**Control de una matriz RGB por sonido**

Andrea Quiñones

Facultad de ingeniería electrónica y en telecomunicaciones de la universidad del Cauca, correo electrónico: @unicauca.edu.co - Código:

Juan David Osorio Navia

Facultad de ingeniería electrónica y en telecomunicaciones de la universidad del Cauca, correo electrónico: judaosorio@unicauca.edu.co - Código: 100619011061

David Santiago Pismag Imbachi

Facultad de ingeniería electrónica y en telecomunicaciones de la universidad del Cauca, Correo electrónico: dpismag@unicauca.edu.co - Código:100619011050

1. **INTRODUCCIÓN**

Como proyecto para el laboratorio II de electrónica se plantea el control de tiras de led RGB mediante el microcontrolador Arduino Mega 2560, siendo posible para el usuario definir una lista de palabras y que estas sean proyectadas en la tira led. También podrá controlar la tira led median comandos sonoros como aplausos o chasquidos.

1. **OBJETIVO PRINCIPAL**

Hacer uso de una matriz de 12x60 leds RGB, la cual será controlada por el microcontrolador Arduino Mega 2560, sensible a comandos sonoros para su manejo.

1. **CRONOGRAMA**

Semana uno, Definir los requisitos que deberá de cumplir el dispositivo con el profesor encargado.

Semana dos, Definir y dividir la estructura del proyecto.

Semana tres, Comenzar la escritura del código correspondiente al control de la tira de led RGB.

Semana cuatro, Diseñar el módulo de potencia para la alimentación de la matriz RGB.

Semana cinco, Comenzar la estructura del código correspondiente a la respuesta del sensor de sonido, y como esta respuesta afecta a los leds RGB.

Semana seis, Realizar pruebas del código realizado en las semanas 2 y 3 en hardware y en la herramienta de simulación.

Semana siete, Realizar una depuración al código en base a los resultados de la semana anterior.

Semana ocho, Comenzar a estructurar el código correspondiente al módulo RTC.

Semana nueve, Realizar la interfaz gráfica para el control de los mensajes a imprimir.

Semana diez, Realizar una depuración al código en base a los resultados de la semana anterior.

Semana once, Realizar el montaje completo a modo de prototipo con todos los elementos hardware requeridos.

Semana doce, En caso de errores en el montaje, se realizarán las debidas correcciones.

Semana trece, Realizar el montaje final del sistema y realizar depuración.

Semana catorce, Presentar el dispositivo junto con el informe completo del proceso.

1. **DESARROLLO**

***Semana uno***

Para esta semana se propuso definir los requisitos que debe de cumplir el sistema final:

Inicialmente se nos plantea el uso de 12 tiras RGB de 59 leds cada una, las cuales deben ser controladas mediante aplausos o chasquidos, para lo cual se plantean las instrucciones para su manejo.

* Un aplauso, encender la matriz y/o cambiar el estado actual en el cual se encuentra.
* Dos aplausos, cambiar a modo reloj, mostrara la hora con ayuda del módulo RTC.
* Tres aplausos, cambiar los colores con la que se muestra la información (se utilizan colores predeterminados).
* Cuatro aplausos, refrescar los mensajes en memoria y apagar la matriz.

Durante el funcionamiento de la impresión, se ha planteado la posibilidad de cambiar los mensajes que se muestren por medio de una conexión bluetooth a una interfaz de computador.

Para realizar el control de la matriz se hará el desarrollo en la plataforma Arduino IDLE y en Studio Visual Code se realizará el diseño de la interfaz.

***Semana dos***

Nos disponemos a reunirnos para charlar como nos vamos a dividir las tareas a realizar como completar el proyecto, el resultado fue el siguiente:

* Imagen que contiene tabla, cama

  Descripción generada automáticamenteJuan David: Encargado del control de los leds, teniendo que crear un código que reciba como entrada una palabra (un string) e imprima dicha palabra en la tira led.
* David Santiago: Encargado de diseñar el dispositivo detector de aplausos y el uso del módulo RTC.
* Andrea: Encargada de diseñar la interfaz de usuario que se conecte al Arduino mediante bluetooth.

***Semana tres***

Para esta semana se propone comenzar la escritura del código correspondiente al control de la tira de leds RGB:

Haciendo uso del lenguaje Arduino se creó un alfabeto en base a matrices, las cuales definen en un tamaño de 8x5 leds cualquier carácter alfanumérico, excluyendo caracteres especiales.

Una botella en la mano

Descripción generada automáticamente con confianza bajaCon esto, se incluyó la posibilidad de presentar mensajes en la matriz de leds de hasta 28 caracteres (Incluidos los espacios), y se añadió el formato que tendrá el reloj que hará uso del módulo RTC.   
La matriz se encuentra compuesta por 12 tiras de leds RGB de 59 pixeles cada una, todas conectadas independientemente a un pin PWM del arduino MEGA, al realizar pruebas, se descubrio que existían cortos en la baquelita, ocasionando problemas al momento de hacer uso de los leds:

Imagen que contiene tabla, circuito, computadora, luz

Descripción generada automáticamentePor este motivo, se decidió desoldar el módulo para así realizar nuevas conexiones a modo de prototipo y poder poner a prueba el codigo realizado.

Se puso a prueba la primera version del codigo, el cual hace uso de la librería Adafruit\_NeoPixel, identificando los siguientes problemas: las conexiones iniciales estaban dañadas y hubo que hacer un reemplazo; la capacidad de procesamiento de arduino es insuficiente para imprimir las letras y por un genera que el mensaje no se vea fluido al desplazarse, adicionalmente, la falta de una potencia adecuada genera que los colores mostrados en la matriz no correspondan con los que fueron programados.

***Semana cuatro***

Imagen que contiene tabla, cuarto

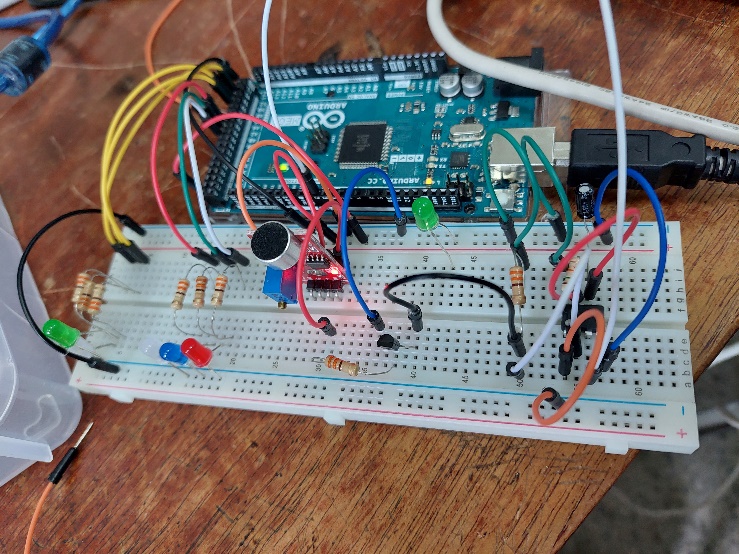
Descripción generada automáticamenteNos disponemos a encontrar una solución para el problema de consumo excesivo de la tira led, dándonos un consumo promedio de dos amperios para un color blanco, después de algunas pruebas, concluimos que lo más adecuado sería utilizar un trasformador de corriente alterna de ciento diez voltios a cinco voltios dos amperios. Se realizaron las pruebas correspondientes y no hubo problema con el funcionamiento.

***Semana cinco***

Se ha comenzado con la experimentación con el módulo de sonido KY-037, usado para la detección de aplausos, puesto que este posee una salida digital activa en alto al ser activado, por lo cual se planteó su uso como una interrupción en el pin digital 18, la cual aumente un contador de aplausos para diferenciar los comandos a ejecutar.

Rápidamente se identificaron 2 problemas al implementar este módulo.

* La salida digital del módulo es insuficiente para activar la interrupción planeada para su funcionamiento.
* El sensor capta, además del aplauso inicial, su eco como entrada, dando como resultado múltiples detecciones para un solo aplauso.

Como solución planteamos un circuito monoestable que se encargue de recibir únicamente el primer aplauso y se mantenga un pulso durante 100ms para evitar recibir el eco, además de ellos fue necesario implementar un circuito negador con ayuda de un transistor debido a que la salida del módulo de sonido es alta y la entrada del monoestable es en bajo.

Una mesa con una computadora

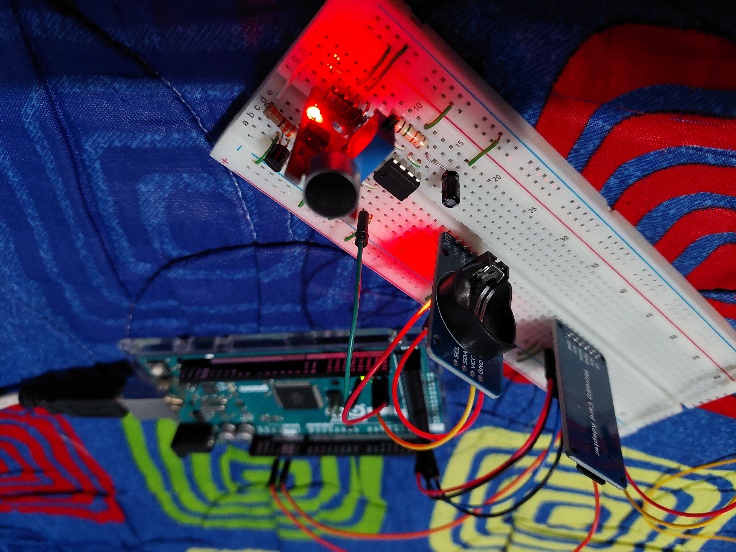
Descripción generada automáticamente con confianza baja***Semana seis***

Empezamos con las pruebas en el laboratorio para cerciorarnos del correcto funcionamiento del dispositivo; el módulo detector de aplausos tuvo el desempeño adecuado detectando correctamente el número de aplausos realizado, al igual que ejecutar los comandos establecidos. Por otra parte, fue posible confirmar que la baja fluidez a la hora de mostrar el mensaje se debía a la incapacidad del Arduino de ejecutar la orden para 12 tiras led en todo momento, razón por la cual se acordó el replanteamiento de la forma de impresión de la matriz.

***Semana siete***

Teniendo en cuenta los problemas encontrados la semana anterior, se procede a hacer una optimización en la impresión de las palabras en la tira LED: Se logro una optimización en el uso de memoria eliminando una matriz de 12x234 usada para almacenar la palabra mensaje que se deseaba imprimir, la cual ocupaba alrededor del 35% de la memoria dinámica del Arduino, evitando los problemas de estabilidad que genera el uso excesivo de esta, permitiendo que la fluidez con la que se mostraba el mensaje mejorara.

Para realizar este cambio tuvieron que adaptarse ciertas funciones encargadas de la impresión del mensaje, las cuales pasaron de leer la matriz mencionada con anterioridad a leer directamente la palabra de una variable, y con ella escribir cada carácter a lo largo de la matriz.

***Semana ocho***

Para el montaje se planteó el uso del DS3231, un módulo RTC que posee una configuración simple en Arduino. Únicamente con la librería RTClib se configura un objeto del tipo RTC, inicialmente se le define el tiempo actual al RTC, entonces será posible extraer esta información posteriormente con ayuda de las funciones propias de la librería, las cuales entregan respectivamente la hora, los minutos y los segundos del RTC, por lo cual el poder mostrar la información se vuelve simple.

Para imprimir la información en la matriz se escogió un formato hh:mm:ss, y se definió un método encargado de realizar el formateo de esta información, incluyendo agregar un 0 a los valores menores que 10 (como, por ejemplo: 09:05:55), una vez teniendo correctamente las cadenas deseadas, estas se unen en una variable, la cual se pasa directamente a la función usada para imprimir un mensaje en la matriz.

***Semana nueve***

Para el diseño e implementación de la interfaz gráfica que controlara algunas funciones del proyecto implementado en Arduino, Control de Luces RGB, se decidió trabajar en lenguaje C haciendo uso del entorno de programación de Visual Studio, el cual proporciona cierta facilidad en la comunicación bluetooth, teniendo en cuenta que el diseño de la interfaz debe incluir la opción de almacenar, buscar y eliminar archivos de texto en una tarjeta de memoria SD.

en primera instancia se procede a programar todo lo referente a la comunicación vía Bluetooth entre el Arduino y la interfaz, agregando al diseño los botones conectar y buscar los cuales son ligados a un cuadro de texto donde se puede observar el puerto correspondiente a la comunicación. A estos botones se les creará un evento que se accionará o tendrá lugar cuando son pulsados, dentro del evento se añaden las líneas de código correspondientes a la habilitación e inhabilitación del puerto de comunicación, además de habilitar el botón escribir texto correspondiente a la función que cumplirá la interfaz, planteando un primer prototipo de esta.

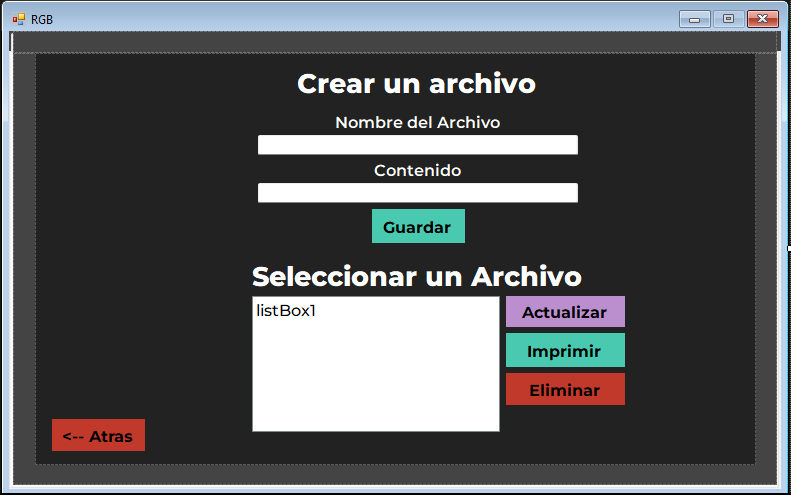
Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza media

***Semana diez***

Se implementa un TabControl en la interfaz para hacer uso de múltiples cuadros o formularios ejecutados en una única ventana, a su vez, se implementaron los eventos correspondientes a los botones Escribir Texto y Atrás, dentro de estos eventos se agregaron las líneas de código correspondientes al movimiento entre ventanas permitiendo habilitar la visualización o no, según corresponda.

Al haber seleccionado el botón Escribir Texto, el usuario se redirigirá a una nueva pestaña donde se agregaron dos contenedores de texto editables: Nombre del Archivo y Contenido, el primero limitado a solo poder recibir 8 caracteres (MaxLeng) debido a que el módulo SD del Arduino solo puede guardar archivos renombrados con este tamaño y el segundo, limitado a solo poder recibir 28 caracteres debido al tamaño de la tira de leds usada; un contenedor seleccionador (ListBox) y cuatro botones con las funciones de: Guardar, Actualizar, Seleccionar y Eliminar.

Teniendo el diseño de la segunda ventana, se procede a crear el evento correspondiente al botón Guardar, el cual se encarga de almacenar el texto digitado por el usuario en variables, verificando que el usuario halla ingresado correctamente valores y se espera a que Arduino confirme la llegada de la información a la aplicación, mostrando un mensaje para indicar el estado del envío de información.

para el caso del botón Eliminar se hace uso de la función remove que borrara el elemento seleccionado de la ListBox, ya que el encargado de borrar el elemento de la SD es el mismo Arduino, así mismo, en este evento se inhabilita el botón eliminar hasta que el usuario vuelva a hacer clic a la ListBox o lista. En otro caso, para el evento del botón Imprimir, después de haberle enviado la información al Arduino, se limpia la selección hecha en la ListBox y se inhabilitan los botones Imprimir y Eliminar hasta que el usuario vuelva a hacer clic a la ListBox o lista.

***Semana once***

Adicionalmente se necesita el uso de un lector de tarjetas SD, para lo cual se usan las librerías SPI y SD, adicionalmente la librería particular “Separador”. Las librerías SPI y SD son usadas para poder inicializar el módulo de la tarjeta y poder establecer la comunicación con el Arduino o el módulo bluetooth, los cuales se encargan de leer un archivo específico que se encuentre almacenado en la SD.

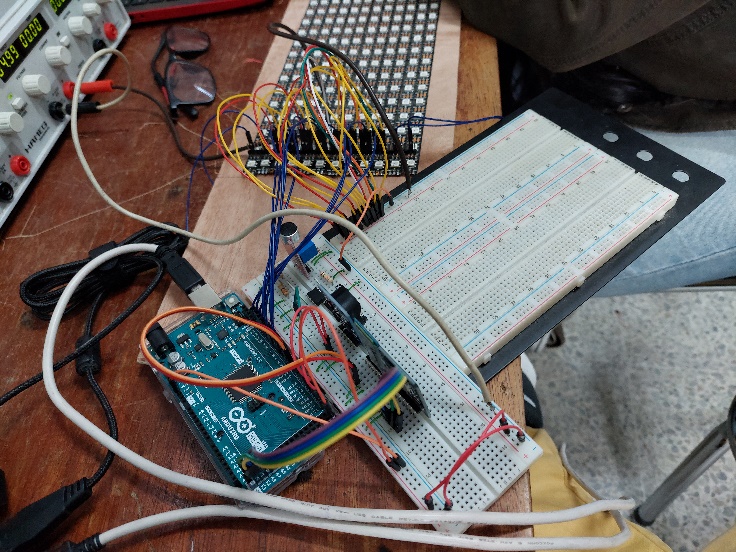
Para poder leer los mensajes al interior de la SD se planteo un archivo que almacenara el nombre de los archivos que contienen el mensaje, por lo cual, para poder leer estos fácilmente se implementó la librería Separador.h, la cual permite separar secciones de una cadena basándose en un carácter usado como separador, y almacenar estos nombres en un vector para poder realizar el llamado del archivo correspondiente, los cuales contienen el mensaje que se quiere mostrar en la matriz, los cuales también se almacenan en un vector en caso de perder la comunicación con el módulo SD.

***Semana doce***

Ya con todos los módulos planteados, es momento de unificar todos estos en un único código que trabaje en conjunto, por lo cual se realizó una implementación de estos por partes, inicialmente se unifico el control de la matriz y el sensor de sonido, para posteriormente adicionar el módulo RTC, y finalmente el lector de tarjeta SD, el modulo Bluetooth no se implemento actualmente puesto que se presentaron diversos errores a la hora de enviar y recibir los mensajes, sin embargo, a excepción de este módulo, los demás trabajaron en conjunto de manera correcta, esto se comprobó con ayuda del monitor serie conectando el Arduino al computador, aplazando el montaje físico una semana, esperando que la respuesta del sistema físico para ese momento fuera la adecuada.

***Semana trece***

Esta semana se realizó el montaje completo del hardware correspondiente a la matriz RGB en conjunto con todos los módulos a excepción del bluetooth, y se realizaron diferentes pruebas para determinar el funcionamiento del sistema.

Al momento de subir el programa al Arduino se presento un problema en el sensor de sonido, motivo por el cual se tuvo que revisar el correcto funcionamiento del módulo y del monoestable, encontrando que el problema fue generado por el capacitor, tras realizar el cambio del componente, nuevamente el sistema se encontró operativo, por lo cual se concluyó que el montaje hasta el momento era operativo, y fue presentado como un primer prototipo del sistema final.

***Semana catorce***

Realizar el montaje final del sistema y realizar depuración.

1. **METODOLOGÍA EXPERIMENTAL**

Metodología experimental

1. **ANÁLISIS DE RESULTADOS**

Análisis de resultados

1. **CONCLUSIONES**

Conclusiones

**Referencias**

Referencias