



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





# **ChipProg Device Programmers**

## **User's Guide**

**ChipProg-481,  
ChipProg-G41,  
ChipProg-48,  
ChipProg-40,  
ChipProg-ISP**

# ChipProg Device Programmers

© 2015 Phyton, Inc. **Microsystems and Development Tools**

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

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: May 2015 in (wherever you are located)

# Table of Contents

Foreword	0
<b>Part I Introduction</b>	<b>9</b>
1 Terms and Definitions.....	9
2 System Requirements.....	11
<b>Part II ChipProg Family Brief Description</b>	<b>12</b>
1 ChipProg-481.....	14
Major features .....	15
Hardware characteristics .....	15
Software features .....	15
2 ChipProg-G41.....	16
Major features .....	17
Hardware characteristics .....	18
Software features .....	18
3 ChipProg-48.....	19
Major features .....	20
Hardware characteristics .....	20
Software features .....	20
4 ChipProg-40.....	21
Major features .....	22
Hardware characteristics .....	23
Software features .....	23
5 ChipProg-ISP.....	24
Major features .....	26
Hardware characteristics .....	27
Software features .....	27
<b>Part III Quick Start</b>	<b>29</b>
1 Installing the ChipProgUSB Software.....	29
2 Installing the USB Drivers.....	31
3 Hardware installation.....	31
ChipProg-481 .....	32
ChipProg-G41 .....	33
ChipProg-48 .....	33
ChipProg-40 .....	34
ChipProg-ISP .....	35
4 Getting Assistance.....	36
On-line Help .....	36
Technical Support .....	36
Contact Information .....	37
<b>Part IV ChipProg Control Options</b>	<b>37</b>
1 Graphical User Interface.....	38

<b>User Interface Overview .....</b>	<b>38</b>
<b>Toolbars .....</b>	<b>39</b>
<b>Menus .....</b>	<b>39</b>
The File Menu.....	40
Configuration Files .....	41
The View Menu.....	41
The Project Menu .....	42
The Project Options Dialog.....	42
The Open Project Dialog.....	43
Project Repository.....	43
The Configure Menu.....	44
The Select Device dialog.....	45
The Buffers dialog.....	45
The Buffer Configuration dialog.....	46
Main Buffer Layer.....	46
Buffer Layers .....	47
The Serialization, Checksum and Log dialog.....	47
General settings.....	48
Device Serialization.....	49
Checksum .....	50
Signature string .....	51
Custom Shadow Areas.....	51
Overlapping data specified in shadow areas.....	51
Log file .....	52
The Preferences dialog.....	53
The Environment dialog.....	55
Fonts .....	55
Colors .....	56
Mapping Hot Keys.....	57
Toolbar .....	57
Messages .....	58
Miscellaneous Settings.....	58
Configuring Editor Dialog.....	59
General Editor Settings.....	59
The Editor Key Mapping.....	61
The Edit Key Command Dialog.....	61
The Commands Menu.....	62
Calculator .....	62
The Script Menu.....	63
The Window Menu.....	64
The Help Menu.....	65
<b>Windows .....</b>	<b>65</b>
The Program Manager Window .....	65
The Program Manager tab.....	66
Auto Programming.....	67
The Options tab .....	68
Split data .....	69
The Statistics tab.....	70
The Device and Algorithm Parameters window .....	71
Buffer Dump Window .....	74
The 'Configuring a Buffer' dialog.....	75
The 'Buffer Setup' dialog.....	76
The 'Display from address' dialog.....	78
The 'Modify Data' dialog.....	78

The 'Memory Blocks' dialog.....	78
The 'Load File' dialog.....	80
File Formats .....	81
The 'Save File' dialog.....	82
The Device Information window .....	83
Phyton programming adapters.....	83
Adapters for in-system programming.....	85
The Console Window .....	86
Windows for Scripts.....	86
<b>Simplified User Interface .....</b>	<b>87</b>
Settings of Simplified User Interface.....	88
Operations with Simplified User Interface.....	91
<b>2 Operations with Projects.....</b>	<b>91</b>
<b>3 Command Line Control.....</b>	<b>92</b>
Command line options .....	94
<b>4 On-the-Fly Control.....</b>	<b>97</b>
On-the-Fly command line options .....	98
On-the-Fly utility return codes .....	102
On-the-Fly Control example .....	103
<b>5 Script Files.....</b>	<b>105</b>
<b>The Script Files Dialog .....</b>	<b>105</b>
<b>How to create and edit script files .....</b>	<b>107</b>
The Editor Window .....	108
Text Edit .....	109
The Search for Text Dialog.....	110
The Replace Text Dialog.....	111
The Confirm Replace Dialog.....	112
The Multi-File Search Results Dialog.....	113
Search for Regular Expressions .....	113
The Set/Retrieve Bookmark Dialogs.....	114
Condensed Mode.....	114
The Condensed Mode Setup Dialog.....	114
Automatic Word Completion.....	115
Syntax Highlighting.....	115
The Display from Line Number Dialog .....	116
The Quick Watch Function.....	116
Block Operations.....	116
<b>How to start and debug script files .....</b>	<b>117</b>
The AutoWatches Pane.....	119
The Watches Window .....	119
The Display Watches Options Dialog.....	120
The Add Watch Dialog.....	121
The User Window .....	121
The I/O Stream Window .....	122
<b>6 Programming Automation via DLL.....</b>	<b>123</b>
<b>Application Control Interface .....</b>	<b>123</b>
<b>ACI Functions .....</b>	<b>125</b>
ACI_Launch.....	128
ACI_Exit .....	128
ACI_LoadConfigFile.....	129
ACI_SaveConfigFile.....	129
ACI_SetDevice.....	129

ACI_GetDevice.....	130
ACI_GetLayer .....	130
ACI_CreateBuffer.....	130
ACI_ReallocBuffer.....	130
ACI_ReadLayer.....	131
ACI_WriteLayer.....	131
ACI_FillLayer.....	131
ACI_GetProgrammingParams.....	132
ACI_SetProgrammingParams.....	132
ACI_GetProgOption.....	132
ACI_SetProgOption.....	133
ACI_AllProgOptionsDefault.....	134
ACI_ExecFunction.....	134
ACI_StartFunction.....	134
ACI_GangStart.....	135
ACI_GetStatus.....	135
ACI_TerminateFunction.....	135
ACI_GangTerminateFunction.....	135
ACI_FileLoad.....	136
ACI_FileSave.....	136
ACI_SettingsDialog.....	136
ACI_SelectDeviceDialog.....	136
ACI_BuffersDialog.....	137
ACI_LoadFileDialog.....	137
ACI_SaveFileDialog.....	138
ACI_SetConnection.....	139
ACI_GetConnection.....	139
<b>ACI Structures .....</b>	<b>140</b>
ACI_Launch_Params .....	141
ACI_Config_Params.....	141
ACI_Device_Params.....	142
ACI_Layer_Params.....	142
ACI_Buffer_Params.....	143
ACI_Memory_Params.....	145
ACI_Programming_Params.....	146
ACI_ProgOption_Params.....	148
ACI_Function_Params.....	152
ACI_GangTerminate_Params.....	154
ACI_PStatus_Params.....	154
ACI_File_Params.....	156
ACI_GangStart_Params.....	157
ACI_Connection_Params.....	158
<b>Examples of use .....</b>	<b>158</b>
<b>7 Control from NI LabVIEW .....</b>	<b>160</b>
Command Line Control from LabVIEW .....	160
Control from LabVIEW with DLL .....	163
<b>Part V Operating with Programmers .....</b>	<b>166</b>
<b>1 Inserting devices to a programming socket.....</b>	<b>166</b>
<b>2 Auto-detecting the device .....</b>	<b>166</b>
<b>3 Basic programming functions.....</b>	<b>167</b>
How to check if a device is blank .....	167

<b>How to erase a device</b> .....	<b>167</b>
<b>How to program a device</b> .....	<b>167</b>
How to load a file into a buffer .....	168
How to edit information before programming .....	168
How to configure the chosen device .....	168
How to write information into the device .....	168
<b>How to read a device</b> .....	<b>169</b>
<b>How to verify programming</b> .....	<b>169</b>
<b>How to save data on a disc</b> .....	<b>169</b>
<b>How to duplicate a device</b> .....	<b>170</b>
<b>4 Programming NAND Flash memory</b> .....	<b>170</b>
<b>NAND Flash memory architectures</b> .....	<b>170</b>
Invalid blocks .....	172
Managing invalid blocks .....	172
Skipping invalid blocks .....	172
Reserved Block Area .....	172
Error Checking and Correction .....	173
Invalid block map .....	173
Marking invalid blocks .....	174
<b>Programming NAND Flash devices by ChipProg</b> .....	<b>175</b>
Access Mode .....	176
Invalid Block Management .....	176
Spare Area Usage .....	176
Guard Solid Area .....	177
Tolerant Verify Feature .....	178
Invalid Block Indication Option .....	178
Access Mode Parameters .....	178
User Area .....	179
Solid Area .....	179
Reserved Block Area .....	180
ECC Frame size .....	180
Acceptable number of errors .....	180
<b>5 Multi- and Gang-programming</b> .....	<b>181</b>
<b>The Program Manager Window</b> .....	<b>182</b>
The Program Manager tab .....	183
The Options tab .....	183
The Statistics tab .....	184
<b>6 In-System Programming</b> .....	<b>186</b>
<b>Part VI References</b> .....	<b>187</b>
<b>1 Errors Messages</b> .....	<b>187</b>
<b>Error Load/ Save File</b> .....	<b>187</b>
<b>Error Addresses</b> .....	<b>187</b>
<b>Error sizes</b> .....	<b>188</b>
<b>Error command-line option</b> .....	<b>188</b>
<b>Error Programming option</b> .....	<b>188</b>
<b>Error DLL</b> .....	<b>189</b>
<b>Error USB</b> .....	<b>189</b>
<b>Error programmer hardware</b> .....	<b>189</b>
<b>Error internal</b> .....	<b>190</b>
<b>Error configuration</b> .....	<b>190</b>
<b>Error device</b> .....	<b>190</b>



Error check box .....	190
Error mix .....	190
Warning .....	191
<b>2 Expressions.....</b>	<b>191</b>
Operations with Expressions .....	191
Numbers .....	193
Examples of Expressions .....	193
<b>3 Script Language.....</b>	<b>194</b>
Simple example .....	194
Description .....	195
Built-in Functions .....	195
Built-in Variables .....	196
Difference between the Script and the C Languages .....	197
Script Language Built-in Functions and Variables .....	198
<b>4 In-System Programming for different devices.....</b>	<b>206</b>
Specific of programming PICmicro .....	206
Specific of programming AVR microcontrollers .....	206
Specific of programming Atmel 8051 microcontrollers .....	207
<b>Index</b>	<b>208</b>

## 1 Introduction



# ChipProg Device Programmers

## User's Guide

ChipProg-481  
 ChipProg-G41  
 ChipProg-48  
 ChipProg-40  
 ChipProg-ISP

Copyright © 2005-2014, Phyton, Inc. Microsystems and Development Tools, All rights reserved

### 1.1 Terms and Definitions

#### Terms used in the document

<b>Target device</b> or <b>Target</b>	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.
<b>Start and End Addresses</b> (of the Target device)	A range of the device physical memory for the programming operations (Read, Write, Verify, etc.).
<b>Device package</b> or <b>Package</b>	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.
<b>Programming socket</b> or <b>Programming ZIF socket</b> or <b>ZIF socket</b>	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. %CPN%>-40, ChipProg-48 and ChipProg-G41 are equipped with 40- and 48-pin ZIF

	sockets allowing operation on any DIP-packed devices with different numbers of leads and different widths and also connecting additional adapters for programming devices in other packages.
<b>Adapter or Package adapter</b>	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 Phytion brand adapters as well as third party adapters are available to support devices in most types of mechanical packages.
<b>File</b>	In the ChipProg context the term <b>file</b> 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.
<b>Buffer or Memory buffer</b>	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.
<b>Buffer layer or sub-layer</b>	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).
<b>Buffer size</b>	The buffers may have different sizes from 128KB to 32GB each.
<b>Buffer start address</b>	The address to display the buffer contents from.
<b>Checksum</b>	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.
<b>Parallel or In-socket programming</b>	Operations on a device being placed into the programmer's ZIF socket or into a programming adapter (opposite to the in-system programming below).
<b>ICP or in-circuit programming</b>	Programming devices mounted on the boards (in the user's equipment) via special adapter-cable connecting the programmer to the target.
<b>ISP or in-system programming</b>	Same as above. Programming devices mounted on the boards (in the user's equipment) via special adapter-cable connecting the programmer with the target.

<b>ISP Mode</b>	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.
<b>ISP JTAG Mode</b>	In-system programming via a JTAG interface.
<b>ISP HV Mode</b>	In-system programming that requires applying a relatively high voltage to the target device, (12V for example).
<b>Command Line mode</b>	A method of the ChipProg control by means of interacting with a computer program where the user issues commands to the program in the form of successive lines of text (command lines).
<b>Project</b>	An integrated set of information 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 EEPROM** and **Identifier 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 device programmers, you need an IBM PC-compatible personal computer with the following components:

- Pentium-V CPU or higher
- At least one USB port
- A hard drive with at least 200MB of free space
- Microsoft Windows XP, Windows 7 and Windows 8 operating systems.

## 2 ChipProg Family Brief Description

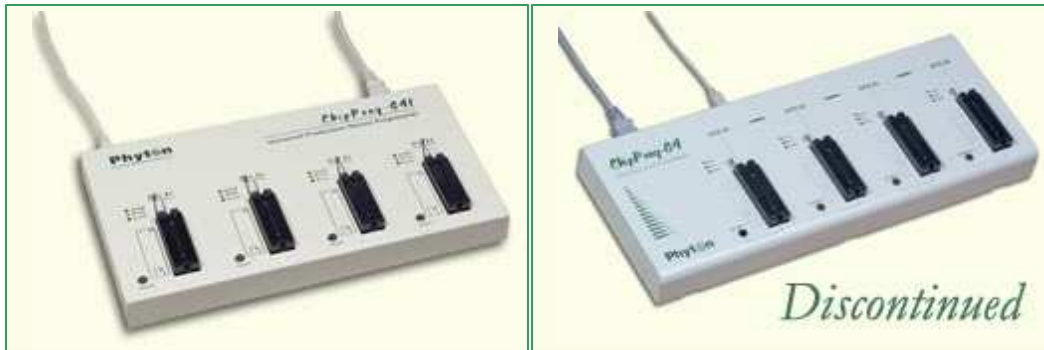
ChipProg is a family of device programmers produced by Phyton, Inc. Microsystems and Development Tools (hereafter Phyton). A current ChipProg portfolio includes several device programmers that belong to the following functional groups:

### Universal parallel programmers for engineering and small volume manufacturing



[ChipProg-481](#), [ChipProg-48](#) and [ChipProg-40](#) single-site device 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 or in-socket programming) as well as on the devices already installed in the user's equipment (the method known as [In-System Programming](#), that uses serial data transmission into the programmable device). [ChipProg-48](#) and [ChipProg-40](#) device programmers are controlled by the [ChipProgUSB](#) software, a newer [ChipProg-481](#) - by the [ChipProgUSB-01](#) software pack. Since both [ChipProgUSB](#) and [ChipProgUSB-01](#) models has the same [user interface](#), control tools, etc. this document operates with the [ChipProgUSB](#) term applicable to both software packages.

### Universal parallel gang programmers for manufacturing



[ChipProg-G41](#) and [ChipProg-G4](#) (, discontinued in 2010-2011) four-site gang device programmers are intended for mass production. The , [ChipProg-G41](#) gang machine is based on four concurrently and independently running ChipProg-481 device programmers. The [ChipProg-481](#) is controlled by the [ChipProgUSB-01](#) software pack.

#### Universal in-system device programmers for engineering and manufacturing



The [ChipProg-ISP](#) is a low-cost device programmer for engineering, field service and manufacturing. Unlimited number of these device programmers can be driven from one PC and can run concurrently to program multiple devices at a time. This device programmer works under control of the [ChipProgUSB](#) software pack.

#### How to choose a right device programmer?

- [ChipProg-40](#): the least expensive Phyton device programmer, intended for working with a limited device list; it does not support PLDs.
- [ChipProg-48](#): truly universal yet inexpensive device programmer; may not be fast enough for NAND programming.
- [ChipProg-481](#): truly universal device programmer: extremely fast, targeted to NAND memory support.
- [ChipProg-G41](#): extremely fast 4-site device programmer for production; can be controlled remotely from ATE.
- [ChipProg-ISP](#): inexpensive in-system programmer; multiple ChipProg-ISPs can be driven in the gang mode.

See [also](#).

## 2.1 ChipProg-481

The **ChipProg-481** 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-481** from the simplified, much slower and less expensive **ChipProg-40** model. The ChipProg-481 is much faster than the ChipProg-48 model, especially on high density NAND and NOR flash memory devices (~15-20 times faster).



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-01 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](#)

### 2.1.1 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 1Gb NAND K9F1G08U0A in 22 sec! (~640 sec by ChipProg-48)
4. Can program target devices in the programmer ZIF socket as well as the devices installed in the equipment (ISP mode).
5. An unlimited number of ChipProg-481 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”).

### 2.1.2 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 Phytion and selected third parties.
3. The programmer is built on the base of a very fast and productive 32-bit embedded microcontroller and a 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 in the FPGA devices.
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 programmers's hardware features 10-bit digital-to-analog converters for accurate settings of the analog signals.
7. The programmers's hardware enables accurate programming of the rising and falling edges of the generated analog signals.
8. The programmers's hardware automatically adjusts the generated analog signals in accordance to the target device programming specifications.
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 programmers's can connect any pin of the device being programmed to the “Ground” level.
11. The programmers's hardware checks if every pin of the target device is reliably fixed by a ZIF socket's contacts (“bad contact” checking).
12. The programmers's hardware protects itself and the target device against incorrect insertions and other issues that cause a sharp increase in the currents through the target device circuits. This “over current” protection is very fast and reliable.
13. The target device's pins are protected against the electrostatic discharge.
14. The programmers's hardware has a programmable clock generator.
15. The self-testing procedure automatically executes at any time the programmer is powered on.

### 2.1.3 Software features

1. The ChipProgUSB-01 software works under control of Windows XP, 7 (32- and 64-bit), 8.
2. Several methods of control are available: friendly and intuitive [Graphic User Interface](#) (GUI), [Command Line](#) control, Remote Control via the [Application Control Interface](#) (the DLL), [on-the-fly control](#) are

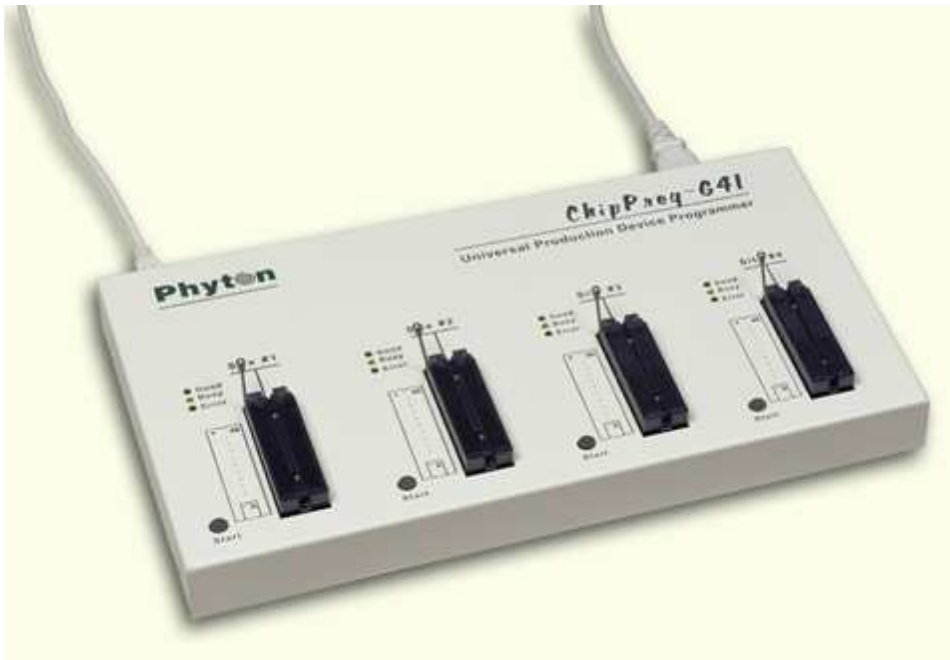


included into the standard ChipProgUSB package.

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.
4. Enables presetting 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 (in the Auto Programming mode only).
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 (in the Auto Programming mode only). 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 (in the Auto Programming mode only).
8. The 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-01 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 computer. A number of single device programmers connected to the programming cluster does not slow down 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, Angstrom SAV. Special non-standard formats can be added upon 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.

## 2.2 ChipProg-G41

The **ChipProg-G41** is a 4-site gang programmer based on four **ChipProg-481** programming modules enclosed in one case and driven from the ChipProgUSB-01 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-01 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. Based on four ChipProg-481 programming modules enclosed in a metal 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. A few ChipProg-G41 units can be cascaded to allow programming on 8-, 12-, 16- and more sites concurrently.
4. 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.
5. Links to a PC USB 2.0 compatible port via one link cable.
6. Provides fast programming; for example, completely writes a 1Gb NAND K9F1G08U0A flash device in 22 sec!
7. Can program target devices in its socket as well as devices installed in the equipment (ISP mode).
8. Each programming site has a 'Start' button for fast manual launch of any single operation or a batch of operations.
9. Each programming site has three LEDs for displaying the programming status (“Good”, “Busy”, “Error”).

## 2.2.2 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-481 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.
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 Phytion and many third parties.
5. Single ChipProg-481 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.

## 2.2.3 Software features

1. The ChipProgUSB-01 software works under control of Windows XP, 7 (32- and 64-bit), 8.
2. Several methods of control are available: friendly and intuitive [Graphic User Interface \(GUI\)](#), [Command Line](#) control, Remote Control via the [Application Control Interface](#) (the DLL), [on-the-fly control](#) are included into the standard ChipProgUSB package.
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.
4. Enables presetting 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 (in the Auto Programming mode only).
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 (in the Auto Programming mode only). 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 (in the Auto Programming mode only).
8. The 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-01 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 computer. A number of single device programmers connected to the programming cluster does not slow down 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, Angstrom SAV. Special non-standard formats can be added upon 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-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](#)

### 2.3.1 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. An unlimited number of 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”).

### 2.3.2 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 Phytion and selected third parties.
3. The programmer is built on the base of a very fast and productive 32-bit embedded microcontroller and a 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 in the FPGA devices.
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 programmers's hardware features 10-bit digital-to-analog converters for accurate settings of the analog signals.
7. The programmers's hardware enables accurate programming of the rising and falling edges of the generated analog signals.
8. The programmers's hardware automatically adjusts the generated analog signals in accordance to the target device programming specifications.
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 programmers's can connect any pin of the device being programmed to the “Ground” level.
11. The programmers's hardware checks if every pin of the target device is reliably fixed by a ZIF socket's contacts (“bad contact” checking).
12. The programmers's hardware protects itself and the target device against incorrect insertions and other issues that cause a sharp increase in the currents through the target device circuits. This “over current” protection is very fast and reliable.
13. The target device's pins are protected against the electrostatic discharge.
14. The programmers's hardware has a programmable clock generator.
15. The self-testing procedure automatically executes at any time the programmer is powered on.

### 2.3.3 Software features

1. The ChipProgUSB software works under control of Windows XP, 7 (32- and 64-bit), 8.
2. Several methods of control are available: friendly and intuitive [Graphic User Interface](#) (GUI), [Command Line](#) control, Remote Control via the [Application Control Interface](#) (the DLL), [on-the-fly control](#) are

- included into the standard ChipProgUSB package.
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.
  4. Enables presetting 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 (in the Auto Programming mode only).
  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 (in the Auto Programming mode only). 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 (in the Auto Programming mode only).
  8. The 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 computer. A number of single device programmers connected to the programming cluster does not slow down 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, Angstrom SAV. Special non-standard formats can be added upon 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.

## 2.4 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 **ChipProg-48** model.



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](#)

[Hardware characteristics](#)

[Software features](#)

### 2.4.1 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. Can program target devices in the programmer ZIF socket as well as the devices installed in the

equipment (ISP mode).

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”).

## 2.4.2 Hardware characteristics

1. The programmer has a 40-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 40.
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 selected third parties.
3. The programmer is built on the base of a very fast and productive 32-bit embedded microcontroller and a 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 in the FPGA devices.
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 programmers's hardware features 10-bit digital-to-analog converters for accurate settings of the analog signals.
7. The programmers's hardware enables accurate programming of the rising and falling edges of the generated analog signals.
8. The programmers's hardware automatically adjusts the generated analog signals in accordance to the target device programming specifications.
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 programmers's can connect any pin of the device being programmed to the “Ground” level.
11. The programmers's hardware checks if every pin of the target device is reliably fixed by a ZIF socket's contacts (“bad contact” checking).
12. The programmers's hardware protects itself and the target device against incorrect insertions and other issues that cause a sharp increase in the currents through the target device circuits. This “over current” protection is very fast and reliable.
13. The target device's pins are protected against the electrostatic discharge.
14. The programmers's hardware has a programmable clock generator.
15. The self-testing procedure automatically executes at any time the programmer is powered on.

## 2.4.3 Software features

1. The ChipProgUSB software works under control of Windows XP, 7 (32- and 64-bit), 8.
2. Several methods of control are available: friendly and intuitive [Graphic User Interface](#) (GUI), [Command Line](#) control, Remote Control via the [Application Control Interface](#) (the DLL), [on-the-fly control](#) are included into the standard ChipProgUSB package.
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.
4. Enables presetting 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 (in the Auto Programming mode only).
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 (in the Auto Programming mode only). 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 (in the Auto Programming mode only).
8. The 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 computer. A number of single device programmers connected to the programming cluster does not slow down 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, Angstrom SAV. Special non-standard formats can be added upon 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.

## 2.5 ChipProg-ISP

The **ChipProg-ISP** is an inexpensive universal device programmer for programming devices installed in the equipment. This type of programming is known as “in-system” or “in-circuit” programming. 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, SWD, 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 AS-ISP-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	/Error
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 adapters.chm 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. See the diagram below: