

Qu'est-ce que MicroPython ?

1. Mise en garde : euphorie imminente !	19
2. Avant-propos : premier contact MicroPython	19
3. Objectifs de l'ouvrage	21
4. Prérequis	22
5. Présentation de MicroPython	23
6. Comparaison MicroPython et Arduino	26
6.1 Arduino	26
6.2 MicroPython	28
7. Intérêt	29
7.1 Python pour l'électronique	30
7.2 Python, un langage populaire	30
7.3 Python et apprentissage rapide	31
7.4 Python et l'enseignement	31
8. Communauté	32
8.1 Bibliothèques et pilotes	

8.2 Forums	32
	34
Plateformes MicroPython	
1. Préambule	35
2. À l'assaut du monde professionnel	36
3. Critères de sélection	36
3.1 Tension logique	37
3.2 Fréquence CPU	38
3.3 Mémoire RAM	39
3.3.1 Durant la compilation des scripts	39
3.3.2 Durant le fonctionnement du script	39
3.4 Mémoire flash	40
3.5 Calcul en virgule flottante (FPU)	40
3.6 Communication	41
3.7 Communauté	41
3.8 Support matériel de MicroPython	41
4. Vue d'ensemble des plateformes disponibles	42
4.1 STMicroelectronics (Pyboard et Pyboard-D)	

4.2 STMicroelectronics (NaDHAT PYB405)	43
4.3 Microchip/Atmel (plateformes Adafruit)	52
4.4 Nordic Semiconductor (Micro:bit)	54
4.5 Espressif (ESP8266 et ESP32)	56
4.5.1 ESP8266	57
4.5.2 ESP32	57
4.6 Pycom.io	60
4.7 STMicroelectronics (carte de développement)	61
4.8 Autres plateformes	62
5. MicroPython et CircuitPython	63
6. Pyboard : extensions et communications	66
6.1 Boîtier officiel	66
6.2 LCD Skin	67
6.3 Audio Skin	68
6.4 Support réseau	69
6.5 Support WiFi	70
6.6 Support Bluetooth	71
6.6.1 Module Bluetooth série HC-05	71
6.6.2 Module Bluefruit LE UART Friend	72

7. Pyboard-D : extensions et communications	73
7.1 LCD Skin	73
7.2 Support Ethernet	74
7.3 Support WiFi et Bluetooth	74
7.4 Cartes breakout	74

MicroPython Pyboard

1. Introduction	77
2. Présentation de la Pyboard	78
2.1 MicroPython	78
2.2 MicroPython Pyboard	79
2.2.1 En cas de problème	79
2.2.2 Premier survol de la Pyboard	80
2.2.3 Examen physique	87
2.3 Système de fichiers MicroPython	88
2.4 Quels connecteurs pour la Pyboard ?	90
2.5 Pyboard et Fritzing	92
3. La Pyboard en détail	94
3.1 Interfaces matérielles	

3.1.1 LED	94
3.1.2 Bouton utilisateur	95
3.1.3 Bouton Reset	97
3.1.4 Accéléromètre	98
3.1.5 Carte microSD	99
3.1.6 Alimentation de la Pyboard	101
4. Pyboard : tension logique et courant	102
4.1 Niveau logique et tensions	105
4.2 Niveau logique et Python	105
4.3 Tolérance 5 V	105
4.4 Courants maximum, source et sink	105
4.5 Injection de courant	107
5. Les fonctions alternatives sur la Pyboard	107
5.1 Sortie PWM	107
5.2 Entrée analogique (ADC)	110
5.3 Sortie analogique (DAC)	112
5.4 Bus I2C	113
5.5 Bus SPI	114
5.6 Bus CAN	116

5.7 UART (port série)	117
6. Brochage de la Pyboard	121
6.1 Partie droite	123
6.2 Partie gauche	124
6.3 Partie basse	125
7. Brochage avancé et timers	126
7.1 Broches du STM32	128
7.2 Timers	129
8. Comment détruire sa Pyboard en sept leçons ?	133
8.1 Placer une broche directement à la masse	133
8.2 Brancher des GPIO ensemble	134
8.3 Appliquer une surtension sur une broche d'entrée	135
8.4 Appliquer une tension d'alimentation inversée sur V+	136
8.5 Appliquer une tension supérieure à 3,3 V sur la broche 3V3	138
8.6 Dépasser le courant max d'une broche	140
8.7 Dépasser le courant max du microcontrôleur	142

MicroPython Pyboard-D

1. Introduction	143
2. MicroPython Pyboard-D	144
2.1 Premier survol de la Pyboard-D	145
2.2 Quels connecteurs pour la Pyboard-D ?	152
2.3 Pyboard-D et Fritzing	156
2.4 Pyboard-D en détail	156
2.4.1 Interfaces matérielles	156
2.4.2 LED RGB	157
2.4.3 Bouton utilisateur	159
2.4.4 Carte microSD	161
2.4.5 Stockage eMMC	162
2.4.6 Horloge temps réel (RTC)	165
2.4.7 Alimentation de la Pyboard-D	167
2.4.8 Contrôle WiFi	169
3. Brochage de la Pyboard-D	171
3.1 Position X	171
3.2 Bus I2C en position X	171
3.3 Position Y	172
4. Carte d'interface WBUS-DIP28	172

Environnement de travail

1. Avant-propos	173
2. Manipulation de fichiers	174
3. Éditeur de texte	175
3.1 Atom (multiplateforme)	176
3.2 Windows	185
3.3 Linux	186
3.4 Raspbian Linux	187
3.5 Mac OS	188
4. Console série et REPL	188
4.1 PuTTY (multiplateforme)	189
4.2 Picocom (Linux)	191
4.3 Screen (Mac, Linux)	193
5. Outils intégrés	193
5.1 RShell	193
5.1.1 Linux et Raspbian	194

5.1.2 Windows	195
5.1.3 Mac OS	196
5.2 Mu Editor	196
5.2.1 Installation Linux	196
5.2.2 Installation Raspbian	198
5.2.3 Installation Windows et Mac	199
Prise de contrôle	
1. Installer une carte MicroPython	201
1.1 Sous Windows	201
1.2 Sous Linux	207
1.3 Sous Raspbian (Raspberry Pi)	212
2. Communiquer avec MicroPython	215
3. Utiliser le périphérique de stockage	216
3.1 Effacer des fichiers	218
3.2 Un seul espace de stockage accessible en USB	219
4. REPL : l'invite en ligne de commande	220
4.1 Séquence de contrôle REPL	221

4.2 Options avancées sur REPL	222
4.2.1 Édition de ligne	222
4.2.2 Historique de commandes	222
4.2.3 Autocomplétion	222
4.2.4 Variable « _ »	224
4.3 Outils Python avancés pour REPL	224
4.3.1 Fonction help()	225
4.3.2 Fonction dir()	226
4.3.3 Fonction listdir()	227
4.3.4 Afficher le contenu d'un fichier	229
4.4 Développer avec REPL	230
5. REPL via Bluetooth	232
5.1 À propos de Bluetooth	233
5.2 Module Bluetooth série	233
5.2.1 Brancher le module sur la Pyboard	235
5.2.2 Répliquer REPL sur le port série	235
5.2.3 Appairage sur PC Linux	236
5.2.4 Appairage avec un Smartphone	241
6. RShell	243
6.1 Ligne de commande RShell	

6.2 REPL sous RShell	244
6.3 Développer avec RShell	247
	250
7. upy-shell	251
8. WebREPL	252
8.1 Activer le démon WebREPL	254
8.2 Client WebREPL	254
8.2.1 WebREPL - Client HTML	255
8.2.2 WebREPL - Client Python	257
	258
9. Support WiFi sur la Pyboard-D	258
9.1 WiFi et réglementation locale	258
9.2 Nom d'hôte et adresse MAC	259
	259
10. Mode station (STA)	260
10.1 Mode STA et scan réseau	261
10.2 Réseau WiFi visible ou masqué	263
10.3 Connexion en mode STA	263
10.4 Contrôle WiFi avancé	264
10.5 Utiliser un socket	264
10.6 Rechercher l'adresse IP d'une Pyboard-D	266

267

11. Mode point d'accès (AP)

268

Séquence de démarrage

1. Séquence de démarrage MicroPython

273

1.1 Séquence de démarrage - Pyboard originale

274

1.2 Séquence de démarrage - Pyboard-D

276

2. Fichier boot.py

276

2.1 Mode de l'interface USB

277

2.2 Connexion WiFi avec boot.py

277

2.3 Activer WebREPL avec boot.py

279

2.4 Script utilisateur à exécuter

280

3. Fichier main.py

281

4. Séquence de démarrage en lumière

282

5. Safe Mode de la Pyboard

286

6. Séquence de démarrage et Pyboard-D

287

Programmer

1. Préambule	289
2. Les bibliothèques MicroPython	290
2.1 Le préfixe u	291
2.2 Bibliothèques dans le firmware	291
2.3 Mécanisme de chargement	292
2.4 Où placer les bibliothèques ?	293
2.5 Écrire ses propres bibliothèques	295
2.5.1 Script de test	295
2.5.2 Création des bus	299
3. Bibliothèques disponibles	299
3.1 Bibliothèques standards et micropythonifiées	299
3.2 Bibliothèques propres à MicroPython	302
3.3 Bibliothèques spécifiques à la carte de développement	302
3.3.1 Bibliothèque pyb	303
3.3.2 Bibliothèque lcd160cr	303
4. Bibliothèque machine	304
4.1 Limitation de la portabilité	305

4.2 Quel intérêt pour la portabilité ?	305
4.3 Contenu de la bibliothèque machine	306
5. Bibliothèque pyb	309
6. Bibliothèque os	310
7. Charger et exécuter un script à la volée	313
8. Exploiter l'accéléromètre	316
9. Entrées/sorties	321
9.1 Entrée numérique	322
9.2 Entrée numérique (pull-up interne)	326
9.3 Entrée numérique et déparasitage	329
9.4 Entrée numérique et interruption	331
9.5 Sortie numérique	334
9.5.1 Commander une LED	334
9.5.2 Broche en sortie et courant de court-circuit	337
9.5.3 Montage drain ou source	338
9.6 Entrée analogique	342
9.6.1 Potentiomètre	343
9.6.2 Lecture analogique	346

9.6.3	Précision des convertisseurs	347
9.6.4	Échantillonnage	349
9.6.5	Acquisition de signal et fréquence d'échantillonnage	352
9.6.6	Autres fonctionnalités	356
9.7	Sortie analogique	356
9.7.1	DAC en résolution 8 bits	356
9.7.2	DAC en résolution 12 bits	357
9.7.3	Reproduire un échantillon	358
9.7.4	Sortie analogique au-delà de 3,3 V	363
9.7.5	DAC et système audio	364
9.7.6	Plus d'information sur le DAC	365
9.8	Sortie PWM	366
9.8.1	Commander l'intensité d'une LED	371
9.8.2	Commande de vitesse de moteur	375
9.9	Sortie Servo	380
9.9.1	Synchroniser des servomoteurs	381
9.9.2	Servomoteurs à rotation continue	383
9.9.3	Alimentation des servomoteurs	384
9.9.4	Calibration des servomoteurs	386
10.	Identification et mode des broches	388

11. Les timers	389
11.1 Timers disponibles	391
11.2 Fonction de rappel/d'interruption	391
11.3 Fonctions d'interruption et bonnes pratiques	393
11.4 Timers avancés	395
11.5 Le timer WatchDog	397
12. Bus I2C	398
12.1 I2C : comment ça marche ?	401
12.1.1 À propos des bits d'adresse	405
12.1.2 I2C comme un périphérique mémoire	406
12.2 Capteur I2C et pilote MicroPython	410
12.2.1 Où trouver des pilotes ?	410
12.2.2 Comment utiliser un pilote I2C	410
12.2.3 Faut-il créer le bus I2C hors du pilote ?	413
12.3 Connecteurs standardisés pour I2C	413
12.3.1 Connecteur UEXT d'Olimex	414
12.3.2 Connecteur NCD de National Control Devices	416
12.3.3 Connecteur Qwiic de SparkFun	420
12.3.4 Écosystème Feather d'Adafruit Industries	423
12.4 Communication I2C par l'exemple	427

12.4.1 Exemple 1 : accéléromètre de la Pyboard	427
12.4.2 Exemple 2 : ADS1115 (entrées analogiques supplémentaires)	431
12.4.3 Exemple 3 : ajouter une carte MOD-IO	434
12.5 Le bus I2C plus en détail	437
12.5.1 Résistances pull-up et tension logique	438
12.5.2 Vitesse limitée	442
12.5.3 Effet capacitif et longueur de ligne	443
12.6 Augmenter la longueur du bus I2C	446
12.6.1 P82B715PN	446
12.6.2 Carte d'extension différentielle de SJTbits	448
13. Bus SPI	449
13.1 Utilisation de l'API SPI	453
13.2 Classe SPI du module pyb	453
13.3 Communication SPI par l'exemple	454
13.3.1 Exemple 1 : MOD-VGA (Gameduino sous MicroPython)	454
13.3.2 Exemple 2 : matrice LED	460
14. Interface UART	463
14.1 La trame série	464
14.2 Configurer une ligne série	465
14.3 Émission/réception	

14.4 Communication UART par l'exemple	465
14.4.1 Exemple 1 : module GPS	466
14.4.2 Exemple 2 : module ESP8266	466
	471
15. Horloge RTC	476
16. Poursuivre l'exploration	479
16.1 Interface CAN	479
16.2 Interface DMX	479
	479
 Capteurs et interfaces	
1. Introduction	481
2. Signal numérique	483
2.1 Module relais	483
2.1.1 Mise en garde	483
2.1.2 À réaliser soi-même	484
2.1.3 Modules relais préassemblés	484
2.1.4 Plusieurs modules relais	484
2.1.5 Relais et impulsions électromagnétiques (EMI)	485
2.1.6 PowerSwitchTail	486

2.2 Contact magnétique (interrupteur)	486
2.3 Capteur à effet Hall	487
2.4 Capteur PIR	489
2.5 Encodeur rotatif	491
2.6 LED RGB	493
2.7 Capteur ultrason HC-SR04	496
2.7.1 Brancher le HC-SR04	503
2.7.2 Tester le HC-SR04	504
2.7.3 Bibliothèque	505
3. Signal analogique	507
3.1 Capteur de température TMP36	508
3.2 Photorésistance	510
3.3 Capteur de pression différentielle MPXV5010DP	512
3.4 Mesure de courant avec ACS712	517
3.4.1 Mise en garde	518
3.4.2 Charge alternative et relevé alternatif	518
3.4.3 Mesure avec la Pyboard	521
3.5 Sortie analogique	526
4. Interface I2C	527

4.1 MCP23017 : extension d'entrée/sortie	527
4.1.1 Brancher le MCP23017	528
4.1.2 Bibliothèque	532
4.1.3 Tester le MCP23017	532
4.2 ADS1115 : entrée analogique	534
4.2.1 Brancher l' ADS1115	537
4.2.2 Mille milliards de mille parasites !	537
4.2.3 Le gain programmable	539
4.2.4 Bibliothèque	540
4.2.5 Tester l'ADS1115	541
4.3 DHT11 / DHT22 : humidité et température	543
4.3.1 Brancher le DHT 11	544
4.3.2 Tester le DHT11	545
4.4 BMP280 / BME280 : humidité, pression et température	546
4.4.1 Brancher le BME280	547
4.4.2 Bibliothèque	548
4.4.3 Tester le BME280	548
4.4.4 Calcul d'altitude	549
4.4.5 Modes de fonctionnement du BME280	549
4.5 TSL2561 / TSL2591 : luminosité	550
4.5.1 Brancher le TSL2561	552

4.5.2 Bibliothèque	552
4.5.3 Tester le TSL2561	553
4.5.4 Tester le TSL2591	553
4.6 MOD-IO : relais et entrées optocoupleurs (24 V)	555
4.7 Contrôleur Wii Nunchuck UEXT	556
4.7.1 Brancher le Nunchuk	556
4.7.2 Bibliothèque	557
4.7.3 Tester	557
4.8 MCP4725 : sortie analogique	559
4.8.1 Brancher le MCP4725	560
4.8.2 Bibliothèque	561
4.8.3 Tester le MCP4725	561
4.8.4 Une onde en dos de chameau	563
4.9 MCP9808 : capteur de température de précision	571
4.9.1 Brancher le MCP9808	572
4.9.2 Bibliothèque	572
4.9.3 Tester le MCP9808	572
5. Contrôle moteur	574
5.1 Servomoteurs sur la Pyboard	577
5.2 L298 : pont-H pour moteur continu	578

5.3 L293D : pont-H pour moteur continu	587
5.4 DRV8833 : contrôleur moteur continu	587
5.5 PCA9685 : contrôleur PWM et servo moteur	594
5.6 A4988 : contrôleur moteur pas-à-pas	597
5.6.1 Découvrir le moteur pas-à-pas	597
5.6.2 Contrôleur de moteur pas-à-pas A4988	599
5.6.3 Tester le A4988	602
6. Afficheurs	606
6.1 LED NeoPixel	606
6.2 SD1306 : afficheur OLED	615
6.2.1 Brancher l'afficheur OLED	616
6.2.2 Installer la bibliothèque	617
6.2.3 Tester l'afficheur OLED	618
6.2.4 Manipulations du FrameBuffer	620
6.2.5 Image PBM	625
6.2.6 Ressources	628
6.3 LCD160CR : afficheur tactile	629
6.3.1 Couleurs en 16 et 24 bits	630
6.3.2 Tester le LCD160CR	632
6.3.3 Autres ressources	641

6.4 MOD-LCD : afficheur à cristaux liquides	642
6.5 Encore plus d'afficheurs	644
6.5.1 Afficheur LCD 2 ou 4 lignes USB/série	645
6.5.2 ILI9341 : afficheur TFT couleur	648
7. Interface UART	650
7.1 Module GPS	650
7.1.1 Brancher le module GPS	651
7.1.2 Bibliothèque	652
7.1.3 Tester le module GPS	653
7.2 Lecteur de carte RFID	655
7.2.1 Brancher le module RFID	658
7.2.2 Bibliothèque	659
7.2.3 Tester le module RFID	659
8. Interface USB HID	661
8.1 Configurer l'interface USB	662
8.2 Fonction <code>usb_mode()</code>	662
8.3 Émuler la souris	664
8.4 Émuler le clavier	666
8.4.1 Activer le mode HID pour un clavier	667
8.4.2 Envoyer une pression de touche	667

8.4.3 Bibliothèque ushid	667
8.4.4 Créer des touches copier/coller	669
	671
9. Interface réseau	676
9.1 Module Ethernet	676
9.2 Module WiFi	678
9.2.1 ESP8266 en mode AT	678
9.2.2 Pyboard-D	682
9.3 Module Bluetooth HC-05	682
9.4 Support Bluetooth 4.0	682

I2C : petit manuel du développeur

1. Préambule	683
2. Introduction	683
3. Manipulation de données	686
3.1 Octet et représentation binaire	686
3.2 Représentation binaire et hexadécimale	687
3.3 Manipulation de bits	688
3.3.1 Activer un bit	

3.3.2 Désactiver un bit	688
3.3.3 Tester un bit	689
3.3.4 Décalage à gauche	689
3.3.5 Décalage à droite	690
3.4 Bytes et bytearray	691
3.5 Petit boutisme (little endian) vers entier 16 bits	691
3.5.1 Dépasser la taille de l'octet (8 bits)	694
3.5.2 Petit boutisme et grand boutisme	694
3.6 Le complément à deux	695
3.7 struct à la rescousse	698
4. Bus I2C	699
4.1 Niveau d'API	701
4.2 Adresses I2C	702
4.3 Scan du bus	703
4.4 Conditions / Start / Stop	703
5. Rétroportage CircuitPython (TSL2591)	704
5.1 Localiser le pilote CircuitPython	705
5.2 Rétroportage étape par étape	705
5.2.1 Mise en place	705

5.2.2 Déroulement du portage	708
5.2.3 Correction des imports et commentaires	708
5.2.4 Correction 1 : I2CDevice -> machine.I2C	709
5.2.5 Correction 2 : _read_u8()	709
5.2.6 Correction 3 : _write_u8()	710
5.2.7 Correction 4 : _read_u16LE()	711
5.3 Et ensuite ?	712
6. Poursuivre l'exploration I2C	712
Classes MicroPython courantes	
1. Les classes MicroPython en français	713
2. La classe ADC	713
2.1 Constructeur	714
2.2 Méthodes	714
3. La classe ADCAII	716
3.1 Constructeur	717
4. La classe DAC	718

4.1 Constructeur	718
4.2 Méthodes	719
5. La classe I2C	722
5.1 Constructeur	723
5.2 Méthodes	724
5.3 Méthodes - opérations primitives	724
5.4 Méthodes - opérations standards	725
5.5 Méthodes - opérations mémoire	726
6. La classe LCD160CR	727
6.1 Constructeur	728
6.2 Méthode statique	729
6.3 Propriétés	729
6.4 Méthodes	730
6.4.1 Configuration de l'écran	730
6.4.2 Manipulation graphique de l'écran	731
6.4.3 Afficher du texte	732
6.4.4 Dessin de primitive	733
6.4.5 Méthodes tactiles	734
6.4.6 Méthodes avancées	735

7. La classe Pin	737
7.1 Constructeur	737
7.2 Méthodes	739
8. La classe RTC	741
8.1 Constructeur	742
8.2 Méthodes	742
9. La classe Servo	743
9.1 Constructeur	744
9.2 Méthodes	744
10. La classe Signal	745
10.1 Constructeur	747
10.2 Méthodes	747
11. La classe SPI	748
11.1 Constructeur	749
11.2 Méthodes	749
12. La classe Timer	750
12.1 Constructeur	751

12.2 Méthodes	751
13. Classe TimerChannel	754
13.1 Méthodes	755
14. Classe UART	755
14.1 Constructeur	757
14.2 Méthodes	757
Annexes	
1. À propos des annexes	761
2. Mise à jour du firmware	761
2.1 Pyboard originale	761
2.1.1 Installer dfu-util	761
2.1.2 Télécharger le nouveau firmware	762
2.1.3 Placer la carte en mode DFU	762
2.1.4 Tester le mode DFU	763
2.1.5 Flasher le nouveau firmware	764
2.2 Pyboard-D	764
3. Safe Mode et Reset Factory	

	766
3.1 Safe Mode	766
3.2 Factory Reset	767
4. Conversion des logiques 3,3 V et 5 V	767
4.1 Logiques TTL 5 V et CMOS 3,3 V	767
4.2 Sortie 3,3 V vers entrée 5 V	767
4.3 Sortie 5 V vers entrée 3,3 V	767
4.4 Conversion bidirectionnelle 5 V - 3,3 V	767
4.5 Niveau logique et bus I2C	767
4.6 Conversion et haut débit	768
5. PWM vers analogique	768
6. Plateformes MicroPython disponibles	768
6.1 STMicroelectronics (Pyboard et Pyboard D)	768
6.2 Microchip (cartes CircuitPython ATSAMD)	768
6.3 Espressif (cartes ESP8266)	768
6.4 Espressif (cartes ESP32)	768
6.5 Espressif (console Odroid-GO (ESP32))	768
6.6 STMicroelectronics (Espruino Pico)	768
6.7 Pycom (WiPy, LoPy, etc.)	768

6.8 Nordic Semiconductor (cartes Micro:bit et autres)	769
6.9 STMicroelectronics (cartes de développement)	769
6.10 NXP (Teensy)	769
7. Schéma de la Pyboard	769
Index	773