

CONTROL DE MOTORES

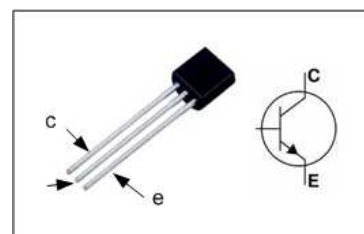
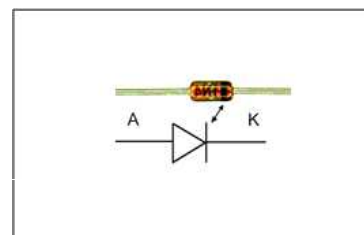
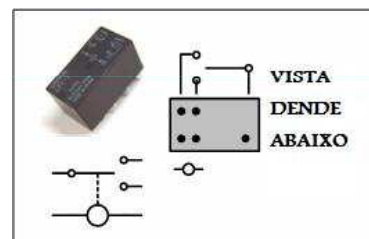
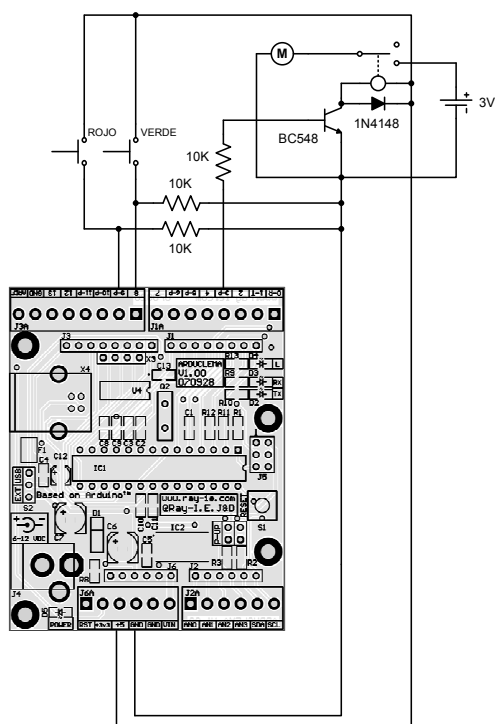
Motores de DC con escobillas

Servos

PaP (Steppers)

CONTROL DE MOTOR CON RELE

ESQUEMA CONEXIÓN:



PRÁCTICA 1 – CONTROL DE MOTOR DC TODO/NADA CON C.I.

DESCRIPCIÓN:

- Realizar un programa que controle un motor mediante o driver L293. Facer que xire 5 s nun sentido, pare 2 segundos e xire 5 s noutro sentido.

EXERCICIO:

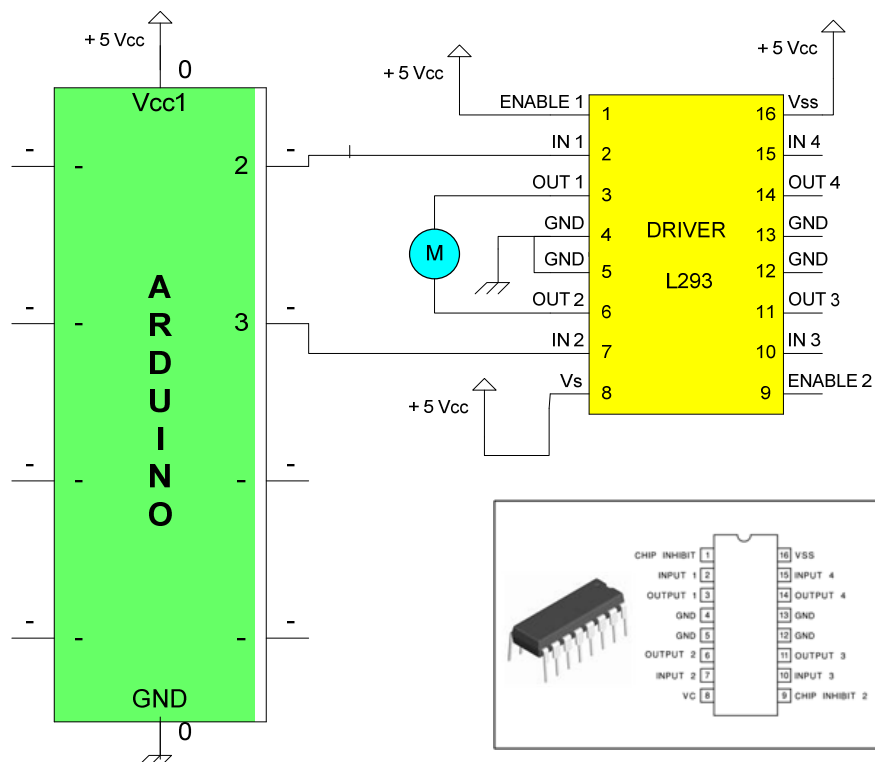
- Realizar o programa atendendo á táboa de funcionamento do driver L293:

ENABLE 1	IN 1	IN 2	SAÍDA
0	X	X	0
1	0	0	0
1	0	1	→
1	1	0	←

3

PRÁCTICA 1 – CONTROL DE MOTOR TODO/NADA CON C.I.

ESQUEMA CONEXIÓN:



4

PRÁCTICA 2 – REGULACIÓN DE VELOCIDADE DE MOTOR

EXERCICIO:

- Modificar o circuío segundo se indica.
- Modificar o programa de forma que o motor vaia incrementando a velocidade en 5 pasos (cada 2 s).

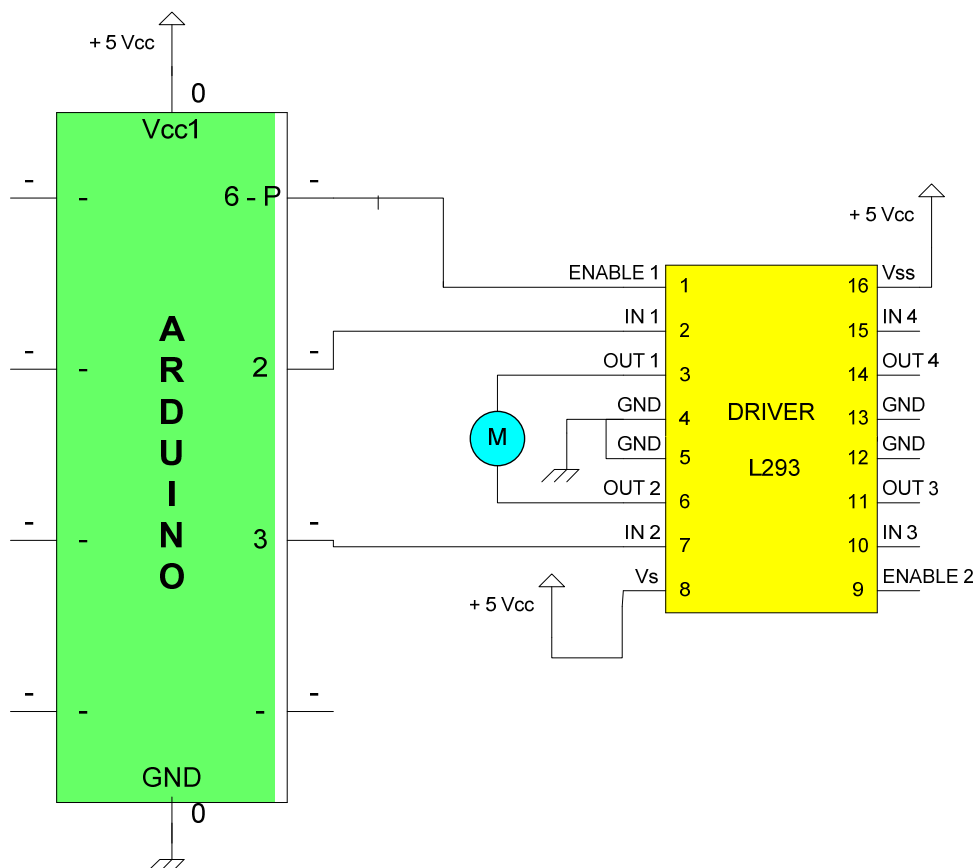
REFERENCIAS:

- *analogWrite()* : escribe na saída analóxica PWM.
- Actuando cun sinal PWM na patiña enable do L293 → Velocidade modulable.

* O C.I. L293 permite o control independente de 2 motores.

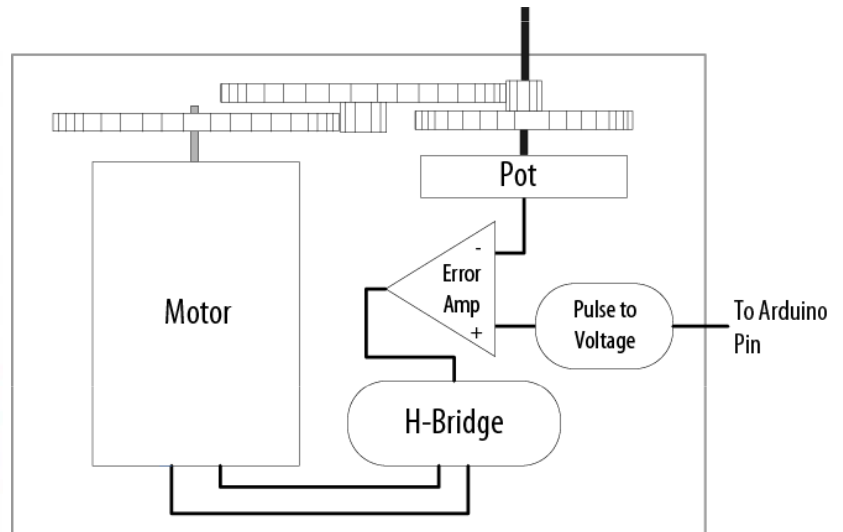
5

PRÁCTICA 2 – REGULACIÓN DE VELOCIDADE DE MOTOR CC

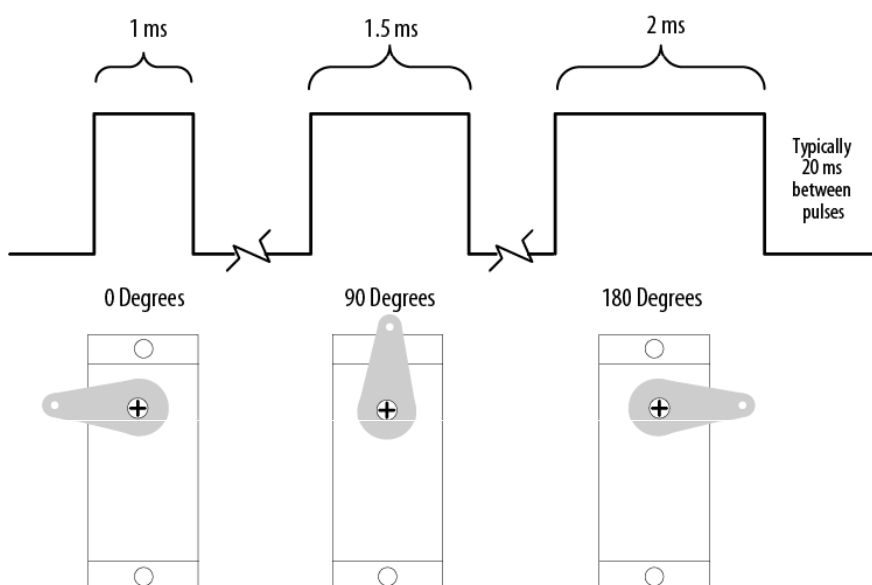


6

Control de Servos



Principio de funcionamento

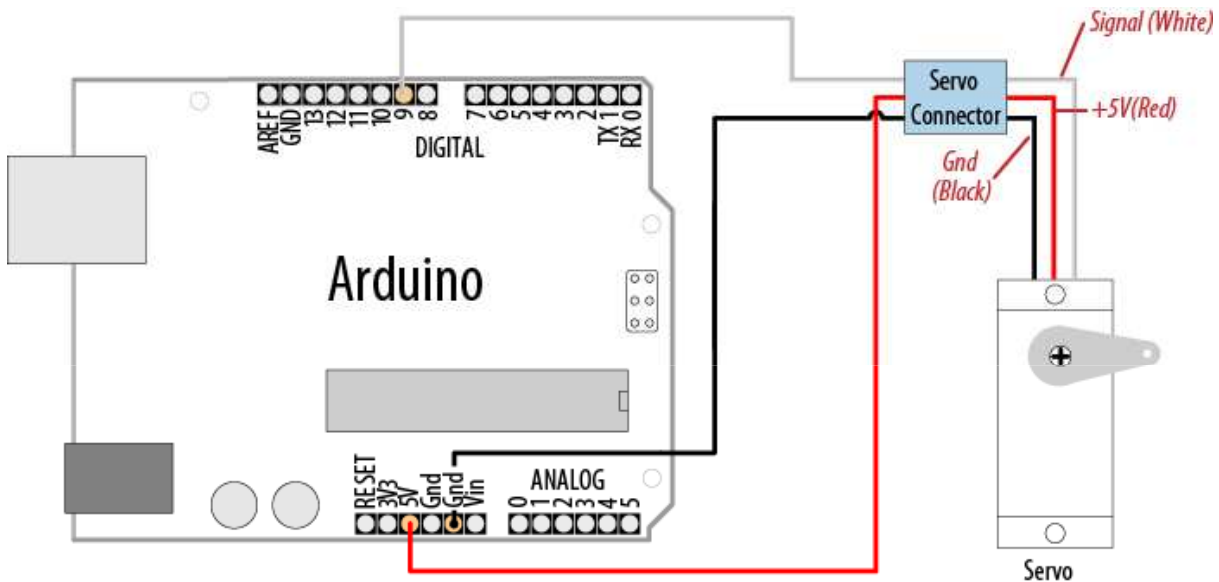


-Os servos funcionan cunhas modulacóns PWM específicas, polo que non funcionan correctamente coas saídas analóxicas con “analogWrite”.

- Para o correcto control existe a librería “Servo.h”.

PRÁCTICA 1 – Control dun servo

- **ESQUEMA CONEXIÓN:**



9

PRÁCTICA 1 – Control dun servo

- **DESCRIPCIÓN:**

Programa no que o servo sigue o movemento dun potenciómetro.

- **EXERCICIO:**

Executar o exemplo “Knob”.

- **REFERENCIAS:**

- `#include <Servo.h>`
- `Servo myservo;` //Declaración dun obxecto tipo “Servo”
- `myservo.attach(pin);` //Asociación cun pin.
- `myservo.write(graos);` //Posiciónase nun ángulo directamente
- `valor = map(valor, 0, 1023, 0, 179);` //Útil para escalado enteiro

10

PRÁCTICA 2 – Servo serial

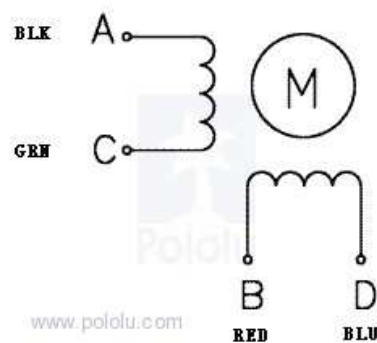
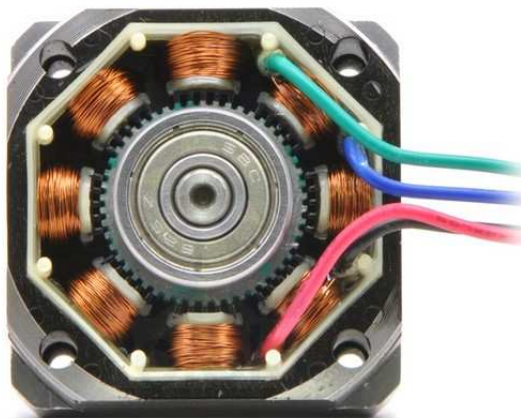
- Descripción:

Facer un programa que reciba polo porto serie unha posición (Esquerda, Centro, Dereita) e posicione o servo en consecuencia.

- Referencias:

- `Serial.available()` // Chegada de datos ao porto serie
- `caracter = Serial.read();` //Lectura caracter

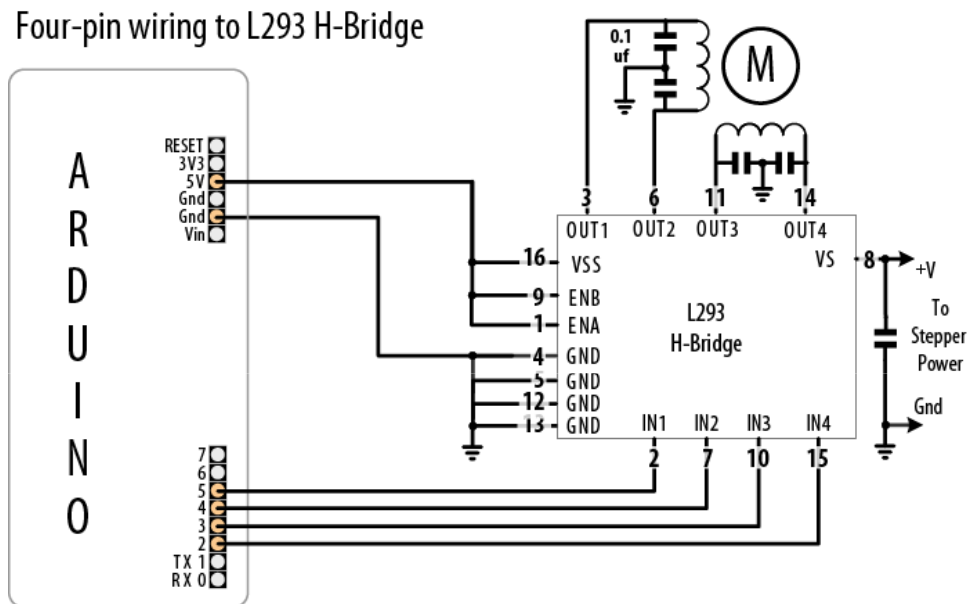
Motores PaP



- Librería “Stepper.h”

PRÁCTICA 1 – Manexo PaP

- **ESQUEMA CONEXIÓN PaP:**



13

PRÁCTICA 1 – Motor Knob PaP

- **DESCRIPCIÓN:**

O programa “Stepper-MotorKnob” fai que un motor PaP siga as variacións no movement dun potenciómetro conectado a unha entrada analóxica.

- **EXERCICIO:**

Cargar o exemplo, interpretar o código e verificar o seu funcionamento.

14

PRÁCTICA 1 – Motor Knob (PaP)

```
#include <Stepper.h>

#define STEPS 200 //Numero de pasos/volta do motor concreto

// Instancia da clase "Stepper" especificando o número de pasos
// e os pins aos que esta conectado.
Stepper stepper(STEPS, 2, 3, 4, 5);

int previous = 0; // Inicializacion valor previo

void setup()
{
  stepper.setSpeed(30); // Axuste da velocidade de xiro a 30 RPMs
}

void loop()
{
  int val = analogRead(0); // Lectura valor actual do potenciometro ( entre 0 ...1023)

  stepper.step(val - previous); // Move o motor un numero de pasos igual ao cambio na valor

  previous = val; // Recorda valor previo potenciometro
}
```