Amateur Radio in Fedora

Presented by
Jaroslav Škarvada / OK2JRQ
jskarvad@redhat.com
Outline

- Introduction of Amateur Radio SIG
- What is Amateur Radio
- HamLib
- SvxLink
- NEC2
- Other Software
- How to help
Introduction

- Amateur Radio SIG: [Amateur Radio SIG](http://fedoraproject.org/wiki/SIGs/AmateurRadio)
  - http://fedoraproject.org/wiki/SIGs/AmateurRadio
- Tasks:
  - Work with upstream to get more software update for modern Linux distribution.
  - Package and test applications.
  - Submit packages for review.
Amateur Radio SIG

- Communication:
  - IRC channel: #Fedora-Hams

- Packages:
  - [https://fedoraproject.org/wiki/SIGs/AmateurRadio/Packages](https://fedoraproject.org/wiki/SIGs/AmateurRadio/Packages)
  - Currently about 60 packages.
  - And more are worked on.
What is Amateur Radio?

- Licensed non-commercial use of designated radio bands:
  - self-education, training, experiments,
  - exchange of messages,
  - recreation,
  - emergency communication.
- Ham-spirit.
More on Amateur Radio

- To obtain license you have to pass exams.
- Callsigns are used to identify stations world-wide.
- Various operations:
  - DXing, contesting, chatting, satellite operations, repeater operations, ...
- Various bands:
  - VLF, LF, HF, VHF, UHF, SHF, ...
- Various modes:
  - Analog (CW, SSB, FM, AM), digital (AX.25, RTTY, AMTOR, PACTOR, SSTV, D-STAR, ...).
Practice

- Make a contact.
- Exchange data – callsigns, reports, (contest codes)
- Exchange QSL cards via QSL-bureau, snail-mail, Internet:
Ham Radio Control Libraries:

- Standardized API to control radios, rotators.
  - Developers need not to cope with various standards and communication protocols.
- API is provided by libhamlib, the frontend.
- Backend libraries implements rig specific communication.
- Bindings for Perl, Python, TCL, C++.
- Frontend library loads the appropriate backend on demand.
Control SW is included:

- **rigctl** – command line interface.
- **rigctld** – TCP daemon.
- **rpc.rigd** – RPC daemon (prognum: 536871065).
- **rot** – rotators control.
- **rigmem** – backup and restore of rig memory.
- **rigswr** – measure SWR vs frequency through rig.
- **rigsmtr** – measure “radiation” pattern through rig S-meter and rotator.
Command line examples:

- List of support rigs:
  - $ rigctl -l

- Set frequency to 7.253 MHz on Icom IC-706 MkIIG:
  - $ rigctl -m 311 -r /dev/ttyUSB1 F 7253000

- Interactive mode through rpc.rigd
  - $ rigctl -m 1901 -r localhost

- Interactive mode through rigctld on port 4530:
  - $ rigctl -m 2 -r localhost:4530
```c
#include <stdio.h>
#include <hamlib/rig.h>

int main(int argc, char *argv[]) {
    RIG *my_rig;
    freq_t freq;

    rig_set_debug(RIG_DEBUG_NONE);
    rig_model_t myrig_model = RIG_MODEL_DUMMY;
    my_rig = rig_init(myrig_model);
    rig_open(my_rig);
    rig_set_freq(my_rig, RIG_VFO_CURR, 7253000);
    rig_get_freq(my_rig, RIG_VFO_CURR, &freq);
    printf("%f\n", freq);
    rig_close(my_rig);
    rig_cleanup(my_rig);
}
```
Echolink

- Developed by K1RFD.
- Linkage of repeaters / amateur radio stations through internet (worm-holes).
- http://echolink.org
- VoIP with OOB signalling
SvxLink

- Modular voice services system:
  - Autonomous radio controller.
  - Modules for simplex repeater, Echolink, voice mail, ...
  - Echolib for Echolink handling.
  - TCL based event handling system.
  - More modules can be added using TCL.
  - Support for multipath systems (software voter).
  - Logic core can be linked to radios via TCP/IP.
$ cat | g++ -o echolib-test -I/usr/include/svxlink/ -lecholib `pkg-config --libs --cflags sigc++-1.2` -lasynccore -lasynccpp -x c++ -
#include <AsyncCppApplication.h>
#include <EchoLinkQso.h>

using namespace Async;
using namespace EchoLink;

class MyClass: public SigC::Object {
    public:
    Qso *qso;
    MyClass(void) {
        qso = new Qso(IpAddress("192.168.1.2"), "OK2JRQ", "Yarda", "A test QSO");
        qso->connect();
        qso->sendChatData("Hello!");
        qso->disconnect();
        Application::app().quit();
    }
    ~MyClass(void) {delete qso;}
};

int main(int argc, char *argv[]) {
    CppApplication app;
    MyClass my_class;
    app.exec();
}
NEC2 – I

- Antenna modelling software for wire and surface antennas.
- NEC2C – Numerical Electromagnetics Code translation from FORTRAN to C.
- XNEC2C – GTK2 GUI for NEC2C.
- NEC2++
  - C++ interface and python bindings (automation).
  - Currently not in Fedora.
NEC2 – II

Amateur Radio in Fedora, Jaroslav Škarvada / OK2JRQ
GNU Radio

- SDK for software defined radios.
  - Written in Python, critical paths in C++.
  - Modular, signal processing blocks.
  - Universal Software Radio Peripheral (USRP2) + Daughterboards:
    - Up to 6 GHz RX / TX
  - It can also run without external HW (simulation, processing of pre-recorded data).
  - A platform for rapid building and exploring radios / communication systems.

- Applications (not counting the amateur radio ;).
  - RFID, GSM BTS, GPS, radar, DVB-T, Bluetooth, ZigBee, ...
Other Software

- Tucnak 2 – VHF/UHF contesting logger.
- QSSTV – Slow Scan Television Receiver.
- Soundmodem – Use your soundcard as Packet Radio modem.
- GridLoc – Tool for calculation of Maidenhead QRA Locators.
How to help?

- Get more AR packages into Fedora.
- Test features & fix bugs.
- Cooperate with upstreams, send patches:
  - Seemless support of latest build tools, kernels, etc.
  - Proper usage of the GNU coding standards.
References

- Amateur Radio SIG:
  - http://fedoraproject.org/wiki/SIGs/AmateurRadio/

- HamLib:
  - http://www.hamlib.org/

- Radio Club OK2KOJ:
  - http://www.radio.feec.vutbr.cz/ok2koj/

- Czech Radio Club (CRK):
  - http://www.crk.cz/
Questions?

Thank you.