**Raspberry Pi Setup Guide**

1. Install Rasperry Pi OS to the Raspberry.
2. Install WiringPi: “apt install wiringpi”
3. Install single\_chan\_pkg\_fwd: “git clone <https://github.com/tftelkamp/single_chan_pkt_fwd>”

Run it to check everything OK between Hat and Rpi (doesn’t say “Unrecognised transceiver”), and take note of the Gateway ID, it will be needed later.

1. [Install ChirpStack software](https://www.chirpstack.io/project/guides/debian-ubuntu/)

**Note**: The guide above is tested for Ubuntu, it had some differences for Raspberry Pi. On the first step of installing dependencies we had problems with PostgreSQL, so we uninstalled it and installed it again with:

**sudo apt install postgresql libpq-dev postgresql-client postgresql-client-common -y**

(line got from [here](https://opensource.com/article/17/10/set-postgres-database-your-raspberry-pi)) and it worked.

Pay attention to the configurations according to the region.

After installing and running the Application Server go to “<http://localhost:8080>” with a browser and the GUI can be seen.

1. In the App Server GUI:
   1. Add a Network Server on “localhost:8000”
   2. Add a Gateway-Profile
   3. Create a Gateway, with the Gateway ID obtained from the hardware
   4. Create an Application
   5. Create a Device Profile. MAC Version 1.0.2, Regional Parameters A, Max EIRP 0.
   6. Create a Device within the Application. Device doesn’t support OTAA.
2. Point single\_chan\_pkg\_fwd to the Network Server:

In main.cpp change SERVER1 to “127.0.0.1:8003”. Run “sudo ./single\_chan\_pkg\_fwd” and check Gateway Status in the GUI, “Last Seen” should be “seconds ago” (Refresh the page).

1. Check messages: Within the “Gateway” and “Applications/Device” tabs, go to “LoRaWAN Frames” and “Device Data” and check if frames are arriving. The original data can’t be seen here out of the box because it’s shown encrypted.
2. Show Decoded message: Go to the Device-Profile “Codec” tab and use this code:

function Decode(fPort, bytes, variables) {

var data = { "cmd": "" };

for (var i = 0; i<bytes.length; i++) {

data.cmd+=String.fromCharCode(bytes[i]);

}

return data;

}