

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



29 PŘÍKLAD Č. 11 – VÝPIS ČÍSLA POMOCÍ PRINTF NA MLCDDGEN

Nyní si ukážeme, jak upravit příklad č. 10 tak, abychom mohli používat funkci **printf** pro výpis textu na modulu **MLCDDGEN**.

PROG_11:

```
#include <regx52.h>
#include <stdio.h> ← vložení hlavičkového souboru s definicí printf

void clear()
{
    TB8=1;
    SBUF=1;
    while(!TI);
    TI=0;
}

char putchar(char c) ← přesměrování výstupu
{
    TB8=0;
    SBUF=c;
    while(!TI);
    TI=0;

    return c;
}

void main()
{
    int a=5,b=8,c=a+b;

    TH1=243; ← nastavení časovače 1 a UART
    TMOD=0x20;
    TR1=1;
    SCON=192;
    PCON=128;

    clear(); ← smazání LCD
    printf("a=%d, b=%d, c=%d",a,b,c);

    while(1); ← výpis přes printf
}
```

Do zdrojového textu musíme vložit hlavičkový soubor **stdio.h**.

Výpis tedy nebudeme řešit pomocí dříve zavedené funkce **write**, ale funkcí **putchar**. Takže úprava je velmi jednoduchá.

Otestování funkce **printf** je ukázáno na výpisu čísel a, b, c. Hodnoty jsou a = 5, b = 8, c = a+b. Na displeji se zobrazí text: a=5, b=8, c=13.

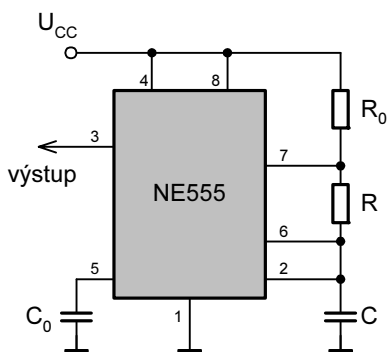
Přípravek **MLCDGEN** připojíme opět na port **P3**.

42 PŘÍPRAVEK MRX555 – MĚŘENÍ ODPORU MEZIPŘEVODEM

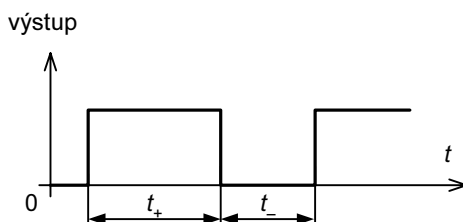
Pro převod odporu nebo kapacity na kmitočet je třeba použít astabilní klopný obvod (multivibrátor). Klasické zapojení multivibrátoru s obvodem NE555 uvádí obr. 42.1.

V tomto obvodu je kondenzátor C nejdříve nabíjen přes rezistory R_0 a R na $2/3$ napájecího napětí (na výstupu je log. 1) a potom vybitý přes rezistor R na $1/3$ napájecího napětí (na výstupu je log. 0). Nabíjení odpovídá interval t_+ , vybíjení pak interval t_- dle obr. 42.2.

Měřený odpor je představován rezistorem R , měřená kapacita pak kondenzátorem C . Součástky R_0 a C_0 jsou nutné pro správnou funkci zapojení. C_0 je blokovací kondenzátor řídicího napětí obvodu 555, R_0 zajišťuje nabíjení kondenzátoru C .



Obr. 42.1 Obvod NE555 zapojený jako multivibrátor



Obr. 42.2 Časové průběhy výstupu obvodu dle obr. 42.1

Pro zapojení z obr. 42.1 platí tyto vztahy:
doby trvání obou fází:

$$t_+ = 0,693 \cdot (R_0 + R) \cdot C, \quad t_- = 0,693 \cdot R \cdot C$$

kmitočet:

$$f = \frac{1}{T} = \frac{1}{t_+ + t_-} = \frac{1,443}{(R_0 + 2R) \cdot C}$$

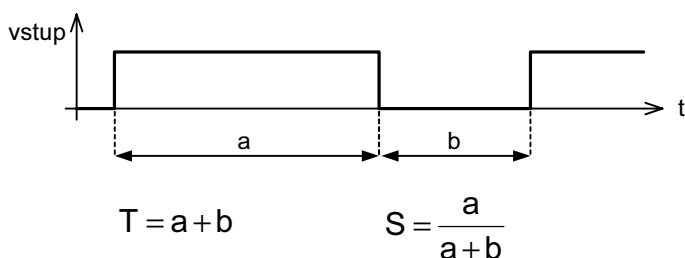
Nejjednodušším způsobem měření odporu R nebo kapacity C je stanovit dobu trvání log. 0 (tedy interval t_-). Měřená hodnota je totiž přímo úměrná této době.

Stanovení kmitočtu je snazší (provede se pomocí čítače), ovšem výpočet měřených hodnot R nebo C je trochu obtížnější.

Měření časových intervalů, periody a střídy

Při programovém měření doby trvání log. 1 a log. 0 je nezbytně nutné stanovit správně okamžik počátku a konce (detekovat signál hranově).

Přesnost měření lze zvýšit tak, že dobu trvání log. 1 a log. 0 vyhodnocujeme z několika period jako aritmetický průměr.



Obr. 42.3 Měření periody a střídy

Měření periody a střídy probíhá podobně. V obou případech musíme určit dobu trvání log. 1 a log. 0 (v obr. 42.3 označeno jako a a b).

Perioda je pak definována jako součet doby trvání log. 1 a log. 0.

Střída je podíl doby trvání log. 1 a periody.

Měření kmitočtu

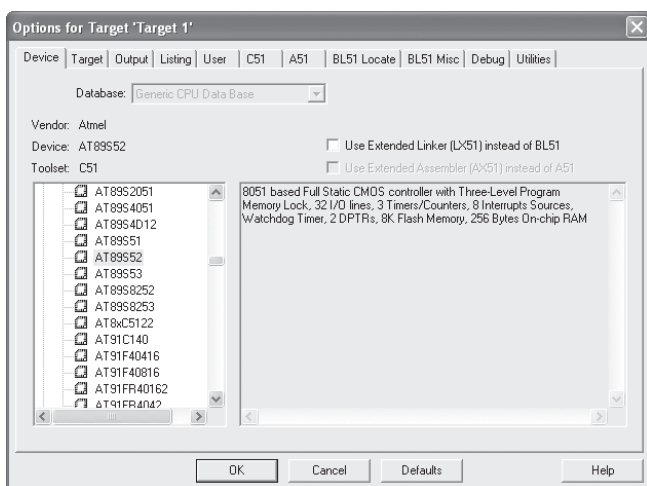
Nejjednodušší způsob měření kmitočtu je založen na použití čítače. Kmitočet je totiž definován jako počet opakování periodického děje za jednotku času.

Pokud budeme čítat impulzy během jedné sekundy, bude jejich počet odpovídat kmitočtu v Hz. Například pro vstupní kmitočet 1 kHz napočítáme za 1 s v ideálním případě 1000 impulzů (takový bude obsah čítače).

C POPIS DŮLEŽITÝCH PRVKŮ VÝVOJOVÉHO PROSTŘEDÍ KEIL μ VISION4

Nyní se seznámíme s příkazy vývojového prostředí Keil μ Vision4, které se nejčastěji používají. Jedná se zejména o tyto položky hlavní nabídky:

- **Project|Open Project** – otevře dříve vytvořený projekt,
- **Project|Close Project** – zavře aktuální projekt,
- **Project|Select Device for Target** – zobrazí dialog pro volbu procesoru, pro který je určen cílový soubor,
- **Project|Options for Target** – zobrazí dialog, který především umožňuje volbu procesoru (viz obr. C.1), paměťového modelu (viz obr. C.2, lze zvolit i kmitočet procesoru, který se následně používá při ladění) a výstupního souboru (viz obr. C.3, volba **Create HEX File** zajistí vytvoření výsledného HEX souboru, zrušením volby **Debug Information** zajistíme odebrání ladicích informací z výsledného programu).
- **Project|Build Target** – přeloží projekt.



Obr. C.1 Výběr procesoru

Ladění

Ladění (Debugging) umožňuje sledovat běh programu simulováním činnosti mikrokontroléru. Ladicí režim vyvoláme pomocí nabídky **Debug|Start/Stop Debug Session**. Po této volbě se aktivují další položky nabídky **Debug**:

- **Reset CPU** – resetuje mikrokontrolér,