

# Introducción: Que es Anakino\_Aquarium



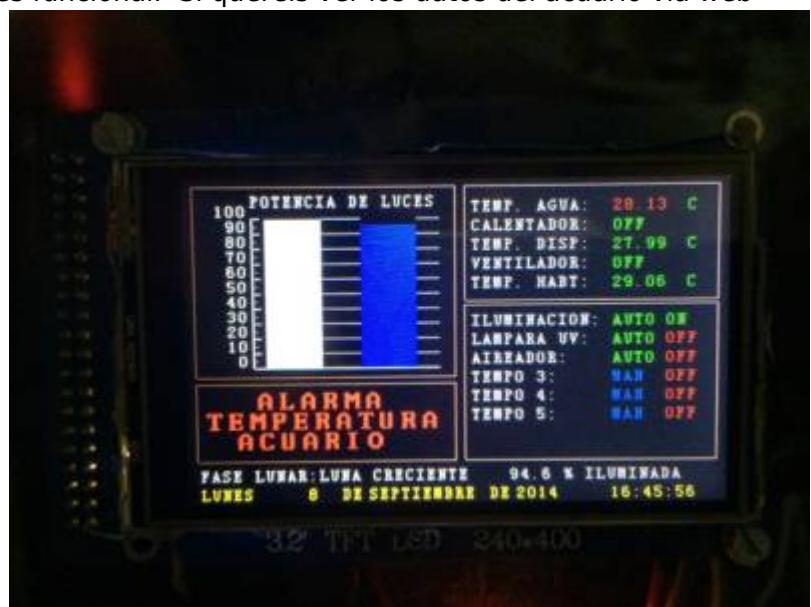
Este proyecto está discontinuado. Ahora estoy con una versión mas sencilla ya que solamente utiliza una pantalla LCD de 20x4.

El proyecto Anakino\_Aquarium es un proyecto basado en el código de Ferduino, pero adaptado a un acuario de agua dulce.

Únete al grupo de telegram en: [https://t.me/Anakino\\_aquarium](https://t.me/Anakino_aquarium)

Aunque no esta terminado, de momento es funcional. Si quereis ver los datos del acuario via web debéis registraros en la web Xively.

También necesitais la ethernetShield que conecta el Arduino a internet.



A continuación voy a explicar como podeis montar un controlador básico de un acuario de agua dulce con un arduino mega.

En primer lugar agradecer a Fernando Garcia todo su trabajo, ya que todo el código es suyo, yo solo he modificado lo necesario para que funcione con un acuario de agua dulce.

Se supone que llegados a este punto tenemos unos conocimientos básicos de electrónica o de Arduino en concreto. Un resumen muy sencillo sería; Arduino es una placa con un chip y una memoria interna que es capaz de guardar y ejecutar una serie de instrucciones escritas en un sketch. Arduino con esas “instrucciones” realizará lo que nosotros le hayamos escrito y en función de diversos valores.

En nuestro caso lo que hará el controlador será:

- Mantener la temperatura del agua del acuario a una prefijada por nosotros con la variación que le hayamos permitido. Por ejemplo, marcamos 25.5º y le damos 0,5º de margen de temperatura.
- Configurar el encendido y el apagado de las luces blancas, rojas, azules, etc y activar la luz de luna real según el día del mes y año en el que nos encontramos.
- Regular la velocidad de los ventiladores del disipador en función de su temperatura.

- Configurar hasta 5 temporizadores (para controlar Lampara UV, bomba de aire )
- Conocer el estado de las variables del acuario desde cualquier pc o teléfono a través de internet.
- Cambiar algunos valores mediante una aplicación móvil

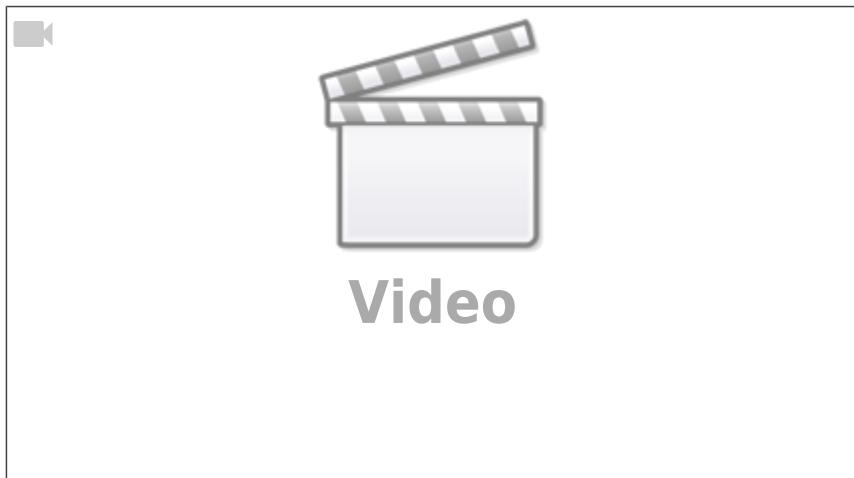
En un futuro: manejar comederos temporizados.

Todo esto será controlado y manejado con una pantalla tactil a color, donde podremos ver los valores del acuario.

Para poder empezar a trabajar necesitamos que tengais descargado el ultimo IDE de arduino, os recomiendo la 1.0.4 de aqui: <http://arduino.cc/en/Main/Software>

Todos los archivos los teneis para descargar de aqui: <https://github.com/Anakinspain>

Aqui teneis un video explicando el funcionamiento:



## Capítulo 1: Componentes

Paso a poner el listado de los materiales necesarios:

Material	Enlace	Precio
Arduino Mega 2560	<a href="#">Ebay</a>	12/25€
Ethernet Shield con chip W5100	<a href="#">Ebay</a>	13 / 25€
TFT tactil	<a href="#">Itead</a>	pte confirmar nuevo modelo
Placa de 8 reles.	<a href="#">Ebay</a>	10€
Sensores de temperatura DS18B20	<a href="#">Ebay</a>	3x 4€
3 TIPS 141	<a href="#">Ebay</a>	3x3€
Reloj RTC DS1307	<a href="#">Ebay</a>	5€
Cables y conectores		
Caja para meterlo todo		

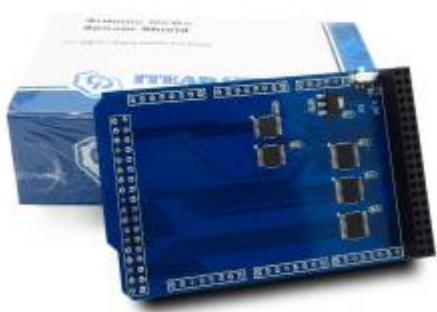
## Tactil 240x400

Esta pantalla es un poco mas grande que lo que normalmente se vende pero es el modelo de pantalla que el que creado el codigo que quiero instalar utiliza.



Es tactil, la comre en: <http://imall.iteadstudio.com/im120419006.html>

Junto a la pantalla pedi tambien el controlador. Para el arduino mega es necesario.  
<http://imall.iteadstudio.com/im120417024.html>



## Iluminación

En mi caso, estoy utilizando iluminación mediante tubos T5. Estos tubos están conectados a uno de los relés, que activan o desactivan los 220v del enchufe. Aunque estoy esperando una pantalla de LEDS que tiene muy buena pinta...

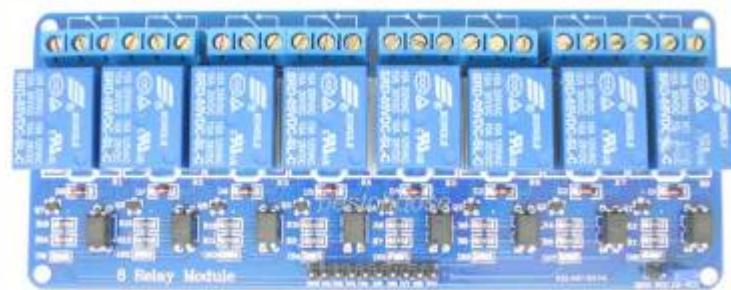
De todos modos, para el que use leds, se puede configurar el amanecer y el anochecer. Para ello, tienen que conectar el cable de salida de la fuente de alimentación a la entrada del TIP y la salida del tip irá conectada a los leds.

En la última version he dejado he abandonado la posibilidad del manejo de los canales, dejando solo el control de la luz de día y noche.

De todos modos, aun podeis encontrar el codigo usado en: [GITHUB](#)

## Caja de reles

Siguiendo el consejo de Fernando Garcia, decidí poner la caja de relés fuera de la caja del Arduino.



Para poder activar o desactivar circuitos externos mediante el arduino es necesario contar con una placa de reles. Yo utilizo una placa de 8 reles a 5v para que con la tension de trabajo del arduino sea suficiente.

De esta manera, podemos gestionar aparatos que funcionen a 220v.

Para esto he utilizado un cable de 10 hilos y un conector D-Sub macho y hembra para conectar la caja de relés con la caja donde va el Arduino.

El conector de la derecha lo tenéis que dejar en modo JD-VCC, luego tenéis que conectar el pin de la izquierda del todo GND a masa, y el VCC a 5v. Cada uno de los pines de IN1, IN2, es para activar cada uno de los ocho reles y deben unirse al pin del Arduino que los active.

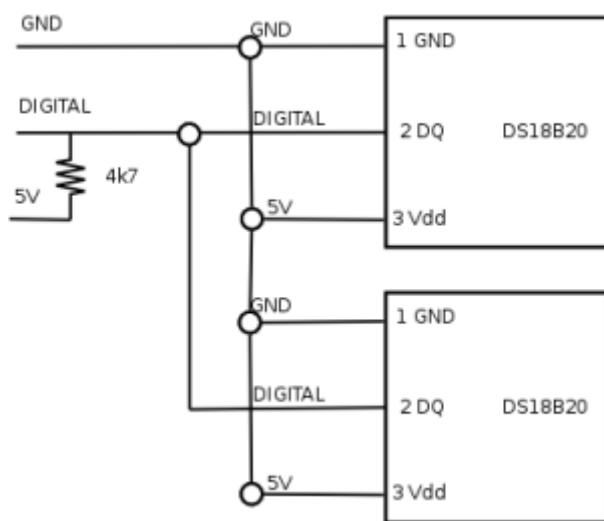
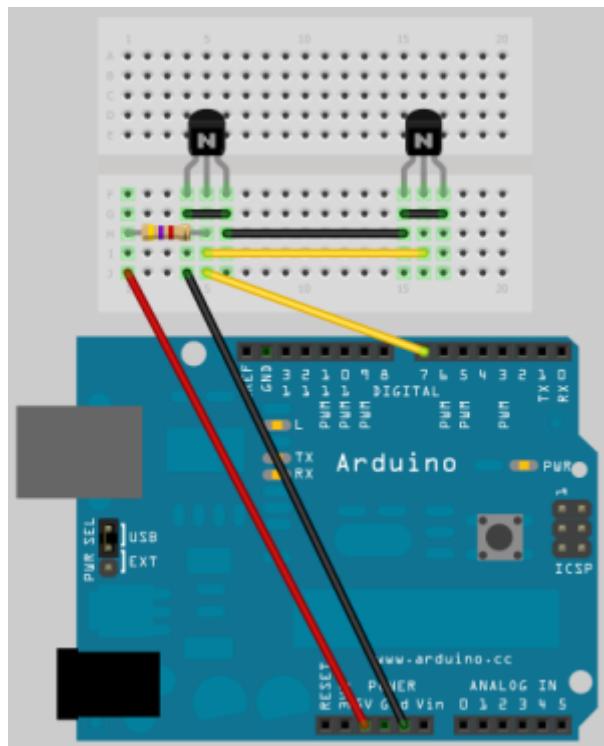
En mi caso compro la placa de reles en eBay y me la enviaron desde china, y como son así funciona al revés, es decir, si quiero activarlo debo decirle LOW y si quiero desactivar el rele debo decir HIGH. De todos modos en el código se ha incluido una función para que no nos compliquemos y que ya explicaré mas adelante.

En la parte trasera de la caja de Arduino es donde tengo la conexión de los relés.

## Sensores de temperatura

Como sonda de temperatura lo mejor que hay es el integrado de 3 patillas DS18B20 de la casa Dallas/Maxim es un termómetro digital de 9 a 12 bits que se conecta a través de un bus 1-wire, mide la temperatura en grados Celsius, tiene una operación de temperatura que va del rango de -55°C a +125°C.

El DS18B20 es un circuito integrado con forma de transistor, tiene tres pines de los cuales dos sirven de alimentación (GND y Vdd) y el otro (DQ) es el que se conecta al bus. Este sensor puede funcionar también sin alimentación, con lo que se conoce como alimentación parásita. En este modo de funcionamiento la alimentación la saca directamente de la línea DQ por lo que solo se necesitan dos cables (DQ y GND). Cada uno de ellos dispone de un identificador único de 64 bits, lo que permite que puedan coexistir varios sensores en un único bus 1-wire. El bus 1-wire puede ser de hasta 100 metros de largo por lo que la colocación de los sensores respecto del módulo no debería ser un problema, además no requiere de elementos externos, el cable de conexión tiene que ser apantallado.



Para este código estamos usando tres sensores de temperatura. Uno de ello, nos tiene que dar el valor del agua del acuario, y en función del valor captado lo compara con el valor deseado y activa o desactiva el rele del calentador.(Es importante, poner el regulador de la temperatura del calentador un poco por encima de la deseada, de esta manera, si el Arduino activa el enchufe, el calentador debe funcionar correctamente).

Los tres sensores van conectados en serie, y tienen el PIN 42 por el que Arduino lee los datos.

## Reloj

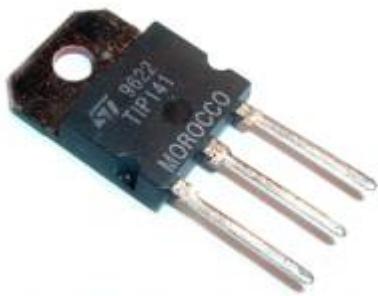
El reloj es necesario ya que el Arduino en si no conoce la hora, así que mediante este pequeño circuito se puede configurar la fecha, necesaria para la fase de la luna y para la configuración de los temporizadores.

El reloj lleva una pequeña pila de botón para que en caso de fallo de la luz o reinicio no se pierda la configuración de la hora.

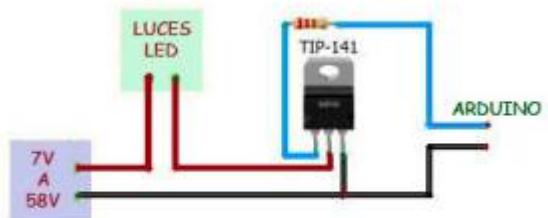
## TIP141

En mi instalación actualmente solo se usa un TIP para regular la luz de luna ya que uso tubos fluorescentes.

De todos modos, un TIP seria para regular la luz nocturna (azul), otro seria para la luz blanca y el ultimo sería para regular la velocidad de los ventiladores.



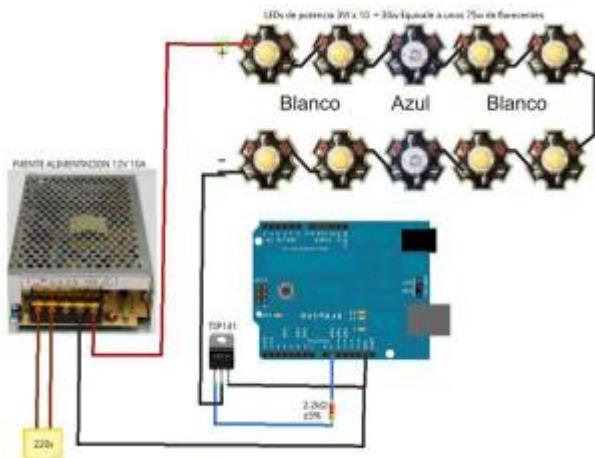
El tip 141 es un transistor de potencia, tienen multiples usos, pero en este caso lo que hace es regular la intensidad o corriente con la que alimentas los led. Tienen 3 patillas, una se llama Base, otra es el colector y la tercera es el emisor.



El colector esta conectada directamente con la v+ de la fuente y con una entrada analogica del arduino. EL emisor va conectado directamente a los led.

La base va conectada a una de las salidas analogicas del arduino que va ha 5v y entre 0-150mA, con una resistencia, que su función es que a la base del transistor no le llegue más de 3v,para que no

casque, pero la intensidad de la base 0-150mA, si varia.



La tensión y intensidad salen del v+ de la fuente 24v y 750mA, van al colector del transistor y a una entrada analoga del arduino(el arduino necesita saber la tensión y intensidad de la fuente para regularla y compararla).

El arduino según lo que hayas programado en el encendido de los led , manda una señal a la base del transistor, que pasa a través de la resistencia y será por ejemplo 3v y 10mA, más tarde 3v y 15mA, 3v y 20mA, etc. Por el emisor del transistor saldrá un voltaje y una intensidad 24v y 300mA, que están relacionados al voltaje e intensidad del colector y la que aplique el arduino en la base del transistor, según valla aumentando la señal de la base que aplica el arduino por el emisor irá aumentando la intensidad que van a los led desde la fuente, 24v y 325mA, 24v y 350mA, 24v y 400mA, 24v y 500mA, ..., 24v y 750mA.

Viene a funcionar muy parecido salvando las distancias a una llave de paso de agua, por donde nos llega el agua es el colector, la base haría la función de ser la llave de apertura y cierre, dejando pasar más o menos agua, nosotros seríamos el arduino cerrando y abriendo la llave de paso(base) y la salida de la llave de paso sería el emisor.

## Fuente de alimentación

Para mi proyecto uso dos fuentes de alimentación.

- La primera es un cargador de USB a 220v, en el conecto una toma con la que alimento la pantalla, el Arduino, los relés y el reloj. Es importante tener en cuenta que ese transformador debe tener al menos 2A para no tener problemas, y que todo va conectado directamente al USB mediante una regleta y no al Arduino, para evitar sobrecargar la placa.
- En segundo lugar es un transformador de 220v a 12v con el que se da corriente a la luz de leds azules y a los ventiladores.

## Capítulo 2: Pantalla utilizada

Actualmente la tecnología del led parece que se va abriendo paso poco a poco, en el mundo de la iluminación en general y en la acuarofilia no podía ser menos.

Ante este nuevo sistema podemos encontrar:

## Ventajas

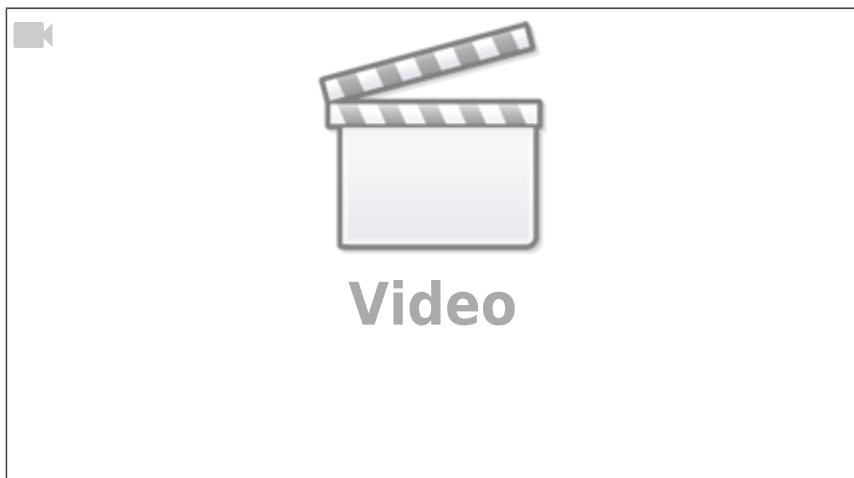
- Iluminación más focalizada ya que toda la luz se emite en 120º (Si no se usa lente)
- Economía de consumo eléctrico al necesitar menos electricidad.
- Vida útil de la luz (unas 50.000Horas)
- Amplio espectro lumínico para las plantas.

## Inconvenientes:

- Coste inicial (aunque luego se ve rentabilizado con el consumo de la luz)
- Complejidad en conseguir el equipo perfecto si lo monta uno mismo
- Necesidad de utilizar disipadores de calidad (caros)

Después de hacer varias pruebas con pantallas caseras LEDs, al final, me he decidido por comprar una fabricada ya que así me evita problemas. La primera prueba que hice fue comprar tiras de leds enfundadas para evitar que se estropearan con la humedad, pero teniendo en cuenta que están 8 horas encendidas todos los días, al final se estropearon y dejaban de iluminar como deberían. Posteriormente, compré leds de 3W con los que hacer una pantalla sobre un disipador, pero nada, seguía sin funcionar ya que estos leds acabaron por estropearse por la humedad.

Así que la pantalla elegida es la: FLUVAL AQUALIGHTS & PLANTS en 35W.



La ventaja de esta pantalla es que al ser a 24v podemos usarla para simular el efecto amanecer y anochecer con el controlador.

Usaremos una fuente de alimentación de 24v que conectaremos a una placa que será controlada por el arduino y así regularemos los ventiladores, y luz de luna con la misma fuente.

## Capítulo 3: Conexiones

Hay que tener en cuenta que la placa Arduino es la base de todo el control por lo tanto hay que buscarle un sitio un poco centrado para que todas las conexiones sean cómodas. Yo he usado una caja de metacrilato que tenía de un antiguo PC para montar todo.

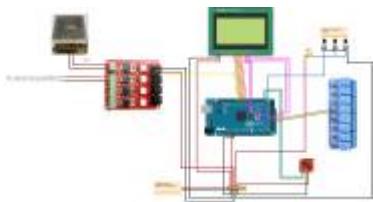
La parte trasera de la caja es donde están las conexiones, de esta forma, la caja es como una minitorre de PC que puedo desconectar por la parte trasera por si tengo que trastear en ella.

Como podeis ver estan las conexiones de la entrada de voltaje de 12V, las de salida para alimentar los leds (en caso de que se instale), de ventiladores, las conexiones de la placa de relés, y el conector de los sensores.

En la parte delantera hice dos agujeros, uno para la pantalla y otro para un conector USB para poder cargar el IDE del Anakino\_Aquarium de manera mas cómoda.

En el interior hay una placa donde tengo la entrada de 5v. Esa placa es como un distribuidor con el que alimento de 5v todos los componentes que lo necesitan como el Arduino, el reloj, la placa de relés, y así no sobrecargo el Arduino.

Aqui teneis una primera version para que vayas haciendo una idea de como está todo conectado



## Capítulo 4: Software

- **1.- IDE de Arduino:** Necesitamos instalar un software, se llama IDE, para poder cargar el “programa”. Recomiendo descargarlos la version 1.0.4 que es la compatible con el sketch que se usa. Descargar de [AQUI](#)
- **2.- Librerías:** Una vez instalado el software necesitamos copiar las **librerías** que necesitamos para que se ejecuten en el Arduino todas las instrucciones del código. Se descargan desde aqui: <https://github.com/AnakinSpain/Libreria>

Se deben descomprimir y copiar dentro de la ruta de instalacion del Software IDE sobre escribiendo las librerias anteriores.

- En Windows copiamos la carpeta de la librería a la siguiente ruta:  
**Documents\Arduino\libraries**
- En MAC copiamos la carpeta de la librería a la siguiente ruta:  
**Documents/Arduino/libraries**

Una vez que se ejecuta por primera vez el IDE, debemos configurar la tarjeta, en nuestro caso Mega 2560.

- **3.- Software Anakino\_Aquarium:**  
[https://github.com/AnakinSpain/Anakino\\_Aquarium/archive/master.zip](https://github.com/AnakinSpain/Anakino_Aquarium/archive/master.zip)

Para poder subirlo al IDE debemos descomprimir el archivo master.zip, luego debemos renombrar la carpeta “Anakino\_Aquarium-master” a “Anakino\_Aquarium” es decir, quitarle el -master al nombre.

Una vez dentro, solamente debemos ejecutar el archivo llamado: Anakino\_Aquarium.ino o bien abrirlo desde el IDE de Arduino.

Automáticamente se generarán las pestañas diferentes que tiene el código. Está así para tratar de organizarme mejor.

Una vez hecho esto solamente queda elegir el puerto correcto de comunicaciones y darle a subir para que "instale" el código en el Arduino.

## Capítulo 5: Configuración tweeter

Podemos configurar el controlador para que nos envía a través de una cuenta de tweeter avisos en forma de tweet cuando los parámetros del acuario estén fuera de valores para que nos avise.

Lo primero es crear una cuenta en tweeter. Una vez que esté creada, debemos autorizar a tweeter a que la aplicación pueda enviar tweets mediante el enlace este: [Enlace aplicación](#)

Tras autorizar a la aplicación, obtenemos el token que buscábamos:



Lo único que se debe hacer en el código es copiar ése código y pegarlo en la línea de tweeter del código

## Capítulo 6: Otros Software

Existen otros controladores en internet de un acuario con Arduino. Alguno de ellos están desactualizados.

### Dukatus

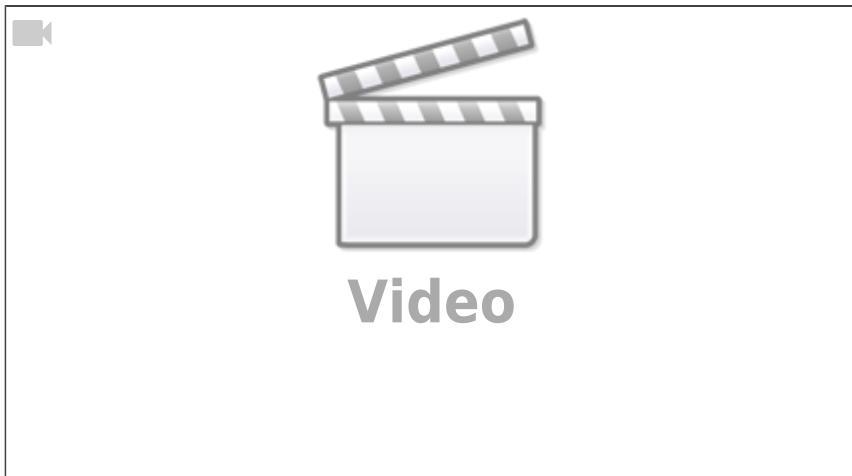
Archivo con información sobre el código de dukatus.

[controlador-acuario-con-arduino-dukatus.pdf](#)

### Ferduino

Visita la pagina de Ferduino.

[www.ferduino.com](#)

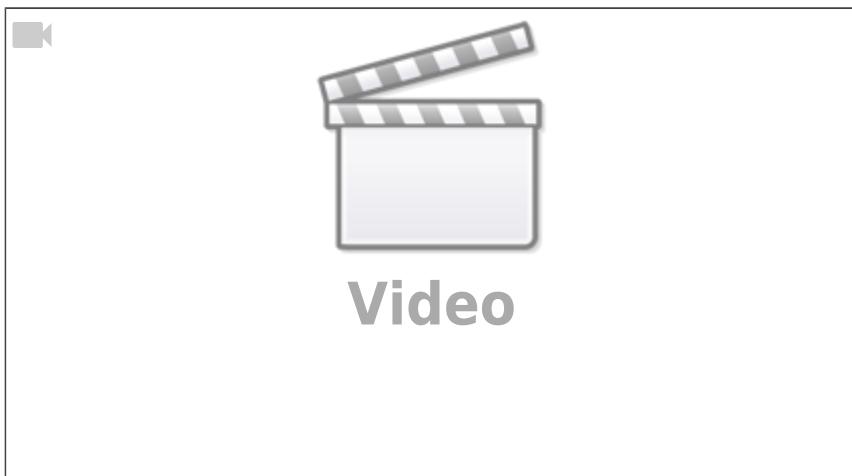


## Jarduino

Aqui tienes el manual traducido para Jarduino v 1.2



[jarduino-manual-de-usuario-v1-2-beta.pdf](#)



From:  
<https://www.atorcha.es/> - **Atorcha**

Permanent link:  
[https://www.atorcha.es/anakino\\_aquarium\\_v1?rev=1666747295](https://www.atorcha.es/anakino_aquarium_v1?rev=1666747295)

Last update: **20:07 19/05/2025**

