

Bootloader Details

Overview:

The bootloader (**stub***) is a permanent program in the PIC, **hexldr** is a program loads an image. The bootloader is programmed into the PIC the normal way. **hexldr** can then reprogram the PIC with a program image. Only one pin and the reset line are used.

The bootloader targets the PIC16F15344 and PIC16F15345 devices. It should be possible to retarget to other PIC devices which have a serial port and pin mapping.

The bootloader size is less than 480 words and it starts at ("top of memory" – 480). For the '344 that is hex 0E20; for the '345 that is hex 1E20. The bootloader ends at 0FC7 and 1FC7 respectively.

Programming Considerations:

The bootloader does not use interrupts so the application is free to use them without restrictions.

The bootloader reserves the first 2 locations in program space:

```
org 0 ;
pagesel btldr
goto btldr; bootloader should not overwrite this.
```

The user's program should start with the following:

```
org 0 ;
pagesel init ; user program entry point
goto init
```

The hexldr program will copy these 2 words to upper memory so on normal entry the program will start. So 2 words are reserved ('344: 0E1E, 0E1F) ('345: 1E1E,1E1F) for the "init vector" (aka appvec).

NOTE: if the 32 word block just before the bootloader is written these 2 words are reserved for the application vector as above by the hexldr program controlling the bootloader. The bootloader itself is protected by the hexldr program.

RE: Scatterload - if the hex file does not include block zero with the init vector the hexldr program will assume it is a scatterload and it will load it into program memory.

Configuration Settings:

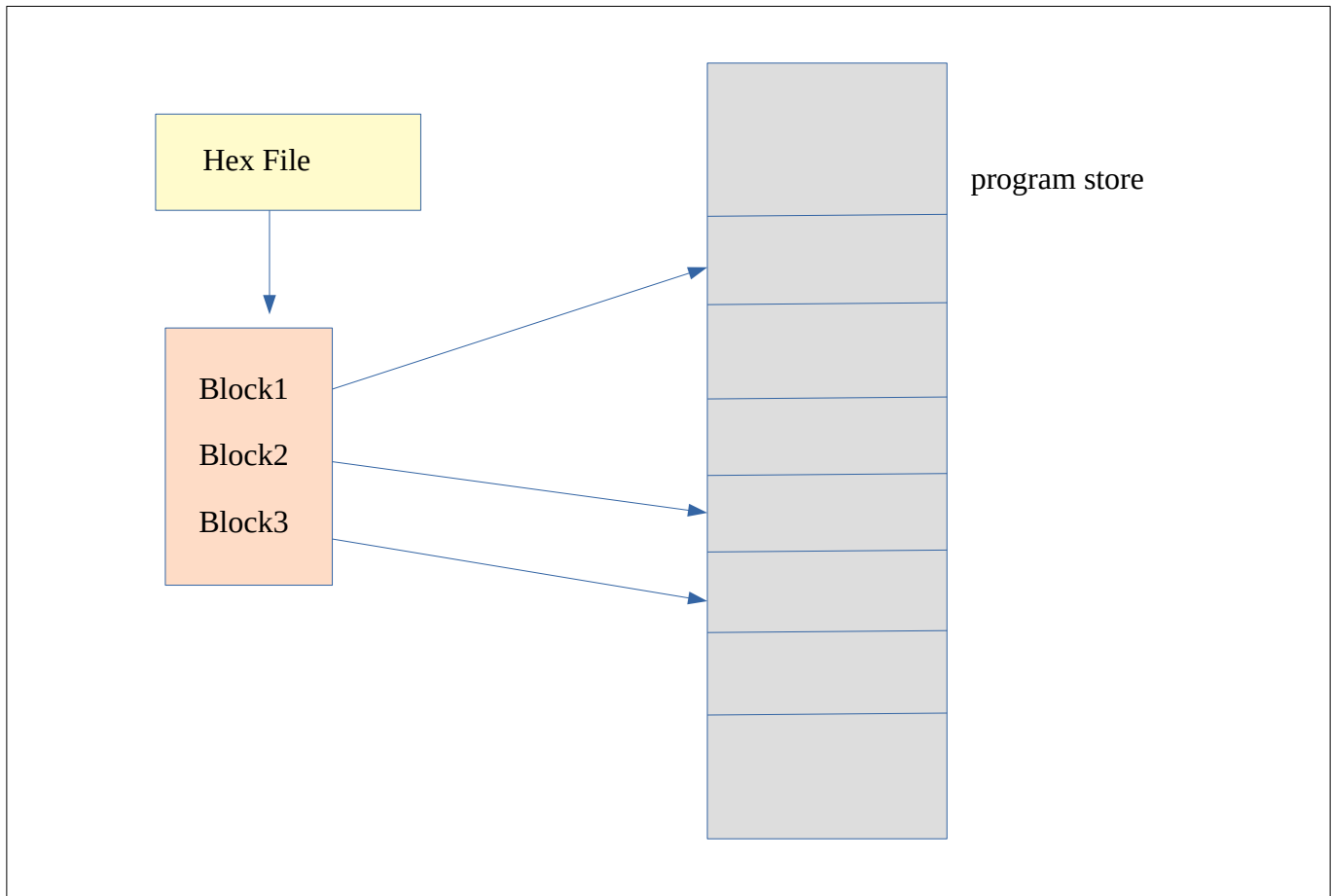
The configuration settings for the PIC are permanently set in the bootloader code. The clock is set to 32 M, MCLR is enabled and write to the whole memory is enabled. The settings can be changed in the bootloader code.

Config Code:

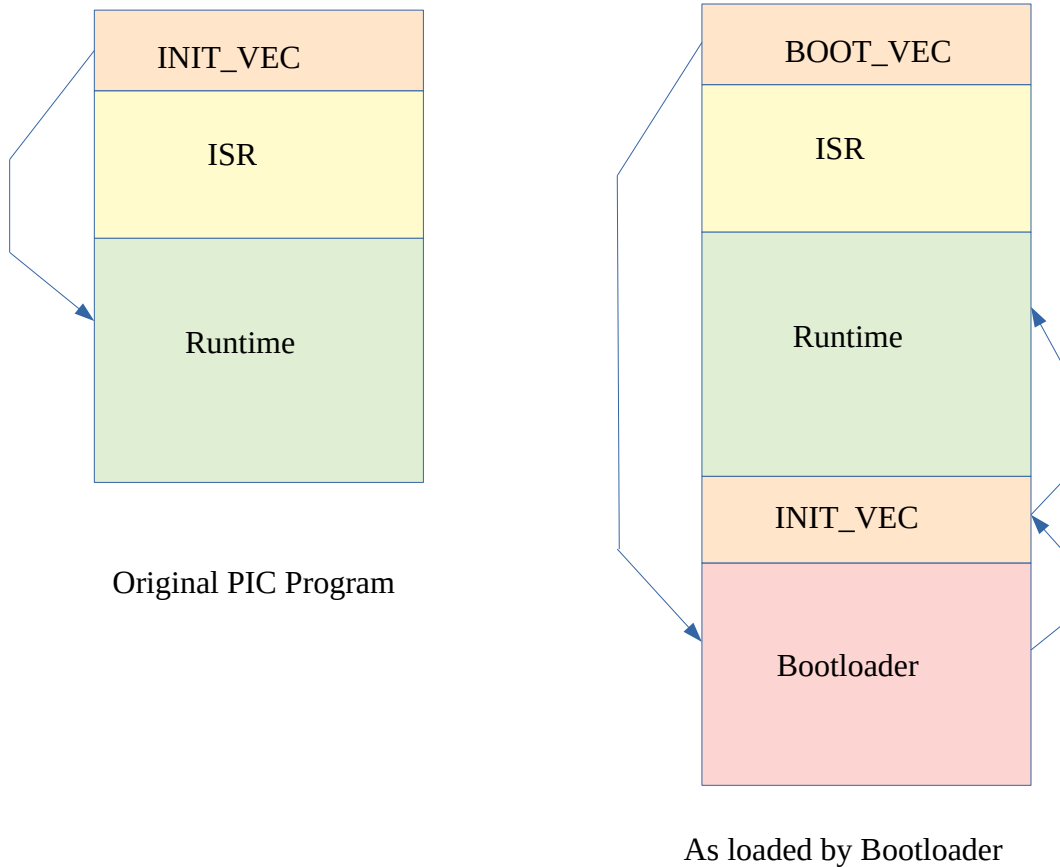
```
__CONFIG _CONFIG1, _RSTOSC_HFINT32 & _CLKOUTEN_OFF & _CSWEN_OFF & _FCMEN_OFF &
_FEXTOSC_OFF
__CONFIG _CONFIG2, _MCLRE_ON & _PWRTE_OFF & _LPBOREN_ON & _BOREN_ON & _BORV_LO &
_ZCD_OFF & _PPS1WAY_OFF & _STVREN_ON
__CONFIG _CONFIG3, _WDTE_OFF
__CONFIG _CONFIG4, _BBEN_OFF & _SAFEN_OFF & _WRTAPP_OFF & _WRTB_OFF & _WRTC_ON &
_WRTSAF_OFF & _LVP_ON
__CONFIG _CONFIG5, _CP_OFF
```

How hexldr uses the Bootloader:

To use the bootloader, first 32 words are written to internal RAM, then that block of memory is written to program store as a block.



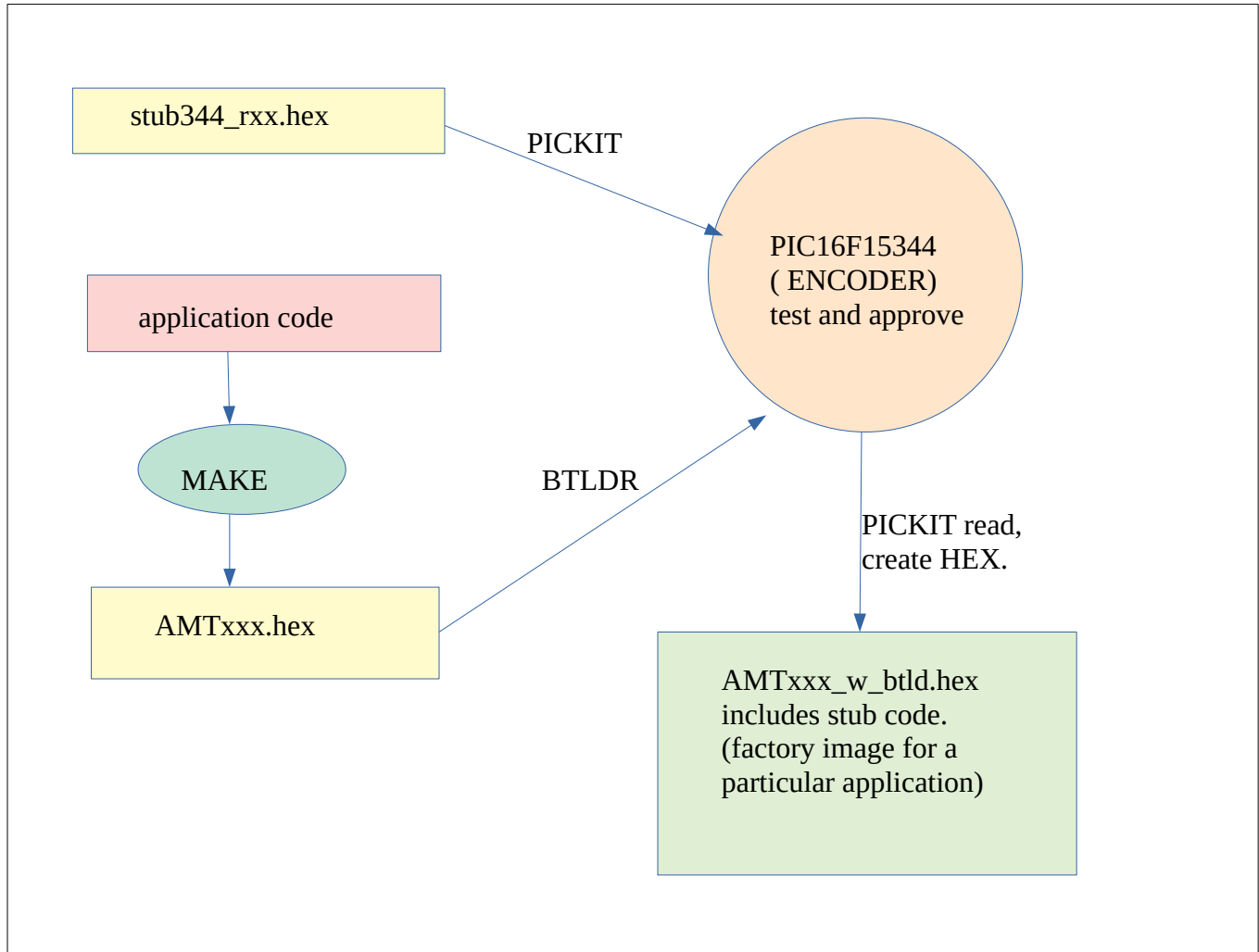
Program image in core:



The hexldr assumes the init_vec is at location 0, it has to relocate it.

It is assumed there is no bootloader in the user image since the location of the bootloader cannot be written. If there is a bootloader in the image it should be removed beforehand and location 0 should be set properly. Otherwise the loaded code will not execute and the PIC will stay in the bootloader.

Creating the image for a particular encoder:



Note: the image is read out of the PIC and used to pre-program factory parts, it must be read out with a hardware programmer and the configuration bits are part of that image.

Commands:

These are all suffix <lf>. *=priviledged instruction (unlock)

Command	Action	Notes
jppmmnn	Read Ram page pp location mm+ for nn bytes	debug command
K<string>	Reply <string>	test for 2mbit link
Q	reset	does SW reset in PIC
Rhhll	read a line of memory 32 words	* reads into RAM
mnnxxx..	write 16 bytes to ram location nn hex	* 16 hex bytes xx
uhhll	program a row of memory hhhll	* hh nvmdrh ll nvmdrl
bhhll	blank a row of memory hhhll	* hh nvmdrh ll nvmdrl
i	read devcode	0x13 for bootloader

Ram is in bank 2 and the m command is used to write to locations 20,30,40,50. 32 words of memory represent one row of program memory.