



Erasmus+



**FPIC**  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



EU-Geschäftsstelle  
Wirtschaft + Berufsbildung



## INDEX

<b>INTRODUCTION</b>	<b>1</b>
<b>SUBMODULE 1. ELECTRONIC CONNECTIONS</b>	<b>1</b>
<b>SUBMODULE 2. RASPBERRY PI AND PYTHON</b>	<b>2</b>
Explaining the main.py code	2
Used libraries	2
Firebase Configuration and connect	3
GPIO Configuration	3
Water sensor detection	4
Main Loop	4
Source Code for main.py	7
Explaining the timelapse code	11
Source Code for timelapse.py	11
<b>SUBMODULE 3. FIREBASE</b>	<b>13</b>
<b>SUBMODULE 4. ANGULAR AND IONIC</b>	<b>13</b>
<b>SUBMODULE 5. JSCHAR</b>	<b>13</b>

## INTRODUCTION

Once we have finished the teaching of the 5 submodules separately. Now we can teach the complete Plant Irrigation IOT Project with all its features, integrating the 5 submodules.

## SUBMODULE 1. ELECTRONIC CONNECTIONS



Erasmus+



**FPIC**  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



## SUBMODULE 2. RASPBERRY PI AND PYTHON

Explaining the main.py code

Used libraries

```
import time
import Adafruit_DHT
import pyrebase
from gpiozero import LED
import RPi.GPIO as GPIO
import re
```

### **Time**

This module provides various time-related functions.

In Plant irrigation is used to format the current data of record data. And also used to delay time for loops.

<https://docs.python.org/3/library/time.html>

### **Adafruit Python DHT Sensor Library**

Python library to read the DHT series of humidity and temperature sensors.

Designed specifically to work with the Adafruit DHT series sensors

[https://testpypi.python.org/pypi/Adafruit\\_DHT](https://testpypi.python.org/pypi/Adafruit_DHT)

### **Pyrebase**

Library used to connect with Firebase API

<https://github.com/thisbejim/Pyrebase>

### **GPIOZERO**

A simple interface to GPIO devices with Raspberry Pi.

<https://gpiozero.readthedocs.io/en/stable/>

### **RPi.GPIO**

By doing it this way, you can refer to it as just GPIO through the rest of your script.

<https://sourceforge.net/p/raspberry-gpio-python/wiki/BasicUsage/>



Erasmus+



FPIC  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



EU-Geschäftsstelle  
Wirtschaft + Berufsbildung



## Import re

This module provides regular expression matching operations similar to those found in Perl <https://docs.python.org/2/library/re.html>

## Firestore Configuration and connect

Connecting configuration of the Firestore database connection and then initialize and create firestore database connection object.

```
config = {
    "apiKey": "AIzaSyD03IMGuSv8LLBKvStk0JJm6cath-yTbKg",
    "authDomain": "model-2.firebaseio.com",
    "databaseURL": "https://model-2.firebaseio.com",
    "storageBucket": "model-2.appspot.com",
}
firebase = pyrebase.initialize_app(config)
db = firebase.database()
```

## GPIO Configuration

```
# Sensor should be set to Adafruit_DHT.DHT11,
# Adafruit_DHT.DHT22, or Adafruit_DHT.AM2302.

sensor = Adafruit_DHT.DHT22

# Example using a Raspberry Pi with DHT sensor
# connected to GPIO4

pin = 4

# Control water level sensor:

pinwater = 17
GPIO.setup(pinwater, GPIO.IN)
```



Erasmus+



FPIC  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



EU-Geschäftsstelle  
Wirtschaft + Berufsbildung



## Water sensor detection

```
def callback(pinwater):  
    if GPIO.input(pinwater):  
        print("No water detected")  
    else:  
        print("Water detected")  
  
GPIO.add_event_detect(pinwater, GPIO.BOTH, bouncetime= 600)  
GPIO.add_event_callback(pinwater, callback)
```

## Main Loop

Code for starting the main loop of the project

```
# Main body of code  
  
try:  
    while True:
```

## Connecting to the JSON node of Firebase

```
#Start firebase  
OpenWater = db.child("TReal").child("values").child("open").get()
```

## Formating the time information

```
hour = time.strftime("%H:%M:%S")  
date = time.strftime("%x")  
secods = time.strftime("%S")  
day = time.strftime("%A")  
hora = time.strftime('%H')
```

## Checking the MotorPumb on/off

```
#If OPenWater is true open the water pump  
  
if OpenWater.val():  
    GPIOWater.on()  
    print("Motor Pump is Open")  
else:  
    GPIOWater.off()
```



Erasmus+



FPIC  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



EU-Geschäftsstelle  
Wirtschaft + Berufsbildung



```
print("Motor Pump is Close")
```

## Getting Humidity and Temperature from sensors

```
# Try to grab a sensor reading. Use the read_retry method which
will retry up
# to 15 times to get a sensor reading (waiting 2 seconds between
each retry).

humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
humidity = round(humidity, 1)
temperature = round(temperature, 1)
```

### Humidity Register :

First check that the reading of the sensor is doing correctly

The temperature and humidity are formatted

Then update the current data in firebase with update(dato)

Every 60 update a new register in firebase with push(dato)

```
# Note that sometimes you won't get a reading and
# the results will be null (because Linux can't
# guarantee the timing of calls to read the sensor).
# If this happens try again!

if humidity is not None and temperature is not None:

    print('Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(temperature,
humidity))

    # Update Firebase

    dato = {"temperature":temperature,"humidity":humidity,
"time":hour,"day":hora + day}
    db.child("TReal/values").update(dato)
    if (count == 60):

        db.child("values").push(dato)
        count = 0
```



Erasmus+



FPIC  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



EU-Geschäftsstelle  
Wirtschaft + Berufsbildung



```
print( temperature)
print(humidity)
print(date, hour)
count = count +1
```

```
else:
    print('Failed to get reading. Try again!')
```

Save the image name with the recorded data  
Every Sunday at 00:00 makes a copy of everything.  
Every week calculate the max and min,

```
print(day)
print(hour)
name = hora + day
print(name)
Temp = []
Hum = []
if (day == "Sunday") and re.match("00:00:2.",hour):

    all_dates = db.child("values").get()

    print(all_dates)

    for value in all_dates.each():

        obj = value.val()
        print(int(obj["temperature"]))
        Temp.append(int(obj["temperature"]))
        Hum.append(int(obj["humidity"]))

    # Update Firebase

    dato
    ={"HMax":max(Hum), "HMin":min(Hum), "TMax":max(Temp), "TMim":min(Temp), "Name":
    name, "Picture": name}
    db.child("Videos").push(dato)
```



Erasmus+



FPIC  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



EU-Geschäftsstelle  
Wirtschaft + Berufsbildung



```
print("Tmperatura max",max(Temp))  
print("Tmperatura min",min(Temp))  
time.sleep(8)
```

Control de keyboard interruption

```
except KeyboardInterrupt: # If there is a KeyboardInterrupt (when you press  
ctrl+c), exit the program and cleanup  
    print("Cleaning up!")
```

Source Code for main.py

```
# Import necessary libraries for commuication and display use  
  
import time  
import Adafruit_DHT  
import pyrebase  
from gpiozero import LED  
import RPi.GPIO as GPIO  
import re  
  
config = {  
    "apiKey": "AIzaSyD03IMGuSv8LLBKvStkOJJm6cath-yTbKg",  
    "authDomain": "model-2.firebaseio.com",  
    "databaseURL": "https://model-2.firebaseio.com",  
    "storageBucket": "model-2.appspot.com",  
  
}  
# GPIO config:  
# GPIO10 is configured for de opening and closing water motor  
  
GPIOWater= LED(10)  
  
firebase = pyrebase.initialize_app(config)  
db = firebase.database()  
  
# Sensor should be set to Adafruit_DHT.DHT11,
```



Erasmus+



FPIC  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



EU-Geschäftsstelle  
Wirtschaft + Berufsbildung



```
# Adafruit_DHT.DHT22, or Adafruit_DHT.AM2302.

sensor = Adafruit_DHT.DHT22

# Example using a Raspberry Pi with DHT sensor
# connected to GPIO4

pin = 4

# Counter to store every minute the values

count = 0

# Control water level sensor:

pinwater = 17
GPIO.setup(pinwater, GPIO.IN)

def callback(pinwater):
    if GPIO.input(pinwater):
        print("No water detected")
    else:
        print("Water detected")

GPIO.add_event_detect(pinwater, GPIO.BOTH, bouncetime= 600)
GPIO.add_event_callback(pinwater, callback)

# Main body of code

try:
    while True:

        #Start firebase

        OpenWater = db.child("TReal").child("values").child("open").get()
        hour = time.strftime("%H:%M:%S")
        date = time.strftime("%x")
        secods = time.strftime("%S")
```





Erasmus+



FPIC  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



EU-Geschäftsstelle  
Wirtschaft + Berufsbildung



```
day = time.strftime("%A")
hora = time.strftime('%H')

#If OPenWater is true open the water pump

if OpenWater.val():
    GPIOWater.on()
    print("Motor Pump is Open")
else:
    GPIOWater.off()
    print("Motor Pump is Close")

# Try to grab a sensor reading. Use the read_retry method which
will retry up
# to 15 times to get a sensor reading (waiting 2 seconds between
each retry).

humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
humidity = round(humidity, 1)
temperature = round(temperature, 1)

# Note that sometimes you won't get a reading and
# the results will be null (because Linux can't
# guarantee the timing of calls to read the sensor).
# If this happens try again!

if humidity is not None and temperature is not None:

    print('Temp={0:0.1f}*C Humidity={1:0.1f}%'.format(temperature,
humidity))

# Update Firebase

dato = {"temperature":temperature,"humidity":humidity,
"time":hour,"day":hora + day}
db.child("TReal/values").update(dato)
if (count == 60):

    db.child("values").push(dato)
```



Erasmus+



FPIC  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



EU-Geschäftsstelle  
Wirtschaft + Berufsbildung



```
        count = 0

        print( temperature)
        print(humidity)
        print(date, hour)
        count = count +1

    else:
        print('Failed to get reading. Try again!')

    print(day)
    print(hour)
    name = hora + day
    print(name)
    Temp = []
    Hum = []
    # (day == "Thursday") and
    if re.match("..:00:2.",hour):

        all_dates = db.child("values").get()

        print(all_dates)

        for value in all_dates.each():

            obj = value.val()
            print(int(obj["temperature"]))
            Temp.append(int(obj["temperature"]))
            Hum.append(int(obj["humidity"]))

        # Update Firebase

        dato
    ={"HMax":max(Hum), "HMin":min(Hum), "TMax":max(Temp), "TMim":min(Temp), "Name":
    name, "Picture": name}
        db.child("Videos").push(dato)

        print("Tmperatura max",max(Temp))
        print("Tmperatura min",min(Temp))
```



Erasmus+



FPIC  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



EU-Geschäftsstelle  
Wirtschaft + Berufsbildung



```
time.sleep(8)
```

```
time.sleep(1)
```

```
except KeyboardInterrupt: # If there is a KeyboardInterrupt (when you press  
ctrl+c), exit the program and cleanup  
    print("Cleaning up!")
```

## Explaining the timelapse code

### Used Libraries

```
import time  
import picamera  
import os  
import pyrebase  
import re
```

New libraries not explain in previous code.

#### **picamera**

is a pure Python interface to the Raspberry Pi camera module

<https://www.raspberrypi.org/documentation/usage/camera/python/README.md>

#### **os**

This module provides a portable way of using operating system dependent functionality.

<https://docs.python.org/2/library/os.html>

### Camera configuration

```
picam = picamera.PiCamera()  
#picam.resolution = (1024,768)  
picam.resolution = (800,600)
```



Erasmus+



FPIC  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



OS Change directory

```
os.chdir("/home/pi/VET4/Raspgarden")
```

Main Loop

```
try:
    while True:
```

Every configured time it capture a photo and save it to firebase . Real time with put.  
And wait every 5 minutes.

```
i = 0

while True:
    hour = time.strftime('%H')
    if re.match("..:00:2.",hour):
        break
    i +=1
    picam.capture('real.jpg')
    storage.child("images/realTime.jpg").put("real.jpg")
    picam.capture('images/img{0:04d}.jpg'.format(i))
    time.sleep(300) #each 5 min
```

Create the video mp4

First naming the video with the correct format to associate to the data register.

Then with os functions convert the stored photos in a mp4 video and rename it with the correct format name and then register the names of video and photo in firebase.

```
##### MP4 o AVI #####

# Before using avconv we need install libav-tools

#     nameVideo = time.strftime("%d") + time.strftime("%B")+ ".mp4"

nameVideo = time.strftime('%H') + time.strftime("%A")+ ".mp4"
```



Erasmus+



FPIC  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



```
namePicture = time.strftime('%H') + time.strftime("%A")+ ".jpg"

print( time.strftime("%H") + time.strftime("%A"))

os.system('avconv -r 10 -i images/img%04d.jpg -r 10 -vcodec libx264
-crf 20 -g 15 video/timelapse.mp4 ')

#os.system('avconv -r 10 -i img%04d.jpg -r 10 -vcodec libx264 -crf 20 -g 15
timelapse.avi ')

os.rename("video/timelapse.mp4", "video/{}".format(nameVideo))

storage.child("video/" + nameVideo).put("video/" + nameVideo)
storage.child("images/" + namePicture).put("real.jpg")

print('done')
time.sleep(8)
```

## Source Code for timelapse.py

```
#!/usr/bin/python

import time
import picamera
import os
import pyrebase
import re

config = {
    "apiKey": "AIzaSyD03IMGuSv8LLBKvStkOJJm6cath-yTbKg",
    "authDomain": "model-2.firebaseio.com",
    "databaseURL": "https://model-2.firebaseio.com",
    "storageBucket": "model-2.appspot.com",
}

firebase = pyrebase.initialize_app(config)
storage = firebase.storage()

picam = picamera.PiCamera()
```



Erasmus+



FPIC  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



EU-Geschäftsstelle  
Wirtschaft + Berufsbildung



```
#picam.resolution = (1024,768)
picam.resolution = (800,600)

os.chdir("/home/pi/VET4/Raspgarden")

try:
    while True:

        # for i in range(270):

            i = 0

            while True:
                hour = time.strftime('%H')
                if re.match("..:00:2.",hour):
                    break
                i +=1
                picam.capture('real.jpg')
                storage.child("images/realTime.jpg").put("real.jpg")
                picam.capture('images/img{0:04d}.jpg'.format(i))
                time.sleep(300) #each 5 min

##### MP4 o AVI #####

# Before using avconv we need install libav-tools

#     nameVideo = time.strftime("%d") + time.strftime("%B")+".mp4"

    nameVideo = time.strftime('%H') + time.strftime("%A")+".mp4"
    namePicture = time.strftime('%H') + time.strftime("%A")+".jpg"

    print( time.strftime("%H") + time.strftime("%A"))

    os.system('avconv -r 10 -i images/img%04d.jpg -r 10 -vcodec libx264
-crf 20 -g 15 video/timelapse.mp4 ')

#os.system('avconv -r 10 -i img%04d.jpg -r 10 -vcodec libx264 -crf 20 -g 15
timelapse.avi ')

```



Erasmus+



FPIC  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



EU-Geschäftsstelle  
Wirtschaft + Berufsbildung



```
os.rename("video/timelapse.mp4", "video/{}".format(nameVideo))
```

```
storage.child("video/" + nameVideo).put("video/" + nameVideo)
```

```
storage.child("images/" + namePicture).put("real.jpg")
```

```
print('done')
```

```
time.sleep(8)
```

```
except KeyboardInterrupt: # If there is a KeyboardInterrupt (when you press  
ctrl+c), exit the program and cleanup
```

```
print("Finalized!")
```

## SUBMODULE 3. FIREBASE

model-2

TReal

values

day: "08Friday

humidity: 41.9

open: false

temperature: 26.8

time: "08:56:31

Videos

-L8atgKwTYQ\_1WzeXT3m

HMax: 46

HMin: 45

Name: "12Tuesday

Picture: "12Tuesday

TMax: 30

TMim: 24



Erasmus+



**FPIC**  
FORMACIÓN PROFESIONAL  
IES CAMPANILLAS



values

-L8aq-zwxsw5nOIZo8Ge

day: "11Tuesday

humidity: 46

temperature: 24

time: "11:44:15

## SUBMODULE 4. ANGULAR AND IONIC

SADASDASD

## SUBMODULE 5. JSCHAR

Asdas

## NEXT STEPS

CAMERA 360

PRINTER 3D