Raspberry Shake

Computer Science with RS

Lesson #3

Why is a seismograph related to Computer Science?

- The Raspberry Shake *is* a computer
- But not a computer that you are used to…

Watch the next video to learn more!

Watch: What makes a computer a computer?

Raspberry Shake

The Raspberry Shake has those main components

- **Input** - Geophone and Digitizer
- **Storage** - Raspberry Pi memory
- **Process** - Raspberry Shake OS (on Raspberry Pi)
- **Output** - The processed data goes out over the internet to the RS network servers

Now let’s try coding with the Raspberry Shake!

Get out computers (or Raspberry Pis) and start Node Red! Follow the tutorial on:

Edu.raspberryshake.org/classroom-curriculum/node-red

Raspberry Shake

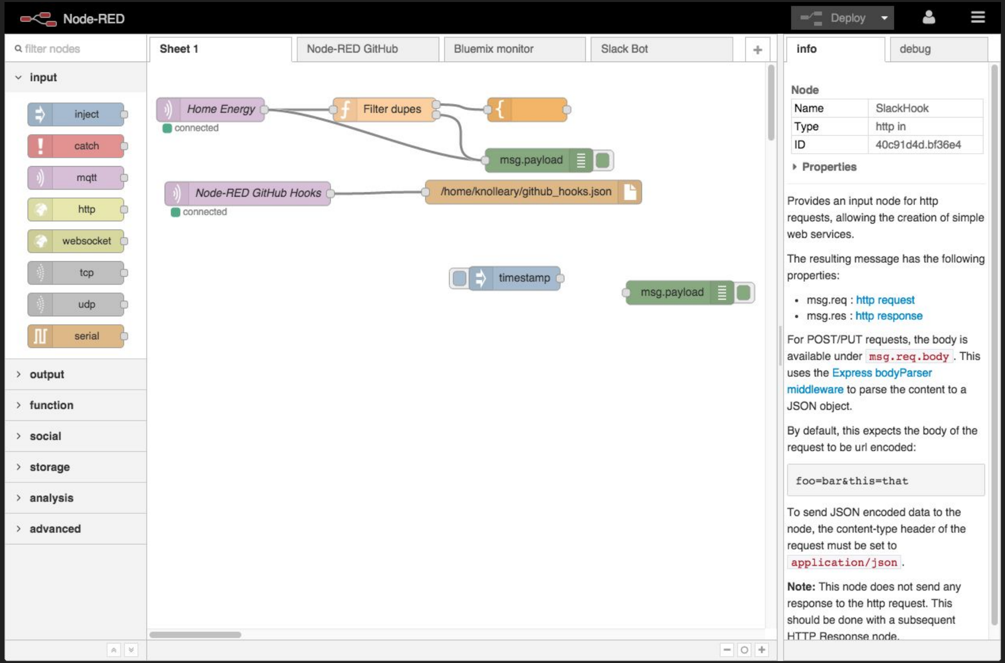
Connecting your RShake to Node-Red

Node-RED

Raspberry Shake

What is Node-Red?

- Visual, flow based programming language
- Browser editable
- Javascript Powered
- Perfect for Education
- Varied Applications



Node

Name	SlackHook
Type	http in
ID	40c91d4d.bf36e4

Properties

Provides an input node for http requests, allowing the creation of simple web services.

The resulting message has the following properties:

- msg.req : http request
- msg.res : http response

For POST/PUT requests, the body is available under `msg.req.body`. This uses the Express bodyParser middleware to parse the content to a JSON object.

By default, this expects the body of the request to be url encoded:

```
foo=bar&this=that
```

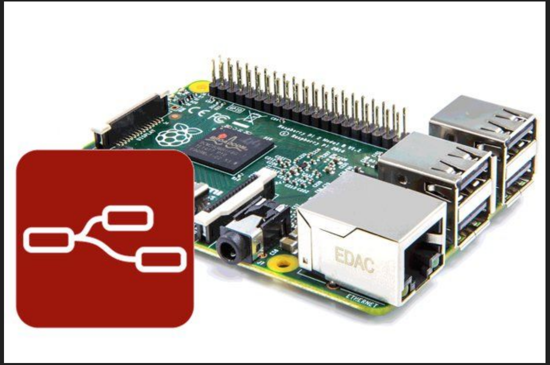
To send JSON encoded data to the node, the content-type header of the request must be set to `application/json`.

Note: This node does not send any response to the http request. This should be done with a subsequent HTTP Response node.

Start with a Separate Device that has Node Red

Most probably this will be another raspberry pi running Raspbian with Node Red enabled.

You can run a Node Red server on your laptop or desktop if another raspberry pi is not available. The process is a bit more difficult, because you first have to download node.js. Instructions can be found here: https://nodered.org/docs/getting-started/installation



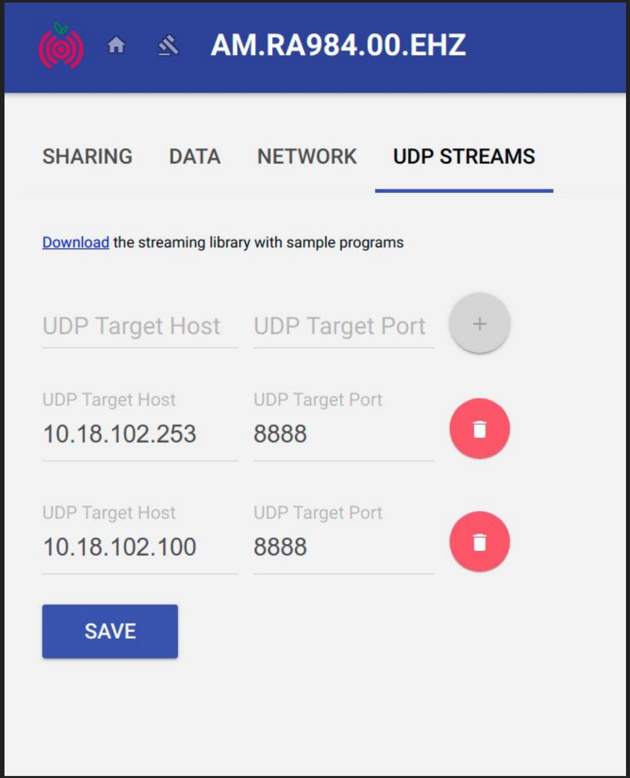
Raspberry Shake

RShake and UDP data forwarding

UDP (User Datagram Protocol) is a protocol for sending data over LAN networks.

The Raspberry Shake can use UDP to forward its raw seismic data to any number of IP addresses and UDP ports

Node Red can serve as a receiver for that data, and can then use that data as variable input data.



Download the streaming library with sample programs

UDP Target Host	UDP Target Port
10.18.102.253	8888
10.18.102.100	8888

Configuring the Raspberry Shake for UDP Stream

Open your Raspberry Shake's "front end" control page on your browser. Click the settings icon ⚙ next to the logo, and then click on the UDP Streams tab on the right.

Where it says UDP Target Host enter the IP address of the device that is running Node Red. Then in the UDP Target Port enter the port number, in our case: 8888.

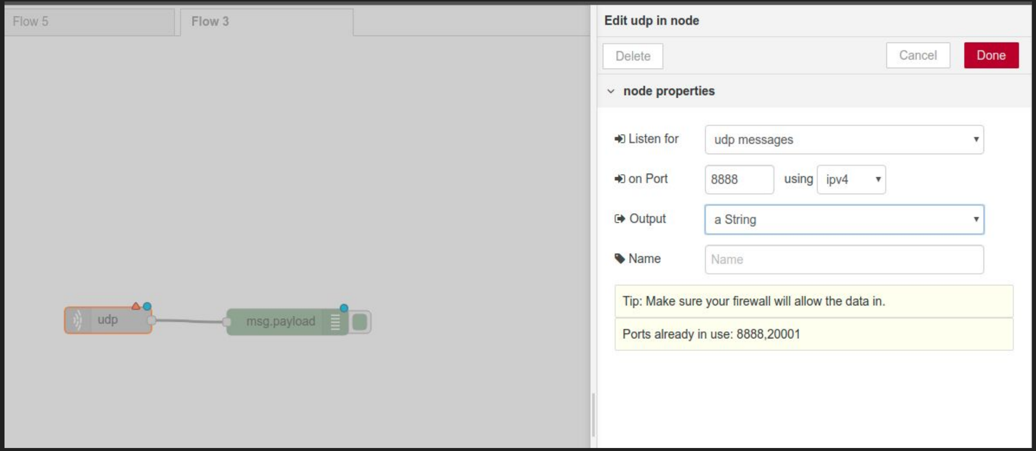
Then click the plus button and then save at the bottom.

Setting up Node Red to Receive UDP Data

Once you have started the node red server and opened Node-Red in your browser, locate the “udp” input node.

Configure the Node to receive data on port 8888 and change output to a string, not a buffer. Connect a debug node to the udp node.

Deploy.



Listen for: udp messages
on Port: 8888 using ipv4
Output: a String
Name: Name

Tip: Make sure your firewall will allow the data in.

Ports already in use: 8888,20001

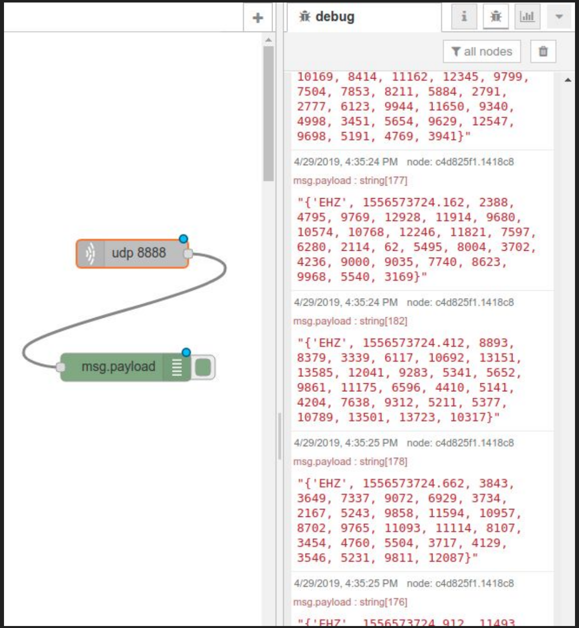
Receiving the Data

Go to the Node Red panel and click on the debug tab in the upper right corner

You should see the data stream, as shown in the image.

If not, then check all steps were done correctly:

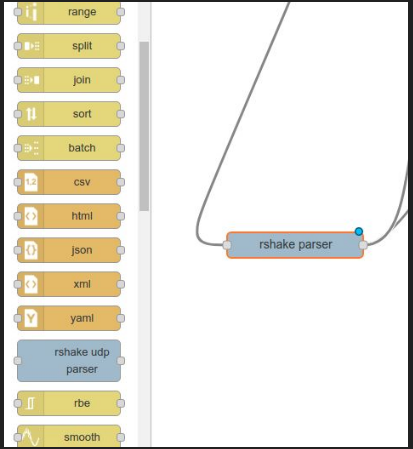
Do the UDP ports match? Is the UDP output a string? Is UDP host the IP address correct? Did you deploy? Did you press SAVE udp stream on the Shake settings?

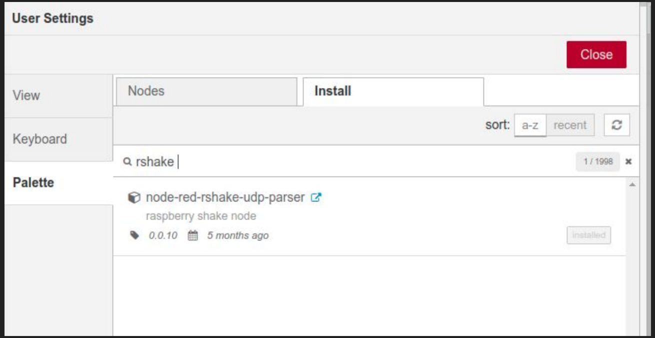


```
10169, 8414, 11162, 12345, 9799,
7504, 7853, 8211, 5884, 2791,
2777, 6123, 9944, 11650, 9340,
4998, 3451, 5654, 9629, 12547,
9698, 5191, 4769, 3941}"

4/29/2019, 4:35:24 PM   node: c4d825f1.1418c8
msg.payload : string[177]
"{'EHZ', 1556573724.162, 2388,
4795, 9769, 12928, 11914, 9680,
10574, 10768, 12246, 11821, 7597,
6280, 2114, 62, 5495, 8004, 3702,
4236, 9000, 9035, 7740, 8623,
9968, 5540, 3169}"

4/29/2019, 4:35:24 PM   node: c4d825f1.1418c8
msg.payload : string[182]
"{'EHZ', 1556573724.412, 8893,
8379, 3339, 6117, 10692, 13151,
13585, 12041, 9283, 5341, 5652,
9861, 11175, 6596, 4410, 5141,
4204, 7638, 9312, 5211, 5377,
10789, 13501, 13723, 10317}"

4/29/2019, 4:35:25 PM   node: c4d825f1.1418c8
msg.payload : string[178]
"{'EHZ', 1556573724.662, 3843,
3649, 7337, 9072, 6929, 3734,
2167, 5243, 9858, 11594, 10957,
8702, 9765, 11093, 11114, 8107,
3454, 4760, 5504, 3717, 4129,
3546, 5231, 9811, 12087}"

4/29/2019, 4:35:25 PM   node: c4d825f1.1418c8
msg.payload : string[176]
"{'EHZ', 1556573724.912, 11493
```

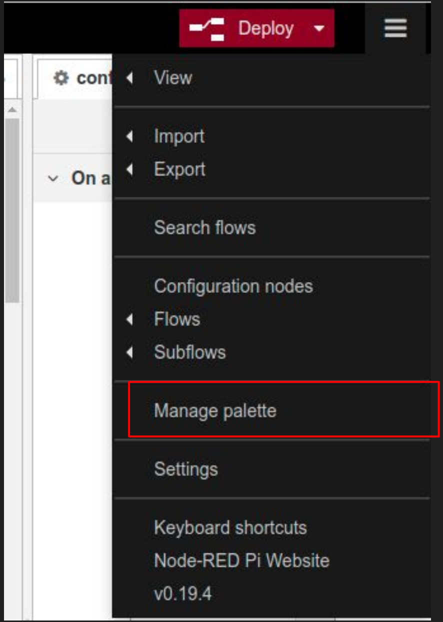
Processing the Data in Node Red

The Shake receives packets of raw data that must be processed before becoming usable input. Go to "manage palette" on your node red menu and download the "rshake udp parser" node.





node-red-rshake-udp-parser
raspberry shake node
0.0.10 5 months ago



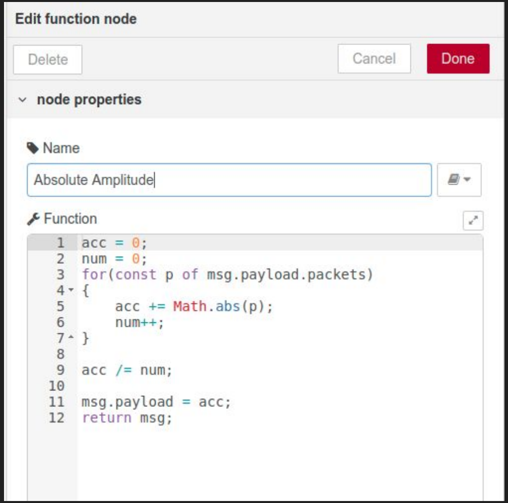
Raspberry Shake

Using the Data in Node Red

After being processed by the parser node, the data is still in a data packet. To convert these packets to easily-used integers, it must pass through another function node.

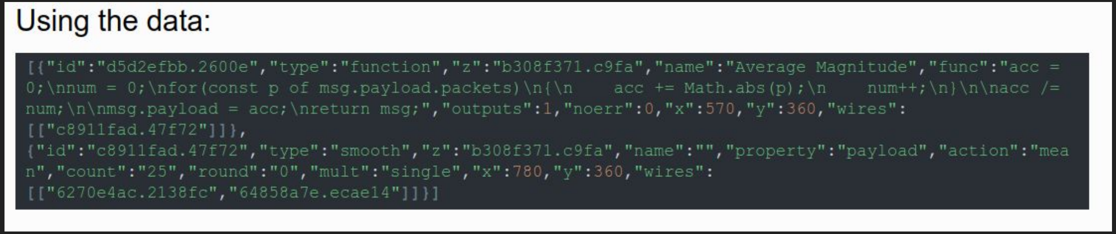
T3 alliance has created a function node for this purpose. It can be found online on T3alliance.org Raspberry Shake tutorials. Copy and paste code into “import from clipboard”

https://t3alliance.org/rpi-streaming-raspberry-shake-data-to-node-red/



Absolute Amplitude

```
acc = 0;
num = 0;
for(const p of msg.payload.packets)
{
    acc += Math.abs(p);
    num++;
}

acc /= num;

msg.payload = acc;
return msg;
```

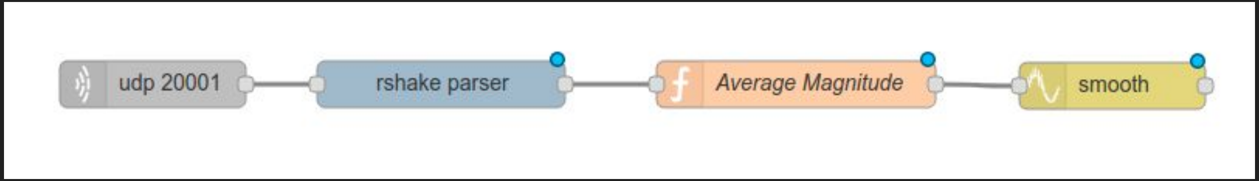


Using the data:

```
[{"id":"d5d2efbb.2600e","type":"function","z":"b308f371.c9fa","name":"Average Magnitude","func":"acc =
0;\nnum = 0;\nfor(const p of msg.payload.packets)\n{\n    acc += Math.abs(p);\n    num++;\n}\n\nacc /=
num;\n\nmsg.payload = acc;\nreturn msg;","outputs":1,"noerr":0,"x":570,"y":360,"wires":
[["c8911fad.47f72"]]},
{"id":"c8911fad.47f72","type":"smooth","z":"b308f371.c9fa","name":"","property":"payload","action":"mea
n","count":"25","round":"0","mult":"single","x":780,"y":360,"wires":
[["6270e4ac.2138fc","64858a7e.ecae14"]]}]
```

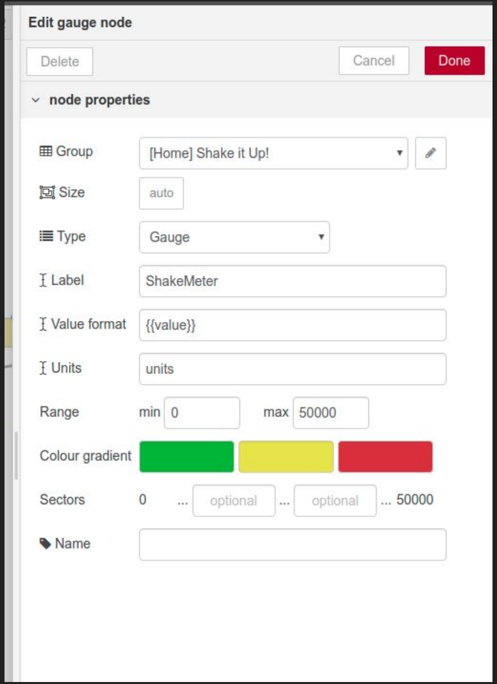
Quantifying the seismicity using the Dashboard

We should now have a flow that looks like this:



Now, we can start to display the data through the Dashboard. We can start with a gauge. Put 100000 in the max range field.

Open a new browser and navigate to http://"yourIPaddress:1880/ui


Edit gauge node

Delete | Cancel | Done

node properties

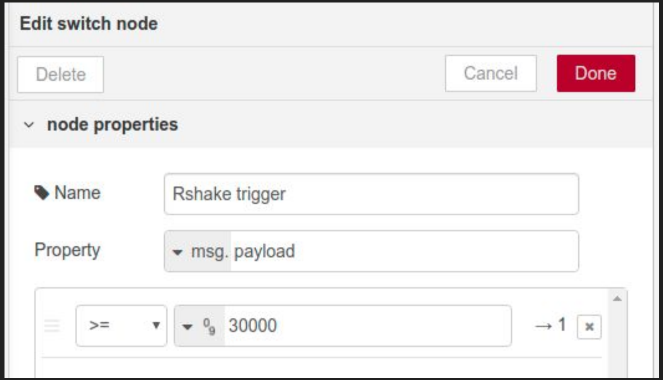
Group: [Home] Shake it Up!
Size: auto
Type: Gauge
Label: ShakeMeter
Value format: {{value}}
Units: units
Range: min 0 max 50000
Colour gradient: 🟩 🟨 🟥
Sectors: 0 ... optional ... optional ... 50000
Name:

Triggering an action

Now we can add a trigger, an action that happens when the amplitude reaches a certain value. To do this, we need two more nodes. A switch node and a change node.

After the smooth node, add a switch node and configure it so that anything >= 30000 goes forward.

We will make a notification that says “crazy high!” as soon as the value goes above 30000



Name: Rshake trigger

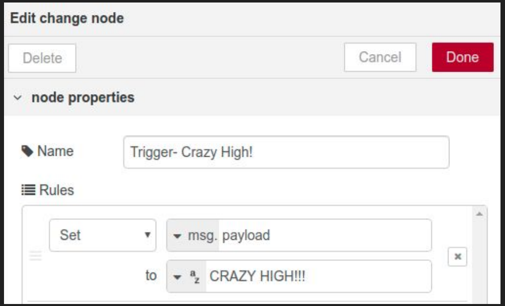
Property: msg. payload

>= 30000 → 1

Raspberry Shake

Triggering an action

Then add a Change node, and set the msg.payload to change to a string that says “Crazy High!”


Edit change node

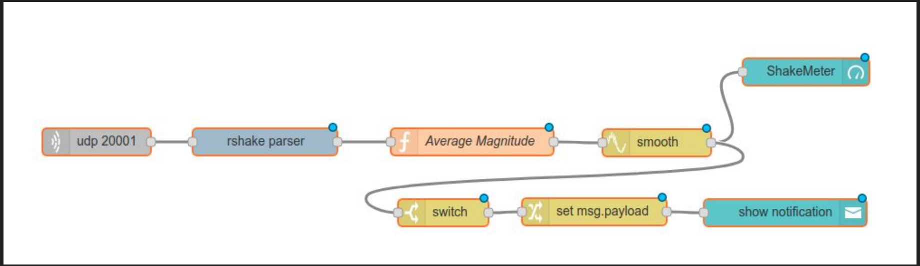
Name: Trigger- Crazy High!

Rules: Set msg. payload to CRAZY HIGH!!!

Then, add the “notification” dashboard node.

Deploy!

Your flow should look similar to this below:


ShakeMeter
udp 20001
rshake parser
Average Magnitude
smooth
switch
set msg.payload
show notification