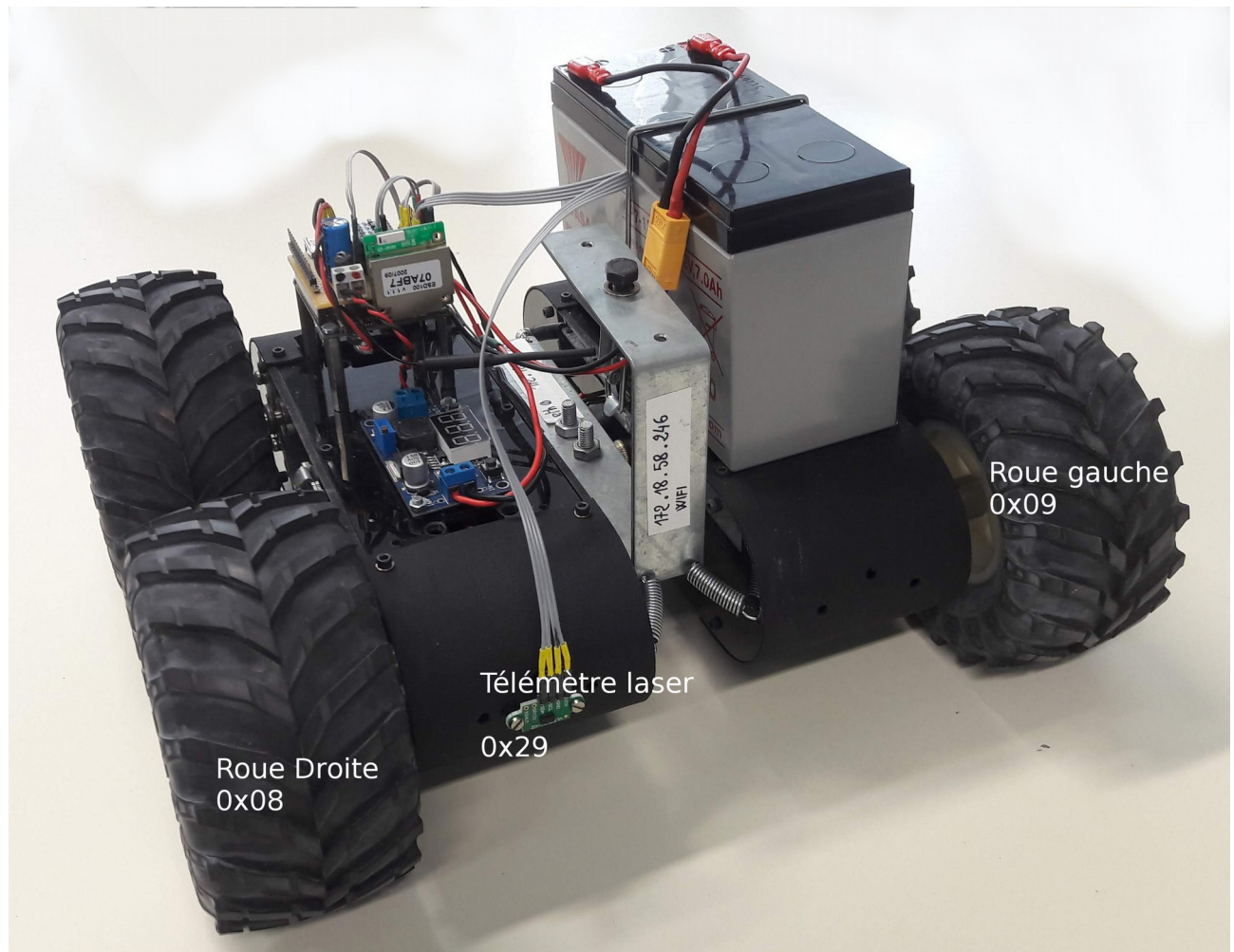


Robot SNIR

Le robot SNIR est une version miniature du mars rover. ce petit robot explorateur va vous faire découvrir de nouveaux horizons et sera votre support pour le challenge.

<http://www.fredzone.org/la-turtle-rover-un-petit-bijou-technologique-020>



Le robot est équipé d'une carte OrangePiZero (compatible avec le raspberry pi).

La connexion peut être établit en SSH via le WIFI à l'adresse **172.18.58.246** ou en Ethernet à l'adresse **172.18.58.245**


L'utilisateur a pour login **pi** et pour mot de passe **raspberrypi**

Le super utilisateur **root** a également pour mot de passe **raspberrypi**

```

psimier@b106tu4p4 ~ $ ssh pi@172.18.58.246
pi@172.18.58.246's password:

```



```

Welcome to ARMBIAN 5.65 stable Debian GNU/Linux 9 (stretch) 4.14.78-sunxi
System load:  0.09 0.03 0.01   Up time:      21 min
Memory usage: 10 % of 493MB   IP:          172.18.58.246
CPU temp:     38°C
Usage of /:   7% of 15G

[ General system configuration (beta): armbian-config ]

Last login: Thu Nov 29 16:24:25 2018 from 172.18.58.191
pi@orangepizero:~$

```

Le Robot est équipé de quatre roues motorisées, qui peuvent être commandées par paire indépendamment. Les moteurs reçoivent leur consigne de vitesse par le biais du bus i2C

```

root@orangepizero:/home/pi# i2cdetect -y 0
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- 08 09 -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- 29 -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
60:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
70:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
root@orangepizero:/home/pi#

```

Comme le montre la copie d'écran, 3 systèmes sont présents sur le bus I2C. Les adresses 08 et 09 correspondent aux systèmes de commande des roues. L'adresse 29 correspond au télémètre laser.

Pour mettre en rotation une paire de roues il suffit d'écrire dans le registre

- **08 pour les roues** situées à droite dans le sens de marche
- **09 pour les roues** situées à gauche

La consigne de vitesse 0 correspond à une vitesse de déplacement nulle et la consigne de vitesse 127 correspond à la vitesse maximale de déplacement.

Le bit de poids fort correspond au sens de rotation.

- **0 pour marche avant**
- **1 pour marche arrière**

Exemple de programme pour la commande des moteurs roues.

```
#include <stdio.h>
#include <stdlib.h>
#include <wiringPiI2C.h>
#include "unistd.h"

int main(int argc, char** argv)
{
    int indice;

    int moteur1 = wiringPiI2CSetup(0x08);
    int moteur2 = wiringPiI2CSetup(0x09);

    wiringPiI2CWrite(moteur1, 0);
    wiringPiI2CWrite(moteur2, 0);

    for (indice = 0; indice < 128; indice++)
    {
        wiringPiI2CWrite(moteur1, indice);
        wiringPiI2CWrite(moteur2, 0x80 | indice);
        sleep(1);
    }
    wiringPiI2CWrite(moteur1, 0);
    wiringPiI2CWrite(moteur2, 0);

    return (EXIT_SUCCESS);
}
```

Pour compiler l'exemple : Voir la documentation netbeans en annexe.

2 Les défis du challenge

Défi n° 1

Réaliser une application avec App Inventor permettant d'envoyer une commande à travers le Bluetooth. (Tester la réception des messages sur orange pi avec minicom).

Défi n° 2

Réaliser une application en langage C capable de recevoir une commande à travers le Bluetooth.

Défi n° 3

Écrire une fonction pour provoquer le déplacement du robot en marche avant et en ligne droite.

Défi n° 4

Écrire une fonction pour provoquer l'arrêt du robot.

Défi n° 5

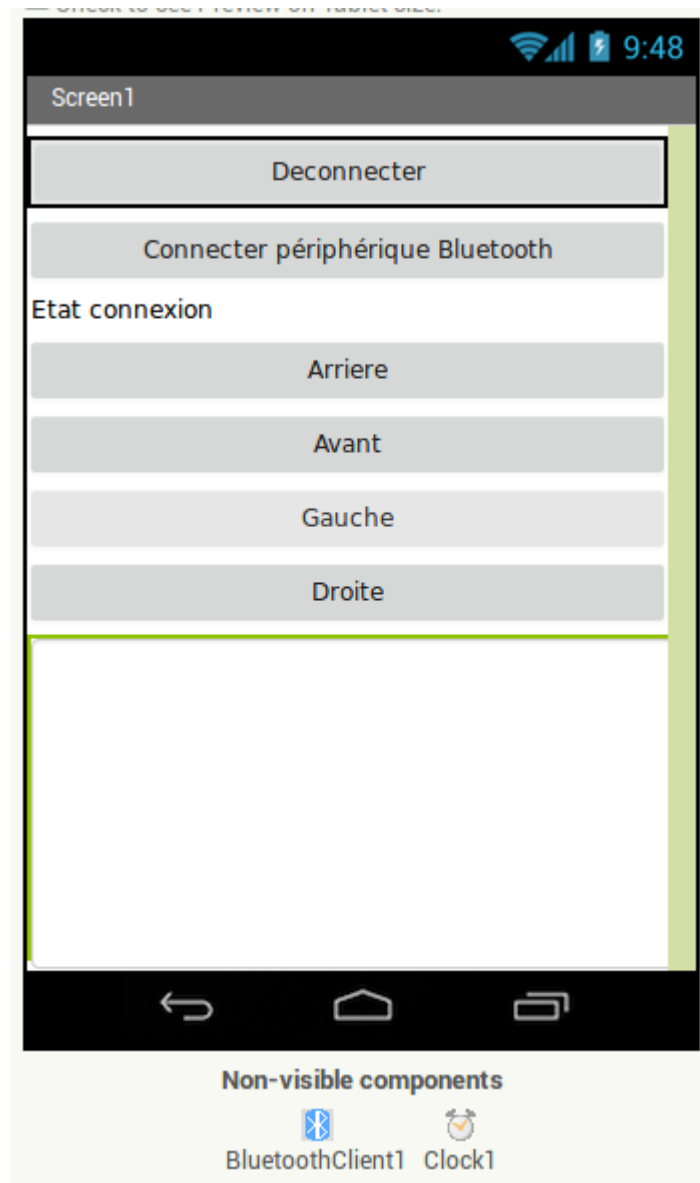
Écrire une fonction pour obtenir une rotation à droite ou à gauche.

Défi n° 6

Écrire une fonction pour obtenir la distance entre l'avant du robot et un obstacle.

Défi n° 7

Écrire un programme pour que le robot puisse être commandé à l'aide d'un téléphone. Les commandes sont obtenues par action sur des boutons.



Défi n° 8

Écrire un programme pour que le robot puisse être commandé à l'aide d'un téléphone. Les commandes sont obtenues par l'inclinaison du téléphone.

Défi n° 9

Ajouter un slider sur l'interface pour donner la consigne vitesse.

Télémètre laser (VL53L00X)

Un télémètre laser est présent à l'avant du robot. Il permet de mesurer la distance (en mm) séparant le robot d'un obstacle).

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>

#include <tof.h> // time of flight sensor library

int main(int argc, char *argv[])
{
    int i;
    int iDistance;
    int model, revision;

    // For Raspberry Pi's, the I2C channel is usually 1
    // For other boards (e.g. OrangePi) it's 0
    i = tofInit(0, 0x29, 1); // set long range mode (up to 2m)
    if (i != 1)
    {
        return -1; // problem - quit
    }
    printf("VL53L0X device successfully opened.\n");
    i = tofGetModel(&model, &revision);
    printf("Model ID - %d\n", model);
    printf("Revision ID - %d\n", revision);

    for (i=0; i<1200; i++) // read values 20 times a second for 1 minute
    {
        iDistance = tofReadDistance();
        if (iDistance < 4096) // valid range?
            printf("Distance = %dmm\n", iDistance);
        usleep(50000); // 50ms
    }

    return 0;
}
```

Annexe 1 : Installation de wiringPi pour orangePi

<https://github.com/zhaolei/WiringOP>

Procédure pour installer wiringPI

```
git clone https://github.com/zhaolei/WiringOP.git -b h3
cd WiringOP
```

```
sudo ./build
```

test de l'installation de la bibliothèque : wiringPi

```
root@orangepizero:/home/pi/WiringOP# gpio readall
```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+												
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+												
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM		
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+												
		3.3v			1	2		5v				
12	8	SDA.0	ALT5	0	3	4		5V				
11	9	SCL.0	ALT5	0	5	6		0v				
6	7	GPIO.7	ALT3	0	7	8	0	ALT3	TxD3	15	13	
		0v			9	10	0	ALT3	RxD3	16	14	
1	0	RxD2	ALT3	0	11	12	0	ALT3	GPIO.1	1	110	
0	2	TxD2	ALT3	0	13	14		0v				
3	3	CTS2	ALT3	0	15	16	0	ALT3	GPIO.4	4	68	
		3.3v			17	18	0	ALT3	GPIO.5	5	71	
64	12	MOSI	ALT3	0	19	20		0v				
65	13	MISO	ALT3	0	21	22	0	ALT3	RTS2	6	2	
66	14	SCLK	ALT3	0	23	24	0	ALT3	CE0	10	67	
		0v			25	26	0	ALT3	GPIO.11	11	21	
19	30	SDA.1	ALT3	0	27	28	0	ALT3	SCL.1	31	18	
7	21	GPIO.21	ALT3	0	29	30		0v				
8	22	GPIO.22	ALT3	0	31	32	0	ALT3	RTS1	26	200	
9	23	GPIO.23	ALT3	0	33	34		0v				
10	24	GPIO.24	ALT3	0	35	36	0	ALT3	CTS1	27	201	
20	25	GPIO.25	OUT	1	37	38	0	ALT5	TxD1	28	198	
		0v			39	40	0	ALT5	RxD1	29	199	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+												
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM		
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+												

```
root@orangepizero:/home/pi/WiringOP#
```

Annexe 2: Installation de la librairie pour le télémètre Laser

<https://github.com/bitbank2/VL53L0X>

```
root@orangepizero:/home/pi# git clone https://github.com/bitbank2/VL53L0X.git
```

```
root@orangepizero:/home/pi# cd VL53L0X/
```

```
root@orangepizero:/home/pi/VL53L0X# make all
```

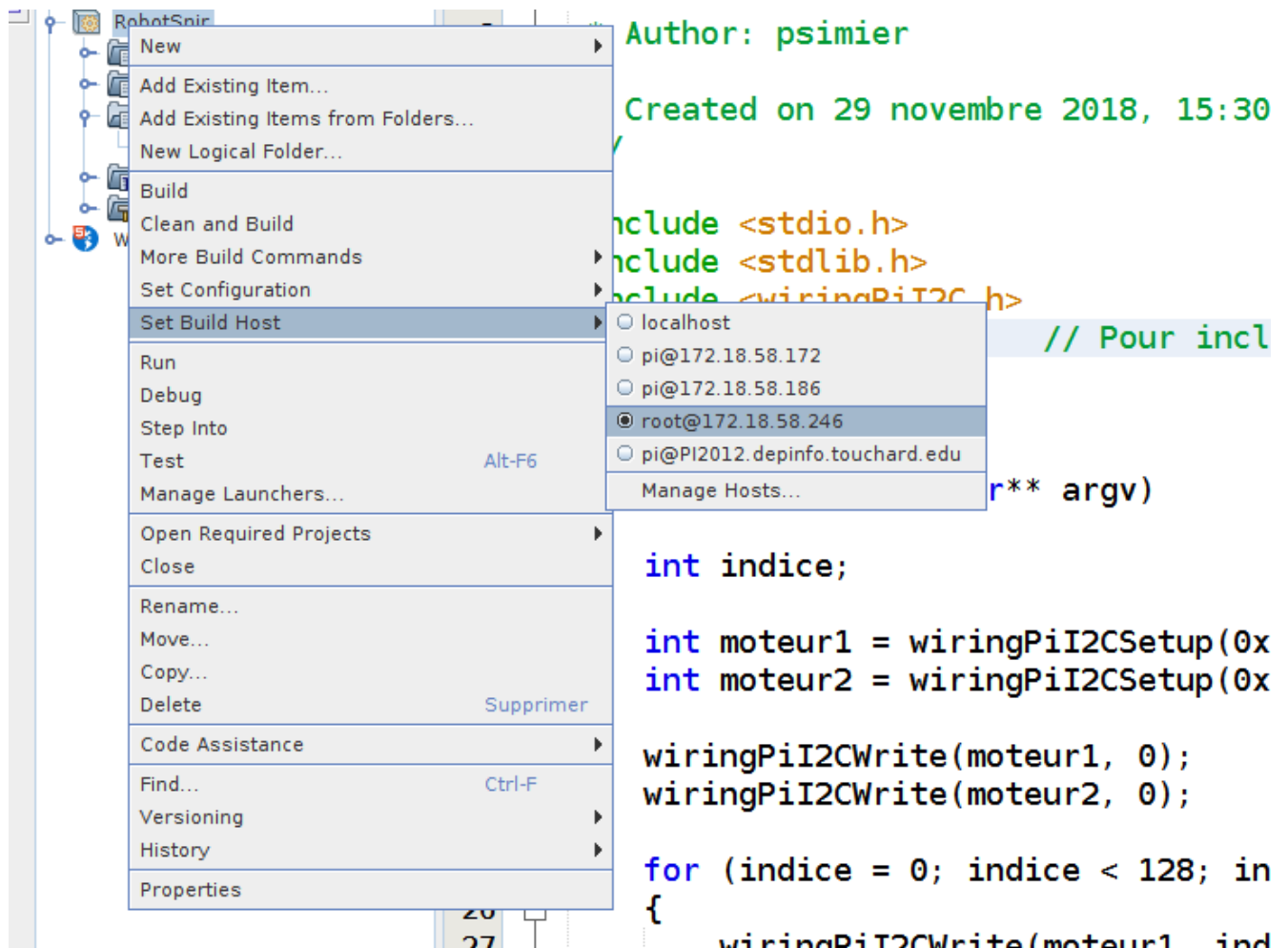
```
root@orangepizero:/home/pi/VL53L0X# make -f make_demo
```

```
test de l'installation de la bibliothèque : tof
```

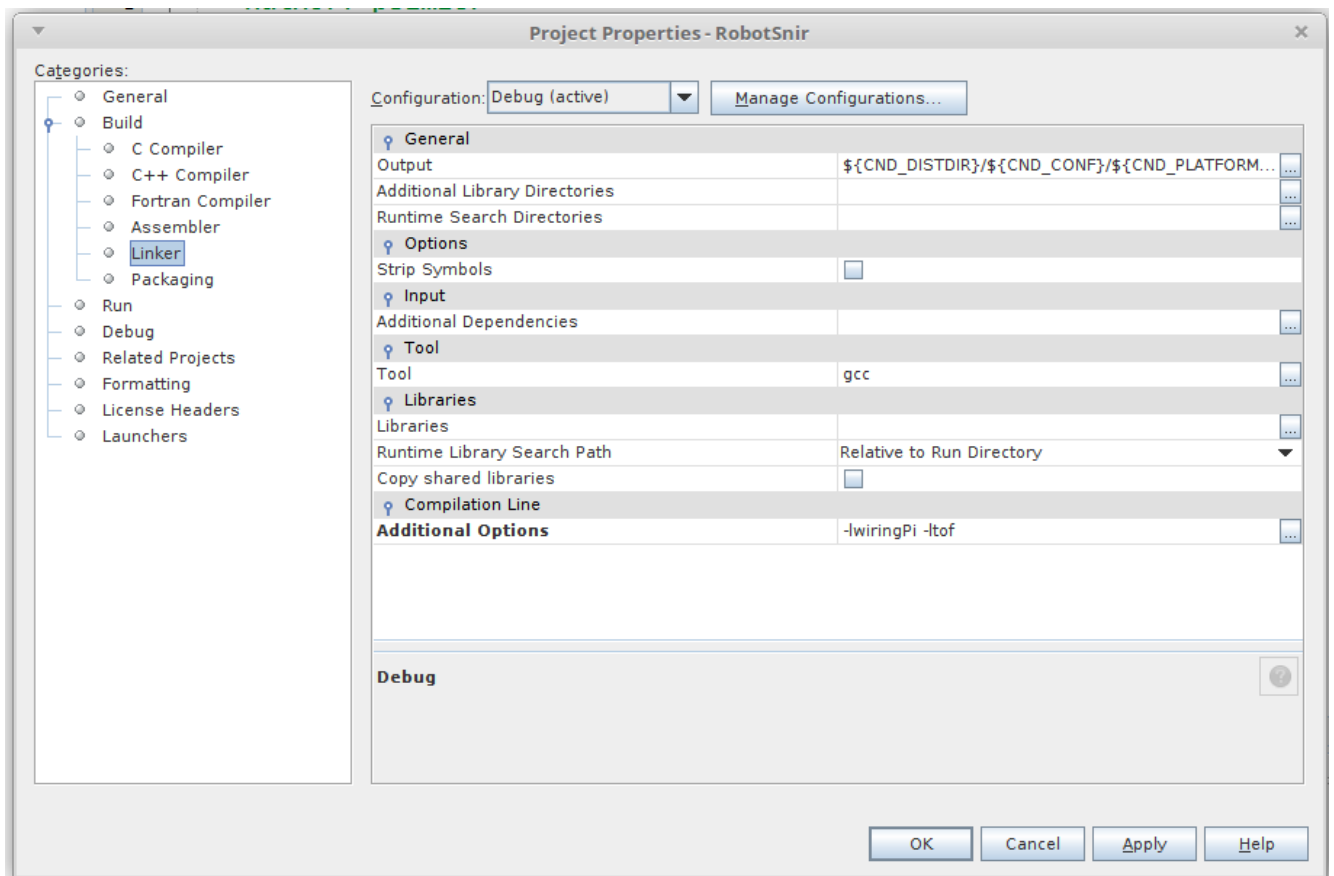
```
root@orangepizero:/home/pi/VL53L0X# ./tofdemo
```


Programmation en C avec Netbeans

Pour programmer le robot vous devez configurer les options suivantes :

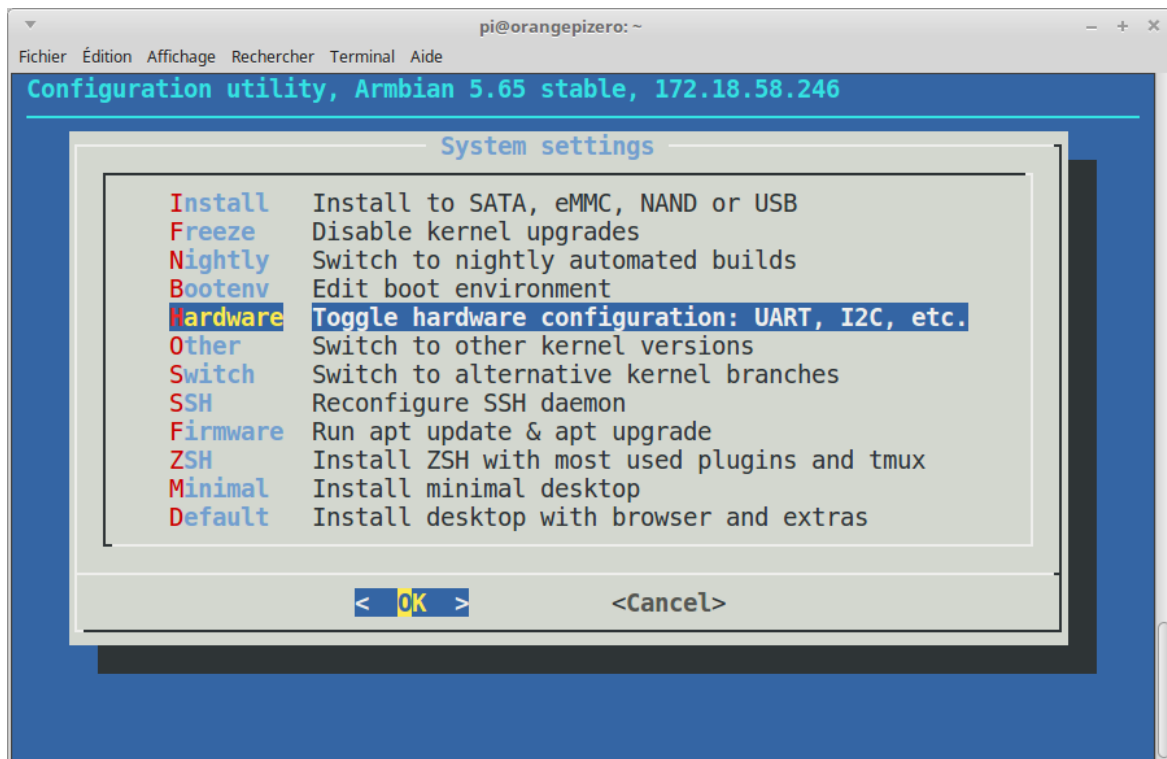


Les bibliothèques WiringPi et tof doivent être linkées. Pour ce faire, clique droit sur le nom du projet puis dans properties

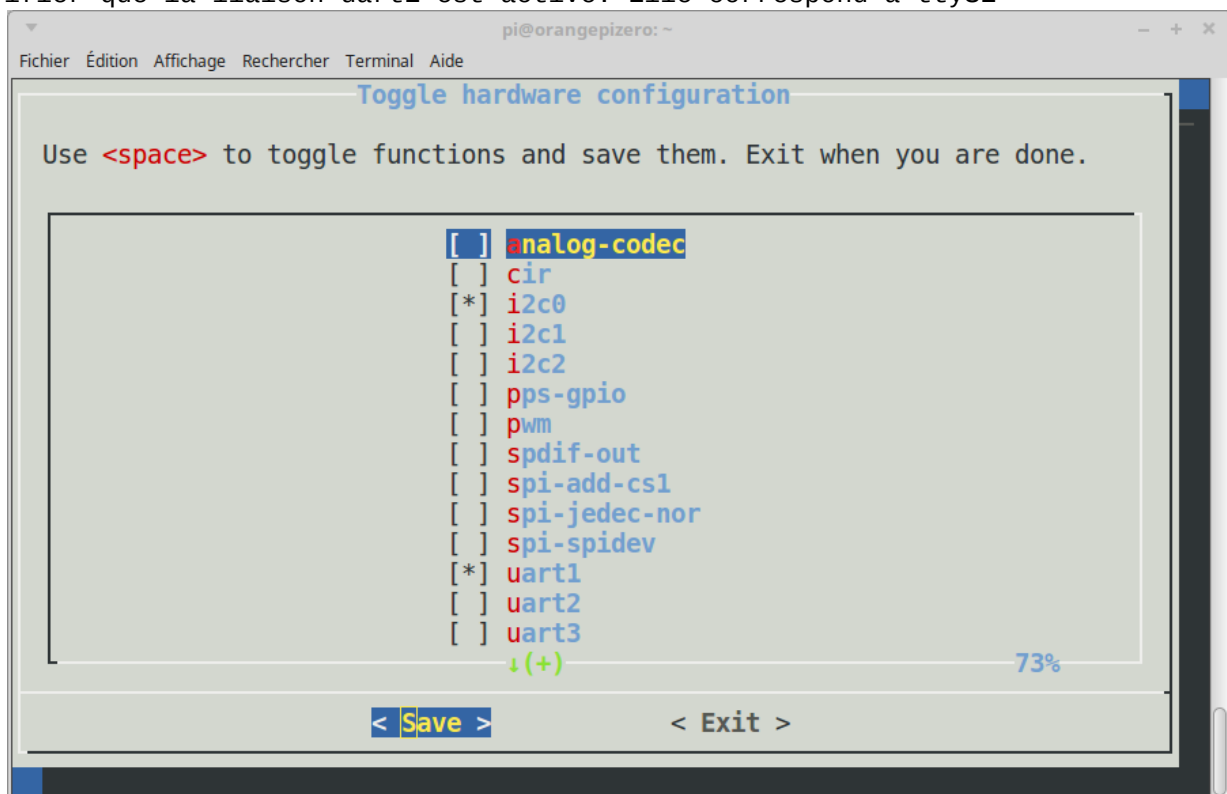


Annexe Liaison série sur orange pi

L'orange pi zero possède plusieurs liaisons série. Pour activée une liaison série lancer la commande **armbian-config** puis sélectionner **Hardware**



Vérifier que la liaison uart1 est active. Elle correspond à ttyS1



Configuration de minicom

lancer la commande `minicom -s`

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup            |
| Modem and dialing            |
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                         |
| Exit from Minicom            |
+-----+-----+

```

Puis configurer la vitesse de communication sur **57600 bps**, Serial Device sur **/dev/ttyS1** pas de contrôle de flux

```
+-----+-----+
| A - Serial Device           : /dev/ttyS1 |
| B - Lockfile Location       : /var/lock   |
| C - Callin Program          :             |
| D - Callout Program         :             |
| E - Bps/Par/Bits            : 57600 8N1  |
| F - Hardware Flow Control   : No          |
| G - Software Flow Control   : No          |
|                               |
| Change which setting? [ ] |
+-----+-----+
| Screen and keyboard          |
| Save setup as dfl             |
| Save setup as..              |
| Exit                         |
| Exit from Minicom            |
+-----+-----+

```

Ne pas oublier de sauvegarder les paramètres avant de quitter. (Save setup as dfl)

Sur votre téléphone Android installer l'application

Serial Bluetooth Terminal

puis activer le bluetooth Le device se nomme **ESD100v1.1.1-07ABF7** le mot de passe est **1234**)

