



**Facultad de Ingeniería
Ingeniería Electrónica**

**Sistemas Embebidos
Tarea Académica 3**

Alumno

- **Carlos Christopher Carbajal Jordán(U201712774)**

Ciclo: 2022-2

Fecha: 29/10/2022

Pregunta 1(Carlos Carbajal Jordan):

Link de la explicacion:

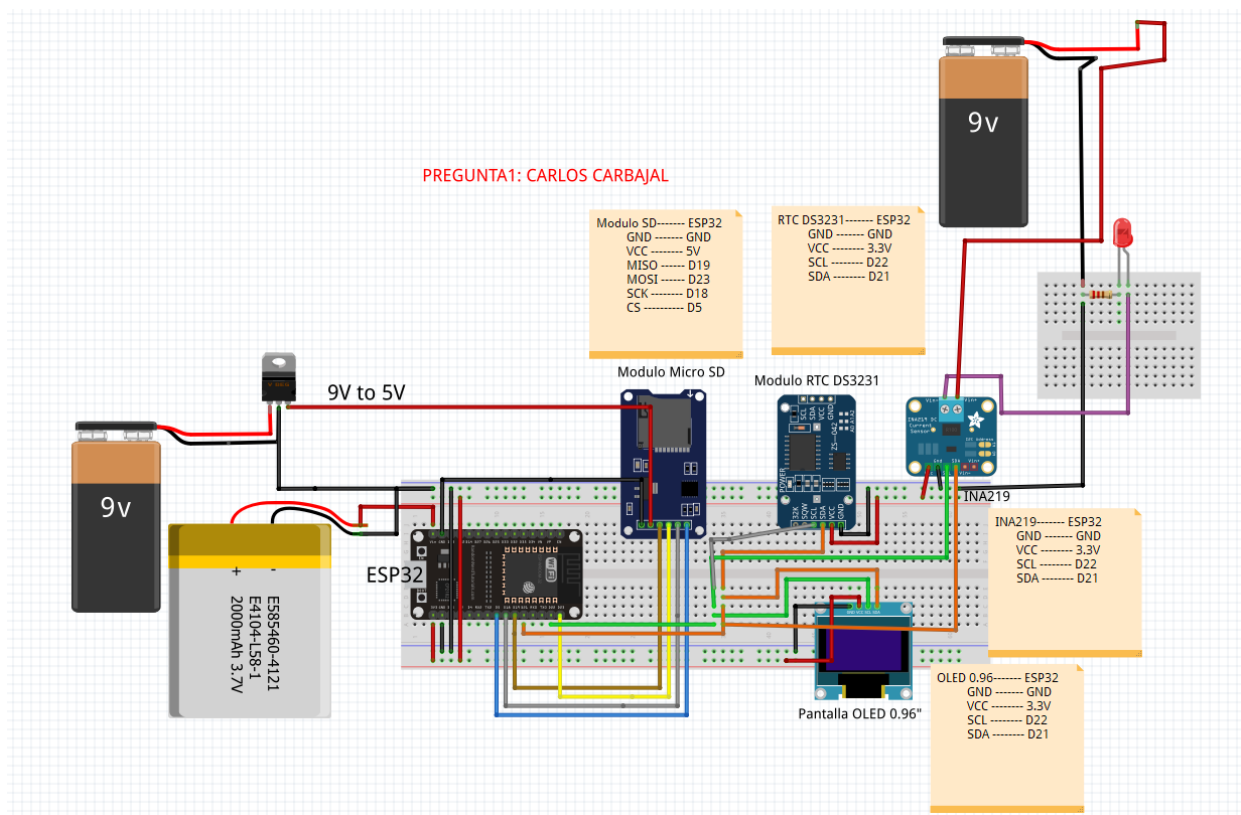
<https://youtu.be/Vt96OK3mOO8>

Objetivo General: Usar módulos de desarrollo con sistemas embebidos SoC de 32 bits ESP32 y/o NodeMCU con Conectividad Wi-Fi a una plataforma IoT para desarrollar proyectos electrónicos.

Objetivo específico: Evaluar el consumo de energía o la duración de la batería de sistemas de energía solar, baterías, motores, controladores o módulos electrónicos.

Indicaciones: Diseñar un voltímetro y amperímetro digital usando un SoC de 32 bits.

- Elabore su esquema electrónico o pictórico



Sustente la teoría de funcionamiento del diseño

Para el diseño del medidor de temperatura y humedad de suelo se utilizaron los siguientes materiales y equipos:

Módulo Soc ESP32

Pantalla OLED 0.96" : GND,VCC(3.3v),SCL,SDA

Módulo Micro SD: GND,VCC(5v),MISO(D19),MOSI(D23),SCK(D18),CS(D5)

Memoria Micro SD de 32 GB: dentro del módulo Micro SD

Modulo Reloj de tiempo real-RTC: GND, VCC(3.3V),SCL(D22),SDA(D21)

INA219: GND,VCC(3.3V),SCL(D22),SDA(D21)

Para el circuito de prueba:

Resistencia 220ohm

Bateria 9V

Explicación:

El circuito consiste en usar el esp32 como el cerebro principal para la lectura del módulo ina219 donde por medio de librería sacará los valores de corriente, voltaje shunt y load estos datos serán enviados a través del protocolo i2c y se imprimirá los valores en una pantalla oled (también por transmisión i2c) y además se le enviará dicha información a la plataforma blynk y al módulo SD con su respectiva memoria para almacenar una base de datos. Queda recalcar que se usa el RTC para calcular el periodo del tiempo que se enviara al SD y a su vez para enviar la fecha y hora del momento en la información del SD. Para la medición se hizo un circuito simple que consiste en una batería de 9V una resistencia y un led. En el código se tomó en cuenta que la impresión en la interfaz serial y en el oled se refrescará a diferente tiempo que en el envío de datos al SD y se tomó en cuenta la impresión en la interfaz serial para verificar que se está enviando la información correctamente

- Realice el código del programa usando el IDE de Arduino. (Ejercicio 1)

```
#define BLYNK_PRINT Serial
#define BLYNK_TEMPLATE_ID "TMPLNXjDP_bx"
#define BLYNK_DEVICE_NAME "multimetro"
```

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_INA219.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include "RTClib.h"
#include <SD.h>
#include <BlynkSimpleEsp32.h>
```

```
/*****
```

```

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET      -1 // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64,
                                0x3C for 128x32

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
Adafruit_INA219 ina219;
RTC_DS3231 rtc;

TimeSpan step = TimeSpan(5);
File file;
DateTime now, later;
String data;
BlynkTimer timer;

char auth[] = "_zh-V47ti_h3IBFWfak1f0ZKX3c8EIlo";
char ssid[] = "CARBAJALJORDAN";
char pass[] = "77wilbermarie177";

unsigned long previousMillis = 0;
unsigned long interval = 2000;
float shuntvoltage = 0.00;
float busvoltage = 0.00;
float current = 0.00;
float loadvoltage = 0.00;
//float energy = 0.00,  energyCost, energyPrevious, energyDifference;
float power = 0.00;

```

```
float capacity = 0.00;

static const uint8_t image_data[] = {
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x08, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x18, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x38, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x78, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x78, 0x00, 0x00, 0x00,
    0x00, 0x80, 0x78, 0x00, 0x10, 0x00,
    0x01, 0x00, 0xfc, 0x00, 0x08, 0x00,
    0x01, 0x00, 0xfc, 0x00, 0x08, 0x00,
    0x03, 0x00, 0xfe, 0x00, 0x0c, 0x00,
    0x06, 0x00, 0xff, 0x00, 0x04, 0x00,
    0x06, 0x00, 0xff, 0x80, 0x06, 0x00,
    0x06, 0x00, 0xff, 0xc0, 0x06, 0x00,
    0x0e, 0x00, 0xff, 0xc0, 0x07, 0x00,
    0x0e, 0x00, 0xff, 0xe0, 0x07, 0x00,
    0x0e, 0x00, 0x7f, 0xe0, 0x07, 0x00,
    0x0e, 0x00, 0x7f, 0xf0, 0x07, 0x00,
    0x1e, 0x00, 0x3f, 0xf0, 0x07, 0x00,
    0x1e, 0x00, 0x3f, 0xf0, 0x07, 0x80,
    0x1f, 0x00, 0x1f, 0xf0, 0x0f, 0x80,
    0x1f, 0x00, 0x0f, 0xf0, 0x0f, 0x80,
    0x1f, 0x00, 0x07, 0xf0, 0x0f, 0x80,
    0x1f, 0x80, 0x03, 0xf0, 0x1f, 0x00,
    0x0f, 0x80, 0x03, 0xf0, 0x1f, 0x00,
    0x0f, 0xc0, 0x01, 0xe0, 0x3f, 0x00,
    0x0f, 0xe0, 0x01, 0xe0, 0x7f, 0x00,
    0x0f, 0xf0, 0x01, 0xc0, 0xff, 0x00,
```

```

    0x07, 0xf8, 0x01, 0xc1, 0xfe, 0x00,
    0x07, 0xfc, 0x01, 0x83, 0xfe, 0x00,
    0x07, 0xff, 0x01, 0x0f, 0xfc, 0x00,
    0x03, 0xff, 0xe2, 0x7f, 0xfc, 0x00,
    0x01, 0xff, 0xff, 0xff, 0xf8, 0x00,
    0x01, 0xff, 0xff, 0xff, 0xf8, 0x00,
    0x00, 0xff, 0xff, 0xff, 0xf0, 0x00,
    0x00, 0x7f, 0xff, 0xff, 0xe0, 0x00,
    0x00, 0x3f, 0xff, 0xff, 0xc0, 0x00,
    0x00, 0x1f, 0xff, 0xff, 0x80, 0x00,
    0x00, 0x07, 0xff, 0xfe, 0x00, 0x00,
    0x00, 0x03, 0xff, 0xfc, 0x00, 0x00,
    0x00, 0x00, 0x7f, 0xe0, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00,
};

void send_sensor() {
    now = rtc.now();
    if(later == now) {
        later = now + step;
        ina219values();
        Serial.println("Hora de grabar.....");
        //loadvoltage,current,power,capacity
        if(isnan(loadvoltage) || isnan(current)||isnan(power) || isnan(capacity))
    {
        Serial.println("Error de lectura");
        return;
    }
    file = SD.open("/Ejercicio1.csv", FILE_APPEND);
    if (file) {
        Serial.println("imprimiendo....");

```

```

        data = String(now.day()) + "/" + String(now.month()) + "/" +
String(now.year()) + "," + String(now.hour()) + ":" +
        String(now.minute()) + ":" + String(now.second()) + " , " +
String(loadvoltage) + "V  , " + String(current)+ "mA  , " + String(power) +
"mW  , " + String(capacity)+"mAh";

        file.println(data);

        file.close();
    } else {
        Serial.println("Error al abrir el archivo");
    }
    Blynk.virtualWrite(V0, loadvoltage);

    Blynk.virtualWrite(V1, current);
    Blynk.virtualWrite(V2, power/1000);
    Blynk.virtualWrite(V4, capacity);
    Blynk.virtualWrite(V7, current);

    // displaydata();
}
}

void setup()
{
    Serial.begin(115200);
    while (!Serial) {
        // will pause Zero, Leonardo, etc until serial console opens
        delay(1);
    }

    uint32_t currentFrequency;

```

```
Serial.begin(9600);

// initialize OLED display
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(2);
display.setCursor(0, 0);
display.print("Multimetro");
display.drawBitmap(40,20, image_data,44, 45, 1);
display.display();


// Inicia INA219.


if (! ina219.begin()) {
    Serial.println("Error encontrando INA219 chip");
    while (1) {
        delay(10);
    }
}

if (!rtc.begin()) {
    Serial.println("Error en el reloj");
    while (1);
}

if (!SD.begin()) {
    Serial.println("Error en la sd");
    while(1);
}

if (rtc.lostPower()) {
```



```

        rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
    }

    if (!SD.exists("/Ejercicio1.csv")) {
        file = SD.open("/Ejercicio1.csv", FILE_WRITE);
        if (file) {
            file.println("Fecha,Hora,Humedad,Temperatura");
            file.close();
        } else {
            Serial.println("Error al crear el archivo.");
        }
    }

    now = rtc.now();
    later = rtc.now() + TimeSpan(60);

    Serial.println("Multimetro con INA219 ...");
    Blynk.begin(auth, ssid, pass);
    timer.setInterval(10, send_sensor);
}

```

```

void loop()
{

    Blynk.run();
    timer.run();
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval)
    {

```

```

    previousMillis = currentMillis;

    ina219values();

    displaydata();
}

send_sensor();
}

void ina219values() {

    shuntvoltage = ina219.getShuntVoltage_mV();
    busvoltage = ina219.getBusVoltage_V();
    current = ina219.getCurrent_mA();
    loadvoltage = busvoltage + (shuntvoltage / 1000);
    power = loadvoltage * current;
    capacity = capacity + current / 1000;

    if (loadvoltage < 1 )loadvoltage = 0;
    if (current < 1 )
    {
        current = 0;
        power = 0;
        capacity = 0;
    }

    Serial.print("Bus Voltage:  "); Serial.print(busvoltage); Serial.println("
V");

    Serial.print("Shunt Voltage: "); Serial.print(shuntvoltage);
    Serial.println(" mV");

    Serial.print("Load Voltage:  "); Serial.print(loadvoltage); Serial.println("
V");

    Serial.print("Current:      "); Serial.print(current); Serial.println("
mA");
}

```

```
Serial.print("Power:      "); Serial.print(power); Serial.println(" mW");  
Serial.print("Capacity:   "); Serial.print(capacity); Serial.println("Ah");
```

```
Serial.println("-----");  
}  
void displaydata() {  
    display.clearDisplay();  
    display.setTextColor(WHITE);  
    display.setTextSize(1);  
    //variables de salida loadvoltage,current,power,capacity  
    // VOLTAJE  
    display.setCursor(0, 5);  
    display.print(loadvoltage, 2);  
    display.print(" V");  
    // CORRIENTE  
    if (current > 1000) {  
        display.setCursor(60, 5);  
        display.print((current / 1000), 2);  
        display.println(" A");  
        display.setCursor(0, 15);  
        display.println("-----");  
    }  
    else  
    {  
        display.setCursor(60, 5);  
        display.print(current, 1);  
        display.println(" mA");  
        display.setCursor(0, 15);  
        display.println("-----");  
    }  
}
```

```
//Impresion para separacion
display.setCursor(60, 20);
display.print("|");
display.setCursor(60, 24);
display.print("|");
display.setCursor(60, 28);
display.print("|");
display.setCursor(60, 32);
display.print("|");
display.setCursor(60, 36);
display.print("|");
display.setCursor(60, 40);
display.print("|");
display.setCursor(0, 46);
display.print("-----");

// Potencia
if (power > 1000) {
    display.setCursor(0, 24);
    display.print("Consumo");
    display.setCursor(0, 36);
    display.print(String((power / 1000), 2));
    display.println(" W");
}
else {
    display.setCursor(0, 24);
    display.print("Consumo: ");
    display.setCursor(0, 36);
    display.print(power, 2);
}
```

```

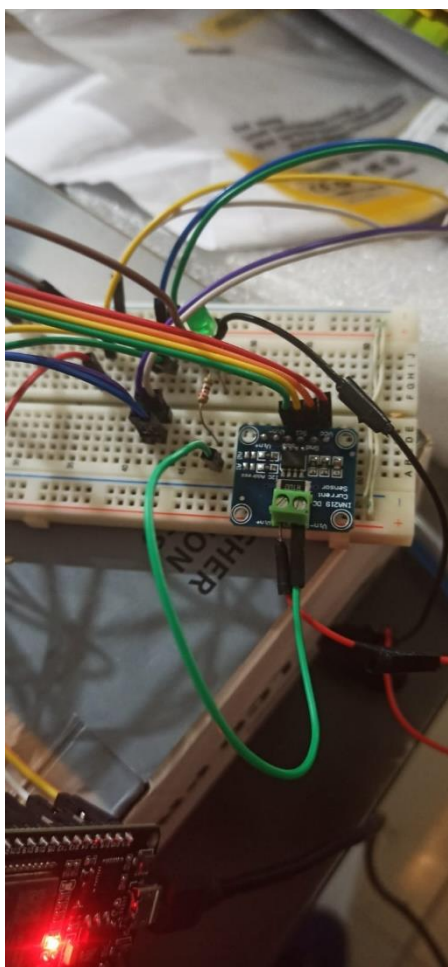
    display.println(" mW");
}

// Capacidad
if (capacity > 1000) {
    display.setCursor(65, 24);
    display.println("Capacidad: ");
    display.setCursor(65, 36);
    display.print((capacity / 1000), 2);
    display.println(" Ah");
}
else
{
    display.setCursor(65, 24);
    display.println("Capacidad: ");
    display.setCursor(65, 36);
    display.print(capacity, 2);
    display.println(" Ah");
}
display.display();
}

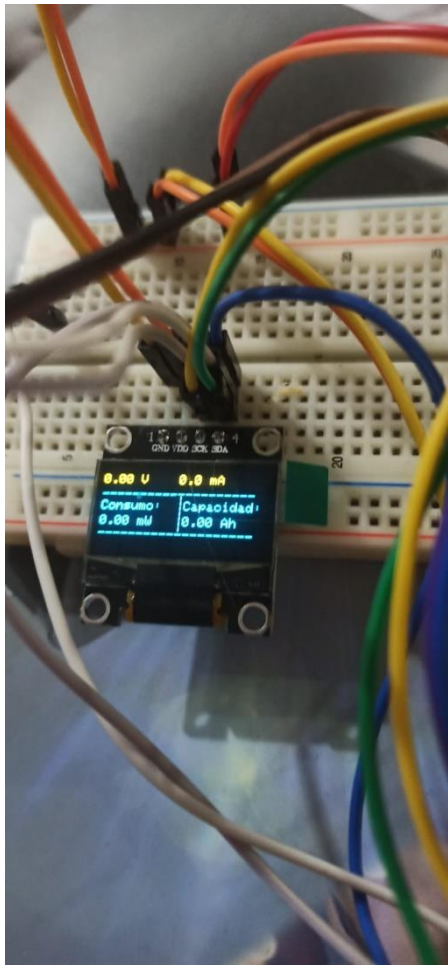
```

-Implemente el prototipo.

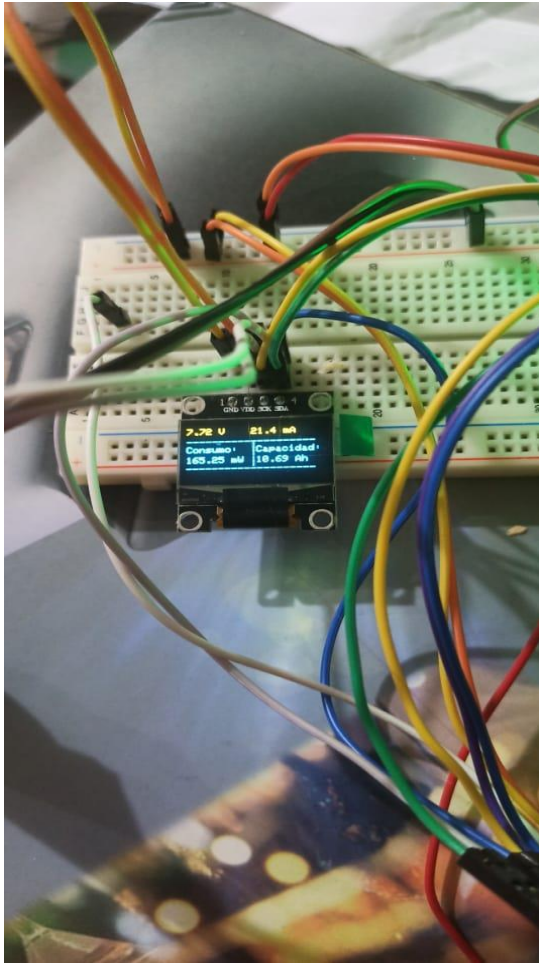
Zona del circuito a analizar (circuito abierto)



Visualización con el circuito abierto (el que se muestrea)



Visualización con el circuito cerrado (batería colocada)



- Grabar los datos obtenidos de temperatura y humedad del suelo en una memoria SD (16GB o 32GB) en formato CSV cada minuto generado por el Reloj de Tiempo Real (RTC).

Se uso esta parte del código para imprimir los datos al SD

```

1  return;
2  }
3  file = SD.open("/Ejercicio1.csv", FILE_APPEND);
4  if (!file) {
5      Serial.println("Imprimiendo...");
6      data = String(now.day()) + "/" + String(now.month()) + "/" + String(now.year()) + " " + String(now.hour()) + ":" +
7      String(now.minute()) + ":" + String(now.second()) + " , " + String(loadvoltage) + "V , " + String(current) + "mA , " + String(power) + "mW , " + String(capacity) + "mAh";
8      file.println(data);
9      file.close();
10 } else {
11     Serial.println("Error al abrir el archivo");
12 }
13 Blynk.virtualWrite(V0, loadvoltage);
14
15 Blynk.virtualWrite(V1, current);
16 Blynk.virtualWrite(V2, power/1000);
17 Blynk.virtualWrite(V4, capacity);
18 Blynk.virtualWrite(V5, current);

```

Dentro del archivo csv en el que se ingresó los datos de la fecha, hora, voltaje, corriente, potencia, capacidad (más de 2000 muestras)

Fecha	Hora	Voltaje	Corriente	Potencia	Capacidad
29/10/2022	00:11:10	6.99	18.4	128.54	0.0702
29/10/2022	00:11:15	6.99	18.2	127.21	0.071
29/10/2022	00:11:20	6.99	18.3	127.84	0.0715
29/10/2022	00:11:25	6.99	18.3	127.84	0.0722
29/10/2022	00:12:40	6.97	18.1	126.23	0.55
29/10/2022	00:12:45	6.97	18.2	126.92	0.6
29/10/2022	00:12:50	6.97	18.1	126.23	0.67
29/10/2022	00:12:55	6.97	18.4	128.32	0.73
29/10/2022	00:13:00	6.97	18.3	127.55	0.8
29/10/2022	00:13:05	6.97	18	125.46	0.85
29/10/2022	00:13:10	6.97	18	125.46	0.93
29/10/2022	00:13:15	6.97	18.1	126.15	0.98
29/10/2022	00:13:20	6.96	18	125.31	1.05
29/10/2022	00:13:25	6.97	17.9	124.69	1.11
29/10/2022	00:13:30	6.97	17.9	124.69	1.18
29/10/2022	00:13:35	6.96	18.2	126.7	1.23
29/10/2022	00:13:40	6.96	18.2	126.7	1.31
29/10/2022	00:13:45	6.96	18.2	126.7	1.36
29/10/2022	00:13:50	6.96	17.9	124.62	1.43
29/10/2022	00:13:55	6.96	18.1	126.01	1.49
29/10/2022	00:14:00	6.96	18	125.31	1.56
29/10/2022	00:14:05	6.96	18.1	126.01	1.61
29/10/2022	00:14:10	6.96	18.1	125.94	1.69
29/10/2022	00:14:15	6.96	18.1	125.94	1.74
29/10/2022	00:14:20	6.96	18	125.31	1.81
29/10/2022	00:14:25	6.96	18	125.24	1.87
29/10/2022	00:14:30	6.96	18.1	125.94	1.94
29/10/2022	00:14:35	6.96	18.2	126.63	1.99

- Presentar evidencias de las pruebas de laboratorio.

Iniciando

```

11:59:58.206 -> [2816] Connecting to blynk.cloud:80
11:59:58.756 -> [3526] Ready (ping: 203ms).
11:59:58.853 -> Bus Voltage: 8.00 V
11:59:58.918 -> Shunt Voltage: 2.23 mV
11:59:58.918 -> Load Voltage: 8.00 V
11:59:58.984 -> Current: 22.50 mA
11:59:58.984 -> Power: 180.05 mW
11:59:58.984 -> Capacity: 0.02 Ah
11:59:58.984 ->

```

Vista general del Arduino enviando a través de interfaz serial

```
134 Serial.begin(9600);
135 // Initialize oled display
136 display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
137 display.clearDisplay();
138 display.setTextColor(WHITE);
139 display.setTextSize(2);
140 display.setCursor(0, 0);
141 display.print("Multimetro");
142 display.drawImage(40, 20, image_data, 44, 45, 1);
143 display.display();
144
145 // Inicia INA219.
146
147 if (!ina219.begin()) {
148   Serial.println("Error encontrando INA219 chip");
149   while (1) {
150     delay(10);
151   }
152 }
153
154 if (!rtc.begin()) {
155   Serial.println("Error en el reloj");
156   while (1);
157 }
```

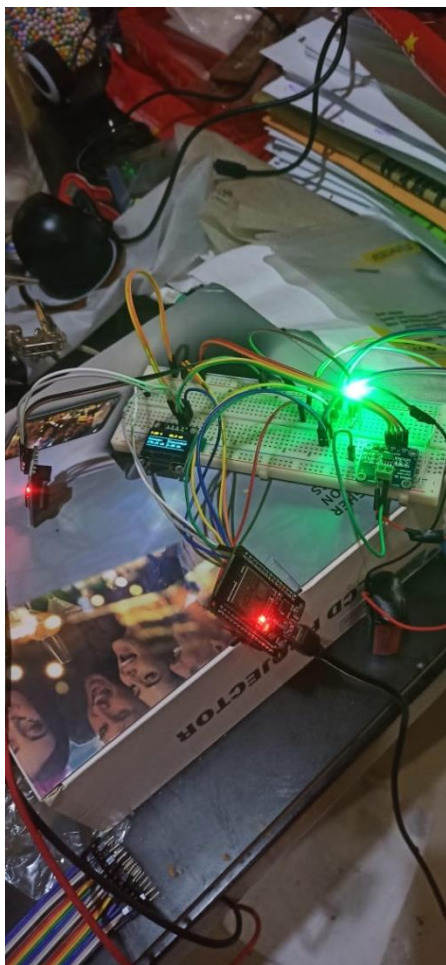
Output Serial Monitor x

Message [Ctrl + Enter] to send message to 'ESP32 Dev Module' on 'COM4'

New Line 9600 baud

```
12:00:44.996 -> Capacity: 0.54 Ah
12:00:45.029 -> -----
12:00:46.861 -> Bus Voltage: 7.93 V
12:00:46.925 -> Shunt Voltage: 2.23 mV
12:00:46.925 -> Load Voltage: 7.93 V
12:00:46.991 -> Current: 22.50 mA
12:00:46.991 -> Power: 178.43 mW
12:00:46.991 -> Capacity: 0.56 Ah
12:00:46.991 -> -----
```

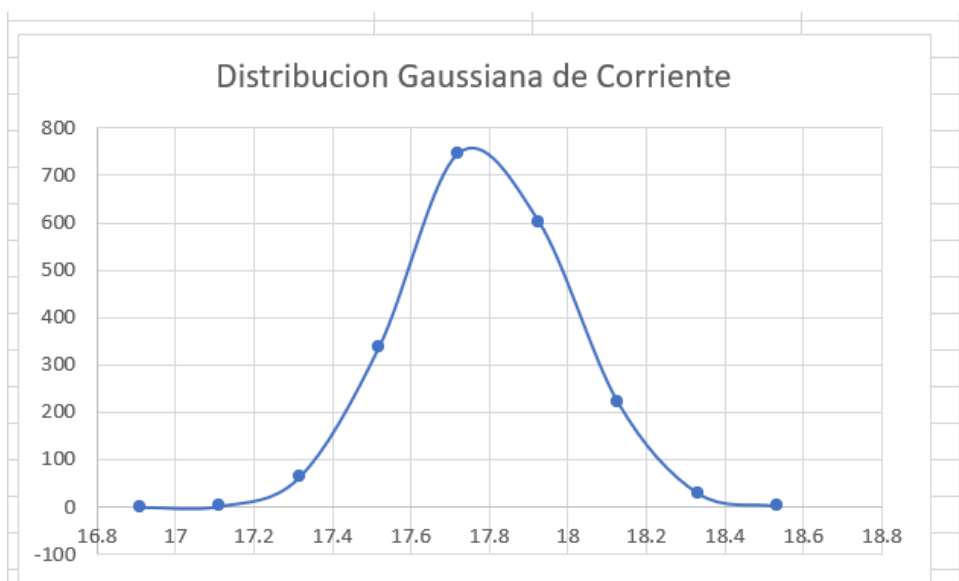
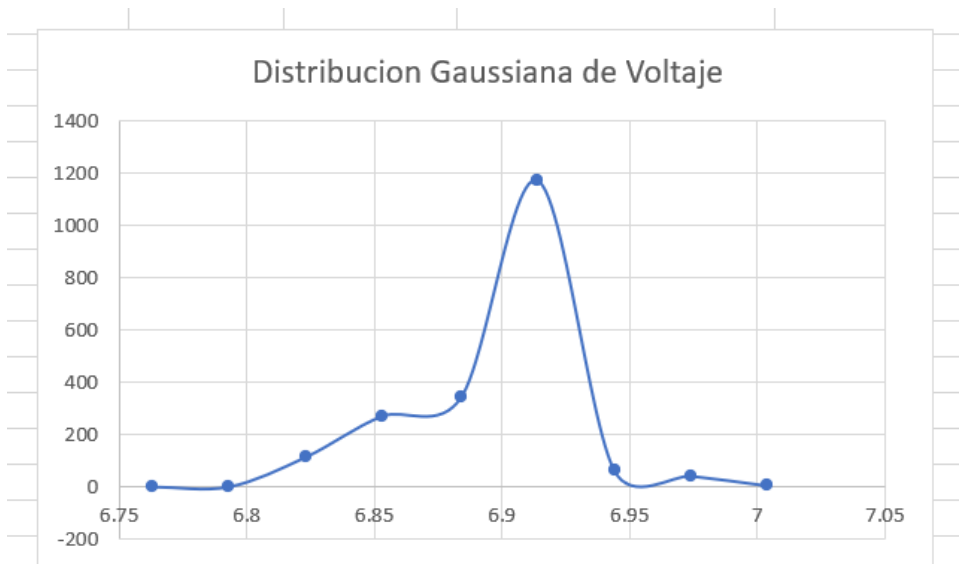
Vista del circuito completo con los datos

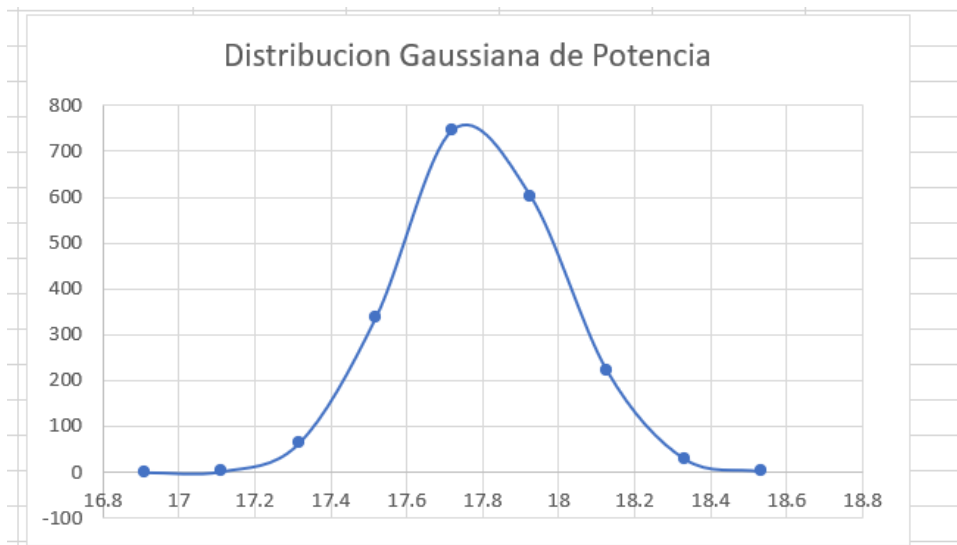


Plataforma blynk:



Graficas sacadas de las tablas de información:





Max:	Valor	Promedio	Valores	CURTOSIS	Media
Voltaje(V):	6.99	Voltaje(V):	6.884	0.62569013	6.89
Amperio(mA):	18.4	Amperio(mA):	17.721	-0.15828778	17.7
Potencia(mW):	128.54	Potencia(mW):	121.989	0.08265324	127.84
Min:	Valor	Varianza	Valores		
Voltaje(V):	6.81	Voltaje(V):	0.001		
Amperio(mA):	17.1	Amperio(mA):	0.041		
Potencia(mW):	116.51	Potencia(mW):	3.234		
		Desviacion estandar	Valores		
En variacion en 10 intervalos	Valor	Voltaje(V):	0.03021388		
Voltaje	0.018	Amperio(mA):	0.20322488		
Amperio	0.13	Potencia(mW):	1.79775819		
Potencia	1.203				
		n	2004		
		Porcentaje de error	z	Error	Precision(%)
		Voltaje	2.52722571	0.0017057	99.8294304
		corriente	1.86317102	0.00845825	99.15417455
		potencia	1.84740508	0.07418986	92.58101399

Conclusiones de la pregunta 1:

- ✓ Se logró utilizar e implementar módulos de desarrollo con sistemas embebidos SoC de 32 bits ESP32 con conectividad Wi-Fi a la plataforma IOT Blynk para desarrollar este proyecto.
- ✓ Se comprobó la evaluación de los parámetros de voltaje, amperio, potencia y capacidad en un circuito simple conformado por una resistencia, una batería y un led.
- ✓ Este circuito puede servir para analizar cuanto se gasta con el consumo al circuito que se analizara con el módulo ina
- ✓ Los datos sacados del modulo SD se uso para saber el porcentaje de precisión de los sensores demostrando que los datos brindados son de alta precisión.