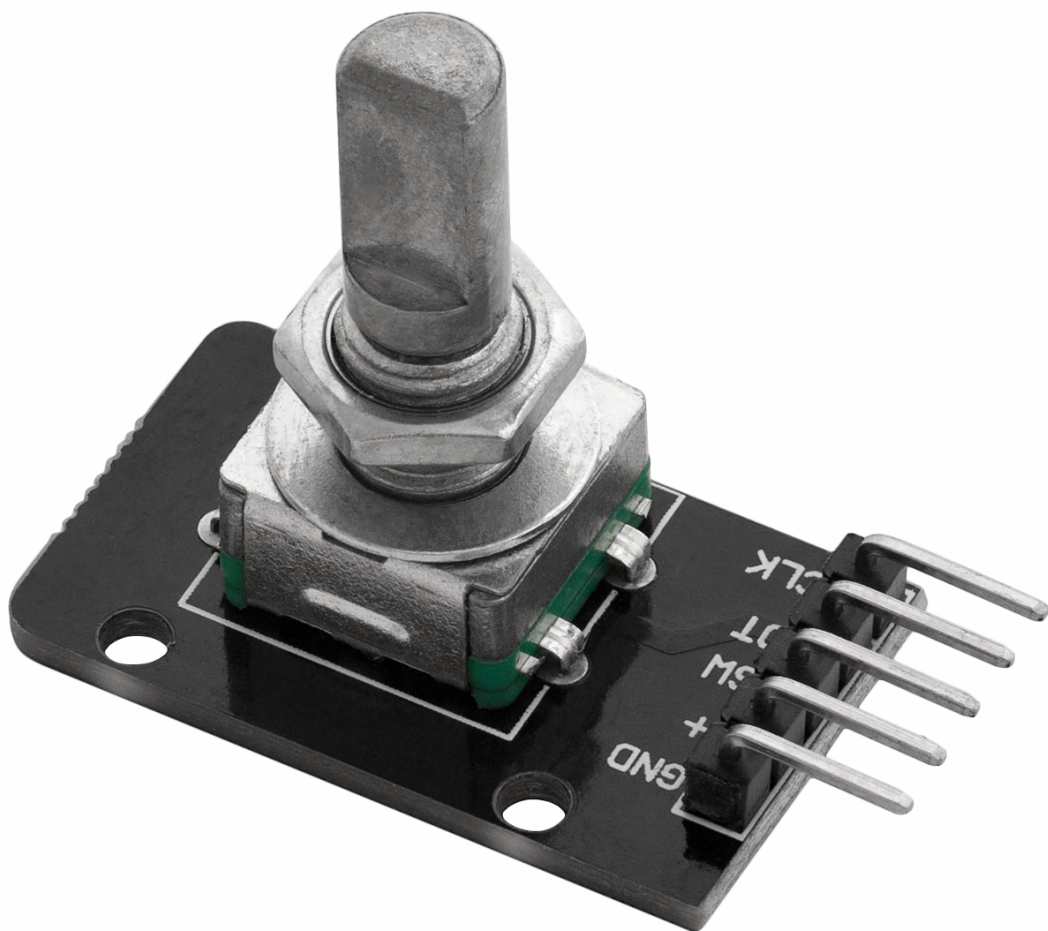


# AZ-Delivery

## Welcome!

Thank you for purchasing our *AZ-Delivery KY-040 Rotary Encoder Module*.  
On the following pages, you will be introduced to how to use and set up this handy device.

**Have fun!**



## Areas of application

Education and teaching: Use in schools, universities and training institutions to teach the basics of electronics, programming and embedded systems. Research and development: Use in research and development projects to create prototypes and experiments in the fields of electronics and computer science. Prototype development: Use in the development and testing of new electronic circuits and devices. Hobby and Maker Projects: Used by electronics enthusiasts and hobbyists to develop and implement DIY projects.

## Required knowledge and skills

Basic understanding of electronics and electrical engineering. Knowledge of programming, especially in the C/C++ programming language. Ability to read schematics and design simple circuits. Experience working with electronic components and soldering.

## Operating conditions

The product may only be operated with the voltages specified in the data sheet to avoid damage. A stabilized DC power source is required for operation. When connecting to other electronic components and circuits, the maximum current and voltage limits must be observed to avoid overloads and damage.

## Environmental conditions

The product should be used in a clean, dry environment to avoid damage caused by moisture or dust. Protect the product from direct sunlight (UV)

## Intended Use

The product is designed for use in educational, research and development environments. It is used to develop, program and prototype electronic projects and applications. The Sensor product is not intended as a finished consumer product, but rather as a tool for technically savvy users, including engineers, developers, researchers and students.

## Improper foreseeable use

The product is not suitable for industrial use or safety-relevant applications. Use of the product in medical devices or for aviation and space travel purposes is not permitted

## disposal

Do not discard with household waste! Your product is according to the European one Directive on waste electrical and electronic equipment to be disposed of in an environmentally friendly manner. The valuable raw materials contained therein can be recycled become. The application of this directive contributes to environmental and health protection. Use the collection point set up by your municipality to return and Recycling of old electrical and electronic devices. WEEE Reg. No.: DE 62624346

## electrostatic discharge

Attention: Electrostatic discharges can damage the product. Note: Ground yourself before touching the product, such as by wearing an anti-static wrist strap or touching a grounded metal surface.

## safety instructions

Although our product complies with the requirements of the RoHS Directive (2011/65/EU) and does not contain any hazardous substances in quantities above the permitted limits, residues may still be present. Observe the following safety instructions to avoid chemical hazards: Caution: Soldering can produce fumes that can be harmful to health. Note: Use a solder fume extractor or work in a well-ventilated area. If necessary, wear a respirator mask. Caution: Some people may be sensitive to certain materials or chemicals contained in the product. Note: If skin irritation or allergic reactions occur, stop use and, if necessary, consult a doctor. Caution: Keep the product out of the reach of children and pets to avoid accidental contact and swallowing of small parts. Note: Store the product in a safe, closed container when not in use. Attention: Avoid contact of the product with food and drinks. Note: Do not store or use the product near food to prevent contamination. Although our product complies with the requirements of the RoHS Directive (2011/65/EU) and does not contain any hazardous substances in quantities above the permitted limits, residues may still be present. Observe the following safety instructions to avoid chemical hazards: Caution: Soldering can produce fumes that can be harmful to health. Note: Use a solder fume extractor or work in a well-ventilated area. If necessary, wear a respirator mask. Caution: Some people may be sensitive to certain materials or chemicals contained in the product. Note: If skin irritation or allergic reactions occur, stop use and, if necessary,

consult a doctor. Caution: Keep the product out of the reach of children and pets to avoid accidental contact and swallowing of small parts. Note: Store the product in a safe, closed container when not in use. Attention: Avoid contact of the product with food and drinks. Note: Do not store or use the product near food to prevent contamination. The product contains sensitive electronic components and sharp edges. Improper handling or assembly can result in injury or damage. Observe the following safety instructions to avoid mechanical hazards: Attention: The product's circuit board and connectors may have sharp edges. Use caution to avoid cuts. Note: Wear appropriate protective gloves when handling and assembling the product. Caution: Avoid excessive pressure or mechanical stress on the board and components. Note: Only mount the product on stable and flat surfaces. Use appropriate spacers and housings to minimize mechanical stress. Attention: Make sure the product is securely fastened to prevent accidental slipping or falling. Note: Use appropriate support or secure mounting in enclosures or on mounting plates. Caution: Make sure all cable connections are connected securely and correctly to avoid strain and accidental unplugging. Note: Route cables so that they are not under tension and do not pose a tripping hazard. The product operates with electrical voltages and currents that, if used improperly, can result in electric shocks, short circuits or other hazards. Observe the following safety instructions to avoid electrical hazards: Attention: Use the product only with the specified voltages. Note: The performance limits of the product can be found in the associated data sheet Caution: Avoid short circuits between the connectors and components of the product Note: Make sure that no conductive objects touch or bridge the circuit board. Use insulated tools and pay attention to the arrangement of connections. Caution: Do not perform any work on the product when it is connected to a power source. Note: Disconnect the product from power before making any circuit changes or connecting or removing components. Caution: Do not exceed the specified current ratings for the product's inputs and outputs. Note: The performance limits of the product can be found in the technical specifications or in the data sheet Attention: Make sure that the power sources used are stable and correctly sized. Note: Only use tested and suitable power supplies to avoid voltage fluctuations and overloads. Attention: Maintain sufficient distance from live parts to avoid accidental contact. Note: Ensure that the cabling is arranged safely and clearly according to the voltage used. Caution: Use insulating housings or protective covers to protect the product from direct contact. Note: Place the product in a non-conductive case to avoid accidental touching and short circuits. The product and the components on it may become warm during operation. Improper handling or overloading the product can result in burns, damage or fire. Observe the following safety instructions to avoid thermal hazards: Caution: Make sure the product is used within recommended operating temperatures. Note: The recommended operating temperature range is typically between -40°C and +85°C. Check the specific information in the product data sheet. Attention: Do not place the product near external heat sources such as radiators or direct sunlight. Note: Ensure that the product is operated in a cool and well-ventilated area. Attention: Make sure the product is well ventilated to avoid overheating. Note: Use fans or heat sinks when operating the product in a closed enclosure or in an environment with limited air circulation. Attention: Mount the product on heat-resistant surfaces and in heat-resistant housings. Note: Use enclosure materials that can withstand high temperatures to avoid damage or fire hazard. Caution: Implement temperature monitoring when using an enclosure and, if necessary, protection mechanisms that shut down the product if it overheats. Note: Note: Use temperature sensors and appropriate software to monitor the temperature of the product and shut down the system if necessary. Caution: Avoid overloads that can cause excessive heating of components. Note: To prevent overheating, do not exceed the specified current and voltage limits. Caution: Short circuits can generate significant heat and cause fires. Note: Make sure that all connections are correct and secure and that no conductive objects can accidentally cause short circuits.



## Table of Contents

Introduction.....	3
Working principle.....	4
The output.....	6
Specifications.....	7
The pinout.....	7
How to set-up Arduino IDE.....	8
How to set-up the Raspberry Pi and Python.....	12
Connecting the module with Atmega328p.....	13
Sketch example.....	14
Connecting the module with Raspberry Pi.....	18
Python script.....	19



## **Introduction**

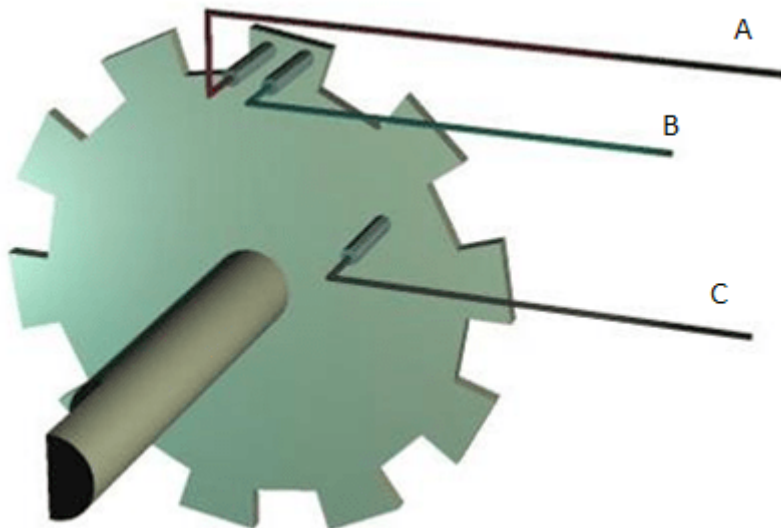
The KY-040 rotary encoder module is a rotary input device (as in knob) that provides an indication of how much the knob has been rotated and in what direction it is rotating. A rotary encoder is an electro-mechanical device that converts rotational motion into digital or analog information. It is a great device for stepper and servo motor control.

A rotary encoder has a fixed number of positions per revolution. These positions are easily felt as small “clicks” when you turn the encoder shaft. The KY-040 rotary encoder module has 20 positions per one full rotation.

## Working principle

There are two switches inside the encoder. One switch connects pin A (or CLK) to pin C and the other switch connects pin B (or DT) to C. In each encoder position, both switches are either opened or closed.

The illustration below is the representation of how the switch is constructed.

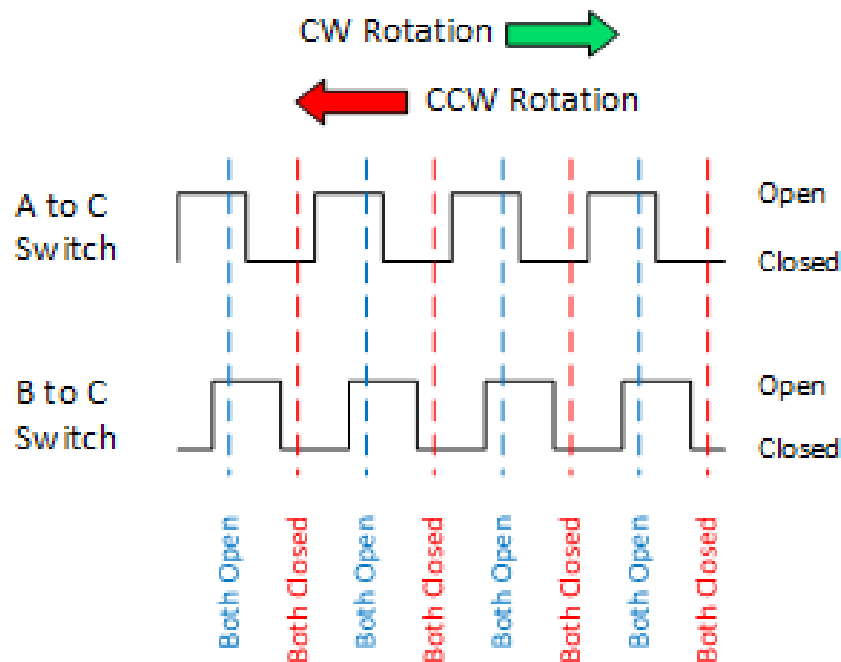


As you can see, the angular position of the A terminal and the B terminal is the following:

- » Rotating the switch clockwise causes the switch connecting A and C to change states first.
- » Rotating the switch counter-clockwise causes the switch connecting B and C to change states first.

# Az-Delivery

If the opening and closing of the switches is represented as wave forms, it would look something as shown on the following image:



Essentially, determining which switch changed states first is how the direction of rotation is determined.

- » If A changed states first, the switch is rotating in a clockwise direction.
- » If B changed states first, the switch is rotating in a counter-clockwise direction.



## The output

The module is designed so that the *LOW* state is an output when the switches are closed and the *HIGH* state when the switches are open. The *LOW* state is generated by placing a ground at pin *C* and passing it to the *CLK* and *DT* pins when switches are closed. The *HIGH* state is generated connecting the power supply via pull-up resistors to the *CLK* and *DT* pins when switches are open.

There is a push button that is integral to the encoder. If the shaft is pressed, a normally open switch closes. The *SW* pin outputs the state of this push button.

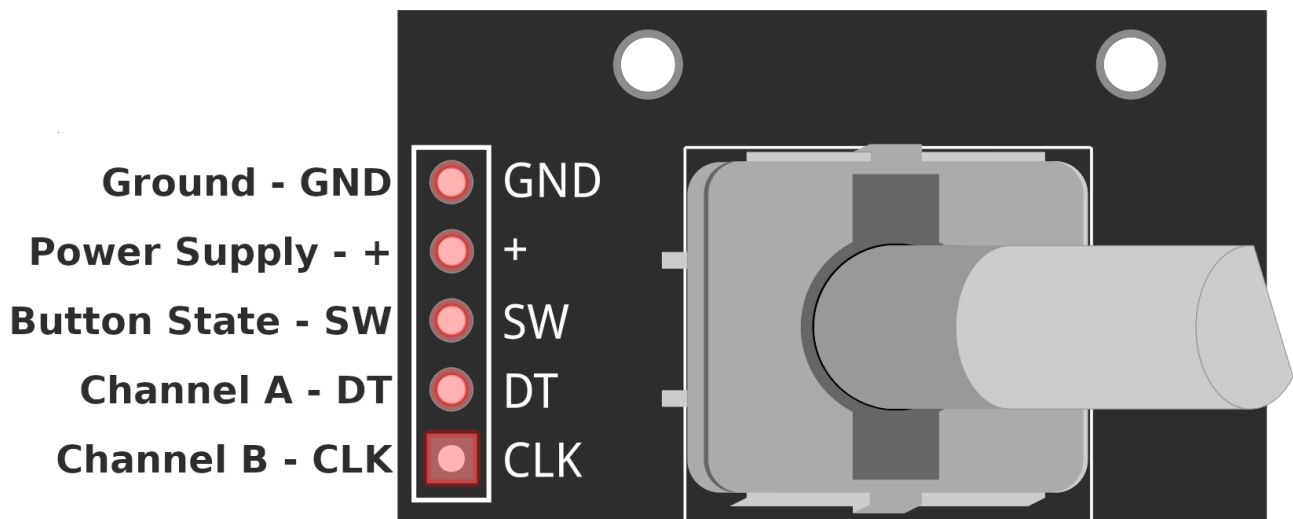


## Specifications

- » Operating voltage range: from 3.3V to 5V DC
- » Operating temperature range: from -40°C to 85°C
- » Output: digital
- » Dimensions: 17 x 29 x 30mm [0.7 x 1.14 x 1.2in]

## The pinout

The KY-040 rotary encoder module has five pins. The pinout diagram is shown on the following image:



## How to set-up Arduino IDE

If the Arduino IDE is not installed, follow the [link](#) and download the installation file for the operating system of choice.

### Download the Arduino IDE



The screenshot shows the Arduino IDE download page. On the left, there is a teal circle containing a white infinity symbol with a minus sign on the left and a plus sign on the right. To the right of this icon, the text reads: **ARDUINO 1.8.9**  
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 the following links and options:

- Windows** Installer, for Windows XP and up
- Windows** ZIP file for non admin install
- Windows app** Requires Win 8.1 or 10
- Get** (with Windows logo icon)
- Mac OS X** 10.8 Mountain Lion or newer
- Linux** 32 bits
- Linux** 64 bits
- Linux** ARM 32 bits
- Linux** ARM 64 bits
- [Release Notes](#)
- [Source Code](#)
- [Checksums \(sha512\)](#)

For *Windows* users, double click on the downloaded .exe file and follow the instructions in the installation window.

# Az-Delivery

For *Linux* users, download a file with the extension *.tar.xz*, which has to be extracted. When it is extracted, go to the extracted directory and open the terminal in that directory. Two *.sh* scripts have to be executed, the first called *arduino-linux-setup.sh* and the second called *install.sh*.

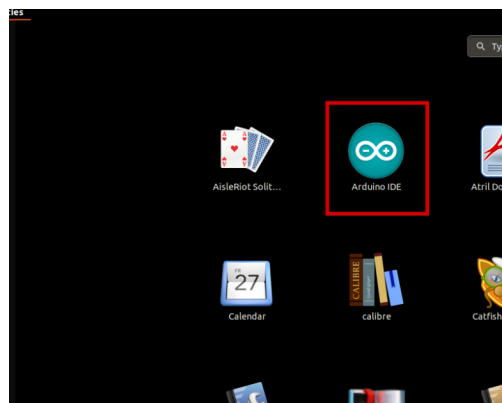
To run the first script in the terminal, open the terminal in the extracted directory and run the following command:

```
sh arduino-linux-setup.sh user_name
```

**user\_name** - is the name of a superuser in the Linux operating system. A password for the superuser has to be entered when the command is started. Wait for a few minutes for the script to complete everything.

The second script called *install.sh* script has to be used after installation of the first script. Run the following command in the terminal (extracted directory): **sh install.sh**

After the installation of these scripts, go to the *All Apps*, where the *Arduino IDE* is installed.



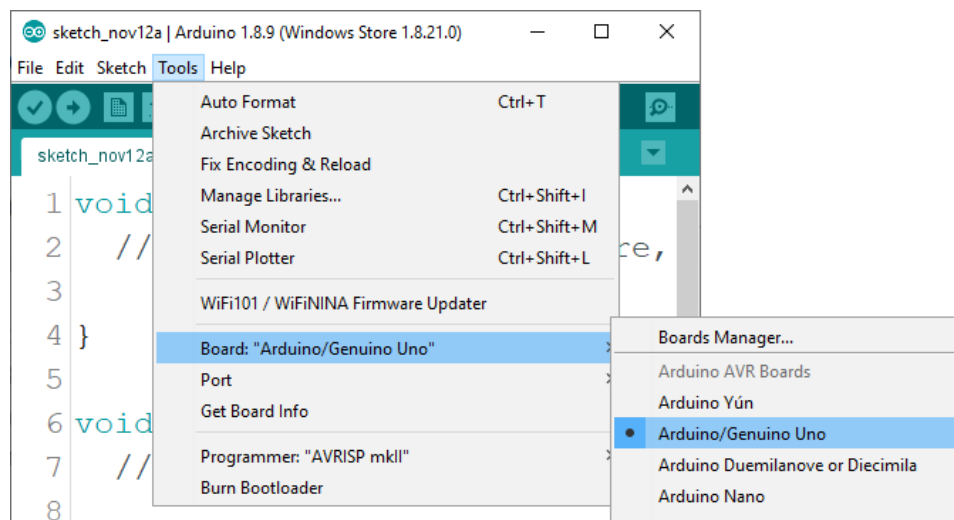
# Az-Delivery

Almost all operating systems come with a text editor preinstalled (for example, *Windows* comes with *Notepad*, *Linux Ubuntu* comes with *Gedit*, *Linux Raspbian* comes with *Leafpad*, etc.). All of these text editors are perfectly fine for the purpose of the eBook.

Next thing is to check if your PC can detect an Atmega328p board. Open freshly installed Arduino IDE, and go to:

*Tools > Board > {your board name here}*

*{your board name here}* should be the *Arduino/Genuino Uno*, as it can be seen on the following image:

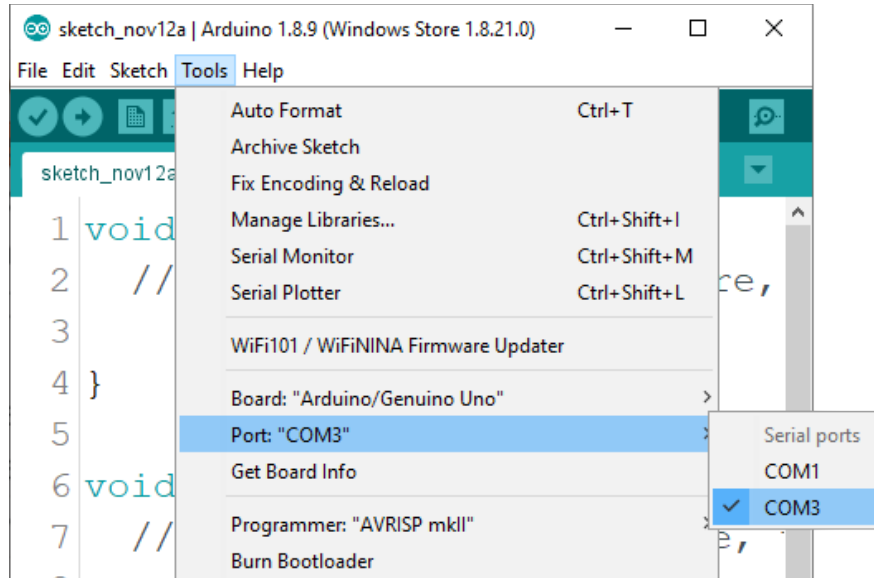


The port to which the Atmega328p board is connected has to be selected. Go to: *Tools > Port > {port name goes here}*

and when the Atmega328p board is connected to the USB port, the port name can be seen in the drop-down menu on the previous image.

# Az-Delivery

If the Arduino IDE is used on Windows, port names are as follows:



For *Linux* users, for example port name is `/dev/ttyUSBx`, where *x* represents integer number between 0 and 9.



## How to set-up the Raspberry Pi and Python

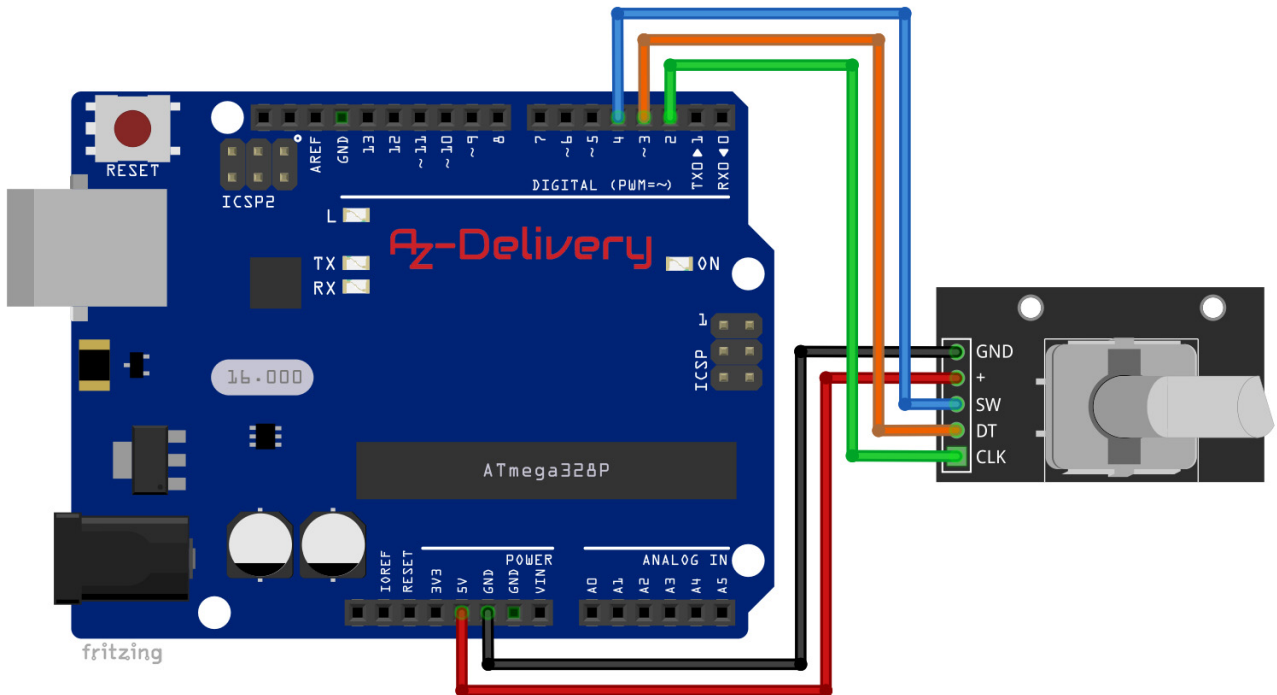
For the Raspberry Pi, first the operating system has to be installed, then everything has to be set-up so that it can be used in the *Headless* mode. The *Headless* mode enables remote connection to the Raspberry Pi, without the need for a *PC* screen Monitor, mouse or keyboard. The only things that are used in this mode are the Raspberry Pi itself, power supply and internet connection. All of this is explained minutely in the free eBook:

[Raspberry Pi Quick Startup Guide](#)

The *Raspbian* operating system comes with *Python* preinstalled.

## Connecting the module with Atmega328p

Connect the KY-039 module with the Atmega328p as shown on the following connection diagram:



KY-040 pin	>	Mc pin
+ (VCC)	>	5V
GND	>	GND
CLK	>	D2
DT	>	D3
SW	>	D4

Red wire

Black wire

Green wire

Orange wire

Blue wire

# Az-Delivery

## Sketch example

```
#define CLK_PIN 2
#define DT_PIN 3
#define SW_PIN 4
int position = 0;
int last_position = 0;
int n = 0;
bool taster = LOW;
bool last_taster = LOW;
void setup() {
    pinMode(CLK_PIN, INPUT_PULLUP);
    pinMode(DT_PIN, INPUT_PULLUP);
    pinMode(SW_PIN, INPUT_PULLUP);
    Serial.begin(9600);
}
void loop() {
    n = digitalRead(CLK_PIN);
    taster = !digitalRead(SW_PIN);
    if(taster != last_taster) {
        Serial.print(position);
        Serial.print("|");
        Serial.println(taster);
        delay(10);
        last_taster = taster;
    }
}
```

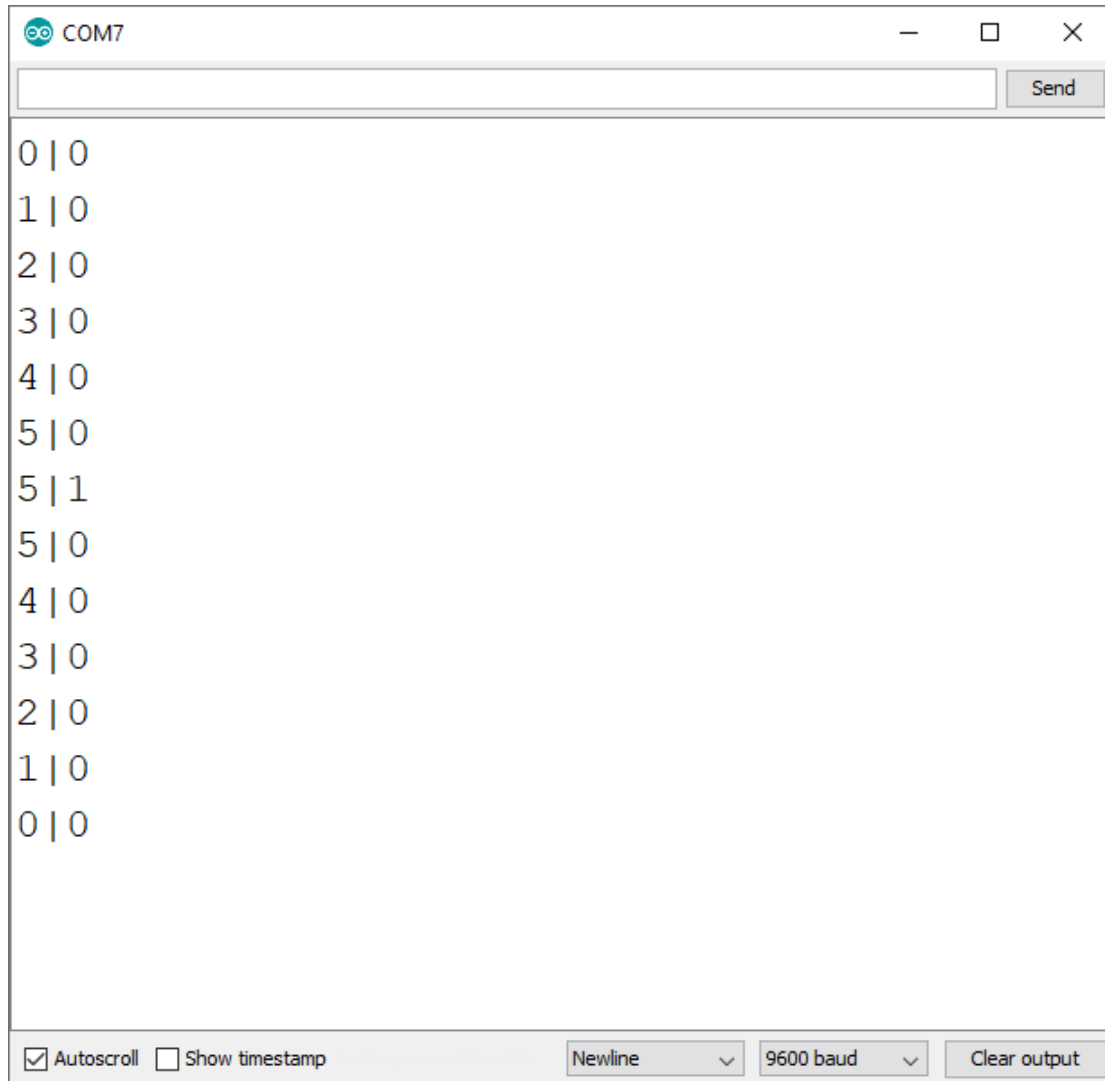


# Az-Delivery

```
// one tab
if((last_position == 0) && (n == HIGH)) {
    if(digitalRead(DT_PIN) == LOW) {
        position++;
    }
    else {
        position--;
    }
    Serial.print(position);
    Serial.print("|");
    Serial.println(taster);
}
last_position = n;
}
```

# Az-Delivery

Upload the sketch to the Atmega328p and open Serial Monitor (*Tools > Serial Monitor*). The result should look like the output on the following image:



To get these values, move or press the shaft of the rotary encoder.

**NOTE:** There is a lot of noise in the output of the module. A high number of errors is expected with high noise. How to clear output out of the noise is not in the scope of this eBook.

# Az-Delivery

The sketch starts with creating three macros called *CLK\_PIN*, *DT\_PIN* and *SW\_PIN*. These macros represent the digital pins of Atmega328p on which pins of the module are connected.

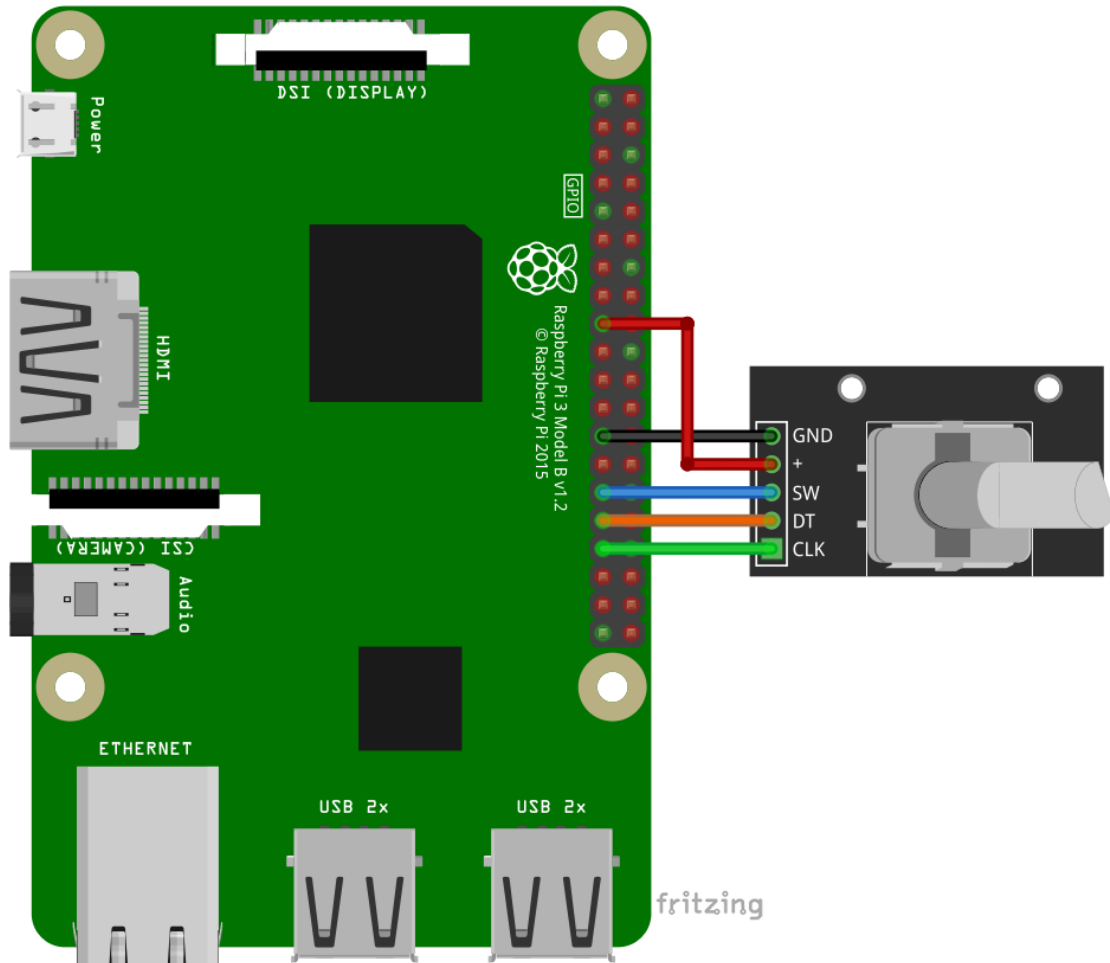
Next, several variables are created which are used in the algorithm that detect when and in which direction the shaft of the encoder is moving. The variable called *position* holds the value which represents the current position of the shaft. At the beginning of the sketch execution this value is zero, but when the shaft of the encoder is moved, it can hold both positive or negative values depending on the direction of moving shaft (positive value - clockwise direction, negative value - counterclockwise direction).

In the *setup()* function the pin modes for all three pins are set to *INPUT\_PULLUP*. Next, the serial communication is started with a baud rate of *9600bps*.

In the *loop()* function the state of the module pins is read. Then, the algorithm for detecting movement of the shaft is executed, checking the previously read states of the pins. This algorithm is not covered in this eBook. The output of this algorithm is a message displayed in the Serial Monitor (image of Serial Monitor on the previous page).

## Connecting the module with Raspberry Pi

Connect the module with the Raspberry Pi as shown on the following connection diagram:



### KY-040 pin > Raspberry Pi pin

+ (VCC)	>	3V3	[pin 17]
GND	>	GND	[pin 25]
SW	>	GPIO5	[pin 29]
DT	>	GPIO6	[pin 31]
CLK	>	GPIO13	[pin 33]

**Red wire**

**Black wire**

**Orange wire**

**Blue wire**

**Green wire**



## Python script

```
import RPi.GPIO as GPIO
from time import sleep

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

SW_PIN = 5
DT_PIN = 6
CLK_PIN = 13

GPIO.setup(CLK_PIN, GPIO.IN)
GPIO.setup(DT_PIN, GPIO.IN)
GPIO.setup(SW_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)

position = 0

a = 0
b = 0

a_last = 1
b_last = 1

taster = False
last_taster = False
```

# Az-Delivery

```
print('[Press CTRL + C to end the script!]\n')
try: # Main program loop
    while True:
        a, b = GPIO.input(CLK_PIN), GPIO.input(DT_PIN)
        taster = not GPIO.input(SW_PIN)
        if taster != last_taster:
            print('{}|{}'.format(position, taster))
            sleep(0.01)
            last_taster = taster
        if a != a_last or b != b_last:
            if a == 0 and b == 1:
                while not (a == 1 and b == 1):
                    a = GPIO.input(CLK_PIN)
                    b = GPIO.input(DT_PIN)
                position += 1
                print('{}|{}'.format(position, taster))
            if a == 1 and b == 0:
                while not (a == 1 and b == 1):
                    a = GPIO.input(CLK_PIN)
                    b = GPIO.input(DT_PIN)
                position -= 1
                print('{}|{}'.format(position, taster))
            a_last = a
            b_last = b

except KeyboardInterrupt:
    print('\nScript end!\n')

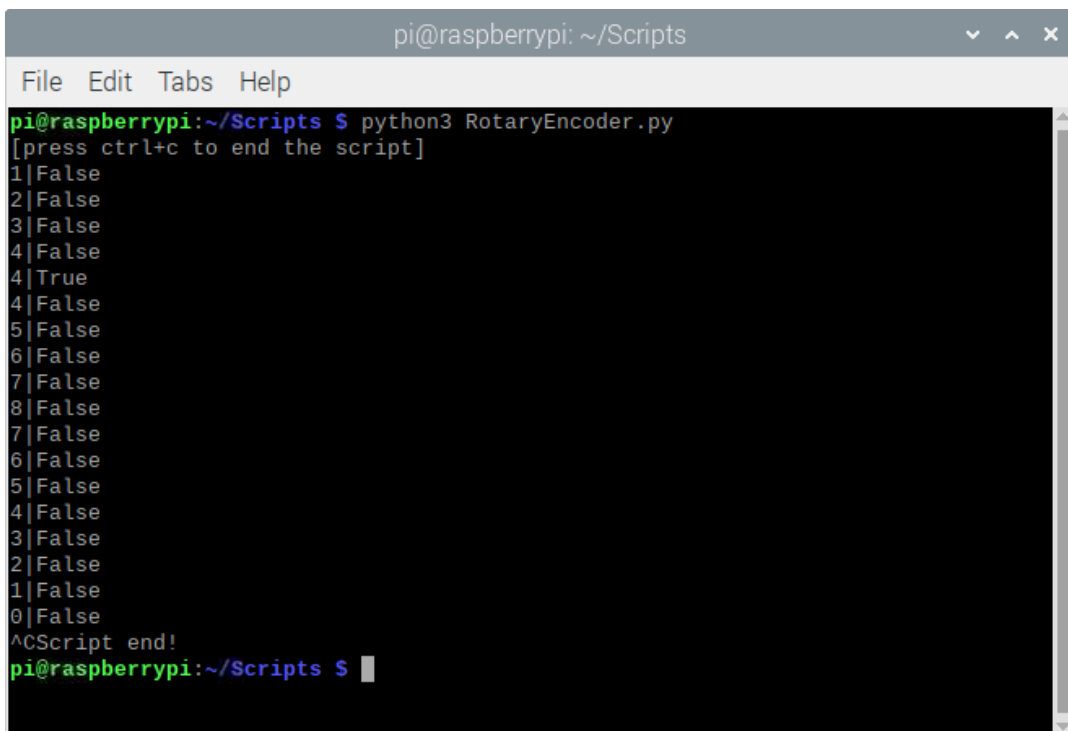
finally:
    GPIO.cleanup()
```

# Az-Delivery

Save the script by the name *RotaryEncoder.py*. To run the script, open the terminal in the directory where the script is saved and run the following command:

**python3 RotaryEncoder.py**

The result should look like the output on the following image:



```
pi@raspberrypi: ~/Scripts
File Edit Tabs Help
pi@raspberrypi:~/Scripts $ python3 RotaryEncoder.py
[press ctrl+c to end the script]
1|False
2|False
3|False
4|False
4|True
4|False
5|False
6|False
7|False
8|False
7|False
6|False
5|False
4|False
3|False
2|False
1|False
0|False
^CScript end!
pi@raspberrypi:~/Scripts $
```

To stop the script press *CTRL* + *C* on the keyboard. To get these values, move or press the shaft of the rotary encoder.

The script does the same thing on the same way as the sketch for the Atmega328p, the only difference is that the script is written in *Python*.

**NOTE:** There is a lot of noise in the output of the module. A high number of errors is expected with high noise. How to clear output out of the noise is not in the scope of this eBook.



Now it is the time to learn and make your own projects. You can do that with the help of many example scripts and other tutorials, which can be found on the internet.

**If you are looking for the high quality microelectronics and accessories, AZ-Delivery Vertriebs GmbH is the right company to get them from. You will be provided with numerous application examples, full installation guides, eBooks, libraries and assistance from our technical experts.**

<https://az-delivery.de>

Have Fun!

Impressum

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