BIGAVR 6™

User manual

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

Development System

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

TO OUR VALUED CUSTOMERS

I want to express my thanks to you for being interested in our products and for having confidence in mikroElektronika.
The primary aim of our company is to design and produce high quality electronic products and to constantly improve the performance thereof in order to better suit your needs.

Nebojsa Matic
General Manager

The Atmel name and logo, the Atmel logo, AVR, AVR (Logo), AVR Freaks, AVR Freaks (Logo), AVR Studio, IDIC, megaAVR, megaAVR (Logo), picoPower ®, tinyAVR ® are trademarks of Atmel Coorporation.

TABLE OF CONTENTS

Introduction to BIG*AVR*6 Development System 4
Key Features 5
1.0. Connecting the System to a PC 6
2.0. Supported Microcontrollers 7
3.0. On-board USB 2.0 AVRprog Programmer 9
4.0. External Programmer AVR ISP mkII 10
5.0 JTAG Connector 11
6.0 Serial EEPROM 11
7.0. Power Supply 12
8.0. Voltage Reference Source 13
9.0. USB Communication Interface 13
10.0. RS-232 Communication Interface 14
11.0. CAN Communication Interface 15
12.0. A/D Converter Test Inputs 16
13.0. DS1820 Temperature Sensor 17
14.0. Real-Time Clock (RTC) 18
15.0. MMC/SD Connector 19
16.0. LEDs 20
17.0. Push Buttons 21
18.0. 2x16 LCD Display 22
19.0. 128x64 Graphic LCD Display 23
20.0. Touch Panel 24
21.0. I/O Ports 25

page

Introduction to BIG*AVR*6 Development System

The *BIGAVR6*™ development system is an extraordinary development tool suitable for programming and experimenting with AVR® microcontrollers from Atmel®. The system includes an on-board programmer providing an interface between the microcontroller and a PC. You are simply expected to write a program in one of the AVR compilers, generate a .hex file and program your microcontroller using the on-board *AVRprog*™ programmer. Numerous on-board modules, such as 128x64 graphic LCD display, alphanumeric 2x16 LCD display, real-time clock, serial EEPROM etc., are provided on the board and allow you to easily simulate the operation of the target device.

AVR DEVELOPMENT BOARD — Full-featured and user-friendly development system for AVR microcontroller based devices

USB 2.0 IN-CIRCUIT PROGRAMMER — USB 2.0 on-board programmer

TOUCHSCREEN — A possibility of connecting a graphic display with touch panel increases the functionality of the development system

MMC/SD MEMORY CARD — A possibility of reading MMC/SD memory cards

GLCD128x64 — Graphic LCD display with backlight

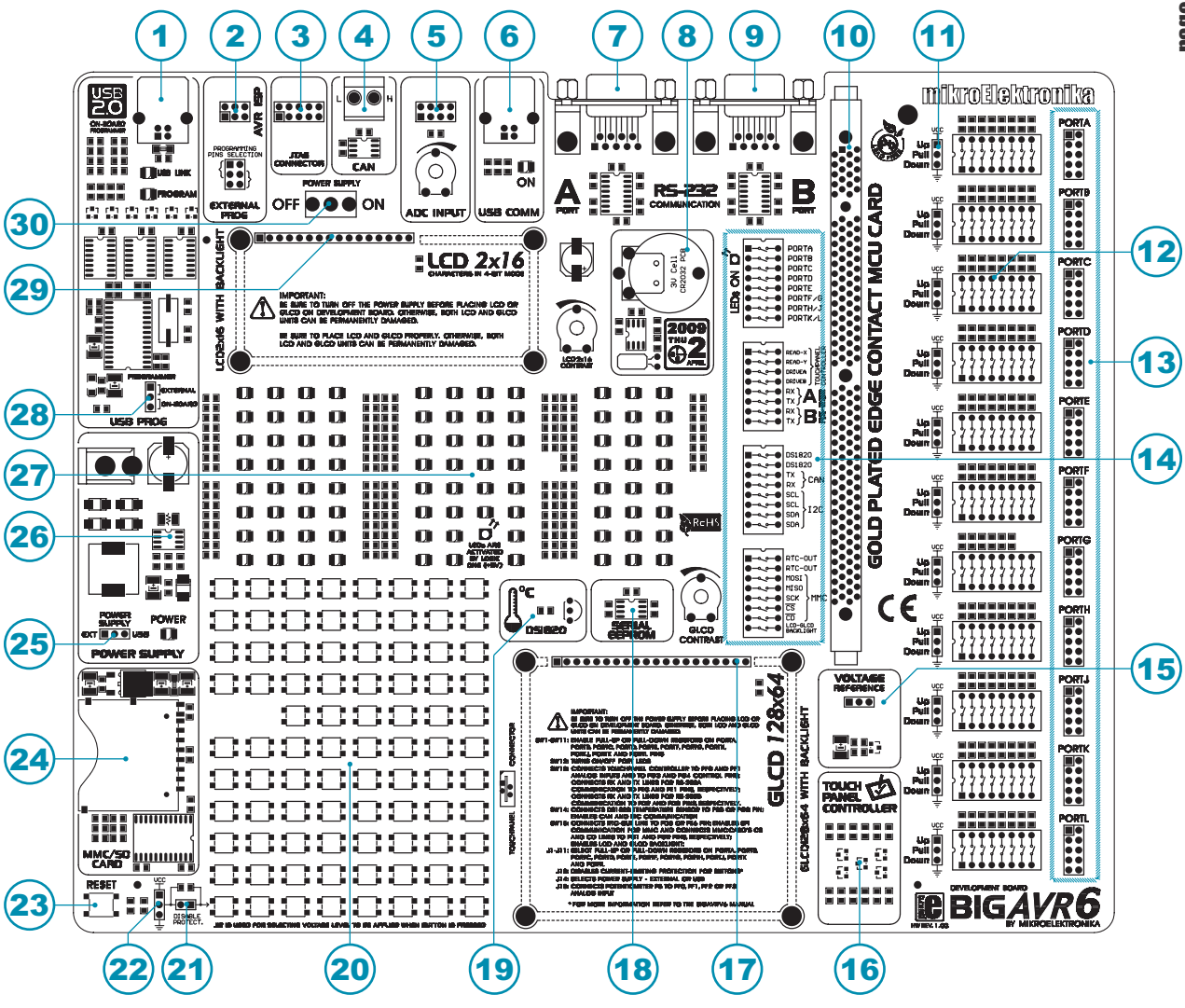
The *AVRflash*™ program for programming provides a complete list of all supported microcontrollers. The latest version of this program with updated list of supported microcontrollers can be downloaded from our website at **www.mikroe.com**

Package contains:

Development System:	BIGAVR6
CD:	product CD with relevant software
Cables:	USB cable
Documentation:	BIGAVR6 and AVRflash manuals, quick guide for installing USB drivers, electrical schematic of the system

System specification:

Power Supply:	over a DC connector (7-23V AC or 9-32V DC); or over a USB cable for programming (5V DC)
Power consumption:	50mA in idle state when all on-board modules are off
Dimension:	26,5 x 22cm (10,4 x 8,6inch)
Weight:	~417g (0.92lbs)



Key Features

1. On-board programmer's USB connector
2. AVR ISP external programmer's connector
3. JTAG connector
4. CAN communication interface
5. A/D converter test inputs
6. USB communication connector
7. Serial communication connector RS-232A
8. Real-time clock
9. Serial communication connector RS-232B
10. MCU card socket
11. Jumper for pull-up/pull-down resistor selection
12. DIP switch enables pull-up/pull-down resistors
13. I/O port connectors
14. DIP switch turns on/off on-board modules
15. Voltage reference source
16. Touch panel controller
17. Graphic LCD display connector
18. Serial EEPROM
19. Temperature sensor connector
20. Push buttons simulate microcontroller digital inputs
21. Jumper for protective resistor shortening
22. Jumper for selecting push buttons' logic state
23. Reset button
24. MMC/SD card connector
25. Power supply source selector
26. Power supply voltage regulator
27. 86 LEDs indicate pins' logic state
28. Programmer selector
29. Alphanumeric LCD display connector
30. POWER SUPPLY switch

page

1.0. Connecting the System to a PC

Step 1:

Follow the instructions provided in the relevant manuals and install the *AVRflash* program and USB drivers from the product CD. USB drivers are necessary for the proper operation of the on-board programmer.
In case you already have one of the MikroElektronika's AVR compilers installed on your PC, there is no need to reinstall USB drivers as they are already installed along with the compiler.

Step 2:

Use the USB cable to connect the *BIGAVR6* development system to a PC. One end of the USB cable, with a USB connector of **B** type, should be connected to the development system, as shown in Figure 1-2, whereas the other end of the cable with a USB connector of **A** type should be connected to a PC. When establishing a connection, make sure that jumper J14 is placed in the USB position as shown in Figure 1-1.

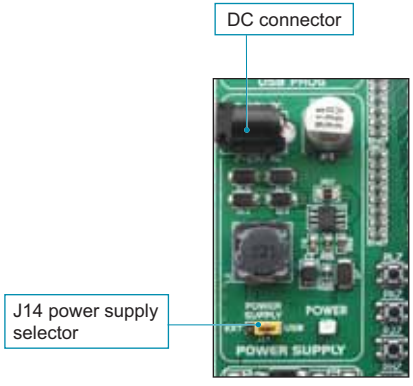
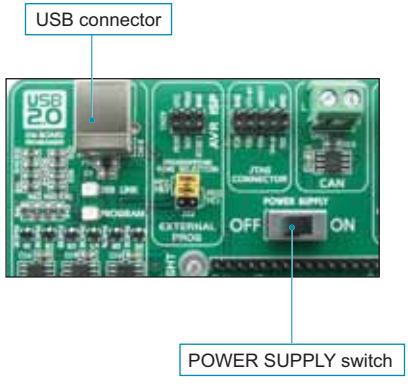


Figure 1-1: Power supply



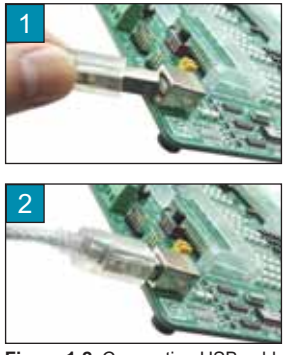


Figure 1-2: Connecting USB cable

Step 3:

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

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

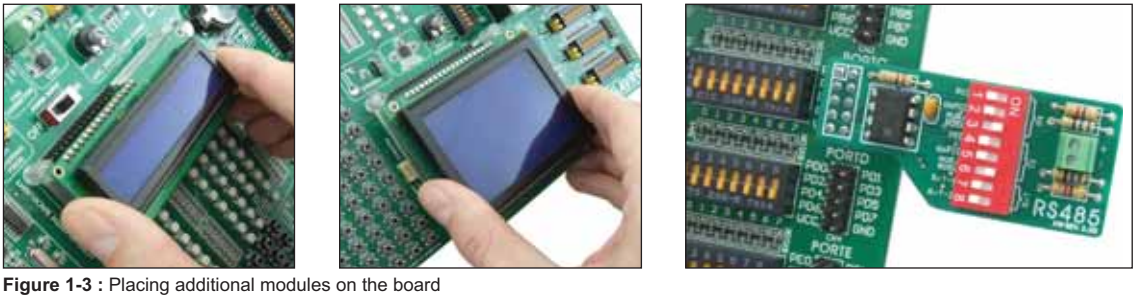


Figure 1-3 : Placing additional modules on the board

2.0. Supported Microcontrollers

The *BIGAVR6* development system provides a DIMM-168P connector to place an MCU card into. This development system comes with an MCU card with the ATMEGA128 microcontroller in 64-pin TQFP package soldered on it, Figure 2-3. Besides, the MCU card alone provides an oscillator as well as 80 soldering pads connected to microcontroller pins. Each pad is marked the same as the pin it is connected to. Soldering pads also make connection between the MCU card and target device's modules easy.

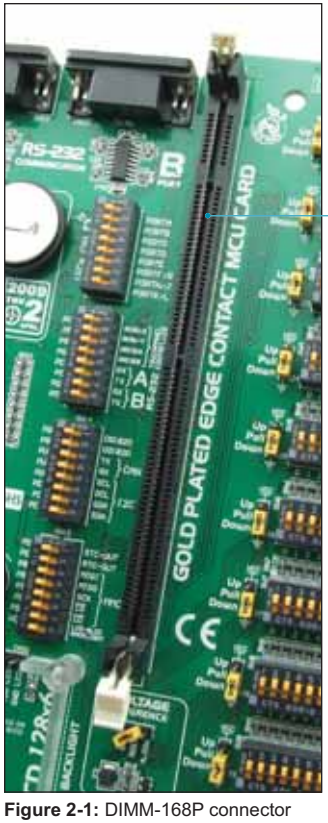


Figure 2-1: DIMM-168P connector

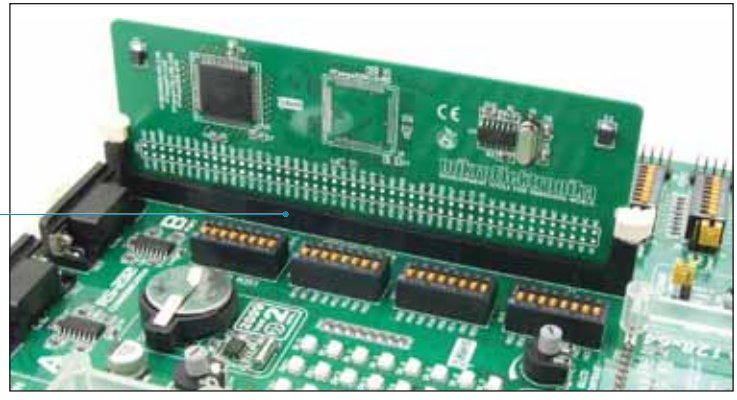


Figure 2-2: DIMM-168P connector with the MCU card inserted

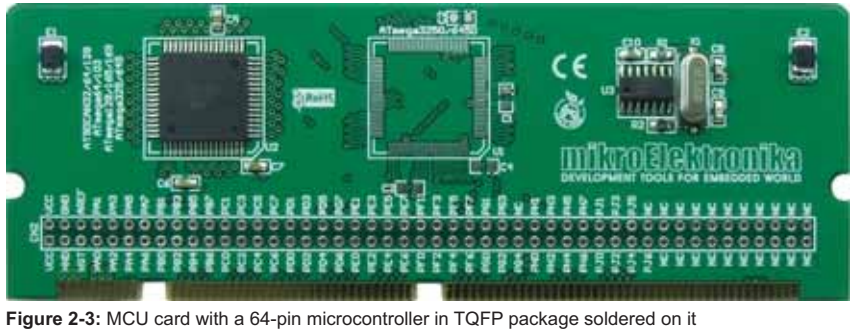


Figure 2-3: MCU card with a 64-pin microcontroller in TQFP package soldered on it

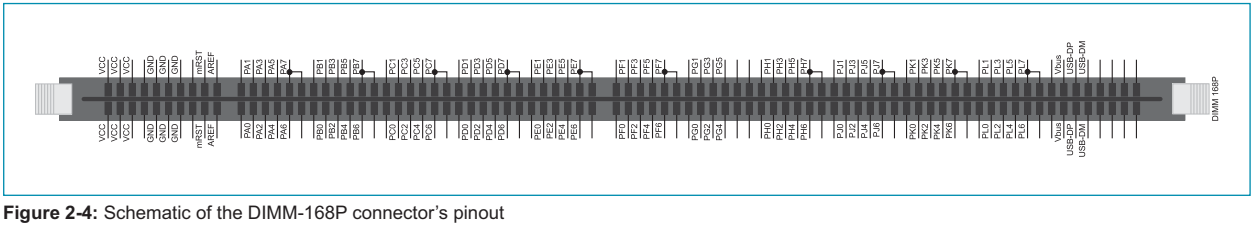
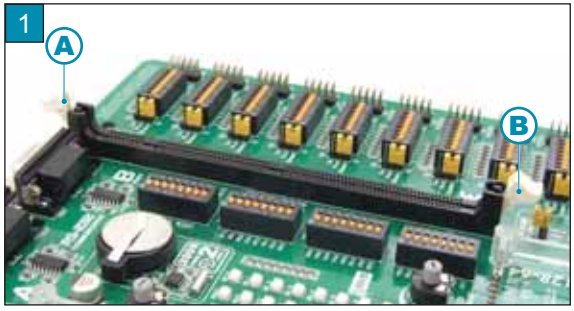
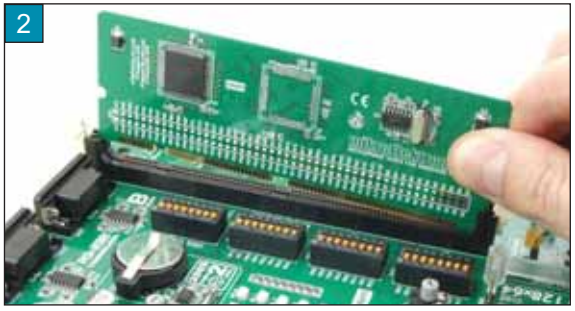


Figure 2-4: Schematic of the DIMM-168P connector's pinout

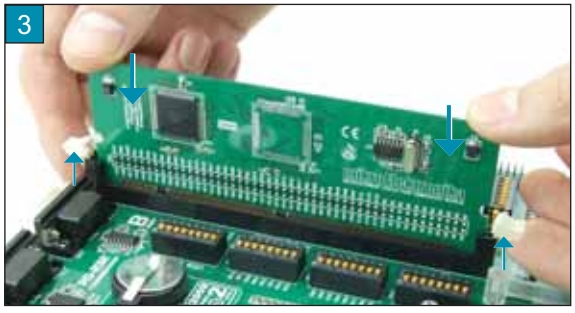
Placing MCU card into the DIMM-168P connector is performed as follows:



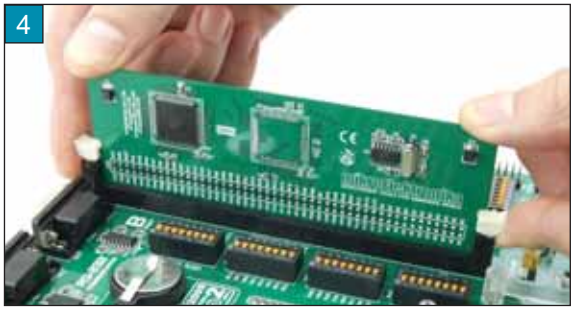
Open extraction levers A and B



Place the MCU card into the DIMM-168P connector



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



Close the extraction levers when the MCU card is properly placed into the connector

Extraction levers used to fix MCU card in the 'open' position

Extraction levers used to fix MCU card in the 'closed' position

In addition to the MCU card with microcontrollers in 64-pin TQFP package, there are also MCU cards with microcontrollers in 100-pin TQFP package which can be ordered separately. They are placed into the appropriate connector in the same manner as described above.

3.0. On-board USB 2.0 AVRprog Programmer

A programmer is a necessary tool when working with microcontrollers. The *BIGAVR6* has an on-board *AVRprog* programmer which provides an interface between the microcontroller and the PC. The *AVRflash* program is used for loading a .hex file into the microcontroller. Figure 3-3 shows connection between the compiler, *AVRflash* program and the microcontroller.

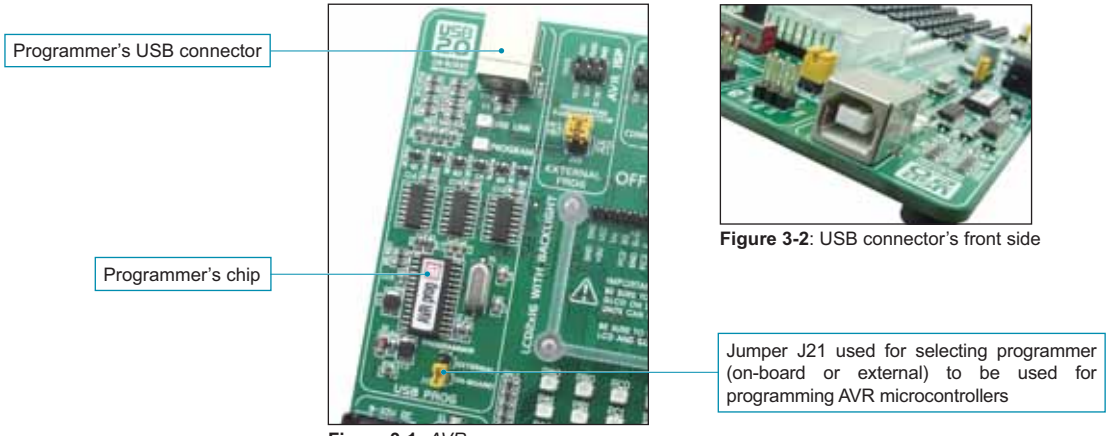


Figure 3-1: *AVRprog* programmer

Figure 3-2: USB connector's front side

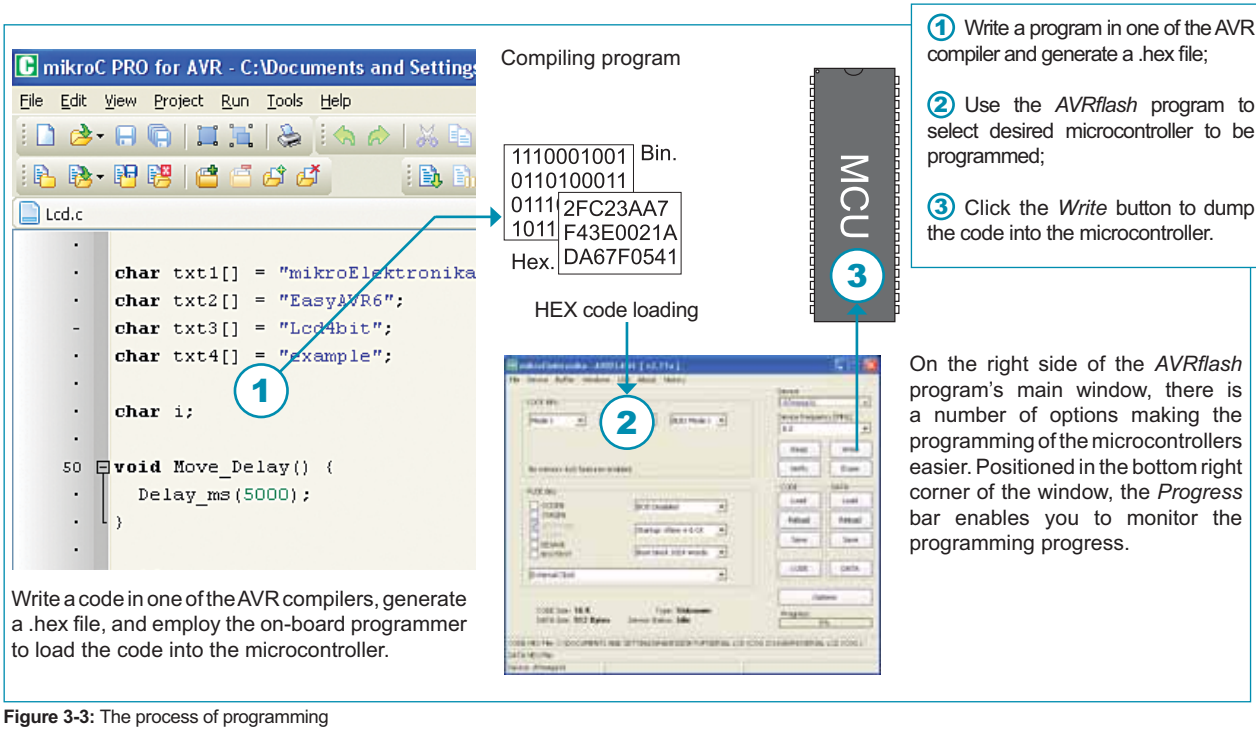


Figure 3-3: The process of programming

NOTE: For more information on the *AVRprog* programmer refer to the relevant manual provided with the system.

page

AVR microcontrollers are programmed using SPI serial communication which employs the following microcontroller pins MISO, MOSI and SCK.

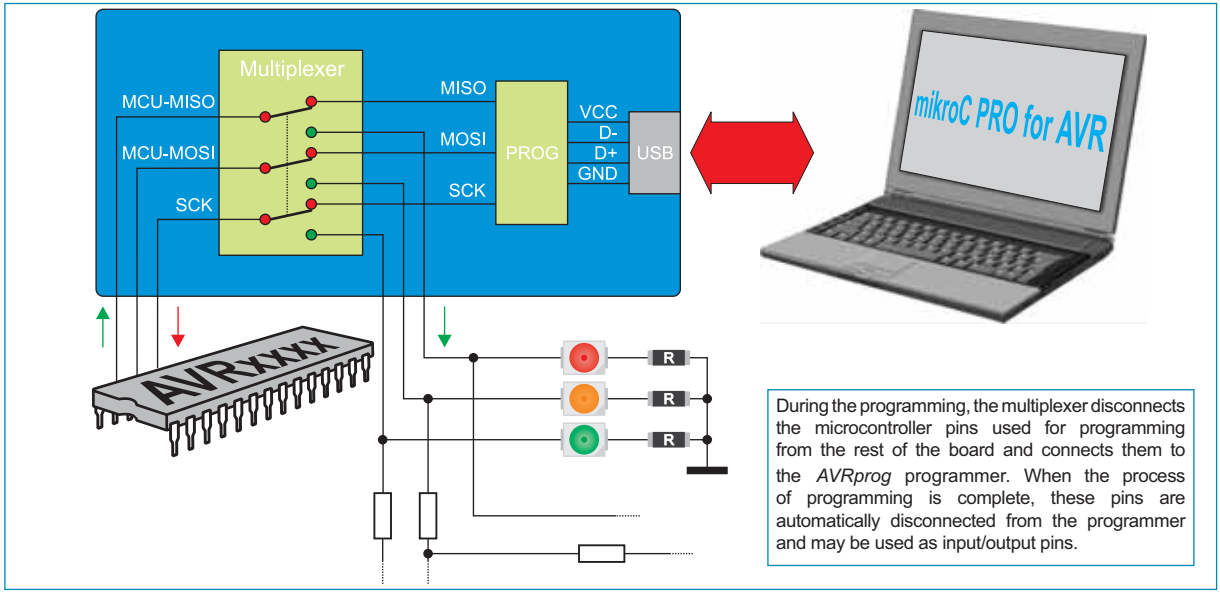
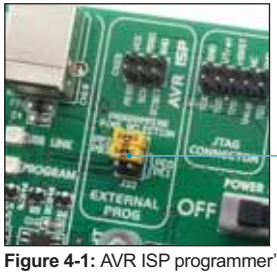


Figure 3-4: The principle of programmer's operation

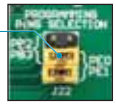
4.0. External Programmer AVR ISP mkII

In addition to the on-board programmer, the *BIGAVR6* development system may also use an external programmer *AVR ISP* from Atmel for programming microcontrollers. It is linked to the connector *AVR ISP*. Prior to connecting and using the external programmer, it is necessary to place jumper J21 in the EXTERNAL position. Then you should use jumper J22 to select the appropriate microcontroller socket.



Jumpers for selecting pins to be fed with programming signal

The position of jumper J22 when the external programmer is used for programming microcontrollers in 100-pin TQFP package



The position of jumper J22 when the external programmer is used for programming microcontrollers in 64-pin TQFP package

Figure 4-1: AVR ISP programmer's connector

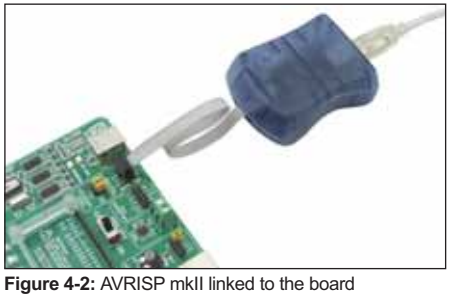


Figure 4-2: AVRISP mkII linked to the board

Jumper J21 set to the *EXTERNAL* position enables the operation of the external programmer *AVRISP*

Jumper J21 set to the *ON-BOARD* position enables the operation of the on-board programmer

5.0. JTAG Connector

JTAG ICEmkII is a programmer/debugger for AVR microcontrollers provided with a built-in JTAG interface. The JTAG ICEmkII is primarily intended for use with the *AVR Studio* program. The JTAG interface built-in into AVR microcontrollers is a modified version of the original JTAG interface, which enables the contents of internal EEPROM and FLASH memory to be altered (programming microcontrollers).

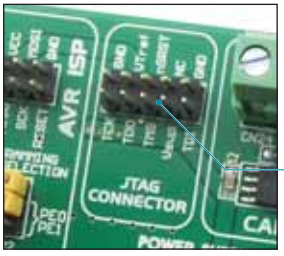


Figure 5-1: JTAG connector

The JTAG ICEmkII programmer/ debugger employs a male 2x5 connector to be interfaced to the development system

Figure 5-2: JTAGICE mkII connected to the board

The JTAG connector is directly connected to microcontroller pins so that it is not necessary to perform jumper settings as is the case with *AVRprog* and *AVR ISPmkII* programmers.

6.0. Serial EEPROM

EEPROM (Electrically Erasable Programmable Read-Only Memory) is a built-in memory module used to store data that must be saved when the power goes off. The 24AA01 circuit can store up to 1Kbit data and uses serial I^2C connection to communicate with the microcontroller via PD0 and PD1 or PE4 and PE5 pins. In order to establish such connection between EEPROM and microcontroller, it is necessary to set switches 5 and 7 on the DIP switch SW14 to the ON position. Switches 6 and 8 may also be used for this purpose, which depends on which pins you want to use for serial communication.

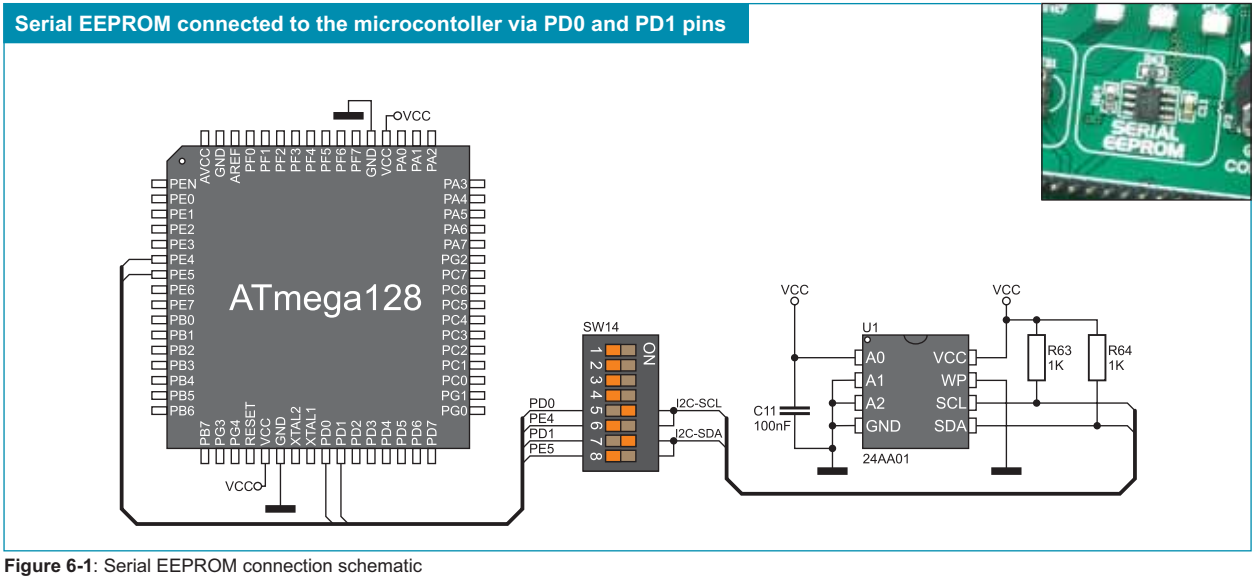


Figure 6-1: Serial EEPROM connection schematic

page

7.0. Power supply

The *BIGAVR6* development system may use one of two power supply sources:

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

The MC34063A voltage regulator and Gretz rectifier are used to enable external power supply voltage to be either AC (in the range of 7V to 23V) or DC (in the range of 9V to 32V). Jumper J14 is used as a selector for a power supply source. To make advantage of the USB power supply, jumper J14 should be placed in the USB position. When using external power supply, jumper J14 should be placed in the EXT position. The development system is turned on/off by switching the position of the POWER SUPPLY switch.

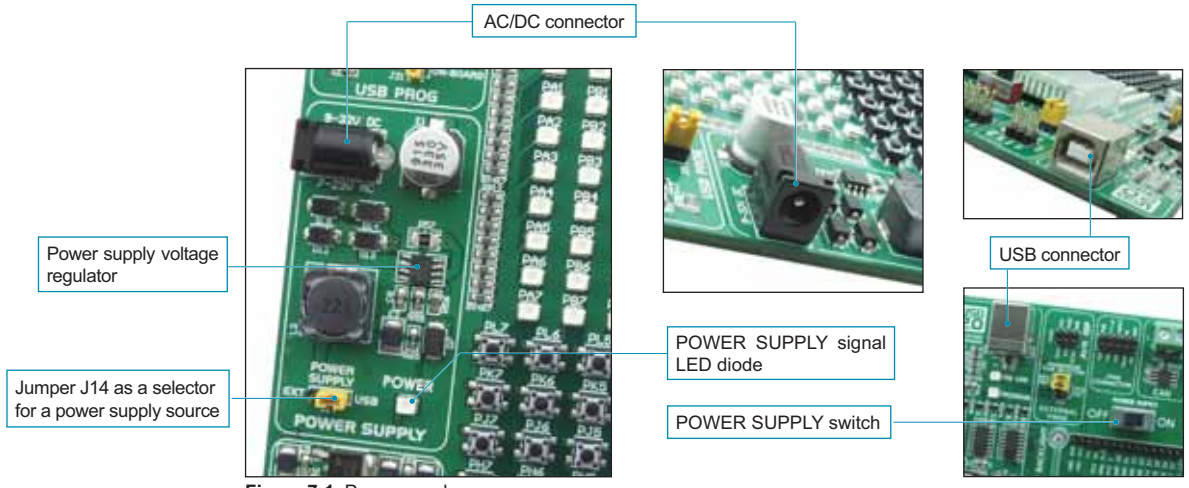


Figure 7-1: Power supply

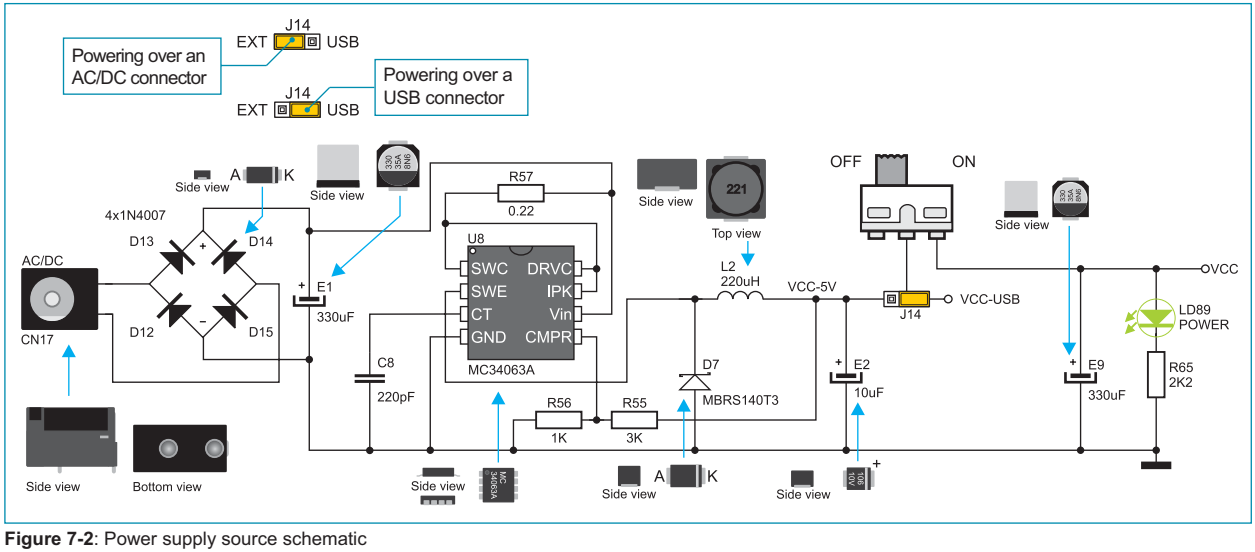


Figure 7-2: Power supply source schematic

8.0. Voltage Reference Source

The *BIGAVR6* development system is provided with the MCP1541 circuit which generates voltage reference used for A/D conversion. The microcontroller is supplied with such voltage, the value of which is 4.096V, through the AREF pin. In addition, the AREF pin can also be supplied with the 5V power supply voltage. The position of jumper J18 determines which of these voltages is to be supplied on the AREF pin.

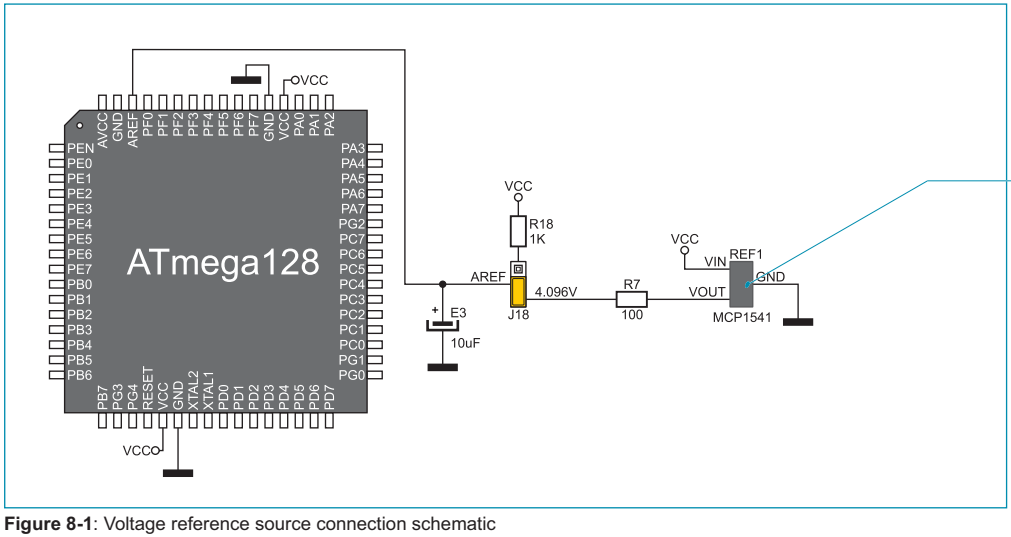


Figure 8-1: Voltage reference source connection schematic

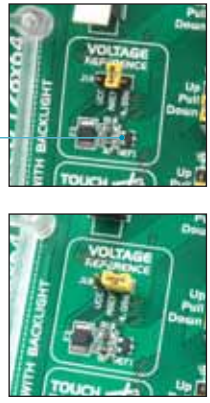


Figure 8-2: Jumper J18 placed to provide 4.096V voltage reference

9.0. USB Communication Interface

The CN25 USB connector enables AVR microcontrollers with a built-in USB interface to be connected to peripheral devices. The microcontroller is connected to the CN25 USB connector via USB-DM and USB-DP lines. The function of the LED diode marked as ON is to indicate connection between USB devices

Figure 9-1: USB connector

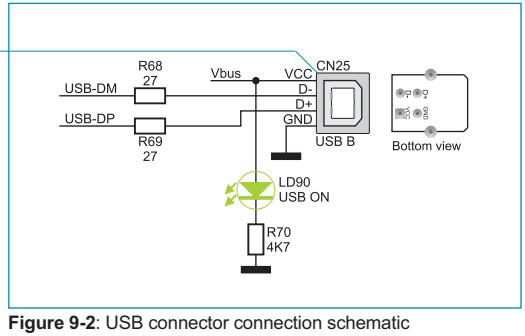


Figure 9-2: USB connector connection schematic

page

10.0. RS-232 Communication Interface

USART (*Universal Synchronous/Asynchronous Receiver/Transmitter*) is one of the most common ways of exchanging data between the PC and peripheral units. RS-232 serial communication is performed through a 9-pin SUB-D connector and the microcontroller USART module. The *BIGAVR6* provides two RS-232 ports, RS-232A and RS-232B. Use switches marked as RX232-A (PE0 pin) and TX232-A (PE1 pin) on the DIP switch SW13 to enable port RS-232A. Likewise, use switches marked as RX232-B (PD2 pin) and TX232-B (PD3 pin) on the DIP switch SW13 to enable port RS-232B. The microcontroller pins used in such communication are marked as follows: RX - *receive data line* and TX - *transmit data line*. Data rate goes up to 115 kbps.

In order to enable the USART module of the microcontroller to receive input signals which meet the RS-232 standard, it is necessary to adjust voltage levels using an IC circuit such as MAX202.

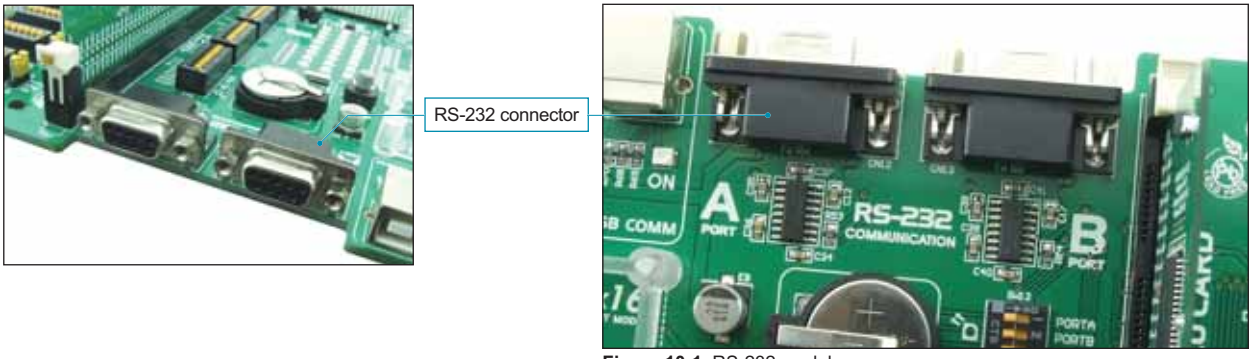


Figure 10-1: RS-232 module

The function of the switches 5-8 on the DIP switch SW13 is to determine which of the microcontroller pins are to be used as RX and TX lines, Figure 10-2.

Port RS-232A is connected to the microcontroller via PE0 and PE1 pins

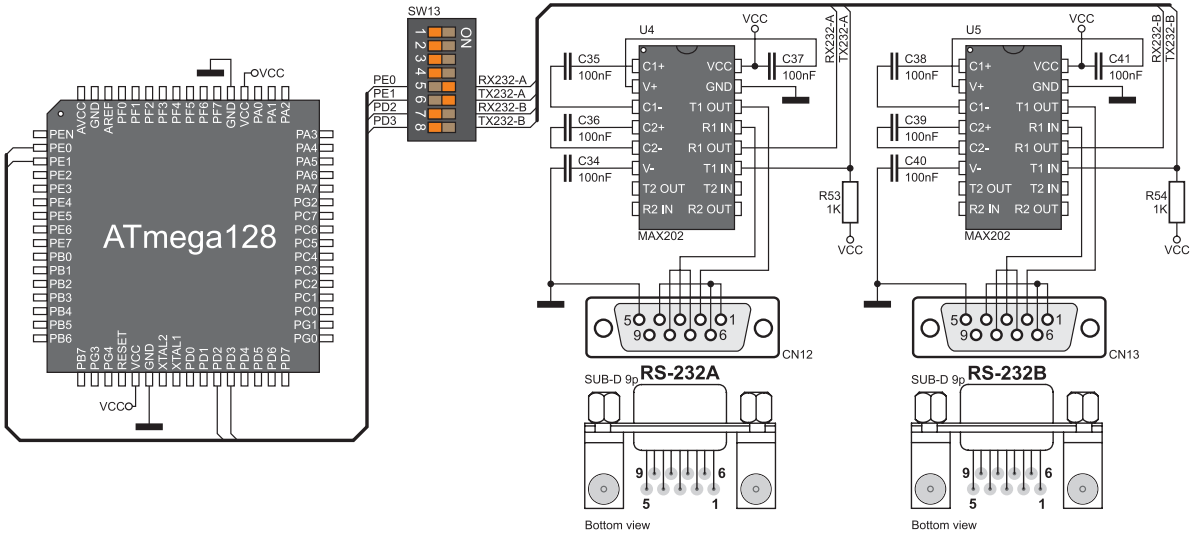


Figure 10-2: RS-232 module connection schematic

11.0. CAN Communication Interface

CAN (Controller Area Network) is a communication standard primarily intended for use in automotive industry. It enables the microcontroller to communicate to a car device without using a host PC. In addition, such communication is widely used in industrial automation. The *BIGAVR6* uses the MCP2551 circuit for CAN communication. This circuit provides an interface between the microcontroller and some peripheral device. To enable connection between the microcontroller and MCP2551, it is necessary to set switches 3 and 4 on the DIP switch SW14 to the ON position.

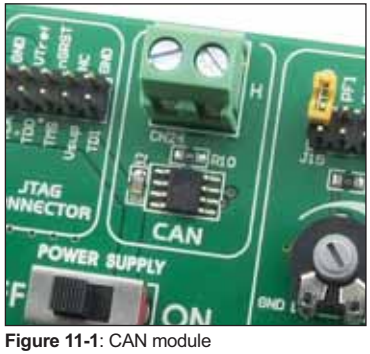


Figure 11-1: CAN module

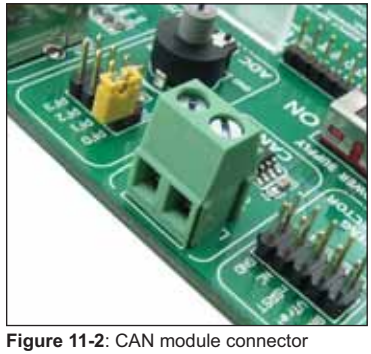


Figure 11-2: CAN module connector

CAN communication is enabled via DIP switch SW14

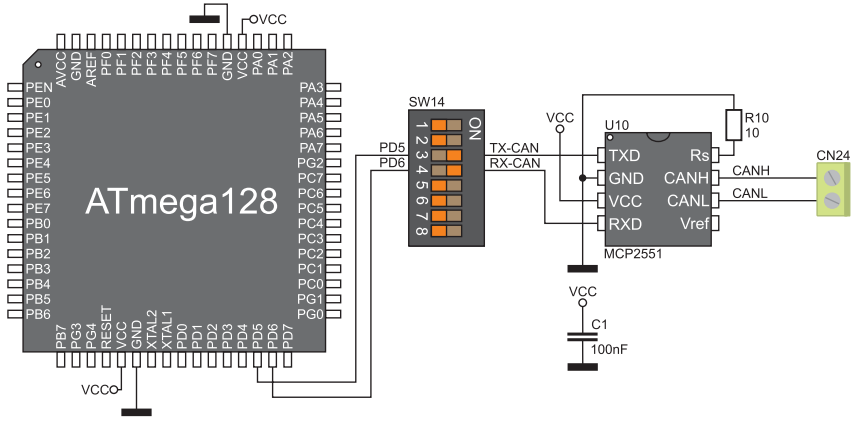


Figure 11-3: Microcontroller and MCP2551 connection schematic

page

12.0. A/D Converter Test Inputs

An A/D converter is used for converting an analog signal into the appropriate digital value. A/D converter is linear, which means that converted number is linearly dependent on the input voltage value. The A/D converter built into the microcontroller converts an analog voltage value into a 10-bit number. Potentiometer P5 enables voltage to vary between 0 and 5V. The microcontroller with a built-in A/D converter is supplied with this voltage via test inputs. Jumper J15 is used for selecting one of the following pins PF0, PF1, PF2 or PF3 for A/D conversion. Resistor R12 has a protective function and is used to limit current flow through the potentiometer or the microcontroller pin.

Figure 12-1: ADC (jumper idefault position)

Figure 12-2: Pin PF0 used as input pin for A/D conversion

A/D conversion is performed via the PF0 pin

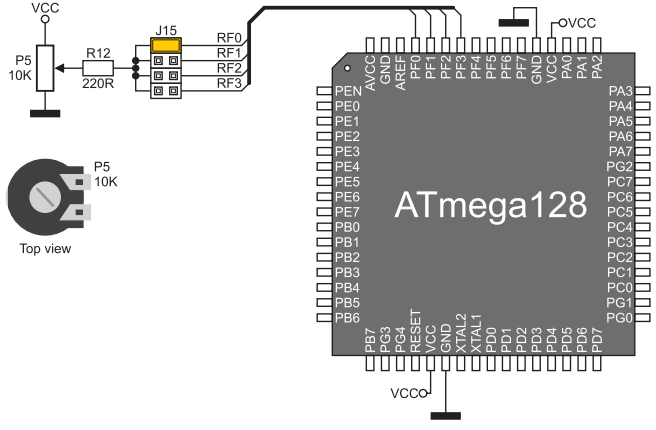


Figure 12-3: Microcontroller and A/D converter test inputs connection schematic

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

13.0. DS1820 Temperature Sensor

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

The DS1820 is a temperature sensor that uses 1-wire communication. It is capable of measuring temperatures within the range of -55 to 125°C and provides ±0.5°C accuracy for temperatures within the range of -10 to 85°C. A power supply voltage of 3V to 5.5V is required for its operation. It takes maximum 750ms for the DS1820 to calculate temperature with a 9-bit resolution. The *BIGAVR6* development system provides a separate socket for the DS1820. It uses either PB0 or PG0 pin for communication with the microcontroller, which depends on the position of switches 1 and 2 on the DIP switch SW14.

Figure 13-1: DS1820 connector (DS1820 is not placed)

Figure 13-2: DS1820 connector with the DS1820 temperature sensor plugged into

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

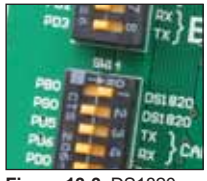


Figure 13-3: DS1820 is connected to the microcontroller via the PB0 pin

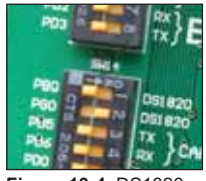


Figure 13-4: DS1820 is connected to the microcontroller via the PG0 pin

Temperature sensor is connected to the microcontroller via the PG0 pin

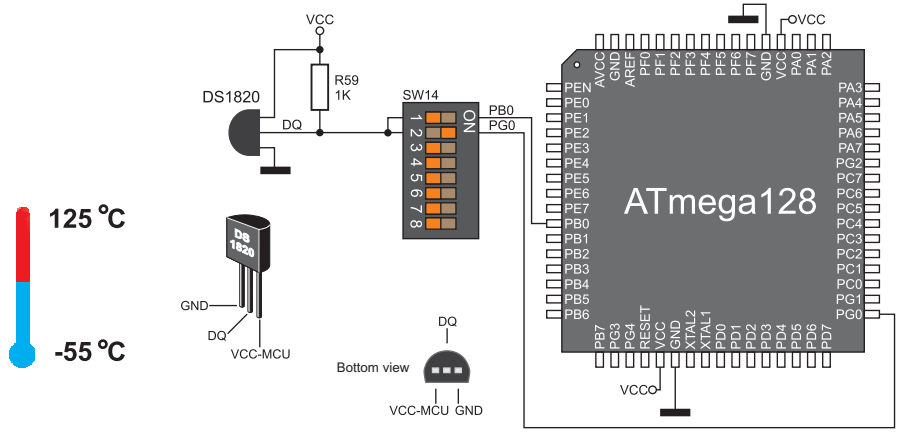


Figure 13-5: DS1820 and microcontroller connection schematic

page

14.0. Real-Time Clock (RTC)

The real-time clock is widely used in alarm devices, industrial controllers, consumer devices etc. Thanks to the DS1307 circuit, the *BIGAVR6* development system is capable of keeping the real time. The main features of the real-time clock are as follows:

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

The real-time clock provided on the *BIGAVR6* development system is used to generate an interrupt at pre-set time. In order to establish connection between the microcontroller and real-time clock it is necessary to set switches PD0 and PD1 on the DIP switch SW14 as well as switch PD3 on the DIP switch SW15 to the ON position. Switches PE4, PE5 and PE6 may be optionaly used here.

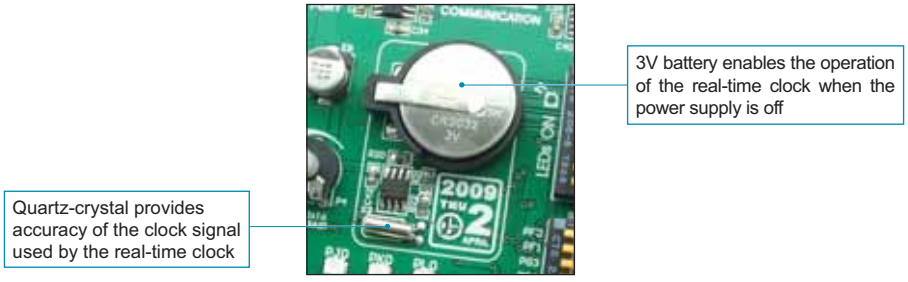


Figure 14-1: Real-time clock

Real-time clock is connected to the microcontroller via PD0, PD1 and PD3 pins

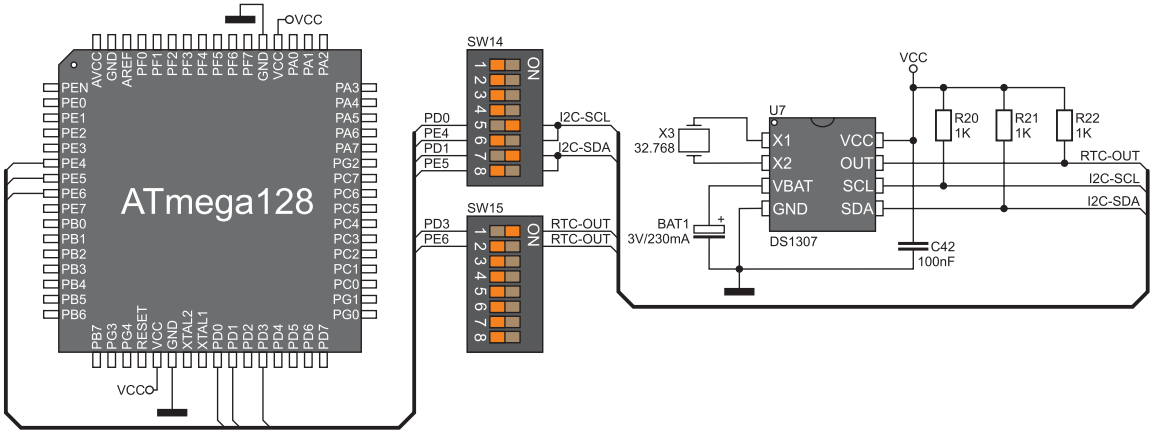


Figure 14-2: Real-time clock and microcontroller connection schematic

15.0. MMC/SD Connector

The MMC/SD connector is used to enable memory cards to be interfaced to the microcontroller. To enable communication between memory card and microcontroller, it is necessary to adjust their voltage levels. Memory card is powered by the 3.3V power supply voltage (VCC-MMC) generated by the REG1 voltage regulator, whereas the value of the microcontroller power supply voltage is 5V (VCC). The 74LVCC3245 bus transceiver is used here to adjust these voltage levels. In addition, communication between memory card and microcontroller can be established only if switches 3, 4, 5, 6 and 7 on the DIP switch SW15 are set to the ON position.

Figure 15-1: MMC/SD connector

Figure 15-2: MMC/SD memory card

MMC/SD card communicates to the microcontroller via MISO, MOSI, SCK, MMC-CS and MMC-CD lines

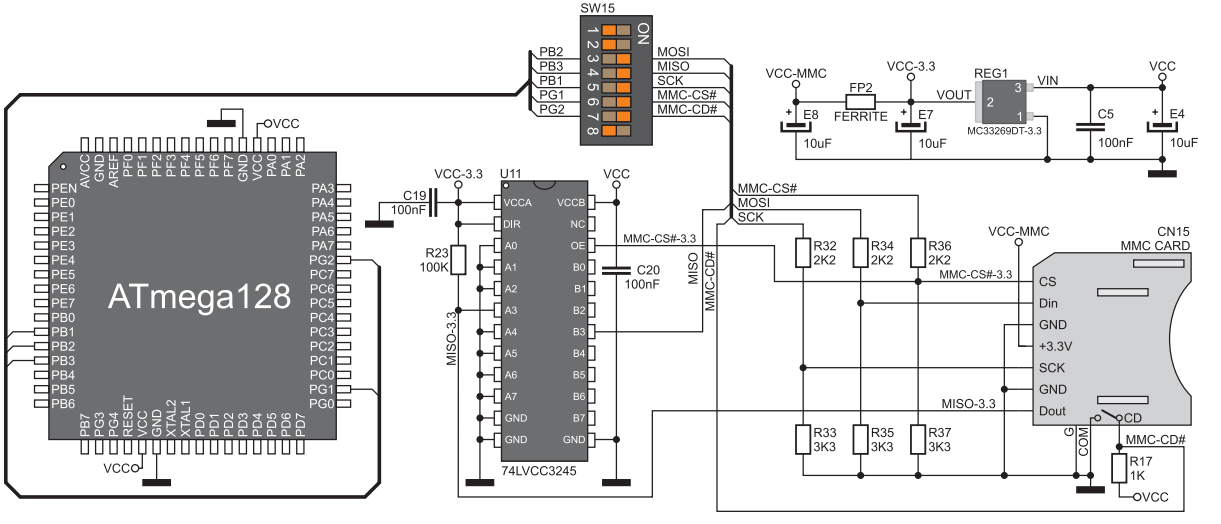
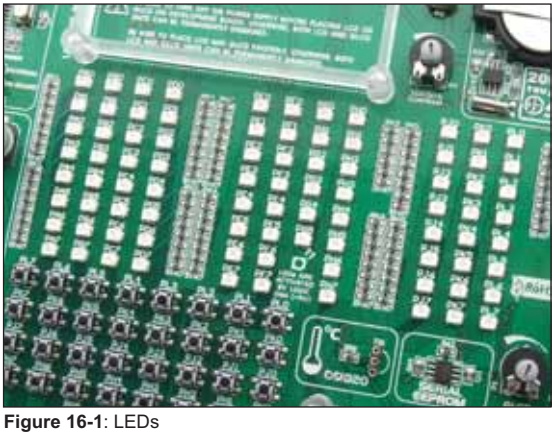


Figure 15-3: MMC/SD connector and microcontroller connection schematic

16.0. LEDs

LED (Light-Emitting Diode) is a highly efficient electronic light source. When connecting LEDs, it is necessary to use a current limiting resistor. A common LED diode voltage is approximately 2.5V, while the current varies from 1 to 20mA depending on the type of LED. The *BIGAVR6* uses LEDs with current I=1mA.

There are 86 LEDs on the *BIGAVR6* development system which visually indicate the state of each microcontroller I/O pin. An active LED indicates that a logic one (1) is present on the pin. In order to enable the pin state to be shown, it is necessary to select appropriate port (PORTA, PORTB, PORTC, PORTD, PORTE, PORTF, PORTG, PORTH, PORTJ PORTK or PORTL) using the DIP switch SW12.



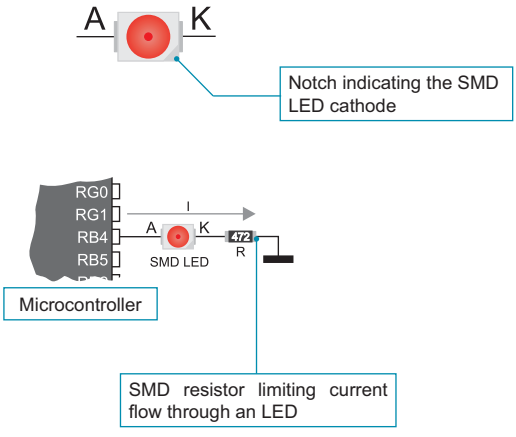


Figure 16-1: LEDs

PORTA LEDs are turned on

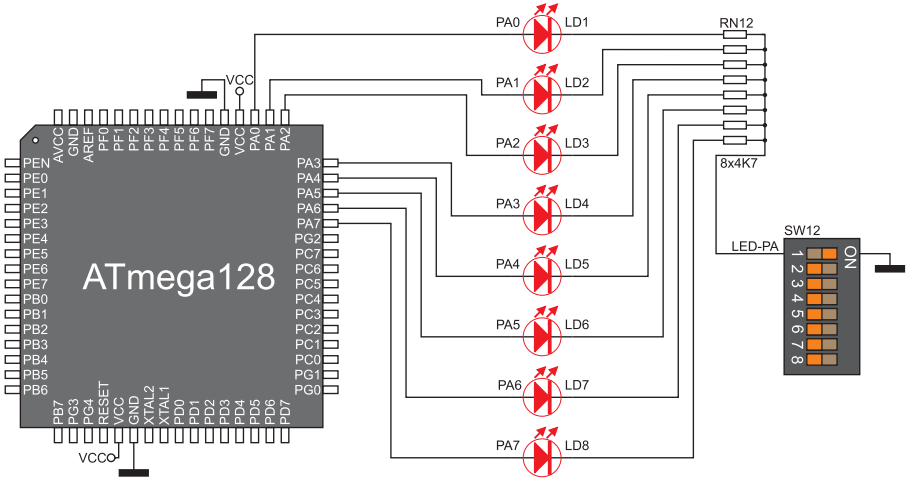


Figure 16-2: LED diodes and port PORTA connection schematic

17.0. Push Buttons

The logic state of all microcontroller input pins may be changed by means of push buttons. Jumper J12 is used to determine the logic state to be applied to the desired microcontroller pin by pressing appropriate push button. The function of the protective resistor is to limit the maximum current, thus preventing the development system and peripheral modules from being damaged in case a short circuit occurs. If needed, advanced users may shorten such resistor using jumper J13. Right next to the push buttons, there is a RESET button which is used to provide the MCLR pin with the microcontroller reset signal over the on-board programmer.

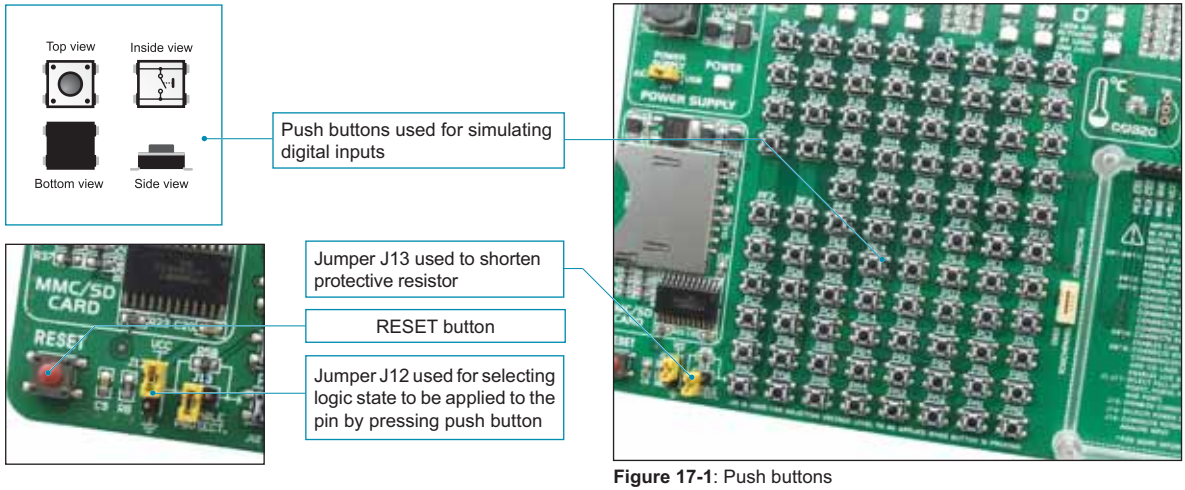


Figure 17-1: Push buttons

By pressing any push button when jumper J12 is in the VCC position, a logic one (5V) will be applied to the appropriate microcontroller pin as shown in Figure 17-2.

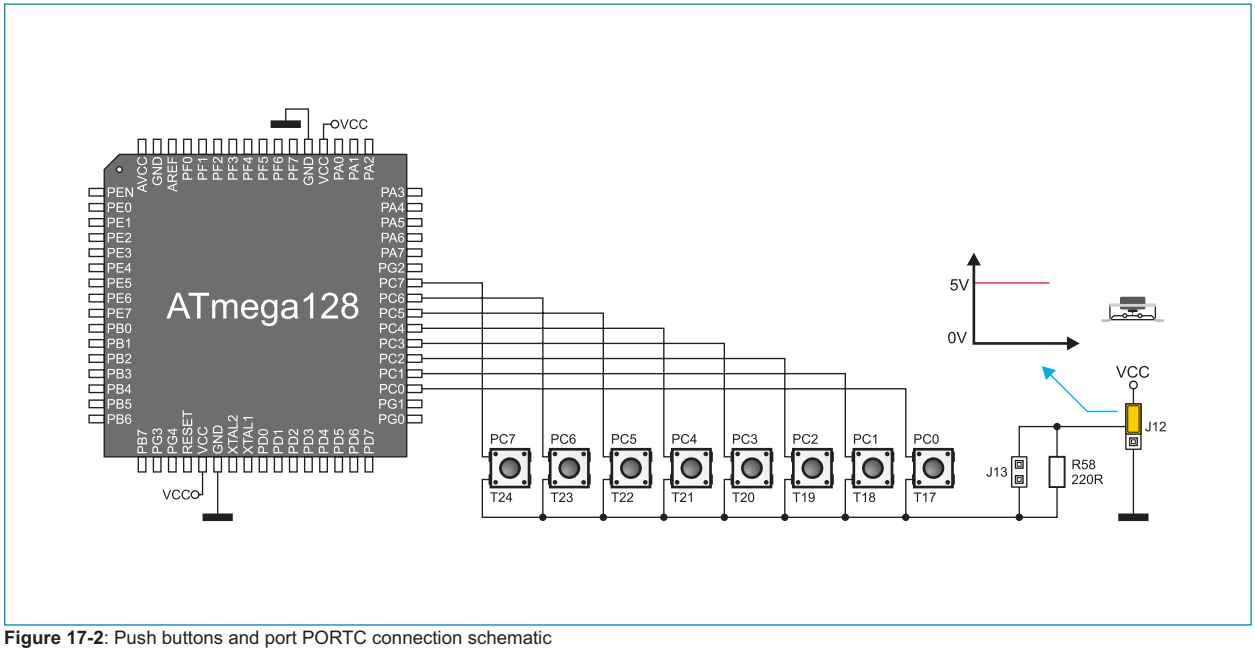


Figure 17-2: Push buttons and port PORTC connection schematic

page

18.0. 2x16 LCD Display

The *BIGAVR6* development system provides an on-board connector for the alphanumeric 2x16 LCD display. Such connector is linked to the microcontroller via port PORTC. Potentiometer P4 is used to adjust display contrast. Switch 8 (LCD-GLCD BACKLIGHT) on the DIP switch SW15 is used to turn the display backlight on/off.
Communication between the LCD display and the microcontroller is performed in a 4-bit mode. Alphanumeric digits are displayed in two lines each containing up to 16 characters of 7x5 pixels.

Figure 18-1: Alphanumeric LCD display connector

Figure 18-2: Alphanumeric 2x16 LCD display

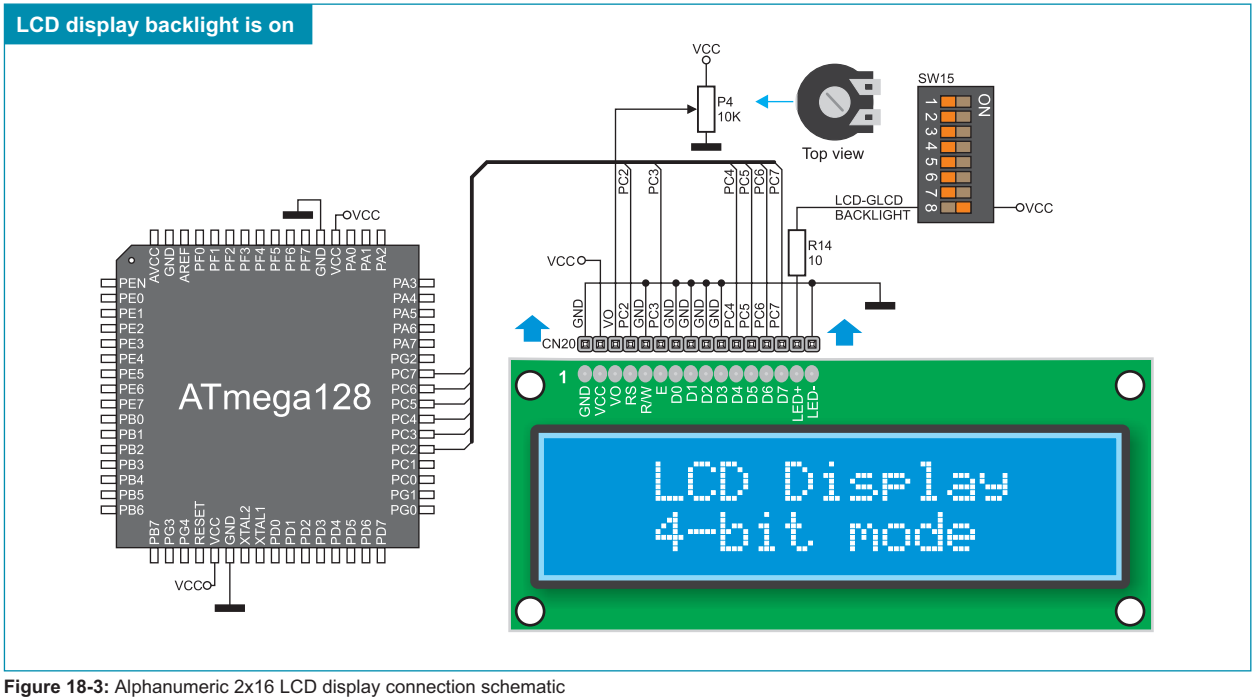


Figure 18-3: Alphanumeric 2x16 LCD display connection schematic

19.0. 128x64 Graphic LCD Display

128x64 graphic LCD display (GLCD) is connected to the microcontroller via PORTA and PORTE ports and enables graphic content to be displayed. It has the screen resolution of 128x64 pixels, which allows diagrams, tables and other graphic content to be displayed. Potentiometer P3 is used for the GLCD display contrast adjustment. Switch 8 (LCD-GLCD BACKLIGHT) on the DIP switch SW15 is used to turn the display backlight on/off.

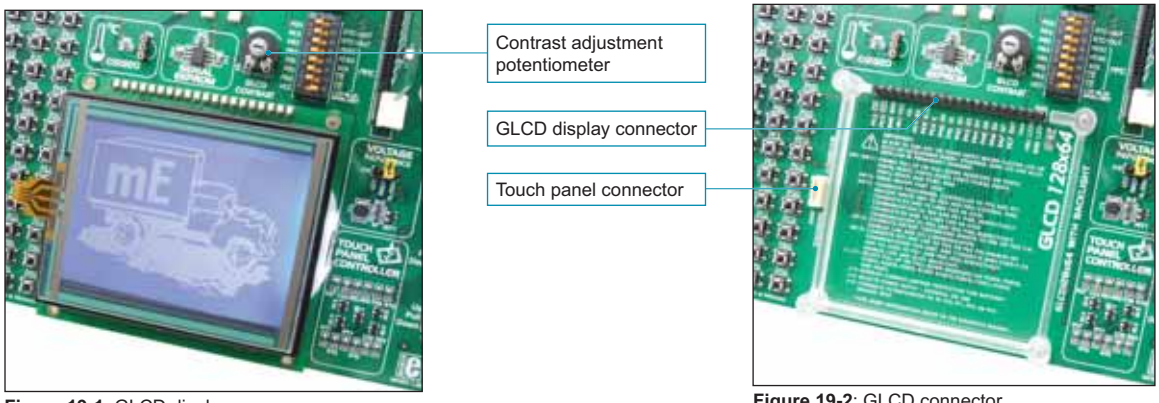


Figure 19-1: GLCD display

Figure 19-2: GLCD connector

GLCD display backlight is on

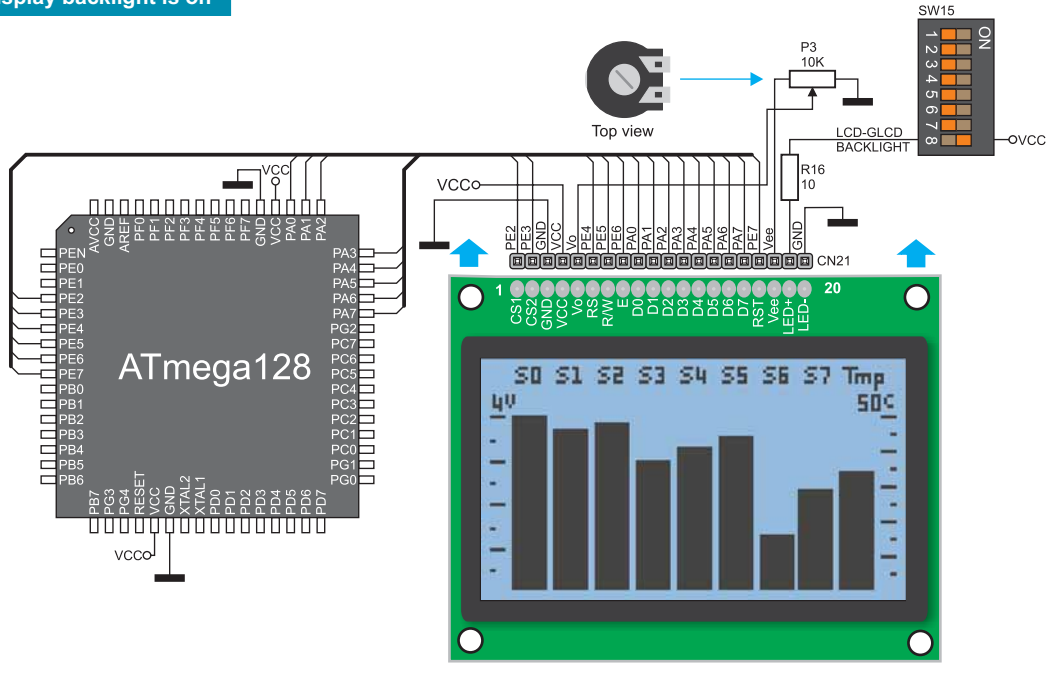


Figure 19-3: GLCD display connection schematic

page

20.0. Touch Panel

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

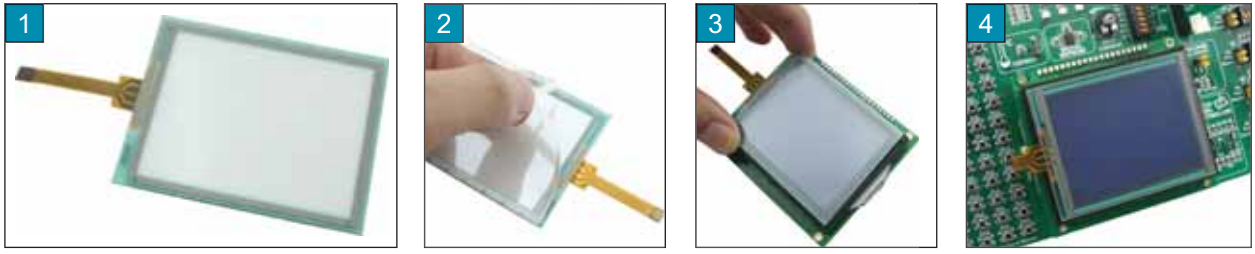


Figure 20-1: Placing touch panel over a GLCD display

Figure 20-1 shows how to place a touch panel over a GLCD display. Make sure that the flat cable is to the left of the GLCD display, as shown in Figure 4.

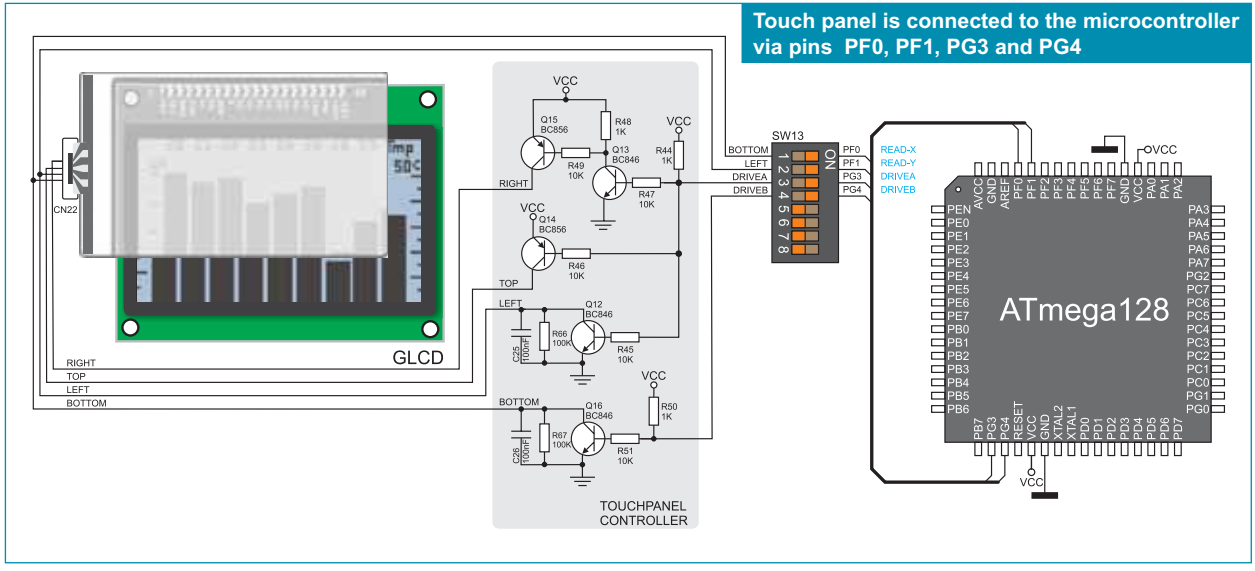


Figure 20-2: Touch panel connection schematic

Figure 20-3: Connecting touch panel

Figure 20-3 shows in detail how to connect a touch panel to the microcontroller. Bring the end of the flat cable close to the CN22 connector (Figure 1). Plug the cable into the connector (Figure 2) and press it easily so as to fully fit the connector (Figure 3). Now, a GLCD display can be plugged into the appropriate connector (Figure 4).

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

21.0. Input/Output Ports

Along the right side of the development system, there are eleven 10-pin connectors which are connected to the microcontroller's I/O ports. Pins PB1, PB2, PB3, PE0 and PE1 are used for programming and therefore are not directly connected to the appropriate 10-pin connectors, but via the programmer's multiplexer. DIP switches SW1-SW11 enable each connector pin to be connected to one pull-up/pull-down resistor. Whether pins of some port are to be connected to a pull-up or a pull-down resistor depends on the position of jumpers J1-J11.

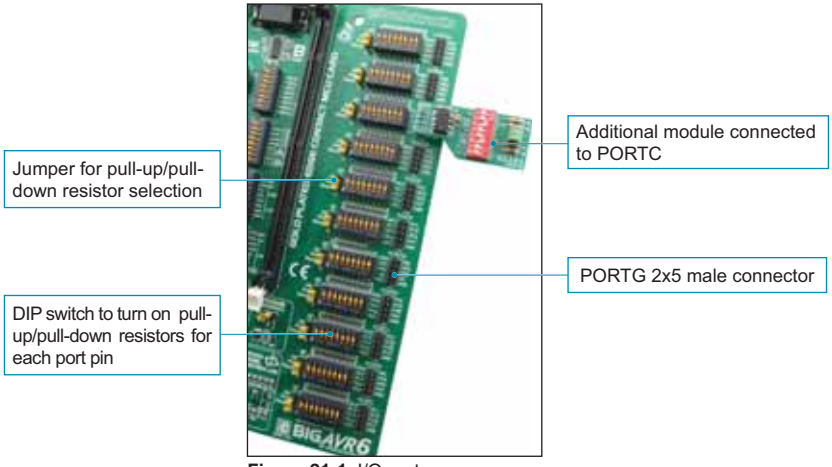


Figure 21-1: I/O ports

Figure 21-2: J2 in pull-down position

Figure 21-3: J2 in pull-up position

Port PORTA pins are connected to pull-down resistors

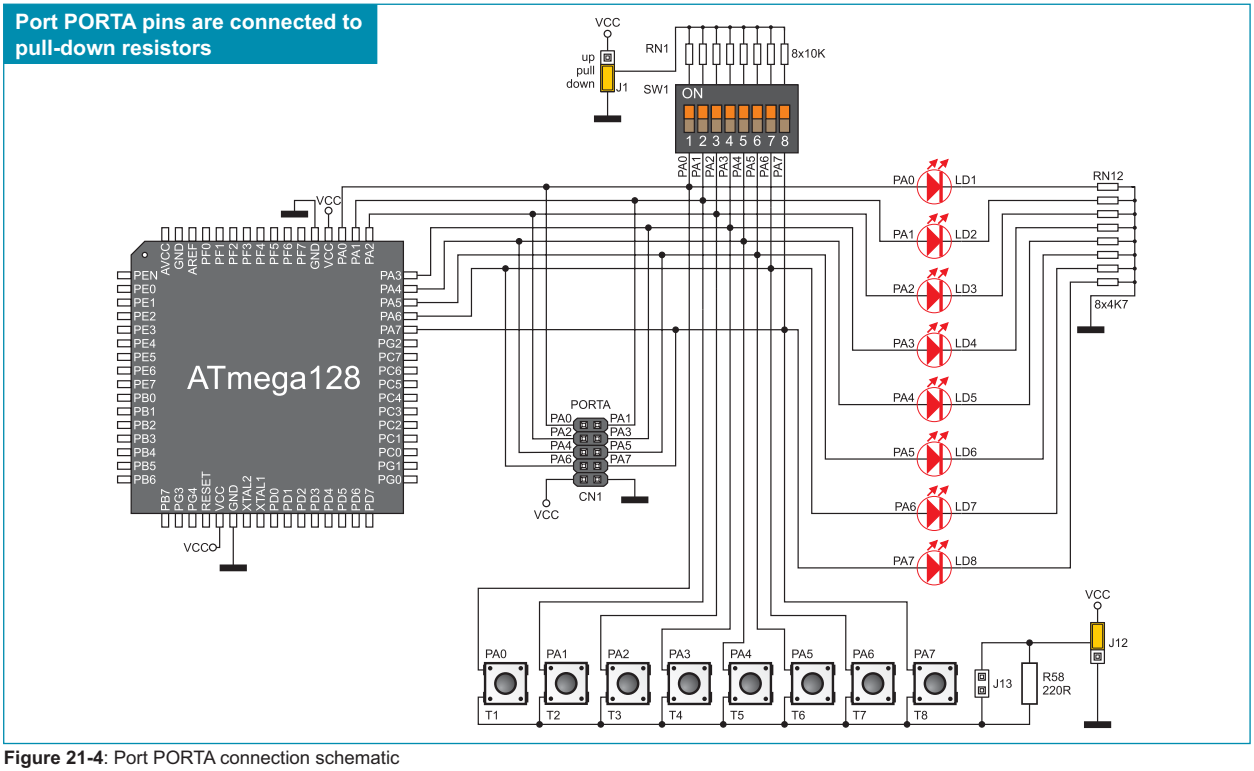


Figure 21-4: Port PORTA connection schematic

Pull-up/pull-down resistors enable you to set the logic level on all microcontroller's input pins when they are in idle state. Such level depends on the position of the pull-up/pull-down jumper. The PG2 pin with the relevant DIP switch SW7, jumper J7 and PG2 push button with jumper J12 are used here for the purpose of explaining the performance of pull-up/pull-down resistors. The principle of their operation is the same as for all other microcontroller pins.

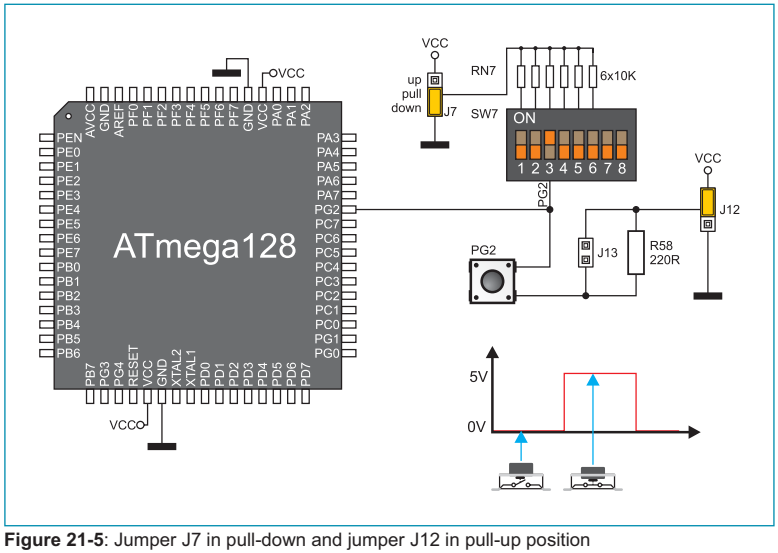


Figure 21-5: Jumper J7 in pull-down and jumper J12 in pull-up position

In order to enable the PORTG port pins to be connected to pull-down resistors, it is necessary to place jumper J7 in the *Down* position first. This enables any PORTG port pin to be supplied with a logic zero (0V) in idle state over jumper J7 and 6x10k resistor network. To provide the PG2 pin with such signal, it is necessary to set switch PG2 on the DIP switch SW7 to the ON position.

As a result, every time you press the PG2 push button, a logic one (VCC voltage) will appear on the PG2 pin, provided that jumper J12 is placed in the VCC position.

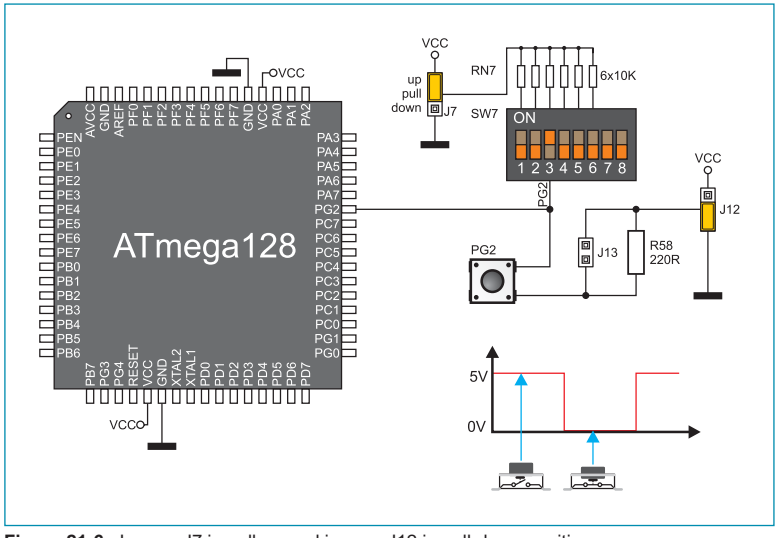


Figure 21-6: Jumper J7 in pull-up and jumper J12 in pull-down position

In order to enable port PORTG pins to be connected to pull-up resistors and the port input pins to be supplied with a logic zero (0), it is necessary to place jumper J7 in the *Up* position and jumper J12 in the GND position. This enables any port PORTG input pin to be driven high (5V) in idle state over the 10k resistor.

As a result, every time you press the PG2 push button, a logic zero (0V) will appear on the PG2 pin, provided that the PG2 switch is set to the ON position.

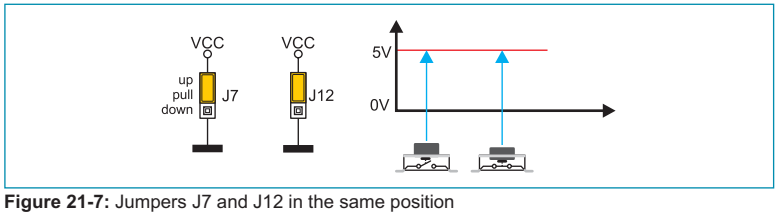


Figure 21-7: Jumpers J7 and J12 in the same position

In case that jumpers J7 and J12 have the same logic state, pressure on any button will not cause input pins to change their logic state.

DISCLAIMER

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

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

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

HIGH RISK ACTIVITIES

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

TRADEMARKS

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

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

©Mikroelektronika™, 2009, All Rights Reserved.

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

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

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