**Projet SansFil : Système de sécurit—je veux dire, système d’arrosage**

**Travail présenté à**

Frédéric Daigle

**Dans le cadre du cours**

243-510-MA COMMUNICATIONS NUMÉRIQUES SANS FIL

gr. 00002

**Réalisé par**

Cédric Vincent

Carlos Mesa

**Collège de Maisonneuve**

09 décembre 2019

Table des matières

[Introduction 2](#_Toc27385977)

[Schéma Bloc: 2](#_Toc27385978)

[Liste des choses obtenues : 3](#_Toc27385979)

[Module WIFI : 3](#_Toc27385980)

[Capteur résistif : 3](#_Toc27385981)

[Valve : 3](#_Toc27385982)

[Tuyau d’arrosage: 3](#_Toc27385983)

[Recherches faites : 4](#_Toc27385984)

[Détails d’access web : 4](#_Toc27385985)

[Mention de PHP : 4](#_Toc27385986)

[Possibilité d’écraser l’API pour le réécrire : 4](#_Toc27385987)

[Adresse serre : 4](#_Toc27385988)

[Journal de bord : 4](#_Toc27385989)

[Premières semaines : 4](#_Toc27385990)

[Projet # 1serrure : 4](#_Toc27385991)

[Semaines : 4](#_Toc27385992)

[Photo du montage matériel final (sauf connecteurs de la valve): 5](#_Toc27385993)

# Introduction

Nous avons été chargé à la conception d’un nouveau système à l’usage de la serre. Notre projet servira donc à ajouter une fonctionnalité nouvelle à cet environnement.

La documentation si présente décrit le processus de conception de notre projet. À cet effet, il comportera les sections suivantes : le schéma de conception logique du système, la liste du matériel mobilisé pour le projet, les recherches préliminaires accomplies, ainsi que journal de bord décrivant l’ensemble du processus employé.

# Schéma Bloc:

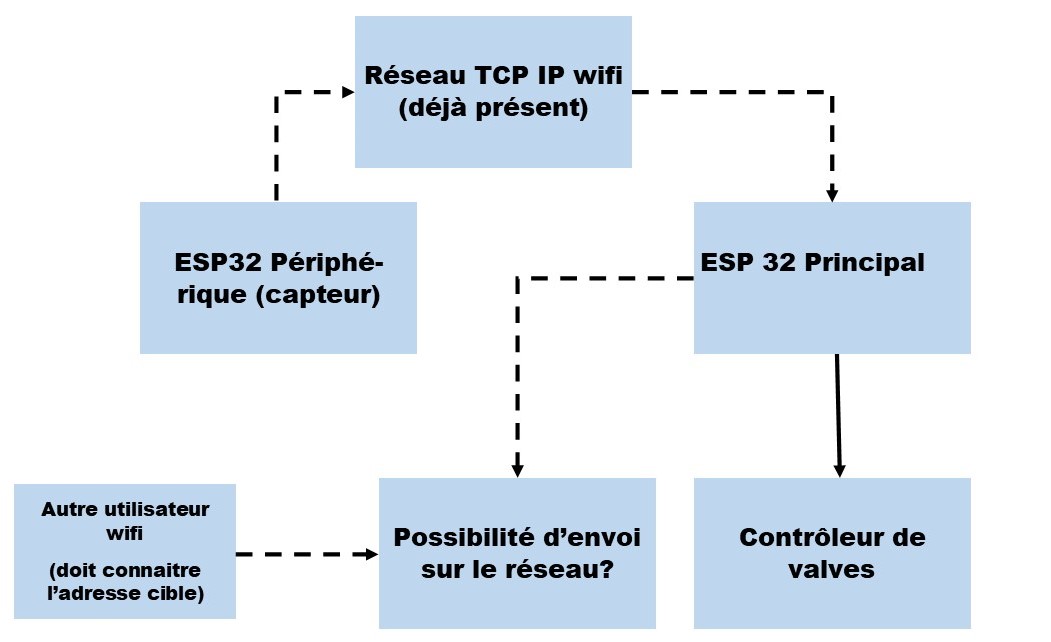


Figure L'envoi sur le réseau n'a pas été implementé

# Liste des choses obtenues :

## Module WIFI :

* **ESP32 ADAFRUIT Feather board.**

## Capteur résistif :

* <https://www.amazon.ca/-/fr/Gikfun-Moisture-Arduino-darrosage-Ek1099/dp/B06Y66TMTB?ref_=fsclp_pl_dp_3>

### Valve :

* <https://www.digikey.ca/product-detail/en/adafruit-industries-llc/997/1528-2003-ND/6827136>

## Tuyau d’arrosage:

Tuyau perforé de 25 pieds, canadian tire

# Recherches faites :

Détails d’access web :

<https://techtutorialsx.com/2017/05/20/esp32-http-post-requests/>

<https://stackoverflow.com/questions/14551194/how-are-parameters-sent-in-an-http-post-request>

## Mention de PHP :

<https://www.php.net/manual/en/mysqli.overview.php>

<https://randomnerdtutorials.com/esp32-esp8266-send-email-notification/>

<https://randomnerdtutorials.com/esp32-esp8266-mysql-database-php/>

Possibilité d’écraser l’API pour le réécrire : <https://www.hanselman.com/blog/HTTPPUTOrDELETENotAllowedUseXHTTPMethodOverrideForYourRESTServiceWithASPNETWebAPI.aspx>

## Adresse serre :

206.167.36.216

# Journal de bord :

## Premières semaines :

### Projet # 1serrure :

On a changé de projet en faveur de notre deuxième option, car une autre équipe a eu l’approbation du professeur pour réaliser un projet qui répondait déjà aux aspects qu’on voulait aborder avec notre première option.

Semaines avant novembre : recherches pour projet #2 et attente des pièces (ESP 32, Capteur d’humidité).

## Semaines :

* 28 octobre test des capteurs pour en choisir un.
  + Attente pour parler avec le professeur (redéfinition du projet).
  + Recherches sur Miwi en but d’intégration pour le fil de données
* 04 novembre tests esp 32 et potentiel ADC/I2C
  + Prototype de programme de contrôle des valves avec un réseau de 2 esp 32 :
    - 1 Access-point et un client. Fonctionne selon ADC (allumage d’un Del), manque quelques détails de mise en forme de la page du point d’accès.
  + Recherche d’information pour possible adaptation des capteurs sur les cartes déjà existantes.
  + Possible solution :
    - Demander à Philipe : un capteur de température, deux esp32.
    - Faire le design d’un PCB d’adaptation : 2 capteurs (humidité et température), un esp32 et un connecteur DB9. Pour faire la même communication que le senseur de base.
* 11 novembre création d’un proto level-shifter.
  + Esp 32 = 3.3V compatible vs I2C Miwi à 5V.
  + Recherche pour faire du I2C sur le esp32.
* 14 novembre
  + Abandon de l’idée d’I2C esp32 vers miwi en faveur de l’utilisation d’un PIC spécialisé en adc/i2c : le signal du capteur sera divisé en deux lignes paralèles : une pour la commande moteurs esp et une pour un feedback I2C vers miwi.

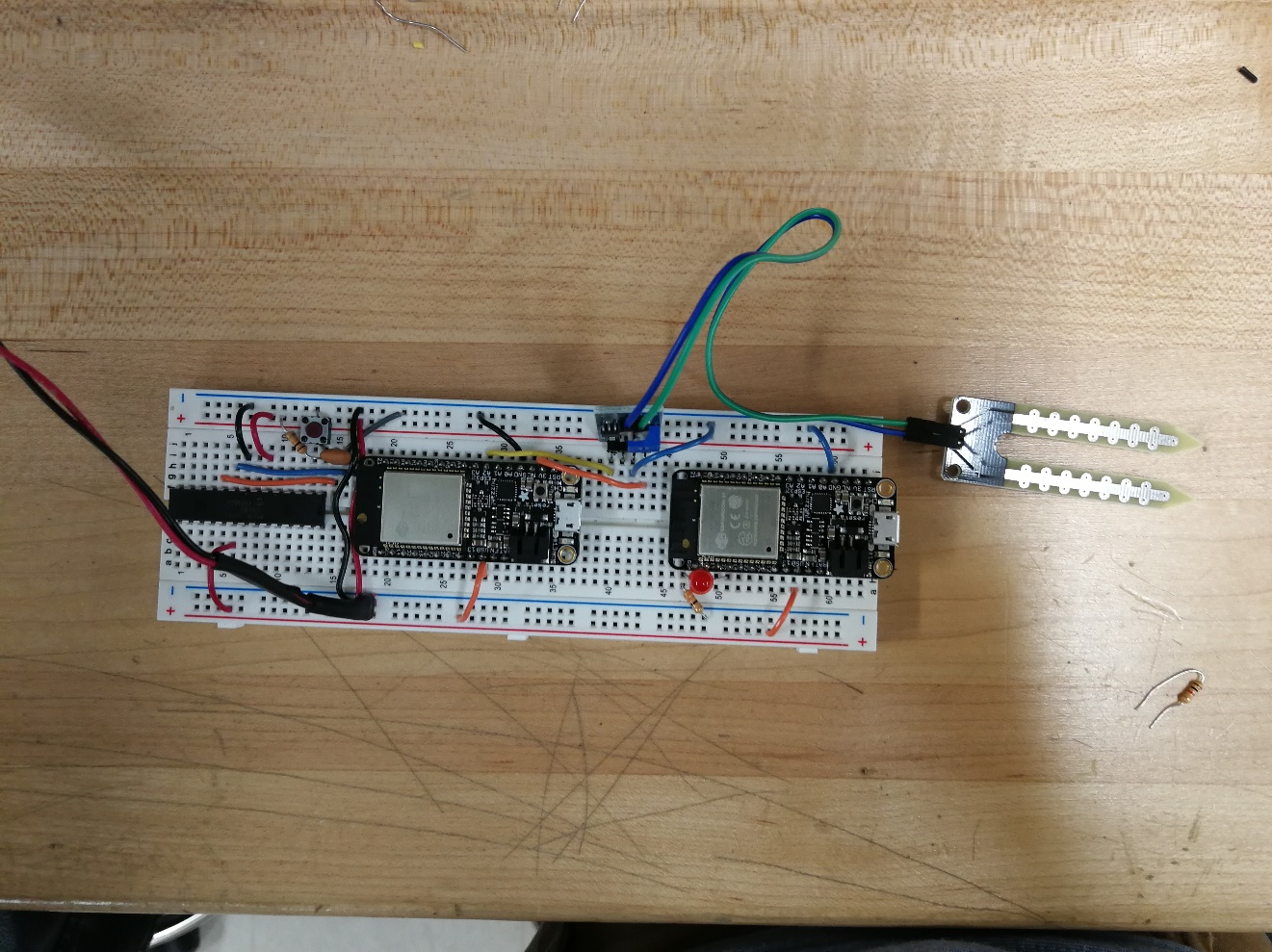
Notre proto-montage :

[Image à mettre]

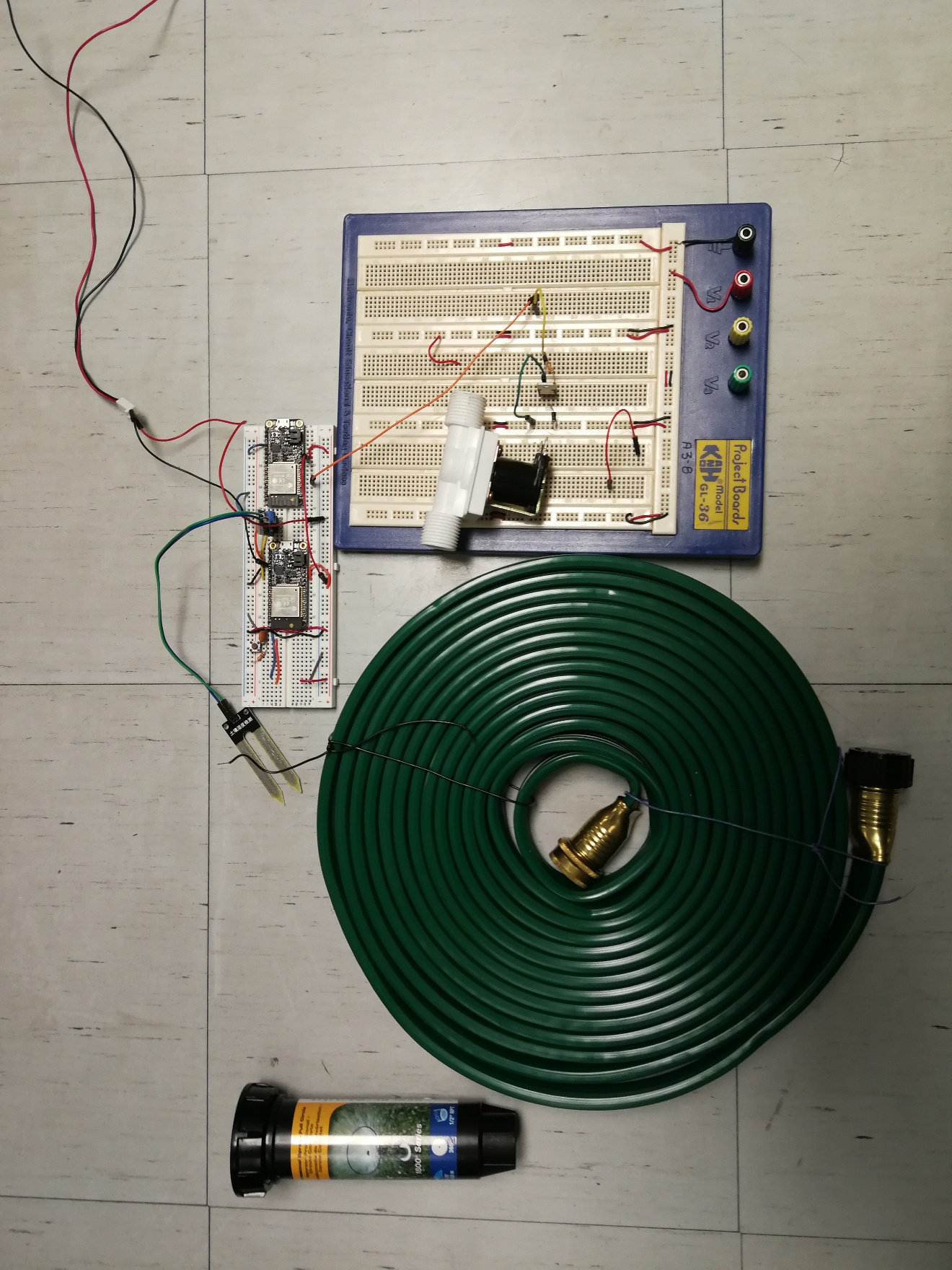
* 18 novembre
  + Lecture ADC sur pic fonctionnelle.
  + Tentative sur lien I 2C (Non opérationnel).
  + Début de création des PCB sur mutisme.
* 25 novembre
  + Discussion au sujet du design du projet : retrait de l’idée i2c pour passer directement sur l’ADC d’une carte Miwi.
  + Mise en commande des pièces liées au système d’arrosage.
  + Vérification de la carte capteur + tentative de reconfiguration (voir point #1).
  + Reprogrammation des modules PAN et FERMIER non fonctionnels afin de pouvoir tester notre circuit.
* 2 décembre
  + Tentative ADC sur Miwi infructueuse : valeurs de l’ensemble des lectures altérées (Température air et tout le reste).
  + Montage pour contrôler la valve, Tests du matériel d’arrosage concluant (merci à notre collègue Bruno pour le Transistor qui nous a fourni TIP120).
  + Création de PCB pour le projet sur muiltisim et ultiboard. Base sur la plaquette de prototypage (voir valve.ms13 et valve.ewprj ).
  + Réédition des paramètres du code pour assurer une meilleure sécurité du système (Arduino sur ESP32).
* 9 décembre
  + Réception de la tête d’arrosage.
  + On ne peut pas le tester car on avait pensé qu’elle n’arriverait pas à temps pour faire des tests, alors on n’a pas commandé le tuyau pour l’alimentation (on a un tuyau perforé pour la distribution de l’eau [dans le cas où l’arroseur serait trop puissant, mais pas d’acheminement.])

# Photo du montage matériel final (sauf connecteurs de la valve):

## Les esp32 de plus près :



## Montage sommaire :



12V in

Valve branché à la pin 1 du TIP120

et au 12Vin : Topologie actif-bas