Vážení zákazníci,

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

To znamená, že ukázka 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 ukázku jakýmkoliv způsobem dále šířit, veřejně či neveřejně např. umisťováním na datová média, na jiné internetové stránky (ani prostřednictvím odkazů) apod.

redakce nakladatelství BEN – technická literatura
redakce@ben.cz

V této kapitole je uvedena nejjednodušší aplikace mikrokontroléru AT89C2051 ovládaného sériovou linkou. Jedná se o použití vývodů portu P1 pro realizaci 8bitového vstupu/výstupu. Tato aplikace byla zvolena záměrně, abychom na ní mohli ukázat základní problematiku sériové komunikace ve spojení s tímto mikrokontrolérem.

Konstrukce je významně rozšířena v kapitole 8.

5.1 PŘÍPRAVEK AT8VV

Schéma zapojení přípravku **AT8VV** je uvedeno na *obr. 5.1*.

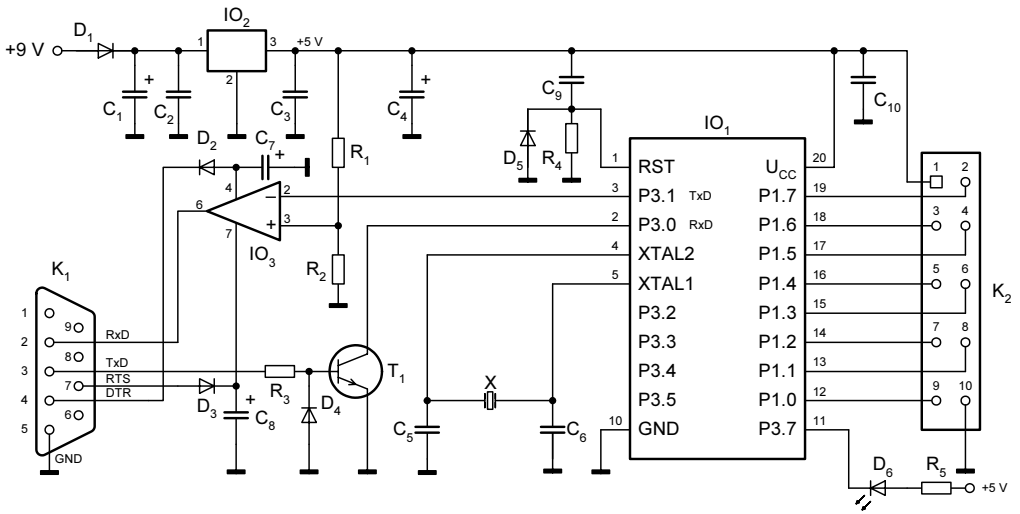
V napájecí části nalezneme obvyklou ochrannou diodu D_1 (zabraňuje přepólování), filtrační kondenzátor C_1 (vyhladí napětí) a stabilizátor IO_1 (7805). Pokud máte k dispozici stabilizovaný 5voltový zdroj, lze všechny tyto součástky vynechat a napájení přivést přímo na vývod 3 nezapojeného stabilizátoru.

Dále je zde zapojen krystal X spolu se zatěžovacími kondenzátory C_5 a C_6.

Vývody portu P1 jsou přímo přivedeny na konektor K_2 (PSL10), kam lze připojovat různé periferie. Z portu P3 je použit pouze vývod P3.7, který budí LED označenou D_6 (svítí při log. 0; připomeňme, že v log. 0 lze odebrat proud až 20 mA).

Zajímavější je řešení nulovacího obvodu. Při zapnutí zdroje se přes kondenzátor C_9 přivede krátký impulz na vývod RST. Před rezistor R_4 se kondenzátor postupně nabije a tak je nakonec napětí vstupu RST rovno nule. Takže se procesor rozběhne. Dioda D_5 zajišťuje rychlé vybití kondenzátoru C_9 po vypnutí zdroje. Tím je chráněn vývod RST.

Nejdůležitější je připojení mikrokontroléru k sériovému portu. V této konstrukci není použit obvyklý obvod MAX232, převodník úrovní TTL – RS232C je vytvořen nízkopříkonovým komparátorem IO_3 (TL061), který je napájen přímo z portu.



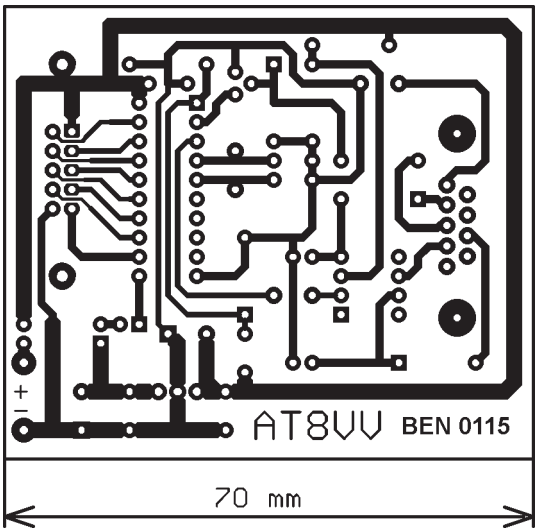
Obr. 5.1 Schéma zapojení přípravku AT8VV

K napájení slouží linky **RTS** (kladný pól) a **DTR** (záporný pól). Funkce byla vysvětlena v kapitole 4.6.

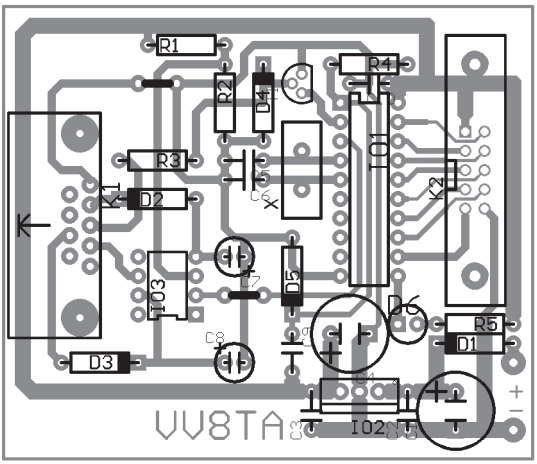
Převodník RS232C – TTL je tvořen tranzistorem T_1, který je doplněn bázovým odporem R_3 a diodou chránící přechod B-E při závěrné polarizaci (D_4). Funkce byla vysvětlena v kapitole 4.5.

Oba převodníky jsou invertující, což povaha linek sériového kanálu vyžaduje. Viz *obr. 4.2*.

Výkres desky plošných spojů přípravku **AT8VV (BEN 0115)** a osazovací plánek uvádí *obr. 5.2* a *obr. 5.3*.



Obr. 5.2 Výkres desky plošných spojů přípravku AT8VV (BEN 0115)



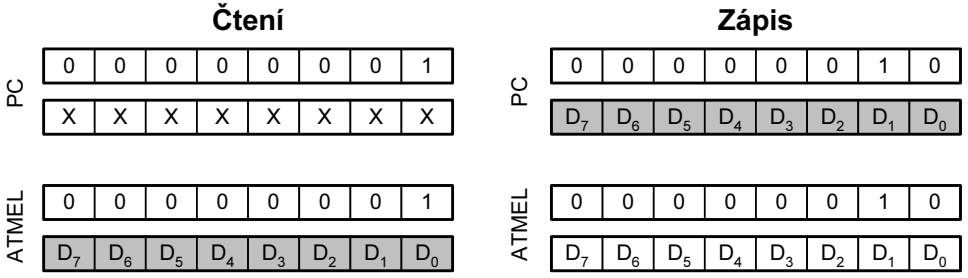
Obr. 5.3 Osazovací plánek přípravku AT8VV

Rozpis součástek pro přípravek AT8VV (cena asi 50 Kč, bez procesoru):

C_1, C_4	470 µF/16 V	2 ks
C_2, C_3, C_9, C_{10}	100n	4 ks
C_5, C_6	33p	2 ks
C_7, C_8	47 µF/16 V	2 ks
D_1	1N4001 až 1N4007	1 ks
D_2 až D_5	1N4148	4 ks
D_6	LED ∅ 5 mm	1 ks
IO_1	AT89C2051-24PI	1 ks
IO_2	7805	1 ks
IO_3	TL061	1 ks
K_1	CAN 9 Z 90	1 ks
K_2	PSL10	1 ks
R_1 až R_4	5k6	4 ks
R_5	330R	1 ks
T_1	BC548	1 ks
X	krystal 11,059MHz	1 ks

5.2 PROGRAM PRO AT89C2051

Než napíšeme programy pro mikrokontrolér a počítač, bude vhodné rozmyslet si, jakým způsobem bude komunikace mezi oběma zařízeními probíhat.



Obr. 5.4 Formát přenosu dat

Při přenosu dat je třeba rozlišit, zda jsou data vstupní či výstupní. Takže první bajt přenosu vyslaný z počítače bude obsahovat buď hodnotu 1 (čtení) nebo 2 (zápis). Následující bajt ponese data, která se mají zapsat na port. Aby se komunikace zjednodušila (formát bude mít délku 2 bajty v obou případech), posílá se druhý bajt i při čtení. Tento bajt je však při přenosu ignorován (může mít libovolnou hodnotu).

Mikrokontrolér vrací hodnotu prvního bajtu beze změny (tak se pozná, že bajt byl správně přijat) a připojí druhý bajt. Při čtení je to hodnota přečtená z portu, při zápisu

je to kopie posílaných dat. Vracení prvního bajtu zpět umožňuje rozpoznat úspěšné připojení přípravku.

Při realizaci programu pro mikrokontrolér si předně musíme zvolit přenosovou rychlost a formát. Obvyklá přenosová rychlost je **9600 Bd**, formát zvolíme **8bitový** s jedním stop-bitem a bez parity (opět obvyklé hodnoty). Přenos bude zabezpečen kontrolním čtením prvního bajtu přijatého zpět. Mikrokontrolér první bajt vždy zopakuje. Stačí tedy porovnat prvně poslaný bajt z počítače s prvně přijatým bajtem, který zopakoval mikrokontrolér.

V realizovaném programu pro mikrokontrolér nalezneme dva vektory **RESET** (reset procesoru), **SERIAL** (příjem nebo odvysílání znaku sériovým kanálem).

Při resetu je třeba nastavit přenosovou rychlost (**TH1**, **SMOD**, **TMOD**) a formát přenosu (**SCON**), aktivovat časovač 1 (**TR1**), který je použit jako synchronizační zdroj sériového přenosu a povolit přerušení od sériového kanálu. Nakonec se instrukcí **SJMP $** program zacyklí, bude tedy čekat na přerušení od sériového kanálu (příjem/odvysílání znaku).

Rutina **SERIAL** obsluhuje přerušení sériového kanálu. Nejdříve se testuje bit **TI**, abychom zjistili, zda se jedná o příjem nebo vysílání. Je-li **TI = 0**, jedná se o příjem (jinak jde o vysílání). Dále se musí rozlišit režim přenosu (čtení nebo zápis). Výchozí hodnota proměnné **REZIM** je **0**, což značí, že režim zatím není určen. Je-li **REZIM = 0**, určuje přijatý bajt režim přenosu. Hodnota se tedy uloží do **REZIM** a odvysílá zpět.

Pokud je **REZIM = CTENI**, přijatá hodnota se „zahodí“ a zpět se odešle načtený stav P1. Pokud je **REZIM = ZAPIS**, je přijatá hodnota zapsána na P1 a vrácena zpět.

Důležitý je závěr rutiny **SERIAL**. Zde se příznaky **RI** a **TI** nulují. Pokud by nebyly vynulovány, aktivovalo by se přerušení znovu! Příznaky **RI** a **TI** se nenulují automaticky po aktivaci obslužné rutiny, protože je nutno rozlišit, proč k aktivaci došlo.

Po každé aktivaci **SERIAL** se stav LED mění na opačný. Takže LED bliká v rytmu komunikace. Tím je indikována činnost zařízení.

AT8VV.ASM:

```
          $MODxx51

PORT      EQU P1          ;vstupně/výstupní port
LED       EQU P3.7        ;indikátor komunikace
BAUD      EQU 250         ;přenosová rychlost
HPCON     EQU 10000000B   ;SMOD nastaven
CTENI     EQU 1           ;režim čtení
ZAPIS     EQU 2           ;režim zápisu

          DSEG AT 30H
REZIM:    DS 1            ;proměnná indikující
                          ;režim
          CSEG
          AJMP RESET
          ORG 0023H
          AJMP SERIA
```

```
RESET:    MOV REZIM,#0         ;žádný režim
          MOV TH1,#BAUD        ;9600 Bd
          MOV TMOD,#00100000B ;č/č 1 8bitový
          MOV SCON,#01010000B ;8bitový přenos dat
          MOV PCON,#HPCON      ;SMOD=1
          SETB TR1             ;č/č 1 spuštěn
          SETB ES              ;povolení přerušení
          SETB EA              ;od sériového kanálu
          SJMP $               ;vyčkává na přerušení

;obsluha sériového kanálu:
SERIAL:   PUSH ACC             ;uložení registrů
          PUSH B
          PUSH PSW
          JB TI,SERIAK         ;test vysílání/příjem
;příjem bajtu:
          MOV A,REZIM          ;do A režim
          MOV B,SBUF           ;přijatý bajt do B
          JZ SERIAR            ;test režimu
          CJNE A,#CTENI,SERIAZ
;jedná se o CTENI:
SERIAC:   MOV SBUF,PORT        ;vyšli hodnotu kanálem
          MOV REZIM,#0         ;nuluj režim
          CPL LED              ;zneguj indikační LED
          SJMP SERIAK
;jedná se o ZAPIS:
SERIAZ:   MOV PORT,B           ;zapiš přijatou hodnotu
          MOV SBUF,B           ;pošli ji zpět
          MOV REZIM,#0         ;znuluj režim
          CPL LED              ;zneguj indikační LED
          SJMP SERIAK
;jedná se o bajt indikující režim:
SERIAR:   MOV REZIM,B          ;ulož režim
          MOV SBUF,REZIM       ;pošli zpět
SERIAK: CLR TI                 ;znuluj příznaky
          CLR RI
          POP PSW              ;obnov registry
          POP B
          POP ACC
          RETI

          END
```