

CHIRP - Feature # 495

| | | | |
|---|--------------------|-------------------|--------|
| Status: | Closed | Priority: | Normal |
| Author: | david billingsley | Category: | |
| Created: | 02/03/2013 | Assignee: | |
| Updated: | 01/25/2022 | Due date: | |
| Chirp Version: | 0.4.0 | | |
| Model affected: | (All models) | | |
| Subject: | PYTHON | | |
| Description | | | |
| Any plans to upgrade to Python 3.3? | | | |
| Software is great, local ham club has nothing but good things to say about Chirp. | | | |
| thanks, David | | | |
| Related issues: | | | |
| related to Feature # 7431: Python3-friendly UI based on wxPython | In Progress | 12/09/2019 | |
| related to Bug # 6327: PyGTK Deprecated in Debian 10 | New | 12/27/2018 | |
| duplicated by Bug # 7859: Python2 EOL prevents installation/running Chirp | Closed | 05/07/2020 | |
| duplicated by Feature # 5461: Python 3 support | Rejected | 12/27/2017 | |
| duplicated by Bug # 7877: Python3 syntax errors | Rejected | 05/14/2020 | |
| duplicated by Bug # 7923: python3 support | Rejected | 05/27/2020 | |
| duplicated by Bug # 7989: Support for Python 3 | Closed | 06/16/2020 | |

Associated revisions

Revision 3027:55db4195cf98 - 02/07/2019 04:12 pm - Dan Smith

[py3] Changes to bitwise and friends for py3 compatibility

Much of this came from 2to3, but all the unicode/str/bytes changes were hand-done. Unit tests pass for bitwise.

#495

Revision 3028:46631cc4ac55 - 02/07/2019 04:38 pm - Dan Smith

[py3] Make setup.py compatible and start a new py3 tox environment

Right now, this runs just the bitwise tests which are the only ones that work on py3.

#495

Revision 3029:e32123791a6c - 02/07/2019 05:12 pm - Dan Smith

[py3] Changes to make test_chirp_common (and deps) work on py3

This is enough to be able to use pytest too, so convert to that.

#495

Revision 3030:e7468ad46bf8 - 02/07/2019 05:59 pm - Dan Smith

[py3] Fix up py2 related issues introduced into bitwise and memmap

The previous patch broke bitwise and memmap for py2, undetected because of a test environment snafu.

#495

Revision 3031:469720636052 - 02/07/2019 06:05 pm - Dan Smith

[py3] Make platform module and tests py3 compatible

#495

Revision 3032:231576875ed6 - 02/07/2019 06:09 pm - Dan Smith

[py3] Add two more test modules that are already py3 compatible

#495

Revision 3033:486e71950be8 - 02/07/2019 06:12 pm - Dan Smith

[py3] Make settings module compatible and add its tests

#495

Revision 3034:791439cb4c01 - 02/07/2019 06:37 pm - Dan Smith

[py3] Changes to many ui modules to make test_memedit_edits work

#495

Revision 3035:eef6ba4c88ac - 02/07/2019 06:43 pm - Dan Smith

[py3] Changes to make test_directory compatible

#495

Revision 3036:18f71d4c5676 - 02/07/2019 06:44 pm - Dan Smith

[py3] Add pytest cache directory to hgignore

#495

Revision 3037:2d040269da09 - 02/07/2019 07:01 pm - Dan Smith

[py3] Changes to make test_shiftdialog compatible

This marks the point at which all the tests/unit tests pass in py3

#495

Revision 3038:af37acf5b2b6 - 02/07/2019 07:24 pm - Dan Smith

[py3] Some minimal driver 2to3 conversions for things imported directly in the ui

#495

Revision 3039:0b8dafeeadd0 - 02/07/2019 07:24 pm - Dan Smith

[py3] chirp/detect and chirp/logger 2to3 conversion

#495

Revision 3041:dc8d5ce1ae54 - 02/07/2019 07:27 pm - Dan Smith

[py3] ui/editorset and ui/importdialog 2to3 and removal of generic_xml

Not going to port generic_xml to py3, so just remove it where used

#495

Revision 3042:32c6ab04f4cc - 02/07/2019 08:28 pm - Dan Smith

[py3] Fix another unicode issue in directory and prevent drivers from crashing us

If a driver raises during detection, we should not kill the whole process.

#495

Revision 3043:333b57e27d81 - 02/07/2019 08:30 pm - Dan Smith

[py3] Fixes to generic_csv and icf relating to the detection process

#495

Revision 3044:11d979412511 - 02/07/2019 08:35 pm - Dan Smith

[py3] Changes to ui so that we can actually start under py3/gi

This adds a chirp.ui.compat module where we can pile things needed to handle python3 differently. Right now there are a bunch of things that just fail for reasons I don't fully grasp yet. Those go under a py3safe() decorator, which logs and then ignores them.

Also, there is an issue with the sensitive=True bit on the columns for empty memories. Need to figure that out.

#495

Revision 3045:d30b8db2c3ea - 02/07/2019 09:47 pm - Dan Smith

[py3] Remove a big debug print from a previous patch

#495

Revision 3046:05e0a3d4abca - 02/07/2019 09:50 pm - Dan Smith

[py3] 2to3 run_tests.py and re-enable importing it in the unittest adapter

Also add pyserial to requirements because this is the first time we have hit that import.

#495

Revision 3047:b8f4eeb75e96 - 02/07/2019 10:04 pm - Dan Smith

[py3] More bitwise changes to be compatible with py2 and py3

With this, the py2 driver tests run successfully, and the bitwise unit tests run on py3.

#495

Revision 3048:af861c242c93 - 02/07/2019 11:07 pm - Dan Smith

[py3] Re-re-do of the charDataElement in bitwise

So, I've been a little bit on unicode autopilot, just trying to get the data types to match up. However, once getting into the work of making the drivers work, we have a ton that do things like this:

```
memobj.name = 'foobar\xFF'
```

because the radio needs the \xFF terminating byte to know where the end of the string is. We need to not encode/decode that even on py3 because we won't end up with the right value in memory if we do (and we'll fail to encode/decode some values).

So, this cheats our unicodeness to end up with single-byte characters and the right things get into/out of memory for now. Ideally all drivers would be converted to bytestrings for things like that, but, babysteps.

Yes, I know how unicode works in Python, and yes I know this is not how it's supposed to work.

#495

Revision 3049:dba08af1a915 - 02/08/2019 12:35 am - Dan Smith

[py3] More bitwise fixes for array operations and set_raw()

Found while converting real drivers.

#495

Revision 3050:f8f0b8f419e9 - 02/08/2019 12:39 am - Dan Smith

[py3] Another obligatory float->int conversion for an index

#495

Revision 3051:678da4def93e - 02/08/2019 12:41 am - Dan Smith

[py3] A workaround for match_model legacy, and test harness updates

This is enough to run the driver tests in py3 to actually start hitting issues. The filestring hack in directory just lets us get to some of them without having to convert match_model on all of them. Same deal with skipping drivers that don't load if CHIRP_DEBUG is set.

#495

Revision 3052:411600862e2b - 02/08/2019 12:56 am - Dan Smith

[py3] Fix generic_csv writing to file

We need to open the file non-binary so we can write strings to it.

#495

Revision 3053:0a6ecc5d0964 - 02/08/2019 12:57 am - Dan Smith

[py3] Another icf detection fix - unicode errors mean no

#495

Revision 3054:e487e7268dcb - 02/08/2019 12:57 am - Dan Smith

[py3] Fix writing config

#495

Revision 3055:5879a98842c7 - 02/08/2019 12:58 am - Dan Smith

[py3] Fix bankedit store - the new gtk does not like type mismatches

#495

Revision 3056:d0ddea005a4 - 02/08/2019 01:00 am - Dan Smith

[py3] Graceful fail on set_url_hook for about dialog

#495

Revision 3057:1297c390771f - 02/08/2019 02:24 am - Dan Smith

[py3] Fix setup import drivers.* routine to be like the others

Probably should DRY this up :P

#495

Revision 3058:c4013437b221 - 02/08/2019 02:27 am - Dan Smith

[py3] Fix up a bunch of gtk3 issues

- Removed the authors on the about box as neither the gtk2 compat or gtk3 APIs seem to work. No offense to the other contributors, we can add it back when we figure it out.
- Fix the get_cell_renderers() missing stuff in memedit and miscwidgets
- Quiet some of the py3safe() warnings which are known and not critical

#495

Revision 3059:435a3c4776b5 - 02/08/2019 02:59 am - Dan Smith

[py3] Fix memedit cut/copy/paste for GTK3

#495

Revision 3060:b757a3421416 - 02/08/2019 04:21 am - Dan Smith

[py3] Fix up miscwidget.make_choice() and clone dialog

#495

Revision 3062:282eb14a4420 - 02/08/2019 09:15 pm - Dan Smith

[py3]: Fix run_tests.py:

- Import hack to work from a variety of places. This is really ugly, but these are going away once the driver files can at least be imported.
- Something in the 2to3ing earlier broke the settings tests. Had to add len and getitem to RadioSettingValueString, which caused an infinite loop in the test's compare_settings. I dunno what I was thinking with the latter, but it seems like a infinite loop waiting to happen and it's hard to see through the blur how that ever worked.

#495

Revision 3063:79af8bd2b626 - 02/08/2019 09:15 pm - Dan Smith

[py3] Make hexprint work in py3 and add a test

#495

Revision 3064:956e3724d777 - 02/08/2019 09:15 pm - Dan Smith

[py3] Add a CompatSerial which will converse in strings

Gate this on the driver having a flag that indicates it needs this help.

#495

Revision 3065:2cffab241886 - 02/08/2019 09:15 pm - Dan Smith

[py3] Make the straight string encoders both work on sequences

#495

Revision 3066:3175366ac9c8 - 02/08/2019 09:15 pm - Dan Smith

[py3] 2to3 for chirpc and quick import * hack

Also fixed float division in the chirp_common.Status so that chirpc will work

#495

Revision 3067:4fa5ebebe6ab - 02/08/2019 09:15 pm - Dan Smith

[py3] Add a str-compatible MemoryMap implementation

I really hate to do this, but the more I dig into making a driver actually work, I think this needs to be the default. I retained the bytes-clean version as a base, and we can/should make new drivers use it, as well as convert drivers to use it in this process. Bitwise still uses it under the covers, so this is really just a usability shim to buy us some more time.

#495

Revision 3068:a8509ba81386 - 02/08/2019 09:15 pm - Dan Smith

[py3] Add a string-compatible version of the struct module

With this, even struct.pack,unpack work on our strings.

#495

Revision 3069:7b4ced760655 - 02/08/2019 09:15 pm - Dan Smith

[py3] Make icf work with python3

Just importing our modified struct module lets me run chirpw in python3, upload and download from an id800.

#495

Revision 3072:dfb129d5ba3e - 02/09/2019 04:28 am - Dan Smith

[py3] Fix some issues introduced by recent bitwise/memmap comap changes

- Make bitwise arrayDataElement *getitem* support integer-ifying slices too
- Support setting a single value in memmap to an integer as apparently a lot of places depend on that working.
- Don't initialize a MemoryMap (which is string-based) with bytes

#495

Revision 3073:7d5cf3c34327 - 02/09/2019 04:30 am - Dan Smith

Update from default branch

#495

Revision 3079:1413d8e4688e - 02/09/2019 05:45 am - Zhaofeng Li

[py3] bitwise.py: Python 3-ize

#495

Revision 3074:52d7717bdb69 - 02/09/2019 05:46 am - Zhaofeng Li

[py3] chirp_common.py: Fix file saving

#495

Revision 3085:6147365892e1 - 02/09/2019 05:49 am - Zhaofeng Li

[py3] uvb5.py: 2to3 with manual fixes

#495

Revision 3086:6f5c2af0b6f1 - 02/09/2019 06:42 am - Zhaofeng Li

[py3] uvb5.py: Adapt for new Serial

#495

Revision 3075:fdebb8acaf89 - 02/09/2019 06:43 am - Zhaofeng Li

[py3] memmap.py: Fix slicing

#495

Revision 3076:aa4dd87eb5d0 - 02/09/2019 06:44 am - Zhaofeng Li

[py3] memedit.py: Convert power to str

#495

Revision 3077:bf10d5382fb1 - 02/09/2019 05:44 pm - Dan Smith

Fix some style and unittest regressions in recent patches

#495

Revision 3078:a6fae4d90d5d - 02/09/2019 06:19 pm - Dan Smith

[py3] Fix bitwise structDataElement indexing with float

#495

Revision 3080:76547116bfb4 - 02/09/2019 07:49 pm - Dan Smith

[py3] Add some tests for the bitwise integer division operators

#495

Revision 3081:5295120a1d72 - 02/09/2019 07:49 pm - Dan Smith

[py3] Make bitwise get_raw() methods take asbytes=bool

This will enable modern drivers to get bytes instead of string

#495

Revision 3082:6a25fcc171b7 - 02/09/2019 08:36 pm - Dan Smith

[py3] Split py3 tox environment into py3unit and py3driver

#495

Revision 3083:ecd9578bfd2e - 02/09/2019 09:39 pm - Dan Smith

[py3] A bunch of fixes to drivers that already import/run under py3

With this, all driver tests pass on py3 and py2, but note that most of the drivers are still not even importing.

#495

Revision 3095:ee99c5f78dff - 02/10/2019 01:49 am - Zhaofeng Li

[py3] th350.py: 2to3, with many manual fixes

#495

Revision 3087:b522105293c8 - 02/10/2019 02:06 am - Zhaofeng Li

[py3] bitwise.py: Always return a bytes-compatible object when asbytes=True

This commit ensures that drivers that ask for bytes representations will always get bytes or a bytes-compatible object, whether on Python 2 or 3.

python-future provides a mostly-complete implementation of bytes for Python 2.

#495

Revision 3088:18667f11caf3 - 02/10/2019 06:13 pm - Dan Smith

[py3] Generalize the safe import routine

#495

Revision 3089:3b5b3b2d142e - 02/10/2019 06:14 pm - Dan Smith

[py3] Include requirements.txt and tests in sdist build

This will help make it easier to build and run tests on py2

#495

Revision 3092:ecb0a3160b24 - 02/10/2019 07:35 pm - Dan Smith

[py3] Fix broken win32 platform 2to3 conversion

The 2to3 of platform.py converted a unicode() to a str(), which made us no longer compatible on win32 with py2.

#495

Revision 3093:e650db8eaf3b - 02/10/2019 09:29 pm - Dan Smith

[py3] Fix safe_import_drivers() for win32

The glob'ing doesn't work on win32, especially in the frozen build, so just do the old thing if we are on win32.

#495

Revision 3094:ac82ed066690 - 02/10/2019 10:41 pm - Dan Smith

[py3] Include future in the py2 test environment

#495

Revision 3097:d1dd7c27f59c - 02/11/2019 08:07 pm - Dan Smith

[py3] Make mmap use future bytes

#495

Revision 3098:ce530aea28a0 - 02/11/2019 08:18 pm - Dan Smith

[py3] Clean up hexprint to not choke on bytes, newbytes, or string

#495

Revision 3099:08c9f62dfa63 - 02/11/2019 09:15 pm - Dan Smith

[py3] Add an icom clone simulator (for non-raw mode) and test

This tests regular icom radios for byte-clean cloning and includes the necessary changes to icf.py to make it work.

Tested on an ID800 and ID880

This doesn't address the live-mode radios, which still function under py2 properly. I don't have an IC-2730 to test with, but wrote unit tests for the frame encoding routines, so hopefully that works as well.

#495

Revision 3100:0aec6e700704 - 02/11/2019 09:24 pm - Dan Smith

[py3] Fix memedit regression in aa4dd87eb5d0 for non-power radios

If a radio doesn't set mem.power, we would str(None) into the liststore, which caused us to fail to detect it being unset when trying to edit a memory.

#495

Revision 3101:2113b42272b4 - 02/11/2019 09:59 pm - Dan Smith

[py3] Fix memedit context menu to be gtk2/3 compatible

#495

Revision 3102:db7233c1a8c4 - 02/11/2019 10:03 pm - Dan Smith

[py3] Fix up the properties dialog in memedit for gtk3

#495

Revision 3104:53ff47955595 - 02/11/2019 11:05 pm - Dan Smith

[py3] Fix initializing memory range SpinButton on gtk3

#495

Revision 3105:80fd08860e44 - 02/12/2019 12:21 am - Dan Smith

[py3] Put a py3clean environment in the list before the py3 jobs

This helps prevent python3 from stumbling over py2 cache files

#495

Revision 3106:d3b3da68341e - 02/12/2019 06:41 pm - Dan Smith

[py3] Fix making empty memories not sensitive on gtk3

We can do this in the render function (probably always should have), which works on both.

#495

Revision 3107:ba8153d6a275 - 02/12/2019 07:04 pm - Dan Smith

[py3] Fix settings editor for gtk3

#495

Revision 3108:7cdaf1d7e87a - 02/12/2019 07:15 pm - Dan Smith

[py3] Minor 2to3 and struct wrapper conversion for uv5r driver

This makes the uv5r driver clone operations work under py3, but with the compatibility layers in place.

#495

Revision 3109:e8c74cd501e0 - 02/12/2019 08:18 pm - Dan Smith

[py3] Convert uv5r driver to byte clean on py3

#495

Revision 3110:a7ab8bbab3d1 - 02/12/2019 10:49 pm - Dan Smith

[py3] Fix editing empty memories on gtk3

In order to go from empty/sensitive=False to editing in gtk3 we have to do a little more work.

#495

Revision 3111:484417eafc7d - 02/12/2019 11:49 pm - Dan Smith

[py3] Make _init_macos() work for pygobject and gtk-mac-integration 2.1.3

#495

Revision 3112:cb07236d561a - 02/13/2019 01:16 am - Dan Smith

[py3] Fix icom cloning when a MemoryMap (not bytes) was handed to us

#495

Revision 3113:af6e7bca1b76 - 02/13/2019 01:41 am - Dan Smith

[py3] 2to3 for id31 driver, related fixes for id51 and id51plus

The id31 driver required minor tweaks to be py3 compatible, which brought the id51 and id51plus modules online. Tested with a real id31, which is probably good enough for the id51s as well.

#495

Revision 3114:169d859a06be - 02/13/2019 05:10 pm - Dan Smith

[py3] Make kenwood_live radios use native serial

All the kenwood_live drivers speak ASCII to the radio using the exact same low-level routine. They don't do any binary mapping or manipulation. Thus, we really only need to do the usual unicode handling, and in one place. With this change, all kenwood live radios should be good and not need independent testing. Verified with a real TM-D700.

#495

Revision 3115:ddc858f0f5c5 - 02/13/2019 05:58 pm - Dan Smith

[py3] Make the StringStruct wrapper encode/decode string arguments

This is needed for further compatibility

#495

Revision 3116:ac6c0198c6cd - 02/13/2019 05:59 pm - Dan Smith

[py3] Straight 2to3, plus StringStruct conversion of h777 and friends

Tested with a real Arcshell AR-6 (BF-888S)

#495

Revision 3117:70ea1d8f6b50 - 02/13/2019 07:11 pm - Dan Smith

[py3] Make h777 driver byte clean for py3

Tested with Arcshell AR-6

#495

Revision 3118:1cbb593169c3 - 02/13/2019 07:50 pm - Dan Smith

[py3] Fix integer division in from_NHz() helpers

NOTE: These appear to be unused, except for one use of from_GHz() in the icq7 driver. AFAICT, these have always been wrong in the tree, so this just converts the division and adds (wrong) tests to validate the division change, but they should be fixed (along with the icq7 driver).

#495

Revision 3119:98b61504c85a - 02/13/2019 07:51 pm - Dan Smith

[py3] 2to3 of the icq7 driver

Untested with real hardware.

#495

Revision 3120:36069e8cbf14 - 02/13/2019 08:01 pm - Dan Smith

[py3] 2to3 of btech.py

Not tested with real hardware

#495

Revision 3121:f865c3640784 - 02/13/2019 09:25 pm - Dan Smith

[py3] Attempt to make btech driver byte-clean

Not tested with real hardware.

#495

Revision 3122:5a6dbbf93a28 - 02/13/2019 09:36 pm - Dan Smith

[py3] Make the clone test on py3 check that serial.write() passes proper data type

If a radio claims NEEDS_COMPAT_SERIAL=False, make sure it is writing with bytes and not a string (and vice versa). This is not perfect because these are just clone error tests, and all of the clone routine for a driver isn't going to get hit here. But, it's something.

#495

Revision 3123:25bb687c2d84 - 02/14/2019 12:00 am - Dan Smith

[py3] 2to3 of ft817 and ft857

Tested with a real FT-817. Expect 857 and 897 to be fine as they share so much in common.

#495

Revision 3124:2c0c075411d3 - 02/14/2019 12:50 am - Dan Smith

[py3] Fix test_icom_clone finding the icom images

When I moved this into tests/unit I didn't realize it was not able to find the images directory, and thus not doing anything. Make it smarter to be based on the location of the test file itself.

#495

Revision 3125:80c89280166f - 02/14/2019 12:53 am - Dan Smith

[py3] Finally, unified test running across all environments and pythons

pytest doesn't support the loadTests() discovery protocol, which means it would not ever find the driver test adapter in tests/__init__.py. So, this adds another adapter just for pytest (et al) which will trigger the test loader and make it find them.

Now, with all environments and python versions, you get nicer pytest output, and can use expressions to run subsets of tests, like:

```
tox -epy3driver 880.*clone
```

#495

Revision 3126:da18bd82d325 - 02/14/2019 12:59 am - Dan Smith

[py3] Fix up style issues introduced by 2to3 in ft8x7

#495

Revision 3133:f90ecaf1c1ee - 02/14/2019 01:42 am - Dan Smith

[py3] Initialize the test radio once per unittest.TestCase instance

Also, don't parse a mmap file just to get_sub_devices(). Makes the tests run like 50% faster on my system.

#495

Revision 3134:991466582379 - 02/14/2019 02:38 pm - Dan Smith

[py3] 2to3 of the ic208 driver

Not tested with real hardware

#495

Revision 3127:09df96cdaa70 - 02/14/2019 09:34 pm - Dan Smith

Change the browser widget to use an HPaned instead of HBox

This means the tree panel can be resized, and is necessary for GTK3 to look and behave sanely.

Related to work for #495

Revision 3137:b2634dca74ce - 02/15/2019 01:13 am - Dan Smith

[py3] Fix a couple style issues introduced in recent ic208 patch

#495

Revision 3138:3b513b02b9a0 - 02/15/2019 01:13 am - Dan Smith

[py3] Make the exception dialog parented to the main window

This is probably applicable to the main branch, but for some reason it's really broken in gtk3 without it.

#495

Revision 3139:9d2894cd3a3d - 02/15/2019 01:15 am - Dan Smith

[py3] Make YaesuChecksum operate on bytes

#495

Revision 3140:f30fddae7674 - 02/15/2019 01:15 am - Dan Smith

[py3] Use bytes() instead of byte literal when constructing test mmap

#495

Revision 3141:75702dd3ff96 - 02/15/2019 01:17 am - Dan Smith

[py3] Make ft817 byte-clean

Tested with a real FT-817, expect ft857 and ft897 to be okay since they share so much.

#495

Revision 3142:12f949800e21 - 02/15/2019 01:17 am - Dan Smith

[py3] 2to3 of ft7800 module

Tested with a real FT-7800

#495

Revision 3143:ff69883a7018 - 02/15/2019 01:18 am - Dan Smith

[py3] Convert ft7800 to byte-clean

Tested with a real FT7800

#495

Revision 3144:4f0c5e639610 - 02/15/2019 01:21 am - Dan Smith

[py3] Convert yaesu_clone to byte-clean

Tested with a real VX-7

#495

Revision 3145:2625e80cb65a - 02/15/2019 01:21 am - Dan Smith

[py3] Fix up python3 related failures found by the new delete memory test

#495

Revision 3146:6902d72607a2 - 02/15/2019 04:16 pm - Dan Smith

[py3] 2to3 conversion of vx8

Tested with a real VX-8DR

#495

Revision 3147:cee266814117 - 02/15/2019 04:23 pm - Dan Smith

[py3] 2to3 conversion of vx2

Not tested with a real radio, but it passes tests and uses the yaesu_clone module

for cloning, so it is likely okay.

#495

Revision 3148:c94669ab4df2 - 02/15/2019 04:37 pm - Dan Smith

[py3] 2to3 of vx3 module

Tested with a real VX-3

#495

Revision 3151:762155359728 - 02/15/2019 04:49 pm - Dan Smith

[py3] Fix generic_tpe import of UserDict for python3

Untested as we have no sample files, but since this depends almost entirely on the generic_csv driver (which works), it should be okay.

#495

Revision 3152:c949f3c6bbe9 - 02/15/2019 05:13 pm - Dan Smith

[py3] 2to3 and minor hand conversion of tk8102

Tested with a real TK-8102

#495

Revision 3153:6af06d9840a6 - 02/15/2019 05:37 pm - Dan Smith

[py3] Convert tk8102 driver to byte-clean

Tested with a real TK-8102

#495

Revision 3154:49d2862932d4 - 02/15/2019 05:47 pm - Dan Smith

[py3] Remove generic_xml driver

This never really caught on, nor got the attention or maintenance it really needed. I dunno if it even works in py2 for much anymore, but I don't think converting it to py3 makes any sense.

#495

Revision 3149:3589a27c8c5f - 02/15/2019 06:31 pm - Dan Smith

Fix Edges::do_oddsteps() test

This test was completely wrong, and not really running for any driver. Realized it was doing something stupid while trying to run it against the CSV driver (which supports DC->daylight).

Some radios were claiming to support steps they don't support because they were taking the full list in chirp_common as a default, which is clearly wrong. This patch changes the default to a very conservative set, which may cause some drivers to no longer accept some frequencies that may be valid because they were not declaring their real supported list.

This also uncovered at least one buggy driver that can't support anything other than 5kHz step frequencies (ic2730).

Found in the pursuit of #495

Revision 3157:d76f1cc19c0f - 02/15/2019 07:04 pm - Dan Smith

[py3] Make RadioSettingGroup sortable in python3

This is a fix for some error handling in the tests that assumes it can sort these objects.

#495

Revision 3158:08ada200f7c4 - 02/15/2019 07:16 pm - Dan Smith

[py3] Fix ict7h driver regression from 2to3 conversion

This fixes the ict7h driver for 12.5kHz step frequencies, uncovered by the recent fix to the oddsteps test.

#495

Revision 3159:6c757439562b - 02/15/2019 07:48 pm - Dan Smith

[py3] Enable testing the CSV and HMK drivers

This fixes some test infrastructure so we can run the tests on these, as well as makes the two drivers work properly in py3. I guess we were never running the tests against things like this, even though we should be.

#495

Revision 3160:b0f3fbc1d36b - 02/15/2019 08:02 pm - Dan Smith

[py3] Minor 2to3 change for kenwood_itm driver

No image in tree, so untested.

#495

Revision 3202:47c355d2b47c - 03/09/2019 11:23 pm - Dan Smith

Add a universal urlretrieve that works on both py2 and py3 and sets the User-Agent

This is needed for py3 compatibility as well, and is related to the RepeaterBook CDN bug.

Related to #495

Related to #6563

Revision 3203:4fc94bb9a79c - 03/09/2019 11:23 pm - Dan Smith

Fix up the miscwidgets.make_choice() for non-editable combo boxes on gtk3

This was never finished before, and required a little more monkeywork to get things working. This also fixes up the RepeaterBook prompting to work with the new mechanism.

#495

Revision 3204:c0da6833e0a0 - 03/09/2019 11:35 pm - Dan Smith

Make RepeaterBook querying work on py3

...and add a test.

#495

Revision 3283:9a35903ad54d - 12/05/2019 09:58 pm - Dan Smith

[py3] Fix headless logger init on python3 because of file() usage

#495

Revision 3284:df97de9c2f26 - 12/06/2019 01:11 am - Dan Smith

[py3] Make directory safe import defensive on win32 too

#495

Revision 3286:0d275dfae990 - 12/09/2019 07:59 pm - Dan Smith

Fix load_mmap() to provide the right mmap class based on the compat flag

#495

Revision 3315:afbde750bbc2 - 12/10/2019 04:11 pm - Dan Smith

[py3] Fix icom simulator and 2730 driver after default and py3 merge

The 2730 had a py3 syntax error and the icom clone simulator was not handling the W32E's off-by-one hack.

Related to #495

Revision 3317:e20f02b78ee2 - 02/06/2020 01:35 am - Dan Smith

[py3] Log an error on memory load failure

...but do not choke.

#495

Revision 3321:38612e24eaaf - 02/08/2020 07:52 pm - Dan Smith

Merge with default

#495

Revision 3322:b750fa9b51ec - 02/08/2020 07:52 pm - Dan Smith

Set row label to mem.number to account for zero-origin radios

Related to #495

Revision 3325:fd05eb0b2f48 - 02/11/2020 03:27 am - Dan Smith

[py3] Auto resize grid columns larger when required after editing

#495

Revision 3326:9fe07865774d - 02/11/2020 03:27 am - Dan Smith

[py3] Fix memory properties editor

#495

Revision 3327:2aead0568fdc - 02/11/2020 03:29 am - Dan Smith

[py3] Implement editing multiple memories via properties at once

#495

Revision 3328:eaf14ed3bdd3 - 02/11/2020 03:32 am - Dan Smith

[py3] Set column editor instead of per-cell

Massive improvement in grid setup time.

#495

Revision 3329:299220c20870 - 02/11/2020 03:41 am - Dan Smith

[py3] Implement developer-mode radio browser

#495

Revision 3332:41f2e6f61d45 - 02/12/2020 01:38 am - Dan Smith

Add a decorator/contextmanager that wraps a dangerous operation with an error message

#495

Revision 3333:8c4b5d81d0e3 - 02/12/2020 01:38 am - Dan Smith

Add cross tone modes and rx_dtcs to the memory editor.

#495

Revision 3334:c81b37b75846 - 02/12/2020 01:38 am - Dan Smith

Center the MemoryDialog and do not reload the browser once loaded.

#495

Revision 3335:aef3decd1657 - 02/12/2020 01:38 am - Dan Smith

Add missing power level to memedit

Also fix off-by-one errors when loading raw memories from 1-origin radios

#495

Revision 3336:79a84f68552a - 02/12/2020 01:41 am - Dan Smith

Allow limiting driver loading for speed during development

This lets you only load the driver you are working on when running from the command line, which speeds up startup quite a bit (especially if working on a network share).

#495

Revision 3337:d21e9c3f6604 - 02/12/2020 01:43 am - Dan Smith

Add Radio->Developer functions

This adds two things for developers to the Radio menu:

1. A "reload driver" option, which reloads the current driver module in-place, and attempt to hot-swap the current UI's editorset with a new copy of the driver with the old driver's memory loaded.
2. An "interact" option which drops the terminal to an interactive console where you can poke at the driver and UI application manually.

#495

Revision 3339:be8b32bc2746 - 02/12/2020 04:19 am - Dan Smith

[py3] More wrestling with the memory properties dialog and non-string choices

#495

Revision 3340:0b48c663d81e - 02/12/2020 04:19 am - Dan Smith

[py3] Make the driver reload avoid closing the editorset early

This helps avoid losing the current state if the driver fails to load.

#495

Revision 3341:e940eb876be7 - 02/12/2020 04:19 am - Dan Smith

[py3] Fix use of deprecated Menu::AppendItem to avoid warnings

#495

Revision 3342:3abe05eda6e7 - 02/12/2020 04:19 am - Dan Smith

[py3] Add --inspect argument to chirpwx for wxPython debugging

#495

Revision 3343:e8df990ff281 - 02/12/2020 04:19 am - Dan Smith

[py3] Fix second+ editors from being shown when selected on macos

For some reason intercepting the EDITOR_CHANGED event was causing the default behavior of showing the editor pane to not happen on macos (only). Using the EDITOR_CHANGING for our selected() purposes instead works around it.

#495

Revision 3344:90389810f4d1 - 02/12/2020 04:19 am - Dan Smith

Make memedit filter memory names according to the radio rules

#495

Revision 3345:a2d6850b370d - 02/12/2020 04:19 am - Dan Smith

[py3] Make the memory editor use fixed-width fonts, like it should

#495

Revision 3346:9ef6c30fe30c - 02/12/2020 04:19 am - Dan Smith

[py3] Add another option to reload the driver and the currently-open file

Makes developing against another piece of software much easier.

#495

Revision 3347:c703cfc0119b - 02/12/2020 04:19 am - Dan Smith

[py3] Make the radio browser pop open the first level by default

#495

Revision 3348:b1a438ea82d9 - 02/12/2020 04:19 am - Dan Smith

[py3] Display type and offset information as a tooltip in browser

#495

Revision 3349:0b279e353b36 - 02/12/2020 04:19 am - Dan Smith

[py3] Center the clone upload/download boxes by default

These don't center by default on macos for some reason, but do elsewhere.

#495

Revision 3350:38ec9f94e7a6 - 02/13/2020 01:15 am - Dan Smith

[py3] Make the browser support edits, validation and fixed-width fonts

#495

Revision 3351:68534f20c141 - 02/14/2020 12:35 am - Dan Smith

[py3] Re-select the last editor when reloading the radio.

#495

History

#1 - 02/03/2013 08:16 pm - Dan Smith

- *Status changed from New to Rejected*

PyGTK is not supported on Python 3.x, and is a core requirement for CHIRP. There are a lot of applications that fit into this category with similar requirements, and as a result, python 2.x isn't going anywhere any time soon.

Closing as rejected since this is not a bug.

Thanks!

#2 - 09/08/2018 05:45 am - Richard Shaw

5 years later this is no longer the case. Fedora is actively working on transitioning to Python 3 to stay ahead of the 2020 EOL for Python 2.

#3 - 02/08/2019 01:43 pm - Dan Smith

- *Status changed from Rejected to In Progress*

- *Chirp Version changed from 0.2.3 to 0.4.0*

Work in progress in the "py3" branch

#4 - 08/28/2019 10:33 am - Richard Shaw

Just an FYI but chirp will likely be removed from Fedora as of F32 (currently working towards beta freeze) due to python 2 removal.

https://bugzilla.redhat.com/show_bug.cgi?id=1731397

#5 - 01/25/2022 01:09 am - Zhaofeng Li

- *Status changed from In Progress to Closed*

- *% Done changed from 0 to 100*

Applied in changeset commit:6147365892e1.