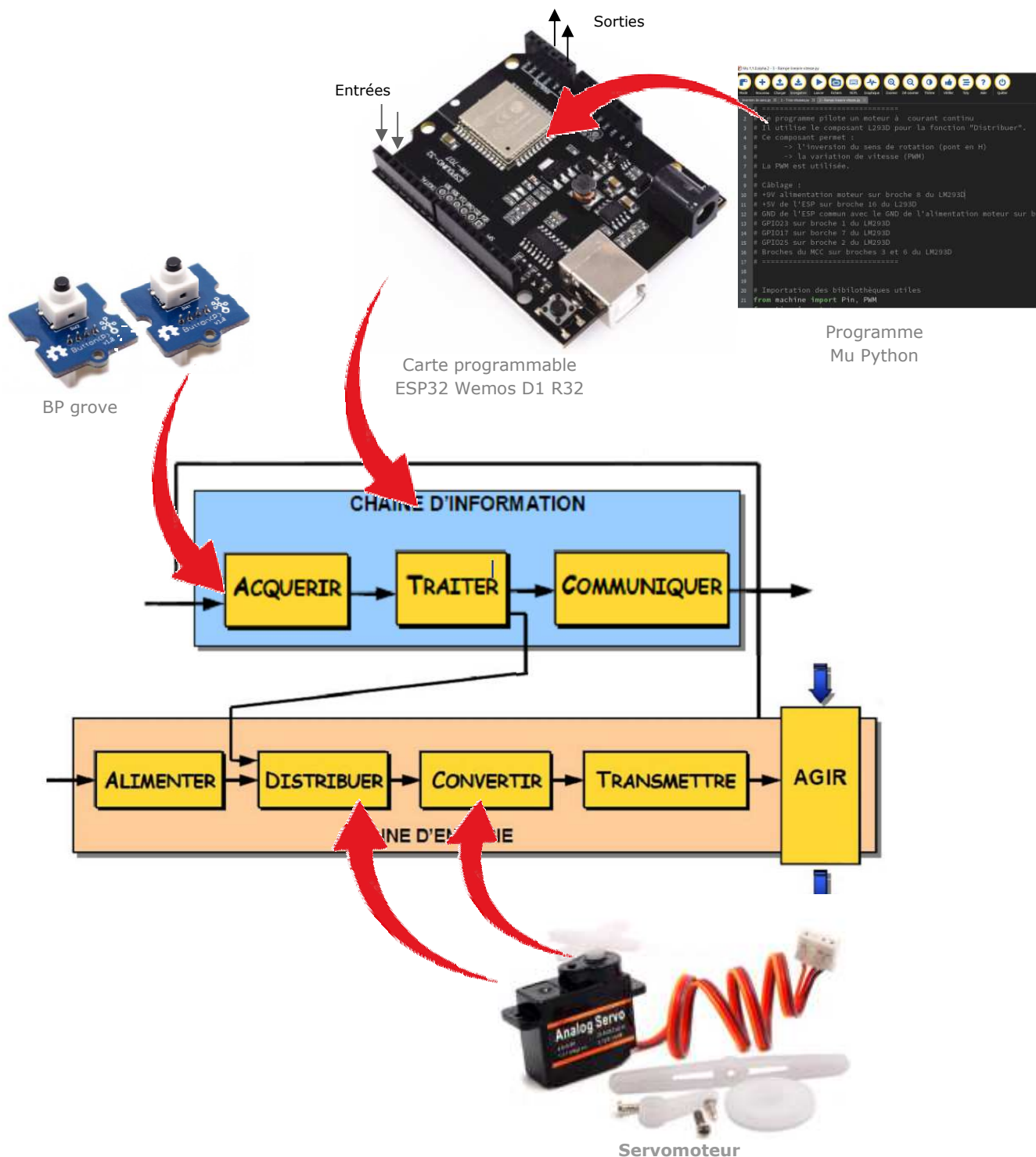




MISE EN ŒUVRE

- **TRAITER** : ESP32 WEMOS (EDI MU)
- **CONVERTIR** : Servomoteur (Moteur et pignons)
- **DISTRIBUER** : Servomoteur (Electronique)

1 – Mise en situation

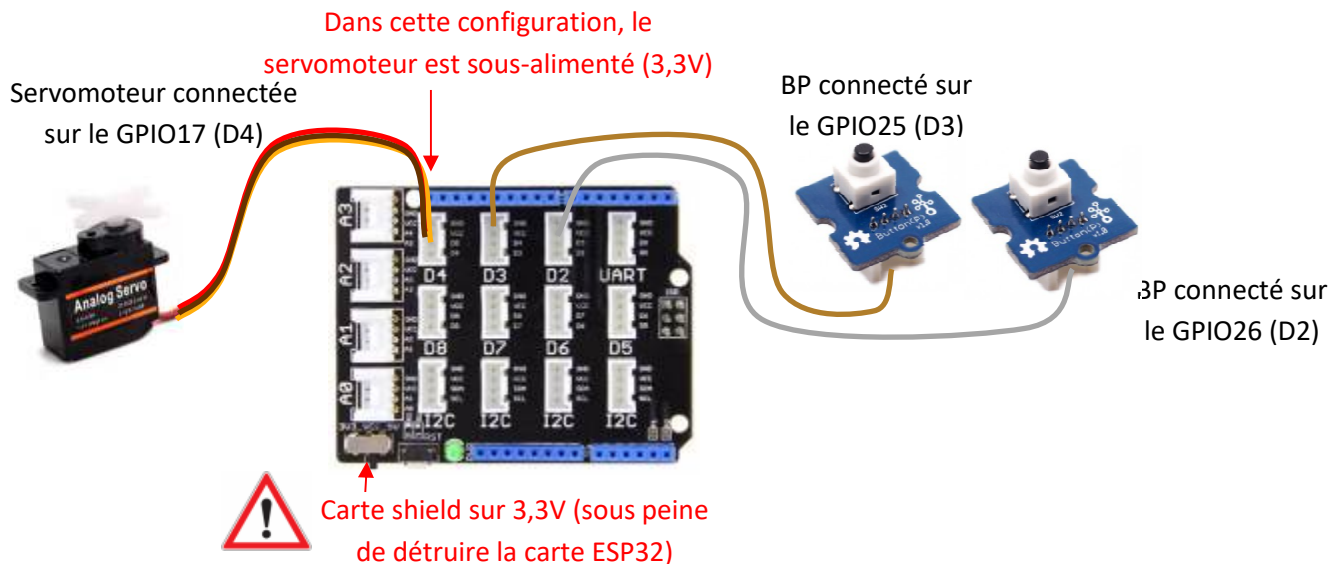


2 – Plan de câblage / Montage

Les servomoteurs de modélisme sont commandés par l'intermédiaire d'un câble électrique à trois fils :

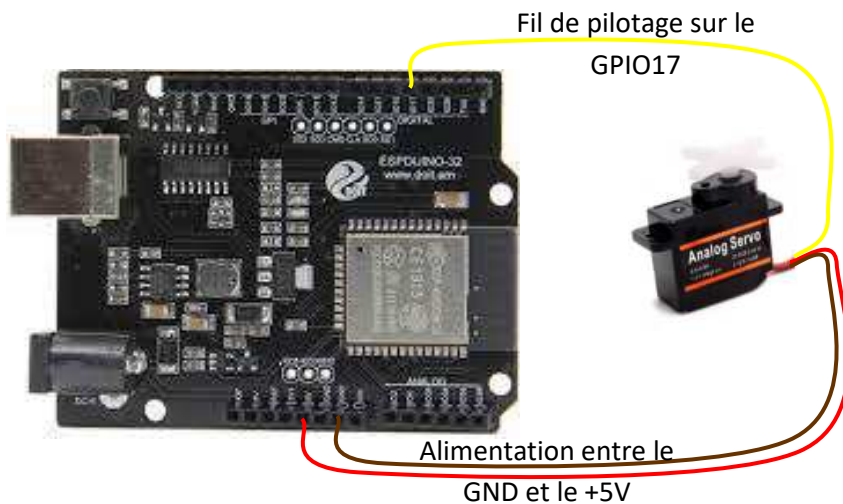
- **Masse (GND)** : Noir ou Marron
- **Alimentation (+)** : Rouge
- **Signal** : Blanc ou Jaune

Alimentation en 3,3V

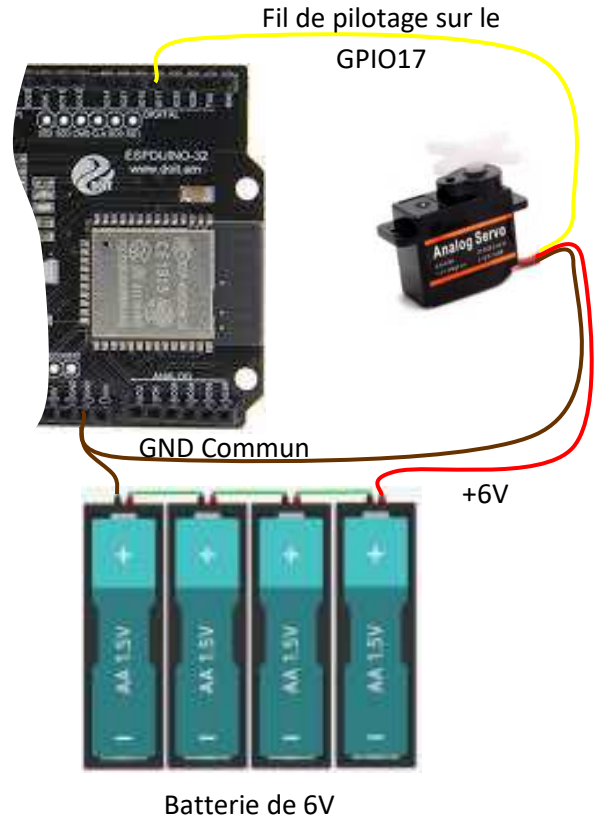


Autres câblages du servomoteur possibles pour éviter la sous-alimentation

Alimentation du servo en 5V



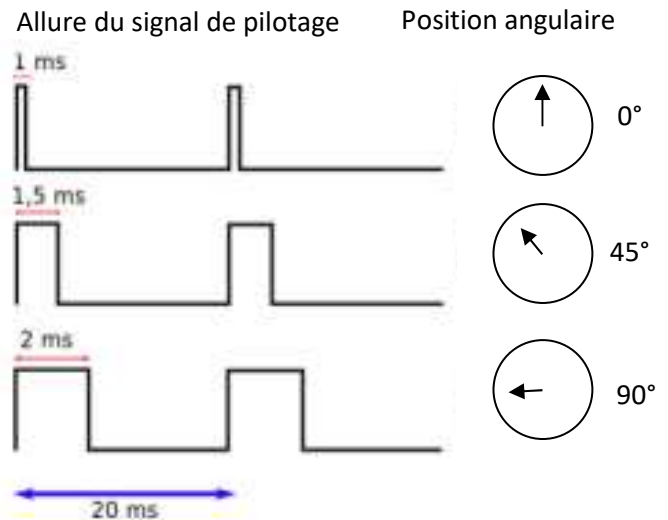
Alimentation du servo en 6V



3 – Programmes

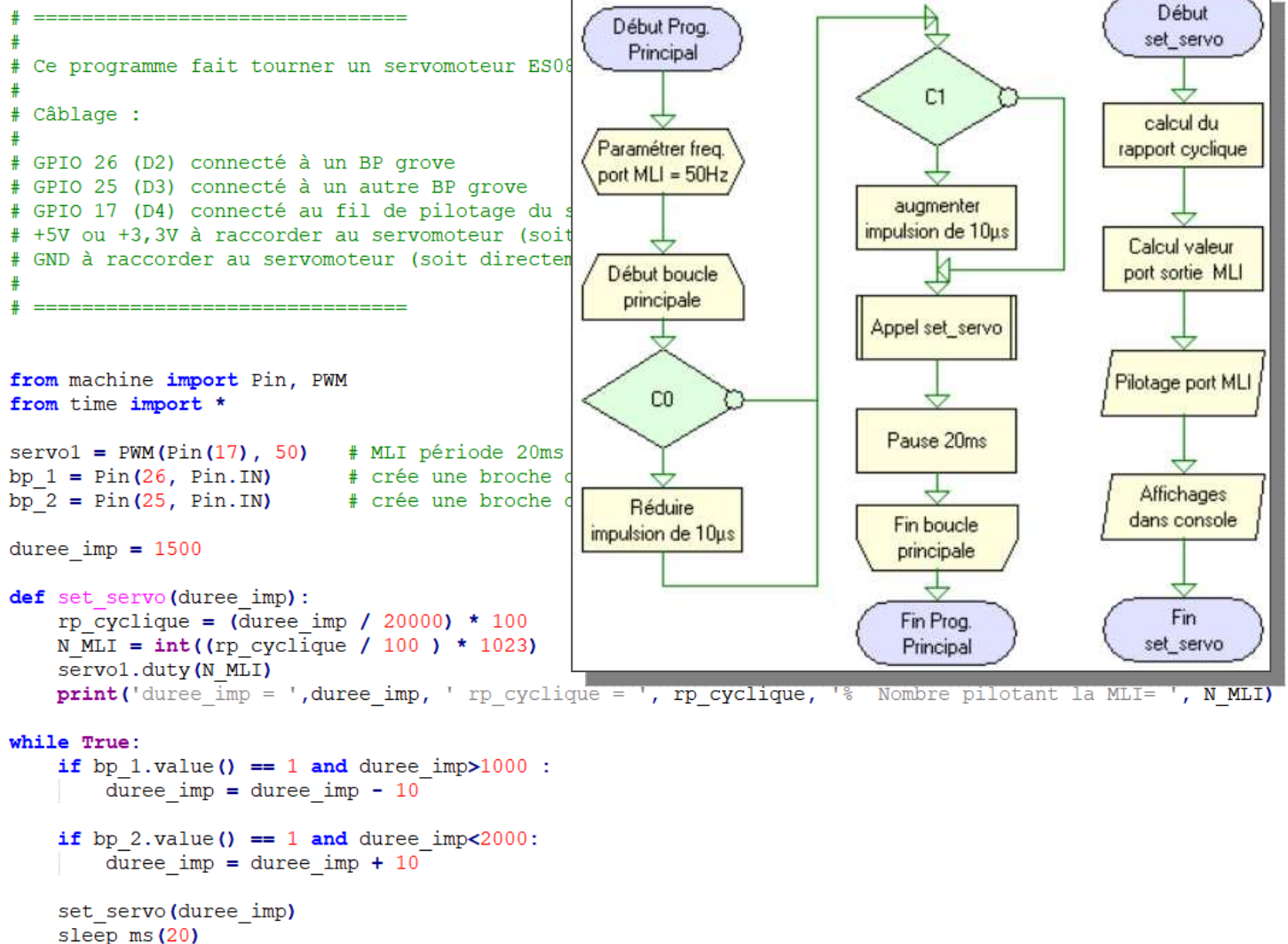
L'électronique intégrée au servomoteur analyse le signal MLI envoyé sur le fil jaune (impulsion de durée 1 à 2 millisecondes). La position angulaire de l'axe de sortie du servomoteur dépend de la largeur de l'impulsion. Le signal est répété périodiquement. C'est un système asservi en position.

Il existe également des servomoteurs dont l'asservissement est en vitesse.



PROGRAMME 1 : « Servo_BP.py »

Ce programme fait tourner l'axe dans un sens ou dans l'autre sens selon l'appui sur un BP ou l'autre.



PROGRAMME 2 : « Servo_BP.py »

Ce programme fait tourner l'axe d'une position angulaire de départ à une position angulaire finale en un certain temps.

```
#
# Ce programme fait tourner un servomoteur ES08A
# dans un sens puis dans l'autre d'un angle à un autre pendant une durée déterminée
#
# Câblage :
#
# GPIO 17 (D4) connecté au fil de pilotage du servomoteur (le pilotage fonctionne avec du 3,3V)
# +5V ou +3,3V à raccorder au servomoteur (soit directement par le connecteur grove soit par fil dupont)
# GND à raccorder au servomoteur (soit directement par le connecteur grove soit par fil dupont)
#
# =====

from machine import Pin, PWM
from time import *

servo1 = PWM(Pin(17), 50) # MLI période 20ms = 50Hz sur broche 17 (D4 shield base 1)

def rp_cy(angle):
    temps_haut = (2000/180) * angle + 500
    rp_cyclique = (temps_haut / 20000) * 100
    return int((rp_cyclique / 100) * 1023)

def servo_deplacement(angle_depart, angle_fin, temps_deplacement_ms):
    delta = angle_fin - angle_depart
    nb_iteration = int(temps_deplacement_ms / 10)
    print('delta=', delta, ' nb_iteration=', nb_iteration)
    angle=angle_depart

    for i in range(nb_iteration):
        angle = angle + (delta / nb_iteration)
        print('angle=', angle, ' i=', i)
        servo1.duty(rp_cy(angle))
        sleep_ms(10)

while True:
    servo_deplacement(50, 140, 1500)
    sleep_ms(2000)
    servo_deplacement(140, 50, 1000)
    sleep_ms(1000)
```

Dé

