

MICRO CONTROLADOR ARDUINO | TALLER DE 1 CRÉDITO

Etapas del Curso

Etapa 1: Descarga el programa desde <http://arduino.cc>.

Etapa 2: Hacer que la luz parpadee

Etapa 3: Leer la luz parpadeante del trabajo de alguien mas

Etapa 4: Gráficamente expone la información obtenida desde tu sensor de luz

Etapa 5: Leer un sensor de presión

Etapa 6: Aprender a operar un motor usando PWM y un puente-H

Etapa 7: Aprender a controlar los motores vía retroalimentación desde el sensor de presión

Etapa 8: Aprender a impermeabilizar motores (opcional)

Etapa 9: Intentar impermeabilizar el mini Arduino

Etapa 10: Crear un mini ROV que pueda alcanzar y mantener una profundidad especificada.

Etapa 1.- Descarga el programa desde <http://arduino.cc> y crea tu plataforma de desarrollo

Objetivos:

- Aprender un poco sobre Arduino y Micro controladores
- Crear tu plataforma Arduino de desarrollo
- Comenzar tu proceso

Paso 1:

Antes de descargar el programa, lee un poco acerca de la plataforma Arduino. En tu diario, discute brevemente algo interesante que aprendas acerca de Arduino a medida que vayas leyéndolo.

Paso 2:

Nuevamente, antes de descargar el programa, anda a <http://Arduino.cc/en/Guide/HomePage> y escoge tu sistema operativo y sigue las instrucciones para “Preparando el comienzo de Arduino”

Paso 3:

Durante el proceso de instalación, asegúrate de discutir cualquier problema que encuentres durante la instalación. Si no encuentras ningún problema, escribe una nota de cuan fácil fue instalar el programa Arduino.

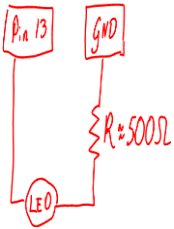
Paso 4:

Adjunta el Arduino Uno al tablero de desarrollo usando los tornillos que se entregan. Adjunta el tablero de circuitos al tablero de desarrollo a través de la almohadilla adhesiva. Si es posible y cuando sea apropiado, es siempre buena idea incluir fotos de tus progresos en tu informe.

Etapa 2.- Hacer que la luz parpadee**Objetivos**

- Aprender como crear un programa Arduino (conocido como un bosquejo),
- Aprender como cargar un programa Arduino sobre el tablero Arduino,
- Hacer una destello LED de una manera controlada.

Paso 1: Crear el siguiente circuito usando tu tablero de desarrollo Arduino. Como siempre, pide ayuda cuando la necesites.



Nota que en este caso estamos usando un Resistor como un limitador de corriente. Una vez que busques las especificaciones de tu dispositivo, tu puedes escoger un resistor que te de un voltaje deseado a través de $V=IR$. En nuestro caso nuestro en la hoja de especificaciones del LED se sugiere una corriente de 1mA. El voltaje de Arduino es 5V. Entonces $R = \frac{V}{I} = \frac{5V}{0.001A} = 5000\Omega$.

Paso 2: Conectar el Arduino en el puerto USB de la computadora. Si el puerto USB esta funcionando correctamente, la luz verde “ON” debería encenderse. La primera vez que hagas esto, la computadora también podría instalar los controladores.

Paso 3: Abrir Arduino.exe para escribir un nuevo boceto.

Paso 4: Asegurarse que debajo de “Herramientas” selecciones el Tablero correcto. También debajo de “Herramientas” asegúrate de seleccionar el puerto COM correcto- simplemente puedes suponer cual es el puerto correcto y cambiar de puerto si el primero no funciona, o puedes ir al panel de control de tu computadora para averiguar cuales puertos están funcionando.

Paso 5: Escribe tu programa en el área de boceto. Nota que hay varios ejemplos de programas debajo de “Archivo”, “Ejemplo”. Aquí hay un programa “blink” muy básico- una variación de este

programa debería estar disponible en el menú Ejemplo- o, puedes escribirlo tu mismo:

```
/*Blink
```

```
Enciende y apaga un LED repetidamente.
```

```
*/
```

```
void setup () {
```

```
// Primero, debes inicializar el pin digital como una salida.
```

```
// Nota que Pin13 tiene un LED conectado en la mayoría de los  
tableros Arduino:
```

```
pinMode (13, SALIDA);
```

```
}
```

```
void loop () {
```

```
digitalWrite(13, HIGH); // activar el LED
```

```
delay(1000);           // esperar por 1000ms
```

```
digitalWrite(13, LOW); // desactivar el LED
```

```
delay(1000);          // esperar por 1000ms
```

```
}
```

Paso 6: Sube tu programa a Arduino haciendo click en “Cargar”. A medida que Cargues el programa, deberías notar que uno de los LEDs puestos en el tablero comienza a parpadear.

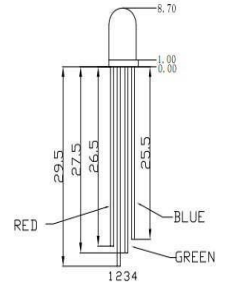
Paso 7: Para convencerte que realmente estás haciendo que la luz parpadee en la forma que lo debe hacer, intenta reprogramar el Arduino para cambiar la frecuencia de parpadeo en una variedad de formas.

Paso 8: Si el tiempo lo permite, consigue o compra un LED de 3 colores y diviértete experimentando.

Solución de problemas:

Algunos problemas se pueden resolver a través de...:

- a) asegurándote que tu computadora tenga los derechos de administrador para permitir el funcionamiento y la instalación de los programas.
- b) Yendo al “administrador de dispositivos” y asegurándote que Arduino esta trabajando



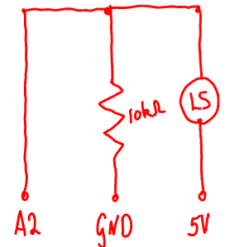
Etapa 3: Leer la luz parpadeante del trabajo de alguien mas.

Objetivos:

- Aprender como usar un Arduino para obtener información desde un sensor

En esta etapa vamos a usar el controlador Arduino para detectar la intensidad de luz que hay en un sensor de luz.

Primero, desconecta tu Arduino de la fuente de poder y/o el computador y saca cualquier componente que pongas durante la Etapa 2.



Luego, re-crea el circuito mostrado en el dibujo. A2 lo puedes encontrar en los pines de entrada Análogos que se encuentran en el tablero Arduino.

Antes que hayas hecho el circuito, escribe en el programa de abajo y sube el programa a Arduino.

```
/*Este programa obtendrá información desde un sensor de luz que esta
conectado en el tablero Arduino
int LS=2;    // define LS como la variable que le dirá al programa de cual
pin reunir información
int LSvalue= 0; // LSvalue será la variable usada para almacenar el valor
que salga desde el sensor

void setup() {
//puerto de serie abierta.
Serial.begin(9600);
pinMode(LS, INPUT); // hazle saber a la computadora que pin LS será un
pin de ENTRADA
}
void loop() {
LSvalue=analogRead(LS); // establece LSvalue igual al valor del sensor
conectado en el pin LS.
Serial.println(LSvalue); // Esto imprimirá el valor de 'LSvalue' a la Serie del
Monitor
delay(30); // esto reduce la velocidad de las cosas un poco
}
```

Cuando ejecutes el programa vas a necesitar ir a “Herramientas” y seleccionar “Serial Monitor” para ver la información siendo enviada a tu computadora a través de Arduino.

1. Cuando hayas cargado el programa a tu Arduino, deberías tener en cuenta intentar bloquear la luz al sensor de luz t luego

permitir que la luz llegue al sensor nuevamente y observar que sucede.

2. Intenta usar el sensor de luz para detectar luz proveniente del Arduino de alguien mas que este tratando de ejecutar el programa de parpadeo.

3. Algo de física: Reemplaza el sensor de luz con un LED. Intenta detectar el LED de alguien mas que este usando el mismo color LED que tu y luego intenta detectar un color de LED diferente.

4. Si el tiempo lo permite, explora la función del mapa. Con una luz suficientemente brillante, deberías ser capaz de obtener la serie de saluda para leer valores tan altos como 1,023. Una luz tan brillante, sin embargo, no podría ser posible sin establecer el laboratorio***. En tu programa Arduino puedes usar la función de mapa para medir la salida de tu variable. Por ejemplo `porciento=map(LS, 0,1023,0,100)` seria un valor almacenado en LS y se convertir así un porcentaje de un valor máximo de 1023.

Etapa 4: Exponer gráficamente la información obtenida de tu sensor de luz.

Objetivos:

- Entender la diferencia entre un Boceto Arduino y un programa VPython.

- Obtener el Arduino para enviar información a un programa VPython.
- Leer suficiente VPython para ser capaz de crear una pantalla de visualización para la representación de la información obtenida.

En esta etapa, vamos a usar VPython para exponer gráficamente la información obtenida desde tu sensor de luz. Este va a ser una etapa difícil de llevar a cabo, por lo tanto asegúrate de pedir ayuda cuando no entiendas muy bien lo que esta sucediendo.

Una breve descripción de VPython: Este es un lenguaje de programación basado en el bien conocido y ampliamente usado lenguaje de programación Python. La principal ventaja de VPython es su rápida forma de aprendizaje y su facilidad relativa en la creación de gráficos y objetos 3D en la pantalla.

VPython es una fuente abierta de lenguaje y puede ser descargado en

<http://www.vpython.org>.

Instrucciones para descargar e instalar VPython.

- 1) Anda a la sección de descarga apropiada de <http://www.vpython.org>. y sigue cuidadosamente las instrucciones para descargar la versión 2.7.
- 2) Toma en cuenta que primero tienes que descargar Python-2.7.2 y solamente después de instalar Python, puedes descargar VPython2.7.

3) Solamente después de que Python y VPython hayan sido descargados, instalados y probados, lo siguiente es instalar PySerial. Un instalador de Windows automático para PySerial que lo puedes encontrar en:

- . <http://sourceforge.net/projects/pyserial/files/pyserial/2.5/>. Otra opción es ir a <http://pypi.python.org/pypi/pyserial> para descargar e instalar Pyserial. Por ultimo, una carpeta llamada “serial” necesita estar en un directorio que probablemente aparezca así c:\Python27\Lib\site-packages. Para computadores Mac podrías intentar <http://randomproblem.wordpress.com/2012/01/14/installing-pyserial-2-5-on-mac-os-x-to-python-2-x-and-3-x/>.

Luego, usa el programa de la Etapa 3 que hace que Arduino obtenga un sensor de lectura de luz y luego envía esa información al computador.

El siguiente programa VPython nos entregara una representación visual del sensor de lectura de luz.

Nota rápida: Asegúrate de apagar el ‘serial monitor’ en la computadora antes de abrir y ejecutar VPython.

```
### El programa VPython de abajo abre una conexión serial con Arduino
### y gráficamente expone el resultado del sensor de luz.
### Es importante que el Arduino Serial Monitor no este abierto en tu computadora
cuando
###ejecutes este programa.
```

```
import serial
from visual import*
import time
```

```
wire=serial.Serial("COM6",9600) # abre el serial port. También podrías necesitar
cambiar el COM port wire.timeout=30# duración del tiempo antes de chequear el port
nuevamente
strn=""
```

```
value=0
```

```
window= display(title='Photo Sensor Display', width=600, height=600, center=(0,0,0),
rango= (500,500,500))
```

```
bulb= sphere(pos=(0,0,0), color= color.green, radius=100,
opacity= 1) #crea una esfera para representar el LED
```

```
while True: #Crea un circuito continuo para leer valores de Arduino
strn=wire.readline()# lee el valor de serial port
print(strn)
value=int(strn)#convierte la lectura en cadena en un entero
print (value)
bulb.radius=value/2#asigna el radio de la esfera al valor/2
bulb.opacity=value/255.0#cambia la opacidad
```

Si el tiempo lo permite:

1) Modifica el programa VPython de modo que, en lugar de una esfera, veas un cilindro con una altura que represente brillo de luz detectado por el sensor de luz.

Etapas 5- Leer la Información de un Sensor de Presión

Objetivos para esta Etapa:

- Obtener Arduino para recolectar información desde un sensor de presión
- Exponer visualmente información del sensor de presión
- Resolver como usar el sensor de presión en el agua
- Crear un grafico en VPython (si el tiempo lo permite)

Paso 1: Crear un programa que lea el voltaje de un sensor de

presión. El sensor de presión A MPX4250AP-ND ha sido incluido en tu equipo. Funciona de manera similar como lo hizo tu sensor de luz. Así, vas a crear un circuito parecido al de sensor de luz. No vas a necesitar el resistor de 500Ω porque el resistor está construido en el sensor de presión.

Paso 2: Crea una salida gráfica que sea visualmente agradable en la que se puedan ver lecturas de presión.

Paso 3: Aquí está el primer paso real en la construcción de tu ROV (vehículo remotamente operado) submarino en donde tu vas a tener que tomar en cuenta nuestro enemigo, el agua. Tu sensor de presión necesita detectar la presión del agua a cierta profundidad, y por su puesto, tu Arduino no se va a llevar muy bien en un ambiente sumergido. Así, eventualmente vas a tener que decidir si vas a mantener tu Arduino a prueba de agua (impermeable)

Algunas herramientas y componentes de impermeabilización que puedes usar son:

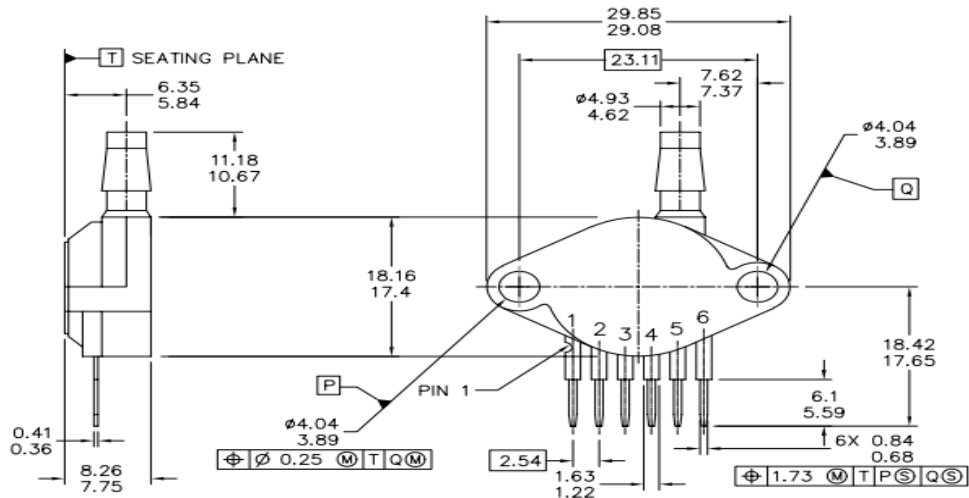
- Pistolas de pegamento
- Cable Ethernet
- Soldador

Paso 4: Si el tiempo lo permite, aprende algo más sobre VPython:

- Aprende como usar un "label" en vpython.org.
- Aprende como crear una salida gráfica VPython de tu información en

<http://www.vpython.org/webdoc/visual/graph.html>

Esquema de MPX4250



Pin 1: Vout | Pin 2: Ground | Pin 3: Vcc | Pin 4: V1 | Pin 5: V2 | Pin 6: Vex

Etapas 6: Aprender como ejecutar un motor usando PWM y un H-Bridge

Objetivos para esta Etapa:

- Aprender como adjuntar cabeceras a un chip.
- Aprender cual es la función de un HBridge (puerto H) y como usarlo.

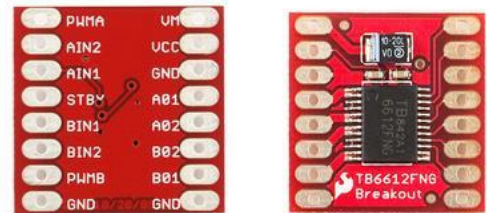
. Los siguientes materiales fueron adaptados de

<http://bildr.org/2012/04/tb6612fng-arduino/>.

Paso 1 : Preparar el HBridge. Vas a necesitar soldar las cabeceras en el HBridge.

Normalmente, tienes que soldar las cabeceras por lo tanto cuando las conectas en el tablero, el chip apunta hacia arriba permitiendo

la mejor posición del HBridge en tu tablero. Sin embargo, en este caso, obtendremos mejores resultados si insertamos las patas de la cabecera de modo que el

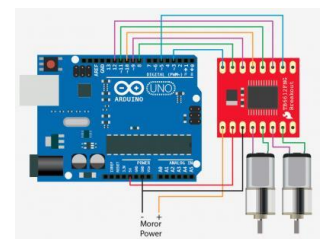


Cuando estés soldando, podría ser mas fácil si tu insertas las cabeceras en el tablero y poner el HBridge en la dirección deseada en las cabeceras. Esto permitirá una plataforma de soldadura estable.

Hay muchos sitios en línea donde puedes obtener datos, consejos y tutoriales para soldar. Un ejemplo es:

<http://www.aaroncake.net/electronics/solder.htm>

Paso 2: Instala el alambrado de tu Arduino como se muestra en la figura. Ten en cuenta que “Motor Power +” en el diagrama va en Vin en el Arduino.



Advertencia: Hay un par de combinaciones que te podrían dañar gravemente a ti o a tu

Vamos a usar el dual HBridge TB6612FNG. Un h-bridge es una combinación de transistores que te permite cambiar la dirección de la ejecución actual a través de tu circuito de motor. Esto te permite ejecutar un motor en ambas direcciones. Usando Pulsed Width Modulated (PWM), tu puedes ejecutar un motor a cualquier velocidad.

El TB6612 tiene 2 H-Bridge que te permiten ejecutar dos motores simultáneamente.

El TB6612 puede suministrar hasta 13V a 12^a continuamente.

Para conectar el TB6612 vamos a necesitar una fuente de poder externa en los rangos de 2.5V-13V. No podemos usar el pin 5V en el Arduino porque no puede proveer suficiente corriente para ejecutar los motores. **ADVERTENCIA:** si tu no conectas esto correctamente, fácilmente puedes quemar tu Hbridge y/o tu Arduino.

El 'Standby' pin en el TB6612 cuando presionas LOW apaga ambos motores.

Cada motor luego tiene 3 pins-dos de estos son para dirección, el tercero es para velocidad. Cuando el pin de una dirección esta HIGH y el otro esta LOW el motor va a girar en una dirección-cambiando el HIGH y LOW cambiaran la dirección.

El pin PWM te permite analogWrite al pin para controlar la velocidad de aquel motor. Enviando 0 a este pin detiene el motor mientras 255 representa la máxima velocidad.

Paso 3: Cargar un nuevo código al Arduino.

```
// este código controla dos motores
```

```
// motor A debería ser conectado entre A01 y A02
```

```
//motor B debería ser conectado entre B01 y B02
```

```
//El código de abajo define los pines de salida en el Arduino y conecta pines específicos en el Hbridge
```

```
inst STBY=10; // este será el pin standby
```

```
//Motor A
```

```
int PWMA=3; // Control de velocidad
```

```
int AIN1=9; // Dirección
```

```
int AIN2=8;// Dirección
```

```
//Motor B
```

```
int PWMB= 5; //Control de velocidad
```

```
int BIN1= 11; Dirección
```

```
int BIN2=12; //Dirección
```

```
void setup(){
```

```
  pinMode (STBY, OUTPUT);
```

```
pinMode(PWMA, OUTPUT);
```

```
pinMode(AIN1, OUTPUT);
```

```
pinMode(AIN2, OUTPUT);
```



```
pinMode(PWMB, OUTPUT);  
pinMode(BIN1, OUTPUT);  
pinMode(BIN2, OUTPUT);  
}
```

```
void loop(){  
  move(1, 255, 1); //motor 1, velocidad completa, izquierda  
  move(2, 255, 1); //motor 2, velocidad completa, izquierda  
  delay(1000); //ir por 1 segundo  
  stop(); //detener  
  delay(250); //pausar por 250ms hasta el siguiente movimiento  
  move(1, 128, 0); //motor 1, velocidad media, derecha  
  move(2, 128, 0); //motor 2, velocidad media, derecha  
  delay(1000); //ir por otro segundo  
  stop();  
  delay(250);  
  
}
```

```
void move(int motor, int speed, int direction){  
  //mover motor especifico en velocidad y direccion  
  //motor: 0 para B 1 para A  
  //velocidad: 0 is apagado, y 255 es velocidad maxima  
  //direccion: 0 segun las agujas del reloj, 1 izquierda segun las  
  agujas del reloj  
  
  digitalWrite(STBY, HIGH); //desactiva el standby (pausa)  
  boolean inPin1 = LOW;
```

```

boolean inPin2 = HIGH;
if(direction == 1){
    inPin1 = HIGH;
    inPin2 = LOW;
}
if(motor == 1){
digitalWrite(AIN1, inPin1);
digitalWrite(AIN2, inPin2);
analogWrite(PWMA, velocidad);
}else{
digitalWrite(BIN1, inPin1);
digitalWrite(BIN2, inPin2);
analogWrite(PWMB, speed);

}

}

void stop(){
//enable standby
    digitalWrite(STBY, LOW);
}

```

Etapa 7: Aprender a controlar los motores via retroalimentación desde el sensor de presión

Aquí es donde se pone divertido. Ahora necesitas tu programa que leyó la información del sensor de presión y convertirlo en un

programa que ejecute tus motores. Tu tienes que hacer el programa de modo que la hélice se mueva en una dirección si la presión es mayor que el promedio (donde tu defines el promedio) y en otra dirección si la presión es menor que el promedio.

Paso 1:

Medir el sensor de presión y comparar el valor con el valor deseado.

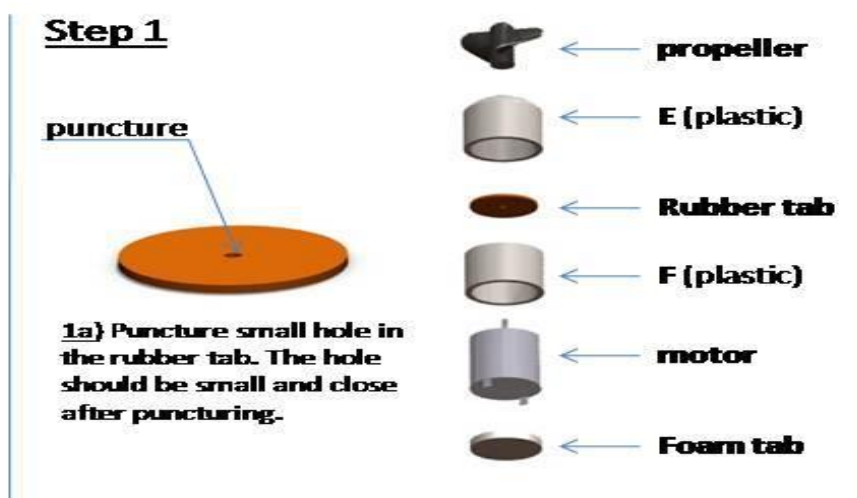
Paso 2:

Si la Presion deseada > Presion actual , ejecuta el motor de una manera

Si la Presion deseada < Presion actual ejecuta el motor de otra manera

Si la presión deseada = Presion actual apaga el motor

Etapa 8 (opcional): Impermeabilizar tus motores.

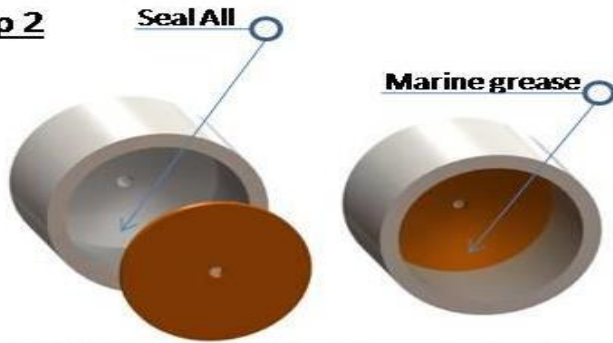


Paso 1

Perforación

1. Perfora un pequeño agujero en la superficie de goma. El agujero

Step 2



2a) Apply Seal All between E and rubber tab generously to make a water tight seal. Join E and rubber tab while getting little to no Seal All on the other side of the rubber tab.

2b) Apply marine grease to the inside of the rubber tab.

Paso 2

Sellado completo

Grasa marina

2ª) Aplica el sellante entre E y en la superficie de goma para crear un sello resistente al

Paso 3 agua. Une E y la Cinta de hilo

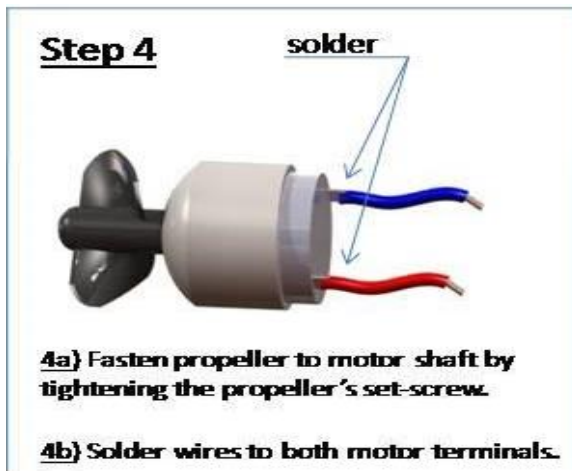
Step 3



3a) Wrap motor with thread tape until the motor is no longer loose when inserted into E.

3b) Insert motor into E.

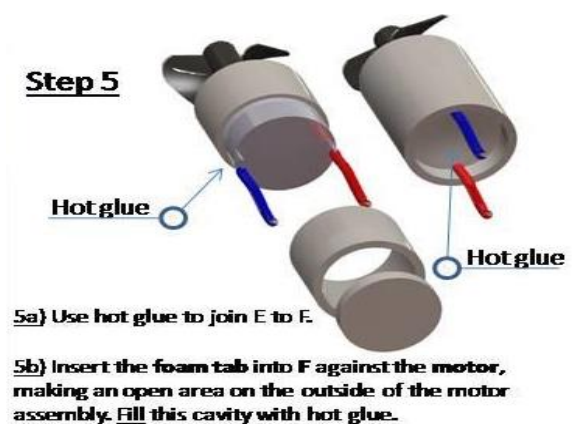
3ª) Envuelve el motor con la cinta de hilo hasta que el motor este bien



Paso 4

Soldar

4ª) Sujeta la hélice al eje del motor apretándola con el tornillo de fijación de



Paso 5

Pegamento Caliente

5ª) Usar pegamento caliente para unir E con F

5b) Inserta la esponja en F en contra del motor, creando un área abierta en

Paso 9: Impermeabilizar tu Arduino -Uno o Arduino-Mini

En una versión anterior de esta clase usamos un micro controlador Arduino-Mini como el cerebro de nuestro ROV. Las ventajas del Mini fueron que es menos costoso, mas pequeño y aun pudo hacer el mismo trabajo. La desventaja del Mini fue que no tiene tantos puertos de información y es muy fácil de destruir accidentalmente.

Si te sientes aventurero y no tienes nada fijo en tu horizonte, te voy a asignar un Mini. Todos tus programas UNO pueden ser transferidos al Mini y funcionara de la misma manera.



El Arduino Mini versus el Arduino-Uno

La inversión de la polaridad de la alimentación de la tensión de voltaje para el Mini es la manera mas común en la que la gente ha quemado sus Minis en el pasado. Es por eso que sugiero fuertemente que antes de hacer cualquier cosa, pongas un diodo en la línea de suministro de energía para ayudar a proteger en caso de posibles errores.

Este trimestre (otoño 2013) también tenemos suministros para que hagas un AUV en lugar de un ROV. Podría ser divertido comprar unos motores extras y acelerómetros de modo que tu ROV pueda subir, bajar, adelantar, retroceder, ir a la izquierda y derecha. Si quieres ser del grupo ROV, podría comprar algunas partes para ti.

Etapa 10: Prueba tu ROV

