

Universidad de Puerto Rico

Departamento de Ingeniería INEL/ICOM

Mayagüez, Puerto Rico

High Striker: Guía de Estudiantes



Edna E. Rivera

Josue Albarran

Pedro Rivera

Este manual es dedicado a todas esas personas quienes nos han ayudado en el proceso de esta investigación y que sin duda alguna fueron crucial para el desarrollo de la misma.

Tabla de Contenido

Introd	ucción	4
Objeti	ivos	5
Conce	eptos Básicos	6
A.	Componentes:	6
B.	Funcionamiento del tablero de circuitos:	8
Mater	iales	10
A.	Cuerpo	10
B.	Circuito Eléctrico	10
Esque	mático y Diagrama de Bloque	11
A.	Diagrama de Bloque	11
Metod	lología	12
A.	Configuración de la Caja:	12
B.	Configuración Circuito Eléctrico:	15
C.	Configuración del Código:	21

Introducción

A través de los años la necesidad de avances tecnológicos ha aumentado exponencialmente; por lo cual, se urge una mayor cantidad de profesionales enfocados en las áreas de ciencia, tecnología, ingeniería y matemáticas (STEM, por sus siglas en inglés). El grupo de investigación "Embedded Research Group" de la Universidad de Puerto Rico Recinto de Mayagüez se propone diseñar e implementar un programa de alcance que pueda incrementar la cantidad de estudiantes que escogen ser profesionales en las áreas ya mencionadas. Se plantea como solución recurrir a juegos de carnaval comunes donde los estudiantes puedan aprender los conceptos básicos de programación, ensamblaje de circuitos y sistemas integrados mientras juegan. De esta manera, podemos captar su atención con mayor facilidad de una manera interactiva. Este manual está diseñado para que los estudiantes puedan recrear el juego de carnaval, "high striker", siguiendo las instrucciones provistas.

El *high striker* fue inventado en el los 1930 como parte de los juegos de carnaval. El objetivo es darle con un martillo a una pesa para que la misma suba hasta chocar con una campana. Dependiendo cuanto la pesa suba es el premio que el jugador va a recibir y si toca la campana entonces recibe el mayor premio. La versión que este módulo va a describir simula el original pero no usa ni una pesa ni un martillo. El mismo utiliza un sensor de fuerza que el jugador va a golpear con su dedo índice. Dependiendo cuan fuerte el jugador golpee el sensor es la cantidad de luces LEDs que prenderán. La luces simulan el trayecto de la pesa utilizando un total de once luces y la última luz representa lo que sería la campana. Una particularidad de este juego es que usualmente está configurado para que alcanzar la campana sea de gran dificultad. Esto lo hace mas entretenido para los clientes dado a que usualmente pierden y se retan ellos mismos a ganar mientras el encargado del juego sigue ganando dinero.

Objetivos

- Aumentar el conocimiento e interés en ingeniería eléctrica y computadoras (ECE, por sus siglas en inglés) en estudiantes de escuela superior mediante la construcción de juegos interactivos.
- Demostrar principios y conceptos básicos de sistemas integrados, circuitos, programación y electrónica.
- Facilitar la enseñanza de sistemas integrados a través de manuales educativos para los los maestros y estudiantes.
- Lograr que los estudiantes puedan recrear por sí mismos el juego de carnaval.
- Por medio de un plan de evaluación, identificar y medir el conocimiento e interés adquirido mediante la actividad.

Conceptos Básicos

A. Componentes:

Componente	Descripción	Figura		
Tablero de circuitos	Tablero que se utiliza para realizar prototipos electrónicos. Nos permite conectar varios componentes eléctricos en un solo lugar.			
Sensor de Fuerza	Estos son sensores que miden la fuerza que se le aplica en la región cuadrada. Con estos valores se pueden determinar distintas posibilidades y acciones que podrá tomar el sistema.			
Arduino UNO R3	El Arduino UNO R3 es un microcontrolador, o en otras palabras un circuito integrado programable y capáz de ejecutar órdenes grabadas en su memoria. En palabras simples, es el cerebro del juego que se encarga de tomar todas las decisiones referentes a la puntuación y tiempo.	COC UND ABACTOR OF THE WAY IN		
Luces LED's	Componente de luz pasivo o un diodo que emite luz.			

Cable USB tipo A/B	Cable utilizada para conectar el Arduino UNO R3 a la computadora para poderlo programar. Este usualmente está incluido con el Arduino UNO R3.	
Cables	Estos cables nos permiten unir los componentes electrónicos como el botón, microcontrolador, pantallas de 7 segmentos y resistor al tablero de circuitos.	
Potenciómetro	Estos cables nos permiten unir los componentes electrónicos como el botón, microcontrolador, pantallas de 7 segmentos y resistor al tablero de circuitos.	

B. Funcionamiento del tablero de circuitos:

El tablero de circuitos, como se muestra en la ilustración 1, es utilizado para conectar múltiples componentes eléctricos en un solo lugar. Este funciona como la imagen lo ilustra, las columnas A, B, C, D y interconectadas Ε están horizontalmente. Sin embargo, los carriles de alimentación de ambos lados están interconectados verticalmente. Esto significa que el carril entero se comporta como un solo nodo y cualquier componente que se conecte en ese carril estará conectado al resto de los componentes en el carril. De la manera ocurre columnas A, B, C, D y E pero esta vez horizontalmente.

En la siguiente página se muestran ejemplos de cómo se deben conectar los componentes en el tablero y como no se deben conectar.

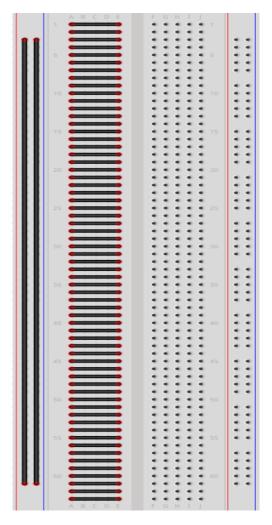


Ilustración 1: Tablero de circuitos

Ejemplos de cómo se deben conectar los componentes en el tablero de circuitos:

Como SI se deben conectar	Como NO se deben conectar		
A B C D E F G H I J C C C C C C C C C C C C C C C C C C	A B C D E F G H I J		
A B C D E 1	A B C D E		
A B C D E F G H I J 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	A B C D E F G H I J		

Materiales

A. Cuerpo

Para el cuerpo todos los materiales necesarios se sugiere que sean reciclables y fácil de conseguir. Si es este el caso, lo único que se necesita es cartón, cartulina,, pega caliente y cinta adhesiva. En el ejemplo que se demuestra en las fotos y se describe en la metodología se utilizan materiales de mejor calidad, pero para una versión más económica solo hay que sustituir los materiales mencionados.

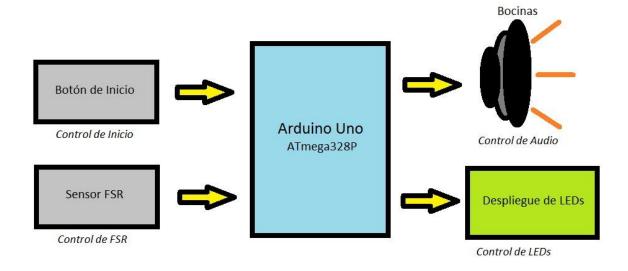
B. Circuito Eléctrico

High Striker BOM									
# Artículo	Descripción	Cantidad	Proveedor	Costo por Unidad	Costo Total				
1 Paquete de LEDs	Paquete que incluye 20 LEDs de diversos colores.	1	Sparkfun	<u>\$2.95</u>	\$2.95				
2 Sensores FSR	Sensores de Fuerza FSR.	1	Adafruit	<u>\$7.95</u>	\$7.95				
3 Arduino UNO R3	Placa de Microcontrolador basado en el ATmega328P.	1	Sparkfun	<u>\$24.95</u>	\$24.95				
4 Cable USB Tipo A/B	Cable USB 2.0 para conectar el Arduino a la computadora.	1	Sparkfun	<u>\$3.95</u>	\$3.95				
5 Resistor de 10k Ohmio	10kOhmio, 1/6 Watt, +/-5% tolerancia.	1	Sparkfun	<u>\$0.25</u>	\$0.25				
6 "Jumper Cables"	20 cables "Male to Male".	1	Sparkfun	<u>\$1.95</u>	\$1.95				
7 "Breadboard"	Placa de Prototipos "Breadboard" grande.	1	Sparkfun	<u>\$5.95</u>	\$5.95				
				Costo Final	\$47.95				

Esquemático y Diagrama de Bloque

A. Diagrama de Bloque

En el caso del diagrama de bloque, este consiste de representar algún sistema por medio de bloques y conectado con líneas. Esta simple herramienta nos permite visualizar el sistema e identificar el límite o "boundary" del sistema al mostrarnos las entradas y salidas del mismo. En la siguiente ilustración se muestra el diagrama de bloque que se utilizó para construir el juego "high striker".



Metodología

A. Configuración de la Caja:

A continuación se muestra el ensamblaje paso a paso, las dimensiones del mismo dependen de cuán grande el usuario quiera su juego o el espacio disponible. Los instructores proveerán las dimensiones específicas para los estudiantes antes del taller.

 Una vez haya escogido la caja, cuidadosamente debe recortar uno de sus lados tomando en cuenta que ese pedazo puede ser utilizado en los próximos pasos. El lado hueco será la parte posterior del proyecto.

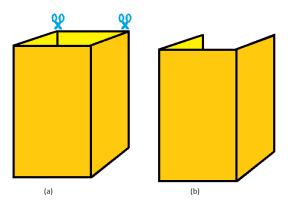


Figura 1. (a) Vista de la caja. (b) Vista de la caja sin la tapa posterior.

 Con precaución utilice unas tijeras o un cortador de precisión para hacer los agujeros de las LEDs. Los once agujeros deben quedar a la misma distancia y del mismo radio.

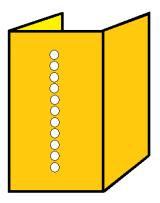


Figura 2. Vista de la caja con los once agujeros para las LEDs.

3) Tome un pedazo de cartón que tenga el mismo ancho de la base y péguelo a la base del proyecto. Considere que esta será la base donde se colocará el sensor de fuerza.

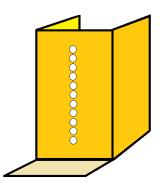


Figura 3. Caja con base para colocar el sensor de Fuerza.

4) **Opcional**: Tome un pedazo de cartón del mismo largo y ancho que la primera pieza y colóquelo como una "tapa" para cubrir todo los circuitos y elementos. De esta manera el jugador no puede ver el circuito y se protege.

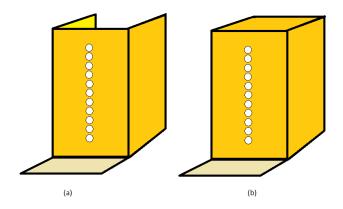


Figura 4. (a) Caja sin tapa. (b) Caja con tapa.

B. Configuración Circuito Eléctrico:

- 1. Conectar los "LEDs" al "breadboard".
 - Primero, debemos considerar las partes de un LED.
 - Un LED tiene 2 patitas, siendo la más larga el ánodo (carga positiva) y la otra el cátodo (carga negativa) según se muestra en la siguiente imagen.

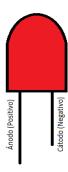


Figura 3: Terminales de una LED

Teniendo esto en cuenta, procedemos a conectar las "LEDs" al "breadboard".
 La siguiente figura muestra la conexión para una "LED".

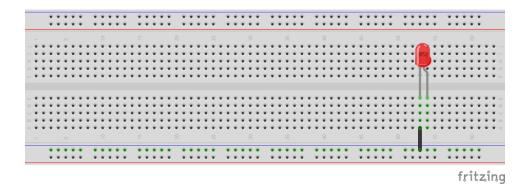


Figura 4: Conexión de una LED con el ánodo libre y el cátodo conectado a la barra designada para la carga negativa (GND).

 Repita el mismo proceso, esta vez utilizando el resto de las "LEDs", tal que hallan 11 en total, como se muestra a continuación.

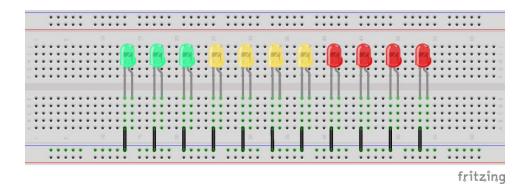


Figura 5: Conexión de las 11 "LEDs" con el ánodo libre y el cátodo conectado a la barra designada para GND.

- 2. Conectar las "LEDs" al Arduino.
 - Debido que se busca controlar 11 "LEDs", utilizaremos 11 pines digitales del Arduino.
 - En este caso tomaremos en consideración los pines del 2 hasta el 12.
 - Comenzaremos asignando las "LEDs" comenzando desde la primera LED roja hasta la ultima LED verde tal como lo muestra la siguiente imágen.

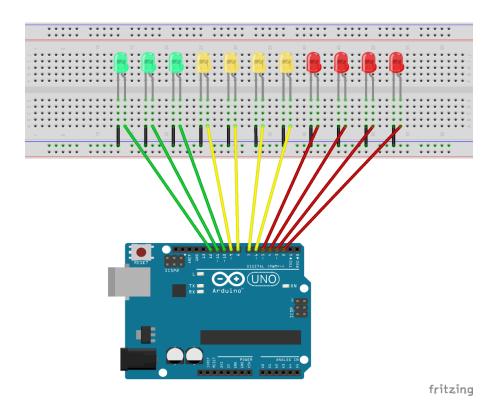


Figura 6: Conexión de las 11 "LEDs" al Arduino.

 Una vez conectadas las "LEDs" al Arduino, necesitamos asegurarnos que todas esten conectadas al GND del Arduino, por lo que se procede a colocar un cable del GND del Arduino hasta un punto de la barra designada para el GND de nuestro "breadboard" como se muestra en la siguiente imagen.

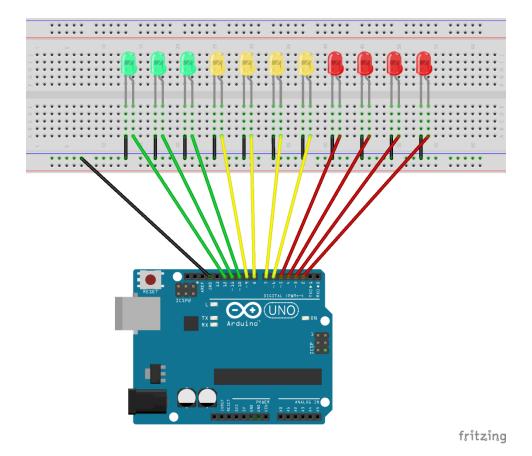


Figura 7: 11 "LEDs" conectadas al Arduino y conexión de GND del Arduino al "breadboard" o tablero de circuitos

2. Conectar el sensor FSR.

- El sensor FSR es el que nos ayuda a determinar la intensidad del impacto y así saber cuántas "LEDs" se encenderán.
- Antes de conectar directamente el sensor, se utiliza un resistor de 10k Ohmios.



Figura 8: Ilustración de un resistor de 10k Ohmios.

 El resistor debe conectarse desde la barra designada para GND en el "breadboard" hasta un nodo que no se esté utilizando en el "breadboard".
 Refiérase a la siguiente imágen.

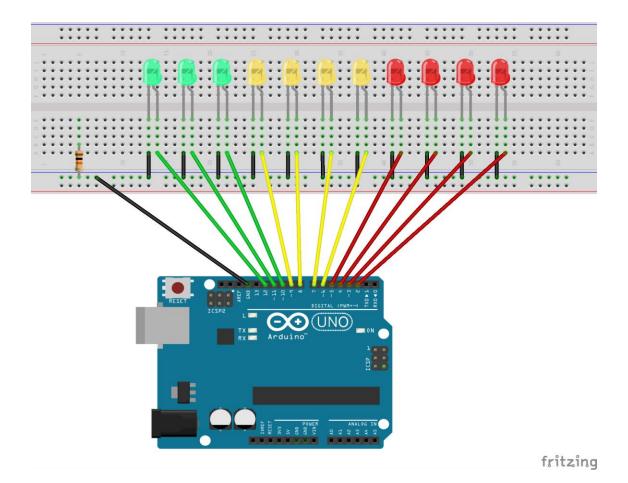


Figura 9: Vista del "breadboard" con las LEDs y el resistor de 10k Ohmios en la parte izquierda conectado a GND.

• Por último, haremos la conexión del sensor FSR.

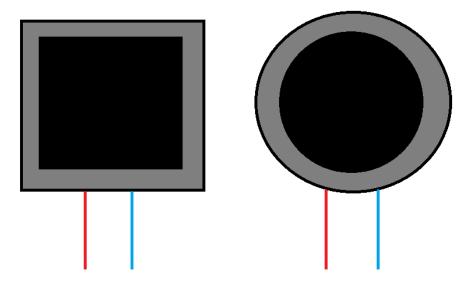


Figura 10: Sensores FSR pueden encontrarse de forma cuadrada o circular, pero su funcionamiento es el mismo. Véase que tienen dos cables los cuales se conectarán en paralelo a los pines 1 y 3 del potenciómetro.

• Uno de los extremos del sensor FSR irá conectado en paralelo a la resistencia y al pin análogo 0 (A0) según se muestra a continuación.

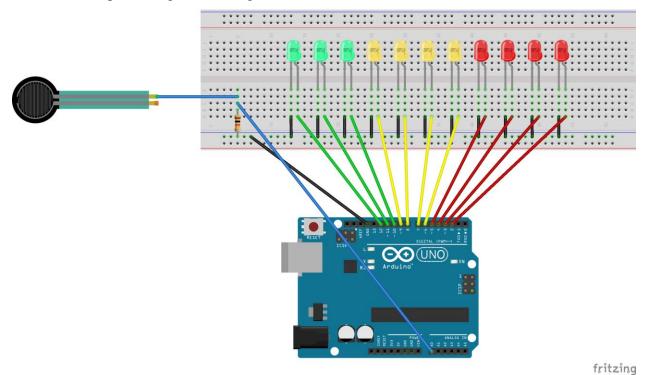


Figura 11: Vista del "breadboard" con las "LEDs" y el Sensor FSR conectado a A0.

• Finalmente, conectaremos el otro extremo del sensor FSR a corriente positiva +5V del Arduino tal como se observa a continuación.

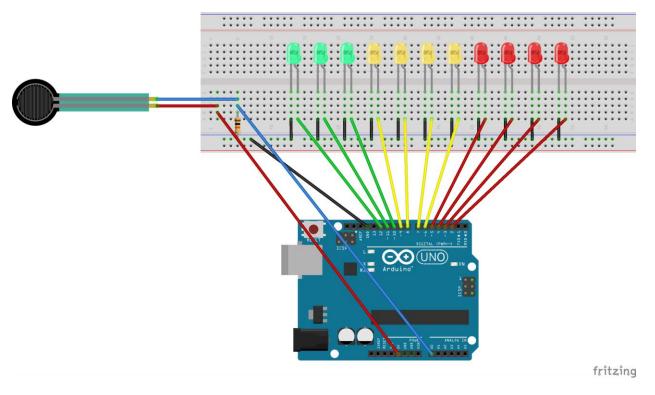


Figura 12: Vista del "breadboard" con las "LEDs" y el sensor FSR conectado a corriente positiva y en paralelo al resistor y A0.

C. Configuración del Código:

- 1. Descargue e instale el Arduino IDE de la siguiente página:
 - https://www.arduino.cc/en/Main/Software
- 2. Conecte la tarjeta Arduino UNO a la computadora por medio del cable USB estándar AB.
- 3. Verifique que el programa esté utilizando la tarjeta Arduino UNO y no otra. Vaya a Tools -> Board y verifique que "Arduino/Genuino UNO" esté seleccionada.
- 4. Crea un nuevo archivo donde va a estar el código del juego. Vaya a File -> New, luego grábelo en File -> Save as.. y póngale un nombre.
- 5. Copie el siguiente código en su nuevo archivo:

```
#define SENS 0
#define LED1 2
#define LED2 3
#define LED3 4
#define LED4 5
#define LED5 6
#define LED6 7
#define LED7 8
#define LED8 9
#define LED9 10
#define LED10 11
#define LED11 12
int value = 0;
void setup() {
 // Parte del código que se ejecútara solo al principio del programa.
 // Se establecen los puertos en modo de salida para enviar señales a los LEDs.
 pinMode(LED1, OUTPUT);
 pinMode(LED2, OUTPUT);
 pinMode(LED3, OUTPUT);
 pinMode(LED4, OUTPUT);
 pinMode(LED5, OUTPUT);
 pinMode(LED6, OUTPUT);
 pinMode(LED7, OUTPUT);
```

```
pinMode(LED8, OUTPUT);
pinMode(LED9, OUTPUT);
pinMode(LED10, OUTPUT);
pinMode(LED11, OUTPUT);
Serial.begin(9600);
void loop() {
// Parte del código que se ejecútara continuamente durante el programa.
// Retraso de 0.1 segundos
delay(100);
// Leyendo el valor análogo dado por el sensor de fuerza FSR
value = analogRead(SENS);
// Reduciendo el valor máximo a 255 en lugar de 1023 para obtener mas control
value = map(value, 0, 1023, 0, 255);
// Verificando el valor del sensor y prendiendo los LEDs de acuerdo al valor dado.
if(value > 243){
 ledUP(11);
  ledDOWN(11);
else if(value > 222){
  ledUP(10);
 ledDOWN(10);
else if(value > 214){
 ledUP(9);
 ledDOWN(9);
else if(value > 206){
 ledUP(8);
 ledDOWN(8);
else if(value > 198){
 ledUP(7);
  ledDOWN(7);
else if(value > 190){
 ledUP(6);
 ledDOWN(6);
else if(value > 182){
 ledUP(5);
  ledDOWN(5);
else if(value > 174){
 ledUP(4);
  ledDOWN(4);
else if(value > 166){
  ledUP(3);
  ledDOWN(3);
else if(value > 158){
 ledUP(2);
```

```
ledDOWN(2);
 else if(value > 150){
  ledUP(1);
  ledDOWN(1);
// Función que se encarga de prender los LEDs hacia arriba
void ledUP(int x){
if(x < 1){
  return;
 else{
  ledUP(x-1);
  digitalWrite(x, HIGH);
  delay(300);
// Función que se encarga de apagar los LEDs hacia abajo
void ledDOWN(int x){
if(x < 1){
  return;
 else{
  delay(300);
  digitalWrite(x, LOW);
  ledDOWN(x-1);
```

- 6. Vuelva a grabarlo en File -> Save.
- 7. Presione Sketch -> Verify / Compile o CTRL+R para verificar y compilar el código.

*Nota: Esta serie de pasos asumen que el usuario ya ha completado la configuración del circuito eléctrico.