QUASAR PROJECT KIT # 3123 - ATMEL 89xxxx PROGRAMMER

INTRODUCTION

This kit is a simple but powerful programmer for the following Atmel 8051 family of microcontrollers:

- 89C1051, 89C1051U, 89C2051 and 89C4051
- 89C51, 89LV51
- 89C52, 89LV52
- 89C55, 89LV55
- 89S8252, 89LS8252
- 89S53, 89LS53
- 87F51, 87F52 (OTPs)

These devices (except 87F51/52) use flash memory to store programs. Flash memory can be electronically erased and programmed – no need for a UV eraser. The 89(L)S8252 chips has EEPROM that can also be programmed with this kit.

All devices have signature bytes that the programmer reads to automatically identify the chip. No need to select the device type – just plug it in and go! Even includes a "bulk programming" feature to reduce keystrokes when programming multiple chips.

All devices also have a number of lock bits to provide various levels of software and programming protection. The 89x8252 and 89x53 chips have a serial programming fuse as well. All fuse and lock bits are programmable using this kit.

The table below shows the key differences between the various microcontrollers. Data sheets for each device can be found on the Atmel web site at

http://www.atmel.com

The programmer connects to a host computer using a standard RS232 serial port. All the programming "intelligence" is built into the programmer so you do not need any special software to run it. This makes the programmer usable with any computer and any operating system.

Just connect using any communications program capable of ASCII text file transfer, such as Windows 3.11 Terminal, Windows 9x HyperTerminal or DOS Telix, ProComm, etc. A simple DOS terminal program, 'term.exe', is available from our web site.

SPECIFICATIONS

Dimensions:	118mm x 70mm (4.7" x 2.8")
Power Supply:	12VDC 150mA "unregulated" PlugPack (16VDC minimum)
Interface:	RS-232, 9-pin "D" connector
Data Speed:	9600 bps
Data format:	8 bits, no parity, 1 stop, no flow control
File format:	Intel 8-bit HEX
Program Sockets:	20 pin DIP - 0.3", 40 pin DIP - 0.6"
	Note: ZIF sockets can be used but are not supplied with the kit.
Program Time:	About 2mS per byte

Four LEDs are used to display the programmer's status:

PWR	Indicates when power is applied to the programming socket.
BUSY	On while programmer is executing command. Off when finished.
OK	Indicates if last command was successful
FAIL	Indicates if last command was unsuccessful

COMMAND SET SUMMARY

The programmer is controlled using a set of simple commands issued from a 'terminal program'.

Following is a brief summary of each command. A more detailed description is given later on.

P	-	Program memory
Q	-	Bulk programming (C,P,L3)
V	-	Verify memory
R	-	Read memory
B	-	Blank check
C	-	Chip erase
Ln	-	write Lock bits (n=1,2 or 3)
Sx	-	Serial Programming (E)nable/(D)isable (89S8252/89S53 only)
F	-	address Flash memory
E	-	address Eeprom memory
M	-	Manual device selection
A	-	Automatic device selection (default)
H	-	Help (prints menu)

	89C1051 89C1051U	89C2051	89C4051	89C51 89LV51 87F51	89C52 89LV52 87F52	89C55 89LV55	89S8252 89LS8252	89S53 89LS53
Pins	20	20	20	40	40	40	40	40
Flash (87F51, 87F52 = OTP)	1K	2K	4K	4K	8K	20K	8K	12K
EEPROM	0	0	0	0	0	0	2K	0
RAM	64	128	128	128	256	256	256	256
I/O Pins	15	15	15	32	32	32	32	32
16-bit Timers	1	2	2	2	3	3	3	3
UART	0 (89C1051) 1 (89C1051U)	1	1	1	1	1	1	1
Interrupt Sources	3	6	6	6	8	8	9	9
On-Chip Analog Comparator	Yes	Yes	Yes	No	No	No	No	No

QUASAR PROJECT KIT # 3123 - ATMEL 89xxxx PROGRAMMER

ASSEMBLY INSTRUCTIONS

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

- 1N4148 diode
- 5.6V and 12V zener diodes
- BC547 and BC557 transistors
- 78L05 regulator
- 4MHz and 20.2752MHz crystals

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.
2. Ceramic and monobloc capacitors
3. IC sockets (not ZIF sockets)
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 can only be inserted one way around.
8. D9 connector and ZIF sockets
9. Proceed to "TESTING" before inserting any ICs.

CIRCUIT DESCRIPTION

The 78L05 regulator provides a stable 5V supply for the ICs. Diode D5 protects the kit against reverse polarity of the power supply. LED D7 indicates power on.

Transistors Q1, Q2 and Q3 are used to control the programming voltage. Q3 switches the programming voltage on or off and is controlled via Q1.

Transistor Q2 controls the programming voltage level. With Q2 on, zener Z1 conducts and 5.6V is applied to the base of Q3, giving a 5V programming voltage. With Q2 off, zener Z2 conducts via diode D4 and 12.6V is applied to the base of Q3. The programming voltage is now 12V. Resistor R8 limits the current supplied to the programming pin.

Transistor Q4 switches the supply voltage to the programming socket. The "PWR" LED indicates when power is applied to the programming socket.

The control software is contained in IC1, a pre-programmed 89C51. It controls all the functions for reading, verifying and programming the 89Cxxxx chips, including EEPROM, lock and fuse bits.

IC2 latches the low order address bits when programming the 40 pin devices. IC4 takes care of the TTL - RS232 level conversions.

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.

Before inserting any ICs, apply power via the DC jack. LED D7 should come on. Check the following:

1. +5V output from IC3
2. VPP voltage - 0V on pin 1 of SKT2
3. VCC voltage - 0V on pin 20 of SKT2
4. Insert a wire link between pins 6 and 20 of the IC1 socket. The VPP voltage should now be 5V.
5. Insert another wire link between pins 7 and 20. The VPP voltage should now be 12V.
6. Move the second wire link to pins 8 and 20. The "PWR" LED should come on and the voltage on pin 20 of SKT2 should be 5V.

If all is well, remove power and insert the ICs.

COMMAND PROMPT

The programmer uses the command prompt to indicate its current mode of operation and which address space is currently selected.

The two modes of operation, 'Auto' and 'Manual', refer to how the programmer identifies the device to be programmed. In 'Auto' mode the programmer identifies the device by reading the signature bytes. In 'Manual' mode the user selects the device type from a list.

As mentioned earlier, some devices also have EEPROM memory that can be programmed. The current memory type that the programmer is addressing is indicated by the word 'Flash' or 'Eeprom' in the prompt.

Eg. **[Auto:Flash]** indicates that the programmer is in automatic identification mode and is addressing the Flash memory space. This is the power up default setting.

OPERATION AND USE

WARNING:
Do not power up the programmer with an IC inserted in either of the programming sockets. Also, do not insert or remove any IC while the "PWR" LED is on.

Connect the programmer to the serial port of a PC (or other host computer) using a 'straight through' cable. Start a terminal emulation program running at **9600 baud, 8 data bits, 1 stop bit and no parity bit**. Apply power to the programmer. A menu will appear and all the status LEDs should be off. The programmer is ready for use.

Insert a chip into the programming socket and enter a command. The programmer first attempts to identify the device and prints an error message if unable to do so. If all is well the function executes and a message printed on

completion. When finished the 'PASS' or 'FAIL' LED will indicate the result.

Before programming a chip it should be blank checked and erased if necessary. **Note**: 'Locked' chips will test blank but cannot be programmed without erasing first.

When programming or verifying the user will be prompted to send a file to the programmer. The file must be in Intel HEX format and is transmitted using an ASCII text transfer protocol.

Note: Verification should be done 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.

Any error while receiving the file, such as an invalid hex character or bad checksum, will cause an error message to be printed. The programmer will stop and the user will be prompted to press 'Reset' to continue. Abort the file transfer before continuing.

HEX FILE FRAGMENTATION

Most programs are written starting at address 0000h and continuing up from there. Assemblers simply take this source file and create a HEX file that is a series of one or more continuous address blocks. However some language compilers such as C or BASIC have a habit of creating HEX files that are full of small blocks of addresses that are not in any continuous order.

This can be a problem when programming the 89Cx051 devices. They use an internal address counter to access the flash memory. This counter can only be reset to zero or incremented by one. If the next address to be accessed is lower than the last one then the counter has to be reset to zero and stepped up until it gets to the new address.

This takes time. Meanwhile data is still being sent to the programmer. The programmer uses a small internal buffer to store data as it is received but it can be overrun. If this happens the programmer will stop and print an error message, usually "Hex error".

The problem is fragmentation of the HEX file. A small DOS utility is available from our web site called 'REORDER.EXE'. It takes a fragmented file and 're-orders' it to be in one continuous address block.

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. Are the electrolytic capacitors and diodes the right way round? Have you mixed up the zener diodes or transistors? Is the power supply voltage at least 16VDC? Is the 12V programming voltage correct?

Web Address & Email

You can email us at **support@quasarelectronics.com** about any problems or requests. Information on other kits in the range is available from our Web page at:

http://www.quasarelectronics.com

PARTS LIST - KIT 3123

Resistors (0.25W carbon)

22R	R11	1
100	R8	1
390	R2,3,4	3
680	R12	1
1K2	R5,6	2
1K5	R15	1
3K3	R7	1
4K7	R9,10,14	3
8K2	R1	1
10K	R13	1
10K SIL resistor network 10 pin	RP1	1

Capacitors

27pF ceramic	C1,2,12,13	4
100nF monobloc	C4,6,14	3
10uF 25V electrolytic	C3,7,8,9,10,11	6
100uF 25V electrolytic	C5	1

Semiconductors

1N4004	D5	1
1N4148	D4	1
LED, 5mm, GREEN	D2,3	2
LED, 5mm, RED	D1,6,7	3
5V6 400mW zener	Z1	1
12V 400mW zener	Z2	1
BC547 transistor, NPN	Q1,2,3	3
BC557 transistor, PNP	Q4	1
AT89C51 Microcontroller, pre-programmed	IC1	1
74HC574 Octal D-Type Flip-Flop	IC2	1
78L05 +5V regulator, TO-92 package	IC3	1
MAX232 Dual RS-232 transmitter/receiver	IC4	1

Miscellaneous

Crystal, 20.2752MHz	Y1	1
Crystal, 4MHz	Y2	1
2.5mm DC jack PCB mounting	X1	1
D9 connector PCB mounting, right-angle, female	X2	1
Pushbutton switch "Zippy"	SW1	1
16-pin IC socket	for IC4	1
20-pin IC socket	for IC2,SKT2***	2
40-pin IC socket	for IC1,SKT1***	2
PCB, 3123		1

***ZIF sockets can be used instead. These are optional items and not supplied with the kit.

COMMAND DESCRIPTION

Commands consist of a single character followed by any parameters where necessary. They can be entered in upper or lower case and are immediate execute (no need to press 'enter').

P   **P**rogram memory
The user will be prompted to "Send file now …". The .HEX file sent to the programmer will be used to program the currently selected memory (Flash or EEPROM) in the device.

Q   Bulk programming (C, P, L3)
The device is automatically erased, programmed and locked. Reduces keystrokes when programming multiple devices.

V   **V**erify memory
The user will be prompted to "Send file now …". The .HEX file sent to the programmer will be used to verify the currently selected memory (Flash or EEPROM) in a programmed device. Verification should be done before writing any lock bits. Devices that have been 'locked' will not verify correctly.

R   **R**ead memory
The contents of the selected memory are sent to the host computer in Intel hex format. Turn on input capture or logging in the terminal program before using this command. The resulting .HEX file will have to be edited with a 'plain text' editor to remove any unwanted command characters or messages.

B   **B**lank check
Check that the device is blank (all locations contain FFh). If the device also has EEPROM then that is checked as well.

C   **C**hip erase
Electrically erase the device. This erases both Flash and EEPROM memory at the same time as well as the lock bits. Fuse bits are not affected.

Ln   write **L**ock bits (**n=1,2 or 3**)
This programs the lock bits. All lower numbered bits will also be programmed eg. L2 will program locks bits 2 and 1. L3 will program all lock bits. See the device data sheets for further details. Lock bits can only be 'unprogrammed' by erasing the device.

Sx   Serial Programming (**E**)nable/(**D**)isable (89S8252/89S53 only)
The 89(L)S8252 and 89(L)S53 devices include a fuse bit that enables or disables Serial Programming Mode of the chip. *These devices are shipped with the Serial programming Mode enabled*. Serial programming should be disabled if maximum system security is required. Refer to data sheets for further information.

F   address **F**lash memory
E   address **E**eprom memory
These two commands are used to select between the program Flash and data EEPROM of devices that have EEPROM. The prompt will change to indicate which address space is selected. EEPROM addressing can be selected for a device that does not have EEPROM. The programmer will only report the error when a memory access command ('P, V, R or B'), is entered.

M   **M**anual device selection
By default, the programmer reads the signature bytes to determine the device type. Sometimes the signature bytes can become corrupted but the device is still usable. However the programmer cannot identify the device and will not program it. This command allows the user to manually select the device type so that the chip can still be programmed. Where required, the user will also be asked for the programming voltage level. The command prompt will change to show the device selected.

A   **A**utomatic device selection (default)
The programmer determines the device type by reading the signature bytes.

H   **H**elp (prints menu)

QUASAR PROJECT KIT # 3123 - ATMEL 89xxxx PROGRAMMER

How to Use 'term.exe' Communications Program

Some kits (3121, 3122, 3123, 3164) 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.

A DOS-based comms program has written called 'term.exe'.
It is a simple, basic terminal program which does its job without a lot of 'bells & whistles'.

You may download it from   **http://www.quasarelectronics.com/software/term.zip**

How to install 'term.exe'

This is a detailed explanation of how we have installed 'term.exe' in our 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
    (We did this because sometimes I use Hyperterminal and all our 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.

QUASAR PROJECT KIT # 3123 - ATMEL 89xxxx 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 3164, 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 3164.) When logging is on note 'LOG' appears on the bottom line.

Programming with Kits 3121, 3122, 3123

We 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 3123 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

If you have suggestions on how to improve this documentation email us at **support@quasarelectronics.com**

August 9, 2001.

- - - - - - - - - - - - - - - -

Title: ATMEL 89C5x/89Cx051 PROGRAMMER
Size: A3 Number: K123 Revision: 2
Date: 28-SEP-2001 Sheet 1 of 1
File: K123/1 Drawn By: FC