New design for remote radio system using Internet Protocols and PiRLP with Raspberry Pi Linux computer

Gary Crum, KK7DV and David Jenkins, W7JFH
Raspberry Pi computer running Linux

Remote control application
Operational remote radio control system
PiRLP web site: http://www.irlp.net/pi/
Custom PTT protection

Controller Power-failure Protection Circuits

Note: The 5-volt supply powering either of these circuits must be the same supply that is powering the controller.
EchoLink connection method
SSH Linux access
Software structure: directories

- repeater@stn3434:~/$ find . -type d -print
- ./scripts
- ./scripts/admin_php_scripts
- ./bin
- ./features
- ./features/controller
- ./features/irlp_vpn
- ./features/speakdate
- ./features/IRLPvCON
- ./features/morse_id
- ./features/EchoIRLP
- ./features/EchoIRLP/scripts
- ./features/EchoIRLP/bin
- ./features/EchoIRLP/audio
- ./features/EchoIRLP/custom
- ./features/EchoIRLP/noupdate
- ./features/EchoIRLP/source
- ./features/EchoIRLP/docs
- ./features/star69
## Software structure: shell scripts

```bash
repeater@stn3434:~/$ ls scripts

<table>
<thead>
<tr>
<th>admin_php_scripts</th>
<th>control~</th>
<th>enable</th>
<th>flash_upgrade</th>
<th>nobackup.list</th>
<th>on,v</th>
<th>statupdate</th>
</tr>
</thead>
<tbody>
<tr>
<td>audio_level_test</td>
<td>control_20140317</td>
<td>enable~</td>
<td>idmorse</td>
<td>off</td>
<td>pi_send_info</td>
<td>statupdate_cron</td>
</tr>
<tr>
<td>audiotest</td>
<td>control_orig_20140128</td>
<td>end</td>
<td>idmorse~</td>
<td>off~</td>
<td>play_my_ids</td>
<td>troubleshoot-irlp</td>
</tr>
<tr>
<td>backup_for_reinstall</td>
<td>control,v</td>
<td>end~</td>
<td>idmorse,v</td>
<td>off_20140317</td>
<td>process_check</td>
<td>update</td>
</tr>
<tr>
<td>call</td>
<td>custom_wavplay</td>
<td>experimental_call</td>
<td>incrementcount</td>
<td>off_orig_20140201</td>
<td>pulsecheck</td>
<td>update_files_cron</td>
</tr>
<tr>
<td>call,v</td>
<td>decode</td>
<td>exp-x-reference</td>
<td>interval_id</td>
<td>off,v</td>
<td>record_my_ids</td>
<td>vu_audio_test</td>
</tr>
<tr>
<td>check_pgp_status</td>
<td>disable</td>
<td>failure</td>
<td>interval_id,v</td>
<td>on</td>
<td>ref-x-reference</td>
<td>vu_dtmf_test</td>
</tr>
<tr>
<td>clean_old_audio</td>
<td>disable~</td>
<td>failure~</td>
<td>ipupdate</td>
<td>on~</td>
<td>remote_admin</td>
<td>wavplay</td>
</tr>
<tr>
<td>common-functions.sh</td>
<td>dropcall</td>
<td>fifoecho</td>
<td>ipupdate_cron</td>
<td>on_20140317</td>
<td>rotatelogs</td>
<td></td>
</tr>
<tr>
<td>connect_to_reflector</td>
<td>dtmf_dial</td>
<td>find_best_server</td>
<td>irlp.crons</td>
<td>on_orig_20140201</td>
<td>send_wave_files</td>
<td></td>
</tr>
<tr>
<td>control</td>
<td>dtmfregen</td>
<td>fix-udev</td>
<td>irlp_reset</td>
<td>on_to_remote</td>
<td>sfswrapper</td>
<td></td>
</tr>
</tbody>
</table>
```
Software customization

```bash
function id_morse {

    # KK7DV addition - log ID
    log "id_morse: starting function id_morse in interval_id"

    # KK7DV addition to omit ID for case where connection is from self
    if [ "`cat $LOCAL/echo_call"" = KK7DV ]; then
        log "id_morse: skipping ID because of connection from KK7DV"
        return 0
    fi

    # KK7DV addition to omit ID for case where KK7DV just disconnected
    if [ -f $LOCAL/echo_KK7DV ]; then
        log "id_morse: skipping ID because of recently disconnected KK7DV EchoLink call"
        rm $LOCAL/echo_KK7DV
        return 0
    fi

    # If the MORSE_TEXT variable is not set, we use the CALLSIGN variable
    if [ -z "$MORSE_TEXT" ]; then
        MORSE_TEXT="$CALLSIGN"; fi

    UU-:----F1 interval_id 11% L62 RCS:1.4 (Shell-script[bash])
```
Log files

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 10 2015</td>
<td>09:38:55</td>
<td>Connected echolink N7FOD 1</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>09:38:56</td>
<td>Connected speakfreely KK7DV-R 2</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>10:22:08</td>
<td>Disconnected bye N7FOD 1</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>10:22:10</td>
<td>Node Disconnect from EchoLink 7769 N7FOD</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>10:22:12</td>
<td>Disconnected bye KK7DV-R 0</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>16:56:10</td>
<td>Connected echolink KK4YRR 1</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>16:56:11</td>
<td>Inbound connection from KK4YRR denied (Locked Out)</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>16:56:22</td>
<td>EchoIRLP: sent chat KK7DV-R&gt;Connection not available, development in progress, contact Gary KK7DV <a href="mailto:gc2488@gmail.com">gc2488@gmail.com</a> 801-449-1426 for info.</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>16:56:22</td>
<td>Disconnected bye KK7DV-R 0</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>17:47:52</td>
<td>Connected echolink KK7DV 1</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>17:47:54</td>
<td>Connected echolink KK7DV-R 2</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>17:48:02</td>
<td>Starting function id_morse in interval_id</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>17:48:00</td>
<td>Skipping ID because of connection from KK7DV</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>17:48:32</td>
<td>EchoIRLP: disconnected rtcp_timeout KK7DV 1</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>17:56:34</td>
<td>EchoIRLP: echo_end: Created /home/irlp/local/echo_KK7DV flag file for ID skip.</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>17:56:34</td>
<td>Node Disconnect from EchoLink 6490 KK7DV</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>17:56:37</td>
<td>Disconnected bye KK7DV-R 0</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>17:58:07</td>
<td>Starting function id_morse in interval_id</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>17:58:07</td>
<td>Skipping ID because of recently disconnected KK7DV EchoLink call</td>
</tr>
<tr>
<td>Feb 10 2015</td>
<td>22:42:21</td>
<td>Connected echolink KC7RH 1</td>
</tr>
</tbody>
</table>
Future possibilities

• FC-301 radio interfacing and frequency-agile RS-232 serial control

• Information lookup using Raspberry Pi internet connection

• More complete logging and recording

• Voice recognition in style of Siri and Google
Similar computer systems

- System-on-a-Chip (SoC) inexpensive, systems running Linux
  - BeagleBone (TI SoC)  
    [http://beagleboard.org/bone](http://beagleboard.org/bone)
  - Edison (Intel SoC)  
  - Raspberry Pi 2 (Broadcom SoC)  
  - ODROID (Amlogic SoC)  
BeagleBoneBlack with Asterisk and AllStar

http://hamvoip.org
https://allstarlink.org
Any similar experience and interests?

Questions?
Contact info

• Gary Crum, KK7DV
  • CrumResearch@gmail.com
  • Tel. 801-449-1426
  • Sometimes on 146.62 - when commuting weekdays at 8:30am

• Dave Jenkins, W7JFH
  • dpjenkinsjr@gmail.com