

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Organización Computacional

PROYECTO BUSCAMINAS

Douglas Josué Martínez Huit 201708975

Pablo Josué Sosa Antonio 202000680

Isai Eliezer Magdiel Guevara 202100179

Billy David Must Ochoa 202000353

Introducción

El juego buscaminas es un juego icónico del sistema operativo Windows. Este proyecto se desarrolló en base al juego buscaminas, implementando tecnologías experimentales, fusionando la electrónica y desarrollo de aplicaciones para móviles y pc.

Para el desarrollo de la aplicación móvil se utilizó el recurso de MIT app inventor, la aplicación de escritorio se desarrolló utilizando Python. Para conectar la aplicación móvil y la de escritorio, se utilizó el módulo Bluetooth HC-06 y el Arduino Mega.

Objetivo General:

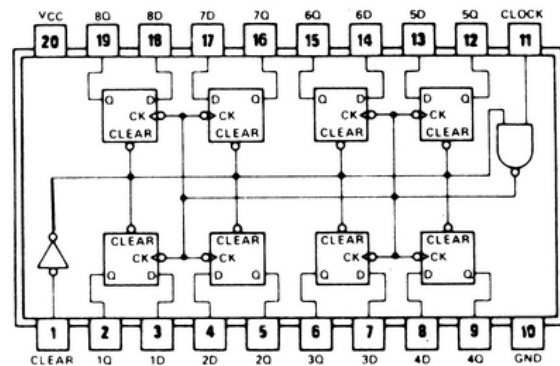
Desarrollar una versión innovadora del juego Buscaminas que integre plataformas móviles y de escritorio mediante tecnologías de comunicación Bluetooth, combinando herramientas de desarrollo de aplicaciones como MIT App Inventor, Python y el uso de Arduino Mega para la conectividad entre dispositivos, a fin de ofrecer una experiencia de juego interactiva y multidispositivo.

Objetivos Específicos:

1. Crear una versión funcional del juego Buscaminas para dispositivos móviles mediante MIT App Inventor, asegurando una interfaz intuitiva y atractiva para los usuarios.
2. Diseñar una aplicación de escritorio del juego Buscaminas en Python, incorporando funcionalidades avanzadas para la experiencia del usuario en computadoras.
3. Implementar la comunicación entre la aplicación móvil y la de escritorio utilizando el módulo Bluetooth HC-06 y Arduino Mega, permitiendo la sincronización de datos y jugabilidad en tiempo real.
4. Integrar y probar el sistema completo, asegurando una interacción fluida y sin interrupciones entre la aplicación móvil, la aplicación de escritorio y el hardware utilizado.
5. Evaluar el rendimiento y la estabilidad de la comunicación Bluetooth entre dispositivos móviles y de escritorio, identificando áreas de mejora en el rendimiento de la aplicación y la conexión.

La configuración del juego está conformada por: 5 push buttons para la entrada de los 4 bits a guardar, 2 push buttons para elegir la fila en la que se desean ingresar los bits, 1 push button para ingresar la información y un botón para hacer reset, cuenta con una matriz construida por 16 leds para mostrar en donde hay bombas y cuenta con 16 flip flops.

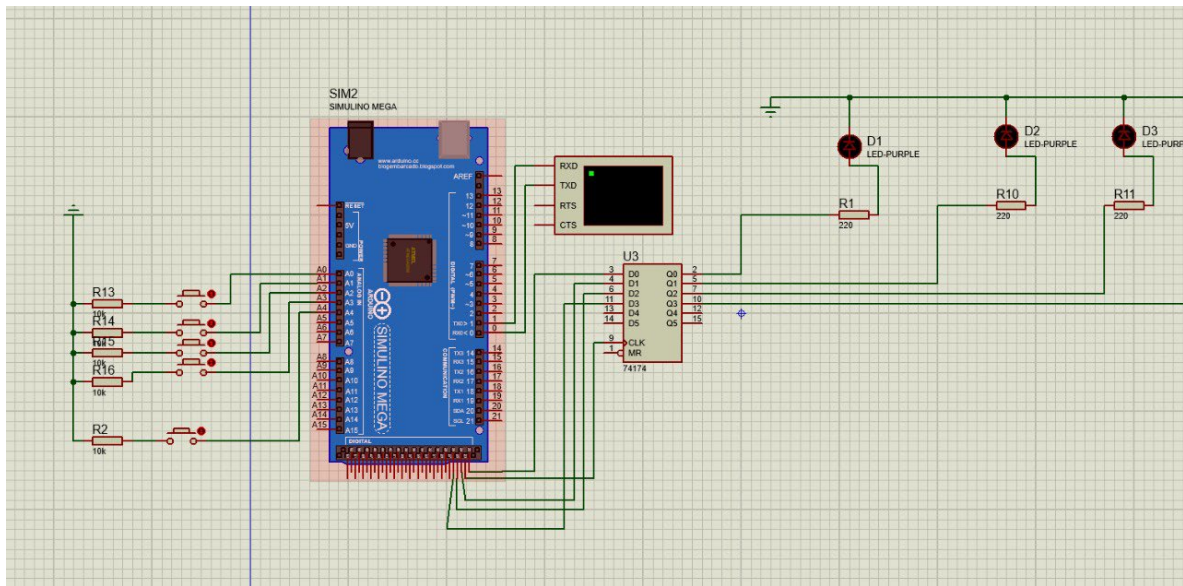
Datasheet del flip flop tipo D 74ls273.



Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

Cada flip flop representa una fila de la matriz, así que las entradas de los 4 flip flops de cada integrado son los 4 bits que se desean guardar, para que la matriz guarde los datos en la fila correcta se hace el uso de un demultiplexor el cual, dependiendo de la elección de la fila envía un pulso al flip flop de la fila elegida.

Circuito completo:



Matriz de led :

Las matrices de LED's están conformadas por una serie de filas y columnas en donde hay un LED en cada una de las intersecciones. Para que un LED encienda se deberá recibir un "0" en la fila y un "1" en la columna simultáneamente.

Módulo Bluetooth:

Este módulo se utilizó para establecer las conexiones entre el móvil y el Arduino que fijará las coordenadas del flip flop para marcar donde estarán las bombas, estableciendo la conexión en las entradas establecidas para hacerlo funcionar.

Display:

El juego simulado tendrá un display que muestre un mensaje acerca de si el jugador perdió o ganó. Conectado en las entradas Rx y Tx.

Generación de el programa de escritorio:

Por medio de esta herramienta se generó los puertos, seriales CCOM4 los cuales se utilizaron para enlazar Python con Arduino, los cuales fueron claves para poder comunicarse con proteus.

Para la virtualización se utilizó una pequeña aplicación de escritorio en Diseñada en Python, la cual cuenta con una pequeña interfaz gráfica con una matriz de botones la cual representa la matriz del juego de Busca minas:

La implementación del código fue la siguiente:

```

1 import serial
2 import time
3 import tkinter as tk
4 from tkinter import messagebox, scrolledtext
5 import threading
6
7 # Configuración del puerto serial
8 try:
9     puertoSerial = serial.Serial('COM4', 9600, timeout=1)
10    time.sleep(2) # Esperar a que se establezca la conexión serial
11 except Exception as e:
12     messagebox.showerror("Error", f"No se pudo abrir el puerto COM4: {str(e)}")
13     exit()
14
15 # Función para leer datos desde el Arduino y mostrarlos en la interfaz
16 def leer_serial():
17     while True:
18         if puertoSerial.in_waiting > 0:
19             linea = puertoSerial.readline().decode('utf-8').rstrip()
20             if linea:
21                 text_area.config(state='normal') # Habilitar el área de texto
22                 text_area.insert(tk.END, linea + '\n')
23                 text_area.config(state='disabled') # Deshabilitar el área de texto
24                 text_area.see(tk.END)
25             time.sleep(0.1)
26
27 # Crear la ventana de la interfaz gráfica
28 root = tk.Tk()
29 root.title("Simulador de Pulsaciones")
30
31 # Marco para la salida de texto
32 text_area = scrolledtext.ScrolledText(root, width=50, height=20, state='disabled')
33 text_area.pack(pady=10)
34
35 # Etiqueta de instrucciones
36 label = tk.Label(root, text="Selecciona un botón para simular la pulsación")

```

Se utilizó la librería, PySerial, la cual se encarga de comunicarse a través de puertos seriales, en este caso se comunica al segundo puerto generado por el virtual serial port, COM4:

También se utilizó la librería time, para poder utilizar un hilo que permita la espera de respuestas del lado de Arduino.

Para la implementación de la interfaz gráfica se utilizó la librería TKinter, de Python, por medio de la cual se generaron las ventanas y los botones y toda la implementación del módulo de usuario que esta aplicación conlleva.

Ejemplo del código utilizado para la interfaz gráfica, creación de ventanas y funcionalidades:

Para realizar el siguiente contador se utilizaron los siguientes materiales Placa de Cobre.

Acido férrico.

Barreno y Broca para perforar la placa.

Fuente de alimentación 5v.

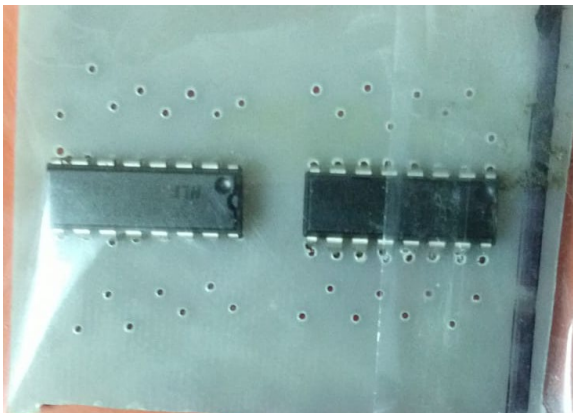
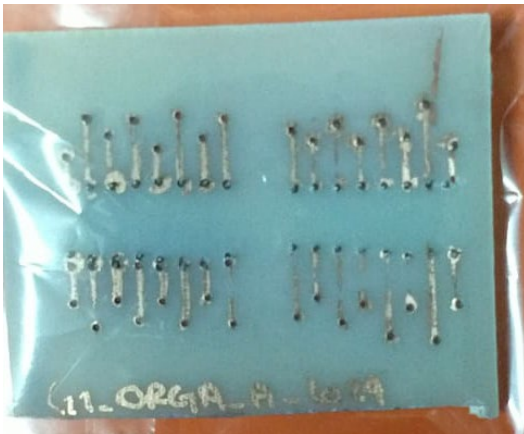
2 Switch o push button.

1 Resistencia de 1k.

9 Resistencias de 330.

8 leds de cualquier color.

Fotos de la placa



Conclusiones:

- La integración de MIT App Inventor para la aplicación móvil y Python para la aplicación de escritorio permitió desarrollar una versión funcional del juego Buscaminas en ambas plataformas, demostrando la viabilidad de utilizar herramientas de desarrollo accesibles y versátiles para crear aplicaciones interactivas.
- La implementación del módulo Bluetooth HC-06 y el Arduino Mega logró una comunicación eficiente entre la aplicación móvil y la de escritorio, lo que evidencia que es posible sincronizar dispositivos de diferentes tipos y sistemas operativos en tiempo real para ofrecer una experiencia de juego unificada.
- La combinación de electrónica y desarrollo de software para crear una experiencia de juego interactiva y multidispositivo no solo resultó en una versión mejorada del juego, sino que también aportó un aprendizaje significativo en la integración de tecnologías experimentales.
- Las pruebas de funcionamiento y conectividad realizadas entre dispositivos móviles y de escritorio revelaron áreas de mejora en cuanto a la estabilidad de la conexión Bluetooth y la optimización del rendimiento, lo cual es fundamental para proyectos futuros que busquen aprovechar tecnologías inalámbricas en aplicaciones de entretenimiento.