

### GUÍA N° 3 – ADQUISICIÓN DE SEÑALES Y GRAFICACIÓN EN ARDUINO

#### 1. OBJETIVOS ESPECÍFICOS DE LA PRÁCTICA

- Adquirir señales conocidas como señal cuadrada, triangular, senoidal, rampa, etc.
- Entender los criterios de selección de la frecuencia de muestreo.
- Manipular y configurar adecuadamente una fuente de alimentación regulable; multímetro digital; Generador de señales y osciloscopio digital.

#### 2. MATERIALES Y EQUIPOS

Equipo   Materiales		Cantidad
Modelo	Descripción	
AFG1022	Generador de Señales	1
TBS 1000C Series	Osciloscopio Digital	1
-	Cable BNC Male-Male	1
-	Punta de osciloscopio con conector BNC (Male)	1
-	Par de cables Male -Male	1
SAMD	Arduino 33 IoT	1

#### 3. PAUTAS DE SEGURIDAD

- Asegurarse que los equipos y material a utilizar se encuentren en una superficie firme y a una distancia prudente de las esquinas o bordes de la superficie.
- No manipular los equipos y/o suministros con las manos mojadas o húmedas.
- Los alumnos deben maniobrar los equipos de acuerdo con las indicaciones del docente y los contenidos en esta guía.
- Si nota algún deterioro físico como fisuras en los cables, abolladuras en la carcasa del equipo o casos similares reportarlo al docente encargado de la sesión del curso.
- Verificar el voltaje de funcionamiento de los equipos (si se conecta 110v o 220v). Si el equipo y/o instrumento no trabaja apropiadamente, comunicar al docente encargado de la sesión. No intentar reparar sin supervisión.
- Usar equipos de protección personal (lentes, guantes y/o de zapatos de seguridad con suela de alta resistencia eléctrica) cuando sea necesario.
- Respetar el tiempo y espacio de trabajo de cada miembro del equipo, sin perturbar su concentración mediante conversaciones inadecuadas, groserías y música a alto volumen.
- Si el alumno o grupo de trabajo no están seguros sobre algún paso presentado en la guía, referente a la manipulación de los equipos, deberán consultarlo al docente encargado de la sesión.
- Si algún equipo y/o suministro sufriera daño debido a un uso inadecuado (se entiende por “uso inadecuado” cuando el equipo es usado en situaciones y pasos ajenos a las indicaciones del docente), el grupo de trabajo responsable (presentado en la ficha de solicitud de materiales) deberá reponer dicho equipo y/o suministro.
- Asegúrese que el equipo se encuentra en las mismas condiciones tanto de funcionamiento como físicas como cuando se le entregó.

## 4. FUNDAMENTOS

### 4.1 Arduino IOT

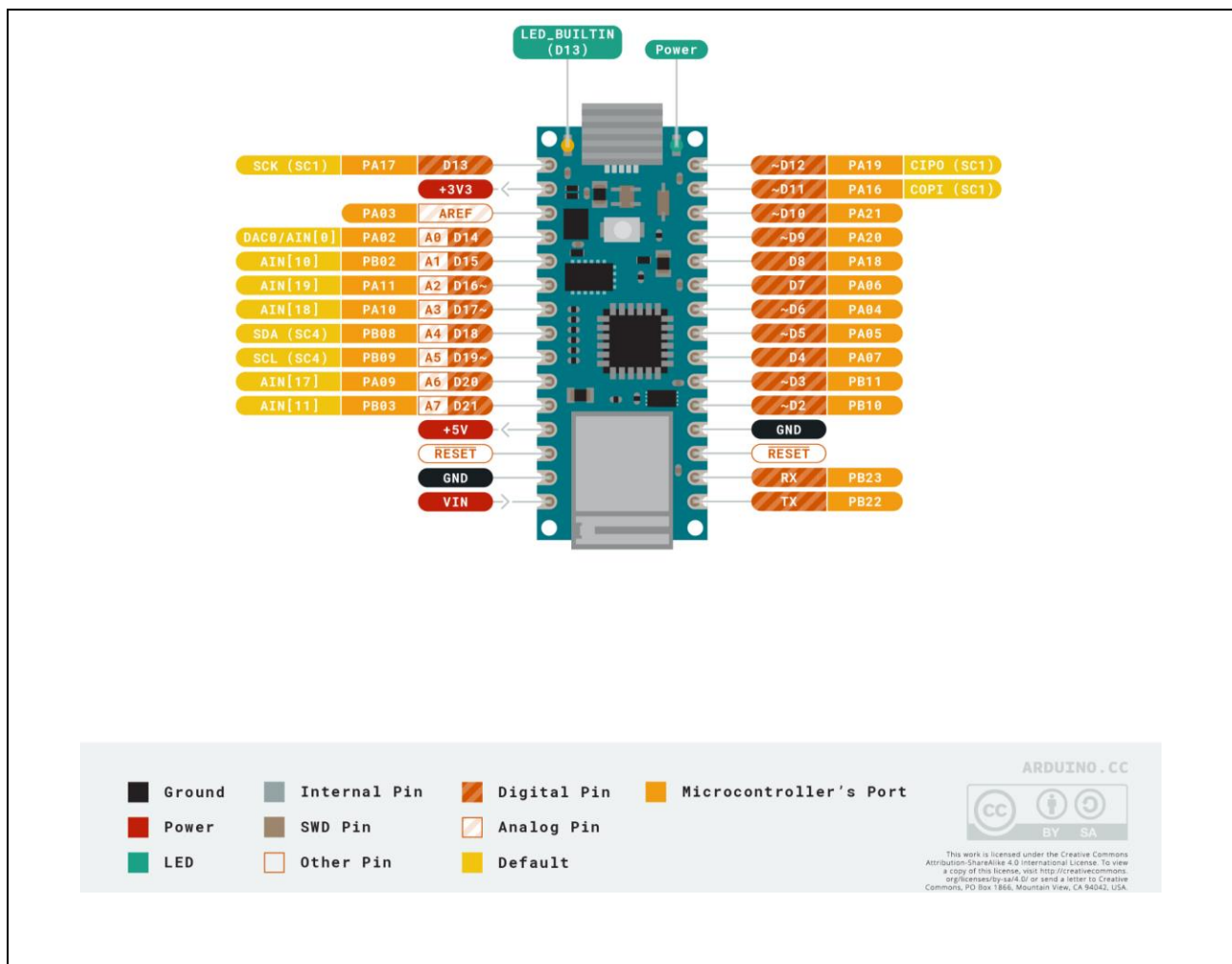
El Arduino Nano 33 IoT es la placa más pequeña de Arduino para comenzar con Internet de las cosas (IoT). Con el popular procesador Arm® Cortex®-M0 SAMD21 de 32 bits, también cuenta con el potente módulo Wi-Fi u-blox NINA-W102 y el chip criptográfico ECC608A para seguridad.

#### Main chip

- CPU and Memory
- ARM® Cortex®-M0+ CPU ATSAMD21G18A running at up to 48 MHz
- Single-cycle hardware multiplier
- Micro Trace Buffer (MTB)
- 256 KB in-system self-programmable Flash
- 32 KB SRAM Memory
- System
- Power-on Reset (POR) and Brown-out Detection (BOD)
- Internal clock 32.768 kHz (RTC), 48 MHz
- 16 external interrupts
- Two-pin Serial Wire Debug (SWD) programming, test and debugging interface
- Low Power
- Idle and Stand-by Sleep modes
- SleepWalking peripherals

#### Peripherals

- 12-channel Direct Memory Access Controller (DMAC)
- 12-channel Event System
- 32-bit Real Time Counter (RTC) with clock/calendar function
- CRC-32 generator
- Up to six Serial Communication Interfaces (SERCOM), each configurable for various protocols.
- One two-channel Inter-IC Sound (I2S) interface
- One 12-bit, 350ksps Analog-to-Digital Converter (ADC) with up to 20 channels
- 10-bit, 350 ksps Digital-to-Analog Converter (DAC)
- Peripheral Touch Controller (PTC)
- 256-Channel capacitive touch and proximity sensing
- PWM Pins: 11 (2, 3, 5, 6, 9, 10, 11, 12, 16 / A2, 17 / A3, 19 / A5)
- UART: 1
- SPI: 1
- I2C: 1
- I2S: 1
- Connectivity: WiFi
- Analog Input Pins: 8 (ADC 8/10/12 bit)
- Analog Output Pins: 1 (DAC 10 bit)
- External Interrupts: All digital pins (all analog pins can also be used as interrupt pins, but will have duplicated interrupt numbers)
- LED\_BUILTIN: 13



## 5. PROCEDIMIENTOS

Identifique y organice su grupo de trabajo. Verifique el estado de los equipos de protección personal (EPP).

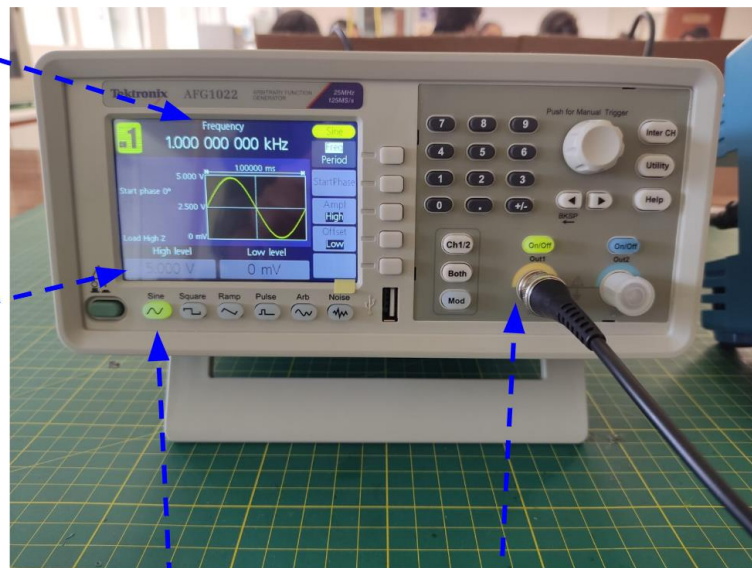
### 5.1. Uso del Generador de Señales y Osciloscopio

- Encender el Generador de Señales y el Osciloscopio
- Configurar el Generador de Señales para proporcionar una señal sinusoidal de 1 KHz de frecuencia, 5V de Amplitud y 0V de offset, por el canal 1.
- Conectar un extremo del cable BNC en el canal 1 del generador de señales y el otro extremo en el canal 1 del osciloscopio.
- Mediante los controles de Posición Vertical, Horizontal y Disparo ajustar la visualización de la señal sinusoidal.
- Haciendo uso de los cursores, calcular y mostrar en el osciloscopio las medidas de Amplitud y Periodo de la señal.

Ejemplo de prueba:

Frecuencia de 1 KHz seleccionada en el canal 1

Offset de 0V y amplitud de 5V ingresado



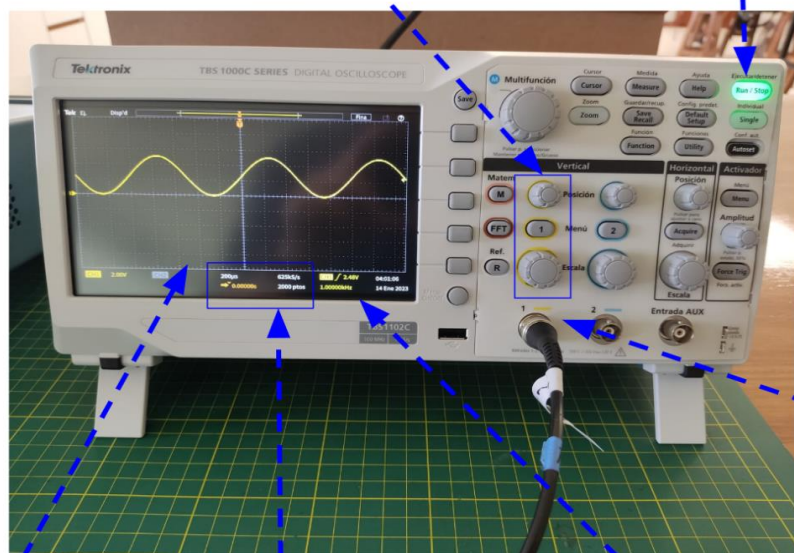
Señal sinusoidal seleccionada

Salida del canal 1 conectada y activa

El offset en un generador de señales es una función que permite ajustar el valor de corriente continua (DC) que se agrega a una señal de corriente alterna (AC). Esto significa que el valor de offset se utiliza para mover el nivel de la señal hacia arriba o hacia abajo en el eje vertical. Esto es útil para calibrar la señal y asegurarse de que esté dentro del rango adecuado para el circuito o dispositivo que se está probando. Por ejemplo, si se está probando un amplificador que solo funciona con señales positivas, se puede ajustar el offset para que la señal siempre sea positiva. Para este ejemplo se está usando una señal positiva.

Ajuste de posición Vertical del canal 1

Osciloscopio activo



Gráfica de la señal recepcionada

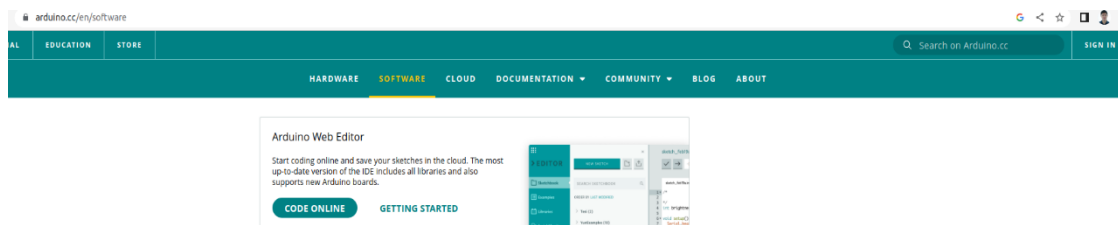
Valores ajustados de gráfica

Valores medidos (Voltaje pico y frecuencia) de la señal de entrada


## 5.2 Uso del arduino IoT

Para empezar a programar el Arduino, necesitarás seguir los siguientes pasos:

- A. Descarga el software: Puedes descargar el software de programación de Arduino de forma gratuita desde el sitio web oficial de Arduino. El software se llama Arduino IDE (Integrated Development Environment) y está disponible para Windows, Mac OS y Linux.
- B. Conecta el Arduino a tu computadora: Una vez que tengas tu placa Arduino y el software instalado, conecta la placa a tu computadora a través de un cable USB.
- C. Abre el software Arduino IDE: Abre el software Arduino IDE en tu computadora. Asegúrate de que la placa esté conectada y selecciona el puerto COM correcto en la herramienta del menú "Herramientas".
- D. Para el caso del arduino IoT, se debe ir a **Herramientas > Boards > Board Manager > arduino SAMD**
- E. Crea un nuevo sketch: Un sketch de Arduino es un programa que puedes cargar en la placa Arduino para controlar sus funciones. Crea un nuevo sketch en Arduino IDE y comienza a escribir el código.
- F. Carga el programa en la placa: Cuando hayas terminado de escribir el código, carga el programa en la placa Arduino haciendo clic en el botón "Subir" en el software Arduino IDE.
- G. Observa la salida: La placa Arduino tiene varios pines que pueden ser utilizados para enviar y recibir señales. Conecta los dispositivos que quieras controlar a los pines correspondientes y observa la salida de tu programa.



### Downloads



#### Arduino IDE 2.0.4

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

**SOURCE CODE**  
The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

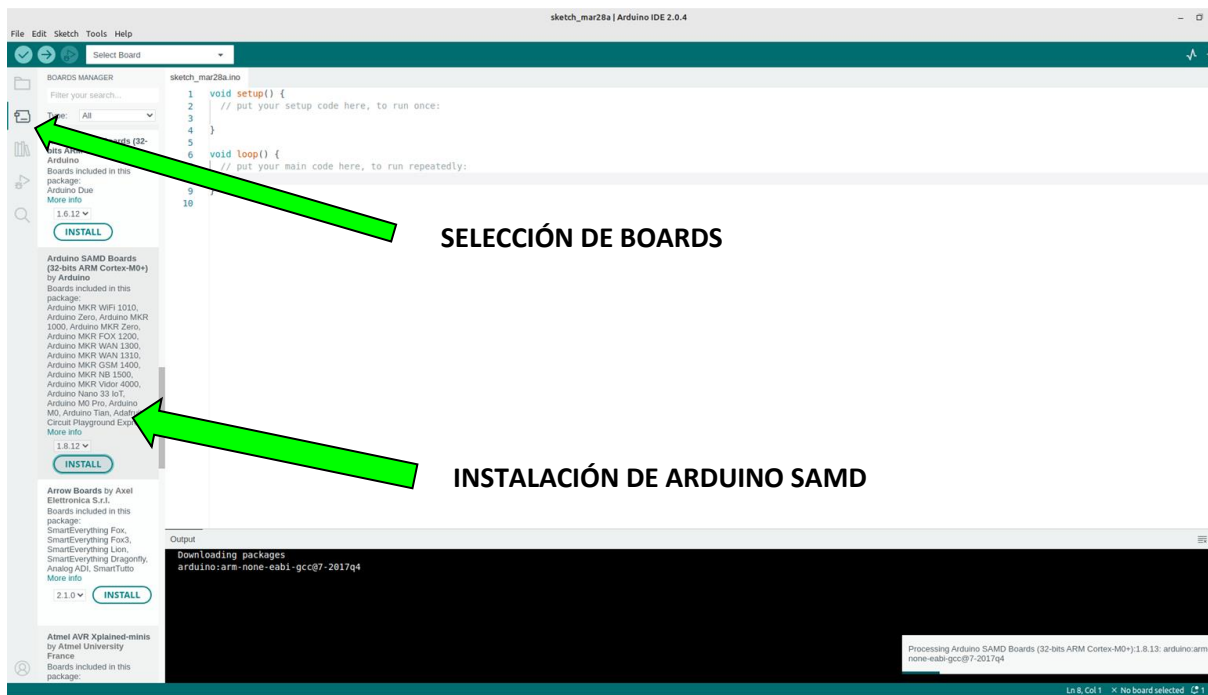
#### DOWNLOAD OPTIONS

**Windows** Win 10 and newer, 64 bits  
**Windows** MSI installer  
**Windows** ZIP file  
**Linux** AppImage 64 bits (X86-64)  
**Linux** ZIP file 64 bits (X86-64)  
**macOS** Intel, 10.14: "Mojave" or newer, 64 bits  
**macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

Nightly Builds

Help



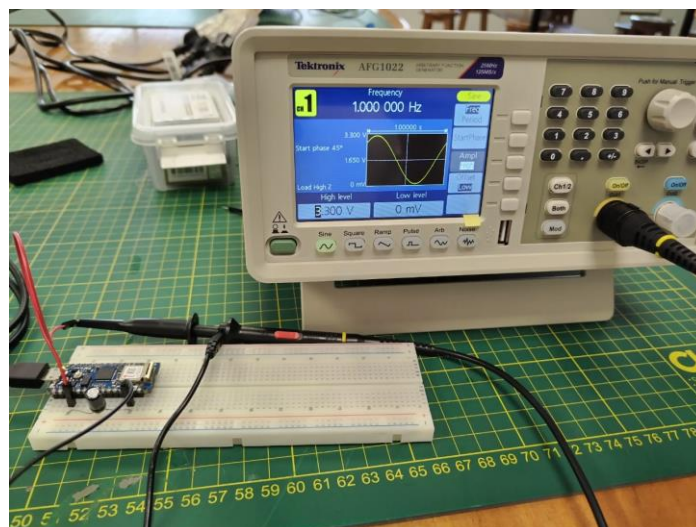
Con estos pasos, puedes comenzar a programar el Arduino y crear tus propios proyectos electrónicos interactivos.

## 6. Entregables

- Plotear al menos 3 señales en Arduino IDE provenientes del generador de señales.
- Comparar las señales graficadas del Arduino IDE con las gráficas obtenidas del osciloscopio.
- Graficar en Arduino cloud.

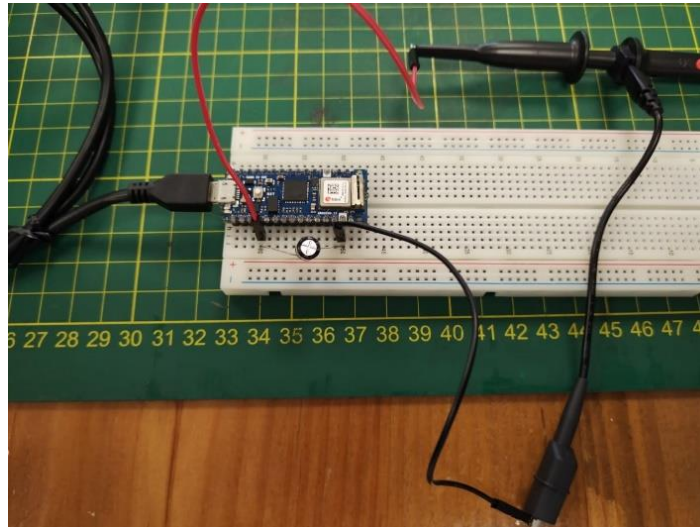
Figuras de referencia:

### 1. Conexión Generador de señales y Arduino nano 33 IoT

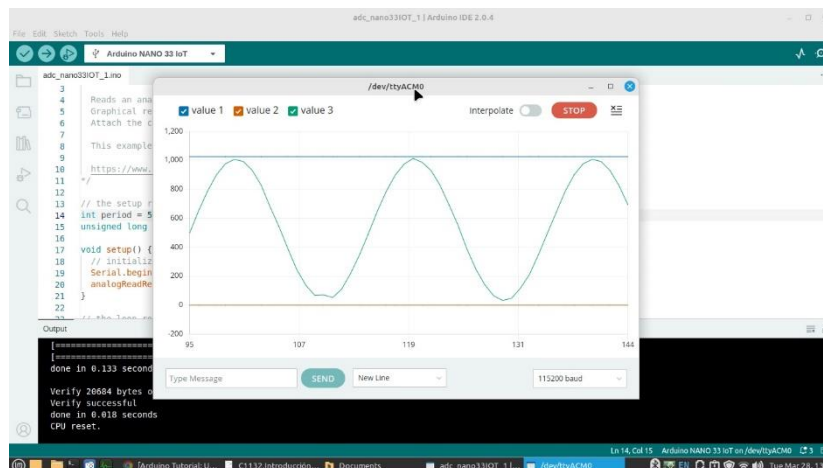




## 2. Conexión Arduino nano 33 IoT y cable BNC



## 3. Ploteo de señal sinusoidal enviada desde el generador de señales



## 4. Ejemplo de ploteo de una señal creada artificialmente determinada por una frecuencia

```
7 unsigned long lastMsg = 0;
8 float F=1; // 1 hz
9 double Fs = 10*F; // 10 hz
10 double Ts_ms = (1/Fs)*1000; // 100 ms
11
12 void setup() {
13   Serial.begin(9600);
14   while (!Serial);
15 }
16
17 void loop() {
18
19   unsigned long now = millis();
20   if (now - lastMsg > Ts_ms) {
21
22     lastMsg = now;
23
24     int r1 = random(10);
25     int r2 = random(10);
26
27     Serial.print("Señal1:");
28     Serial.print(r1);
29     Serial.print(",");
30     Serial.print("Señal2:");
31     Serial.println(r2);
32   }
```