

Wiki

Pages & Files

Search this workspace

If you are citizen of an European Union member nation, you may not use this service unless you are at least 16 years old.

Dokkio Sidebar (from the makers of PBworks) is a Chrome extension that eliminates the need for endless browser tabs. You can search all your online stuff without any extra effort. And Sidebar was **#1 on Product Hunt!** [Check out what people are saying by clicking here.](#)

VIEW

EDIT

Motores DC

last edited by [pabloevaristo](#) 13 years, 3 months ago

Page history

1. [Nota preliminar](#)
- i. [Conexiones con Arduino](#)

ii. [Power Supply](#)
2. [DEFINICIÓN](#)
3. [VELOCIDAD](#)
4. [SENTIDO](#)
5. [REPOSO](#)
6. [EJEMPLO](#)

Nota preliminar

La librería AFMotor.h para la se puede descargar de http://www.ladyada.net/media/mshield/AFMotor_18-2-09.zip
Hay que copiar la carpeta que se descomprime (AFMotor) en ~/arduino-0013/hardware/libraries.

Conexiones con Arduino

Cuando utilizamos la placa MOTOR_SHIELD, algunos de los contactos de Arduino quedan ocultos, (e/s analógicas/digitales) sin embargo se pueden hacer algunos arreglos (ver fotos).

De todas maneras aunque consigamos tener accesibles todas las e/s, hay algunas que son utilizadas por la placa motorshield y que por tanto no debemos utilizar:

Digital pin 2 LIBRE
Digital pin 3: DC Motor #2 / Stepper #1. Sólo si se usa el motor 2/1
Digital pin 4: Sólo ocupado si se usa los motores DC/Stepper
Digital pin 5: DC Motor #3 / Stepper #2. Sólo si se usa el motor 3/2
Digital pin 6: DC Motor #4 / Stepper #2. Sólo si se usa el motor 4/2
Digital pin 7: Sólo ocupado si se usa los motores DC/Stepper
Digital pin 8: Sólo ocupado si se usa los motores DC/Stepper
Digital pin 9: Servo #1. Sólo ocupado si se usa el servo1
Digital pin 10: Servo #2. Sólo ocupado si se usa el servo2
Digital pin 11: DC Motor #1 / Stepper #1
Digital pin 12: Sólo ocupado si se usa los motores DC/Stepper
Digital pin 13 LIBRE

Las entradas analógicas (ANALOG IN) pueden ser utilizadas como pins digitales (referenciándolas como pines del 14 al 20).

Power Supply

Para que los motores funcionen correctamente necesitamos mantener un nivel de corriente considerable. Tenemos varias opciones:

1.- Utilizar la corriente de Arduino:

No se recomienda puesto que además de reducir considerablemente la vida de la batería principal, puede haber problemas con la ejecución del código sobre Arduino, y malfuncionamiento para el reseteo de la placa. Para ello deberemos utilizar el jumper (PWR) del motorshield.

2.-Añadir una fuente de corriente externa (recomendado):

Tags: [arduino](#), [motor shield](#), [motores DC](#)

Check for plagiarism

To join this workspace, [request access.](#)

Already have an account? [Log in!](#)

Navigator

Archivos de texto-Guías

ArduinoMusical

MotoresPasoApaso

Practicas con arduino

Ruleta

analogRead

PagesFilesoptions

SideBar

INDICE

1. [INTRODUCCIÓN](#)
2. [INSTALACIÓN DEL ENTORNO DE PROGRAMACIÓN](#)
3. [INICIO CON LA CONTROLADORA ARDUINO](#)
4. [CONSTRUCCIÓN DEL ROBOT](#)
5. [PROGRAMACIÓN DEL ROBOT](#)

i. [Estructura de un programa Arduino hecho](#)

ii. [Variables hecho](#)

iii. [Operadores de comparación hecho](#)

iv. [Control de flujo](#)

v. [Funciones Digitales](#)

vi. [Funciones Analógicas](#)

vii. [Control de motores](#)

viii. [Comunicación en serie](#)
6. [PARA SABER MÁS](#)

Recent Activity

Practica5_Pulsador_ como_ interruptor

edited by cmartinez

Practica5_Pulsador_ como_ interruptor

edited by cmartinez

Practica5_Pulsador_ como_ interruptor

edited by cmartinez

Practica5_Pulsador_ como_ interruptor

edited by cmartinez

FrontPage

edited by pabloevaristo

SensoresLDR

edited by pabloevaristo

FrontPage

edited by cmartinez

Alimentaremos de forma independiente la Arduino y los motores. Debemos eliminar el jumper(PWR) de la motorshield y alimentar la Arduino de forma independiente (USB o Vin); y conectar una fuente externa al conector EXT_PWR del motorshield.

En cualquiera de los 2 casos deberemos estar atentos al led del motorshield, si no está encendido, algo va mal.

DEFINICIÓN

AF_DCMotor nombre_motor(num, MOTOR12_freqKHZ);

- **nombre_motor**: el nombre que queramos darle al motor.
- **num**: número del 1 al 4 que indica el conector que utilizaremos en la placa.
- **freq**: puede ser 64, 12, 8 o 1. A más frecuencia, más exactitud y menos ruido pero más consumo.
- Ejemplo: **AF_DCMotor motor2(2, MOTOR12_64KHZ);** crea un motor llamado motor2 que se deberemos poner en el conector número 2 y que cuya frecuencia de PWM será de 64KHz.
- **NOTA**: para los motores 1 y 2 podemos utilizar cualquier frecuencia, pero para los motores 2 y 4 sólo se usa la frecuencia de 1kHz, cualquier otra frecuencia indicada en la función se desechará.

VELOCIDAD

nombre_motor.setSpeed(num);

- **nombre_motor**: el nombre con el que hemos definido del motor.
- **num**: número del 0 al 255 que indicará el nivel de voltaje de salida aplicado al motor: 0 mínimo, 255 máximo.
- Ejemplo: **motor2.setSpeed(100);** hace que al motor que hemos definido como motor2 se le aplique un nivel de voltaje de 100, siendo el máximo 255.

SENTIDO

nombre_motor.run(SENTIDO);

- **nombre_motor**: el nombre con el que hemos definido del motor.
- **SENTIDO**: la palabra FORWARD o BACKWARD según queramos que el motor gire en un sentido o en otro.
- Ejemplo: **motor2.run(FORWARD);** hace que al motor que hemos definido como motor2 gire hacia adelante.

REPOSO

nombre_motor.run(RELEASE);

- **nombre_motor**: el nombre con el que hemos definido del motor.
- Ejemplo: **motor2.run(RELEASE);** hace que al motor que hemos definido como motor2 se pare.

EJEMPLO

```
#include <AFMotor.h>
// La librería AFMotor.h para la se puede descargar de http://www.ladyada.net/media
// Hay que copiar la carpeta que se descomprime en ~/arduino-0013/hardware/libraries
// para probar una placa arduino motor shield, simplemente cargar el programa,
// alimentar la placa arduino con el USB y seleccionar USB como fuente de alimentaci
// pero conectar una fuente DC con al menos 4.5 V al conector de la placa arduino
// y poner el jumper en los pines PWR de la placa del motor de forma que la placa
// de motor se alimente de la fuente DC.
// Después, simplemente conectar un motor DC a las distintas clemas y probar si gira
// un lado durante 1 segundo, para 2 segundos, gira hacia el otro lado 1 segundo, pa
AF_DCMotor motor1(1, MOTOR12_64KHZ); // crea motor #1, 64KHz pwm
AF_DCMotor motor2(2, MOTOR12_64KHZ); // crea motor #2, 64KHz pwm
AF_DCMotor motor3(3, MOTOR12_64KHZ); // crea motor #3, 64KHz pwm
AF_DCMotor motor4(4, MOTOR12_64KHZ); // crea motor #4, 64KHz pwm
void setup() {
  Serial.begin(9600);          // set up Serial library at 9600 bps
  Serial.println("Motor test!");
  // PONEMOS LAS VELOCIDADES
```

```
motor1.setSpeed(135);    // velocidad a 135 - el máximo sería 255
motor2.setSpeed(135);
motor3.setSpeed(135);
motor4.setSpeed(135);
}
void loop() {
  Serial.print(" derecha");
  motor1.run(FORWARD);    // turn it on going forward
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);
  delay(1000);
  Serial.print(" paro");
  motor1.run(RELEASE);    // stopped
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
  delay(1000);
  Serial.print(" izquierda");
  motor1.run(BACKWARD);   // the other way
  motor2.run(BACKWARD);
  motor3.run(BACKWARD);
  motor4.run(BACKWARD);
  delay(1000);

  Serial.print(" paro");
  motor1.run(RELEASE);    // stopped
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
  delay(1000);
}
```



Comments (0)

You don't have permission to comment on this page.

[Printable version](#)

[PBworks](#) / [Help](#)

[Terms of use](#) / [Privacy policy](#) / [GDPR](#)

[About this workspace](#)

[Contact the owner](#) / [RSS feed](#) / This workspace is **public**