

**UNCUYO**  
UNIVERSIDAD  
NACIONAL DE CUYO

# Trabajo Práctico 1 Informe

Sistemas Embebidos

LCC 2025  
Lucas Moyano 13446

1) Controle el brillo de los led 9, 10 y 11 del Arduino UNO utilizando el conversor Digital a Analógico del mismo. El brillo de estos leds debe controlarse de forma independiente para cada led.

```
void changeLedIntensity(int number, int intensity) {  
    pinMode(number, OUTPUT);  
    analogWrite(number, intensity);  
  
    switch (number) {  
        case 9:  
            intensityLed9 = intensity;  
            break;  
        case 10:  
            intensityLed10 = intensity;  
            break;  
        case 11:  
            intensityLed11 = intensity;  
            break;  
    }  
}
```

2) Encienda y apague el led 13 del Arduino UNO. Deberá utilizar dicho pin como salida digital.

```
pinMode(LED_BUILTIN, OUTPUT);  
if (input == "switch led 13:True"){  
    digitalWrite(13, HIGH);  
} else if (input == "switch led 13:False") {  
    digitalWrite(13, LOW);  
}
```

3) Muestre la intensidad luminosa captada por el LDR (conectado al pin A3 del Arduino UNO). Deberá utilizar el conversor Analógico a Digital.

```
void printLdrAnalogOutput() {  
    char ldrLuxBuffer[5];
```

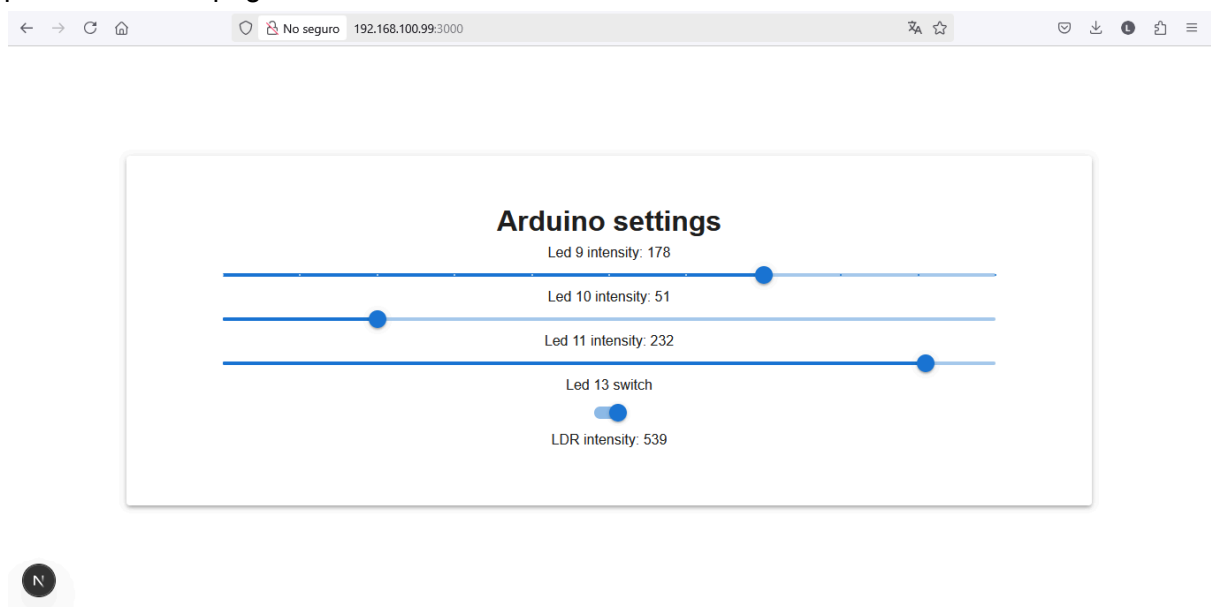
```

sprintf(ldrLuxBuffer, "%04d", analogRead(A3));
Serial.print("LDR lux: " + String(ldrLuxBuffer));
}

```

4) El control del brillo de los led 9, 10 y 11, el pin 13 y el valor de la intensidad luminosa sensado por el LDR deberá realizarse y mostrarse a través de una página web que deberá poder accederse desde otras computadoras conectadas en la misma red LAN o Internet.

Usé NextJS con MUI para el frontend y Flask con socketio para el backend, en la [Figura 1] podemos ver la página web desarrollada.



[Figura 1] - Pagina web con configuración arduino funcionando

```

async function handleChange9(event: Event, newValue: number |
number[]) {
  setValueLed9(newValue as number);
  const response = await fetch(backendURL + '/ledLuminosity/9', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({luminosity: newValue}),
  });
  const data = await response.json();
}

```

```
    console.log(data);  
  };
```

Código del frontend donde se envía al backend la luminosidad que se desea del led 9.

```
async function switchLed(event: ChangeEvent, checked: boolean) {  
  setLed13On(checked);  
  // Sends a POST request to the backend with the new value of the  
  switch  
  const response = await fetch(backendURL + '/switchLed/13', {  
    method: 'POST',  
    headers: {  
      'Content-Type': 'application/json',  
    },  
    body: JSON.stringify({on: checked}),  
  });  
  const data = await response.json();  
  console.log(data);  
};
```

Código del frontend donde se envía al backend un booleano indicando si quiero el led 13 prendido o apagado.

```
useEffect(() => {  
  
  socket.on('update_arduino_data', (data2) => {  
    setLed9Luminosity(data2.led9_intensity);  
    setLed10Luminosity(data2.led10_intensity);  
    setLed11Luminosity(data2.led11_intensity);  
    setLDRluminosity(data2.ldr_luminosity);  
    console.log("got arduino data: ", data2);  
  });  
  
  }, [socket, LDRluminosity, led9Luminosity, led10Luminosity,  
  led11Luminosity]);
```

Código del frontend donde se recibe del backend la luminosidad del LDR y los leds 9, 10 y 11.

```
# Changes led number luminosity  
@app.post("/ledLuminosity/<number>")  
def post_led_luminosity(number):  
    data = request.get_json()  
    luminosity = data.get('luminosity', 0)
```



```

        led11Intensity = int(line[78:81])
        socketio.emit("update_arduino_data",
{"ldr_luminosity": ldrLuminosity,

"led9_intensity": led9Intensity,

"led10_intensity": led10Intensity,

"led11_intensity": led11Intensity}))
        #else:
        #    print("not a case: ", line)
    except Exception as e:
        print(f"Error reading serial data: {e}")
    time.sleep(1.0)

```

Código del backend donde se lee desde el arduino la intensidad del led 9, 10, 11 y además la luminosidad del LDR.

5) La página web debe mostrar el valor actual del brillo de los leds, el estado del pin 13 y el valor leído por el LDR.

Leer punto 4.