



Título:	CUSTOM LIGHTS
----------------	----------------------

Ciclo Lectivo 2021	Curso N°	R2003	Grupo N°	1
---------------------------	-----------------	--------------	-----------------	----------

Integrantes	Apellido Nombres	Legajo	Calificación individual	Fecha
	Almaraz, Nicolás	1756333		
	Glorioso, Guido	1756916		
	Jimenez Copati, Juan	1756898		
	Zuran, Tomás	1756928		

Calificación grupal:		Fecha: 20/12/2021
-----------------------------	--	--------------------------

Profesor:	Nahuel Gonzalez
Auxiliar/es Docente:	Federico Búa, Nicolás Campos y Lisandro Sugezki

Observaciones primera entrega	
Observaciones segunda entrega	



Índices

Presentación de la idea fuerza	3
Introducción	3
Objetivos.....	3
Características:	4
Diagramas en bloques.....	4
Descripción detallada	5
Bloques y especificaciones.....	5
Descripción del HW utilizado	5
Circuitos.....	6
Máquinas de estado de aplicación	8
Problemas encontrados.....	15
Velocidad de transmisión LPC-Tira Led.....	15
Protocolo UDP	16
Lectura de sonido	16
Lectura de pantalla	17
Problemas de hardware.....	17
Beneficios encontrados a lo largo del TPO.....	17
Valoración del TPO	18
Observaciones y mejoras.....	18
Conclusiones.....	19



Presentación de la idea fuerza

Introducción

En este proyecto se propone incorporar luces ambientales a la pantalla de una computadora. El objetivo de esto es expandir de forma difuminada la imagen de la pantalla para lograr una experiencia de mayor inmersión. Para poder configurar diferentes parámetros y modos se dispondrá una aplicación.



Ejemplo de lo que se busca

Objetivos

Lograr una mejor experiencia a la hora de usar la PC. Principalmente que destaque a la hora de jugar o ver películas.

Objetivos técnicos:

1. Poder controlar las luces tanto en forma remota (wifi) como también vía comunicación serie.
2. Implementar varios modos que puedan ser configurables por el usuario de forma sencilla.
3. Implementar una interfaz usuario que permite configurar las luces, de forma que no sea necesario seguir una única configuración. (se pueden colocar tiras de mayor o menor tamaño y en formas distintas).



4. Llevar a cabo la configuración wifi lo más sencilla y autónoma posible, buscando que su utilización sea cómoda para el usuario final.
5. Controlar las luces mediante el uso del microcontrolador proporcionado por la cátedra
6. Procesar tanto imagen como sonido de la PC.

Características:

Para lograr este proyecto se utilizarán tiras de LEDs RGB programables. Es decir, se podrá programar cada led individualmente por el protocolo SPI.

Por otro lado, se podrán instalar modularmente. Esto consiste en desarrollar módulos con una cantidad de LEDs determinada y determinado tamaño. De esta manera, cada usuario podrá adaptar el proyecto al tamaño de su pantalla conectándolos en serie. En función del tamaño de pantalla que tenga va a requerir una cantidad diferente de módulos.

En cuanto a la interfaz de usuario, se realizará un programa en QT donde se pueda configurar diferentes modos y personalizaciones (por ejemplo: fijar un color; fijar brillo; que varíe en función de la pantalla; etc.).

Para la comunicación entre el gabinete donde estará el microcontrolador con su respectiva electrónica y la PC se optará por utilizar un módulo wifi que permita conectar la computadora vía internet. En cuanto al módulo WIFI se utilizará un ESP01.

Luego este módulo se conectará vía UART al microcontrolador, de modo que se usará el módulo únicamente para la comunicación con la PC.

Diagramas en bloques

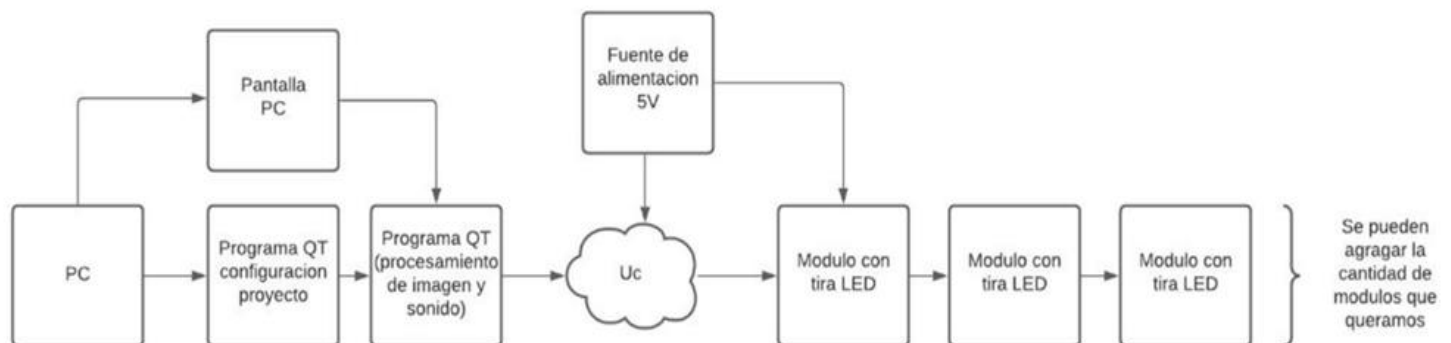


Diagrama hecho con LucidChard



Descripción detallada

Bloques y especificaciones

El proyecto consta de tres bloques principales:

PC/Monitor:

En este bloque se tiene todo lo relacionado con el procesamiento de la información, interfaz usuario, tramas de comunicación etc. En este bloque también llamado “aplicación” se busca tener todo lo que a modos de control refiere, es decir que se independiza de lo que se pueda tener en los siguientes bloques permitiendo así que no implique un gran esfuerzo un cambio de hardware.

Comunicación Wifi:

En este bloque se concentra todo aquello que corresponda a la comunicación wifi, es decir la conexión a la red, la recepción de tramas, y su posterior envío por puerto serie. Se implementa además una interfaz usuario que permite hacer más sencilla la conexión a la red.

Control de luces:

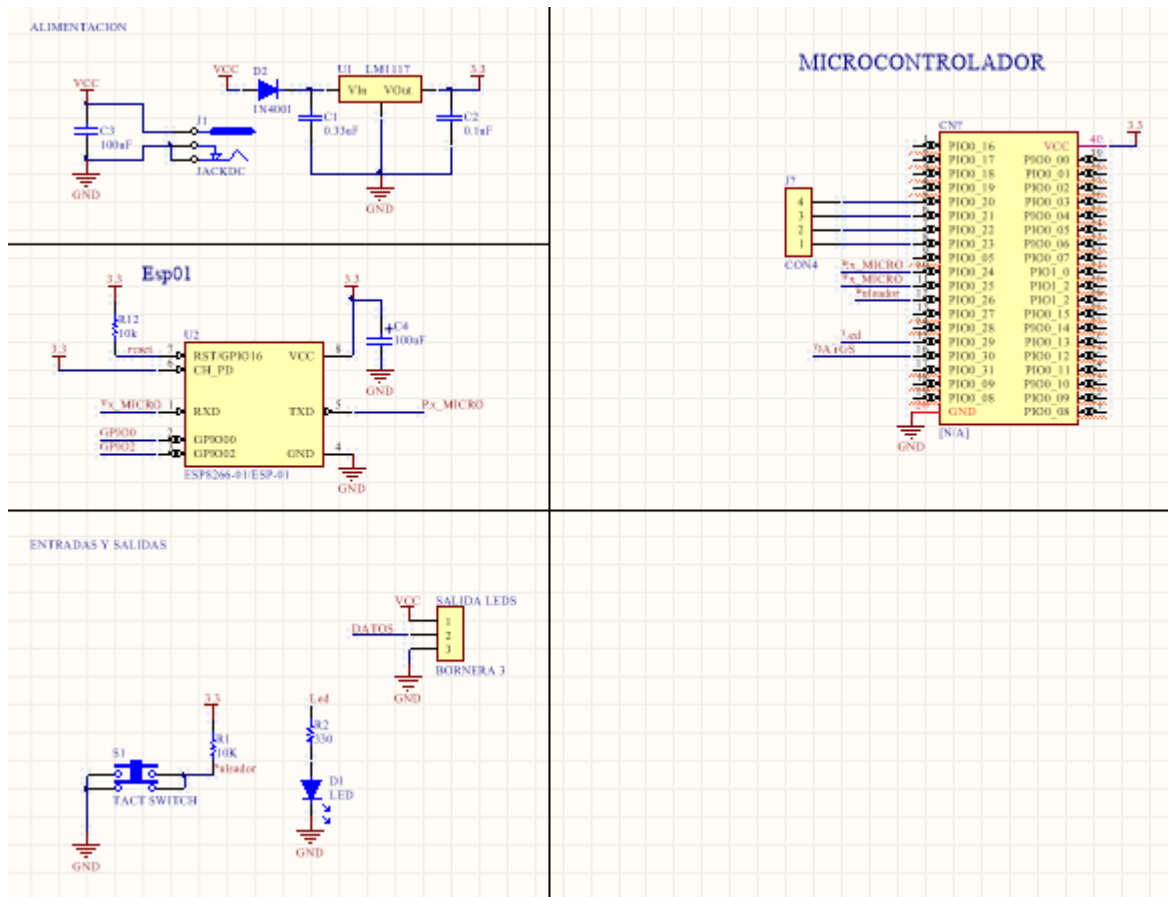
Finalmente, en esta etapa se incluye todo lo que concierne al manejo de las luces, es decir adaptar la trama que se procesó en el primer bloque a algo que las tiras led puedan comprender. En esta etapa buscamos realizar el menor procesamiento posible de modo que únicamente reciba tramas y encienda los leds.

Externo a estos bloques se tiene la alimentación, sin embargo no se realiza ningún tipo de control sobre ella porque no se considera en ninguna de las etapas.

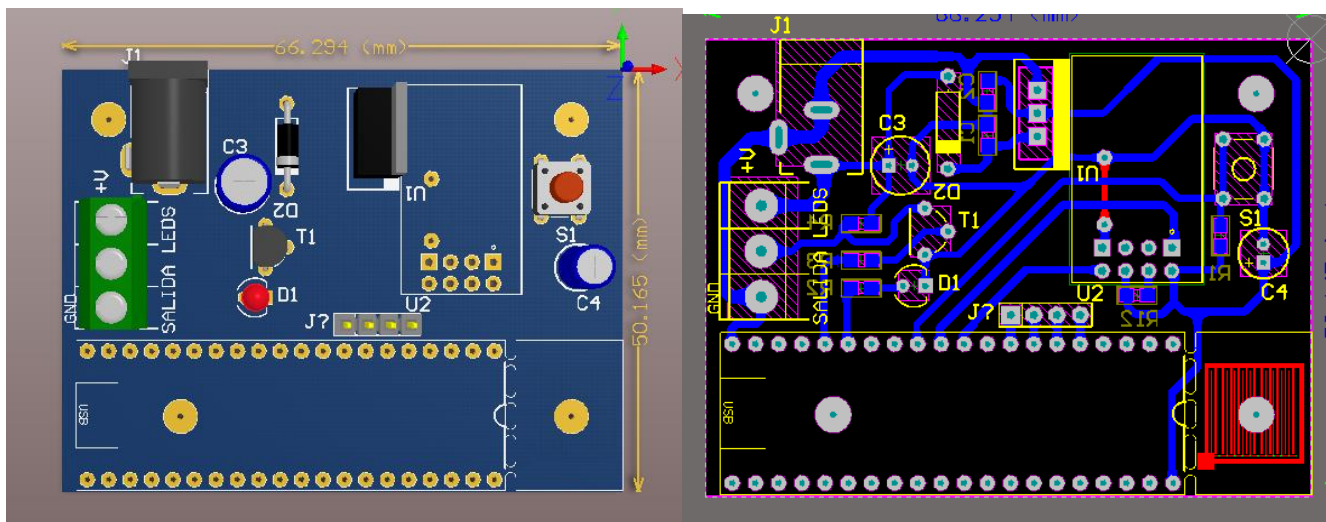
Descripción del HW utilizado

Para llevar a cabo este proyecto es necesario el siguiente hardware:

- Placa esp01 (modulo WIFI)
- microcontrolador LPC845
- Luces led WS2812 (la cantidad depende de cada usuario)
- Shield para acoplar ambas placas (contiene tanto interfaces como regulaciones de tensión)
- Fuente de alimentación 5V (depende de la cantidad de leds)
- conectores de luces led.



Para el shield se diseñó en ALTIUM un pcb y luego se hizo la placa.





Links hojas de datos

- <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf> (Hoja de datos de tira led)
- <https://www.microchip.ua/wireless/esp01.pdf> (Hoja de datos módulo wifi ESP01)

Máquinas de estado de aplicación

Máquina de estados del LPC845:

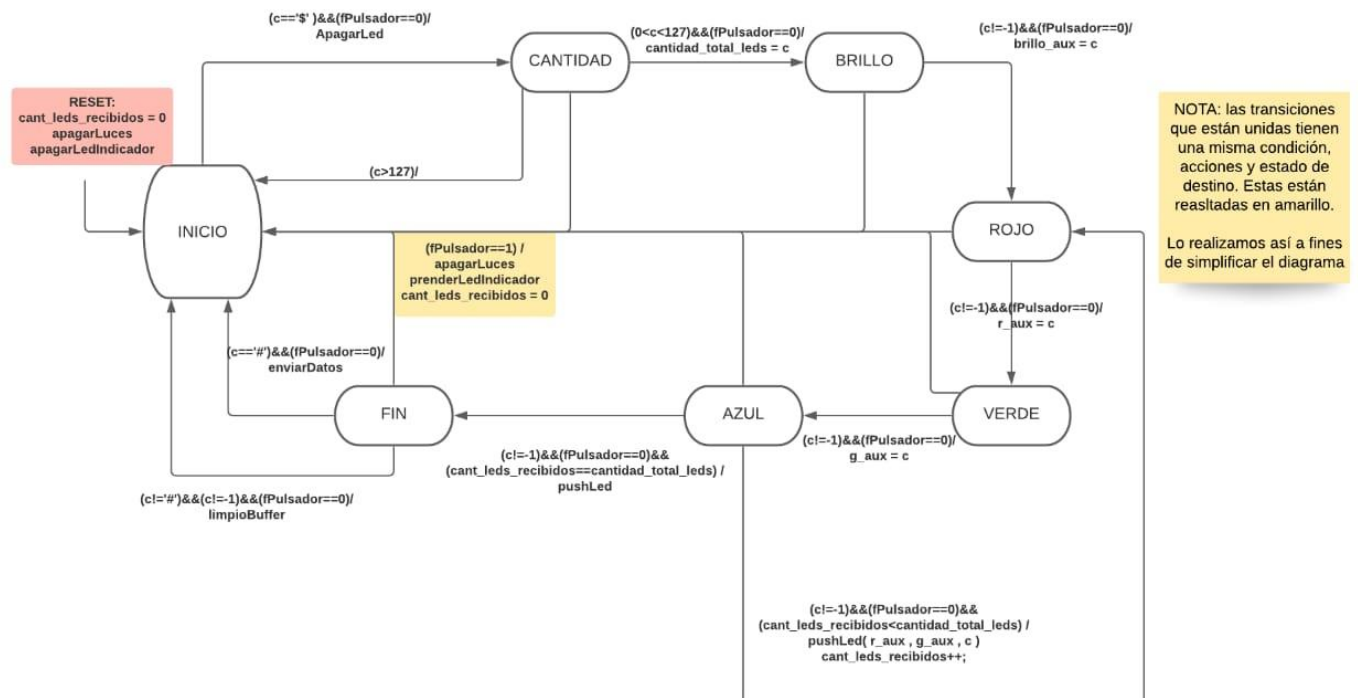


Diagrama hecho con LucidChart

Es la máquina de estados que interpreta la trama serie que le llega desde el ESP01, básicamente toma los datos que le llegan desde la trama como la cantidad de leds, el brillo general y los colores, y los guarda en memoria para su posterior uso.



Máquina de estados del ESP01:

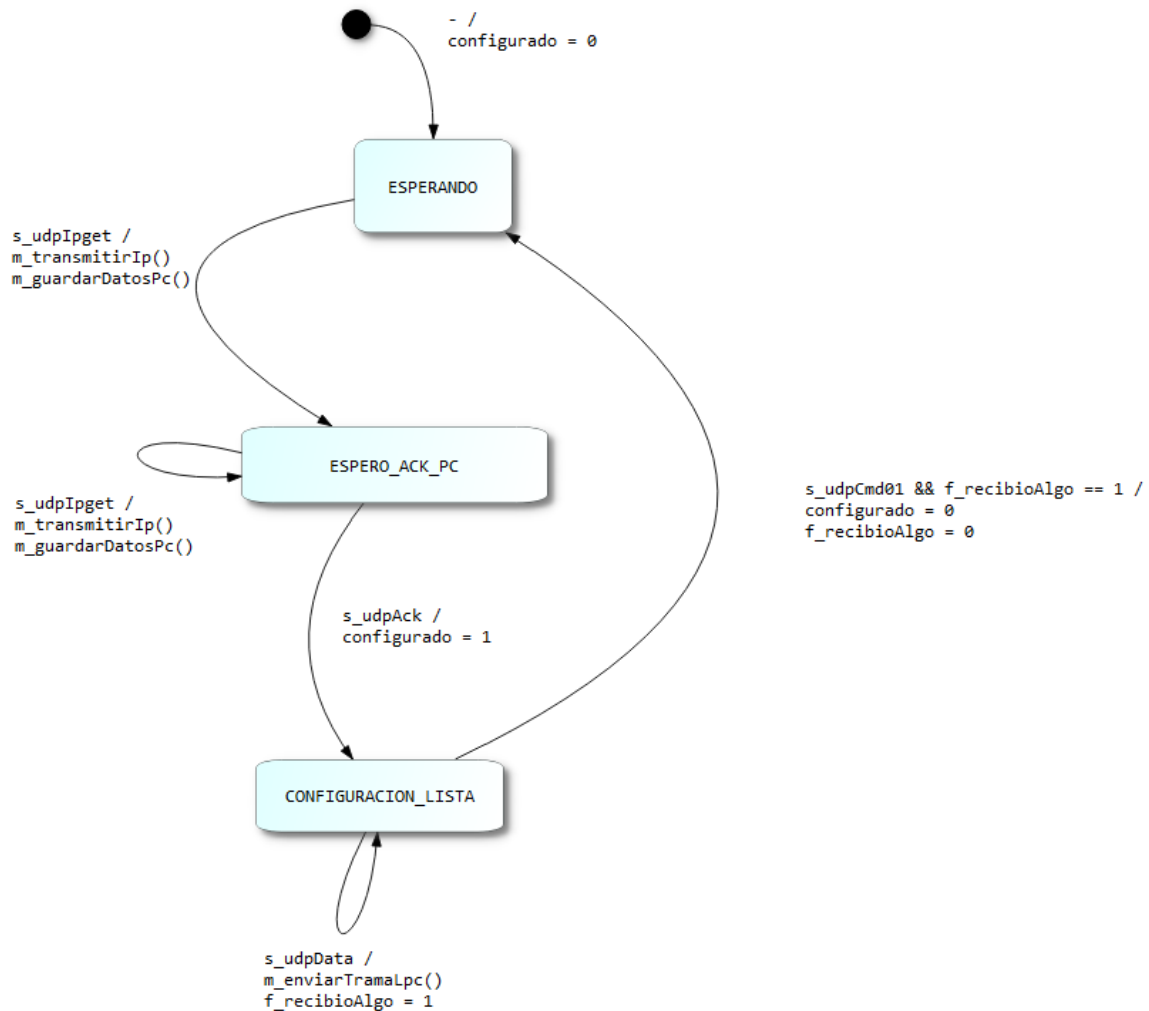


Diagrama hecho con uModelFactory

Es la máquina que usamos para la configuración inicial del dispositivo, donde toma la IP de la computadora y la guarda en memoria para su posterior uso. Permite reiniciar la máquina al llegar un comando 01 (Cmd01) por si se reinicia la PC. Como la transmisión es UDP y no serie entre este dispositivo y la app en Qt, no hay máquina de recepción ya que



se trabaja más como un evento: al llegar un mensaje por el puerto UDP 10800, se interpreta automáticamente ya que todo el mensaje llega en un mismo paquete.

Máquina de estados de la app en Qt:

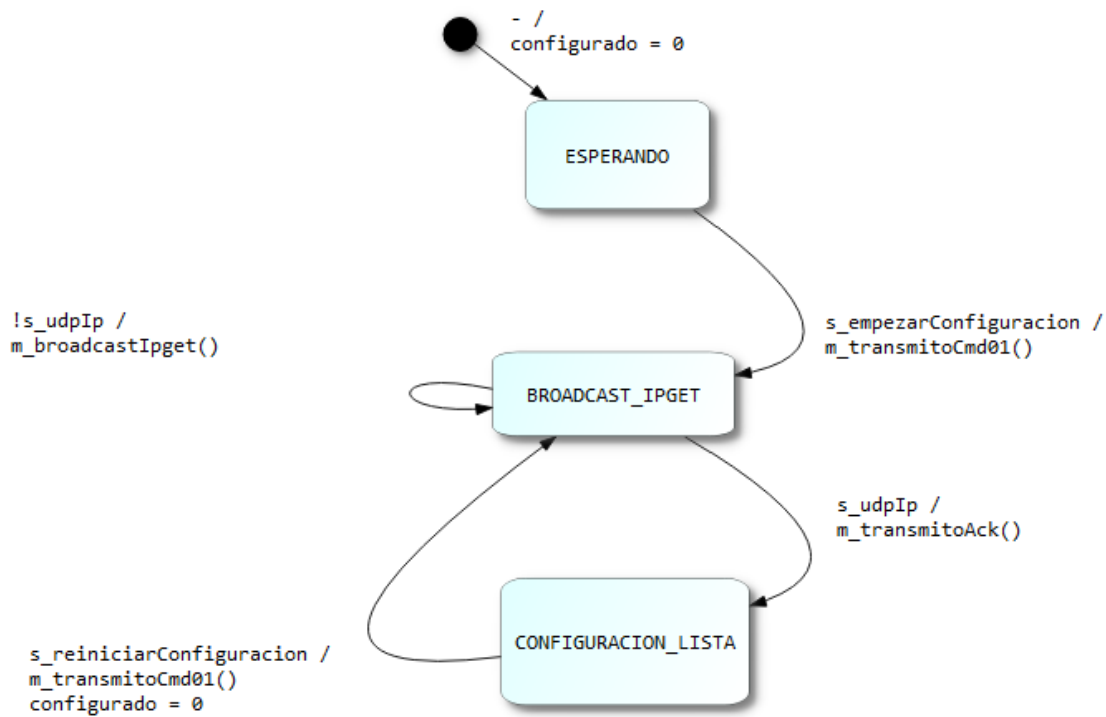


Diagrama hecho con uModelFactory

Esta es la máquina encargada de encontrar en la red local un dispositivo de *Custom Lights*. Cuando empieza la configuración se hace un broadcast con un comando para que el ESP01 responda con su IP. Una vez que se recibe el acknowledge del ESP, ambos equipos saben las IPs privadas de cada uno, haciendo posible una comunicación sin el broadcast. No se implementó por falta de tiempo, pero este sistema podría soportar más de un dispositivo en la red. Se transmite un Cmd01 al principio de la configuración (en un



broadcast) para asegurarse que la máquina de configuración del ESP se encuentra reiniciada (en un caso en el que se cierre la app de Qt pero no se reinicie el dispositivo por ejemplo). No se aclaró, pero las IPs privadas se sacan de los headers del mensaje UDP.

Comandos PC->ESP	Payload UDP	Explicación
Cmd01	<cmd>01</cmd>	Es un comando que resetea la máquina de estados de configuración del ESP, por si se reinicia la app de Qt pero no el dispositivo. Cuando se llama al principio de la máquina de estados es un broadcast.
broadcastIpget	<ipget></ipget>	Se hace un broadcast a todo el puerto UDP 10800. El ESP responde con el comando IP.
Ack XX	<ack>XX</ack>	Comando de acknowledge.
Datos	<data>PAYLOAD</data>	Comando donde la PC manda la información necesaria para prender los leds. No tiene un largo definido, pero en el principio de la trama se manda la cantidad de leds que indica indirectamente cuán larga tiene que ser esta.



Comandos ESP->PC	Payload UDP	Explicación
Ack XX	<ack>XX</ack>	Comando de acknowledge. Además de usarse en la máquina de estados, el ESP responde con un Ack 00 cada vez que le llega una trama de datos correcta, y un Ack 01 cuando llega una trama de datos incorrecta o irreconocible.
Transmitir IP	<ip>XXX.XXX.XXX.XXX</ip>	El ESP envía su ip privada.

Diagrama de clases QT

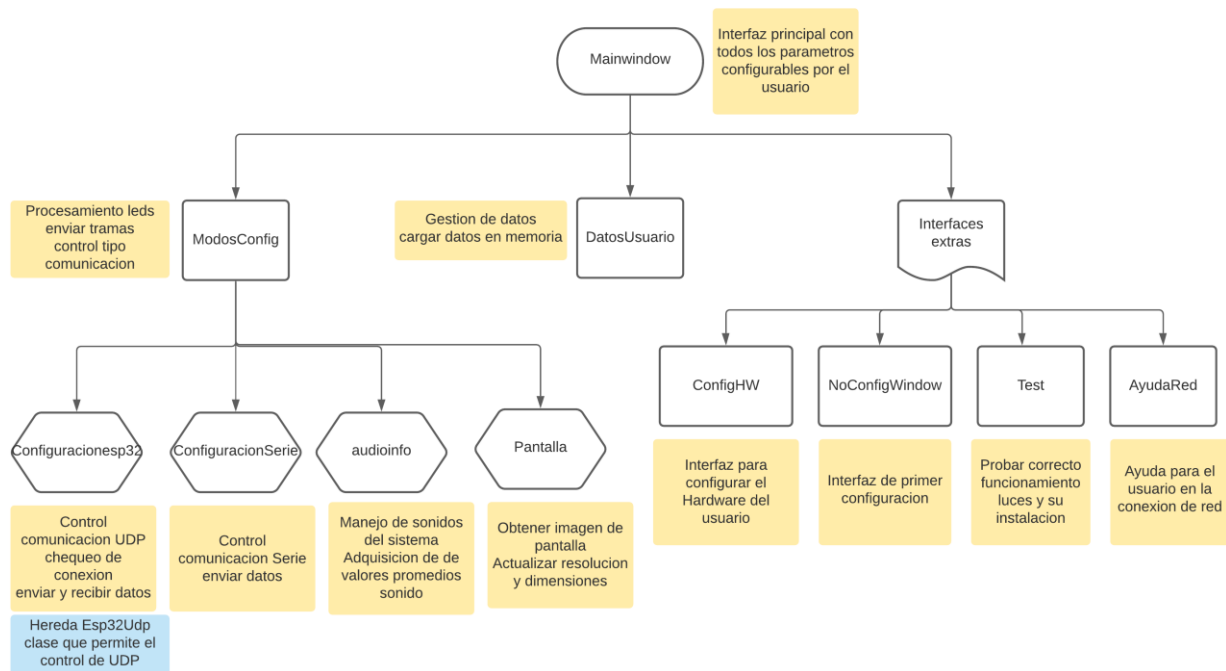
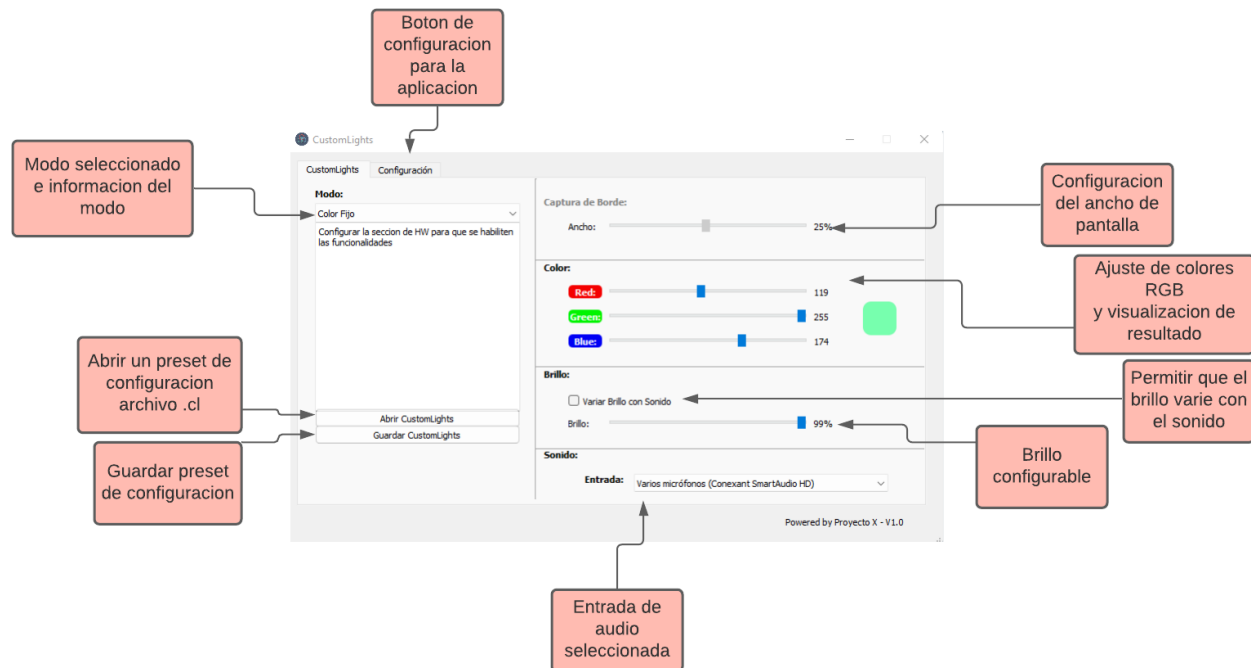


Diagrama hecho con LucidChart

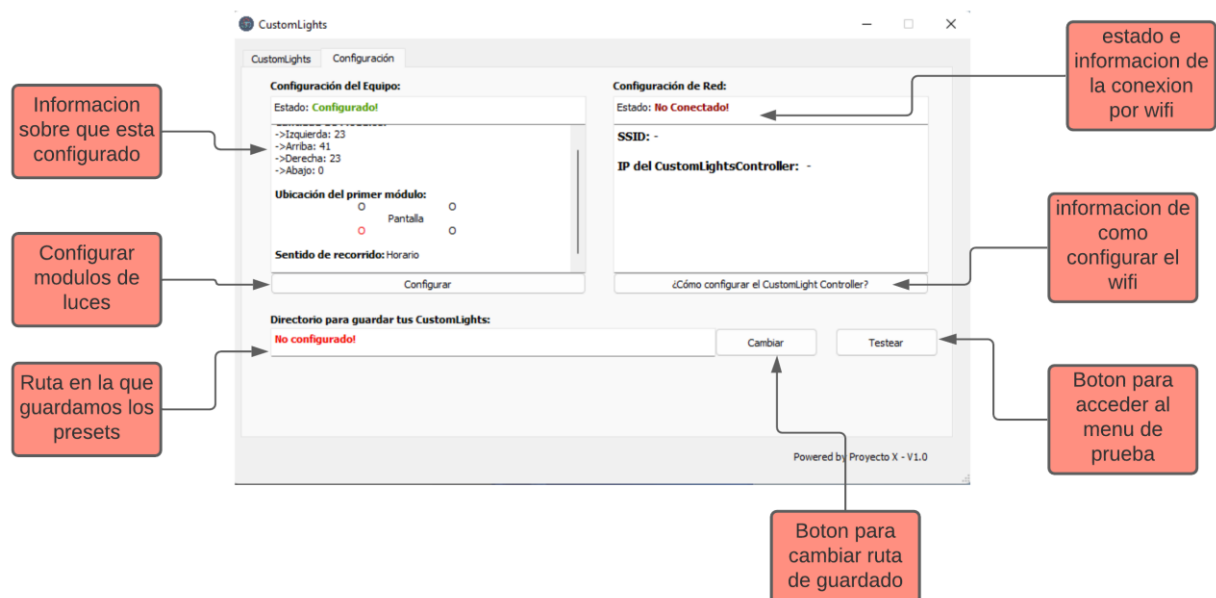


Interfaces Graficas QT:

Interfaz principal

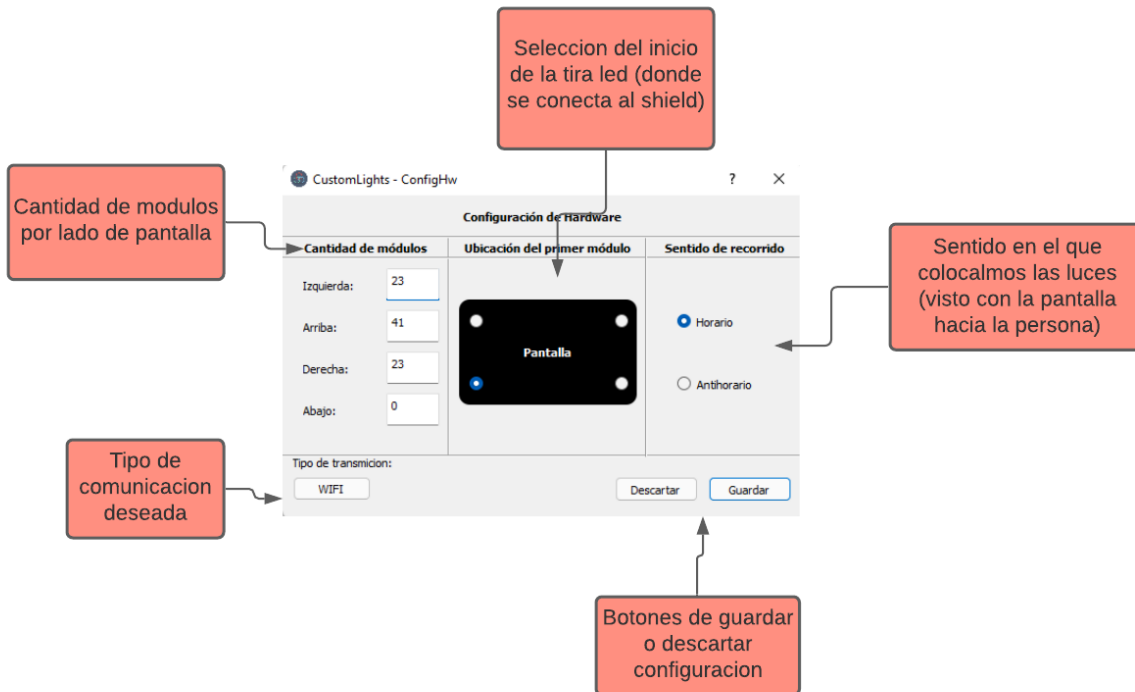


Pestaña configuración:

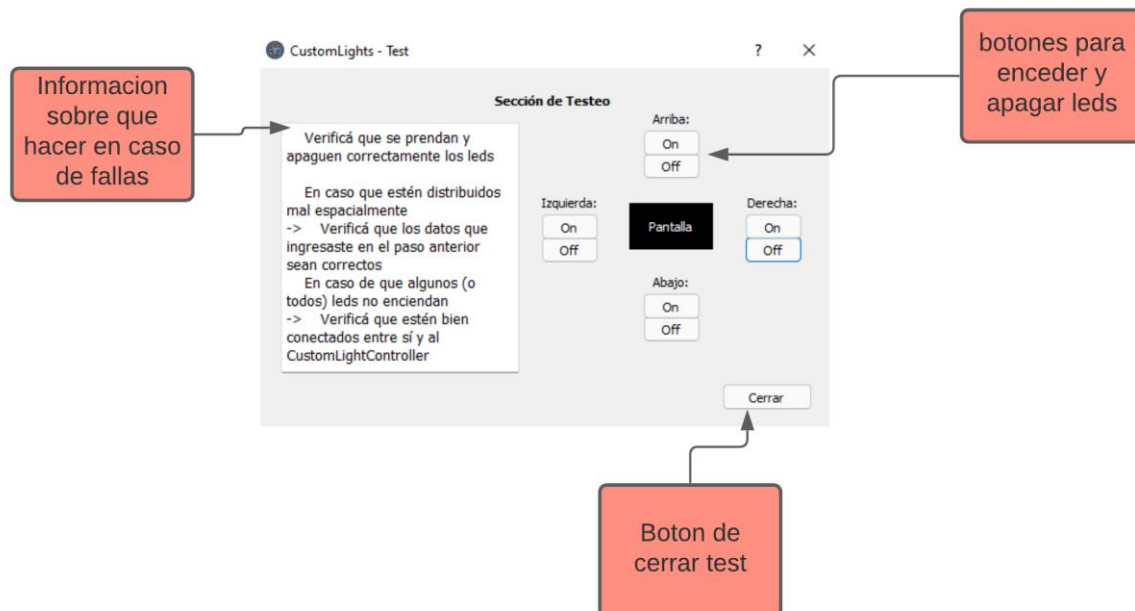




Menú de configuración

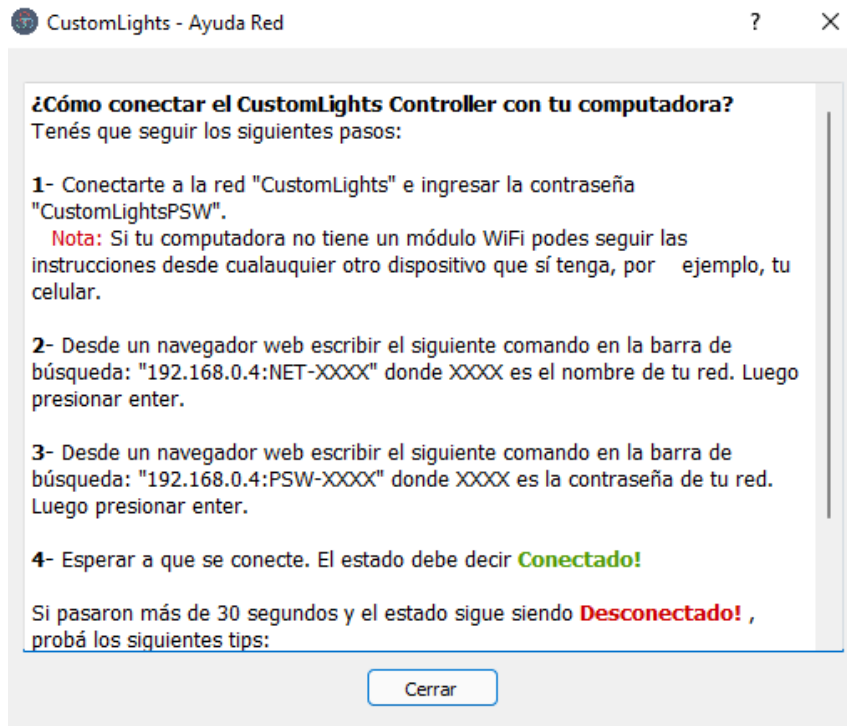


Pestaña de test





Ayuda de red



Problemas encontrados

Velocidad de transmisión LPC-Tira Led

Uno de los problemas encontrados fue que el protocolo de transmisión de la tira maneja tiempos muy pequeños (orden de los 500ns). Para poner en contexto. El protocolo era el siguiente:

Data transfer time($T_H + T_L = 1.25\mu s \pm 600ns$)

T0H	0 code ,high voltage time	0.4us	$\pm 150ns$
T1H	1 code ,high voltage time	0.8us	$\pm 150ns$
T0L	0 code , low voltage time	0.85us	$\pm 150ns$
T1L	1 code ,low voltage time	0.45us	$\pm 150ns$
RES	low voltage time	Above 50 μs	



Es decir que resultaba inviable utilizar el systick para poder conseguir estos tiempos. Es por ello que para poder solucionar esto se recurre al periférico SPI. Utilizando la salida MOSI en un baudrate adecuado se forman dos tramas: 11111000 o 11100000 para representar un “1” o “0” respectivamente.

Protocolo UDP

En la comunicación UDP se presentó como problema que no se podían transmitir caracteres superiores a 127. Tras intentar varias formas de solucionarlo (se intentó trabajar con caracteres del tipo signed char pero no se consiguió solucionarse) se tuvo que adaptar la trama de forma tal que se puedan transmitir todos los datos necesarios sin necesidad de enviar valores superiores a 127.

Lectura de sonido

En la lectura del sonido se presentó como principal problema la variación de los rangos medibles. Esto es que, dependiendo del programa, dispositivo con el que estemos tomando el audio, sonido del sistema operativo y demás factores, es prácticamente imposible conseguir los valores máximos y mínimos de volumen con los que se trabajarán.

Esto presenta como problema que dependiendo la situación el audio puede saturar constantemente el vúmetro de las luces o no alcanzar nunca niveles altos del mismo.

Para solucionar esto lo que se realizó es un promedio de los valores medidos del sonido, que se iba moviendo en el tiempo (media móvil), de esta forma tomando las últimas mediciones se puede conocer el nivel aproximado de audio que se está manejando, y así poder adaptar los niveles máximos y mínimos del vúmetro según la circunstancia.

Otro problema del sonido es que la única forma que se encontró de tomar el audio de la computadora directamente con el programa es mediante un programa externo “Audio Cable” que permite hacer un loopback de todo lo que se reproduce en nuestra pc. Además de esto es necesario disponer de un mezclador de audio que nos permita tener más de una salida de audio a la vez.



Lectura de pantalla

En la lectura de pantalla se presentó como inconveniente la falta de información que había, siendo bastante complicada su implementación. Una vez se pudo implementar se tuvieron problemas como el promedio de los colores para que sea lo más cercano a lo que se aprecia en pantalla.

Otro inconveniente encontrado fue que ciertas aplicaciones (principalmente aquellas que son de streaming como netflix) no permiten grabar la pantalla, por lo que en situaciones como estas el programa “ve” una pantalla negra y no responde a lo que se reproduce realmente en pantalla.

Problemas de hardware

Uno de los problemas fue que a la hora de diseñar el shield para las placas no se tuvo en cuenta la velocidad con la que se transmitían los datos, esta interfaz en un principio era necesaria ya que el fabricante de las tiras led indica que es necesario un nivel de tensión de 5v para poder controlarlas. Al intentar prender las luces con dicha interfaz se observó que el transistor no llegaba a saturar en forma adecuada por lo que se distorsionaba completamente la señal, haciendo imposible el control de las luces.

Sin embargo, al probar cómo responden las luces a la señal de 3,3v, se observó que estas no presentaban ningún problema por lo que se optó en quitar dicha interfaz y enviar la señal a las tiras directamente desde el microcontrolador. Si bien esto no es lo más adecuado ya que puede presentar algún problema en el futuro, lo ideal hubiera sido realizar una interfaz más veloz que permita manejar estos tiempos.

Beneficios encontrados a lo largo del TPO

Como beneficios se destaca principalmente los conocimientos que fueron adquiridos en áreas que teníamos prácticamente cero experiencias, como puede ser el tratamiento de sonido e imagen, o el uso de UDP, el manejo de SPI, entre otras tantas cosas que fueron necesarias.

Además, el hecho de realizar un proyecto de esta índole nos permite poder seguir haciendo mejoras ya sea agregando modos, personalizando más la interfaz usuario entre otras cosas.



Nos resultó reconfortante poder tener resultados palpables y que se pudiese apreciar realmente la esencia del proyecto mejorando la calidad en el consumo de multimedia.

Links de videos muestra

https://drive.google.com/drive/folders/1BQcozJ1MPGvtescNf1tSFeAtL_6h90Pf?usp=sharing

(ingresar con cuenta frba)

Valoración del TPO

En cuanto al TPO podemos decir que es una gran propuesta y realmente se aprenden muchos conceptos nuevos que de otra forma no sería posible.

Nos hubiese gustado poder tener una mejor organización en ciertos apartados como el versionado de códigos, o el comentado código, pero creemos que aun así se ha realizado un gran trabajo.

En cuanto a la conducción por parte del cuerpo docente, creemos que fue de gran ayuda en ciertas situaciones donde nos aconsejaron cómo resolver ciertas cuestiones, teniendo así una opinión más profesional para tener en cuenta.

Sentimos que en ciertas partes del proceso nos hizo falta equipación, como osciloscopios o fuentes de alimentación de laboratorio, sin embargo, entendemos que esto es por causas mayores.

Se tuvieron algunos problemas debido a la carga horaria de las otras asignaturas, por lo que hubo ciertos aspectos del proyecto que no pudimos llevar a cabo como nos hubiera gustado. Sin embargo, creemos que el resultado final es bastante competente.

Observaciones y mejoras

Como mejoras siempre está la posibilidad de agregar nuevos modos para que el usuario tenga aún más personalización.

Otras áreas donde se podrían mejorar aspectos del proyecto son la captura de la pantalla de modo que no se vea afectada con ciertas aplicaciones como mencionamos anteriormente, la captura de audio para que no se requiera tener un programa externo.



Otra mejora es la implementación de threads para el procesamiento de la imagen, ya que esta demanda bastante al programa principal y puede ocasionar algún tipo de retrasos o lag en la interfaz usuario.

En relación a la mejora anterior, la implementación de un thread que procese todo el control de las luces en segundo plano, sin necesidad de la interfaz gráfica también sería una gran mejora.

Por último, agregar compatibilidad con linux sería una gran mejora de cara al futuro para así poder tener una mayor inclusión.

Conclusiones

Al concluir este proyecto se obtuvieron las siguientes conclusiones:

Si bien la idea de poder controlar las luces mediante el procesamiento de una pc, encontramos que por ahí no es la mejor de las implementaciones, ya que nos “ata” a disponer de una computadora, perdiendo toda posibilidad de ser usado con un televisor. Una forma de abordar este proyecto y no tener este y otros muchos problemas sería capturando la información directamente del HDMI que se conecta al Monitor/televisor.

La implementación del mismo creemos que fue llevada a cabo de manera correcta, y que se obtuvieron buenos resultados, consiguiendo el principal objetivo del proyecto que es el de adquirir una mejor inmersión en el contenido que estamos consumiendo.

Sin embargo, ahora que tenemos más experiencia con el protocolo y las tramas UDP nos damos cuenta que nuestra elección de tramas no fue la mejor considerando los problemas que tuvimos por los tipos de datos. Nos dimos cuenta que este tipo de protocolos está más pensado para caracteres ASCII que para valores numéricos, ya que nos encontramos que dependiendo la plataforma que estábamos usando podía estar o no implementado el envío y la recepción de valores del tipo unsigned char.

En cuanto a cómo resolvimos las problemáticas que se nos fueron presentando creemos que hubo algunas que podrían haber sido resueltas mejor de otra forma, sin embargo, esto es parte de la experiencia y nos brinda el conocimiento para un futuro.

El único aspecto que consideramos que no es del todo práctico es en la implementación del LPC845 para el control de las luces ya que esta tarea puede ser realizada sin ningún problema por el ESP01 (al que pensamos únicamente como periférico) ahorrando recursos y tiempo.



Con respecto a la organización pensamos que fue la adecuada, aunque nos hubiese gustado implementar herramientas como doxygen o gitlab para poder documentar y ser más ordenados.