

Recherche de Solutions pour l'Acquisition des Données

 Créateur **Benjamin Bachelard**  Créé **Jan 23, 2025, 10:09**  Dernière mise à jour **Jan 23, 2025, 10:17**

1. Introduction

Ce document présente les solutions envisagées pour l'acquisition des données dans le cadre du projet de ruche autonome connectée. Les données à collecter incluent la température, l'humidité, la masse de la ruche, ainsi que des images provenant de caméras IR et standard. L'objectif est de garantir la fiabilité des données collectées tout en assurant leur sauvegarde et leur accessibilité.

2. Données à Acquérir

- **Température et humidité** à l'intérieur et à l'extérieur.
- **Masse de la ruche** pour suivre la production de miel.
- **Images IR** pour observer les abeilles à l'intérieur.
- **Images standard** pour observer les vols extérieurs.
- **Températures globales (matrice IR)** à travers le capteur AMG8833.

3. Protocoles de Communication des Capteurs

3.1 I2C (Inter-Integrated Circuit)

Le protocole I2C est utilisé pour connecter des capteurs tels que le DHT22 et le AMG8833. Il est également adapté pour les dispositifs nécessitant un transfert de données rapide sur des courtes distances.

Solutions logicielles avec Qt :

- Utilisation de la classe **QSerialBus** pour interfacer les capteurs via I2C. `#include <QSerialBus>`

3.2 SPI (Serial Peripheral Interface)

Le protocole SPI est utilisé pour le capteur HX711 afin de mesurer la masse de la ruche.

Solutions logicielles avec Qt :

- Intégration avec les classes Qt pour l'accès à bas niveau. `#include <QIODevice>`

3.3 Caméras (CSI IR et Standard)

Les caméras seront connectées via l'interface CSI (Camera Serial Interface).

Solutions logicielles avec Qt et OpenCV :

- Utilisation de **QImage** combinée avec OpenCV pour capturer et afficher des images.

`#include <QImage>`

`#include <opencv2/opencv.hpp>`

4. Stockage des Données

4.1 Sauvegarde Locale

- Utilisation de **QFile** pour écrire les données dans un fichier local.

```
#include <QFile>
```

```
#include <QTextStream>
```

4.2 Sauvegarde Distant

- Intégration avec des sockets Qt (QTcpSocket, QUdpSocket) ou MQTT via QtMqtt pour transmettre les données à un serveur distant.

5. Analyse et Traitement des Données

- Qt avec OpenCV :

6. Tests et Validation

- **Tests Unitaires :**
 - Intégration avec **Qt Test** pour automatiser les tests unitaires des capteurs et des modules de traitement.
- **Tests d'Intégration :**
 - Validation que toutes les données circulent correctement entre les capteurs et l'interface Qt.

7. Conclusion

L'utilisation de Qt en combinaison avec le langage C++ offre une solution robuste et flexible pour l'acquisition, le stockage et le traitement des données. Qt permet une intégration transparente avec les capteurs et offre des outils puissants pour développer une interface utilisateur intuitive.