

La carte Raspberry Pi

K. Boudjelaba

BTS CIEL – IR, 1ère année



Introduction

Système embarqué

Introduction aux architectures x86 et ARM

À savoir sur le Raspberry Pi5

Caractéristiques du Raspberry Pi 3 – B+

Installation du système d'exploitation

Précautions et bonnes pratiques

Les ports GPIO

Les bibliothèques

Connexion en SSH au Raspberry Pi pour le contrôler depuis un ordinateur

Connexion à distance via VNC (Virtual Network Computing)

Annexes

Système embarqué

Un **système embarqué** est un système informatique autonome, combinant du matériel et du logiciel conçus conjointement pour remplir une ou plusieurs fonctions spécifiques. Il est généralement intégré dans un appareil plus large (véhicule, électroménager, équipement médical, etc.).

Ces systèmes fonctionnent souvent en **temps réel** : les opérations doivent être réalisées dans un délai contraint en réponse à un événement externe (interruption matérielle, signal capteur, etc.).

Les systèmes embarqués sous Linux sont largement utilisés dans les technologies modernes : transport, multimédia, téléphonie mobile, domotique, robotique, etc.

Introduction aux architectures x86 et ARM

Dans le monde de l'informatique, différentes architectures matérielles coexistent, chacune adaptée à des usages spécifiques. Deux des plus courantes sont l'architecture **x86**, utilisée principalement dans les ordinateurs personnels et serveurs, et l'architecture **ARM**, omniprésente dans les systèmes mobiles et embarqués.

L'architecture **x86**, développée initialement par Intel, est conçue pour offrir une puissance de calcul élevée, une compatibilité logicielle étendue, et un large éventail de fonctionnalités. C'est le cœur des PC traditionnels, mais elle consomme généralement plus d'énergie et est plus complexe. À l'inverse, l'architecture **ARM** (Qualcomm, Apple Silicon, Samsung) privilégie la faible consommation d'énergie, la simplicité, et une taille compacte. C'est ce qui la rend idéale pour les **systèmes embarqués** : des dispositifs intégrés dans des objets du quotidien (domotique, robotique, automobiles, appareils médicaux), smartphones, tablettes, qui doivent fonctionner de manière autonome, souvent sur batterie, avec des contraintes matérielles strictes.

Le **Raspberry Pi** utilise un processeur ARM, idéal pour l'apprentissage des systèmes embarqués, grâce à ses ports GPIO (General Purpose Input/Output) qui permettent d'interagir avec des capteurs et actionneurs.

Dans ce cours, nous verrons donc comment programmer ces interfaces matérielles, en utilisant des bibliothèques adaptées à l'architecture ARM et à la nature spécifique du Raspberry Pi.

À savoir sur le Raspberry Pi5

Le Raspberry Pi 5 apporte plusieurs améliorations matérielles importantes par rapport aux modèles précédents comme le Raspberry Pi 3 B+ :

- ▶ **Processeur plus puissant** : un CPU ARM plus rapide, offrant de meilleures performances pour les applications embarquées.
- ▶ **Mémoire améliorée** : plus de RAM disponible, permettant de gérer des projets plus gourmands.
- ▶ **Ports et connectivité** : ports USB-C pour l'alimentation, USB 3.0 natifs, meilleure gestion des périphériques externes.
- ▶ **Compatibilité logicielle** : le système d'exploitation Raspberry Pi OS et les bibliothèques GPIO (pigpio, gpiozero) fonctionnent toujours de manière compatible, sans modifications majeures.

En résumé, les exercices et codes réalisés sur un Raspberry Pi 3 B+ sont directement applicables sur un Raspberry Pi 5. Les différences matérielles améliorent la fluidité et les possibilités, mais ne changent pas les bases de la programmation GPIO ou l'utilisation des bibliothèques.

Raspberry Pi

Le **Raspberry Pi** est un nano-ordinateur monocarte équipé d'un processeur ARM. Il permet d'exécuter différentes variantes du système d'exploitation libre GNU/Linux — principalement **Raspberry Pi OS**, basé sur Debian — installé sur une carte microSD.

Il est conçu à la fois pour l'éducation à l'informatique, les projets d'électronique, la robotique, ou encore les systèmes embarqués.

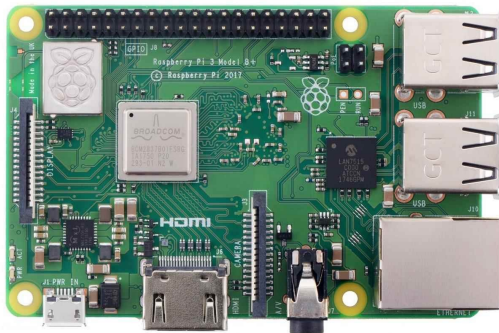


Figure 1: Carte Raspberry Pi 3 - Modèle B+

Attention : il s'agit de matériel relativement fragile, à manipuler avec précaution.

Le Raspberry Pi n'est pas une carte Arduino : c'est un véritable ordinateur fonctionnant sous un système d'exploitation.

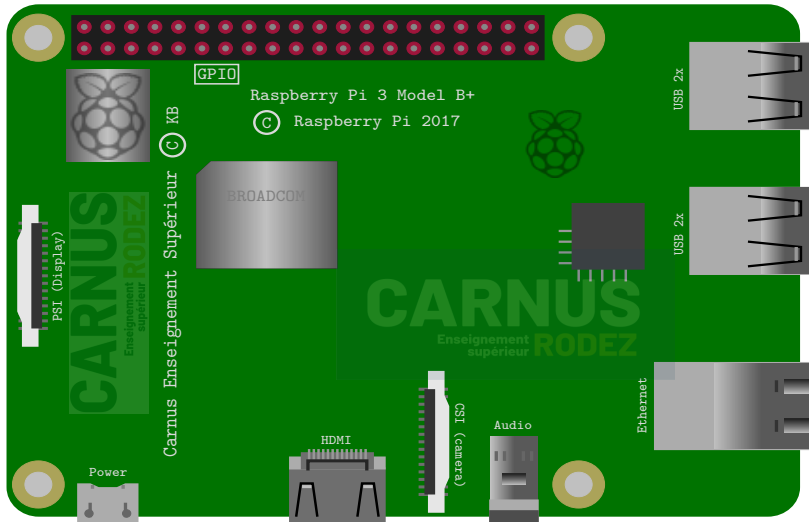


Figure 2: Carte Raspberry Pi 3 - Modèle B+ : Les différents ports E/S

Connectivité

- ▶ **Sans fil** : Wi-Fi Dual Band (802.11 b/g/n/ac) et Bluetooth 4.2 BLE pour une connectivité rapide et à faible latence.
- ▶ **Filaire** : Port Ethernet Gigabit (limité à 300 Mbps en pratique via USB 2.0).

Caractéristiques techniques

- ▶ Processeur : ARM Cortex-A53 (64-bit, quad-core, 1.4 GHz)
- ▶ Mémoire RAM : 1 Go LPDDR2
- ▶ Stockage : lecteur de carte micro-SD
- ▶ GPIO : connecteur 40 broches
- ▶ Affichage : port HDMI + port DSI (écran tactile officiel)
- ▶ Caméra : port CSI
- ▶ Ports USB : 4 × USB 2.0
- ▶ Audio : prise jack 3.5 mm (audio + vidéo composite)
- ▶ Réseau : Wi-Fi, Bluetooth, Ethernet
- ▶ Alimentation : 5V / 2.5A (connecteur micro-USB)

Installation du système d'exploitation

Raspberry Pi OS (anciennement Raspbian)

Raspberry Pi OS est un système d'exploitation libre basé sur Debian GNU/Linux, optimisé pour les cartes Raspberry Pi.

Il ne s'agit pas simplement d'un système minimal : il est livré avec plus de 35 000 paquets (logiciels précompilés) faciles à installer via les gestionnaires de paquets comme apt.

En tant que dérivé de Debian, Raspberry Pi OS bénéficie de la riche documentation et de la stabilité de cette distribution.

Pré-requis

- ▶ Carte microSD (minimum 8 Go, formatée en FAT32)
- ▶ Ordinateur avec lecteur de carte SD ou adaptateur USB
- ▶ Connexion Internet

Téléchargement de l'outil Raspberry Pi Imager

- ▶ Télécharger le logiciel officiel (Raspberry Pi Imager pour faciliter l'installation du Raspberry Pi OS sur le Raspberry Pi) :
<https://www.raspberrypi.com/software>
- ▶ Installer Raspberry Pi Imager sur l'ordinateur (Windows, macOS ou Linux)

Installation de l'OS sur la carte microSD

1. Insérer la carte SD dans le lecteur de l'ordinateur
2. Lancer **Raspberry Pi Imager**
3. Cliquer sur **“Choose OS”** → Sélectionner **Raspberry Pi OS (32-bit)** recommandé
4. Cliquer sur **“Choose Storage”** → Sélectionner la carte SD
5. (Optionnel) Cliquer sur la roue dentée pour préconfigurer :
 - ▶ Activer SSH
 - ▶ Choisir le nom d'utilisateur et mot de passe (**à ne pas modifier**)
 - ▶ Configurer le Wi-Fi
 - ▶ Définir la langue et le fuseau horaire
6. Cliquer sur **“Write”** → L'écriture de l'image commence

Une barre de progression indique l'avancement. À la fin, retirer la carte SD en toute sécurité.

Premier démarrage

- ▶ Insérer la carte microSD dans le Raspberry Pi
- ▶ Connecter un écran HDMI, un clavier et une souris
- ▶ Brancher l'alimentation (5V / 2.5A)
- ▶ Le système démarre automatiquement
- ▶ À la première utilisation :
 - ▶ Sélectionner la langue, le fuseau horaire et le clavier
 - ▶ Se connecter au Wi-Fi si nécessaire
 - ▶ Effectuer les mises à jour proposées

Tutoriel vidéo

Un tutoriel complet est disponible ici :

<https://www.youtube.com/watch?v=ntaXWS8Lk34>

Important : Précautions et bonnes pratiques

Avant de démarrer

1. Connecter clavier, souris et écran AVANT d'alimenter le Raspberry Pi.
2. Utiliser une alimentation fiable (5V / 2.5A minimum).
3. Ne jamais insérer ou retirer la carte SD lorsque le Raspberry Pi est alimenté.

Pendant l'utilisation

- ▶ Manipuler le Raspberry Pi par les bords, éviter tout contact avec les composants.
- ▶ Éviter l'électricité statique (décharger ses mains sur un objet métallique avant manipulation).
- ▶ Ne pas poser la carte sur une surface conductrice.

Éteindre correctement le Raspberry Pi

- ▶ Ne jamais débrancher brutalement l'alimentation.
- ▶ Pour éteindre proprement :
 - ▶ Via l'interface graphique : Menu Framboise → Déconnexion → Arrêter
 - ▶ Ou via le terminal : `sudo shutdown -h now`
- ▶ Attendre que la LED rouge s'éteigne avant de débrancher.

Conseil

Utiliser un boîtier de protection pour éviter les courts-circuits accidentels.

Précautions et bonnes pratiques

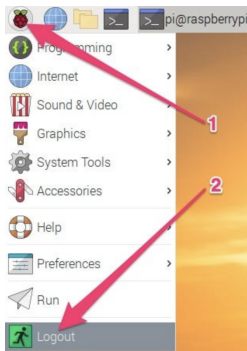


Figure 3: Étapes pour éteindre proprement le Raspberry Pi

Les ports GPIO

Définition

Les GPIO (General Purpose Input/Output) sont des broches d'entrée/sortie programmables sur le Raspberry Pi.

Elles permettent de :

- ▶ Lire l'état d'un capteur (entrée)
- ▶ Commander un composant (LED, relais, moteur...) (sortie)
- ▶ Communiquer avec d'autres modules via des protocoles série (I2C, SPI, UART)

Les ports GPIO

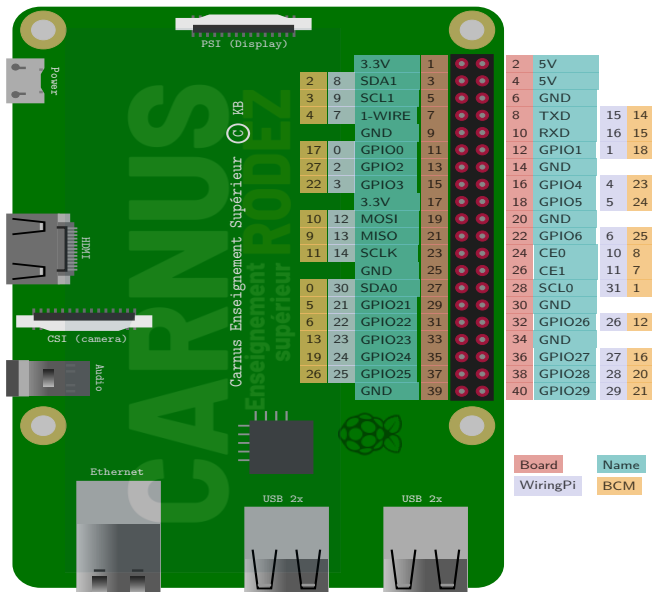


Figure 4: Disposition physique des ports GPIO (vue BOARD)

Description des broches

La carte Raspberry Pi 3 modèle B+ dispose de 40 broches GPIO, regroupées comme suit :

- ▶ **GPIO** : Broches numériques générales (entrées/sorties)
- ▶ **GND** : Masse électrique
- ▶ **Power (3.3V / 5V)** : Alimentation pour composants externes
- ▶ **I2C** : Bus de communication série (pour capteurs, horloges, etc.)
- ▶ **SPI** : Bus série rapide pour cartes SD, écrans, etc.
- ▶ **UART** : Communication série classique

Deux modes de numérotation

Il existe deux manières de numéroter les GPIO :

- ▶ **Numérotation BOARD** : basée sur l'emplacement physique des broches (1 à 40)
- ▶ **Numérotation BCM** : basée sur la désignation interne du processeur (numérotation logique selon le SoC Broadcom)

Important : Le choix du mode dépend de la bibliothèque utilisée (Python, C/C++, etc.)

Exemple en Python (numérotation BOARD)

```
GPIO.setmode(GPIO.BOARD)
```

Les ports GPIO

BCM (Broadcom SOC Channel)	WiringPi	Name	Physical (BOARD)		Name	WiringPi	BCM (Broadcom SOC Channel)
		3.3V	1	2	5V		
2	8	SDA.1	3	4	5V		
3	9	SCL.1	5	6	GND		
4	7	1-WIRE	7	8	TXD	15	14
		GND	9	10	RXD	16	15
17	0	GPIO.0	11	12	GPIO.1	1	18
27	2	GPIO.2	13	14	GND		
22	3	GPIO.3	15	16	GPIO.4	4	23
		3.3V	17	18	GPIO.5	5	24
10	12	MOSI	19	20	GND		
9	13	MISO	21	22	GPIO.6	6	25
11	14	SCLK	23	24	CE0	10	8
		GND	25	26	CE1	11	7
0	30	SDA.0	27	28	SCL.0	31	1
5	21	GPIO.21	29	30	GND		
6	22	GPIO.22	31	32	GPIO.26	26	12
13	23	GPIO.23	33	34	GND		
19	24	GPIO.24	35	36	GPIO.27	27	16
26	25	GPIO.25	37	38	GPIO.28	28	20
		GND	39	40	GPIO.29	29	21

Figure 5: Correspondance des broches GPIO (BOARD vs BCM)

Les bibliothèques

Utilisation du langage C/C++ avec le Raspberry Pi

Le Raspberry Pi étant un véritable ordinateur, il est possible de l'utiliser avec divers langages, notamment le C/C++.

L'utilisation des broches GPIO en C/C++ peut être complexe, car elle nécessite d'interagir directement avec des fichiers système. Pour simplifier cela, des bibliothèques comme **WiringPi** et **pigpio** ont été développées.

Remarque

Python est souvent préféré pour sa simplicité, mais C/C++ offre plus de contrôle bas niveau, ce qui peut être utile dans les projets temps réel ou performants.

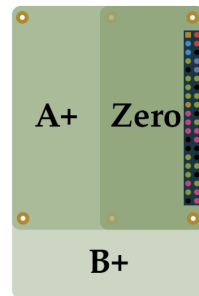
WiringPi

WiringPi est une bibliothèque inspirée d'Arduino, développée pour faciliter l'utilisation des ports GPIO sur Raspberry Pi avec une interface simple en C/C++. Très populaire pendant des années, elle a permis de nombreux projets et tutoriels. Cependant, son développement officiel a été arrêté en 2019, et elle n'est plus maintenue ni mise à jour. Cela peut poser des problèmes de compatibilité avec les dernières versions de Raspberry Pi OS ou avec certains matériels récents. Son usage est désormais déconseillé sur les versions récentes de Raspberry Pi OS.

Numérotation des broches

WiringPi utilise un schéma de numérotation spécifique qui ne correspond ni au mode BOARD, ni au mode BCM, ce qui peut créer de la confusion. Voir figure 6.

3v3 Power	1	●	●	2	5v Power
GPIO 2 (WiringPi 8)	3	●	●	4	5v Power
GPIO 3 (WiringPi 9)	5	●	●	6	Ground
GPIO 4 (WiringPi 7)	7	●	●	8	GPIO 14 (WiringPi 15)
Ground	9	●	●	10	GPIO 15 (WiringPi 16)
GPIO 17 (WiringPi 0)	11	●	●	12	GPIO 18 (WiringPi 1)
GPIO 27 (WiringPi 2)	13	●	●	14	Ground
GPIO 22 (WiringPi 3)	15	●	●	16	GPIO 23 (WiringPi 4)
3v3 Power	17	●	●	18	GPIO 24 (WiringPi 5)
GPIO 10 (WiringPi 12)	19	●	●	20	Ground
GPIO 9 (WiringPi 13)	21	●	●	22	GPIO 25 (WiringPi 6)
GPIO 11 (WiringPi 14)	23	●	●	24	GPIO 8 (WiringPi 10)
Ground	25	●	●	26	GPIO 7 (WiringPi 11)
GPIO 0 (WiringPi 30)	27	●	●	28	GPIO 1 (WiringPi 31)
GPIO 5 (WiringPi 21)	29	●	●	30	Ground
GPIO 6 (WiringPi 22)	31	●	●	32	GPIO 12 (WiringPi 26)
GPIO 13 (WiringPi 23)	33	●	●	34	Ground
GPIO 19 (WiringPi 24)	35	●	●	36	GPIO 16 (WiringPi 27)
GPIO 26 (WiringPi 25)	37	●	●	38	GPIO 20 (WiringPi 28)
Ground	39	●	●	40	GPIO 21 (WiringPi 29)



- GPIO (General Purpose IO)
- SPI (Serial Peripheral Interface)
- I²C (Inter-integrated Circuit)
- UART (Universal Asynchronous Receiver/Transmitter)
- PCM (Pulse Code Modulation)
- Ground
- 5v (Power)
- 3.3v (Power)

Figure 6: Numérotation des broches dans WiringPi

Attention

La bibliothèque `WiringPi` n'est plus maintenue. Elle peut ne pas fonctionner sur les systèmes récents ou les modèles récents de Raspberry Pi.

Alternative recommandée : `pigpio`

`pigpio` est une bibliothèque moderne et maintenue qui permet un contrôle précis des GPIO :

- ▶ Fonctionne en C, C++ et Python
- ▶ Permet le contrôle à distance des GPIO via réseau
- ▶ Déjà préinstallée sur Raspberry Pi OS (versions récentes)

pigpio et GPIO Zero

Aujourd'hui, il est recommandé d'utiliser des bibliothèques plus modernes et activement maintenues comme :

- ▶ **pigpio** : offre un contrôle précis et bas niveau des GPIO, avec une interface en C et Python, idéale pour les projets nécessitant un timing rigoureux.
- ▶ **GPIO Zero** : bibliothèque Python haut niveau, très pédagogique et simple à prendre en main, parfaite pour les débutants et les projets rapides.

Ces bibliothèques bénéficient d'une communauté active, de mises à jour régulières, et sont intégrées dans la documentation officielle.

Comparaison des bibliothèques pour Raspberry Pi

Fonctionnalité	wiringPi	pigpio	GPIO Zero
Langage principal	C	C / Python	Python
Maintenance	Arrêtée (2019)	Active	Active
Facilité d'utilisation	Moyenne	Avancée (bas niveau)	Très simple (haut niveau)
Support hardware	Bon (mais vieux)	Excellent	Bon
Documentation	Limitée / obsolète	Complète	Très pédagogique
Public cible	Utilisateurs avancés	Utilisateurs avancés	Débutants / éducatif

- ▶ Page officielle wiringPi (archivée) : <http://wiringpi.com/>
- ▶ pigpio : <http://abyz.me.uk/rpi/pigpio/>
- ▶ GPIO Zero : <https://gpiozero.readthedocs.io/en/stable/>

Connexion en SSH au Raspberry Pi

Connexion en SSH au Raspberry Pi pour le contrôler depuis un ordinateur

SSH

SSH (Secure SHell) est un protocole sécurisé permettant de se connecter à un autre ordinateur via une interface en ligne de commande. Il chiffre toutes les communications pour garantir la sécurité des échanges.

Utilité

SSH est très utile pour administrer un Raspberry Pi à distance, sans avoir besoin d'un écran, clavier ou souris. On peut ainsi :

- ▶ Lancer des commandes à distance
- ▶ Transférer des fichiers
- ▶ Accéder aux logs système

Connexion en SSH au Raspberry Pi pour le contrôler depuis un ordinateur

Pré-requis

Pour établir une connexion SSH, il faut :

- ▶ Un serveur SSH activé sur le Raspberry Pi
- ▶ Un client SSH installé sur l'ordinateur (ex. PuTTY sous Windows, ou terminal sous Linux/macOS)

Attention

Le Raspberry Pi doit être connecté au même réseau local (Wi-Fi ou Ethernet) que l'ordinateur pour établir une connexion SSH.

Connexion en SSH au Raspberry Pi pour le contrôler depuis un ordinateur

Méthode 1 : Activer SSH depuis le Raspberry Pi

Si on dispose d'un clavier et d'un écran branchés au Raspberry Pi :

1. Ouvrir un terminal
2. Taper la commande : `sudo raspi-config`
3. Aller dans Interfacing Options → SSH
4. Sélectionner Enable

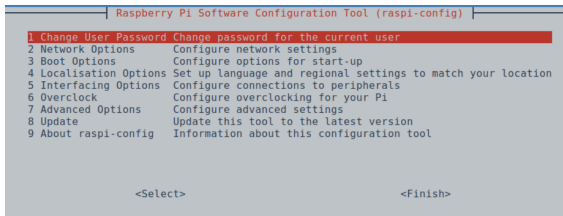


Figure 7: Activation de SSH via raspi-config

Connexion en SSH au Raspberry Pi pour le contrôler depuis un ordinateur

Méthode 2 : Activer SSH sans écran (via carte SD)

Si vous ne disposez pas d'écran/clavier sur le Raspberry Pi :

1. Insérer la carte SD dans l'ordinateur
2. Accéder à la partition boot
3. Créer un fichier nommé `ssh` (sans extension, vide)
4. Éjecter la carte SD et l'insérer dans le Raspberry Pi

Au prochain démarrage, SSH sera activé automatiquement.

Connexion en SSH au Raspberry Pi pour le contrôler depuis un ordinateur

Connexion depuis l'ordinateur

Sous Windows, installer et utiliser le logiciel **logiciel PuTTY** qui est un client SSH. :

- ▶ Entrer l'adresse IP du Raspberry Pi (ou `raspberrypi.local` si le réseau le permet)
- ▶ Laisser le port par défaut : 22
- ▶ Cliquer sur "Open"
- ▶ Saisir :
 - ▶ Login : `pi`
 - ▶ Mot de passe : `raspberrypi` (par défaut)

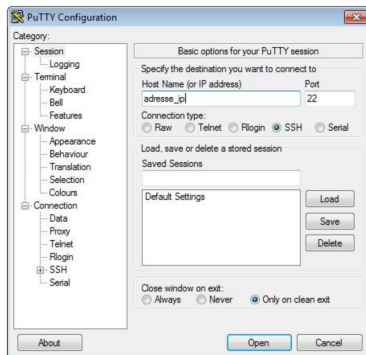


Figure 8: Connexion SSH avec PuTTY

Connexion en SSH au Raspberry Pi pour le contrôler depuis un ordinateur

Astuces utiles

- ▶ Pour connaître l'adresse IP du Raspberry, utiliser :
`ping raspberrypi.local` ou inspecter le routeur.
- ▶ Commande pour quitter la session SSH : `exit`
- ▶ Pour redémarrer ou arrêter le Raspberry à distance :
 - ▶ `sudo shutdown -h now` → arrêt immédiat
 - ▶ `sudo reboot` → redémarrage

Connexion à distance via VNC (Virtual Network Computing)

Connexion à distance via VNC (Virtual Network Computing)

VNC

VNC (Virtual Network Computing) permet de contrôler l'interface graphique d'un autre ordinateur à distance via une connexion réseau. Il s'agit d'un accès "bureau à distance" : on voit l'écran du Raspberry Pi sur l'ordinateur et on peut interagir comme si on était physiquement devant.

Avantages

- ▶ Contrôle complet de l'interface graphique sans clavier, écran ou souris connectés au Raspberry
- ▶ Pratique pour la configuration à distance
- ▶ Possibilité de transférer des fichiers entre PC et Raspberry

Connexion à distance via VNC (Virtual Network Computing)

Activer VNC sur le Raspberry Pi

1. Ouvrir un terminal sur le Raspberry
2. Lancer : `sudo raspi-config`
3. Aller dans : Interfacing Options → VNC
4. Sélectionner Yes pour activer VNC

Une fois activé, une icône VNC apparaît en haut à droite de l'écran (barre de tâches). En cliquant dessus, on accède :

- ▶ à l'adresse IP de la machine
- ▶ au statut de la connexion
- ▶ aux options de sécurité

Connexion à distance via VNC (Virtual Network Computing)

Installer le client VNC Viewer sur l'ordinateur

- ▶ Télécharger VNC Viewer sur le site officiel :
<https://www.realvnc.com/...>
- ▶ L'installer, puis l'ouvrir
- ▶ Entrer l'adresse IP du Raspberry Pi dans la barre de connexion
- ▶ Un écran de connexion s'affiche :
Saisir l'identifiant `pi` et le mot de passe (`raspberrypi` par défaut)

Remarque

Le Raspberry Pi et le PC doivent être sur le même réseau local.

Connexion à distance via VNC (Virtual Network Computing)

Transfert de fichiers avec VNC

Une fonctionnalité pratique de VNC Viewer est le transfert de fichiers dans les deux sens :

De l'ordinateur vers le Raspberry Pi :

- ▶ Cliquer sur l'icône VNC en haut à gauche dans la fenêtre de connexion
- ▶ Sélectionner Transférer des fichiers...
- ▶ Choisir les fichiers à envoyer

Du Raspberry Pi vers le PC :

- ▶ Ouvrir le menu du serveur VNC sur le Raspberry
- ▶ Cliquer sur Transfert de fichiers...
- ▶ Choisir les fichiers à rapatrier

Annexes

Sites officiels

- ▶ Site officiel du Raspberry Pi
- ▶ Téléchargement Raspberry Pi OS (Imager)
- ▶ Projets éducatifs Raspberry Pi

Bibliothèques et développement

- ▶ Documentation pigpio (GPIO C/Python)
- ▶ Bibliothèque GPIO Zero (Python)
- ▶ Site officiel de Python

Questions fréquentes

► Quel login par défaut ?

Nom d'utilisateur : `pi` | Mot de passe : `raspberrypi`

► Comment connaître l'adresse IP du Raspberry ?

`ping raspberrypi.local`

► Comment accéder au Raspberry sans écran ?

En activant SSH via un fichier `ssh` placé dans la partition boot

► Comment éviter de corrompre la carte SD ?

Toujours éteindre proprement le Raspberry Pi avant de le débrancher.

► Puis-je utiliser une carte SD de 32 Go ou plus ?

Oui, à condition de la formater en FAT32. Utilisez un outil comme SDFormatter.