

Univerzitet u Sarajevu
Elektrotehnički fakultet
Ugradbeni sistemi 2023/24.

Izvještaj za laboratorijsku vježbu 3

Višebitni digitalni ulazi i izlazi

Ime i prezime: **Kanita Kadušić**
Broj index-a: **19327**

Sarajevo, mart 2024.

Sadržaj

| | | |
|----------|--|-----------|
| 1 | Pseudokod | 1 |
| 1.1 | Zadatak 1 | 1 |
| 1.2 | Zadatak 2 | 3 |
| 1.3 | Zadatak 3 – izbor 2 | 4 |
| 2 | Analiza programskog rješenja | 5 |
| 2.1 | Zadatak 1 | 5 |
| 2.2 | Zadatak 2 | 5 |
| 2.3 | Zadatak 3 – izbor 2 | 6 |
| 3 | Korišteni hardverski resursi | 7 |
| 3.1 | LPC1114ETF | 7 |
| 3.2 | picoETF | 7 |
| 4 | Povezivanje komponenti sa razvojnim sistemima | 8 |
| 4.1 | Zadatak 1 | 8 |
| 4.2 | Zadatak 2 | 9 |
| 4.3 | Zadatak 3 – izbor 2 | 10 |
| 5 | Zaključak | 11 |
| 6 | Prilog | 12 |
| 6.1 | Zadatak 1: Izvorni kôd | 12 |
| 6.2 | Zadatak 2: Izvorni kôd | 14 |
| 6.3 | Zadatak 3 – izbor 2: Izvorni kôd | 16 |

1 Pseudokod

1.1 Zadatak 1

```
act ← output(LED_ACT)  
leds ← {output(LED0..7)}
```

```
matrix ← 
$$\begin{bmatrix} 1 & 2 & 3 & A \\ 4 & 5 & 6 & B \\ 7 & 8 & 9 & C \\ * & 0 & \# & D \end{bmatrix}$$

```

```
rows ← {output(dp16,15,17,18)}  
columns ← {input(dp9,10,11,13)}
```

```
function ledsOff()  
  for i in 0..7  
    value(leds[i]) ← 0  
  pause(0.5)
```

```
function scanKeypad()  
  for i in 0..3  
    value(rows[i]) ← 1  
  
    for j in 0..3  
      if value(columns[j]) = 1  
        value(rows[i]) ← 0  
        return matrix[i][j]  
  
    value(rows[i]) ← 0  
  
  return ''
```

```

value(act) ← 0
ledsOff()

key ← ''
released ← true

while true
    key ← scanKeypad()
    if key = ''
        released ← true

    if key ≠ '' and released = true
        if key ≥ 1 and key ≤ 8
            value(leds[key - 1]) ← 1
        else if key = C
            ledsOff()

    released ← false

```

1.2 Zadatak 2

```
counter ← 0
released ← true
auto ← false

while true
  if checkButtons(0,0,0,0)
    released ← true

  if not checkButtons(0,0,0,0) and released
    released ← false

    if checkButtons(1,0,0,0) and not auto
      counter ← (counter + 1) mod 10000
    else if checkButtons(0,1,0,0) and not auto
      counter ← (counter - 1) mod 10000
    else if checkButtons(0,0,1,0) and not auto
      counter ← 0
    else if checkButtons(0,0,0,1)
      auto ← not auto

  if auto
    counter ← (counter + 1) mod 10000
    for i in 0..24
      displayNumber(counter)

displayNumber(counter)
```

1.3 Zadatak 3 – izbor 2

```
function enterPin()
    released  $\leftarrow$  true

    while true
        key  $\leftarrow$  scanKeypad()

        if key == ''
            released  $\leftarrow$  true
        else if key == '#'
            displayCharacter(' ')
            return pin[-4:]

        if key  $\geq$  '0' and key  $\leq$  '9' and released
            released  $\leftarrow$  false
            pin  $\leftarrow$  pin + key

        displayNumber(int(pin))
        pause(0.01)

function checkPin(x)
    attempt  $\leftarrow$  0

    while enterPin()  $\neq$  x
        setDigits(0,0,0,0)
        attempt  $\leftarrow$  attempt + 1

        if attempt == 3
            startCountdown()
            attempt  $\leftarrow$  0
        else
            for i in 0..4
                displayCharacter('–')
                pause(0.5)
                displayCharacter('–')
                pause(0.5)

    setDigits(0,0,0,0)
    for i in 0..4
        value(decimalPoint)  $\leftarrow$  0
        pause(0.5)
        value(decimalPoint)  $\leftarrow$  1
        pause(0.5)
```

2 Analiza programskog rješenja

Napomena: Ponavljajući dijelovi kôda u različitim zadacima će se analizirati samo prilikom prvog pojavljivanja.

2.1 Zadatak 1

[1-15] uvoz odgovarajućih biblioteka, te deklaracija i inicijalizacija potrebnih varijabli

[17-22] funkcija koja gasi prethodno deklarirane LED diode

[24-38] funkcija koja očitava unos ostvaren putem matrične tastature

[40-58] glavni program koji realizira čitanje znaka unesenog putem matrične tastature i vrši odgovarajući prikaz koristeći LED diode, pritom vodeći računa da se skeniranje tastature odvija samo na uzlaznu ivicu sata

2.2 Zadatak 2

[1-7] uvoz odgovarajućih biblioteka, te deklaracija i inicijalizacija potrebnih varijabli

[9-28] tabela koja se sastoji od znakova čiji se prikaz na 7-segmentnom displeju želi omogućiti, uz potrebne vrijednosti odgovarajućih segmenata za prikaz željenog znaka na displeju, pritom uzimajući u obzir da displej koristi zajedničku anodu

[30-33] funkcija koja upali/ugasi pojedine cifre na 7-segmentnom displeju

[35-37] funkcija koja upali željenu cifru na 7-segmentnom displeju

[39-42] funkcija koja prikazuje željeni znak na 7-segmentnom displeju

[44-49] funkcija koja prikazuje željeni broj na 7-segmentnom displeju

[51-55] funkcija koja provjerava stanja tastera

[57-61] funkcija koja provjerava ispravnost ožičenja displeja sa razvojnim sistemom

[63-89] funkcija koja implementira brojač korištenjem prethodno navedenih funkcija

2.3 Zadatak 3 – izbor 2

[1-53] analogno ili isto kao u prethodnom zadatku

[55-65] funkcija koja očitava unos ostvaren putem matrične tastature

[67-85] funkcija koja implementira unos šifre korištenjem prethodno datih funkcija

[87-90] funkcija koja realizira odbrojavanje počevši od 10, uz prikaz cifara na četverocifrenom 7-segmentnom displeju

[92-115] funkcija koja vrši provjeru ispravnosti unesene šifre

3 Korišteni hardverski resursi

3.1 LPC1114ETF

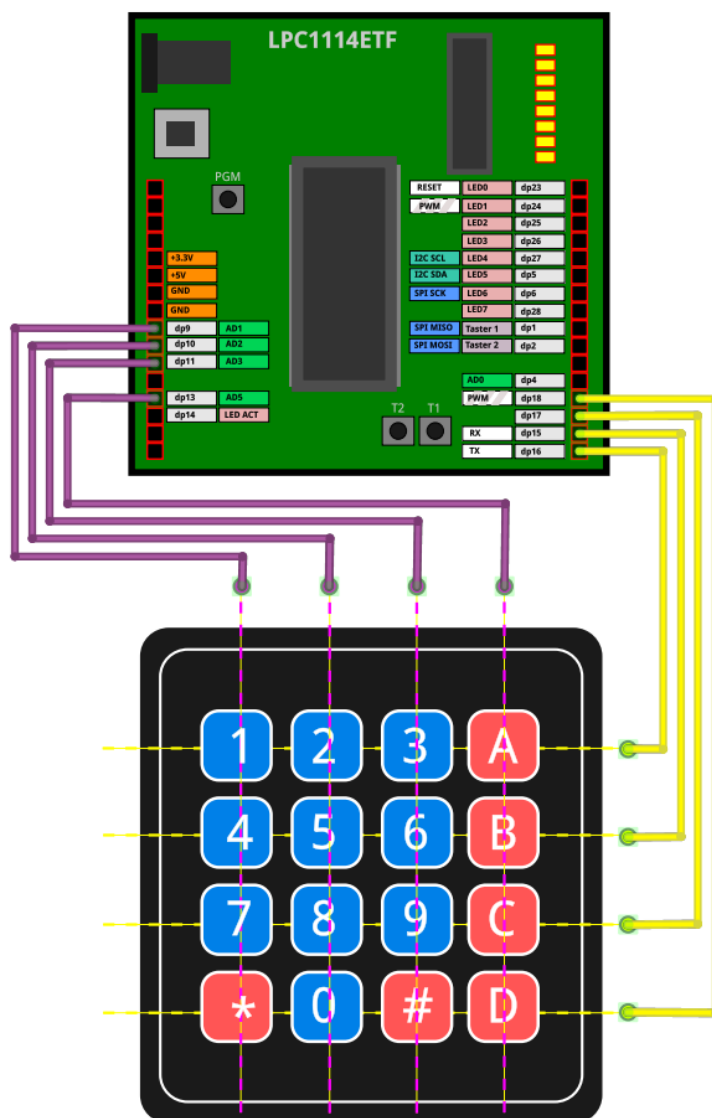
| | <i>Komponenta</i> | <i>Opis</i> | <i>Količina</i> |
|---|--------------------|--------------------------|-----------------|
| 1 | LPC1114FN28 | mikrokontroler | 1 |
| 2 | matrična tastatura | ulazna komponenta | 1 |
| 3 | LED dioda | digitalni izlaz | 8 |
| 4 | USB A kabal | napajanje i komunikacija | 1 |
| 5 | LED_ACT pin | uključivanje LED dioda | 1 |

3.2 picoETF

| | <i>Komponenta</i> | <i>Opis</i> | <i>Količina</i> |
|---|-------------------------------|--------------------------|-----------------|
| 1 | RP2040 | mikrokontroler | 1 |
| 2 | matrična tastatura | ulazna komponenta | 1 |
| 3 | 4-cifreni 7-segmentni displej | izlazna komponenta | 1 |
| 4 | taster | digitalni ulaz | 4 |
| 5 | konektor | uvezivanje sistema | |
| 6 | USB A – USB Micro | napajanje i komunikacija | 1 |

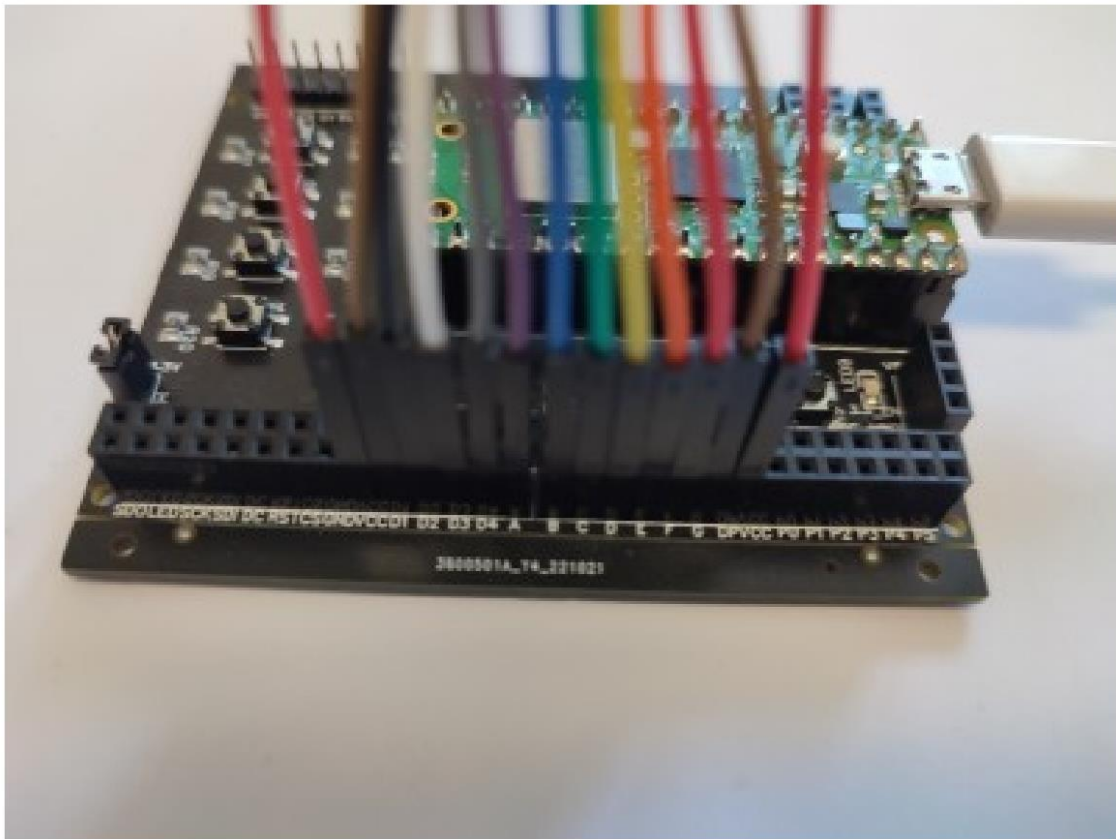
4 Povezivanje komponenti sa razvojnim sistemima

4.1 Zadatak 1



Slika 1: Shema povezivanja matrične tastature sa razvojnim sistemom LPC1114ETF

4.2 Zadatak 2



Slika 2: Povezivanje 4-cifrenog 7-segmentnog displeja sa razvojnim sistemom picoETF

4.3 Zadatak 3 – izbor 2



Slika 3: Povezivanje matrične tastature sa razvojnim sistemom picoETF

5 Zaključak

Svi zadaci u okviru Laboratorijske vježbe 3 su bili veoma zanimljivi.

Pisanje kôda za Zadatak 2 je iziskivalo dosta vremena, jer su „sitnice“ odlučivale da li će prikaz na displeju biti adekvatan. Posljednji zadatak je objedinio zahtjeve prvog i drugog zadatka, stoga nije bio težak za implementirati.

Nova znanja, stečena izradom Laboratorijske vježbe 3, obuhvataju rad sa dodatnim hardverskim resursima, konkretno, matričnom tastaturom i 4-cifrenim 7-segmentnim displejem.

6 Prilog

6.1 Zadatak 1: Izvorni kôd

```
01: #include "mbed.h"
02: #include "lpc1114etf.h"
03:
04: DigitalOut act(LED_ACT);
05: DigitalOut leds[] = {LED0, LED1, LED2, LED3, LED4, LED5, LED6, LED7};
06:
07: char matrix[4][4] = {
08:     {'1', '2', '3', 'A'},
09:     {'4', '5', '6', 'B'},
10:     {'7', '8', '9', 'C'},
11:     {'*', '0', '#', 'D'}
12: };
13:
14: DigitalOut rows[] = {dp16, dp15, dp17, dp18};
15: DigitalIn columns[] = {dp9, dp10, dp11, dp13};
16:
17: void ledsOff() {
18:     for (int i = 0; i < 8; i++)
19:         leds[i].write(0);
20:
21:     wait_us(5e5);
22: }
23:
24: char scanKeypad() {
25:     for (int i = 0; i < 4; i++) {
26:         rows[i].write(1);
27:
28:         for (int j = 0; j < 4; j++)
29:             if (columns[j].read()) {
30:                 rows[i].write(0);
31:                 return matrix[i][j];
32:             }
33:
34:         rows[i].write(0);
35:     }
36:
37:     return ' ';
38: }
39:
```

```

40: int main() {
41:     act.write(0);
42:     ledsOff();
43:
44:     char key = ' ';
45:     bool released = true;
46:
47:     for (;;) {
48:         key = scanKeypad(); // read the current key pressed
49:         if (key == ' ') released = true; // set the flag when all keys are released
50:
51:         if (key != ' ' && released) { // if key is pressed AND previous was released
52:             if (key >= '1' && key <= '8') leds[int(key) - 48].write(1);
53:             else if (key == 'C') ledsOff();
54:
55:             released = false; // clear the flag to indicate that key is still pressed
56:         }
57:     }
58: }

```

6.2 Zadatak 2: Izvorni kôd

```
01: from machine import Pin
02: from time import sleep
03: from collections import OrderedDict
04:
05: buttons = [ Pin(i, Pin.IN) for i in range(4) ]
06: digits = [ Pin(i, Pin.OUT) for i in range(4, 8) ]
07: segments = [ Pin(i, Pin.OUT) for i in range(8, 15) ]
08:
09: tableAnode = OrderedDict({
10:     '0': (0, 0, 0, 0, 0, 0, 1),
11:     '1': (1, 0, 0, 1, 1, 1, 1),
12:     '2': (0, 0, 1, 0, 0, 1, 0),
13:     '3': (0, 0, 0, 0, 1, 1, 0),
14:     '4': (1, 0, 0, 1, 1, 0, 0),
15:     '5': (0, 1, 0, 0, 1, 0, 0),
16:     '6': (0, 1, 0, 0, 0, 0, 0),
17:     '7': (0, 0, 0, 1, 1, 1, 1),
18:     '8': (0, 0, 0, 0, 0, 0, 0),
19:     '9': (0, 0, 0, 0, 1, 0, 0),
20:     'A': (0, 0, 0, 1, 0, 0, 0),
21:     'B': (1, 1, 0, 0, 0, 0, 0),
22:     'C': (0, 1, 1, 0, 0, 0, 1),
23:     'D': (1, 0, 0, 0, 0, 1, 0),
24:     'E': (0, 1, 1, 0, 0, 0, 0),
25:     'F': (0, 1, 1, 1, 0, 0, 0),
26:     '*': (1, 0, 0, 1, 0, 0, 0),
27:     ' ': (1, 1, 1, 1, 1, 1, 1)
28: })
29:
30: def setDigits(x):
31:     displayCharacter(' ')
32:     for i in range(4):
33:         digits[i].value(x[i])
34:
35: def turnOnDigit(x):
36:     setDigits([1, 1, 1, 1])
37:     digits[x].value(0)
38:
39: def displayCharacter(x):
40:     values = tableAnode.get(x)
41:     for i in range(7):
42:         segments[i].value(values[i])
43:
44: def displayNumber(x):
45:     x = [ chr((x // i) % 10 + ord('0')) for i in (1000, 100, 10, 1) ]
46:     for i in range(4):
47:         turnOnDigit(i)
48:         displayCharacter(x[i])
49:         sleep(0.01)
50:
```



```

51: def checkButtons(x):
52:     for i in range(4):
53:         if buttons[i].value() != x[i]:
54:             return False
55:     return True
56:
57: def testSystem():
58:     setDigits([0, 0, 0, 0])
59:     for key in tableAnode.keys():
60:         displayCharacter(key)
61:         sleep(0.5)
62:
63: def startCounter():
64:     counter = 0
65:     released = True
66:     auto = False
67:
68:     while True:
69:         if checkButtons([0, 0, 0, 0]):
70:             released = True
71:
72:         if not checkButtons([0, 0, 0, 0]) and released:
73:             released = False
74:
75:         if checkButtons([1, 0, 0, 0]) and not auto:
76:             counter = (counter + 1) % 10000
77:         elif checkButtons([0, 1, 0, 0]) and not auto:
78:             counter = (counter - 1) % 10000
79:         elif checkButtons([0, 0, 1, 0]) and not auto:
80:             counter = 0
81:         elif checkButtons([0, 0, 0, 1]):
82:             auto = not auto
83:
84:         if auto:
85:             counter = (counter + 1) % 10000
86:             for i in range(25):
87:                 displayNumber(counter)
88:
89:         displayNumber(counter)
90:
91: # testSystem()
92: startCounter()

```

6.3 Zadatak 3 – izbor 2: Izvorni kôd

```
001: from machine import Pin
002: from time import sleep
003: from collections import OrderedDict
004:
005: digits = [ Pin(i, Pin.OUT) for i in range(4, 8) ]
006: segments = [ Pin(i, Pin.OUT) for i in range(8, 15) ]
007: point = Pin(15, Pin.OUT)
008:
009: matrix = (
010:     ('1', '2', '3', 'A'),
011:     ('4', '5', '6', 'B'),
012:     ('7', '8', '9', 'C'),
013:     ('*', '0', '#', 'D')
014: )
015:
016: rows = [ Pin(i, Pin.OUT) for i in [21, 22, 26, 27] ]
017: columns = [ Pin(i, Pin.IN, Pin.PULL_DOWN) for i in [0, 1, 2, 3] ]
018:
019: tableAnode = OrderedDict({
020:     '0': (0, 0, 0, 0, 0, 0, 1),
021:     '1': (1, 0, 0, 1, 1, 1, 1),
022:     '2': (0, 0, 1, 0, 0, 1, 0),
023:     '3': (0, 0, 0, 0, 1, 1, 0),
024:     '4': (1, 0, 0, 1, 1, 0, 0),
025:     '5': (0, 1, 0, 0, 1, 0, 0),
026:     '6': (0, 1, 0, 0, 0, 0, 0),
027:     '7': (0, 0, 0, 1, 1, 1, 1),
028:     '8': (0, 0, 0, 0, 0, 0, 0),
029:     '9': (0, 0, 0, 0, 1, 0, 0),
030:     '-': (1, 1, 1, 1, 1, 1, 0),
031:     ' ': (1, 1, 1, 1, 1, 1, 1)
032: })
033:
034: def setDigits(x):
035:     displayCharacter(' ')
036:     for i in range(4):
037:         digits[i].value(x[i])
038:
039: def turnOnDigit(x):
040:     setDigits([1, 1, 1, 1])
041:     digits[x].value(0)
042:
043: def displayCharacter(x):
044:     values = tableAnode.get(x)
045:     for i in range(7):
046:         segments[i].value(values[i])
047:
```

```

048: def displayNumber(x):
049:     x = [ chr((x // i) % 10 + ord('0')) for i in (1000, 100, 10, 1) ]
050:     for i in range(4):
051:         turnOnDigit(i)
052:         displayCharacter(x[i])
053:         sleep(0.01)
054:
055: def scanKeypad():
056:     for i in range(4):
057:         rows[i].value(1)
058:
059:         for j in range(4):
060:             if columns[j].value():
061:                 rows[i].value(0)
062:                 return matrix[i][j]
063:
064:         rows[i].value(0)
065:     return ' '
066:
067: def enterPin():
068:     pin = "0000"
069:     released = True
070:
071:     while True:
072:         key = scanKeypad()
073:
074:         if key == ' ':
075:             released = True
076:         elif key == '#':
077:             displayCharacter(' ')
078:             return pin[-4:]
079:
080:         if key >= '0' and key <= '9' and released:
081:             released = False
082:             pin = pin + key
083:
084:         displayNumber(int(pin))
085:         sleep(0.01)
086:
087: def countdown10():
088:     for i in "9876543210":
089:         displayCharacter(i)
090:         sleep(1)
091:

```

```

092: def checkPin(x):
093:     point.value(1)
094:     attempt = 0
095:
096:     while enterPin() != x:
097:         setDigits([0, 0, 0, 0])
098:         attempt = attempt + 1
099:
100:         if attempt == 3:
101:             countdown10()
102:             attempt = 0
103:         else:
104:             for i in range(5):
105:                 displayCharacter('-')
106:                 sleep(0.5)
107:                 displayCharacter(' ')
108:                 sleep(0.5)
109:
110:     setDigits([0, 0, 0, 0])
111:     for i in range(5):
112:         point.value(0)
113:         sleep(0.5)
114:         point.value(1)
115:         sleep(0.5)
116:
117: checkPin("1234")

```