

K řízení byl použit procesor, neboť algoritmy, které je potřeba zpracovávat jsou natolik složité, že by řešení na bázi logických polí bylo příliš složité a drahé. Zvolen byl procesor typu 8052 (např. firmy Atmel 89C52), který koncepčně vychází z 8051.

Jádro původního procesoru se stále zdokonaluje a vybavuje o další periferie. 89C52 má 8 kB Flash EPROM, což je kapacita, která pro menší aplikace vhodnější než použití externí EPROM. Navíc má výhodu v možnosti vícenásobného přepisu bez nutnosti mazání UV zářením.

Počet přepisovacích cyklů se udává až 10000. Dále má 256 bytů RAM, což usnadňuje práci vzhledem k možnosti definování vlastních proměnných a v případě této práce je to dokonce nutností, neboť se zde zadává mnoho konfiguračních parametrů ukládaných právě do paměti RAM. Ke styku s okolím jsou k dispozici 4 osmibitové vstupně výstupní brány. 89C52 má 3 časovače/čítače, využitelné např. pro vykonávání více činností v režimu přerušení nebo k záznamu vnějších podnětů na pozadí. Pro sériovou komunikaci se používá zabudovaný UART, s volbou různých režimů. Lze volit počty datových a stop bitů, rychlost atd.. Taktování se provádí pomocí externího krystalu a kapacit a procesor zvládá až 24 MHz. Pouzdro je nejčastěji typu DIL40 (zde zvoleno 11Mhz).

## Klávesnice

Klávesnice je zapojena jako matice 4x3. Je připojena přímo na I/O bránu P2, protože je takto lépe ovladatelná.

Čtení se provádí následovně. Vstupy na které jsou připojeny sloupce se nastaví jako vstupní tj. zapíše se na ně log 1. Potom se na výstupech na řádky nechá rotovat log 0 (log 0 je vždy na jednom, na ostatních jsou log 1), a v každém cyklu se přečtou všechny sloupce. Jestliže je na některém z nich log 0, znamená to, že byl spojen s řádkem tj. byla stisknuta klávesa. Zjistí se, který řádek a sloupec se spojily, a je patrné, o jakou klávesu se jedná.

Odpory slouží jako upínací.

## Signalizační LED

Jsou zde použity nízkopříkonové LED (2 mA). Jsou zapojeny na I/O bránu P0 a jejich činnost je tedy přímo určena stavem bitů na konkrétních adresách (viz výše). V případě log 1 svítí, při log 0 jsou zhasnuté.

## Připojení čidlových okruhů

Stavy čidlových okruhů jsou reprezentovány logickými hodnotami, kde log 1 značí připravenost okruhu tj. klidový stav, a log 0 narušení resp. poruchu na čidlovém okruhu. Jednotlivé vstupy jsou připojeny na I/O bránu P1. To znamená, že jsou opět přímo adresovatelné.

## Popis zařízení

Zařízení je velmi jednoduchou zabezpečovací ústřednou spadající do 4. kategorie - tedy bez možnosti rozsáhlejší konfigurace. Vzhledem k nedostupnosti jakýchkoli konkrétnějších specifikací jsem ji založil na TTL logice.

Hlavní ovládací částí je procesor 8052, který má na starosti řízení a vyhodnocování všech událostí ze čtyř nezávislých smyček. (Potencionálně až osmi.)

U čidel byl udělán předpoklad, že jejich klidové výstupní napětí je na úrovni logické 1, v případě, že senzor zaregistruje podnět přepne do logické 0, čímž je vyvolán poplach. Logická 1 jako klidové napětí byla zvolena z důvodu jednoduché ochrany. Příklad poškození čidla totiž bude signalizován jako poplach.

Ve své podstatě je jedno co bude připojeno na danou smyčku a jaký význam bude mít který signál, nicméně je doporučení hodné, aby na každé ze smyček byl jednotný soubor čidel (a doporučuje to tak i norma) - tedy např. na jedné smyčce čidla pohybová na druhé oken a dveří (magnetické snímače), na třetí střežení krytů čidel a na čtvrté např. kouřová čidla.

Jak je vidět z popisu hardware, skládá se celá ústředna z vlastního procesoru na jehož tři vstupní brány jsou připojeny klávesnice (4x3 klávesy), LED a akustická signalizace, a vstupy z čidlových smyček.

## Software

Obecně pro programování menších microcontrollerů je vhodné použít přímo strojový kód. Protože se mi však nepodařilo sehnat žádný volně šiřitelný kompilátor přímo pro assembler, napsal jsem celý program v jazyce C++ pro nějž již byl dostupný freeware kompilátor SDCC (<http://sdcc.sourceforge.net/>). Od laděn byl program na dostupném simulátoru 8052.

### Základní stavba programu

Základním úkolem procesoru a tedy programu je obsluhovat vstupně výstupní zařízení - tedy v tomto případě vstupní klávesnici a čidlové smyčky a výstupní indikaci stavu na LED a sirénu.

Ústředna má dva základní módy činnosti, mezi kterými přechází zadáním správného hesla, které v tomto případě tvoří čtyřmístný číselný kód. Po zapnutí napájení ústředny je kód defaultně nastaven na '0000'. Ústředna svou inicializaci oznámí rozsvícením a zhasnutím všech signalizačních diod, po té by již měla zůstat svítit jen dioda signalizující napájení a ústředna se nachází v neaktivním módu, tedy nereaguje na vstupy z čidel, pouze čeká na zadání kódu.

Pokud bude kód zadán správně přejde ústředna do aktivního módu a uživatel má 5s na to, aby opustil střežené prostory. Na zadání každé, krom první, klávesy je časový limit 2s. Pokud je kód zadán špatně nebo není stisknuta klávesa v daném limitu je to oznámeno blikáním všech 4 okružových diod a je dána možnost opravy. Pokud je zadán podruhé špatně, nebo není zadána klávesa v časovém limitu, je vyvolán poplach.

V případě, že během aktivního módu dojde k narušení, musí být do 5s zadáno heslo jinak je opět vyvolán poplach - opticky je signalizována smyčka na které došlo k narušení a je spuštěna siréna na dobu 25s případně do doby než je zadán správný vstupní kód. LED signalizace zůstává až do okamžiku zadání správného kódu nebo do nového narušení, neboť po 25s přejde ústředna opět do aktivního módu.

Pokud přecházíme zadáním kódu z módu aktivního do neaktivního je nám dána možnost kód změnit. Signalizováno je to semaforovým blikáním diod 1-2 a 3-4.

Dále uvádím komentovaný výpis celého programu

## ustredna.c.

---

```
#include "reg51.h"
// #include "stdio.h" //pro debugovani

typedef unsigned char byte;

byte Pass[4];
byte LEDMode;
byte UMode;
byte LEDState;
byte CounterLED;
unsigned int CounterAl;
byte naruseni;

/*
    EchoPort: byte; Absolute P0; // LED, akustika na portu 2
    LoopPort: byte; Absolute P1; // Vstup ze smycek
    KyePort: byte; Absolute P2; // I/O klavesnice
*/

void Wait(unsigned char Delay)
{
    byte i;
    unsigned int j;
    for (i=0; i<=Delay; i++)
        for (j=0; j<=11520; j++) //melo by byt asi 0.05s
            {}
}

byte porovnej(byte PassO[3], byte PassZ[3])
{
    byte i;

    for (i=0; i<=3; i++) if (PassO[i]!=PassZ[i]) return(0);
    return(1);
}

byte KBD() //vlastni pristup na HW klavesnice
{
    byte cisl;

    P2=0x0df; //zapis nulu do 1. radky
    Wait(1); //pockej
    do {
        cisl=P2;
        Wait(1);
    } while (P2 != cisl);
    P2=0x0ff;
    cisl=cisl & 0x0f;
    switch (cisl) {
        case 0x07: return(1);
        case 0x0b: return(2);
        case 0x0d: return(3);
    }

    P2=0x0bf; //zapis nulu do 2. radky
    Wait(1); //pockej
    do {
        cisl=P2;
        Wait(1);
    }
```

```

} while (P2 != cisl);
P2=0xff;
cisl=cisl & 0x0f;
switch (cisl) {
    case 0x07: return(4);
    case 0x0b: return(5);
    case 0x0d: return(6);
    case 0x0e: return(0);
}

P2=0x07f;          //zapis nulu do 2. radky
Wait(1);           //pockej
do {
    cisl=P2;
    Wait(1);
} while (P2 != cisl);
P2=0xff;
cisl=cisl & 0x0f;
switch (cisl) {
    case 0x07: return(7);
    case 0x0b: return(8);
    case 0x0d: return(9);
}
return(10);
}

void AlarmCheck() interrupt 1          //snad preruseni 1 - Timer 0 owerflow
{
    byte alarm;

    alarm=P1;
    if ((alarm!=0x0ff) && (UMode==1)) {
//      LEDMode=4;                      //mod alarmu vyvolaneho narusenim smycky
        UMode=2;                        //mod ustredny - alarm - jeste neni dan na venek
        naruseni=~alarm;                //rosvit diodu prislusne smycky
        CounterAl++;                    //zvys counter
    }
//    else CounterAl++;                  //Counter>=100 = 5s pauza pred vyvolanim poplachu
    if (UMode==2) {
        if ((CounterAl>=100) && (CounterAl<=3000)) {
            LEDState=naruseni;          //dej do LEDDtate stav pri vyvolani poplachu
            LEDMode=4;                  //mod poplach
            CounterAl++;                //v dobe od 5s po naruseni do 30s po nar. sirena
        }
        else {
            LEDMode=5;                  //jinak pouze indikace stavu na diodach (zustane do dalsiho
naruseni)
            UMode=1;                    //prejdi zpet do aktivniho modu
            CounterAl=0;
        }
    }
    if (UMode==0) {
        CounterAl=0;                    //pokud neaktivni stav - nuluj Counter
        LEDMode=1;                      //nastav diody na nesviti
        naruseni=0;
    }

    switch (LEDMode) {
        case 0: LEDState=0x02f; break; //rozsvit vsechny diody
        case 1: LEDState=0x20; break;  //zhasni vsechny diody krom power
        case 2: if (CounterLED==20) {   //blikej po sekunde vsemi ctyrmi diodami
            LEDState=0x02f;
            CounterLED++;
        }
        if (CounterLED==0) {
            LEDState=0x20;
            CounterLED++;
        }
    }
}

```

```

        if (CounterLED==40) CounterLED=0;
        else CounterLED++;
        break;
    case 3: if (CounterLED==20) {          //vlackovy semaforek po sekunde
        LEDState=0x023;
        CounterLED++;
    }
    if (CounterLED==0) {
        LEDState=0x02b;
        CounterLED++;
    }
    if (CounterLED==40) CounterLED=0;
    else CounterLED++;
    break;
    case 4: LEDState=LEDState | 0x10; break;    //poplach - pridej sirenu
    case 5: LEDState=LEDState & 0x10; break;    //poplach ale jiz bez sireny
    }
    P0=LEDState;                                //posli stav na port
}

//scanuje klavesnici, pocita cas na klavesu
byte ScanKey()
{
    byte znak;
    byte wtime;

    wtime=255;                                //interval kdy musi byt zadana klavesa
    znak=10;                                  //inertni znak
    while ((wtime!=0) || (znak==10)) {        //dokud neutece cas, nebo neni stisknuta klavesa
        scanuj KBD
        znak=KBD();
        wtime--;
        Wait(1);
    }
    return(znak);
}

/* na zacatku mainu ji odmital volat... kdo vi proc...
void UInitx()                                //INICIALIZACE
{
    byte i;

    // printf("Here!!");
    P0=0xff;                                //rozsvit vsechny diody na uvitanou
    Wait(20);                                //pockej
    UMode=0;                                //mod ustredny - neaktivni
    for (i=0; i<=3; i++)                    //vynuluj heslo
        Pass[i]=0;
    P0=0;                                    //zhasni diody

    //nastaveni preruseni (zrejme ve strojaku)
    CounterAl=0;
    //Timer0 = 46079 = 0.05s => 40*timer1=2s
    _asm
    mov _TH0,179;                            //maximalni hodnota Timeru - horni bajt
    mov _TL0,255;                            //dolni bajt
    mov _TMOD,1;                            //16bitovy mod citace
    setb _ET0;                                //umozni preruseni pri pretečení timeru0
    setb _TR0;                                //spust Timer0
    setb _EA;                                //povol vsechna preruseni
    _endasm;
}
*/

byte WCode(byte Typ)                        //WaitCode - cekej na zadani kodu
{
    byte TmpPass[4],i;

```

```

if (Typ==1) {                                     //Typ cekani 1 - cekame na zadani poprve
    TmpPass[0]=10;
    while (TmpPass[0]==10) TmpPass[0]=ScanKey(); //cekej na prvni stisk klavesy
    for (i=1; i<=3; i++) {
        TmpPass[i]=ScanKey();
        if (TmpPass[i]==10) return (0);
    }
    if (porovnej(Pass,TmpPass)) return (1);
    else return (0);
}
if (Typ==2) {                                     //Typ 2 - cekame na opravu
    Wait(40);                                     //chvili pockej at se vzpamatuje (2s)
    LEDMode=2;                                    //varovne blikame diodami
    for (i=0; i<=3; i++) { //druha sance zadat heslo
        TmpPass[i]=ScanKey();
        if (TmpPass[i]==10) return (0); //spatne nebo nestihl
    }
    if (porovnej(Pass,TmpPass)) { //bylo zadano spravne?
        return (1); //ano
        LEDMode=1; //zhasni LED
    }
    else return (0); //zadano spatne, nebo nestihl
}
}

main()
{
    byte i;
    byte TmpPass[4];

//inicializace -----
    P0=0xff;                                     //rozsvit vsechny diody na uvitanou
    Wait(40);                                    //pockej
    UMode=0;                                     //mod ustredny - neaktivni
    for (i=0; i<=3; i++) //vynuluj heslo
        Pass[i]=0;
    P0=0x20;                                     //zhasni diody krom power diody - ta bude svitit do vypnuti
    proudu

//nastav LED mod na 0
    LEDMode=0;

//nastav smycky bez poplachu; poplach = 1 na vstupu P1.X
    P1=0xff;

    naruseni=0;

//nastaveni preruseni (zrejme ve strojaku)
    CounterAl=0;                                //Counter pro pocitani casu do spusteni alarmu po
    naruseni
    CounterLED=0;                                //Counter pro blikani diod
                                                //oba se zvysuji pri preruseni Timer) overflow
//Timer0 = 46079 = 0.05s => 40*timer1=2s
_asm
    mov _TH0,179;                               //maximalni hodnota Timeru - horni bajt
    mov _TL0,255;                               //dolni bajt
    mov _TMOD,1;                                //16bitovy mod citace
    setb _ET0;                                   //umozni preruseni pri pretečení timeru0
    setb _TR0;                                   //spust Timer0
    setb _EA;                                    //povol vsechna preruseni
_endasm;
//konec inicial -----

do {
    if (WCode(1) && (UMode==0)) {
        _asm

```

```

        clr _EA;                                //zakaz preruseni a tedy CheckAlarm
        _endasm;
        Wait(100);                             //cekej nez odejde (5s)
        _asm
        _setb _EA;                             //povol preruseni a tedy CheckAlarm
        _endasm;
        UMode=1;                               //nastav mod ustredny jako aktivni
    }
    else if (WCode(2) && (UMode==0)) {          //druhy pokus o zadani
        _asm
        _clr _EA;
        _endasm;
        Wait(100);                             //cekej nez odejde
        _asm
        _setb _EA;
        _endasm;
        UMode=1;
    }
    else UMode=2;                               //mod naruseni
    if (WCode(1)) UMode=0;                     //cekej na opetovne zadani spravneho kodu - pro
deaktivaci
    else if (WCode(2)) UMode=0;
    else UMode=2;

    if (UMode==0) {                             //pri zadani spravneho hesla pro deaktivaci
        LEDMode=3;                             //moznost zmenit heslo - roluj diodami
        for (i=0; i<=3; i++) {
            TmpPass[i]=ScanKey();
            if (TmpPass[i]==10) {
                TmpPass[0]=10;
                break;
            }
        }
        if (TmpPass[0]!=10) for (i=0; i<=3; i++) Pass[i]=TmpPass[i];
    }
} while (0);
}

```