# Documentation Technique

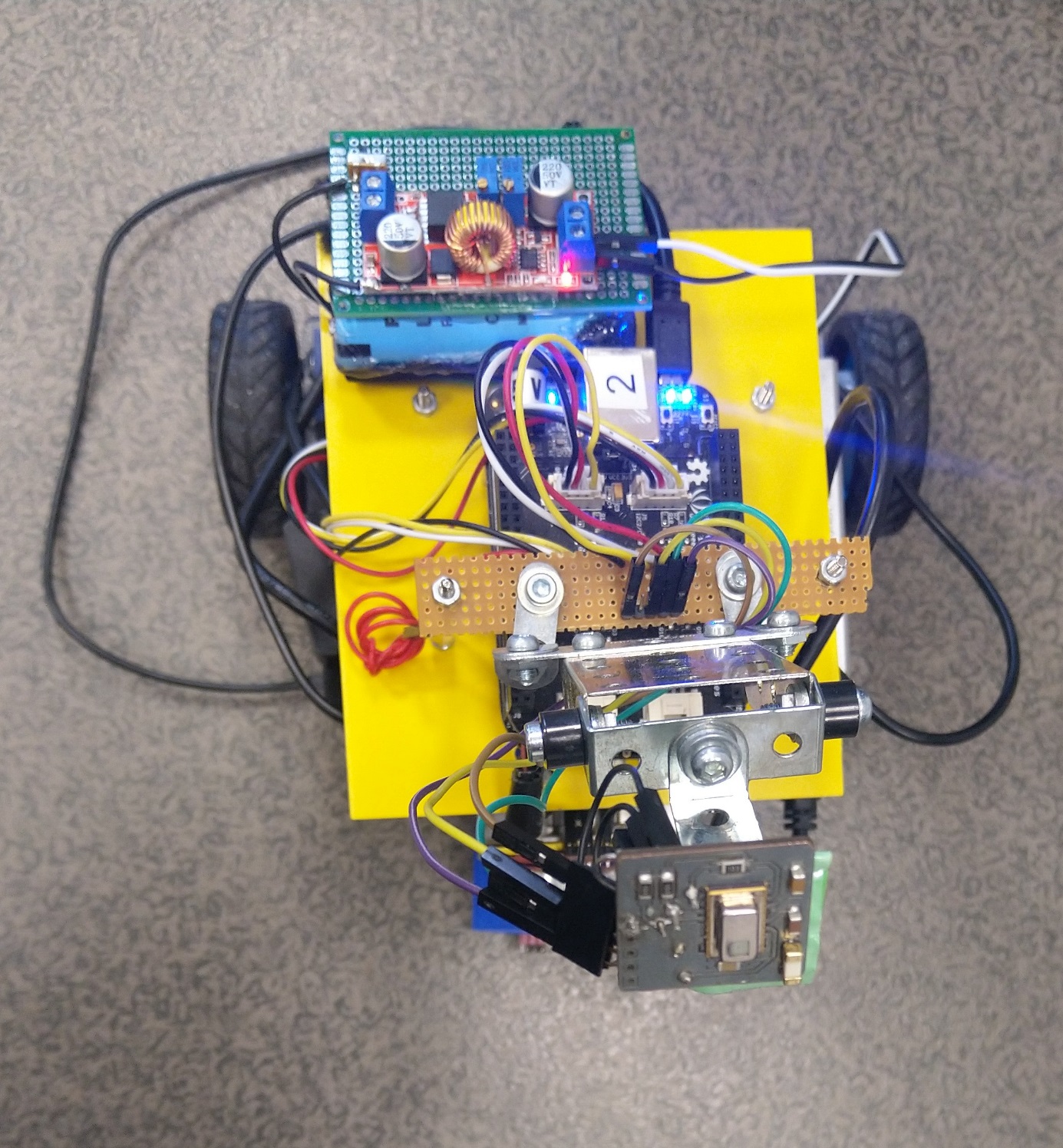


Table des matières

[Documentation Technique 1](#_Toc62124158)

[Configuration de la machine virtuelle 3](#_Toc62124159)

[1) Ajouter la VM 3](#_Toc62124160)

[2) Automatisation de l’ajout de l’USB serial 3](#_Toc62124161)

[3) Démarrage 3](#_Toc62124162)

[4) Installation et configuration du serveur NFS 3](#_Toc62124163)

[5) Construction du système de fichiers, noyau et bootloader : 4](#_Toc62124164)

[6) Actions effectuées sur le device-tree 5](#_Toc62124165)

[Configuration du serveur Boa 6](#_Toc62124166)

[Génération exécutable 6](#_Toc62124167)

[Automatisation du lancement de service 6](#_Toc62124168)

[Configuration de la Carte Teensy 7](#_Toc62124169)

[1) Prérequis logiciels 7](#_Toc62124170)

[2) Commandes disponibles 7](#_Toc62124171)

[Création d’une carte SD 9](#_Toc62124172)

[1) Sur Linux à partir de Targetfs 9](#_Toc62124173)

[2) Avec le fichier image 9](#_Toc62124174)

[Algorithmes 10](#_Toc62124175)

[1) Fonctionnement global du robot 10](#_Toc62124176)

[2°) Calibrage 10](#_Toc62124177)

[Programme 10](#_Toc62124178)

[Description 10](#_Toc62124179)

[moteur.c 11](#_Toc62124180)

[caméra.c 11](#_Toc62124181)

[ultrason.c 11](#_Toc62124182)

[robot.c 11](#_Toc62124183)

[Les programmes de débogage 11](#_Toc62124184)

[moteur 11](#_Toc62124185)

[camera 11](#_Toc62124186)

[ultrason 12](#_Toc62124187)

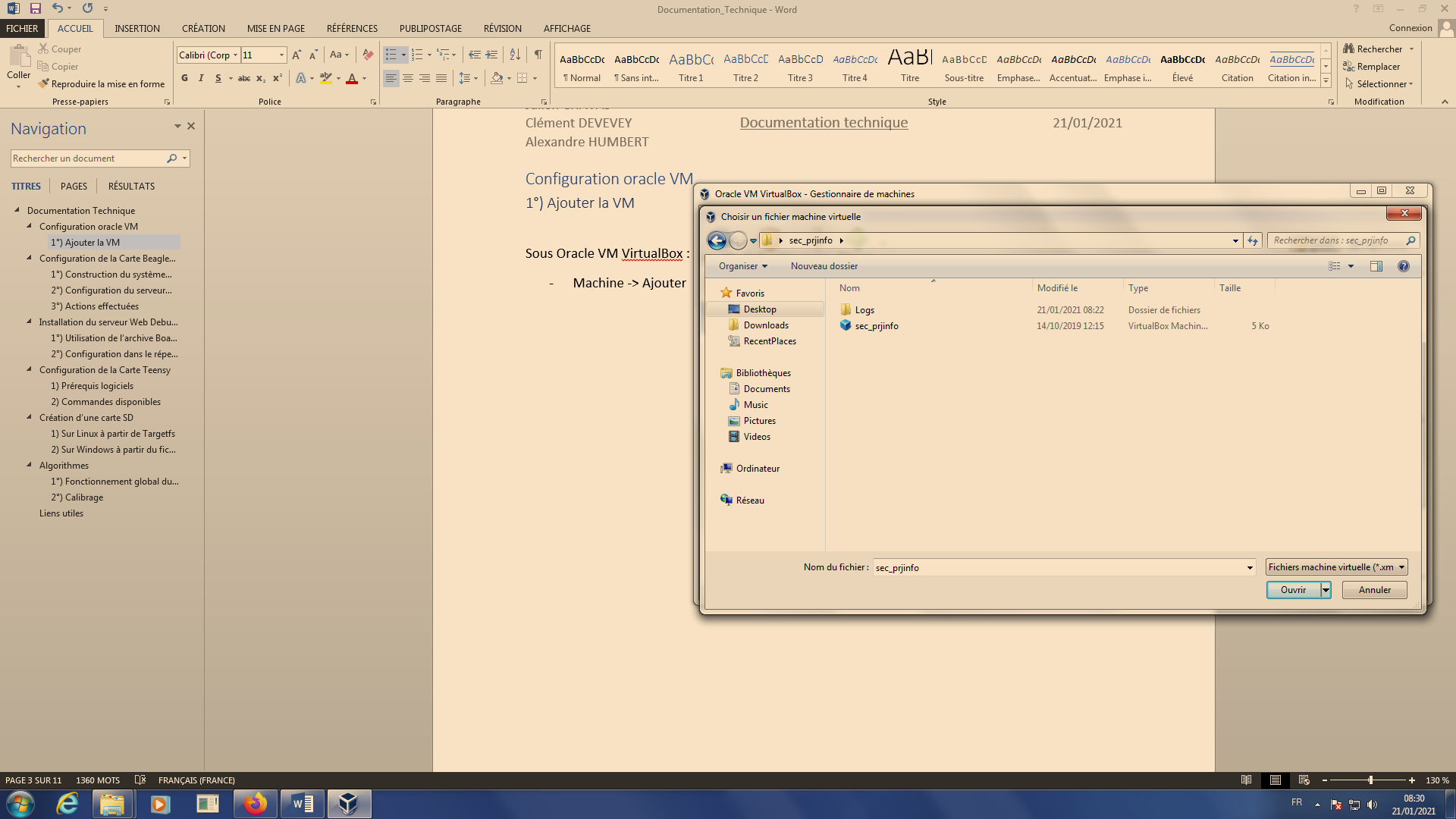
[Liens utiles 12](#_Toc62124188)

## Configuration de la machine virtuelle

### 1) Ajouter la VM

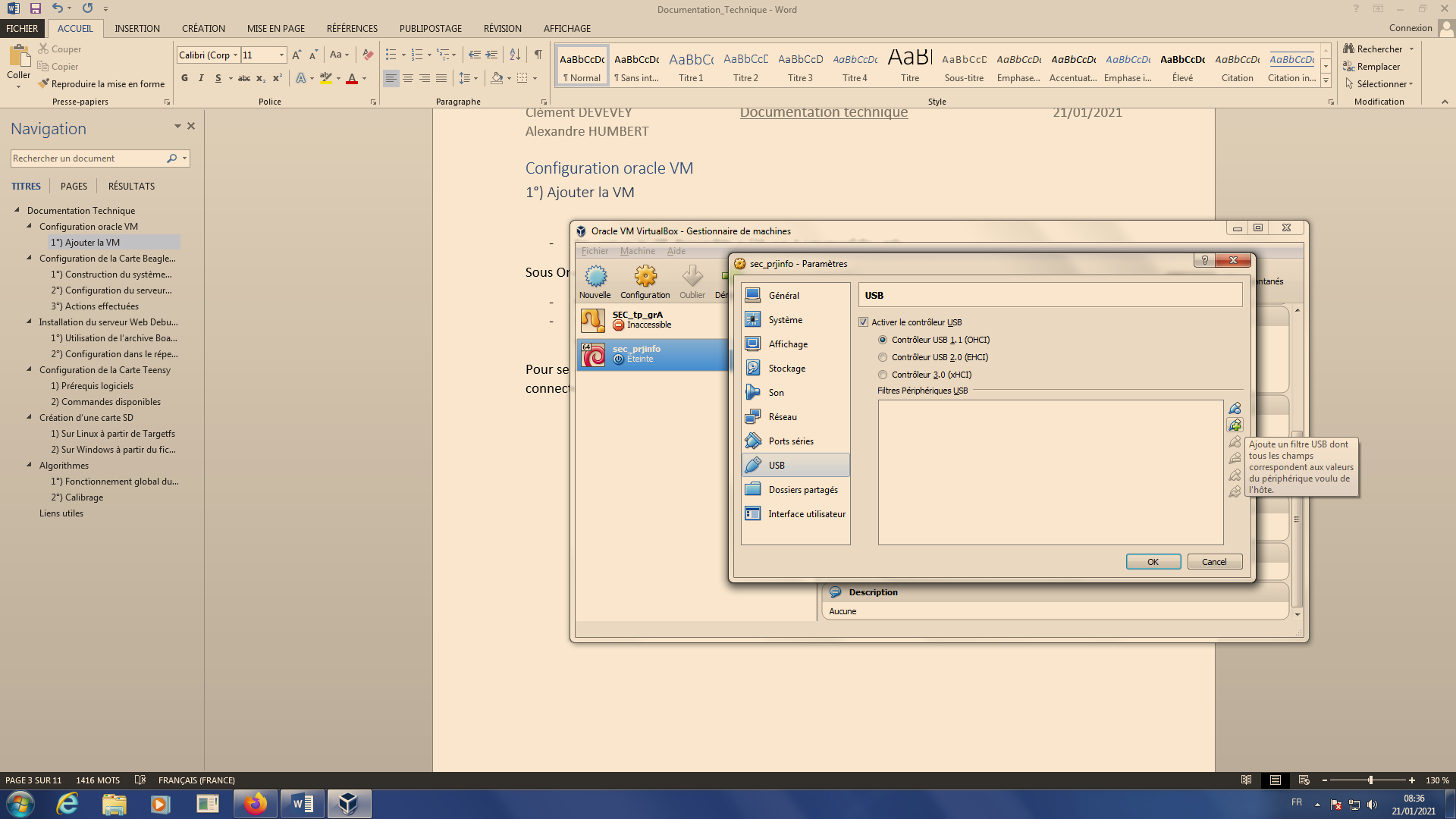
* Récupérer le ZIP disponible à TSE sur Y:\mvpool\Prj\_info.

Sous Oracle VM VirtualBox :

* Machine -> Ajouter ->
* Dans configuration -> réseau -> choisir la carte 1 NAT -> carte 2 : accès par pont (choisir votre Ethernet)
* Créer un dossier vide (mv\_share) sur votre windows
* Dans configuration -> dossiers partagés -> et modifier le chemin dans Dossiers permanents

### 2) Automatisation de l’ajout de l’USB serial

Pour se simplifier la vie, il est possible d’automatiser le branchement du port serial pour contrôler la beaglebone :

* Alimenter la BeagleBone black.
* Brancher le connecteur série au PC.
* Configuration -> USB -> Ajouter un filtre USB.
* Sélectionner le port serial.

### 3) Démarrage

Utiliser le compte tpuser (mdp : tpuser) pour un utilisateur lambda.

Utiliser le compte root (mdp : admintest) pour un super-utilisateur.

* En **root** : **apt-get install net-tools**

Ceci vous permet d’utiliser ifconfig.

Rendez-vous en **root** dans /etc/network puis faire

- **mousepad interfaces** et recopier le contenu du fichier interfaces (fournit dans le zip ici : [fs-VM/etc/network/](https://github.com/alexandre-humbert/robot-thermique/commit/32426c7c98e1d39707ce4d05b4699768787f9193)interfaces)

- **reboot** afin de redémarrer la machine virtuelle avec les bonnes configurations IP.

Ceci vous permet d’avoir internet sur la machine virtuelle ainsi qu’attribuer une IP au PC (192.168.101.36).

### 4) Installation et configuration du serveur NFS

* Installation du serveur NFS:
  + **(en root)** dpkg –i /opt/packages/server/nfs-kernel-server\_1.2.8-9\_amd64.deb
  + **(en root)** mkdir /targetfs
  + **(en root)** mkdir /targetfs/boot
* Ajouter la ligne dans « /etc/exports » :
  + /targetfs \*(rw, no\_root\_squash, sync, no\_subtree\_check)
* Executer en super-utilisateur :
  + systemctl restart nfs-kernel-server
* Régler l’IP PC et Carte (via Uboot : régler vos IPs si vous les avez changé) :
* Run import bootenv , run loadbootenv, setenv ipaddr 192.168.101.29, setenv serverip 192.168.101.36, saveenv.

### 5) Construction du système de fichiers, noyau et bootloader :

En **root** :

* Sous /opt/packages/gcc : **bash gcc\_inst**
* Sous /opt/packages/build : **bash build\_inst**

En **tpuser** :

* Sous /sec/download : **tar xvf buildroot.tar.gz -C /sec**
* Sous /sec/buildroot : **make beaglebone\_defconfig**
* Sous /sec/buildroot : **mkdir dl**
* N’importe ou : **cp –a /sec/download/dltool/\* /sec/buildroot/dl**
* Sous /sec/buildroot : **make toolchain**
* **Copier** le fichier « am335x-projet.dtsi » dans :
  + /sec/buildroot/output/build/linux- headers-tse/arch/arm/boot/dts
* Sous /sec/buildroot/output/build/linux- headers-tse/arch/arm/boot/dts
  + **Rajouter** la ligne : **#include "am335x-projet.dtsi"**
* Sous /sec/buildroot/output/build/linux-headers-tse/arch/arm/boot/dts executer :
  + **make**
* Sous /sec/buildroot/ executer :
  + **make menuconfig** 
    - **Cocher** Target packages -> Networking application -> [\*]**Boa**
    - **Cocher** Filesystem -> Filesystem images -> [\*] **tar the root filesystem**
  + make **device-tree**
  + make **kernel**
  + make **filesystem**

En **root** :

* Sous /sec/buildroot/ output/images executer
  + **tar xvf rootfs.tar -C /targetfs**
  + **cp zImages /targetfs/boot**
  + **cp am335x-boneblack.dtb /targetfs/boot**

### 6) Actions effectuées sur le device-tree

* Activation de l’UART1 :
  + Communication avec la carte teensy
* Activation I2C1 et I2C2
  + Communication avec le capteur thermique
  + Communication avec le capteur ultrason

## Configuration du serveur Boa

En root :

Copier le mini-site dans /targetfs/var/www/

Copier le fichier de configuration boa.conf du répertoire fs-BB dans /targetfs/etc/boa/boa.conf

Copier le script disponible dans fs-BB/usr/lib/chi-bin/interface.cgi dans /targetfs/usr/lib/cgi-bin (créer le répertoire s’il n’existe pas)

Mettre les droits d’exécution : chmod 777 /targetfs/usr/lib/cgi-bin/interface.cgi

## Génération exécutable

Récupérer tous les fichiers sources disponibles dans le dossier programme.

Pour que le make réussisse, il faut créer un dossier build (**mkdir build**) à l’endroit où les fichiers sources sont. Ce dossier contiendra tous les fichiers objets (.o) générés par le Makefile.

Il ne vous reste plus qu’à make !

## Automatisation du lancement de service

Récupérer les fichiers disponibles dans /fs-BB/etc/init.d/ et les copier dans /targetfs/etc/init.d

Cette opération vous permet de lancer des services qui exécutent vos programmes au démarrage de la carte Beaglebone Black. Attention à ne pas laisser votre robot sur la table lors du démarrage.

## Configuration de la Carte Teensy

### Prérequis logiciels

Nous avons créé notre propre fichier .ino à la suite de problèmes d’asservissement des moteurs. Il n’y a plus d’asservissement mais des coefficients pour supprimer les différences de vitesse entre les deux roues.

Pour compiler et téléverser ce fichier sur la carte Teensy, il faut utiliser les logiciels [**Teensyduino**](https://www.pjrc.com/teensy/td_download.html) avec [**Arduino**](https://www.arduino.cc/en/software) :

1. Ouvre le .ino avec ArduinoIDE
2. Branchez la Teensy au PC en USB
3. Dans ArduinoIDE : Outil -> Type de carte -> Choisir Teensy 3.2/3.1
4. Choisir son port dans Outils -> Port
5. Cliquer sur la flèche pour téléverser (Teensyduino s’ouvre si la compilation réussi)

Ce fichier inclut d’autres fichiers .h et .c, attention à bien les mettre dans le dossier « arduino/library ». **Sinon** changer les #include<fichier.h> en #include"fichier.h" et les mettre dans le même dossier que le .ino.

### Commandes disponibles

Via **l’UART1** (/dev/ttyO1) on va pouvoir envoyer des commandes (sous forme de deux caractères) à la Teensy depuis la beaglebone. Exemple : echo av > /dev/ttyO1

Voici le tableau récapitulatif des commandes disponibles :

|  |  |
| --- | --- |
| Nom commande | Action |
| av | Avancer |
| st | Stop |
| ar | Reculer |
| dr | Avancer droite |
| ga | Avancer gauche |
| ad | Reculer droite |
| ag | Reculer gauche |
| td | Tourner gauche |
| tg | Tourner gauche |
| pd | Pivoter droite |
| pg | Pivoter gauche |

**Pour les commandes av et ar uniquement :**

* "av n" permet d'avancer à la vitesse n (n compris entre 0 et 255, de préférence entre 120 et 255).
* "av n m" permet d'avancer à la vitesse n pour la roue droite et m pour la roue gauche.
* "av 1", "av 2", "av 3" et "av 4" permettent d'avancer à des vitesses prédéfinies.
* "ar n" permet de reculer à la vitesse n (n compris entre 0 et 255, de préférence entre 120 et 255).
* "ar n m" permet de reculer à la vitesse n pour la roue droite et m pour la roue gauche.
* "ar 1", "av 2", "av 3" et "av 4" permettent de reculer à des vitesses prédéfinies.

## Création d’une carte SD

### Sur Linux à partir de Targetfs

* Pourquoi créer une carte SD : une fois le programme au point, vous pourrez recréer un linux embarqué afin que votre programme puisse tourner sur le robot directement au démarrage de la carte. Voici comment procéder :
* Trouvez quel périphérique correspond à votre carte SD avec la commande : **ls /dev**
* Sur la VM, ce périphérique sera surement **/dev/sdd**  
  Les instructions suivantes sont données pour un périphérique **sdd**. Adaptez les commandes si votre chemin est différent.
* Ensuite **en root:**
  + Effacez la table de partition : **dd if=/dev/zero of=/dev/sdd bs=1M count=1**
  + Créez une nouvelle table : **sfdisk -H 255 -S 63 /dev/sdd << EOF  
    0,1,c,\*  
    ,2  
    EOF**
  + Ensuite créez la partition de boot : **mkfs.vfat /dev/sdd1 -n boot**
  + Créez le point de montage : **mkdir /media/boot**
  + Puis Montez cette partition : **mount /dev/sdd1/boot /media/boot**
  + Copiez les fichiers sur cette partition : **cd /sec/buildroot/output/images**

**cp MLO zImage u-boot.img am335x-boneblack.dtb /media/boot**

* + Créez la seconde partiton : **mkfs.ext2 /dev/sdd2 -n**
  + Montez cette partition puis placez-vous dans le répertoire
  + Copiez votre système de fichier : **cp -r /targetfs/\* ./**

Enfin, il faudra aller du côté de /etc/init.d et créer un service qui lancera votre programme au démarrage. (Voir dans fs-BB -> etc -> init.d pour plus d’informations)

### Avec le fichier image

Sur Linux :

Télécharger sdcard.img et installer avec la commande: **dd bs=1M if=sdcard.img of=/dev/sdd**

Sur Windows :

Il est possible d’installer l'image avec un logiciel comme [Win32Disk](https://sourceforge.net/projects/win32diskimager/).

Télécharger sdcard.img (disponible dans le dossier source) puis choisissez ce fichier dans le logiciel, sélectionnez votre carte SD et cliquez sur "Write".

## Algorithmes

### 1) Fonctionnement global du robot

Le robot, pour effectuer son suivi thermique, utilise un capteur thermique possédant une matrice de 8x8 pixels. Sachant que l’on veut suivre une personne, donc un « objet » vertical, on se fît aux colonnes de la matrice en faisant la moyenne colonne par colonne. Ensuite si la moyenne la plus élevée dépasse le seuil de déclanchement, elle va donc être traduite en une instruction au moteur suivant la colonne.

Si le capteur ultrason détecte un obstacle à moins de 15cm alors :

* Si la matrice détecte une source de chaleur de proche, avec la matrice du capteur est « pleine » c’est-à-dire que moyenne total de la matrice du capteur thermique dépasse un seuil :   
  8 x 6 x 30 + 8 x 2 x temp\_amb
* Si la matrice ne détecte pas une source de chaleur proche, le robot effectue une séquence prédéfinie pour éviter l’obstacle, pendant cette séquence le robot reste capable de s’arrêter en cas d’objet proche.

Si le robot est connecté en Ethernet un site web Boa peut servir à visualiser les informations issues des capteurs.

* Des flèches permettent le fonctionnement du robot manuellement
* Un affichage de la matrice avec la température
* Affichage température ambiante.
* Un affichage couleur en fonction de la température minimum
* Le retour de distance du capteur ultra-son
* Possibilité de modifier
  + Le min et le max pour régler les couleurs
  + Taper une commande au clavier
  + Afficher ou non les températures sur chaque carré de la matrice

### 2°) Calibrage

Le programme est calibré pour 25° de température ambiante (afficher par le capteur), pour une température inférieure à cette dernière, le fonctionnement sera le même. En revanche si la température est plus élevée, d’une le programme sera moins performant car la différence de température avec le corps sera moindre, mais aussi il faudra ajuster le seuil « COEF\_TEMP » dans « camera.h ». Par exemple pour une température ambiante de 30° affichée par le capteur, il faut un « COEF\_TEMP » de 1,2.

Par défaut le « COEF\_TEMP » est réglé à 1,04.

## Programme

### Description

Ce programme en langage C permet au robot de suivre une personne avec une matrice thermique.  
Il est composé de 4 fichiers :

* le fichier moteur.c
* le fichier camera.c
* le fichier ultrason.c
* le fichier robot.c

### moteur.c

Il contient tout le code permettant de communiquer avec la Teensy en UART. Elle permet d'avancer, reculer, s'arrêter.

### caméra.c

Il contient le code pour acquérir une image depuis la matrice thermique en I2C et le code pour exploiter ces données.

### ultrason.c

Il contient le code pour mesurer une distance avec le capteur à ultrason

### robot.c

Il contient le programme principal.  
Il permet de piloter le robot en fonction des données des capteurs.

## Les programmes de débogage

### moteur

Ce programme permet d'enoyer des commandes aux moteurs.  
Exemples:

* "moteur av" permet d'envoyer la commande avancer.
* "moteur av 1" permet d'envoyer la commande avancer à la vitesse 1.  
  Se référer à la partie "Programme de pilotage des moteurs de la Teensy" de la documentation pour plus d'information sur les commandes.

### camera

Ce programme permet d'afficher une image de la caméra sous forme de matrice ainsi que la température ambiante et d'autres informations issues de la caméra.  
Il dispose de plusieurs options :

* -h pour afficher l'aide
* -t pour ajouter un délai
* -m pour chosir une temprérature minimale
* -M pour chosir une tempréature maximale
* -W pour afficher une image au format SVG

### ultrason

Ce programme permet d'afficher la distance et la luminosité mesurées par le capteur ultrason. Il dispose de plusieurs options :

* -h pour afficher l'aide
* -t pour ajouter un délai
* -g pour changer le gain
* -r pour changer la portée

## Liens utiles

* [Grove Cape for BeagleBone](https://wiki.seeedstudio.com/Grove_Cape_for_BeagleBone_Series/)
* [Grid-Eye Panasonic](https://industry.panasonic.eu/components/sensors/grid-eye)
* [BeagleBone Black](https://beagleboard.org/black)
  + [Schéma des GPIO](https://www.mathworks.com/help/supportpkg/beagleboneio/ug/the-beaglebone-black-gpio-pins.html)
* [Capteurs à ultrasons SRF08](https://www.robot-electronics.co.uk/htm/srf08tech.html)