PICPLC16 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.

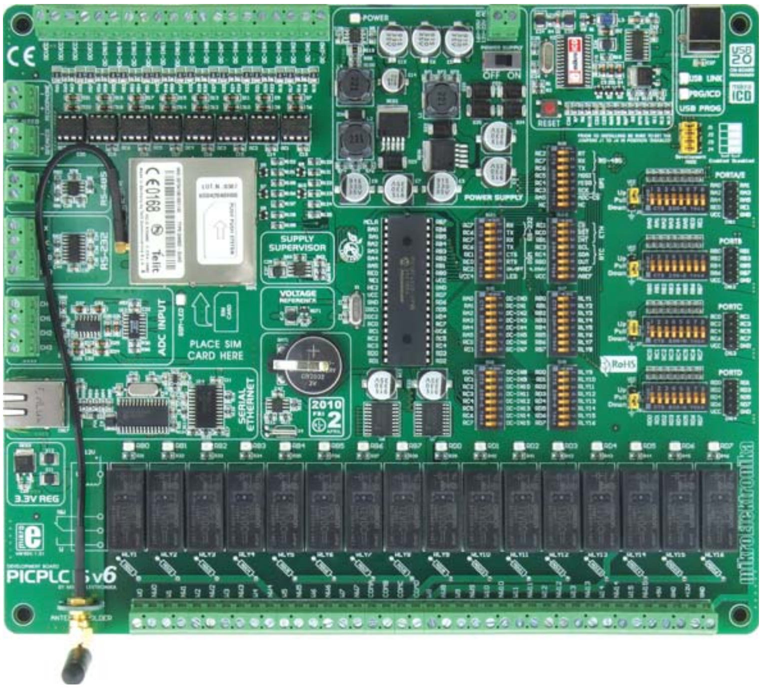
Nebojsa Matic
General Manager

TABLE OF CONTENTS

Introduction to PICPLC16 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 USB 2.0 PICflash with mikroICD Programmer 8
4.0. MikroICD (In-Circuit Debugger) 9
5.0 Power Supply 10
6.0 RS-232 Communication Module 11
7.0. RS-485 Communication Module 12
8.0. Ethernet Module 13
9.0. GSM Connector 14
10.0. A/D Converter Test Inputs 15
11.0. Real-Time Clock (RTC) 16
12.0. Relays and Optocouplers 17
13.0. I/O Ports 18

Introduction to PICPLC16 v6 Development System

The *PICPLC16 v6™* development system provides a development environment for experimenting with industrial devices. Connection between the development system and these devices is established by means of relays. In addition, the *PICPLC16 v6* features additional modules which also enable the microcontroller to be connected to external devices. The *PICPLC16 v6* may be used as a stand-alone controller which communicates to remote devices through communication modules. Numerous modules, such as RS-232 communication module, real-time clock, ethernet controller, GSM module etc. are provided on the board and allow you to easily experiment with your microcontroller.



Development system may be used as a stand-alone controller

Development system for PIC microcontroller based devices

Four inputs for testing A/D coverter in 12-bit resolution

Built-in debugger for testing program being executed in real time at a hardware level

On-board USB 2.0 programmer

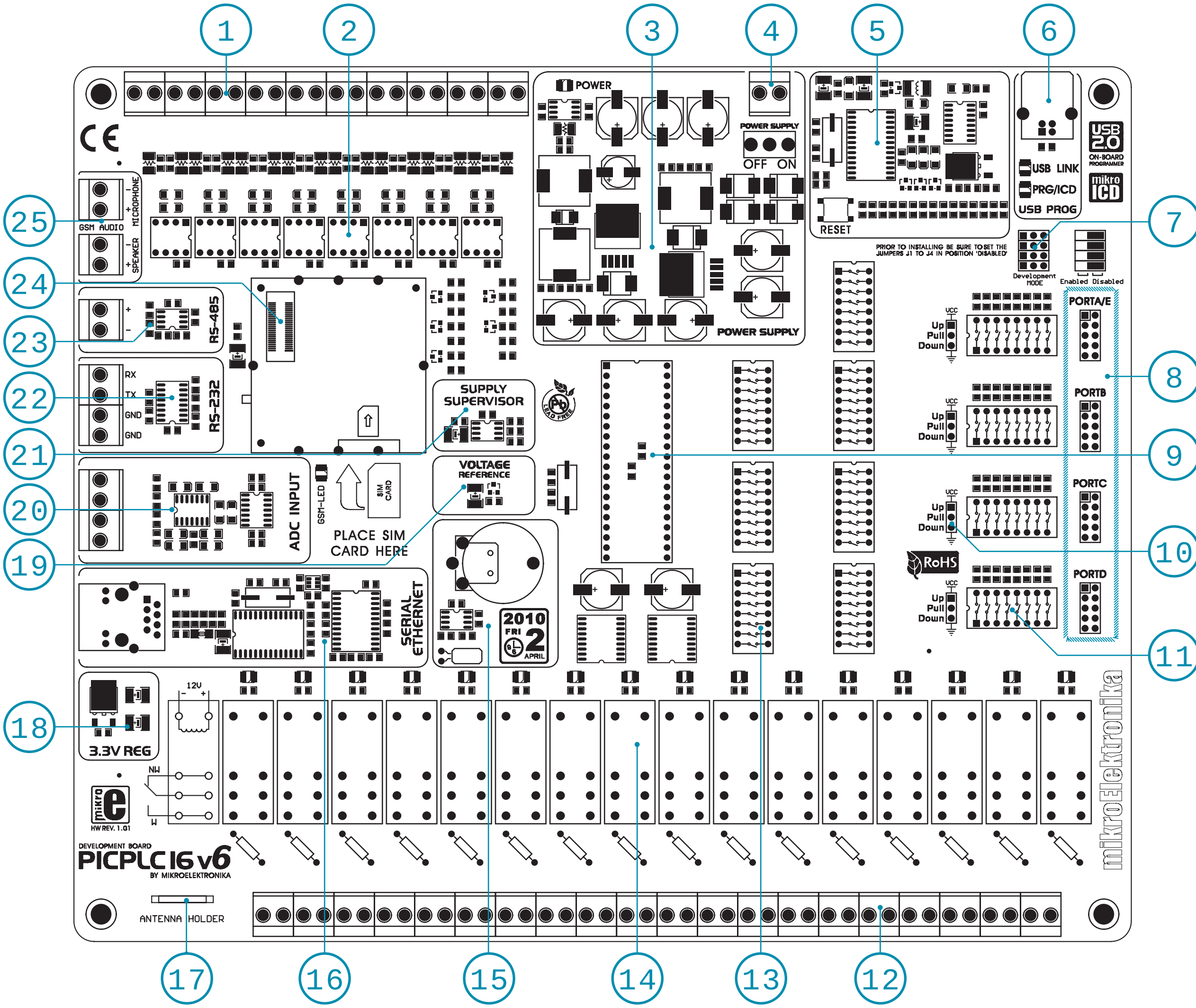
The *PICflash™* program provides a complete list of all supported microcontrollers. The latest version of this program with updated list of supported microcontrollers can be downloaded from our website at **www.mikroe.com**

Package contains:

Development system:	*PICPLC16 v6*
CD:	product CD with relevant software
Cables:	USB cable
Documentation:	Manuals for PICPLC16 v6, PICflash and mikroICD, quick guide for installing USB drivers and electrical schematic of the development system

System specification:

Power supply:	over the CN1 connector (12-22V AC or 16-30V DC)
Power consumption:	120mA when all on-board modules are off
Dimension:	26,5 x 22cm (10,4 x 8,6inch)
Weight:	~750g (1.65lbs)



Key Features

1. Connectors for optocouplers
2. Optocouplers
3. Power supply voltage regulator
4. Power supply connector CN1
5. On-board programmer with mikroICD support
6. On-board programmer’s USB connector
7. Jumpers for isolating the on-board programmer from the development system
8. I/O port connectors
9. Microcontroller socket
10. Jumper for pull-up/pull-down resistor selection
11. DIP switch to enable pull-up/pull-down resistors
12. Connectors to link external devices with relays
13. DIP switches to enable/disable on-board modules
14. Relays
15. Real-time clock
16. Ethernet module
17. Connector for placing GSM antenna
18. 3.3V voltage regulator
19. Voltage reference source
20. A/D converter test inputs
21. Power supply voltage control
22. RS-232 communication module
23. RS-485 communication module
24. Connector for GSM module
25. Connectors for speaker and microphone

1.0. Connecting the System to a PC

Step 1:

Follow the instructions provided in the relevant manuals and install the *PICflash* program and USB drivers 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 PIC compilers installed on your PC, there is no need to reinstall USB drivers as they are already installed along with the compiler.

Step 2:

Prior to connecting the development system to a PC, it is necessary to connect it to the power supply source. Follow the instructions given in figure 1-2 to establish this connection. You need two wires to be placed into the power supply connector and fixed by using screws. Refer to figure 1-2 (2).

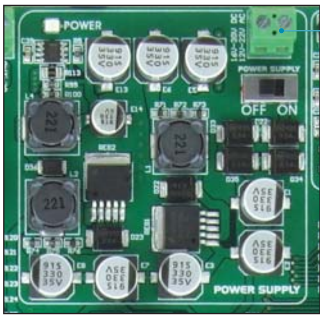


Figure 1-1: Power supply

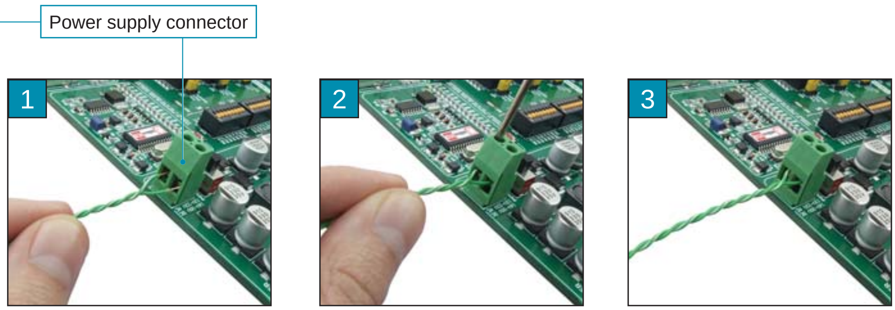


Figure 1-2: Connecting power supply source

Step 3:

When the development system is connected to the power supply source, it is necessary to plug in a USB cable into the on-board USB connector. Connection between the USB cable and the development system makes the on-board programmer to be connected to a PC. Now it is possible to load a hex code from the PC into the microcontroller.

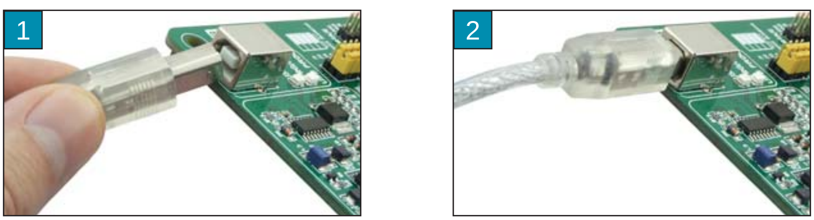


Figure 1-3: Plugging in a USB cable into the development system

Step 4:

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 automatically turned on indicating that your development system is ready to use. Use the on-board programmer and the *PICflash* program to dump a code into the microcontroller and employ the system to test and develop your projects.

NOTE: Make sure the power supply source is connected. Otherwise, the on-board programmer cannot be enabled.

2.0. Supported Microcontrollers

The *PICPLC16 v6* development system comes with the PIC18F4520 microcontroller in DIP40 package. In case this microcontroller doesn't suit your needs, it is possible to replace it with another one. When choosing the appropriate replacement for the existing microcontroller, the most important thing to pay attention to is the pinout.

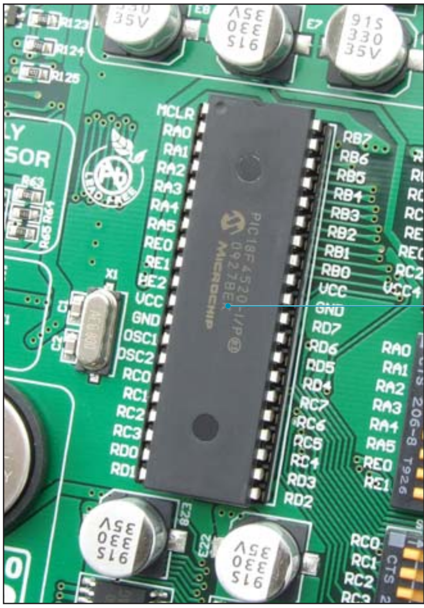


Figure 2-1: Microcontroller in DIP40 package

The existing microcontroller can be replaced with another one in DIP40 package

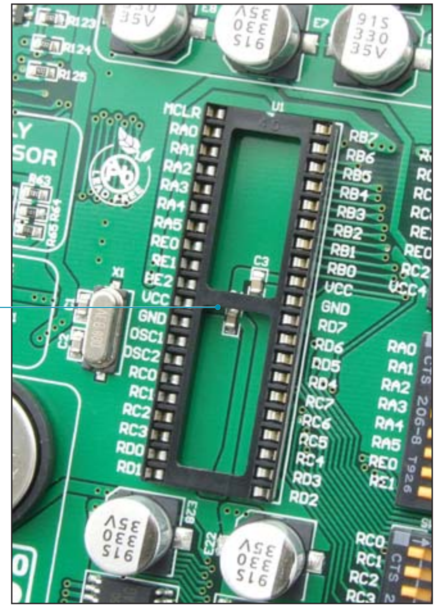


Figure 2-2: DIP40 socket

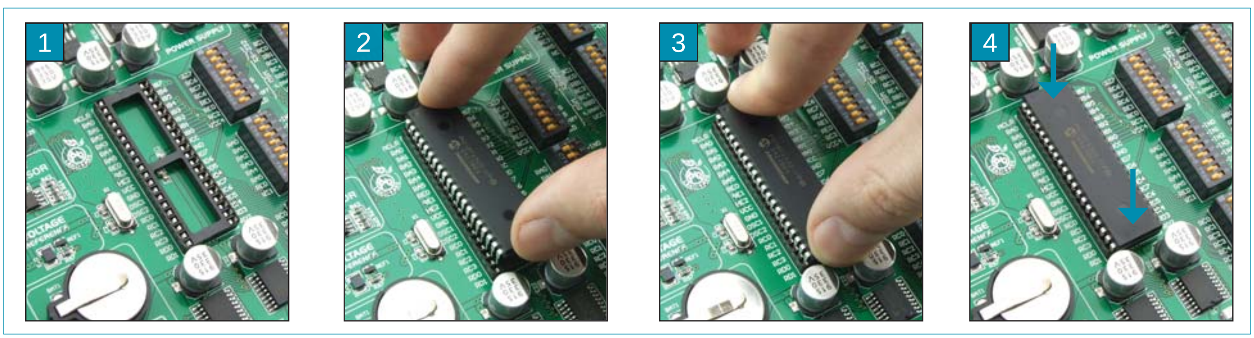


Figure 2-3: Placing microcontroller into the socket

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

3.0. On-board USB 2.0 PICflash Programmer with mikroICD Support

A programmer is a necessary tool when working with microcontrollers. The *PICPLC16 v6* has an on-board *PICflash* with *mikroICD* programmer which provides an interface between the microcontroller and the PC. The *PICflash* program is used for loading a .hex file into the microcontroller. Figure 3-2 shows connection between the compiler, *PICflash* program and the microcontroller.

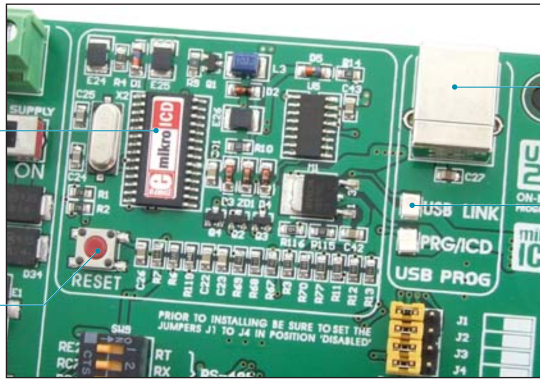
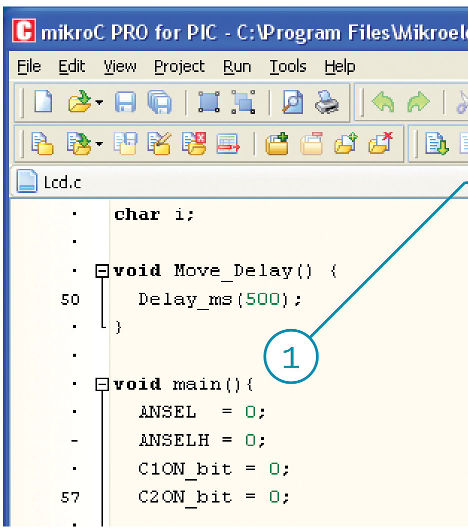


Figure 3-1: *PICflash* programmer



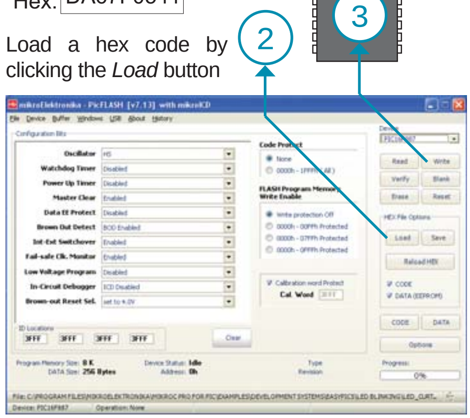
Write a code in one of the PIC compilers, generate a .hex file, and employ the on-board programmer to load the code into the microcontroller.

Compiling program

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

MCU

Load a hex code by clicking the *Load* button



① Write a program in one of the PIC compilers and generate a .hex file;

② Use the *PICflash* program to select desired microcontroller to be programmed;

③ Click the *Write* button to dump the code into the microcontroller.

On the left side of the *PICflash* program's main window, there is a number of options for setting the operation of the microcontroler 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 3-2: The process of programming

4.0. mikroICD (In-Circuit Debugger)

The mikroICD (In-Circuit Debugger) is an integral part of the on-board programmer. It is used for the purpose of 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.

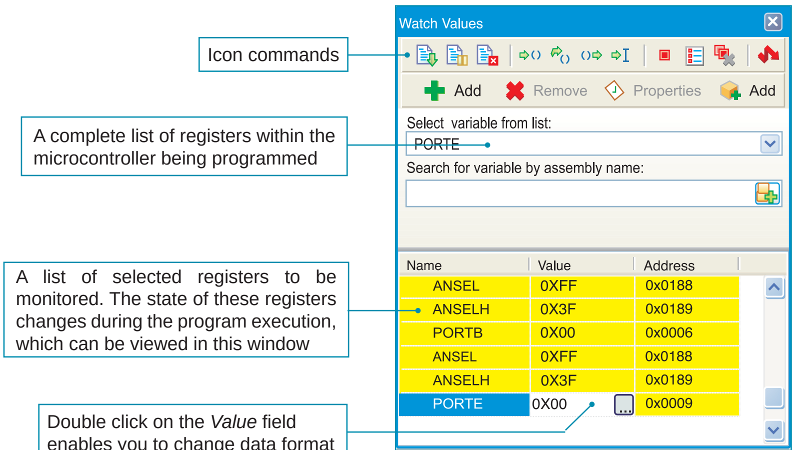


Figure 4-1: *Watch Values* window

mikroICD debugger options:

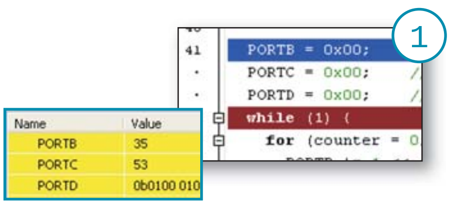
Start Debugger	[F9]
Run/Pause Debugger	[F6]
Stop Debugger	[Ctrl+F2]
Step Into	[F7]
Step Over	[F8]
Step Out	[Ctrl+F8]
Toggle Breakpoint	[F5]
Show/Hide Breakpoints	[Shift+F4]
Clear Breakpoints	[Ctrl+Shift+F5]

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:

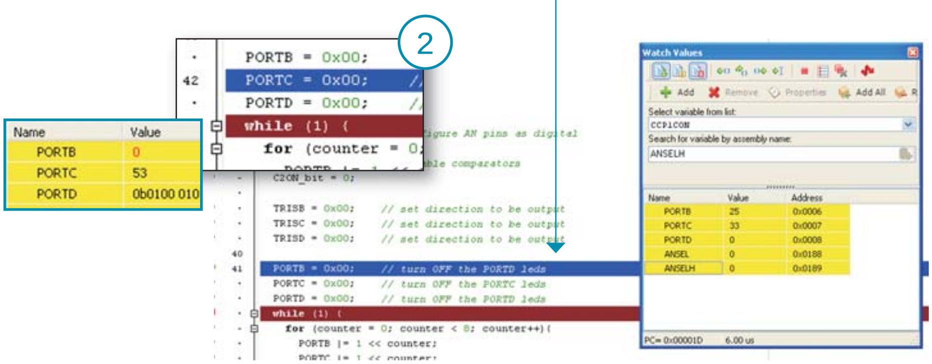
In this example the 41st program line is highlighted in blue, which means that it will be executed next. The current state of all registers within the microcontroller can be viewed in the mikroICD *Watch Values* window.



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.

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.



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

5.0. Power Supply

The *PICPLC16 v6* development system is connected to the power supply source via the CN1 connector. The power supply voltage can be either DC or AC. A DC power supply voltage can be in the range of 16V to 30V, whereas the AC power supply voltage can range between 12V and 22V. Have in mind that the on-board programmer cannot operate without being connected to the power supply source although it is connected to a PC via the USB cable.

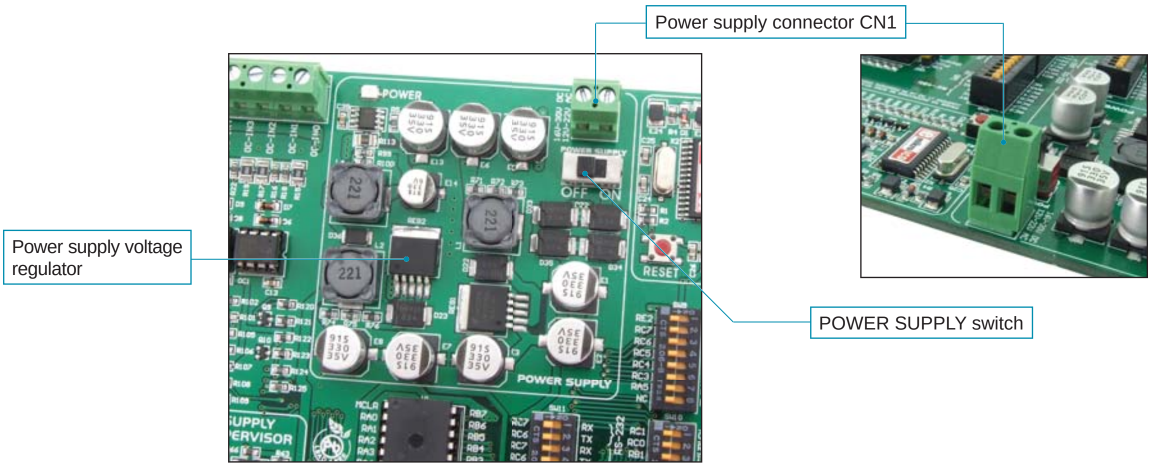


Figure 5-1: Power supply

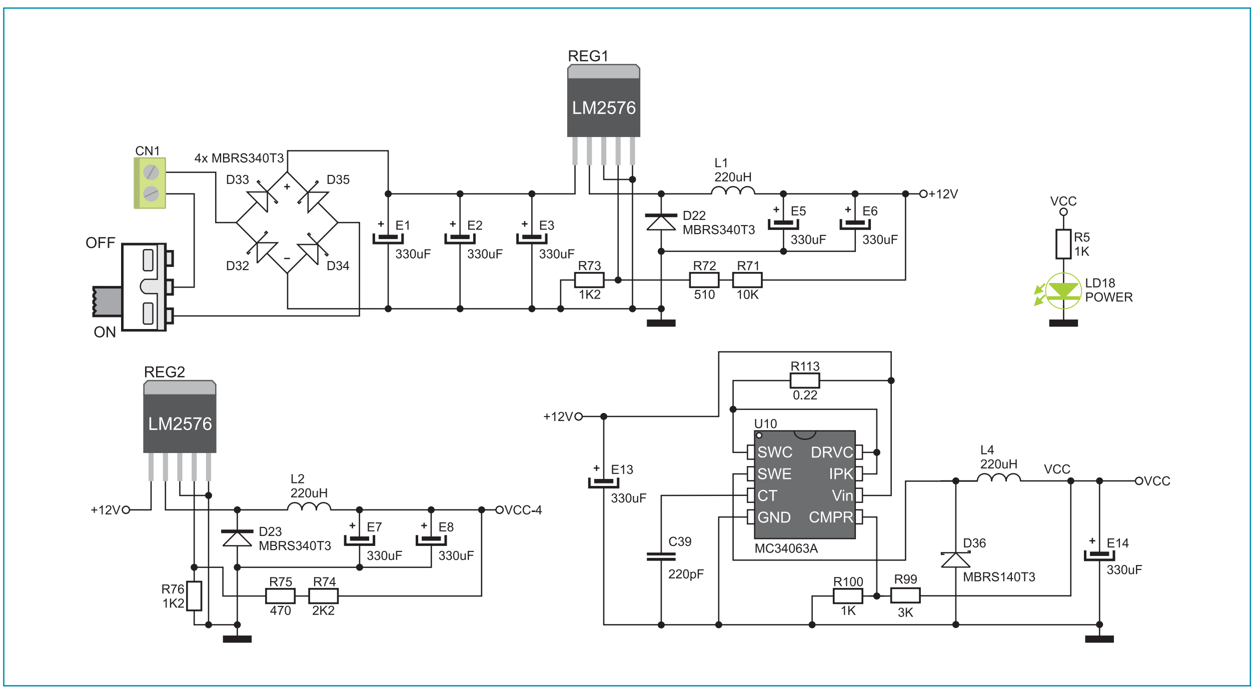


Figure 5-2: Power supply connection schematic

6.0. RS-232 Communication Module

USART (*Universal Synchronous/Asynchronous Receiver/Transmitter*) is one of the most common ways of exchanging data between the PC and peripheral units. The RS-232 serial communication is performed through CN4 and CN5 connectors and the microcontroller USART module. There is one RS-232 port provided on the *PICPLC16 v6*. Use switches marked as RX232 and TX232 on the DIP switch SW11 to enable this port. The microcontroller pins used for the RS-232 communication are marked as follows: RX - *receive data line* and TX - *transmit data line*. Data rate goes up to 115 kbps.

In order to enable the USART module of the microcontroller to receive input signals which meet the RS-232 standard, it is necessary to adjust voltage levels using an IC circuit such as (MAX232).

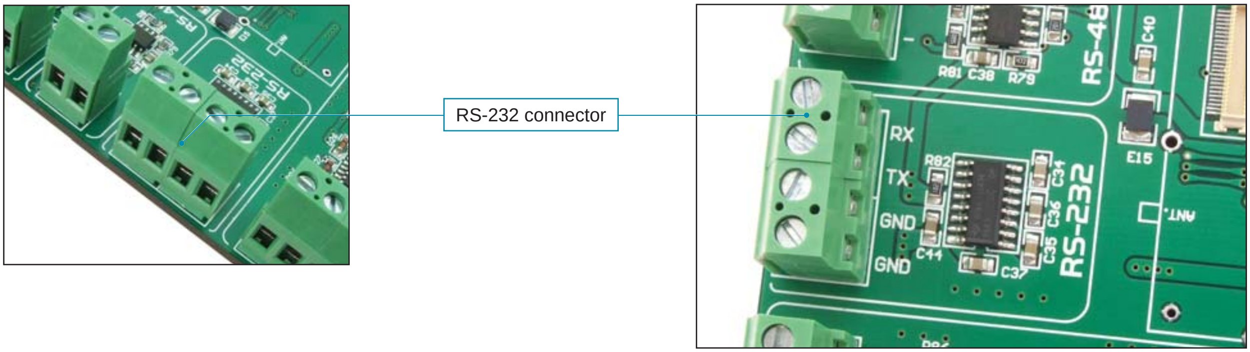


Figure 6-1: RS-232 module

Port RS-232 is connected to the microcontroller

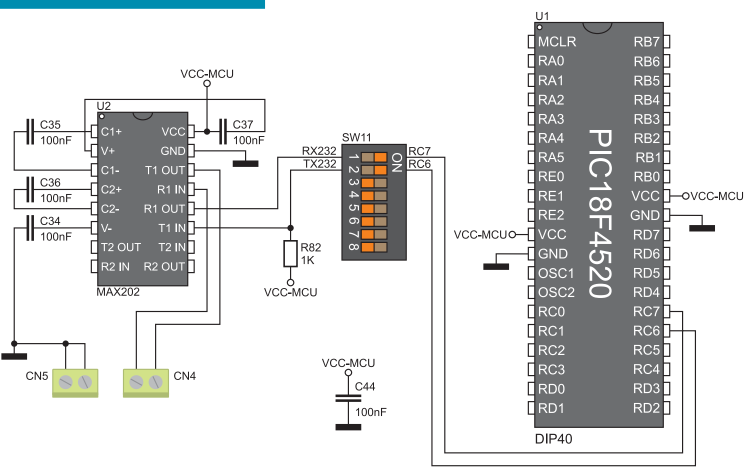


Figure 6-2: RS-232 module connection schematic

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

7.0. RS-485 Communication Module

The RS-485 communication is a communication standard primarily intended for the use in industrial applications. The main features of this communication standard is the ability to exchange data between distant points (up to 1200 m) and high tolerance to accompanying noise. The *PICPLC16 v6* development system features a connector which enables devices using RS-485 communication to be linked. The ADM485 circuit acts as a transciever between an external device and the microcontroller. To enable connection between the microcontroller and the RS-485 communication module, it is necessary to set switches 1, 2 and 3 on the DIP switch SW9 to the ON position.

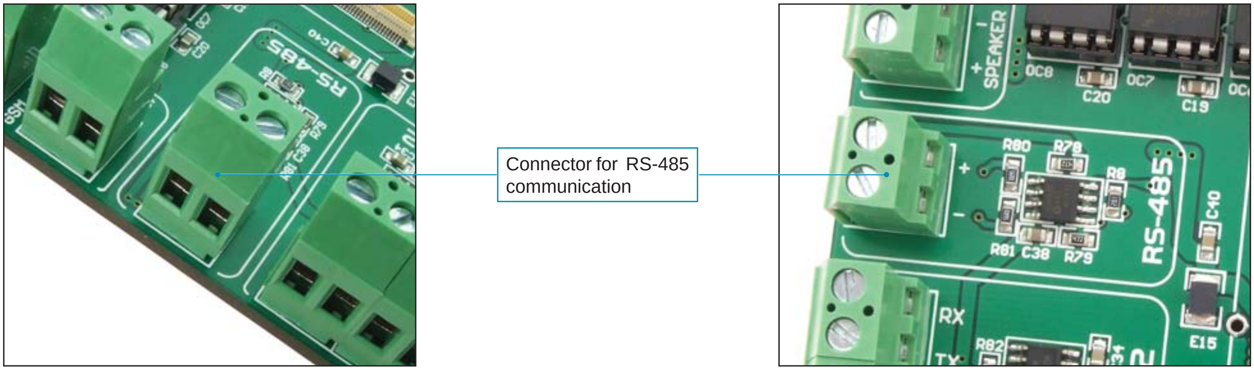


Figure 7-1: RS-485 module

RS-485 communication is enabled via DIP switch SW9

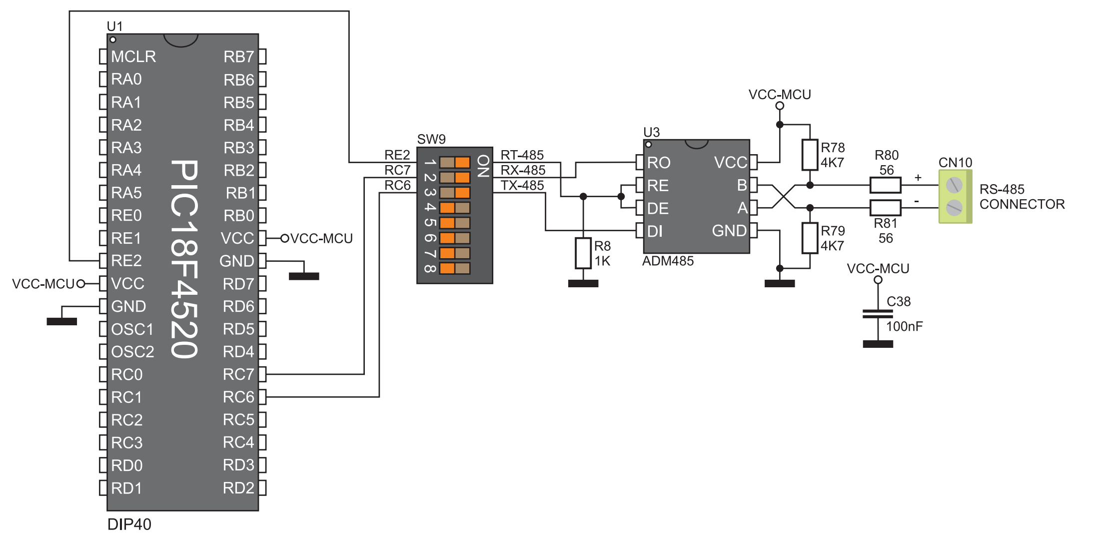


Figure 7-2: RS-485 module connection schematic

8.0. Ethernet Module

The *PICPLC16 v6* development system features an ethernet module providing an interface between the microcontroller and LAN (local area network). The ENC28J60 stand-alone controller enables ethernet communication on the development system. It is used to transfer data from LAN to the microcontroller using serial communication. The 3.3V voltage is required for the operation of this controller. To enable data to be transferred to the microcontroller powered with the 5V power supply voltage, it is necessary to adjust these voltage levels by means of transceivers 74LVCC3245 and 74LVC1T45. To enable connection between the ethernet module and the microcontroller, switches 1, 2 and 3 on the DIP switch SW10, as well as switches 4, 5 and 6 on the DIP switch SW9 should be set to the ON position.

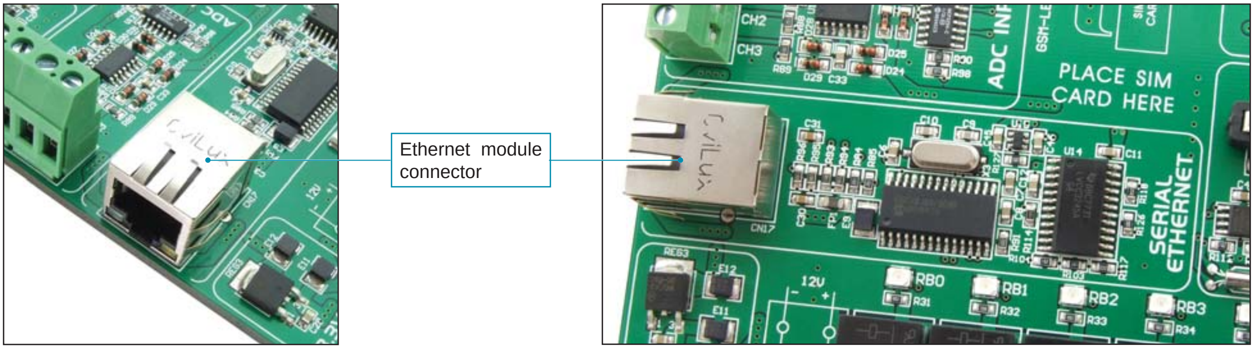


Figure 8-1: Ethernet module

Ethernet module is connected to the microcontroller via DIP switches SW9 and SW10

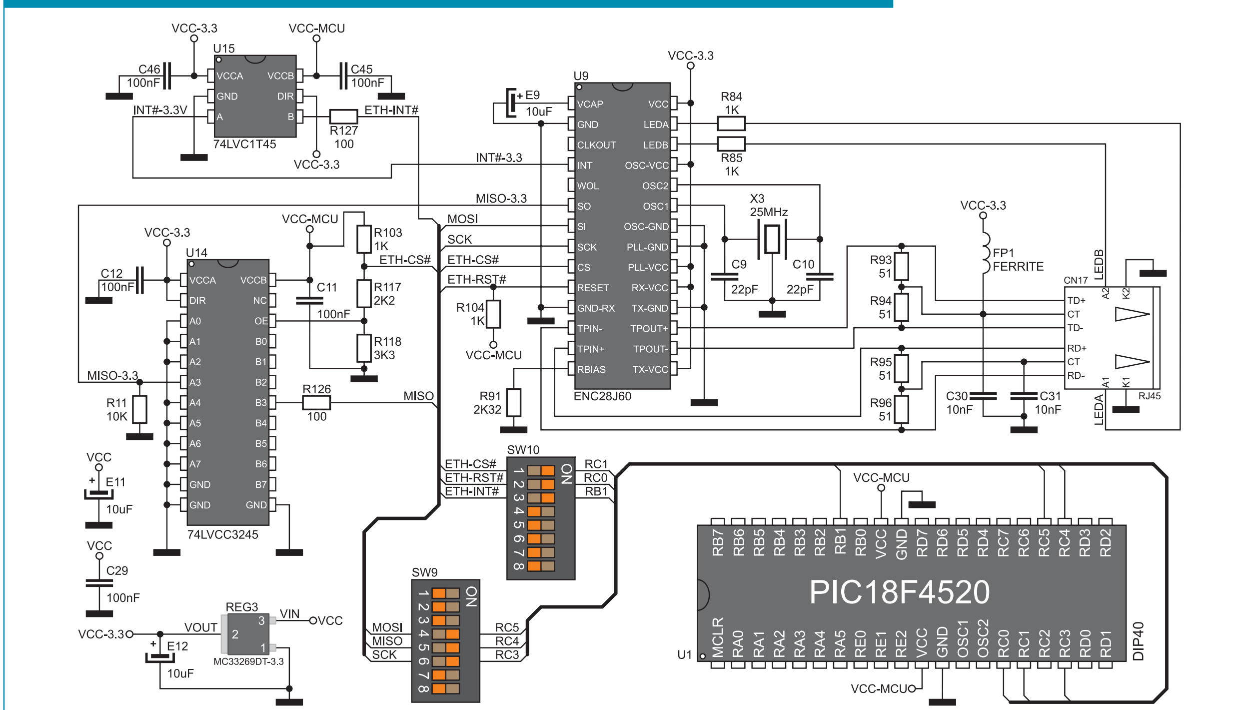


Figure 8-2: Ethernet module connection schematic

9.0. GSM Connector

Owing to a built-in connector for GSM module, the *PICPLC16 v6* development system is capable of communicating with the outside world using GSM network. A GM862-QUAD GSM module from Telit can be ordered with the development system. This module features a slot for placing a SIM card as well as a connector for external antenna. For the GSM module to be connected to the microcontroller, it is necessary to set switches 3-8 on the DIP switch SW11 to the ON position.

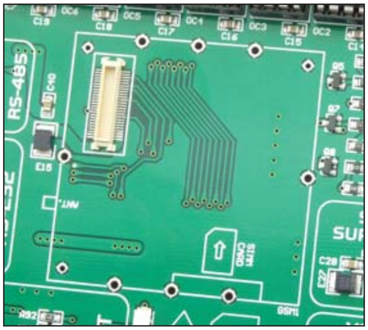


Figure 9-1: GSM connector

Figure 9-2: GSM module

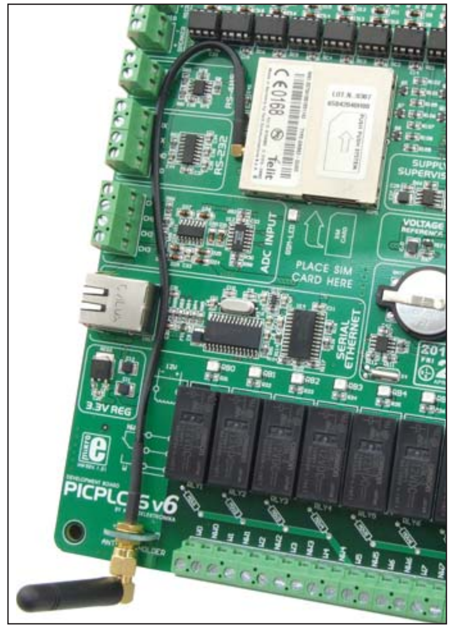


Figure 9-4: GSM module with antenna

In case that the GSM module is employed for the audio communication, it is necessary to plug in a speaker and a microphone into appropriate connectors, as shown in Figure 9-3. In addition to the audio signal transmission, the GSM module can be used for sending data in accordance with the GPRS standard used in mobile applications.

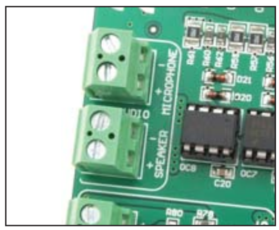


Figure 9-3: Audio connectors

GSM module is connected to the microcontroller via DIP switch SW11

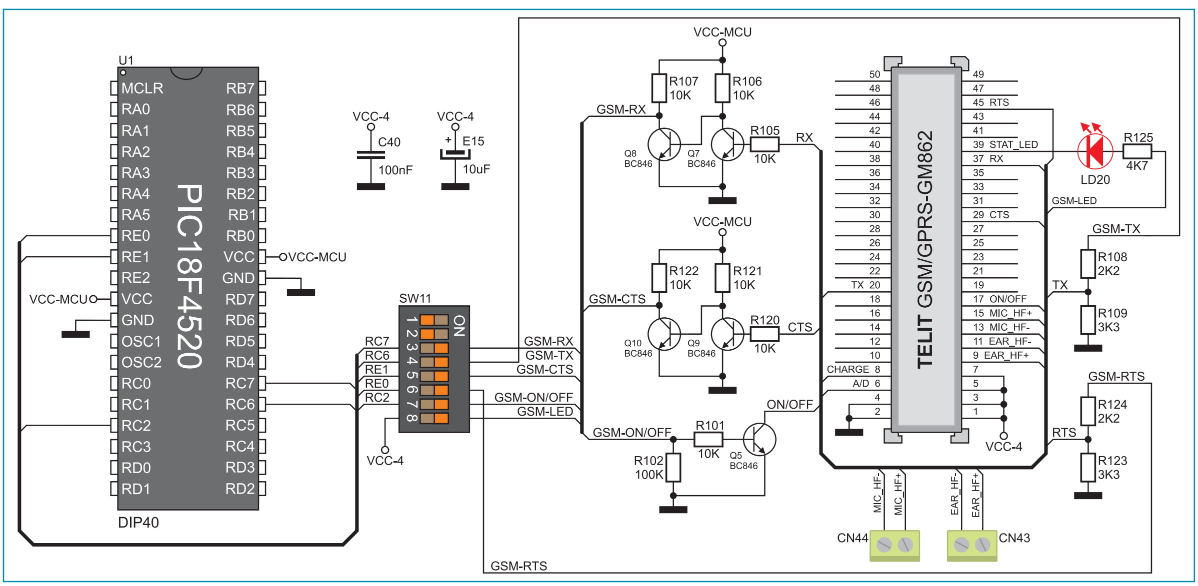


Figure 9-5: Microcontroller and GSM connector connection schematic

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 converted number is linearly dependent on the input voltage value. The MCP3204 circuit is used as an A/D converter on the *PICPLC16 v6* development system. Voltage to be converted into a 12-bit number is brought to the A/D converter input pins by means of the MCP6284 operational amplifier. The result of conversion is transferred to the microcontroller by means of serial communication. For the digital signal to be transferred to the microcontroller, it is necessary to set switches 4, 5, 6 and 7 on the DIP switch SW9 to the ON position. In case that voltage reference is used during A/D conversion, it is necessary to select desired voltage reference by means of DIP switch SW10. If the power supply voltage is used as a voltage reference, then switch 8 on the DIP switch SW10 should be set to the ON position. If the 4.096V voltage is used as a voltage reference, the switch 7 on the DIP switch SW10 should be set to the ON position.

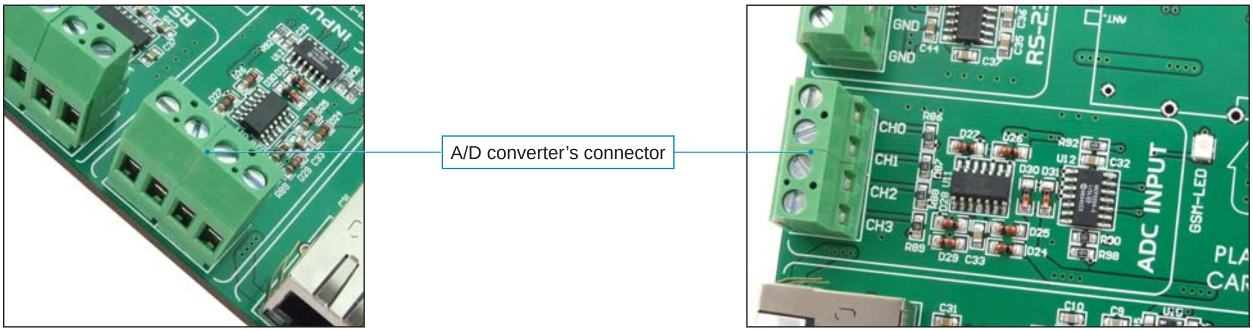


Figure 10-1: A/D module

A/D converter communicates to the microcontroller using SPI communication

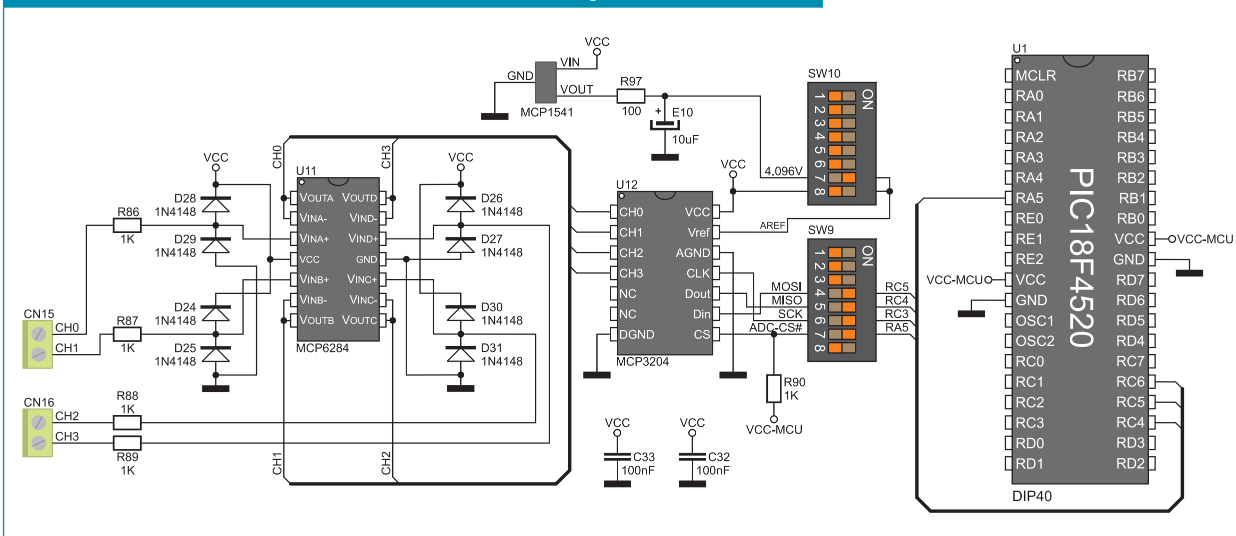


Figure 10-2: AD module and microcontroller connection schematic

11.0. Real-Time Clock (RTC)

As a result of the built-in DS1307 circuit, the *PICPLC16 v6* development system is capable of keeping the real time. The main features of the real-time clock are as follows:

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

A real-time clock is widely used in alarm devices, industrial controllers, consumer devices etc. The real-time clock provided on the *PICPLC16 v6* development system is used to generate an interrupt at pre-set time. In order to establish connection between the microcontroller and the real-time clock it is necessary to set switches 4, 5 and 6 on the DIP switch SW10 to the ON position.

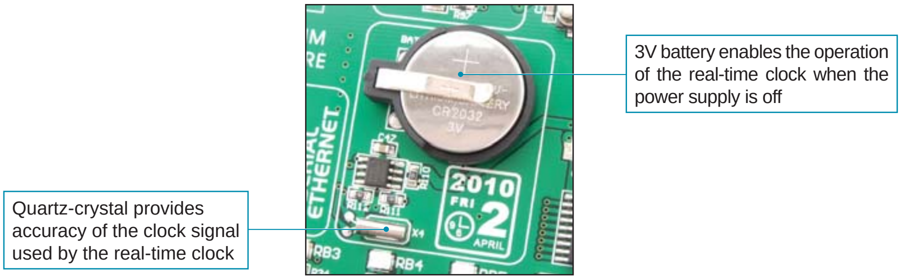


Figure11-1: Real-time clock

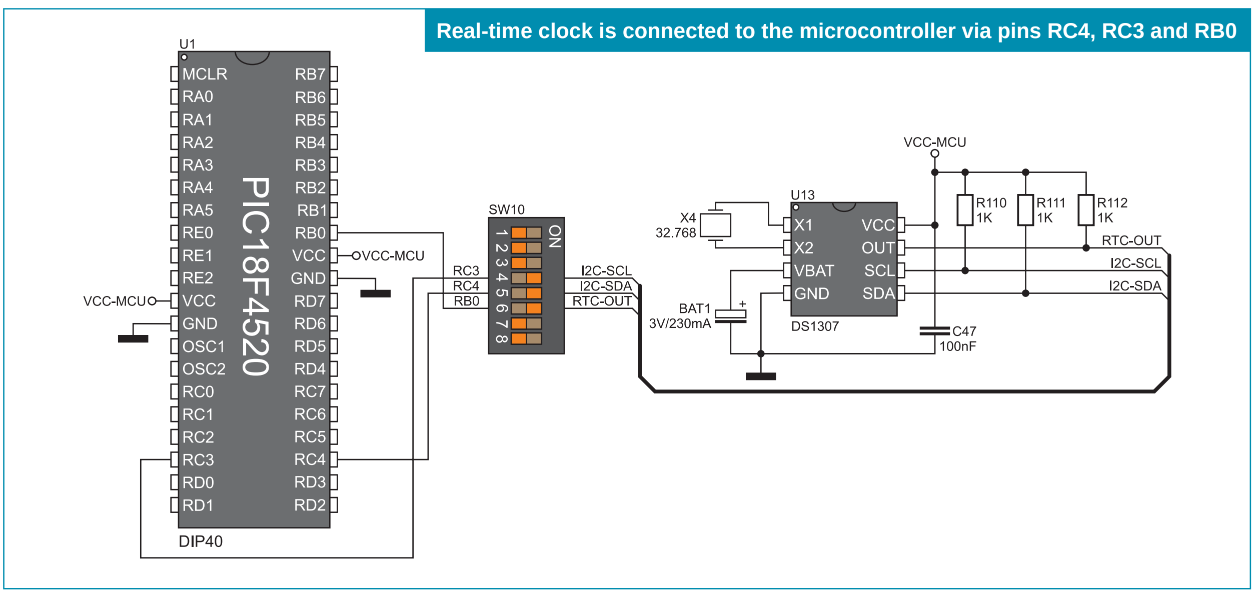


Figure 11-2: Real-time clock and microcontroller connection schematic

12.0. Relays and Optocouplers

Industrial devices usually utilize more power than the microcontroller can provide via its I/O ports. To enable the microcontroller to be connected to such devices, the development system is provided with 16 relays by means of which it is possible to provide up to 250V power supply. Each relay has one normally-open (W0, W1...) and one normally-closed (NW0, NW1...) contact. Sixteen relays are divided in four groups each consisting of four relays. Relays of one group are connected to one common contact. Accordingly, there are COMA, COMB, COMC and COMD common contacts. Figure 12-3 illustrates the connection between one group of relays and the relevant COMA common contact. In addition to relays, the development system also features optocouplers the function of which is to galvanically isolate signals brought to the microcontroller inputs from industrial devices. As can be seen in Figure 12-3, optocouplers are also linked to one common connector OCVCC.

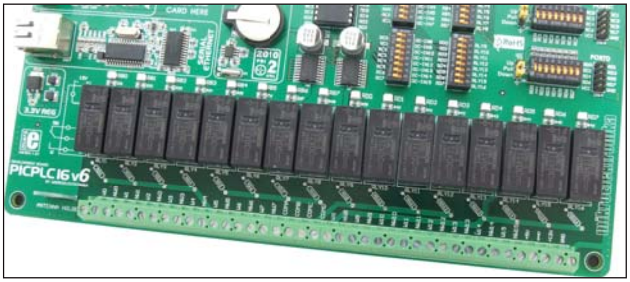


Figure 12-1: Relays with relevant connectors

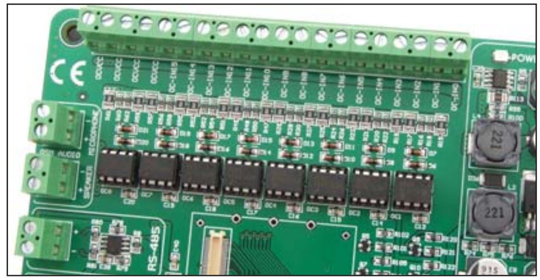


Figure 12-2: Optocouplers with relevant connectors

Optocouplers galvanically isolate microcontroller inputs from high voltage

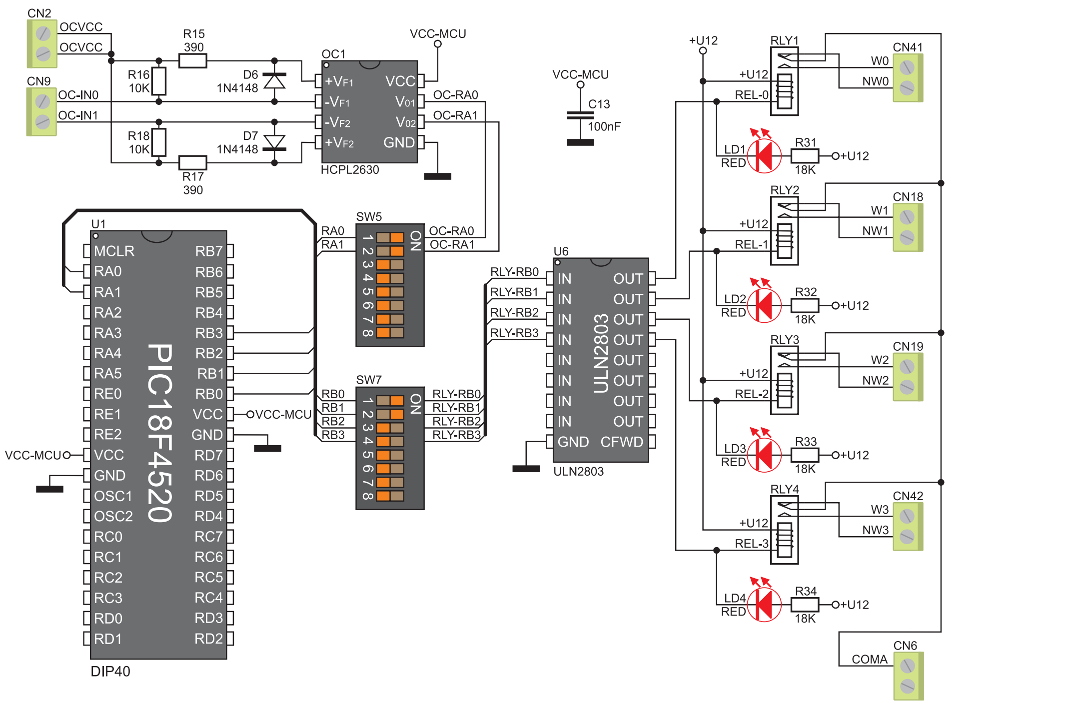


Figure 12-3: Relays and optocouplers and microcontroller connection schematic

13.0. Input/Output Ports

Along the right side of the development system, there are four 10-pin connectors which are linked to the microcontroller's I/O ports. Microcontroller pins used for programming are not directly connected to the appropriate 10-pin connector CN2 (PORTB), but via a multiplexer. DIP switches SW1-SW4 enable each connector pin to be connected to one pull-up/pull-down resistor. It depends on the position of jumpers J5-J8 whether the port pins are to be connected to pull-up or pull-down resistors.

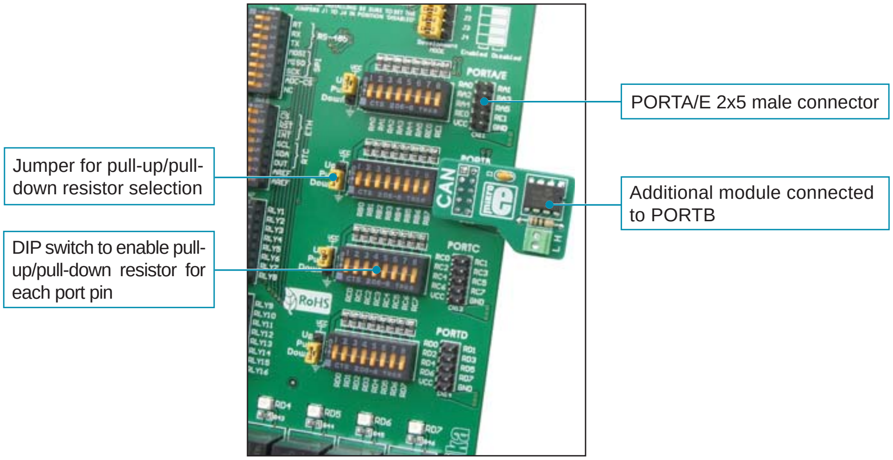


Figure 13-1: I/O ports

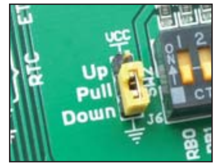


Figure 13-2: J6 in pull-down position

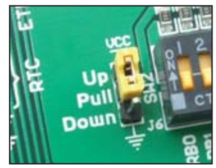


Figure 13-3: J6 in pull-up position

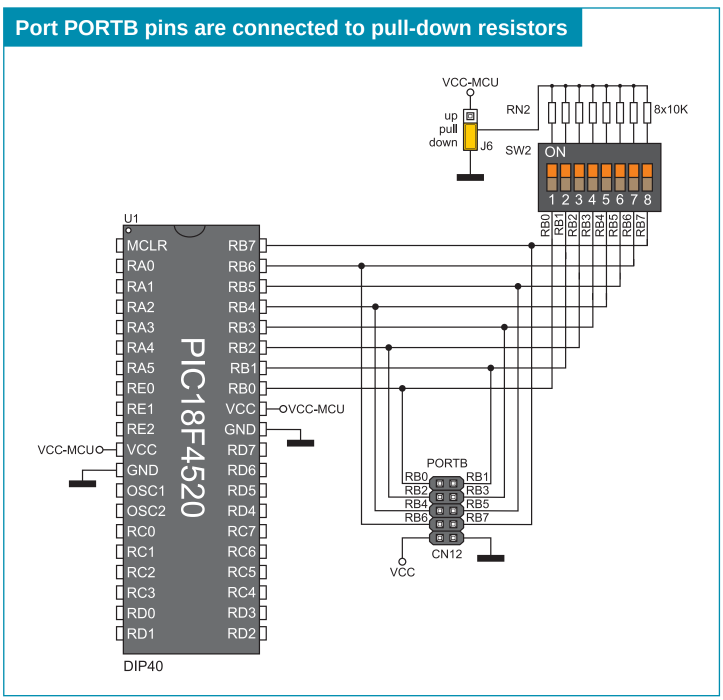


Figure 13-4: Port PORTB connection schematic

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 jumper J6. When this jumper is in pull-up position, the input pins will be supplied with the 5V power supply voltage, which means that they will be driven high (logic one (1)). When this jumper is in pull-down position, the input pins will be supplied with 0V, i.e. they will be fed with a logic zero (0).

In order to provide some of the microcontroller pins with a desired logic level, it is necessary to enable connection between that pin and the resistor using the appropriate DIP switch.

Refer to figure 13-4. Port PORTB pins are driven low (0). It means that jumper J6 is in pull-down position, whereas switches on the DIP switch SW2 are in the ON position.

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, AVRprog, 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.

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