

# Practica No.1 GNU/Linux Microcontrolador

Leslie P. Brenes, Diego Villafuerte

**Resumen**—El propósito de esta práctica fue intrductorio en los comandos de Linux así como del funcionamiento de Arduino y GIT. La práctica se dividió en 3 partes: uso de git y comandos de linux básicos para administrar archivos y versiones, diseño de circuitos básicos, y uso de la consola de arduino.

## 1. INTRODUCCIÓN

Esta fue la primera práctica del semestre, la cual consistió de dos sesiones. En la primera sesión el objetivo principal fue dar a conocer los criterios de evaluación, el calendario de prácticas y una breve introducción al uso de diferentes comandos en la terminal de Linux. Durante la segunda sesión se hizo entrega del material y la caja de laboratorio. Para esta materia, se nos entregó un Arduino el cual consta de microprocesadores y microcontroladores. Este dispositivo lo estaremos utilizando a lo largo del semestre y durante esta práctica fue importante familiarizarnos con el funcionamiento, el lenguaje de programación, y la configuración de algunos circuitos básicos. De manera concreta, los objetivos fueron los siguientes:

- Aprender a utilizar la terminal de GNU/Linux
- Aprender a utilizar Git
- Aprender a utilizar Arduino

## 2. MARCO TEÓRICO

Arduino es una plataforma de hardware y software de uso libre. Consta de distintas placas que permite leer ~~inputs~~ como sensores, botones, o inclusive pedazos de código y a través de un microcontrolador programar distintas funcionalidades mediante el Software de Arduino.[1] Por otro lado, Linux es un sistema operativo de ~~uso~~ libre. Consta de una terminal la cual permite realizar distintas modificaciones. En particular, podemos controlar directamente el almacenamiento y uso de archivos mediante distintos comandos. [2] Finalmente, GIT es un controlador de versiones cuya principal ventaja es el uso y administración de branches. El cual permite desarrollar de manera paralela distintos proyectos y posteriormente descartar o fusionar los cambios. Para poder hacer uso de ~~git~~ es necesario instalarlo y aprender comandos básicos para su correcto funcionamiento. GitHub es una plataforma digital para poder administrar y controlar distintos proyectos en la nube y de manera local mediante GIT.

## 3. DESARROLLO

### 3.1. Parte 1: Uso de Linux

- Mostrar el funcionamiento de los siguientes comandos: ~~pwd, mkdir, cd, ls, rmdir, rm, touch, cp, mv, less, more, cat, head, tail, ps, top, man~~. Los resultados fueron mostrados al profesor durante la clase.

### 3.2. Parte 2: Microcontrolador

#### 3.2.1. Blink

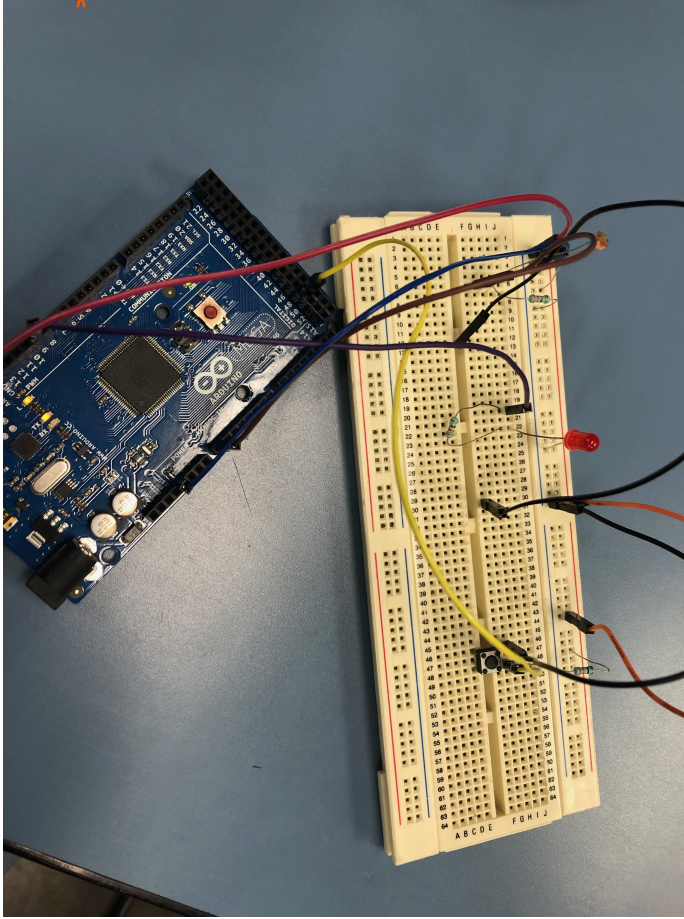
Utilizar el programa demo "Blink" incluido en el ambiente de desarrollo de Arduino para demostrar la funcionalidad de la tarjeta. A continuación incluimos el código básico del programa. Es básicamente un loop infinito en el cual el LED del Arduino se prende por 1 segundo y se apaga por el siguiente.

```
void setup() {  
    // initialize digital pin LED_BUILTIN as an output.  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH);  
    // turn the LED on (HIGH is the voltage level)  
    delay(1000);  
    // wait for a second  
    digitalWrite(LED_BUILTIN, LOW);  
    // turn the LED off by making the voltage LOW  
    delay(1000);  
    // wait for a second  
}
```

#### 3.2.2. Sensor de luminosidad

Creación de un circuito que responde a cambios en la intensidad luminosa del ambiente. Mediante una fotoresistencia, se sensa la luminosidad del ambiente, al ser menor a un valor preestablecido se enciende un led. Adicionalmente, el led cambia de estado (encendido/apagado) al presionar un botón. A continuación se incluyen los elementos principales del código utilizado. Se trata de un ciclo infinito en el cual se evalúan las lecturas tanto de la fotoresistencia como del botón y con base en ella se genera una señal de salida que será mostrada por el LED. Se incluye también una imagen del circuito.

Figura 1. El circuito utilizado para la subsección 2



```
void setup() {
    // initialize the digital pin as an output.
    pinMode(ledPin, OUTPUT);
    pinMode(pResistor, INPUT);
    pinMode(boton, INPUT);
    Serial.begin(9600);
}
// the loop routine runs over and over again forever.
void loop() {
    value2 = digitalRead(boton);
    Serial.println("hola");
    if(!value2){
        value = analogRead(pResistor);
        if(value > 25){
            digitalWrite(ledPin, LOW);
        }
    }
    else{
        digitalWrite(ledPin, HIGH);
    }
    //delay(500);
}
else{
    digitalWrite(ledPin, HIGH);
}
}
```

### 3.2.3. Display de 7 segmentos

Se ~~desarrollo~~ un programa que recibe como entrada desde el teclado un número hexadecimal (0-f), lo valida entrada y lo despliega utilizando un display de 7 segmentos. Se utilizó un display de 7 segmentos de catodo común y un set de resistencias de 330Ohms para construir el circuito físico. A continuación se incluye el código básico necesario para elaborar esta solución.

```
void setup() {
    // initialize the digital pin as an output.
    pinMode(2,OUTPUT);
    pinMode(3,OUTPUT);
    pinMode(4,OUTPUT);
    pinMode(5,OUTPUT);
    pinMode(6,OUTPUT);
    pinMode(7,OUTPUT);
    pinMode(8,OUTPUT);
    Serial.begin(9600);
}
// the loop routine runs over and over again forever:
void loop() {
    // loop to turn leds of seven seg ON

    if(Serial.available()>0){
        char entrada = Serial.read();
        Serial.println(entrada);
        switch (entrada) {
            case '0':
                digitalWrite(2, HIGH);
                digitalWrite(3, HIGH);
                digitalWrite(4, HIGH);
                digitalWrite(5, HIGH);
                digitalWrite(6, HIGH);
                digitalWrite(7, HIGH);
                digitalWrite(8, LOW);
                break;
        }
    }
}
```

Con módulos similares al usado para el caso "0", se generaron las salidas para todos los caracteres posibles. 16 en total.

## 4. RESULTADOS

Para las primeras dos secciones, los resultados fueron mostrados directamente al profesor. Sin embargo, cabe resaltar que al principio tuvimos que recurrir al profesor para el diseño de los circuitos ya que no entendíamos como era la relación de VCC, tierra, las resistencias y arduino. Otro punto importante fue el uso de los puertos de Arduino, ya que una sección está destinada a componentes analógicos y otro a digitales. Finalmente para el uso del display de 7 segmentos nos sorprendió no encontrar configuraciones en internet por lo cual tuvimos que programarlas a mano. Sin embargo, obtuvimos los resultados esperados.

## 5. CONCLUSIONES

- Leslie: En lo personal, la principal conclusión de esta práctica fue consolidar y recordar el diseño de circuitos básicos, familiarizarnos con el uso de

Linux y Git para control de versiones y entender el funcionamiento básico de Arduino, desde aprender como funciona su IDE hasta entender de manera general como funcionan los distintos bloques de registros y puertos. En esta práctica tuvimos algunas dudas y errores con la configuración de un botón y al final, lo estábamos colocando en un puerto de entradas análogas, lo cual no hacía sentido. Así, para próximas ocasiones, tendremos más cuidado con esos detalles.

- Diego: Con esta práctica entendimos el funcionamiento del arduino, de los principales comandos para manejar archivos en la terminal de linux, a configurar el repositorio local y subirlo a Github y finalmente a utilizar los puertos de Arduino para generar un circuito sencillo que permite configurar entradas y salidas.

## 6. ROL O PAPEL

- Leslie: Mi principal rol fue colaborar al armado del circuito, ir por los elementos físicos necesarios y colaborar en la programación del display de 7 segmentos.
- Diego: Yo me encargué de realizar la primera parte de la práctica en la terminal de linux y fui quien diseño los circuitos.

## 7. FUENTES CONSULTADAS

### REFERENCIAS

- [1] "What is arduino?" [Online]. Available: <https://www.arduino.cc/en/Guide/Introduction>
- [2] "What is linux?" [Online]. Available: <https://www.linux.com/what-is-linux>