

# How to add Custom Decoders and Sensor Definitions

## Introduction

By default, the system supports MultiTech/Radio Bridge, ADEUNIS and ELSYS LoRaWAN sensors. The sensor definitions, a **sensor definition JSON file,** and a **sensor decoder file**, are embedded into the firmware and cannot be modified or deleted by the user.

The system also allows importing sensor definitions, which allows adding new sensors or importing a custom sensor definition. This guide will instruct you on the process.

## Requirement

- MultiTech Gateway with Payload Management License with mPower 6.3.2+
- LoRaWAN sensor with javascript decoder.
  - A Conduit gateway with mPower OS v6.3.2 or higher that has been optimized to use the standard of TTN JavaScript decoders. You can check out their sensor library here https://github.com/TheThingsNetwork/lorawan-devices

# **Downloading Decoder from TTN Repository**

If your sensor is included in the sensor repository you can download it from the TTN GitHub and use that decoder to create a sensor definition file.

- 1. In the link above, navigate the repository to find your sensor manufacturer and model and download the file.
- 2. MultiTech/Radio Bridge Sensor decoders are pre-installed on our devices, if you'd like to modify it, they can be found here: <u>lorawan-devices/vendor/radio-bridge/radio\_bridge\_packet\_decoder.js at master</u> <u>• TheThingsNetwork/lorawan-devices GitHub</u>

luct ~ Solutions ~ Open	Source > - Pricing				
ningsNetwork / lorawar	n-devices (Fullic)			( Q. Notifications ) ( Y Fork	r (347) ) (12 Star (17
🗇 lasues 🙀 🏦 Pul re	cuests 12 💿 Actions 🛈 Security 🗠 Insights				
	2' muster - 2' 16 Branches 🛇 0 Lags	Q. Ga to film	C Code *	About	
	🛞 det Innovation GentiH and Jaime Trinidad :	alord on calcilor thermodal (in L. 📼 🖌 aletta-tal seek	(3) SEA Committe	Device Repository for LoRaWAN devices	
	🖿 .github	Upclate PULL_REQUEST_TEMPLATE/rid (#725)	4 months ago	Readme     Ar Activity	
	in bin	Update validation is (and comments) (#725)	3 months ago	Custom properties	
	i doc	Operate dominientation and examples	4 years ago	<ul> <li>TA stars</li> <li>TB wold being</li> </ul>	
	🖿 (b	Normalized schema additions (#524)	2 years ago	Y 347 forks	
	tools	Bump golang.org/k/net from 0.7.0 to 0.17.0 in /tools/runsork	5 months ago	Report repository	
	verda 🗧	added del sociator thermostal (del be $\sim 100)$ (4752)	And write	Releases	
	website	Fix device weight set as hugo front matter weight (#723)	4 months ago	No releases published	
	🗅 Makefia.snp	add milesight new sensors (#682)	5 months ago	Packages	
	🕒 exitorizatio	Acts in had booking	4 years repo	No packages published	_
	.gitignore	Ignore Windows executables of validation tooling	2 years ago	Contributors 175	
	.prettierrc	Incorporate JSON schema (#506)	2 years ago	🙈 १ 🖨 📾 📥 🔊 🛶	
	CODIOWNERS	Updale code normen-	2 years ago		<u>م</u>
	C Makefie	Add validation to check images can me be decoded by $gola_m$	3 years ago		
	C READEME and	Improve REACIVIE & exemple (#728)	4 months ago	* TO F CONDICATED	
	peckage took pen	Add Yo Ambrence and Yo and the pro-dual (#724)	4 months argo	Languages	



```
function decodeUplink(input) {
                                                                   ۸
    try{
       var bytes = input.bytes;
       var data = \{\};
       const toBool = value => value == '1';
       var calculateTemperature = function (rawData) {return
(rawData - 400) / 10;
       var calculateHumidity = function(rawData) {return (rawData *
100) / 256;
       var decbin = function (number) {
           if (number < 0) {
               }
           number = number.toString(2);
           return "00000000".substr(number.length) + number;
        }
       function handleKeepalive(bytes, data){
           var tempHex = '0' + bytes[1].toString(16) +
bytes[2].toString(16);
           var tempDec = parseInt(tempHex, 16);
           var temperatureValue = calculateTemperature(tempDec);
           var humidityValue = calculateHumidity(bytes[3]);
           var batteryHex = '0' + bytes[4].toString(16) +
bytes[5].toString(16);
                                                                   ▼
                          are Coloriated - marcel
           trow bottor
```

## **Create Sensor Definitions File**

After downloading your JavaScript decoder either from the TTN repository or directly from the sensor manufacturer, you will need to create a sensor definitions file. This is JSON file that defines all the data types that the decoder reads from the sensor.

In this example, we are using the mClimate CO2 Display decoder we downloaded from the step above. See the code snippet above.

Make note of the sections where the function returns 'data.' These are the specific data points that are added to our sensor definitions file. Below is the JSON sensor definitions file created for the CO2 Display sensor above.

Sensor definitions require the following objects,

- "description": A description of the sensor
- "properties": a list of all the properties the sensor returns including their *name*, *type*, and *units* if applicable.
- "decoder": the name of the JavaScript decoder file.

A few things to note when creating the sensor definitions file.

- The name of the properties must match the spelling and case of the name in the decoder.
- Make sure the data type returned corresponds to the decoder file. For ex., if you list "data.hardwareVersion" as a float, make sure that in the decoder that value is a number. You may have to adjust your decoder.
- The decoder name must match the file name of the JavaScript decoder.



Below is a sample sensor definitions file for a Radio Bridge Temperature Sensor.

```
{
     "description" : "Temperature Sensor",
     "properties" : {
             veviceType"
"HardwareVersion"
"FirmwareVersion"
                                               : {"type" : "uint8"},
                                             : {"type" : "uint8"},
             "FirmwareVersion"
                                              : {"type" : "uint16"},
           "BatteryLevel" : {"type" : "float",
"AccumulationCount" : {"type" : "uint16"},
"TamperSinceLastReset" : {"type" : "bool"},
"CurrentTamperState" : {"type" : "bool"},
                                                                               "units" : "volts"},
            "ErrorWithLastDownLink" : {"type" : "bool"},
"BatteryLow" : {"type" : "bool"},
            "RadioCommError"
                                             : {"type" : "bool"},
            "TamperState"
                                             : {"type" : "bool"},
                                             : {"type" : "uint8"},
            "CurrentSubBand"
            "RSSILastDownlink"
                                             : {"type" : "int8"},
            "SNRLastDownlink"
                                             : {"type" : "int8"},
            "TemperatureEvent"
           "TemperatureEvent" : {"type" : "uint8"}
"CurrentTemperature" : {"type" : "int8",
                                             : {"type" : "uint8"},
                                                                              "units" : "celsius"},
                                             : {"type" : "int8"}
            "RelativeMeasurement"
     },
     "decoder": "radiobridge-decoder.js"
  }
```

#### **Upload Sensor Definitions Files**

Now it's time to upload both files to the Gateway.

1. Navigate to Payload Management-->Sensor Definitions





- 2. Click Import tab
- 3. Enter the Manufacturer and Sensor Type
- 4. Choose the **Sensor Definition** and **Sensor Decoder** files from your machine.

MULTITEC	CHO mPower™ Edge Int MTCDT-L4N1-247A Fi	elligence Conduit - Application Enablement Platform mware 63.2	ᆂ admin 🔹
		IMPORT SENSOR DEFINITION 1	
	Home	i Sensor Definitions 🔮 Default 🕹, Custom ■, Import	
	R LoRaWAN ®	Details	
	Network Settings	Manufacturer	
	Key Management	mClimate Climate	
	Gateways	Sensor Type Source	
	Devices	C02-Display ora	
	Device Groups	Files	
	Profiles	Sancer Definition	
	Packets	mClimate_C02.json (1.33 KB)	
	Downlink Queue	Sensor Decoder	
	Operations	co2-display.js (9.14 KB)	
	ック Payload Management		
	BACnet Configuration	✓ Import	
	BACnet Objects		

5. Click Import

## **Troubleshooting Sensor Definitions & Decoders**

#### From the UI

Error messages from the Payload Decoder can be found under Status & Logs  $\rightarrow$  Payload Management under the LoRaWAN Decoder.

TITECHO MPower" Ed	Prince Longuit - Apprecation Enablement Platform			1 adr
	LORAWAN DECODER LOGS			
E Home	O BACreel @ LoflamAn Decoder			
🛞 LoRaWAN 8	LoRwWAN Decoder Logs	-	* Download Logs	
Re Payload Managemen	And an Arrest of the sectors of the sector devices devices a			
BACHest Contrigutation BACHest Dejects Managed Sensors Sensor Definitions	[20] K. (1771). A D. WHERE & M. R. LER, Downson accesses - Processing 2014 (E), 2017 [A, 17, 7711]; St. W. Wold, Downson-Accesses - TraNAM, 5 (2014).		1. <b></b>	
Setup				
D+ Cellular			76	
🐨 Wreless	C Refeatiling			
ts Frewall				
🚓 Tunnels				
* Edministration				

# **Through SSH**

For those with more programming experience, you can test the decoders directly on the operating system of the gateway. Enable SSH through Administration  $\rightarrow$  Access Configuration

SSH into the gateway. You can find custom decoder files in the /var/config/scada/sensors/ directory.



There you can run the following command below to test the decoder directly. We recommend testing before uploading the decoder onto the device.

```
admin@mtcdt:~$ packet-decoder-cli \
    --js-uplink-decoder c02-didplay.js \
    --port 1 \
    --packet 0102A33E0BE01A0000000 \
    --stdout
```