Spotify Connect on CS

1. Event Scripts
-> Make sure you have last version of CS (3.361 or above) with Spotify event scripts located in *"/opt/spotify"*.

```
imm@conn-server:/opt/spotify $ ls -ll
total 8
-rwxrwxrwx 1 imm imm 2218 Mar  2 08:15 event.py
-rwxrwxrwx 1 imm imm   60 Feb 28 10:18 event.sh
```

2. Raspotify installation
-> Creates Spotify Connect device, which is available from Spotify App. For use this feature in Spotify App you have to have Spotify Premium Account.

Run command:

curl -sL https://dtcooper.github.io/raspotify/install.sh | sh

Open file:

sudo nano /etc/default/raspotify

Edit line OPTIONS with:

OPTIONS="--onevent /opt/spotify/event.sh"

```
#
# To choose a different output device (ie a USB audio dongle or HDMI audio out),
# use `--device` with something like `--device hw:0,1`. Your mileage may vary.
#
OPTIONS="--onevent /opt/spotify/event.sh"

# Uncomment to use a cache for downloaded audio files. Cache is disabled by
# default. It's best to leave this as-is if you want to use it, since
# permissions are properly set on the directory `/var/cache/raspotify'.
```

3. User imm in group audio
-> Adding user *"imm"* to group *"audio"*.

Run command:

sudo adduser imm audio

4. ALSA Loopback
-> Creates virtual recording device.

Run command:

sudo modprobe snd-aloop

Open file:

sudo nano /etc/modules

Add new two lines if not included:

snd_bcm2835
snd-aloop

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.

i2c-dev
snd_bcm2835
snd-aloop
```

5. Enable Audio module

-> Enable Audio module.

Open file:

sudo nano /boot/config.txt

Add or uncomment line:

dtparam=audio=on

```
# Enable audio (loads snd_bcm2835)
dtparam=audio=on
start_x=0
enable_uart=0
gpu_mem=32
```

6. ALSA default soundcard

-> Set default soundcard for ALSA.

Open file:

sudo nano /usr/share/alsa/alsa.conf

Edit lines:

defaults.ctl.card 1
defaults.pcm.card 1

```
defaults.namehint.basic on
# show extended name hints
defaults.namehint.extended on
#
defaults.ctl.card 1
defaults.pcm.card 1
defaults.pcm.device 0
defaults.pcm.subdevice -1
defaults.pcm.nonblock 1
```

7. Reboot
-> Make reboot for apply changes.

Run command:

sudo reboot

8. Check settings
-> Check if all is correct before next steps.

Run command and check that you got the same output as below screen. You should see Loopback device under Card 1:

aplay -l

```
imm@conn-server:~ $ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: ALSA [bcm2835 ALSA], device 0: bcm2835 ALSA [bcm2835 ALSA]
  Subdevices: 8/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
card 0: ALSA [bcm2835 ALSA], device 1: bcm2835 ALSA [bcm2835 IEC958/HDMI]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
card 1: Loopback [Loopback], device 0: Loopback PCM [Loopback PCM]
  Subdevices: 8/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
card 1: Loopback [Loopback], device 1: Loopback PCM [Loopback PCM]
  Subdevices: 8/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
```

Run command and check that you got the same output as below screen. You should see Loopback device under Card 1:

arecord -l

```
imm@conn-server:~ $ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: Loopback [Loopback], device 0: Loopback PCM [Loopback PCM]
  Subdevices: 8/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
card 1: Loopback [Loopback], device 1: Loopback PCM [Loopback PCM]
  Subdevices: 8/8
  Subdevice #0: subdevice #0
  Subdevice #1: subdevice #1
  Subdevice #2: subdevice #2
  Subdevice #3: subdevice #3
  Subdevice #4: subdevice #4
  Subdevice #5: subdevice #5
  Subdevice #6: subdevice #6
  Subdevice #7: subdevice #7
```

9. IceCast2 installation

-> Creates audio streaming server on port 50000.

Run command and during installation select **"NO"** to skip configuration:

sudo apt-get install icecast2

Open file:

sudo nano /etc/icecast2/icecast.xml

Edit block:

```
<limits>
   <burst-on-connect>0</burst-on-connect>
   <burst-size>0</burst-size>
</limits>
```

```
<limits>
    <clients>100</clients>
    <sources>2</sources>
    <threadpool>5</threadpool>
    <queue-size>524288</queue-size>
    <client-timeout>30</client-timeout>
    <header-timeout>15</header-timeout>
    <source-timeout>10</source-timeout>
    <!-- If enabled, this will provide a burst of data when a client
         first connects, thereby significantly reducing the startup
         time for listeners that do substantial buffering. However,
         it also significantly increases latency between the source
         client and listening client.  For low-latency setups, you
         might want to disable this. -->
    <burst-on-connect>0</burst-on-connect>
    <!-- same as burst-on-connect, but this allows for being more
         specific on how much to burst. Most people won't need to
         change from the default 64k. Applies to all mountpoints  -->
    <burst-size>0</burst-size>
</limits>
```

Edit block:

<listen-socket>
<port>50000</port>
</listen-socket>

```
<!-- You may have multiple <listener> elements -->
<listen-socket>
    <port>50000</port>
    <!-- <bind-address>127.0.0.1</bind-address> -->
    <!-- <shoutcast-mount>/stream</shoutcast-mount> -->
</listen-socket>
```

Open file:

sudo nano /etc/init.d/icecast2

Add new lines below:

LOG_PATH=/var/log/icecast2
sudo mkdir $LOG_PATH
sudo touch $LOG_PATH/access.log
sudo touch $LOG_PATH/error.log
sudo chown -R 777 $LOG_PATH
sudo chown -R icecast2 $LOG_PATH

```
# Defaults
CONFIGFILE="/etc/icecast2/icecast.xml"
CONFIGDEFAULTFILE="/etc/default/icecast2"
USERID=icecast2
GROUPID=icecast
ENABLE="false"


LOG_PATH=/var/log/icecast2
sudo mkdir $LOG_PATH
sudo touch $LOG_PATH/access.log
sudo touch $LOG_PATH/error.log
sudo chown -R 777 $LOG_PATH
sudo chown -R icecast2 $LOG_PATH

# Reads config file (will override defaults above)
[ -r "$CONFIGDEFAULTFILE" ] && . $CONFIGDEFAULTFILE

set -e

cancel() { log_failure_msg "$NAME daemon is disabled - read $CONFIGDEFAULTFILE"; };
```

Open file:

sudo nano /etc/default/icecast2

Edit line:

ENABLE=true

```
# Edit /etc/icecast2/icecast.xml and change at least the passwords.
# Change this to true when done to enable the init.d script
ENABLE=true
```

Run commands:

sudo systemctl start icecast2
sudo systemctl enable icecast2

10. Darkice installation

-> Creates audio bridge between ALSA Loopback and IceCast2.

Run command:

sudo apt-get install darkice

Create and open file:

sudo nano /etc/darkice.cfg

Write new lines below:

```
# see the darkice.cfg man page for details
# this section describes general aspects of the live streaming session
[general]
duration = 0 # duration of encoding, in seconds. 0 means forever
bufferSecs = 1 # size of internal slip buffer, in seconds
reconnect = yes # reconnect to the server(s) if disconnected
realtime = yes
rtprio = 3

# this section describes the audio input that will be streamed
[input]
device = hw:Loopback,1,0 # Alsa soundcard device for the audio input
sampleRate = 44100 # sample rate in Hz. try 11025, 22050 or 44100
bitsPerSample = 16 # bits per sample. try 16
channel = 2 # channels. 1 = mono, 2 = stereo

# this section describes a streaming connection to an IceCast2 server
# there may be up to 8 of these sections, named [icecast2-0] ... [icecast2-7]
# these can be mixed with [icecast-x] and [shoutcast-x] sections
[icecast2-0]
bitrateMode = vbr # variable bit rate
format = mp3 # format of the stream: mp3
quality = 1.0 # quality of the stream sent to the server
server = localhost # host name of the server
port = 50000 # port of the IceCast2 server
password = hackme # source password to the IceCast2 server
mountPoint = spotify # mount point of this stream on the IceCast2 server
name = Spotify # name of the stream
description = CS Spotify # description of the stream
genre = Custom # genre of the stream
#public = no # advertise this stream?
#localDumpFile = recording.mp3 # Record also to a file
```

```
# see the darkice.cfg man page for details

# this section describes general aspects of the live streaming session
[general]
duration        = 0               # duration of encoding, in seconds. 0 means forever
bufferSecs      = 1               # size of internal slip buffer, in seconds
reconnect       = yes             # reconnect to the server(s) if disconnected
realtime        = yes
rtprio          = 3

# this section describes the audio input that will be streamed
[input]
device          = hw:Loopback,1,0  # Alsa soundcard device for the audio input
sampleRate      = 44100            # sample rate in Hz. try 11025, 22050 or 44100
bitsPerSample   = 16               # bits per sample. try 16
channel         = 2                # channels. 1 = mono, 2 = stereo

# this section describes a streaming connection to an IceCast2 server
# there may be up to 8 of these sections, named [icecast2-0] ... [icecast2-7]
# these can be mixed with [icecast-x] and [shoutcast-x] sections
[icecast2-0]
bitrateMode     = vbr              # variable bit rate
format          = mp3              # format of the stream: mp3
quality         = 1.0              # quality of the stream sent to the server
server          = localhost        # host name of the server
port            = 50000            # port of the IceCast2 server
password        = hackme           # source password to the IceCast2 server
mountPoint      = spotify          # mount point of this stream on the IceCast2 server
name            = Spotify          # name of the stream
description     = CS Spotify       # description of the stream
genre           = Custom           # genre of the stream
#public          = no               # advertise this stream?
#localDumpFile   = recording.mp3    # Record also to a file
```

Create and open file:

sudo nano /etc/systemd/system/darkice.service

Add new lines below:

[Unit]
Description=DarkIce Live audio streamer
After=icecast2.service

[Service]
ExecStart=/usr/bin/darkice

[Install]
WantedBy=multi-user.target

```
[Unit]
Description=DarkIce Live audio streamer
After=icecast2.service

[Service]
ExecStart=/usr/bin/darkice

[Install]
WantedBy=multi-user.target
```

Run commands:

sudo chmod 777 /etc/systemd/system/darkice.service
sudo systemctl daemon-reload
sudo systemctl start darkice.service
sudo systemctl enable darkice.service

11. Reboot

-> Make reboot for apply changes.

Run command:

sudo reboot

12. All done

-> Now you can go to Spotify App and select Spotify Connect device. Device name can be changed in CS Media tab. Each time, when you click on play or stop, event script is called and LARA (configured by LARA IP address) is handled. Make sure LARA firmware is 3.6.006 or above.