

DIY KIT 122. ATMEL AVR PROGRAMMER

This kit is a simple but powerful programmer for the Atmel AT90Sxxxx (“AVR”) family of microcontrollers.

The Atmel AVR devices are a low-power CMOS 8-bit microcontroller using a RISC architecture. By executing instructions in a single clock cycle, these chips achieve throughputs approaching 1 MIPS per MHz. Get all information and data sheets from

atmel.com

These devices feature onboard Flash program memory, EEPROM for nonvolatile data storage, as well as a number of fuse and lock bits. They are electrically erasable; they can be re-programmed over and over again without the need for UV erasers.

Programming of all these features is supported with this programmer.

DEVICES SUPPORTED

- **AT90S1200** (20 pin DIP)
1K program, 64 bytes EEPROM
- **AT90S2313** (20 pin DIP)
2K Flash, 128 bytes EEPROM & SRAM
- **AT90S4414** (40 pin DIP)
4K Flash, 256 bytes EEPROM & SRAM
- **AT90S8515** (40 pin DIP)
8K Flash, 512 bytes EEPROM & SRAM

- **AT90S4434 (40 pin DIP)**
4K Flash, 256 bytes EEPROM & SRAM
(requires adaptor K122ADT)
- **AT90S8535 (40 pin DIP)**
8K Flash, 512 bytes EEPROM & SRAM
(requires adaptor K122ADT)

This programmer does not support In-System-Programming.

The programmer uses a **serial** port for communication, which has several advantages:

- It allows the programmer to be used with any computer and operating system.
- The programmer does not require special software other than a communications program that can send an ASCII text file. Windows 3.11 & 9x come with this program – Terminal or Hyperterminal – built-in. Or you may use our **term.exe** from our website.

Any comms program will do. We use our own term.exe. You may download this from

<http://kitsrus.com/zip/term.zip>

The zip file includes a detailed explanation of how to setup and use term.exe under W98. Also printed below.

The kit is constructed on a double-sided, through hole plated printed circuit board (PCB) measuring 110mm x 69mm (4.3” x 2.7”). Protel Autotrax & Schematic were used to design the PCB.

The kit is powered by a mains adaptor with an open circuit output voltage of at least 16VDC. Most 12VDC adaptors should supply this quite easily. Current capacity of the adaptor should be at least 150mA.

Provision is made on the PCB to fit ZIF programming sockets. **However these are not supplied with the kit.**

ASSEMBLY INSTRUCTIONS

A number of the components are physically similar and can be easily mixed up. Before starting, identify the following components:

- 1N4148 diode **1N4148 marked on body**
- 5.6V zener diode **5V6 marked on body**
- 12V zener **12 or 12V marked on body**
- BC547 transistor
- BC557 transistor
- 78L05 regulator

Using the component overlay on the PCB, insert the components in the following order:

1. Resistors and diodes
The diodes must be inserted the correct way around. The “bar” on the diode body lines up with the “bar” on the component overlay. The dot on the resistor network RP1 **must** go into the box marked on the overlay for RP1. RP1 may be 10P9R or 9P8R.
2. Ceramic and monobloc capacitors
3. IC sockets (not the ZIF sockets if used)
4. Transistors and 5V regulator
5. LEDs
6. Electrolytic capacitors
Make sure that the electrolytic capacitors are inserted the correct way around. The positive lead is marked on the overlay. The negative is marked on the body of the capacitor.
7. Crystal, pushbutton switch and DC jack
The switch has a flat side which lines up with the “bar” on the component overlay.
8. D25 connector and ZIF sockets (if used)
9. Proceed to “TESTING” before inserting any ICs.

CIRCUIT DESCRIPTION

The 78L05 regulator provides a stable 5V supply for the ICs. Diode D1 protects the kit against reverse polarity of the power supply. Transistors Q1 and Q2 are used to control the programming voltage. Q2 switches the programming voltage on or off and is controlled via Q1. A high on the base of Q1 pulls the base of Q2 low and the programming voltage is switched off.

A low on the base of Q1 causes zener Z1 and diode D2 to conduct and 12.6V is applied to the base of Q2. The programming voltage is now 12V. Resistor R2 limits the current supplied to the programming pin. With a 12V

DIY KIT 122. ATMEL AVR PROGRAMMER

programming voltage zener Z2 conducts and the “Vpp On” LED lights.

Transistor Q3 switches the supply voltage to the programming socket. A low on the base supplies 5V to the socket and the “Vcc On” LED lights.

The control software for this kit is contained in IC1, a pre-programmed 89C2051. It controls all the functions for reading, verifying and programming of the AVR chips, including the EEPROM, lock and fuse bits. Port pins P1.0 to P1.7 are used to transfer data to or from the programming socket. IC2, an 8-bit addressable latch, provides additional outputs and is controlled via port pins P3.2 to P3.6 and P3.7. The in-built serial port on pins P3.0 and P3.1 is used to communicate with the host PC. IC3 converts between TTL and RS232 signal levels.

TESTING

Before applying power, check that all parts are inserted in the correct position. Make sure the electrolytic capacitors and diodes are the right way around.

With no ICs inserted apply power via the DC jack. The “VPP” LED should come on. Check the following:

1. +5V rail - measure between pins 10 and 20 of IC1 socket.
2. VPP voltage - approximately 12V on pin 1 of SKT1 and pin 9 of SKT2 (“Vpp” LED on)
3. VCC voltage - 0V on pin 20 of SKT1 and pin 40 of SKT2.
4. Insert a wire link to pins 6 and 16 of the IC2 socket. The VPP voltage should now be 0V.
5. Move the wire link to pins 7 and 8. The “Vcc” LED should be on and the voltage on pin 20 of SKT1 and pin 40 of SKT2 should be 5V.

If all is well, remove power and insert the ICs. When power is then applied the Vcc LED will turn on for 0.5 sec then turn off. Both LEDs will be off.

OPERATION & USE

1. Connect the programmer to the serial port of a PC or other host using a “straight through” cable.
2. Start a ‘terminal program’ such as Windows Terminal, Windows 95/98 HyperTerminal or DOS Telix, ProComm, Boyan. etc.
3. We use our own term.exe. You may download this from

<http://kitsrus.com/zip/term.zip>

The zip file includes a detailed explanation of how to setup and use term.exe under W98.

4. Set the communications parameters to **9600 baud, 8 data bits, 1 stop bit and no parity bit**. If you use term.exe enter ‘term 9600’. (Make a desktop icon with ‘term.exe 9600’ as the ending in the path.)
5. Apply power. A menu will appear and both LEDs should be off. (If the menu does not appear hit Reset

on the K122 PCB, or Esc on the keyboard followed by Reset. Or Enter, followed by Reset.)

6. The programmer is ready for use.

NOTE: Do not insert or remove ICs in the programming sockets until the programmer is powered up. Once powered do not insert or remove ICs until both LEDs are off.

PROGRAMMING COMMANDS

- Commands may be entered in upper or lower case.
- There is no need to select the chip being programmed. The programmer automatically identifies the device before any programming option is executed. An error message is printed for unknown devices.

With term.exe using Kit 122 to program AT1200 the usual keypress sequence is

C P alt-S enter R P L

The first time we hit alt-S we have to put in the name of the file to be sent to the programmer (eg, ‘k129.hex’) where k129.hex is in the same folder as term.exe. The next time you enter alt-S the entry is remembered. You might like to do **V alt-S enter** a after the **P alt-S** the first few times to verify the program is correct. **X** as the optional final command shows the lock bits LB1 & LB2 status (0 for locked; 1 unlocked.) **C** may be omitted if the IC’s are new and known to be blank.

DESCRIPTION OF COMMANDS

P Program memory

This will program the currently selected memory (flash or eeprom). You will be prompted to send the file, which must be in Intel HEX format. Use an ASCII or text transfer to send it.

Before programming a chip it should be erased. Chips that have been “locked” cannot be programmed without erasing first. If an error occurs while programming, a message will be printed and the programmer will stop. Stop the file transfer and press the reset switch to continue.

V Verify memory

Verify the selected memory (flash or eeprom) against an Intel HEX file. You will be prompted to send the file. An error message will be printed if verification fails.

Note: Verification should be performed BEFORE writing the lock bits. Writing the lock bits prevents the code from being read out. All data will read as FFh and verification will fail.

D Dump memory

Read the contents of the selected memory (flash or eeprom) and send it to the PC. The data is converted into Intel HEX format before sending. A terminal

DIY KIT 122. ATMEL AVR PROGRAMMER

program with input capture or logging allows the data to be saved to a disk file.

L write Lock bits

Used to program lock bits 1 and 2. All lock bits are programmed – there is no choice. Lock bits are set to ‘0’ when they are programmed (locked).

C Chip erase

Erase the device (electrically). Erasing the device does NOT affect the fuse bits. These are erased using the ‘S’ and ‘R’ commands.

F address Flash memory

AVR devices have two types of programmable memory – flash (code) memory and eeprom (data) memory. This command, together with the ‘E’ command, is used to select which memory type is being referenced by the ‘P’, ‘V’ and ‘D’ commands.

The command prompt will show which memory type is currently selected.

E address Eeprom memory

See ‘F’ command above.

Sx SPIEN fuse (P)rogram/(E)rase

Programs or erases the AVR fuse bit “SPIEN”. “SP” programs the fuse, “SE” erases it.

Rx RCEN (FSTRT) fuse (P)rogram/(E)rase

Programs or erases the AVR fuse “RCEN” (“FSTRT”)

On the AT90S1200 chip it is called the “RCEN” fuse. On each of the other chips it has a different function and is call the “FSTRT” fuse. In either case this command is used to program or erase this fuse bit.

“RP” programs the fuse, “RE” erases it.

X display Fuse and Lock bits

This command displays the current status of the fuse and lock bits.

IF IT DOES NOT WORK

Poor soldering (“dry joints”) is the most common reason for the circuit not working. Check all soldered joints carefully under a good light. Re-solder any that look suspicious. Check that all components are in their correct position on the PCB. Are the electrolytic capacitors and diodes the right way round? Is the power supply voltage at least 16VDC? Is the programming voltage (12V) correct? Is the reset switch the right way round? Check the resistor network RP1 is the right way around.

Web Address & Email

You can email us at peter@kitsrus.com if you have any problems or requests. Our web page is at:

<http://kitsrus.com>

NOTES:

September, 2000.

An Adaptor Board for programming the AT90S4434/8535 chips is now available. It is supplied with 2 x 20 in SIL headers and a 40 pin IC socket. The user can use a 40 pin ZIF socket if they wish.

March, 2001.

Correct connection of C8 on the serial interface. It was around the wrong way but it did not affect operation. The schematic with this documentation is the correct one.

Small firmware bug fix for 2313 programming. Last hex byte was not programming properly.

PARTS LIST - KIT 122

Resistors (0.25W carbon)

100	R2,10	2
680	R9	1
1K2	R3,6,11	3
3K3	R4	1
4K7	R5,8	2
8K2	R1	1
10K	R7	1
10K SIL resistor network.....	RP1 9 or 10 pin ‘A’.....	1

Capacitors

27pF ceramic	C3,4	2
100nF monobloc	C1,2,12,13	4
10uF 25V electrolytic	C5,6,7,8,9,11	6
100uF 25V electrolytic	C10	1

Semiconductors

1N4004	D1	1
1N4148	D2	1
LED, 5mm, red.....	D3,4.....	2
5V6 400mW zener.....	Z2	1
12V 400mW zener.....	Z1	1
BC547 transistor, NPN.....	Q1,2.....	2
BC557 transistor, PNP	Q3.....	1
AT89C2051.....	IC1 preprogrammed	1
74HC259 8-bit latch	IC2.....	1
MAX232 or equivalent.....	IC3.....	1
Dual RS-232 transmitter/receiver		
78L05 regulator.....	IC4.....	1

Miscellaneous

Crystal, 20.2752MHz.....	Y1	1
D9 connector	X1	1
PCB mounting, right-angle, female		
Pushbutton switch.....	SW1	1
2.5mm DC jack.....	X2.....	1
16-pin IC socket.....	for IC2,3.....	2
20-pin IC socket.....	for IC1, SKT1.....	2
40-pin IC socket.....	for SKT2.....	1
PCB, K122.....		1

DIY KIT 122. ATMEL AVR PROGRAMMER

How to Use 'term.exe' Communications Program

Some kits (K121, 122, 123, 164) require a communications program to transfer data between the serial port on the PC and the kit. In Windows 9x there is a comms program called 'Hyperterminal' already built-in. In Windows 3.1x comes with the 'Terminal' comms program built in. There are many comms program you can download from software sites.

Frank Crivelli (www.ozitronics.com) has written his own DOS-based comms program called 'term.exe'. It is a simple, basic terminal program which does its job without a lot of 'bells & whistles'. Also as of this moment he does not know how to write Windows software!

You may download it from <http://kitsrus.com/zip/term.zip>

How to install 'term.exe'

This is a detailed explanation of how I have installed 'term.exe' in my Windows 98 system. There are slight differences with Windows 95 and 2000 but it gives you an idea of how to do it.

1. Unzip 'term.zip'. You will get this file and 'term.exe'.
2. Move 'term.exe' to the same folder as Hyperterminal.
C:\Program Files\Accessories\Hyperterminal
(I did this because sometimes I use Hyperterminal and all my hex files for programming firmware are in the same folder.)
3. Right click on some blank space on the desktop and select "New → Shortcut".
4. Click on the "Browse" button and find 'term.exe' on your hard disk in the above mentioned folder
5. Click on it and select "open". Click "Next", "Next" then "Finish".

You should now have an icon on your desktop. If you want to change the icon's name then right click on it and select "Rename".

6. Right click on this icon and select "Properties".
7. Click on the "Program" tab.
8. Go to the end of the "Cmd line" box and type in " 9600" (you must include the leading space). If you are using COM2 then type " /2 9600" instead. Notice that the "Working" directory/folder is set to the same as the "Cmd line". You can change this if necessary.
9. Tick the "Close on exit" box. This will shutdown the DOS window when you quit 'term.exe'.
10. Click on the "Change Icon..." button if you want to change the icon associated with this shortcut.
11. Click on the "Screen" tab and select "Full-screen".
12. Click "OK".

To run 'term.exe' simply click on the desktop icon.

Now see if you can communicate with the kit.

DIY KIT 122. ATMEL AVR PROGRAMMER

Communicating with the Kit

Each of our kits has a Menu in the kit Firmware. Power the kit, attach a straight-thru serial cable, then click on the term.exe icon. Or start term.exe, connect power and the serial cable. The order does not really matter.

Now usually pushing the tact switch or, in the case of Kit 164, hitting Enter, will bring up the Menu. But sometimes not. Do Alt-Q to exit the program, then click on the term.exe icon again to start it again. Almost always this restart will bring up the Menu after pressing the tact switch or hitting Enter. If not do the restart again. You should not need more than 2 restarts.

Note F1 brings up the Help features. In particular, Alt-D goes to a DOS shell so you can do a dir *.* or dir *.hex to see what files are in the folder. Type 'exit' to get out of the DOS shell. Alt-L starts the logging to a file feature (needed for Kit 164.) When logging is on note 'LOG' appears on the bottom line.

Programming with Kits 121 122 123

I suggest you put the hex file you want to program into the same folder as term.exe. Then when you do an 'Alt-D' followed by a 'dir *.hex' all the hex files will be in the same folder. You will not have to change folders using DOS commands (which an increasing number of people today do not know.) Exit the folder by typing 'exit' and hitting Enter.

Put in the chip to be programmed. The Blank (B) and Erase (C or E – read the menu) commands are obvious.

To send the hex file to the programmer hit P for program. You will be prompted 'Send file now ...' Now hit Alt-S which is (press F1) 'Send text file'. Enter the hex file name. You will be prompted for the hex file name. Type it in then press Enter. The file will now be sent to the chip to be programmed on the programmer. You can do a Verify you will again be prompted to 'Send File Now ...'. Just hit Alt-S. The correct file name is remembered. Hit Enter and the file will be sent to be verified.

Finally do the required Lock bit command(s).

In the latest Kit 123 firmware there is a Q Bulk Programming feature built in which does C P L. You still have to do the Alt-S to send the file to the Programmer.

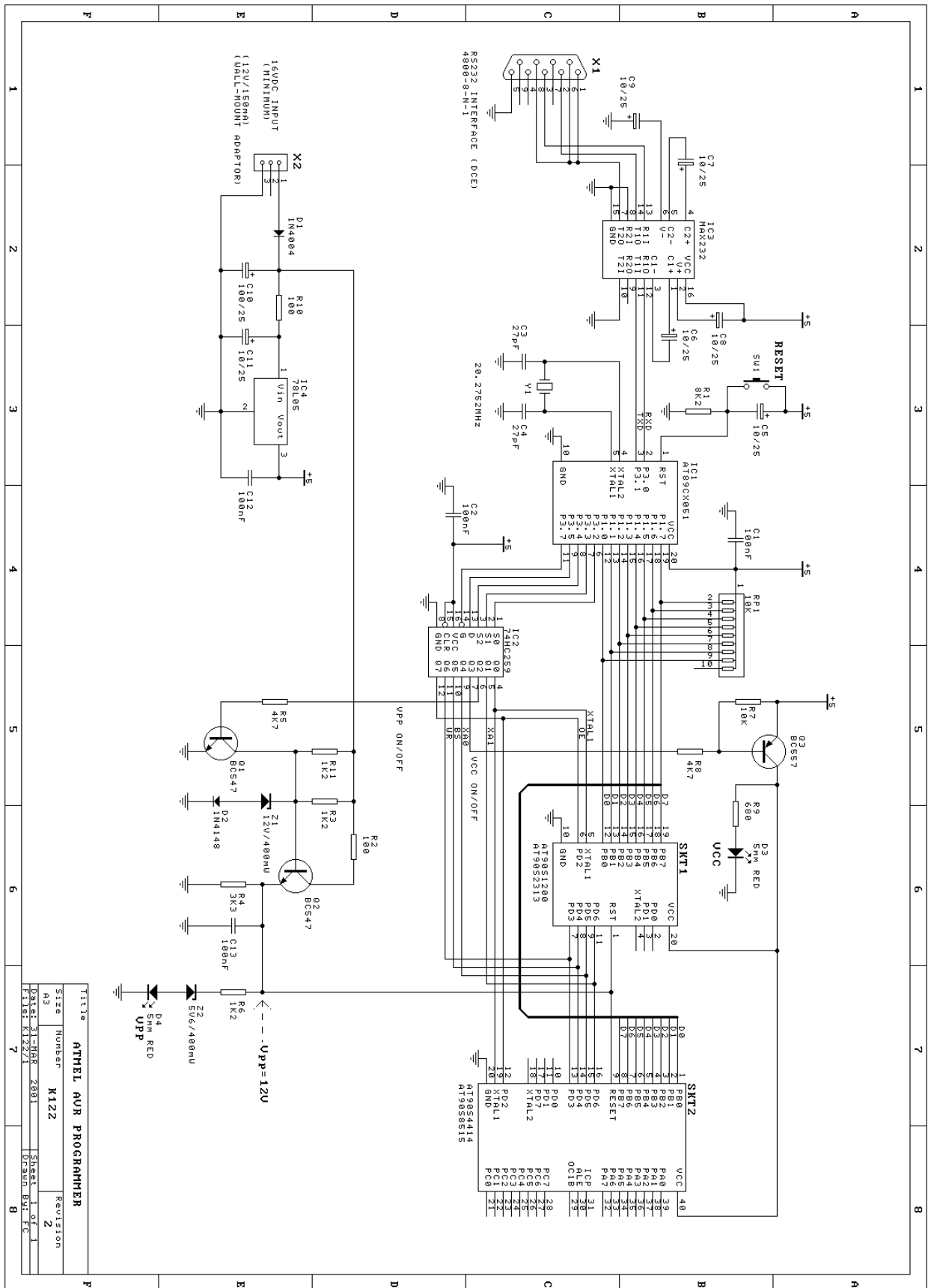
Contact Us

You may contact the term.exe writer at frank@ozitronics.com

If you have suggestions on how to improve this documentation email me at peter@kitsrus.com

August 9, 2001.

DIY KIT 122. ATMEL AVR PROGRAMMER



Title		ATMEL AVR PROGRAMMER	
Size	Number	Revision	
A3	K122	2	
Date: 31-08-2001		Sheet 1 of 1	
File: K122/1		Drawn: BJC	