# CHIRP - Bug # 1557

| | | | |
|---|---|---|---|
| **Status:** | Closed | **Priority:** | Normal |
| **Author:** | Dan Drogichen | **Category:** | |
| **Created:** | 04/11/2014 | **Assignee:** | Dan Drogichen |
| **Updated:** | 01/24/2017 | **Due date:** | |
| **Chirp Version:** | 0.4.0 | | |
| **Model affected:** | FT-60 | | |
| **Platform:** | All | | |
| **Subject:** | [FT-60] The implementation of memory skip is incorrect. | | |

**Description**

The current code related to memory skipping is quite broken. Both memory locations and Skip/Priority sense are scrambled.
There are two issues causing the visible bugs, and a third that is exposed with the developer tools if you go looking for it.

Each memory location's scan "skip" property can be set to one of "Skip" (S), "Only" (priority scan, P), or "Off" (default). In the radio's button interface, this is done with Set Mode item 46. See the FT-60 User's Manual pages 37 and 38.

This sequence demonstrates the problems with the current code. Initially, skips in all memory locations are off. Using the radio's "set" interface set memory 16 to "Skip". Download into chirp. Chirp thinks memory 12 is P (Only) !!

The start of flags[500]:

28360: 00 00 00 40 00 00 00 00

Set memory 1 to "Only" = Priority = P. Download into chirp. Chirp thinks memory 3 is "S".

28360: 02 00 00 40 00 00 00 00

Flip them: Set memory 1 to Skip, memory 16 to P. Download. Chirp thinks memory 3 is P and mem 12 is S. Same wrong memories, and consistently reversed sense.

28360: 01 00 00 80 00 00 00 00

Analysis of some other examples and the code have determined:

A) The radio doesn't include S/P flags for the Chirp array element memory[0] of array memory[1000]. There are actually only normal memories 1-999. In order to have the array offset be the same as the memory number, Chirp defined the array as starting 16 bytes before it really does in the radio's eeprom, but then functionally ignores fictitious memory[0]. This will need to be changed to fill out the memory map, but it doesn't need to be changed to fix this bug.

There is actually a memory 000, but it lives at the end, where memory[1000] would be if the array were defined as memory[1001].

B) It looks like a complete bit reversal, both with respect to memory order and S/P flags, within each byte.

For byte 28360, flags[0], which has S/P flags for Mem1 - Mem4, the
correct interpretation of the bits is:

(msb) Mem4P  Mem4S  Mem3P  Mem3S  Mem2P  Mem2S  Mem1P Mem1S (lsb)

while Chirp is currently interpreting them as

(msb) Mem0S  Mem0P  Mem1S  Mem1P  Mem2S  Mem2P  Mem3S Mem3P (lsb)

One more example to make the pattern clearer:
For byte 28368, flags[1], which has S/P flags for Mem5 - Mem8, the correct interpretation of the bits is:
(msb) Mem8P  Mem8S  Mem7P  Mem7S  Mem6P  Mem6S  Mem5P Mem5S (lsb)

while Chirp is currently interpreting them as
(msb) Mem4S  Mem4P  Mem5S  Mem5P  Mem6S  Mem6P  Mem7S Mem7P (lsb)

C) It's not causing any functional bugs I've noticed yet, but there's something else wrong with this code: If there are 999 (or 1000) memories, stored in memory[1000], and two flag bits packed into bytes at four memories per byte, then flags should be flags[250], not flags[500].

u8 flags[] starts at 0x6EC8. 500 bytes would make the next byte 0x70BC. 250 would make the next byte 0x6FC2. The checksum is at 0x6FC8. Makes the case for 250 being correct pretty strong.
This bug can be exposed when using the developer tools in the current code:

1. Use a current daily build, I'd been using 20131204.
2. Open up any random FT-60 image file.
3. Open the Browser tab.
4. Expand root, then flags.
5. Scroll down and select flags[500][255]. Looks normal. So would 0-254.
6. Select flags[500][256], again normal.
7. Select flags[500][257]. Whoops, we're not in Kansas anymore.
8. You will then find an exception and stack trace in the debug log.

The fix involves:

1. Changing flags[500] to skipflags[250]. The memory offset is unchanged. The name change is in anticipation of bankflags[] being added, and to avoid confusion with an unrelated variable in the file named flags.
2. Indexing skipflags[] with [memory - 1] rather than [memory].
3. Reversing the order of both the elements of skipflags[] and the non-zero elements of SKIPS[].

Testing the change shows that skips defined in the radio interface are now correctly interpreted by chirp, and editing skips in chirp and uploading results in the expected interpretation by the radio.

**Related issues:**

| | | |
|---|---|---|
| related to Bug # 1419: FT-60R loss of skip feature during upload and download | **In Progress** | **01/31/2014** |

### Associated revisions

**Revision 2216:de8ad7a11b6a - 04/12/2014 12:04 pm - Dan Drogichen**

[FT-60] The implementation of memory skip is incorrect. - Fixes #1557

- Change flags[500] to skipflags[250]. The memory offset is unchanged.
The name change is in anticipation of bankflags[] being added, and to
avoid confusion with an unrelated variable in the file named flags.

- Index skipflags[] with [memory - 1] rather than [memory].

- Reverse the order of both the elements of skipflags[] and the
non-zero elements of SKIPS[].

### History

**#1 - 04/11/2014 05:28 pm - Dan Drogichen**

Blech. This bugtrack program has formatted what was intended as array sizes or offsets in square brackets as a superscript. Read the above with that in mind, and it might make more sense. What value does that format munging have in this context?

**#2 - 04/11/2014 06:36 pm - Tom Hayward**

*- Description updated*

**#3 - 10/02/2014 09:09 pm - Patrick Lang**

Why is this still marked 90%?  I just tried the latest version out and skip flags are working as expected.  If any more testing is needed, please let me know how I can help.

**#4 - 10/03/2014 10:11 am - Tom Hayward**

*- Status changed from New to Resolved*
*- Target version set to 0.4.1*
*- % Done changed from 90 to 100*

Thanks for confirming. I think it was just overlooked.

Redmine is silly so I have to do this in two steps...

**#5 - 10/03/2014 10:11 am - Tom Hayward**

*- Status changed from Resolved to Needs Backport*

**#6 - 01/24/2017 11:21 am - Tom Hayward**

*- Status changed from Needs Backport to Closed*