

How to Program the Scope

Model: 06201

The firmware of the scope resides in its core controller, ATmega64. This is an AVR series micro-controller from Atmel. There two ways to program the chip for 06201. One is using a programmer. The other is via the pre-loaded bootloader. This design note explains both.

Program the Scope with a Programmer

To do this you need to prepare following tools:

- 1) An AVR programmer
- 2) Software controlling the programmer

AVR is one of the most popular micro-controllers. It shouldn't be too difficult to find a ready-made programmer for it. There are also plenty of resources on the web for building one yourself if you don't want to buy. Usually they include information for related controlling software. More over, you can use FPGA/CPLD programmers from some big names as an AVR programmer, like that from Altera or Xilinx. You can also use AVR STK200 parallel programmer (or its compatible). The good thing for a STK200 programmer is that it works with PonyProg, a famous free programming software (www.lancos.com). This article will use Altera programmer ByteBlasterMV as an example since the programming header on board was designed compatible with it. For any other programmer you just need to make a simple dangle to route signals to their correct pins respectively. (JYE Tech provides programmers that are compatible with AVR STK200 and Altera ByteBlasterMV. Please see <http://www.jyetechnology.com/en/default.html>.)

The controlling software for Altera's ByteBlasterMV was designed by Chan. It can be downloaded from http://elm-chan.org/works/avr/report_e.html. This program runs in a DOS interface. Detailed instructions for usage can be found within the download.

These are general programming steps:

- 1) Identify the programming header (J4) and its orientation (Fig. 1 and Fig 2).

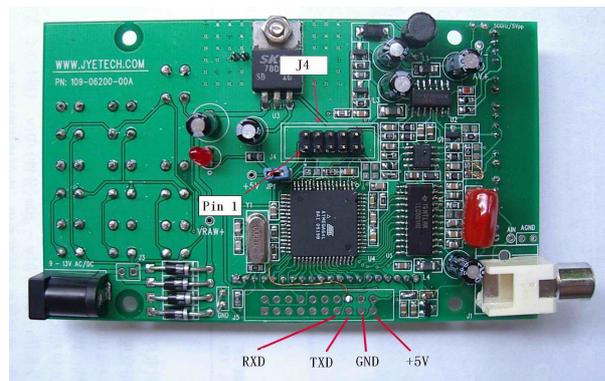


Fig. 1

J4 pin out and signal names

SCK	1	2	GND
MISO	3	4	VCC
nRST	5	6	
	7	8	
MOSI	9	10	GND

Fig. 2

- 2) Correctly connect J4 to programmer with programming cable.
- 3) Connect the programmer to PC parallel port.
- 4) Power up the scope.
- 5) Check if the programmer and connections work properly by typing in command "avrsp -pl1 -rf". This command reads fuse information of ATmega64. If it returns valid results then go to next step.
- 6) Run command

```
avrsp -pl1 113-06201-050.hex [113-06202-010.hex]
to burn in the main firmware [ and bootloader].
```

- 7) Run command

avrsp -pl1 113-06201-050.eep

to burn in the EEPROM data (You need to program EEPROM if the board is the first time being programmed).
- 8) Program fuse bytes in the following order (This is only needed at the chip is programmed the first time or the fuses was changed).

avrsp -pl1 -fx11111111 (Fuse extended byte)

avrsp -pl1 -fh11000000 (Fuse higher byte)

avrsp -pl1 -fl10101110 (Fuse lower byte)
- 9) If no error reported the programming is done. Power down scope and remove programming cable.

Update the Scope Firmware via Bootloader

The scope was preloaded with a bootloader. Each time the scope is powered up it enters the bootloader first and stay there for about 2 – 3 seconds. During this period the bootloader will check serial port for communication request. If a request is found and communication established it will accept firmware download and flash the newly downloaded firmware into ATmega64 itself. If nothing happens it will boot into scope main firmware and start normal operations.

To use the bootloader following tools are needed:

- 1) Serial level converter.
- 2) Download program running on PC.

Level converter is needed because the scope serial port signal level (0 - +5V) is not compatible with standard serial port signal level (+/- 3V - +/- 25V). A simple level converter can be easily made with a few common components. The schematic is available at JYE Tech web site (http://www.jyetechnology.com/Products/FunctionModules/SerialAdapter/105-04000-00B_S.pdf). Finished product can also be purchased from the same web site (PN: 100-04000-00B).

The downloading program on PC side (avrubd.exe) was designed by [Shao Ziyang](http://blog.ednchina.com/shaoziyang/) (<http://blog.ednchina.com/shaoziyang/>). This program communicates with the bootloader and transfer codes to be burn to the bootloader. It can be downloaded from JYE Tech web site (<http://www.jyetechnology.com/Products/LcdScope/avrubd.rar>).

Steps to use the bootloader:

- 1) Connect serial level converter to the serial port pins (shown in Fig. 1) of the scope board.
- 2) Connect the other end of level converter to PC (Fig. 3).

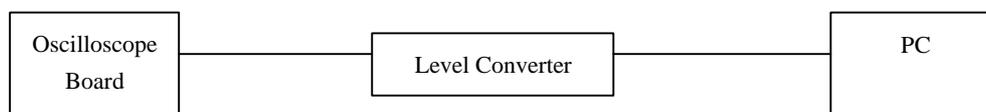


Fig. 3

- 3) Run AVRUBD.EXE. Click “Language” and select “English”.
- 4) Click on “Options” at main menu. Change settings on tab “AVR” to those shown in Fig. 4. Connect Key has to be “HEX 64”. Flash size should be 64K.
- 5) Change settings on tab “System” to those shown in Fig. 5. Communication protocol must be selected to be “Lite Xmodem”.
- 6) Change settings on tab “Com Port” to those shown in Fig. 6. Com port number should be choose according to actual port used. Communication format is 9600 bps, 8N1 with no parity and flow control.
- 7) Load HEX file to be downloaded into PC after finishing settings above. Click “File –

Load” and select your file.

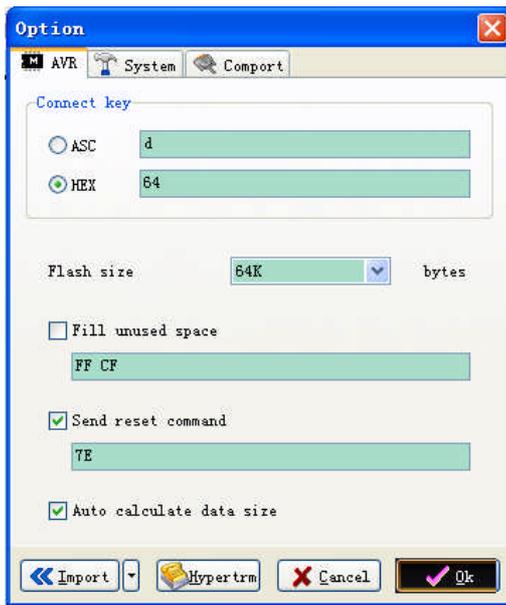


Fig. 4

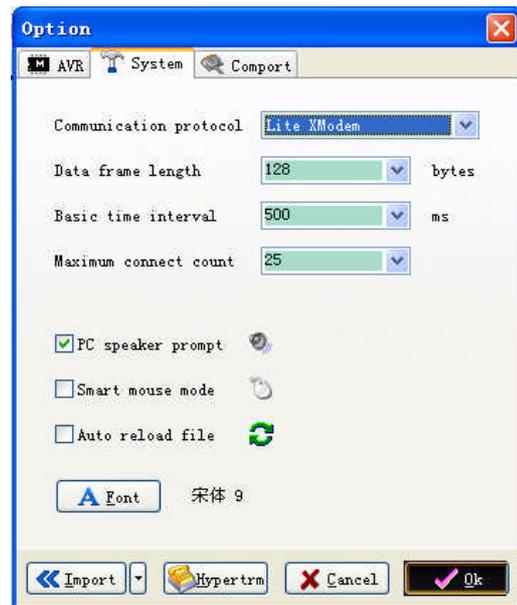


Fig. 5

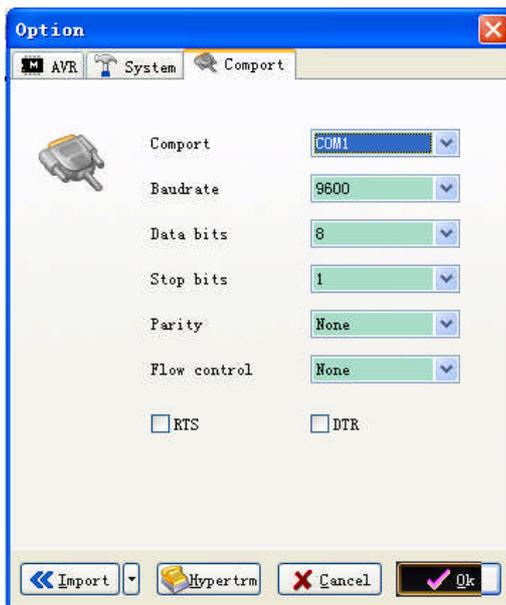


Fig. 6

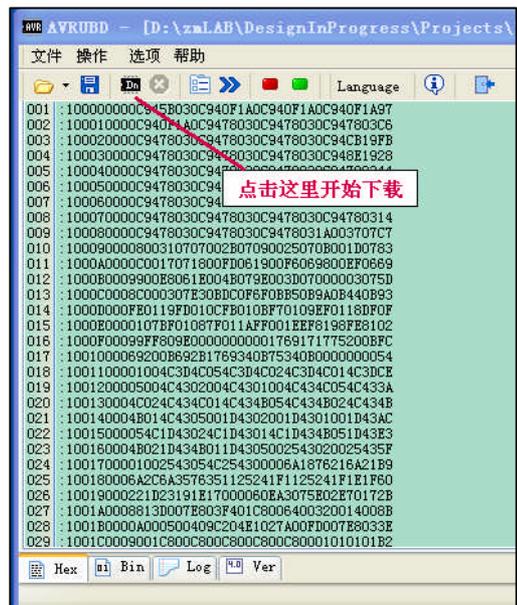


Fig. 7

- 8) Keep scope power down off. Click on “Dn” icon (see Fig. 7. Alternatively you can click on “Operate – Download” from the main menu) and power up the scope right after. You will see AVRUBD is trying to communicate with the scope board. If handshake failed you would see screen like that of Fig. 8.
- 9) Otherwise, Download is successful if you see screen like Fig. 9.

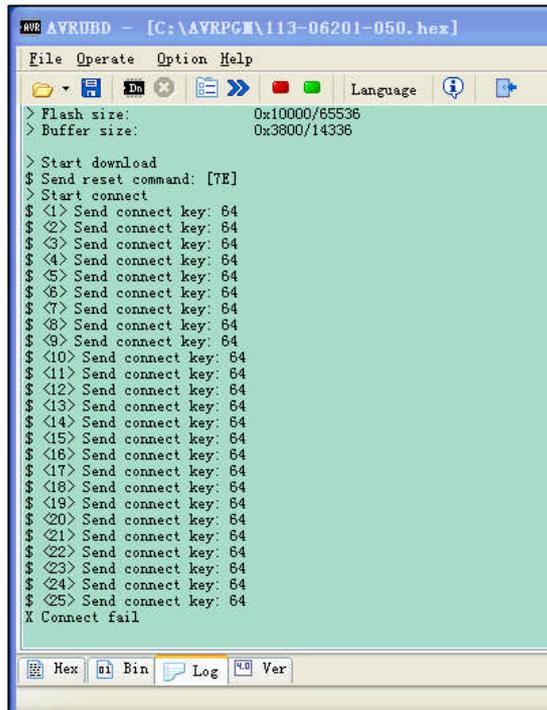


Fig. 8



Fig. 9

Notes

Fuse configuration of ATmega64 is as follows. Usually these settings should not be changed. In case of these bytes are modified programmer is needed to restore them. They can not be modified with bootloader.

- Fuse Extended Byte: 0bxxxxx11**
- Fuse Higher Byte: 0b11x00000**
- Fuse Lower Byte: 0b10101110**

Revision History

Version	Date	Descriptions
01	2008.09.28	First created