

RPi: Streaming raspberry shake data to node-red

Goal:

Learn how to send seismology data from a raspberry shake into a separate raspberry pi with node-red. And how to use the data once it arrives.

Resources:

Raspberry shake, setup following this guide: <https://t3alliance.org/rpi-setting-up-a-raspberry-shake/>

A separate raspberry pi with node-red, running on the same network

Introduction:

Raspberry shake usually processes its own data, and publishes it to its own public databases. However it is possible to configure raspberry shake to send a raw data-stream from itself to any device your network can reach. Raspberry shake uses a UDP stream to send data from itself to any list of ip addresses and ports that happen to be listening. Using this raw data stream you can create your own graphs, or set up alerts and triggers in node-red.

This means that the raspberry shake could be physically located anywhere in a building, or using the proper networking setup anywhere in the world and you can still comfortably

receive the data in node-red on your own computer. Imagine an email alert being sent-out by your RPi at home, because your RPi shake at school detected some unusual vibrations. Or you could trigger a camera to go off, and email you a picture whenever footsteps were detected in your room, the possibilities are endless!

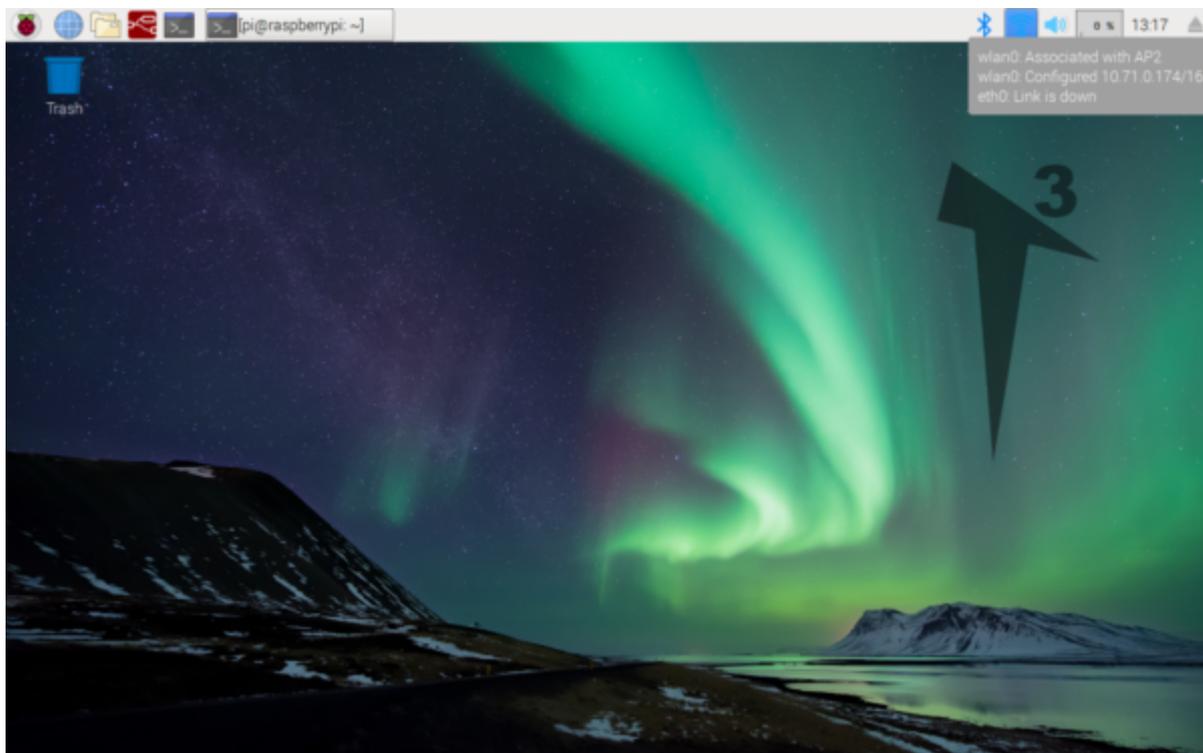
What is UDP?

When stuff is sent over the internet it is sent using a protocol, which is a way of organizing data so that both parties know what's going on. If you made an agreement with a friend to always use red envelopes when a message was urgent, that would be an example of a protocol. Most of the time when you download a file, or send an email it is sent using TCP protocol; which is rather strict. For example, if something messes up and the TCP packets (analogous to envelopes with letters inside) arrive out of order, they will be reorganized and only sent to the destination when everything is sorted out. UDP is much less strict, and is more like a raw flow of data vs an orderly sequence of letters. If UDP packets arrive out of order, or have some missing, or even if the data inside is messed up they'll usually be sent straight along to the destination. This is useful in projects like our seismometer because it is more lightweight with less latency, also it comes with a feature called multicast which allows a data stream to be sent to multiple destinations at the same time; which is exactly what is possible with the raspberry shake.

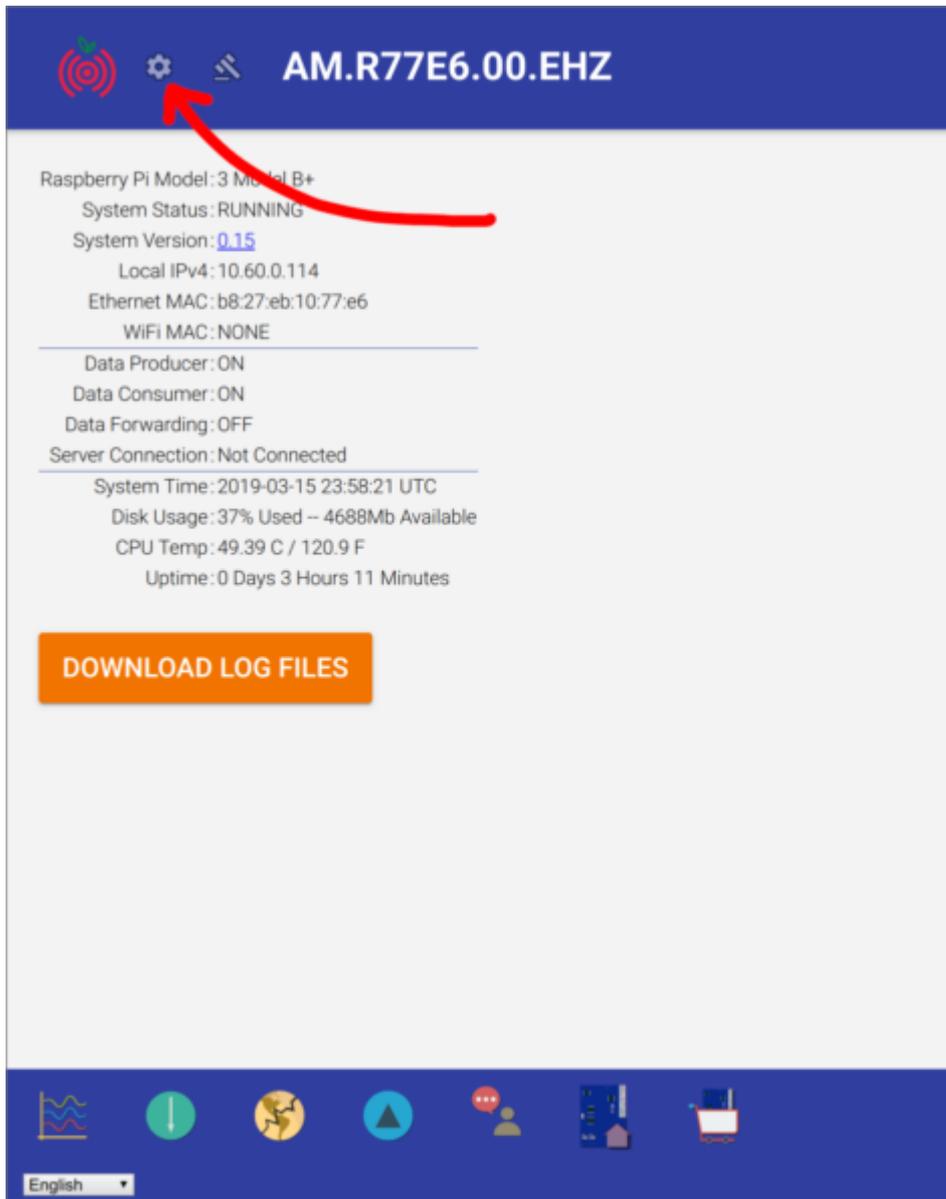
Getting Started:

The first thing you'll need is the IP address of the pi that will listen for the data, this can be found just by hovering over the network applet here. The easiest way to ensure the listening pi and the shake-pi are on the same network is to

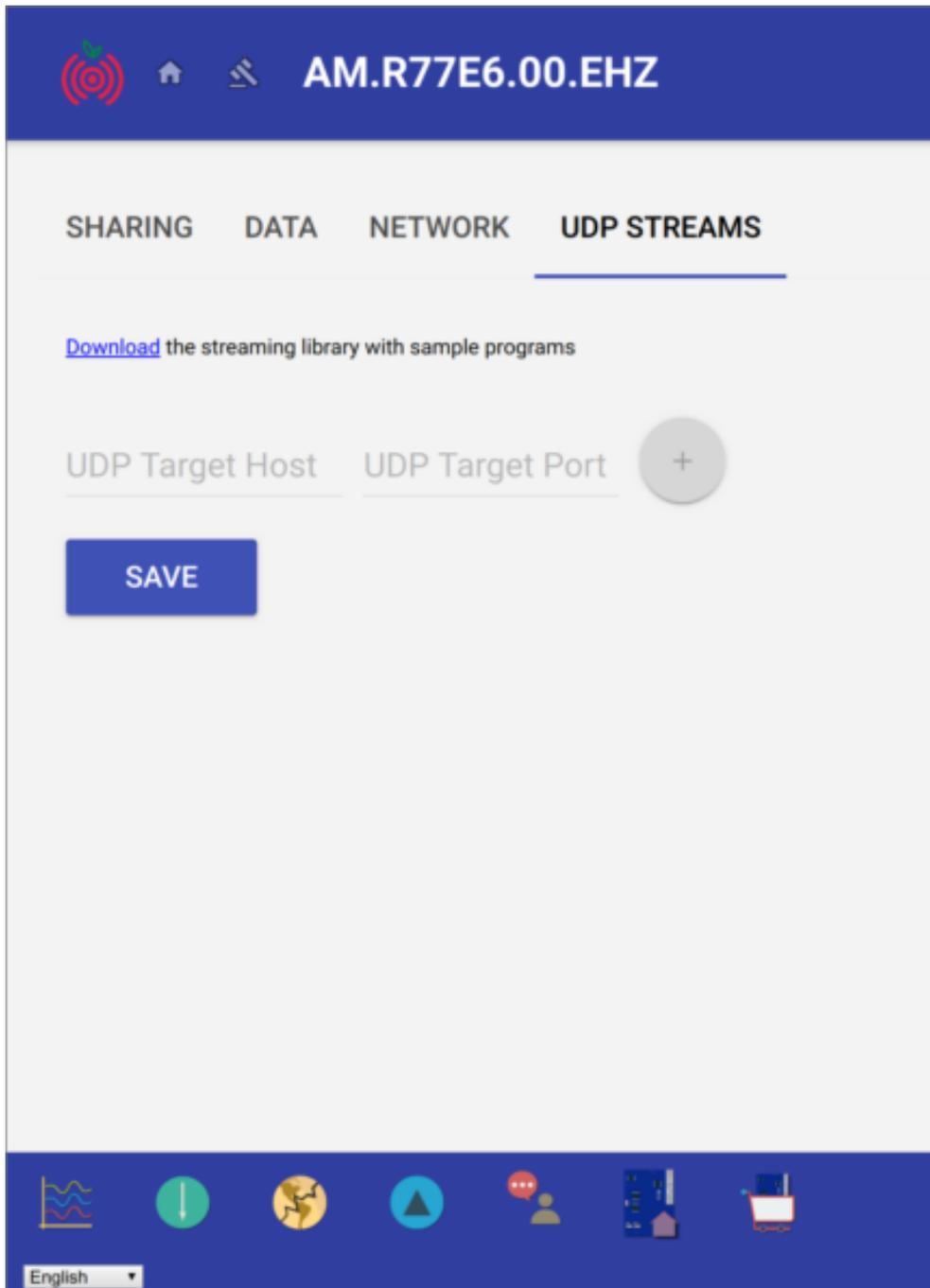
plug them both into the same router using ethernet cables. Otherwise setting up a WiFi connection on the shake can be tricky and even interfere with the measurements.



Connecting to the rs.local or raspberrysshake.local or even the `ip_address:80` of the shake,



From the settings page, come to the UDP streams tab



Here is where you'll need the ip address of your listening pi, it's also important to make sure it's connected to the same network as the shake. If the shake is using LAN this means you'll probably have to plug your pi into LAN to.

Here I've entered the IP of my listening pi, and used the default port 8888

[Download](#) the streaming library with sample programs

UDP Target Host	UDP Target Port	
10.70.0.170	8888	



Click the plus button and don't forget to hit save.

Reading the stream on raspberry pi:

Starting with a fresh flow, we'll drag the UDP IN node in.

Q ud ×

Air Quality OLED flow

input

- udp

output

- udp
- play audio

function

- rshake udp parser

social

storage

analysis

advanced

Raspberry Pi

network

performance

Air Quality

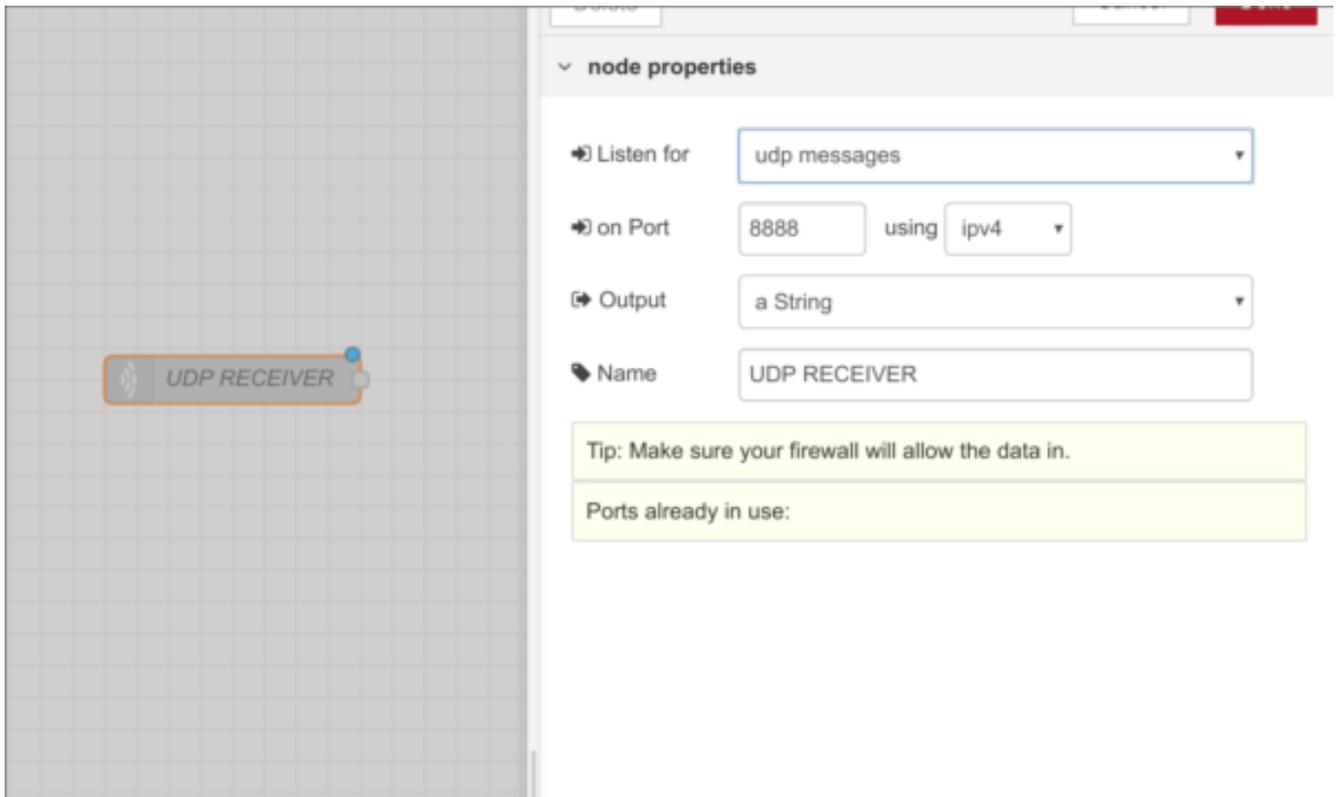
LED Matrix

oled

dashboard

- audio out





For the configuration chose the same port as we used in the `rs.local` config, that would be 8888, and make sure to set output to 'a String' instead of 'Buffer'.

If you hook your UDP RECIEVER up to a debug node you should already be able to see a stream of data coming in, if not something is wrong with your connection (probably raspbri isn't on the same network as raspberry shake).

The image shows a network traffic analysis tool interface. On the left, a diagram shows a 'UDP RECEIVER' node connected to a 'msg.payload' node. On the right, a log window displays three messages received from node '70fdab51.308ec4' on 3/15/2019 at 2:33:55 PM. Each message contains a JSON payload with a list of integers. The payloads are:

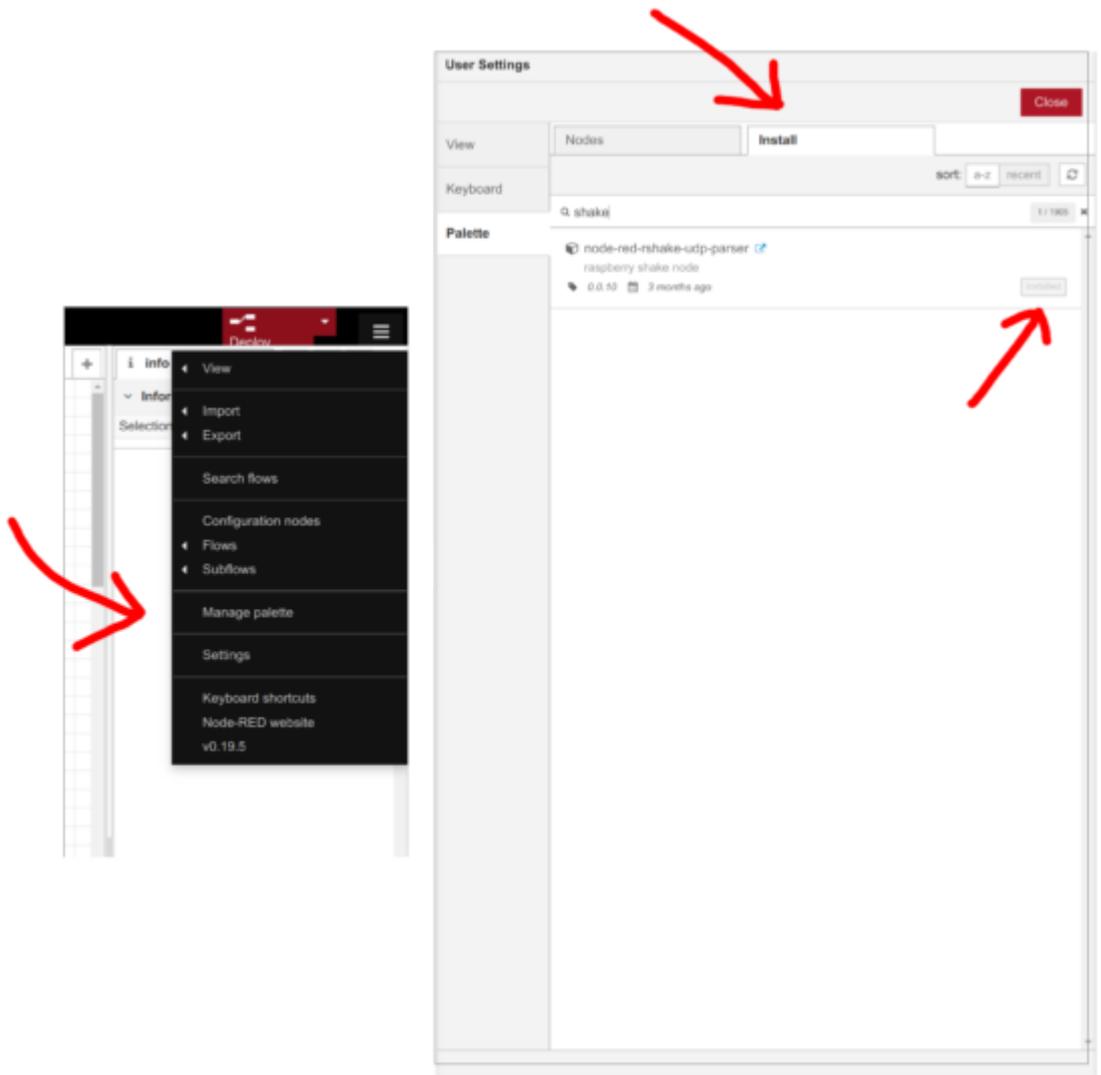
1. `"{'EHZ', 1552696434.655, -14580, 42816, 131235, 167617, 60780, -128644, -165232, -36373, 26909, 58932, 139837, 82898, -73785, -88954, -30154, -25852, 13959, 77381, 99978, 103123, 11649, -131345, -152776, -79625, -466}"`

2. `"{'EHZ', 1552696434.905, 70651, 129702, 154002, 51164, -180254, -367988, -445724, -484727, -226238, 351090, 660287, 621011, 522435, 413244, 149776, -327166, -545456, -222183, 177778, 186919, 75319, 87614, -88418, -215859, -65782}"`

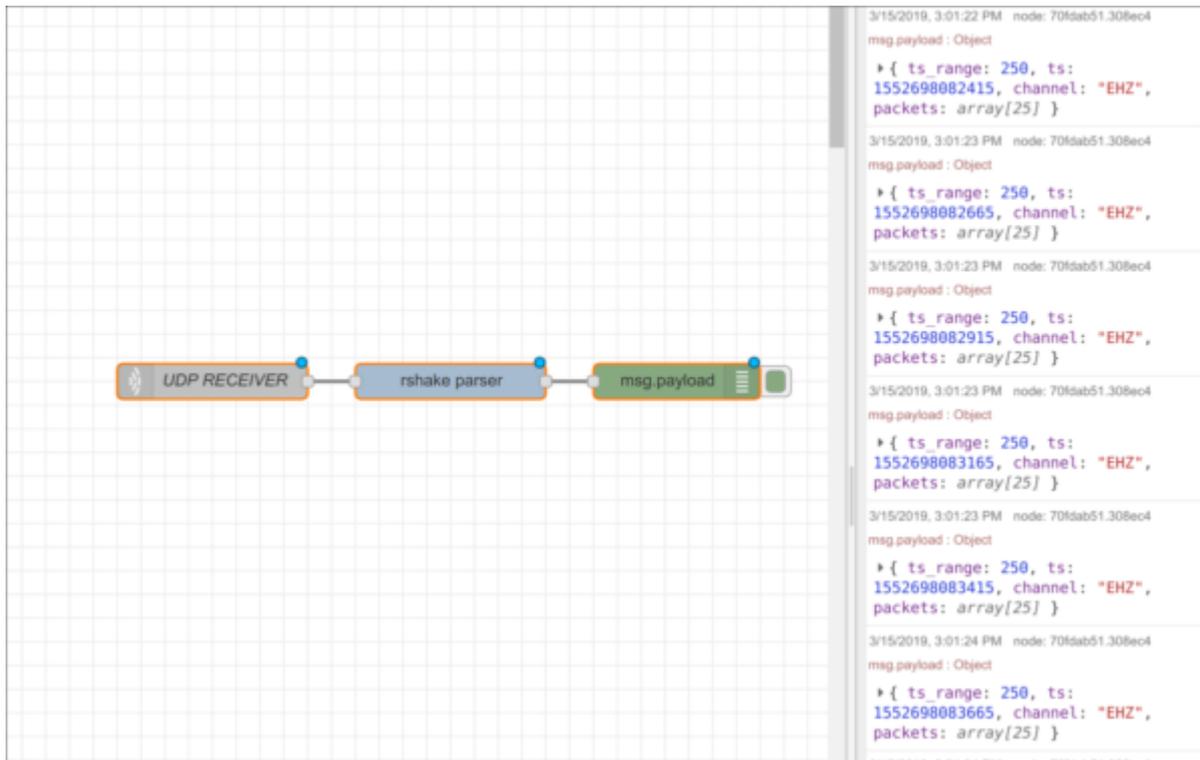
3. `"{'EHZ', 1552696435.155, 56494, 116990, 90346, 3566, 15875, -56984, -240420, -123412, 203951, 274149, 104367, -92738, -218034, -217540, -128390, 26699, 211177}"`

Next we need the raspberry shake parser node, made by the guys who build the SHAKE.

This can be installed by going into the palette manager, as of now it's the only result when you search 'shake'.



The installation will take a few minutes, and then you can chain it with the UDP node like so.



All the rshake parser node does is take the raw thing outputted by the UDP node, and turn it into an object with various data sorted into properties. Using a a debug node we can examine the output. For example, the 'channel' property allows you to discriminate between different sensors on the shake. We are using a raspberry-shake 'ID' so we only get one channel 'EHZ'.

Now that we have received the data through the UDP ports, we can start to use the data. The data can be used as any other variable data sensor data, and to practice using the data follow the links below.

To make a real-time, Dashboard Seismogram: <https://t3alliance.org/rpi-building-a-raspberry-shake-applausemeter-in-node-red/>

To make a Shakermeter: <https://t3alliance.org/rpi-building-a-raspberry-shake-seismogram-in-node-red>

