# Une image contenant capture d’écran Description générée automatiquementSFL5 : Système Escape Game 13e porte

Etudiant n°4

## Développement sur Arduino :

Les 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 » :

### Mécanisme 5 (l’élément EAU) :

Matériels utilisés :

* Capteur d’humidité (Water Sensor)
* Une image contenant équipement électronique, circuit

  Description générée automatiquementUn Arduino Nano
* Une image contenant équipement électronique

  Description générée automatiquementUn interrupteur à bascule
* Electroaimant
* Relais
* Moteur d’une fontaine

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 (E1).

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

Les sorties de l’Arduino :

* S1
  + Rôle : Activer/désactiver une gâche électrique via un relais.
  + Condition : Si E1 est à 1 alors activer la sortie, sinon, désactiver la sortie.
  + Voltage : 0-5V
* S2
  + Rôle : Activer/désactiver une le moteur d’une fontaine (220 Volt) via un relais.
  + Condition : Si E1 est à 1 alors activer la sortie, sinon, désactiver la sortie.
  + Voltage : 0-5V
* S\_EAU
  + Rôle : Entrée de la gestion des quatre éléments.
* S\_Purge
  + Rôle : Activer/désactiver une gâche électrique via un relais pour purger l’eau.
  + Condition : Si E2 est à 1 alors activer la sortie, sinon, désactiver la sortie.
  + Voltage : 0-5V

### Mécanisme 9 (Les quatre éléments) :

Une image contenant microscope

Description générée automatiquementMatériels utilisés :

* Bouton poussoir
* Le 4 Arduino Nano correspondant aux 4 éléments (Terre, Feu, Eau, Air)
* Electroaimant
* LED
* Relais

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 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 :

* Bouton poussoir
  + Type de capteur : Bouton poussoir
  + Référence : A-000000-01319
  + Signal : Numérique
  + Voltage : 0-5V
* E\_TERRE
  + Signal : Numérique
  + Voltage :  0-5V
* E\_FEU
  + Signal : Numérique
  + Voltage :  0-5V
* E\_EAU
  + Signal : Numérique
  + Voltage :  0-5V
* E\_AIR
  + Signal : Numérique
  + Voltage :  0-5V

Les sorties de l’Arduino :

* S1
  + 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 + 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.

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.

* **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.

## 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.

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

**Voici une maquette de l’interface :**