PowerLab Universal Programmer User's Manual

Release Notes

Version 1.1 May 14th, 2003

Content

Disclaimer	2
Technical Support	3
1. Introduction 1.1 About this manual 1.2 Product description 1.3 Package content	4
2. Hardware Overview 2.1 PowerLab programmer 2.2 The Converters	5
 3. Setting Up the Hardware 3.1 Check your computer system 3.2 Setting up the hardware 	6
4. Installing the Software	7
 5. Tutorial for a Quick Start 5.1 Before you start 5.2 Programming Philips P89C51UBP in DIL40 package with PowerLab A. Executing PowerLab user interface software B. Setting up communication with PC C. Configuring PowerLab system to serve your preference D. Selecting the device you want to program E. Download the data into PowerLab programmer F. Configure the process flow and related programming parameters Process Flow Address relocation Special bit editor Programming parameters Serial code programming 	8

- G. Program your device
- H. Save all your work to a Project

Disclaimer

System General has made every effort to ensure the integrity of the software and the hardware systems of our IC programmers. However, System General disclaims any liability for any errors, damages or financial losses that are incurred directly or indirectly from the use of System General IC programmers.

System General reserves the right to modify or to upgrade the programming systems at any time without prior notices to the users. The customers are recommended to refer to the *ReleaseNote.txt* file attached to the software or to contact System General for the details.

Technical Support

It is often the case that users experience problems when installing or using a product for the first time. PowerLab comes with a user-friendly Windows software interface to make it an easy and comfortable learning task.

If you have questions or run into any problems, please consult the following list for help. As with all other System General programmers, software download for new device supports are available on the web.

1. This Manual

2. On-Line Help

Press <F1> for help any time after activating the software.

3. Internet Web Site

You can access either <u>http://www.sg.com.tw</u> (server in Taiwan) or <u>http://www.systemgeneral.com</u> (server in U.S.) to download the latest version of software, supported chip list, and many other useful information. PowerLab software contains demo mode which can run all by itself for practice or evaluation purpose.

4. Your local distributor

Check out our web site to find the distributor nearest to you.

5. System General E-mail and FAX

If all the above cannot resolve your problems, you can contact System General offices at

System General Corp. (Taiwan)

5F, No. 9, Alley 6, Lane 45 Bao-Shing Road, Shin-Dian Taipei, Taiwan Tel: +886-2-2917-3005 Fax: +886-2-2911-1283 E-mail: sales@sg.com.tw

System General Corp. (U.S.A)

1623 South Main Street Milpitas, CA 95035 U.S.A. Tel: 408-263-6667 Fax: 408-262-9220 E-mail: sales@systemgeneral.com

We welcome feedback or comments from you to improve our products and service quality.

1.0 Introduction

Thank you for choosing System General PowerLab universal programmer. If you run into any difficulties using this machine, you can consult the followings for help (1) this manual (2) on-line help (3) your local distributor. We are making every effort possible to make it a comfortable and rewarding experience using PowerLab.

1.1 About This Manual

This manual is written to serve as a handy guide for you to get started quickly and acquainted with PowerLab programming system. The software that comes with PowerLab contains comprehensive on-line help topics to lead you step by step toward operational details you may want to know. We suggest that you read through this manual before using the programmer but resort to the abundant help resources embedded in the software in case of difficulties.

1.2 Product Description

PowerLab is a high speed universal device programming system with USB communication interface. This programmer was designed for engineering application as well as small scale production through its continuous mode operation. It is equipped with 48 equally powerful and universal pin drivers along with its own CPU and DRAM resources.

PowerLab programmer needs to be controlled by PC software interface for operation. It can be hooked up to with a PC through RS-232 serial port or USB communication interface. The devices supported on this programming system include EPROM, EEPROM, Flash EPROM, Micro-controller, CPLD, EPLD and many other devices. PowerLab is also capable of supporting the green device technology which requires 1.2 volt or lower VCC voltage. We will keep on adding new device programming supports based on the market's and the customers' needs.

PowerLab deals with different device packaging technology by the use of various "converters". The device selection function in the software will provide all information you need in terms of how to program a particular device and what converters need to be used with. There are also a lot of functions and stand-alone utilities contained in PowerLab software which you will find handy and you are welcomed to tell us what kinds of functions you need.

All the new device drivers will be made available on the web site free for download. You are recommended to register yourself through System General's web site to become registered users with us and enjoy the resources provided over the Internet.

1.3 Package Content

When you unpack PowerLab box, you should find the followings:

(1) PowerLab Main Unit	1
(2) Power Cable	1
(3) $100V \sim 240V$ AC/DC Adapter	1
(4) USB Cable	1
(5) System Software CD	1
(6) Customer Registration Form	1

2.0 Hardware Overview

2.1 PowerLab Programmer



2.2 The Converters



The converters are designed to deal with the variations of the device packages. The converter is normally comprised of a Socket and a Printed Circuit Board (PCB), as shown in the picture on the left. Users would insert the device to be programmed into the socket and make sure the contact in between is good by feeling or exercising pin continuity check function of PowerLab. When a particular device with some specific package type is selected by users, the software would display on the same line the name of the correct converter to be used with, or, in some cases, no converters are needed if the devices are in Dual-In-Line packages. Converters for PowerLab are designed to be hot-swappable meaning one can replace on the fly one inserted converters with another as long as PowerLab is

not in the middle of any non-interruptable operation such as programming.

Most of PowerLab converters below and up to 48 pins are designed to be universal ones thanks to the powerful pin drivers in PowerLab. One converter built of a particular socket type will serve practically all the devices that could fit themselves in this socket. In some packages types such as TSSOP or SOIC, although not recommended, there are good chances that devices in a lower pin count could fit in the converters in the same package category but for higher pin count. When in operation, PowerLab will check if the correct converter is in place for the selected device and will alarm you by popping up a warning message if not.

Note:

Make sure there is NO converter or device gets installed on PowerLab programmer right before you turn on the power of the system. Otherwise it may cause the Self-Test program to fail on the system.

3.0 Setting Up the Hardware

3.1 Check Your Computer System

You need to connect PowerLab to a PC with the minimal hardware and software requirements as below:

PC Pentium II or better. Microsoft Windows 95/98/ME/2000/XP (Windows NT not available yet) CD-ROM RS232 or USB port 64 MB DRAM 32 MB free disk space.

Note:

The USB driver of Windows NT for PowerLab is not available yet.

3.2 Setting Up The Hardware

(1) Connect the PC and PowerLab programmer with RS232 serial cable or USB cable.

Depending on your PC configuration and operation system, you may choose to use either serial port or USB interface to make connections between PC and PowerLab.

RS-232 Serial Port Connection: Connect your PC and PowerLab with the RS232 serial interface cable.

USB port connection (Windows 95/98/ME/2000/XP): Start by taking the USB cable shipped with PowerLab to plug in the USB port on your PC. Watch out for the two different connector types on the ends of the USB cable. Choose the right connector that matches the USB port on your PC. Plug the connector on the other end of the cable into the USB port on the side panel of PowerLab.

- (2) Plug in the DC jack of the AC/DC adapter (included) onto the DC power input of PowerLab, which is located right beside the On/Off switch on the side panel of the machine.
- (3) Take one end the power cord (included) to plug into the AC input of the AC/DC adapter and the other end into the power socket (outlet) on the wall.

Depending on the country you are in, you may need an adapter to change the male plug into the type that matches the power socket on the wall.

(4) Turn on the power of PowerLab (Make sure there is no converter or devices on PowerLab !)

Turn on the power of the PC firstly, followed by that of PowerLab. Caution should be exercised to make sure that there is no converters nor any device on PowerLab. The LED's on PowerLab will flash in oranges and greens for about 5 seconds then settle down with green in ON and PASS LED's.

Note:

If this is the first time ever you hook up PowerLab to your PC through USB interface, after you turn PowerLab on, Windows operating system will automatically detect this new piece of hardware and prompt you for the driver. Simply ignore it by clicking on Cancel button for now because the USB driver will automatically be installed later when you perform software installation.

4.0 Installing the Software

4.1 Install The Software

Follow the procedures below to install PowerLab control software on your PC:

Windows 2000/XP users:

If you have Windows 2000 or XP installed on your PC. It is required that you login as Administrator on your PC to install the software.

Step 1: Insert PowerLab software CD in your PC's CD-ROM.

Step 2: Double-click the CD-ROM icon to display the content of the CD on your PC display.

Step 3: Point your mouse on Setup.exe file and double-click to initiate the software installation process.

Step 4: Follow the step-by-step instructions on the PC display until the installation is finished.

<u>TIP</u>: Installation procedure will install not only PowerLab software but also a lot of utilities such as Device Help, Converter User's Guide, File Transformation utility, Serial Code Template Generator, On-line help and Release notes.

Step 5: Restart your computer

Reminder: (For users of software before version 1.1)

If you are installing PowerLab software before version 1.1, it is mandatory that for the first time right after you install the software to run the Set up PowerLab USB Driver utility program to properly install PowerLab USB driver on your PC and do remember to restart your PC after this. Starting version 1.1, this operation is automatically taken care of by the software behind the scene. Please download the latest software from web!

Step 6: Click on the *Start* button at the lower-left corner of your screen again, point your mouse to **Programs** | **PowerLab** in that sequence to display respective side menus and finally click on **PowerLab** to launch the program. You may want to establish a link to this program on your PC desktop for easy access in the future.

Caution: The software may perform a firmware update automatically

If this is the first time you invoke PowerLab software after installation, the software will check the firmware version in PowerLab to see if it is the latest one. If the currently invoked software contains a newer version of firmware than the one existing in connected PowerLab, the software will automatically update the firmware in PowerLab to make it consistent.

Tips for "No USB Device Detected" warning message

Right after you launch the software as prescribed above, you may go ahead hit the "Connect" button on the communication dialog box to initiate the default USB connection. If you run into an error message like "No USB device detected", you may try to resolve this by activating Hardware Manager of Windows and locating an "Unknown USB device" item in the list. Remove this item and restart your PowerLab. If this does not work for you and the problem remains, call your local distributor for assistance.

5.0 Tutorial for A Quick Start

5.1 Before You Start

Make sure that you have gone through the steps described in earlier sections of Setting Up the Hardware and Installing the Software. Check if the RS-232 or USB cables are properly connected between PowerLab and PC. Make sure that there is no device and/or converters inserted in the socket of PowerLab and then turn on the power on your PC followed by that of PowerLab. Wait for about 5 seconds till you see the first two LED's (ON and PASS) settled down in green.

5.2 Programming Philips P89C51UBP in DIL40 package with PowerLab under USB communication.

A. Executing PowerLab User Interface Software

Click on the **Start** button at the lower-left corner of your screen again, point your mouse to **Start** | **Programs** | **PowerLab** in that sequence to display respective side menus and finally click on **PowerLab** to launch the program. You will see the picture as shown below on your PC display:

Communication configuration
Connection made by: • USB • COM1 • COM2 • COM3 • COM4 • COM5 • COM6
Please connect your PowerLab to PC through USB or RS232 port. Then press "Connect" button to start the communication.
Demo ⊘ Chip List Viewer
Lato-connect when program start
Baud Rate © 115200 © 57600 © 38400 © 19200 © 9600

<u>TIPS</u>: *Demo* and *Chip List Viewer* buttons can be activated as stand-alone utilities on your PC without PowerLab physically connected. They are the perfect tools for you to practice the software and get a feeling of how the software works. *Chip List Viewer* becomes handy when one wants to find out if a particular device is supported on PowerLab and, if supported, what type of converter is needed to program that particular device.

B. Setting up Communication with PC

Check the radio box on the top of the Communication Configuration dialog window shown above to tell PowerLab which communication port you choose to connect PowerLab to your PC. In this tutorial, check the radio box marked with "USB". (You may have to specify the baud rate your PC supports if you choose serial port connection. The default baud rate is 115200 bits per second. All done, go ahead click *Connect* button to establish connection between PowerLab and the PC. You will be reminded by a message on the screen to remove the device/converter in the sockets. After that, you are entering the user interface of PowerLab upon completion of successful **Self-Test** and **DRAM Test**, when the following **Console window** appears on your screen.



At the bottom of the console window, there is an area which shows you information about the selected device. As no device selection has been made so far, there is no information in this area. However, after you make selection on the device you want to work with, the related information about the selected device will be available in the area, which makes it easy for you to make reference to.

C. Configuring PowerLab System to Serve Your Preferences.

It may be a good time to take a quick look at the configuration options you may have on PowerLab software system. For instance, you may be interested in saving the operation history into a file. To do this, activate such a logging function by selecting from main menu item of **Configure** | **System Options** to bring you to the following window display: (see below)

Check the box titled **"Save console messages to file (append to \$Installed_base\Log\log.txt)**". This will make the system save and append all the console messages to a file named **log.txt** in one of the subdirectory (named **Log**) under the base directory of PowerLab software installation. Note that this log file will keep growing in size so some time in the near future, you may have to clear it by hitting the button **"Clear Log File"** and the system will create a new one for you from scratch.

In addition to the logging function just described. You can also configure the beeper and how detailed you want the console messages reveal system responses.

"Clear DRAM buffer right after select device operation"

if checked, the system will clear DRAM content every time after user makes selection of device. This was meant to get the system RAM buffer clean to be ready for receiving data download from PC.

"Do not calculate checksum right after select device operation"

If checked, the system will NOT perform checksum calculation out of the data kept in system RAM buffer. This was meant to save some time for users who don't think it is necessary to verify the checksum valued every time a device selection is made.

🔊 Option	
System Options Option Information	System Option Beeper Disable Beeper Beep if Fail Beep Always Message Type Concise Verbose Save console messages to file (append to \$Installed Dir\Log\log.txt)
🗸 ОК	X Cancel

D. Selecting the Device You Want to Program

Click on the menu item **Device** | **Select Device** to enter the following *Select Devices* window:

ᇌ Select Devices				_	
Search :		Package : ALL	•	Device Help Select	by POF
Type : ALL	•	Converter : ALL	-	👯 🖌 🗹 🕺	<u>C</u> ancel
Device Selec	cted: Philips	P89C51UBP			
Atmel	Manufacturer	Part Number	Package	Converter	
Catalyst	Philips	P87C51SBBB	PQFP44	CVT-PQFP44-D48-00	
Lypress	Philips	P87C52SBPN	DIP40		
Huniy	Philips	P87C54SBAA	PLCC44	CVT-PLCC44-D48-00	
ISSI	Philips	P87C54SBBB	PQFP44	CVT-PQFP44-D48-00	
Intel	Philips	P87C54SBPN	DIP40		
Lattice	Philips	P87C58SBAA	PLCC44	CVT-PLCC44-D48-00	
MT MOSEL	Philips	P87C58SBBB	PQFP44	CVT-PQFP44-D48-00	
Macronix	Philips	P89C51RC+IA	PLCC44	CVT-PLCC44-D48-00	
Microchip	Philips	P89C51RC+IN	DIP40		
Micron	Philips	P89C51RD+	DIP40		
Motorola	Philips	P89C51RD+	PQFP44	CVT-PQFP44-D48-00	
National	Philips	P89C51RD+IA	PLCC44	CVT-PLCC44-D48-00	
Philips	Philips	P89C51RD+IN	DIP40		
	Philips	P89C51UBP	DIP40		
IST	Philips	P89C51X2BA	PLCC44	CVT-PLCC44-D48-00	
Samsung	Philips	P89C52BA	PLCC44	CVT-PLCC44-D48-00	
TI	Philips	P89C52BP	DIP40		
Toshiba	Philips	P89C52UBA	PLCC44	CVT-PLCC44-D48-00	
WSI	Philips	P89C52X2BA	PLCC44	CVT-PLCC44-D48-00	
Winbond	Philips	P89C54UBA	PLCC44	CVT-PLCC44-D48-00	- 1
XILINX -	Philins	P89C58RP	DIP40		النے ا
XICOI					

In this tutorial, we want to program Philips P89C51UBP in DIP 40 package. To select this device, one can first narrow the device list content by clicking on "Philips" in the list of vendors on the left-hand side of the window. Then, move your mouse to the *Search* area on the upper-left corner and type "89C51" in it. Note that the searching function is automatically triggered whenever a new character is typed in.

There are alternative ways of narrowing your search. You can try specifying the package type of the device or finding out what devices are supported on a particular PowerLab converter. Play around with the *Package, Converter* and *Type* drop-down menus across the top of the window to find out how they can help you quickly locate the device(s) you are interested in.

The *Device Help* button on the upper right corner of the window does not become active in this example. However, whenever PowerLab needs to advise or warn the users of some special features of the selected device, this button will become active and you are suggested to click on it to find out what you need to know before programming the selected device.

Click on **OK** button to register your selection with PowerLab. After the device selection is made, you will note that the information columns at the bottom of the console window will display the related information about the selected device.

E. Download the Data into PowerLab Programmer

Now PowerLab knows what device you want to work with, we should prepare the data to be programmed into that device. Such data normally resides in your PC so all we have to do is to download the data from the PC into PowerLab RAM buffer. Select from the menu item **File | Download** to pop up the window as shown below:



To specify the data file to be downloaded, provide the path of the file in your PC or use the **Browse** button to search and specify the data file in your system. The format of the selected data file will be automatically identified after you hit the **OK** button. If you want to make sure PowerLab RAM buffer gets cleared before filled with the downloaded data, check the box of "**Pre-fill RAM buffer with the following data**" and provide the content of it (default is "FF" in hexadecimal) in the text area right below it. This is normally recommended practice to assure the checksum value won't get messed up by some unwanted left-over data in PowerLab RAM buffer.

As a matter of fact, you can also fill PowerLab RAM buffer with the intended data for programming by reading a pre-programmed master device first. This is relatively easier and not demonstrated in this example. (Simply insert the master device in PowerLab and perform a **Device** | **Read** function.)

F. Configure the process flow and related programming parameters

Now the device has been selected and source date ready in PowerLab RAM buffer. The next thing we are to be concerned ourselves with is to configure the process flow that the selected device shall be gone through during intended operation as well as to fine-tune the parameters that the intended operation should be conducted with.

In most of the cases, PowerLab will configure a set of default values or settings for you, which should be applicable to the selected device in standard applications. It is only when you intend to do something no so common with the device then you need to change the default settings to others.

PowerLab gives you total flexibilities to configure the followings:

(a) Process Flow (Menu item Configure | Process Flow)

In the process flow configuration window (see below), you can configure the detailed steps / sequence that different operations would work on the selected device. Such operations include *Program, Read, Erase, Verify and Secure*. Any changes you made here will affect the detailed procedures the device will go through when you activate the menu item of **Device** | **Program, Device** | **Read, Device** | **Erase, Device** | **Verify and Device** | **Secure**.

Option		×
System Options Option Information Device Process Flow Program Option Programming Param Address Range Special Bit Editor Information	Process Flow Functions Applied To All Operations Auto Start Insetion Check Device Existence Pin Continuity Check Dheck ID-Code Always Program Read Erase Verify Secure Program Enable Verify: Enabled Verify: Disabled Margin Verify: Disabled Secure: Disabled Test: No support	
•		
🖌 ОК	🗙 Cancel	

At the top of the window, you are allowed to activate some general checking functions which would apply to all the operations (Program, Read, Erase, Verify and Secure) once they got configured here.

Functions Applied To All operations include:

Auto Start:

Automatically start the operation right after the lever on the DIL socket get pressed down

Insertion Check:

There are two types of checking you could ask PowerLab to conduct as the first step after operation being started.

Device Existence: Will only check if the device is there in the socket or not

Pin-Continuity Check:	Will check to ensure 100% connection between the
	device and the socket

Check ID code Always:

In most of the programmable memory devices as well as some micro-controllers and logic devices, There is an unique identification code embedded inside the device, which can be read back by programmer to check if the selected device matches what actually get inserted in the socket of the programmer.

In this example, we only want PowerLab to check the existence of the chip before going into any further operation on Philips P89C51UBP.

(b) Address Range Relocation (Menu item Configure | Address Range)

Option		×I
Option System Options Option Information Device Process Flow Program Option Address Range Special Bit Editor Serial Code Program Information	Address Range Device Start Address : Device End Address : FFF (Hex) RAM Buffer Start Address : 0 (Hex) RAM Buffer End Address : FFF (Hex) RAM Buffer End Address : FFF (Hex)	×
	Cancel	
_ ок	Cancel	

This configuration option would become handy when you want PowerLab to program a specific range of data residing in her RAM buffer, dictated by *RAM Buffer Start Address* and *RAM Buffer End Address*, into an address range on the selected device prescribed by *Device Start Address* and *Device End Address*.

The standard way of setting this configuration is to decide the start and end address on the device which needs to be programmed, followed by providing the start address of PowerLab RAM buffer in which the intended source data resides. The end address of PowerLab RAM buffer is automatically set by the software to match the size of the specified memory range in the device.

In this example, we don't want to do any address relocation when programming Philips 89C51 so we leave the default values without making any modification.

Option	×
Option System Options Option Information Device Process Flow Program Option Programming Param Address Range Special Bit Editor Serial Code Program Information	Special Bit Editor The selected device has the following special options to be se Secure Mode Select Lock Level: Level 1 - LB1 Level 2 - LB2 Level 3 - LB3

(c) Special Bit Editor (Menu item Configure | Special Bit Editor)

Depending on the device selected, Special Bit Editor allows you to configure some characteristics of the device. Different device will have different configurable characteristics which are dictated by the design philosophy of the chip. The typical characteristics Special Bit Editor will be dealing with are **Secure** and **Protect**. In this tutorial, we do not want to turn on any *Secure* function on Philips P89C51UBP so we leave it as it.

(d) Programming Parameters (Menu item Configure | Programming Parameters)

PowerLab gives you the power to adjust any desired parametric values that could have significant effect on the programming result of a device. You are welcomed to conduct such a sophisticated configuration when you are well aware of what you are doing and what is going to be happening.

Vendor-provided limits on each of the parameters will be in effect, preventing you from hurting the device by accident.

For the case we have at hand, we leave the default values untouched for programming P89C51UBP.

Option		×
System Options	Programming Parameter	
		·
Device	Vccr: 500	x1UmV (4/5~525)
Process Flow	Vccp: 500	x10mV (475~525)
🖻 Program Option	Vcce: 500	x10mV (475~525)
Programming Paran	Vccs: 500	x10mV (475~525)
	Hi-Vcc: 525	x10mV (475~525)
Serial Code Program	Lo-Vcc: 475	x10mV (475~525)
Information	Vpp1: 1225	x10mV (1200~1250)
	lpp1: 400	x 500uA (60 ~ 1024)
	Vih: 400	x10mV (350~525)
	Voh: 200	x10mV (200~500)
	Vol: 40	x10mV (0~80)
	tPW: 20	x1uS (20~105)
	tEW: 500	x1ms (450 ~ 550)
· ·		Default
С		X Cancel

(e) Serial Code Programming

Select from the main menu item Device | Serial Code Program Mode to pop up the window as shown below:

ᇌ Choose Your Batch Serial Data Fil	
Create New Batch S	erial Data File
😅 O Use Exists Batch S	erial Data File
File Name	
	🗎 <u>B</u> rowse
🗸 ОК	🗙 Cancel

The design of the procedure to conduct a serial code programming with Powerlab was made as follows. Users are requested firstly to generate a serial data file or the so-called BSD file (Batch Serial Data) which is actually a Text file in which each line or record represents one piece of serial code information intended for one device. In this window, you are requested to create or advise the system of such a data file. Just for demonstration purpose, let's go through the procedure of creating a new serial code data file. Check the radio box of **"Create New Batch Serial Data File"** and click the **OK** button to bring you to the next window display.

🔦 Serial Code Programmi	ng	
Serial Code Information	n Kept in Programmer RAM Buffer	
Start Address:	0	
Size (bytes):	4	÷
Byte Order:	LSB first (Intel, Little Endian)	•
Format:	Binary	•
Serial Data		
Start Serial Num	ber: 1	
C Loop Count:	5	
C End Serial Numb	per: 999	
Increment by	1	_
C Decrement by	<u>l</u> i	
	Create Serial Code Data File	
	0%	
Advanced	itart Serial Code Programming	🗶 Cancel

With this window display, you have to do the followings:

- (a) Fill out the information requested in the Serial Code Programming window, and;
- (b) Press the "Create Serial Code Data File" button to generate a template file (a BSD file) which you don't really need to understand but the system would know how to deal with it, then;
- (c) Press the "Start Serial Code Programming" button to initiate the real operation.

For advanced users, if some very special serial code application is intended, one could resort to a more sophisticated approach of creating such a serial code data file by pressing the **Advanced** button. In the advanced approach, a software utility will be guiding you through a step-by-step detailed procedure of specifying the content of a BSD file and then creating such a file for you based on your specification. Luckily, for most users, it would suffice to fill out the data in the serial code programming window.

We don't really want to do any serial code programming in programming Philips P89C51UBP device, so hit **Cancel** button to bail us out of this utility environment.

G. Program Your Devices

Now Let's go for some real actions. In this example, the Philips P89C51 we chose happened to be in DIL 40 pin package, we don't need any converter for it. However, in general cases, if you want to program devices in packages other than Dual-In-Line, you have to make ready the needed converter first and *this is the time for you to insert the converter or device onto PowerLab programmer*.

Click on the main menu item **Device** to drop down a series of operations you can do now such as **Read, Erase, Program, Verify, Blank Check, Continuous Program Mode and Serial Code Program Mode** etc, to just name a few. In this tutorial, we will now demonstrate the programming operation and leave the rest to your own practices.

Depending on the number of devices you want to program, you can select from the main menu item either

- (a) **Device** | **Program**, if you only want to program one device, or
- (b) Device | Continuous Program mode, if you want to program more than one device.

Let's do this demonstration with the Continuous Program Mode. Below are what you will go through:

First of all, a Special Bit Editor window as shown below will pop up to remind you of setting the applicable and optional features you would like to have with the device.

N Special Bit Editor	
The selected device has the following special options to be set.	
Secure Mode Select Lock Level: Level 1 - LB1 Level 2 - LB2 Level 3 - LB3	
I I I I I I I I I I	

In this example, we don't want any locking function activated for the selected 89C51 device, so we just hit **OK** button to get to the next display. (See below)

Let's look at the left half of the picture shown below, note that the selected device and target checksum value were displayed at the top left corner of the window. The graphical representation of a DIL 48 pin socket was used to illustrate the pin continuity check function in action in a real-time fashion. And down at the bottom of the window the procedure or the process flow the device will go though were also illustrated in graphical icons, based on how you have configured it in the **Configure | Process flow** function.

On the right half of the window, the detailed failure cause statistics were shown. In addition, one can choose to activate the counter function to help keep track of the number of programmed devices, failed or passed. To enable the counter, you have to first specify the target counts of the passed and/or failed devices by pressing the **Set Target** button at the lower right corner of the window. Once the target count is set, click on the **Start Count** button on the upper right corner to trigger the counting function and start to count. A warning message box will pop up if any of the target count limits is reached.

Understanding the above, all you have to do now in this example is to press the **Start** button right under the DIL-48 socket drawing to start programming your device. Replace the programmed device with another blank one when it is done and hit **Start** button again until you are done.

📉 Continuous Program Mode			
Project Name:		Counter	
Device Name: Philips P89C51RD+	Pass :	0	Stop Count
Package: DIP40	Eail:	0	Reset Counter
Checksum: 0x007F8000			
	Failure Cause Statistics		
		Counter P	ercentage
			0.0%
			0.0%
			0.0%
			0.0%
			0.0%
			0.0%
			0.0%
	LVF:		0.0%
	SF:	0	0.0%
	PtF :	0	0.0%
	UPtF :	0	0.0%
	OF :	0	0.0%
	Count to Stop		
Ctot Abort	Pass :	0	Set Target
	Fail ·	0	 Reset Target
	1 Car - 1	V	reserrarger
Inse. Auto ID Erase Blank Prog.	Verify	-Ver L-Ver	Secure
0%			
Ready.			

G. Save All Your Work to a Project File.

If you are happy about the programming configurations you have set up so far, click on the **Project** | **Save as Project** main menu item to save all the selections, configuration and optional settings you have made so far. Next time, when you need to program the same device again with the exact same configuration, you can simply load this project file by **Project** | **Load Project** menu item to automatically configure PowerLab into the exact same configuration we are making now.

This will conclude the tutorial.