BigPIC® 6

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 having confidence in MikroElektronika.
It is our intention to provide you with the best quality products. Furthermore, we will continue to improve our performance to better suit your needs.

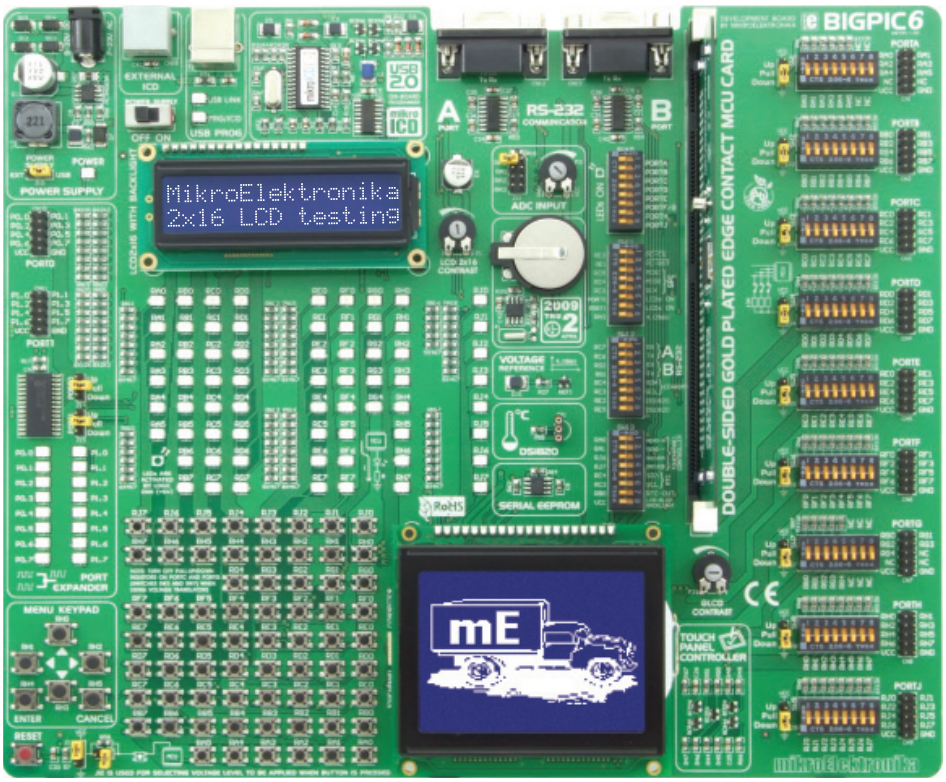
Nebojsa Matic
General Manager

TABLE OF CONTENTS

page

Introduction to BigPIC6 Development System

The *BigPIC®6* is a great development tool suitable for programming and experimenting with PIC® microcontrollers from Microchip®. Such development system includes an on-board programmer with mikroICD support providing an interface between the microcontroller and the PC. You are simply expected to write a code in one of the PIC compilers, generate a .hex file and program your microcontroller using the on-board *PICflash®* programmer. Numerous on-board modules, such as 128x64 graphic LCD display, alphanumeric 2x16 LCD display, port expander etc., allow you to easily simulate the operation of the target device.



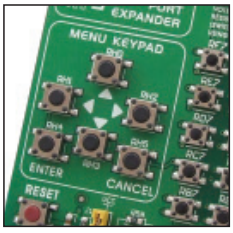
PIC DEVELOPMENT BOARD — Full-featured and user-friendly development board for PIC microcontrollers

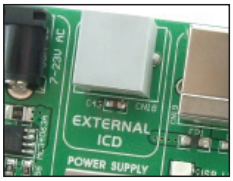
USB 2.0 IN-CIRCUIT PROGRAMMER — High-performance USB 2.0 On-board programmer

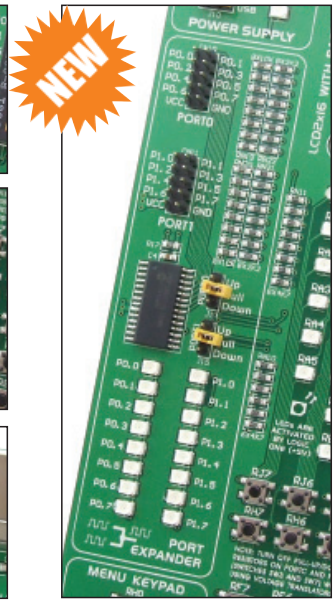
mikro ICD IN-CIRCUIT DEBUGGER — Hardware In-Circuit Debugger for real time debugging at hardware level

PORT EXPANDER — Port expander provides easy I/O expansion by 2 additional ports

GRAPHIC LCD CONNECTOR ON-BOARD — Graphic LCD display with backlight







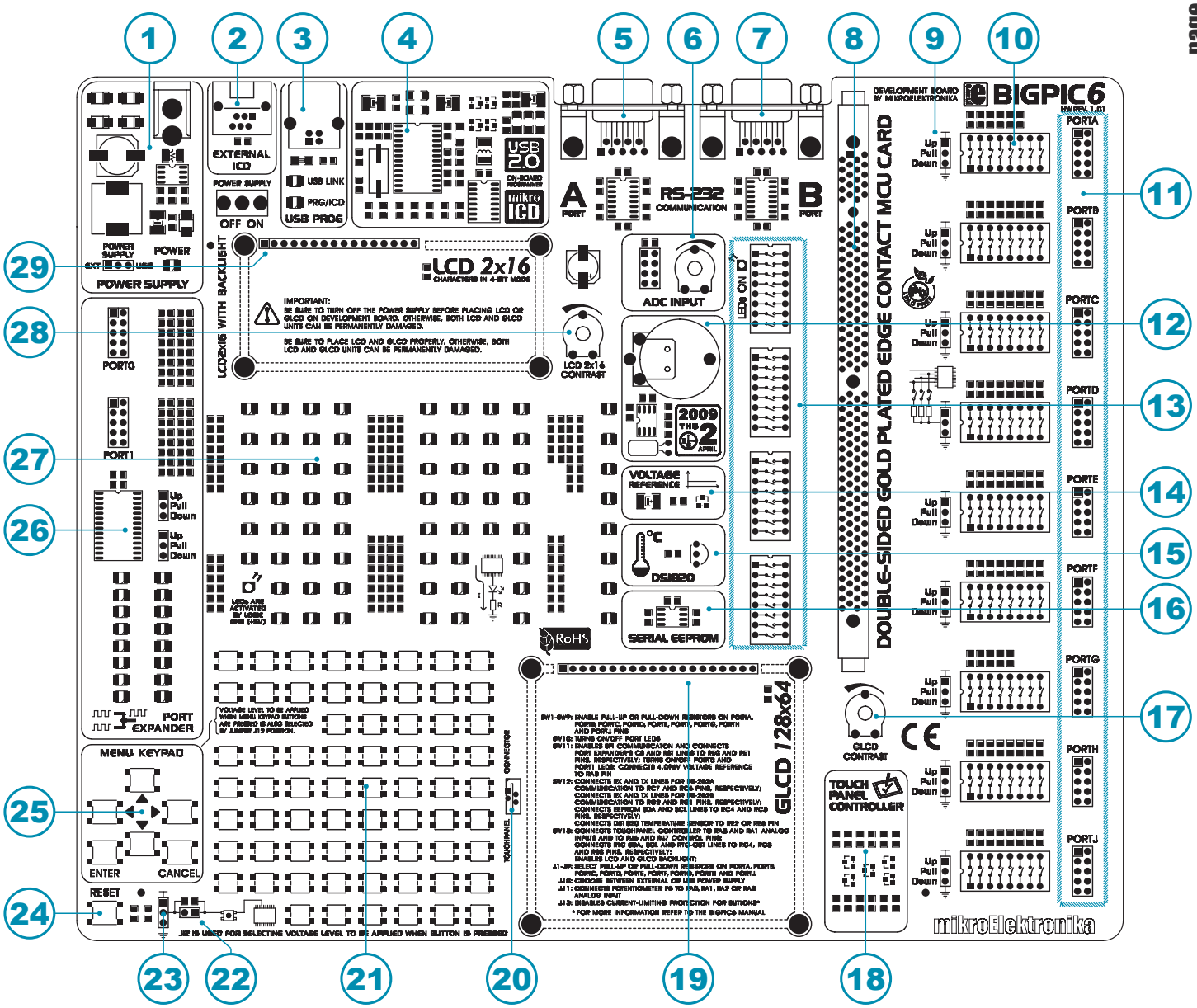
The *PICflash* program provides a complete list of supported microcontrollers. The latest version of this program with updated list of supported microcontrollers can be downloaded from our website at **www.mikroe.com**

Package contains:

Development system:	BigPIC6
CD:	product CD with appropriate software
Cables:	USB cable
Documentation:	BigPIC6 and PICflash manuals, Installing USB drivers quick guide and Electrical Schematic of the BigPIC6 development system

System specification:

Power supply:	over a DC connector (7 to 23V AC or 9 to 32V DC); or over a USB cable for programming (5V DC)
Power consumption:	40mA in idle state (when on-board modules are inactive)
Size:	26,5 x 22cm (10,4 x 8,6inch)
Weight:	~404g (0.89lbs)



Key Features

1. Power supply voltage regulator
2. Microchip debugger connector (ICD2 or ICD3)
3. On-board programmer's USB connector
4. USB 2.0 programmer with *mikroICD* support
5. A connector for RS-232 communication
6. A/D converter test inputs
7. B connector for RS-232 communication
8. DIMM-168P connector for MCU card
9. Pull-up/pull-down resistor selection
10. DIP switches to enable pull-up/pull-down resistors
11. I/O port connectors
12. Real-time clock (RTC) module
13. DIP switches to enable/disable integrated modules
14. 4.096V voltage reference
15. DS1820 temperature sensor
16. Serial EEPROM
17. Graphic LCD display contrast adjustment
18. Touch panel controller
19. Graphic LCD display connector
20. Touch panel connector
21. Push buttons to simulate digital inputs
22. Protective resistor ON/OFF jumper
23. Pins' logic state selector
24. Reset button
25. MENU keypad
26. Port expander
27. 67 LEDs to indicate pins' logic state
28. Alphanumeric LCD display contrast adjustment
29. Alphanumeric LCD display connector

1.0. Connecting the System to your PC

Step 1:

Follow the instructions for installing USB drivers and the *PICflash with mikroICD* programmer provided in the relevant manuals. It is not possible to program PIC microcontrollers without having these devices installed first.
In case you already have some of the MikroElektronika's compilers installed on your PC, there is no need to reinstall drivers as they will be automatically installed along with the compiler.

Step 2:

Use the USB cable to connect the *BigPIC6* development system to your PC. One end of the USB cable provided with a connector of the USB **B** type should be connected to the development system as shown in Figure 1-2, whereas the other end of the cable (USB **A** type) should be connected to your PC. When establishing a connection, make sure that jumper J10 is placed in the USB position as shown in Figure 1-1.

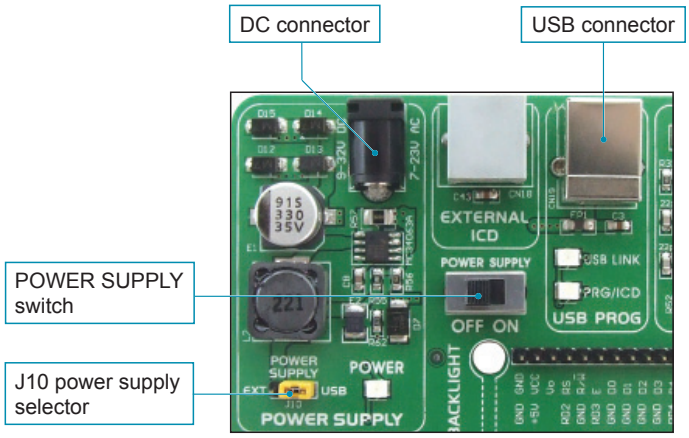


Figure 1-1: Power supply

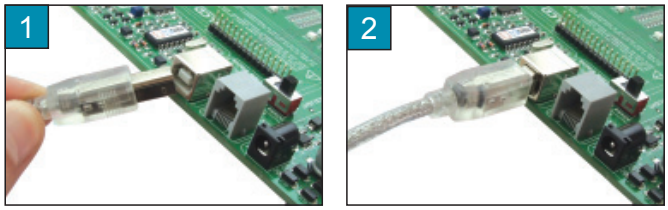


Figure 1-2: Connecting USB cable

Step 3:

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

NOTE: If you use additional modules, such as LCD, GLCD etc., it is necessary to place them properly on the development board before it is turned on. Otherwise, both additional modules and development system can be permanently damaged. Refer to Figure 1-3 for their proper placing.

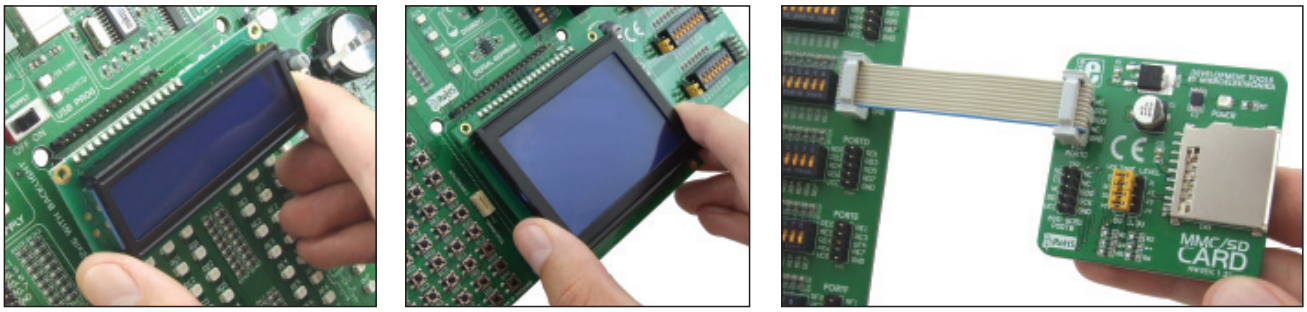


Figure 1-3: Placing additional modules on the board

2.0. Supported Microcontrollers

The *BigPIC6* development system provides a DIMM-168P connector used for placing MCU card. The *BigPIC6* comes with an MCU card with the 80-pin microcontroller in TQFP package soldered on it, as shown in Figure 2-3. Besides, an oscillator and 80 pads connected to the microcontroller pins are also provided on the MCU card. Each pad is marked as its relevant pin. These pads make placing of MCU card easy when it is used in the target device.

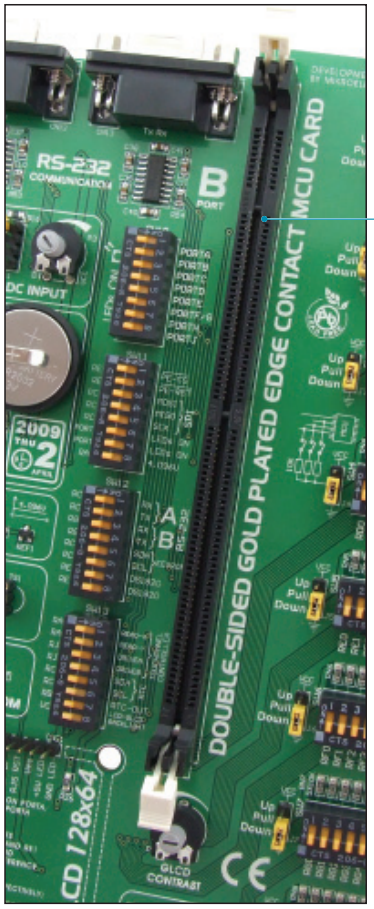


Figure 2-1: DIMM-168P connector

DIMM-168P connector for placing MCU card with the microcontroller in TQFP package

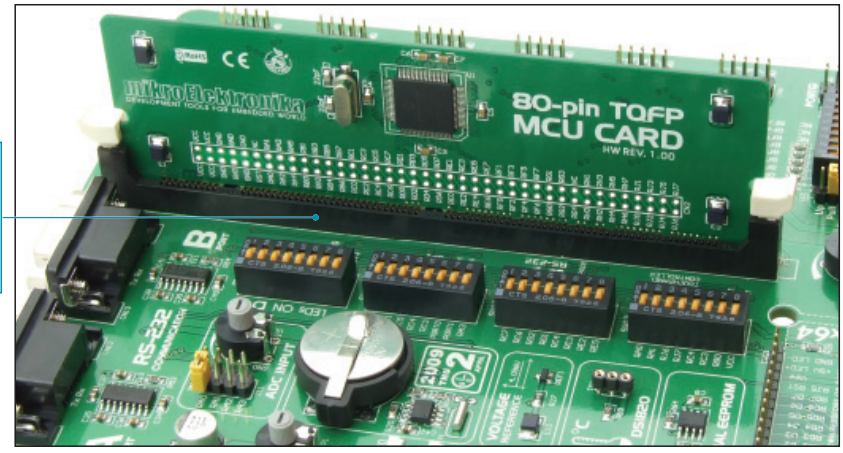


Figure 2-2: DIMM-168P connector with MCU card plugged into it

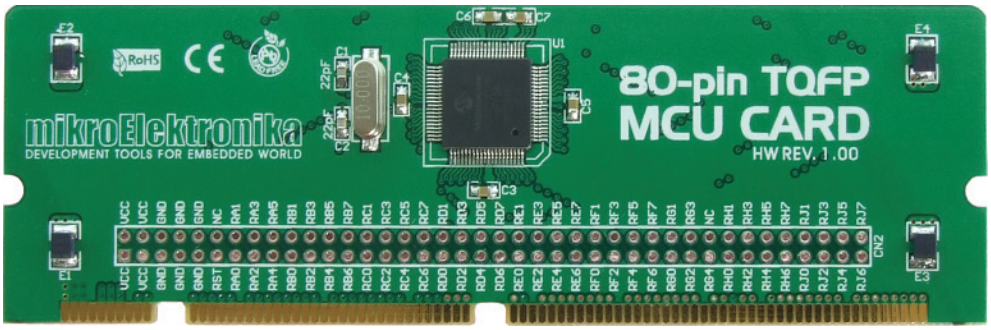


Figure 2-3: MCU card with soldered 80-pin microcontroller in TQFP package

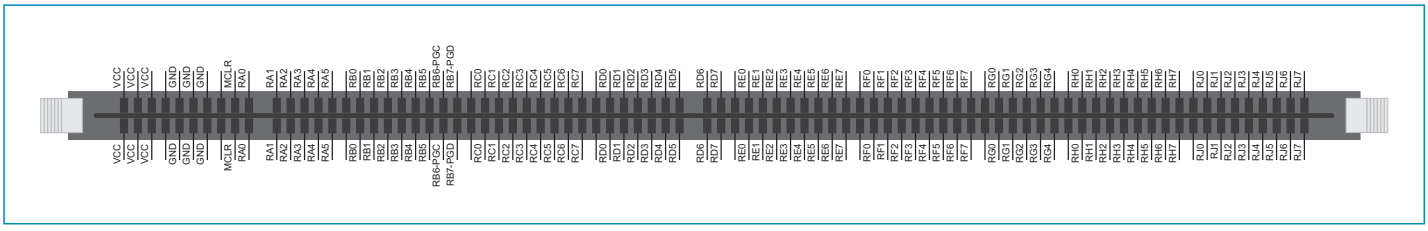
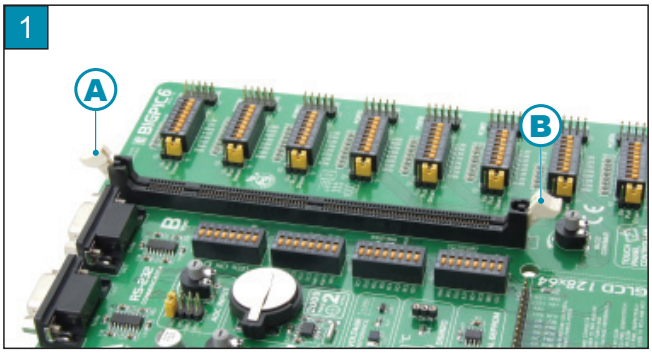
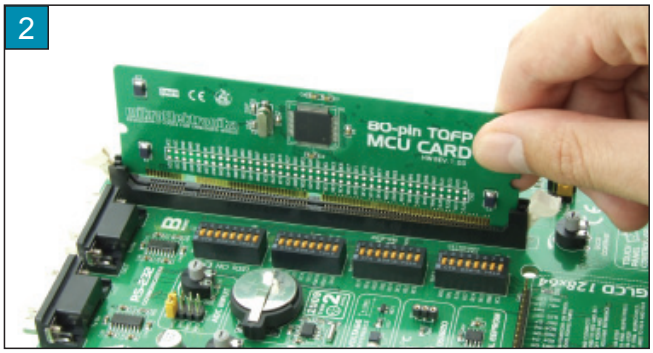


Figure 2-4: Schematic of DIMM-168P connector pinout

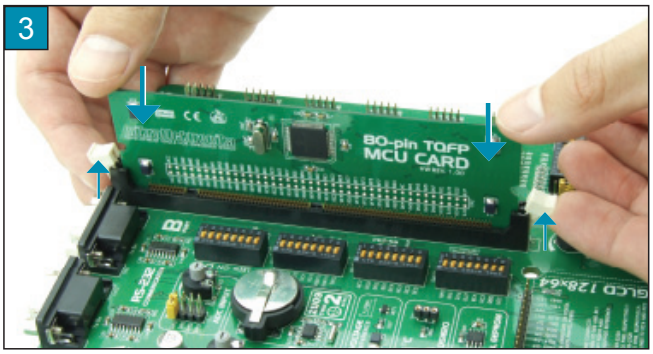
Plugging MCU card into the DIMM-168P connector is performed as follows:



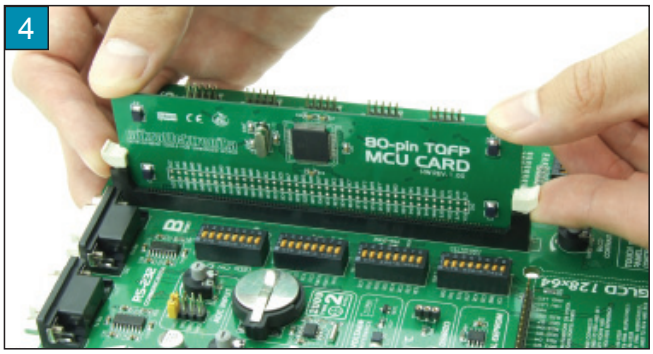
Open extraction levers A and B



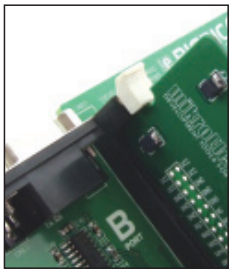
Plug the MCU card into DIMM-168P connector



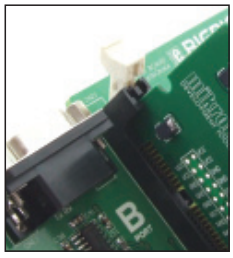
Push down gently MCU card into DIMM-168P connector and slowly lift extraction levers



When the MCU card is properly placed into the connector, the extraction levers must be closed



Extraction levers used for fixing MCU card in the closed position



Extraction levers used for fixing MCU card in the opened position

In addition to MCU card with the microcontroller in 80 pin TQFP package, there are also cards with microcontrollers in 64 pin TQFP package which can be ordered separately. They are plugged into the connector in the same way as the above mentioned card.

3.0. On-board USB 2.0 PICflash with mikroICD Programmer

A programmer is a necessary tool when working with the microcontroller. The *BigPIC6* development system has an on-board *PICflash* programmer with mikroICD support which allows you to establish a connection between the microcontroller and your PC. Use the *PICflash* program to load a .hex file into the microcontroller. Figure 3-2 shows the connection between a compiler, *PICflash* programmer and microcontroller.

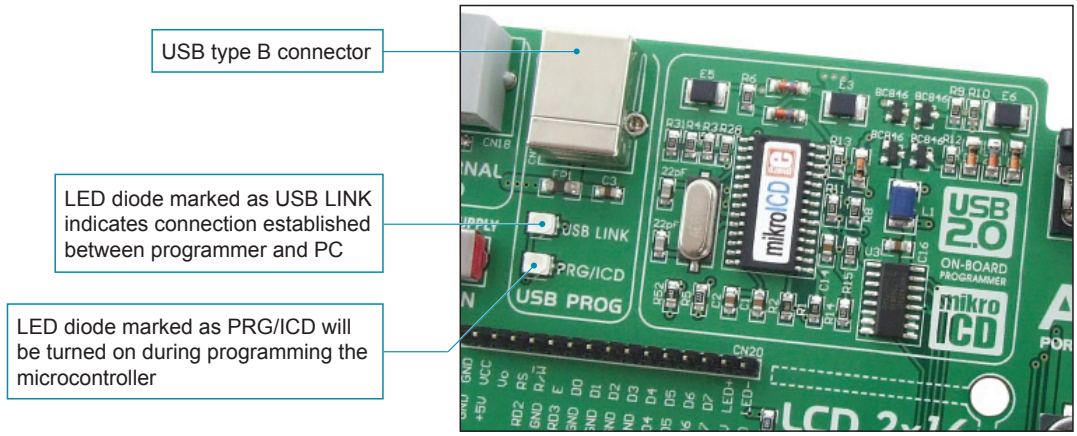


Figure 3-1: *PICflash* programmer

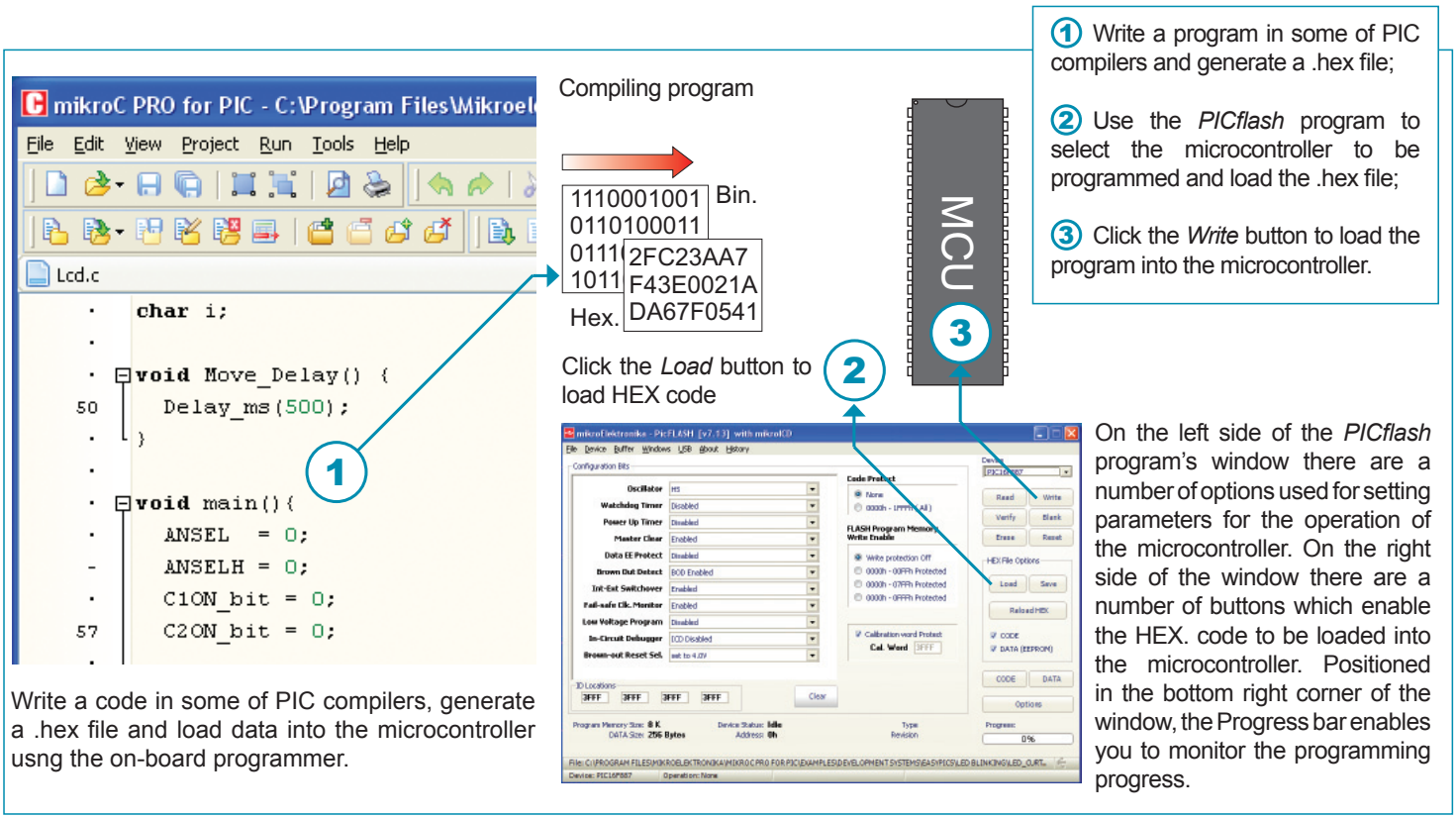


Figure 3-2: The principle of programmer's operation

NOTE: For more information on the *PICflash* programmer refer to the relevant manual provided in the *BigPIC6* development system package.

page

PIC microcontrollers can be programmed either in *Low Voltage* or *High Voltage* programmings modes. The *PICflash* programmer uses solely *High Voltage* programming mode for its operation. Such mode requires voltage higher than the microcontroller's power supply voltage to be brought to the MCLR/Vpp pin in order for the programming process to be performed. The voltage value usually ranges between 8 and 14V depending on the type of the microcontroller in use.

The *Low Voltage* programming mode can be enabled/disabled using configuration bits of the microcontroller. If the *Low Voltage* programming mode is enabled, the programming process is initiated by applying a logic one (1) to the PGM pin. Unlike this mode, the *High Voltage* programming mode is always enabled and the programming process starts by applying a high voltage to the MCLR/VPP pin.

All the settings that have to do with programming the microcontrollers are automatically performed and no extra work is needed. However, there is a number of options for additional programming settings provided in the *PICflash* program. It is not recommended for beginners to change the default settings.

One of the advantages offered by the on-board *PIClash* programmer is a multiplexer.

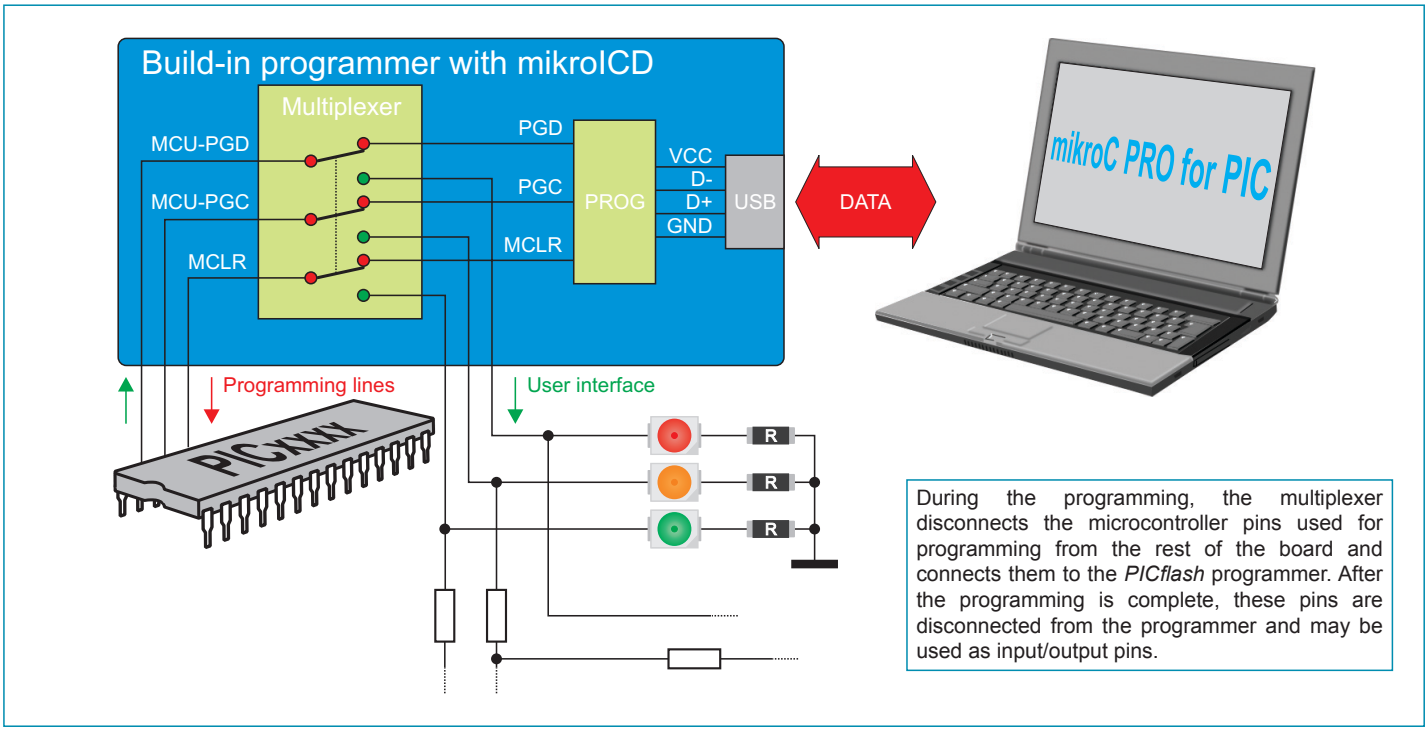


Figure 3-3: The principle of programmer's operation

4.0. ICD Connector

ICD connector enables communication between the microcontroller and external ICD debugger/programmer from Microchip (ICD2® or ICD3®).

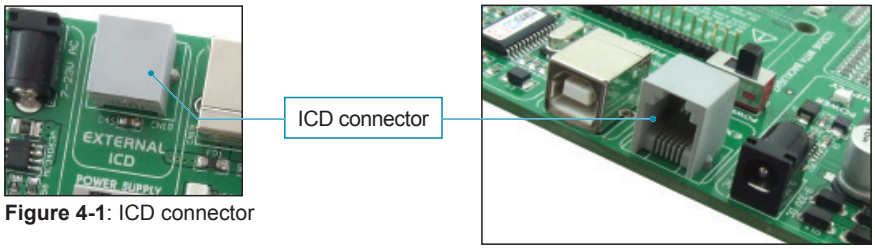


Figure 4-1: ICD connector

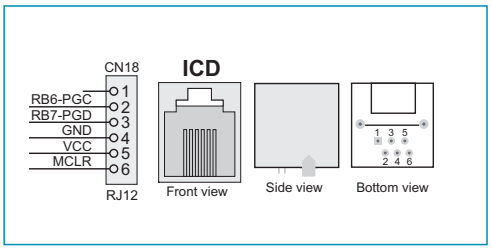


Figure 4-2: ICD connector pinout and pin designations

5.0. 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® etc.). As soon as the *mikroICD* debugger starts up, the *Watch Values* window, as shown in figure 5-1 below, appears. Communication between *mikroICD* debugger and microcontroller is enabled via programming pins.

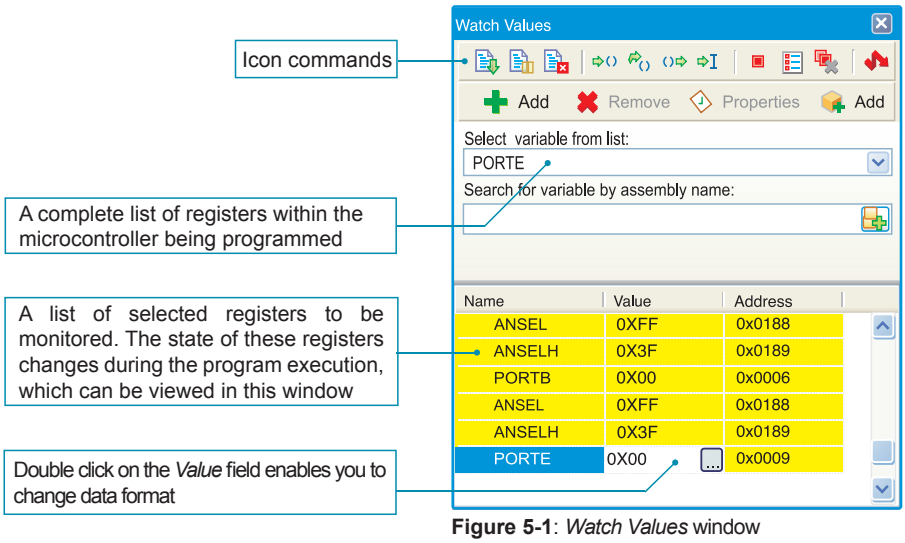


Figure 5-1: *Watch Values* window

mikroICD debugger options:

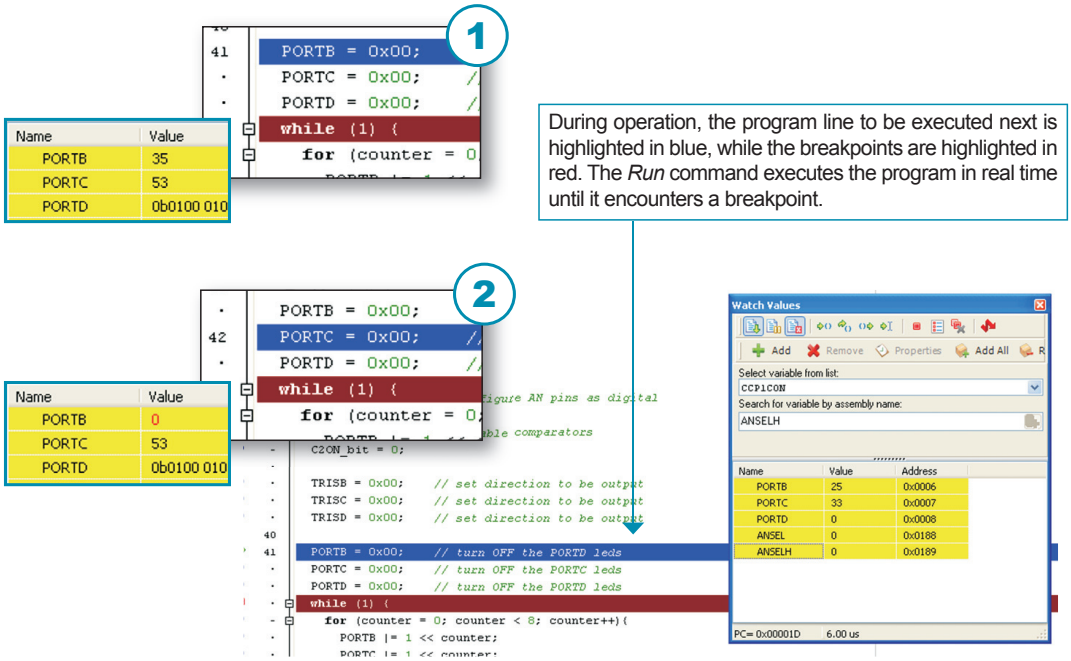
Start Debugger	[F9]
Run/Pause Debugger	[F6]
Stop Debugger	[Ctrl+F2]
Step Into	[F7]
Step Over	[F8]
Step Out	[Ctrl+F8]
Toggle Breakpoint	[F5]
Show/Hide Breakpoints	[Shift+F4]
Clear Breakpoints	[Ctrl+Shift+F4]

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

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.



During operation, the program line to be executed next is highlighted in blue, while the breakpoints are highlighted in red. The *Run* command executes the program in real time until it encounters a breakpoint.

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

page

6.0. Power Supply

The *BigPIC6* development system may use one of two power supply sources:

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

The MC34063A voltage regulator and Gretz rectifier are used for enabling 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 power supply selector. When using USB power supply, jumper J10 should be placed in the USB position. When using external power supply, jumper J10 should be placed in the EXT position. The development system is turned on by setting the POWER SUPPLY switch to the ON position.

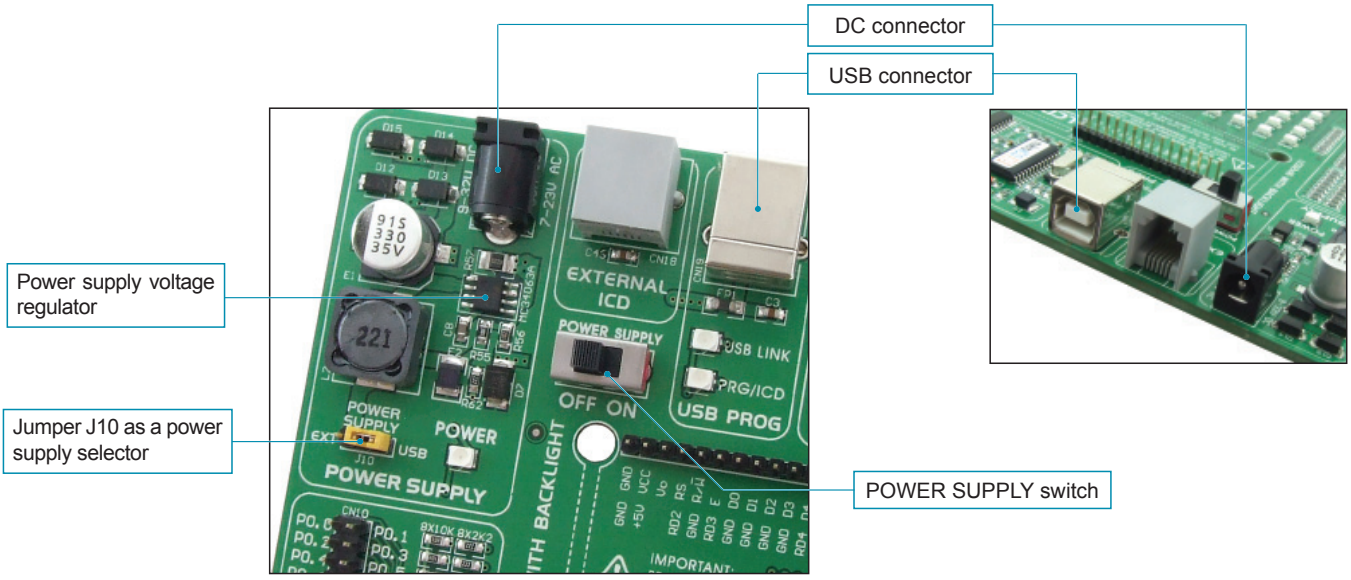


Figure 6-1: Power supply

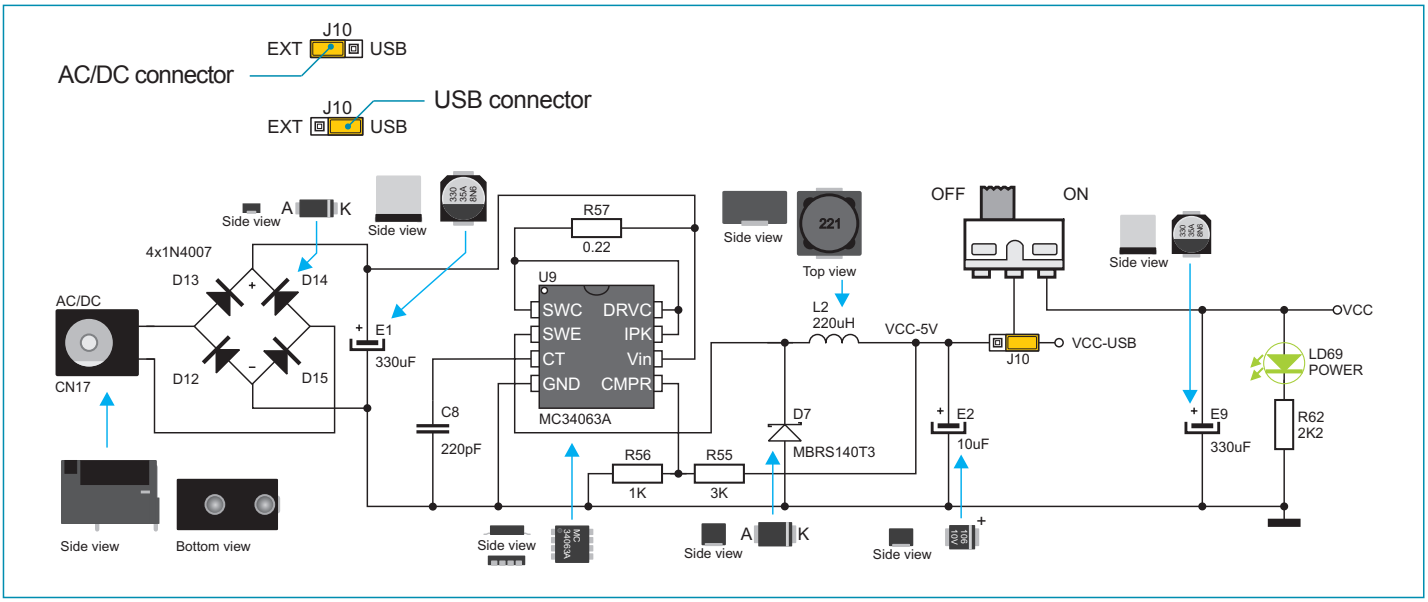


Figure 6-2: Power supply source connection schematic

7.0. RS-232 Communication Interface

The USART (universal synchronous/asynchronous receiver/transmitter) is one of the most common ways of exchanging data between the PC and peripheral components. RS-232 serial communication is performed through a 9-pin SUB-D connector and the microcontroller USART module. The *BigPIC6* provides two RS-232 ports, RS-232A and RS-232B. Use switches marked as RX232-A and TX232-A on the DIP switch SW12 to enable RS-232A port. Likewise, use switches RX232-B and TX232-B on the DIP switch SW12 to enable RS-232B port. The microcontroller pins used in such communication are marked as follows: RX - *receive data* and TX - *transmit data*. Baud rate goes up to 115 kbps.

In order to enable the USART module of the microcontroller to receive input signals with different voltage levels, it is necessary to provide a voltage level converter such as MAX202C (MAX232).

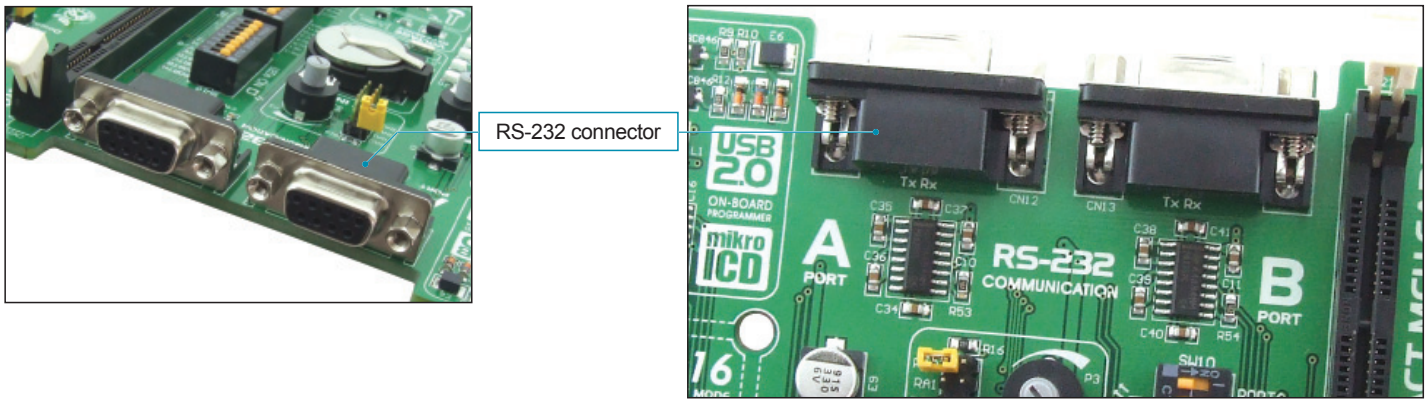


Figure 7-1: RS-232 module

The function of switches 1, 2, 3 and 4 on the DIP switch SW12 is to determine which of the microcontroller pins are to be used as RX and TX lines, Figure 7-2.

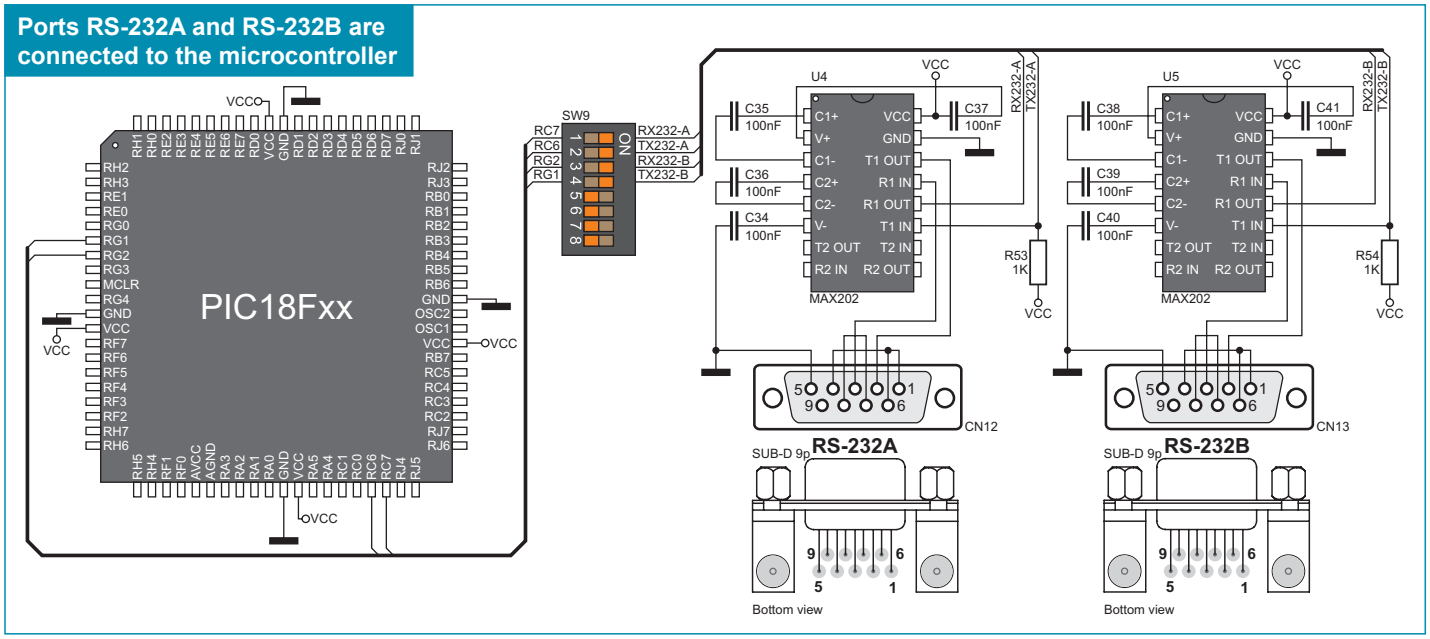


Figure 7-2: RS-232 module schematic

NOTE: Make sure that your microcontroller is provided with the USART module as it is not necessarily integrated in all PIC microcontrollers.

page

8.0. Serial EEPROM

EEPROM (Electrically Erasable Programmable Read-Only Memory) is a built-in memory module used to store data that must be saved when power goes off. The 24AA01 circuit can store up to 1Kbit data and uses serial I^2C communication via RC3 and RC4 pins for communication with the microcontroller. In order to enable connection between EEPROM and microcontroller, it is necessary to set switches 5 and 6 on the DIP switch SW12 to ON position.

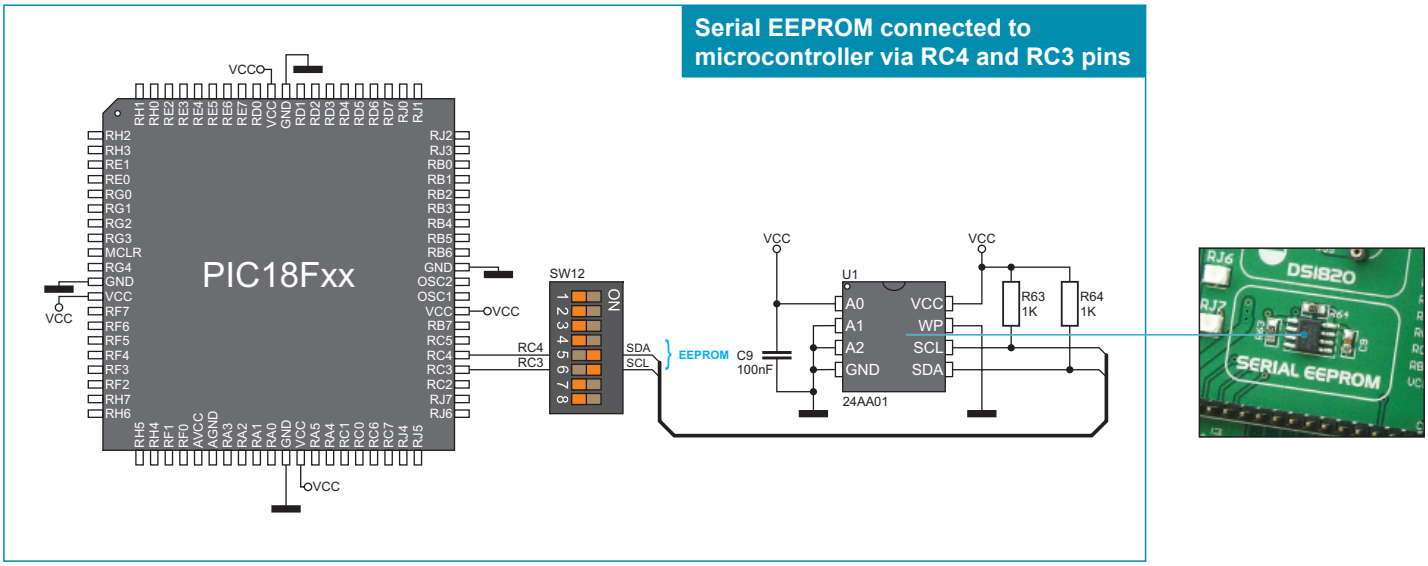


Figure 8-1: Serial EEPROM connection schematic

9.0. Voltage Reference

The *BIGPIC6* development system provides an MCP1541 circuit which generates the voltage reference used for A/D conversion. The value of the voltage reference is 4.096V and it is brought to the microcontroller via the RA3 pin.

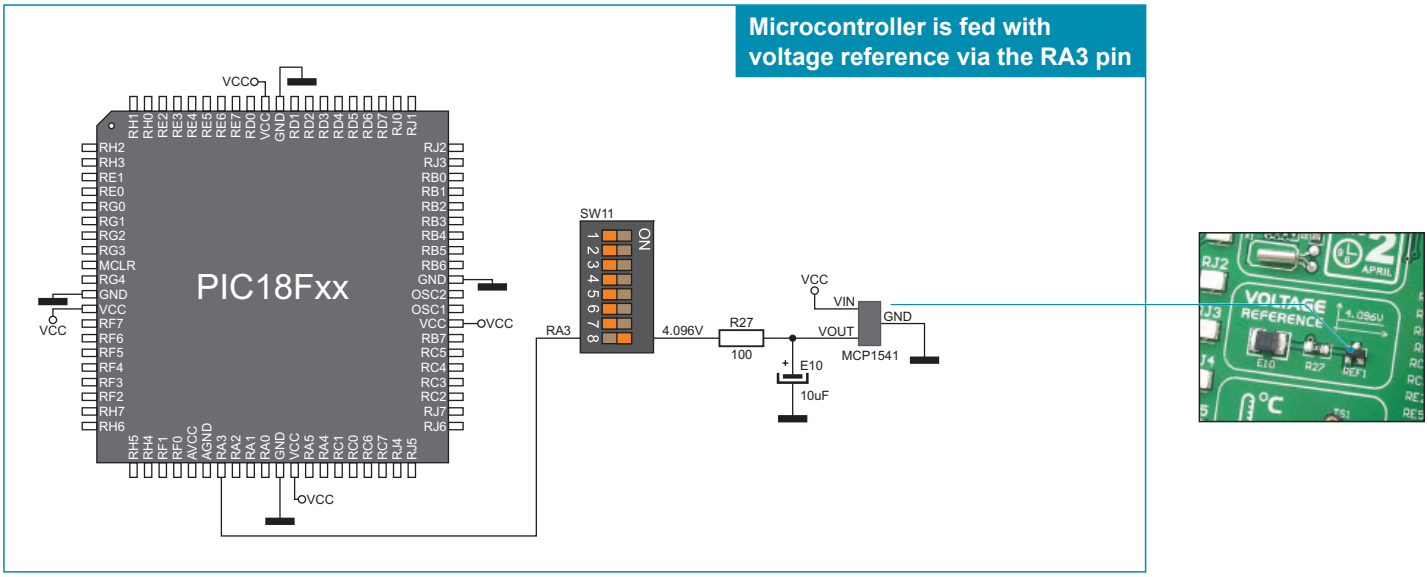


Figure 9-1: Voltage reference connection schematic

10.0. A/D Converter Test Inputs

An A/D converter is used for converting an analog signal into the appropriate digital value. A/D converter is linear, which means that the converted number is linearly dependent on the input voltage value. The A/D converter within the microcontroller converts an analog voltage value into a 10-bit number. Voltages varying from 0V to 5V DC may be supplied through the A/D test inputs. Jumper J11 is used for selecting some of the following pins RA0, RA1, RA2 or RA3 for A/D conversion. The R16 resistor has a protective function as it is used for limiting current flow through the potentiometer or the microcontroller pin. The value of the input analog voltage can be changed linearly using potentiometer P3.

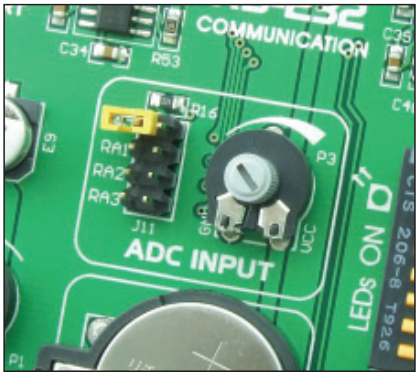


Figure 10-1: ADC (jumper in default position)

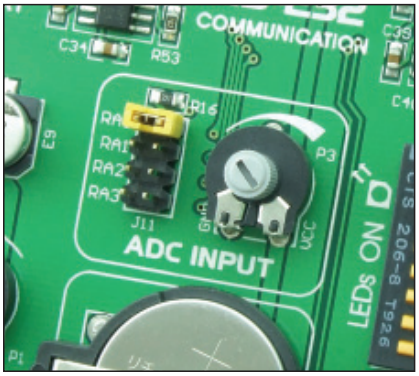


Figure 10-2: Pin RA0 used as input pin for A/D conversion

A/D conversion is performed via the RA0 microcontroller pin

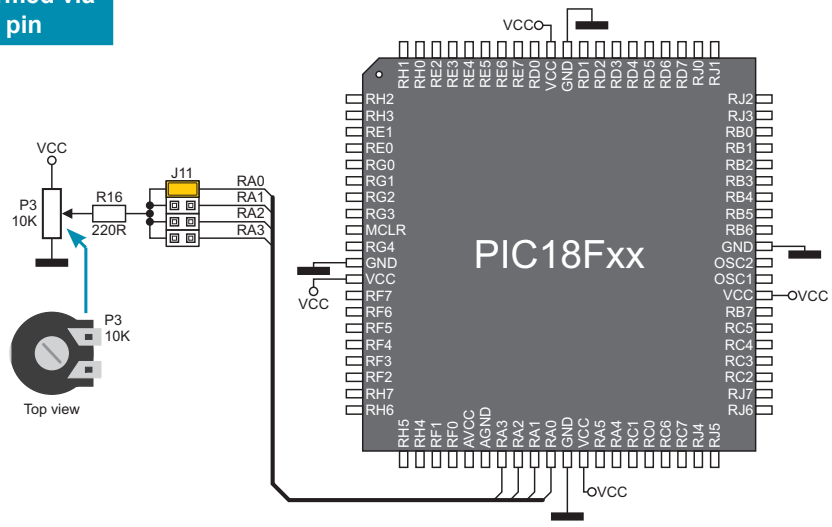


Figure 10-3: Microcontroller and A/D converter test inputs connection

NOTE: In order to enable the microcontroller to accurately perform A/D conversion, it is necessary to turn off LED diodes and pull-up/pull-down resistors on port pins used by the A/D converter. For higher A/D conversion accuracy use the voltage reference.

page

11.0. DS1820 Temperature Sensor

1-wire® serial communication enables data to be transferred over one single communication line while the process itself is under the control of the *master* microcontroller. 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 interface.

DS1820 is a temperature sensor that uses 1-wire standard for its operation. It is capable of measuring temperatures within the range of -55 to 125°C and provides ±0.5°C accuracy for temperatures within the range of -10 to 85°C. Power supply voltage of 3V to 5.5V is required for its operation. It takes maximum 750ms for the DS1820 to calculate temperature with 9-bit resolution. The *BigPIC6* development system provides a separate socket for the DS1820. It may use either RE2 or RE5 pin for communication with the microcontroller, which depends on the position of switches 7 and 8 on the DIP switch SW12. In Figure 11-5, switch 8 on the DIP switch SW12 is in ON position which means that the communication is enabled via the RE5 pin.

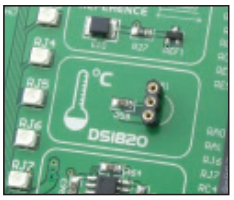


Figure 11-1: DS1820 connector (DS1820 is not placed)

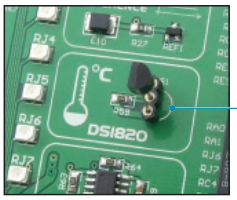


Figure 11-2: DS1820 is plugged into the connector

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

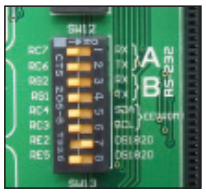


Figure 11-3: Switch 7 on the DIP switch SW12 is in ON position, DS1820 is connected to the PE2 pin

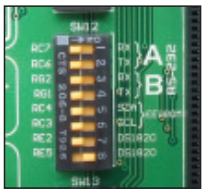


Figure 11-4: Switch 8 on the DIP switch SW12 is in ON position, DS1820 is connected to the PE5 pin

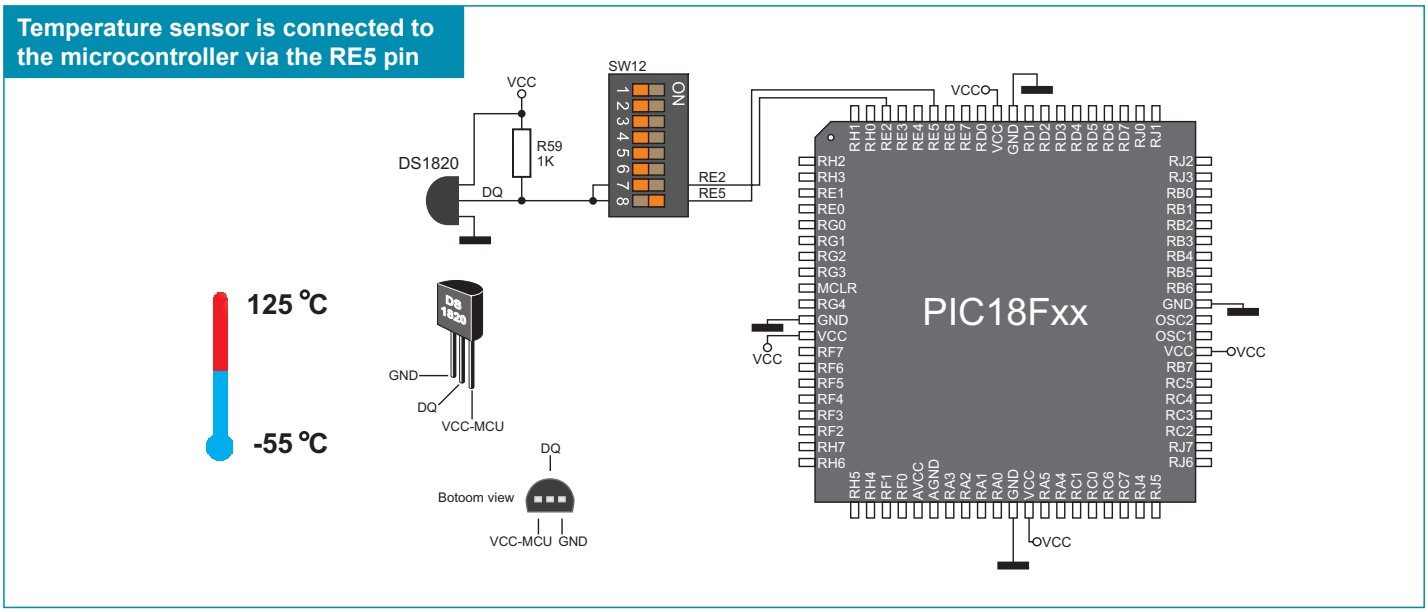


Figure 11-5: DS1820 and microcontroller connection schematic

12.0. Real-Time Clock (RTC)

The DS1307 circuit enables the *BigPIC6* development system to keep the real time. The real-time clock's main features are as follows:

- providing information on seconds, minutes, hours, days, days in a week and dates including corrections for a leap year
- I^2C serial interface
- Automatic power-fail detection
- Power consumption less than 500nA

The real-time clock is widely used in alarm devices, industrial controllers, mass-consumption products etc. The real-time clock provided on the *BigPIC6* development system is used to generate an interrupt at pre-set time. In order to establish the connection between the microcontroller and real-time clock it is necessary to set switches RC4, RC3 and RB0 on the DIP switch SW13 to ON position.

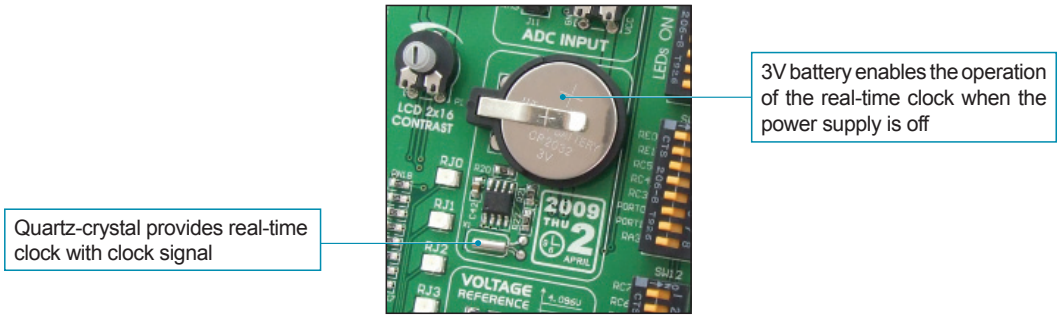


Figure 12-1: Real-time clock

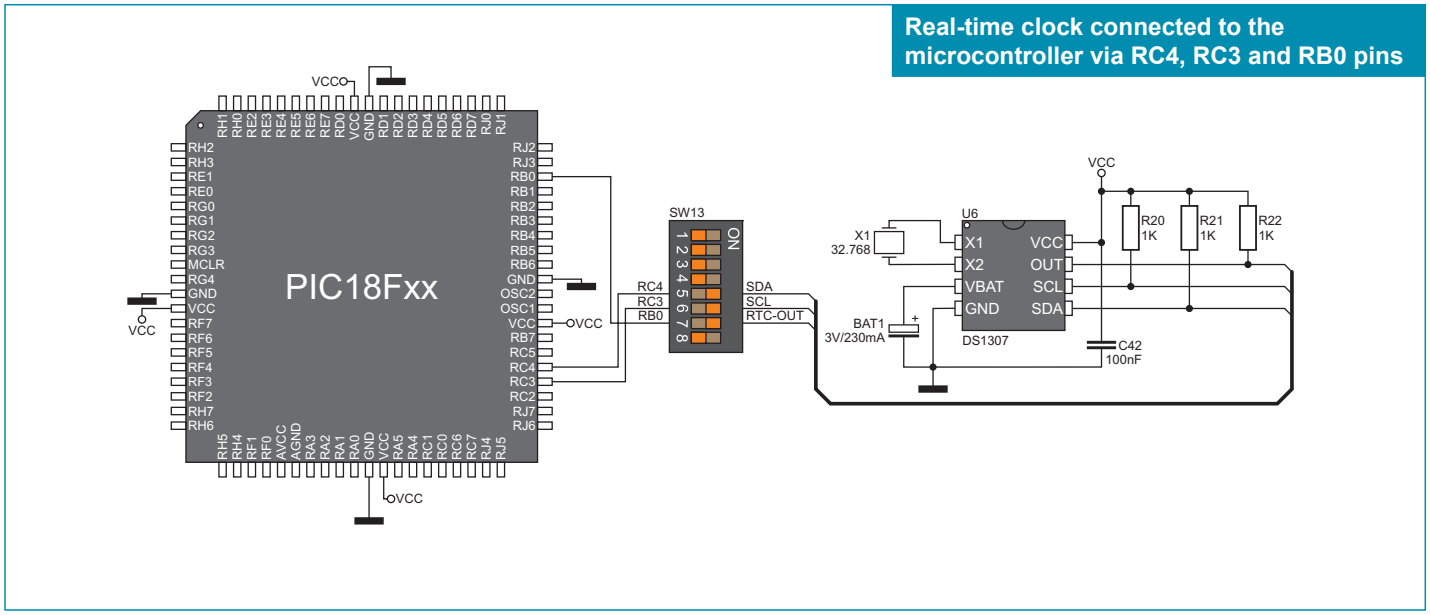


Figure 12-2: Real-time clock and microcontroller connection schematic

page

13.0. LEDs

LED diode (Light-Emitting Diode) is a highly efficient electronic light source. When connecting LEDs, it is necessary to use a current limiting resistor, the value of which is calculated using formula R=U/I where R is referred to resistance expressed in ohms, U is referred to voltage on the LED and I stands for LED diode current. A common LED diode voltage is approximately 2.5V, while the current varies from 1mA to 20mA depending on the type of LED diode. The *BigPIC6* development system uses LEDs with current I=1mA.

The *BigPIC6* development system has 67 LEDs which visually indicate the state of each microcontroller I/O pin. An active LED diode 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, PORTE, PORTF/G, PORTH or PORTJ using the DIP switch SW10.

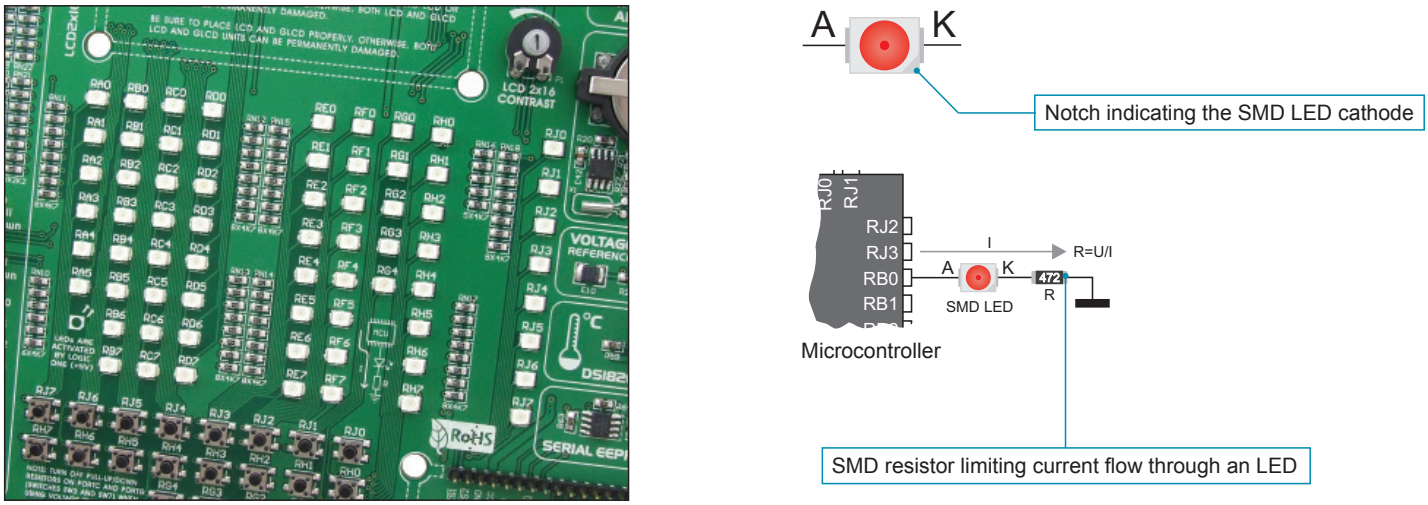


Figure 13-1: LEDs

PORTB LEDs are turned on

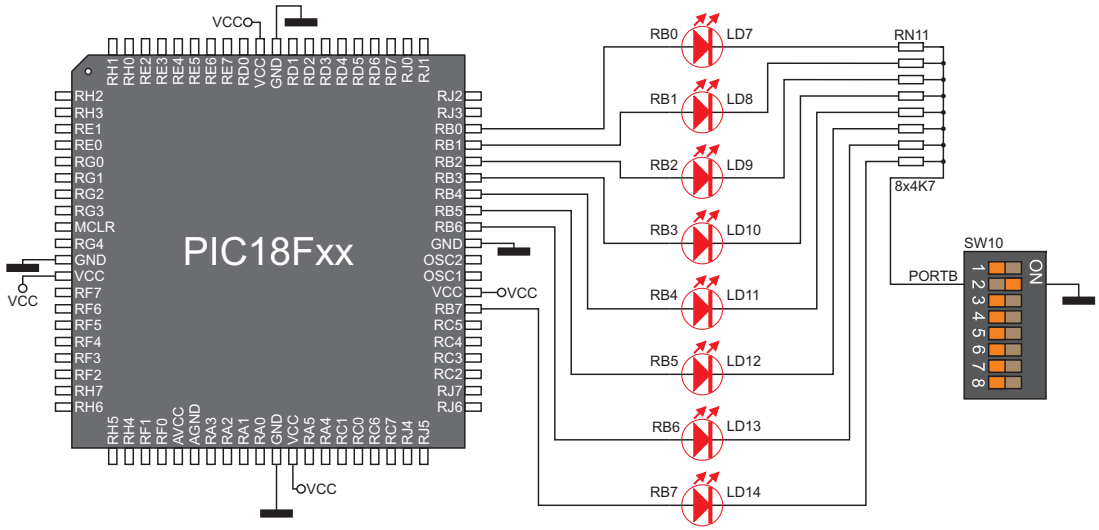


Figure 13-2: LED diode and port PORTB connection schematic

14.0. Push Buttons

The logic state of all microcontroller digital inputs may be changed using the push buttons. Jumper J12 is used to determine the logic state to be applied to the desired microcontroller pin by pressing the appropriate push button. The purpose of the protective resistor is to limit the maximum current thus preventing a short circuit from occurring. If needed, advanced users may short such resistor using jumper J13. Right next to the push buttons, there is a RESET button which is not connected to the MCLR pin. The reset signal is generated by the programmer.

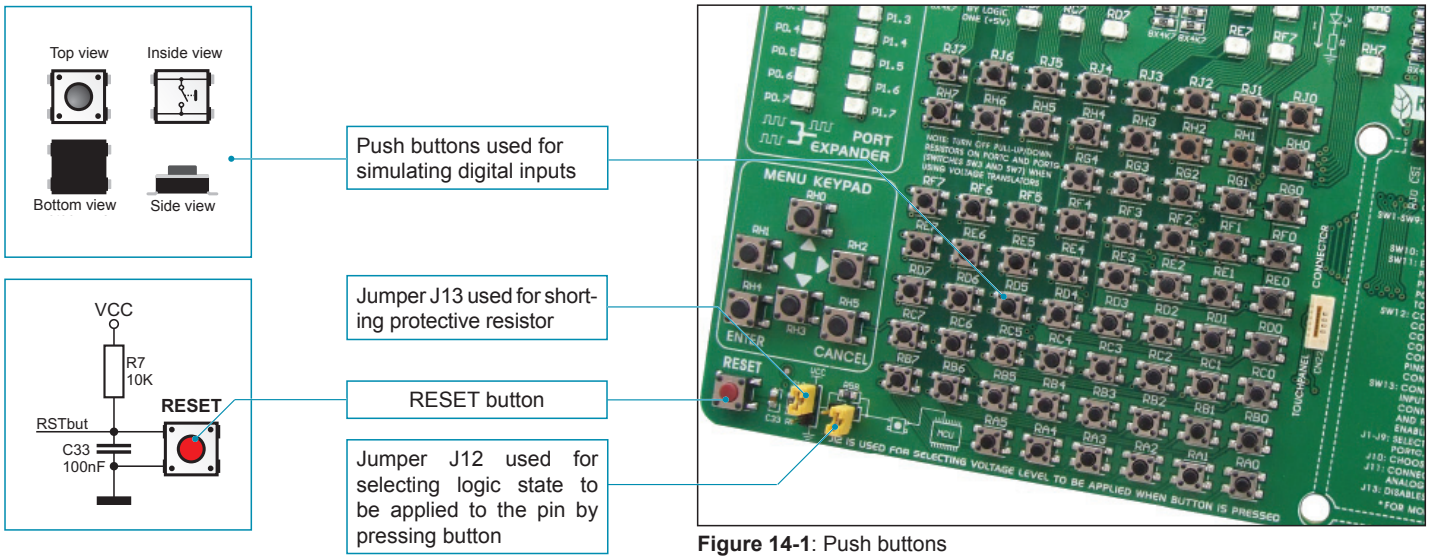


Figure 14-1: Push buttons

By pressing any push button when jumper J12 is in the VCC position, a logic one (5V) will be applied to the appropriate microcontroller pin as shown in Figure 14-2.

Jumper J12 in the VCC position

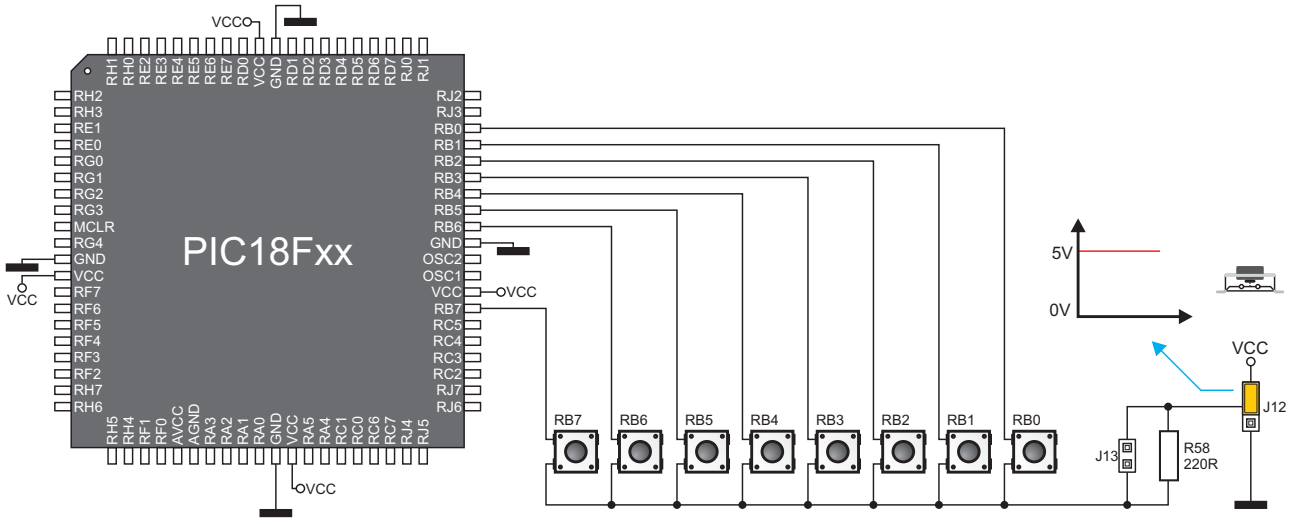


Figure 14-2: Push buttons and port PORTB connection schematic

page

15.0. MENU Keypad

There is a navigation keypad called MENU provided on the *BigPIC6* development system. It primarily consists of four push buttons marked as left, right, up and down arrow. Besides, there are also two additional push buttons marked as ENTER and CANCEL. MENU push buttons are connected in the same way as the port PORTH push buttons. Their function is determined by the user when writing the program for the microcontroller.

Have in mind when writing the program for the microcontroller that MENU keypad is connected to the port PORTH

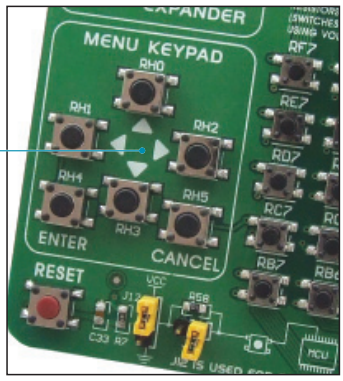


Figure 15-1: MENU keypad

MENU keypad is connected the same as port PORTH push buttons

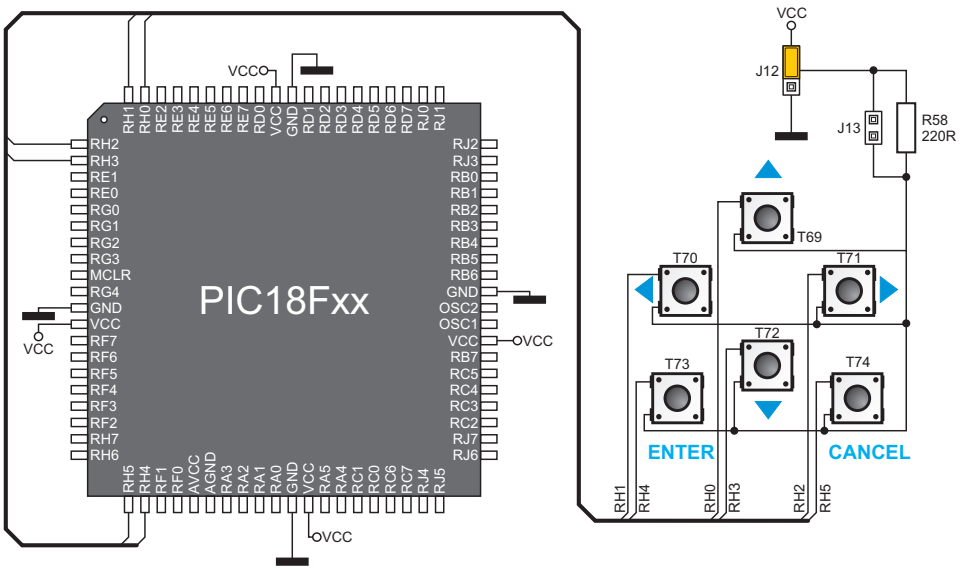


Figure 15-2: MENU keypad and microcontroller connection schematic

16.0. 2x16 LCD Display

The *BigPIC6* development system provides an on-board connector so that the alphanumeric 2x16 LCD display can be plugged into. Such connector is linked to the microcontroller through the PORTD port. Potentiometer P1 is used for display contrast adjustment. The *LCD-GLCD BACKLIGHT* switch on the DIP switch SW13 is used for turning on/off display backlight.
Communication between such LCD display and the microcontroller is established by using a 4-bit mode. Alphanumeric digits are displayed in two lines each containing up to 16 characters of 7x5 pixels.

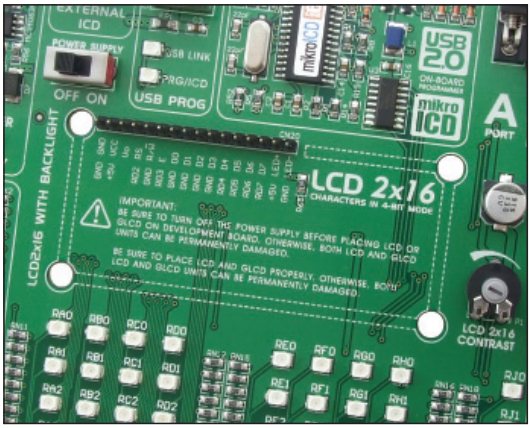


Figure 16-1. Alphanumeric 2x16 LCD display connector

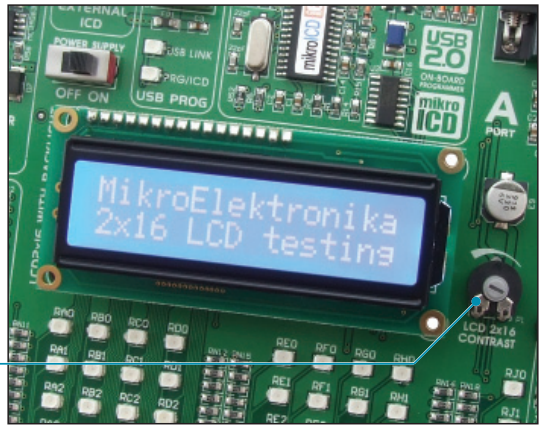


Figure 16-2: Alphanumeric 2x16 LCD display

LCD display backlight is turned on

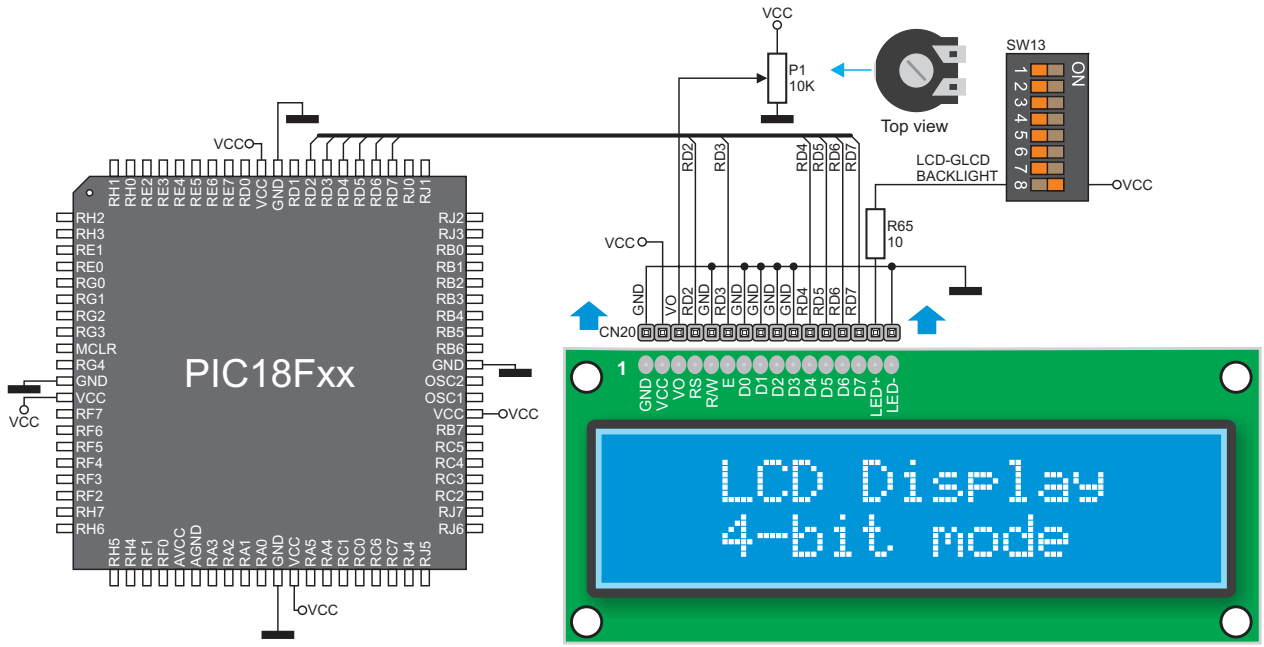


Figure 16-3: Alphanumeric 2x16 LCD display connection schematic

page

17.0. 128x64 Graphic LCD Display

128x64 graphic LCD display (128x64 GLCD) provides an advanced method for displaying graphic messages. It is connected to the microcontroller through PORTD and PORTJ. GLCD display has the screen resolution of 128x64 pixels which allows you to display diagrams, tables and other graphic content. Since the PORTD port is also used by 2x16 alphanumeric LCD display, you cannot use both displays simultaneously. Potentiometer P2 is used for the GLCD display contrast adjustment. Switch 8 (LCD-GLCD BACKLIGHT) on the DIP switch SW13 is used for turning on/off display backlight.

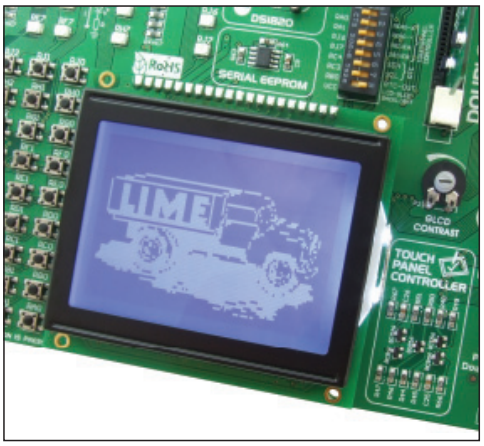


Figure 17-1: GLCD display

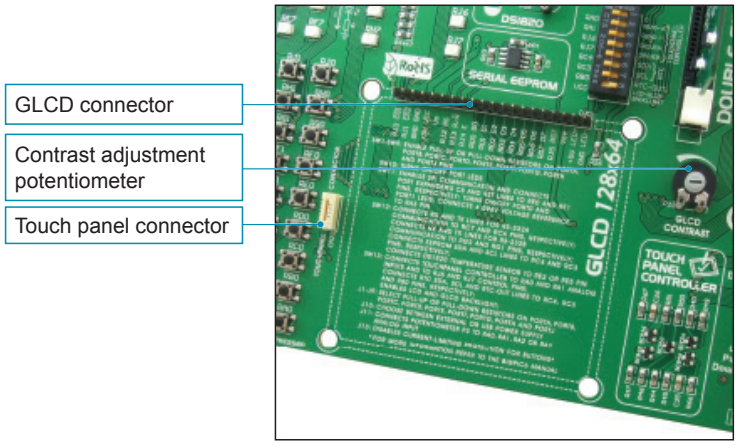


Figure 17-2: GLCD connector

GLCD display backlight is turned on

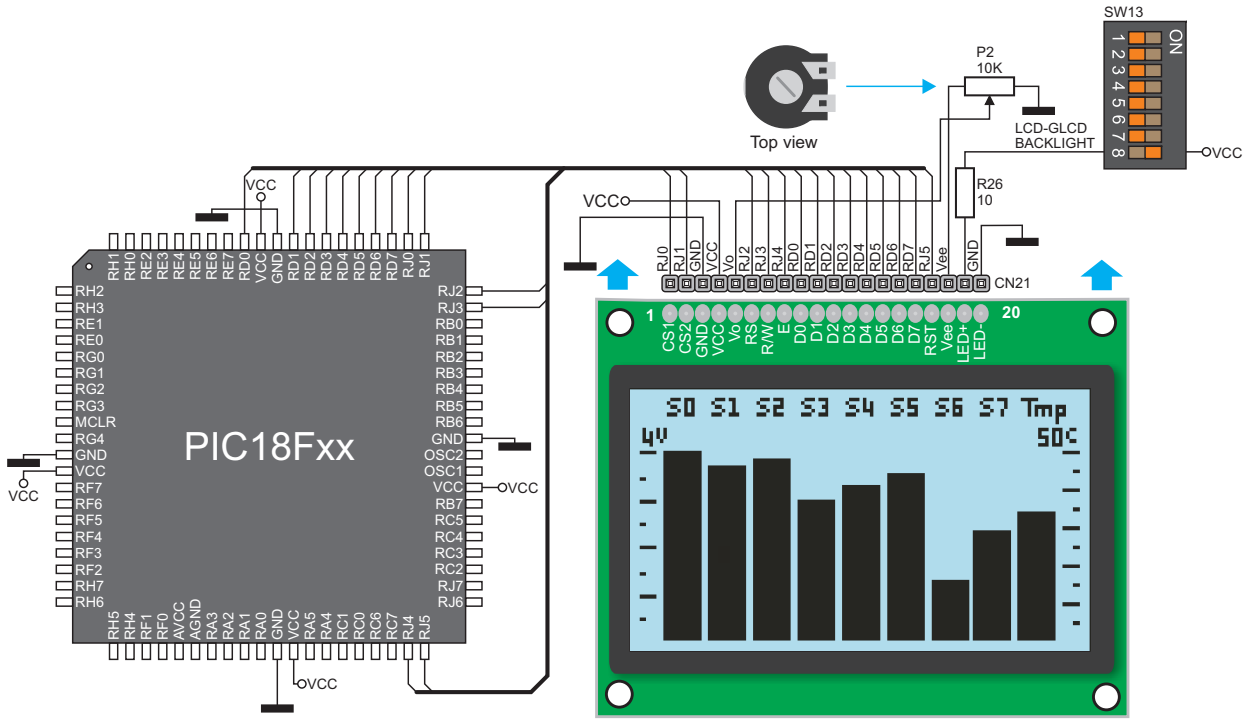


Figure 17-3: GLCD display connection schematic

18.0. Touch Panel

The touch panel is a thin, self-adhesive, transparent panel sensitive to touch. It is placed over a GLCD display. The main purpose of this panel is to register pressure at some specific display point and to forward its coordinates in the form of analog voltage to the microcontroller. Switches 1, 2, 3 and 4 on the DIP switch SW13 are used for connecting touch panel to the microcontroller.

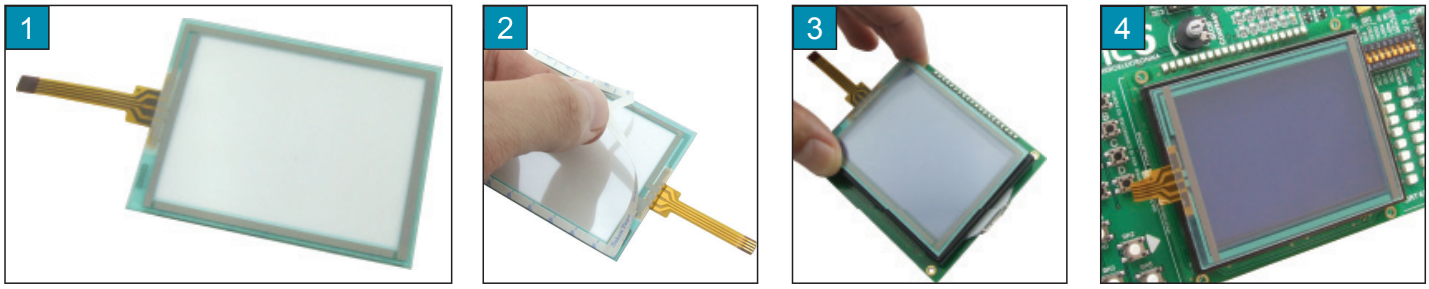


Figure 18-1: Touch panel

Figure 18-1 shows how to place a touch panel over a GLCD display. Make sure that the flat cable is to the left of the GLCD display, as shown in Figure 4.

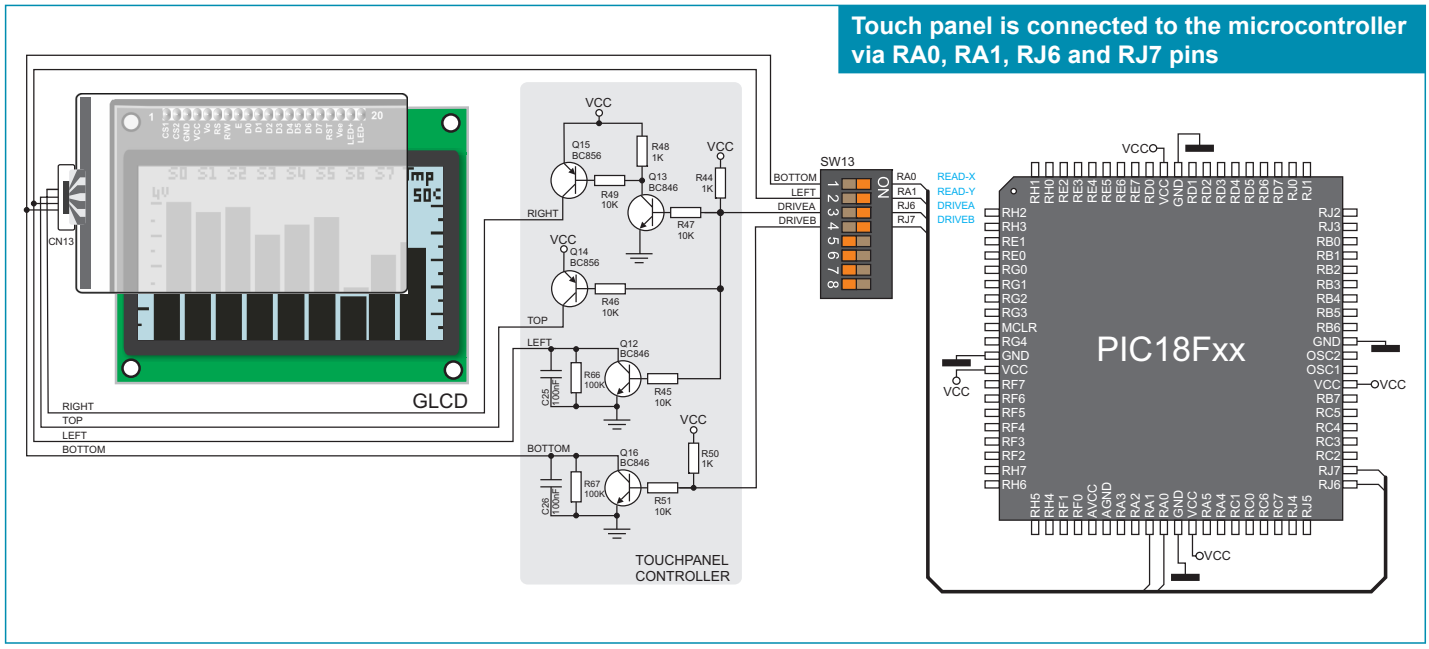


Figure 18-2: Touch panel connection schematic

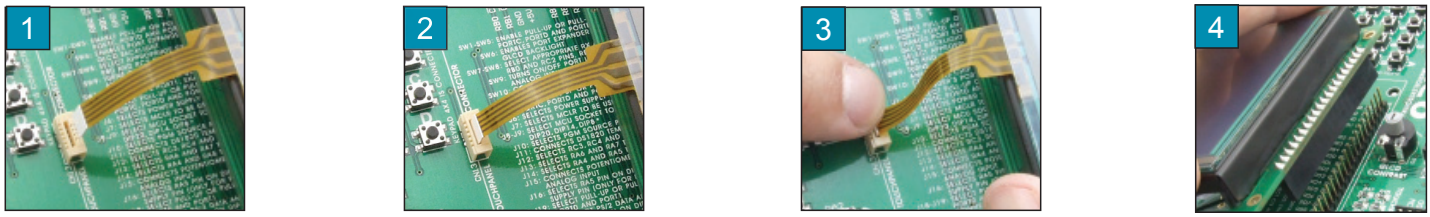


Figure 18-3: Placing touch panel

Figure 18-3 shows in detail how to connect a touch panel to the microcontroller. Bring the end of the flat cable close to the CN13 connector as shown in Figure 1. Plug the cable into the connector, as shown in Figure 2, and press it easily so as to fit the connector, as shown in Figure 3. Now you can plug a GLCD display into the appropriate connector as shown in Figure 4.

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

page

19.0. Input/Output Ports

Along the right side of the development system, there are nine 10-pin connectors which are connected to the microcontroller's I/O ports. Pins RB6 and RB7 are not directly connected to the appropriate 10-pin connector, but via programmer's multiplexer. DIP switches SW1-SW9 enable each connector pin to be connected to one pull-up/pull-down resistor. Whether port pins are to be connected to a pull-up or pull-down resistor depends on the position of jumpers J1-J9.

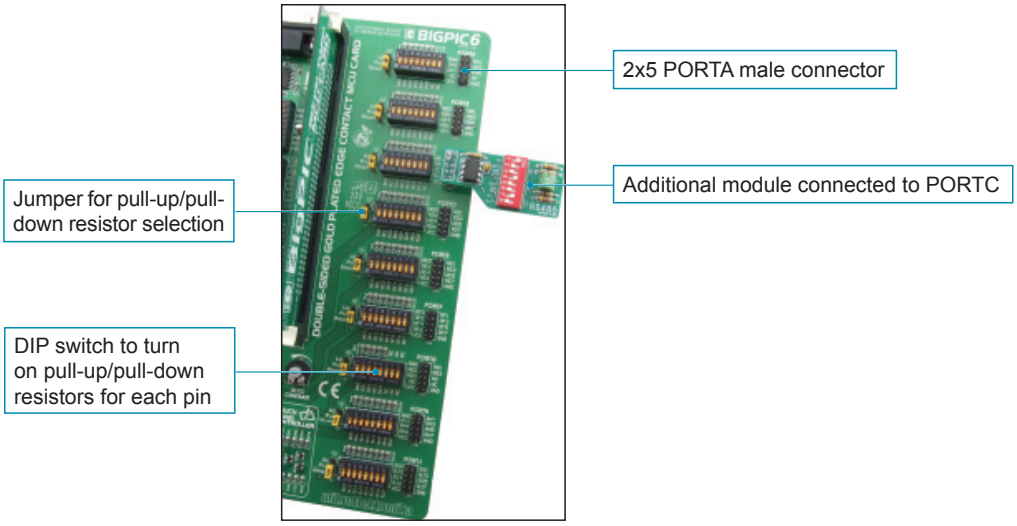


Figure 19-1: I/O ports

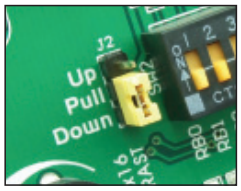


Figure 19-2: J2 in the pull-down positon

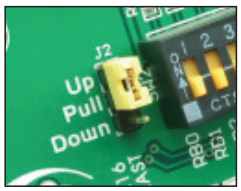


Figure 19-3: J2 in the pull-up position

Microcontroller port PORTB pins connected to pull-down resistors

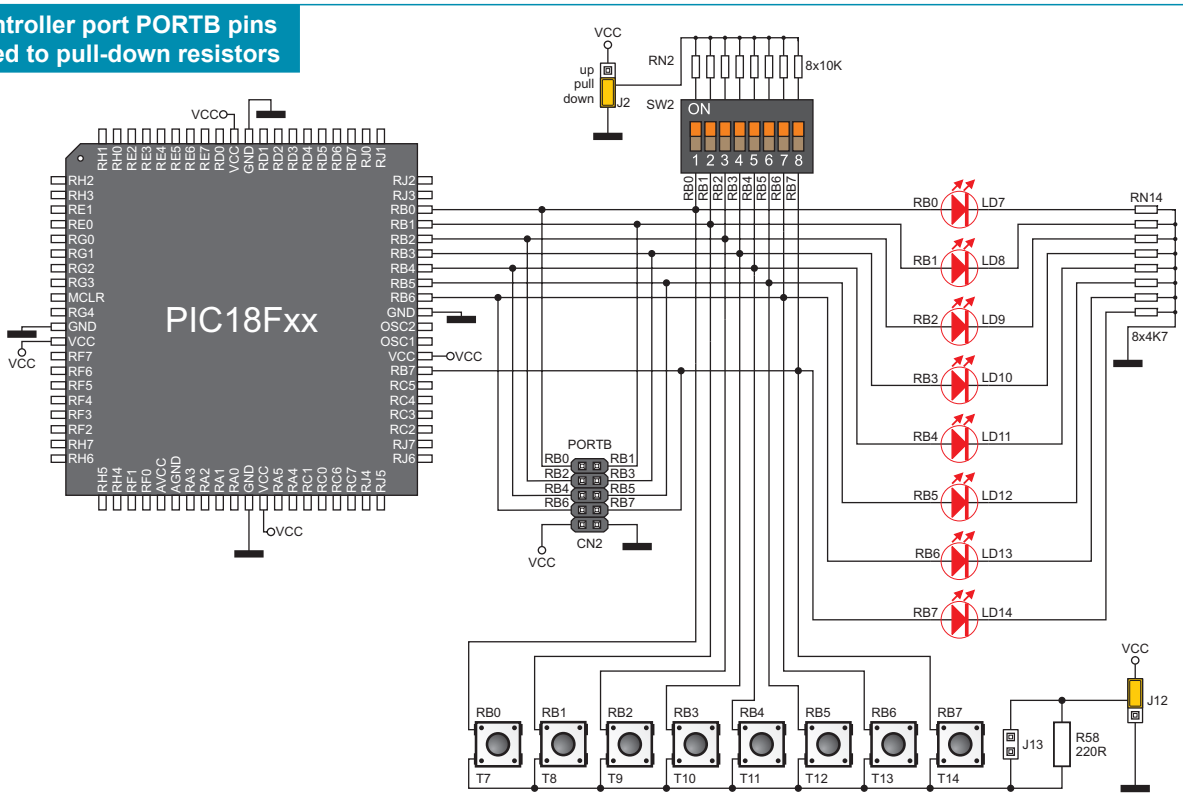


Figure 19-4: Port PORTB connection schematic

Pull-up/pull-down resistors enable you to set the logic level on all microcontroller input pins when they are in idle state. Such level depends on the position of the pull-up/pull-down jumper. The RB0 pin with the relevant DIP switch SW2, jumper J2 and RB0 push button with jumper J12 are used here for the purpose of explaining the performance of pull-up/pull-down resistors.The principle of their operation is identical for all the microcontroller pins.

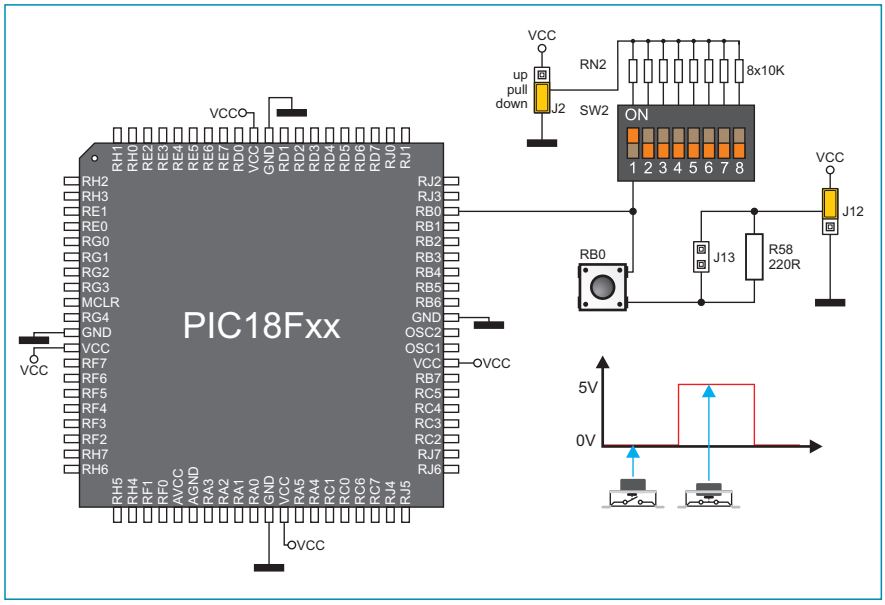


Figure 19-5: JumperJ2 in pull-down and jumper J12 in pull-up position

In order to enable the port PORTB pins to be connected to the pull-down resistors, first it is necessary to set jumper J2 in the *Down* position. This enables any port PORTB pin to be provided 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 RB0 on the DIP switch SW2 in the ON position.

As a result, every time you press the RB0 push button, a logic one (1) will appear on the RB0 pin, provided that jumper J12 is set in the VCC position.

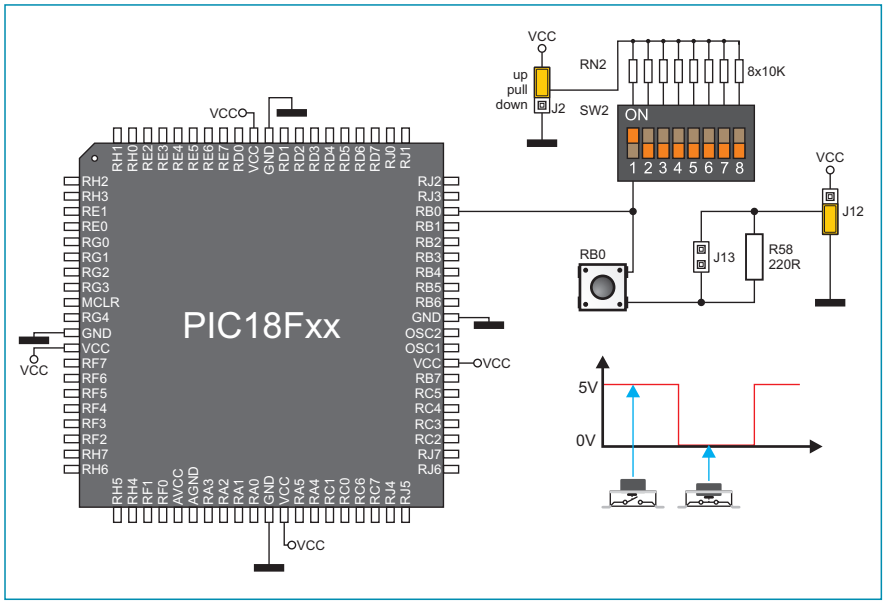


Figure 19-6: Jumper J2 in pull-up and jumper J12 in pull-down position

In order to enable port PORTB pins to be connected to pull-up resistors and port input pins to be activated with logic zero (0), it is necessary to set jumper J2 in the *Up* position (5V) and jumper J12 in the GND position (0V).
This enables any port PORTB input pin to be provided with a logic one (5V) in idle state over the 10k resistor. The RB0 switch should be set in the ON position afterwards.

As a result, every time you press the RB0 push button, a logic zero (0) will appear on the RB0 pin.

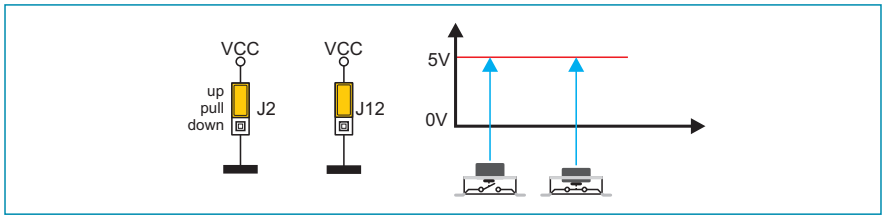


Figure 19-7: Jumpers J2 and J12 in the same position

In case that jumpers J2 and J12 have the same logic state, pressure on any button will not cause input pins to change their logic state.

page

20.0. Port Expander (Additional Input/Output Ports)

The SPI communication lines and MCP23S17 circuit provide the *BigPIC6* development system with a means of increasing the number of available I/O ports by two. If the port expander communicates to the microcontroller over the DIP switch SW11 then the microcontroller pins RE0, RE1, RC5, RC4 and RC3, used for the operation of port expander, cannot be used as I/O pins.

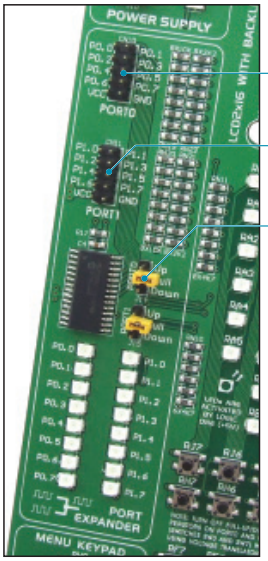


Figure 20-1: Port expander

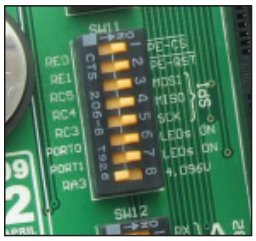


Figure 20-2: Position of DIP switch SW11 when port expander is enabled

The microcontroller communicates to the port expander (MCP23S17 circuit) using serial communication (SPI). The advantage of such communication is that only five lines are used for transmitting and receiving data simultaneously:

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 when the CS pin is driven low (0V). It causes the microcontroller to send clock signal (SCK) and therefore starts data exchange. 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.

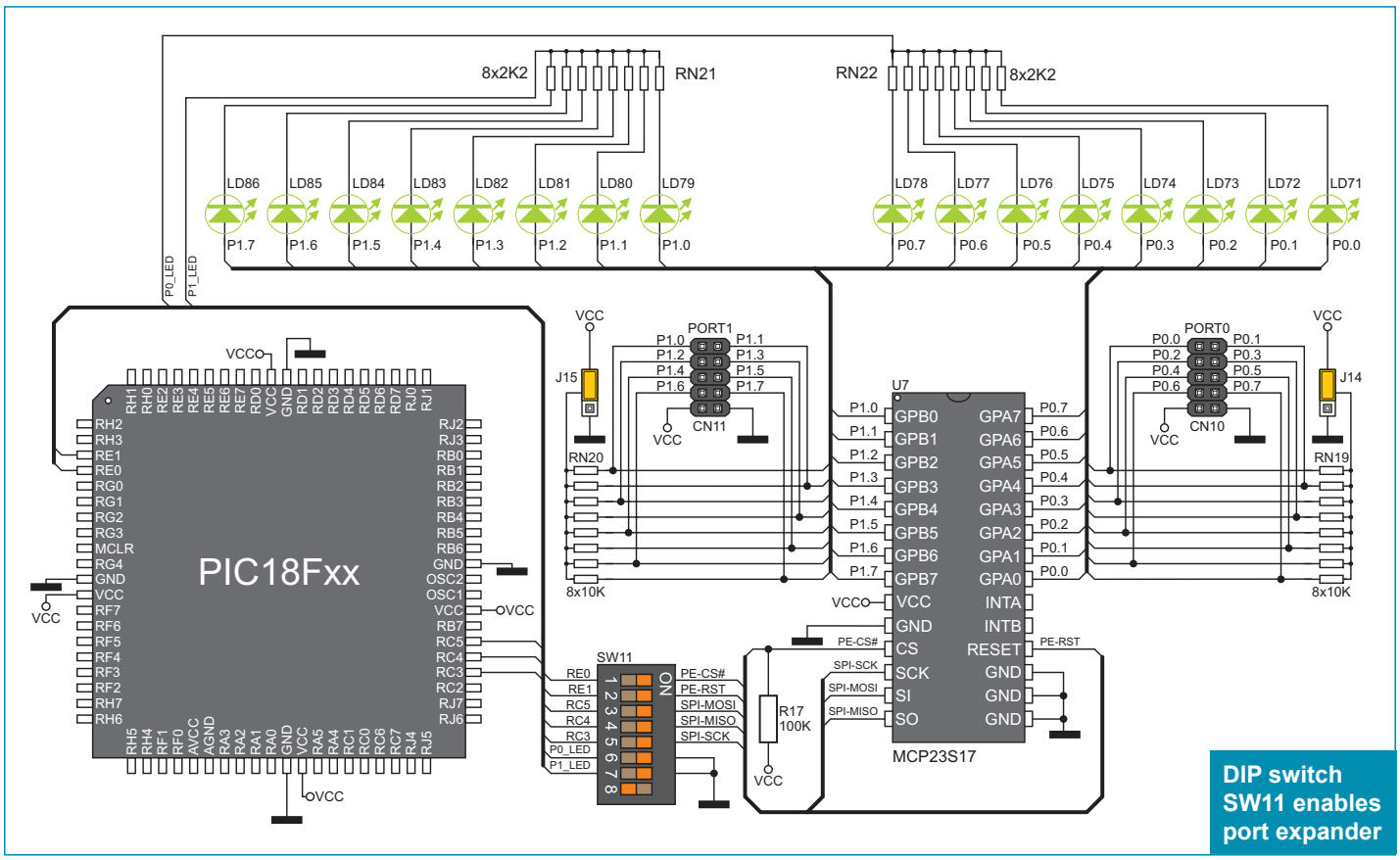


Figure 20-3: Port expander schematic

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.

All the 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.

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.

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