

# 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.

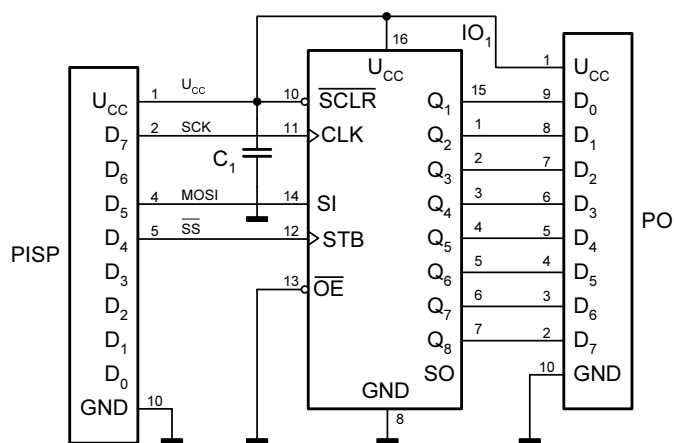
*redakce nakladatelství BEN – technická literatura*  
[redakce@ben.cz](mailto:redakce@ben.cz)



### 11.3.5 Přípravek ATSPITST

Pro test funkce SPI rozhraní byl navržen přípravek **ATSPITST**.

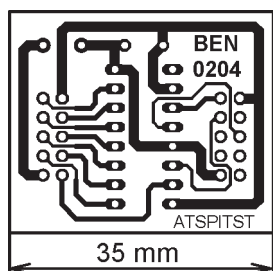
Jedná se o jednoduchý SPI přijímač vystavěný na základě posuvného registru typu SIPO s označením **74HCT595** (IO<sub>1</sub>). Posuvný registr typu 74HCT595 byl popsán například v [1, 5 nebo 6] a lze jej bez vážnějších potíží připojit na SPI sběrnici. Schéma zapojení je uvedeno na obr. 11.21.



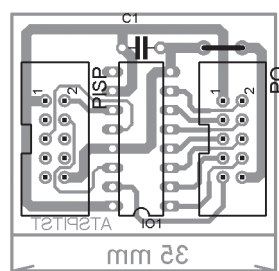
Obr. 11.21 Schéma zapojení přípravku ATSPITST

Vývod **SI** je vstup sériových dat, připojí se tedy na signál **MOSI** od mastera. Podobně **CLK** jsou hodiny přenosu, připojí se tedy na signál **SCK** mastera. Poslední je hodinový vstup **STB**, který byl připojen na vývod **SS** (mohl by být ale připojen i jinak). Ten jediný musíme ovládat programově (jeho obsluhu nezajistí SPI rozhraní). Náběžná hrana tohoto signálu provede přepis sériově přijatých dat na paralelní výstup tvořený vývody **Q<sub>1</sub>** až **Q<sub>8</sub>**.

Signály SPI sběrnice jsou přivedeny na konektor **PISP**, 8bitová výstupní sběrnice včetně napájení je k dispozici na konektoru **PO**. Na konektor **PO** tedy můžeme připojit libovolný z výstupních přípravků, například **AT8LED** nebo **ATDISP**.



Obr. 11.22 Plošný spoj přípravku ATSPITST (BEN 0204)



Obr. 11.23 Osazovací plánek

### Rozpis součástek pro přípravku ATSPITST (cena asi 50 Kč):

C <sub>1</sub>	CK 100N/63V	1 ks
IO <sub>1</sub>	74HCT595	1 ks
PISP, PO	MLW10G	2 ks

## 11.3.6 Příklad PROG\_12

Tento příklad ukazuje použití SPI rozhraní pro výstup dat.

**Příklad č.12:** Připojte konektor **PISP** přípravku **ATSPITST** na port **P1**. Na konektor **PO** připojte přípravek **AT8LED**.

Vytvořte program, který pomocí SPI rozhraní odesílá sekvenci dat odpovídající běžícímu světlu (**PROG\_03**, varianta **RLC**).

V tomto případě budeme používat pouze port **P1**. Pro jednoduchost jej nakonfigurujeme tak, že všechny jeho bity budou kvaziobousměrné. Dále musíme povolit alternativní funkci vývodů **P1.5 až P1.7** (vývody SPI rozhraní).

Nyní se dostáváme ke konfiguraci SPI rozhraní. Musíme tedy správně nastavit obsah registru **SPCR**:

- nebudeme používat přerušování (SPIE = 0),
- bit **SPE** musíme nastavit až po volbě režimu (SPE = 0),
- bity musíme odeslat v pořadí LSB až MSB, jinak budou přetočeny na výstupu (DORD = 0),
- hodinový signál generuje mikrokontrolér (MSTR = 1),
- v souladu s *obr. 11.4* volíme CPOL = 0 a CPHA = 0 (datové bity se potvrzují náběžnou hranou hodin, mají malý předstih),
- použijeme nejrychlejší hodinový signál, SPR1 = 0, SPR0 = 0 (pro 16 MHz krystal vychází v souladu s *tab. 11.1* hodinový kmitočet 8 MHz).

Takže do registru **SPCR** zapíšeme hodnotu **00010000B**. Následně nastavíme bit **SPE**, takže zapíšeme podobnou hodnotu **01010000B**. Tím je SPI rozhraní konfigurováno a aktivováno.

Vlastní odesílání dat bude probíhat jejich zápisem do registru **SPDR**. Narozdíl od příkladu **PROG\_03** nebudeme používat rotaci, ale data budeme „tahat“ z tabulky označené návěštím **TAB**. Tato tabulka je vytvořena v programové paměti, pro přístup k těmto konstantám se používá instrukce **MOVC A, @A+DPTR**. Čili adresa do programové paměti je dána součtem obsahu akumulátoru a registru **DPTR**, přečtený údaj se uloží do akumulátoru.

Před spuštěním smyčky **CYKL** naplníme **DPTR** adresou tabulky konstant a registr **B** zase jejich počtem. Vývod **SS** ovládá signál **STB**, zápis probíhá náběžnou hranou, na zápis se tedy připravíme instrukcí **CLR SS** (SS = 0). Při čtení konstant bude akumulátor nulový, takže budeme zvyšovat adresu obsaženou v registru **DPTR** (uchovávání jeho obsahu by bylo velmi pracné).

Po vykonání instrukce **MOV SPDR, A** začne vysílání. Bude však nějakou dobu trvat. Nové vysílání nemůžeme začít dříve, než se dokončí započatý přenos. Proto se testuje příznak **SPIF** obsažený v registru SPSR. Nelze k němu bohužel přistoupit bitovou adresu, musíme jej nahrát do akumulátoru a pak jej testovat (jedná se o sedmý bit).

Pokud bude příznak SPIF nastaven, přečtením registru SPSR dojde k jeho hardwarovému vynulování. Takže můžeme pokračovat dále.

Nakonec tedy přenesený údaj zapíšeme na výstupy obvodu **74HCT595** vykonáním instrukce **SETB SS** (STB = 1, generovala se náběžná hrana), zařadíme krátkou čekací smyčku realizovanou podprogramem **CEKANI**, zvýšíme adresu obsaženou v registru DPTR. Pokud není vynulován registr B, pokračuje dalším bajtem.

Po odeslání celé tabulky (B = 0) se provede skok na návěští **SMYCKA** a celá sekvence se opakuje.

---

#### PROG\_12.ASM:

```

    $INCLUDE (LP2052.INC)

SS      EQU P1.4           ;/SS signál
POCET   EQU 9             ;počet bajtů

RESET:  ;inicializace:
          MOV P1M0,#0         ;P1 konfigurován
          MOV P1M1,#0         ;kvaziobousměrně
          MOV P1,#11100000B   ;alt.f. P1.5 až P1.7
          MOV SPCR,#0010000B  ;konfig. SPI
          MOV SPCR,#01010000B ;nastav SPE

SMYCKA: MOV DPTR,#TAB      ;DPTR na TAB
          MOV B,#POCET        ;B na počet

CYKL:   CLR SS             ;STB=0
          CLR A                ;A=0
          MOVC A,@A+DPTR      ;čti z TAB
          MOV SPDR,A          ;pošli přes SPI

SPITST: MOV A,SPSR        ;test konce
          JNB ACC.7,SPITST    ;SPIF
          SETB SS              ;STB=1
          ACALL CEKANI        ;čekání 0,1 s
          INC DPTR            ;zvyš pozici
          DJNZ B,CYKL         ;zpět do cyklu
          SJMP SMYCKA        ;začni znovu
          ;čekací rutina:

CEKANI: MOV R0,#0          ;nastavení
          MOV R1,#208         ;čekací doby
          MOV R2,#10          ;zhruba 0,1 s

```

```

CEKEJ:  DJNZ R0,CEKEJ          ;1. smyčka
        DJNZ R1,CEKEJ          ;2. smyčka
        DJNZ R2,CEKEJ          ;3. smyčka
        RET

        ;data pro vysílání:
TAB:  DB 11111111B           ;zhasnuto
        DB 11111110B           ;D0
        DB 11111101B           ;D1
        DB 11111011B           ;D2
        DB 11110111B           ;D3
        DB 11101111B           ;D4
        DB 11011111B           ;D5
        DB 10111111B           ;D6
        DB 01111111B           ;D7

        END

```

Příklad najdete na doprovodném CD-ROM v adresáři **PROGRAMYPROG\_12**.

### 11.3.7 Přípravek ATOSC

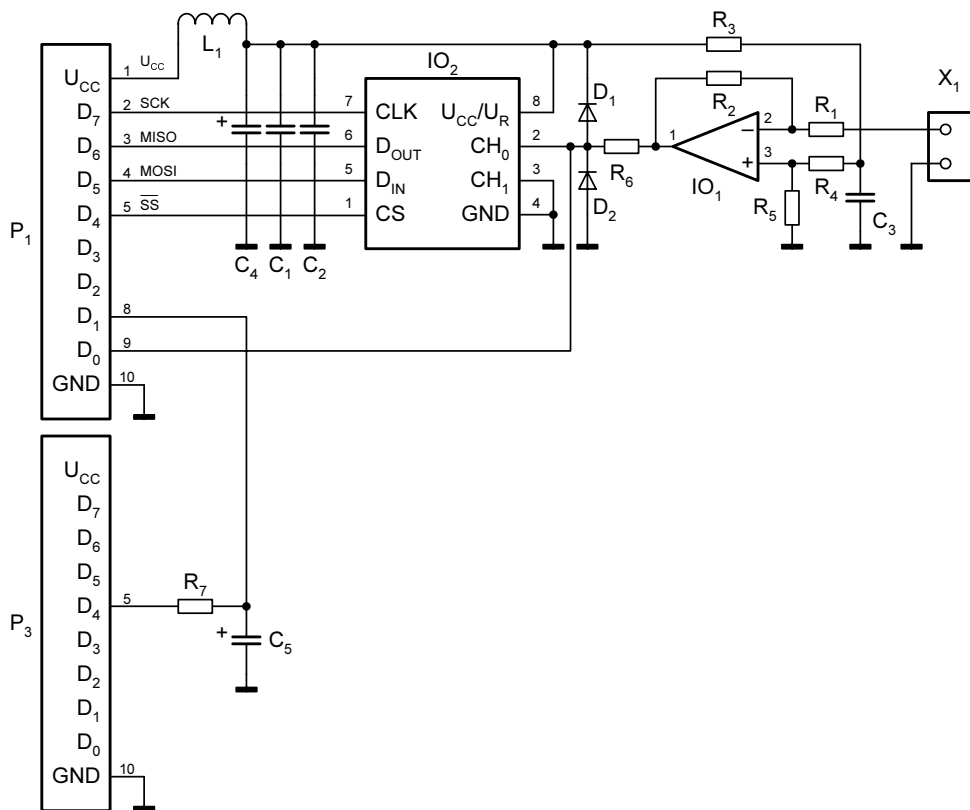
Přípravek **ATOSC** nám umožní vybudovat z mikroprocesoru jednoduchý digitální osciloskop (viz kapitolu 13.4). Schéma zapojení je uvedeno na *obr. 11.24*.

Konektor **P<sub>1</sub>** je určen pro připojení na port P1, konektor **P<sub>3</sub>** pak zase na port P3.

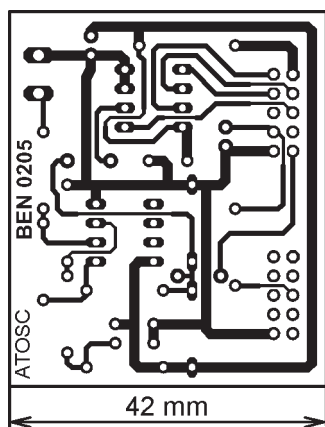
Jádrum zapojení je A/D převodník **MCP3002** (IO<sub>2</sub>). Napájecí a referenční napětí je získáno přímo z vývojového kitu (není použit vnější zdroj), pouze je vyhlazeno článkem L<sub>1</sub>, C<sub>4</sub>. Vývody **CLK**, **DOUT**, **DIN** a **CS** jsou připojeny na odpovídající signály SPI kanálu: SCK, MISO, MOSI a  $\overline{SS}$ . Z analogové části se používá pouze kanál **CH<sub>0</sub>**, je schopen zpracovat napětí v rozsahu 0 až U<sub>CC</sub>. Ochrana proti přepětí na vstupu je zajištěna omezovačem R<sub>6</sub>, D<sub>1</sub>, D<sub>2</sub>.

Operační síť vystavěná kolem zesilovače IO<sub>1</sub> (MCP602) slouží k posuvu vstupního napětí. Napětí přivedené na svorkovnici X<sub>1</sub> může být bipolární v rozsahu zhruba  $-1/2 U_{CC}$  až  $+1/2 U_{CC}$  (tedy asi  $\pm 2,5$  V). Vstupní napětí přivedené se získá  $-1$  (tedy invertovaně) na výstup se totiž sčítá s polovinou U<sub>CC</sub>. Aby nebyl nutný záporný zdroj napětí, byl zvolen operační zesilovač **MCP602**, který vystačí s nesymetrickým napájením.

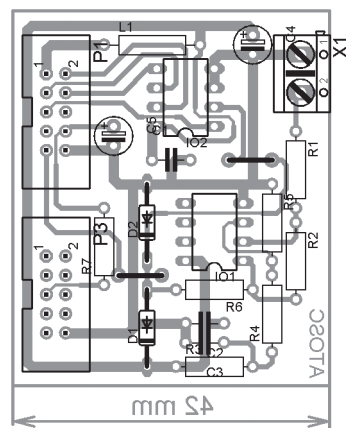
Výstup operačního zesilovače je přiveden ještě na neinvertující vstup analogového komparátoru, který je zabudován v mikrokontroléru (viz kapitolu 12). Invertující vstup je pak připojen na integrační článek D/A převodníku tvořený R<sub>7</sub>, C<sub>5</sub>. Tento převodník je vytvořen z PWM signálu produkovaného na vývodu T0 (P3.4). Dané zapojení bude sloužit pro spouštění osciloskopu. Když vstupní napětí dosáhne spouštěcí úrovně (nastavené D/A převodníkem), odstartuje se snímání vstupního signálu.



Obr. 11.24 Schéma zapojení přípravku ATOSC



Obr. 11.25 Plošný spoj přípravku ATOSC (BEN0205)



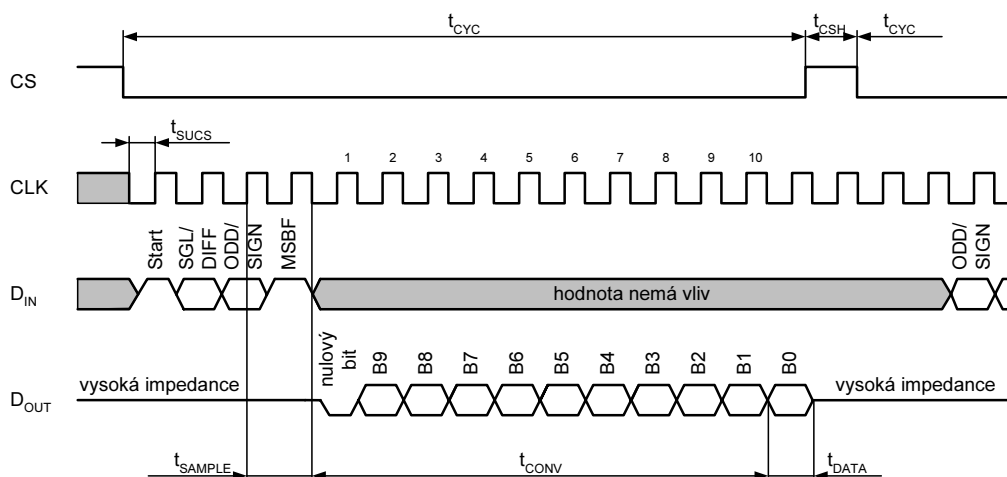
Obr. 11.26 Osazovací plánek

**Rozpis součástek pro přípravek ATOSC (cena asi 150 Kč):**

C <sub>1</sub> až C <sub>3</sub>	CK 100N/63V	3 ks
C <sub>4</sub>	E100M/16V	1 ks
C <sub>5</sub>	E1M/50V	1 ks
D <sub>1</sub> , D <sub>2</sub>	1N4148	2 ks
IO <sub>1</sub>	MCP602-I/P	1 ks
IO <sub>2</sub>	MCP3002-I/P	1 ks
L <sub>1</sub>	TL. 100μH	1 ks
P <sub>1</sub> , P <sub>3</sub>	MLW10G	2 ks
R <sub>1</sub> , R <sub>2</sub>	RR 150K	2 ks
R <sub>3</sub> , R <sub>5</sub>	RR 100K	2 ks
R <sub>4</sub>	RR 200K	1 ks
R <sub>6</sub>	RR 1K	1 ks
R <sub>7</sub>	RR 10K	1 ks
X1	ARK500/2	1 ks

Nyní si povíme něco více o obvodu **MCP3002**. Jedná se o dvoukanálový A/D převodník s rozlišením až 10 bitů. Pracuje na principu postupné aproximace a převedený údaj vysílá sériově, sběrnicí SPI. Jedná se o poměrně rychlý převodník, takže z něj můžeme data načítat velmi rychle. Podrobnější popis lze nalézt například v [2].

Na obr. 11.27 je ukázáno, že pro spuštění převodu se musí po lince D<sub>IN</sub> vyslat startovací sekvence. První bit (start) má vždy hodnotu log. 1, ostatní tři bity určují: režim funkce (diferenční × proti zemi), kanál, způsob vysílání. Nejdříve je z obvodu odeslán potvrzovací nulový bit a následuje 10 bitů v pořadí od nejvíce významného po nejméně významný. Počet přijímaných bitů je možno zkrátit, přijmout například jen 8 bitů (nejnižší dva bity prostě vynecháme), předčasným vytažením linky CS na log. 1.



Obr. 11.27 Časový diagram komunikace