

EasyLV-18F v6™

User manual

All MikroElektronika's development systems represent irreplaceable tools for programming and developing microcontroller-based devices. Carefully chosen components and the use of machines of the last generation for mounting and testing thereof are the best guarantee of high reliability of our devices. Due to simple design, a large number of add-on modules and ready to use examples, all our users, regardless of their experience, have the possibility to develop their project in a fast and efficient way.

Development System

 **MikroElektronika**

SOFTWARE AND HARDWARE SOLUTIONS FOR EMBEDDED WORLD ...making it simple

TO OUR VALUED CUSTOMERS

I want to express my thanks to you for being interested in our products and for having confidence in mikroElektronika.

The primary aim of our company is to design and produce high quality electronic products and to constantly improve the performance thereof in order to better suit your needs.



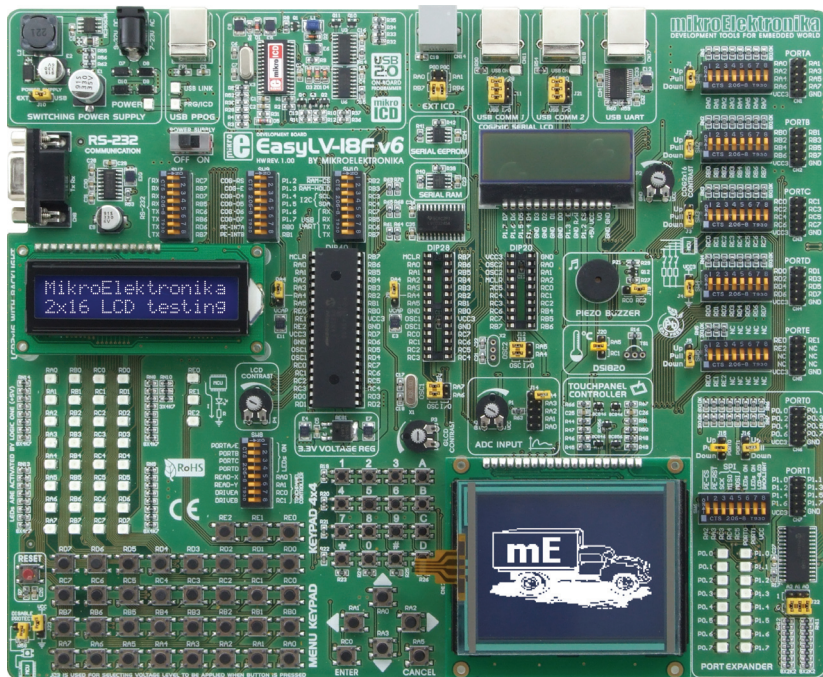
Nebojsa Matic
General Manager

TABLE OF CONTENTS

Introduction to EasyLV-18F v6 Development System	4
Key Features	5
1.0. Connecting the System to a PC	6
2.0. Supported Microcontrollers	7
3.0. On-board LV18PICprog Programmer	8
4.0. mikroLCD (In-Circuit Debugger)	9
5.0. ICD Connector	10
6.0. 3.3V Voltage Regulator	10
7.0. Power Supply	11
8.0. USB Communication Module	12
9.0. A/D Converter Test Inputs	13
10.0. USB UART Module	14
11.0. RS-232 Communication Module	15
12.0. Serial EEPROM Module	16
13.0. RAM Module	17
14.0. DS1820 Temperature Sensor	18
15.0. Piezo Buzzer	19
16.0. LEDs	20
17.0. 4x4 Keypad	21
18.0. MENU Keypad	22
19.0. Push Buttons	23
20.0. On-board 2x16 LCD	24
21.0. 2x16 LCD	25
22.0. 128x64 Graphic LCD	26
23.0. Touch Panel	27
24.0. I/O Ports	28
25.0. Port Expander (Additional I/O Ports)	30

Introduction to EasyLV-18F v6 Development System

The *EasyLV-18F v6™* development system provides a development environment for experimenting with PIC® microcontrollers from Microchip®. The system includes an on-board programmer with mikroICD providing an interface between the microcontroller and a PC. You are simply expected to write a program in one of the PIC compilers, generate a .hex file and program your microcontroller using the on-board *LV18PICprog™* programmer. Numerous modules, such as 128x64 graphic LCD, alphanumeric 2x16 LCD, port expander, on-board 2x16 LCD etc, are provided on the board and allow you to easily simulate the operation of the target device.



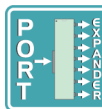
Full-featured development system for PIC microcontroller based devices



On-board USB 2.0 programmer



On-board alphanumeric 2x16 LCD



Port expander provides an easy I/O expansion by 2 additional ports



Graphic LCD with backlight



NEW

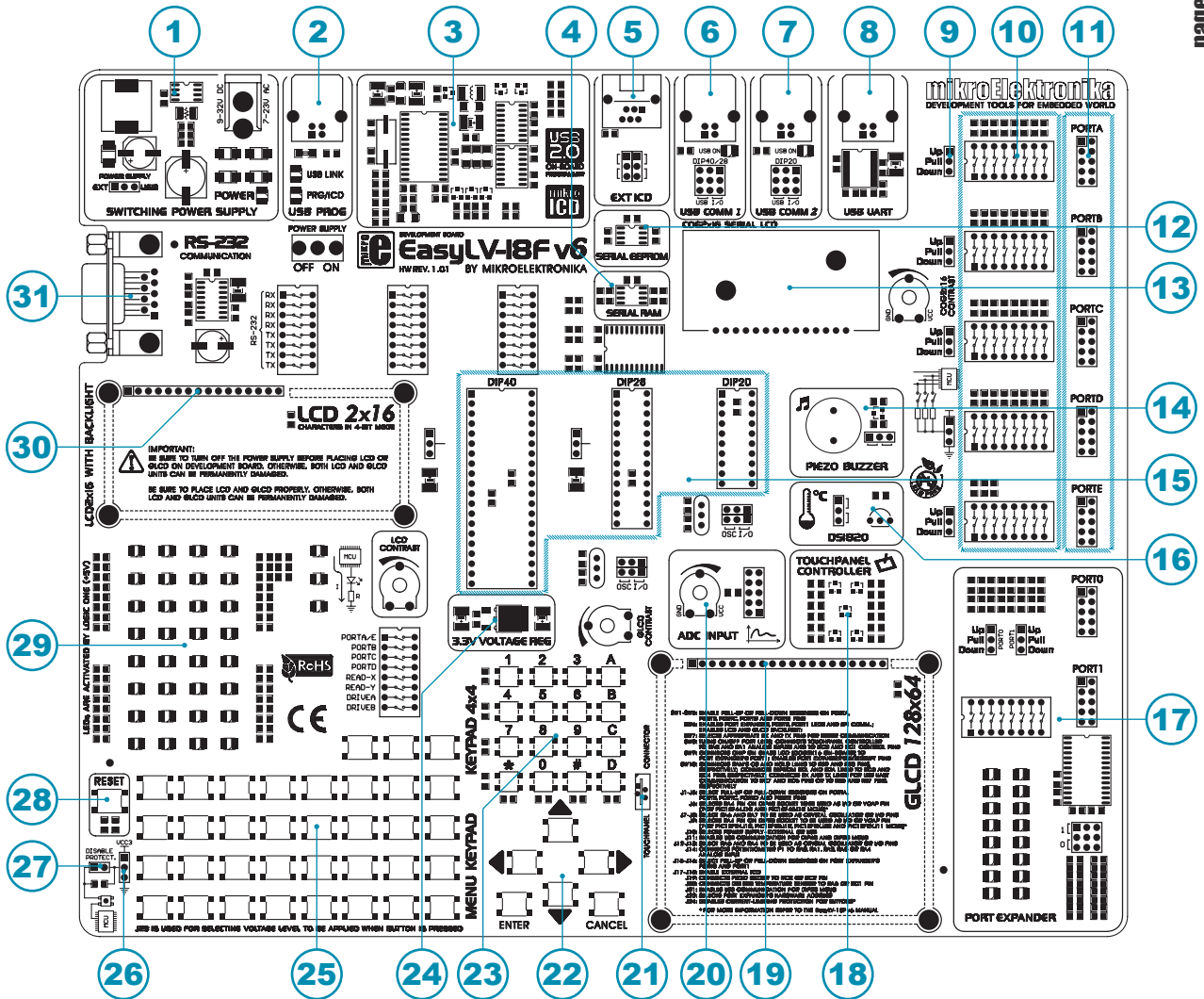
The *Lv18PicFLASH™* program for programming provides a complete list of all supported microcontrollers. The latest version of this program with updated list of supported microcontrollers can be downloaded from our website at www.mikroe.com

Package contains:

Development system: EasyLV-18F v6
 CD: product CD with relevant software
 Cables: USB cable
 Documentation: Manuals for EasyLV-18F v6 and mikroICD, quick guide for installing USB drivers, electrical schematic of the system

System specification:

Power Supply: over an AC/DC connector (7-23V AC or 9-32V DC); or over a USB cable for programming (5V DC)
 Power consumption: ~40mA in idle state when all on-board modules are off
 Dimension: 26,5 x 22cm (10,4 x 8,6inch)
 Weight: 416g (0.91lbs)



Key Features

1. Power supply voltage regulator
2. On-board programmer's USB connector
3. USB 2.0 programmer with mikroICD support
4. Serial RAM module
5. Microchip's programmer (ICD2 or ICD3)
6. USB communication connector 1
7. USB communication connector 2
8. USB UART Module
9. Jumper for pull-up/pull-down resistor selection
10. DIP switches enable pull-up/pull-down resistors
11. I/O port connectors
12. Serial EEPROM module
13. On-board 2x16 LCD
14. Piezo buzzer
15. Microcontroller sockets
16. DS1820 temperature sensor
17. Port expander
18. Touch panel controller
19. Graphic LCD connector
20. A/D converter test inputs
21. Touch panel connector
22. MENU keypad
23. 4x4 keypad
24. 3.3V voltage regulator
25. Push buttons simulate microcontroller digital inputs
26. Jumper for selecting pins' logic state via push buttons
27. Jumper for shortening protective resistor
28. Reset button
29. 35 LEDs indicate pins' logic state
30. Alphanumeric LCD connector
31. RS-232 communication module

1.0. Connecting the System to a PC

Step 1:

Follow the instructions provided in the relevant manuals and install the *Lv18PicFLASH* program and the USB driver from the product CD. USB drivers are essential for the proper operation of the on-board programmer.

In case you already have one of the Mikroelektronika's compilers installed on your PC, there is no need to reinstall USB drivers as they are already installed along with the compiler.

Step 2:

Use the USB cable to connect the *EasyLV-18F v6* development system to a PC. One end of the USB cable, with a USB connector of B type, should be connected to the development system, as shown in Figure 1-2, whereas the other end of the cable with a USB connector of A type should be connected to a PC. When establishing a connection, make sure that jumper J10 is placed in the USB position as shown in Figure 1-1.

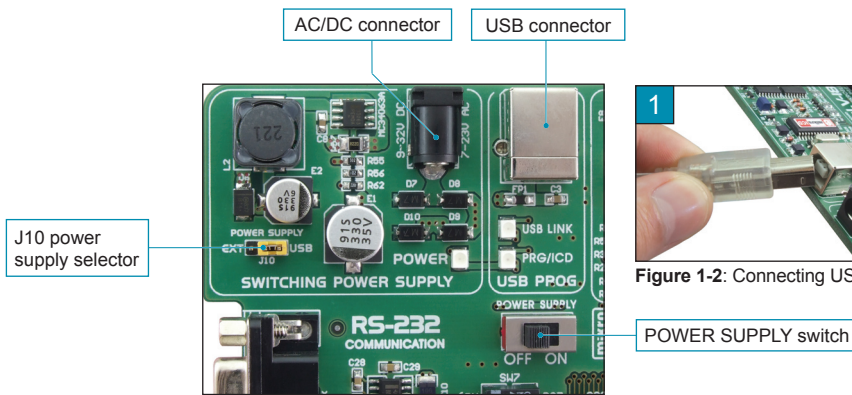


Figure 1-1: Power supply

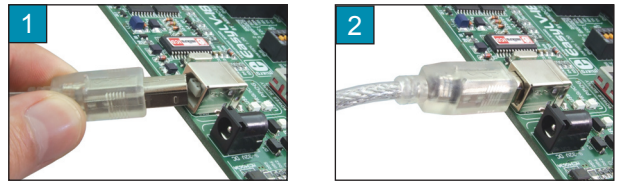


Figure 1-2: Connecting USB cable

Step 3:

Turn on your development system by setting the POWER SUPPLY switch to ON position. Two LEDs marked as POWER and USB LINK will be automatically turned on indicating that your development system is ready to use. Use the on-board programmer *LV18PICprog* and *Lv18PicFLASH* program to dump a hex code into the microcontroller and employ the system to test and develop your projects.

NOTE: If some additional modules are used, such as LCD, GLCD etc, it is necessary to place them properly on the development system while it is turned off. Otherwise, either can be permanently damaged. Refer to figure below for the proper placing of the additional modules.

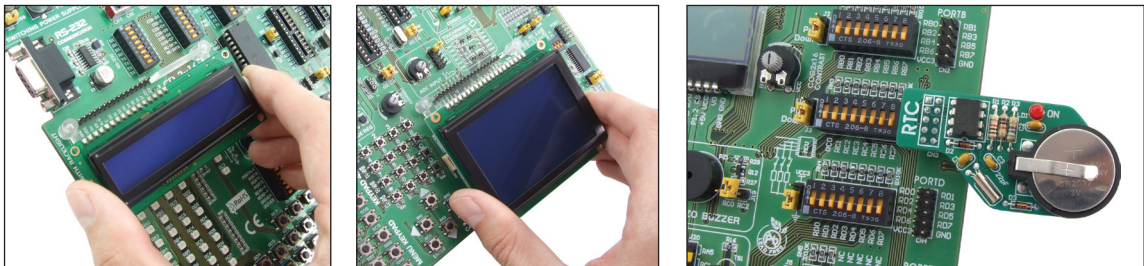


Figure 1-3: Placing additional modules on the board

2.0. Supported Microcontrollers

The *EasyLV-18F v6* development system provides three separate sockets for PIC microcontrollers in DIP40, DIP28 and DIP20 packages. These sockets allow supported microcontrollers in DIP packages to be plugged directly into the development board. Next to each socket are jumpers whose purpose is to select the function of the microcontroller pins. Table below specifies the functions of these jumpers.

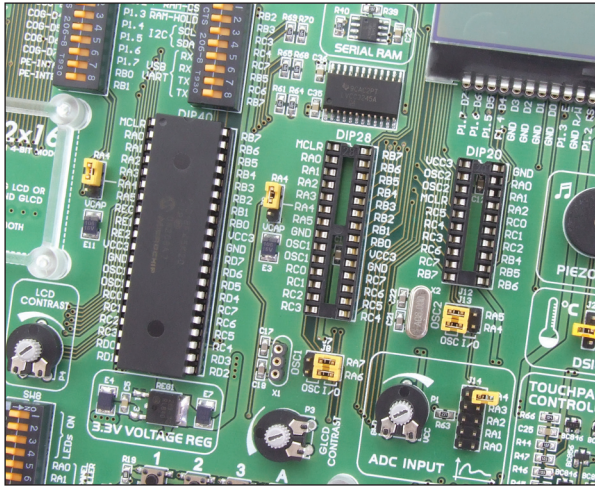


Figure 2-1: Microcontroller sockets

Jumper	Position	Fncion
J6	RA4	RA4 is configured as an I/O pin
	VCAP	Pin RA4 is connected to a filter capacitor
J9	RA4	RA4 is configured as an I/O pin
	VCAP	Pin RA4 is connected to a filter capacitor
J7	I/O	RA7 is configured as an I/O pin
	OSC	Pin RA7 is provided with clock frequency
J8	I/O	RA6 is configured as an I/O pin
	OSC	Pin RA6 is provided with clock frequency
J12	I/O	RA5 is configured as an I/O pin
	OSC	Pin RA5 is provided with clock frequency
J13	I/O	RA4 is configured as an I/O pin
	OSC	Pin RA4 is provided with clock frequency

PIC microcontrollers normally use a quartz crystal for stabilizing clock frequency. The *EasyLV-18F v6* provides two sockets for quartz crystal. Microcontrollers in DIP40 and DIP28 packages use socket X1 (OSC1) for quartz crystal. If microcontrollers in DIP20 package are used, it is necessary to move quartz crystal from socket X1 to socket X2 (OSC2). Besides, it is possible to replace the existing quartz crystal with another one. The value of the quartz crystal depends on the maximum clock frequency allowed.

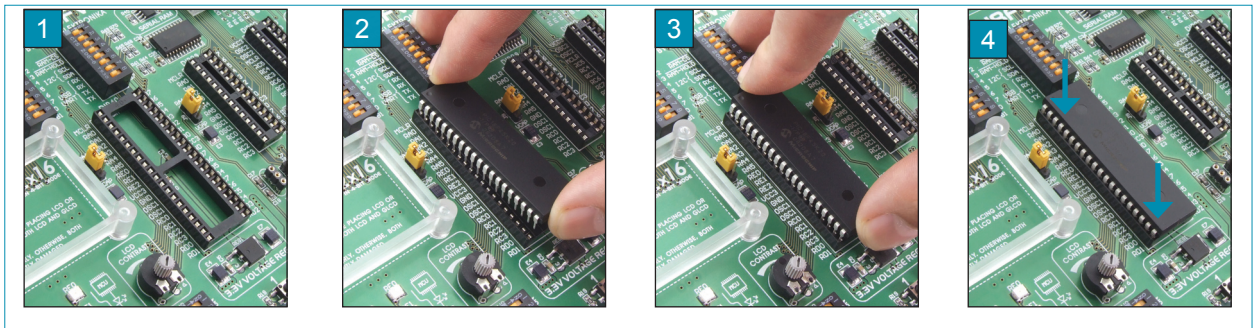


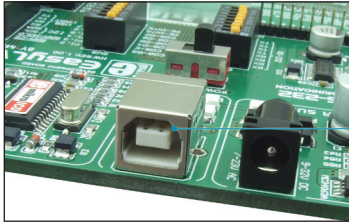
Figure 2-2: Plugging microcontroller into appropriate socket

Prior to plugging the microcontroller into the appropriate socket, make sure that the power supply is turned off. Figure 2-2 shows how to correctly plug a microcontroller. Figure 1 shows an unoccupied DIP40 socket. Place one end of the microcontroller into the socket as shown in Figure 2. Then put the microcontroller slowly down until all the pins thereof match the socket, as shown in Figure 3. Check again that everything is placed correctly and press the microcontroller easily down until it is completely plugged into the socket, as shown in Figure 4.

NOTE: Only one microcontroller may be plugged into the development board at one time.

page 3.0. On-board LV18PICprog Programmer

A programmer is a necessary tool when working with microcontrollers. It is used to load a hex code into the microcontroller and provides an interface between the microcontroller and a PC. The *EasyLV-18F v6* has an on-board *LV18PICprog* programmer. All you need is a .hex file to be loaded into the microcontroller using the *Lv18PicFLASH* program. Figure 3-3 shows connection between the compiler, the *Lv18PicFLASH* program and the microcontroller.



Programmer's USB connector

Figure 3-1: USB connector's front side

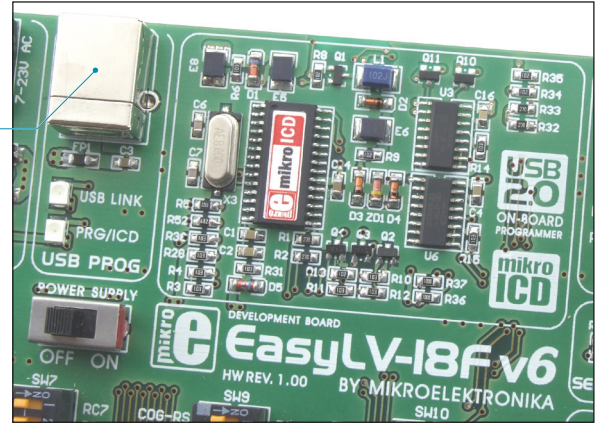


Figure 3-2: On-board programmer

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Compiling program

```

1110001001 Bin.
0110100011
0111 2FC23AA7
1011 F43E0021A
Hex. DA67F0541

```

hex code loading

- ① Write a program in one of PIC compilers and generate a .hex file;
- ② Use the *Lv18PicFLASH* program to select desired microcontroller to be programmed;
- ③ Click the *Write* button to dump the code into the microcontroller.

Write a code in one of the PIC compilers, generate a .hex file, and employ the on-board programmer to load the code into the microcontroller.

On the left side of the *Lv18PicFLASH* program's main window, there are a number of options for setting the operation of the microcontroller to be used. A number of options which enable the programming process are provided on the right side of the window. Positioned in the bottom right corner of the window, the *Progress* bar enables you to monitor the programming progress.

Figure 3-3: Programming process

4.0. mikroICD (In-Circuit Debugger)

The mikroICD (In-Circuit Debugger) is an integral part of the on-board programmer. It is used for testing and debugging programs in real time. The process of testing and debugging is performed by monitoring the state of all registers within the microcontroller while operating in real environment. The mikroICD software is integrated in all PIC compilers designed by Mikroelektronika (mikroBASIC PRO, mikroC PRO, mikroPASCAL PRO etc). As soon as the mikroICD debugger starts up, a window called *Watch Values*, appears on the screen, Figure 4-1. The *mikroICD* debugger communicates to the microcontroller through the microcontroller's pins used for programming.

mikroICD debugger options:

Start Debugger	[F9]
Run/Pause Debugger	[F6]
Stop Debugger	[Ctrl+F2]
Step Into	[F7]
Step Over	[F8]
Step Out	[Ctrl+F8]
Toggle Breakpoint	[F5]
Show/Hide Breakpoints	[Shift+F4]
Clear Breakpoints	[Ctrl+Shift+F5]

Each of these commands is activated via keyboard shortcuts or by clicking appropriate icon within the *Watch Values* window.

Figure 4-1: Watch Values window

The mikroICD debugger also offers functions such as running a program step by step (single stepping), pausing the program execution to examine the state of currently active registers using breakpoints, tracking the values of some variables etc. The following example illustrates a step-by-step program execution using the *Step Over* command.

Step 1:

In this example the 41st program line is highlighted in blue, which means that it will be executed next. The current state of all registers within the microcontroller can be viewed in the mikroICD *Watch Values* window.

Name	Value
PORTB	35
PORTC	53
PORTD	0b0100 010

Step 2:

After the *Step Over* command is executed, the microcontroller will execute the 41st program line. The next line to be executed is highlighted in blue. The state of registers being changed by executing this instruction may be viewed in the *Watch Values* window.

Name	Value
PORTB	0
PORTC	53
PORTD	0b0100 010

Name	Value	Address
PORTB	25	0x0006
PORTC	33	0x0007
PORTD	0	0x0008
ANSEL	0	0x0188
ANSELH	0	0x0189

NOTE: For more information on the mikroICD debugger refer to the *mikroICD Debugger* manual.

5.0. ICD Connector

In addition to the on-board *LV18PICprog* programmer, one of the external programmers ICD1 or ICD3 from Microchip can be used for programming microcontrollers. The external programmer is connected to the microcontroller through the ICD connector. The function of jumpers J18 and J17 is to determine which microcontroller pins are to be fed with the programming signal.

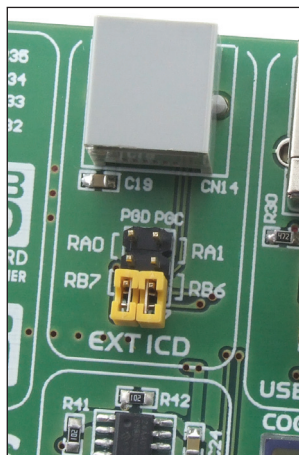


Figure 5-1: ICD connector

External programmer is connected to the microcontroller pins RA0 and RA1

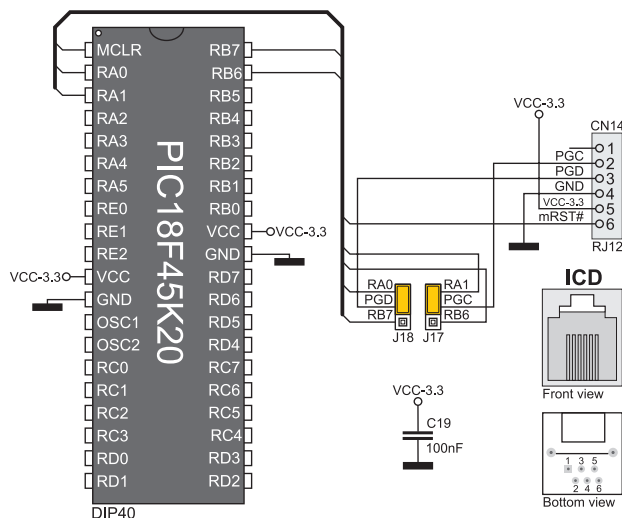


Figure 5-2: ICD connector and microcontroller connection schematic

You need the appropriate program to enable .hex file to be loaded into the microcontroller. Use the MPLAB program from Microchip to write a code to be loaded into the microcontroller. The MPLAB program includes the *Programmer* option which enables you to select the programmer to be used for programming.

6.0. 3.3V Voltage Regulator

The *EasyLV-18F v6* development system is provided with a voltage regulator whose purpose is to reduce the power supply voltage from 5V to 3.3V. The 3.3V power supply voltage is then used for powering the microcontroller and other on-board modules. The MC33269DT-3.3 circuit provided on the development system is used as a voltage regulator capable of providing 800mA current.

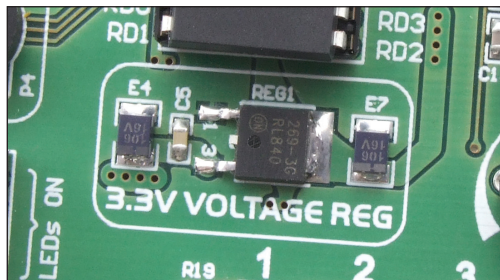


Figure 6-1: Voltage regulator

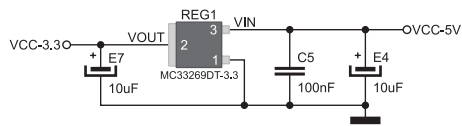


Figure 6-2: Voltage regulator connection schematic

7.0. Power supply

The EasyLV-18F v6 development system may use one of two power supply sources:

1. +5V PC power supply through the USB programming cable; and
2. External power supply source connected to an AC/DC connector provided on the development board.

The MC34063A voltage regulator and Gretz rectifier are used to enable external power supply voltage to be either AC (in the range of 7V to 23V) or DC (in the range of 9V to 32V). Jumper J10 is used as a selector for a power supply source. To make advantage of the USB power supply, jumper J10 should be placed in the USB position. When using the external power supply, jumper J10 should be placed in the EXT position. The development system is turned on/off by switching the position of the POWER SUPPLY switch.

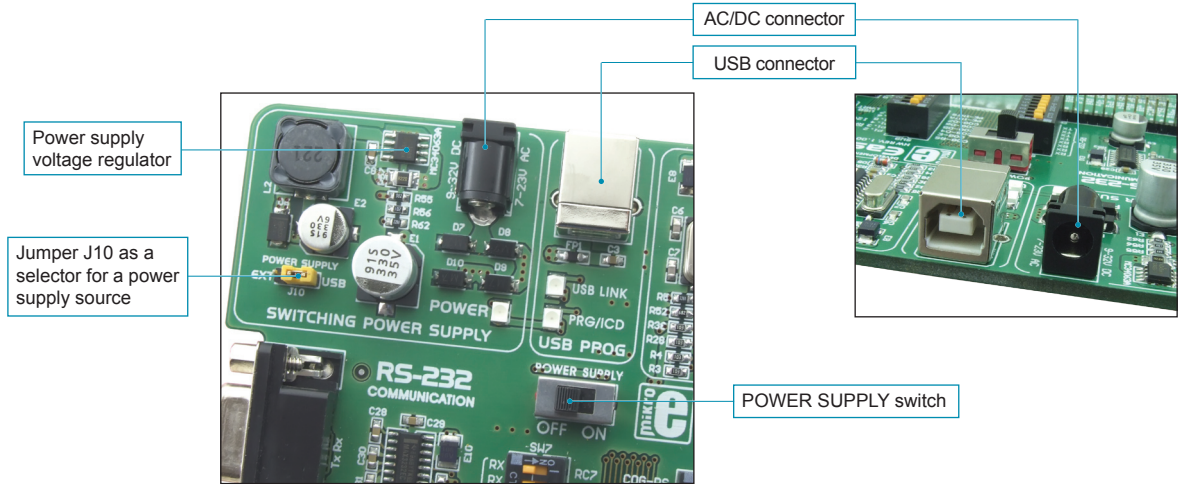


Figure 7-1: Power supply

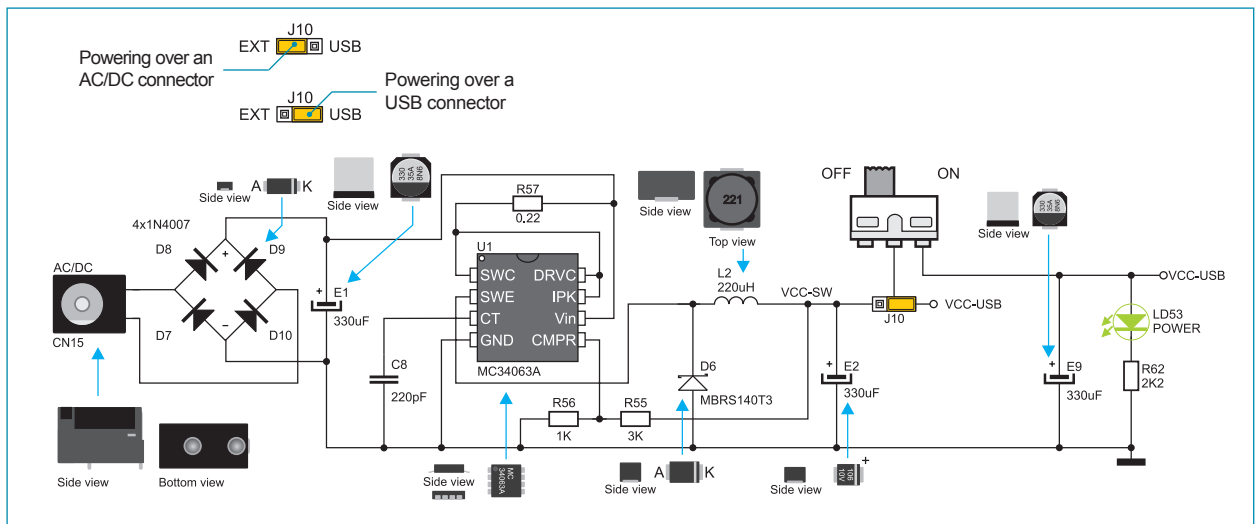


Figure 7-2: Power supply source schematic

page 8.0. USB Communication Module

The *EasyLV-18F v6* development system can communicate with external devices through USB connectors CN11 and CN18. Only microcontrollers with the built-in USB communication module are capable of using USB communication. Which of these two connectors is to be used for USB communication depends on the microcontroller in use. The CN11 USB connector along with jumpers J11 are used for the PIC18F2xJ50 microcontroller family. The CN18 USB connector with jumpers J21 is used for the PIC18F1xK50 and PIC18LF1xK50 microcontroller families. Jumpers J11 and J21 are used to determine the function of microcontroller pins: RC3, RC4 and RC5 are connected to CN11, whereas RA0, RA1 and RA2 are connected to CN18. When jumpers are in the USB position (Figure 8-1), microcontroller pins are used for USB communication. When they are in the I/O position (Figures 8-2 and 8-3), microcontroller pins are used as I/O pins.

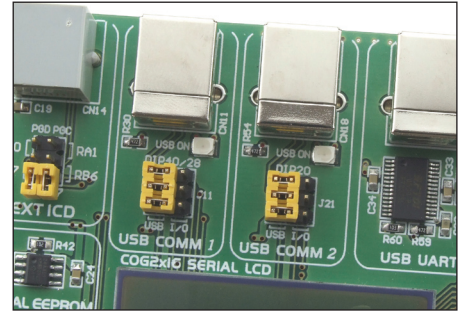


Figure 8-1: USB connector

Pins RC3, RC4 and RC5 are configured as I/O pins

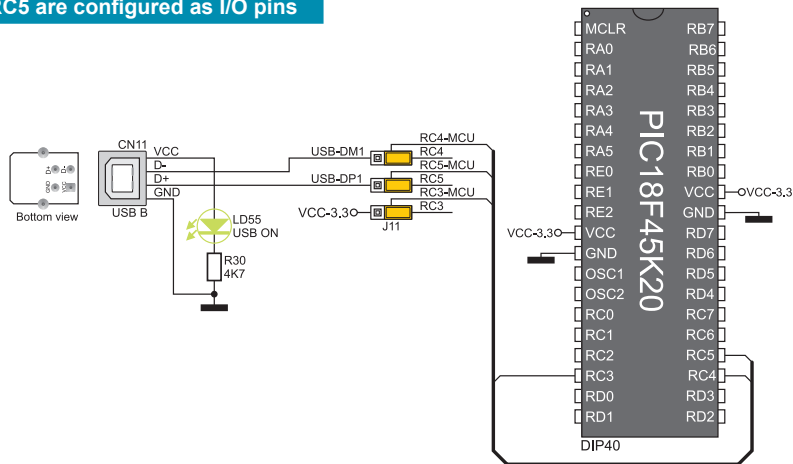


Figure 8-2: CN11 USB connector and microcontroller schematic

Pins RA0, RA1 and RA2 are configured as I/O pins

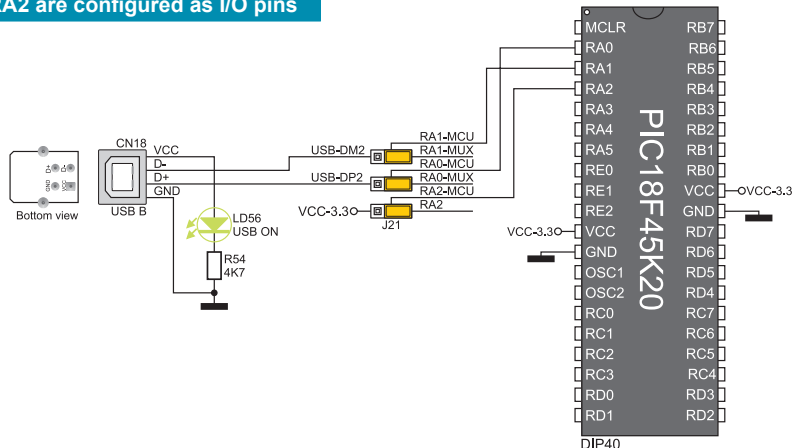


Figure 8-3: CN18 USB connector and microcontroller schematic

9.0. A/D Converter Test Inputs

An A/D converter is used for converting an analog voltage into the appropriate digital value. The A/D converter is linear, which means that converted number is linearly dependent on the input voltage value. The A/D converter built into the microcontroller converts an analog voltage value into a digital number. Potentiometer P1 enables voltage to vary between 0 and 3.3V. The microcontroller with a built-in A/D converter is supplied with this voltage via test inputs. Jumpers J14 are used for selecting one of the following microcontroller pins (RA0 - RA4) to be supplied with A/D conversion voltage. Resistor R63 has a protective function and is used to limit current flow through the potentiometer, that is to say the microcontroller pin.

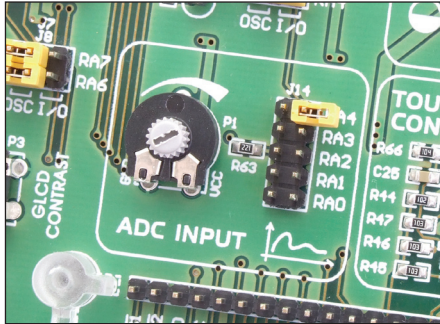


Figure 9-1: ADC (jumper's default position)

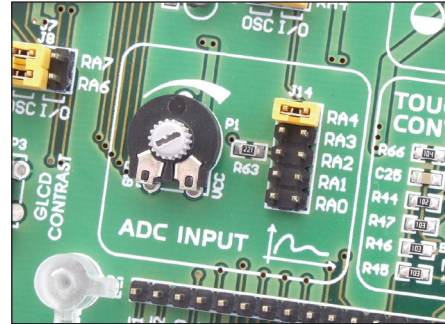


Figure 9-2: Pin RB4 as input pin for A/D conversion

A/D conversion is performed via the RA4 microcontroller pin

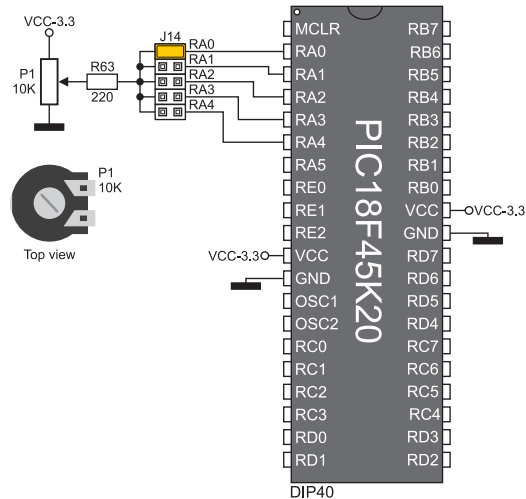


Figure 9-3: Microcontroller and A/D converter test inputs connection schematic

NOTE: In order to enable the microcontroller to accurately perform A/D conversion, it is necessary to turn off LEDs and pull-up/pull-down resistors on the port pins used by the A/D converter.

10.0. USB UART Module

The USB UART module enables the microcontroller provided on the development system to be connected to external USB devices. The USB UART module features the FT232RL circuit providing an interface between a USB device and microcontroller's UART module. In order to establish connection between the microcontroller and USB UART module, it is necessary to set switches 5 (RC7) and 7 (RC6) on the DIP switch SW10 to ON position. Switches 6 (RB5) and 8 (RB7) on the DIP switch SW10 may be optionally used. It depends on the microcontroller pins used for serial communication which of these switches are to be used.

When connecting a UART device to the microcontroller, it is necessary to cross-connect send and receive data lines. In other words, the device's RX pin should be connected to the microcontroller's TX pin, whereas the device's TX pin should be connected to the microcontroller's RX pin. It is not possible to directly connect the UART device's send data line TX to the microcontroller's send data line TX. If you did that, the microcontroller wouldn't be able to receive nor to send data.

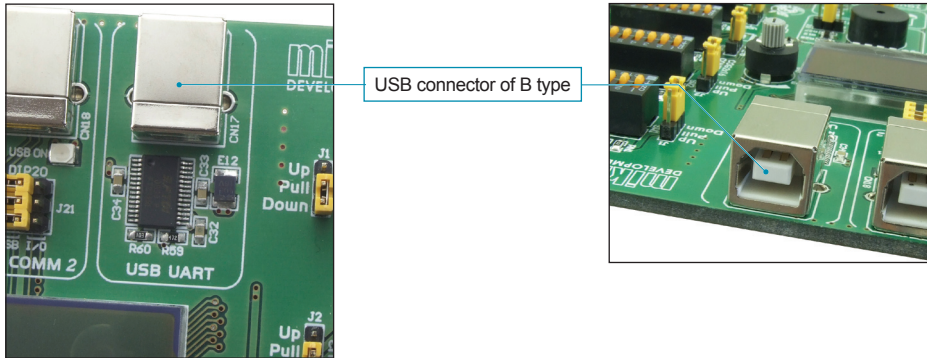


Figure 10-1: USB UART module

USB connector of B type connected to the microcontroller

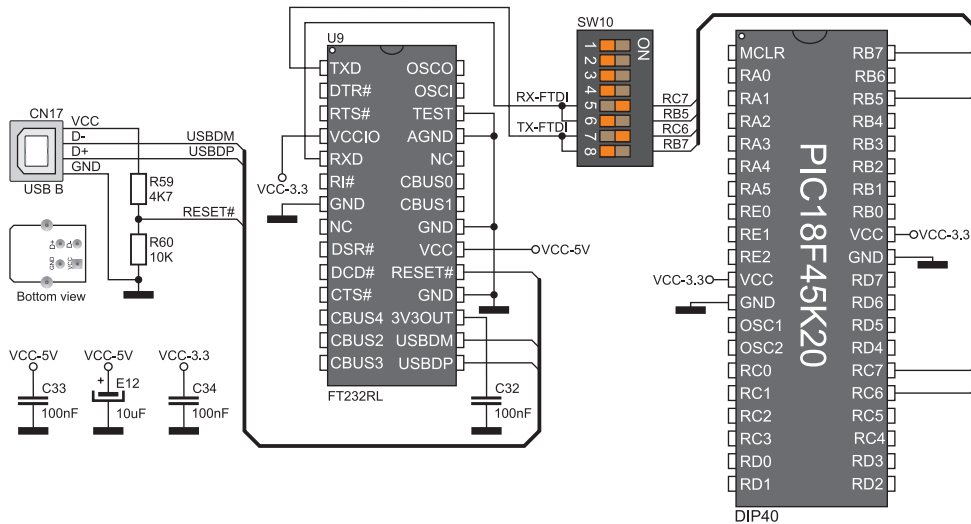


Figure 10-2: USB connector and microcontroller connection schematic

12.0. Serial EEPROM Module

EEPROM (Electrically Erasable Programmable Read-Only Memory) is a built-in memory module used to store data that should be saved when the power supply goes off. The 24AA01 circuit can store up to 1Kbit data and communicates with the microcontroller through pins RC3 and RC4. In order to enable connection between the EEPROM module and the microcontroller, it is necessary to set switches 3 and 4 on the DIP switch SW10 to ON position.

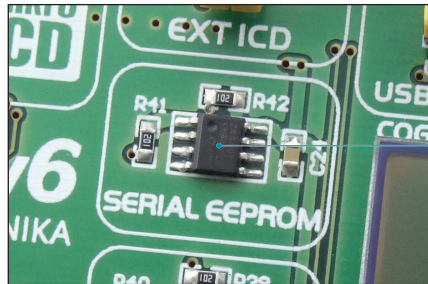


Figure 12-1: EEPROM module

Serial EEPROM is connected to the microcontroller via pins RC3 and RC4

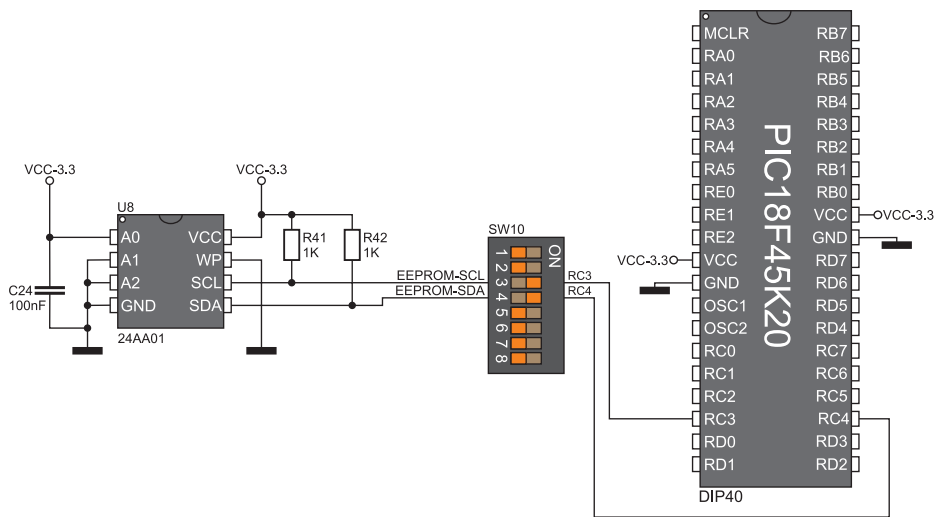


Figure 12-2: Serial EEPROM module and microcontroller connection schematic

13.0. RAM Module

Owing to the 23K640 circuit, the development system is provided with 64Kbit RAM module that can be connected to the microcontroller via SPI interface. The function of the serial RAM module is to expand available RAM of the microcontroller. To establish connection between the microcontroller and RAM module, it is necessary to set switches 3, 4 and 5 on the DIP switch SW6 to ON position. The function of the RAM-HOLD# (RB3) on the DIP switch SW10 is to hold communication between the microcontroller and RAM module. The RAM-CS (RB2) switch on the DIP switch SW10 is used to control the access to RAM.

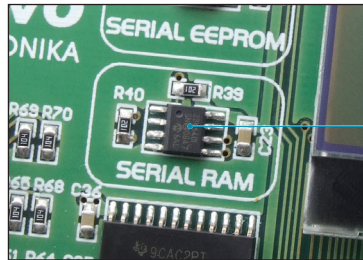


Figure 13-1: Serial RAM module

Serial RAM is connected to the microcontroller via RB2, RB3, RC3, RC4 and RC5 pins

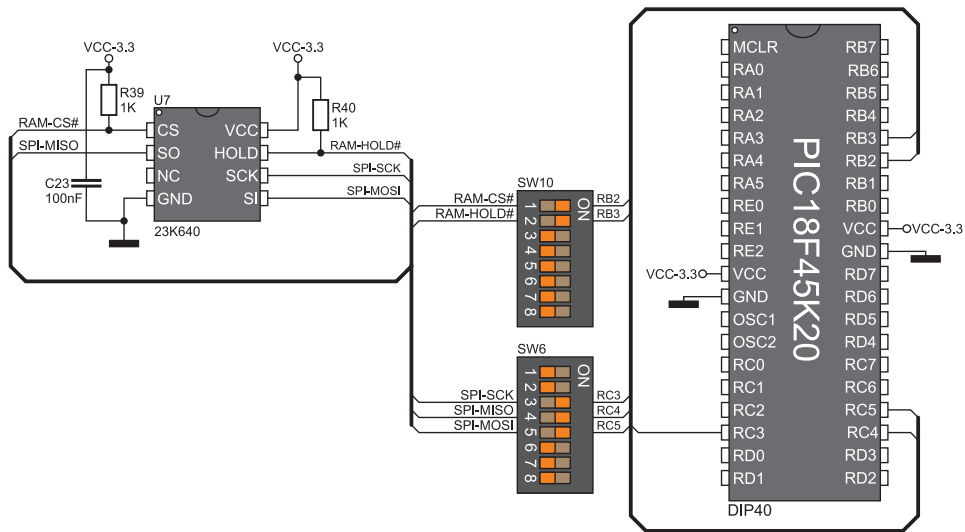


Figure 13-2: Serial RAM module and microcontroller connection schematic

page 14.0. DS1820 Temperature Sensor

1-wire® serial communication enables data to be transferred over one single communication line while the process itself is under control of the master device. The advantage of such communication is that only one microcontroller pin is used. All slave devices have by default a unique ID code, which enables the master device to easily identify all devices sharing the same communication interface.

The DS1820 is a temperature sensor that uses 1-wire communication. It is capable of measuring temperatures within the range of -55 to 125°C and provides $\pm 0.5^{\circ}\text{C}$ accuracy for temperatures within the range of -10 to 85°C . A power supply voltage of 3.3V is required for its operation. It takes maximum 750ms for the DS1820 to calculate temperature with a 9-bit resolution. The *EasyLV-18F v6* development system provides a separate socket for the DS1820. It may use either RA5 or RC0 pin for communication with the microcontroller. Pins to be used for 1-wire communication are selected by jumper J20. In this example, the RA5 pin is used for 1-wire communication with the microcontroller, as shown in Figure 14-4.

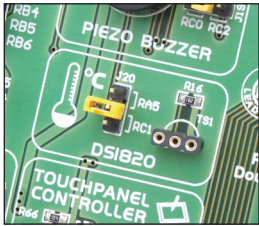


Figure 14-1: DS1820 connector (DS1820 is not used)

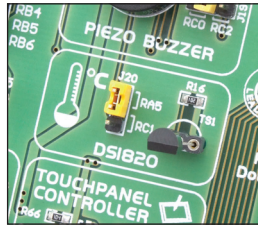


Figure 14-2: J20 in RA5 position (DS1820 is connected to the RA5 pin)

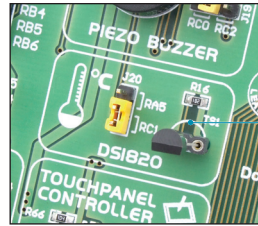


Figure 14-3: J20 in RC1 position (DS1820 is connected to the RC1 pin)

NOTE:
Make sure that half-circle on the board matches the round side of the DS1820

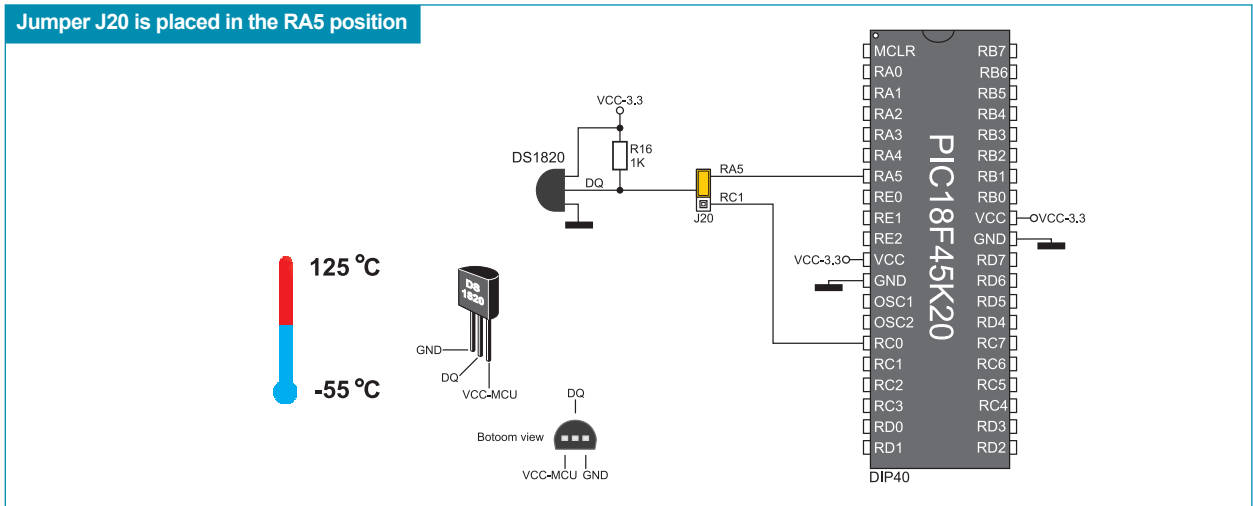


Figure 14-4: DS1820 temperature sensor and microcontroller connection schematic

15.0. Piezo Buzzer

Due to a built-in piezo buzzer, the development system is capable of emitting audio signals. For the piezo buzzer to operate normally it is necessary to generate a voltage signal of specified frequency. The voltage signal is generated in the microcontroller by the appropriate code written to its memory. Remember, when writing the voltage signal generation code, the piezo buzzer's resonant frequency is 3.8kHz. In addition, other frequencies in the range between 20Hz and 20kHz can be used, but the best performance is achieved by frequencies ranging between 2kHz and 4kHz. The voltage signal can be generated via microcontroller pin RC0 or RC2. It depends on jumper J19 which of these two pins is to be used.

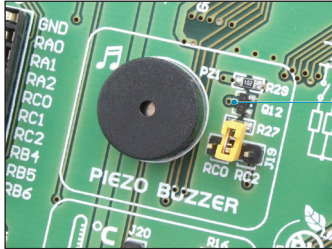


Figure 15-1: Piezo buzzer

Transistor Q12 is used to amplify voltage signal generated by the microcontroller

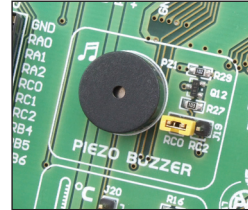


Figure 15-2: Jumper J19 is in RC0 position (voltage signal is generated via the RC0 pin)

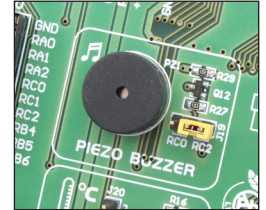


Figure 15-3: Jumper J19 is in RC2 position (voltage signal is generated via the RC2 pin)

Piezo buzzer is connected to the microcontroller via pin RC0

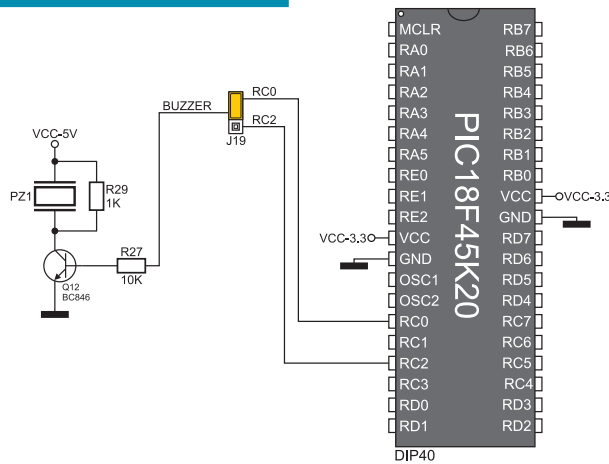


Figure 15-4: Piezo buzzer and microcontroller connection schematic

16.0. LEDs

LED (Light-Emitting Diode) is a highly efficient electronic light source. When connecting LEDs, it is necessary to use a current limiting resistor. A common LED voltage is approximately 2.5V, while the current varies from 1 to 20mA depending on the type of LED. The *EasyLV-18F v6* uses LEDs with current $I=1\text{mA}$.

There are 35 LEDs on the *EasyLV-18F v6* development system which visually indicate the state of each microcontroller I/O pin. An active LED indicates that a logic one (1) is present on the pin. In order to enable the pin state to be shown, it is necessary to select appropriate port (PORTA, PORTB, PORTC, PORTD or PORTE) using DIP switch SW8.

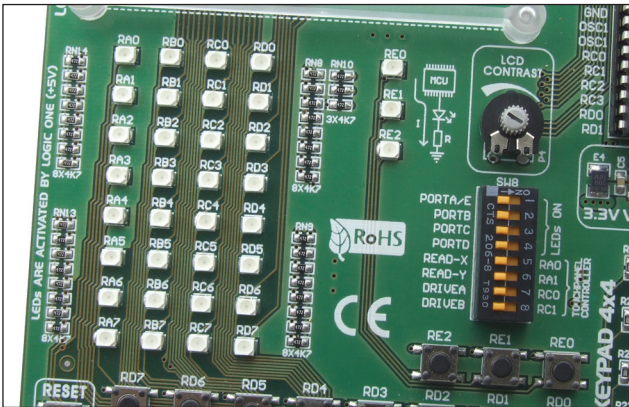
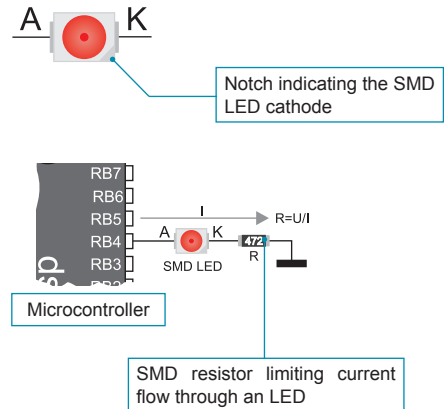


Figure 16-1: LEDs



Port PORTB LEDs are turned on

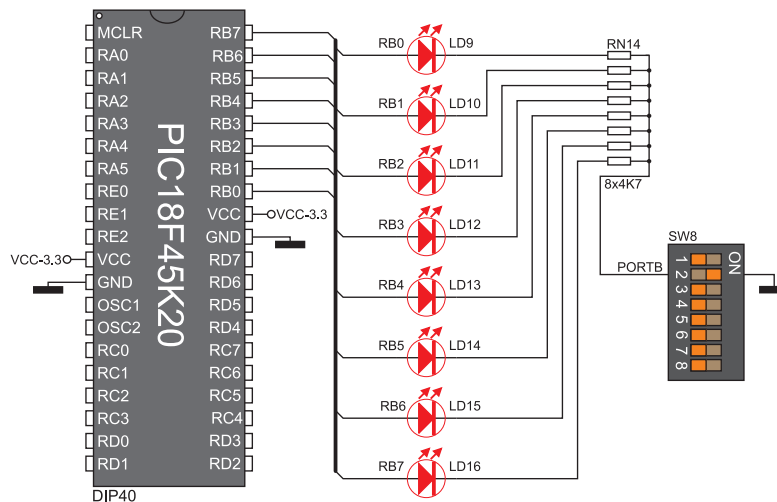


Figure 16-2: LEDs and port PORTB connection schematic

17.0. 4x4 Keypad

The 4x4 keypad is a standard alphanumeric keypad connected to the microcontroller port PORTD. The performance of this keypad is based on the 'scan and sense' principle where pins RD0, RD1, RD2 and RD3 are configured as inputs connected to pull-down resistors. Pins RD4, RD5, RD6 and RD7 are configured as outputs and generate a logic one (1). Pressing any button will cause a logic one (1) to be applied to one of input pins. It is determined from within the software which push button is pressed. For example, by pressing button 6, a logic one (1) will appear on the RD2 pin. The microcontroller detects a button press first, then by applying logic ones to pins RD4, RD5, RD6 and RD7 it is possible to determine which push button is pressed.

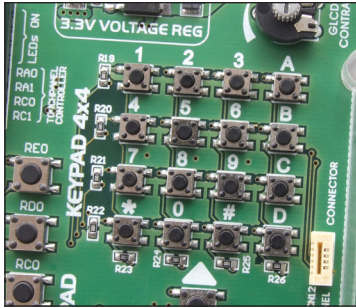


Figure 17-1: Keypad 4x4

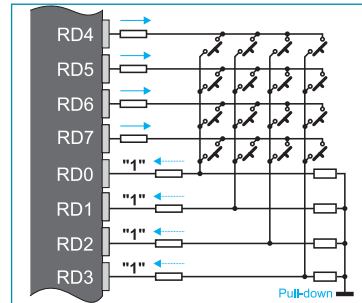


Figure 17-2: Keypad 4x4 performance

Keypad 4x4 is connected to the microcontroller via port PORTD

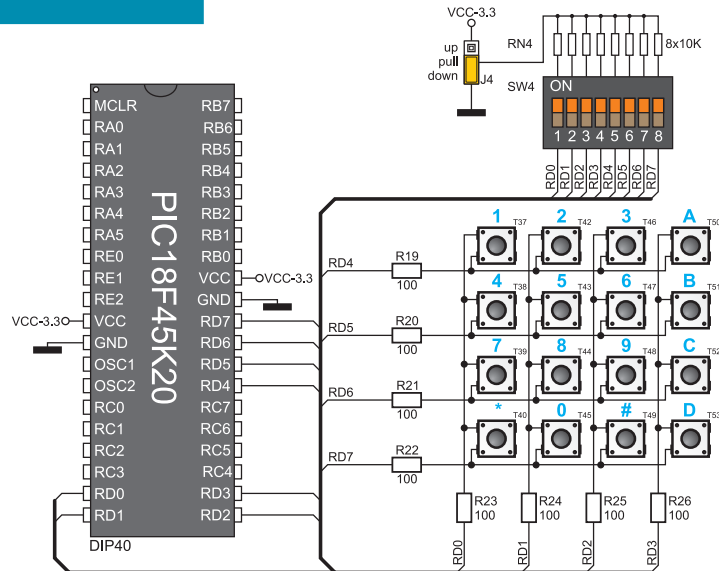


Figure 17-3: Keypad 4x4 and microcontroller connection schematic

18.0. MENU Keypad

There is a set of push buttons serving as a navigation keypad called MENU provided on the *EasyLV-18F v6* development system. It primarily consists of four push-buttons marked as left, right, up and down arrows. There are also two additional push buttons marked as ENTER and CANCEL. MENU push buttons are connected in the same way as the push buttons of ports PORTA and PORTC. Their function is determined by the user when writing the program for the microcontroller.

Have in mind when writing program for the microcontroller that MENU keypad is connected to ports PORTA and PORTC

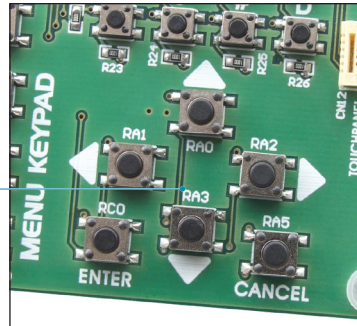


Figure 18-1: MENU keypad

MENU keypad's push buttons are connected the same way as those of ports PORTA and PORTC

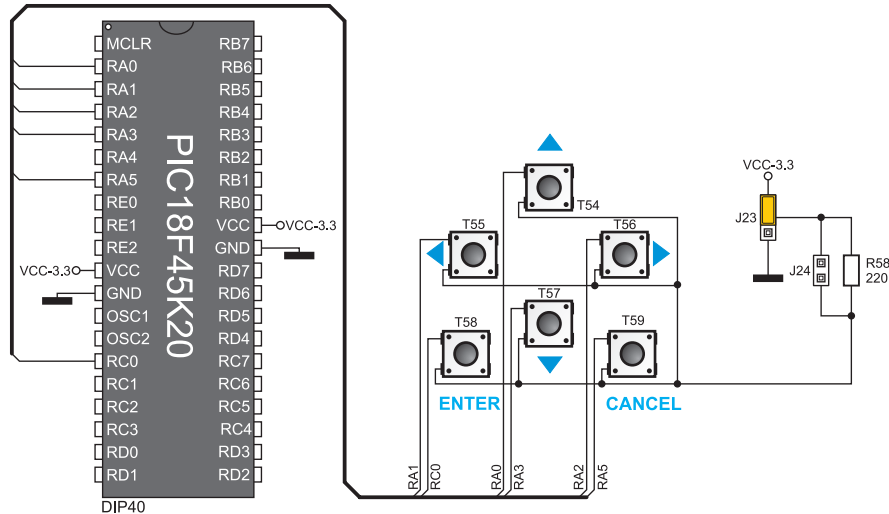


Figure 18-2: MENU keypad and microcontroller connection schematic

19.0. Push Buttons

The logic state of all microcontroller input pins may be changed by means of push buttons. Jumper J23 is used to determine the logic state to be applied to the desired microcontroller pin by pressing the appropriate push button. The function of the protective resistor is to limit the maximum current, thus preventing the development system and peripheral modules from being damaged in case a short circuit occurs. If needed, advanced users may shorten such resistor using jumper J24. Right next to the push buttons, there is a RESET button which is used to provide the MCLR pin with the microcontroller reset signal over the on-board programmer.

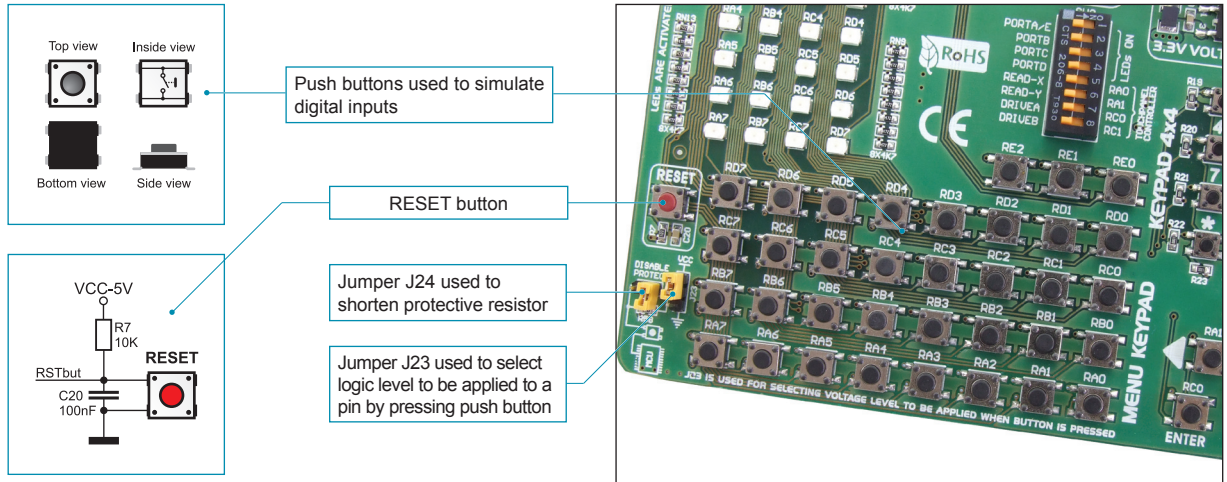


Figure 19-1: Push buttons

By pressing any push button when jumper J23 is in the VCC position, a logic one (3.3V) will be applied to the appropriate microcontroller pin, as shown in Figure 19-2.

By pressing a push button, the appropriate pin will be driven high (1)

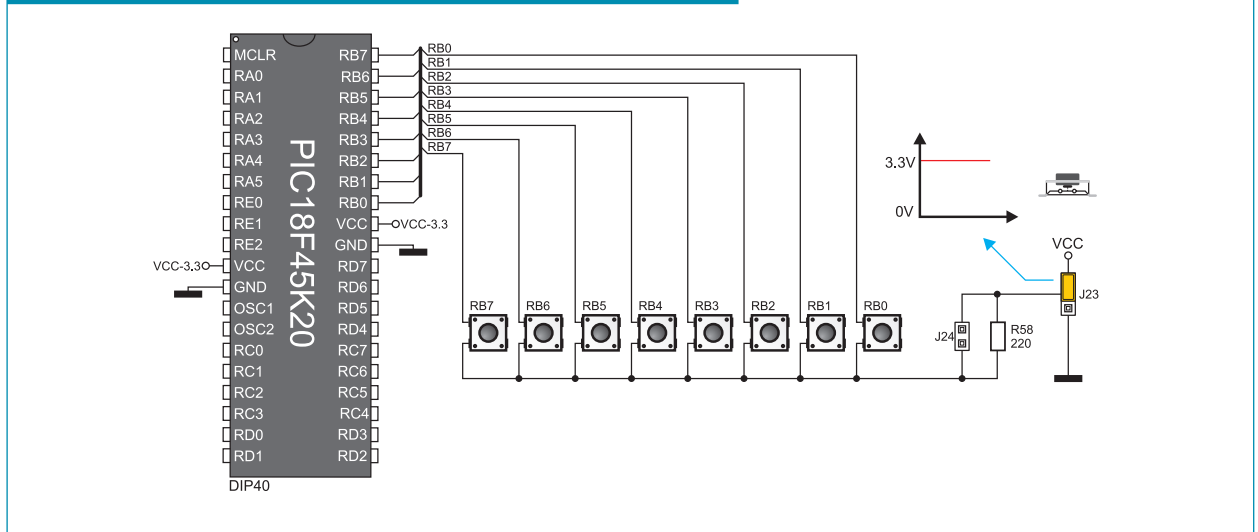
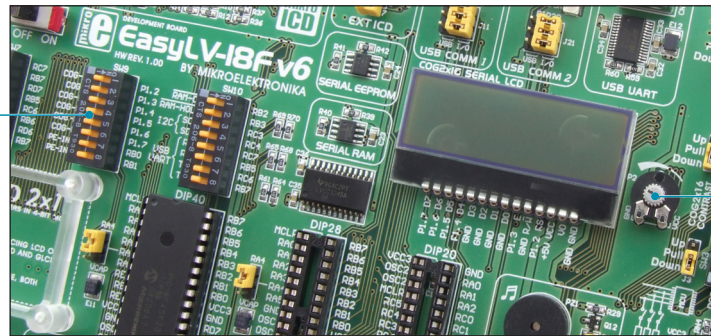


Figure 19-2: Push buttons and port PORTB connection schematic

20.0. On-Board 2x16 LCD

On-board 2x16 LCD display is connected to the microcontroller by means of a port expander. In order to use this display, it is necessary to set switches 1-6 on the DIP switch SW9 on ON position, thus connecting the on-board LCD to the port expander's PORT1. DIP switch SW6 enables SPI communication between the port expander and the microcontroller. Potentiometer P2 is used to adjust the on-board display contrast.

Unlike 2x16 LCD, the on-board LCD has no backlight, but similar to 2x16 LCD, this display also displays digits in two lines each containing up to 16 characters of 7x5 pixels.



DIP switch SW9 used to turn on on-board 2x16 LCD

Contrast adjustment potentiometer P2

Figure 20-1: On-board 2x16 LCD

On-board programmer is connected to the microcontroller via port expander

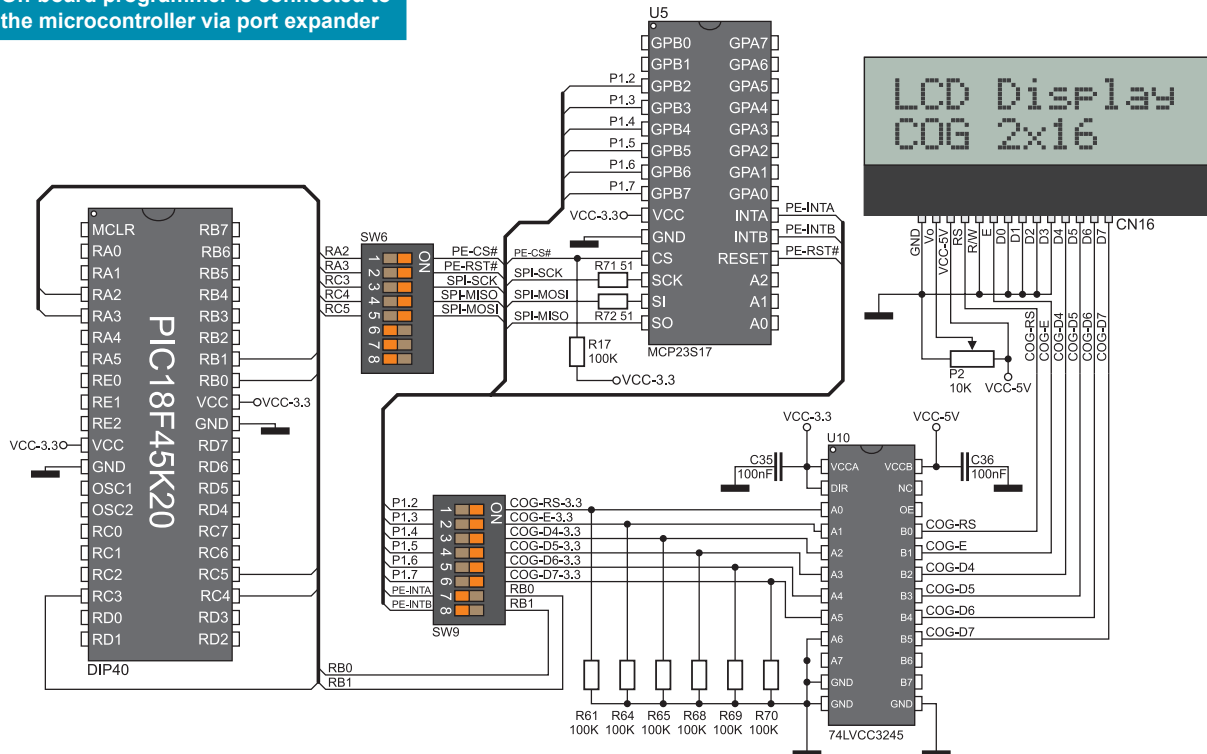


Figure 20-2: On-board 2x16 LCD and microcontroller connection schematic

21.0. 2x16 LCD

The *EasyLV-18F v6* development system provides an on-board connector for the alphanumeric 2x16 LCD. This connector is linked to the microcontroller via port PORTB. Potentiometer P4 is used to adjust display contrast. Switch 8 (LCD-GLCD BACKLIGHT) on the DIP switch SW6 is used to turn the display backlight on/off.

Communication between this LCD and the microcontroller is performed in a 4-bit mode. Alphanumeric digits are displayed in two lines each containing up to 16 characters of 7x5 pixels.



Figure 21-1: Alphanumeric LCD connector



Figure 21-2: Alphanumeric 2x16 LCD

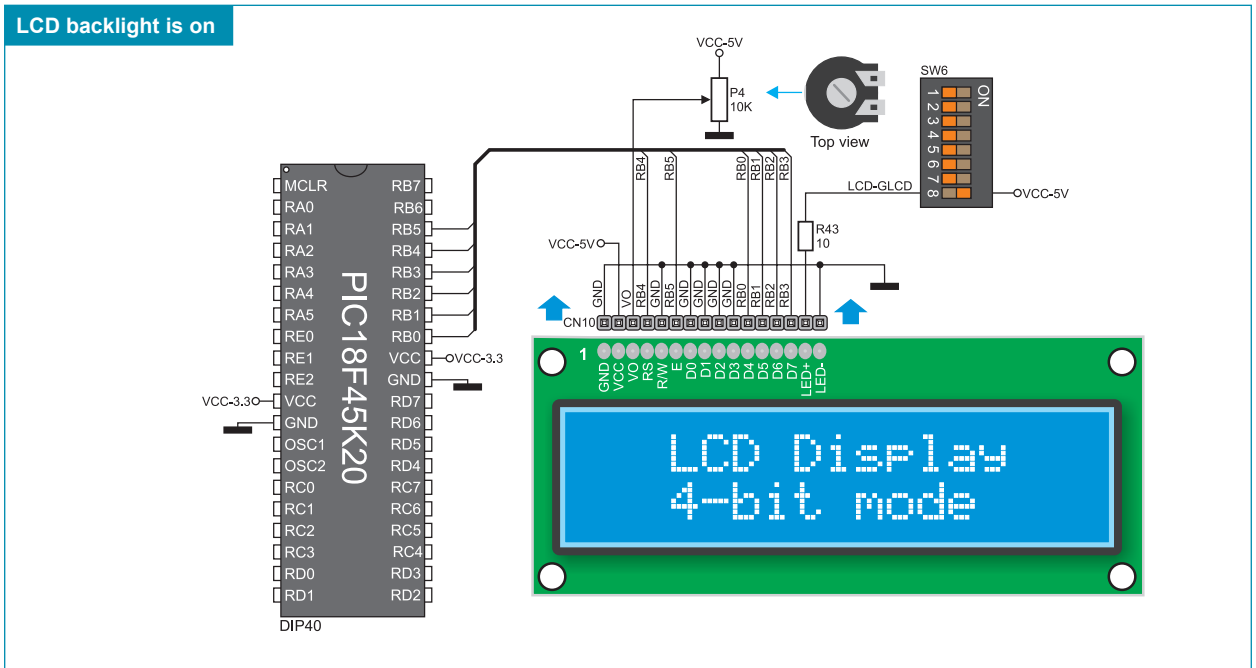


Figure 21-3: Alphanumeric 2x16 LCD and microcontroller connection schematic

23.0. Touch Panel

The touch panel is a thin, self-adhesive, transparent, touch-sensitive panel. It is placed over a GLCD. Its main function is to register pressure at some specific display point and to forward its coordinates in the form of analog voltage to the microcontroller. Switches 5, 6, 7 and 8 on the DIP switch SW8 are used for connecting the microcontroller and touch panel.

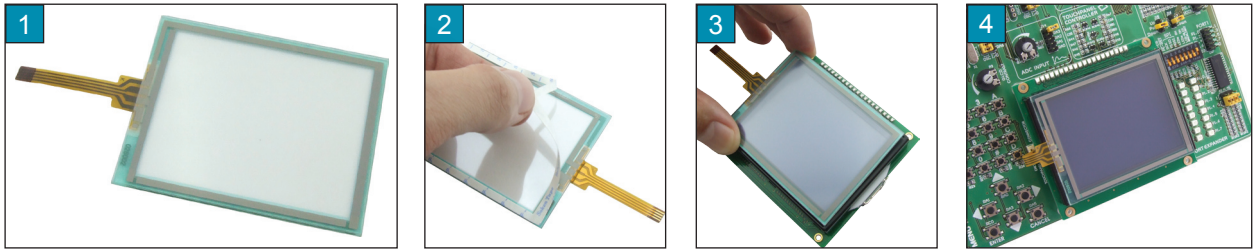


Figure 23-1: Placing touch panel over a GLCD

Figure 23-1 shows how to place a touch panel over a GLCD display. Make sure that the flat cable is to the left of the GLCD, as shown in Figure 4.

Touch panel is connected to the microcontroller via pins RA0, RA1, RC0 and RC1

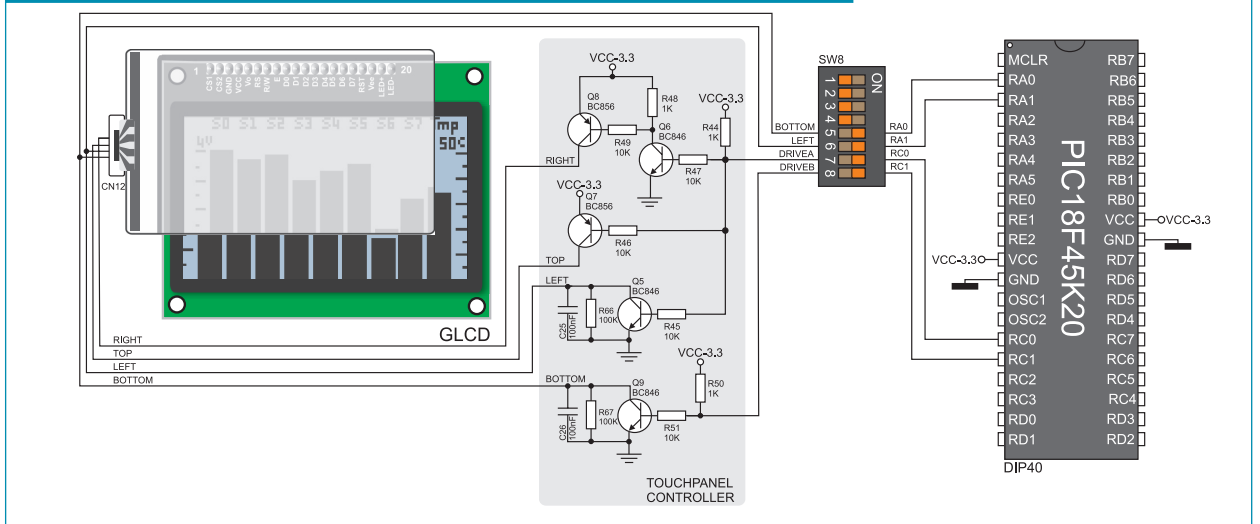


Figure 23-2: Touch panel and microcontroller connection schematic

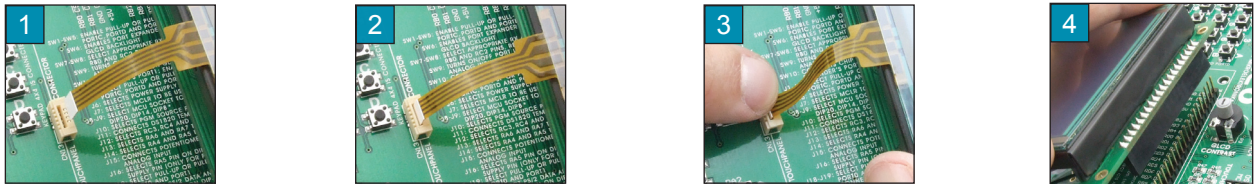


Figure 23-3: Connecting touch panel

Figure 23-3 shows in detail how to connect a touch panel to the microcontroller. Bring the end of the flat cable close to the CN12 connector (Figure 1). Plug the cable into the connector (Figure 2) and press it easily so as to fully fit the connector (Figure 3). Now, a GLCD can be plugged into the appropriate connector (Figure 4).

NOTE: LEDs and pull-up/pull-down resistors on ports PORTA and PORTC must be off when using a touch panel.

24.0. Input/Output Ports

Along the right side of the development system, there are five 10-pin connectors connected to the microcontroller's I/O ports. Pins used for programming are not directly connected to the appropriate 10-pin connectors, but via the programmer's multiplexer. Owing to DIP switch SW1-SW5, each connector pin can be connected to one pull-up/pull-down resistor. Whether pins of some port are to be connected to a pull-up or a pull-down resistor depends on the position of jumpers J1-J5.

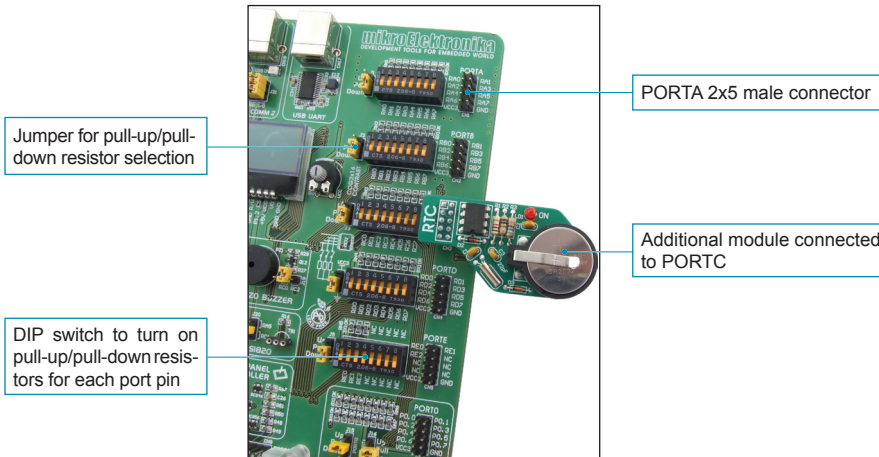


Figure 24-1: I/O ports

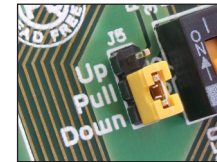


Figure 24-2: J5 in pull-down position

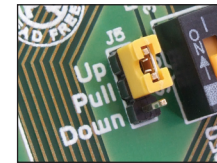


Figure 24-3: J5 in pull-up position

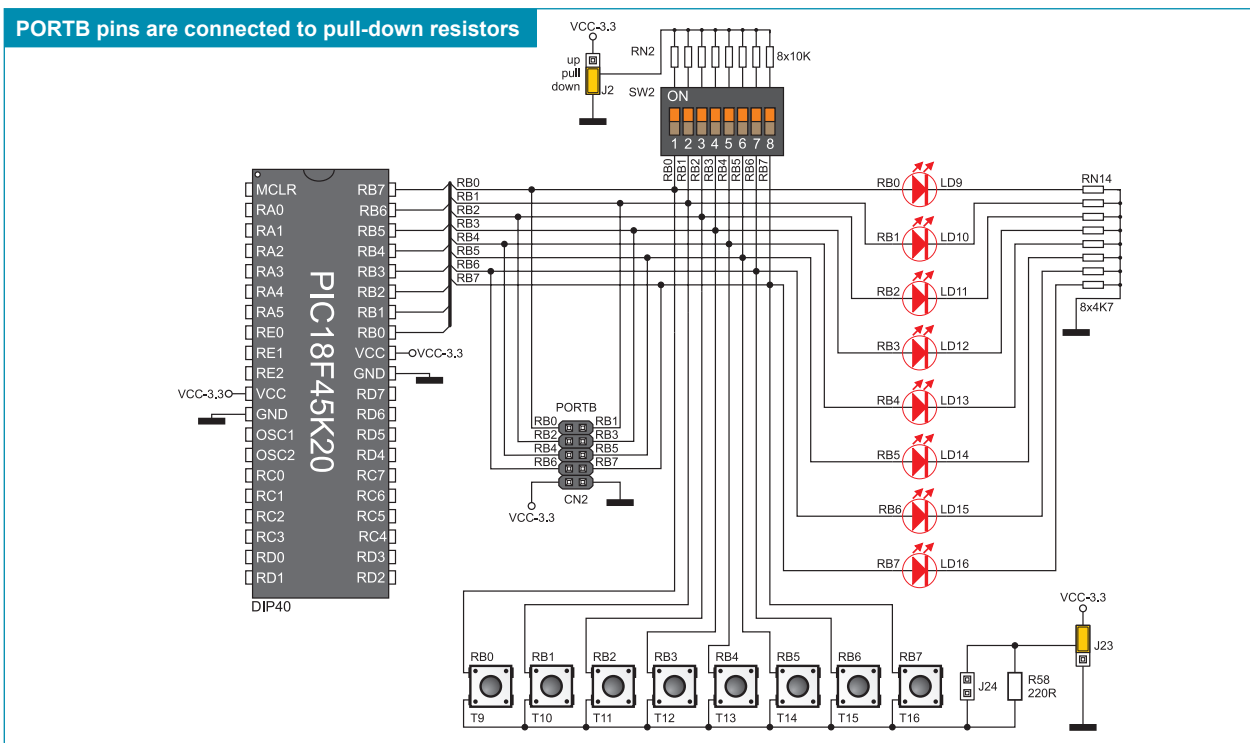


Figure 24-4: Port PORTB connection schematic

Pull-up/pull-down resistors enable you to set the logic level on all microcontroller's input pins when they are in idle state. This level depends on the position of the pull-up/pull-down jumpers. The RB0 pin with the relevant DIP switch SW2, jumper J2 and push button RB0 with jumper J23 are used here for the purpose of explaining the performance of pull-up/pull-down resistors. The principle of their operation is the same as for all other microcontroller pins.

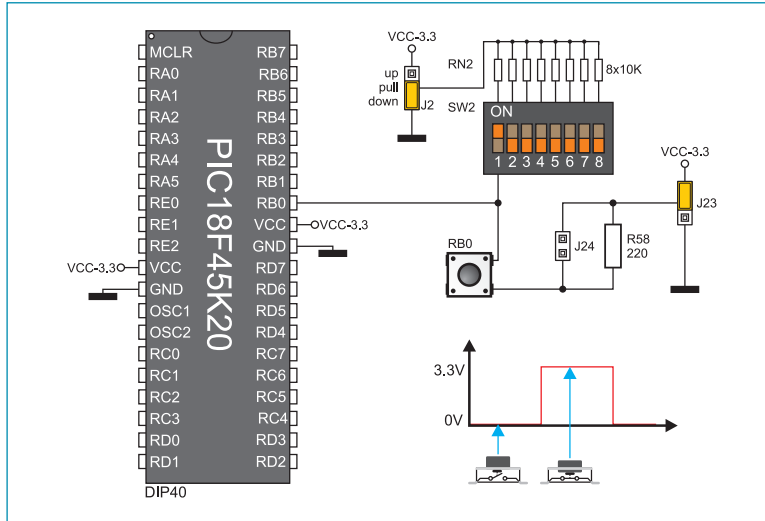


Figure 24-5: Jumper J2 in pull-down and jumper J23 in pull-up position

In order to enable the PORTB port pins to be connected to pull-down resistors, it is necessary to place jumper J2 in the *Down* position first. This enables any PORTB port pin to be supplied with a logic zero (0V) in idle state over jumper J2 and 8x10k resistor network. To provide the RB0 pin with such signal, it is necessary to set switch 1 on the DIP switch SW2 to ON position.

As a result, every time you press the RB0 button, a logic one (VCC-3.3) will appear on the RB0 pin, provided that jumper J23 is placed in the VCC position.

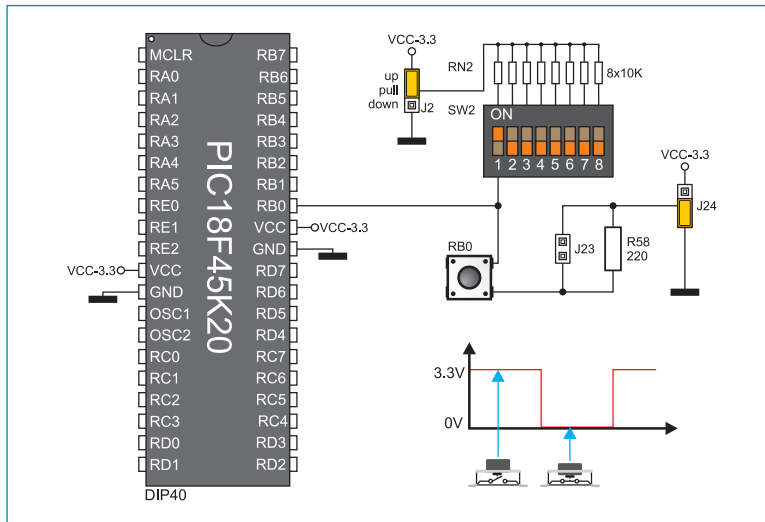


Figure 24-6: Jumper J2 in pull-up and jumper J23 in pull-down position

In order to enable port PORTB pins to be connected to pull-up resistors and the port input pins to be supplied with a logic zero (0), it is necessary to place jumper J2 in the *Up* position and jumper J23 in the GND position. This enables any port PORTB input pin to be driven high (3.3V) in idle state over the 10k resistor.

As a result, every time you press the RB0 push button, a logic zero (0V) will appear on the RB0 pin, provided that switch 1 on the DIP switch SW2 is set to ON position.

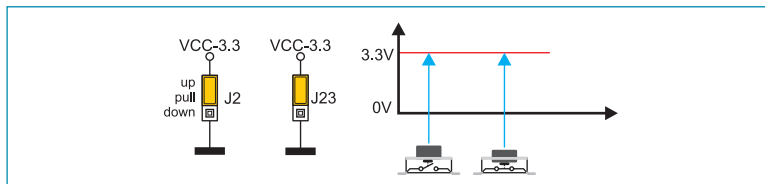


Figure 24-7: Jumpers J2 and J23 in the same position

In case that jumpers J2 and J23 have the same logic state, pressure on any button will not cause input pins to change their logic state.

25.0. Port Expander (Additional Input/Output Ports)

SPI communication lines and MCP23S17 circuit provide the *EasyLV-18F v6* development system with means of increasing the number of available I/O ports by two. If the port expander communicates to the microcontroller over DIP switches SW6 and SW9, then the microcontroller pins RA2, RA3, RC3, RC4 and RC5 used for the operation of port expander, cannot be used as I/O pins.

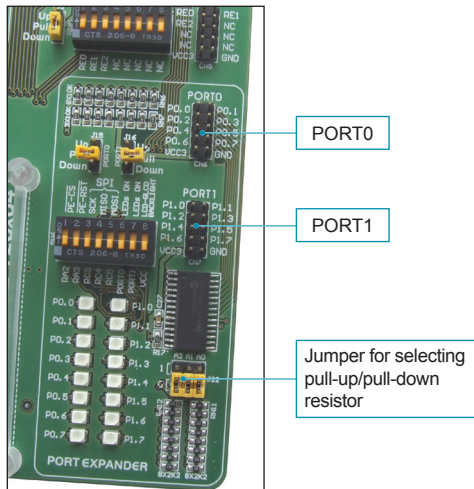


Figure 25-1: Port expander

The microcontroller communicates with the port expander (MCP23S17 circuit) using serial communication (SPI). The advantage of such communication is that only five lines are used for simultaneous data transmit and receive:

- MOSI - Master Output, Slave Input (microcontroller output, MCP23S17 input)
- MISO - Master Input, Slave Output (microcontroller input, MCP23S17 output)
- SCK - Serial Clock (microcontroller clock signal)
- CS - Chip Select (enables data transfer)
- RST - Reset

Data transfer is performed in both directions simultaneously by means of MOSI and MISO lines. The MOSI line is used for transferring data from the microcontroller to the port expander, whereas the MISO line transfers data from the port expander to the microcontroller. The microcontroller initializes data transfer by sending a clock signal when the CS pin is driven low (0V). The principle of operation of the port expander's ports 0 and 1 is almost identical to the operation of other ports on the development system. The only difference here is that port signals are received in parallel format. The MCP23S17 converts then such signals into serial format and sends them to the microcontroller. The result is a reduced number of lines used for sending signals from ports 0 and 1 to the microcontroller and vice versa.

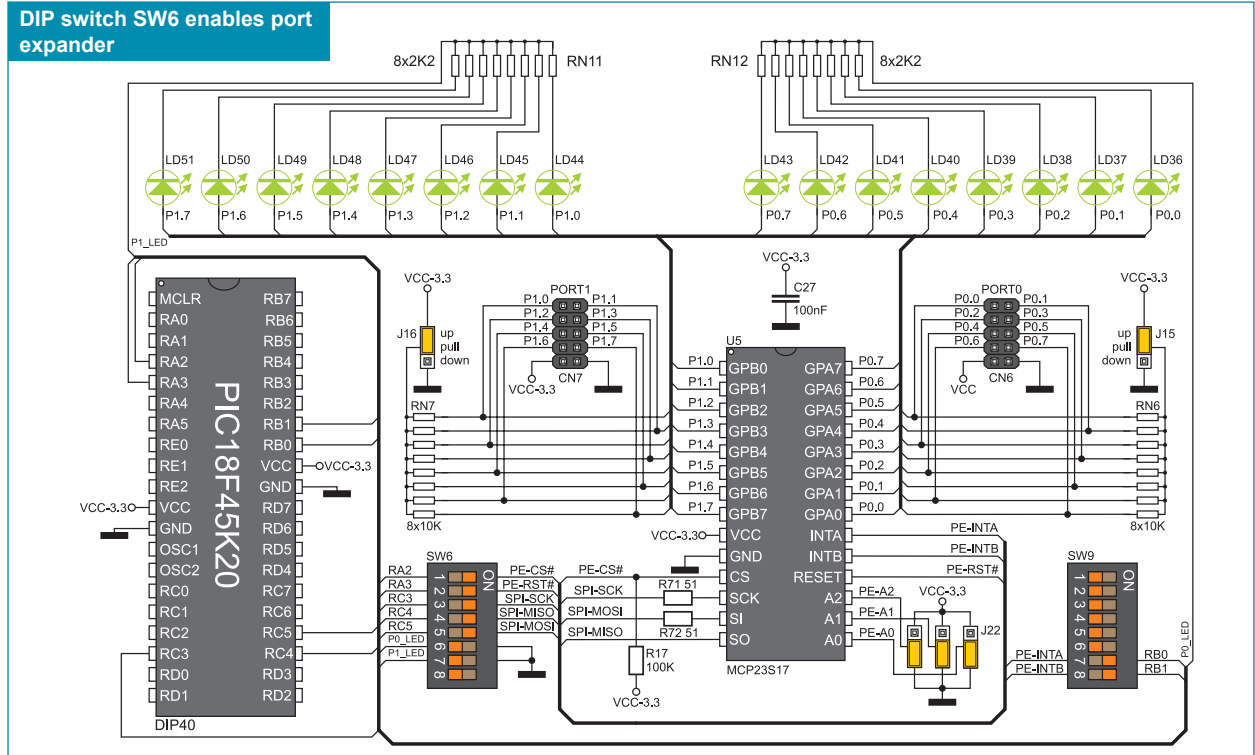


Figure 25-2: Port expander and microcontroller connection schematic

DISCLAIMER

All the products owned by MikroElektronika are protected by copyright law and international copyright treaty. Therefore, this manual is to be treated as any other copyright material. No part of this manual, including product and software described herein, may be reproduced, stored in a retrieval system, translated or transmitted in any form or by any means, without the prior written permission of MikroElektronika. The manual PDF edition can be printed for private or local use, but not for distribution. Any modification of this manual is prohibited.

MikroElektronika provides this manual 'as is' without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties or conditions of merchantability or fitness for a particular purpose.

MikroElektronika shall assume no responsibility or liability for any errors, omissions and inaccuracies that may appear in this manual. In no event shall MikroElektronika, its directors, officers, employees or distributors be liable for any indirect, specific, incidental or consequential damages (including damages for loss of business profits and business information, business interruption or any other pecuniary loss) arising out of the use of this manual or product, even if MikroElektronika has been advised of the possibility of such damages. MikroElektronika reserves the right to change information contained in this manual at any time without prior notice, if necessary.

HIGH RISK ACTIVITIES

The products of MikroElektronika are not fault – tolerant nor designed, manufactured or intended for use or resale as on – line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of Software could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). MikroElektronika and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

TRADEMARKS

The Mikroelektronika name and logo, the Mikroelektronika logo, mikroC, mikroC PRO, mikroBasic, mikroBasic PRO, mikroPascal, mikroPascal PRO, AVRflash, PICflash, dsPICprog, 18FJprog, PSOCprog, AVR-prog, 8051prog, ARMflash, EasyPIC5, EasyPIC6, BigPIC5, BigPIC6, dsPIC PRO4, Easy8051B, EasyARM, EasyAVR5, EasyAVR6, BigAVR2, EasydsPIC4A, EasyPSoC4, EasyVR Stamp LV18FJ, LV24-33A, LV32MX, PIC32MX4 MultiMedia Board, PICPLC16, PICPLC8 PICPLC4, SmartGSM/GPRS, UNI-DS are trademarks of Mikroelektronika. All other trademarks mentioned herein are property of their respective companies.

All other product and corporate names appearing in this manual may or may not be registered trademarks or copyrights of their respective companies, and are only used for identification or explanation and to the owners' benefit, with no intent to infringe.



MikroElektronika
SOFTWARE AND HARDWARE SOLUTIONS FOR EMBEDDED WORLD ...making it simple

If you want to learn more about our products, please visit our website at www.mikroe.com

If you are experiencing some problems with any of our products or just need additional information, please place your ticket at www.mikroe.com/en/support

If you have any questions, comments or business proposals, do not hesitate to contact us at office@mikroe.com

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Mikroe:](#)

[MIKROE-498](#)