

## **[Zámek + Alarm]**

**22-04-2022 (klávesnice, LCD, tlačítko)**

### **Zadání:**

1. Vstupní kód snímejte z klávesnice a kód zobrazte na LCD displeji. Vstupní kombinaci na 1s zobrazte na LCD displeji a pak nahraďte znakem \*  
Při stisku správné kombinaci kláves (1234) zobrazte na LCD displeji text **vstupte.** Byla-li stlačena jiná kombinace zobrazte, **chyba.**
2. Při trojnásobné špatné kombinaci zobrazte na LCD displeji **alarm (i zvukově).** A zablokujte další zadávání čísel.
3. Upravte program tak, aby kód po otevření zámku byl možno měnit. Po zadání servisního kódu, umožněte přepsat otevírací kód svým kódem. Při neznalosti servisního kódu není možno měnit otevírací kód. Servisní kód = (321).

### **Řešení na stránce 2**

#### **Bugy:**

- Při dalším pokusu o zadání hesla, se to buguje.
- Před dalším pokusem zadání hesla, je třeba 1x zmáčknout # jinak se to bude psát do 1. řádku
- Zatím nefunguje servisní režim

```

6  #include "stm32f4xx.h"           // Device header
7  #include <stdlib.h>
8  #include <stdio.h>
9
10 void SysTick_Handler(void);
11 uint8_t getkey();
12 void Delay (uint32_t ms);
13 void LCD_config(void);
14 void LCD_ctrlWR(uint16_t cmd);
15 void LCD_dataWR(uint16_t cmd);
16 void LCD_ini (void);
17 void puts_LCD(int radek, char* ukaz);
18 void klavesnice_config(void);
19
20 int main() {
21
22     // Promenne //
23     uint8_t klavesa, pwd[] = "          ", vstup[] = "          ";
24     int i = 0, pwd_pocet = 0;
25     int password, vstup_uzivatele, pocet_chyb = 0, servis_mozny = 0;
26     char vysledek[8], hvezdicky[8], hvezdickyl[8];
27
28     SystemCoreClockUpdate();
29     SysTick_Config(SystemCoreClock/10000);
30     LCD_config();
31     LCD_ini();
32     klavesnice_config();
33
34
35 while(1) {
36
37     do{                               // Nastavovani hesla, jako spravce
38         klavesa = getkey();
39         if(klavesa == '#'){
40             break;
41         }
42         else
43         {
44             pwd[i] = klavesa;
45             puts_LCD(1, pwd);
46
47             pwd_pocet++;
48             i++;
49         }
50     }while(klavesa != '#');
51

```

```

52     for(i = 0; i < pwd_pocet; i++) // nahrazovani cisel na hvezdicky (delay 1000ms)
53     {
54         hvezdicky[i] = '*';
55     }
56
57     Delay(1000);
58     puts_LCD(1, hvezdicky);
59     Delay(1000);
60     i = 0;
61
62     do{                                     // Uzivatel zadava heslo, bud bude spravne nebo spatne
63         klavesa = getkey();
64         if(klavesa == '#')
65         {
66             break;
67         }
68         else
69         {
70             vstup[i] = klavesa;
71             hvezdicky1[i] = '*';
72             puts_LCD(2, vstup);
73             Delay(10000);
74             puts_LCD(2, hvezdicky1);
75             i++;
76         }
77     }while(klavesa != '#');
78
79     password = atoi(pwd);
80     vstup_uzivatele = atoi(vstup);
81
82     if (password == vstup_uzivatele) // Pri spravnem heslu, se zpristupni servisni rezim
83     {
84         strcpy(vysledek, "Vstupte");
85         servis_mozny = 1;           // servisni rezim povolen/mozny
86
87         klavesa = getkey();         // vstup do servisniho rezimu
88         if(klavesa == '*')
89         {
90             puts_LCD(1, "SERVIS ");
91
92         }
93     }
94     else if(pocet_chyb == 3)         // 3 spatny pokus => alarm
95     {
96         puts_LCD(1, "ALARM! ");
97         puts_LCD(2, "ALARM! ");

```

```
99  
100     else  
101     {  
102         strcpy(vysledek, "Chyba");  
103         pocet_chyb++;  
104     }  
105     puts_LCD(2, vysledek);  
106 }  
107  
108  
109  
110     return 0;  
111 }
```