

# Práctica 1. Microcontroladores y Manejo de Versiones

Emiliano Sotomayor González  
155902  
uboat46@gmail.com

Jaime Limón Samperio  
162395  
jalisa2808@gmail.com

**Abstract**—El manejo del intérprete de comandos de alguna distribución del sistema operativo Linux y el conocimiento de su sistema de archivos facilitan el desarrollo de distintas tareas que se presentan en el día a día en el desarrollo de firmware. Manejar además un controlador de versiones como GIT permite realizar código de una manera eficiente y controlada. Como se observa en la práctica, el uso de estas herramientas y el uso de la plataforma Arduino nos permite desarrollar firmware de una manera más eficiente y segura.

## I. INTRODUCCIÓN

El sistema operativo Ubuntu es una distribución de libre acceso basada en la tecnología de software conocida como GNU, al ser un ambiente del tipo Linux ofrece una variedad de tareas que el usuario puede ejecutar a través de la terminal de comandos. Es de particular interés para el curso el manejo de ROS (Robot Operating System) y sus librerías más importantes.

El sistema de control de versiones GIT es una herramienta de software que permite a los desarrolladores de código fuente llevar la administración de los cambios realizados, especialmente cuando un mismo proyecto se compone de las aportaciones de un amplio grupo de trabajo que produce una gran cantidad de archivos y documentación. La plataforma GITHUB ofrece a los usuarios un entorno amigable de creación, transferencia y administración de archivos fuente. Es recomendable utilizar una herramienta de este tipo para los proyectos generados durante el curso.

La plataforma de prototipado rápido y programación conocida como Arduino ofrece un lenguaje basado en C/C++ con funciones optimizadas para implementarse en las tarjetas de desarrollo propietarias que generalmente incluyen un microcontrolador AVR. Arduino constituye una alternativa confiable y versátil como base o apoyo para la creación prototipos académicos embebidos de baja y mediana complejidad.

## II. MARCO TEÓRICO

Utilizando los distintos comandos de la terminal de linux se pueden crear amplios sistemas de archivos que permitan manejar proyectos de desarrollo de software.

Mediante el uso de un manejador de versiones como GIT, manejar estos sistemas de archivos se facilita. El

poder revertir los cambios hechos al sistema y poder progresivamente agregar cambios permite un control refinado sobre lo sucedido en cualquier proyecto.

Además, utilizando el software proporcionado por la plataforma Arduino, se pueden realizar proyectos de firmware de media y baja complejidad. Utilizando esta plataforma se desarrollará un pequeño programa que nos permita controlar la intensidad luminosa de un led dado ciertos parámetros analógicos, tales como la luz o una resistencia variable (potenciometro).

Se utilizará lo siguiente:

- Sistema Operativo Ubuntu 16.04 Xenial (recomendado) o superior
- Tarjeta de desarrollo Arduino Mega ®
  - Microcontrolador - AVR ATmega2560
  - Voltaje de operación - 5V
  - Voltaje de alimentación - 7-12V
  - Puertos digitales de entrada/salida - 54 (15 con capacidades PWM)
  - Puertos de entrada analógica - 16
  - Corriente límite en puertos de e/s - 20 mA
  - Memoria flash - 256 KB (8 KB para bootloader)
  - SRAM/EEPROM - 8 KB / 4KB
  - Velocidad de reloj - 16 MHz
- Potenciometro
- Led
- Resistencias
- Fotorresistencia
- Cables de alambrado

## III. DESARROLLO

Se deberán de seguir las siguientes instrucciones:

- Abrir una instancia de la terminal de Linux y practicar ejecutando apropiadamente cada uno de los siguientes comandos:  
pwd, mkdir, cd, ls, rmdir, rm, touch, cp, mv, less, more, cat, head, tail, ps, top, man.  
Reportar para cada uno la función que cumplen y/o los argumentos o datos que se deben proporcionar para su funcionamiento.

- Crear la estructura de directorios y archivos mostrada en el laboratorio. Reportar a manera de script la secuencia de comandos necesarios para crear la estructura.
- Eliminar el archivo Arc2A y posteriormente el directorio DIR\_2AB, crear una copia el directorio DIR\_2AA, cambiar el nombre del directorio copia a Dir\_X.  
NOTA: Al final el ejercicio, realizar los ajustes pertinentes para que los distintos archivos y directorios del repositorio correspondan al material del curso.
- Crear una cuenta en GITHUB para fines del curso.
- Crear un nuevo repositorio con el nombre PM\_Practica\_1
- Clonar el directorio DIR\_PM\_01 y su contenido dentro del repositorio.
- A través del portal de GITHUB, añadir al directorio DIR\_PM\_01 un archivo de texto conteniendo el nombre de los integrantes del equipo. El nombre del archivo corresponde al nombre del equipo (elegir un nombre creativo para todo el curso e incluirlo en el reporte de la práctica).
- Acceder al archivo de texto por medio de la consola de Linux e incluir los números de cuenta de los integrantes del equipo.
- Iniciar el software de Arduino y cargar el código de ejemplo Blink
- Duplicar la frecuencia de oscilación del diodo LED integrado en la tarjeta de desarrollo.
- Simular mediante un diodo LED externo el encendido automático de un sistema de iluminación cuando se llega a un valor preestablecido de oscuridad medido a través un sensor fotorresistivo
- Sustituir el control por valor preestablecido por un control manual a través de un potenciómetro.
- Desplegar los valores de oscuridad (establecido por el potenciómetro y medido en el sensor) a través del Monitor Serial. Ajustar estos valores para que se presenten en una escala del 0 al 100%.
- Añadir los códigos generados a un repositorio en la cuenta GIT del equipo.

#### IV. RESULTADOS

El intérprete de comandos de las distribuciones del sistema operativo Linux nos ofrece una serie importante de programas que nos facilita su uso. Algunos de los comandos de dicho intérprete son:

- pwd: lista el directorio activo
- mkdir: crea un nuevo directorio dado un nombre de directorio
- cd: cambia de directorio dado una ruta al directorio requerido
- ls: lista los contenidos de un directorio dado una ruta del directorio
- rmdir: remueve un directorio dado su ruta
- rm: remueve un archivo dado su ruta
- touch: crea un archivo dado su nombre
- cp: copia un archivo o directorio dado el objeto a copiar y la dirección destino
- mv: mueve un archivo o directorio dado el objeto a mover y la dirección destino
- less: un programa que permite observar los contenidos de un archivo de manera paginada dado un nombre de archivo (distinto a more en el sentido que no lee todo el archivo de una vez y además ofrece otras funciones)
- more: un programa que permite observar los contenidos de un archivo de manera paginada dado un nombre de archivo
- cat: concatena e imprime archivos dado un nombre de archivo
- head: permite observar las primeras líneas de un archivo dado su nombre
- tail: permite observar las últimas líneas de un archivo dado su nombre
- ps: permite observar el status de los procesos del sistema
- top: imprime de manera periódica un listado ordenado de algunos de los procesos del sistema
- man: imprime y le da formato a página de manual sobre los distintos programas del intérprete dado el nombre de un programa

Dados estos comandos y el programa git, pudimos realizar el sistema de archivos solicitado, así como crear un control de versiones del mismo y subirlo a una plataforma en línea (Github) que se dedica a manejar repositorios de archivos que utilizan el sistema GIT. Además, creamos un programa que cumplía con las especificaciones dadas para el microcontrolador. Utilizando la instrucción delay(tiempo en milisegundos); Pudimos variar el tiempo en que el led interno de la tarjeta funcionaba, así como utilizar distintos comandos como analogRead y analogWrite para poder leer señales analógicas o otorgar un voltaje al led.

#### V. CONCLUSIONES

Desarrollar software hoy en día puede volverse más sencillo conociendo las herramientas correctas. Por un lado, un sistema de control de versión es más que necesario. El esquema de quince archivos nombrado de la misma manera diferenciados por un carácter es cosa del pasado. Un simple comando de terminal puede recuperar y/o deshacer cambios o bien, crear una versión alterna dentro del mismo repositorio. Un control de versiones potenciado con un software en la nube (Github, Gitlab, Bitbucket) brinda muchas más opciones para un trabajo en equipo. Para un manejo correcto y bien aprovechado de las herramientas antes mencionadas, es sumamente recomendable dominar los comandos más comunes en las distribuciones LINUX. Es posible que algunas herramientas no cuenten con un GUI público y la única opción sea a través de un CLI.

#### VI. REPOSITORIO

<https://github.com/uboa46/principios-de-mecatronica.git>

#### REFERENCES

- [1] "Linux man pages" Internet: "<https://linux.die.net/man/>"
- [2] "Arduino reference" Internet: "<https://www.arduino.cc/reference/en/>"
- [3] "GIT" Internet: "<https://git-scm.com/>"