

**IT Technology**

**Assignment A21 - Project**

A picture containing text, computer

Description automatically generated**First semester**

**Temperature and humidity station.**

*Authors*

**Team 6**

Simon Storm [ssch38571@edu.ucl.dk](file:///C:\Users\Marko\Downloads\ssch38571@edu.ucl.dk)

Marko Kozinec [mako37478@edu.ucl.dk](file:///C:\Users\Marko\Downloads\mako37478@edu.ucl.dk)

Bogdan Buterchi <bdbu37436@edu.ucl.dk>

Romulus Virgiliu Prundis [rvpr37603@edu.ucl.dk](mailto:rvpr37603@edu.ucl.dk)

Simon Slamka <sisl37451@edu.ucl.dk>

Dominik Tarcsi <dota39057@edu.ucl.dk>

Silvija Bazaraite <siba36733@edu.ucl.dk>

Thurdsay 4 November 2021

Table of Contents

[2 Introduction 1](#_Toc86924457)

[3 Requirements 1](#_Toc86924458)

[4 Inventory for solving the assignment 1](#_Toc86924459)

[5 Solution 1](#_Toc86924460)

[5.1 The code 1](#_Toc86924461)

[5.2 Use case showing one practical usage scenario for customer 3](#_Toc86924462)

[5.3 Specification of requirements and prioritization using MoSCoW. 3](#_Toc86924463)

[5.4 Description about hardware and software: - 4](#_Toc86924464)

[5.5 DHT11 Documentation 4](#_Toc86924465)

[Specifications: 4](#_Toc86924466)

[5.6 Circuit diagram of device with GPIO pins used. 5](#_Toc86924467)

[6 Literature used 6](#_Toc86924468)

# Introduction

OLA 21 is an assignment that needs to be passed in order to be admitted for the final exam. This specific OLA 21 is for *Project* where the knowledge of the student is being checked combining both programming and embedded systems until week 43. This means that student is able to:

* Turn on the microprocessor (Raspberry Pi 4) and use it to run the code
* Why conductors are essential when we use a flow of electrons (especially for LED)
* How to connect the male cable with the ground and the pins, resistors, as it needs to be an input and an out.
* A breadboard for connecting all the components
* Understand the pins on the DHT11 sensor that can be used for temperature measurement and
* humidity measurement
* Programming knowledge in Python about loops, variables, indentation and functions.

# Requirements

“The hand-in is a report in pdf format. The report needs to be handed in on wiseflow

before the deadline. Between 7-8 standard pages.

“Using one mix of the hardware described below, you must make an IoT device that is capable of measuring temperature and humidity. The data (temperature + humidity readings) must continuously be logged to a file. Finally, a red LED should light up when the temperature reads above 25 degrees.

Minimum requirements:

* Detailed description of the hardware and software (including OS, versions of the software, etc.) used.
* Specification of requirements and prioritization using MoSCoW.
* Use case showing one practical usage scenario for customer/user who might need this type of system.
* Circuit diagram of device with GPIO pins used. “ (Nielsen, 2021)

# Inventory for solving the assignment

A laptop

A [breadboard](https://en.wikipedia.org/wiki/Breadboard#/media/File:400_points_breadboard.jpg)

A [Raspberry Pi 4](https://www.raspberrypi.com/products/raspberry-pi-4-model-b/)

[Python 3.9.7](https://docs.python.org/3/library/idle.html) / [Thonny](https://thonny.org/)

[Male wires](https://duckduckgo.com/?q=male+wires&atb=v290-1&iax=images&ia=images&iai=https%3A%2F%2Fpotentiallabs.com%2Fcart%2Fimage%2Fcache%2Fcatalog%2Fnov-dec%2Fm-m-800x800.jpg) (4)

One [red Led](https://duckduckgo.com/?q=red+led&atb=v290-1&iax=images&ia=images&iai=http%3A%2F%2Fwww.taydaelectronics.com%2Fmedia%2Fcatalog%2Fproduct%2Fcache%2F1%2Fimage%2F500x500%2F9df78eab33525d08d6e5fb8d27136e95%2Fa%2F-%2Fa-1554.jpg)

Two [resistors 220R / 220 ohm](https://duckduckgo.com/?q=resistor+220&atb=v290-1&iax=images&ia=images&iai=https%3A%2F%2Fwww.diyelectronics.co.za%2Fstore%2F10512-thickbox_default%2Fresistor-220-ohm-14w-5.jpg)

[DHT11](https://components101.com/sensors/dht11-temperature-sensor)

# Solution

## The [code](https://gitlab.com/simtoon1011/UCL-ITtech/-/blob/team6/project/OLA_A21/temp-humidity.py)

import RPi.GPIO as GPIO

import dht11

from time import sleep

import requests

import sys

from datetime import datetime

sys.stderr = object

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)

GPIO.cleanup()

sensorPin = dht11.DHT11(pin = 11)

values = sensorPin.read()

GPIO.setup(26, GPIO.OUT)

def get\_time\_now(): # get system time

return datetime.now().strftime(' %H:%M:%S')

def getValues():

if values.is\_valid():

print(get\_time\_now())

print("Current temp:", values.temperature)

print("Current humidity:", values.humidity)

requests.post('http://auth.ongakken.com:2005/api/postMsgToUCLchannelDiscord', headers={"Content-Type":"application/json"}, json={'msg': f"Temperature: {values.temperature}C \n Humidity: {values.humidity}%RH"})

if values.temperature >= 25:

GPIO.output(26, GPIO.HIGH)

requests.post('http://auth.ongakken.com:2005/api/postMsgToUCLchannelDiscord', headers={"Content-Type":"application/json"}, json={"msg": "https://media.giphy.com/media/LMC8paGihNTuo/giphy.gif"})

else:

GPIO.output(26, GPIO.LOW)

else:

print("Return values invalid! Check pinout!!", values.error\_code)

requests.post('http://auth.ongakken.com:2005/api/postMsgToUCLchannelDiscord', headers={"Content-Type":"application/json"}, json={"msg": "Return values invalid! Check pinout!!"})

sleep(600)

while True:

print("Probing for temperature")

try:

getValues()

except KeyboardInterrupt:

print("Exiting.....")

break

except:

continue

Figure 2 – code in VSCode

## Use case showing one practical usage scenario for customer

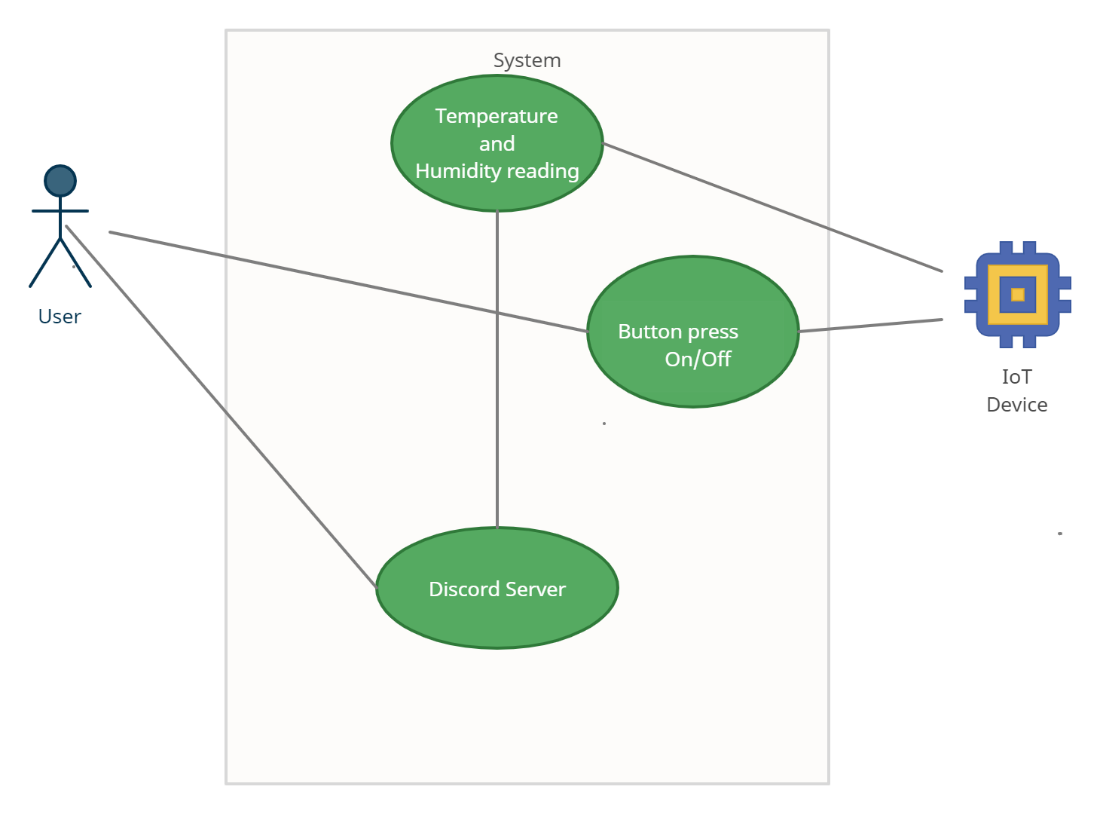


Figure 3 – Use case diagram

As a user I would like the LED to turn on when the temperatures is above 25 degrees so that I can open the window.

As a user I would like to be alerted when the temperature is too high by having the red light turn on so that I can open the window.

As a user I would like to have the data about the temperature stored, so that I am aware between which time I need to open the window, also when the humidity is not favourable.

## Specification of requirements and prioritization using MoSCoW.

Must have

Temperature measurement capability

Humidity measurement capability

Microprocessor (Raspberry Pi 4)

Red LED

Resistors

Should have

stdout data output

A blue LED for a temperature that is too low

A green LED for the sweet spot

A chronological database logging capability

Could have

A remote output capability (Discord)

An LCD screen for a quick visual output

LEDs for humidity

A 360dg thermal camera

A capability to call the fire department when a very high temp is reached

An LTE redundancy

Won’t have

Any other readings or hardware

A buzzer so when a critical temp is reached, the user is warned acoustically too

A coffee-making capability

## Description about hardware and software: -

We chose Raspberry Pi 4B as hardware, which is the latest release from the Raspberry series. It can be purchased with 1, 2, or 4 GB of memory (Everad, 2021). It runs on a high-performance 64-bit quad-core processor. The Raspberry Pi 4B has its own operating system called Raspberry Pi OS. Raspberry Pi OS is a Debian-based operating system. - We used the DHT11 sensor for our project. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness. Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration.

## DHT11 Documentation

## Specifications:

The sensor has Operating Voltage of 3 – 5.5 Volts, so when using the Raspberry Pi you can connect it to 3.3 Volts. Its operating current is 0.3mA if it’s measuring and 60uA if it’s in standby. Sensors output is transmitted over the serial interface. The temperature range in which the sensor can operate is  0°C to 50°C , and the humidity range in which the sensor can operate is 20% to 90%. The resolution in which it displays the values of temperature and humidity is 16-bit. Accuracy of the sensor is  ±1°C and ±1%. (Co, 2021)

Other similar sensors are: DHT22 , AM2302 and SHT71*.*

DHT11 sensor is a commonly used temperature and humidity sensor that comes with an NTC Thermistor to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data.

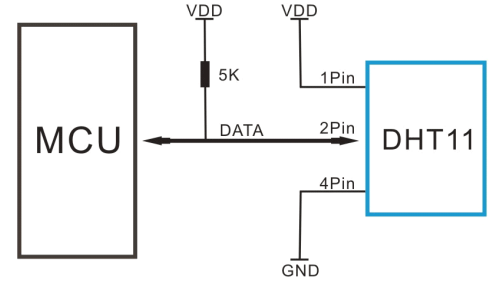


Figure 4 – “Schematics for physical connection of DHT11” (Co, 2021)

As you can see the data pin is connected to an Input/Output pin of the Microcontroller and a 5K pull-up resistor is used. This data pin outputs the value of both temperature and humidity as serial data. If you are trying to interface DHT11 with Arduino or in our case the Raspberry PI then there are ready-made libraries for it which will give you a quick start.

The DHT11 sensor can either be purchased as a sensor or as a module. Either way, the performance of the sensor is same. The sensor will come as a 4-pin package out of which only three pins will be used. The module will come with three pins as shown above. (Co, 2021)

If you are trying to interface it with some other MCU then the datasheet given below will come in handy. The output given out by the data pin will be in the order of 8bit humidity integer data + 8bit the Humidity decimal data +8 bit temperature integer data + 8bit fractional temperature data +8 bit parity bit. To request the DHT11 module to send these data the I/O pin has to be momentarily made low and then held high as shown in the timing diagram below

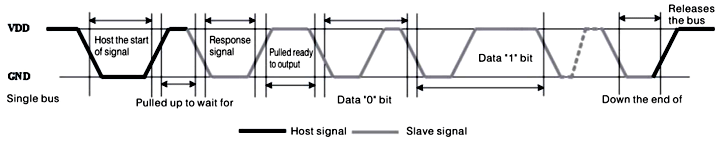


Figure 5 – “Data Timing Diagram” (Co, 2021)

The DHT11 sensor can be used for many different things such as dehumidifiers, testing and inspection equipment, consumer goods, automotive, automation, data loggers, weather stations, home appliances, humidity regulator, medical and other relevant humidity measurement and control (Co, 2021).

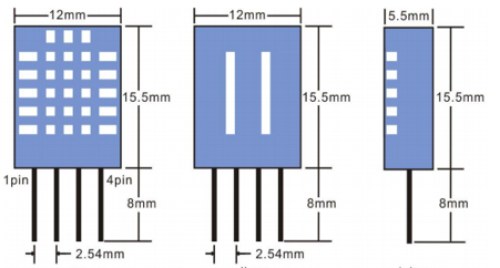


Figure 6 – “2D Image of the DHT11 Sensor” (Co, 2021)

The DHT11 sensor is cheap but it has a lot of problems. Some of the problems are:

-The temperature reads only in full degrees (the decimals are always zero)

-The setup is very slow to respond (if we try to blow into the sensor it will take almost 4 seconds to get the response)

-The temperature its reading is not correct so it is not so accurate.

(Co, 2021)

## Circuit diagram of device with GPIO pins used.

RaspberryPi 4 GPIO pins used:

GPIO17, related to the Pin 11 on the board

GPIO26, related to the Pin 37 on the board

GND Pin

DHT11 Pins:

VCC

SDA

NC

GND

GPIO17 from the RPi connected to the data pin from the DHT11 sensor represented by SDA, which will transmit the temperature and humidity values to our microprocessor.

GND from the RPi, which is the ground, connected to the GND from the DHT11 and the red LED

GPIO26 from the RPi connected to the 2.2k ohm resistor which limits current and gives power to the red LED.

VCC from the DHT11, connected through a 10k ohm resistor, to the 3.3V from the RPi giving the sensor power.

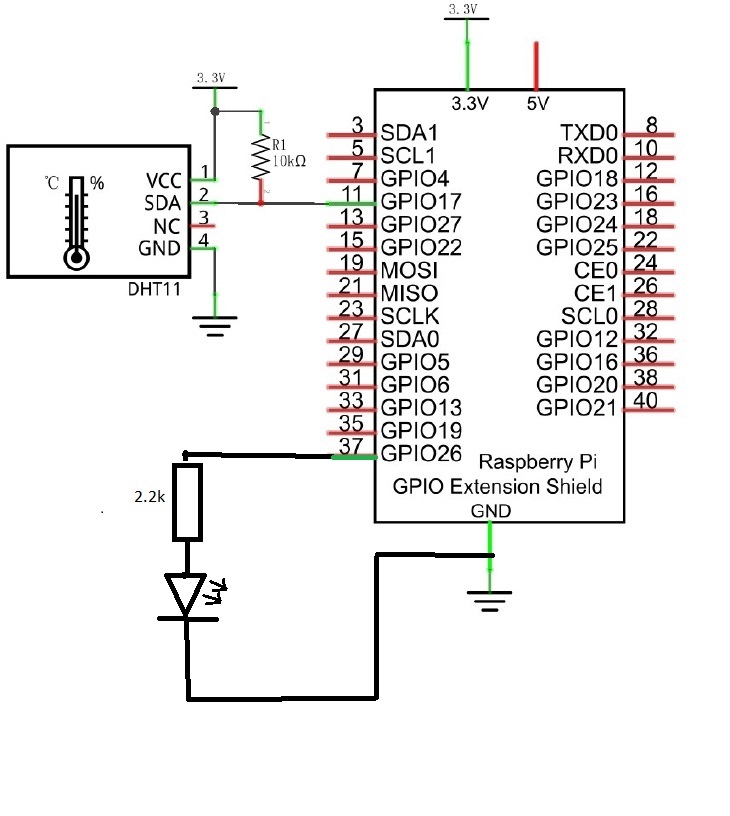


Figure 7 – circuit diagram

# Literature used

Co, A. G. (16. July 2021). *Components101.com*. Hentet fra www.aosong.com: https://components101.com/sensors/dht11-temperature-sensor

Everad, G. H. (2021). *Get started with MicroPython.* St. John's Innovation Park, Cowley Road, Cambridge: Raspberry Pi Trading Ltd.

Nielsen, H. A. (2021). UCL - PROJECT OLA A21. *UCL*, 2.