

Tiny PIC bootloader

[/HOME](#)

Tiny Bootloader

This is a bootloader for the Microchip PIC microcontrollers.

- It is the smallest bootloader, taking less than 100 words of program space;
- Supports families of PIC devices: 16F, 18F, dsPIC30 (those devices that support self-programming)
- Does not support yet PIC24, dsPIC33, PIC18xxJyy

[Download Tiny PIC Bootloader](#)

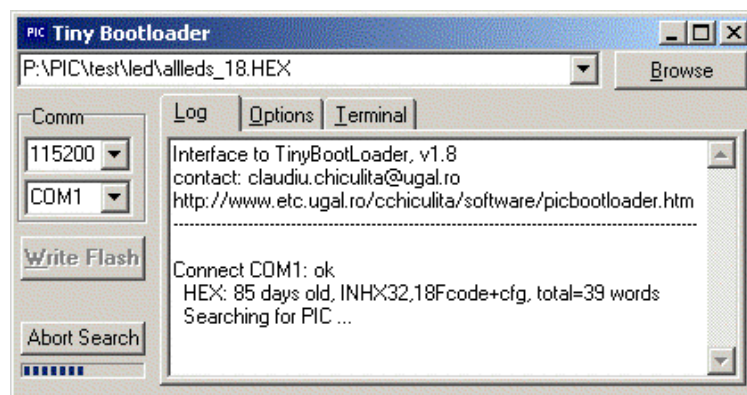
Features of the firmware

- Size of only 100 words; (all versions, for 16F,18F,dsPIC occupy less than 100 words);
- Can write flash, eeprom and configuration bytes(18F);
- On reset, waits 1 second (adjustable) for a message from the PC, if not received, launch user application;
- The .asm file can be easily modified and adapted for any frequency (or baudrate);

Features of the PC software

- Can upload programs into flash (in current version eeprom and cfg bytes can be modified only manually, depending on the device type);
- Works with PIC16F,PIC18F and dsPIC types; automatically detects HEX content and PIC model;
- Remembers last settings;
- In case of errors, performs retransmissions or tries to resynchronize with pic;
- The communication settings are editable so you can write any COM number or desired baud;
- If a filename is specified as a command line parameter, it will automatically try to write it;

How to use TinyBld



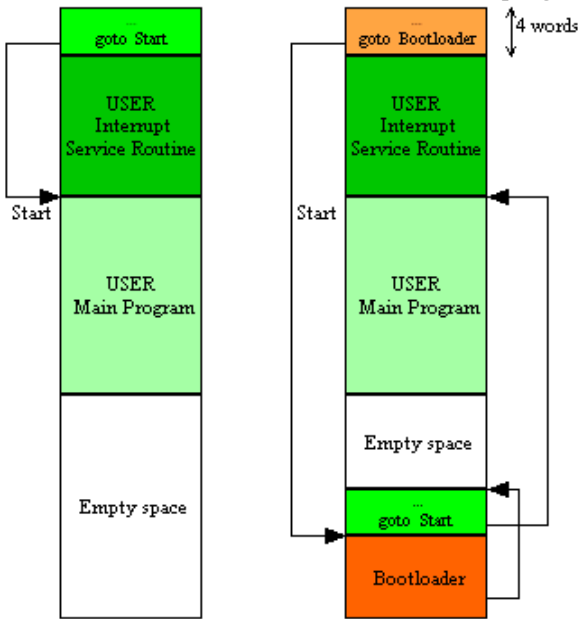
Supported/Tested Devices

Device	Devices of the same type	Flash	ram	EE	ser	adc	osc	pin	
PIC 16F Family									
PIC16F876	877, 873, 874								
PIC16F876A	877A , 873A , 874A	8k	.3	.2	1	y		dD	
PIC16F887	PIC16F886								
PIC16F88	87	4k	.3	.2	1	y	y	d	
PIC 18F Family									
PIC18F252	452, 242, 442; 2420 , 2520 , 4420 , 4520	16k	1.5k	.2	1a	y		dD	
PIC18F258	458, 248, 448; 2480 , 2580 , 4480 , 4580	16k	1.5k	.2	1a	y		dD	CAN

PIC18F2620	4620, 2525, 4525	32k	3.9k	1k	1e	y	y	dD	
PIC18F1320	1220, 2220, 2320	4k	.2	.2	1e	y	y	d	
PIC18F8720	6520, 8520, 6620, 8620, 6720; 6621	64k	3.8k	1k	2	y		Q	ext.mem.
PIC18F2550	4550, 2455, 4455	16k	1.5k	.2	1e	y	Y	dD	usb
PIC18F4431	2331, 2431, 4331	8k	.7	.2	1e	Y		dD	advPWM
PIC18F4680	2585,2680,4585	32k	3.3	1k	1	y		dD	ECAN
PIC18F4580	2480,2580,4480	16k	1.5	.2	1	y		dD	ECAN
PIC18F4620	2525,2620,4525	32k	3.9	1	1	y		dD	
PIC18F4320	2220,2320,4220	4k	.5	.2	1	y		dD	
dsPIC Family									
Device	Devices of the same type	Flash	ram	EE	ser	adc	osc	pin	
dsPIC20F2010		4k	.5	1k	1	y	y	d	6pwm
dsPIC30F6014	6012, 6013, 6011, 6012A	44k	8k	4k	2	yy	y	Q	2CAN
dsPIC30F3013	3012, 2012, 2011	8k	2k	1k	(2)	yy	YY	d	
dsPIC30F4012	4011	16k	2k	1k	(2)	y	Y	dD	6pwm, CAN
dsPIC30F3011	3010	8k	1k	1k	(2)	y	Y	dD	6pwm
dsPIC30F6010		44k	8k	4k	2	y	y	Q	8pwm, 2CAN
blue - tested for longer periods blue - verified (or user source provided) navy - reported to work (may require modifications!)		Flash, RAM, EEprom: max. amount ser: a=addr.; e=enh.(brg/lin) adc: yy=12bit osc: internal+features pin: d<=dip28; D=dip40; Q>=tqfp							

General info about Bootloaders

A bootloader is a program that stays in the microcontroller and communicates with the PC (usually through the serial interface). The bootloader receives a user program from the PC and writes it in the flash memory, then launches this program in execution. Bootloaders can only be used with those microcontrollers that can write their flash memory through software. The bootloader itself must be written into the flash memory with an external programmer. In order for the bootloader to be launched after each reset, a "goto bootloader" instruction must exist somewhere in the first 4 instructions; There are two types of bootloaders, some that require that the user reallocate his code and others that by themselves reallocate the first 4 instructions of the user program to another location and execute them when the bootloader exits.



Without bootloader user code is in colored in green With bootloader bootloadercode is colored in red

(Some of the) Available bootloaders (as reported by Google) in May, 2003:

Bootloader Name /			
-------------------	--	--	--

Author	Supported models	Size(words)	Comments
From Microchip	16F,18F	1000	uses Hyperterminal to upload hex files
From MicrochipC	16F,16F*A,18F	256/2000	+ok
WLoader Wouter van Ooijen	16f877	1000	+does not use the UART, +the serial interface use only one I/O pin
ZPL Wouter van Ooijen	18F	384	+unusual method using mclr: uses zero I/O pins !
KarlLunt	16f87x	512	-activation on input pin -derived from Microchip boot877.asm (uses Hyperterminal)
PICLOADER Rick Farmer	PIC16F87x	2000	-program must start at 0x3; +password (uses Hyperterminal)
bootload	PIC16F877	800	-written in C -command line DOS program
theByteFactory	16F877	1000	-written in C (uses Hyperterminal)
Jolt Martin Dubuc	18F	256	-user code and interrupt vectors need to be relocated; Java GUI, +auto detect baud
2 HI-TECH Software	16F87x	256	-written in C
PIC downloader Petr Kolomaznik	16F876	256	-is rewritten and modified from HI-TECH +Windows interface
Ivar Johnsrud	18Fxx2/18Fxx8	360	-bootloader based on HiTech's -downloader based on Petr Kolomaznik's
B Bootloader	PIC16F87x, PIC16F87xA	340	-called only by user application -written in C +Linux uploader
SGupta	16f876	256	
I put mine here, for comparison:			
Tiny	16F, 18F, dsPIC	100	+details above

This project was born because:

- Some bootloaders I used previously had some unpleasant bugs or didn't support the devices I had.
- I wanted to do it small.
- If you wonder how much smaller than this can it get, the answer is 32words :), but it loses from features, reliability and speed.
- TinyBootloader was build with the idea that all the hard work could be moved to the PC application, while the firmware can implement only the basic functions.

Critics and suggestions at: **Claudiu.Chiculita@ugal.ro**

(Due to mail server spam filters (beyond my control) stopping many good emails, and deleting them after a short period, it is possible to miss some of the messages)