

Getting Started with the **Grove Pi+** Starter Kit



About Grove

Grove is a modular, ready-to-use tool set. Much like Lego, it takes a building block approach to assembling electronics. The Grove system consists of a base shield and various modules with standardized connectors. The base shield allows for easy connection of any microprocessor input or output from the Grove modules, and every Grove module addresses a single function, such as a simple button or a more complex heart rate sensor. Each one comes with clear documentation and demo code to help you get started quickly.

GrovePi+

GrovePi+ is a system that brings Grove sensors to the Raspberry Pi. GrovePi+ is the newest version of this system compatible with Raspberry model B, B+, 2 and 3. Grove is an easy to use collection of more than 100 plug and play modules that work out of the box with the Arduino platform. Imagine the power that people can get if they have a way to access hundreds of sensors directly from the Raspberry Pi. GrovePi+ is all about integrating Grove modules with the performance and ease of the Raspberry Pi.

seeed.cc/grovepi

About SeeedStudio

Seeed is a hardware innovation platform for makers to grow inspirations into differentiating products. Seeed provides accessible technologies with quality, speed and supply chain knowledge. Seeed helps productize 1 to 1,000 pcs using in-house engineering, supply chain management and agile manufacture forces. Seeed also team up with incubators, China tech ecosystem, investors and distribution channels to portal Maker startups beyond.

<http://www.seeedstudio.com>

About Dexter Industries

Dexter Industries develops robotic sensors for the Raspberry Pi, LEGO® MINDSTORMS® system, and Arduino that are easy to use, fun, and have real-life applications for scientists and engineers ... and those aspiring to be.

Dexter Industries' products have been used around the world in universities, classrooms, camps, and labs.

<http://www.dexterindustries.com>

About Problems

If you encounter any problems, please refer to the following website for solutions.

For all the details of GrovePi+ :

seeed.cc/grovepi

<http://www.dexterindustries.com/GrovePi/>

For all the open source resources of GrovePi+ :

<https://github.com/DexterInd/GrovePi>

For technology support :

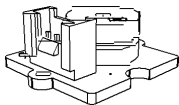
<http://forum.dexterindustries.com/c/grovepi>

Table Of Contents

Brief Info about Grove	3
Setting up the SD card for Raspberry Pi	7
Option 1: Configuring your own image	7
Option 2: Download and use the modified Raspbian image	7
Option 3: An SD Card from Dexter Industries	8
Hardware connection for GrovePi+	9
Connecting the GrovePi+ to the Raspberry Pi	9
Power up the Raspberry Pi	9
Access to Grove Pi+	10
Option 1: Access as Desktop	10
Option 2: Access via Ethernet Cable	11
Option 3: Access via WiFi	13
Running Examples of Grove Pi+	16
LED Blink	16
LED Fade	18
Buzzer	22
Ultrasonic	23
Home Weather Display	25
Twitter the Weather	30
FAQ's, Troubleshooting and getting help	32

Brief Info about the Grove

Grove - Buzzer



What have we got here?

This is a simple yet enjoyable twig to use. The piezo can be connected to digital outputs, and will emit a tone when the output is high. Alternatively it can be connected to an analog pulse-width modulation output to generate various tones and effects.

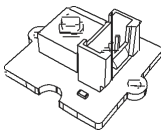
How to control:

One digital input pin.

More details:

<http://www.seeedstudio.com/depot/Grove-Buzzer-p-768.html>

Grove - Button



What have we got here?

This product is a momentary push button. It contains one independent "momentary on/o " button. "Momentary" means that the button rebounds on its own after it is released. The button outputs a HIGH signal when pressed, and LOW when released.

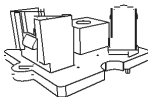
How to control:

One digital input pin

More details:

<http://www.seeedstudio.com/depot/Grove-Button-p-766.html>

Grove - Red LED



What have we got here?

This product is designed for monitoring controls from digital ports. It can be mounted to the surface of your box or desk easily and used as pilot lamp for power or signal. Its brightness can be adjusted by the potentiometer on the board.

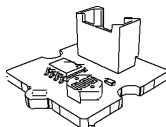
How to control:

One digital input pin

More details:

<http://www.seeedstudio.com/depot/Grove-Red-LED-p-1142.html>

Grove - Light Sensor



What have we got here?

The Light sensor module detects the light intensity of the environment. The resistance of the sensor decreases when the light intensity of the environment increases.

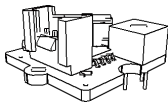
How to control:

One analog output pin.

More details:

<http://www.seeedstudio.com/depot/Grove-Light-Sensor-p-746.html>

Grove - Sound Sensor



What have we got here?

The Sound sensor module is a simple microphone. It can be used to detect the sound strength of the environment. The resistance of the sensor decreases when the sound intensity of the environment increases. The value of output can be adjusted by the potentiometer.

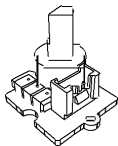
How to control :

One analog output pin.

More details:

<http://www.seeedstudio.com/depot/Grove-Sound-Sensor-p-752.html>

Grove - Rotary Angle Sensor



What have we got here?

The potentiometer twig produces analog output between 0 and Vcc on the D1 connector. The angular range is 300 degrees with a linear change in value. This sensor is also known as a rotary angle sensor.

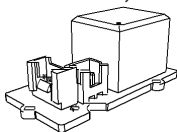
How it works :

One analog output pin.

More details:

<http://www.seeedstudio.com/depot/Grove-Rotary-Angle-Sensor-p-770.html>

Grove - Relay



What have we got here?

The Relay is a digital normally open switch that controls a relay capable of switching high voltages. When set to HIGH, the LED will light up and the relay will close to allow currents to flow. The peak voltage capability is 250V at 10 amps.

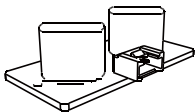
How it works:

One digital input pin.

More details:

<http://www.seeedstudio.com/depot/Grove-Relay-p-769.html>

Grove - Ultrasonic Ranger Sensor



What have we got here?

This Grove-ultrasonic sensor is a non-contact distance measurement module. The sensor is highly modular and can be used in many different projects.

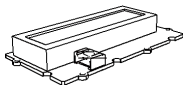
How it works:

One digital output pin.

More details:

<http://www.seeedstudio.com/depot/BackorderGrove-Ultrasonic-Ranger-p-960.html>

Grove - RGB Backlight LCD



What have we got here?

This product shows text and character messages on the display. The background can be changed across the red, green, and blue spectrum.

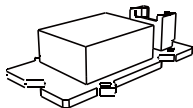
How it works:

I2C Connector

More details:

<http://www.seeedstudio.com/depot/Grove-LCD-RGB-Backlight-p-1643.html>

Grove - Temperature Humidity Sensor



What have we got here?

Humidity sensor, and it is used in environmental monitoring. It consists of a capacitive sensor element used for measuring relative humidity and a negative temperature coefficient (NTC) thermistor used for measuring temperature.

How it works:

One analog output pin.

More details:

<http://www.seeedstudio.com/depot/Grove-TemperatureHumidity-Sensor-Pro-p-838.html>

Setting up the SD card for the Raspberry Pi

Before you get started with the Grove Pi+, you need a working SD Card to use with the Raspberry Pi.

A standard installation of “Jessie”, “Stretch”, or Raspbian will not work with the GrovePi alone; there are some specific modifications that must be made to the software to successfully operate the GrovePi. We have listed some options below:

Option 1: Configuring your own image

Before beginning your Raspberry Pi must be connected to the internet.

Steps:

- 1). Type the following command in the command line:

```
sudo curl https://raw.githubusercontent.com/DexterInd/Raspbian_For_Robots/master/  
upd_script/fetch_grovepi.sh | bash
```

- 2). The Raspberry Pi should automatically restart. If it does not, restart the Raspberry Pi:

```
sudo reboot
```

Option 2: Download and use the modified Raspbian image

If you want to download and use our modified “Jessie” image on a dedicated SD card you can download it from the following locations:

(<http://sourceforge.net/projects/dexterindustriesraspbianavor>)

You will need a minimum 8 GB card to install this image.

Steps:

1). Download the latest image from Sourceforge:

(<http://sourceforge.net/projects/dexterindustriesraspbianflavor/>).

2). Download and install Etcher Software (<http://etcher.io/>)

3). Unzip the image you downloaded.

4). Install the image on your SD card. **Be careful to select the SD Card Drive!! If you select the wrong drive you can destroy your computer's hard disk or other information!**

Option 3: An SD Card from Dexter Industries

If you purchased a Dexter Industries SD card for your GrovePi+

(<https://www.dexterindustries.com/shop/sd-card-raspbian-wheezy-image-for-raspberry-pi/>), you can skip this step: your SD card is already configured!

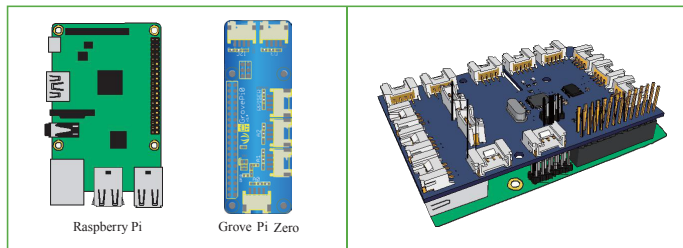
You can buy the SD card from Dexter Industries

(<https://www.dexterindustries.com/shop/sd-card-raspbian-wheezy-image-for-raspberry-pi/>).

Hardware connection for GrovePi+

Connecting the GrovePi+ to the Raspberry Pi

First, mount your GrovePi+ on the Raspberry Pi. The GrovePi+ slides over top of the Raspberry Pi as shown in the picture below.



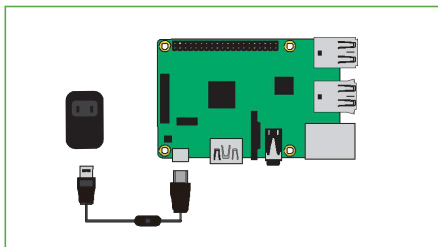
Ensure that the pins are properly aligned when stacking the GrovePi+.

Powering up the Raspberry Pi

To power the GrovePi+ and the Raspberry Pi, you can use the micro USB power port on the Raspberry Pi.

Remember to use a good power adapter capable of supplying 1A at 5V and you should be fine with the power.

If you want to run the GrovePi+ in a standalone configuration, then you should use a USB power bank.



Access to GrovePi+

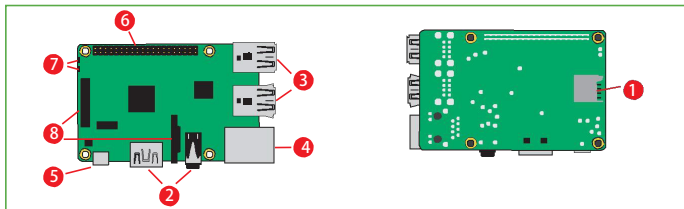
With some of your projects, hooking up a monitor and micro USB cable power supply might be impractical. We have developed extra options to make network configuration quick with the GrovePi+.

Here we explain the options we have worked into the GrovePi+ Image for Network Configuration here and offer some hints on how to get started using the GrovePi+.

These are “run once” operations. To setup, you can make these changes, and you should only have to make these changes once.

The last two options are for controlling your Raspberry Pi; once you have a connection setup, you can use the Raspberry Pi command line or VNC to start programming.

Option 1: Access as Desktop

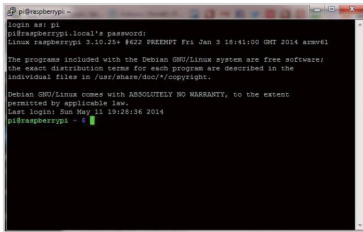


- | | |
|---|--------------------------|
| 1. MicroSD Slot | 2. HDMI and Audio output |
| 3. 4 USB2.0 ports | 4. Ethernet port |
| 5. Micro USB port, USB power port | 6. GPIO ports |
| 7. LED indicator | |
| 8. DSI/CSI, Camera Module and Display Module interface. | |

Using a monitor, mouse, keyboard: the bullet-proof option. The GrovePi+ case is designed to allow space for a keyboard, monitor, and mouse.

You may find that you need a USB hub to add a WIFI device to the GrovePi. The advantage of this configuration is that you don't have to configure the WIFI on the Raspberry Pi, or login remotely on the computer.

For more information on this, we recommend browsing the Raspberry Pi Foundations Quickstart Guide here (<http://www.raspberrypi.org/quick-start-guide>).



```
p@raspberrypi ~
login as: pi
pi@raspberrypi.local's password:
Linux raspberrypi 3.10.25+ #6622 PREEMPT Fri Jan 3 18:41:00 GMT 2014 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun May 11 19:28:36 2014
pi@raspberrypi ~ $
```

Mac

You can log into your Raspberry Pi using Bash on your Mac.

Again, we'll open up a new terminal: Press **“Command Space”**, which opens Spotlight.

Type **“Terminal”** and the terminal will come to the top.

Click on the terminal icon. In Bash, type:

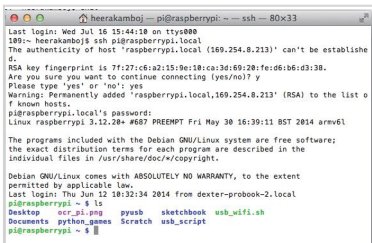
```
ssh pi@dex.local
```

You'll be prompted to verify you're trying to login to the Raspberry Pi.

Type **“yes”** and press return.

Type the password. The default password for the BrickPi image is **“raspberry”**. Type **“raspberry”** and press return.

Boom! You're logged into your Raspberry Pi and ready to start programming the GrovePi!



```
heerakamboy ~ pi@raspberrypi: ~ -- ssh -- 80x33
Last login: Wed Jul 16 15:44:18 on ttyS000
1801~ heerakamboy ssh pi@raspberrypi.local
The authenticity of host 'raspberrypi.local (169.254.8.213)' can't be established.
RSA key fingerprint is 7f:27:c6:a2:15:9e:18:ca:3d:69:20:fe:d6:b6:d3:38.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added 'raspberrypi.local,169.254.8.213' (RSA) to the list of
known hosts.
pi@raspberrypi.local's password:
Linux raspberrypi 3.12.28+ #687 PREEMPT Fri May 30 16:39:11 BST 2014 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jun 12 18:32:34 2014 from dexter-probook-2.local
pi@raspberrypi ~ $ ls
Desktop  ocr_pi.png  pyusb  sketchbook  usb_wifi.sh
Documents python_games Scratch  usb_script
pi@raspberrypi ~ $
```

Option 3: Access via WiFi

The most autonomous way to connect to your GrovePi+ is through a WiFi dongle.

The most reliable way to set the GrovePi+ up on your local WiFi network is to use

Option 1 or Option 2, and change your wifi settings on the Raspberry Pi.

You can setup WiFi on the GrovePi using the command line. Below are step-by-step instructions and some troubleshooting tips for setting up WiFi on the GrovePi+.

1. A static IP

We DO NOT recommend that you use a static IP for wifi interface with the GrovePi+.

The latest version of the GrovePi+ Raspbian comes with Bonjour. In short, this means

that instead of using an IP address, you can connect using “**raspberrypi.local**” or

“**dex.local**”. The Pi can be accessed over the network (via VNC or SSH) by any computer that has Bonjour installed.

● **Mac:** If you have a Mac, Bonjour is installed.

● **Linux:** If you have a computer with iTunes installed, Bonjour is installed.

● **Windows:** If you need to install Bonjour, you can find it on Apple's website here (<http://support.apple.com/kb/DL999>).

2. Insert WiFi Dongle

Boot the Raspberry Pi with the WiFi dongle plugged in, and open a terminal session in the Raspberry Pi.

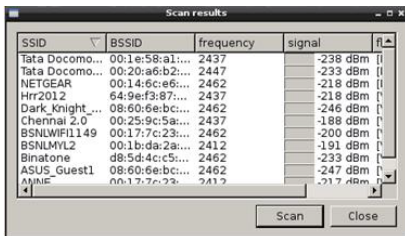
3. With the GUI

If you want to setup WiFi using the GUI, login to the Raspberry Pi.

3.1 Click on WiFi Config

The icon on the desktop labeled “WiFi Config”. Click on it. Change tabs to “Manage Networks” and click “Scan”. Clicking “Scan” reveals the WiFi networks available.

Click the network you want to connect to.



3.2 Enter your WiFi information

Enter in your WiFi information and click “Add”. Your Raspberry Pi should connect automatically to the WiFi network.

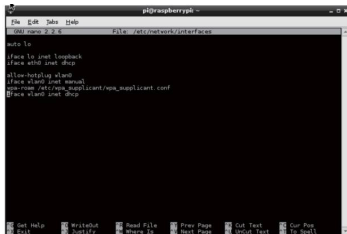
4. With the Command Line

If you want to setup WiFi using the terminal or SSH, In the terminal, open your interfaces file for editing with the following command:

```
sudo nano /etc/network/interfaces
```

4.1 Interfaces File

Your interfaces file should look like this:

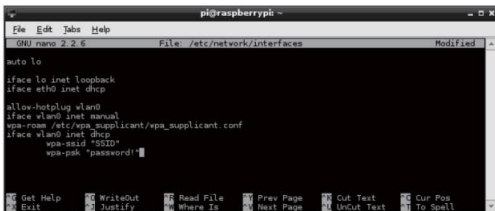


```
pi@raspberrypi ~  
File Edit Tabs Help  
GNU nano 2.2.6 File: /etc/network/interfaces  
auto lo  
iface lo inet loopback  
iface eth0 inet dhcp  
allow-hotplug wlan0  
iface wlan0 inet manual  
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf  
iface wlan0 inet dhcp  
Get Help WriteOut Read File Prev Page Cut Text Cur Pos  
Exit Justify Where Is Next Page UnCut Text To Spell
```

4.2 Edit the File

Edit the file by adding the network configuration information, and your network’s SSID and passphrase below the last line. Indent one tab!

```
wpa-ssid "ssid"  
wpa-psk "password"
```



```
pi@raspberrypi ~  
File Edit Tabs Help  
GNU nano 2.2.6 File: /etc/network/interfaces Modified  
auto lo  
iface lo inet loopback  
iface eth0 inet dhcp  
allow-hotplug wlan0  
iface wlan0 inet manual  
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf  
iface wlan0 inet dhcp  
    wpa-ssid "SSID"  
    wpa-psk "password!"  
Get Help WriteOut Read File Prev Page Cut Text Cur Pos  
Exit Justify Where Is Next Page UnCut Text To Spell
```

4.3 Test your WiFi Adapter

After adding your information, save and exit by typing “Ctrl-x” and choosing to save the file. And then reboot your raspberry pi. After reboot success, test that your Pi recognizes the WiFi Adapter. You can test by typing “ifconfig” in the terminal of the GrovePi+. If “wlan0” is showing, the wifi dongle is recognized.

```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:5c:5f:0b
          inet addr:169.254.189.61  Bcast:169.254.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:252 errors:0 dropped:0 overruns:0 frame:0
          TX packets:118 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:37506 (36.6 KiB)  TX bytes:23024 (22.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:68 errors:0 dropped:0 overruns:0 frame:0
          TX packets:68 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6484 (6.3 KiB)  TX bytes:6484 (6.3 KiB)

wlan0     Link encap:Ethernet  HWaddr 00:17:7c:22:81:f1
          inet addr:192.168.1.123  Bcast:255.255.255.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

pi@raspberrypi ~ $
```

5. Check for a Connection

Check that you are successfully connected to the network. In the command line, type:

```
ifconfig
```

When RX and TX are showing 0 bytes, no data has been exchanged. Below is a successfully connected Raspberry Pi to a wifi network.

```
wlan0     Link encap:Ethernet  HWaddr 00:17:7c:22:81:f1
          inet addr:192.168.2.10  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:222 errors:0 dropped:6 overruns:0 frame:0
          TX packets:81 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:27392 (26.7 KiB)  TX bytes:9901 (9.6 KiB)

pi@raspberrypi ~ $
```

You have successfully setup the network on the Raspberry Pi.

Running GrovePi+ Examples

Here we have a list of example projects that show just how easy it is to start a project with the GrovePi+. These GrovePi projects combine easy-to-use Grove sensors with the powerful Raspberry Pi.

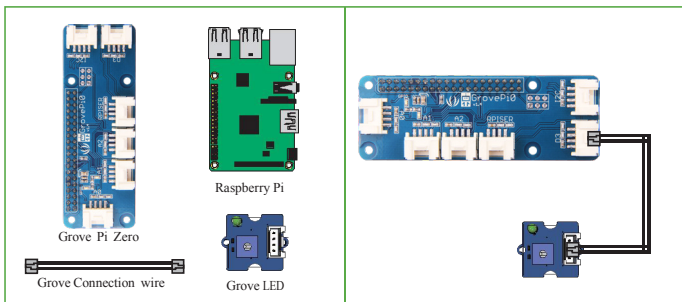
Example Project: LED Blink

This tutorial shows the simplest thing that you can do with the GrovePi: Blink a LED. This is a great first project to learn how to connect hardware to the Raspberry Pi. The blinking LED is the hardware version of the software world's “Hello World” program.

Once mastered, you can move on to more complicated projects like connecting a display or other sensors to the Raspberry Pi. This example is meant to be your first project with the GrovePi+. All the parts used in this project are available in the GrovePi+ Starter Kit.

Hardware for This Tutorial

All of the hardware comes with the GrovePi+ starter kit. Specifically, for the Raspberry Pi LED Tutorial, you'll need:



Hardware Setup: Connecting the LED to the Raspberry Pi

Connect the **LED to Port D3** and power on the Raspberry Pi, using the Grove wire connector.

Running the Program

The example program for this project is shown below. To run the program, download the GrovePi Zero Example Programs from Github.

```
cd ~/
git clone https://github.com/DexterInd/GrovePi_Zero_Example_Programs.git
cd ~/GrovePi_Zero_Example_Programs/Projects/
```

Run the LED_Blink.py python program:

```
sudo python LED_Blink.py
```

The Grove LED will start blinking.

Source Code

The source code for the example can be found here:

https://github.com/DexterInd/GrovePi_Zero_Example_Programs/blob/master/Projects/LED_Blink.py

```
# Print "Error" if communication error encountered print "Error"
import time
from grovepi import *

# Connect the Grove LED to digital port D3 on the GrovePi Zero
led = 3

pinMode(led,"OUTPUT")
time.sleep(1)

print ("This example will blink a Grove LED connected to the GrovePi+ on the port labeled D3.\nIf
you're having trouble seeing the LED blink, be sure to check the LED connection and the port
number.\nYou may also try reversing the direction of the LED on the sensor.")
print ( " ")
print ("Connect the LED to the port labeled D3!" )

while True:
    try:
        #Blink the LED
        digitalWrite(led,1)                                # Send HIGH to switch on LED
```

```
print ("LED ON!")
time.sleep(1)
digitalWrite(led,0)           # Send LOW to switch off LED
print ("LED OFF!")
time.sleep(1)

except KeyboardInterrupt:    # Turn LED off before stopping
    digitalWrite(led,0)
    break
except IOError:              # Print "Error" if communication error encountered
    print ("Error")
```

Outcome

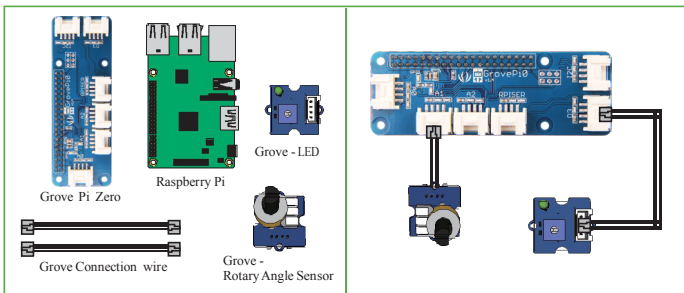
This project is very straightforward: you should see the LED blink on and off at a 1 second interval.

Example Project: LED Fade

This project shows how to fade an led with the Raspberry Pi. A rotary angle sensor (a fancy name for a potentiometer) is used to get the analog values and the Raspberry Pi sends commands to the Grove Pi to set the LED's intensity.

Setting Up The Hardware

Simply plug the LED Module in **Port D3** and the Rotary encoder in **Port A2**. That's it.



Running the Program

The example program for this project is shown below. To run the program, download the GrovePi Zero Example Programs from Github(Skip this step if you have downloaded the Example Programs).

```
cd ~/
git clone https://github.com/DexterInd/GrovePi_Zero_Example_Programs.git
cd ~/GrovePi_Zero_Example_Programs/Projects/
```

Run the LED_Fade.py python program:

```
sudo python LED_Fade.py
```

Source Code

Below is the code for running this example. You can access the code here:

https://github.com/DexterInd/GrovePi_Zero_Example_Programs/blob/master/Projects/LED_Fade.py

```
# Adjust LED brightness by rotating Potentiometer

# GrovePi + Rotary Angle Sensor (Potentiometer) + LED
# http://www.seeedstudio.com/wiki/Grove_-_Rotary_Angle_Sensor
# http://www.seeedstudio.com/wiki/Grove_-_LED_Socket_Kit
```

```
import time
import grovepi
```

```
# Connect the Rotary Angle Sensor to analog port A2
potentiometer = 2
```

```
# Connect the LED to digital port D3
led = 3
```

```
grovepi.pinMode(led,"OUTPUT")
time.sleep(1)
i = 0
```

```
while True:
```

```
    try:
```

```
        # Read resistance from Potentiometer
        i = grovepi.analogRead(potentiometer)
        print(i)
```

```
        # Send PWM signal to LED
        grovepi.analogWrite(led,i//4)
```

```
    except IOError:
        print("Error")
```

Outcome

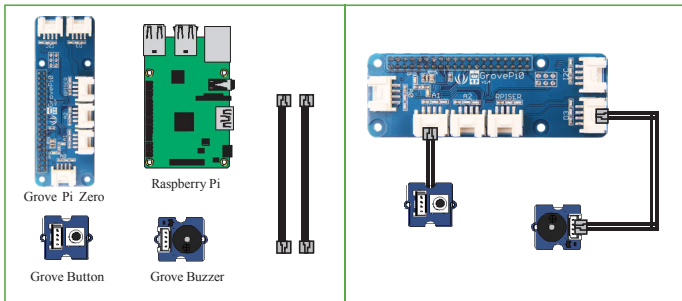
This project, like the previous projects, has a straightforward outcome: as you turn the potentiometer back and forth, you should see the LED increase and decrease in strength. In your Raspberry Pi terminal, you will see a number printing, which represents the position of the potentiometer.

Example Project: Buzzer

In this project, we connect a Grove Buzzer to the Raspberry Pi.

Setting Up The Hardware

Simply plug the Buzzer Module to Port D3. That's it.



Running the Program

The example program for this project is shown below. To run the program, copy the source code to your Raspberry Pi:

```
cd ~/
nano buzzer.py
```

Run the `buzzer.py` python program:

```
sudo buzzer.py
```

Source Code

The source code for the example can be found here:

```
# buzzer.py
#
# This is an project using the Grove Buzzer from the GrovePi zero starter kit
#
# In this project, the buzzer will making a sound per 1second.
import time
from grovepi import *
```

```
buzzer_pin = 3                #Port for buzzer

pinMode(buzzer_pin,"OUTPUT")  # Assign mode for buzzer as output
while True:
    try:
        digitalWrite(buzzer_pin,1)
        time.sleep(1)
        digitalWrite(buzzer_pin,0)
        time.sleep(1)

    except KeyboardInterrupt:  # Stop the buzzer before stopping
        digitalWrite(buzzer_pin,0)
        break
    except (IOError,TypeError) as e:
        print("Error")
```

Outcome

The outcome of this project will show up on the Raspberry Pi terminal, and with the sound of the buzzer making noise.

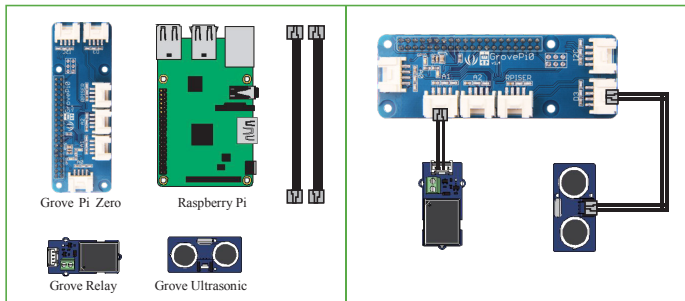
Example Project: Ultrasonic

In this project, we connect a Grove Ultrasonic to the Raspberry Pi .

You can use this project to detect the distance of any object in the front of your machine, it can be a guard robot or used to automatically open a door.

Setting Up The Hardware

Simply plug the Grove Ultrasonic Module to Port D3. That's it.



Running the Program

The example program for this project is shown below. To run the program, download the GrovePi Zero Example Programs from Github(Skip this step if you have downloaded the Example Programs).

```
cd ~/
git clone https://github.com/DexterInd/GrovePi_Zero_Example_Programs.git
```

Change directory on your Raspberry Pi to the
~/GrovePi_Zero_Example_Programs/Python folder.

```
cd ~/GrovePi_Zero_Example_Programs/Python/
```

Run the grove_ultrasonic.py python program:

```
sudo python grove_ultrasonic.py
```

Source Code

The source code for the example can be found here:

https://github.com/DexterInd/GrovePi_Zero_Example_Programs/blob/master/Python/grove_ultrasonic.py

```
#!/usr/bin/env python

# GrovePi Example for using the Grove Ultrasonic Ranger

#(http://www.seeedstudio.com/wiki/Grove_-_Ultrasonic_Ranger)

# The GrovePi connects the Raspberry Pi and Grove sensors. You can learn
#more about GrovePi here: http://www.dexterindustries.com/GrovePi
# Have a question about this example? Ask on the forums here:
#http://forum.dexterindustries.com/c/grovepi

import grovepi

# Connect the Grove Ultrasonic Ranger to digital port D3
ultrasonic_ranger = 3

while True:
    try:
        # Read distance value from Ultrasonic
        print(grovepi.ultrasonicRead(ultrasonic_ranger))

    except TypeError:
        print ("Error")
    except IOError:
        print ("Error")
```

Outcome

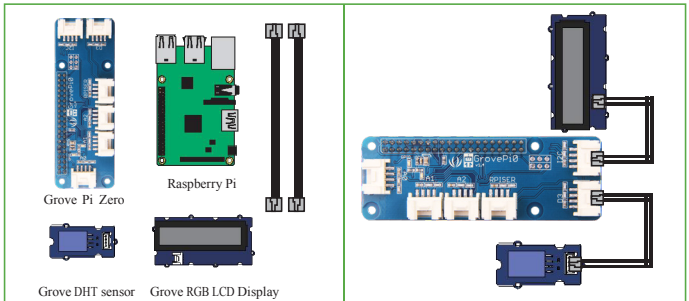
The outcome of this project will show up on the Raspberry Pi terminal. You should put your hand close to the ultrasonic sensor: Raspberry Pi will display the distance between the sensor and your hand.

Example Project: Home Weather Display

In this project, we use a Grove DHT (Digital Humidity and Temperature) sensor as a Raspberry Pi temperature sensor. This project uses a Grove RGB LCD display, connected to the Raspberry Pi, to show the temperature and humidity. You can use this project as designed: a simple weather station for your home.

Connecting the Raspberry Pi temperature sensor

Connect the DHT sensor to Port D3 and the RGB LCD display to any of the I2C ports. Now power on the Raspberry Pi.



Running the Program

The example program for this project is shown below. To run the program, download the GrovePi Zero Example Programs from Github(Skip this step if you have downloaded the Example Programs).

```
cd ~/
git clone https://github.com/DexterInd/GrovePi_Zero_Example_Programs.git
```

Change directory on your Raspberry Pi to the
~/GrovePi_Zero_Example_Programs/Projects folder.

```
cd ~/GrovePi_Zero_Example_Programs/Projects/
```

Run the Home_Weather_Display.py program:

```
sudo python Home_Weather_Display.py
```

Source Code

The source code for the example can be found here:

https://github.com/DexterInd/GrovePi_Zero_Example_Programs/blob/master/Projects/Home_Weather_Display.py

```
# Home_Weather_Display.py

#

# This is an project for using the Grove RGB LCD Display and the Grove DHT Sensor from the GrovePi
starter kit.

# - Connect the RGB LCD Display to the I2C Port

# - Connect the blue Temperature and Humidity sensor to Por D3

# In this project, the Temperature and humidity from the DHT sensor is printed on the RGB-LCD
Display

#

# Note the dht_sensor_type below may need to be changed depending on which DHT sensor you
have:

# 0 - DHT11 - blue one - comes with the GrovePi+ Starter Kit

# 1 - DHT22 - white one, aka DHT Pro or AM2302

# 2 - DHT21 - black one, aka AM2301

#

from grovepi import *
from grove_rgb_lcd import *
from time import sleep
from math import isnan

dht_sensor_port = 3 # connect the DHT sensor to port 3 on the Raspberry Pi Zero
dht_sensor_type = 0 # use 0 for the blue-colored sensor and 1 for the white-colored sensor

# set green as backlight color

# we need to do it just once

# setting the backlight color once reduces the amount of data transfer over the I2C line
setRGB(0,255,0)
```

while True:

try:

get the temperature and Humidity from the DHT sensor

[temp,hum] = dht(dht_sensor_port,dht_sensor_type)

print("temp =", temp, "C\thumidity =", hum,"%")

check if we have nans

if so, then raise a type error exception

if isnan(temp) is True or isnan(hum) is True:

raise TypeError('nan error')

t = str(temp)

h = str(hum)

instead of inserting a bunch of whitespace, we can just insert a \n

we're ensuring that if we get some strange strings on one line, the 2nd one won't be affected

setText_norefresh("Temp:" + t + "C\n" + "Humidity :" + h + "%")

except (IOError, TypeError) as e:

print(str(e))

and since we got a type error

then reset the LCD's text

setText("")

except KeyboardInterrupt as e:

print(str(e))

since we're exiting the program

it's better to leave the LCD with a blank text

setText("")

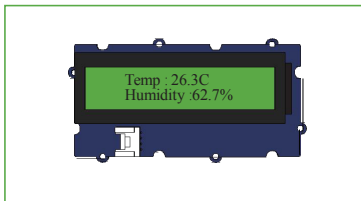
break

wait some time before re-updating the LCD

sleep(0.05)

Outcome

The outcome of the project should show up on your Raspberry Pi and the screen. Below you can see the temperature printed on the Raspberry Pi terminal. Further down, you can see the same information printed on the attached display. The LCD display will start showing the live temperature data and humidity data from the DHT sensor.

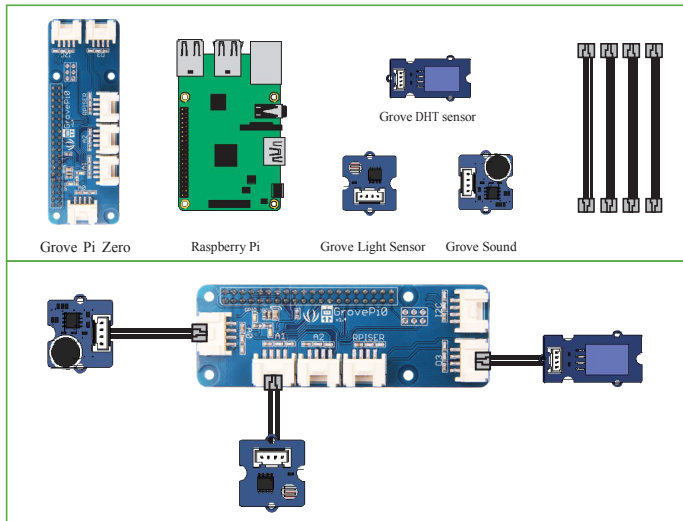


Example Project: Twitter the Weather

In this example we tweet the temperature, light, and sound levels with our Raspberry Pi. The data is taken by the GrovePi and pushed to our Twitter feed with the Raspberry Pi.

Hardware Setup

This example uses a sound sensor on Port A0, a light sensor on Port A1 and a DHT sensor On Port D3.



Running the Program

The example program for this project is shown below. To run the program, download the GrovePi Zero Example Programs from Github(Skip this step if you have downloaded the Example Programs).

```
cd ~/
git clone https://github.com/DexterInd/GrovePi_Zero_Example_Programs.git
```

Change directory on your Raspberry Pi to the
~/GrovePi_Zero_Example_Programs/Projects folder.

```
cd ~/GrovePi_Zero_Example_Programs/Projects/
```

Run the Tweet_the_Weather.py program:

```
sudo python Tweet_the_Weather.py
```

Source Code

The source code for the example can be found here:

https://github.com/DexterInd/GrovePi_Zero_Example_Programs/blob/master/Projects/Tweet_the_Weather.py

```
# Tweet the temperature, light, and sound levels with our Raspberry Pi
# http://www.dexterindustries.com/GrovePi/projects-for-the-raspberry-pi/raspberry-pi-twitter-sensor-feed/

# GrovePi + Sound Sensor + Light Sensor + Temperature Sensor + LED
# http://www.seeedstudio.com/wiki/Grove_-_Sound_Sensor
# http://www.seeedstudio.com/wiki/Grove_-_Light_Sensor
# http://www.seeedstudio.com/wiki/Grove_-_Temperature_and_Humidity_Sensor_Pro
# http://www.seeedstudio.com/wiki/Grove_-_LED_Socket_Kitimport twitter

import twitter
import time
import grovepi
import math

# Connections
sound_sensor = 0    # Sound Sensor on port A0
light_sensor = 1    # Light Sensor on port A1
temperature_sensor = 3 # Temp Sensor on port D3

intro_str = "DI Lab's"

# Connect to Twitter
api = twitter.Api(
    consumer_key='YourKey',
    consumer_secret='YourKey',
    access_token_key='YourKey',
    access_token_secret='YourKey'
)

while True:
```

Error handling in case of problems communicating with the GrovePi
try:

Get value from light sensor

light_intensity = grovepi.analogRead(light_sensor)

Give PWM output to LED

grovepi.analogWrite(led,light_intensity/4)

Get sound level

sound_level = grovepi.analogRead(sound_sensor)

time.sleep(0.5)

Get value from temperature sensor

[t,h]=[0,0]

[t,h] = grovepi.dht(temperature_sensor,0)

Post a tweet

out_str = "%s Temp: %d C, Humidity: %d, Light: %d, Sound: %d"

%(intro_str,t,h,light_intensity/10,sound_level)

print (out_str)

api.PostUpdate(out_str)

except IOError:

print("Error")

except KeyboardInterrupt:

exit()

except Exception as e:

print("Duplicate Tweet or Twitter Refusal: {}".format(e))

time.sleep(60)

Outcome

You should see two outcomes. The first is output on your terminal screen, shown below. You should see “Twitter Connected” if the connection is made successfully. Then a series of Temperature, Light, and Sound values.

```
Twitter Connected
DI Lab's Temp: 24.35, Light: 69, Sound: 0
DI Lab's Temp: 24.35, Light: 67, Sound: 0
DI Lab's Temp: 24.43, Light: 70, Sound: 0
DI Lab's Temp: 24.43, Light: 70, Sound: 0
```

You should also see your Twitter stream start to update. It might take a few minutes for Twitter to update.



FAQ's, Troubleshooting and getting help

Q. What languages are supported by GrovePi+?

We officially support C and Python, but the GrovePi+ runs on many other languages. Other languages with libraries include Go, Java, NodeJS, and Scratch. (<https://github.com/DexterInd/GrovePi/tree/master/Software>).

We love to support more languages. If you are interested in porting the GrovePi+ to other language we'll love to hear more from you.

Q. Which port on GrovePi+ does what? Please check out the port description here:

<http://www.dexterindustries.com/GrovePi/engineering/port-description/>.

Q. How are the ports connected and where is the schematic?

Please check the GrovePi Schematic. GrovePi is open source and you can check out the hardware files here:

<https://github.com/DexterInd/GrovePi/tree/master/Hardware>. The schematic is available in PDF format.

Q. Will my sensor work with the GrovePi+? Please check the supported sensors page here: <http://www.dexterindustries.com/GrovePi/sensors/supported-sensors/> to see if your sensor is officially supported. If it is not, the next question should help you figure out how to get it to work with the GrovePi+.

Q. I just got a Grove sensor I can't get it to work with the GrovePi+. First check if your sensor is officially supported by GrovePi+ here:

<http://www.dexterindustries.com/GrovePi/sensors/supported-sensors/>.

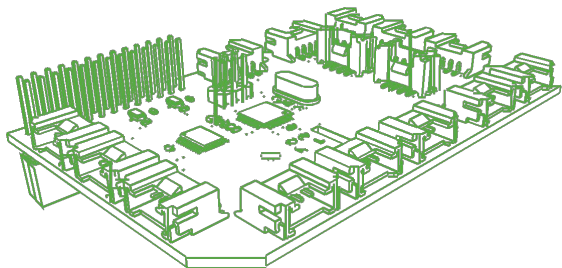
If it is then you might need a software and firmware update, here's the guide:

<http://www.dexterindustries.com/GrovePi/get-started-with-the-grovepi/updating-firmware/>.

Forum

If you have any other problems or suggestions post them on our forum:

<https://forum.dexterindustries.com/c/grovepi>



ATTENTION
Best to keep away
from re.



<http://www.seeed.cc>
<http://www.dexterindustries.com>

For more information, please visit
seeed.cc/grovepi-kit