

# UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



## CONTROL DE ENTRADAS Y SALIDAS ANALÓGICAS

**ASIGNATURA:** ROBÓTICA INDUSTRIAL

**DOCENTE:** ELIAS ALI ALVAREZ

### **INTEGRANTES:**

MIGUEL HUAYHUA CONDORI

CRISTHIAN VILLCA MAMANI

MARCOS BERRIOS PANCATA

CECILIO POMA MUÑOZ

**GESTIÓN 2023**

## 1. Anexe a su informe los resultados obtenidos del punto VII:

### a. SKETCHES

#### Ejercicio 2.1

```
int pinPot = 0;
int valorPot = 0;
void setup(){
  Serial.begin(9600);
}
void loop(){
  valorPot = analogRead(pinPot);
  Serial.println(valorPot);
  delay(500);
}
```

#### Ejercicio 2.3

```
int pinLet = 3;
void setup(){
  pinMode(pinLet, OUTPUT);}
void loop(){
  for(int i=0;i<255;i++){
    analogWrite(pinLet,i);
    delay(10);
  }
  for(int i=255;i>0;i--){
    analogWrite(pinLet,i);
    delay(10);}
}
```

#### Ejercicio 2.5

```
int letRojo = 9;
int letVerde = 10;
int letAzul = 11;
void setup(){
  pinMode(letRojo, OUTPUT);
  pinMode(letVerde, OUTPUT);
  pinMode(letAzul, OUTPUT);
}
void loop(){
  analogWrite(letRojo,255);
  analogWrite(letVerde,0);
  analogWrite(letAzul,0);
  delay(500);
  analogWrite(letRojo,0);
```

#### Ejercicio 2.2:

```
int pinLet = 13;
int pinPot = 1;
int valorPot = 0;
void setup(){
  pinMode(pinLet, OUTPUT);
  Serial.begin(9600);
}
void loop(){
  valorPot = analogRead(pinPot);
  if ( valorPot > 300) {
    digitalWrite(pinLet, HIGH);
  } else {
    digitalWrite(pinLet, LOW);
  }
  Serial.println(valorPot);
  delay(10);
}
```

#### Ejercicio 2.4

```
int pinLed = 9;
int pinPot = 0;
int valorPot = 0;
int brilloLet = 0;
void setup () {
  pinMode (pinLed, OUTPUT);}
void loop () {
  valorPot = analogRead(pinPot);
  brilloLet = valorPot / 4;
  analogWrite(pinLed, brilloLet);
}
```

#### Ejercicio 2.6

```
int pinFot = 0;
int valorFot = 0;
void setup(){
  Serial.begin(9600);
}
void loop(){
  valorFot = analogRead(pinFot);
  Serial.println(valorFot);
  delay(200);
}
```

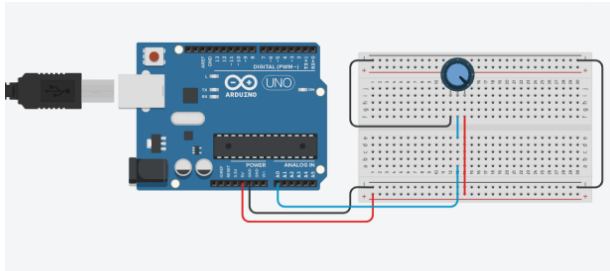
```

analogWrite(ledVerde,255);
analogWrite(ledAzul,0);
delay(500);
analogWrite(ledRojo,0);
analogWrite(ledVerde,0);
analogWrite(ledAzul,255);
delay(500);
}

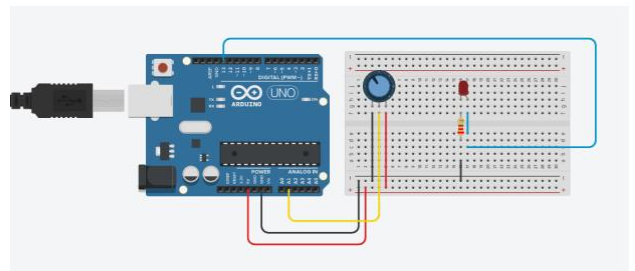
```

**b. Simulación en Proteus 8.1 (se simuló en TinkerCad)**

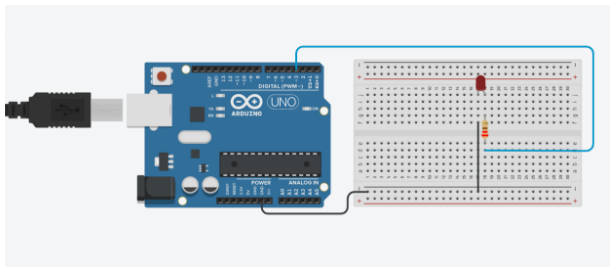
**Simulación 2.1**



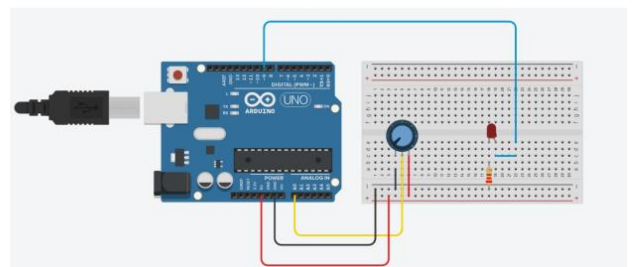
**Simulación 2.2**



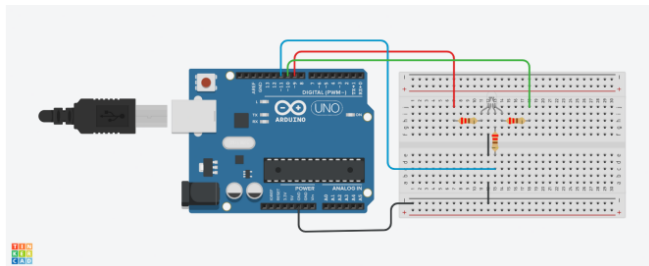
**Simulación 2.3**



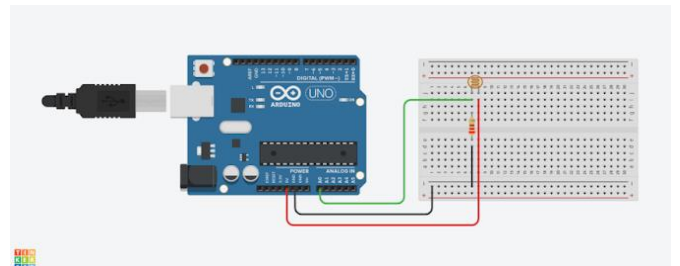
**Simulación 2.4**



**Simulación 2.5**



**Simulación 2.6**

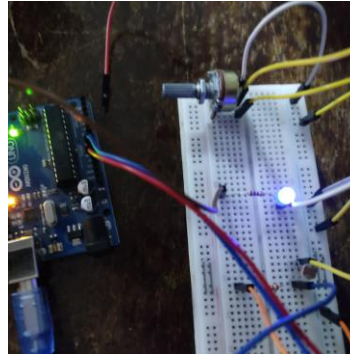


### c. Fotografías de los circuitos

2.1



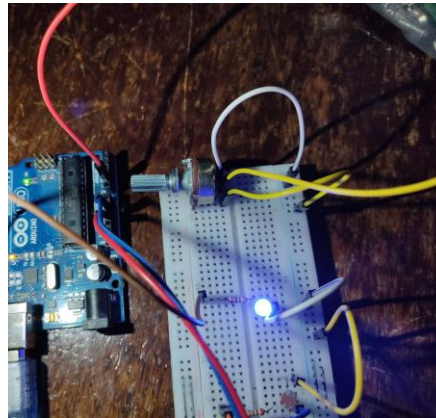
2.2



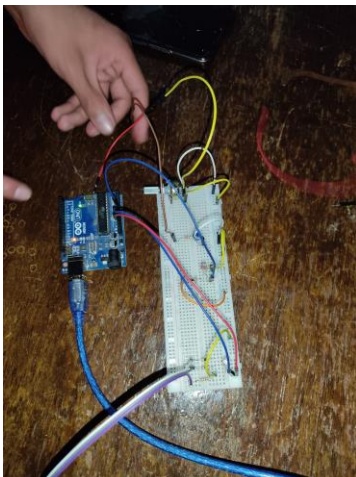
2.3



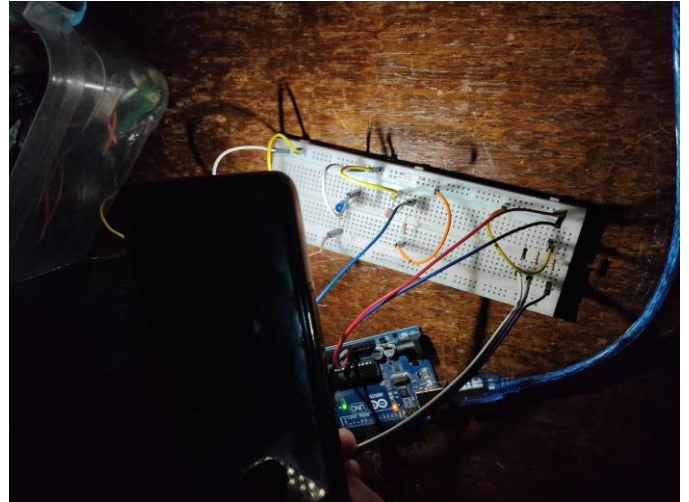
2.4



Ejercicio 2.5



Ejercicio 2.6



2. Realice e implemente un Sketch, el cual deberá generar los colores rojo, amarillo, verde, rosado, naranja, celeste, violeta, azul, verde claro en un led RGB

Código Rojo	Código Amarillo	Código Verde	Código Rosado	Código Naranja	Código Celeste	Código Violeta	Código Azul	Código Verde Claro
(255,0,0)	(255,255,0)	(0, 255,255)	(188,143,143)	(252,147,3)	(0, 170,228)	(140, 0, 211)	(0, 0, 255)	(144,238,144)

**CÓDIGO**

```

int letRojo = 9;
int letVerde = 10;
int letAzul = 11;
void setup(){
  pinMode(letRojo, OUTPUT);
  pinMode(letVerde, OUTPUT);
  pinMode(letAzul, OUTPUT);
}
void loop(){
  //rojo
  analogWrite(letRojo,255);
  analogWrite(letVerde,0);
  analogWrite(letAzul,0);
  delay(500);
  //amarillo
  analogWrite(letRojo,255);
  analogWrite(letVerde,255);
  analogWrite(letAzul,0);
  delay(500);
  //verde
  analogWrite(letRojo,0);
  analogWrite(letVerde,255);
  analogWrite(letAzul,0);
  delay(500);

```

```

//rosado
analogWrite(letRojo,188);
analogWrite(letVerde,143);
analogWrite(letAzul,143);
//naranja
analogWrite(letRojo,252);
analogWrite(letVerde,147);
analogWrite(letAzul,3);

//celeste
analogWrite(letRojo,0);
analogWrite(letVerde,170);
analogWrite(letAzul,228);
delay(500);
//violeta
analogWrite(letRojo,140);
analogWrite(letVerde,0);
analogWrite(letAzul,211);
delay(500);
//azul
analogWrite(letRojo,0);
analogWrite(letVerde,0);
analogWrite(letAzul,255);
delay(500);
//verde claro
analogWrite(letRojo,144);
analogWrite(letVerde,238);
analogWrite(letAzul,144);
}

```

### 3. Investigue las características principales de la modulación PWM:

La modulación de ancho de pulso o PWM (del inglés Pulse Width Modulation) es una técnica utilizada para controlar la cantidad de energía entregada a un dispositivo eléctrico mediante la modulación de la anchura de los pulsos de una señal eléctrica.

En la modulación PWM, se genera una señal periódica con una frecuencia constante y un ciclo de trabajo variable. El ciclo de trabajo se refiere al porcentaje de tiempo en que la señal está en un estado "alto" en relación con el tiempo total del ciclo. Por ejemplo, si el ciclo de

trabajo es del 50%, la señal está en estado alto durante el 50% del tiempo y en estado bajo durante el 50% restante.

El control del ciclo de trabajo en una señal PWM permite variar la cantidad de energía que se entrega a un dispositivo eléctrico, como un motor o un LED. Al variar el ciclo de trabajo se puede variar la potencia entregada al dispositivo y por lo tanto su velocidad, brillo, etc.

La modulación PWM se utiliza comúnmente en aplicaciones como el control de motores, la regulación de la intensidad de la luz en LEDs, la carga de baterías, entre otros.

#### **4. Investigue las características de un CAD (conversor análogo digital) y el de un CDA (conversor digital análogo)**

Un conversor analógico-digital (ADC, por sus siglas en inglés) es un dispositivo que convierte una señal analógica en una señal digital. A continuación, se presentan algunas características comunes de un conversor analógico-digital:

**Resolución:** La resolución de un ADC se refiere al número de bits que se utilizan para representar la señal digitalizada. Cuanto mayor sea la resolución, mayor será la precisión del conversor. La resolución se mide en bits y suele estar en el rango de 8 a 24 bits.

**Frecuencia de muestreo:** La frecuencia de muestreo se refiere a la cantidad de veces que se mide la señal analógica en un segundo. La frecuencia de muestreo debe ser lo suficientemente alta como para capturar la información relevante de la señal analógica. La frecuencia de muestreo se mide en muestras por segundo (SPS).

**Error de cuantificación:** El error de cuantificación es la diferencia entre el valor analógico real de la señal y el valor digitalizado por el conversor. Este error se debe al hecho de que la señal analógica es continua y el conversor debe dividirla en pasos discretos. Cuanto mayor sea la resolución, menor será el error de cuantificación.

**Linealidad:** La linealidad se refiere a la capacidad del conversor para producir una señal digital que sea proporcional a la señal analógica de entrada. Un ADC ideal sería perfectamente lineal, pero en la práctica siempre hay algún grado de no linealidad.

**Rango dinámico:** El rango dinámico es la relación entre el nivel máximo y mínimo de la señal de entrada que el conversor puede digitalizar con precisión. Cuanto mayor sea el rango dinámico, mayor será la capacidad del conversor para manejar señales de entrada de alta amplitud.

**Tiempo de conversión:** El tiempo de conversión se refiere al tiempo que tarda el conversor en convertir una muestra de la señal analógica en una señal digital. El tiempo de conversión puede variar de unos pocos microsegundos a varios milisegundos, dependiendo del tipo de conversor y su velocidad de operación.

## **5. Conclusiones y observaciones:**

El laboratorio nos dio buena parte del entendimiento primario del cómo funciona la electrónica analógica, siendo de que está presente en casi todos los casos de uso del mundo moderno con la implementación de las tecnologías presentes, así mismo, el manejo de estas con microcontroladores básicos pero funcionales, el poder manipular casi todo aquello que nos genere datos con el simple hecho de mover cosas físicas, las observaciones van más en el enfoque de simulación, haciendo que este no sea tan parecido con el mundo real cuando se trata de implementar la energía, pudiendo variar con el simple hecho de no otorgar la energía exacta a ciertos componentes, obteniendo resultados con alguna pequeña diferencia a la simulación realizada en cada uno de los ejercicios.