

Reporte de la Práctica 1 del Laboratorio de Principios de Mecatrónica

Andrea Marín Alarcón
Luis Felipe Landa Lizarralde
Miguel González Borja

Resumen—En esta práctica se plantea un primer acercamiento a la terminal de Linux, Git y Arduino. Aquí se realizan ejercicios sencillos utilizando las tres tecnologías anteriores: la creación y eliminación de directorios desde la terminal, la creación de un repositorio Git asociado a un repositorio en Github y la implementación de tanto la simulación *blink* como un programa que acepte de entrada un número en hexadecimal y lo despliegue en un display de 7 segmentos. La práctica exhibe como resultado un repositorio en Github que contiene el código necesario para la implementación de los dos programas con Arduino mencionados anteriormente.

1. INTRODUCCIÓN

Estar familiarizado con los elementos básicos de tecnologías como la terminal de Linux, Git y Arduino permite desarrollar proyectos de mayor complejidad a futuro. Son herramientas ubicuas en ciertos ámbitos del desarrollo de software y tener la capacidad de utilizarlas es de importancia fundamental. Es esencial tener los fundamentos de estas tecnologías firmes para ejercer buenas prácticas de desarrollo como estructuración de código y versionado.

El entorno que ofrece la terminal de Linux es más versátil y útil que su contraparte en Windows y requiere una atención especial pues permite ejecutar una gran variedad de comandos. La terminal funge como un puente entre el usuario y el sistema operativo y se usa para expedir ciertas tareas como la creación de directorios y la navegación entre ellos. También se puede utilizar para ejecutar programas y se le pueden agregar comandos que amplíen su utilidad e.g. con Git.

Git es una herramienta particular para el control de versiones. Aunque particularmente útil para el desarrollo de código, sus usos se extienden a cualquier tipo de archivos que se editen periódicamente y se requiera revertir a versiones anteriores en cualquier momento. El uso de herramientas que permitan el control de versiones facilitan trabajar sobre código y se considera buena práctica de desarrollo de software. Con proyectos sencillos y con pocos componentes el uso de Git puede parecer excesivo pero, al escalar la complejidad del proyecto, igual escala su utilidad. También se puede acoplar con repositorios que no residen en la computadora usando Github u otros servicios y de esta manera distribuir aún más el versionado.

La capacidad de Arduino de percibir entradas del mundo exterior y convertirlas en *outputs* que controlan otros dispositivos

los hacen en una plataforma electrónica altamente versátil. Al ser programable, el espectro de situaciones en las que puede ser utilizado es amplio y permite la interacción entre software y el mundo físico.

Este reporte describirá a mayor detalle las tecnologías utilizadas en la práctica; se detallará el proceso que se siguió; presentará los resultados de la práctica; expondrá las conclusiones individuales de los participantes al igual que la capacidad en la que estuvieron involucrados en el desarrollo de la práctica.

2. MARCO TEÓRICO

En esta práctica se utilizaron tres tecnologías principales: Git, la terminal de Linux y un arduino.

Git es un sistema de control de versiones distribuido, es decir, registra los cambios realizados en un conjunto de archivos a lo largo del tiempo, de manera que se puedan recuperar versiones anteriores de los mismos. Una de las ventajas de este tipo de sistemas es que, en caso de cometer algún error, puedes revertir un archivo o el proyecto entero a un estado anterior. Además, puedes comparar cambios a lo largo del tiempo y ver quién modificó los archivos. Al ser un sistema distribuido, los clientes no sólo descargan una copia de la última versión del repositorio, sino que lo replican completamente. [1]

También se utilizó la terminal de Linux, la cual es una herramienta muy poderosa, pues te permite realizar todo tipo de tareas mediante comandos de texto y sin la necesidad de utilizar la interfaz gráfica del sistema operativo. En pocas palabras, la terminal es un intérprete de órdenes que hace de interfaz entre el usuario y el sistema operativo. Es por esto que cuando se utiliza la terminal, la velocidad de ejecución de los comandos aumenta, ya que, te comunica directamente con el sistema. [2] En esta práctica sólo se utilizaron comandos básicos de la terminal, los cuales nos permiten trabajar con los directorios y archivos.

Por último, Arduino es una plataforma de hardware y software libre, basada en una placa con un microcontrolador y un entorno (IDE) de desarrollo. Este entorno implementa el lenguaje de programación de arduino y el bootloader que permite cargar código en la placa. Como ya se mencionó, la parte del hardware del arduino consiste en una placa con todos los elementos necesarios para conectar periféricos a

las entradas y salidas de un microcontrolador. Un microcontrolador es un circuito integrado programable, capaz de ejecutar las instrucciones grabadas en su memoria. Incluye las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida. [3]

El **arduino** contiene diferentes tipos de pins, algunos de ellos son [4]:

- Análogo: pueden leer la señal de un sensor analógico y convertirla a un valor digital
- Digital: se pueden utilizar, tanto para **inputs**, como para **outputs** digitales.
- PWM: se usan para modulación por ancho de pulso con lo cual, se puede simular una salida analógica.

Por último, el IDE de Arduino consiste en un editor de código, un compilador, un depurador y una interfaz gráfica. Además, incluye herramientas para cargar el programa ya compilado en la memoria flash del hardware.

3. DESARROLLO

Se desarrollaron dos sistemas para Arduino. El primer sistema debía encender un LED si el valor de una resistencia estaba debajo de un valor determinado. Además, si se presionaba un botón, el LED ~~debía de~~ cambiar de estado. Para esto, se conectó una fotoresistencia al Arduino como se muestra en la Figura 1, y despues se determinaba el valor correspondiente para el LED.

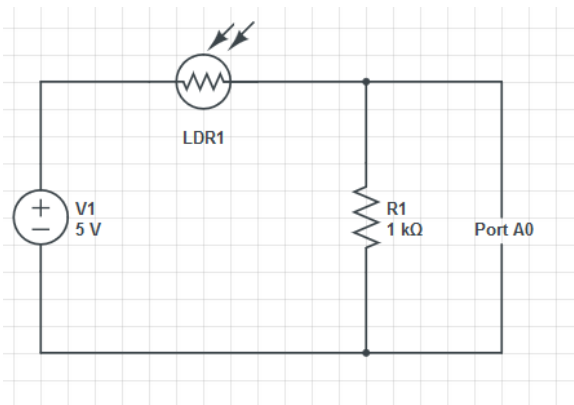


Figura 1. Conexión de la fotoresistencia al Arduino

Para el segundo sistema, se conectó un display de siete segmentos al Arduino como se muestra en la Figura 2.

Después, el Arduino recibía dígitos hexadecimales (0-F) a través del puerto serial y los desplegaba en el display.

4. RESULTADOS

El repositorio del la práctica puede ser encontrado **aquí**. Este incluye el código para los dos sistemas desarrollados, así como un log de todas las acciones tomadas en la terminal.

Para el primer sistema, se configuró el Arduino como se muestra en el Cuadro 1.

En la Figura 3 se puede ver este funcionando. El LED se enciende cuando la fotoresistencia da un valor menor a **80**. Además, el botón funciona como un switch para el LED.

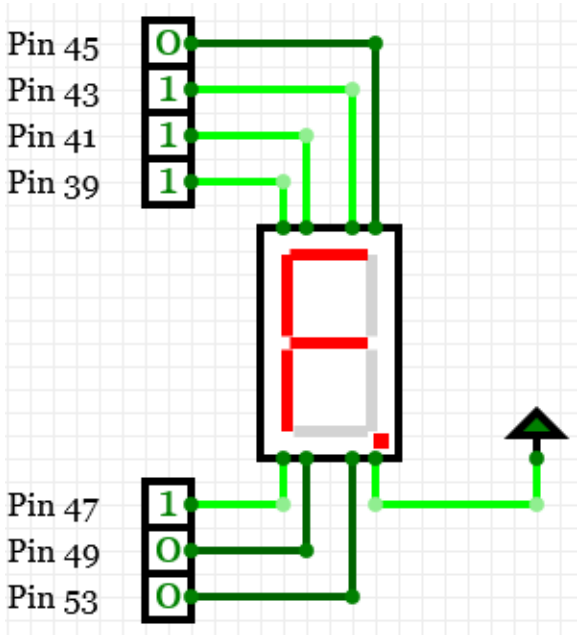


Figura 2. Conexión del 7 segmentos al Arduino

Pin	I/O	Elemento
A0	Input	Fotoresistencia
30	Input	Botón
13	Output	LED

Cuadro 1
Configuración del sistema 1

Para el segundo sistema, el Arduino se configuró como muestra el Cuadro 2.

Se utilizo un display de ánodo común, que se puede ver funcionando en la Figura 4.

5. CONCLUSIONES

El Arduino permite configurar circuitos de manera mucho más simple que el FPGA que se utilizo anteriormente. Además, el uso de la terminal y git permite llevar un mejor control sobre el proyecto.

-Miguel González Borja

El uso de los diferentes comandos disponibles en la terminal agilizan el desarrollo del proyecto. El comando man permite un rápido acceso a la documentación de los comandos y constituye una gran ayuda. Git y Github hacen que el proyecto sea consistente y portable a otras computadoras en poco tiempo. El Arduino presenta una plataforma sencilla de usar con posibilidad de usos de gran complejidad.

-Luis Felipe Landa Lizarralde

Herramientas como Git y la terminal son muy útiles en el desarrollo de proyectos. Por un lado, Git permite tener control sobre las versiones del proyecto además de que lo hace portable, lo que hace muy fácil revertir errores y trabajar en equipo. Además, la terminal es una buena herramienta para administrar los archivos del sistema de manera eficiente. Comandos como cp y mv permiten

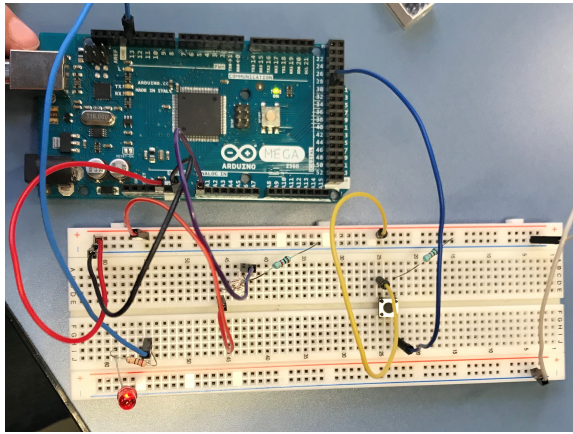


Figura 3. Primer Sistema

Pin	I/O	Elemento
43	Output	Segmento 'a'
45	Output	Segmento 'b'
53	Output	Segmento 'c'
49	Output	Segmento 'd'
47	Output	Segmento 'e'
41	Output	Segmento 'f'
39	Output	Segmento 'g'

Cuadro 2
Configuración del sistema 2

modificar archivos en tan sólo dos líneas sin la necesidad de abrir varias ventanas en la interfaz. Por último, el Arduino es una plataforma que permite crear diferentes proyectos de electrónica de manera simple. -Andrea Marín Alarcón

6. ROL

Miguel González Borja: Armado de circuitos de los sistemas 1 y 2. Secciones de desarrollo y resultados.

Andrea Marín Alarcón: Código de los sistemas 1 y 2. Sección de marco teórico.

Luis Felipe Landa Lizarralde: Uso de la terminal de Linux y código. Sección de introducción.

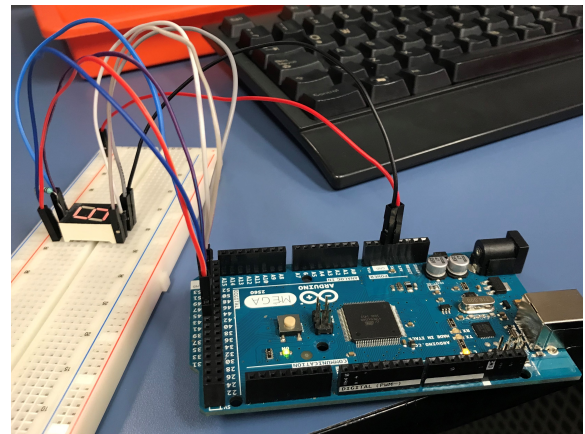


Figura 4. Segundo Sistema

7. FUENTES CONSULTADAS

Este formato fue utilizado en la elaboración de esta práctica.

REFERENCIAS

- [1] S. B. Chacon S. Empezando - fundamentos de git. [Online]. Available: <https://git-scm.com/book/es/v1/Empezando-Fundamentos-de-Git>
- [2] J. William E. Shotts. What is "the shell". [Online]. Available: http://linuxcommand.org/lc3_lts0010.php
- [3] Qué es arduino. [Online]. Available: <https://aprendiendoarduino.wordpress.com/2016/09/25/que-es-arduino/>
- [4] What is an arduino? [Online]. Available: <https://learn.sparkfun.com/tutorials/what-is-an-arduino/all>