GRANTRONICS PTY LTD

GP-AT Programmer
for
Atmel Flash CPUs

GRANTRONICS PTY LTD

ACN 070 123 643

45 Monash Street

PO Box 275

Wentworthville. NSW. 2145

Australia

Phone: (02)9896-7150

Fax: (02)9631-1236

Internet: http://grantronics.com.au

Introduction

The Grantronics model GP-AT Programmer is designed to program the FLASH program memory of the Atmel AT89C51 family (including the AT89C1051/2051) and the AT90Sxxxx ("AVR") family of microcontrollers. It can also program many 8-pin EEPROMS (I^2C and MicroWire).

Features

Simple to operate in basic form
Comprehensive command set for more sophisticated applications
Programs the 20 pin AT89C1051 and AT89C2051
Programs the 40/44 pin AT89C, AT89S, AT89LV families
Programs the AT90S AVRs (20 pin - serial and parallel, 40/44 pin - serial only)
Programs many 8 pin EEPROMS (SPI, Microwire and I^2C)
ZIF sockets for both DIP and PLCC packages included (SOIC 20 via an adaptor)
SPI connector for in-system programming CPUs that support SPI programming
9 pin RS-232 serial interface connector for direct connection to a PC
Powered by external PlugPak (included in the Australian model)
May be automated by simple DOS commands in batch files
CPU watchdog ensures predictable operation during a power failure
4 LEDs display programmer status
PC software required: any terminal emulator that can send an ASCII text file
Programmer uses an AT89C52 CPU - software is easily upgraded. Software upgrades are free (freight charges extra) for the first 6 months.

Specifications

Dimensions:	102mm x 102mm (4" x 4") unenclosed PCB with rubber feet.
Power Supply:	12Vdc 150mA "unregulated" PlugPak (14Vdc minimum).
Ambient Temperature:	0^oC to 50^oC
Humidity:	5% to 95% (non-condensing)
Connectors:	RS-232: 9 pin 'D' female
	SPI: RJ11 (6 pin modular telephone socket).
	Power: 2.1mm DC connector (centre positive)
Data speed:	9600 bps.
Data format:	8 bits, no parity, 1 stop, hardware flow control (to GP-AT).
IC Sockets:	40 pin DIP universal (0.3" and 0.6" wide)
	44 pin PLCC
	A low cost adaptor is available for the 20 pin SOIC devices
Programming Time:	Depends on device, about 2ms per byte for CPUs

NOTE 1: Specifications may be improved at any time without notice. Some features may not yet be available. Please see Appendix A at the end of this manual.

NOTE 2: **The 44-pin PLCC socket is not 'keyed'.** To prevent incorrect insertion of a device in this socket, please ensure that the angled corner of the IC corresponds with the angled corner of the IC socket. Also, **pin 1 should be closest to the 44-pin socket.**

NOTE 3: For all DIP devices, pin 1 of the IC goes to pin 1 of the socket.

Connectors

J1: RS-232 data to/from PC

J1 provides a “4 wire” RS-232 PC interface

Pin	Description
1, 4, 6	CD, DTR, DSR (linked in programmer, not used)
2	TXD (data to PC)
3	RXD (data from PC)
5	GND
7	CTS (not used)
8	RTS (for hardware flow control, only used with slower devices such as small EEPROMs)

J2: Power

Centre pin is positive.

J3: In System Programming cable

Pin	Description	Conductor colour
1	GND	Blue
2	MOSI (Master Out Slave In)	Yellow
3	MISO (Master In Slave Out)	Green
4	SCK (clock)	Red
5	RESET	Black
6	GND	White

Notes: Pin 1 is closest to connector J2 (power). Conductor colours are for our standard SPI cable. Pins 2-5 have 4k7 resistors to “pullup common” (see below).

JP1: In System Programming Options (Revision 1 PCB)

This 10 pin header allows selection of options for J3.

GND	J3 pin 6	Switched +5V	Pullup common	Switched +5V
	+5V		+5V	

Defaults: J3 pin 6 to GND, Pullup common to +5V.

Commands

Commands may be issued interactively from a 'terminal program' such as Windows Terminal, Windows 95 HyperTerminal or DOS Telix, ProCom etc. Alternatively, commands may be sent from simple batch files allowing some simple automation to be achieved.

Commands are presented in what might be "typical usage" order rather than alphabetically.

H **H**elp screen (command summary)

DS **D**evice **S**elect
Selects a particular device type. Use the full part number after manufacturer's prefix, eg, 89C2051, 24C16. Use **DD** to list currently available types.

DD **D**evice Table **D**isplay
Current device table is displayed. The information displayed depends on the device type but will usually include the device memory capacity and signature codes.

DA **D**evice **A**dd
Allows a new device to be added to the device table if the new device uses an available programming algorithm. The syntax is:
DA<desc>,<algorithm>,<FLast>,<ELast>,<Id1>,<Id2>,<Id3>
where
<desc> = device part number (to be used in DS command). 8 characters maximum.
<algorithm> = see the section on Programming Algorithms later in this manual
<FLast> = Flash (program) memory last address (size-1) in hexadecimal.
<ELast> = EEPROM (data) memory last address (size-1) in hexadecimal (0 for none).
<Id1> = first signature byte (hex).
<Id2> = second signature byte (hex).
<Id3> = third signature byte (hex). Use FF if this device does not have a third byte.

NOTES: For 8-pin EEPROMS, set FLast to 0, ELast according to the device. Also, set Id3 to the value that should be read if the device is erased (some erase to 00, other FF). See Device Specific Comments for other uses of the Id fields.

DC **D**evice **C**hange
Allows an existing device table entry to be changed. Syntax is the same as the DA command. Use **DS** to select the table entry to change before using **DC**.

AF **A**ddress **F**lash and
AE **A**ddress **E**EPROM. These two commands are used to select between the program Flash and data EEPROM of devices that have EEPROM (such as the AVRs and the 89S8252). The GP-AT prompt will change to indicate which address space is currently selected. This selection is only required for CPUs - 8 pin EEPROMs automatically select EEPROM but the prompt does not change.

MI Program **M**ode **I**nternal. The device to be programmed should be placed into one of the on-board ZIF sockets.

MS Program **M**ode **S**PI (external). The device to be programmed should be connected via a cable connected to J3. SCK timing is suitable for AVRs using a 500kHz or greater crystal and for the 89S series with a crystal of 3MHz or greater.

MP Program **M**ode, **P**rogram. The .HEX file sent will to the GP-AT will be used to **program** the device.

MV Program **M**ode, **V**erify. The .HEX file sent will to the GP-AT will be used to **verify** a programmed device.

? Display current device and mode selections.

I **I**dentify device. Returns the device signature bytes where applicable.
Example: Signature: 1E 52 FF (consult data sheets for correct values).
8 pin EEPROMs do not support a device signature.

B **B**lank check the device.

E **E**rase the device (electrically).

NOTE: It is not necessary to erase 8 pin EEPROMS before programming.

: Accept data in Intel hex format. This is the “colon” character that heads each hex record. No special programming command is required. Simply send an Intel hex file to the programmer and your device will be programmed. The address from each record is sent to the terminal program as a ‘progress indicator’. Some terminal programs may not be able to display these addresses while sending the hex file to the GP-AT programmer.

R **R**ead device. The contents of the device are sent in Intel hex format. Your terminal program should be placed into a capture mode before using this command. You will have to edit the resulting .HEX file with a “plain ASCII text” editor to remove the command characters. Read may be aborted by sending any character to the programmer (eg space).

Note: 24 bit addresses are supported using the Extended Linear Address Record (hex record type 4) for I^2C devices only. Other devices use 16 bit addresses.

Fsf Program **F**use bits. Programs the AVR fuse bits. s=S for SPIEN, f=F for FSTRT or RCEN (0 to deselect). Example: FS0 to enable SPI programming and disable the RC oscillator (AT90S1200). For the **AT17C512/010** EEPROMs, F0 will program active low reset, F1 will program active high reset.

Lx Program **L**ock bits. Used to program lock bits 1, 2 and 3. All lower numbered bits will also be programmed. Example: L2 will program lock bits 2 and 1. See device data sheets - some devices do not support 3 lock bits.

RF **R**ead **F**use and lock bits on devices that support this function.

S **S**tatus of last command. Returns the result of the last operation. Also resets a fail condition. Device programming is not possible while a fail condition exists. You will probably have to wait for your terminal program to stop sending data to the GP-AT before you can use the ‘S’ command.

The following commands are for use with the 93Sxx/93CSxx series of Microwire EEPROMs.

PR **P**rotect register **R**ead. Returns the current value in the protection register.

PC **P**rotect register **C**lear. Removes all device protection.

PWhh **P**rotect register **W**rite. Writes the hexadecimal address value 'hh' to the protection register. Note that this is the equivalent byte address, not a word address. This address will be converted to the corresponding word address. Odd addresses will be "rounded" down.

Note: If the Proect Register Disable command has been used (not available with the GP-AT), the previous 2 commands will not function. You may be able to erase part of the device.

Command Prompt

The GP-AT command prompt provides indication of some options. The default is
IPF>
The first letter will be I for **I**nternal (parallel) or S for **S**PI (external serial) programming.
The next letter will be P for **P**rogram mode or V for **V**erify mode.
The third letter will be F for **F**lash memory or E for **E**EPROM memory (CPUs only).

Programming Algorithms

Currently implemented programming algorithms for use in the 'DA' and 'DC' commands are as follows:

1. 89C1051/2051
2. 89C51/52/55 including LV types
3. I^2C 8-pin EEPROMs (24C01 to 24C256)
4. Microwire 8-pin EEPROMs (93C46 style, older types requiring constant clocking not supported). See algorithm 12 below for 93Sxx/93CSxx
7. Atmel AT17Cxxx Configurator EEPROMs (not complete for 17C256 and smaller)
8. 90S1200/2313
9. 90S4414/8515 (external serial only)
A. 89S8252 and 89S53 including LS types
B. Xicor X2444 and X24C44 (X24C45 should also work but not verified)
C. 93Sxx/93CSxx Microwire EEPROMs

Device Specific Comments

24Cxx and AT17Cxxx series EEPROMs

To reduce the programming time, the GP-AT uses the 'page write' feature of these devices. A 'page' (device dependant) will be written in a single write cycle when possible. Consequently, no verify after write is performed. Set Id1 to the device page size. Some manufacturers use a different size page - check your data sheets! If the page size is set too big, programming errors will occur but no error will be signalled.

AT17Cxxx EEPROMS

Reset polarity is set to active high or active low by using the F1 or F0 commands. Use the RF command to read the current reset polarity.

93Cxx, 93Sxx and 93CSxx series Microwire EEPROMs

No verify after write is performed.
Id1 is a bit-field controlling various options. Set Id1 according to the following list:

bit 7 = the level to connect to pin 7
bit 6 = the level to connect to pin 6
bit 5 = hex file byte order (16-bit mode). 0=low/high, 1=high/low.
bit 4 = word size (0 = 8 bits, 1 = 16 bits), usually 1.
bits 3..0 = number of address bits (not always related to device capacity!)

The GP-AT always handles data in 8-bit bytes. Some 93C devices may be programmed in 8-bit bytes (these usually have an ORG pin). Others must be programmed in 16-bit words and the data sheets usually specify the address accordingly. Therefore, you should specify the address bit count according to the equivalent 8-bit byte capacity. Example, the 93C56 is 16-bit only and uses 8 address bits (A7..A0). Set the address bit count to 9.

93Sxx and 93CSxx series Microwire EEPROMs

The Protect Register Disable feature is not supported.
Erase is implemented as writes to each location. Locations which are protected will not be erased.

AT90Sxxxx (AVR)

If lock bit 2 is programmed, you will not be able to change the fuse bits until the device is erased.

AT89S8252

The EEPROM exists as an extension to the Flash memory in parallel mode but as a separate address space in serial mode. The GP-AT treats the EEPROM as a separate space in both modes which means that the .HEX files will be the same for both situations.
Experience suggests that P1.7 (SCK) should be high before reset is asserted to enter programing mode.

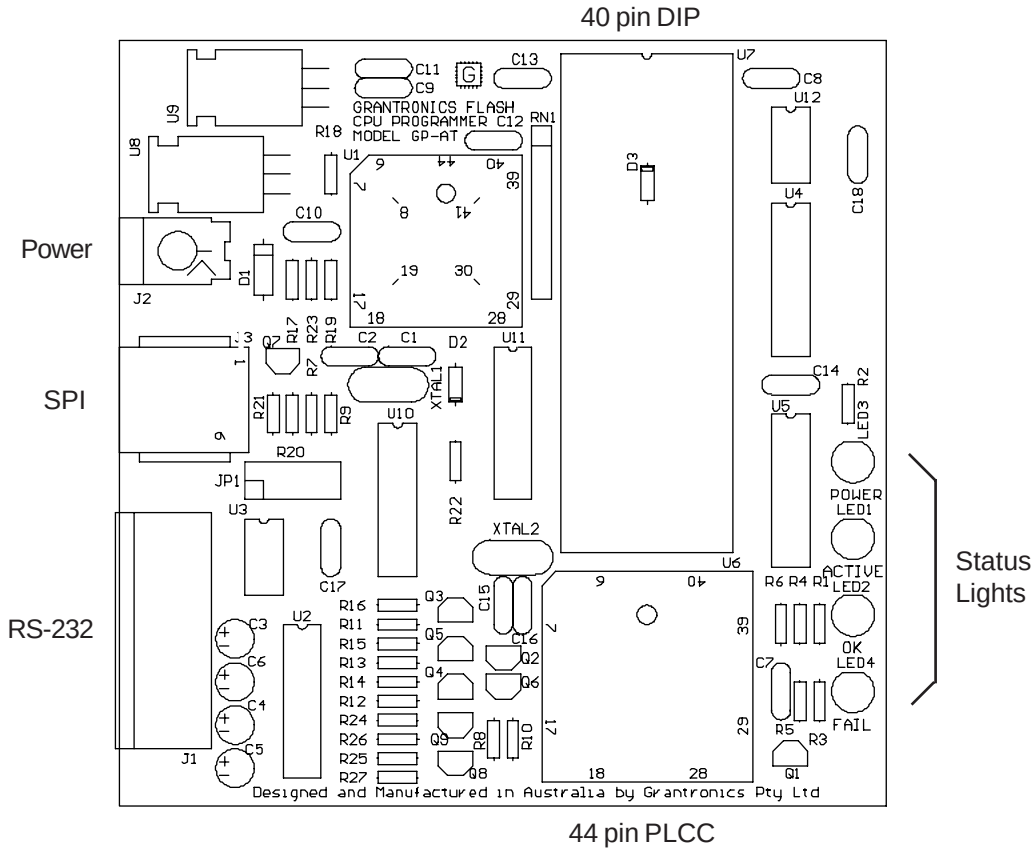
X24C44

Id1 is a bit field as for Microwire devices except only bit 5 (byte order) is used.

Component Placement Detail

The following diagram (component overlay) shows component placement for the GP-AT programmer. This diagram is approximately to scale. The PCB is 102mm x 102mm.

8-pin and 20-pin DIP packaged devices should be place in the 40 pin DIP socket with pin 1 at the top end of the socket (“pin 1 in pin 1”).



More will be added soon.

Software support and upgrade policy

As more devices become available, their programming algorithms will be added to the software of the GP-AT. If a particular device that you wish to use is not currently supported by the GP-AT, we may be able to add a new algorithm to the software - especially if you are able to supply a sample device for verification.

We will upgrade the GP-AT software free of charge (freight charges extra) for 6 months from the date of purchase of the programmer. The following conditions apply:

1. The programmer must be returned to our office packed in suitable anti-static packaging (original packaging prefered).
2. Payment for return freight must be included.
3. Original purchase details must be included (date, supplier, invoice number).
4. You must have sent us your registration details (may be included at upgrade time).

Alternatively, a replacement CPU with upgraded software may be ordered. Please call for details. Note that you will require a suitable PLCC extraction tool for this option.

After the 6 month period has elapsed, a small upgrade fee will apply. Please call for details before ordering your upgrade.

Appendix A

At the time the GP-AT was announced, we expected that all the originally specified features would be ready by the planned release date. Unfortunately, development has not progressed as quickly as was expected! So, we are reluctantly releasing the programmer with a small number of features not complete.

Features or devices not available in this release include:

SPI EEPROMs (8 pin)
Gate array configurator EEPROMs
AVR devices (40/44 pin) in internal parallel mode (external SPI is Ok)

If a feature that you need is missing either because it is on the list above or not in the main specifications, please let us know and we will endeavour to have the feature added in the next release.

Functional changes in this and recent releases:

17 August 1998
SPI connector details changed to agree with PCB screen print overlay. No actual pin functions changed.

5 September 1998
Rev 1 PCB released. Includes pullup resistor options for J3 and extra hardware for 40/44 pin AVRs.
Software now V2.0. Still needs mods for 40/44 pin AVRs in parallel mode.
Read operations may now be aborted by sending any character to the programmer.
Operation with the 89S8252 (serial mode) is now more reliable.
The photo on the front cover is of the rev 0 PCB.

11 October 1998
Front cover photo is Rev 1 PCB.
Fixed bug in 24Cxx and 93Cxx erase (invalid function, introduced in last rev).
Software V2.1.
Added Xicor X24C44.
Resnet on CPU changed from 10k to 3k3 to improve rise times.

23 November 1998
Added comment about P1.7 (SCK) for the AT89S8252.

21 December 1998
Added note about DIP packages with less than 40 pins.

28 December 1998
Software V2.2 includes AT17C512/010 and 93Sxx/93CSxx.
I^2C devices with a 24 bit address are supported.
Id3 byte in device table used to set the value of an erased byte for EEPROMs.