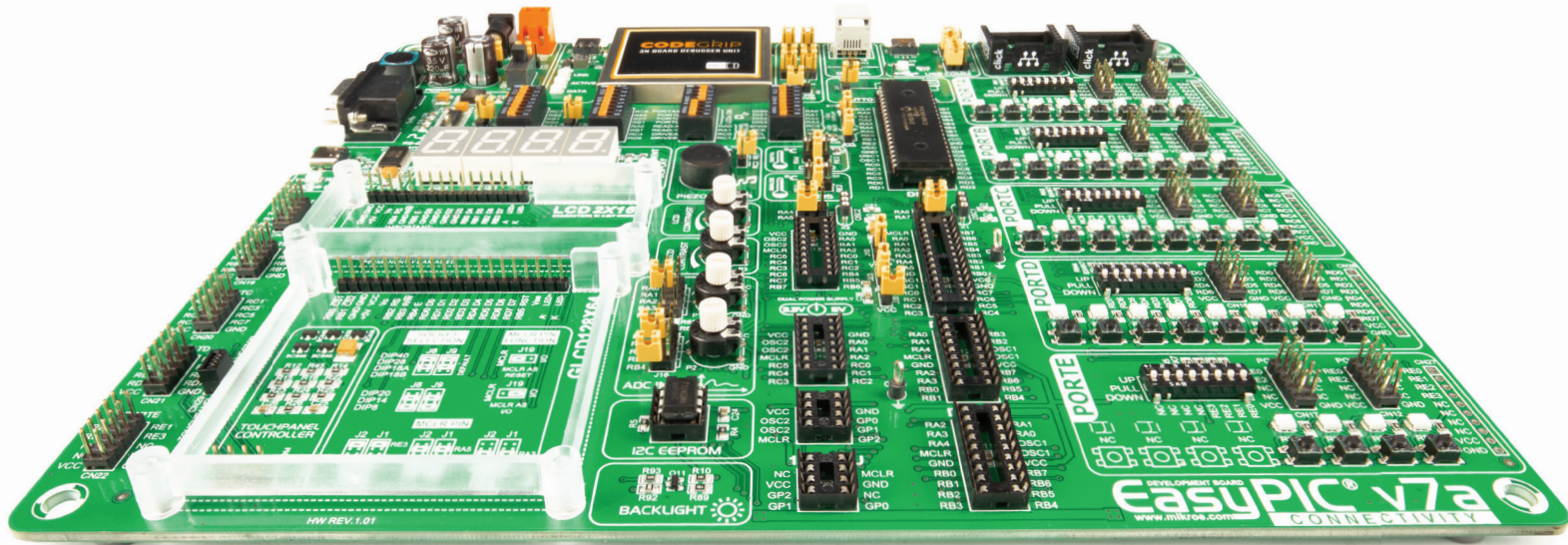


EasyPIC v7a

CONNECTIVITY



260+

MCUs
SUPPORTED



CODEGRIP
ON BOARD



mikroBUS™
SOCKETS



AMAZING
CONNECTIVITY



DUAL POWER
SUPPLY

Thank you for being our valued customer!

We are excited to have the opportunity to present you the NEW generation of our flagship development board – the **EasyPIC v7a**!

From 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 the best for an engineer. I've personally assembled hundreds of early EasyPIC boards myself with my home soldering iron.

Today, at the beginning of the 2021, we present you the **A** generation of the board, which brings us some awesome new features. We hope that you will like it as much as the previous one.

Use it wisely and have fun!

A stylized, handwritten signature in black ink, consisting of several loops and a long horizontal stroke extending to the right.

Nebojsa Matic,
CEO

Table of contents

Introduction

Introduction	6
It's good to know	8

Power supply unit

Dual power supply	9
How to power the board?	10

Supported MCUs

Supported microcontrollers	11
How to properly place your MCU into the DIP socket?	12
Using crystal oscillators	13

Programmer/debugger

On-board programmer	14
What is the CODEGRIP?	14
Where to start?	14
MCLR PIN selection	14
Programming with icd2/icd3	15

Communication

UART via RS-232	16
UART via USB	17
USB - connection	18

Connectivity

mikroBUS™ sockets	19
Click boards™	20
INPUT/OUTPUT group	22
Headers	23
Buttons	23
LEDs	23

Displays

LCD 2x16 characters	24
GLCD 128x64	25
Touch panel controller	26
4 digit 7-seg display	27

Other modules

DS1820 Digital temp. sensor	28
LM32 Analog temp. sensor	29
ADC INPUTS	30
I2C EEPROM	31
Piezo buzzer	32
Additional GNDs	33

All images shown in the manual are for illustration purpose only.

INTRODUCTION

EasyPIC is an old friend. 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 equipped it with our revolutionary **new programmer & debugger - CODEGRIP**. We now present you with the **new version 7a** that brings so much more, and we hope that you will be thrilled with your new board, just as we are.



4 CONNECTORS FOR EACH PORT

Amazing connectivity

EasyPIC v7a is all about onnectivity. 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

CODEGRIP on board

Powerful on-board CODEGRIP USB-C programmer and In-Circuit debugger can program and debug over 260 microcontrollers. You will need it, whether you are a professional or a beginner.



3.3V AND 5V POWER SUPPLY

Dual Power Supply

EasyPIC v7a 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

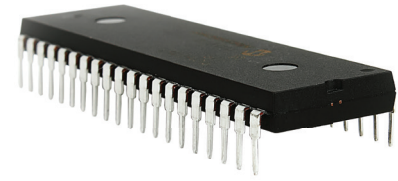
mikroBUS™ support

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 default microcontroller!



PIC18F45K22 is the default chip of EasyPIC v7a. 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].

SYSTEM SPECIFICATIONS



POWER SUPPLY
7–23V AC or 9–32V DC
or via USB cable [5V DC]



POWER CONSUMPTION
~30mA
[no docked Hexiwear]



BOARD DIMENSION
266 x 220 mm
[10.47 x 8.66 inch]



BOARD WEIGHT
490g
[1.080 lbs]

PACKAGE CONTAINS

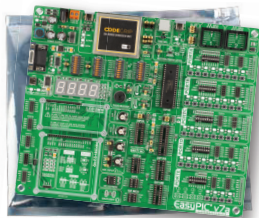
1

Damage
resistant
protective
box



2

EasyPIC v7a
development
board



3

USB
cable



4

User Manual &
Schematics



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 1]. Power LED (Green ON) will indicate the presence of power supply.

Figure 1: Dual power supply unit of EasyPIC v7a

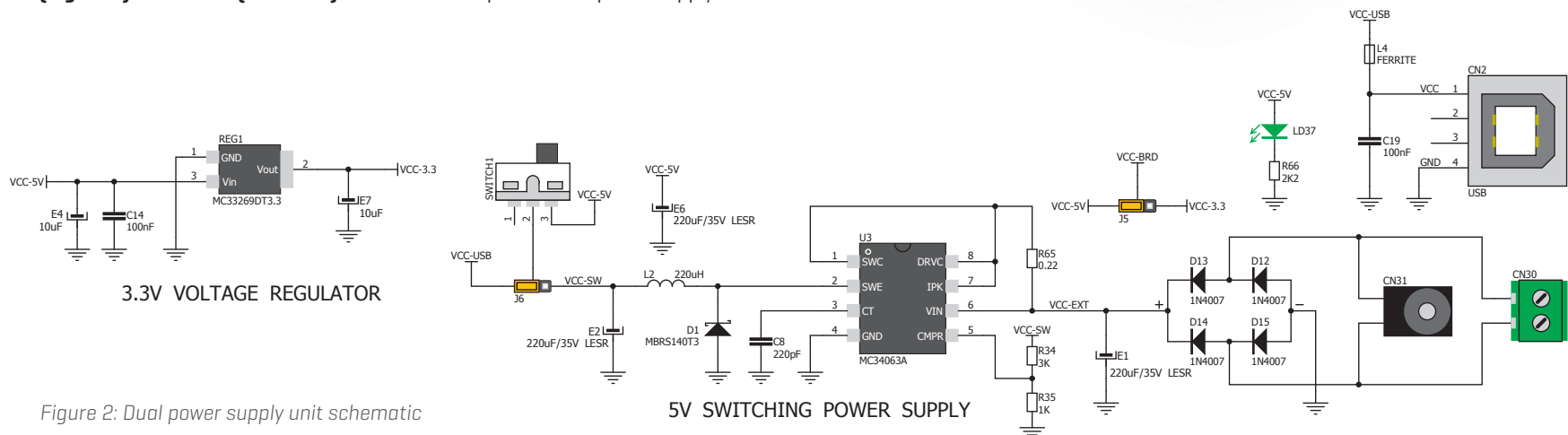


Figure 2: Dual power supply unit schematic

HOW TO POWER THE BOARD?

DUAL PWR SUPPLY



The EasyPIC v7a 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

1. With USB cable

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 image Figure 3, and turn the power switch ON.

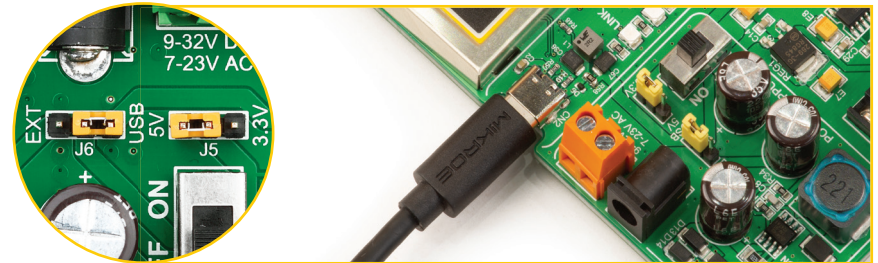


Figure 3: Powered by USB

2. By using wall-adapter

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 image Figure 4, and turn the power switch ON.

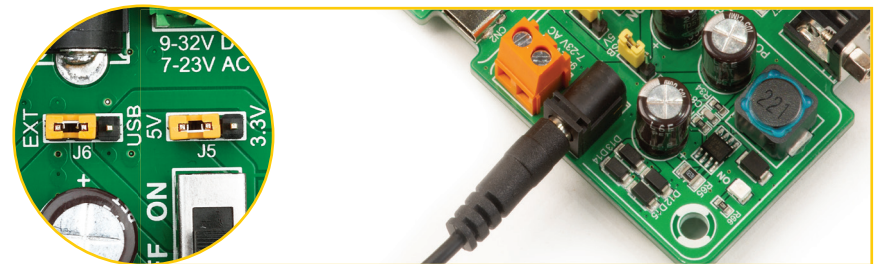


Figure 4: Powered by wall-adapter

3. With laboratory PSU

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 image Figure 5, and turn the power switch ON.

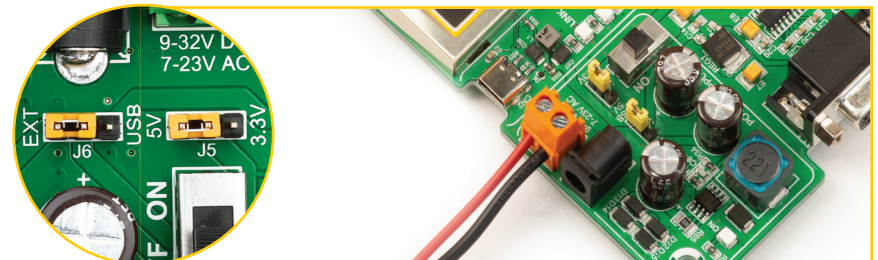


Figure 5: Powered by laboratory PSU

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 v7a development system comes with the PIC18F45K22 microcontroller in a DIP40 package.

NOTE When using PIC18F2331 or PIC18F2431 microcontrollers it is necessary to place J20 jumper, in order to route VCC power line to RA5 pin – Figure 7.

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 v7a board provides jumpers which are used for this purpose. Here is list of devices that require jumpers placed in VCAP position:

- 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

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

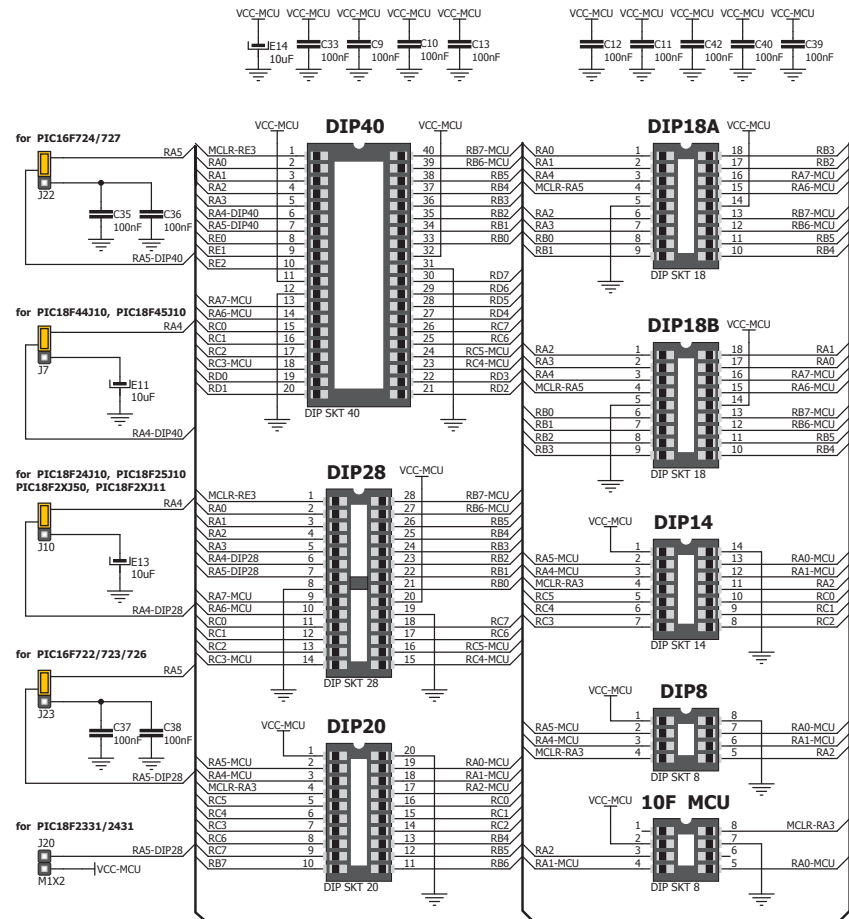


Figure 6: Schematic of on-board DIP sockets and VCAP jumpers

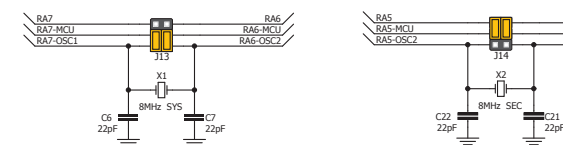


Figure 7: crystal oscillators

How to properly place your microcontroller into the DIP socket?

Before you plug the microcontroller into the socket, make sure that the **power supply is turned off**. Images below show how to correctly plug the 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 8**. Then put the microcontroller slowly down until all the pins match the socket as shown in **Figure 9**. Check again if everything is placed correctly and press the

microcontroller until it is completely plugged into the socket as shown in **Figure 10**. If done correctly all pins should be fully inserted. Only now can you turn on the power supply.

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

Figure 8: Place both ends of microcontroller on the socket so the pins are aligned correctly

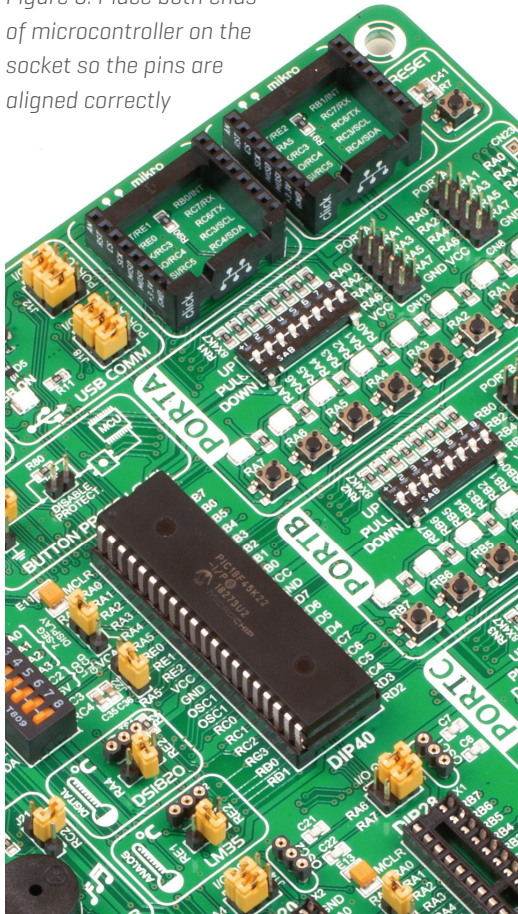


Figure 9: with both fingers, evenly distribute the force and press the chip into the socket.

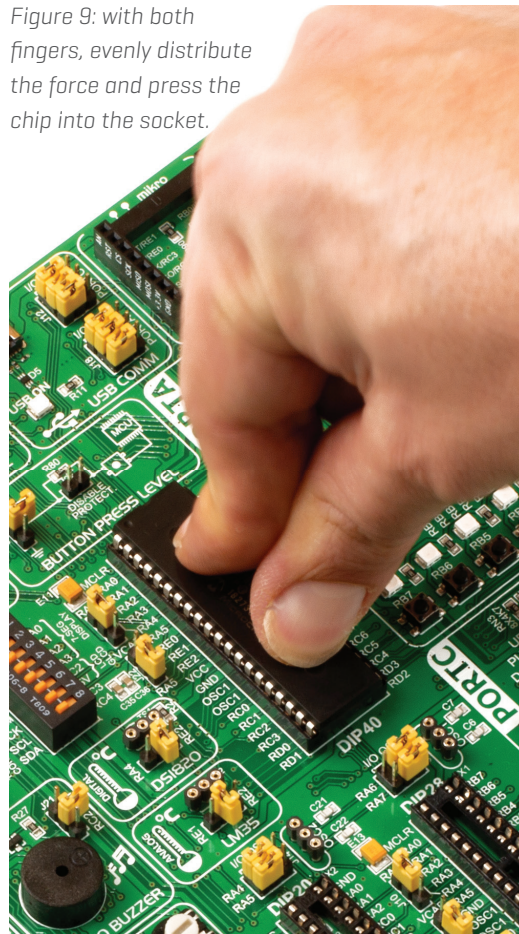
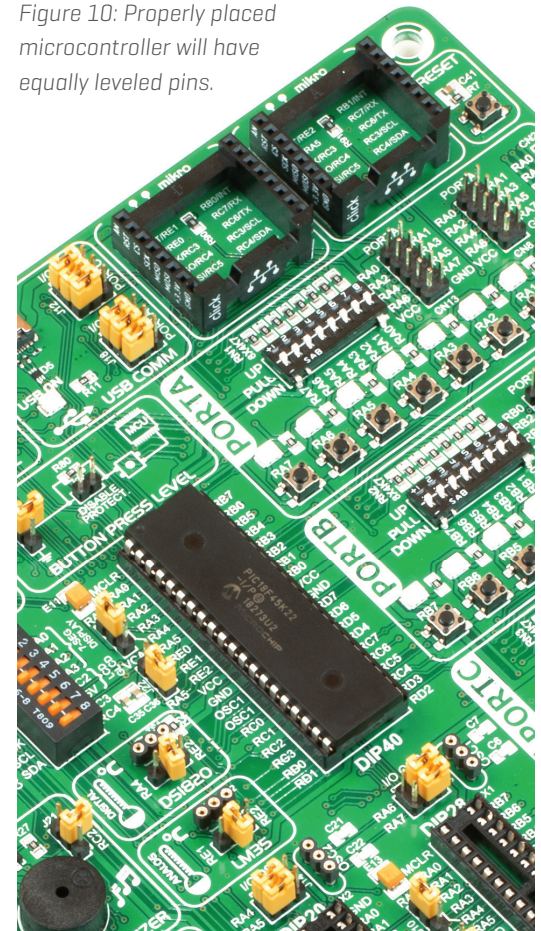


Figure 10: Properly placed microcontroller will have equally leveled pins.



USING CRYSTAL OSCILLATORS



Figure 11: RA6 and RA7 as I/O pins
[when using internal oscillator]



Figure 12: RA6 and RA7
connected to X1 quartz-crystal

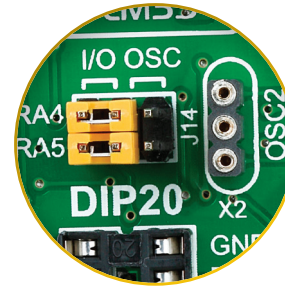


Figure 13: RA4 and RA5 as I/O pins
[when using internal oscillator]



Figure 14: RA4 and RA5 connected
to X2 quartz-crystal

PIC microcontrollers normally use a quartz crystal for the purpose of providing clock frequency. The **EasyPIC v7a** provides two sockets for quartz-crystal. Microcontrollers in **DIP18A**, **DIP18B**, **DIP28** and **DIP40** packages use socket **X1 [OSC1]** for 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.

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

NOTE

ON BOARD PROGRAMMER

WHAT IS CODEGRIP?

CODEGRIP is a unified solution, designed to perform programming and debugging tasks on a range of different microcontroller devices [MCUs] based on the Microchip PIC architecture. The USB-C connector offers improved performance and reliability, compared to traditionally used USB Type A/B connectors.

EasyPIC v7a development board is supported by a powerful CODEGRIP Suite, offering complete control over the development board. It is used to intelligently manage programming and debugging tasks, and to configure various other options and settings, providing visual feedback through its clean and comprehensive Graphical User Interface [GUI].

To better understand how to operate and configure EasyPIC v7a development board and its integrated CODEGRIP module, check out the CODEGRIP Suite quick start guide on the www.mikroe.com/easypic-v7a

WHERE TO START?

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

1. Install the necessary software

- Install USB drivers
- Install CODEGRIP Suite software

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

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

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

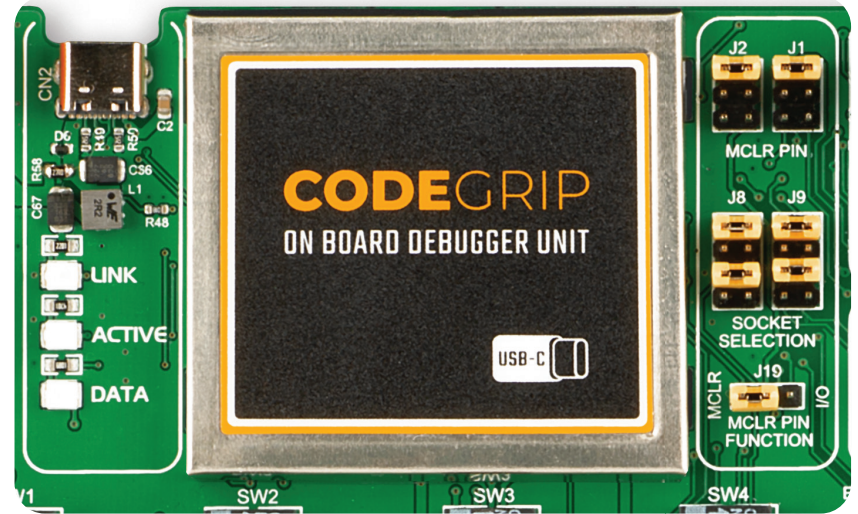


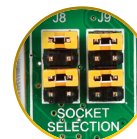
Figure 15: CODEGRIP on board programmer

Why so many LEDs?

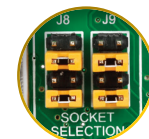
Three LEDs indicate specific programmer operation. **Link** LED lights up when USB link is established with your PC, **Active** LED lights up when programmer is active. **Data** is on when data is being transferred between the programmer and PC software. [Compiler or CODEGRIP].

PROGRAMMING 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.



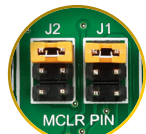
DIP40, DIP28,
DIP18A, DIP18B



DIP20, DIP14,
DIP8

MCLR PIN 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

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

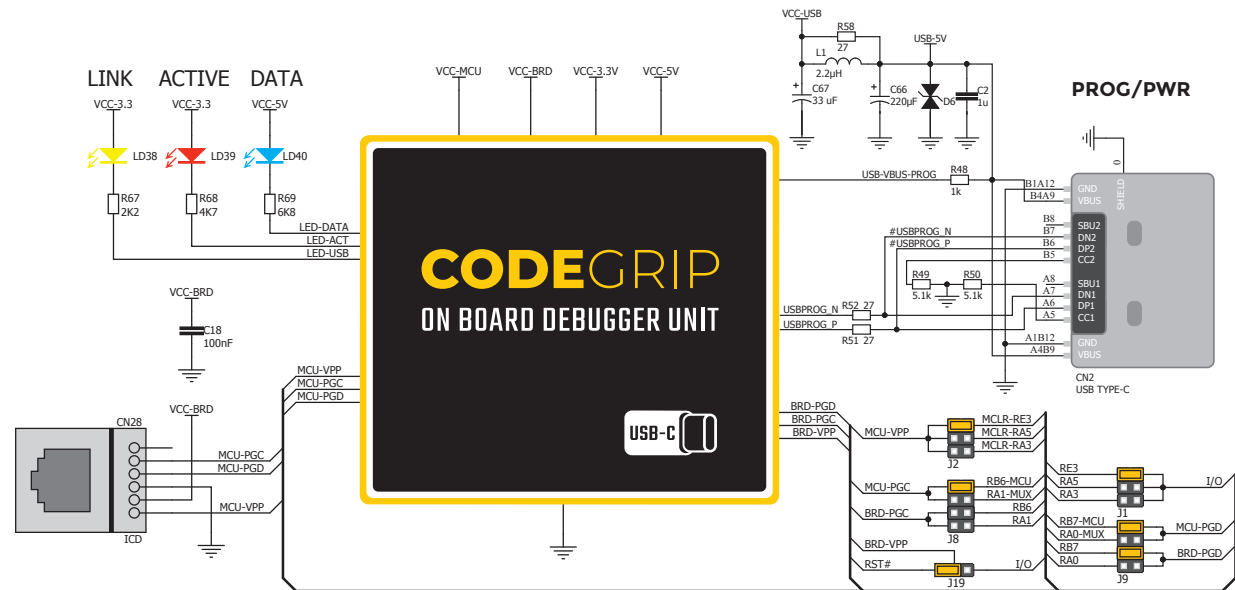


Figure 16: CODEGRIP on board programmer schematic

PROGRAMMING WITH ICD2/ICD3/ICD4

EasyPIC v7a is equipped with RJ-12 connector compatible with Microchip® ICD2, ICD3, and ICD4 external programmers. You can either use the on-board CODEGRIP 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 image.

The DIP switch located next to the RJ-12 connector allows control of the interface between onboard CODEGRIP module and target MCU:

ONBOARD (down): Interface is enabled [1]. If an external debugger probe-device is connected, there is a possible collision in communication.

EXTERNAL (up): Interface is disabled [2]. External debugger probe-device can reliably communicate with target MCU.

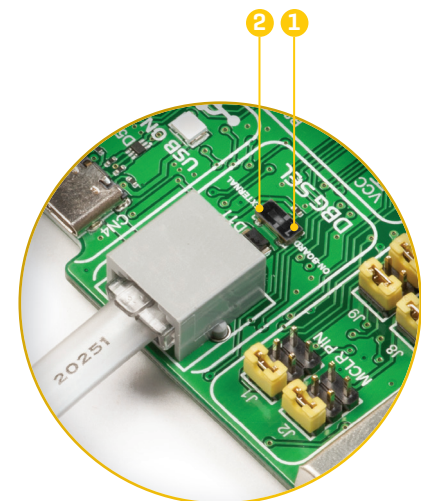
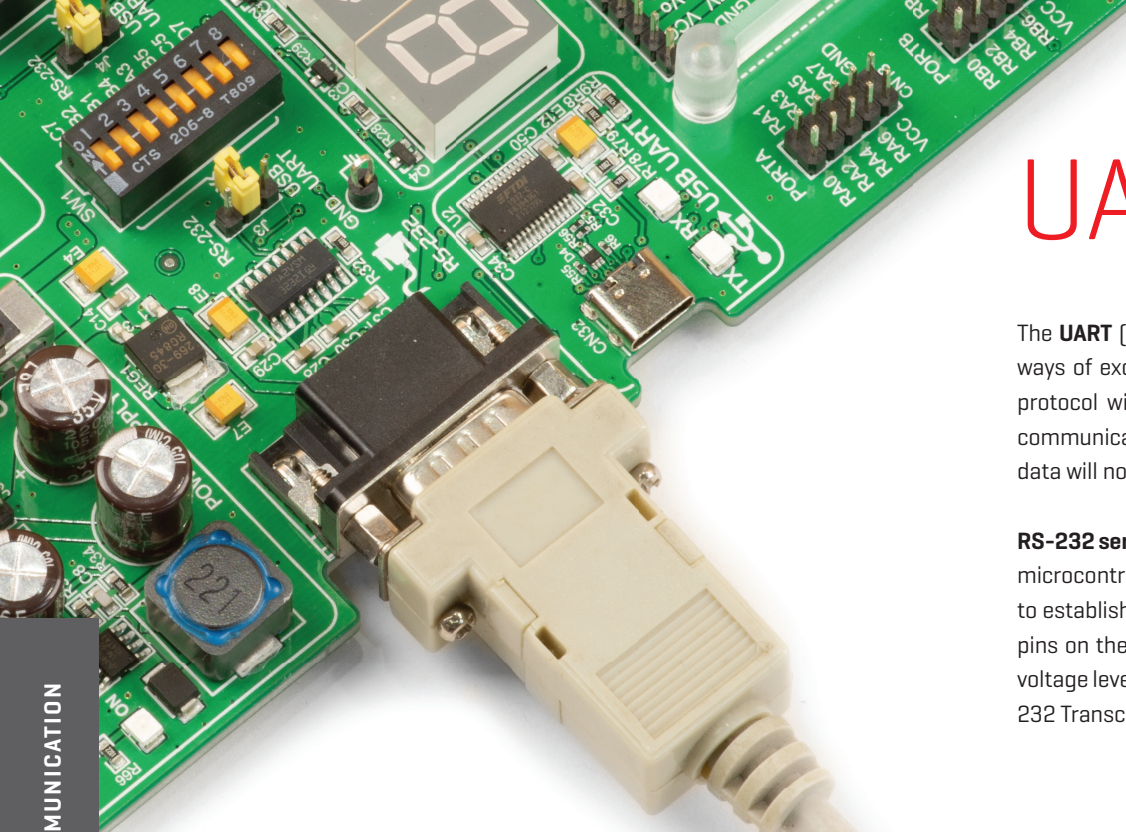


Figure 17: RJ-12 connector view



UART via RS-232

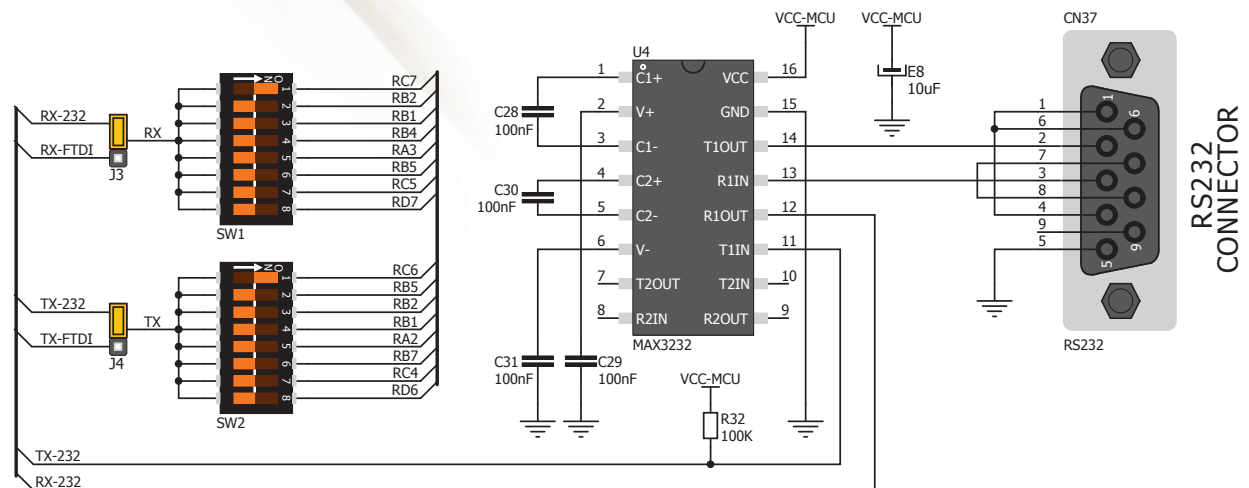
The **UART** [universal asynchronous receiver/transmitter] is one of the most common ways of exchanging data between the MCU and peripheral components. It is a serial protocol with separate transmit and receive lines, and can be used for full-duplex communication. Both sides must be initialized with 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 18**.

Figure 18: RS-232 connection schematic

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.

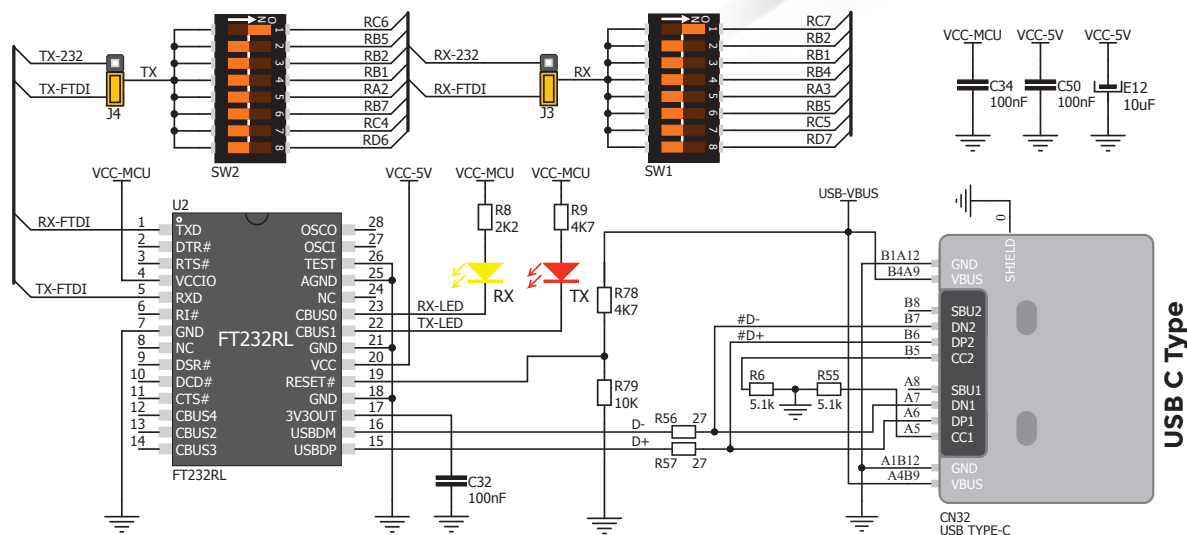


UART via USB

Modern PC computers, laptops and notebooks are no longer equipped with RS-232 connectors and UART controllers. They are nowadays 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 v7a, you must first install FTDI drivers on your computer. Drivers can be found on the following link: www.ftdichip.com/Drivers/VCP.htm

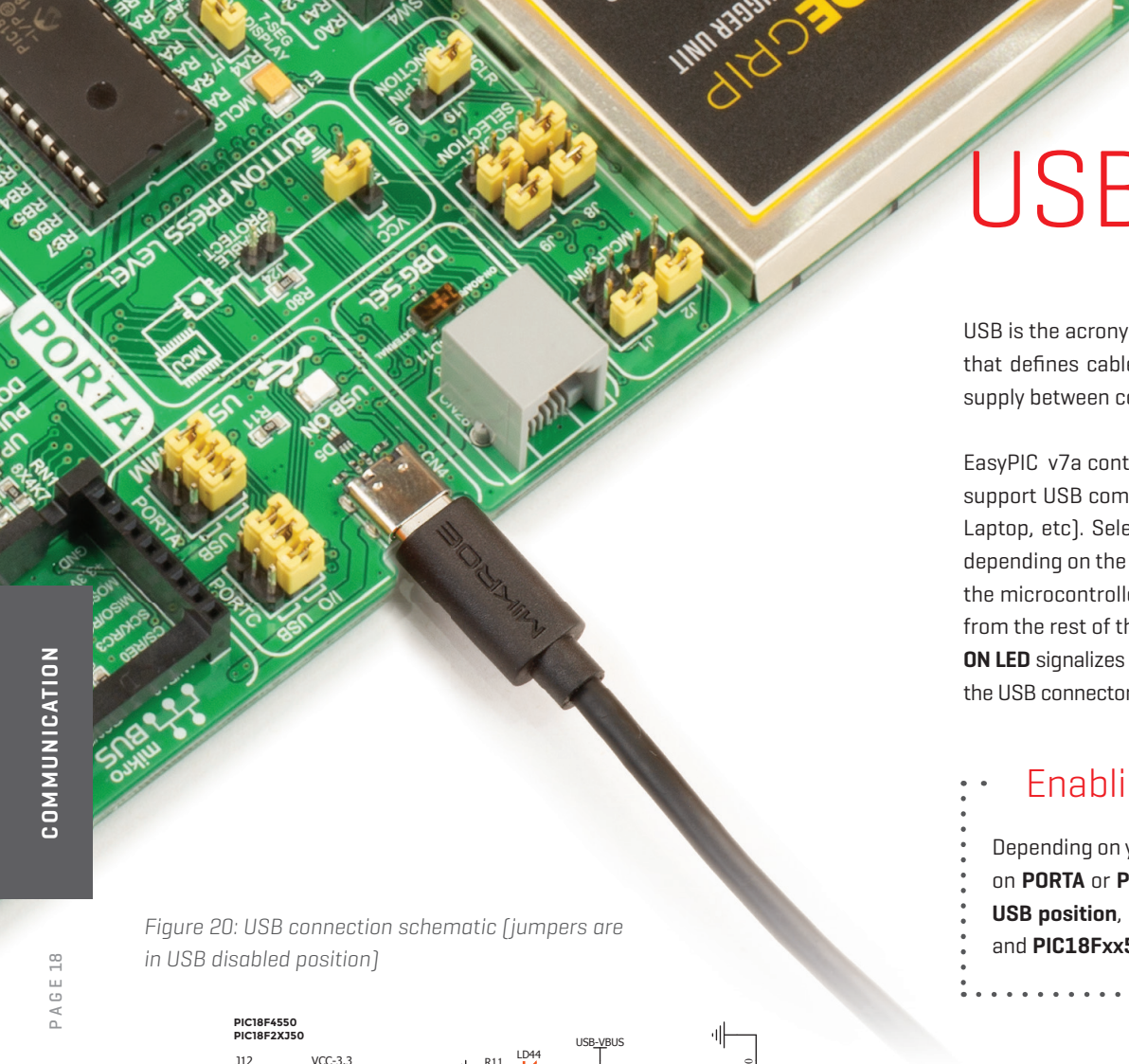
USB-UART communication is being done through a FT232RL controller, USB connector **[CN32]**, and microcontroller UART module. To establish this connection, you must put **J3** and **J4** jumpers in the **USB-UART position**, and connect **RX** and **TX** lines to the appropriate pins of the microcontroller.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

Figure 19: USB-UART connection schematic



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.



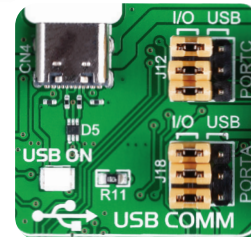
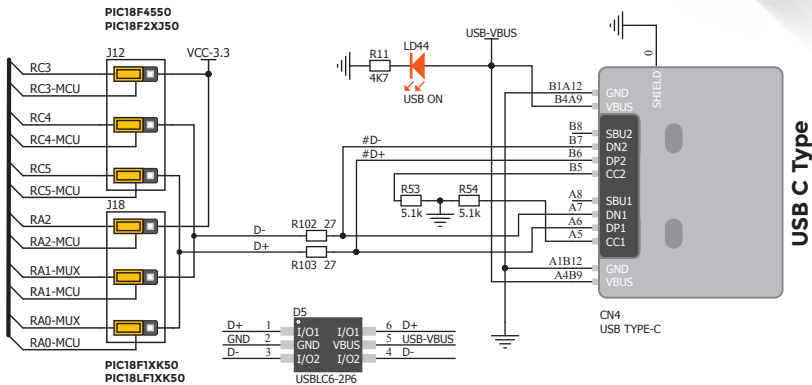
USB CONNECTION

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.

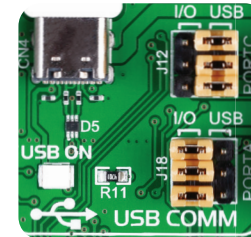
EasyPIC v7a contains USB connector **[CN4]** which enables microcontrollers that support USB communication to establish a connection with the target host [eg. PC, Laptop, etc]. Selection of communication lines is done using jumpers **J12** or **J18**, depending on the target microcontroller. When communication 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 conn.

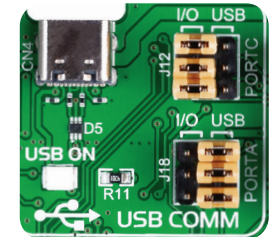
Depending on your target microcontroller, USB communication can be enabled on **PORTA** or **PORTC**. For **PIC18(L)F1XK50** you should put **J18** jumpers in the **USB position**, **Figure 22**. For **PIC18Fxx[J]50**, **PIC18Fxx[J]53**, **PIC18Fxx[J]55** and **PIC18Fxx58** place **J12** jumpers in the **USB position**, **Figure 21**.



USB function disabled



USB enabled on PORTC



USB enabled on PORTA

mikroBUS™ SOCKETS

The superior connectivity features of the EasyPIC v7a development board are rounded up with two standardized mikroBUS™ host connectors. It is a considerable upgrade for the board, as it allows interfacing with the vast amount of **Click boards™**.

mikroBUS™ is the add-on board standard that offers maximum expandability with the smallest number of pins. More information at www.mikroe.com/mikrobus

All the **mikroBUS™ sockets** are now redesigned and improved. They became much sturdier, allowing a better grip of the Click board™. In addition to added horizontal supports between two 8-pin connectors, the right lower edge is chamfered at the angle of 45°, the same as the Click board™, making it seamlessly fit into the mikroBUS™ socket. The new mikroBUS™ socket is now a fully-fledged, standardized monolithic component with all its pins clearly labeled, offering very good grip for the Click board™, preventing it to flip over or to be placed incorrectly.

The mikroBUS™ socket supports the following communication interfaces: **SPI, UART, and I²C**. There are also single pins reserved for **PWM, Interrupt, Analog input, Reset,** and **SPI Chip Select**. The mikroBUS™ socket also contains pins with the two power rails (**3.3V and 5V**), along with the **GND pins**.

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:

www.mikroe.com/mikrobus

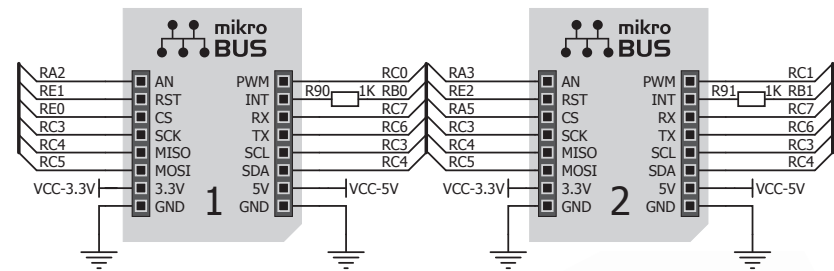
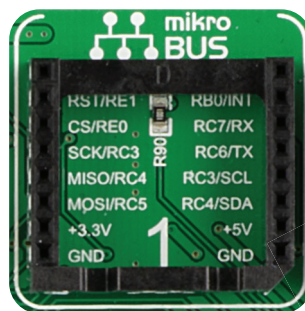


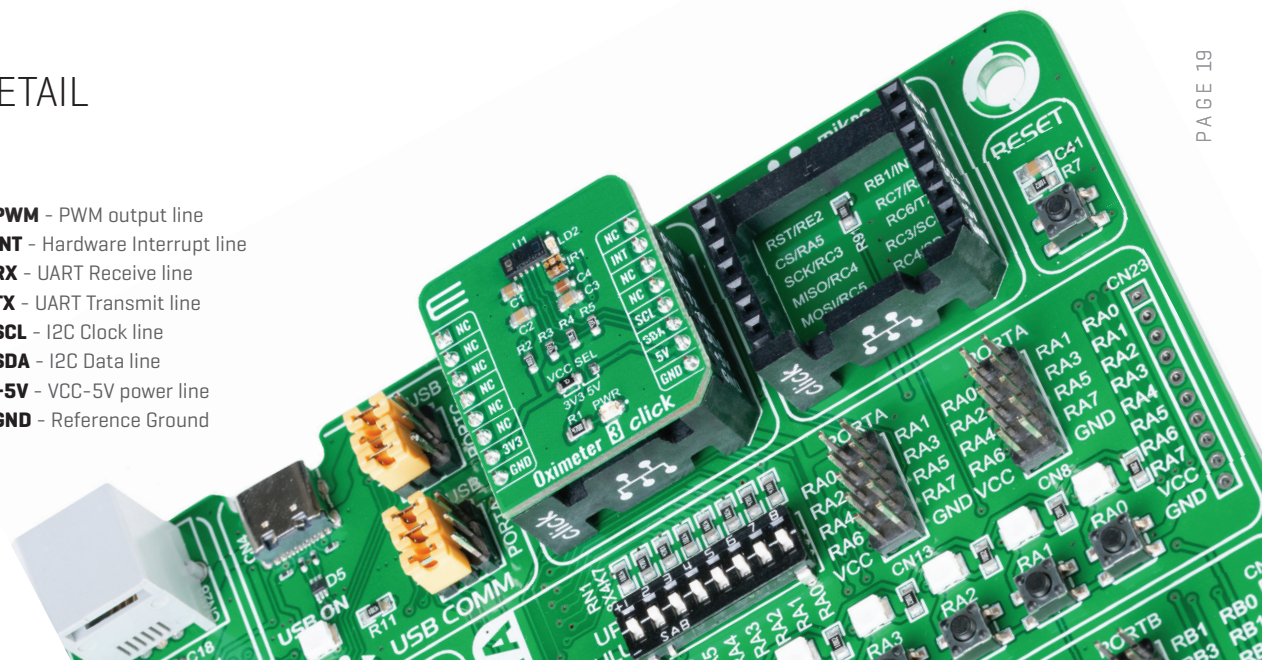
Figure 21: mikroBUS™ sockets schematic

mikroBUS™ SOCKET IN DETAIL

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



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



Click Boards™

THE LARGEST AND FASTEST-GROWING BASE OF
ADD-ON BOARDS IN THE WORLD!

Click boards™ are standardized add-on boards that carry a variety of different electronic devices. They are designed to perfectly fit the mikroBUS™ socket. Engineered to deliver the best performances for the used components, they save developers of testing and troubleshooting often associated with the prototyping phase. They enhance rapid development and accelerate time to market. These ready-to-use boards require no additional hardware configuration. More information at

www.mikroe.com/click



LIBSTOCK® A PLACE FOR CODE®

BROWSE CODE | SHARE CODE | REQUEST CODE | ABOUT LIBSTOCK

Use your [forum](#) account to login.

LOGIN

Username:

Password:

[▶ LOGIN](#)

Not registered? [Sign Up](#) for free!

CODE CATEGORIES

All codes (789)

- ▶ Audio & Speech (20)
- ▶ Communication (194)
- ▶ Graphics & LCD (94)
- ▶ GSM / GPRS (12)
- ▶ GPS (11)

CATEGORIES

☒ All codes ☐ Libraries ☐ Projects ☐ Visual Projects ☐ Examples

COMPILERS

☐ mikroC ☐ mikroBasic ☐ mikroPascal ☐ Other Compilers

SORT BY

☐ best rank ☐ oldest ☐ by author ☐ alphabetically ☒ most down

CATEGORY: ALL CODES

GSM click - Example

Rating: ★★★★★

Author: [MikroElektronika Team](#)

Version: 1.0.0.0

Last Updated: 2014-05-09

Category: [GSM / GPRS](#)

Downloaded: 7921 times

Followed by: 2 users

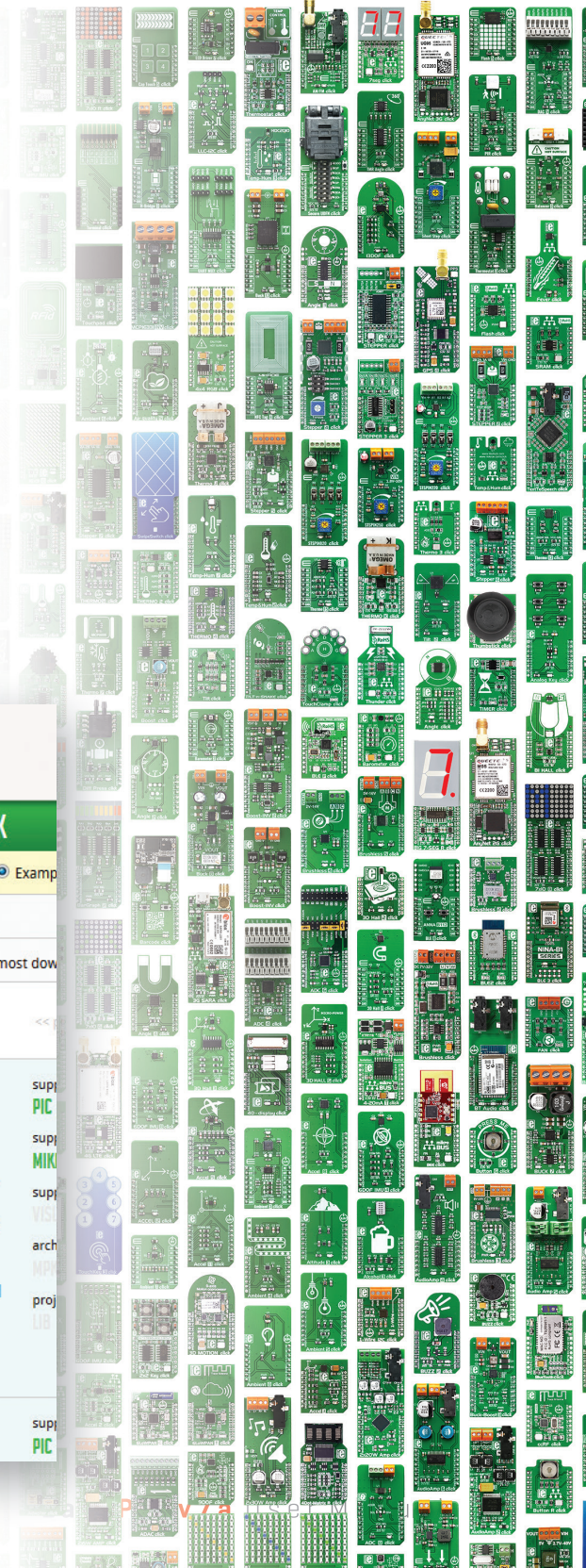
[Download \[648.97KB\]](#)

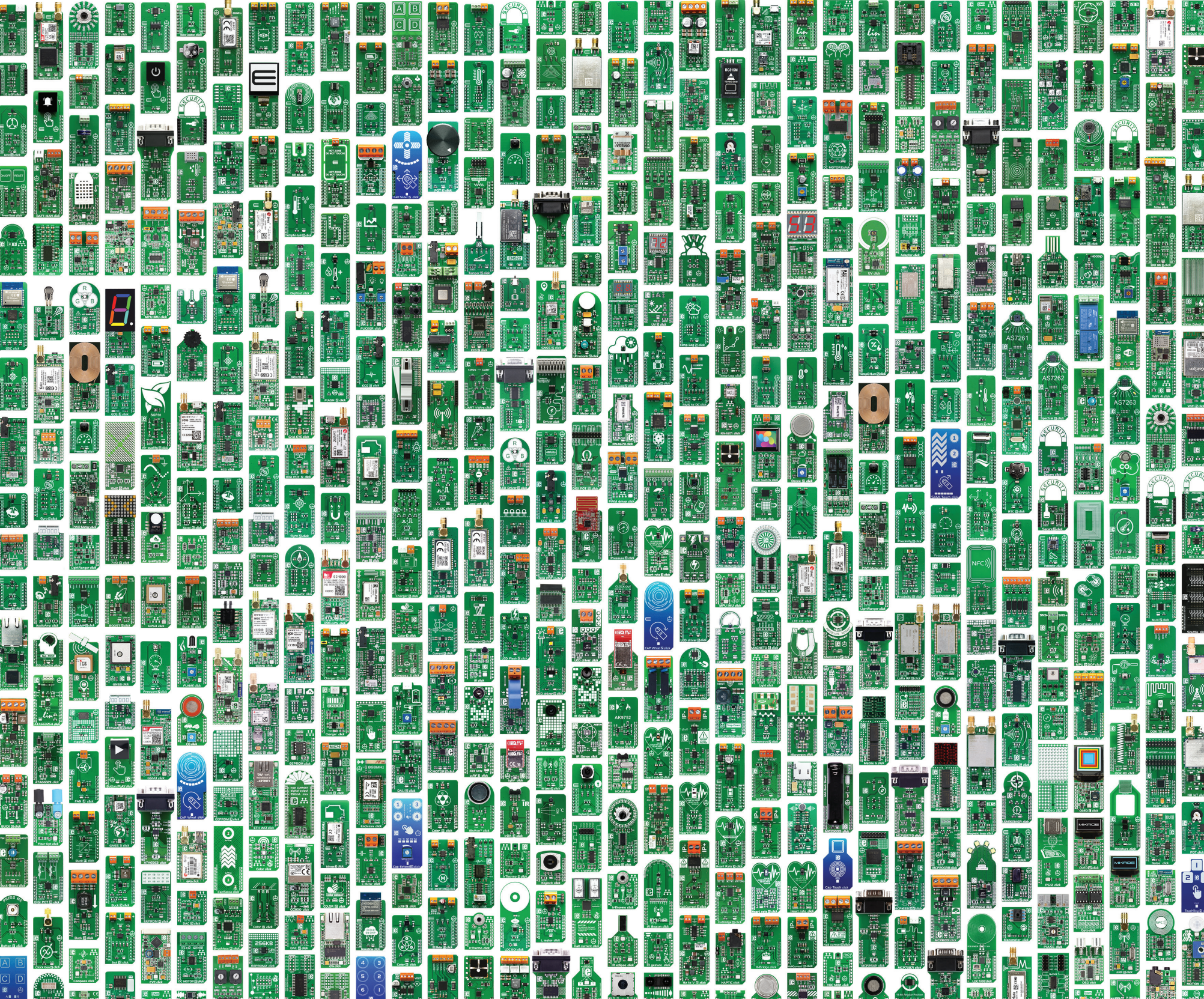
Description

This is a sample program which demonstrates the use of **GSM click**. Program uses GSM module GL865 for sending/receiving SMS and establishing calls with dialed number. Dialing is done using GLCD+Touch Panel GUI.

[\[learn more\]](#)

microSD click - Example





INPUT/OUTPUT GROUP

One of the most distinctive features of EasyPIC v7a is Input/Output PORT groups. They add so much to the connectivity potential of the board. These groups contain buttons, LEDs and headers. They allow interaction with the EasyPIC v7a, either by pressing buttons, displaying states of its pins on the LEDs, or using headers to interface it to an external device.

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 v7a 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 onboard modules are enabled via jumpers [for example USB jumpers, **J12** and **J18**].

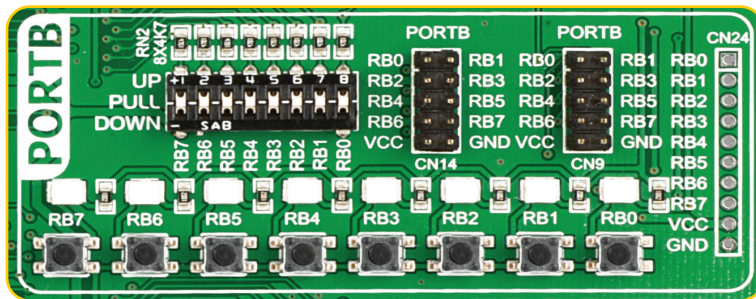
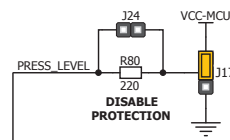


Figure 22: Tri-state DIP switch on PORTC

Figure 23: Schematic of the single I/O group connected to microcontroller PORTC

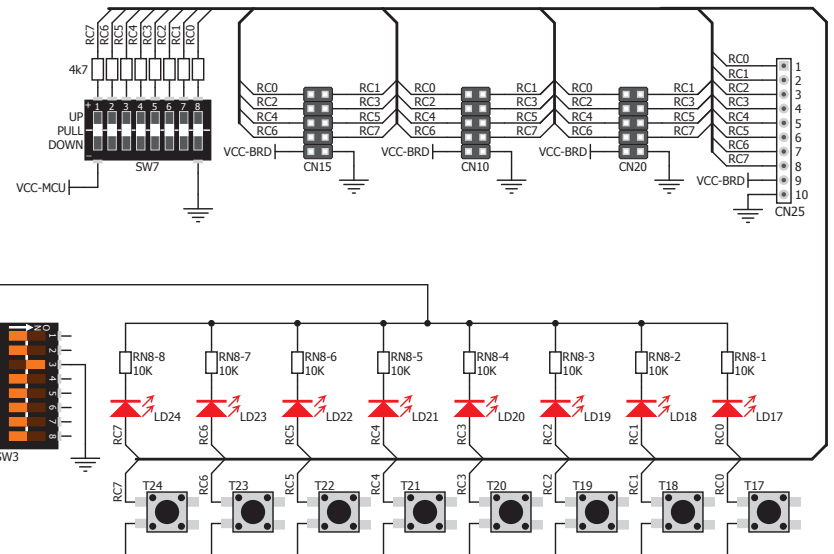
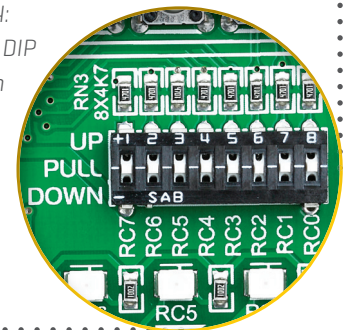


TRI-STATE DIP SWITCHES

Tri-state DIP switches, like **SW7** on **Figure 24**, are used to enable 4K7 pull-up or pull-down resistor on any desired port pin. Each of these switches has three states:

- MIDDLE POSITION** disables both pull-up and pull-down feature from the PORT pin
- UP POSITION** connects the resistor in pull-up state to the selected pin
- DOWN POSITION** connects the resistor in pull-down state to the selected PORT pin.

Figure 24:
Tri-state DIP
switch on
PORTC



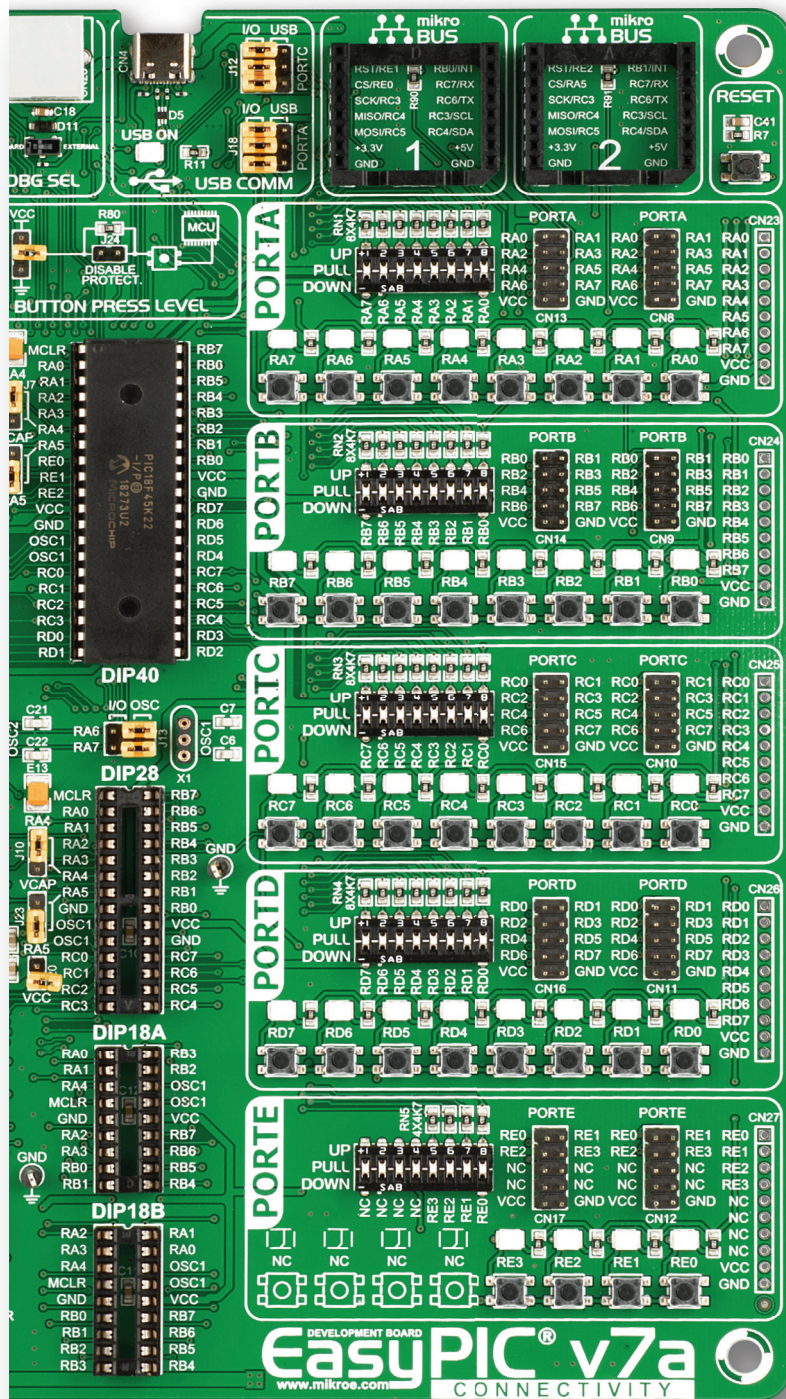


Figure 25: INPUT/OUTPUT ports view

HEADERS

With enhanced connectivity as one of the key features of EasyPIC v7a, we have provided **four connection headers for each PORT**. I/O PORT group contains two male IDC10 headers (like **CN10** and **CN15** on **Figure 28**). 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.

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.

NOTE

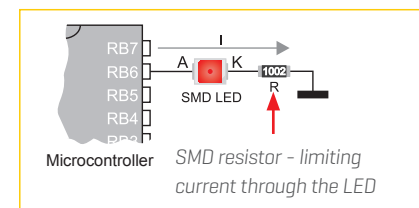
I/O PORT group also contains 1x10 connection pad (like **CN25** on **Figure 25**) which can be used for connecting MikroElektronika PROTO boards, or custom user boards.

BUTTONS

The logic state of all microcontroller digital inputs may be changed using **push buttons**. Jumper **J17** is used to determine the logic state to be applied to the desired microcontroller pin when pressing the appropriate push button. The purpose of the **protective resistor** is to limit maximum current thus preventing a short circuit from occurring. Advanced users may, if needed, disable such resistor using jumper **J24**. By pressing any push button when jumper **J17** is in the VCC-MCU position, a logic one [3.3V or 5V] will be applied to the appropriate microcontroller pin.

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 v7a 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 indicativon 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 23**).

LCD 2x16 characters

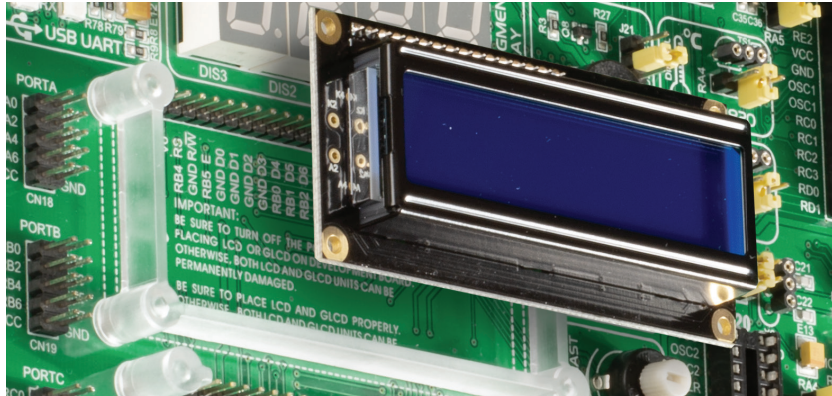


Figure 26: 2x16 LCD connector and display view

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 v7a 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.

Standard & 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.

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

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

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

NOTE

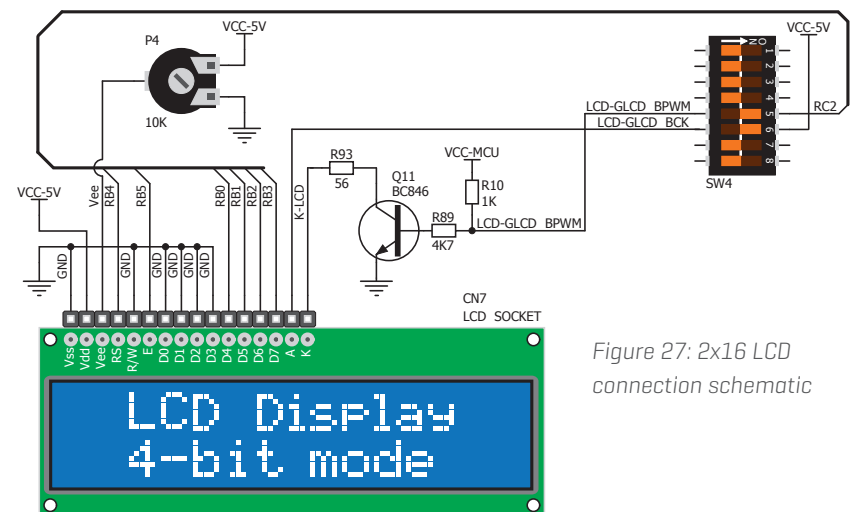


Figure 27: 2x16 LCD connection schematic

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 v7a 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.

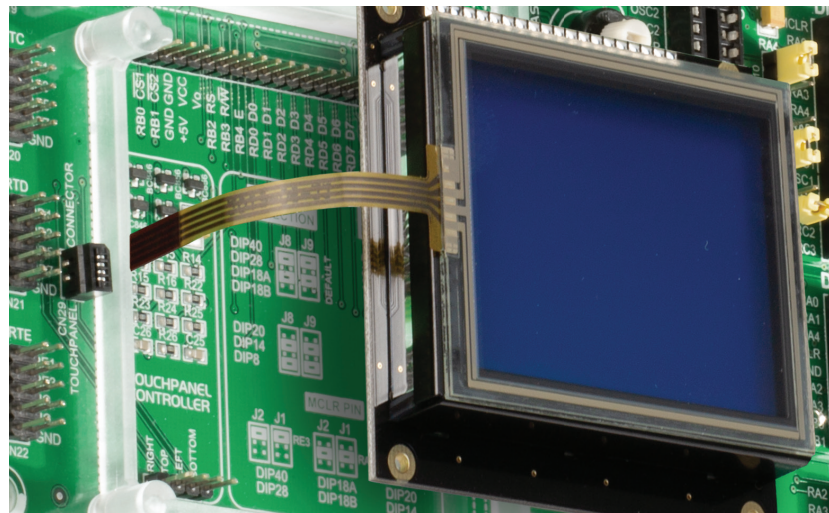
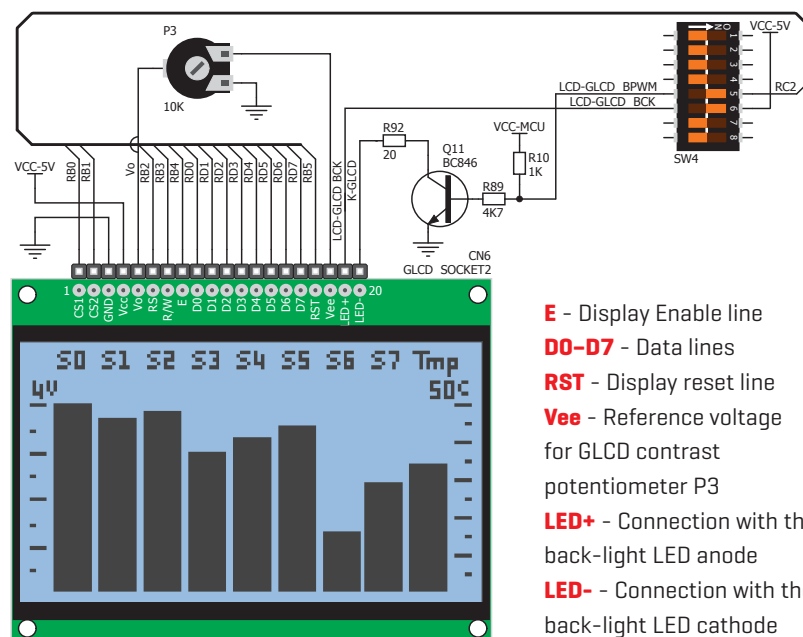


Figure 29: GLCD 128x64 connection schematic

CONNECTOR PINOUT EXPLAINED

CS1 & **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



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 **SW4.5** switch, and PWM-driven back light with **SW4.6** switch.

Figure 28: GLCD 128x64 connector and display view

Standard & PWM-driven back-light

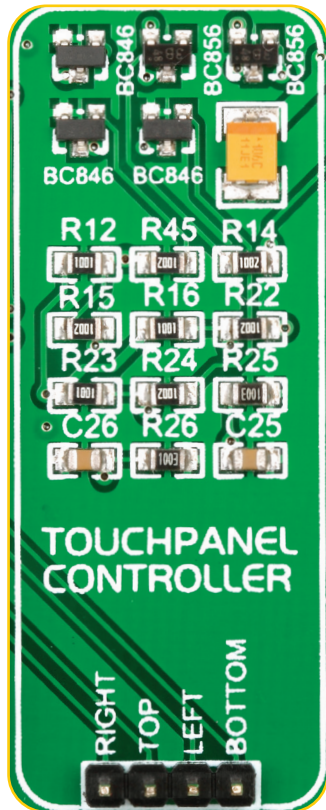
As for LCD, we have allowed GLCD back-light to be enabled in two different ways:

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.

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

NOTE

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 v7a 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.

Figure 30: Touch panel controller module view

CORRECTLY PLACING THE TOUCH PANEL CABLE INTO THE CONNECTOR

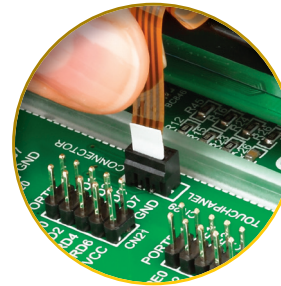


Figure 31: Put and push Touch panel flat cable in the connector

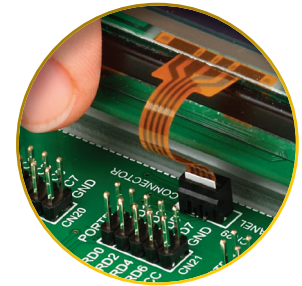


Figure 32: Now place GLCD with Touch panel into GLCD socket

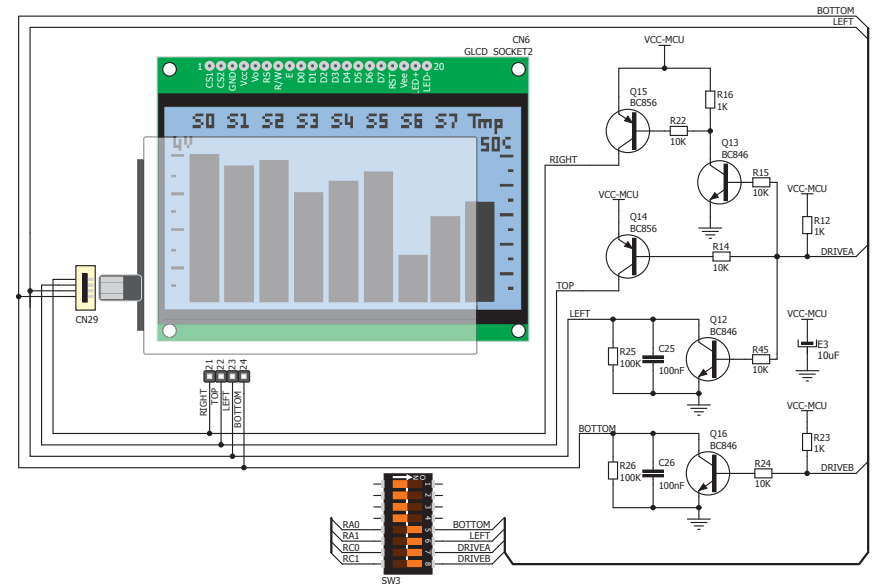


Figure 33: Touch Panel controller and connection schematic

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.

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 v7a 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.

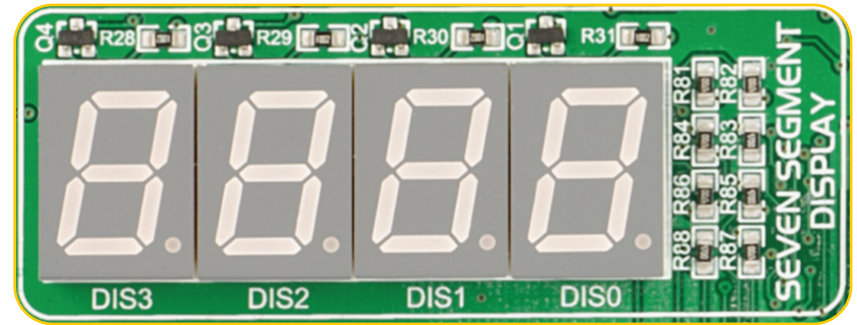


Figure 34: 4 digit 7-segment display view

- **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.

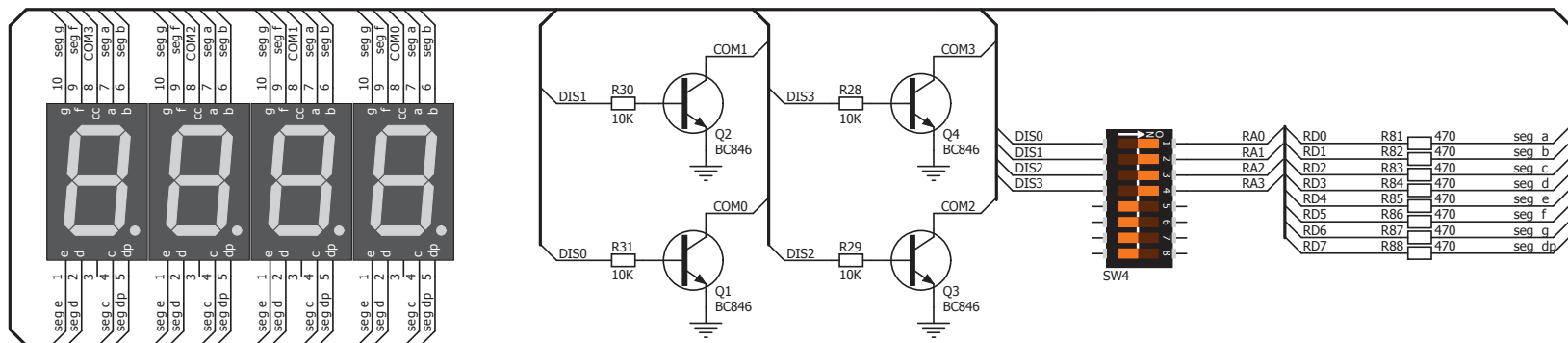


Figure 35: 4 digit 7-segment display schematic

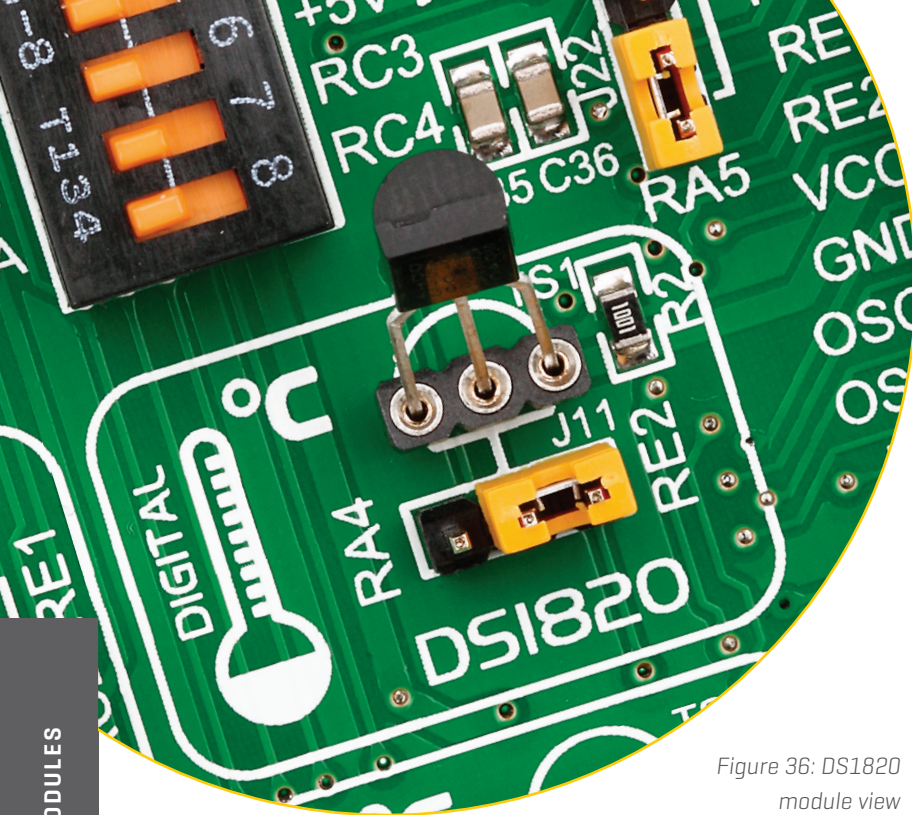


Figure 36: DS1820 module view

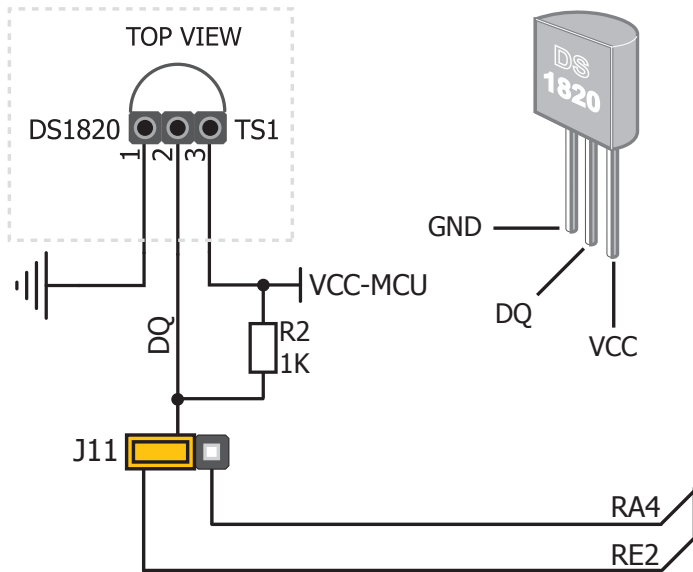


Figure 37: DS1820 connected to RE2 pin schematic

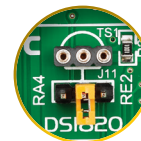
DS1820 - DIGITAL TEMP. SENSOR

DS1820 is a digital temperature sensor that uses **1-wire® interface** for its operation. It is capable of measuring temperatures within the range of -55 to 128°C , and provides $\pm 0.5^{\circ}\text{C}$ accuracy for temperatures within the range of -10 to 85°C . It requires 3V to 5.5V power supply for stable operation. It takes max. 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.

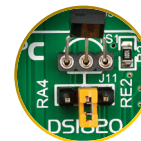
EasyPIC v7a provides a separate socket [TS1] for the DS1820. Communication line with the microcontroller is connected via jumper J11.

Enabling DS1820 sensor

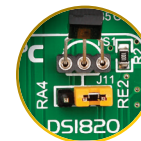
EasyPIC v7a enables you to establish 1-wire® communication between **DS1820** and the microcontroller via **RA4** or **RE2** microcontroller pins. The selection of either of those two lines is done using **J11** jumper. When placing the sensor in the socket make sure that half-circle on the board's silkscreen 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.



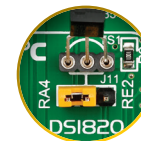
Not connected



Placed in socket



Connected to RE2 pin



Connected to RA4 pin

LM35 - ANALOG TEMP. 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 $+10.0 \text{ mV}/^{\circ}\text{C}$ scale factor and less than $60 \mu\text{A}$ current drain. As it draws only $60 \mu\text{A}$ from its supply, it has very low self-heating, less than 0.1°C in still air. EasyPIC v7a enables you to get analog readings from the LM35 sensor in restricted temperature range from $+2^{\circ}\text{C}$ to $+150^{\circ}\text{C}$. Board provides a separate socket [TS2] for the LM35 sensor in T0-92 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

EasyPIC v7a 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.



Not connected



Placed in socket



Connected to RE2 pin



Connected to RE1 pin

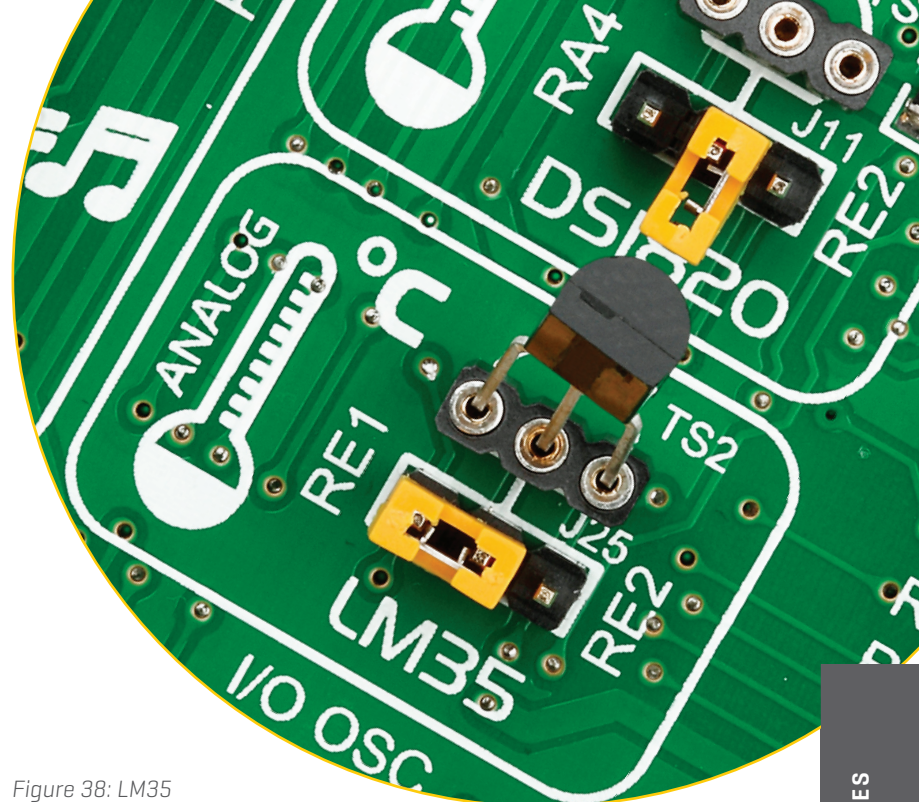


Figure 38: LM35 module view

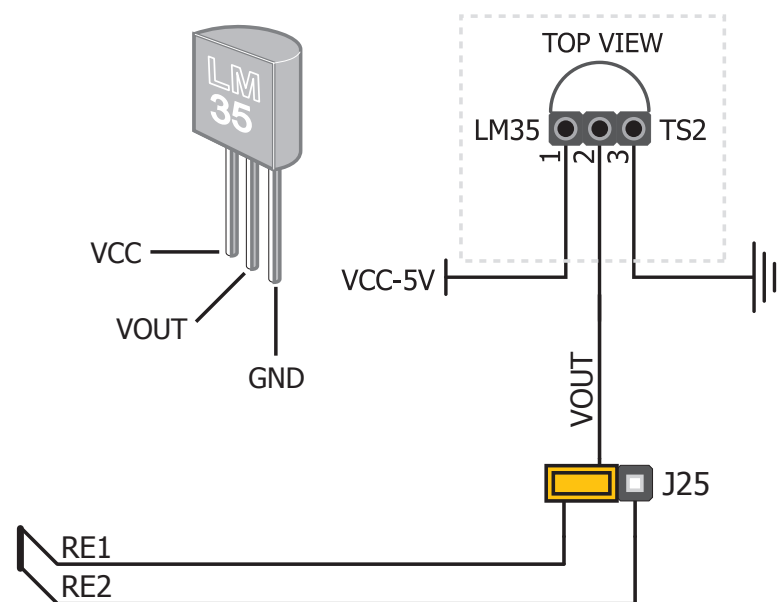


Figure 39: LM35 connected to RE1 pin schematic

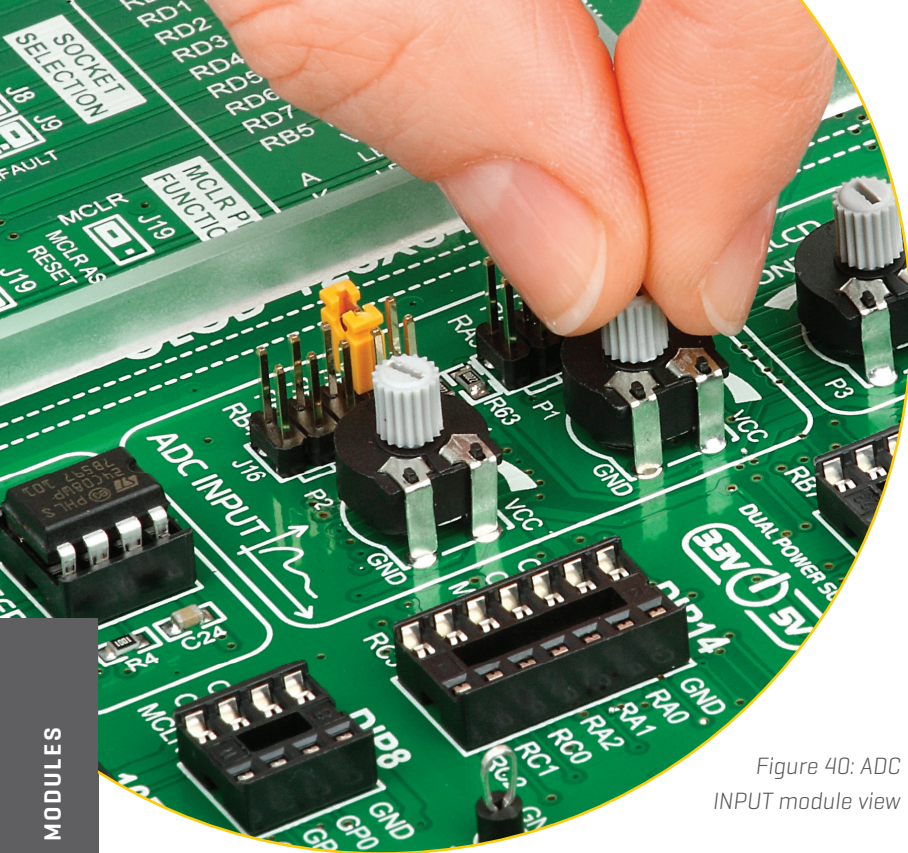


Figure 40: ADC
INPUT module view

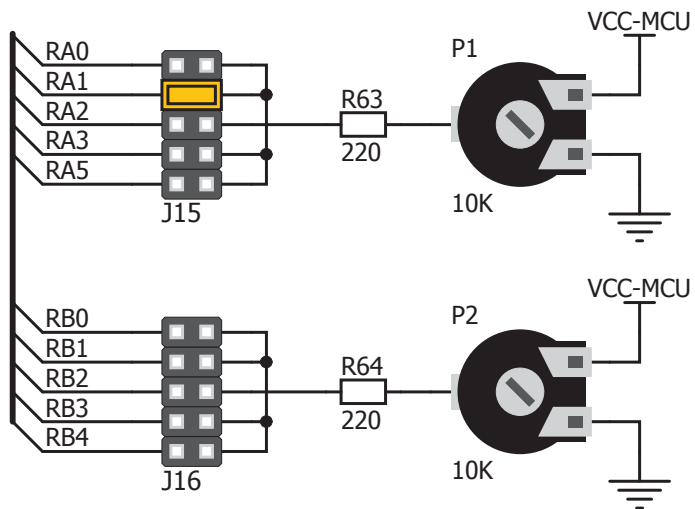


Figure 41: ADC input schematic

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 v7a 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.

Enabling ADC inputs

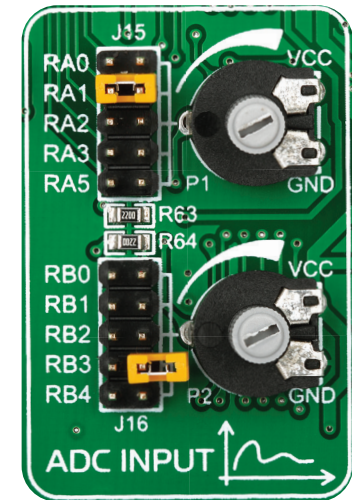


Figure 42: use J15 and J16 jumpers to connect analog input lines with potentiometers P1 and P2

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

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 loses 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 v7a 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.

Enabling I²C EEPROM

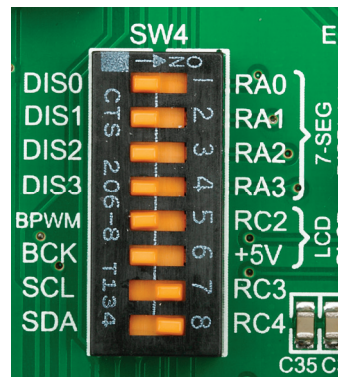


Figure 43: Activate SW4.7 and SW4.8 to connect MCU I²C lines to Serial EEPROM

In order to connect I²C EEPROM to the microcontroller you must enable **SW4.7** and **SW4.8** switches, as shown on **Figure 43**. **1k Ω** 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.

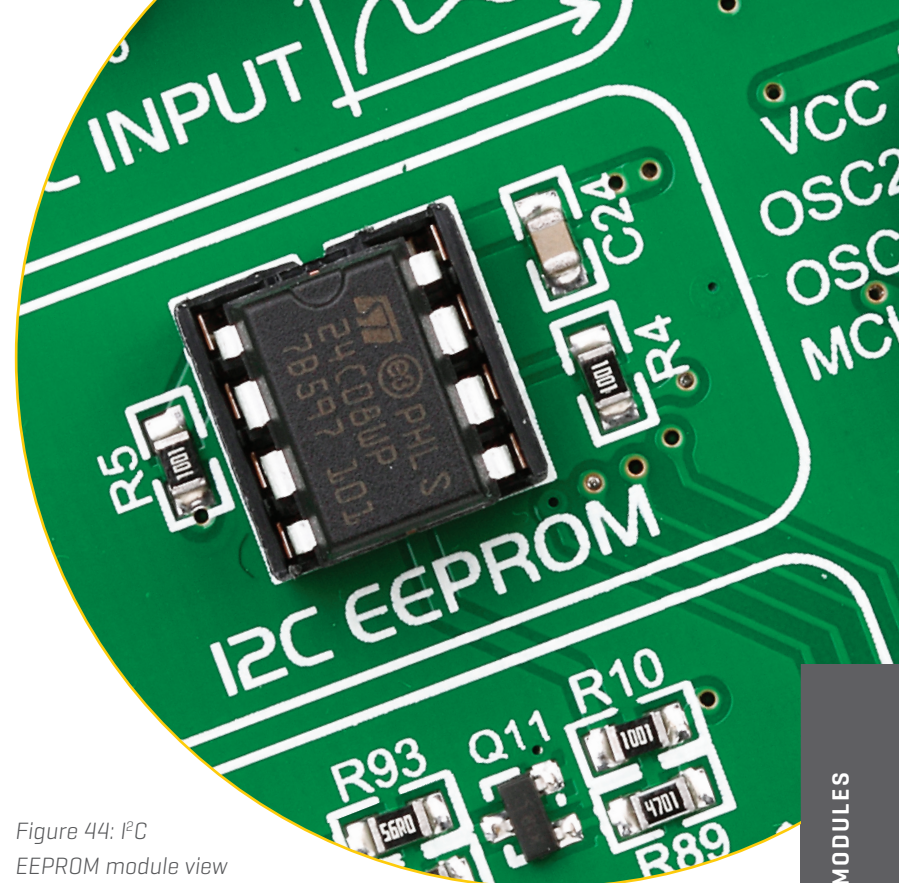


Figure 44: I²C EEPROM module view

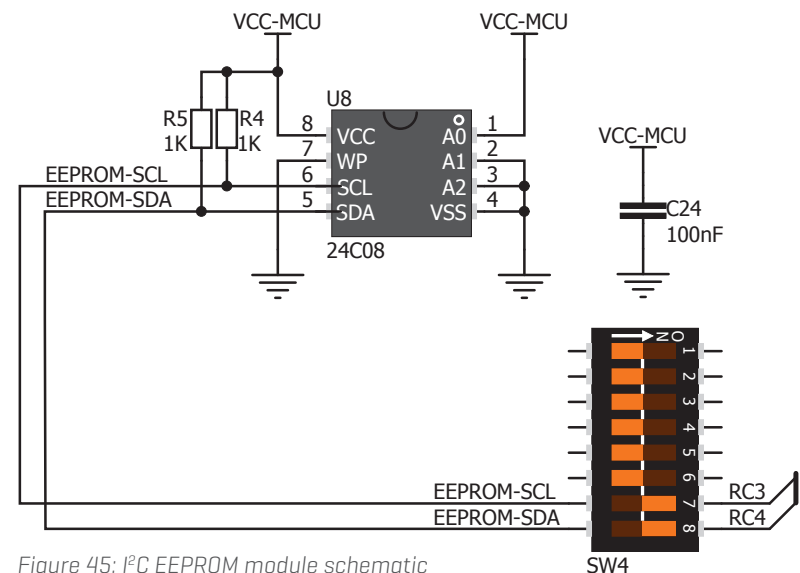


Figure 45: I²C EEPROM module schematic

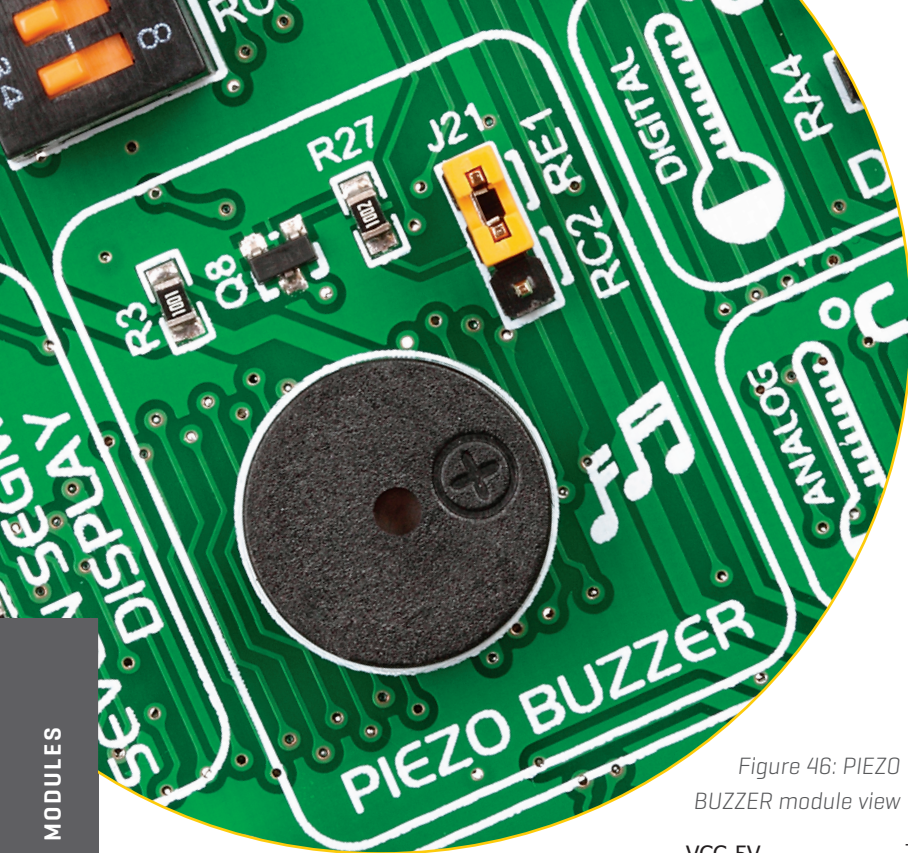


Figure 46: PIEZO BUZZER module view

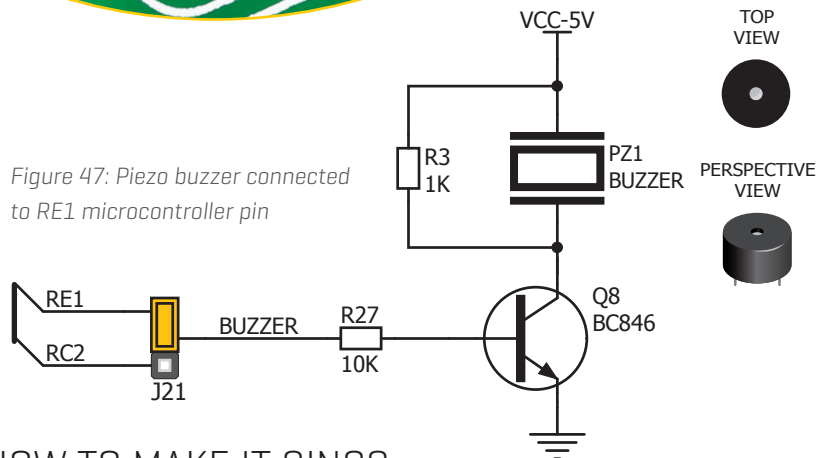


Figure 47: Piezo buzzer connected to RE1 microcontroller pin

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.



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 v7a 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 47]. 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 its best performance) is **3.8kHz**, but you can also use it to create sound in the range between **2kHz** and **4kHz**.

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.



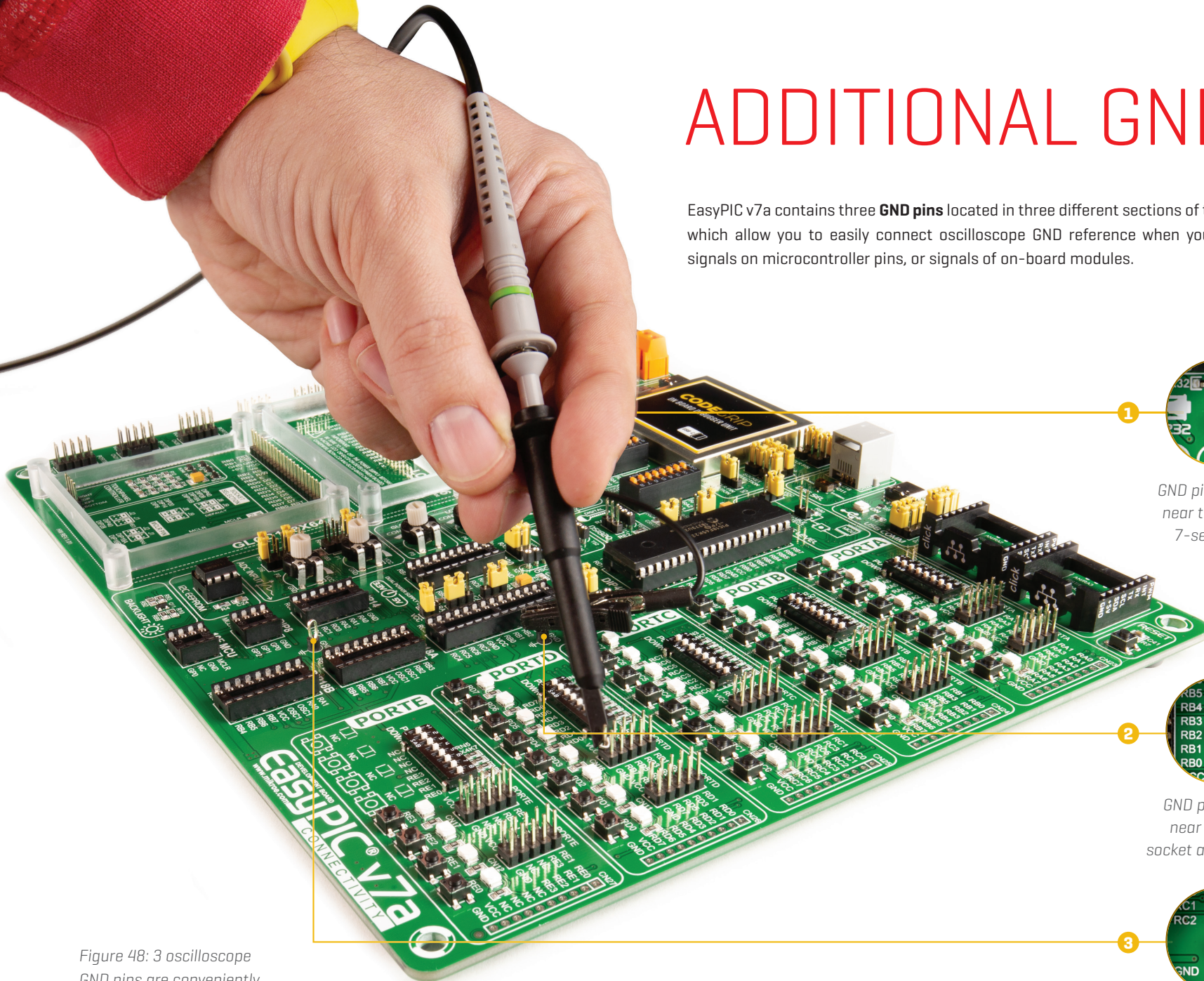
Connected to RE1



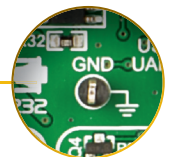
Connected to RC2

ADDITIONAL GNDs

EasyPIC v7a 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 pin located
near the 4 digit
7-seg display

2



GND pin located
near the DIP28
socket and PORTD

3



GND pin located
near the DIP8 and
DIP18B socket

Figure 48: 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 the EasyPIC v7a development board. You have become familiar with its modules, organization, and the programmer/debugger. Now you are ready to start using your new board. We suggest following steps which are highly recommended to begin with.

1 COMPILERS

Easy programming, clean interface, powerful debugging, great support – our compilers come in three different flavors: mikroC PRO for PIC, mikroBASIC PRO for PIC and mikroPASCAL PRO for PIC, offering a complete rapid embedded development solution for these 3 major programming languages.

www.mikroe.com/compilers/compilers-pic

Fast, professional, multiplatform, and multi-architectural **Necto Studio** is already in the air. Support for PIC compiler inside the Necto studio is ready. For more information, please visit: **www.mikroe.com/necto**

3 COMMUNITY

We invite you to join thousands of users of Mikroee development tools. You will find useful projects and tutorials and get help from a large user community. 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.mikroe.com

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 v7a development board. This makes an excellent starting point for future custom projects. Just load the example, read well commented code, and see how it works on hardware.

4 SUPPORT

Mikroe offers free Tech Support to the end of its life span, so if anything goes wrong, we are ready and willing to help. We know how important it is to be able to rely on someone in the moments when we are stuck with our projects for any reason, or facing a deadline. This is why our Support Department, as one of the pillars upon which our company is based, now also offers the Premium Technical Support to business users, ensuring even shorter timeframe for solutions. The WiFi debugger feature offers additional level of technical support, allowing our team to provide help by directly connecting to the end users hardware.

<https://helpdesk.mikroe.com/>

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 board, 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 boards, 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, mikromedia, EasyPIC, Click boards™ and mikroBUS™ 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.

Copyright © MikroElektronika, 2021. All Rights Reserved.



Time-saving embedded tools



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, please contact us at office@mikroe.com