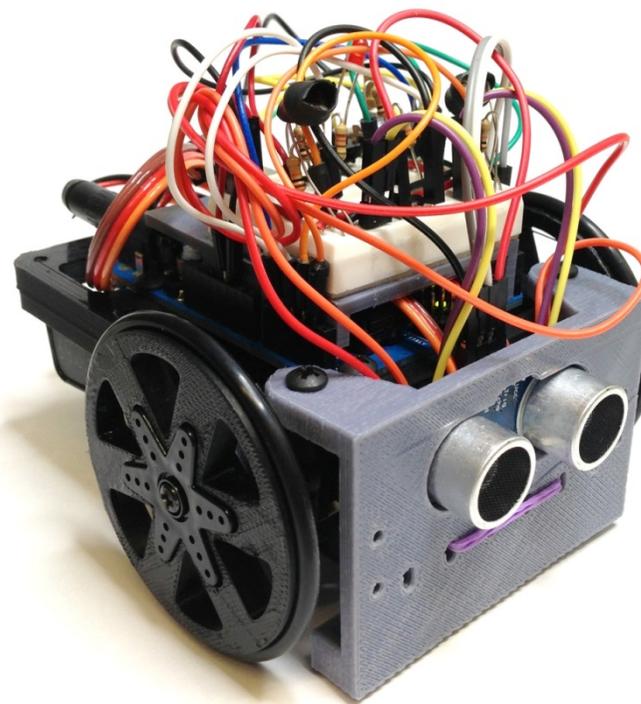


GUIA DE MONTAJE Y PROGRAMACIÓN

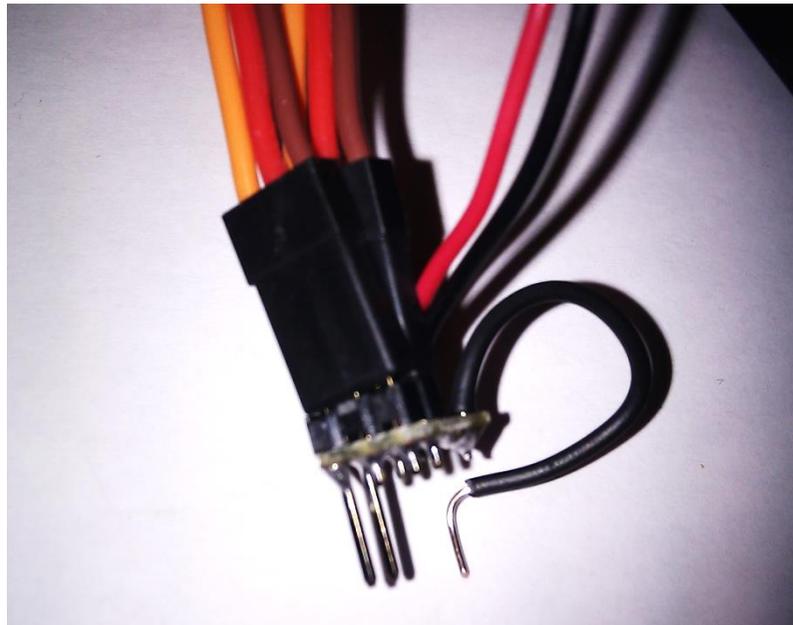
MOTORES



Conectamos los motores



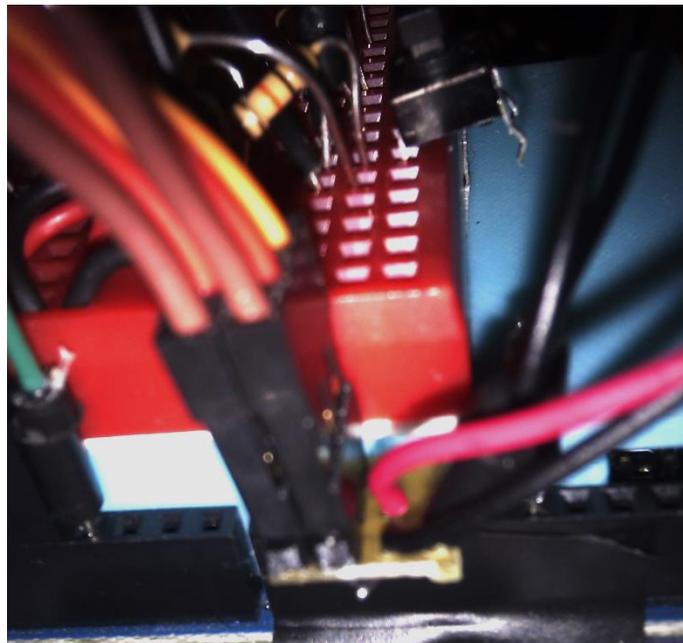
- Conecta los conectores de los servos.
 - **¡CUIDADO! Podemos romper los motores.**
 - El cable **marrón** de los servos tiene que ir unido con el cable **negro**.
 - El cable **rojo** con el cable **rojo** de las pilas
 - Las señales de control **amarillas** van conectados directamente al Arduino UNO.



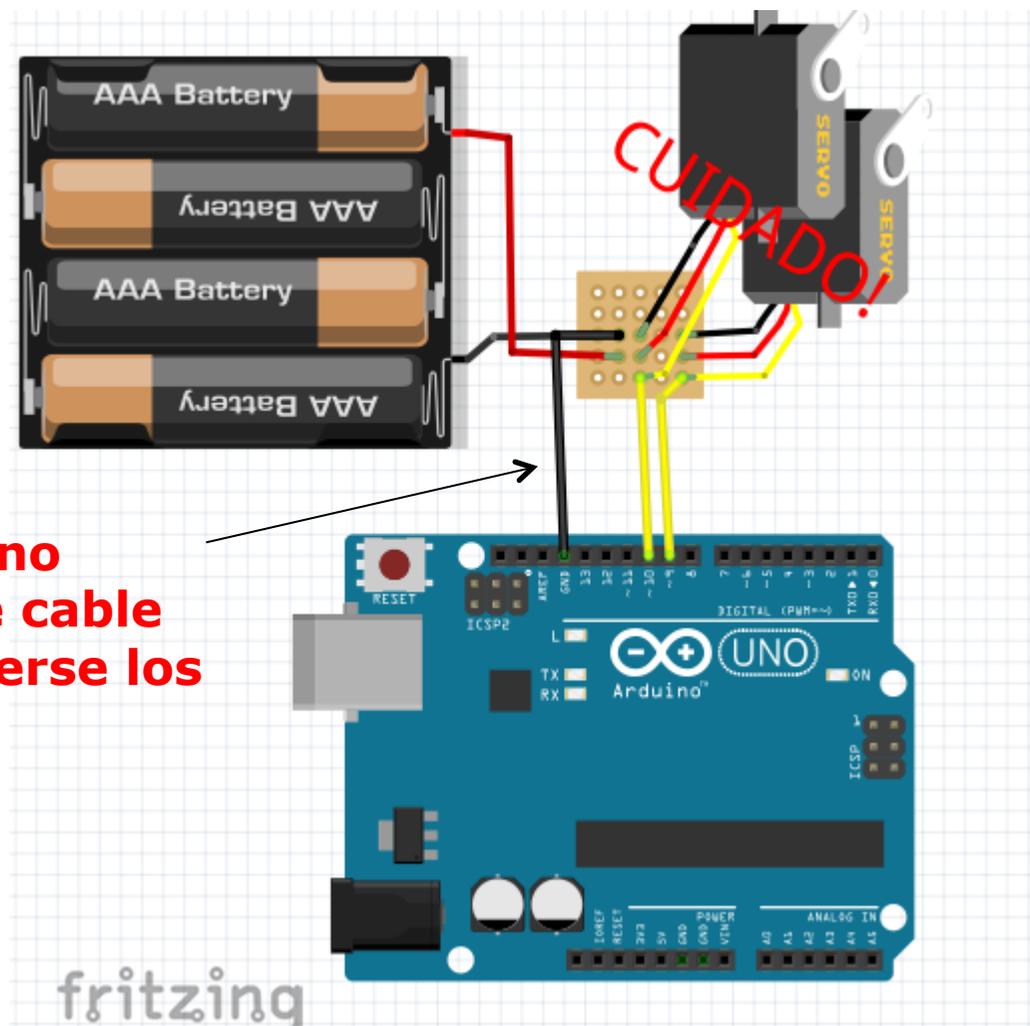
Conectamos los motores



- Conecta los conectores de los servos.
 - Las señales de control **amarillas** van conectados directamente al Arduino UNO.
 - Van a ir a los PIN 9 y PIN 10.



Conectamos los motores



CUIDADO. Si no conectas este cable pueden romperse los motores

Control de motores



- Ejemplo: Probando un motor (motorSimple → motorLeft)
 - Este ejemplo mueve el motor izquierdo a diferentes velocidades
 - Modifica el ejemplo para probar con el motor derecho

```
#include <Servo.h>
#include <tubot_motor.h>

MotorTubot motorIzda;
int pinMotorIzda = 9;

void setup(){
  motorIzda.begin(pinMotorIzda);
}
```

Hay que declarar una "variable" (objeto) de tipo **MotorTubot**

```
void loop(){
  delay(2000);
  motorIzda.setSpeed(50);
  delay(2000);
  motorIzda.setSpeed(100);
  delay(2000);
  motorIzda.setSpeed(0);
  delay(2000);
  motorIzda.setSpeed(-50);
  delay(2000);
  motorIzda.setSpeed(-100);
  delay(2000);
}
```

Control de motores



- Ejemplo: Probando un motor (motorSimple → motorLeft)
 - Este ejemplo mueve el motor izquierdo a diferentes velocidades
 - Modifica el ejemplo para probar con el motor derecho
 - Carga el programa: ¿Funciona como pensabas que iba a funcionar?
 - Llama a un profesor y se lo cuentas

```
#include <Servo.h>
#include <tubot_motor.h>
```

```
MotorTubot motorIzda;
int pinMotorIzda = 9;
```

```
void setup(){
  motorIzda.begin(pinMotorIzda);
}
```

Para configurar la velocidad del motor de -100 a + 100

```
void loop(){
  motorIzda.setSpeed(0);
  delay(2000);
  motorIzda.setSpeed(50);
  delay(2000);
  motorIzda.setSpeed(100);
  delay(2000);
  motorIzda.setSpeed(0);
  delay(2000);
  motorIzda.setSpeed(-50);
  delay(2000);
  motorIzda.setSpeed(-100);
  delay(2000);
}
```



Control de motores:

Para controlar cada uno de los motores lo que tenemos que hacer es:

- 1) Crear un objeto "Motor" por cada uno de los motores.
Actuando sobre estos objetos estaremos actuando sobre los motores reales.
- 2) Asociar a cada motor un "pin" de la placa Arduino.
Para controlar los motores necesitamos pasarle la información a través de un pin.
- 3) Establecer el 0 del motor. **Calibrar**
- 4) Establecer la velocidad del motor deseada

Control de motores

- Ejecuta el programa de calibración (GetMotorsZero_example)
 - Abre el monitor serie de Arduino
 - Pulsando las teclas A y D para el motor izquierdo y J y L para el motor derecho, ajusta los motores hasta que se paren.
 - Apunta los valores que se muestran en pantalla cuando estén los motores completamente parados. Estos son los valores de calibración.
 - Vuelve a cargar el ejemplo (motorSimple → motorLeft) añadiendo como parámetro del motorLeft.begin() el valor obtenido en la calibración.
 - Por ejemplo: **motorLeft.begin(9, 1250);**
 - Ejecuta el programa y mira a ver si se comporta de una manera más lógica que antes.
 - Pregunta al profesor si tienes dudas.

Control de motores



- Realiza los siguientes programas:
 - Prueba 1: Haz que el robot se mueva en línea recta
 - Si se tuerce un poco también valdría
 - Prueba 2: Haz un programa que haga que el robot rote en el sitio

Uso de la librería TuBot



□ Permite controlar el robot de forma más fácil

■ Primero hay que declarar el "Robot"

```
MoverRobot tubot;
```

■ Configurar los pines y los valores de calibración

```
tubot.begin(9, 10, 1238, 1250);
```

■ Usar los comandos (las velocidades van de +100 a -100)

```
tubot.recto(velocidad);
```

```
tubot.para();
```

```
tubot.dcha(velocidad_de_giro, velocidad_en_recto);
```

```
tubot.izda(velocidad_de_giro, velocidad_en_recto);
```

■ Si no se indica `velocidad_en_recto` se entiende que es cero

Uso de la librería TuBot



- Prueba el ejemplo Motors_example
 - Comprueba que se comporta como esperabas a la vista del código
 - No olvides poner TUS parámetros de calibración de los motores

```
#include <Servo.h>
#include <tubot_motor.h>

int pinmotorIzda = 9;
int pinmotorDcha = 10;
MoverRobot tubot;

void setup(){
  tubot.begin(pinmotorIzda , pinmotorDcha,1238 ,1250 );
}
```

Uso de la librería TuBot



- Prueba el ejemplo Motors_example
 - Comprueba que se comporta como esperabas a la vista del código
 - No olvides poner TUS parámetros de calibración de los motores

```
void loop(){  
  
  tubot.para(); // Parar robot  
  delay(1000); // El robot estara parado 1s  
  
  tubot.recto(50); // Mover robot hacia delante  
  delay(1000); // Durante 1s  
  
  tubot.dcha(20); // Girar el robot hacia la dcha  
  delay(700); // Durante 0.7s  
  
  tubot.recto(-20); // Mover robot hacia atras  
  delay(1500); // Durante 1.5s  
  
  tubot.izda(40); // Girar el robot hacia la izda  
  delay(500); // Durante 0.5s  
  
}
```

Uso de la librería TuBot

- Crea un programa que haga que el robot siga un cuadrado de unos 20cm de lado
- En el ejemplo de “Funciones_Avanzadas” tienes muchas funciones que te pueden servir. Pruébalas cambiando el contenido del loop()
 - Si quieres que una línea no se ejecute puedes poner un // delante.

Programa de Control General



Este programa sirve para poder medir, comprobar y “afinar” los sensores y actuadores del robot. Mediante el teclado podremos modificar los diferentes actuadores:

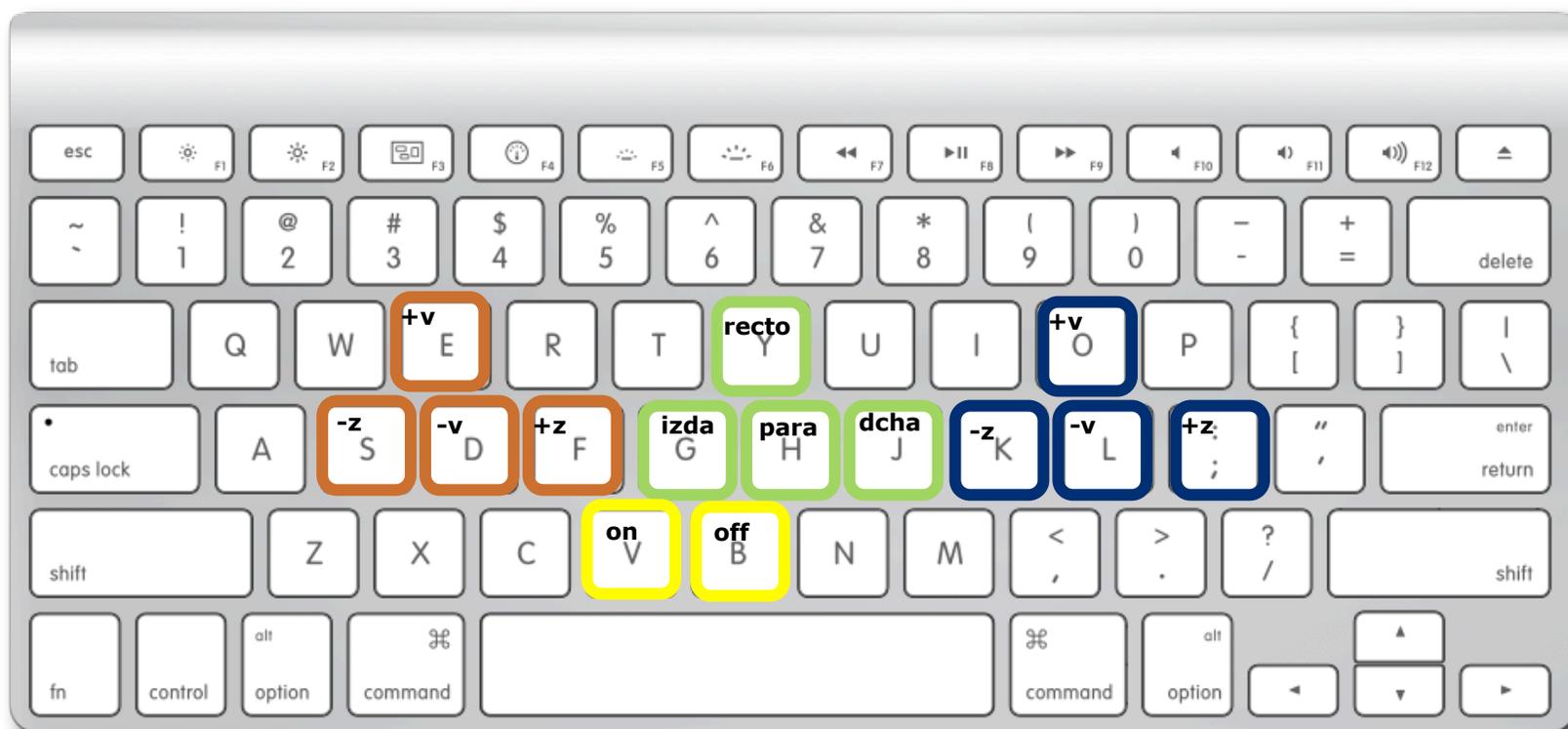
- Motor izdo: ``a'y'd'`: Modifica el 0 del motor
``w'y's'`: Modifica la velocidad del motor

- Motor dcho: ``j'y'l'`: Modifica el 0 del motor
``i'y'k'`: Modifica la velocidad del motor

- Control general: ``t'`: Mover robot hacia delante.
``g'`: Parar robot
``f'`: Girar izquierda
``h'`: Girar derecha

- Control LED: ``v'`: Enciende LED
``b'`: Apaga LED

Diagrama de teclas para el control general del robot.



<http://www.apple.com/keyboard/#hero-2>



Monitor Serie

COM6

Enviar

Distancia	Pulsador	Potenciometro	SensorIzd	SensorDer	LDRIzd	LDRDer	VelMotIzd	VelMotDer	CalMotorIzd	CalMotorDer
5	1	781	609	530	820	813	30	30	1238	1250
Distancia	Pulsador	Potenciometro	SensorIzd	SensorDer	LDRIzd	LDRDer	VelMotIzd	VelMotDer	CalMotorIzd	CalMotorDer
5	1	781	609	530	818	812	30	30	1238	1250
Distancia	Pulsador	Potenciometro	SensorIzd	SensorDer	LDRIzd	LDRDer	VelMotIzd	VelMotDer	CalMotorIzd	CalMotorDer
5	1	781	610	529	817	811	30	30	1238	1250
Distancia	Pulsador	Potenciometro	SensorIzd	SensorDer	LDRIzd	LDRDer	VelMotIzd	VelMotDer	CalMotorIzd	CalMotorDer
5	1	781	610	530	815	809	30	30	1238	1250
Distancia	Pulsador	Potenciometro	SensorIzd	SensorDer	LDRIzd	LDRDer	VelMotIzd	VelMotDer	CalMotorIzd	CalMotorDer
5	1	781	609	529	813	808	30	30	1238	1250
Distancia	Pulsador	Potenciometro	SensorIzd	SensorDer	LDRIzd	LDRDer	VelMotIzd	VelMotDer	CalMotorIzd	CalMotorDer
5	1	781	609	529	812	807	30	30	1238	1250
Distancia	Pulsador	Potenciometro	SensorIzd	SensorDer	LDRIzd	LDRDer	VelMotIzd	VelMotDer	CalMotorIzd	CalMotorDer
5	1	782	610	530	811	806	30	30	1238	1250
Distancia	Pulsador	Potenciometro	SensorIzd	SensorDer	LDRIzd	LDRDer	VelMotIzd	VelMotDer	CalMotorIzd	CalMotorDer
5	1	781	610	532	815	807	30	30	1238	1250

Desplazamiento automático

No hay fin de línea 19200 baud