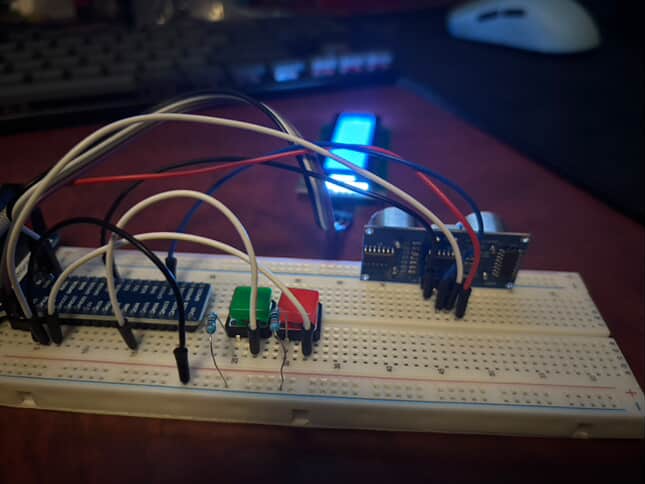
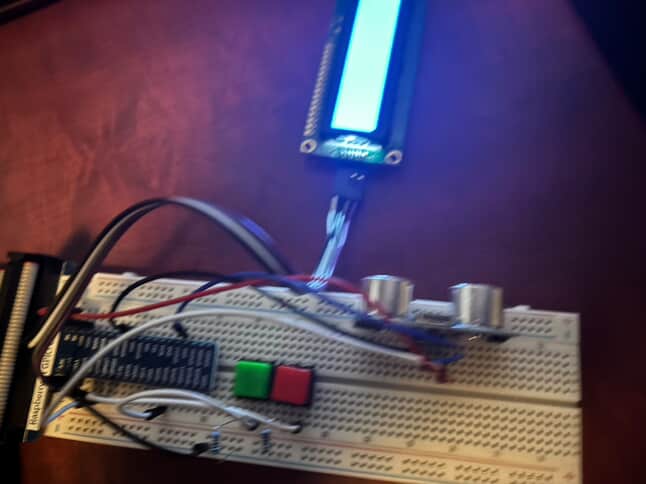
# **Rapport de montage du projet GPIO**

## **(6) Validation 1 : description et exécution du montage GPIO (LEDs/boutons/LCD)**

### **Introduction**

Ce rapport présente la validation du montage GPIO réalisé pour notre projet. Cette première phase de validation se concentre sur l'interaction avec les composants physiques de base : LEDs, boutons-poussoirs et écran LCD. Nous allons décrire le montage réalisé, les tests effectués et les résultats obtenus.

### **Description du montage**

Notre montage GPIO est composé des éléments suivants :

1. **Boutons-poussoirs** :
   1. Un bouton vert (GPIO 5) : démarrage/arrêt du système
   2. Boutons supplémentaires pour la navigation et le contrôle
2. **LEDs** :
   1. LED verte (GPIO 17) : indicateur d'état du système (allumée = système actif)
   2. LED rouge (GPIO 27) : indicateur d'alerte ou d'erreur
   3. LED jaune (GPIO 22) : indicateur de traitement en cours
3. **Écran LCD** :
   1. Écran LCD 16x2 connecté via I2C (GPIO 2 pour SDA et GPIO 3 pour SCL)
   2. Affiche les informations d'état et les messages du système

Le montage a été réalisé sur une plaque d'expérimentation (breadboard) connectée au Raspberry Pi. Chaque LED est associée à une résistance de 220Ω pour limiter le courant, et les boutons-poussoirs sont configurés avec des résistances de pull-up internes.

### **Tests réalisés**

#### **1. Test des boutons-poussoirs**

Nous avons développé des tests unitaires pour valider le fonctionnement des boutons, notamment le bouton vert de démarrage/arrêt :

python

@patch('gpiozero.Button')  
def test\_execution\_bouton\_vert(self, mock\_button):  
 """Test du comportement du bouton vert"""  
 self.assertFalse(self.platine.est\_systeme\_actif())  
   
 resultat = self.platine.basculer\_systeme()  
 self.assertTrue(resultat, "Premier appui: activer le système")  
 self.assertTrue(self.platine.est\_systeme\_actif())  
   
 resultat = self.platine.basculer\_systeme()  
 self.assertFalse(resultat, "Second appui: désactiver le système")  
 self.assertFalse(self.platine.est\_systeme\_actif())

Ces tests confirment que :

* Le bouton vert est correctement initialisé sur le GPIO 5
* Le premier appui active le système (état passe à actif)
* Le second appui désactive le système (état revient à inactif)

#### **2. Test des LEDs**

Pour les LEDs, nous avons vérifié :

* L'allumage et l'extinction via les commandes GPIO
* La correspondance entre l'état du système et l'état des LEDs
* La LED verte s'allume quand le système est actif
* La LED rouge s'allume en cas d'erreur
* La LED jaune clignote pendant les traitements

#### **3. Test de l'écran LCD**

Pour l'écran LCD, nous avons validé :

* L'initialisation correcte de l'interface I2C
* L'affichage de texte sur les deux lignes
* La mise à jour dynamique des informations
* La lisibilité des caractères et le contraste

### **Résultats et observations**

**Fonctionnalités validées :**

* Le bouton vert fonctionne comme prévu, basculant l'état du système à chaque pression
* Les LEDs s'allument et s'éteignent correctement en fonction des états du système
* L'écran LCD affiche clairement les informations et se met à jour en temps réel

### **Démonstration du montage**

Lors de la démonstration, nous avons exécuté une séquence complète :

1. Appui sur le bouton vert → Système activé → LED verte allumée → Message "Système actif" sur LCD
2. Traitement démarré → LED jaune clignotante → Indication de progression sur LCD
3. Simulation d'erreur → LED rouge allumée → Message d'erreur sur LCD
4. Second appui sur bouton vert → Système désactivé → Toutes LEDs éteintes → Message "Système inactif" sur LCD

### **Conclusion**

Cette première phase de validation confirme le bon fonctionnement de base de notre interface GPIO. Les interactions entre les composants physiques (boutons, LEDs, LCD) et le logiciel sont établies et fonctionnent comme prévu. Le système répond correctement aux événements d'entrée et produit les sorties attendues.

Pour la prochaine phase, nous prévoyons d'intégrer des capteurs supplémentaires et d'améliorer l'interface utilisateur en ajoutant des fonctionnalités plus avancées à notre système.