

Responsable du document : Martin ARONDEL

1. INTRODUCTION

a. RAPPELS DU PROJET

L'objectif de projet SCOUTBOT est d'intégrer un système de mapping et de positionning sur le robot pour qu'il puisse par la suite se déplacer dans une pièce de manière autonome.

Afin de réaliser le mapping d'une pièce, on utilise le RPLIDAR A1 de chez SLAMTEC.



Figure 1: RPLIDAR A1 SLAMTEC

b. GENERALITES

Pour que le lidar fonctionne correctement sur la STM32MP157-DK2, le driver de communication port série CP210x doit être installé.

SLAMTEC propose un SDK complet avec quelques projets exemples pour utiliser ce lidar.

Lien du SDK : https://github.com/Slamtec/rplidar_sdk

2. INSTALLATION DU DRIVER CP210X SUR LE STM32MP157-DK2

Etape 1 : Accéder au SDK de la STM32MP1

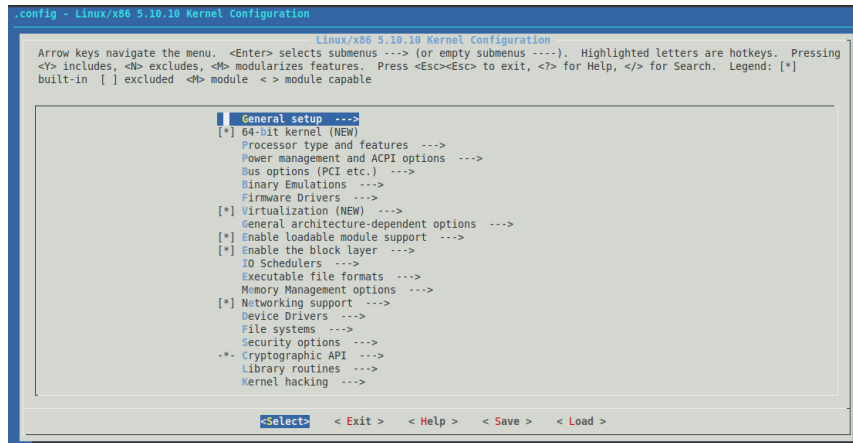
```
PC $ cd /home/ «votre home»/STM32MPU_workspace/STM32MP15-Ecosystem-v3.0.0/Developer-  
Package/stm32mp1-openstlinux-5.10-dunfell-mp1-21-03-31/sources/arm-ostl-linux-gnueabi/linux-  
stm32mp-5.10.10-r0/linux-5.10.10
```

Etape 2 : Sourcer le SDK pour utiliser le bon compilateur

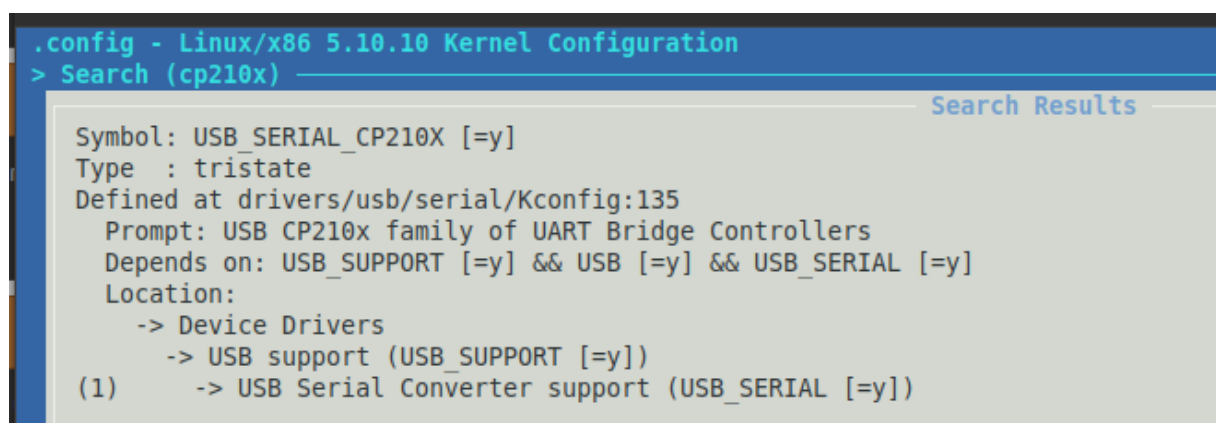
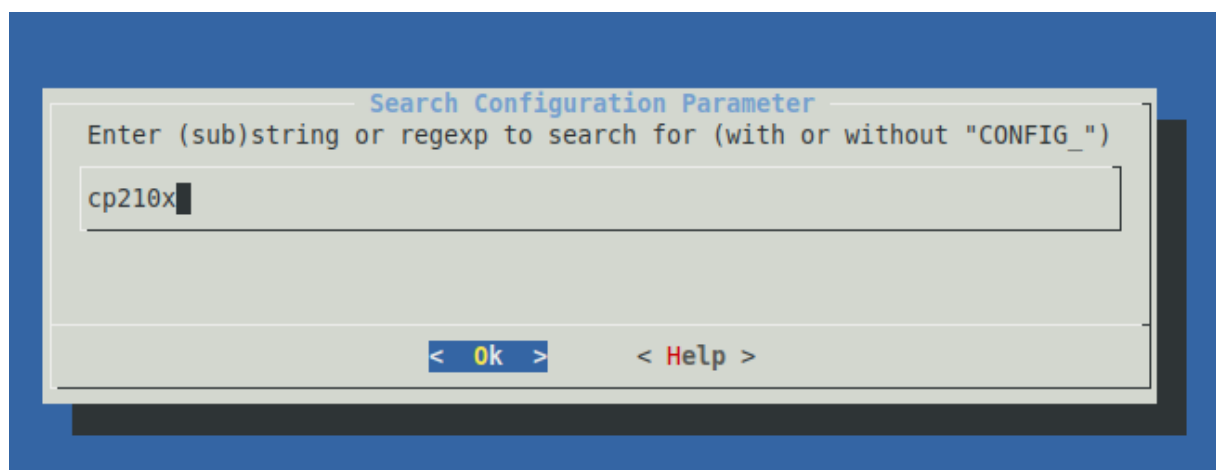
```
PC $ source /home/«votre home»/STM32MPU_workspace/STM32MP15-Ecosystem-  
v3.0.0/Developer-Package/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi
```

Etape 3 : Accéder au menu config du Kernel pour activer le driver

```
PC$ make arch=ARM menuconfig
```



Une fois lancé, pour effectuer une recherche faire un « / »



On retrouve la location du driver sur le menu config du kernel

Accéder au Device Driver

Accéder à USB support

Appuyer sur « espace » pour changer l'état du dossier (M = module, étoile = built-in) en « étoile »

```
[*] HDMI CEC drivers --->
<M> Multimedia support --->
    Graphics support --->
<M> Sound card support --->
    HID support --->
[*] USB support --->
<*> MMC/SD/SDIO card support --->
< > Sony MemoryStick card support ----
[*] LED Support --->
[ ] Accessibility support ----
```

Accéder à USB Serial Convert support

Mettre aussi une étoile en appuyant sur espace

```
<*> Enable Tegra UDC glue driver
<*> NXP ISP 1760/1761 support
    ISP1760 Mode Selection (Dual Role mode) --->
    *** USB port drivers ***
<*> USB Serial Converter support --->
    *** USB Miscellaneous drivers ***
```

Puis activer le driver CP210x

```
< > USB Winchiphead CH341 Single Port Serial Driver
< > USB ConnectTech WhiteHEAT Serial Driver
< > USB Digi International AccelePort USB Serial Driver
<*> USB CP210x family of UART Bridge Controllers
< > USB Cypress M8 USB Serial Driver
```

Save et quitter le menuconfig

Etape 4 : Save les configurations du menu config

PC \$ make arch=ARM savedefconfig

Etape 5 : Build kernel images

```
PC $ make arch_ARM ulmage vmlinux LOADADDR=0xc2000040
```

Etape 5 : Envoyer l'image Linux sur le MP1

```
PC $ cp arch/arm/boot/ulmage install_artifact/boot/
```

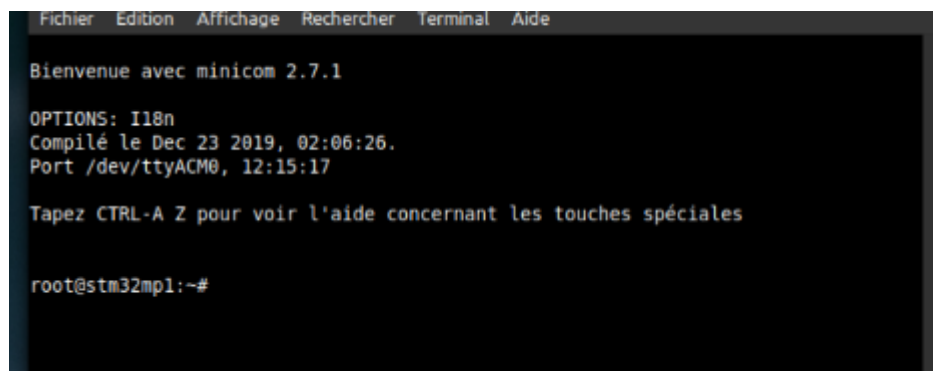
```
PC $ scp install_artifact/boot/ulmage root@192.168.7.1:/boot/
```

192.168.7.1 est l'adresse ip du MP1 lorsque qu'il est connecté en USB OCD (ssh par cable)

Etape 6 : Se connecter au MP1 par le minicom via le ST-Link

Via le ST-Link de la carte, s'assurer que le câble micro USB est bien connecté.

```
PC $ minicom -D /dev/ttyACM0
```



```
Fichier Edition Affichage Rechercher Terminal Aide
Bienvenue avec minicom 2.7.1
OPTIONS: I18n
Compilé le Dec 23 2019, 02:06:26.
Port /dev/ttyACM0, 12:15:17
Tapez CTRL-A Z pour voir l'aide concernant les touches spéciales
root@stm32mp1:~#
```

Etape 7 : Vérifier si l'image Linux est bien sur le MP1

```
MP1 $ cd /boot/
```

```
MP1 $ ls
```

Etape 7 : Synchroniser les partitions

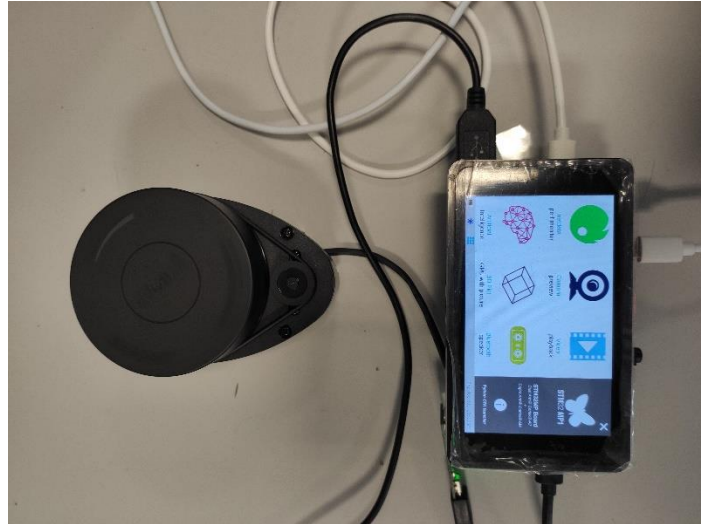
```
MP1 $ sync
```

Etape 8 : Reboot la carte

```
MP1 $ systemctl reboot
```

Vérification de l'installation – Test du Lidar

Etape 1 : Brancher le lidar au STM32MP1



Etape 2 : Vérifier si le device ttyUSB0 est reconnue par le STM32MP1

PC \$ cd /dev

PC \$ ls

```
root@stm32mp1:/dev# ls
autofs          i2c-2          null           ram7           tty19          tty41          tty7           vcsa1
block           initctl        ptmx           ram8           tty2           tty42          tty8           vcsa2
bus             input          ptp0           ram9           tty20          tty43          tty9           vcsa3
cec0            kmsg           pts            random          tty21          tty44          ttySTM0        vcsa4
char            log            ptyp0          rfkill          tty22          tty45          ttyUSB0        vcsa5
console         loop-control   ptyp1          rtc             tty23          tty46          ttyp0          vcsa6
cpu_dma_latency loop0           ptyp2          rtc0            tty24          tty47          ttyp1          vcsa7
disk            loop1          ptyp3          serial          tty25          tty48          ttyp2          vcsu
dri             loop2          ptyp4          shm             tty26          tty49          ttyp3          vcsu1
fd              loop3          ptyp5          snd             tty27          tty5           ttyp4          vcsu2
full            loop4          ptyp6          stderr          tty28          tty50          ttyp5          vcsu3
fuse            loop5          ptyp7          stdin           tty29          tty51          ttyp6          vcsu4
gpiochip0       loop6          ram0           stdout          tty3           tty52          ttyp7          vcsu5
gpiochip1       loop7          ram1           tty             tty30          tty53          ubi_ctrl       vcsu6
gpiochip2       mem            ram10          tty0            tty31          tty54          uinput         vcsu7
gpiochip3       mmcblk0        ram11          tty1            tty32          tty55          urandom        watchdog
gpiochip4       mmcblk0p1      ram12          tty10           tty33          tty56          vcs            watchdog0
gpiochip5       mmcblk0p2      ram13          tty11           tty34          tty57          vcs1           zero
gpiochip6       mmcblk0p3      ram14          tty12           tty35          tty58          vcs2
gpiochip7       mmcblk0p4      ram15          tty13           tty36          tty59          vcs3
gpiochip8       mmcblk0p5      ram2           tty14           tty37          tty6           vcs4
gpiochip9       mmcblk0p6      ram3           tty15           tty38          tty60          vcs5
hwrng           mmcblk0p7      ram4           tty16           tty39          tty61          vcs6
i2c-0           mqueue         ram5           tty17           tty4           tty62          vcs7
i2c-1           net            ram6           tty18           tty40          tty63          vcsa
```

L'activation du driver a fonctionné le lidar est bien reconnu.

3. UTILISATION DU SDK DE SLAMTEC

Télécharger le SDK sur votre linux : https://github.com/Slamtec/rplidar_sdk

Déplacer votre rplidar_sdk-master.zip dans :

/home/ «votre home »/STM32MPU_workspace/STM32MP15-Ecosystem-v3.0.0/Developer-Package/stm32mp1-openstlinux-21-03-31/sources

```
PC $ cd /home/ «votre home »/STM32MPU_workspace/STM32MP15-Ecosystem-v3.0.0/Developer-Package/stm32mp1-openstlinux-21-03-31/sources
```

```
PC $ unzip rplidar_sdk-master.zip
```

```
PC $ cd rplidar_sdk-master/sdk/
```

```
PC $ source /home/ «votre home»/STM32MPU_workspace/STM32MP15-Ecosystem-v3.0.0/Developer-Package/SDK/environment-setup-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi
```

```
PC $ $CC --version
```

```
arm-ostl-linux-gnueabi-gcc (GCC) 9.3.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

```
PC $ make
```

```
cd output/Linux/Release/
```

Envoyer sur la mp1 l'exécutable exemple du SDK « simple_grabber » ou « ultra_simple »

```
PC $ scp ultra_simple root@192.168.7.1:/usr/local
```

Connecter vous en minicom à la carte comme vu précédemment et exécuter le fichier « ultra_simple » :

```
MP1 $ cd /usr /local
```

```
MP1 $ ./ultra_simple /dev/ttyUSB0
```

```
theta: 123.34 Dist: 00236.00 Q: 188
theta: 123.72 Dist: 00235.00 Q: 188
theta: 124.07 Dist: 00235.00 Q: 188
theta: 124.50 Dist: 00234.00 Q: 188
theta: 124.85 Dist: 00232.00 Q: 188
theta: 125.28 Dist: 00231.00 Q: 188
theta: 125.64 Dist: 00230.00 Q: 188
theta: 126.06 Dist: 00228.00 Q: 188
theta: 126.48 Dist: 00226.00 Q: 188
theta: 126.90 Dist: 00224.00 Q: 188
theta: 127.33 Dist: 00222.00 Q: 188
```

Votre programme du SDK fonctionne bien sur la stm32MP1 !