

# DIY KIT 121. ATMEL 89Cx051 PROGRAMMER

Microcontrollers (uC's) have been around for quite a while now. With relatively few I/O lines, a limited instruction set and almost no peripherals they are far better suited for control applications than uC's such as 8051, 6800, etc.

This kit is a simple but powerful programmer for the Atmel 89C1051, 89C1051U, 89C2051 & 89C4051 microcontrollers and uses a serial port for communication. Data Sheets may be obtained from their website [atmel.com](http://atmel.com). Go to **Products/8051-Architecture**.

Using the serial port has several advantages:

1. The programmer does **not** require special software other than a communications program. Windows 3.11 & 95/98 come with **Terminal & Hyperterminal** built-in. 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.**

2. It allows the programmer to be used with **any** computer and operating system.

The Atmel devices have a number of advantages:

- Flash technology. No need for EPROMs, UV erasers and OTPs.
- software compatible with the industry standard 8051. The 8051 family has been around for years and a vast library of software is available such as assemblers and simulators. Even code fragments for doing things such as initializing the serial port or BCD arithmetic are also available.

The kit is constructed on a double-sided, through hole plated printed circuit board (PCB) measuring 85mm x 69mm (3.3" 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 center positive. Most 12VDC adaptors should supply this quite easily. Current capacity should be at least 150mA.

## ASSEMBLY INSTRUCTIONS

We have supplied a 20 pin IC socket for small volume programming. For volume programming the user should buy a 20 pin ZIF socket. 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 diodes, **5V6** marked on body
- 12V zener, **12** or **12V** marked on body
- BC547 and BC548 transistor
- 78L05 regulator

For the two zener diodes there is brand information as well as the 5V6 and 12V or 12. 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 & LEDs
4. Transistors and 5V regulator
5. 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.
6. Crystal, pushbutton switch and DC jack
7. The switch has a flat side which lines up with the "bar" on the component overlay.
8. DB9 connector and the user-supplied 20-pin ZIF socket if supplied
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, Q2 and Q3 are used to control the programming voltage. Q3 switches the programming voltage on or off and is controlled via Q1. A high on the base of Q1 pulls the base of Q3 low and the programming voltage is switched off.

Transistor Q2 controls the programming voltage level. With the programming voltage enabled (Q1 base low), a high on the base of Q2 causes zener Z1 to conduct and 5.6V is applied to the base of Q3, giving a 5V programming voltage. A low on the base of Q2 causes zener Z2 and diode D2 to conduct and 12.6V is applied to the base of Q3. The programming voltage is now 12V. Resistor R2 limits the current supplied to the programming pin. With a 12V programming voltage zener Z3 conducts and the "Vpp On" LED lights.

Transistor Q4 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 89Cx051 chips, including setting the lock 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 I/O 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 are used to communicate with the 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 12V DC center positive. The "VPP" LED should come on. Check the following:

1. +5V rail - between pins 10 and 20 of IC1 socket.

# DIY KIT 121. ATMEL 89Cx051 PROGRAMMER

2. VPP - approx 12V on pin 1 of the prog. socket.
3. VCC voltage - 0V on pin 20 of the programming socket.
4. Insert a wire link between pins 5 and 16 of the IC2 socket. The VPP voltage should now be 5V.
5. Move the wire link to pins 6 and 16. The VPP voltage should now be 0V.
6. Move the wire link to pins 7 and 8. The "VCC" LED should come on and the voltage on pin 20 of the ZIF socket should be 5V.

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

## OPERATION & USE

Connect the programmer to the serial port of a PC or other host using a "straight through" 9 pin cable. Start a terminal emulation program (telix, procomm etc) running at **9600 baud, 8 data bits, 1 stop bit and no parity bit**

For Windows3.11 use Terminal. In Windows 95/98 you can use Hyperterminal. Use **TTY emulation, no flow control**. Save As K121.ht. Apply power to the programmer. Press Reset if necessary. A menu should appear and both LEDs should be off. The programmer is ready for use. If the menu does not appear check the above specifications. Check that the com port is correct.

Or use 'term.exe' as described below.

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

The programmer has six command functions:

- P ..... Program code data
- R ..... Read code data
- V ..... Verify code data
- L ..... write Lock bits
- E ..... Erase chip
- B ..... Blank check

Insert a chip into the programming socket and choose an option. The programmer first attempts to identify the device and prints an error message if unable to do so. If all is well the function will be carried out and a message printed on completion. Before programming a chip it should be blank checked and erased if necessary. When programming or verifying data 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 or text transfer protocol. If an error occurs while programming or verifying, an error message will be printed and the programmer will stop. The VCC LED will remain on. The user will be prompted to press the reset switch. The usual programming sequence will be E B P V L. In Hyperterminal use Transfer/Send Text File after P tells you to Send file. (The text file in this case will be a \*.hex file.) VPP will flicker on/off during programming. When you Send file for Verification VPP will not flicker. If all is OK both LEDs will go off. L)ock the code.

**Note:** The "verify" function must be performed BEFORE writing the lock bits. Writing the lock bits prevents the code from being read out. All data will read as 0FFh and verification will fail. When reading data the programmer converts the data into an Intel HEX file before sending it to the PC. The user should turn on text capture before starting the read function so that the data can be saved to a disk file.

## IF IT DOES NOT WORK

Poor soldering and components in the wrong position make up over 95% of problems.

## Contact the Developer

You can email the developer of this kit, Frank Crivelli at

**frank@ozitronics.com**

For a good Atmel web site see

**[http://come.to/Stelios\\_Cellar](http://come.to/Stelios_Cellar)**

## PARTS LIST - KIT 121

### Resistors (0.25W carbon, 5%)

100 brown black brown .....	R2,11 .....	2
680 blue grey brown .....	R10 .....	1
1K2 brown red red .....	R3,6,12 .....	3
3K3 orange orange red.....	R4 .....	1
4K7 yellow violet red.....	R5,7,9 .....	3
8K2 grey red red.....	R1 .....	1
10K brown black orange.....	R8 .....	1
10K SIL resistor network .....	RP1 .....	1
9 or 10 pin network is OK. 'A' type.		

### 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
1N4004.....	D1 ( <b>1N4004</b> on body) ...	1
1N4148.....	D2 ( <b>1N4148</b> on body) ...	1
LED, 5mm, red.....	D3,4 .....	2
5V6 400mW zener.....	Z1,3 ( <b>5V6</b> on body).....	2
12V 400mW zener.....	Z2 ( <b>12</b> or <b>12V</b> on body) ..	1
BC547 transistor, NPN .....	Q1,2,3 .....	3
BC557 transistor, PNP.....	Q4 .....	1
AT89C2051-24PC.....	IC1 pre-programmed.....	1
74HC259 8-bit latch.....	IC2 .....	1
MAX232 or similar brand .....	IC3 .....	1
Dual RS-232 transmitter/receiver		
78L05 5V regulator.....	IC4 .....	1
Crystal, 20.2752MHz.....	Y1 (20.48MHz also ok) ...	1
DB9 right angled, female connector	X1 .....	1
Pushbutton hatkey switch ...	SW1 .....	1
DC power jack .....	X2 .....	1
16-pin IC socket .....	for IC2,3 .....	2
20-pin IC socket .....	IC1, programming .....	2
PCB, K121 .....		1

# DIY KIT 121. ATMEL 89Cx051 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.

### **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.

# DIY KIT 121. ATMEL 89Cx051 PROGRAMMER

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.

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](mailto:frank@ozitronics.com)

If you have suggestions on how to improve this documentation email me at [peter@kitsrus.com](mailto:peter@kitsrus.com)

August 9, 2001.

-----

