Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

Phyton™

ChipProg Device Programmers

User's Guide

ChipProg-48

ChipProg-40

ChipProg-G4

ChipProg-ISP

ChipProg Device Programmers

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: August 2010 in (whereever you are located)

Table of Contents

1 Introduction

Phyton™

ChipProg Device Programmers

User's Guide

ChipProg-48
ChipProg-40
ChipProg-G4
ChipProg-ISP

1.1 Terms and Definitions

Terms used in the document

Target device or **Target**
The device to be programmed by a programmer either in the programmer socket or by an additional adapter or by a cable for in-system programming.

Start and **End Addresses** (of the Target device)
A range of the device physical memory for the programming operations (Read, Write, Verify, etc.).

Device packagePackage
Mechanical characteristics of the target device; ChipProg programmers enable operations on the devices packed in the DIP (DIL) packages with no additional adapters as well as on most non-DIP packed devices, including but not limited to the devices in the PLCC, SOIC, SSOP, TSOP, SSOP, QFP, BGA, QNF and other packages.

Programming socket or **Programming ZIF socket** or **ZIF socket**
A socket installed on a programmer unit or on an adapter (see below) to accommodate the target device for programming. All ChipProg models use ZIF (or Zero Insertion Force) programming sockets that allow for the temporary installation of the target device in the programmer site and easily

removing it after completing the programming procedure.-40, ChipProgChipProg

Adapter or **Package adapter** A small transition board with dual-in-line rows of pins pluggable into the programmer ZIF socket on the bottom side and with a package-specific ZIF socket (TSOP, PLCC, etc.) on the top. The adapters for in-system programming by means of the parallel programmers are implemented as ribbon cables that connect to the target board via a special header. The adapter boards can carry passive components (ZIF sockets, pins and cables) and active components (drivers, latches, transistors, etc.). Hundreds of Phyton brand adapters as well as third party adapters are available to support devices in most types of mechanical packages.

File In the ChipProg context the term **file** may represent: a) an image of information on a PC hard drive or other media that is supposed to be written into the target device's physical memory or b) an image read out from the target device and then stored on the disk or other media. Files in a ChipProg can be loaded from and saved on a PC hard drive or CD.

Buffer or **Memory buffer** A memory segment, physically assigned from the computer operational memory (RAM), for temporarily storing, editing and displaying the data to be physically written to the target device's memory or read out from the device. The program allows opening an unlimited number of buffers of any size while it is not restricted by the computer memory.

Buffer layer or **sub-layer** A buffer may have a few layers (in some topics also known as sub-layers) that are defined by a particular architecture and memory model of the target device. For example, for some microcontrollers one buffer can include the code and data memory layers (see more details below).

Buffer size The buffers may have different sizes from 128KB to 32GB each.

Buffer start address The address to display the buffer contents from.

Checksum An arithmetic sum of the data located within a specified part of the buffer calculated by the programmer to control the data integrity. The program enables different algorithms for the checksum calculation and enables writing the checksum into a specified location of the target device.

Parallel or **In-socket programming** Operations on a device being placed into the programmer's ZIF socket or into a programming adapter (opposite to the in-system programming below).

ICP or **in-circuit programming** Programming devices mounted on the boards (in the user's equipment) via special adapter-cable connecting the programmer to the target.

ISP or **in-circuit programming** Same as above. Programming devices mounted on the boards (in the user's equipment) via special adapter-cable connecting the programmer with the target.

ISP Mode Mode of the in-system programming that is usually defined by the programming signals voltage or the ISP interface (JTAG, UART, SPI, etc.). Distinct ISP modes are enabled for different target devices and more than one mode may exist for one device.

ISP JTAG Mode In-system programming via a JTAG interface.

ISP HV Mode In-system programming that requires applying a relatively high voltage to the target device, (12V for example).

Project An integrated set of information in the ChipProgUSB that completely describes the target device, properties of the data buffers, programming options and settings, list of the source and destination files with all their properties, etc.. Each project that has its own unique name can be stored and promptly reloaded for immediate execution. Usually a user creates a project to work with one type of device. Working with projects saves a lot of time for the initial configuration of the programmer every time you start working with a new device.

File - Buffer - Target structure

Buffers are intermediate layers between the data in files and the data in the target device. The ChipProg enables no direct interaction between the files and target devices. All the file operations, such as loading and saving files are applicable to the buffers only. All the physical manipulations with the target device memory content pass through the buffers as well. This is a fundamental principle of the programmer operations with data and devices

Examples of the buffer's layer structures of different devices:

1. In the Intel 87C51FA microcontroller each opened buffer includes two layers: **Code** and **Encryption table**.

2. In the Microchip PIC16F84 microcontroller each opened buffer includes three layers: **Code, Data EEPROMIdentifier locations**.

Each buffer layer can be opened for watching or editing by clicking its tab on the top of the buffer window.

1.2 System Requirements

To run ChipProgUSB and to control a ChipProg programmer, you need an IBM PC-compatible computer with the following components:

- Pentium-III CPU or higher
- Windows XP/Vista/7 for the ChipProg-48, ChipProg-40 and ChipProg-ISP programmers in the all modes
- Windows 98/ME for the ChipProg-48, ChipProg-40 and ChipProg-ISP programmers in the basic mode
- Windows 2000/XP/Vista/7 for the ChipProg-G4 programmer
- Windows XP/Vista/7 for using the Application Control Interface control
- 256MB of RAM
- At least one USB port
- A hard drive with at least 200MB of free space

2 ChipProg Family Brief Description

ChipProg is a family of device programmers produced by Phyton, Inc. Microsystems and Development Tools (hereafter Phyton). All modern ChipProg models are driven via a personal computer USB ports. This line of Phyton programmers works under control of the ChipProgUSB universal software, one for all USB hosted Phyton device programnmers available now and those are planned to be introduced soon. The ChipProg programmers support thousands of programmable memory devices, including EPROM, EEPROM, FLASH, NVRAM and OTP; programmable microcontrollers and logical devices: PAL, PLD and CPLD. The family includes four models shown below: ChipProg-48 andChipProg-40 (top row), ChipProg-G4 and ChipProg-ISP (bottom row). New **ChipProg** models will be added soon.

ChipProg-48 and ChipProg-40 programmers are intended for engineering and small volume manufacturing. These models allow operating on the devices before they are installed in the equipment (parallel programming) as well as on the devices already installed in the user's equipment (the method known as In-System Programming, or ISP, that uses serial data transmission into the programmable device). The ChipProg-ISP is a low-cost programmer for engineering, field service and manufacturing uses. The ChipProg-G4 is a gang programmer intended for small and middle-volume production; it has four programming sockets.

The ChipProgUSB software is intuitive and easy-to-use. See the User Interface topics. The software package includes an embedded script language that enables the automation of many routine operations – see the .

The ChipProgUSB software runs on the IBM PC hardware platform under the control of several

Windows™ versions (see the System requirements).

2.1 Comparisson matrix

Programmer Model	ChipProg-G4	ChipProg-48	ChipProg-40	ChipProg-ISP
Major features				
Primarily intended for	Production and chip replication	Engineering and low volume production	Engineering and low volume production	Engineering, low volume production and field service
Method of writing / reading information	Multi-site, concurrent, parallel, in socket	Single-site, parallel, in socket	Single-site, parallel, in socket	Single-site, serial, in system
Target devices	FLASH, EPROM, EEPROM, NVRAM, MCU, PLD	FLASH, EPROM, EEPROM, NVRAM, MCU, PLD	FLASH, EPROM, EEPROM, NVRAM, MCU	FLASH, EEPROM, MCU with ISP capability only
Universality in terms of the target support	Yes	Yes	Yes	Yes
	USB, 2.0		USB, 2.0	USB, 2.0
Multi-programming mode, Number of programmers driven from one PC	Yes, Unlimited	Yes, Unlimited	Yes, Unlimited	Yes, Unlimited
PC interface	USB, 2.0	USB, 2.0	USB, 2.0	USB, 2.0
Programming socket or cable	4 by 48 pin, DIL	48 pin, DIL	40 pin, DIL	Programming cable, 14 pin max
Adapters availability	Phyton brand and third party adapters	Phyton brand and third party adapters	Phyton brand and third party adapters	Phyton brand cables
Software update	Lifetime free of charge	Lifetime free of charge	Lifetime free of charge	Lifetime free of charge
Technical characteristics				
Built-in microcontroller, Fclk	Yes, 32-bit, 60 MHz	Yes, 32-bit, 60 MHz	Yes, 32-bit, 60 MHz	Yes, 8-bit, 10 MHz
Built-in FPGA, Fclk	Yes, up to 100 MHz	Yes, up to 100 MHz	Yes, up to 100 MHz	Yes, up to 10 MHz
Logical pin drivers	Universal, 1.8V to 5.5V	Universal, 1.8V to 5.5V	Universal, 1.8V to 5.5V	Universal, 1.8V to 5.5V
Analog drivers	Universal, 10-bit DAC	Universal, 10-bit DAC	Not universal	Not universal
Adjustability of the write impulses edges' slopes	Yes	Yes	Yes	Yes
	Unlimited	Unlimited	Limited by implementation of the analog drivers	Limited by implementation of the analog drivers
In-system programming capability	Yes, with additional cables	Yes, with additional cables	Yes, with additional cables	Yes
Chip insertion auto detect capability	Yes	Yes	Yes	
Correct chip insertion testing	Yes	Yes	Yes	Yes

Checking bad contact in the programming socket	Yes	Yes	Yes	No
Project management by the software shell	Yes	Yes	Yes	No
Serialization of the programmed devices	Yes	Yes	Yes	No
Writing signatures into the programmed devices	Yes	Yes	Yes	No
Logging programming sessions to files	Yes	Yes	Yes	No
Host computer and operation system	IBM PC, Windows 2000/XP/Vista	IBM PC, Windows 98/ME/ 2000/XP/Vista	IBM PC, Windows 98/ME/ 2000/XP/Vista	IBM PC, Windows 98/ME/ 2000/XP/Vista
Compare the programming + verification time for the selected devices (min: sec)				
M25P20				00:07
SST39VF016Q	00:45	00:45	00:45	02:50
MX28F640C3BB	00:56	00:56	00:56	02:27
MX29LV017A	00:23	00:23	00:23	02:56
MX29LV160CT	00:16	00:16	00:16	01:17
SST49LF008A	00:19	00:19	00:19	01:43
PIC18LF8722	00:11	00:11	00:11	00:19
AT89S51	00:01	00:01	00:01	00:01

2.2 ChipProg-48

The **ChipProg-48** universal programmer can be effectively used for both engineering and low-volume manufacturing. It supports in-socket and in-system programming of thousand of devices and has no valuable limitations in supporting future devices. The unlimited future device support differs **ChipProg-48** from the simplified and less expensive **ChipProg-40** model.

The programmer has a 48-pin DIP ZIF socket that enables inserting any wide or narrow DIP-packed devices with up to 48 leads without the necessity to use any additional adapters. Programming of other devices requires the use of additional adapters available from Phyton and a few selected vendors. The programmer has three LEDs for displaying the programming status (“Good”, “Busy”, “Error”) and the “Start” button for fast launch of the pre-programmed command chains. The palm-size programmer has a wall-plugged power adapter that is not shown on the picture above.

Standard package contents:

- One programmer unit
- One power adapter 12V/1A+
- One USB link cable
- One CD with the ChipProgUSB software

Optionally the package may include one or more programming adapters (if ordered with the programmer) and a “QuickStart” printed manual. See also for more details:

Major features
Hardware characteristics
Software features

2.2.1 Major features

1. Equipped with a 48 pin ZIF socket that allows insertion of the DIP-packed devices with the package width from 300 to 600 mil (7.62 to 15.24 mm) and the number of leads up to 48 without additional adapters.
2. Links to a PC USB 2.0 compatible port, e.g. slower USB connection is also supported.
3. Provides fast programming; for example, completely writes a 64M bit NOR FLASH in less than 50 sec.

4. Can program target devices in the programmer ZIF socket as well as the devices installed in the equipment (ISP mode).
5. ChipProg-48 tools can be driven from multiple USB ports of one computer (or via a USB hub) to provide concurrent programming of multiple devices of the same type.
6. Has a button for fast manual launch of any single operation or a bunch of operations.
7. Has three LEDs for displaying the programming status (“Good”, “Busy”, “Error”).

2.2.2 Hardware characteristics

1. The programmer has a 48-pin ZIF socket with a lever that enables the insertion and clamping of any DIP-packed devices with the package width from 300 to 600 mil (7.62 to 15.24 mm) and with the number of leads up to 48.
2. Adapters for programming devices in the SDIP, PLCC, SOIC, SOP, PSOP, TSOP, TSOPII, TSSOP, QFP, TQFP, VQFP, QFN, SON, BGA, CSP and other packages are available from Phyton and many third parties.
3. The programmer is built on the base of a very fast and productive 32-bit embedded microcontroller and FPGA. These resources allow adding new targets to the device list by a simple software update.
4. Most timing-critical parts of the programming algorithms are implemented on the FPGA devices and do not involve any operations on the embedded microcontroller that would slow down the programming speed.
5. Implementation in the FPGA devices logical drivers enables outputting logical signals of any level (low, high, Pullup, Pulldown and external clock generator) to any pin of the programming ZIF socket.
6. The tool hardware features 10-bit digital-to-analog converters for accurate settings of the analog signals.
7. The tool hardware enables accurate programming of the rising and falling edges of the generated analog signals.
8. The tool hardware automatically adjusts the generated analog signals.
9. The generated analog signals for both the target supplying and programming can be outputted to any pins of the device being programmed.
10. The tool hardware can connect any pin of the device being programmed to the “Ground” level.
11. The tool hardware checks if every pin of the target device is reliably fixed by a ZIF socket’s contacts (“bad contact” checking).
12. The tool hardware protects itself and the target device against incorrect insertions and other issues that cause a sharp increase in the currents though the target device circuits. This “over current” protection is very fast and reliable.
13. The target device pins are protected against the electrostatic discharge.
14. The tool’s hardware has a programmable clock generator.
15.

2.2.3 Software features

1. Works under control of Windows 95/98/ME/2000/XP/Vista.
2. Friendly, intuitive graphic user interface.
3. Includes a set of basic commands: Blank Check, Erase, Read, Write, Verify, Lock, Set Configuration Bits, Data Memory Support, etc., executed by a single mouse click or via menu.
4. Enables programming a batch of the commands above executed one after one either by a manual start or by a mouse click or automatically upon the device insertion into the programming socket.
5. Allows serialization of the programming devices with auto incrementing the device numbers and storing a serialization log.
6. The program can calculate checksums of the selected data array and then write the checksum into a specified memory location of the target device. Several methods of the checksum calculation can be

used.

7. The program allows writing a unique signature into a specified memory location of the target device for the device identification.
8. Project support speeds up and simplifies switching between different programming tasks.
9. The software allows pre-programming a particular operation (or a chain of operations), which is supposed to be automatically triggered when the programmer hardware detects insertion of the target device into the programming socket.
10. An unlimited number of memory buffers can be opened in the main ChipProgUSB window.
11. The software supports a multiple programming mode for concurrent programming of the same type of target devices on the same type of programmers connected to one cluster. The cluster size has no influence on the programming speed.
12. The software includes a full-scale binary editor allowing manual modification of the data in buffers as well as such helpful functions as Search and Replace, Fill, Compare, Copy, Invert, Calculate Checksum, and OR, AND, XOR logical operations on the blocks of data.
13. Loading and saving files in several standard and proprietary formats: Binary, Standard Extended Intel HEX, Motorola S-record, POF, JEDEC, PRG, Holtek OTP, ASCII HEC, ASCII OCTAL, Angstrem SAV. Special non-standard formats can be added on request.
14. The software is featured by a script language and a mechanism of handling the script scenarios for automation of the routine operations and chip replications.

2.3 ChipProg-40

The **ChipProg-40** universal programmer can be effectively used for both engineering and low-volume manufacturing. It supports in-socket and in-system programming of thousand of devices. The programmer hardware has some limitations for supporting certain devices. It does not support any PLDs. This is a difference between the cheaper **ChipProg-40** and the enhanced **-48**

The programmer has a 40-pin DIP ZIF socket that enables inserting any wide or narrow DIP-packed devices with up to 40 leads without the necessity to use any additional adapters. Programming of other devices requires use of additional adapters available from Phyton and a few selected vendors. The programmer has three LEDs for displaying the programming status ("Good", "Busy", "Error") and the "Start" button for fast launch of the pre-programmed command chains. The palm-size programmer has a wall-plugged power adapter that is not shown on the picture above.

Standard package contents:

- One programmer unit
- One power adapter 12V/1A+
- One USB link cable
- One CD with the ChipProgUSB software

Optionally the package may include one or more programming adapters (if ordered with the programmer) and a "QuickStart" printed manual. See also for more details:

Major features

Software features

2.3.1 Major features

1. Equipped with a 40 pin ZIF socket that allows insertion of any DIP-packed devices with the package width from 300 to 600 mil (7.62 to 15.24 mm) and the number of leads up to 40 without additional adapters.
2. Links to a PC USB 2.0 compatible port, e.g. slower USB connection is also supported.
3. Provides fast programming; for example, completely writes a 64M bit NOR FLASH in less than 50 sec.
4.
5. An unlimited number of ChipProg-40 tools can be driven from multiple USB ports of one computer (or via a USB hub) to provide concurrent programming of multiple devices of the same type.
6. Has a button for fast manual launch of any single operation or a batch of operations.
7. Has three LEDs for displaying the programming status ("Good", "Busy", "Error").

2.3.2 Hardware characteristics

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit risus, placerat et, rutrum nec, condimentum at, leo. Aliquam in augue a magna semper pellentesque. Suspendisse augue. Nullam est nibh, molestie eget, tempor ut, consectetuer ac, pede. Vestibulum sodales hendrerit augue. Suspendisse id mi. Aenean leo diam, sollicitudin adipiscing, posuere quis, venenatis sed, metus. Integer et nunc. Sed viverra dolor quis justo. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis elementum. Nullam a arcu. Vivamus sagittis imperdiet odio. Nam nonummy. Phasellus ullamcorper velit vehicula lorem. Aliquam eu ligula. Maecenas rhoncus. In elementum eros at elit. Quisque leo dolor, rutrum sit amet, fringilla in, tincidunt et, nisi.

Donec ut eros faucibus lorem lobortis sodales. Nam vitae lectus id lectus tincidunt ornare. Aliquam sodales suscipit velit. Nullam leo erat, iaculis vehicula, dignissim vel, rhoncus id, velit. Nulla facilisi. Fusce tortor lorem, mollis sed, scelerisque eget, faucibus sed, dui. Quisque eu nisi. Etiam sed erat id lorem placerat feugiat. Pellentesque vitae orci at odio porta pretium. Cras quis tellus eu pede auctor iaculis. Donec suscipit venenatis mi.

Aliquam erat volutpat. Sed congue feugiat tellus. Praesent ac nunc non nisi eleifend cursus. Sed nisi massa, mattis eu, elementum ac, luctus a, lacus. Nunc luctus malesuada ipsum. Morbi aliquam, massa eget gravida fermentum, eros nisi volutpat neque, nec placerat nisi nunc non mi. Quisque tincidunt quam nec nibh sagittis eleifend. Duis malesuada dignissim ante. Aliquam erat volutpat. Proin risus lectus, pharetra vel, mollis sit amet, suscipit ac, sapien. Fusce egestas. Curabitur ut tortor id massa egestas ullamcorper. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec fermentum. Curabitur ut ligula ac ante scelerisque consectetuer. Nullam at turpis quis nisl eleifend aliquam. Sed odio sapien, semper eget, rutrum a, tempor in, nibh.

2.3.3 Software features

1. Works under control of Windows 95/98/ME/2000/XP/Vista.
2. Friendly, intuitive graphic user interface.
3. Includes a set of basic commands: Blank Check, Erase, Read, Write, Verify, Lock, Set Configuration Bits, Data Memory Support, etc., executed by a single mouse click or via menu.
4. Enables programming a batch of the commands above executed one after one by a manual start, by a mouse click or automatically upon the device insertion into the programming socket.
5. Allows serialization of the programming devices with auto incrementing the device numbers and storing a serialization log.
6. The program can calculate checksums of the selected data array and then write the checksum into a specified memory location of the target device. Several methods of the checksum calculation can be used.
7. The program allows writing a unique signature into a specified memory location of the target device for the device identification.
8. Project support speeds up and simplifies switching between different programming tasks.
9.
10. ChipProgUSB window.
11. The software supports a multiple programming mode for concurrent programming of of the programmers connected to one cluster. The cluster size has no influence on the programming speed.
12. The software includes a full-scale binary editor allowing manual modification of the data in buffers as well as such helpful functions as Search and Replace, Fill, Compare, Copy, Invert, Calculate Checksum, and OR, AND, XOR logical operations on the blocks of data.
13.
14. The software is featured by a script language and a mechanism of handling the script scenarios for automation of the routine operations and chip replications.

2.4 ChipProg-G4

The **ChipProg-G4** is a 4-site gang programmer based on four **ChipProg-48** tools enclosed in onecase and driven from the ChipProgUSB software. It is intended for middle- and low-volume manufacturing. It supports in-socket and in-system programming of thousand of devices and has no valuable limitations for supporting future devices.

Standard package contents:

- One programmer unit
- One power cable
- One USB link cable
- One CD with the ChipProgUSB software

Optionally the package may include one or more programming adapters (if ordered with the programmer) and a “QuickStart” printed manual. See also for more details:

Major features
Hardware characteristics
Software features

2.4.1 Major features

1. Based on four ChipProg-48 tools enclosed in one case and connected to a PC via an embedded USB hub.
2. Allows independent and concurrent programming of up to four devices of the same type.
3. 48 pin ZIF sockets allow insertion of any DIP-packed devices with the package width from 300 to 600 mil (7.62 to 15.24 mm) and the number of leads up to 48 without additional adapters.
4. Links to a PC USB 2.0 compatible port via one link cable.
5. Provides fast programming; for example, completely writes a 64M bit NOR FLASH in less than 50 sec.
6. Can program target devices in its socket as well as devices installed in the equipment (ISP mode).
7. Each programming site has a 'Start' button for fast manual launch of any single operation or a batch of operations.
8. Each programming site has three LEDs for displaying the programming status (“Good”, “Busy”, “Error”).

2.4.2 Hardware characteristics

1. Enclosed in a durable steel case to be used in an industrial environment.
2. The tool gets power from a standard outlet 110-240V, 50-60 Hz.

3. Each programming site based on a single ChipProg-48 programmer has aof any DIP-packed devices with the package width from 300 to 600 mil (7.62 to 15.24 mm) and with the number of leads up to 48.
4. Adapters for programming devices in the SDIP, PLCC, SOIC, SOP, PSOP, TSOP, TSOPII, TSSOP, QFP, TQFP, VQFP, QFN, SON, BGA, CSP and other packages are available from Phyton and many third parties.
5. Single ChipProg-48 programmers inside of the tool enclosure are connected to an embedded USB 2.0 hub
6. Each programming site is built on the base of a very fast and powerful 32-bit embedded microcontroller and FPGA. These resources allow adding new targets to the device list by a simple software update.
7. Most timing-critical parts of the programming algorithms are implemented on the FPGA devices and do not involve any operations on the embedded microcontroller that would slow down the programming speed.
8. Implementation in the FPGA devices logical drivers enable outputting logical signals of any level (low, high, Pullup, Pulldown and external clock generator) to any pin of the programming ZIF socket.
9. The tool hardware features 10-bit digital-to-analog converters for accurate settings of the analog signals.
10. The tool hardware enables accurate programming of the rising and falling edges of the generated analog signals.
11. The tool hardware automatically adjusts the generated analog signals.
12. The generated analog signals for both the target supplying and programming can be outputted to any pins of the device being programmed.
13. The tool hardware can connect any pin of the device being programmed to the “Ground” level.
14. The tool hardware checks if every pin of the target device is reliably fixed by a ZIF socket’s contacts (“bad contact” checking).
15. The tool hardware protects itself and the target device against incorrect insertions and other issues that cause a sharp increase in the currents though the target device circuits. This “over current” protection is very fast and reliable.
16. The target device pins are protected against the electrostatic discharge.
17. The tool’s hardware has a programmable clock generator.
18. The self-testing procedure automatically executes at any time the programmer is powered on.

2.4.3 Software features

1. Works under control of Windows 2000/XP/Vista.
2. Friendly, intuitive graphic user interface allows monitoring the programming sites statuses and can zoom in on operations on each of four programming sites.
3. Includes a set of basic commands: Blank Check, Erase, Read, Write, Verify, Lock, Set Configuration Bits, Data Memory Support, etc., executed by a single mouse click or via menu.
4. Enables programming a batch of the commands above and executed one after one by a manual start, by a mouse click or automatically upon the device insertion into the programming socket.
5. Allows serialization of the programming devices with auto incrementing the device numbers and storing a serialization log.
6.
7. The program allows writing a unique signature into a specified memory location of the target device for the device identification.
8. Project support speeds up and simplifies switching between different programming tasks.
9. The software allows pre-programming a particular operation (or a chain of operations), which is supposed to be automatically triggered when the programmer hardware detects insertion of the target device into the programming socket.
10. Unlimited number of memory buffers can be opened in the main ChipProgUSB window for each of four programming sites.

11. The software includes a full-scale binary editor allowing manual modification of the data in buffers as well as such helpful functions as Search and Replace, Fill, Compare, Copy, Invert, Calculate Checksum, and OR, AND, XOR logical operations on the blocks of data.
12. Loading and saving files in several standard and proprietary formats: Binary, Standard Extended Intel HEX, Motorola S-record, POF, JEDEC, PRG, Holtek OTP, ASCII HEC, ASCII OCTAL, Angstrem SAV. Special non-standard formats can be added on request.
13. The software is featured by a script language and a mechanism of handling the script scenarios for automation of the routine operations and chip replications.

2.5 ChipProg-ISP

The **ChipProg-ISP** is a low-cost universal programmer specifically created for programming devices without removing them from the equipment where they are installed. This type of programming is known as “in-system” or “in-circuit”. The **ChipProg-ISP** supports serial EPROM and EEPROM flash memory devices and embedded microcontrollers with the code and data memory programmable via different types of serial ports: UART, JTAG, SPI and other types, including proprietary interfaces.

The programmer has three LEDs for displaying the programming status (“Good”, “Busy”, “Error”) and the “Start” button for fast launch of the pre-programmed command chains. The tool shown on the picture is very small and requires no power adapter for the operations - it gets power from the USB computer port.
.

Connecting ChipProg-ISP to the target

The programmer has a 14-pin output connector BH-14R. A variety of Phyton adapting cables allow connecting to the target. A simple pin-to-pin ribbon cable is supplied with the programmer by default, and other cables (adapters) can be ordered on demand. The BH-14R connector output information signals for the chip programming and some service signals that enable using the programmer in the automated programming and testing equipment. See the BH-14R pinout:

ChipProg-ISP BH-14R connector	Logical signal
1	Target specific*

2	Target specific*
3	Target specific*
4	Target specific*
5	Target specific*
6	Target specific*
7	Target specific*
8	Target specific*
9	GND
10	Target specific*
11	/Start
12	
13	/Good
14	/Busy

Signals on the pins #1 to #9 and on the pin #10 are used for transmitting and receiving information and synchro impulses to and from the target device. These signals are target specific and depend on the type of target device or a family in general (AVR, PIC, etc.) - see here. They also are shown in the adapters wiring diagrams; see the file included in the ChipProgUSB set.

The pin #9 must be connected to the target's ground.

The signals on the output pins #12, #13 and #14 represent the programmer statuses - logical '0' means an active status, logical '1' - passive. E.g.:

/Error – the operation has failed;
/Good – the operation completed successfully;
/Busy – the programmer is in a process of executing some operation.

An active signal on the input pin #11 (log.'0') starts the preset operation, the device programming by default. Activation of this signal, e.g. a falling edge, is equivalent to pushing the "Start" button on the programmer.

Read also In-System Programming for different devices.

Standard package contents:

- One programmer unit
- One universal ribbon cable wired pin-to-pin
- One USB link cable
- One CD with the ChipProgUSB software

Optionally the package may include one or more programming cable-adapters (if ordered with the programmer) and a “QuickStart” printed manual. See also for more details:

Major features

Hardware characteristics
Software features

2.5.1 Major features

1. Has a 14 pin socket for connecting to the target equipment by means of several cable-adapters.
2. Protects itself and the target equipment against incorrect wiring.
3. Links to a PC USB 2.0 compatible port, e.g. slower USB connection is also supported.
4. An unlimited number of ChipProg-ISP tools can be driven from multiple USB ports of one computer (or via a USB hub) to provide concurrent programming of multiple devices of the same type.
5. Has a button for fast manual launch of any single operation or a batch of operations.
6. Has three LEDs for displaying the programming status (“Good”, “Busy”, “Error”).

2.5.2 Hardware characteristics

1. Has a standard 14 pin connector.
2. Adapters for programming devices with in-system programming capability.
3. The programmer is built on the base of a very fast and productive 32-bit embedded microcontroller and FPGA devices. These resources allow adding new targets to the device list by a simple software update.
4. Most timing-critical parts of the programming algorithms are implemented on the FPGA devices and do not involve any operations on the embedded microcontroller that would slow down the programming speed.
5. Implementation in the FPGA devices logical drivers enable outputting logical signals of any level (low, high, Pullup, Pulldown and external clock generator) to any pin of the programming connector.
6. The tool hardware features 10-bit digital-to-analog converters for accurate settings of the analog signals.
7. The tool hardware enables accurate programming of the rising and falling edges of the generated analog signals.
8. The tool hardware automatically adjusts the generated analog signals.
9. The generated analog signals for both the target supplying and programming can be outputted to any pins of the device being programmed.
10. The tool hardware protects itself and the target device against incorrect connection.
11. The target device pins are protected against the electrostatic discharge.
12. Can be started from the external signal.
13. Three status signals “Good”, “Busy”, “Error” are outputted to the programmer connector.
14. The self-testing procedure automatically executes at any time the programmer is powered on.

2.5.3 Software features

1. Works under control of Windows 95/98/ME/2000/XP/Vista.
2. Friendly, intuitive graphic user interface.
3. Includes a set of basic commands: Blank Check, Erase, Read, Write, Verify, Lock, Set Configuration Bits, Data Memory Support, etc., executed by a single mouse click or via menu.
4. Enables programming a batch of the commands above executed one after one by a manual start, by a mouse click or automatically upon the device insertion into the programming socket.
5. Allows serialization of the programming devices with auto incrementing the device numbers and storing a serialization log.
6. The program can calculate checksums of the selected data array and then write the checksum into a specified memory location of the target device. Several methods of the checksum calculation can be used.