

# **BE de C++**

## **Système de lavage des mains sans contact**

**AUBERT Jérémy  
XU Qiang**

# I. Introduction - Contexte

La crise sanitaire actuelle du Covid-19 a mis en exergue l'importance du lavage régulier des mains. Cependant, cette action requiert une certaine interaction manuelle avec son environnement et celle-ci peut aider à propager le virus. Afin de réduire les risques de propagation, il faut réduire les zones de contact dans les actions au quotidien. Dans ce sens, nous nous sommes penché sur la question du lavage des mains "sans contact". Un tel système, simple d'installation, facile d'utilisation et peu coûteux, permettrait sa propagation rapide à beaucoup de ménages. Dans ce rapport, nous allons nous attarder sur l'architecture de notre projet, sur les problèmes et difficultés que nous avons pu rencontrer, et des perspectives futures de ce projet.

## II. Diagramme de classe

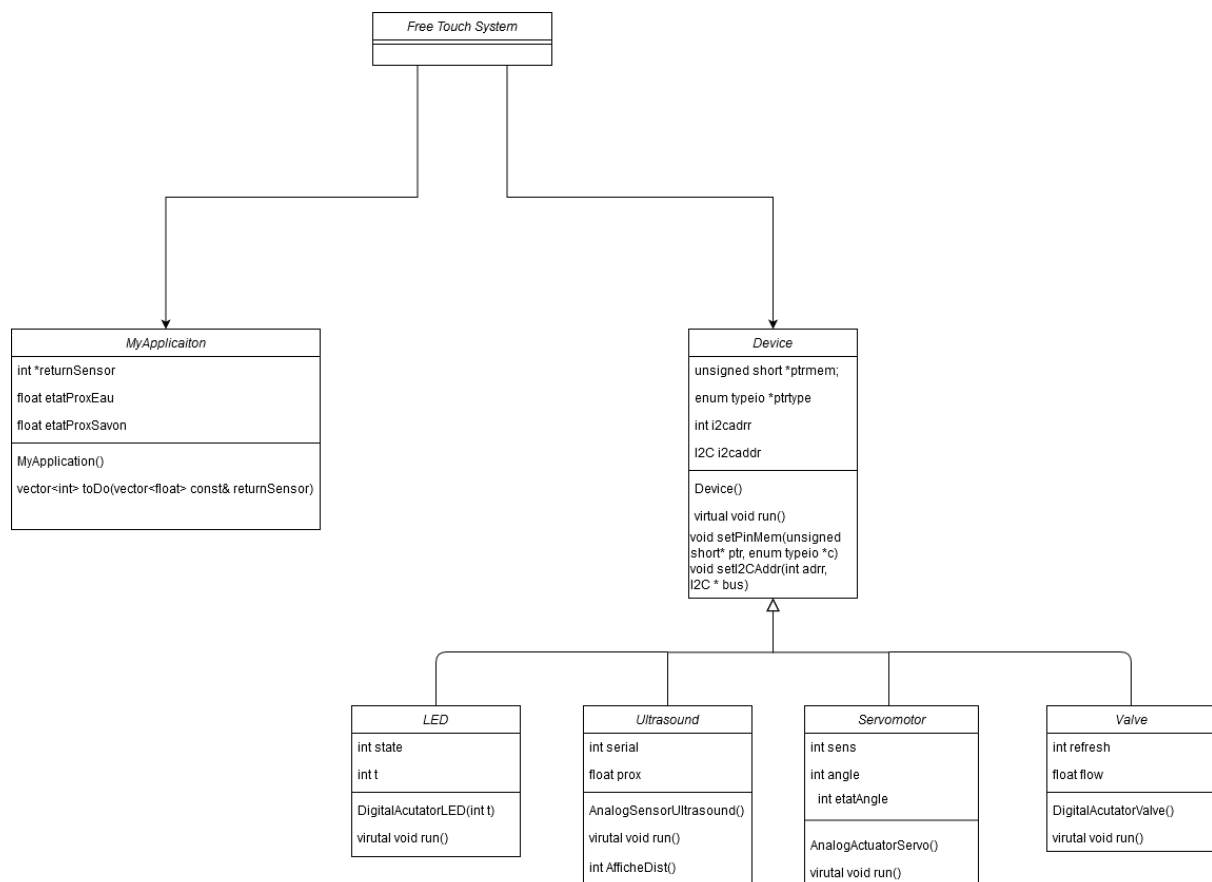


Figure 1. Diagramme de classe de notre projet

Nous pouvons observer que ce diagramme se décomposer en deux parties :

### 1. Software

La partie software ne contient qu'une classe et nous permet la simulation dans l'interaction entre les capteurs et les actionneurs. Cette classe va nous permettre, via l'envoi d'un tableau dynamique, de parcourir une liste d'action qui seront ou non exécutée,

selon les valeurs présentes dans le tableau. Le détail de cette classe sera à retrouver sur les fichiers `core_simulation.h` et `core_simulation.cpp`

## 2. Hardware

Nous avons ici 4 classes (1 capteur, 3 actionneurs) qui héritent de la classe hardware `Device()`. On retrouve, sur le système final, 2 capteurs (2 ultrasons) ainsi que 3 actionneurs (1 LED, 1 servomoteur et 1 valve). Ces classes sont orientées vers notre système, elles ne sont donc pas à utilisation universelles. Le détail de ces classes sera à retrouver sur les fichiers `mydevices.h` et `mydevices.cpp`

# III. Schéma de fonctionnement

## 1. Diagramme de cas d'utilisation

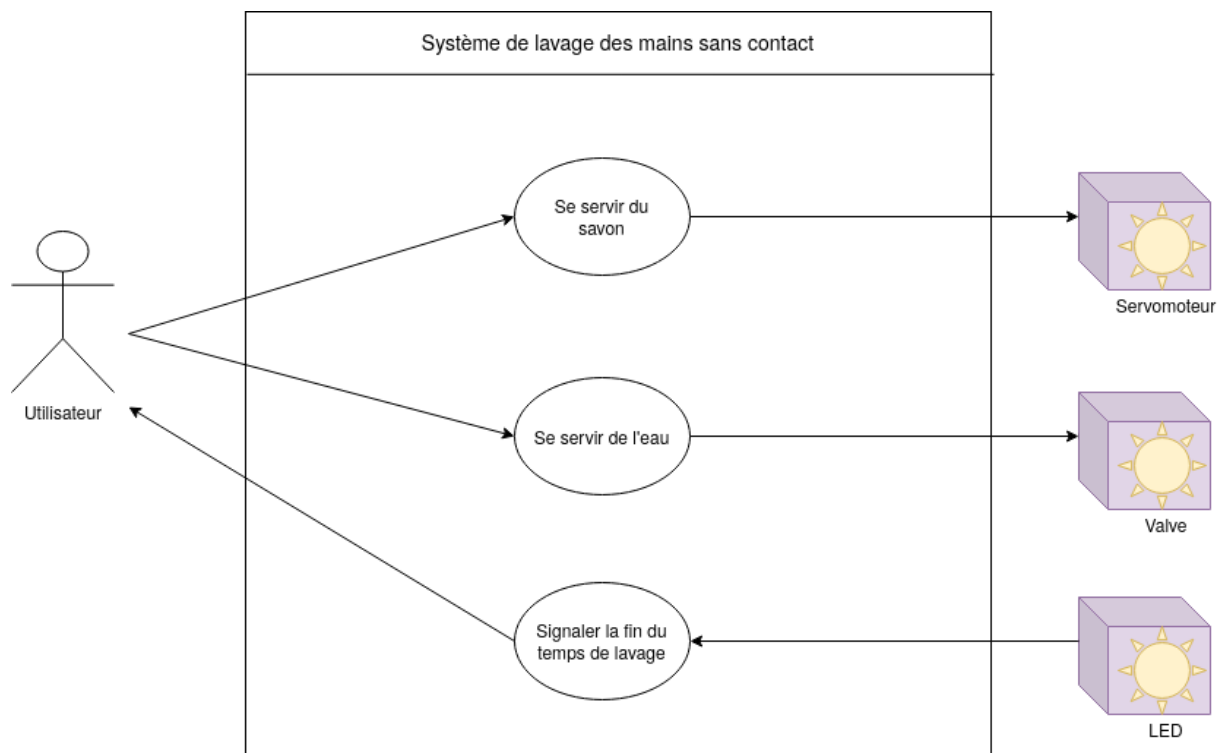


Figure 2. Diagramme de cas d'utilisation du système

Comme on peut le voir dans ce diagramme de cas d'utilisation, le système reste assez simple dans son utilisation. En effet, on retient trois actions principales :

- Se servir du savon : il s'agit de la distribution du savon lorsque l'utilisateur approche sa main du capteur de proximité à ultrason lié au savon ;
- Se servir de l'eau : Il s'agit de l'ouverture de la vanne d'eau lorsque l'utilisateur approche sa main du capteur de proximité à ultrason lié au robinet ;
- Signaler la fin du temps de lavage : Au bout de 20 secondes après l'enclenchement de la distribution de savon, une LED indique la fin du temps réglementaire pour cette

action, en restant allumée, jusqu'au déclenchement d'une nouvelle action (*i.e.* nouvelle distribution). À noter que pendant ces 20 secondes, la LED est dans un état clignotant afin d'indiquer à l'utilisateur l'état en cours.

## 2. Algorithme

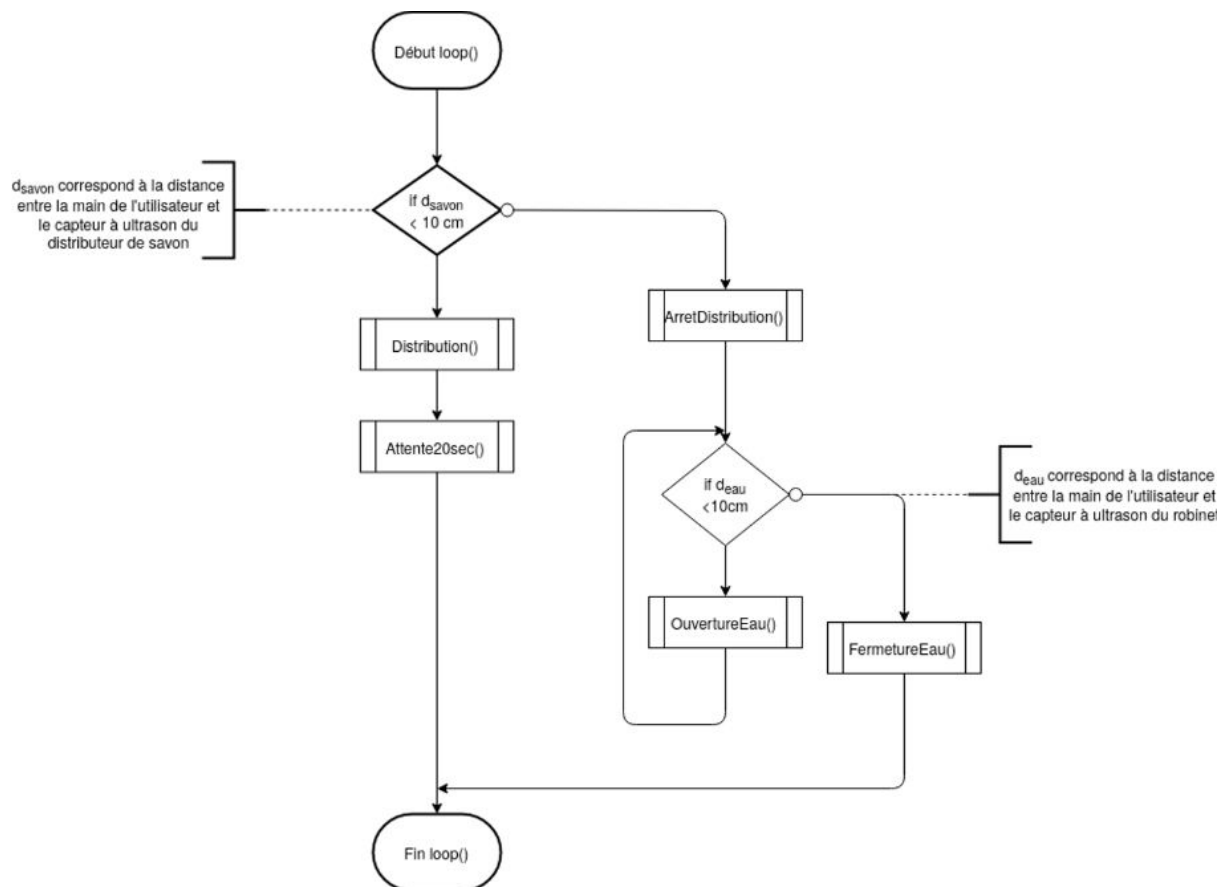


Figure 3. Diagramme d'algorithme du processus général

Nous pouvons considérer l'ensemble du système de lavage des mains comme un cycle. Cette boucle est la fonction Loop() qui a été exécutée.

- $d_{\text{savon}}$  est la distance entre les mains et le savon. Il est obtenu par un capteur à ultrasons.
- $d_{\text{eau}}$  est la distance entre les mains et l'eau. Il est obtenu par un capteur à ultrasons.
- Lorsque  $d_{\text{savon}}$  est inférieur à 10 cm, notre système détecte qu'il y a une main à nettoyer, nous avons donc commencé à lui fournir du savon. Ce processus est exécuté par la fonction Distribution().
- Afin de respecter le temps nécessaire pour laver efficacement les mains, soit 20 secondes, nous mettons en place une interface pour l'utilisateur afin de lui indiquer la fin de la tâche ou non. Donc, à chaque fois que nous fournissons le savon, nous attendons 20 secondes grâce à la fonction Attente20sec().

- Après avoir fourni du savon, vous devez fournir de l'eau. Nous l'avons atteint grâce à l'équation `ArretDistribution()`. À noter qu'il est possible d'avoir une distribution de savon et d'eau en même temps.
- Lorsque  $d_{\text{eau}}$  est inférieur à 10 cm, notre système estime que cette main est assez proche pour fournir de l'eau. Via la fonction `OuvertureEau()`. Lorsque nous avons fini de nous laver les mains,  $d_{\text{eau}}$  est supérieur à 10 cm. Nous avons cessé de fournir de l'eau. Via la fonction `FermetureEau()`.

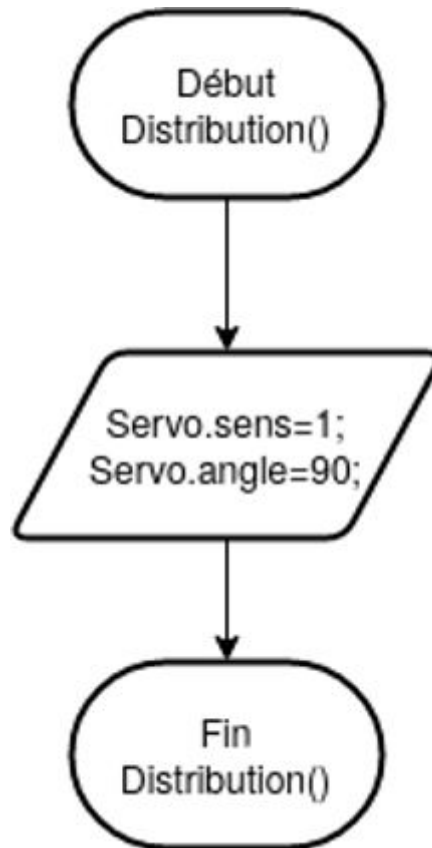


Figure 4. Diagramme d'algorithme du `Distribution()`

`Servo.sens` égal à 1 signifie que notre système commence à fournir du savon, et `Servo.angle` égal à 90 degrés signifie que notre porte-savon est incliné à 90 degrés pour fournir du savon.

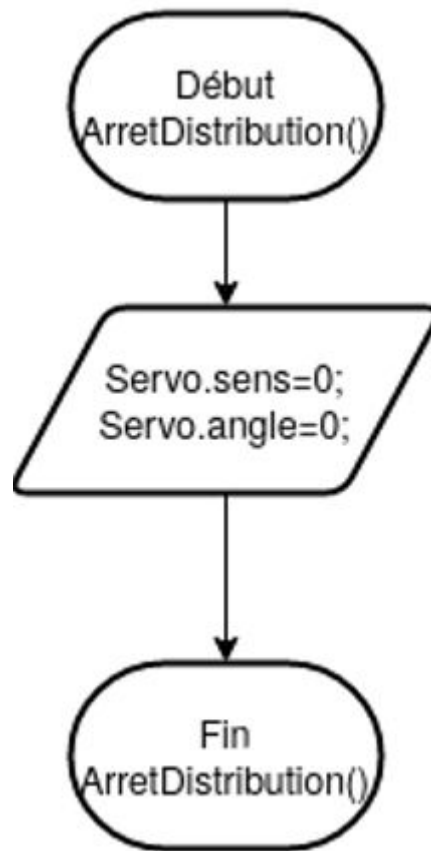


Figure 5. Diagramme d'algorithme du `ArretDistribution()`

Le principe est le même que `Distribution()`.

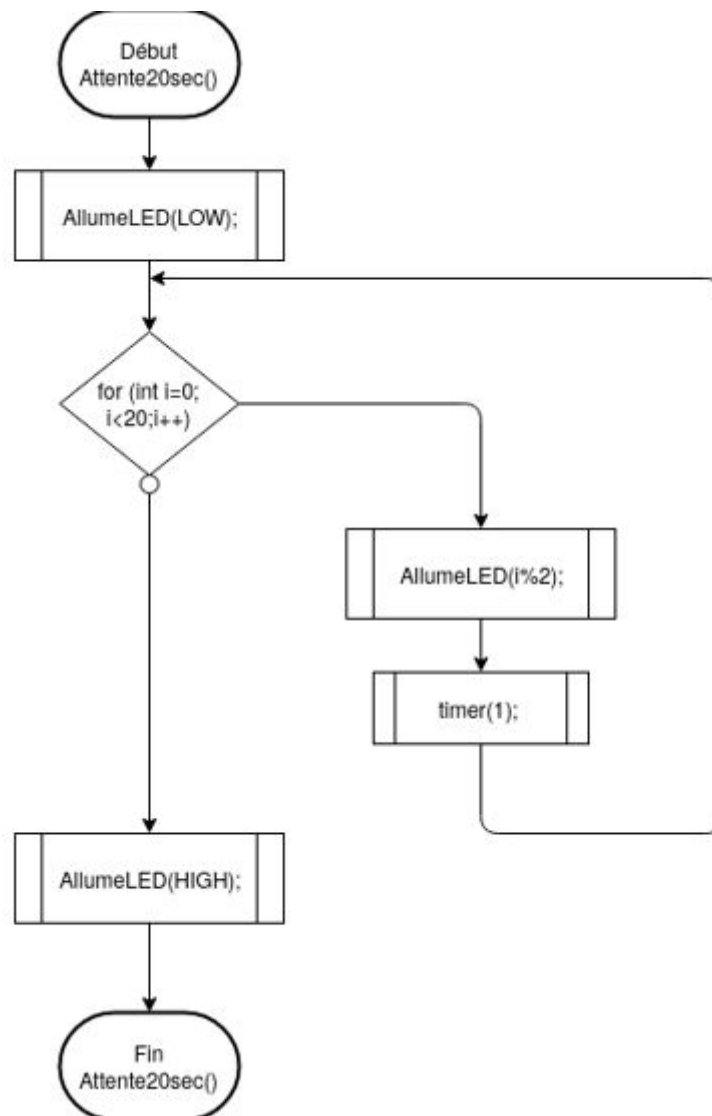


Figure 6. Diagramme d'algorithme du Attente20sec()

Le système maintient la lumière LED à l'état LOW. Dans les 20 secondes d'attente, lorsque le nombre de secondes est pair, nous allumons la LED. Ceci réalise une LED clignotante. La fréquence de clignotement est de 2 s à chaque fois, 0.5 Hz.

Après 20 secondes, nous arrêtons de clignoter et gardons la LED allumée. Cela signifie que notre système est bien préparé pour recevoir le prochain utilisateur.

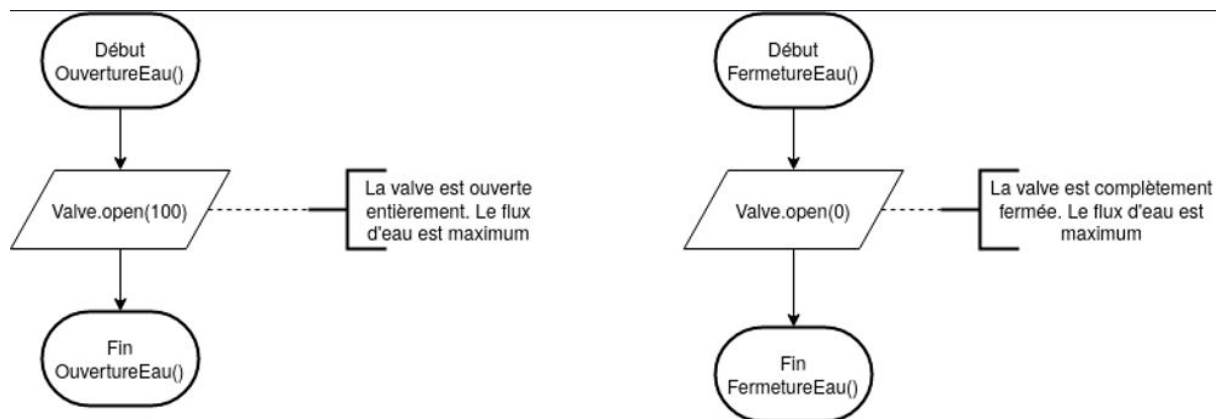


Figure 7. Diagramme d'algorithme du OuvertureEau() et FermetureEau()

Nous modifions le débit d'eau grâce à la fonction Valve.open(x), x = 100 signifie le débit maximal et x = 0 signifie qu'aucune eau ne s'écoule.

## IV. Problèmes rencontrés et perspectives d'évolution

### 1. Problèmes rencontrés

Au lancement du projet, un membre du binôme a rencontré des soucis concernant la compilation sur machine virtuelle. Ce soucis ne nous a pas totalement arrêtés mais nous a freinés dans le démarrage du projet. De plus, nous n'avions pas d'expérience dans l'utilisation de git. Malgré une séance de présentation au premier semestre, cela ne nous a pas suffi à être suffisamment à l'aise dans son utilisation. De plus, l'article sur l'utilisation de git fourni par le corps enseignant nous a plus laissé dans l'inconnu que de nous donner des pistes claires. Nous nous sommes basé pour la suite sur une vidéo de présentation de git d'une durée de 30min. Cette vidéo, avec un contenu assez exhaustif, était suffisante pour utiliser git pour ce projet. Le retard pris à cause des problèmes listés ci-dessus sont estimés à l'équivalent d'environ 1 à 2 séances.

Le temps de développement d'une classe software "MyApplication" - pour passer de notre version initiale à une version implémentant cette classe - s'est faite au détriment de développement de nouvelles fonctionnalités.

### 2. Perspectives d'évolution

- Remplacement des capteurs à ultrason par des capteurs de présence IR

Cette solution permettrait de réduire l'encombrement du système, et ainsi, augmenter sa facilité d'installation dans les ménages. Cette solution est notamment utilisée dans les toilettes des aéroports/gares/aires de repos et autres. Néanmoins, cette solution nous fait



perdre la notion de distance, ce qui pourrait dégrader la qualité de notre système. L'importance de cette information sera à tester sur un système réel.

- Remplacement de la LED par un écran

Afin de signaler à l'utilisateur que le temps de lavage de main de 20 secondes est dépassé, nous utilisons une LED qui, après une séquence de clignotement, reste allumée. Une modification possible serait de remplacer cette LED par un écran. Cela permettrait à l'utilisateur de voir le temps restant ainsi qu'une indication écrite que le temps est dépassé. Néanmoins, l'écran pourrait être, selon sa qualité et son installation, difficile à lire pour l'utilisateur. De plus, nous perdons cette communication "directe" que l'on avait avec la LED. L'utilisation d'une LED permet en effet de voir directement que le temps est dépassé : c'est lorsque la LED reste allumée.

- Indiquer le volume d'eau consommé

La présence d'une valve est idéale puisqu'elle nous permet de suivre la consommation d'eau. Avec ce capteur, nous avons la possibilité de suivre la consommation quotidienne d'eau. L'administrateur du système pourrait alors recevoir cette information soit sur son téléphone - ce qui nécessiterait l'ajout d'autres modules, tel qu'un module Bluetooth par exemple - ou bien directement sur l'écran si ce dernier est utilisé.

- Limiter la distribution de savon

Dans la même idée pour la LED, il est possible de limiter la distribution de savon. En effet, dans la version actuelle, il n'y a aucune limitation dans cette action. Ainsi, l'utilisateur peut se servir pendant une vingtaine de secondes s'il le souhaite.

## V. Conclusion

Pour ce projet, notre objectif est d'atteindre une application du C++ au domaine des objets connectés. Compte tenu de la situation actuelle, nous avons conçu un système. Un système de lavage des mains sans contact. Le projet étant réalisé dans le cadre de l'enseignement à distance, nous n'avons pas la possibilité d'utiliser un véritable tableau de développement. Cependant, nous avons la possibilité d'utiliser Git. Et la possibilité de simuler et de compiler des programmes sur les systèmes Windows et Linux. Dans ce projet, nous avons non seulement contacté la connaissance de C++, mais également utilisé la connaissance du diagramme de classe de dessin et du diagramme d'algorithme appris dans COO.