

# MP PIK+

MicroProcessor PIC® family  
programmer plus ISP

MANUALE UTENTE

USER MANUAL



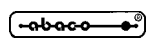
**grifo**®  
ITALIAN TECHNOLOGY

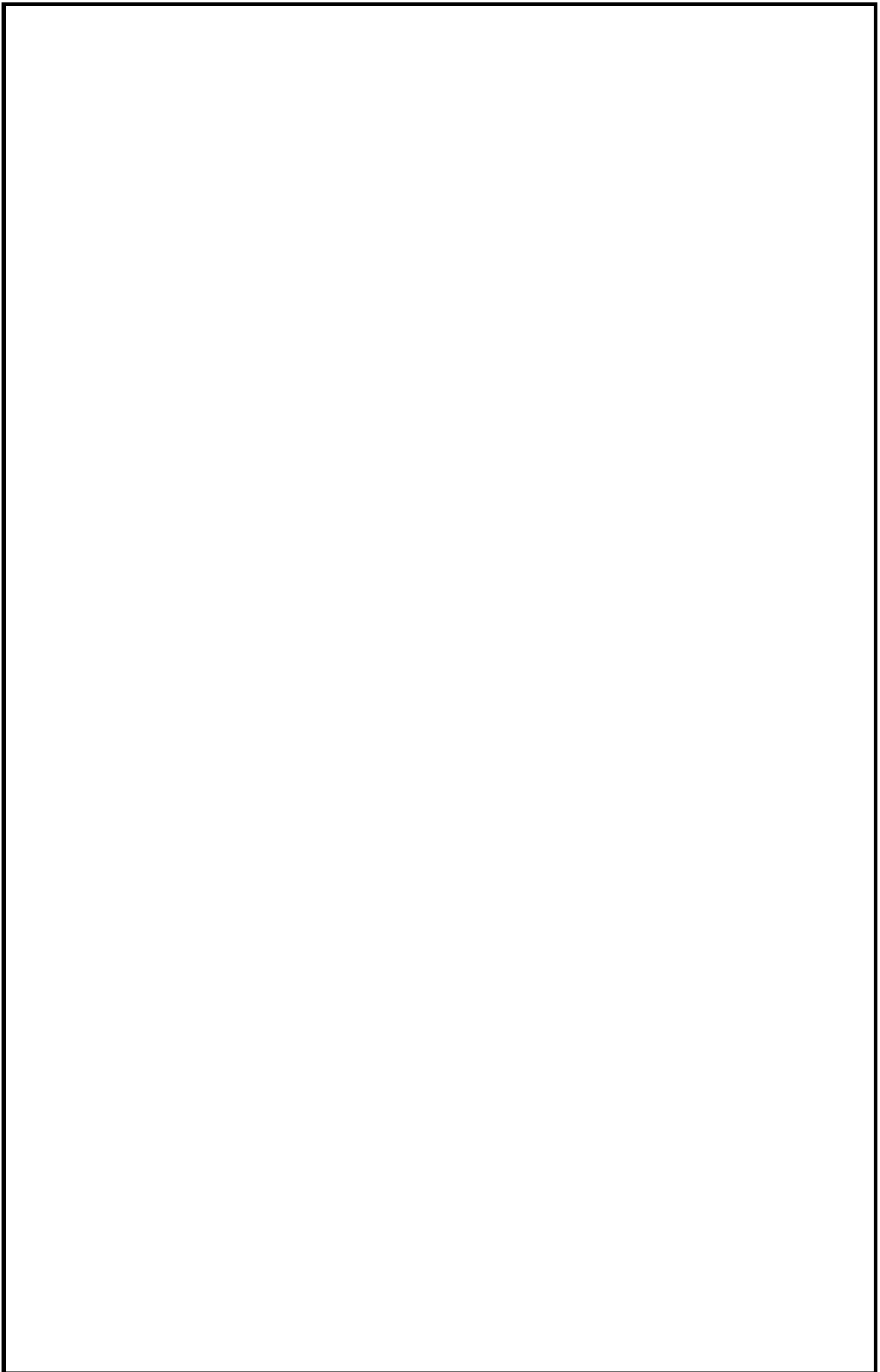
Via dell' Artigiano, 8/6  
40016 San Giorgio di Piano  
(Bologna) ITALY  
E-mail: [grifo@grifo.it](mailto:grifo@grifo.it)



<http://www.grifo.it>      <http://www.grifo.com>  
Tel. +39 051 892.052 (a. r.)      FAX: +39 051 893.661

MP PIK +      Edition 3.10      Rel. 09 September 2002

, GPC®, grifo®, are trade marks of grifo®



# MP PIK+

**MicroProcessor PIC® family  
programmer plus ISP**

**MP PIK+** è un programmatore compatto, veloce e potente in grado di supportare tutti i microcontrollori programmabili della famiglia PIC®.

Tramite il connettore ISP è inoltre in grado di programmare un dispositivo supportato direttamente su scheda.

Inoltre è in grado di programmare EEPROM seriali dotate di interfaccia IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx).

La sua concezione permette di aggiungere facilmente nuovi dispositivi e di fornire un prezzo molto competitivo insieme ad una elevata affidabilità nella programmazione. Indubbiamente il miglior rapporto prezzo/prestazioni tra i programmatori della sua classe.

**MP PIK+** is little, very fast and powerful portable programmer for PICmicro® family microcontrollers and serial EEPROM with IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx) interface types. Using build-in in-circuit serial programming (ISP) connector programmer is able to program PICmicro® family microcontrollers using serial algorithms.

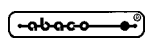
**grifo**®  
ITALIAN TECHNOLOGY

Via dell' Artigiano, 8/6  
40016 San Giorgio di Piano  
(Bologna) ITALY  
E-mail: [grifo@grifo.it](mailto:grifo@grifo.it)



<http://www.grifo.it>      <http://www.grifo.com>  
Tel. +39 051 892.052 (a. r.)      FAX: +39 051 893.661

**MP PIK +**      Edition 3.10      Rel. 09 September 2002

, **GPC**®, **grifo**®, are trade marks of **grifo**®

## DOCUMENTATION COPYRIGHT BY grifo®, ALL RIGHTS RESERVED

No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, either electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written consent of **grifo®**.

### IMPORTANT

Although all the information contained herein have been carefully verified, **grifo®** assumes no responsibility for errors that might appear in this document, or for damage to things or persons resulting from technical errors, omission and improper use of this manual and of the related software and hardware.

**grifo®** reserves the right to change the contents and form of this document, as well as the features and specification of its products at any time, without prior notice, to obtain always the best product.

For specific informations on the components mounted on the card, please refer to the Data Book of the builder or second sources.

### SYMBOLS DESCRIPTION

In the manual could appear the following symbols:

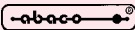


Attention: Generic danger



Attention: High voltage

### Trade Marks

, **GPC®**, **grifo®** : are trade marks of **grifo®**.

Other Product and Company names listed, are trade marks of their respective companies.

# INDICE GENERALE

<b>INTRODUZIONE .....</b>	<b>1</b>
<b>INFORMAZIONI PRELIMINARI .....</b>	<b>2</b>
<b>CONVENZIONI E TERMINOLOGIA .....</b>	<b>2</b>
<b>CONVENZIONI.....</b>	<b>2</b>
<b>TERMINOLOGIA .....</b>	<b>2</b>
<b>CARATTERISTICHE GENERALI .....</b>	<b>3</b>
<b>INTRODUZIONE.....</b>	<b>3</b>
<b>CARATTERISTICHE TECNICHE .....</b>	<b>6</b>
<b>SOCKET, PINDRIVER E DAC .....</b>	<b>6</b>
<b>DISPOSITIVI SUPPORTATI.....</b>	<b>6</b>
<b>TEMPI DI PROGRAMMAZIONE .....</b>	<b>6</b>
<b>OPERAZIONI SUI DISPOSITIVI .....</b>	<b>8</b>
<b>OPERAZIONI SUL BUFFER .....</b>	<b>8</b>
<b>TIPI DI FILES SUPPORTATI .....</b>	<b>8</b>
<b>SPECIFICHE TECNICHE.....</b>	<b>9</b>
<b>CONTENUTO DELLA CONFEZIONE .....</b>	<b>9</b>
<b>ISP (IN-SYSTEM SERIAL PROGRAMMING) .....</b>	<b>10</b>
<b>REGOLE GENERALI PER IN-SYSTEM SERIAL PROGRAMMING .....</b>	<b>10</b>
<b>STRUTTURA CONSIGLIATA DEL CONNETTORE ISP .....</b>	<b>11</b>
<b>INSTALLAZIONE .....</b>	<b>12</b>
<b>CONNESSIONE DI MP PIK+ COL PC .....</b>	<b>12</b>
<b>PROBLEMI DI COLLEGAMENTO MP PIK+ E PC .....</b>	<b>14</b>
<b>LEDS .....</b>	<b>14</b>
<b>SELFTTEST E CALIBRAZIONE .....</b>	<b>15</b>
<b>INSTALLAZIONE DEL PROGRAMMA .....</b>	<b>15</b>
<b>CONFIGURAZIONE MINIMA DEL PC .....</b>	<b>15</b>
<b>CONFIGURAZIONE CONSIGLIATA DEL PC .....</b>	<b>15</b>
<b>INSTALLAZIONE SOFTWARE .....</b>	<b>15</b>
<b>NUOVE VERSIONI DEL PROGRAMMA DI CONTROLLO .....</b>	<b>16</b>
<b>AGGIORNARE IL PROGRAMMA DI CONTROLLO .....</b>	<b>16</b>
<b>AVVIO RAPIDO.....</b>	<b>17</b>
<b>INSTALLAZIONE HARDWARE.....</b>	<b>17</b>
<b>INSTALLAZIONE SOFTWARE .....</b>	<b>18</b>
<b>PROGRAMMARE O COPIARE UN DISPOSITIVO .....</b>	<b>18</b>
<b>DESCRIZIONE SOFTWARE.....</b>	<b>20</b>
<b>UTILIZZO DEL SOFTWARE .....</b>	<b>20</b>
<b>DESCRIZIONE DELLA SCHERMATA PRINCIPALE .....</b>	<b>20</b>
<b>LISTA DEI TASTI SPECIALI .....</b>	<b>21</b>
<b>COMANDI DEL PROGRAMMA DI CONTROLLO .....</b>	<b>22</b>

CARATTERISTICHE SOFTWARE AGGIUNTIVE .....	45
ULTERIORI INFORMAZIONI.....	45
TERMINI DELLA GARANZIA .....	46
RISOLUZIONE DEI PROBLEMI .....	47
ERRORI DI COMUNICAZIONE .....	47
PROBLEMI DI LETTURA O PROGRAMMAZIONE .....	47
DISPOSITIVI NON SUPPORTATI .....	48
ARTICOLI AGGIUNTIVI .....	49
INDICE ANALITICO .....	50

# INDICE DELLE FIGURE

<b>FIGURA 1: FOTO DELL'MP PIK+</b> .....	5
<b>FIGURA 2: TEMPI DI PROGRAMMAZIONE DELL'MP PIK+</b> .....	6
<b>FIGURA 3: CONTENUTO DELLA CONFEZIONE</b> .....	7
<b>FIGURA 4: PINOUT CONNETTORE ISP</b> .....	11
<b>FIGURA 5: ESEMPIO DI CONNESSIONE ISP</b> .....	13
<b>FIGURA 6: CONNETTORE ISP CONSIGLIATO PER LA SCHEDA TARGET</b> .....	13
<b>FIGURA 7: DRIVER ISP DELL'MP PIK+</b> .....	13
<b>FIGURA 8: CONNETTORE ISP DELL'MP PIK+</b> .....	13
<b>FIGURA 9: DESCRIZIONE LEDs DELL'MP PIK+</b> .....	14
<b>FIGURA 10: POD DEL SELFTEST</b> .....	16
<b>FIGURA 11: PIANTA DEGLI ELEMENTI</b> .....	19
<b>FIGURA 12: CAVO PROGRAMMAZIONE ISP IN DOTAZIONE</b> .....	19
<b>FIGURA 13: LISTA DEI CONVERTITORI DI FORMATO</b> .....	49

# GENERAL INDEX

<b>GENERAL INTRODUCTION</b> .....	53
<b>PRELIMINARY INFORMATION</b> .....	54
<b>CONVENTIONS AND TERMINOLOGY</b> .....	54
<b>CONVENTIONS USED IN THE MANUAL</b> .....	54
<b>TERMINOLOGY USED IN THE MANUAL</b> .....	54
<b>GENERAL FEATURES</b> .....	55
<b>INTRODUCTION</b> .....	55
<b>TECHNICAL FEATURES</b> .....	58
<b>SOCKET, PIN DRIVERS AND DACS</b> .....	58
<b>DEVICE SUPPORT</b> .....	58
<b>PROGRAMMING SPEED</b> .....	58
<b>DEVICE OPERATIONS</b> .....	60
<b>BUFFER OPERATIONS</b> .....	60
<b>SUPPORTED FILE FORMATS</b> .....	60
<b>TECHNICAL SPECIFICATION</b> .....	61
<b>PACKAGE INCLUDED</b> .....	61
<b>ISP (IN-SYSTEM PROGRAMMING)</b> .....	62
<b>GENERAL RULES FOR IN-SYSTEM PROGRAMMING</b> .....	62
<b>THE RECOMMENDED ISP CONNECTOR LAYOUT IN TARGET SYSTEM</b> .....	63
<b>INSTALLATION</b> .....	64
<b>CONNECTING MP PIK+ TO THE PC</b> .....	64
<b>PROBLEMS RELATED TO MP PIK+ TO PC CONNECTION</b> .....	66
<b>LEDS</b> .....	66
<b>SELFTTEST AND CALIBRATION</b> .....	67
<b>INSTALLING PROGRAMMER SOFTWARE</b> .....	67
<b>MINIMAL PC CONFIGURATION</b> .....	67
<b>RECOMMENDED PC CONFIGURATION</b> .....	67
<b>SOFTWARE INSTALLATION</b> .....	67
<b>NEW VERSIONS OF PROGRAMMER SOFTWARE</b> .....	68
<b>UPGRADING THE PROGRAMMER SOFTWARE</b> .....	68
<b>QUICKSTART</b> .....	69
<b>INSTALLING PROGRAMMER HARDWARE</b> .....	69
<b>SOFTWARE INSTALLATION</b> .....	70
<b>QUICK INSTRUCTIONS TO PROGRAM OR COPY A DEVICE</b> .....	70
<b>SOFTWARE DESCRIPTION</b> .....	72
<b>USING THE PROGRAMMER SOFTWARE</b> .....	72
<b>LIST OF HOT KEYS</b> .....	73
<b>CONTROL PROGRAM COMMANDS</b> .....	74
<b>SOFTWARE ADDITIONAL FEATURES</b> .....	97
<b>OTHER</b> .....	97



**WARRANTY TERMS ..... 98**

**TROUBLESHOOTING ..... 99**

**COMMUNICATION ERRORS ..... 99**

**READING OR PROGRAMMING PROBLEMS ..... 99**

**UNSUPPORTED TARGET DEVICE ..... 100**

**ADDITIONAL TOOLS ..... 101**

**ALPHABETICAL INDEX ..... 102**



# FIGURES INDEX

<b>FIGURE 1: PHOTO OF PROGRAMMER MP PIK+</b> .....	<b>57</b>
<b>FIGURE 2: MP PIK+ PROGRAMMING TIMES</b> .....	<b>58</b>
<b>FIGURE 3: PACKAGE CONTENT</b> .....	<b>59</b>
<b>FIGURE 4: ISP PINOUT</b> .....	<b>63</b>
<b>FIGURE 5: ISP CONNECTION EXAMPLE</b> .....	<b>65</b>
<b>FIGURE 6: TARGET BOARD SUGGESTED ISP CONNECTOR</b> .....	<b>65</b>
<b>FIGURE 7: ISP DRIVER OF MP PIK+</b> .....	<b>65</b>
<b>FIGURE 8: ISP CONNECTOR OF MP PIK+</b> .....	<b>65</b>
<b>FIGURE 9: MP PIK+ LEDs DESCRIPTION TABLE</b> .....	<b>66</b>
<b>FIGURE 10: SELFTEST POD</b> .....	<b>68</b>
<b>FIGURE 11: ELEMENTS MAP</b> .....	<b>71</b>
<b>FIGURE 12: ISP PROGRAMMING CABLE</b> .....	<b>71</b>
<b>FIGURE 13: PACKAGE CONVERTER LIST</b> .....	<b>101</b>

## INTRODUZIONE

Scopo di questo manuale é la trasmissione delle informazioni necessarie all'uso competente e sicuro dei prodotti. Esse sono il frutto di un'elaborazione continua e sistematica di dati e prove tecniche registrate e validate dal Costruttore, in attuazione alle procedure interne di sicurezza e qualità dell'informazione.

Per un corretto rapporto coi prodotti, é necessario garantire leggibilità e conservazione del manuale, anche per futuri riferimenti. In caso di deterioramento o più semplicemente per ragioni di approfondimento tecnico ed operativo, consultare i nostri siti [www.grifo.it](http://www.grifo.it) o [www.grifo.com](http://www.grifo.com) o direttamente l'Assistenza Tecnica autorizzata.

Al fine di non incontrare problemi nell'uso di tali dispositivi, é conveniente che l'utente - **PRIMA DI COMINCIARE AD OPERARE** - legga con attenzione tutte le informazioni contenute in questo manuale. In una seconda fase, per rintracciare più facilmente le informazioni necessarie, si può fare riferimento all'indice generale e all'indice analitico, posti rispettivamente all'inizio ed alla fine del manuale.

Le informazioni contenute in questo manuale sono aggiornate alla data di rilascio del medesimo, ma i nostri prodotti subiscono continui aggiornamenti e miglioramenti. Si prega di consultare il file `README_P.TXT` per i cambiamenti dell'ultimo minuto.

Il programmaqui descritto è coperto da diritto d'autore, tutti i diritti sono riservati. Nè il programma nè alcuna sua parte possono essere analizzati, disassemblati o modificati in alcun modo, con qualunque mezzo, per qualunque scopo.

Questo documento è coperto da diritto d'autore, tutti i diritti sono riservati. Questo documento non può essere copiato, riprodotto o tradotto in alcun modo o mediante alcun mezzo, nè per intero nè in parte, senza il permesso scritto della **grifo®**.

**grifo®** non si assume alcuna responsabilità per l'uso errato di questo manuale.

**grifo®** si riserva il diritto di apportare cambiamenti o miglioramenti ai prodotti descritti in questo manuale in qualunque momento senza darne notizia.

Questo manuale contiene nomi di aziende, software, prodotti, ecc. che sono marchi registrati dai rispettivi proprietari. **grifo®** rispetta tale diritto.

## INFORMAZIONI PRELIMINARI

**MP PIK+** è un programmatore compatto, veloce e potente in grado di supportare tutti i microcontrollori programmabili della famiglia PIC®.

Tramite il connettore ISP è inoltre in grado di programmare un dispositivo supportato direttamente su scheda.

Inoltre è in grado di programmare EEPROM seriali dotate di interfaccia IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx).

La sua concezione permette di aggiungere facilmente nuovi dispositivi e di fornire un prezzo molto competitivo insieme ad una elevata affidabilità nella programmazione. Indubbiamente il miglior rapporto prezzo/prestazioni tra i programmatori della sua classe.

## CONVENZIONI E TERMINOLOGIA

Alcuni termini e convenzioni specifici vengono usati in questo manuale:

### CONVENZIONI

I riferimenti alle funzioni del programma di controllo sono in maiuscolo, ad esempio **LOAD**, **FILE**, **DISPOSITIVO**, etc.

I riferimenti ai pulsanti di controllo sono racchiusi tra le parentesi angolari <>, ad esempio <**F1**>.

### TERMINOLOGIA

<b>Dispositivo target</b>	Un qualunque tipo di dispositivo o circuito integrato programmabile.
<b>Zoccolo ZIF</b>	Zoccolo a forza di inserzione nulla usato per inserire il dispositivo target nel programmatore.
<b>Buffer</b>	Area di memoria o porzione di disco rigido utilizzata per memorizzare temporaneamente dei dati.
<b>Porta Stampante</b>	Tipo di porta per PC (parallela) principalmente usata per connessione con la stampante.
<b>ISP</b>	In-circuit Serial-Programming, ovvero la capacità di programmare un dispositivo target senza toglierlo dall'applicativo in cui si trova, anche se è saldato. <u>La scheda applicativo deve essere compatibile con questa caratteristica.</u>
<b>Formato HEX</b>	Formato di file di date che si può leggere con i normale editor di testo; ad esempio il byte 5AH viene memorizzato come i caratteri '5' ed 'A' corrispondenti ad i bytes 35H e 41H. Una linea di questo tipo di file (un record) contiene indirizzo iniziale e dati da scrivere, inoltre ogni record è provvisto di checksum.

Questo manuale utente è comune a tutti i programamtori, quindi contiene informazioni valide per tutti i modelli ed informazioni specifiche per **MP PIK+**.

## CARATTERISTICHE GENERALI

Questo capitolo contiene una descrizione completa ed esauriente del programmatore, incluso un sommario delle sue caratteristiche tecniche ed operazionali.

### INTRODUZIONE

**MP PIK+** è un programmatore portatile specializzato di ultima generazione (piccolo, molto veloce e potente) basato sull'esperienza **grifo®** per Windows 95/98/Me/NT/2000/XP.

**MP PIK+** è in grado di supportare tutte le serie attualmente disponibili di microcontrollori Microchip™ PICmicro® da 8 a 40 pins, grazie alla sua tecnologia ISP incorporate è in grado di programmare dispositivi Microchip™ PICmicro® sia in modalità parallela che seriale. Inoltre è in grado di programmare EEPROM seriali dotate di interfaccia IIC (24Cxx), Microwire (93Cxx) ed SPI (25Cxx).

**MP PIK+** è stato sviluppato in stretta collaborazione con Microchip™, quindi l'hardware del programmatore è particolarmente mirato al supporto dei futuri microcontrollori Microchip™ PICmicro®, ha un prezzo molto competitivo ma è un programmatore affidabile grazie al suo eccellente progetto hardware. Il miglior rapporto prezzo/prestazioni nella sua gamma.

**MP PIK+** è in grado di supportare la maggior parte dei dispositivi in ISP grazie al connettore incorporato che rispetta lo standard Atmel.

**MP PIK+** si interfaccia con personal computer IBM PC, AT o superiori, sia portatili che desktop. Il programmatore permette una connessione diretta col PC tramite la porta parallela (stampante) standard (non servono interfacce specifiche). Si raccomanda l'uso di una porta parallela (stampante) sul bus PCI che sia pienamente compatibile con lo standard IEEE 1284 (ECP/EPP). Anche il programma di controllo del **MP PIK+** supporta pienamente lo standard IEEE 1284.

**MP PIK+** ha una notevole velocità di programmazione grazie all'hardware basato su FPGA ed al supporto della porta parallela ECP/EPP.

**MP PIK+** è dotato di base con 40 potenti pindrivers TTL che generano livelli H/L/pull\_up/pull\_down e forniscono la capacità di lettura. I nostri pindriver sono avanzati tecnologicamente e permettono di ottenere programmazioni di alta qualità ad alta velocità senza provocare sovralongazioni o rimbalzi di massa per tutte le tipologie di dispositivi programmabili. Inoltre sono in grado di lavorare fino a 1,8 V cosicché sarete già da adesso in grado di programmare la totalità dei dispositivi a bassa tensione attualmente disponibili.

Il programmatore esegue il **test di corretto inserimento** di un dispositivo e di **buon contatto elettrico** prima di iniziare la programmazione. Tali capacità, integrate dalla protezione contro le sovracorrenti e dal controllo del byte di signature aiutano a prevenire i danni dovuti ad errori dell'operatore.

Le risorse hardware permettono addirittura di eseguire un **selftest**, in modo che il programma di controllo sia sempre in grado di verificare i pindrivers, i livelli di tensione, le temporizzazioni e la comunicazione col PC.

**MP PIK+** effettua **verifiche** di programmazione **a livelli minimi** di tensione di alimentazione, il che, ovviamente, migliora l'affidabilità e garantisce una lunga conservazione dei dati.

**MP PIK+** è pilotato da un programma di controllo facile da usare, con menù a tendina, tasti di scelta rapida ed aiuto in linea. Si può selezionare un dispositivo target mediante la classe a cui appartiene, dal nome del produttore o semplicemente digitando una parte del nome del rivenditore e/o del dispositivo stesso. Oltre alle normali operazioni eseguibili sui dispositivi (lettura, verifica se è vuoto, programmazione, verifica di programmazione, cancellazione), sono state implementate alcune funzioni di verifica (verifica di inserzione, controllo del byte di signature) ed alcune funzioni speciali (autoincremento del numero di serie, modalità di produzione, inizio della programmazione all'inserzione). Il programma di controllo permette la manipolazione dei dati dentro il buffer, tra il buffer ed i files su disco, esegue la rilevazione automatica del tipo di file e la eventuale conversione. Il programma è disponibile in inglese ed ulteriormente in altre lingue.

La ricca funzione di **autoincremento** di **MP PIK+** permette di assegnare numeri di serie individuali ad ogni dispositivo programmato. Tale funzione semplicemente incrementa un numero di serie nel buffer ogni volta che un nuovo dispositivo viene inserito. Inoltre, la funzione permette all'operatore di prelevare i numeri di serie e/o le signature di identificazione del dispositivo da un file.

Sono disponibili diversi convertitori dal formato DIL a PLCC, SOIC e altri formati.



FIGURA 1: FOTO DELL'MP PIK+

## CARATTERISTICHE TECNICHE

### SOCKET, PINDRIVER E DAC

- Porta stampante IEEE 1284 basata su FPGA, fino ad 1MB/s
- Zoccolo 40-pin DIL ZIF (Zero Insertion Force) accetta dispositivi 300/600 mil fino a 40-pin
- 2 D/A converter per VCCP e VPP, con tempi di discesa e risalita controllabili
- VCCP range 0..7V/250mA
- VPP range 0..15V/150mA
- Potente processore di bordo (20MHz) con macchina a stati basata su FPGA, con clock 20MHz
- Pindriver GND/VCC/VPP specifici per dispositivi PICmicro®
- Pin di GND, VCCP e VPP liberi per facilitare la compatibilità con dispositivi futuri
- Driver TTL basato su FPGA per generare H, L, CLK, pull-up e pull-down o lettura su ogni pin
- Livello H selezionabile da 1, 8 V fino a 5V
- In-circuit Serial Programming (ISP): capacità di ISP già abilitate

### DISPOSITIVI SUPPORTATI

- Microcontrollori Microchip™ PICmicro®: 12xxx, 14xxx, 16xxx, 17xxx and 18xxx series, da 8 a 40 pin (\*1), sia parallelo che ISP
- E(E)PROM Seriali: serie 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx, pieno supporto per serie LV

### TEMPI DI PROGRAMMAZIONE

Questi tempi dipendono fortemente dalla velocità del PC, dal tipo di porta LPT (parallela) e dalle risorse del sistema operativo disponibili. Pertanto i valori indicati prendono in considerazione due diverse configurazioni del PC.

**Configurazione A:** *Pentium MMX, 250 MHz, ECP/EPP, WIN98.*

**Configurazione B:** *Athlon, 750 MHz, ECP/EPP su bus PCI, WIN98.*

DISPOSITIVO	OPERAZIONE	TEMPO A	TEMPO B
PIC16C67	Programmazione e verifica	18 sec	16 sec

**FIGURA 2: TEMPI DI PROGRAMMAZIONE DELL'MP PIK+**





FIGURA 3: CONTENUTO DELLA CONFEZIONE

## **OPERAZIONI SUI DISPOSITIVI**

### **STANDARD**

- Selezione intelligente del dispositivo per tipo, produttore o typed frazione del nome
- Programmazione ISP
- Verifica se è vuoto e di corretta programmazione, lettura
- Programmazione
- Cancellazione
- Programmazione di bit di configurazione e sicurezza
- Test bit per illegalità
- Checksum

### **SICUREZZA**

- Verifica di inserzione corretta e rovesciata
- Controllo della bontà dei contatti elettrici
- Controllo del byte di ID

### **OPERAZIONI SPECIALI**

- Incremento automatico del numero di serie
- Statistiche
- Conto alla rovescia dei dispositivi programmati

## **OPERAZIONI SUL BUFFER**

- Esamina/modifica, trova/sostituisci
- Riempi/copia, sposta, scambia byte, spezza word o dword
- Checksum (byte, word)
- Stampa

## **TIPI DI FILES SUPPORTATI**

- Binario
- HEX: Intel, Intel EXT, Motorola S-record, MOS, Exormax, Tektronix, ASCII-space-HEX

## **SPECIFICHE TECNICHE**

### **GENERALI**

- Tensione di lavoro 12...15V AC/max. 0,5A (alimentatore compreso nella confezione)
- Consumo di potenza max. 5 W in piena attività, circa 0,5W in sospensione
- Dimensioni 132x66x30 [mm] (5.2x2.6x1.2 [inch])
- Peso (senza adattatori esterni) 200 g
- Temperatura 5°..40°C
- Umidità 20%..80%, senza condensa

### **CONTENUTO DELLA CONFEZIONE**

- Programmatore
- Cavo con due connettori tipi D da 25 pin, per collegare il programmatore al PC
- Cavo ISP
- Adattatore di rete
- POD diagnostico per auto diagnosi del programmatore
- Coperchio di metallo per zoccolo ZIF (anti polvere)
- Mini-CD con il programma di controllo e altre di utilità (compreso il presente manuale utente)
- Confezione di cartone

## **ISP (IN-SYSTEM SERIAL PROGRAMMING)**

In-system Serial Programming permette di programmare diverse volte un dispositivo già saldato sul sistema applicativo. Utilizzando una semplice interfaccia, il programmatore ISP comunica via linea seriale con il dispositivo, riprogrammandone la memoria non volatile di bordo.

Questo fatto elimina la necessità di rimuovere il dispositivo dalla scheda su cui è installato ogni volta che lo si deve riprogrammare risparmiando così tempo e denaro sia durante lo sviluppo in laboratorio sia per l'aggiornamento dei parametri o del software sul campo.

## **REGOLE GENERALI PER IN-SYSTEM SERIAL PROGRAMMING**

Il paragrafo fa riferimento alla figura 5.

Si raccomanda specialmente di rispettare le seguenti regole per evitare il danneggiamento del programmatore ISP e/o del PC e/o del dispositivo target e/o della scheda su cui è installato:

- Si assicuri che ci sia un unico punto comune di terra tra la scheda su cui si trova il dispositivo target, il programmatore ISP ed il PC.
- Per chi utilizza portatili o altri PC non connessi ad un punto comune di terra: si effettui una connessione mediante un conduttore verso il punto comune di terra (si usi, ad esempio, la vaschetta metallica esterna del connettore parallelo del PC).

Anche i dispositivi collegati alla scheda su cui avviene la programmazione devono essere collegati al punto comune di terra.

Durante la programmazione ISP avviene una connessione elettrica tra due dispositivi (il dispositivo target ed il programmatore ISP). Un connessione non effettuata propriamente potrebbe danneggiare entrambi i dispositivi.

**NOTA: Se il programmatore viene danneggiato durante una programmazione e non sono state rispettate le specifiche di cui sopra la garanzia non è valida.**

1. Spegnerne sia il programmatore che il dispositivo target.
2. Collegare lo stesso punto comune di massa per tutti i dispositivi, ad esempio collegare tutte le masse insieme mediante un conduttore.
3. Inserire il cavo in dotazione al programmatore nel connettore ISP di quest'ultimo, alimentare il programmatore e lanciare il programma di controllo.
4. Selezionare il dispositivo target e le sue eventuali opzioni nel programma di controllo.
5. Iniziare le operazioni (lettura, programmazione).
6. Collegare il connettore ISP alla scheda con il dispositivo target quando il programma di controllo indica di farlo.
7. Rimuovere il connettore ISP dalla scheda con il dispositivo target quando il programma di controllo indica di farlo.
8. Se dovete effettuare un'altra operazione sul dispositivo target, tornate al punto 5.

### **NOTA:**

Se la scheda del dispositivo target utilizza gli stessi segnali della programmazione ISP, è necessario separarli mediante opportuni resistori.

## STRUTTURA CONSIGLIATA DEL CONNETTORE ISP

L'interfaccia implementata dal programmatore rispetta le specifiche della application note Atmel AVR910: In-System Programming. Tale application note viene usata da tutti i programmatori ISP e descrive l'interfaccia ISP consigliata da installare sulla scheda che ospita il dispositivo target (vista dall'alto in figura 6).

PIN	NOME	SCOPO
SCK	Serial Clock	Clock di programmazione, generato dal programmatore ISP (master).
MOSI	Master Out Slave In	Linea di comunicazione dal programmatore ISP (master) al dispositivo target (slave).
MISO	Master In Slave Out	Linea di comunicazione dal dispositivo target (slave) al programmatore ISP (master).
RESET	Target MCU Reset	Per abilitare la programmazione ISP la linea di reset del dispositivo target deve essere mantenuta attiva. Per semplificare questa operazione il programmatore ISP dovrebbe poter pilotare il reset del dispositivo target.
V <sub>cc</sub>	Target Power Supply	Il dispositivo target può essere alimentato dal programmatore ISP durante la programmazione. SEP 40 non può essere a sua volta alimentato dal dispositivo target.
GND	Common Ground	Il punto di massa deve essere comune a tutti i dispositivi.

**FIGURA 4: PINOUT CONNETTORE ISP**

La configurazione dei pin nel connettore ISP del programmatore può dipendere da quale dispositivo viene programmato e viene sempre specificata nella finestra **Additional info window**. Tale specifica corrisponde alle application notes pubblicate dal costruttore del dispositivo target.

**Note:** Il pin numero 1 viene indicato da un triangolo sul connettore ISP.

*Il driver H/L/read dell'MP PIK+ viene mostrato nella figura 7.*

### Avvertimenti:

- **Quando MP PIK+ viene usato come programmatore ISP, non inserite dispositivi nello zoccolo ZIF.**
- **Quando MP PIK+ viene usato come programmatore parallelo, non inserite il cavo ISP.**
- **Usate solo il cavo ISP in dotazione.** Con altri cavi ISP (di diverso materiale, lunghezza, ecc.), la programmazione potrebbe essere non affidabile.
- **MP PIK+ può alimentare il dispositivo target (in maniera limitata anche la scheda su cui è montato), ma il viceversa non è possibile.**
- **MP PIK+ applica la tensione di programmazione al dispositivo target e ne verifica il valore (la scheda su cui è montato ne può alterare il valore).** Se il valore misurato è diverso da quello atteso non verrà effettuata alcuna operazione sul dispositivo target.

## INSTALLAZIONE

### CONNESSIONE DI MP PIK+ COL PC

Spegnete PC e programmatore. Inserite il cavo di comunicazione fornito in una porta parallela libera del vostro PC. Rimuovete un cavo per stampante eventualmente già presente. Collegate l'altra estremità **MP PIK+**. Stringete le viti dei connettori, questo è molto importante. Anche se potete trovare che è scomodo dover ogni volta staccare il cavo della stampante e ricollegare quello dell'**MP PIK+**, dovete comunque considerare che non è possibile collegare il programmatore tramite un data switch, sia meccanico che elettronico. Potete comunque installare una scheda supplementare di I/O per dedicare una porta, esempio LPT2, all'**MP PIK+**, mentre la stampante può rimanere sulla porta LPT1.

Collegate l'alimentatore fornito in dotazione alla presa di alimentazione di rete e collegate il connettore mini DIN alla presa di alimentazione etichettata "12 VAC" dell'**MP PIK+**. Appena collegata l'alimentazione tutti i LED si accendono simultaneamente, poi i tre LED di stato si spengono e si riaccendono in sequenza uno dopo l'altro, infine il LED di alimentazione si pone in bassa luminosità, a questo punto l'**MP PIK+** è pronto a ricevere comandi. Accendete anche il PC e fate partire il programma di controllo.

#### **! ATTENZIONE !**

Se non volete spegnere il PC mentre collegate il programmatore, procedete come segue:

- Quando connettete il programmatore al PC: **PRIMA** inserite il cavo di comunicazione **POI** il connettore di alimentazione.
- Quando scollegate il programmatore dal PC: **PRIMA** scollegate il connettore di alimentazione **POI** il cavo di comunicazione.

Dal punto di vista del programmatore la sequenza di collegamento e scollegamento è irrilevante, i circuiti di protezione lo tengono al sicuro, **ma ciò non vale per il vostro PC.**

#### **NOTA**

Quando l'**MP PIK+** non viene usato, il programma di controllo commuta il programmatore in sospensione (sleep mode), in questo stato il consumo è meno di 1 W per cui il programmatore non ha un interruttore di spegnimento. Quando il LED di alimentazione è acceso a bassa intensità il programmatore è in sospensione.

L'elettronica incorporata nell'**MP PIK+** protegge il programmatore stesso ed il dispositivo da programmare contro assenze di alimentazione lunghe o brevi e, in parte, contro malfunzionamenti del PC.

Non è possibile, comunque, garantire l'integrità del dispositivo da programmare se l'utente inserisce parametri errati nel programma di controllo.

Il dispositivo target può non essere distrutto da interruzioni forzate del programma di controllo (reset o spegnimento del PC) o dalla rimozione della connessione fisica col programmatore ma il suo contenuto può rimanere indefinito.

Non togliate il dispositivo target dallo zoccolo ZIF durante l'accesso da parte dell'**MP PIK+**, ovvero quando il LED BUSY è acceso.

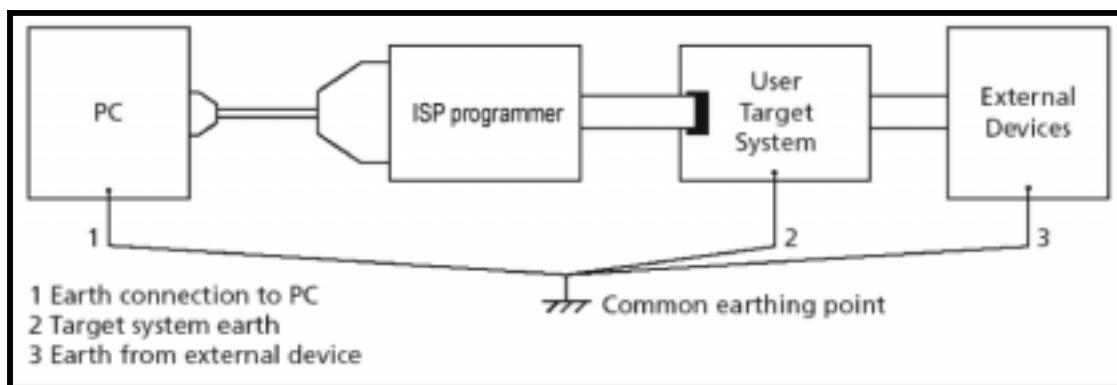


FIGURA 5: ESEMPIO DI CONNESSIONE ISP

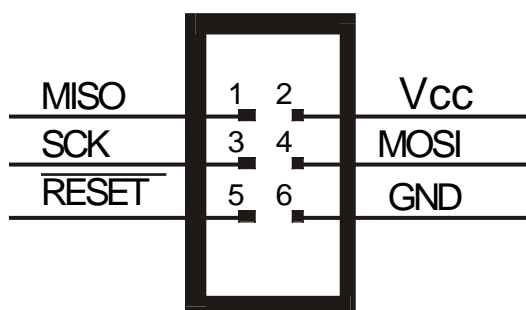


FIGURA 6: CONNETTORE ISP CONSIGLIATO PER LA SCHEDA TARGET

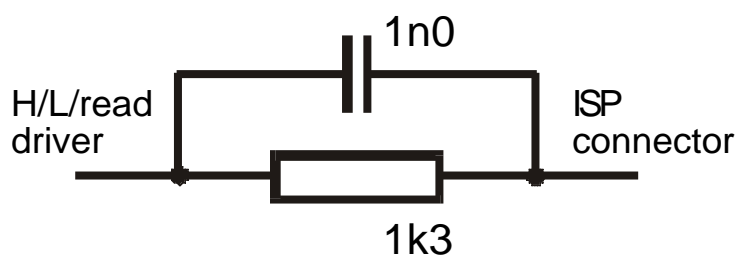


FIGURA 7: DRIVER ISP DELL'MP PIK+

La corrispondenza tra segnali ISP e pin del connettore può variare da dispositivo a dispositivo.  
 Per conoscerla, selezionare il dispositivo desiderato nel programma di controllo (menu Device / Select Device) e premere il pulsante "Additional Info" o premere Ctrl+F1.

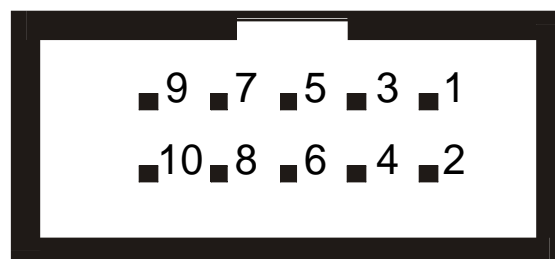


FIGURA 8: CONNETTORE ISP DELL'MP PIK+

## PROBLEMI DI COLLEGAMENTO MP PIK+ E PC

A causa della grande varietà di porte parallele esistenti, può capitare che il programmatore non riesca ad intendersi con il PC. Il problema può manifestarsi come comunicazione tra PC e programmatore indaffidabile o del tutto assente.

- Se si verifica questo comportamento, provate a collegare il programmatore ad altre porte parallele o ad altri PC.
- Ricordate che la porta parallela alla quale è connesso il programmatore **deve** essere configurata come compatibile IEEE 1284. Per verificare la configurazione il modo migliore è entrare nel BIOS del vostro computer e vedere se la porta è impostata come ECP/EPP. O, se usate una scheda multi I/O, assicuratevi che sia configurata come compatibile IEEE 1284.
- Provate a configurare la porta parallela come compatibile Centronics (o standard, o normale o SPP; a seconda del produttore del BIOS che state usando si trovano nomi diversi per indicare la stessa cosa). Questo portebbe diminuire le prestazione ma assicura la massima affidabilità nella comunicazione.

Se non trovate una soluzione, potete contattare la **grifo**® direttamente. Preparatevi a dare una descrizione precisa della configurazione del vostro PC ed una descrizione del problema che vi è capitato. Non va tralasciato il tipo di PC, il produttore, la velocità, il sistema operativo e la versione, programmi residenti e la configurazione della porta parallela.

## LEDS

**MP PIK+** è fornito di 2 LEDs per visualizzare lo stato dell'alimentazione e lo stato interno del programmatore. Riferitevi alla figura 11 per localizzare i LEDs.

LED	COLORE	DESCRIZIONE
Power LED	Verde	All'accensione si illumina per alcuni secondi, poi riduce l'intensità.
BUSY LED	Rosso	Quando acceso indica che il programmatore sta accedendo al dispositivo target in lettura o scrittura. Non toccare il dispositivo target quando questo LED è acceso.

**FIGURA 9: DESCRIZIONE LEDs DELL'MP PIK+**



## **SELFTEST E CALIBRAZIONE**

Se ritenete che il programmatore non reagisca secondo le vostre aspettative, potete lanciare il self test che fa uso del POD diagnostico incluso nella confezione. Per avere prestazioni ottimali vi suggeriamo di effettuare un self test ogni 6 mesi circa. Per ulteriori informazioni consultate il paragrafo “DIAGNOSTICS” nel capitolo “DESCRIZIONE SOFTWARE”.

## **INSTALLAZIONE DEL PROGRAMMA**

La confezione contiene un Mini-CD contenente il programma di controllo, diversi programmi di utilità ed ulteriori informazioni. Differenze e modifiche rispetto al presente manuale (se esistenti) si possono trovare nel file README\_P.TXT.

## **CONFIGURAZIONE MINIMA DEL PC**

- PC AT 10 MHz (versione DOS), PC 486 (versione WIN)
- 512 kB di RAM libera (versione DOS), 16MB di RAM libera (versione WIN)
- Un lettore CD
- HDD, 5 MB di spazio libero
- Sistema operativo MS/PC-DOS versione 3.2 o superiore, WIN 3.xx, WIN95/98
- Una porta per stampante libera con nessun dispositivo collegato

## **CONFIGURAZIONE CONSIGLIATA DEL PC**

- Pentium PC 100Mhz o superiore
- 32 MB di RAM libera
- Un lettore CD
- HDD, 10 MB di spazio libero almeno
- Sistema operativo: MS Windows 95/98/ME/NT/2000/XP
- Una porta parallela libera sul bus PCI, IEEE 1284 o pienamente compatibile

## **INSTALLAZIONE SOFTWARE**

### **WINDOWS 95/98/ME/NT/2000/XP**

Inserite il Mini-CD nel lettore CD, il programma si installerà automaticamente.

L'installatore (setup.exe) vi guiderà passo passo attraverso il processo di installazione ed eseguirà le operazioni necessarie precedenti alla prima esecuzione.

### **DOS**

Eseguite il programma **Install** dal CD.

Per le versioni DOS del programma di controllo eseguite sotto WINDOWS 3.11/95/98, suggeriamo le seguenti operazioni:

- Esecuzione del programma a schermo intero (in modo testo). In modo grafico il programma potrebbe essere più lento.
- Se il programma di controllo deve girare in **background** bisogna impostare: (WIN95/98) premendo il tasto destro del mouse sull'icona DOS scegliete Proprietà poi non spuntare la casella Altre/Sfondo/Sospendi sempre; (WIN3xx) nel file PIF appropriato, impostate l'opzione Background.

I programmi PG4U.EXE (per DOS) e PG4UW.EXE (per Windows) sono comuni a tutti i programmatori **grifo®**. Noi garantiamo la corretta esecuzione di questi programmi sotto tutti i sistemi operativi sopra menzionati senza problemi. Anche le operazioni in background sotto Windows sono prive di errori.

### NUOVE VERSIONI DEL PROGRAMMA DI CONTROLLO

Per ottenere il massimo dalle capacità del programmatore consigliamo di usare la versione più aggiornata di PG4U.EXE o PG4UW.EXE. Potete scaricare l'ultima versione del programma di controllo dai siti internet <http://www.grifo.com> o <http://www.grifo.it>.

### AGGIORNARE IL PROGRAMMA DI CONTROLLO

Dopo aver scaricato l'ultima versione del file, semplicemente clickateci sopra. Per il programma DOS è necessario anche clickare sul file "install.exe". Rispondete sempre "Yes" alle richieste di "Overwrite".

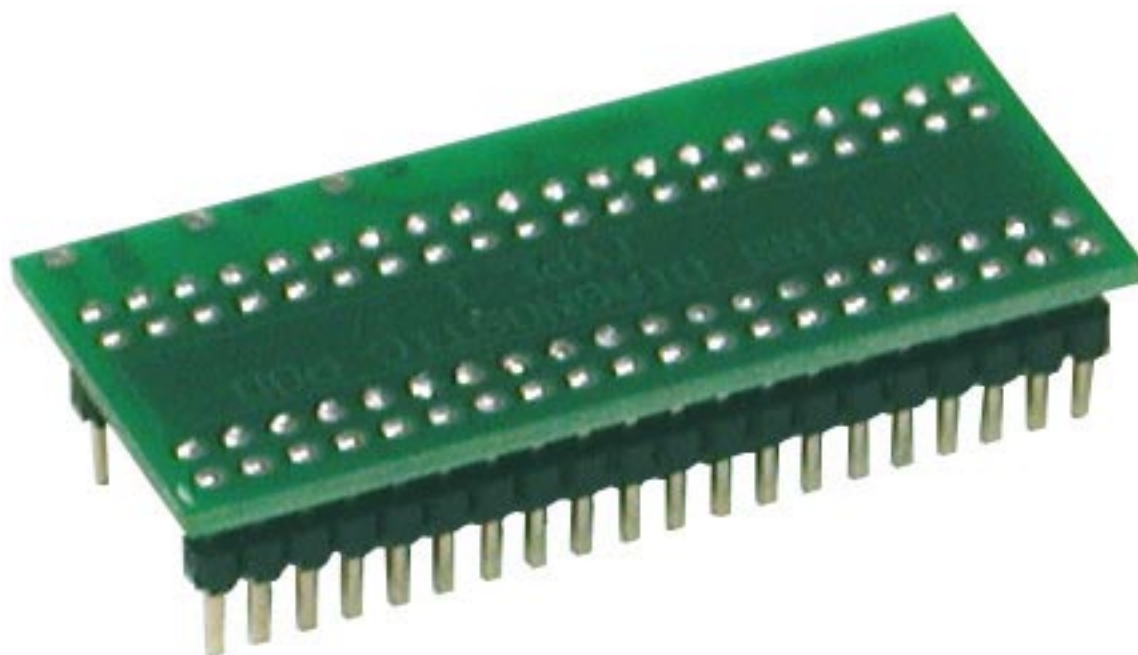


FIGURA 10: POD DEL SELFTEST

## AVVIO RAPIDO

Questi paragrafi contengono tutte le informazioni essenziali per la connessione del programmatore al PC, installare il programma di controllo ed iniziare ad usare il programmatore in brevissimo tempo. **Si prega di leggere completamente questi paragrafi prima di tentare qualunque utilizzo del programmatore.** Per ottenere informazioni più dettagliate sulle caratteristiche tecniche del programmatore e del software di controllo potete riferirvi ai successivi capitoli di questo manuale.

## INSTALLAZIONE HARDWARE

Riferendosi alla figura 11:

- Spegnete il PC ed il programmatore.
- Inserite il cavo di comunicazione fornito in una porta parallela libera del vostro PC.
- Collegate l'altra estremità al programmatore.
- Stringete le viti dei connettori, questo è molto importante.
- Collegate l'alimentatore fornito in dotazione alla presa di alimentazione di rete e collegate il connettore mini DIN alla presa di alimentazione sul programmatore.
- Accendete PC e programmatore.
- Appena collegata l'alimentazione tutti i LED si accendono simultaneamente, poi i tre LED di stato si spengono e si riaccendono in sequenza uno dopo l'altro, infine il LED di alimentazione si pone in bassa luminosità.

Anche se può essere scomodo dover ogni volta staccare il cavo della stampante e ricollegare quello del programmatore, non è possibile collegare il programmatore tramite un data switch, sia meccanico che elettronico. Potete comunque installare una scheda supplementare di I/O per dedicare una porta, esempio LPT2, al programmatore, mentre la stampante può rimanere sulla porta LPT1.

### **! ATTENZIONE !**

Se non volete spegnere il PC mentre collegate il programmatore, procedete come segue:

- Quando connettete il programmatore al PC: **PRIMA** inserite il cavo di comunicazione **POI** il connettore di alimentazione.
- Quando scollegate il programmatore dal PC: **PRIMA** scollegate il connettore di alimentazione **POI** il cavo di comunicazione.

## INSTALLAZIONE SOFTWARE

### **WINDOWS 95/98/ME/NT/2000/XP**

Inserite il Mini-CD nel lettore CD, il programma si installerà automaticamente.

L'installatore (setup.exe) vi guiderà passo passo attraverso il processo di installazione ed eseguirà le operazioni necessarie precedenti alla prima esecuzione.

### **DOS**

Eseguite il programma **Install** dal CD.

## PROGRAMMARE O COPIARE UN DISPOSITIVO

- Eseguite PG4U.EXE (DOS) o PG4UW.EXE (WIN) per entrare nel programma di controllo.
- Se l'hardware è stato collegato correttamente il programma dovrebbe mostrare una finestra per informare che ha trovato il programmatore, e nella schermata principale dovrebbe mostrare lo stato del programmatore nel rettangolo in alto a destra. Se questo non dovesse accadere, ripercorrete le istruzioni del paragrafo "INSTALLAZIONE HARDWARE" oppure consultate il capitolo "RISOLUZIONE DEI PROBLEMI".
- Scegliete il dispositivo che volete programmare sia tramite il menu Device/Select sia tramite la combinazione di tasti <Alt+F5> (premete <F5> mentre tenete il tasto Alt premuto).
- Caricate i dati con cui programmare il dispositivo selezionato:
  - § Se volete caricare un file da disco, usate il menu File/Load o premete il tasto <F3>. Se avete bisogno di un tipo specifico di file consultate il capitolo "DESCRIZIONE SOFTWARE".
  - § Se volete copiare il contenuto di un dispositivo già programmato, dovete prima leggere il suo contenuto tramite il menu Device/Read o premendo il tasto <F7>.
- Inserite il dispositivo da programmare nello zoccolo ZIF del programmatore e scrivete il contenuto attuale del buffer col menu Device/Program o premendo il tasto <F9>.
- Potete confrontare il contenuto attuale del buffer con i dati appena scritti nel dispositivo da programmare mediante il menu Device/Verify o premendo il tasto <F8>.

Il menu Device contiene i comandi per manipolare il dispositivo da programmare. Il menu File mette a disposizione i comandi di gestione files e directory. Il menu Buffer viene usato per la manipolazione del contenuto del buffer.

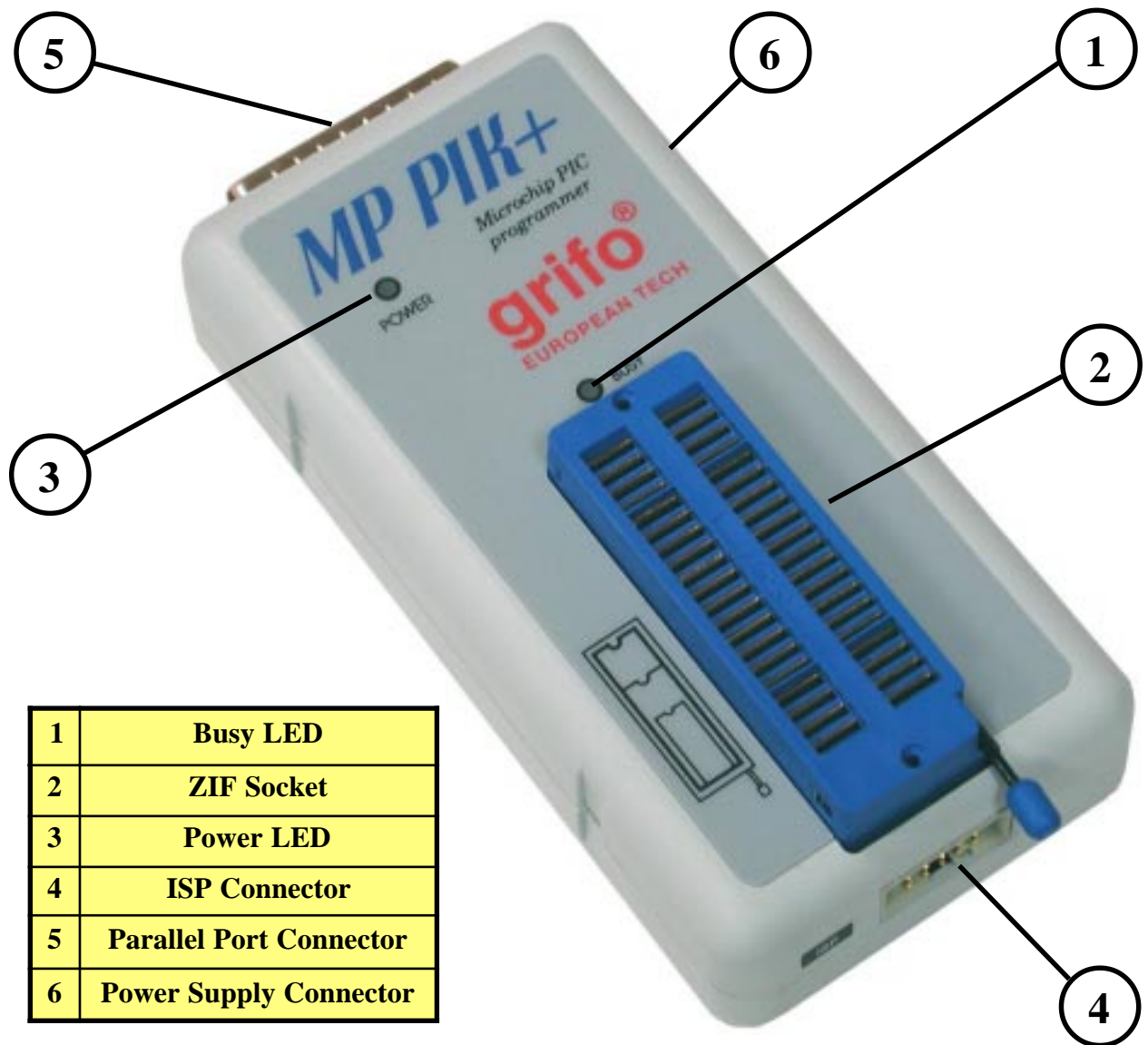


FIGURA 11: PIANTA DEGLI ELEMENTI

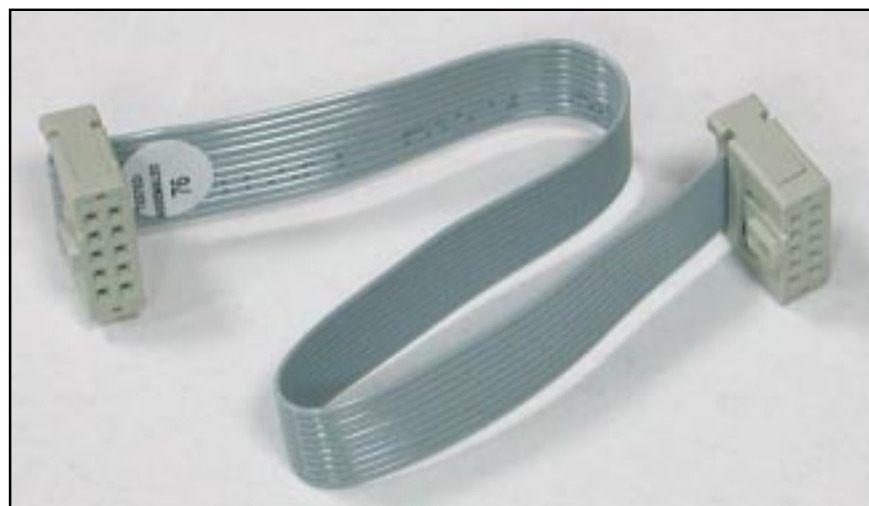


FIGURA 12: CAVO PROGRAMMAZIONE ISP IN DOTAZIONE

## DESCRIZIONE SOFTWARE

Questo capitolo contiene una descrizione essenziale del programma di controllo di tutti i programmatori. Si ricorda che l'help in linea del suddetto programma viene aggiornato prima del manuale.

### UTILIZZO DEL SOFTWARE

#### **! IMPORTANTE !**

Il programma di controllo fornito, contenuto nel Mini-CD incluso nella confezione, è garantito essere esente da qualunque virus al momento della consegna. Per incrementare la sicurezza i nostri programmi implementano un algoritmo speciale per rilevare possibili infezioni da virus.

### **ESEGUIRE IL PROGRAMMA DI CONTROLLO**

Per eseguire il programma di controllo da linea di comando di MS/PC DOS scrivete: PG4U  
Mentre sotto Windows fate doppio click sull'icona di PG4UW.

Dopo la partenza, PG4U/PG4UW scandisce tutte le porte esistenti per trovarci un programmatore **grifo**<sup>®</sup> collegato. PG4U/PG4UW è il programma comune a tutti i programmatori **grifo**<sup>®</sup> per cui cercherà di trovare tutti i tipi.

**NOTA:** *Una volta partito PG4U/PG4UW, verifica la propria integrità. Infine il programma mostra il menu utente standard e attende istruzioni.*

Se il programma di controllo non riesce a comunicare col programmatore, appare sullo schermo un messaggio di errore che fornisce un codice di errore ed una descrizione delle possibili cause (programmatore scollegato, cattiva connessione, problemi di alimentazione, porta parallela incompatibile, ...).

Eliminate la causa dell'errore e premete un tasto qualunque.

Se la condizione di errore persiste, il programma entra in modalità demo e impedisce l'accesso al programmatore. Se non riuscite ad identificare la causa dell'errore, seguite le istruzioni nel capitolo "RISOLUZIONE DEGLI ERRORI".

Inoltre, il programma di controllo verifica la comunicazione col programmatore prima di fare qualunque altra operazione col dispositivo da programmare.

### **DESCRIZIONE DELLA SCHERMATA PRINCIPALE**

barra del **TITOLO** - mostra avvisi di copyright, nome e versione di PG4U/PG4UW

barra dei **MENU** - lista delle funzioni fondamentali

riquadro **FILE** - informazioni sul contenuto corrente del buffer

riquadro **STATUS** - informazioni sullo stato interno di PG4U/PG4UW

riquadro **ADDRESSES** - organizzazione, dimensione, indirizzi di inizio e fine del dispositivo target, del buffer e del file

riquadro **DEVICE** - informazioni principali sull'attuale dispositivo target

barra di **HELP** - breve descrizione del comando selezionato.

La selezione del **MENU** avviene nel tipico modo delle GUI - cioè muovendo il cursore e premendo <ENTER> o digitando la lettera evidenziata nel menu selezionato o, naturalmente, usando il mouse. Esistono anche delle associazioni tra tasti e i comandi di uso più frequente.

### **! ATTENZIONE !**

I dati inseriti da tastiera sono in formato esadecimale, tranne i blocchi di caratteri ASCII nei menu Buffer View/Edit.

### **LISTA DEI TASTI SPECIALI**

<F1>	HELP	Help
<F2>	SAVE	Salva un file
<F3>	LOAD	Carica un file nel buffer
<F4>	EDIT	Esame/Modifica del buffer
<F5>	SELECT/default	Scelta del dispositivo target tra gli ultimi 10 dispositivi selezionati
<Alt+F5>	SELECT/manual	Scelta del dispositivo target digitando il nome o quello del produttore
<Ctrl+F5>	SELECT/type	Scelta del dispositivo target dalla famiglia
<Shift+F5>	SELECT/vendor	Scelta del dispositivo target dalla produttore
<F6>	BLANK	Blank check
<F7>	READ	Copia nel buffer il contenuto del dispositivo target
<F8>	VERIFY	Confronta il contenuto del buffer con il contenuto del dispositivo target
<F9>	PROGRAM	Programma il dispositivo target
<Alt+Q>	QUIT	Termina il PG4U/PG4UW
<Alt+X>	QUIT & save	Termina il PG4U/PG4UW e salva le impostazioni
<Ctrl+F1>		Mostra informazioni aggiuntive sul dispositivo target corrente
<Ctrl+F2>	ERASE	Riempie il buffer con un valore assegnato
<Ctrl+Shift+F2>		Riempie il buffer con un valori casuali

## COMANDI DEL PROGRAMMA DI CONTROLLO

La seguente lista contiene i comandi disponibili con la versione Windows del programma di controllo. Accanto al nome del comando è presente, se esiste, la combinazione di tasti che lo richiama rapidamente e una breve descrizione del suo scopo e modalità d'uso. Si ricorda che l'aggiornamento più recente delle spiegazioni di questi comandi è sempre presente nell'help in linea sensibile al contesto del programma di controllo, che quindi andrebbe consultato prima del presente manuale.

### **File**

Viene usato per gestire i files sorgenti, esaminare le cartelle su disco, cambiare drive e modificare gli indirizzi di inizio e di fine nel salvataggio e caricamento dei file di tipo **binario**, **MOTOROLA**, **MOS Technology** e **Intel (extended) HEX**.

#### **File / Load**

Analizza il formato del file e carica i dati nel buffer. Si può selezionare il tipo di file desiderato (**binario**, **MOTOROLA**, **MOS Technology**, **Tektronix**, **Intel (extended) HEX**, **ASCII space**, **JEDEC** e **POF**). Il programma di controllo memorizza l'ultimo tipo di file selezionato. Per rendere la memorizzazione persistente, salvare la configurazione con **Options / Save options**.

Spuntando la casella **Automatic file format recognition** il programma riconosce automaticamente il formato. Se il formato non può essere riconosciuto, il file viene trattato come binario.

Togliendo lo spunto alla casella **Automatic file format recognition** il programma permette all'utente di specificare il tipo di file desiderato nella lista della casella di riepilogo **Selected file format**. Si può specificare l'indirizzo di inizio nel buffer con il formato binario. Tale indirizzo indica da quale posizione nel buffer saranno scritti i dati letti dal file.

Il tasto <F3> farà apparire questo menu da qualunque posizioni in qualunque momento.

#### **File / Save**

Salva su disco i dati contenuti nel buffer, creati, modificati o letti da un dispositivo. Si può selezionare il tipo di file desiderato (**binario**, **MOTOROLA**, **MOS Technology**, **Tektronix and Intel (extended) HEX**).

Il tasto <F2> farà apparire questo menu da qualunque posizioni in qualunque momento.

#### **File / Load project**

Carica un file di progetto, che contiene la configurazione per i dispositivi da usare e la configurazione dell'interfaccia utente.

La finestra standard **Load project** contiene un'altra finestra - **Project description** - nella parte bassa. Quest'ultima mostra le informazioni contenute nel file selezionato al momento dalla finestra Load project.



Le informazioni contenute in un progetto sono:

- nome e produttore del primo dispositivo selezionato nel progetto
- data e ora di creazione del progetto
- versione del programma con cui il progetto è stato creato
- descrizione testuale del progetto redatta dall'utente (si tratta di testo a piacere, di solito il nome degli autori ed alcune note)

## File / Save project

Quest'opzione serve a salvare il file di progetto, il quale contiene i dati di configurazione del dispositivo salvati. I dati di un progetto salvato possono essere richiamati in ogni momento col comando **File / Load project**.

La finestra **Save project** contiene tre finestre aggiuntive nel pannello **Project description** posto nella parte bassa della finestra **Save project**. Le finestre mostrano le informazioni contenute nel file di progetto al momento selezionato dalla finestra **Save project** e che stanno per essere salvate su disco.

Le informazioni contenute in un progetto sono:

- nome e produttore del primo dispositivo selezionato nel progetto
- data e ora di creazione del progetto
- versione del programma con cui il progetto è stato creato
- descrizione testuale del progetto redatta dall'utente (si tratta di testo a piacere, di solito il nome degli autori ed alcune note)

La prima finestra (più in alto) contiene informazioni sul progetto al momento selezionato dalla finestra **Save project**.

La seconda finestra (a metà) mostra informazioni sull'attuale configurazione del programma compresi i dispositivi attualmente selezionati, lo stato di attività del programmatore, la data, l'ora e la versione del programma. Questi dati vengono inseriti nella descrizione del progetto.

La terza finestra (in basso) è modificabile dall'utente e contiene una descrizione del progetto (testo a piacere), di solito il nome degli autori ed alcune note.

## File / Reload file

Questa opzione permette di ricaricare un file usato di recente.

Un file, dopo essere stato aperto, viene aggiunto alla lista chiamata **Reload file**. I nomi dei files sono mostrati nell'ordine con cui sono stati aperti, ovvero in cima si trovano quelli usati di recente.

Per ricaricare un file:

1. Dal menu File, scegliere **Reload file**.
2. Viene mostrata la lista dei files usati di recente, clickare sul nome del file desiderato.

**Nota:** *Quando un file viene ricaricato, si assume che sia del formato con cui è stato salvato l'ultima volta.*

## File / Reload project

Questa opzione permette di ricaricare un progetto usato di recente.

Un progetto, dopo essere stato aperto, viene aggiunto alla lista chiamata **Reload project**. I nomi dei progetti sono mostrati nell'ordine con cui sono stati aperti, ovvero in cima si trovano quelli usati di recente.

Per ricaricare un file:

1. Dal menu File, scegliere **Reload project**.
2. Viene mostrata la lista dei progetti usati di recente, clickare sul nome del file desiderato.

## File / Project options

Questa opzione viene usate per esaminare/modificare le informazioni del progetto attualmente caricato. Queste comprendono una descrizione essenziale del progetto ed i seguenti dati:

- nome dispositivo e nome produttore
- data creazione progetto
- versione del programma sotto cui il progetto è stato creato
- definizione del progetto fornita dall'utente (testo a piacere), ad esempio i nomi degli autori ed ulteriori informazioni che descrivono il progetto

L'utente può modificare a piacimento il solo testo di descrizione e definizione del progetto. Le altre informazioni sono aggiornate automaticamente dal programma.

## File / Load encryption table

Questo comanda carica i dati da un file binario su disco e li salva nella parte della memoria riservata alla tavola di criptazione ( di sicurezza).

## File / Save encryption table

Questo comanda salva i dati in un file binario su disco prendendoli dalla parte della memoria riservata alla tavola di criptazione ( di sicurezza).

## File / Exit without save

Questo comando dealloca lo heap, cancella il buffer su disco (se esiste) e restituisce le risorse al sistema operativo.

## File / Exit and save

Questo comando dealloca lo heap, cancella il buffer su disco (se esiste), salva nel file di configurazione le impostazioni relative agli ultimi 10 dispositivi usati e restituisce le risorse al sistema operativo.

## Device

Questo menu contiene le funzioni che servono per selezionare e lavorare con un dispositivo programmabile, ovvero selezione, programmazione, verifica e cancellazione.

### Device / Select from default devices

Questa finestra permette di selezionare il dispositivo desiderato da una lista di dispositivi di default. La lista è ciclica e contiene gli ultimi 10 dispositivi selezionati comprese le opzioni di configurazione. La lista può essere salvata col comando **File / Exit and save**.

Se desiderate visualizzare ulteriori informazioni sul dispositivo corrente, premete la combinazione di tasti **<Ctrl+F1>**. Apparirà una finestra che mostra la dimensione del dispositivo, la sua organizzazione, l'algoritmo di programmazione e una lista di programmatori che lo supportano, compresi gli eventuali moduli. Vi sono anche altre informazioni specifiche.

Usate il tasto **<Del>** per cancellare il dispositivo corrente dalla lista. Non è possibile svuotare completamente la lista. Sull'ultimo dispositivo il tasto **<Del>** viene ignorato.

### Device / Select device ...

Questa finestra permette di selezionare il dispositivo desiderato dalla lista di quelli supportati dal programmatore attivo. I criteri sono nome (**name**), tipo (**type**) o produttore (**manufacturer**).

Il dispositivo selezionato viene salvato automaticamente nella lista dei 10 dispositivi di default raggiungibile mediante il comando **Device / Select from default devices**.

Se desiderate visualizzare ulteriori informazioni sul dispositivo corrente, premete la combinazione di tasti **<Ctrl+F1>**. Apparirà una finestra che mostra la dimensione del dispositivo, la sua organizzazione, l'algoritmo di programmazione e una lista di programmatori che lo supportano, compresi gli eventuali moduli. Vi sono anche altre informazioni specifiche.

### Select device ... / All

Questa finestra permette di selezionare il dispositivo desiderato dalla lista completa dei dispositivi supportati dal programmatore attivo. I nomi vengono mostrati in una casella a scorrimento.

Il dispositivo può essere selezionato con un doppio click sulla lista o inserendo manualmente parte del nome del dispositivo e/o del produttore (usate **<Space>** come separatore) e premete **<Enter>** o cliccate il pulsante **OK**.

Premete **<Esc>** o cliccate il pulsante **Cancel** per annullare la selezione e rimanere all'ultimo dispositivo selezionato.

Il dispositivo selezionato viene salvato automaticamente nella lista dei 10 dispositivi di default raggiungibile mediante il comando **Device / Select from default devices**.

Se desiderate visualizzare ulteriori informazioni sul dispositivo corrente, premete la combinazione di tasti **<Ctrl+F1>**. Apparirà una finestra che mostra la dimensione del dispositivo, la sua organizzazione, l'algoritmo di programmazione e una lista di programmatori che lo supportano, compresi gli eventuali moduli. Vi sono anche altre informazioni specifiche.

### **Select device ... / Only selected type**

Questa finestra permette di selezionare il tipo di dispositivo desiderato (ad esempio EPROM) ed il sottotipo (esempio 64Kx8 (27512)), tramite il mouse o i tasti cursore. Verrà visualizzata una lista di produttori e dispositivi.

Il dispositivo può essere selezionato con un doppio click sulla lista o inserendo manualmente parte del nome del dispositivo e/o del produttore (usate **<Space>** come separatore) e premete **<Enter>** o clickate il pulsante **OK**.

Premete **<Esc>** o clickate il pulsante **Cancel** per annullare la selezione e rimanere all'ultimo dispositivo selezionato.

Il dispositivo selezionato viene salvato automaticamente nella lista dei 10 dispositivi di default raggiungibile mediante il comando **Device / Select from default devices**.

Se desiderate visualizzare ulteriori informazioni sul dispositivo corrente, premete la combinazione di tasti **<Ctrl+F1>**. Apparirà una finestra che mostra la dimensione del dispositivo, la sua organizzazione, l'algoritmo di programmazione e una lista di programmatori che lo supportano, compresi gli eventuali moduli. Vi sono anche altre informazioni specifiche.

### **Select device ... / Only selected manufacturer**

Questa finestra permette di selezionare il dispositivo desiderato di un particolare produttore, tramite il mouse o i tasti cursore. Verrà visualizzata una lista di produttori e dispositivi.

Il dispositivo può essere selezionato con un doppio click sulla lista o inserendo manualmente parte del nome del dispositivo e/o del produttore (usate **<Space>** come separatore) e premete **<Enter>** o clickate il pulsante **OK**.

Premete **<Esc>** o clickate il pulsante **Cancel** per annullare la selezione e rimanere all'ultimo dispositivo selezionato.

Il dispositivo selezionato viene salvato automaticamente nella lista dei 10 dispositivi di default raggiungibile mediante il comando **Device / Select from default devices**.

Se desiderate visualizzare ulteriori informazioni sul dispositivo corrente, premete la combinazione di tasti **<Ctrl+F1>**. Apparirà una finestra che mostra la dimensione del dispositivo, la sua organizzazione, l'algoritmo di programmazione e una lista di programmatori che lo supportano, compresi gli eventuali moduli. Vi sono anche altre informazioni specifiche.

## **Device / Select EPROM /Flash by ID**

Si usa questo comando per selezionare automaticamente una EPROM o una Flash leggendo l'ID del dispositivo. Il programmatore può identificare il dispositivo leggendo l'ID del produttore e

del dispositivo stesso “marchiato” dentro il chip. Questa possibilità esiste solo per EPROM o Flash che la supportano. Se il dispositivo non supporta questo metodo di riconoscimento, un messaggio che indica un dispositivo sconosciuto o non supportato verrà mostrato.

Se più di un dispositivo dello stesso costruttore condividono lo stesso ID verrà mostrata la lista contutti i loro nomi. Quello desiderato può essere selezionato posizionandosi sul suo nome e premendo <Enter> (o clickando il pulsante **OK**). Premete <Esc> o clickate il pulsante **Cancel** per annullare la selezione e rimanere all'ultimo dispositivo selezionato.

**ATTENZIONE:** *Il programma di controllo supporta solo EPROM e Flash con 28 e 32 piedini. Tutti i programmatori determinano automaticamente il numero dei piedini. Con altri programmatori dovrete inserire manualmente il numero dei piedini..*

Il programmatore applica una tensione elevata agli appositi contatti dello zoccolo. Questo è necessario per attivare il sistema di lettura degli ID. Non inserite nello zoccolo un dispositivo diverso da una EPROM o da una Flash, potrebbe essere danneggiato dalla tensione elevata.

Vienesconsigliato di applicare questo comando ad EPROM tipo 2764 e 27128 perchè quasi tutte non supportano la lettura dell'ID.

## Device / Device options

Questo menu imposta parametri di programmazione, serializzazione e gestione dei file associati.

### Device / Device options / Operation options

Le impostazioni di questo comando sono usate per controllare il processo di programmazione. Questo è un ambiente flessibile che contiene impostazioni associate al dispositivo selezionato, quelle non supportate dal programmatore attivo sono disabilitate. Le impostazioni vengono salvate su disco eseguendo il comando **File / Exit and save**.

Lista delle impostazioni:

#### - gruppo **Addresses:**

indirizzo iniziale del dispositivo	(default 0)
indirizzo finale del dispositivo	(default dimensione-1)
indirizzo iniziale del buffer	(default 0)

#### - gruppo **Insertion test:**

verifica di inserzione	(default abilitato, cioè ENABLE)
verifica dei bytes di ID	(default abilitato, cioè ENABLE)

#### - gruppo **Command execution:**

verifica se è vuoto	(default disabilitato, cioè DISABLE)
cancella prima di programmare	(default disabilitato, cioè DISABLE)
verifica dopo la lettura	(default disabilitato, cioè DISABLE)
verifica dopo la programmazione	(una volta: ONCE, due volte: TWICE)
opzioni verifica	(VCC nominale e 5%, VCC nominale e 10%, VCCmin VCCmax)

### Device / Device options / Serialization

La serializzazione è una particolare modalità di programmazione. Quando è attivata, un particolare codice numerico viene automaticamente scritto ad un indirizzo specificato del buffer prima di ogni programmazione. Quando molti dispositivi vengono programmati singolarmente, il numero di serie viene cambiato automaticamente prima della programmazione, così ogni dispositivo ha un numero di serie univoco.

Ci sono due tipi di serializzazione:

- Incrementale
- Da disco

Quando viene selezionato un nuovo dispositivo la serializzazione viene automaticamente disabilitata. Gli attuali parametri di serializzazione per il dispositivo corrente vengono salvati nella configurazione generale con il comando **File / Exit and save**.

Quando è abilitato il modo incrementale le seguenti configurazioni vengono salvate su file: indirizzo, dimensione, numero seriale, passo dell'incremento e le impostazioni di ASCII / BIN, DEC / HEX, LS / MS Byte .

Quando è abilitato il prelievo dei numeri da file le seguenti configurazioni vengono salvate: nome del file dei numeri ed etichetta corrente, quest'ultima indica la linea del file dalla quale verrà preso il prossimo numero.

Quando è in corso la multiprogrammazione (è stato selezionato un modulo multi-zoccolo) la sezione speciale - chiamata **Action on not programmed serial values due to error** - viene mostrata nella finestra **Serialization**. Qui sono possibili due scelte:

1. Ignorare i numeri non programmati
2. Aggiungere al file i numeri non programmati

**Ignore not programmed serial values** significa che i numeri non ancora programmati vengono ignorati e non vengono coinvolti in nessuna azione.

**Add not programmed serial values to file** significa che i numeri di serie non programmati vengono aggiunti al file. Quest'ultimo ha lo stesso formato testuale del file usato per la serializzazione "da file" per cui è possibile riutilizzarlo in seguito con la normale modalità di serializzazione "da file".

Se la programmazione del dispositivo viene interrotta dall'utente, non ci saranno conseguenze sui numeri pronti per il lotto successivo di dispositivi. La stessa situazione si verifica se la programmazione rimane incompleta, ad esempio per una inserzione errata.

Ignorare o aggiungere i numeri non programmati ha senso solo se almeno un dispositivo del lotto corrente sul modulo multi-zoccolo viene completamente programmato e supera la verifica senza errori.

### Device / Device options / Serialization / Incremental mode

La modalità incrementale (**Incremental mode**) permette di assegnare numeri di serie individuali ad ogni dispositivo che viene programmato. Un valore iniziale viene incrementato ad ogni dispositivo programmato correttamente e scritto nel formato selezionato all'indirizzo del buffer desiderato.

L'utente può modificare le seguenti opzioni in modalità incrementale:

#### S / N size

definisce la lunghezza in bytes del numero di serie che verrà scritto nel buffer. Per serializzazioni binarie i valori validi sono da 1 a 4, mentre per le serializzazioni in ASCII i valori validi sono da 1 ad 8.

#### Address

Specifica l'indirizzo nel buffer dove il numero di serie viene scritto. Si noti che tale valore deve essere compreso nel campo di indirizzi tra quello iniziale e quello finale e deve essere specificato correttamente, ovvero l'intero numero di serie deve ricadere entro il campo di indirizzi validi.

#### Start value

Specifica il valore iniziale dei numeri di serie. Generalmente il valore massimo per serializzazioni binarie con 4 bytes è \$1FFFFFFF.

Quando il valore del numero di serie supera il massimo vengono azzerati i tre bit più significativi del numero di serie corrente. Dopo tale azione il numero si trova ad essere sempre compreso tra 0 e \$1FFFFFFF (gestione degli overflow secondo lo stile del basic).

#### Step

Specifica il passo di incremento del numero di serie.

#### S / N mode

Definisce la forma con la quale il numero di serie viene scritto:

- ASCII
- Binario

**ASCII** - Significa che il numero verrà scritto come stringa ASCII. Per esempio il numero di serie \$0528CD in ASCII viene scritto nel buffer come la sequenza di bytes 30h 35h 32h 38h 43h 44h ('0' '5' '2' '8' 'C' 'D'), ovvero 6 bytes.

**Bin** - Significa che il numero di serie viene scritto direttamente nel buffer. Se il numero è composto da più di un byte, ci sono due possibili ordini di scrittura. L'ordine di scrittura può essere cambiato con il comando "Save to buffer" item.

#### Style

Definisce la base per i numeri di serie, che può essere:

- Decimale
- Esadecimale.

I numeri decimali (**Decimal**) vengono inseriti e mostrati usando i caratteri da '0' a '9'.

I numeri esadecimali (**Hexadecimal**) vengono inseriti e mostrati usando anche i caratteri da 'A' ad 'F'.

Vi è il caso speciali dei numeri decimali codificati in binario, o BCD. Ciò significa che ogni cifra decimale viene memorizzato in una cifra esadecimale, ovvero ogni nibble vale da 0 a 9, non sono ammessi i valori da A ad F in un nibble BCD.

La base va scelta con l'opzione "Style" prima di inserire il numero di serie iniziale ed il passo.

### Save to buffer

Specifica l'ordine di scrittura dei bytes del numero di serie. Questa opzione viene usata quando il numero di serie è di tipo binario, non si applica ai numeri ASCII.

Vi sono due opzioni:

- LSByte first (usato dai processori Intel) posizionerà il byte meno significativo all'indirizzo più basso del buffer.
- MSByte first (usato dai processori Motorola) posizionerà il byte più significativo all'indirizzo più basso del buffer.

## Device / Device options / Serialization / From file mode

In modalità di serializzazione dal file i numeri vengono letti da un file specificato dell'utente e scritti nell'indirizzo del buffer indicato dal file.

Ci sono due opzioni: Nome file e Etichetta di partenza.

### File name

Specifica il nome del file da cui vengono letti gli indirizzi ed i relativi numeri di serie. Il file per la serializzazione in questa modalità deve avere un formato specifico, che viene descritto nella sezione più sotto.

### Start label

Definisce l'etichetta iniziale all'interno del file di numeri. La lettura dei valori dei numeri di serie inizia da questa etichetta.

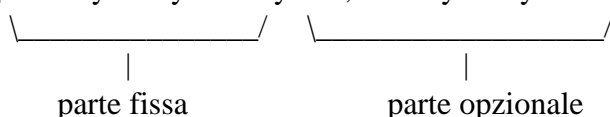
### From file serialization file format

Il file per la serializzazione da file comprende indirizzi e numeri di serie, questi ultimi specificati come gruppi di bytes. Il file è di tipo testuale, la sua struttura è:

[label $l$ ] addr byte $0$  byte $1$  .. byte $n$

...

[label $n$ ] addr byte $0$  byte $1$  .. byte $m$  , addr byte $0$  byte $1$  ... byte $k$



; Commento

legenda:

parte fissa

La parte fissa definisce l'indirizzo nel buffer ed i bytes che vanno scritti a partire da tale indirizzo. Tali dati vanno sempre scritti dopo l'etichetta.

parte opzionale

La parte opzionale definisce un secondo gruppo di indirizzo e bytes da scrivere nel buffer. Una parte opzionale può essere definita dopo la parte fissa dei dati.

etichette (label $l$ , label $n$  - labels)

Le etichette identificano ogni linea del file in ingresso. Vengono usate per indirizzare individualmente ogni riga del file, per cui ogni etichetta dovrebbe essere univoca. Lo scopo è indirizzare la linea da cui inizia il conteggio dei numeri di serie



**addr -**

Definisce l'indirizzo nel buffer da cui iniziare a scrivere il numero di serie.

byte0..byten, byte0..bytem, byte0..bytek -

I gruppi di bytes byte0..byten, byte0..bytem e byte0..bytek sono i dati che verranno scritti nel buffer. Al massimo ci possono essere 64 bytes in un campo dati seguente ad un indirizzo.

I dati vengono scritti nel buffer dall'indirizzo addr all'indirizzo addr+n:

byte0 in addr

byte1 in addr + 1

byte2 in addr + 2

....

byten in addr + n

La parte opzionale viene separata dal carattere “ , “ (virgola) ed ha la stessa struttura della parte precedente, ad esempio un indirizzo ed un gruppo di bytes.

**Caratteri di uso speciale:**

[ ] - le etichette devono essere racchiuse da parentesi quadre

‘,’ - carattere separatore la parte fissa e quella opzionale

‘;’ - il carattere di punto e virgola indica l'inizio di un commento. Tutti i caratteri dal punto e virgola alla fine della linea vengono ignorati. Un commento può anche essere a fine linea.

**Note:**

- Un nome di etichetta può contenere tutti i caratteri tranne '[' e ']'. I nomi di etichetta non tengono conto della capitalizzazione, ovvero 'a' è uguale ad 'A', 'b' è uguale a 'B' ecc..

- Tutti gli indirizzi ed i valori da scrivere devono essere in esadecimale.

- Gli indirizzi possono essere composta da 1 fino a 4 bytes.

- I gruppi di bytes di dati possono essere composti da 1 a 64 elementi. Quando ci sono due gruppi su una stessa linea, il numero totale di elementi deve essere al massimo 80.

- Fate attenzione ad impostare l'indirizzo corretto. Questo deve essere definito nel campo tra gli indirizzi iniziale e finale del dispositivo che viene programmato. Se l'indirizzo fosse fuori da questo campo apparirebbe una finestra e si la serializzazione verrebbe disabilitata.

**Esempio:**

[nav1] A7890 78 89 56 02 AB CD ; commento 1

[nav2] A7890 02 02 04 06 08 0A

[nav3] A7890 08 09 0A 0B A0 C0 ; commento 2

[nav4] A7890 68 87 50 02 0B 8D

[nav5] A7890 A8 88 59 02 AB 7D

;la linea successiva contiene anche la parte opzionale

[nav6] A7890 18 29 36 42 5B 6D , FFFF6 44 11 22 33 99 88 77 66 55 16

; ultima linea - fine del file

Nel file di esempio vengono definiti sei valori di numeri di serie etichettati con “nav1”, “nav2”, ...“nav6” . ogni numero viene scritto all'indirizzo \$A7890 ed è composto da 6 bytes. La linea etichettata con “nav6” ha anche la parte opzionale, che viene scritta nel buffer dall'indirizzo \$FFFF6 ed è di 10 bytes, quindi il suo ultimo byte viene scritto all'indirizzo \$FFFFF.

## Device / Device options / Statistics

Le statistiche forniscono informazioni circa l'attuale conteggio delle operazioni su un particolare dispositivo. Se su ogni dispositivi viene effettuata una sola operazione allora il conteggio delle operazioni effettuate coincide con il conteggio dei dispositivi programmati.

Un'altra funzione statistica è il conto alla rovescia (**Count down**). Questa permette di controllare sia il numero di operazioni che il numero di dispositivi sui quali vengono effettuate. Dopo ogni operazione andata a buon fine su un certo dispositivo il contatore viene decrementato. L'utente deve stabilire il valore iniziale del conto alla rovescia, quando questo raggiunge lo zero si intende che è stato raggiunto il limite del lotto e viene mostrato un messaggio di conteggio completato.

La finestra **Statistics** contiene le seguenti opzioni:

Le caselle spuntabili **Program**, **Verify**, **Blank**, **Erase** e **Read** decidono le operazioni che, se andate buon fine, incrementano il conteggio.

Le caselle spuntabili **Count Down** gestisce l'attività di conto alla rovescia (lo abilita o disabilita). La casella di testo accanto alla casella spuntabile definisce il valore iniziale del conteggio.

La finestra **Statistics** si può aprire anche premendo il tasto destro del mouse sul pannello Statistics e cliccando sulla voce visualizzata.

I valori delle statistiche vengono mostrati nella finestra principale, nel pannello Statistics.

Quest'ultimo contiene le informazioni statistiche – **Success**, **Failure**, **Total** (ovvero programmazioni a buon fine, fallite e totali) e sul conto alla rovescia - **Count down** e **Remains** (ovvero stato del conteggio e pezzi rimanenti).

Il significato delle voci è:

<b>Success</b>	numero di operazioni completate con successo
<b>Failure</b>	numero di operazioni non completate con successo
<b>Total</b>	numero totale di operazioni
<b>Count down</b>	stato del conteggio (abilitato o disabilitato)
<b>Remains</b>	oggetti ancora da scalare nel conteggio alla rovescia

Le operazioni completate con successo sono una qualunque tra le seguenti:

- programmazione
- verifica di programmazione
- verifica se è vuoto
- cancellazione
- lettura

Se una operazione viene completata con errori non la si considera completata con successo.

Quando viene selezionato un nuovo dispositivo, tutti i valori delle statistiche vengono azzerati e **Count down** viene disabilitato (**Disabled**).

Il pulsante **Reset** nel pannello **Statistics** azzeri i valori delle statistiche.

Il pulsante **Reload Count down** nel pannello **Statistics** ricarica il valore iniziale di **Count down**.

## Device / Device options / Associated file

Associa un file al dispositivo corrente. Il file viene caricato automaticamente nel buffer non appena il dispositivo corrente viene di nuovo selezionato dalla lista dei dispositivi o se era già selezionato all'avvio del programma.

Il nome (completo di path) viene modificato nell'apposita casella di testo. Il programma controlla automaticamente la presenza del file, il caricamento si può abilitare o disabilitare.

Entrambi questi settaggi (nome del file ed abilitazione del caricamento) si possono salvare su disco con il comando **File / Exit and save**.

## Device / Blank check

Verifica se un dispositivo è vuoto, completamente o in parte, a seconda del dispositivo. Il risultato dell'operazione viene comunicato da una finestra di avviso.

Il comando **Device / Device options / Operation options** permette di modificare l'area standard del dispositivo che viene verificata da questo comando.

## Device / Read

Legge il contenuto di un dispositivo, completamente o in parte, a seconda del dispositivo. Il termine dell'operazione viene comunicato da una finestra di avviso.

Il comando **Device / Device options / Operation options** permette di modificare l'area del dispositivo che viene letta da questo comando. Abilitare la verifica dopo la lettura (comando **Verify data after reading**) in questo menu aumenta l'affidabilità della lettura stessa.

## Device / Verify

Confronta il contenuto di un dispositivo con il buffer, completamente o in parte, a seconda del dispositivo. Il risultato dell'operazione viene comunicato da una finestra di avviso.

Il comando **Device / Device options / Operation options** permette di modificare l'area sulla quale lavora questo comando.

Il comando **Options / Display errors** permette di scrivere le eventuali differenze sullo schermo o sul file VERIFY.ERR. Sullo schermo vengono mostrate al massimo le prime 45 differenze complete degli indirizzi.

## Device / Program

Programma il contenuto di un dispositivo, completamente o in parte, a seconda del dispositivo. L'esito dell'operazione viene comunicato da una finestra di avviso.

Il comando **Device / Device options / Operation options** permette di specificare un'altra area di lavoro e di impostare ulteriori elementi della sequenza di operazioni di programmazione.

## Device / Erase

Cancella il contenuto di un dispositivo. L'esito dell'operazione viene comunicato da una finestra di avviso..

## Device / Test

Esegue il test dei dispositivi indicati nella lista di quelli che possono ammettere un test (ad esempio RAM statiche) col il programmatore attuale.

## Device / Device info

Fornisce informazioni aggiuntive sul dispositivo attuale (dimensione, organizzazione, algoritmo di programmazione e una lista dei programmatori che lo supportano, corredata degli eventuali moduli adattatori). Si trovano inoltre informazioni sul package ed eventuali ulteriori informazioni specifiche.

La combinazione <Ctrl+F1> farà apparire questo menù immediatamente da ogni posizione.

## Buffer

Permette di manipolare il buffer compiendo operazioni su blocchi di memoria, riempiendone una parte con una stringa, cancellandolo, calcolando varie checksum e naturalmente visualizzando e modificando ogni singolo valore (inoltre si può cercare una stringa, stampare il buffer, etc.).

### Buffer / View/Edit

Visualizza (View) o modifica (Edit) ogni singolo dato nel buffer (visualizza solo in DUMP mode). Usate i tasti cursore per selezionare il dato da modificare, un dato modificato viene mostrato con colore diverso.

Potete usare anche il tasto <F4>.

View/Edit Buffer

- |                      |  |
|----------------------|--|
| <b>F1</b>            | mostra la schermata di aiuto   |
| <b>F2</b>            | riempie il buffer dall'indirizzo iniziale a quello finale con la stringa (ASCII o esadecimale) specificata.  |
| <b>Ctrl+F2</b>       | cancella il buffer riempiendolo col valore specificato   |
| <b>Ctrl+Shift+F2</b> | riempie il buffer con valori casuali   |
| <b>F3</b>            | copia una sequenza di bytes da un indirizzo iniziale ad uno finale a partire da un terzo indirizzo specificati dall'utente. Il terzo indirizzo può essere incluso tra i primi due.   |
| <b>F4</b>            | sposta una sequenza di bytes da un indirizzo iniziale ad uno finale a partire da un terzo indirizzo specificati dall'utente. Il terzo indirizzo può essere incluso tra i primi due.  |
| <b>F5</b>            | scambia i bytes alti con i byte bassi nel campo di indirizzi specificato. L'indirizzo iniziale deve essere pari ed il numero di bytes compresi deve essere pari. Se questa condizione non viene rispettata, il programma prende come indirizzo iniziale il primo pari precedente a quello specificato e come indirizzo finale il primo pari successivo a quello specificato. |
| <b>F6</b>            | stampa il contenuto del buffer   |

<b>F7</b>	stringa da trovare (massimo 16 caratteri ASCII)
<b>F8</b>	stringa da trovare e da sostituire (massimo 16 caratteri ASCII)
<b>F9</b>	vai ad un altro indirizzo
<b>F10</b>	commuta tra visualizza e modifica
<b>F11</b>	commuta tra 8 e 16 bit. Può essere fatto anche con il mouse clickando il pulsante a destra di View/Edit. Il pulsante indica anche il modo di visualizzazione attuale (8 o 16 bit).
<b>F12</b>	calcola i vari checksum disponibili
<b>Arrow keys</b>	spostano il cursore
<b>Home/End</b>	salta all'inizio / alla fine della linea corrente
<b>PgUp/PgDn</b>	salta alla pagina precedente / successiva
<b>Ctrl+PgUp/PgDn</b>	salta all'inizio / alla fine della pagina corrente
<b>Ctrl+Home/End</b>	salta all'inizio / alla fine del contenuto del dispositivo attuale
<b>Backspace</b>	sposta il cursore indietro di una posizione (a sinistra)

**Nota:** i caratteri 20H - FFH (ASCII) ed i numeri 0..9, A..F (esadecimali) modificano immediatamente il contenuto del buffer.

**Attenzione:** Non si possono gestire in ASCII i dati con dispositivi a 16 bit.

### Print buffer

Invia la parte selezionata del buffer su stampante o su un file. Il programma utilizza un editor di testi esterno nel quale viene visualizzata la parte selezionata del buffer e dal quale questa può essere salvata o stampata. Di default viene usato il semplice editor di testi **Notepad.exe**, il quale viene fornito con qualunque versione di Windows.

In questa finestra le opzioni sono:

- Definisci l'indirizzo iniziale del blocco selezionato nel buffer.
- Definisci l'indirizzo finale del blocco selezionato nel buffer.

### External editor

Definisce il nome completo di percorso del programam esterno che deve essere usato come visualizzatore di testo per il blocco selezionato del buffer. Per default viene indicato il Notepad standard di Windows. L'utente può inserire un visualizzatore a propria scelta, per esempio Wordpad.exe, capace di gestire file di testo grandi.

Il nome completo di percorso viene salvato su disco automaticamente.

### Find dialog box

Inserite la stringa di ricerca da trovare (**Find**) e premete il pulsante <**Find**> per iniziare la ricerca o scegliete il pulsante <**Cancel**> per lasciare perdere.

**Direction** (direzione) specifica in quale direzione deve avvenire la ricerca. **Forward** (in avanti) a partire dalla posizione corrente del cursore è il default. **Backward** (all'indietro) oltrepassa l'inizio del buffer. In visualizzazione cerca in tutto il buffer.

**Origin** specifica da dove deve iniziare la ricerca.

### Find & Replace dialog box

Inserite la stringa da cercare nella casella di testo **Text to find** e la stringa con cui verrà sostituita nella casella **Replace with**.

In **Options** potete scegliere se la sostituzione avverrà automaticamente o se volete essere avvertiti prima che avvenga per poterla annullare.

**Origin** indica dove deve iniziare la ricerca.

**Direction** (direzione) specifica in quale direzione deve avvenire la ricerca. **Forward** (in avanti) a partire dalla posizione corrente del cursore è il default. **Backward** (all'indietro) oltrepassa l'inizio del buffer. In visualizzazione cerca in tutto il buffer.

Premete <Esc> o clickate sul pulsante **Cancel** per chiudere la finestra.

Premendo il pulsante **Replace** la finestra viene chiusa ed appare un questionario che contiene le seguenti scelte:

<b>Yes</b>	sostituisci l'elemento trovato e cerca il prossimo
<b>No</b>	non sostituire l'elemento trovato e cerca il prossimo
<b>Replace All</b>	sostituisci tutto
<b>Abort search</b>	cancella questa operazione
View/Edit buffer per PLD	
<b>Ctrl+F2</b>	cancella il buffer riempiendolo col valore specificato
<b>Ctrl+Shift+F2</b>	riempi il buffer di valori casuali
<b>F9</b>	vai all'indirizzo specificato
<b>F10</b>	cambia modalità view / edit
<b>F11</b>	commuta tra 1 e 8 bit. Può essere fatto anche con il mouse clickando il pulsante a destra di View/Edit. Il pulsante indica anche il modo di visualizzazione attuale (1 o 8 bit).
<b>Arrow keys</b>	spostano il cursore
<b>Home/End</b>	salta all'inizio / alla fine della linea corrente
<b>PgUp/PgDn</b>	salta alla pagina precedente / successiva
<b>Ctrl+PgUp/PgDn</b>	salta all'inizio / alla fine della pagina corrente
<b>Ctrl+Home/End</b>	salta all'inizio / alla fine del contenuto del dispositivo attuale
<b>Backspace</b>	sposta il cursore indietro di una posizione (a sinistra)

**Note:** I caratteri 0 ed 1 cambiano immediatamente i valori.

### Buffer / Fill block

Riempie il blocco selezionato del buffer con il valore specificato. Indicare l'indirizzo iniziale e finale del blocco ed il valore di riempimento.

### Buffer / Copy block

Copia il blocco selezionato del buffer a partire dall'indirizzo specificato. Indicare l'indirizzo iniziale e finale del blocco ed l'indirizzo di destinazione, anche compreso tra i primi due.

### Buffer / Move block

Sposta il blocco selezionato del buffer a partire dall'indirizzo specificato. Indicare l'indirizzo iniziale e finale del blocco ed l'indirizzo di destinazione, anche compreso tra i primi due. Il blocco iniziale verrà riempito con il carattere di blank.

### Buffer / Swap block

Scambia i bytes alti con i byte bassi nel campo di indirizzi specificato. L'indirizzo iniziale deve essere pari ed il numero di bytes compresi deve essere pari. Se questa condizione non viene rispettata, il programma prende come indirizzo iniziale il primo pari precedente a quello specificato e come indirizzo finale il primo pari successivo a quello specificato.

## Buffer / Erase

Riempie il buffer con il carattere di blank specificato dall'utente.

La combinazione <Ctrl+F2> farà apparire questo menu in ogni momento.

## Buffer / Fill random data

Riempie il contenuto del buffer con valori casuali.

La combinazione <Shift+Ctrl+F2> farà apparire questo menu in ogni momento.

## Buffer / Duplicate buffer

Duplica il contenuto del buffer da una EPROM in una EPROM destinazione più ampia. Questa procedura può, ad esempio, copiare in una 27C512 il contenuto di una 27C256 nella posizione desiderata.

**Nota:** L'indirizzo iniziale del buffer è sempre 00000h.

## Buffer / Checksum

Calcola il checksum nel blocco del buffer selezionato. Il checksum viene calcolato con i seguenti algoritmi :

<b>Byte</b>	somma di bytes ad una "word". Il flag CY viene ignorato
<b>Word</b>	somma di word ad una "word". Il flag CY viene ignorato
<b>Byte (CY)</b>	somma di bytes ad una "word". Il flag CY viene aggiunto al risultato
<b>Word (CY)</b>	somma di word ad una "word". Il flag CY viene aggiunto al risultato
<b>CRC-CCITT</b>	somma bytes a "word" con $VAL=PRECEDENTE+(x^{16} + x^{12} + x^5 + 1)$
<b>CRC-XModem</b>	somma bytes a "word" con $VAL=PRECEDENTE+(x^{16} + x^{15} + x^2 + 1)$

La colonna **Neg.** è il complemento a uno della somma cosicché  $Somma + Neg. = FFFFH$ .

La colonna **Suppl.** è il complemento a due della somma cosicché  $Somma + Suppl. = 0 (+ carry)$ .

La finestra di checksum contiene i seguenti oggetti:

**From address:** Indirizzo del primo byte del blocco selezionato sul quale viene calcolato il checksum.

**To address:** Indirizzo dell'ultimo byte blocco del selezionato sul quale viene calcolato il checksum.

**Insert checksum:** Permette di selezionare il tipo di checksum verrà scritta nel buffer all'esecuzione di **Calculate & insert**.

**Insert address:** Permette di impostare la dimensione della checksum calcolata che verrà scritta nel buffer all'esecuzione di **Calculate & insert**. L'indirizzo non si può specificare come un intervallo tipo <From address> fino a <To address> ma come indirizzo di Byte.

**Size:** Determina la dimensione della checksum che verrà scritta nel buffer. Tale dimensione può essere 8 (byte) o 16 (word) bits. Se viene selezionato 16 l'intera checksum verrà scritta nel buffer, altrimenti verrà scritto solo il byte basso.

**Nota:** *Se viene selezionata 16 come dimensione verrà scritto nel buffer il byte basso della checksum all'indirizzo specificato mentre il byte alto verrà scritto all'indirizzo successivo.*

**Calculate:** Clickando su questo pulsante si effettua il calcolo della checksum per il blocco selezionato nel buffer. Non viene scritto nulla dentro il buffer.

**Calculate & insert:** Clickando sul pulsante **Calculate & insert** si effettua il calcolo della checksum per il blocco selezionato nel buffer. Il valore viene scritto nulla dentro il buffer all'indirizzo specificato da **Insert address**.

## Options

Permette di verificare e modificare diverse impostazioni di default.

### Options / General options

Permette di controllare le seguenti opzioni.

#### File options

Permette di impostare la maschera per i file, attivare il caricamento automatico del file selezionato e selezionare il tipo del file che verrà caricato.

**File format masks** imposta un filtro per selezionare i nomi dei file in **File / Save** e **File / Load file** per tutti i formati. La maschera deve contenere almeno un carattere speciale (\*, ?) .

**Project file default extension** imposta l'estensione del file usata di default in **File / Load project** e **File / Save project**.

Il gruppo **When current file is modified by another process** permette di decidere la politica di ri-caricamento del file attualmente selezionato. Ci sono tre opzioni:

1. Chiedi prima di ricaricare
2. Ricarica automaticamente
3. Ignora i cambiamenti del file

**Load file format** permette di decidere la politica di riconoscimento dei file che vengono caricati. Quando viene messo in automatico, il programma analizza il formato cercando ogni tipo di file supportato. Se il file corrisponde allo schema di validazioni di un formato supportato, verrà caricato nel buffer usando quel formato.

Se messo in manuale l'utente può indicare esplicitamente il formato del file che sta per essere caricato dalla lista dei formati supportati. In caso il formato indicato non corrisponda a quello effettivo del file, quest'ultimo potrà essere caricato parzialmente o non correttamente.

#### Hex file options

Contiene diverse opzioni per controllare il caricamento dei file in formato HEX.

La prima opzione abilita la cancellazione (**erasing**) del buffer (col valore desiderato) che avviene automaticamente prima di caricare il file.



La seconda opzione imposta un offset negativo (**negative offset**) usato per modificare l'indirizzo di caricamento di ogni file HEX in modo che ricada in un campo di indirizzi validi.

**Esempio:**

Un file contiene dati in formato Motorola S. Un blocco di dati inizia dall'indirizzo FFFF0H. Il formato è S2 quindi l'indirizzo è un gruppo di 3 bytes. Potete associare un offset negativo di FFFF0H alla lettura di ogni dato. Quindi l'offset verrà sottratto dall'effettivo indirizzo specificato nel file e i dati verranno caricati a partire dall'indirizzo 0.

**Avvertimento:** *Il valore di offset viene sottratto all'indirizzo effettivo quindi il risultato potrebbe essere un indirizzo minore di 0. Fate molta attenzione ad impostare l'offset negativo.*

La terza opzione regola il riposizionamento (**redirecting**) dei dati da file in formato Intel HEX esteso al buffer. Quindi all'utente verrà chiesto di inserire un valore di segmento, che verrà associato al più basso segmento del file (record tipo 02) e gli altri segmenti verranno cambiati tenendo conto di questa differenza. L'opzione non ha effetto per i file Intel HEX non di tipo esteso, di default è disabilitata.

**Esempio:**

Il file contiene record di tipo 02 con indirizzi F000H e F800H. Inserendo un nuovo valore di segmento, per esempio 0H, i dati del segmento F000H verranno memorizzati nel segmento 0H e analogamente i dati nel segmento F800H andranno nel segmento 800H.

Di default non è attiva.

**Language**

Permette di selezionare un'altra lingua per l'interfaccia utente ovvero menu, pulsanti, finestre, informazioni e messaggi. Permette anche di avere l'help in un'altra lingua. Per il supporto a questa funzione è necessario che ci sia un file di definizione della lingua.

**Sound**

Permette all'utente di associare degli effetti sonori al completamento di determinate operazioni come, ad esempio, programmazione, verifica, lettura, etc. Il programma genera un suono anche quando viene mostrato un avvertimento o un errore. L'utente può selezionare un suono dal sistema audio di MS Windows (richiede la corretta installazione di una scheda audio), dallo speaker del PC o disabilitare i suoni.

**Other**

Permette di impostare ulteriori parametri del programma.

Si possono disabilitare gli hint (suggerimenti) che appaiono sui pulsanti delle barre posizionandoci sopra la freccia per alcuni secondi infine si può modificare la directory di partenza all'avvio del programma, a scelta tra la directory di installazione e l'ultima directory visitata prima di chiudere il programma.

**Save options**

Permette di decidere cosa salva il programma alla chiusura:

**Don't save options** - Non salvare le opzioni e non chiedere se salvare le opzioni.

**Auto save options** - Salva le opzioni senza chiedere conferma.

**Prompt for save options** - Chiede all'utente se vanno o non vanno salvate le opzioni.

## Options / View

Seleziona se mostrare o nascondere vari elementi dell'interfaccia utente come le barre.

## Options / View / Main toolbar

Mostra o nasconde la barra dei pulsanti principale.

## Options / View / Additional toolbar

Mostra o nasconde la barra dei pulsanti secondaria.

## Options / View / Programming conditions before programming

Mostra o nasconde l'elenco delle opzioni di programmazione per un certo dispositivo prima che la programmazione inizi.

## Options / Display errors

Permette di decidere la modalità con cui gli errori vengono comunicati. Questi ultimi possono essere mostrati sullo schermo (**screen**) (al massimo 45 differenze durante la verifica di programmazione), salvati (**saved**) su un file che verrà chiamato **VERIFY.ERR** e creato nella directory corrente o non visualizzati (**not displayed**). Se la visualizzazione degli errori è disabilitata, verrà mostrata solo una finestra di INFO.

Queste impostazioni si possono salvare su disco con il comando **Options / Save options**. Di default gli errori vengono mostrati sullo schermo.

## Options / Find programmer

Cerca un altro programmatore o ripristina la comunicazione con il programmatore già collegato. Offre le seguenti opzioni:

**Programmer** - imposta uno tra i programmatori supportati dal programma di controllo. Se viene selezionata la voce "Search all" verranno cercati tutti.

**Establish communication** - stabilisce la comunicazione con un nuovo programmatore o ripristina la comunicazione con quello già collegato.

**Speed** - imposta la velocità di comunicazione, se è stata impostata la modalità di comunicazione manuale. La velocità viene espressa in percentuale della massima possibile.

Modificare la velocità di comunicazione è importante con i PC che hanno porte parallele "lente" o che non hanno sufficiente potenza per pilotare il cavo di comunicazione PC<->programmer cable (laptop, notebook, ...). Usare questa opzione per tentare di risolvere eventuali problemi di comunicazione tra PC e programmatore (esempio, il programma di controllo indica che il programmatore è assente la comunicazione col programmatore non è affidabile, ecc.).

Se è stata abilitata la comunicazione automatica allora il programma di controllo imposta la massima velocità possibile.

**Port** - sceglie la porta parallela dove verrà cercato il programmatore. Se la voce "All ports" viene selezionata allora il programmatore verrà cercato su tutte le porte presenti agli indirizzi standard.

**Address for special port** - Imposta l'indirizzo per la porta parallela special , se selezionata.

Premendo <Enter> o il pulsante **OK** parte la scansione basata sui precedenti parametri. La stessa scansione avviene all'avvio del programma di controllo. Il comando cancella la lista dei dispositivi di default ma non il dispositivo corrente se il nuovo programmatore lo supporta.

Queste impostazioni vengono salvate col comando **Options / Save options**.

## Options / Handler

La finestra **Handler** permette di definire un dispositivo esterno che viene abilitato ad effettuare un controllo speciale controllare delle attività effettuate sui dispositivi programmabili. Quando non viene selezionato alcun Handler il programma di controllo funziona come da default, ovvero l'utente controlla direttamente le attività, altrimenti il programma di controllo si trova in modalità speciale, quindi le operazioni sui dispositivi sono controllate in co-operazione con un dispositivo esterno.

La finestra **Handler** contiene le seguenti opzioni:

<b>Selected Handler</b>	selezione l'Handler desiderato.
<b>Search at port</b>	seleziona una porta seriale (COM) dove verrà cercato l'Handler.

Premendo <Enter> o il pulsante **OK** si attiva la ricerca. Se l'Handler selezionato è di tipo **None**, non verrà effettuata alcuna ricerca. Le attuali impostazioni sull'Handler vengono salvate su disco mediante il comando **Options / Save options** o quando il programma di controllo viene chiuso.

L'Handler non è disponibile per le vendite.

## Options / Module options

Viene usato con un modulo multi-zoccolo per definire lo zoccolo MASTER e il tipo di attività per ogni zoccolo. Il gruppo di opzioni **MASTER socket** permette di stabilire quale zoccolo viene usato per le operazioni di lettura. Le caselle sputabili **Enable/Disable socket** abilitano o disabilitano individualmente gli zocchi. Lo zoccolo disabilitato non effettua alcuna operazione.

## Options / Automatic YES!

Imposta la modalità **Automatic YES!**. In tale modalità basta inserire correttamente un dispositivo nello zoccolo e l'ultima operazione effettuata verrà ripetuta. Automaticamente, il programma rileva l'inserzione di un nuovo dispositivo nello zoccolo ed esegue l'ultima operazione effettuata senza dover premere alcun tasto o pulsante. Viene visualizzata sullo schermo l'avvenuta inserzione di un nuovo dispositivo. La modalità può essere annullata premendo il tasto <Esc> durante l'attesa di inserzione di un nuovo dispositivo.

**Nota:** Durante l'attesa di un nuovo dispositivo il LED BUSY lampeggia.

Questa modalità può essere abilitata o disabilitata con l'oggetto **Automatic YES!**. Se viene selezionato un nuovo programmatore con **Options / Find programmer**, questa modalità viene disabilitata.

Con **Response time** è possibile impostare un intervallo di tempo entro il quale deve essere inserito il dispositivo nuovo. DI default viene preimpostato un intervallo standard. Se viene usato uno zoccolo adattatore di formato, si consiglia di aumentare tale intervallo.

Con **Pins with capacitors** si può specificare una lista di pins che vengono connessi attraverso delle capacità (ad esempio se viene usato uno zoccolo convertitore con un condensatore inserito tra Vcc e GND), che potrebbe creare problemi durante l'inserzione di un nuovo dispositivo.

Tale lista deve essere nella forma:

pinA, pinB, pinC....

**Esempio:** 4,6,17

La lista viene svuotata se viene eseguito il comando **Device / Select default** or **Device / Select device ...**

Queste impostazioni vengono salvate sul disco con il comando **Options / Save options**.

## Options / Log file

Questa opzione viene usata in congiunzione con **Log window**. Tutti i rapporti inviati alla predetta finestra si possono inviare anche ad un file (**Log file**). Tale file ha il nome **REPORT.REP** e viene creato nella directory corrente.

Impostando **New** si provoca la cancellazione di un file esistente e la creazione di un nuovo file. Con **Append** si aggiungono i nuovi rapporti a quelli già presenti nel file. Se il file non esiste, verrà creato automaticamente.

Queste impostazioni vengono salvate sul disco con il comando **Options / Save options**. Di default i rapporti vanno allo schermo senza che vengano memorizzati nel file..

## Options / Protected mode

La modalità protetta è una speciale modalità del programmatore. In questa situazione sono disabilitate le operazioni di programmazione, quelle che possono modificare il contenuto del buffer o che possono variare la configurazione del dispositivo da programmare. Quindi in questa situazioni l'operatore non può modificare i parametri significativi della programmazione per errore, risulta utile nella programmazione di grandi lotti.

Ci sono due modi per attivare la modalità protetta:

1. con il comando **Options / Protected mode**. Questo comando richiede la password, l'utente la deve inserire due volte per dare conferma. Dopo l'accettazione della password avviene il passaggio in modalità protetta, la stessa password serve per tornare in modalità normale.
2. aprendo un progetto precedentemente salvato in modalità protetta.

Per tornare in modalità normale usate il comando **Options / Normal mode**. La password che verrà richiesta è la stessa usate per commutare in modalità protetta.

Un altro modo per uscire dalla modalità protetta consiste nel chiudere il programma, in quanto la modalità protetta è attiva solo fino all'uscita dal software. Al prossimo avvio il programma si troverà di nuovo in modalità normale (unica eccezione è il caso di un progetto caricato dalla linea di comando precedentemente salvato in modalità protetta).

### **Options / Save options**

Questo comando salva tutte le impostazioni al momento abilitate ad essere salvate, persino se il salvataggio automatico viene disabilitato. Sono salvate le seguenti opzioni: quelle del menu **Options**, gli ultimi dieci dispositivi selezionati, gli ultimi file aperti, posizione e dimensione della finestra principale.

### **Diagnostics**

Questo comando attiva la diagnostica interna e i test sugli integrati logici.

#### **Diagnostics / Selftest**

Esegue una procedura di auto-diagnostica senza il relativo POD (presente nella confezione). **LO ZOCCOLO ZIF DEVE ESSERE VUOTO PRIMA DI ESEGUIRE QUESTO COMANDO.** Si consiglia di eseguire anche l'auto-diagnostica **Diagnostics / Selftest plus**.

#### **Diagnostics / Selftest plus**

Esegue una procedura di auto-diagnostica usando il relativo POD (presente nella confezione). Si consiglia di eseguire questo comando il più frequentemente possibile, ad esempio una volta al mese.

#### **Diagnostics / IC test**

Attiva la sezione di test per circuiti logici che si trova separate sul mini-CD di distribuzione per compatibilità. Prima si deve selezionare la libreria appropriata, poi il dispositivo desiderato, infine una modalità del test tra continua e a passo singolo (**Loop, Single step**). La sequenza di controllo e il risultato del test vengono mostrati in finestra.

#### **Diagnostics / Create diagnostic report**

Crea il report diagnostico contenente particolari informazioni interna e lo scrive nella finestra principale, poi copia il contenuto della finestra nel clipboard di Windows. Dalla clipboard il testo contenuto può essere copiato in un qualunque editor di testi. Il report diagnostico è utile quando si verifica un qualche errore nel programma di controllo o nel programmatore che l'utente non è in grado di rimediare in autonomia, quindi deve contattare il produttore. In tal caso è indispensabile allegare al messaggio per il produttore anche il report diagnostico, in quanto questo può aiutare il produttore a localizzare più rapidamente la causa del problema e a risolverlo.

## Help

Premendo <F1> si accede all'Help. Premendo <F1> mentre è selezionato un particolare elemento di un menu, si accede all'help sensibile al contesto. Se PG4U / PG4UW sta eseguendo una qualche operazione sul programmatore <F1> non genera alcuna reazione.

Sono abilitati i seguenti elementi dell'Help:

- testo che descrive gli elementi richiesti all'Help
  - tutte le altre parole significative in current cross-references
- clickate sul cross-reference per ottenere ulteriori informazioni.

Poichè il sistema di Help viene aggiornato continuamente con il programma di controllo, può contenere informazioni non incluse in questo manuale.

Informazioni dettagliate sui comandi individuali si possono trovare nell'Help in linea.

**Nota:** *Le informazioni sono precise al momento della pubblicazione dello stesso. I nostri prodotti sono aggiornati di continuo. Si prega di consultare il sito [www.grifo.it](http://www.grifo.it) o [www.grifo.com](http://www.grifo.com).*

### Help / Supported devices

Mostra una lista con tutti i dispositivi supportati da almeno un programmatore.

Torna utile quando l'utente vuole trovare un qualunque dispositivo purchè sia supportat da almeno uno di quelli compatibili con il programma di controllo.

Il prefisso "g\_" prima del nome indica che il dispositivo è supportato dal modulo multi-zoccolo.

### Help / Supported programmers

Mostra informazioni sui programmatori supportati dal programma di controllo.

### Help / Device list (current programmer)

Crea una lista di tutti i dispositivi supportati dal programmatore corrente nel file di testo **?????DEV.txt** e nel file HTML **?????DEV.htm** che vengono creati nella directory dalla quale viene eseguito il programma di controllo. Il segno **?????** viene sostituito con una forma abbreviata del nome del programmatore al quale la lista è riferita.

### Help / Device list (cross reference)

Crea una cross reference tra tutti i dispositivi supportati da tutti i programmatori presenti sul mercato e quelli supportati dal programmatore corrente. Il risultato è in formato HTML e consiste dei seguenti file:

- un file principale in HTML **TOP\_DEV.htm** che lista i produttori dei dispositivi
- file HTML parziali che listano i dispositivi per ogni produttore

Il file HTML principale viene creato nella stessa directory ove si trova il programma di controllo. I file parziali HTML vengono messi nelle sotto directory **DEV\_HTML** che partono dalla directory dove si trova il programma di controllo.

## About

Fa apparire una finestra che mostra le informazioni di copyright ed il numero di versione.

## CARATTERISTICHE SOFTWARE AGGIUNTIVE

PG4U/PG4UW è il programma di controllo comune a tutti i programmatori **grifo**®. Quindi potreste trovare alcune voci di menu che non si riferiscono al vostro programmatore.

Si può associare un file particolare ad ogni dispositivo, in modo che venga caricato automaticamente nel buffer se l'opzione **Device options/Associated file/Automatic load** è impostata a Yes. Il file associato è l'ultimo caricato, a meno che non venga esplicitamente dichiarato nella casella di testo **Device options/Associated file/File name**.

Una caratteristica denominata **Automatic YES!** permette in concreto, di ripetere l'ultima azione eseguita senza bisogno di premere un pulsante o un tasto. Questo è possibile poichè il programmatore determina autonomamente l'inserimento di un nuovo dispositivo nello zoccolo e ripete subito l'ultima operazione. Tutti i parametri di questa caratteristica sono impostabili da menu **Options**. Se usate adattatori di formato che connettono dei condensatori ai pin, inserite tali pin nella apposita lista.

Alcuni dispositivi speciale (ad esempio i PHILIPS della famiglia Coolrunner) richiedono dei file DAT esterni non forniti di default nel mini-CD. Se avete bisogno di programmarli, esaminate la sezione Download dei siti internet **www.grifo.it** o **www.grifo.com**.

## ULTERIORI INFORMAZIONI

Fate attenzione con i sistemi operativi multitasking (WIN 3.11/95/98/Me/NT/2000/XP). Per un corretto funzionamento del programma di controllo è indispensabile che la porta parallela a cui è collegato il programmatore venga riservata ad esso.

Il programma di controllo è in grado di gestire tutti i tipi di porta parallela (pieno supporto IEEE 1284), quindi non c'è bisogno di configurare la porta parallela in modalità particolari.

Windows 98 ha un'impostazione nel file MSDOS.SYS che costringe le applicazioni DOS a funzionare più lentamente. Per rimediare, impostate la variabile DoubleBuffer=0.

Sotto Windows, siete pregati di non spostare mai alcuna finestra mentre il BUSY LED è acceso, il circuito di protezione del programmatore contro errori di comunicazione o eccessiva lentezza del PC potrebbe attivarsi e mettere il programmatore in modalità di sicurezza.

## TERMINI DELLA GARANZIA

La garanzia copre il programmatore e tutte le sue parti, purchè esenti da difetti di fabbricazione, i materiali e la mano d'opera fino a tre anni dalla data di acquisto. La garanzia è altresì limitata a 25000 cicli per lo zoccolo ZIF. Se il prodotto viene diagnosticato come difettoso, il centro autorizzato di riparazioni riparerà o sostituirà le parti difettose gratuitamente. Le parti sostituite e/o l'intero programmatore saranno coperti solo per il rimanente periodo di garanzia.

Per riparazioni entro il periodo di garanzia, il cliente deve dimostrare la data d'acquisto. Tutti gli altri possessori di programmatori usufruiranno di soli 6 mesi di garanzia.

La garanzia non copre danni prodotti da usura, lacerazioni o danneggiamenti meccanici. Allo stesso modo non copre prodotti manipolati e/o riparati e/o aperti da personale non autorizzato dalla **grifo®**, o prodotti che sono stati danneggiati durante un trasporto oppure usati e/o installati e/o maneggiati impropriamente.

Prima di inviare il materiale in riparazione, si prega di contattare la **grifo®**. In ogni caso, se il materiale giunge senza descrizione del problema la riparazione può essere respinta. Analogamente, non si può garantire la riparazione se non vengono inviati anche tutti gli accessori esterni usati quando è stata riscontrato il difetto.

Richieste di riparazioni non necessarie o fuori garanzia verranno addebitate. La **grifo®** od i suoi distributori determineranno se il prodotto difettoso debba essere riparato o sostituito e giudicheranno se le condizioni di garanzia sono da applicarsi oppure no.

Si veda inoltre il capitolo “RISOLUZIONE DEI PROBLEMI”.



## RISOLUZIONE DEI PROBLEMI

Il prodotto ed il materiale impiegato per costruirlo sono di ottima qualità. Nonostante ciò, se doveste sperimentare dei problemi, rileggete attentamente la documentazione contenuta in questo manuale, probabilmente vi troverete la risposta cercata. Se così non fosse, seguite le istruzioni qui sotto.

### ERRORI DI COMUNICAZIONE

- Il programmatore deve essere alimentato correttamente, il LED verde di alimentazione deve essere sempre acceso, l'alimentatore deve essere quello fornito in dotazione. Per verificare l'integrità fisica del cavo di alimentazione provate a piegarlo leggermente dove si collega al programmatore e vedete se succede qualcosa.
- Aggiornate il programma di controllo, spesso una versione più recente migliora le caratteristiche di comunicazione. Consultate il capitolo "NUOVE VERSIONI DEL PROGRAMMA DI CONTROLLO" per ulteriori informazioni.
- Il programmatore va collegato direttamente alla porta parallela, non ci devono essere data switch nè chiavi hardware nè replicatori di porte tra PC e programmatore.
- La porta parallela va dedicata totalmente al programmatore, sotto Windows non va condivisa in rete nè usata per una stampante.
- Ricordate che la porta parallela alla quale è connesso il programmatore **deve** essere configurata come compatibile IEEE 1284. Per verificare la configurazione il modo migliore è entrare nel BIOS del vostro computer e vedere se la porta è impostata come ECP/EPP. O, se usate una scheda multi I/O, assicuratevi che sia configurata come compatibile IEEE 1284.
- Il cavo di comunicazione parallela deve essere quello fornito nella confezione oppure uno **con tutti i piedini collegati**. Per verificare l'integrità fisica del cavo di comunicazione provate a piegarlo leggermente dove si collega al programmatore mentre PG4U/PG4UW tenta di riconoscere il programmatore e vedete se succede qualcosa.
- Provate ad installare il programmatore ed il PG4U/PG4UW su un altro computer. Se questi funzionano bene sull'altro computer potreste avere dei problemi col primo PC. Confrontate le differenze tra questi due computers.
- Provate a configurare la porta parallela come compatibile Centronics (o standard, o normale o SPP; a seconda del produttore del BIOS che state usando si trovano nomi diversi per indicare la stessa cosa). Questo potrebbe diminuire le prestazioni ma assicura la massima affidabilità nella comunicazione.

### PROBLEMI DI LETTURA O PROGRAMMAZIONE

- Assicuratevi che il dispositivo target sia allineato correttamente nello zoccolo. Consultate la serigrafia accanto allo zoccolo o sull'adattatore se ne state usando uno.
- Il dispositivo di target deve essere selezionato su PG4U/PG4UW prima di tentare di leggerlo o programmarlo.
- Il nome del dispositivo selezionato ed il nome del dispositivo inserito nello zoccolo devono essere **ESATTAMENTE GLI STESSI!** Se tentate di operare su un dispositivo dal nome diverso o anche solo simile potreste danneggiare sia il programmatore sia il dispositivo.
- Aggiornate il programma di controllo. I produttori di elettronica creano nuovi dispositivi quasi ogni giorno per cui la cosa migliore è cercare il dispositivo che serve nell'ultima versione del PG4U/PG4UW. Se non lo trovate consultate il paragrafo "DISPOSITIVI NON SUPPORTATI".

- Se il dispositivo è vergine o è stato usato e poi cancellato si ritrova privo di contenuto. Effettuate un “Blank check test” tramite il menu Device/Blank Check o premendo il tasto <F6> per accertarvi che il dispositivo sia leggibile ma vuoto.
- **MP PIK+** può effettuare un test di auto diagnostica per verificare il corretto funzionamento dei componenti interni. Selezionate il menu Diagnostic e seguite le istruzioni a schermo. Tenete presente che il selftest deve essere eseguito con lo **zoccolo ZIF vuoto**, altrimenti sia il programmatore sia il dispositivo che vi si trova potrebbero essere danneggiati. Il self test plus deve essere eseguito con il POD diagnostico inserito.  
Riferitevi alla figura 7 per vedere come inserire correttamente il POD diagnostico.
- Alcuni dispositivi proteggono il contenuto delle loro EPROM tramite fusibili di protezione. Per poter leggere questi dispositivi i fusibili vanno disinnescati (usando le opzioni specifiche in Device/Device Options e Device/Program). Altrimenti l'utilizzo di questi dispositivi è impossibile, anche se i dispositivi, il PC ed il programmatore funzionano correttamente.

### **DISPOSITIVI NON SUPPORTATI**

Se avete bisogno di usare dispositivi non supportati dal programma di controllo, leggete i consigli seguenti:

- Controllate l'elenco dei dispositivi supportati nell'ultima versione del programma di controllo disponibile dai nostri siti internet (sezione programmatori, scegliete il nome del vostro programmatore). Il dispositivo che vi interessa può essere già incluso nel database dell'ultima versione.
- Contattate la **grifo**® direttamente. Potremmo avere bisogno dei data sheets del componente e, possibilmente, avere dei campioni. I campioni verranno restituiti dopo che il dispositivo da voi richiesto sarà stato incluso nella prossima versione del programma di controllo.

## ARTICOLI AGGIUNTIVI

Per estendere con facilità la già vasta quantità di dispositivi programmabili con i programmatori **grifo®**, i convertitori di formato permettono di usare formati diversi dal classico DIL.

Per ulteriori informazioni su quale convertitore sia adatto al vostro programmatore si prega di contattare la **grifo®** direttamente.

CONVERTITORI DI FORMATO	DESCRIZIONE
DIL20/PLCC20 ZIF	Convertitore di formato da DIL 20 a PLCC 20 ZIF.
DIL28/PLCC28 ZIF	Convertitore di formato da DIL 28 a PLCC 28 ZIF.
DIL32/PLCC32 ZIF	Convertitore di formato da DIL 32 a PLCC 32 ZIF.
DIL40/PLCC44 ZIF	Convertitore di formato da DIL 40 a PLCC 44 ZIF.
DILxx/PLCCyy ZIF	Convertitore di formato da DIL xx a PLCC yy ZIF-CS.
DIL16/SOIC16 150 mil	Convertitore di formato da DIL 16 a SOIC 16 type 150 mil ZIF.
DIL16/SOIC16 200 mil	Convertitore di formato da DIL 16 a SOIC 16 type 200 mil ZIF.
DIL20/SOIC20 300 mil	Convertitore di formato da DIL 20 a SOIC 20 type 300 mil ZIF.
DIL24/SOIC24 300 mil	Convertitore di formato da DIL 24 a SOIC 24 type 300 mil ZIF.
DIL28/SOIC28 300 mil	Convertitore di formato da DIL 28 a SOIC 28 type 300 mil ZIF.
DIL34/SOIC34 200 mil	Convertitore di formato da DIL 34 a SOIC 34 type 200 mil ZIF.

**FIGURA 13: LISTA DEI CONVERTITORI DI FORMATO**

## INDICE ANALITICO

**A**

AVVIO RAPIDO 17

**B**

BUFFER 2, 8

**C**

CALIBRAZIONE 15

COMANDI DEL PROGRAMMA 22

BUFFER 34

DEVICE 25

DIAGNOSTICS 43

FILE 22

HELP 44

OPTIONS 38

CONFIGURAZIONE PC 15

CONNESSIONE 12

CONSUMO 9

**D**

DAC 6

DIMENSIONI 9

DISPOSITIVI 6

DISPOSITIVI NON SUPPORTATI 48

**F**

FILES 8

**G**

GARANZIA 46

**H**

HEX 2, 22, 38

**I**

IEEE 1284 6, 14, 47  
IN-SYSTEM SERIAL PROGRAMMING 10  
INSTALLAZIONE 12  
INSTALLAZIONE SOFTWARE 15  
INTRODUZIONE 1  
ISP 2, 10

**L**

LEDS 14  
  BUSY 14  
  POWER 14

**O**

OPERAZIONI 8

**P**

PESO 9  
PINDRIVER 6  
PORTA STAMPANTE 2

**R**

RISOLUZIONE DEI PROBLEMI 47

**S**

SELFTEST 15  
SOCKET 6  
SOFTWARE 20

**T**

TARGET 2  
TASTI SPECIALI 21  
TEMPERATURA 9  
TEMPI 6

**U**

UMIDITÀ 9

**V**

VCCP 6  
VPP 6

Z

ZIF 2



## GENERAL INTRODUCTION

The purpose of this handbook is to give the necessary information to the cognizant and sure use of the products. They are the result of a continual and systematic elaboration of data and technical tests saved and validated from the manufacturer, related to the inside modes of certainty and quality of the information.

To be on good terms with the products, is necessary guarantee legibility and conservation of the manual, also for future references. In case of deterioration or more easily for technical updates, consult the Web site [www.grifo.com](http://www.grifo.com) or the AUTHORIZED TECHNICAL ASSISTANCE directly.

To prevent problems during product utilization, it is a good practice to read carefully all the informations of this manual. After this reading, the User can use the general index and the alphabetical index, respectly at the begining and at the end of the manual, to find information in a faster and more easy way.

Information provided in this manual is intended to be accurate at the moment of release, but we continuously improve all our products. Please consult the README\_P.TXT on the enclosed distribution disc for last minute changes.

This control program is copyrighted, all rights reserved. The control program or any part of it may not be analyzed, disassembled or modified in any form, on any medium, for any purpose.

This document is copyrighted, all rights reserved. This document or any part of it may not be copied, reproduced or translated in any form or in any way without the prior written permission of **grifo®**.

**grifo®** assumes no responsibility for misuse of this manual.

**grifo®** reserves the right to make changes or improvements to the product described in this manual at any time without notice.

This manual contains names of companies, software products, etc., which may be trademarks of their respective owners. **grifo®** respects those trademarks.

## PRELIMINARY INFORMATION

**MPPIK+** is little, very fast and powerful portable programmer for PICmicro® family microcontrollers and serial EEPROM with IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx) interface types. Using build-in in-circuit serial programming (ISP) connector programmer is able to program PICmicro® family microcontrollers using serial algorithms.

## CONVENTIONS AND TERMINOLOGY

There are some special conventions and terms used in this manual:

### CONVENTIONS USED IN THE MANUAL

References to the control program functions are in capitals, e.g. **LOAD**, **FILE**, **DEVICE**, etc.  
References to control keys are written in brackets <>, e.g. <F1>.

### TERMINOLOGY USED IN THE MANUAL

<b>DEVICE</b>	Any kind of programmable integrated circuits or programmable devices.
<b>ZIF socket</b>	Zero Insertion Force socket used for insertion of target device.
<b>BUFFER</b>	Part of memory or disk, used for temporary data storage.
<b>PRINTER PORT</b>	Kind of PC port (parallel), which is primarily dedicated to printer connection.
<b>ISP</b>	In-circuit Serial-Programming, that is the ability to program a device without removing it from its application board, even if it is soldered. <u>The board must be compatible with this feature.</u>
<b>HEX data format</b>	Format of data file, which may be read with standard text viewers; e.g. byte 5AH is stored as characters '5' and 'A', which means bytes 35H and 41H. One line of this HEX-file (one record) contains start address, data bytes and all records are secured with checksum.

This User manual is common to all programmers, so You will find information useful for every programmer and also information specific for each programmer.



## GENERAL FEATURES

This chapter contains a complete and exhaustive description the programmer, including a summary of its technical features and operations.

### INTRODUCTION

**MP PIK+** is a member of the new generation of Windows 95/98/ME/NT/2000/XP based ELNEC specialized programmers. Programmer is capable to support all currently available Microchip™ PICmicro® series microcontrollers (8-40 pins) using parallel and serial algorithms. The MP PIK+ programmer isn't development grade programmer, but meets all Microchip's requirements to manufacturing grade of programmers. The MP PIK+ has been developed in close cooperation with Microchip™ company, therefore programmer's hardware is focused to support all current and future PICmicro® family microcontrollers.

**MP PIK+** is a small, very fast and powerful portable programmer for PICmicro® family microcontrollers and serial EEPROM with IIC (24Cxx), Microwire (93Cxx) and SPI (25Cxx) interface types. Using build-in in-circuit serial programming (ISP) connector programmer is able to program PICmicro® family microcontrollers using serial algorithms.

Provides very competitive price but excellent hardware design for reliable programming. Nice "value for money" in this class.

Very fast programming due to high-speed FPGA driven hardware and support of IEEE1284 (ECP/EPP) high-speed parallel port. Surely faster than competitors in this category.

**MP PIK+** interfaces with the IBM PC, AT or above, portable or desktop personal computers through any standard parallel (printer) port (no special interface card needed). Therefore you can take programmer and move it to another PC without assembly/disassembly of PC.

**MP PIK+** has 40 powerful TTL pindrivers provide H/L/pull\_up/pull\_down and read capability for each pin of socket. Advanced pindrivers incorporate high-quality high-speed circuitry to deliver signals without overshoot or ground bounce for all supported devices. Pin drivers operate down to 1.8V so you'll be ready to program the full range of today's advanced low-voltage devices.

The programmer performs **device insertion test** (wrong device position in socket) and contact check (poor contact pin-to-socket) before it programs each device. These capabilities, supported by signature-byte check help prevent chip damage due to operator error.

**MP PIK+** programmer performs programming **verification** at the marginal level of supply voltage, which, obviously, improves programming yield, and guarantees long data retention.

**MP PIK+** is driven by an easy-to-use control program with pull-down menus, hot keys and on-line help. Selecting of device is performed by its class, by manufacturer or simply by typing a fragment of vendor name and/or part number. Standard device-related commands (read, blank check, program, verify, erase) are enhanced by some test functions (insertion test, signature-byte check), and some special functions (autoincrement). All known data formats are supported. Automatic file format detection and conversion during load of file.

The rich-featured **autoincrement function** enables to assign individual serial numbers to each programmed device - or simply increments a serial number, or the function enables to read serial numbers or any programmed device identification signatures from a file.

Various socket converters are available to handle device in PLCC, SOIC and other packages.



FIGURE 1: PHOTO OF PROGRAMMER MP PIK+

## TECHNICAL FEATURES

### SOCKET, PIN DRIVERS AND DACS

- FPGA based IEEE 1284 slave printer port, up to 1MB/s transfer rate
- 40-pin DIL ZIF (Zero Insertion Force) socket accepts both 300/600 mil devices up to 40-pins
- two D/A converters for VCCP and VPP, controllable rise and fall time
- VCCP range 0..7V/250mA
- VPP range 0..15V/150mA
- pindriver: 40 TTL pindrivers, special GND/VCC/VPP pindriver for PICmicro® devices
- spare GND, VCC and VPP driver, which add additional made-by-wire GND/VCC/VPP pin capability for future devices
- FPGA based TTL driver provides H, L, CLK, pull-up, pull-down on all pindriver pins
- level H selectable from 1.8 V up to 5V
- in-circuit serial programming (ISP) capability included
- continuity test: each pin is tested before every programming operation
- selftest capability
- autocalibration

### DEVICE SUPPORT

- microcontrollers Microchip™ PICmicro®: 12xxx, 14xxx, 16xxx, 17xxx and 18xxx series, 8 to 40 pin (\*1), parallel and serial mode
- serial E(E)PROM: 24Cxxx, 24Fxxx, 25Cxxx, 59Cxxx, 85xxx, 93Cxxx series

(\*1) suitable adapters are available for non-DIL packages and devices with more than 40 pins

### PROGRAMMING SPEED

These times strongly depend on PC speed, LPT port type and operating system free resources. Therefore are given values of two different PC configurations for comparison.

**Time A conditions:** *Pentium MMX, 250 MHz, ECP/EPP, WIN98.*

**Time B conditions:** *Athlon, 750 MHz, ECP/EPP on PCI bus, WIN98.*

DEVICE	OPERATION	TIME A	TIME B
PIC16C67	programming and verify	18 sec	16 sec

FIGURE 2: MP PIK+ PROGRAMMING TIMES



FIGURE 3: PACKAGE CONTENT

## **DEVICE OPERATIONS**

### **STANDARD**

- Intelligent device selection by device type, manufacturer or typed fragment of part name
- ISP programming
- Blank check, read, verify
- Program
- Erase
- Configuration and security bit program
- Illegal bit test
- Checksum

### **SECURITY**

- Insertion test
- Contact check
- ID byte check

### **SPECIAL**

- Statistic
- Auto device serial number increment
- Count-down mode

## **BUFFER OPERATIONS**

- View/edit, find/replace
- Fill/copy, move, byte swap, word/dword split
- Checksum (byte, word)
- Print

## **SUPPORTED FILE FORMATS**

- Unformatted (raw) binary
- HEX: Intel, Intel EXT, Motorola S, MOS, Exormax, Tektronix, ASCII-space-HEX

## **TECHNICAL SPECIFICATION**

### **GENERAL**

- Operating voltage 12...15V DC, max. 0.5A (adapter included in package)
- Power consumption max. 5W active, about 0.5W sleep
- Dimensions 132x66x30 [mm] (5.2x2.6x1.2 [inch])
- Weight (without external adapter) ca. 200g
- Temperature 5°..40°C
- Humidity 20%..80%, non condensing

### **PACKAGE INCLUDED**

- Programmer
- Cable with two 25 pin, D-type connectors for connecting the programmer to the PC
- ISP cable
- External power supply (suitable for respective programmer)
- Diagnostic POD for selftest of programmer
- Cap for ZIF socket (anti-dust cover)
- Mini-CD with the control program, this User's manual and additional files
- Shipping case

## ISP (IN-SYSTEM PROGRAMMING)

In-system programming allows programming and reprogramming of device positioned inside the end system. Using a simple interface, the ISP programmer communicates serially with the device, reprogramming nonvolatile memories on the chip. In-system programming eliminates the physical removal of chips from the system. This will save time, and money, both during development in the lab, and when updating the software or parameters in the field.

Target device is the device (microcontroller, PLD, etc...), which is to be in-system programmed. Target system is the physical Printed Circuit Board (PCB) which contains the device to be in-system programmed.

## **GENERAL RULES FOR IN-SYSTEM PROGRAMMING**

Please refer to figure 5.

We recommend respect following rules to avoid damages to PC, ISP programmer, target device or target system:

- Ensure common earth point for target system, ISP programmer and PC.
- For laptop or other PC which is not connected to common earth point: make hard - wired connection from laptop to common earth point (for example use LPT or COM port D – connector).

Any devices connected to target system must be connected to common earth point too.

During in-system programming you connect two electrical devices – ISP programmer and target system. Unqualified connection can damage these devices.

**NOTE: *When you don't keep below directions and you damage programmer during in-system programming, it is damage of programmer by unqualified manipulation and is out of warranty.***

1. Turn off both devices – ISP programmer and target device.
2. Assign same GND potential for all devices, e.g. connect GND of all devices by wire.
3. Insert one connector of ISP cable to ISP programmer, turn on programmer and control program.
4. In control program select target device and operation options.
5. Start action on target device (read, program).
6. After direction of control program, connect other ISP cable connector to target system and turn on it.
7. After direction of control program, disconnect other ISP cable connector from target system and turn off it.
8. If you need another action on target device, you continue with step 5.

### **NOTE:**

If target system use signals needed for ISP programming, it's necessary separate them from target device by appropriate resistors.



## THE RECOMMENDED ISP CONNECTOR LAYOUT IN TARGET SYSTEM

This interface corresponds with Atmel application note AVR910: In-System Programming. This application note is used in every ISP programmer and describes the recommended ISP interface connector layout in target system (top view in figure 6).

PIN	NAME	PURPOSE
SCK	Serial Clock	Programming clock, generated by the In-System programmer (master).
MOSI	Master Out Slave In	Communication line from In-System programmer (master) to target MCU being programmed (slave).
MISO	Master In Slave Out	Communication line from target MCU (slave) to In-System programmer (master).
RESET	Target MCU Reset	To enable In-System programming, the target MCU Reset must be kept active. To simplify this, the In-System programmer should control the target MCU Reset
Vcc	Target Power Supply	The target can have power supplied through the In-System programming connector for the duration of the programming cycle. Target system cannot supply SEP 40.
GND	Common Ground	The two systems must share the same common ground.

FIGURE 4: ISP PINOUT

Specification of ISP connector pins depends on programming device and is displayed at **Additional info window**. This specification correspond with application notes published of device manufacturers.

**Note:** Pin no. 1 is signed by triangle scratch on ISP cable connectors.

*H/L/read MP PIK+ driver is shown on figure 7.*

### Warnings:

- **When you use MP PIK+ as ISP programmer, don't insert device to ZIF socket.**
- **When you programm devices in ZIF sosket, don't insert ISP cable to ISP connector.**
- Use only **attached ISP cable**. When you use other ISP cable (other material, length...), programming may occur unreliable.
- **MP PIK+ can supply target device (target system with limitation), but target system cannot supply MP PIK+.**
- **MP PIK+ apply programming voltage to target device and checks his value (target system can modify programming voltage). If the programming voltage is different as expected, no action with target device will be executed.**

## INSTALLATION

### CONNECTING MP PIK+ TO THE PC

Switch off PC and programmer. Insert the communication cable included with Your **MP PIK+** programmer package to a free printer port on your PC. If Your computer is equipped with only one printer port, substitute the programmer cable for the printer cable.

Connect the opposite cable end to the programmer. Screw on both connectors to counter-connectors. This is very important.

It may be uncomfortable to switch between printer cable and programmer cable, but it is not possible to operate the programmer through a mechanical or an electronic. However You can install a second multi-I/O in your computer, thus obtaining a supplementary printer port, say LPT2. So your printer may remain on LPT1 while the programmer on LPT2.

Connect the mains connector of the power supply (or the wall-plug power supply itself) to a mains plug, then connect the mini-DIN connector to the programmer's connector labeled "12 VAC".

At this time all 'work result' LEDs (and 'POWER' LED) light up successive and then switch off. Once the POWER LED lights with low brightness then the **MP PIK+** programmer is ready to run. Next switch on the PC and run the control program for **MP PIK+**.

#### **! CAUTION !**

If You don't want to switch off your PC when connecting the programmer, proceed as follows:

- When connecting the programmer to the PC: **FIRST** insert the communications cable and **THEN** the power-supply connector.
- When disconnecting the programmer from the PC: **FIRST** disconnect the power-supply connector and **THEN** the communication cable.

From programmer's point of view the connecting and disconnecting sequence is irrelevant. Protection circuits on all programmer inputs keep it safe. **But think of your PC, please.**

#### **NOTE**

When the **MP PIK+** programmer isn't active, control program switch the programmer into "sleep mode". In this state it consumes less than 1 W of power. Therefore the programmer has no On/Off switch.

When the power LED indicator glows with a low intensity the **MP PIK+** is in sleep mode.

Programmer's protection electronics protect the target device and the programmer itself against either short or long-term power failures and, partly, also against a PC failure.

However, it is not possible to grant the integrity of the target device due to incorrect, user-selected programming parameters.

Target device may be not destroyed by forced interruption of the control program (reset or switch-off PC), by removing the physical connection to the programmer, but the content of actually programmed cell may remain undefined. Don't unplug the target device from the ZIF socket during work with devices (LED BUSY shine).

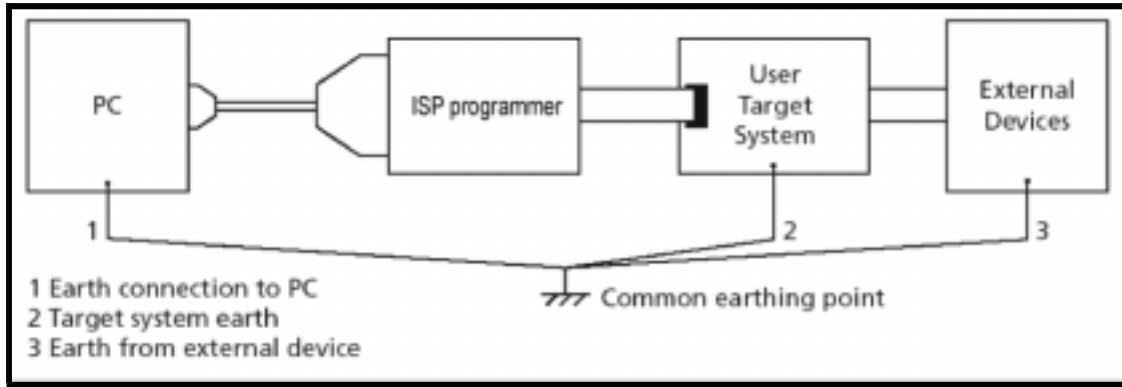


FIGURE 5: ISP CONNECTION EXAMPLE

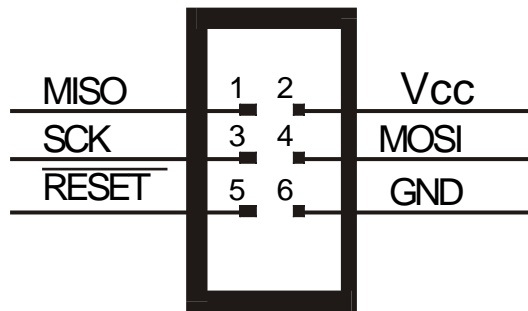


FIGURE 6: TARGET BOARD SUGGESTED ISP CONNECTOR

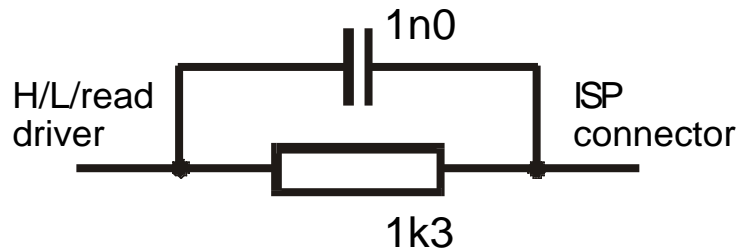


FIGURE 7: ISP DRIVER OF MP PIK+

Correspondance between ISP signals and connector pins may vary according to the device to be programmed.

To know this correspondance, please select the device desired in the control program (menu Device / Select Device) and press “Additional Info” button or press **Ctrl+F1**.

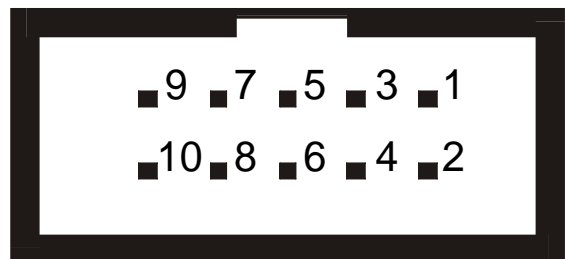


FIGURE 8: ISP CONNECTOR OF MP PIK+

## PROBLEMS RELATED TO MP PIK+ TO PC CONNECTION

Due a large variety of parallel port types, a case may occur when the programmer cannot “get concerted” with the PC. This problem may be shown as none communication between the PC and the programmer, or by unreliable communication.

- If this behavior occurs, try to connect your programmer to some other PCs or other parallel ports near you.
- Please remember that the printer port where the programmer is connected **must be** configured as IEEE 1284 compatible. You should check in the BIOS configuration of Your computer that the port is set as ECP/EPP. Or, if You use a multi I/O card, assure that it can be configured and used as IEEE 1284 compatible.
- Try to set the parallel port as Centronics compatible (or standard, or normal, or SPP, according to the BIOS manufacturer, all these denominations refer to the same setting for the parallel port). This could decrease performance but assure the highest communication reliability.

If You can't find any solution, please contact **grifo**® directly. Provide us an accurate description of Your PC configuration, including some other circumstances bearing on the problem in question. Don't forget please to tell PC type, manufacturer, speed, operating system type and version, resident programs; your parallel port I/O manufacturer and type.

## LEDS

**MP PIK+** is provided with 2 LEDs to visualize power supply status and operating programmer status.

Please refer to figure 11 to locate the LEDs.

LED	COLOUR	DESCRIPTION
Power LED	Green	At power on, it lights for a few seconds, then turns off.
BUSY LED	Red	When on, indicated that the programmer is accessing the target device for reading or writing. Do not touch the target device when this LED is on.

**FIGURE 9: MP PIK+ LEDS DESCRIPTION TABLE**

## **SELFTTEST AND CALIBRATION**

If you feel that your programmer does not react according to your expectation, please run the programmer selftest using Diagnostic POD, enclosed with the standard delivery package. For optimal results with programmer we recommend you undertake every 6 months, an extended test and to check the calibration. See instructions for selftest in the **Diagnostics** menu of PG4UW.

## **INSTALLING PROGRAMMER SOFTWARE**

The programmer package contains a mini-CD with the control program, useful utilities and additional information. Differences and modifications to this manual (if they exist) may be found in README\_P.TXT file.

## **MINIMAL PC CONFIGURATION**

- PC AT 10 MHz (DOS-version software), PC 486 (WIN-version software)
- 512 kB free RAM (DOS-version software), 16MB free RAM (WIN-version software)
- One CD drive
- HDD, 5 MB free space
- Operating system MS/PC-DOS version 3.2 or later, WIN 3.xx, WIN95/98
- One free printer port with nothing attached

## **RECOMMENDED PC CONFIGURATION**

- Pentium PC 100Mhz or higher
- 32 MB free RAM
- One CD drive
- HDD with minimum 10 MB free space
- Operating system: MS Windows 95/98/ME/NT/2000/XP
- One free bi-directional printer port on PCI Bus, IEEE 1284 or fully compatible

## **SOFTWARE INSTALLATION**

### **WINDOWS 95/98/ME/NT/2000/XP**

Insert delivered mini-CD to your CD drive and instal program starts automatically.

Install program (setup.exe), which will guide you through the installation process and which will do all the necessary steps before you can first run the control program.

### **DOS**

Run **Install** program from CD.

For DOS versions of the control program running under WINDOWS 3.11/95/98, we suggest the following: (adjust “Properties” on the icon on your screen):

- Run the control program in **full-screen DOS** (in text mode). In graphic mode the control program might be somewhat slower.
- If you want the control program to run in the **background**, it is necessary to set: (WIN95/98) RIGHT mouse button on the DOS icon/ Properties/ Others/ Background/ Enable; (WIN3xx) in the appropriate PIF file, set the option to Background

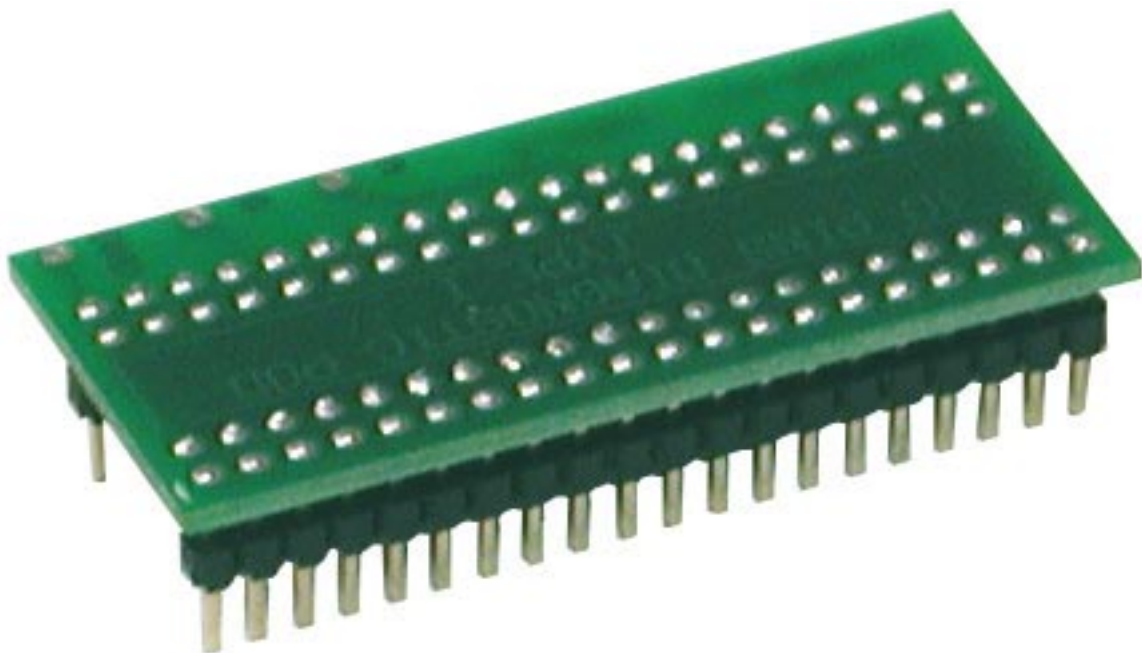
Programs PG4U.EXE (for DOS) and PG4UW.EXE (for Windows) are common control programs for all **grifo**® programmers. We guarantee running of these programs under all of above mentioned operating systems without any problems. Also background operation under Windows is error-free.

### NEW VERSIONS OF PROGRAMMER SOFTWARE

In order to exploit all the capabilities of programmer we recommend using the latest version of PG4U.EXE or PG4UW.EXE . You may download the latest version of programmer software from Internet site <http://www.grifo.com> or <http://www.grifo.it>.

### UPGRADING THE PROGRAMMER SOFTWARE

After downloading the file, just click on it. DOS program needs you to click also the “install.exe” file. Please answer “All” to the eventual “Overwrite” request.



**FIGURE 10: SELFTEST POD**

## QUICKSTART

These paragraphs contain all the information essential to connect the programmer to the PC, install the software and begin to use the programmer quickly. **Please read completely these paragraphs before attempting any use of Your programmer.** For more detailed information about the technical features of each the and the software options, please refer to the specific chapters further in this manual.

### INSTALLING PROGRAMMER HARDWARE

Please refer to figure 11:

- Switch off PC and programmer.
- Insert the communication cable included with Your programmer package to a free printer port on Your PC. If Your computer is equipped with only one printer port, substitute the programmer cable for the printer cable.
- Connect the opposite cable end to the programmer.
- Screw on both connectors to counter-connectors. This is very important.
- Connect the power supply connector to the programmer and the power supply plug to a mains socket.
- Turn on PC and programmer.
- You will see the green LED of power supply turning on and then switching off, while the three status LEDs turn on and off in sequence. On all other programmers the green LED of power supply simply turns on and remains on.

It may be uncomfortable to switch between printer cable and programmer cable, but it is not possible to operate the programmer through a mechanical or an electronic. However You can install a second multi-I/O in your computer, thus obtaining a supplementary printer port, say LPT2. So your printer may remain on LPT1 while the programmer on LPT2.

#### **! CAUTION !**

If You don't want to switch off your PC when connecting the programmer, proceed as follows:

- When connecting the programmer to the PC: **FIRST** insert the communications cable and **THEN** the power-supply connector.
- When disconnecting the programmer from the PC: **FIRST** disconnect the power-supply connector and **THEN** the communication cable.

## SOFTWARE INSTALLATION

### **WINDOWS 95/98/ME/NT/2000/XP**

Insert delivered CD to your CD drive and instal program starts automatically.

Install program (setup.exe), which will guide you through the installation process and which will do all the necessary steps before you can first run the control program.

### **DOS**

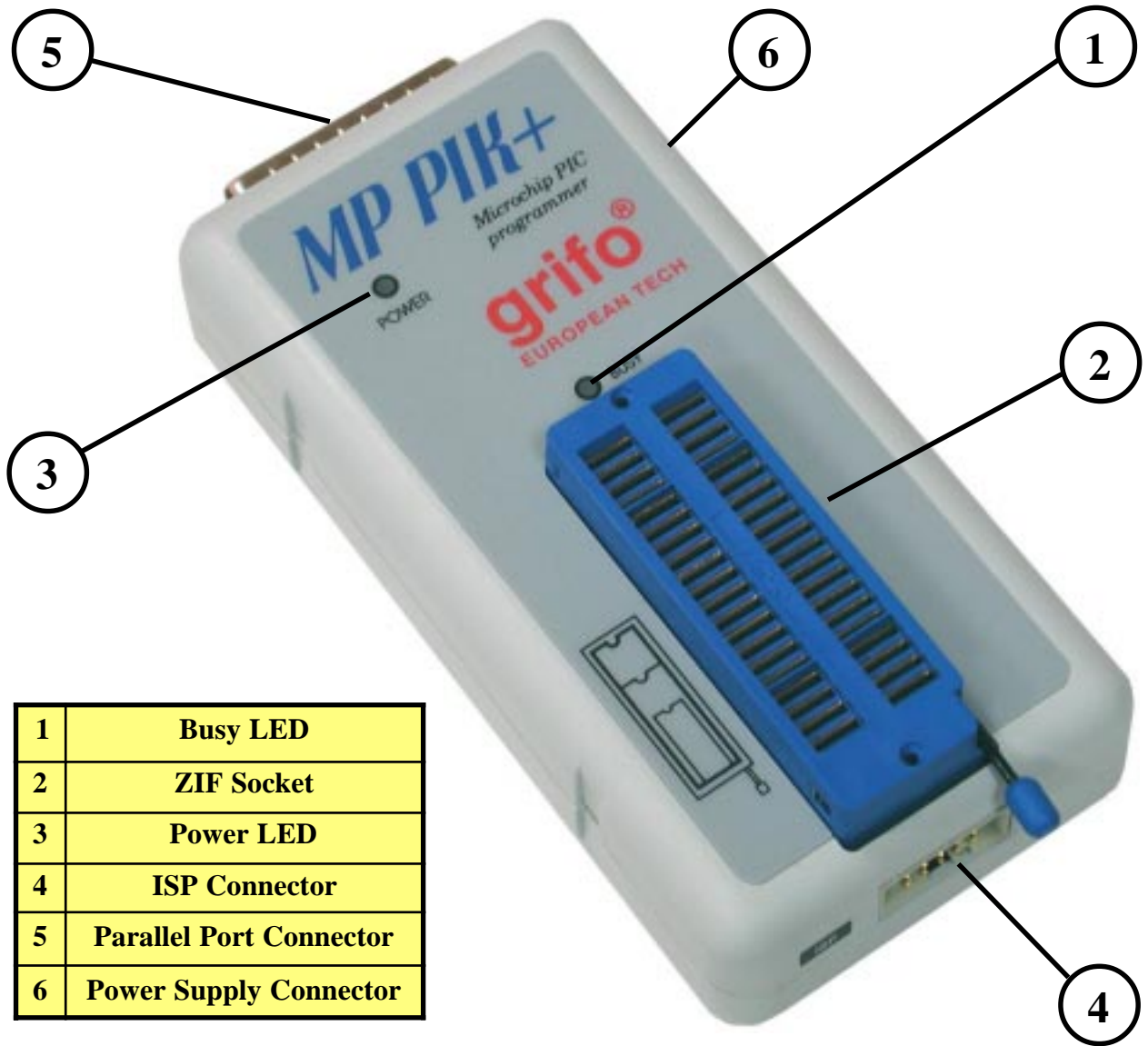
Run **Install** program from CD.

## QUICK INSTRUCTIONS TO PROGRAM OR COPY A DEVICE

- Execute PG4U.EXE (DOS) or click PG4UW.EXE (WIN) icon to enter the control program.
- If the hardware has been connected properly, the program now should display on the screen a window to inform that has found the programmer, then, in the main screen, it should display the programmer status in the top right frame. Should this not happen, please repeat the steps described in “INSTALLING PROGRAMMER HARDWARE” or refer to the paragraph “TROUBLESHOOTING”.
- Select the device You want to program, either through the Device/Select menu or the keyboard shortcut <Alt+F5> (press key <F5> while keeping Alt key pressed).
- Load the data by which to program the selected device:
  - § If You need to use a file on disk, use File/Load menu or the keyboard shortcut <F3>. If You need to load specific file types please refer to chapter “SOFTWARE DESCRIPTION” .
  - § If You need to copy the content of an already programmed device, You must read its content through the Device/Read menu or the keyboard shortcut <F7>.
- Insert the device to be programmed into the ZIF socket of the programmer and program it with the current buffer content through the Device/Program menu or the keyboard shortcut <F9>.
- After programming, You can verify the content of the programmed device with the content of the memory buffer through the Device/Verify menu or the keyboard shortcut <F8>.

The menu Device contains the device manipulation commands. The menu File contains commands for files and directories. The menu Buffer is to be used for buffer manipulation.





1	Busy LED
2	ZIF Socket
3	Power LED
4	ISP Connector
5	Parallel Port Connector
6	Power Supply Connector

FIGURE 11: ELEMENTS MAP



FIGURE 12: ISP PROGRAMMING CABLE

## SOFTWARE DESCRIPTION

This chapter contains an essential description of the control software for all the programmers. Please remark that on line help is updated before the manual.

### USING THE PROGRAMMER SOFTWARE

#### **! IMPORTANT !**

The control program delivered, included on the mini-CD in Your package, is granted to be free from any viruses at the moment of delivery. To increase their safety our programs include a special algorithm for detecting possible virus infections.

#### **RUN THE CONTROL PROGRAM**

From command line of MS/PC-DOS write: PG4U  
in Windows environment: double click to icon PG4UW.

After start, control program PG4U / PG4UW automatically scans all existing ports and search for the connected any **grifo**® programmer. Program PG4U/PG4UW is common for all the **grifo**® programmers, hence program try to find all supported programmers.

**Notes:** *When the PG4U/PG4UW program is started, program is checked for its integrity. Than the program display a standard user menu and waits for your instructions.*

If the control program cannot communicate with the programmer, an error message appears on the screen, including error code and description of possible reasons (disconnected programmer, bad connection, power supply failure, incompatible printer port, ...). Eliminate the error source and press any key. If error condition still exist, the program resumes its operation in the demo mode and access to the programmer is not possible. If you cannot find the cause of the error, follow the instructions in **Troubleshooting** section. In addition, the control program checks communication with programmer prior to any operation with the programmed device.

#### **DESCRIPTION OF THE USER SCREEN**

Windows program PG4UW

<b>Header bar</b>	name, copyright statement and version of PG4U/PG4UW control program
<b>Menu bar</b>	list of basic functions
<b>File window/Filename</b>	information on the currently loaded file in buffer
<b>Status window</b>	information about the status of the programmer and PG4U/PG4UW
<b>Addresses window</b>	organization, size, start and end addresses of the target device, buffer and file
<b>Device window</b>	all relevant information about current target device
<b>Help bar</b>	a brief description of selected command



Menu selection is carried out in the normal GUI fashion - either by cursor moving plus pressing <Enter>, or by typing the highlighted letter in the wanted menu or - of course - by mouse. Hot-keys are available for even quicker selection of intensely used commands.

### ! CAUTION !

Data entered through keyboard is in HEX format, excepting ASCII blocks in Buffer/View/Edit command.

### LIST OF HOT KEYS

<F1>	HELP	Calls Help
<F2>	SAVE	Save file
<F3>	LOAD	Load a file into the buffer
<F4>	EDIT	Viewing/editing of buffer
<F5>	SELECT/default	Target-device selection from 10 last selected devices list
<Alt+F5>	SELECT/manual	Target-device selection by typing device or vendor name
<Ctrl+F5>	SELECT/type	Target-device selection by device family
<Shift+F5>	SELECT/vendor	Target-device selection by device manufacturer
<F6>	BLANK	Blank check
<F7>	READ	Reads device's content into the buffer
<F8>	VERIFY	Compares contents of the target device with the buffer
<F9>	PROGRAM	Programs target device
<Alt+Q>	QUIT	Terminates the PG4U
<Alt+X>	QUIT & save	Terminates the PG4U and saving settings too
<Ctrl+F1>		Displays additional information about current device
<Ctrl+F2>	ERASE	Fills the buffer with a given value
<Ctrl+Shift+F2>		Fills the buffer with random values

## CONTROL PROGRAM COMMANDS

The following is a list of all the commands available with Windows version of control program. The command name is matched with its keyboard shortcut (if present) and a brief description of its purpose and employ modalities. Please remember that the most recently updated instructions list for the control program is always the on line help, which should be consulted before this manual.

### **File**

This submenu is used for source files manipulation, settings and viewing directory, changes drives, changes start and finish address of buffer for loading and saving files by **binary**, **MOTOROLA**, **MOS Technology**, **Intel (extended) HEX**, **Tektronix**, **ASCII space**, **JEDEC**, and **POF** format.

#### **File / Load**

Analyse file format and loads the data from specified file to the buffer. You can choose the format desired (**binary**, **MOTOROLA**, **MOS Technology**, **Tektronix**, **Intel (extended) HEX**, **ASCII space**, **JEDEC** and **POF**). The control program stores a last valid mask for file listing. You can save the mask into the config. file by command **Options / Save options**.

Checking the check box **Automatic file format recognition** tells program to detect file format automatically. When program can't detect file format from one of supported formats, the binary file format is assumed.

When the check box **Automatic file format recognition** is unchecked program allows user to manually select wished file format from list of available file formats on panel **Selected file format**. When Binary file format is selected, there can be specified Buffer start value. Buffer start value is buffer address from which data read from file will be written to buffer.

The reserved key <F3> will bring out this menu from any menu and any time.

#### **File / Save**

Saves data in the buffer, which has been created, modified, or read from a device onto a specified disk. You can choose the format desired (**binary**, **MOTOROLA**, **MOS Technology**, **Tektronix** and **Intel (extended) HEX**).

The reserved key <F2> will bring out this menu from any menu and any time.

#### **File / Load project**

This option is used for loading project file, which contains device configuration buffer data saved and user interface configuration.

The standard dialog **Load project** contains additional window - **Project description** - placed at the bottom of dialog. This window is for displaying information about currently selected project file in dialog Load project.

Project information consists of:

- name and manufacturer of the first device selected in the project
- date and time of project creation
- version of program by which project was created
- user written description of project (it can be arbitrary text, usually author of project and some notes)

## File / Save project

This option is used for saving project file, which contains settings of device configuration and buffer data saved. Data saved to project file can be restored anytime by menu command **File / Load project**.

The dialog **Save project** contains three additional windows in **Project description** panel placed at the bottom of dialog **Save project**. The windows are for displaying information about currently selected project file in dialog **Save project** and information about current project which has to be saved.

Project information consists of:

- name and manufacturer of the first device selected in the project
- date and time of project creation
- version of program by which project was created
- user written description of project (it can be arbitrary text, usually author of project and some notes)

The first (upper) window contains information about currently selected project file in dialog **Save project**.

The second (middle) windows displays information about actual program configuration including currently selected device, active programmer, date and time and program version. These actual program settings are used for creation of project description header.

The third (bottom) window is user editable and contains project description (arbitrary text) which usually consists of project author and some notes.

## File / Reload file

Choose this option to reload a recently used file.

When you use a file, it is added to the **Reload file** list. Files are listed in order depending on time of use of them. Lastly used files are listed before files used far off.

To Reload a file:

1. From the File menu, choose Reload file.
2. List of lastly used files is displayed. Click the file you want to reload.

**Note:** When reloading a file the file format is used, by which the file was lastly loaded/saved.

## File / Reload project

Choose this option to reload a recently used project.

When you use a project, it is added to the **Reload project** list. Projects are listed in order depending on time of use of them. Lastly used projects are listed before projects used far off.

To Reload a project:

1. From the File menu, choose Reload project.
2. List of lastly used projects is displayed. Click the project you want to reload.

## File / Project options

This option is used for display/edit project options of actually loaded project. Project options means basic description of project including following project data:

- device name and manufacturer
- project creation date
- version of program by which project was created
- user defined project description (arbitrary text), e.g. project author and other text data for more detailed project description

User can directly edit user defined project description only. Device name, manufacturer, project date and program version are generated automatically by program.

## File / Load encryption table

This command loads the data from binary file from disk and it saves them into the part of memory, reserved for an encryption (security) table.

## File / Save encryption table

This command writes the content of the memory's part, reserved for an encryption table, into the file on the disk as a binary data.

## File / Exit without save

The command deallocates heap, cancels buffer on disk (if exists) and returns back to the operation system.

## File / Exit and save

The command deallocates heap, cancels buffer on the disk (if exists), saves current setting of last 10 selected devices to disk and returns back to the operation system

## Device

The functions for a work with selected programmable devices - device select, read data from device, device blank check, device program, device verify and device erase.

### Device / Select from default devices

This window allows selecting the desired type of the device from list of default devices. This one is a cyclic buffer in which are stored last 10 selected devices including its device options. This list is saved to disk by command **File / Exit and save**.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules), that supported this device. You can find here a package information and other general information about current device too.

Use a **<Del>** key for delete of current device from list of default devices. There isn't possible to empty this list, if you repeat this access. The last device stays in buffer and the **<Del>** key isn't accepted.

### Device / Select device ...

This window allows selecting the desired type of the device from all devices supported by current programmer. It is possible to choose device by **name**, by **type** or by **manufacturer**.

Selected device is automatically saved to buffer of default devices (max. 10 devices). This buffer is accessible with **Device / Select from default devices** command.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules), that supported this device. You can find here a package information and other general information about current device too.

### Select device ... / All

This window allows selecting the desired type of the device from all devices supported by current programmer. Supported devices are displayed in a list box.

Device can be select by double click on a line from list with desired manufacturer name and device number or by entering manufacturer name and/or device number in a search box (use a key **<Space>** as a separation character) and press **<Enter>** or click **OK** button.

Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

Selected device is automatically saved to buffer of default devices (max. 10 devices). This buffer is accessible with **Device / Select from default devices** command.

If you wish display additional information about the current device, use an **<Ctrl+F1>** key.

This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) which supported this device. You can find here a package information and other general information about current device too.

### Select device ... / Only selected type

This window allows selecting the desired type of the device. At the first - you must select a device type (e.g. EPROM) and device subtype (e.g. 64Kx8 (27512)), using mouse or cursor keys. It will cause a list of manufacturers and devices will be displayed.

Device can be select by double click on a line from list with desired manufacturer name and device number or by entering manufacturer name and/or device number in a search box (use a key <Space> as a separation character) and press <Enter> or click **OK** button.

Press a key <Esc> or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

Selected device is automatically saved to buffer of default devices (max. 10 devices). This buffer is accessible with **Device / Select from default devices** command.

If you wish display additional information about the current device, use an <Ctrl+F1> key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) which supported this device. You can find here a package information and other general information about current device too.

### Select device ... / Only selected manufacturer

This window allows selecting the desired device type by manufacturer. First select a required manufacturer in Manufacturer box using mouse or cursor keys. It will cause a list of selected manufacturer devices will be displayed.

Device can be select by double click on a line from list with desired manufacturer name and device number or by entering device number in a search box (use a key <Space> as a separation character) and press <Enter> or click **OK** button.

Press a key <Esc> or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

Selected device is automatically saved to buffer of default devices (max. 10 devices). This buffer is accessible with **Device / Select from default devices** command.

If you wish display additional information about the current device, use an <Ctrl+F1> key. This command provides a size of device, organization, programming algorithm and a list of programmers (including auxiliary modules) which supported this device. You can find here a package information and other general information about current device too.

## Device / Select EPROM /Flash by ID

Use this command for autoselect an EPROM or Flash as active device by reading the device ID. The programmer can automatically identify certain devices by the reading the manufacturer and



the device-ID that are burnt into the chip. This only applies to EPROM or Flash that supports this feature. If the device does not support a chip ID and manufacturer's ID, a message will be displayed indicating this as an unknown or not supported device.

If more devices with identical chip ID and manufacturer's ID were detected, the list of these devices will be displayed. A corresponding device can be chosen from this list by selecting its number (or manufacturer name) from list and press **<Enter>** (or click **OK** button). Press a key **<Esc>** or click **Cancel** button at any time to cancel device selection without affecting the currently selected device.

**WARNING:** *The control program only support this time EPROM's and Flash with 28 and 32 pins. Any of programmers determines pins number automatically. For other programmers you must enter this number manually.*

The programmer applies a high voltage to the appropriate pins on the socket. This is necessary to enable the system to read the device ID. Do not insert into the socket a device that is not an EPROM or Flash. It may be damaged when the programmer applies the high voltage.

We don't recommend apply this command to 2764 and 27128 EPROM types, because most of them ID not supports.

## Device / Device options

Settings of this menu are used for programming process, serialization and associated file control.

### Device / Device options / Operation options

All settings of this command are used for programming process control. This is a flexible environment which content items associated with current device and programmer type. Items, which are valid for the current device but aren't supported by current programmer, are disabled. These settings are saving to disk along with associated device by **File / Exit and save** command. List of items:

- group **Addresses:**

device start address	(default 0)
device end address	(default device size-1)
buffer start address	(default 0)

- group **Insertion test:**

insertion test	(default ENABLE)
check ID bytes	(default ENABLE)

- group **Command execution:**

blank check before programming	(default DISABLE)
erase before programming	(default DISABLE)
verify after reading	(default ENABLE)
verify after programming	(ONCE, TWICE)
verify options	(nominal VCC 5%, nominal VCC 10%, VCCmin VCCmax)

### Device / Device options / Serialization

Serialization is special mode of program. When a serialization mode is activated, a specified value is automatically inserted on predefined address into buffer before programming each device. When more devices are programmed one by one, the serial number value is changed for each device automatically and inserted into buffer before programming device, so each device has unique serial number.

There are two types of serialization:

- Incremental mode
- From file mode

If a new device is selected, the serialization function is set to a default state i.e. disabled. Actual serialization settings for actually selected device are saving to disk along with associated device by **File / Exit and save** command.

When incremental mode is active following actual settings are saved to configuration file: address, size, serial value, incremental step and settings of modes ASCII / BIN, DEC / HEX, LS byte / MS Byte first.

When from-file mode is active following actual settings are saved to configuration file: name of input serialization file and actual label, which indicates the line with actual serial number in input file.

When program is in multiprogram mode (multiple socket programmer is actually selected) the special section - **Action on not programmed serial values due to error** - is displayed in dialog **Serialization**. In this section two choices are available:

1. Ignore not programmed serial values
2. Add not programmed serial values to file

**Ignore not programmed serial values** means the not programmed serial values are ignored and no action is done with them.

**Add not programmed serial values to file** means the not programmed serial values are added to file. The file of not programmed serial values has the same text format as serialization file for "From-file" serialization mode. So there is possible to program the serial values later on by "From-file" serialization mode.

If device programming is stopped by user, program will not change the serial values ready for next batch of devices. The same situation is if device program is incomplete, e.g. for device insertion test error.

Ignoring or writing not programmed serial values is only used when at least one device from current batch of devices in multiple socket module programmer is completely programmed and verified without errors.

## Device / Device options / Serialization / Incremental mode

The **Incremental mode** enables to assign individual serial numbers to each programmed device. A starting number entered by user will be incremented by specified step for each device program operation and loaded in selected format to specified buffer address prior to programming of each device.

There are following options, that user can modify for incremental mode:

### S / N size

S / N size option defines the number of bytes of serial value which will be written to buffer. For Bin (binary) serialization modes values 1-4 are valid for S / N size and for ASCII serialization modes values 1-8 are valid for S / N size.

### Address

Address option specifies the buffer address, where serial value has to be written. Note that address range must be inside the device start and device end addresses. Address must be correctly specified so the last (highest or lowest) byte of serial value must be inside device start and device end address range.

### Start value

Start value option specifies the initial value, from which serialization will start. Generally the max. value for serialization is \$1FFFFFFF in 32 bit long word.

When the actual serial value exceeds maximum value, three most significant bits of serial number are set to zero. After this action the number is always inside 0..\$1FFFFFFF interval (this is basic style of overflow handling).

### Step

Step options specifies the increment step of serial value incrementation.

### S / N mode

S / N mode option defines the form in which serial value has to be written to buffer. Two options are available:

- ASCII
- Bin

**ASCII** - means the serial number is written to buffer as ASCII string. For example number \$0528CD is in ASCII mode written to buffer as 30h 35h 32h 38h 43h 44h ('0' '5' '2' '8' 'C' 'D'), i.e. six bytes.

**Bin** - means the serial number is written directly to buffer. If the serial number has more than one byte length, it can be written in one of two possible byte orders. The byte order can be changed in „Save to buffer“ item.

### Style

Style option defines serial number base. There are two options:

- Decimal
- Hexadecimal.

**Decimal** numbers are entered and displayed using the characters '0' through '9'.

**Hexadecimal** numbers also use characters 'A' through 'F'.

The special case is Binary Dec, that means BCD number style. BCD means the decimal number is stored in hexadecimal number, i.e. each nibble must have value from 0 to 9. Values A to F

are not allowed as nibbles of BCD numbers.

Select the base in „Style“ options before entering numbers of serial start value and step.

### Save to buffer

Save to buffer option specifies the serial value byte order to write to buffer. This option is used for Bin S / N mode (for ASCII mode it has no effect).

Two options are available:

- LSByte first (used by Intel processors) will place the Least Significant Byte of serial number to the lowest address in buffer.
- MSByte first (used by Motorola processors) will place the Most significant Byte first to the lowest address in buffer.

## Device / Device options / Serialization / From file mode

Using the From-file method, serial values are read from the user specified input file and written to buffer on address specified in input file.

There are two user options: File name and Start label.

### File name

File name option specifies the file name from which serial addresses and values will be read. The input file for From file serialization must have special format, which is described in From file serialization file format below.

### Start label

Start label defines the start label in input file. The reading of serial values from file starts from defined start label.

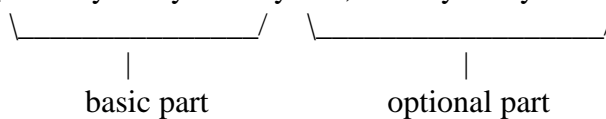
### From file serialization file format

From file serialization input file includes addresses and arrays of bytes defining buffer addresses and data to write to buffer. Input file has text type format, which structure is:

[label $l$ ] addr byte $0$  byte $1$  .. byte $n$

...

[label $n$ ] addr byte $0$  byte $1$  .. byte $m$  , addr byte $0$  byte $1$  ... byte $k$  ; Comment



meaning is:

basic part

Basic part defines buffer address and array of bytes to write to buffer. Basic part must be always defined after label in line.

optional part

Optional part defines the second array of bytes and buffer address to write to buffer. One optional part can be defined after basic part of data.

label $l$ , label $n$  - labels

Labels are identifiers for each line of input file. They are used for addressing each line of file. The labels should be unique. Addressing lines of file means, the required start label entered by user defines line in input file from which serial values reading starts.

**addr -**

Addr defines buffer address to write data following the address.

byte0..byten, byte0..bytem, byte0..bytek -

Bytes arrays byte0..byten, byte0..bytem and byte0..bytek are defining data, which are assigned to write to buffer. Maximum count of bytes in one data field following the address is 64 bytes.

Data bytes are written to buffer from address addr to addr+n.

The process of writing particular bytes to buffer is:

byte0 to addr

byte1 to addr + 1

byte2 to addr + 2

....

byten to addr + n

Optional part is delimited from the first data part by character “ , ” (comma) and its structure is the same as in the first data part, i.e. address and following array of data bytes.

**Characters with special use:**

[ ] - labels must be defined inside square brackets

‘,’ – character which delimiters basic part and optional part of data

‘;’ - the semicolon character means the beginning of a comment. All characters from „;„, to the end of line are ignored. Comment can be on individual line or in the end of definition line.

**Note:**

- Label names can contain all characters except ‘[’ and ‘]’. The label names are analysed as non case sensitive, i.e. character ‘a’ is same as ‘A’, ‘b’ is same as ‘B’ etc..

- All address and byte number values in input file are hexadecimal.

- Allowed address value size is from 1 to 4 bytes.

- Allowed size of data arrays in one line is in range from 1 to 64 bytes. When there are two data arrays in one line, the sum of their size in bytes can be maximally 80 bytes.

- Be careful to set correct addresses. Address must be defined inside device start and device end address range. In case of address out of range, warning window appears and serialization is set to disabled (None).

**Example:**

```
[nav1] A7890 78 89 56 02 AB CD ; comment1
```

```
[nav2] A7890 02 02 04 06 08 0A
```

```
[nav3] A7890 08 09 0A 0B A0 C0 ; comment2
```

```
[nav4] A7890 68 87 50 02 0B 8D
```

```
[nav5] A7890 A8 88 59 02 AB 7D
```

;next line contains also second definition

```
[nav6] A7890 18 29 36 42 5B 6D , FFFF6 44 11 22 33 99 88 77 66 55 16
```

; this is last line - end of file

In the example file six serial values with labels „nav1“, „nav2“, ...“nav6“ are defined. Each value is written to buffer on address \$A7890. All values have size 6 bytes. The line with „nav6“ label has also second value definition, which is written to buffer on address \$FFFF6 and has size 10 bytes, i.e. the last byte of this value will be written to address \$FFFFF.

## Device / Device options / Statistics

Statistics gives the information about actual count of device operations, which were proceeded on selected type device. If one device is corresponding to one device operation, e.g. programming, the number of device operations will be equal to number of programmed devices.

The next function of statistics is **Count down**. Count down allows checking the number of device operations, and then number of devices, on which device operations have to be done. After each successful device operation the value of count down counter is decremented. Count down has user defined start number of devices to do. When count down value reach zero, it means, specified number of devices is complete and user message about complete count down will be displayed.

**Statistics** dialog contains following options:

Check boxes **Program**, **Verify**, **Blank**, **Erase** and **Read** define operations, after which statistics values increment.

Check box **Count down** sets Count down activity (enable or disable). Edit box following the Count down check box defines initial number of count down counter, from which count down starts.

**Statistics** dialog can be also opened by pressing right mouse button on Statistics panel and clicking displayed item Statistics.

Actual statistics values are displaying in main window of control program in Statistics panel.

Statistics panel contains three statistics values – **Success**, **Failure**, **Total** and two **Count down** information values **Count down** and **Remains**.

Meaning of the values is:

<b>Success</b>	number of operations which where successfully completed
<b>Failure</b>	number of operations which where not successfully completed
<b>Total</b>	number of all operations
<b>Count down</b>	informs about Count down activity (Enabled or Disabled)
<b>Remains</b>	informs about remaining number of device operations to do

Successful operation means any device operation of these, which is completed without errors:

- program
- verify
- blank check
- erase
- read

If device operation is finished with error(s) it is not successful operation.

When new device type is selected, all statistics values are set to zero and **Count down** is set to **Disabled**.

**Reset** button in **Statistics** panel reset statistics values.

**Reload Count down** button in **Statistics** panel reloads initial value to **Count down**.

## Device / Device options / Associated file

This command is used for setting associated file with current device. This is a file, which can be automatic loaded to buffer after device is selected from default devices select list or by start control program.

You can edit the associated file name in file name box, put a full pathname. The control program checks the present of this file on the disk. Also is possible enabling or disabling automatic load of this file.

You can save both settings i.e. associated file and enabling of automatic load of this file to disk by command **File / Exit and save**.

## Device / Blank check

This command allows to blank check of all devices or its part if possible. The control program reports a result of this action by a write of a warning message to INFO window.

The menu command **Device / Device options / Operation options** allows to set another working area as the standard.

## Device / Read

This command allows to read all device or its part into the buffer. The control program reports a finish of this action by write a message to INFO window.

The menu command **Device / Device options / Operation options** allows to set another working area as the standard. Setting an option Verify data after reading in this menu command means a higher reliability for device reading.

## Device / Verify

This command compares the programmed data of the all device or its part with data in buffer. The control program reports a result of this action by a write of an error message to INFO window.

The menu command **Device / Device options / Operation options** allows to set another working area as the standard.

By the setting in the menu **Options / Display errors** the command lets to write the found errors on the display or write the found errors to VERIFY.ERR file. In the Display errors mode to the screen can display the program max. 45 the first found differences, which are located by the address where they were caused.

## Device / Program

This command allows to programming of the all device or its part by the data of the buffer. The control program reports a result of this action by a write of an error message to INFO window.

The menu command **Device / Device options / Operation options** allows to set another working area as the standard, and set other operation options for programming process control.

## Device / Erase

This command allows to erase the all programmable device. The program reports the end without error or end with the error by writes the warning report on the display.

## Device / Test

This command executes a test with device selected from list of supported devices (e.g. static RAM) on programmers, which support this test.

## Device / Device info

The command provides additional information about the current device - size of device, organization, programming algorithm and a list of programmers (including auxiliary modules), that supported this device. You can find here a package information and other general information about current device too.

The reserved key <Ctrl+F1> will bring out this menu from any menu and any time immediately.

## Buffer

This menu is used for buffer manipulation, block operation, filling a part of buffer with string, erasing, checksum and of course editing and viewing with other items (find and replace string, printing...).

### Buffer / View/Edit

This command is used to view (view mode) or edit (edit mode) data in buffer (for viewing in DUMP mode only). Use arrow keys for select the object for edit. Edited data are signified by colour.

You can use <F4> hot key also.

View/Edit Buffer

- |                      |   |
|----------------------|---|
| <b>F1</b>            | display help of actual window   |
| <b>F2</b>            | fill block causes filling selected block of buffer by requested hex (or ASCII) string. Sets start and end block for filling and requested hex or ASCII string.  |
| <b>Ctrl+F2</b>       | erase buffer with specified blank value   |
| <b>Ctrl+Shift+F2</b> | fill buffer with random data  |
| <b>F3</b>            | copy block is used to copy specified block of data in current buffer on new address. Target address needn't be out from source block addresses.   |
| <b>F4</b>            | move block is used to move specified block of data in current buffer on new address. Target address needn't be out from source block addresses. Source address block (or part) will be filled by topical blank character.   |
| <b>F5</b>            | swap bytes command swaps a high- and low- order of byte pairs in current buffer block. This block must started on even address and must have an even number of bytes. If this conditions do not fulfil, the program modifies addresses itself (start address is moved on lower even address and/or end address is moved on higher odd address). |
| <b>F6</b>            | print buffer  |



<b>F7</b>	find string (max. length 16 ASCII characters)
<b>F8</b>	find and replace string (max. 16 ASCIIchars.)
<b>F9</b>	change current address
<b>F10</b>	change mode view / edit
<b>F11</b>	switch the mode of buffer data view between 8 bit and 16 bit view. It can be also do by mouse clicking on the button to the right of View/Edit mode buffer indicator. This button indicates actual data view mode (8 bit or 16 bit), too.
<b>F12</b>	checksum dialog allows to count checksum of selected block of buffer change mode view / edit
<b>Arrow keys</b>	move cursor up, down, right and left
<b>Home/End</b>	jump on start / end current line
<b>PgUp/PgDn</b>	jump on previous / next page
<b>Ctrl+PgUp/PgDn</b>	jump on start / end current page
<b>Ctrl+Home/End</b>	jump on start / end current device
<b>Backspace</b>	move cursor one position left (back)

**Note:** characters 20H - FFH (mode ASCII) and numbers 0..9, A..F (mode HEX) immediately changes content of edit area.

**Warning:** Editing of ASCII characters for word devices is disabled.

### Print buffer

This command allows write selected part of buffer to printer or to file. Program uses an external text editor in which selected block of buffer is displayed and can be printed or saved to file, too. By default is set simple text editor **Notepad.exe**, which is standard part of all versions of MS Windows.

In Print buffer dialog are following options:

Defines start address of selected block in buffer.

Block end

Defines end address of selected block in buffer.

### External editor

Defines path and name of external program, which has to be used as text viewer for selected block of buffer. By default is set simple text editor Notepad.exe, which is standard part of all versions of MS Windows. User can define any text editor for example Wordpad.exe, which is able to work with large text files. In user defined text editor user can print or save to file selected block of buffer.

The external editor path and name is saved automatically to disk.

### Find dialog box

Enter the search string to **Find** to text input box and choose <**Find**> to begin the search or choose <**Cancel**> to forget it.

**Direction** box specifies which way you want to search, starting from the current cursor position (In edit mode). **Forward** (from the current position or start of buffer to the end of the buffer) is the default. **Backward** searches toward the beginning. In view mode searches all buffer.

**Origin** specifies where the search should start.

### Find & Replace dialog box

Enter the search string in the **Text to find** string input box and enter the replacement string in the **Replace with** input box.

In **Options** box you can select prompt on replace: if program finds instance you will be asked before program change it.

**Origin** specifies where the search should start.

**Direction** box specifies which way you want to search, starting from the current cursor position (In edit mode). **Forward** (from the current position or start of buffer to the end of the buffer) is the default. **Backward** searches toward the beginning. In view mode searches all buffer.

Press <Esc> or click **Cancel** button to close dialog window.

By pressing **Replace** button the dialog box is closed and a Question window is displayed. This window contains following choices:

<b>Yes</b>	replaces found item and finds next
<b>No</b>	finds next item without replacing current one
<b>Replace All</b>	replaces all found items
<b>Abort search</b>	aborts this command
View/Edit buffer for PLD	
<b>Ctrl+F2</b>	erase buffer with specified blank value
<b>Ctrl+Shift+F2</b>	fill buffer with random data
<b>F9</b>	goto address...
<b>F10</b>	change mode view / edit
<b>F11</b>	switch the mode of buffer data view between 1 bit and 8 bit view. It can be also do by mouse clicking on the button to the right of View/Edit mode buffer indicator. This button indicates actual data view mode (1 bit or 8 bit), too.
<b>Arrow keys</b>	move cursor up, down, right and left
<b>Home/End</b>	jump on start / end current line
<b>PgUp/PgDn</b>	jump on previous / next page
<b>Ctrl+PgUp/PgDn</b>	jump on start / end current page
<b>Ctrl+Home/End</b>	jump on start / end edit area
<b>Backspace</b>	move cursor one position left (back)

**Note:** Characters 0 and 1 immediately changes content of edit area.

## Buffer / Fill block

Selecting this command causes filling selected block of buffer by requested hex (or ASCII) string. Sets start and end block for filling and requested hex or ASCII string.

## Buffer / Copy block

This command is used to copy specified block of data in current buffer on new address. Target address needn't be out from source block addresses.

## Buffer / Move block

This command is used to move specified block of data in current buffer on new address. Target address needn't be out from source block addresses. Source address block (or part) will be filled by topical blank character.

## Buffer / Swap block

This command swaps a high- and low- order of byte pairs in current buffer block. This block must started on even address and must have an even number of bytes. If this conditions do not fulfil, the program modifies addresses itself (start address is moved on lower even address and/or end address is moved on higher odd address).

## Buffer / Erase

If this command is selected, the content of the buffer will be filled with topical blank character.

The reserved key <Ctrl+F2> will bring out this menu from any menu and any time.

## Buffer / Fill random data

If this command is selected, the content of the buffer will be filled with random data.

The reserved key <Shift+Ctrl+F2> will bring out this menu from any menu and any time.

## Buffer / Duplicate buffer

This command performs duplicate buffer content in range of source EPROM to range of destination EPROM. This procedure is suitable if there is used for example 27C512 EPROM to 27C256 EPROM position.

**Note:** *The procedure always uses buffer start address 00000h.*

## Buffer / Checksum

The checksum dialog is used for calculate checksums of selected block in buffer. The checksums are calculated by next way :

<b>Byte</b>	sum by bytes to "word". CY flag is ignored
<b>Word</b>	sum by words to "word". CY flag is ignored
<b>Byte (CY)</b>	sum by bytes to "word". CY flag is added to result.
<b>Word (CY)</b>	sum by words to "word". CY flag is added to result.
<b>CRC-CCITT</b>	sum by bytes to "word" using $RESULT=PREVIOUS+(x^{16}+x^{12}+x^5+1)$
<b>CRC-XModem</b>	sum by bytes to "word" using $RESULT=PREVIOUS+(x^{16}+x^{15}+x^2+1)$

Column marked as **Neg.** is a negation of checksum so, that  $Sum + Neg. = FFFFH$ .

Column marked as **Suppl.** is complement of checksum so, that  $Sum + Suppl. = 0 (+ carry)$ .

Dialog checksum contains following items:

**From address:** This is a start address of block selected for calculating checksums in buffer. Address is defined as Byte address.

**To address:** This is an end address of block selected for calculating checksums in buffer. Address is defined as Byte address.

**Insert checksum:** This is special item used for select which kind of checksum will be written into the buffer when, the **Calculate & insert** was executed.

**Insert address:** This is special item that specifies an address from the buffer where a result of chosen checksum will be written, when the **Calculate & insert** was executed. Address can not be specified inside the range <From address> to <To address>. Address is defined as Byte address.

**Size:** This item is used for setting a size of chosen checksum result, which will be written into the buffer. A size of checksum result may be 8 (byte) or 16 (word) bits long. If word size was selected, whole checksum value will be written into the buffer. In other case will be written only low byte of checksum value.

**Note:** *If word size was selected, a low byte of checksum value will be written on address specified in box Insert address and a high byte will be written on address incremented by one.*

**Calculate:** Click on the button Calculate starts calculating checksums for selected block in buffer. No writes into the buffer are executed.

**Calculate & insert:** Click on the button **Calculate & insert** starts calculating checksums for selected block in the buffer and writes the chosen checksum into the buffer on address specified by **Insert address**.

## Options

The Options menu contains commands that let you view and change various default settings.

### Options / General options

General options dialog allows user to control following options of program.

#### File options

File options page allows you to set file masks, auto-reload of current file and choose file format recognizing for loaded files.

**File format masks** is used for setting file-name masks to use as a filter for file listing in **File / Save** and **File / Load file** window for all file formats. Mask must contain one of wildcards (\*, ?) at least and must be applied correctly by syntax.

**Project file default extension** is used for setting project files-extension used as default extension in **File / Load project** and **File / Save project** dialogs.

In group **When current file is modified by another process** can be set mode of reloading of actually loaded (current) file. There are three choices:

1. Prompt before reloading file
2. Reload automatically
3. Ignore change scanning of current file

**Load file format** allows to set mode of file format recognition for loading files. When automatic file format is selected, program analyses format of loading file and test file for each of supported formats, that are available in program. If file format matches one of supported formats, the file is read to buffer in detected format.

Manual file format allows user to select explicitly wished file format from list of supported file formats. File may be loaded no completely or incorrectly, if file format does not match to user selected format.

#### Hex file options

This page contains several options for loading control by any of HEX formats.

The first option sets **erasing** buffer (with desired value) automatically before the loading by any of HEX formats.

The second option sets a **negative offset**, which is used for data addresses modification by loading from any HEX file so, that data can be written to existing buffer addresses.

#### Example:

A file contents data by Motorola S - format. A data block started at address FFFF0H. It is a S2 format with length of address array of 3 bytes. For all data reading you can set a value of negative offset to FFFF0H. It means, that the offset will be subtracted from current real addresses and so data will be written from buffer address 0.

**Warning:** *The value of negative offset is subtracted from real address and therefore a result of subtraction can be negative number. Because take care of correct setting of this value.*

The third option sets data **redirecting** from file by extended Intel HEX format to physically existing buffer addresses. It means, that the user will be prompted to enter segment, which is associated with the lowest segment in file (record type 02) and the other segments will be modified with this difference. This option isn't valid for "simple" Intel HEX format. Default set means inactive status.

#### Example:

The file contains two records of type 02 with addresses F000H and F800H. When we enter new segment for example 0H, data from segment F000H will be directed to segment 0H and similar data from segment F800H to segment 800H.

Default setting is no active.

#### Language

This page allows you to select another language for user interface such as menu, buttons, dialogs, information and messages. It also allows to select wished help file in another language. For another language support of user interface the language definition file is required.

#### Sound

Sound page allows user to select the sound mode of program. Program generates sounds after some activities, e.g. activities on device (programming, verifying, reading, etc.). Program generates sound also when warning or error message is displayed. User can now select sound from MS Windows system sound (required installed sound card), PC speaker or none sound.

#### Other

Page Other allows user to manage other program settings. In the panel Tool buttons, hint display options on toolbar buttons in main program window can be modified. In the panel Start-up directory can be selected mode of selecting directory when program starts. Default start-up directory means directory, from which program is called. Directory in which program was lastly ended means the last current directory when program was lastly ended. This directory assumes the first directory from directory history list.

### Save options

Save options page allows you to select the program options saving when exiting program. Three options are available here:

**Don't save options** - don't save options during quitting program and don't ask for saving options.

**Auto save options** - save options during quitting program without asking for saving options.

**Prompt for save options** - program asks user for saving options before quitting program. User can select to save or not to save options.

### Options / View

Use the View menu commands to display or hide different elements of program environment such as toolbars.

#### Options / View / Main toolbar

Choose this command to show or hide the Main toolbar.

#### Options / View / Additional toolbar

Choose this command to show or hide the Additional toolbar.

#### Options / View / Programming conditions before programming

Choose this command to enable/disable display of Programming conditions before device program operation is confirmed.

#### Options / Display errors

This option allows you set a form of errors displaying as a result of programmed data verifying. Errors can be displayed to the **screen** (max. 45 differences), **saved to VERIFY.ERR** file on the disk in current directory or it will **not displayed**. In case the displaying errors are turned off, the control program reports a warning message in INFO window only.

This setting can be saved to disk by command **Options / Save options**. Default form is set to a screen displaying.

#### Options / Find programmer

Selects a new type of programmer and communication parameters. This command contains following items:

**Programmer** - sets a new type of programmer for find. If a Search all is selected, the control program finds all supported programmers.

**Establish communication** - allows manual or automatic establishing communication for a new programmer.

**Speed** - sets speed, if a manual establishing communication is selected, which PC sends data into the programmer. Speed is expressed as a percent from a maximal speed.

The communication speed modification is important for PCs with “slow” LPT ports, which haven’t sufficient driving power for a PC<->programmer cable (laptop, notebook, ...). Use this command, if you have any communication problems with connected programmer on the LPT port of your PC (e.g. control program reports a programmer absence, the communication with the programmer is unreliable, etc.).

If automatic establishing communication is selected, then control program sets a maximal communication speed.

**Port** - selects a LPT port, which will be scanned for a requested programmer. If All port is selected, the control program scans all LPT ports, which are available on standard addresses.

**Address for special port** - sets address of LPT port, if a Special port is selected.

Pressing key <Enter> or button **OK** initiates scanning for programmer by set parameters. There is same activity as at start the control program. The command clears a list of default devices without the current device, if the new selected programmer supports this one.

This setting is saved to disk by command **Options / Save options**.

## Options / Handler

In dialog **Handler** a Handler type and Handler communication parameters can be set. Handler is an external device for special control of device operations in control program. When None Handler is selected, this means default state of control program, i.e. device operations are controlled directly by user otherwise control program is in special mode, when device operations are controlled automatically with co-operation with Handler.

Dialog **Handler** contains following items:

**Selected Handler** select wished Handler type.

**Search at port** select a COM port, which will be scanned for a requested Handler.

Pressing key <Enter> or button **OK** initiates scanning for Handler by set parameters. If selected Handler type is **None**, no Handler scanning will be processed. Current Handler settings are saved to configuration file by command **Options / Save options** or when control program is closed.

Handler is not available for sale.

## Options / Module options

This option is used for multiple socket programmers for defining MASTER socket and activity of each socket. **MASTER socket** group box allows user to set socket which is preferentially used for device reading operation. **Enable/Disable socket** checkbox array allows user to set enabling and disabling of each socket individually. Disabled sockets are ignored for any device operation.

## Options / Automatic YES!

This command is used for setting **Automatic YES!** mode. In this mode you just put a device into ZIF socket and a last operation will be repeated automatically. Program automatically detects an insertion of a new device and runs last executed operation without pressing any key or button. An insertion of device into ZIF is displayed on the screen. Repeated operation executing will be cancelled by pressing key <Esc> during waiting for insert/remove a device to/from ZIF.

**Note:** *During waiting for an insertion a new device into ZIF socket, the LED BUSY on the programmer is blinking.*

This mode may be enabled or disabled by item **Automatic YES!** mode. If a new programmer is selected **Options / Find programmer**, this mode will be disabled.

In **Response time** is possible to set a time interval within must be detected device in ZIF socket to accept an insertion of a new device. Default is set standard interval. If socket adapter is used then is recommended to set an elongated interval.

In **Pins with capacitors** bar may be entered a list of a pins interconnected by capacitors (for example: if a converter, which have connected capacitor between VCC and GND, is used), which may makes problems at detecting insertion of a new device.

List of pins of device is in form:

pinA, pinB, pinC....

**Example:** 4,6,17

This list is erased if a new device is selected by **Device / Select default** or **Device / Select device** ...

This setting is saved to disk by command **Options / Save options**.

## Options / Log file

This options associates with using of **Log window**. All reports for this window can be written into the **Log file** too. The Log file name as **REPORT.REP** and the control program creates this file in current directory.

Sets **New** caused deleting old Log file if exist and creating a new file for reports. Sets **Append** adds all reports into existing Log file. If file not exist, the new file will be created. Settings are applied only at program start.

This setting can be saved to disk by command **Options / Save options**. Default form is set to a using Log window without Log file i.e. all reports will be displayed to a Log window only.

## Options / Protected mode

Protected mode is special mode of program. When program is in Protected mode, there are disabled program operation and commands which can modify buffer or device settings. Protected mode is used for prevent operator from modify buffer or device settings due to insignificance. Protected mode is suitable for the programming of a large amount of the same type of devices.



There are two ways how to switch program to Protected mode:

1. by using menu command **Options / Protected mode**. This command displays password dialog. User have to enter password twice to confirm the password is correct. After password confirmation program switches to Protected mode. The entered password is then used to switch off Protected mode.
2. by reading project, which was previously saved in Protected mode.

To switch program from Protected mode to normal mode use the menu command **Options / Normal mode**. The "Password required" dialog appears. User have to enter the same password as the password entered during switch to Protected mode.

Other way to cancel Protected mode of program is closing of program, because program Protected mode is active until program is closed. The next program start will be to Normal (standard) mode (the only exception is case of project loaded by command line parameter name of project and the project was saved in Protected mode).

### Options / Save options

This command saves all settings that are currently supported for saving, even if auto-save is turned off. Following options are saved: options under the **Options** menu, ten last selected devices, file history, main program window position and size.

### Diagnostics

This command includes selftest for programmers and IC test.

#### Diagnostics / Selftest

Command executes a selftest of current programmer without diagnostic POD. **ZIF SOCKET MUST BE EMPTY TO EXECUTE THIS COMMAND**. We recommend execute also **Diagnostics / Selftest plus** of programmer.

#### Diagnostics / Selftest plus

Command executes a selftest of current programmer using diagnostic POD, which is included in standard delivery of programmer. We recommend run this test as often as possible, e.g. once per month.

#### Diagnostics / Create diagnostic report

Command Create Diagnostic report is used for writing more particular diagnostic information to **Log window** and consequently copy **Log window** content to clipboard. The Log window content can be placed from clipboard to any text editor. Diagnostic report is useful when error occurs in control program or programmer and kind of the error is, that user can not resolve it oneself and he must contact programmer manufacturer. In this case when customer send message to manufacturer about his problem it is good to send also diagnostic report. Diagnostic report can help manufacturer to localise the reason of error and resolve it sooner.

## Help

Pressing the <F1> key accesses the Help. When you selecting menu item and press <F1>, you access context-sensitive help. If PG4U / PG4UW is executing an operation with the programmer <F1> generates no response.

The following Help items are highlighted:

- words describing the keys referred to by the current Help
- all other significant words

current cross-references; click on this cross-reference to obtain further information.

Since the Help system is continuously updated together with the control program, it may contain information not included in this manual.

Detailed information on individual menu commands can be found in the integrated on-line Help.

**Note:** *Information provided in this manual is intended to be accurate at the moment of release, but we continuously improve all our products. Please consult [www.grifo.com](http://www.grifo.com) or [www.grifo.it](http://www.grifo.it).*

### Help / Supported devices

This command displays list of all devices supported by at least one type of all supported programmers. It is usefull especially when user wants to find any device supported by at least one type of programmers.

Prefix "g\_" before name of device means the device is supported by multi-socket programmer.

### Help / Supported programmers

This command displays information about programmers, where supported this program.

### Help / Device list (current programmer)

This command makes a list of all devices supported by current programmer and saves its to **?????DEV.txt** text file and **?????DEV.htm** HTML file in the directory where control program is run from. Marks **?????** are replaced by abbreviated name of current programmer, the device list is generated for.

### Help / Device list (cross reference)

This command makes cross reference list of all devices supported by all programmers available on market and supported by this control program. The resulting list is in HTML format and

consists of following files:

- one main HTML file **TOP\_DEV.htm** with supported device manufacturers listed
- partial HTML files with list of supported devices for each device manufacturer

Main HTML file is placed to directory where this control program for programmers is located. Partial HTML files are placed to subdirectory **DEV\_HTML** placed to the directory where control program for programmers is located.

## About

When you choose the Info command from the menu, a window appears, showing copyright and version information.

## SOFTWARE ADDITIONAL FEATURES

PG4U/PG4UW is common control program for all of the **grifo®** programmers. Thus, during work with him its possible to find some items, those refer not to current selected programmer.

There is possible to attach to selected file “device associated file” too. This file may be automatically loaded into the buffer, if item **Device options/Associated file/Automatic load** is set to Yes. File can be associated manually or accepting last loaded file into the buffer. For this purpose is used item **Device options/Associated file/File name**.

There exist feature of control program, named **Automatic YES!**. In concrete, it mean there isn't necessary to press “YES!” labeled button to repeat last activity, because auto-sensing feature of programmer detect insertion of new chip into ZIF socked and start last executed automatically. This feature can be set in menu **Options**. If you use package converter with come pins coupled by capacitor, enter list of these pins. **Automatic YES!** feature is available for **JetProg, UEP 48** and **MP PIK+** programmer only.

Some special devices (e.g. Philips Coolrunner family) require external DAT files, that isn't present in standard PG4U/PG4UW SW delivery on Mini-CD. If you need to program these devices, look at **www.grifo.com** or **www.grifo.it**.

## OTHER

Attention to multitasking OS's (WIN 3.11/95/98/Me/NT/2000/XP). There is needful for regular running of control program for any **grifo®** programmer that printer port, on which is programmer connected, must be reserved for this programmer only. Otherwise, any other program must not simultaneously to use (or any way to modify) this printer port.

PG4U/PG4UW SW can handle all modes of LPT port (full IEEE 1284 support), thus you don't need to configure LPT port for connection of **grifo®** programmers.

WIN98 have bug (or wittingly) in the MSDOS.SYS file. Initial setting of variable DoubleBuffer is 1, therefore DOS applications run slowly. Write please DoubleBuffer=0.

In case of WIN software, please don't move any window during BUSY LED is on - watching circuit can be activate to switch the programmer in safe status as in case communication PC-programmer error

## WARRANTY TERMS

Warranty guarantees on failure-free operating of the programmer and all its parts, materials and workmanship for three-years from the date of purchase. This warranty is limited to 25,000-cycles on DIL-ZIF socket or 10,000-cycles on PLCC-ZIF sockets). If the product is diagnosed as defective, the authorized repair center will repair or replace defective parts at no charge. Parts used for replacement and/or whole programmer are warranted only for the reminder of the original warranty period.

For repair within the warranty period, the customer must prove the date of purchase. All other programmer owners only carry a standard 6 months warranty.

The warranty does not apply to products that are of wear and tear or mechanically damaged. Equally, the warranty does not apply to products opened and/or repaired and/or altered by personnel not authorized by **grifo®**, or to products that have been misused, abused, accidentated or that were improperly installed.

Before sending the programmer, please contact **grifo®** directly to agree about the shipping modalities. However it is essential to return to **grifo®** the transport packing of the programmer in good state, the cable, the power supply and include a brief and exhaustive description of the problem encountered.

For unwarrantable repairs You will be billed according to the costs of replacement materials, service time and freight. **grifo®** or its distributors will determine whether the defective product should be repaired or replaced and judge whether or not the warranty applies.

Please also see “TROUBLESHOOTING” section.

## TROUBLESHOOTING

We really want you to enjoy our product. Nevertheless, problems can occur. In such cases please read carefully all the enclosed documentation again. Probably you will find the needed answer right away. Should the problem persist, please follow the instructions below.

### COMMUNICATION ERRORS

- Programmer must be supplied correctly, so the green supply LED must be ON and the power supply must be the one delivered with the programmer. To test the supply cable's physical integrity twist it delicately and see if something changes.
- Update the control program, often a more recent version includes enhancements also in communication reliability. Please refer to "FREE SOFTWARE UPDATES" paragraph for more information.
- Programmer must be connected directly to parallel port, no data switch, hardware keys or port duplicators can be put between PC and programmer.
- Parallel port must be completely dedicated to the programmer, under Windows it must not be shared as a network resource and no printer driver must use it.
- The printer port where the programmer is connected **must be** configured as IEEE 1284 port. You should check in the BIOS configuration of Your computer that the port is set correctly, or, if You use a multi I/O card, assure that it can be configured and used as IEEE 1284.
- The parallel cable must be the one delivered with the programmer or a printer cable **with all the pins connected**. To test the cable's physical integrity twist it delicately when PG4U/PG4UW tries to connect to the programmer and see if something changes.
- Try to install programmer and PG4U/PG4UW on another computer. If your system works normally on the other computer you might have a problem with the first one PC. Compare differences between both computers.
- Try to set the parallel port as Centronics compatible. This may decrease communication efficiency but assures the greatest compatibility level Your PC can give.

### READING OR PROGRAMMING PROBLEMS

- Assure the correct alignment of target device in the socket. Check the serigraph on the programmer or see the adapter serigraph if using one.
- Target device name must be selected on PG4U/PG4UW before attempting to read or to program it.
- Target device name and selected device on PG4U/PG4UW **MUST BE EXACTLY THE SAME !** If You attempted to program or to read a device with a name different or just similar to the target device name You may have damaged both the target device and the programmer.
- Update the control program. Electronic manufacturers create new devices almost everyday so it is essential to search target device names in the latest version of PG4U/PG4UW. If You still can't find Your target device please refer to paragraph "UNSUPPORTED TARGET DEVICE".
- If the target device has never been programmed before or has been deleted it has returned to blank state. Perform "Blank check test" through Device/Blank Check menu or the keyboard shortcut <F6>.

- **MP PIK+** can perform a self test to assure the correct working of its internal components. Select menu Diagnostics and follow the on screen instructions. Please remark that self test must be performed with **ZIF socket empty**, otherwise both the programmer and the device inserted will be damaged, and that self test plus must be performed with the specific diagnostic POD inserted in the ZIF socket.

Please refer to figure 7 to see how to insert correctly the diagnostic POD.

- Some devices protect the content of their EPROM through protection fuses. These fuses must be disengaged (using the specific options in Device/Device Options and Device/Program menus). Otherwise programming or reading one of these devices results impossible, even if the device itself, the programmer and the PC are working correctly.

## **UNSUPPORTED TARGET DEVICE**

If You need to operate on a target device not supported by the control program for programmer, please do not despair and follow the next steps:

- Look in the device list of the latest version of the control program on our Internet site (section Download, file corresponded to your programmer). Your new target device might already be included in this version! If yes, download the file PG4UARC.EXE (for DOS) or PG4UWARC.EXE (for WIN) and install the new version of the control program.
- Contact **grifo®** directly. We may need detailed data sheets of your target device and, if possible, samples. The samples will be returned to you after we include your target device in a new version of PG4U.

## ADDITIONAL TOOLS

To extend easily the already large amount of devices programmable through the **grifo®** programmers, the package converters allow to use also packaging types different from the classic DIL. For more information about which converter can be used with your programmer please contact **grifo®** directly.

PACKAGE CONVERTER	DESCRIPTION
DIL20/PLCC20 ZIF	Package converter from DIL 20 to PLCC 20 ZIF socket.
DIL28/PLCC28 ZIF	Package converter from DIL 28 to PLCC 28 ZIF socket.
DIL32/PLCC32 ZIF	Package converter from DIL 32 to PLCC 32 ZIF socket.
DIL40/PLCC44 ZIF	Package converter from DIL 40 to PLCC 44 ZIF socket.
DILxx/PLCCyy ZIF	Package converter from DIL xx to PLCC yy ZIF-CS socket.
DIL16/SOIC16 150 mil	Package converter from DIL 16 to SOIC 16 type 150 mil ZIF socket.
DIL16/SOIC16 200 mil	Package converter from DIL 16 to SOIC 16 type 200 mil ZIF socket.
DIL20/SOIC20 300 mil	Package converter from DIL 20 to SOIC 20 type 300 mil ZIF socket.
DIL24/SOIC24 300 mil	Package converter from DIL 24 to SOIC 24 type 300 mil ZIF socket.
DIL28/SOIC28 300 mil	Package converter from DIL 28 to SOIC 28 type 300 mil ZIF socket.
DIL34/SOIC34 200 mil	Package converter from DIL 34 to SOIC 34 type 200 mil ZIF socket.

**FIGURE 13: PACKAGE CONVERTER LIST**

## ALPHABETICAL INDEX

**A**

ADDITIONAL TOOLS 101

**B**

BUFFER 54

**C**

CALIBRATION 67

CHECKSUM 89

COMMUNICATION ERRORS 99

CONNECTION 66

CONTROL PROGRAM COMMANDS 74

ABOUT 97

BUFFER 86

DEVICE 77

FILE 74

HELP 96

OPTIONS 90

**D**

DAC 58

DEVICE 54

DEVICE SUPPORT 58

DIAGNOSTICS 95

**G**

GENERAL INTRODUCTION 53

**H**

HEX 54, 74

HOT KEYS 73

**I**

IEEE 1284 58, 66

IN-SYSTEM PROGRAMMING 62

INSTALLATION 64

ISP 54, 62





**L**

LEDS **66**  
  BUSY **66**  
  POWER **66**  
LPT **64**

**P**

PC CONFIGURATION **67**  
PIN DRIVERS **58**  
PORT **54**  
PRINTER **54**  
PROJECT **75**

**Q**

QUICKSTART **69**

**R**

READING OR PROGRAMMING PROBLEMS **99**

**S**

SELFTEST **67, 95**  
SOFTWARE DESCRIPTION **72**  
SPEED **58**

**T**

TROUBLESHOOTING **99**

**U**

UNSUPPORTED TARGET DEVICE **100**

**W**

WARRANTY TERMS **98**

**Z**

ZIF **54**

