TP BRAS DARWIN

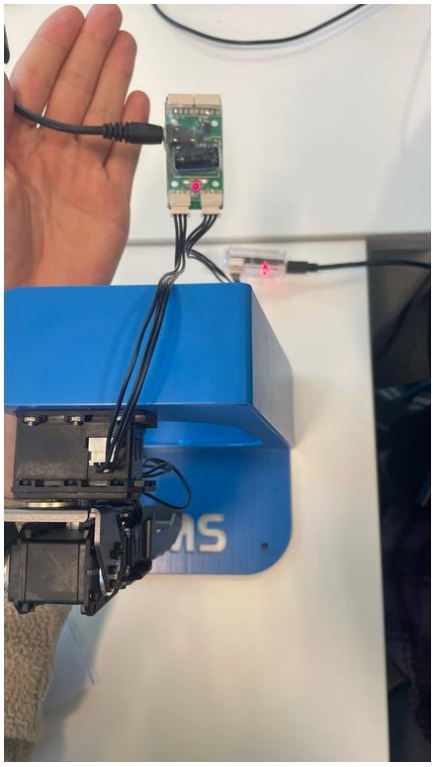
Thomas Pruvost

UIMM

LA FABRIQUE DE L'AVENIR

Procédure de dépannage :

Le robot sera câblé de la manière suivante :



L'exploitant se plaint de dysfonctionnements :

- Les LED des moteurs ne s'éteignent pas
- Le mouvement est trop lent
- L'information affichée sur la console n'est pas assez complète
- Pas de possibilité de piloter par internet

On va tout d'abord constater le problème en exécutant le script Python fourni. Lors de la première exécution, celui-ci ne va fonctionner car il manque des librairies que l'on doit installer :

- ✓ pip install simplelogging
- ✓ pip install pydxl

Une fois cela fait, on peut Exécuter le programme et on constate donc les dysfonctionnements.

Le programme génère un journal d'exécution (log.txt), étudiez le programme, ajoutez les informations sur les moteurs :

Pour faire cela, il faut décommenter les lignes permettant d'afficher les informations du moteur :

```
60    # info_moteur(epaule1)
61    # info_moteur(epaule2)
62    # info_moteur(coude)
```

Puis, il faut les couper/coller après le code qui gère la vitesse des moteurs afin d'avoir les infos utiles comme la vitesse et les Leds qui ne s'éteignent pas :

```
59    log.info("Mise en route des moteurs")
60    epaule1.torque_enable = True
61    epaule2.torque_enable = True
62    coude.torque_enable = True
63
64    log.info("Mise en position initiale")
65    epaule1.goal_position = conversion_degre_position(180)
66    epaule2.goal_position = conversion_degre_position(220)
67    coude.goal_position = conversion_degre_position(260)
68    time.sleep(1)
69    epaule1.moving_speed = 40
70    epaule2.moving_speed = 40
71    coude.moving_speed = 40
72
73    info_moteur(epaule1)
74    info_moteur(epaule2)
75    info_moteur(coude)
```

Confirmez que les LED sont allumées :

- Visuellement
- Avec le journal d'exécution

Proposez une solution pour le problème des LED en analysant le script Python :

Dans le programme Python, lorsque les LEDs sont à l'état « True (1) », cela signifie que le LEDs sont allumées. Quand elles sont à l'état « False (0) » cela signifie qu'elles sont éteintes.

Pour que les LEDs s'éteignent après l'exécution su script, il faut donc changer leur état de True à False :

```
107   log.info("Extinction des LED des moteurs")
108   epaule1.led = True
109   epaule2.led = True
110   coude.led = True
```

```
106   log.info("Extinction des LED des moteurs")
107   epaule1.led = False
108   epaule2.led = False
109   coude.led = False
```

Quelle est la vitesse angulaire ? Trouvez-là à partir des éléments du journal d'exécution :

Regardons sur la doc technique officielle (https://emanual.robotis.com/docs/en/dxl/mx/mx-28/#performance-graph) :

2. 4. 18. Moving Speed (32)

- Join Mode, Multi-Turn mode It is a moving speed to Goal Position.
  0~1023 (0X3FF) can be used, and the unit is about 0.114rpm.
  If it is set to 0, it means the maximum rpm of the motor is used without controlling the speed.
  If it is 1023, it is about 116.62rpm.
  For example, if it is set to 300, it is about 34.2 rpm.
  However, the rpm will not exceed the No Load Speed.
- Wheel Mode It is a moving speed to Goal direction.
  0~2047 (0X7FF) can be used, and the unit is about 0.114rpm.
  If a value in the range of 0~1023 is used, it is stopped by setting to 0 while rotating to CCW direction.
  If a value in the range of 1024~2047 is used, it is stopped by setting to 1024 while rotating to CW direction.
  That is, the 10th bit becomes the direction bit to control the direction.

NOTE : Wheel mode allows to check max rpm. Any values higher than max rpm will not take effect.

On peut voir que pour 1 unité, la vitesse est de 0.114 tr/min et que la vitesse max est que l'on peut monter jusqu'à 1023 unités au maximum (116.62 tr/min). Donc pour calculer notre vitesse angulaire, un simple produit en croix suffit :

Ici, la Vmax de nos moteurs est de 50 tr/min donc :

1	1023	?
0,114	116,62	50

1	1023	438
0,114	116,62	50

Notre vitesse angulaire est donc de 438. Cette valeur sera donc a modifier dans le code afin que les moteurs tournent à leur vitesse maximale :

```
64  log.info("Mise en position initiale")
65  epaule1.goal_position = conversion_degre_position(180)
66  epaule2.goal_position = conversion_degre_position(220)
67  coude.goal_position = conversion_degre_position(260)
68  time.sleep(1)
69  epaule1.moving_speed = 438
70  epaule2.moving_speed = 438
71  coude.moving_speed = 438
```

L'exploitant voudrait piloter son équipement par internet. Étudiez la faisabilité d'ajouter cela au programme Python :

Cela serait faisable en utilisant par exemple Flask qui permettrais de piloter le bras à distance via l'exécution de différents scripts.

Étudiez le protocole de communication avec les servo-moteurs :

Sur la doc technique officielle, nous avons l'indication suivante :

> **NOTE**: MX(2.0) is a special firmware for the DYNAMIXEL MX series supporting the DYNAMIXEL Protocol 2.0. The MX(2.0) firmware can be upgraded from the Protocol 1.0 by using the Firmware Recovery in DYNAMIXEL Wizard 2.0 or R+ Manager.

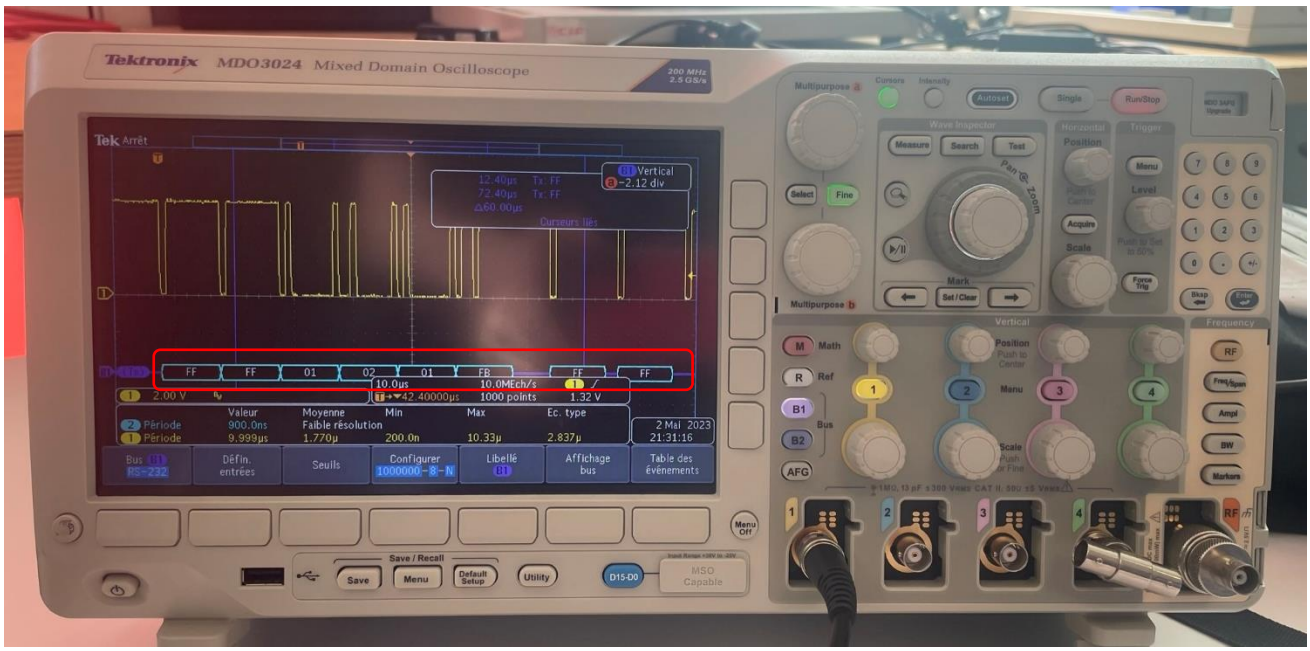
Cela signifie que le protocole Dynamixel peut être en version 1 ou version 2. Si l'on regarde dans le programme Python, nous avons cette ligne :

```
10  log.info("Ouverture de la liaison série (/dev/ttyUSB0)")
11  link = SerialLink(
12      device="/dev/ttyUSB0", baudrate=1_000_000, protocol_version=1.0
13  )
```

Le protocole est donc en version 1.0.

Capturez une trame avec un oscilloscope. Expliquez le contenu de la trame

Voici la trame capturée :



Une fois la trame capturée, il faut effectuer une recherche Google sur le Dynamixel Protocol v1.0.

On se rend ensuite sur la doc officielle (https://emanual.robotis.com/docs/en/dxl/protocol1/). Cette doc va nous fournir toutes les explications nécessaires au décodage de la trame.

Regardons cette partie de la doc :

4. 1. 1. Example

4. 1. 1. 1. Conditions

- ID 1(RX-64) is connected to the PC with an identical baudrate.

4. 1. 1. 2. Ping Instruction Packet

H1	H2	Packet ID	LEN	INST	CKSM
0xFF	0xFF	0x01	0x02	0x01	0xFB

4. 1. 1. 3. ID 1 Status Packet

H1	H2	Packet ID	LEN	ERR	CKSM
0xFF	0xFF	0x01	0x02	0x00	0xFC

On peut voir que notre trame correspond à la trame d'un ping. Qui plus est, si on regarde notre code Python, on peut voir également qu'un ping est effectué lors de la connexion aux moteurs :

```
14  log.info("Connexion aux moteurs")
15  epaule1 = Mx28(identifier=1, serial_link=link)
16  epaule1.ping()
17  epaule2 = Mx28(identifier=2, serial_link=link)
18  epaule2.ping()
19  coude = Mx28(identifier=3, serial_link=link)
20  coude.ping()
```

La trame capturée correspond bien.

Nous souhaitons faire la même procédure de dépannage mais cette fois-ci le programme fourni est en C et non en Python.

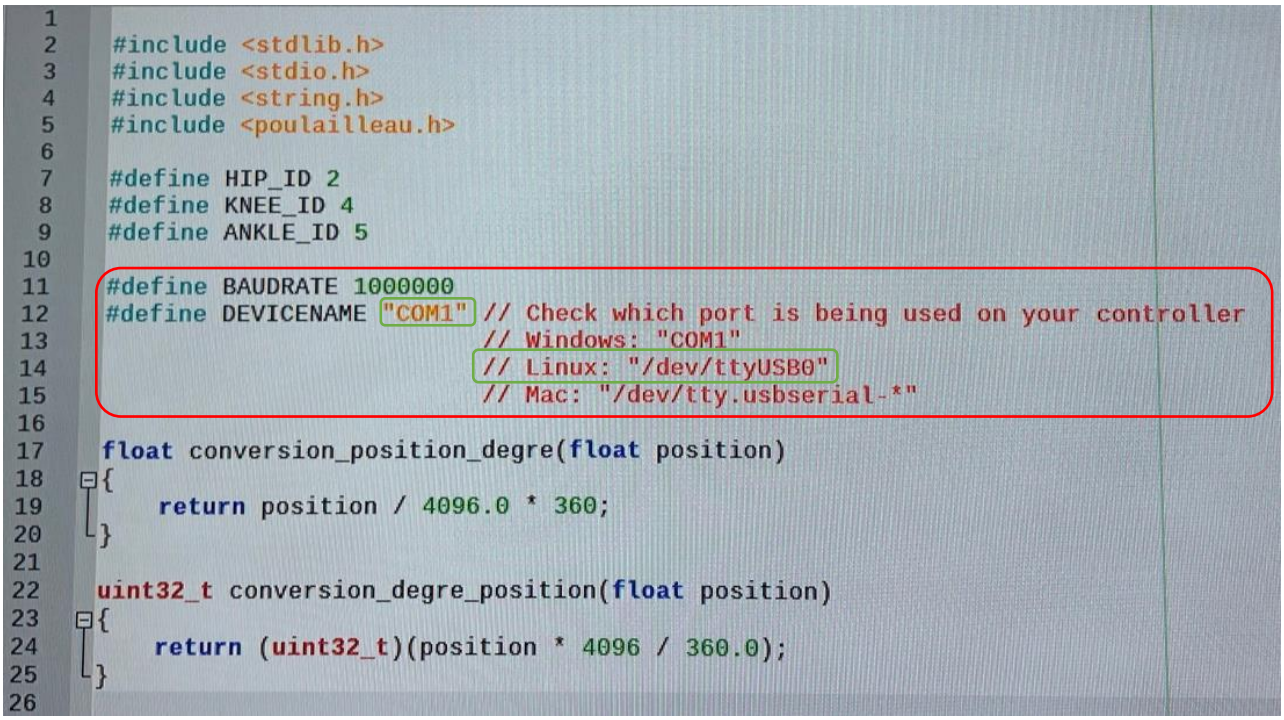
La première étape va être de faire un git clone du répertoire contenant le code.

git clone https://github.com/vpoulailleau/DynamixelSDK

Ensuite, il faut ouvrir le fichier README puis copier/coller le contenu bash (bras ou jambe) dans un fichier.sh pour ensuite pouvoir l'exécuter :

```bash
git clone https://github.com/vpoulailleau/DynamixelSDK
cd DynamixelSDK/c/build/linux_sbc
make # compilation driver
sudo make install # installation driver
cd -
cd DynamixelSDK/c/example/protocol2.0/bras/linux_sbc
make # compilation
./bras # exécution
cd -
cd DynamixelSDK/c/example/protocol2.0/jambe/linux_sbc
make # compilation
./jambe # exécution
```

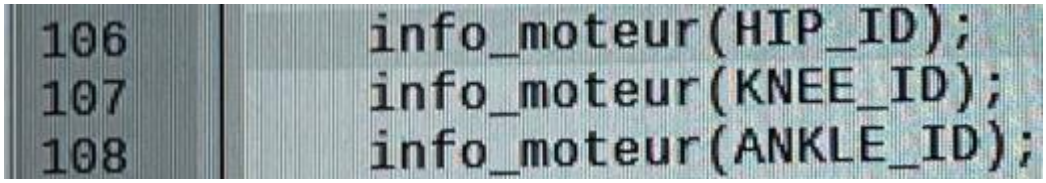
Lors de la première exécution, on peut voir que les ports ne s'ouvrent pas. Il faut donc modifier le fichier .c correspondant pour faire fonctionner le programme (DynamixelSDK/c/exemple/protocol2.0 => jambe.c). Changer le DEVICENAME de « COM1 » qui est de base pour Windows à « /dev/ttyUSB0 » qui est pour Linux



```c

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <poulailleau.h>

#define HIP_ID 2
#define KNEE_ID 4
#define ANKLE_ID 5

#define BAUDRATE 1000000
#define DEVICENAME "COM1" // Check which port is being used on your controller
                          // Windows: "COM1"
                          // Linux: "/dev/ttyUSB0"
                          // Mac: "/dev/tty.usbserial-*"

float conversion_position_degre(float position)
{
    return position / 4096.0 * 360;
}

uint32_t conversion_degre_position(float position)
{
    return (uint32_t)(position * 4096 / 360.0);
}

```

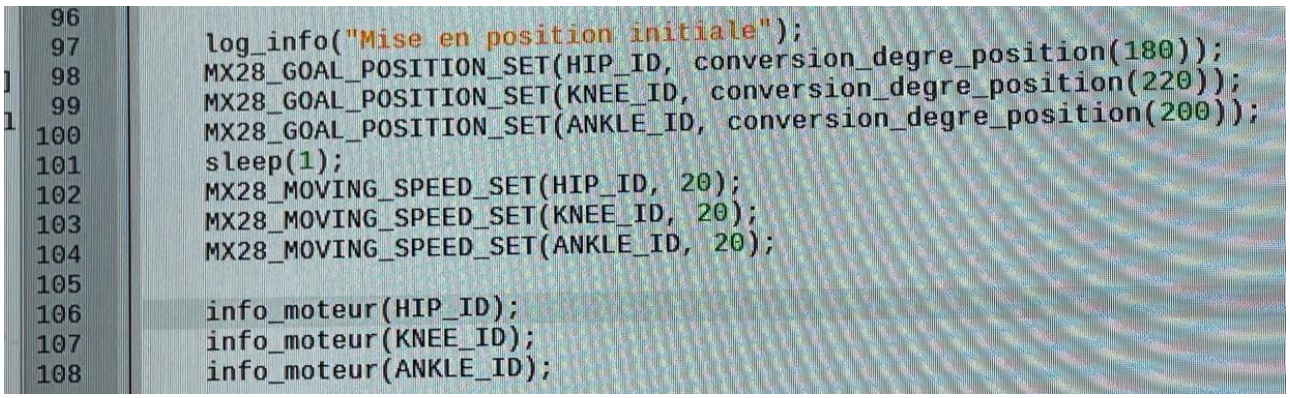
Ajoutez les informations sur les moteurs :

Pour faire cela, il faut décommenter les lignes permettant d'afficher les informations du moteur :



```
106    info_moteur(HIP_ID);
107    info_moteur(KNEE_ID);
108    info_moteur(ANKLE_ID);
```

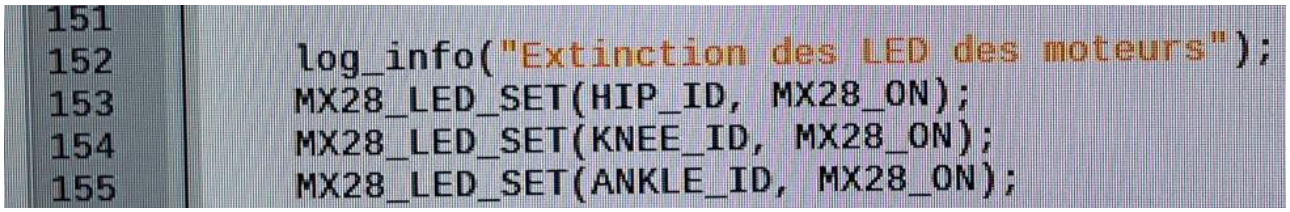
Puis, il faut les couper/coller après le code qui gère la vitesse des moteurs afin d'avoir les infos utiles comme la vitesse et les Leds qui ne s'éteignent pas :



```
96
97     log_info("Mise en position initiale");
98     MX28_GOAL_POSITION_SET(HIP_ID, conversion_degre_position(180));
99     MX28_GOAL_POSITION_SET(KNEE_ID, conversion_degre_position(220));
100    MX28_GOAL_POSITION_SET(ANKLE_ID, conversion_degre_position(200));
101    sleep(1);
102    MX28_MOVING_SPEED_SET(HIP_ID, 20);
103    MX28_MOVING_SPEED_SET(KNEE_ID, 20);
104    MX28_MOVING_SPEED_SET(ANKLE_ID, 20);
105
106    info_moteur(HIP_ID);
107    info_moteur(KNEE_ID);
108    info_moteur(ANKLE_ID);
```

Proposez une solution pour le problème des LED :

Il faut changer l'état des LEDs de « ON » à « OFF » pour que les LEDs moteur s'éteignent :



```
151
152    log_info("Extinction des LED des moteurs");
153    MX28_LED_SET(HIP_ID, MX28_ON);
154    MX28_LED_SET(KNEE_ID, MX28_ON);
155    MX28_LED_SET(ANKLE_ID, MX28_ON);
```

Quelle est la vitesse angulaire ?

Même chose qu'en Python.

Capturez une trame avec un oscilloscope. Expliquez le contenu de la trame

Même chose qu'en Python.