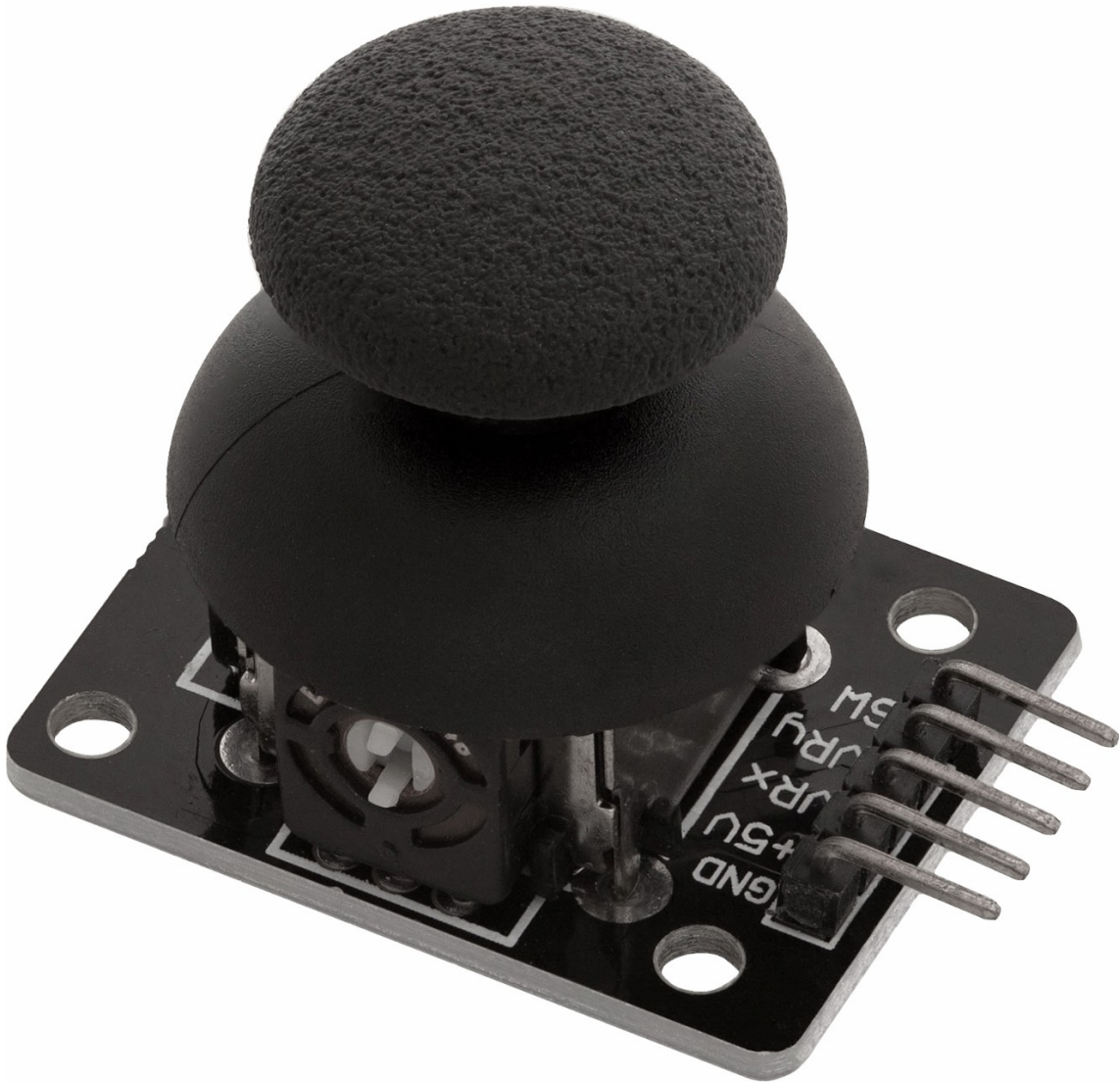


AZ-Delivery

Bienvenido.

Gracias por adquirir nuestro *Módulo Joystic AZ-Delivery KY-023*. En las páginas siguientes, se le explicará cómo utilizar y configurar este práctico dispositivo.

¡Que te diviertas!



Áreas de aplicación

Los productos están destinados al soporte y montaje de componentes y circuitos electrónicos.

Conocimientos y habilidades requeridos.

El uso de estos productos requiere conocimientos básicos de ingeniería eléctrica y manejo de componentes electrónicos. Los usuarios deben poder instalar los productos correctamente y tomar las precauciones de seguridad necesarias.

Condiciones ambientales

Los productos deben utilizarse en un ambiente libre de humedad, polvo y luz solar directa. No deben operarse cerca de fuentes de calor o en ambientes químicamente agresivos para evitar daños y riesgos de seguridad.

Uso previsto

Los productos eléctricos pasivos, como disipadores de calor, soportes de baterías y clips o placas de conexión, deben operarse en entornos que cumplan con los rangos de temperatura y voltaje especificados de los productos respectivos. Estos componentes suelen estar diseñados para uso en interiores.

Uso inadecuado previsible

Los usos inadecuados pero previsibles incluyen el uso en ambientes húmedos o extremadamente calientes o el funcionamiento por parte de personas no capacitadas o discapacitadas. El producto debe mantenerse alejado de niños y mascotas.

desecho

¡No lo deseche 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. convertirse en. 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

instrucciones de seguridad

Atención: La eliminación inadecuada de componentes electrónicos puede poner en peligro el medio ambiente y la salud. Nota: Deseche los componentes electrónicos de acuerdo con las regulaciones locales y utilice opciones de reciclaje adecuadas. Atención: Los medios químicamente agresivos pueden dañar los materiales de los productos. Nota: No utilice los productos en ambientes corrosivos o químicamente agresivos. Atención: La eliminación inadecuada de componentes electrónicos puede poner en peligro el medio ambiente y la salud. Nota: Deseche los componentes electrónicos de acuerdo con las regulaciones locales y utilice opciones de reciclaje adecuadas. Atención: Los medios químicamente agresivos pueden dañar los materiales de los productos. Nota: No utilice los productos en ambientes corrosivos o químicamente agresivos. Precaución: Los golpes mecánicos o la flexión pueden dañar los productos y los componentes conectados. Nota: Evite el estrés mecánico y proteja los productos de influencias físicas. Atención: Una fijación inadecuada puede provocar fallos de funcionamiento y daños. Nota: Asegúrese de que todos los productos estén ensamblados de forma segura y firme. Precaución: Los productos dañados pueden presentar riesgos para la seguridad. Nota: Revise los productos periódicamente para detectar daños visibles y reemplace las piezas defectuosas inmediatamente. Atención: La sobrecarga puede provocar sobrecalentamiento y fallos de los productos. Nota: Utilice los productos únicamente dentro de los límites de carga especificados. Atención: El sobrecalentamiento puede causar daños a los productos y a los componentes electrónicos conectados. Nota: Asegúrese de que, por ejemplo, los disipadores de calor o los componentes que se calientan estén adecuadamente ventilados y que no se excedan los rangos de temperatura especificados.

Az-Delivery

Índice

Introducción.....	3
Especificación	4
El pinout.....	4
Cómo configurar Arduino IDE	5
Cómo configurar la Raspberry Pi y Python.....	9
Conexión del módulo con Atmega328p	10
Ejemplo de croquis	11
Conexión del módulo con Raspberry Pi.....	13
Script Python para el módulo KY-023	24



Introducción

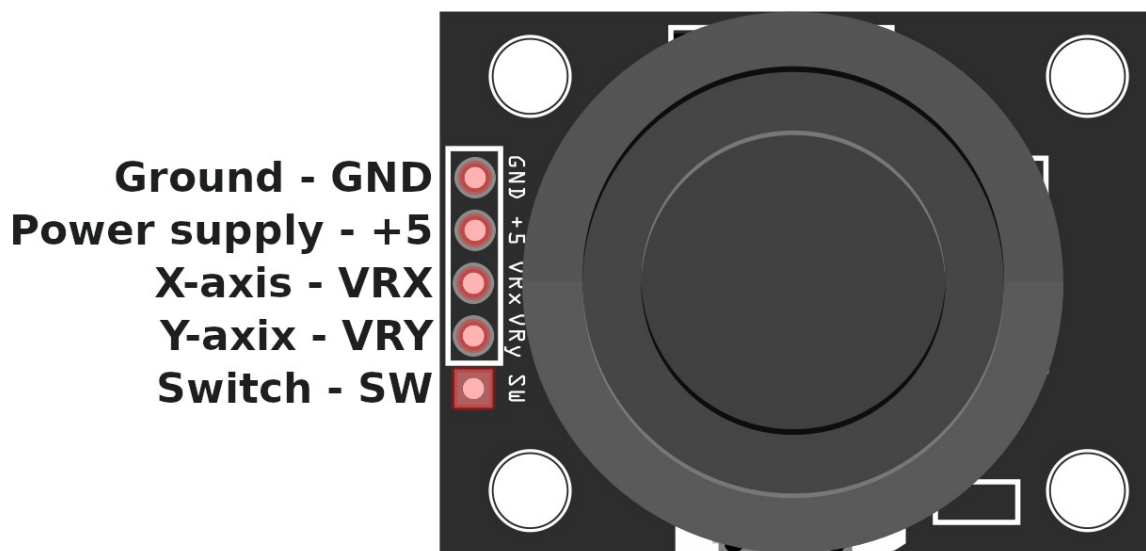
El módulo joystick es un joystick analógico fácil de usar para un microcontrolador. El joystick dispone de dos ejes (ejes *X* e *Y*), así como de un interruptor de botón que se activa cuando se pulsa el joystick. El joystick es básicamente una combinación de dos potenciómetros. Esto significa que cuando el joystick se mueve a lo largo del eje *X*, la resistencia del potenciómetro cambia y cuando se aplica tensión, el cambio de resistencia se traduce en un cambio de tensión. El voltaje se puede utilizar para detectar la posición *X* conectando el pin *VRX* del joystick a un pin de entrada analógica *A0* del Atmega328p. Lo mismo se aplica para el eje *Y*. La posición del eje *Y* puede leerse conectando *VRX* al pin *A1* de entrada analógica del Atmega328p. El estado de pulsación o el estado del pin *SW* se puede conectar al pin digital *D2* del Atmega328p. Debido a que el módulo consta de dos potenciómetros, se puede utilizar con cualquier tensión de alimentación, por ejemplo, puede trabajar tanto en +3,3V y +5V DC.

Especificación

"	Tensión de	de 3,3 V a 5 V CC
"	funcionamiento:	digital, con alta sensibilidad
	Salida:	
"	Dimensiones:	26 x 34 x 32 mm [1,02 x 1,34 x 1,26 pulg].

El pinout

El módulo joystic KY-023 tiene cinco pines. El diagrama de pines se muestra en la siguiente imagen:



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 links for different operating systems: **Windows** (Installer, for Windows XP and up; 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, 64 bits, ARM 32 bits, ARM 64 bits), **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*.

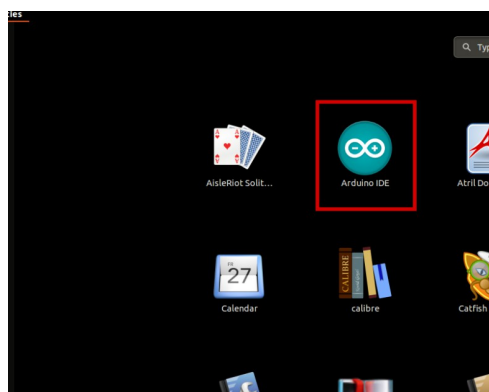
Para ejecutar el primer script en el terminal, abra el terminal en el directorio extraído y ejecute el siguiente comando:

```
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* script tiene que ser utilizado después de la instalación del primer script. Ejecute el siguiente comando en el terminal (directorio extraído): **sh install.sh**

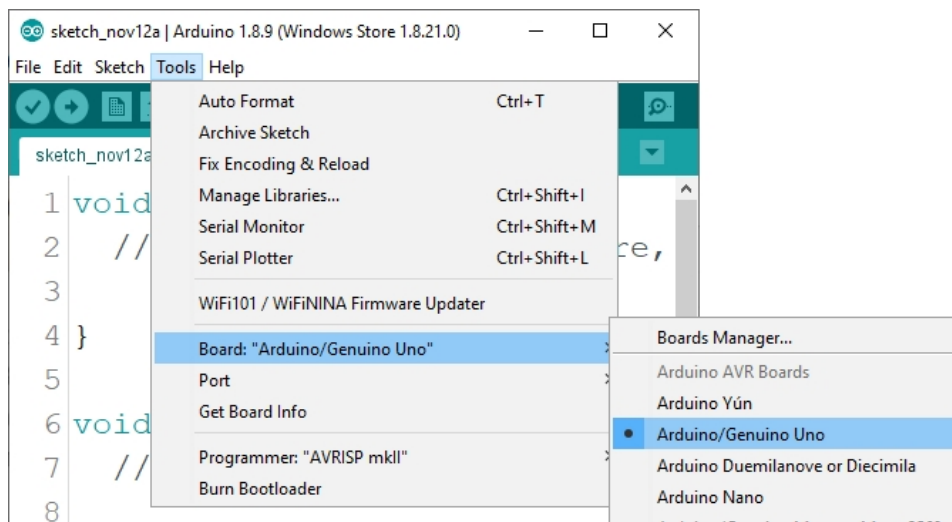
Después de la instalación de estas secuencias de comandos, vaya a *Todas las aplicaciones*, donde el *Arduino IDE* está instalado.



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 libro electrónico.

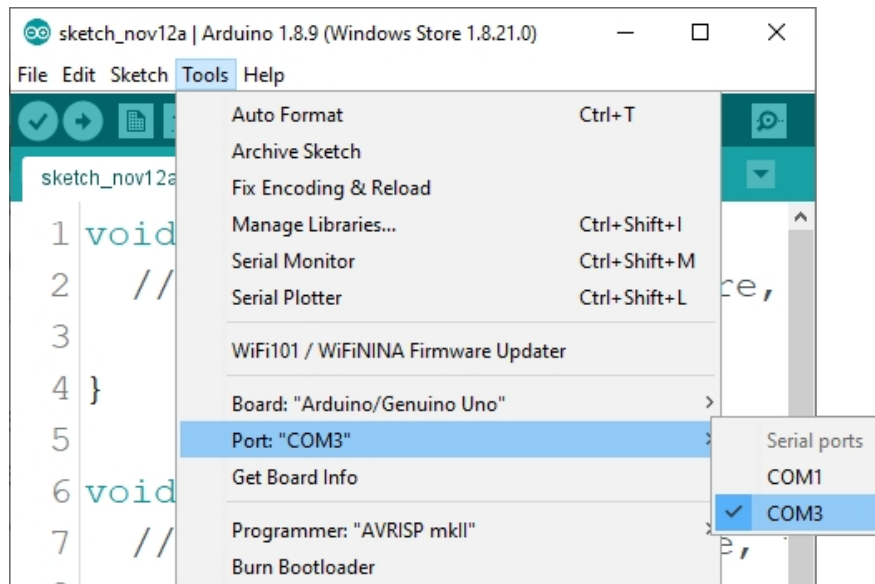
Lo siguiente es comprobar si tu PC puede detectar una placa Atmega328p. Abra recién instalado Arduino IDE, y vaya a:

Herramientas > Tablero > {nombre de su tablero aquí}
{el nombre de tu placa aquí} debería ser el *Arduino/Genuino Uno*, como se puede ver en la siguiente imagen:



Hay que seleccionar el puerto al que está conectada la placa Atmega328p. Ir a: *Herramientas > Puerto > {el nombre del puerto va aquí}* y cuando la placa Atmega328p 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

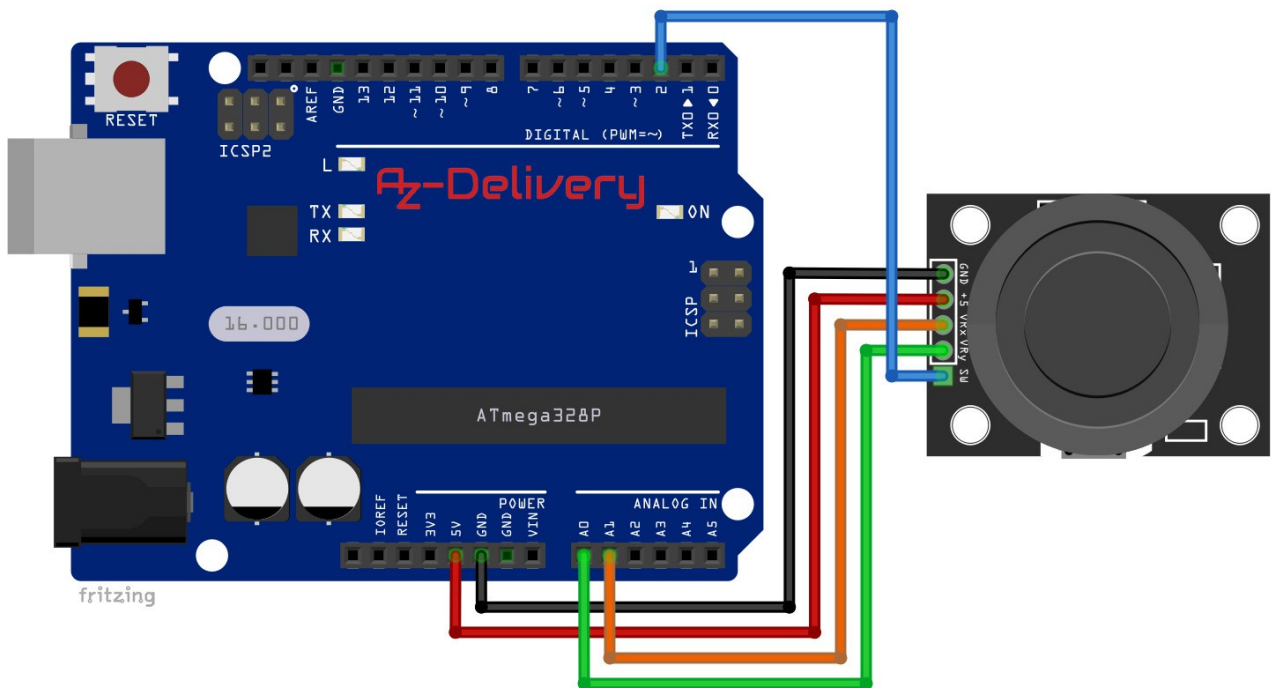
En el caso de la Raspberry Pi, primero hay que instalar el sistema operativo y, a continuación, configurar todo para que pueda utilizarse en el *modo Headless*. El modo *Headless* permite la conexión remota a la Raspberry Pi, sin necesidad de un monitor de pantalla de *PC*, ratón o teclado. Las únicas 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 KY-023 con el Atmega328p como se muestra en el siguiente diagrama de conexión:



Clavija KY-023

GND

+5V

VRX

VRY

SW

>

>

>

>

>

>Mc Pin

GND

5V

A1

A0

D2

Cable negro

Cable rojo

Cable
naranja

Cable verde

Cable azul

Az-Delivery

Ejemplo de croquis

```
#define VX_PIN 0
#define VY_PIN 1
#define BUTTON_PIN 2
uint8_t value = 0;

void setup() {
    pinMode(BUTTON_PIN, INPUT_PULLUP);
    Serial.begin(9600);
}

void loop() {
    value = analogRead(VX_PIN);
    Serial.print("X:");
    Serial.print(value, DEC);

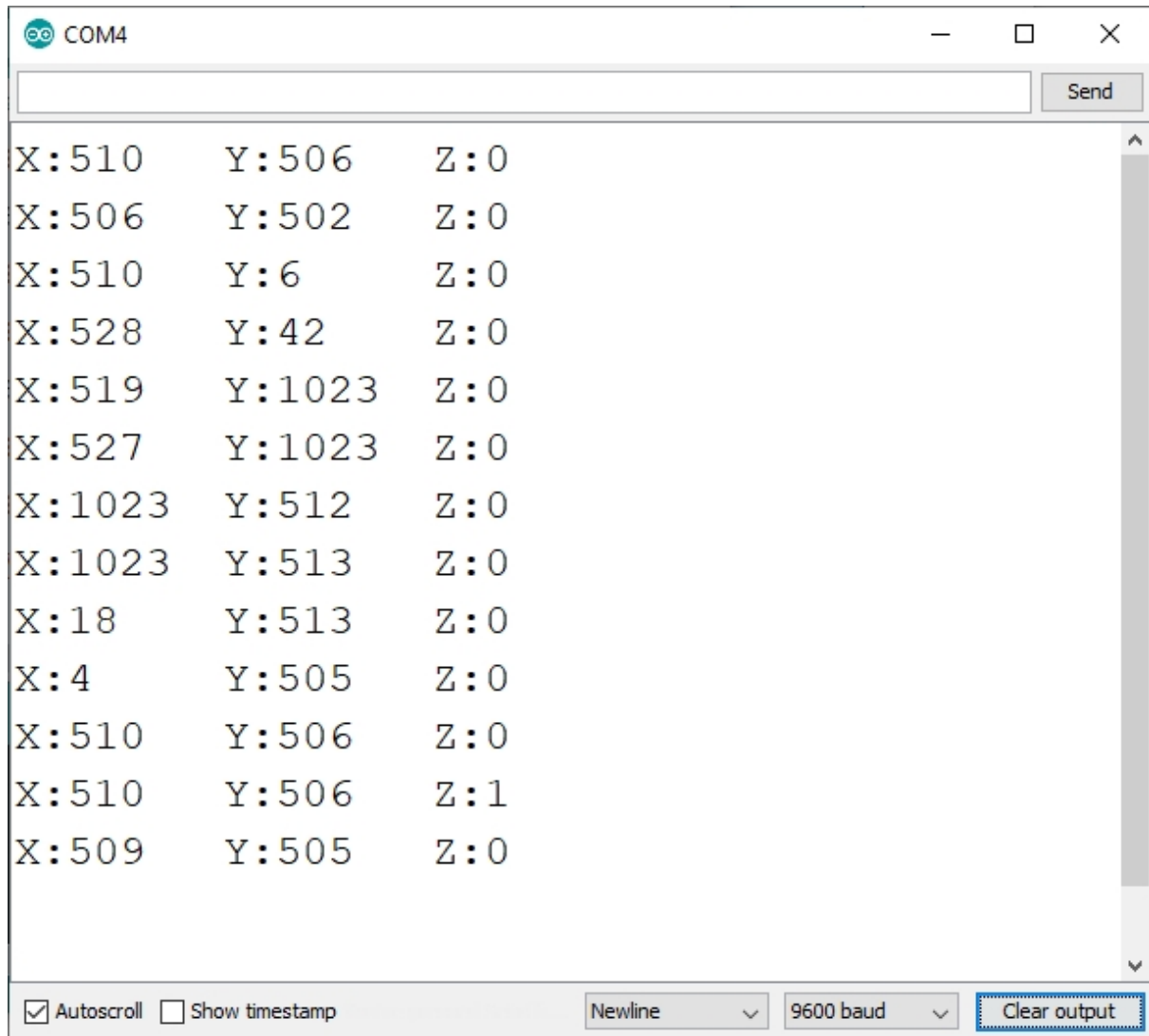
    value = analogRead(VY_PIN);
    Serial.print("\tY:");
    Serial.print(value, DEC);

    value = !digitalRead(BUTTON_PIN);
    Serial.print("\tZ:");
    Serial.println(value, DEC);

    delay(500);
}
```

Az-Delivery

Sube el sketch al Atmega328p y abre Serial Monitor (*Herramientas > Serial Monitor*). El resultado debería ser como el de la siguiente imagen:



Para obtener la salida de la imagen mueva o pulse el Joystick.



Conexión del módulo con Raspberry Pi

Debido a que la Raspberry Pi no tiene convertidor analógico-digital (ADC), pero con el fin de utilizar el módulo KY-023 con la Raspberry Pi, la Raspberry Pi tiene que ser capaz de leer voltajes analógicos. Para ello se puede utilizar el Atmega328p. Para ello, el Atmega328p se utiliza en el sistema operativo *Linux Raspbian*. El Atmega328p 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 descargado e instalado en el Raspbian. *En* segundo lugar, el firmware para Atmega328p tiene que ser descargado y cargado en el Atmega328p, y por último, la biblioteca de Python tiene que ser descargada e instalada.

Para ello, enciende la Raspberry Pi y conéctala a Internet. Inicie la aplicación *RealVNC* en el ordenador remoto y conecte la aplicación a la Raspberry Pi (como se explica en el eBook para la Raspberry Pi).

Lo primero que hay que hacer al arrancar la Raspberry Pi es actualizar Raspbian; abre el terminal y ejecuta el siguiente comando: **sudo apt-get update && sudo apt-get upgrade -y**

Y esperar a que el comando termine su trabajo.

Ahora, el sistema operativo Raspbian está actualizado.

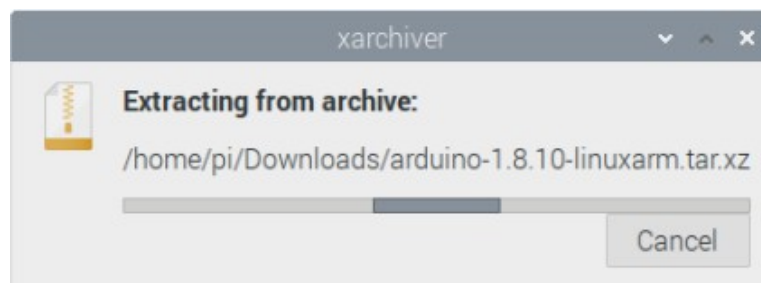
Az-Delivery

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

Download the Arduino IDE



A continuación, hay que extraer 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 se complete el proceso de extracción.

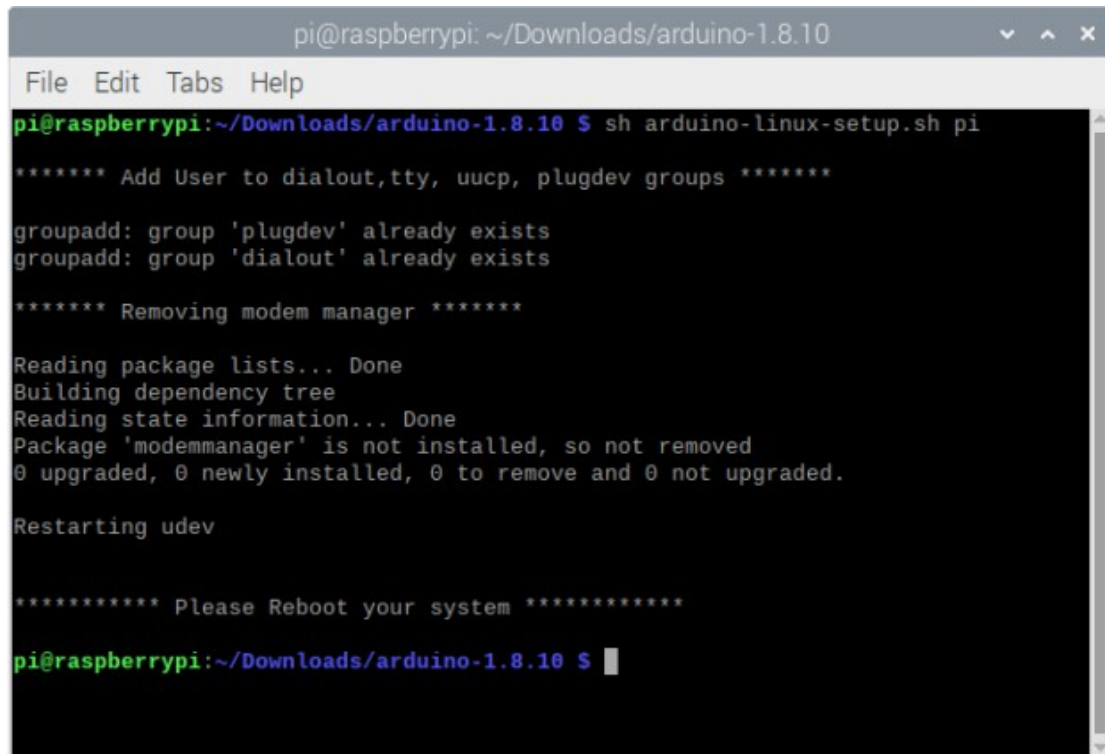


Az-Delivery

Abra el terminal en la carpeta extraída y ejecute el siguiente comando:

```
sh arduino-linux-setup.sh pi
```

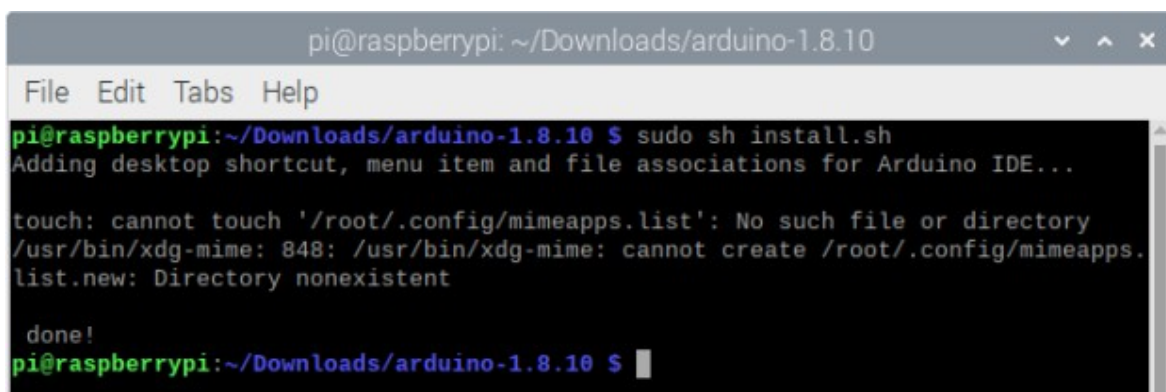
donde *pi* es el nombre del superusuario en Raspbian.



```
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 de Arduino, ejecute el siguiente comando:

```
sudo sh install.sh
```

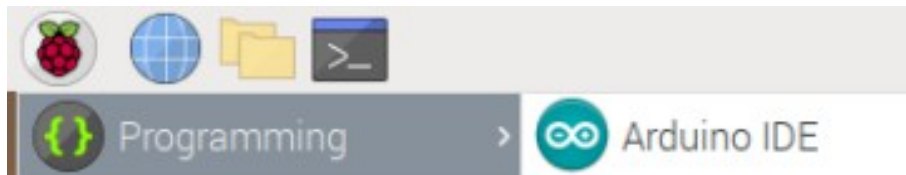


```
pi@raspberrypi: ~/Downloads/arduino-1.8.10
File Edit Tabs Help
pi@raspberrypi:~/Downloads/arduino-1.8.10 $ sudo sh install.sh
Adding desktop shortcut, menu item and file associations for Arduino IDE...
touch: cannot touch '/root/.config/mimeapps.list': No such file or directory
/usr/bin/xdg-mime: 848: /usr/bin/xdg-mime: cannot create /root/.config/mimeapps.
list.new: Directory nonexistent
done!
pi@raspberrypi:~/Downloads/arduino-1.8.10 $
```

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*; 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

Después de instalar la librería *nanpy*, descarga un firmware

A t m e g a 3 2 8 p 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
```

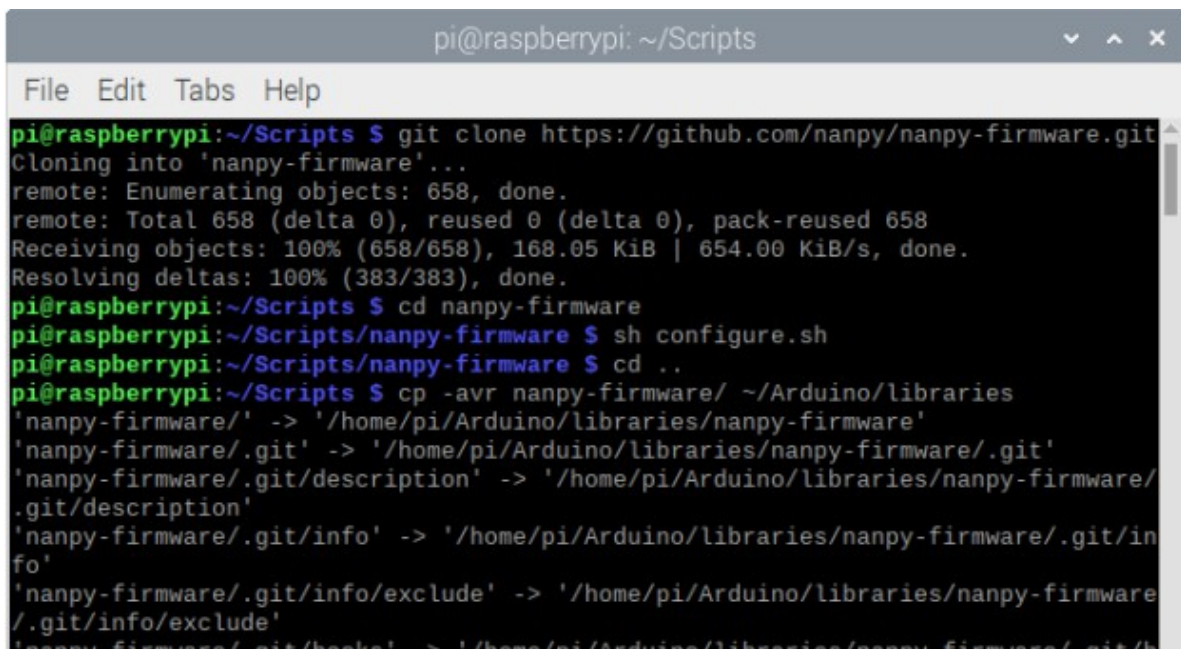
Y ejecuta el siguiente comando:

```
sh configure.sh
```

Siguiente, copie el directorio *nanpy-firmware* en *Arduino/bibliotecas*

directorio. 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.

Az-Delivery

Conecte el Atmega328p a través del cable USB a la Raspberry Pi y, a continuación, abra el IDE Arduino en el sistema operativo Raspbian. Compruebe si el IDE Arduino puede detectar el puerto USB al que está conectado el Atmega328p:

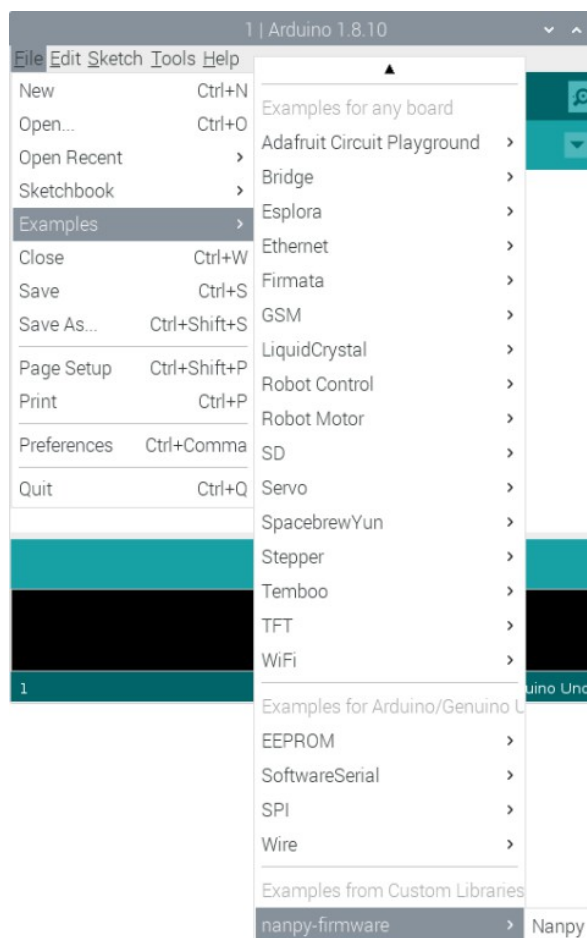
Herramientas > Puerto > dev/ttyUSB0

A continuación, vaya a: *Herramientas > Tablero > {nombre del tablero}*

y selecciona *tu* tabla.

A continuación, para abrir un boceto para el *nanpy-firmware*, vaya a:

Archivo > Ejemplos > nanpy-firmware > Nanpy

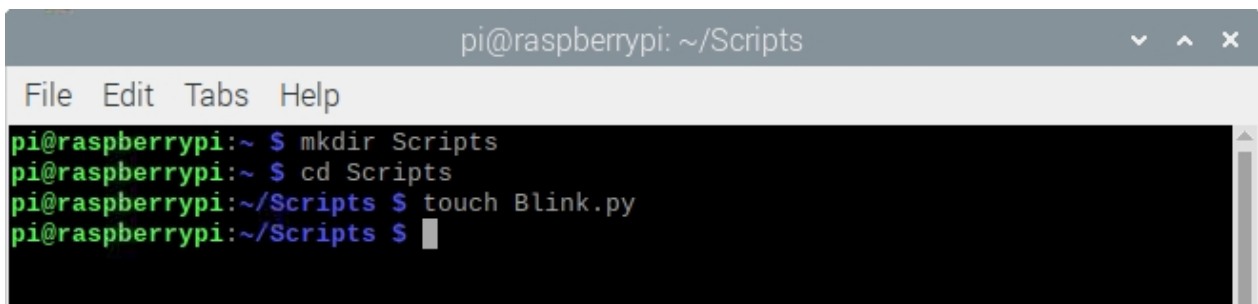


Az-Delivery

Sube el sketch al Atmega328p. Para probar si everything funciona correctamente, hay que crear el sencillo script *Blink*, donde parpadea el LED de a bordo del Atmega328p.

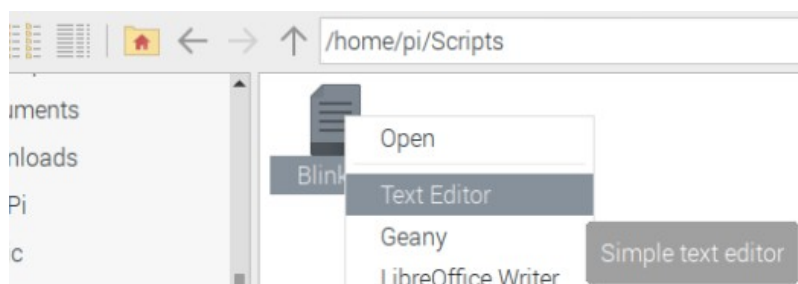
Abre el terminal, crea el directorio *Scripts* y el script *Blink.py*. Para ello, ejecute los siguientes comandos, uno por uno:

mkdir Scripts	- crear en el directorio <i>Scripts</i>
cd /Scripts	- cambie al directorio <i>Scripts</i>
touch Blink.py	- crear un nuevo archivo llamado <i>Blink.py</i>



```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~ $ mkdir Scripts
pi@raspberrypi:~ $ cd Scripts
pi@raspberrypi:~/Scripts $ touch Blink.py
pi@raspberrypi:~/Scripts $
```

Abra el *Explorador de archivos*, vaya al directorio *Scripts* y abra *Blink.py* script en el editor de texto por defecto:



Az-Delivery

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(conexión=conexión1)
excepto:
    print('Error al conectar con el Arduino')

print('[¡Presiona CTRL + C para finalizar el script!])'
a.pinMode(ledPin, a.OUTPUT) # Configurar Arduino
```

```
Inté
ntal mientras sea verdad:
o:
    a.digitalWrite(ledPin, a.HIGH)
    print('Bulit en led HIGH')
    sleep(1)
    a.digitalWrite(ledPin, a.LOW)
    print('Bulit en led LOW')
    sleep(1)
```

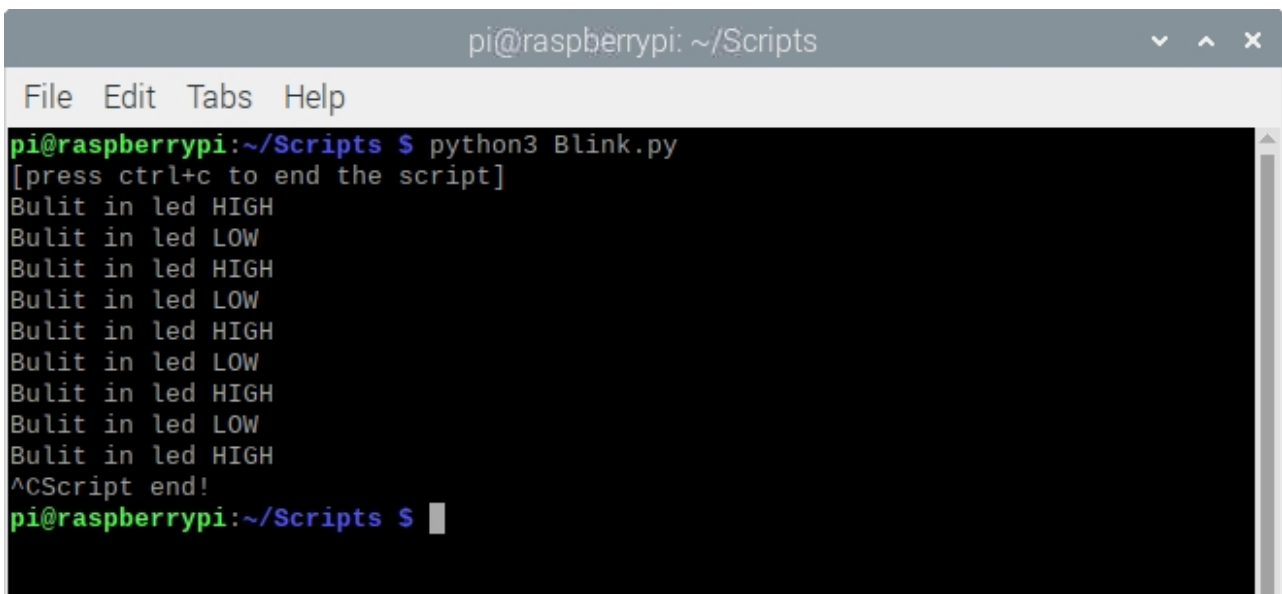
```
except KeyboardInterrupt:
    print('\n¡Fin del script!')
    a.digitalWrite(ledPin, a.LOW)
```

Az-Delivery

Guarde el script. 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 parecerse al de la imagen siguiente:



```
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 Atmega328p debería empezar a parpadear cada segundo.

Az-Delivery

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

A continuación, se crea la variable llamada *ledPin* y se inicializa con el número 13. El número 13 representa el número del pin digital en el que está conectado el LED (LED integrado del Atmega328p).

Después, el bloque de código *try-except* se utiliza para intentar conectar con el Atmega328p. Si la conexión no tiene éxito, el mensaje:

Error de conexión
aparece en el terminal.

Si la conexión tiene éxito, se crea e inicializa el objeto de comunicación llamado "a". El objeto "a" representa la placa Atmega328p. 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 del pin digital 13:

```
a.pinMode(ledPin, a.OUTPUT)
```

Luego, 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 .

Az-Delivery

En el bloque de bucle indefinido, el LED se *enciende* primero durante un segundo y luego se *apaga* durante un segundo. Esto se denomina *parpadeo del LED*.

El intervalo de tiempo de un parpadeo puede modificarse en la siguiente línea de código: `sleep(1)`

Donde el número `1` representa el número de segundos para la duración del intervalo de tiempo.

Para terminar el bucle infinito 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 expect* se *apaga* el LED de la placa.



Script Python para el módulo KY-023

Conecta el módulo KY-023 con el Atmega328p como se muestra en el diagrama de conexión del capítulo *Conectar el módulo con el Atmega328p*, y luego conecta el Atmega328p con la Raspberry Pi mediante un cable usb. A continuación, carga el firmware *nanpy* en el Atmega328p, y utiliza el siguiente código para controlar el módulo KY-023:

```
from nanpy import (ArduinoApi, SerialManager)
from time import sleep

Inténtalo:
    connection1 = SerialManager()
    a = ArduinoApi(conexión=conexión1)
excepto:
    print('Error al conectar con el Arduino')

# Configuración del pin para el
Arduino a.pinMode(2,
a.INPUT_PULLUP) valueX, valueY,
valueZ = 0, 0, 0
print('[¡Presiona CTRL + C para terminar el
script!]) try: # Bucle principal del programa
    mientras sea verdad:
        valueX = a.analogRead(0)
        valueY = a.analogRead(1)
        valueZ = not a.digitalRead(2)
        print('X:{}\tY:{}\tZ:{}'.format(valorX, valorY, valorZ))
        sleep(0.5)

except KeyboardInterrupt:
    print('\n¡Fin del
```

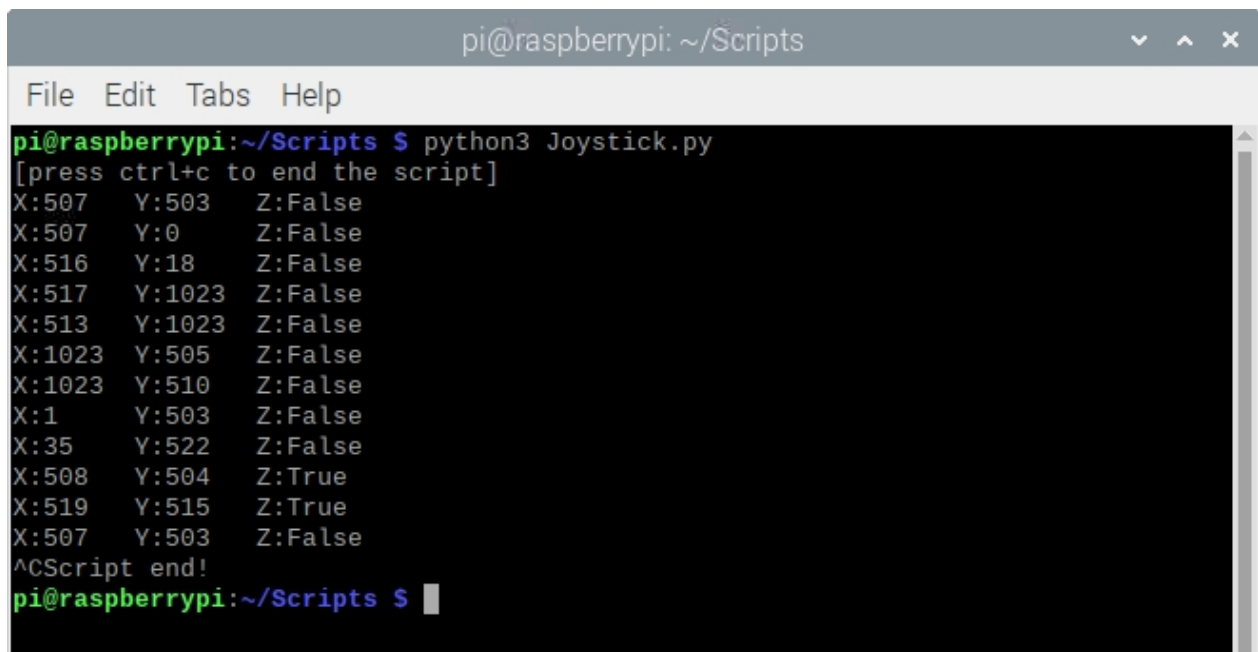
Az-Delivery

script!')

Az-Delivery

Guarde el script con el nombre *Joystick.py*. Para ejecutar el script abra el terminal en el directorio donde está guardado el script y ejecuta el siguiente comando: **python3 Joystick.py**

El resultado debería parecerse al de la imagen siguiente:



```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ python3 Joystick.py
[press ctrl+c to end the script]
X:507 Y:503 Z:False
X:507 Y:0 Z:False
X:516 Y:18 Z:False
X:517 Y:1023 Z:False
X:513 Y:1023 Z:False
X:1023 Y:505 Z:False
X:1023 Y:510 Z:False
X:1 Y:503 Z:False
X:35 Y:522 Z:False
X:508 Y:504 Z:True
X:519 Y:515 Z:True
X:507 Y:503 Z:False
^CScript end!
pi@raspberrypi:~/Scripts $
```

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>