Executive Summary

This document first walks through a manual analysis of how to interpret the payload of a TBHH100 temp/humidity sensor, then shows a Helium Console decoder that was developed for the TBHH100.

The manual analysis involved converting the base 64 encoded payload into binary form, then using the information in the TBHH100 Reference Manual to interpret the bits.

The end result of the manual decode was a payload of "CPs3M/////8=" read from the Helium Console debugger interpreted as

- Voltage = 3.6V
- Temp = 73.4F
- Relative Humidity=51%

The decoder development involved finding source code for a device very similar to the TBHH100, then modifying that source code. The Reference table has links to information resources used in the development of this document.

Contents

References

Reference	Description
RM_Temperature _ Humidity Sensor.pdf	Reference Manual for the Browan TBHH100 temperature and humidity sensor
https://base64.guru/converter/decode/hex	Base64 to Hex converter
https://www.rapidtables.com/convert/number/index.html	Binary/Octal/Decimal/Hex converter
https://www.rapidtables.com/convert/number/binary-to-decimal.html	Binary to Decimal converter
https://www.thethingsnetwork.org/forum/t/payload-functions-howto/3441	DID NOT USE – but a nice tutorial HowTo document on Payload Functions [HowTo] from Things Network
https://learn.adafruit.com/the-things-network-for-feather/payload-decoding	DID NOT USE – but a nice tutorial Payload Decoding tutorial from Things Network

Author – Mike Boucher, mikeboucher@yahoo.com

https://github.com/SensationalSystems/smart_building_sensors_decoder	Decoder repository where I found a decoder for a sensor very similar to the TBHH100, healthyhome.js. I modified the healthyhome decoder to work for the TBHH100.

Payload Analysis

This section analyzes the payload returned from the Browan TBHH100 temperature and humidity sensor. From the Helium Console debugger, grab the payload information for data returned by the TBHH100.

"payload": "CPs3M/////8=", "payload_size": 8,v "port": 103

The payload is Base 64 encoded. To interpret, the payload needs to be converted to binary.

CPs3M/////8 This is the payload in Base 64 format
08fb3733ffffffff Payload in Hex format from https://base64.guru/converter/decode/hex
08 fb 37 33 ff ff ff ff
00001000 11111011 00110111 00110011 (1st half of) Payload in binary format

Byte	Field	Reading (Hex)	Reading (binary)	Payload Key	Interpretation
0	Status	08	0000 1000	**Sensors status** 0x00: VOC sensor 0x08: Temperature and humidity sensor	Device is a Temp and humidity sensor
1	Battery	fb	1111 1011	**Battery level** Bits [3:0] unsigned value v, range 1 – 14; battery voltage in V = (25 + v) ÷ 10. Bits [7:4] RFU	Voltage = (25+11)/10 Voltage = 3.6V
2	Temp	37	0011 0111	**Temperature as measured by the digital sensor** Bits [6:0] unsigned value τ, range 0 – 127; temperature in °C = τ - 32. Bit [7] RFU measurement range -32 to 95°C	Value = 55 -32+55 = 23C = 73.4F Temp = 73.4F
3	RH	33	0011 0011	**Relative humidity as measured by digital sensor** Bits [6:0] unsigned value in %, range 0-100. The value 127 indicates measurement error. Bit [7]RFU	Value = 51 = 51% Humidity Relative Humidity=51%
4	CO2	ff	1111 1111	**Equivalent CO 2 level as measured by digital**	

Author – Mike Boucher, mikeboucher@yahoo.com

5	CO2	ff	1111 1111	**sensor** Bits [15:0] RFU The value is always displayed as 0xffff because no CO 2 sensor is installed in this module.	
6	VOC	ff	1111 1111	**Total Volatile Organic Compound Level as measured by the digital sensor** Bits [15:0] RFU The value is always displayed as 0xffff because no VOC sensor is installed in this module.	
7	VOC	ff	1111 1111		

4.1.2 Payload

Port	107
Payload Length	8 bytes

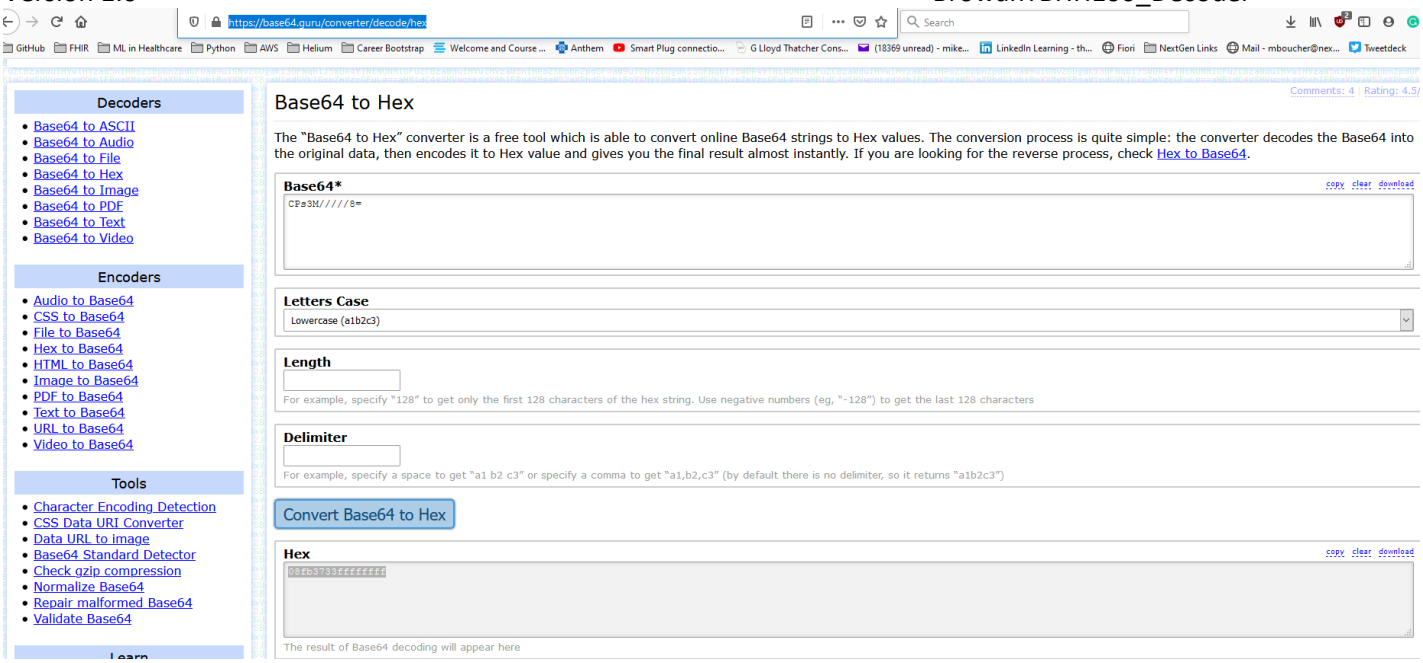
Bytes	0	1	2	3	4	5	6	7
Field	Status	Battery	Temp. (environment)	RH	CO_2		VOC	

Author – Mike Boucher, mikeboucher@yahoo.com

4.1.2 Payload (continue)

Status	**Sensors status**	
	Bits [7:0]	0x00: VOC sensor 0x08: Temperature and humidity sensor
Battery	**Battery level**	
	Bits [3:0]	unsigned value v, range 1 – 14; battery voltage in V = (25 + v) ÷ 10.
	Bits [7:4]	RFU
Temp(environment)	**Temperature as measured by the digital sensor**	
	Bits [6:0]	unsigned value T, range 0 – 127; temperature in °C = T - 32.
	Bit [7]	RFU measurement range -32 to 95°C
RH	**Relative humidity as measured by digital sensor**	
	Bits [6:0]	unsigned value in %, range 0-100. The value 127 indicates measurement error.
	Bit [7]	RFU
CO_2	**Equivalent CO_2 level as measured by digital sensor**	
	Bits [15:0]	RFU The value is always displayed as 0xffff because no CO_2 sensor is installed in this module.
VOC	**Total Volatile Organic Compound Level as measured by the digital sensor**	
	Bits [15:0]	RFU The value is always displayed as 0xffff because no VOC sensor is installed in this module.

Author – Mike Boucher, mikeboucher@yahoo.com



(screenshot above) Converting sensor payload from Base 64 to Hex on https://base64.guru/converter/decode/hex

Decoder Function

```
/*
* Decoder for the Browan TBHH100 Temperature and Humidity Sensor
*
* The starting source code for TBHH100.js came from the file, healthyhome.js found in this repository:
* https://github.com/SensationalSystems/smart_building_sensors_decoder
* which was created by Cameron Sharp at Sensational Systems - cameron@sensational.systems
 *
 *
 */

function Decoder(bytes, port) {

  var params = {
    "bytes": bytes
  };

  // VOC Measurement
            // Disabled on the TBHH100, i.e. is always ffff so comment this section out
  /*
            voc = (bytes[7] << 8) | bytes[6];
  if (voc === 65535) {
    voc_error = true;
  } else {
    voc_error = false;
  }
            */

  // CO2 Measurement
            // Disabled on the TBHH100, i.e. is always ffff so comment this section out
            /*
  co2 = (bytes[5] << 8) | bytes[4];
  if (co2 === 65535) {
    co2_error = true;
  } else {
    co2_error = false;
  }
            */
```

```
  // Humidity Measurement
  rh = bytes[3] &= 0x7f;
  if (rh === 127) {
    rh_error = true;
  } else {
    rh_error = false;
  }

  // temp measurement (in degrees C)
  temp = bytes[2] & 0x7f;
  temp = temp - 32;



  // Battery measurements
  batt = bytes[1] & 0x0f;
  batt = (25 + batt) / 10;

  //params.voc = voc;
  //params.voc_error = voc_error;
  //params.co2 = co2;
  // params.co2_error = co2_error;
  params.rh = rh;
  //params.rh_error = rh_error;
  params.temp = temp;
  params.batt = batt;

  return params;
}
```

Helium Console Debug Output

The JSON below is captured from the Console debug output.
Relative Humidity = 61%
Temperature = 28C = 82.4F


```
{
  "channels": [
    {
      "debug": {
        "req": {
          "body": {
            "app_eui": "58A0CB0000210000",
            "dc": {
              "balance": 1010000,
              "nonce": 1
            },
            "decoded": {
              "payload": {
                "batt": 3.6,
                "bytes": [
                  8,
                  251,
                  60,
                  61,
                  255,
                  255,
                  255,
                  255
                ],
```


Author – Mike Boucher, mikeboucher@yahoo.com

```
            "rh": 61,
            "temp": 28
          },
          "status": "success"
        },
        "dev_eui": "58A0CB000011C1D4",
        "devaddr": "07010048",
        "fcnt": 195,
        "hotspots": [
          {
            "channel": 12,
            "frequency": 904.7000122070312,
            "id": "112kk7sLkuPybrPDE4ZPAYcAXzuPZbV3F2MH5adatdGohmRX5zJW",
            "lat": 34.04490799366432,
            "long": -83.90869057221556,
            "name": "clever-orange-mustang",
            "reported_at": 1596920100,
            "rssi": -45,
            "snr": 14,
            "spreading": "SF10BW125",
            "status": "success"
          }
        ],
        "id": "759ab813-673b-45fb-bb8e-6abdb66e7860",
        "metadata": {
          "labels": [
            {
              "id": "7338d945-47a7-4b0b-ad43-18f1693adade",
              "name": "TBHH100",
              "organization_id": "27706bf3-96b3-4ccf-ae94-439c44858866"
            }
          ],
          "organization_id": "27706bf3-96b3-4ccf-ae94-439c44858866"
        },
        "name": "Gary's Browan TBHH100",
        "payload": "CPs8Pf////8=",
        "port": 103,
        "reported_at": 1596920100
      }
    }
  },
  "description": "console debug",
  "id": "no_integration_id",
  "name": "Console Debug Integration",
  "status": "success"
 }
],
"devaddr": "07010048",
"device_name": "Gary's Browan TBHH100",
"frame_down": 2,
"frame_up": 195,
"hotspots": [
 {
  "frequency": 904.7000122070312,
  "id": "112kk7sLkuPybrPDE4ZPAYcAXzuPZbV3F2MH5adatdGohmRX5zJW",
  "name": "clever-orange-mustang",
  "rssi": -45,
  "snr": 14,
  "spreading": "SF10BW125"
 }
```

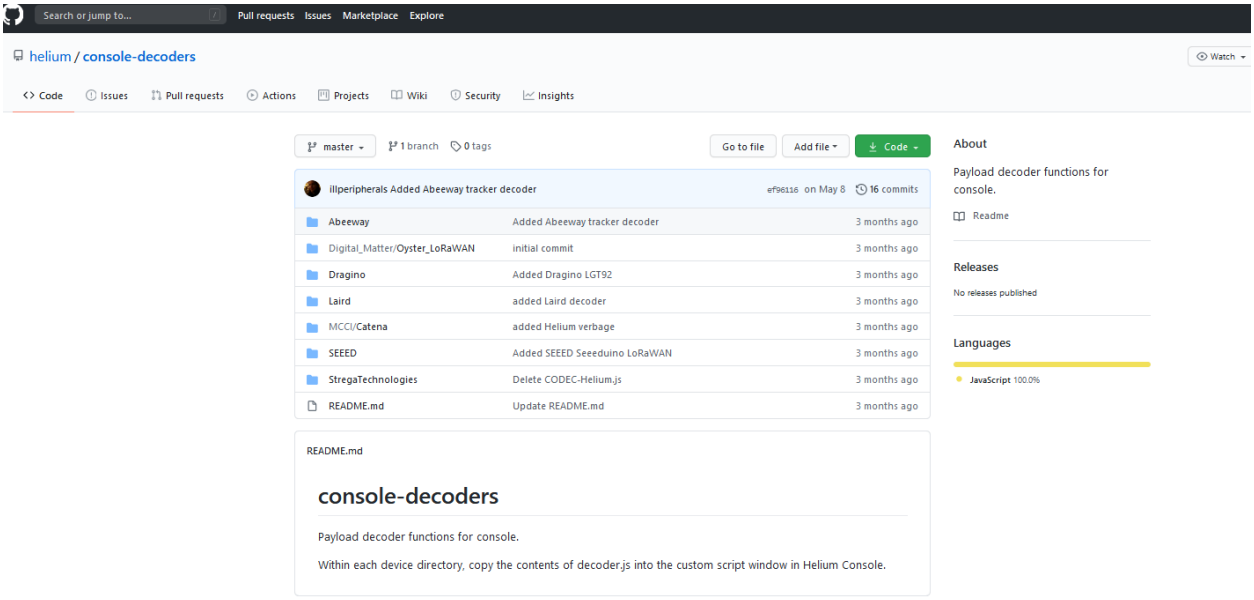

Author – Mike Boucher, mikeboucher@yahoo.com

```
  ],
  "id": "3c8bb4f9-a2ff-441c-a7c1-1686e4456ea5",
  "payload": "CPs8Pf////8=",
  "payload_size": 8,
  "port": 103
```


- ~~Ask on Helium sensor channel if anyone knows where to find or is willing to share a TBHH100 decoder~~

Existing Decoder Research

The Helium decoder repository is here:
https://github.com/helium/console-decoders



(above) Snapshot of the Helium decoder repository

Folder	Device Name	Description	Folder Contents	Note
Abeeway	Abeeway Master Tracker	Low Power Industrial GPS Tracker	decoder.js	
Digital_Matter	Oyster/Yabby	GPS tracker	decoder.js	Helium docs here
Dragino	LGT92	GPS tracker	decoder-v1_5_0-01.js	
Dragino	LHT65	Temp and Humidity Sensor	decoder.js	
Dragino	LSN50	Long Range LoRa Sensor Node – open source Development Kit	decoder.js	
Dragino	LT22222-L	I/O Controller	decoder.js	Documentation included
Dragino	Soil Probe	Soil Moisture & EC Sensor for IoT of Agriculture. It detects Soil Moisture, Soil Temperature and Soil Conductivity	decoder.js	More info here

Author – Mike Boucher, mikeboucher@yahoo.com

Laird	SentriusTM RS1xx Sensor	Temp and Humidity Sensor	decoder.js	Documentation included
MCCI	MCCI Catena® 4612	Complete single-board IoT device that measures temperature, pressure, humidity, and lux, (and maybe soil)	• decoder-generic.js • decoder-port2.js • decoder-port3.js	Well commented code, specifically mentions Helium
SEEED	Seeeduino_LoRaWAN	Arduino development board	decoder.js	More info here
SEED	SenseCAP	SenseCAP is a series of industrial IoT products.	decoder.js	Helium write-up here
StregaTechnologies		wireless smart valve	• DECODER-V3-V4.js • ENCODER-V3-V4.js	More info here