

**in**

Sander ter Steege

**COLLABORATORS**

	<i>TITLE :</i> in		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Sander ter Steege	April 13, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

---

# Contents

<b>1</b>	<b>in</b>	<b>1</b>
1.1	Welcome to the standalone PIC12C508 programmer guide . . . . .	1
1.2	DISCLAIMER . . . . .	2
1.3	introduction . . . . .	2
1.4	parts . . . . .	3
1.5	hardware description . . . . .	4
1.6	building the hardware . . . . .	5
1.7	adjusting the voltages . . . . .	5
1.8	operating the programmer . . . . .	6
1.9	standalone programming . . . . .	7
1.10	status led explanation . . . . .	7
1.11	assembler . . . . .	7
1.12	the calibration location . . . . .	8
1.13	cardware . . . . .	8
1.14	history . . . . .	8
1.15	todo list . . . . .	9
1.16	author and bug reports . . . . .	9

---

# Chapter 1

## in

### 1.1 Welcome to the standalone PIC12C508 programmer guide

MiniPIC 1.2

Standalone pic12c508 programmer for the Amiga  
Developed by Dennis van Weeren (d.vanweeren@worldonline.nl)  
on the amazing A1200T.  
© 1998,1999 Dennis van Weeren  
AmigaGuide by Sander ter Steege

This Package is

CARDWARE

Please read this document entirely before building or using  
this programmer!

DISCLAIMER

Contents:

Introduction

Partlist

Hardware description

Building the Hardware

Adjusting the voltages

Operating the Programmer

Standalone Programming

Status LED explanation

Assembler

The Calibration location

CARDWARE

History

Todo List (small list)

Author and Bug reports

## 1.2 DISCLAIMER

```
***** IMPORTANT! *****
This Software and Hardware description is provided "as is".
I do not take any responsibility for any damage caused to you, your
computer or your goldfish. Toasted harddisks, burnt CPU'S, exploded PIC's
or burnt fingers, it's not my responsibility! So use this entirely
at YOUR OWN RISK!
*****
```

## 1.3 introduction

Introduction to release 1.2

This Archive contains the schematics and software for a standalone programmer for the PIC12C508 microcontroller from MICROCHIP. It has been designed to operate independently from the Amiga and to be fully compliant to the programming specifications layed down by MIRCOCHIP.

It's main features are:

- latest production Quality Programming algorithm
- 2400 baud serial interfacing with Amiga
- Standalone operation makes it possible to program large series while leaving the Amiga free.
- on board EEPROM with 16bit checksum to store programfile for maximum reliability
- automatic handling of the Calibration value at 0x1ff for foolproof operation.
- supports standard INHX8M HEX-files.

This archive should contain:

```
Connections.pic           ;ic layouts
Connections.pic.info
Controll2                 ;download utility
Controll2_tolerant       ;modified download utility
```

```
control12_tolerant.readme ;Note
PIC12.inc ;PIC12C508 assemble include file
PIC16C84programfile.hex ;HEX file for the PIC16C84
PIC16C84programfile.readme ;Note
Power.pic ;schematics of powersupply
Power.pic.info
Programmer.pic ;main schematics
Programmer.pic.info
MiniPIC.guide ;this document
MiniPIC.guide.info
MiniPIC.doc ;text version of this document
MiniPIC.doc.info
```

## 1.4 parts

### Needed Parts

These are the parts you need to build the miniPIC project:

#### Semiconductors:

- 1 Microchip PIC16C84 microcontroller
- 1 MAX232 or ICL232 RS232 converter
- 1 CA3130 OpAmp
- 1 24C16 serial EEPROM
- 1 SN 74LS145 BCD to Decimal converter
- 2 LM317T Voltage regulator
- 1 7805 Voltage regulator
- 2 BC547B Transistor
- 1 BS170 N-channel mosfet
- 1 BS250 P-channel mosfet
- 1 1N4002 Diode
- 4 BAT85 Diodes
- 1 Duoled red/green or 2 seperate led's
- 1 12V/400mW ZenerDiode

#### Condensensors:

- 8 100nF ceramic multilayer condensensors
- 1 220 $\mu$ F/35V electrolytic condensor
- 5 10 $\mu$ F/16V electrolytic condensensors
- 2 15pF ceramic condensensors

#### Resistors carbon or better, 0.25W/5% :

- 4 3.3K
- 4 220
- 2 22K
- 1 47K
- 2 10K
- 1 4.7K
- 1 1.8K
- 1 100

- 1 470
- 1 680

Adjustable (with a screwdriver) resistors:

- 1 250 Ohms
- 3 500 Ohms

And the rest:

- 1 3.579545Mhz crystal
- 1 printed circuit board (experiment board)
- sockets for the DIL IC's
- Ziff socket or something similar to hold the PIC12c508
- power connector for AC adapter
- simple switch button

## 1.5 hardware description

Hardware description

The Hardware is rather complex because of the standalone function and the production grade powersupply but anyone who has worked with pic's before should be able to build it.

The PIC16C84 forms the heart of the circuit, it holds the programming algorithm, I2C routines and the RS-232 routines.

The MAX232/ICL232 translates the Amiga's serial port logic levels (+/-12V) to the standard logic levels of 0 and +5V as used by the PIC16C84.

The 24C16 is an 2K\*8 serial EEPROM which is connected to the PIC16C84 through an I2C-bus.

It is used to store the ProgramFile for the PIC12C508.

The 74LS145 together with an LM317T and a BC547B forms the programmable VDD powersupply for the PIC12C508.

The 74LS145 is a bcd to decimal decoder with 15V open collector outputs (you MUST use this type, no ls45, ls445 or HC/HCT types !).

It can select four voltages: 0V (off) ,VddMIN(typ. 3V), VddP( 5V) and VddMAX (typ. 5.5V).

The circuitry connected to pin RA4 is used to switch VPP (Programming voltage) on and off.

The circuitry connected to pin RB7,RB6,RB5 is used for the serial communications with the PIC12C508.

The opamp CA3130 is necessary to interface to the PIC16C84 because of the programmable VDD, I'm sorry I couldn't find a cheaper replacement for this, I tried several other opamps, but this was the only one which worked properly.

---

Only two parts remain, the 7805 which is used to provide a fixed +5V powersupply and the LM317T, which is used to provide +13.25V for the VPP (programming voltage).

## 1.6 building the hardware

### Building the Hardware

Building is straightforward, all needed info is included in this archive. The power needed is about 16-17volts at a current of 100mA (I have not tested the current consumption so this is just a guess), you can always use a non-regulated AC plug adapter at 12V because these usually deliver a much higher voltage. The maximum input voltage is about 30 Volts.

### Power Schematic

A couple of things need attention when building:

- Use a socket for at least the PIC16C84 for easy upgradeability and easy adjustment of the power supply.
- The power regulators don't need a heatsink, the 7805 can become warm but not hot.
- Use ceramic multilayer or siba types if possible for the 100nF decoupling capacitors.
- If you can get hold of a ICL232, it is to be preferred above a MAX232 (MAX232's can be very picky about their powersupply's!).
- Last note: most IC's used are CMOS types, so take ESD precautions !
- connections of the pic12c508:

VPP : programming voltage  
VSS : ground  
VDD : power supply

### Programmer Schematic

### Connections Schematic

- I have not designed nor tested this programmer as an in-system programmer, so use a zif socket or something like that

## 1.7 adjusting the voltages

Adjusting the voltages.

Adjusting the VPP is fairly straightforward, just place a multimeter at the output of the LM317T and adjust it with the trimmer to +13.25V

The VDD supply is a little bit more complicated.

Adjusting the programmable Vdd supply Step 1:  
Switch on power without the PIC16C84 inserted.

---



The 74LS145 will now output a 'low' on the Q3\_NOT output because a TTL-input goes to a logic one when left open. Now adjust the output of the LM317T (associated with the VDD supply) to VddMAX with the 5000ohms trimmer connected to Q3\_NOT.(typ. 5.5V)

Step2:

Connect a wire between input A of the 74LS145 and ground. The selected input code is 10 which is Q2\_NOT. Adjust the output of the LM317T (associated with the VDD supply) to VddP with the 5000ohms trimmer connected to Q2\_NOT.(5.0V)

Step3:

Connect a wire between input B of the 74LS145 and ground. The selected input code is 01 which is Q1\_NOT. Adjust the output of the LM317T (associated with the VDD supply) to VddMIN with the 2500ohms trimmer connected to Q1\_NOT.(typ. 3.0V)

After these adjustments have been made, switch of power (!) and insert all other IC'S.

Be sure to program the PIC16C84 before inserting!

## 1.8 operating the programmer

### Operating the Programmer

Included in this archive is a utility called "Controll2". It's a small cli (GUI is on the way) program used to download .hex files to the programmer. It is also used to perform a blank check (To see if a PIC is empty) and to program a pic online.

Downloading:

Select d. The program will ask for a .hex file. After loading the hex file the program will ask you the settings of the config register.

You must select:

- Codeprotection on/off.
- MCLR pin enable/disable.
- Watchdogtimer on/off.
- Oscillator type.

After typing 'y' to confirm, the program will download the .hex file and the configuration to the programmer. (Be sure it is swtiched on and online!)

You can always abort by ctrl\_c.

If the program reports "contact made" everything is going well.

Downloading is rather slow at the moment because of the 1200baud connection and the heavy handshaking, but it is reliable, so please be patient when downloading.

The status LED will light up green while downloading.

Blank Test:

Select 'b', the status LED will light red and the program will report the result of the test.

Online Programming:

---

Select 'p', the status LED will light red and the programming begins.

The program will report:

- 1) Blank Test result.
- 2) Programming result.
- 3) VddMIN verify result.
- 4) VddMAX verify result.
- 5) Configuration Programming result.

If an error occurs, the status LED will flash red and the program will report the failure.

If a fatal hardware error occurs, the status led will flash green and the program will hang.

abort with ctrl\_c.

## 1.9 standalone programming

Standalone Programming.

Because this is a standalone programmer, it can be used indepently from the Amiga.

When there is a valid .hex file in the programmer's eeprom memory, you can program a PIC12C508 simply by pressing the start button.

The LED will light red during programming, when it lights green again, programming was OK, when it flashes red, a programming error occurred.

You can insert and remove PIC's with the programmer switched on, as long as the LED is green, flashing red or flashing green.

So when the LED is continueosly red, the programmer is accessing the PIC and you can't remove the PIC.

## 1.10 status led explanation

Status LED explanation

The status LED indicates the programmer's status

red: the programmer is accessing the PIC (programming or blank testing).

green: the last action was succesfull.

flash green: A hardware error occurred (eeprom checksum failed or I2C error)  
you need to reset the programmer in this case.

flash red: the last action was not succesfull

## 1.11 assembler

Assembler

You'll be missing one important thing in this archive: the assembler!

That's right, I've not written an assembler for it.

This is because there are already very good PIC assemblers out there.

Any assembler capable of handling PIC16C5x processor's can handle the PIC12C508 too.

Just set the assembler (by example) to PIC16C54 and tada! the generated code is directly compatible with the PIC12C508. The PIC12C508 is based on the PIC16C5x family and therefore has the same 12bit core. I've included a .inc file with all declarations and settings needed to use a PIC16C5x assembler for the PIC12C508.

## 1.12 the calibration location

The Calibration location

As you know the PIC12C508 has a calibration value programmed at location 0x1ff.

The programmer will leave this location untouched, even when you have used this location in your program.

However, if the programmer can't find a valid movlw 0xXX instruction (erased by UV light in the windowed version by example), it will program a movlw 0x00 instruction at location 0x1ff.

## 1.13 cardware

Disclaimer

This package is CARDWARE which means you may use and distribute it freely, as long as you don't change the archive.

If you like it and use it, I would like it very much to receive a postcard from you (maybe a nice tourist card ?).

MICROCHIP and PIC12C508 are registered trademarks of Microchip Technologies Inc.

## 1.14 history

History

27 May 1998

V1.0

-First release

28 October 1998

V1.1

-Serial Speed improved to 2400 baud

-Programming algorithm updated to new specs as described in doc DS30557D from [www.microchip.com](http://www.microchip.com)  
This wil fix all problems with the infamous "9820" series of MCU's although it seemed that my programmer never had any problems with this series, Some commercial programmers did .....

- Error fixed in main schematic
- several Errors fixed in readme

11 Januari 1999 V1.2

- improved HEX-loader, the loader can now read any dialect of the infamous INHX8M format.  
I also added a special version of the downloader which ignores checksums, as some hexfiles work OK but mess around with checksums.
- made some timing changes in the firmware so update the firmware too!

## 1.15 todo list

Future or todo list

- Add a selftest function to simplify adjustment and testing
- Create a MUCH simpler version of the programmer as I got very little response to this programmer, presumably due to the complexity of the hardware  
I am thinking of a version with only a pic16c84,max232 and some transistors

## 1.16 author and bug reports

Author and Bug reports

I am a 21 year old electrical engineering student from the Netherlands. I was not very good in english class so this document has probably lot's of errors (well at least less than the last one).

my Adress (for the postcard of course) is:

Dennis van Weeren  
Reina Prinsen Geerlingsdreef 59  
2135 HS Hoofddorp  
The Netherlands

My email adress is: d.vanweeren@worldonline.nl  
If you have bug reports or other troubles or suggestions, feel free to email me.

Happy Programming with the smallest MCU in the world!

---