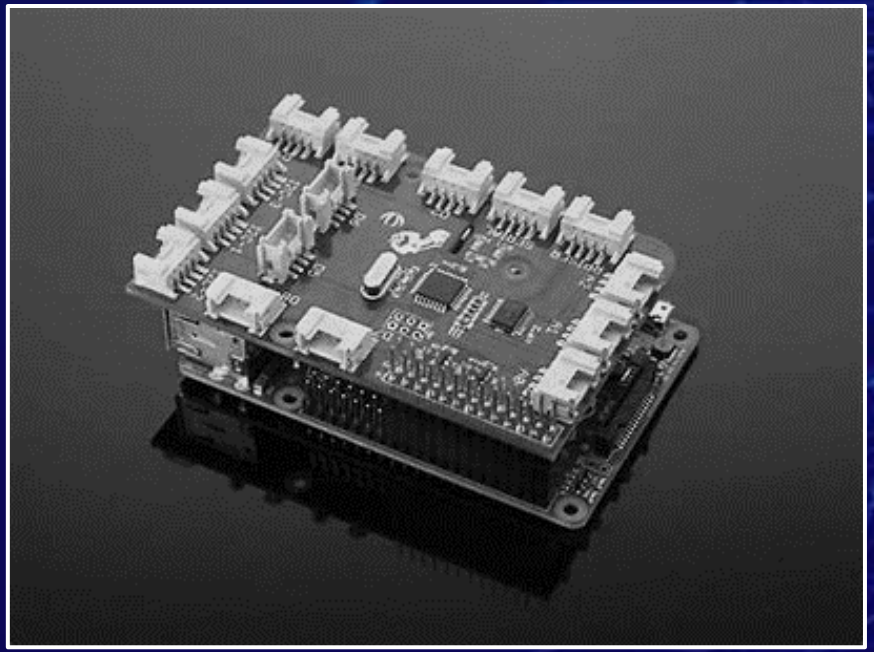


Unit 5

Getting started with Grove Pi+

Disclaimer

The content is curated from online/offline resources and used for educational purpose only

Getting started with Grove Pi+

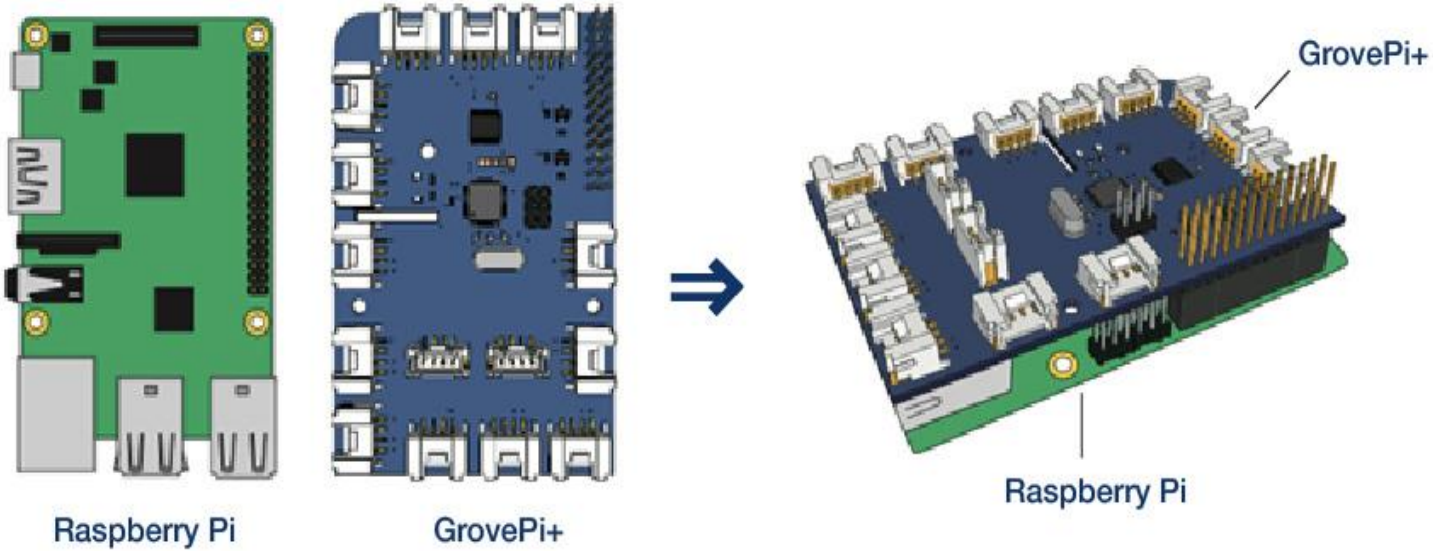
Learning Objectives

- Introduction of Grove Pi+
- Grove Pi+ with Raspberry Pi
- Grove Pi+ Setup
- Different Type of Grove Pi+ Components

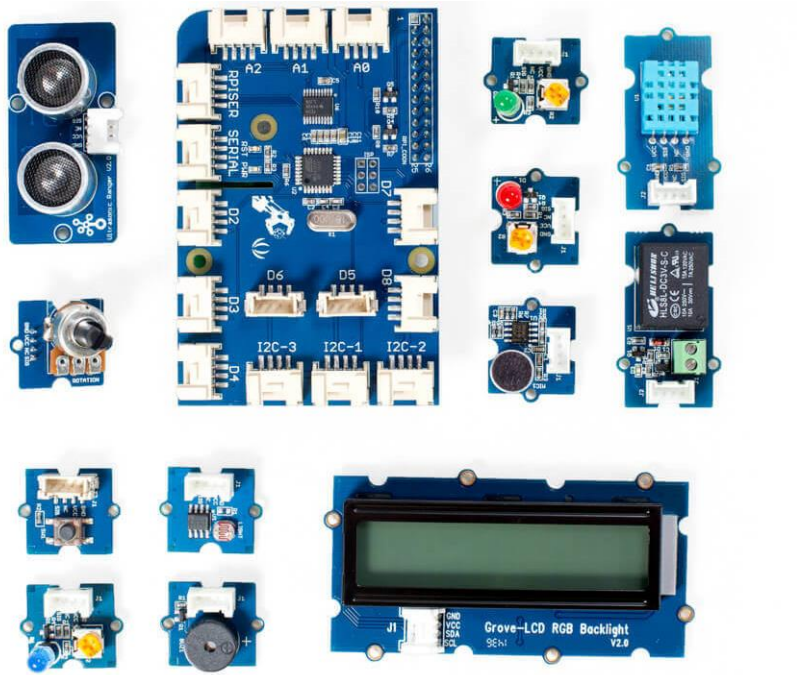
Grove Pi+ Shield

- GrovePi+ is an easy-to-use and modular system for hardware hacking with the Raspberry Pi, no need for soldering or breadboards:
- Plug in your Grove sensors and start programming directly.
- Grove is an easy-to-use collection of more than 100 inexpensive plug-and-play modules that sense and control the physical world.

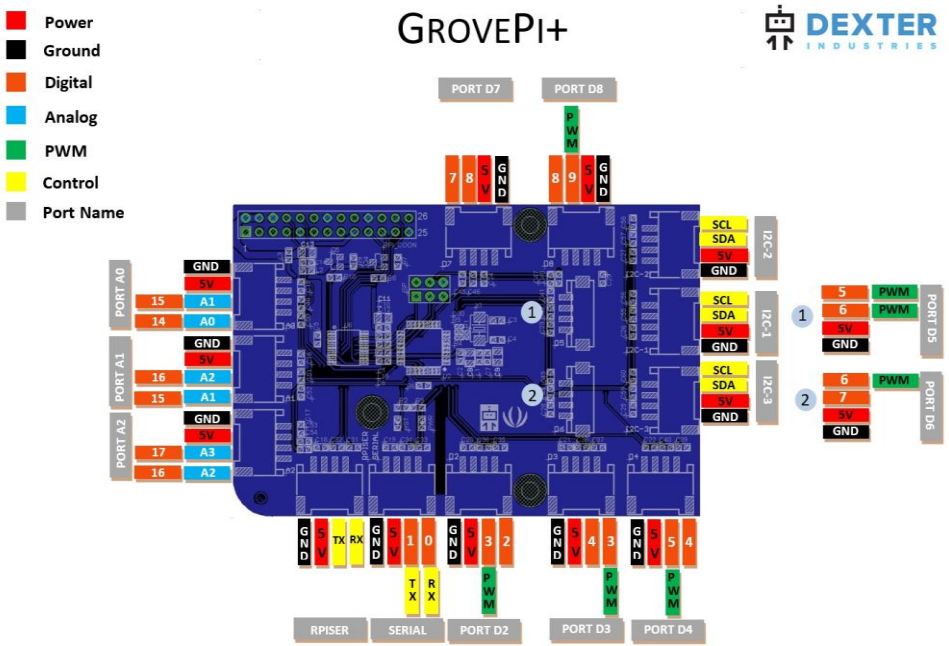
Grove Pi+ Shield with Raspberry Pi



Grove Pi+ kit


A2
A1
A0
RPISER
SERIAL
D2
D3
D4
D6
D5
I2C-3
I2C-1
I2C-2
Grove-LCD RGB Backlight
V2.0
J1

Hardware Ports - GrovePI+



Software Setup - Grove Pi+

1.Clone and install GrovePI repository

- curl -kL dexterindustries.com/update_grovepi | bash

- NOTE : Do the next command only after connecting GrovePi Shield on Raspberry Pi

2. Update Firmware

- cd ~/Dexter/GrovePi/Firmware
- bash firmware_update.sh

3. Running Tests (Optional)

- cd ~/GrovePi/Troubleshooting
- sudo bash all_tests.sh

4. Check version

- python
- >>import grovepi
- >>grovepi.version()

Embedded Code
(Grove Pi Plus)

Lab 1 : interface LED with Grovepi and raspberry

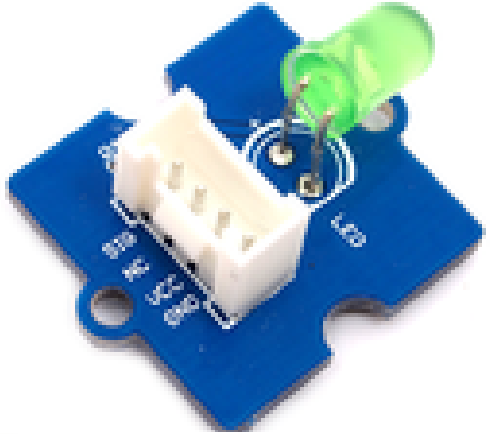
Lab 1 - interface LED with GrovePi and raspberry

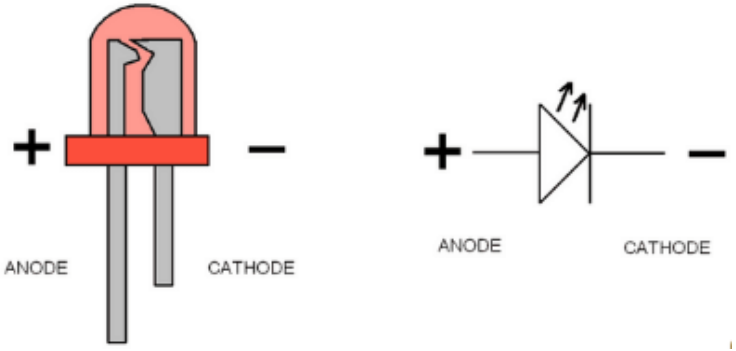
LED – Light Emitting Diode

- A light-emitting diode (LED) is a semiconductor device that emits light when an electric current flows through it.
- When current passes through an LED, the electrons recombine with holes emitting light in the process.
- LEDs allow the current to flow in the forward direction and blocks the current in the reverse direction.

Codes -

1. Automatically blinking the LED
2. Manual Control of LED



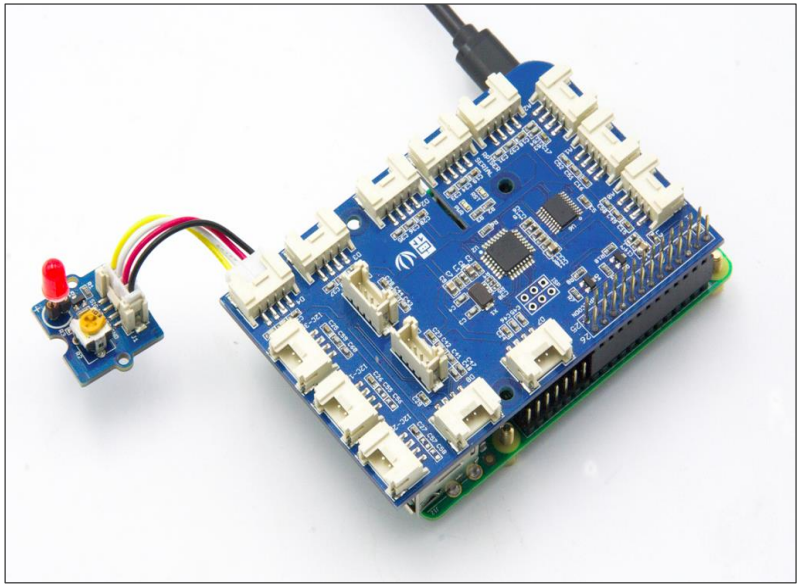


Digital LED

1. Connect Led to D4

2. Git clone the Github repository.
   cd ~
   git clone https://github.com/DexterInd/GrovePi.git

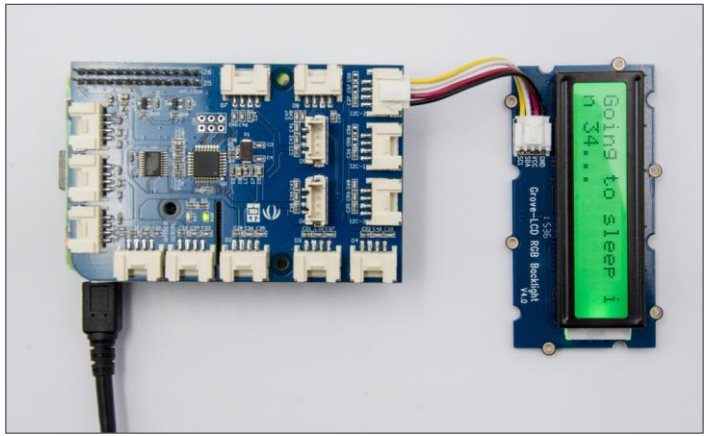
3. Then try:
   cd ~/GrovePi/Software/Python
   python3 grove_led_blink.py



LCD - Liquid Crystal Display

```
from grove_rgb_lcd import *
setRGB(0,255,0) # (Green) RGB Pattern
setText("Hello World")
```

- Connect LCD to any I2C pin to display
- I2C stands for Inter-Integrated Circuit.
- It is a bus interface connection protocol incorporated into devices for serial communication.
- It was originally designed by Philips Semiconductor in 1982.
- Recently, it is a widely used protocol for short-distance communication.

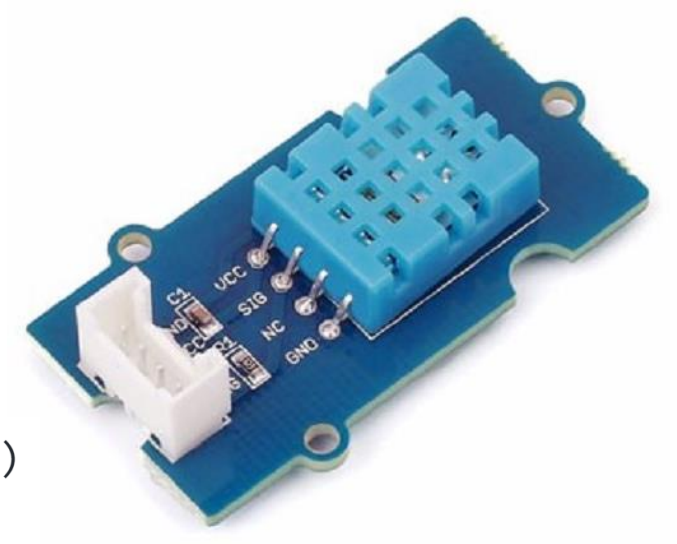


Grove Button

- Grove - Button is a momentary push button. It contains one independent "momentary on/off" 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.

DHT 11 Module

- This Temperature & Humidity sensor provides a pre-calibrated digital output.
- A unique capacitive sensor element measures relative humidity and the temperature is measured by a negative temperature coefficient (NTC) thermistor.
- It has excellent reliability and long term stability.
- Please note that this sensor will not work for temperatures below 0 degree.



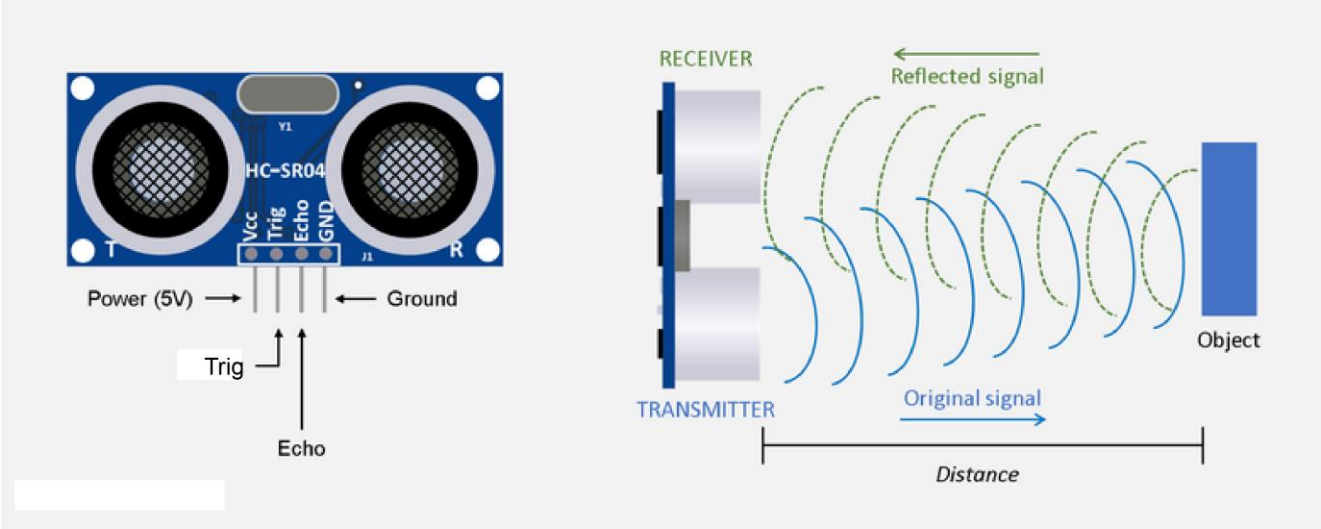

```
[ t,h ] = dht(dht_sensor_port,dht_sensor_type)
print(f"Temp:{t} C Humidity:{h}%")
```

DHT 11 vs DHT22

DHT11	DHT22
$5	$9.95
3 to 5V power and I/O	3 to 5V power and I/O
2.5mA max current use during conversion (while requesting data)	2.5mA max current use during conversion (while requesting data)
Good for 20-80% humidity readings with 5% accuracy	Good for 0-100% humidity readings with 2-5% accuracy
Good for 0-50°C temperature readings ±2°C accuracy	Good for -40 to 80°C temperature readings ±0.5°C accuracy
No more than 1 Hz sampling rate (once every second)	No more than 0.5 Hz sampling rate (once every 2 seconds)
Body size 15.5mm x 12mm x 5.5mm	Body size 15.1mm x 25mm x 7.7mm
4 pins with 0.1" spacing	4 pins with 0.1" spacing

```
dht_sensor_type = 0 # 0 for DHT11 and 1 for DHT22
```

Ultrasonic Sensor

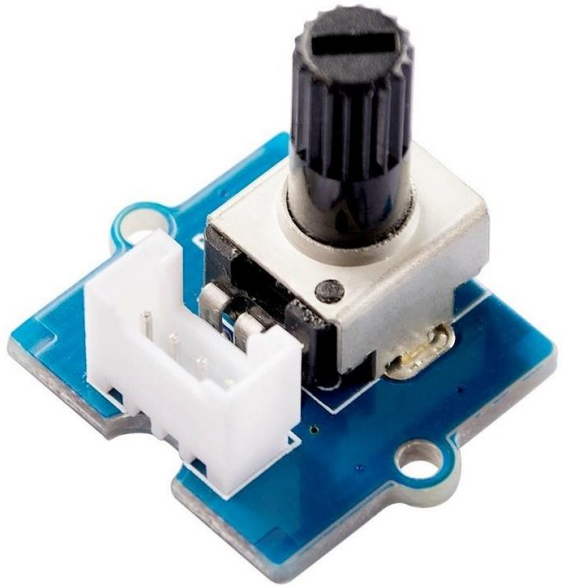



```
distant = ultrasonicRead(ultrasonic_ranger)
print(distant,'cm')
```

Potentiometer – Rotary Angle Sensor

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

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



Do not stop here…

Thank you...!