CK1705 ATMEL AVR PROGRAMMER

This kit is a powerful programmer for the Atmel AT90Sxxx (“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.

AVR 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

This programmer does not support In-System-Programming.

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

- The programmer does not require special software other than a terminal emulator program that can send an ASCII text file. Windows 3.11 &9x come with this program (**Terminal** & **Hyperterminal**) built-in. You may download our own term.exe from our website.
- It allows the programmer to be used with any computer and operating system.

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 for DOS was used.

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 the body
- 5.6V zener diode **1TTB 5V6** marked on body
- 12V zener **BZX 55C 12 ST** 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 goes into the box marked on the overlay for RP1. Network may be 10P9R (last resistor not used) 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 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
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.

OPERATION & USE

1. Connect the programmer to the serial port of a PC or other host using a "straight through" 9 pin cable.
2. Start a 'terminal program' such as Windows Terminal, Windows 95/98 HyperTerminal or DOS Telix, ProComm, etc. Or use the term.exe program which you may download from our website.
3. Set the communications parameters to **<u>9600 baud, 8 data bits, 1 stop bit and no parity bit</u>**. (The pcb has 4800 written on it. Disregard this. After the first PCBs were made we found that 9600 baud could be achieved.) If you get term.exe enter 'term 9600'
4. Apply power to the programmer. A menu will appear and both LEDs should be off. You may have to press the RESET button to get the menu,
5. 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 CK1705 to program the AT1200 for our Kit 129 the usual command sequence we use is

C P alt-S enter RP L

The first time we hit alt-S we have to put in the name of the file to be sent to the programmer ('k129.hex' in our case.) The next time we enter alt-S the name is remembered. You might like to do a **V alt-S enter** after the **P alt-S** the first few times to verify that the program has gone correctly. **X** as the optional final command shows the lock bits status (=0 for locked; =1 unlocked.)

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 using the **C** command – see below. 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.

<u>Note:</u> Verification must be performed BEFORE writing the lock bits. Writing the lock bits prevents the code from being read out or the fuses being set. 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 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). They are '1' if unlocked.

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 Eprom 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.

Note: setting the fuses must be done **before** Locking the program with the **L** command.

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?

Web Address & Email

You can email the developer of this kit at

frank@ozitronics.com

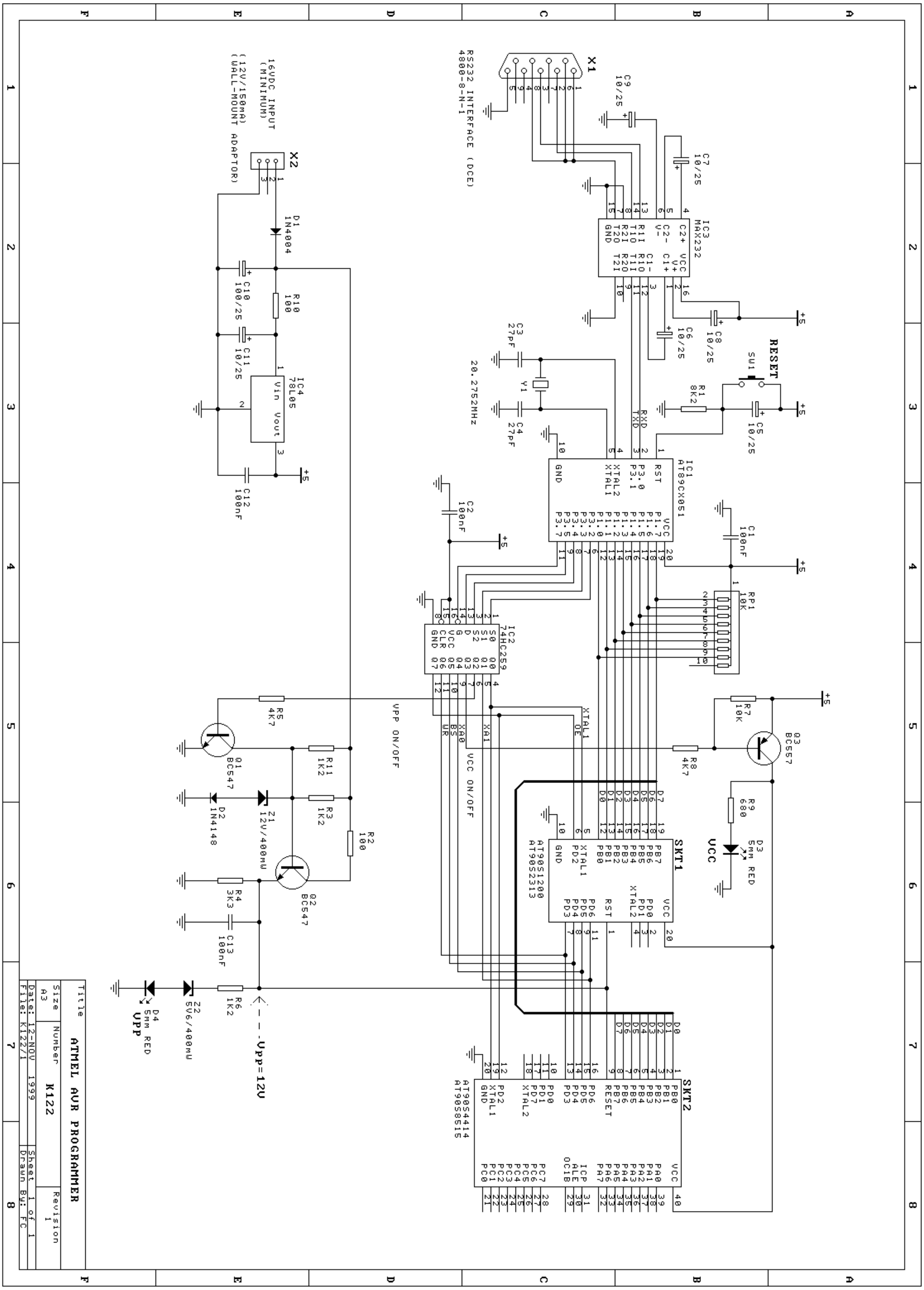
if you have any problems or requests. Information on other kits in the range is available from our Web page at:

http://www.EletronicKits.com

PARTS LIST - KIT 122

Resistors, 5%, 0.25W carbon:		
100R	R2,10	2
680R	R9	1
1K2	R3,6,11	3
3K3	R4	1
4K7	R5,8	2
8K2	R1	1
10K	R7	1
10K SIL resistor network 9 or 10 pin	RP1 10P9R or 9P8R	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 (**1N4004** on body)	1
1N4148	D2 (**1N4148** on body)	1
LED, 5mm, red	D3,4	2
5V6 400mW zener	Z2 (**1TTB 5V6**)	1
12V 400mW zener	Z1 (**BZX 55C 12ST**)	1
BC547 transistor, NPN	Q1,2	2
BC557 transistor, PNP	Q3	1
AT89C2051 Microcontroller, pre-programmed	IC1	1
74HC259 8-bit addressable latch	IC2	1
MAX232 or similar Dual RS-232 transmitter/receiver	IC3	1
78L05 +5V regulator, TO-92 package	IC4	1
Miscellaneous		
Crystal, 20.2752MHz	Y1	1
D9 connector PCB mounting, right-angle, female	X1	1
Pushbutton switch	SW1	1
2.5mm DC jack PCB mounting	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

CK1705 ATMEL AVR PROGRAMMER


X1
RS232 INTERFACE (DCE)
4800-8-N-1
C9 10/25
C7 10/25
IC3 MAX232
C6 10/25
C8 10/25
RESET
SW1
R1 8K2
C5 10/25
C3 27pF
Y1
20.2752MHz
C4 27pF
RXD
TXD
IC1 AT89CX051
C2 100nF
C1 100nF
RP1 10K
IC2 74HC259
R7 10K
Q3 BC557
R8 4K7
R9 680
D3 5mm RED
VCC
SKT1
AT90S1200
AT90S2313
SKT2
AT90S4414
AT90S8515
X2
16VDC INPUT (MINIMUM)
(12V/150mA)
(WALL-MOUNT ADAPTOR)
D1 1N4004
C10 100/25
R10 100
C11 10/25
IC4 78L05
Vin Vout
C12 100nF
R5 4K7
Q1 BC547
R11 1K2
D2 1N4148
Z1 12V/400mW
R3 1K2
R2 100
Q2 BC547
R4 3K3
C13 100nF
R6 1K2
Z2 5V6/400mW
D4 5mm RED
VPP
Vpp=12V
VPP ON/OFF
VCC ON/OFF
XTAL1
OE
XA0
XA1
BS
WR
+5
Title
ATMEL AVR PROGRAMMER
Size A3
Number K122
Revision 1
Date: 12-NOV 1999
Sheet 1 of 1
File: K122/1
Drawn By: FC