

configHermes

Net and Light

2023 - 2024

Résumé

Cette documentation décrit le fichier de configuration (json) utilisé pour le projet de surveillance et de contrôle d'un API (Automate Programmable Industriel). Il inclut une vue d'ensemble des paramètres de configuration et de leurs utilisations.

Table des matières

| | | |
|----------|--|----------|
| 1 | Introduction | 2 |
| 2 | Configuration du fichier JSON | 2 |
| 2.1 | Structure générale | 2 |
| 2.2 | Détail des paramètres | 2 |
| 2.2.1 | Connexion à la base de données | 2 |
| 2.2.2 | Variables API en sortie | 2 |
| 2.2.3 | Variables API en entrée | 3 |
| 2.2.4 | Identifiant du châssis | 3 |
| 2.2.5 | Modes possibles | 3 |
| 2.2.6 | Journalisation | 4 |
| 2.2.7 | Configuration des broches GPIO | 4 |
| 2.2.8 | Serveur Flask | 5 |
| 2.2.9 | Connexion au Raspberry Pi | 5 |
| 2.2.10 | Fichiers locaux et distants | 5 |
| 3 | Conclusion | 6 |

1 Introduction

Cette documentation présente le fichier de configuration JSON "configHermes.json" pour un projet de surveillance et de contrôle d'API. Ce fichier définit les paramètres de connexion à la base de données, les variables API, la journalisation, les configurations GPIO, le serveur Flask, et autres configurations nécessaires pour le bon fonctionnement du système.

2 Configuration du fichier JSON

2.1 Structure générale

Le fichier de configuration est structuré en plusieurs sections :

- **Login_Turtle** : Paramètres de connexion à la base de données.
- **Var_API_Out** : Variables API en sortie.
- **Var_API_In** : Variables API en entrée.
- **id_chassis** : Identifiant du châssis.
- **possibles_modes** : Modes possibles pour l'API.
- **logging** : Paramètres de journalisation.
- **GPIO** : Configuration des broches GPIO.
- **flask_app** : Configuration du serveur Flask.
- **RASP_login** : Paramètres de connexion au Raspberry Pi.
- **LOCAL_file** : Configuration des fichiers locaux et distants.

2.2 Détail des paramètres

2.2.1 Connexion à la base de données

Les paramètres de connexion pour accéder à la base de données du châssis master.

- **user** : Nom d'utilisateur pour la connexion.
- **password** : Mot de passe pour la connexion.
- **host** : Adresse IP de la base de données.
- **database** : Nom de la base de données.

```
1 {  
2     "Login_Turtle": {  
3         "user": "root",  
4         "password": "billes1234",  
5         "host": "172.16.0.2",  
6         "database": "nuc_db"  
7     }  
8 }
```

Listing 1 – Connexion à la base de données

2.2.2 Variables API en sortie

Liste des variables API en sortie.

- **Var_API_Out** : Liste des noms des variables en sortie.

```
1 {  
2   "Var_API_Out": [  
3     "Mw_API_CVSortieStandard",  
4     "Mw_API_CVSortieSecours",  
5     "Mw_API_CVEntree"  
6   ]  
7 }
```

Listing 2 – Variables API en sortie

2.2.3 Variables API en entrée

Liste des variables API en entrée.

— **Var_API_In** : Liste des noms des variables en entrée.

```
1 {  
2   "Var_API_In": [  
3     "Mw_master_nb_billes_sortie_normal",  
4     "Mw_master_nb_billes_sortie_secours",  
5     "Mw_master_nb_billes_entree"  
6   ]  
7 }
```

Listing 3 – Variables API en entrée

2.2.4 Identifiant du châssis

L'identifiant unique pour le châssis.

— **id_chassis** : Identifiant numérique du châssis.

```
1 {  
2   "id_chassis": 0  
3 }
```

Listing 4 – Identifiant du châssis

2.2.5 Modes possibles

Liste des modes d'opération possibles pour l'API.

— **possibles_modes** : Liste des modes disponibles.

```
1 {  
2   "possibles_modes": ["mono", "multi"]  
3 }
```

Listing 5 – Modes possibles

2.2.6 Journalisation

Les paramètres de journalisation définissent où et comment les événements sont enregistrés.

- **file_path** : Chemin du fichier log.
- **log_level** : Niveau de log (ex : INFO, DEBUG).

```
1 {  
2   "logging": {  
3     "file_path": "/home/pi/Desktop/hermes.log",  
4     "log_level": "INFO"  
5   }  
6 }
```

Listing 6 – Configuration de la journalisation

2.2.7 Configuration des broches GPIO

Les paramètres des broches GPIO pour la matrice de LEDs et les combinaisons de couleurs.

- **led_matrix** : Configuration des positions et des couleurs des LEDs.
- **color_combination** : Combinaisons de couleurs pour les LEDs.
- **color_list** : Liste des couleurs disponibles.

```
1 {  
2   "GPIO": {  
3     "led_matrix": {  
4       "Position1": {"red": 17, "green": 27, "blue": 22,  
5         "white": 10},  
6       "Position2": {"red": 9, "green": 11, "blue": 5, "  
7         "white": 6},  
8       "Position3": {"red": 13, "green": 19, "blue": 26,  
9         "white": 18},  
10      "Position4": {"red": 23, "green": 24, "blue": 25,  
11        "white": 8},  
12      "Position5": {"red": 7, "green": 12, "blue": 16,  
13        "white": 20}  
14    },  
15    "color_combination": {  
16      "yellow": ["red", "green"],  
17      "cyan": ["green", "blue"],  
18      "magenta": ["red", "blue"]  
19    },  
20    "color_list": ["red", "yellow", "green", "cyan", "  
21      blue", "magenta", "white"]  
22  }  
23 }
```

Listing 7 – Configuration des broches GPIO

2.2.8 Serveur Flask

Les paramètres pour le serveur Flask définissent l'adresse et le port sur lesquels le serveur tourne.

- **host** : Adresse IP de l'hôte, ici "0.0.0.0" pour accepter toutes les connexions.
- **port** : Port sur lequel le serveur écoute, ici 8000.

```
1 {  
2     "flask_app": {  
3         "host": "0.0.0.0",  
4         "port": 8000  
5     }  
6 }
```

Listing 8 – Configuration du serveur Flask

2.2.9 Connexion au Raspberry Pi

Les paramètres de connexion pour accéder au Raspberry Pi.

- **user** : Nom d'utilisateur pour la connexion.
- **password** : Mot de passe pour la connexion.

```
1 {  
2     "RASP_login": {  
3         "user": "pi",  
4         "password": "pi"  
5     }  
6 }
```

Listing 9 – Connexion au Raspberry Pi

2.2.10 Fichiers locaux et distants

Les paramètres de configuration pour les fichiers locaux et distants utilisés dans l'application.

- **launch** : Script de lancement.
- **local_file_1** à **local_file_5** : Noms des fichiers locaux.
- **remote_file_1** à **remote_file_5** : Noms des fichiers distants.

```
1 {  
2     "LOCAL_file": {  
3         "launch": "~/Desktop/HermesEnlighteNed.py",  
4  
5         "local_file_1": "HermesEnlighteNed.py",  
6         "local_file_2": "ip_addressesRasp.py",  
7         "local_file_3": "CommunicationAPIS7.py",  
8         "local_file_4": "variablesAPI.py",  
9         "local_file_5": "configHermes.json",  
10  
11         "remote_file_1": "/home/pi/Desktop/HermesEnlighteNed.  
py",
```

```
12     "remote_file_2": "/home/pi/Desktop/ip_addresses.py",
13     "remote_file_3": "/home/pi/Desktop/CommunicationAPIS7
14         .py",
15     "remote_file_4": "/home/pi/Desktop/variablesAPI.py",
16     "remote_file_5": "/home/pi/Desktop/configHermes.json"
17 }
```

Listing 10 – Configuration des fichiers locaux et distants

3 Conclusion

Ce fichier de configuration JSON est essentiel pour la personnalisation et le bon fonctionnement du système de surveillance et de contrôle d'API. Il permet de régler les paramètres de connexion, de journalisation, de serveur, et d'autres configurations spécifiques à l'application.

Références

- [1] JSON Documentation, <https://www.json.org/json-en.html>.