



MARCO DE FOTOS
CON ILUMINACIÓN LED



MARCO DE FOTOS CON ILUMINACIÓN LED

- 1. Descripción del proyecto y sus componentes.
 - 1.1. Placa compatible con Arduino.
 - 1.2. Placa Protoboard Mini.
 - 1.3. El potenciómetro.
 - 1.4. Conjunto de tres diodos LED.
- 2. Montaje del proyecto.
 - 2.1. Montaje.
 - 2.2. Esquema del circuito.
- 3. Programación.

MARCO DE FOTOS CON ILUMINACIÓN LED

¿Te gustaría ver tus fotos en un marco con luz led regulable? Con este proyecto vas a hacerlo entendiendo el funcionamiento y el porqué de cada uno de sus componentes. Además, vas a ampliar tus conocimientos de robótica y programación entendiendo cada instrucción del programa y lo más importante ¡vas a divertirte haciéndolo!

Con el proyecto MARCO DE FOTOS CON ILUMINACIÓN LED de Inputmakers vas a aprender sobre:

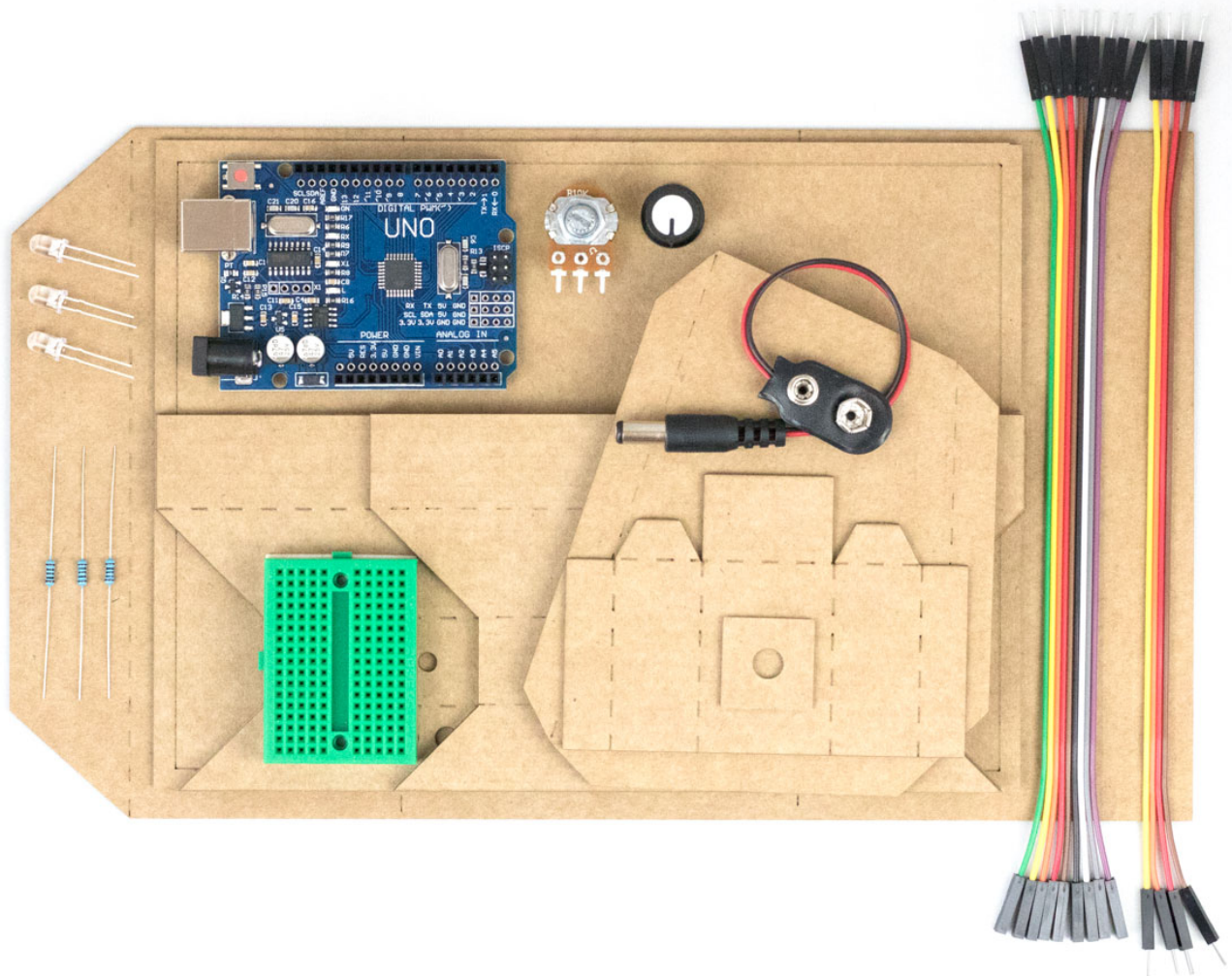
- La placa compatible con Arduino.
- La placa protoboard mini.
- Potenciómetro.
- Diodo led.
- Resistencia de polarización del led.
- Entradas analógicas.
- Salidas PWM.
- Montaje de circuitos.
- Programación.

1. DESCRIPCIÓN DEL PROYECTO Y SUS COMPONENTES:

El marco de fotos led es un proyecto de robótica con el que podrás regular el nivel de luz con el que iluminar tu foto favorita. Girando un simple potenciómetro daremos la orden a la placa compatible con Arduino para que gradúe el nivel de luz de tres diodos led con los que iluminaremos la fotografía.

El marco de fotos led de InputMakers está compuesta por:

- Placa compatible con Arduino con cable USB.
- Placa protoboard mini.
- 4 cables de conexión macho-macho.
- 9 cables de conexión macho-hembra.
- 1 potenciómetro de 10 k Ω .
- 3 diodos led blancos.
- 3 resistencias de 100 Ω
- Pila de 9V.
- Conector para pila.
- Piezas de cartón.



1.1. LA PLACA COMPATIBLE CON ARDUINO:

Se puede entender como un pequeño ordenador que recibe información desde diferentes sensores. Estos sensores pueden ser diferentes y muy variados, por ejemplo: de temperatura y humedad, de sonido, botones, interruptores final de carrera, etc.

Tras ejecutar un programa con una serie de órdenes, enviaremos las señales de salida para controlar diferentes dispositivos, por ejemplo: un led, un motor, un zumbador, mostrar información en una pantalla...

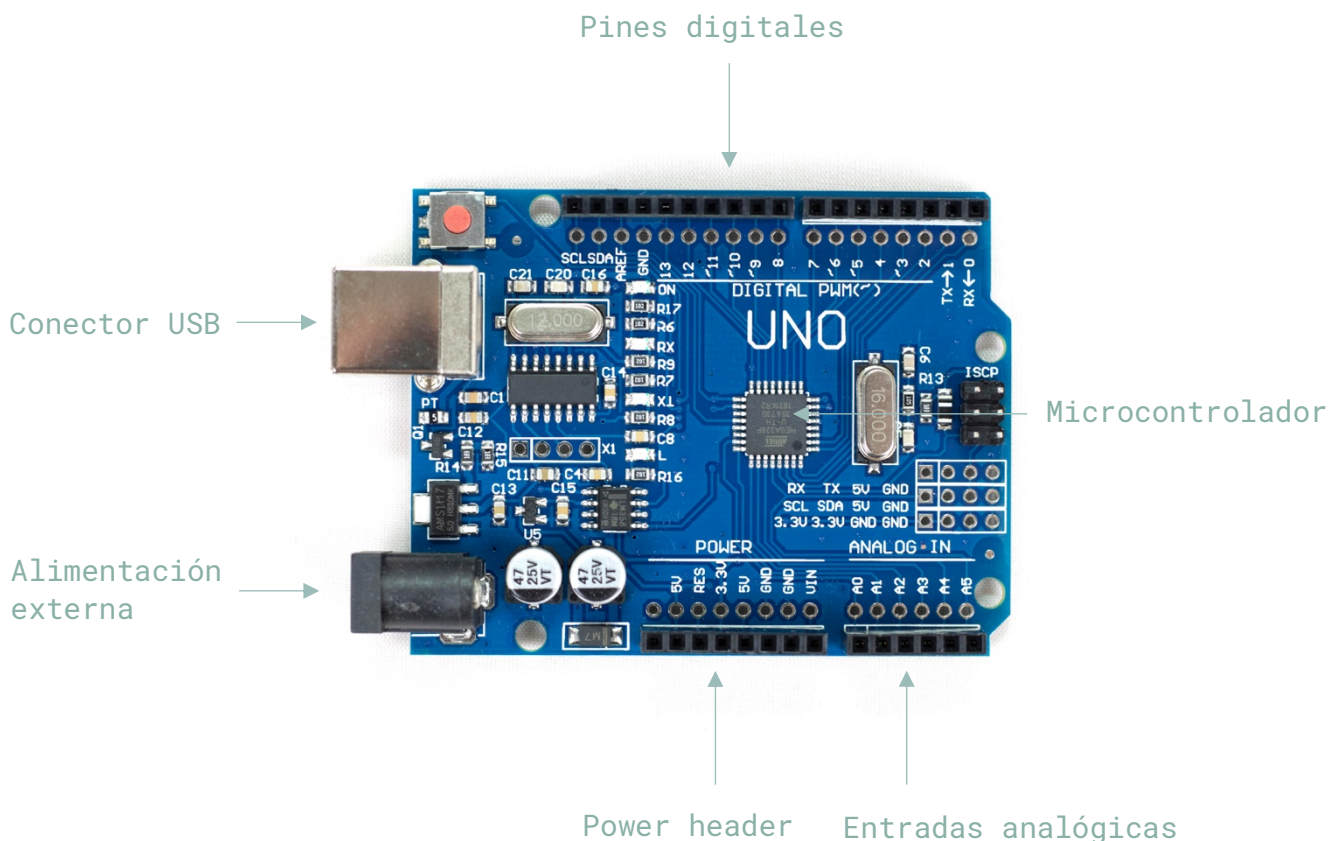
Dicha placa está formada por:

- Un microcontrolador.
- Pines digitales: 14 entradas/salidas digitales que llamaremos pines.

Cada uno de estos pines pueden configurarse como entrada o como salida. Solo los 6 que van acompañados del símbolo ~ pueden utilizarse como salidas PWM (Modulación por ancho de pulso).

Dentro de los pines digitales podemos encontrar en la posición 0 el pin RX SERIAL IN (recepción) y en la posición 1 el pin TX SERIAL OUT (transmisión).

- 6 entradas analógicas: pines destinados a leer señales analógicas.
- Power header.
- Conector de alimentación externa: acepta tensiones entre 7 y 12 V,
- Conector USB (5V).



1.2. PLACA PROTOBOARD MINI:

Es un componente plano de plástico con agujeros conectados por filas. Sirve para montar circuitos electrónicos sin tener que soldar sus componentes, esto favorece un entorno de trabajo donde podemos probar, montar y desmontar componentes y circuitos tantas veces como queramos.

En nuestro caso utilizamos la protoboard mini debido al reducido espacio que tenemos en nuestro proyecto.

Podemos dividir la placa protoboard en dos zonas idénticas separadas por un surco central. En cada zona, los orificios de cada una de las filas están conectados eléctricamente entre sí. Sin embargo, cada fila está aislada eléctricamente de las demás, tal y como se explica abajo.

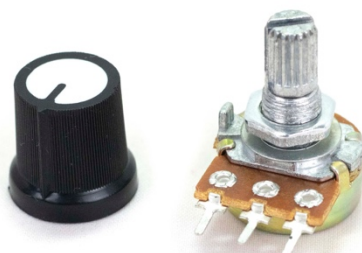


LOS ORIFICIOS DE LA COLUMNA VERDE ESTÁN CONECTADOS ELÉCTRICAMENTE ENTRE SÍ, PERO AISLADOS DE CUALQUIER OTRO, POR EJEMPLO, LA AZUL.

Utilizaremos la placa protoboard mini para montar el conjunto formado por la resistencia y el zumbador.

1.3. EL POTENCIÓMETRO:

Un potenciómetro es una resistencia variable. Esto es un tipo de resistencia en la que de manera manual podemos variar su valor de resistencia entre un valor mínimo de $0\ \Omega$ y un valor de resistencia máximo $R_{\text{máx}}$. En nuestro caso $R_{\text{máx}} = 10\ \text{k}\Omega$.



¿Cómo funciona un potenciómetro? En él tenemos tres terminales de conexión y un mando rotativo. Los dos terminales externos nos ofrecen la resistencia máxima (en nuestro caso 10 kΩ) o mínima (0 kΩ). El terminal central nos da el valor de resistencia proporcional al giro del mando rotativo.

Como las resistencias no tienen polaridad, podemos conectar los terminales externos en un sentido o en otro, ello hará que el giro o bien aumente o bien disminuya el valor de resistencia.

Entonces ¿para qué sirve y cómo vamos a utilizar el potenciómetro para nuestro proyecto? Es el dispositivo que va a regular el valor de la señal de entrada.

Respecto a la entrada de señal en la placa, en este proyecto vamos a trabajar con señal de entrada analógica. Esta señal corresponderá al valor de tensión que vendrá del terminal central del potenciómetro.

Para ello, conectaremos uno de los terminales exteriores a 5 V y el otro a tierra (0 V) de esta manera girando la rueda del potenciómetro al variar la resistencia obtendremos por la ley de Ohm una variación del voltaje en el terminal central que estará conectado a una de las entradas analógicas de la placa A0, A1, A2...

Así, hemos conseguido nuestro objetivo de introducir manualmente con un simple giro valores analógicos en nuestra placa.

Las entradas analógicas convierten el valor de tensión recibida entre 0 V y 5 V en un número entre 0 y 1023. ¿Por qué un número entre 0 y 1023? Porque las entradas analógicas tienen una resolución de 10 bits, esto implica $2^{10} = 1024$ valores. Como empezamos desde el cero llegamos a 1023.

CONEXIONES	
Terminal exterior	Conectando a 5V
Terminal central	Conectado a una entrada analógica (A0, A1, A2, ...)
Terminal exterior	Conectando a GND (0V)

1.4. CONJUNTO DE TRES DIODOS LED CON SUS RESISTENCIAS:

Lo que llamamos coloquialmente como LED es realmente un tipo de diodo que emite luz (diodo LED). Un LED emite luz con el paso de corriente eléctrica y para que esto ocurra debe haber aplicada una diferencia de potencial o voltaje entre sus terminales. Un LED a diferencia de una resistencia, tiene polaridad. Esto significa que solo podrá permitir el paso de intensidad en un único sentido, de positivo a negativo.

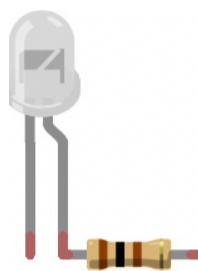
Como vemos en la fotografía, existe un terminal más largo que otro. El terminal positivo corresponde a la pata más larga, mientras que el negativo corresponde a la pata más corta.



A grandes rasgos el funcionamiento de un led se basa en la unión de dos semiconductores con diferente dopaje, la cual cosa crea una barrera de potencial (V_d) que hay que superar para que el diodo se encienda.

Si no llegamos a superar esa V_d entonces el diodo led estará apagado en lo que se llama la zona de corte. Si superamos la tensión V_d entonces el diodo led estará en la zona de conducción y empezará a emitir luz.

El problema es que cuando el diodo entra en conducción porque se supera V_d prácticamente no existe oposición al paso de corriente ya que su resistencia interna es muy baja. Esto provoca que aumente mucho la corriente que lo atraviesa y por tanto que exista peligro de quemar el componente, es por ello que hemos de colocar una resistencia en serie con el diodo led para limitar el paso de corriente y así protegerlo.



¿Cómo calculamos la resistencia de polarización del led?

Simplemente, aplicando la Ley de Ohm sobre la resistencia de polarización:

$$R = \frac{V_R}{I}$$

Teniendo en cuenta que el voltaje aplicado sobre la resistencia será igual al voltaje de alimentación de arduino (5V), menos la barrera de potencial del led (V_d), tenemos que:

$$V_R = V - V_d$$

Sabiendo que las salidas digitales de la placa compatible con Arduino están pensadas para funcionar a un valor de intensidad alrededor de 20 mA (con un máximo de 40 mA), el valor de la resistencia vendrá dado por:

$$R = \frac{V - V_d}{I}$$

La tensión del diodo V_d tiene un valor diferente según el color del diodo LED. Para nuestro caso, luz blanca $V_d = 3,6V$. Así la resistencia de polarización del diodo LED en nuestro caso dará:

$$R = \frac{5 - 3,6}{0,020} = 70 \, \Omega$$

Debemos coger la resistencia normalizada inmediatamente superior.

$$R = 100 \, \Omega$$

Podemos aumentar un poco el valor de la resistencia y aunque perdamos un poco de brillo en el LED ganaremos en tiempo de vida útil. En la práctica tampoco vamos a notar grandes diferencias en el brillo de los LEDs.

Por último, vamos a calcular la potencia que disipará la resistencia y nos aseguraremos que la resistencia que montemos sea de una potencia superior para evitar que se queme:

$$P = V_R \cdot I$$

$$P = (V - V_d) \cdot I$$

$$P = (5 - 3,6) \cdot 0,02 = 0,028 \, W$$

Cumplimos la condición de potencia holgadamente pues utilizamos resistencias de 1/4W (muy comunes).

Respecto a la salida.

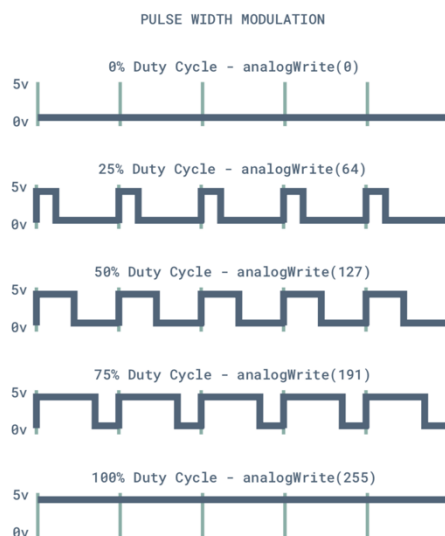
Nuestro objetivo en este proyecto será leer los valores que nos da el potenciómetro y

según estos, variar la intensidad de luz de los LEDs. Así para controlar la intensidad de luz de los LEDs debemos conectarlos a las salidas PWM.

Pero ¿qué son los pines de salida PWM?

Son un tipo de pines que al definirlos como salidas, pueden ser utilizados como pines PWM. Las siglas PWM significan Modulación por Ancho de Pulso (en inglés). Gracias a esta transferimos la información mediante una señal cuadrada, de 5 V como valor máximo y 0 V como valor mínimo.

Como vemos en la siguiente figura el “ciclo de trabajo” es la relación entre el tiempo que la señal está a 5 V y el tiempo que está a 0 V, expresado en tanto por ciento %.



Así con los pines PWM podemos variar el ciclo de trabajo (Duty cycle) utilizando la función: `analogWrite (pin, valor)`

Donde:

- Pin es el pin PWM donde está conectada la señal.
- Valor es un valor de 0 a 255, donde 0 corresponde a 0% del ciclo de trabajo y 255 al 100% del ciclo de trabajo.

Los pines PWM son salidas de 8 bits, esto significa que tendremos 2^8 valores, es decir, podemos discretizar el ciclo de trabajo en 256 niveles, esto lo haremos con un número de 0 a 255. Ya que se empieza a contar desde el 0.

En definitiva, lo que hacemos es simular una señal analógica variando el tiempo del ciclo en que mantenemos activada la señal, de tal manera que la tensión que obtenemos en promedio será equivalente al valor de tensión que queremos obtener. Dicho de otra manera, en vez de dar diferentes valores de forma constante, gracias a la variación de la señal de salida en los ciclos de trabajo se obtienen valores medios equivalentes.

Por ejemplo, si escribimos: `analogWrite (pin, 127)` obtendremos ciclos de trabajo del 50% y por tanto será equivalente a dar una señal de salida de 2,5 V constante.

Realmente si conectáramos a esta salida un diodo (con su resistencia de polarización), lo que estaría sucediendo es que el diodo se estaría encendiendo y apagando el 50% del tiempo, pero esto el ojo humano no es capaz de verlo, sino que en definitiva la percepción es que el diodo brilla a menor intensidad.

Por tanto, hay que tener en cuenta que la señal PWM NO ES UNA SEÑAL ANALÓGICA y que por tanto según lo que alimentemos puede generar algún tipo de problema por ejemplo en cargas inductivas (motores, electroimanes...) donde se generen corrientes inducidas, etc.

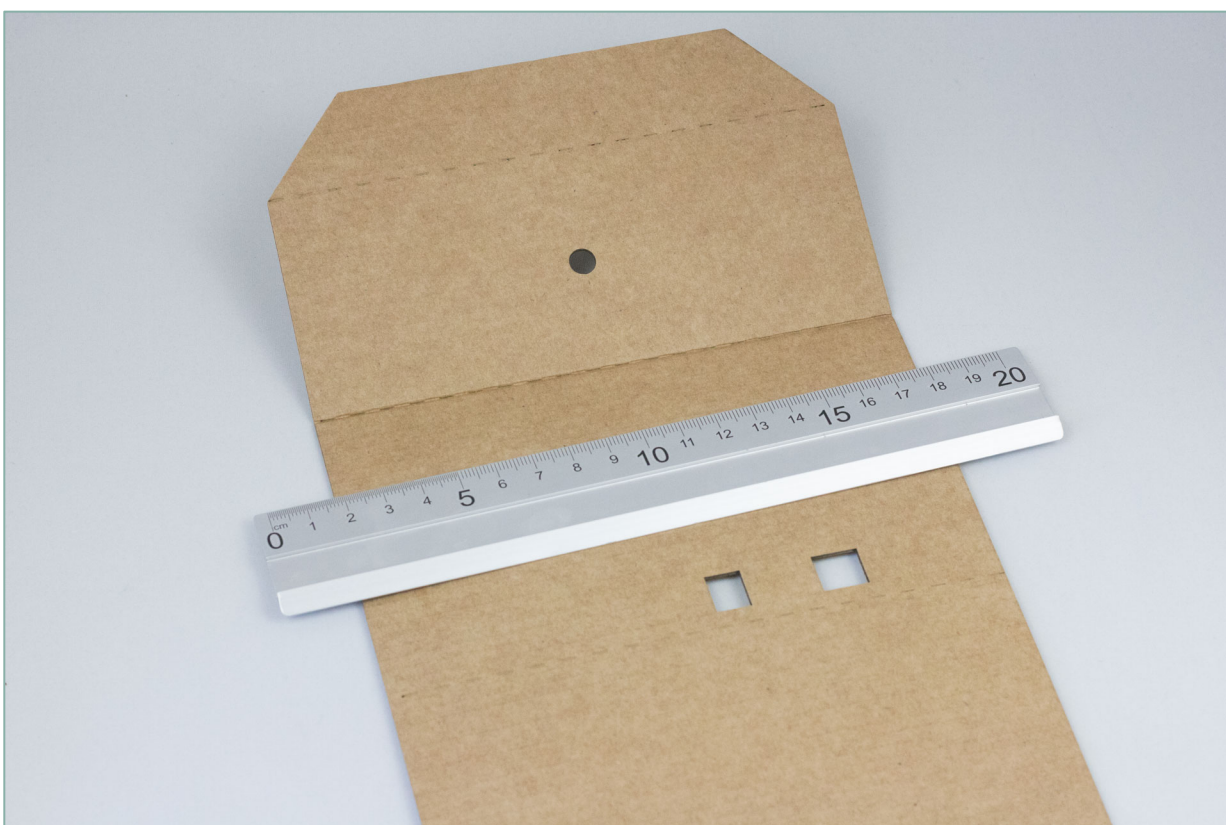
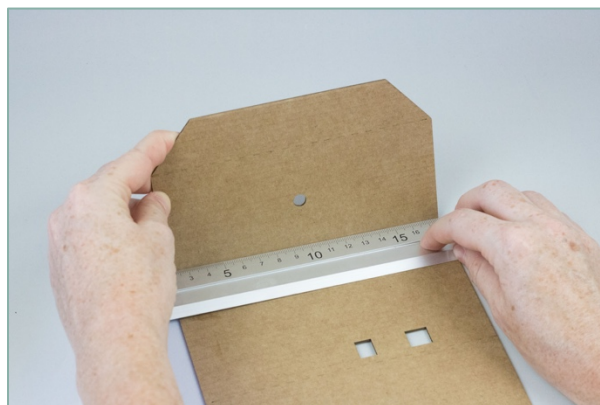
También hay que tener en cuenta que si estamos alimentando un dispositivo que solo admite por ejemplo 3,3 V mediante señal PWM que va de 5 V a 0 V esto puede ocasionarle daños ya que le estamos suministrando cierto porcentaje de tiempo 5 V.

2. MONTAJE DEL PROYECTO:

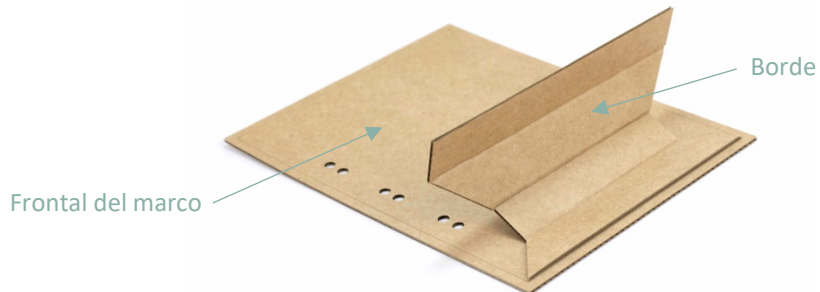
2.1. MONTAJE:

*Se recomienda usar silicona caliente para pegar las diferentes partes del proyecto.

- a) Las piezas del proyecto se han de doblar por las líneas discontinuas. Presionamos con el canto de una regla sobre dicha línea y doblamos la pieza.



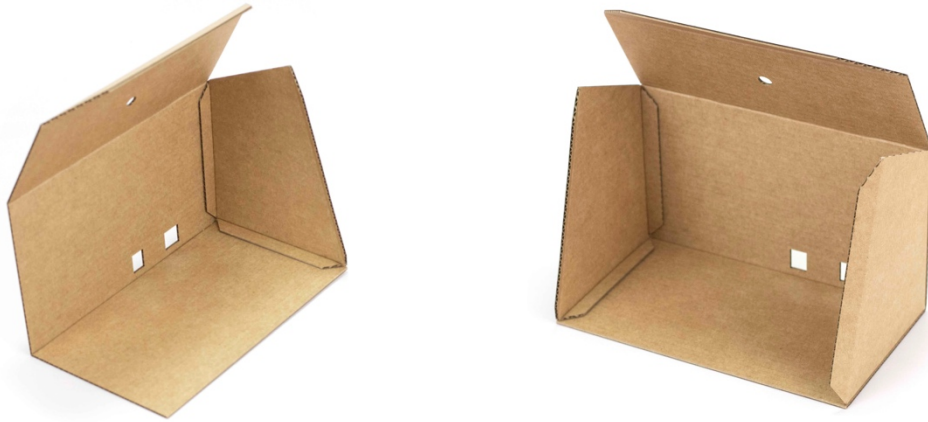
- b) Montamos los cuatro bordes sobre el frontal del marco (pieza A). Para ello alineamos el canto de las piezas con las marcas del frontal del marco y pegamos con silicona caliente.



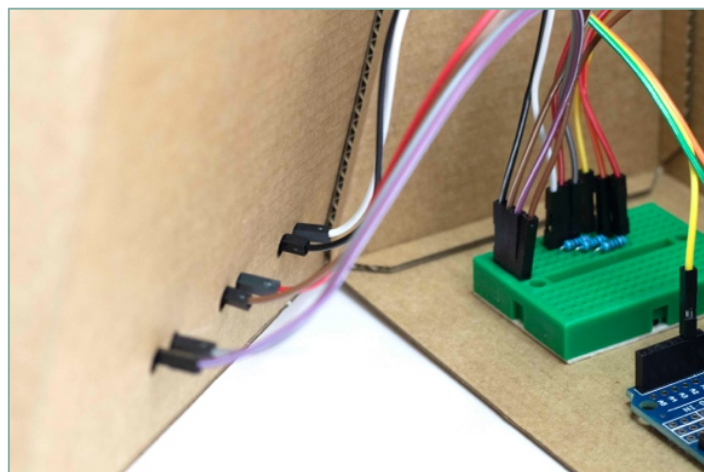
- c) Cerramos los bordes de las piezas laterales y superior al frontal del marco pegándolas con silicona. Dejamos el borde inferior sin cerrar, donde se sitúan los tres LEDs, para montarlos más tarde.



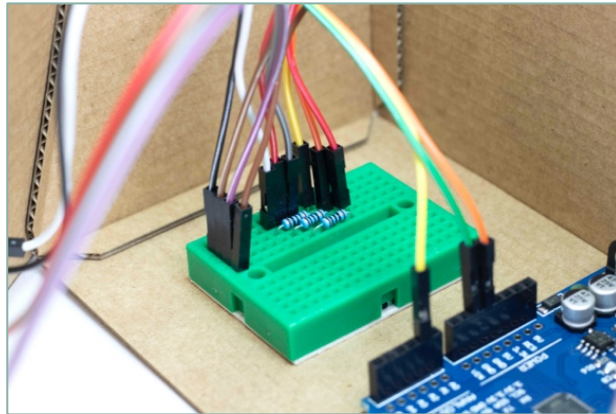
- d) Pegamos las piezas laterales sobre la pieza posterior utilizando las solapas. De esta manera permitimos que la tapa se abra y se cierre cómodamente.



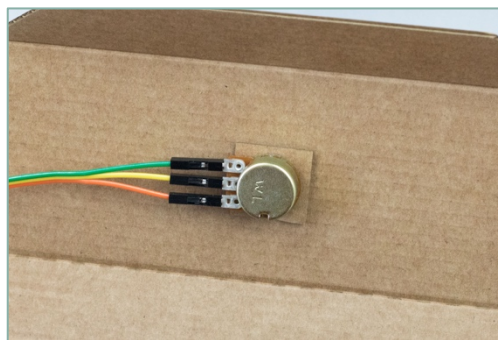
- e) Realizamos las conexiones de los componentes como se indica en el esquema del circuito en la página 16 y colocamos cada componente en su lugar.
1. Coloca los LEDs tal como se indica en la fotografía antes de pegar el canto del borde inferior al frontal del marco.



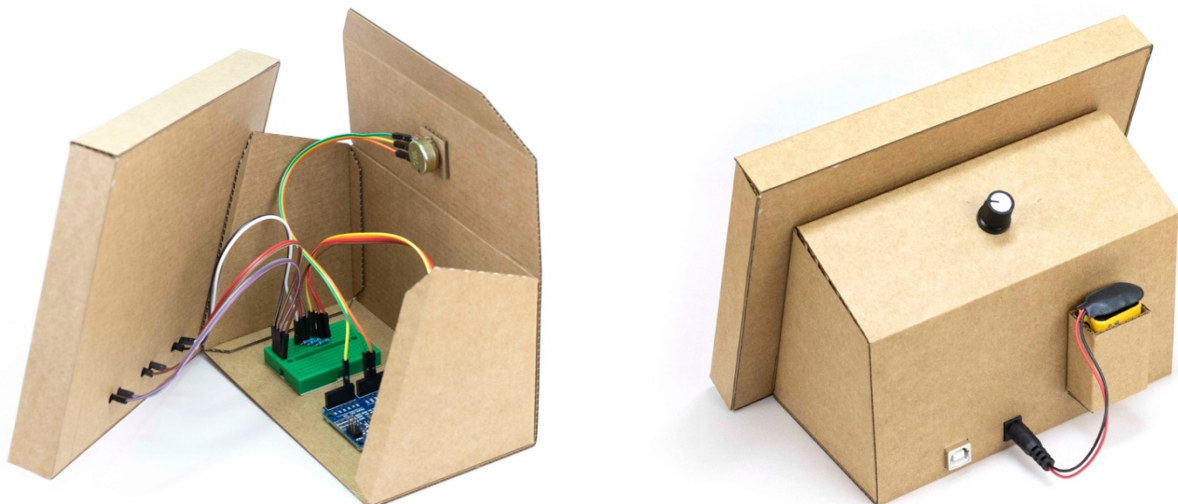
2. Despegamos el plástico que hay en la parte inferior para pegar la placa protoboard mini a la base.



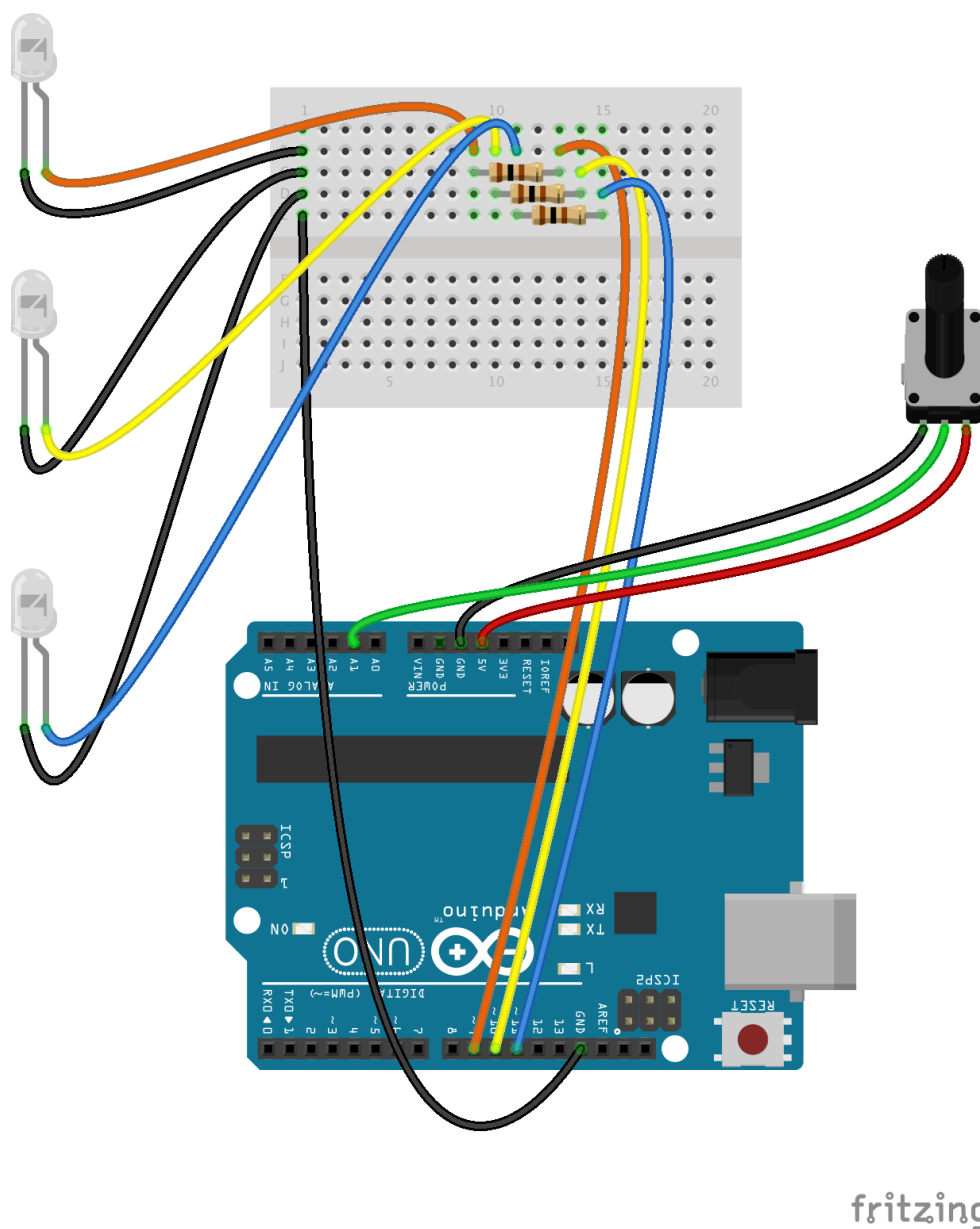
3. Coloca la arandela de cartón al montar el potenciómetro en la parte interior de la tapa del marco de fotos.



4. Pegamos el conjunto posterior, donde va alojada la electrónica del proyecto, al frontal del marco de fotos por su cara posterior. Para ello utilizamos las solapas de las piezas laterales. Para centrar ambas partes utilizamos las dos marcas situadas en la cara posterior del frontal del marco.



2.2. ESQUEMA DEL CIRCUITO:



*Todos nuestros esquemas estan diseñados con el software libre y de código abierto fritzing (Fritzing is open source, free software). Dicho software utiliza en sus esquemas recreaciones de placas originales Arduino Uno, en nuestro kit proporcionamos una placa compatible con Arduino.

Programación:

1. Descarga el programa Arduino IDE (software libre y gratuito) de la página oficial de Arduino e instálalo en tu ordenador.
2. Conecta la placa al ordenador con el cable USB.
3. Comprueba que tenemos seleccionado el modelo de la placa Arduino/Genuino Uno.

Vamos a: Herramientas> Placa: "Arduino/Genuino Uno"

La placa se conecta por defecto, pero es importante comprobar que está seleccionada correctamente. Si no es así, no nos funcionará el programa.

4. Comprueba que la placa está conectada al puerto USB.
Herramientas> Puerto: usbserial (se conecta por defecto).
5. Escribe el código de programación (página 19) en el IDE de Arduino.

Aconsejamos copiarlo a mano para así interiorizar mejor lo que estamos programando. Si se te olvida escribir alguna coma, paréntesis, etc te dará un error de sintaxis. Revisa dicho error en la línea que el programa te marque.

Si decides copiar y pegar el código, en función del lector de pdf puede que no se copie correctamente dejándose alguna línea de programación, si es así siempre puedes copiar y pegar el código por partes o completar escribiendo la parte que no se pegue correctamente.

Aunque decidas copiar y pegar el programa, intenta entender cada línea de código, para ello tienes su explicación en el programa.

Todo lo que esté a la derecha de // o entre /* y */ no es ejecutado por el programa.

6. Presionamos el icono VERIFICAR  para comprobar que el programa no contiene ningún error en el código. Si la verificación es correcta y no tenemos

ningún error, pulsamos en el icono SUBIR.



Ahora ya tenemos el programa subido a la placa, comprueba cómo al girar el potenciómetro regulamos la intensidad de luz de los leds.

7. Una vez subido el programa a la placa, podemos desconectarla del cable USB y alimentarla con la pila de 9V.

Aclaraciones sobre la programación en el IDE de Arduino:

Si no estás familiarizado con la programación, no te preocupes, ahora vamos darte una serie de ideas básicas a partir de las cuales puedas empezar a entender de qué va todo esto.

Existen diferentes lenguajes de programación, unos más parecidos a otros, pero digamos que el lenguaje que se utiliza en Arduino es muy parecido a uno de los tipos de lenguajes más utilizados en programación, el lenguaje C++

A quien sabe programar en C++ le cuesta minutos adaptarse al IDE de Arduino.

Lo que te vas a encontrar en el código de un programa en IDE de Arduino son en términos generales, 3 partes:

1. La primera parte en la que básicamente se van a incluir librerías necesarias para ejecutar el programa y se van a definir variables.
2. La segunda parte, el void setup, aquí definimos si los pines de Arduino van a corresponder a variables de entrada o salida (con el comando pinMode) y inicializamos elementos del sistema como sensores, pantallas...
3. La tercera parte, el void loop, implementamos las órdenes necesarias para que el programa realice lo que queramos.

Dependiendo del programa y del programador puede haber variaciones.

Aquí tienes el programa:

```
/* InputMakers
Proyecto Marco de Fotos con Iluminación Led: Mediante el
potenciómetro regulamos la intensidad de luz de los tres
leds que iluminan la fotografía.*/

int led9 = 9, led10 = 10, led11 = 11;    // Variables asociadas a cada led.
int brillo;    // Variables donde guardamos la intensidad de brillo.
int pot = 0;    // Variable donde guardamos la lectura del potenciómetro.
int potpin = A1; // Pin del potenciómetro conectado a la entrada analógica A1.

void setup() {
  pinMode (led9, OUTPUT);    // Definimos los pines que van conectados a los
  leds como salidas.
  pinMode (led10, OUTPUT);
  pinMode (led11, OUTPUT);
}

void loop() {

  pot = analogRead(potpin);    // Lectura del valor del potenciómetro con
  analogRead.

  /* A continuación escalamos los valores del potenciómetro (que van de 0 a 1023
  para la entrada analógica) al led (que van de 0 a 255 para la salida PWM).*/
  brillo = map (pot, 0, 1023, 0, 255); // Escalar valores del potenciómetro
  (entrada analógica) al led (salida digital).

  /* Por último asignamos el valor de la variable de brillo a cada led con
  analogWrite.*/
  analogWrite (led9, brillo);
  analogWrite (led10, brillo);
  analogWrite (led11, brillo);
}
```

¡Enhorabuena! ¡ya has realizado desde cero tu Marco de fotos con iluminación led!

Desde el equipo de InputMakers queremos felicitarte por haberte dado la oportunidad de aprender algo nuevo. Esperamos que te hayas divertido haciendo tu MARCO DE FOTOS CON ILUMINACIÓN LED tanto como nosotros al prepararla.

Aquí van una serie de puntos sobre los conceptos más importantes que has aprendido en este proyecto:

- La placa compatible con Arduino.
- La placa protoboard mini.
- El potenciómetro.
- El diodo led.
- La resistencia de polarización del led.
- Entradas analógicas.
- Salidas PWM.
- Montaje de circuitos.
- Programación en el IDE de Arduino.

Saludos del equipo de InputMakers y ¡sigue siempre aprendiendo!

*Los esquemas de circuitos y figuras realizados con el software libre y de código abierto fritzing (Fritzing is open source, free software) tienen licencia Creative Commons Atribución Compartir Igual CC By-SA. El resto del manual tiene todos los derechos reservados.