

FACULTAD DE  
INGENIERÍA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## Informe inicial Cartel lumínico configurable por WiFi

Taller de proyecto 1 (E0306)

Grupo 07

García	Agustín
Levy	Santiago
Romero Dapozo	Ramiro
Ternouski	Sebastian Nahuel

La Plata, Octubre de 2017

# Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Objetivos del proyecto</b>	<b>1</b>
<b>3. Análisis de requerimientos</b>	<b>1</b>
<b>4. Diseño del hardware</b>	<b>2</b>
<b>5. Diseño de software</b>	<b>2</b>
5.1. Panel de configuración (PC) . . . . .	2
5.2. Software embebido . . . . .	2
<b>6. Ensayos y mediciones</b>	<b>2</b>
<b>7. Conclusiones parciales</b>	<b>3</b>
<b>Bibliografías</b>	<b>4</b>
<b>Apéndice</b>	<b>5</b>
<b>A. title</b>	<b>5</b>
<b>B. Protocolo para control de cartel luminoso</b>	<b>5</b>
B.1. Codificación . . . . .	5
B.1.1. Strings . . . . .	5
B.2. Establecimiento de conexión . . . . .	5
B.3. Desconexión . . . . .	5
B.4. Interacciones . . . . .	5
B.4.1. Authenticate . . . . .	6
B.4.2. SetText . . . . .	6
B.4.3. SetAnimationParameters . . . . .	6
B.4.4. SetWifiConfiguration . . . . .	7
B.4.5. GetText . . . . .	7
B.4.6. GetAnimationParameters . . . . .	7
B.4.7. GetWifiConfiguration . . . . .	7

## 1. Introducción

En el ámbito universitario resulta necesario mantener informadas a las personas sobre una amplia variedad de hechos, noticias y acontecimientos que sucedieron o sucederán, desde la ubicación de un aula hasta la notificación de la cancelación de una clase. Muchas veces estas notificaciones son sobre cuestiones muy efímeras, lo que requiere rapidez para empezar a transmitir las y facilidad para tener el alcance necesario.

En las facultades de la Universidad Nacional de La Plata se consumen muchos recursos para cumplir este fin, a través de afiches, pancartas, panfletos, etc. los cuales, pese a ser de barata fabricación, no tienen una vida útil muy extensa. Además todas estas formas de comunicación se basan en el uso de papel, que tras ser utilizado debe desecharse debido a la imposibilidad de reutilizarlo, generando una cantidad de residuos significativa. Si se tiene en cuenta que también generan una contaminación visual considerable, por la gran cantidad de estos distribuidos en todos los lugares transitables, resulta prudente considerar una nueva forma de comunicación.

Surge así la idea de desarrollar de un cartel electrónico reutilizable, capaz de ser configurado remotamente por las autoridades competentes, con el fin de proveer una forma de comunicación masiva más limpia, clara y menos dañina para el medio ambiente.

## 2. Objetivos del proyecto

El objetivo principal del proyecto es implementar un cartel de LEDs que pueda ser configurado remotamente por un usuario. El mismo se puede subdividir en subobjetivos, los cuales se mencionan a continuación:

Diseñar e implementar el Hardware (PCB, matrices de LEDs) del cartel para que visualice correctamente el mensaje configurado. Deberá ser modularizable, es decir, que sea posible agregar módulos funcionales y expandir la cantidad de caracteres en un renglón.

Desarrollar el software embebido que se ejecutará en el microcontrolador.

Desarrollar el software que controla el cartel. Éste podrá ser usado desde una pc y tendrá una interfaz gráfica con formato de panel para controlar las características del mensaje a mostrar por el cartel.

Diseñar e implementar protocolo de comunicación para la comunicación entre el software controlador del panel y el cartel. El protocolo debe establecer un método de autenticación que impida el acceso de personal no autorizado al cartel. Esto implica que sea seguro antes ataque del tipo *man-in-the-middle* y otros tipos de ataques bien conocidos.

## 3. Análisis de requerimientos

El panel proveerá una interfaz gráfica para que el usuario pueda configurar un mensaje para el cartel, así como modificar características secundarias como parpadeo, velocidad de movimiento o estaticidad del mensaje. A través del panel, el usuario podrá ingresar por teclado el mensaje a mostrar mediante los caracteres que se establecen en el estándar (ver [8859-1, 1998]). Además, el panel permitirá relevar la configuración y el estado actual de lo que se está mostrando en el cartel. La autenticación correspondiente al ingreso al panel será verificada mediante el uso de un token de identificación.

El cartel contará con matrices de LEDs blancos que en conjunto y sintonía formarán las letras correspondientes al mensaje a transmitir, permitiendo mostrar el mensaje configurado por el usuario.

El software compila ahora y aguarde esto

## 4. Diseño del hardware

El cartel se compone de varios módulos. Cada módulo contiene una matriz de LED 8x8, un chip shifteamor [MAX7219, 2003], y componentes pasivos varios, logrando así una unidad funcional independiente (ver figura 1).

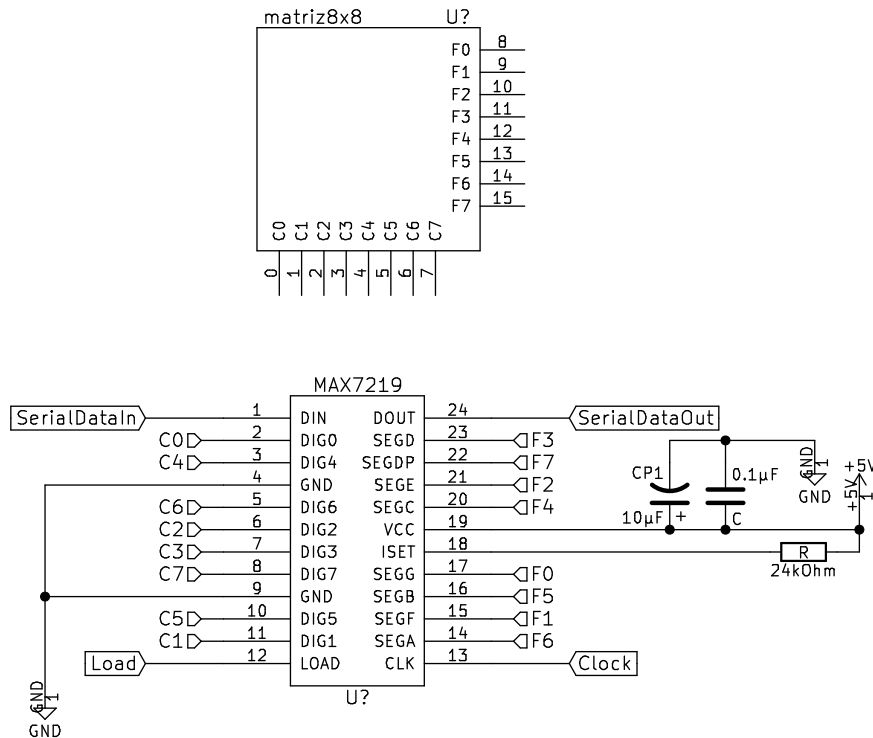


Figura 1: Esquema de conexiones de un módulo funcional.

El MAX7219 recibe una secuencia de bytes de forma serial y los deriva a la columna que corresponda para cada instante en la matriz de LEDs del módulo correspondiente en el cartel.

## 5. Diseño de software

### 5.1. Panel de configuración (PC)

### 5.2. Software embebido

## 6. Ensayos y mediciones

Se desarrolló un prototipo de uno de los módulos funcionales del cartel, conectando los componentes sobre un protoboard y soldando una matriz de LEDs con la disposición a utilizar en el modelo final (ver Figura 2). A través del mismo se hicieron pruebas para corroborar el correcto funcionamiento particular de un módulo funcional y la claridad del carácter mostrado. Se corroboró que el mensaje comienza a mostrarse pocos instantes después de que el mismo empieza

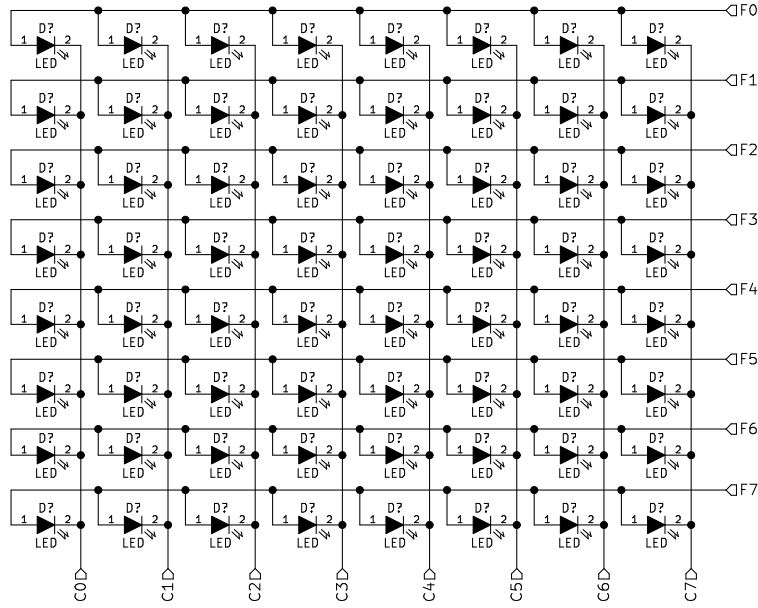


Figura 2: Esquema de conexiones de un módulo funcional.

a ser recibido por el cartel. También se puso a prueba la tensión máxima soportada por los LEDs utilizados, los cuales demostraron la capacidad de soportar hasta 4,2 V de tensión con una corriente continua de 20 mA. En dicho prototipo se usaron resistencias que diferían levemente de las planteadas en el esquema original debido a problemas de disponibilidad de las mismas, confirmando que el módulo funciona correctamente con un margen de tolerancia para los valores de tensión y corriente en los LEDs. El prototipo también permitió confirmar que el sistema puede prescindir de un disipador térmico ya que los componentes no levantan una temperatura considerable, aún después de un funcionamiento prolongado.

## 7. Conclusiones parciales

Se planteó el esquema de hardware y se desarrolló el primer prototipo del módulo funcional.

Se corroboró que la implementación práctica de la matriz de LEDs usando el esquema planteado originalmente permite mostrar correctamente el mensaje que el usuario disponga.

Se diseñó un protocolo para la comunicación entre el panel de configuración y el cartel (ver Apéndice B).

Se desarrolló la interfaz gráfica del panel a utilizar para que el usuario configure el mensaje y otro programa que simula una matriz de 8x8 (con checkbox) y permite al usuario diseñar los caracteres a su gusto. El sistema entonces transforma el pedido del cliente a un archivo de bitmaps.

Se escribió un programa base en el módulo ESP8266 que acepta conexiones TLS y, a modo de prueba, se programó para que responda haciendo `.echo`<sup>a</sup> los mensajes que recibe desde el otro extremo.

Se desarrolló una aplicación de escritorio, que consiste en un cliente QT, con el objetivo de

probar las conexiones TLS que se utilizarán para el intercambio de información entre el usuario y el sistema.

El desarrollo de las actividades viene cumpliendo satisfactoriamente con los tiempos del cronograma estipulado.

Hasta el momento el presupuesto invertido por los alumnos en el proyecto para el desarrollo del prototipo de un módulo funcional ronda los 500 pesos.

## Bibliografías

- 8859-1. [1998]. *Codificación del alfabeto latino* [Estandar]. ISO/IEC 8859-1:1998. (Link: [Más información](#))
- MAX7219. [2003]. *Leds display driver* [DataSheet v4]. Maxim Integrated. San Jose, CA 95134 USA. (Link: [Datasheet](#))

# Apéndice

## A. title

## B. Protocolo para control de cartel luminoso

### B.1. Codificación

Cada mensaje del protocolo es un paquete de tamaño variable, con un límite máximo de 255 octetos. (Incluyendo todos los headers)

Todos los valores numéricos se transmiten con orden de bytes Big-Endian.

En notación C, cada mensaje tiene el formato:

Message.version es un número que identifica la versión del protocolo, por si se hicieran cambios y se quiera mantener retrocompatibilidad.

Message.type es un enumerativo que codifica el tipo de mensaje. Message.size indica el tamaño del mensaje, tiene rango [3, 253].

Un mensaje inválido se descarta y termina la conexión. (bajo TLS, no puede tratarse de corrupción, así que es un cliente mal implementado o malicioso)

Message.content tiene una estructura que depende de Message.type y se define para cada procedimiento más adelante.

#### B.1.1. Strings

Los strings tienen un límite de longitud dado por el lugar remanente en el mensaje y están codificados según el estándar ISO/IEC 8859-1:1998. Antes del contenido del string debe haber un byte que indique la longitud del string.

### B.2. Establecimiento de conexión

El establecimiento de una conexión está conformado por el establecimiento de una conexión TLS al servidor iniciada por el cliente, seguida de la interacción Authenticate. Si la interacción Authenticate fallase, el servidor debe cerrar la conexión inmediatamente.

El acceso al servidor dura lo que dure la conexión; no debe mantenerse estado de autenticación fuera de una conexión.

### B.3. Desconexión

La desconexión es sólo la de TLS. Una desconexión a nivel TCP sin el procedimiento de desconexión de TLS se considera un error.

### B.4. Interacciones

Una interacción es una secuencia de dos mensajes: un mensaje de pedido y un mensaje de respuesta. Cada interacción está formada por un mensaje de pedido seguido por un mensaje de respuesta. Las interacciones posibles son:

- Authenticate
- SetText



- SetAnimationParameters
- SetWifiConfiguration
- GetText
- GetAnimationParameters
- GetWifiConfiguration

La respuesta proveniente del servidor tiene el siguiente formato:  
 ResponseCode puede tomar los siguientes valores:

- Respuesta genérica indicando éxito: 0
- Error genérico: -1
- Paquete inválido: -2
- Paquete incompatible: -3
- Configuración de WiFi inválida: -4

El campo response es opcional, puede ser de tamaño nulo y su contenido depende de la interacción de la cual es respuesta. Se dice que una interacción tiene una respuesta vacía cuando el campo response está ausente. Toda interacción tiene respuesta vacía salvo que se indique lo contrario.

#### **B.4.1. Authenticate**

Cuando se inicia la conexión SSL, el cliente manda la password para entrar al sistema. El servidor responde con una respuesta de OK y la conexión permanece establecida hasta que el cliente decida cerrarla. (Cerrar la conexión SSL implica primero señalar su fin, solo hacer FIN o RST se considera como una intrusión a la conexión y es detectable por ambas partes)

#### **B.4.2. SetText**

Actualiza el mensaje del cartel con el enviado en el pedido.  
 El servidor responde con un OK o con un código de error.  
 Descripción del contenido del mensaje:  
 El campo msg es una cadena de caracteres terminada en cero.

#### **B.4.3. SetAnimationParameters**

Actualiza la configuración de la animación del cartel.  
 SetAnimParamsRequest.brate y SetAnimParamsRequest.srate son la frecuencia de parpadeo en Hz y la velocidad de deslizamiento en píxeles por segundo.  
 Se asume que si SetAnimParamsRequest.brate es cero, no se debe parpadear el contenido. De la misma forma, si SetAnimParamsRequest.srate es cero, no se debe deslizar el contenido.  
 El tipo de dato ufp844 es un número en punto fijo sin signo con 4 bits de parte entera y 4 bits de parte fraccionaria. El tipo de dato sfp844 es lo mismo que ufp844 pero en complemento a dos.

#### **B.4.4. SetWifiConfiguration**

Actualiza la configuración de WiFi del servidor. En caso de éxito, el servidor debe responder con el código de respuesta de éxito, luego debe cerrar la conexión y por último debe conectarse a la red indicada bajo la IP indicada. Si fallara en hacer eso, vuelve a la configuración anterior.

Descripción del contenido del mensaje:

SSID es el nombre de la red, password es la contraseña de la red, ip es la IP que va a tomar el cartel y subnet es la máscara de la subred.

#### **B.4.5. GetText**

Devuelve la estructura SetTextRequest con el mensaje actual del cartel. Siempre retorna código de éxito.

#### **B.4.6. GetAnimationParameters**

Devuelve la estructura SetAnimationParameters con la configuración de animación actual del cartel. Siempre retorna código de éxito.

#### **B.4.7. GetWifiConfiguration**

Devuelve la estructura SetWifiConfigurationRequest con los datos de la red a la que el cartel está conectado y la IP que tiene en ella. Siempre retorna código de éxito.