



Escuela de Física

Ingeniería Física

## **Instructivo de Laboratorio**

Instrumentación I

**26 de agosto de 2025**

Versión: 0.1

---

Realizado por: Juan J. Rojas

# Índice general

<b>1. Medición de estímulos eléctricos</b>	<b>2</b>
1.1. Objetivos . . . . .	2
1.2. Materiales y equipo . . . . .	2
1.3. Procedimiento . . . . .	2
1.4. Preguntas para el análisis de resultados . . . . .	5

# Laboratorio 1

## Medición de estímulos eléctricos

### 1.1. Objetivos

Al finalizar este laboratorio el estudiante estará en capacidad de:

- Configurar un sensor digital para lograr una medición eléctrica con la mayor resolución posible.
- Calcular el error asociado a la medición utilizando una multímetro digital de alta precisión.

### 1.2. Materiales y equipo

A suministrar por la Escuela:

- 1 ARDUINO UNO R4 MINIMA
- 1 sensor digital de voltaje, corriente y potencia (INA219)
- 1 mini breadboard
- Cables de interconexión (jumpers) macho-macho

### 1.3. Procedimiento

- a. Realice las conexiones del circuito tal y como se indica en la Figura ??.
- b. Abra el programa Arduino IDE

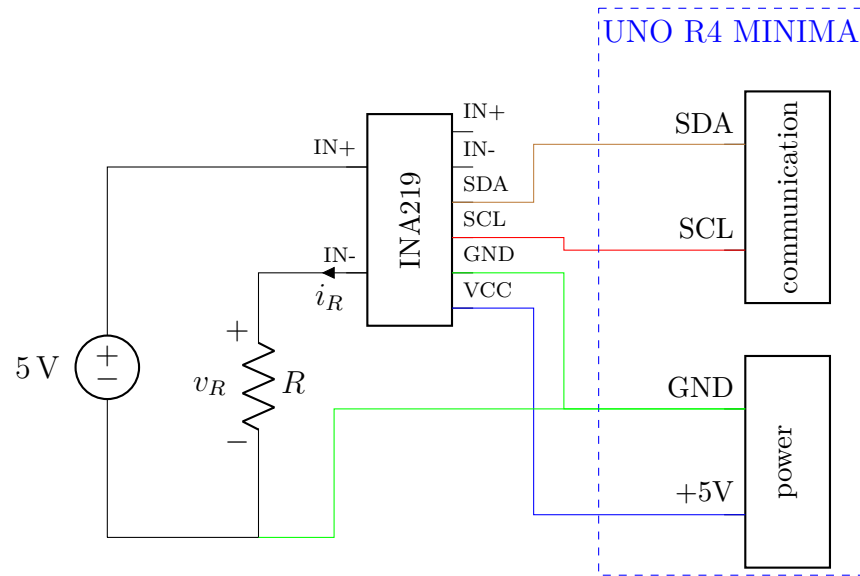


Figura 1.1: Conexión de sensor de corriente y voltaje INA219

c. Incluya el siguiente código en el Arduino IDE

```
#include <Wire.h>
#include <INA219.h>
INA219 ina219 (0x40);

//Instrucciones: Una vez subido, correr el monitor serial y presionar Enter para
// iniciar la medición, en caso de que no inicie automáticamente.
// Usar libreria INA219 (programada por Rob Tillart), de la lista de librerías.

void setup() {
  Serial.begin(115200);
  Wire.begin();
  delay(2000);
  if (!ina219.begin()) {
    Serial.println("INA219 not found");
    while (1){ delay(10); }}
  Serial.println("INA219 initialized");
  //Here we can set max current shunt value to calibrate the Calibration
  //Register (8.5.1)
  ina219.setMaxCurrentShunt(1, 0.1);    // adjust if needed
  Serial.print("Shunt value:\t");
  Serial.println(ina219.getShunt(), 4);
  Serial.print("Max current:\t");
  Serial.println(ina219.getMaxCurrent(), 4);
  //Set gain based on desired range
  ina219.setGain(1);
  Serial.print("Gain:\t");
  Serial.println(ina219.getGain());
  //Set bus voltage range
  Serial.print("Bus voltage range:\t");
  ina219.setBusVoltageRange(16);
  Serial.println(ina219.getBusVoltageRange());

  Serial.print("Calibrated?:\t");
```

```
Serial.println(ina219.isCalibrated());
Serial.print("CLSB:\t");
Serial.println(ina219.getCurrentLSB());
Serial.println("press any key");
while (!Serial.available()) {
}
}

float curre = 0;
float volta = 0;
float power = 0;
float shunt = 0;
int counter = 0;
// Here we set the oversampling
int samples = 100;
int delayTime = 1000/samples;

void loop() {
    counter++;
    curre += ina219.getCurrent_mA();
    volta += ina219.getBusVoltage();
    power += ina219.getPower_mW();
    shunt += ina219.getShuntVoltage_mV();
    if (counter >= samples) {
        Serial.print(curre/samples);
        Serial.print(",");
        Serial.print(volta/samples);
        Serial.print(",");
        Serial.print(power/samples);
        Serial.print(",");
        Serial.println(shunt/samples);
        counter = 0;
        curre = 0;
        volta = 0;
        power = 0;
        shunt = 0;
    }
    delay(delayTime);
}
```

- d. Tomando en cuenta el valor de la resistencia que se le asignó, calcule el valor de la corriente esperada en el circuito utilizando la Ley de Ohm.
- e. Investigue en el datasheet que significa el parametro PGA y como se relaciona con el rango del sensor. Ver 8.5.1
- f. Configure el sensor para que el rango de medición sea el más adecuado para la resistencia que se le asignó, suba el código modificado al Arduino.
- g. Realice dos mediciones de 10s de duración con una frecuencia de 1Hz al mismo tiempo que mide la corriente con un multímetro digital de alta precisión.
- h. Calibre el sensor utilizando un multímetro digital de alta precisión y cambie nuevamente el código para que el valor medido por el sensor sea lo más cercano posible al valor medido por el multímetro. Ver 8.5.2.1. Suba el código modificado al Arduino.

- i. Realice dos mediciones de 10s de duración con una frecuencia de 1Hz al mismo tiempo que mide la corriente con un multímetro digital de alta precisión.
- j. Modifique el código para que tener oversampling de 100 muestras por segundo y suba el código modificado al Arduino.
- k. Realice dos mediciones de 10s de duración con una frecuencia de 100Hz al mismo tiempo que mide la corriente con un multímetro digital de alta precisión.
- l. Para cada ser de mediciones calcule lo siguiente e incluya en una tabla:
  - El valor promedio,  $\overline{i_R}$
  - La desviación estándar,  $\sigma$
  - El valor de la incertidumbre estándar,  $\sigma_x = \sigma/\sqrt{n}$
  - La exactitud
  - La precisión
  - La repetibilidad
- m. Tome en cuenta lo siguiente:
  - La exactitud se calculará como el mayor error relativo obtenido en cualquier punto de los datos en todas las corridas de medición.
  - La precisión se calculará como la mayor desviación estándar entre todas las corridas de medición.
  - La repetibilidad se calculará de la siguiente manera:

$$\text{repetibilidad} = \sqrt{\frac{\sum_{i=1}^n (\text{error absoluto})^2}{n}}$$

## 1.4. Preguntas para el analisis de resultados

- ¿Cómo afecta el valor de la resistencia al rango de medición del sensor?
- ¿Como afecta el valor de PGA al rango de medición del sensor?
- ¿En que consiste el oversampling y cómo afecta la resolución de la medición?
- ¿Que es lo que más afecta la precisión de la medición?
- ¿Qué es lo que más afecta la exactitud de la medición?