

## SAÉ 304 : Découvrir et mettre en place un réseau IoT

### Jalon 3: Permettre à un joueur d'interagir avec le jeu en utilisant des boutons poussoirs

Ce jalon, lié à l'**UserStory 3**, était centré sur l'intégration de boutons poussoirs connectés à un ESP8266 afin de permettre une interaction avec le jeu via des messages MQTT. Les boutons ont été programmés avec l'Arduino IDE, et configurés pour détecter les pressions et publier des messages spécifiques sur des topics MQTT. Ces messages ont été ensuite interprétés par le jeu pour déclencher des actions telles que l'activation de pouvoirs ou l'ajout de malus."

#### Étape 1 : Mise en place des boutons poussoirs et test de leur fonctionnement sur ESP8266

La première étape a été de connecter les boutons poussoirs aux **pins GPIO** de l'ESP8266 et de tester leur bon fonctionnement. Nous avons connecté chaque bouton poussoir aux broches **GPIO 0, 2, 12 et 14** de l'ESP8266, et l'autre côté des boutons a été relié à la terre (**GND**) pour compléter le circuit. Ainsi, chaque pression sur un bouton peut être détectée par le microcontrôleur.

#### Connexion des boutons poussoirs

- **Pin GPIO 0** → Bouton "Index"
- **Pin GPIO 2** → Bouton "Majeur"
- **Pin GPIO 12** → Bouton "Annulaire"
- **Pin GPIO 14** → Bouton "Auriculaire"

```
// Dclaration des broches des boutons
#define BP_INDEX 0
#define BP_MAJEUR 2
#define BP_ANNUAIRE 12
#define BP_AURICULAIRE 14
```

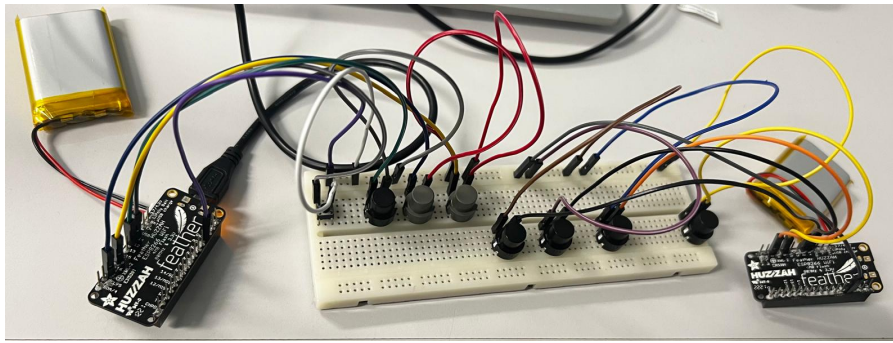


Figure 1: Schéma de connexion des boutons poussoirs avec les microcontrôleurs ESP8266

## Étape 2 : Association des boutons à des topics MQTT spécifiques

Pour permettre une communication claire entre les boutons poussoirs du gant et le réseau MQTT, chaque bouton a été associé à un **topic MQTT unique**. Cette association permet d'envoyer un message spécifique sur le réseau lorsque chaque bouton est pressé.

### Déclaration des topics MQTT

Chaque bouton a été lié à une constante définie avec `#define`, représentant un topic MQTT unique. Voici les définitions des topics associés à chaque bouton poussoir :

```
// Déclaration des topics pour chaque bouton
#define TOPIC_INDEX "gant1/index"           // Topic pour l'index
#define TOPIC_MAJEUR "gant1/majeur"        // Topic pour le majeur
#define TOPIC_ANNUAIRE "gant1/annuaire"    // Topic pour l'annuaire
#define TOPIC_AURICULAIRE "gant1/auriculaire" // Topic pour l'auriculaire
```

### Code d'envoi des messages

Lorsqu'un bouton est pressé, un message est publié sur le topic correspondant. Voici un exemple de code utilisé pour gérer cet envoi :

```
// Exemple de code pour publier un message lors de l'appui sur un bouton
// Poussoir INDEX
if (digitalRead(BP_INDEX) == LOW) {
    if (currentTime - lastPressTimeIndex > debounceDelay) {
        Serial.println("Gant1 BP_Index press ");
        client.publish(TOPIC_INDEX, "ON");
        lastPressTimeIndex = currentTime;
    }
}
```

Ce code détecte l'appui sur le bouton connecté au GPIO 0 (INDEX) grâce à `digitalRead`. Pour éviter les lectures répétées, un délai d'antirebond est appliqué. Lorsqu'il est pressé, un message "ON" est publié sur le topic MQTT correspondant.

## Étape 3 : Test des messages MQTT et des actions dans le jeu

Après avoir associé les boutons à leurs topics MQTT respectifs, nous avons vérifié que les messages étaient correctement publiés en surveillant le broker. Une fois cette étape validée, nous avons testé leur intégration dans le jeu. À chaque pression de bouton, un message MQTT est envoyé, interprété par le jeu pour déclencher l'action correspondante, comme l'activation d'un pouvoir ou l'application d'un malus.



Figure 2: Malus envoyé via les boutons poussoirs

Dans cette étape, nous avons permis à un joueur d'interagir avec le jeu en appuyant sur des boutons poussoirs connectés à un **ESP8266**, envoyant des messages MQTT pour déclencher des actions dans le jeu. Cela a permis de valider la mise en place de toute l'infrastructure nécessaire dans le projet.