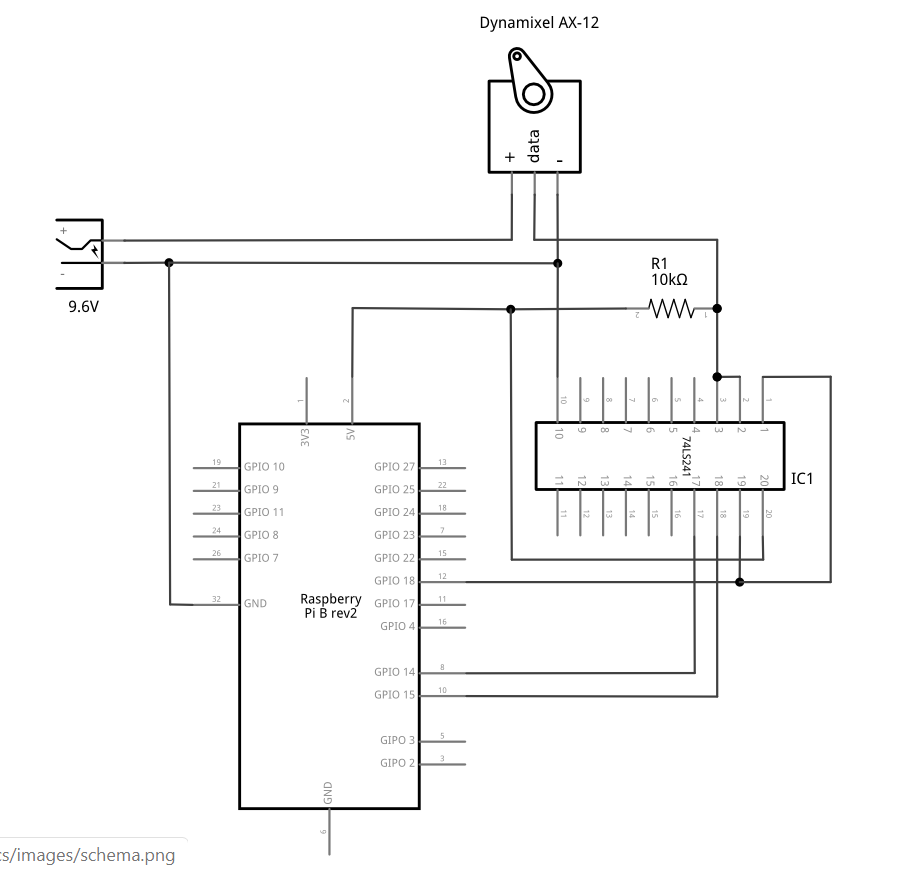
Avancée du contrôle des servomoteurs au 23/02/2020

Tout d’abord voici les e-manuel des 2 servomoteurs pour savoir comment les utiliser (Control Table Description)

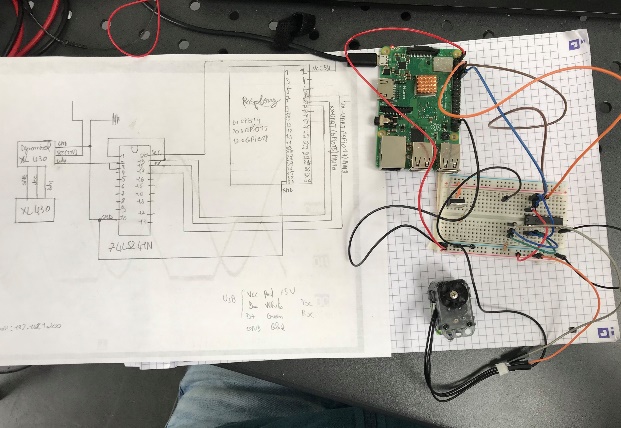
XL320 : <http://emanual.robotis.com/docs/en/dxl/x/xl320/#id>

XL430 : <http://emanual.robotis.com/docs/en/dxl/x/xl430-w250/#control-table-of-eeprom-area>

Et le protocole 2.0 pour les tram de données (la communication) <http://emanual.robotis.com/docs/en/dxl/protocol2/>

**J’ai fait le schéma électronique :**

**Que j’ai récupéré de ce gitHub :** <https://github.com/jeremiedecock/pyax12>



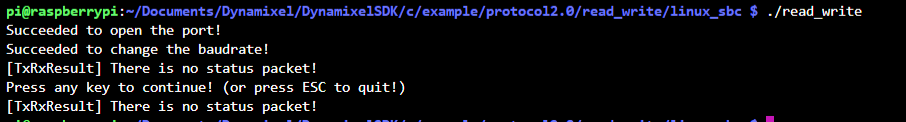
J’ai utilisé Dynamixel\_SDK un kit de développement (une API) pour les servomoteurs dynamixel : <https://github.com/ROBOTIS-GIT/DynamixelSDK>

Le E-manual pour utiliser et installer est ici : <http://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_sdk/overview/>

Pour une raspberry ou carte SBCs linux il faut installer de cette façon pour utiliser les librairies DynamixelSDK : <https://github.com/tonnesfn/DynamixelSDK/wiki/3.2.1.2-C-Linux-(or-Linux-for-SBCs)>

(Cette vidéo est à la fin du e-manual dynamixel sdk)

Ensuite dans les codes (programmes) ce qu’il faut changer c’est le Baudrate , l’ID (254 broadcast et 1 valeur initiale), ou le serial Port (la sortie de la raspberie qui envoie les données RX et TX de l’UART). TtyS0 correspond à la sortie pour les GPIO 15 et 14. TtyAMA0 correspond à un autre truc et ttyUSB0 correspond à la sortie USB.

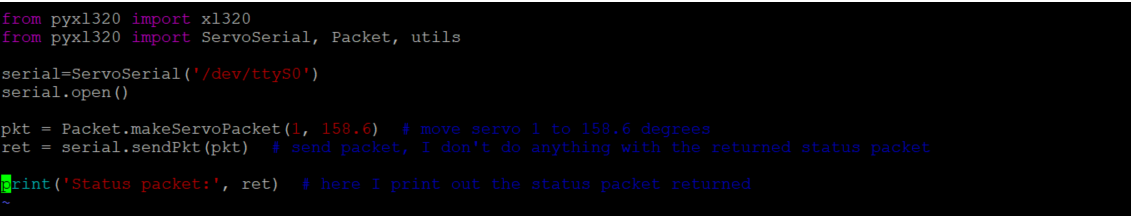
J’ai essayé de lancer des programmes comme (read\_write.py) et l’erreur que je retrouve sur tous est : 

Je suis toujours bloqué ici, j’ai essayé différentes façons d’y parvenir comme d’utiliser l’USB0, de changer le baudrate… Il faut que je test avec l’oscilloscope savoir si j’ai bien des signaux sur mon circuit électronique à la sortie RX/TX de la Raspberry et TTL du servomoteur.

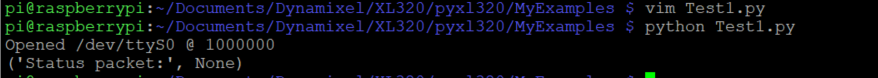
**Restant complètement bloqué à ce point j’ai cherché d’autres méthodes.**

Et j’ai découvert d’autres librairies python pour commander les servomoteurs XL320 et XL430.

Voici la librairie que j’ai téléchargé pour le XL320 : <https://pypi.org/project/pyxl320/> et un exemple sur gitHub : <https://github.com/MultipedRobotics/pyxl320>

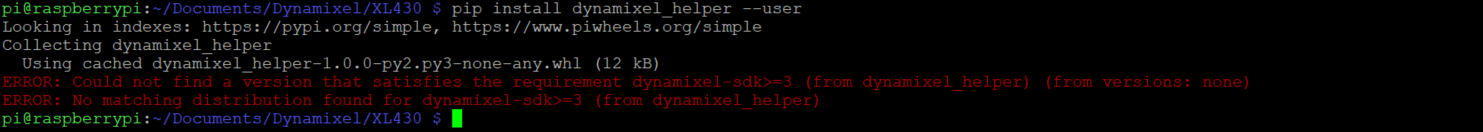
Avec ma faible connaissance de linux, j’ai effectué quelques erreurs, mais j’ai tout de même réussi à écrire un petit programme python qui est sensé faire bouger le servomoteur d’ID=1 à la position 158,6. 

Et j’arrive à cette erreur pour la librairie XL320 quand je lance le programme ci-dessus.



**Restant complètement bloqué à ce point j’ai cherché à contrôler le servomoteur XL430 :**

Il y a une librairie python pour contrôler facilement le XL430 : <https://pypi.org/project/dynamixel-helper/> . Il y aussi tout un github pour dire comment l’utiliser : <https://github.com/ryul1206/easy-dynamixel-helper>. J’ai donc réalisé les premières installations : Pip install dynamixel\_helper –user

Mais dès la première ligne j’ai une erreur :

J’en suis ici avec 3 erreurs différentes, je vais dans un premier temps vérifier avec un oscilloscope que j’envoie bien les données sur les PIN Rx et Tx pour savoir si cela vient de mon petit circuit SN74LS241N ou alors si c’est un problème de software.

Quelques autres liens utiles :

<https://github.com/corgitronics/dynamixel-arm-python/blob/master/arm-1.py>

<https://www.hackster.io/the_marmik/dynamixel-ax12-and-the-raspberry-pi-6536e3>

Avec un mec qui contrôle un bras en utilisant une raspberry mais qui passe de l’USB puis TTL.

<https://www.youtube.com/watch?v=M20P-k8dF4g&feature=youtu.be>

D’autres exemples :

<https://www.okhmer.com/video/M20P-k8dF4g/a-simple-dynamixel-robot-arm>

**. Suite le 28/02/2020 :**

**1-) Préparation** :

Brancher le circuit avec le SN74LS241

Ecrire ce programme ci-dessous dans les exemples avec les librairies XL320 :

* if you want to control AX-12 actuators through GPIO connector, don't forget to set rpi\_gpio to True in the Connection constructor (check the first following example)
* the port value should be "/dev/ttyAMA0" ou ttyS0

Lancer ça dans mes exemples et modifier le code pour utiliser le code si

from pyxl320 import xl320

from pyxl320 import ServoSerial, Packet, utils

import time

import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(18,GPIO.OUT)

serial = ServoSerial('/dev/ttyS0’, baudrate=57600, rpi\_gpio=True) # tell it what port you want to use

# serial = ServoSerial('/dev/tty.usbserial', fake=True) # use a dummy serial interface for testing

serial.open()

while True:

GPIO.output(18,GPIO.HIGH)

pkt = Packet.makeServoPacket(1, 158.6) # move servo 1 to 158.6 degrees

ret = serial.sendPkt(pkt) # send packet, I don't do anything with the returned status packet

time.sleep(0.1)

gpio.output(18, gpio.LOW)

time.sleep(3)

GPIO.output(18,GPIO.HIGH)

pkt = packet.makeLEDPacket(1, pyxl320.XL320\_LED\_GREEN)

ret = serial.sendPkt(pkt)

time.sleep(0.1)

gpio.output(18, gpio.LOW)

time.sleep(3)

print('Status packet:', ret) # here I print out the status packet returned

**Le circuite TTL🡪 UART fonctionne la Trame est bien amplifiée**

**Prochain coup essayer avec bonne tensions d’alimentation pour les servomoteurs et ça devrait fonctionner**