

AT Mini BOARD™

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 projects in a fast and efficient way.

Development System

 **MikroElektronika**

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

AT-Mini Board

The AT-Mini Board is a miniature development system that enables you to experiment with the ATMEGA328 microcontroller from Atmel®.

Key features:

- Bootloader program loaded into the ATMEGA328 microcontroller;
- USB-UART communication;
- Serial SPI communication;
- UART communication;
- 3.3 to 16V power supply.

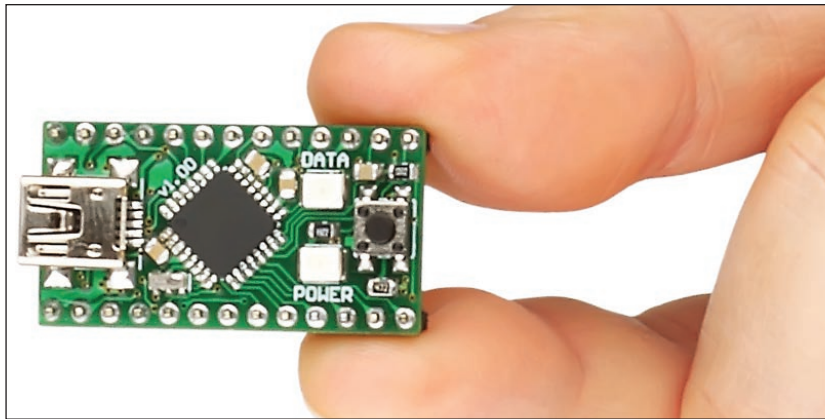


Figure 1: AT-Mini Board development system

Appliance:

The AT-Mini Board development system is used for experiments or can be built into some devices. Due to bootloader program which is preprogrammed, the programming of MCU does not require additional programmer. For connection with other devices the AT-Mini Board uses pads. Every pad is clearly marked so the access to individual pins on MCU is simple.

Power supply:

Power supply can be attached to the development system via a USB connector, VCC-IN or VCC pad. Regardless of the way in which the power supply is attached LED marked with POWER indicates that power supply is connected. Power supply voltage depends on the pad it is attached to.

- VCC-IN pad can be attached to +3.3 to 16V DC power supply source for 3.3V 8MHz board and from 5V to 16V for 5V 16MHz board.
- VCC pad can be attached to +5V DC for 5V 16MHz board
- VCC pad can be attached to +3.3V DC for 3.3V 8MHz board

When using VCC-IN power supply, the development system uses a built-in voltage regulator to step down VCC-IN voltage to 3.3 or 5V. Regulated voltage can be used via the VCC pad for supplying external devices (3.3V for 3.3V 8MHz board or 5V for 5V 16MHz board).

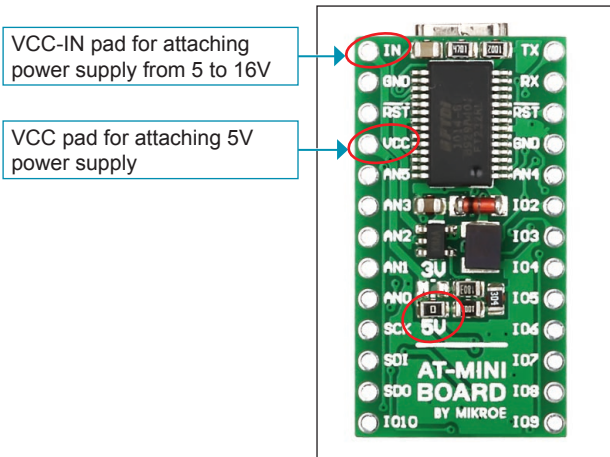


Figure 2: 5V 16MHz board

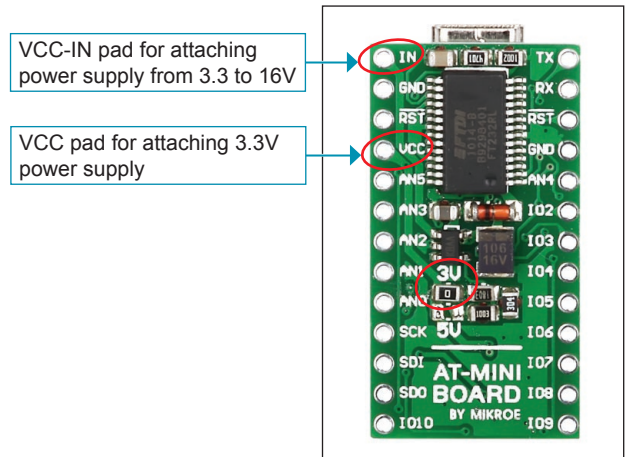


Figure 3: 3.3V 8MHz board

Pinout:

- IN: Power supply input for 3.3 to 16V DC;
- GND: Ground;
- VCC: Power supply 3.3 or 5V input/output;
- RST: External reset;
- AN1 to AN5: Analog input pins for A/D conversion;
- SCK: Master Clock output;
- SDI: Master Data input, Slave Data output;
- SDO: Master Data output, Slave Data input;
- IO2 to IO10: bidirectional I/O pins
- RX: UART Receive data
- TX: UART Transmit data

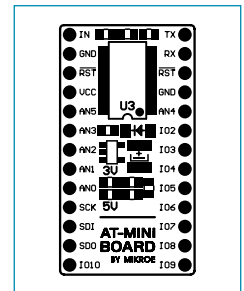


Figure 4: Pinout

RST, SCK, SDI, SDO, RX and TX pins can also be used as bidirectional I/O pins.

LED marked with DATA is connected to the SCK pin and can be used for signaling purposes. The development system also has RESET button which is connected to the RST pin. In addition, the development system is shipped with two 1x13 male pin headers which can be soldered on pads, Figure 5.

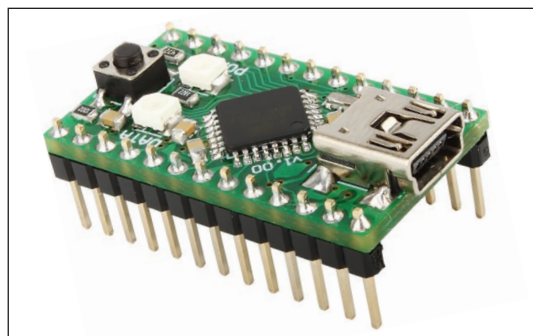


Figure 5: AT-Mini Board with headers

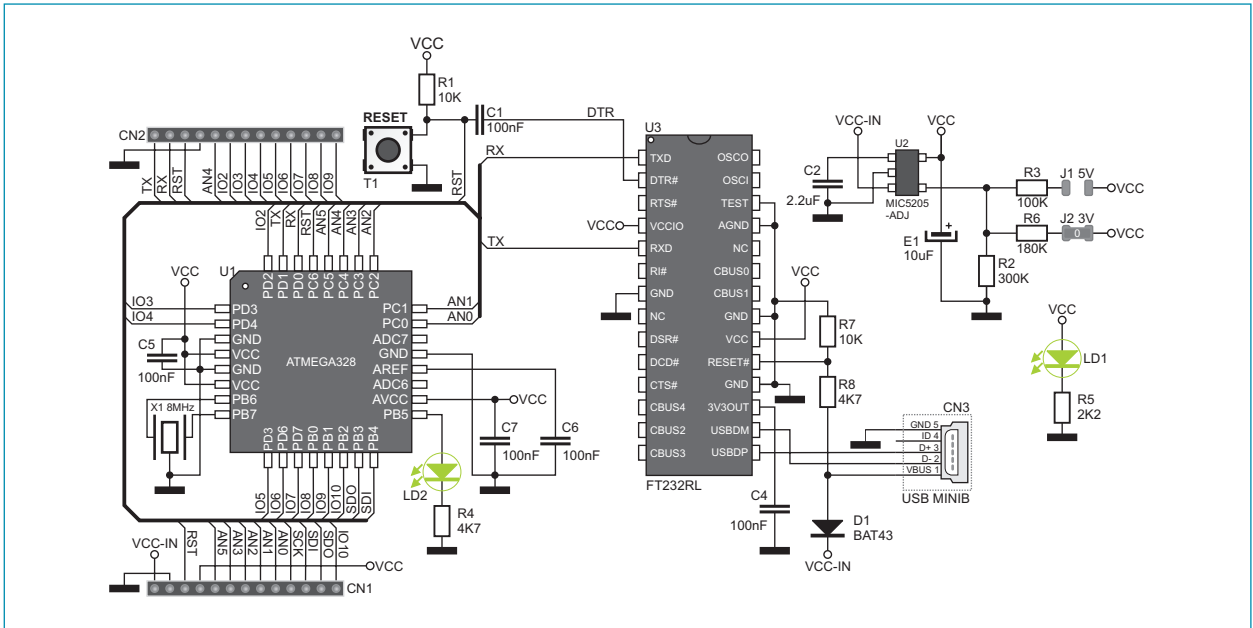


Figure 6: 3.3V 8MHz board connection schematic

NOTE: Connection schematic for the 5V 16MHz board is the same as for the 3.3V 8MHz board with exception of jumpers J1 and J2 and oscillator X1. For the 5V 16 MHz board jumper J1 is soldered (while J2 is removed) and oscillator X1 has a value of 16MHz.

Programming MCU via bootloader:

Bootloader is a program which enables the microcontroller to be programmed without external programmer. Everything you need is to connect the development system to a PC via a USB cable and to install one of MikroElektronika's compilers for AVR microcontrollers such as mikroC PRO for AVR, mikroBASIC PRO for AVR or mikroPASCAL PRO for AVR. In this example we will use the mikroC PRO for AVR compiler.

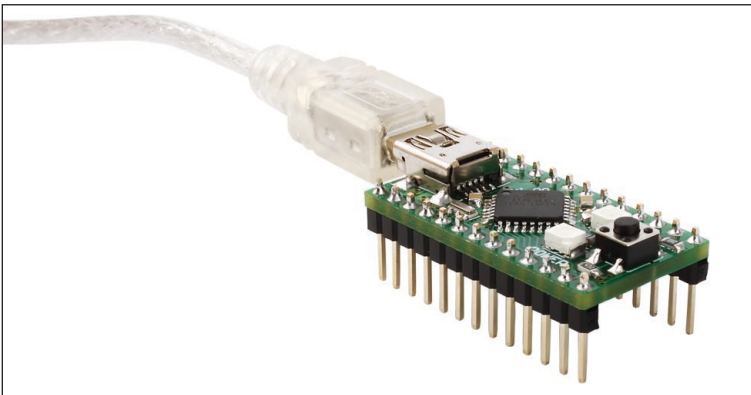


Figure 7: AT-Mini Board connected to a PC via a USB cable

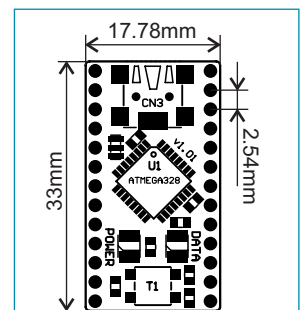


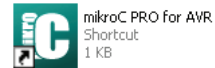
Figure 11: Board dimensions

STEP 1: Start compiler

From MikroElektronika's website download mikroC PRO for AVR:

<http://www.mikroe.com/eng/products/view/228/mikroc-pro-for-avr/>

After the compiler installation is completed click on the mikroC PRO for AVR icon

The screenshot shows the mikroC PRO for AVR IDE window. The title bar reads 'mikroC PRO for AVR v.4.60.0.0 - C:\AT Mini\LedBlinking.mc.pov'. The menu bar includes 'File', 'Edit', 'View', 'Project', 'Build', 'Run', 'Tools', and 'Help'. The toolbar contains various icons for file operations and compilation. The main window displays the source code for 'LedBlinking.c'. The code includes project metadata, configuration details for an ATmega16 microcontroller, and a C program that blinks LEDs on PORTB, PORTC, and PORTD. The code is as follows:

```
1 /*
 * Project name:
 * LED_Blinking (Simple 'Hello World' project)
 * Copyright:
 * (c) Mikroelektronika, 2010.
 * Revision History:
 * 20080930:
 * - initial release.
 * Description:
10 This is a simple 'Hello World' project. It turns on/off diodes connected to
 * PORTA, PORTB, PORTC and PORTD.
 * Test configuration:
 * MCU: ATmega16
 * http://www.stmel.com/dyn/resources/prod_documents/doc2466.pdf
 * Dev. Board: EasyAVR6 - ac:LEDs
 * http://www.mikroe.com/eng/products/view/321/easyavr6-development-system/
 * Oscillator: External Clock 08.0000 MHz
 * Ext. Modules: -
 * SW: mikroC PRO for AVR
 * http://www.mikroe.com/eng/products/view/228/mikroc-pro-for-avr/
20 * NOTES:
 * - Make sure you turn ON the PORTA, PORTB, PORTC and PORTD LEDs (SW8.1, SW8.2, SW8.3 and SW8.4).
 */
void main() {
 * DDRB = 0xFF; // Set direction to be output
 * DDRC = 0xFF; // Set direction to be output
 * DDRD = 0xFF; // Set direction to be output
30 do {
 * PORTB = 0x00; // Turn OFF diodes on PORTB
 * PORTC = 0x00; // Turn OFF diodes on PORTC
 * PORTD = 0x00; // Turn OFF diodes on PORTD
 * Delay_ms(1000); // 1 second delay
 * PORTB = 0xFF; // Turn ON diodes on PORTB
 * PORTC = 0xFF; // Turn ON diodes on PORTC
 * PORTD = 0xFF; // Turn ON diodes on PORTD
 * Delay_ms(1000); // 1 second delay
40 } while(1); // Endless loop
}
```

Figure 8: mikroC PRO for AVR window




In the compiler window write a source code for your application. In this example we are using source code for LedBlinking which will make DATA LED on the development system blink.

STEP 2: Download additional files

From the following link: http://www.mikroe.com/eng/downloads/get/1591/atmini_bootloader_v100.zip download .zip file which contains data for tool settings in mikroC PRO for AVR.

Unpack .zip file  atmini_bootloader_v101 542 KB →  atmini_bootloader_v101 →  AT-MINI bootloader →  Software

Create a new folder on "C:\\" disk and name it "avrdude". Copy avrdude, avrdude.conf and ReadMe files into this folder (C:\avrdude).

Name	Size	Type
 avrdude	1,878 KB	Application
 avrdude.conf	408 KB	CONF File
 ReadMe	1 KB	Text Document

STEP 3: Add AT Mini tool

In compiler window select **Options** from the **Tools** menu or press F12 on keyboard.

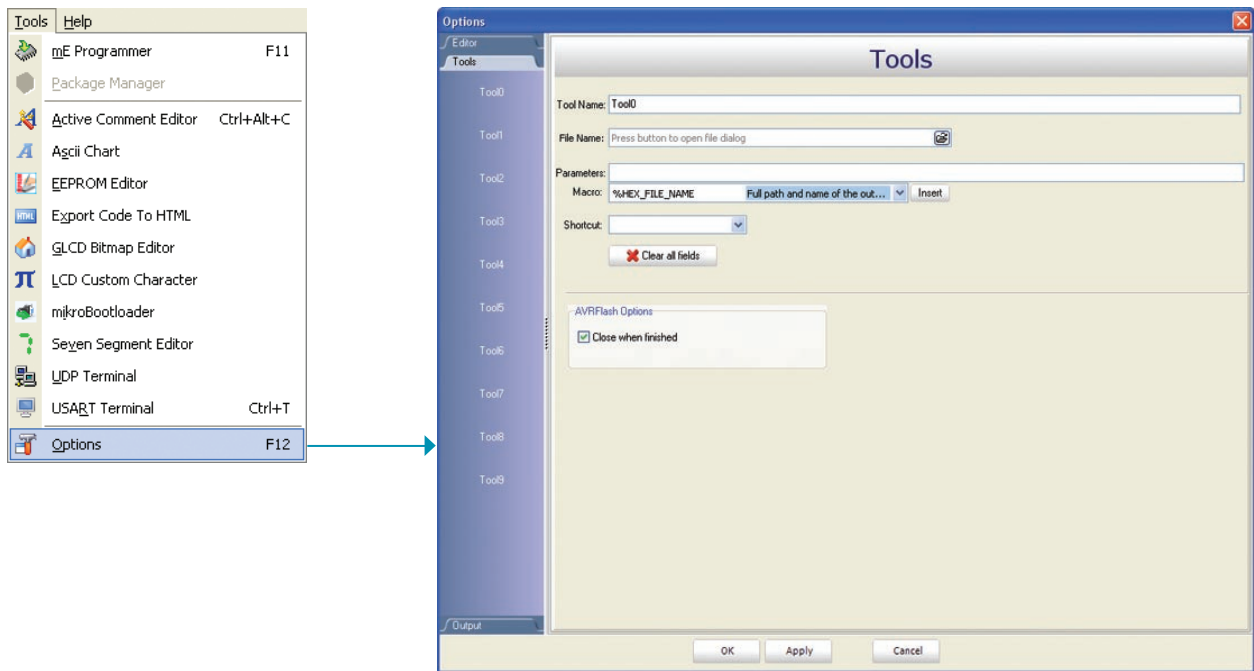
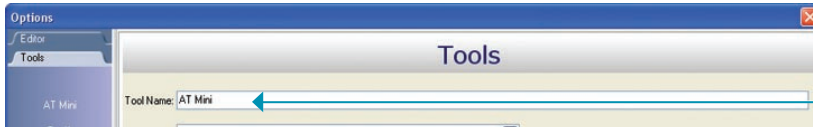


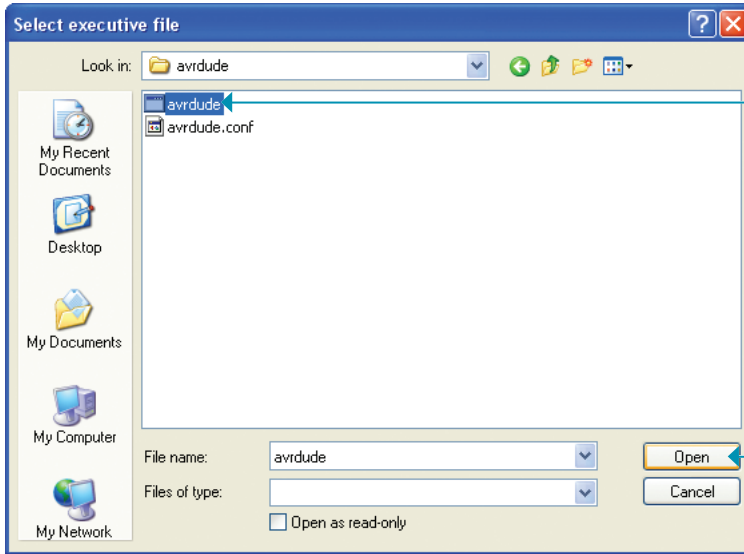
Figure 9: Options window



Rename Tool0 to AT Mini



Click on the open button

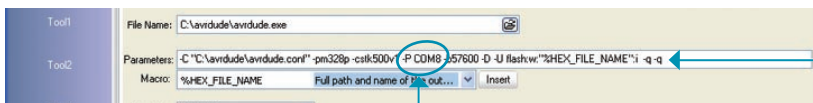


Find avrduide.exe in C:\avrduide folder

Click Open



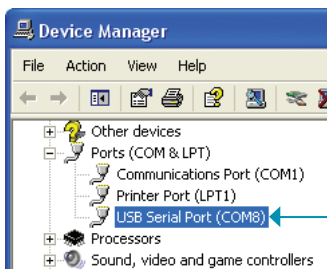
A path to avrduide.exe file will appear in the File Name box



In parameters text box type in command line path: `-C "C:\avrduide\avrduide.conf" -pm328p -cstk500v1 -P COM8 -b57600 -D -U flash:w:"%HEX_FILE_NAME";i -q -q`

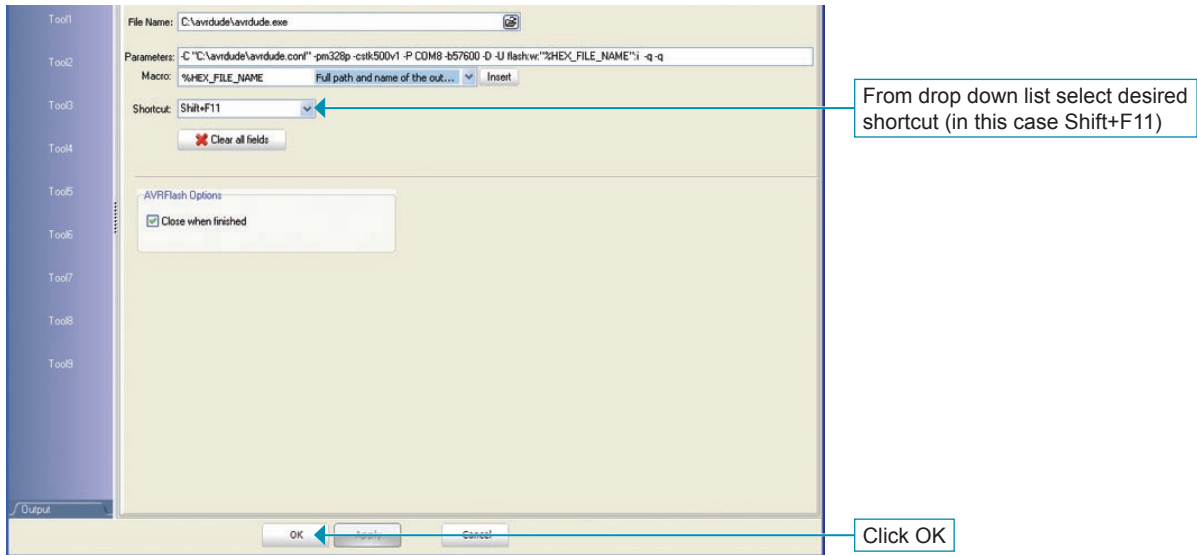
-P COM8 represents USB port on your PC which is connected to development system. In the Device Manager window find correct COM port and type it instead of **COM8**.

NOTE: If you copy command line path retype every quotation mark



In Device Manager on your PC look for which COM port the development system is attached to

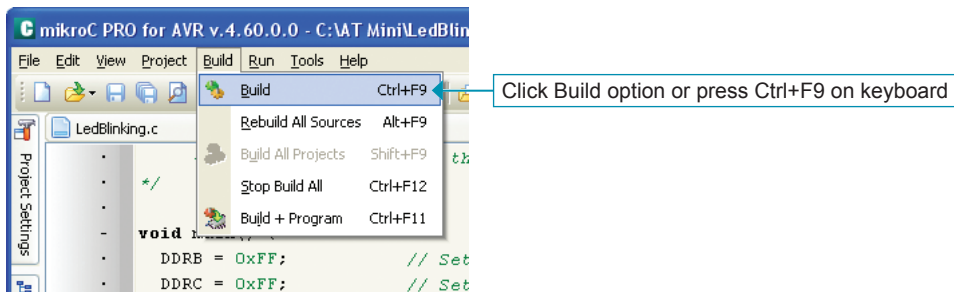
NOTE: AT Mini Board can establish communication with a PC only via a single digit COM port. If your PC automatically assigns double digit COM port change it to single digit COM port (explained on page 12).



Now AT Mini tool is created and you can proceed with MCU programming.

STEP 4: Compile source code

After the source code is written, select the Build option from the Build menu to compile it.



When the source code is compiled the .hex file is generated and ready to be uploaded into the microcontroller.

STEP 5: Upload .hex file into MCU

For .hex file uploading we will use AT Mini tool that was created in step 3. Just press Shift+F11 (or chosen shortcut) and .hex file will be transferred into the microcontroller, Figure 10.

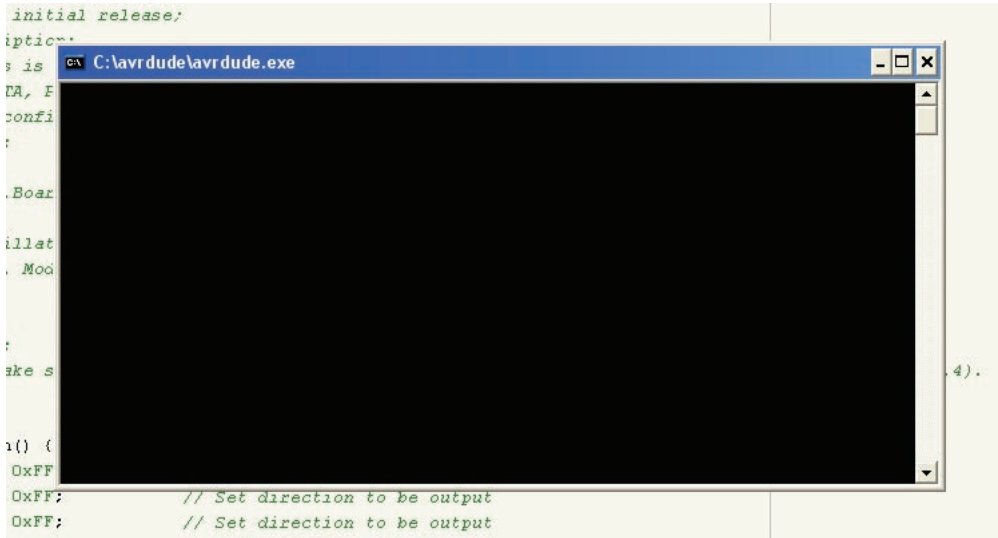


Figure 10: Uploading .hex file

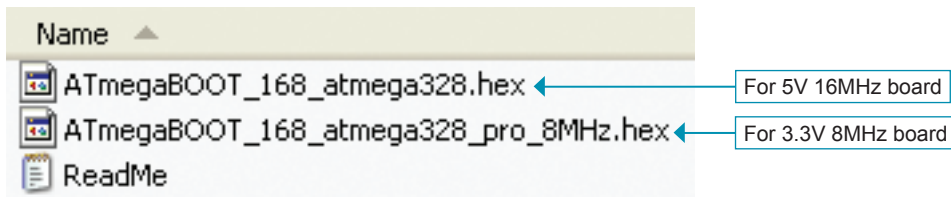
After uploading is over, a black window (figure 10) will disappear which means that the .hex file is uploaded and DATA LED on the development system will start to blink.

NOTE: If the black window just flickers on your monitor and the .hex file hasn't been uploaded into the microcontroller it is possible that you need to install libusb0.dll file on your PC. This file can be found on the internet, for example at: <http://www.dll-files.com/dllindex/dll-files.shtml?libusb0>

Firmware

If you accidentally overwrite the bootloader program it is possible to load it again. In the Firmware folder you can find bootloader .hex files which can be loaded into the microcontroller via the AVR ISP programmer.

DO NOT try to upload these files via bootloader program which is preinstalled into the microcontroller because you can damage the bootloader program.

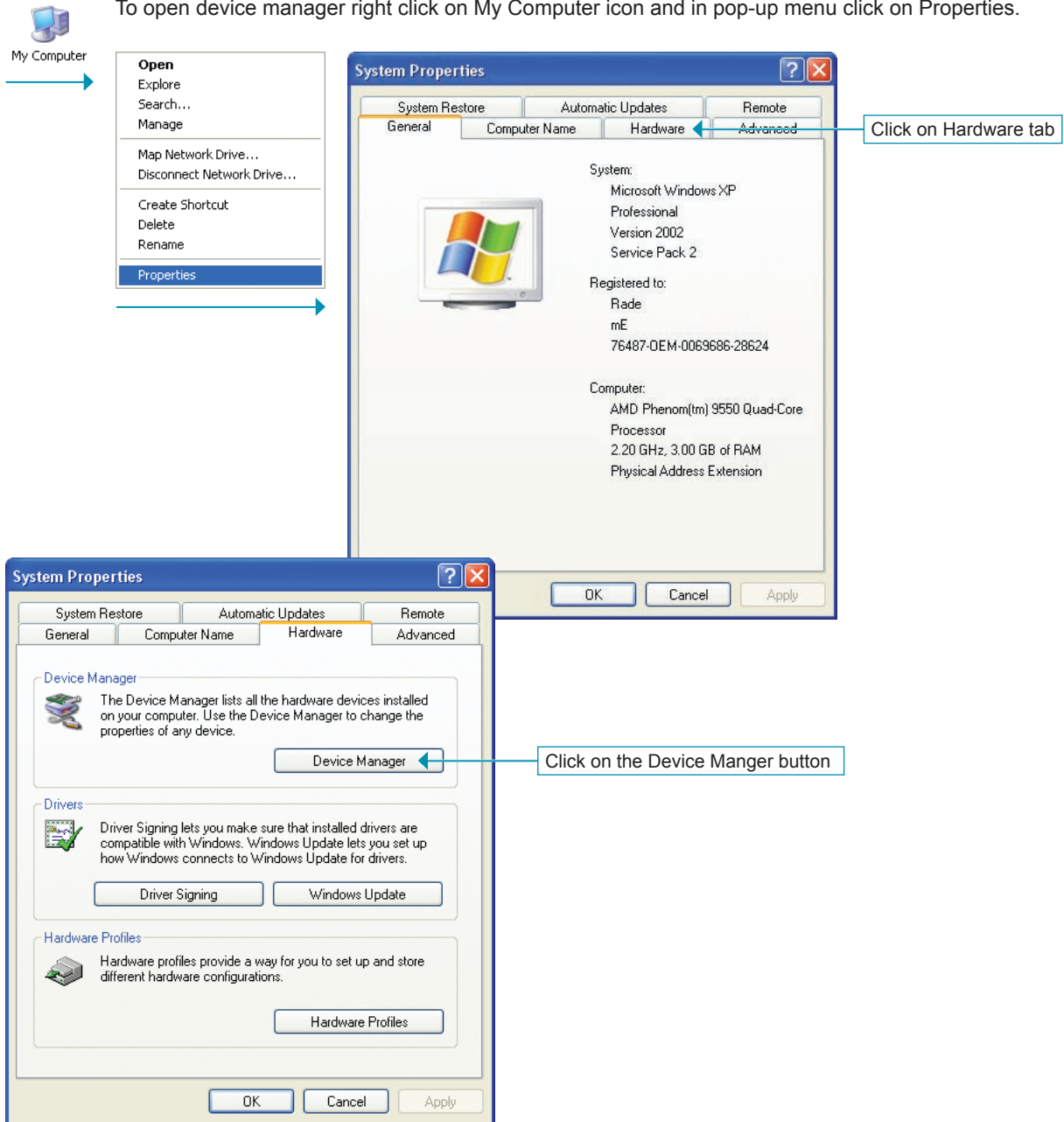


Changing COM port

To change COM port on you PC follow the steps below:

STEP 1: Open Device Manager

To open device manager right click on My Computer icon and in pop-up menu click on Properties.



STEP 2: Changing COM port

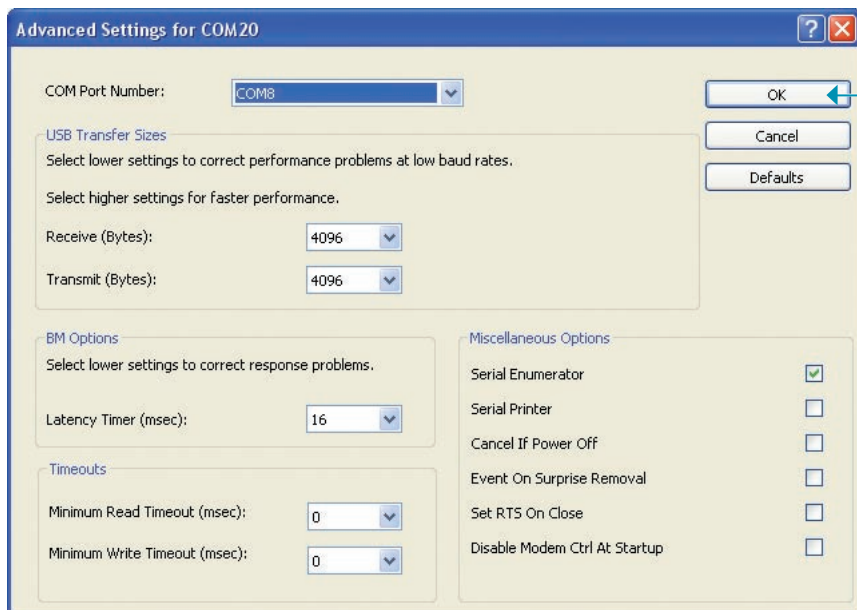
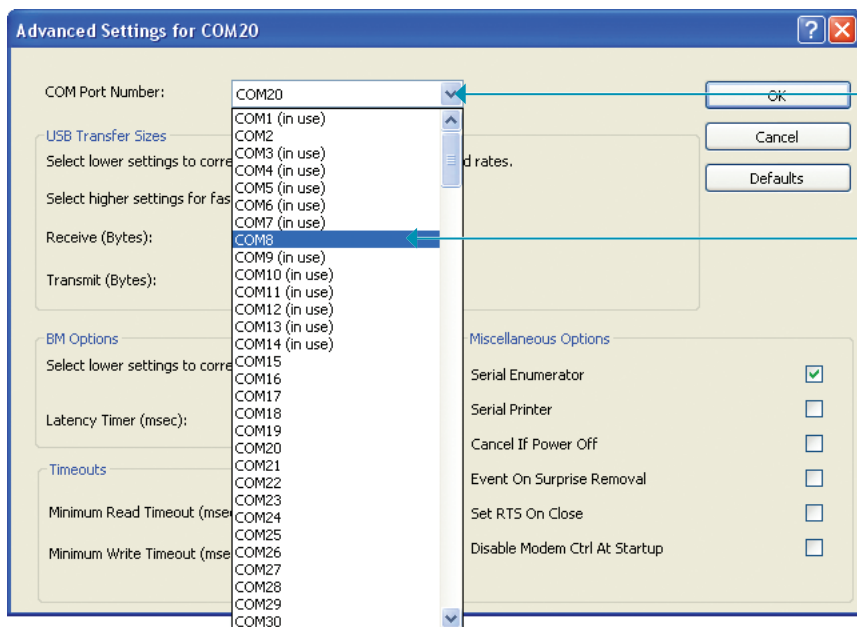
The image shows two screenshots from a Windows operating system illustrating the steps to change the COM port for the AT Mini Board.

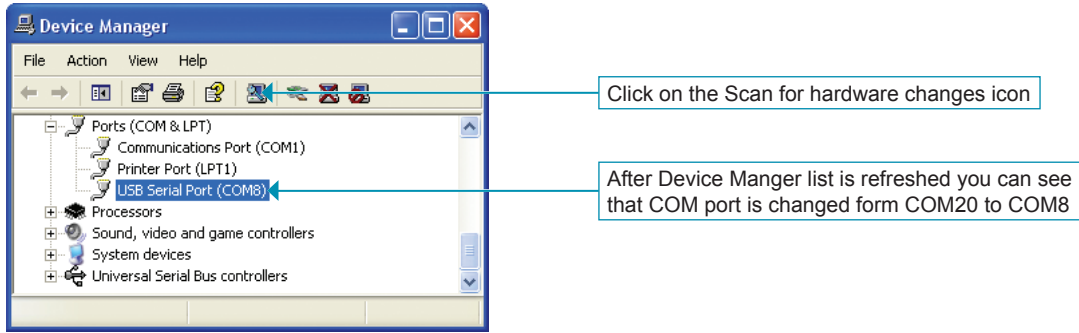
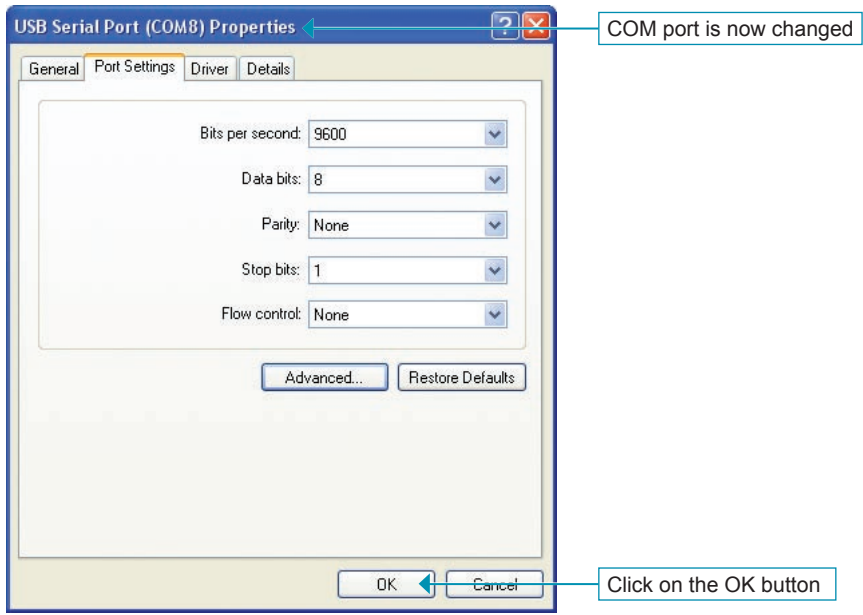
Top Screenshot: Device Manager

- The **Ports (COM & LPT)** category is expanded, showing **Communications Port (COM1)**, **Printer Port (LPT1)**, and **USB Serial Port (COM20)**.
- A right-click context menu is open over **USB Serial Port (COM20)**. The **Properties** option is highlighted.
- Callout boxes indicate: "From Device Manager select COM port which is used for communication with AT Mini Board (in this case COM20)" pointing to the selected port, and "Click on Properties" pointing to the highlighted menu item.

Bottom Screenshot: USB Serial Port (COM20) Properties

- The **Port Settings** tab is selected.
- The configuration is set to: **Bits per second: 9600**, **Data bits: 8**, **Parity: None**, **Stop bits: 1**, and **Flow control: None**.
- The **Advanced...** button is highlighted.
- Callout boxes indicate: "Click on Port Settings" pointing to the selected tab, and "Click on Advanced" pointing to the highlighted button.







MikroElektronika

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

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

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

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