



Ingeniería Civil en
Telecomunicaciones
Universidad de Concepción

UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

TIC 2
549305-1

2024-1

Profesor Vincenzo Alexo Caro Fuentes

Mini proyecto 1

Integrantes:

Gabriel Nicolas Figueroa Garcia.
Martin Nicolas Manriquez Molina
Ivan Ignacio Runin Diaz

Concepción, 22 de abril del 2024

Indice:

Resumen:	3
Actividad 1.1:	4
Esquemático del circuito:	4
Comentarios:	4
-Actividad 2.1	5
Esquemático del circuito:	5
Comentarios:	6
-Actividad 2.2	7
Esquemático del circuito:	7
Comentarios:	7

Resumen:

Whack-a-mole: En este MiniProyecto 1 del curso, el cual consistió en la implementación de un juego interactivo llamado "Whack-A-Mole" utilizando el microcontrolador Arduino, usando los sensores proporcionados en el kit como LEDs como "topos" y botones como martillos. Los aspectos clave incluyen definir distintos niveles de dificultad asociados con tiempos de duración del encendido de LEDs y el tiempo total para cada ronda, generación aleatoria de secuencias de LEDs, monitoreo de botones para golpear "topos" iluminados, puntaje del jugador con condiciones de término del juego, y una melodía de finalización.

Ítem 1.1: Se establece un circuito con LEDs y botones, y se desarrolla un programa en Arduino. El programa inicia seleccionando una dificultad inicial y genera secuencias aleatorias de LEDs. Monitorea los botones para detectar golpes exitosos o errados, actualiza el puntaje y reproduce sonidos. La condición de término del juego se define por el puntaje del jugador.

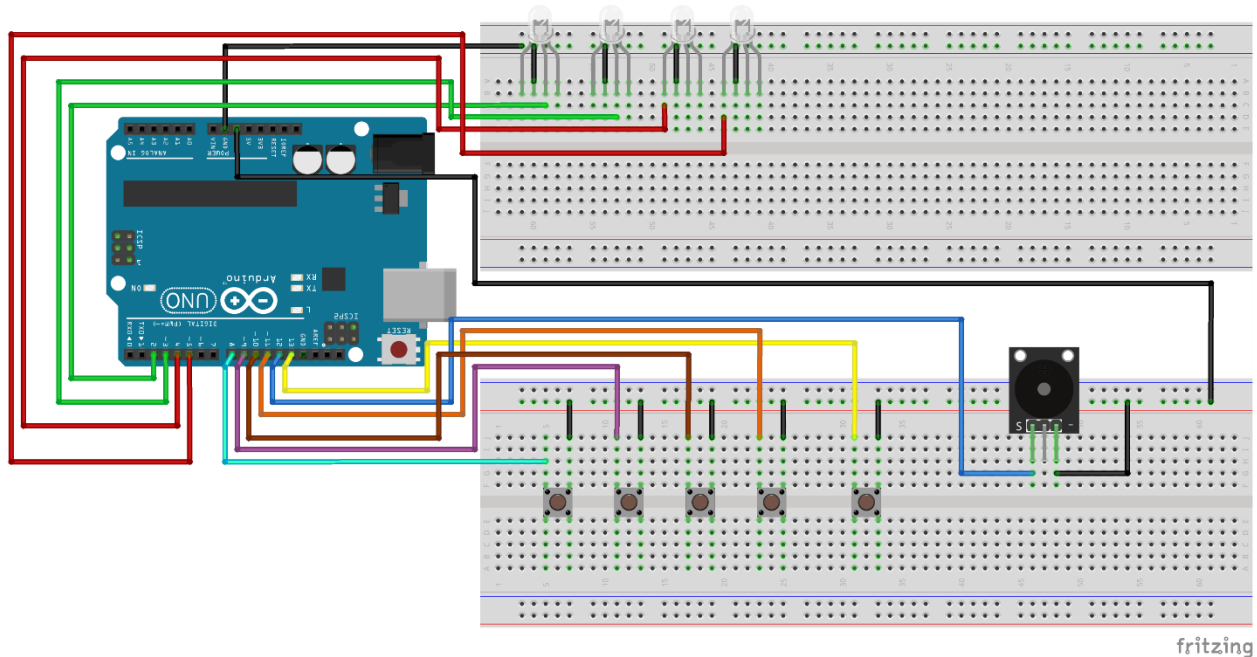
The Game of Life: En este proyecto, se trabajó en una especie de juego llamado "The Game of Life", donde se simulan pequeñas células que pueden estar vivas o muertas en una cuadrícula. Estas células interactúan según algunas reglas simples.

Ítem 2.1: En esta parte del proyecto, se adaptó un código base para crear una versión visual del juego. Se utilizó un programa llamado PyQt6 para construir una interfaz gráfica, y se implementaron temporizadores para actualizar el juego en tiempo real. También se estableció una comunicación entre un programa de Arduino y el juego creado, para controlar algunos aspectos del juego desde el hardware.

Ítem 2.2: En esta etapa, se introdujeron nuevas reglas al juego. Ahora, las células tienen una vida total y se ven afectadas por condiciones climáticas extremas y una guerra. Se integraron sensores de temperatura y botones para representar estas condiciones en el juego. Además, se implementaron botones adicionales en la interfaz gráfica para controlar las acciones de guerra, como lanzar una bomba nuclear o una bomba de curación.

Actividad 1.1:

Esquemático del circuito:



Comentarios:

Este código define la estructura y la lógica de un juego en el que los jugadores deben presionar los botones correctos en respuesta a la iluminación aleatoria de los LEDs. primero, se establecen los pines para los LEDs, botones y otros componentes. El juego inicia con un puntaje y nivel predeterminados, y presenta funciones para reiniciar el juego y reproducir melodías de derrota. En el bucle principal, se selecciona aleatoriamente un LED para encenderlo, se espera un tiempo determinado antes de apagarlo, y se verifica si el botón correspondiente al LED activo fue presionado. Dependiendo de la acción del jugador, se ajusta el puntaje, se reproduce un sonido y se actualiza el nivel de ser necesario. El juego finaliza si el puntaje del jugador llega a cero o menos, momento en el que se reproduce una melodía de derrota y se reinicia el juego automáticamente.

En el desarrollo de la actividad 1, el código que proporcionamos fue para lograr de manera exitosa el juego interactivo. Tratamos de organizarlo en secciones claras que hacen que sea lo más fácil de entender qué hace cada parte de este. Mientras trabajamos en él, tuvimos desafíos como decidir cómo estructurar la lógica del juego o cómo manejar la interacción entre botones y LEDs. Algunos de los desafíos más importantes fueron el funcionamiento irregular de los LEDs, ya que no respondían como esperábamos; actuaban de manera extraña y no tenían el funcionamiento esperado con los botones. Además, nos encontramos con el problema de que, en un momento dado, solo un botón era el que acertaba los golpes a los

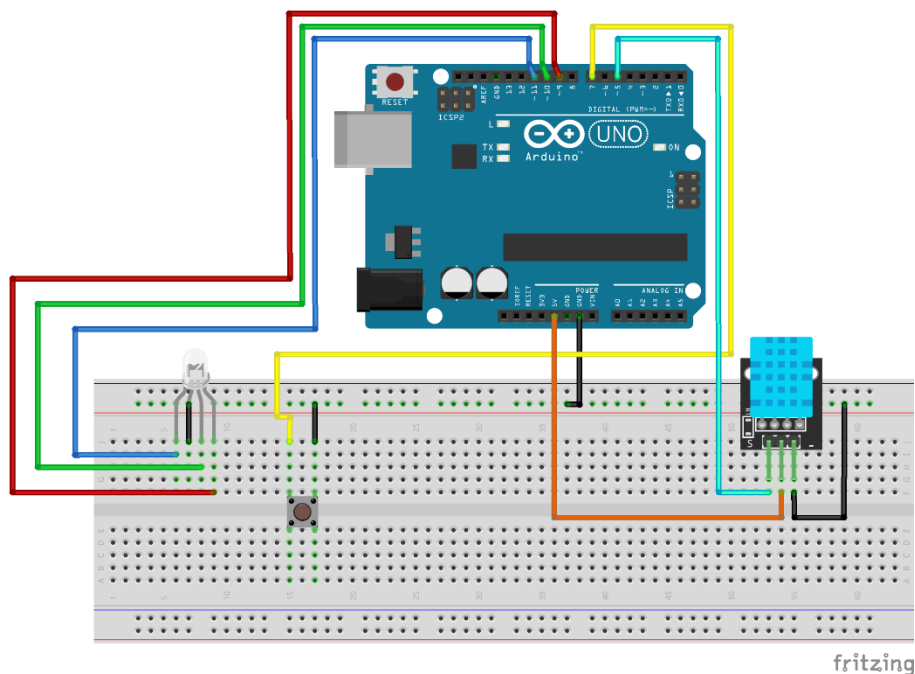
topos, no importaba qué LED estuviera encendido, solo funcionaba con ese botón para todos los LEDs, y al presionar otro, se interpretaba como un error. Este obstáculo nos llevó a revisar detalladamente el código y realizar pruebas para identificar y corregir el problema.

También queríamos agregar que notamos que el nivel subía demasiado rápido y no se podía disfrutar el juego como es debido así que también lo ajustamos para tener progresión más disfrutable.

Al final luego todos estos desafíos se pudo lograr el correcto funcionamiento del juego para así lograr con el resultado esperado para este miniproyecto 1.

-Actividad 2.1

Esquemático del circuito:



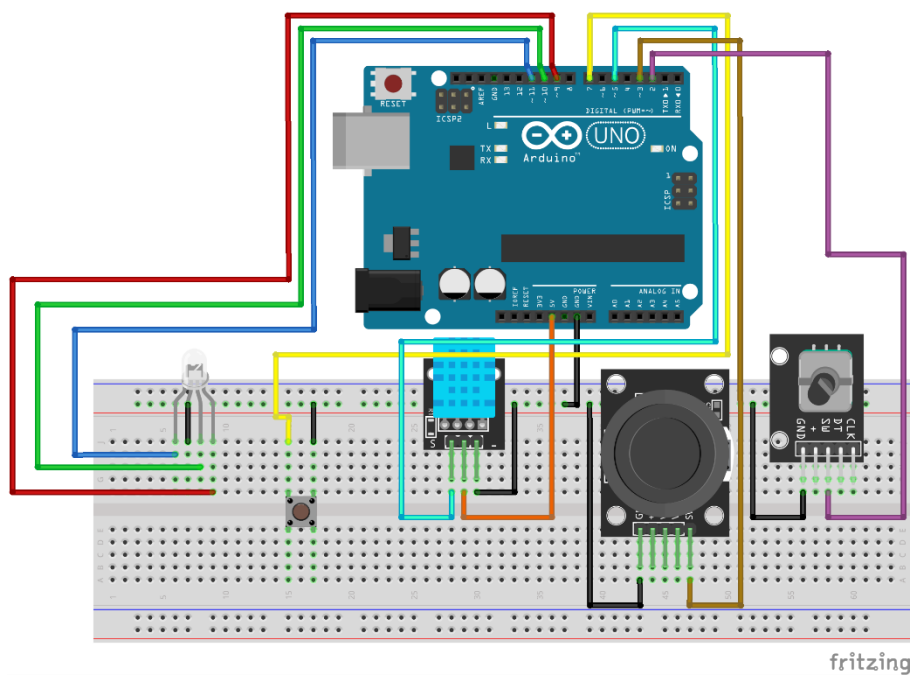
Comentarios:

El código proporcionado para la actividad 2.1 combina la funcionalidad de controlar un sistema de LEDs RGB y un botón desde un Arduino mediante comunicación serial, con la simulación del "Juego de la Vida" de Conway en Python utilizando la biblioteca ``numpy``, ``matplotlib`` y ``serial``. En el código para Arduino, se establecen los pines para los LEDs y el botón, y se configura la comunicación serial. En el bucle principal, se espera la recepción de datos desde Python, representando el estado de las células en el juego, y se ajustan los estados de los LEDs en consecuencia. Además, se verifica si se ha presionado el botón para reiniciar el juego. Por otro lado, en el código de Python, se configura un puerto serial para comunicarse con el Arduino, se implementa la lógica del "Juego de la Vida", y se crea una animación para visualizar la simulación. Se utilizan temporizadores para controlar la velocidad de la animación y enviar datos al Arduino sobre la cantidad de células vivas en la simulación. Si se detecta un mensaje de reinicio desde el Arduino, se reinicia la simulación en Python. Esta integración permite interactuar con la simulación del "Juego de la Vida" a través de la interfaz física del Arduino.

En esta actividad algunos de los problemas que tuvimos a la hora de realizar el correcto funcionamiento del código fue la complejidad del problema que nos superó en muchos momentos a la hora de tratar de hacerlo funcionar, tuvimos algunos problemas con el Arduino dado que era la primera vez usando este microcontrolador y el lenguaje c, el mayor error que tuvimos en este item fue que como no habíamos usado mucho el Arduino estuvimos probando códigos sin cargarlos como tal al arduino, es decir, solo verificándolos, lo que nos atrasó mucho en este punto.

-Actividad 2.2

Esquemático del circuito:



Comentarios:

En el código para arduino, se establecen los pines para los LEDs, el botón, el sensor DHT11 y el retary encode, y se configura la comunicación serial. En el bucle principal, se leen la temperatura del sensor DHT11 y el retary encode KY-040, y se envían al programa en Python

a través de la comunicación serial. Además, se verifica si se ha presionado el botón para reiniciar el juego y se ajustan los estados de los LEDs según los datos recibidos de Python.

En el código de Python, se configura un puerto serial para comunicarse con el Arduino, se implementa la lógica del juego, y se crea una animación para visualizar la simulación. Se utilizan temporizadores para controlar la velocidad de la animación y enviar datos al Arduino sobre la cantidad de células vivas en la simulación. Se han añadido botones para realizar acciones específicas en la simulación (como nuke y heal). Si se detecta un mensaje de reinicio desde el Arduino, se reinicia la simulación en Python. Además, se actualiza la vida de las células de acuerdo con la temperatura recibida del Arduino y el LED muestra en colores los rangos de población.

Esta integración permite interactuar con la simulación del juego y recibir datos ambientales del Arduino, proporcionando una experiencia interactiva y visualmente atractiva.

Esta fue sin duda la actividad más desafiante de la tarea, tuvimos muchos problemas a la hora de manejar la comunicación entre el código de Arduino y el de Python, definir la bomba nuke fue fácil, sin embargo, la bomba de heal presentó un reto, estuvimos mucho tiempo también intentando ejecutar el audio de derrota sin que se crashee el código, también fue complicado probar las condiciones de temperatura pues no disponemos de una forma efectiva de variar los valores del sensor, es por eso que enviamos 2 videos con diferentes condiciones de temperatura, uno en el que la temperatura está en un rango estable y otro donde está por debajo del umbral de frío.