

SERREHYDRA : 2.0

SERVICE MANUAL

Table des matières

Table des matières	2
Partie 1 : Physique	6
Entretien de la serre	6
Serre physique	6
Serre palram	6
Panneau	6
Fenêtre	6
Solidification	7
Roulement à bille	7
Entretien des composants	7
Entrée d'eau	7
Réservoir (eau filtrée et engrais)	7
Filtreur	8
Manuel d'utilisation	8
Installation	8
Tuyauterie	9
Introduction	9
Connecter la pompe 1 au système de filtreur	10
Connecter la pompe 2 au système d'arrosage	10
Pompes	10
Introduction	10
Emplacement	10
Utilisation	11
Système de burette	11
Burette	11
Aiguille	11
Servomoteur	12

Support à burette	13
Calvettes	13
Rotation des calvettes	13
Calvettes physiques	13
Partie 2 : Électronique	14
La boîte de relais	14
Entretien des capteurs	14
DHT11	14
Capteur de niveau d'eau	15
Capteur de ph	16
Relais	16
Servo-moteurs	17
Partie 3 : informatique	18
Maintenance de l'app mobile	18
Survol de l'application	18
Recevoir les données du serveur	19
Envoyer les données au serveur	20
Données sur la température ambiante	21
L'intervalle entre les requêtes	21
Conversion JSON en object	22
Page des views	23
Interface utilisateur	24
Maintenance du serveur	25
Base de données	25
Méthodes utilisées	26
GET	26
POST	26
Maintenance des microcontrôleurs	27
Relier les arduinos	27

Reprogrammer un Arduino	27
Connexions sérieelles	28
Les pins	29
Les seuils	30
Les lectures	31
Les commandes	32
L'ajout d'engrais	32
Le temps de déversement des engrais	33
L'envoi de données au Raspberry Pi	34
La réception des données	35
Communication externe (Raspberry Pi)	36
Le physique	36
La connexion au Arduino	36
La connexion Wifi	36
L'alimentation	36
Perte de connexion (dépannage)	36
Schéma de communications	37
Explication générale	37
#1 Arduino	37
#2 Raspberry Pi	38
#4 Serveur	38
#5 Application mobile	38

Partie 1 : Physique

Entretien de la serre

Serre physique

Serre palram

Pour toute question sur l'installation, l'entretien et la maintenance de la serre, veuillez-vous référer au manuel d'utilisation de ce dernier : <https://www.shedsforlessdirect.com/manuals/Palram-6x8-Mythos-Hobby-Greenhouse-hg5008g-Manual.pdf>.

Panneau

Dans l'éventualité qu'un panneau de poly carbonate soit emportée par le vent lors d'une dure tempête. Il est possible de le remplacer en acheter des panneaux de poly carbonate transparent et de le couper selon les mesures du panneau perdu.

Fenêtre

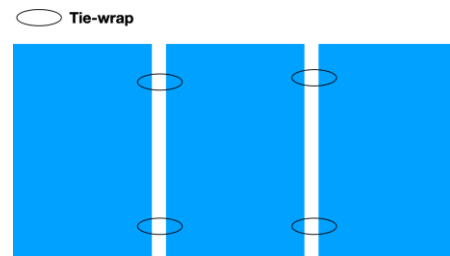
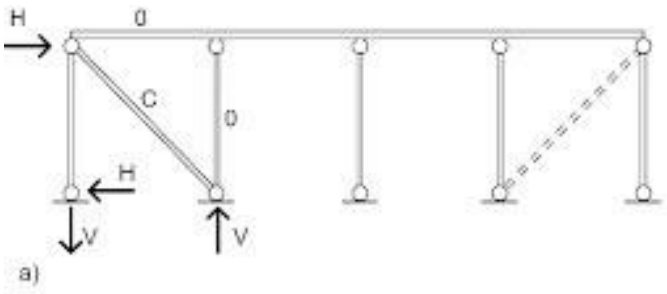
Il est également possible le mécanisme de la fenêtre original se brise. Dans un tel cas, il est possible de le réparer relativement simplement en fixant le mécanisme avec des vis.



Exemple d'un panneau envolé

Solidification

Puisque la serre est située sur un quai sur la rivière Richelieu, il est possible qu'elle fasse face à de vent violent qui pourraient endommager drastiquement la serre. Pour remédier à ce problème, il devrait avoir une solidification de la serre. Chaque tuile du toit devrait avoir des trous pour y faire passer des « tie-wrap ». Ces « tie-wrap » devraient attacher chaque tuile voisine question de les connecter et de les empêcher de partir au vent. De plus, la structure de la serre devrait être renforcée avec des poteaux de métal supplémentaires positionnés diagonalement.



Roulement à bille

Dans le cas où un des roulements à bille brise, vous pouvez remplacer seulement la pièce manquante. Le fichier avec chacune des composantes se trouve dans le GitHub du projet. Les billes utilisées dans le roulement sont des billes en acier inoxydable d'un demi pouce (voir dans la liste de pièces sur le GitHub).

Entretien des composants

Entrée d'eau

Dans le cas d'un problème lié à l'entrée de l'eau sous la serre, veuillez vérifier sous la clavette la plus près du bac d'eau. Vous retrouverez une plaque en aluminium arborant une pompe. Cette dernière peut être remplacée

Réservoir (eau filtrée et engrais)

Si un bris survient avec le réservoir, vous devez acheter un bac avec une capacité de 53L et y percer un trou inférieur à 3/8" au milieu du bac. Par la suite vous devez coller en dessous du bac un adaptateur mâle femelle vis-à-vis avec le trou à l'aide de la colle goop marine et attendre 24h avant d'y retoucher. Ensuite, vous pouvez réparer les pattes de bois au bac en pré perçant le bac et les pattes et après visser les pattes de bois au bac avec délicatesse. Important : appliquer de la goop sur chaque trou, mettre un « washer », rappliquer de la goop pour ensuite visser. Il restera à visser les pattes du bac en les accotant sur le support à burettes.



Filtreur

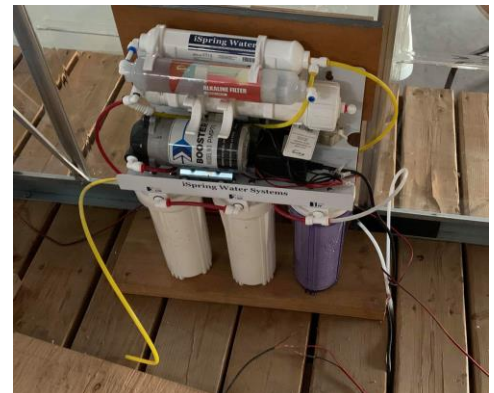
Manuel d'utilisation

Pour tout questionnement technique à propos du filtre, allez visiter le manuel d'instruction suivant :

<https://www.123filter.com/ac/kb/ISPRING%20Residential%20RO%20manual%20Version%202015-6.pdf>. (CONSEIL : lire le manuel avant l'installation)

Installation

Accrocher le filtreur sur le support de bois en photo ci-dessous grâce aux trous dans la base de métal et les vis qui sont fixées sur la structure de bois. Pour l'installation, il ne faut pas le connecter à la bonbonne d'eau puisque l'eau ira directement dans le bac. Assurer vous les deux fils du filtreur sont branchés avant de l'utiliser. IMPORTANT : vider l'eau des réservoirs du filtreur lors de gel, s'assurer que les tuyaux sont toujours bien connectés à leur embout. Le tuyau de sortie doit se vider dans le réservoir principal de la serre.



Tuyauterie

Introduction

S'assurer qu'à chaque fois que vous connectez un tuyau avec un autre pièce pour y faire passer un liquide, vous y ajoutez un « hose-clamp » pour s'assurer de l'étanchéité des connexions. Visser la bien fort et accoter la bien à l'extrémité du tuyau.

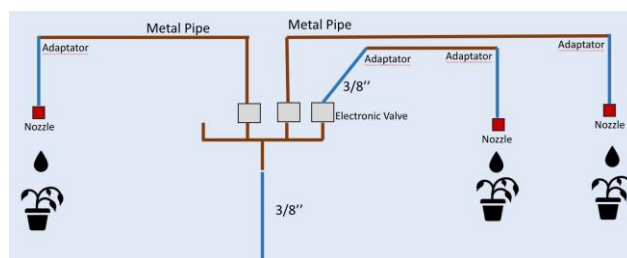
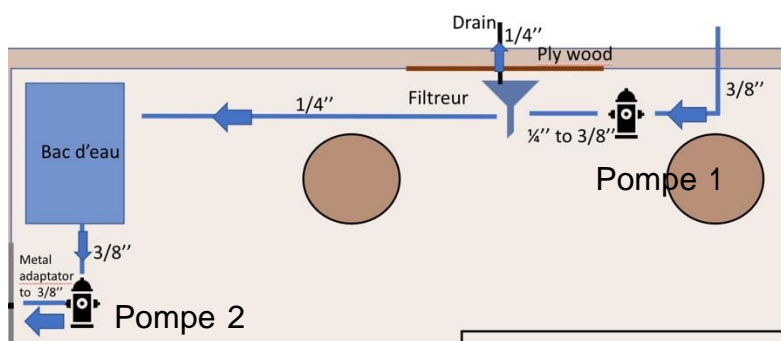
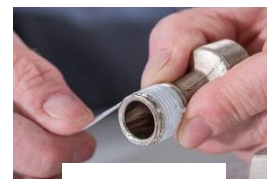
Mettre au bout de l'embout qui sera connecté au tuyau du Téflon, appliquez une à deux couches sur toute la surface dans le sens horaire (visser). Ceci assurera aussi une étanchéité des connexions.

Si vous devez connecter deux tuyaux de différentes tailles, utiliser les adaptateurs femelle-femelle en métal correspondant aux tailles des tuyaux (ex : 3/8 à 1/2).

Si vous désirez connecter un tuyau à un tuyau de métal ayant des filets, veuillez utiliser des adaptateurs femelles et mâles avec les bonnes tailles.

Si vous désirez connecter deux tuyaux de métal ayant des filets, veuillez utiliser des adaptateurs mâles vers mâles de bonne taille.

Pour fixer les tuyaux à la serre, vous devez utiliser des serre-joints. Vous devez en mettre au minimum à chaque mètre de tuyaux. Vous devez les serrer en clouant ou en vissant de chaque côté des serre-joints.



Connecter la pompe 1 au système de filtreur

Il vous faut une longueur voulue de tuyau de taille 3/8" pour les connecter aux deux embouts de la pompe. Ajouter de téflon aux deux embouts et lorsque les tuyaux sont fixés, serrer la connexion avec des « hose-clamp ». Le tuyau entrant du filtreur doit être de taille 1/4" et doit être connecté au tuyau de la pompe sortant grâce aux adaptateurs.

Connecter la pompe 2 au système d'arrosage

Il vous aussi la quantité désirée de tuyaux 3/8" à connecter à la pompe par les deux embouts de la pompe. Après avoir connecté les tuyaux à la pompe, vous devez insérer le tuyau (entrant) de la pompe à la connexion de métal situé en dessous du bac bleu (réservoir). Le tuyau (sortant) de la pompe doit être connecté au tuyau de métal (entrant) d'arrosage grâce aux adaptateurs.

Pompes

Introduction

Le système comprend 2 pompes, la pompe #1 : 5L/min, la pompe #2 : 24L/min. Ces deux pompes doivent être remplies d'eau avant de les démarrer. Aussi, avant l'hiver, il faut vider l'eau restante dans les pompes.

Emplacement

La pompe #1 doit être fixée en dessous à gauche du réservoir en ayant les tuyaux vers l'extérieur du bac. La pompe #2 doit être fixée sur une barre de métal vissée en dessous du quai. La plaque de métal est d'environ 10/70cm et est vissée sur sa longueur sur les parois intérieures du quai en dessous d'un des trous de calvettes. Pour visser les pompes, il y a sur eux des petites « pattes » avec des trous pour y insérer des vis et visser ceux-ci.



Utilisation

La pompe #1 sert à transporter l'eau qui contient les engrais du réservoir jusqu'au système des buses. Puisqu'il n'est pas nécessaire d'avoir une grosse pression pour y amener cette eau il est conseillé d'utiliser cette pompe de 5L/min. La pompe #2 doit transporter l'eau de la rivière jusqu'au filtreur. Pour faire cette opération, il est recommandé d'utiliser une pompe au-dessus de 20L/min puisque qu'une grande résistance est appliquée par l'entrée d'eau du filtreur. **IMPORTANT** : le filtreur n'est pas compris avec, la pompe #2 est essentielle à l'entièreté du système.

Système de burette

Burette

En cas de bris de burette, il est possible d'effectuer un remplacement en commander une burette 50ml puis en la sectionnant à la ligne de 35 ml. Important de noter qu'il est préférable de sectionner ces dernières avec des outils spécialisés pour éviter un bris imprévu.

Aiguille

En cas de perte d'une aiguille, vérifier dans la boîte étanche qu'il n'en reste effectivement plus avant d'en commander. Il est important d'ajouter de l'adhésif à usage maritime (Goop) aux 2 extrémités de ces dernières pour s'assurer de leur maintien (voir figure #1).



Servomoteur

Dans l'éventualité d'un bris de servomoteur, il est possible d'en racheter (https://components101.com/sites/default/files/component_datasheet/MG996R-Datasheet.pdf) Il est néanmoins possible que le problème puisse être réglé plus simplement. Exemple : si l'aiguille est sortie de son axe avec le servomoteur il est possible de le réinsérer puis de réappliquer de l'adhésif marin (voir Figure #1). Dans le cas où un remplacement est nécessaire, vous devez retirer les embouts pour exposer le câble conducteur des fils.

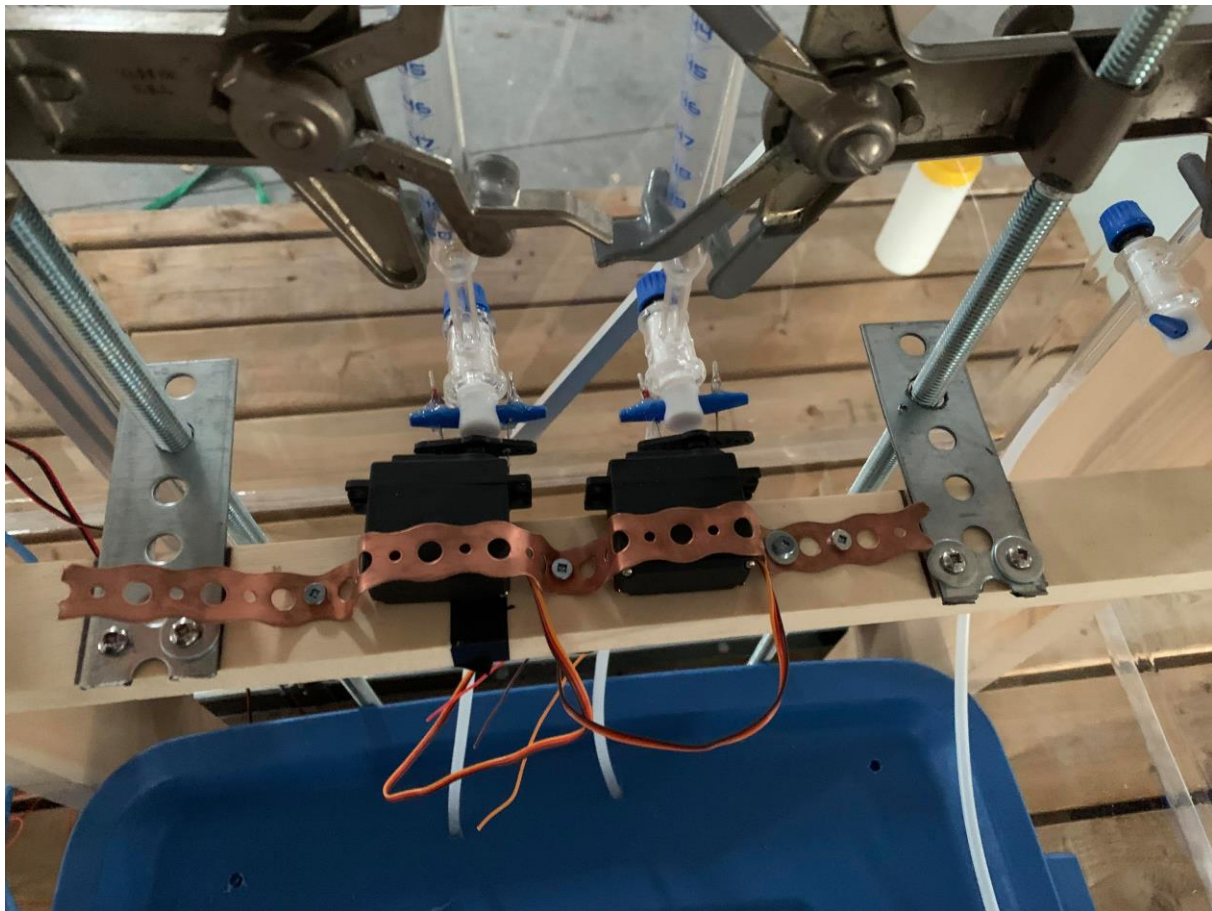


Figure #1 : Aiguille et servomoteur

Support à burette

Les supports à burettes sont composés de 2 parties, premièrement une tige métallique filetée de diamètre 7/16" et de pince à burette de laboratoire. Les premières sont trouvables dans toutes bonnes quincailleries et ne nécessitent aucune manipulation supplémentaire pour les remplacer. Les seconds devront être commandés en ligne dans l'éventualité d'un bris. Il sera toutefois nécessaire de retirer la planche supportant les servomoteurs avant d'installer l'une ou l'autre des pièces.

Calvettes

Rotation des calvettes

Il est possible que le roulement à billes qui permettent la rotation des calvettes soient bloqués par des débris ou par l'oxydation. Pour résoudre ce problème, 2 options émergent : remplacer la pièce défectueuse ou les graisser. Dans le premier cas, le roulement peut être usiné à partir du document qui se trouve si le GitHub du projet. Dans l'autre cas, de la graisse standard peut être appliquée pour permettre une rotation.



Calvettes physiques

Si dans un cas extrême une clavette doit être remplacée, ces dernières sont en réalité des ponceaux de 16". Il sera nécessaire par la suite de trouser ce dernier pour pouvoir y insérer les pots à plantes.

Partie 2 : Électronique

La boîte de relais

La boîte de relais se situe près des valves et s'occupe d'activer la pompe du filtreur, la pompe des buses ainsi que les 3 valves qui contrôlent les buses. Les câbles noirs sont le responsable pour la mise à terre « ground ». Les câbles en dessous de la boîte qui contient les relais alimente les relais avec du 12V. Les câbles sortant du haut de la boîte de relais sont utilisés afin de contrôler les relais. Chaque couleur a un contrôle sur un élément différent de la serre.



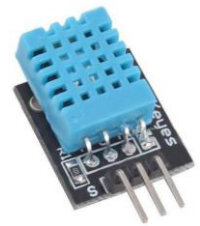
- Rouge : Pompe du filtreur
- Jaune : Pompe des buses
- Orange : La buse numéro 1 (valve de gauche)
- Vert : La buse numéro 2 (valve au centre)
- Bleu : La buse numéro 3 (valve de droite)



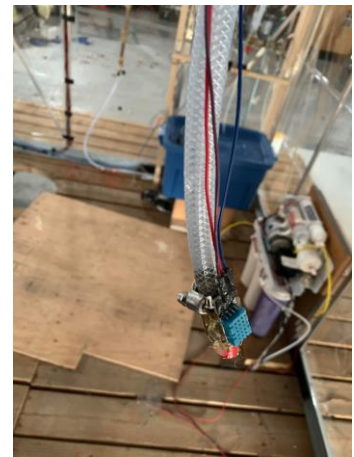
Entretien des capteurs

DHT11

Cette pièce sert à mesurer la température en degrés Celsius et le pourcentage d'humidité. La tige avec le symbole « + » doit être reliée à l'entrée 5 volts du Arduino, la tige avec le symbole « - » doit être reliée au « ground » du Arduino et la tige avec le symbole « OUT » doit être reliée à la tige identifiée DHT11_PIN, SECOND_DHT11_PIN ou THIRD_DHT11_PIN dans le code du microcontrôleur appelé Input Arduino.



La pièce est collée avec du goop sur chacune des buses et les fils doivent passer par le tuyau de la buse pour éviter de les déchirer lors des rotations de la calvette.



Capteur de niveau d'eau

Cette pièce doit être fixée à la verticale, les trois pins vers le haut, et permet de lire un entier entre 0 et 625 correspondant à la portion de la pièce qui est trempée dans l'eau.

La pin - doit être reliée au ground du Arduino des input et le pin S, au pin noté WATER_SENSOR_PIN ou WATER_SENSOR2_PIN dans le code du InputArduino. Il est important de noter que, pour éviter des problèmes de corrosion, la tige « + » de la pièce n'est pas reliée au 5 volts du Arduino, mais bien à la pin notée WATER_SENSOR_POWER_PIN ou WATER_SENSOR2_POWER_PIN. Ainsi, le capteur n'est pas continuellement alimenté, mais est alimenté seulement lors de la prise de lectures.



Deux de ces pièces sont fixées dans le bac d'eau avec de la goop. Il est important que le capteur correspondant à waterLevel1 dans le code ait celui du haut, et que le capteur correspondant à waterLevel2 dans le code soit celui du bas.

Le capteur du haut ne doit pas être fixé en haut complètement, pour laisser du jeu pour l'ajout d'engrais et pour le temps de réaction du filtreur et de l'entrée d'eau.



Il faut toujours s'assurer que les connexions entre les fils et le capteur soient scellées avec du goop pour les protéger de l'eau.

Capteur de pH

Ce capteur prend des mesures de pH dans le bac d'eau et la valeur lue dans la variable `phValue` du Arduino des input correspond à la moyenne des mesures prises par le capteur à chaque itération du Arduino. Le tube du capteur doit être relié à la puce par la connexion correspondante.



La pin VCC doit être relié au 5v du Arduino des input. Il y a deux pins ground qui doivent être reliés au ground du Arduino, une pour le tube et une pour la puce. Finalement, la pin P0 doit être reliée à la pin notée `PH_SENSOR` dans le code du `InputArduino`. Les pins T0 et D0 ne seront pas utilisés.

La puce doit être fixée avec de la goop à l'extérieur du bac, et le tube doit être à l'intérieur du bac.

Relais

Cette pièce sert à activer et désactiver les pompes et les valves selon un courant analogue envoyé par le Arduino. Du côté des pins, on doit reliée celle notée « + » ou VCC au 5 volts du Arduino des outputs, la pin noté `gnd` au ground de ce Arduino, et l'autre à la pin identifié par `FILTER_PUMP_RELAY_PIN` (pour la pompe qui active l'entrée d'eau et le filtreur), `SPRAYER_PUMP_RELAY_PIN` (pour la pompe qui envoie l'eau aux buses) ou à une des trois pins identifiés par `SPRAYER_SWITCH_PIN` (les relais qui activent les valves pour chacune des trois buses).



De l'autre côté du relais (celui sur la photo), il y a 3 entrées pour le fil sectionné alimentant le dispositif voulu. Il faut insérer les 2 sections de fils dans les deux entrées identifiées par un dessin de circuit fermé. Ainsi, lorsque le Arduino va envoyer du courant, le circuit va être ouvert (les dispositifs désactivés) et lorsque le Arduino va arrêter d'envoyer du courant, le circuit va être fermé et le dispositif correspondant, activé.

Servo-moteurs

Cette pièce sert à ouvrir et fermer les valves des burettes qui distribuent l'engrais. Il faut choisir une tête de servomoteur à 2 pales et relier ces pales à la valve de la burette à l'aide d'aiguilles.

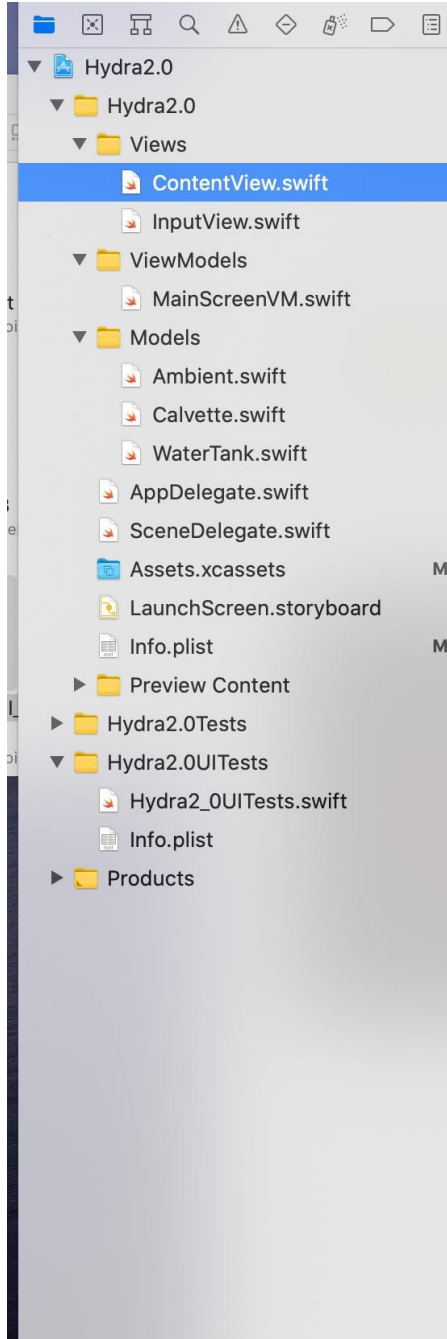


Le servomoteur a 3 fils. Celui noté + doit être relié au 5 volts du Arduino des outputs, celui noté « - » au ground du Arduino, et le 3e à la tige identifiée par SERVO1_PIN, SERVO2_PIN, SERVO3_PIN ou SERVO4_PIN dans le code du OutputArduino.

Partie 3 : informatique

Maintenance de l'app mobile

Survol de l'application



Cliquez sur l'icône de dossier (ici en bleu en haut à gauche) pour faire apparaître l'arborescence des fichiers de l'application.

Dans le dossier « Views » se trouvent les fichiers qui définissent l'aspect visuel de l'application.
« ContentView.swift » représente la page d'accueil et
« InputView.swift », la page des configurations.

Dans le dossier « Models » se trouvent les classes qui contiennent les données de l'application à un moment précis, ainsi que les fonctions qui créent ces données à partir des réponses obtenues du serveur.

Dans le dossier « ViewModels » se trouve le fichier « MainScreenVM.swift », qui effectue les requêtes au serveur.

Finalement, dans le dossier « Assets.xcassets » se trouvent les images associées à l'application.

Recevoir les données du serveur

```
func callHydraConfigApi() {
    let url = URL(string: "http://134.122.126.29/api/v1/config")!
    let task = URLSession.shared.dataTask(with: url) { (data, response, error) in
        guard let data = data else { return }
        _ = String(data: data, encoding: .utf8)
        self.handleHydraConfigRequest(data: data)
    }
    task.resume()
}

func callHydraThresholdsApi() {
    let url = URL(string: "http://134.122.126.29/api/v1/thresholds")!
    let task = URLSession.shared.dataTask(with: url) { (data, response, error) in
        guard let data = data else { return }
        _ = String(data: data, encoding: .utf8)
        self.handleHydraThresholdsRequest(data: data)
    }
    task.resume()
}
```

Voici les requêtes qui vont chercher les données sur la serre du serveur. Les deux URL correspondent à l'adresse du serveur.

Envoyer les données au serveur

```
func sendData(data: [String: Any]) {
    let url = URL(string: "http://134.122.126.29/api/v1/thresholds")!
    let parameters = data
    var request = URLRequest(url: url)
    request.httpMethod = "POST"
    request.httpBody = parameters.percentEncoded()
    let task = URLSession.shared.dataTask(with: request) { data, response, error in
        guard let data = data, error == nil else {
            print(error?.localizedDescription ?? "No data")
            return
        }
        let responseJSON = try? JSONSerialization.jsonObject(with: data, options: [])
        if let responseJSON = responseJSON as? [String: Any] {
            print(responseJSON)
            if let httpResponse = response as? HTTPURLResponse {
                print("statusCode: \(httpResponse.statusCode)")
            }
        }
    }
    task.resume()
}
```

Voici la requête qui est effectuée lorsque vous appuyez sur le bouton sauvegarder de l'écran des configurations.

Données sur la température ambiante

```
func callOpenWeatherApi() {
    let url = URL(
        string:
            "https://api.openweathermap.org/data/2
            .5/weather?q=Saint-Ours
            ,ca&units=metric&appid=0423f1d8ad5cbacd8a4f2a0776e44a71"
    )!
    let task = URLSession.shared.dataTask(with: url) { (data, response, error) in
        guard let data = data else { return }
        self.handleOpenWeatherRequest(data: data)
    }
    task.resume()
}
```

L'application effectue des requêtes à « OpenWeatherMapAPI » pour connaître la température et l'humidité ambiantes. Si vous pensez déménager la serre, vous n'avez qu'à changer le nom de la localité dans l'URL de la requête.

L'intervalle entre les requêtes

```
init() {
    timer = Timer.scheduledTimer(timeInterval: 5, target: self, selector:
        #selector(update), userInfo: nil, repeats: true)
    timer.fire()
}
```

Ceci peut être très intéressant à connaître pour l'utilisateur. Ici, vous pouvez changer la valeur de la variable « `timeInterval` » qui correspond au temps en secondes entre chaque requête au serveur.

Conversion JSON en object

```
func handleHydraConfigRequest(data: Data) {
    do {
        let decoded = try JSONDecoder().decode(CalvetteConfig.self, from: data)
        DispatchQueue.main.async {
            self.calvettesConfig = decoded
        }
    } catch {
        print(error)
    }
}
```

Fichier mainScreenVM

```
required init(from decoder: Decoder) throws {
    let container = try decoder.container(keyedBy: CodingKeys.self)
    temperature1 = try container.decode(Int.self, forKey: .temperature1)
    humidity1 = try container.decode(Int.self, forKey: .humidity1)
    temperature2 = try container.decode(Int.self, forKey: .temperature2)
    humidity2 = try container.decode(Int.self, forKey: .humidity2)
    temperature3 = try container.decode(Int.self, forKey: .temperature3)
    humidity3 = try container.decode(Int.self, forKey: .humidity3)
}
```

Fichier Calvette

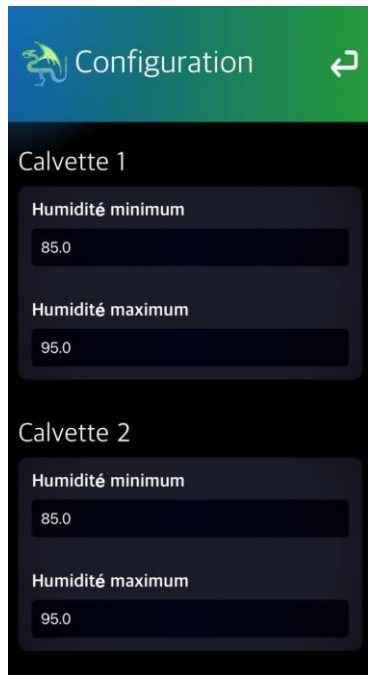
Ce sont ces fonctions qui convertissent l'objet Json reçu du serveur en un objet constitué de classes qui est utilisable dans les fichiers Views.

Page des views

```
NavigationView {
  ScrollView {
    VStack {
      HStack {
        Image("hydraLogo").resizable().frame(width: 50, height: 50)
        Text("Hydra").foregroundColor(Color.white).font(
          .custom("AppleSDGothicNeo-Light", size: 36))
        Spacer()
        NavigationLink(destination: InputView(showSelf: $showInput), isActive: $showInput) {
          Image(systemName: "gear").resizable().frame(width: 30, height: 30).foregroundColor(
            .white
          ).font(Font.system(.largeTitle).bold()).padding(.leading, 40)
        }
      }.frame(minWidth: 0, maxWidth: .infinity, minHeight: 90, alignment: .center).padding(20)
      .background(
```

L'application a été construite à partir de « SwiftUI ». Si vous avez les connaissances nécessaires, c'est sur les pages « ContentView » et « InputView » que vous pourrez configurer le look de l'application.

Interface utilisateur



Voici la manière qu'est affichée les informations que serveur reçoit de la serre. Sur cette page il y a la température, l'humidité de l'endroit (extérieur), les informations sur les calvettes (température et humidité). Sur la page de configuration on peut modifier les infos sur les calvettes.



Maintenance du serveur

Base de données

La base de données étant basée sur « mongodb » n'est pas un schéma relationnel, mais bien « nosql ». Soit un objet JSON qui est modifié et réinséré dans la base de données. Il a deux collections, une pour les données de la serre (config) et une pour les thresholds de la serre.

Pour y accéder, le nom de la base de données est Hydra et la collection utilisée se nomme « threshold ». L'objet json config va s'en suit :

- 1- La température de la calvette # 1
- 2- La température de la calvette # 2
- 3- La température de la calvette # 2
- 4- L'humidité actuelle de l'humidité de la calvette #1
- 5- L'humidité actuelle de l'humidité de la calvette #2
- 6- L'humidité actuelle de l'humidité de la calvette #3
- 7- Le ph du bac d'eau
- 8- Le niveau de l'eau du capteur #1
- 9- Le niveau de l'eau du capteur #2

L'objet json threshold va s'en suit:

- 1- La valeur maximale de l'humidité de la calvette #1
- 2- La valeur minimale de l'humidité de la calvette #1
- 3- La valeur maximale de l'humidité de la calvette #2
- 4- La valeur minimale de l'humidité de la calvette #2
- 5- La valeur maximale de l'humidité de la calvette #3
- 6- La valeur minimale de l'humidité de la calvette #3
- 7- Le ph désiré du bac

Méthodes utilisées

Pour la plupart des utilisations nécessaires pour le fonctionnement de la serre, il ne suffit que de 2 requêtes HTTP pour afficher et modifier les données. Une se servant de la méthode GET et l'autre POST.

GET

Pour accéder aux informations actuelles de la serre vous pouvez effectuer une requête HTTP à cette URL : <http://134.122.126.29/api/v1/config/> ou <http://134.122.126.29/api/v1/thresholds/>

Les données sont renvoyées selon l'objet énoncé plus haut.

POST

Pour effectuer des modifications sur la base de données veuillez effectuer une requête POST sur la même URL soit : <http://134.122.126.29/api/config/> ou <http://134.122.126.29/api/thresholds/>

La réponse à cette requête si tout c'est bien passé est le revoie de l'objet.

Maintenance des microcontrôleurs

Relier les arduinos

On utilise 2 Arduino. Le premier récolte les données et les envoie au Raspberry Pi, et envoie aussi des commandes à l'autre Arduino. Le deuxième reçoit les commandes de l'Arduino des inputs et s'en sert pour activer les relais et les servomoteurs.

Les deux « Arduinos » doivent être reliés ensemble selon la méthode décrite plus loin dans la section LES CONNEXIONS SÉRIELLES.

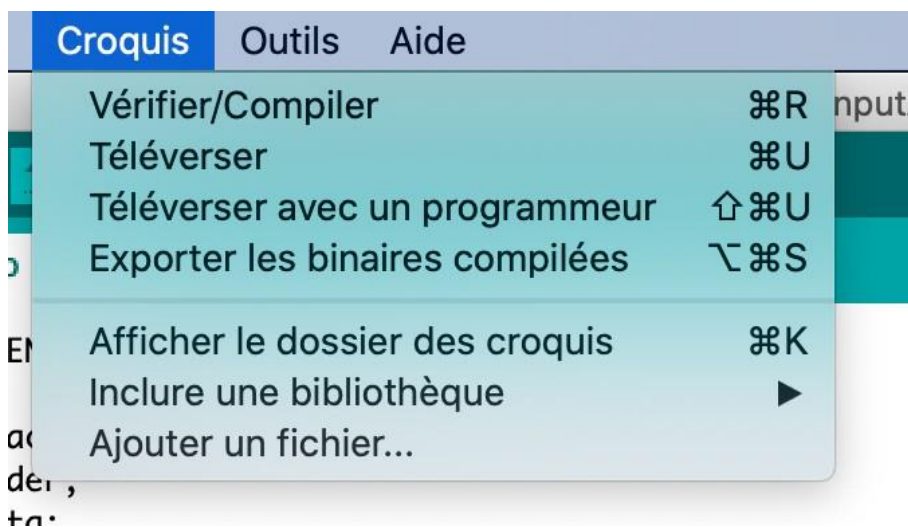
Reprogrammer un Arduino

Si vous voulez effectuer des manipulations dans le code, il faut tout d'abord savoir reprogrammer le Arduino pour que ces modifications s'appliquent. Pour ce faire, aller dans l'onglet Outils du menu. Choisissez :

Le type de carte : Arduino Uno

Le port : Arduino ISP pour le Uno

Le programmeur Arduino ISP pour le Uno



Ensuite, il suffit de choisir **téléverser** dans le menu Croquis pour effectuer la programmation.

Connexions sérieelles

```
#define DHT11_PIN 8
#define SECOND_DHT11_PIN 9
#define THIRD_DHT11_PIN 10
#define DHTTYPE DHT11
#define WATER_SENSOR_PIN A1
#define WATER_SENSOR_POWER_PIN 7
#define WATER_SENSOR2_PIN A2
#define WATER_SENSOR2_POWER_PIN 4
#define PH_SENSOR A0

DHT dht1(DHT11_PIN, DHTTYPE);
DHT dht2(SECOND_DHT11_PIN, DHTTYPE);
DHT dht3(THIRD_DHT11_PIN, DHTTYPE );

SoftwareSerial mySerial(2, 3);

long receiveTimerStart = 0;

// Données de l'API
float humidityLowerTreshold1 = 85;
float humidityUpperTreshold1 = 95;
float humidityLowerTreshold2 = 85;
float humidityUpperTreshold2 = 95;
float humidityLowerTreshold3 = 85;
float humidityUpperTreshold3 = 95;
float waterLevelLowerTreshold = 70;
float waterLevelUpperTreshold = 550;
float phTreshold = 6.5;

// Données des capteurs
float phCalibration = 21.20;
float phValue;
float temperature1;
float temperature2;
float temperature3;
float humidity1;
float humidity2;
float humidity3;
float waterLevel1;
float waterLevel2;
```

Arduino des Input

On utilise la librairie « SoftwareSerial » pour créer des ports sériels supplémentaires. Ainsi, le port de base (Serial), reste réservé pour l'envoi de données au Raspberry Pi.

Le port « mySerial » sert à la communication entre les 2 « Arduinos ». Dans le code ci-dessus, on voit que le RX (receveur) du Arduino des input correspond à la pin 2 et le TX (transmetteur) à la pin 3. Sur le Arduino des outputs, le RX de « mySerial » correspond à la pin 2, et le TX à la pin 3.

Il est important de relier le RX d'un Arduino au TX de l'autre Arduino et vice-versa. Ainsi, si on ne modifie pas le port sériel, la pin 2 de l'Arduino des inputs sera reliée à la pin 3 de l'Arduino des outputs, et la pin 3 de l'Arduino des inputs sera reliée à la pin 2 de l'Arduino des outputs.

Les pins

```
#define DHT11_PIN 8
#define SECOND_DHT11_PIN 9
#define THIRD_DHT11_PIN 10
#define DHTTYPE DHT11
#define WATER_SENSOR_PIN A1
#define WATER_SENSOR_POWER_PIN 7
#define WATER_SENSOR2_PIN A2
#define WATER_SENSOR2_POWER_PIN 4
#define PH_SENSOR A0

DHT dht1(DHT11_PIN, DHTTYPE);
DHT dht2(SECOND_DHT11_PIN, DHTTYPE);
DHT dht3(THIRD_DHT11_PIN, DHTTYPE );

SoftwareSerial mySerial(2, 3);
```

Ici, vous pouvez changer l'endroit où se connecte chaque capteur sur le Arduino. Voir la section sur les capteurs pour savoir quelle variable correspond à quel capteur.

Les seuils

```
// Données de l'API
float humidityLowerTreshold1 = 85;
float humidityUpperTreshold1 = 95;
float humidityLowerTreshold2 = 85;
float humidityUpperTreshold2 = 95;
float humidityLowerTreshold3 = 85;
float humidityUpperTreshold3 = 95;
float waterLevelLowerTreshold = 70;
float waterLevelUpperTreshold = 550;
float pHThreshold = 6.5;
```

Vous pouvez changer ici les seuils correspondant à l'activation des relais et des servomoteurs. Le seuil d'humidité minimum active la buse d'arrosage. Le seuil d'humidité maximum la désactive. Le seuil de niveau d'eau minimum active la pompe du filtreur et celui du niveau d'eau maximum la désactive. Finalement, le seuil de pH active l'ajout de « pHDown ».

Les lectures

```
temperature1 = dht1.readTemperature();
temperature2 = dht2.readTemperature();
temperature3 = dht3.readTemperature();
```

```
humidity1 = dht1.readHumidity();
humidity2 = dht2.readHumidity();
humidity3 = dht3.readHumidity();
```

```
waterLevel1 = readWaterLevel();
waterLevel2 = readSecondWaterLevel();
```

```
pHValue = readPh();
```

Arduino des inputs

```
float readPh(){
    int buf[10];
    for(int i=0;i<10;i++){
        buf[i] = analogRead(PH_SENSOR);
        //delay(30); // use for a bigger average
    }
    bSort(buf, 10);
    return calculateAveragePh(buf, 10);
}

float calculateAveragePh(int *buf, int l){
    unsigned long int avgValue = 0;
    for(int i=2;i<l-2;i++){
        avgValue += buf[i];
    }
    float pHVol=(float)avgValue*5.0/1024/6;
    float pHValue = -5.70 * pHVol + pHCalibration;
    return pHValue;
}

void bSort(int *buf, short l){
    int temp;
    for(int i=0;i<l-1;i++){
        for(int j=i+1;j<l;j++){
            if(buf[i]>buf[j]){
                temp=buf[i];
                buf[i]=buf[j];
                buf[j]=temp;
            }
        }
    }
}
```

Ce sont ces fonctions sur l'Arduino des inputs qui permet de prendre les mesures. Veuillez noter que pour le pH, il y a une fonction pour trier les lectures (bSort) et une fonction pour calculer la moyenne des lectures (calculateAveragePh). Ainsi, la valeur de pH obtenue à chaque itération du Arduino est une moyenne.

Les commandes

```
if ( waterLevel1 > waterLevelUpperTreshold ){  
|  
    mySerial.println("Filter pump off");  
}  
  
if(waterLevel2 < waterLevelLowerTreshold ){  
    mySerial.println("Filter pump on");  
}
```

Arduino des inputs

```
while(mySerial.available() > 0){  
    input = mySerial.readStringUntil('\n');  
    input.trim();  
    Serial.println("Input: " + input);  
    if ( input.equals("Filter pump on") ){  
        digitalWrite(FILTER_PUMP_RELAY_PIN, LOW);  
        fertilizer1Available = true;  
    }  
}
```

Arduino des outputs

Quand les seuils sont atteints, le Arduino des inputs envoie des commandes sous forme de string au Arduino des output, qui exécute les commandes.

L'ajout d'engrais

```
boolean fertilizer1Available = true;  
boolean fertilizer1Added = false;  
boolean fertilizer2Available = false;  
boolean fertilizer2Added = false;  
boolean fertilizer3Available = false;  
boolean fertilizer3Added = false;  
long fertilizerTimerStart = 0;
```

Les engrais s'ajoutent un après l'autre lorsque le seuil maximum de niveau d'eau est atteint. Le premier engrais s'ajoute selon le temps de déversement correspondant (voir section suivante). Lorsque le temps est écoulé, la variable fertilizer1Added passe à true, la valeur fertilizer1Available passe à false, et la valeur fertilizer2Available passe à true.

Le temps de déversement des engrais

```
void addFertilizer1(){  
    int interval = 82000; //82000  
  
    if (millis() - fertilizerTimerStart >= interval ){  
        Serial.println( millis() - fertilizerTimerStart );  
        fertilizerTimerStart = millis();  
        if(fertilizer1Added){  
            servo1.write(90);  
            fertilizer1Available = false;  
            fertilizer1Added = false;  
            fertilizer2Available = true;  
        }  
    } else{  
        servo1.write(0);  
        fertilizer1Added = true;  
    }  
}
```

Arduino des outputs

Il y a 3 fonctions addFertilizer() dans le code du Arduino des outputs, une pour chaque burette. Dans les fonctions correspondant aux engrais, la variable interval correspond au nombre de millisecondes (1/1000 de secondes) pendant lesquelles chaque engrais va se déverser. Entrez le nombre de secondes voulues multipliées par 1000 et n'oubliez pas de **reprogrammer le Arduino** (voir section plus haut).

Il est important de ne pas toucher à l'intervalle dans la fonction addPhDown. L'intervalle de 500 millisecondes est vraiment le maximum pour ne pas que le phDown s'ajoute trop rapidement. Il ne sert à rien de choisir un intervalle plus petit parce que l'intervalle entre les itérations du Arduino est de 500 millisecondes.

L'envoi de données au Raspberry Pi

```
void sendData() {  
  String data = "{  
    \"temperature1\": \" + String(temperature1) +  
    \", \"temperature2\": \" + String(temperature2) +  
    \", \"temperature3\": \" + String(temperature3) +  
    \", \"humidity1\": \" + String(humidity1) +  
    \", \"humidity2\": \" + String(humidity2) +  
    \", \"humidity3\": \" + String(humidity3) +  
    \", \"ph\": \" + String(phValue) +  
    \", \"waterLevel1\": \" + String(waterLevel1) +  
    \", \"waterLevel2\": \" + String(waterLevel2) +  
    \"}\";  
  Serial.println("Data" + data);  
}
```

Les variables sont toutes envoyées au Raspberry Pi en même temps selon l'intervalle défini dans la fonction sendData de l'Arduino des inputs. Les variables sont envoyées dans le print du moniteur sériel.

La réception des données

Le Raspberry Pi écoute sur le moniteur sériel du Arduino des Inputs. Lorsque l'Arduino envoie les données, le Raspberry Pi les lit pour ensuite les envoyer au serveur dans la collection config.

```
import requests
import serial
import json

thresholdsUrl = "http://134.122.126.29/api/v1/thresholds"
configUrl = "http://134.122.126.29/api/v1/config"

if __name__ == '__main__':
    ser = serial.Serial('COM4', 9600, timeout=1)
    ser.reset_input_buffer()
    while True:
        if ser.in_waiting > 0:
            try:
                line = ser.readline().decode('utf-8').rstrip()
                if line == 'Envoi-moi les thresholds':
                    response = requests.get(thresholdsUrl).text
                    ser.write(str(response).encode('utf-8'))
                elif line.startswith('Data'):
                    data = json.loads(line[4:])
                    requests.post(configUrl, data=data, timeout=1)
            except:
                print("")
```

Communication externe (Raspberry Pi)

Le physique

Le « Raspberry Pi » est responsable de la communication entre le serveur sur le nuage et la serre Hydra. Il accomplit cela en communiquant avec un Arduino et avec le nuage à l'aide d'une connexion Wifi. Ainsi, le « Raspberry Pi » n'est qu'une passerelle entre plusieurs technologies qui fonctionnent différemment, un peu comme un traducteur.

La connexion au Arduino

L'Arduino des Inputs doit être branché dans le port USB du Raspberry Pi. Il faut brancher l'Arduino avant d'allumer le Raspberry Pi pour que le script marche.

La connexion Wifi

La connexion au Wifi devrait se faire automatiquement après un certain temps. Si c'est la première fois qu'il est allumé ou connecté au réseau en question, il faudra le connecter, en utilisant un écran, un clavier et une souris.

L'alimentation

Pour fonctionner correctement, le Raspberry Pi a besoin d'être branché à l'électricité. Sa source de courant se situe sous la trappe avec le reste des composants électriques. Il peut être fermé ou rallumé à l'aide de l'interrupteur vert dans le panneau de contrôle.

Perte de connexion (dépannage)

Dans ce cas, il est probable que votre Hydra se soit déconnectée de son serveur. Pour la reconnexion, vous devez fermer le Raspberry Pi (se contrôle avec l'interrupteur vert dans le panneau de contrôle) et les Arduinos (se contrôle avec l'interrupteur rouge toujours dans le panneau de contrôle). Par la suite, attendez environ cinq secondes. Pour terminer, vous pouvez réactiver les Arduinos (bleu) suivis du Raspberry Pi (Vert). Votre Hydra devrait recommencer à communiquer.

Schéma de communications

Explication générale

Si un problème semble émerger de la communication entre 2 composants, il est possible que des ajustements doivent être faits à un niveau particulier. Pour ce faire, ce guide va tenter de vous expliquer la communication complète entre tous les différents composants. Pour une vue d'ensemble des communications vous pouvez vous référer à la figure #4.

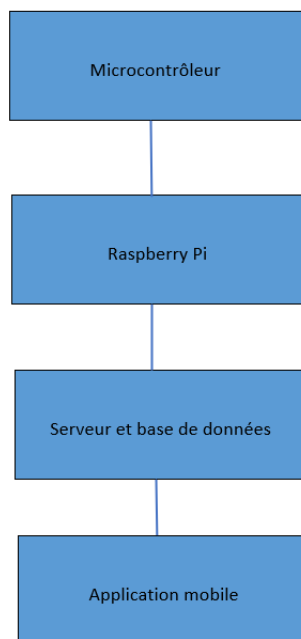


Figure #4 : Schéma des communications

#1 Arduino

En premier lieu, le Arduino connecté au Raspberry Pi peut recevoir des données également en envoyé. Si les informations des capteurs ne semblent pas être valides, la première étape est de vérifier l'intégrité de ces derniers sur le Arduino.

#2 Raspberry Pi

Par la suite, ces informations sont passées au Raspberry Pi. Très peu de processus et de calcul se passe à ce niveau, à moins d'une déconnexion complète, le problème n'est probablement pas de ce côté. (Pour résoudre une déconnexion complète veuillez-vous référer à la section : Perte de connexion (dépannage))

#3 Serveur

Cependant, il est tout de même possible qu'un problème émerge au niveau du serveur HTTP. Par exemple, si la modification de la serre à partir de l'application mobile ne s'exécute pas ou si les données sont reçues, mais pas actualisé. Dans ce cas il est nécessaire de s'assurer que la base de données est encore fonctionnelle et que l'objet JSON est encore adéquat (voir section : Base de données).

#4 Application mobile

Et finalement, l'application mobile. Cette dernière pourrait être la source de certains problèmes notamment par son côté réseau, il est possible que le réseau que vous utilisez peut bloquer la requête de cette dernière au serveur causant un dysfonctionnement de toutes les fonctions de l'application mobile.