

# Meteo Data via LoRaWAN

DuPe ([www.dupedup.cz](http://www.dupedup.cz))  
Prototype Version: 2.0 - 09/2025

## Purpose:

Measure meteo data (temperature, pressure, humidity) by sensor. Data is periodically sent via LoRaWAN network to server at provider LoRaWAN. Received data is POSTed to web server into CSV file (PHP code). Data is readed by WEB browser and displayed as charts (JS code using google charts API).

## Hardware components:

### 1) MCU Cube Cell HTTC AB01 + Antenna 868 MHz (Heltec Automation)

- Master Chip ASR6501 (48 MHz ARM® Cortex® M0+ MCU )
- Memory 128KB internal FLASH; 16KB internal SRAM
- LoRa Chipset SX1262
- Frequency 470~510 MHz, 863~923 MHz
- Max TX Power 22dB ± 1dB
- Receiving sensitivity -135 dBm
- UART x 1; SPI x 1; I2C x 1; SWD x 1; 12-bits ADC input x 1; 8-channel DMA engine; GPIO x 8
- Micro USB x 1; LoRa Antenna interface(IPEX) x 1;
- USB to Serial Chip CP2102
- Battery 3.7V Lithium(SH1.25 x 2 socket)
- Solar Energy 5.5~7V solar panel
- Low Power Deep Sleep 3.5µA
- Operating temperature -20 ~ 70 °C
- Dimensions 41.5 x 25 x 7.6 mm

### 2) SENSOR TEMPERATURE/HUMIDITY - SHT40

- Digital interface I<sup>2</sup>C
- Supply voltage 1.1-3.6V DC (3.3V DC)
- Resolution 16bit
- Operating range Temperatuer -40-125°C, 0.01°C , ±0.2°C
- Operating range Humidity 0-100% RH, 0.01 RH, ±1.8% RH

### VARIANT : SENSOR DIGITAL TEMPERATURE/HUMIDITY/ PRESSURE - BME280

- Digital interface I<sup>2</sup>C (up to 3.4 MHz)
- Supply voltage: 1.71 V to 3.6 V (3.3V DC)
- Operating range Temp: -40 +85°C, + 1°C, 0.01° C
- Operating range Press: 300-1100hPa, ± 1hPa, 0.18Pa
- Operating range Humi: 0 -100% RH, + 3% , 0,008%

### 3) Solar Panel 6V

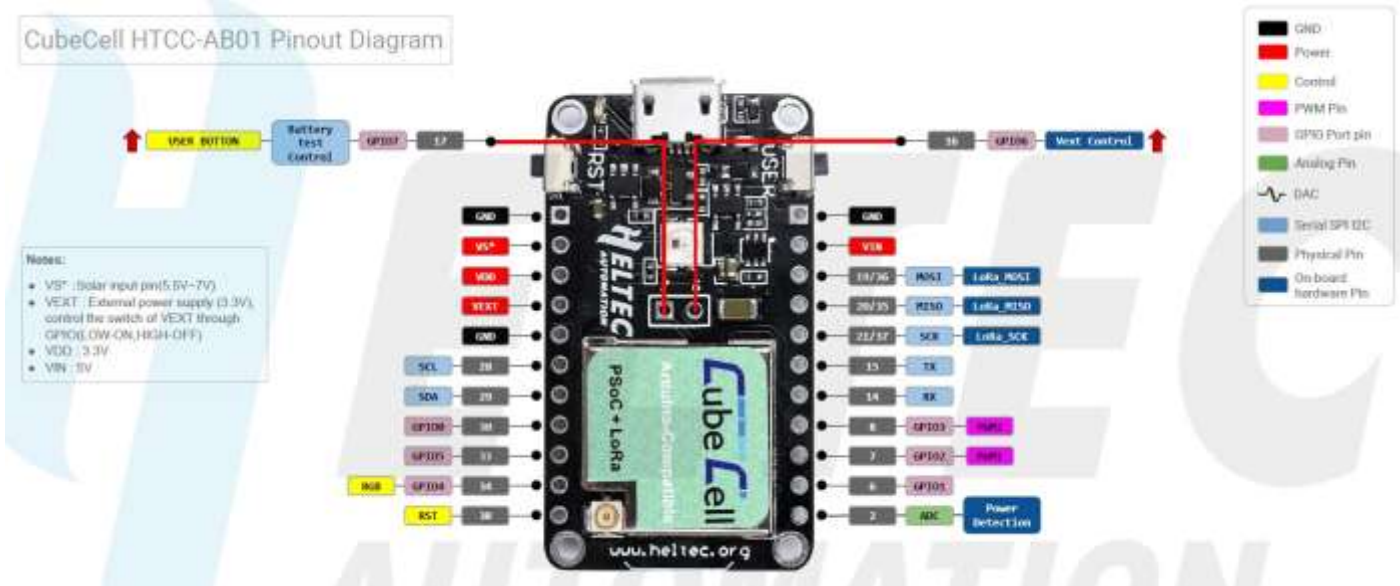
- Output min 300mA (2W)
- Size 110x136x3mm / 52g

### 4) LiPol 1S 1000 mAh 3.7V

## Circuit:

Cube Cell Modul (PIN 5V tolerant)	SENZOR (I2C)	Other
28	SCL	
29	SDA	
VDD (3.3 V OUT)	VCC	
VS ( 5.5 - 7V)		Solar Panel

Battery Connector		ACU LiPo 1S
GND	GND	GND ACU/Solar



### JST-PH DATA

SCL	SDA	VDD	GND
-----	-----	-----	-----



### JST-PH SOLAR

VS	GND
----	-----



(SENSOR JST-SH: red: VCC (3.3V), black: GND, yellow: SCL, blue: SDA)

## Processing:

SENSOR -> SENSOR REGISTRY -> SENSOR BUS (I2C) -> MCU DATA BUS (I2C) -> MCU (STM32) -> MCU DATA BUS -> OLED (I2C) -> LORA RADIO chip (SPI) -> RADIO 868MHz -> LORAWAN GW (UP TO 10 km) -> LORAWAN SERVER REST API -> DATA POST -> WWW SERVER (PHP CODE) -> DATA FILE  
 DATA FILE -> HTML/JS CODE ( API GOOGLE CHARTS ) -> WWW USER BROWSER

Data : 10-20 B

Cycle: 300s

LoRa: 868 MHz , SF 12, BW 125, DR about 300 bps

LoRaWAN: OTTA parameters (devEUI, apEUI, appKEY) were generated on [The Things Network](https://www.thingsnetwork.io/)

LoRaWAN Provider: CeskeRadiokomunikace [Datové centrum, přenos dat, Cloud a IoT | České Radiokomunikace](https://www.datovcentrum.cz/)

## Software Components:

- C/C++ VSCode / Platform IO + Heltec Support
- <https://github.com/HelTecAutomation/CubeCell-Arduino>
- <https://github.com/milesburton/Arduino-Temperature-Control-Library>
- [adafruit/Adafruit\\_SHT4X](https://github.com/adafruit/Adafruit_SHT4X)
- [GitHub - Sensirion/arduino-i2c-sht4x: Arduino library for Sensirion SHT4x sensors](https://github.com/Sensirion/arduino-i2c-sht4x)
- ([https://github.com/adafruit/Adafruit\\_BME280\\_Library](https://github.com/adafruit/Adafruit_BME280_Library))

## Code:

### platformio.ini

```
[env:cubecell_board]
platform = heltec-cubecell
board = cubecell_board
framework = arduino
monitor_speed = 115200
build_flags =
  -D ACTIVE_REGION=LORAMAC_REGION_EU868
  -D REGION_EU868
  -D LORAWAN_CLASS=CLASS_A
  -D LORAWAN_DEVEUI_AUTO=0
  -D LORAWAN_NETMODE=true
  -D LORAWAN_ADR=true
  -D LORAWAN_UPLINKMODE=true
  -D LORAWAN_Net_Reserve=false
  -D LORAWAN_AT_SUPPORT=0
  -D LORAWAN_PREAMBLE_LENGTH=8
  -D LORAWAN_DebugLevel=2
  -D RGB=1
lib_deps = adafruit/Adafruit SHT4x Library@^1.0.5
lib_deps = adafruit/Adafruit BME280 Library@^2.2.4
```

### setup-lora.h

```
#define AT_SUPPORT 0
#define RGB 1
#define DELAYTIME 1000
#define ACTIVE_REGION LORAMAC_REGION_EU868
#define CLASS_MODE CLASS_A
#define LORAWAN_DEVEUI_AUTO 0
#define CubeCell_BoardPlus
#define REGION_EU868
#define LORAWAN_NETMODE true
#define LORAWAN_ADR true
#define LORAWAN_UPLINKMODE true
#define LORAWAN_Net_Reserve false
#define LORAWAN_AT_SUPPORT 0
#define LORAWAN_PREAMBLE_LENGTH 8
#define LORAWAN_DebugLevel 2
/* OTAA para*/
uint8_t devEui[] = { };
uint8_t appEui[] = { };
uint8_t appKey[] = { };
/* ABP para*/
uint8_t nwkSKey[] = { };
uint8_t appSKey[] = { };
uint32_t devAddr = ( uint32_t )0x007e6ae1;
/*LoraWan channelsmask, default channels 0-7*/
uint16_t userChannelsMask[6]={ 0x00FF,0x0000,0x0000,0x0000,0x0000,0x0000 };
/*LoraWan region, select in arduino IDE tools*/
LoRaMacRegion_t loraWanRegion = ACTIVE_REGION;
/*LoraWan Class, Class A and Class C are supported*/
DeviceClass_t loraWanClass = LORAWAN_CLASS;
/*the application data transmission duty cycle. value in [ms].*/
uint32_t appTxDutyCycle = 300000;
/*OTAA or ABP*/
bool overTheAirActivation = LORAWAN_NETMODE;
/*ADR enable*/
bool loraWanAdr = LORAWAN_ADR;
/* set LORAWAN_Net_Reserve ON, the node could save the network info to flash, when node reset not need to join again */
bool keepNet = LORAWAN_NET_RESERVE;
```

/\* Indicates if the node is sending confirmed or unconfirmed messages \*/

bool isTxConfirmed = LORAWAN\_UPLINKMODE;

uint8\_t appPort = 2;

uint8\_t confirmedNbTrials = 4;

## MCU Source Code (sensor STH40)

```
/*
 * METEO HUMIDITY + TEMPERATURE LORA868 - nyní Skalka
 * DUPE 11/2024 09/2025
 * HELTEC CUBECELL AB01 LORA 868 MHz, senzor SHT40 (SPI), LORAWAN Ceske Radiokomunikace
 * SPI SDA 29 | SCL 28
 * CO https://heltec-automation-docs.readthedocs.io/en/latest/index.html
 * set LoraWan_RGB to Active,the RGB active in loraWan
 * RGB red means sending;
 * RGB purple means joined done;
 * RGB blue means RxWindow1;
 * RGB yellow means RxWindow2;
 * RGB green means received done;
 */
#include "LoRaWan_APP.h"
#include "Arduino.h"
#include "Adafruit_SHT4x.h"
#include <Wire.h>
#include "setup-lora.h"
#define SDAI2CPIN 29 // I2C SDA bus
#define SCLI2CPIN 28 // I2C SCL bus
uint16_t baseline;
int count = 0;
int maxtry = 50;
bool data = true;
Adafruit_SHT4x sht4 = Adafruit_SHT4x();

static void prepareFakeFrame( uint8_t port)
{
  count++;
  appDataSize = 15;
  char payload[15];
  String payloads = "NODATA-";
  uint16_t vbat = getBatteryVoltage();
  payloads.concat(count);
  payloads.concat("-");
  payloads.concat(vbat);
  payloads.toCharArray(payload,15);
  // appData = (unsigned char *)payload;
  strcpy ((char*)appData, payload);
  Serial.println(payloads);
}

static void prepareDataFrame( uint8_t port )
{
  count++;
  appDataSize = 14;
  char payload[14];
  String payloads = "";
  sensors_event_t humidity, temp;
  sht4.getEvent(&humidity, &temp);
  delay(500);
  float t = temp.temperature ;
  float h = humidity.relative_humidity;
  uint vbat = getBatteryVoltage();
  Serial.print("Temperature: ");
  Serial.print(t);
```

```

Serial.print(" Humidity: ");
Serial.print(h);
Serial.print(" Voltage: ");
Serial.println(vbat);
int temperature = 100*(t+50); // 4 bytes
int humi = 100*(h+100); // 5 bytes
payloads.concat(temperature);
payloads.concat(humi);
payloads.concat(vbat);
payloads.toCharArray(payload,15);
// appData = (unsigned char *)payload;
strcpy ((char*)appData, payload);
Serial.print("Message=");
Serial.println(payloads);
}
void setup() {
  Serial.begin(115200);
  boardInitMcu();
  deviceState = DEVICE_STATE_INIT;
  Wire.begin(SDAI2CPIN, SCLI2CPIN);
  Serial.println("Adafruit SHT4x test");
  if (! sht4.begin()) {
    Serial.println("Couldn't find SHT4x");
    data = false;
    // while (1) delay(1);
  }
  else {
    Serial.println("Found SHT4x sensor");
    Serial.print("Serial number 0x");
    Serial.println(sht4.readSerial(), HEX);
    sht4.setPrecision(SHT4X_HIGH_PRECISION);
    sht4.setHeater(SHT4X_NO_HEATER);
    data = true;
  }
  LoRaWAN.ifskipjoin();
}
void loop()
{
  switch( deviceState )
  {
    case DEVICE_STATE_INIT:
    {
      #if(AT_SUPPORT)
        getDevParam();
      #endif
      printDevParam();
      LoRaWAN.init(loraWanClass,loraWanRegion);
      deviceState = DEVICE_STATE_JOIN;
      break;
    }
    case DEVICE_STATE_JOIN:
    {
      LoRaWAN.join();
      break;
    }
    case DEVICE_STATE_SEND:
    {
      (data) ? prepareDataFrame( appPort ): prepareFakeFrame( appPort );
      LoRaWAN.send();
      deviceState = DEVICE_STATE_CYCLE;
      break;
    }
    case DEVICE_STATE_CYCLE:

```

```

    {
        // Schedule next packet transmission
        txDutyCycleTime = appTxDutyCycle + randr( 0, APP_TX_DUTYCYCLE_RND );
        LoRaWAN.cycle(txDutyCycleTime);
        deviceState = DEVICE_STATE_SLEEP;
        break;
    }
    case DEVICE_STATE_SLEEP:
    {
        LoRaWAN.sleep();
        break;
    }
    default:
    {
        deviceState = DEVICE_STATE_INIT;
        break;
    }
}
}

```

## PHP Code

```

<!DOCTYPE html>
<head>
  <meta http-equiv="content-language" content="cs" />
  <meta http-equiv="content-type" content="text/html; charset=windows-1250" />
  <link rel='stylesheet' href='/styles/styl2.css' type='text/css' />
  <meta name="author" content="dupe" />
  <title>
    LORAWAN
  </title>
</head>
<body>
<?php
header("Content-Type: text/html; charset=windows-1250");
?>
<?php
// DUPE 21/09/2020 21/11/2024
// DATA LORAWAN HELTEC CUBECCELL-1 (SENZOR)
// JSON POST FROM LoRaWAN CRO -> TXT
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // $write_data = "-- POST DATA --";
  $raw_data = file_get_contents('php://input');
  //echo $raw_data;
  // parsovani JSON prvni urovne
  $json_data = json_decode($raw_data);
  //echo $json_data->data;
  // parsovani JSON druhe urovne
  $data1 = json_decode($json_data->data);
  // parsovani JSON druhe urovne - data z gateways
  // $json_poledat = json_decode($json_data->data, true);
  // print_r($data1);
  // prevod payload z HEX na STRING na FLOAT
  $data=$data1->data;
  $temp=substr($data,0,8);
  $vbat=substr($data,8,10);
  $temp=(Hex2String($temp)/100)-40;
  $vbat=Hex2String($vbat)/1000;
  $EUI = $data1->EUI;
  $ts = $data1->ts;
  $seqno = $data1->seqno;
  $fcnt = $data1->fcnt;
  $toa = $data1->toa;
}

```

```

$dr = $data1->dr;
$freq = $data1->freq;
$data2 = ($data1->gws);
$datumcas= date("Y-m-d H:i:s", substr($ts, 0, 10));
//2016-10-20 20:30:41
$datetime= $data2[0]->time;
$time=substr($datumcas,11,8);
$hour=intval(substr($time,0,2));
$timeCET=$hour . substr($time,2);
$date=substr($datumcas,0,10);
$rssi= $data2[0]->rssi;
$snr= $data2[0]->snr;
$lat= $data2[0]->lat;
$lon= $data2[0]->lon;
// echo $data;
$write_data1 = $date . "," . $time . "," . $temp . "," . $vbat . "," . $snr . "," . $rssi . "," . $fcnt . "," . $toa . "," . $lon . "," . $lat .
"," . $freq . "," . $dr . " " . "\n";
$write_data = $datumcas . "," . $temp . "," . $vbat . "\n";
$write_data_raw = $raw_data . "\n";
// echo $write_data;
} else {
    $write_data = "-- NO POST DATA --";
    echo $write_data;
}
// append TXT
// echo "\n";
$file = "../xxxxx/data-lora-cube1.csv" or die("Unable to open file!");
$SavedFile = fopen($file, "a");
fwrite($SavedFile, $write_data);
fclose($SavedFile);
//echo "Done" . "\n";
// append ALL TXT1
// echo "\n";
$file = "../xxxxx/data-lora-cube1-all.csv" or die("Unable to open file!");
$SavedFile = fopen($file, "a");
fwrite($SavedFile, $write_data1);
fclose($SavedFile);
//echo "Done" . "\n";
// append RAW
// $file = "../xxxxx/data-lora-cube1-raw" or die("Unable to open file!");
// $SavedFile = fopen($file, "a");
// fwrite($SavedFile, $write_data_raw);
// fclose($SavedFile);
// echo "Done" . "\n";
// funkce HEX do Stringu
function Hex2String($hex){
    $string="";
    for ($i=0; $i < strlen($hex)-1; $i+=2){
        $string .= chr(hexdec($hex[$i].$hex[$i+1]));
    }
    return $string;
}
?>
</body>
</html>

```

## HTML Code

```
<!DOCTYPE html>
```

```

<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>LORAWAN</title>
<meta name="description" content="test">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
<script src="/jscripts/jquery.csv.min.js"></script>
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
// Load the Visualization API
google.charts.load('current', {'packages':['corechart']});
// Set a callback to run when the Google Visualization API is loaded.
google.charts.setOnLoadCallback(drawVisualization);
// Callback that creates a data table, instantiates the chart, passes in the data and draws it.
function drawVisualization() {
$.get("data-lora-cube1.csv", function(csvString) {
// transform the CSV string into a 2-dimensional array
var arrayData = $.csv.toArrays(csvString, {onParseValue: $.csv.hooks.castToScalar});
// this new DataTable object holds all the data
var data = new google.visualization.arrayToDataTable(arrayData);
var options = {
title: 'LoRaWAN (HELTEC CubeCell-1 + sensor DS18B20)',
titleTextStyle : {color: 'grey', fontSize: 15},
// curveType: 'function',
width: 1000,
height: 500,
series: {
0: {targetAxisIndex: 0},
1: {targetAxisIndex: 1}
},
vAxes: {
0: {title: 'Temperature [°C]', titleTextStyle: {color: 'blue'}, format: '#'},
1: {title: 'VBAT [V]', titleTextStyle: {color: 'red'}}
},
hAxes: {
0: {title: 'Date Time ', titleTextStyle: {color: 'grey'}}
},
};
var chart = new google.visualization.LineChart(document.getElementById('chart_div'));
chart.draw(data, options);
});
}
function zobrazData () {
$.get("data-lora-cube1-all.csv", function(csvString) {
var arrayData = $.csv.toArrays(csvString, {onParseValue: $.csv.hooks.castToScalar});
var delka = arrayData.length;
var radek = arrayData[delka-1];
// document.getElementById("radek").innerHTML = radek;
radekData = String(radek).split(",");
document.getElementById("div0").innerHTML = "Last Data Record";
document.getElementById("div1").innerHTML = "<strong>Date: </strong>" + radekData[0];
document.getElementById("div2").innerHTML = "<strong>Time (CET) : </strong>" + radekData[1];
document.getElementById("div3").innerHTML = "<strong>Temp [°C] : </strong>" + radekData[2];
document.getElementById("div4").innerHTML = "<strong>Vbat [V] </strong>:" + radekData[3];
document.getElementById("div5").innerHTML = "<strong>Sequence Number:</strong>" + radekData[4];
document.getElementById("div6").innerHTML = "<strong>Row Number : </strong>" + delka;
document.getElementById("div7").innerHTML = "<strong>RSSI [dBm] : </strong>" + radekData[5];
document.getElementById("div8").innerHTML = "<strong>SNR [dB] : </strong>" + radekData[6];
document.getElementById("div9").innerHTML = "<strong>TimeOnAir [ms] : </strong>" + radekData[7];
document.getElementById("div10").innerHTML = "<strong>GPS : </strong>" + radekData[8];
document.getElementById("div11").innerHTML = "<strong>Frequency : </strong>" + radekData[9];

```

```
document.getElementById("div12").innerHTML = "<strong>Spreading Faktor : </strong> " + radekData[10];
});
}
zobrazData();
document.write ("© DUPE (www.dupedup.cz) 09/2020");
</script>
</head>
<body>
<div id="chart_div" style="width: 1000px; height: 500px"></div>
<div id="radek" style="color:blue; width: 1000px; height: 20px"></div>
<div id="div0" style="font-weight: bold; color:red; width: 1000px; height: 20px"></div>
<div id="div1" style="width: 1000px; height: 20px"></div>
<div id="div2" style="width: 1000px; height: 20px"></div>
<div id="div3" style="width: 1000px; height: 20px"></div>
<div id="div4" style="width: 1000px; height: 20px"></div>
<div id="div5" style="width: 1000px; height: 20px"></div>
<div id="div6" style="width: 1000px; height: 20px"></div>
<div id="div7" style="width: 1000px; height: 20px"></div>
<div id="div8" style="width: 1000px; height: 20px"></div>
<div id="div9" style="width: 1000px; height: 20px"></div>
<div id="div10" style="width: 1000px; height: 20px"></div>
<div id="div11" style="width: 1000px; height: 20px"></div>
<div id="div12" style="width: 1000px; height: 20px"></div>
<div id="div13" style="width: 1000px; height: 20px"></div>
</body>
</html>
```