

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
 Saisissez la référence ENI de l'ouvrage **LF3RASPYP** dans la zone de recherche
 et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

Chapitre 1

Raspberry Pi 4, premier contact

| | |
|---|----|
| 1. Introduction au Raspberry Pi 4 | 7 |
| 2. Démarrage de Raspberry Pi OS | 10 |
| 2.1 Historique | 10 |
| 3. Comprendre l'écosystème Python : quelle version utiliser ? | 15 |
| 4. Installer des bibliothèques Python | 16 |
| 4.1 La méthode aptitude | 16 |
| 4.2 La méthode pip | 18 |
| 4.3 pip ou aptitude ? | 20 |
| 5. IDLE : l'éditeur de code en Python pour Python | 20 |
| 6. Conclusion | 25 |

Chapitre 2

Python : bases et concepts avancés

| | |
|--|----|
| 1. Hello World | 27 |
| 2. Les types de base : int, float, str et bool | 28 |
| 2.1 L'entier : int | 28 |
| 2.2 Le flottant : float | 29 |
| 2.3 La chaîne de caractères : str | 29 |
| 2.4 Le booléen | 33 |

2 Python et Raspberry Pi - Apprenez à développer sur votre nano-ordinateur

| | |
|--|----|
| 3. Les structures de données : list, dict, tuple | 33 |
| 3.1 La liste | 33 |
| 3.2 Le tuple | 35 |
| 3.3 Le dictionnaire (ou table de hachage) | 35 |
| 3.4 L'ensemble : le set | 36 |
| 4. Les instructions, conditions et boucles | 37 |
| 4.1 La condition if | 37 |
| 4.2 La condition else | 37 |
| 4.3 La boucle for | 38 |
| 4.4 L'instruction break | 39 |
| 4.5 L'instruction continue | 39 |
| 4.6 La boucle while | 40 |
| 5. Les opérateurs | 41 |
| 5.1 Opérateurs arithmétiques | 41 |
| 5.2 Opérateurs logiques | 43 |
| 5.3 Opérateur d'appartenance | 44 |
| 5.4 Opérateur d'affectation | 44 |
| 6. La classe | 45 |
| 6.1 Premiers pas | 47 |
| 6.2 Exposer les attributs d'une classe | 47 |
| 6.3 Composition de classes | 49 |
| 6.4 Réutiliser du code | 50 |
| 7. Les fonctions | 52 |
| 7.1 Définir une fonction | 52 |
| 7.2 La fonction anonyme | 55 |
| 8. La syntaxe en compréhension | 57 |
| 9. Itérateur et générateur | 59 |
| 10. La gestion des exceptions | 64 |
| 11. L'import des modules avec le mot-clé import | 69 |
| 12. Le gestionnaire de contexte | 71 |
| 13. Conclusion | 74 |

Chapitre 3

Administration du Raspberry Pi en Python

| | |
|--|----|
| 1. Introduction | 75 |
| 2. Naviguer dans le système de fichiers avec les modules os et pwd | 76 |
| 2.1 Manipuler et interroger le système de fichiers | 78 |
| 2.2 Explorer le système de fichiers du Raspberry Pi | 79 |
| 3. Interagir avec l'interpréteur Python via le module sys | 83 |
| 4. Lancer des commandes shell avec le module subprocess | 85 |
| 5. Chercher des fichiers avec le module glob | 87 |
| 6. Comparer des fichiers ou répertoires avec le module filecmp | 89 |
| 7. Capturer des signaux UNIX avec le module signal | 91 |
| 8. Écriture de scripts avec le module argparse | 94 |
| 9. Conclusion | 98 |

Chapitre 4

Le Raspberry Pi en console avec urwid

| | |
|---|-----|
| 1. Introduction | 99 |
| 2. urwid, les fondamentaux | 99 |
| 3. Projet #1 : une horloge en console | 102 |
| 4. Projet #2 : un navigateur de fichiers en console | 104 |
| 5. Projet #3 : un hôte virtuel | 109 |
| 6. Conclusion | 111 |

4 Python et Raspberry Pi - Apprenez à développer sur votre nano-ordinateur

Chapitre 5

Programmation d'interfaces graphiques avec tkinter

| | |
|---|-----|
| 1. Les fondamentaux | 113 |
| 2. Projet #1 : Hello world avec tkinter | 116 |
| 3. Projet #2 : une visionneuse d'images | 123 |
| 4. Projet #3 : un éditeur de texte | 131 |
| 5. Conclusion | 142 |

Chapitre 6

À l'assaut du Web avec le Raspberry Pi

| | |
|--|-----|
| 1. Webscraping facile avec les modules requests et html | 143 |
| 2. Développer un serveur HTTP avec le module http.server | 148 |
| 3. Exécuter des scripts avec le module cgi | 154 |
| 4. Envoyer des e-mails avec le module smtplib | 161 |
| 5. Écrire une API légère avec Flask | 164 |
| 6. Conclusion | 172 |

Chapitre 7

Multimédia et audio sur le Raspberry Pi

| | |
|--|-----|
| 1. Dessiner avec Pillow | 173 |
| 1.1 Créer et manipuler des images | 173 |
| 1.2 Dessiner des figures géométriques | 180 |
| 2. Contrôler les entrées et sorties audio avec pyalsaaudio | 182 |
| 3. Projet #1 : un enregistreur/lecteur audio | 189 |
| 4. Conclusion | 194 |

Chapitre 8

Persistance de données sur le Raspberry Pi

| | |
|---|-----|
| 1. Introduction | 195 |
| 2. Sérialisation et désérialisation avec les modules pickle et shelve | 195 |
| 3. Traiter des fichiers CSV avec le module csv | 199 |
| 3.1 Création et lecture d'un fichier CSV | 199 |
| 3.2 Créer son propre dialecte CSV | 201 |
| 4. Manipuler des données XML avec le module xml.etree.ElementTree | 202 |
| 4.1 Créer et sérialiser un fichier XML | 203 |
| 4.2 Interroger un fichier XML | 204 |
| 4.3 Ajouter et supprimer des nœuds | 205 |
| 5. Travailler avec le format d'échange de données JSON via le module json | 207 |
| 6. Gestion d'une base de données SQL légère avec le module sqlite3 | 209 |
| 7. Conclusion | 214 |

Chapitre 9

Documenter et tester ses scripts en Python

| | |
|---|-----|
| 1. Introduction | 215 |
| 2. Consulter de la documentation avec pydoc3 | 216 |
| 3. Documenter et tester son code en une seule fois avec le module doctest | 223 |
| 4. Écriture de tests unitaires avec le module unittest | 227 |
| 5. Benchmarkner son code avec le module timeit | 232 |
| 6. Déboguer ses programmes avec le module pdb | 236 |
| 6.1 Déboguer pas à pas | 237 |
| 6.2 Déboguer à un endroit précis du programme | 243 |
| 7. Conclusion | 244 |

6 Python et Raspberry Pi - Apprenez à développer sur votre nano-ordinateur

Chapitre 10

Raspberry Pi et GPIO

| | |
|--|-----|
| 1. Les GPIO, comment ça marche ? | 245 |
| 2. Connecter un écran LCD 16x2 au Raspberry Pi | 247 |
| 3. Projet #1 : communiquer avec l'écran LCD | 253 |
| 4. Projet #2 : créer un tube FIFO dédié à l'écran LCD | 256 |
| 5. Projet #3 : écrire des messages depuis une interface en ligne de commande | 261 |
| 6. Projet #4 : piloter l'écran LCD depuis une interface graphique tkinter | 264 |
| 7. Conclusion | 267 |

| | |
|-------------|--|
| Index | |
|-------------|--|

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
 Saisissez la référence ENI de l'ouvrage **LF2PYRASPFL** dans la zone de recherche
 et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Préface

Chapitre 1

Présentation

| | |
|---|----|
| 1. Avant-propos..... | 17 |
| 2. Motivations | 18 |
| 3. Présentation du projet | 19 |
| 4. Objectifs de l'ouvrage | 21 |
| 5. Prérequis | 22 |
| 6. Matériel utilisé | 23 |
| 6.1 Raspberry Pi..... | 23 |
| 6.2 Feather ESP8266 Huzzah | 25 |
| 6.3 Feather ESP32 Huzzah | 26 |
| 6.4 Feather OLED Wing..... | 27 |
| 6.5 Module relais | 27 |
| 6.6 DHT11 - Humidité | 28 |
| 6.7 AM2315 - Température et humidité | 29 |
| 6.8 DS18B20 - Température..... | 29 |
| 6.9 BME280 et BMP280 - Pression, humidité, température..... | 30 |
| 6.10 TSL2561 - Luminosité | 31 |
| 6.11 ADS1115 - Lecture analogique | 31 |
| 6.12 TMP36 - Température | 32 |
| 6.13 Photorésistance - Luminosité | 32 |
| 6.14 PIR - Détection de mouvement..... | 33 |
| 6.15 Contact magnétique..... | 33 |
| 6.16 Capteur à effet Hall numérique..... | 34 |
| 6.17 CCS811 - Mesure de CO2..... | 35 |

2 Python, Raspberry Pi et Flask - Données télémétriques et tableaux de bord web

| | |
|--|----|
| 6.18 Capteur de courant non invasif | 36 |
| 7. Code source | 37 |
| 7.1 Téléchargement | 37 |
| 7.2 GitHub | 37 |
| 8. Configuration | 39 |
| 8.1 Installation du Raspberry Pi | 39 |
| 8.2 Utilitaires : des outils pour travailler | 54 |
| 8.2.1 Connexion SSH | 54 |
| 8.2.2 Éditeur de texte Nano | 55 |
| 8.2.3 Transfert de fichiers via SSH (sftp) | 57 |
| 8.2.4 Système de fichiers SSH | 59 |
| 8.2.5 Bureau à distance | 62 |
| 9. Type de données collectées | 64 |

Chapitre 2

Le broker MQTT

| | |
|--|----|
| 1. Présentation et concepts | 65 |
| 1.1 Le broker MQTT, élément central du réseau MQTT | 67 |
| 1.2 Les éléments de MQTT | 69 |
| 1.3 Le broker MQTT | 69 |
| 1.4 Les topics | 70 |
| 1.5 Les publishers | 70 |
| 1.6 Les subscribers | 71 |
| 1.7 Le ClientId | 72 |
| 2. Révisions de MQTT | 72 |
| 3. Les topics en détail | 73 |
| 3.1 Contenu du message | 74 |
| 3.1.1 Le message selon MQTT | 74 |
| 3.1.2 En marge du standard | 75 |
| 3.2 Création de topic et bonnes pratiques | 76 |
| 3.3 Les topics système | 78 |
| 4. Souscription et expression de filtrage | 79 |
| 4.1 Expression de filtrage sans joker | 79 |
| 4.2 Le joker de niveau | 80 |

| | |
|---|-----|
| 4.3 Le joker multiniveau..... | 80 |
| 5. Les qualités de service MQTT..... | 81 |
| 5.1 Les niveaux de qualité..... | 82 |
| 5.1.1 QoS 0 : une fois maximum..... | 82 |
| 5.1.2 QoS 1 : au moins une fois..... | 82 |
| 5.1.3 QoS 2 : exactement une fois..... | 83 |
| 5.2 Rétrogradation de QoS..... | 84 |
| 5.3 Quel QoS utiliser et quand ?..... | 85 |
| 6. La rétention de messages..... | 86 |
| 7. Les clients persistants..... | 87 |
| 8. Message testament..... | 88 |
| 9. Quel broker MQTT ?..... | 90 |
| 10. Installation de Mosquitto..... | 93 |
| 10.1 Mise à jour..... | 93 |
| 10.2 Installation..... | 94 |
| 11. Test avec Mosquitto.org..... | 95 |
| 11.1 La souscription..... | 96 |
| 11.2 La publication..... | 96 |
| 11.3 Tester le broker MQTT du Raspberry Pi..... | 98 |
| 12. Topics du projet..... | 99 |
| 13. QoS du projet..... | 102 |
| 14. Sécurité..... | 103 |
| 15. Configurer le login du broker MQTT..... | 104 |
| 15.1 Modifier la configuration..... | 104 |
| 15.2 Tester la configuration..... | 105 |
| 16. MQTT en Python..... | 107 |
| 16.1 test-mqtt-client-sub.py..... | 107 |
| 16.2 test-mqtt-client-pub.py..... | 111 |
| 16.3 Documentation complémentaire..... | 113 |
| 17. MQTT en MicroPython..... | 113 |

4 Python, Raspberry Pi et Flask - Données télémétriques et tableaux de bord web

Chapitre 3

ESP8266 et ESP32 sous MicroPython

| | |
|--|-----|
| 1. Présentation des ESP | 115 |
| 1.1 Les possibilités offertes par ESP | 118 |
| 1.2 Les plateformes ESP populaires | 120 |
| 1.2.1 Les plateformes ESP8266 | 120 |
| 1.2.2 Les plateformes ESP32 | 123 |
| 1.3 Programmer un ESP | 126 |
| 1.4 Feather Huzzah ESP8266 et ESP32 en détail | 127 |
| 1.5 Brochage des Feather Huzzah ESP8266 et ESP32 | 130 |
| 1.5.1 Alimentation | 130 |
| 1.5.2 Port série | 131 |
| 1.5.3 Broches d'entrée/sortie | 132 |
| 1.5.4 Les autres broches | 138 |
| 2. Charger le firmware MicroPython | 138 |
| 2.1 Identifier le firmware MicroPython | 139 |
| 2.2 Préparatifs | 141 |
| 2.3 Reflasher un ESP | 143 |
| 3. Prise de contrôle | 147 |
| 3.1 Communiquer avec MicroPython | 147 |
| 3.2 Communiquer avec un ESP sous MicroPython | 148 |
| 3.3 REPL : l'invite de commandes MicroPython | 149 |
| 3.4 RShell | 151 |
| 3.5 Ampy | 160 |
| 4. WebREPL | 166 |
| 4.1 Le démon WebREPL | 167 |
| 4.1.1 Activer WebREPL sur l'ESP | 167 |
| 4.1.2 Le mot de passe WebREPL | 169 |
| 4.2 Le client WebREPL | 169 |
| 5. Nom d'hôte et adresse MAC | 170 |
| 6. Le mode point d'accès (AP) | 171 |
| 7. Le mode station (STA) | 174 |
| 7.1 Mode STA et scan réseau | 175 |
| 7.2 Réseau Wi-Fi visible ou masqué | 176 |
| 7.3 Connexion en mode STA | 177 |

| | | |
|-------|---|-----|
| 7.4 | WebREPL en mode STA | 178 |
| 7.5 | Désactivation du point d'accès | 179 |
| 7.6 | Rechercher l'adresse IP d'un ESP | 180 |
| 8. | Séquence de démarrage MicroPython | 181 |
| 8.1 | Fichier boot.py | 181 |
| 8.2 | Fichier main.py | 182 |
| 8.3 | Un fichier boot.py pour ESP | 182 |
| 8.3.1 | Script trop optimiste et conséquences | 183 |
| 8.3.2 | RunApp - Activation de l'application | 184 |
| 8.3.3 | Un script de boot avancé | 186 |
| 9. | Programmer | 188 |
| 9.1 | Création d'une bibliothèque | 188 |
| 9.2 | Les bibliothèques MicroPython | 191 |
| 9.2.1 | Bibliothèques standards et microbibliothèques | 191 |
| 9.2.2 | Bibliothèques spécifiques à MicroPython | 192 |
| 9.2.3 | Bibliothèque spécifique à l'ESP | 193 |
| 9.2.4 | Autres bibliothèques MicroPython | 193 |
| 9.2.5 | Mécanisme de chargement d'une bibliothèque | 193 |
| 9.3 | Charger et exécuter un script à la volée | 194 |
| 9.4 | RunApp : exécution conditionnelle de main.py | 196 |
| 9.5 | Entrées/sorties sur ESP | 198 |
| 9.5.1 | Entrée numérique | 198 |
| 9.5.2 | Entrée numérique (pull-up interne) | 200 |
| 9.5.3 | Entrée numérique et déparasitage logiciel | 202 |
| 9.5.4 | Sortie numérique | 205 |
| 9.5.5 | Entrée analogique sur ESP8266 | 206 |
| 9.5.6 | Entrées analogiques sur ESP32 | 209 |
| 9.5.7 | Ajout d'entrée/sortie avec MCP23017 | 212 |
| 9.5.8 | Lecture analogique avec l'ADS1115 | 216 |
| 9.6 | Capteur et interface sur ESP | 219 |
| 9.6.1 | Capteur PIR - capteur de proximité | 220 |
| 9.6.2 | Contact magnétique | 221 |
| 9.6.3 | DHT11 - humidité | 223 |
| 9.6.4 | Capteur à effet Hall | 225 |
| 9.6.5 | TSL2561 - luminosité | 227 |
| 9.6.6 | TSL2591 - luminosité | 231 |
| 9.6.7 | BME280 - température, humidité et pression barométrique | 233 |

6 Python, Raspberry Pi et Flask - Données télémétriques et tableaux de bord web

| | |
|--|-----|
| 9.6.8 CCS811 – capteur de qualité d'air | 235 |
| 9.6.9 MOD-ENV – capteur environnemental | 237 |
| 9.6.10 Module relais | 240 |
| 9.6.11 OLED FeatherWing – afficheur OLED | 244 |
| 10. MQTT sous ESP | 248 |
| 10.1 Publication MQTT sous MicroPython | 248 |
| 10.2 Souscription MQTT sous MicroPython | 251 |
| 11. Asyncio sur ESP | 257 |
| 11.1 Asyncio en quelques mots | 257 |
| 11.2 Installer Asyncio | 258 |
| 11.3 Asyncio par l'exemple | 258 |
| 11.4 Fonction run_every pour Asyncio | 262 |
| 11.5 Plus d'informations sur Asyncio | 264 |
| 12. Abstraction matérielle ESP8266/ESP32 | 265 |
| 12.1 Abstraction du bus I2C | 265 |
| 12.2 Abstraction de la LED utilisateur | 266 |

Chapitre 4

Les objets ESP

| | |
|--|-----|
| 1. Informations pratiques | 269 |
| 1.1 Prérequis et configurations | 269 |
| 1.2 LED de statut | 270 |
| 1.3 Les topics MQTT | 271 |
| 1.4 Télécharger et préparer le code des objets IoT | 271 |
| 2. Fonctionnement général d'un objet IoT | 274 |
| 2.1 Principales sections | 277 |
| 2.2 Paramètres d'un objet IoT | 278 |
| 2.3 RunApp et LED d'activité | 279 |
| 2.4 La fonction led_error() | 279 |
| 2.5 Les tâches et fonctions asynchrones des objets IoT | 280 |
| 3. Objet 1 : Météo cabane de jardin | 283 |
| 3.1 Schéma de raccordement | 283 |
| 3.2 Téléverser les scripts | 284 |
| 3.3 Fonctionnement du script | 285 |

| | | |
|-----|---|-----|
| 3.4 | Tester l'objet | 290 |
| 4. | Objet 1a : Cabane complément | 291 |
| 4.1 | Schéma de raccordement | 291 |
| 4.2 | Téléverser les scripts | 291 |
| 4.3 | Fonctionnement du script | 292 |
| 5. | Objet 2 : Surveillance salon | 293 |
| 5.1 | Téléverser les scripts | 294 |
| 5.2 | Fonctionnement du script | 295 |
| 5.3 | La fonction capture_1h0 | 300 |
| 5.4 | Capteur PIR – variables et utilisation | 301 |
| 5.5 | Senseur PIR - la fonction pir_activated | 302 |
| 5.6 | Capteur PIR - la fonction pir_alert | 302 |
| 5.7 | Capteur PIR - la fonction pir_update | 303 |
| 5.8 | Problèmes de concurrence | 304 |
| 5.9 | Tester l'objet | 304 |
| 6. | Objet 3 : Surveillance de la véranda | 305 |
| 6.1 | Téléverser les scripts | 306 |
| 6.2 | Fonctionnement du script | 306 |
| 6.3 | La fonction capture_1h0 | 311 |
| 6.4 | La fonction check_contact0 | 312 |
| 6.5 | La fonction check_ldr0 | 313 |
| 6.6 | Tester l'objet | 315 |
| 7. | Objet 4 : Chaufferie | 316 |
| 7.1 | Téléverser les scripts | 317 |
| 7.2 | Fonctionnement du script | 318 |
| 7.3 | La fonction capture_1h0 | 325 |
| 7.4 | La fonction capture_10m0 | 326 |
| 7.5 | La fonction check_mqtt_sub0 | 326 |
| 7.6 | La fonction sub_cb0 | 327 |
| 7.7 | La fonction chaud_exec_cmd0 | 328 |
| 7.8 | Tester l'objet | 329 |
| 8. | Objet 5 : Commande de chaufferie | 330 |
| 8.1 | Téléverser les scripts | 332 |
| 8.2 | Fonctionnement du script | 332 |
| 8.3 | Gestion du changement d'état | 337 |
| 8.4 | Réception des données MQTT | 338 |

8 Python, Raspberry Pi et Flask - Données télémétriques et tableaux de bord web

| | |
|--|-----|
| 9. Objet 5 : Surveillance environnementale | 341 |
| 9.1 Téléverser les scripts | 342 |
| 9.2 Fonctionnement du script | 342 |
| 9.3 La fonction capture_5min() | 347 |
| 9.4 Tester l'objet | 348 |
| 10. Dépannage d'un objet IoT | 349 |

Chapitre 5

Persistance des données

| | |
|---|-----|
| 1. Introduction | 351 |
| 1.1 Pourquoi utiliser une base de données ? | 351 |
| 1.2 Quel moteur de base de données ? | 352 |
| 1.3 Principe de fonctionnement de push-to-db | 353 |
| 2. SQLite 3 | 354 |
| 2.1 Présentation | 354 |
| 2.2 Classe de stockage, type de données et affinité | 355 |
| 2.2.1 Classe de stockage | 356 |
| 2.2.2 Stockage des date et heure | 357 |
| 2.2.3 Affinité de type pour les colonnes | 357 |
| 2.2.4 Résolution de l'affinité de type | 359 |
| 2.3 Affinité, expressions, comparaison et tri | 359 |
| 2.3.1 Affinité des expressions | 359 |
| 2.3.2 Comparaison, tri et groupage | 362 |
| 2.4 Clé primaire et auto-incrément | 363 |
| 2.4.1 Définir une clé primaire | 363 |
| 2.4.2 Table rowid et clé primaire | 364 |
| 2.5 SQLite 3 et accès concurrents | 365 |
| 2.6 Installation | 367 |
| 2.6.1 Installer SQLite 3 | 367 |
| 2.6.2 Installer le support Python | 368 |
| 2.7 Premiers pas avec SQLite 3 | 368 |
| 2.7.1 Documentation SQL pour SQLite | 370 |
| 2.7.2 Commandes de l'interpréteur SQLite | 370 |

| | |
|---|-----|
| 2.8 SQLite et Python | 373 |
| 2.8.1 Opération de lecture SQLite | 373 |
| 2.8.2 Opération d'insertion SQLite | 376 |
| 2.8.3 Row Factory de SQLite | 378 |
| 3. Approches techniques de push-to-db | 379 |
| 3.1 Approche base de données de push-to-db | 380 |
| 3.1.1 topicmsg – dernier message reçu | 380 |
| 3.1.2 ts_xxx – historique de messages | 381 |
| 3.2 Approche logicielle de push-to-db | 383 |
| 3.2.1 Diagramme des classes (partie 1) | 383 |
| 3.2.2 Fichier de configuration de push-to-db | 389 |
| 3.2.3 Diagramme des classes (partie 2) | 392 |
| 4. Configuration de push-to-db | 414 |
| 4.1 Les répertoires de stockage de push-to-db | 415 |
| 4.2 Création des tables de push-to-db | 416 |
| 4.3 push-to-db.ini | 417 |
| 4.4 Le script d'installation de push-to-db | 421 |
| 5. Logger Python | 423 |
| 5.1 Logger et fichier de configuration | 423 |
| 5.2 Configuration du logger | 424 |
| 5.3 Utilisation du logger | 426 |
| 6. Exécution du script push-to-db | 426 |
| 7. Service systemd pour push-to-db | 427 |
| 7.1 Quand démarrer le service ? | 428 |
| 7.2 Créer le fichier Unit | 428 |
| 7.3 Configurer, démarrer, contrôler | 429 |
| 7.4 Documentation sur systemd | 430 |
| 8. Améliorations | 430 |

10 Python, Raspberry Pi et Flask - Données télémétriques et tableaux de bord web

Chapitre 6

Développement web en Python

| | |
|---|-----|
| 1. Présentation de Flask..... | 433 |
| 1.1 Pourquoi Flask ?..... | 434 |
| 1.2 La flexibilité de Flask..... | 435 |
| 1.3 Les nombreuses extensions Flask..... | 435 |
| 1.4 Flask plus en détail..... | 437 |
| 1.4.1 Werkzeug..... | 438 |
| 1.4.2 WSGI..... | 439 |
| 1.4.3 Application Flask..... | 440 |
| 1.4.4 Jinja..... | 440 |
| 1.4.5 Base de données..... | 441 |
| 1.5 Documentations..... | 441 |
| 2. Anatomie d'un projet Flask..... | 442 |
| 3. Installation et prise en main..... | 443 |
| 3.1 L'utilitaire flask..... | 445 |
| 3.2 Prise en main avancée..... | 446 |
| 3.3 Déboguer avec Flask..... | 450 |
| 3.4 Application Flask en production..... | 454 |
| 4. Les fondamentaux de Flask..... | 455 |
| 4.1 Routes et paramètres..... | 455 |
| 4.2 Retourner une erreur..... | 464 |
| 4.3 Utilisation de template..... | 465 |
| 4.4 Création d'URL..... | 468 |
| 4.5 Redirection..... | 469 |
| 4.6 Requêtes GET et POST..... | 472 |
| 4.7 Contexte applicatif..... | 476 |
| 4.7.1 L'objet g..... | 476 |
| 4.7.2 Connexion à la base de données..... | 477 |
| 4.8 Les cookies..... | 477 |
| 4.9 Les sessions..... | 479 |
| 4.10 Journalisation..... | 480 |
| 4.11 Mini-projet Fruits..... | 483 |
| 4.11.1 Sources du mini-projet..... | 484 |
| 4.11.2 La connexion SQLite 3..... | 485 |
| 4.11.3 Organisation du mini-projet..... | 486 |

| | | |
|--------|---|-----|
| 4.11.4 | Détails du mini-projet | 486 |
| 4.12 | Ressources et documentations | 500 |
| 5. | Templates Jinja | 501 |
| 5.1 | Exécution d'un template | 501 |
| 5.2 | Tester un template | 502 |
| 5.2.1 | Créer une application Flask | 502 |
| 5.2.2 | Test avec serveur web Flask et string Python | 502 |
| 5.2.3 | Test en console et string Python | 503 |
| 5.2.4 | Utiliser le projet Jinja Live Parser | 504 |
| 5.3 | Évaluation des balises | 506 |
| 5.3.1 | {{ ... }} : évaluation d'expression | 506 |
| 5.3.2 | {% ... %} : instructions de contrôle de flux | 507 |
| 5.3.3 | {# ... #} : insertion de commentaire | 507 |
| 5.3.4 | # ... : ligne d'instruction | 507 |
| 5.4 | Variables et expressions | 508 |
| 5.4.1 | Variables spéciales | 510 |
| 5.4.2 | Séquence d'échappement | 511 |
| 5.4.3 | Assignation | 511 |
| 5.5 | Branchement | 511 |
| 5.6 | Itération | 512 |
| 5.7 | Les macros | 516 |
| 5.8 | Contrôle des espaces | 518 |
| 5.9 | Filtres Jinja | 520 |
| 5.10 | Inclusion de template | 526 |
| 5.11 | Importer des macros | 529 |
| 5.12 | Héritage de template | 530 |
| 5.12.1 | Les éléments de l'héritage | 533 |
| 5.12.2 | Heritage-app : l'héritage Jinja par la pratique | 533 |
| 5.12.3 | Template de base et blocs | 534 |
| 5.13 | Template enfant | 536 |
| 5.13.1 | Superbloc | 537 |
| 5.13.2 | Ressources | 539 |
| 5.14 | Message Flash | 539 |

12 Python, Raspberry Pi et Flask - Données télémétriques et tableaux de bord web

Chapitre 7

Le tableau de bord

| | |
|---|-----|
| 1. Présentation | 551 |
| 1.1 Préambule | 551 |
| 1.2 Dépôt du projet Dashboard | 553 |
| 1.3 Éléments principaux | 554 |
| 1.4 Fonctionnalités du projet Dashboard | 559 |
| 2. Structure HTML | 571 |
| 2.1 Disposition de la page | 571 |
| 2.2 Les blocs d'informations | 576 |
| 2.3 La liste | 578 |
| 3. Template Jinja | 579 |
| 3.1 Le template de base | 579 |
| 3.2 Utilisation du template de base | 585 |
| 4. Configuration | 588 |
| 4.1 Base de données dashboard.db | 588 |
| 4.1.1 Schéma de la base de données | 589 |
| 4.1.2 Répertoire de stockage | 592 |
| 4.1.3 Création des tables de Dashboard | 592 |
| 4.1.4 Copie de la base de données | 593 |
| 4.2 Fichier de configuration de Dashboard | 593 |
| 5. Détails de l'application Flask | 596 |
| 5.1 Répertoires et fichiers | 596 |
| 5.2 Les routes de Dashboard | 602 |
| 5.3 Accès aux données | 603 |
| 5.3.1 La fonction get_db(db_key) multibases de données | 603 |
| 5.3.2 Les classes DBHelper de Dashboard | 607 |
| 5.3.3 Exemple : liste des topics disponibles pour Dashboard | 611 |
| 5.3.4 Exemple : extraction de l'historique dans Dashboard | 612 |
| 5.3.5 Affichage d'un tableau de bord | 615 |
| 5.4 Les filtres Jinja personnalisés | 622 |
| 5.5 Affichage du tableau de bord | 625 |
| 5.6 Les macros Jinja | 630 |
| 5.6.1 La macro make_block | 630 |
| 5.6.2 La macro block_icon | 632 |

| | | |
|-------|--|-----|
| 5.6.3 | La macro <code>block_big_text</code> | 634 |
| 5.6.4 | La macro <code>select_color</code> (édition d'un bloc) | 635 |
| 6. | Bloc SWITCH (marche/arrêt) | 640 |
| 6.1 | Développements complémentaires | 643 |
| 6.1.1 | MQTT sources | 643 |
| 6.1.2 | Bloc et paramètres additionnels | 646 |
| 6.2 | Ajout du bloc SWITCH | 649 |
| 6.2.1 | <code>Block_config</code> du switch | 649 |
| 6.2.2 | Ajouter le nouveau type de bloc | 650 |
| 6.3 | Le switch et MQTT | 653 |
| 6.3.1 | Client MQTT JavaScript | 655 |
| 6.3.2 | MQTT en JavaScript et WebSocket | 656 |
| 6.3.3 | Activer le support WebSocket sur Mosquitto | 657 |
| 6.3.4 | Tester le client MQTT JavaScript | 657 |
| 6.3.5 | Alternative à MQTT en JavaScript | 659 |
| 6.3.6 | La route <code>MqttProxyPublish</code> | 661 |
| 6.3.7 | Événement <code>on_switch_change</code> | 662 |
| 6.4 | Tester le bloc SWITCH | 664 |
| 7. | Améliorations | 668 |

Chapitre 8

InfluxDB

| | | |
|-----|--|-----|
| 1. | Introduction | 669 |
| 2. | InfluxDB | 669 |
| 2.1 | Présentation | 669 |
| 2.2 | Concepts InfluxDB | 673 |
| 3. | Installation | 674 |
| 4. | Premiers pas avec InfluxDB | 675 |
| 4.1 | Quelques commandes utiles | 675 |
| 4.2 | Autocritique | 680 |
| 5. | Commandes de l'interpréteur | 681 |
| 5.1 | Commandes InfluxDB | 681 |
| 5.2 | Paramètres de l'utilitaire <code>influx</code> | 682 |

14 Python, Raspberry Pi et Flask - Données télémétriques et tableaux de bord web

| | |
|---|-----|
| 6. La commande SELECT | 682 |
| 6.1 Multiple sources | 683 |
| 6.2 Nom de table qualifié | 685 |
| 6.3 La clause WHERE | 686 |
| 6.4 Expressions régulières | 687 |
| 6.5 Casting de type | 688 |
| 7. Importer des données | 688 |
| 8. Règles de rétention | 693 |
| 8.1 Accès aux données | 694 |
| 8.2 Requêtes continues | 695 |
| 8.3 Rétention et requêtes continues par la pratique | 695 |
| 8.4 Bonnes pratiques de conception | 700 |
| 8.5 Documentation pour InfluxDB | 701 |
| 9. InfluxDB API | 702 |
| 9.1 Créer un utilisateur | 703 |
| 9.2 Activer l'authentification sur InfluxDB | 703 |
| 9.3 Tester l'authentification | 704 |
| 9.4 Tester InfluxDB API | 705 |
| 10. Capture des données MQTT | 706 |
| 11. Greffon Telegraf MQTT pour InfluxDB | 708 |
| 11.1 Installation | 708 |
| 11.1.1 Mise à jour des dépôts | 709 |
| 11.1.2 Installation de telegraf | 709 |
| 11.2 Configuration | 710 |
| 11.2.1 Configuration du service Telegraf | 710 |
| 11.2.2 Configuration telegraf avancée | 714 |
| 12. InfluxDB et Python | 717 |
| 12.1 Installation | 717 |
| 12.2 Opération de lecture | 718 |
| 12.3 Opération d'écriture | 720 |
| 13. Duplicata de champs | 721 |
| 14. Ensuite | 721 |

Chapitre 9

Grafana

| | |
|---|-----|
| 1. Introduction | 723 |
| 2. Les concepts fondamentaux | 724 |
| 2.1 Les sources de données | 725 |
| 2.2 Éditeur de requêtes | 726 |
| 2.3 Organisation | 726 |
| 2.4 Utilisateur | 727 |
| 2.5 Panneau | 727 |
| 2.6 Ligne | 729 |
| 2.7 Tableau de bord | 729 |
| 3. Installer Grafana | 731 |
| 3.1 Installation du dépôt | 731 |
| 3.2 Première connexion et mot de passe | 732 |
| 4. Les données InfluxDB | 734 |
| 4.1 Collecter rapidement des données | 734 |
| 4.2 Améliorer la fiabilité des mesures | 736 |
| 4.3 Contrôler la capture de données | 737 |
| 5. Afficher les données InfluxDB | 737 |
| 5.1 Source de données InfluxDB | 738 |
| 5.2 Créer un tableau de bord | 742 |
| 5.2.1 Ajouter un panneau | 743 |
| 5.2.2 Sélectionner la série pour la cabane | 745 |
| 5.3 Ajouter une seconde série dans le graphique | 748 |
| 5.4 Exploration des données | 750 |
| 5.5 Autre visualisation | 751 |
| 5.6 Quelques astuces Grafana | 756 |
| 5.6.1 Redimensionner un panneau | 756 |
| 5.6.2 Glisser/déposer un panneau | 756 |
| 5.6.3 Transformer en ligne | 757 |
| 5.6.4 Enregistrer le tableau de bord | 759 |
| 6. Limites de Grafana | 759 |
| 6.1 Liens Internet sur un panneau | 760 |
| 6.2 Lien vers MqttProxyPublish | 762 |
| 6.3 Liens Grafana et méthode GET | 763 |

16 Python, Raspberry Pi et Flask - Données télémétriques et tableaux de bord web

Conclusion

| | |
|--|-----|
| 1. Introduction | 765 |
| 2. Remerciements..... | 766 |
| 3. Retour sur client MQTT JavaScript | 767 |

Annexes

| | |
|------------------------------------|-----|
| 1. Installation rapide..... | 769 |
| 1.1 Prérequis..... | 769 |
| 1.2 Début de l'installation | 769 |
| 1.3 Récupération des sources | 770 |
| 1.4 Poursuivre l'installation..... | 771 |

| | |
|------------|-----|
| Index..... | 775 |
|------------|-----|