**Sommaire**

[I. Introduction 2](#_Toc39594070)

[II. Réalisation du projet 3](#_Toc39594071)

[A. Rappel de la tâche de l’étudiant 3](#_Toc39594072)

[B. Communication entre les différents éléments 3](#_Toc39594073)

[C. Information de connexion serveur 3](#_Toc39594074)

[D. Structure du protocole de communication 3](#_Toc39594075)

[III. Plateau chauffant 3](#_Toc39594076)

[A. Schéma électrique du câblage 3](#_Toc39594077)

[B. Evolution de la température 3](#_Toc39594078)

[IV. Réalisation du programme de régulation 3](#_Toc39594079)

[A. Carte Raspberry 3](#_Toc39594080)

[B. Carte Arduino 3](#_Toc39594081)

[C. Journalisation 3](#_Toc39594082)

[V. Réalisation de l’application « Journalisation » 4](#_Toc39594083)

[VI. Test unitaire 4](#_Toc39594084)

[VII. Fiches recettes 4](#_Toc39594085)

[D. Connexion Raspberry 4](#_Toc39594086)

[E. Lancer le système de régulation 4](#_Toc39594087)

[F. Lecture des événement journalisés 4](#_Toc39594088)

[VIII. Conclusion 4](#_Toc39594089)

[G. Communication de groupe 4](#_Toc39594090)

[H. Regard critique du projet 4](#_Toc39594091)

[I. Connaissances apportées 4](#_Toc39594092)

[J. Ce qui me reste à faire 4](#_Toc39594093)

[K. Poursuite d’étude 5](#_Toc39594094)

[Annexe 5](#_Toc39594095)

# Etudiant 4 (Thomas CADEAU)

## Rappel du cahier des charges

### Aperçu des tâches réalisées

Dans le système Escape Game, ma partie de développement consiste à créer plusieurs programmes :

* Deux programmes pour deux mécanismes sur Arduino :
* Le mécanisme 5 : l’Eau, avec un capteur d’humidité (Langage Arduino ~C++)
* Le Mécanisme 9 : les 4 Éléments, avec un bouton poussoir (Langage Arduino ~C++)
* Une interface WEB pour la gestion des actionneurs (HTML/CSS/PHP):
  + - Envoyé un ordre d’activation ou de désactivation d’un actionneur à la Raspberry.
    - Création d’un serveur sockets pour l’envoie des ces ordres à la Raspberry. (Langage Python)

Une image contenant capture d’écran

Description générée automatiquement Le langage Arduino est utilisé avec Wire pour l’i2c et avec 0X16 pour le capteur d’humidité.

Ma tâche

Image 1 : Ma tâche personnelle

### Contraintes liées au développement

A cause du confinement le programme concernant le mécanisme 9 et la programmation par socket n’as pas pu être testé par manque de matériels.

# Réalisation du programme Mécanisme 5 : l’Eau

## Schéma câblage

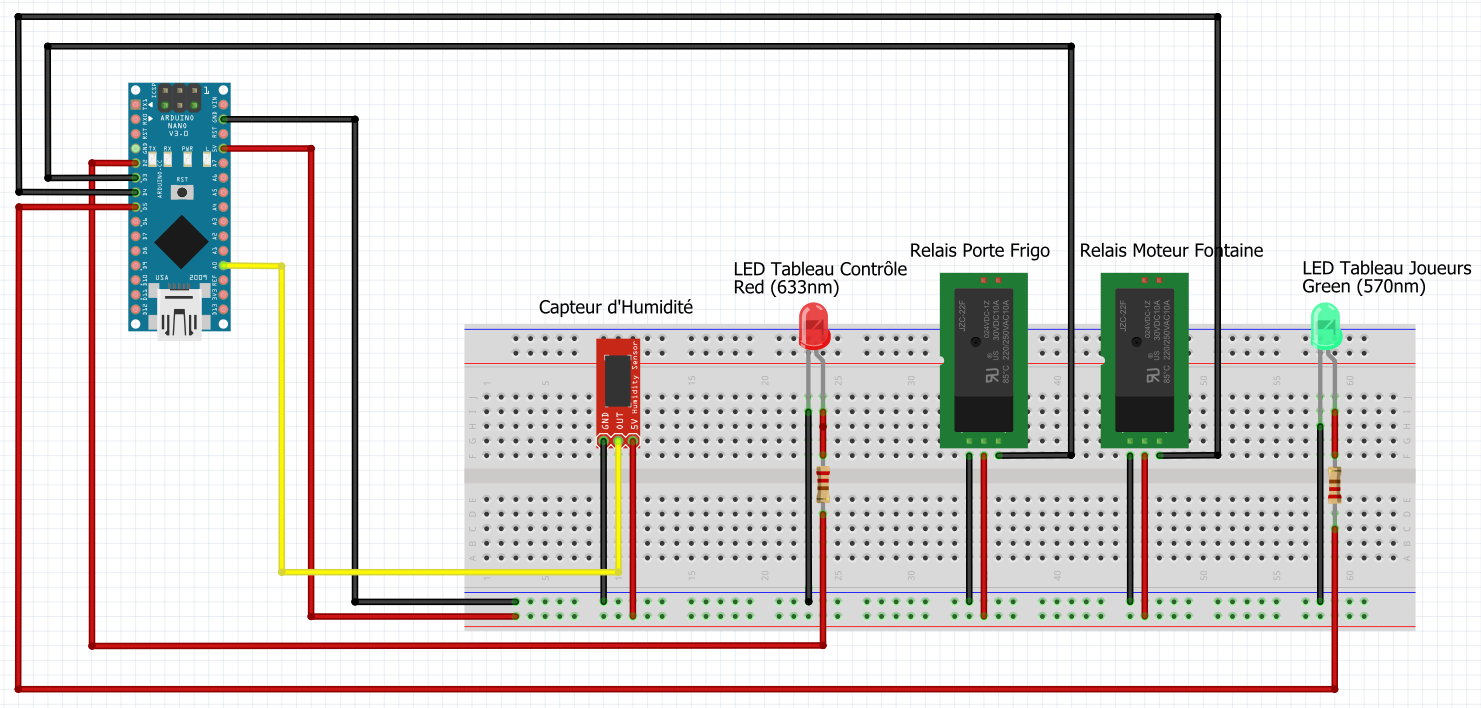


Image 2 : Schéma câblage du mécanisme 5.

Une image contenant table, jouet, homme

Description générée automatiquement

Image 3 : Schéma fonctionnement mécanisme 5.

## Programme

### Initialisation

On définit des variables pour les pins utilisées de l’Arduino :

Une image contenant écran, portable, moniteur, tenant

Description générée automatiquement

Image 4 : Code en tête de Eau.ino

On initialise le matériel utilisé pour le mécanisme :

## Mécanisme n°5 : l’élément EAU

### Matériels utilisés

* Une image contenant équipement électronique, circuit

  Description générée automatiquementCapteur d’humidité (Water Sensor)
* Un Arduino Nano
* Gâche électrique (Solénoïde 12 V)
* 2 Relais (5 V à 220 V)
* Moteur d’une fontaine (220 V)
* 2 LEDS et 2 Résistances (220 Ω)

### Description du Sous-système

Durant l’étape « Elément EAU » de l’escape game, les joueurs doivent verser de l’eau dans une tasse troué. Cette eau s’écoule sur un capteur d’eau (Water Sensor).

Quand le joueur réussis cette énigme, le mécanisme répond ceci :

* Une LED témoin s’allume au tableau de contrôle.
* Une fontaine (220 volts) se met en marche via un relais (5 volts).
* Une gâche électrique (Solénoïde 12 volts) se met en marche afin d’ouvrir la porte d’un frigo suivant la séquence suivante :



L’activation de la gâche électrique via cette séquence permets aux joueurs d’entendre la gâche électrique se mettre en marche et ainsi réaliser que le frigo est désormais ouvert.

* L’élément EAU (LED) est allumé sur la tablette à destination des joueurs.

### Capteur d’humidité (Water Sensor)

Le capteur d’eau est un capteur assez sensible qui renvoi une valeur numérique à l’Arduino. Quand le système est mis en route, ce capteur renvoi une première valeur à l’Arduino.   
Au niveau du code, cette valeur est stockée dans une variable. Enfin les Sortie S1, S2, S\_EAU s’activent lorsque le capteur renvoie une valeur supérieure de 180 points de base.

Nous procédons de la sorte car la valeur renvoyée par le capteur dépend du taux d’humidité résiduel encore présent sur le capteur. Celui-ci varie constamment. Ce capteur est le seul élément à l’entrée de l’Arduino (C\_Humidite).

* Référence : BDT 250
* Signal : Numérique
* Voltage : 0-5V

### Les sorties de l’Arduino

* S\_Frigo
  + Rôle : Activer/désactiver une gâche électrique via un relais.
  + Condition : Si C\_Humidite est à 1 alors activer la sortie, sinon, désactiver la sortie.
  + Voltage : 0-5V
* S\_Fontaine
  + Rôle : Activer/désactiver une le moteur d’une fontaine (220 Volt) via un relais.
  + Condition : Si C\_Humidite est à 1 alors activer la sortie, sinon, désactiver la sortie.
  + Voltage : 0-5V
* S\_Led
  + Rôle : Activer/désactiver une led sur le panneau de contrôle.
  + Condition : Si C\_Humidite est à 1 alors allumer led, sinon, éteindre led.
  + Voltage : 0-3V
* S\_Eau
  + Rôle : Entrée de la gestion des quatre éléments.
  + Condition : Si mecanism\_status = true alors allumer led verte sur la tablette, sinon, allumer led rouge.
  + Voltage : 0-3V

## Programmation du mécanisme

Pour programmés nos différents mécanismes, la société 13 ème Porte nous avais fournis leurs codes de base, pour améliorer un maximum leurs codes. Leurs programmes étaients très brouillon et désorganisée.

Nous nous sommes tout de suite renseigné sur le logiciel Arduino et son langage de programmation. Ainsi, nous avons apprient que le langage Arduino était un langage qui se rapprochait beaucoup du C et du C++ : nous en avons conclut qu’il fallais réorganisé touts ces programmes en classes et fonctions.

Tout d’abord on renseigne sur quelle PIN se trouvent les différents relais et capteurs :

Une image contenant écran, téléphone, pièce, tenant

Description générée automatiquement

Ensuite, nous déclarons la classe Eau.

### Déclaration de la classe Eau :

### 

Comme vous pouvez le voir ci-dessus, les attributs privée de la classe Eau sont tous des Boolean, afin de simplifier l’envoie des états des différents capteurs et actionneurs (true : **Activer** ; false : **Désactiver**). En seconde partie, les fonctions de la classe Eau en public :

* **Eau()** : Constructeur par défaut.
* **void setupMecanism()** : La fonction setup, qui informe au logiciel Arduino quelles capteurs ou actionneurs est en sortie ou en entrée de l’arduino. Pour ce mécanisme, seul le capteur d’humidité est en entrée de l’arduino.
* **void execute()** : La fonction où le mécanisme s’exécute.

### Fonction void execute() :

Tout d’abord, il faut activer le capteur et lire sa première valeur (ValeurCapteur est initialisé à 0) :



On stocke ensuite cette valeur dans une variable « ValeurCapteurInitiale ». Ensuite, on utilise une deuxième valeur du capteur pour pouvoir faire la différence entre la première valeur : ce qui nous donneras la valeur exacte du capteur que l’on stockera dans **C\_Humidite**.

Pour finir, si **C\_Humidite** est supérieur ou égale à 180, alors le mécanisme se lanceUne image contenant capture d’écran

Description générée automatiquement.

## Une image contenant équipement électronique, circuit Description générée automatiquementMécanisme n°9 : les 4 éléments

### Matériels utilisés :

* Une image contenant microscope

  Description générée automatiquementBouton poussoir (5 V)
* Le 4 Arduino Nano correspondant

aux 4 éléments (Terre, Feu, Eau, Air).

* Electroaimant (24V)
* 5 LEDS et 5 Résistances (220 Ω)
* 1 Relais (5 V à 220 V)

### Description du Sous-système :

Les joueurs appuient sur le bouton poussoir situé sous « La tablette des 4 éléments ». Si les 4 éléments n’ont pas été validés alors une LED rouge à proximité immédiate s’allume quelques secondes. Si les 4 éléments ont préalablement été validé alors l’électroaimant de la porte de sortie est désactivé, ouvrant ainsi la porte Final.

### Les entrées de l’arduino :

* C\_Poussoir
  + Type de capteur : Bouton poussoir
  + Référence : A-000000-01319
  + Signal : Numérique
  + Voltage : 0-5V
* S\_Terre
  + Signal : Numérique
  + Voltage :  0-5V
* S\_Feu
  + Signal : Numérique
  + Voltage :  0-5V
* S\_Eau
  + Signal : Numérique
  + Voltage :  0-5V
* S\_Air
  + Signal : Numérique
  + Voltage :  0-5V

### Les sorties de l’Arduino :

* S\_PorteFinal
  + Rôle : Activer/désactiver une électro aimant via un relais (sortie finale).
  + Condition : Si les 4 entrées sont à 1 lorsque le joueur appuis sur le bouton poussoir alors activer la sortie, sinon, désactiver la sortie.
  + Voltage : 0-5V
* S\_Led
  + Rôle : Activer/désactiver une LED rouge via une résistance (sortie finale).
  + Condition : Si les 4 entrées sont à 1 lorsque le joueur appuis sur le bouton poussoir alors activer LED Rouge.
  + Voltage : 0-5V

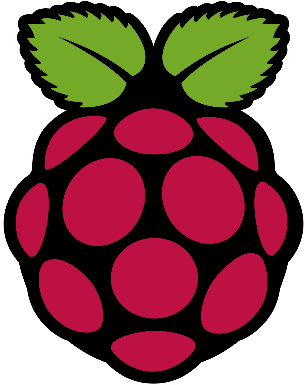
# Développement de l’application sur Raspberry

Pour cette partie, le PC de supervision doit pouvoir gérer et visualiser à distance l’état de chacun des mécanismes via une application WEB. De plus il doit pouvoir récupérer les informations transmis par les mécanismes depuis la Raspberry.

Une image contenant horloge, dessin

Description générée automatiquement

La Raspberry relié par liaison I2C avec tous les mécanismes, (les 9 Arduino Nano) reçois toutes les informations des mécanismes qui ensuite les envois par sockets au PC de supervision. L’installation et la configuration de la Raspberry seras faites en commun avec l’étudiant 1.



Une image contenant dessin

Description générée automatiquement

* **Langage de développement :** Programmation par sockets en Python.
* **Logiciel utilisé :** Putty (Émulateur de terminal).

## Raspberry (Définition)

Le Raspberry pi est un nano ordinateur de la taille d'une carte de crédit que l'on peut brancher à un écran et utilisé comme un ordinateur standard. Sa petite taille, et son prix intéressant fait du Raspberry pi un produit idéal pour tester différentes choses, et notamment la création d'un serveur Web chez soi.

Une image contenant équipement électronique, circuit

Description générée automatiquement

## Programmation par socket

Typiquement, une socket respecte un flux spécifique d'événements pour qu'elle fonctionne. Pour un modèle client-serveur orienté connexion, la socket du processus serveur attend la demande d'un client. Pour ce faire, le serveur doit d'abord établir une adresse que les clients peuvent utiliser pour trouver et se connecter au serveur. Lorsqu'une connexion est établie avec succès, le serveur attend que les clients demandent un service. L'échange de données client-serveur aura lieu si le client se connecte au serveur via la socket. Le serveur répondra alors à la demande du client et lui enverra une réponse.

## Putty (Définition)

PuTTY est un [émulateur de terminal](https://fr.wikipedia.org/wiki/%C3%89mulateur_de_terminal) doublé d'un [client](https://fr.wikipedia.org/wiki/Client_(informatique)) pour les protocoles [SSH](https://fr.wikipedia.org/wiki/Secure_Shell), [Telnet](https://fr.wikipedia.org/wiki/Telnet), [rlogin](https://fr.wikipedia.org/wiki/Rlogin" \o "Rlogin), et [TCP brut](https://fr.wikipedia.org/wiki/Transmission_Control_Protocol). Il permet également des connexions directes par liaison série [RS-232](https://fr.wikipedia.org/wiki/RS-232).

Une image contenant capture d’écran

Description générée automatiquementUne image contenant dessin

Description générée automatiquement

# Développement de l’application WEB de supervision

Une application WEB doit être crée pour le poste de supervision de l’administrateur. Une interface pour pouvoir piloter les différents actionneurs doit être réalisé. Le superviseur pourra démarrer ou arrêter chaque actionneur. L’ordre sera tout d’abord transmis par liaison WIFI à la Raspberry, qui transmettra par liaison I2C cet ordre au mécanisme visés (Arduino nano correspondant au mécanisme). Le pilotage à distance des actionneurs devra inhiber la décision décrite dans la section Gérer les neuf mécanismes.

Une image contenant signe, vert, peint, assis

Description générée automatiquementUne image contenant assiette, dessin

Description générée automatiquement

* **Langage de développement :** Programmation PHP/HTML/CSS
* **Logiciel utilisé :** NetBeans + plugin PHP

## Programmation PHP/HTML/CSS

### HTML

L'HTML est un langage informatique utilisé sur l'internet. Ce langage est utilisé pour créer des pages web. L'acronyme signifie « HyperText Markup Language », ce qui signifie en français "Langage de balisage d'hypertexte".

Ce n'est pas à proprement parlé un langage de programmation, mais plutôt un langage qui permet de mettre en forme du contenu. Les balises permettent de mettre en forme le texte et de placer des éléments interactif, tel des liens, des images ou bien encore des animations. Ces éléments ne sont pas dans le code source d'une page codé en HTML mais "à côté" et la page en HTML ne fait que reprendre ces éléments. Pour visualiser une page en HTML il est nécessaire d'utiliser un [navigateur web](http://glossaire.infowebmaster.fr/navigateur-web/).

### CSS

CSS est l'acronyme de « Cascading Style Sheets » ce qui signifie « feuille de style en cascade ». Le CSS correspond à un langage informatique permettant de mettre en forme des pages web (HTML ou XML).

### PHP

PHP est connu comme langage de script utilisé côté serveur. Il est utilisé dans le développement web ainsi que comme langage de programmation général.

Les fichiers PHP peuvent contenir du code utilisé pour exécuter différents processus en ligne. Le moteur PHP sur le serveur web analyse le code PHP contenu dans le fichier et génère dynamiquement le code HTML. C'est ce code HTML et non pas le code PHP sous-jacent qui est visible par l'internaute qui visite une page web.

## Maquette de l’interface

Une image contenant capture d’écran

Description générée automatiquement