

**Keypad control using PIC16F887 MCU**

Computers and computerized systems in automatics

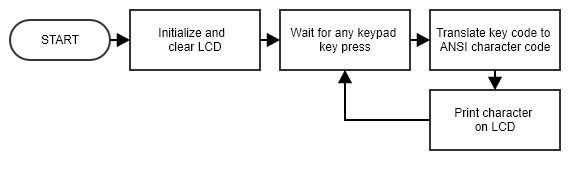
Task 1

Done by: EKSfmu-16 gr. st. Arūnas Butkus

Checked by: prof. Algirdas Baškys

**Work goal:** create a program in C language for PIC16F887 MCU, which would allow printing hexdecimal value output and signs \* and # on included LCD display using 4x4 keypad on EasyPIC 6 development system.

Algorithms used in tasks.



**Fig. 1.** Program algorithm.

Results analysis.

Main issue is figuring out how keypad presses could be pulled and how to print to LCD. MikroC libraries have the required function to pull keypad button press and it only requires that the port to which the keypad is connected be defined as “keypadPort”. Having done that the key that was pressed code is pulled by Keypad\_Key\_Click() function. Key code is then translated into character ANSI code utilizing switch structure.

Data display on LCD is also handled by libraries provided by MikroC IDE. For operation this library requires 12 registers to be named specific names and then it works. To start of Lcd\_Init() is called to initialize LCD display. Then few Lcd\_Cmd() commands are called: parameter \_LCD\_CLEAR clears the LCD screen and parameter \_LCD\_CURSOR\_OFF hides the cursor, that shows where would cursor utilizing commands output data.

For data output I used Lcd\_Chr() function, but the way I implemented it instead I should’ve used Lcd\_Chr\_CP(). That would’ve saved a few memory segments and few machine cycles involved in comparison of registers and possible increment/assignment. The difference between these is that former requires user to specify coordinates where the character should be put, whereas the latter one prints at the cursor point.

Conclusions.

Having done task 2 I now see few locations for improvement in the solution of this task, but the initial solution is functional so I’ll leave it as is for now.

Source code.

01 /\*

02 Sudarykite programa C kalba mikrovaldikliui PIC16F887,

03 kuri leistu naudojantis EasyPIC 6 stende esancia klaviatura

04 4x4 ikelti i LCD ekranA skaicius nuo 0 iki 9, raides A,B,C,D

05 ir ženklus\*, #. Programa sudarykite naudodamiesi

06 Mikro C kompiliatoriumi.

07 \*/

08

09 // LCD connection

10 sbit LCD\_RS at RB4\_bit**;**

11 sbit LCD\_EN at RB5\_bit**;**

12 sbit LCD\_D4 at RB0\_bit**;**

13 sbit LCD\_D5 at RB1\_bit**;**

14 sbit LCD\_D6 at RB2\_bit**;**

15 sbit LCD\_D7 at RB3\_bit**;**

16

17 sbit LCD\_RS\_Direction at TRISB4\_bit**;**

18 sbit LCD\_EN\_Direction at TRISB5\_bit**;**

19 sbit LCD\_D4\_Direction at TRISB0\_bit**;**

20 sbit LCD\_D5\_Direction at TRISB1\_bit**;**

21 sbit LCD\_D6\_Direction at TRISB2\_bit**;**

22 sbit LCD\_D7\_Direction at TRISB3\_bit**;**

23

24 // keypad connection

25 char keypadPort at PORTD**;**

26

27 unsigned short kp**,** col **=** 1**,** row **=** 1**;**

28

29 void main**()** **{**

30 Keypad\_Init**();** // keypad init

31 ANSEL **=** 0**;** // make I/O digital

32 ANSELH **=** 0**;**

33

34 // LCD setup

35 Lcd\_Init**();**

36 Lcd\_Cmd**(**\_LCD\_CLEAR**);**

37 Lcd\_Cmd**(**\_LCD\_CURSOR\_OFF**);**

38

39 **while** **(**1**)** **{**

40 kp **=** 0**;** // reset key press variable

41

42 // wait for keypress

43 **while** **(!**kp**)** **{**

44 kp **=** Keypad\_Key\_Click**();**

45 **}**

46

47 // interpret the keypress

48 **switch** **(**kp**)** **{**

49 **case** 1**:** kp **=** 49**;** **break;** // 1

50 **case** 2**:** kp **=** 50**;** **break;** // 2

51 **case** 3**:** kp **=** 51**;** **break;** // 3

52 **case** 4**:** kp **=** 65**;** **break;** // A

53 **case** 5**:** kp **=** 52**;** **break;** // 4

54 **case** 6**:** kp **=** 53**;** **break;** // 5

55 **case** 7**:** kp **=** 54**;** **break;** // 6

56 **case** 8**:** kp **=** 66**;** **break;** // B

57 **case** 9**:** kp **=** 55**;** **break;** // 7

58 **case** 10**:** kp **=** 56**;** **break;** // 8

59 **case** 11**:** kp **=** 57**;** **break;** // 9

60 **case** 12**:** kp **=** 67**;** **break;** // C

61 **case** 13**:** kp **=** 42**;** **break;** // \*

62 **case** 14**:** kp **=** 48**;** **break;** // 0

63 **case** 15**:** kp **=** 35**;** **break;** // #

64 **case** 16**:** kp **=** 68**;** **break;** // D

65 **}**

66

67 Lcd\_Chr**(**row**,** col**,** kp**);** // print it on LCD

68

69 // following is playing around

70 col**++;**

71 **if** **(**col **>** 16**)** **{**

72 row**++;**

73 col **=** 1**;**

74 **}**

75 **if** **(**row **>** 2**)** **{**

76 row **=** 1**;**

77 **}**

78 **}**

79 **}**

Electrical Schemes

