Welcome!

Thank you for purchasing our *AZ-Delivery U64 LED Matrix Panel module*. On the following pages, you will be introduced to how to use and set-up this handy device.

Have fun!

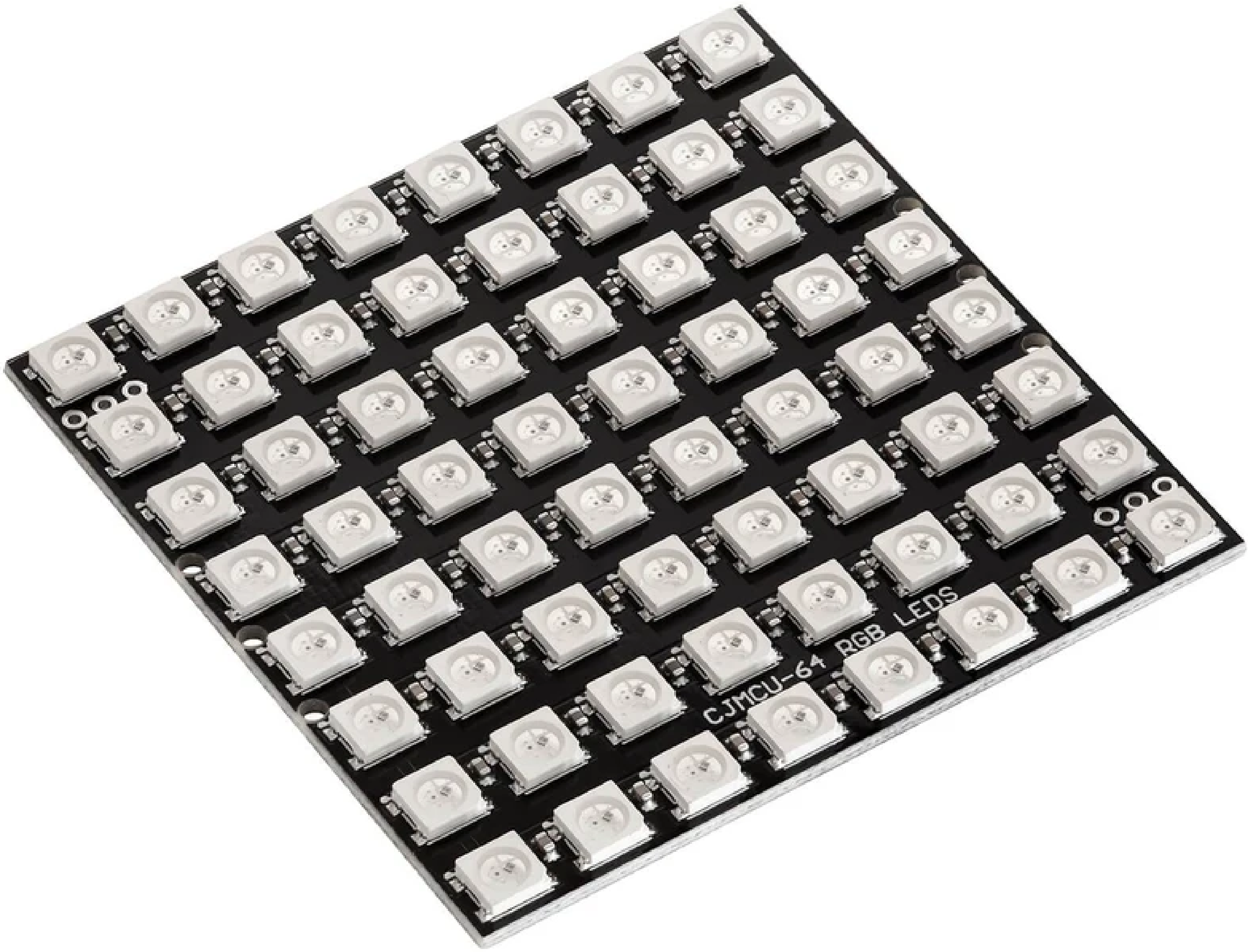


Table of Contents

Introduction

The U64 LED matrix panel module is a board with 64 (8x8) RGB LEDs, also called pixels. It needs only one microcontroller pin to control all the pixels using a single-wire control protocol.

Each LED pixel consists of 3 LED diodes and integrated WS2812 control chip. All LEDs in the matrix are connected in series and are individually addressable.

The state, brightness, and color of all LEDs in each pixel can be controlled individually by a microcontroller.

The LED panel board has a two ports (Input and output) so that multiple panels can be connected in series (daisy chaining).

The LED matrix can be used for light painting, light effects, animations, etc.
It is also used in variety of applications such as LED decorative lighting, Indoor/outdoor LED video irregular screens, etc.

Specifications

Operating voltage	5VDC
Operating current	64x60mA=3840mA(3.84A)
Power consumption	11W @max. brightness (white color)
LED count	64(8x8)
LED type	WS2812B (SMD 5050)
LED color range	16milion (Full range)
Interface	IO
Dimensions	65x65x3mm (0.9x0.5x0.3inch)

Power consumption equation example:

To estimate power supply requirements for one LED matrix panel, the number of pixels has to be multiplied by 20, then the result divided by 1000 as the “rule of thumb” power supply rating in Amperes. For maximum current consumption, the number 60 is to be used (instead of 20), if absolute margin of safety is to be guaranteed for all situations.

For example:

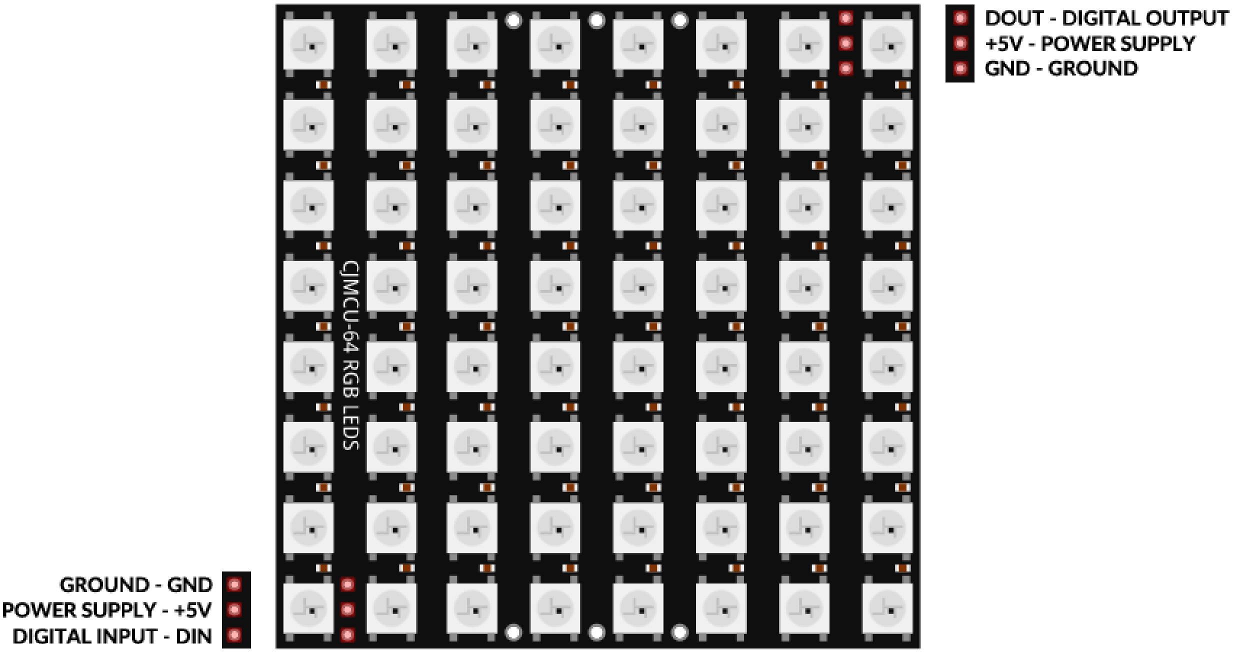
60 pixels × 20 mA ÷ 1000 = 1.2A minimum

60 pixels × 60 mA ÷ 1000 = 3.6A minimum

WARNING: When the light levels are set close to the maximum, power consumption of one matrix pannel exceeds the current limit that is supplied from the Atmega328p 5V pin. It is preferable to always use external power supply otherwise the Atmega328p may be damaged.

The pinout

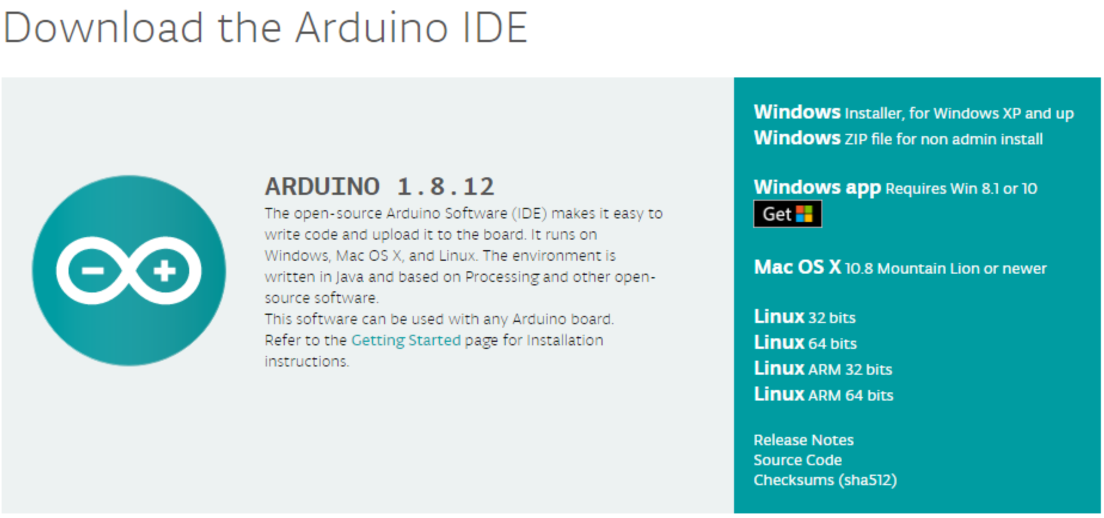
The module has six pins. The pinout is shown on the following image:



Note: Before connecting the pannel to any live (working) power supply, make sure that ground is always connected first before any other wire.

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. The Arduino IDE version used for this ebook is **1.8.12.**



Download the Arduino IDE

ARDUINO 1.8.12

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.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install
Windows app Requires Win 8.1 or 10
Get
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.

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*.

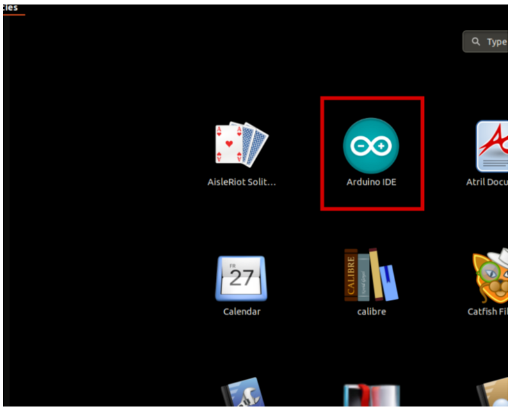
To run the first script in the terminal, open the terminal in the extracted directory and run the following command:

```
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.

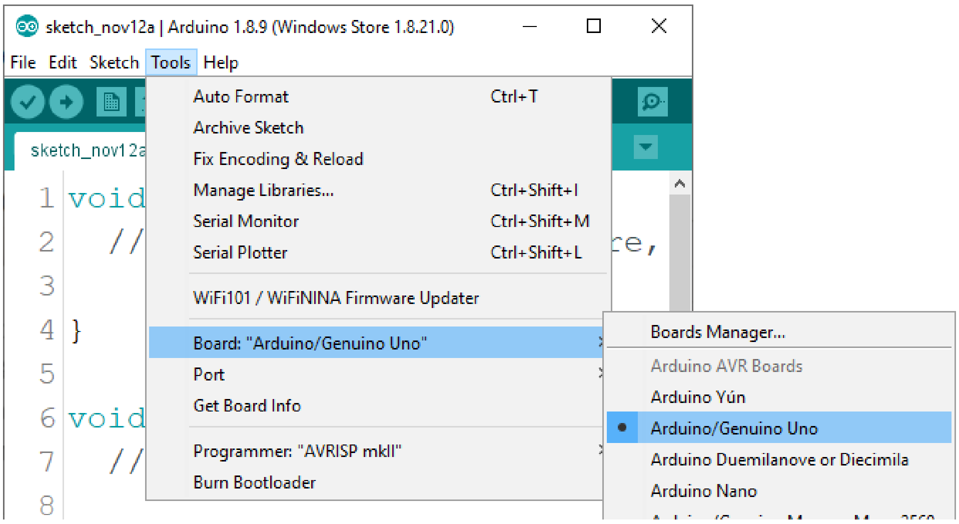
The second script, called *install.sh*, has to be used after the installation of the first script. Run the following command in the terminal (extracted directory): **sh install.sh**

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



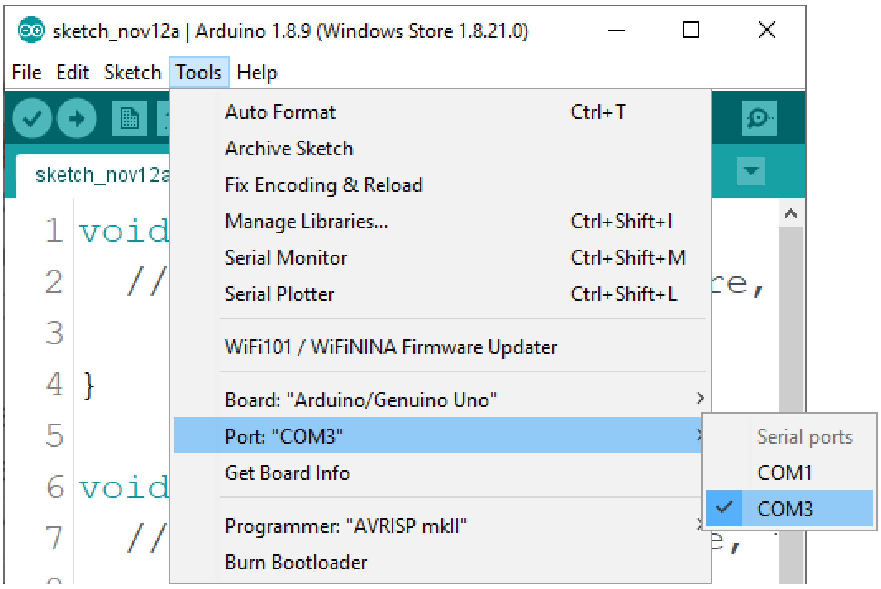
Almost all operating systems come with a text editor preinstalled (for example, *Windows* comes with *Notepad*, *Linux Ubuntu* comes with *Gedit*, *Linux Raspbian* comes with *Leafpad*, etc.). All of these text editors are perfectly fine for the purpose of the eBook.

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:


sketch_nov12a | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
sketch_nov12a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino/Genuino Uno"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino Yún
● Arduino/Genuino Uno
Arduino Duemilanove or Diecimila
Arduino Nano

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.

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


sketch_nov12a | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help
sketch_nov12a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Get Board Info
Burn Bootloader
Serial ports
COM1
COM3

Board: "Arduino/Genuino Uno"

Port: "COM3"

Programmer: "AVRISP mkII"

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

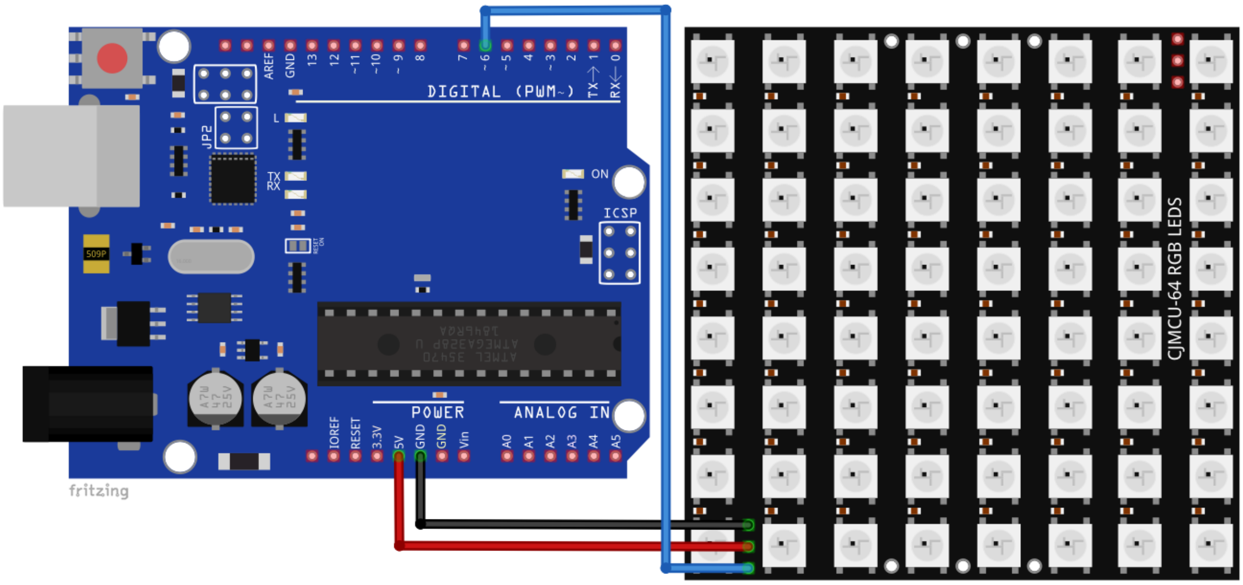
How to set-up the Raspberry Pi and Python

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 *Raspberry Pi OS* comes with *Python* preinstalled.

Connecting the module with Atmega328p

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



Module pin	Mc pin	Wire color
VDD	5V	**Red wire**
GND	GND	**Black wire**
DIN	D6	**Blue wire**

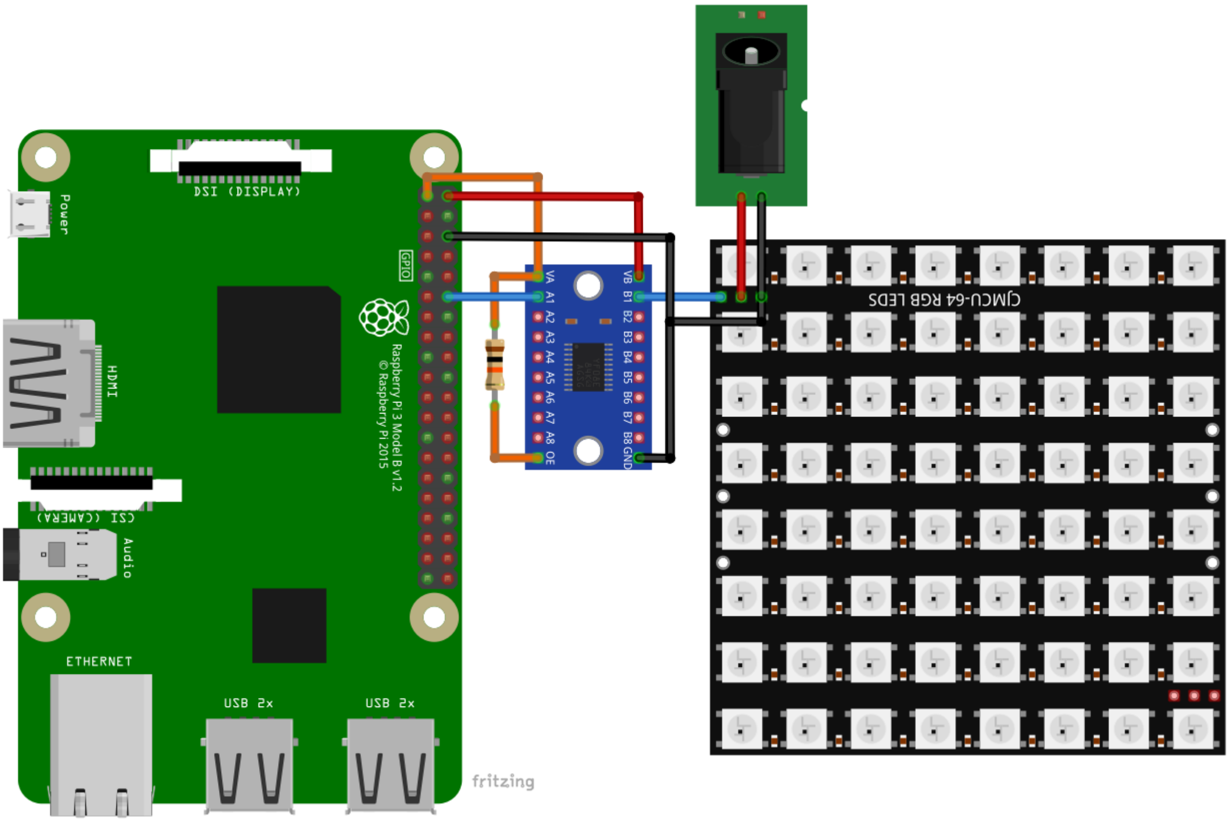
NOTE: For testing purposes, only one LED matrix panel should be connected to the Atmega328p. The brightness level in the sketch example should not exceed the value of 50 so that power consumption should not exceed the Atmega328p voltage regulator capabilities, otherwise Atmega328p may be damaged.

Sketch examples

```
#include <Adafruit_GFX.h>
#include <Adafruit_NeoMatrix.h>
#include <Adafruit_NeoPixel.h>

#define PIN 6

Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(16, 16, PIN,
  NEO_MATRIX_TOP     + NEO_MATRIX_RIGHT +
  NEO_MATRIX_COLUMNS + NEO_MATRIX_PROGRESSIVE,
  NEO_GRB            + NEO_KHZ800);

const uint16_t colors[] = {
  matrix.Color(255, 0, 0), matrix.Color(0, 255, 0), matrix.Color(0, 0, 255) };

void setup() {
  matrix.begin();
  matrix.setTextWrap(false);
  matrix.setBrightness(20); // Brightness level
  matrix.setTextColor(colors[0]);
}

int x    = matrix.width();
int pass = 0;

void loop() {
  matrix.fillScreen(0);
  matrix.setCursor(x, 0);
  matrix.print(F("AZ-DELIVERY"));
  if(--x < -50) { //84
    x = matrix.width();
    if(++pass >= 3) pass = 0;
    matrix.setTextColor(colors[pass]);
  }
  //two spaces
    matrix.show();
    delay(80);
  }
```

Connecting the module with Raspberry Pi

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



TXS0108E pin	Raspberry Pi pin	Physical pin	Wire color
VA	3V3	1	**Orange wire**
VB	5V	2	**Red wire**
GND	GND	6	**Black wire**
A1	GPIO18	12	**Blue wire**
Module pin	**TXS0108E pin**		
DIN	B1		**Blue wire**
GND	GND		**Black wire**
Module pin	**DC Power Supply**		
+5V	+5V		**Red wire**
GND	GND		**Black wire**

Note: The module pins are working in the 5V range*.* In order to use the module with the Raspberry Pi, the Logic Level Converter must be used. The module pins are working with the 5V input, since the Raspberry Pi pins are operating in the 3.3V range, there is no sufficient voltage to drive the matrix panel accordingly. For this purpose use the device called TXS0108E 8ch Logic Level Converter that AZ-Delivery offers.

Libraries and tools for Python

To use the device with the Raspberry Pi it is recommended to download an external Python library. The library that is used in this eBook is called the *Adafruit_CircuitPython_NeoPixel*.

Before the library can be used, run the following commands:

```
sudo apt-get update
sudo pip3 install rpi_ws281x adafruit-circuitpython-neopixel
sudo python3 -m pip install --force-reinstall adafruit-blinka
```

Next, to download an external library, run the following command:

```
git clone https://github.com/adafruit/Adafruit_CircuitPython_NeoPixel.git
```

To install it, first change directory to the *Adafruit_CircuitPython_NeoPixel*, by running the following command:

```
cd  Adafruit_CircuitPython_NeoPixel
```

and install the library with the following command:

```
sudo python3 setup.py install
```

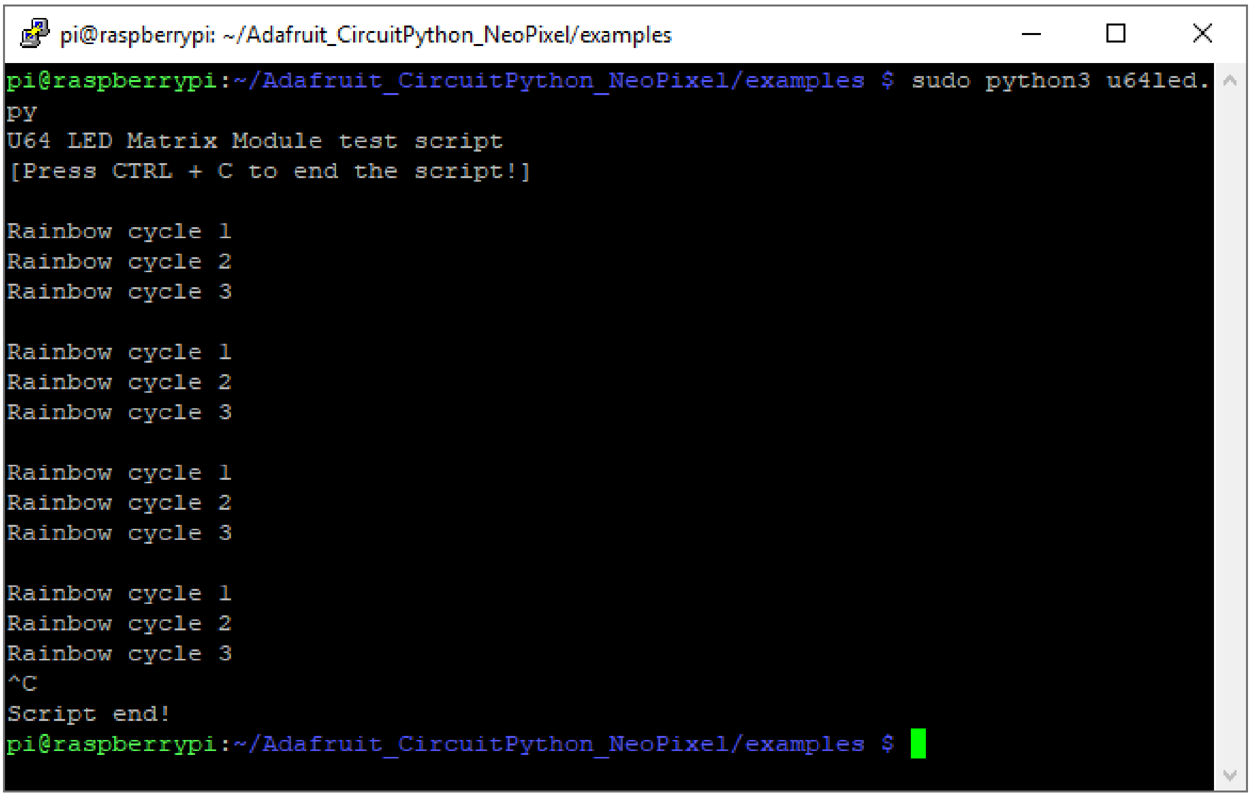
Python script

```
import time
import board
import neopixel

pixel_pin = board.D18
num_pixels = 64
ORDER = neopixel.GRB

pixels = neopixel.NeoPixel(
    pixel_pin, num_pixels, brightness=0.2, auto_write=False, pixel_order=ORDER
)

def wheel(pos):
    if pos < 0 or pos > 255:
        r = g = b = 0
    elif pos < 85:
        r = int(pos * 3)
        g = int(255 - pos * 3)
        b = 0
    elif pos < 170:
        pos -= 85
        r = int(255 - pos * 3)
        g = 0
        b = int(pos * 3)
    else:
        pos -= 170
        r = 0
        g = int(pos * 3)
        b = int(255 - pos * 3)
    return (r, g, b) if ORDER in (neopixel.RGB, neopixel.GRB) else (r, g, b,
0)
```

```
def rainbow_cycle(wait):
    for j in range(255):
        for i in range(num_pixels):
            pixel_index = (i * 256 // num_pixels) + j
            pixels[i] = wheel(pixel_index & 255)
        pixels.show()
        time.sleep(wait)

print('U64 LED Matrix Module test script')
print('[Press CTRL + C to end the script!]')

try:
    while True:
        print('\nRainbow cycle 1')
        pixels.fill((255, 0, 0))
        pixels.show()
        time.sleep(1)
        print('Rainbow cycle 2')
        pixels.fill((0, 255, 0))
        pixels.show()
        time.sleep(1)
        print('Rainbow cycle 3')
        pixels.fill((0, 0, 255))
        pixels.show()
        time.sleep(1)

        rainbow_cycle(0.001)

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

Save the script under the name *u64led.py*. To run the script open the terminal in the directory where the script is saved and run the following command: **sudo python3 u64led.py**

The result should look like as on the following image:



```
pi@raspberrypi:~/Adafruit_CircuitPython_NeoPixel/examples $ sudo python3 u64led.
py
U64 LED Matrix Module test script
[Press CTRL + C to end the script!]

Rainbow cycle 1
Rainbow cycle 2
Rainbow cycle 3

Rainbow cycle 1
Rainbow cycle 2
Rainbow cycle 3

Rainbow cycle 1
Rainbow cycle 2
Rainbow cycle 3

Rainbow cycle 1
Rainbow cycle 2
Rainbow cycle 3
^C
Script end!
pi@raspberrypi:~/Adafruit_CircuitPython_NeoPixel/examples $
```

To end the script press 'CTRL + C' on the keyboard.

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