

```

<?php
error_reporting(0);

header("Refresh: 30; url=" . $_SERVER["PHP_SELF"]);

// PREMESSE

// questo codice è liberamente utilizzabile ed è da considerarsi "as is - visto
e piaciuto". E' stato realizzato nell'ambito del progetto sperimentale di
realizzazione di una stazione meteo IoT LoRaWAN a basso costo, come indicato
nell'articolo
http://meteoravanel.it/approfondimento/come-autocostruire-una-stazione-meteo-con
-tecnologia-iot-lorawan-parte-1/ e seguenti a cura di Andrea Costantini Tecnico
Meteorologo certificato (WMO 1083 - registro DEKRA DTC-TMT-001-17 secondo UNI
CEI EN ISO/IEC 17024:2012). www.meteoravanel.it

// Errori, bug o altre imprecisioni sono sempre possibili: una collaborazione
reciproca può portare all'ottimizzazione e al miglioramento complessivo

// PARTE A) dichiarare il metodo di "cattura" dati mediante la funzione curl
della stringa in formato json ed importare l'ultimo dato disponibile

$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, '[inserire il vostro url da database TTN]');
//richiama il database indicando la chiave di autorizzazione e la tipologia di
chiamata. Chiama le ultime 24 ore per far si che la sommatoria delle
precipitazioni alle 23.59.59 sia quella delle 24 ore precedenti (la gestione
dei valori intermedi viene gestita dalla rispettiva funzione di comparazione del
tempo nel ciclo for).

curl_setopt($curl, CURLOPT_HTTPHEADER, array(
    'Authorization: key ttn-account-[inserire la chiave di decodifica]',
    'Content-Type: application/json',
));
curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($curl, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);

$result = curl_exec($curl); //legge il contenuto del curl

if(!$result){die("Dati non aggiornati - riprovare più tardi");}
curl_close($curl);

$response = json_decode($result, true); // trasforma la risposta JSON del
server in un array

// PARTE B) dichiarare le variabili ed eseguirne le elaborazioni (es.
conversione formato timestamp, compensazione umidità relativa, pressione
atmosferica, calcolo accumulo pluviometrico giornaliero, estrazione estremi
temperatura etc)

```

```

$midnight_timestamp = strtotime('today Europe/Rome'); //acquisisce l'orario
attuale e lo converte in Unix timestamp al fine di poterlo confrontare con il
timestamp che è già in questo formato, fornito dal sensore

$total_rain_counter = 0; //inizializzazione della variabile contatore pioggia
$data=null; //inizializzazione della variabile data. Se $response è vuota,
$data resterà uguale a NULL, altrimenti assumerà il valore dell'ultimo invio del
sensore
$primo_campione_del_giorno=0;

for($i=0; $i < count($response); $i++) //si apre il ciclo for con indice i per
il calcolo della somma degli impulsi da convertire in pioggia giornaliera e
altre variabili

{
    $current_time = preg_replace('/^([\^\.\.]+)\.\.([0-9]{1,6})[0-9]*Z$/', '$1.$2Z',
$response[$i]['time']); // Limita a 6 cifre decimali per evitare che la funzione
strtotime fallisca

    $timestamp = strtotime($current_time); //acquisisce l'orario attuale del
sensore e lo converte in unix timestamp

    if($timestamp < $midnight_timestamp) // se il campione è stato inviato il
giorno precedente, passa alla prossima iterazione del ciclo for
    {
        continue;
    }

    if($response[$i]['digital2'])
    {
        $total_rain_counter++; //incrementa l'indice di un'unità solo se digital2 è
diverso da zero
    }

    if($response[$i]['externalTemperature'])
    {
        $sample_time = preg_replace('/^([\^\.\.]+)\.\.([0-9]{1,6})[0-9]*Z$/', '$1.$2Z',
$response[$i]['time']); // Limita a 6 cifre decimali per evitare che la funzione
strtotime fallisca
        $data = $response[$i]; // Salva il campione se contiene un valore valido di
temperatura esterna ma potrei usare altre variabili

        // Inizializza valore min/max
        if($primo_campione_del_giorno==0)
        {
            $primo_campione_del_giorno=1;
            $t_min=$response[$i]['externalTemperature']; // inizializzo con il primo
valore trovato
            $t_max=$response[$i]['externalTemperature']; // inizializzo con il primo
valore trovato
            $t_min_orario=date('H:i:s', strtotime($sample_time)); // inizializzo con
il primo valore trovato
            $t_max_orario=date('H:i:s', strtotime($sample_time)); // inizializzo con
il primo valore trovato

```

```

}

// Aggiornamento valore min/max
if ($t_min > $response[$i]['externalTemperature'])
{
    $t_min=$response[$i]['externalTemperature'];
    $t_min_orario=date('H:i:s', strtotime($sample_time));
}
if ($t_max < $response[$i]['externalTemperature'])
{
    $t_max=$response[$i]['externalTemperature'];
    $t_max_orario=date('H:i:s', strtotime($sample_time));
}
}
}

if(!$data){die("Dati non aggiornati - riprovare più tardi");}

// definisco delle costanti di conversione per le unità di misura

$costante_pl=0.3; //è la costante del pluviometro, ad ogni impulso
corrispondono 0.3mm di pioggia
$Precipitazione = ($costante_pl*$total_rain_counter);

$input_pressione=$data['pressure'];
$costante_pressione=5; //è la correzione da apporre al sensore, ottenuta per
confronto con altre stazioni in 2 mesi di dati
$Pressione = ($costante_pressione+$input_pressione);
$pressione_corr = round($Pressione, 1); //arrotondo a 1 cifra decimale con la
funzione round

// correzione da apporre al sensore umidità ottenuta per confronto con altre
stazioni in base ai valori sotto riportati (in echo)

$costante_umidita_1=20; //ur del sensore tra 0 e 20
$costante_umidita_2=20; //ur del sensore tra 21 e 30
$costante_umidita_3=20; //ur del sensore tra 31 e 40
$costante_umidita_4=20; //ur del sensore tra 41 e 50
$costante_umidita_5=20; //ur del sensore tra 51 e 60
$costante_umidita_6=12; //ur del sensore tra 61 e 70
$costante_umidita_7=6; //ur del sensore tra 71 e 80
$costante_umidita_8=4; //ur del sensore tra 81 e 90
$costante_umidita_9=2; //ur del sensore tra 91 e 95
$costante_umidita_10=0; //ur del sensore tra 96 e 100 nessuna correzione

//in base al valore originale, appongo le corrispondenti correzioni
$input_umidita = $data['humidity']; //legge il valore di umidità del sensore

```

```
$umidita_1 = ($costante_umidita_1+$input_umidita); //costruisco il valore
$umidita_corr_1 = round($umidita_1, 1); //arrotondo il valore a 1 cifra decimale
con la funzione round
$umidita_2 = ($costante_umidita_2+$input_umidita); //costruisco il valore
$umidita_corr_2 = round($umidita_2, 1); //arrotondo il valore a 1 cifra decimale
con la funzione round
$umidita_3 = ($costante_umidita_3+$input_umidita); //costruisco il valore
$umidita_corr_3 = round($umidita_3, 1); //arrotondo il valore a 1 cifra decimale
con la funzione round
$umidita_4 = ($costante_umidita_4+$input_umidita); //costruisco il valore
$umidita_corr_4 = round($umidita_4, 1); //arrotondo il valore a 1 cifra decimale
con la funzione round
$umidita_5 = ($costante_umidita_5+$input_umidita); //costruisco il valore
$umidita_corr_5 = round($umidita_5, 1); //arrotondo il valore a 1 cifra decimale
con la funzione round
$umidita_6 = ($costante_umidita_6+$input_umidita); //costruisco il valore
$umidita_corr_6 = round($umidita_6, 1); //arrotondo il valore a 1 cifra decimale
con la funzione round
$umidita_7 = ($costante_umidita_7+$input_umidita); //costruisco il valore
$umidita_corr_7 = round($umidita_7, 1); //arrotondo il valore a 1 cifra decimale
con la funzione round
$umidita_8 = ($costante_umidita_8+$input_umidita); //costruisco il valore
$umidita_corr_8 = round($umidita_8, 1); //arrotondo il valore a 1 cifra decimale
con la funzione round
$umidita_9 = ($costante_umidita_9+$input_umidita); //costruisco il valore
$umidita_corr_9 = round($umidita_9, 1); //arrotondo il valore a 1 cifra decimale
con la funzione round
$umidita_10 = ($costante_umidita_10+$input_umidita); //costruisco il valore
$umidita_corr_10 = round($umidita_10, 1); //arrotondo il valore a 1 cifra
decimale con la funzione round
```

```
//calcolo dewpoint in base ai vari valori di umidità corretta
$input_temperatura = $data['externalTemperature']; //prendo il valore dal
sensore posto esternamente in schermo solare
$cost1=100;
$cost2=5;
$dewpoint_1 = ($input_temperatura - (($cost1 - $umidita_corr_1)/$cost2));
//formula semplificata per il calcolo del dewpoint
$dewpoint_2 = ($input_temperatura - (($cost1 - $umidita_corr_2)/$cost2));
//formula semplificata per il calcolo del dewpoint
$dewpoint_3 = ($input_temperatura - (($cost1 - $umidita_corr_3)/$cost2));
//formula semplificata per il calcolo del dewpoint
$dewpoint_4 = ($input_temperatura - (($cost1 - $umidita_corr_4)/$cost2));
//formula semplificata per il calcolo del dewpoint
$dewpoint_5 = ($input_temperatura - (($cost1 - $umidita_corr_5)/$cost2));
//formula semplificata per il calcolo del dewpoint
$dewpoint_6 = ($input_temperatura - (($cost1 - $umidita_corr_6)/$cost2));
//formula semplificata per il calcolo del dewpoint
$dewpoint_7 = ($input_temperatura - (($cost1 - $umidita_corr_7)/$cost2));
//formula semplificata per il calcolo del dewpoint
```

```

$dewpoint_8 = ($input_temperatura - (($cost1 - $umidita_corr_8)/$cost2));
//formula semplificata per il calcolo del dewpoint
$dewpoint_9 = ($input_temperatura - (($cost1 - $umidita_corr_9)/$cost2));
//formula semplificata per il calcolo del dewpoint
$dewpoint_10 = ($input_temperatura - (($cost1 - $umidita_corr_10)/$cost2));
//formula semplificata per il calcolo del dewpoint

//calcolo tensione in volt
$costante_tens_batt=1000; //è il fattore per cui dividere il valore originale
in mV per ottenerlo in V
$input_batteria= $data['vdd']; //valore originale da sensore in mV
$batteria = ($input_batteria/$costante_tens_batt); //divido per 1000 per
ottenere un valore in V

//calcolo tempo attuale con conversione formato dati
$current_time = preg_replace('/^([\^\.\.]+)\.\.([0-9]{1,6})[0-9]*Z$/', '$1.$2Z',
$data['time']); // Limita a 6 cifre decimali per evitare che la funzione
strtotime fallisca
$data_attuale = date('d/m/Y H:i:s', strtotime($current_time)); //converte la
data e ora del sensore da English textual datetime in orario italiano
//$timezone = date('P', strtotime($data['time'])); //restituisce il fuso orario
ma nel mio caso preferisco non mostrarlo

//PARTE C) visualizzare i dati finali elaborati in formato leggibile

?>
    <span style="color: #0000FF">Stazione meteo sperimentale IoT LoRaWan
ELSYS ELT2-HP - dati in tempo reale da Vittorio Veneto ogni 5 minuti</span><br
/><br />

<?php

//echo " Stazione meteo sperimentale IoT LoRaWan ELSYS ELT2-HP - dati in tempo
reale da Vittorio Veneto ogni 5 minuti". "<br />". "<br />";
echo " Orario ultimo dato: ".$data_attuale. "<br />" . "<br />";
echo " Temperatura attuale: ".$data['externalTemperature']. "°C " . "<br />";
echo " Temperatura minima odierna: ".$t_min. "°C " . "alle ore ". $t_min_orario.
"<br />";
echo " Temperatura massima odierna: ".$t_max. "°C " . "alle ore ".
$t_max_orario. "<br />";

//in base al valore di umidità e relative correzioni, vengono eseguite delle

```

```
condizioni e relativi output su soglia
if($input_umidita <= 20)
{
  echo " Umidità relativa: ".$umidita_corr_1. "% ". "<br />";
  echo " Dew Point: " .$dewpoint_1. "°C ". "<br />";
}

else if($input_umidita >= 21 && $input_umidita <= 30)
{
  echo " Umidità relativa: ".$umidita_corr_2 . "% ". "<br />";
  echo " Dew Point: " .$dewpoint_2. "°C ". "<br />";
}

else if($input_umidita >= 31 && $input_umidita <= 40)
{
  echo " Umidità relativa: ".$umidita_corr_3 . "% ". "<br />";
  echo " Dew Point: " .$dewpoint_3. "°C ". "<br />";
}

else if($input_umidita >= 41 && $input_umidita <= 50)
{
  echo " Umidità relativa: ".$umidita_corr_4 . "% ". "<br />";
  echo " Dew Point: " .$dewpoint_4. "°C ". "<br />";
}

else if($input_umidita >= 51 && $input_umidita <= 60)
{
  echo " Umidità relativa: ".$umidita_corr_5 . "% ". "<br />";
  echo " Dew Point: " .$dewpoint_5. "°C ". "<br />";
}

else if($input_umidita >= 61 && $input_umidita <= 70)
{
  echo " Umidità relativa: ".$umidita_corr_6 . "% ". "<br />";
  echo " Dew Point: " .$dewpoint_6. "°C ". "<br />";
}

else if($input_umidita >= 71 && $input_umidita <= 80)
{
  echo " Umidità relativa: ".$umidita_corr_7 . "% ". "<br />";
  echo " Dew Point: " .$dewpoint_7. "°C ". "<br />";
}

else if($input_umidita >= 81 && $input_umidita <= 90)
{
  echo " Umidità relativa: ".$umidita_corr_8 . "% ". "<br />";
  echo " Dew Point: " .$dewpoint_8. "°C ". "<br />";
}
```

```
else if($input_umidita >= 91 && $input_umidita <= 95)
{
  echo " Umidità relativa: ".$umidita_corr_9 . "% ". "<br />";
  echo " Dew Point: " . $dewpoint_9. "°C ". "<br />";
}

else if($input_umidita >= 96 && $input_umidita <= 100)
{
  echo " Umidità relativa: ".$umidita_corr_10 . "% ". "<br />";
  echo " Dew Point: " . $dewpoint_10. "°C ". "<br />";
}

echo " Pressione atmosferica relativa: " . $pressione_corr . "hPa ". "<br />";
echo " Precipitazione giornaliera: " . $Precipitazione. "mm". "<br />";
echo " Tensione batteria: " . $batteria. "V ". "<br />". "<br />";
echo " Immagine dimostrativa della stazione meteo". "<br />". "<br />";

// PARTE D) inserire la webcam e/o l'immagine della stazione
$path="https://opendata.piancansigliometeowebcam.it/lab/test.jpg"; // file
contenente l'immagine >> potrebbe essere una webcam!
echo '';

?>
```