Napsal uživatel Vašek Král
Středa, 06 Duben 2011 07:33

Kompatibilní zapojení: LED panel s ATmega8
Ke stažení: LED_002.c == LED_002.pdf == LED_002.htm
.
.

Untitled <!-- body { color: #000000; background-color: #FFFFFF; } .cpp1-assembler { } .cpp1-brackets { } .cpp1-comment { color: #008000; font-style: italic; } .cpp1-float { color: #000080; } .cpp1-hexadecimal { color: #000080; } .cpp1-character { } .cpp1-identifier { } .cpp1-illegalchar { } .cpp1-number { color: #000080; } .cpp1-octal { color: #0000FF; } .cpp1-preprocessor { } .cpp1-reservedword { font-weight: bold; } .cpp1-space { color: #008080; } .cpp1-string { color: #800000; } .cpp1-symbol { } --> /*
V minulém "programu" se nám podařilo rozsvítit vybrané Led diody. Protože se však nejednalo o program v pravém slova smyslu, ale o pouhé jednorázové nastavení pinů, pokusíme se nyní toto nastavení v průběhu času pomocí programu měnit. Vytvoříme program, který bude ledky střídavě rozsvěcet a zhasínat.
*/
/*
Protože už Led diody nebudou svítit stále, ale pouze určitý čas, budeme potřebovat čekací funkce z knihovny "delay.h" která se nachází v adresáři "util". Tuto knihovnu načteme do programu příkazem:
#include <util/delay.h>
*/
/*
Blikání je periodická činnost, a bylo by neúčelné vypisovat do nekonečna program typu:
rozsviť;
čekej;
zhasni;
čekej;
rozsviť;
čekej;
...
Takovýto program by byl velice objemný, a dokázal by bliknout pouze tolikrát, kolikrát bychom mu to napsali. My však chceme, aby ledky blikaly stále.
K tomuto účelu se hodí takzvaná nekonečná smyčka. Nekonečná smyčka je programová konstrukce, která zajistí nekonečné opakování instrukcí ve svém

těle.
Možnosti jak zapsat nekonečnou smyčku jsou dvě:
Buďto jako cyklus for:

```
for(;;)
{
opakované příkazy;
}
```

Nebo jako cyklus while:

```
while (1)
{
opakované příkazy;
}
```

Použití je víceméně libovolné, protože AVR GCC oba cykly překládá stejně.
*/
//Náš program by tedy mohl vypadat takto:

```
/*
Funkce čekání z knihovny delay.h potřebuje vědět, na jaké frekvenci procesor
poběží, protože musí vypočítat, kolik cyklů má procesor počkat, než bude
pokračovat dál. Tuto informaci očekává v konstantě "F_CPU", a my jí tedy
musíme nadefinovat:
*/
#define F_CPU 1000000UL  // 1 MHz (základní frekvence)

#include <avr/io.h>  //Nahrajeme zase knihovnu vstupů a výstupů (PORT, DDR)
#include <util/delay.h> //Nahrajeme knihovnu čekacích funkcí
/*
Je vhodné si předem nadefinovat konstanty, které se později objeví v programu.
Když pak budeme chtít program upravovat, nemusíme jej celý prohledávat, ale
stačí pouze přepsat tyto konstanty na začátku.
*/
#define CEKANI 500  //konstanta "CEKANI" má hodnotu 500 (500 milisekund)
#define STAV1 0b01000101 //ledky které budou svítit nejdřív (1 = svítí)
#define STAV2 0b10111010 //ledky, které budou svítit potom.

int main (void) //jako vždy - hlavní funkce
{
DDRB = 0b11111111; //Piny 0 - 7 budou výstupní
for (;;) // Nekonečná smyčka
 {
 PORTB = STAV1;  //Pošleme konstantu STAV1 na nožičky - rozsvítí se LED
 _delay_ms (CEKANI); //Čekání tolik milisekund, kolik je v konstantě CEKANI
 PORTB = STAV2;  //Rozsvítíme jiné ledky...
 _delay_ms (CEKANI); //...a opět čekáme (funkce z knihovny delay.h)
 }  // ...a zase na začátek! (cyklu for)
} //konec programu - sem se program nikdy nedostane. Bude běhat pořád v
```

```
 //cyklu for.
/*
Po nahrání programu do přípravku "ATmega8 - LED panel" budou střídavě blikat
LED diody tak, jak je to definováno v konstantě STAV1 a STAV2. Čas mezi
jednotlivými změnami stavu je definován v konstantě CEKANI.
Tyto konstanty lze libovolně měnit..
*/

//Pro radioklub OK1KVK naspal Vašek Král
```