TRACK YOUR ATMOSPHERE

Intellectual Output IO4

Station Météo

Interface matérielle et logicielle

GNSS = Global Navigation Satellite Systems

Erasmus+ Projekt 2017-1-DE02-KA202-004229

Version 04/20*FR*

Co-funded by the Erasmus+ Programme of the European Union

Table des matières

Co-funded by the Erasmus+ Programme of the European Union

Antenne GNSS et station de météo sur le toit du Lycée Saint-Cricq

Co-funded by the Erasmus+ Programme of the European Union

Le soutien de la Commission européenne à la production de cette publication ne constitue pas une approbation du contenu, qui reflète les vues des seuls auteurs, et la Commission ne peut être tenue responsable de l'usage qui pourrait être fait des informations qui y sont contenues.

A. Préface

Dans le cadre de la production intellectuelle 4, nous développerons une unité d'apprentissage interactive ("environnement d'apprentissage") en mettant l'accent sur les éléments d'innovation, l'informatique et l'électronique attendus. Alors que dans IO2, la structure du matériel (Arduino et capteurs) et du serveur (impact et potentiel de transférabilité) est facilement donnée à l'élève (ce qui facilite l'essai et la modification du kit de démarrage et la collecte de données), ici, les élèves n'ont qu'un problème à résoudre, à savoir la collecte de données environnementales. Ce problème leur est donné sous la forme d'une commande de l'industrie "Surveillance d'une installation d'énergie renouvelable - mesure de la force du vent et du soleil ainsi que de la puissance électrique".

Ce document présente les différentes manières de produire et d'utiliser les données des stations météorologiques.

B. Interface Homme-Machine

I. Introduction

Dans ce chapitre, nous verrons comment créer une interface web pour une application de traitement de données.
Plusieurs langages seront mis en œuvre : HTML, Javascript, PHP, le but de ce cours n'est pas d'étudier ces langages en profondeur mais de comprendre leur utilisation à travers des exemples.
Les étudiants ont des bases de développement informatique : algorithmique de base ; connaissances de base des langages structurés.
Il sera préférable d'utiliser un ordinateur avec un système d'exploitation Linux, mais tous ces programmes peuvent être lancés sur un autre système d'exploitation avec des services web et python3.x

II. HTML

1. Definition

"HyperText Markup Language", généralement abrégé en HTML, est le langage de balisage conçu pour représenter les pages web. Il s'agit d'un langage pour écrire l'hypertexte, d'où son nom. Le HTML permet également la structuration et la mise en forme sémantique et logique du contenu des pages, ainsi que l'inclusion de ressources multimédia, notamment des images, des formulaires de saisie et des programmes informatiques. Il permet la création de documents interopérables avec une grande variété d'équipements, d'une manière conforme aux exigences d'accessibilité du web. "

2. Example

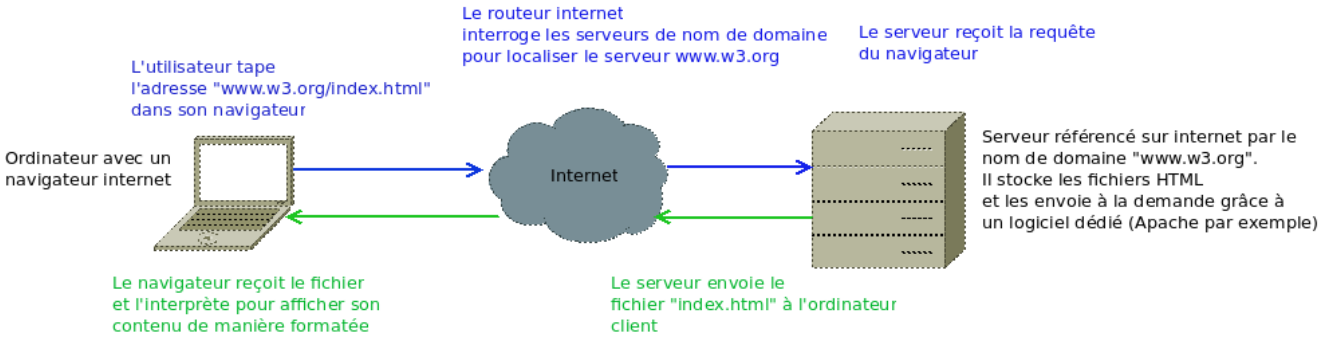
```
<DOCTYPE html>
<html lang="fr">
  <Head.
        <meta charset="utf-8" />
```


Co-funded by the Erasmus+ Programme of the European Union

```
                    <title>Page title displayed in the title bar</title>
            <meta name="generator" content="Geany 1.32" />
  </Head.
  <Body.
            C'est le corps de la page, ce texte est affiché lorsque le fichier est lu par le <b> navigateur
web</b>
            <Brr!
            <img src="./nsi.png" alt="ImageNsi" width="500", height="500">
            <Brr!
            <a href="http://www. tryat.eu">Link to the Erasmus - Tryat"</a> website.
  </Body.
</html>
```

- *La page est composée entre deux balises <head> et </head>.*
- *Les balises <meta> donnent des informations supplémentaires sur la page, par exemple l'auteur ou les mots clés qui seront lus par les moteurs de recherche pour référencer la page.*
- *Le contenu de la page qui sera affichée doit être placé entre les balises <body> et </body>.*
- *La balise <img> permet d'afficher une image dont l'emplacement du fichier doit être sonorisé, vous pouvez modifier sa taille et donner des informations qui seront affichées/lues à la place de l'image si le navigateur est configuré de cette manière (pour les personnes aveugles par exemple).*

3. Fonctionnement client-serveur



4. Exercices

a. *Fichier HTML à lecture directe*

- Écrivez un fichier HTML qui vous présente (nom, prénom, classe, photo), vous pouvez utiliser l'éditeur de texte " Geany " (éditeur de logiciel libre) et lui demander de générer un nouveau fichier selon le modèle " file.html ".
- Faites un lien vers le site " www.elektronslibres.fr ",
- Ouvrez la page avec votre navigateur web,
- Cliquez sur la page avec le bouton droit de la souris et regardez le code source

Co-funded by the Erasmus+ Programme of the European Union

b. Fichier déposé sur un serveur WEB

- Installez un serveur "Apache" sur votre ordinateur avec la commande "sudo apt-get install apache2" (Linux), ou installez WAMP sur un système d'exploitation Windows.
- Placez une page web à la racine de ce site avec le nom "index.html" avec la commande "sudo cp nom_fichier /var/www/html",
- Modifiez votre page web pour créer des liens avec vos camarades de classe.

III. Javascript

Le langage HTML permet très peu d'interaction entre l'utilisateur et la page web, c'est pourquoi très rapidement un autre langage a été développé, inclus dans les pages web qui offre des fonctionnalités plus avancées.

Ce langage, le Javascript, utilise le même mécanisme client-serveur que le HTML, il est directement écrit dans la page web et interprété par le navigateur.

"Javascript" est un langage de script principalement utilisé dans les pages web interactives mais aussi pour les serveurs2 avec l'utilisation (par exemple) de Node.js3.

Avec HTML et CSS, JavaScript est parfois considéré comme l'une des technologies de base du World Wide Web5. Le langage JavaScript permet de créer des pages web interactives et, à ce titre, il constitue une partie essentielle des applications web. Une grande majorité de sites web l'utilisent , et la plupart des navigateurs web disposent d'un moteur JavaScript dédié pour l'interpréter, indépendamment de toute considération de sécurité qui pourrait survenir. "

(source : wikipedia)

Ces dernières années, il s'est considérablement développé et fournit désormais un grand nombre de bibliothèques d'outils pour faciliter le développement de pages web interactives.

1. Premiers pas en Javascript

Modifiez votre page web en ajoutant cette portion de code dans la section <body> :

```
<script type= "text/javascript" >
>
        alert('Bienvenue sur la page de la station Météo Tryat');
</Script>
```

To add an interaction modify the code as follows:

```
<script type= "text/javascript" >
>
        var FirstNameName;
        FirstnameLastname=prompt("Donnez votre nom et votre prénom");
        alert("Bonjour "+FirstnameName);
</Script>
```

Vous remarquez qu'en Javascript, il est nécessaire de déclarer les variables.

Il est aussi possible de faire :

```
    function invite(FirstNameName)
{
        alert("Bonjour"+FirstnameName);
```


Co-funded by the Erasmus+ Programme of the European Union

```
}

var FirstNameName;
FirstnameLastname=prompt("Donnez votre nom et votre prénom");
invite(FirstnameName);
```

2. Bibliothèques

a. Bibliothèque personnelle

- sauvegarder la fonction que vous avez effectuée dans un fichier " myScripts.js ".
- l'appeler en modifiant le fichier principal comme suit :

```
<DOCTYPE html>
<html lang="fr">
  <Head.
        <meta charset="utf-8" />
            <title>Titre affiché dans la barre de la fenêtre</title>
        <meta name="generator" content="Geany 1.32" />
  </Head.
  <script type= "text/javascript" src="./mesScripts.js"></script>
  <Body.
        This is the body of the page, this text is displayed when the file is read by the <b> web browser</
b>
        <Brr!
        <img src="./nsi.png" alt="ImageNsi" width="500", height="500">
        <Brr!
        <script type= "text/javascript">
            var FirstNameName;
            FirstnameLastname=prompt("Give your first and last name");
            invite(FirstnameName);
        </Scripting.
        <a href="http://www.w3.org">Link to the W3 web standardization site"</a>
  </Body.
</html>
```

b. Bibliothèques externes

javascript peut utiliser des bibliothèques situées sur d'autres sites, par exemple :

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.8.0/Chart.js" integrity="sha256-
arMsf+3JJK2LoTGqxfnuJPFTU4hAK57MtIPdFpiHXOU=" crossorigin="anonymous">
</Scripting.
```

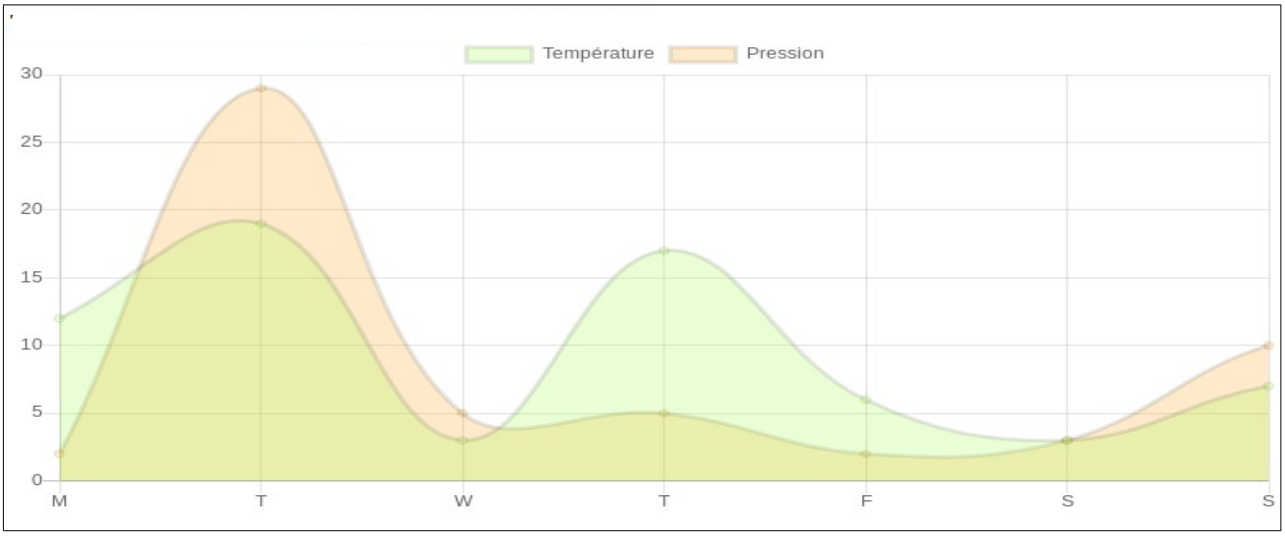
Ces bibliothèques contiennent des fonctions graphiques.
Dans ce cas, le test d'intégrité est conçu pour empêcher l'introduction de code malveillant dans la bibliothèque téléchargée sur Internet.

Co-funded by the Erasmus+ Programme of the European Union

Exemple d'utilisation d'une bibliothèque graphique externe

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.8.0/Chart.js" integrity="sha256-
arMsf+3JJK2LoTGqxfnuJPFTU4hAK57MtIPdFpiHXOU=" crossorigin="anonymous"></script>
<Body.
  Chart.js" test for CLF teaching
<Canvas id="myChart"> canvas>
<Script
  var ctx = document.getElementById('myChart').getContext('2d');
  var myChart = new Chart(ctx,
  {
   type: 'line',
   data:
   {
            labels: ['M', 'T', 'W', 'T', 'F', 'S', 'S'],
            datasets:
            [
                     {
                      label: 'Temperature',
                      data: [12, 19, 3, 17, 6, 3, 7, 8],
                      backgroundColor: "rgba(153,255,51,0.2)"
                     },
                     {
                      label: 'Pressure',
                      data: [2, 29, 5, 5, 5, 2, 3, 10, 12],
                      backgroundColor: "rgba(255,153,0,0.0.2)"
                     }
            ]
   }
  });
</Scripting.
</Body.
```

Le rendu va être :



Co-funded by the Erasmus+ Programme of the European Union

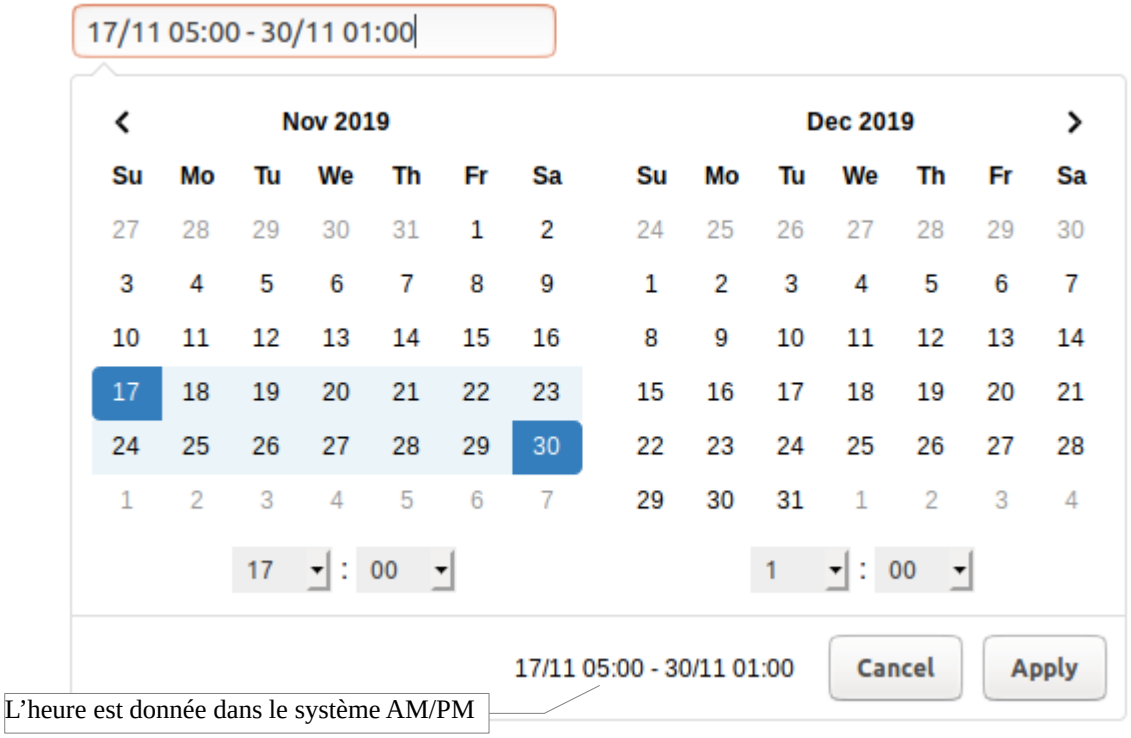
Exemple d'un sélecteur de dates

Cette fonction est utilisée pour obtenir les valeurs d'une date.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

<Head.
  <"My Picker Period" title.
  <meta http-equiv="content-type" content="text/html;charset=utf-8" />
  <meta name="generator" content="Geany 1.32" />
</Head.

<script type="text/javascript" src="https://cdn.jsdelivr.net/jquery/latest/jquery.min.js"></script>
<script type="text/javascript" src="https://cdn.jsdelivr.net/momentjs/latest/moment.min.js"></script>
<script type="text/javascript"
src="https://cdn.jsdelivr.net/npm/daterangepicker/daterangepicker.min.js"></script>
<link rel="stylesheet" type="text/css"
href="https://cdn.jsdelivr.net/npm/daterangepicker/daterangepicker.css" />

<Body.
  <input type="text" name="datetimes" />

  <Script
  $(function() {
   $('input [name="datetimes"]').daterangepicker({
              timePicker: true,
              "timePicker24Hour": true,
              startDate: moment().startOf('hour'),
              endDate: moment().startOf('hour').add(32, 'hour'),
              local: {
               format: 'DD/MM/YYY hh:mm'.
              }
   });
  });
  </Scripting.
</Body.

</html>
```


Co-funded by the Erasmus+ Programme of the European Union

Rendering:



Nous essayons maintenant d'analyser la variable renvoyée :
Nativement, la bibliothèque attribue la valeur retournée par l'opérateur au champ "text" référencé par le nom "datetimes". Une modification du html et du javascript dans la section "Body" permet de vérifier cela :

```
<Body.
  <form method="post" action=" RecupFormulaire.php">
        <p>Please choose a period: </p>
        <input type="text" name="datetimes" id="datetimes" onchange="Verif()"/>
        <input type="text" name="datesLues" id="datesLues"/>
        <Brr!
  <input type="submit" value="Display" />

  <Script
        Verif() function
        {
              var dates = document.getElementById("datetimes").value;
              document.getElementById("datesLues").value=dates;
        }
        ...
```

Dans ce cas, en appuyant sur le bouton "Afficher", on appelle la page "traitement.php" qui sera générée par le serveur web et qui recevra comme paramètres les variables de formulaire "datetimes" et "datesLues".
De plus, la fonction javascript "Verif()" assignera le contenu de "datetimes" au champ "datesRead".

Co-funded by the Erasmus+ Programme of the European Union

IV. Le PHP

« PHP est un langage de script polyvalent particulièrement adapté au développement web. Le code PHP est généralement traité sur un serveur web par un interpréteur PHP mis en œuvre sous la forme d'un module, d'un démon ou d'un exécutable CGI (Common Gateway Interface). Sur un serveur web, le résultat du code PHP interprété et exécuté - qui peut être n'importe quel type de données, telles que des données d'images binaires ou HTML générées - constituerait la totalité ou une partie d'une réponse HTTP. »

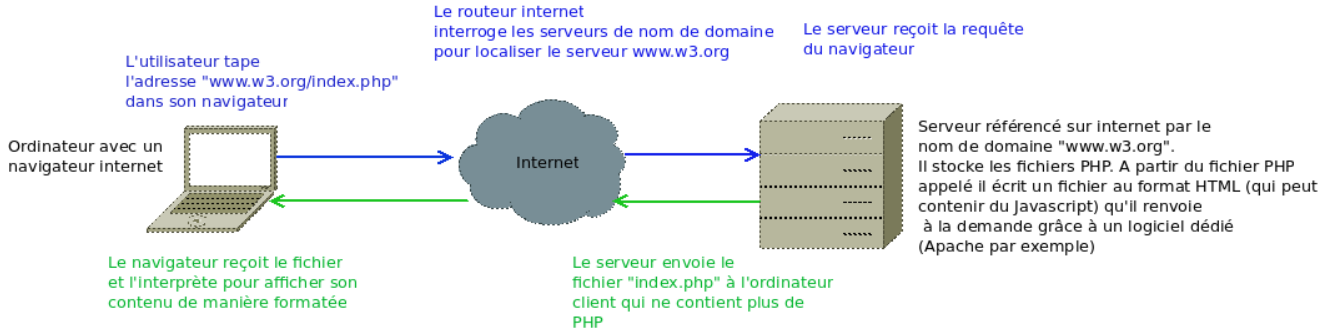
(source - Wikipedia)

1. Opération Client-serveur

a. Principe

Se référer page 5 « Opérations Client-serveur ».

b. Exemple



Code source en php de la page " page1.php ":

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

<Head.
  <title> untitled>
  <meta http-equiv="content-type" content="text/html;charset=utf-8" />
  <meta name="generator" content="Geany 1.32" />
</Head.

<Body.
  <?php
        echo("This text will appear in the body of the page <br>");
        $a=2;
        $b=5;
        echo("The result of the addition is ");
        echo($a+$b);
  ?>
</Body.
</html>
```


Co-funded by the Erasmus+ Programme of the European Union

Ce programme est déposé dans l'arborescence " /var/www/html " du serveur (qui peut être votre ordinateur). Avec le navigateur Internet, vous appelez la page : " http://localhost/page1.php ", elle s'affiche en retour :

```
This text will appear in the body of the page
The result of the addition is 7
```

et le code source de la page reçue par le navigateur web est :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<Head.
 <title> untitled>
 <meta http-equiv="content-type" content="text/html;charset=utf-8" />
 <meta name="generator" content="Geany 1.32" />
</Head.
<Body.
This text will appear in the body of the page <br>The result of the addition is 7
</Body.
</html>
```

→ on remarque qu'il n'y a plus d'instructions en PHP, elles ont été interprétées par le serveur PHP et leur résultat a été codé en HTML et renvoyé au navigateur.

2. Applications

a. Réception des données saisies en JavaScript

Les données sont envoyées au serveur web avec la méthode "POST" et PHP doit les récupérer avec la même méthode :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

<Head.
  <Title> Data       retrieval. Title>
  <meta http-equiv="content-type" content="text/html;charset=utf-8" />
  <meta name="generator" content="Geany 1.32" />
</Head.

<Body.
<?php
            $value=$_POST["datetimes"];
            echo "<hr>";
            echo "<br> Collect value: ".$value;
?>
</Body.
</html>
```

La chaîne est maintenant stockée dans la variable $value, et sera affichée en un seul morceau.
Vous devez en extraire les heures et les dates afin de pouvoir ensuite lancer une requête dans la base

Co-funded by the Erasmus+ Programme of the European Union

de données MySql qui héberge les données météorologiques. Deux possibilités, soit le faire en JavaScript avant d'envoyer le formulaire, soit le faire en PHP après avoir reçu les données.

Prétraitement JavaScript

Le résultat est obtenu en utilisant la méthode "split" appliquée à la variable contenant la période sélectionnée :

```
<Body.
  <form method="post" action="RecupFormulaire.php">
        <p>Please choose a period: </p>
        <input type="text" name="datetimes" id="datetimes" onchange="Verif()"/>
        <input type="hidden" name="datesLues" id="datesLues"/>
        <input type="hidden" name="dateDebut" id="dateDebut"/>
        <input type="hidden" name="dateFin" id="dateFin"/>
        <input type="hidden" name="starttime" id="starttime"/>
        <input type="hidden" name="endtime" id="endtime"/>

        <Brr!
  <input type="submit" value="Display" />

  <Script
        Verif() function
        {
              var dates = document.getElementById("datetimes").value;
              // format: 22/12/19 04:00 - 02/01/20 12:00
              document.getElementById("datesLues").value=dates;
              detail=dates.split(" ");
              document.getElementById("dateDebut").value=detail[0];
              document.getElementById("dateFin").value=detail[3];
              document.getElementById("starttime").value=detail[1];
              document.getElementById("endtime").value=detail[4];
        }
```

Remarques :

- plusieurs champs ont été remplacés par "hidden" au lieu de "text" pour éviter de surcharger inutilement la page web.
- La méthode de découpage stocke dans un tableau le résultat de l'opération, chaque élément correspond à un élément trouvé séparé par le caractère espace : " ".

Ensuite, vous devez modifier la page PHP comme suit :

```
<?php
  $value=$_POST["datetimes"];
  $dateDebut=$_POST["dateDebut"];
  $dateFin=$_POST["dateFin"];
  $HourStart=$_POST["hourStart"];
  $fin-hour=$_POST["fin-hour"];
  echo "<hr>";
  echo "<br> Collect value: ".$value;
```


Co-funded by the Erasmus+ Programme of the European Union

```
  echo "<br> Start Date: ".$dateDebut;
  echo "<br> End date : ".$end date;
  echo "<br> Start time: ".$starttime;
  echo "<br> End time : "...hourFin;
?>
```

Post-traitement en PHP

L'équivalent du PHP sera la fonction "explode" :

```
<?php
  $value=$_POST["datetimes"];
  $detail=explode(" ",$value);
  $dateDebut=$detail[0];
  $dateFin=$detail [3];
  $hourDebut=$detail[1];
  $hourEnd=$detail [4];
  echo "<hr>";
  echo "<br> Collect value: ".$value;
  echo "<br> Start Date: ".$dateDebut;
  echo "<br> End date : ".$end date;
  echo "<br> Start time: ".$starttime;
  echo "<br> End time : "...hourFin;
?>
```

Dans ce cas, le traitement de la chaîne sera effectué par le serveur Internet et non par le navigateur du client, qu'est ce qui est préférable ?

b. Accès aux données stockées sur un serveur MySql

Les données auxquelles nous voulons accéder sont automatiquement stockées dans une table dans une base de données MySql hébergée sur un serveur MySql accessible sur Internet. Le format des données est le suivant :

	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut
☐	1	**date**	int(11)			Non	*Aucun(e)*
☐	2	**direction**	text	utf8_general_ci		Non	*Aucun(e)*
☐	3	**ventMoyen**	float			Non	*Aucun(e)*
☐	4	**ventMax**	float			Non	*Aucun(e)*
☐	5	**pluvio**	float			Non	*Aucun(e)*
☐	6	**temperature**	float			Non	*Aucun(e)*
☐	7	**pression**	float			Non	*Aucun(e)*
☐	8	**humidite**	float			Non	*Aucun(e)*

Timestamp

La date est stockée sous forme d'un "timestamp" ↔ nombre de secondes écoulées depuis le 01/01/1970

En PHP, la fonction " $timestampDebut=strtotime($dateDebut." ".$timeDebut.":00"); " permet d'effectuer cette conversion à partir d'une analyse de chaîne de caractères. Une autre fonction "mktime" effectue également cette transformation de manière plus rigoureuse, d'où la fonction :

```
function calcStamp($dateTxt,$timeTxt)
{
  //format: 09/12/19 12:00:00
  $detailDate=explode("/",$dateTxt);
  $detailHour=explode(":",$hourTxt);
```


Co-funded by the Erasmus+ Programme of the European Union

```
    return(mktime($detailHour[0],$detailHour[1],$detailDate[1],$detailDate[0],$detailDate[2]);
}
```

Qui va rendre le timestamp.

En JavaScript, le timestamp est compté en millisecondes, la valeur retournée doit donc être divisée par 1 000.

Accès à la base de données Mysql

L'objectif sera de tracer la courbe de la vitesse du vent en fonction du temps et de deux dates saisies par l'utilisateur.

Dans un premier temps, pour tester l'accès à la base de données, nous lancerons une requête pour obtenir les dates minimales et maximales dans le tableau "acquisitions".

```
// access to the weather database
$tabDate=array();
$dateMin=time(); //today
dateMax=0; // January 1, 1970
try
{
// One connects to MySQL by providing the address, the name of the database, the identifier and the password.
        $bdd = new PDO('mysql:host=srv36.haisoft.net;dbname=C07400_meteo;charset=utf8', 'cnp2019', 'Suf767!Hr');
}
wrestling(Exception $e)
{
        // If an error occurs, a message is displayed and everything is stopped.
        die('Error: '.$e->getMessage());
}
// we retrieve all the dates from the acquisition table
$response = $bdd->query('SELECT date FROM acquisitions');
while($data = $response->fetch())
{
        $tabDate[]=$data['date'];
        if($dateMin>$data['date']) $dateMin=$data['date'];
        if($dateMax<$data['date']) $dateMax=$data['date'];
}
reply->closeCursor(); // Ends processing of the query
// Displaying the data :
echo "<br>Minimum date: ".$dateMin." / ".date('d/m/Y H:i:s', $dateMin);
echo "<br>Maximum date: ".$dateMax." / ".date('d/m/Y H:i:s', $dateMax);
```

Il est préférable de connaître ces dates avant de choisir, en JavaScript, les dates pour lesquelles vous souhaitez obtenir les mesures. Il est donc nécessaire de modifier la page dans laquelle se trouve le script du calendrier. Elle deviendra donc une page de type "PHP" et contiendra trois langages : html/javascript/php et il suffira d'y insérer le code PHP.

Co-funded by the Erasmus+ Programme of the European Union

Lecture des données sur le vent

La page appelée exécutera la requête dans la base de données, le code sera le suivant :

```
// we retrieve all the dates from the acquisition table
  $request='SELECT date,windMean FROM acquisitions WHERE date >= '.$tmpMin.' and date <= '.
$tmpMax.';';
  $response = $bdd->query;
  while($data = $response->fetch())
  {
            $tabDate[]=$data['date'];
            $tabVent[]=$data['averageVent'];
            if($dateMin>$data['date']) $dateMin=$data['date'];
            if($dateMax<$data['date']) $dateMax=$data['date'];
  }
  reply->closeCursor(); // Ends processing of the query
  // Displaying the data :
  echo "<ul>";
  echo "<li> Selected range: ".$value."</li>";
            echo "<li>Minimum date: ".date('d/m/Y H:i:s', $dateMin)."</li>";
            echo "<li>Maximum date: ".date('d/m/Y H:i:s', $dateMax)."</li>";
            echo "<li> Number of measurements: ".count($tabDate)."</li>";
  echo "</ul>";
  echo "<hr>";
  for ($i=0;$i<count($tabDate);$i++)
  {
            echo("<br>".date('d/m/y h:m',$tabDate[$i])." ".$tabVent[$i]);
  }
```

c. Affichage des statistiques météorologiques

La seule chose qui reste à faire est de représenter ces données avec la bibliothèque graphique "JavaScript/Charts.js", qui donne le code suivant :

```
<Script
  var ctx = document.getElementById('myChart').getContext('2d');
  var myChart = new Chart(ctx,
  {
   type: 'line',
   data:
   {
            <?php
                     echo "labels: [";
                     for ($i=0;$i<count($tabDate);$i++)
                     {
                              echo("'".date('d/m/y h:m',$tabDate[$i])."',");
                     }
                     echo "],";
            ?>
```


Co-funded by the Erasmus+ Programme of the European Union

```
            datasets:
            [
                        {
                         label: 'Vent',
            <?php
                        echo "data: [";
                        for ($i=0;$i<count($tabDate);$i++)
                        {
                                    echo($tabVent[$i].",");
                        }
                        echo "],";
            ?>
                         backgroundColor: "rgba(255,153,0,0.0.2)"
                        }
            ]
  }
  });
</Scripting.
```

C. Interruptions matérielles

"En informatique, une interruption est une suspension temporaire de l'exécution d'un programme informatique par le microprocesseur afin d'exécuter un programme prioritaire (appelé service d'interruption). "(Wikipedia)

I. Prérequis

- Connaissances de base en Arduino,
- Algorithmique,
- Fonctions C-Arduino
- Échantillonnage

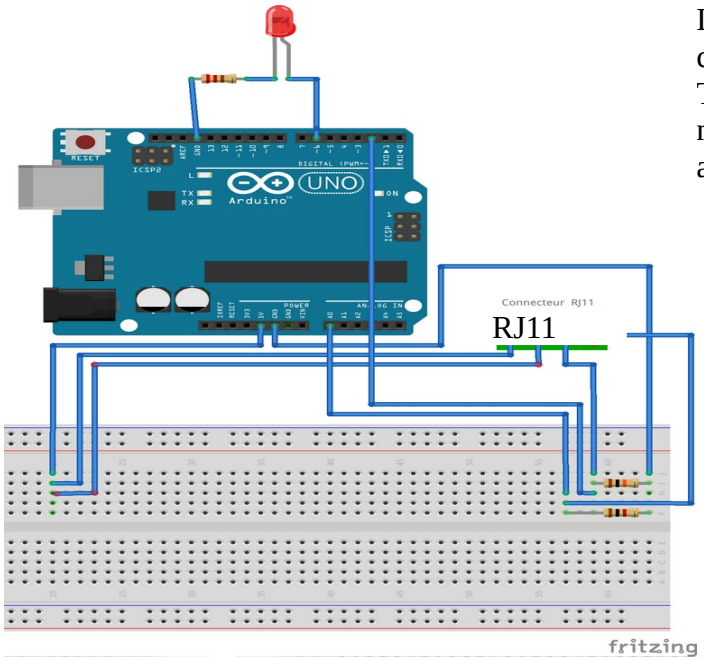
Source L.V

II. Matériel

- 1 microcontrôleur Arduino avec son câble USB et l'IDE installé sur un ordinateur client,
- Fils de test pour la première partie, puis Leds et résistances
- 1 station météorologique pour la deuxième partie (anémomètre et girouette Silver Data Systems). Vous pouvez l'obtenir ici :
- « https://www.robotshop.com/eu/fr/kit-station-meteorologique.html »

Co-funded by the Erasmus+ Programme of the European Union

III. Assemblage final



Le système météorologique utilisé comprend une girouette et un anémomètre. Tous deux utilisent des interrupteurs magnétiques ILS qui se ferment lorsqu'un aimant passe à proximité :

IV. Objectif

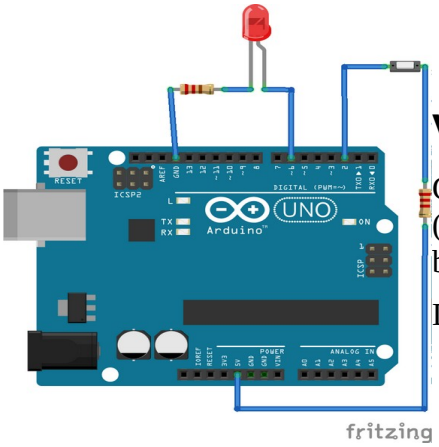
En comptant les interruptions envoyées par l'anémomètre, évaluez la vitesse du vent.
Nous allons d'abord déclencher les interruptions avec un simple fil et lancer un programme de test, puis nous mesurerons la vitesse du vent.

V. Principe d'une interruption

L'Arduino a deux broches (2 et 3) qui peuvent transmettre des interruptions. Chaque fois que ces broches reçoivent un changement d'état, l'Arduino en est informé et démarre une fonction particulière. Les états interprétés peuvent être LOW, CHANGE, RISING, FALLING.

Leur déclaration à l'IDE se fait de la manière suivante :

```
pinMode(n°broche, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(n°broche), NomFonction, CHANGE);
```



VI. Assemblage

Ce montage simple utilise une LED connectée à la broche 2 (interruption 0) ainsi qu'un simple bouton poussoir connecté à la broche 6.

De l'Arduino :

- La broche 2 est une entrée ☐ sortie ☐
- La broche 6 est une entrée ☐sortie ☐

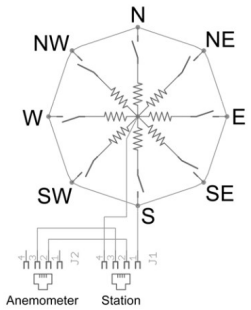
Co-funded by the Erasmus+ Programme of the European Union

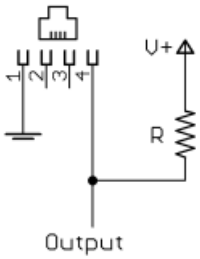
`Application`

- Câblez cet ensemble et réalisez un programme Arduino dont la LED changera d'état à chaque impulsion exercée sur le bouton-poussoir.

VII. Montage avec l'anémomètre

Le système de mesure du vent (vitesse et orientation) est connecté à une prise RJ11, dont seuls les 4 fils centraux sont utilisés. Ils sont connectés à l'ILS seul pour l'anémomètre et en série avec des résistances pour la girouette, ce qui permet de faire un pont diviseur de tension :





Direction (Degrees)	Resistance (Ohms)	Voltage (V=5v, R=10k)
0	33k	3.84v
22.5	6.57k	1.98v
45	8.2k	2.25v
67.5	891	0.41v
90	1k	0.45v
112.5	688	0.32v
135	2.2k	0.90v
157.5	1.41k	0.62v
180	3.9k	1.40v
202.5	3.14k	1.19v
225	16k	3.08v
247.5	14.12k	2.93v
270	120k	4.62v
292.5	42.12k	4.04v
315	64.9k	4.78v
337.5	21.88k	3.43v

source : Argent Data Systems

Nous allons connecter les broches 2 et 3 à la terre et l'entrée 2 pour déclencher les interruptions. La rotation de l'anémomètre déclenche la LED deux fois par tour, le fabricant donne cette information :

« Un vent de 2,4 km/h (1 492 MPH) provoque la fermeture de l'interrupteur une fois par seconde. »

Vérifier le bon fonctionnement de l'assemblage.

VIII. Utilisation de l'horloge

" *millis() : Retourne le nombre de millisecondes écoulées depuis que le conseil d'administration d'Arduino a commencé à exécuter le programme actuel. Ce nombre va déborder (revenir à zéro), après environ 50 jours.* »

- Changez la fonction appelée sur l'interruption pour incrémenter une variable agissant comme un compteur,
- Après 4s (assez longtemps pour lisser l'erreur), calculez et affichez la vitesse du vent sur la console.

IX. Montage avec la girouette

Effectuez le montage avec la girouette.

- Nous cherchons où se trouve le nord sur la girouette, quelle tension correspond à la valeur 0° ?
- L'échantillonnage est sur 10 bits, quelle est la valeur décimale pleine échelle correspondant à la valeur analogique de 5 V ?
- Nous pouvons en déduire quelle valeur sera donnée par l'échantillonneur pour 0°.
- Modifiez le programme pour que le terminal série affiche les différentes valeurs lues sur le port A0 et les compare avec celles données par le tableau.

Co-funded by the Erasmus+ Programme of the European Union

D. Bus I2C

I2C : Inter Integrated Circuit

Ce bus a été développé au début des années 80 par "Philips Semiconductor" pour permettre de connecter facilement les différents circuits dans le domaine de la domotique à un microprocesseur.

Le but était de faire communiquer entre eux des composants électroniques très différents grâce à seulement 3 fils :

- Signal de données : SDA
- Signal d'horloge : SCL
- Signal électrique de référence : la terre
- Un quatrième fil alimentera l'appareil.

Les échanges ont toujours lieu entre un seul maître et un (ou tous) esclave(s), toujours à l'initiative du maître (jamais de maître à maître ou d'esclave à esclave) et chaque périphérique a une adresse unique.

I. Prérequis

- Connaissances de base en Arduino,
- Algorithmique,
- Fonctions C-Arduino
- Échantillonnage

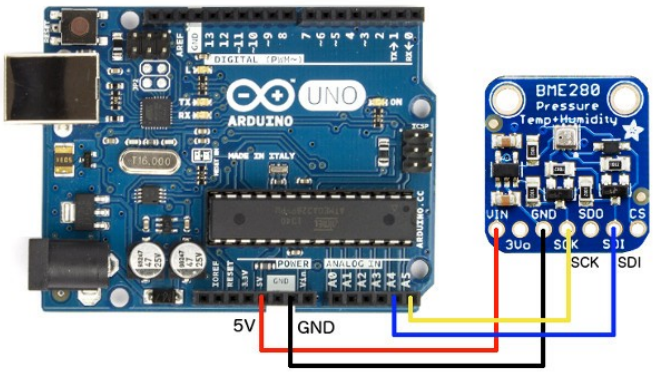
II. Matériel

- microcontrôleur Arduino avec son câble et l'IDE installé sur un ordinateur client,
- 1 appareil I2C (BME 280 pour cette étude), cette carte dispose de trois capteurs : température, pression et humidité.

III. Assembly

En raison de sa conception, l'assemblage est extrêmement simple :

- Vin → 5 V
- Gnd
- SDI (SDA) → A4
- SCK (SCL) → A5



IV. Objectif

Le capteur étudié renvoie les trois valeurs suivantes : température, humidité, pression. L'objectif sera de les lire afin d'intégrer les données dans un site de station météorologique.

V. Détection des périphériques I2C

L'adresse est codée sur 7 bits, la bibliothèque Arduino Wire permet d'utiliser le port série. Le

Co-funded by the Erasmus+ Programme of the European Union

programme suivant enverra une requête à tous les appareils possibles (1→ 126) et affichera les adresses de ceux qui répondent.

```
#include <Wire.h>

void setup()
{
 Wire.begin(); //connection initialisation
 Serial.begin(9600);
 Serial.println("\n I2C Device detector ");
}

void loop()
{
 byte erreur, adresse;
 int nPeripheriques=0;
 Serial.println("Scanning the different addresses...");

 for(adresse = 1; adresse < 127; adresse++ ) // we're running down every possible address
 {
  Wire.beginTransmission(adresse);
  erreur = Wire.endTransmission(); //does the device respond?
  if (erreur == 0)
  {
   Serial.print("I2C device detected address 0x");
   if (adresse<16)
    Serial.print("0");
   Serial.print(adresse,HEX);
   Serial.println("  !");
   nPeripheriques++;
  }
  else if (erreur==4)
  {
   Serial.print("Unknown error at address 0x");
   if (adresse<16)
    Serial.print("0");
   Serial.println(adresse,HEX);
  }
 }
 if (nPeripheriques == 0)
  Serial.println("No devices detected.\n");
 else
  Serial.println("Scan completed \n");
 delay(5000);
}
```

Le programme renvoie ces informations au moniteur en série :

- *Détecteur de dispositif I2C*

Co-funded by the Erasmus+ Programme of the European Union

- *Scanner les différentes adresses…*
- *Le dispositif I2C a détecté l'adresse 0x77 !*
- *Balayage terminé*

VI. Lecture des valeurs mesurées

La fiche technique du volet donne cette information :" Les données sont lues dans un format non signé de 20 bits pour la pression et la température et dans un format non signé de 16 bits pour l'humidité. »

Le tableau ci-dessous donne les adresses des registres de température :

Register Name	Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state
hum_lsb	0xFE	hum_lsb<7:0>								0x00
hum_msb	0xFD	hum_msb<7:0>								0x80
temp_xlsb	0xFC	temp_xlsb<7:4>				0	0	0	0	0x00
temp_lsb	0xFB	temp_lsb<7:0>								0x00
temp_msb	0xFA	temp_msb<7:0>								0x80
press_xlsb	0xF9	press_xlsb<7:4>				0	0	0	0	0x00
press_lsb	0xF8	press_lsb<7:0>								0x00
press_msb	0xF7	press_msb<7:0>								0x80
config	0xF5	t_sb[2:0]				filter[2:0]			spi3w_en[0]	0x00
ctrl_meas	0xF4	osrs_t[2:0]				osrs_p[2:0]		mode[1:0]		0x00
status	0xF3					measuring[0]			im_update[0]	0x00
ctrl_hum	0xF2						osrs_h[2:0]			0x00
calib26..calib41	0xE1…0xF0	calibration data								individual
reset	0xE0	reset[7:0]								0x00
id	0xD0	chip_id[7:0]								0x60
calib00..calib25	0x88…0xA1	calibration data								individual

Registers:	Reserved registers	Calibration data	Control registers	Data registers	Status registers	Chip ID	Reset
Type:	do not change	read only	read / write	read only	read only	read only	write only

```
#include "BME280I2C.h"
#include <Wire.h>
#define SERIAL_BAUD 9600 // better than the original value
BME280I2C bme;    // Default : forced mode, standby time = 1000 ms
                  // Oversampling = pressure ×1, temperature ×1, humidity ×1, filter off,
const byte ledPin = 6;
const byte interruptPin = 2;// caution 1 is attached to pin 3 and 0 to pin 2
volatile byte state = LOW;

void setup()
{
 Serial.begin(SERIAL_BAUD);
 while(!Serial) {} // Wait
 Wire.begin();

 while(!bme.begin())
 {
  Serial.println("Could not find BME280 sensor!");
  delay(1000);
 }
```


Co-funded by the Erasmus+ Programme of the European Union

```
 pinMode(ledPin, OUTPUT);
 pinMode(interruptPin, INPUT_PULLUP);
}

void loop()
{
  readBme();
  delay(10000) ;

void readBme() // to get the data from the three sensors
{
  float temp(NAN), hum(NAN), pres(NAN);
  BME280::TempUnit tempUnit(BME280::TempUnit_Celsius);
  BME280::PresUnit presUnit(BME280::PresUnit_Pa);
  bme.read(pres, temp, hum, tempUnit, presUnit);
  Serial.print(" Pressure : ");
  Serial.println(pres);
  Serial.print(" Humidity : ");
  Serial.println(hum);
  Serial.print(" Temperature : ");
  Serial.println(temp);
}
```

→ *vous pouvez alors obtenir ces données via le port série d'un ordinateur.*

E. Envoi des données des stations météorologiques

Dans ce chapitre, nous allons décrire la connexion entre la station météorologique et la base de données MySql

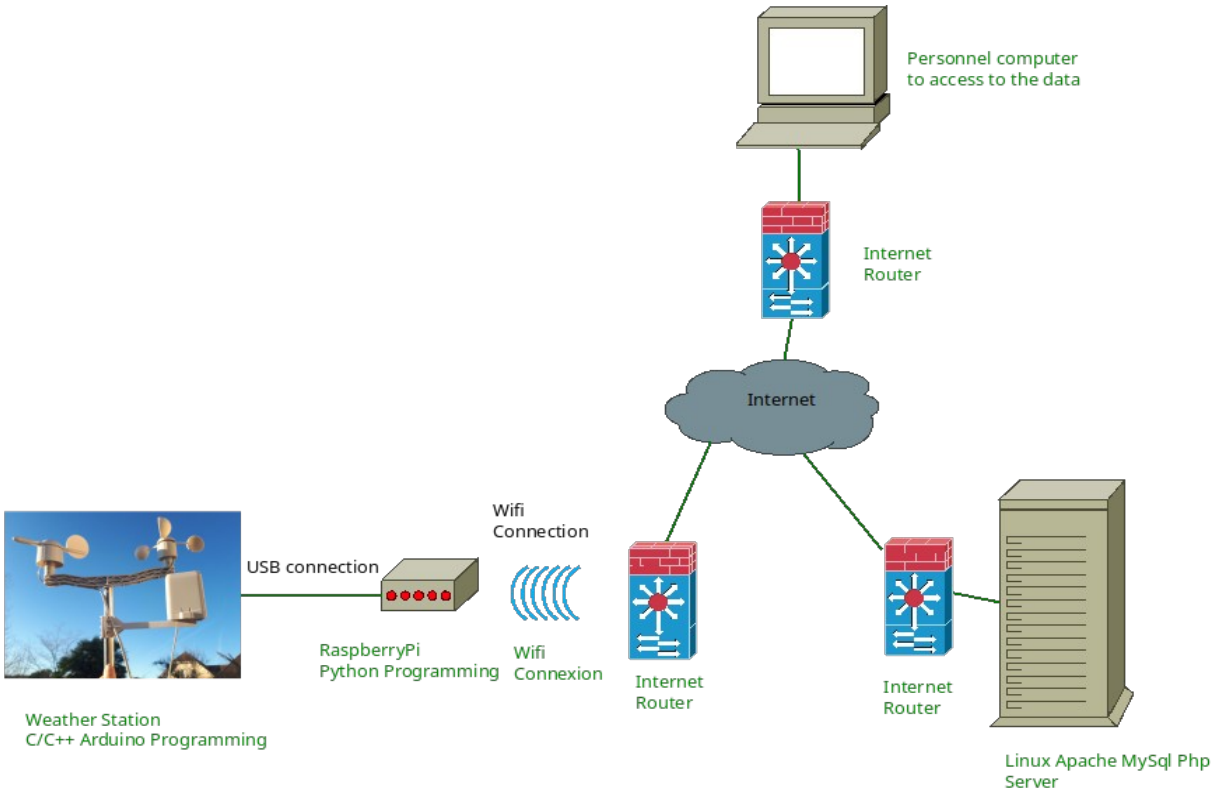
I. Principe

La station météorologique, avec la carte arduino, collectera les données des capteurs. Elle les enverra ensuite à un ordinateur (un RaspberryPi par exemple) par le biais du port série USB.

En utilisant une connexion Internet, l'ordinateur enverra ces données au serveur de stockage.

La description du matériel est décrite dans les autres chapitres ci-dessus (C, D)

Co-funded by the Erasmus+ Programme of the European Union



II. Code Arduino

Voici le code complet sur la carte Arduino pour collecter toutes les données de la station météo

```
#include "BME280I2C.h"
#include <Wire.h>

#define SERIAL_BAUD 9600 // better than the original value

BME280I2C bme;    // Default : forced mode, standby time = 1000 ms
                  // Oversampling = pressure ×1, temperature ×1, humidity ×1, filter off,


const byte ledPin = 6;
const byte interruptPin = 2;//  caution 1 is attached to pin 3 and 0 to pin 2
volatile byte state = LOW;
float tempsInitial, temps, vitesseVent;
int compteur;
int directionVent;


/*
*** pin choice : ***
Arduino d2 : Press Button
```


Co-funded by the Erasmus+ Programme of the European Union

```
Arduino d6 : Led
*** angles values for the  prototype ***
0 : 246
45 : 134
90 : 76
135 : 392
180 : 736
225 : 839
270 : 930
315 : 561
*/

void setup()
{
 Serial.begin(SERIAL_BAUD);
 while(!Serial) {} // Wait
 Wire.begin();

 while(!bme.begin())
 {
  Serial.println("Could not find BME280 sensor!");
  delay(1000);
 }

 pinMode(ledPin, OUTPUT);
 pinMode(interruptPin, INPUT_PULLUP);
 attachInterrupt(digitalPinToInterrupt(interruptPin), allumeLed, CHANGE); // pin get low
 digitalWrite(ledPin, HIGH);
 delay(1000);
 digitalWrite(ledPin, LOW);
 tempsInitial=millis();
 compteur=0;

}

void loop()
{
 digitalWrite(ledPin, state);
 if((millis()-tempsInitial)>4000)// we count interruptions during 4s
 {
  tempsInitial=millis();
  vitesseVent=(compteur/4)*2.4;
  compteur=0;
  lectureBme();
  Serial.print("cnp:");
  Serial.println(vitesseVent);
```


Co-funded by the Erasmus+ Programme of the European Union

```
   directionVent = analogRead(A0);
   Serial.print(" Wind Direction : ");// 0° correspond to 786 proportional law

   Serial.println(directionVent);

  }

}

void allumeLed()
{
   //Serial.print(" Interruption detected, time laps : ");
   //Serial.println(-millis()-tempsInitial);
   state = !state;
   compteur++;
}

void lectureBme()
{
  float temp(NAN), hum(NAN), pres(NAN);
  BME280::TempUnit tempUnit(BME280::TempUnit_Celsius);
  BME280::PresUnit presUnit(BME280::PresUnit_Pa);
  bme.read(pres, temp, hum, tempUnit, presUnit);
  Serial.print(" Pressure : ");
  Serial.println(pres);
  Serial.print(" Humidity : ");
  Serial.println(hum);
  Serial.print(" Temperature : ");
  Serial.println(temp);
}
```

III. Code Python sur le Raspberry-Pi

Il nous faut maintenant un programme sur l'ordinateur pour faire passer les données de la carte arduino par le port série usb :

```
# -*- coding: utf-8 -*-
import serial
import string
import time
import pymysql
import requests



mysqlhost = "*******" # put the server adress there
mysqluser = "*******" # the user login
mysqlpass = "*******" # the mysql user's password
datenbank = "*******" # the name from the database
```


Co-funded by the Erasmus+ Programme of the European Union

```
class meteo:
    def _init_(self):
        self.pluie
        self.temperature
        self.humidite
        self.pression
        self.direction
        self.ventMoyen
        self.ventMax
acquis=meteo

ser = serial.Serial('/dev/ttyACM0', 9600)
print("Serial port  initialised :")
# print(ser)
# you can read the value on the arduino IDE 'tool/serial port'
#File Methode : read file and send characters
print("Start of the data retrieval...")
valeurs=[]
dictValeurs={}
while(1):
    print("Wiating for the data...")
    donnee=ser.readline()
    if(donnee) :
        print("Data retrieved")
        donneeUTF8=donnee.decode("utf8")
        print(donneeUTF8) # you can see the data on a Linux computer with the console instruction : sudo
screen /dev/ttyACM0
        detail=donneeUTF8.split(";")
        if(detail[0]=="cnp") :
            print("Data validated")
            acquis.pluie=str(detail[1])
            acquis.temperature=str(detail[2])
            acquis.humidite=str(detail[3])
            acquis.pression=str(detail[4])
            #acquis.direction=str(detail[5])
            # conversion direction
            detail[5]=int(detail[5])
            if(detail[5]>20 and detail[5]<35)  : acquis.direction ="O" # about 27
            if(detail[5]>90 and detail[5]<120) : acquis.direction ="NO" # about  107
            if(detail[5]>490 and detail[5]<530) : acquis.direction ="N" # about  505
            if(detail[5]>290 and detail[5]<330) : acquis.direction ="NE" # about  315
            if(detail[5]>190 and detail[5]<220) : acquis.direction ="E" # about  204
            if(detail[5]> 40 and detail[5]<60) : acquis.direction ="SE" # about  57
            if(detail[5]> 0 and detail[5]<9) : acquis.direction ="S" # about  6
```


Co-funded by the Erasmus+ Programme of the European Union

```
        if(detail[5]> 8 and detail[5]<18) : acquis.direction ="SW" # about  13

        acquis.ventMoyen=str(detail[6])
        acquis.ventMax=str(detail[7])
        print("N rain interrupt : "+str( acquis.pluie))
        print("Temperature    : "+str(acquis.temperature)+" °C")
        print("Humidity       : "+str(acquis.humidite)+" %")
        print("Pressure       : "+str(acquis.pression)+" mBar")
        print("Wind direction : "+str(acquis.direction)+" °")
        print("Wind average speed   : "+str(acquis.ventMoyen)+" km/h")
        print("wind maximal speed   : "+str(acquis.ventMax)+" km/h")
        #connexion = pymysql.connect( host=mysqlhost, user=mysqluser, passwd=mysqlpass,
db=datenbank, charset='utf8' )
        # http request
        # r = requests.get(adresse+"/ecriture.php?
temp="+acquis.temperature+"&hum="+acquis.humidite+"&pres="+acquis.pression+"&direc="+acquis.
direction+"&vit="+acquis.vitesse)
        # data writing
        connexion = pymysql.connect( host=mysqlhost, user=mysqluser, passwd=mysqlpass, db=datenbank,
charset='utf8' )
        sql = """INSERT INTO acquisitions(date,direction,humidite,pluvio,pression,temperature,ventMoyen,
ventMax) VALUES ("""+str(time.time()+7200)+""","""+str(acquis.direction)
+""","""+str(acquis.humidite)+""","""+str(acquis.pluie)+""","""+str(acquis.pression)
+""","""+str(acquis.temperature)+""","""+str(acquis.ventMoyen)+""","""+str(acquis.ventMax)
+""")"""
        cur = connexion.cursor(pymysql.cursors.DictCursor)
        cur.execute( sql, ())
        result = cur.fetchall()# return to a dictionnary
        connexion.commit()
        connexion.close()
# You can read the data that you writed :
        # data display
        connexion = pymysql.connect( host=mysqlhost, user=mysqluser, passwd=mysqlpass, db=datenbank,
charset='utf8' )
        sql = """SELECT * FROM acquisitions;"""
        cur = connexion.cursor(pymysql.cursors.DictCursor)
        cur.execute( sql, ())
        result = cur.fetchall()#  return to a dictionnary
        connexion.commit()
        connexion.close()
        dictValeurs=result
        #print(dictValeurs)
```

F. Internet des objets

« *L'internet des objets (IoT) décrit le réseau d'objets physiques - les "choses" - qui sont intégrés à des capteurs, des logiciels et d'autres technologies dans le but de se connecter et d'échanger des données*

Co-funded by the Erasmus+ Programme of the European Union

avec d'autres dispositifs et systèmes sur l'internet. » (source Wikipedia)

L'objet connecté que nous utiliserons dans ce projet est une station météorologique, qui effectue des mesures pour les besoins de TRYAT. Ces mesures sont ensuite envoyées à un serveur qui les "pousse" à son tour vers le serveur TRYAT.

La station météorologique est décrite dans un autre document (IO4 : Capteurs et serveur de données pour les microcontrôleurs et les systèmes embarqués)

I. Sigfox

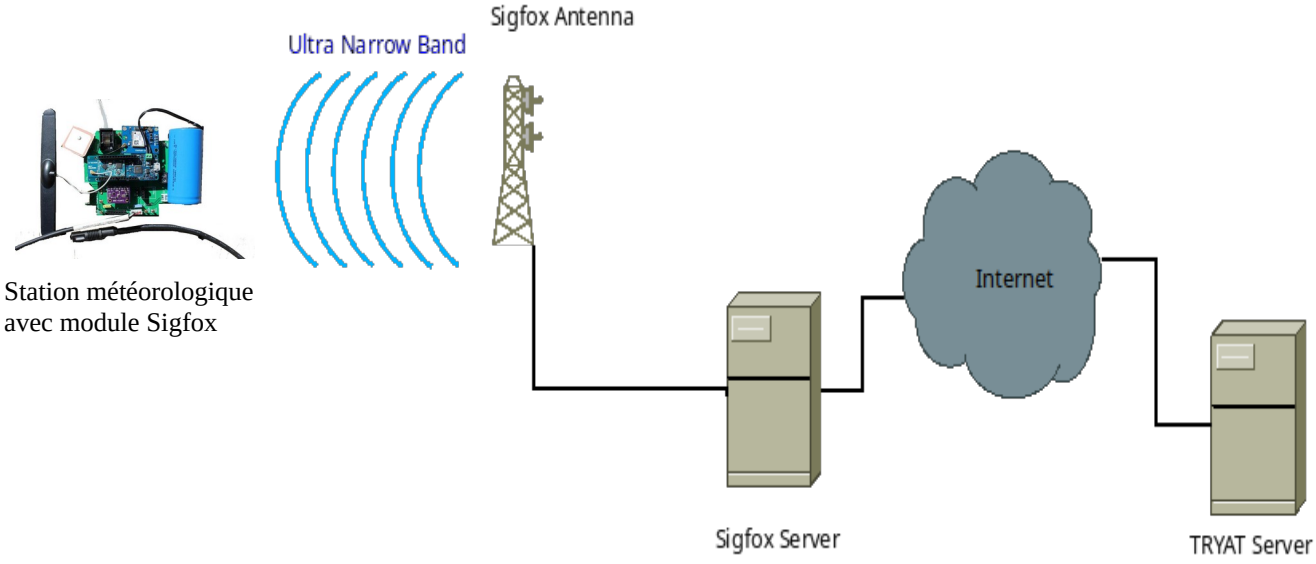
1. Description

« *« Sigfox est un opérateur français de réseau mondial fondé en 2010 qui construit des réseaux sans fil pour connecter des objets de faible puissance tels que des compteurs d'électricité et des montres intelligentes, qui doivent être allumés en permanence et émettre de petites quantités de données.*

Sigfox utilise la modulation par déplacement de phase binaire différentielle (DBPSK) et la modulation par déplacement de fréquence gaussienne (GFSK) qui permettent de communiquer en utilisant la bande radio ISM industrielle, scientifique et médicale qui utilise 868 MHz en Europe et 902 MHz aux États-Unis. Elle utilise un signal de grande portée qui passe librement à travers les objets solides, appelé "bande ultra étroite" et qui nécessite peu d'énergie, appelé "réseau étendu de faible puissance" (LPWAN). Le réseau est basé sur une topologie en étoile à guichet unique et nécessite un opérateur mobile pour acheminer le trafic généré. Le signal peut également être utilisé pour couvrir facilement de grandes zones et atteindre des objets souterrains. En octobre 2018, le réseau Sigfox IoT couvrait un total de 4,2 millions de kilomètres carrés dans 50 pays et devrait atteindre 60 pays d'ici à la fin de l'année 2018. » (source Wikipedia).

Le module Sigfox peut envoyer jusqu'à 140 messages par jour. La taille de chaque message est de 12 octets. Le coût annuel pour l'envoi des messages est de 5 à 15 €.

2. Configuration de la station météorologique Sigfox

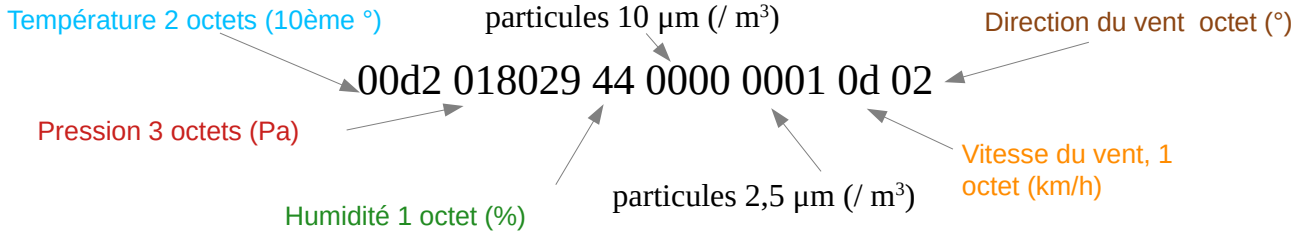


Co-funded by the Erasmus+ Programme of the European Union

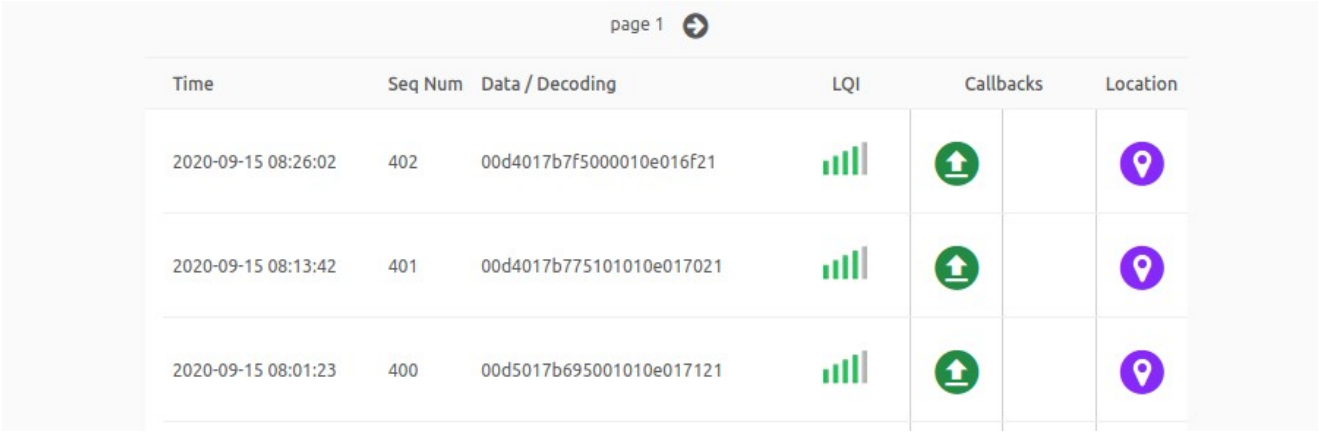
3. Trame de données

Le module Sigfox de la station météorologique envoie un message toutes les 12 minutes au serveur Sigfox. La trame d'envoi contient plusieurs données météorologiques :

temp::int:16 pres::uint:24 humi::uint:8 part1::uint:16 part25::uint:16 windspeed::uint:8 winddir::uint:8



Sur le backend sigfox, après avoir sélectionné l'appareil, vous pouvez les voir en vous connectant à l'interface web depuis le service web sigfox et en choisissant le module utilisé :

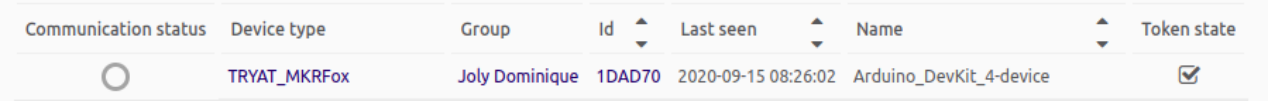


II. Stockage des données

1. Sigfox Callbacks

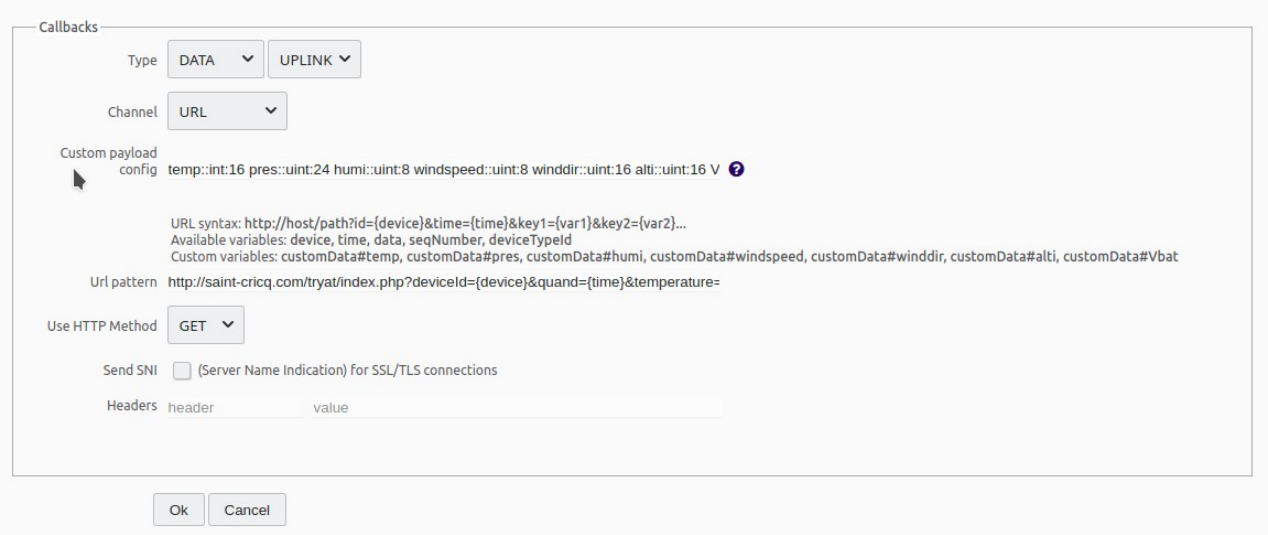
Pour obtenir ces données sur le serveur TRYAT, vous avez plusieurs solutions. Notre choix est de créer une " API ACCESS ", de les pousser à vrai dire une requête HTTP, en utilisant la méthode " GET ".

1. Sélectionnez l'appareil dans la liste des appareils en cliquant sur son "type d'appareil" (dans ce cas, TRYAT_MKRFox) :



2. Cliquez sur le menu " CALLBACKS " à gauche
3. Créez vos propres " rappels ".

Co-funded by the Erasmus+ Programme of the European Union



Notre URL, sur ce test, est :

http://saint-cricq.com:82/tryat/index.php?deviceId={device}&quand={time}&temperature={customData#temp}&pression={customData#pres}&humidite={customData#humi}&part1={customData#part1}&part25={customData#part25}&vitVent={customData#windspeed}&dirVent={customData#winddir}

La configuration « Custom paylod » est :

temp::int:16 pres::uint:24 humi::uint:8 part1::uint:16 part25::uint:16 windspeed::uint:8 winddir::uint:8

2. Sur le serveur TRYAT

Le serveur web apache2 recevra cette demande GET :

```
185.110.97.87 - - [24/Sep/2020:16:18:30 +0200] "GET /tryat/index.php?deviceId=C05E8&quand=1600957108&temperature=210&pression=98352&humidite=67&part1=2&part25=3&vitVent=2&dirVent=255 HTTP/1.1" 200 696 "-" "SIGFOX"
```

Nous devons maintenant stocker ces données.

a. Méthode PHP/CSV

Maintenant, il faut écrire la page web " index.php " sur le serveur pour stocker les données. Notre choix, pour des raisons de sécurité, est de les écrire dans un fichier CSV :

```
<?php
/*


 *
 Retrieved data :
{
 "Device-ID" : "{device}",
 "Date" : {time},
 "Temperature" : {customData#temp},
 "Pression" : {customData#pres},
 "Humidite" : {customData#humi},
 "part1" : {customData#part1}
 "part25" : {customData#part25}
```


Co-funded by the Erasmus+ Programme of the European Union

```
 "Vit_Vent" : {customData#windspeed},
 "Dir_Vent" : {customData#winddir}
}
 */
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>TRYAT - Récupération des données</title>
  <meta http-equiv="content-type" content="text/html;charset=utf-8" />
  <meta name="generator" content="Geany 1.36" />
</head>

<body>
  Récupération des données Sigfox
<?php

  $device=$_GET["deviceId"];
  $quand=$_GET["quand"];
  $temperature=$_GET["temperature"];
  $pression=$_GET["pression"];
  $humidite=$_GET["humidite"];
  $part1=$_GET["part1"]; // 10um particul/m3
  $part25=$_GET["part25"]; // 2.5um particul/m3
  $vitVent=$_GET["vitVent"];
  $dirVent=$_GET["dirVent"];
  $fichier = fopen('./sigfoxData.csv', 'a');
  // preparation des données :
        //$acquisition=$device.";".$quand.";".$temperature.";".$pression.";".$humidite.";".$vitVent.";".
$dirVent."\n";
        $acquisition=$device.";".$quand.";".$temperature.";".$pression.";".$humidite.";".$vitVent.";".
$dirVent."\n";
        fputs($fichier,$acquisition);
  fclose($fichier);
?>
</body>

</html>
```

Chaque fois que la page index.php récupère des données, elle les écrit :

```
1DAD70;1600126036;212;96925;64;1;135
1DAD70;1600126775;211;96931;64;0;135
1DAD70;1600127514;211;96953;63;0;135
1DAD70;1600128253;208;96952;64;0;135
1DAD70;1600128992;209;96960;65;0;135
1DAD70;1600129731;211;96959;63;0;135
1DAD70;1600130470;210;96978;66;1;135
```


Co-funded by the Erasmus+ Programme of the European Union

```
1DAD70;1600131208;209;97006;66;0;135
1DAD70;1600131948;208;97017;66;2;251
1DAD70;1600132686;206;97020;69;3;135
1DAD70;1600133425;207;97019;67;0;135
1DAD70;1600134165;207;97028;69;1;135
1DAD70;1600134904;206;97044;70;0;135
1DAD70;1600135643;207;97036;71;0;153
1DAD70;1600136381;207;97049;72;1;270
1DAD70;1600137121;208;97047;71;0;270
```

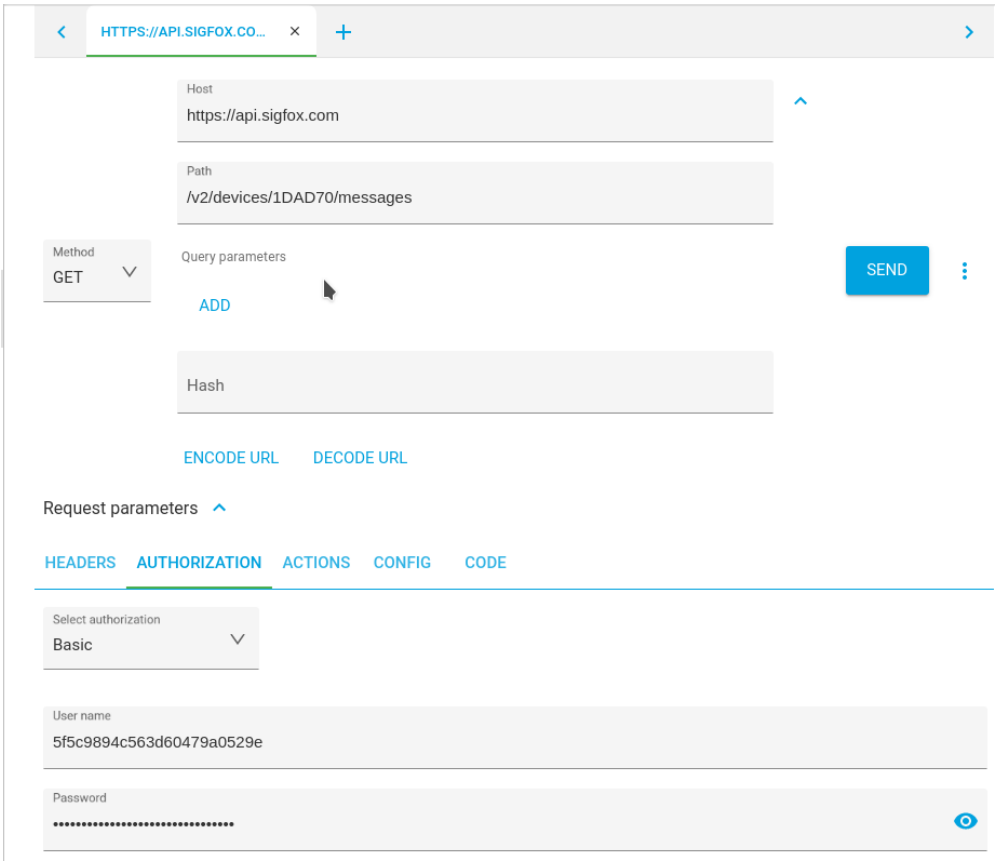
Maintenant, vous pouvez obtenir ce fichier avec un programme Python pour utiliser les données à des fins statistiques ou pour les mettre dans la base de données TRYAT.

b. Méthode PHP / Mysql

Vous pouvez appeler une page php qui stockera les données dans la base de données Mysql. Attention ! Vous devez analyser ces données pour éviter que le serveur ne soit victime d'une "attaque par injection".

3. API REST

« *Le Representational state transfer (REST) est un style d'architecture logicielle qui définit un ensemble de contraintes à utiliser pour la création de services Web. Les services web conformes au style architectural REST, appelés services web RESTful, assurent l'interopérabilité entre les systèmes informatiques sur l'internet. Les services Web RESTful permettent aux systèmes demandeurs d'accéder à des représentations textuelles de ressources Web et de les manipuler en utilisant un ensemble uniforme et prédéfini d'opérations apatrides. D'autres types de services Web, tels que les services Web SOAP, exposent leurs propres ensembles d'opérations arbitraires.* »



Co-funded by the Erasmus+ Programme of the European Union

Source : wikipedia

a. Configuration du compte API

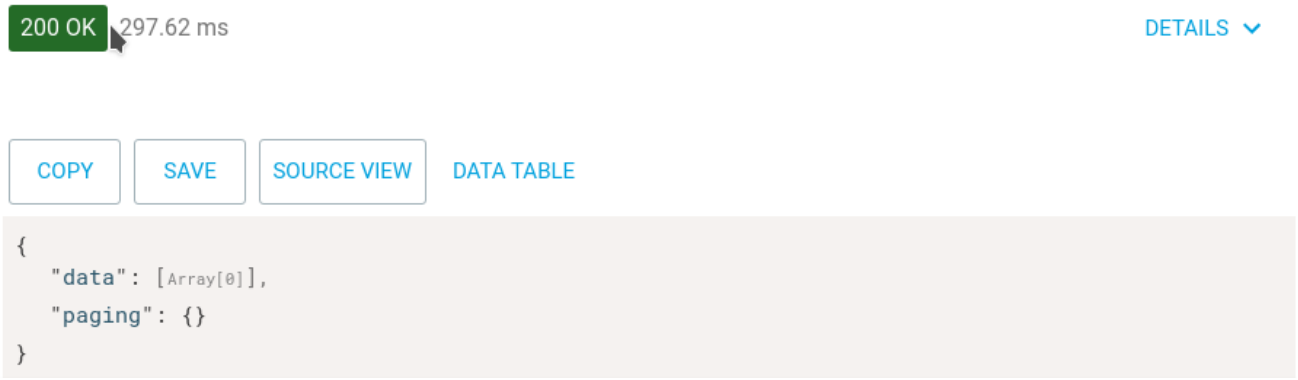
Vous devez définir un mot de passe et un login sur votre compte Sigox pour ouvrir un accès authentifié aux données enregistrées :

- Aller dans « Group »
- Sélectionnez ou créez un groupe dans votre liste
- Maintenant, allez à ti API ACCESS et créez-en un nouveau dans votre fuseau horaire. Vous obtiendrez un identifiant et un mot de passe.
- Vous pouvez tester l'accès avec votre navigateur en entrant l'URL suivante « https://api.sigfox.com/v2/devices/1DAD70/messages » (1DAD70 est l'identifiant de votre module).

b. Advanced Rest Client

Vous pouvez obtenir les données avec l'extension Firefox " RESTClient " par exemple, vous pouvez utiliser le logiciel open source " Advanced Rest Client " :

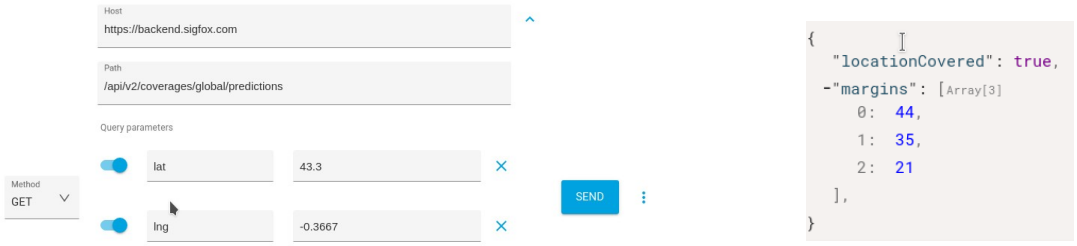
Vous obtenez alors un message de retour réussi :



c. Un autre exemple : test de couverture d'une localisation

Avant d'utiliser le service sigfox, vous pouvez consulter la couverture de l'emplacement pour être sûr de pouvoir envoyer les données au Sigfox Cloud :

Nous allons tester si notre localisation (PAU-France) est couverte par le réseau Sigfox ;



Co-funded by the Erasmus+ Programme of the European Union

Et bonne nouvelle, la ville de Pau est couverte.

Pour obtenir plus d'informations, vous pouvez consulter le site :https://support.sigfox.com/docs/api-documentation

Co-funded by the Erasmus+ Programme of the European Union