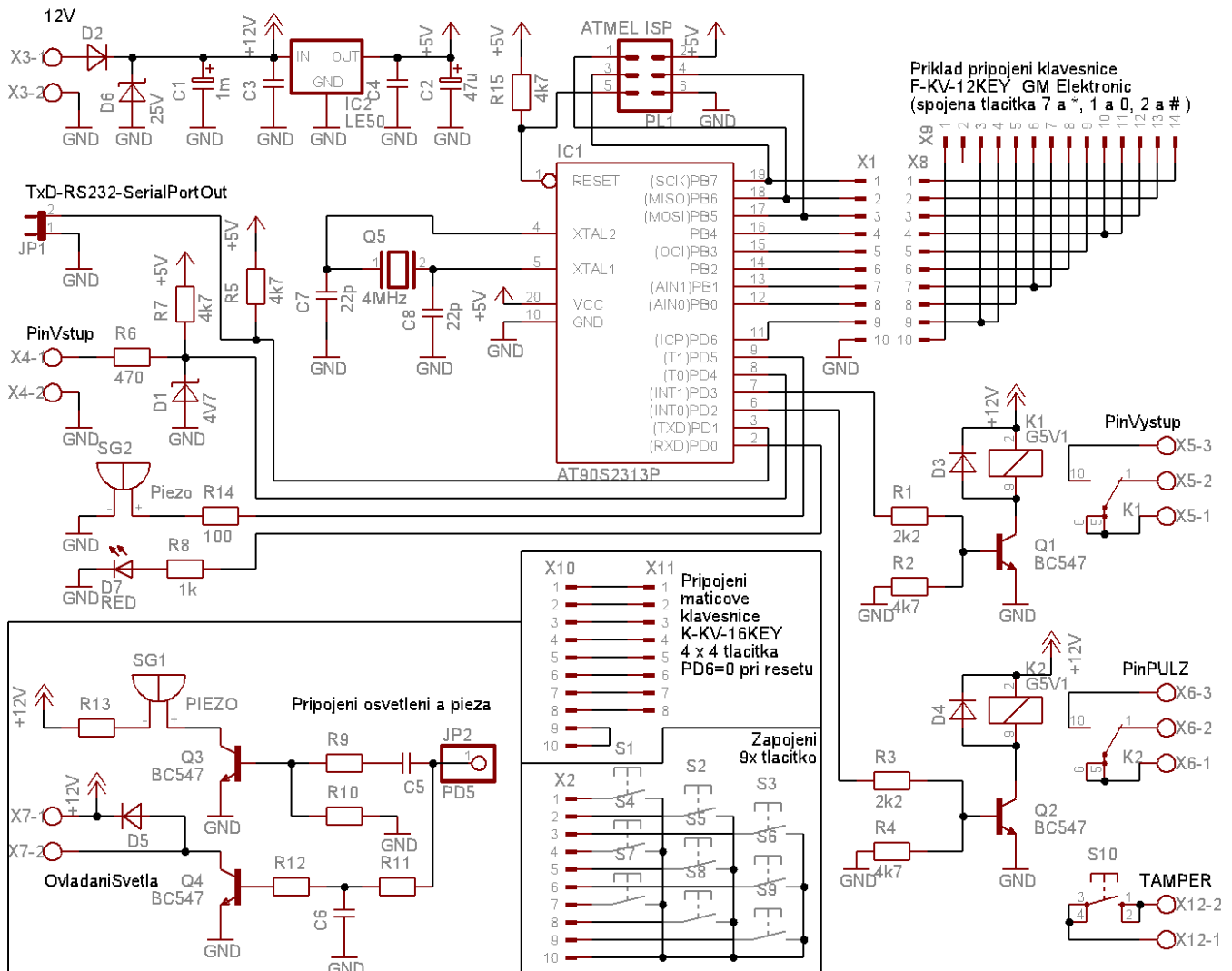


Kódová klávesnice Alarm_keypad

Emil Hašl

Tuto konstrukci kódové klávesnice jsem vymyslel jako doplněk k alarmu. Slouží ke střežení prostor, v kterých se mají pohybovat pouze lidé, kteří znají kód. Funkce je jednoduchá, pokud vstoupíte do místnosti, zaregistruje Vás pohybové čidlo, začne se odpočítávat čas příchodu, po zadání platného kódu se odpočet vypne, jinak se vyvolá poplach na nadřazeném alarmu nebo se spustí připojený pager, který někam oznámí poplach. Při opouštění hlídaných prostor zadáme platný kód a máme zase nějaký čas na odchod. Konstrukce i schéma je velmi jednoduché, vše řídí jednočipový mikroprocesor řady AVR AT90S2313. Další součástky slouží jen na ochranu vstupů a na posílení a oddělení výstupů. Tento procesor má řadu výhod: vnitřní paměť EEPROM, možnost programování v aplikaci ISP (lze programovat po zapájení), možnost konfigurace PINů jako vstupy 3tí-stav nebo s vnitřním pull-up (posílení na +) nebo jako výstupy, a další. Vše lze najít v katalogovém listu. Jistá nevýhoda je v tom, že výrobce již tento procesor asi nevyrábí, ale půjde po úpravě programu nahradit obvodem AT TINY2313.

Schéma zapojení:



Obrázek 1: Schéma zapojení: 3 možnosti připojení klávesnic, připojení ovládání světla

Popis vlastností:

- K procesoru kódové klávesnice lze připojit 2 typy klávesnic:
 - klávesy připojené přímo každá zvlášť, lze přímo připojit 9 tlačítek, 8 tlačítek na port PB, 1 tlačítko na PD6, další lze připojit přes diody, aby se sepnulo více vstupů najednou (max. 255 kombinací, při resetu musí být PD6 = 1, jindy se PD6 = 0 bere jako stisk kód 0xFF)
 - nebo maticová klávesnice o maximální konfiguraci 4 x 4 (při resetu musí být PD6=0)
- Kódová klávesnice standardně reaguje a pamatuje si 4 kódy po 4 cifrách (lze softwarově upravit)
 - Proti sabotáži se chrání povolením zadat kód jen 5x, pak se klávesnice na nějakou dobu zablokuje, po stisku klávesy se také zahájí odpočet příchodu (toto lze deaktivovat)
 - Platné kódy a nastavení módu se naučí ve speciálním servisním módu a uloží do vnitřní paměti EEPROM (proto nezáleží na fyzickém zapojení kláves, ani na umístění číslic, každé klávese se přiřadí sejmутý 8-bit kód)
 - Stav zajištěno a odjištěno si procesor také uchovává pro případ výpadku napájení v EEPROM
- Vstupy a výstupy kódové klávesnice, LED a zvuková signalizace:
 - má 1 vstup na čidlo pohybu, 1 výstup poplachu a další výstup, který lze nakonfigurovat buď jako pulz při platném kódu nebo kopíruje stav zajištěno/odjištěno
 - Stav kódové klávesnice lze poznat na stavové LED diodě (bliká – zajištěno, svítí – odpočítává odchod, bliká rychle – odpočítává příchod, nesvítí – odjištěno)
 - Klávesnice krátce pípá při stisku kláves, dále pípne při zadání platného nebo neplatného kódu, má i pomocný výstup na ovládání osvětlení – používá se pin na piezo viz. schéma (např. Led podsvícení klávesnice, nebo po oddělení a posílení velké světlo)
 - Výstup všeho, co se děje v procesoru, lze sledovat na sériové lince, jen výstup v konfiguraci 9600 Bd, 8 bit data, 1 stop bit
- Další různé módy kódové klávesnice:
 - Klávesnice může pracovat bez funkcí alarmu, jen pošle puls po zadání platného kódu
 - dále může pracovat jen jako jednoduchý terminál bez kontroly kódů (výstup sériová linka)

Příklad datového výstupu na sériové lince (konfigurace 9600Bd - 8bit - 1stopbit)

```
Alarm KeyPad by SeTeL.cz ver. 1.01 11/2004.
enter code for save, please.
0x20 0x10 0x08 0x04 0xC0 0x30 0x80 0x40 0xFF 0xFF 0xFF 0xFF 0x02 0x01 0x06 0xFF

*Sabotage. 0x20 0x10 0x08 0x04 c0 = Code OK.
-Alarm off.
0xC0 0x30 0x80 0x40 c1 = Code OK.
Alarm off -> on.
+Alarm on.
#Alarm.
!!! Alarm !!!
*Sabotage. 0x01 0x08 0x10 0x20 = Code Error.
0x02 0x10 !!! Alarm !!!
*Sabotage. 0x01 0x10 = Code Error.
0x01 0x10 0x30 0x01 = Code Error.
0x01 0x10 0x01 0x10 = Code Error.
0x01 0x10 0x01 0x10 = Code Error.
5 times code error.
!!! Alarm !!!
*Sabotage. 0x40 0x40 0x40 0x40 = Code Error.
0xFF 0xFF 0xFF 0xFF c2 = Code OK.
-Alarm off.
```

Všechny texty a konstanty uložené v procesoru a ve vnitřní EEPROM

```
// DEFINICE KONSTANT PROGRAMU

#define CODE_LENGTH      4      // pocet stisku v jednom kodu
#define CODE_NUMBER      4      // pocet ulozenych kodu

#define STISKREAKCE      5      // klavesnice reakce po 50 ms
#define STISKDALSI      100    // klavesnice opakovani dalsi po 1000 ms
#define INICNTRESET     60     // reset stisku po 60 sekundach
#define INICNTSTOP      240    // 4 minuty nejde klavesnice
#define INICNTKODFALSE  5      // 5 spatnych pokusu
#define DOBAPIPANI      10     // pocet sekund doby pipani = 10 sekund

// TEXTY SERIOVEHO PORTU

flash char TXT_Ini[]    = "\rAlarm KeyPad by SeTeL.cz ver. 1.01 11/2004.\r";
flash char TXT_IniSet[] = "enter code for save, please.\r";
flash char TXT_IniOk[]  = "\rcodes saved. Settings: ";
flash char TXT_IniOk2[] = "\rsettings saved.\r";
flash char TXT_CodeOk[] = " = Code OK.\r";          // pismeno "O" = platny kod
flash char TXT_CodeError[] = " = Code Error.\r";    // "E" = neplatny kod
flash char TXT_AlarmCON[] = "Alarm off -> on.\r";    // ">" = odpocitava odchod
flash char TXT_AlarmOn [] = "+Alarm on.\r";         // "+" = alarm zapnut
flash char TXT_AlarmOff[] = "-Alarm off.\r";        // "-" = alarm vypnut
flash char TXT_Poplach [] = "#Alarm.\r";           // "#" = odpocitava prichod
flash char TXT_PoplachS[] = "*Sabotage. ";         // "*" = odpocitava prichod
flash char TXT_Alarm [] = "!!! Alarm !!!\r";        // "!" = poplach
flash char TXT_CodeError5[]={ INICNTKODFALSE+'0', " times code error.\r"};
// "5" = vickrat spatny kod

// OBSAZENI PORTU

#define PortStisk      PINB      // PinB vstup stisku z klavesnic
#define PortStiskOut   PORTB     // výstup na buzení maticove klavesnice

#define PortIn         PIND      // vstupy ostatni
#define PortOut        PORTD     // vystupy ostatni
#define PortDir        DDRD      // nastaveni smeru portu

#define PinStavLed     0         // PD0 - pripojeni stavove LED prez odpor 1k na GND
#define PinTxD         1         // PD1 - TxD - vystup ser. linky, nebo aktivace nastavovaciho modu
#define PinPulz        2         // PD2 - Vystup pulzu 2 sec 1 = kod ok, nebo stav zajisteno
#define PinVystup      3         // PD3 - Vystup alarmu 1 = poplach, puls 2 sec
#define PinVstup       4         // PD4 - Vstup od cidla (zapnem pullup), neaktivni spojeno s GND
#define PinTon         5         // PD5 - pripojeni pieza prez odpor 100R na GND
#define Pin4x4         6         // PD6 - Vstup 1 = portB je 8bit klavesnice, 0 = klavesnice 4x4

// BITY V NASTAVENI

#define BitAlarm       0        // stav alarmu 1 = zamceno
#define BitCodeDis     1        // zakaze dlouhe pipani a nacistani kodu, pokud se pouzije jako terminal
#define BitAlarmDis    2        // zakaze alarm, funguje jen jako kodovy zamek, pouze vystup pulzu
#define BitSabotageDis 3        // je zakazan sabotazni poplach po 5 zadanich kodu, vyhlasi poplach
#define BitPulzPrek    4        // vystupni pulz bude preklapeci podle BitAlarm 1= zajisteno
#define BitPulzInv     5        // zmena na 0=zajisteno
#define BitLightEn     6        // je povoleno sviceni prez PinTon

#define IniByteStav    0b00000001 // inicializace stavoveho byte
#define IniStavMaska   0x7F       // maska na nastaveni

// OBSAZENI VNITRNI PAMETI EEPROM

eeprom byte eeByteStav = IniByteStav; // stavovy byte nacistany z eeprom
eeprom byte DOBAODCHODU = 30;        // pocet sekund odchodu = 30 sekund
eeprom byte DOBAPRICHODU= 30;        // pocet sekund prichodu = 30 sekund
eeprom byte DOBASVICENI = 60;        // pocet sekund doby sviceni = 1 munuta

eeprom byte code[CODE_NUMBER] [ CODE_LENGTH ] = { { 1,1,1,1}, {2,2,2,2}, {3,3,3,3}, {4,4,4,4}};
```

Nastavení konfigurace:

Připojte si na výstupní sériovou linku X1 nějaký terminál (počítač se sériovým portem nebo Serial2USB převodníkem) přepnutý na 9600 Bd, 1 stopbit, žádná parita. Pokud tuto možnost nemáte, musíte se spokojit s posloucháním a odpočítáváním.

Do speciálního servisního módu se dostanete tak, že **před připojením napájení zkratujete výstup sériového portu X1** na Gnd. Po zapnutí tento zkrat odstraňte a čekejte na přihlášení procesoru logem na sériovém portu (pokud sériovou linku nesledujete, čekejte na pípnutí, asi 10 sec). Pak namačkejte čtyři přístupové kódy po 4 klávesách. Poslední stisknutou klávesnici si zapamatujte.

Procesor znovu pípne a ukáže konfiguraci: „eeByteStav“ - popis viz. výpis programu – BITY V NASTAVENÍ - po naprogramování main.eep má ukázat 0x01. Každým stiskem klávesy jiné než dříve zapamatované se toto číslo zvyšuje o 1 až do 0x7F pak znova od 0x00. Po stisku zapamatované klávesy si procesor tuto konfiguraci zapamatuje do eeprom.

Konfiguraci samozřejmě můžete rovnou naprogramovat do eeprom, můžete i měnit časy příchodu a odchodu.

Snímací kódy kláves je možné odvodit pouze z jednoduché klávesnice: spojení na Gnd u PB0 = 1, PB1 = 2, PB2 = 4, PB3 = 8, PB4 = 10, PB5 = 20, PB6=40, PB7=80, PD6=FF (vše hexadecimální čísla, při použití diod a spínání více vstupů najednou jako součet).

Způsob připojení:

Kódová klávesnice musí mít vstup X2 trvale sepnut na GND, např. detektorem pohybu nebo dveřním kontaktem. Po rozpojení se začíná odpočítávat doba příchodu (viz. Konstanty v programu) a je čas na zadání platného kódu. Při nezadání správného kódu v tomto čase přepne relé K2. Naopak při odchodu máte po zadání platného kódu čas na odchod. Všechny tyto funkce se lépe pochopí při testech na stole se zapojeným terminálem na sériové lince.

Kódová klávesnice může alarm také přímo ovládat pulzem 2 sec na relé K1.

Spínač S1 je jako TAMPER rozpínací kontakt, musí rozepnout při poškození nebo rozebrání víka klávesnice.

Použití je ale více univerzální, např. ovládání elektromagnetického zámku u společných dveří, apod.

Závěr:

Popsaná konstrukce slouží jen jako jednoduchý doplněk k pageru nebo alarmu.

Program do procesoru je volně ke stažení na [www stránkách autora](http://www.seTEL.cz), proto můžete použít k naprogramování procesoru i ISP programátor a naprogramovat procesor až po zapájení.

Spojení na autora: Emil Hašl, email: emil@SeTeL.cz

Program do procesoru a popsany zdrojový kód je na: <http://www.SeTeL.cz>