	FORMATO		Versión:	0
	PRÁCTICA DE LABORATORIO		Fecha emisión:	12/septiembre/2018
			Página:	5 de 16

CARRERA	PLAN DE ESTUDIO	CLAVE DE ASIGNATURA
Sistemas Computacionales		SCC-1023
NOMBRE DE LA ASIGNATURA	PRACTICA No.	NOMBRE DE LA PRÁCTICA
Sistemas Programables	3	Secuencias de encendido con cubo de LEDs 3X3X3 y sensor infrarrojo

2	COMPETENCIA A DESARROLLAR
---	----------------------------------

El estudiante será capaz de programar el encendido y apagado de Leds mediante el Arduino uno para generar secuencias con retardos de tiempo y controladas por un sensor infrarrojo

3	INTRODUCCIÓN.
---	----------------------

Un secuenciador de luces es un circuito que maneja una determinada cantidad de lámparas distribuidas en distintas formas para dar la sensación visual de luces en movimiento.

Antes de la aparición de los circuitos integrados digitales, estos secuenciadores se construían con un motor de baja velocidad que llevaba en su eje una escobilla, la cual activaba secuencialmente unos contactos eléctricos fijos situados a su alrededor.

Estos contactos servían de interruptores para las lámparas. Este sistema funcionaba bien pero tenía la desventaja del desgaste mecánico de los contactos, lo cual, con el tiempo producía un mal funcionamiento del circuito.

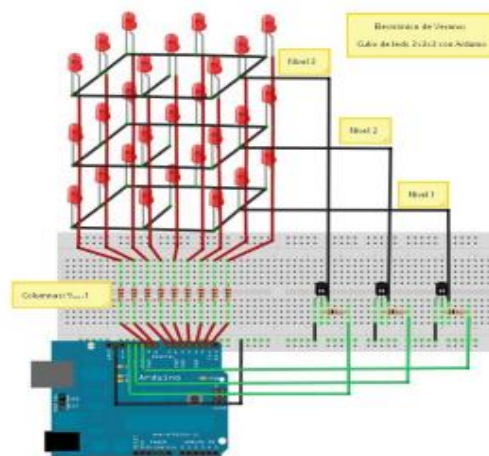
Con la invención de dispositivos electrónicos como los LEDs y hoy en día los microcontroladores, estos circuitos son más fáciles de implementar y además tienen la propiedad de una larga vida útil y menos consumo de energía.

4	MATERIALES Y EQUIPO
---	----------------------------

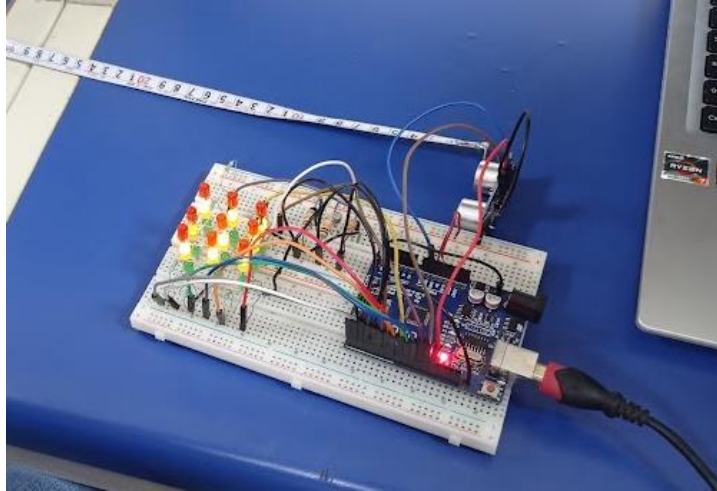
1 protoboard
1 Arduino Uno
1 Computadora con software instalado para arduino
1 Cubo de LEDs de 3X3
1 Sensor infrarrojo
3 Transistores 2N2222
13 Resistencias de 220ohms @ cuarto watt
1 Resistencias de 1000ohms @ cuarto watt
Cables
Pinzas para pelar cables
Multímetro
Pinzas para pelar cables

5 PROCEDIMIENTOS (DESCRIPCIÓN)

1. Implementa el siguiente circuito:



2. Conecta el sensor infrarrojo en el pin digital que no se utilice.
3. Desarrolla un programa que ejecute dos secuencias de encendido/apagado para los LEDs del cubo. Emplea tu creatividad para hacerlas vistosas. Además, mientras el sensor no se active, que solo ejecute una secuencia y cuando sea lo contrario, cambie de secuencia.
4. Registra tus evidencias.



En la imagen anterior se muestra la placa de Arduino conectada al protoboard y al cubo de LEDs 3x3, así como el sensor de distancia. En este caso, se usó el sensor de distancia, asegurándose de utilizar los pines de alimentación y de señal según la configuración elegida. Cada nivel de proximidad detectado por el sensor activa diferentes patrones de iluminación en el cubo de LEDs, representando visualmente la distancia medida.

```

Practica_cubo_leds.ino
26
27 void loop() {
28   digitalWrite(TRIG, HIGH);
29   delay(1);
30   digitalWrite(TRIG, LOW);
31   DURACION = pulseIn(ECHO, HIGH);
32   DISTANCIA = DURACION / 58.2;
33   Serial.println(DISTANCIA);
34   delay(200);
35   // int sequence_1_is_activated = digitalRead(A0);
36   turn_off_cube();
37   turn_on_columns();
38   if (DISTANCIA >= 15 && DISTANCIA <= 30) {
39     digitalWrite(2, HIGH);
40   } else if (DISTANCIA >= 31 && DISTANCIA <= 45) {
41     digitalWrite(3, HIGH);
42   } else if (DISTANCIA >= 46 && DISTANCIA <= 55) {
43     digitalWrite(4, HIGH);
44   }
45 }
46

Practica_cubo_leds.ino
47 void execute_sequence_1() { // Fuente de Ortiz
48   turn_off_cube();
49   Serial.println("Fuente de Ortiz ejecutándose...");
50   digitalWrite(2, HIGH); // Floor 1
51   digitalWrite(9, HIGH); // Column 5
52   delay(DELAY_1);
53   digitalWrite(3, HIGH); // Floor 2
54   delay(DELAY_1);
55   digitalWrite(2, LOW);
56   digitalWrite(4, HIGH); // Floor 3
57   delay(DELAY_1);
58   digitalWrite(3, LOW);
59   turn_on_columns(); // All columns
60   delay(DELAY_2);
61   digitalWrite(3, HIGH);
62   digitalWrite(9, LOW);
63   delay(DELAY_2);
64   digitalWrite(2, HIGH);
65   digitalWrite(4, LOW);
66   delay(DELAY_2);
67   digitalWrite(3, LOW);
68   delay(DELAY_2);
69   digitalWrite(2, LOW);
70 }

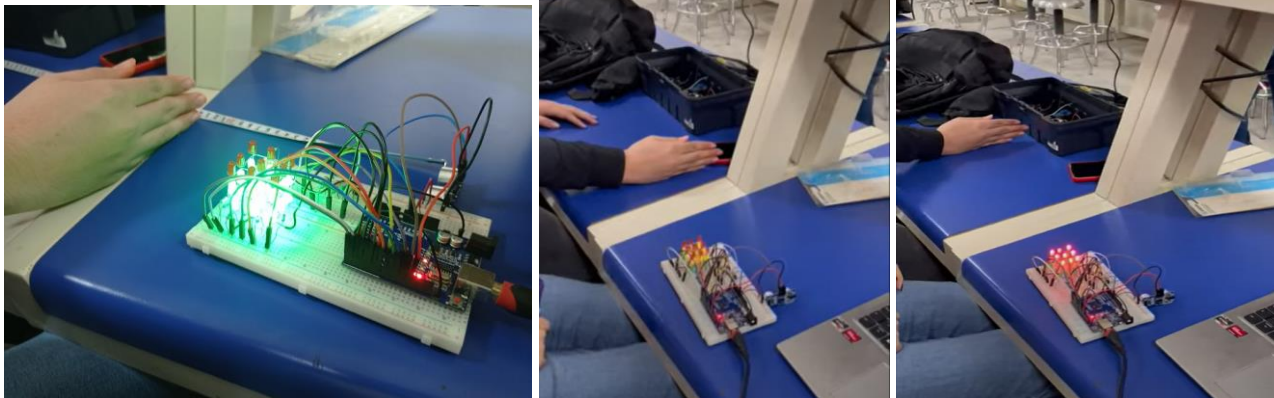
```

Las imágenes anteriores son partes del código que controlan el cubo de LEDs. En el ciclo principal (loop()), el código envía una señal a través del pin de trigger (TRIG) del sensor para medir la distancia, y luego recibe el tiempo que tarda la señal en reflejarse mediante el pin de echo (ECHO). Este tiempo de respuesta se convierte en una distancia real, calculada con la fórmula $DISTANCIA = DURACION / 58.2$, que luego se imprime en el monitor serie.

Dependiendo de la distancia medida por el sensor, se activan diferentes filas de LEDs en el cubo. Si la distancia está entre 15 y 30 cm, se enciende la primera fila de LEDs (en el pin 2); si la distancia está entre 31 y 45 cm, se enciende la segunda fila (pin 3); y si la distancia está entre 46 y 55 cm, se enciende la tercera fila (pin 4). Esto permite que el cubo de LEDs responda dinámicamente a la proximidad del objeto que está siendo medido por el sensor de distancia.

La función `execute_sequence_1()` maneja una secuencia predefinida de activación de LEDs en el cubo.

Primero apaga todos los LEDs, luego enciende las filas y columnas de manera secuencial, simulando una animación. Esta secuencia tiene retardos definidos por las variables DELAY_1 y DELAY_2, y se ejecuta repetidamente para crear una visualización dinámica en el cubo de LEDs.



6

SUGERENCIAS DIDÁCTICAS

Conteste las siguientes preguntas:

1. ¿Es posible con este arduino utilizar un cubo de 4X4? Explique su respuesta.

Sí, es posible, aunque un cubo de LEDs 4x4 tiene muchos LEDs y el Arduino no tiene suficientes pines para conectarlos directamente, se pueden usar métodos como encender los LEDs por capas, es decir se puede hacer que los LEDs se enciendan rápidamente por secciones, dando la ilusión de que todos están encendidos al mismo tiempo. De esta forma, el Arduino puede controlar el cubo sin necesitar un pin para cada LED.

2. ¿Qué pasará si cambiamos los transistores y que sean tipo PNP?

Si usamos transistores PNP, el circuito funcionaría al revés: en lugar de enviar señales positivas para encender los LEDs, el Arduino tendría que enviar señales negativas. Esto implica que habría que ajustar tanto el circuito como el código para controlar los LEDs correctamente, ya que el comportamiento de encendido y apagado cambiaría.