Instalación del entorno de desarrollo

1 Introducción

Para los laboratorios vamos a usar placas de desarrollo Arduino Zero a las que les conectaremos módulos de comunicación y antenas. Estas placas de desarrollo están basadas en el microcontrolador **SAMD21** de **Atmel** (ARM Cortex M0+ de 32 bit). La forma más sencilla de programarlos es utilizar el entorno de desarrollo (**IDE**) de Arduino con su propio lenguaje (lenguaje Arduino: una mezcla de C++, Wiring, Processing y mucha creatividad). En este curso utilizaremos el IDE de Arduino versión 2 que incluye una interfaz, sencilla pero potente, para depurar (debuggear) software embebido.

Advertencia: Esta placa de Arduino, a diferencia de otras, utiliza tensiones de alimentación máximas de 3,3 V. Colocar una tensión mayor a ésta en cualquiera de sus pines puede romperla.

2 Instalación

2.1 Instalación de Arduino IDE en Linux:

Descargar la versión más nueva del software "**Arduino IDE 2.x.x**" del sitio web de Arduino (2.3.3 al momento de escribir esta guía). En el caso de Linux lo más sencillo es descargar la "**AppImage 64 bits**", y colocar el archivo en un lugar de cómodo acceso (el escritorio por ejemplo).

Luego, se debe agregar permiso de ejecución y ejecutarlo:

```
# Agregar permiso de ejecución del archivo:
chmod a+x arduino-ide_2.3.2_Linux_64bit.AppImage
# Ejecutarlo:
./arduino-ide_2.3.2_Linux_64bit.AppImage
```

En Linux además hay que agregar las reglas *udev*, para eso descargar el archivo[1] *99-platformioudev.rules* y copiarlo (como super usuario) en /etc/udev/rules.d/99-platformio-udev.rules para luego reiniciar el sistema udev de alguna de estas dos maneras:

sudo service udev restart

o

[1] Por comodidad vamos a utilizar el archivo de reglas del proyecto PlatformIO que contiene reglas y configuraciones para una amplia variedad de microcontroladores y placas de desarrollo.

sudo udevadm control --reload-rules
sudo udevadm trigger

Finalmente hay que agregar el usuario al grupo correspondiente:

Ubuntu/Debian:
sudo usermod -a -G dialout $USER
sudo usermod -a -G plugdev $USER

Arch:
sudo usermod -a -G uucp $USER
sudo usermod -a -G lock $USER

2.2 Instalación de Arduino IDE en Windows:

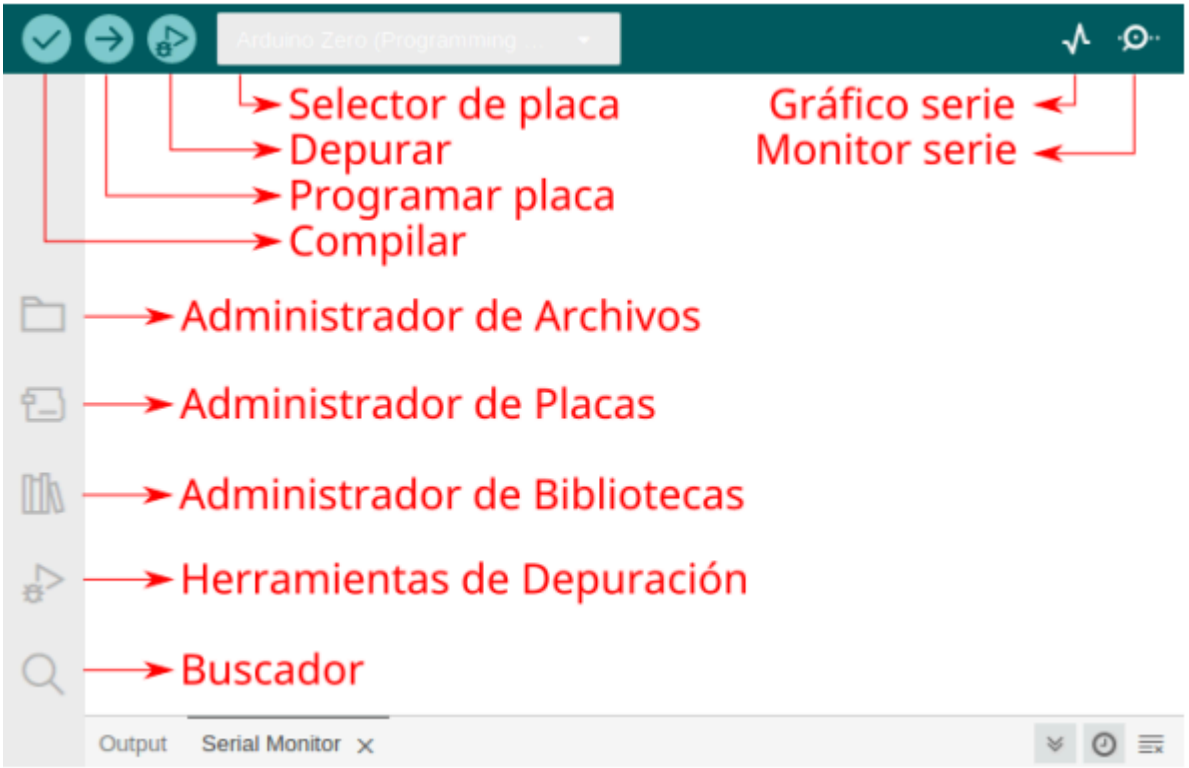
La instalación en Windows 10 u 11 se realiza siguiendo los siguientes pasos:

1. Conectar la placa de desarrollo Arduino Zero al puerto USB de la computadora. Windows automáticamente instalará los drivers necesarios. Para comprobar se debe entrar en Configuración/Dispositivos; mientras el sistema instala automáticamente los drivers aparecerá "Dispositivo desconocido", mientras que cuando haya finalizado la instalación, aparecerá **"Arduino Zero"** o **"EDBG CMSIS-DAP"** (dependiendo el puerto USB que se haya utilizado).
2. Descargar la versión más nueva del software "**Arduino IDE 2.x.x**" del sitio web de Arduino (2.0.4 al momento de escribir esta guía). Elegir la opción "Windows Win 10 and newer, 64 bits". Guardar el archivo **arduino-ide_2.0.4_Windows_64bit.exe** en la carpeta de Descargas.
3. Finalmente buscar el archivo en la carpeta de Descargas, hacer doble clic sobre él y comenzar el proceso de instalación.

3 Primeros pasos con el entorno de desarrollo:

Ejecutar el software Arduino IDE. La primera vez que se ejecuta puede demorar un tiempo considerable debido a que se descarga varias bibliotecas y configuraciones.

La siguiente figura muestra un esquema de la interfaz de usuario del Arduino IDE:



Ahora podemos conectar la placa Arduino Zero utilizando un cable USB-microUSB.

Nota: En la placa Arduino el conector que utilizaremos será el que dice Programming, ubicado en la posición más alejada del botón

3.1 Instalación de librerías y placas:

Para utilizar la Arduino Zero debemos instalar el conjunto de herramientas para programarla. Puede que esto se haga de forma automática. En caso contrario utilizaremos el administrador de placas (*boards manager*), buscamos *Arduino Zero* e instalamos el archivo **Arduino SAMD Boards (32-bits ARM Cortex-M0+)**. Esto agregará en el *selector de placa* la opción *Arduino Zero (Programming Port)* cuando la placa esté conectada. La seleccionaremos para comenzar a utilizarla. En los Laboratorios utilizaremos diversas bibliotecas de software para interactuar con hardware; como ejemplo vamos a leer la temperatura interna del microcontrolador. Para ello utilizando el administrador de bibliotecas (*library manager*), buscamos e instalamos el archivo **TemperatureZero**.

3.2 Uso básico del IDE:

Una vez conectada la placa y seleccionada su opción correspondiente en el *selector de placas* vamos a abrir el ejemplo para leer la temperatura interna del microcontrolador. Yendo a **File->Examples->TemperatureZero->Example1_BasicTemperatureReading** se abrirá una nueva instancia del IDE.

Lo primero cuando abrimos el código es entender a grandes rasgos qué hace. En general busca ser intuitivo pero podemos necesitar una referencia. Por ejemplo, si queremos conocer detalles de la función **delay()** podemos hacer click derecho sobre el nombre de la función y luego ir a *Go to Definition*. Otra opción es ir a la referencia de lenguaje Arduino: delay().

Para probar el código del ejemplo, podemos hacer click en *Compilar (Verify)*, ver en la salida de la consola (*Output*) que dice **Compilation complete**, y luego programar la placa usando el botón *Programar placa (Upload)* obteniendo finalmente la confirmación en la consola *upload complete*.

Si no hubo errores en el proceso anterior, se puede abrir el monitor serial (lupita que está arriba a la derecha) y ver la temperatura interna del microcontrolador.

Si esto funciona correctamente, también se puede ver la información en forma gráfica, abriendo el gráfico serial en **Tools->Serial Plotter**.

3.3 Depurando con el Arduino IDE:

Para depurar (*debuggear*) el software es necesario tener en cuenta dos cosas:

- Se recomienda optimizar el compilado para usar el depurador. Esto implica en el menú **Sketch** seleccionar la opción **Optimize for Debugging**.
- Siempre se debe compilar (*verify*) antes de depurar. Dicho de otro modo, el programa que se *depura* es el último que fue compilado.