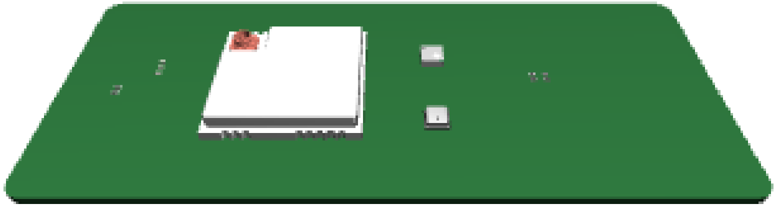
Portenta H7 LoRa Air Station

Quick Start



1. Download and install the Arduino IDE.

Please visit the Arduino download page to get the open-source Arduino Software (IDE): https://www.arduino.cc/en/software

2. Add Portenta H7 support in the board manager.

Open the Arduino IDE and the Boards Manager from *Tools > Board > Boards Manager* and search the name of your board (Portenta). Once the right core shows up in the search results, click *Install*.

3. Connect the LoRa Air Station to Portenta H7 and plug into the USB port of your computer.

Navigate to *Tools > Board* and make sure the correct board has been chosen.

4. Find AppKey and AppEUI of your RAK module.

Load an example project for serial connection by clicking *File > Examples > 04. Communication > SerialPassThrough*. Change the baud rate to 115200 bsp and upload the code. Open the Serial Monitor and enter "at+get_config=lora:status". It will get the channel configuration at the node end including AppKey and AppEUI.

5. Add your device in a LoRaWAN Network Server.

Register your device with its DeviceEUI (60C5A8FFFE76FD6D) using the default Over The Air Activation (OTAA). To communicate with a device via The Things Network, follow the instruction: https://www.thethingsnetwork.org/docs/devices/registration/

6. Use a sample code to get real-time sensor data.

The example code requires "Adafruit_SGP40", "Adafruit_BME680", "TheThingsNetwork", "CayenneLPP", etc. To install a new library into the Arduino IDE, click *Sketch > Include Library > Manage Libraries*.

The Cayenne Low Power Payload (LPP) will be used to encode and decode data from sensors. For more information, please visit https://www.thethingsnetwork.org/docs/devices/arduino/api/cayennelpp/. Open The Things Network Console and change the Payload formatter type to CayenneLPP.

Upload the following code and check the Serial Monitor or The Things Network Console to watch the real-time data from sensors. Make sure to use the appropriate AppEUI and AppKey.

```
#include <Wire.h>
#include <CayenneLPP.h>
#include <TheThingsNetwork.h>
#include <Adafruit_SGP40.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME680.h>

#define SEALEVELPRESSURE_HPA (1013.25)

const char *appEui = "<your-app-eui>";
const char *appKey = "<your-app-key>";

TheThingsNetwork ttn(Serial1, Serial, TTN_FP_US915);
CayenneLPP lpp(51);

Adafruit_BME680 bme;
Adafruit_SGP40 sgp;

void setup() {
    lpp.reset();
    Serial.begin(115200);
    Serial1.begin(115200);

    while(!Serial);
    sendCommand("at+join\r\n");

    Serial.println("-- STATUS");
    ttn.showStatus();

    Serial.println("-- JOIN");
    ttn.join(appEui, appKey);

    if (!sgp.begin(&Wire1))  {
        Serial.println("SGP40 not found");
        while (1) delay(10);
    }
    Serial.println("Sensirion SGP40 Gas Sensor connected successfully");

    if (!bme.begin(0x76)) {
        Serial.println("BME680 not found");
        while (1) delay(10);
    }
    Serial.println("Bosch BME680 Environmental Sensor connected successfully");
      Serial.println("Welcome to Portenta H7 LoRa Air Station!");
    Serial.println("Sending real-time data ... ");
    delay(5000);
}

String response = "";
String command = "";
uint8_t temperature;
uint8_t humidity;

void loop() {
    if (Serial.available()) {
    Serial1.write(Serial.read());
    }

    if (Serial1.available()) {
    Serial.write(Serial1.read());
    }

      // option 1: using the CayenneLPP library
    printValues();

      // option 2: sending payload
      // available payload formatters:
https://www.thethingsindustries.com/docs/integrations/payload-formatters/
      sendCommand("at+send=lora:1:01670110026801\r\n");
    delay(3000);
}

void printValues() {
      // the serial monitor shows real-time sensor data
    Serial.println("SGP40 Measurement: " + String(sgp.measureRaw()));
    Serial.println("Pressure = " + String(bme.readPressure() / 100.0F) + " hPa");
    Serial.println("Approx. Altitude = " +
String(bme.readAltitude(SEALEVELPRESSURE_HPA)) + " m");

      // adding data to the CayenneLPP buffer
      temperature = bme.readTemperature();
      Serial.println("Temperature = " + String(temperature) + " °C");
      lpp.addTemperature(1, temperature);
      humidity = bme.readHumidity();

      Serial.println("Humidity = " + String(humidity) + " %");
      lpp.addTemperature(1, humidity);

    Serial.println();
    delay(3000);
}

void sendCommand(String atComm){
response = "";
Serial1.print(atComm);
  while(Serial1.available()){
    char ch = Serial1.read();
    response += ch;
  }
  Serial.println(response);
}
```