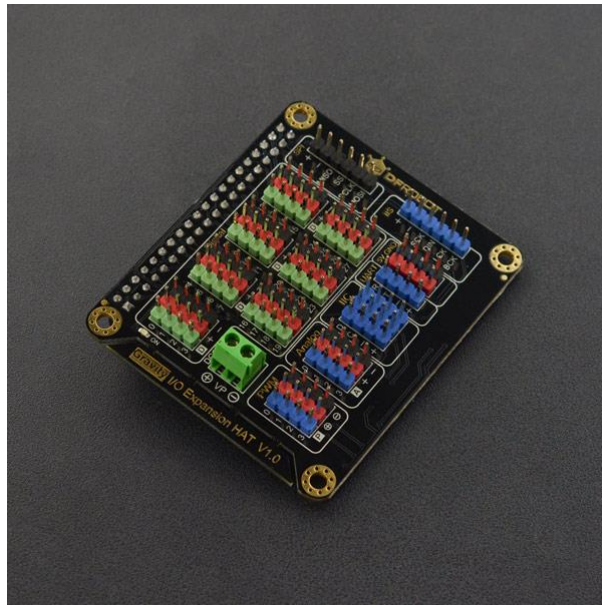


IO Expansion HAT for Raspberry Pi 4B/3B+

SKU:DFR0566



Introduction

This IO Expansion HAT from DFRobot is the perfect companion for your Raspberry Pi 4B/3B+! It leads out all of the IO ports on Raspberry Pi including digital port, analog port, PWM, IIC, UART, SPI, and IIS. Besides that, the HAT is totally compatible with DFRobot Gravity Series which frees users from complicated connection work, and enables them to just concentrate on their projects building.

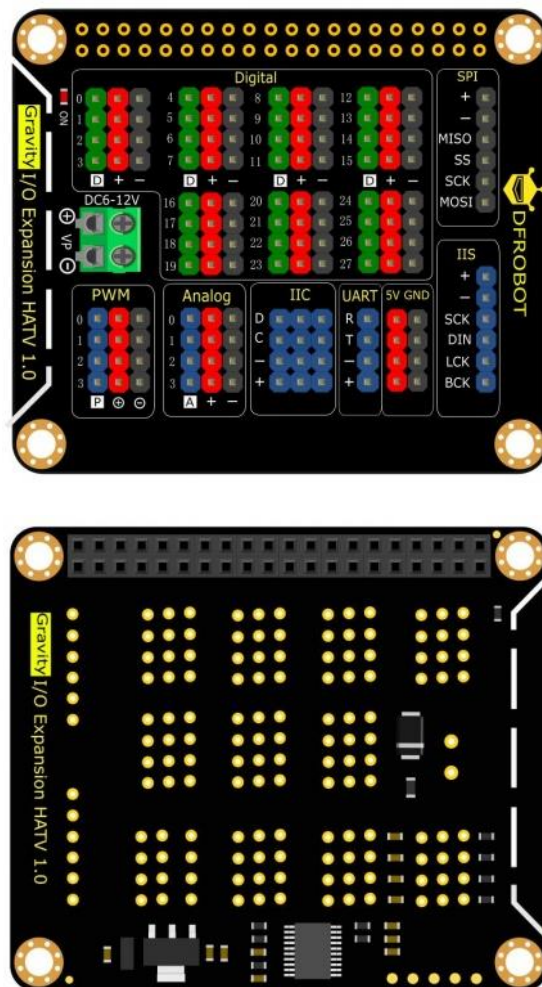
Raspberry Pi GPIO pins work with a maximum logic level of 3.3V. Besides the 3.3V power sensor and module, the product also supports:

- Sensor and module with 5V power supply and 3.3V level
- PWM external power supply (6~12V)
- Controlling multiple servos

Specification

- Driver Maincontroller: STM32
- Operating Voltage: 5V
- PWM External Power Supply: 6-12V
- PWM Pin Voltage: 5V
- Sensor Interface Power Supply: 3.3V
- Communication Interface: 28 digital Ports, 4 groups of analog port, 3 groups of IIC port, 1 group of UART, 4 groups of 5V power interfaces, 1 group of SPI, 1 group of IIS port
- Device Address: 0x10
- Outline Dimension: 65×56mm/2.56×2.20"

Board Overview



Silkscreen	Label	Description
+	+	3.3V Positive
-	-	Negative
⊕	⊕	Binding post connect to external power positive
⊖	⊖	Binding post connect to external power negative
Digital	0-27	Raspberry Pi GPIO0-GPIO27
PWM	0-3	PWM signal output pin 0-3
Analog	0-3	Analog signal input pin 0-3
IIC	C	IIC port clock line
	D	IIC port data line
UART	T	UART Transmit port
	R	UART Receive port
5V	5V	5V positive
GND	GND	Negative
IIS	SCK	IIS serial clock line
	DIN	IIS serial data input

Silkscreen	Label	Description
	LCK	IIS L/R channel selection line
	BCK	IIS system clock line
SPI	MISO	SPI data output line
	SS	SPI enable pin
	SCLK	SPI serial clock line
	MOSI	SPI data input line



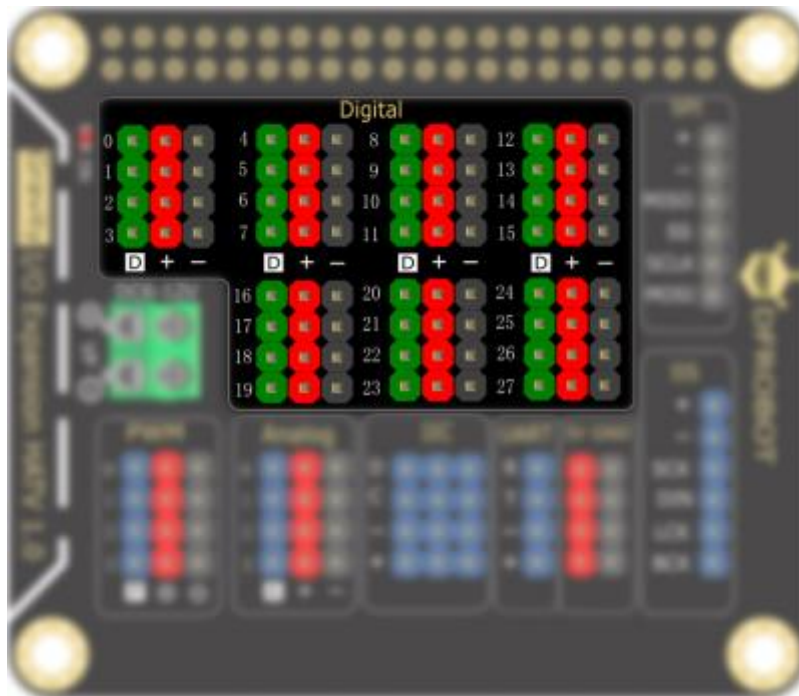
Note:

- The GPIO number in this board adopts BCM codes.
- When the VP port is not power by external power, the voltage of PWM ⊕ is 5V.
- When the VP port is powered by external power, the voltage of PWM ⊕ is equal to that of the VP external power (6-12V).

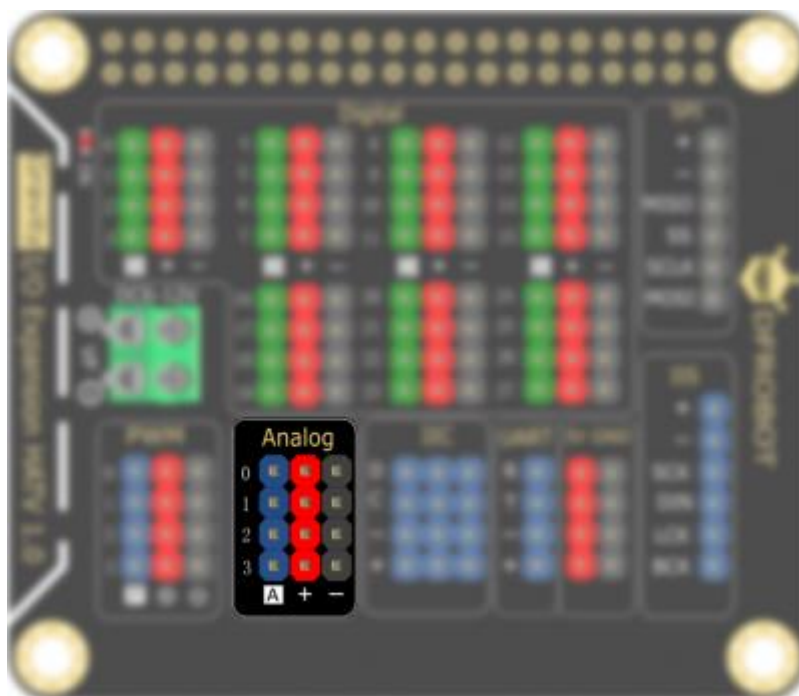
Tutorial

Port and Learning Guide

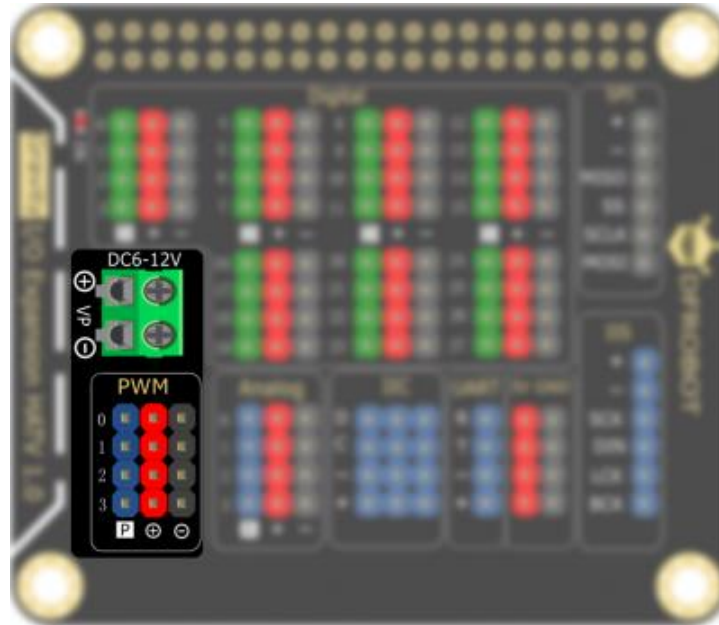
- Digital Port: IO expansion board offers 28 groups (D0-D27) of digital ports that are led out via Raspberry Pi ports GPIO0~GPIO27 (BCM codes). [GPIO Learning Guide](#).



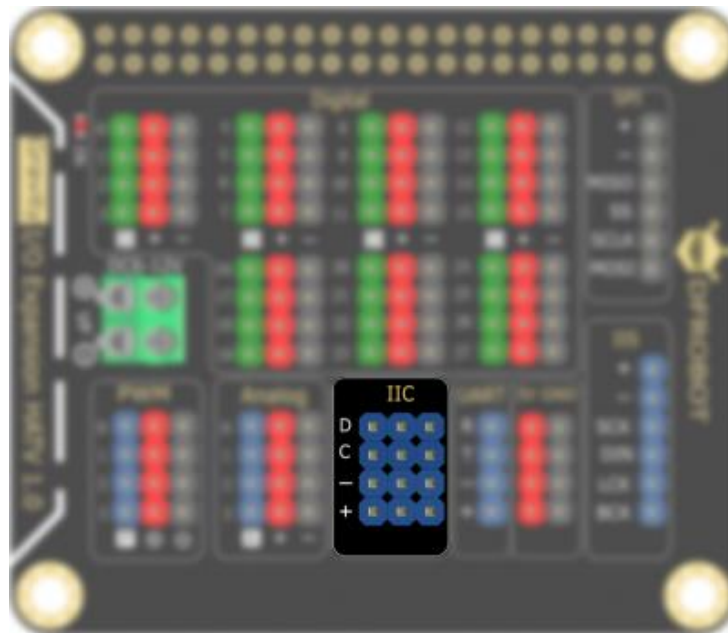
- Analog Port: IO expansion board has four groups of analog ports A0-A3. The board integrates on-board MCU STM32, and 12-bits ADC. The input voltage of analog sensor is 12-bit ADC. After the analog data is converted into digital data, it will be sent to Raspberry Pi via IIC communication. [Download Example Program.](#)



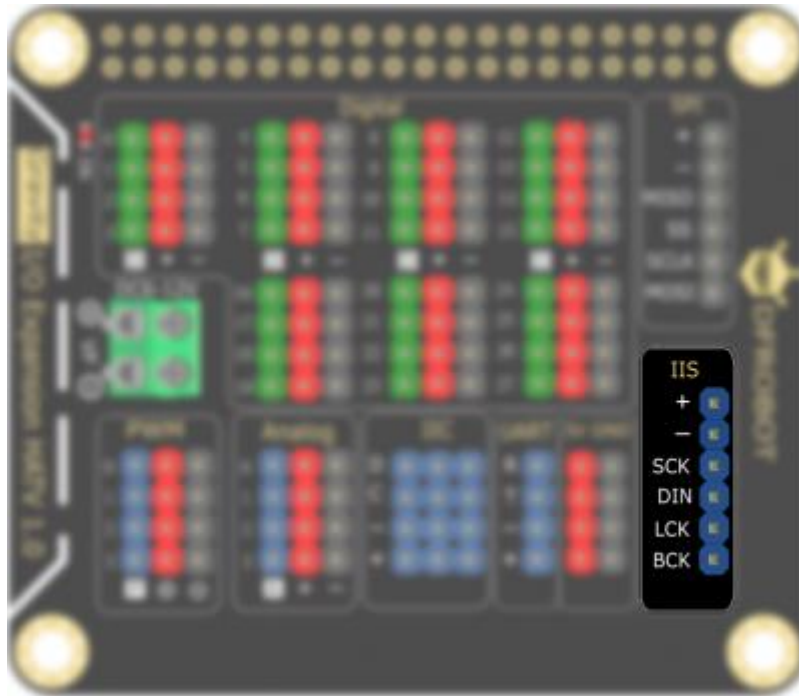
- PWM Port: IO expansion board provides four groups of PWM ports. Connect the STM32 to PWM. Raspberry Pi will send data to STM32 via IIC to control. VP port can supply 6-12V external power to PWM port. When not powered, the voltage of PWM \oplus is 3.3V. [Download Example Program.](#)



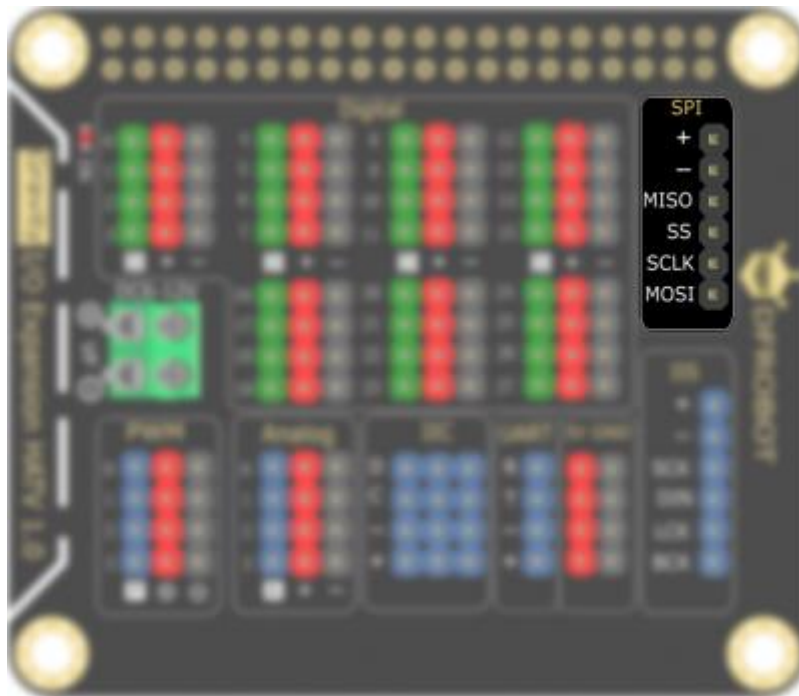
- IIC Port: IO expansion board has 3 groups of IIC ports that are led out via Raspberry Pi GPIO2 (SDA.1) and GPIO3 (SCL.1) (BCM code). [IIC Learning Guide.](#)



- IIS Port: there is 1 group of IIS port that is led out via Raspberry Pi GPIO ports: GPIO21 (SCK), GPIO20(DIN), GPIO19(LCK), GPIO18(BCK).



- SPI Port: IO expansion board leads out a group of SPI ports via its GPIO port (BCM code): GPIO10(MOSI)and GPIO9(MISO), GPIO11(SCLK), GPIO8(SS). [SPI Learning Guide](#).



Example

This example demonstrates how to use the sensors of all ports on Raspberry Pi expansion board.

- **Hardware**
 - Raspberry Pi Board x1
 - IO Expansion HAT for Raspberry Pi x1
 - HDMI Cable x1
 - Display x1
 - Keyboard and Mouse x1

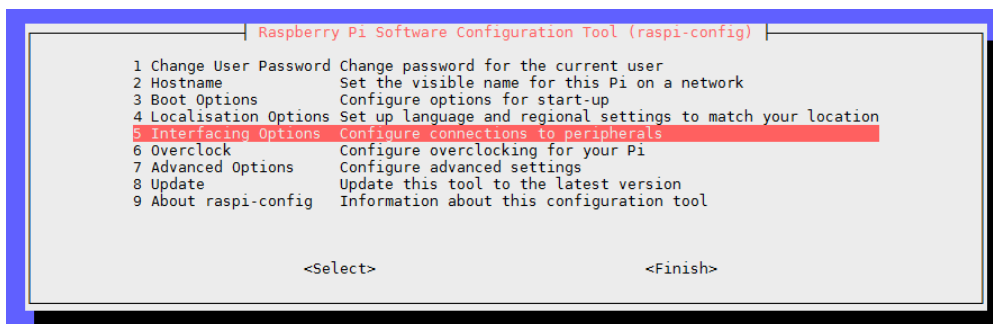
IIC Usage Operation and Program Execution Introduction

- Enable Raspberry Pi I2C interface. (Way to enable SPI is the same with IIC). Skip this step if it is already enabled.

Open Terminal and input the following commands, press "Enter":

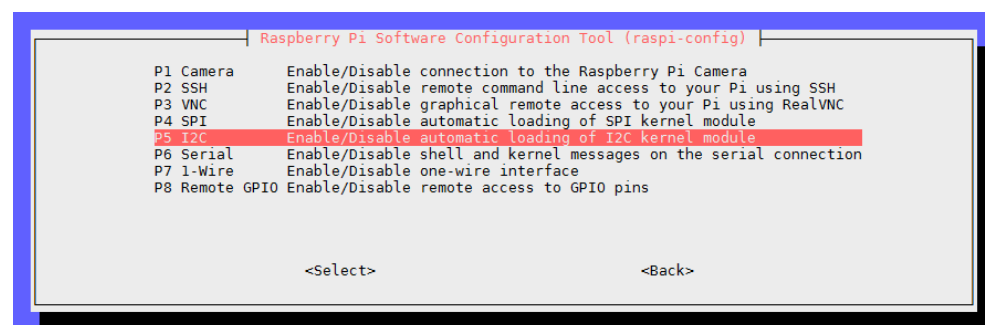
```
sudo raspi-config
```

Use the "Enter" key to select: [Interfacing Options] or ([Advanced Options])->[I2C]->[Yes]->[OK]->[Finish]:



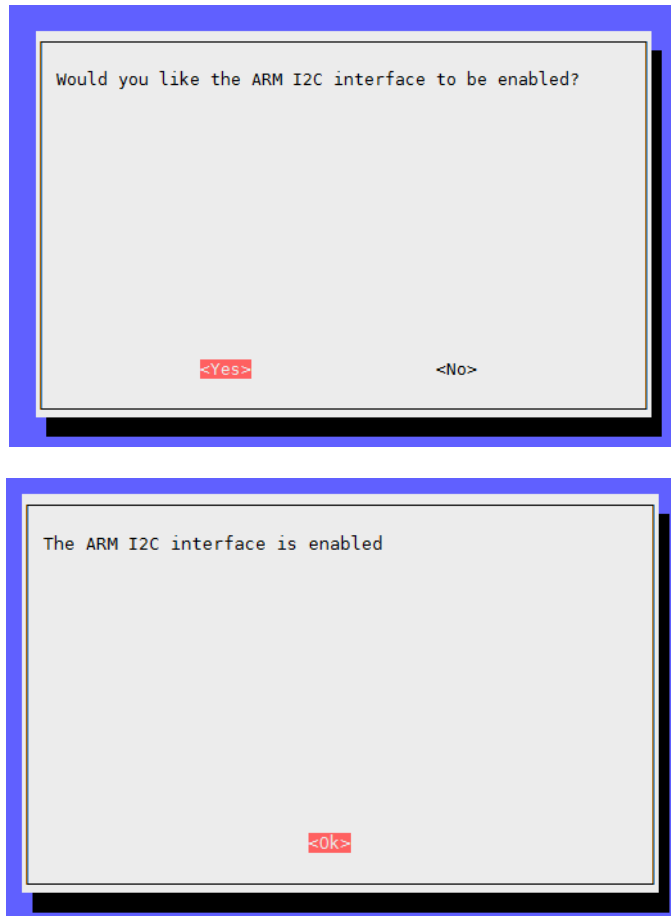
```
Raspberry Pi Software Configuration Tool (raspi-config)
1 Change User Password  Change password for the current user
2 Hostname              Set the visible name for this Pi on a network
3 Boot Options          Configure options for start-up
4 Localisation Options  Set up language and regional settings to match your location
5 Interfacing Options   Configure connections to peripherals
6 Overclock             Configure overclocking for your Pi
7 Advanced Options      Configure advanced settings
8 Update                Update this tool to the latest version
9 About raspi-config    Information about this configuration tool

<Select>                                <Finish>
```



```
Raspberry Pi Software Configuration Tool (raspi-config)
P1 Camera              Enable/Disable connection to the Raspberry Pi Camera
P2 SSH                 Enable/Disable remote command line access to your Pi using SSH
P3 VNC                 Enable/Disable graphical remote access to your Pi using RealVNC
P4 SPI                 Enable/Disable automatic loading of SPI kernel module
P5 I2C                 Enable/Disable automatic loading of I2C kernel module
P6 Serial              Enable/Disable shell and kernel messages on the serial connection
P7 I-Wire              Enable/Disable one-wire interface
P8 Remote GPIO         Enable/Disable remote access to GPIO pins

<Select>                                <Back>
```

- Install Python demo library and git, and make sure the network connection of Raspberry Pi is fine. Skip this step if these have been installed.

Input the following commands into the terminal:

```
sudo apt-get update
sudo apt-get install build-essential python-dev python-smbus git
```

- Download the driver library and run.

Input the following commands into the terminal:

```
cd ~
git clone https://github.com/DFRobot/DFRobot_RaspberryPi_Expansion_Board.git
```

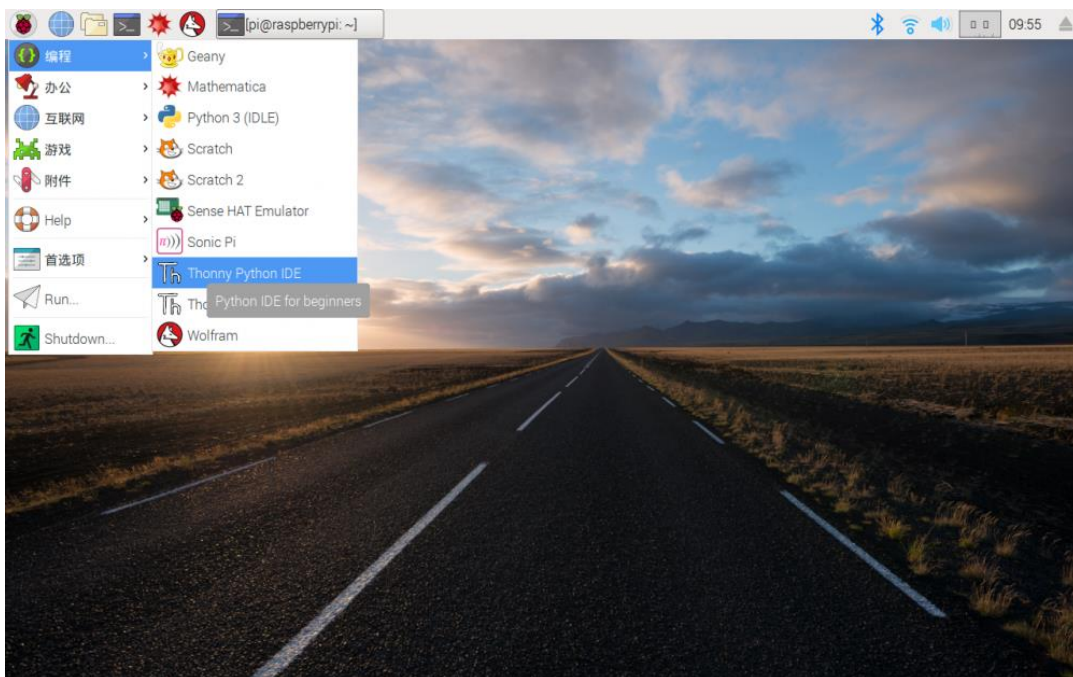
- Two ways to run the program after the library is installed:

1. Input command:

```
2. cd DFRobot_RaspberryPi_Expansion_Board/raspberry
```

```
python demo_adc.py
```

3. Open Thonny Python IDE under Raspberry Pi system to check the downloaded file.



IIC Usage Example

IO expansion board has 3 groups of IIC ports that are led out via Raspberry Pi GPIO2 (SDA.1) and GPIO3 (SCL.1) (BCM code).

Take SEN0303 VEML6075 UV sensor as a example, click to download

[DFRobot_VEML6075 Library file]

(https://github.com/DFRobot/DFRobot_VEML6075 "DFRobot_VEML6075 Library file"), or input the following command into the terminal:

```
cd ~  
git clone https://github.com/DFRobot/DFRobot_VEML6075.git
```

Run the DFRobot_VEML6075_demo.

```
# -*- coding: utf-8 -*-
'''
* file DFRobot_VEML6075_demo.py
* normal test for VEML6075
* UVA index, UVB index and UV index will print on terminal
*
* Copyright [DFRobot](http://www.dfrobot.com), 2018
* Copyright GNU Lesser General Public License
* version V1.0
* date 2018-12-18
'''

import time
import sys
sys.path.append("..")

from DFRobot_VEML6075 import DFRobot_VEML6075

if __name__ == '__main__':

    VEML6075 = DFRobot_VEML6075(1, 0x10) # use i2c bus 1, module address is 0x10

    while VEML6075.begin() != True:
        print("VEML6075 begin faild")
        time.sleep(2)
    print("VEML6075 begin succeed")

    while True:
        Uva = VEML6075.getUva() # get UVA
        Uvb = VEML6075.getUvb() # get UVB
        Uvi = VEML6075.getUvi(Uva, Uvb) # get UVI
        print("")
        print("=====start print=====")
        print("UVA: %.2f" %(Uva))
        print("UVB: %.2f" %(Uvb))
        print("UVA: %.2f" %(Uvi))
        print("mw/cm^2: %.2f" %(VEML6075.Uvi2mwpcm2(Uvi)))
        print("=====end print=====")
        time.sleep(1)
```

Read UV data on Raspberry Pi.

```
pi@raspberrypi:~/DFRobot_VENL6076-master/raspberry/examples $ python DFRobot_VENL6076_demo.py
DFRobot_VENL6076 begin succeed
----- start print -----
UVA: 0.00
UVB: 0.00
UVI: 0.00
uv/cm^2: 0.00
----- end print -----
----- start print -----
UVA: 19.00
UVB: 13.00
UVI: 0.02
uv/cm^2: 44.44
----- end print -----
----- start print -----
UVA: 19.00
UVB: 12.00
UVI: 0.02
uv/cm^2: 41.66
----- end print -----
----- start print -----
UVA: 19.00
UVB: 17.00
UVI: 0.02
uv/cm^2: 43.05
----- end print -----
----- start print -----
UVA: 20.00
UVB: 13.00
UVI: 0.02
uv/cm^2: 45.03
----- end print -----
----- start print -----
UVA: 19.00
UVB: 13.00
UVI: 0.02
```

Servo Controlling Example

Gravity:IO Expansion HAT for Raspberry Pi has 4-way PWM ports to convenient users to use on Raspberry Pi.

Input the following commands to install the library:

```
cd ~
git clone https://github.com/DFRobot/DFRobot_RaspberryPi_Expansion_Board.git
```

Run the demo-servo program, then we can see the servo will rotate from 0 to 180 and then rotate back. Click "ctrl+c" to exit the program execution.

```
# -*- coding:utf-8 -*-
...
# demo_servo.py
#
# Connect board with raspberryPi.
# Run this demo.
#
# Connect servo to one of pwm channels
# All or part servos will move to 0 degree, then move to 180 degree, then loop
# Test Servo: https://www.dfrobot.com/product-255.html
# Warning: Servos must connect to pwm channel, otherwise may destory Pi IO
#
# Copyright [DFRobot](http://www.dfrobot.com), 2016
# Copyright GNU Lesser General Public License
#
# version V1.0
# date 2019-3-28
...
```

```

import time

from DFRobot_RaspberryPi_Expansion_Board import DFRobot_Expansion_Board_IIC as Board
from DFRobot_RaspberryPi_Expansion_Board import DFRobot_Expansion_Board_Servo as Servo

board = Board(1, 0x10)    # Select i2c bus 1, set address to 0x10
servo = Servo(board)

''' print last operate status, users can use this variable to determine the result of
a function call. '''
def print_board_status():
    if board.last_operate_status == board.STA_OK:
        print("board status: everything ok")
    elif board.last_operate_status == board.STA_ERR:
        print("board status: unexpected error")
    elif board.last_operate_status == board.STA_ERR_DEVICE_NOT_DETECTED:
        print("board status: device not detected")
    elif board.last_operate_status == board.STA_ERR_PARAMETER:
        print("board status: parameter error")
    elif board.last_operate_status == board.STA_ERR_SOFT_VERSION:
        print("board status: unsupport board framware version")

if __name__ == "__main__":

    while board.begin() != board.STA_OK:    # Board begin and check board status
        print_board_status()
        print("board begin faild")
        time.sleep(2)
        print("board begin success")

    servo.begin()    # servo control begin

    while True:
        print("servo move to 0")
        servo.move(board.ALL, 0)
        time.sleep(1)
        print("servo move to 180")
        servo.move(board.ALL, 180)
        time.sleep(1)

        print("part servos move to 0")
        servo.move(0, 0) #pwm0
        #servo.move(1, 0) #pwm1
        #servo.move(2, 0) #pwm2
        #servo.move(3, 0) #pwm3
        time.sleep(1)
        print("part servos move to 180")
        servo.move(0, 180) #pwm0
        #servo.move(1, 180) #pwm1
        #servo.move(2, 180) #pwm2
        #servo.move(3, 180) #pwm3
        time.sleep(1)

```

Digital Port Example

- Plug a LED on the Pin27 on the expansion board, then it will begin to blink on and off.

```
import RPi.GPIO as GPIO
import time
import atexit
blinkPin=27
atexit.register(GPIO.cleanup)
GPIO.setmode(GPIO.BCM)
GPIO.setup(blinkPin,GPIO.OUT)
while True:
    GPIO.output(blinkPin,GPIO.HIGH)
    time.sleep(1)
    GPIO.output(blinkPin,GPIO.LOW)
    time.sleep(1)
```

UART Port Example

Gravity: IO Expansion HAT for Raspberry Pi has 1-way UART port. Take SEN0285 Gesture&Touch Sensor as an example, click to download [DFRobot_Gesture_Touch](#) library.

Input the following commands into the terminal:

```
cd ~
git clone https://github.com/DFRobot/DFRobot_Gesture_Touch.git
```

Run the `demo_gesture_touch` program.

Analog Port Example

Gravity : IO Expansion HAT for Raspberry Pi has 4-way analog ports. Input the following commands into the terminal to install the library:

```
cd ~
git clone https://github.com/DFRobot/DFRobot_RaspberryPi_Expansion_Board.git
```

Download and run the `demo_adc` program.

```
# -*- coding:utf-8 -*-
...
# demo_adc.py
#
```



```

# Connect board with raspberryPi.
# Run this demo.
#
# All or part adc channels value will print on terminal
#
# Copyright [DFRobot](http://www.dfrobot.com), 2016
# Copyright GNU Lesser General Public License
#
# version V1.0
# date 2019-3-28
'''

import time

from DFRobot_RaspberryPi_Expansion_Board import DFRobot_Expansion_Board_IIC as Board

board = Board(1, 0x10) # Select i2c bus 1, set address to 0x10

def board_detect():
    l = board.detecte()
    print("Board list conform:")
    print(l)

''' print last operate status, users can use this variable to determine the result of
a function call. '''
def print_board_status():
    if board.last_operate_status == board.STA_OK:
        print("board status: everything ok")
    elif board.last_operate_status == board.STA_ERR:
        print("board status: unexpected error")
    elif board.last_operate_status == board.STA_ERR_DEVICE_NOT_DETECTED:
        print("board status: device not detected")
    elif board.last_operate_status == board.STA_ERR_PARAMETER:
        print("board status: parameter error")
    elif board.last_operate_status == board.STA_ERR_SOFT_VERSION:
        print("board status: unsupport board framware version")

if __name__ == "__main__":

    board_detect() # If you forget address you had set, use this to detected them,
must have class instance

    # Set board controler address, use it carefully, reboot module to make it effective
    '''
    board.set_addr(0x10)
    if board.last_operate_status != board.STA_OK:
        print("set board address faild")
    else:
        print("set board address success")
    '''

    while board.begin() != board.STA_OK: # Board begin and check board status
        print_board_status()
        print("board begin faild")
        time.sleep(2)

```

```

print("board begin success")

board.set_adc_enable()
# board.set_adc_disable()

while True:
    val = board.get_adc_value(board.A0)    # A0 channels read
    #val = board.get_adc_value(board.A1)    # A1 channels read
    #val = board.get_adc_value(board.A2)    # A2 channels read
    #val = board.get_adc_value(board.A3)    # A3 channels read
    print("channel: A0, value: %d" %val)
    print("")

    time.sleep(2)

```

PWM Example

Gravity : IO Expansion HAT for Raspberry Pi offers 4-way PWM ports. Input the following commands into the terminal to install the library:

```

cd ~
git clone https://github.com/DFRobot/DFRobot_RaspberryPi_Expansion_Board.git

```

Download and run the demo_pwm program

```

# -*- coding:utf-8 -*-

...

# demo_pwm.py
#
# Connect board with raspberryPi.
# Run this demo.
#
# All pwm channel will set frequency to 1000HZ, duty to 50%, attention: PWM voltage
depends on independent power supply
# If there is DC motors connect to pwm channle, they will move slow to fast, then
loop
#
# Copyright [DFRobot](http://www.dfrobot.com), 2016
# Copyright GNU Lesser General Public License
#
# version V1.0
# date 2019-3-28
...

import time

from DFRobot_RaspberryPi_Expansion_Board import DFRobot_Expansion_Board_IIC as Board

```

```

board = Board(1, 0x10)    # Select i2c bus 1, set address to 0x10

def board_detect():
    l = board.detecte()
    print("Board list conform:")
    print(l)

''' print last operate status, users can use this variable to determine the result of
a function call. '''
def print_board_status():
    if board.last_operate_status == board.STA_OK:
        print("board status: everything ok")
    elif board.last_operate_status == board.STA_ERR:
        print("board status: unexpected error")
    elif board.last_operate_status == board.STA_ERR_DEVICE_NOT_DETECTED:
        print("board status: device not detected")
    elif board.last_operate_status == board.STA_ERR_PARAMETER:
        print("board status: parameter error")
    elif board.last_operate_status == board.STA_ERR_SOFT_VERSION:
        print("board status: unsupport board framware version")

if __name__ == "__main__":

    board_detect()    # If you forget address you had set, use this to detected them,
must have class instance

    # Set board controler address, use it carefully, reboot module to make it effective
    '''
    board.set_addr(0x10)
    if board.last_operate_status != board.STA_OK:
        print("set board address faild")
    else:
        print("set board address success")
    '''

    while board.begin() != board.STA_OK:    # Board begin and check board status
        print_board_status()
        print("board begin faild")
        time.sleep(2)
        print("board begin success")

    board.set_pwm_enable()                # Pwm channel need external power
    # board.set_pwm_disable()
    board.set_pwm_frequency(1000)        # Set frequency to 1000HZ, Attention: PWM
voltage depends on independent power supply

    while True:
        print("set all pwm channels duty to 30%")
        board.set_pwm_duty(board.ALL, 30)    # Set all pwm channels duty
        time.sleep(1)

        print("set part pwm channels duty to 60%")
        board.set_pwm_duty(0, 60)    # Set pwm0 channels duty
        #board.set_pwm_duty(1, 70)    # Set pwm1 channels duty
        #board.set_pwm_duty(2, 80)    # Set pwm2 channels duty

```

```
#board.set_pwm_duty(3, 90) # Set pwm3 channels duty  
time.sleep(1)
```

SPI Port Example

Refer to [DFR0413 Inch OLED Display Module For Raspberry Pi Wiki](#).

FAQ

For any questions, advice or cool ideas to share, please visit the [DFRobot Forum](#)