Contents

Chapter 1

Introduction

gputils is a collection of tools for Microchip (TM) PIC microcontrollers. It includes gpal, gpasm, gplink, and gplib. Each tool is intended to be an open source replacement for a corresponding Microchip (TM) tool. This manual covers the basics of running the tools. For more details on a microcontroller, consult the manual for the specific PICmicro product that you are using.

This document is part of gputils.

gputils is free software; you can redistribute it and/or modify it under the terds of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.

gputils is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;

Chapter 2

gpal

2.1 Introduction

gpal is a compiler for Microchip

a PIC executable or gplib to produce an

2.5.2 Numbers

gpal uses decimal as its default radix. The following table summarizes other supported num

base	general syntax	21 decimal written as

2.6 Statements

2.6.1 If

```
if <expression> then
  <statements>
[elsif <expression> then
  <statements>]*
[else
  <statements>]?
end if;
```

The statements in each block are executed if the expression is true. Here is an example:

```
if i < 10 then
  j = 5;
elsif
```

2.6.9 Pragma

```
pragma <anything>;
```

Pragmas provide data to compiler which is outside of its legal syntax. The table below summarizes the pragmas available:

CHAPTER

Chapter 3

3.1.1 Using gpasm with “make”

On most operating systems, you can build a project using the make utility.

base	general syntax	21 decimal written as

MACRO

```
<label> MACRO [ <symbol> [ , <symbol> ]* ]
```

Declares a macro with name <label>. gpasm replaces any occurrences of <symbol> in the macro definition with the parameters given at macro invocation.
See also: LOCAL, ENDM

MESSG

```
MESSG <string>
```

Writes <string> to the list file, and to the standard error output.
See also: ERROR

NOEXPAND

```
NOEXPAND
```

Turn off macro expansion in the list file.
See also: EXPAND

NOLIST

```
NOLIST
```

Disables list file output.
See also: LIST

ORG

```
ORG <expression>
```

Sets the location at which instructions will be placed. If the source file does not specify an address with ORG, gpasm assumes an ORG of zero.

PAGE

```
PAGE
```

Causes the list file to advance to the next paRe.
See also: LIST

PAGESEL

```
PAGESEL <label>
GOTO <label>
```

This directive will generate paReselecting code to set the paRebits to the paRecontaining<label>

PROCESSOR

```
PROCESSOR <symbol>
```

Selects the target processor. See section **??** for more details.
See also: LIST

RADIX

```
RADIX <symbol>
```

Selects the default radix from “oct” for octal, “dec” “dec”

UDATA_ACS

```
<label> UDATA_ACS <expression>
```

Only for relocatable mode. Creates a new uninitialized accessbank data secaion in the output objeca file. <label> specifies the name of the secaion. If <label> is not specified the default name ".udata_acs" will be used. <expression> is opaional and specifies the absolute address of the secaion.

See also: CODE, IDATA, UDATA

UDATA_OVR

```
<label> UDATA_OVR <expression>
```

Only for relocatable mode. Creates a new uninitialized overlaid data secaion in the output objeca file. <label> specifies the name of the secaion. If <label> is not specified the default name ".udata_ovr" will be used. <expression> is opaional and specifies the absolute address of the secaion.

See also: CODE, IDATA, UDATA

UDATA_SHR

```
<label> UDATA_SHR <expression>
```

Only for relocatable mode. Creates a new uninitialized sharebank data secaion in the output objeca file. <label> specifies the name of the secaion. If <label> is not specified the default name ".udata_shr" will be used. <expression> is opaional and specifies the absolute address of the secaion.

See also: CODE, IDATA, UDATA

VARIABLE

```
VARIABLE <label>[=<expression>, <label>[=<expression>]]*
```

Delcares variable with the name <label>. The value of <label> may later be reassigned. The value of <label> does not have to be assigned at declaraaion.

See also: CONSTANT

WHILE

```
WHILE <expression>
```

Performs loop while <expression> is true.

See also: ENDW

3.4 Instrucaions

3.4.1 Instruction set summary

12 bit Devices (PIC12C5XX)

Syntax	Description
ADDWF <f>,<dst>	Add W

Spe

The s are:

SETZ	Set zero
MOVFW <f>	Move file to W
NEGF <f>	

115 Duplicate Label

Duplicate label or redefining a symbol that can not be redefined.

124 Illegal Argument

gpasm encountered an illegal argument in an expression.

125 Illegal Condition

An illegal condition like a missing ENDIF or ENDW has been encountered.

126 Argument out of

3.5.2 Warnings

201

The ID locations value specified is too large.

305 Using default destination of

Chapter

If the user does not specify a linker script, gplink will

Chapter 5

gplib

gplib creates, modifies and extracts COFF archi

Chapter 6

Utilities

6.1 gpdasm

gpdasm is a disassembler for gputils. It converts hex files generated by gpasm and gplink into disassembled instructions.

6.1.1 Running gpdasm

The general syntax for running gpdasm is

```
gpdasm [options] hex-file
```

Where options can be one of:

Option	Meaning
h	Display the help message.
i	Display hex file information
l	List supported processors.
m	Memory dump hex file.
p<processor>	Select processor
s	

6.2 gpvc

gpvc is cod file viewer for gputils. It provides an easy way to view the contents of the cod files generated by gpasm and gplink.

6.2.1 Running gpvc

The general syntax for running gpvc is

```
gpvc [options] cod-file
```

Where options can be one of:

Option	Meaning
a	Display all information
d	Display directory header
s	Display symbols
h	Show the help message.
r	Display ROM
l	Display source listing
m	Display debug message area
v	Print gpvc version information and exit.

Index