

# Vážení zákazníci,

dovolujeme si Vás upozornit, že na tuto ukázkou knihy se vztahují autorská práva, tzv. copyright.

To znamená, že ukáзка má sloužit výhradně pro osobní potřebu potenciálního kupujícího (aby čtenář viděl, jakým způsobem je titul zpracován a mohl se také podle tohoto, jako jednoho z parametrů, rozhodnout, zda titul koupí či ne).

Z toho vyplývá, že není dovoleno tuto ukázkou jakýmkoliv způsobem dále šířit, veřejně či neveřejně např. umístováním na datová média, na jiné internetové stránky (ani prostřednictvím odkazů) apod.

*redakce nakladatelství BEN – technická literatura*  
[redakce@ben.cz](mailto:redakce@ben.cz)



V této kapitole budou popsány základní aplikace portů P0 až P3. Jedná se o připojení osmi LED, osmi spínačů, šestnácti LED a spínačů (řízení sériovou sběrnici).

Další aplikace naleznete v [1] a v následujících kapitolách.

## 7.1 VNITŘNÍ ZAPOJENÍ PORTŮ

Než se seznámíme s prvním z přípravků, je třeba podat informaci o vnitřním zapojení portu.

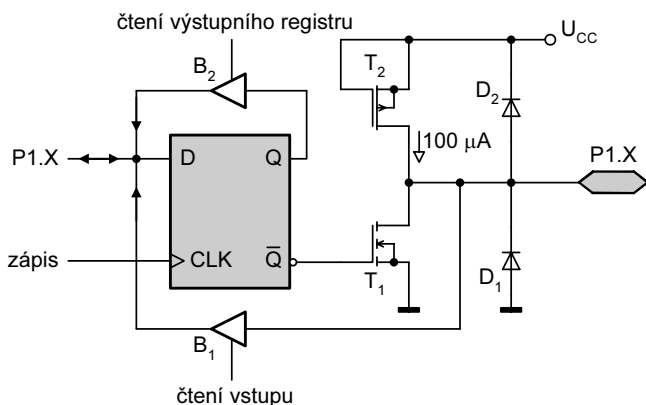
Pokud nebudeme uvažovat alternativní funkci některých vývodů, lze říci, že se porty P1 až P3 chovají vnitřně shodně. Vnitřní zapojení jednoho bitu ukazuje obr. 7.1. Každý bit je vybaven záchytným registrem (klopným obvodem D).

Port P0 se odlišuje skutečností, že nemá zabudovány zvyšovací odpory (pull-upy). Takže zde chybí tranzistor  $T_2$ , který funkci zvyšovacího odporu zastupuje.

Schéma dle obr. 7.1 platí pro port P1. Po resetu jsou všechny porty nastaveny na 1111111B. Protože je hradlo tranzistoru  $T_1$  připojeno na invertovaný výstup registru, je  $T_1$  rozpojen. Tranzistor  $T_2$  má kanál zabudovaný (vede již při nulovém napětí  $U_{GE}$ ) a slouží pouze jako zdroj proudu  $100 \mu\text{A}$  (můžeme si jej zjednodušeně představit jako odpor  $50 \text{ k}\Omega$ ), takto je realizován pull-up (zvyšovací) rezistor. Po resetu je tedy na vývod P1.X přivedena log. 1 přes vysokoimpedanční pull-up. Vnější obvody mohou takto „slabou“ log. 1 „přetáhnout“. To umožní čtení stavu vývodu P1.X přes budič  $B_1$ . Vývod se chová jako vstup.

Po zápisu log. 0 na daný bit, je na hradlo  $T_1$  přivedena log. 1, která jej sepne. Nyní je vývod P1.X připojen přes sepnutý tranzistor  $T_1$  na log. 0 (totemem teče stále příčný proud z proudového zdroje  $T_2$ ). Vývod P1.X se chová jako výstup s log. 0.

Diody  $D_1$  a  $D_2$  chrání vývod před vstupním napětím vyšším než  $U_{CC} + 0,7 \text{ V}$  nebo nižším než  $-0,7 \text{ V}$ . Prakticky je jim třeba předřadit rezistor o velikosti odporu asi  $1 \text{ k}\Omega$ .

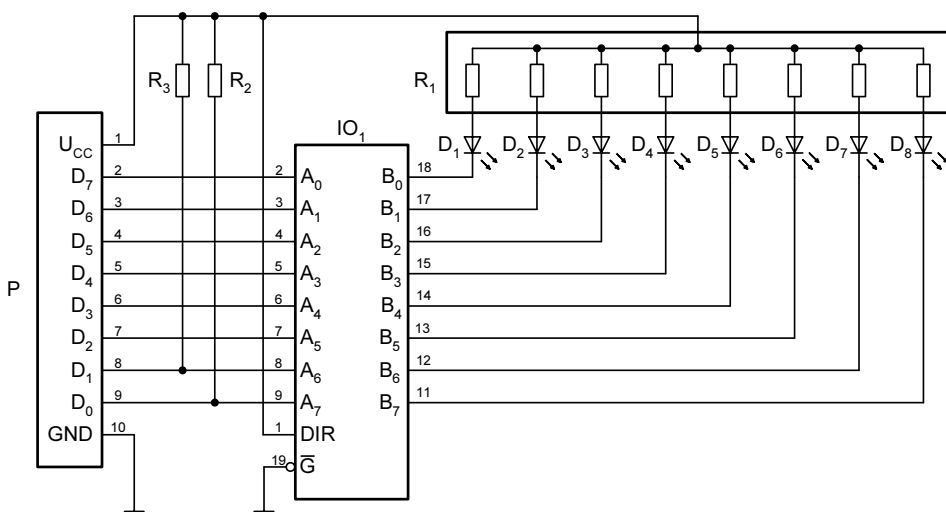


Obr. 7.1 Vnitřní zapojení jednoho bitu portu P0 až P3 (pro port P0 je tranzistor  $T_2$  vynechán)

Shrňme nyní chování paralelního portu:

- vývod se chová jako *vstup*, pokud je na daný bit zapsána log. 1 (při čtení log. 0 musí být vnějším obvodem odveden proud 100  $\mu$ A; tuto podmínku splní výstupy TTL, CMOS 4000 i HC/HCT obvodů),
- po resetu jsou všechny vývody konfigurovány jako vstupní (všechny bity mají hodnotu log. 1),
- pokud na daný bit zapíšeme log. 0, je vývod tažen poměrně „tvrdě“ k napětí 0 V (na vývodu je log. 0). Z jednoho vývodu lze maximálně odebrat proud 10 mA. Součet proudů jednotlivých vývodů portů P1, P2 a P3 je však pouze 15 mA (pro port P0 pak 26 mA). Celkově lze ze všech vývodů portů P0 až P3 (dohromady) odebrat maximálně 71 mA (viz kapitolu 2),
- zátěž je vhodné budít proti  $U_{CC}$ , protože pak lze odebrat proud až 10 mA (při buzení proti 0 V je možno odebrat proud maximálně 100  $\mu$ A),
- instrukce typu čtení-modifikace-zápis (read-modify-write, viz kapitolu 5.11) nečtou stavy vstupů, ale obsah registru přes budič  $B_2$ ,
- porty jsou bitově adresovatelné, takže je možno zapisovat/číst hodnotu libovolného vývodu nezávisle na ostatních vývodech,
- vývody portu P0 nemají zabudovaný pull-up (chybí tranzistor  $T_2$ ). Zde je vhodné připojit vnější pull-up,
- porty P0 a P2 se také používají pro připojení vnější paměti.

## 7.2 AT8LEDR – BUZENÍ OSMI LED



Obr. 7.2 Schéma zapojení přípravku AT8LEDR

Nejjednodušší aplikací je patrně připojení osmi LED na paralelní port. Přímé buzení LED však není možné, protože pak by každou LED mohl téci proud maximálně 2 mA (svítila by pak poměrně slabě).

**POZNÁMKA:** Přímé buzení lze použít maximálně pro čtyři LED (každou musíme navíc připojit na zvláštní port – jedna bude na P0, druhá na P1 atd.). LED musí být buzena proti napájecímu napětí.

Pro připojení osmi LED je tedy nutné použít nějaký obvod běžně označovaný jako posilovač sběrnice. Ve shodě s [1] jsem použil zapojení s obvodem **74HCT245**. Přípravek **AT8LEDR** vychází ze zapojení přípravku **AT8LED**, který byl popsán v [1]. Důvody jsou dva: Jednak jsem chtěl, aby čtenáři [1] mohli použít stávající přípravek AT8LED (přes nepatrnou změnu zapojení je funkce stejná jako u AT8LEDR). Dále je jasné, že takový přípravek je právě pro svou jednoduchost univerzálně použitelný.

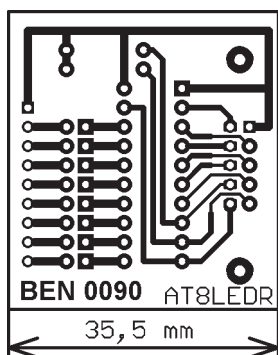
Protože je budič 74HCT245 neinvertující a LED jsou připojeny anodou na napájecí napětí, svítí LED v případě, že je na odpovídající vstup budiče připojena log. 0.

Výkres desky plošných spojů v měřítku 1:1 a osazovací plánek ukazují obr. 7.3 a obr. 7.4.

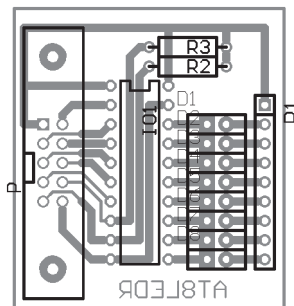
Vzhledem k tomu, že je na přípravku osm LED, je vhodný pro připojení na porty P0, P2 nebo P3. Přípravek lze s mírným omezením používat i na portu P1, pak ovšem LED připojené na P1.5 až P1.7 nepracují.

U mikrořadiče **AT89C2051** nemají vývody P1.0 a P1.1 zabudované pull-up rezistory, proto jsou použity rezistory  $R_2$  a  $R_3$ , které definují log. 1 (jinak by přípravek nezachoval zpětnou kompatibilitu k přípravku AT8LED z [1]; pokud tento ohled není důležitý, lze tyto rezistory vynechat).

V konstrukci jsem použil obdélníkové LED, které jsou sice dražší než běžné kruhové LED, ale plošný spoj je pak celkově méně rozměrný.



Obr. 7.3 Výkres desky plošných spojů přípravku AT8LEDR (BEN 0090)

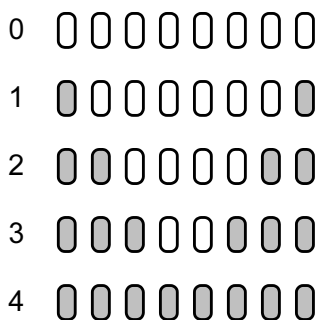


Obr. 7.4 Osazovací plánek přípravku AT8LEDR

### Rozpis součástek pro přípravek AT8LEDR (cena asi 60 Kč):

IO <sub>1</sub>	74HCT245 (74HC245, 74LS245)	1 ks
R <sub>1</sub>	RR 8X330R (odporová síť osmi odporů 330R)	1 ks
R <sub>2</sub> , R <sub>3</sub>	10k	2 ks
D <sub>1</sub> až D <sub>8</sub>	obdélníková červená LED	8 ks
P	PSL10	1 ks

**Příklad č. 1:** Pomocí přípravku **AT8LEDR** realizujte běžící světlo (připojte jej na port P3). Vždy souběžně svítí LED na krajích displeje. Nejdříve svítí D<sub>1</sub> a D<sub>8</sub>. Potom svítí D<sub>1</sub>, D<sub>2</sub> a D<sub>8</sub>, D<sub>7</sub>. V dalším kroku svítí D<sub>1</sub>, D<sub>2</sub>, D<sub>3</sub> a D<sub>8</sub>, D<sub>7</sub>, D<sub>6</sub>. Nakonec svítí všechny LED. Potom se sekvence opakuje. Doba svícení jedné kombinace je zhruba 0,5 s.



Obr. 7.5 Znáznornění požadované sekvence pro příklad č. 1

Při realizaci tohoto programu musíme uvážit několik skutečností:

- LED svítí při log. 0,
- posuv můžeme zajistit pomocí instrukcí rotace **RLC A** a **RRC A**,
- výsledky dílčích rotací sčítáme instrukcí **ANL** (tím, že pracujeme se zápornou logikou – LED svítí při log. 0, nebude se jednat o logický součet, ale součin),
- součet je nutný proto, že LED se posouvají z obou stran,
- čekání realizujeme použitím instrukce **DJNZ** nad několika registry,
- sekvence končí vynulováním (všechny LED svítí).

Nejdříve začneme zhasnutým stavem (OFFh).

V dalším kroku musí být zleva i zprava vsunuta nula. Tuto operaci zajistíme vynulováním příznaku C a následným použitím instrukcí **RRC A** a **RLC A**.

Každou rotaci musíme provést zvlášť. Proto nejdříve 2× uložíme obsah akumulátoru do zásobníku. Provedeme rotaci pomocí **RRC A** a výsledek uložíme do B. Potom obnovíme akumulátor a provedeme rotaci pomocí **RLC A**. Nakonec obnovíme původní stav akumulátoru a vše sečteme.

Takto lze realizovat posuv v každém kroku, konec se testuje jako nulovost akumulátoru (svítí všechny LED) pomocí instrukce **JZ**.

Při použití instrukcí **PUSH** a **POP** se musí použít označení **ACC** místo **A**. Důvod je prostý: Tyto instrukce používají přímé adresování (místo názvu registru se musí uvést jeho adresa).

Pro větší obecnost je port **P3** nahrazen symbolem **PORT**. Tak je možno elegantně změnit port, který používáme pro připojení přípravku **AT8LEDR**.

---

PROG\_1.ASM:

```
    $MOD8252
    ORG 0
    PORT EQU P3      ;výstupní port
    MOV R0,#0        ;nuluj počítadla pro
    MOV R1,#0        ;čekací smyčku (R0, R1)
SMYCKA: MOV A,#0FFh  ;do A 11111111B
                ;(vše zhasnuto)
SEKVENCE: MOV PORT,A ;pošli na PORT
    ACALL CEKEJ     ;vybav na 0,5 s
    JZ SMYCKA      ;znovu?
    CLR C          ;nuluj C
    PUSH ACC       ;2×schovej A
    PUSH ACC
    RRC A         ;rotuj doprava A přes C
    MOV B,A        ;mezivýsledek do B
    CLR C          ;nuluj C
    POP ACC
    RLC A         ;rotuj doleva A přes C
    ANL A,B      ;vlevo a vpravo do A
    POP B
    ANL A,B      ;to je nová hodnota
    SJMP SEKVENCE

;čekání cca 0,5 s
CEKEJ:  MOV R2,#7
CEKEJS: DJNZ R0,CEKEJS
          DJNZ R1,CEKEJS
          DJNZ R2,CEKEJS
          RET
          END
```

---

**POZNÁMKA:** Příklad č. 2 ukazuje realizaci běžícího světla bez použití instrukcí rotace, za pomoci konverzní tabulky popisující všechny kombinace.