

PICSim

Ioannes Petroglou

Copyright © 1996-1997 Ioannes Petroulou

COLLABORATORS

	<i>TITLE :</i> PICSim		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Ioannes Petroglou	December 31, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	PICSim	1
1.1	PICSim Manual	1
1.2	What's new?	2
1.3	Overview	4
1.4	short	4
1.5	Registration	5
1.6	Support	6
1.7	Installation	6
1.8	notes	6
1.9	Using	6
1.10	System requirements	8
1.11	Questions and answers	8
1.12	Assembler	8
1.13	Copyright	8
1.14	Bug reports	9
1.15	Credits	10
1.16	The author	10
1.17	History	10
1.18	Project	10
1.19	Controls	11
1.20	Device	12
1.21	Reset	12
1.22	Tools	13
1.23	Preferences	13
1.24	keys	13

Chapter 1

PICSim

1.1 PICSim Manual

```

                                PICSim
                                Version 1.4.1

What's new?
                                A Pic Simulator
                                for Amiga computers

© Copyright 1996-1997 Ioannes Petroglou

                                18.12.1997

short
  short overview

Overview
  What is PICSim?

Registration
  Limitations of the unregistered version

Support
  Call me!

Installation
  How to install PICSim

Requirements
  System Requirement

Using PICSim
  The main part
```

Keytable
The Keytable

Notes on the simulation
Things to pay attention to

Questions and answers
Solutions to common problems

Copyright
Legal mush

Bug reports
How to report bugs

Credits
The author wishes to thank...

The author
Programmer's address

History
Revision history of PICSim

1.2 What's new?

third release:

- listview font select
- cpu part about 30% faster
- PICAsm loader
- Watchdog support
- source overwork

fourth release:

- added symbol manager
- added watched registers window
- added conditionl register break
- ...?

fifth release:

- reworked programcounter access (16c84 bug with addwf)
- interrupt support for 16c84
- reworked registers window
- keyfile for easy updates
- ...?

sixth release:

- history buffer fixed

seventh release:

- PIC16c84 EMU hardware support (prefs saveable)
- function key toggle support

- file register font size select (prefs saveable)
- file register font select (prefs saveable)

- added xtal directive
- no xtal reset anymore

fixed 16c84 commands:

- addlw
- sublw
- decf

eighth release:

- fixed watch window
- source overwork
- added turbo mode:
 - F5 - Same as F1 but Faster. Warning: In this "Turbo" Mode, the History Function, Time and Watchdog Calculation are Disabled.
 - Usefull for maximum excecution speed at using the Port Hardware.

this release:

- addlw carry bit bug fixed
 - 16c84 mnemonic jumtable optimized
 - archive includes 68040/060 Version of PICSim
-

1.3 Overview

PICSim is a versatile software simulator for Microchip PIC16C5X and 16C84 series microcontrollers. It allows the user to simulate their code on Amiga to verify its proper operation.

PICSim uses the listfile or the objectfile that is produced by various assemblers.

PICSim requires an Amiga or compatible computer (OS3.0 or above) with at least some bytes of free memory to run.

1.4 short

Simulate

- load a listfile (project/load/PICAsm)
- execute single cycle with "RETURN"
- animated execute with "F2"
- running without screen update with "F2"
- stop executing with any key
- run until pc + 1 with "F3" (usefull with the call command)
- press "Amiga i" to activate interrupt. If you have the 16c84 selected and the GIE bit is set (register 0B, bit 7) PICSim will call the interrupt vektor 0004 after two additional cycles. The GIE bit is cleared until the return from interrupt command (retfie)
- with cursor up and down you can walk through the listview. The actuall program counter is highlighted in the selected colour.

The HELP key causes a jump back to actuall programcounter, the RETURN key causes the program counter to set on the actuall position.

History

with cursor right and left you can walk through the history buffer. Maximum history buffers can be setted in the preference menu.

The history steps can be increased pressing the shift key. The step size can set in the preference menu.

Register window

here are the registers from the selected CPU. Changed registers are highlighted in the selected colour. you can increase, decrease registers or set and clear bits by clicking on it with the right and left mouse button.

You can enter a value clicking on the register while pressing the shift key.

1.5 Registration

PICSim is shareware. To ease your decision whether to pay, the unregistered version has one limitation:

- PIC size is limited to 64 bytes

The shareware fee is DM 40,-. Registered users will receive a personalized keyfile with the missing function and support for the assembler.

How do you become a registered user?

The simple way is to fill in the file "OrderForm", to print it out, sign it (in this order, if possible), put it in an envelope together with the registration fee and send it to

me

. I won't

accept order forms which are not signed.

PLEASE DON'T SEND CHEQUES! I would prefer if you send me an International Postal Money Order.

snail mail:

Dirk Düsterberg
Jahnstr. 9
31860 Emmerthal
Germany

or Bank Draft:

Name : Joannis Petroglu
Institut : STADTSPARKASSE BAD PYRMONT
BLZ : 254 513 45
Konto-Nr. : 10346708

I will send you the program per EMAIL, if you didn't want this or

if there is no EMAIL adress I will sent the program on disk to your postal address. The shipment on disk may take 4..5 weeks, please be patient! Shipping per Disk costs 10 DM extra.

1.6 Support

The official PICSim homepage in the WWW has always the latest version and other information related to PICSim:

<http://linux.rz.fh-hannover.de/~duesterb>

1.7 Installation

just copy the files in a directory on your harddisk

1.8 notes

no notes

1.9 Using

PICSim determines the device type being simulated (e.g., PIC16C54, ←
55, 56, 57,
84) by a variety of methods. The recommended method is to include the "device" directive in source code (see assembler instructions for details). The simulator locates the "device" directive in the list file and sets the device type accordingly.

Another method to select the device type is to select it in the pulldown menu.

It may be necessary to load the listfile again if you switch from a 5x type to the 16c84.

You may scroll through the source code displayed in the listview using the up and down arrow keys, you can jump with the shift key. The Current position the programcounter address, and breakpoints are marked different. When you press the up or down arrow keys, you can scroll up or down through the code. The programcounter is marked white when you scroll off of the current program counter line. To set a breakpoint in your code, you may scroll through the code until the desired line is highlighted at the center, then press the space bar. This will set a breakpoint at that line and denote this by highlighting it in blue. There are no limits to the number of breakpoints that

you can set.

To alter a register during simulation, you can move the mouse to the register. Then press the left mouse button to increment the register contents or the right mouse button to decrement it. The upper and lower nibbles of a register can be incremented or decremented separately for those that are displayed in hex. For those that are displayed in binary, you may alter each bit separately. Some of the registers are not allowed to change. You may not change the indirect address register (f0). The indirect address is not physically implemented in the processor and altering it would have no effect. Also, if the selected device is a PIC54 or 55 you may not change the upper two bits of the program counter, since these devices do not implement these bits. For the PIC56, you are not allowed to alter the upper bit of the program counter.

You can also press the shift key and click on the desired register, now you can give in the value with the keyboard.

The menus

- Project
- Controls
- Device
- Reset
- Tools
- Preferences
- SPACE.

Toggle the breakpoint at the highlighted line. To insert a breakpoint at a specific line, use the cursor keys to scroll the program display up or down until the desired line is displayed in the center highlighted bar, then press the space bar. The line will then turn blue to indicate that the breakpoint is set. Pressing space again on the same line will clear the breakpoint.

BACK STEP. Pressing the left arrow key steps backward one line in your program. The entire state of the microcontroller is stored in a 1000 element history buffer to allow the user to step back through the code up to 1000 steps. The right arrow key steps forward in the history buffer. You can jump to end of history with "e" and to start with the "s" button.

HELP. On pressing the HELP key the listview jumps to the actually Program counter.

1.10 System requirements

Requirements:

- The Amiga must have at least a 68020 processor. PICSim will run on every Amiga 1200/3000/4000, but not on stock Amiga 500/2000.
- 2MB of free memory are necessary to use over 1000 history buffers
- PICSim runs under AmigaOS 3.0 and higher

1.11 Questions and answers

Question:

Where is the assembler ?

Answer:

Assembler

Question:

What to do when PICSim ignores the Pic type ?

Answer:

Choose the correct Pic type in the device menu and load the listfile again.

1.12 Assembler

PICAsm has an own guide, please look there.

1.13 Copyright

The programs "PICSim", "PicProgger", "PicASM" may be freely distributed as long as they remain unchanged (archiving and packing are allowed).

No profit must be made by distributing PICSim, especially the price of a disk containing PICSim may not exceed US\$ 5,- (or equivalent amounts in other currencies).

Please feel free to distribute PICSim over bulletin board systems and networks as part of shareware/freeware CD-ROMs. All rights for commercial use remain at the

author

.

The Program that registered users will receive, must only be installed one one computer and in no case passed on to others. Offences will result in penal prosecution by me. With your signature on the order form, you accept these conditions.

The program is presented to the users as it is, without any warranty of any kind, be it expressed or implicit. Anyone using this program agrees to incur the risk of using it for himself. In no way can the author be made responsible for any damage directly or indirectly caused by the use or misuse of the program.

The user interface of the program was designed with GadToolsBox
© Copyright 1991-1993 Jaba Development.

Parts of the program are © Copyright 1992-1993 Jaba Development.

"Amiga" and "Commodore" are registered trademarks.

Names of other hardware and software items mentioned in this manual and in program texts are in most cases registered trade marks of the respective companies and not marked as such. So the lack of such a note may not be used as an indication that these names are free.

1.14 Bug reports

If you find a bug or a misfeature in PICSim, or have an idea how to make some things better, then please drop me a note so I'll be able to improve PICSim in the future. My address can be found here

.

Important for a bug report is the following information:

- Version of PICSim

- Used AmigaOS version (e.g. 3.1, 3.0 etc.)
- Used Listfile, objectfile
- Installed hardware, if of interest for the problem
- Information about installed startup programs on the Amiga
- Detailed description what program produces the bug and how it can be reproduced

But first please look
here
if there's a solution to your problem.

1.15 Credits

- Stephen Marsden with his EPIC1.2.lha where the IBMKEY25 example came from

1.16 The author

There was no Pic Simulator on Amiga.
So I had to do it. :-)

My address is:

Mail:
petroglo@unixserv.rz.fh-hannover.de

WWW:
<http://linux.rz.fh-hannover.de/~duesterb/>

Questions, criticism, suggestions and
bug reports
are always welcome.

1.17 History

1.18 Project

LOAD >> PICAsm this loads a PICAsm listfile
Parallax this loads a Parallax listfile
MPASM this loads a Microchip listfile

INHX8M this loads and disassembles a Intelhex file

Save as Source Save the Hexobjectfile as source code
Labels are the word addresses from Pic, it works
only on Pics with one Page correctly, because every
jump or call determines in the actually Page.

About Program Information

Quit Quit the Program. A pop-up prompt will allow you to confirm
the quit command.

1.19 Controls

Run F1. This causes the simulator to execute the code
until a breakpoint is reached or any key is pressed.
The screen is not updated until execution is stopped.
The execution starts at the marked programcounter line.

Go Pressing F2 will start the simulator running, and will
not stop until a key is pressed or a breakpoint is reached.
The screen is updated and the changes are highlighted after
each line is executed. The execution starts at the marked
programcounter line.

Run to PC+1 F3. execute the code without screen update until the next
programcounter is reached. Usefull with loops and calls.

Single Step This causes one line to be executed. The changes in the registers

are highlighted after the line is executed. If you scrolled through the code the current list address is taken for the programcounter. Pressing F4 causes executing single lines repeatedly.

1.20 Device

16C54	18pin Pic Type, 2 Ports (PortA=4bit, PortB=8bit), 512 * 12bit
16C55	28pin Pic Type, 3 Ports (PortA=4bit, PortB and PortC=8bit), 512 *12bit
16C56	18pin Pic Type, 2 Ports (PortA=4bit, PortB=8bit), 1024 * 12bit
16C57	28pin Pic Type, 3 Ports (PortA=4bit, PortB and PortC=8bit), 2048 *12 bit ←
16C84	18pin Pic Type, 2 Ports (PortA=4bit, PortB=8bit), 1024 * 14bit Program and Data EEPROM

1.21 Reset

Pic	This simulates a hardware reset.
Time	This resets the Time counter
Cycles	This resets the cycle counter
File Register	This resets the file register
XTal	setting up the XTal
Watchdog	resets the Watchdogtimer
Breakpoints	clear all set breakpoints

1.22 Tools

Convert converts ascii, binary, hex and dezimal

Stack Window shows on 16C84 all 8 Stacks

EEProm Window shows the EEPROM Window on 16C84

Symbol Manager manages the symbols
(conditional break, register watching)

Watch Window shows selected symbols with their values

Random FR fills the fileregister with random values

1.23 Preferences

Screen >> Select Mode requester to choose screen mode

 Use Public uses the public screen

 Palette Palette Preferences

Listview Font selects Font for the Listview

Register Font selects Font for the Register window

Settings switch and adjust several functions

Save Saves the Preferences, PicType and Window positions

1.24 keys

PICSim Key`s

----- Project ←

RAMIGA + _ Load PICAsm listfile (Symbol data supported)

RAMIGA + L Load Parallax listfile

RAMIGA + M Load Microchip listfile

RAMIGA + O Load and disassemble IntelHex8 Object file.

RAMIGA + - Same as Load PICAsm listfile, but without Filerequester.
 Your watched symbol list stays alive, symbol data will be updated.

RAMIGA + S Save ASM source, created by disassembler (Load INHX8M).
RAMIGA + Q Open quit PICSim requester.

----- Instruction Execution ↔

RETURN - Execute single PIC Instruction with Screen Update, no stop at breakpoint ↔

F1 - RUN
Execute PIC Instructions with no Screen update, stop at breakpoint.
Use F1 or ESC to Stop execution.

F2 - GO
Execute PIC Instructions with Screen update, stop at breakpoint.
Use F2 or ESC to Stop execution.

F3 - Run to PC+1
Execute PIC Instructions with no Screen update, stop at breakpoint.
Use F3 or ESC to Stop execution.
Usefull for subroutine (CALL instruction) debugging.

F4 - Hold F4 to Execute single PIC Instruction, stop at breakpoint.
Release F4 to Stop execution.

F5 - Same as F1 but Faster. Warning: In this "Turbo" Mode, the History
Function, Time and Watchdog Calculation are Disabled.
Usefull for maximum excecution speed at using the Port Hardware.

RAMIGA + I - Initiated Interrupt sequence, works only on PIC16C84.

----- Edit/Positioning functions ↔

SPACE - Set/Clear Breakpoint
CURSOR UP - Set "Cursor" to PC-1
CURSOR DOWN - Set "Cursor" to PC+1
CURSOR UP + SHIFT - Set "Cursor" to PC-Stepsize
CURSOR DOWN + SHIFT - Set "Cursor" to PC+Stepsize
HELP - Reset "Cursor" to actual PC

----- History related ↔

S - Go to History start.
E - Go to History end.
CURSOR LEFT - One step back into Execution History.
CURSOR RIGHT - One step forward into Execution History.
CURSOR LEFT + SHIFT - X step's back into Execution History.
CURSOR RIGHT + SHIFT - X step's forward into Execution History.
(Use Preferences->Settings window to set "X" Stepsize)

----- Device Select ↔

(Listview Window must be active)

RAMIGA + 1 - PIC16C54
RAMIGA + 2 - PIC16C55
RAMIGA + 3 - PIC16C56

RAMIGA + 4 - PIC16C57
RAMIGA + 5 - PIC16C84

----- Reset functions ↔

(Listview Window must be active)

RAMIGA + R - Reset the PIC (Fileregs, Specialregisters, time, watchdogtimer, ↔
cycles)

Clears the Historybuffers.

RAMIGA + T - Reset Time to zero.

RAMIGA + C - Reset Cycles to zero.

RAMIGA + F - Reset Fileregisters into defined state.

RAMIGA + X - Set new oscillator frequency.

RAMIGA + W - Reset Watchdog timer to zero.

RAMIGA + B - Removes all Breakpoints.

----- Tools related ↔

(Listview Window must be active)

RAMIGA + N Open converter Window. Allows converting of decimal, hexadecimal,
binary and ascii values.

RAMIGA + U Open Stack Window (16C84 only)

RAMIGA + E Open EEPROM Window (16C84 only)

RAMIGA + J Open Symbol Manager. Use it to build your Watched Symbols list.

RAMIGA + Y Open Symbol Watch Window. Shows the Symbol list, created by Symbol ↔
Manager.

RAMIGA + Z Set Random values into Fileregisters, excepting the special registers.

RAMIGA + . Start Editor, defined by Preferences->Settings->Editor.

The file to open is taken from PICASM listfilename + .asm suffix.

----- Preferences ↔

RAMIGA + D Open Screenmode Requester

RAMIGA + P Force PICSim to public WB screen.

RAMIGA + K Open Palette Window.

RAMIGA + A Open advanced settings window.

RAMIGA + T Save actual PICSim settings.