# 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

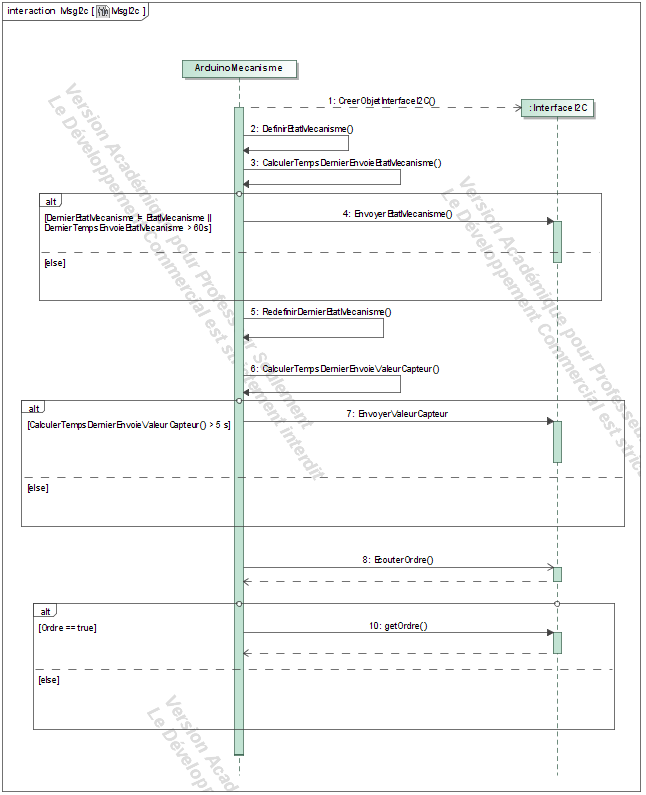
Description générée automatiquement

## 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 :



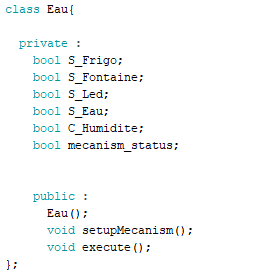
## Programmation du mécanisme :

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.

### 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.