

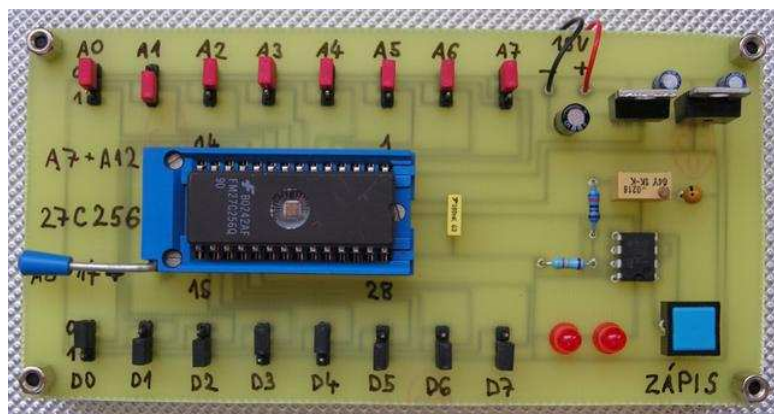
# Hrajeme si s EPROM

## Hrajeme si s EPROM - 1

První a několik příštích dílů našeho nového seriálu věnujeme podrobnému popisu konstrukce programátoru. Kromě nezbytného schématu zapojení, bude vše doplněno i plošným spojem, který vám snad napomůže pro jeho snadnější konstrukci.

Programátor pro naše pokusy – schéma zapojení

První a několik příštích dílů našeho nového seriálu, věnujeme podrobnému popisu konstrukce programátoru. Kromě nezbytného schématu zapojení, bude vše doplněno i plošným spojem, který Vám snad napomůže pro jeho snadnější konstrukci.



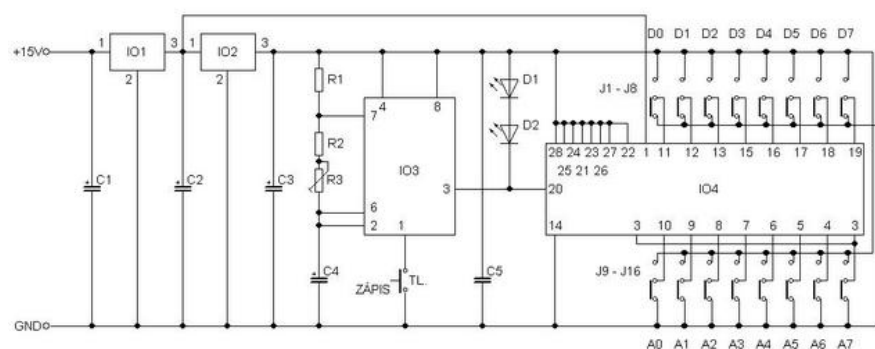
Celý tento cyklus, bude orientován pouze paměť 27C256. Setkali jsme se s ní již v předešlém seriálu. Z tohoto důvodu bude programátor uzpůsoben jen pro ni a je tak o poznání jednodušší, než ten, jehož konstrukci jsme si uvedli dříve. Pochopitelně, že lze použít i tento, pokud ho někdo má již zhotoven a není třeba vyrábět programátor další. Jedinou odlišností při použití předešlého programátoru bude trochu odlišný způsob zadávání adres paměti.

Oproti předchozímu typu, který sloužil i jako případná čtečka obsahu paměti, obsahuje v tomto případě zapojení jen nezbytně nutné součástky a je tak co nejjednodušší z hlediska konstrukce i používání.

Nejlépe nám ho přiblíží vlastní schéma zapojení. Začneme od zdrojové části. Pro chod celého kompletu je zapotřebí dvou napěťových úrovní 5V a 12V. Nižší slouží pro vlastní napájení paměti, která je plně kompatibilní s logikou TTL a dalších součástí, kdežto hodnota vyšší pro zaznamenání požadované informace na dané paměťové místo.

Externí zdroj musí dosahovat nejméně hodnoty 15V, s dovoleným proudovým odběrem minimálně 100mA.

S touto napěťovou hodnotou dále v našem zapojení pracují dva integrované stabilizátory. Jedná se o řadu 78xx. Konkrétně o 7812 a dále pak 7805. V obou případech jsou použity jednoampérové verze, které poskytují dostatečnou napájecí rezervu.



trvat po přesně stanovenou dobu a musí být aktivní v logické nule. Zdrojem tohoto programovacího impulsu, není nikdo jiný než oblibený časovač NE555, zapojený v astabilním řízeném režimu. Řízení je provedeno přes odpojení napájení, které je ovládáno tlačítkem značeným jako ZÁPIS. Po jeho stlačení, vyšle obvod programovací impuls do paměti. Tento úkon je indikován zabliknutím dvou LED diod, které jsou napojeny na jeho výstup.

Poslední a dá se říci, že i nejdůležitější částí zapojení je vlastní paměť EPROM. Jak bylo již naznačeno v úvodu, je zapojení o poznání jednodušší jak z hlediska konstrukce tak i vlastního programování. V praxi je to zařízeno tak, že je počet adresových vstupů zredukován na pouhých osm. Tedy stejný počet jako datových výstupů/vstupů. Zbylé adresové vstupy jsou trvale nastaveny na jednu z

První v pořadí je 7812, ten slouží jednak jako zdroj programovacího napětí pro paměť a také jako primární zdroj napětí pro stabilizátor druhý 7805, ze kterého napájíme zbytek zapojení.

Dalším blokem je vlastní časovač, který řídí zápis. Jak již víme z předešlého seriálu, je potřeba pro zaznamenání informace do paměti programovací impuls, který musí

logických hodnot. S touto úpravou tak musíme počítat i dále při konstrukci dalších zapojení, kde bude použita takto naprogramovaná paměť. Tento fakt nás však bude zajímat až v dalších dílech seriálu.

Řízení adresových a datových vstupů je tentokrát realizováno pomocí jednoduchých jumperových propojek. Před programováním, je tedy vždy potřeba nastavit každý z adresových i datových vstupů pomocí propojky která mu náleží na hodnotu, kterou bude vyžadovat zapojení ve kterém paměť použijeme. To se nijak neliší od programátoru předešlého.

Zvěrem si dnes ještě uvedeme seznam potřebných součástek, které budou pro konstrukci potřeba.

### Seznam součástek

IO1 – 7812/1A

IO2 – 7805/1A

IO3 – NE555

IO4 – EPROM 27C256, plus testovací patice

D1, 2 – LED 5mm, zelená, 2mA

C1 – 10uF/25V

C2, 3 – 10uF/16V

C4 – 10uF/16V, tantal

C5 – 100nF

R1 – 430k/0,6W

R2 – 6k2/0,6W

R3 – 1k/0,15W, trimr

TL. – tlačítko do DPS

J1 – J16 – jumperová lišta, 3pin. + propojky

- jednostranný kuprextit 100 x 160 mm

- distanční sloupky 4 x M3 x 10mm, plastové, závit IN/OUT + 4 x matička s podložkou

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu:

[cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: (14. 6. 2005)

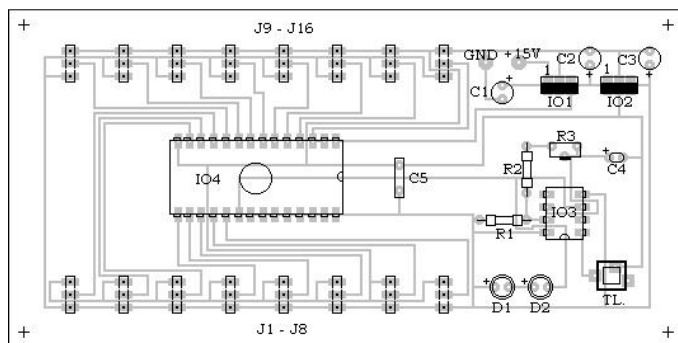
### Hrajeme si s EPROM 2

Dnes budeme pokračovat ve výrobě programátoru pro naše pokusy. Zhotovíme si desku plošného spoje, kterou také osadíme součástkami, jejichž rozpis jsme si uvedli na konci minulého dílu. Jak již víme, je plošný spoj také zároveň i nosnou konstrukcí celého zařízení. Pro jeho výrobu je použita jednostranná kuprextitová deska o rozměrech 140 x 70 mm.

Programátor pro naše pokusy – výroba a osazení plošného spoje

Dnes budeme pokračovat ve výrobě programátoru pro naše pokusy. Zhotovíme si desku plošného spoje, kterou také osadíme součástkami, jejichž rozpis jsme si uvedli na konci minulého dílu.

Jak již víme, je plošný spoj také zároveň i nosnou konstrukcí celého zařízení. Pro jeho výrobu je použita jednostranná kuprextitová deska o rozměrech 140 x 70 mm. V obchodě lze zakoupit desku s rozměry 100 x 160, kterou si upravíme na požadovaný rozměr. Desku již máme a můžeme přistoupit k přenesení obrazce plošného spoje. Ten se nachází na připojeném obrázku a je uveden v měřítku 1:1.

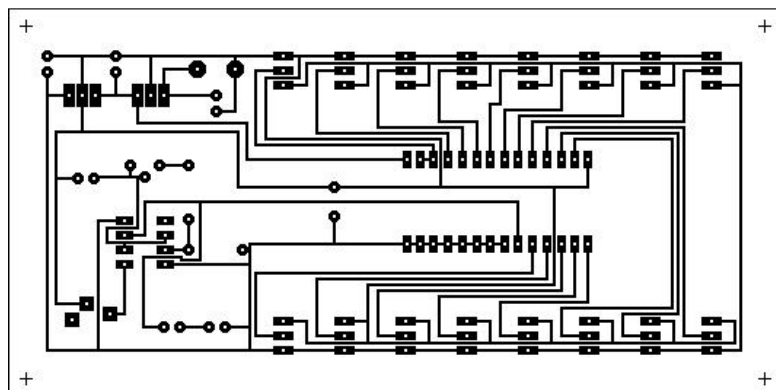


Způsobů přenosu předlohy je jak víme celá řada. Zde asi nejlépe vyhoví přenos fotocestou, nebo v krajním případě i prosté ruční překreslení.

Deska s přenesenou předlohou se dá klasicky vyleptat do roztoku chloridu železitého.

Vyleptaný spoj se po vytažení z lázně opláchne tekoucí vodou a vizuálně překontroluje. Je-li vše v pořádku, nebrání nic v dalším kroku, kterým je vyvrtání otvorů pro vývody součástek. Pro většinu postačí vrtáček o průměru 1mm, pro vývody integrovaných stabilizátorů pak o něco větší. V rozích plošného spoje jsou umístěny otvory pro zašroubování distančních sloupků, které slouží jako jednoduché nožičky. Zde je vhodné nejprve předvrtat otvory vrtákem menším a pak je rozšířit zhruba na průměr 3mm.

Vyvrtaný plošný spoj můžeme začít osazovat součástkami. Ještě je možné použít pro povrchovou úpravu vodivých cest, některou z bezproudých pokovovacích lázní (stříbrící lázeň). Před touto úpravou je však nutné důkladně očistit pokovovaný povrch.



Při vlastním osazování postupujeme postupně od těch nejmenších součástek, které se nacházejí nejbližší k plošnému spoji, až k těm větším. Jako poslední osadíme testovací patici, kterou lze nahradit i obyčejnou DIL paticí v precizním provedení.

Po osazení všech součástek a připojení napájecího napětí (15V), můžeme provést jednoduchý test. Po stlačení tlačítka nesoucí označení ZÁPIS, by mělo dojít k zabliknutí obou LED diod. Po jeho dalším stlačení, by se mělo vše opakovat.

Příště budeme pokračovat v konstrukci nastavením jednotlivých parametrů a začneme se učit programátor používat v praxi.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala

### Hrajeme si s EPROM - 3

**V dnešním pokračování našeho seriálu dokončíme konstrukci jednoduchého programátoru pro naše pokusy a zároveň si z části objasníme jak s ním zacházet v praxi. Programátor je již plně osazen součástkami a zbývá tedy jen nastavit čas programovacího impulsu. Tento krok se však dá i vynechat. Součástky jsou zvoleny tak, aby vše fungovalo i bez zdlouhavého nastavování, tedy doslova na první pokus.**

Programátor pro naše pokusy - nastavení programátoru

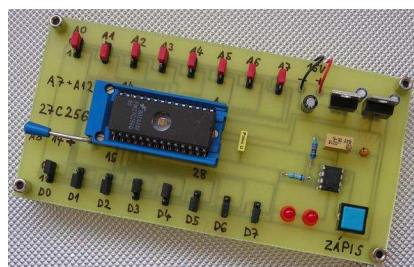
Doba trvání programovacího impulsu se dá ovlivnit přesným trimrem R3, kterým lze tuto dobu buď prodloužit, nebo naopak zkrátit.

Při nastavení do kterékoliv z krajních poloh by však programátor měl fungovat bez problémů. Výrobce pamětí, které budeme používat, tedy 27C256 uvádí ideální dobu trvání programovacího impulsu kolem 50 ms (milisekund). Ta je přibližně nastavena v prostřední poloze potenciometru, kde mezi jedním z jeho krajních vývodů a jezdcem hodnota odporu 500 Ohmů.

Toto je jediné nastavení, které lze a je potřeba na programátoru učinit. Nyní již můžeme přistoupit k vlastnímu principu programování.

Ten se nijak výrazně neliší od předešlého, který jsme si podrobněji popisovali v několika posledních dílech seriálu Číslicová technika.

I tentokrát bude potřeba nejprve nastavit adresu, na kterou budeme chtít zaznamenat požadovaná data. Dále pak vlastní data a posléze přivést programovací impuls, který zapíše daná data na nastavenou adresu.



Nastavení programovacích dat i adresových vstupů se provádí pomocí jednoduchých jumperových

propojek.

Celkem se jich na programátoru nachází šestnáct. Osm pro data a dalších osm pro adresy.

Počet adresových vstupů byl uměle snížen na pouhých osm. Zbylé adresové strupy se pevně nacházejí na stanovených logických úrovních. S tímto faktem je patřeba počítat při návrhu zapojení, ve kterém bude paměť naprogramovaná tímto programátorem figurovat.

Příště si řekneme pár slov o vlastní paměti EPROM, konkrétně o typu 27C256 a od dalšího dílu se již budeme věnovat jednotlivým pokusným zapojením.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala (13. 7. 2005)

## Hrajeme si s EPROM - 4

**Dnešní díl zasvětime podrobnému popisu parametrů paměti, kterou budeme používat v našich dalších zapojeních. Téma to nebude nijak nové, touto problematikou jsme se zabývali již v několika dílech předešlého seriálu Číslicová technika. Naznačíme si tedy jen pár základních parametrů a zopakujeme si klíč k určování parametrů daného integrovaného obvodu podle popisu, který se nachází na jeho těle.**

### EPROM 27C256

Jak bylo již několikrát naznačeno, bude se jednat o typ 27C256. Tedy o paralelní paměť typu EPROM, kompatibilní s technologií TTL. Celý obvod je umístěn v 28 pinnovém DIL pouzdru, které je na vrchu opatřeno průzorem pro mazání paměti pomocí UV světla. Princip mazání jsme si již také uvedli dříve, proto se k němu nebudeme vracet.

Začneme tedy rovnou dešifrováním parametrů paměti podle údajů uvedených na jejím pouzdru.

### Parametry paměti

Jak je patné z obrázku, lze označení rozdělit na tři části. První část definuje velikost paměťové matice, část druhá dobu odezvy a část třetí, rozmezí pracovních teplot.

Pro naše pokusy bude nejdůležitější část první. Tedy ta, která definuje velikost paměti. Budeme požadovat pouze typ 256. Typy nesoucí 512, nebo 128 jsou neslučitelné s naším programátorem a proto je nemůžeme ani používat.

Druhá část označuje dobu odezvy paměti, tedy čas za který se po nastavení dané adresy objeví data na výstupech paměti. Pro naše budoucí pokusná zapojení vystačí i klidně verze nejpomalejší s odezvou 200ns (nanosekund).

Rozsah pracovních teplot pro nás také není důležitý. Pro běžný provoz vyhovují všechny varianty.

### Zápis a čtení

I zde se bude jednat o opakování. Logické úrovně jednotlivých vstupů pro módy čtení - Read, zápisu - Program, nebo stavu s nízkým odběrem proudu - Standby, si opět doplníme o názorné grafy, které budou v časových úsecích vždy znázorňovat logické úrovně jednotlivých nastavovacích vstupů.

Jednotlivá vyobrazení obsahují tyto položky: Adress - adresové vstupy, Outputs - vstupy/výstupy informací, to podle toho zda se jedná o zápis, nebo čtení a dále pak nastavovací stupy CE a OE. Graf pak ukazuje, zda se daný kontakt nachází ve stavu VIH, nebo VIL (logická jednička, nebo logická nula).

U grafu pro zápis pak ještě nalezneme vstup VPP, který se při programování musí nalézat na napěťové úrovni 13 V a při čtení na 5 V (logická jednička).

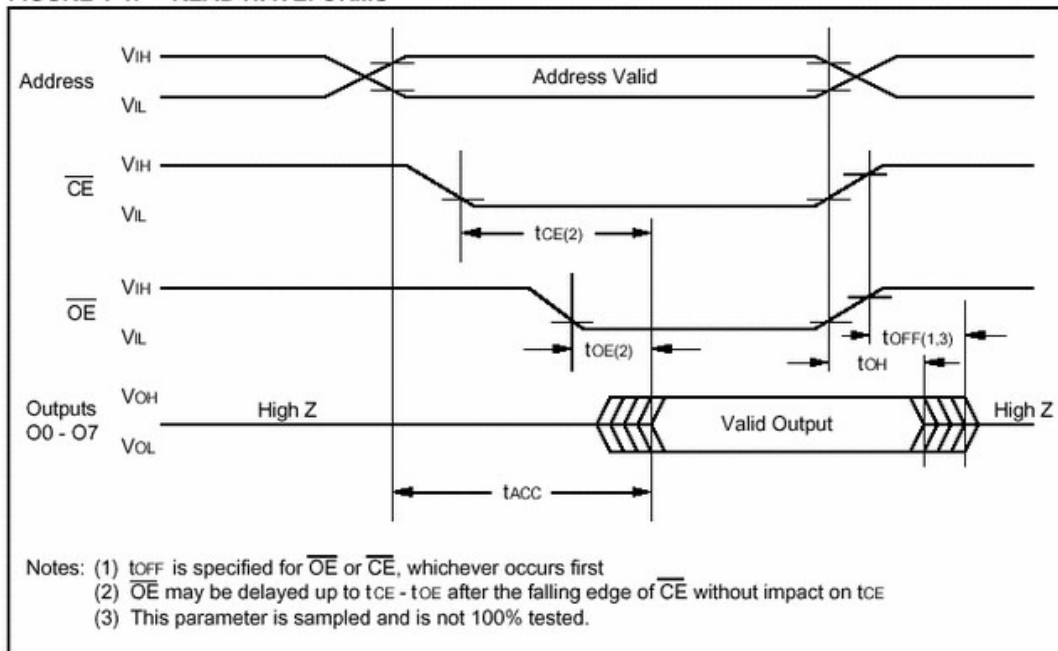
Nejlépe bude začít režimem čtení (READ WAVEFORMS). Budeme předpokládat, že v paměti jsou již nějaká data zaznamenána.

Prvním krokem je nastavení patřičné adresy. Tu zvolíme kombinací logických úrovní na adresových vstupech. Adresa je tedy nastavená a zbývá pouze přivést i určené logické úrovně k negovaným nastavovacím vstupům OE a CE. Oba se musí nalézat v logické nule.

Po tomto úkonu se na datových výstupech (outputs) objeví data zaznamenaná na zvolené adrese.

Druhým módem je režim zápisu (PROGRAMMING WAVEFORMS).

FIGURE 1-1: READ WAVEFORMS



Zde je situace o něco komplikovanější, ale není to zas tak strašné. Prvním krokem je opět nastavení adresy, kam ovšem tentokrát chceme data zapsat.

Druhým krokem je třeba nastavit data, která chceme zaznamenat. To se provede přivedeným patřičných logických úrovní k datovým vstupům. Ty slouží při čtení i jako datové výstupy.

Nyní přijde na řadu vstup VPP, zde je nutné zvýšit napětí z 5 V na 13 V. Předposledním krokem je nastavení negovaného vstupu OE do logické jedničky.

Negovaný vstup CE, který zbyl, slouží pro přivedení programovacího impulsu. Ten jak již víme, musí být aktivní v logické nule a trvat přesně 50ns (45).

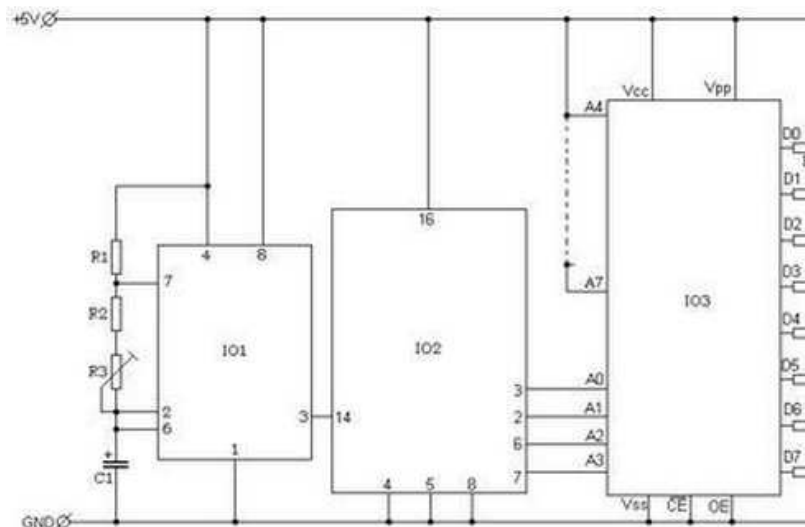
Dnes jsme si tedy zopakovali základní parametry paměti EPROM 27C256 a od příště se budeme věnovat již konkrétním konstrukcím, které budou obsahovat i předlohy plošných spojů.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislivotechnika@volny.cz](mailto:cislivotechnika@volny.cz)

Autor: Jindřich Fiala

Datum: 13. července 2005

### Hrajeme si s EPROM - 5



Dnešním dílem zahájíme řadu několika pokusných zapojení, ve kterých bude figurovat paměť EPROM naprogramovaná naším programátorem a u kterých budou kromě programů uveřejněny i plošné spoje.

## Jednoduché běžící světlo

První zapojení nebude nijak složité. Dalo by se říci, že je až triviální, ale pro začátek je to právě to pravé. Pokud jste pečlivě sledovali předešlý seriál Číslicová technika, bude se vám možná zdát schéma zapojení poněkud povědomé. Pro první pokus je použito trochu upravené zapojení uvedené v 99. dílu seriálu.

Dnes si uvedeme schéma zapojení se seznamem součástek. Příště bude následovat již vlastní výroba, programování a hlavně předloha plošného spoje.

Nyní už k vlastnímu zapojení. Předvedeme si výrobek napodobující efekt běžícího světla, které přebíhá přes střed ze strany na stranu. K výstupům paměti je zde připojeno osm LED diod, které se postupně rozsvěčí a zase zhasínají podle programu zaznamenaném v paměti. Postup při programování a hlavně program si blíže přiblížíme až v příštím dílu, dnes nás tedy bude zajímat jen vlastní princip zapojení

Za řídicí prvek celého zapojení, by se dal označit obvod NE555, který vysílá řídicí hodinové impulsy do BCD čítače, který ovládá adresové vstupy paměti a který jak víme, dokáže na svých čtyřech výstupech zobrazit deset rozličných binárních informací.

Z toho tedy jasně vyplývá, že můžeme ovládat deset samostatných paměťových adres, do kterých lze vždy zaznamenat po osmibitové binární informaci.

Při pohledu na datovou část tabulky, je na první pohled patrný motiv kříže sestavený z jedniček. Ty, jak víme, představují svítící LED diodu. Pokud budeme postupovat po jednotlivých řádcích, zjistíme, že se poloha jedniček postupně mění. Od krajních poloh se postupně přibližují s každým dalším řádkem ke středu a po dosažení středu se zas navrací zpět. Výsledkem je efekt světla, které přebíhá z obou stran přes střed a zase zpět. To jsme ale poněkud odbočili.

Nás bude dnes zajímat ještě onen obvod NE555. Jedná se o velice oblíbený časovač, který dokáže pracovat v několika režimech a který je také schopen řídit počet impulsů, které vysílá. V našem případě se řízení výstupní hodinové frekvence, kterou časovač generuje, dá ovlivnit trimrem R3. S jeho pomocí můžeme ovlivňovat rychlost, s jakou se bude světlo pohybovat. Čím bude tato frekvence větší, tím se také bude světlo pohybovat rychleji.

Dalším prvkem odlišností je naše paměť EPROM. Díky jednoduššímu programátoru, nesmíme zapomenout přivést nepoužité adresové vstupy na dané logické úrovni, které zaznamená programátor do paměti při programování.

Příště si již uvedeme předlohu plošného spoje pro dnešní zapojení spolu s programem.

	ADRESA				DATA							
	A0	A1	A2	A3	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	0	0	1	0	0	0	0	0	0	1
1	1	0	0	0	0	1	0	0	0	0	1	0
2	0	1	0	0	0	0	1	0	0	1	0	0
3	1	1	0	0	0	0	0	1	1	0	0	0
4	0	0	1	0	0	0	1	0	0	1	0	0
5	1	0	1	0	0	1	0	0	0	0	1	0
6	0	1	1	0	1	0	0	0	0	0	0	1
7	1	1	1	0	0	0	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	0	0	0	0

### Seznam součástek:

IO1 – NE555

IO2 – 74HC190

IO3 – EPROM 27C256

LED – 8 x LED, 5mm,

T – 8 x BC546

C1 – 1uF/16V

R1 – 27k/0,6W

R2 – 1k/0,6W

R3 – trimr, 1M/0,15W

Rt – 8 x 30k/0,6W

Rd – 8 x 120R/0,6W

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala

Datum: 13. července 2005

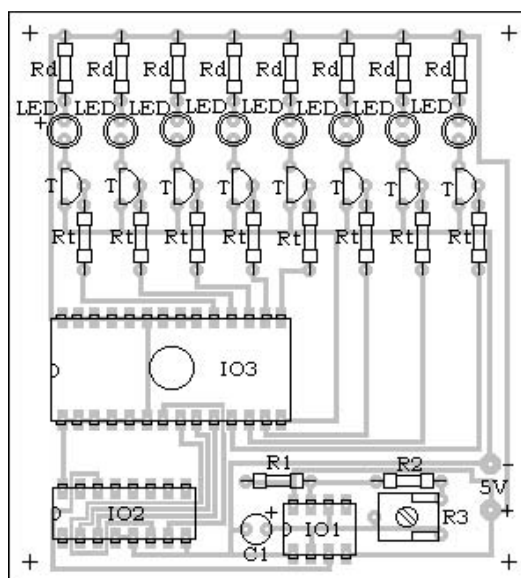
### Hrajeme si s EPROM - 6

Jak bylo řečeno na konci minulého dílu, uvedeme si dnes předlohu plošného spoje spolu s programem, který je třeba zaznamenat do paměti. Jistý náznak jsme si uvedli již minule.

Jednoduché běžící světlo – dokončení konstrukce

Plošný spoj není nijak výrazně složitý a s trochou zručnosti a pečlivosti by ho neměl být problém zhotovit.

Celé zařízení je postaveno na jednostranné cuprextitové desce, která je v rozích, stejně jako programátor, opatřena jednoduchými nožičkami v podobě distančních sloupků.



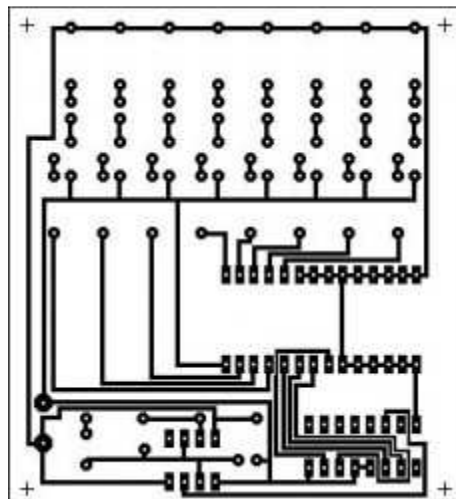
Postup výroby, přenesení předlohy na cuprextitovou desku, se dá jako vždy provést několika způsoby. Díky poměrně jednoduchému zapojení ho tedy lze překreslit i ručně.

Dále se tradičně spoj vyleptá a vyvrtá. Případně zastříhne na patřičný rozměr. Ten dnes činí 78 x 85 mm.

Při osazování postupujeme od těch nejmenších součástek, které se nacházejí nejbližší desce k těm největším. Pro umístění paměti, použijeme buď testovací patici s nulovou silou, nebo spíše obyčejnou 28 pinnovou DIL patici v precizním provedení.

Po odzkoušení, se vodivé cesty přetřou ochranným lakem na osazené plošné spoje. To však předbýváme. Nejprve je nutné naprogramovat paměť.

### Programování



Princip programování a práci s naším programátorem jsme si již vysvětlovali. Nejpodrobněji se tomuto tématu věnuje 3. díl seriálu. Nás tedy bude spíše zajímat vlastní program zaznamenaný v binární podobě. Vše je uvedeno v tabulce. Levá část je věnována adresám, část pravá zaznamenaným datům.

U adresové části ještě musíme připomenout, že jednak je tato část minimalizována v programátoru na pouhých osm adresových vstupů a dále pak v našem zapojení na pouhé čtyři.

S minimalizací, kterou obstarává programátor, si nemusíme lámat hlavu. Ta je již zakomponována do plošného spoje pokusného zapojení. Nás spíše bude zajímat minimalizace druhá. A sice ta, která zužuje počet adresových vstupů na pouhé čtyři. S tímto faktem je třeba počítat při programování. Vše je zajištěno tak, že nepotřebné adresové vstupy pro toto zapojení jsou nastaveny na logické jedničky. Při programování je třeba zaznamenat na tyto vstupy právě logické jedničky. Zařízení by jinak nefungovalo.

	ADRESA				DATA							
	A0	A1	A2	A3	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	0	0	1	0	0	0	0	0	0	1
1	1	0	0	0	0	1	0	0	0	0	1	0
2	0	1	0	0	0	0	1	0	0	1	0	0
3	1	1	0	0	0	0	0	1	1	0	0	0
4	0	0	1	0	0	0	1	0	0	1	0	0
5	1	0	1	0	0	1	0	0	0	0	1	0
6	0	1	1	0	1	0	0	0	0	0	0	1
7	1	1	1	0	0	0	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	0	0	0	0

Dále je programování naprosto rutinní záležitostí.

Nejprve se nastaví adresa, dále data a pak se vyšle do paměti programovací impuls, který zaznamená nastavená data do dané adresy.

Princip efektu je patrný z tabulky. Světlo přeběhne přes střed z jedné strany na druhou a po tomto ději nastane malá odmlka, po které se bude vše opakovat. Rychlost posunu se dá ovlivnit pomocí trimru R3.

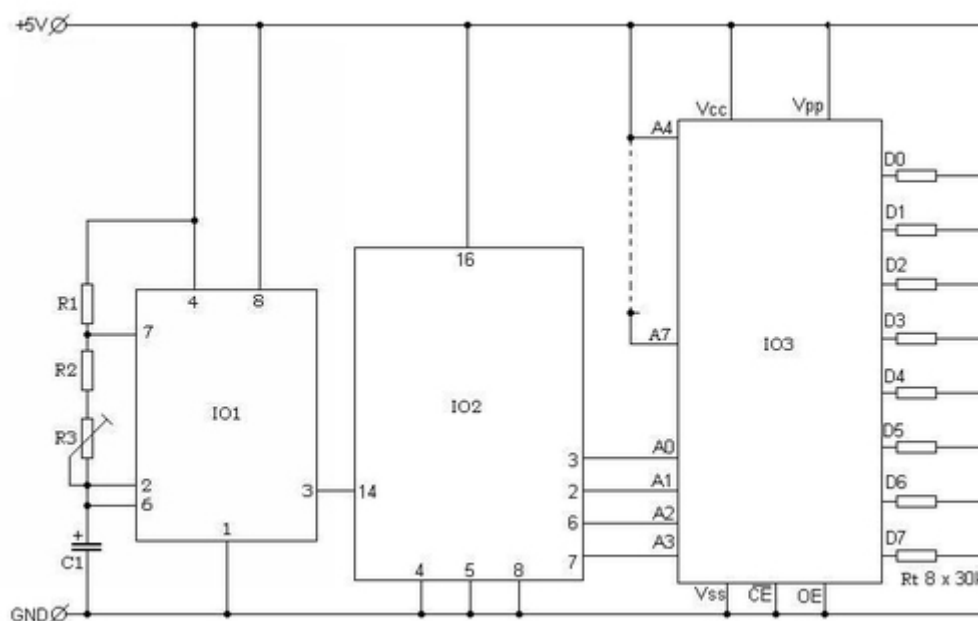
Příště se budeme teoreticky věnovat zapojení dalšímu, které bude již o něco složitější.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala

Datum: 21. července 2005

## Hrajeme si s EPROM - 7





**Dnes se budeme věnovat již druhému pokusnému zapojení. Cílem našeho pokusu, bude sestavit jednoduchý dekodér BCD kódu, pro buzení sedmissegmentového LED displeje. Srdcem celého zapojení bude opět svépomocí naprogramovaná paměť EPROM.**

### **Jednoduchý dekodér BCD/7.seg.**

Princip takového zapojení není nijak složitý a něčemu podobnému jsme se již věnovali v předešlém seriálu Číslicová technika.

Malé opakování však nikdy neuškodí. Úkolem dekodéru, jak již napovídá jeho název, je dekodování přivedené informace a její další zpracování do podoby, kterou je již člověk schopen pochopit a s kterou může dále pracovat. V našem konkrétním případě půjde o dekodování dekadického čísla zakódovaného do binární podoby a jeho zobrazení na LED displeji.

S tímto jednoduchým požadavkem se můžeme v číslicové technice setkat velice často. Způsoby provedení jsou však rozličné.

Ten, který si uvedeme, nebude patřit k těm nejjednodušším, protože nejjednodušší by bylo použití obyčejného integrovaného obvodu, který by v sobě obsahoval dekodér. Nám půjde spíše jen o to, abychom si uvedli další příklad z praxe spolu s plošným spojem, kde je možné takovou paměť EPROM s jednoduchým programem použít.

Zapojení se dá rozdělit na čtyři hlavní celky. Generátor impulsu, BCD čítač, dekodér v podobě naprogramované EPROM a LED displej. Zdroj napětí by se dal označit jako část pátá, tu však podrobněji řešit nebudeme.

Generátor impulsu je tvořen obyčejným časovačem NE555 zapojeném v astabilním režimu. Schéma i použité součástky jsou shodné s předchozím zapojením.

Další v pořadí je BCD čítač. Ten čítá impulsy přivedené od generátoru a zobrazuje jejich počet v zakódované podobě na svých výstupech.

Třetím v pořadí je součást nejdůležitější. Paměť EPROM. Tu zde máme v roli již zmiňovaného dekodéru.

I v tomto případě stejně jako v předešlém zapojení, je počet adresových vstupů snížen na pouhé čtyři.

K těmto vstupům přivádíme zakódovaná data od BCD čítače. K výstupům paměti je naopak připojen LED displej, který zobrazuje dekodovanou informaci.

O dekodování se postará program uložený v paměti. Princip je velice jednoduchý. Každému číslu, které je zakódováno do binární podoby je určena jedna adresa. Na té je uložena informace pro LED displej, který ze segmentů se má rozsvítit. Vše je jako vždy uvedeno v tabulce.

Příště si uvedeme již vlastní program, spolu s předlohou plošného spoje.

### **Seznam součástek:**

IO1 – NE555

IO2 – 74190

IO3 – 27C256

T – 8 × BC546B

C1 – 1uF/16V

R1 – 27k/0,6W

R2 – 1k/0,6W

R3 – trimr, 1M/0,15W

Rt – 8 × 30k/0,6W

Rd – 8 × 120R/0,6W

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Filip Janovský

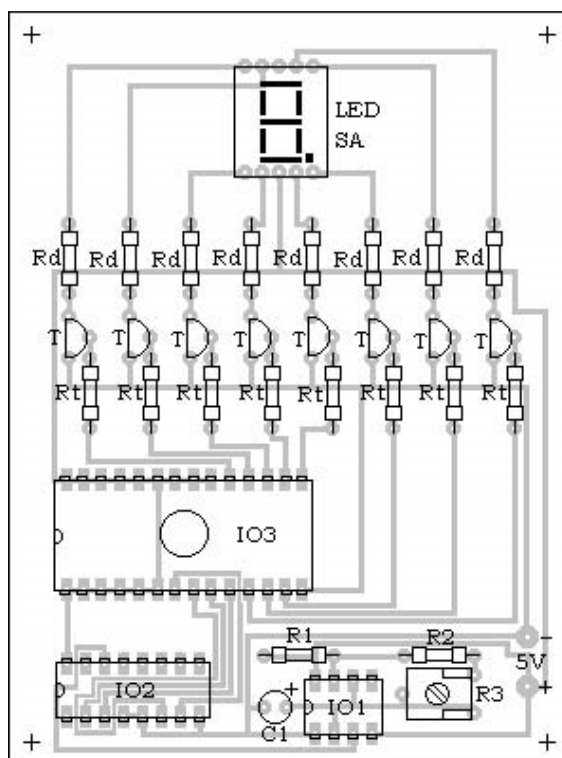
Datum: 26. července 2005

### **Hrajeme si s EPROM - 8**

**Dnes si uvedeme předlohu plošného spoje, spolu s vlastním programem pro zapojení dekodéru BCD/7. seg. z minulého dílu.**

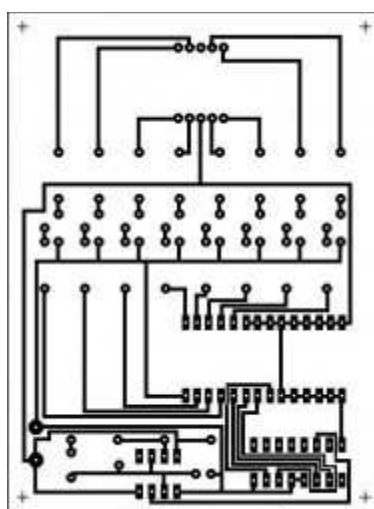
## Jednoduchý dekodér BCD/7.seg.– dokončení konstrukce

Plošný spoj není opět nijak výrazně složitý a s trochou zručnosti a pečlivosti by ho neměl být problém zhotovit. V podstatě je z větší části shodný jako u konstrukce předešlé. Liší se pouze v zobrazovací části. Zde nejsou použity již LED diody, ale sedmissegmentový LED displej, se společnou anodou.



Zapojení je postaveno na jednostranné cuprexitové desce, která je v rozích opatřena jednoduchými nožičkami v podobě distančních sloupků.

Předlohu spoje tedy máme a nebrání nám nic v tom, začít s vlastní výrobou. Postup přenesení předlohy na cuprexitovou desku, se dá jako vždy provést několika způsoby. Díky poměrně jednoduchému zapojení, ho lze překreslit i ručně. Ideální je jak víme však cesta fotoleptání.



Cuprexit s přenesenou předlohou se tradičně vyleptá, po té se vyvrtá a osadí součástkami. Rozměry dnešní konstrukce činí  $78 \times 114$ mm.

Při osazování postupujeme od těch nejmenších součástek, které se nacházejí nejbližší desce k těm největším. Pro umístění paměti, použijeme buď testovací patici s nulovou silou, nebo spíše obyčejnou 28 pinnovou DIL patici v precizním provedení.

Po odzkoušení, se vodivé cesty přetřou ochranným lakem na osazené plošné spoje. To však předbýváme. Nejprve je nutné zapsat do paměti požadovaný program. Ten nalezneme opět v tabulce.

## Programování

KROK	ADRESA				DATA							
	A0	A1	A2	A3	D0	D1	D2	D3	D4	D5	D6	D7
					a	b	dp	c	d	e	f	g
0.	0	0	0	0	1	1	1	1	1	1	1	0
1.	1	0	0	0	1	1	0	0	0	0	0	0
2.	0	1	0	0	1	0	1	1	1	0	1	1
3.	1	1	0	0	1	1	0	1	0	0	1	1
4.	0	0	1	0	1	1	1	0	0	1	0	1
5.	1	0	1	0	0	1	0	1	0	1	1	1
6.	0	1	1	0	0	1	1	1	1	1	1	1
7.	1	1	1	0	1	1	0	0	0	0	1	0
8.	0	0	0	1	1	1	1	1	1	1	1	1
9.	1	0	0	1	1	1	0	1	0	1	1	1

Popisu minimalizace adresové části, se již věnovat nebudeme. Zaměříme se rovnou na vlastní data. Programování je naprosto rutinní záležitostí.

Nejprve se nastaví adresa, dále data, po kterých vyšleme do paměti programovací impuls, který zaznamená nastavená data do dané adresy. To je ve zkratce vše podstatné.

Princip činnosti dekodéru je pak patrný z tabulky. Po přivedení napájecího napětí, se začnou na LED displeji objevovat číslice 0 až 9. Rychlost jejich posunu ovlivníme pomocí trimru R3.

Sled událostí je následující. Generátor (NE555) vysílá impulsy do BCD čítače, který je napojen na adresové vstupy paměti. Na výstupech čítače se objevují zakódované číslice 0 až 9 v BCD kódu. Ten paměť dekoduje a dále pak rozsvěcí příslušné segmenty LED displeje pro zobrazení přivedeného BCD čísla v dekadické podobě, která je již zcela srozumitelná široké veřejnosti. Díky automatickému nulování čítače, se sled číslic neustále opakuje. Pokud by se například vynechal generátor a na vstup čítače by se napojil spínač, mohlo by zařízení sloužit k postupnému čítání impulsů v rozmezí 0 až 9.

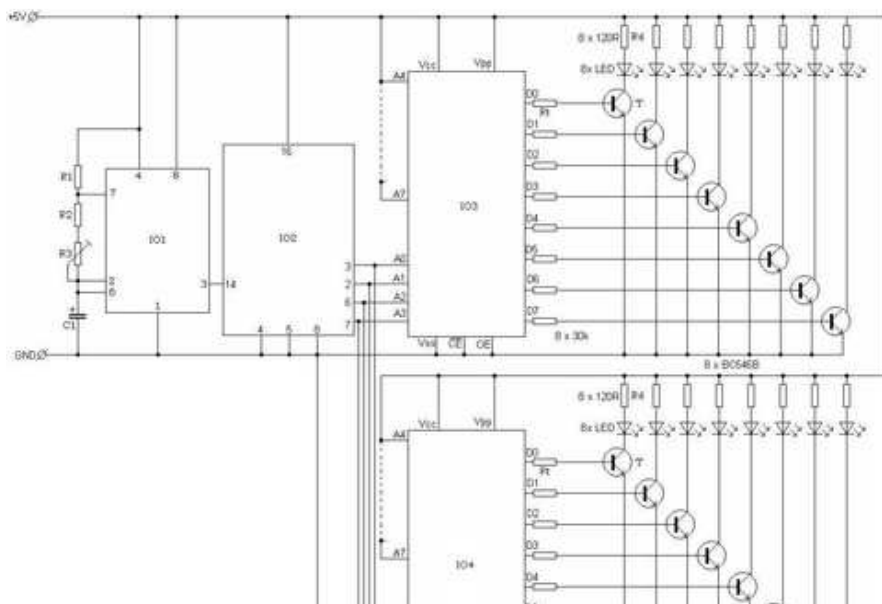
Příští díl věnujeme již zapojení dalšímu, které bude opět o něco složitější, než konstrukce dnešní.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala

Datum: 2. srpna 2005

## Hrajeme si s EPROM - 9



**Dnešní pokračování našeho seriálu, zasvětime dalšímu pokusnému zapojení. Tentokrát v něm však budou místo jedné, figurovat již dvě paměti EPROM. Nepůjde však o nic až tak výrazně složitého.**

### **Světelné efekty se dvěma paralelně řízenými EPROM**

Paměti budou mít totiž společné adresové vstupy a lišit se budou tedy jen informace, které zaznamenané do každé z pamětí.

Schéma si opět můžeme rozdělit na několik částí. Přesněji řečeno na tři. Každou však již velice dobře známe.

Princip generátoru, čítače i vlastních pamětí jsme si již podrobně popsali v předešlých dílech seriálu.

Použití dvou samostatných pamětí, které mají společné řízení adresových vstupů, nám v našem konkrétním způsobu propojení umožní zaznamenat dvojnásobek binárních informací při stejném počtu adres. Toho můžeme využít pro již zmiňované světelné efekty, které lze tímto zajímavě obohatit. Použitím vhodného programu do každé z pamětí, společně s vhodným rozmístěním LED diod, lze dosáhnout opravdu zajímavých efektů. Například i běžícího textu. Tomu se však zatím věnovat nebudeme. Nám postačí jednoduché efekty, které nám spíše ukáží, jak lze paměti EPROM hojně využívat pro různá zapojení.

Příště si tradičně uvedeme již vlastní program pro každou z pamětí, společně s předlohou plošného spoje.

#### *Seznam součástek:*

IO1 – NE555

IO2 – 74HC190

IO3, 04 – EPROM 27C256

LED – 16 x LED, 5mm,

T – 16 x BC546

C1 – 1uF/16V

R1 – 27k/0,6W

R2 – 1k/0,6W

R3 – trimr, 1M/0,15W

Rt – 16 x 30k/0,6W

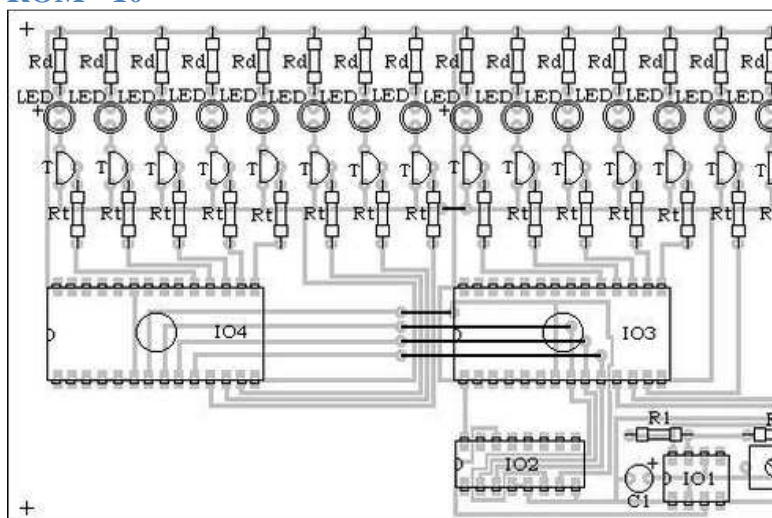
Rd – 16 x 120R/0,6W

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala

Datum: 9. srpna 2005

### **Hrajeme si s EPROM - 10**



**Dnes dokončíme započatou konstrukci z minula. Jednalo se o již složitější zapojení, ve kterém se nacházejí již dvě samostatné paměti EPROM 27C256, se společným řízením adresových vstupů.**

Světelné efekty se dvěma paralelně řízenými EPROM – dokončení konstrukce.

Výrobu plošného spoje i vlastní postup při programování je již zcela zbytečné popisovat, proto se budeme věnovat rovnou využití dnešního zapojení v praxi. Pokud by si však někdo v těchto částech nebyl



Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala

Datum: 16. srpna 2005

### Hrajeme si s EPROM - 11

**Stejně jako i v minulém pokračování našeho seriálu, budeme se dnes opět věnovat zapojení, které bude obsahovat dvě paralelně řízené paměti EPROM.**

11. Světelné efekty se dvěma paralelně řízenými EPROM – II.

Při podrobném prozkoumání schématu zapojení zjistíte, že je naprosto totožné s předchozím zapojením. Jediné co se tedy bude v této konstrukci měnit, bude předloha plošného spoje a program. U předlohy plošného spoje, se bude změna týkat části obsahující LED diody. Půjde hlavně o jejich uspořádání. Princip propojení na výstupy paměti a tedy i jejich řízení, zůstane naprosto stejný.

U předchozí konstrukce byly diody umístěny v jedné řadě. To do jisté míry omezovalo jejich použití pro různá zapojení, jiná než běžícího světla a různých světelných efektu, které se odehrávají v řadě.

Dnes proto řadu "ohneme". LED diody tentokrát poskládáme do dvou soustředných kruhů. Kolem jednoho menšího se bude nacházet kruh se stejným počtem LED diod, jen s větším průměrem a tedy i většími rozestupy mezi jednotlivými ledkami.

Program bude v modelové situaci nastaven tak, že se proti sobě budou kruhy jakoby otáčet. Oproti jednoduché řadě běžícího světla, jistě příjemná obměna. Vyzkoušíme si tak další možnost použití pamětí EPROM.

Příště si již tradičně uvedeme předlohu plošného spoje, spolu s programem pro každou z pamětí.

Seznam součástek:

IO1 – NE555

IO2 – 74HC190

IO3, 04 – EPROM 27C256

LED – 16 x LED, 5mm,

T – 16 x BC546

C1 – 1uF/16V

R1 – 27k/0,6W

R2 – 1k/0,6W

R3 – trimr, 1M/0,15W

Rt – 16 x 30k/0,6W

Rd – 16 x 120R/0,6W

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala

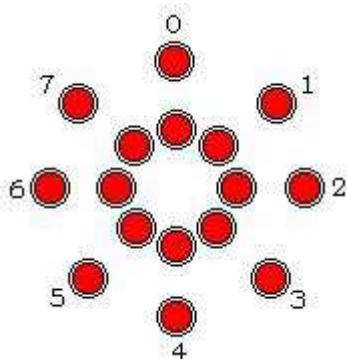
Datum: 24. srpna 2005

### Hrajeme si s EPROM 12

**Dnes budeme pokračovat v konstrukci započaté již minule. Předmětem našeho zájmu se staly skupiny LED diod umístěné do dvou soustředných kruhů.**

**Světelné efekty se dvěma paralelně řízenými EPROM – II, dokončení konstrukce.**

Jak jsme si uvedli, řízení tohoto seskupení obstarávají dvě paralelně řízené paměti EPROM. Programu se však budeme věnovat až za chvíli.



Konstrukční provedení celého zapojení se dá provést několika způsoby. Od úplně nové předlohy plošného spoje, kde by byly diody uspořádány do kruhů, až po cestu uspořádání diod na samostatném čelním panelu.

Variantu druhou zvolíme dnes. Předloha plošného spoje stejně jako i vlastní schémata zůstává totožné od minula.



Program je vzorově sestaven tak, aby se kruhu pootáčely proti sobě a to rychlostí jakou nastavíme pomocí časovače NE555.

Po oběhnutí celé trasy oba kruhy střídavě problíknou a celý efekt se opět opakuje.

Své případné podněty a dotazy týkající se seriálu můžete zasílat na adresu: [cislicovatechnika@volny.cz](mailto:cislicovatechnika@volny.cz)

Autor: Jindřich Fiala

Datum: 1. září 2005

*Zdroj: [Hrajeme si s EPROM – kurz ABC](#)*

<http://abc.blesk.cz/clanek/serialy/6292/hrajeme-si-s-eprom-2.html>

<http://abc.blesk.cz/clanek/serialy/6325/hrajeme-si-s-eprom-4.html>

<http://abc.blesk.cz/clanek/serialy/6326/hrajeme-si-s-eprom-5.html>

<http://abc.blesk.cz/clanek/serialy/6415/hrajeme-si-s-eprom-6.html>

<http://abc.blesk.cz/clanek/serialy/6416/hrajeme-si-s-eprom-7.html>

<http://abc.blesk.cz/clanek/serialy/6456/hrajeme-si-s-eprom-8.html>

<http://abc.blesk.cz/clanek/serialy/6467/hrajeme-si-s-eprom-9.html>

<http://abc.blesk.cz/clanek/serialy/6495/hrajeme-si-s-eprom-10.html>

<http://abc.blesk.cz/clanek/serialy/6506/hrajeme-si-s-eprom-11.html>

<http://abc.blesk.cz/clanek/serialy/6539/hrajeme-si-s-eprom-12.html>