

---

# Manuel d'installation

ProSE B1 2024 - CANvengers  
Passerelle Android-CAN vers banc CAN réel ou simulé

---

Responsable du document	Elisa DECLERCK
État du document	En réalisation
Version	1.0
Révision	0

## AVERTISSEMENT :

Le présent document a été réalisé sous la direction de l'équipe CANvengers. Ce document est la propriété de l'équipe CANvengers, du groupe ESEO. Toute utilisation, diffusion ou reproduction de ce document sans autorisation écrite préalable de l'équipe CANvengers est interdite. Nous tenons à souligner que toute violation de cette politique pourrait engager la responsabilité civile et pénale de son auteur. Nous vous demandons de prendre toutes les précautions nécessaires pour assurer la sécurité et la confidentialité de ce document.

Date	Actions	Auteur	Version	Révision
08/06/2023	Création du document et rédaction de la configuration de la Raspberry Pi	Elisa DECLERCK	0.0	0
09/06/2023	Révision mineure (nom fichiers, versionnage)	Paul TREMOUREUX	0.0	1
12/06/2023	Rédaction de l'installation des outils pour compiler le programme CANgateway	Elisa DECLERCK	0.0	2
12/06/2023	Rédaction de l'installation de IC-Sim	Elisa DECLERCK	0.0	3
12/06/2023	Rédaction de l'installation de JMeter	Thomas Rocher	0.0	4
13/06/2023	Rédaction de l'introduction	Elisa DECLERCK	0.0	4
13/06/2023	Rédaction de l'installation de Android Studio et configuration de l'appareil	Camille Lenne	0.0	4

TABLE 2 – Table des évolutions et validations internes du document

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Objet . . . . .	4
1.2	Portée . . . . .	4
1.3	Définitions, acronymes et abréviations . . . . .	4
1.4	Vue d'ensemble . . . . .	5
<b>2</b>	<b>Installation de l'application CANdoid</b>	<b>6</b>
2.1	Installation de Android Studio . . . . .	6
2.2	Configurer l'appareil . . . . .	6
<b>3</b>	<b>Installation de Robot Framework</b>	<b>7</b>
3.1	Installation de PyCharm . . . . .	7
3.2	Configuration de PyCharm . . . . .	7
<b>4</b>	<b>Installation du programme CANgateway</b>	<b>7</b>
4.1	Configuration de la Raspberry Pi . . . . .	7
4.1.1	Configuration du bus CAN . . . . .	7
4.1.2	Installation du Hotspot . . . . .	10
4.2	Installation des outils pour compiler le programme CANgateway . . . . .	11
4.2.1	Installation du compilateur . . . . .	11
4.2.2	Installation de CMocka . . . . .	11
4.2.3	Compilation du programme CANgateway . . . . .	12
<b>5</b>	<b>Installation de ICSim</b>	<b>13</b>
<b>6</b>	<b>Installation de JMeter</b>	<b>13</b>

# 1 Introduction

## 1.1 Objet

Ce manuel d'utilisation a pour objectif de fournir toutes les informations nécessaires à l'installation des outils pour faire fonctionner le logiciel "Passerelle Android-CAN vers banc CAN réel ou simulé".

## 1.2 Portée

Ce document a pour but de présenter des tutoriels d'installation du Système à l'Étude (SàE). Il est destiné à plusieurs parties prenantes :

- L'équipe de développement C et celle de développement Android, pour installer les outils.
- Le Client, pour lui permettre de continuer le développement du SàE sereinement.

## 1.3 Définitions, acronymes et abréviations

Les abréviations utilisées dans le présent document sont répertoriées et expliquées dans le tableau présenté ci-dessous.

Acronymes/Abréviations	Définitions
Client	Société KEREVAL, Numéro SIRET 44278921000030.
IHM (Interface Homme Machine)	Moyens permettant aux utilisateurs de l'application CANdroid d'interagir avec le programme CANgateway.
SàE (Système à l'Étude)	Ensemble composé de l'application Android, CANdroid, et du programme en C, CANgateway.
SSH ( <i>Secure Shell</i> )	Protocole de communication sécurisé.

## 1.4 Vue d'ensemble

Ce manuel d'installation est structuré en 5 parties :

- La première partie présente les objectifs et la portée de ce document.
- La deuxième partie concerne l'installation de l'application CANdroid.
- La troisième partie présente l'installation de PyCharm et Robot Framework (utilisé pour réaliser des tests automatisés).
- La quatrième partie présente l'installation du programme CANGateway (comprenant la configuration de la Raspberry Pi, l'installation de CMocka et la compilation du programme).
- La cinquième partie présente l'installation de JMeter.

## 2 Installation de l'application CANdroid

CANdroid est lancé depuis Android Studio. Pour cela, nous vous proposons un tutoriel d'installation.

### 2.1 Installation de Android Studio

Rendez-vous sur le site officiel : Android Studio et téléchargez le fichier .zip. Placez le dans le dossier pertinent pour vos applications (par exemple dans /usr/local/). Dans un terminal, rendez-vous dans le répertoire android-studio/bin/. Ensuite, exécutez studio.sh.

Choisissez, si vous le souhaitez, d'importer les paramètres anciens de Android Studio. Cliquez sur OK pour valider. Poursuivez l'installation avec l'assistant de configuration d'Android Studio. Cela implique le téléchargement des composants SDK Android nécessaires au développement.

Si vous possédez une version 64 bits d'Ubuntu, il est nécessaire d'installer des bibliothèques 32 bits avec la commande suivante :

```
sudo apt install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 libbz2-1.0:i386
```

Si vous exécutez Fedora 64 bits, tapez :

```
sudo yum install zlib.i686 ncurses-libs.i686 bzip2-libs.i686
```

### 2.2 Configurer l'appareil

Sur votre Smartphone, dans les paramètres, activez les option pour les développeurs. Appuyez sept fois sur l'option Build Number (Numéro de version) de manière à ce que le message "You are now a developer!" s'affiche. Cela permet d'activer les options pour les développeurs sur votre appareil.

Vous devez ensuite activer le débogage USB de manière à ce que les outils SDK reconnaissent l'appareil s'il est connecté via USB. Activez le débogage USB dans les paramètres système de l'appareil dans Developer options (Options pour les développeurs). Le chemin d'accès dépend de la version Android :

- Android 9 (niveau d'API 28) ou version ultérieure : Settings > System > Advanced > Developer Options > USB debugging (Paramètres > Système > Avancé > Options pour les développeurs > Débogage USB)
- Android 8.0.0 (niveau d'API 26) et Android 8.1.0 (niveau d'API 27) : Settings > System > Developer Options > USB debugging (Paramètres > Système > Options pour les développeurs > Débogage USB)
- Android 7.1 (niveau d'API 25) ou version antérieure : Settings > Developer Options > USB debugging (Paramètres > Options pour les développeurs > Débogage USB)

Lorsque Android Studio est fonctionnel, branchez votre appareil. Dans Open'Edit Run/Debug configurations' Dialog > Edit Configuration... > CANdroid > Apply > OK. Puis cliquez sur Run 'CANdroid'.

### 3 Installation de Robot Framework

Afin d'exécuter des tests automatisés, nous vous proposons un tutoriel d'installation de PyCharm et Robot Framework.

#### 3.1 Installation de PyCharm

Commencez par installer python, dans un terminal tapez :

```
sudo apt install python3  
sudo apt install python3-pip
```

Téléchargez l'archive de PyCharm sur le site officiel : [Pycharm](https://www.jetbrains.com/pycharm/). Prenez la version 2022.3.3 (Community edition) pour votre OS. Désarchivez l'archive dans le dossier de votre choix.

Téléchargez appium :

```
sudo apt install npm  
npm install appium
```

Téléchargez Robot Framework :

```
pip install robotframework  
pip install robotframework-appiumlibrar  
pip install --force-reinstall -v "Appium-Python-Client==2.8.1"  
pip install --force-reinstall -v "selenium==4.8.2"
```

#### 3.2 Configuration de PyCharm

Commencez par lancer PyCharm :

```
./<path>/<to>/pycharm-community-2022.3.3/bin/pycharm.sh
```

Allez dans le menu File > Settings > Plugins, tapez "robot", installez "Robot Runner" et "IntelliBot #patched". Redémarrez l'IDE.

### 4 Installation du programme CANgateway

#### 4.1 Configuration de la Raspberry Pi

##### 4.1.1 Configuration du bus CAN

Si vous récupérez une Raspberry Pi neuve, il y aura un certain nombre de configurations à faire avant de pouvoir faire fonctionner le programme CANgateway. Vous pouvez également vous servir des informations suivantes pour vérifier que la Raspberry Pi que vous avez est correctement configurée.

Commencez par placer le RS485 CAN Hat sur la Raspberry Pi.

**Attention, si vous souhaitez utiliser le Banc de Test, il vous faut un RS485 CAN Hat sans résistance de terminaison.**

Le protocole CAN utilise le bus SPI pour communiquer, il faut donc configurer votre Raspberry Pi pour qu'elle active ce bus. Pour cela, éditez le fichier config.txt (en root) :

```
sudo nano /boot/config.txt
```

Parcourez le fichier, si vous ne les trouvez pas, tapez les lignes suivantes :

```
dtoverlay=spi=on
dtoverlay=mcp2515-can0,oscillator=12000000,interrupt=25,spimaxfrequency=2000000
```

Redémarrez la Raspberry Pi :

```
sudo reboot
```

Maintenant que le bus SPI est fonctionnel, il reste à configurer le bus CAN. Commencez par installer le package qui permettra de configurer le bus CAN :

```
sudo apt install can-utils
```

Afin de faciliter l'utilisation de la Raspberry Pi, voici un tutoriel pour que le bus CAN se configure correctement à chaque démarrage de la Raspberry Pi :

1. Créez un fichier can0.service et éditez le :

```
touch /etc/systemd/system/can0.service
sudo nano /etc/systemd/system/can0.service
```

2. Ecrivez les lignes suivantes et sauvegardez :

```
[Unit]
Description=CAN0 interface initialization

[Service]
ExecStart=/bin/bash -c "/sbin/ifconfig can0 down && /sbin/ip link set can0
type can bitrate 125000 && /sbin/ifconfig can0 up"

[Install]
WantedBy=multi-user.target
```

3. Tapez la ligne suivante pour activer le service :

```
sudo systemctl enable can0.service
```

4. Redémarrez la Raspberry Pi.

Après le démarrage de la Raspberry Pi, vous pouvez vérifier que le service s'est correctement lancé en tapant :

```
systemctl status can0.service
```

Vous devriez voir apparaître ceci :

```
can0.service — CAN0 interface initialization
Loaded : loaded (/etc/systemd/system/can0.service; enabled; vendor preset : enabled)
Active : inactive (dead) since Mon 2023-06-05 19 :53 :59 CEST; 22min ago
Process : 350 ExecStart=/bin/bash -c /sbin/ifconfig can0 down && /sbin/ip link set
can0 type can bitrate 125000 && /sbin/ifconfig can0 up (code=exited, status=0/
SUCCESS)
```



Main PID : 350 (code=exited, status=0/SUCCESS)  
CPU : 54ms

juin 05 19 :53 :59 raspberrypi systemd[1] : Started CAN0 interface initialization.  
juin 05 19 :53 :59 raspberrypi systemd[1] : can0.service : Succeeded.

Vous pouvez connaître l'état du bus CAN grâce à cette commande :

```
sudo ip -details link show can0
```

Le bus peut être dans divers états :

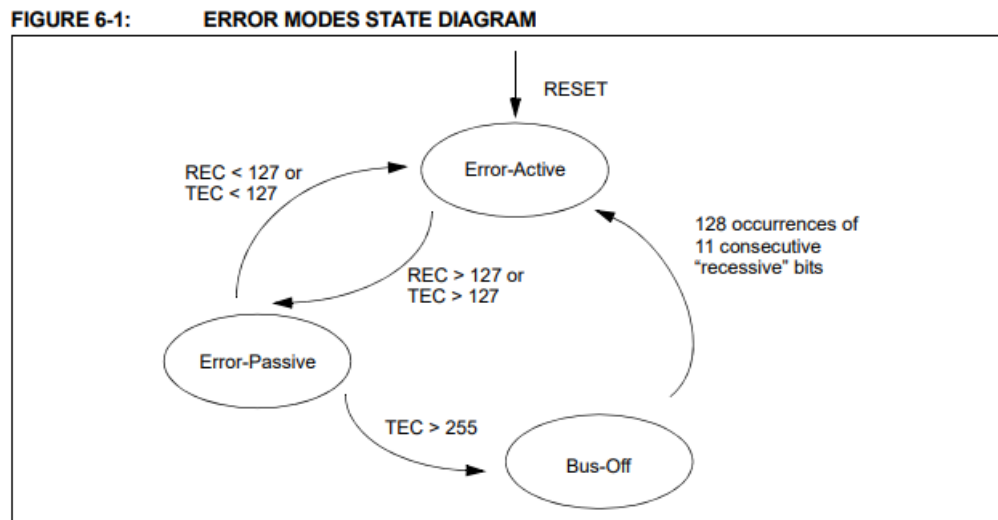


FIGURE 1 – Diagramme d'état des modes d'erreurs (extrait de la datasheet du MCP2515)

Lorsque le bus est éteint, il est dans l'état STOPPED :

```
3 : can0 : <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode
  DEFAULT group default qlen 10
link/can promiscuity 0 minmtu 0 maxmtu 0
can state STOPPED (berr-counter tx 0 rx 0) restart-ms 0
  bitrate 125000 sample-point 0.875
  tq 500 prop-seg 6 phase-seg1 7 phase-seg2 2 sjw 1
pcan_usb : tseg1 1..16 tseg2 1..8 sjw 1..4 brp 1..64 brp-inc 1
  clock 8000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs
    65535 parentbus usb parentdev 2-2.1 :1.0
```

Lorsque le bus est fonctionnel, il est dans l'état ERROR-ACTIVE :

```
3 : can0 : <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode
  DEFAULT group default qlen 10
link/can promiscuity 0 minmtu 0 maxmtu 0
can state ERROR-ACTIVE (berr-counter tx 0 rx 0) restart-ms 0
  bitrate 125000 sample-point 0.875
  tq 500 prop-seg 6 phase-seg1 7 phase-seg2 2 sjw 1
pcan_usb : tseg1 1..16 tseg2 1..8 sjw 1..4 brp 1..64 brp-inc 1
```

```
clock 8000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs
65535 parentbus usb parentdev 2-2.1 :1.0
```

Lorsque le bus est en erreur, il est dans l'état ERROR-PASSIVE :

```
3 : can0 : <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode
  DEFAULT group default qlen 10
link/can promiscuity 0 minmtu 0 maxmtu 0
can state ERROR-PASSIVE (berr-counter tx 0 rx 0) restart-ms 0
  bitrate 125000 sample-point 0.875
  tq 500 prop-seg 6 phase-seg1 7 phase-seg2 2 sjw 1
  pcan_usb : tseg1 1..16 tseg2 1..8 sjw 1..4 brp 1..64 brp-inc 1
  clock 8000000 numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs
  65535 parentbus usb parentdev 2-2.1 :1.0
```

Dans cet état, il faut redémarrer le bus pour qu'il recommence à fonctionner. Pour cela, vous pouvez soit redémarrer la Raspberry Pi (dans le cas où le service est "enable" et se lance tout seul au boot de la Raspberry Pi), soit taper la commande :

```
sudo systemctl restart can0.service
```

#### 4.1.2 Installation du Hotspot

Afin de permettre au Smartphone de se connecter à la Raspberry Pi, il lui faut générer un Hotspot.

Pour cela, vous pouvez suivre le tutoriel de ce site : Créer un hotspot sur Raspberry Pi.

1. Installez RaspAP :

Création d'une sauvegarde du fichier de configuration WiFi :

```
sudo cp /etc/wpa_supplicant/wpa_supplicant.conf /etc/wpa_supplicant/
wpa_supplicant.conf.sav
```

Suppression du fichier de configuration WiFi pour retourner à une configuration vierge :

```
sudo cp /dev/null /etc/wpa_supplicant/wpa_supplicant.conf
```

Téléchargement et installation de RaspAP :

```
wget -q https://git.io/voEUQ -O /tmp/raspap && bash /tmp/raspap
```

2. Attendez la fin du téléchargement et redémarrez la Raspberry Pi.

Ce réseau est automatiquement déployé sur wlan0. Vous pouvez taper la commande suivante pour vérifier que le hotspot est en place :

```
ip a
```

Vous devriez voir à minima :

```
wlan0 : <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
  state UP group default qlen 1000
link/ether b8:27:eb:7d:39:15 brd ff:ff:ff:ff:ff:ff
inet 10.3.141.1/24 brd 10.3.141.255 scope global noprefixroute wlan0
```

```
valid_lft forever preferred_lft forever
inet6 fe80 ::bc4f :144b :a962 :a45f/64 scope link
valid_lft forever preferred_lft forever
```

Vous pouvez connecter votre Smartphone ou votre PC au hotspot, le nom du réseau est "raspi-webgui". Le mot de passe est "ChangeMe".

L'interface d'administration de RaspAP est accessible en vous connectant au hotspot, tapez "10.3.141.1" dans un navigateur web. Par défaut, le nom d'utilisateur est "admin" et le mot de passe est "secret". Vous pouvez configurer le hotspot à votre convenance.

## 4.2 Installation des outils pour compiler le programme CANgateway

### 4.2.1 Installation du compilateur

Afin de compiler le programme pour Raspberry Pi, il faut installer le compilateur ARM. Vous pouvez utiliser celui que vous souhaitez mais nous vous proposons ce tutoriel pour récupérer le compilateur utilisé lors du développement :

1. Récupérer les outils de développement croisés :

Dans le dossier de votre choix, tapez la commande suivante :

```
git clone https://github.com/raspberrypi/tools.git
```

2. Modifier le fichier "variable.mk" afin de renseigner le chemin absolu vers le compilateur (dossier où vous avez cloné le fichier précédemment) : modifiez la variable "RASP-BERRY\_TOOLS".

Si ce n'est pas déjà fait, pensez également à installer gcc sur votre machine, dans n'importe quel terminal, tapez :

```
sudo apt install gcc
```

### 4.2.2 Installation de CMocka

Le programme CANgateway est fourni avec des tests automatisés. Afin de compiler ces tests, vous avez besoin de la librairie CMocka. Les tests sont autant exécutables sur votre machine que sur cible. Si votre machine ne possède pas la même architecture que la Raspberry Pi, vous avez donc besoin de deux librairies différentes. Afin de faciliter l'utilisation de CMocka, nous vous proposons de compiler la librairie en statique.

Le tutoriel est le même pour les deux architectures, la seule différence est que pour une compilation sur cible, vous devez compiler la librairie sur la cible, pour une compilation locale, vous devez compiler la librairie sur votre machine.

Attention, afin de compiler la librairie, vous avez besoin de cmake et make, dans un terminal, tapez :

```
sudo apt install cmake
sudo apt install make
```

1. Téléchargez CMocka CMocka, version 1.1.5.
2. Consultez les fichier README et INSTALL pour plus d'informations sur l'installation de CMocka.
3. Dans le fichier "DefineOptions.cmake" :  

```
option(WITH_STATIC_LIB "Build with a static library" ON)
```
4. Créez un répertoire "build" dans le dossier de CMocka et allez dedans.
5. Générez le Makefile avec cmake :  

```
cmake -DCMAKE_INSTALL_PREFIX=<choisir un emplacement> ..
```
6. Compilez et installez la librairie :  

```
make  
make install
```

Si vous avez compilé la librairie sur cible, vous pouvez récupérer les dossier "include" et "lib" dans le dossier d'installation que vous avez choisi et les installer sur votre machine. Vous pouvez utiliser la commande "scp" pour copier les dossiers de la cible vers votre machine : Sur votre machine, tapez :

```
scp -r <user>@<ip> :<chemin vers le dossier> <chemin vers le dossier de destination>
```

Après avoir récupéré les deux librairies, modifiez le fichier "variable.mk" afin de renseigner les chemins absolus vers les librairies : modifiez la variable "CMOCKA" (Attention, elle est déclarée deux fois dans le fichier selon si la variable "TARGET" est égale à "raspberrypi" ou non).

#### 4.2.3 Compilation du programme CANGateway

Vous avez désormais tous les outils nécessaires pour compiler le programme CANGateway.

Comme expliqué précédemment, il est possible d'exécuter les tests automatisés sur cible et sur votre machine.

Lorsque le programme est compilé sur cible, un script est appelé afin de transmettre l'exécutable à la Raspberry Pi. Pour que cela fonctionne, assurez-vous de pouvoir vous connecter à la Raspberry Pi en SSH.

Vous pouvez modifier le fichier "transfert\_bin.sh" afin de renseigner le nom d'utilisateur et l'adresse IP de la Raspberry Pi.

Dans le cas où votre Raspberry Pi aurait un port de connexion SSH différent de celui par défaut, pensez à modifier la commande "scp" du fichier afin de lui indiquer le port à utiliser : utilisez le paramètre "-P" suivi du numéro du port.

Pour compiler :

1. Ouvrez un terminal dans le dossier "production".
2. Pour compiler sur cible, tapez :  

```
make TARGET=raspberrypi
```

3. Pour compiler sur votre machine, tapez :

```
make
```

Une fois la compilation terminée, vous pouvez voir les exécutables "CANgateway.out" et "CANgateway\_test.out" dans le dossier "bin".

Pour une compilation sur cible, les exécutables sont visibles dans le dossier /home/pi de la Raspberry Pi.

## 5 Installation de ICSim

Nous vous proposons un tutoriel d'installation du simulateur ICSim que nous avons utilisé pour développer notre système.

ICSim est un logiciel open source, pour l'installer, nous avons suivi les instructions du README.md du projet disponible sur GitHub : ICSim.

Dans ce tutoriel, ICSim est lancé avec l'interface virtuelle vcan0. Il est cependant possible de l'utiliser avec une interface réelle. Pour cela, il faut remplacer vcan0 par can0 dans les commandes du README.

Attention, néanmoins, pour que l'interface CAN réelle fonctionne, vous devez avoir connecté votre PC à un device CAN. Dans le cas contraire, les commandes de configurations de l'interface CAN réelle ne fonctionneront pas.

## 6 Installation de JMeter

Pour la réalisation des tests fonctionnels du serveur TCP implémenté dans le module Postman, nous avons utilisé JMeter.

JMeter est un outil open-source de test de performance et de charge développé par Apache. Il permet de mesurer et d'évaluer les performances d'une application web ou d'un serveur en simulant différents scénarios d'utilisation. JMeter offre une grande flexibilité pour la création de tests et fournit des statistiques détaillées sur les performances, la fiabilité et la capacité de l'application testée.

Tout d'abord, assurez-vous d'avoir une version récente de Java installée sur votre système, car JMeter est basé sur Java. Pour l'installer, rendez-vous sur le site officiel d'Oracle Java à l'adresse suivante : <https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>. Dans notre cas nous avons installé le JDK (Java Development Kit) d'OpenJDK 11.

Pour installer JMeter il faut :

— Allez sur le site officiel d'Apache JMeter à l'adresse suivante : <https://jmeter.apache.org/>.

- Dans la section "Downloads", cliquez sur le lien correspondant à la version la plus récente de JMeter. Vous serez redirigé vers la page de téléchargement.
- Sur la page de téléchargement, choisissez le fichier binaire qui correspond à votre système d'exploitation (Windows, Linux, Mac OS, etc.) et téléchargez-le sur votre ordinateur.
- Une fois le téléchargement terminé, décompressez le dossier dans le répertoire de votre choix.
- Accédez au répertoire extrait, allez dans le répertoire bin/ et lancez JMeter en exécutant le fichier "jmeter.bat" pour Windows ou "jmeter.sh" pour Linux/Mac OS.
- JMeter s'ouvrira avec une interface graphique. Vous êtes maintenant prêt à commencer à créer et exécuter des tests de performance.