LAB MANUAL 3

Interfacing PWM Sensors

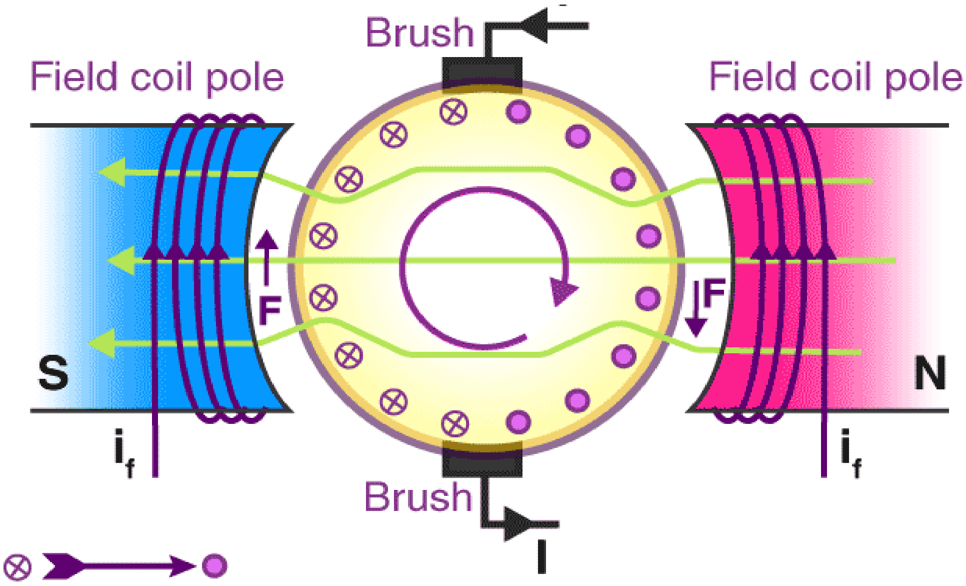
Disclaimer: The content is curated from online/offline resources and used for educational purpose only

DC Motor Fan

A DC motor is an electrical machine that converts electrical energy into mechanical energy. In a DC motor, the input electrical energy is the direct current which is transformed into the mechanical rotation.

DC Motor Working

A magnetic field arises in the air gap when the field coil of the DC motor is energised. The created magnetic field is in the direction of the radii of the armature. The magnetic field enters the armature from the North pole side of the field coil and "exits" the armature from the field coil's South pole side.



DC Motor Working

The conductors located on the other pole are subjected to a force of the same intensity but in the opposite direction. These two opposing forces create a torque that causes the motor armature to rotate.

Working principle of DC motor

When kept in a magnetic field, a current-carrying conductor gains torque and develops a tendency to move. In short, when electric fields and magnetic fields interact, a mechanical force arises. This is the principle on which the DC motors work.

DC Motor Interface with GrovePI & RAspberryPI

Github Link
https://github.com/Code-Unnati/Advance-Course/blob/master/Module-2/Unit-3/GrovePI_Codes/15_DC_Motor_Control.py

Code

```
import time
import grovepi
from grove_rgb_lcd import *
# Connect the Rotary Angle Sensor to analog port A2
potentiometer = 2

# Connect the dc_motor to digital port D5
# Check for PWM pin
dc_motor = 5

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

while True:
    try:
        # Read resistance from Potentiometer
        i = grovepi.analogRead(potentiometer)
        print(i)

        # Send PWM signal to dc_motor
        grovepi.analogWrite(dc_motor,i//4)
        setRGB(i//4,i//4,i//4)

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

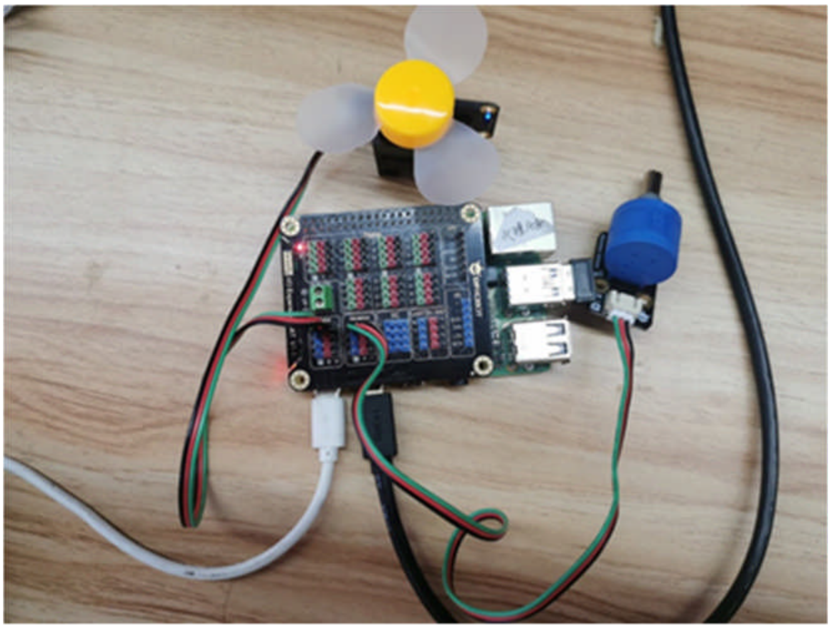
DC Motor Interface with DFRobot Hat & RAspberryPI

This module is really interesting. It can be easily driven by Arduino/RaspberryPI without an additional motor driver board. You can also use the PWM pulse width to adjust its speed, which is suitable for light applications or small DIY. Simple, but useful.

DC Motor Fan

Use 130 DC Motor Fan on Your Raspberry Pi

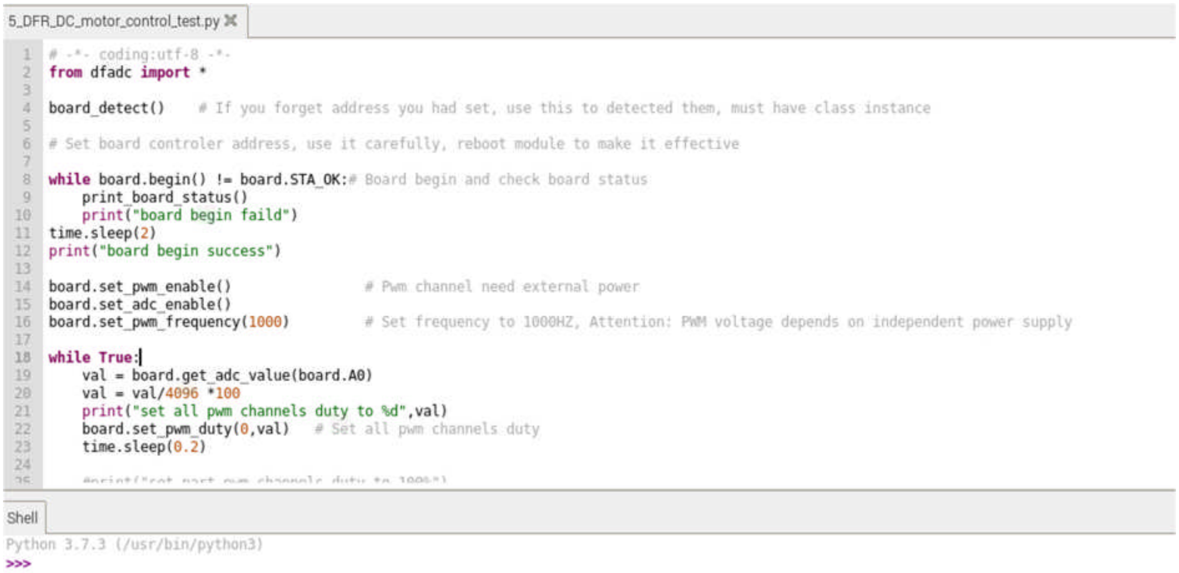
1. Power the Raspberry Pi on and install the Raspberry Pi expansion board correctly
2. Connect the fan to PWM port 0 on the expansion board, and the analog rotation sensor to analog port 0
3.



Github Link
https://github.com/Code-Unnati/Advance-Course/blob/master/Module-2/Unit-3/DFRobot_IoT_Codes/5_DFR_DC_fan_control.py

Code

```
# -*- coding:utf-8 -*-
from dfadc import *

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

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_adc_enable()
board.set_pwm_frequency(1000)              # Set frequency to 1000HZ, Attention:
PWM voltage depends on independent power supply

while True:
    val = board.get_adc_value(board.A0)
    val = val/4096 *100
    print("set all pwm channels duty to %d",val)
    board.set_pwm_duty(0,val)   # Set all pwm channels duty
    time.sleep(0.2)

    #print("set part pwm channels duty to 100%")

    #board.set_pwm_duty(0, 50)   # 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(5)
```



5_DFR_DC_motor_control_test.py

```python
# -*- coding:utf-8 -*-
from dfadc import *

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

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_adc_enable()
board.set_pwm_frequency(1000)          # Set frequency to 1000HZ, Attention: PWM voltage depends on independent power supply

while True:
    val = board.get_adc_value(board.A0)
    val = val/4096 *100
    print("set all pwm channels duty to %d",val)
    board.set_pwm_duty(0,val)   # Set all pwm channels duty
    time.sleep(0.2)

    #print("set part pwm channels duty to 100%")
```

```
Python 3.7.3 (/usr/bin/python3)
>>>
```

Analog Light Sensor Controlled FAN Speed

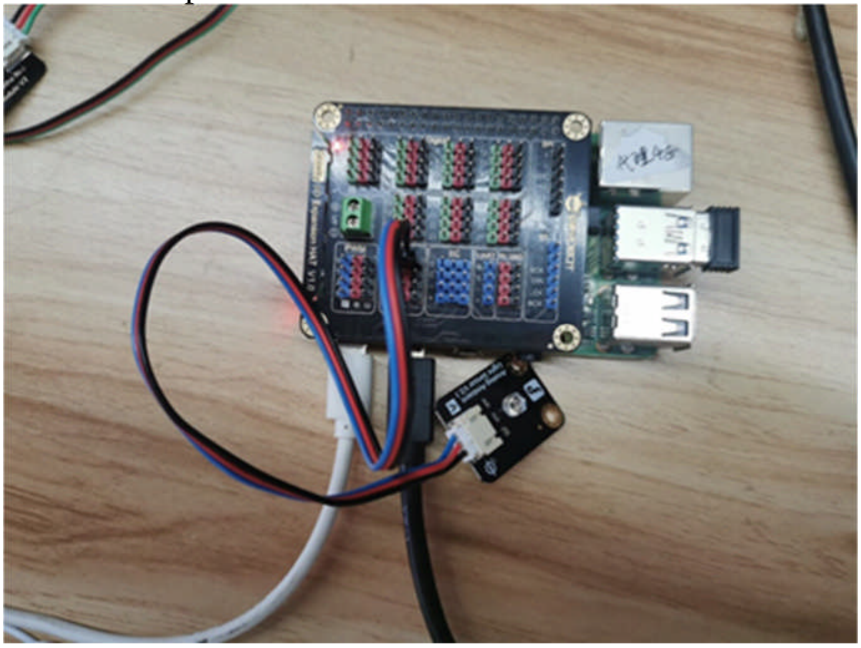
Interfacing Analog Light Sensor Using DFRobot Hat & RaspberryPI

Based on PT550 environmentally friendly photodiode, this light sensor can be used to detect the intensity of ambient light. It is usually used to produce interactive works that produce special effects with changes in light intensity. The entire module is connected to the IO expansion board with a 3P analog data cable. As long as the color is corresponding, it will not be inserted wrong, really convenient.

DFR Light Sensor

Use Analog Light Sensor on Your Raspberry Pi

- Power the Raspberry Pi on and install the Raspberry Pi expansion board correctly
- Connect the sensor to analog port 0 and connect DC motor Fan to PWM0 on the expansion board

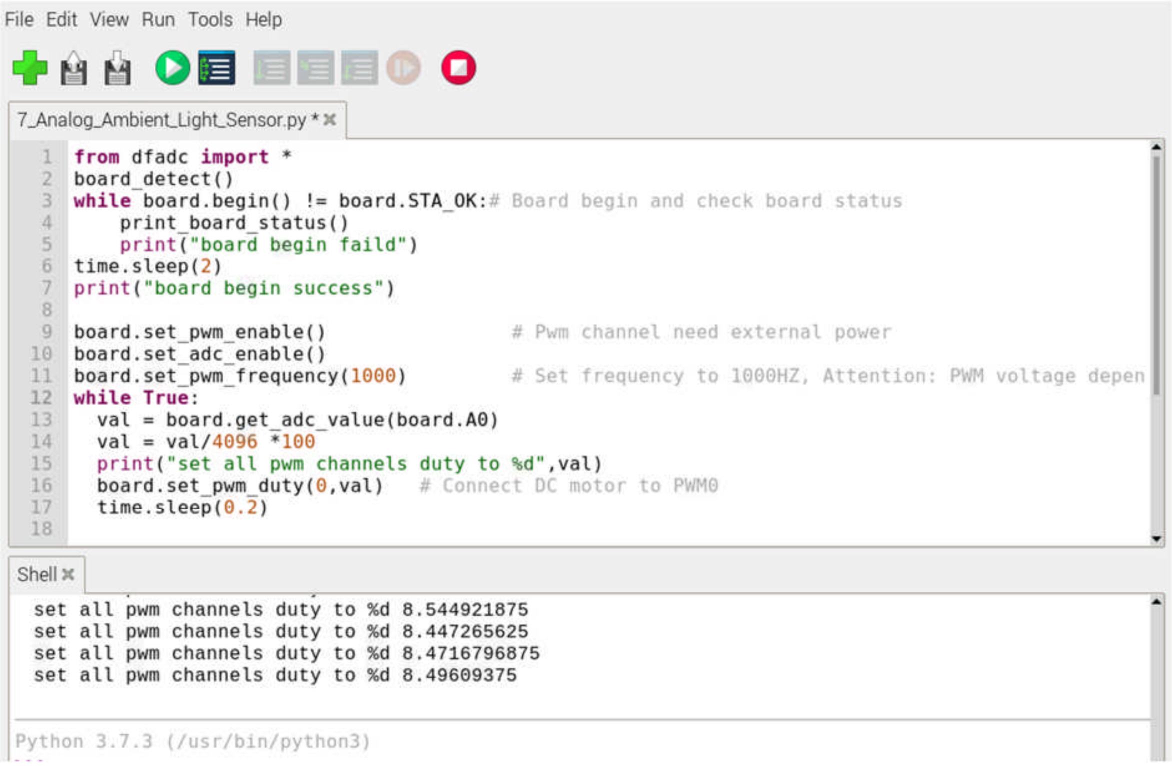


Github Link
https://github.com/Code-Unnati/Advance-Course/blob/master/Module-2/Unit-3/DFRobot_IoT_Codes/9_DFR_Analog_Sound_Sensor.py

Code-

```
from dfadc import *

board_detect()

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_adc_enable()
board.set_pwm_frequency(1000)               # Set frequency to 1000HZ, Attention: PWM voltage depends on independent power supply

while True:
  val = board.get_adc_value(board.A0)
  val = val/4096 *100
  print("set all pwm channels duty to %d",val)
  board.set_pwm_duty(0,val)   # Connect DC motor to PWM0
  time.sleep(0.2)
```


File Edit View Run Tools Help
7_Analog_Ambient_Light_Sensor.py *
Shell

```
from dfadc import *
board_detect()
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_adc_enable()
board.set_pwm_frequency(1000)         # Set frequency to 1000HZ, Attention: PWM voltage depen
while True:
  val = board.get_adc_value(board.A0)
  val = val/4096 *100
  print("set all pwm channels duty to %d",val)
  board.set_pwm_duty(0,val)   # Connect DC motor to PWM0
  time.sleep(0.2)
```

```
set all pwm channels duty to %d 8.544921875
set all pwm channels duty to %d 8.447265625
set all pwm channels duty to %d 8.4716796875
set all pwm channels duty to %d 8.49609375

Python 3.7.3 (/usr/bin/python3)
```

Sound Sensor Controlled FAN Speed

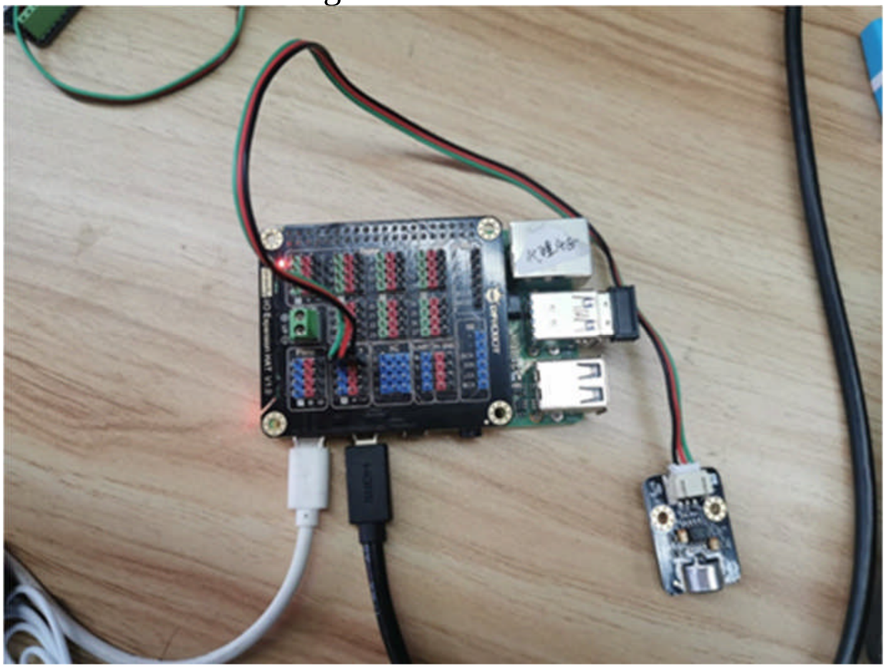
Interfacing Sound Sensor with DFRobot & RaspberryPI

This is a simple and affordable microphone through which the RaspberryPI can sense the level of the sound and convert it into an analog signal. That is, the volume is reflected by the feedback voltage value.

The sensor reads 0 when completely muted. When there is music nearby, it will read various readings with the volume.

Use Analog Sound Sensor on Your Raspberry Pi

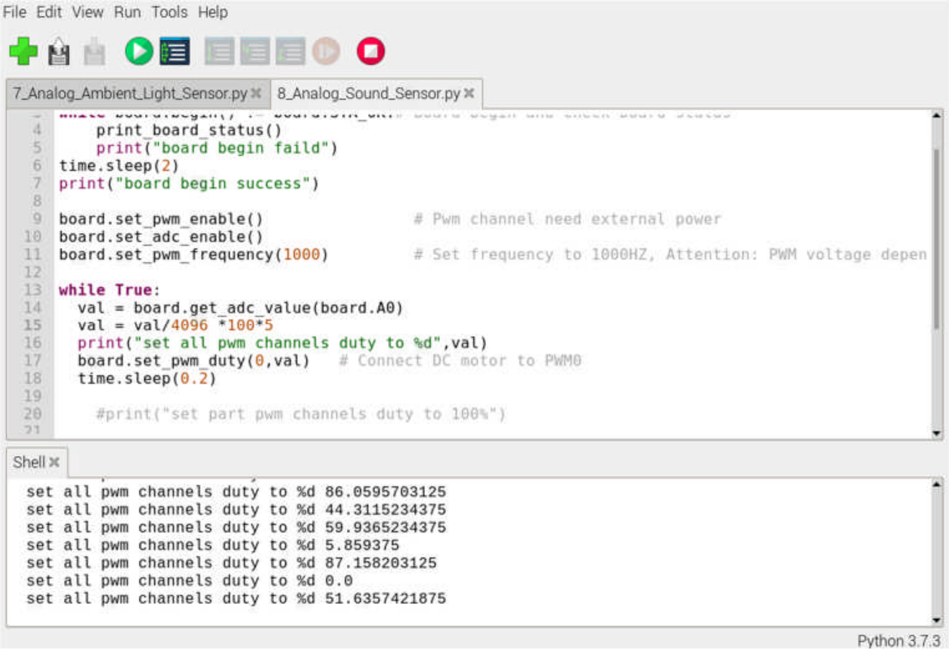
1. Power the Raspberry Pi on and install the Raspberry Pi expansion board correctly
2. Connect the sensor to analog port 0 on the expansion board
3. Also connect DC motor at PWM0 port
4. Use mobile songs to test the loundness in music



Github Link
https://github.com/Code-Unnati/Advance-Course/blob/master/Module-2/Unit-3/DFRobot_IoT_Codes/9_DFR_Analog_Sound_Sensor.py

Code-

```
from dfadc import *
board_detect()
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_adc_enable()
board.set_pwm_frequency(1000)              # Set frequency to 1000HZ, Attention: PWM voltage depends on independent power supply

while True:
  val = board.get_adc_value(board.A0)
  val = val/4096 *100*5
  print("set all pwm channels duty to %d",val)
  board.set_pwm_duty(0,val)   # Connect DC motor to PWM0
  time.sleep(0.2)
```


File Edit View Run Tools Help
7_Analog_Ambient_Light_Sensor.py
8_Analog_Sound_Sensor.py
Shell
Python 3.7.3

```
    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_adc_enable()
board.set_pwm_frequency(1000)         # Set frequency to 1000HZ, Attention: PWM voltage depen

while True:
  val = board.get_adc_value(board.A0)
  val = val/4096 *100*5
  print("set all pwm channels duty to %d",val)
  board.set_pwm_duty(0,val)   # Connect DC motor to PWM0
  time.sleep(0.2)

    #print("set part pwm channels duty to 100%")
```

```
set all pwm channels duty to %d 86.0595703125
set all pwm channels duty to %d 44.3115234375
set all pwm channels duty to %d 59.9365234375
set all pwm channels duty to %d 5.859375
set all pwm channels duty to %d 87.158203125
set all pwm channels duty to %d 0.0
set all pwm channels duty to %d 51.6357421875
```

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