



Actividad | 2 |

Alarma para Incendios

Internet de las Cosas

Ingeniería en Desarrollo de Software



academi**ag**lobal

TUTOR: Marco Alonso Rodríguez

ALUMNO: Yanira Lizbeth Lopez Navarro

FECHA: 27/10/2024

Índice

Introducción	3
Descripción	4
Justificación	5
Armado del Circuito	6
Codificación.....	8
Emulación del Circuito	11
Conclusión	12
Referencias	13

Introducción

En la actualidad, el Internet de las Cosas (IoT) se ha convertido en una herramienta fundamental en el desarrollo de soluciones tecnológicas que mejoran la seguridad y la calidad de vida en nuestros hogares. Esta actividad, titulada "Alarma para Incendios", se enmarca dentro de esta tendencia, utilizando componentes electrónicos para crear un sistema de alerta que responda a la detección de gases peligrosos. A través de la integración de un sensor de gas y un sensor piezoeléctrico controlados por una placa Arduino, los estudiantes aprenderán a programar un sistema que emite una alarma cuando se identifican niveles de gas que podrían representar un riesgo de incendio.

Esta actividad permite comprender la importancia de la automatización y el monitoreo en tiempo real, conceptos centrales en el IoT. Al aprender a conectar y programar los componentes, los participantes desarrollan habilidades técnicas que son altamente valoradas en el mundo laboral actual. También se fomenta la creatividad y la resolución de problemas, ya que deben diseñar un sistema que sea eficaz y confiable. A través de esta experiencia práctica, los estudiantes no solo adquirirán conocimientos teóricos, sino que también podrán ver cómo sus proyectos pueden contribuir a un entorno más seguro y eficiente. En definitiva, "Alarma para Incendios" es una oportunidad para integrar la teoría con la práctica en un contexto relevante y contemporáneo.

Descripción

La actividad propuesta se centra en el diseño de un sistema de alarma contra incendios utilizando tecnología de sensores. Partiendo de un contexto previo donde se utilizó un sensor de movimiento para crear una alarma contra robos, esta nueva tarea implica un enfoque diferente, enfocándose en la detección de gases potencialmente peligrosos.

Para llevar a cabo esta actividad, se requiere la integración de varios componentes, como un sensor de gas, un sensor piezoeléctrico, una placa Arduino, entre otros. Este conjunto no solo facilita la detección de gases, sino que también permite la emisión de una alerta sonora cuando se detecta un nivel de gas que supere el umbral establecido (600 en este caso).

El proceso comienza con la conexión de los componentes en una placa de pruebas, seguido de la codificación necesaria en el entorno de Arduino. Este incluye la declaración de variables y la configuración de los pines para los sensores. La lógica programada en el ciclo principal (void loop) es crucial: activa la alarma cuando se alcanza la concentración de gas especificada, asegurando que se tomen medidas de seguridad adecuadas.

Esta actividad no solo permite a los estudiantes aplicar sus conocimientos de programación y electrónica, sino que también subraya la importancia de la seguridad en el hogar y la utilización de tecnologías inteligentes para prevenir situaciones de riesgo. Además, refuerza el aprendizaje sobre cómo los dispositivos pueden comunicarse y reaccionar ante el entorno, un principio fundamental en el Internet de las Cosas.

Justificación

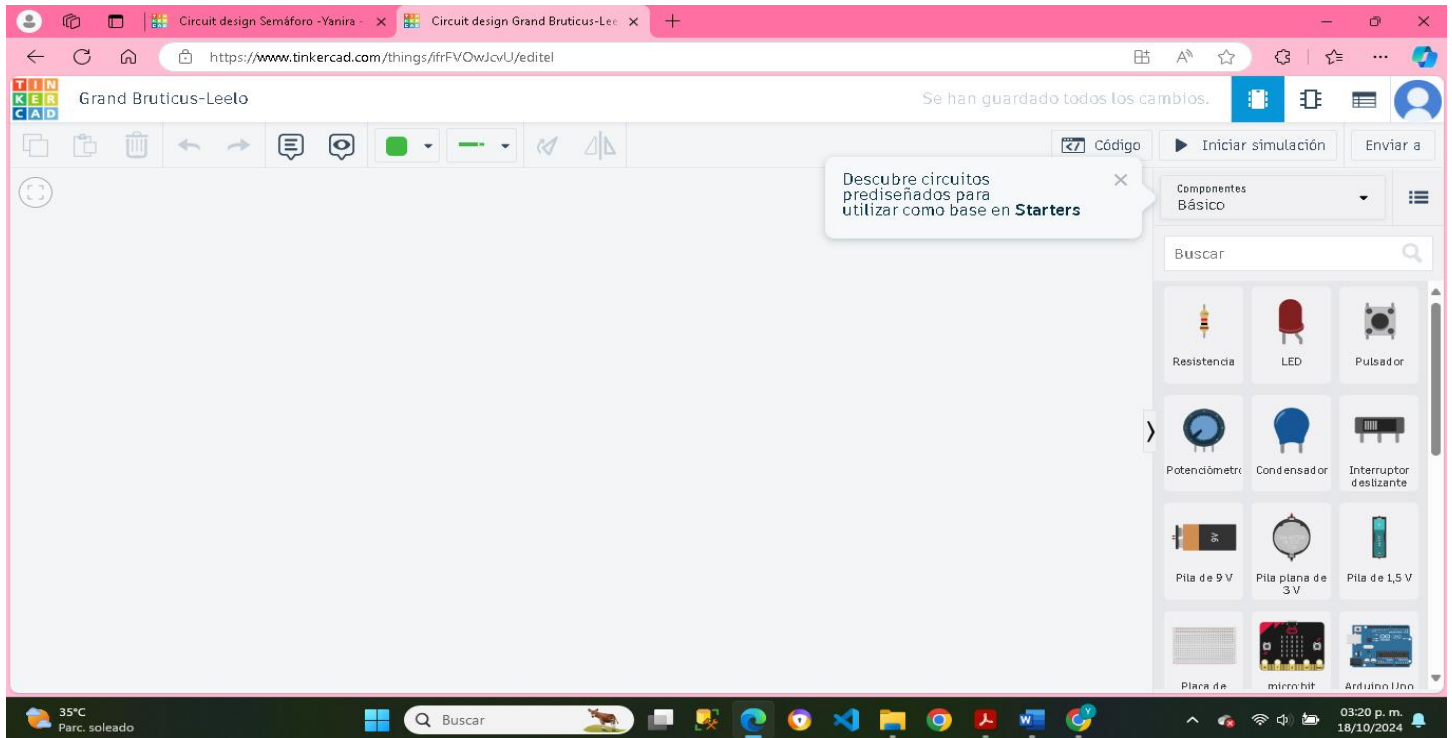
La implementación de un sistema de alarma para incendios basado en el Internet de las Cosas (IoT) es una solución altamente efectiva y pertinente en el contexto actual. En primer lugar, la capacidad de detectar gases peligrosos de manera temprana es crucial para prevenir incendios y minimizar riesgos a la vida y la propiedad. Este tipo de tecnología no solo mejora la seguridad en los hogares, sino que también proporciona tranquilidad a los usuarios, sabiendo que cuentan con un sistema de alerta que funciona de manera autónoma.

Además, el uso de componentes como el sensor de gas y el sensor piezoeléctrico, junto con la placa Arduino, permite a los estudiantes involucrarse en un proceso práctico de aprendizaje que combina electrónica y programación. Este enfoque práctico fomenta el desarrollo de habilidades técnicas que son esenciales en el mercado laboral actual, donde la innovación tecnológica es constante.

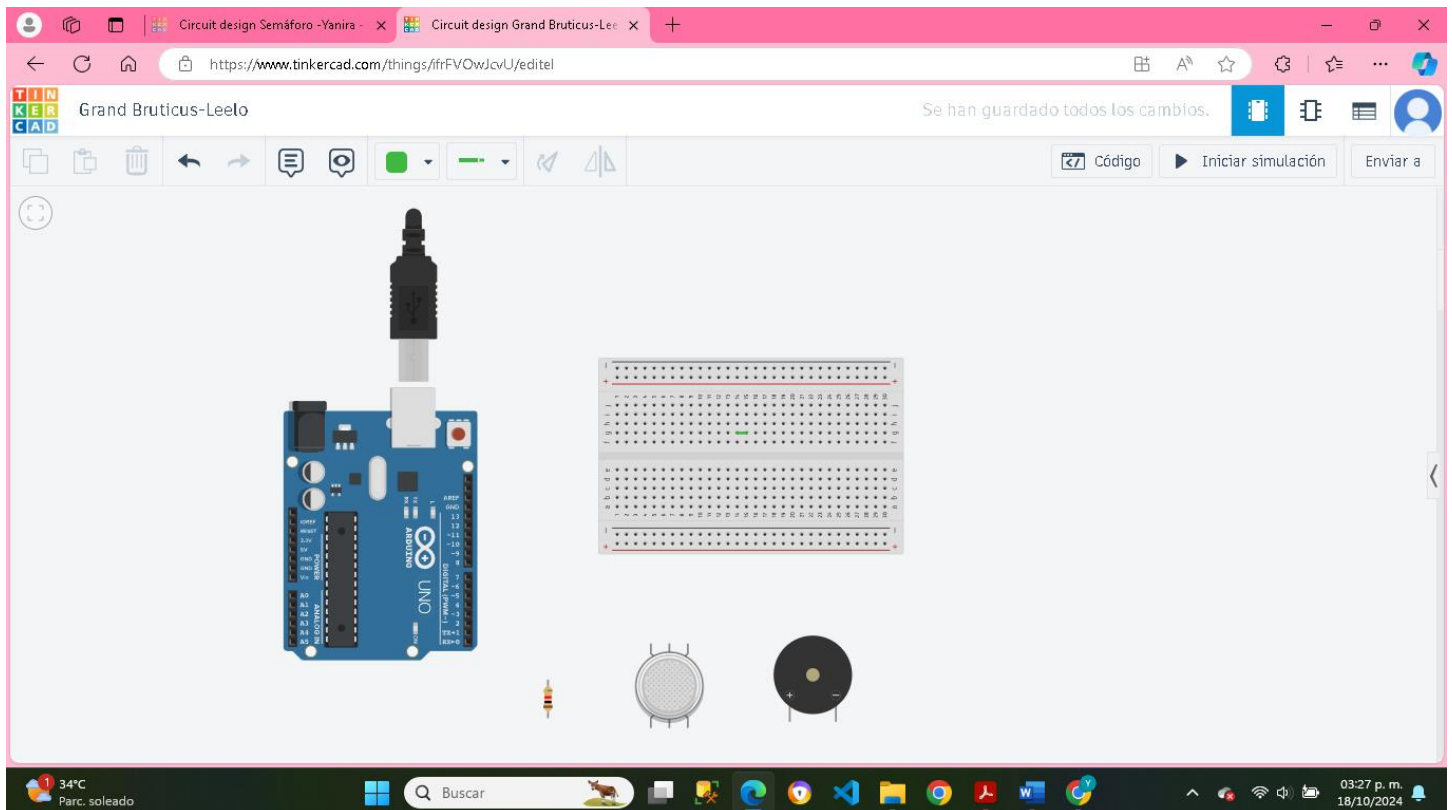
Al integrar sistemas de alerta en la vida cotidiana, se promueve una cultura de prevención y responsabilidad frente a emergencias. La educación en estas áreas prepara a los estudiantes no solo para ser consumidores de tecnología, sino también creadores y solucionadores de problemas, capaces de diseñar sistemas que contribuyan a la seguridad y bienestar social.

Armado del Circuito

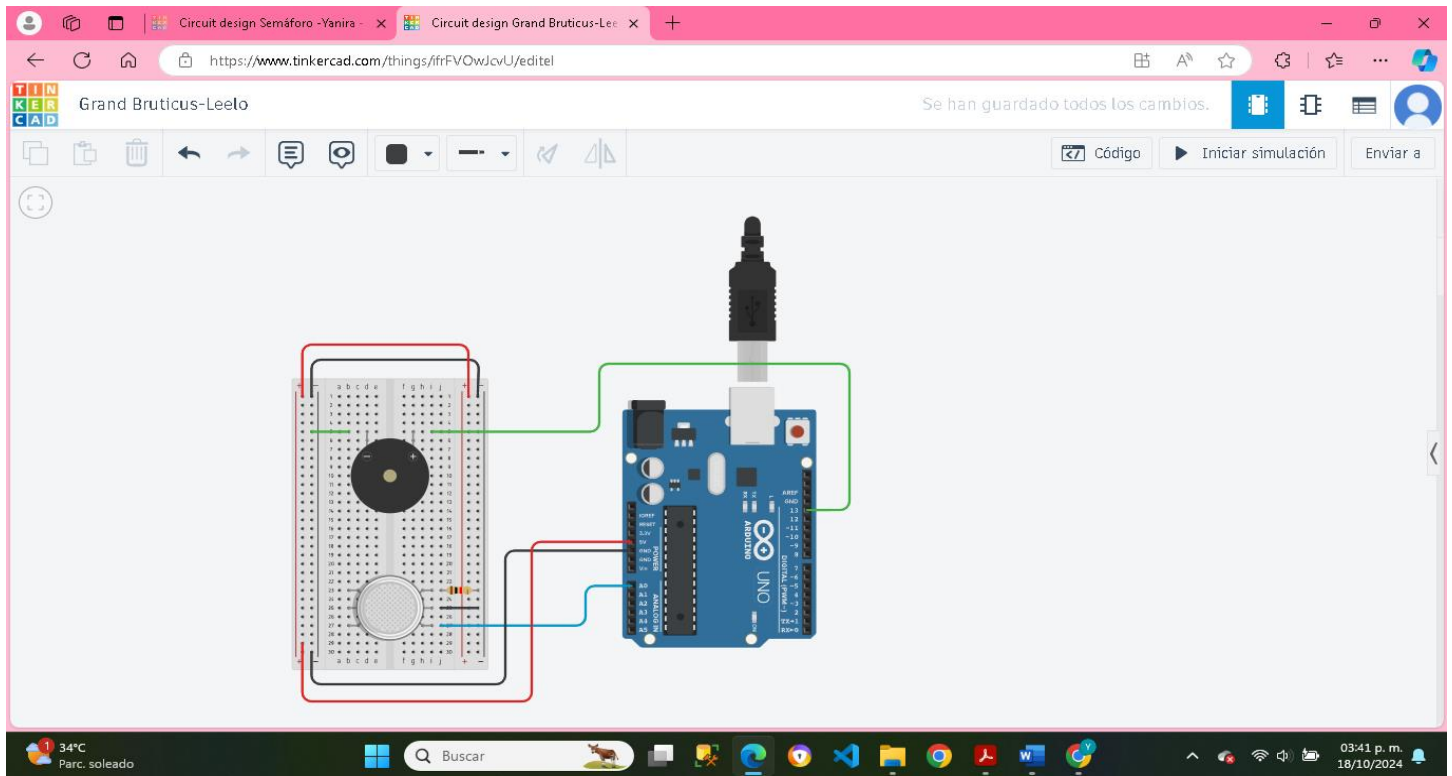
En la siguiente imagen podemos observar el inicio en la herramienta Tinkercad, la cual nos permitirá diseñar nuestro circuito Alarma para Incendios.



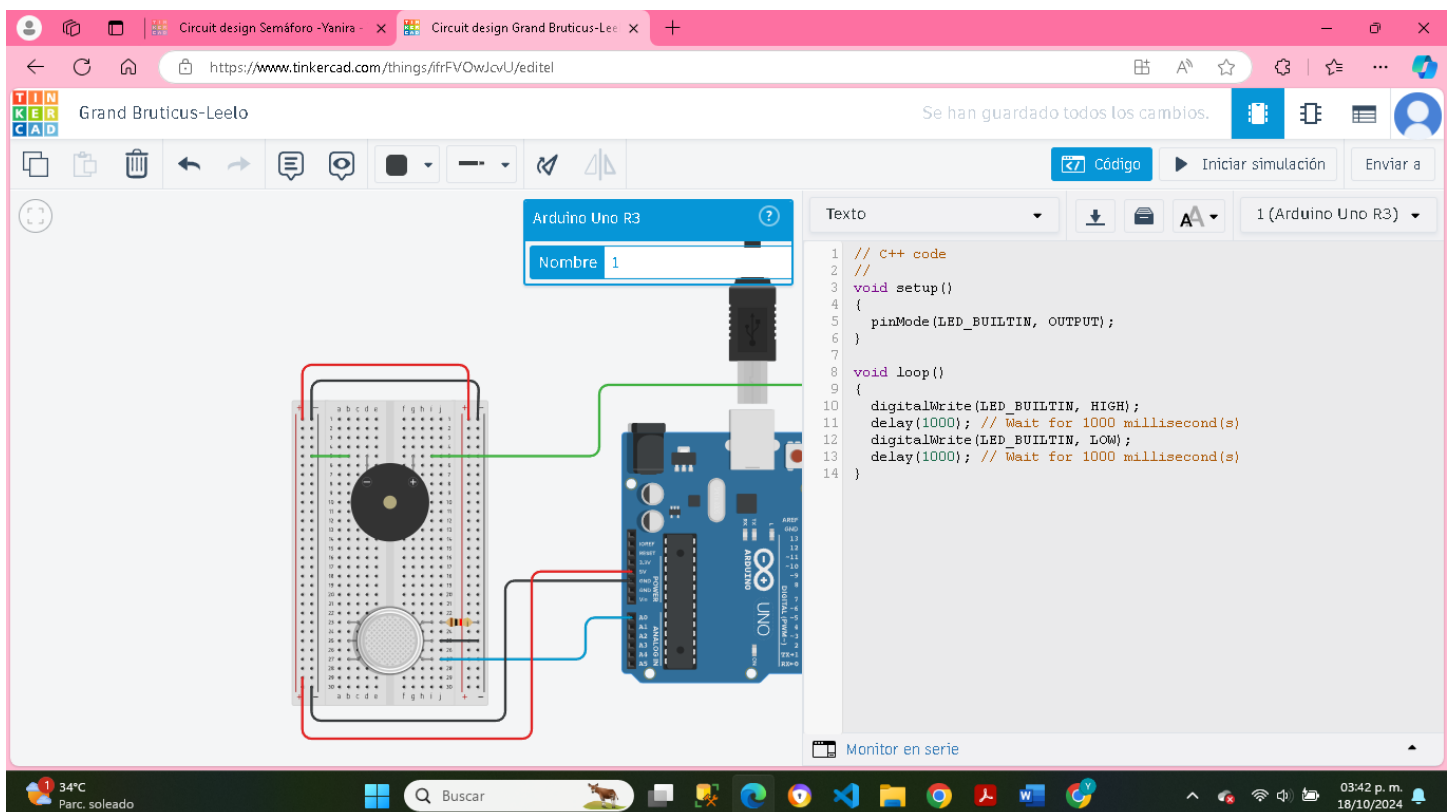
A continuación podemos observar cada uno de los componentes que formaran nuestro circuito.



En la siguiente imagen se muestra nuestro circuito armado con cada uno de los componentes anteriores.



En seguida se muestra el inicio del código, al cual se le realizarán cambios que permitan cumplir con lo solicitado en la actividad.



Codificación

Esta parte del código define tres variables enteras: `gasSensor`, `piezo` y `gasValue`. La variable `gasSensor` se asigna al valor `A0`, representando el pin analógico 0 del Arduino, donde se conectará el sensor de gas. La variable `piezo` se asigna al valor `13`, que corresponde al pin digital 13, donde se conectará el piezoeléctrico que emitirá el sonido de alarma. Por último, `gasValue` se inicializa en `0`, y servirá para almacenar el valor que el sensor de gas leerá, facilitando así su uso en la lógica del programa.

```
int gasSensor = A0
int piezo = 13;
int gasValue = 0;
```

En esta sección se define la función `setup`, que configura los modos de los pines. La función `pinMode (gasSensor, INPUT);` establece el pin del sensor de gas como entrada, permitiendo que el Arduino reciba datos de este sensor. La línea `pinMode (piezo, OUTPUT);` configura el pin del piezoeléctrico como salida, habilitando al Arduino para enviar señales que lo activan. Además, `pinMode (LED_BUILTIN, OUTPUT);` define el LED integrado como salida, lo que permitirá usarlo como indicador visual. Finalmente, `Serial.begin(9600);` inicia la comunicación serial a 9600 baudios, facilitando la depuración y el monitoreo de datos en el monitor serial.

```
void setup () {
  pinMode (gasSensor, INPUT);
  pinMode (piezo, OUTPUT);
  pinMode (LED_BUILTIN, OUTPUT);
  Serial.begin(9600);
}
```

Esta parte define la función `loop`, que se ejecuta repetidamente. La línea `gasValue = analogRead(gasSensor);` lee el valor analógico del sensor de gas a través del pin `A0` y lo almacena en la variable `gasValue`. Esto permite que el programa monitoree continuamente el nivel de gas. A continuación, `Serial.print("Valor del sensor de gas: ");` envía un mensaje al monitor serial, preparando al usuario para el valor que se mostrará. Finalmente, `Serial.println(gasValue);` imprime el valor leído, permitiendo al usuario visualizar en tiempo real la concentración de gas detectada.


```
void loop () {  
  gasValue = analogRead(gasSensor);  
  Serial.print("Valor del sensor de gas: ");  
  Serial.println(gasValue);
```

Aquí se establece una condición usando `if (gasValue >= 600)`. Si el valor del sensor de gas es mayor o igual a 600, se activa la alarma. La función `tone (piezo, 1000, 500)`; genera un tono de 1000 Hz en el piezoeléctrico durante 500 milisegundos, emitiendo una alerta sonora. Además, `Serial.println("¡Alarma activada!");` envía un mensaje al monitor serial, informando que la alarma ha sido activada.

```
  if (gasValue >= 600) {  
    tone (piezo, 1000,500);  
    Serial.println("¡ Alarma activada!");
```

En esta sección, `digitalWrite (LED_BUILTIN, HIGH)`; enciende el LED integrado, proporcionando una señal visual que indica que la alarma está activa. Luego, `delay (100)`; introduce un retraso de 100 milisegundos para mantener el LED encendido brevemente. Después, `digitalWrite (LED_BUILTIN, LOW)`; apaga el LED, y `delay (100)`; añade otro retraso, permitiendo que el sistema tenga un breve intervalo antes de volver a leer el sensor.

```
    digitalWrite (LED_BUILTIN, HIGH);  
    delay (100);  
    digitalWrite (LED_BUILTIN, LOW);  
    delay (100);
```

Esta parte contiene el bloque `else`, que se ejecuta si el valor del gas es menor a 600. La línea `noTone(piezo)`; detiene cualquier sonido en el piezoeléctrico, desactivando la alarma sonora. Luego, `digitalWrite (LED_BUILTIN, LOW)`; asegura que el LED permanezca apagado, indicando que la alarma no está activa. Finalmente, `Serial.println("Alarma desactivada.");` imprime un mensaje en el monitor serial, informando al usuario que la alarma ha sido desactivada.

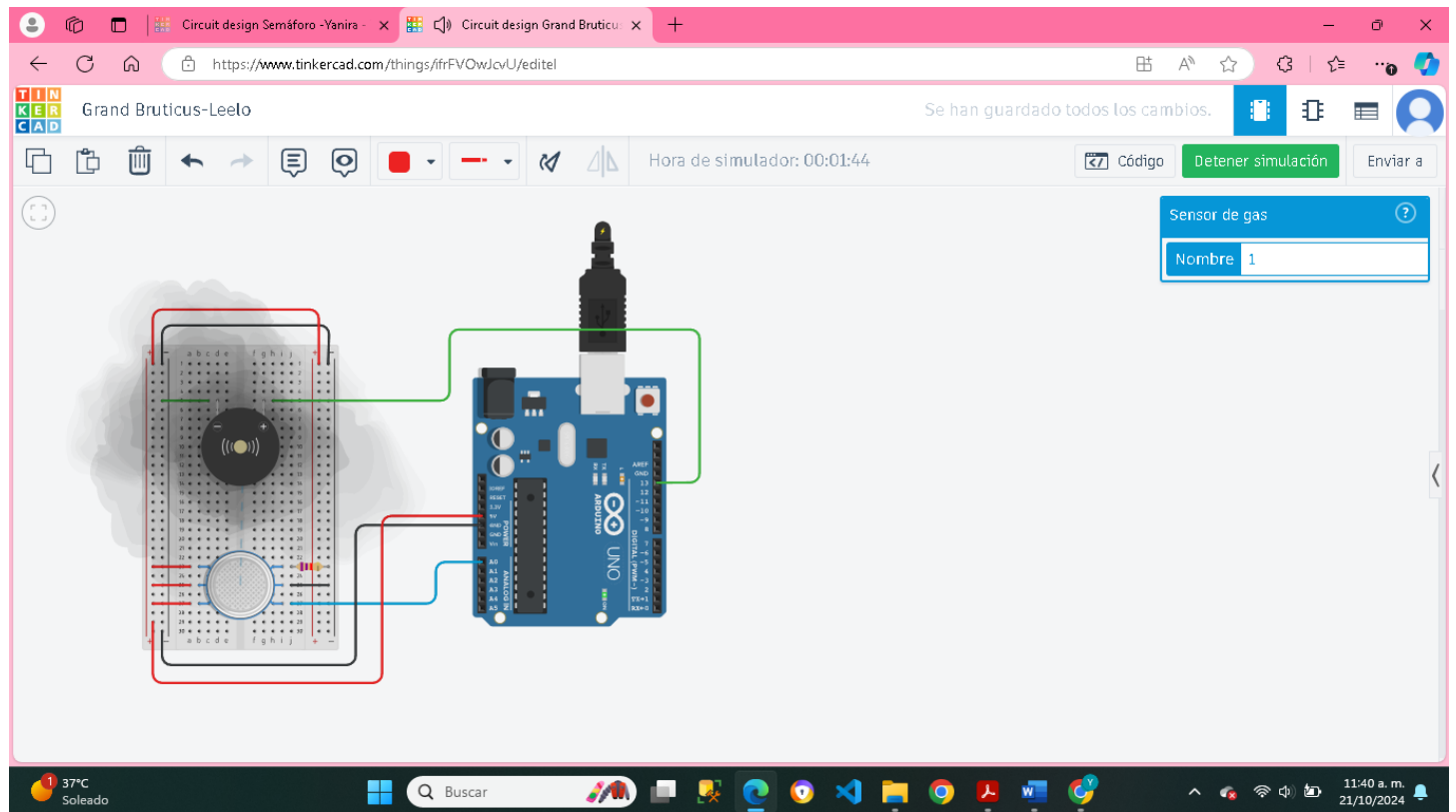
```
  } else {  
    noTone(piezo);  
    digitalWrite (LED_BUILTIN, LOW);  
    Serial.println("Alarma desactivada.");
```

Por último, `delay (1000);` introduce un retraso de 1000 milisegundos (1 segundo) antes de que el ciclo loop comience de nuevo. Esto permite que el sistema pause entre lecturas del sensor, evitando que se active la alarma por fluctuaciones menores en los niveles de gas.

```
}  
    delay (1000);  
}
```

Emulación del Circuito

A continuación, podemos observar el circuito andando correctamente en la simulación, una vez que se realizaron las configuraciones solicitadas.



Conclusión

La actividad de desarrollar un sistema de alarma para incendios utilizando tecnología del Internet de las Cosas es de gran relevancia tanto en el ámbito laboral como en la vida cotidiana. En el contexto profesional, los conocimientos adquiridos en programación, electrónica y diseño de sistemas son altamente demandados en un mercado que valora la innovación y la automatización. Al aprender a integrar diferentes componentes tecnológicos, los estudiantes se preparan para enfrentar desafíos reales en la creación de soluciones efectivas y seguras.

En la vida cotidiana, la implementación de sistemas de alarma mejora la seguridad en los hogares, brindando a las personas una mayor sensación de protección. Este tipo de tecnologías promueve la prevención y la rápida respuesta ante situaciones de emergencia, lo que puede salvar vidas y reducir daños materiales. Además, al fomentar una mentalidad de innovación y responsabilidad, estas actividades preparan a los estudiantes para convertirse en agentes de cambio en sus comunidades. En definitiva, la actividad no solo desarrolla habilidades técnicas, sino que también tiene un impacto significativo en la seguridad y el bienestar de la sociedad.

Referencias

Ingeniería en desarrollo de software. Universidad México Internacional. Recuperado el día 21 de octubre de 2024
<https://umi.edu.mx/coppel/IDS/mod/scorm/player.php>

Video conferencing, web conferencing, webinars, screen sharing. (s. f.-c). Zoom. https://academiaglobal-mx.zoom.us/rec/play/Sv7BKXzo3WNjmgCXsoXqHxQp6DTdMFDP5hH2fsvrX0ljqTAapHNyZ5O1QwRf5Z6Z2t_KYmG1AuJ9CFgb.BxpqhnXfziRbSIsM?canPlayFromShare=true&from=share_recording_detail&continueMode=true&componentName=rec-play&originRequestUrl=https%3A%2F%2Facademiaglobal-mx.zoom.us%2Frec%2Fshare%2FjllLSxd5Y9OEGpbTkYp_DIH9Ge7195PtXtkdtqOI7sdrA9Dn514ONfzLL6mtARHk.Vxsx7AyYXzMivya

Video conferencing, web conferencing, webinars, screen sharing. (s. f.-d). Zoom. https://academiaglobal-mx.zoom.us/rec/play/cDs0rIs8eIZcYEtmEol9pf177oXJ3hms2XDRjQXTGNLei_AcfICnXfOROULN_T2xRzmCNWg10WyfFo.ovaXVyKksJN_SNr?canPlayFromShare=true&from=share_recording_detail&continueMode=true&componentName=recplay&originRequestUrl=https%3A%2F%2Facademiaglobal-mx.zoom.us%2Frec%2Fshare%2Favxh4XrCEuyb0U75EMiltE7k2R5FArnUaq5GGgdKR8R5uC_YK5TdX8s4ofaOSv.HSg38bxNYciKp-be