

# BitPath-DevKit

## User Guide Document

RadioStudio



## 1 Table of Contents

1	Table of Contents .....	2
2	Revision History .....	3
3	Login to the device.....	4
4	Verification.....	5
5	Configuration .....	6
6	Logging .....	7
7	Local Redis Client .....	8
8	Accessing Redis Over the Network .....	9
	Step 1. Accessing Redis over the local network.....	9
	Step 2. Accessing Redis using direct ethernet cable.....	10
9	References .....	11

## 2 Revision History

Version No	Date	Author	Change Log
1.0	25/04/2024	RadioStudio	

### 3 Login to the device

The Golden Image, built from the Raspberry Pi OS Lite version with a Debian version of Bullseye 64-bit, plays a crucial role in this setup process. The Working With Golden Image document provides detailed steps to download the Golden Image from the S3 Bucket and install it on the SD card. Once the rover boots up, find its IP address and SSH to the device using the default credentials provided.

UserName – bitpath

Password – S@sh@J@v1d

IP Address – <Device's IP Address>

Port No – 1729

**Example:** - ssh -p 1729 bitpath@<IP Address of the device>

## 4 Verification

- All the required libraries for the receiver are preinstalled in the golden image. Check the following locations to verify that the golden image was installed successfully.

```
Kernel-Version(Command – uname -r) – 6.1.19-v8+
Sony Module Mount Name – /dev/usb-it930x0
Receiver Repo – /home/bitpath/source/BP-Receiver/
Log Location – /var/log/DriveTest/bitpath.log
```

- The Receiver is configured to run on every boot, and its status can be verified using the command, **“sudo systemctl status LSRxIPTester-daemon”**

```
• LSRxIPTester-daemon.service - LSRxIPTester Daemon Service
Loaded: loaded (/etc/systemd/system/LSRxIPTester-daemon.service; enabled; vendor preset: enabled)
Active: active (running) since Thu 2024-04-18 12:11:29 UTC; 3 days ago
Process: 667 ExecStart=/home/bitpath/source/BP-Receiver/LSRxIPTester (code=exited, status=0/SUCCESS)
Main PID: 669 (LSRxIPTester)
Tasks: 4 (limit: 3933)
CPU: 25min 24.818s
CGroup: /system.slice/LSRxIPTester-daemon.service
└─669 /home/bitpath/source/BP-Receiver/LSRxIPTester
```

- The Redis server status can be verified using the command **“sudo systemctl status redis-server.service”**

```
• redis-server.service - Advanced key-value store
Loaded: loaded (/lib/systemd/system/redis-server.service; enabled; vendor preset: enabled)
Active: active (running) since Thu 2024-04-18 12:11:27 UTC; 3 days ago
Docs: http://redis.io/documentation,
     man:redis-server(1)
Main PID: 561 (redis-server)
Status: "Ready to accept connections"
Tasks: 5 (limit: 3933)
CPU: 21.824s
CGroup: /system.slice/redis-server.service
└─561 /usr/bin/redis-server *:6379
```

## 5 Configuration

The receiver configuration file(bitpath.cfg) is in the BP-Receiver directory. The default configuration for the receiver is as follows,

```
log-level = "err"
frequency = 569000
bandwidth = 6000
plp-id = 0
dst-ip = [127, 0, 0, 1]
dst-port = [0x2C, 0x49]
redis-data-channel = "BitPath-Data"
redis-stats-channel = "BitPath-Stats"
```

The above configuration can be edited as per your requirements. The bitpath.cfg file is monitored by the **LSRxIPTester-daemon-restart.path** daemon file. Therefore, any change saved in the configuration file will automatically restart the **LSRxIPTester-daemon.service**.

**Note:** - Even if the file is saved with space or newline, the script will automatically restart.

## 6 Logging

- The default receiver logging file is in the “**/var/log/DriveTest/**” directory.
- The receiver logger file location can be changed by editing the file “**/etc/rsyslog.d/bitpath.conf**”

```
local6.*    /var/log/DriveTest/bitpath.log    # Edit here for custom location
& stop
```

- Save and close the file.
- If the log file location changes, restart the rsyslog service using the following command: “**sudo systemctl restart rsyslog.service**”.
- The receiver logging can be monitored using the command **tail -f /var/log/DriveTest/bitpath.log | grep -i “Published packets”** or **tail -f <Custom location and filename> | grep -i “Published packets”**

```
***Accepted IPV4s: 17, Long IPV4s: 0, Regular_LLs: 0, Segmented IPV4s: 0, Concatenated IPV4s:
0, Published packets: 18
***Accepted IPV4s: 18, Long IPV4s: 0, Regular_LLs: 0, Segmented IPV4s: 0, Concatenated IPV4s:
0, Published packets: 19
***Accepted IPV4s: 19, Long IPV4s: 0, Regular_LLs: 0, Segmented IPV4s: 0, Concatenated IPV4s:
0, Published packets: 20
***Accepted IPV4s: 20, Long IPV4s: 0, Regular_LLs: 0, Segmented IPV4s: 0, Concatenated IPV4s:
0, Published packets: 21
***Accepted IPV4s: 21, Long IPV4s: 0, Regular_LLs: 0, Segmented IPV4s: 0, Concatenated IPV4s:
0, Published packets: 22
***Accepted IPV4s: 22, Long IPV4s: 0, Regular_LLs: 0, Segmented IPV4s: 0, Concatenated IPV4s:
0, Published packets: 23
```

When it receives data, the receiver publishes the collected packets to the Redis server. Therefore, the published packet values will increase in the log, as shown above.

## 7 Local Redis Client

When the receiver starts publishing data to the Redis server, it can be monitored locally using the Redis client(`redis-cli`, which uses `localhost` as the hostname). The receiver will publish the correction stream to the default channel name **"BitPath-Data"**. The Atsc3 stats data will be published in the default channel name **"BitPath-Stats"**. The channel name should be the same as the one in the `bitpath.cfg` file.

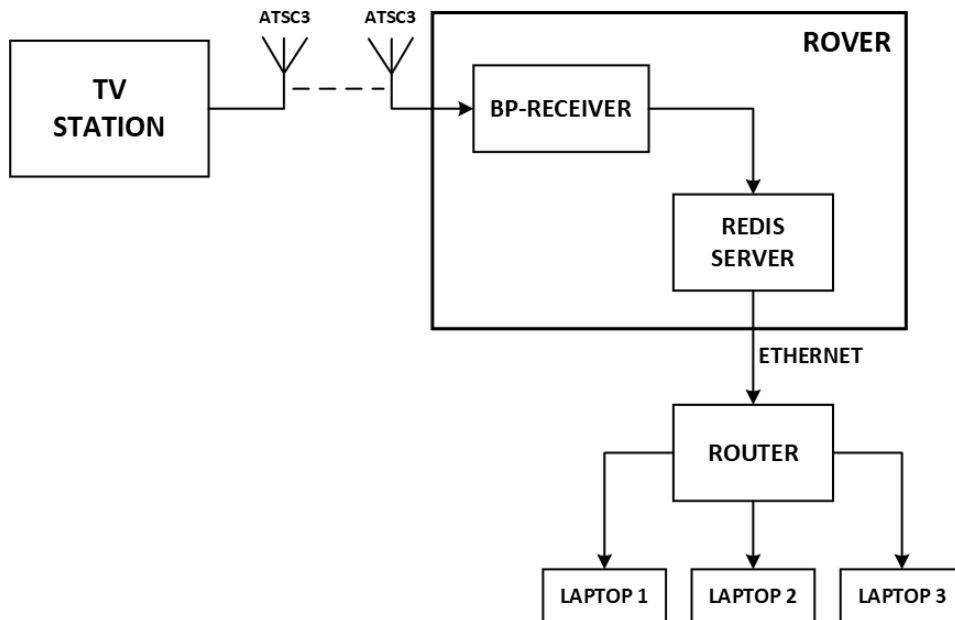
```
bitpath@bitpath-dev-xx:~ $ redis-cli
127.0.0.1:6379> subscribe BitPath-Data BitPath-Stats
Reading messages... (press Ctrl-C to quit)
1) "subscribe"
2) "BitPath-Data"
3) (integer) 1
1) "subscribe"
2) "BitPath-Stats"
3) (integer) 2
```



## 8 Accessing Redis Over the Network

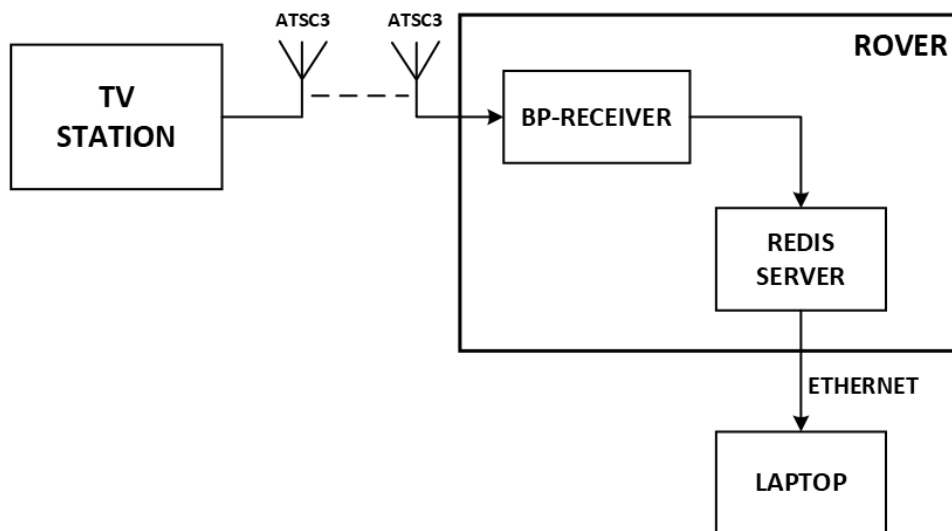
Redis data can be accessed from the network by connecting the Ethernet cable directly between the rover and the laptop or by connecting the device to the local network and accessing the Redis data from multiple devices.

### Step 1. Accessing Redis over the local network



1. Connect the rover to the router using an ethernet cable. Then, find the rover's IP address from the router or open the serial access to the rover using the debug port to obtain its IP address.
2. Access the rover's Redis server from your laptop using the rover's IP address and port number 6379.
3. The connection from your laptop can be made with a Redis client or by writing scripts using Python, c, java, etc.... as per your requirement.
  - a. Redis client can be installed using the following link
    - i. <https://redis.com/redis-enterprise/redis-insight/>
4. Once the connection is established, we can obtain the ATSC3 data transmitted in the data and stats channel mentioned in the bitpath.cfg file.

## Step 2. Accessing Redis using direct ethernet cable



1. To establish a connection between the rover and the laptop/PC, we need to set a static IP address and enable the DHCP server in the rover.
2. Log in to the rover using the debug port and follow the following steps
  - a. Open **/etc/dhcpd.conf** and uncomment the following lines
    - i. interface eth1
    - ii. static ip\_address=10.10.0.1/24
    - iii. static routers=10.10.0.1
    - iv. static domain\_name\_servers=10.10.0.1
  - b. Enable the dnsmasq service using the following command
    - i. `sudo systemctl enable dnsmasq.service`
  - c. `sudo reboot`
3. Connect your laptop or PC to the rover with an ethernet cable once rebooted.
4. Rover will automatically assign your laptop's IP address.
5. Follow the second point in Step 1 to retrieve the data from the rover.

(Optional) If you have changed the static ip\_address field in dhcpd.conf, then you must modify " dhcp-range=10.10.0.50, 10.10.0.150, 12h" in /etc/dnsmasq.conf for the above to work.

## 9 References