

<b>Título:</b>	<b>Cuatro en Línea</b>
----------------	------------------------

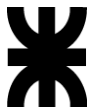
<b>Ciclo Lectivo 2023</b>	<b>Curso N°</b>	<b>R2001</b>	<b>Grupo N°</b>	<b>3</b>
---------------------------	-----------------	--------------	-----------------	----------

<b>Integrantes</b>			Calificación individual	Fecha
Apellido Nombres	Legajo			
Little, Martín Sebastián	203.273-9			
Ortt, Nicolás Agustín	208.635-9			
Upstein, Elias Román	208.649-9			
Vallecillo, Iñaki Manuel	208.681-5			

<b>Calificación grupal:</b>		<b>Fecha:</b>
-----------------------------	--	---------------

<b>Profesor:</b>	<b>Ing. Giura, Marcelo Horacio</b>
<b>Auxiliar/es Docente:</b>	<b>Ing. Soccodato, Gabriel Omar y Gianmuso, Franco</b>

Observaciones primera entrega	
Observaciones segunda entrega	



## Índice:

- I. Prefacio*
- II. Introducción e Idea Fuerza*
  - a. Objetivos*
  - b. Diagramas en Bloque*
- III. Descripción detallada*
- IV. Descripción del Hardware*
- V. Máquina de Estados de la Aplicación*
- VI. Problemas encontrados en el desarrollo*
- VII. Beneficios encontrados en el desarrollo*
- VIII. Conclusiones*
- IX. Bibliografía*



## **I. Prefacio:**

En este informe se llevará a cabo la presentación teórica del trabajo práctico obligatorio (TPO) de Informática II. En primer lugar, se desarrollará el marco de lo realizado y aplicado, seguido de la exposición de las dificultades que aparecieron a lo largo de todo el mismo, para luego concluir con las herramientas que fueron adquiridas durante el transcurso del proyecto.

## **II. Introducción e Idea Fuerza:**



Se realizó como proyecto el conocido juego “Cuatro en Línea” o “Cuatro en Raya” según la versión, en donde dos jugadores se turnan para colocar una ficha a la vez en un tablero, tradicionalmente de seis filas por siete columnas. Las fichas tienen distintos colores para diferenciar a los jugadores entre sí. Quién logre colocar cuatro fichas del mismo color de manera contigua, ya sea en diagonal, horizontal o vertical, ganará la partida.

Para representar el tablero de juego se utilizó una matriz de tipo LED RGB de ocho filas por ocho columnas, en donde las fichas de cada jugador se representan por un color distinto. Mediante una interfaz gráfica conectada al juego, los jugadores pueden seleccionar el color que más sea de su agrado. Luego, podrán desplazarse por la primera fila del tablero para elegir dónde dejar caer su ficha a través de la propia interfaz, o por medio de un teclado presente en la placa Infotronic utilizada en la cátedra.

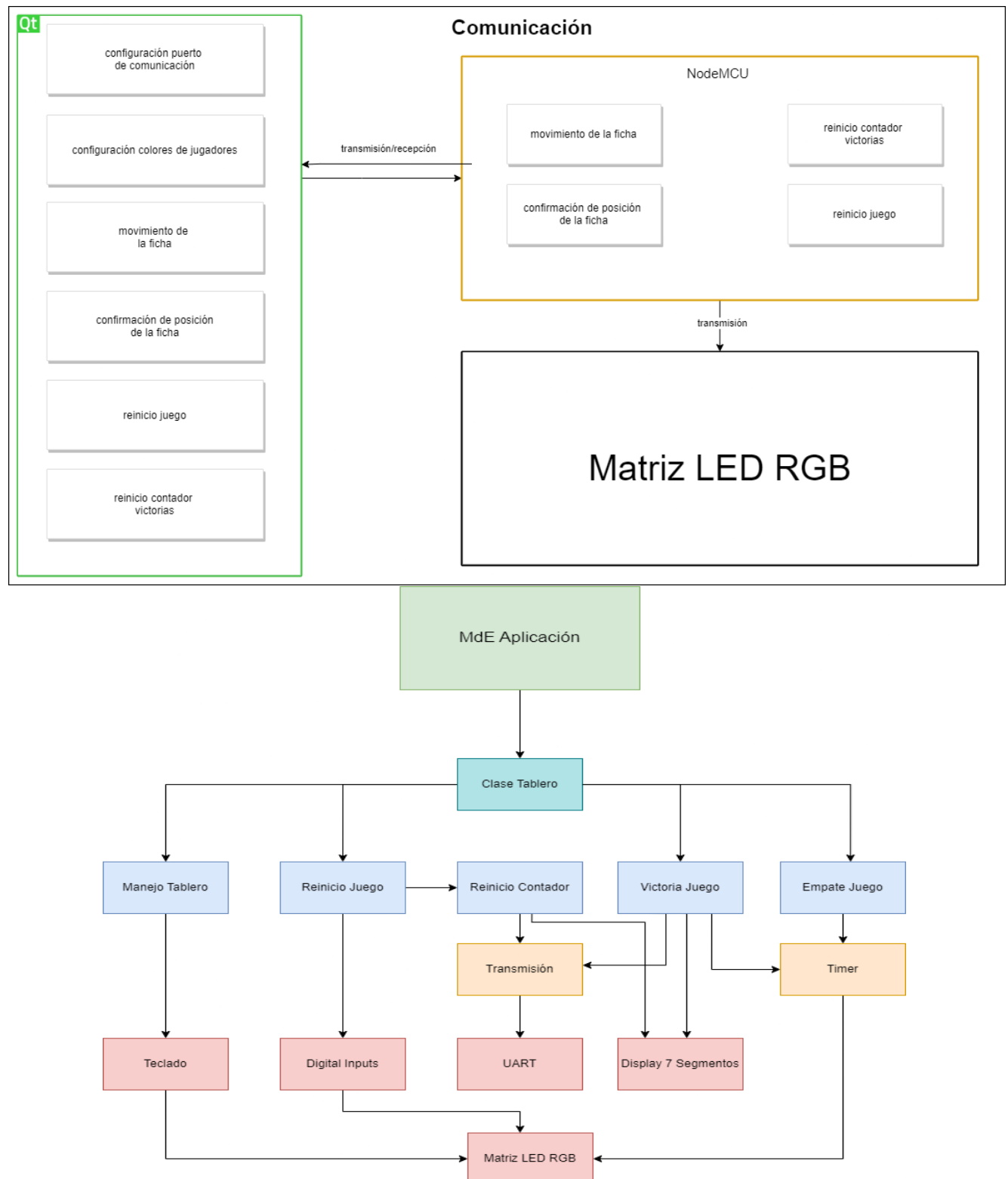
Además, los jugadores cuentan tanto en la interfaz gráfica como en la placa Infotronic con dos display's del tipo siete segmentos en los que podrán visualizar la cantidad de victorias acumuladas que lleva cada uno, pudiendo resetear estos contadores a través de un pulsador de la placa o un botón de la interfaz.

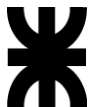
### **a. Objetivos:**

- Lógica del juego.
- Implementación de diversos periféricos.
- Comunicación de la matriz con el LPC845.
- Comunicación de la interfaz gráfica con el LPC845.



b. Diagramas en bloques:





### **III. Descripción detallada:**

Al proyecto se le pueden atribuir dos diagramas de bloques principales, los cuales fueron presentados anteriormente. El primer diagrama describe la comunicación de la interfaz gráfica con el LPC845 y con la matriz LED-RGB, mientras que el segundo corresponde a la interfaz compuesta entre la aplicación y los elementos correspondientes de hardware y software.

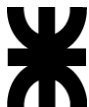
A la hora de profundizar en el primer diagrama, es pertinente realizar algunas aclaraciones. Al iniciar la interfaz gráfica, se pedirá configurar el puerto de comunicación por el cual se realizará la transmisión/recepción de los datos. La cantidad de puertos disponibles dependerá del ordenador en donde se esté ejecutando. Independientemente de esto, se recomienda establecer la comunicación serie con el LPC845 para obtener más funcionalidades.

Una vez configurado el puerto serie, la interfaz gráfica permitirá que los usuarios elijan los colores que utilizarán para el juego. Hecho esto, el LPC845 validará los datos recibidos e inicializará la matriz para que pueda comenzar la partida. Los jugadores podrán interactuar con el juego mediante el teclado o la interfaz gráfica.

Pasando al segundo diagrama de bloques, se explicará en detalle la comunicación en capas entre la máquina de estado de la aplicación y los distintos periféricos de la misma. En primer lugar, es importante destacar la clase tablero. Esta clase es una abstracción de lo que sería un tablero de juego físico, la cual también está vinculada con la matriz LED-RGB para poder vincular el movimiento de las fichas con la visibilidad del mismo. El manejo de las fichas se realiza por medio del teclado del kit, ya sea la acción de desplazar la ficha de izquierda a derecha o confirmar la posición en donde se quiere que ésta caiga.

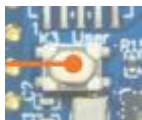
La función de resetear juego se encarga de reiniciar los display's pertenecientes al kit, borra lo que se encuentre en la matriz y se comunica con la interfaz gráfica mediante la UART para que la misma se reinicie y así poder seleccionar nuevamente los colores y reiniciar los contadores por medio de la transmisión de los valores.

Finalmente, con respecto a "victoria juego" y "empate juego", cuando se llega a la situación donde alguno de los jugadores ganó o ninguno lo hizo, todos los leds pertenecientes a la matriz se prenden del color del ganador o en lo contrario de color blanco, por cinco segundos. Esto quiere decir que se dispara un timer de cinco segundos, donde luego se reinicia el juego para una nueva partida.



## IV. Descripción del Hardware:

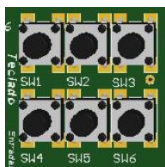
- Entrada Digital (pulsador del stick o final de carrera):



Se utilizó un pulsador del stick, el cual coincide su conexión eléctrica con el final de carrera montado en la placa Infotronic, conectado mediante un pull up como entrada activa baja.

Para su codificación, se hizo uso de la clase DigitalInputs desarrollada y estudiada durante el desarrollo de la cursada.

- Teclado Matricial



La aplicación utiliza un teclado matricial de tres teclas para la interacción con el usuario. El mismo se encuentra presente en la placa Infotronic y también se aprovechó del material de clase para su programación mediante la clase Teclado.

El teclado cuenta con un pin de scan y tres de return; si bien para el caso es menos eficiente el uso de este tipo de teclados ya que consume un pin más del necesario al conectar los pulsadores como entradas, se decidió en el diseño hacer uso del mismo para sacar provecho de los temas vistos en clase y no generar complicaciones de hardware desarrollando placas PCB que no hacen al contenido del TPO.

- Display's 7 segmentos



De igual forma que en los casos anteriores, se hizo uso del material disponible utilizando la clase Display y el hardware presente en el Infotronic. Se puede destacar la presencia de los integrados CD4017 y CD4511 los cuales se ocupan de generar el barrido de los display's y la conversión BCD a siete segmentos.

- Matriz LED RGB 8x8



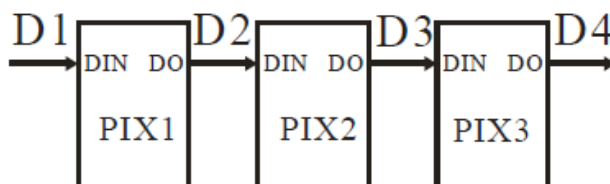
A modo de desafío para el proyecto, se integró esta matriz que contiene leds rgb del tipo WS2812B, los cuales precisan de un protocolo one wire para comunicarse. Cada led consta de 24 bits (8 bits para cada color) los cuales deben ser mandados uno a uno (del MSB a LSB) mediante la siguiente secuencia:



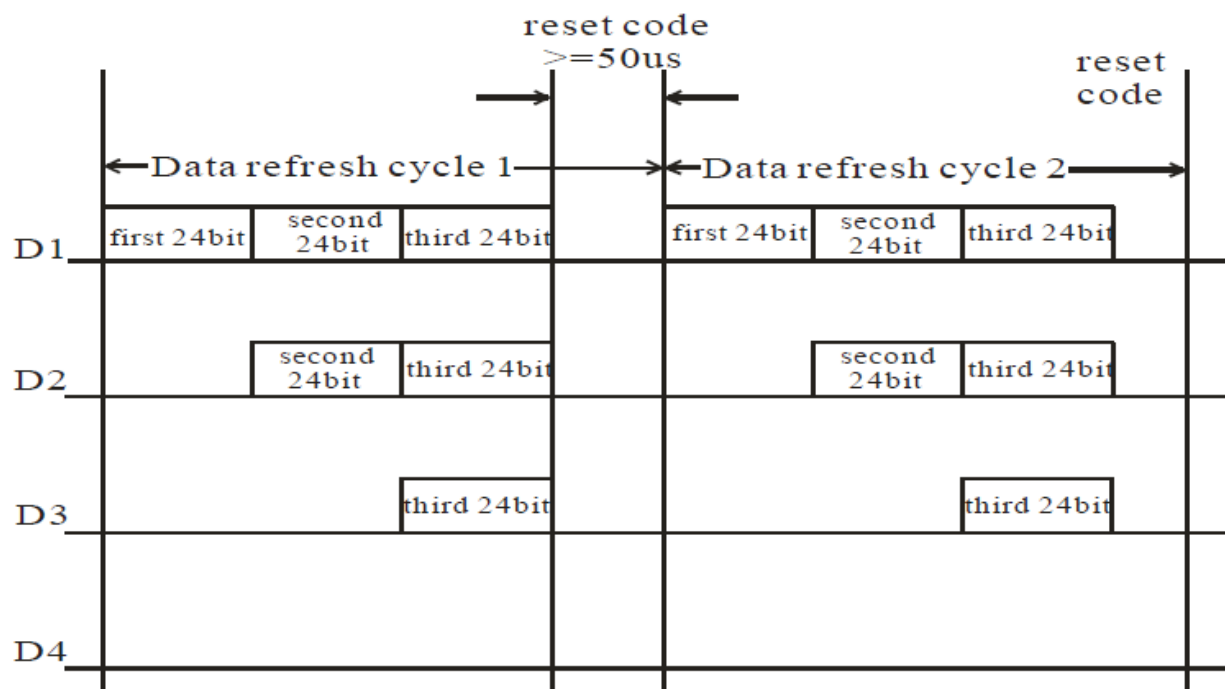
Data transfer time( TH+TL=1.25μs±600ns)			
0 code			
1 code			
RET code			
	T0H	0 code ,high voltage time	0.4us
	T1H	1 code ,high voltage time	0.8us
	T0L	0 code , low voltage time	0.85us
	T1L	1 code ,low voltage time	0.45us
	RES	low voltage time	Above 50μs

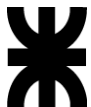
El fabricante informa que, para comunicarse con más de un led, se deben encadenar los 24 bits sin dejar pasar los 50 μS de reset:

**Cascade method:**



**Data transmission method:**

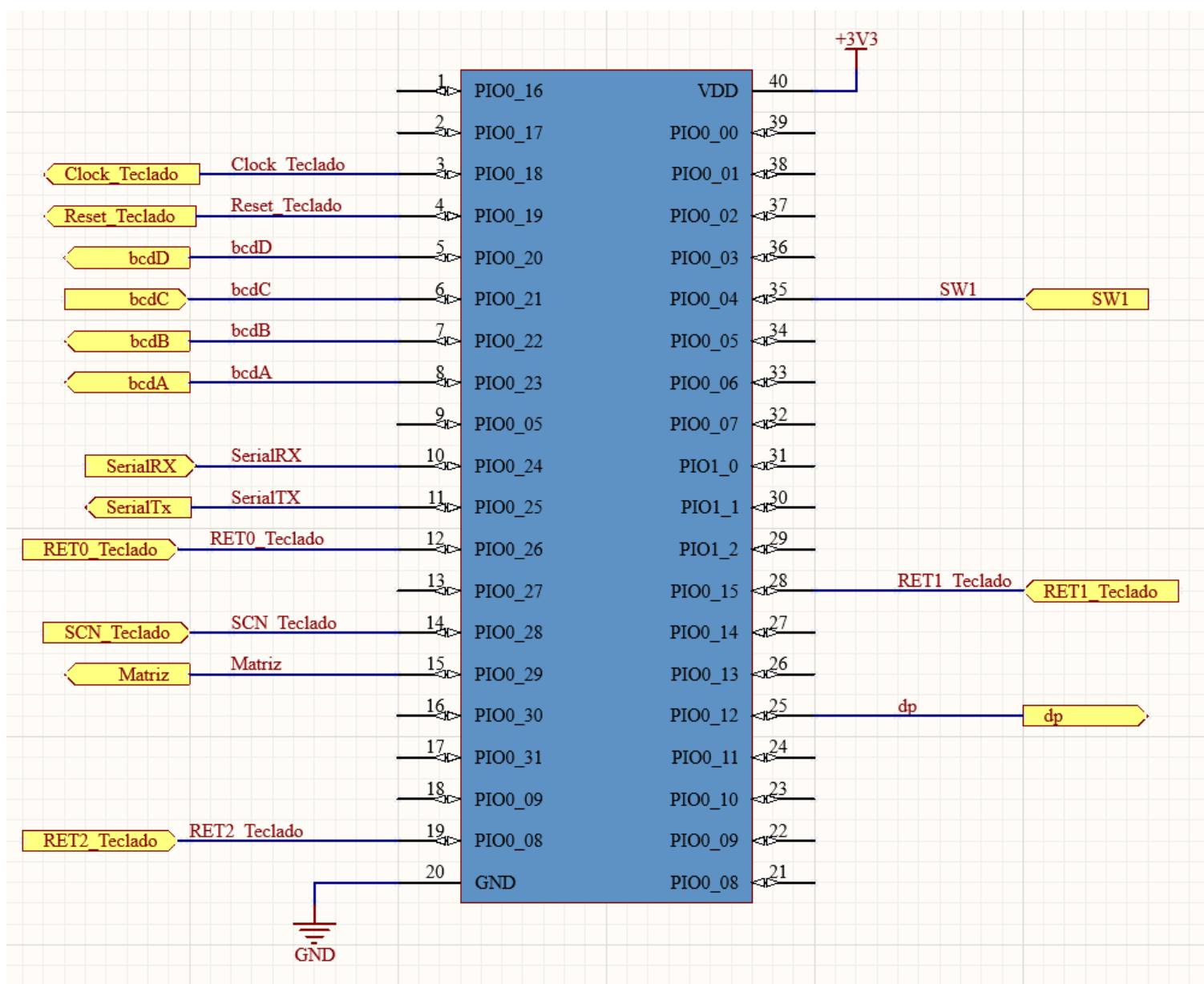




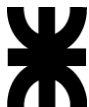
En este punto cabe resaltar la dimensión de los tiempos que requiere cada bit, lo cual será desarrollado más adelante en este informe.

Datasheet: [WS2812B - Adafruit](#)

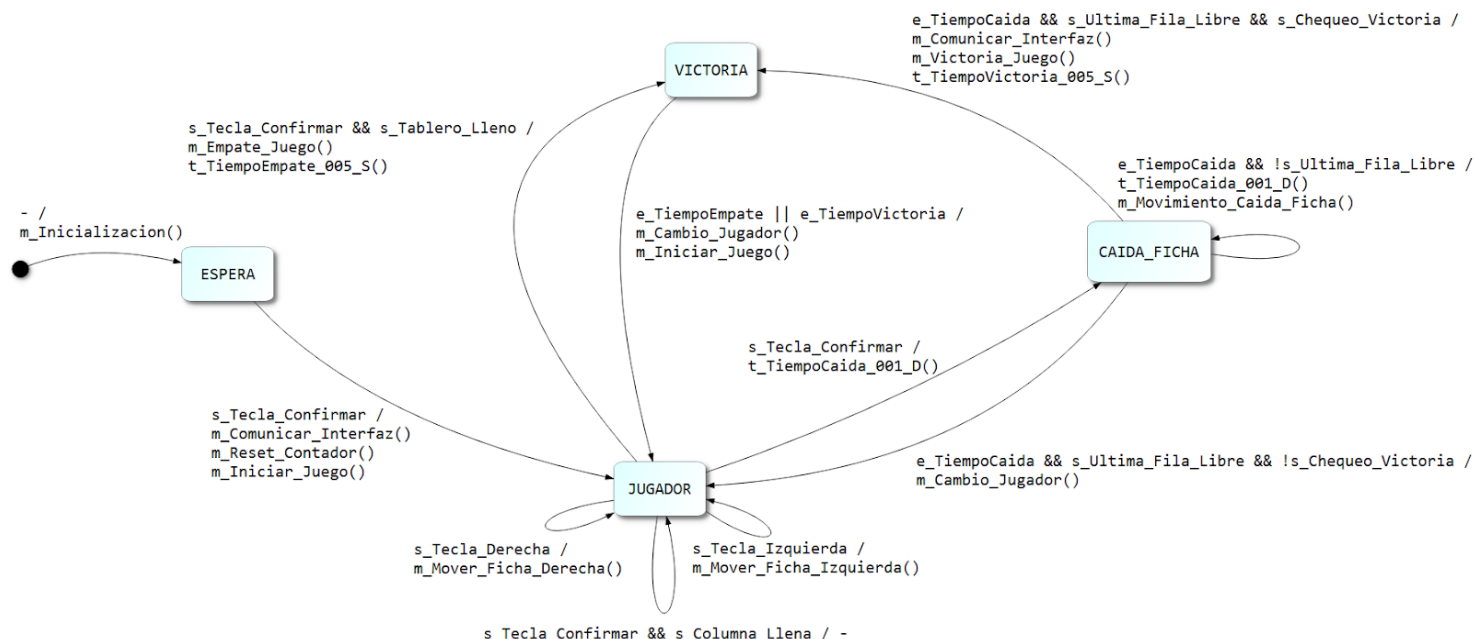
Circuito esquemático de conexión:







## V. Máquina de Estados de la Aplicación:



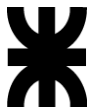
## VI. Problemas encontrados en el desarrollo:

Como se mencionó anteriormente, el gran problema a la hora de desarrollar el TPO se encontró en conseguir la temporización necesaria para comunicarse con la matriz desde el microcontrolador.

Resulta conveniente primero mencionar, que el diseño de la clase MatrizLed se basa en la [ya desarrollada para Arduino](#), y que se barajó como segunda opción, en caso de no conseguir comunicarse de manera directa con la matriz, utilizar un Arduino UNO o Nano como controlador mediante comunicación serie con el LPC845.

El proceso para comunicarse con la matriz fue extenso. Primero se planteó la posibilidad de utilizar el CTimer y, mediante el evento del match, realizar en la interrupción la lógica necesaria para enviar los datos. Sin embargo, notamos que realizar este algoritmo con interrupciones generaba que el microcontrolador se “colgara” ya que no le alcanzaba el tiempo para realizarlo de manera correcta.

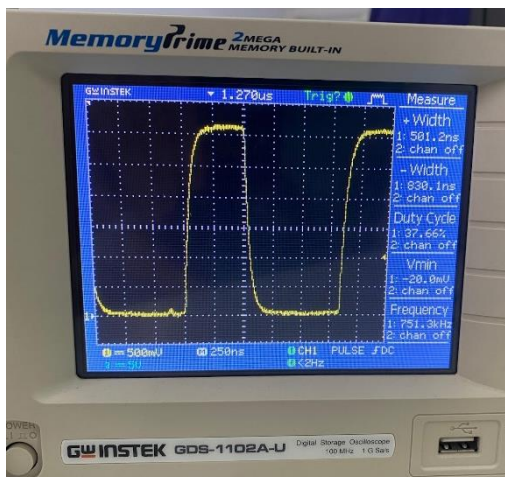
Se propuso entonces que el envío de datos fuera por pooling y que el método que se ocupe de esto sea de tipo bloqueante, ya que el micro no se puede permitir perder tiempo en otras acciones que no sea la comunicación; cabe destacar que para esto se deben suspender temporalmente todas las interrupciones activas. Similar al caso anterior, si bien se consiguió que el programa ya no se cuelgue y se ejecute de manera completa,



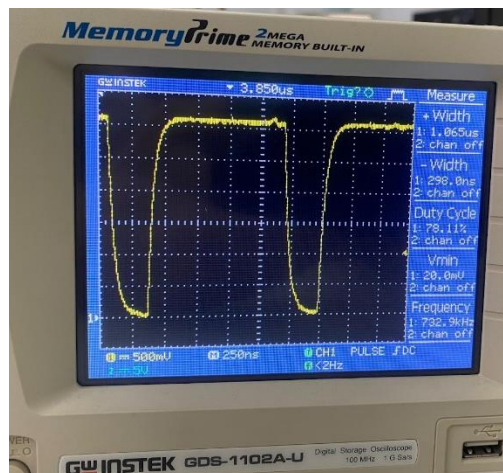
los leds de la matriz se encendían en blanco en su máxima intensidad, lo que hizo suponer que estaba recibiendo todos los bits en alto, y chequeando esto con las señales presentadas más arriba, esto implicaba que los tiempos no eran los necesarios.

Finalmente, y con la ayuda de los docentes, se debuggeó la señal que se emitía con un osciloscopio, verificando la suposición de que el tiempo era excesivo. Para solucionar esto, se setearon los Wait States de la flash en cero, aprovechando al máximo la velocidad del micro. Además, la lógica del programa dejó de ser por pooling y pasó a ser secuencial, configurando las pérdidas de tiempo (delays) con el comando de Assembler “nop”. Otra cosa a mencionar es que se implementó una función para conseguir que el dato que se estaba enviando se almacene directamente en el registro del micro, cosa que no pudo optimizarse mediante un “for” ya que esto rompía nuevamente la temporización.

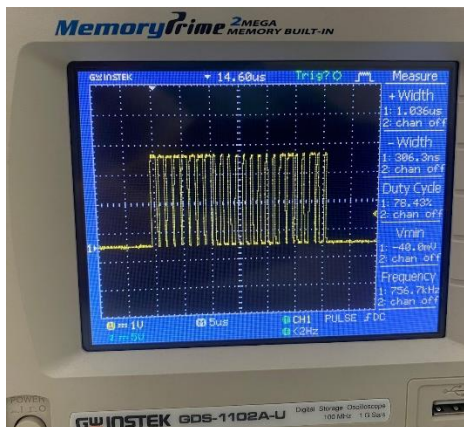
A continuación, se adjuntan las imágenes de las señales conseguidas con la cual la matriz funciona de manera correcta (en la medición del osciloscopio se puede observar la temporización):



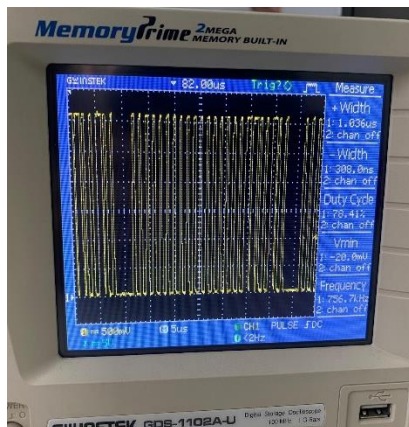
\*Señal de bit 0.



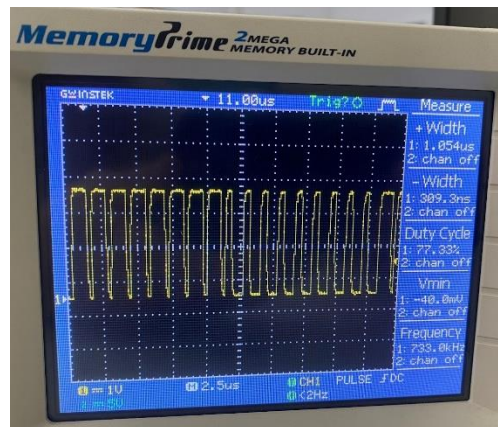
\*Señal de bit 1.



\*Trama de 24 bits.



\*Trama de 24 bits para varios leds.



\*Señal de datos de prueba enviada.



## **VII. Beneficios encontrados en el desarrollo:**

Esta clase de trabajos prácticos permiten encontrarse con numerosos beneficios, entre los cuales se destaca el obtener experiencia en problemas reales que podrían aparecer a lo largo de la vida profesional. Así también la posibilidad de poner en práctica el contenido desarrollado y aprendido durante la cursada de la materia. Además, la necesidad de investigar y aprender sobre nuevas herramientas para llevar a cabo un proyecto representa una gran práctica para futuros trabajos.

Para mejorar a futuro deseamos que haya material opcional para temas no vistos en la cursada, pero que comúnmente aparezcan en proyectos. Esto se podría solucionar agregando en el aula una carpeta que contenga todo el material de estudio posible, y así agilizar enormemente la comprensión del mismo.

## **VIII. Conclusiones:**

Se logró desarrollar de manera correcta la idea planteada para poder jugar al juego “Cuatro en Línea” mediante el proyecto. Esto incluye el agregado de funcionalidades no planteadas hasta el momento del desarrollo.

Se cumplieron los objetivos propuestos en cuanto a lograr una lógica de programa clara, implementar diversos periféricos vistos en la cursada (como el teclado y el display) y desarrollar una interfaz gráfica funcional al proyecto. Se destaca por sobre el resto el haber podido alcanzar la comunicación efectiva con la matriz led, siendo este el *desafío* a superar.

Además de lo ya descrito, se incorporaron herramientas tanto de software para programar como de hardware en electrónica desconocidas hasta el momento. También podemos mencionar las ventajas de realizar este tipo de proyectos en equipo ya que se asemejan a trabajos del mercado laboral. Esto fortalece nuestras habilidades de compromiso, responsabilidad y división de tareas, entre otras.

En cuanto a aspectos a mejorar, se reconoce la necesidad de realizar una división de tiempos más certera, con el fin de no especular con la fecha de entrega.

En líneas generales, los integrantes del grupo se muestran satisfechos con su desarrollo y aprendizaje a lo largo del Trabajo Práctico Obligatorio.

## **IX. Bibliografía:**

Repositorio del proyecto: [https://github.com/EliasUpstein/LPC845\\_4enLinea](https://github.com/EliasUpstein/LPC845_4enLinea)

Página de la matriz del fabricante: <https://www.adafruit.com/product/1487>

<https://learn.adafruit.com/adafruit-neopixel-uberguide/downloads>

Librería del fabricante para otros microcontroladores: [https://github.com/adafruit/Adafruit\\_NeoPixel](https://github.com/adafruit/Adafruit_NeoPixel)