# Configuring Your Heltec Sensor

After assembling the sensor device you will need to configure it to work with the Connected Humber website using a serial communications program such as the serial monitor in the Arduino IDE or PuTTY.

Arduino IDE V 1.8.10 or later is recommended because it has serial command history allowing you to more easily enter the commands required.

If using PuTTY you need to enable local echo.

You will need a USB cable to connect to your computer and you will have to set the baud rate to 115200. The default 8N1 serial format is ok.

## Before you start

You need to determine the device display name for the Connected Humber web page.

Our convention dictates that this will be something like CHASH-abcdef-01 where abcdef are the last 6 hex digits of the Heltec MAC address. This is printed to the serial port when you reset the device.

The sensor’s proposed GPS location is also required to place it on the map.

This information can be added to our database in advance.

## Which way to send us your data

The sensor can send data to our database using either WiFi or LoRa.

If you have a local WiFi network in range we recommend using that as it is less likely to be affected by weather.

Alternatively you can configure the device to use LoRa and send the data via TTN but you need to check if you are able to connect to the LoRa network first. Range is affected by buildings in your line of site to a LoRa gateway.

Do not configure the sensor to send the data both ways.

## Configuring for WiFi & MQTT

If using WiFi you will need to enable your sensor to use the local WiFi AND you will need to configure the MQTT settings.

{"v":1, "t" : "Sensor01", "c" : "wifi", "o" : "off"}

{"v":1, "t":"sensor01", "c" : "wifi", "o" : "ssid", "set":1, "val":"Your SSID"}

{"v":1, "t" : "sensor01", "c" : "wifi", "o" : "pwd", "set":1, "val":"password"}

{"v":1, "t" : "Sensor01", "c" : "wifi", "o" : "on"}

## Configuring for LoRa

We will add your device to the Things Network and provide you with three pieces of information: Device Address, Network Session Key and Application EUI. We need to use the following commands to put these into the sensor using the serial communications link:

{"v":1, "t" : "sensor01", "c" : "lora", "o" : "state", "val" : "off"}

{"v":1, "t" : "sensor01", "c" : "lora", "o" : "access", "val":"abp"}

{"v":1, "t" : "sensor01", "c" : "lora", "o" : "abpdev", "val":"Device Address"}

{"v":1, "t" : "sensor01", "c" : "lora", "o" : "abpnwk", "val":"Network Session Key"}

{"v":1, "t" : "sensor01", "c" : "lora", "o" : "abpapp", "val":"Application EUI"}

{"v":1, "t" : "sensor01", "c" : "lora", "o" : "state", "val" : "on"}

The next step is to hard reset the device. You can do this by pressing the reset button on the device.

## Setting the Reporting Frequency

By default the sensor sends its information once per minute but we require that to be changed to once every 6 minutes (360 seconds). This gives 10 readings per hour.

Use one of the following commands. You will hear the SDS011 sensor switch on and off. It turns on for 30 seconds, the sensor transmits the data and the SDS011 is then turned off for the remaining 330 seconds. This imporves the SDS011 lifespan by reducing wear on the fan bearings.

If using LoRa

{"v":1, "t" : "sensor01", "c" : "lora", "o" : "gap", "val":360}

If using WiFi

{"v":1, "t" : "sensor01", "c" : "mqtt", "o" : "gap", "val":360}