

1. Apéndice H. Configuración Nodos Tiempo UDP y Tiempo TCP

Figura 1

Nodo que configura el tiempo.



Nota. Este nodo recibe datos continuamente y los acumula en un buffer hasta que se alcanza el umbral de 1000 datos. Si el umbral se alcanza, se limpia el buffer y se actualiza el estado del nodo.

Si no se alcanza el umbral, se siguen añadiendo datos al buffer.

Figura 2

Configuración del nodo Control

```

1  // Establecemos un umbral de 1000 datos
2  var threshold = 1000; // Puedes ajustarlo según sea necesario
3  var accumulatedData = flow.get('accumulatedData') || []; // Obtenemos los datos almacenados
4
5  // Verificamos si el número de datos ha alcanzado el umbral
6  if (accumulatedData.length >= threshold) {
7      // Limpiar los datos si se ha alcanzado el umbral
8      flow.set('accumulatedData', []); // Limpiar el buffer
9      node.status({fill: "green", shape: "dot", text: "Buffer vacío - Datos procesados"});
10     // Aquí podrías agregar un paso para almacenar los datos antes de vaciar el buffer
11     //si lo deseas (guardar en archivo, base de datos, etc.)
12 } else {
13     // Si no se alcanzó el umbral, añadimos los nuevos datos
14     accumulatedData.push(msg.payload);
15     flow.set('accumulatedData', accumulatedData); // Guardamos los datos actualizados
16     node.status({fill: "blue", shape: "dot", text: "Datos añadidos al buffer"});
17 }
18
19 // Devuelve el mensaje para continuar el flujo
20 return msg;
21

```

Nota. Este nodo establece un umbral de 1000 datos acumulados y, cuando se alcanza este umbral, limpia el buffer de datos. Si no se alcanza el umbral, continúa acumulando los datos y mantiene el estado del nodo actualizado para reflejar la acción tomada.

1.1 Datos UDP

Este nodo toma los datos de UDP, los convierte a float32, los organiza y parsea, y genera un mensaje (msg.payload) con todas las propiedades necesarias para el template en la interfaz de usuario de Node-RED.

Figura 3

Configuración del nodo Datos UDP parte A.

```

1 // Function "Parse GNURadio UDP → UI Payload"
2 ///////////////////////////////////////////////////////////////////
3 // 1) Acumular bytes en contexto UDP
4 ///////////////////////////////////////////////////////////////////
5 let chunk = msg.payload; // Buffer entrante
6 let bufBytes = flow.get('bufBytesUDP') || Buffer.alloc(0);
7 bufBytes = Buffer.concat([bufBytes, chunk]);
8 let totalB = bufBytes.length - (bufBytes.length % 4);
9 let bufRead = bufBytes.slice(0, totalB);
10 let bufLeft = bufBytes.slice(totalB);
11 flow.set('bufBytesUDP', bufLeft);
12
13 // 2) Convertir a floats LE y acumular
14 ///////////////////////////////////////////////////////////////////
15 let newFloats = [];
16 for (let i = 0; i < bufRead.length; i += 4) {
17   newFloats.push(bufRead.readFloatLE(i));
18 }
19 let floats = (flow.get('bufFloatsUDP') || []).concat(newFloats);
20
21 // 3) Bucle de parsing: espectro seguido de constantes
22 ///////////////////////////////////////////////////////////////////
23 let outMsgs = [];
24 while (true) {
25   let iS = floats.indexOf(-1.0); // Inicio espectro
26   let iE = (iS >= 0) ? floats.indexOf(-2.0, iS + 1) : -1; // Fin espectro
27   if (iS < 0 || iE < 0) break;
28
29   let iC = floats.indexOf(-3.0, iE + 1); // Inicio constantes
30   let iCE = (iC >= 0) ? floats.indexOf(-4.0, iC + 1) : -1; // Fin constantes
31   if (iC < 0 || iCE < 0) break;
32
33   // Extraer espectro y constantes
34   let finicial = floats[iS + 1];
35   let fpaso = floats[iS + 2];
36   let spec = floats.slice(iS + 4, iE - 1);
37
38   let c = floats.slice(iC + 1, iCE);
39   let lat = c[0], lon = c[1], salto = c[2], Azimut = c[3], Elevacion = c[4], Altitud = c[5];
40   let descCodes = c.slice(6);
41   let Descripcion = String.fromCharCode(...descCodes);
42

```

Nota. Esta imagen representa la configuración parte A del nodo que procesa los datos UDP recibidos.

Figura 4

Configuración del nodo Datos UDP parte B.

```
42
43 // Generar msg para ui_template
44 let Fecha = new Date().toLocaleString("es-CO", { timeZone: "America/Bogota" });
45 let out = {
46     payload: {
47         Fecha: Fecha,
48         latitud: lat,
49         longitud: lon,
50         Azimut: Azimut,
51         Elevacion: Elevacion,
52         Altitud: Altitud,
53         Descripcion: Descripcion,
54         finicial: finicial,
55         fpaso: fpaso,
56         N: spec.length,
57         Datos: spec.join(',')
58     },
59     topic: 'udp'
60 };
61 outMsgs.push(out);
62
63 // Limpiar floats ya procesados
64 floats = floats.slice(ICE + 1);
65 }
66
67 // Guardar resto de floats
68 flow.set('bufFloatsUDP', floats);
69
70 // Devolver mensajes
71 return outMsgs;
72
```

Nota. Este nodo procesa los datos UDP recibidos, acumulando bytes y convirtiéndolos en valores de tipo float32. Luego, parsea los paquetes de espectro y constantes, generando el formato adecuado para ser mostrado en el ui_template.

1.2. Tiempo UDP

Este código obtiene la fecha y hora actual, la convierte al formato local colombiano (es-CO) y la asigna al msg.payload para que pueda ser utilizado en otros nodos conectados.

Figura 5

Configuración del nodo Tiempo UDP

```

1  msg.payload = new Date().toLocaleString('es-CO', { hour12: false });
2  return msg;
3  |

```

Nota. Este nodo toma la hora actual en formato local de Colombia y la asigna al mensaje para su posterior visualización en el sistema.

2. TCP Paralelo

Este código de Node-RED permite visualizar en un gráfico el espectro de la señal TCP y actualizar la hora de la última actualización recibida.

Figura 6**Configuración del nodo TCP Paralelo**

```

1  <h2 style="text-align:center; color: red; font-weight: bold;">
2  | ESPECTRO TCP
3  </h2>
4  <p id="hora-tcp-paralelo" style="text-align:center;font-weight:bold;color:red;margin:4px 0;">Última TCP: ---:---:---</p>
5
6  <div id="chart" style="width: 100%; height: 500px;"></div>
7
8  <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
9  <script>
10
11  (function(scope){
12    let last=null;
13    const lbl=document.getElementById('hora-tcp-paralelo');
14    function setTime(t){ if(lbl) lbl.textContent='Última TCP: '+t; }
15    const saved=localStorage.getItem('hora-tcp-paralelo'); if(saved) setTime(saved);
16    function draw(d){
17      if(!d||!d.x||!d.y) return;
18      last=d;
19      Plotly.newPlot('chart',[[{x:d.x,y:d.y,type:'scatter',mode:'lines',line:{color:'red'}}]],{autosize:true,margin:{t:40,r:40,b:40,l:40},xaxis:{title:'Frecuencia (GHz)',yaxis:{title:'dB'}}),(responsive:true));
20    }
21    scope.$watch('msg',function(msg){
22      if(msg) return;
23      if(msg.topic==='ui_control')(setInterval(()=>{Plotly.Plots.resize(document.getElementById('chart'));},300);return;}
24      if(msg.payload.x && msg.payload.y){draw(msg.payload);
25        else if(typeof msg.payload==='string'){localStorage.setItem('hora-tcp-paralelo',msg.payload);setTime(msg.payload);}
26      });
27    })(scope);
28  </script>

```

Nota. Este nodo visualiza el espectro y la hora más reciente de la conexión TCP.

3. Última Actualización TCP**Figura 7**

Configuración del nodo Última Actualización TCP

The screenshot shows the 'Edit text node' configuration window. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below these is a 'Properties' section with a gear icon and three sub-sections: 'Group', 'Size', and 'Label'. The 'Group' section has a dropdown menu set to '[Inicio] Navegación' and a '+' button. The 'Size' section has a text box with '6 x 3'. The 'Label' section has a text box with 'Última actualización TCP'. Below these is a 'Value format' section with a text box containing '{{msg.payload}}'. The 'Layout' section shows a visual representation of the node layout with five boxes labeled 'label value'. The 'Style' section has a checkbox for 'Apply Style'. The 'Class' section has a text box with 'Optional CSS class name(s) for widget'. The 'Name' section has a text box with 'Última actualización TCP'.

Nota. Este nodo muestra la hora más reciente de la conexión TCP.

Figura 8

Último registro de datos ingresados.

UDP Paralelo	TCP Paralelo
<div>VOLVER AL INICIO</div> <div>CLEAR</div> <div>ESPECTRO UDP</div> <div>Última UDP: 29/5/2025, 15:15:36</div>	<div>VOLVER AL INICIO</div> <div>CLEAR</div> <div>ESPECTRO TCP</div> <div>Última TCP: 29/5/2025, 15:15:36</div>

Nota. Este nodo muestra la hora más reciente de la conexión TCP/UDP.

De esta manera se ve el tiempo del registro en ambos protocolos y deteniéndose cuando no se ingresen datos nuevos.