

*This application note is used in this [blog post](#).*

## Play Bebop stream on VLC

Among the things we improved between the Bebop and the Bebop 2 is the video streaming: the video feedback that is sent through Wifi to the pilot on his smartphone/tablet or on the SkyController with HDMI glasses. We call this improved streaming "stream v2".

The stream v2 has been backported to the Bebop 1 since firmware version 3.2.0.

Stream v2 is based on an H.264 video encoding streamed over the RTP protocol. This means we are now compatible with common players supporting RTP such as VLC or mplayer. There is no session control protocol such as RTSP to initialize the video streaming, an ARSDK command must be used to start the streaming.

This article explains how to play a video stream from a Bebop with a third-party player (VLC for example) on a Linux PC.

*Please note that you cannot use the DeviceController to do this, as it consumes the incoming RTP packets. The program must simply open and maintain a connection to the drone and issue the "enable video streaming" command. It's VLC who will consume the incoming RTP packets.*

In order to start a stream, it is necessary to connect to the Bebop and issue an "enable video streaming" command. We can do this using a small program using the ARSDK. The BebopDroneStartStream program provided within this application note does this.

## Build the application

### Prerequisite: build the SDK

To use the BebopDroneReceiveStream Unix program you must have the 3.9.0 SDK compiled for the arsdk-native target. To do that, get the SDK:

```
repo init -u https://github.com/Parrot-Developers/arsdk_manifests.git -b
refs/tags/ARSDK3_version_3_9_0 -m release.xml
repo sync
./build.sh -p arsdk-native -t build-sdk -j
```

*If you encounter any problem, please read the [install documentation](#).*

## Download the application note

The source code of the program can be downloaded from [the release hosted on Github](#).  
Unzip this file in the folder of your choice.

## Build the application note

Create an environment variable named ARSDK\_ROOT that points to the root of your arsdk (the folder where you did the `repo init` command).

```
export ARSDK_ROOT=<SDK>
```

And finally, compile the program:

```
cd FOLDER_WHERE_YOU_EXTRACTED_THE_ZIP_FILE  
make
```

## Run the example

Firstly your PC must be connected to the Bebop Wifi access point and have a valid 192.168.42.x IP address. Launch the BebopDroneStartStream program. It connects to the drone and sends the "enable video streaming" command. Then it just waits for a CTRL+C key press to exit. Meanwhile the drone is streaming the video to your PC. The streaming stops when the BebopDroneStartStream program is ended.

In the application note files, there is an .sdp file with the following content:

```
c=IN IP4 192.168.42.1  
m=video 55004 RTP/AVP 96  
a=rtpmap:96 H264/90000
```

As there is no session control protocol such as RTSP on the Bebop, this SDP file describes the video streaming for the player to know where to find the media and what its type is. In this file 192.168.42.1 is the drone IP address and 55004 is the client-side RTP port. If you want to change the client port value, you must also change it in the BebopDroneStartStream program. RTP ports are exchanged between the drone and the controller in the discovery connection phase.

Now launch VLC and open the SDP file: choose open a network stream and as URL enter `file://FOLDER_WHERE_YOU_EXTRACTED_THE_ZIP_FILE/bebop.sdp`. You may want to click on "show more options" and reduce the buffering time: it is 1000 ms by default but it can be reduced to 200 ms for example to have less latency on the video.

If you want to use mplayer instead, simply open the SDP file as the mplayer input:

```
mplayer FOLDER_WHERE_YOU_EXTRACTED_THE_ZIP_FILE/bebop.sdp
```

Note that RTCP is not used with the RTP streaming at the moment, but may be added in the future.

*Don't forget to read our [documentation](#) and if you have any questions, feel free to ask them on our [forum](#) (if other devs haven't already asked/answered them).*