

Frameworks et librairies

- **PlatformIO** : IDE facilitant le développement de logiciel embarqué, compatible avec de nombreux contrôleurs du marché. Facile aussi le débogage et la mise en place de tests unitaires.
- **TaskScheduler**: permet la programmation multi-tâches avec différents niveaux de priorité permettant leur exécution dans un ordre souhaité. Cela répond au besoin de fournir des réponses en temps réel à des événements se produisant dans le monde réel.
- **M5Stack (inclus M5Core)** : donne accès au matériel intégré des contrôleurs M5Stack Core. Les fonctionnalités de la librairie sont accessibles via la classe M5Stack référencé M5. **Exemples :**

- M5.Power // contrôle la batterie
- M5.Lcd // permet de contrôler l'écran LCD
- M5.BtnA, M5.BtnB, M5.BtnC // permet de configurer les boutons
- M5.Speaker // controle de la sortie audio
- M5.Imu, M5.Mpu6886, M5.Sh200Q (accéléromètre)

- **Module GRBL** : permet de contrôler le moteur pas à pas selon 3 axes et une vitesse particulière (à noter que le GRBL 13.2 peut contrôler 3 moteurs simultanément). **Exemples :**

- _GRBL_A.setMotor(int x=0, int y=0, int z=0, int speed=300)
- _GRBL_.sendGcode("G-10")

- **Module GoPlus2** : permet de contrôler le servomoteur. **Exemples :**

- Servo_write_angle(SERVO_NUM0, 0)
- Servo_write_plusewidth(SERVO_NUM0_PW, 2500)

- **M5Unit-UHF-RFID** : librairie permettant d'utiliser le lecteur RFID. Exemples :

```
uhf.begin(&Serial2, 115200, 16, 17, false);
```

```
if (uhf.readCard(reade_buffer, sizeof(reade_buffer), 0x04, 0, 0x00000000)) {  
  log("Read OK");  
  log("Data Content");  
  for (uint8_t i = 0; i < sizeof(reade_buffer); i++) {  
    Serial.printf("%x", reade_buffer[i]);  
  }  
}
```

- **WebConfig** : configuration du controller par interface web
 - Typage
 - Persistance (basé sur SPIFFS)

Architecture

Architecture globale s'appuyant sur les machines à états. Schéma Lucid

Système global et ses sous-systèmes

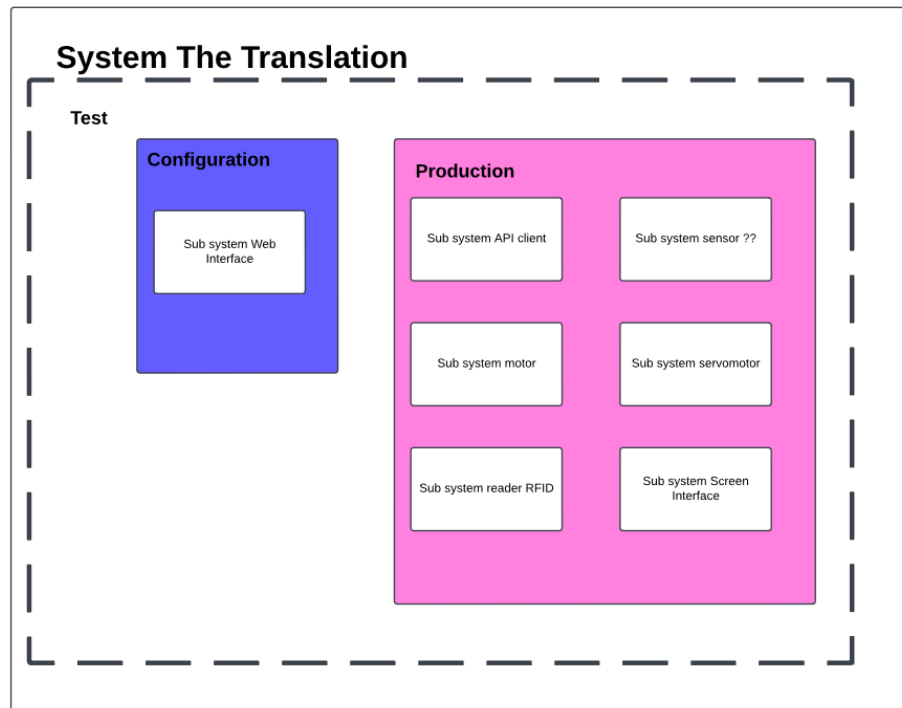


Figure 1: image

Description des états globaux

Description des différents états des sous-systèmes

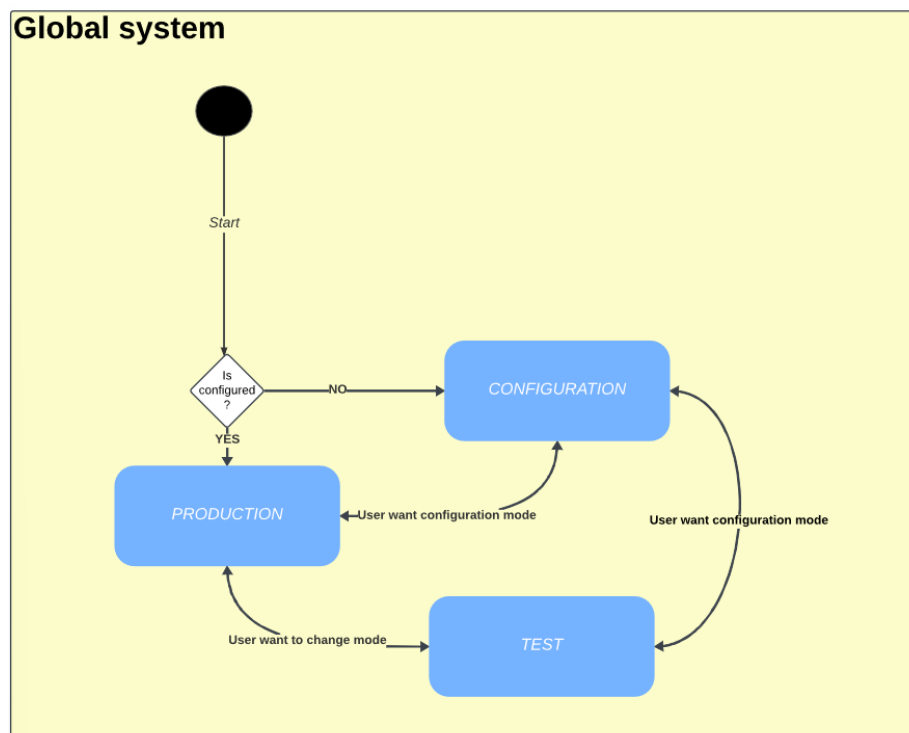


Figure 2: image

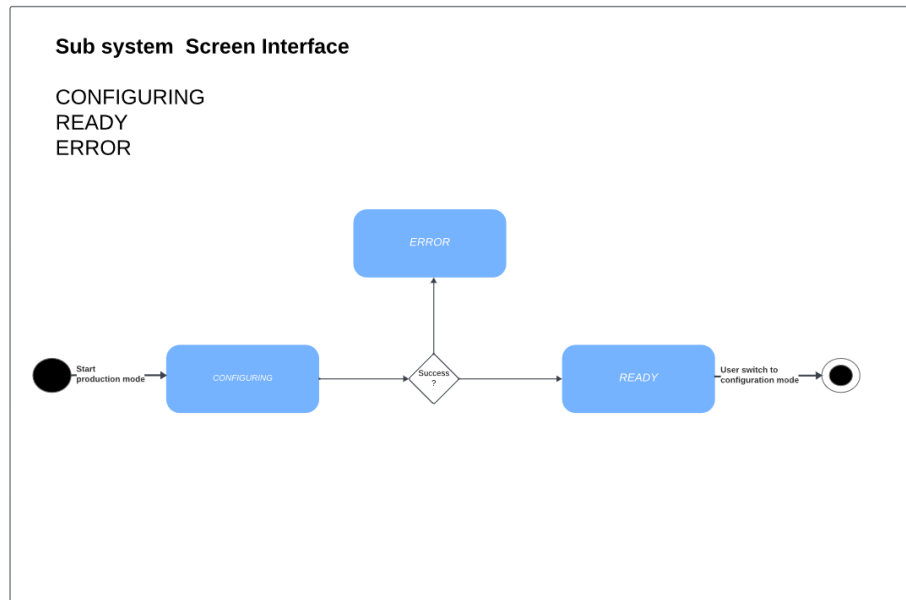


Figure 3: image

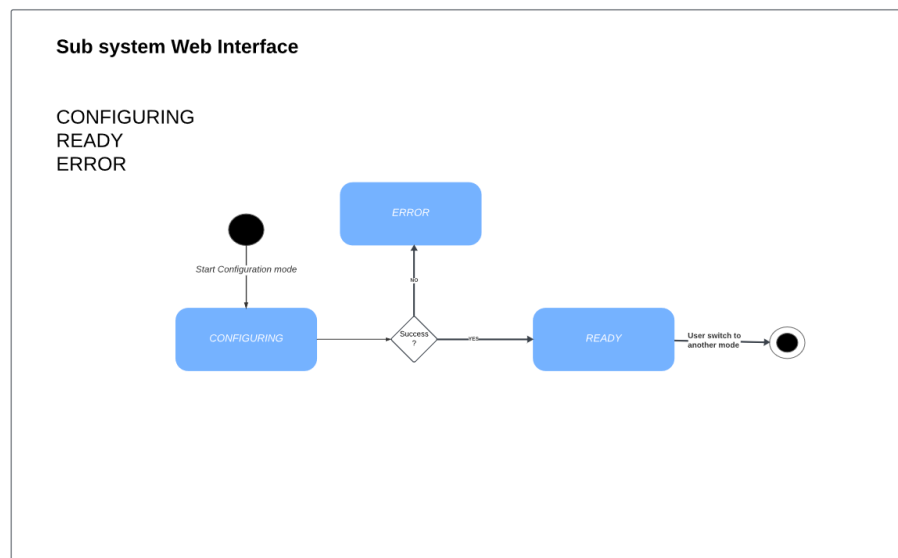


Figure 4: image

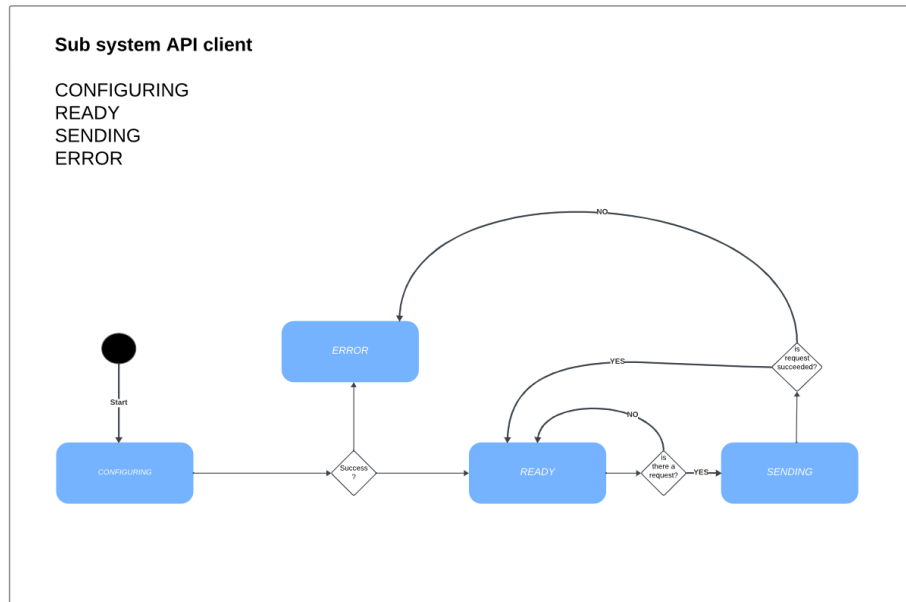


Figure 5: image

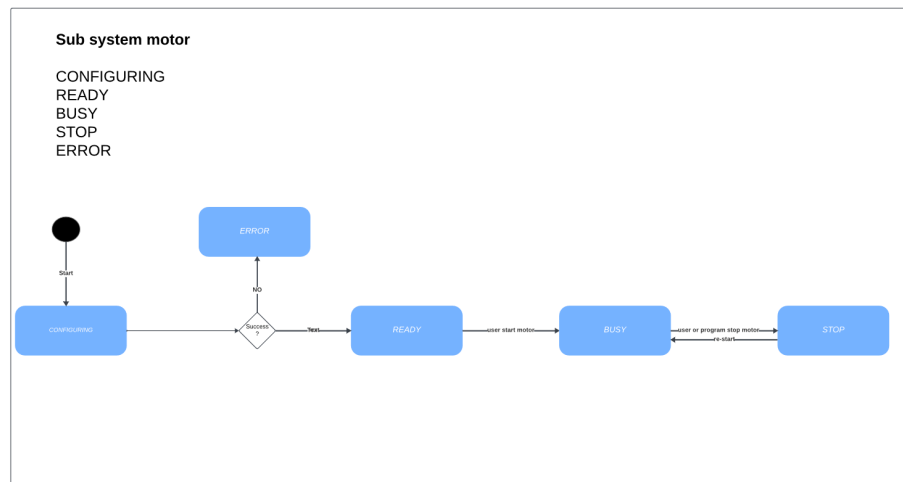


Figure 6: image

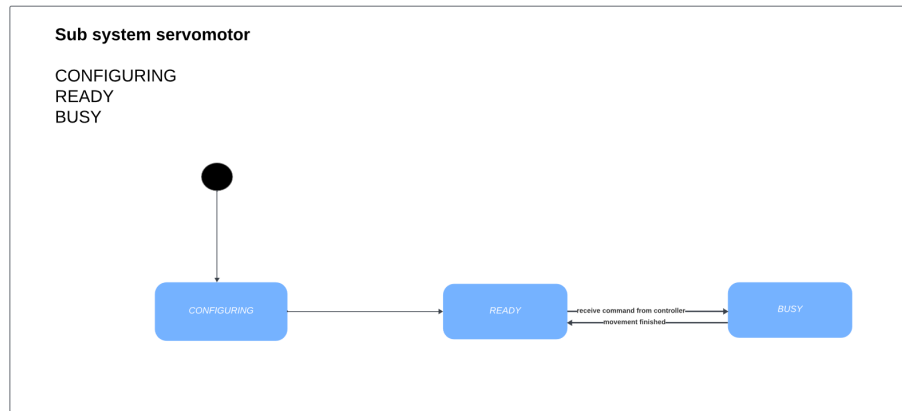


Figure 7: image

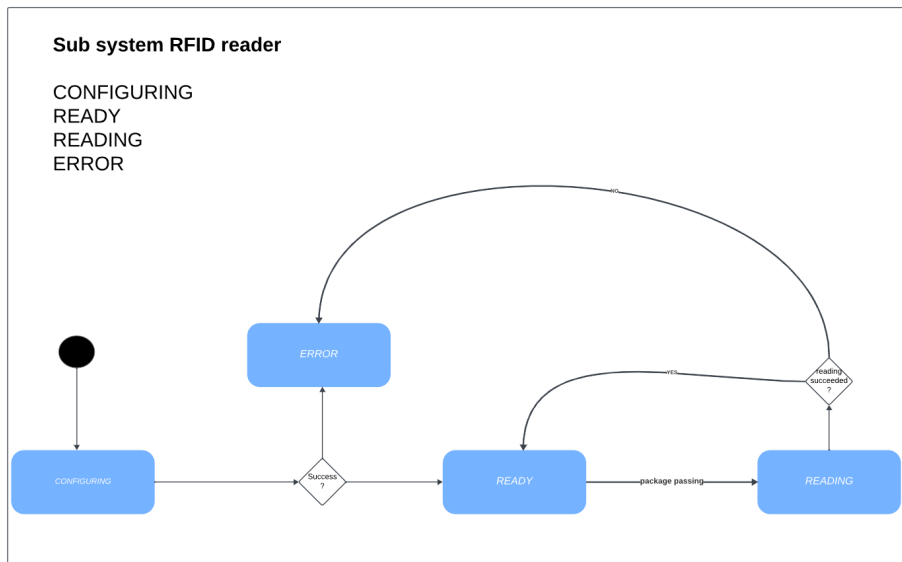


Figure 8: image