https://www.halvorsen.blog

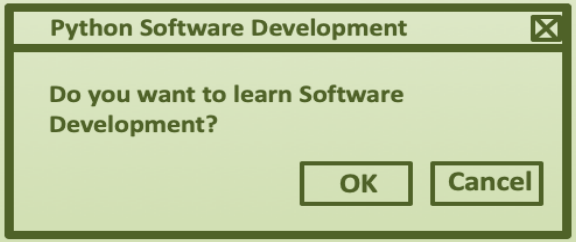
Raspberry Pi and CircuitPython

Hans-Petter Halvorsen

Free Textbook with lots of Practical Examples

Python for Software Development

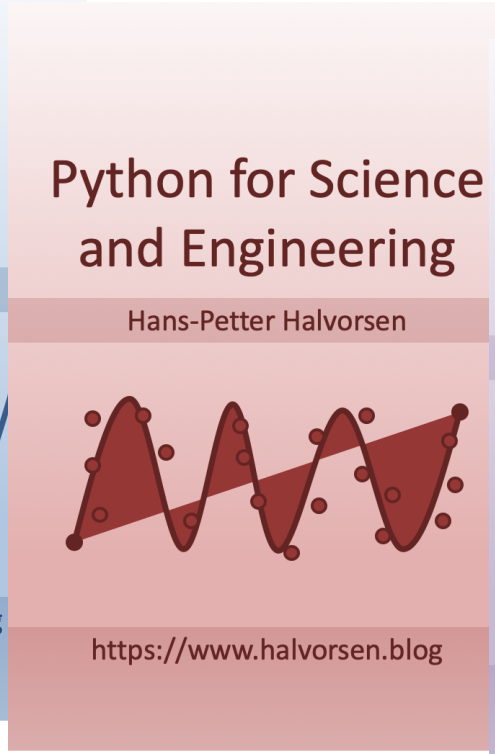
Hans-Petter Halvorsen

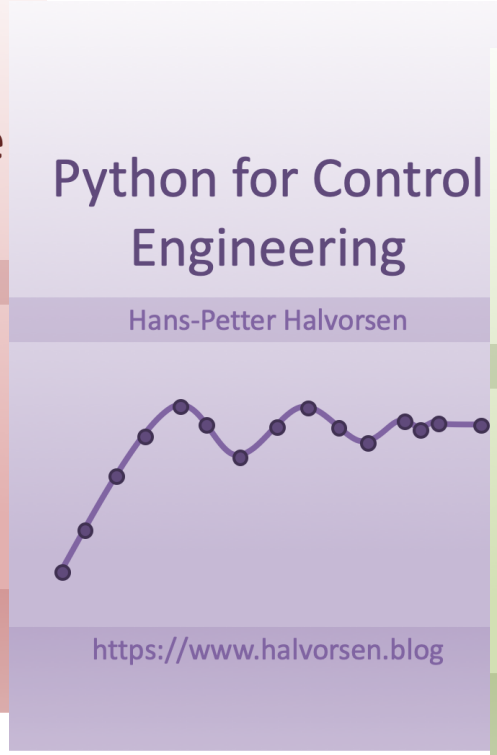


https://www.halvorsen.blog

https://www.halvorsen.blog/documents/programming/python/

Additional Python Resources





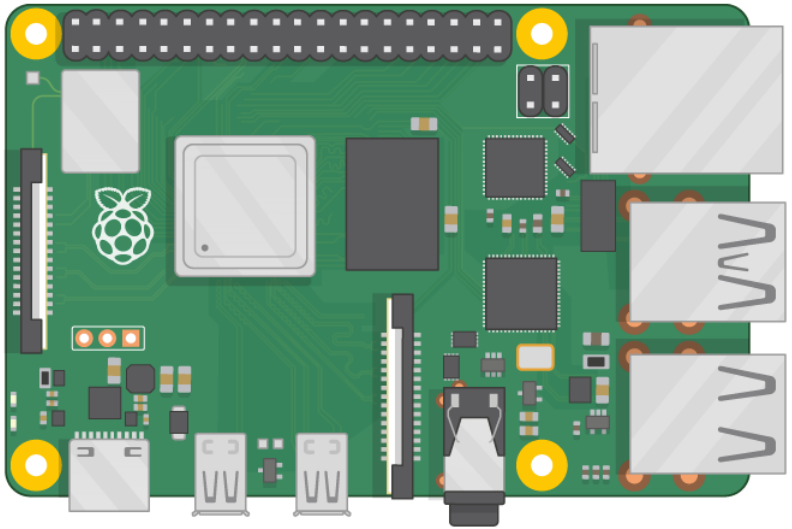
https://www.halvorsen.blog/documents/programming/python/

Contents

- Raspberry Pi
- Raspberry PI GPIO
- CircuitPython and Adafruit-Blinka
- Python Examples:
  - LED
  - Button + LED
  - BME280
  - DTH11/DTH22

Raspberry Pi

Raspberry Pi is a tiny (about 9x6cm), low-cost ($35+), single-board computer that supports embedded Linux operating systems



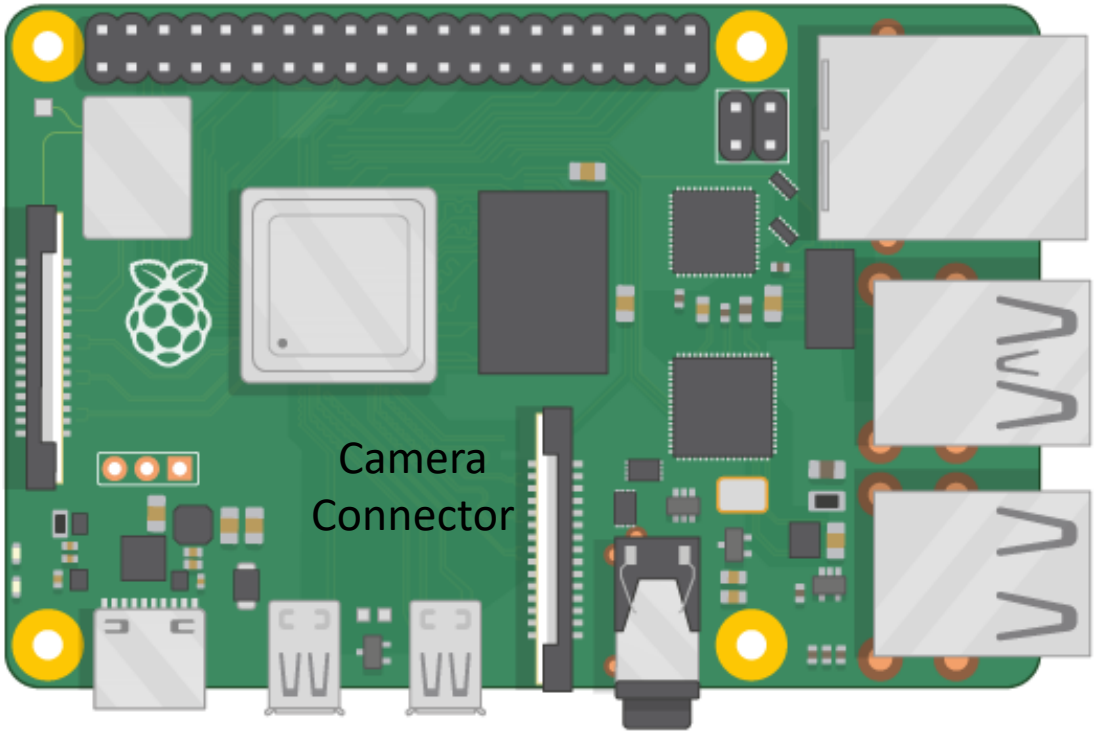
The recommended Operating System is called Raspberry Pi OS (Linux based)

https://www.raspberrypi.org

Raspberry Pi

GPIO Pins

SD Card
(the Back)



Ethernet

Camera
Connector

USB A x 4

Power Supply (USB C) micro HDMI x 2

What Do you Need?

- Raspberry Pi
- microSD Card (+ Adapter)
- Power Supply
- microHDMI to HDMI Cable
- Monitor
- Mouse
- Keyboard

Raspberry Pi OS

- In order make your Raspberry Pi up and running you need to install an Operating System (OS)
- The OS for Raspberry Pi is called “Raspberry Pi OS“ (previously known as Raspbian)
- Raspberry Pi runs a version of an operating system called Linux (Windows and macOS are other operating systems).
- To install the necessary OS, you need a microSD card
- Then you use the “Raspberry Pi Imager“ in order to download the OS to the microSD card.

https://www.raspberrypi.org/software/

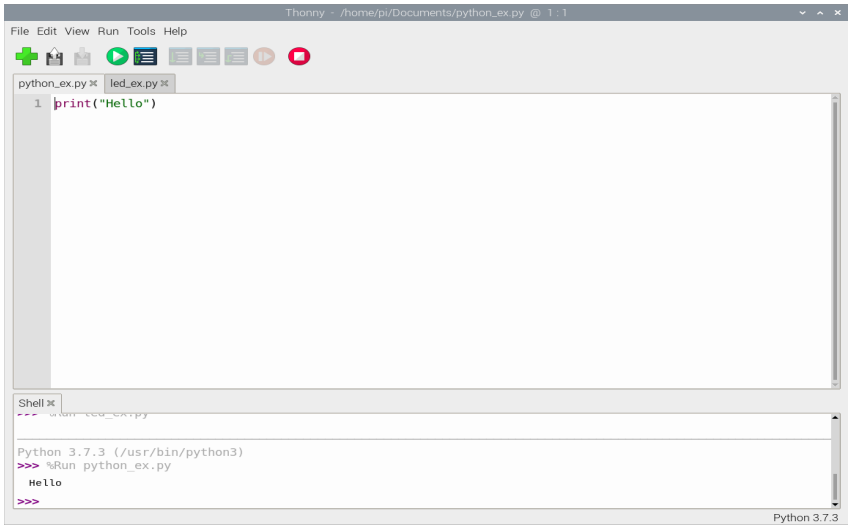
Start using Raspberry Pi

Raspberry Pi OS

- Put the microSD card into the Raspberry Pi
- Connect Monitor, Mouse and Keyboard
- Connect Power Supply
- Follow the Instructions on Screen to setup Wi-Fi, etc.

Python on Raspberry Pi

- The Raspberry Pi OS comes with a basic Python Editor called ”Thonny“



You can install and use others if you want

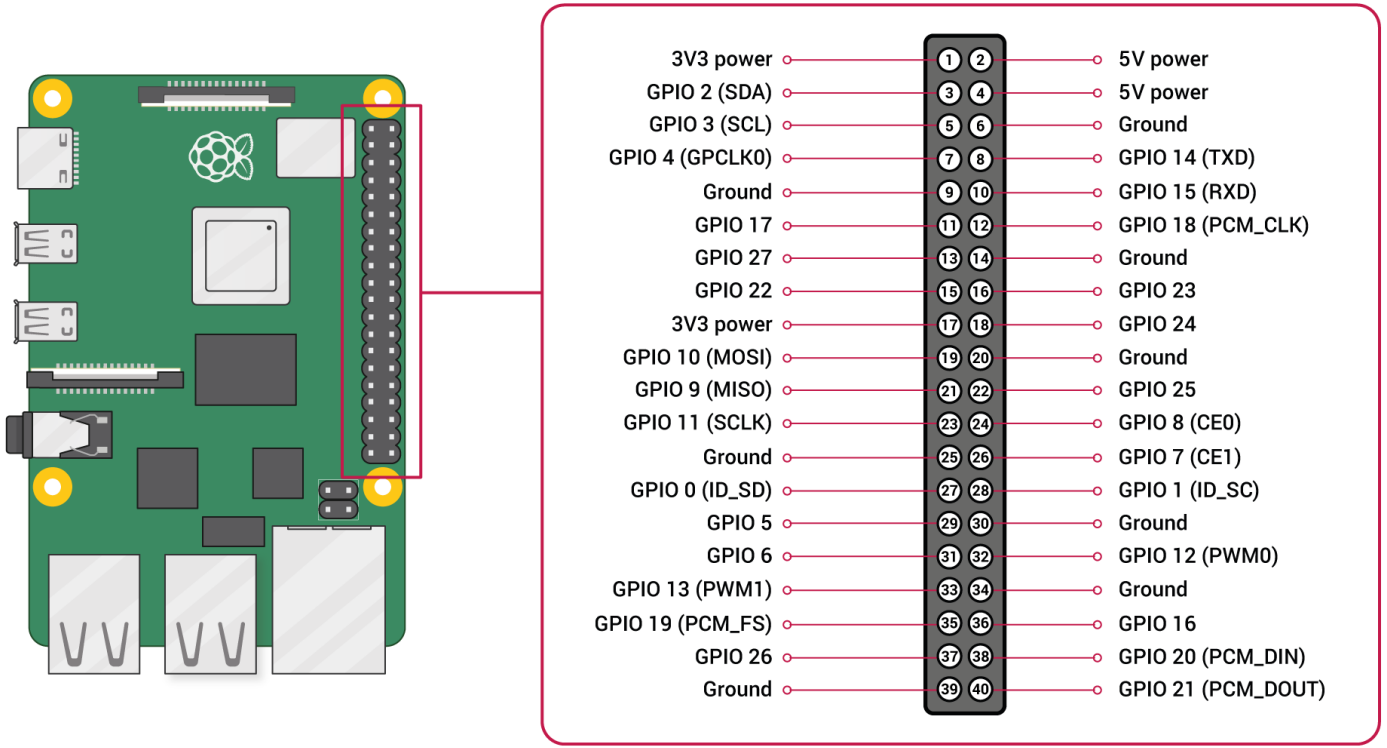
https://www.raspberrypi.org/documentation/usage/python/

https://www.halvorsen.blog

Raspberry PI GPIO

Hans-Petter Halvorsen

GPIO



A powerful feature of the Raspberry Pi is the GPIO (general-purpose input/output) pins. The Raspberry Pi has a 40-pin GPIO header as seen in the image

GPIO Features

The GPIO pins are Digital Pins which are either True (+3.3V) or False (0V). These can be used to turn on/off LEDs, etc.

The Digital Pins can be either Output or Input.

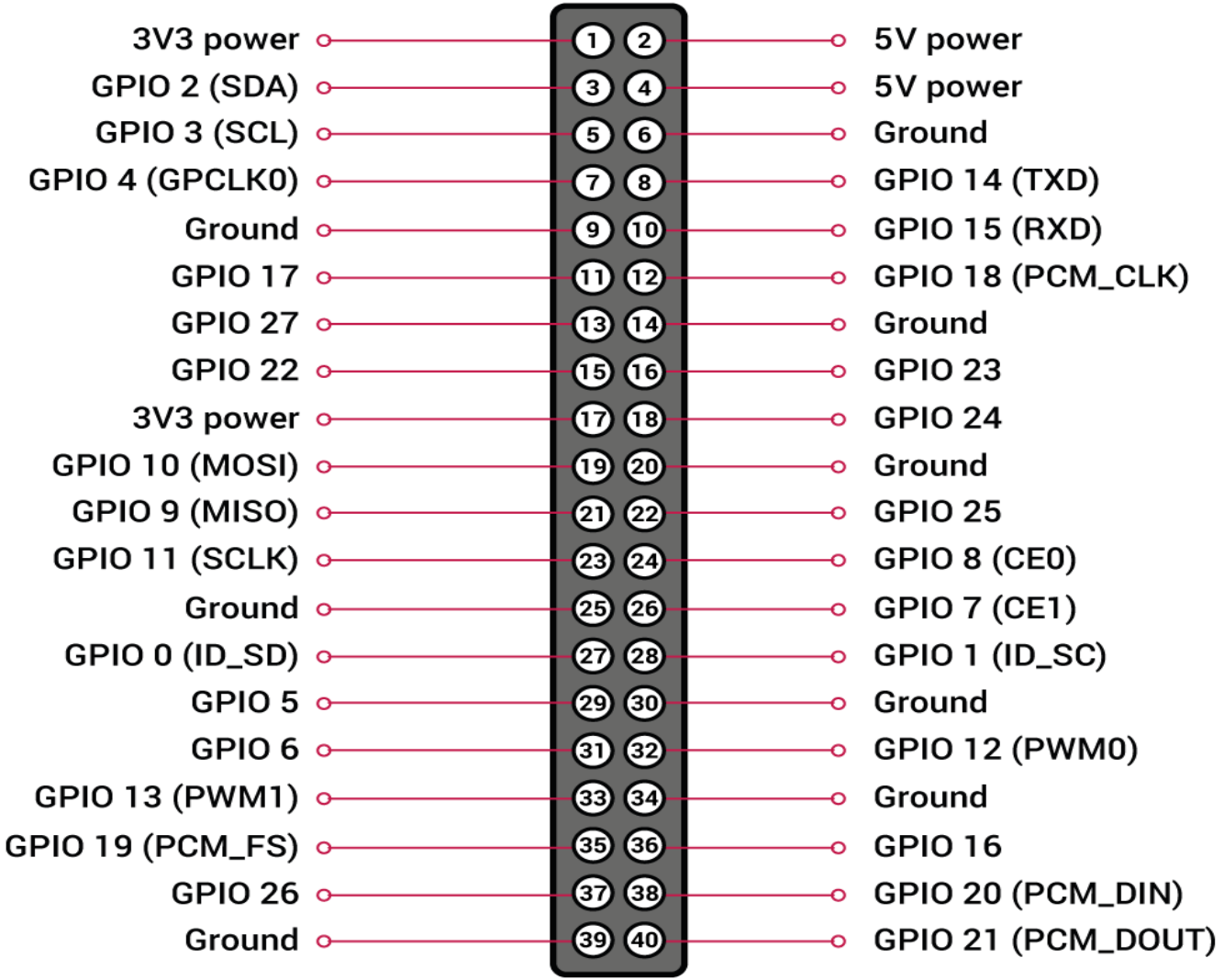
In addition, some of the pins also offer some other Features:

- PWM (Pulse Width Modulation)

Digital Buses (for reading data from Sensors, etc.):

- SPI
- I2C

GPIO


3V3 power
GPIO 2 (SDA)
GPIO 3 (SCL)
GPIO 4 (GPCLK0)
Ground
GPIO 17
GPIO 27
GPIO 22
3V3 power
GPIO 10 (MOSI)
GPIO 9 (MISO)
GPIO 11 (SCLK)
Ground
GPIO 0 (ID_SD)
GPIO 5
GPIO 6
GPIO 13 (PWM1)
GPIO 19 (PCM_FS)
GPIO 26
Ground
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
17 18
19 20
21 22
23 24
25 26
27 28
29 30
31 32
33 34
35 36
37 38
39 40
5V power
5V power
Ground
GPIO 14 (TXD)
GPIO 15 (RXD)
GPIO 18 (PCM_CLK)
Ground
GPIO 23
GPIO 24
Ground
GPIO 25
GPIO 8 (CE0)
GPIO 7 (CE1)
GPIO 1 (ID_SC)
Ground
GPIO 12 (PWM0)
Ground
GPIO 16
GPIO 20 (PCM_DIN)
GPIO 21 (PCM_DOUT)

https://www.halvorsen.blog

CircuitPython and Adafruit-Blinka

Hans-Petter Halvorsen

CircuitPython and Adafruit-Blinka

- CircuitPython adds the Circuit part to the Python part.
- Letting you program in Python and talk to Circuitry like sensors, motors, and LEDs!
- Typically, you would use the Python GPIO Zero Library, but it does not work with SPI/I2C Sensors
- On Raspberry Pi we need to install Adafruit-Blinka. This is a CircuitPython API that can be used on Linux devices such as the Raspberry Pi
- Adafruit-Blinka: https://pypi.org/project/Adafruit-Blinka/

https://learn.adafruit.com/circuitpython-on-raspberrypi-linux/

Install Adafruit-Blinka

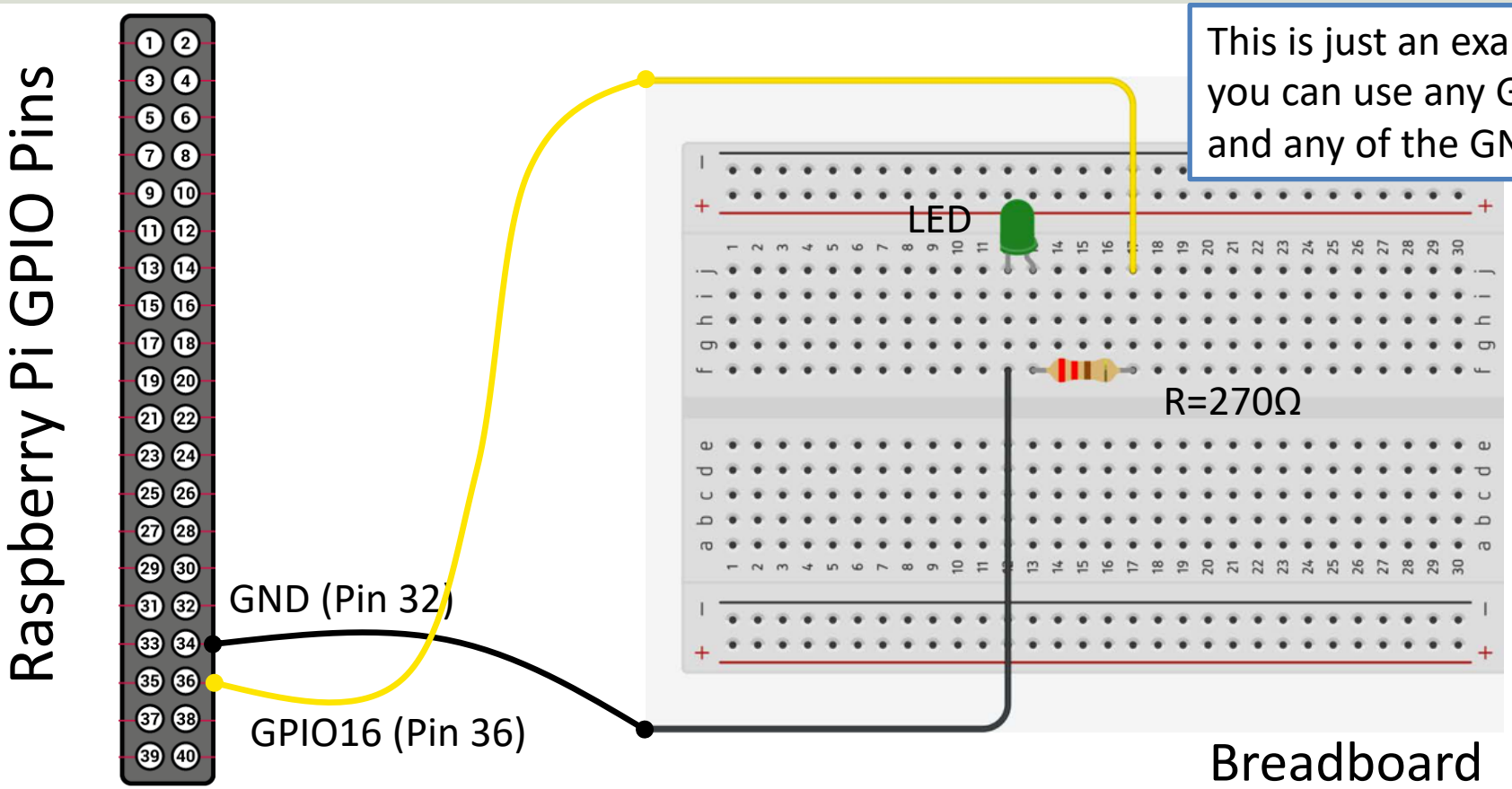
- Adafruit-Blinka: https://pypi.org/project/Adafruit-Blinka/
- Do it from the Thonny Python Editor (Tools -> Manage packages…). Search for “Adafruit-Blinka“
- or use pip:

```
pip3 install Adafruit-Blinka
```

Test of Adafruit-Blinka

```
import board
import digitalio
import busio

print("Hello blinka!")

# Try to great a Digital input
pin = digitalio.DigitalInOut(board.D4)
print("Digital IO ok!")

# Try to create an I2C device
i2c = busio.I2C(board.SCL, board.SDA)
print("I2C ok!")

# Try to create an SPI device
spi = busio.SPI(board.SCLK, board.MOSI, board.MISO)
print("SPI ok!")

print("done!")
```

Blinking LED

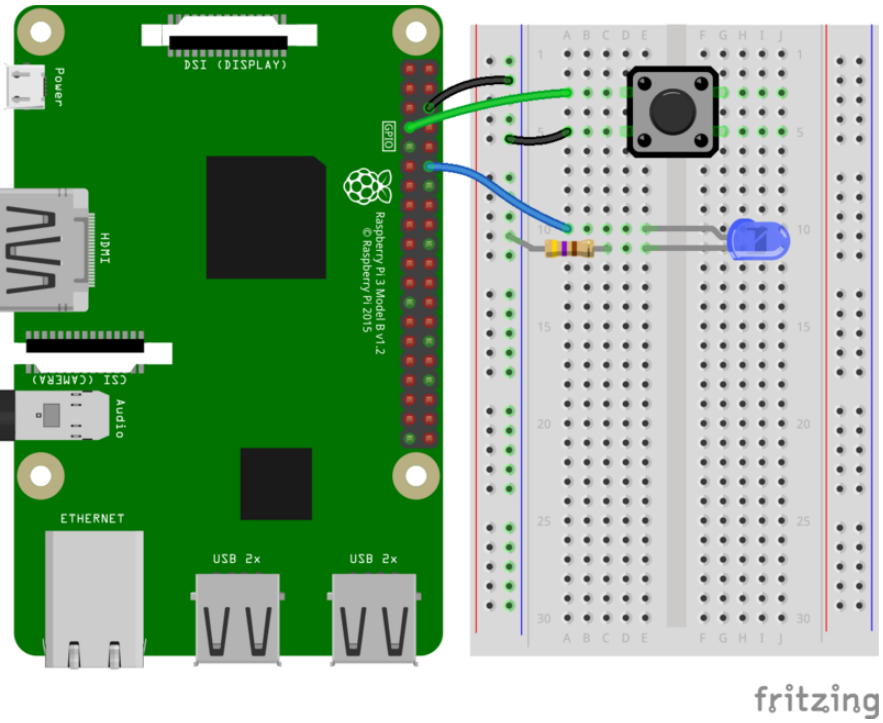


Blinking LED

```
import time
import board
import digitalio

led = digitalio.DigitalInOut(board.D16)
led.direction = digitalio.Direction.OUTPUT

while True:
    led.value = True
    time.sleep(0.5)
    led.value = False
    time.sleep(0.5)
```

https://learn.adafruit.com/circuitpython-on-raspberrypi-linux/digital-i-o

Button + LED




```
import time
import board
import digitalio

print("press the button!")

led = digitalio.DigitalInOut(board.D18)
led.direction = digitalio.Direction.OUTPUT

button = digitalio.DigitalInOut(board.D4)
button.direction =
digitalio.Direction.INPUT
button.pull = digitalio.Pull.UP

while True:
    led.value = not button.value # light
when button is pressed!
```

https://learn.adafruit.com/circuitpython-on-raspberrypi-linux/digital-i-o

https://www.halvorsen.blog

BME280

Bosch BME280 Temperature, Humidity and Barometric Pressure Sensor

Hans-Petter Halvorsen

BME280

- BME280 is a Digital Humidity, Pressure and Temperature Sensor from Bosch
- The sensor provides both SPI and I2C interfaces
- Adafruit, Grove Seeed, SparkFun, etc. have breakout board bords for easy connection to Arduino, Raspberry Pi, etc.
- The Price for these breakout boards are $1-20 depending on where you buy these (ebay, Adafruit, Sparkfun, …)

BME280

- Humidity ±3% accuracy
- Barometric pressure ±1 hPa absolute accuraccy
- Temperature ±1.0°C accuracy

Datasheet:
https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/

BME280

Adafruit

SparkFun

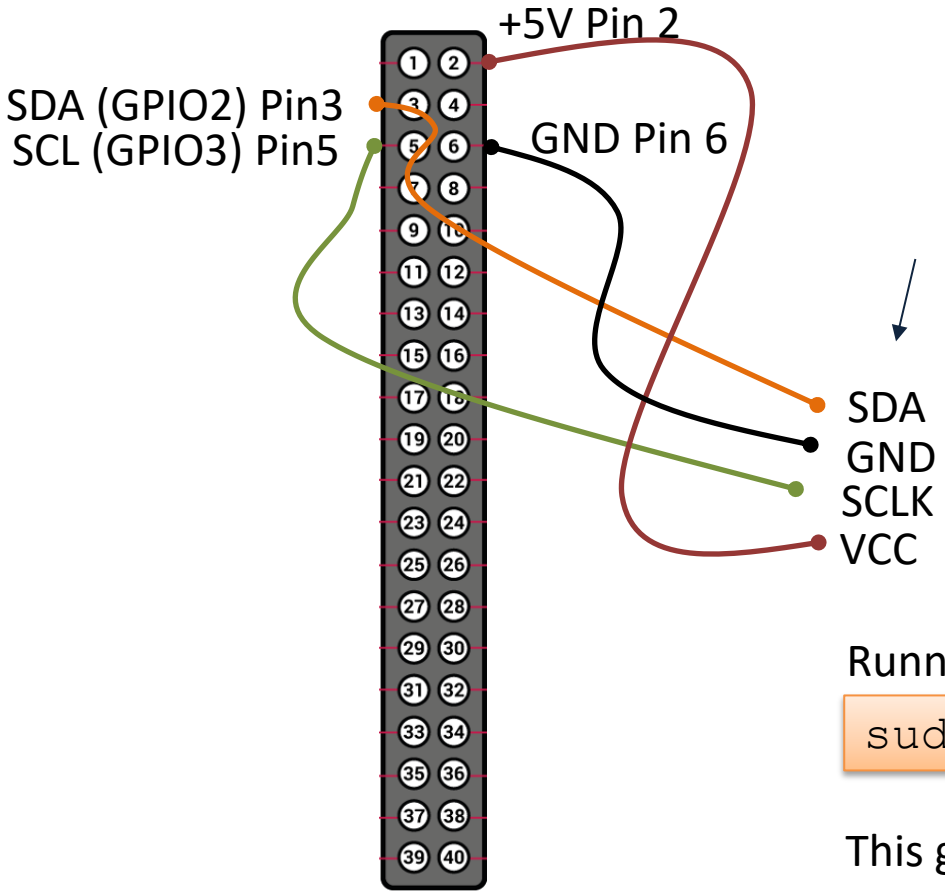
The size is about 2.5x2.5mm

So, to connect it to Raspberry Pi, you typically will use a breakout board

Grove Seeed

BME280 Wiring

Raspberry Pi GPIO Pins



SDA - Serial Data – Bidirectional
SCLK - Serial Clock Input
VDD – Power Supply Input
GND – Ground
NC - Not in use (Not Connected)

Running the following in the Terminal:

```
sudo i2cdetect -y 1
```

This gives the TC74 address `0x76`

BME280 Python

- Install the CircuitPython BME280 Library
- Do it from the Thonny Python Editor (Tools -> Manage packages…). Search for “adafruit-circuitpython-bme280“
- or use pip:

```
pip3 install adafruit-circuitpython-bme280
```

BME280 Python Example

https://circuitpython.readthedocs.io/projects/bme280/en/latest/

```
import time
import board
import busio
import adafruit_bme280

# Create library object using our Bus I2C port
i2c = busio.I2C(board.SCL, board.SDA)
bme280 = adafruit_bme280.Adafruit_BME280_I2C(i2c)

# OR create library object using our Bus SPI port
# spi = busio.SPI(board.SCK, board.MOSI, board.MISO)
# bme_cs = digitalio.DigitalInOut(board.D10)
# bme280 = adafruit_bme280.Adafruit_BME280_SPI(spi, bme_cs)

# change this to match the location's pressure (hPa) at sea level
bme280.sea_level_pressure = 1013.25

while True:
    print("\nTemperature: %0.1f C" % bme280.temperature)
    print("Humidity: %0.1f %%" % bme280.relative_humidity)
    print("Pressure: %0.1f hPa" % bme280.pressure)
    print("Altitude = %0.2f meters" % bme280.altitude)
    time.sleep(2)
```

DHT11/DHT22

Temperature and Humidity Sensors

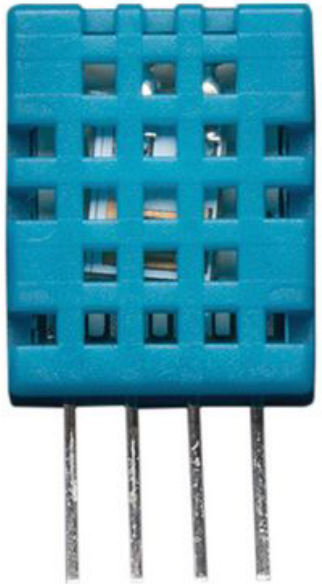
Hans-Petter Halvorsen

DHT11/DHT22

They are Breadboard friendly and easy to wire. They use a single-wire to send data.

DHT11

- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50°C temperature readings ±2°C accuracy
- 1 Hz sampling rate (once every second)
- Price: a few bucks



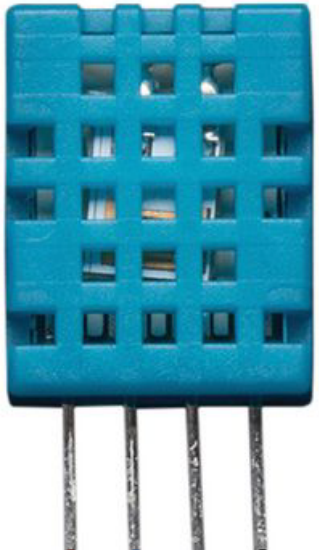
DHT22

DHT22 is more precise, more accurate and works in a bigger range of temperature and humidity, but its larger and more expensive

- 0-100% RH
- -40-125°C

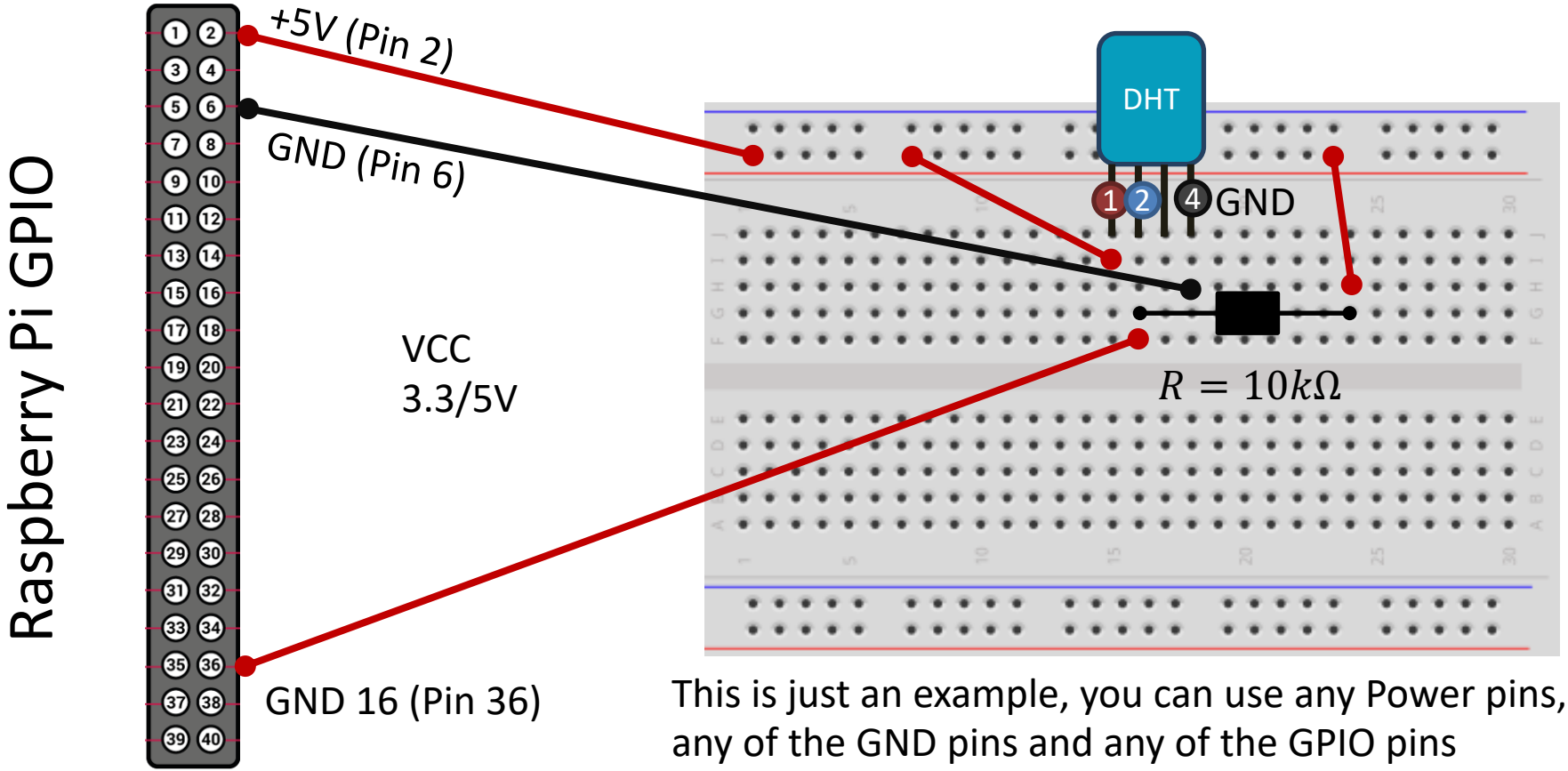
Typically you need a 4.7K or 10K resistor, which you will want to use as a pullup from the data pin to VCC. This is included in the package

DHT11/DHT22



Pin 3 is not in use

DHT11/DHT22



This is just an example, you can use any Power pins, any of the GND pins and any of the GPIO pins

DHT11/DHT22 Python

- Install the CircuitPython-DHT Library
- Do it from the Thonny Python Editor (Tools -> Manage packages…). Search for “adafruit-circuitpython-dht“
- or use pip:
  `adafruit-circuitpython-dht`

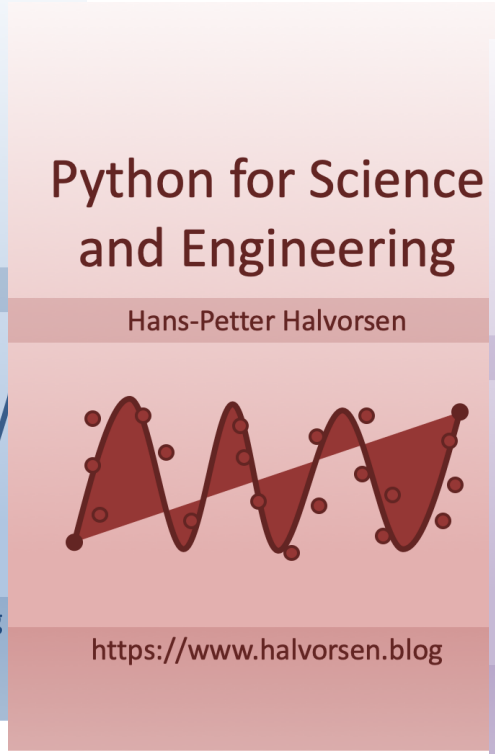
DHT11/DHT22 Python Example

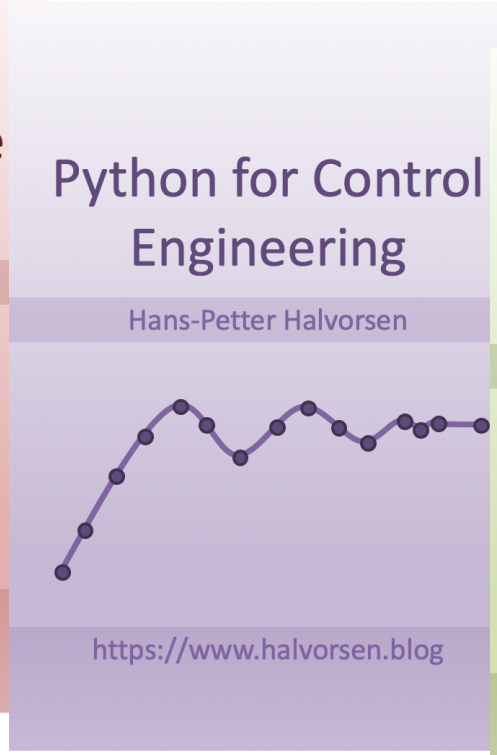
```
import time
import board
import adafruit_dht

dhtDevice = adafruit_dht.DHT22(board.D18, use_pulseio=False)

while True:
    try:
        temperature_c = dhtDevice.temperature
        humidity = dhtDevice.humidity
        print(
            "Temp: {:.1f} C    Humidity: {}% ".format(
            temperature_c, humidity
            )
        )

    except RuntimeError as error:
        # Errors happen fairly often, DHT's are hard to read, just keep going
        print(error.args[0])
        time.sleep(2.0)
        continue
    except Exception as error:
        dhtDevice.exit()
        raise error

    time.sleep(2.0)
```

Errors happen fairly often, DHT's are hard to read because it needs precise timing. That's why you should use **try** in your code

https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/python-setup

Additional Python Resources





https://www.halvorsen.blog/documents/programming/python/

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog