

# KIT 81. 16F84 PIC PROGRAMMER V4

The 16F84 micro-controller uC from Microchip technologies has become the point of entry into the field for many beginners & hobbyists. This is because the cost of the tools needed to support the device are very low and the device itself is easier to use than other uC's on the market. The main feature is that it does not require a UV light to erase the program inside it. The stored program is electrically erasable in seconds. It can be used over & over.

This development programmer - hardware & software - is designed only for the 16F84. (It will also work with the discontinued 16C84 but we will just refer to the 16F84 in this text.) This eliminates many sources of confusion for the first time user.

All the software & hardware to build & run your first 16F84 program as well as a PIC16F84 are contained in this kit. A program and components to flash an LED comes with the Kit.

Earlier versions of this kit were based on an article published in *Electronics Australia* in 11/95 & 4/96. These articles are reprinted here. The current version of the kit (V4) supplied here has a simpler power supply section. These articles should be read for background information.

## REFERENCES

Before you use the kit review the references below. The Internet is now the place to get References, data sheets and technical support.

Microchip's web page is at

**[www.microchip.com](http://www.microchip.com)**

Download the 16F84 data sheet from here. Or you can download it from the Microchip ftp site at

**<ftp://ftp.futureone.com/pub/microchip/download>**

30430c.pdf is the data sheet index number for the 16F84.

30430c.pdf is the data sheet for the new 16F84A released november, 1999.

Websites come and go. For a list of working and up-to-date websites go to my Links page at

**<http://kitsrus.com/back.html#pic>**

**<http://dontronics.com>** Don McKenzie has many PIC links and up to date PIC information here.

The following books are well worth getting since they start at the very basics. There are four books in the series so far.

**Easy PIC'n, PIC'n Up The Pace, PICn Techniques** and the latest book **Serial PIC'n**. They published by Square One Electronics. See **<http://www.sq-1.com>**

See also: **Programming and Customizing the PIC Microcontroller** by Mike Predko. His website is

**[www.myke.com](http://www.myke.com)**

Only after you have read some documentation and references should you start construction.

## FEATURES OF THE 16F84 & the KIT

- device is EEPROM. You can use the 16F84 over & over without the need to use a UV eraser
- software is only for the PIC16F84 so no other programming options to confuse you
- software can verify, erase & read as well as load. Program & EEPROM spaces can be treated separately
- configuration fuses can be read, loaded & verified separately
- programmer test option in software after assembly
- introductory manual supplied on disk
- powered by unregulated 17-30VDC, or 13-20VAC
- test program *lights* provided to immediately test your programmer. Components are supplied in this kit.

## ASSEMBLY

First check the components supplied in the Kit against the Component listing printed below. Identify all the components. It is generally best to solder the lowest height components first: the resistors, & IC sockets. **NOTE:** there is one link to add to the board. Use a leg cut-off from a resistor. One of the 0.1uF monobloc capacitors C2, fits within an IC socket.

Note also that C6 is **now** supplied with this Kit. It has been reported that is a small number of cases a 330pF capacitor is needed in this position. Recently we have seen several cases so the capacitor is now supplied.

Check the orientation and name each IC before you put it into its socket. Match the dimple on the overlay with the dimple on the IC. Power the kit with 17-30VDC, or 13-20VAC. A bridge rectifier is used so it does not matter which way around the power is applied to the power jack. Both SIL resistor networks have a dot to indicate the common pin.

## CIRCUIT DESCRIPTION

This is described in the *Electronics Australia* article. But note that a simpler power supply. For a more detailed specification about the programming sequence see AN589 in the Microchip Embedded Controller Handbook.

**Schematic.** The schematic for the programmer is given below in Figures 1 & 2. The circuit is also described in manual.doc on the floppy disk. It is in Word 6 format. You should print this out in full.

## SOFTWARE

# KIT 81. 16F84 PIC PROGRAMMER V4

Over the last three years both the hardware and the software for this kit have evolved. PC's have also got very fast and more complicated. We have now combined software from two sources in this kit to give 100% assembly and programming results.

**To run all this software close Windows and restart in MS-DOS mode.**

The original software supplied with this kit has not been given ongoing full support. So there are problems reported with pf84 when used for programming. So this is what we have done.

1. Use **pf84 /t** to test your programmer after assembly. There must be no 16F84 in the socket. You may need to add **/p1** or **/p2** to force printer port recognition. That is, enter **pf84 /p1 /t** Cycle through the sixteen tests by pressing the spacebar. If everything is OK then 'Automatic input tested passed' will come up. If you do not get this message something is wrong in the assembly. Check the supply voltage. Check all components. Is the 74LS07 IC around the correct way.

2. Use **af84 /HEX <program names>** to assemble your code. Make sure to always use the **/HEX** option. Make sure the include file, 16f84reg.inc. is in the same folder.

3. Use P16Pro Lights to program the 16F84.

P16Pro Lite is a special freeware version of the P16Pro software used in our Kits 96 and 119. It may be downloaded from

**[picallw.com](http://picallw.com)**

The author of the software, Bojan Dobaj, actively supports his software. He is always available by email. The read.me file on the floppy disk gives more description of the software which comes with this Kit.

**First Program.** We have included a test program with this kit so you can prove that you have got your programming method correct. We have also supplied all the components necessary to built the first program on a breadboard. They are in a separate packet. Connect power to the programmer and leave it connected. The LED indicates when programming, verification and erasing are taking place. It is best not to remove the PIC when the LED is on.

This program, *lights.pic* flashes 5 LED's in a binary counting pattern. The circuit to build this is on the bottom of the V4 PCB supplied with this Kit. Components for this are supplied.

Assemble the program *lights.pic* by entering  
**af84 /HEX lights.pic lights.hex**

If you forget the options just enter af84 by itself and they will appear. Note that the suffixes can be anything but use

the hex suffix for the output file so you know what it is. The output whatever its name is a hex file which you can read using any text editor or wordprocessing program. Make sure the include file called on by the assembler is in the same folder.

Now to program the 16F84 enter  
**pf84 /e**  
to erase the PIC, then  
**pf84 /hex /e /p1 /la lights.hex**

(You can see what all these options mean by entering pf84 alone.) If it programs, then good, you have done it. But some PC's now are giving an Error at this stage. Not to worry. Use P16Pro Lite.

**P16Pro Lite.** Extract p16pro11.zip in a folder. Then rename two files to match the hardware of Kit 81 to Bojans program.

Rename p16pro11.ini to p16pro11.old.  
Rename kit81.ini to p16pro11.ini

Run p16pro11.exe

Do Alt-S Device and enter 1 for the 16F84. You can do the same thing by hitting F3.

Hit F1 to enter the program. It must have a .hex suffix to be automatically detected.

Do a Blank check by hitting F7. Do an F9, Erase, if the chip is not blank.

Now just check the Fuse Settings by hitting F2. They should read oscillator RC, Watchdog timer enabled (that is how this program lights works), and the Code Protect should be not enabled.

Now hit F4 to program. Programming should be successful. (If it is not I want to know.) Remove the PIC and place it in a breadboard setup according to Fig. 3. The LEDs should count.

## WHAT TO DO IF IT DOES NOT WORK

Since the hardware has already passed its test the programming should be successful if you followed the above steps. If it is not then please email me. I have thrashed this program for some hours on different computers trying to make it fail.

**PIC Newsgroup.** There is an active PIC newsgroup. To subscribe email **listserv@mitvma.mit.edu** with the **SUB PICLIST** in the body of the message. You will receive an email back within minutes. I suggest you then send another message **SET PICLIST DIGEST** to the same address. Then you will start to receive a single message each day containing all the messages sent to the list for the previous day. Read the messages for a week or so to get the feel of

# KIT 81. 16F84 PIC PROGRAMMER V4

the newsgroup before asking questions yourself. Note that the List is rather Conservative so stick to PIC topics.

There is a mine of PIC information at the piclist website. Personally, I find it a confusing site to navigate. There seem to be two entry points. Try the following:

<http://www.piclist.com>  
<http://www.piclist.com/techref/index.htm>

**Software Difference Between C & F Versions.** The 'C' versions are rare now but if you have one the simple answer is that there is no difference. When you have done some programming for a while you will find there is one difference which is in the configuration word. It is in the **Power Up Timer Enable Bit**. This bit has reversed its meaning. In the F you must write a 0 to enable the timer while for the C you had to write a 1. You will see this in the program examples. In *lights.pct* the config word changes from 0f7h in the C version to 0f7h for the F version. So long as you set it to 0 in the config word it will be enabled. Remember bits left unprogrammed in the config word are read as 1.

There are two other hardware changes in the code protection bits (8 to 13) and the RAM file registers (increased from 36 bytes to 68 bytes.) Code protection will not normally be used by beginners.

## COMPONENTS

Resistors 1/W 5%:		
1K brown black red	R1	1
3K3 orange orange red	R3 R6	2
10K brown black orange	R2 R4 R5 R7	4
10K 4pin, 3 resistor network		1
DB25 25 pin right-angle, pcb-mounted connector		1
470uF/35V ecap	C3	1
330pF	C6	1
BC557	Q1 Q2	2
0.1uF 104 monoblok	C1 C2 C4	3
3mm led	LED	1
78L08 voltage regulator	IC2	1
78L05 voltage regulator	IC1	1
PIC16F84-4/P		1
74LS07 or 7407	IC3	1
14 pin IC socket		1
18 pin IC socket		2
DC power jack		1
WO2M bridge rectifier		1
Kit 81 V4 pcb		1
Floppy disk with software		1
lights.hex testing components:		
3mm led		5
22pF ceramic capacitor		1
3K9 resistor		1
SIL 6pin 5resistor network, 1K 102		1

**16F84 and 16F84A.** In november, 1999 a faster (20MHz) version of the 16F84 was released, the 16F84A. The data eeprom write time is about 4mS compared to 10mS for the 16F84. There were no other changes.

This programmer design is a so-called 'David Tait' based design. It will work with a wide range of freely available

software. All that will need changing is the configuration file and the hardware file depending on whether you use a 7406 or 7407. See **dontronics.com** for the Nigel Goodwin & the P16PRO software and further references. You can design cheaper PIC16F84 programmers than this but they may not be as robust in the wide range of PC's and parallel ports now existing.

The floppy disk which comes with this kit, as well as the documentation, may be downloaded from the software download page on my website:

<http://kitsrus.com/soft.html>

If you want to move on and program other PICs get my Kit 96 or Kit 117. The software is also by Bojan Dobaj so you will feel at home with it.

See details at **dontronics.com** and **kitsrus.com**

**History.** Assembler & programmer software just for the 16F84 added 3/99. New P16PRO Lite software added 5/99. V3 PCB issued 11/99 and documentation rewritten focussing on the F84 (dropping the C84) & P16Pro Lite software (moving away from the unsupported p84 and pf84 programming software.) V4 PCB with the on-board test circuit shipping january, 2000.

4/2001. Corrected error in PCB track for the lights testing circuit. K81V4.2 PCB.

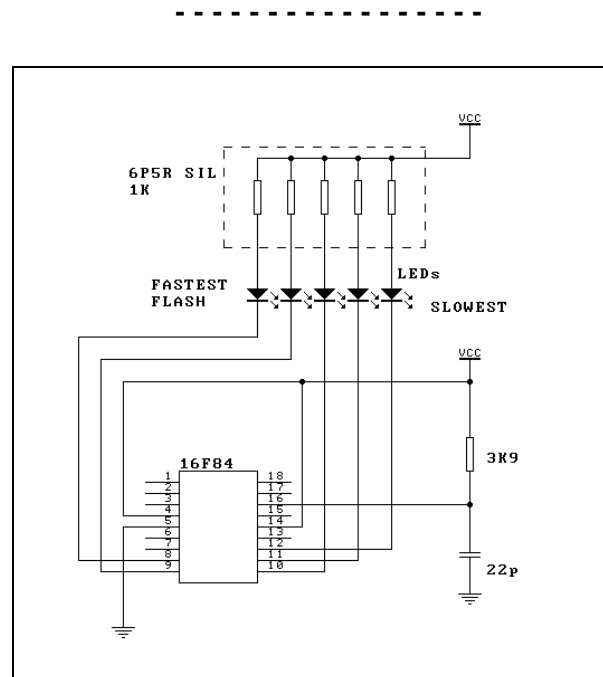


Figure 1. Lights.hex Test Circuit



## Using PICALLW.EXE to test Kit 81

In 2002, the Windows version of PICALL is the programming program to use with my Kits 117, 96 81 and 119. This program **picallw.exe** is where the developer Bojan Dobaj is putting all his effort.

Forget P16Pro, pf84 and 'picall for dos'. Use the latest version of Picall for programming all the above programmers. (It will also program Dontronics DT-001.) Instead of changing the documentation for all these kits I have initially done this general information sheet. Download version 0.10c, november 2001 of picallw.exe from

**kitsrus.com**  
**dontronics.com**

and install it.

Picallw.exe will work **free** for PIC16F84 programming in all my PIC programmers. No registration is required. If you have a paid registration for P16PRO then move the p16pro.reg file to the same folder as picallw.exe.

**Kit 81.** Just have the SN74LS07 in its socket. Do not put in the 16F84 (but it does not matter if it is there.) Connect a **straight-through** Male/Female cable from the PC parallel port to kit 81. (**Not** a modem cable; **not** a lap link cable – a **straight-through** cable.) If the cable is not correct you will get an error message. Connect power. Start picallw.exe. In the left drop-down box make sure **P16PRO** is selected. **MICROCHIP** should be in the center box and **PIC16F84** or **PIC16F84A** in the right hand box. (Look on the PIC chip supplied with the kit.)

- Settings/LPT Port. Set to 'auto' is what I use.
- Settings/Hardware Setup-Test. Click on **Kit 81**. In the right hand column are 6 Set/Clear boxes. Click each box in turn. The number should go to 1 then back to 0 with the next click. When you click the VDD box the LED on the Kit 81 PCB should turn on. Clicking again will turn it off.

That is it. Click **OK** and do your programming. If you get any Error Messages you can find the details at:

Help/Contents/Hardware/Picall Hardware/PICALL Error Codes

**Programming.** I assume you have produced **lights.hex** and put it in the same folder as picallw.exe.

Do File/Open and click on lights.hex. Click OK.

In the column on the right hand side do not uncheck the top two boxes.

Put the 16F84 into the programming socket. See if it is Blank. If not then Erase it.

Click F4 to program. Do a verify. Sometimes the Buffer may indicate an error. If so do not worry. Move the PIC to the Test IC socket and put it in. The running lights program should start immediately. The 5 LEDs will start to flash in a fast binary counting pattern.

The programmer works.

-----