TRABAJO 6
ACTIVIDAD EN EL AULA – PUERTA CORREDERA AUTOMÁTICA

Diseña una actividad para realizar en el aula con Mblock y Arduino.

La actividad debe constar de los siguientes puntos:

- **Objetivos.**
- **Material.**
- **Programa de Arduino**
- **Circuito de TinkerCad**

La entrega constará de dos partes:

1. Archivo .sb2 generado por el programa mBlock o código Arduino resultante.

2. Captura de pantalla de la simulación realizada con TINKERCARD o fotografía del montaje realizado con Arduino.

3. Archivo en word con la descripción de la actividad y las fotos de comprobación de la actividad realizada.

ÍNDICE

ENUNCIADO DE LA PRÁCTICA EN EL AULA

Montaje de apertura y cierre de una puerta corrediza como las habituales en los centros comerciales y tiendas. Estas puertas disponen de un sensor de proximidad para detectar si hay un cliente cerca y actuar dejando paso o cerrando el acceso.

La puerta debe de abrirse si nota la cercanía de alguien y emitir una melodía de bienvenida al abrirse. Además, han de añadirse luces que señalicen el paso (verde) cuando la puerta esté completamente abierta y que adviertan de no pasar (rojo) en caso de que la puerta se esté cerrando. Puede utilizarse para este diseño un led por cada color o un led RGB.

OBJETIVOS DE LA PRÁCTICA

- Saber realizar el montaje y programación de una instalación domótica con los elementos dados.
- Conocer el funcionamiento de un servomotor sencillo de Arduino.
- Entender la relación de la posición del servomotor y los ángulos de giro necesarios.
- Enviar diferentes datos hacia el servomotor y que este los interprete modificando su ángulo de giro.
- Conocer el funcionamiento de un sensor de ultrasonidos y como mediante cálculos matemáticos se puede conocer una distancia.
- Trabajar mediante programación de bloques con estos datos y poder utilizarlos en los actuadores.
- Recibir y enviar datos por los diferentes pines de la placa de Arduino.
- Interrelacionar los valores del sensor de distancia con el servomotor, y realizar una programación estable y funcional.
- Conocer el funcionamiento de un buzzer, su trabajo con frecuencias y como estas equivalen a las notas musicales.
- Enviar mediante los pines de la placa de Arduino dichas frecuencias y que sean reproducidas por el buzzer.
- Conocer el funcionamiento de los diodos LED – LED tricolor y la necesidad de los cálculos para utilizar resistencias y no estropearlos.

MATERIAL OBLIGATORIO A UTILIZAR

El material necesario para la realización del a práctica es el siguiente:

- Placa de Arduino.
- Protoboard.
- Sensor de ultrasonidos.
- Servomotor.
- Buzzer.
- Resistencias (220Ω por cada color que se pretenda utilizar y 100Ω para el buzzer).
- LEDs de colores o LED RGB.
- Cableado de conexión.

Placa de Arduino

Se trata de una placa programable de código abierto (open source) tanto por el software como el hardware. Esto hace que sea accesible a todos los niveles de aprendizaje al haber millones de programaciones y diseños compartidos por la red, estos diseños pueden enseñarnos desde cero o ser la base de proyectos más complicados a desarrollar.

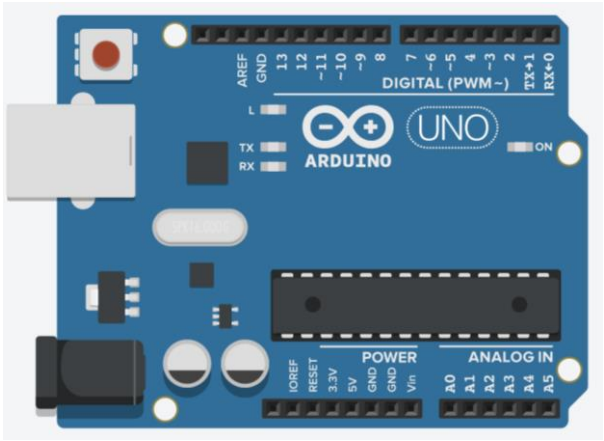

Figura 1.- Placa de Arduino.

Protoboard

Se trata de una placa con puntos interconexionados por la parte de debajo, como se muestra en la figura 2, que sirve para colocar los elementos de circuitería y trabajar con ellos en los proyectos. Puede utilizarse de manera virtual mediante programas gratuitos como TINKERCAD, o de manera física.

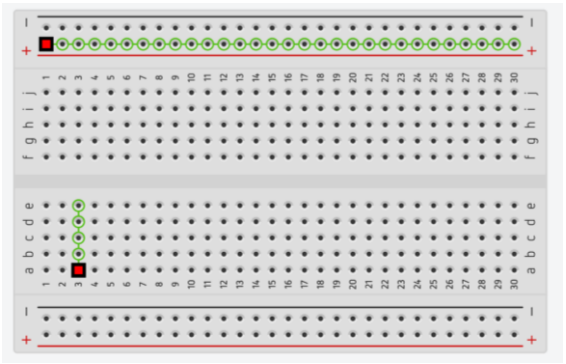

Figura 2.- Protoboard.

Sensor de ultrasonidos

El sensor de ultrasonidos, se utiliza para medir distancias o salvar posibles obstáculos. Su funcionamiento es muy simple: se envía desde uno de sus cilindros (transductor) un infrasonido o vibración no audible que rebota en el obstáculo u objeto y se recibe por el segundo cilindro (receptor). Lo que recibe el Arduino es el tiempo que este ultrasonido tarda en regresar, y por lo tanto puede calcular la distancia al objeto.

Las distancias en las que trabaja este sensor son de 3 metros a 3 centímetros, con una precisión de 3 milímetros. Hay que tener en cuenta que no es capaz de detectar nada a menos de 3 centímetros de distancia del sensor.

Existen sensores de ultrasonidos de 3 y 4 patillas. En el caso de tener 4 patillas:

- Vcc: alimentación de 5V.
- GND: conexión a tierra.
- TRIG: envía el pulso ultrasónico.
- ECHO: recibe el pulso ultrasónico.

Y en el caso de tener solamente 3 patillas, significaría que tanto TRIG como ECHO irían por el mismo PIN del sensor de ultrasonidos, simplificando así la programación por bloques.

Figura 3.- Sensor de ultrasonidos.

Servomotor

Se trata de un pequeño motor que gira hasta 360º. Existen diferentes modelos, de los cuales podemos ver uno de ellos en la figura 4, y todos ellos tienen el mismo cableado:

- Vcc: 5V.
- GND: tierra.
- Señal.

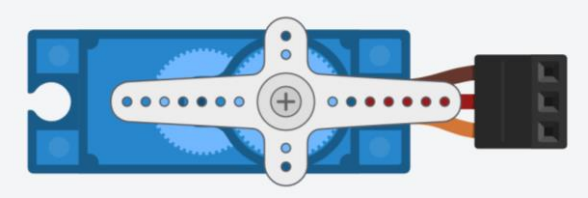

Figura 4.- Servomotor.

Vcc y GND nos sirven para mantener el servo alimentado, mientras que la señal es necesaria para dar las instrucciones pertinentes. Por medio de esta señal, se envían las órdenes (vía PIN) que el servomotor traducirá en giros de X grados.

Buzzer

El buzzer o zumbador es un actuador sencillo que consta de un electroimán y una fina lámina de acero que al enviar los parámetros adecuados emite un rango de notas, pudiendo así avisar mediante pitidos o crear melodías básicas.

Figura 5.- Buzzer

En caso de necesitar variaciones de notas que no existen en la programación por bloques, se puede modificar directamente en el código de Arduino por los valores que vemos en la siguiente tabla.

COMPONENTE

Buzzer o zumbador:

	0	1	2	3	4	5	6	7	8
Do	16,35	32,70	65,41	130,81	261,63	523,25	1046,50	2093,00	4186,01
Do#	17,32	34,65	69,30	138,59	277,18	554,37	1108,73	2217,46	4434,92
Re	18,35	36,71	73,42	146,83	293,66	587,33	1174,66	2349,32	4698,64
Re#	19,45	38,89	77,78	155,56	311,13	622,25	1244,51	2489,02	4978,03
Mi	20,60	41,20	82,41	164,81	329,63	659,26	1318,51	2637,02	5274,04
Fa	21,83	43,65	87,31	174,61	349,23	698,46	1396,91	2793,83	5587,65
Fa#	23,12	46,25	92,50	185,00	369,99	739,99	1479,98	2959,96	5919,91
Sol	24,50	49,00	98,00	196,00	392,00	783,99	1567,98	3135,96	6271,93
Sol#	25,96	51,91	103,83	207,65	415,30	830,61	1661,22	3322,44	6644,88
La	27,50	55,00	110,00	220,00	440,00	880,00	1760,00	3520,00	7040,00
Si	30,87	61,74	123,47	246,94	493,88	987,77	1975,53	3951,07	7902,13

Las filas representan las notas y las columnas las escalas.
En cada celda está la frecuencia en Hz de la nota en la escala correspondiente.

3729,31 7458,62

Tabla 1.- Valores de frecuencia de sonidos del buzzer.

Resistencias y cableado de conexión

Las resistencias se oponen al flujo de corriente enviado mediante el cableado para adaptarlo a nuestras necesidades y solucionar posibles problemas de saturación en nuestros circuitos.

Existen resistencias de múltiples valores, para su adecuación concreta a nuestros diseños, que van descritas mediante un código de colores en su superficie. En la parte de LEDS, calcularemos concretamente varias resistencias necesarias para el diseño de nuestro circuito y así comprender su función.

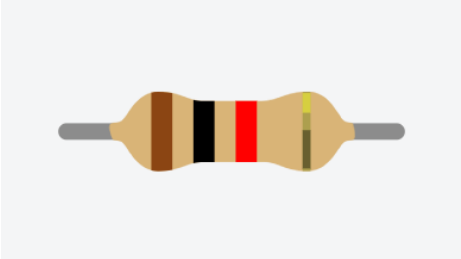

Figura 6.- Resistencia.

El cableado a utilizar en estar prácticas se trata de pequeños hilos de cobre que transmiten la corriente entre los elementos de nuestro circuito. El cobre, o aleaciones de cobre, es un medio transmisor económico que nos permite interconectar nuestros circuitos.

LEDS

Un diodo LED es un elemento electrónico que solamente permite pasar la corriente en una dirección (para ello dispone de ánodo y cátodo). Al ser emisor de luz, cuando la corriente pasa en la dirección correcta, además de dejarla pasar, se ilumina.

Dicho de otra manera: un diodo Led es un diodo que cuando está polarizado correctamente emite luz.

Este elemento dispone de dos patillas:

- Larga: ánodo, polo positivo.
- Corta: cátodo, polo negativo o tierra.

Para la realización de la práctica lo primero que debemos hacer son los cálculos de las resistencias que debemos utilizar de la misma manera que se nos muestra en la teoría. En nuestro caso utilizaremos 8 diodos rojos para su programación con los siguientes valores:

- I=17mA
- V=1.8V

Para que nuestro diodo tenga valores de tensión funcionales debemos tener en cuenta que la placa de Arduino nos da 5V, y utilizándolo al completo quemaríamos nuestros LEDs. Con éste y los anteriores datos, es suficiente para calcular matemáticamente (mediante la ley de OHM) los valores de las resistencias que necesitamos para su correcto funcionamiento:

V = I x R -> R = V / I -> R = (5V-1.8V) / 0.017A = 3.2V / 0.017A = 188Ω

En caso de no tener resistencias de ese valor, se utilizará la inmediatamente superior. Esto causará que se ilumine un poco menos al bajar el valor de la tensión de entrada en el LED, pero será totalmente funcional.

En este caso, al utilizar el simulador de TINKERCAD, es posible introducir el valor concreto. Si no fuera así, habría que utilizar una resistencia de 200 o 220Ω.

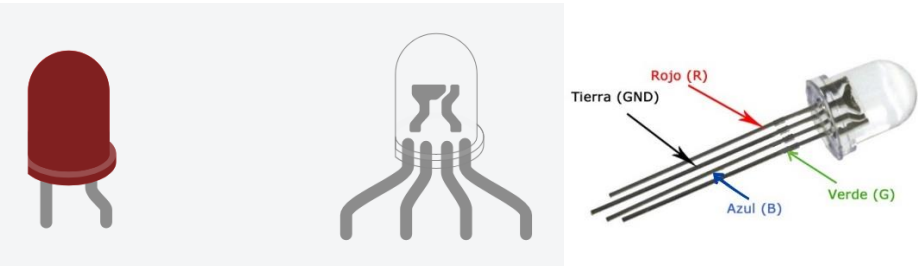

Figura 7.- Led simple y LED RGB con su conexionado.

Existe un tipo de led que integra 3 simples que funciona de la misma manera que sus predecesores. Este led dispone de una patilla o ánodo para cada color, y un cátodo o tierra compartido.

De esta manera pueden hacerse las combinaciones que se quieran de esto tres colores básicos. Podemos comprobar el conexionado en la figura 7 a la derecha.

PROGRAMACIÓN DE ARDUINO EN MBLOCK

Para la realización del programa de Arduino, lo primero que debemos realizar es la carga de nuestro dispositivo (Arduino UNO) en mBlok en la pestaña correspondiente. Una vez realizado esto, comenzamos con el diseño del programa.

Figura 8.- Programación de bloques en mBlock.

Vamos a utilizar los siguientes PINES de la placa de Arduino para la programación, que deberán corresponder con el conexionado en la protoboard que veremos más adelante (figura 9).

ELEMENTO	PIN	TIPO
Servomotor	0	Salida de datos
LED verde	1	Salida de datos
LED rojo	2	Salida de datos
Buzzer	3	Salida de datos
Sensor infrasonidos	4	Lectura de datos

Tabla 2.- Asignación de pines de la placa de Arduino.

Como podemos observar en la figura 8, mediante un bucle infinito, cada 0.5 segundos se comprueba mediante el pin de lectura de datos (4) la posición del objeto en frente de l sensor de infrasonidos. Esto nos da la posición del cliente, diciéndonos si se encuentra delante de la puerta o no.

Después de recoger estos datos en la variable DIST, mediante un bucle de condiciones, describimos las dos opciones que pueden ejecutarse:

1. El cliente se encuentre a menos de 20 centímetros del sensor (SI DIST < 20): en este caso, se activará el servomotor hasta los 90º abriendo así la puerta. Mientras se está abriendo, se estima un tiempo de 1 segundo que se mantiene el LED rojo para advertir de que no se pase ya que se está procediendo a la apertura.

   Una vez se termina la apertura de la puerta, el led cambia de posición y pasa a verde y a su vez sonaría una melodía de bienvenida mientras el cliente pasa el umbral.

2. El cliente se encuentra a más de 20 centímetros del sensor (CASO CONTRARIO A DIST < 20): en este caso se procederá al cierre total de la puerta con el LED rojo encendido para advertir de no pasar.

En el caso de aparecer un cliente mientras la puerta este en cierre, nuevamente se entraría en la condición de apertura y por como está diseñada la programación, se mantendría el LED rojo encendido para advertir de no cruzar por seguridad.

Los tiempos de cambios de iluminación de los LEDs con respecto del giro del servomotor y distancia del sensor de infrasonidos pueden reajustarse dependiendo de la estimación de tiempo que tardarían los clientes en cruzar el arco de la puerta y las dimensiones de esta.

De la programación de bloques realizada en mBlock, podemos exportar el código para utilizar en nuestro esquema del simulador TINKERCAD:

```
// generated by mBlock5 for <your product>
// codes make you happy

#include <Servo.h>
#include <Arduino.h> //ELIMINAR EN TINKERCAD PARA CORRECTO FUNCIONAMIENTO
#include <Wire.h>
#include <SoftwareSerial.h>

float DIST = 0;

float getDistance(int trig,int echo){
    pinMode(trig,OUTPUT);
    digitalWrite(trig,LOW);
    delayMicroseconds(2);
    digitalWrite(trig,HIGH);
    delayMicroseconds(10);
    digitalWrite(trig,LOW);
    pinMode(echo, INPUT);
    return pulseIn(echo,HIGH,30000)/58.0;
```

```
}
Servo servo_0;

void _delay(float seconds) {
  long endTime = millis() + seconds * 1000;
  while(millis() < endTime) _loop();
}

void setup() {
  servo_0.attach(0);
  pinMode(2,OUTPUT);
  pinMode(1,OUTPUT);
  pinMode(3,OUTPUT);
  while(1) {
      DIST = getDistance(4,4);
      _delay(0.5);
      // Cuando se acerca alguien al sensor a menos de 20 cm, primero
      // se abre la puerta y depués (por seguridad) se enciende la
      // luz verde. 1 segundo es lo que suponemos que tarda en abrir.
      if(DIST < 20){
          servo_0.write(90);
          _delay(1);
          digitalWrite(2,0);
          digitalWrite(1,1);
          tone(3,110,0.5*1000);
          delay(0.5*1000);
          _delay(0.2);
          // Melodía de bienvenida.
          tone(3,123,0.1*1000);
          delay(0.1*1000);
          tone(3,131,0.1*1000);
          delay(0.1*1000);
          tone(3,175,0.25*1000);
          delay(0.25*1000);
          _delay(1);

      }else{
          // Cierre de la puerta al no estar el cliente u objeto a menos
de 20cm.
          servo_0.write(0);
          digitalWrite(2,1);
          digitalWrite(1,0);

      }

      _loop();
  }
```

```
}

void _loop() {
}

void loop() {
  _loop();
}
```

Para que el código funcione correctamente en TINKERCAD, por problemas de compatibilidad, debemos de eliminar la línea resaltada en amarillo en la parte superior del código:

```
#include <Arduino.h>
```

Se realizan comentarios en el propio código (en color verde y precedidos por //) para localizar las diferentes partes de la programación, que unido a la descripción del código en esta parte de la práctica describe su desarrollo.

DISEÑO DEL CIRCUITO EN TINKERCAD

El cableado (diseño) de nuestro circuito para que se corresponda con lo descrito en la tabla 1 es el siguiente:

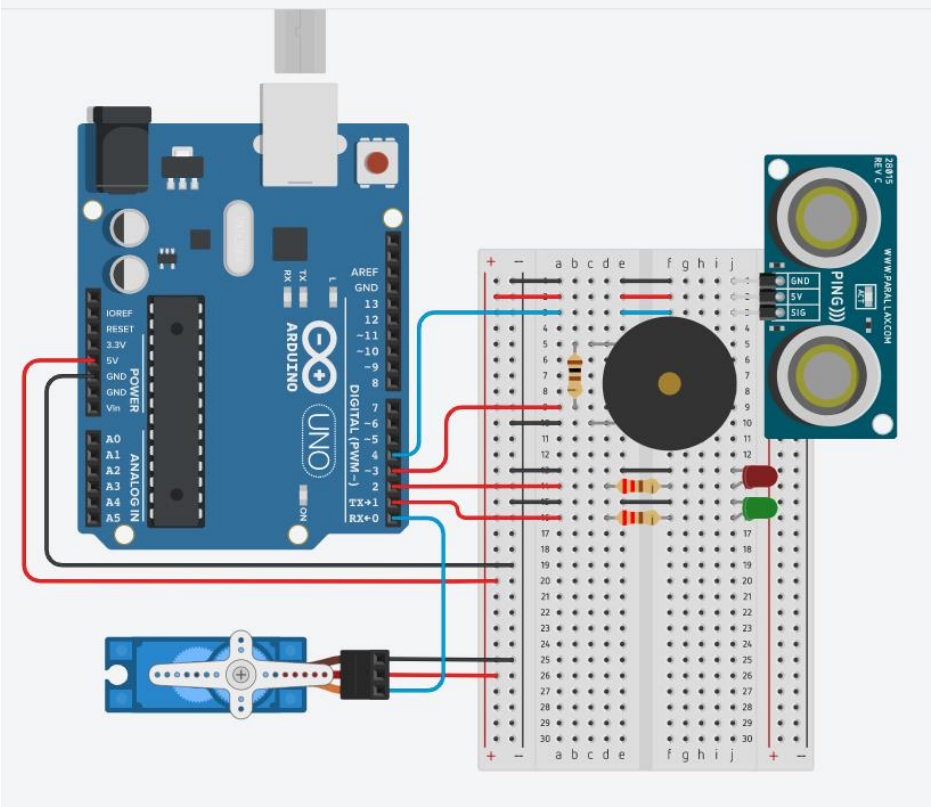

Figura 9.- Diseño del circuito en TINKERCAD.

BIBLIOGRAFÍA

Material:

Folgado, L., & Sampedro, A. (2021). Iniciación a Arduino. Presentación, Curso on-line consejería de educación de Castilla y León.

Fotografías:

Tinkercad | From mind to design in minutes. Tinkercad. (2021). Retrieved 10-14 February 2021, from https://www.tinkercad.com/.

mBlock - One-Stop Coding Platform for Teaching and Learning. Mblock.makeblock.com. (2021). Retrieved 10-14 February 2021, from https://mblock.makeblock.com/en-us/.