



## **Bienvenido.**

Gracias por adquirir nuestro *Módulo Sensor de Gas AZ-Delivery MQ-135*.  
En las páginas siguientes, se le explicará cómo utilizar y configurar  
este práctico dispositivo.

**¡Que te diviertas!**



## Áreas de aplicación

Educación y enseñanza: uso en escuelas, universidades e instituciones de formación para enseñar los conceptos básicos de electrónica, programación y sistemas integrados. Investigación y desarrollo: Uso en proyectos de investigación y desarrollo para crear prototipos y experimentos en los campos de la electrónica y la informática. Desarrollo de prototipos: Uso en el desarrollo y prueba de nuevos circuitos y dispositivos electrónicos. Proyectos Hobby and Maker: utilizado por entusiastas y aficionados a la electrónica para desarrollar e implementar proyectos de bricolaje.

## Conocimientos y habilidades requeridos.

Conocimientos básicos de electrónica e ingeniería eléctrica. Conocimientos de programación, especialmente en el lenguaje de programación C/C++. Capacidad para leer esquemas y diseñar circuitos simples. Experiencia trabajando con componentes electrónicos y soldadura.

## Condiciones de operación

El producto sólo puede funcionar con los voltajes especificados en la hoja de datos para evitar daños. Se requiere una fuente de alimentación CC estabilizada para su funcionamiento. Al realizar la conexión a otros componentes y circuitos electrónicos, se deben observar los límites máximos de corriente y voltaje para evitar sobrecargas y daños.

## Condiciones ambientales

El producto debe utilizarse en un ambiente limpio y seco para evitar daños causados por la humedad o el polvo. Proteger el producto de la luz solar directa (UV)

## Uso previsto

El producto está diseñado para su uso en entornos educativos, de investigación y desarrollo. Se utiliza para desarrollar, programar y crear prototipos de proyectos y aplicaciones electrónicos. El producto Sensor no pretende ser un producto de consumo terminado, sino más bien una herramienta para usuarios con conocimientos técnicos, incluidos ingenieros, desarrolladores, investigadores y estudiantes.

## Uso inadecuado previsible

El producto no es adecuado para uso industrial o aplicaciones relevantes para la seguridad. No se permite el uso del producto en dispositivos médicos o para fines de aviación y viajes espaciales.

## desecho

¡No lo deseches con la basura doméstica! Su producto es acorde al europeo. Directiva sobre residuos de aparatos eléctricos y electrónicos que deben eliminarse de forma respetuosa con el medio ambiente. Las valiosas materias primas contenidas en ellos se pueden reciclar. La aplicación de esta directiva contribuye a la protección del medio ambiente y la salud. Utilice el punto de recogida habilitado por su municipio para devolver y Reciclaje de aparatos eléctricos y electrónicos viejos. N.º registro RAEE: DE 62624346

## descarga electrostática

Atención: Las descargas electrostáticas pueden dañar el producto. Nota: Conéctese a tierra antes de tocar el producto, por ejemplo usando una muñequera antiestática o tocando una superficie metálica conectada a tierra.

## instrucciones de seguridad

Aunque nuestro producto cumple con los requisitos de la Directiva RoHS (2011/65/UE) y no contiene sustancias peligrosas en cantidades superiores a los límites permitidos, es posible que aún queden residuos. Observe las siguientes instrucciones de seguridad para evitar riesgos químicos: Precaución: La soldadura puede producir humos que pueden ser perjudiciales para la salud. Nota: Utilice un extractor de humos de soldadura o trabaje en un área bien ventilada. Si es necesario, use una máscara respiratoria. Precaución: algunas personas pueden ser sensibles a ciertos materiales o químicos contenidos en el producto. Nota: Si se produce irritación de la piel o reacciones alérgicas, suspenda su uso y, si es necesario, consulte a un médico. Precaución: Mantenga el producto fuera del alcance de los niños y las mascotas para evitar el contacto accidental y la ingestión de piezas pequeñas. Nota: Guarde el producto en un recipiente cerrado y seguro cuando no esté en uso. Atención: Evite el contacto del producto con alimentos y bebidas. Nota: No almacene ni utilice el producto cerca de alimentos para evitar la contaminación. Aunque nuestro producto cumple con los requisitos de la Directiva RoHS (2011/65/UE) y no contiene sustancias peligrosas en cantidades superiores a los límites permitidos, es posible que aún queden residuos. Observe las siguientes instrucciones de seguridad para evitar riesgos químicos: Precaución: La soldadura puede producir humos que

pueden ser perjudiciales para la salud. Nota: Utilice un extractor de humos de soldadura o trabaje en un área bien ventilada. Si es necesario, use una máscara respiratoria. Precaución: algunas personas pueden ser sensibles a ciertos materiales o químicos contenidos en el producto. Nota: Si se produce irritación de la piel o reacciones alérgicas, suspenda su uso y, si es necesario, consulte a un médico. Precaución: Mantenga el producto fuera del alcance de los niños y las mascotas para evitar el contacto accidental y la ingestión de piezas pequeñas. Nota: Guarde el producto en un recipiente cerrado y seguro cuando no esté en uso. Atención: Evite el contacto del producto con alimentos y bebidas. Nota: No almacene ni utilice el producto cerca de alimentos para evitar la contaminación. El producto contiene componentes electrónicos sensibles y bordes afilados. Un manejo o montaje inadecuado puede provocar lesiones o daños. Observe las siguientes instrucciones de seguridad para evitar riesgos mecánicos: Atención: La placa de circuito y los conectores del producto pueden tener bordes afilados. Tenga cuidado para evitar cortes. Nota: Utilice guantes protectores adecuados al manipular y montar el producto. Precaución: Evite una presión excesiva o tensión mecánica en la placa y los componentes. Nota: Monte el producto únicamente en superficies estables y planas. Utilice espaciadores y carcasas adecuados para minimizar la tensión mecánica. Atención: Asegúrese de que el producto esté bien sujeto para evitar resbalones o caídas accidentales. Nota: Utilice un soporte adecuado o un montaje seguro en gabinetes o en placas de montaje. Precaución: Asegúrese de que todas las conexiones de los cables estén conectadas de forma segura y correcta para evitar tensiones y desenchufes accidentales. Nota: Tienda los cables de manera que no estén bajo tensión y no representen un peligro de tropiezo. El producto funciona con voltajes y corrientes eléctricas que, si se usan incorrectamente, pueden provocar descargas eléctricas, cortocircuitos u otros peligros. Observe las siguientes instrucciones de seguridad para evitar riesgos eléctricos: Atención: Utilice el producto únicamente con los voltajes especificados. Nota: Los límites de rendimiento del producto se pueden encontrar en la hoja de datos asociada. Precaución: Evite cortocircuitos entre los conectores y componentes del producto. Nota: Asegúrese de que ningún objeto conductor toque o puente la placa de circuito. Utilice herramientas aisladas y preste atención a la disposición de las conexiones. Precaución: No realice ningún trabajo en el producto cuando esté conectado a una fuente de alimentación. Nota: Desconecte el producto de la alimentación antes de realizar cambios en el circuito o conectar o quitar componentes. Precaución: No exceda las clasificaciones actuales especificadas para las entradas y salidas del producto. Nota: Los límites de rendimiento del producto se pueden encontrar en las especificaciones técnicas o en la ficha técnica. Atención: Asegúrese de que las fuentes de alimentación utilizadas sean estables y del tamaño correcto. Nota: Utilice únicamente fuentes de alimentación probadas y adecuadas para evitar fluctuaciones de voltaje y sobrecargas. Atención: Mantenga una distancia suficiente de las partes vivas para evitar el contacto accidental. Nota: Asegúrese de que el cableado esté dispuesto de forma segura y clara según el voltaje utilizado. Precaución: Utilice carcasas aislantes o cubiertas protectoras para proteger el producto del contacto directo. Nota: Coloque el producto en un estuche no conductor para evitar contactos accidentales y cortocircuitos. El producto y sus componentes pueden calentarse durante el funcionamiento. La manipulación inadecuada o la sobrecarga del producto pueden provocar quemaduras, daños o incendios. Observe las siguientes instrucciones de seguridad para evitar riesgos térmicos: Precaución: Asegúrese de que el producto se utilice dentro de las temperaturas de funcionamiento recomendadas. Nota: El rango de temperatura de funcionamiento recomendado suele estar entre -40 °C y +85 °C. Consulta la información específica en la ficha técnica del producto. Atención: No coloque el producto cerca de fuentes de calor externas como radiadores o luz solar directa. Nota: Asegúrese de que el producto funcione en un área fresca y bien ventilada. Atención: Asegúrese de que el producto esté bien ventilado para evitar el sobrecalentamiento. Nota: Utilice ventiladores o disipadores de calor cuando utilice el producto en un recinto cerrado o en un entorno con circulación de aire limitada. Atención: Monte el producto sobre superficies resistentes al calor y en carcasas resistentes al calor. Nota: Utilice materiales de gabinete que puedan soportar altas temperaturas para evitar daños o riesgos de incendio. Precaución: Implemente un control de temperatura cuando utilice un gabinete y, si es necesario, mecanismos de protección que apaguen el producto si se sobrecalienta. Nota: Utilice sensores de temperatura y software adecuado para controlar la temperatura del producto y apague el sistema si es necesario. Precaución: Evite sobrecargas que puedan causar un calentamiento excesivo de los componentes. Nota: Para evitar el sobrecalentamiento, no exceda los límites de corriente y voltaje especificados. Precaución: Los cortocircuitos pueden generar mucho calor y provocar incendios. Nota: Asegúrese de que todas las conexiones sean correctas y seguras y que ningún objeto conductor pueda causar cortocircuitos accidentalmente.



## Índice

|   |           |
|---|-----------|
| <b>Introducción</b>                                     | <b>3</b>  |
| <b>Especificaciones</b>                                 | <b>4</b>  |
| <b>El pinout</b>  | <b>5</b>  |
| <b>Cómo configurar la Raspberry Pi y Python</b>         | <b>10</b> |
| <b>Conexión del módulo con ATmega328p</b>               | <b>11</b> |
| <b>Ejemplo de croquis</b>                               | <b>12</b> |
| <b>Conexión de Nano V3.0 como ADC para Raspberry Pi</b> | <b>17</b> |
| <b>Conexión del módulo con Raspberry Pi</b>             | <b>21</b> |
| <b>Script en Python para el módulo MQ-135</b>           | <b>28</b> |



## Introducción

El módulo sensor de gas MQ-135 es un dispositivo que se utiliza para detectar y medir la concentración de gases en el aire. Puede detectar gases como: GLP, propano, metano, hidrógeno, alcohol, humo y monóxido de carbono. Aunque puede detectar esos gases, no es capaz de distinguir la diferencia entre ellos.

El MQ-135 es un semiconductor de óxido metálico (MOS), también conocido como quimiresistor. El sensor contiene un material sensor cuya resistencia cambia con diferentes concentraciones de gas. Este cambio de la resistencia se utiliza para la detección de gases. El sensor también tiene un potenciómetro incorporado, con el que podemos ajustar su sensibilidad.

El sensor está encerrado en dos capas de fina malla de acero inoxidable denominada *red antiexplosión*. Gracias a ello, es capaz de detectar gases inflamables sin incidentes. Asimismo, proporciona protección al sensor y filtra las partículas en suspensión. De este modo, sólo los gases pueden pasar al interior de la cámara de detección.

El módulo incorpora un chip comparador LM393 que convierte las lecturas en señales digitales y analógicas. También hay un potenciómetro que se utiliza para calibrar la sensibilidad de detección.



### Especificaciones

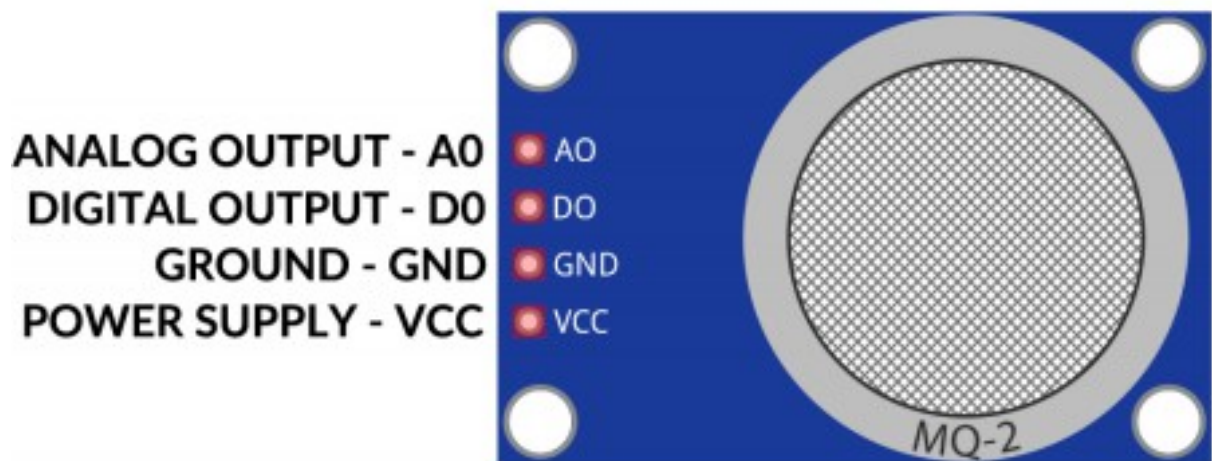
|                              |                                    |
|------------------------------|------------------------------------|
| Tensión de funcionamiento:   | 5V                                 |
| Corriente de funcionamiento: | 150 mA                             |
| Consumo de energía:          | 900mW                              |
| Resistencia a la carga:      | 20k $\Omega$                       |
| Resistencia del calentador:  | 33 $\Omega$ +5%                    |
| Resistencia de detección     | 10k $\Omega$ - 60k $\Omega$        |
| Tiempo de precalentamiento:  | 24h                                |
| Ámbito de concentración:     | 200 - 10000ppm (partes por millón) |
| Salida:                      | analógico, digital                 |
| Dimensiones:                 | 33x21x22mm (1.3x0.8x0.9in)         |

Para obtener los mejores resultados de detección, el sensor de gas debe precalentarse. El mejor tiempo de precalentamiento para el sensor es superior a 24 horas. Para obtener información detallada sobre las especificaciones del sensor, consulte la hoja de datos.

La sensibilidad del módulo puede ajustarse con un potenciómetro integrado. Moviendo el eje del potenciómetro en el sentido de las agujas del reloj aumenta la sensibilidad. Mover el eje del potenciómetro en el sentido contrario a las agujas del reloj disminuye la sensibilidad del módulo.

## El pinout

El módulo sensor de gas tiene cuatro patillas. El pinout se muestra en la siguiente imagen:



**NOTA:** La Raspberry Pi no tiene un convertidor digital-analógico y no se puede utilizar para leer voltajes analógicos.

## Cómo configurar Arduino IDE

Si el IDE de Arduino no está instalado, sigue el [enlace](#) y descarga el archivo de instalación para el sistema operativo que elijas.

### Download the Arduino IDE



The screenshot shows the Arduino IDE download page. On the left, there is a large teal circle containing the Arduino logo (an infinity symbol with a minus and plus sign). To the right of the logo, the text reads: **ARDUINO 1.8.9**. Below this, it says: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions." On the right side of the page, there is a teal sidebar with white text. It lists the following options: "Windows Installer, for Windows XP and up", "Windows ZIP file for non-admin install", "Windows app Requires Win 8.1 or 10" with a "Get" button, "Mac OS X 10.8 Mountain Lion or newer", "Linux 32 bits", "Linux 64 bits", "Linux ARM 32 bits", and "Linux ARM 64 bits". At the bottom of the sidebar, there are links for "Release Notes", "Source Code", and "Checksums (sha512)".

Para los usuarios de *Windows*, haga doble clic en el archivo .exe descargado y siga las instrucciones de la ventana de instalación.



# Az-Delivery

Para los usuarios de *Linux*, descarga un archivo con la extensión *.tar.xz*, que hay que extraer. Cuando esté extraído, ve al directorio extraído y abre el terminal en ese directorio. Hay que ejecutar dos scripts *.sh*, el primero llamado *arduino-linux-setup.sh* y el segundo llamado *install.sh*.

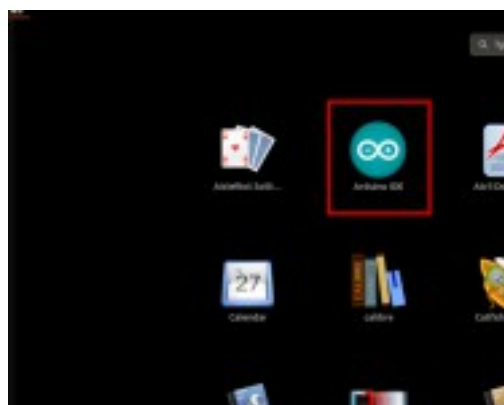
Para ejecutar el primer script en el terminal, abra el terminal en el directorio extraído y ejecute el siguiente comando:

**sh arduino-linux-setup.sh nombre\_usuario**

**nombre\_usuario** - es el nombre de un *superusuario* en el sistema operativo Linux. Se debe introducir una contraseña para el *superusuario* cuando se inicie el comando. Espera unos minutos a que el script lo complete todo.

El segundo script, llamado *install.sh*, debe utilizarse después de la instalación del primer script. Ejecute el siguiente comando en el terminal (directorio extraído): **sh install.sh**

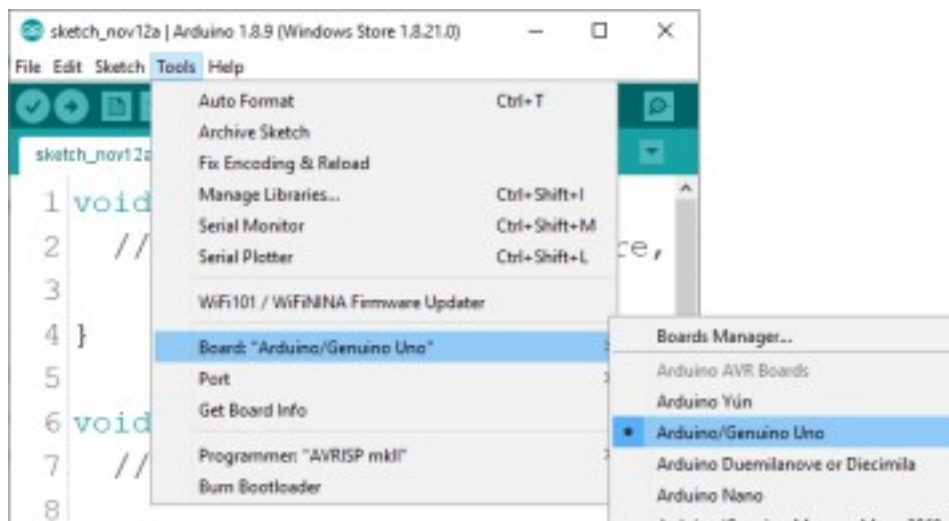
Después de la instalación de estos scripts, vaya a *Todas las aplicaciones*, donde está instalado el *IDE Arduino*.



# Az-Delivery

Casi todos los sistemas operativos vienen con un editor de texto preinstalado (por ejemplo, *Windows* viene con *Notepad*, *Linux Ubuntu* viene con *Gedit*, *Linux Raspbian* viene con *Leafpad*, etc.). Todos estos editores de texto son perfectamente adecuados para el propósito del eBook.

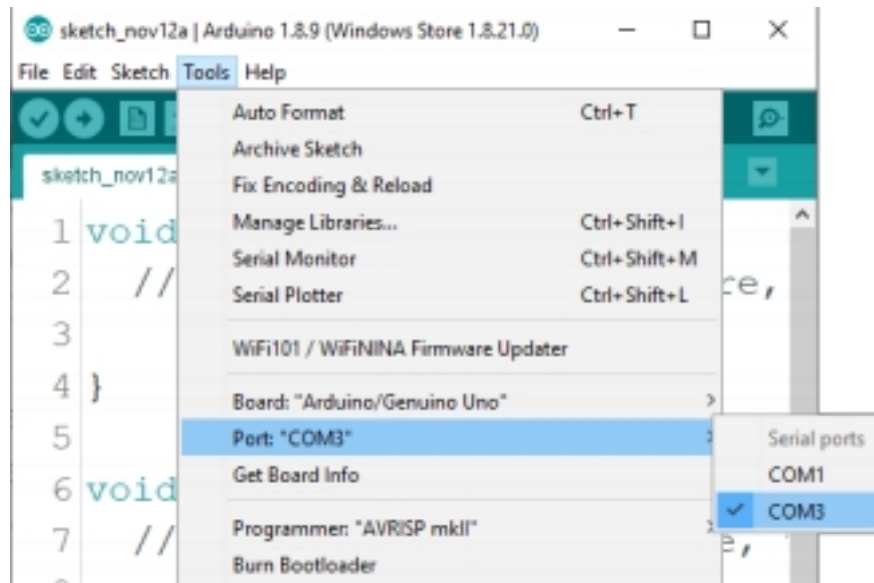
Lo siguiente es comprobar si tu PC puede detectar la placa microcontroladora. Abra recién instalado Arduino IDE, y vaya a:  
*Herramientas > Tablero > {nombre de su tablero aquí}*  
*{el nombre de tu placa aquí}* debería ser el *Microcontrolador/Genuino Uno*, como se puede ver en la siguiente imagen:



Hay que seleccionar el puerto al que está conectada la placa microcontroladora. Vaya a: *Herramientas > Puerto > {el nombre del puerto va aquí}*

y cuando la placa microcontroladora está conectada al puerto USB, se puede ver el nombre del puerto en el menú desplegable de la imagen anterior.

Si se utiliza el IDE Arduino en Windows, los nombres de los puertos son los siguientes:



Para los usuarios de *Linux*, por ejemplo, el nombre del puerto es */dev/ttyUSBx*, donde *x* representa un número entero entre 0 y 9.



## Cómo configurar la Raspberry Pi y Python

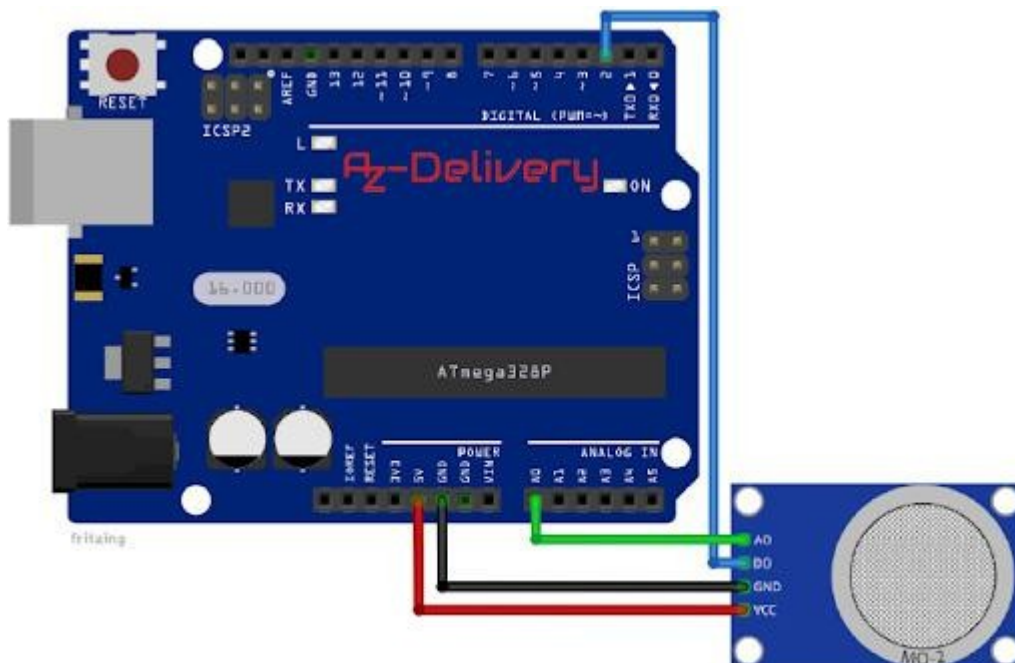
En el caso de la Raspberry Pi, primero hay que instalar el sistema operativo y, a continuación, configurar todo para que pueda utilizarse en el *modo Headless*. El modo *Headless* permite la conexión remota a la Raspberry Pi, sin necesidad de un monitor de pantalla de *PC*, ratón o teclado. Las únicas cosas que se utilizan en este modo son la propia Raspberry Pi, fuente de alimentación y conexión a Internet. Todo esto se explica minuciosamente en el eBook gratuito:

[Guía de inicio rápido de Raspberry Pi](#)

El sistema operativo *Raspbian* viene con *Python* preinstalado.

## Conexión del módulo con ATmega328p

Conecte el módulo con el ATmega328p como se muestra en la siguiente imagen:



| Clavija del módulo | Pin ATmega 328p | Color del cable |
|--------------------|-----------------|-----------------|
| VCC                | 5V              | Cable rojo      |
| GND                | GND             | Cable negro     |
| D0                 | D2              | Cable azul      |
| A0                 | A0              | Cable verde     |



## Sketch ejemplo

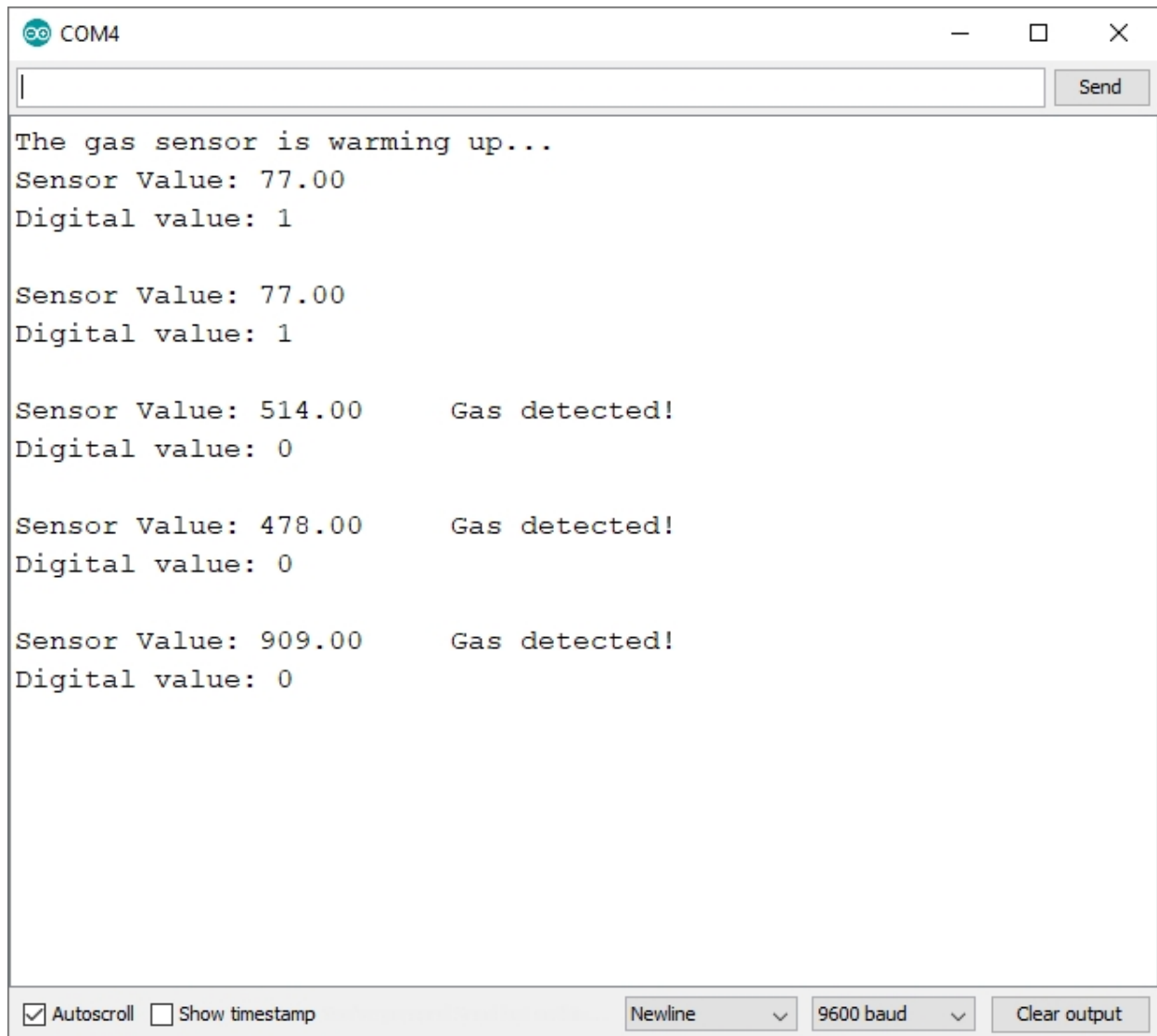
```
#define DIGITAL_PIN 2
#define ANALOG_PIN 0
uint16_t gasVal;
boolean isgas = false;
String gas;
void setup() {
  Serial.begin(9600);
  Serial.println("El sensor se está calentando...");
  delay(30000);
  pinMode(DIGITAL_PIN, INPUT);
}
void loop() {

  gasVal =
  analogRead(ANALOG_PIN); isgas =
  digitalRead(DIGITAL_PIN);

  if (isgas) {
    gas = "No";
  }
  si no {
    gas = "Sí";
  }
  gasVal = map(gasVal, 0, 1023, 0, 100);
  Serial.print("Gas detectado: ");
  Serial.println(gas);
  Serial.print("Porcentaje de gas: ");
  Serial.print(gasVal);
  Serial.print("%\n");
  retraso(2000);
}
```

# Az-Delivery

Sube el sketch al ATmega328p y abre Serial Monitor (*Herramientas > Serial Monitor*). El resultado debería ser como en la siguiente imagen:



The screenshot shows the Serial Monitor window for COM4. The window has a title bar with standard OS controls and a 'Send' button. The main area displays the following text:

```
The gas sensor is warming up...  
Sensor Value: 77.00  
Digital value: 1  
  
Sensor Value: 77.00  
Digital value: 1  
  
Sensor Value: 514.00      Gas detected!  
Digital value: 0  
  
Sensor Value: 478.00      Gas detected!  
Digital value: 0  
  
Sensor Value: 909.00      Gas detected!  
Digital value: 0
```

At the bottom, there are checkboxes for 'Autoscroll' (checked) and 'Show timestamp' (unchecked). To the right are dropdown menus for 'Newline' and '9600 baud', and a 'Clear output' button.



El sketch comienza con la definición y creación de dos macros llamadas *CLAVIJA\_DIGITAL*, *CLAVIJA\_ANALÓGICA*.

El *DIGITAL\_PIN* representa el pin digital del ATmega328p que se utiliza para conectar el pin de salida digital del sensor.

El *ANALOG\_PIN* representa el pin de entrada analógica del ATmega328p que se utiliza para conectar el pin de salida analógica del sensor.

Los datos del módulo se pueden leer de dos formas. Una es leyendo el pin de salida analógica del módulo, y la otra es leyendo el pin de salida digital del módulo. Para leer el pin de salida analógica del módulo, la variable llamada *gasVal* se utiliza para almacenar el valor de retorno de la función *analogRead()*. El valor de retorno es un número entero en el rango de 0 a 1023. Para convertirlo en un porcentaje, se utiliza la función *map()*. Esta es una función incorporada del IDE de Arduino. Tiene cinco argumentos y devuelve un valor entero.



# Az-Delivery

Por ejemplo:

```
gasVal = map(entrada, in_min, in_max, out_min, out_max)
```

El primer argumento es el valor de *entrada*, que está en el rango de *in\_min* a *in\_max*. El valor de retorno es un número entero en el rango de *out\_min* a *out\_max*. Esta función asigna un número del rango de entrada a otro número que se encuentra en un rango diferente.

Para leer el pin de salida digital del módulo, se utiliza la variable *isGas* para almacenar el valor de retorno de la función *digitalRead()*.

Al final de la función *loop()*, los datos se muestran en el monitor serie. Entre dos mediciones hay 2 segundos de pausa: `delay(2000);`



## Conexión de Nano V3.0 como ADC para Raspberry Pi

Debido a que la Raspberry Pi no tiene convertidor analógico-digital (ADC), la tarea es hacer que la Raspberry Pi sea capaz de leer voltajes analógicos. Para ello se puede utilizar ATmega328p o Nano V3.0. Para ello, Nano V3.0 tiene que estar conectado al sistema operativo Raspbian. Nano V3.0 puede leer voltajes analógicos, y puede utilizar la interfaz serie a través del puerto USB para enviar datos a la Raspberry Pi.

En primer lugar, el IDE de Arduino tiene que ser instalado en el Raspbian. En segundo lugar, el firmware de la placa microcontroladora tiene que ser cargado en Nano V3.0 y la biblioteca Python tiene que ser descargada.

Para ello, inicie el Raspbian, abra el terminal y ejecute el siguiente comando para actualizar el Raspbian:

**`sudo apt-get update && sudo apt-get upgrade -y`**

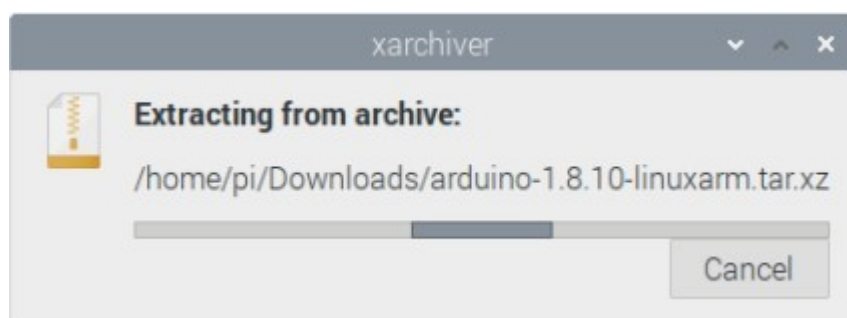
# Az-Delivery

Para descargar e instalar el IDE de Arduino, vaya al [sitio de Arduino](#): y descargue el archivo *tar.xz* del IDE de Arduino para **Linux ARM 32 bits** como se muestra en la siguiente imagen:

## Download the Arduino IDE



A continuación, extraiga el *archivo tar.xz*. Abra el explorador de archivos en el directorio donde se descargó el archivo *tar.xz*, haga clic con el botón derecho del ratón sobre él y ejecute la opción Extraer aquí. Espere unos minutos a que finalice el proceso de extracción.

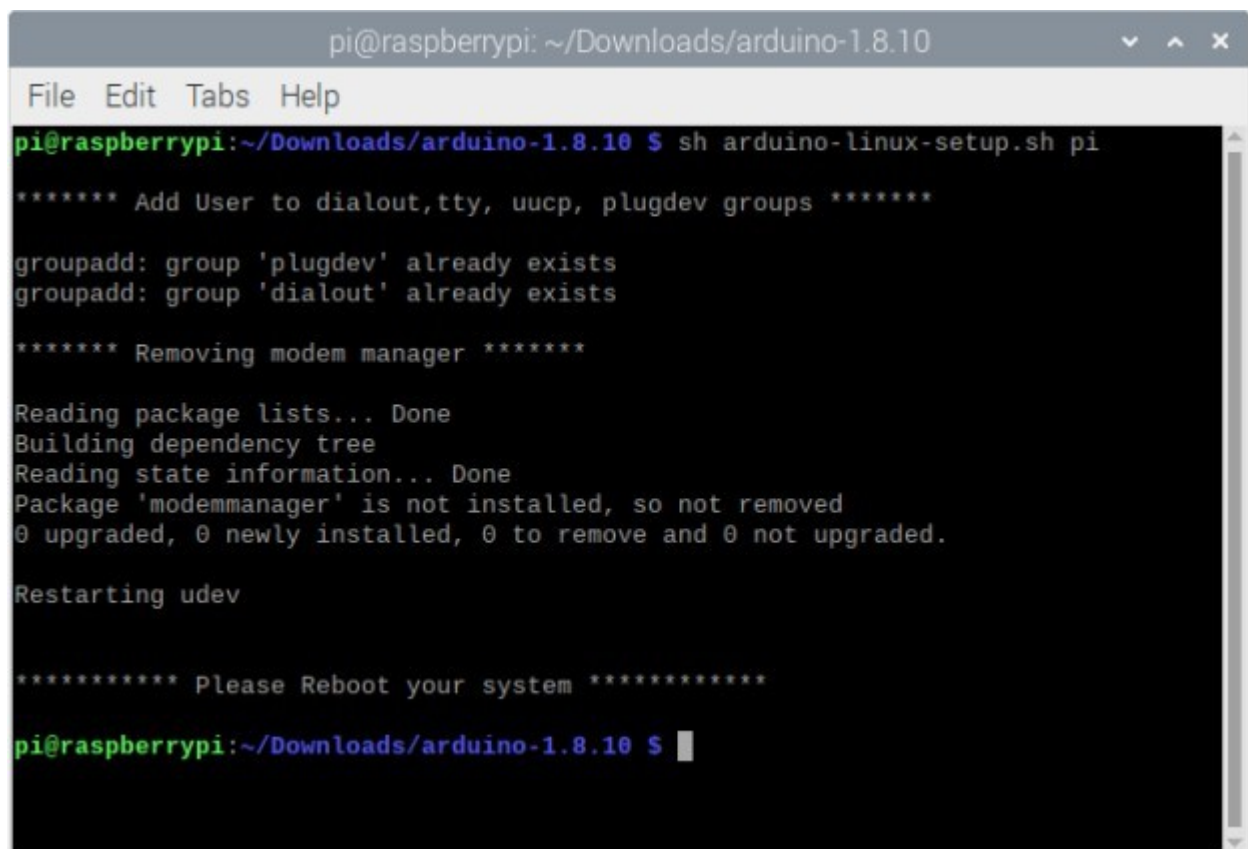


# Az-Delivery

Abra el terminal en el directorio donde se han extraído los archivos de instalación y ejecute el siguiente comando:

**sh arduino-linux-setup.sh pi**

donde *pi* es el nombre del superusuario en Raspbian.



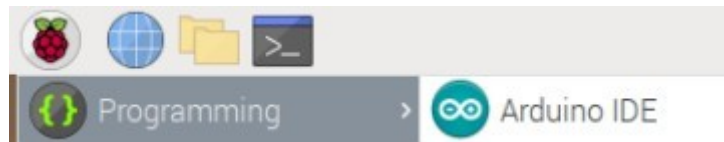
```
pi@raspberrypi: ~/Downloads/arduino-1.8.10
File Edit Tabs Help
pi@raspberrypi:~/Downloads/arduino-1.8.10 $ sh arduino-linux-setup.sh pi
***** Add User to dialout, tty, uucp, plugdev groups *****
groupadd: group 'plugdev' already exists
groupadd: group 'dialout' already exists
***** Removing modem manager *****
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package 'modemmanager' is not installed, so not removed
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Restarting udev
***** Please Reboot your system *****
pi@raspberrypi:~/Downloads/arduino-1.8.10 $
```

Después de esto, para instalar el IDE Arduino, ejecute el siguiente comando: **sudo sh install.sh**

# Az-Delivery

El IDE Arduino ya está instalado. Para ejecutar Arduino IDE, abra la aplicación:

*Menú Aplicaciones > Programación > Arduino IDE*



Antes de los siguientes pasos, primero hay que instalar las aplicaciones *pip3* y *git*; Abre el terminal y ejecuta el siguiente comando.

**sudo apt install python3-pip git -y**

La biblioteca para Python se llama *nanpy*. Para instalarla, abra el terminal y ejecute el siguiente comando: **pip3 install nanpy**

```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ pip3 install nanpy
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting nanpy
  Downloading https://www.piwheels.org/simple/nanpy/nanpy-0.9.6-py3-none-any.whl
    (47kB)
    100% |#####| 51kB 209kB/s
Requirement already satisfied: pyserial in /usr/lib/python3/dist-packages (from
nanpy) (3.4)
Installing collected packages: nanpy
Successfully installed nanpy-0.9.6
pi@raspberrypi:~/Scripts $
```

# Az-Delivery

Tras la instalación de la biblioteca *nanpy*, descarga un firmware ejecutando el siguiente comando:

```
git clone https://github.com/nanpy/nanpy-firmware.git
```

Cambia el directorio a *nanpy-firmware* ejecutando el siguiente comando:

```
cd nanpy-firmware
```

Y ejecuta el siguiente comando:

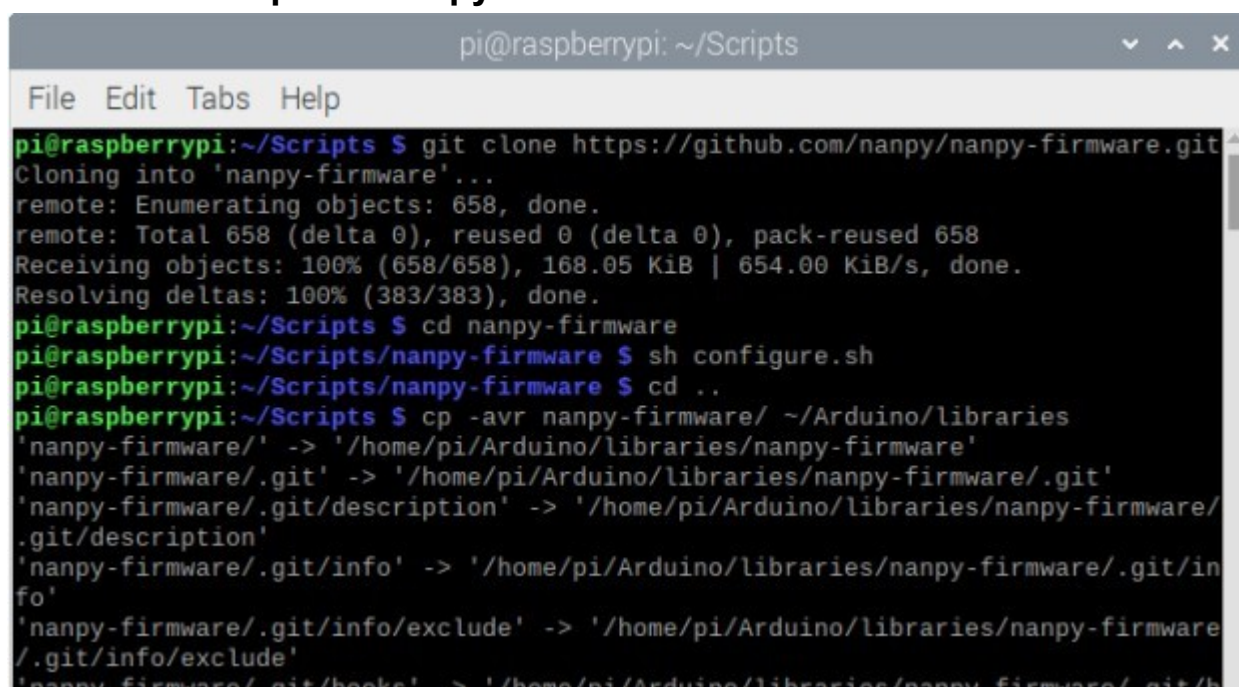
```
sh configure.sh
```

A continuación, copie el directorio *nanpy-firmware* en:

Directorio *Arduino/libraries*.

Para ello, ejecute el siguiente comando:

```
cp -avr nanpy-firmware/ ~/Arduino/libraries
```



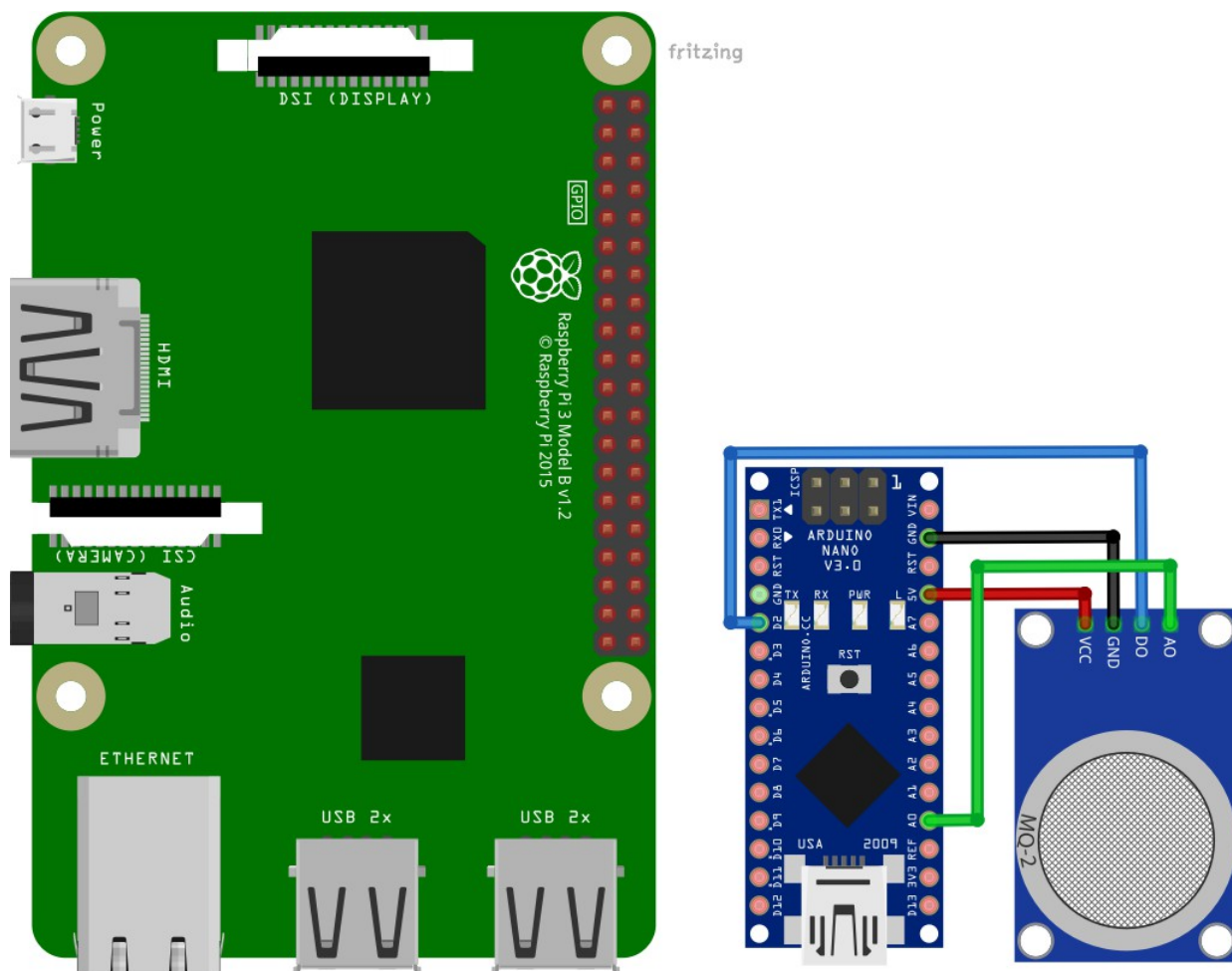
```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ git clone https://github.com/nanpy/nanpy-firmware.git
Cloning into 'nanpy-firmware'...
remote: Enumerating objects: 658, done.
remote: Total 658 (delta 0), reused 0 (delta 0), pack-reused 658
Receiving objects: 100% (658/658), 168.05 KiB | 654.00 KiB/s, done.
Resolving deltas: 100% (383/383), done.
pi@raspberrypi:~/Scripts $ cd nanpy-firmware
pi@raspberrypi:~/Scripts/nanpy-firmware $ sh configure.sh
pi@raspberrypi:~/Scripts/nanpy-firmware $ cd ..
pi@raspberrypi:~/Scripts $ cp -avr nanpy-firmware/ ~/Arduino/libraries
'nanpy-firmware/' -> '/home/pi/Arduino/libraries/nanpy-firmware'
'nanpy-firmware/.git' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git'
'nanpy-firmware/.git/description' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git/description'
'nanpy-firmware/.git/info' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git/info'
'nanpy-firmware/.git/info/exclude' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git/info/exclude'
'nanpy-firmware/.git/hooks' -> '/home/pi/Arduino/libraries/nanpy-firmware/.git/hooks'
```

El *nanpy-firmware* ya está instalado y listo para ser utilizado.



## Conexión del módulo con Raspberry Pi

Conecta el módulo con el Nano V3.0 como se muestra en la siguiente imagen:



| Clavija del módulo | Clavija Nano V3.0 | Color del cable |
|--------------------|-------------------|-----------------|
| VCC                | 5V                | Cable rojo      |
| GND                | GND               | Cable negro     |
| D0                 | D2                | Cable azul      |
| A0                 | A0                | Cable verde     |



A continuación, conecta el Nano V3.0 a través del cable USB a la Raspberry Pi y abre el IDE Arduino en el sistema operativo Raspbian. Comprueba si Arduino IDE puede detectar el puerto USB en el que está conectado el Nano V3.0: *Herramientas > Puerto > dev/ttyUSB0*

A continuación, vaya a: *Herramientas > Tablero > {nombre del tablero}* y selecciona la placa *Nano V3.0*.

Después de eso, para abrir un boceto para el *nanpy-firmware*, vaya a: *Archivo > Ejemplos > nanpy-firmware > Nanpy*

Sube el sketch al Nano V3.0. Para probar si todo funciona correctamente, hay que crear el sencillo script *Blink*, donde se utiliza el LED integrado de la Nano V3.0 para parpadear.

Crea el script *Blink.py*, y ábrelo en el editor de texto por defecto.





En el script *Blink.py* escribe las siguientes líneas de código:

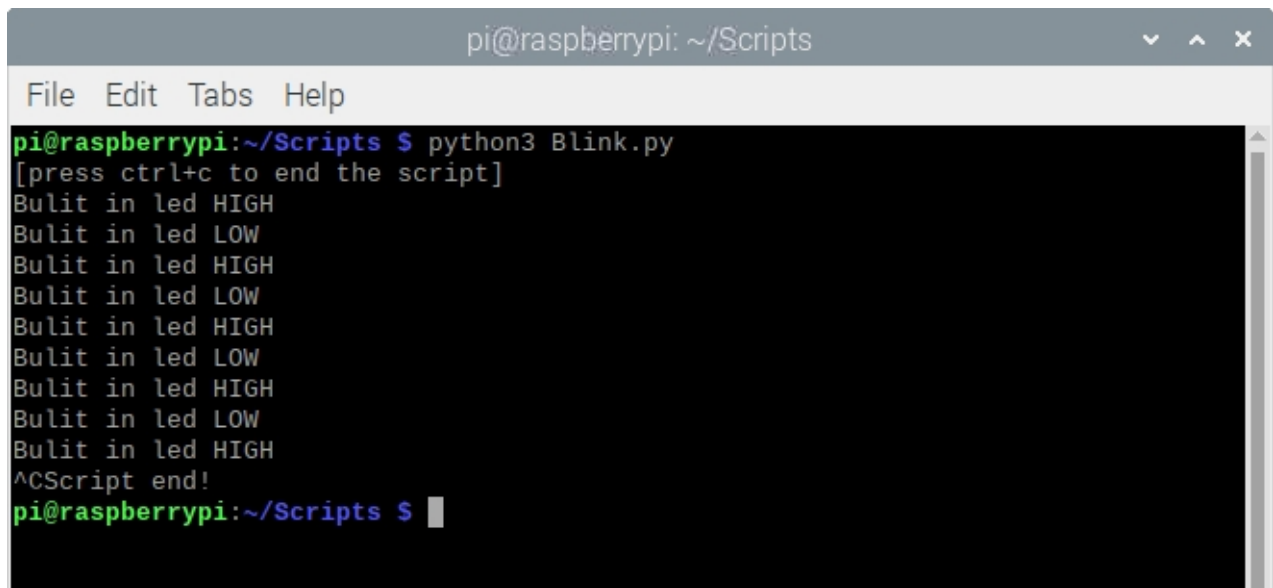
```
from nanpy import (ArduinoApi, SerialManager) from time
import sleep
ledPin = 13
try:
    connection1 = SerialManager()
    a = ArduinoApi(connection=connection1) except: print('Error
    al conectar con el Arduino') print('[Pulsa CTRL
    + C para finalizar el script]] a.pinMode(ledPin, a.OUTPUT) #
    Configurar Arduino intentar:
    while True:
        a.digitalWrite(ledPin, a.HIGH)
        print('Bulit en led HIGH')
        sleep(1)
        a.digitalWrite(ledPin, a.LOW)
        print('Bulit en led LOW')
        sleep(1)

except KeyboardInterrupt:
    print("\n;Fin del script!")
    a.digitalWrite(ledPin, a.LOW)
```

# Az-Delivery

Guarde el script con el nombre `Blink.py`. Para ejecutar el script, abra el terminal en el directorio donde está guardado el script y ejecute el siguiente comando: **python3 Blink.py**

El resultado debería ser como el de la siguiente imagen:



```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ python3 Blink.py
[press ctrl+c to end the script]
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
Bulit in led LOW
Bulit in led HIGH
^CScript end!
pi@raspberrypi:~/Scripts $
```

Para detener el script pulse 'CTRL + C' en el teclado.

El LED conectado al pin digital 13 del Nano V3.0 debería empezar a parpadear cada segundo.



El script comienza con la importación de dos bibliotecas, las funciones de la biblioteca *nanpy*, y el *tiempo*.

Luego, la variable llamada *ledPin* es creada e inicializada con el número 13. El número 13 representa el número del pin digital en el que está conectado el LED (LED integrado del Nano V3.0).

A continuación, se utiliza el bloque de código *try-except* para intentar conectar con el Nano V3.0. Si la conexión no tiene éxito, mensaje:

*No se ha podido conectar a la placa del microcontrolador* se muestra en el terminal.

Si la conexión tiene éxito, se crea e inicializa un objeto de comunicación llamado "a". El objeto "a" representa la placa Nano V3.0. Cualquier función utilizada en el IDE de Arduino se puede utilizar con el objeto "a", como se puede ver en el código.

Con la siguiente línea de código, se configura el modo pin para el pin digital 13: `a.pinMode(ledPin, a.OUTPUT)`

A continuación, en el bloque de bucle indefinido (*while True:*) se utiliza la función `digitalWrite()` para establecer el estado del pin digital 13 (estado HIGH o LOW). Con la función *digitalWrite()* se puede encender o apagar el LED conectado al pin 13.



En el bloque de bucle indefinido, el LED se *enciende* primero durante un segundo y luego se *apaga* durante un segundo. A esto se le llama *parpadear el LED*. El intervalo de tiempo de un solo parpadeo se puede cambiar en la siguiente línea de código: *sleep(1)*

Donde el número *1* representa el número de segundos de duración del intervalo de tiempo.

Para finalizar el bucle indefinido, pulsa 'CTRL + C' en el teclado. Esto se llama la interrupción del teclado, que se establece en el bloque *except* (*except KeyboardInterrupt*). En el bloque *except* se apaga el LED de la placa.



## Script Python para el módulo MQ-135

```
from nanpy import (ArduinoApi, SerialManager)
import tiempo
```

Inténtalo:

```
connection_1 = SerialManager()
a = ArduinoApi(conexión=conexión_1) excepto:
print('Error al conectar con el Arduino')
DIGITAL_PIN = 2
ANALOG_PIN = 0
time.sleep(2)
print('El sensor se está calentando...')
print('[Pulsa CTRL+C para finalizar el
script]') time.sleep(5) # El sensor se está
calentando...
```

Inténtalo:

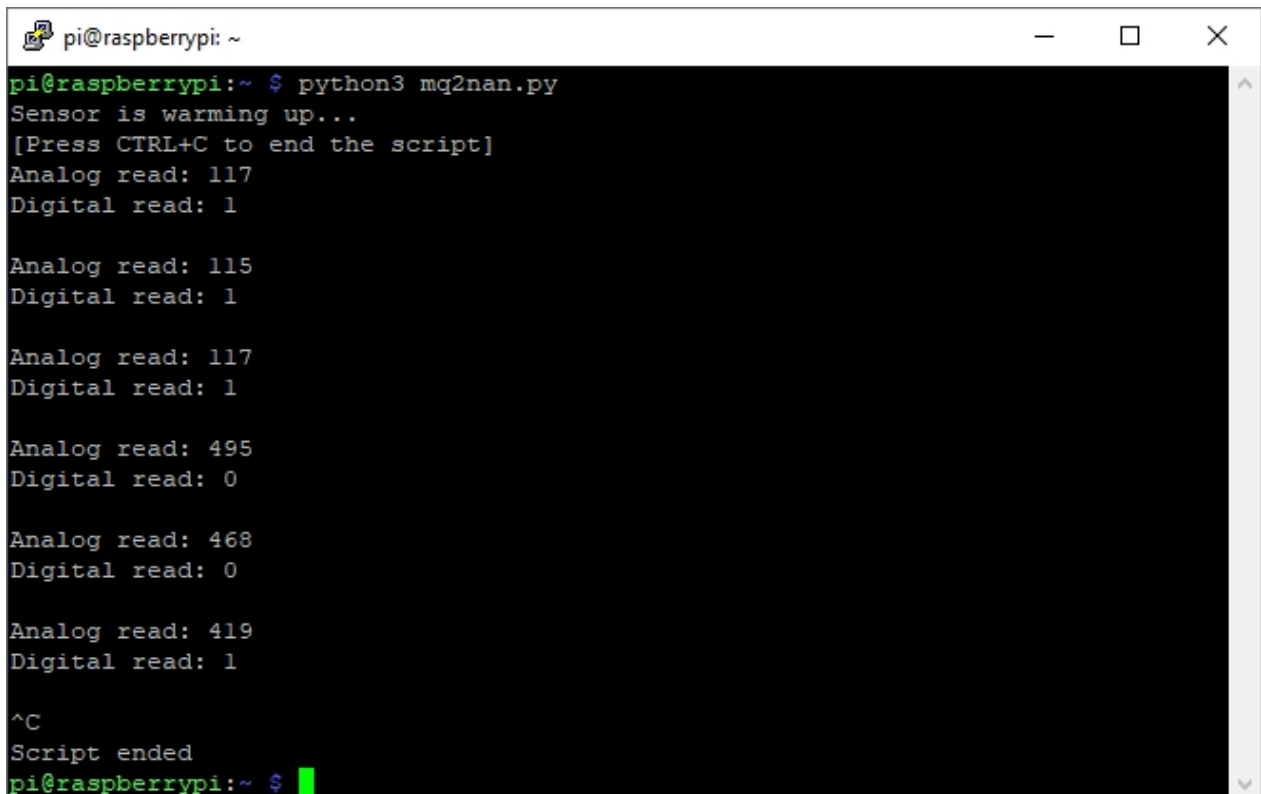
mientras sea verdad:

```
analogReading = a.analogRead(ANALOG_PIN)
digitalReading = a.digitalRead(DIGITAL_PIN)
print("Lectura analógica: {}\nLectura digital: {}".format(lecturaanalógica, lecturadigital))
time.sleep(2)
except KeyboardInterrupt:
print("\n¡Fin del script!")
```

# Az-Delivery

Guarde el script con el nombre `mq2nan.py`. Para ejecutar el script, abra el terminal en el directorio donde está guardado el script y ejecute el siguiente comando: **`python3 mq2nan.py`**

El resultado debería ser como el de la siguiente imagen:

A screenshot of a terminal window titled 'pi@raspberrypi: ~'. The terminal shows the execution of the command 'python3 mq2nan.py'. The output of the script is as follows:

```
pi@raspberrypi:~ $ python3 mq2nan.py
Sensor is warming up...
[Press CTRL+C to end the script]
Analog read: 117
Digital read: 1

Analog read: 115
Digital read: 1

Analog read: 117
Digital read: 1

Analog read: 495
Digital read: 0

Analog read: 468
Digital read: 0

Analog read: 419
Digital read: 1

^C
Script ended
pi@raspberrypi:~ $
```

The terminal window has a standard Linux window title bar with minimize, maximize, and close buttons. The output shows several readings of analog and digital values, followed by a Ctrl+C signal and the message 'Script ended'. The prompt 'pi@raspberrypi:~ \$' is visible at the bottom.

Para detener el script pulse 'CTRL + C' en el teclado.



Ahora es el momento de aprender y hacer tus propios proyectos. Puedes hacerlo con la ayuda de muchos scripts de ejemplo y otros tutoriales, que puedes encontrar en Internet.

**Si busca microelectrónica y accesorios de alta calidad, AZ-Delivery Vertriebs GmbH es la empresa adecuada. Dispondrá de numerosos ejemplos de aplicación, guías de instalación completas, libros electrónicos, bibliotecas y asistencia de nuestros expertos técnicos.**

<https://az-delivery.de>

**¡Diviértete!**

**Impresionante**

<https://az-delivery.de/pages/about-us>