

Práctica 5.- E/S con Arduino – Adenda TinkerCad

Aclaración y objetivos

Con motivo de la pandemia COVID-19 se consideró desaconsejable utilizar los kits Elegoo “*The Most Complete Starter Kit Mega 2560 Project*” en el curso 2020-21, dada la dificultad de limpiar la multitud de diminutos componentes de los numerosos kits. Usaremos en su lugar el **simulador Autodesk Tinkercad Circuits** sugerido en el guion original de la práctica.

Esta Adenda comenta los aspectos más básicos para realizar la práctica con el simulador, aspectos que como ya comenta el guion al final de la primera página, no se cubren en el mismo. En todos los demás aspectos la práctica se continúa realizando igual, lo cual incluye también la **lectura del guion original**.

Profesores

Se sugiere que los profesores se creen cuenta en *Tinkercad Circuits* (<https://www.tinkercad.com>) activando la opción de profesor, para posteriormente crear clases para sus respectivos grupos y poder revisar los diseños realizados de forma más organizada y sistemática. Se sugiere usar la dirección de email profesor@ugr.es (u otra personal) para el rol de profesor. La identificación mediante usuario Google con profesor@go.ugr.es podría utilizarse para probar el rol de estudiante, si se desea.

Estudiantes

Se sugiere que los estudiantes inicien sesión en *Tinkercad Circuits* (<https://www.tinkercad.com>) con el usuario Google e.estudiante@go.ugr.es, y posteriormente se unan a la clase con el código indicado por el profesor.

Monitorización y Evaluación

Respecto a monitorización, el profesor puede comprobar si los estudiantes han entrado en la clase en Perfil > Clases. Se muestra un resumen de cuántos estudiantes han entrado en cada clase (grupo). Entrando en la clase se puede consultar cuáles son los estudiantes. De esta forma se puede avisar a algún estudiante que no haya entrado en la clase de que entre para que sus diseños le aparezcan al profesor.

Respecto a evaluación, dentro de cada clase, en Actividad se pueden ver los diseños realizados por cada estudiante. Si se considera necesario, se podrían bloquear los diseños antes de empezar a corregir, desbloqueándolos en el momento de corregirlos. Aparentemente el navegador cachea el estado y puede ser necesario salir y volver a entrar en la cuenta Tinkercad para visualizar correctamente el estado de bloqueo de un diseño. También se puede revisar estudiante por estudiante, seleccionándolo en la página inicial de la clase.

<https://blog.tinkercad.com/2019/08/05/introducing-tinkercad-classroom/>

<https://tinkercad.zendesk.com/hc/en-us/articles/360026236693-How-to-use-Tinkercad-Classrooms>

<https://tinkercad.zendesk.com/hc/en-us/articles/360058003713-How-do-I-create-a-Class-code-or-link->

<https://tinkercad.zendesk.com/hc/en-us/sections/201003458-Parents-and-Teachers>

Descargar pero no instalar el IDE y el Kit: extraer ejemplos

Se sugiere leer, seguir (realizar) y estudiar el guion original. Los objetivos de la práctica se modifican ligeramente (Conocer el ~~IDE~~ **simulador**, Escribir programas sencillos, Montar pequeños circuitos **simulados**). La instalación del IDE en las páginas 2-3 se puede obviar, aunque es importante dominar los conceptos:

- Alternativa 1. Web Editor (Online IDE). Se puede crear una cuenta en la web de Arduino y editar allí programas Arduino (.ino) que se pueden cargar/subir (*upload*) a una placa Arduino (*board*) conectada en el ordenador local habiendo instalado previamente el Arduino *plugin* en el navegador.
- Alternativa 2. Desktop IDE. Aplicación para editar programas Arduino y subirlos a la placa Arduino conectada al ordenador local.
- Alternativa 3. TinkerCad. Se puede crear una cuenta en TinkerCad y usar la opción Circuits para diseñar el esquemático de un circuito, redactar el código Arduino y simular su funcionamiento. La opción de unirse a una clase permite al profesor revisar los trabajos centralizadamente.

Se sugiere descargar el IDE Arduino (<https://www.arduino.cc/> > Software > Downloads, por ejemplo Windows ZIP) y el kit de desarrollo de Elegoo (<https://www.elegoo.com> > Support > Download > Arduino > Mega 2560 > The Most Complete..., notar que utiliza cuenta Google). No instalaremos ninguna de las descargas, sino que aprovecharemos algunos ejemplos, los mencionados en el guion:

- IDE Arduino: extraer Examples > 01.Basics > **Blink** y 10.StarterKit_BasicKit > p06_LightTheremin
- Elegoo Starter Kit: English > extraer Manual > code > extraer Lesson 7 Passive **Buzzer** y Lesson 26 **PhotoCell**

El resto de la práctica se debería poder realizar leyendo el guion original. Se ofrece el siguiente resumen *telegráfico* para animar a leer dicho guion en lugar de alargar innecesariamente la lectura de este.

Introducción

Leer Guion Práctica 5: Introducción, IDE Arduino, Kit Elegoo, Configuración

Primer Programa (1 Blink)

Leer Guion

Simular con TinkerCad > Circuits (buscar con Google TinkerCad) Acceder con usuario@go.ugr.es

<https://www.tinkercad.com> > Iniciar sesión > Google

Ignorar tutorial diseños 3D > TinkerCad (Diseños) > Circuits

Crear nuevo circuito > Título “1 Blink” > componentes Básico Arduino Uno R3

Iniciar simulación (parpadea L, comentado en guion) > Detener simulación

Código > bloques + texto > texto > bloques

Comentarios guion sobre setup(), loop(), LED_BUILTIN==13

Código > texto, incorporar 01.Basics Blink, comprobar mensaje error redefinición setup/loop

corregir (comprobar LED_BUILTIN==13, leer comentarios, explicación setup/loop, **visitar web Blink**)

realizar modificación sugerida (**acelerar x2**)

probar debugger (parece ir con 2 líneas de retraso)

salir pulsando en TinkerCad

Leer Guion, Visitar webs Arduino

- Fundamentos: <https://www.arduino.cc/en/Tutorial/Foundations>
- Manual Referencia: <https://www.arduino.cc/reference/en/>

Programa 1b (1b Intermitencia)

Crear nuevo circuito > Título “1b Intermitencia” > Componentes Starters Arduino > Intermitencia

Código > bloques + texto > ver mismo código > ocultar código > Iniciar simulación > Detener

Conectar a pin 12, no 13 > simular > código bloqs **cambiar pin 12 500ms** > simular (pin12 = “pasador”12)

Notar marcar cable, 220Ohm limita, led pata + más larga lado no plano, **ahora L no parpadea** (es pin 13)

Programa 1c (1c Blink ASM)

Crear nuevo circuito > Título “1c Blink ASM” > Componentes Starters Arduino > Intermitencia sin delay()

Código > sólo texto > ver millis(), notar previousmillis ámbito global > Iniciar simulación > Detener

Leer comentarios, visitar web Blink w/o delay

Sustituir por código Blink_ASM SWAD > Iniciar simulación > Detener

Simulando paso a paso parece que el tiempo lo cuenta bastante bien pero no se ilumina led

¿simulador no captura asm(cbi/sbi)? Pinmode? Aparentemente my_delay() sí lo interpreta

Corregir **digitalWrite**(LED_BUILTIN,HIGH/LOW) y tal vez **tiempos 500/500ms** para que se vea bien

Leer **Guion**, **entra en examen** (DDRB, PORTB, SBI, CBI, SBIW, BRNE, MOV %A0, %A2)

Segundo Programa (2 Buzzer)

Crear nuevo circuito > Título “2 Buzzer” > Componentes Starters Arduino > Melodía de tonos

Código > sólo texto > ver código similar > Iniciar simulación > Detener

Visitar **web Tone**, viene mal en el comentario

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/toneMelody>

Sustituir por código Elegoo Lesson 7 Passive Buzzer > incorporar **pitches.h** a mano

Hacer algún **cambio a la melodía** para puntuar.

Tercer Programa (3a Light Theremin)

Crear nuevo circuito > Título “3a Light Theremin” > Componentes Starters Arduino > Seguidor de tono

Código > bloques + texto > ver código similar > Iniciar simulación (clickar photocell para slider) > Detener

Comparar con p06_LightTheremin.ino, lleva calibración, usa sólo map() sin pow(constrain), y usa pin8

Consultar map en <https://www.arduino.cc/reference/en/language/functions/math/map/>

Preferimos p06_LightTheremin, reemplazar código, cambiar **código a pin9** o **esquemático a pin8**

Notar que los 5s de calibración pasan muy lento, **reducir a 1s**.

Programa 3b (3b PhotoCell)

Crear nuevo circuito > Título “3b PhotoCell” > CompStarters Arduino > Placa Pruebas + Seguidor de tono

Código > texto > Placa Pruebas viene con Blink, Seguidor de tono con el ya visto en 3a

combinar códigos p06_LightTheremin (controla buzzer) y Lesson 26 PhotoCell (controla leds)

adaptar esquemático en pp.179-181/225 Manual Elegoo

conveniente: dejar buzzer en pin 9, cambiar código **tone(8->9)**, **clockPin=9->10**.