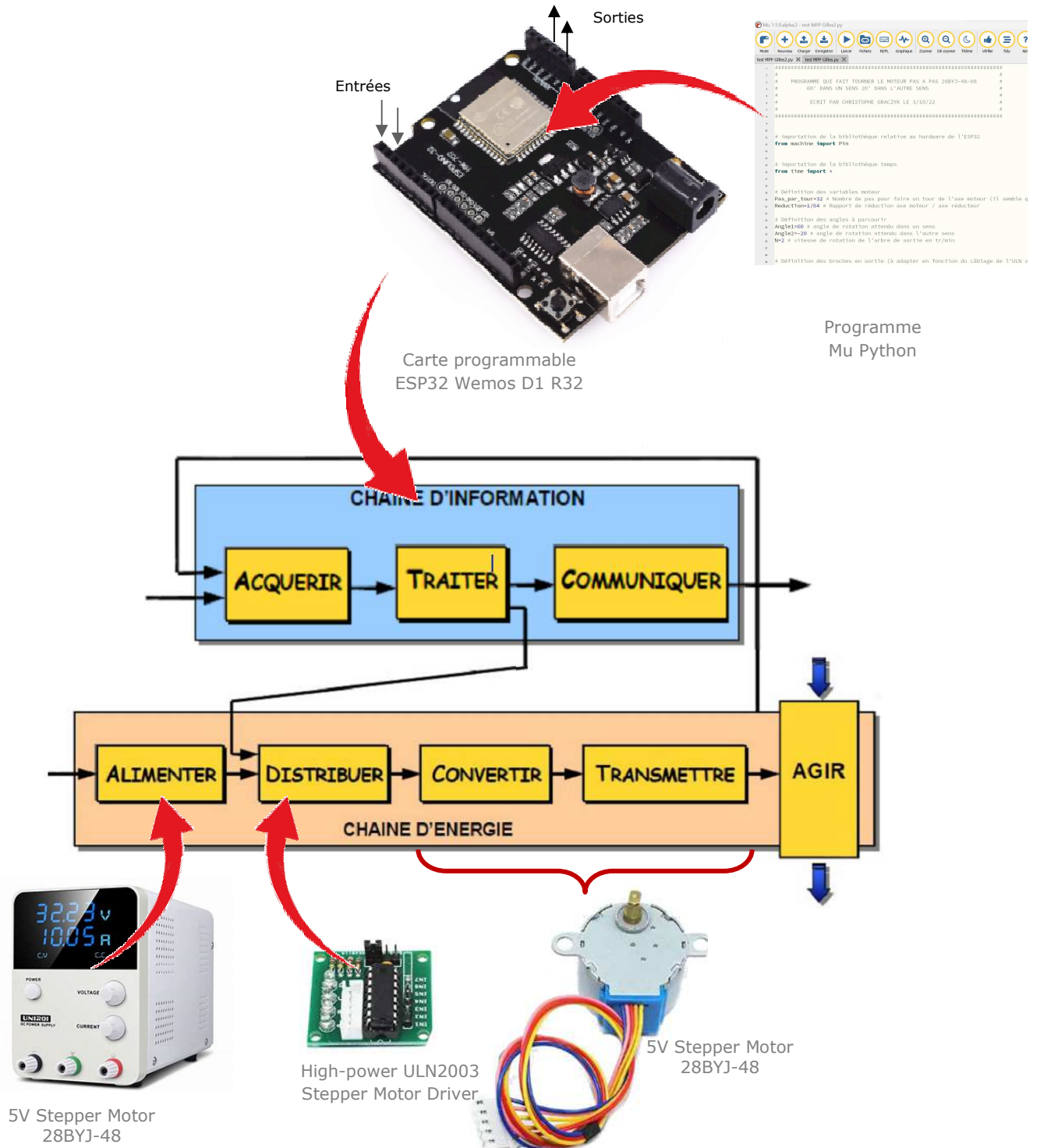




MISE EN ŒUVRE

- **TRAITER** : ESP32 WEMOS (EDI MU)
- **CONVERTIR** : Moteur pas à pas
- **DISTRIBUER** : Stepper motor driver ULN2003

1 – Mise en situation



2 – Vues de détail

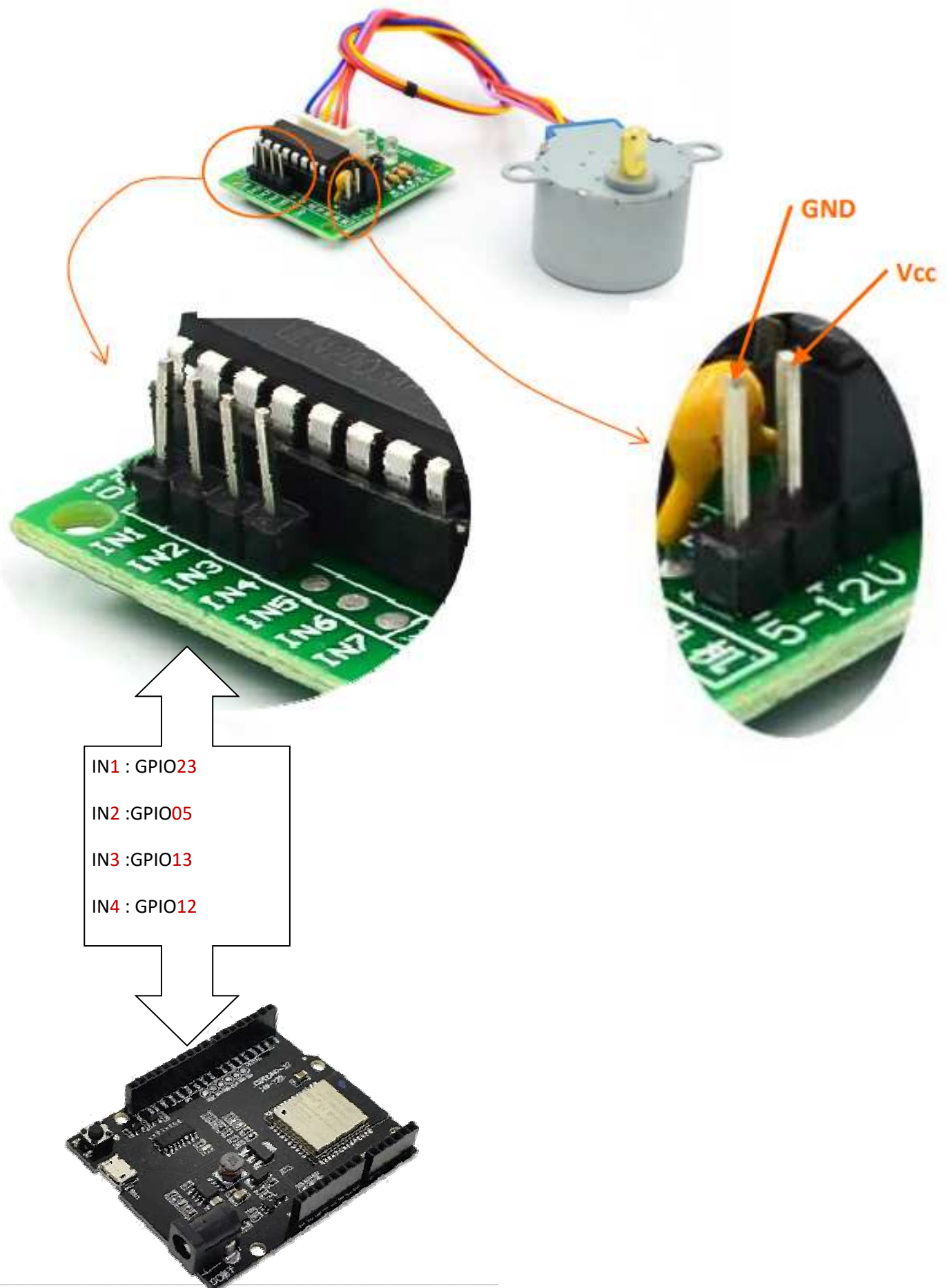
High-power ULN2003 Stepper Motor Driver



5V Stepper Motor 28BYJ-48



3– Plan de câblage / Montage



4 – Programmes

PROGRAMME 1 : « MPP.py »

Ce programme fait tourner le moteur pas à pas d'un angle donné dans un sens puis d'un autre angle donné dans l'autre sens.

L'angle et la vitesse sont paramétrés.

```
1 #####
2 #
3 # PROGRAMME QUI FAIT TOURNER LE MOTEUR PAS A PAS 28BYJ-48-08 #
4 # 60° DANS UN SENS 20° DANS L'AUTRE SENS #
5 #
6 # ECRIT PAR CHRISTOPHE GRACZYK LE 3/10/22 #
7 #
8 #####
9
10
11 # importation de la bibliothèque relative au hardware de l'ESP32
12 from machine import Pin
13
14
15 # importation de la bibliothèque temps
16 from time import *
17
18
19 # Définition des variables moteur
20 Pas_par_tour=32 # Nombre de pas pour faire un tour de l'axe moteur (il semble qu'il y ait un facteur 2 par rapport au D1 : 360°/(5,625°/pas)
21 Reduction=1/64 # Rapport de réduction axe moteur / axe réducteur
22
23 # Définition des angles à parcourir
24 Angle1=60 # angle de rotation attendu dans un sens
25 Angle2=-20 # angle de rotation attendu dans l'autre sens
26 N=2 # vitesse de rotation de l'arbre de sortie en tr/min
27
28
29 # Définition des broches en sortie (à adapter en fonction du câblage de l'ULN sur l'ESP32)
30 IN1 = Pin(23, Pin.OUT) # broche 23 en sortie
31 IN2 = Pin(25, Pin.OUT) # broche 25 en sortie
32 IN3 = Pin(13, Pin.OUT) # broche 13 en sortie
33 IN4 = Pin(12, Pin.OUT) # broche 12 en sortie
34
35
36 # Création d'une fonction qui fait tourner l'axe du moteur d'un angle donnée en fonction de l'angle d'un pas et de la vitesse attendue
37 def rotation(angle, deg_pas, vitesse):
38     if (angle>0):
39         sens=1
40     else:
41         sens=-1
42     T=int(1000*60/vitesse/360*deg_pas) # Calcul de la durée en ms entre 2 impulsions en fonction de la vitesse de rotation attendue
43     print(1/deg_pas)
44     Nb_pas_final = abs(angle/deg_pas)
45     Nb_pas = 0
46     print(sens)
47     print(Nb_pas_final)
48     sleep_ms(2000) # pause
49     while (Nb_pas< Nb_pas_final):
50         Nb_pas=Nb_pas+4
51         print(Nb_pas)
52         if (sens==1):
53             IN4.value(0) # Eteint la phase 4
54             IN1.value(1) # pilote la phase 1
55             sleep_ms(T) # pause
56             IN1.value(0) # Eteint la phase 1
57             IN2.value(1) # pilote la phase 2
58             sleep_ms(T) # pause
59             IN2.value(0) # Eteint la phase 2
60             IN3.value(1) # pilote la phase 3
61             sleep_ms(T) # pause
62             IN3.value(0) # Eteint la phase 3
63             IN4.value(1) # pilote la phase 4
64             sleep_ms(T) # pause
65         else :
66             IN2.value(0) # Eteint la phase 2
67             IN1.value(1) # pilote la phase 1
68             sleep_ms(T) # pause
69             IN1.value(0) # Eteint la phase 1
70             IN4.value(1) # pilote la phase 4
71             sleep_ms(T) # pause
72             IN4.value(0) # Eteint la phase 4
73             IN3.value(1) # pilote la phase 3
74             sleep_ms(T) # pause
75             IN3.value(0) # Eteint la phase 3
76             IN2.value(1) # pilote la phase 2
77             sleep_ms(T) # pause
78
79
80 # Programme principal
81 while True:
82     # calcul du nombre de degré d'axe de sortie par impulsion
83     deg_imp=360/Pas_par_tour*Reduction
84
85     # Appel de la fonction qui fait tourner le moteur
86     rotation(Angle1,deg_imp,N)
87     sleep_ms(2000) # pause
88
89     # Appel de la fonction qui fait tourner le moteur
90     rotation(Angle2,deg_imp,N)
91     sleep_ms(2000) # pause
```