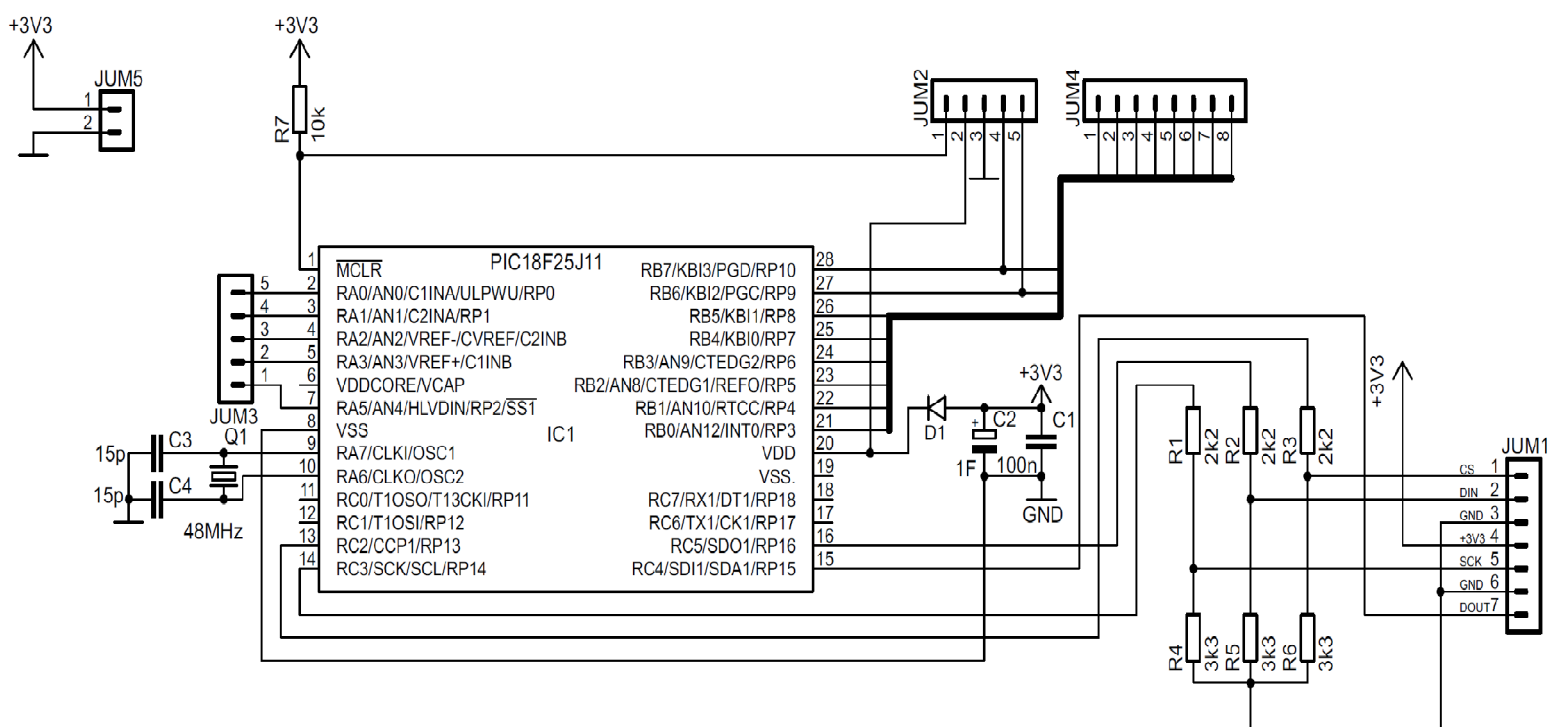
Na blokovém schématu vidíte 2 hlavní části. Pevná část představuje šasi, na které je pak připevněna pohyblivá část, technická konstrukce bude blíže popsána dále. V pevné části jsou tedy umístěny napájecí zdroje, PWM regulátor otáček motoru, motor a jednoduchý ovládací panel. Napájení motoru je realizováno pomocí toroidního transformátoru se vstupním napětím 230V a výstupním 12V. Pomocí 8A můstkového usměrňovače, který je připevněn na chladič, je napětí z transformátoru



## Rotující LED zobrazovač

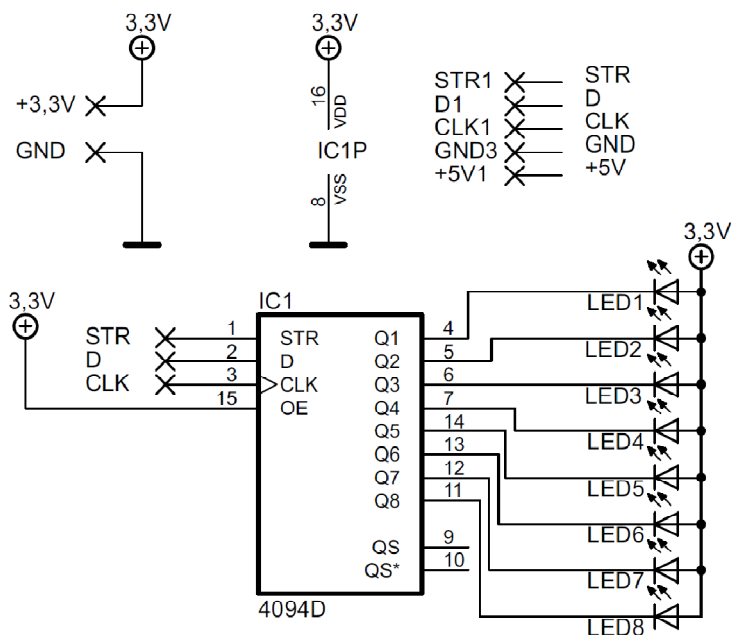
usměrněno a je přivedeno na vstup PWM regulátoru. Pro PWM regulátor jsem použil stavebnici od firmy Tipa, regulace otáček je nutná, ale není to předmětem mé práce, proto jsem se s tímto tématem nezdržoval. Regulace se provádí na ovládacím panelu pomocí potenciometru. Motor je připojen na výstup PWM regulátoru. Na ovládacím panelu je také umístěn vypínač, který celé zařízení zapne či vypne. Dále je zde umístěn další napájecí zdroj, který dodává 3,3V do pohyblivé jednotky přes 2 ložiska. Pohyblivá část obsahuje řídicí jednotku s PIC18F25J11, na ni je připojeno 9 desek pro rozšíření výstupů, pomocí 9ti serio-paralerních převodníků. Na výstupy těchto převodníků je připojeno 72 LED. Také jsou k řídicí jednotce připojena IR čidla, která snímají kotouč.

### 4.1.2. Schéma zapojení řídicí jednotky



Na schématu výše je řídicí jednotka, jak je vidět, je velmi jednoduchá, je to proto, že je umístěna na pohyblivé části zařízení a proto není míst o na zbytečné součástky. K mikropočítači PIC18F25J11 (IC1) je připojen krystal 48MHz a také konektor pro ICSP programování (JUMP2). Dioda D1 zabraňuje průchodu proudu do jiných částí, než do mikropočítače při programování. Na jumper (JUM5) je připojeno napájení 3,3V. Porty PORTA a PORTB jsou vyvedeny na jumpery (JUM3, JUM4). K pinům PORTC, 2, 3, 4, 5 je připojena SD paměťová karta, která bude sloužit k rozšíření paměti a hlavně bude jakýmsi uživatelským rozhraním.

### 4.1.3. Schéma zapojení serio-paralerního převodníku



Na schématu je zapojen serio-paralerní převodník CMOS4094. Z řídicí jednotky je přivedeno napětí 3,3V a dále signály CLK, STR a D. Napájení i signály CLK a STR jsou propojeny mezi sebou a signál D je přiveden ke každému z devíti serio-paralerních převodníku zvlášť. Signál CLK je hodinový signál, signál STR neboli storage slouží k podržení aktuálních hodnot na výstupu v době přenosu a signál D slouží k přenosu dat.

### 4.1.4. IR snímače

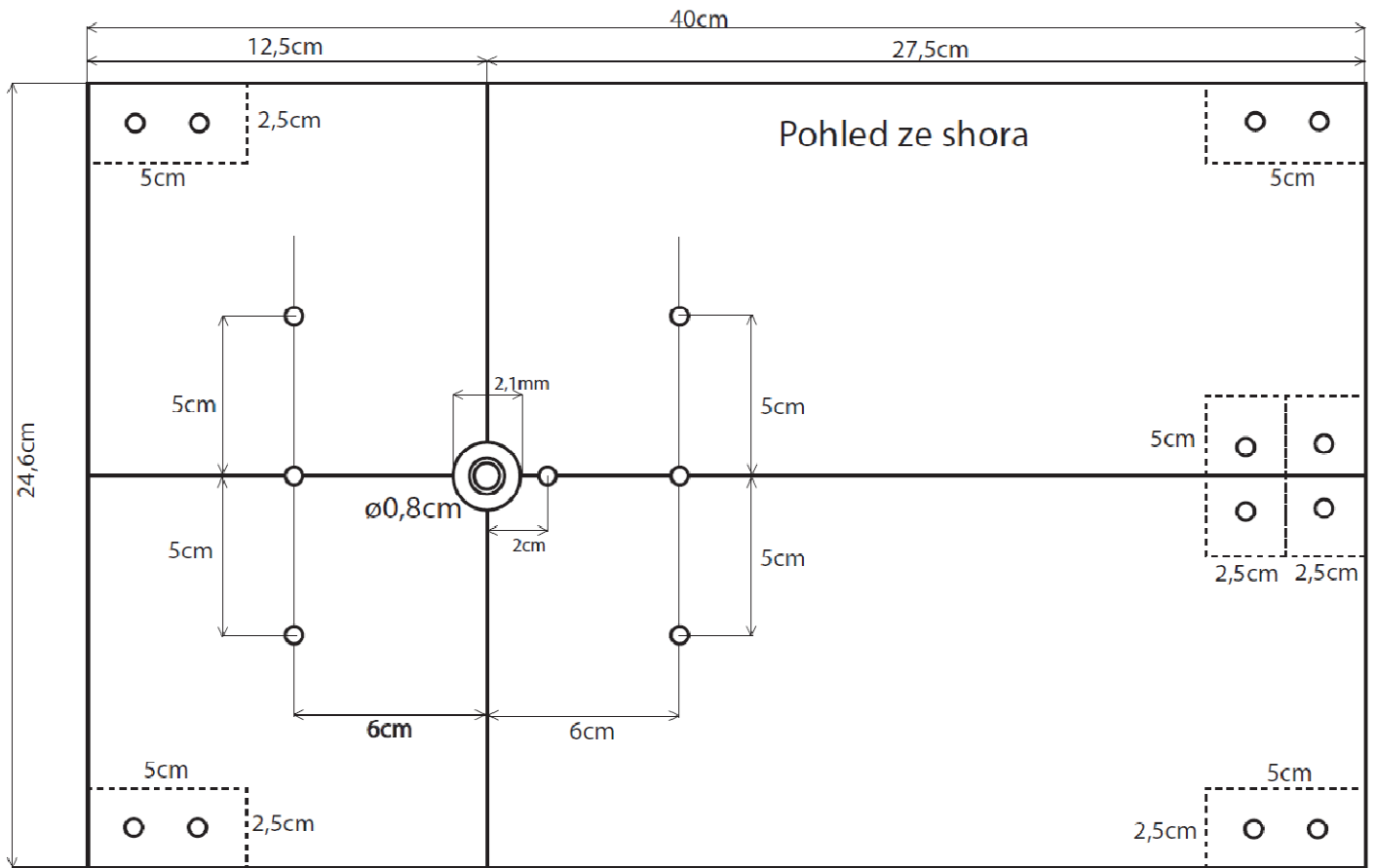
Schéma zapojení IR čidla je totožné s předchozí verzí. Rozdíl je ale v tom, že již není použit enkodér. Je zde opět kruh, ten je ale celý bílý až na jeden černý proužek.

Zjištění polohy se tedy děje odlišně od předchozí verze. Při každé otáčce se změří čas jednoho otočení, tento čas se v mikro počítači vydělí počtem požadovaných sloupců, pro tuto verzi je to 128. Při další otáčce se vždy porovnává s aktuálním časem vypočtený jeden dílek (=sloupec) a podle toho se zobrazuje příslušný sloupec.

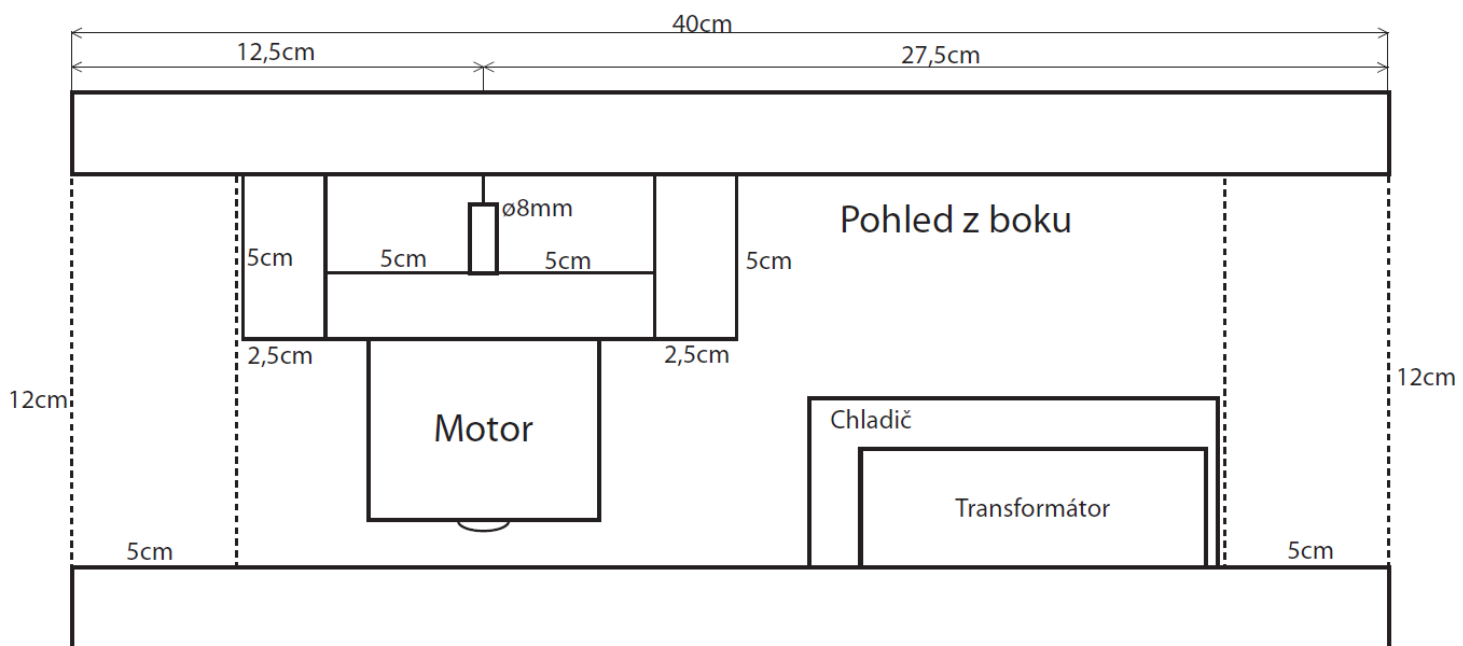
## 4.2. Konstrukce

### 4.2.1. Technická konstrukce

Pohled ze shora

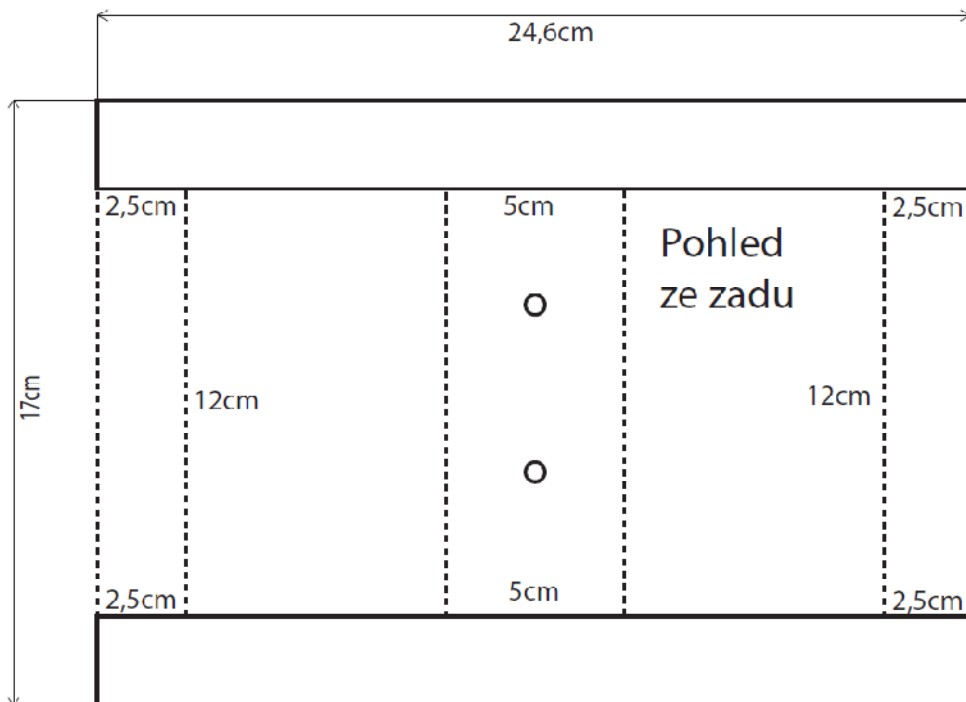


Pohled z boku



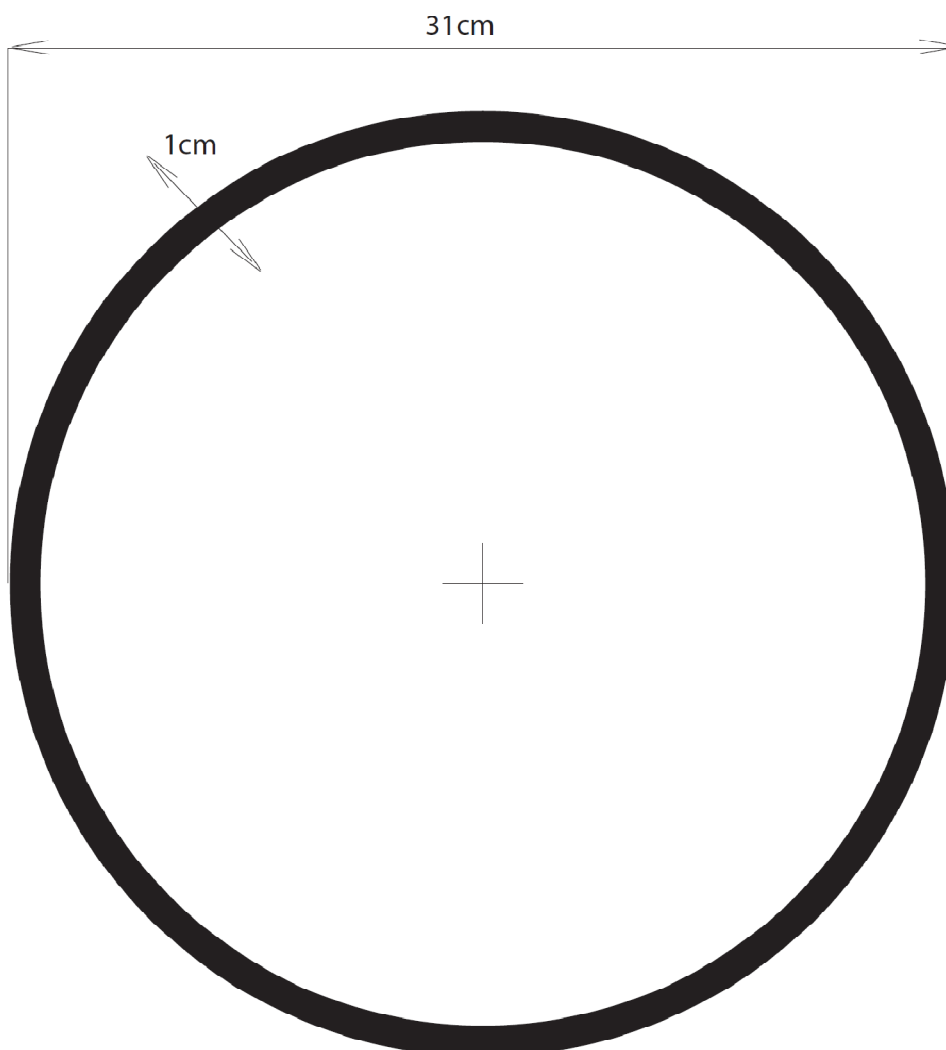
## Rotující LED zobrazovač

Pohled ze zadu



Nosná i krycí skříň je ze dřeva. Rozměry a konstrukce jsou vidět v předchozích třech obrázcích. Skříň se skládá z dvou desek o rozměrech 40x24,6x2,5cm a ty jsou umístěny nad sebou a navzájem spojeny 4mi špalíčky o rozměrech 12x5x2,5. Na pohledu shora lze vidět ještě ložisko a okolo něj díry pro uchycení motoru.

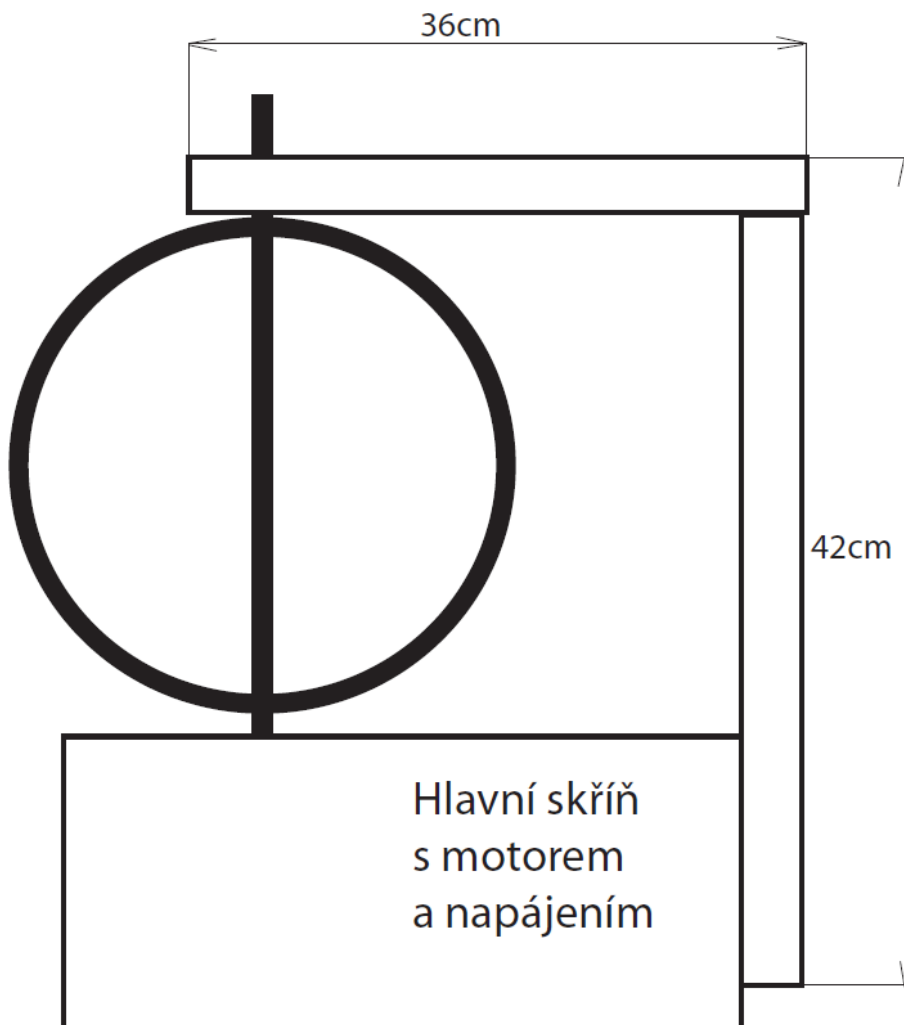
Kruh



Pro uchycení LED a rotaci jsem použil plastový kruh jehož náčrt vidíte vpravo. Na něj jsou uchyceny LED, DPS pro rozšíření výstupů a také řídicí jednotka a IR čidla.

## Rotující LED zobrazovač

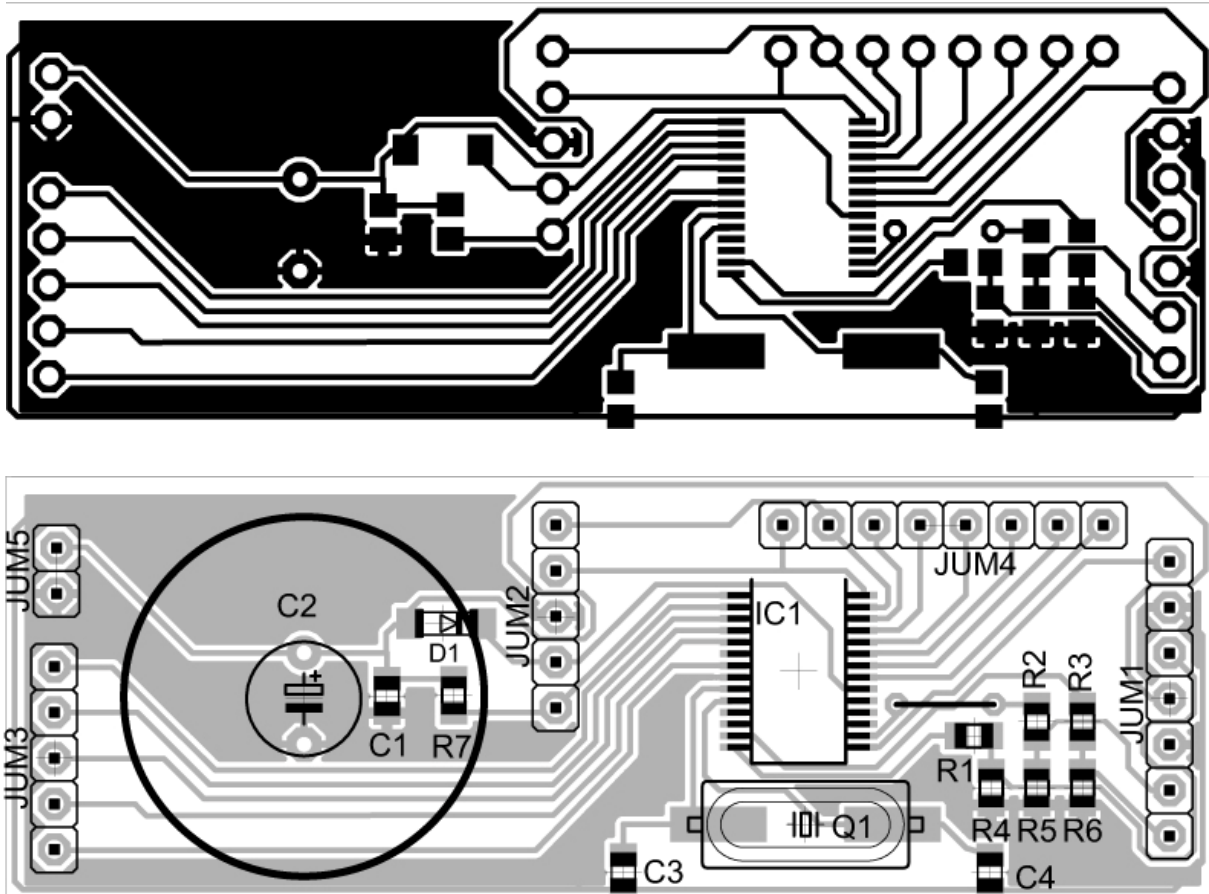
Zde vidíte plastový kruh s LED připojenými k devíti serio-paralerním převodníkům. Ještě chybí připojit řídicí jednotku.



Zde vidíte jakým způsobem je uchycen kruh k hlavní skříni. Jsou zde 2 ložiska, které jednak drží kruh v ose, ale také slouží k napájení pohyblivé části.

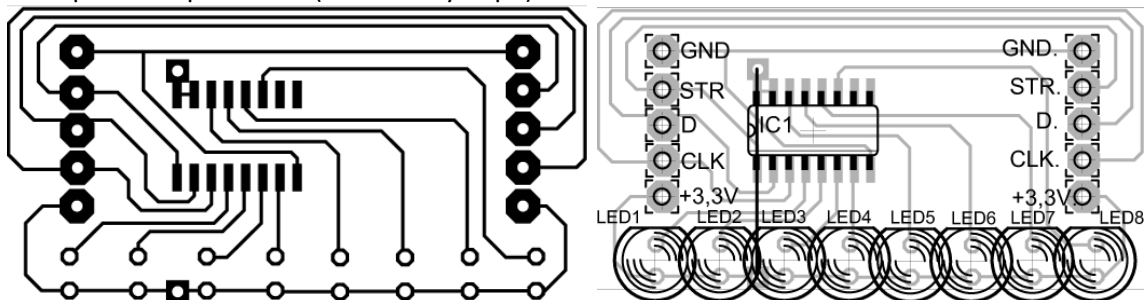
4.2.2. DPS

Řídící jednotka



Tato deska je, jak je vidět, velmi jednoduchá. Na horním obrázku je DPS a na spodním je zobrazeno rozmístění součástek. Na JUM1 bude připojena SD karta, JUM2 slouží k ICSP programování, na JUM3 a JUM4 jsou připojeny řídicí a datové signály pro rozšíření výstupů připojených na LED a také jsou zde 2 vstupy IR senzorů. JUM5 slouží k napájení 3,3V. Tato deska je uchycena na vnitřní straně kruhu naproti LED.

Serio-paralerní převodník (rozšíření výstupů)

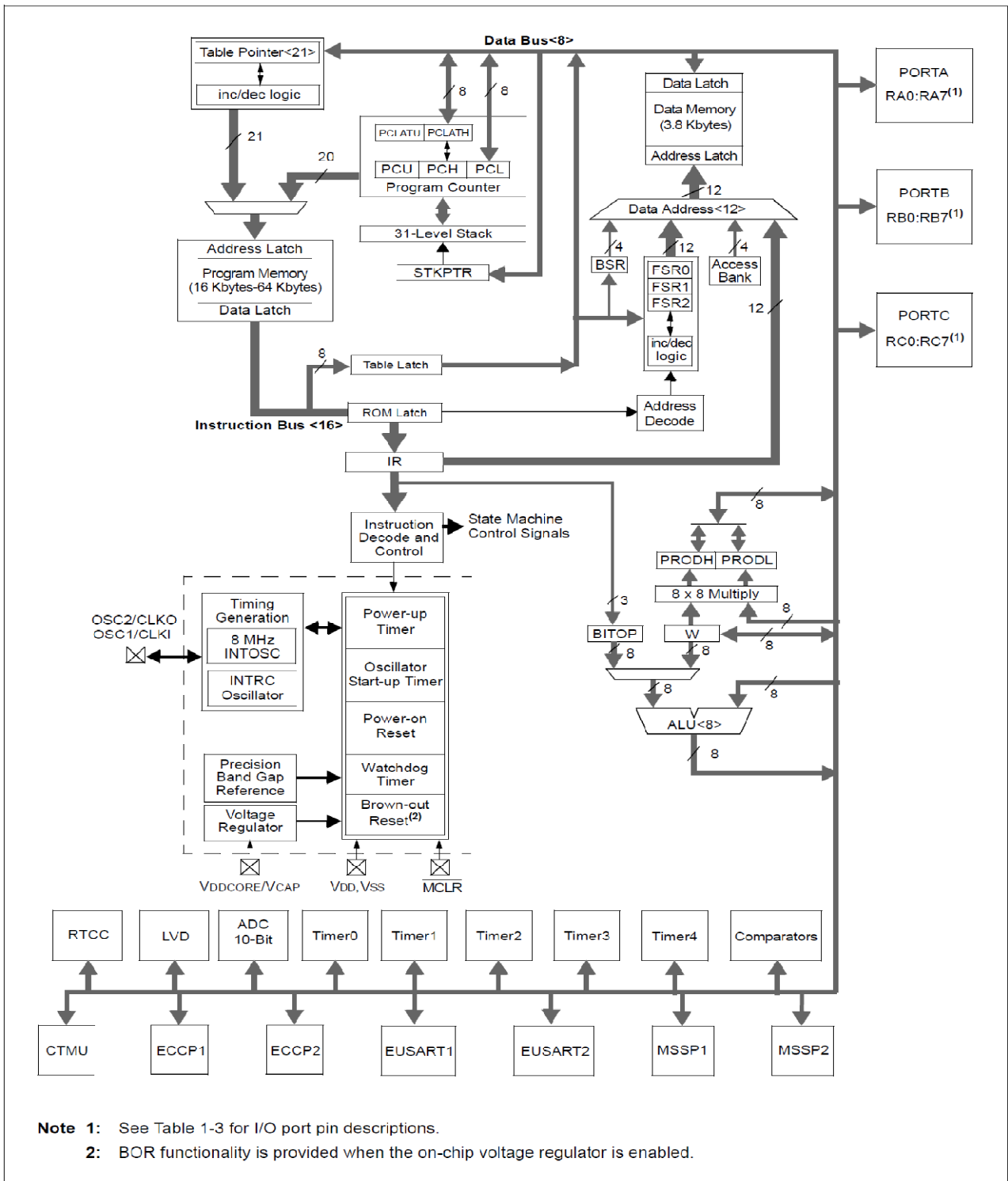


Těchto desek je celkem 9. Jsou umístěny na vnitřní straně kruhu a jsou propojeny se 72 LED, které jsou prostrčeny z vnější strany kruhu 144 dírkami a připájeny k těmto deskám. To zajišťuje upevnění jak DES tak LED. Všechny 9 desek má také propojené napájení (+3,3V a GND), hodinový signál (CLK) a storage (STR). Signál D je vyveden z každé desky samostatně.

### 4.3. PIC18F25J11

Tento mikro počítač je stejně jako u předchozí verze od firmy Microchip a RISC architektury. Je ovšem oproti řady PIC16FXXX na mnohem vyšší úrovni. Zahrnuje více vnitřních periférií, taktovací frekvence je více než 2x větší než u PIC16FXXX, dále je zde více paměťového prostoru, také má menší napájecí napětí a nižší spotřebu.

Konkrétně PIC18F25J11 jsem vybral jak pro jeho nízké napájení což umožní připojení LED bez předřadných odporů, dále má poměrně velkou paměť (32KB) což je neméně důležitý faktor, protože budou muset být uloženy pole s daty obrazů.



Na předchozím obrázku je blokové schéma vnitřní struktury mikropočítače. Vpravo nahoře jsou vidět vstupně-výstupní porty PORTA, PORTB, PORTC. Celkem 24vstupně-výstupních pinů. Dole jsou znázorněny vnitřní periférie: 10bit. ADC (analogově-digitální převodník), čítače/časovače TIMERO, TIMER1, TIMER2, TIMER3 a TIMER4, komunikační rozhraní, RTCC obvod reálného času, komparátor a další.

### 4.4. Program

Program bylo třeba vytvořit tak, aby byl schopen vykreslovat obrazy a zároveň, aby je byl schopen měnit v zadaných intervalech.

#### Programovací jazyk

Použil jsem jazyk C konkrétně mikroC od firmy Mikroelektronika. Prostředí od této firmy má implementované množství podporovaných obvodů od firem Microchip, Atmel, 8051 a Sensory. V tomto prostředí lze také použít množství jak hardwarových, tak i softwarových knihoven např. komunikace s multimediální kartou (SD), kterou mám v plánu realizovat uživatelské rozhraní. Také jsou k tomuto prostředí dodány další praktické programy, jako bitmap editor, ve kterém se jednoduše převede obrázek na proměnnou v jazyku C a tento prográmeček samozřejmě využívám v této práci.

#### Čtení enkodéru

Enkodér je, jak už bylo popsáno výše, snímán IR čidly a ty jsou nastaveny tak, aby bílé místo na enkodéru znamenalo logickou 1 a černé místo logickou 0.

Obvod největší kružnice, kterou opisují LED je asi 99 cm, a při průměru jedné LED 0,5mm je ideální počet kroků 198. To znamená, že na povrchu koule by se mělo rozložit 198 sloupců LED. Na enkodéru je ovšem jen 99 čar a to zajišťuje větší přesnost, čím méně dílků, tím jsou větší a čtení větších dílků zaručuje méně chyb. Program čte každý přechod z černé na bílou i z bílé na černou. To se provádí jednoduše pomocí dvou příznaků. V inicializaci se oba příznaky vynulují, dále program čte IR čidlo, a pokud přečte log. 1, tak nastaví příznak pro log. 1 a zkontrolují se oba příznaky. Po té pokračuje program ve čtení. Jestliže přečte log. 0, tak nastaví příznak pro log. 0. Nyní se opět zkontrolují oba příznaky a teď jsou již oba v 1 a tak se provede akce a po jejím vykonání se opět oba příznaky nulují. Nyní se proces opakuje, jen bude nejprve 0 a po té 1.

#### Vykreslování obrazu

V paměti programu jsou uložena data obrazu, pomocí již zmíněného bitmap editoru. Akce, která se vykoná při přečtení změny barvy zobrazí jeden z 198 sloupců obrazu. Jak již bylo zmíněno, tak celé zařízení pracuje v multiplexu, takže je nutné obraz vykreslovat stále, i když se vykonávají další akce. To je zajištěno v hlavním programu tak, že je rychlý a neobsahuje čekací smyčky. Prodlevy se proto musí realizovat pomocí čítačů. Akce vykreslující obraz pracuje tak, že v jedné proměnné je uloženo aktuální číslo sloupce, podle tohoto čísla se načtou příslušné hodnoty z obrazu a odešlou se na serio-parallelní převodníky, aby je LED zobrazily. Proměnná určující aktuální sloupec je inkrementována při každé změně čtené IR čidlem nad vnitřním kotoučem enkodéru. Také se kontroluje, jestli už číslo sloupce nepřesáhlo 198, jestli ano, to znamená 1 otáčku, tak se toto číslo vynuluje. Pokud by nastala chyba v pořadí sloupce, tak se stále kontroluje IR čidlo nad vnějším kotoučem enkodéru a ten čte pouze 1 černou čárku za otáčku. Pokud je tedy čidlo na této čáře, tak bez ohledu na aktuální číslo sloupce toto číslo vynuluje a to zajistí, že se obraz nebude otáčet a bude mít vždy stejnou orientaci.



