# Une image contenant capture d’écran Description générée automatiquementEtudiant n°4

Étudiant 4

# Développement sur Arduino

Une image contenant capture d’écran

Description générée automatiquementLes Arduino ont un rôle très important dans notre projet, car en effet ce sont les Arduino qui piloteront et gèrerons les mécanismes. Chaque mécanisme est dépendant d’un Arduino, pour la programmation de ces Arduino nous utiliserons le logiciel « Arduino Genuino » :

Une image contenant objet, signe, horloge

Description générée automatiquement

## 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

## Analyse

### Une image contenant capture d’écran Description générée automatiquementDiagramme de classe

### 

### Diagramme de séquence (Partie 1)

Une image contenant texte, carte

Description générée automatiquement

### Diagramme de séquence (Partie 2)

Une image contenant texte, carte

Description générée automatiquement

## Message I2C (MsgI2C)

Toutes les arduinos comme vous avez vus précédemment, devront envoyés l’état de chaque actionneurs et les valeurs de chaque capteurs, par l’aide d’un BUS I2C à la raspberry. La raspberry quant à elle, devras transmette toutes ces données au poste de supervision. Le poste de supervision stockera toutes ces données dans une base de données, pour ainsi les affichées via une IHM.

C’est donc pour cela que sur la fin du diagramme de séquence de l’élément EAU, il y a une **utilisation d’interaction** (joint une séquence de messages définis dans un autre diagramme)qui se nôme **MsgI2C.** Touts les mécanismes sont concernés par ce diagramme MsgI2C, cette utilisation d’interaction seras donc présentes sur toutes les fin des diagrammes de séquences de chaque mécanismes.

### MsgI2C

### 

## 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 :

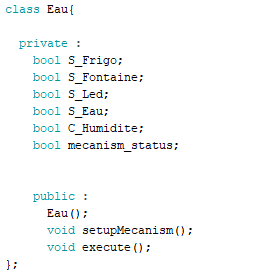
Une image contenant écran, téléphone, pièce, tenant

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 les 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 valeur :



En fonction de la valeur du capteur, on vérifie si la condition suivante à bien été remplies : si le capteur renvoie une valeur supérieure de 180 points de base, alors exécuter ceci :

Une image contenant capture d’écran

Description générée automatiquement

* La led du panneau de contrôle s’allume (**S\_Led = true**).
* Le moteur de la fontaine s’allume (**S\_Fontaine = true**).
* **SFrigo\_PIN** qui est le relais de la porte du frigo, ne fais que de s’activer et de se désactiver pour aider le joueur à savoir qu’une porte c’est ouverte (le solénoïde claque).
* **S\_EAU = true** signifie que le mécanisme s’est déroulé avec succès.

## Mécanisme n°9 : les 4 éléments

Une image contenant équipement électronique, circuit

Description générée automatiquement

### 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 de sortie.

### 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

## Analyse

### Digramme de classe

### Une image contenant capture d’écran Description générée automatiquement

### Diagramme de séquence (Partie 1)

Une image contenant texte, carte

Description générée automatiquement

### Diagramme de séquence (Partie 2)

### Une image contenant texte, carte Description générée automatiquement

## Programmation du mécanisme

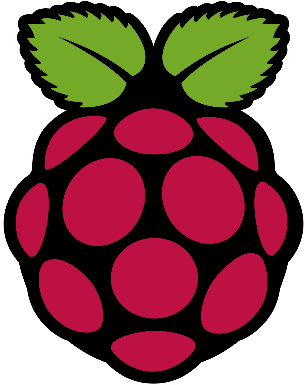
# Développement de l’application sur Raspberry

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), 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 assiette, dessin

Description générée automatiquementUne image contenant signe, vert, peint, assis

Description générée automatiquement

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