

DaWeather

MANUEL D'UTILISATION



LE PROJET

Contexte

C'est dans le cadre de projet tutoré de Licence Professionnelle DAWIN (Développeur d'Applications Web et Images Numériques) mené au sein de l'IUT Informatique de Bordeaux que le projet DaWeather a été réalisé. Ce projet, d'une durée de 4 semaines, a été proposé par Mr Fossé, professeur au sein de l'IUT Informatique de Bordeaux.

Le but de ce projet était la mise en place d'une station météorologique personnelle et connectée, le département nous a fourni un ensemble de 6 capteurs (anémomètre, direction du vent, pluie, humidité, température et pression atmosphérique), d'une carte Arduino Uno et d'un Raspberry Pi. Afin de réaliser ce projet, nous étions un groupe de 4 étudiants, de la Licence Professionnelle DAWIN, présenté dans la suite de ce document.

Equipe

Afin d'effectuer un projet de cet ampleur, nous avons travaillé en un groupe composé de 4 étudiants :

ABELA Christopher : En charge de l'administration globale du projet, Christopher a mis en place un échange hebdomadaire avec Mme Uny, professeur gérant le suivi des projets au sein de l'IUT Informatique de Bordeaux. Il a travaillé sur la mise en oeuvre de la base de données complexe nous permettant de gérer les données reçues de l'Arduino et sur le développement des scripts Python permettant l'interaction entre l'Arduino et la Raspberry PI.

IACHI Dimitri : Afin de développer une application optimale pour le recueil des données météorologiques, Dimitri a mis en place l'architecture des webservices permettant à l'application d'envoyer des requêtes à la Raspberry PI afin d'afficher les valeurs récupérées de l'Arduino. Il a notamment élaboré le design de l'application ainsi que l'ensemble du développement de celle-ci en s'aidant du framework Cordova avec lequel il a pu compiler l'application et la rendre multi-plateformes.

MASSET Antoine : Le département nous ayant donné un ensemble de capteurs météorologiques séparément, il nous a fallu les assembler mécaniquement mais aussi électroniquement afin de récupérer les données en temps réels. Antoine a donc travaillé sur la conception et la réalisation des données des capteurs. Voyant la complexité de l'architecture de la base de données, Antoine a donc aidé Christopher à la mettre en place. Enfin tout comme Christopher, il a participé au développement des scripts Python.

MILLERIOUX Nicolas : Ayant auparavant étudié le génie électrique, Nicolas a pu travailler sur la partie électronique du projet afin de récupérer les valeurs électroniques des capteurs et les traiter par un programme Arduino. Une fois la récupération des données mise en place, il a travaillé sur le design, en coordination avec Dimitri, afin de rendre l'application ergonomique et utilisable par l'ensemble des personnes. Enfin, il a participé au développement de l'application afin d'aider Dimitri.

PARTIE ELECTRONIQUE

Le projet DaWeather est composé d'une partie électronique et d'une partie informatique. Dans cette première partie, nous expliquerons en détails toutes les parties électroniques du projet, les capteurs, la carte Arduino, la Raspberry PI et la carte "Capteurs".

Capteurs

Pour pouvoir mettre en place une station météorologique personnelle, le département informatique de Bordeaux nous a mis à disposition un ensemble de plusieurs capteurs. Certains capteurs étaient prêts à être utilisé directement mais ceux récoltant la température, l'humidité et la pression avaient besoin d'être connectés en "fils volants" (pour ces capteurs nous avons imprimé une carte électronique que nous expliquerons après).

Vent, Pluie & Direction : SEN-08942

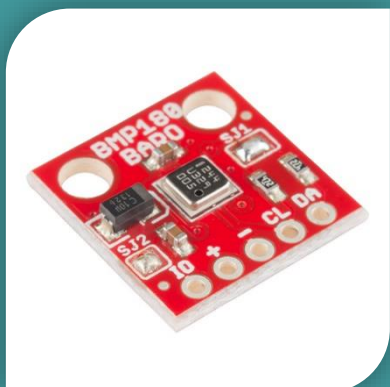


Le département nous a fourni un ensemble de capteurs météorologiques prêt à l'emploi et à monter nous-même. Ces capteurs sont assemblés et prêt à être connectés au périphérique gérant les valeurs météorologiques par 2 connecteurs RJ11 (un pour la pluie et un pour l'ensemble Vitesse du vent et Direction du vent). Ces capteurs seront connectés à une carte électronique. Pour plus d'informations à

propos de ces capteurs, vous pouvez consulter la datasheet fournie dans le répertoire.

Pour l'acheter : <https://www.sparkfun.com/products/8942> (62€)

Température & Humidité : BMP180



Pour permettre d'avoir une station météorologique complète, le département nous a fourni un composant électronique permettant de récupérer la température et l'humidité de l'endroit où la carte se situe. Nous avons commencé par brancher ce composant à la carte Arduino grâce à une carte de test pour effectuer des tests de mesures et nous assurer que le composant fonctionnait correctement. Par la suite, nous avons intégré ce composant à la carte électronique supportant les connecteurs des autres capteurs. Pour plus d'informations à propos de ce capteur, vous pouvez consulter la datasheet fournie dans le répertoire.

Pour l'acheter : <https://www.sparkfun.com/products/11824> (10€)

Pression : RHT03



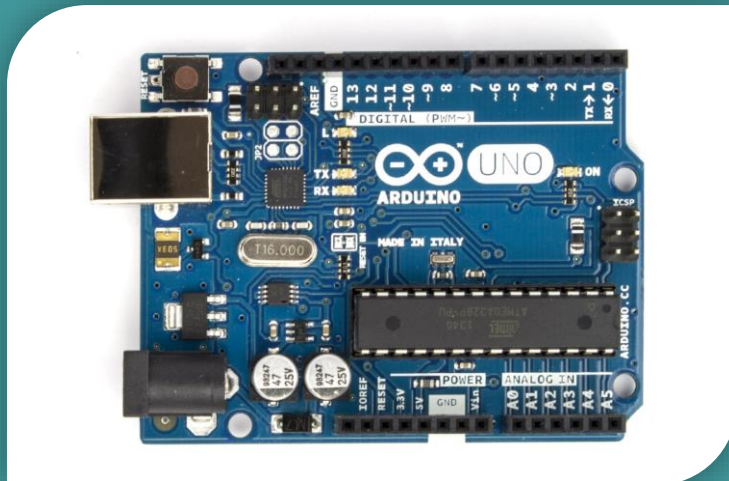
Le département nous a aussi permis d'utiliser un capteur de pression pour récupérer la pression atmosphérique de la position de la carte électronique. Comme le capteur de température et humidité, nous avons dû effectuer des tests pour récupérer la valeur correcte puis nous l'avons intégré à la carte électronique "Capteurs". Pour plus d'informations à propos de ce capteur, vous pouvez consulter la datasheet fournie dans le répertoire.

Pour l'acheter : <https://www.sparkfun.com/products/10167> (9€)

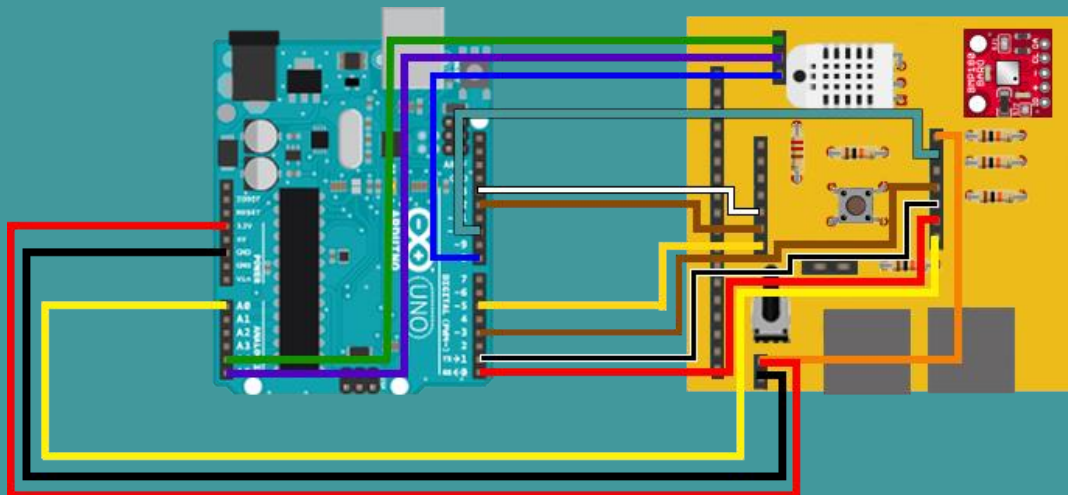
Arduino

Voulant une solution libre et simple d'utilisation, le département Informatique a mis à notre disposition une carte Arduino Uno que nous avons utilisé pour récupérer les valeurs des capteurs météorologiques. Nous avons rassemblé les algorithmes permettant de récupérer les valeurs des capteurs un par un afin de créer un programme complet qui récupèrent l'ensemble des valeurs des capteurs.

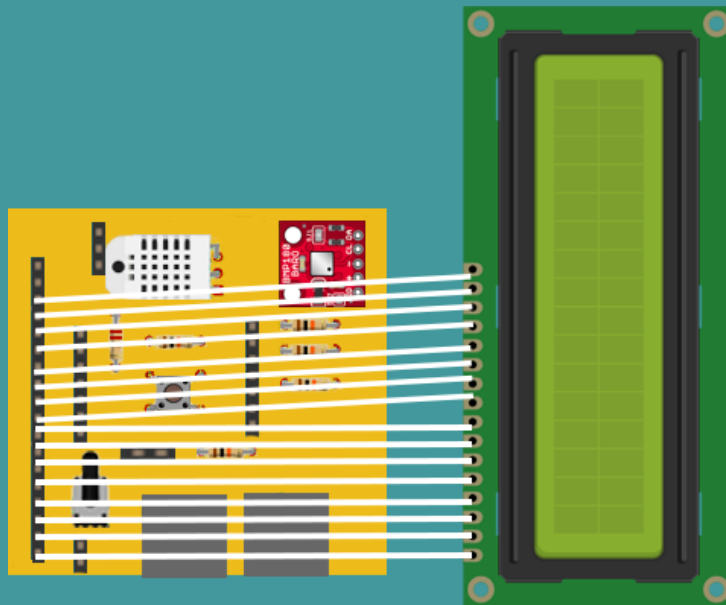
Pour l'acheter : <http://store.arduino.cc/product/A000066> (20€)



Pour pouvoir connecter l'ensemble des capteurs à la carte Arduino, veuillez suivre le schéma électrique suivant :



Le programme était prévu pour fonctionner en utilisant l'écran LCD pour afficher les données. Si vous voulez l'utiliser, vous juste devez remplacer « Serial » par « LCD » dans le programme Arduino ainsi que câbler la carte comme le schéma suivant.



Vous pouvez retrouver ces schémas en annexe pour une meilleure lisibilité.

Raspberry PI

Le but du projet était de constituer une station météorologique connectée et autonome, ainsi pour parfaire l'autonomie de la station, le département Informatique de Bordeaux nous a prêté un Raspberry PI afin de pouvoir l'utiliser comme ordinateur indépendant.

Raspberry PI est un ordinateur "low cost", utilisé pour apprendre les principes de l'informatique dans les écoles. Il est présenté sous forme d'une carte électronique ayant les branchements basiques d'un ordinateur (USB2.0, HDMI, Alimentation par Micro-USB, Prise Audio, Micro SD, Ethernet).

Pour notre projet, nous avons eu à connecter la Raspberry PI à internet par le port Ethernet prévu à cette effet, insérer la carte SD contenant l'image du système d'exploitation (expliqué par la suite), brancher un écran HDMI pour la configuration, connecter la carte Arduino

et la Raspberry en USB et enfin brancher le port Micro USB à une prise secteur 230V.

Pour acheter celle utilisée en projet :
<http://uk.rs-online.com/web/p/processor-microcontroller-development-kits/7568308/> (33€)



Carte Electronique

Après avoir testé l'ensemble des capteurs, nous nous sommes rendu compte que pour rendre le projet plus concret, il serait préférable de faire notre propre carte électronique afin d'assembler les différents capteurs et rendre les connexions entre la carte Arduino et les capteurs plus aisées.

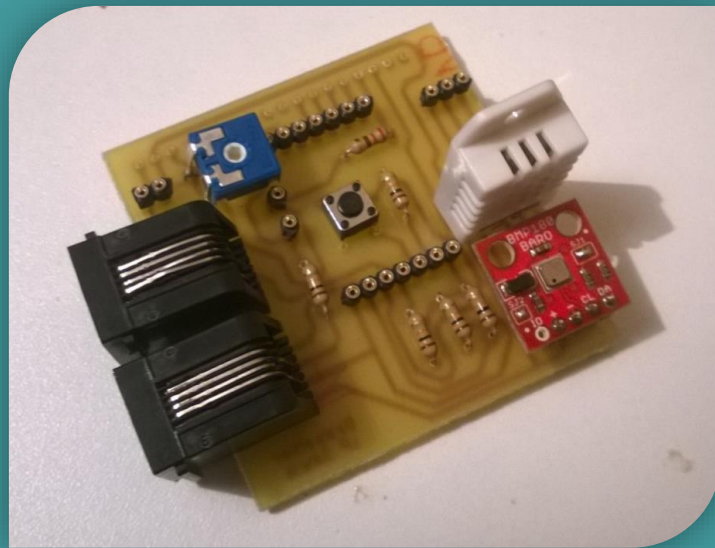
Pour faire cette carte, nous avons dû effectuer le schéma électrique en utilisant le logiciel Kicad. Puis après avoir disposé les composants, nous avons constitué la carte imprimée appelée PCB (Printed Circuit Board) et pour finir nous sommes allés l'imprimer au département Génie Electrique et Informatique Industrielle de Bordeaux.

Pour pouvoir ouvrir et utiliser les fichiers Kicad, veuillez télécharger le programme à l'adresse suivante :

<http://www.kicad-pcb.org/display/KICAD/KiCad+EDA+Software+Suite>

Vous pouvez retrouver l'ensemble du projet Kicad et le typon utilisé dans le répertoire du projet.

Si vous ne possédez pas le matériel nécessaire, Sparkfun met en vente une carte contenant l'ensemble des composants disponible à cette adresse : <https://www.sparkfun.com/products/12081>



PARTIE PROGRAMMATION

Arduino

Arduino est une plateforme Open-source se reposant sur un ensemble Logiciel / Matériel dit "Facile à utiliser". Après avoir vu l'ensemble des connexions électroniques vues précédemment dans ce document, nous vous expliquerons ici les parties du programme nécessaire à la récupération des données météorologiques. Le programme est codé en Arduino, il s'agit de leur propre langage basé sur une structure du langage C, il est donc principalement impératif. Afin de pouvoir copier/coller le programme directement, les explications de l'algorithme seront insérés en commentaires. La totalité du programme est disponible dans le dossier Arduino du répertoire. Vous devrez installer le logiciel Arduino pour pouvoir l'utiliser (<http://arduino.cc/en/Main/Software>).

De plus, pour pouvoir utiliser les capteurs de pression et celui d'humidité / température, il vous faut télécharger les librairies suivantes et les copier/coller dans le dossier C:\Program Files (x86)\Arduino\libraries :

- SFE_BMP180 (Humidité et Température) :
https://github.com/sparkfun/BMP180_Breakout/archive/master.zip
- DHT :
<https://github.com/adafruit/DHT-sensor-library/archive/master.zip>

Si vous rencontrez des problèmes pour installer ces librairies, veuillez-vous rendre à cette adresse :

<https://learn.sparkfun.com/tutorials/bmp180-barometric-pressure-sensor-hookup-/installing-the-arduino-library>
pour la librairie SFE_BMP180

<http://playground.arduino.cc/Main/DHTLib>
pour la librairie DHT

Raspberry PI

Afin de rendre la station météo connectée et autonome, nous avons utilisé un Raspberry PI pour gérer l'interaction des webservices et pour supporter la base de données. Pour utiliser la Raspberry, nous avons choisi d'utiliser la distribution Raspbian dont vous pourrez télécharger l'image à cette adresse : <http://www.raspbian.org/RaspbianInstaller>

Afin d'utiliser la raspberry et toutes les fonctionnalités utiles au projet, il faut installer une suite de paquets, ces paquets sont installables en ouvrant une fenêtre de commande sur la raspberry et en utilisant la commande :

-> `sudo apt get "le nom du paquet "`

Listes des packages à installer :

- php5
- php pear
- php5 mysql
- mysql server
- apache2
- curl
- python
- python mysqldb
- cron
- phpmyadmin
- arduino
- git
- php5 curl

Une fois l'ensemble de ces modules installés, vous devrez importer la base de données que nous avons créée afin qu'elle soit stockée sur la Raspberry. Pour cela vous devrez vous rendre sur l'interface de gestion PhpMyAdmin (localhost/phpmyadmin) et importer la base de données que vous trouverez dans le répertoire du projet.

Une fois avoir installé la base de données, il vous suffit de récupérer les fichiers contenus dans `www.zip` et les copier dans le dossier `/var/www` en tant qu'utilisateur `root` et groupe `www-data`.

Ensuite, il vous faudra lancer le script `python` et le script `update.sh` (disponible également dans le répertoire du projet) qui va pour le premier, insérer automatiquement les données dans la base et pour le second, faire le lien qui permet d'envoyer automatiquement les données à l'application.

Une fois cette étape terminée, vous aurez installé l'ensemble des composants nécessaires pour pouvoir utiliser votre propre station météorologique.

Application

Afin de rendre notre station météorologique connectée, nous avons développé une application ergonomique et multiplateforme. Cette application permet au possesseur de la station météo de voir ces propres données météorologiques via la création d'un compte personnel.

Cette application est une application hybride, compilée avec Cordova/PhoneGap, et codée en HTML5, CSS3, JAVASCRIPT. L'application utilise également deux librairies JavaScript jQuery et AngularJS afin de dynamiser le fonctionnement de l'application.

Toutes les sources de cette application sont disponibles dans le fichier zip de l'application. Une fois le code téléchargé il vous suffira de changer seulement le lien menant vers le webservice (expliqué dans la suite).

Webservice

Le Webservice est ici un système d'échange de données entre l'application et la station météo. Comme la station météo, plus particulièrement la Raspberry Pi, qui est connectée à Internet, peut être bloquée en entrée derrière le pare-feu de votre box ou derrière votre infrastructure réseau, il nous a donc fallu trouver une solution alternative utilisant un serveur Web distant pour échanger les données collectées par la station météo à l'application mobile et tablette.

Ce Webservice est codé en PHP sous le framework Laravel afin qu'il soit facilement maintenable par quiconque. Il requiert la connexion avec une base de données MySQL, SQLite, PostgreSQL ou encore Microsoft SQL. Toute fois le script SQL fourni pour l'installation est propre à MySQL Server, si vous souhaitez utiliser un autre type de base de données, il vous suffira de recréer la structure suivant la convention SQL du SGBD choisi.

Le Webservice requiert également un compte mail accessible par le protocole SMTP. Ainsi lorsque la station météo détecte des valeurs suspectes elle contacte le Webservice afin de prévenir son propriétaire. De même lorsque l'utilisateur s'enregistre sur son application, un mail lui est envoyé contenant sa clé privée qu'il doit saisir sur la station.

Pour l'installation de ce Webservice sur un serveur Web classique, il vous suffit de :

- dézipper le fichier zip contenant le webservice sur votre espace web.
- installer la base de données fournie avec sur votre SGBD, paramétrer le fichier database.php se trouvant dans le dossier app/config/.
- paramétrer le fichier mail.php se trouvant dans le même dossier.
- paramétrer le fichier workbench.php se trouvant dans le même dossier.
- modifier dans le script python l'URL menant vers le webservice

- modifier dans le code JavaScript de l'application (se trouvant dans `www/app/controller/base_CTRL.js` à la variable `$scope._root.webservice`) l'URL menant vers le webservice.

A PROPOS

Licences

Arduino : L'ensemble du contenu concernant Arduino est sous licence Creative Commons Attribution ShareAlike 3.0.

Pour plus d'informations : <http://www.arduino.cc/>

Raspberry : La documentation concernant le Raspberry PI et son utilisation est sous licence Creative Commons.

Pour plus d'informations : <http://www.raspberrypi.org/>

Application : L'ensemble du code et du contenu de ce projet est sous licence Open Source. Toutefois pour utiliser l'intégralité de ce projet, le groupe DaWeather souhaite être maintenu au courant et cité dans son utilisation.

Remerciements

Nous souhaitons remercier l'ensemble du département Informatique pour nous avoir mis à disposition le matériel utile au bon déroulement de ce projet, que ce soit par le prêt de la Raspberry ou de la tablette utilisée pour le développement de l'application.

Nous remercions tout particulièrement Mr Fossé, qui de par sa disponibilité et ses idées, nous a aidé à nous orienter quand il le fallait. Nous remercions aussi Mr Journet, qui nous a suivis tout au long du projet. Nous souhaitons aussi remercier le département Génie Electrique et Informatique Industrielle pour l'impression de la carte électronique et leur disponibilité.

Enfin nous remercions l'ensemble des personnes qui ont rendu ce projet vivant, que ce soit par leurs remarques ou leur aide.

Nous contacter

Vous pouvez nous contacter si vous avez des questions ou des problèmes.

DaWeather

- twitter : @DaWeatherGroup
- email : dawin.weatherstation@gmail.com
- site : daweather.ovh

ABELA Christopher

- twitter : @cabela_
- email : chrismeteo@gmail.com

IACHI Dimitri

- twitter : @Dimer47
- email : dimitri@iachi.fr

MASSET Antoine

- twitter : @EINeffe
- email : antoine.masset@outlook.com

MILLERIOUX Nicolas

- twitter : @paniiko_
- email : nico.millerioux@gmail.com



DaWeather

ANNEXE

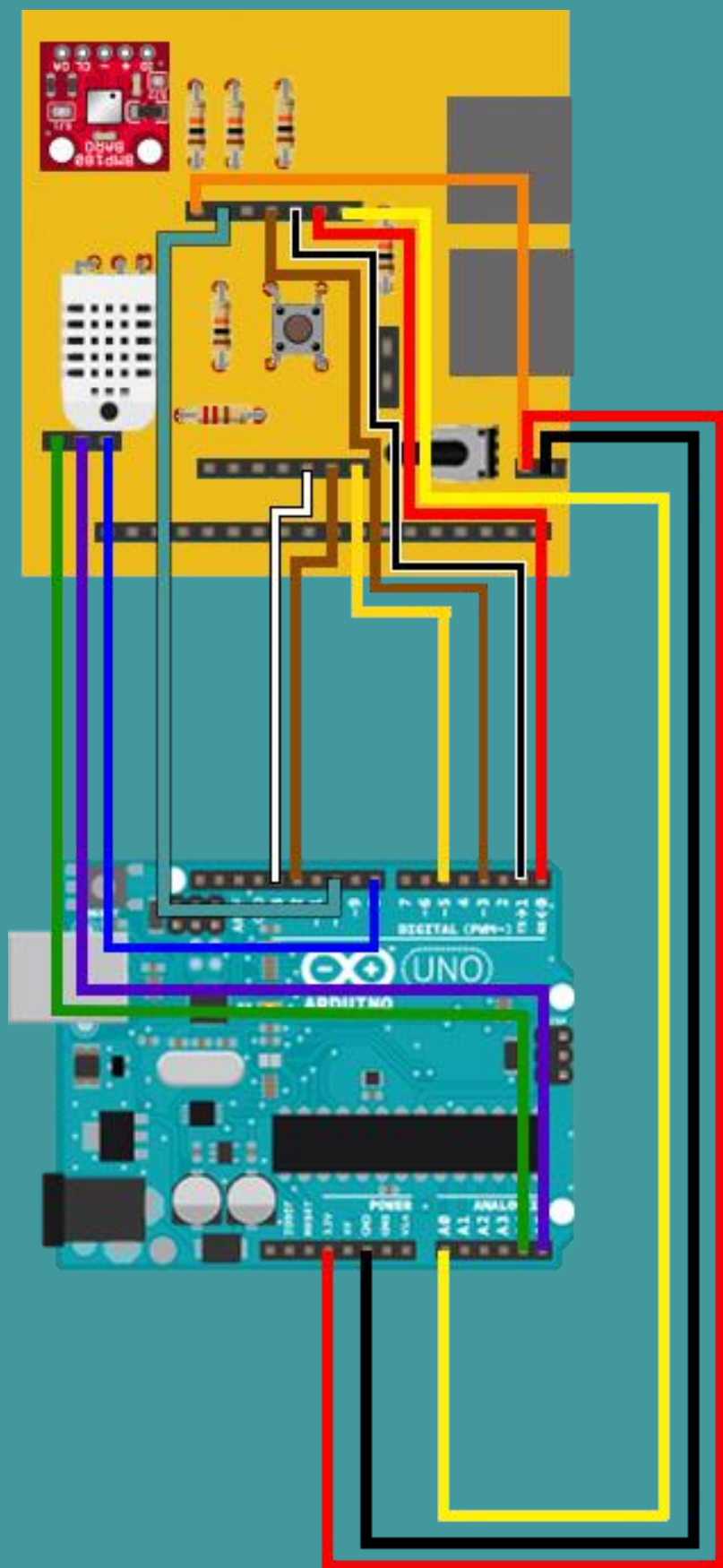


Figure 1 Schéma de câblage Arduino – Capteurs

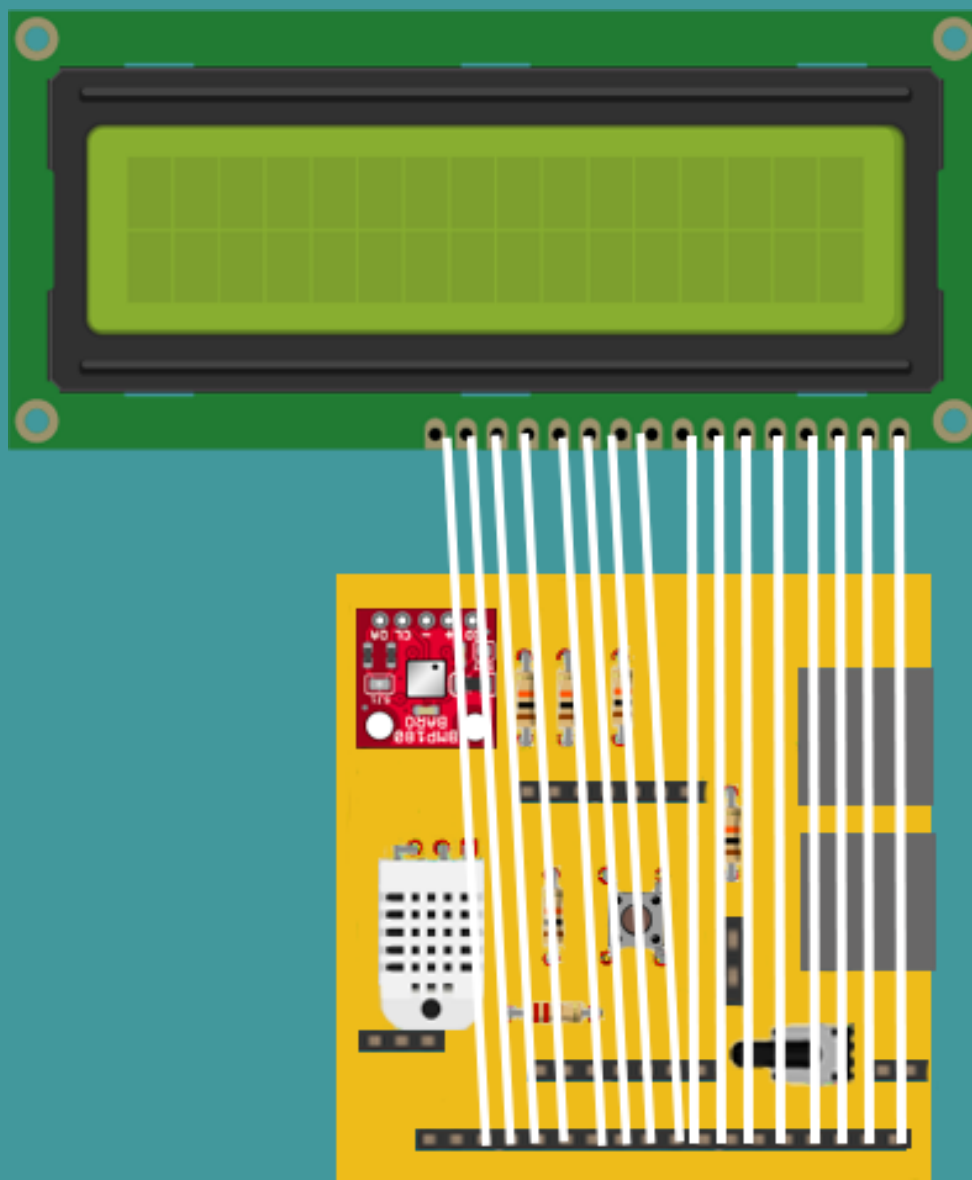


Figure 2 Schéma de câblage LCD - Capteurs