

# Práctica 1

## Introducción a Arduino

Daniela Arely Morales Hernández  
Stephanie Lizeth Malvaes Diaz

### *Abstract*

Como primer acercamiento al laboratorio de Principios de Mecatrónica realizamos algunos experimentos que sirven para refrescar el conocimiento adquirido en cursos anteriores respecto al armado de circuitos eléctricos, fórmulas y cálculos teóricos, así como comparación con datos obtenidos prácticamente.

### *I. Objetivos*

El objetivo de la práctica es familiarizarse y aprender a usar comandos del sistema operativo Ubuntu basado en GNU/Linux, y con el sistema de archivos, aprender a administrar proyectos de software utilizando el entorno GIT, conocer programación de microcontroladores mediante Arduino.

### *II. Introducción*

Arduino está compuesto por una placa AVR de la familia de microcontroladores RISC de la compañía de semiconductores ATMEL. La forma en la que el sistema trabaja es mediante la recepción de datos a través de sus pines analógicos y digitales, los cuales pueden ser conectados a una gran cantidad de sensores, motores u otros actuadores para enviar y recibir información.

**GIT:** es un software de control de versiones, pensado para en la eficiencia y el manejo confiable de versiones de

aplicaciones cuando tienen muchos archivos de código.

**UBUNTU:** es un sistema operativo de código abierto para computadoras. Se basa en la arquitectura de Debian. Está orientado al usuario promedio, enfocado en la experiencia de usuario y facilidad de uso.

Los comandos de linux utilizados fueron:

**pwd:** (personal working directory) imprime el nombre del directorio actual

**mkdir:** (make directory) crea un nuevo subdirectorio o carpeta del sistema de archivos. Su sintaxis es: **mkdir {nombre\_de\_la\_carpeta}**

**cd:** (change directory) permite moverse entre directorios del sistema.

**ls:** lista los archivos y directorios de la carpeta actual.

**rmdir:** (Remove directory) elimina carpetas vacías. Su sintaxis es: **rmdir {nombre\_carpeta}**

**rm:** (Remove) elimina archivos. Su sintaxis es **rm {nombre\_del\_archivo}**

**touch:** crea nuevos archivos en el directorio actual.

**cp:** (Copy) copia archivos o directorios. Su sintaxis es **cp {origen} {destino}**

**mv:** (Move) mueve directorios o archivos de una ubicación a otra. Su sintaxis es:

**mv{ubicación\_actual}{nueva\_ubicación}**

**less:** permite consultar páginas hacia atrás y hacia adelante.

more: se encarga de paginar texto, mostrando una pantalla cada vez

cat: muestra el contenido de un fichero dado

head: Se muestran las diez primeras líneas.

tail: Imprime las diez últimas líneas de un fichero. Ejemplo: `tail -f -n 20 /var/log/httpd/httpd.log`

ps: informa del estado de los procesos. Nos muestra una instantánea de los procesos actuales

top: se encarga de mostrar una lista de los principales procesos y su uso de CPU entre otros valores.

man: para consultar el manual, se escribe man y el comando del que quieres información

### III. Desarrollo

La primera parte del laboratorio consistió en realizar diferentes operaciones en la terminal de Linux para familiarizarse con los conceptos y comandos que se utilizan para poder realizar distintos procedimientos. Al acceder a GitHub creamos y clonamos un repositorio donde se guardaron las carpetas que eliminamos, movimos y cambiamos de nombre directorios y archivos. Al final logramos ver los cambios ejecutados.

Se realizó la estructura de carpetas en Linux mediante la terminal, se eliminó el archivo Arc2A y posteriormente el directorio DIR\_2AB, se creó una copia el directorio DIR\_2AA y se cambió el nombre del directorio copia a Dir\_X. Se creó una cuenta en GITHUB y se agregó un nuevo repositorio con el nombre PM\_Practica\_1 Luego se clonó el directorio DIR\_PM\_01 y su contenido dentro del repositorio. Mediante github se añadió un archivo al directorio DIR\_PM\_

Para la parte de arduino ejecutamos el código del blink el cual prendía intermitentemente el LED conectado al cuerpo especificado. El siguiente inciso consistía en lograr que un led de la placa AVR se prendiera y apagara al sobrepasar una frecuencia determinada usando como sensor una fotoresistencia. En esta parte logramos entender la manera en que se inicializan variables y la manera en que son empleadas cada una en el código. Para su conexión usamos un fotoresistor, un led y dos resistencias diferentes para cada elemento. Se conectaron a los pines del Arduino y con el comando *digitalAnalog()* se leía la señal del fotoresistor. Agregamos un botón que se conectó a una resistencia de 1000 ohms, para que cambiara el estado del LED sin importar la intensidad de la luz recibida por el fotoresistor.

Se sustituyó el control por valor preestablecido por un control manual a través de un potenciómetro y se observaron los valores en el monitor de arduino.

### IV. Resultados

Se probaron los comandos pwd, mkdir, cd, ls, rmdir, rm, touch, cp, mv, less, more, cat, head, tail, ps, top y man.

Se realizó la creación de carpetas y archivos por medio de linux la primera carpeta se puede sintetizar así: [Practica 1/PM\\_01/Practica1/DIR\\_1A/Arc1A](#) y se subió a GITHUB

La carpeta de Practica5 se sintetiza así: [PM\\_Practica\\_1/PM\\_01/Practica5/DIR\\_5A/Arc5A](#)

También se realizaron los movimientos de la carpeta práctica 2, se eliminó el Arc2a y se cambió el nombre de DIR\_2AA por Dir\_X

La simulación mediante el led se encendió y apagó según las especificaciones sobre luminiscencia que recibía el fotoresistor y se sustituyó efectivamente el valor requerido por el potenciómetro, controlando mediante este los cambios en la luminiscencia del led. Se apreció en el monitor serial el cambio de valores cuando se movía manualmente el potenciómetro.

### *V. Conclusiones*

Arely:

Con esta práctica pudimos investigar y aprender sobre la forma de programar, los comandos y el lenguaje de Arduino. Es una buena introducción para aprender a usarlo. Durante el proceso tuvimos varios tropiezos con errores que al principio no entendíamos y poco a poco los fuimos investigando y solucionando.

Stephanie:

Esta práctica esclareció el tema de los sistemas embebidos, al realizar funciones dedicadas, viendo como se comporta un sistema en tiempo real, usando señales tanto analógicas como digitales, usando un microcontrolador, y recordando los circuitos electrónicos. Además la práctica introdujo el uso de los comandos de linux, y el uso de la terminal, lo que es de suma utilidad. Todo esto me permitió ver la gran utilidad de realizar sistemas de menor complejidad, pero que requieren de sensores y actuadores y que sean usables en la vida diaria.

### *VI. Referencias*

Arduino desde cero. (2019). Retrieved 15 November 2019, from [tutoriales?download=1:manual-de-programacion-de-arduino](#)

10 comandos Linux que debes conocer. (2019). Retrieved 15 November 2019, from <https://ed.team/blog/10-comandos-linux-que-debes-conocer>

Los 42 comandos más importantes en GNU/Linux - ochobitshacenunbyte. (2019). Retrieved 15 November 2019, from <https://www.ochobitshacenunbyte.com/2018/12/04/los-42-comandos-mas-importantes-en-gnu-linux/>