

Programador Bluetooth para Display con Leds RGB Inteligentes

Facundo D. Vogel ^{a, *}; Mateo A. Mazur ^{a, *}; Guillermo A. Fernandez ^{a, b, *}

^a Facultad de Ingeniería, Universidad Nacional de Misiones (FI-UNaM), Oberá, Misiones, Argentina.

^b GID-IE, FI-UNaM, Juan Manuel de Rosas 325, Oberá, Misiones, Argentina.

e-mails: facundovog@gmail.com, mateo.amz@gmail.com, guillermo.fernandez.fio@gmail.com

Resumen

Este trabajo es realizado en la asignatura Técnicas Digitales 2 de la carrera Ingeniería Electrónica y propone el desarrollo de un sistema electrónico basado en el microcontrolador ATmega 328p (bajo la plataforma Arduino UNO), capaz de controlar un display compuesto por 600 leds RGB inteligentes. Cada led posee un chip cuya entrada serial de datos permite programar su color y brillo. El display se gobierna mediante el circuito desarrollado, mostrando un texto con variedad de efectos. El texto es programable por el usuario con una aplicación de celular (desarrollada en este trabajo) la cual se comunica mediante bluetooth con el circuito mencionado. La aplicación funciona bajo Android y proporciona opciones para modificar el texto, cambiar su color, velocidad de desplazamiento y brillo en el display. El circuito electrónico del sistema es implementado a modo de *shield* para Arduino UNO y soporta al módulo bluetooth utilizado, incorporando las conexiones de la fuente de alimentación del display y la línea de datos de los leds. Para comprobar el funcionamiento del sistema, se realizaron pruebas escribiendo en el display palabras con diferentes efectos y niveles de brillo. Los resultados obtenidos fueron aceptables, ya que el sistema respondió correctamente ante las acciones requeridas por el usuario mediante la aplicación del celular.

Palabras Clave – Programador, Bluetooth, Display, Led, Arduino, Atmega328p, Microcontrolador.

1. Introducción

Este trabajo resulta como actividad integradora de la asignatura “Técnicas Digitales 2”, aplicado las capacidades y conocimientos adquiridos respecto al desarrollo de sistemas embebidos utilizando el microcontrolador ATmega 328p [1] con la plataforma Arduino Uno [2]. Atendiendo a esto, el trabajo aquí presente propone el proyecto y diseño de un sistema capaz de controlar un display construido a partir de leds inteligentes que están configurados en un arreglo de 15x40 (filas x columnas). Estos leds, denominados comercialmente NeoPixel, incorporan individualmente el circuito integrado WS2812 [3], que está encargado de permitir la programación y también de controlar el color y brillo del led. Los parámetros de iluminación mencionados se programan a través de una entrada serial de datos que posee dicho circuito integrado. Cada led también incorpora una salida serial de datos, a través de la cual se interconecta con otro led. De esta forma, el display antes indicado se constituye como una tira de 600 leds en cascada, donde sólo el primer led posee conexión directa de datos con el microcontrolador, mientras que cada uno de los demás leds recibe información (para su programación) del anterior según la secuencia de conexión. El sistema desarrollado permite visualizar un texto en el display y generar efectos tales como el cambio de color, movimiento y brillo, siendo estos configurables por el usuario mediante una aplicación desarrollada para funcionar con teléfonos que utilizan sistemas operativos Android y que además

* Autores en correspondencia

poseen conectividad bluetooth. La comunicación entre celular y microcontrolador se realiza a través del módulo HC-06 [4], que permite la transferencia de datos a través del protocolo bluetooth [5].

2. Metodología

El desarrollo del trabajo se divide en tres partes fundamentales; el programa para el microcontrolador, la aplicación que se ejecuta en el celular y el circuito de control. Estas tres partes se describen a continuación.

2.1. Programación del microcontrolador

El código del programa que ejecuta el microcontrolador Atmega 328p es desarrollado utilizando el entorno Arduino IDE, el cual se basa en los lenguajes C y C++ pero integrando reglas especiales que facilitan el uso de la plataforma Arduino y de módulos que se pueden incorporar a esta.

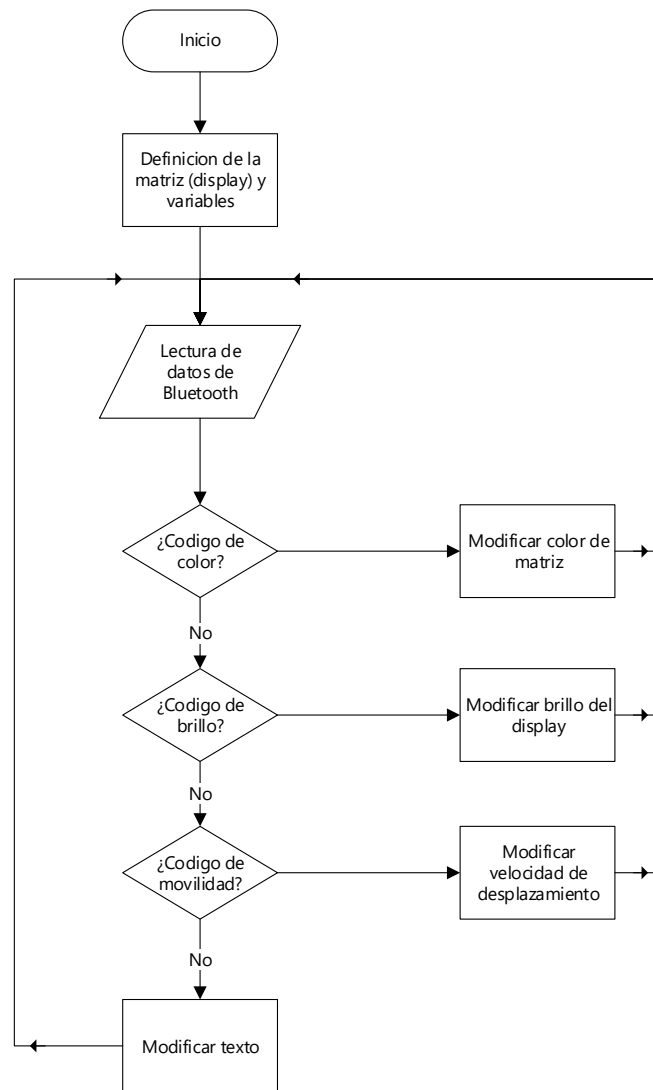


Fig. 1 – Diagrama de flujo para código embebido en el microcontrolador.

En el programa se utilizan diversas librerías tales como “Adafruit Neopixel” para el control básico de los leds [6], “WS2918fx” la cual incorpora efectos visuales [7], “Pololou led strip” que incorpora efectos asociados a una tira de leds [8], “NeoPixel Bus” diseñada para el control de múltiples leds [9], y “Octo Ws2812” la cual se utiliza para optimizar la comunicación del microcontrolador con los leds [10]. Para el desarrollo de esta experiencia se utilizó el microcontrolador Atmega 328p bajo la plataforma Arduino UNO, sin embargo, la utilización del entorno de desarrollo Arduino IDE permite utilizar otra versión con otro microcontrolador sin la necesidad de realizar grandes cambios en el código.

Como se observa en la Fig. 1, el microcontrolador ejecuta un bucle principal que verifica la conexión bluetooth, lee los datos recibidos desde el celular y verifica si estos corresponden a un código reservado que se interpreta como una solicitud de cambio de color en el display, en caso negativo procede a verificar si el código está asociado al cambio de brillo, si esto resulta nuevamente negativo, verifica si los datos recibidos se corresponden a un cambio en la velocidad de desplazamiento del texto, cuando las tres comprobaciones anteriores resultan negativas, el texto recibido por bluetooth se escribe en el display y se reinicia el bucle. Si alguna de las tres comprobaciones antes mencionadas (color, brillo o velocidad) resulta positiva, el microcontrolador realiza la modificación pertinente, reinicia el bucle y verifica la adquisición de nuevos datos provenientes desde la aplicación del celular, a través de bluetooth.

2.2. *Diseño de la aplicación para el celular*

El objetivo de la aplicación desarrolla es brindar una interfaz sencilla para el usuario del display, que le permita modificar el texto mostrado en el mismo y agregar algunos efectos tales como el color, la velocidad de desplazamiento y el brillo. El diagrama de flujo de la aplicación se presenta en la Fig. 2.

Como puede observarse en la Fig. 2, al iniciar, el código consulta si la comunicación bluetooth del celular está activada para seguir con el resto del programa. Una vez que la misma se encuentra activada, se consulta si el usuario activó el botón para la conexión del celular con el módulo bluetooth disponible en el display, esto es para que el celular puede entablar conexión con el mismo. A continuación, se procede a realizar una serie de preguntas referentes a las configuraciones que se realizan al display, como ser: envío de texto, color, velocidad, brillo, etc. Todas estas preguntas, si son negativas, pasan a la siguiente pregunta Mientras que si son positivas, antes de pasar a la siguiente pregunta, envían la configuración requerida al display donde se encuentra el microcontrolador y el módulo bluetooth HC-05.

La aplicación es elaborada con el entorno de desarrollo “App Inventor-2” [11] y puede utilizarse en teléfonos celulares inteligentes que poseen sistema operativo Android y disponen de bluetooth. Este entorno posee una programación del tipo gráfica, lo cual facilita el desarrollo del código correspondiente a la aplicación, ya que las instrucciones y bucles se ordenan por bloques. El entorno también dispone de un bloque para la conexión bluetooth, resultando esto muy práctico en aplicaciones que requieren comunicación a través de este protocolo.

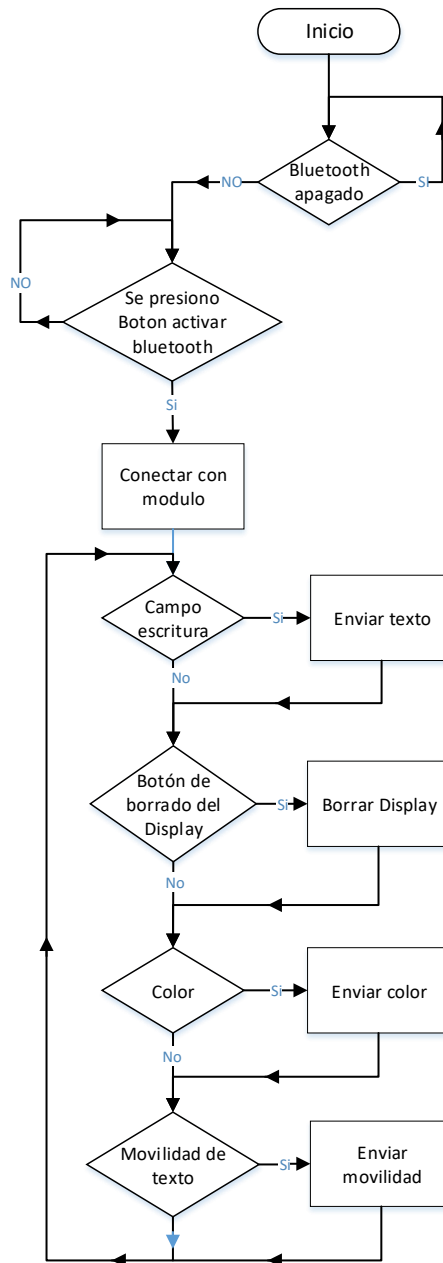


Fig. 2 – Diagrama de flujo para la aplicación del celular.

El entorno mencionado permite construir la aplicación tanto en su funcionalidad como en su aspecto estético. Para esto el desarrollador debe permutar entre dos ventanas de trabajo, una donde se modifica el aspecto de la aplicación (lo que verá el usuario en el celular) y otra asociada a las acciones que son ejecutadas al presionar un botón o ejecutar alguna opción programada de la aplicación. Una vez terminada la programación de la aplicación, App Inventor-2 genera un archivo con extensión “apk”, siendo este el que debe ejecutar en el sistema operativo Android del celular.

2.3. Diseño del circuito de control

El circuito de control está conformado por el Arduino UNO y una placa diseñada para insertarse en los bornes que posee esta plataforma. La placa tiene la finalidad de soportar al módulo HC-06 que establece la comunicación vía bluetooth con el microcontrolador del Arduino UNO y también incorpora los bornes de conexión, tanto de la fuente de alimentación como los correspondientes al display (alimentación y datos). Esta placa también posee llaves a través de las cuales se puede conectar/desconectar el módulo HC-06 y el display del Arduino UNO ante una eventual reprogramación de su microcontrolador. La Fig. 3 muestra el circuito impreso de la placa mencionada.

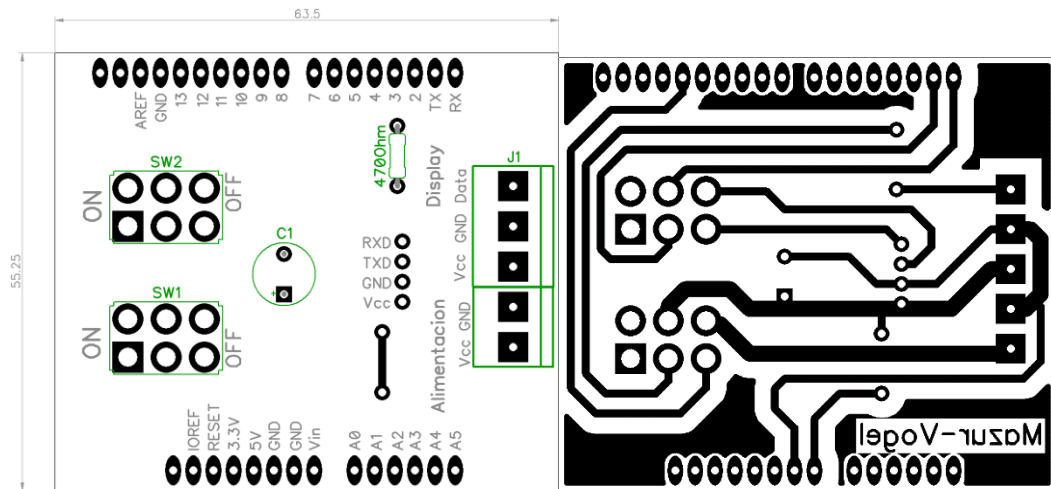


Fig. 3 – Circuito impreso de la placa del circuito de control.

La placa mostrada en la Fig. 3 también conecta el pin de datos del microcontrolador a la entrada de datos del primer led del display, mediante un resistor de 470 Ohm lo indica la hoja de datos del led RGB inteligente utilizado [3].

3. Resultados

La Fig. 4 presenta el display construido a partir de los leds inteligentes y sobre el cual se desarrolló este trabajo. Cabe mencionar que el mismo fue provisto en la asignatura, siendo su construcción realizada un grupo de alumnos que cursaron la asignatura años anteriores.



Fig. 4 - Display con matriz de leds inteligentes.

En la Fig. 5 puede observarse la placa del circuito de control, la cual se ha desarrollado para energizar al Arduino UNO y al módulo bluetooth, así como también para realizar la interconexión entre Arduino, display y módulo HC-06, estableciendo la comunicación entre estos dispositivos.

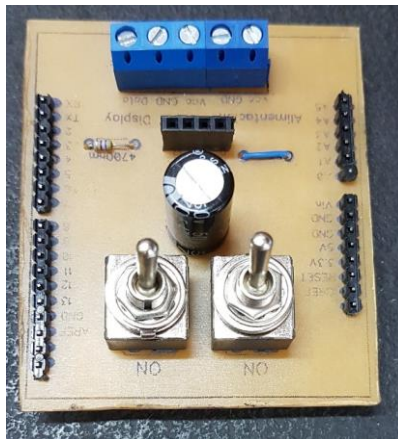


Fig. 5 - Foto de la placa del circuito de control.

La aplicación de celular desarrollada en este trabajo para la programación del display, presenta el aspecto indicado en la Fig. 6. El entorno gráfico presentado en esta figura, muestra diversos botones. De arriba hacia abajo, se observa los botones para establecer la conexión/desconexión con el modulo bluetooth, junto con una indicación de cuál es el estado actual de la conexión. Debajo existe un campo de escritura donde el usuario escribe el texto alfanumérico a visualizar en el display, que luego se envía a través de bluetooth accionado el botón “Enviar”. Asociado a esto se encuentra el botón para borrar el texto escrito en el campo de texto.



Fig. 6 - Pantalla correspondiente al menú de la aplicación para el celular.

Debajo del campo de escritura, se encuentra un botón que borra directamente el texto en el display. Luego de este botón, se observa el área llamada “Configuración de color”, donde se observan botones que permiten el cambio de color de los leds en el display. Más abajo aparece una sección para el movimiento del texto en el display, donde se tiene tres opciones de velocidad: “Lento”, “Medio” y “Rápido”. Por último, en la parte inferior de la pantalla hay cuatro botones (“1”, “2”, “3” y “4”) para configurar la intensidad luminosa de los leds. Cada botón se corresponde con un brillo distinto, siendo 1 el más tenue y 4 el más brillante.

Como vemos, la aplicación desarrollada para programar al display de leds, es fácil de utilizar debido a la disposición y tipo de botones implementados.

4. Conclusiones

La situación problemática planteada en la actividad integradora pudo resolverse a través de la implementación de un sistema embebido que utiliza el microcontrolador ATmega 328p disponible en la placa Arduino UNO. Esta implementación pudo lograrse gracias a los conocimientos adquiridos sobre la programación de microcontroladores y al estudio de los mismos en la asignatura “Técnicas Digitales 2”. La experiencia no sólo permitió aplicar las capacidades adquiridas, sino que también resultó enriquecedora ya que posibilitó el aprendizaje sobre tecnologías ampliamente utilizadas en el desarrollo de sistemas embebidos (por ejemplo: comunicación bluetooth y aplicaciones Android).

Por otra parte, en el desarrollo de este trabajo se utilizaron librerías especiales para la programación de los leds en arreglos matriciales. Si bien esto facilitó el manejo del texto visualizado en el display (para la modificación de efectos como el color, movimiento y brillo), fue necesario comprender correctamente el funcionamiento de las mismas para desarrollar el programa embebido en el microcontrolador.

La incorporación de la placa del circuito de control, permite la reprogramación del microcontrolador sin necesidad extraerlo. Además, la misma provee un soporte para las conexiones y para el módulo bluetooth.

Por otra parte, se obtuvo una aplicación para celular que resulta ser intuitiva y fácil de utilizar para el usuario del sistema. También es de destacar que el uso del entorno App-Inventor 2, facilita la implementación de la misma.

5. Referencias

- [1] Atmel Corporation. (2016). “Atmega 328p Datasheet Complete”. California, EU.
- [2] Arduino. (s.f.). Introduction to the Arduino Board. Recuperado de <https://www.arduino.cc/en/Reference/Board>
- [3] WS2812. (s.f.). Intelligent control LED integrated light source. Recuperado de <https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>
- [4] Guangzhou HC Information Technology Co., Ltd. (s.f.). Product Data Sheet. Recuperado de <https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>
- [5] Wikipedia. (s.f.). Bluetooth. Recuperado de <https://es.wikipedia.org/wiki/Bluetooth>
- [6] Adafruit Industries. (2018). Arduino library for controlling single-wire LED pixels. San Francisco, EU.: GitHub. Recuperado de https://github.com/adafruit/Adafruit_NeoPixel
- [7] Adafruit Industries. (2018). WS2812 FX Library for Arduino and ESP8266. San Francisco, EU.: GitHub. Recuperado de <https://github.com/kitesurfer1404/WS2812FX>
- [8] Adafruit Industries. (2018). Arduino library for addressable RGB LED strips from Pololu. San Francisco, EU.: GitHub. Recuperado de <https://github.com/pololu/pololu-led-strip-arduino>
- [9] Adafruit Industries. (2018). Adafruit enhanced NeoPixel support library. San Francisco, EU.: GitHub. Recuperado de <https://github.com/Makuna/NeoPixelBus>
- [10] Paul Stoffregen. (s.f.). OctoWS2811 LED Library. Recuperado de https://www.pjrc.com/teensy/td_libs_OctoWS2811.html
- [11] Google Labs (2008). App Inventor. Recuperado de <http://ai2.appinventor.mit.edu/>