

Práctica 1. Microcontroladores y Manejo de Versiones

1. Objetivos

- Hacer un reconocimiento inicial de la utilidad y propiedades de la terminal de comandos del sistema operativo Ubuntu basado en GNU/Linux. Analizar y comprender el sistema de archivos.
- Adoptar una estrategia de administración de proyectos de software y familiarizarse con las ventajas y procedimientos del sistema control de versiones en el entorno de GIT.
- Conocer las bases de la programación de microcontroladores a través de la plataforma Arduino.



2. Antecedentes

- 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.

3. Aspectos técnicos

- Sistema Operativo Ubuntu 16.04 Xenial (recomendado) o superior
- ROS 10 Kinetic Kame (recomendado)
- Características de la 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

4. Procedimientos

LINUX



- 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 Figura 1.
Reportar a manera de script la secuencia de comandos necesarios para crear la estructura.

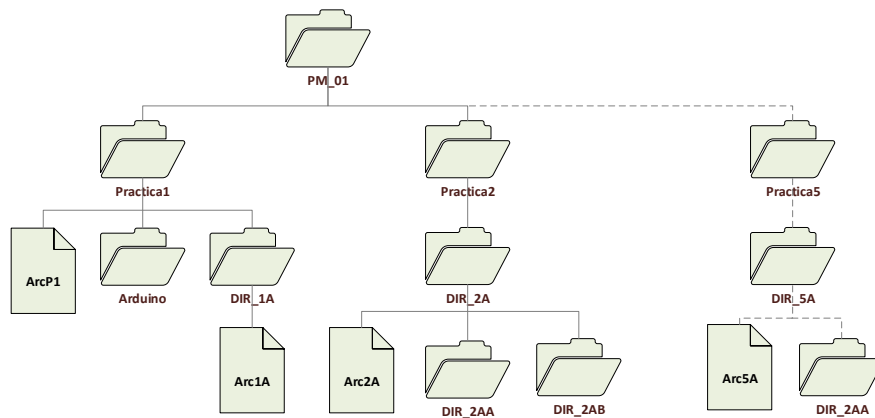


Figura 1. Directorios y Archivos.

- 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.

GITHUB



- 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.

ARDUINO



- 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 (Figura 2).
- Sustituir el control por valor preestablecido por un control manual a través de un potenciómetro.

- m. 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%.
- n. Añadir los códigos generados a un repositorio en la cuenta GIT del equipo.

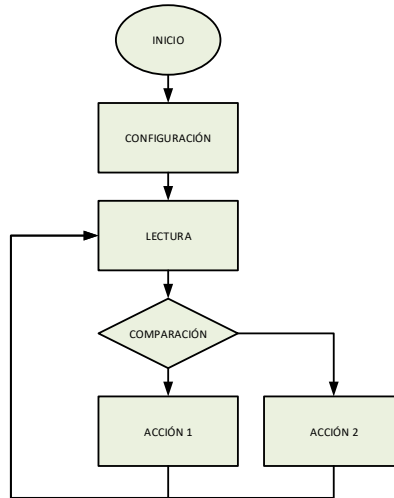


Figura 2. Diagrama de flujo básico.

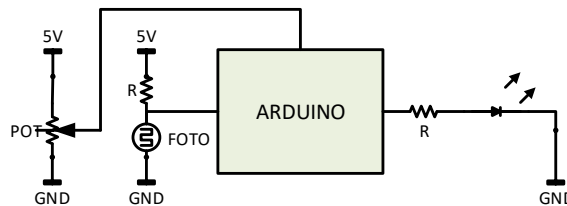


Figura 3. Conexiones básicas.

5. Estructura del reporte

El reporte de la práctica de laboratorio debe incluir bajo el formato a dos columnas de conferencia IEEE (https://www.ieee.org/conferences_events/conferences/publishing/templates.html) lo siguiente:

Resumen (abstract), Introducción, Marco Teórico, Desarrollo, Resultados, Conclusiones y Referencias

Incluyendo los puntos resaltados durante los procedimientos de la práctica.

6. Referencias

- Linux man pages <https://linux.die.net/man/>
- Vim cheat sheet <https://vim.rtorr.com/>
- Arduino reference <https://www.arduino.cc/reference/en/>
- Git <https://git-scm.com/book/en/v2>