

## IoT exercises – Week 3

This week, we will connect a sensor with NodeMCU/ESP8266 and read the sensory data from it. Particularly, the **temperature and humidity** sensor **DHT11** will be used together with built-in modules provided by the platform. Furthermore, we will look at the **button based ON/OFF** control and implement it. If you have not finished the previous exercises, please go on with them. Should you need further support, please just let me know it during the practical.

**The official documentation is provided here again for your reference. Please remember, when you are building your own IoT project in the future, always refer to the bespoke built-in modules and read their **documentation** first, which will be helpful!!!**

<https://nodemcu.readthedocs.io/en/master/>

In the lecture we have talked about the types of sensors. And the DHT11 and SR501 are introduced as examples for you to understand how to read sensory data from a sensor and how to refer to the datasheet to know the principles and specifications of the sensor. This week we will mainly practice on the DHT11 sensor using **dht** module, which has been included in your firmware bin file. And we will combine our previous exercises with the new sensor.

**The details of **dht** module can be found in**

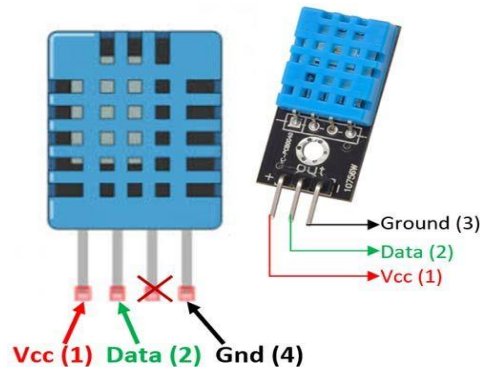
<https://nodemcu.readthedocs.io/en/master/modules/dht/>

The **infrared Motion Detection** sensor **SR501** will be optional for you to play with off the campus. Please come to me to bring it home during the practical if you would love to have a look into it.



## Exercise 1:

In this exercise, you will need to connect the **DHT11** sensor to the NodeMCU/ESP8266 using a breadboard and Dupont wires or using only wires. Please note that the DHT11 sensor has 3 pins, which can only function and communicate with your platform when they are correctly connected and powered.



- For the 3 pins in a DHT11.
  - Ground (GND) pin should connect to the GND.
  - VCC pin should connect to the 3.3V output pin. (Please refer to the datasheet (available on Moodle) for working **Power Supply**).
  - The Data pin (Out) should be connected to a GPIO pin of the NodeMCU to send data to the platform.
- Once you connected all the pins correctly, read and display the temperature and humidity info using the **dht** module.

=====

```
dhtPin = 2

status, temp, humi, temp_dec, humi_dec = dht.read11(dhtPin)

if status == dht.OK then

--3 different status

--dht.OK, dht.ERROR_CHECKSUM, dht.ERROR_TIMEOUT

    print("DHT Temperature:" ..temp..";".."Humidity:"..humi)

-- 2 dots are used for concatenation

elseif status == dht.ERROR_CHECKSUM then

    print( "DHT Checksum error." )
```

```
elif status == dht.ERROR_TIMEOUT then
```

```
    print( "DHT timed out." )
```

```
end
```

=====

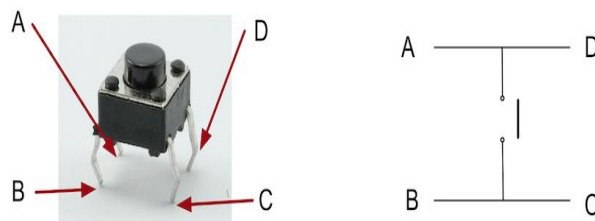
## Exercise 2:

In this exercise, you will need to combine the Timer and DHT11 sensor using the **tmr** and **dht** to provide real-time temperature and humidity info.

- Try different interval for the tmr, and check when there will be the error message
  - **DHT timed out from the status “dht.ERROR\_TIMEOUT”.**
- Determine the best interval for the sensor to operate and think about why it happens.
- Hold your sensor to see the change of the measured temperature and humidity.

## Exercise 3:

Now you are asked to include the 4pin button in the loop to control the ON/OFF of sensor reading.



- When the **button** is **pushed**, the AD and BC are connected, the LED is lighted and the DHT11 starts working and you get the temperature and humidity for once.
- Button click can be detected by **gpio** as follow

=====

```
buttonPin = 7
```

```
gpio.mode(buttonPin, gpio.INPUT)
```

```
gpio.write(buttonPin, gpio.LOW)
```

```
pushed = 0
```

```

mytimer = tmr.create()

mytimer:register(100, 1, function()

    if gpio.read(buttonPin)==1 then

        pushed = 1

        print("Button detected")

    end

end)

mytimer:start()

```

=====

- You may find a problem that **"Button Pushed" keeps being detected**. If you have experience in related issues, you should be able to solve it yourself and please go on with the rest exercises.
- If you don't know how to solve it, recall the optional **exercise in Week1**, you would need to reset the **Timer**.

#### **(Optional) Exercise 4:**

- Now go on with the button based ON/OFF control. This time the button is used to switch working modes of the DHT11.
- When the button is being **pushed without bouncing back for more than 2 seconds**, the DHT11 changes its working status and keeps providing temperature and humidity for every 5 seconds.
- With another **pushed without bouncing back for more than 2 seconds**, the DHT11 changes its working status and stops the cycle.
- If you don't know how to detect the **long push**, use the single **push detection from Exercise 3** instead, next week we will solve the problem together.

**Now you have your own local temperature and humidity meter, better with a LED/LCD screen when using it at home.**

#### **(Optional) Exercise 5:**

In this exercise, you can explore how you can read the **SR501** sensor detection

(Datasheet available on Moodle) results and used it as a controller in your application. Let me know if you need any other resources. If you can't play with SR501, let me know why.