



**INTRODUCTION À LA GESTION ET AU DÉVELOPPEMENT DE PRODUITS
EN GÉNIE ET EN INFORMATIQUE**

GNG 2501

Livrable E – Contraintes de conception et prototype 2

Équipe FB3.4

Numéros étudiants

300084197

300242712

300200871

300259720

Noms complets

Juliette Leprohon

Jean-Paul Merrick-Binnendyk

Assia Rguib

Dounia Tsamo

Professeur : Patrick Dumond

TA : Karim Rizki

PM : Reda Yassin

Faculté de génie

Hiver 2023

Table de matières

Table de matières.....	ii
Liste des figures.....	iii
Listes des tableaux.....	iii
Introduction.....	1
E.1 Contraintes de conception.....	2
1.1 Contraintes de conception non fonctionnelles identifiées.....	2
1.2 Changements à apporter pour satisfaire chacune des contraintes.....	2
1.3 Preuves de l’habileté de nos changements à satisfaire les contraintes (calculs, simulations simple, analyse, recherche, etc.).....	3
1.4 Mise à jour du concept détaillé pour satisfaire les contraintes.....	4
E.2 Prototype 2.....	6
1.1 Résumé de la rétroaction de la cliente.....	6
1.2 Hypothèses de produit les plus critiques non confirmée à présent.....	6
1.3 Prototype 2A – Circuit électronique physique.....	8
Description du prototype 2A.....	8
Documentation du prototype 2A.....	8
Essai du prototype 2A.....	11
1.4 Prototype 2B – Jonction entre dispositif de chlore et l’eau.....	13
Description du prototype 2B.....	13
Documentation du prototype 2B.....	13
Essai du prototype 2B.....	13
1.5 Prototype 2C – Système de détection du niveau du chlore.....	14
Description du prototype 2C.....	14
Documentation du prototype 2C.....	14
Essai du prototype 2C.....	19
1.6 Prototype 2D – Boîtier pour les composantes électroniques.....	20
Description du prototype 2D.....	20
Documentation du prototype 2D.....	20
Essai du prototype 2D.....	21
E.3 Plan de projet.....	22
Conclusion.....	22

Liste des figures

Figure 1 : I2C LCD Adapter Pinout	4
Figure 2 : Schéma représentatif du concept global.....	4
Figure 3 : Schéma interne du système.....	5
Figure 4 : Schéma externe du système	5
Figure 5 : Schéma du circuit physique modifié.....	8
Figure 6 : Prototype jonction plus électrovanne.....	13
Figure 7: Interface utilisateur sur IOS	14
Figure 8 : Boîtier Arduino 3D par Onshape	20

Listes des tableaux

Tableau 1 : Essais effectués sur le circuit physique	12
--	----

Introduction

Dans le cas de notre projet, ce livrable a pour principale objectifs e définir des contraintes de conception non fonctionnelles qui nous permettront d'améliorer notre concept et de savoir mieux répondre au besoin de notre cliente. Également, l'application de la méthodologie CPX (Conception pour X) sera adopté lors de la phase de prototypage 2 qui est crucial à notre projet. Ce deuxième prototype nous donnera un cheminement direct vers la création du prototype final ainsi, il est nécessaire de savoir à temps, les ressources mis à disposition et de produire de façon détailler quel processus de fabrication on utilisera pour les différents sous système de notre produit. Tout en étant toujours sur la même longueur d'onde avec notre cliente, les différents prototypes lui seront présenter afin d'obtenir des rétroactions et les incorporer pour le produit final. Une mise à jour de notre plan de projet sera faite de manière à être toujours à date sur les différents jalons à soumettre.

E.1 Contraintes de conception

1.1 Contraintes de conception non fonctionnelles identifiées

- **Contrainte non fonctionnelle #1 : Assemblage/Désassemblage**

La cliente nous a fait part à plusieurs reprises qu'elle aimerait que le dispositif que l'on conçoit puisse être laissé installer lors de la saison hivernale ou qu'il soit facilement détachable. Suite à de la recherche, nous avons déterminé qu'il allait être difficile de concevoir un dispositif résistant à toute température et nous avons convenu qu'il serait mieux de l'enlever avant l'hiver pour éviter tout bri de composantes. Ainsi, nous avons déterminé que la conception pour l'assemblage/désassemblage est une contrainte de conception non fonctionnelle importante dans le cadre de notre projet.

- **Contrainte non fonctionnelle #2 : Facilité d'utilisation**

L'objectif principale de ce projet est de concevoir un dispositif automatisé. Il serait possible de concevoir un dispositif complètement automatisé sans que cela soit facile d'utilisation. On veut donc faciliter la compréhension de notre système de façon que son utilisation soit facile et que cela soit intuitif. On veut donc permettre à la cliente de comprendre ce qui se passe en montrant visuellement l'interaction entre le capteur et la distribution

1.2 Changements à apporter pour satisfaire chacune des contraintes

Pour satisfaire la contrainte de l'assemblage/désassemblage, nous allons limiter le nombre de pièce détachable à deux, soient le réservoir et la tuyauterie qui comprend la valve, et la pièce en T qui rejoint le système de pompe actuel de la piscine. Ainsi, l'ajout de filet aux deux extrémités de la pièce en T permettra une installation facile et fiable de cette pièce au système de tuyauterie existant. La pièce reliant le réservoir de chlore et l'électrovanne sera détachable afin de permettre un désassemblage plus facile et les pièces et composantes entre l'électrovanne et la fonction en T seront bien scellées entre elles pour que tout soit étanche et restent en un morceau.

Pour ce qui est de la facilité d'utilisation, plusieurs modifications seront faites au concept global pour répondre à cette contrainte. Premièrement, l'ajout d'une valve manuelle permettra de bloquer la distribution du chlore si nécessaire que ce soit en cas de problème ou bien lorsque la piscine doit être vidée, etc. Aussi, en cas de problème avec la lecture du niveau du chlore, une notification sera envoyée avec l'application afin d'alerter la cliente qu'il y a une erreur. De plus, afin de faciliter l'utilisation de l'application, une notification sera envoyée lorsque le niveau de chlore est faible, et cela quotidiennement jusqu'à ce que le réservoir soit rempli. Et finalement, un écran permettra d'afficher des informations supplémentaires sur le dispositif, soit le niveau de chlore et le taux de chlore mesuré. Un écran était déjà prévu mais l'augmentation de la taille de celui-ci permettra d'afficher davantage d'informations.

1.3 Preuves de l'habileté de nos changements à satisfaire les contraintes (calculs, simulations simple, analyse, recherche, etc.)

Deux pièces détachables seulement :

Pour ce qui est de l'ajout de filet aux deux extrémités de la pièce en T, selon les recherches effectuées, il a été déterminé que la façon la plus efficace serait de déterminer le filetage intérieur de la tuyauterie actuelle de la cliente et d'installer une pièce ayant ce filetage extérieur aux extrémités de la pièce.

Ajout d'une valve manuelle :

En observant la configuration d'une valve manuelle, il a été déterminé que son implémentation au système actuelle allait être très faisable.

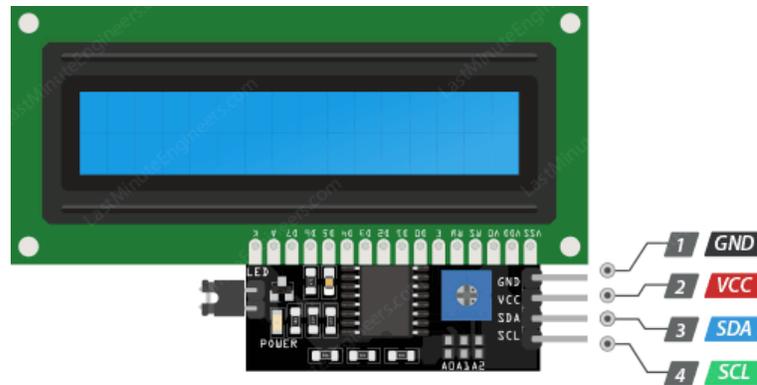
Notification de l'application :

Lors de la configuration de l'application, il sera facile de modifier les paramètres d'alerte pour notifier la cliente quotidiennement. Pour ce qui est du message d'erreur, il sera possible de définir le paramètre d'envoi de ce message comme étant lors que la lecture du capteur ne se situe pas dans les limites attendues.

Écran d'affichage :

Comme mentionné plus haut, l'utilisation d'un écran était déjà prévue mais nous avons pris la décision d'augmenter la taille de cet écran. À la suite de nos recherches, nous avons déterminé qu'il serait très faisable d'utiliser un écran LCD 20x4 au lieu de 16x2 et d'utiliser un I2C LCD Module qui permet de faciliter l'utilisation de l'afficheur. Cet adaptateur permet de réduire le filage nécessaire pour faire

fonctionner l'écran comme le montre la figure ci-dessous. (Note : l'écran démontré est un écran 16x2, il ne s'agit pas de celui qui sera utilisé dans notre cas, mais l'adaptateur fonctionne de la même façon)



I2CLCD Pinout

Last Minute ENGINEERS.com

Figure 1 : I2C LCD Adapter Pinout

(Source : <https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/>)

Il sera donc plus facile de connecter cet écran au circuit de l'Arduino que ce qui était prévu initialement.

1.4 Mise à jour du concept détaillé pour satisfaire les contraintes

Les figures ci-dessous présentent le concept global détaillé qui a été mis à jour pour satisfaire les contraintes mentionnées ci-dessus et qui représente les changements qui ont été fait à présent.

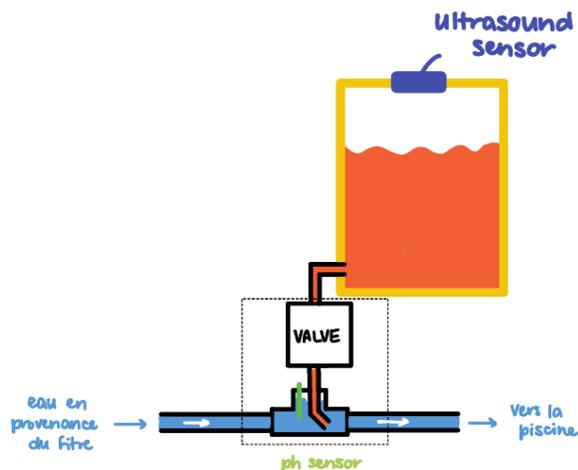


Figure 2 : Schéma représentatif du concept global

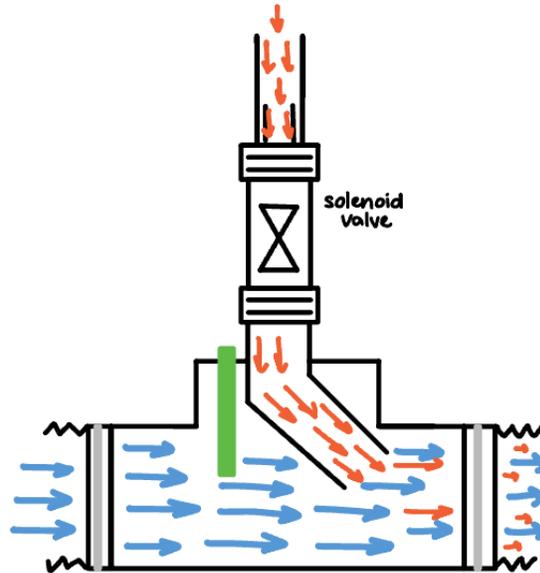


Figure 3 : Schéma interne du système

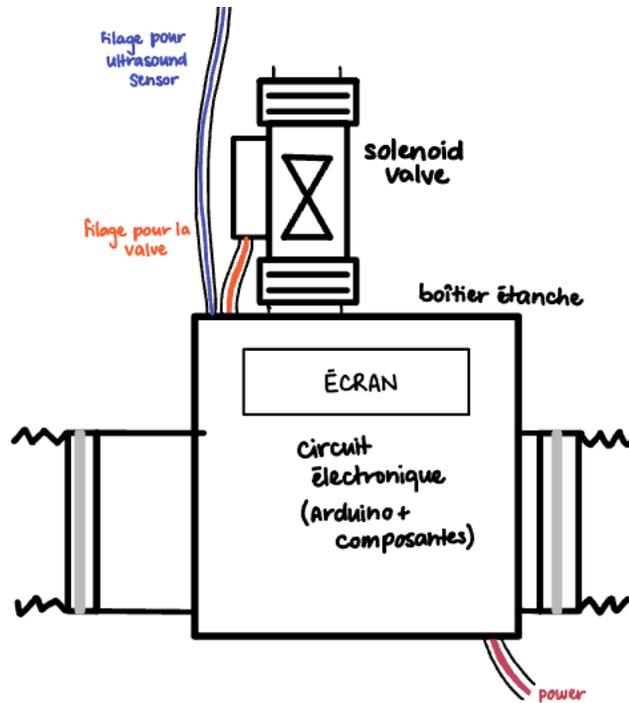


Figure 4 : Schéma externe du système

Pour ce qui est du circuit électronique, un schéma complet sera présenté lors de la description du deuxième prototype.

E.2 Prototype 2

1.1 Résumé de la rétroaction de la cliente

Nous n'avons pas eu de nouvelle rencontre avec la cliente depuis la rencontre #2. Ainsi, les points clés de cette rencontre demeurent les points à améliorer de notre concept. Ces points sont les suivants :

1. Mettre le système après le filtre
2. Changer le capteur de pH pour un capteur de chlore
3. Développer en détails notre concept à partir des projets des années précédentes
4. Trouver une autre méthode pour voir le niveau de chlore restant

Le premier point est modifiable très facilement et a déjà été modifié dans notre concept global. Pour ce qui est du deuxième point, le capteur de chlore, le code de programmation pour l'Arduino a déjà été modifié pour être en mesure de fonctionner avec ce changement de capteur, il ne reste plus qu'à tester le circuit physique et faire des tests avec le capteur de chlore. Pour ce qui est du troisième point, la cliente a mentionné à plusieurs reprises de se baser sur les concepts des équipes précédentes afin de ne pas perdre notre temps en partant de zéro. Ainsi, le développement du circuit électronique et du code a été fait en se basant sur ce qu'une équipe avait accompli précédemment, ce qui nous a permis de passer plus rapidement à la conception de notre circuit physique. Finalement le quatrième point, le système de détection du niveau du chlore, la cliente n'était pas satisfaite avec le système qui lui a été présenté, alors nous avons déterminé que l'un des objectifs de ce deuxième prototype est de trouver un système de détection du niveau du chlore plus efficace et de le présenter à la cliente lors de la prochaine rencontre.

1.2 Hypothèses de produit les plus critiques non confirmée à présent

Plusieurs hypothèses critiques n'ont pas encore été confirmées et présentent un enjeu pour la complétion de notre concept.

Pour ce qui est du circuit électronique, différents paramètres ne sont pas encore conclus ce qui retarde la conception du circuit physique. Le voltage d'entrée cause un problème puisque l'électrovanne nécessite un voltage de 12V alors que le reste des composantes nécessite un voltage de 5V. Afin de résoudre ces hypothèses, les composantes physiques devront être procurées afin de pouvoir tester le circuit physiquement puisque les logiciels de simulation utilisés précédemment ne permettent pas de tester ceci efficacement.

Pour ce qui est du système de tuyauterie qui permet de relier le réservoir et l'électrovanne et d'introduire le chlore dans le flux d'eau de la piscine, la jonction du déversement du chlore est encore à tester. Deux principales hypothèses critiques à vérifier sont si le système élaboré pour introduire le chlore est efficace et si cette pièce est étanche. Une autre hypothèse critique est le fonctionnement de l'électrovanne sous différente pression; on ne sait pas encore à quel point le flux sortant de l'électrovanne varie en fonction de la pression, c'est-à-dire le niveau de chlore dans le réservoir.

La plus grande des hypothèses critiques présentement reste le système de détection du niveau du chlore. Nous voulons ne pas avoir à modifier le contenant de la cliente et de seulement avoir à changer le capuchon entre les différents réservoirs mais il semblerait que ce ne soit pas possible. Nous devons donc déterminer un moyen de détection du niveau du chlore dans le réservoir ainsi qu'un système de notification.

Une dernière hypothèse critique est l'étanchéité du système électronique, nous devons concevoir un boîtier qui permet de garder les composantes électroniques du circuit ensemble et à l'abri des éléments.

Ainsi, lors de ce deuxième prototype, nous tenterons de répondre à ces divers hypothèses critiques afin de faire avancer notre concept.

1.3 Prototypé 2A – Circuit électronique physique

Description du prototype 2A

L'objectif de cette partie du deuxième prototype est de comprendre plus en détails le fonctionnement de certaines composantes, principalement de l'électrovanne et du relai, et d'afficher l'information voulue sur l'écran. Avec la recherche, nous avons déterminé qu'un signal d'entrée de 12V est nécessaire pour que l'électrovanne fonctionne, mais les autres composantes du circuit fonctionnent avec un signal de 5V et le circuit conçu lors du premier prototype ne conviendrait pas pour ce critère. Après de longues heures de recherches, il a été possible de reconcevoir un circuit comprenant toutes les composantes nécessaires et en utilisant un signal d'entrée de 12V. Il s'agit du circuit présenté à la prochaine figure. Il faut par contre tester si les composantes utilisant un signal de 5V fonctionnent comme prévu, et c'est sur ça que porte les tests effectués sur le prototype physique ciblé.

Documentation du prototype 2A

La figure ci-dessous présente un schéma modifié du circuit électronique qui comprend les différentes composantes et qui utilise du 12V comme entrée.

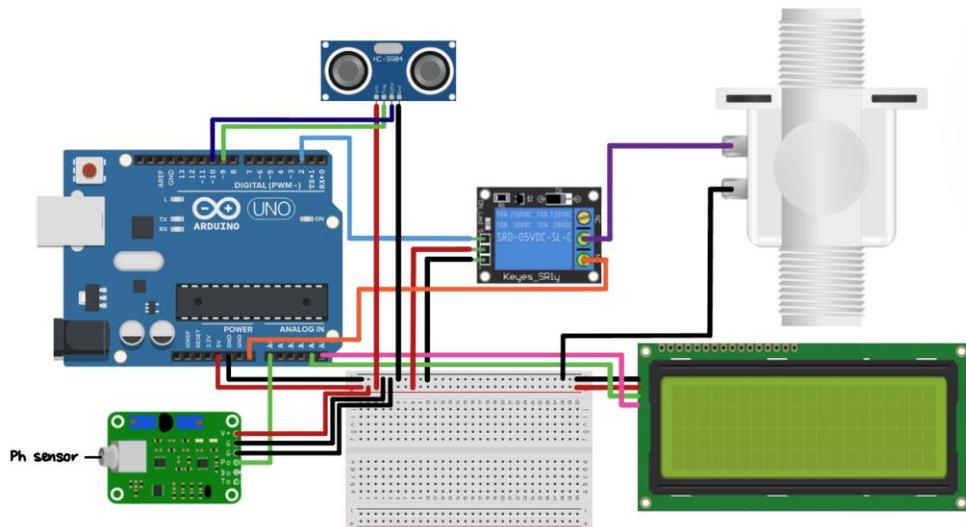


Figure 5 : Schéma du circuit physique modifié

Le code suivant est une version modifiée du code présenté lors du prototype 1 et constitue une version préliminaire du code pour le système et nécessitera plusieurs essais avec le circuit en entier pour adapter le code.

```

//**** GNG2501 - PROJECT - CODE ****//
//**** By Juliette Leprohon ****//

//LIBRARIES
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <LiquidCrystal.h>
#include <Arduino.h>

//DEFINE PINS
#define trigPin = 9; //ultrasound sensor
#define echoPin = 10; //ultrasound sensor
#define pHSensor A0; //ph sensor
#define valve 2; //solenoid valve

//VARIABLES - ULTRASOUND SENSOR
long duration; // Value found by ultrasound sensor
int distance; // Calculated value of distance from duration
int volume; // Calculated value of volume from distance - to display
int capacity; // Capacity of the recipient
int height; // Height of the recipient

//VARIABLES - LCD SCREEN
LiquidCrystal_I2C lcd(0x27, 20, 4); // Set the LCD address to 0x27 for a
20x4 display

//VARIABLES - PH SENSOR
int nbSample = 10; //Number of sample we want to average
int sampleDelay = 1000 //Time inbetween samples in ms (1s=1000ms)
int measureDelay = 10000 //Time inbetween measures in ms
(10s=10000ms)
float adc_resolution = 1024.0; //from pHSensor parameter
double pH; // Calculated pH value from voltage

//VARIABLES - VALVE
int openTime; //Time for the valve to remain open

void setup() {

// ULTRASOUND SENSOR -----
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input

// LCD DISPLAY -----
lcd.begin(20, 4);
lcd.init();
lcd.backlight();

```

```
// SOLENOID VALVE -----
pinMode(valve, OUTPUT);

Serial.begin(9600);      // Starts the serial communication
}

void loop() {

// ULTRASOUND SENSOR -----
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);

// Calculating the distance
distance = duration * 0.034 / 2;

// Calculating the volume of chlorine
volume = capacity*((height-distance)/height);

// PH SENSOR -----
int measurings = 0;

for (int i = 0; i < nbSample; i++)
{
  measurings += analogRead(pHSense);
  delay(sampleDelay);
}

float voltage = 5 / adc_resolution * measurings/nbSample;
ph = 7 + ((2.5 - voltage) / 0.18);

// CALCULATION TO KNOW FOR HOW LONG DOES THE VALVE NEEDS TO BE OPEN

// SOLENOID VALVE -----
digitalWrite(valve, HIGH); //Open the valve for openTime
delay(openTime);
digitalWrite(valve, LOW); //Close the valve

//LCD DISPLAY -----
  lcd.setCursor(0,0);
  lcd.print("R servoir : "+ (String)volume + " L");
  lcd.setCursor(0,1);
```

```
    lcd.print("pH : "+ (String)ph + (String)etat);

    delay(measureDelay);
}
```

Afin de tester l'écran seulement, le relai et l'électrovanne ont été retiré du circuit pour les essais présentés dans la prochaine section et le code utilisé est le suivant.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <LiquidCrystal.h>

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 20, 4);

void setup()
{
  lcd.begin(20, 4);
  lcd.init();
  lcd.backlight();
}

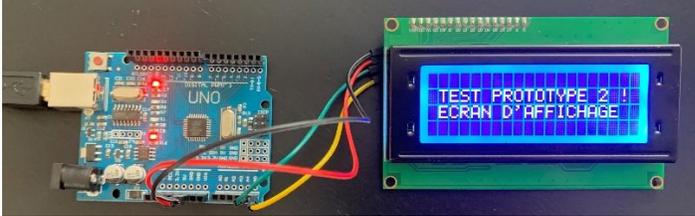
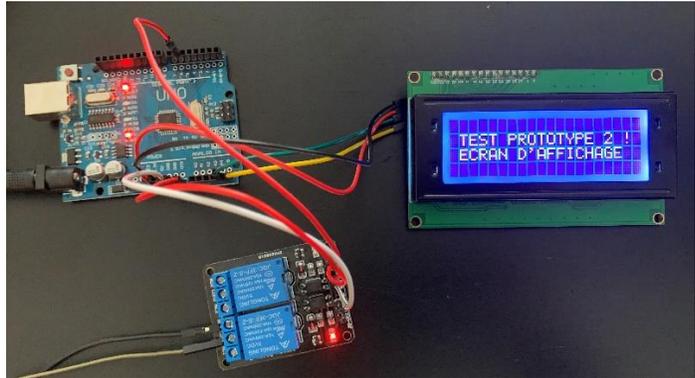
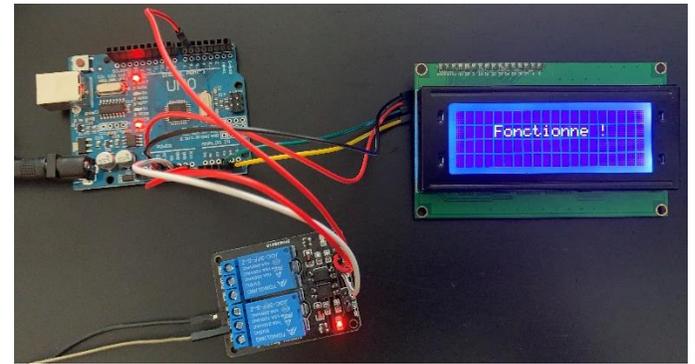
void loop()
{
  lcd.setCursor(1,1);
  lcd.print("TEST PROTOTYPE 2 !");
  lcd.setCursor(1,2);
  lcd.print("ECRAN D'AFFICHAGE");

  delay(5000);
  lcd.clear();
  lcd.setCursor(4,1);
  lcd.print("Fonctionne !");
  delay(5000);
}
```

Essai du prototype 2A

Les essais de cette partie du prototype consistaient principalement à vérifier le fonctionnement de l'écran LCD avec une source de courant de 12V à l'Arduino et de modifier le texte qui affiche sur l'écran. Le tableau suivant présente le résumé des tests effectués sur cette partie du prototype.

Tableau 1 : Essais effectués sur le circuit physique

No	Description et Image	Résultats
1	<p>Fonctionnement de l'écran LCD avec 5V input</p> 	<ul style="list-style-type: none"> • Facilité d'utilisation : 9/10 – Le message est bien lisible, compréhensif, mais on doit être en face de l'écran pour le voir (nécessite un déplacement)
2	<p>Fonctionnement de l'écran LCD avec 12V input</p> 	<ul style="list-style-type: none"> • Fonctionne comme le test 1 – c'est ce qu'on veut • Facilité d'utilisation : 9/10 – (même que test 1)
3	<p>Changement de l'affichage</p> 	<ul style="list-style-type: none"> • Il est facile de changer la disposition du texte affiché et les informations, il sera possible de le modifier pour satisfaire la cliente • Facilité d'utilisation : 9/10 – (même que test 1&2)

Il est possible de voir avec le tableau ci-dessus que l'affichage ne cause pas de problème et qu'il est possible d'utiliser un signal de 12V qui passe par la plaquette Arduino pour faire fonctionner l'écran LCD avec un signal de 5V.

1.4 Prototype 2B – Jonction entre dispositif de chlore et l'eau

Description du prototype 2B

Le prototype 2B comprend d'un système de plomberie qui puisse connecter le chlore, la pompe et la piscine.

Documentation du prototype 2B

La figure ci-dessous présente des photos du sous-système.



Figure 6 : Prototype jonction plus électrovanne

Essai du prototype 2B

Après avoir été remplis d'eau pour 30 minutes, il n'y en manquait pas alors on peut assumer une bonne étanchéité. Cependant, après avoir été inspecté un peu plus tard, on a remarqué que l'adhésif s'est décollé. Pour éviter la faillite du système dans le futur il faut se procurer un adhésif de meilleure qualité.

1.5 Prototype 2C – Système de détection du niveau du chlore

Description du prototype 2C

Dans cette partie du sous-système du prototype, on s'est concentré sur la réalisation d'une application qui notifie la cliente quand le niveau de chlore devient insuffisant (moins un tiers du conteneur). Premièrement, j'ai développé l'application iOS, vu que la cliente utilise un iPhone, avec Swift sur Xcode, puis j'ai développé le code pour mesure de niveau de chlore dans Arduino. Vu que notre système à pour critères principale c'est qu'il soit automatique, c'est pour ça qu'on a décidé de développer une application qui va permettre à la cliente de voir le niveau d'eau, et le notifiera si elle a besoin de remplir le conteneur de chlore.

Documentation du prototype 2C

La figure ci-dessous présente des Screenshots du sous-système.

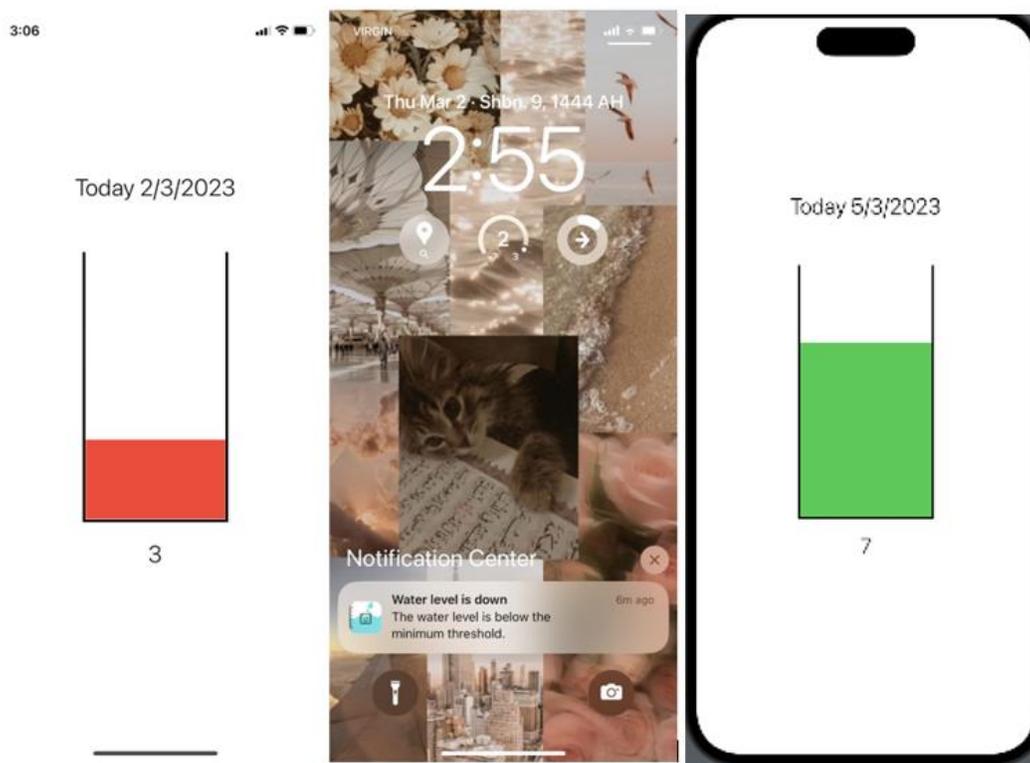


Figure 7: Interface utilisateur sur IOS

Parmi les codes rédigés:

```
// ContentView.swift
```

```
// Created by Assia Rguib on 2023-02-19.
```

```
import SwiftUI
```

```
import UserNotifications
```

```
struct ContentView: View {
```

```
    var body: some View {
```

```
        VStack{
```

```
            Text(CurrentDate())
```

```
                .font(.title)
```

```
                .fontWeight(.light)
```

```
                .padding(.bottom, 50.0)
```

```
            Rectangle()
```

```
                .trim(from:0.1735,to:1)
```

```
            .stroke(Color.black,lineWidth: 3)
```

```
            .frame(width: 170.0, height: 320.0)
```

```
            ZStack {
```

```
                Rectangle()
```

```
                    .fill(chloreLevelColor())
```

```
                    .frame(width: 166.0, height: (CGFloat(filling()*31.5))
```

```
                ).padding(.top,-((CGFloat(filling()*31.5)+10))
```

```
                Text(String(filling()))
```

```
                    .font(.title)
```

```
                    .fontWeight(.light)
```

```
                    .padding(.bottom, 50.0)
```

```
            }
```

```
        }
```

```
    }
```

```
extension Date {
```

```
func dayOfWeek() -> String? {
    let dateFormatter = DateFormatter()
    dateFormatter.dateFormat = "EEEE"
    return dateFormatter.string(from: self).capitalized
}
}

func CurrentDate()->String{
    let date = Date()
    let calendar = Calendar.current
    let day = calendar.component(.day, from: date)
    let month = calendar.component(.month, from: date)
    let year = calendar.component(.year, from: date)
    return("Today \(day)/\(month)/\(year)")
}

func chloreLevelColor() -> Color {
    var color = Color.green
    let center = UNUserNotificationCenter.current()
    center.requestAuthorization(options: [.alert, .sound]) { granted, error in
        if let error = error {
            print(error.localizedDescription)
            return
        }
        if granted {
            print("sssss")
            if (filling() <= 3){
                print(color)
                let content = UNMutableNotificationContent()
                content.title = "Water level is down"
                content.body = "The water level is below the minimum threshold."
                content.sound = UNNotificationSound.default

                let trigger = UNTimeIntervalNotificationTrigger(timeInterval: 5.0,
repeats: false)
```

```
        let request = UNNotificationRequest(identifier: "notification",
content: content, trigger: trigger)
        center.add(request) { error in
            if let error = error {
                print("Error scheduling notification:
\\(error.localizedDescription)")
                return
            }
        }
    }}
}
if (filling() <= 3){
    color = Color.red}
return color
}

func filling()->Int{
    let lvl = 7
    print(lvl)
    return 7
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

On est encore en train d'essayer de faire fonctionner Arduino avec Fire base.

```
// Example NewPing library sketch that does a ping about 20 times per second.
```

```
#include <NewPing.h>
```



```
// #define _GLIBCXX_USE_CXX11_ABI 0
// #include <FirebaseArduino.h>

#define TRIGGER_PIN 13 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN 2 // Arduino pin tied to echo pin on the ultrasonic sensor.
#define MAX_DISTANCE 10 // Maximum distance we want to ping for (in centimeters).
Maximum sensor distance is rated at 400-500cm.

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of pins and maximum
distance.

// #define FIREBASE_HOST "https://ch-17-52da7-default-rtdb.firebaseio.com/"
// #define FIREBASE_AUTH "0mGQQIriuHgZ6nWU2qSdpAg55TnDuqakzzlnEGF"
// #define WIFI_SSID "eduroam"
// #define WIFI_PASSWORD "Jan_rio@allah17"

void setup() {
  Serial.begin(9600); // Open serial monitor at 115200 baud to see ping results.
  // WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  // Serial.print("connecting");
  // while (WiFi.status() != WL_CONNECTED) {
  // Serial.print(".");
  // delay(500);
  // }
  // Serial.println();
  // Serial.print("connected: ");
  // Serial.println(WiFi.localIP());
  // Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {
  // Firebase.setFloat("number", sonar.ping_cm());
  // if (Firebase.failed()) {
```

```
// Serial.print("setting /number failed:");  
// Serial.println(Firebase.error());  
// return;  
DynamicJsonDocument doc(1024);  
doc["data"]=sonar.ping_cm();  
String output;  
serializeJson(doc, output);  
Serial.println(output);  
}  
}}
```

Essai du prototype 2C

Les essais de cette partie du prototype consistaient principalement à vérifier le fonctionnement de mesurer niveau de chlore et envoyer des notifications par l'application, on a essayé le code Arduino avec un capteur ultrasonique et on a pu mesurer l'évolution de distance. Et pour l'application quand on l'ouvre pour la première fois ça demande la permission d'envoyer des notifications puis après quand le niveau de chlore est bas ça envoie une notification, si ça demeure insuffisante, ça envoie une autre notification pour rappeler la cliente (nous avons codé les valeurs en direct pour les tester).

1.6 Prototype 2D – Boîtier pour les composantes électroniques

Dans cette partie du sous-système du prototype, l'on s'est concentré sur la réalisation d'une boîte qui compte contenir toutes les composantes électroniques du projet. De ce fait, la création du boîtier s'est fait grâce à l'utilisation d'Onshape, un logiciel de conception assisté ordinateur. Vu que notre système à pour critères principale l'étanchéité et la résistance à la température, il en convient donc d'avoir un boîtier fait d'un matériau imperméable à l'eau et au chlore et qui résiste de très fortes ou très basses températures.

Description du prototype 2D

La conception d'un boîtier sur Onshape a pour but de visualiser le type de matériau que nous allons utiliser afin de rendre notre système étanche qui était un critère assez sensible pour notre cliente. Grâce au logiciel Onshape, on a pu tester les dimensions que nous recherchions pour le boîtier, les matériaux et des essais d'assemblage de la partie du fond et du couvercle du boîtier.

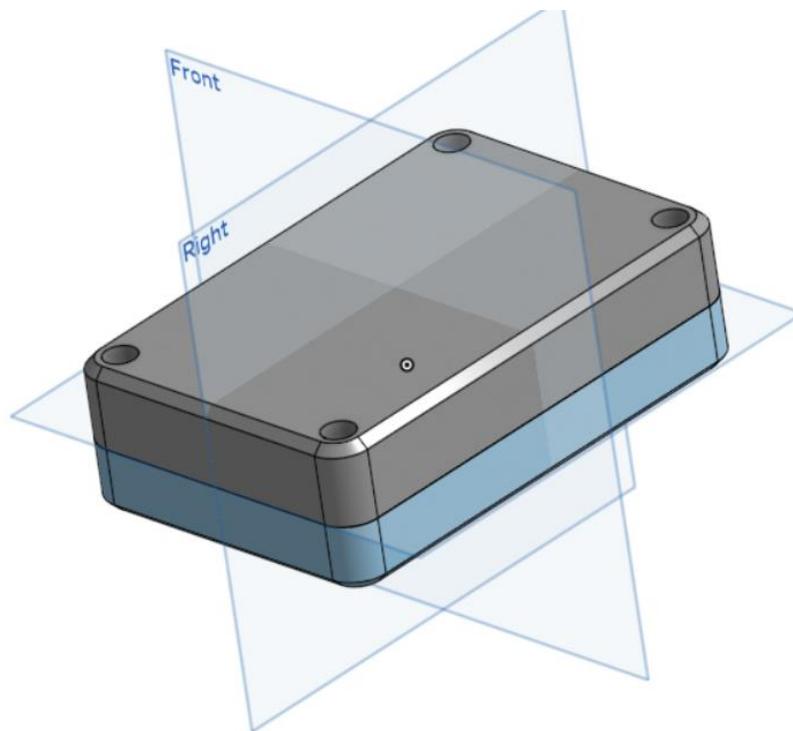


Figure 8 : Boîtier Arduino 3D par Onshape

Documentation du prototype 2D

<https://cad.onshape.com/documents/e8f0d3e4bccefad110abc794/w/4500a010b49edd3dfb322f48/e/bbb943bacd775645d093708f?renderMode=0&uiState=64002b83cdfbd04d2a2e04ff>

Essai du prototype 2D

Pour un premier essai du prototype du sous-système, il en ressort que notre boîtier est faisable avec les dimensions requis c'est-à-dire 150mm x100mm x40mm, ce qui contiendra des parties de notre système surtout les composantes électroniques et les protégera lors des intempéries ou lorsqu'ils subiront des températures critiques. Le choix du matériau se tournant vers le PVC car c'est un matériau assez étanche, et non corrosif au chlore. Il est beaucoup utilisé pour des piscines. Pour finir, les dimensions choisies ne sont pas celles exactes mais le boîtier à été conçu de façon à être facile à modifier aux niveaux des dimensions.

E.3 Plan de projet

Lien du plan du projet mis-à-jour :

<https://www.wrike.com/frontend/ganttchart/index.html?snapshotId=WkK6x0CEhqjifuokj8tcbDUxWI5AoZlr%7CIE2DSNZVHA2DELSTGIYA>

Conclusion

En guise de conclusion, on pourrait dire que nous sommes en constante évolution en ce qui concerne le projet. La réalisation de ces divers prototypes nous ont permis de mieux comprendre de notre système, d'en apprendre sur ce qu'on voulait et pouvait vraiment faire afin de satisfaire les besoins de notre cliente. Pour un deuxième prototype, qui signifie un grand pas vers la réalisation du produit finale, ces prototypes physiques permettrons un assemblage assez direct et efficace et une simple implémentation au niveau du système de piscine déjà installé chez notre cliente. Dans la même lancé d'un apprentissage continue, il nous suffira de terminer la communication existante entre notre microcontrôleur, capteur de détection de niveau de chlore et l'interface utilisateur Xcode pour IOS, la réalisation d'un autre boitier pour la protection de notre électrovanne, la finalisation du circuit physique intégrée ainsi que leur assemblage final pour en faire un prototype unique. Une mise à jour du plan détaillé sera aussi fournie pour mieux se préparer et avoir de l'avance sur le projet.