Supplement for module D061 incl. ATMega128 Prozessor

V 1.3
16. March 2006

Congratulations for buying this ATMega128-module.

This module contains the electronics for the color display as well as PowerBooster Technology, a RS232 interface and a ATMega128 processor. This module allows you to keep the complete electronics of your solutions on one board with minimal space.

You need to complete this module by clicking the display into the connector and by soldering your needed cables.

When the module is inevitably frequently moved, this load could damage the patch cord of the LCD in the long run. Therefore we suggest that you fix the display at the lower edge with a drop adhesive or a piece to double-sided tape on the plate.
The display can be removed however only then surely again, if you stick it only at the edge of the plastic. If you are sticking it however on the back, then the back foil of the display is possibly damaged when taking the display off.
Note: the glass of the display or the electronics below may become damaged by a too strong pressing at the display surface.

This manual shows you just the connection plan of this board and gives you some hints for using it. For the programming information of the color display please check the separate manual for the color display.

CAUTION:
1) Never attach the display or remove it, as long as power is switched on
2) Connect the display to its connector always correctly (see illustrations on next pages). Never attach differently around! If you wrongly attach the display, it is inevitably destroyed.

Port usage of this module

To use this module with your own software you need to know which port of the processor is connected to which display connector. **When using our sample software you may need to change the port selection at the beginning as follows:**

Display connection	Port at integrated micro controller	Sample programs in Bascom should show at the beginning:

RS	B.6	`Const Rs = 6`
		`Const CS = 5`
		`Const Sdata = 2`
		`Const Sclk = 1`

SD	B.2	
SC	B.1	
CS	B.5	

Delivery:

What you get delivered:
1 x PCB with micro processor ATMega 128 etc.
1 x color display
6 x switches
3 x pins (each 2x5, 0,1" spacing), 1 connector for ISP programming
software, documentation

Voltage supply

This board is delivered with two voltage regulators to provide you an easy usage and to avoid any damage to the processor and the display.
You may run this module with any DC voltage from 5 up to 20 Volt. The processor and the RS232 part is running with 5 Volt, the display electronics with 3 Volt. These voltage regulators are very-low-drop-regulators so you really can offer 5,0 Volt as a minimum to the board (and not approx. 6,5 V as a minimum as you would need with a usual 7805 regulator).

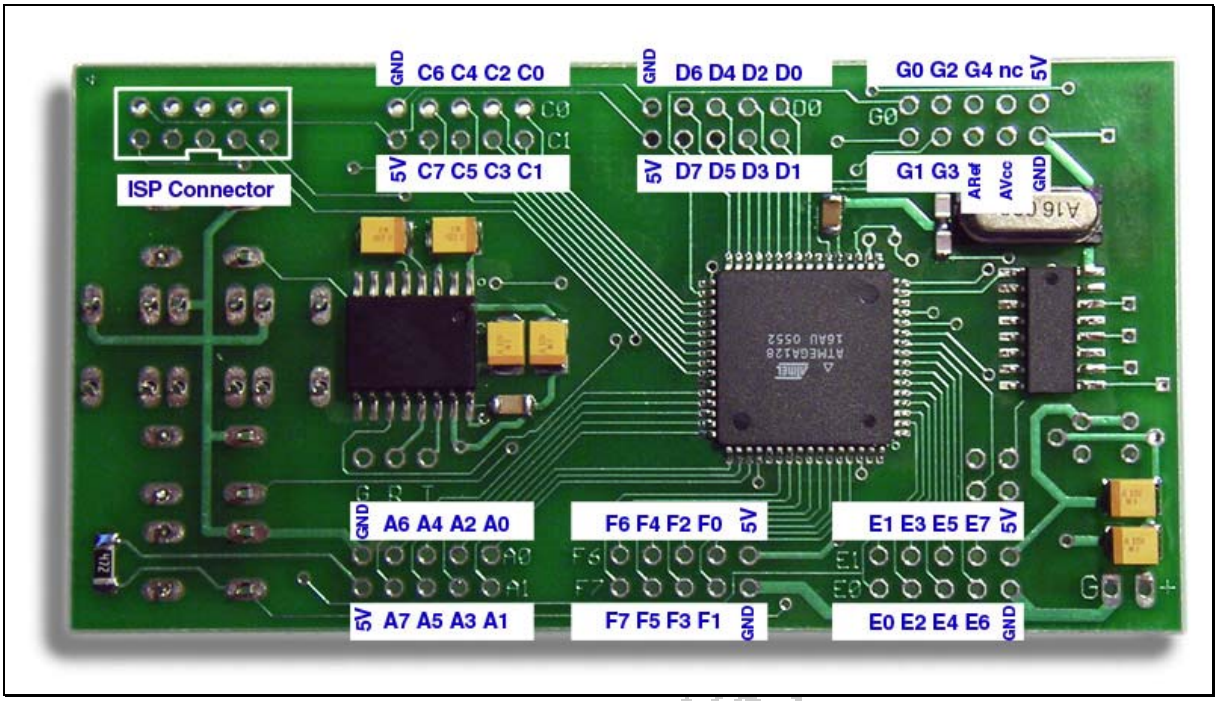
Caution:
The 5 Volt regulator is able to deliver up to 160mA current – the board itself needs about 50 mA as the 8 Volt for the display lighting is being produced at the board too. Please do not draw too much current from this board for any other items you are planning to connect to this board, you just have approx. 100mA left.
If you need more current, it would be a good idea to bridge the internal 5 Volt regulator (the one directly beneath the Vcc pad) and to use an external regulator which may provide larger current.
But then never provide a larger voltage to the board than 5,0 Volt. A higher voltage may destroy the display lighting and/or the processor on the board.

How to get it up and running

Connect the display and the switches (just click in, no soldering needed) and provide a voltage of 5 up to 20 Volts to the pads Vcc and GND. We already programmed the processor so you will see a program showing you the status of the switches and each single ports. The switch at the left side is the reset button.
Comment: The ports F4 to F7 show „0" when they are prepared for JTAG and are not usable otherwise until JTAG is disabled (see last page on details).

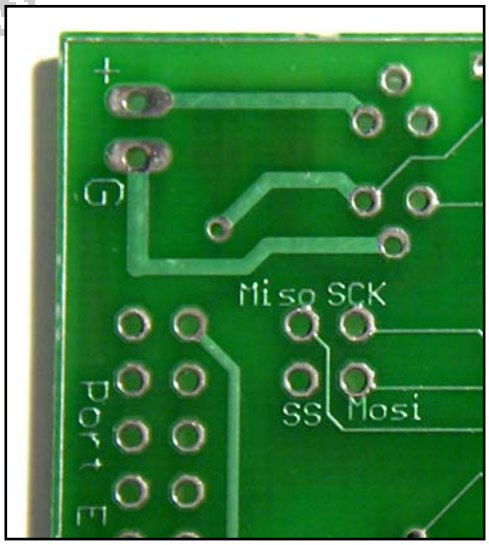
You need a separate ISP-programmer to be able to reprogram this board. This programmer is connected to the D061 board (to the black ISP-connector) and to your PC.

The pads of the module



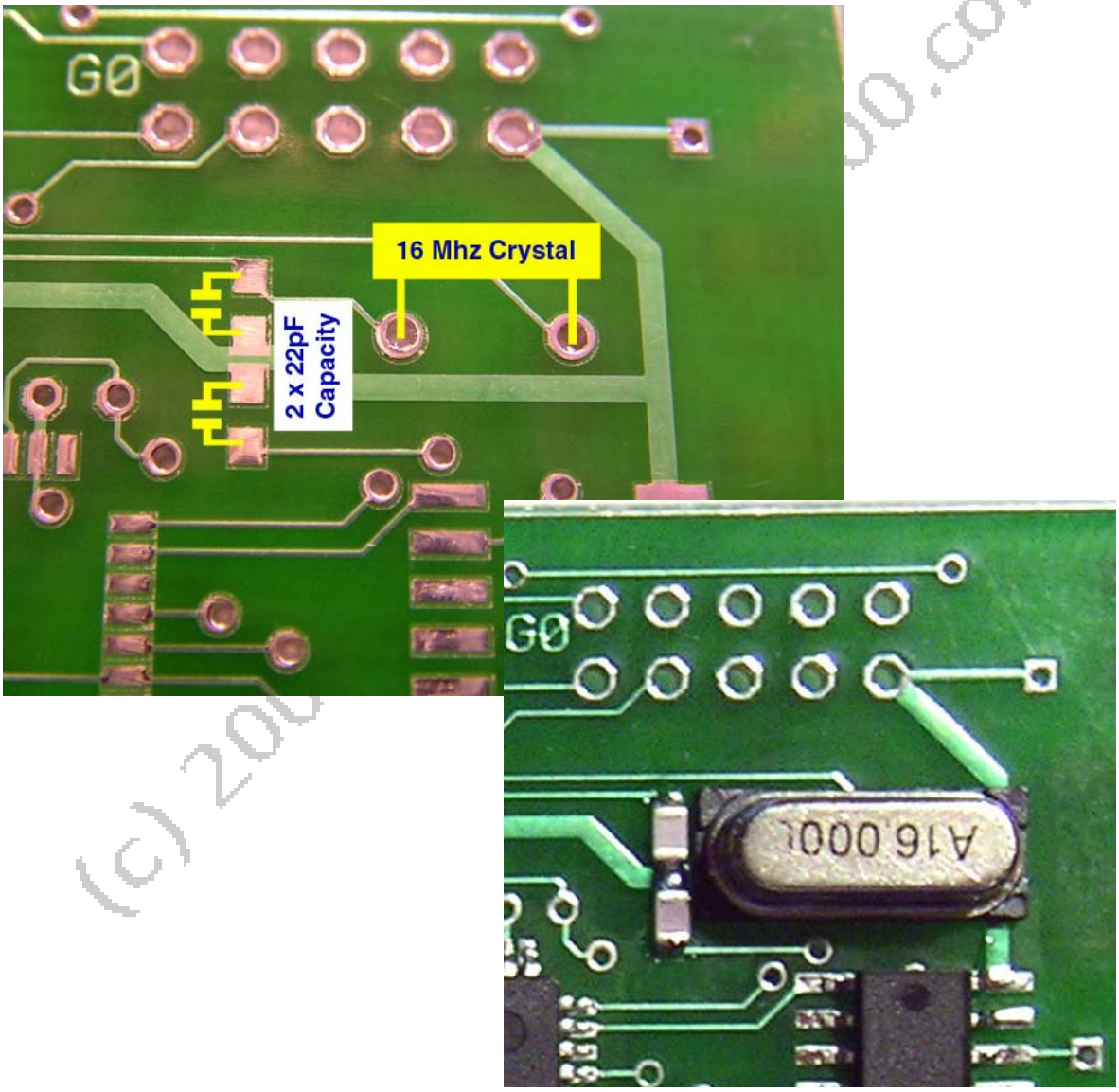
The photo above shows you the pads of the board. To make life easier to connect any additional modules we provide 5V-Vcc and GND at each Port-Connector.

A closer look shows that the port B is not available as the display electronics is connected to this port. The Ports B.0 to B.3 (SS, SCK, MOSI, MISO) are available through 4 extra connector pads which location are shown in the following picture.



Speed
The processor actually runs at 8 MHz (if you didn't order the 16 MHz option). You may double the speed by soldering a 16 MHz crystal and 2 x 22pF ceramic capacities to the board. You then need to reprogram the speed fuse from 0100 (8Mhz) to 1111 (external crystal 16 MHz).
Caution: any other selection than 0001, 0100 and 1111 **may cause a malfunction** of your board!! It will also not work anymore if you select 1111 without having a crystal soldered in. **Fuses are very delicate** – if you are not experienced with them, **do not experiment with changes – quickly the board will not function anymore!!**

The position of the crystal and the two capacities is shown at the following pictures – you see the empty PCB and also a PCB with the mounted devices.

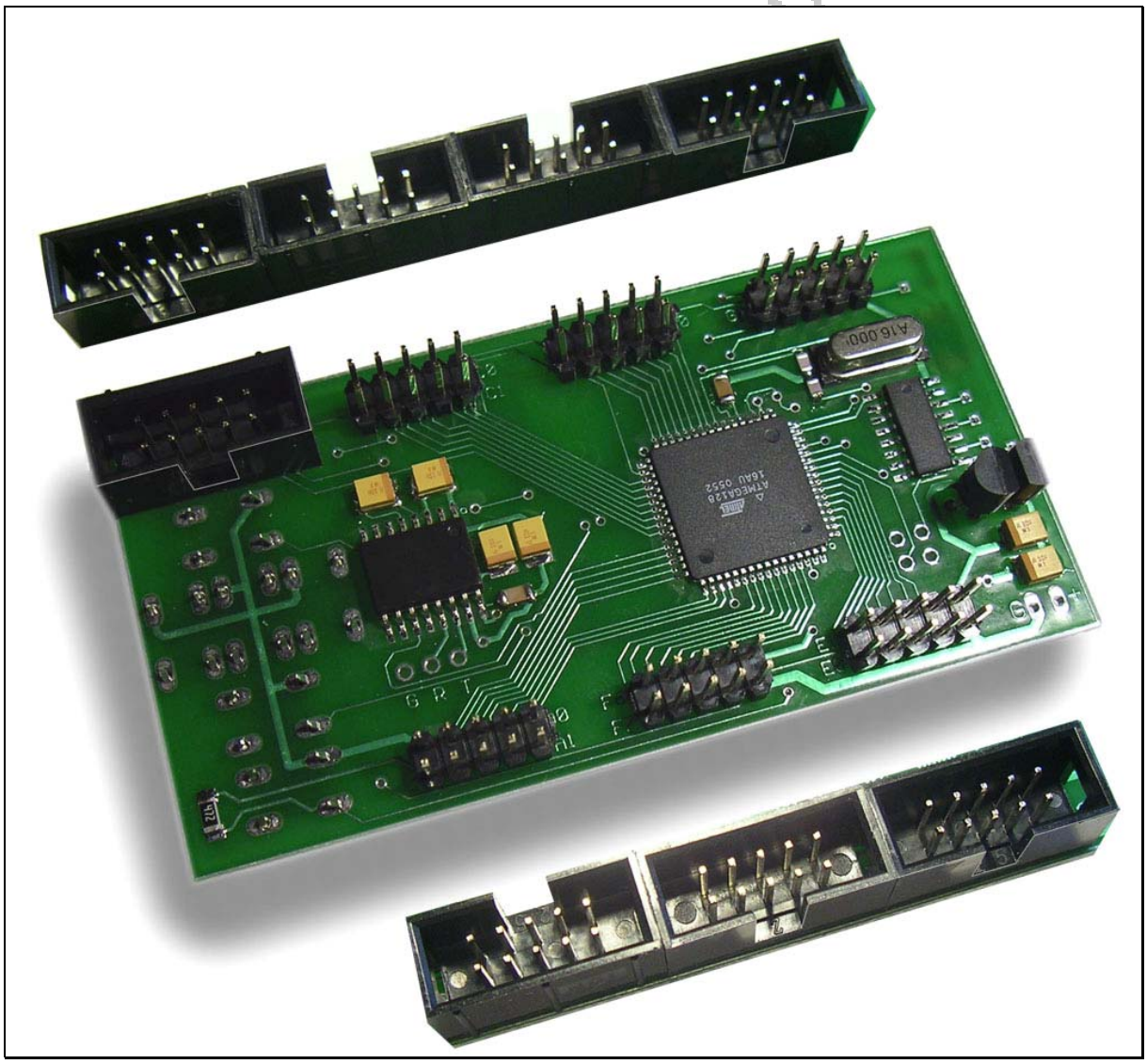


The connection pins

We did not solder the delivered pins, as you might want to use a different kind of connection. Also the switches are not connected yet of the same reason. You may just push them into place now and they will work (for testing purposes) without soldering as of the through plated holes.

We did provide the port pins with such a large space between the single ports to offer you the possibility to use standard connectors with the plastic frame around. But of course you can also use the regular pins with 0.1" spacing.
The following picture shows both.

If you solder the framed connectors as shown, then you always have GND at pin 9 and Vcc (+5 Volt) at pin 10 available – very useful if you need power at external devices which are connected to the controller board..

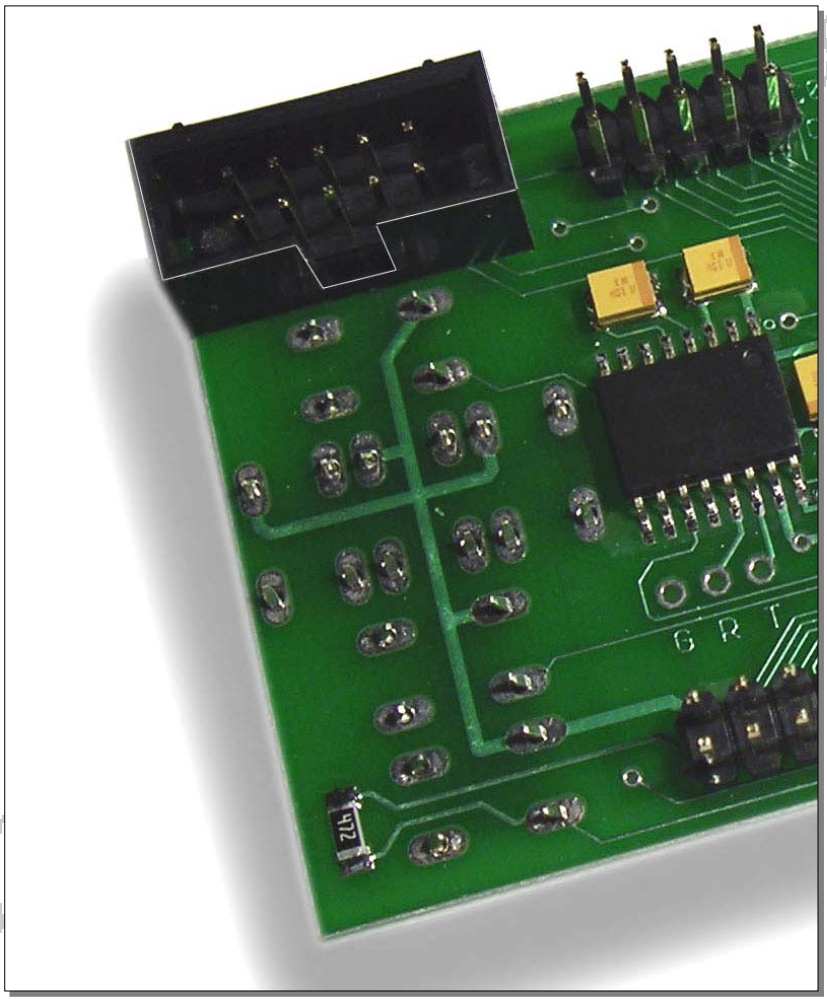


The ISP connector

Your modul is beeing programmed by using the ISP connector at the board.

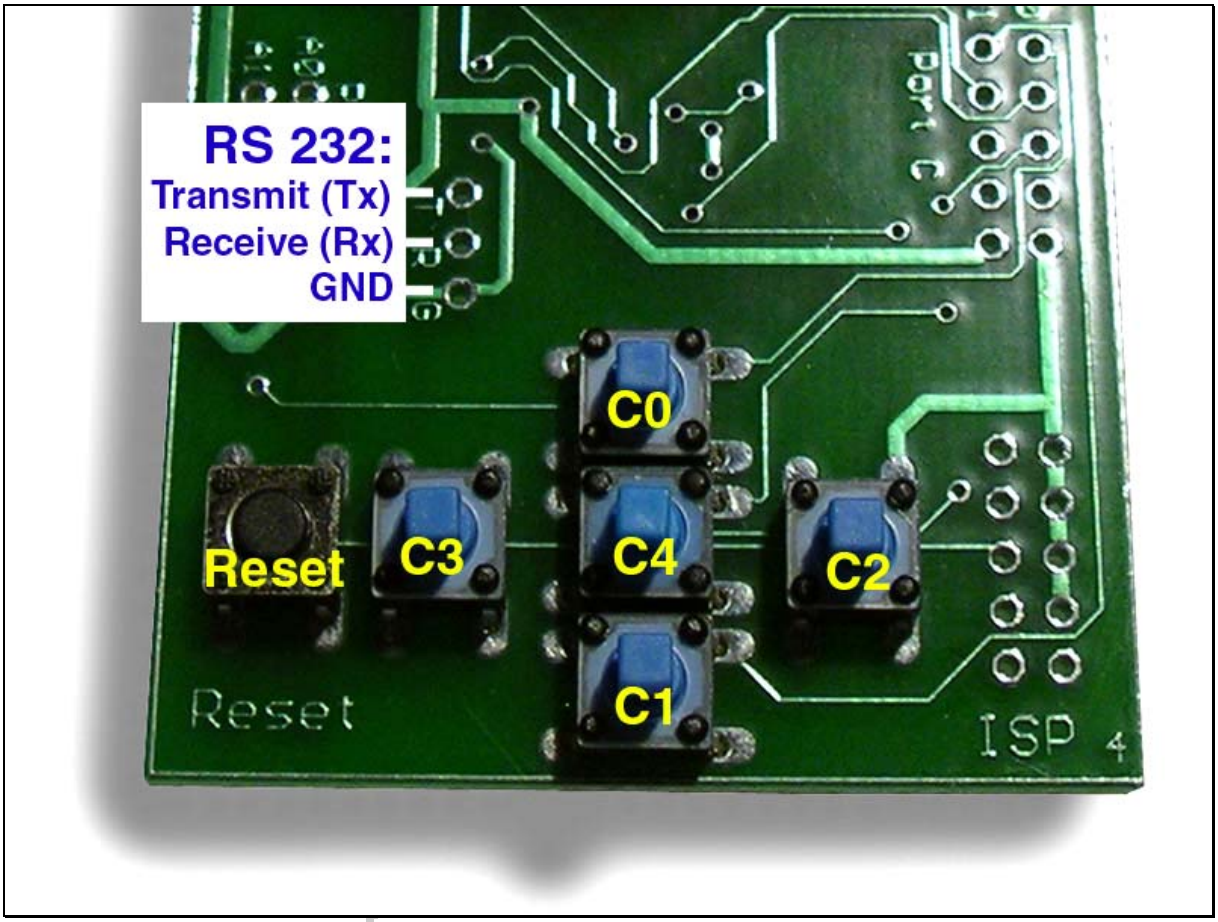
If you go and solder the shipped ISP connector please be aware of:

a) The connector has to be mounted at the lower side of the board (not the side with the display)
b) The „nose“ of the connector has to show to the direction of the switches.



The switches

The switches of the module are already connected with port C (see photo below) and will connect the port to GND when pressed. If you are going to use these switches, you should take care of, that these ports are only being used as input ports, not output ports as pressing a switch would then create a short-circuit.

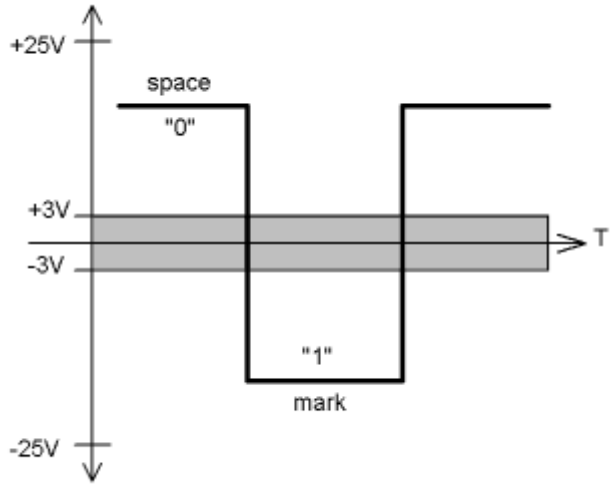


RS 232-usage

On the board there is a RS 232-connection available – please check the photo above to find the location. More information on RS-232 is given on the next pages.

RS232
RS-232 is a serial communication protocol. It sends information as bit after bit and has two signal levels:

- a voltage between -3 and -25 Volts is a logic one (1)
- a voltage between +3 and +25 Volts is a logic zero (0)



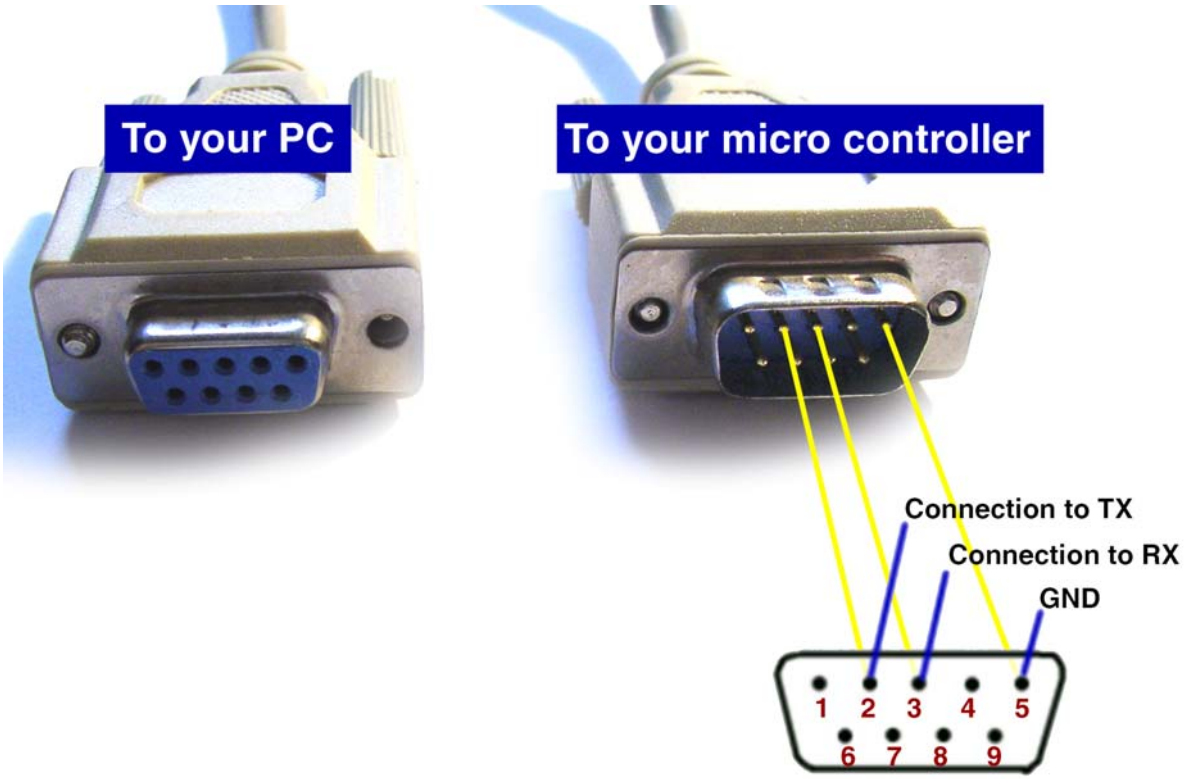
As the picture above shows, the voltage level between -3 and +3 Volts is undefined. In practice this is not so. Most often, any voltage level above 2.5 Volts is seen as a logic zero, anything below as a logic one.
The electrical specification of RS-232 is quite robust, all outputs must be able to sustain a full short-circuit and all inputs must have a schmitt-trigger action. This makes a full-standard RS232 port on a PC much less vulnerable than a TTL-level parallel port.
RS-232 is an a-synchronous protocol, meaning that no separate clock is transmitted with the data. Both sides must know the communication speed (we use the term baud-rate) beforehand.

RS-232 usually defines a complete hardware handshaking system using several wiring pins. We use only the most important three:

- RxD : receive data, pin number 2
- TxD : transmit data, pin number 3
- Ground, pin number 5

These pin numbers refer to a standard male DB9 connecter on your PC or laptop.

If you want to build a cable to connect our board to your PC you need a regular serial cable with one male and one female DB9-connector. The female will be connected to your PC, the male needs to become connected to you board. The following picture will help you to build a adapter and to connect the cable to your board.



Pin 2 is the receive channel of the PC –you need to connect this channel to the transmit channel (TX) of the micro controller. Also at Pin 3 the data of the PC are transmitted to our board and therefore you need to connect Pin 3 with the receive channel (RX) of the board.

The ATMega 128 offers two separate RS232 interfaces, we are only using interface 1 (the other one is interface 0 – see ATMega128 data sheet for this). At the ports D2 and D3 the used RS232 interface 1 of the ATmega is located. These two ports (D.2 and D.3) are connected to the RS232 interface chip at the board and this chip is connected to the Rx and Tx Pad. The chip decouples the high voltage RS232 signals from your PC. If you would connect the PC directly with the ATMega, the micro controller would become destroyed.

If you want to use the RS232 interface, the following example might be helpful for you. Using the interface is also helpful during debugging of your code, as you just "print" variable values to the interface and check at the terminal program of your connected PC if the variables contain what you expect. At MS Windows ® you may use either the Hyperterminal® which comes with MS Windows® or with Bascom® you may use the internal monitor for this. You may use the following program to test the output of your module and the terminal program of your PC.

```
'sample program RS232 output
$regfile = "m128def.dat"
$crystal = 8000000
$baud1 = 9600

Open "COM2:" For Binary As #1
Do
  Print #1 , "Hello world"
  Wait 1
Loop
Close #1
End
```

RS232 and the crystal / Overclocking the board

If you are planning of sending a lot of information through the RS232 interface you need to know, that the frequency for the selected baud rate is calculated by the micro controller using the current clock rate. Two facts are important to know:

a) The internal resonator is not very accurate and varies with different temperatures etc. So if the board runs with internal 8 MHz, using of the RS232 interface may result in transmission problems. You better use an external crystal then.

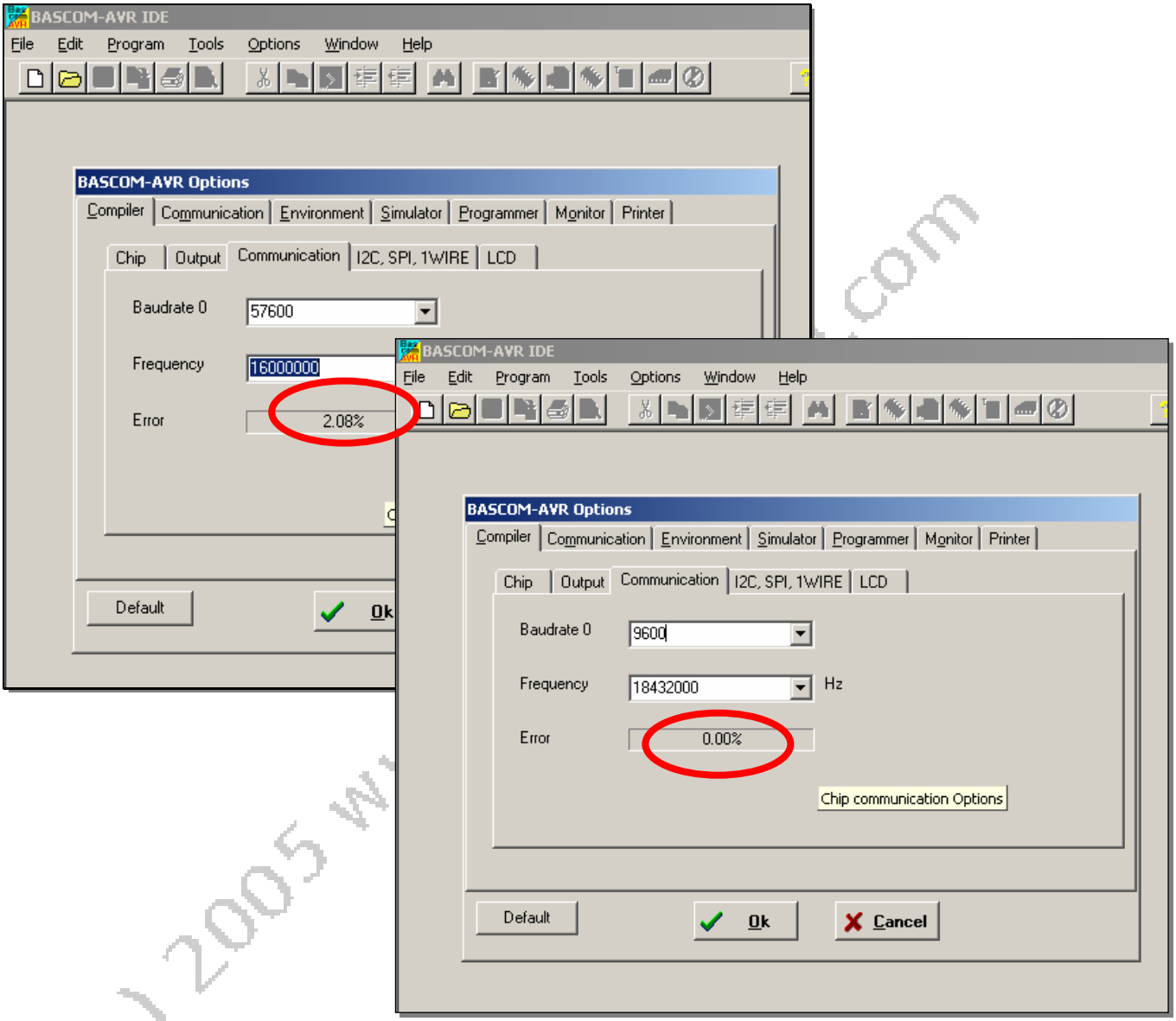
b) Using the usual 16 MHz crystal as the external clock will result in a not 100% correct frequency of the RS232 interface – varying by the selected baud rate. The perfect match would be a crystal of 14.7456 MHz or 18.432 MHz instead, as they will result in 100% correct frequency. At 14.7456 the micro controller is a bit slower, with 18.432 you are overclocking the micro controller. Usually this will not result in problems as the ATMega128 can easily run at a even higher clock rate. The internal EEprom is the area which will first show errors during overclocking – you will not be able to read or write correctly to it. If you do not need the Eeprom you may run the board even at 20 MHz.

c) You need to tell the controller, what clock frequency you are providing – otherwise the wrong calculation is done and the transmission will not work. In Bascom® you do this with the command **`$crystal = 8000000`** at the beginning (8000000 for 8 MHz; 16000000 for 16 MHz, 14745600 for 14.7456 MHz etc.).

You always need to enter the exact speed of your crystal – do not enter any different value, as this will cause in a wrong timing.

At Bascom®, there is a calculator included, which baudrate is possible with the selected frequence. You will find this at the menu **Options / Compiler / Communications**.

See Screenshots:



This will help you to check if you get an error free transmission with your crystal.

In short:
As the internal 8 Mhz is not accurate you should always solder an external crystal in if you want to use RS232.
16 MHz creates usually 0,16% error rate at slower speeds which is OK. Usually an error rate of less than 1% should not result in a higher error rate during transmission.

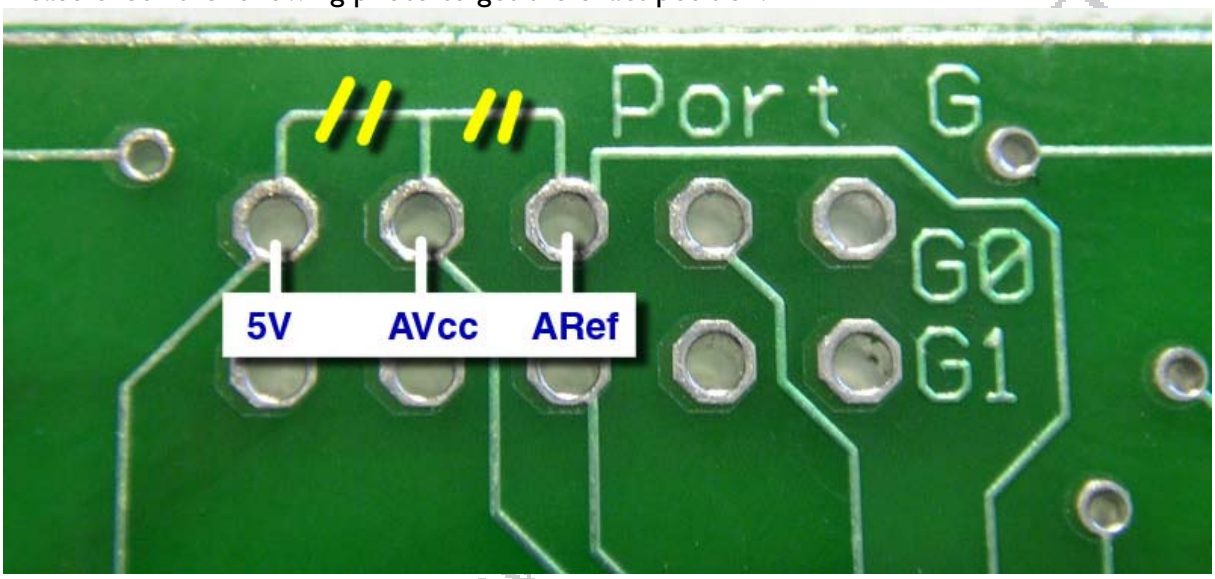
14.7456 MHz or 18.432 MHz will result in 0,00% error rate and is the perfect selection when using RS232.

Special case: analog inputs

Some ports of the processor may work with analog inputs. If somebody needs very exact values they would like to use the three ports for reference voltage AGnd, AVcc and ARef. Most of the time these ports are not used for this purpose and therefore AGnd has to be connected to GND and AVcc to Vcc. We prepared the board to allow both usages:

The pins AVcc and ARef are alreafy connected to Vcc by a connection. If you want input your own voltage at AVcc and/or ARef you need to cut the wires at the PCB at the shown location. The connection of AVcc and ARef are possible at the connector of Port B.

Please check the following photo to get the exact position:



The Port AGnd usually will be connected to central GND – so we did in this case – AGnd is not available at a connector.

To make sure: Most people do not need to change the default. Even if you want to use the analog inputs you do not need to open the wire – only if you need very exact accuracy this might help. As long as you do not cut the wires, the pins AVcc and ARef are connected to the usual Vcc line (5 Volt).