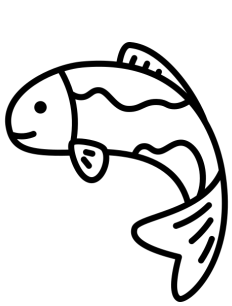
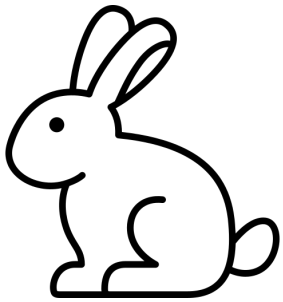
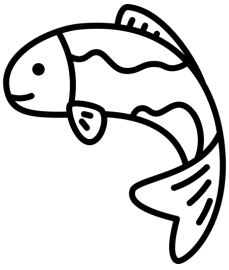
RC5 & moteurs





Dessins : https://www.flaticon.com/fr/auteurs/mihimihi



Codage RC-5

Télécommande InfraRouge

- http://en.wikipedia.org/wiki/RC-5
- http://www.positron-libre.com/electronique/protocole/code-rc5/code-rc5.php
- http://www.oumnad.123.fr/Telecommande-IR/RC5.htm
- http://scan78.free.fr/Elektor/Elektor%202001%20FR/f013024.PDF
- http://www.pcbheaven.com/userpages/The_Philips_RC5_Protocol/
- http://www.sonelec-musique.com/electronique_realisations_alim_led.html

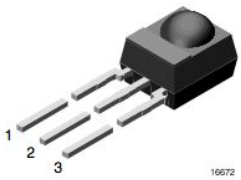
Histoire

C'est en 1956 qu'apparaît la première télécommande à ultrason pour une télévision de Zenith Electronics.

La télécommande à infrarouge apparaît dans les années 1980

Philips développe le protocole RC-5 à la fin des années 1980 pour la commande des appareils électroniques grand public avec le souhait d'une interopérabilité entre les télécommandes et les appareils.

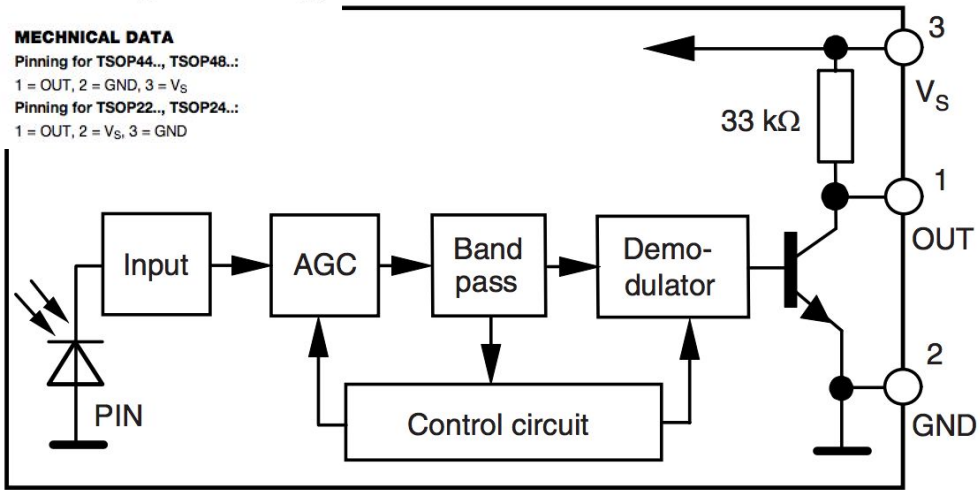
Block Diagram du récepteur TSOP4838



TSOP4838

- Comportement identique au TSOP1736
- VS de 2.5V à 5.5V

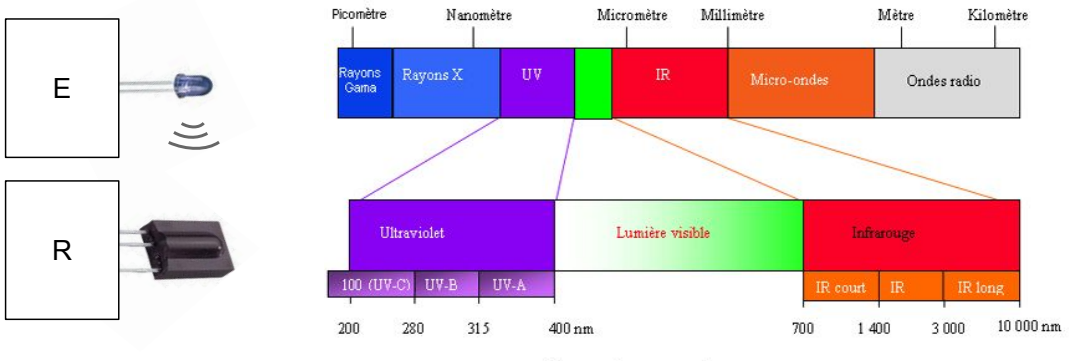
MECHNICAL DATA
Pinning for TSOP44.., TSOP48..:
1 = OUT, 2 = GND, 3 = V_S
Pinning for TSOP22.., TSOP24..:
1 = OUT, 2 = V_S, 3 = GND



Principe

Une diode infrarouge produit une lumière de longueur d'onde de 700 à 950nm.

Un récepteur capte la lumière.



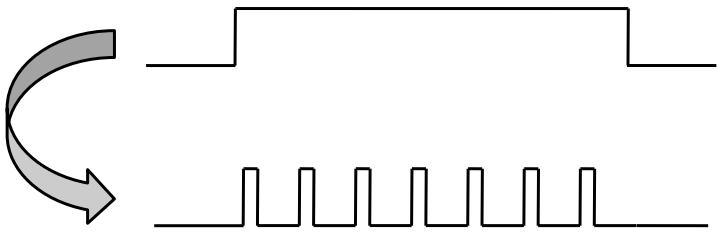
L'information est codée dans une séquence d'allumage et d'extinction (ON-OFF) de la diode infrarouge.

Modulation de l'infrarouge

La chaleur produit aussi un rayonnement infrarouge, il faut :

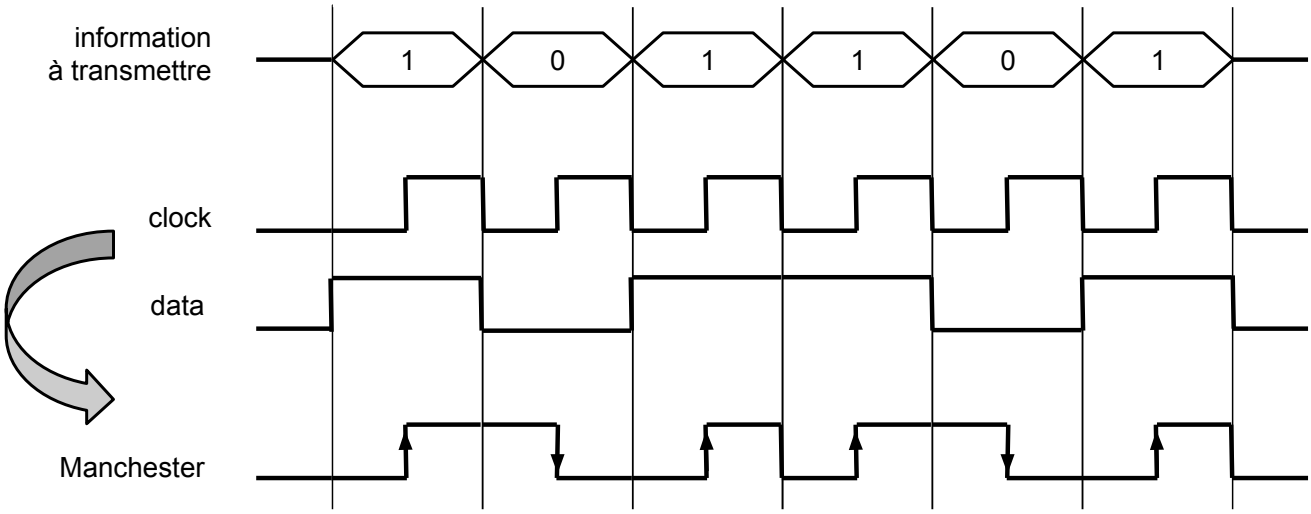
⇒ que le récepteur ne voit que le rayonnement de la diode,
⇒ "allumer la diode" signifie effectuer une séquence de ON-OFF à une fréquence de l'ordre de 38kHz, on parle du modulation

Le récepteur est muni d'un filtre qui ne voit que l'IR modulé à 38kHz.

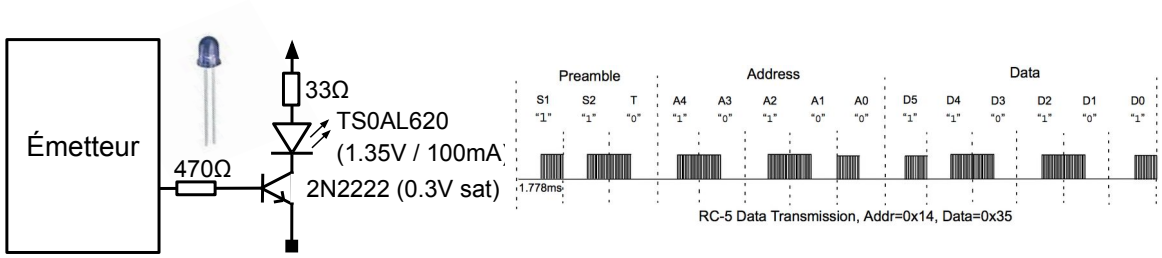


Codage Manchester

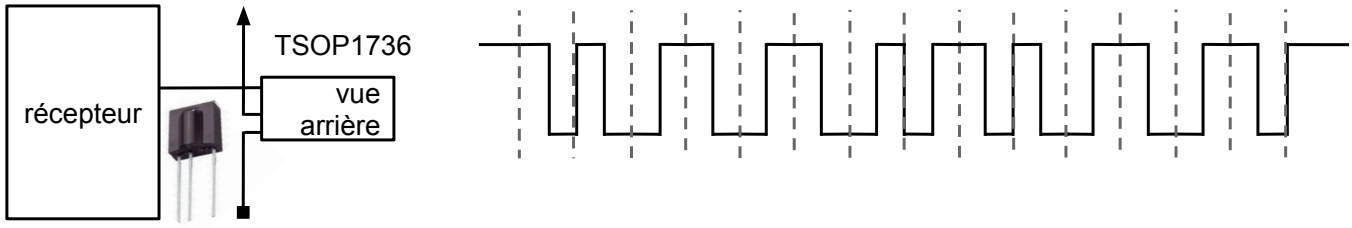
Le signal émis doit contenir la donnée ET l'horloge,
⇒ c'est-à-dire le moyen de distinguer les début et fin de bit



Emetteur - Récepteur RC-5



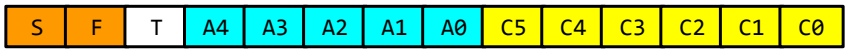
Le processeur reçoit un signal inversé



Codage RC-5

Une trame RC-5 est une suite de 14 bits

- 2 bits de départ Start(1)+Field(1|0)
- 1 bit de basculement
- 5 bits d'adressage
- 6 bits de commandes



Les 2 bits de départ servent à calibrer le gain du circuit intégré de réception.

Le bit de basculement indique une nouvelle transmission de données.

⇒ Sa valeur change à chaque nouvel appui d'une touche pour distinguer une nouvelle pression d'une pression continue sur la même touche.

Les 5 bits suivants déterminent l'adresse du dispositif commandé

⇒ 2^5 = 32 groupes d'adressage (il y a un standard)

La commande destinée à l'appareil est codée dans les 6 derniers bits

⇒ 2^6 = 64 commandes (il y a aussi un standard)

⇒ Version étendue le bit F (Field) code le bit n°6 de la commande 2^{1+6} = 128

Codes RC-5

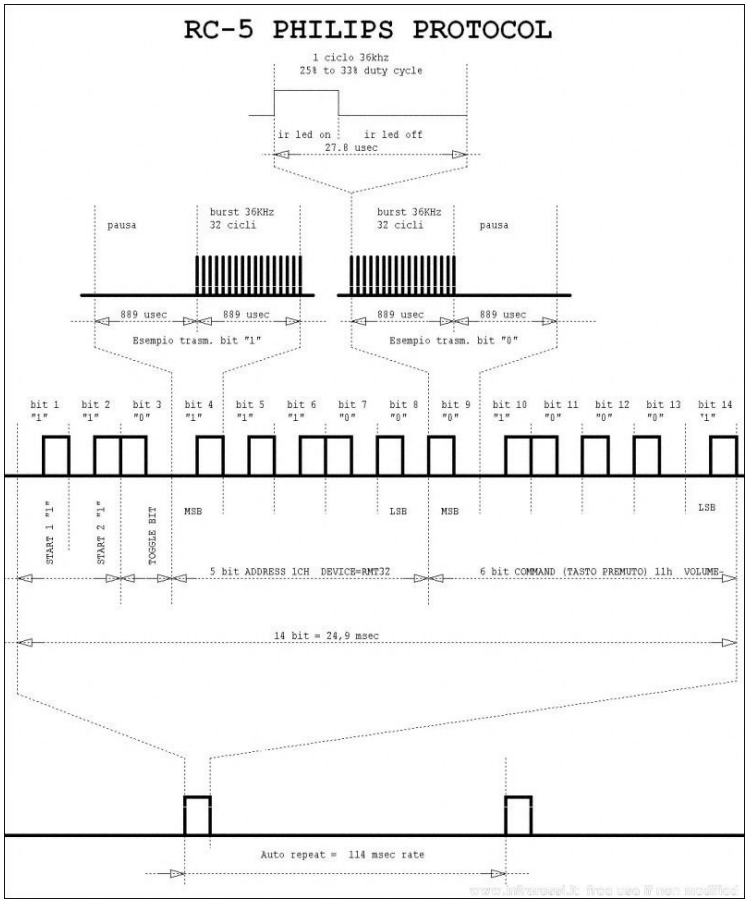
Les codes sont standards afin de permettre l'interopérabilité.

Ce n'est toutefois pas toujours respecté.

RC-5 Address	Device
$00 - 0	TV1
$01 - 1	TV2
$02 - 2	Teletext
$03 - 3	Video
$04 - 4	LV1
$05 - 5	VCR1
$06 - 6	VCR2
$07 - 7	Experimental
$08 - 8	Sat1
$09 - 9	Camera
$0A - 10	Sat2
$0B - 11	
$0C - 12	CDV
$0D - 13	Camcorder
$0E - 14	
$0F - 15	
$10 - 16	Pre-amp
$11 - 17	Tuner
$12 - 18	Recorder1
$13 - 19	Pre-amp
$14 - 20	CD Player
$15 - 21	Phono
$16 - 22	SatA
$17 - 23	Recorder2
$18 - 24	
$19 - 25	
$1A - 26	CDR
$1B - 27	
$1C - 28	
$1D - 29	Lighting
$1E - 30	Lighting
$1F - 31	Phone

RC-5 Command	TV Command	VCR Command
$00 - 0	0	0
$01 - 1	1	1
$02 - 2	2	2
$03 - 3	3	3
$04 - 4	4	4
$05 - 5	5	5
$06 - 6	6	6
$07 - 7	7	7
$08 - 8	8	8
$09 - 9	9	9
$0A - 10	-/--	-/--
$0C - 12	Standby	Standby
$0D - 13	Mute	
$10 - 16	Volume +	
$11 - 17	Volume -	
$12 - 18	Brightness +	
$13 - 19	Brightness -	
$20 - 32	Program +	Program +
$21 - 33	Program -	Program -
$32 - 50		Fast Rewind
$34 - 52		Fast Forward
$35 - 53		Play
$36 - 54		Stop
$37 - 55		Recording

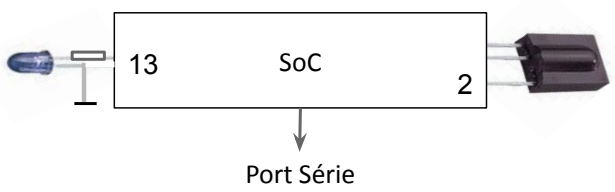
Le document PHILIPS officiel
de décembre 1992



qui résume :

- la modulation de l'IR
- le codage Manchester
- la trame RC-5
- la répétition automatique

Lecture bloquante



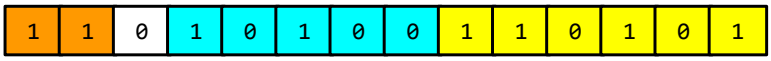

```
int readRC5(byte p)
```

MSB : 1 1 T A4 A3 A2 A1 A0
LSB : 1 F C5 C4 C3 C2 C1 C0

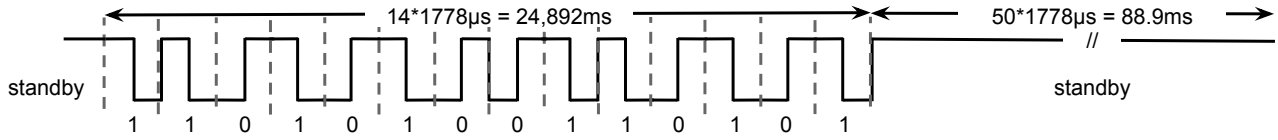
Exemple

Emission

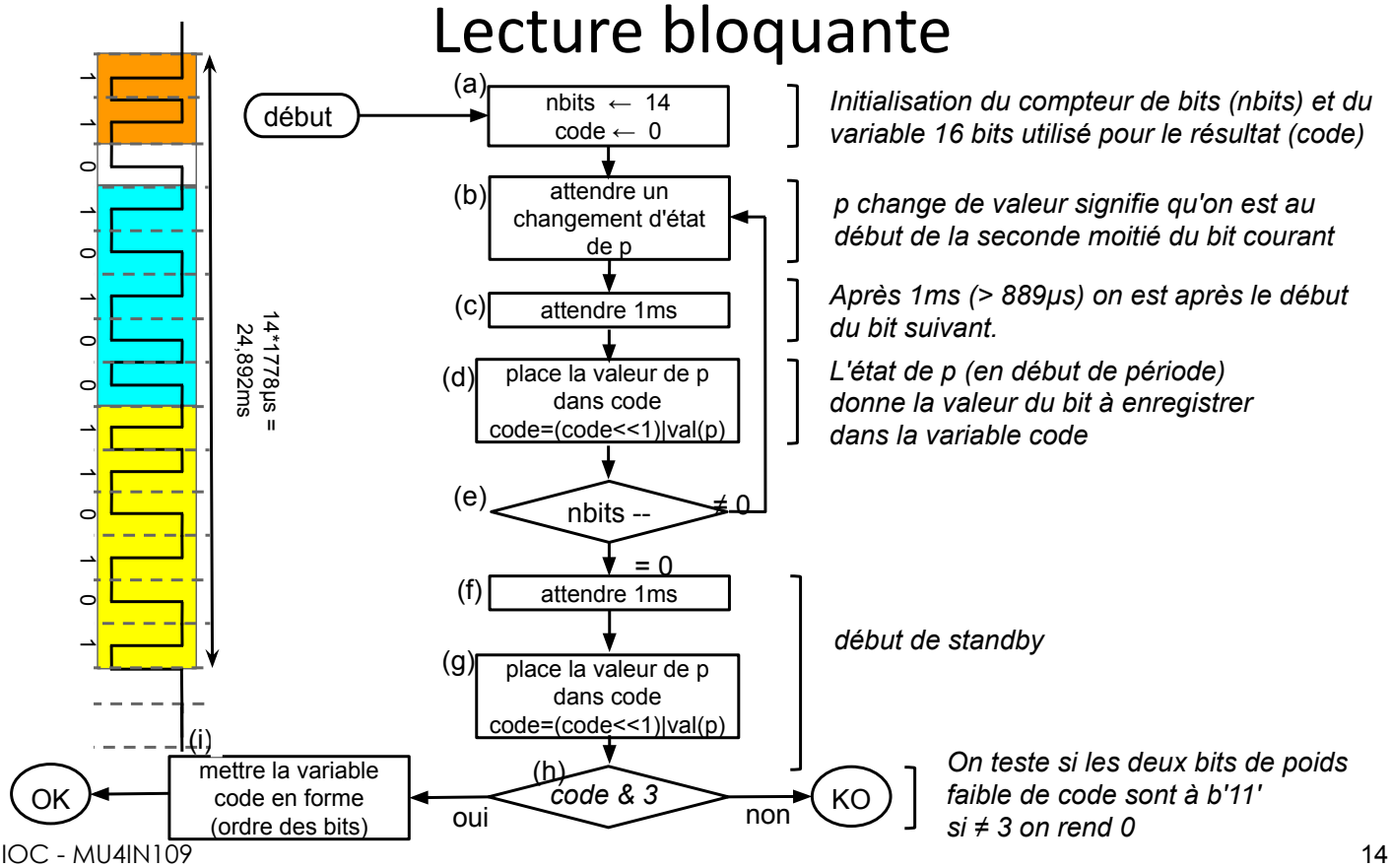
adresse = 0x14, code = 0x35



Réception



Lecture bloquante



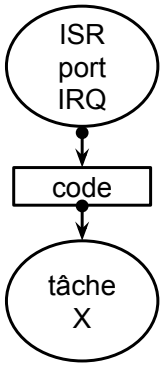
Lecture non bloquante

Le but est de pas bloquer l'exécution de loop pendant la lecture du capteur

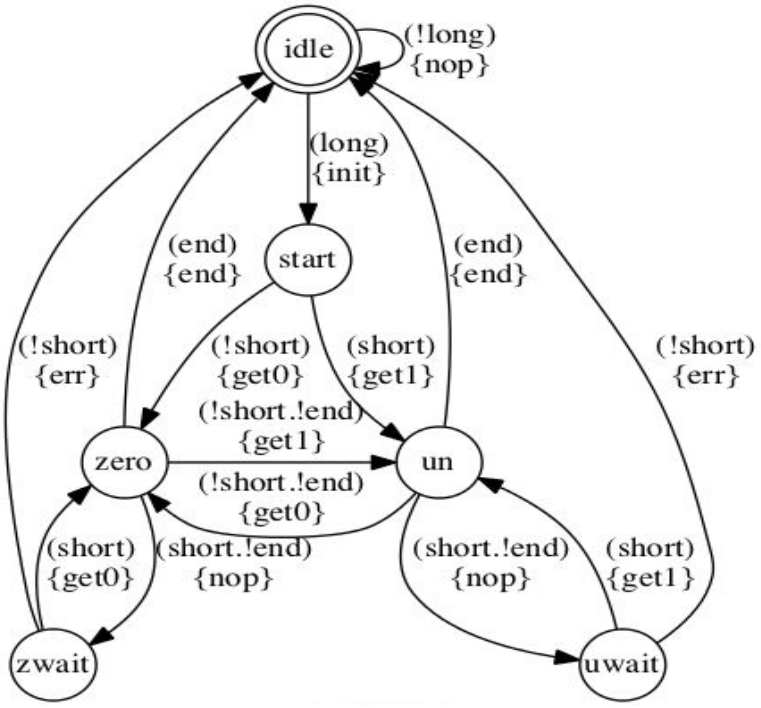
- Le capteur IR est connecté sur une entrée d'interruption
- L'ISR associée va être déclenchée à chaque changement d'état
- et décoder la trame pour l'écrire dans **code**

L'ISR contient la tâche de décodage,

elle reconnaît un '1' ou un '0' en fonction du temps qu'il y a entre deux exécutions en utilisant le compteur micros() qui donne le nombre de microseconde écoulées depuis le démarrage du processeur.



Automate de lecture



INPUTS
long : ((time-begin) > 10ms)
short : ((time-begin) < 1ms)
end : (nbit==14)

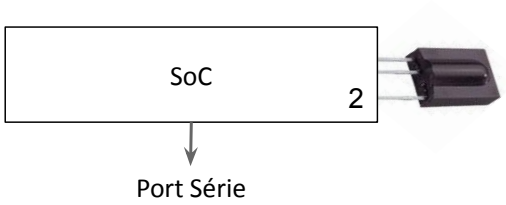
ACTIONS
init : {begin=time; nbit=0; val=0;}
get0 : {begin=time; nbit++; val = val<<1;}
get1 : {begin=time; nbit++; val = val<<1|1;}
nop : {begin=time;}
end : {res=val; flagres=1;}

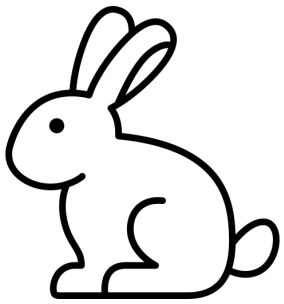
Conclusion

Réutilisation d'une télécommande

1) apprentissage des codes de chaque touche

2) exploitation pour la configuration d'un projet



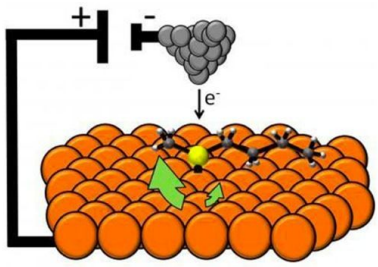


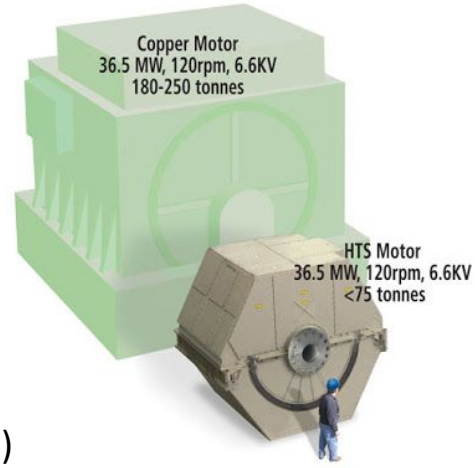
Moteurs

Types de Moteurs

Il existe un très grand nombre de moteurs :

de 1 seule molécule = 1 nanomètre



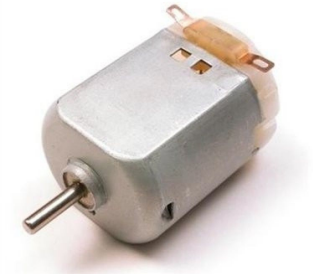


à 250 Tonnes

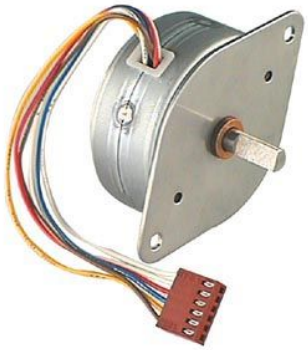
(ou 75 tonnes à supraconducteur à -241°)

Types de moteurs pour l'embarqué

Moteur CC à courant continu



Moteur pas à pas



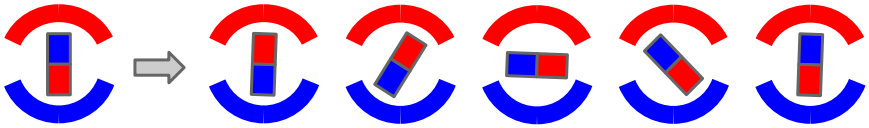
Servo moteur

Principe du moteur électrique

Le but d'un moteur est de transformer l'énergie électrique en énergie mécanique

Un moteur est constitué de deux parties:

1. partie fixe : le stator constitué de deux aimants immobiles
2. partie mobile : le rotor constitué de deux aimants sur un axe de rotation



Rouge SUD

Bleu NORD

stable changement polarité du rotor stable

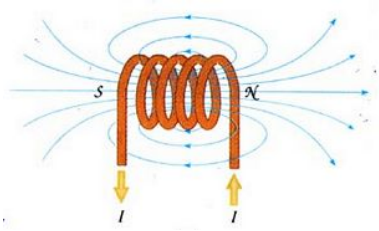
Ici le stator est un aimant permanent et
le rotor est un aimant dont on peut inverser la polarité,

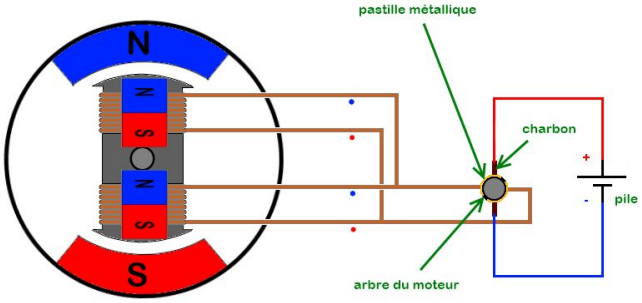
Cela peut être l'inverse ou les deux.

moteur à courant continu

Principe du moteur électrique CC (suite)

Pour créer un aimant inversable, on utilise une bobine qui fabrique
un champ magnétique dont le direction dépend de la direction du courant.

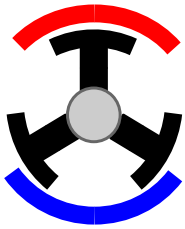




http://eskimon.fr/

Le courant s'inverse grâce à
des balais qui glissent sur l'axe
sur lequel sont placées
les extrémités des bobines.

En réalité, il y a 3 bobinages pour que l'axe soit toujours mal placé
et tourne. Il existe plein de variantes !!!



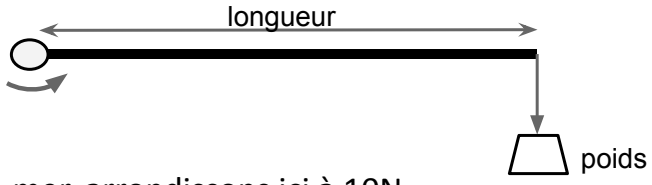
Principe du moteur électrique (suite)

On caractérise un moteur par son couple, sa vitesse, sa puissance

Couple

capacité à faire tourner son axe mesuré en Newton-mètre

- Le **couple de blocage** est la capacité du moteur à maintenir un poids à l'extrémité d'un axe.



- 1 kg = 9.81N au niveau de la mer, arrondissons ici à 10N.
- Un moteur dont le couple est 10Nm signifie qu'il est capable de soulever un poids d'un kilo accroché à un axe de 1m. C'est une limite.
- Le **couple utile** est le couple de blocage divisé par deux.
  ici on évite de dépasser 500g dans la pratique.
- Si on divise la longueur par 2, on augmente le poids par deux.

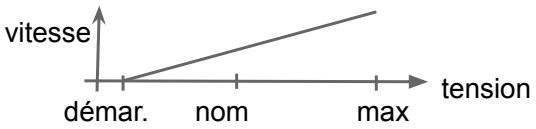
Principe du moteur électrique (suite)

On caractérise un moteur par son couple, sa vitesse, sa puissance

Vitesse

Nombre de rotations de l'axe par seconde en rad/sec ou en tour/min.

- La vitesse dépend du champ magnétique créé et donc de la tension.
- Typiquement 30 tours/sec par Volt ⇒ sous 3V 3*30*60=5400 t/min
- La vitesse est mesurée à vide



Puissance

La puissance mécanique d'un moteur est égale à son couple * vitesse

- $P_{mécanique}$ = N.m.rad/s = $P_{électrique}$. rendement
- rendement (entre 80% et 95%).

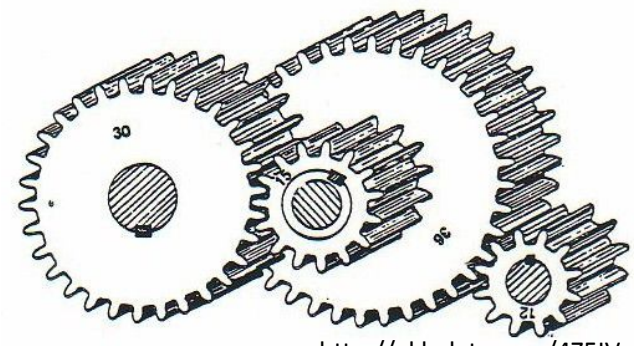
Pour un moteur donné, il existe une tension maximale
au-delà de laquelle le moteur chauffe et finit par casser.

→ La tension nominale est la moitié de la tension maximale.

Lien entre le couple et la vitesse

Si on veut augmenter le couple sans augmenter la tension d'alimentation

→ on utilise des engrenages



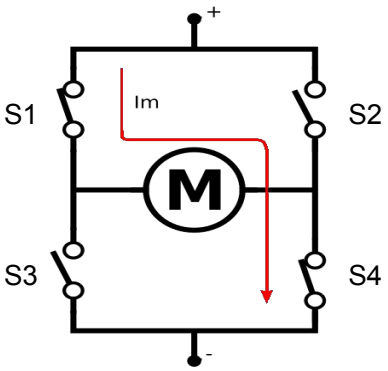
http://ekladata.com/4Z5IVndsoqWrjdxOBjwFg2qzYfc.jpg

→ Réduire la vitesse de rotation par X augmente de couple de X
(si on néglige les pertes en chaleur)

Pont en H

Le sens de rotation d'un moteur dépend du sens du courant dans les bobines au démarrage.

Pour commander le sens du courant on utilise un pont en **H**



Combinaisons d'états des commutateurs

État des commutateurs				Résultat à la charge
S1	S2	S3	S4	
✗	✗	✗	✗	Aucune tension aux bornes de la charge.
✓	✗	✗	✓	Courant positif à travers la charge.
✗	✓	✓	✗	Courant négatif à travers la charge.
✓	✓	✗	✗	Charge court-circuitée.
✗	✗	✓	✓	Charge court-circuitée.

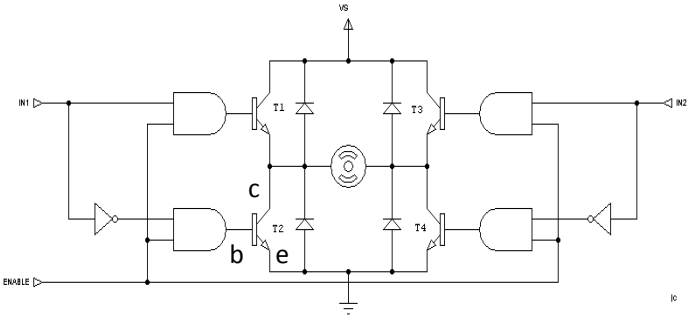
http://fr.wikipedia.org/wiki/Pont_en_H

commande d'un moteur cc

Les interrupteurs sont des transistors bipolaires passant si leur Base est à 1.

Les diodes sont présentes pour éliminer le courant produit lorsque le moteur s'arrête et qu'il fonctionne en générateur.

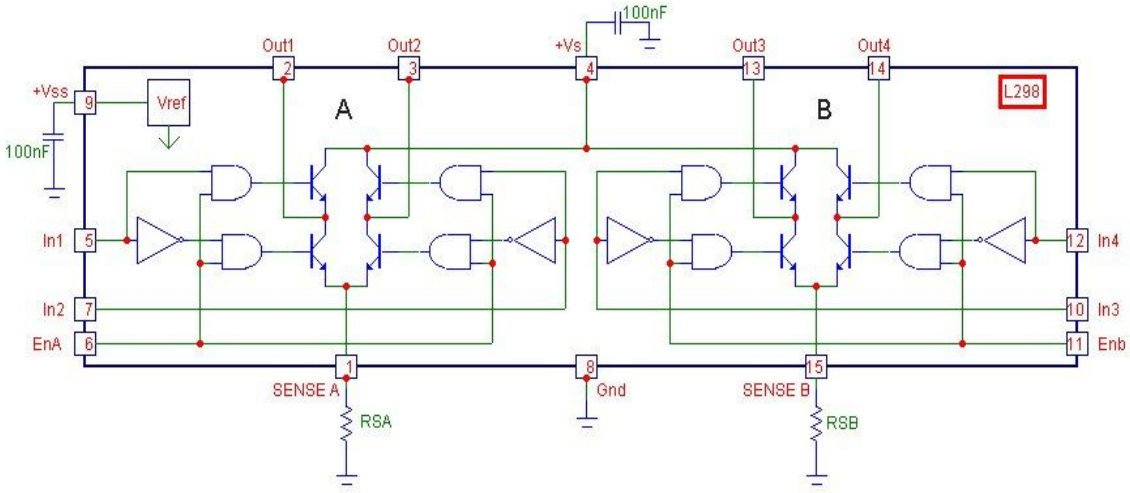
Les diodes sont de type Schottky (rapide et faible seuil 0.4V), elles protègent les transistors des surtensions lors des changement de sens de rotations.



http://naytech.blogspot.fr/2013/09/le-circuit-l298.html

commande d'un moteur cc (suite)

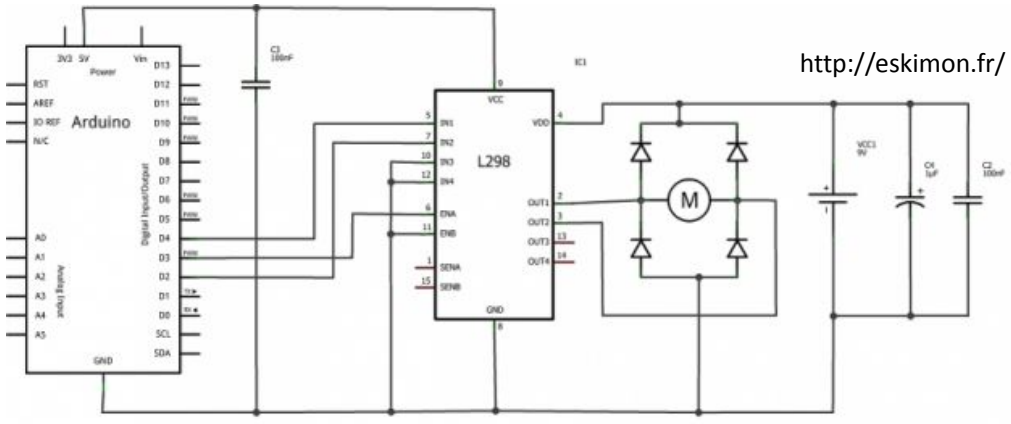
Le L298 contient 2 ponts en H



http://forums.futura-sciences.com/electronique/137217-l298-demande-explication-pont-h.html

commande d'un moteur cc (suite)

Le L298 contient 2 ponts en H : il ne manque que les diodes de protections et des capacités pour éliminer les parasites (liés aux sur-tentions d'arrêt) et fournir un courant de démarrage.



http://eskimon.fr/

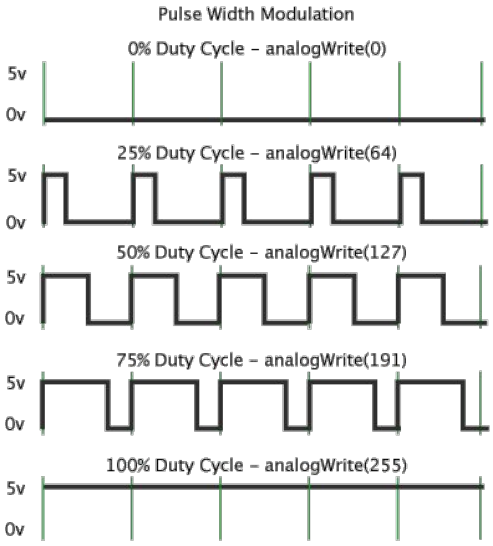
commande moteur cc avec Arduino

Pour contrôler la vitesse, il faut contrôler la tension.
En numérique, c'est réalisé par un signal modulé en largeur d'impulsion : PWM

En Arduino, la fonction

`void analogWrite(int pin, int value)`
produit un signal envoyé sur la broche `enable` du L298/293

- Pins
  - Arduino Uno 3, 5, 6, 9, 10, 11
  - Arduino Mega 2 à 13 et 44 à 46
  - ESP32 16 cannaux
- Fréquence
  - autour de 500Hz
- Précision
  - 8 bits = 256 états différents
  - 16 bits = 65536 états (ESP32)



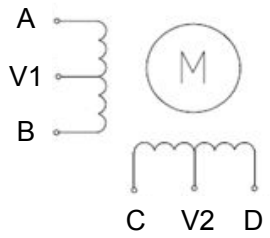
http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.AnalogWrite

moteur pas à pas

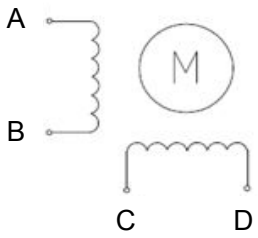
moteur pas à pas

- Comme un moteur à courant continu, un moteur pas à pas est constitué d'un rotor et d'un stator, mais il n'est pas instable, un courant dans les bobines va entraîner une rotation et une position d'équilibre qui va se maintenir.
- Un changement de sens du courant va entraîner à nouveau le moteur qui tourne jusqu'à l'équilibre suivant :

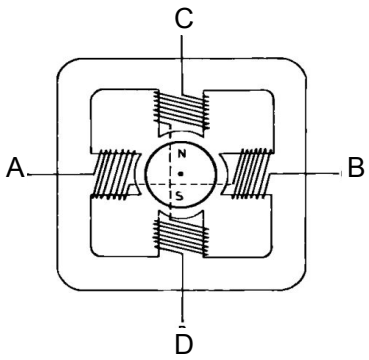
  ⇒ Le moteur avance d'un pas à chaque fois
- Il existe deux types de moteurs



unipolaire : 6 fils



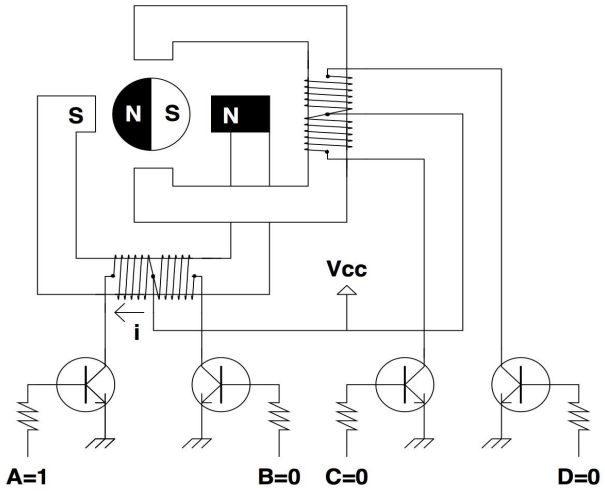
bipolaire : 4 fils



moteur pas à pas unipolaire

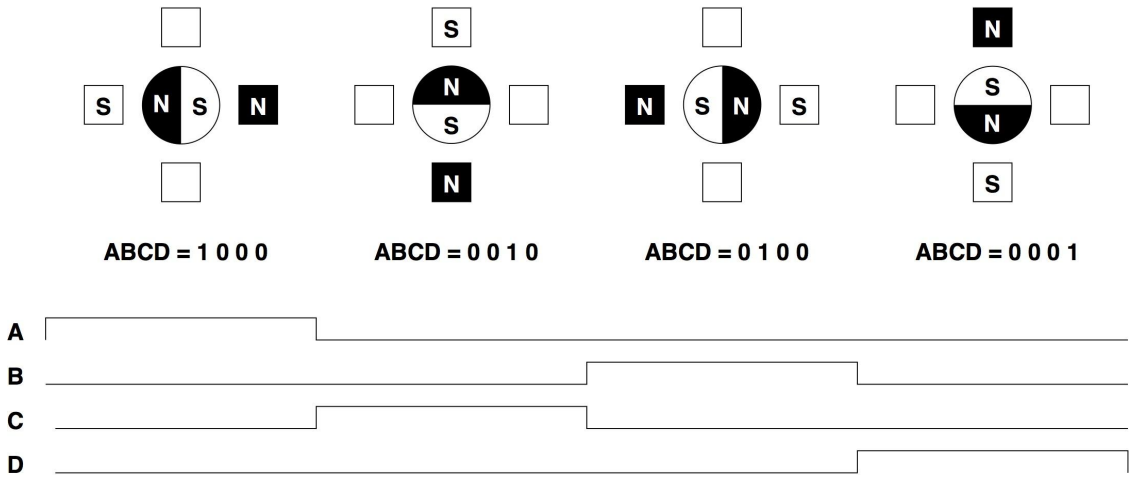
Ici le courant circule que dans un seul sens

- La polarisation d'une des bobines du stator définit une position d'équilibre du rotor.
- La rotation va se faire en commandant en séquence les signaux A, B, C et D
- Ici les commandes A et B sont opposées :
  - soit A = 1 ⇒ S - N
  - soit B = 1 ⇒ N - S
- C'est simple mais seule la moitié de la bobine est utilisée à la fois.



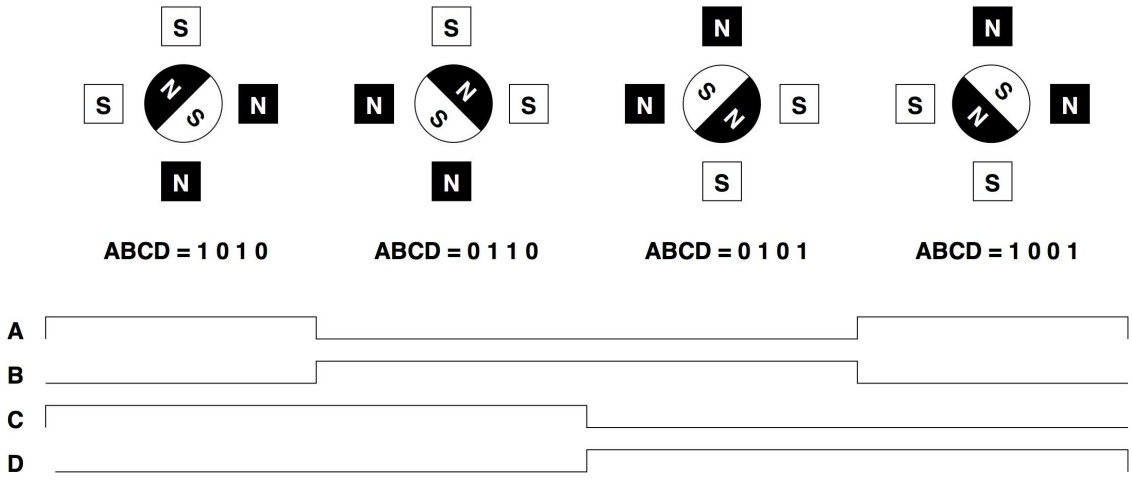
commande moteur p-à-p unipolaire monophasé

Séquence de valeur pour une rotation en 4 pas



commande moteur p-à-p unipolaire biphasé

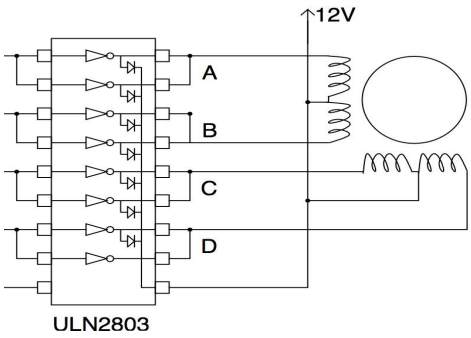
On peut aussi commander 2 bobines à la fois pour avoir plus de couple

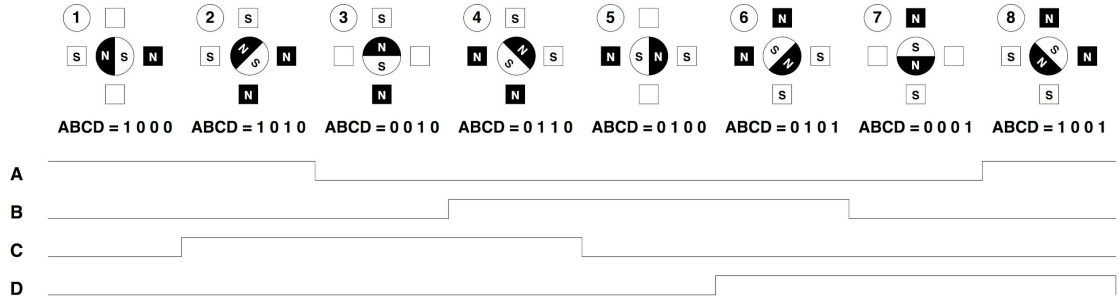


commande moteur p-à-p unipolaire biphasé

Si on combine les modes précédents
→ on obtient 8 demi-pas

On peut encore augmenter le nombre de pas
en modulant la tension sur les bobines (PWM)

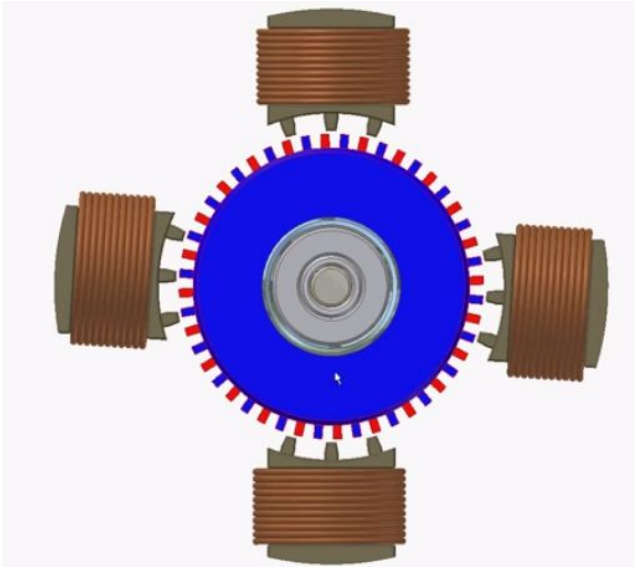




Augmentation du nombre de pas

Avec 2 bobines, il est possible
d'avoir un grand nombre de pas

Les vidéos citées montrent
le fonctionnement détaillé
de ce type de moteur

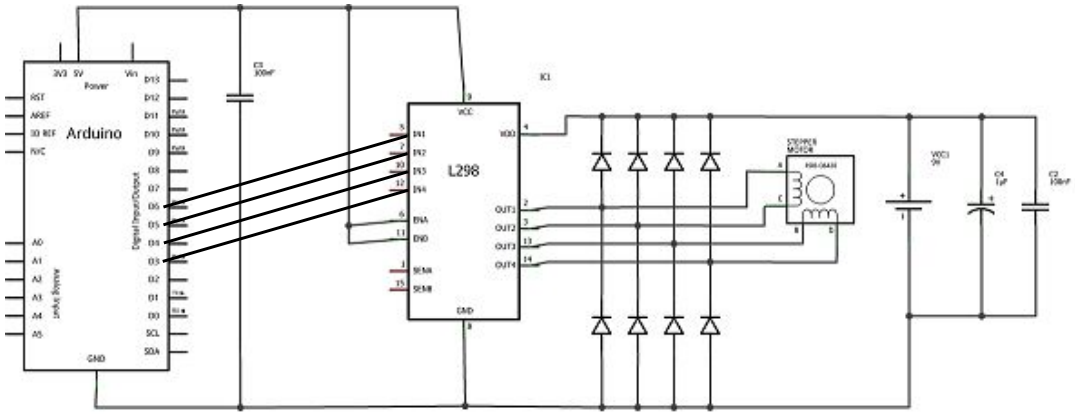


Vidéos intéressantes

https://www.youtube.com/watch?v=t-3VnLadIbc
https://www.youtube.com/watch?v=Ew6eVGnj7r0

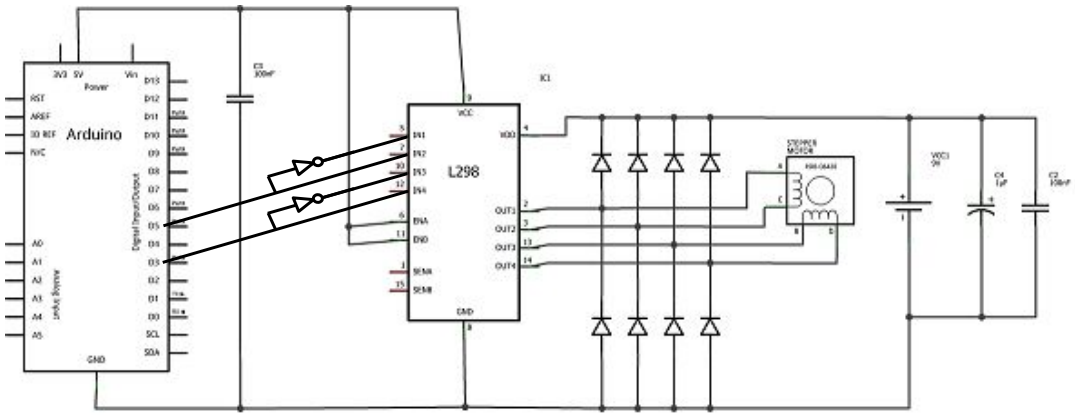
Commande d'un moteur p-à-p bipolaire

Le L298 contient deux ponts en H, celà fait un par bobine.



Commande d'un moteur p-à-p bipolaire

En ajoutant 2 inverseurs on peut économiser deux fils en remarquant que les bornes des bobines sont toujours opposées.



Commande d'un moteur p-à-p avec Arduino

Library Stepper

- Stepper(steps, pin1, pin2)
  Stepper(steps, pin1, pin2, pin3, pin4)
  - steps nombres de pas par tour
  - pin1, pin2 broches de connexion
  - pin3, pin4 broches de connexion lorsqu'on connecte les 4 extrémités
- setSpeed(rpm)
  - rpm nombre de tours par minute (entier > 0)
- step(steps)
  - steps nombre de pas souhaité (positif ou négatif)

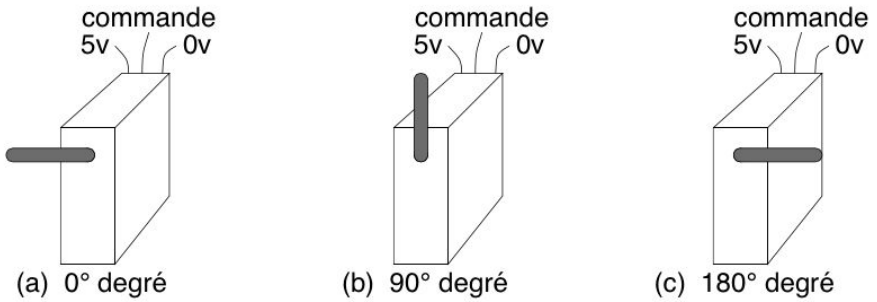
Je ne sais pas si cette librairie gère les micro-pas

servo moteur

servo moteur

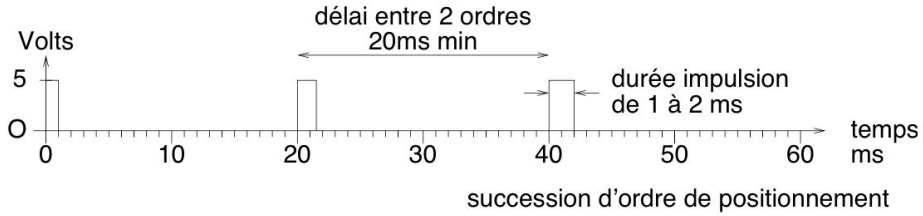
Un servo-moteur est un dispositif constitué d'un moteur CC asservi,
c.-à-d. dont la position est contrôlée par une interface de commande.

- L'axe du servo-moteur peut être placé dans une position variant de 0° à 180° (ou 270°).
- Ici, le servo-moteur est dans trois positions, (a) 0° : position la plus extrême à gauche, (b) 90° : la position centrale, (c)180° : position la plus extrême à droite.

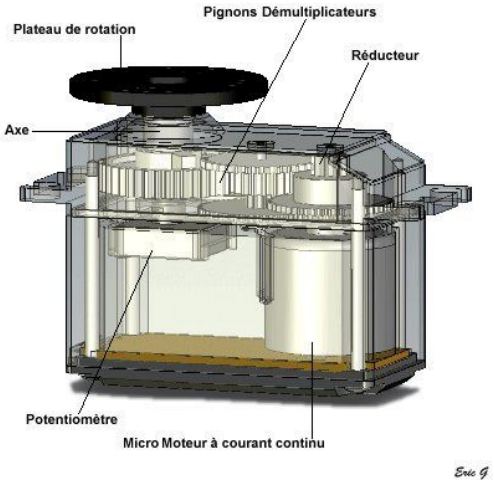


Principe de fonctionnement

- La commande d'un servo utilise un fil unique de commande et 2 fils d'alimentation.
- Par défaut, le signal de commande est à 0V. Pour placer l'axe dans une position donnée, il faut mettre le signal de commande à 5V pendant une durée comprise entre 1ms et 2ms.
- Une impulsion à 5V sur le signal de commande pendant 1ms, met l'axe à 0°
- Une impulsion de 2ms le met à 180°.
- En l'absence d'ordre, le servo-moteur est en roue libre.
- On peut envoyer une succession d'ordres, mais le délai entre deux ordres successifs est de l'ordre de 20 ms (dépend du moteur et de son couple)
- La vitesse de rotation est telle que le moteur peut faire tourner son axe à 1 tour/sec donc 0.5 seconde pour aller de 0° à 180°.
- Il y a un asservissement qui corrige la position si l'axe dérive.



Anatomie d'un servo



fils

- noir ← 0V
- rouge ← 4,5 à 6V
- jaune, blanc, orange ← commande

http://eskimon.fr/287-arduino-602-un-moteur-qui-de-la-tete-le-servo-moteur

Commande d'un servo avec Arduino

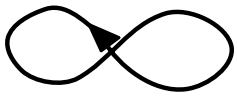
- attach(pin)
  attach(pin, min, max)
  - connecte le servo à la broche pin.
    Rends 0 en cas de problème
  - min, max sont les durée en µs
    des positions extrêmes
- detach()
  - déconnecte le servo
- write(int)
  - définie la position angulaire
    de 0 to 180.
- read()
  - rend la dernière position demandée.
- attached()
  - rend 1 si le servo est attaché
- refresh()
  - doit être appelée au minimun toutes les 50ms

```
#include <SoftwareServo.h>

SoftwareServo myservo;

int potpin = 0;
int val;

void setup()
{
  myservo.attach(2);
}

void loop()
{
  val = analogRead(potpin);
  val = map(val, 0, 1023, 0, 179);
  myservo.write(val);
  delay(15);
  SoftwareServo::refresh();
}
```

Conclusion

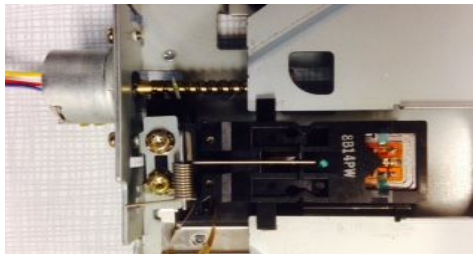
Pleins de possibilités

- moteur CC
  - mesurer la vitesse de rotation
  - dessiner une forme

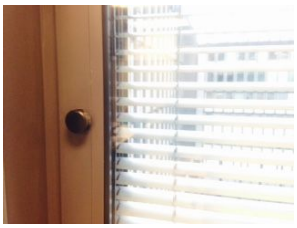


http://www.mecadroid.com/

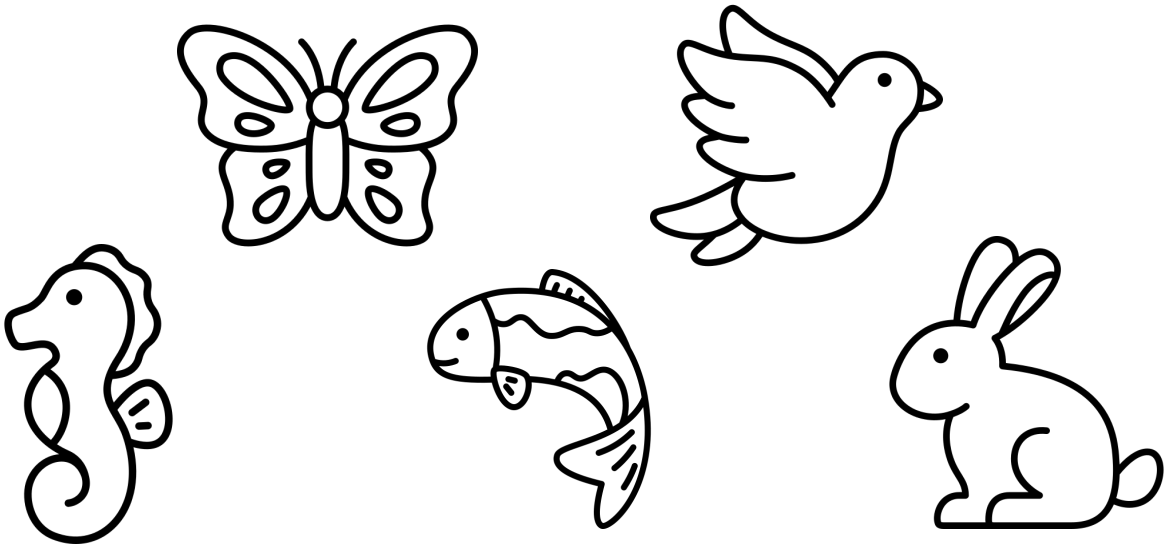
- moteur pas-à-pas
  - mesurer la vitesse
  - déplacer le berceau d'un lecteur de disquette



- servo
  - ouverture des stores



Si vous avez du temps et une idée…



Dessins : https://www.flaticon.com/fr/auteurs/mihimihi