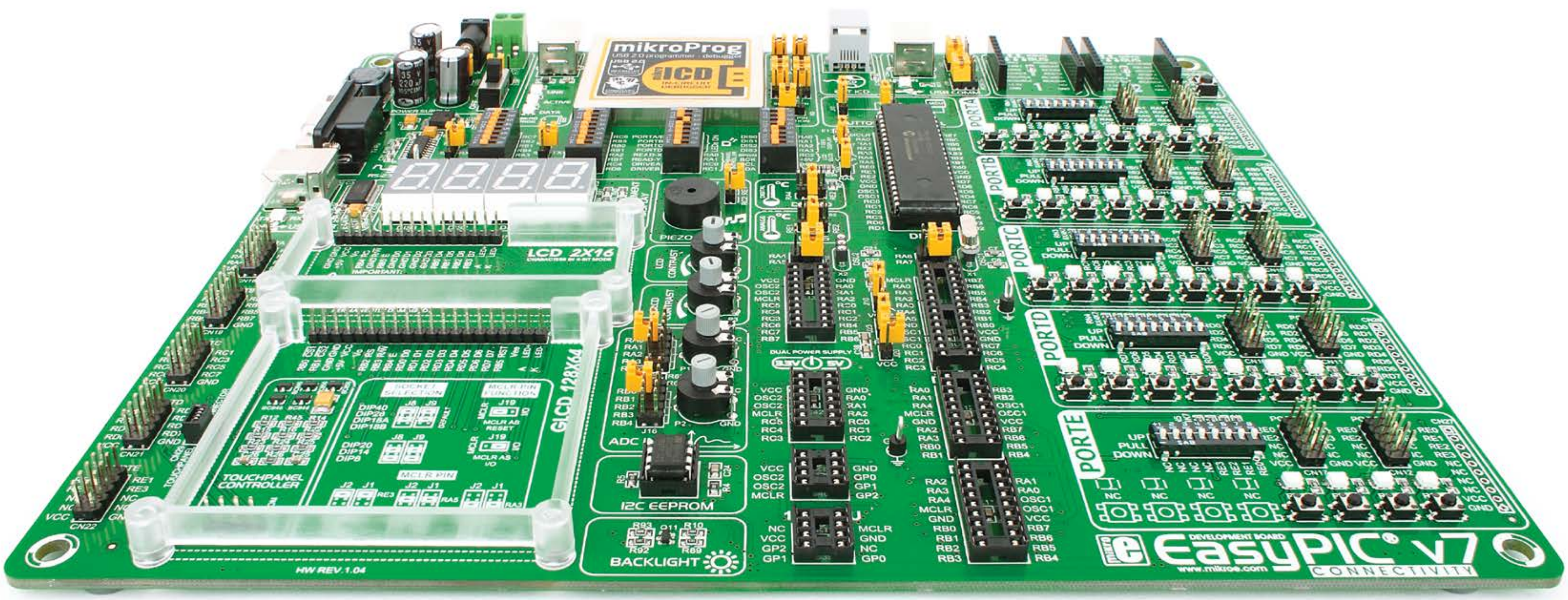
EasyPIC™ v7 connectivity

USER'S GUIDE



350

microcontrollers supported

The ultimate PIC® board

DUAL POWER SUPPLY 3.3V 5V

Supports 3.3V and 5V devices

Dual Power Supply

mikro BUS

Easy-add extra boards

mikroBUS™ sockets

connectivity

Four connectors for each port

Amazing Connectivity

Fast USB 2.0 programmer and

In-Circuit Debugger

MikroElektronika

DEVELOPMENT TOOLS | COMPILERS | BOOKS

To our valued customers

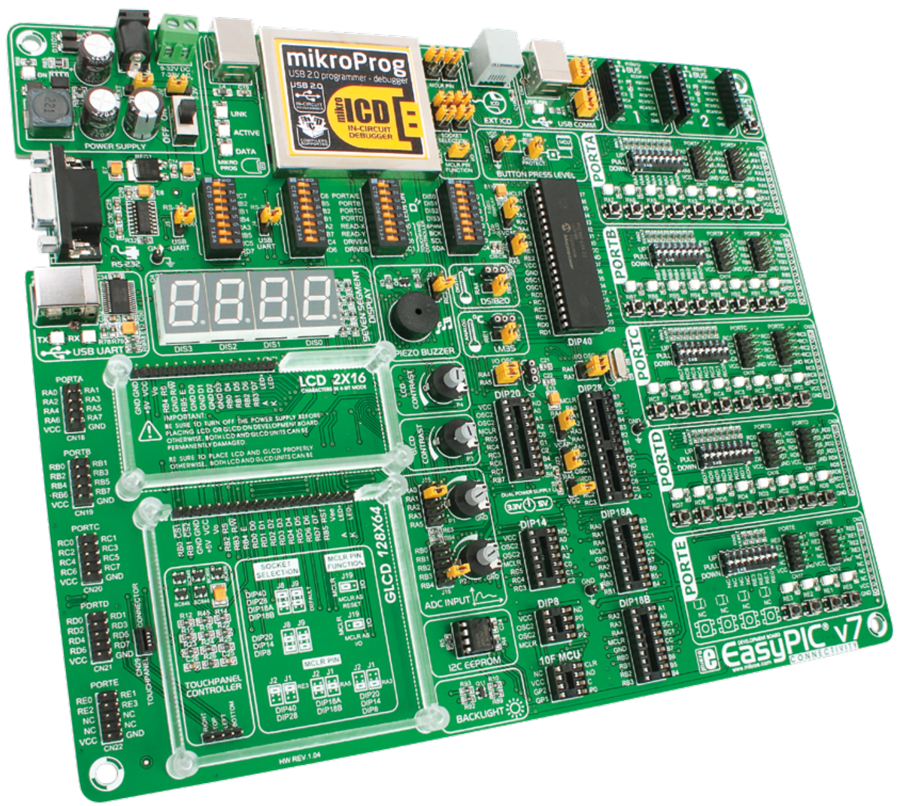
From the day one, we in MikroElektronika gave ourselves the highest possible goals in pursuit of excellence. That same day, the idea of EasyPIC™ development board was born. And we all grew together with EasyPIC™. In its each and tiniest piece we had put all of our energy, creativity and sense of what's best for an engineer. I've personally assembled hundreds of early EasyPIC™ boards myself with my home soldering iron.

Today, we present you the 7th generation of the board, which brings us some exciting new features. We hope that you will like it as much as we do.

Use it wisely and have fun!

Nebojsa Matic,
Owner and General Manager
of MikroElektronika

Table of contents



Introduction

EasyPIC™ is an old friend. It has been with us for six generations. Many of us made our first steps in embedded world with EasyPIC™. Today it has thousands of users: students, hobbyists, enthusiasts and professionals. It's used in many schools and other educational institutions across the globe. We may say that it's the most famous PIC development system in the world. We asked ourselves what we can do to make such a great board even greater. And we made some brilliant changes. We focused all of our creativity and knowledge into making a revolutionary new design, unlike any previous version of the board. We now present you with the new version 7 that brings so much more, and we hope that you will be thrilled with your new board, just as we are.

EasyPIC™ development Team

Four connectors for each port

Amazing connectivity

EasyPIC™ v7 is all about connectivity. Having four different connectors for each port, you can connect accessory boards, sensors and your custom electronics easier then ever before.

Everything is already here

mikroProg™ on board

Powerful on-board mikroProg™ programmer and In-Circuit debugger can program and debug over 350 microcontrollers. You will need it, whether you are a professional or a beginner.

3.3V and 5V power supply

Dual Power Supply

EasyPIC™ v7 is among few development boards which support both 3.3V and 5V microcontrollers. This feature greatly increases the number of supported MCUs. It's like having two boards instead of one!

For easier connections

mikroBUS™ support

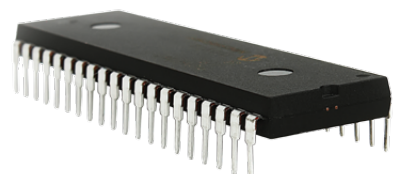
Just plug in your Click™ board, and it's ready to work. We picked up a set of the most useful pins you need for development and made a pinout standard you will enjoy using.

It's good to know

PIC18F45K22 is the new default microcontroller!

Until now, EasyPIC™ development boards were equipped with PIC16® as the default chip. Now we are giving you more power than ever before. **PIC18F45K22** is the new default chip of EasyPIC™ v7! It has **16 MIPS** operation, **32K bytes** of linear program memory, **1536 bytes** of linear data memory, and support for a wide range of power supply from **1.8V to 5V**. It's loaded with great modules: 36 General purpose **I/O pins**, 30 Analog Input pins (**AD**), Digital-To-Analog Converter (**DAC**), support for Capacitive Touch Sensing using Charge Time Measurement Unit (**CTMU**), three **8-bit timers** and four **16-bit timers**. It also has pair of **CCP, Comparators** and **MSSP** modules (which can be either **SPI** or **I²C**).

- *More power than ever before*
- *Great choice for both beginners and professionals*
- *Rich with modules*
- *Enough RAM and Flash*
- *Comes with examples for mikroC, mikroBasic and mikroPascal compilers*



System Specification

power supply
7-23V AC or 9-32V DC
or via USB cable (5V DC)

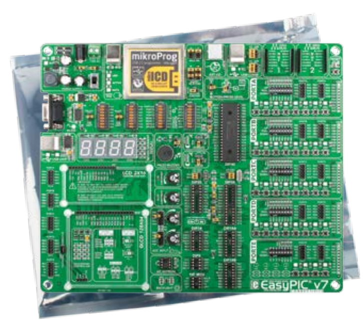
power consumption
~85mA when all peripheral modules are disconnected

board dimensions
266 x 220mm (10.47 x 8.66 inch)

weight
~445g (0.981 lbs)

Package contains

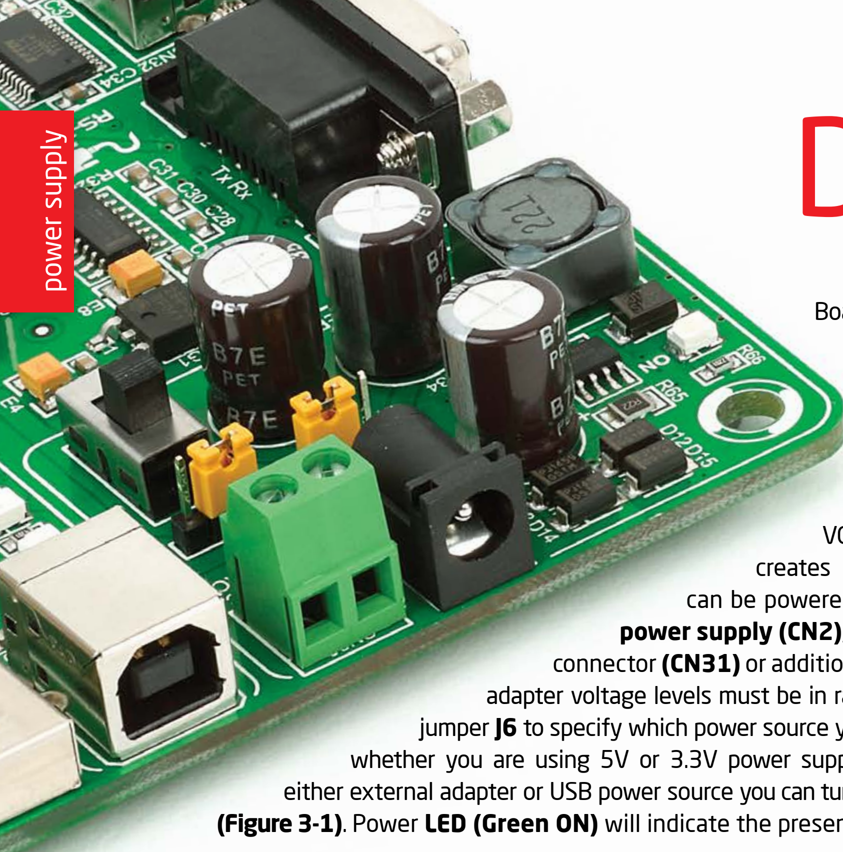
1. Damage resistant protective box



2. EasyPIC™ v7 board in antistatic bag

3. USB cable

4. User Manuals and Board schematic



Dual power supply

Board contains switching power supply that creates stable voltage and current levels necessary for powering each part of the board. Power supply section contains two power regulators: **MC34063A**, which generates VCC-5V, and **MC33269DT3.3** which creates VCC-3.3V power supply. The board can be powered in three different ways: with **USB power supply (CN2)**, using external adapters via adapter connector **(CN31)** or additional screw terminals **(CN30)**. External adapter voltage levels must be in range of **9-32V DC or 7-23V AC**. Use jumper **J6** to specify which power source you are using and jumper **J5** to specify whether you are using 5V or 3.3V power supply. Upon providing the power using either external adapter or USB power source you can turn on power supply by using **SWITCH 1 (Figure 3-1)**. Power **LED (Green ON)** will indicate the presence of power supply.

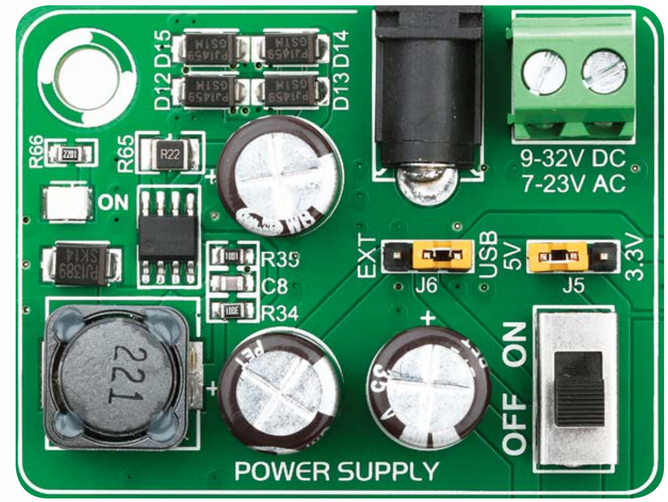


Figure 3-1: Dual power supply unit of EasyPIC™ v7

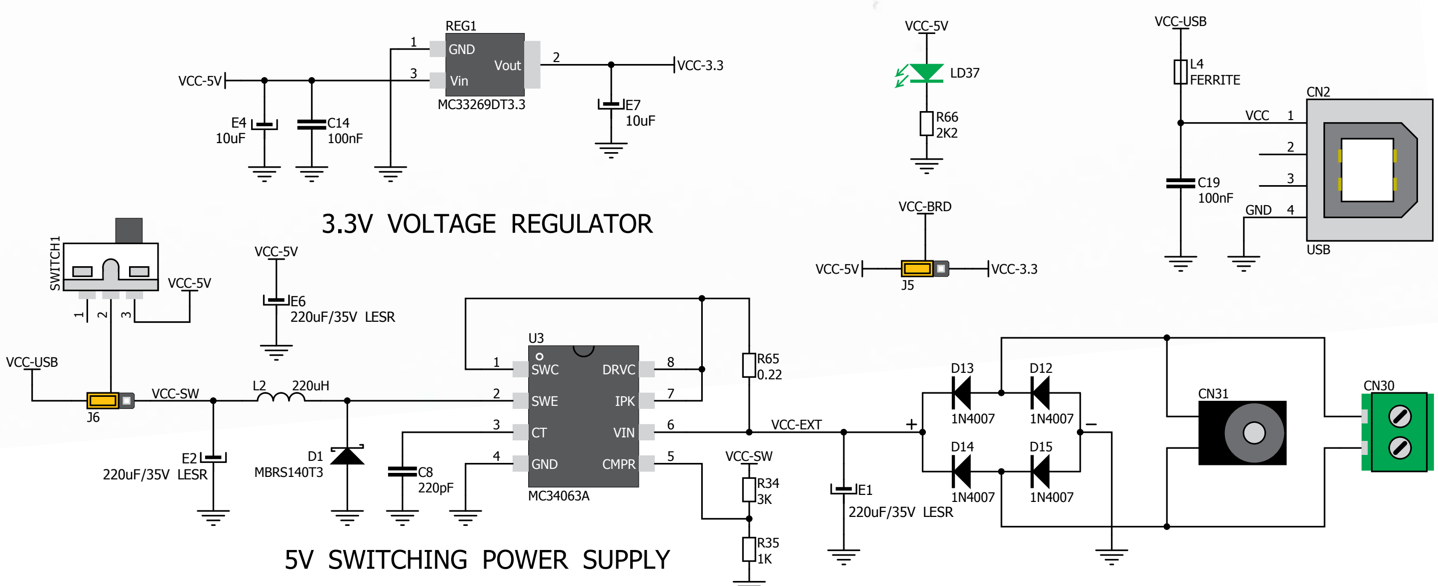


Figure 3-2: Dual power supply unit schematic

DUAL POWER SUPPLY
3.3V 5V

EasyPIC™ v7 development board supports both 3.3V and 5V power supply on a single board. This feature enables you to use wide range of peripheral boards.

Power supply: via DC connector or screw terminals (7V to 23V AC or 9V to 32V DC), or via USB cable (5V DC)

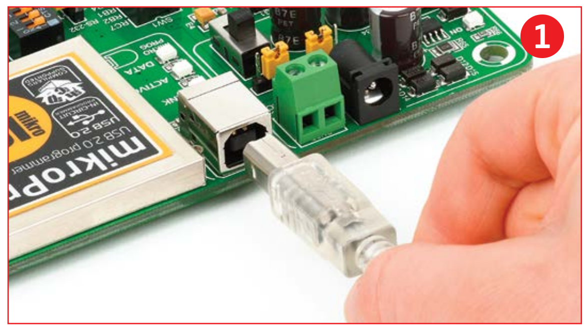
Power capacity: up to 500mA with USB, and up to 600mA with external power supply

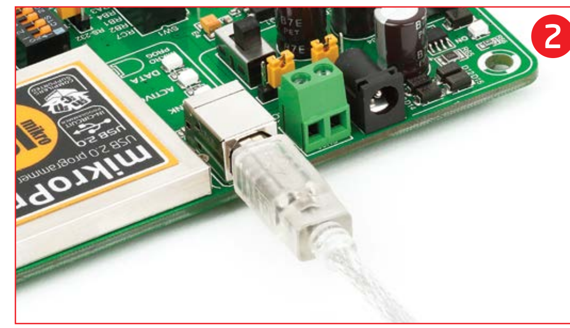
How to power the board?

1. With USB cable

Set J6 jumper to USB position

To power the board with USB cable, place jumper **J6** in USB position and place jumper **J5** in 5V or 3.3V position. You can then plug in the USB cable as shown on images 1 and 2, and turn the power switch ON.

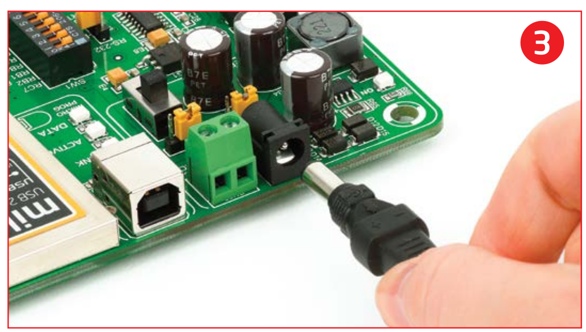


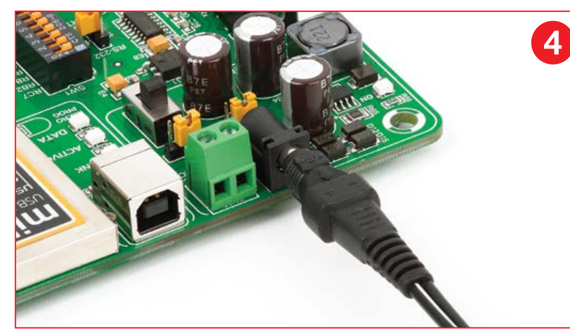


2. Using adapter

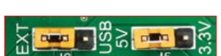
Set J6 jumper to EXT position

To power the board via adapter connector, place jumper **J6** in EXT position, and place jumper **J5** in 5V or 3.3V position. You can then plug in the adapter cable as shown on images 3 and 4, and turn the power switch ON.



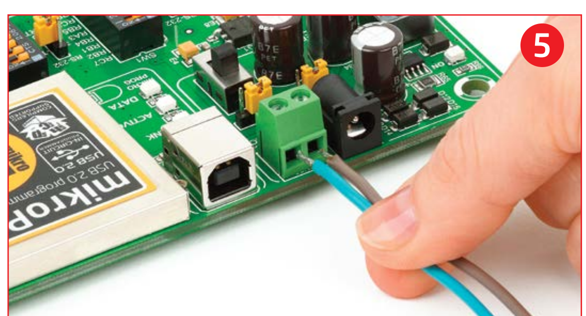


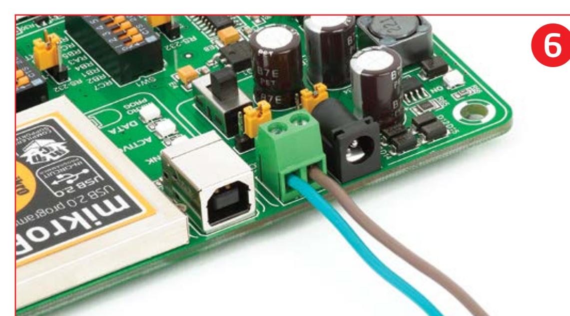
3. With laboratory power supply



Set J6 jumper to EXT position

To power the board using screw terminals, place jumper **J6** in EXT position, and place jumper **J5** in 5V or 3.3V position. You can then screw-on the cables in the screw terminals as shown on images 5 and 6, and turn the power switch ON.





Supported microcontrollers

The board contains eight DIP sockets: DIP40, DIP28, DIP20, DIP18A, DIP18B, DIP14, DIP8 and support for PIC10F MCUs. With dual power supply and smart on-board mikroProg, board is capable of programming over 350 microcontrollers from PIC10F, PIC12F, PIC16F, PIC16Enh, PIC18F, PIC18FJ and PIC18FK families.

There are two DIP18 sockets for PIC microcontrollers provided on the board - **DIP18A** and **DIP18B**. Which of these sockets you will use depends solely on the pinout of the microcontroller in use. The **EasyPIC™ v7** development system comes with the **PIC18F45K22** microcontroller in a **DIP40** package.

IMPORTANT: *When using PIC18F2331 or PIC18F2431 microcontrollers it is necessary to place J20 jumper, in order to route VCC power line to RA5 pin (Figure 4-1)*

VCAP jumpers explained

Some PIC16F, PIC18FK and all PIC18FJ microcontrollers have cores that work on 1.8V-2.5V voltage range, and peripherals that work with 3.3V and 5V voltages. Internally, those microcontrollers have power regulators, which adjust the core voltage levels. In order for those devices to have a stable operation of the core, manufacturer recommends that decoupling capacitive filters should be provided, and connected between specific microcontroller pins designated with VCAP and GND. EasyPIC v7 board provides jumpers which are used for this purpose. Here is list of devices that require jumpers placed in VCAP position:

(see figure 4-1)

J22	**VCAP position when using PIC16F724/16F727**
J7	**VCAP position for PIC18F44J10 and PIC18F45J10**
J10	**VCAP for PIC18F24J10, PIC18F25J10 PIC18F2XJ50, PIC18F2XJ11**
J23	**VCAP for PIC16F722, PIC16F723, PIC16F726**

IMPORTANT: *If you do not place VCAP jumper for the MCUs that need it, you might experience some instabilities in program execution.*

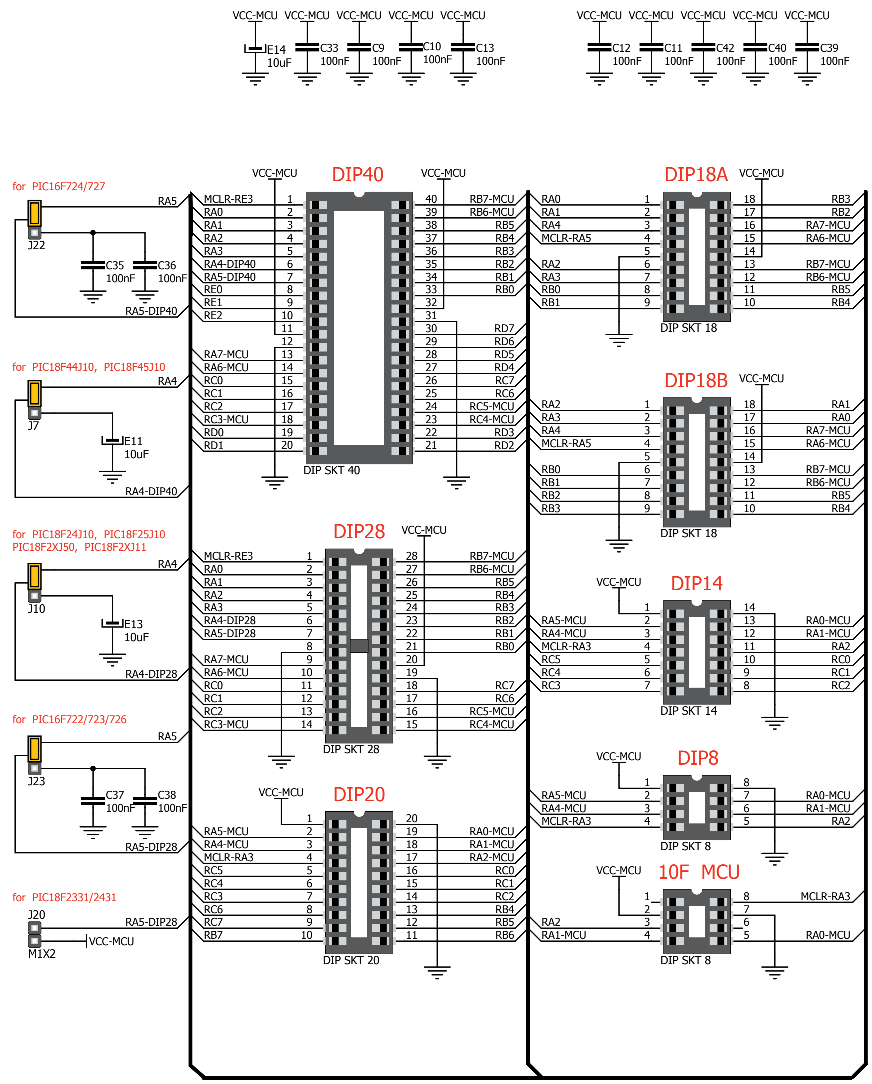
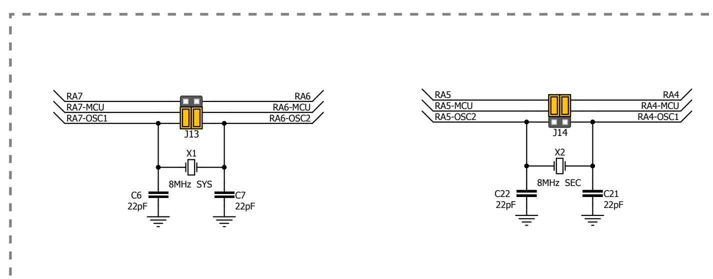


Figure 4-1: Schematic of on-board DIP sockets and VCAP jumpers

Figure 4-2: crystal oscillators



How to properly place your microcontroller into the DIP socket?

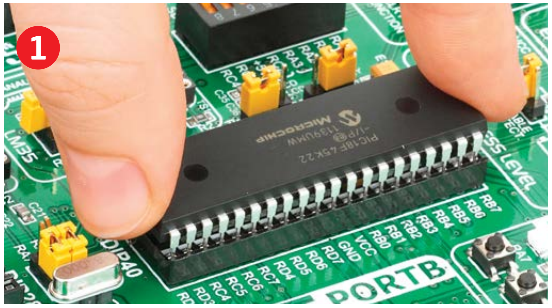


Figure 4-3: Place both ends of microcontroller on the socket so the pins are aligned correctly

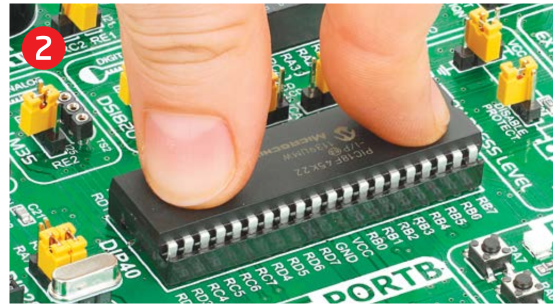


Figure 4-4: with both fingers, evenly distribute the force and press the chip into the socket.


Figure 4-5: Properly placed microcontroller will have equally leveled pins.

Before you plug the microcontroller into the appropriate socket, make sure that the **power supply is turned off.** Images above show how to correctly plug a microcontroller. First make sure that a half circular cut in the microcontroller DIP packaging matches the cut in the DIP socket. Place both ends of the microcontroller into the socket as shown in **Figure 4-3**. Then put the microcontroller slowly down until all the pins match the socket as shown in **Figure 4-4**. Check again if everything is placed correctly and press the microcontroller until it is completely plugged into the socket as shown in **Figure 4-5**.

IMPORTANT: *Only one microcontroller may be plugged into the development board at the same time.*

Using crystal oscillators

Figure 4-6: RA6 and RA7 as I/O pins (when using internal oscillator)

Figure 4-7: RA6 and RA7 connected to X1 **quartz-crystal**

PIC microcontrollers normally use a quartz crystal for the purpose of providing clock frequency. The **EasyPIC™ v7** provides two sockets for quartz-crystal. Microcontrollers in **DIP18A**, **DIP18B**, **DIP28** and **DIP40** packages use socket **X1 (OSC1)** for quartz-crystal.

Figure 4-8: RA4 and RA5 as I/O pins (when using internal oscillator)

Figure 4-9: RA4 and RA5 connected to X2 **quartz-crystal**

If you want to use microcontrollers in **DIP8**, **DIP14** and **DIP20** packages, it is necessary to put quartz crystal into socket **X2 (OSC2)**. The value of the quartz-crystal depends on the maximum clock frequency allowed and your application. You can always exchange the default 8MHz crystal with another one.

IMPORTANT: *Microcontrollers which are plugged into socket 10F use their own internal oscillator and are not connected to any of the mentioned quartz-crystal sockets.*

On-board programmer

What is mikroProg™?

mikroProg™ is a fast USB 2.0 programmer with mikroICD™ hardware In-Circuit Debugger. Smart engineering allows mikroProg™ to support all PIC10, PIC12, PIC16, PIC18, devices in a single programmer! It supports over 350 microcontrollers from Microchip®. Outstanding performance and easy operation are among it's top features.

How do I start?

In order to start using mikroProg™ and program your microcontroller, you just have to follow two simple steps:

1. Install the necessary software

- Install USB drivers
- Install mikroProg Suite™ for PIC® software

2. Power up the board, and you are ready to go.

- Plug in the programmer USB cable
- LINK LED should light up.

Why so many LEDs?

Three LEDs indicate specific programmer operation. **Link** LED lights up when USB link is established with your PC, **Active** LED lights up when programmer is active. **Data** is on when data is being transferred between the programmer and PC software (compiler or mikroProg Suite™ for PIC®).

NOTE: *If you use other than the default PIC18F45K22 MCU, make sure that programmer jumpers are placed in proper positions for your microcontroller socket.*

MCLR pin selection

Before using the programmer, make sure to set MCLR pin jumpers **J1** and **J2**, so that MCLR line is routed to the correct socket for your microcontroller. If you are using the default **PIC18F45K22**, jumpers are supposed to be set for DIP40, as shown below.

DIP40, DIP28

DIP18A, DIP18B

DIP20 DIP14 DIP8

Programing lines selection

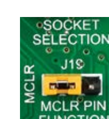
Jumpers **J8** and **J9** are used to select **PGC** and **PGD** programming lines for your microcontroller. Make sure to place jumpers in the proper position for your socket.

DIP40, DIP28, DIP18A, DIP18B

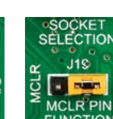
DIP20, DIP14, DIP8

MCLR pin function

Using jumper **J19** you can specify whether **MCLR** pin of your microcontroller is connected to the on-board reset circuit, or acts just as I/O pin.



MCLR as MCLR



MCLR as I/O

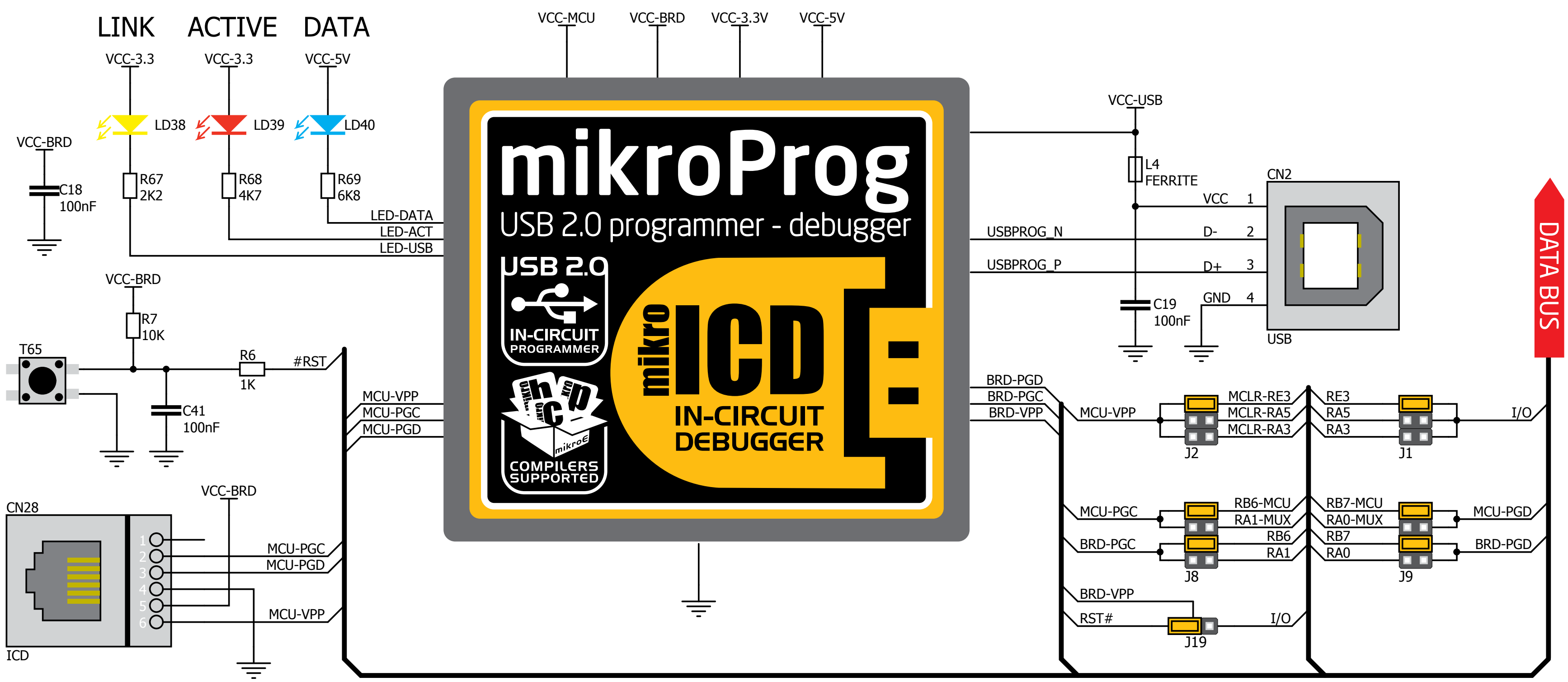
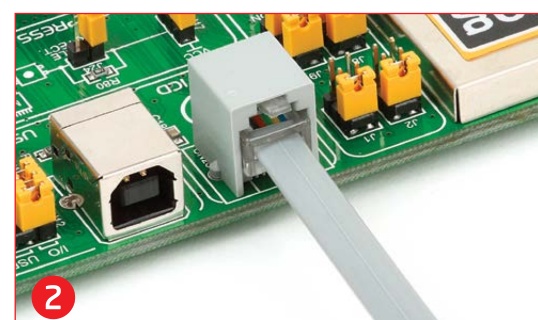


Figure 5-1: mikroProg™ block schematic

Programming with ICD2/ICD3

EasyPIC™ v7 is equipped with RJ-12 connector compatible with Microchip® ICD2® and ICD3® external programmers. You can either use the on-board mikroProg™ programmer or external programming tools as long as you use only one of them in the same time. But you still have to **set the appropriate jumpers**, as described in the previous page. Insert your ICD programmer cable into connector **CN28**, as shown in images 1 and 2.


1

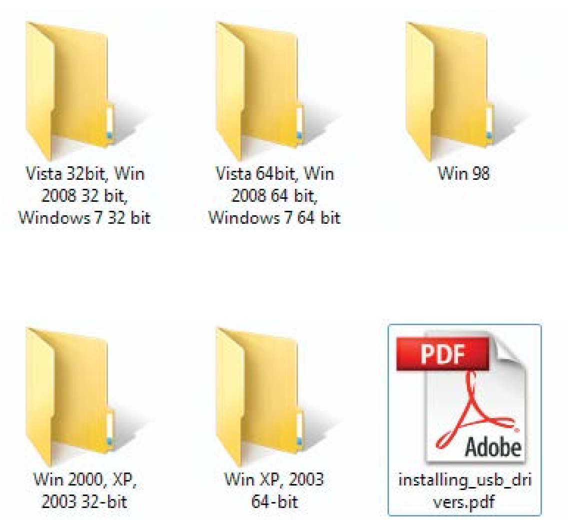

2

Installing programmer drivers

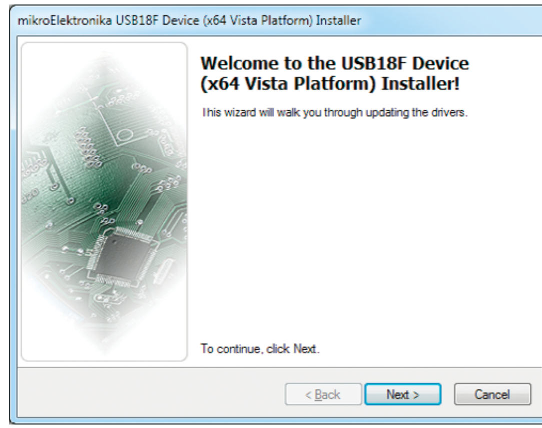
On-board mikroProg™ requires drivers in order to work. Drivers can be found on the link below:

http://www.mikroe.com/downloads/get/1202/mikroprog_for_pic_drivers_v200.zip

When you locate the drivers, please extract files from the ZIP archive. Folder with extracted files contains sub folders with drivers for different operating systems. Depending on which operating system you use, choose adequate folder and open it.

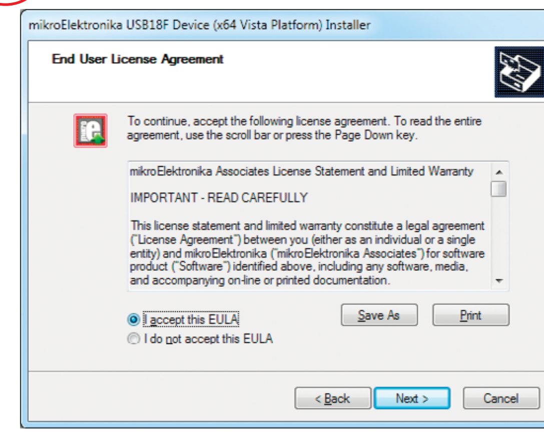


In the opened folder you should be able to locate the driver setup file. Double click on setup file to begin installation of the programmer drivers.



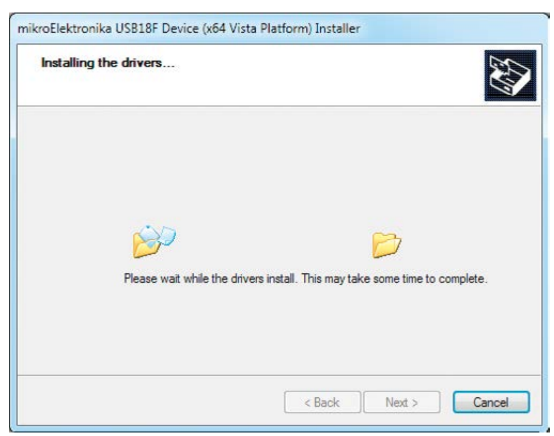
Step 1 - Start Installation

Welcome screen of the installation. Just click on **Next** button to proceed.



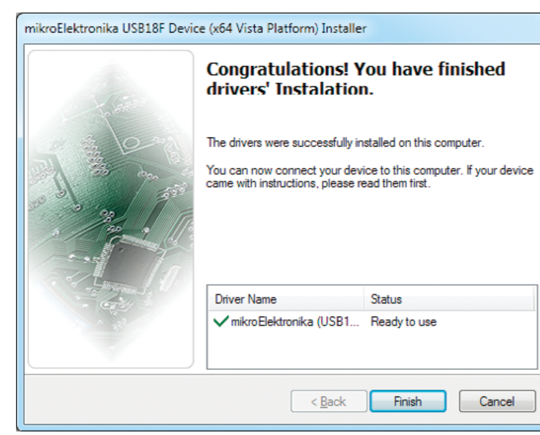
Step 2 - Accept EULA

Carefully read **E**nd **U**ser **L**icense **A**greement. If you agree with it, click **Next** to proceed.



Step 3 - Installing drivers

Drivers are installed automatically in a matter of seconds.



Step 4 - Finish installation

You will be informed if the drivers are installed correctly. Click on **Finish** button to end installation process.

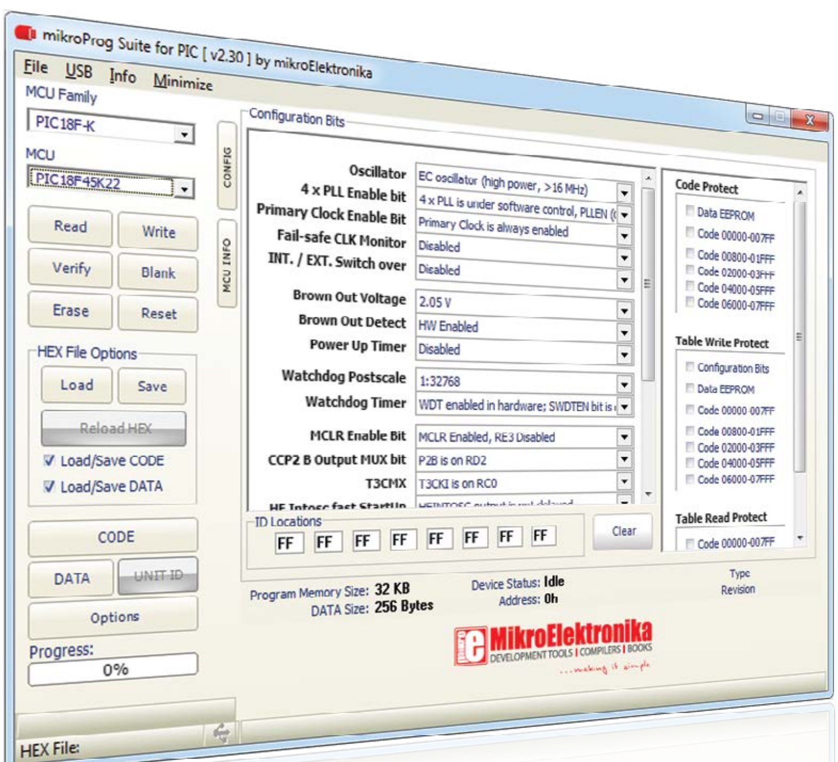
Programming software

mikroProg Suite™ for PIC®

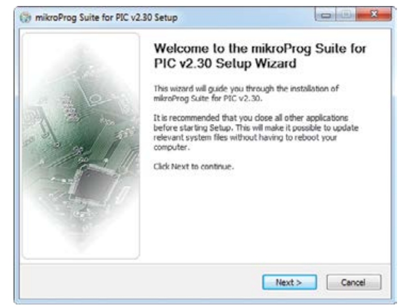
On-board **mikroProg™** programmer requires special programming software called **mikroProg Suite™ for PIC®**. This software is used for programming all of Microchip® microcontroller families, including PIC10, PIC12, PIC16, PIC18, dsPIC30/33, PIC24 and PIC32. Software has intuitive interface and **SingleClick™** programming technology. To begin, first locate the installation archive on our website:

http://www.mikroe.com/downloads/get/1201/mikroprog_suite_for_pic.zip

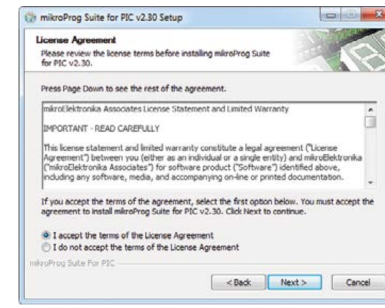
After downloading, extract the package and double click the executable setup file, to start installation.



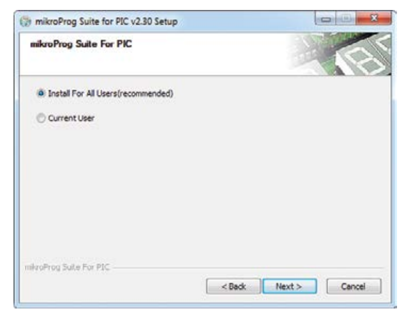
Installation wizard - 6 simple steps



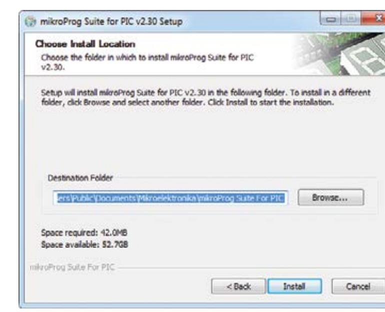
Step 1 - Start Installation



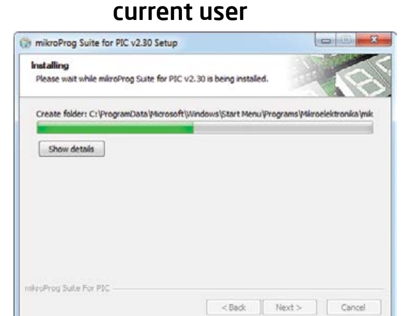
Step 2 - Accept EULA and continue



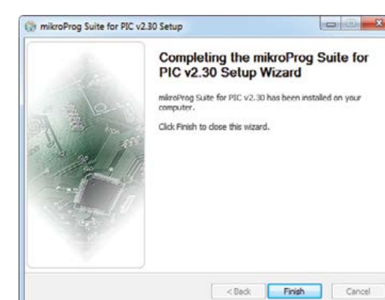
Step 3 - Install for All users or current user



Step 4 - Choose destination folder



Step 5 - Installation in progress



Step 6 - Finish Installation

mikroICD™ - In Circuit Debugger

What is Debugging?

Every developer comes to a point where he has to monitor the code execution in order to find errors in the code, or simply to see if everything is going as planed. This hunt for bugs, or errors in the code is called **debugging**. There are two ways to do this: one is **the software simulation**, which enables you to simulate what is supposed to be happening on the microcontroller as your code lines are executed, and the other, most reliable one, is monitoring the code execution on the MCU itself. And this latter one is called **In-Circuit debugging**. "In-Circuit" means that it is the real deal - code executes right on the target device.

What is mikroICD™?

The on-board **mikroProg™** programmer supports **mikroICD™** - a highly effective tool for a **Real-Time debugging** on hardware level. The mikroICD™ debugger enables you to execute your program on the host PIC microcontroller and view variable values, Special Function Registers (SFR), RAM, CODE and EEPROM memory along with the mikroICD™ code execution on hardware. Whether you are a beginner, or a professional, this powerful tool, with intuitive interface and convenient set of commands will enable you to track down bugs quickly. mikroICD™ is one of the fastest, and most reliable debugging tools on the market.

Supported Compilers

All MikroElektronika compilers, **mikroC™**, **mikroBasic™** and **mikroPascal™** for PIC®, dsPIC® and PIC32® natively support mikroICD™. Specialized mikroICD™ DLL module allows compilers to exploit the full potential of fast hardware debugging. Along with compilers, make sure to install the appropriate **programmer drivers** and **mikroProg Suite™ for PIC®** programming software, as described on **pages 12** and **13**.

How do I use the debugger?

When you build your project for debugging, and program the microcontroller with this HEX file, you can start the debugger using **[F9]** command. Compiler will change layout to debugging view, and a blue line will mark where code execution is currently paused. Use **debugging toolbar** in the **Watch Window** to guide the program execution and stop anytime. Add the desired variables to Watch Window and monitor their values. Complete guide to using mikroICD™ with your compiler is provided within the EasyPIC™ v7 package.

Figure 5-2: mikroICD™ manual explains debugging thoroughly

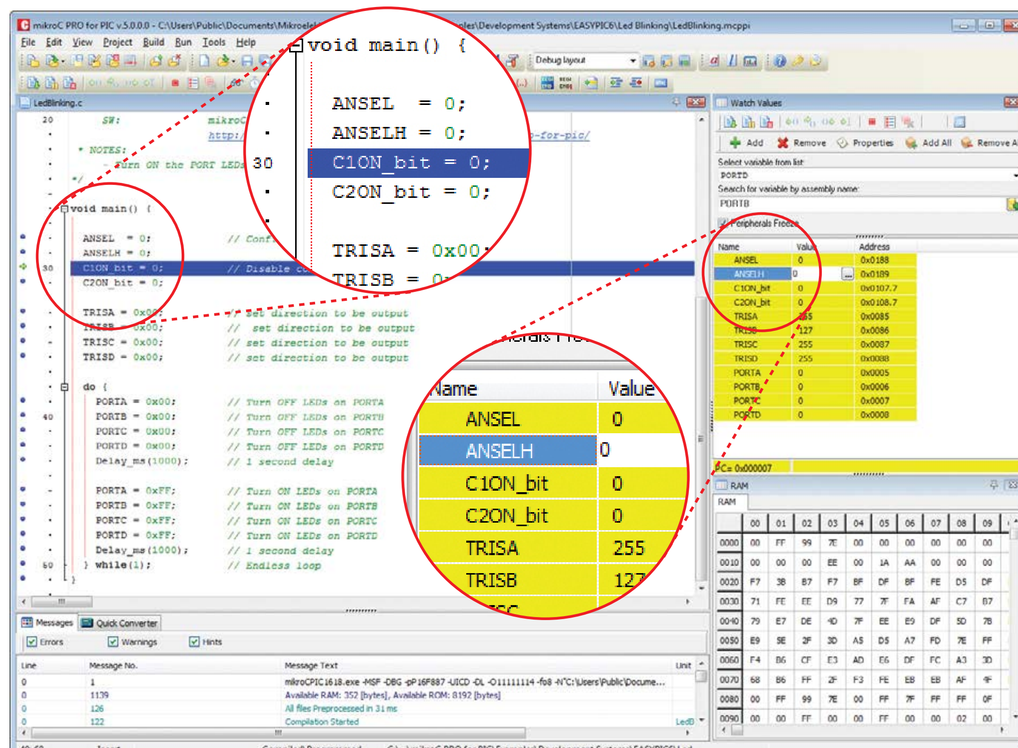


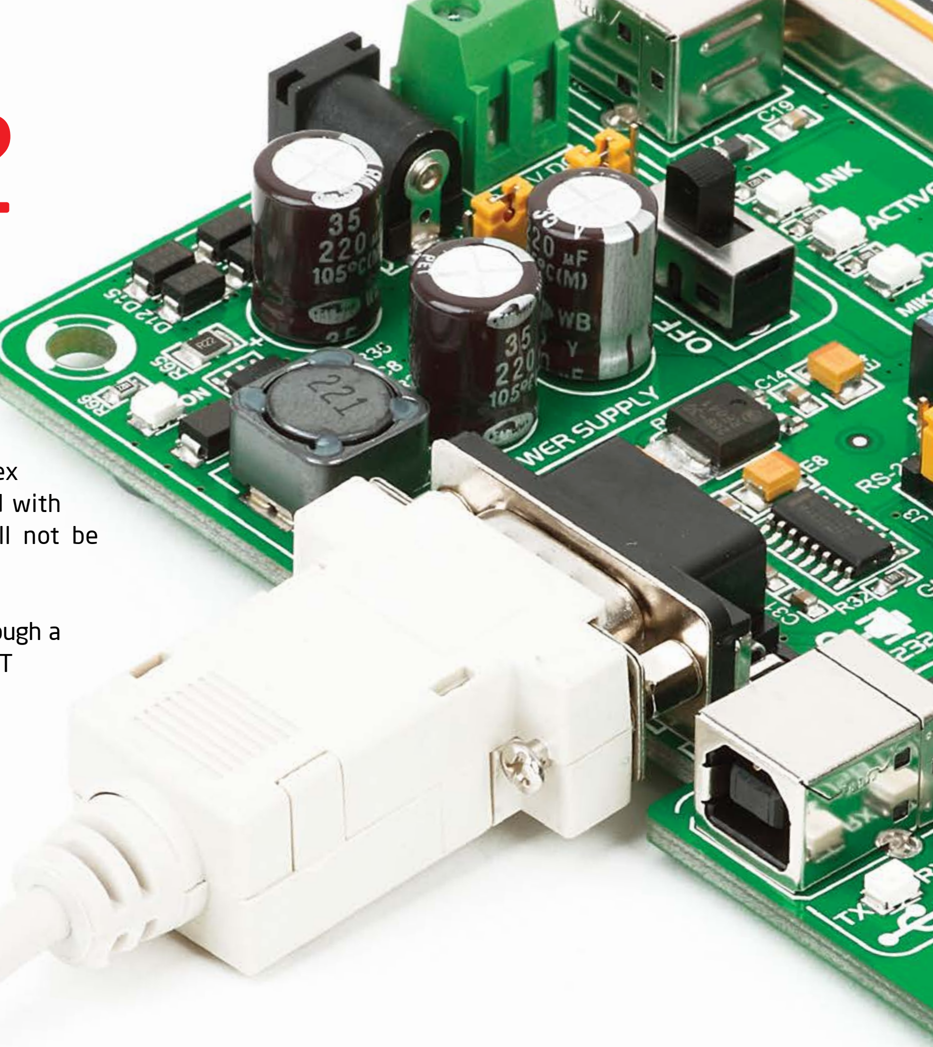
Figure 5-3: mikroC PRO for PIC® compiler in debugging view, with SFR registers in Watch Window

mikroICD™ commands

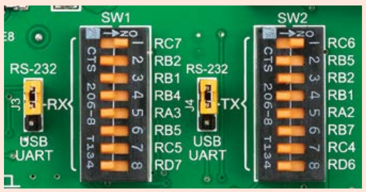
Here is a short overview of which debugging commands are supported in MikroElektronika compilers. You can see what each command does, and what are their shortcuts when you are in debugging mode. It will give you some general picture of what your debugger can do.

Toolbar Icon	Command Name	Shortcut	Description
	Start Debugger	[F9]	Starts Debugger.
	Run/Pause Debugger	[F6]	Run/Pause Debugger.
	Stop Debugger	[Ctrl + F2]	Stops Debugger.
	Step Into	[F7]	Executes the current program line, then halts. If the executed program line calls another routine, the debugger steps into the routine and halts after executing the first instruction within it.
	Step Over	[F8]	Executes the current program line, then halts. If the executed program line calls another routine, the debugger will not step into it. The whole routine will be executed and the debugger halts at the first instruction following the call.
	Step Out	[Ctrl + F8]	Executes all remaining program lines within the subroutine. The debugger halts immediately upon exiting the subroutine.
	Run To Cursor	[F4]	Executes the program until reaching the cursor position.
	Toggle Breakpoint	[F5]	Toggle breakpoints option sets new breakpoints or removes those already set at the current cursor position.
	Show/Hide breakpoints	[Shift+F4]	Shows/Hides window with all breakpoints
	Clears breakpoints	[Shift+Ctrl+F5]	Delete selected breakpoints
	Jump to interrupt	[F2]	Opens window with available interrupts (doesn't work in mikroICD™ mode)

UART via RS-232



Enabling RS-232



In order to enable RS-232 communication, you must set **J3** and **J4** jumpers in the **RS-232** position, and enable desired **RX** and **TX** lines via **SW1** and **SW2** DIP switches. For example, if you want to enable RS-232 connection on UART1 module of the default **PIC18F45K22** chip, you should enable **SW1.1** (RC7) and **SW2.1** (RC6) lines.

The **UART** (universal asynchronous receiver/transmitter) is one of the most common ways of exchanging data between the MCU and peripheral components. It is a serial protocol with separate transmit and receive lines, and can be used for full-duplex communication. Both sides must be initialized with the same baud rate, otherwise the data will not be received correctly.

RS-232 serial communication is performed through a 9-pin SUB-D connector and the microcontroller UART module. In order to enable this communication, it is necessary to establish a connection between **RX** and **TX** lines on SUB-D connector and the same pins on the target microcontroller using DIP switches. Since RS-232 communication voltage levels are different than microcontroller logic levels, it is necessary to use a RS-232 Transceiver circuit, such as **MAX3232** as shown on **Figure 6-1**.

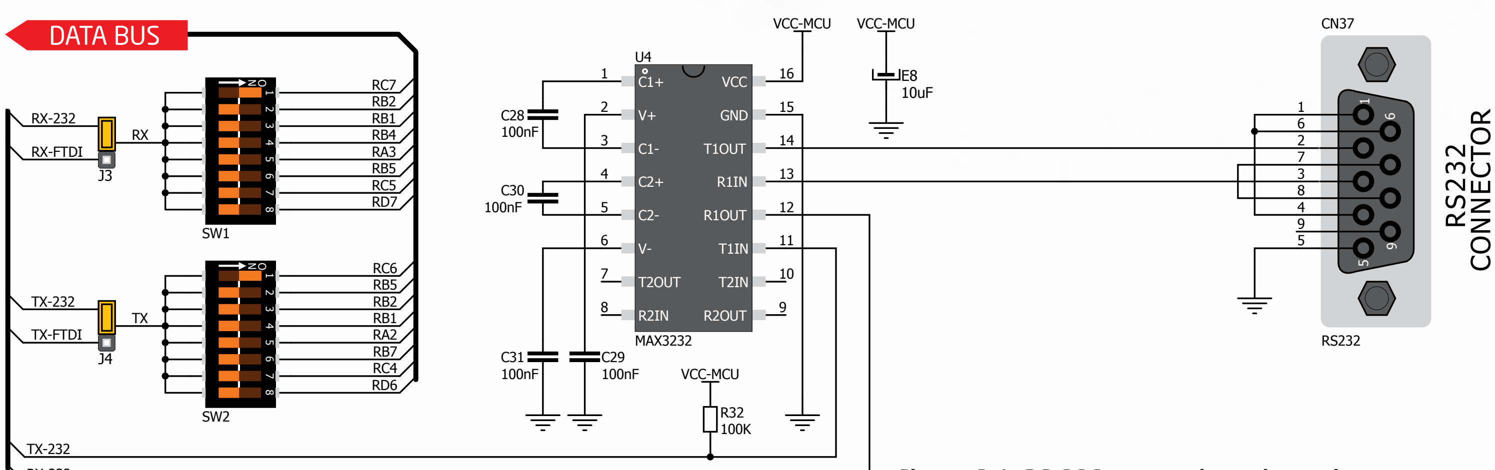
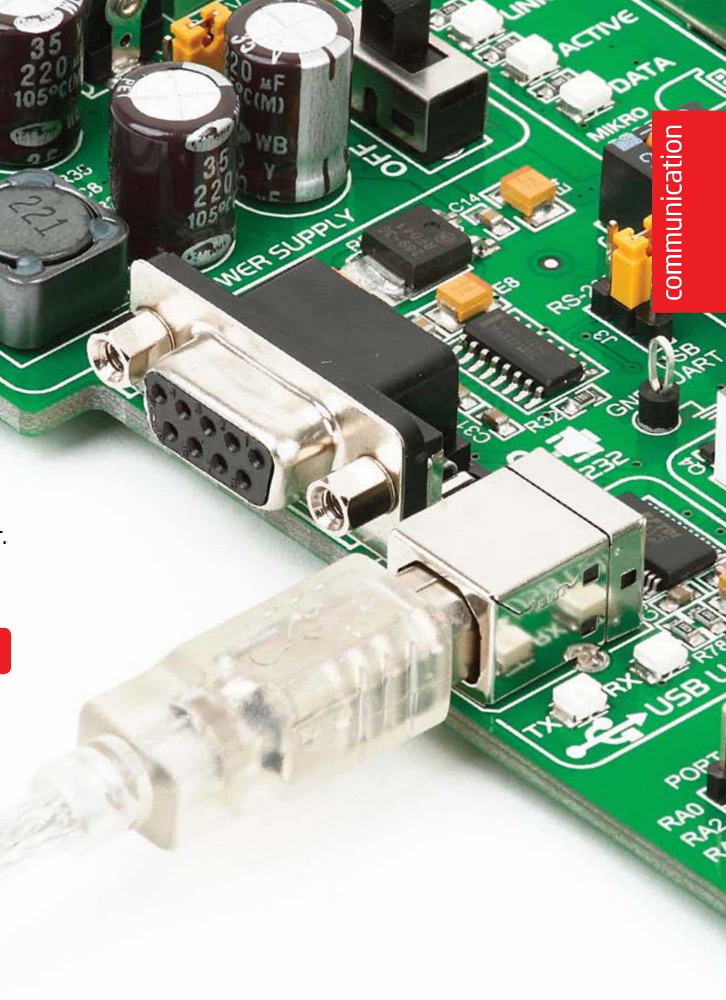
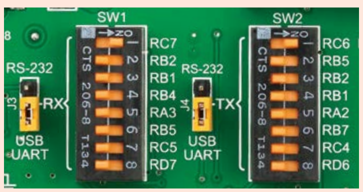


Figure 6-1: RS-232 connection schematic

UART via USB



Enabling USB-UART



In order to enable USB-UART communication, you must set **J3** and **J4** jumpers in the **USB-UART** position, and enable desired **RX** and **TX** lines via **SW1** and **SW2** DIP switches. For example, if you want to enable USB-UART connection on UART1 module of the default **PIC18F45K22** chip, you should enable **SW1.1** (RC7) and **SW2.1** (RC6) lines.

Modern PC computers, laptops and notebooks are no longer equpped with RS-232 connectors and UART controllers. They are nowdays replaced with USB connectors and USB controllers. Still, certain technology enables UART communication to be done over USB connection. Controllers such as **FT232RL** from FTDI® convert UART signals to the appropriate USB standard. In order to use USB-UART module on EasyPIC™ v7, you must first install FTDI drivers on your computer. Drivers can be found on link bellow:

http://www.ftdichip.com/Drivers/VCP.htm

USB-UART communication is being done through a FT232RL controller, USB connector **(CN32)**, and microcontroller UART module. To establish this connection, you must put **J3** and **J4** jumpers in the **USB-UART** position, and connect **RX** and **TX** lines to the appropriate pins of the microcontroller. This connection is done using DIP switches **SW1** and **SW2**.

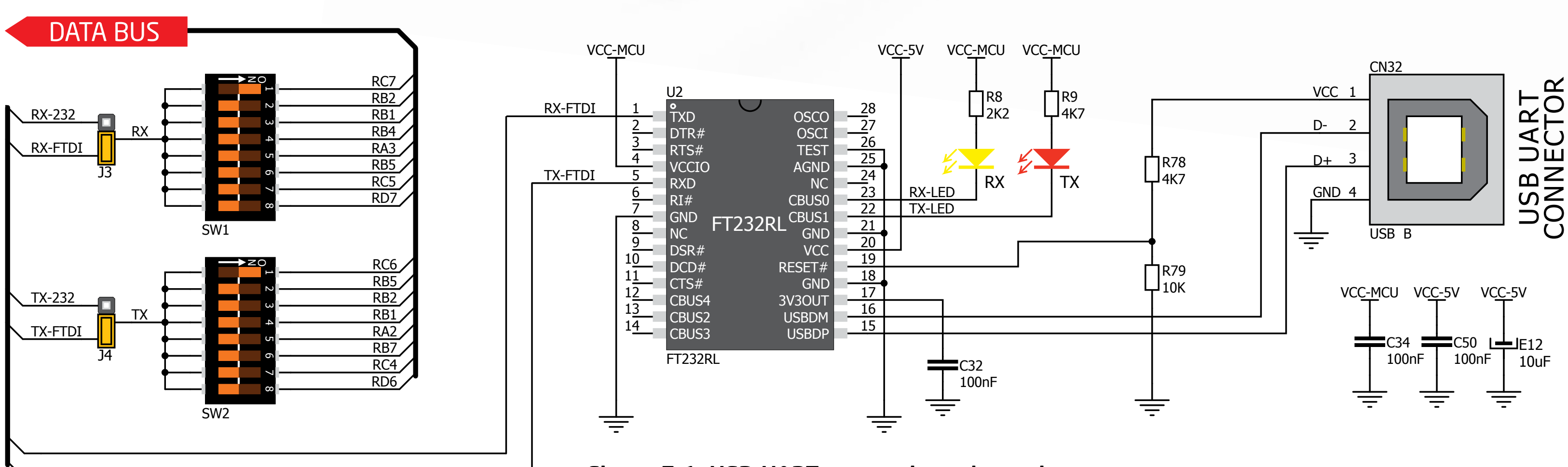


Figure 7-1: USB-UART connection schematic

USB connection

USB is the acronym for **Universal Serial Bus**. This is a very popular industry standard that defines cables, connectors and protocols used for communication and power supply between computers and other devices. EasyPIC™ v7 contains USB connector **(CN4)** which enables microcontrollers that support USB communication to establish a connection with the target host (eg. PC, Laptop, etc). Selection of communication lines is done using jumpers **J12** or **J18**, depending on the target microcontroller.

When communication lines are routed from the microcontroller to the USB connector using mentioned jumpers, they are cut off from the rest of the board, and cannot be accessed via PORT headers.

Dedicated **USB ON** LED signalizes the presence of USB connection, when the USB cable is inserted into the USB connector.

Enabling USB connection

Depending on your target microcontroller, USB communication can be enabled on **PORTA** or **PORTC**. For **PIC18(L)F1XK50** you should put **J18** jumpers in the **USB position** (**Figure 8-3**). For **PIC18Fxx(J)50**, **PIC18Fxx(J)53**, **PIC18Fxx(J)55** and **PIC18Fxx58** place **J12** jumpers in the **USB position** (**Figure 8-2**).

Figure 8-1:
USB function disabled

Figure 8-2:
USB enabled on PORTC

Figure 8-3:
USB enabled on PORTA

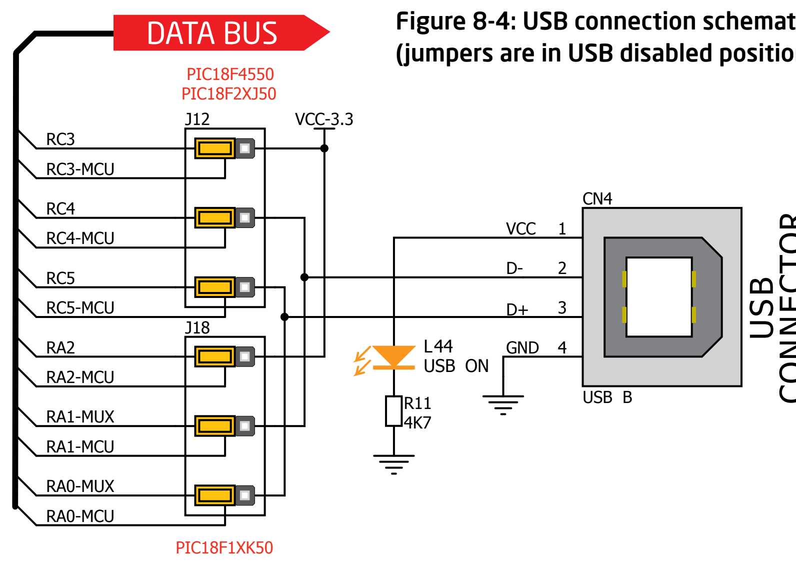


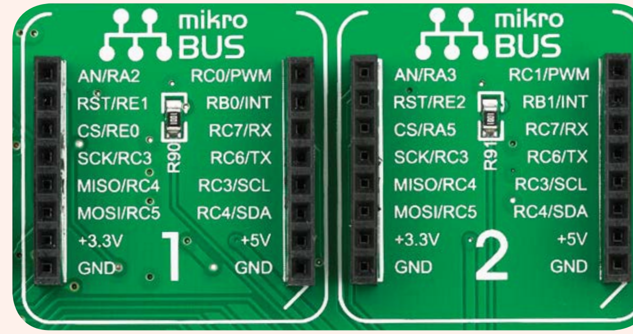
Figure 8-4: USB connection schematic (jumpers are in USB disabled position)

mikroBUS™ sockets

Easier connectivity and simple configuration are imperative in modern electronic devices. Success of the USB standard comes from it's simplicity of usage and high and reliable data transfer rates. As we in MikroElektronika see it, Plug-and-Play devices with minimum settings are the future in embedded world too. This is why our engineers have come up with a simple, but brilliant pinout with lines that most of today's accessory boards require, which almost completely eliminates the need of additional hardware settings. We called this new standard the **mikroBUS™.** EasyPIC™ v7 is the first development board in the world to support mikroBUS™ with two on-board sockets. As you can see, there are no additional DIP switches, or jumper selections. Everything is already routed to the most appropriate pins of the microcontroller sockets.

mikroBUS™ host connector

Each mikroBUS™ host connector consists of two 1x8 female headers containing pins that are most likely to be used in the target accessory board. There are three groups of communication pins: **SPI**, **UART** and **I²C** communication. There are also single pins for **PWM**, **Interrupt**, **Analog input**, **Reset** and **Chip Select**. Pinout contains two power groups: **+5V and GND** on one header and **+3.3V and GND** on the other 1x8 header.



mikroBUS™ pinout explained

AN - Analog pin
RST - Reset pin
CS - SPI Chip Select line
SCK - SPI Clock line
MISO - SPI Slave Output line
MOSI - SPI Slave Input line
+3.3V - VCC-3.3V power line
GND - Reference Ground

PWM - PWM output line
INT - Hardware Interrupt line
RX - UART Receive line
TX - UART Transmit line
SCL - I2C Clock line
SDA - I2C Data line
+5V - VCC-5V power line
GND - Reference Ground

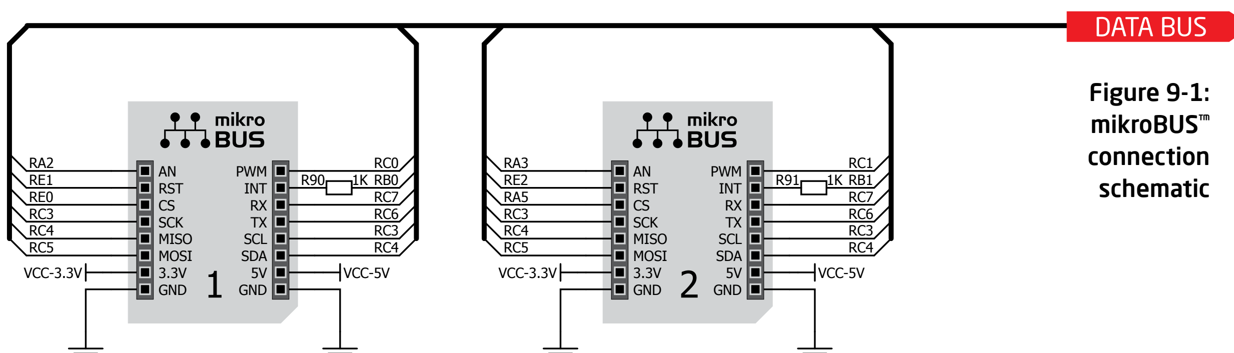
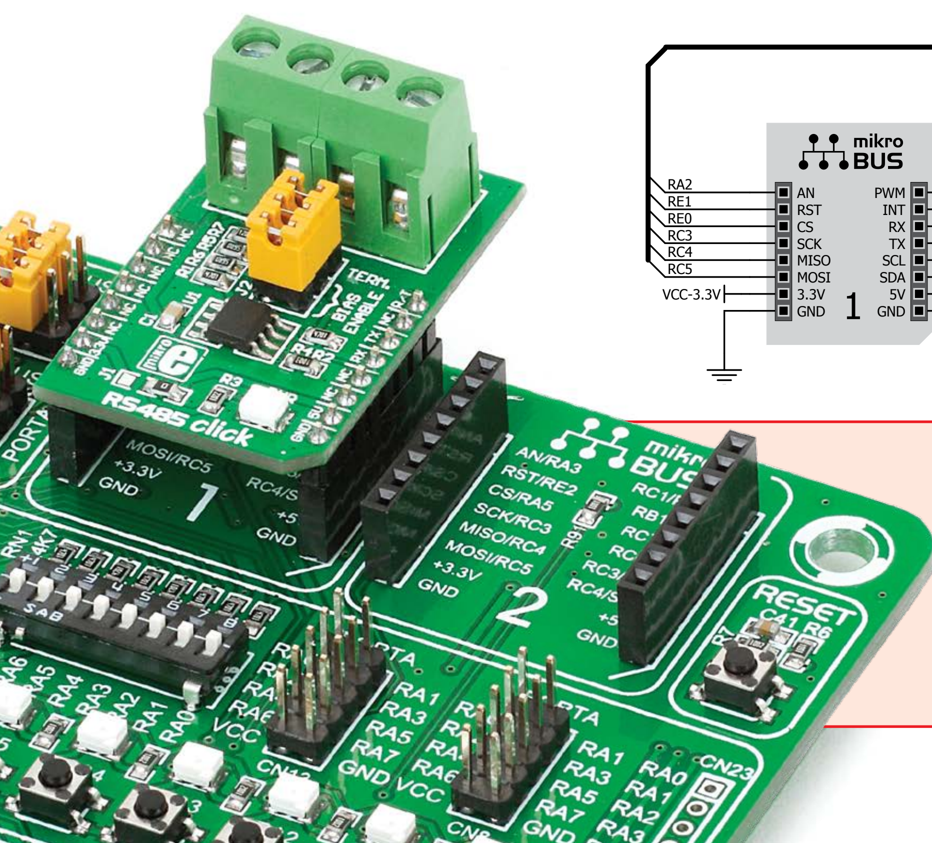


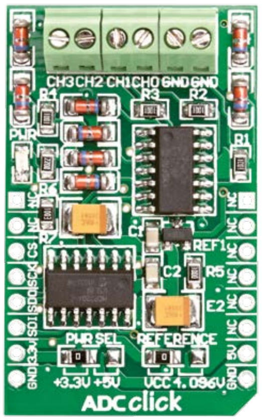
Figure 9-1: mikroBUS™ connection schematic



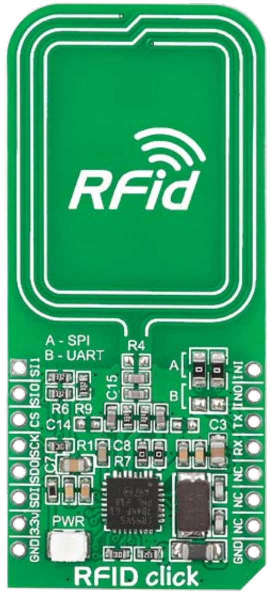
Integrate mikroBUS™ in your design

mikroBUS™ is not made to be only a part of our development boards. You can freely place mikroBUS™ host connectors in your final PCB designs, as long as you clearly mark them with mikroBUS™ logo and footprint specifications. For more information, logo artwork and PCB files visit our website:

http://www.mikroe.com/mikrobus/

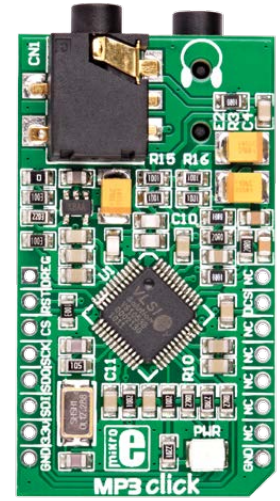


ADC click™



RFiD click™

BlueTooth click™



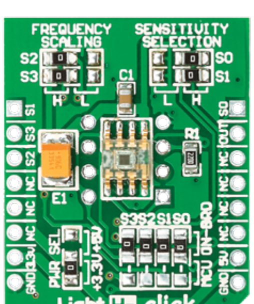
MP3 click™

GSM click™

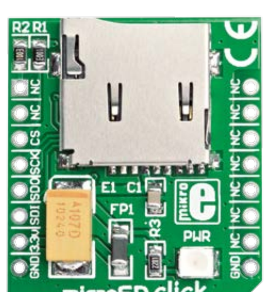
Click Boards™ are plug-n-play!

MikroElektronika portfolio of over 200 accessory boards is now enriched by an additional set of mikroBUS™ compatible **Click Boards™**. Almost each month several new Click boards™ are released. It is our intention to provide the community with as much of these boards as possible, so you will be able to expand your EasyPIC™ v7 with additional functionality with literally zero hardware configuration. Just plug and play. Visit the Click boards™ webpage for the complete list of available boards:

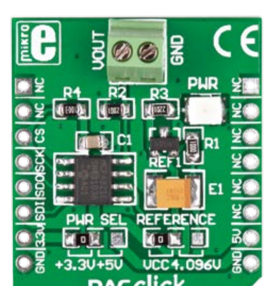
http://www.mikroe.com/click/



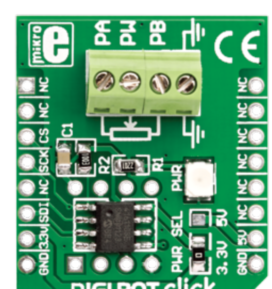
LightHz click™



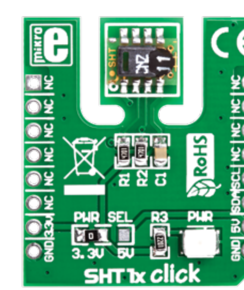
microSD click™



DAC click™



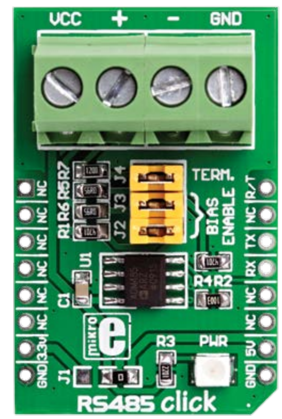
DIGIPOT click™



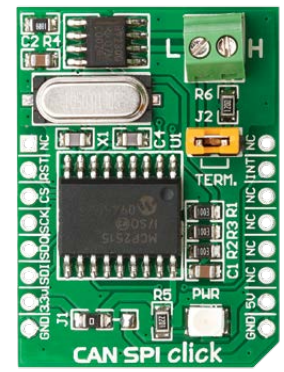
SHT1x click™

WiFi PLUS click™

GPS click™



RS485 click™



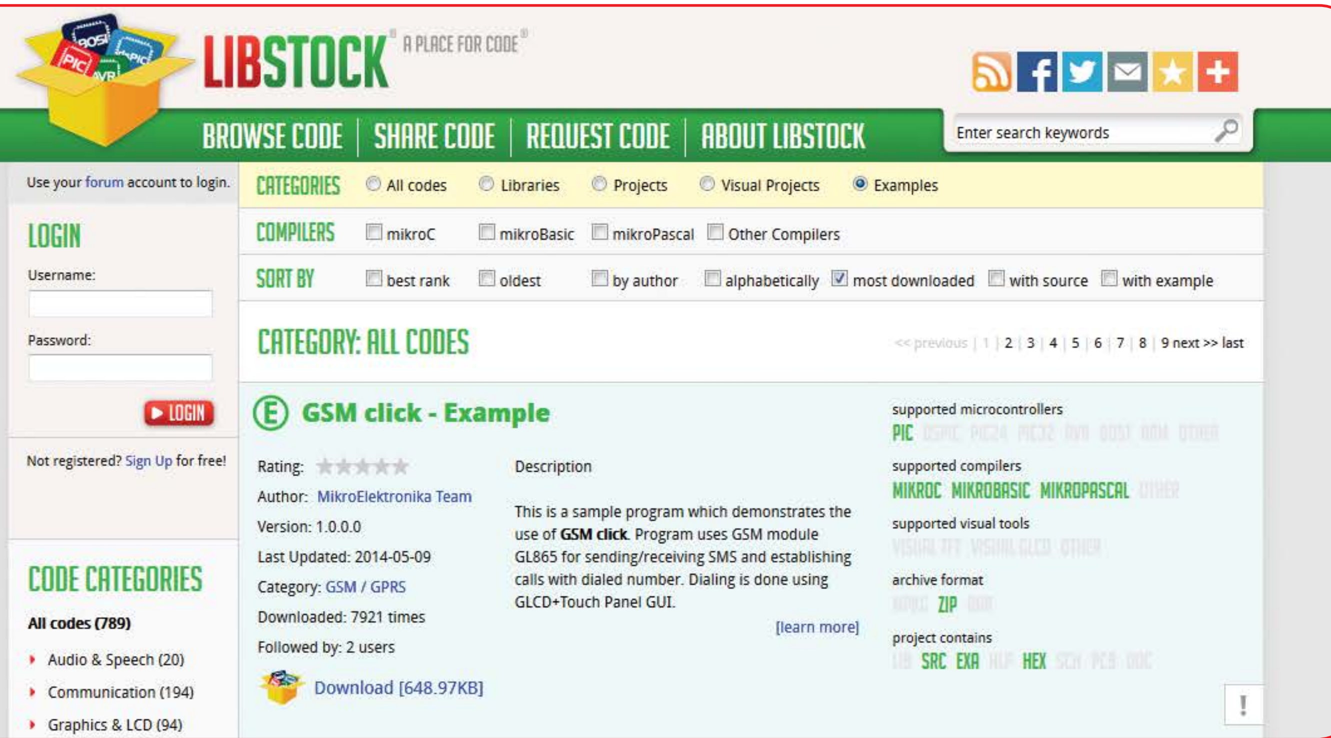
CAN SPI click™

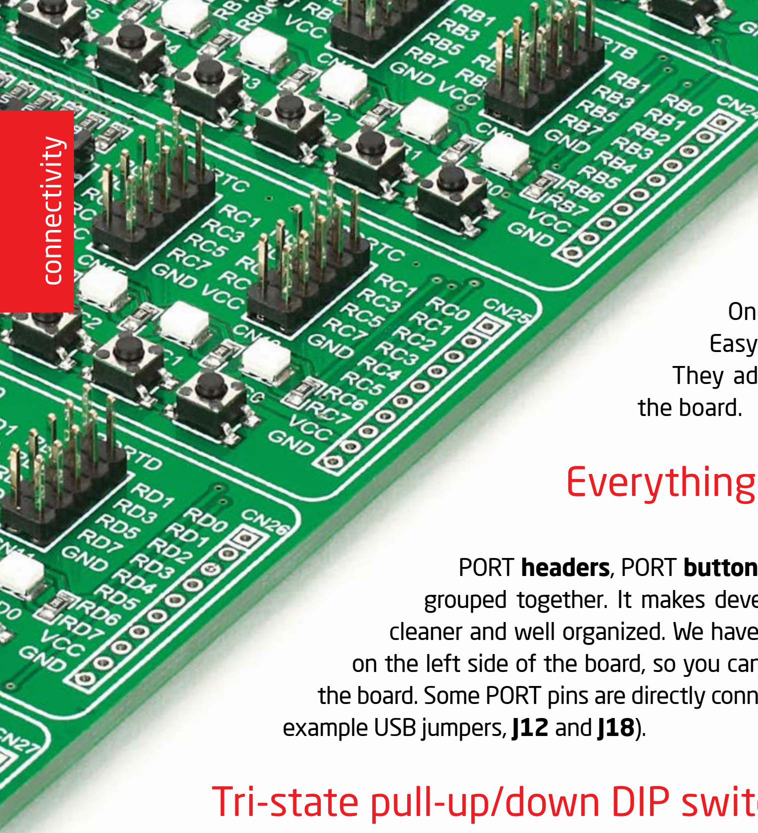
THERMO click™

Code Examples

It easy to get your Click™ board up and running. We provided the examples for mikroC™, mikroBasic™ and mikroPascal™ compilers on our **Libstock community website**. Just download them and you are ready to start:

http://www.libstock.com





Input/Output Group

One of the most distinctive features of EasyPIC™ v7 are it's Input/Output PORT groups. They add so much to the connectivity potential of the board.

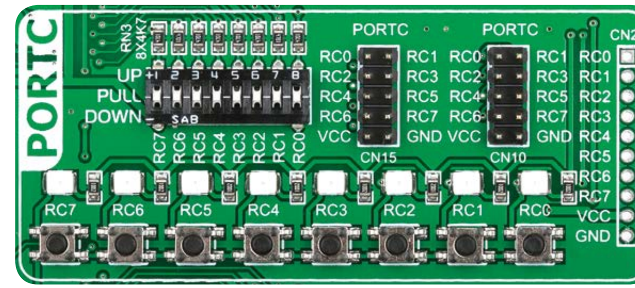


Figure 10-1: I/O group contains PORT headers, tri-state pull up/down DIP switch, buttons and LEDs all in one place

Everything is grouped together

PORT **headers**, PORT **buttons** and PORT **LEDs** are next to each other, and grouped together. It makes development easier, and the entire EasyPIC™ v7 cleaner and well organized. We have also provided an **additional PORT headers** on the left side of the board, so you can access any pin you want from both sides of the board. Some PORT pins are directly connected to the microcontroller, and some that are connected to other on-board modules are enabled via jumpers (for example USB jumpers, **J12** and **J18**).

Tri-state pull-up/down DIP switches

Tri-state DIP switches, like **SW7** on **Figure 10-2**, are used to enable 4K7 pull-up or pull-down resistor on any desired port pin. Each of these switches has three states:

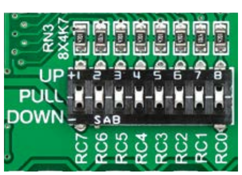
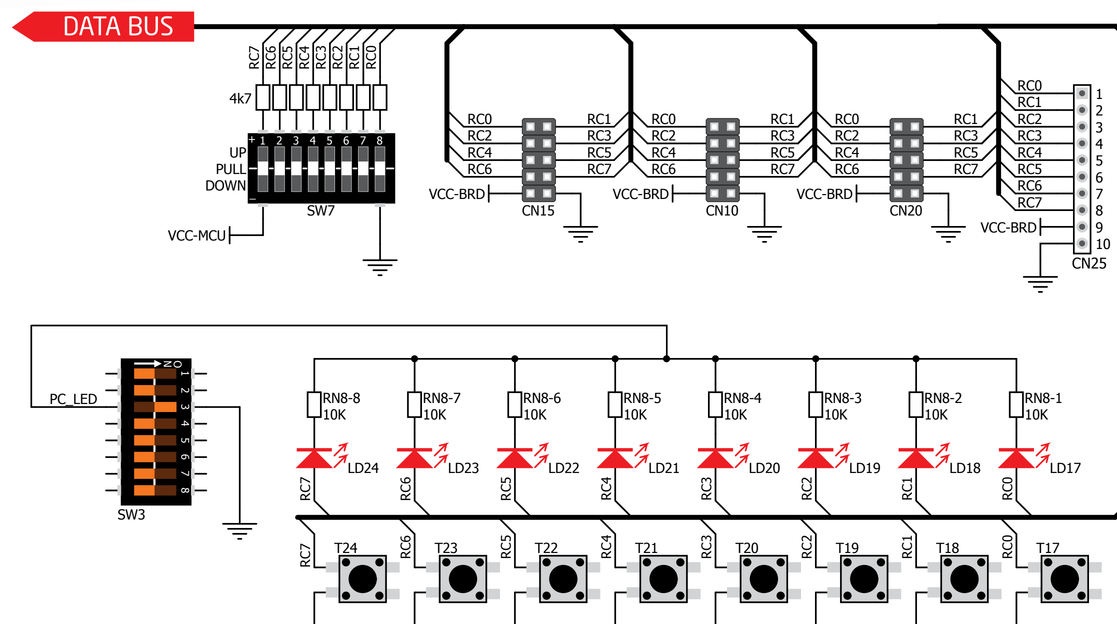


Figure 10-2: Tri-state DIP switch on PORTC

1. **middle position** disables both pull-up and pull-down feature from the PORT pin
2. **up position** connects the resistor in pull-up state to the selected pin
3. **down position** connects the resistor in pull-down state to the selected PORT pin.



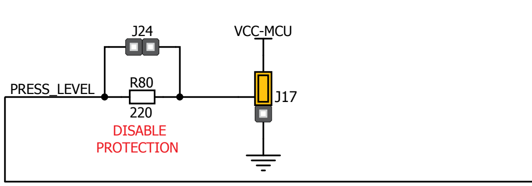
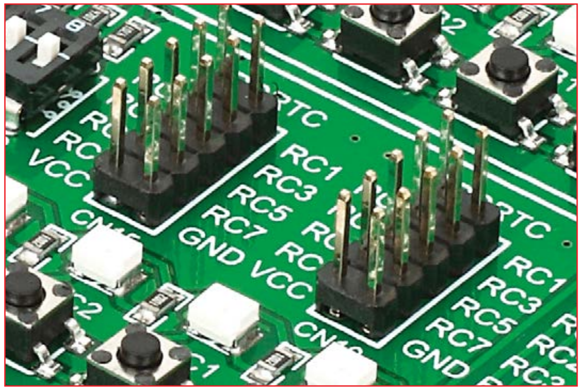
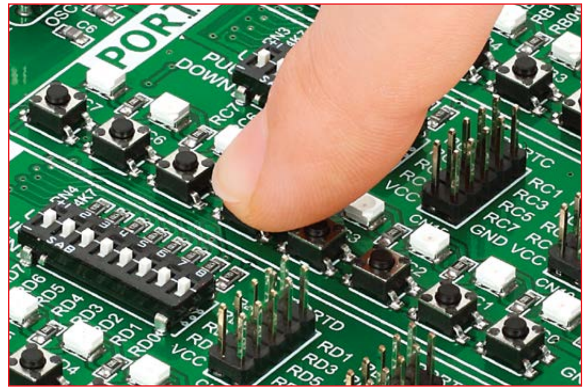
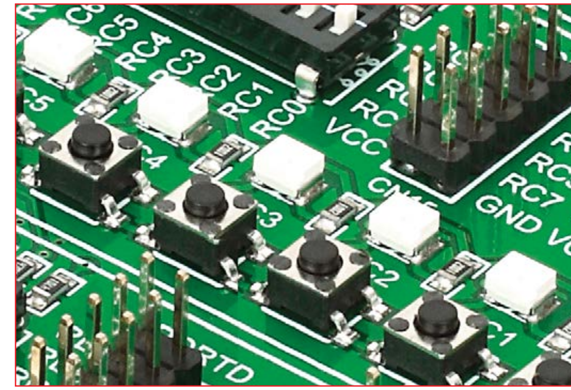


Figure 10-3: Schematic of the single I/O group connected to microcontroller PORTC







Headers

With enhanced connectivity as one of the key features of EasyPIC v7, we have provided **four connection headers for each PORT**. I/O PORT group contains two male IDC10 headers (like **CN10** and **CN15** on **Figure 10-3**). These headers are all compatible with over 70 MikroElektronika accessory boards, and enable simple connection. There is **one more IDC10 header** available on the left side of the board, next to the section with displays.

NOTE: Because of it's orientation, header on the left side of the board is not meant for placing accessory boards directly. Instead, use wire jumpers or other ways to establish connection and utilize these pins.

I/O PORT group also contains 1x10 connection pad (like **CN25** on **Figure 10-3**) which can be used for connecting MikroElektronika PROTO boards, or custom user boards.

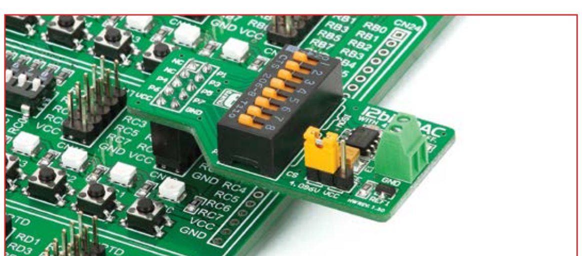


Figure 10-4: IDC10 male headers enable easy connection with MikroElektronika accessory boards

Buttons

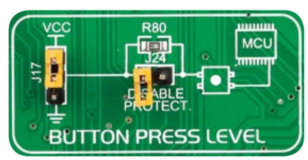


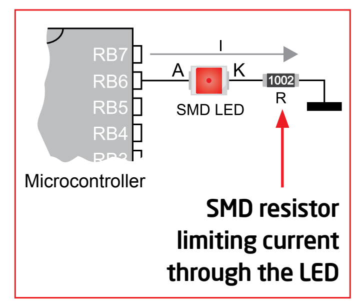
Figure 10-5: Button press level jumper (J17)

The logic state of all microcontroller digital inputs may be changed using **push buttons**. Jumper **J17** is available for selecting which logic state will be applied to corresponding MCU pin when button is pressed in any I/O port group. If you, for example, place **J17** in **VCC** position, then pressing of any push button in PORT I/O group will apply logic one to the appropriate microcontroller pin. The same goes for **GND**. If the jumper is taken out, then neither of two logic states will be applied to the appropriate microcontroller pin. You can disable pin protection 220ohm resistors by placing jumper **J24**, which will connect your push buttons directly to VCC or GND. Be aware that doing so you may accidentally damage MCU in case of wrong usage.

Reset Button

In the far upper right section of the board, there is a **RESET button**, which can be used to manually reset the microcontroller. This button is directly connected to the **MCLR pin**.

LEDs



LED (Light-Emitting Diode) is a highly efficient electronic light source. When connecting LEDs, it is necessary to place a current limiting resistor in series so that LEDs are provided with the current value specified by the manufacturer. The current varies from 0.2mA to 20mA, depending on the type of the LED and the manufacturer.. The EasyPIC™ v7 board uses low-current LEDs with typical current consumption of 0.2mA or 0.3mA, depending of VCC voltage selection. Board contains 36 LEDs which can be used for visual indication of the logic state on PORT pins. An active LED indicates that a logic high (1) is present on the pin. In order to enable PORT LEDs, it is necessary to enable the corresponding DIP switches on **SW3** (**Figure 10-6**).

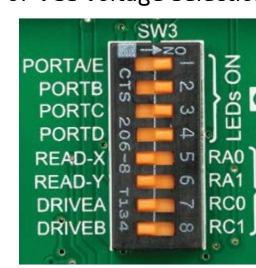
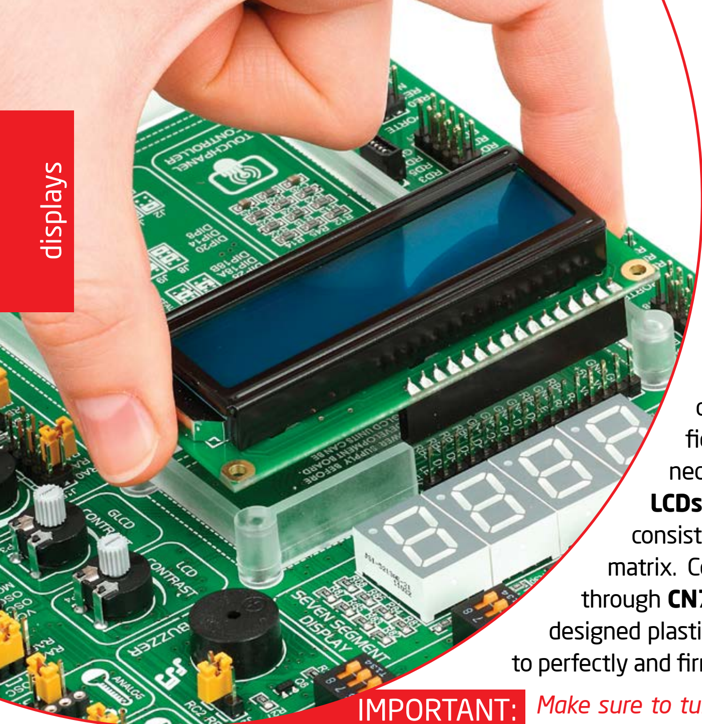


Figure 10-6: SW3.1 through SW3.4 switches are used to enable PORT LEDs



LCD 2x16 characters

Liquid Crystal Displays or **LCDs** are cheap and popular way of representing information to the end user of some electronic device. Character LCDs can be used to represent standard and custom characters in the predefined number of fields. EasyPIC™ v7 provides the connector and the necessary interface for supporting **2x16 character LCDs** in 4-bit mode. This type of display has two rows consisted of 16 character fields. Each field is a 7x5 pixel matrix. Communication with the display module is done through **CN7** display connector. Board is fitted with uniquely designed plastic display distancer, which allows the LCD module to perfectly and firmly fit into place.

IMPORTANT: *Make sure to turn off the power supply before placing LCD onto the board. Otherwise your display can be permanently damaged.*

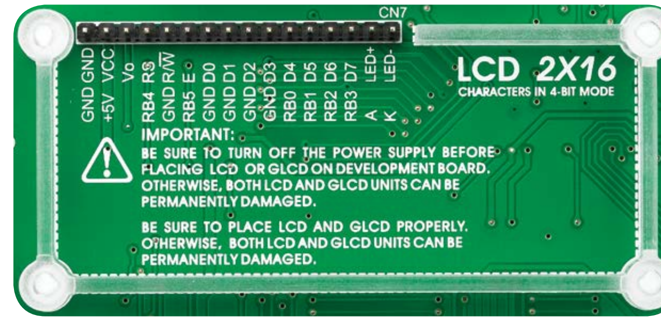


Figure 11-1: On-board LCD 2x16 display connector

Connector pinout explained

GND and VCC - Display power supply lines
Vo - LCD contrast level from potentiometer P4
RS - Register Select Signal line
E - Display Enable line
R/W - Determines whether display is in Read or Write mode. It's always connected to GND, leaving the display in Write mode all the time.
D0-D3 - Display is supported in 4-bit data mode, so lower half of the data byte interface is connected to GND.
D4-D7 - Upper half of the data byte
LED+ - Connection with the back-light LED anode
LED- - Connection with the back-light LED cathode

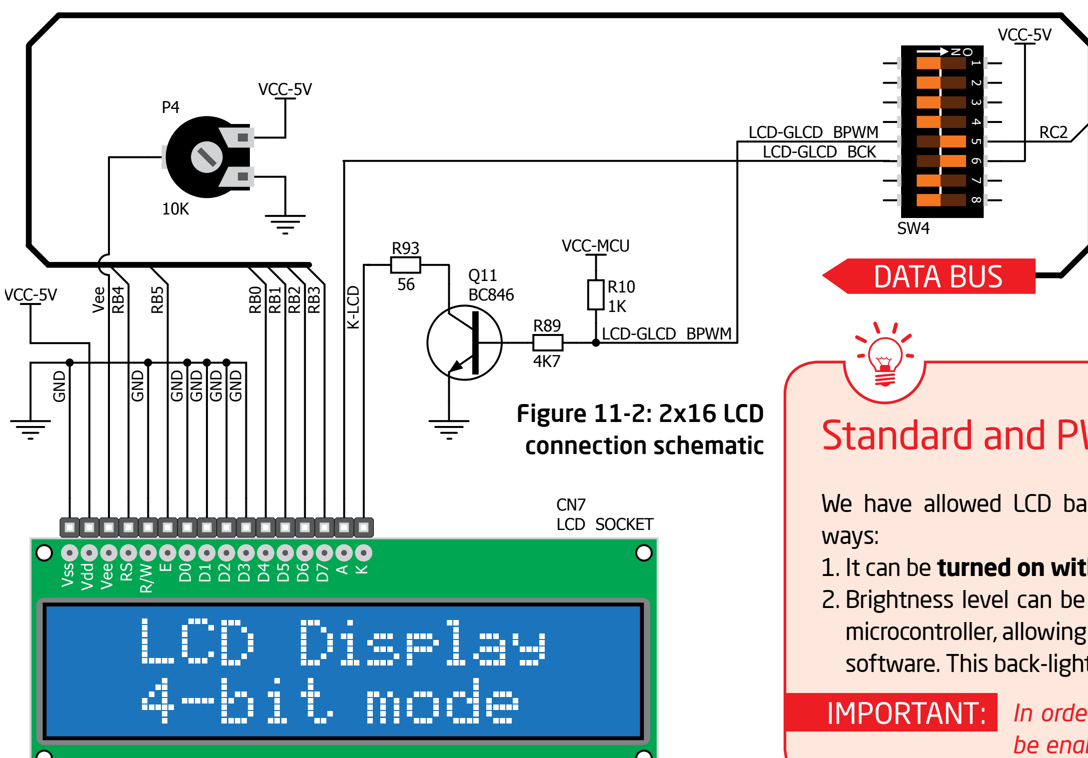
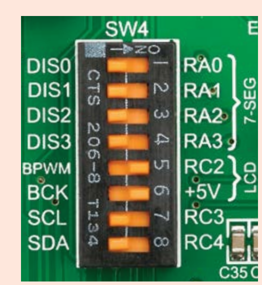


Figure 11-2: 2x16 LCD connection schematic

Standard and PWM-driven back-light

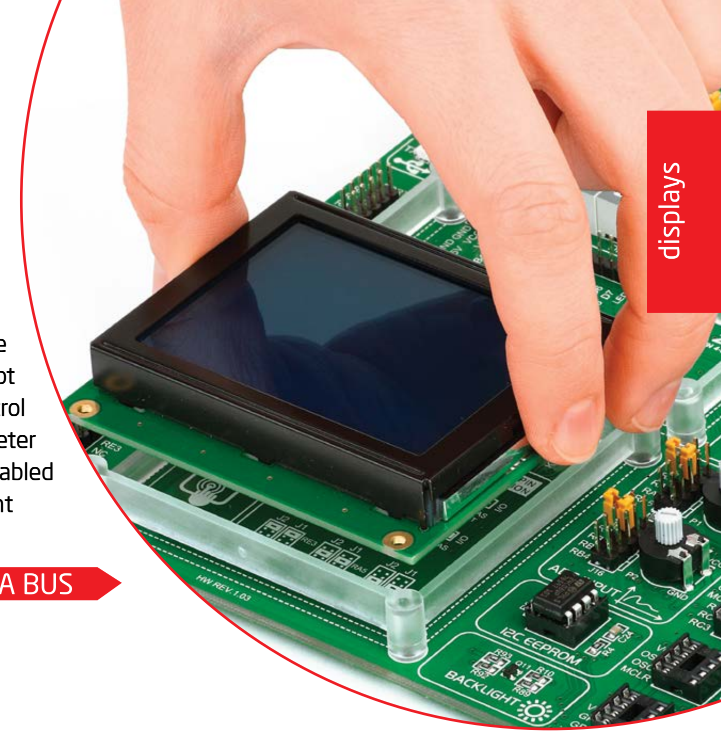
We have allowed LCD back-light to be enabled in two different ways:

1. It can be **turned on with full brightness** using **SW4.6** switch.
2. Brightness level can be determined **with PWM signal** from the microcontroller, allowing you to write custom back-light controlling software. This back-light mode is enabled with **SW4.5** switch.



IMPORTANT: *In order to use PWM back-light both **SW4.5** and **SW4.6** switches must be enabled at the same time.*

GLCD 128x64



Graphical Liquid Crystal Displays, or **GLCDs** are used to display monochromatic graphical content, such as text, images, human-machine interfaces and other content. EasyPIC™ v7 provides the connector and necessary interface for supporting GLCD with resolution of 128x64 pixels, driven by the KS108 or compatible display controller. Communication with the display module is done through **CN6** display connector. Board is fitted with uniquely designed plastic display distancer, which allows the GLCD module to perfectly and firmly fit into place.

Display connector is routed to PORTB (control lines) and PORTD (data lines) of the microcontroller sockets. Since the same ports are used by 2x16 character LCD display, you cannot use both displays simultaneously. You can control the display contrast using dedicated potentiometer **P3**. Full brightness display back light can be enabled with **SW4.6** switch, and PWM-driven back light with **SW4.5** switch.

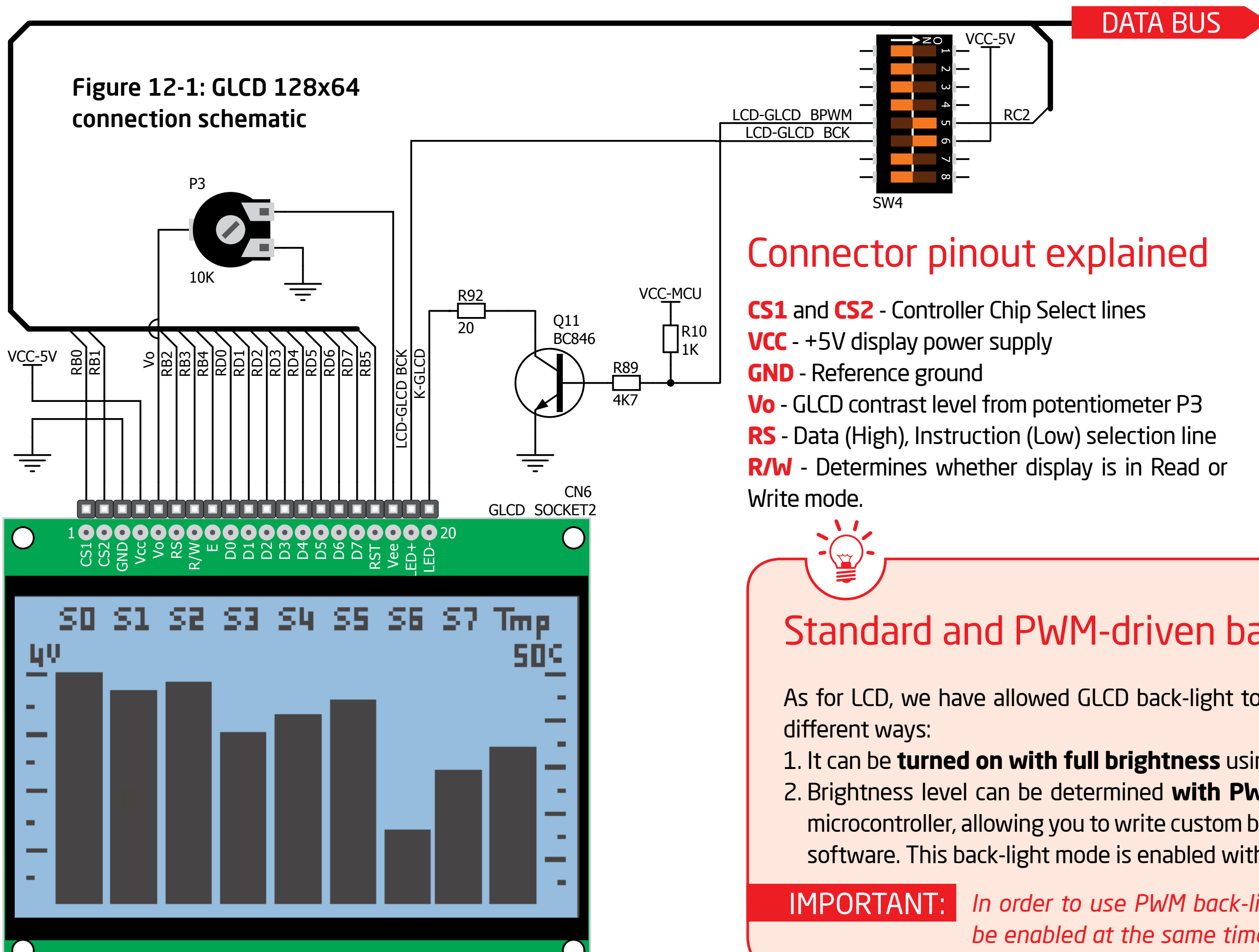


Figure 12-1: GLCD 128x64 connection schematic

Connector pinout explained

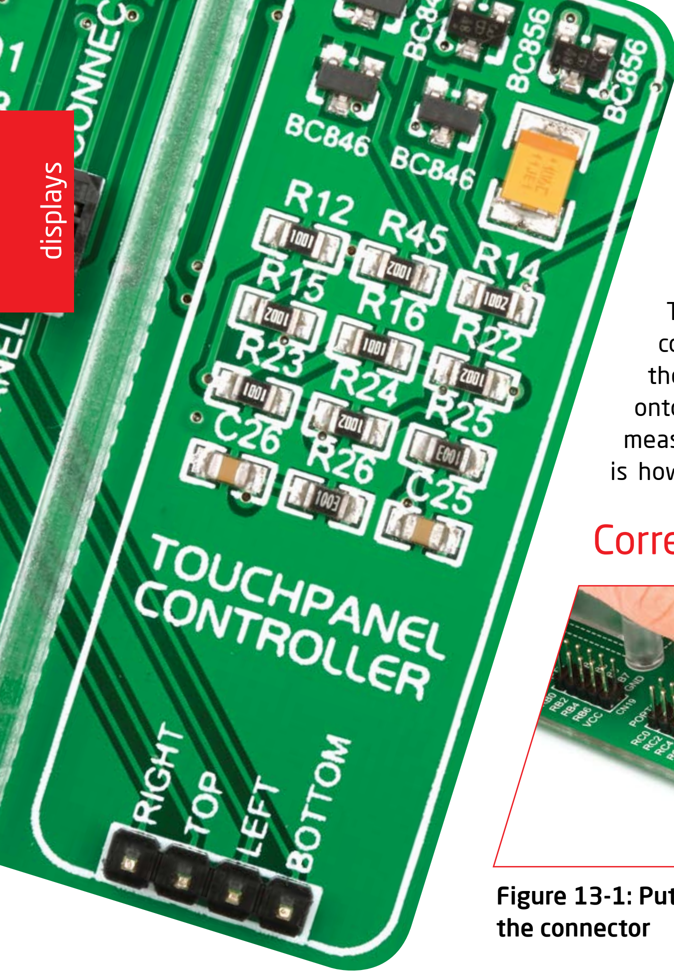
CS1 and **CS2** - Controller Chip Select lines
VCC - +5V display power supply
GND - Reference ground
Vo - GLCD contrast level from potentiometer P3
RS - Data (High), Instruction (Low) selection line
R/W - Determines whether display is in Read or Write mode.
E - Display Enable line
D0-D7 - Data lines
RST - Display reset line
Vee - Reference voltage for GLCD contrast potentiometer P3
LED+ - Connection with the back-light LED anode
LED- - Connection with the back-light LED cathode

Standard and PWM-driven back-light

As for LCD, we have allowed GLCD back-light to be enabled in two different ways:

1. It can be **turned on with full brightness** using **SW4.6** switch.
2. Brightness level can be determined **with PWM signal** from the microcontroller, allowing you to write custom back-light controlling software. This back-light mode is enabled with **SW4.5** switch.

IMPORTANT: *In order to use PWM back-light both* ***SW4.5*** *and* ***SW4.6*** *switches must be enabled at the same time.*



Touch panel controller

Touch panel is a glass panel whose surface is covered with two layers of resistive material. When the screen is pressed, the outer layer is pushed onto the inner layer and appropriate controllers can measure that pressure and pinpoint its location. This is how touch panels can be used as an input devices.

EasyPIC™ v7 is equipped with touch panel controller and connector for **4-wire resistive touch panels**. It can very accurately register pressure at a specific point, representing the touch coordinates in the form of analog voltages, which can then be easily converted to X and Y values. Touch panel comes as a part of display.

Correctly placing the touch panel cable into the connector

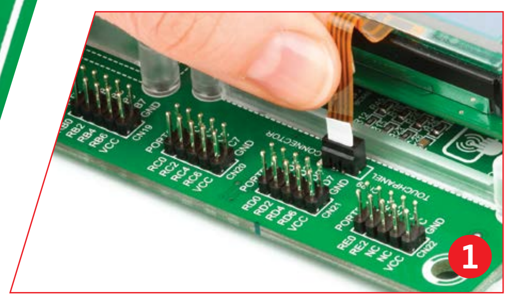


Figure 13-1: Put Touch panel flat cable in the connector

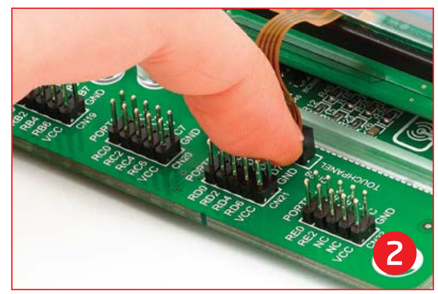


Figure 13-2: Use a tip of your finger to push it inside

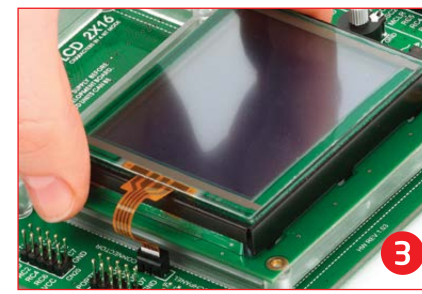


Figure 13-3: Now place GLCD with Touch panel into GLCD socket

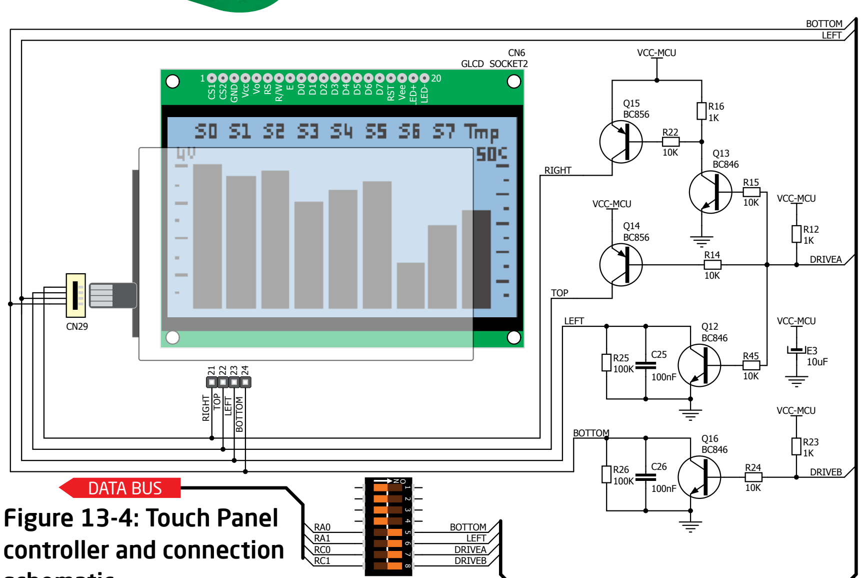


Figure 13-4: Touch Panel controller and connection schematic

Enabling Touch panel

Touch panel is enabled using **SW3.5**, **SW3.6**, **SW3.7** and **SW3.8** switches. They connect **READ-X** and **READ-Y** lines of the touch panel with **RA0** and **RA1** analog inputs, and **DRIVEA** and **DRIVEB** with **RC0** and **RC1** digital outputs on microcontroller sockets. Make sure to disconnect other peripherals, LEDs and additional pull-up or pull-down resistors from the interface lines in order not to interfere with signal/data integrity.

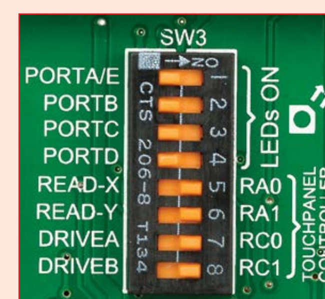
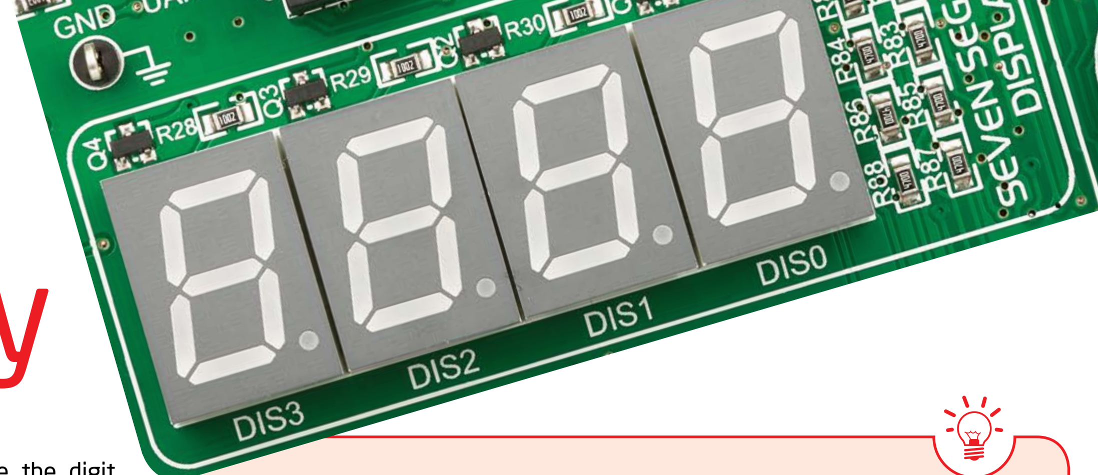


Figure 13-5: Turn on switches 5 through 8 on SW3 to enable Touch panel controller

4 digit 7-seg display



One seven segment digit consist of 7+1 LEDs which are arranged in a specific formation which can be used to represent digits from 0 to 9 and even some letters. One additional LED is used for marking the decimal dot, in case you want to write a decimal point in the desired segment. EasyPIC™ v7 contains four of these digits put together to form 4-digit 7-segment display. Driving such a display is done using **multiplexing techniques.** Data lines are shared between segments, and therefore the same segment LEDs in each digit are connected in parallel. Each digit has it's **unique digit select line**, which is used to enable the digit to which the data is currently being sent. By multiplexing data through all four segments fast enough, you create an illusion that all four segments are in operation simultaneously. This is possible because human eye has a slower reaction time than the mention changes. This way you can represent numbers in decimal or hexadecimal form. Eight data lines that are common for all the digits are connected to PORTD, and digit select lines are connected to RA0-RA3 lines on the microcontroller sockets.

Enabling the display

To enable digit select lines for the 4-digit 7-segment display you have to turn on **SW4.1**, **SW4.2**, **SW4.3** and **SW4.4** switches. Digit select lines are connected to **RA0 - RA3** pins on the microcontroller sockets, while data lines are connected to **RD0 - RD7** pins. Make sure to disconnect other peripherals from the interface lines in order not to interfere with signal/data integrity.

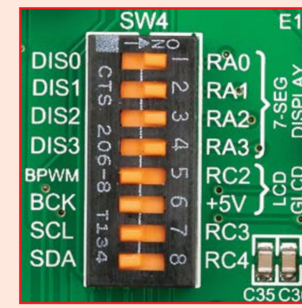


Figure 14-1: Turn on switches 1 through 4 on SW4 to enable 4-digit 7-seg display

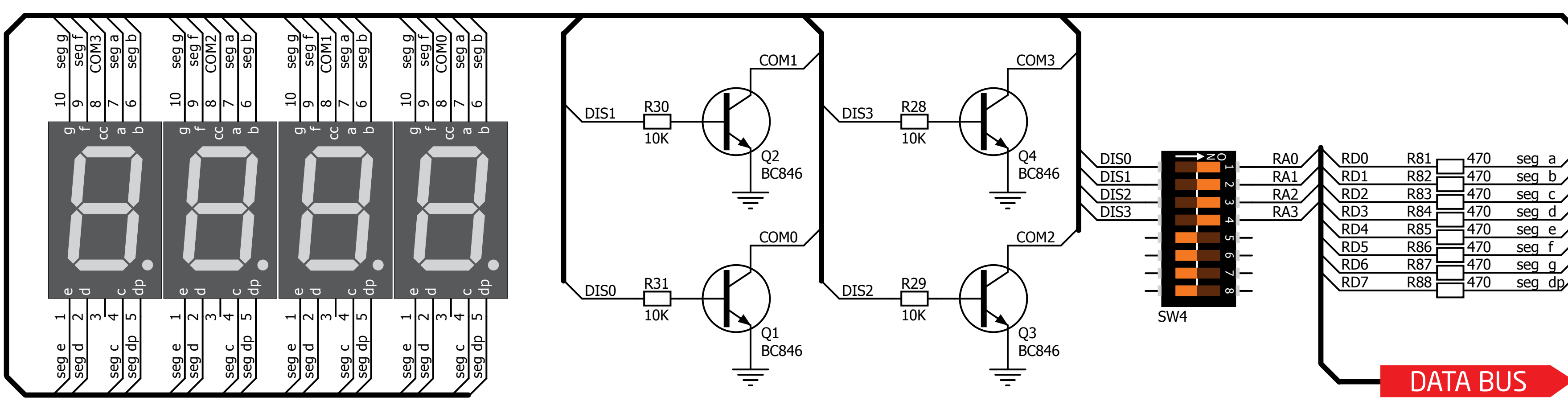
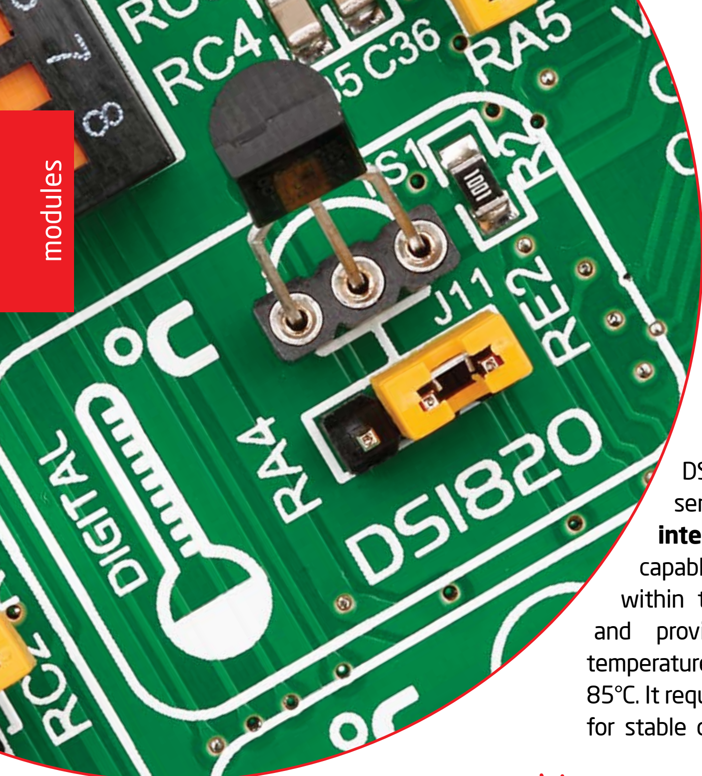


Figure 14-2: 4-digit 7-segment display schematic



DS1820 - Digital Temperature Sensor

DS1820 is a digital temperature sensor that uses **1-wire**® **interface** for it's operation. It is capable of measuring temperatures within the range of -55 to 128°C, and provides ±0.5°C accuracy for temperatures within the range of -10 to 85°C. It requires 3V to 5.5V power supply for stable operation. It takes maximum of 750ms for the DS1820 to calculate temperature with 9-bit resolution. **1-wire**® **serial communication** enables data to be transferred over a 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. Multiple sensors can be connected on the same line. All slave devices by default have a unique ID code, which enables the master device to easily identify all devices sharing the same interface. EasyPIC™ v7 provides a separate socket (**TS1**) for the DS1820. Communication line with the microcontroller is connected via jumper **J11**.

Enabling DS1820 Sensor

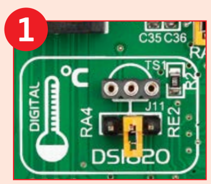


Figure 15-1: DS1820 not connected

Figure 15-2: DS1820 placed in socket

Figure 15-3: DS1820 connected to RE2 pin

Figure 15-4: DS1820 connected to RA4 pin

EasyPIC™ v7 enables you to establish 1-wire® communication between **DS1820** and the microcontroller via **RA4** or **RE2** microcontroller pins. The selection of either of those two lines is done using **J11** jumper. When placing the sensor in the socket make sure that half-circle on the board's silkscreen markings matches the rounded part of the DS1820 sensor. If you accidentally connect the sensor the other way, it may be permanently damaged. Make sure to disconnect other peripherals (except 1-wire), LEDs and additional pull-up or pull-down resistors from the interface lines in order not to interfere with signal/data integrity.

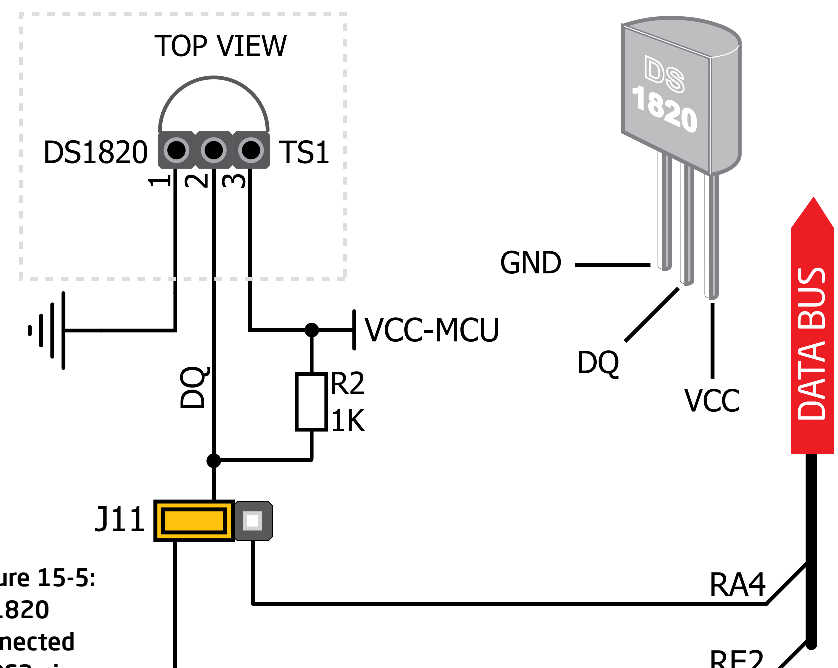
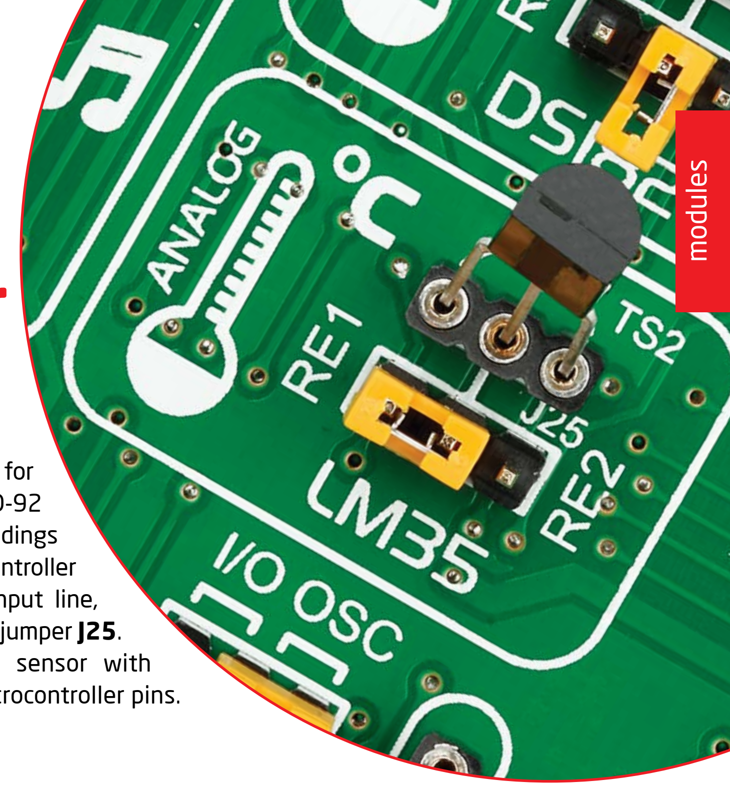


Figure 15-5: DS1820 connected to RE2 pin

LM35 - Analog Temperature Sensor



The **LM35** is a low-cost precision integrated-circuit temperature sensor, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. It has a linear +10.0 mV/°C scale factor and less than 60 μA current drain. As it draws only 60 μA from its supply, it has very low self-heating, less than 0.1°C in still air. EasyPIC™ v7 enables you to get analog readings from the LM35 sensor in restricted temperature range from +2°C to +150°C. Board provides a separate socket (**TS2**) for the LM35 sensor in TO-92 plastic packaging. Readings are done with microcontroller using single analog input line, which is selected with jumper **J25**. Jumper connects the sensor with either **RE2** or **RE1** microcontroller pins.

Enabling LM35 Sensor

Figure 16-1: LM35 not connected

Figure 16-2: LM35 placed in socket

Figure 16-3: LM35 connected to RE1 pin

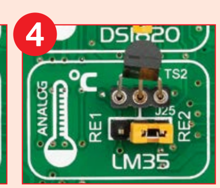


Figure 16-4: LM35 connected to RE2 pin

EasyPIC™ v7 enables you to get analog readings from the LM35 sensor using **RE1** or **RE2** microcontroller pins. The selection of either of those two lines is done using **J25** jumper. When placing the sensor in the socket make sure that half-circle on the board's silkscreen markings matches the rounded part of the LM35 sensor. If you accidentally connect the sensor the other way, it can be permanently damaged and you might need to replace it with another one. During the readings of the sensor, make sure that no other device uses the selected analog line, because it may interfere with the readings.

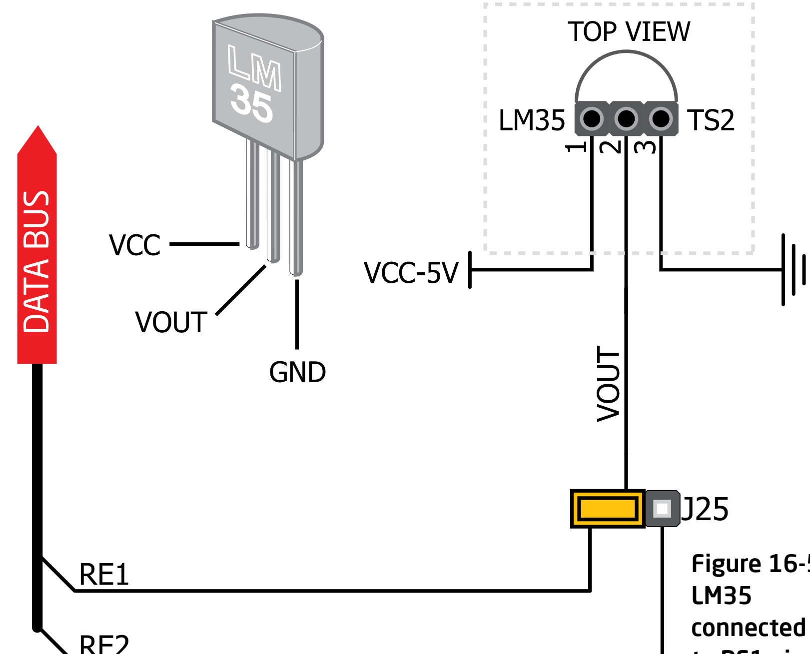
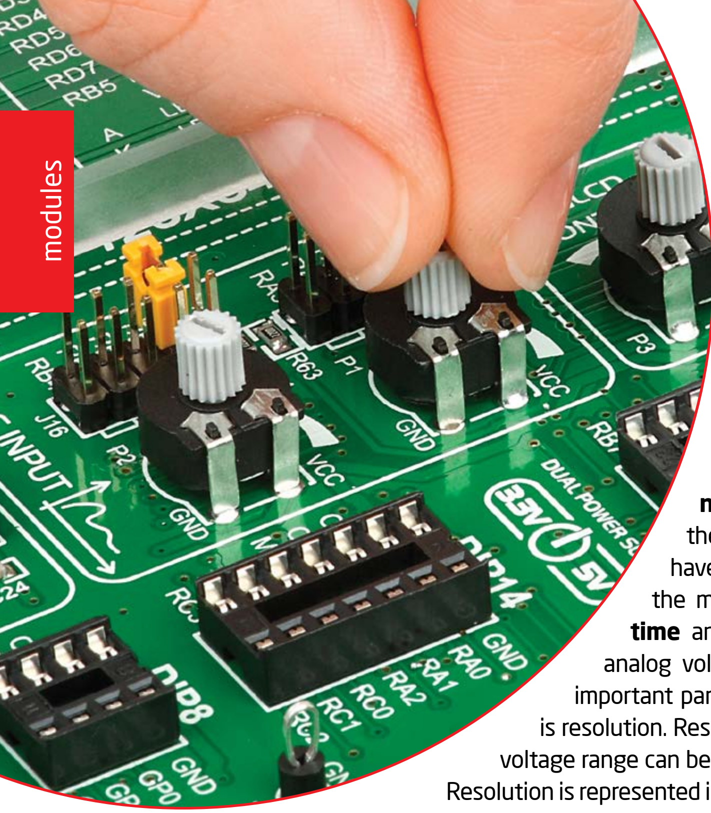


Figure 16-5: LM35 connected to RE1 pin



ADC inputs

Digital signals have **two discrete states**, which are decoded as **high** and **low**, and interpreted as **logic 1** and **logic 0**. **Analog signals**, on the other hand, are **continuous**, and can have any value within defined range. **A/D converters** are specialized circuits which can convert analog signals (voltages) into a digital representation, usually in form of an **integer number**. The value of this number is **linearly dependent** on the input voltage value. Most microcontrollers nowadays internally have A/D converters connected to one or more input pins. Some of the most important parameters of A/D converters are **conversion time** and **resolution**. Conversion time determines how fast can an analog voltage be represented in form of a digital number. This is an important parameter if you need fast data acquisition. The other parameter is resolution. Resolution represents the number of discrete steps that supported voltage range can be divided into. It determines the sensitivity of the A/D converter. Resolution is represented in maximum number of bits that resulting number occupies. Most PIC® microcontrollers have 10-bit resolution, meaning that maximum value of conversion can be represented with 10 bits, which converted to integer is 2^{10}=1024. This means that supported voltage range, for example from 0-5V, can be divided into 1024 discrete steps of about 4.88mV.

EasyPIC™ v7 provides an interface in form of two potentiometers for simulating analog input voltages that can be routed to any of the 10 supported analog input pins.

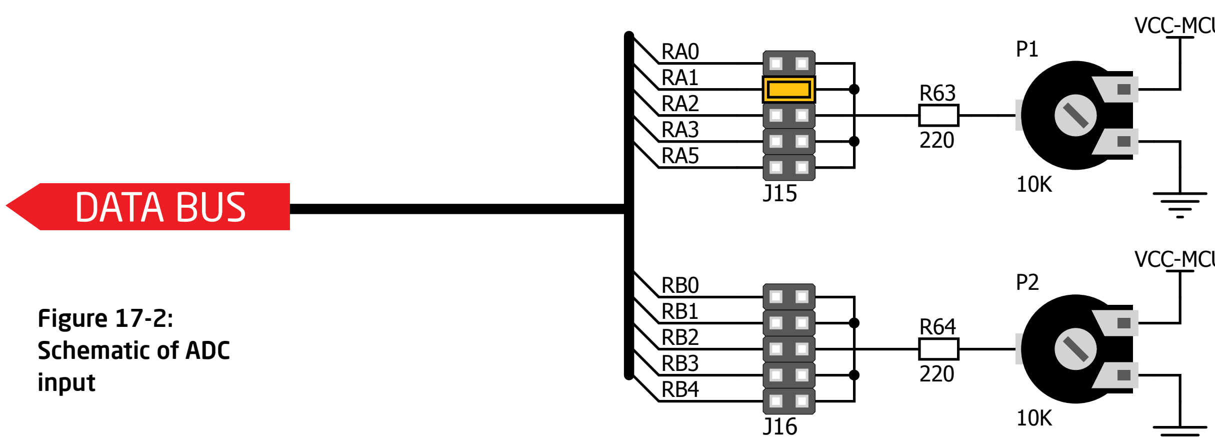


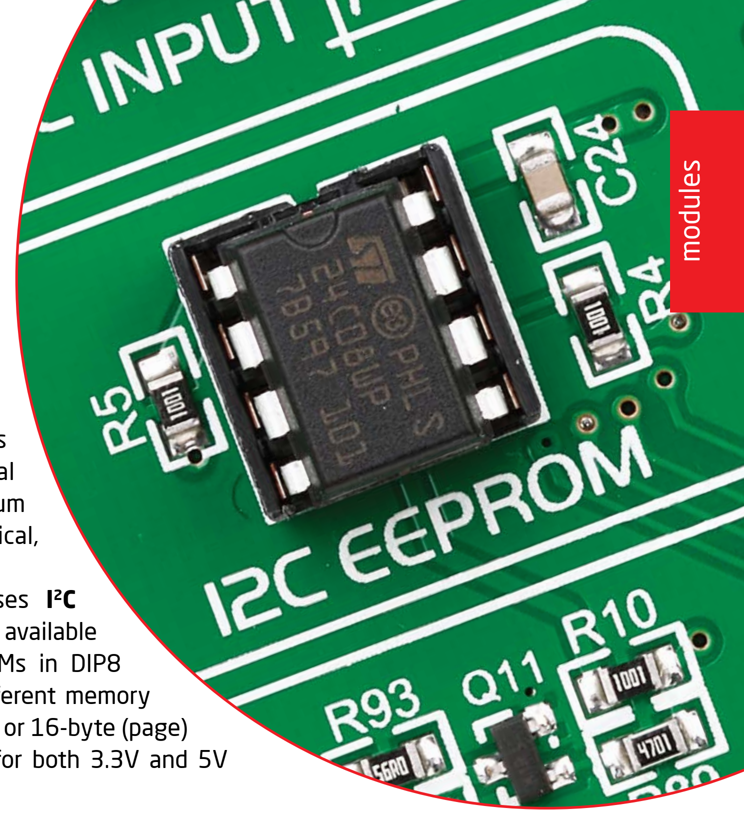
Figure 17-2: Schematic of ADC input

Enabling ADC inputs

Figure 17-1: use J15 and J16 jumpers to connect analog input lines with potentiometers P1 and P2

In order to connect the output of the potentiometer **P1** to **RA0**, **RA1**, **RA2**, **RA3** or **RA5** analog microcontroller inputs, you have to place the jumper **J15** in the desired position. If you want to connect potentiometer **P2** to any of the **RB0 - RB4** analog microcontroller inputs, place jumper **J16** in the desired position. By moving the potentiometer knob, you can create voltages in range from **GND** to **VCC**.

I²C EEPROM



Enabling I²C EEPROM

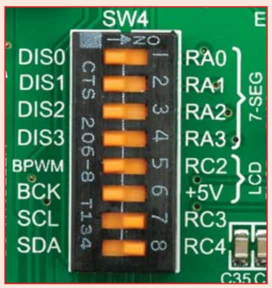


Figure 18-1: Activate SW4.7 and SW4.8 switches to connect microcontroller I²C lines to Serial EEPROM.

In order to connect I²C EEPROM to the microcontroller you must enable **SW4.7** and **SW4.8** switches, as shown on **Figure 18-1**. **1kΩ** pull-up resistors necessary for I²C communication are already provided on **SDA** and **SCL** lines once switches are turned on. Prior to using EEPROM in your application, make sure to disconnect other peripherals, LEDs and additional pull-up or pull-down resistors from the interface lines in order not to interfere with signal/data integrity.

EEPROM is short for **Electrically Erasable Programmable Read Only Memory**. It is usually a secondary storage memory in devices containing data that is retained even if the device looses power supply. Because of the ability to alter single bytes of data, EEPROM devices are used to store personal preference and configuration data in a wide spectrum of consumer, automotive, telecommunication, medical, industrial, and PC applications.

EasyPIC™ v7 supports serial EEPROM which uses **I²C communication interface** and has **1024 bytes** of available memory. Board contains socket for serial EEPROMs in DIP8 packaging, so you can easily exchange it with different memory size EEPROM IC. EEPROM itself supports single byte or 16-byte (page) write and read operations. Data rate is 400 kHz for both 3.3V and 5V power supply.

What is I²C?

I²C is a multi-master serial single-ended bus that is used to attach low-speed peripherals to computer or embedded systems. I²C uses only two open-drain lines, **Serial Data Line (SDA)** and **Serial Clock (SCL)**, pulled up with resistors. **SCL** line is driven by a master, while **SDA** is used as bidirectional line either by master or slave device. Up to 112 slave devices can be connected to the same bus. Each slave must have a unique address.

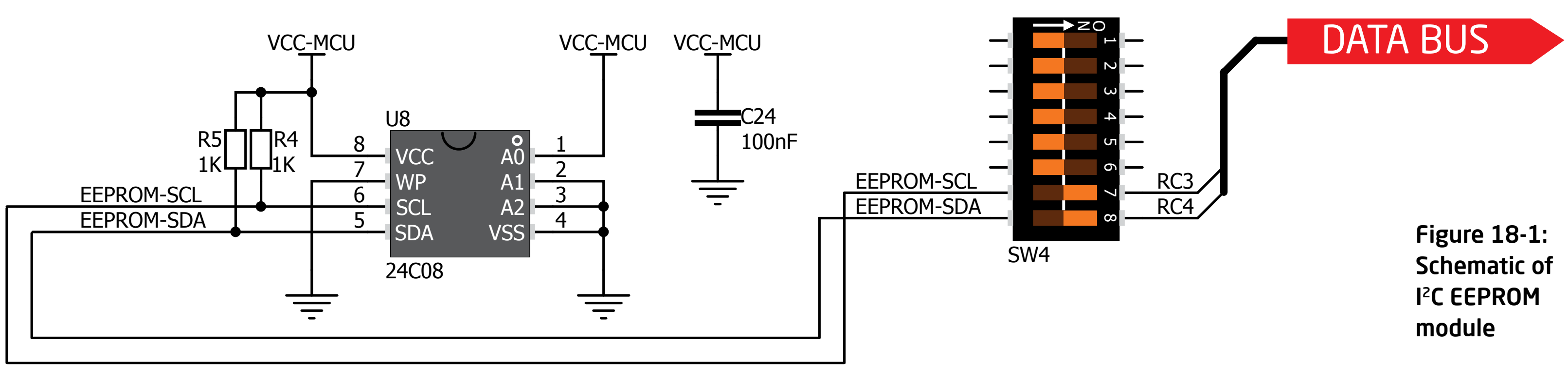
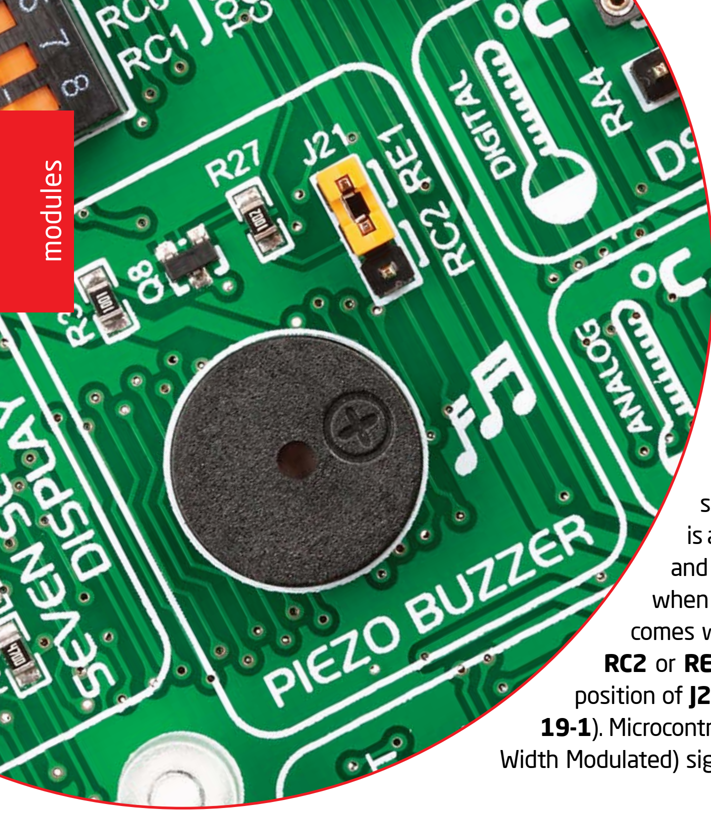


Figure 18-1: Schematic of I²C EEPROM module



Piezo Buzzer

Piezo electricity is the charge which accumulates in certain solid materials in response to mechanical pressure, but also providing the charge to the piezoelectric material causes it to physically deform. One of the most widely used applications of piezo electricity is the production of sound generators, called piezo buzzers. **Piezo buzzer** is an electric component that comes in different shapes and sizes, which can be used to create sound waves when provided with analog electrical signal. EasyPIC™ v7 comes with piezo buzzer which can be connected either to **RC2** or **RE1** microcontroller pins, which is determined by the position of **J21** jumper. Buzzer is driven by transistor **Q8** (**Figure 19-1**). Microcontrollers can create sound by generating a PWM (Pulse Width Modulated) signal – a **square wave** signal, which is nothing more than a sequence of logic zeros and ones. Frequency of the square signal determines the pitch of the generated sound, and duty cycle of the signal can be used to increase or decrease the volume in the range from 0% to 100% of the duty cycle. You can generate PWM signal using hardware capture-compare module, which is usually available in most microcontrollers, or by writing a custom software which emulates the desired signal waveform.

Supported sound frequencies

Piezo buzzer's resonant frequency (where you can expect it's best performance) is **3.8kHz**, but you can also use it to create sound in the range between **2kHz** and **4kHz**.

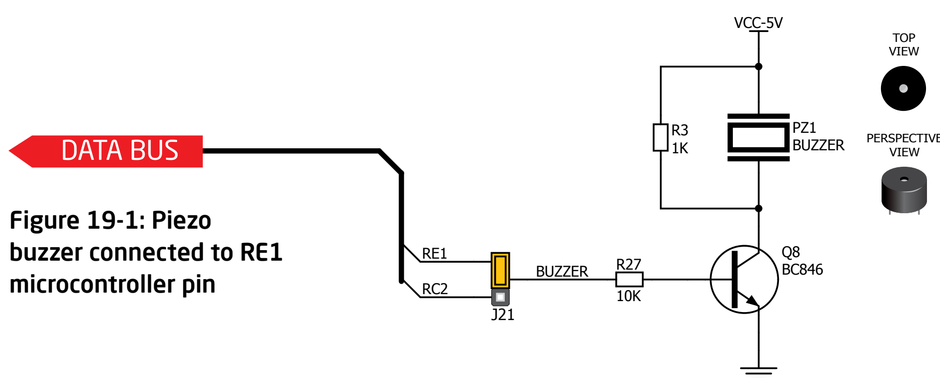
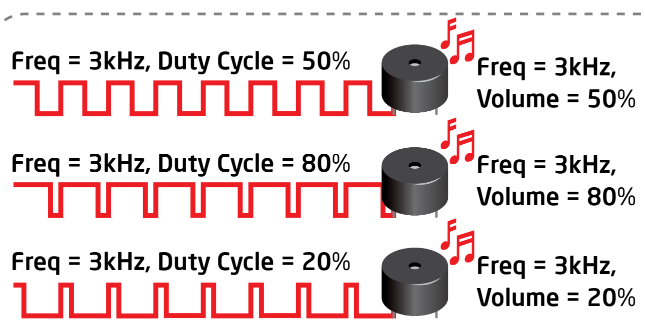


Figure 19-1: Piezo buzzer connected to RE1 microcontroller pin



How to make it sing?

Buzzer starts "singing" when you provide PWM signal from the microcontroller to the buzzer driver. The pitch of the sound is determined by the frequency, and amplitude is determined by the duty cycle of the PWM signal.

Enabling Piezo Buzzer

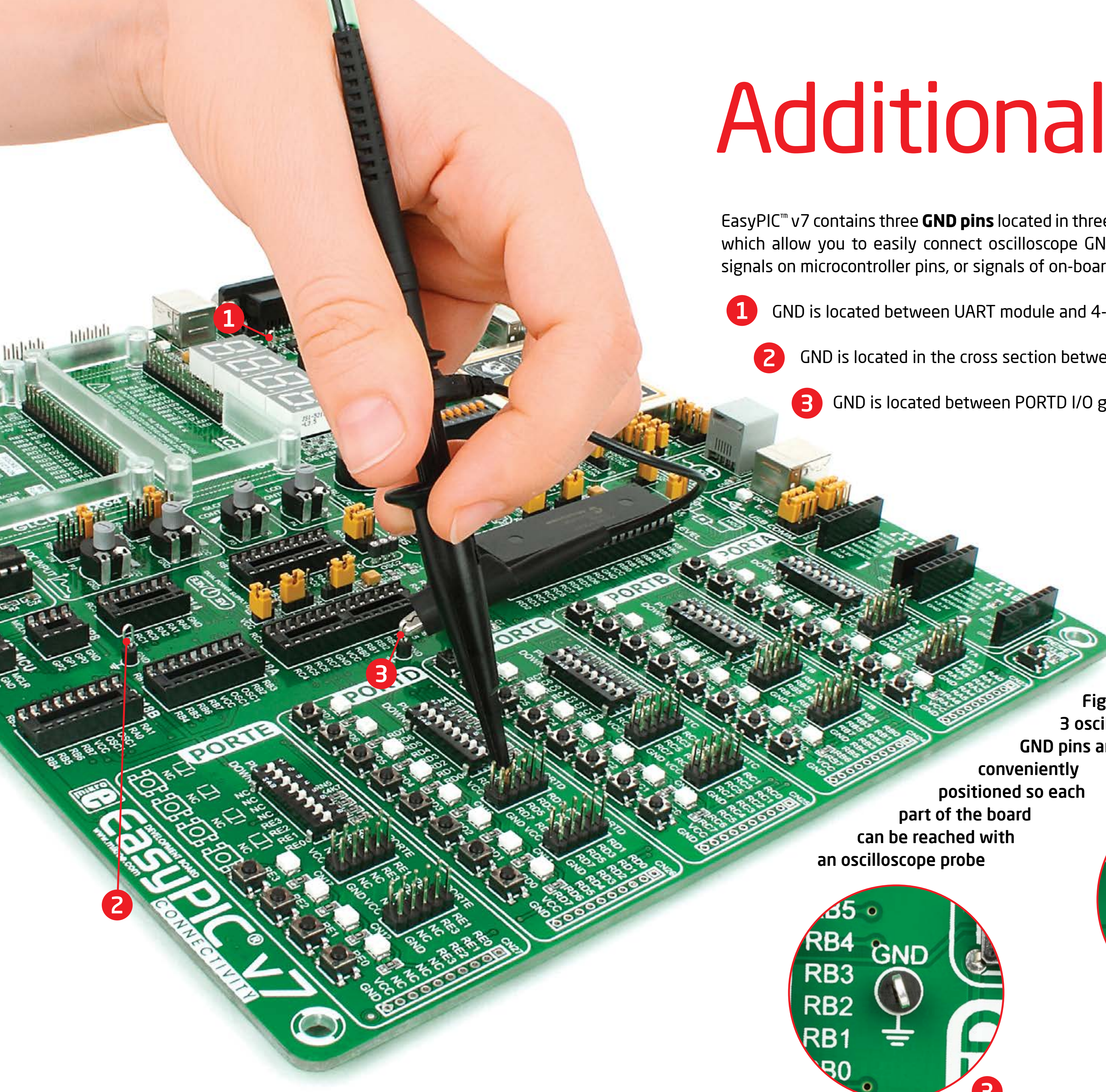
In order to use the on-board Piezo Buzzer in your application, you first have to connect the transistor driver of piezo buzzer to the appropriate microcontroller pin. This is done using jumper **J21**. You can place the jumper in two positions, thus connecting the buzzer driver to either **RE1** or **RC2** microcontroller pin.

Figure 19-2: Use jumper J12 to connect Piezo buzzer on RE1 or RC2 pin

Additional GNDs

EasyPIC™ v7 contains three **GND pins** located in three different sections of the board, which allow you to easily connect oscilloscope GND reference when you monitor signals on microcontroller pins, or signals of on-board modules.

1. GND is located between UART module and 4-digit 7-seg display.
2. GND is located in the cross section between DIP18 and DIP14 sockets
3. GND is located between PORTD I/O group and DIP28 socket.



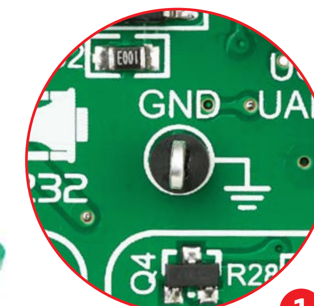


Figure 20-1: 3 oscilloscope GND pins are conveniently positioned so each part of the board can be reached with an oscilloscope probe

What's Next?

You have now completed the journey through each and every feature of **EasyPIC™ v7 board**. You got to know it's modules, organization, supported microcontrollers, programmer and debugger. Now you are ready to start using your new board. We are suggesting several steps which are probably the best way to begin. We invite you to join thousands of users of EasyPIC™ brand. You will find very useful projects and tutorials and can get help from a large ecosystem of users. Welcome!

1 Compiler

You still don't have an appropriate compiler? Locate **PIC® compiler** that suits you best on our website:

www.mikroe.com/pic/compilers/

Choose between **mikroC™**, **mikroBasic™** and **mikroPascal™**, and download fully functional demo version, so you can begin building your PIC® applications.

2 Projects

Once you have chosen your compiler, and since you already got the board, you are ready to start writing your first projects. We have equipped our compilers with dozens of examples that demonstrate the use of each and every feature of the EasyPIC™ board, and all of our accessory boards as well. This makes an excellent starting point for your future projects. Just load the example, read well commented code, and see how it works on hardware. Browse through the compiler **Examples** available on this link:

www.mikroe.com/easypic/

3 Community

If you want to find answers to your questions on many interesting topics we invite you to visit our forum at **http://www.mikroe.com/forum** and browse through more than 150 thousand posts. You are likely to find just the right information for you. On the other hand, if you want to download free projects and libraries, or share your own code, please visit the **Libstock™ website**. With user profiles, you can get to know other programmers, and subscribe to receive notifications on their code.

www.libstock.com/

4 Support

We all know how important it is that we can rely on someone in moments when we are stuck with our projects, facing a deadline, or when we just want to ask a simple, basic question, that's pulling us back for a while. We do understand how important this is to people and therefore our Support Department is one of the pillars upon which our company is based. MikroElektronika offers **Free Tech Support** to the end of product lifetime, so if something goes wrong, we are ready and willing to help!

www.mikroe.com/support/

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, must 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™, mikroBasic™, mikroPascal™, mikroProg™, EasyPIC™, EasyPIC PRO™, mikroBUS™, mikromedia™, MINI™ and Click boards™ 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/support/
If you have any questions, comments or business proposals,
do not hesitate to contact us at office@mikroe.com

EasyPIC v7 User Manual
ver. 1.04c
0 100000 025925