

DIGITÁLNÍ HODINY

Michal Struhelka
4. ročník SŠPH Uh. Hradiště

ABSTRAKT

Ve své práci jsem navrhoval digitální hodiny s využití procesoru Atmega16. Čas a datum se bude zobrazovat na dvouřádkovém LCD displeji. Nastavení času a datumu se bude provádět pomocí dvou tlačítek na desce STK500.

Programování jsem prováděl v programovacím jazycku C pomocí vývojového prostředí CodevisionAVR. Zavedení a odzkoušení programu jsem použil AVR Studio 4 a programátoru AVR JTAGICE .

KLÍČOVÁ SLOVA

hodiny, Atmel, Atmega16, CodeVisionAVR C Compiler, AVR Studio 4

1 ÚVOD

Práce se zabývá návrhem programu digitálních hodin pomocí mikrokontroléru Atmega16 a zobrazováním času a datumu na dvouřádkovém LCD displeji.

K programování je použito vývojové prostředí CodeVisionAVR C. V tomto vývojovém prostředí se programuje pomocí programovacího jazyka C. Tento program umožňuje překlad z jazyku C do strojového kódu (soubor s příponou .hex). Součástí tohoto programu jsou kromě standardních knihoven jazyka C další knihovny určené pro použití např. LCD.

K nahrání programu do procesoru bylo využito prostředí AVR Studio 4 a programátoru AVR JTAGICE mkII.

K odzkoušení programu byla použita vývojová deska s označením STK500 od výrobce Atmel.

2 ROZBOR PROBLÉMU

Nejprve potřebujeme definovat a nastavit jednu sekundu. Provedeme to tím způsobem, že nastavíme 16-bit čítač/časovač tak, aby nastalo přerušení přesně za jednu sekundu. Budeme používat krystal 16MHz. Nastavením registru TCCR1B=0x14 docílíme toho, že tento čítač bude čítat hodinové pulzy z předděličky 256, tj. na jeho vstupu bude kmitočet $16\,000\,000/256 = 62\,500$ Hz. Pomocí následujícího vzorce vypočítáme vrchol čítače/časovače:

$$TOP = \frac{f_{KR}}{2 \cdot N \cdot f} = \frac{16000000}{2 \cdot 256 \cdot 1} = 31250$$

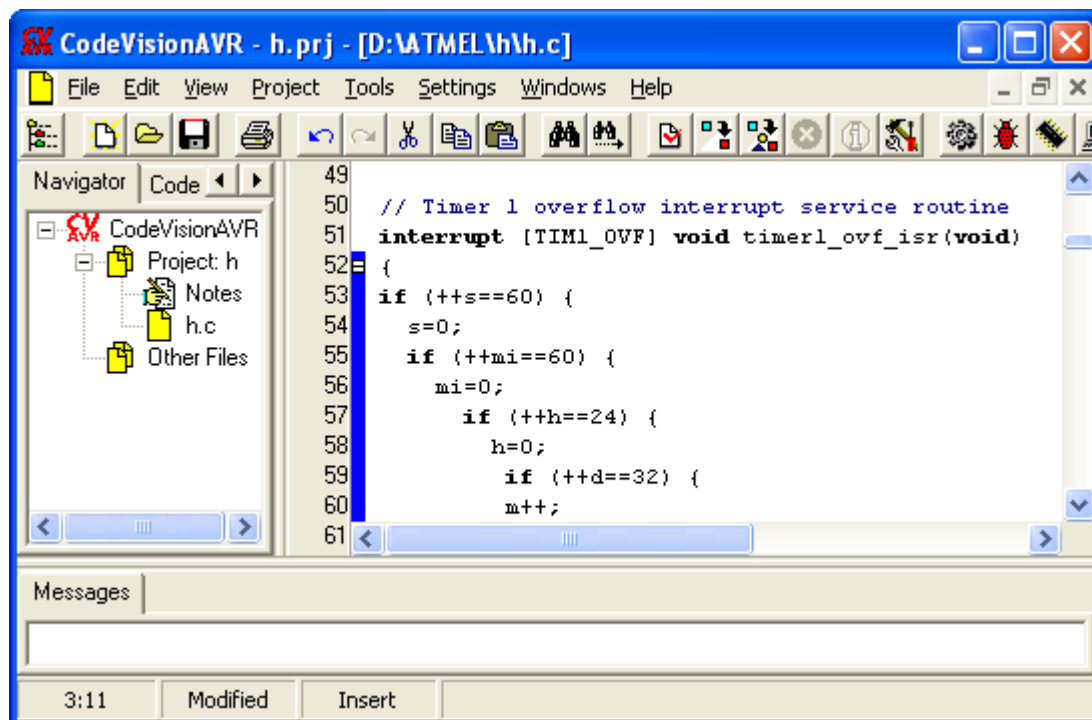
Tuto hodnotu nastavíme do registrů TCNT1H=0x7A a TCNT1L=0x12. K volání obsluhy přerušení vyvolané přetečením čítače/časovače1 pak bude docházet vždy za 1 sekundu. Součástí této obsluhy přerušení bude zvýšení o jedničku obsahu proměnné obsahující údaj o počtu sekund. Bude-li však výsledek roven 60, musí se o 1 zvýšit počet minut a údaj o počtu sekund nastavit na nulu. Obdobně v případě dosažení 60 minut či 24 hodin i počet dnů v jednotlivých měsících včetně uvažování přestupných roků.

Čas a datum budou zobrazeny na LCD, na prvním řádku bude čas ve formátu hh:mm:ss a na druhém řádku bude datum ve formátu dd/mm/rrrr.

Nastavení datumu a času bude probíhat dvěma tlačítky sw0 a sw1 (značení na desce STK500). Po přidržení tlačítka sw0 po dobu asi 3s se zakáže přerušení (to způsobí zastavení hodin) a začne blikat hodnota která po stisku sw1 se přičte o 1, po dalším stisku sw0 se nastavení přesune na další hodnotu atd. Po posledním stisku sw0 se zapne přerušení.

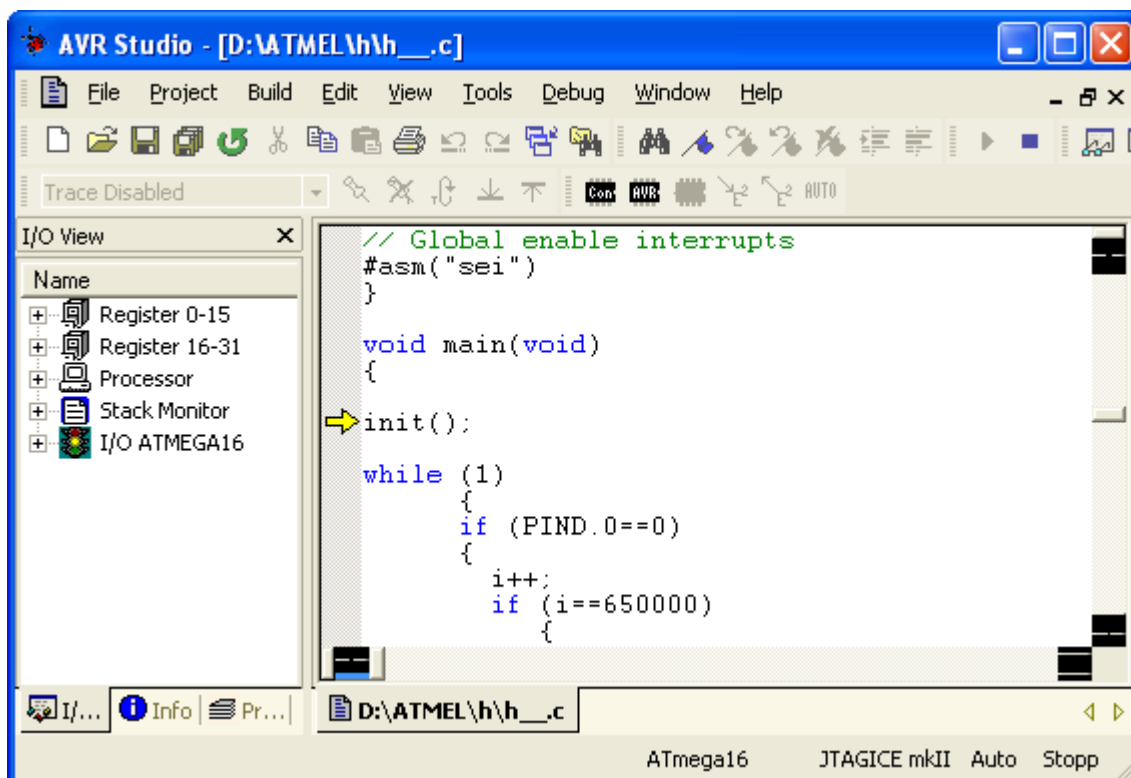
3 VÝVOJOVÉ PROSTŘEDÍ

Pro programování jsem použil vývojové prostředí CodeVisionAVR C Compiler (Obr.01), kde se programuje v programovacím jazyku C. Pro program jsem použil knihovny mega16.h - zde jsou uloženy adresy registrů procesoru, Delay.h - zde jsou zpožďovací smyčky a Stdlib.h – zde je převod z celého čísla na řetězec znaků.



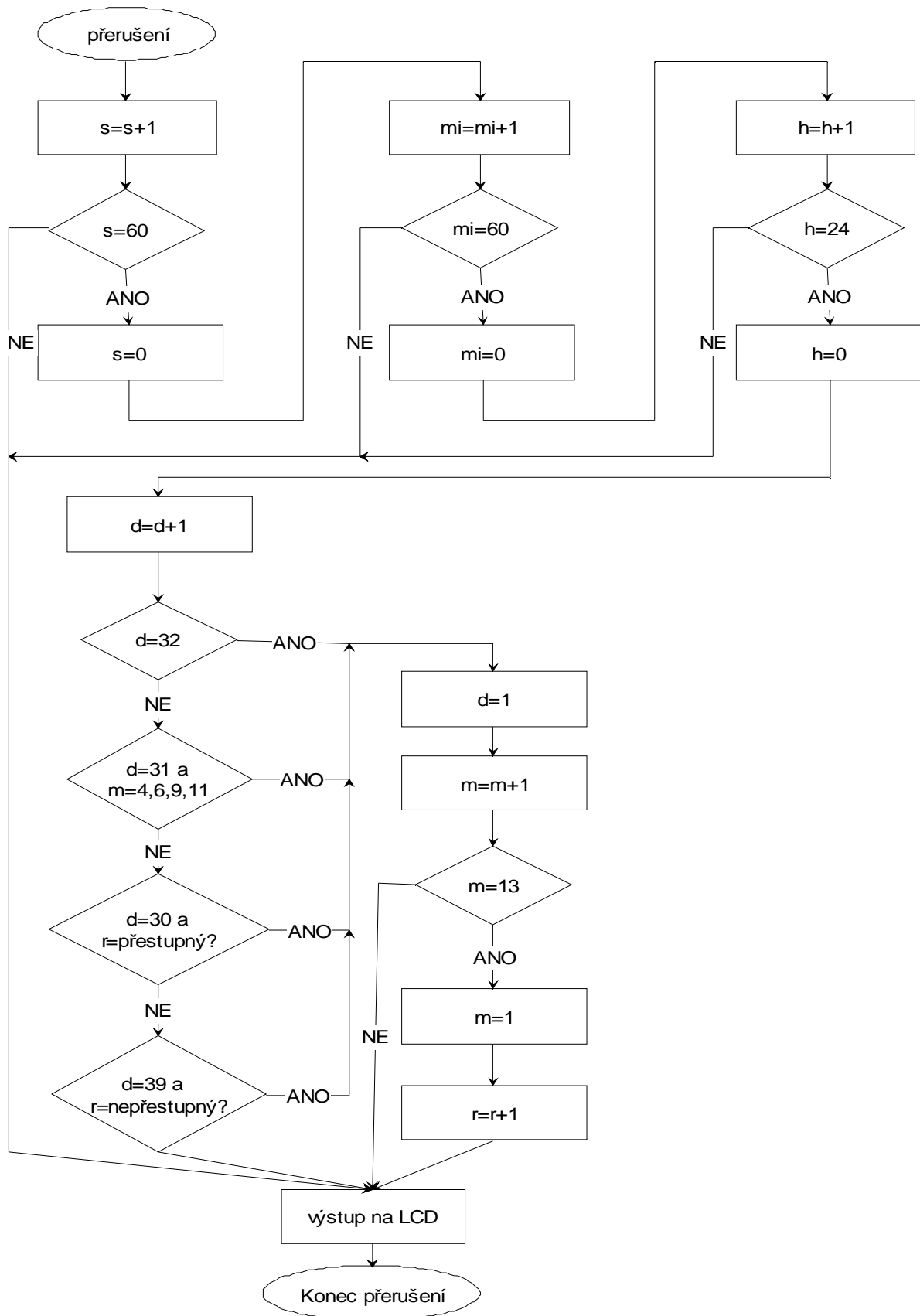
Obr. 1 CodeVisionAVR C Compiler

Pro nahrání programu do procesoru jsem použil program AVR Studio 4, kde lze program odkrokovat po jednotlivých instrukcích (lze se podívat na data v jednotlivých registrech) a případně opravit chyby.

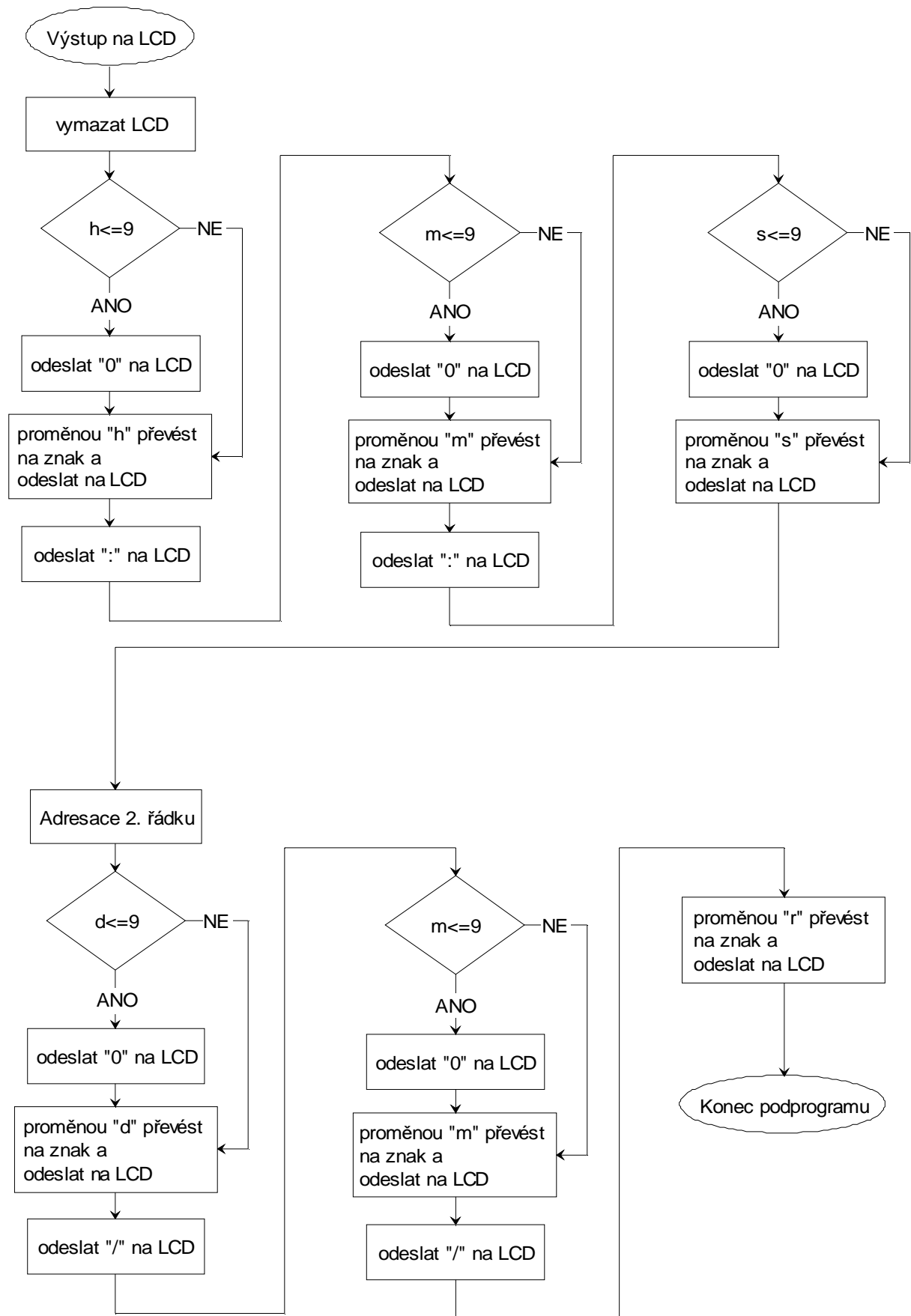


Obr.2 AVR Studio 4

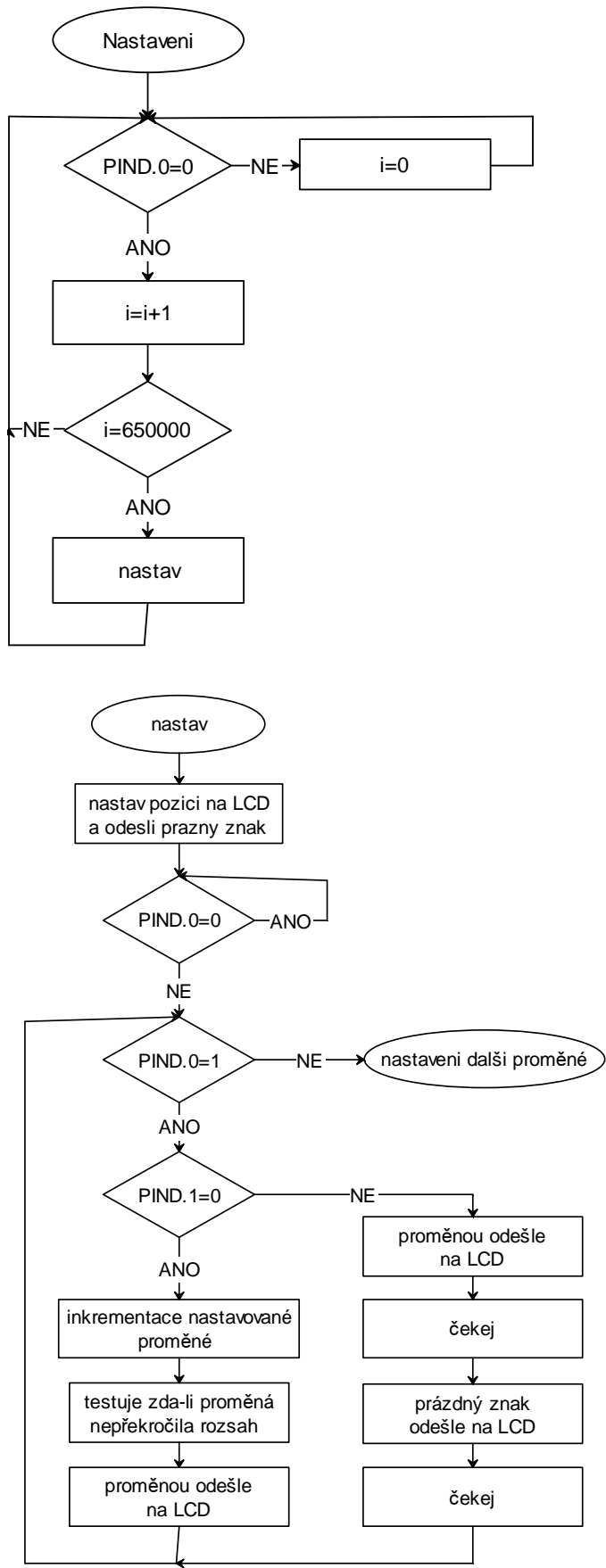
4 VÝVOJOVÉ DIAGRAMY



Vývojový diagram podprogramu, který se spouští když nastane přerušení



Vývojový diagram zobrazování na LCD



Vývojové diagramy pro nastavování

5 PROGRAM

Program pro realizaci digitálních hodin je napsán v programovacím jazyku C.

```
#include <mega16.h>
#include <Delay.h>
#include <stdlib.h>
#include <bcd.h>

// Kde bude připojen LCD
#asm
.equ __lcd_port=0x1B ;PORTA
#endasm
#include <lcd.h>

unsigned char s, mi, h;
unsigned char d=1;
unsigned char m=1;
unsigned int r=2009;
unsigned char cas[2];
unsigned char rok[4];
unsigned long int i;

void vystup_lcd()
{
    lcd_clear();
    lcd_gotoxy(0,0);
    if (h<=9) lcd_putsf("0");
    itoa(h,cas);
    lcd_puts(cas);
    lcd_putsf(":");
    if (mi<=9) lcd_putsf("0");
    itoa(mi,cas);
    lcd_puts(cas);
    lcd_putsf(":");
    if (s<=9) lcd_putsf("0");
    itoa(s,cas);
    lcd_puts(cas);
    lcd_gotoxy(0,1);
    if (d<=9) lcd_putsf("0");
    itoa(d,cas);
    lcd_puts(cas);
    lcd_putsf("/");
    if (m<=9) lcd_putsf("0");
    itoa(m,cas);
    lcd_puts(cas);
    lcd_putsf("/");
    itoa(r,rok);
    lcd_puts(rok);
    // Přerušení nastane při dosažení vrcholu čítače/časovače
    interrupt [TIM1_OVF] void timer1_ovf_isr(void)
    {
        if (++s==60) {
            s=0;
            if (++mi==60) {
                mi=0;
                if (++h==24) {
                    h=0;
                    if (++d==32) {
```



```

    m++;
    d=1;
}
else if (d==31)
{
    if ((m==4) || (m==6) || (m==9) || (m==11)) {
        m++;
        d=1;
    }
}
else if ((d==30) && (m==2) && ((r%4)==0)) {
    m++;
    d=1;
}
else if ((d==29) && (m==2) && ((r%4)!=0)) {
    m++;
    d=1;
}
if (r==13) {
    m=1;
    r++;
}
}
}
}
vystup_lcd();
}

```

```

void init()
{
//Nastavení portu D jako vstup
PORTD=0x00;
DDRD=0x00;
//Nastavení čítače/časovače
TCCR1A=0x03;
TCCR1B=0x14;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x7A;
OCR1AL=0x12;
OCR1BH=0x00;
OCR1BL=0x00;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x04;

// Nastavení LCD
lcd_init(20);
lcd_clear();

// Povolení přerušení
#ifdef sei
asm("sei")
#endif
}

```

```

void main(void)
{
init();
}

```

```

while (1)
{
//Nastavení času pomocí tlačítek
if (PIND.0==0)
{
i++;
if (i==650000)
{
#asm("cli")
lcd_gotoxy(0,0);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
while (PIND.0==0) { }
delay_ms(200);
while (PIND.0)
{
if (PIND.1==0) {
h++;
if (h>=24) h=0;
lcd_gotoxy(0,0);
if (h<=9) lcd_putsf("0");
itoa(h,cas);
lcd_puts(cas);
delay_ms(250);
} else {
lcd_gotoxy(0,0);
if (h<=9) lcd_putsf("0");
itoa(h,cas);
lcd_puts(cas);
i=0;
while ((PIND.0) & (PIND.1) & (i<80000)) i++;
lcd_gotoxy(0,0);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
i=0;
while ((PIND.0) & (PIND.1) & (i<80000)) i++;
}
}
vystup_lcd();
//Nastaveni minut
lcd_gotoxy(3,0);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
while (PIND.0==0) { }
delay_ms(200);
while (PIND.0)
{
if (PIND.1==0) {
mi++;
if (mi>=60) mi=0;
lcd_gotoxy(3,0);
if (mi<=9) lcd_putsf("0");
itoa(mi,cas);
lcd_puts(cas);
delay_ms(250);
} else {
lcd_gotoxy(3,0);
if (mi<=9) lcd_putsf("0");
}
}
}
}

```

```

itoa(mi,cas);
lcd_puts(cas);
i=0;
while ((PIND.0) & (PIND.1) & (i<80000)) i++;
lcd_gotoxy(3,0);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
i=0;
while ((PIND.0) & (PIND.1) & (i<80000)) i++;
}
}
vystup_lcd();
// Nastaveni sekund
lcd_gotoxy(6,0);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
while (PIND.0==0) { }
delay_ms(200);
while (PIND.0)
{
if (PIND.1==0) {
s++;
if (s>=60) s=0;
lcd_gotoxy(6,0);
if (s<=9) lcd_putsf("0");
itoa(s,cas);
lcd_puts(cas);
delay_ms(250);
} else {
lcd_gotoxy(6,0);
if (s<=9) lcd_putsf("0");
itoa(s,cas);
lcd_puts(cas);
i=0;
while ((PIND.0) & (PIND.1) & (i<80000)) i++;
lcd_gotoxy(6,0);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
i=0;
while ((PIND.0) & (PIND.1) & (i<80000)) i++;
}
}
vystup_lcd();
//Nastaveni roku
lcd_gotoxy(6,1);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
while (PIND.0==0) { }
delay_ms(200);
while (PIND.0)
{
if (PIND.1==0) {
r++;
if (r==10000) r=1000;
lcd_gotoxy(6,1);
itoa(r,rok);
lcd_puts(rok);
}
}

```

```

delay_ms(250);
} else {
lcd_gotoxy(6,1);
itoa(r,rok);
lcd_puts(rok);
i=0;
while ((PIND.0) & (PIND.1) & (i<80000)) i++;
lcd_gotoxy(6,1);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
i=0;
while ((PIND.0) & (PIND.1) & (i<80000)) i++;
}
}
vystup_lcd();
//Nastaveni mesice
lcd_gotoxy(3,1);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
while (PIND.0==0) { }
delay_ms(200);
while (PIND.0)
{
if (PIND.1==0) {
m++;
if (m>=13) m=1;
lcd_gotoxy(3,1);
if (m<=9) lcd_putsf("0");
itoa(m,cas);
lcd_puts(cas);
delay_ms(250);
} else {
lcd_gotoxy(3,1);
if (m<=9) lcd_putsf("0");
itoa(m,cas);
lcd_puts(cas);
i=0;
while ((PIND.0) & (PIND.1) & (i<80000)) i++;
lcd_gotoxy(3,1);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
i=0;
while ((PIND.0) & (PIND.1) & (i<80000)) i++;
}
}
vystup_lcd();
//Nastaveni dnu
lcd_gotoxy(0,1);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
while (PIND.0==0) { }
delay_ms(200);
while (PIND.0)
{
if (PIND.1==0) {
d++;
if (d>=32) d=1;

```

```

else if ((d==31) && ((m==4) || (m==6) || (m==9) || (m==11))) d=1;
else if ((d==30) && (m==2) && ((r%4)==0)) d=1;
else if ((d==29) && (m==2) && ((r%4)!=0)) d=1;
lcd_gotoxy(0,1);
if (d<=9) lcd_putsf("0");
itoa(d,cas);
lcd_puts(cas);
delay_ms(250);
} else {
lcd_gotoxy(0,1);
if (d<=9) lcd_putsf("0");
itoa(d,cas);
lcd_puts(cas);
i=0;
while ((PIND.0) & (PIND.1) & (i<80000)) i++;
lcd_gotoxy(0,1);
lcd_putchar(0xFE);
lcd_putchar(0xFE);
i=0;
while ((PIND.0) & (PIND.1) & (i<80000)) i++;
}
}

delay_ms(200);
#asm("sei")
i=0;
}
}
else i=0;

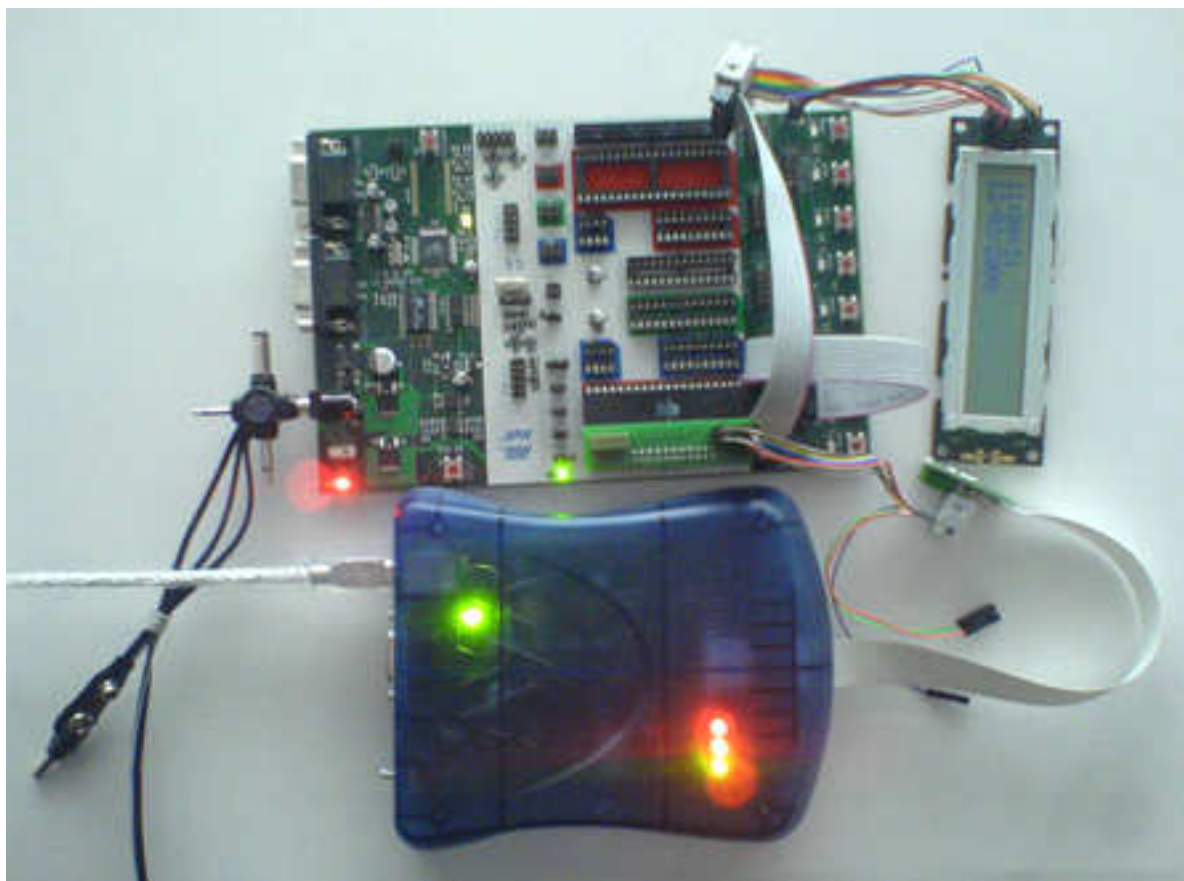
};
}

```

6 TESTOVÁNÍ PROGRAMU

Testování programu jsem prováděl na vývojové desce STK500 od výrobce Atmel s použitím procesoru Atmega16, inteligentního LCD s obvodem HD44780 a krystalem 16MHz. Programování jsem prováděl pomocí AVR JTAGICE mkII.

K portu A je připojen LCD a k portu D jsou připojeny tlačítka.



Obr. 3 Zapojení při testování

7 ZÁVĚR

Cílem bylo naprogramovat digitální hodiny prostřednictvím mikrokontroleru Atmel Atmega16. Při řešení jsem nejprve použil interní RC oscilátor. Tato volba nebyla vhodná, protože hodiny se začaly opožďovat. Dalším krokem bylo použití externího krystalu. Výpočtem a experimentálními pokusy jsem došel k závěru, že nejvhodnější hodnota krystalu je 16 MHz. Za použití tohoto krystalu hodiny mají velkou přesnost.

Nastavování hodin se provádí pomocí dvou tlačítek na desce STK500. Naprogramování ovládacích tlačítek bylo velmi komplikované. Při použití zpoždovací smyčky z knihovny delay.h docházelo k velkým chybám. Pro dané řešení jsem si navrhl svou vlastní zpoždovací smyčku, která je předběžně ukončena stiskem jednoho ze dvou nastavovacích tlačítek.

Podařilo se mě navrhnout velmi přesné digitální hodiny se zobrazováním na dvouřádkovém LCD displeji. Navržený program digitálních hodin lze po určitých úpravách použít i pro jiné zobrazovací jednotky.

SEZNAM POUŽITÉ LITERATURY

- [1] Váňa Vladimír: Atmel AVR programování v jazyce C, BEN, Praha 2003.
- [2] Datasheets Atmega16, dostupné z <http://www.atmel.com>