**LBRES2104 / HYDR0015 - Irrigation**

# Project report : Green walls


Group 3

**CASSIERS Adrien**

**DELESPESSE Matthieu**

**GONZALEZ Eloy**

**HENQUIN Jonathan**

**HENRION Maud**

1st master environment option

Professors : GARRE Sara, JAVAUX Mathieu, DUPRIEZ Christophe


**Academic year 2019 - 2020**

# Contents

# List of Figures

# List of Tables

# 1 Introduction

## 1.1 Global water context

At the beginning of the 1960s, the Soviet Union undertook a major water diversion project in the arid plains of Kazakhstan. The two main rivers in the region, the Syr Darya and the Amu Darya have been used unconsciously to irrigate cotton an other crops. The consequences of the misuse are dramatic. The Aral sea dried up, fisheries and communities that depended on them collapsed. The increasingly salty water became polluted with fertilizers and pesticides emitted by intensive agriculture.The blowing dust from the exposed lakebed, contaminated with agricultural chemicals, became a public health hazard. In fact, in this region the child mortality rate is one of the bigger in the world and cancers are rising steadily [48]. After fifty years the fourth biggest salted lake in the world, measuring twice the size of Belgium has decreased by 90 %.

Fresh water is the most important resource for humanity, at the crossroads of all human activities: social, economic, environmental. Water is an essential element for life on our planet and without it, life would be not possible. In recent decades, water becomes scarce and precious taking the name “blue gold”. Today, freshwater accounts for less than 1 % of the water on earth and more than 70 % of this water is used for agriculture. Industry and domestic needs cover the remaining 30 %. Since the late 1970s, the number of irrigated agricultural areas has increased by more than 100 % and cultivated land by more than 15 %. Agriculture has therefore a significant impact on water resources and will be in the future a key player in the management of the blue gold. So all the human activities have to make efforts to preserve water resources [47].

## 1.2 Water crisis

Despite its apparent abundance, water is unevenly distributed between countries and humans. According to [54], six countries including Brazil and Russia own together 60 % of the world’s water resources. The resulting social inequalities are also very significant. For example, in some countries, rural population doesn’t have the same accessibility to drink water as urban population. According to the World Health Organization, currently, nearly 30 % of the world’s population has still no access to drinking water and in nearly 70 % of cases, these populations are rural. It should also be noted that major inequalities in access to water can be observed within urbanized areas such as large cities in South America: Sao Paulo, Buenos Aires. In addition to unequal access to the resources, water becomes scarce and the main threat is the population explosion that began at the beginning of the 19th century. Globally, the potential of water availability for humans, i.e. the quantity of water available for each person per year is decreasing and this mainly in countries with high population growth in arid regions. We find for example Pakistan or Uganda. In addition to this demographic growth, living standards and urbanization are increasing. The main consequence of this quantitative decrease in water resources is the overexploitation of resources which results in the drying up of rivers and the pumping of water tables. The general decrease of the quality of the water is also a problem. In fact, due to the use of chemical products in agriculture and industries, the quality of water tends to deteriorate and amplify its scarcity [51].

## 1.3 Biodiversity crisis

The concept of biodiversity was introduced at the Rio convention in 1992. The United Nations conference on environment and development adopted a statement for the rights and the responsibilities of countries in the field of environment [50]. By definition, biodiversity is the variety of plant and animal life in the world or in a particular habitat. The current biodiversity losses may be part of the sixth mass extinction and the origin of this crisis are the side effects of human activities [26]. The urban environment and infrastructures play a central role in the decrease in biodiversity and ecosystem services. In fact, the area occupied by cities is growing twice as fast as the urban population [34] and the main causes are population explosion and rural exodus. Currently, cities are also experiencing a new phenomenon of urban exodus (population leaves the city center). Furthermore, according to the United Nations, in 2050, the population in the cities will increase by more than 2 million people [51]. That represents 68 percent of the global population. This expansion will have impacts on human health and also on the environment [15]. Finally, the urban environment takes over on nature [34]. There is an increase in temperature in cities because vegetation is replaced by dark surfaces and also an increase in runoffs because vegetation is replaced by impervious surfaces. However, nature provides many ecosystem services which provide social and environment such as pollination, water purification, decrease of pollutants, protection against erosion, improving the soil quality [32]. Therefore the development of vegetated infrastructures may be a solution to the problem of urbanisation and the loss of ecosystem services [31]. The integration of nature into our living spaces is increasingly popular in the design of buildings and cities and supported by the European Union. Nature-based solutions to mitigate impact of climate change, retain water and increase human well-being are increasingly investigated. The construction of green walls is part of this process. This project is at the crossroads of the two main concepts mentioned above: the water crisis and the decrease in natural areas in cities which causes a loss of biodiversity.

## 1.4 Green walls

A green wall is a vertical structure covered by vegetation. It is also referred to as living walls or vertical gardens. They are sometimes designed as aesthetic elements, sometimes as element of urban art or as elements of urban ecology. In addition, they can provide ecosystem services. The first green walls dates from the $7^{th}$ century BC and the world famous example is the hanging garden of Babylon considered one of the seven wonders of the ancient world [25]. The green wall were also used by Romans. They used vines and roses to protect themselves from the sun. Green roofs were the first structures covered by vegetation. They were used for centuries and they can be found in different countries as Mongolia, Norway or Turkey. In contrast to green roofs, green walls are studied since a few years and the subject is in constant evolution [19]. Green walls assets are multiple, but their design must be well-thought to deliver a maximum number of benefits. When applied in a significant urban scale, they can improve the urban environment by contributing to air quality [29] ; [27], stormwater management [33], temperature reduction [1] and biodiversity [18]. Besides those environmental aspects, they can provide social and economic benefits.

Today, most living walls rely on energy-consuming irrigation systems and non-native ornamental plants. The aim of this project is to propose a sustainable alternative to the irrigation design, techniques and automation of a modular living wall.

## 1.5 Project objective

Before Covid-19 crisis, the objective of the project was to design and implement an automated irrigation system for a living wall module of 1 m x 2.20 m based on expanded metal sheets and combined with a chosen mix of plants and locations within the module from a set of preselected native species.
At the end of the semester, an automated irrigation system was implemented on some little pots on Christophe's balcony due to the crisis.

## 1.6 Report structure

The first part of this report examines the system design of the wall. It describes the plants, the substrate, the hydraulic design and the wall construction. The following part is the system characterisation and monitoring which is valid for the wall and the pots. In this part of the report, the different sensors used are described such as the humidity sensors and those included in the weather station. The fourth part is devoted to the irrigation choices with the different strategies and the flowcharts, these choices are presented for the pots and for the wall. After that, the results (for the pots) and the critical analysis (for the wall) are given.

# 2 System design

## 2.1 Components

A completely automated irrigation system of the green wall module based on irrigation sensors and a weather station will be implemented via the WatWall hardware and software project. The system is composed of many different components in interaction with each other (Figure 1).

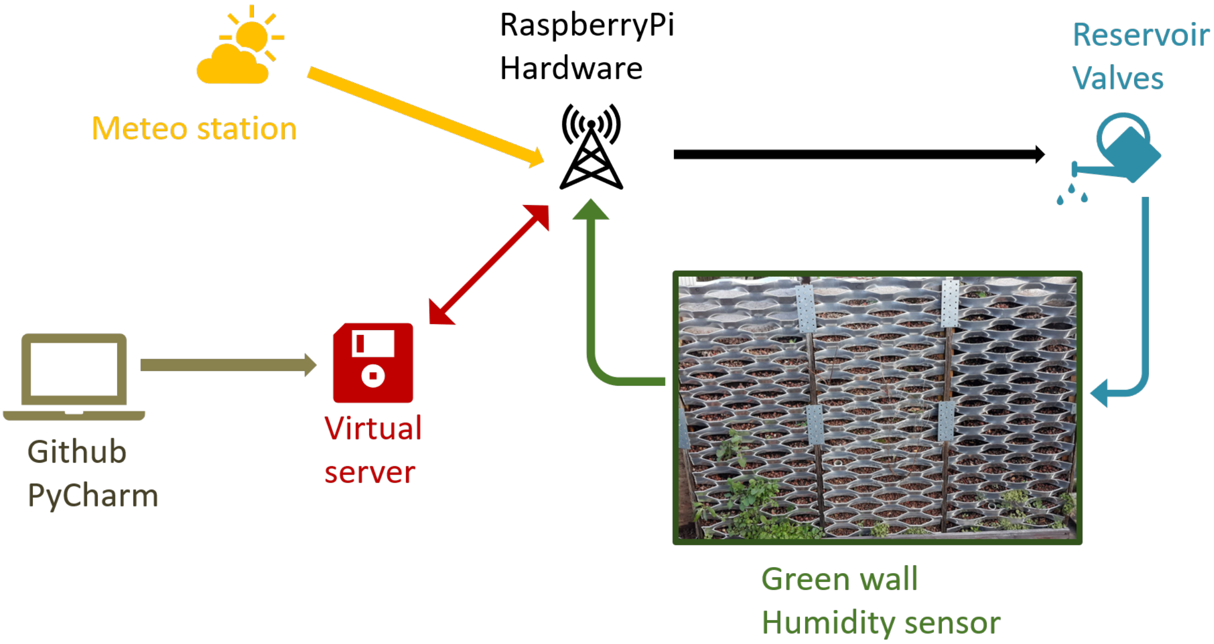



Figure 1: **Components of the green wall system and their interactions.**

These components are :

- A water source : here it is a 1000 liters reservoir located at 3 meters high on a scaffold collecting rainwater.
- An electrical ball valve that opens and closes the irrigation system when asked by the program.
- The green wall module (1.1 m height, 2.2 m long and 13 cm width) oriented South based on expanded metal sheets delivered by Metal Deployé.
- Substrate in the module.
- Plants.
- An irrigation system to bring water from the reservoir to the module. It consists of three Gardena Micro Drips pipes.
- Three humidity sensors (LICOR® EC-5) located in the module. They measure soil volumetric water content.
- A weather station (ATMOS® 41) attached near the reservoir at three meters high. It records : air temperature (°C), relative humidity (%), vapour pressure (kPa), barometric pressure (kPa), rainfall (mm/h), solar radiation (W/m$^2$), wind speed (m/s) and wind direction.

- A local computer (AKUINO® central). It features relays to open/close the valves, a SDI 12 serial interface board (LIUDR®) to collect data from the weather station, an analog-to-digital converter board (ADC differential Pi®) to collect data from the EC-5 sensors, an expander board to power sensors and micro-computer containing software to read sensors and order pumps and/valves (Raspberry P model B). A schematic description of the entire system and connections can be found in the appendix A.
- A virtual server on which the data is copied and that allows to execute irrigation programs (python 2.7 programs).

A scheme of the green wall module with the location of the irrigation pipes and the humidity sensors is presented in Figure 2. The location of the wall and the reservoir is shown in Figure 3.

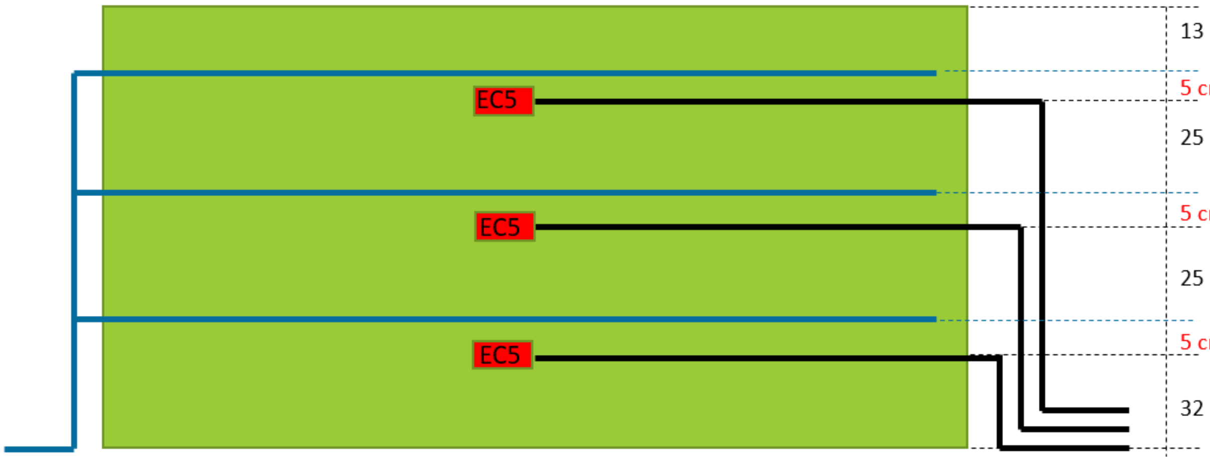



Figure 2: **Scheme of the green wall module and position of the dripping system and the EC-5 sensors.**

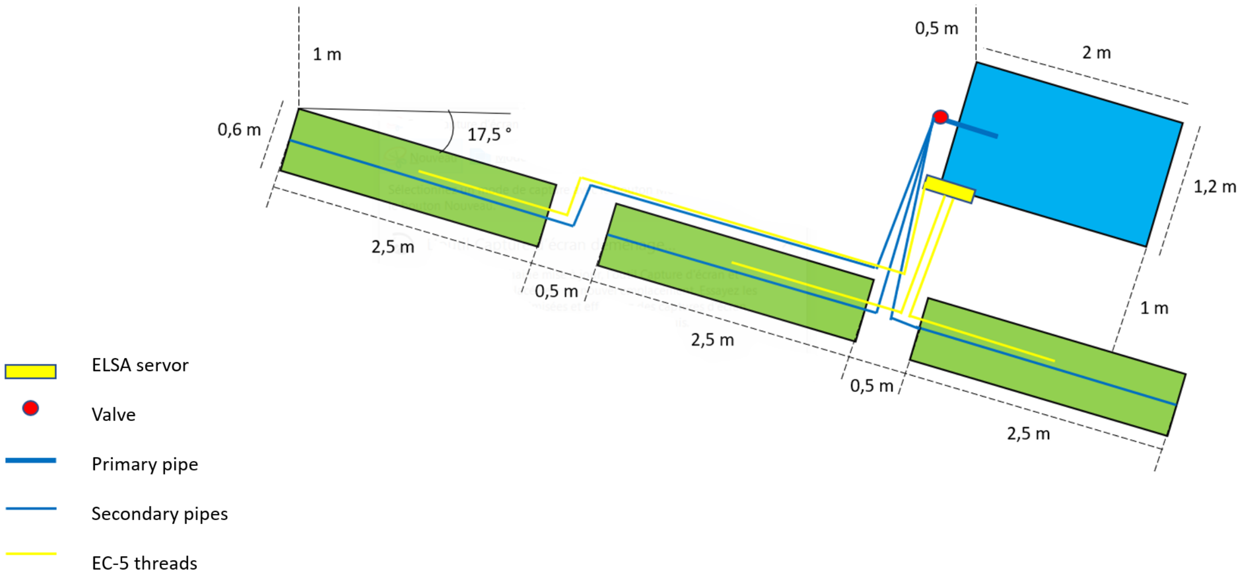



Figure 3: **Location of the three modules and the reservoir on the parking.**

## 2.2 Plants

### 2.2.1 Plants choice

A minimum of five native plants had to be installed in the green wall. The species made available for the project were the following:

- Edible plants : *Thymus pulegioides*, *Origanum vulgare*
- Ornamental plants : *Dianthus carthusianorum*, *Asplenium trichomanes*, *Geranium sanguineeum*, *Melica ciliata*, *Tragopogon pratensis*.

The research made about plants usually used in green walls led to the selection of six species. There is one species more than the minimum number required in order to always have the required number even if a particular species encounters problems. *Thymus pulegioides* and *Origanum vulgare* are among the most commonly used species in green walls [12] ; [21] ; [4]. Furthermore these two plants are edible which is a clear advantage. *Dianthus carthusianorum* is also used in green walls [21] as well as *Geranium sanguineeum* [20]. Then, *Asplenium trichomanes* [35] and *Melica ciliata* [14] where found in green walls too. These two plants are more voluminous and could bring texture and make the wall more aesthetically pleasing . Concerning *Tragopogon pratensis* no proof was found that it was largely used in green walls therefore this species was not retained.
The six plant species chosen are shown in the Figure 4.

The number of plants ordered per species is as follows: 8 *Thymus p.* and *Origanum v.*, 9 of *Dianthus c.* and *Geranium s.* And because *Asplenium t.* and *Melica c.* are more voluminous plants only 6 plants of each were ordered.

((a)) *Thymus pulegioides*

((b)) *Origanum vulgare*

((c)) *Dianthus carthusianorum*

((d)) *Asplenium trichomanes*

((e)) *Geranium sanguineeum*

((f)) *Melica ciliata*

Figure 4: **Pictures of the species chosen for the green wall** [49].

### 2.2.2 Plants location

After that some research was done to decide plants location in the module. Plants preferences in terms of soil moisture and sun exposure were considered to achieve this aim. This information is presented in the Table 1.

Table 1: **Soil and exposition characteristics of the chosen plants** [30] [52].

| Species | Environmental preferences | |
|---|---|---|
| | Soil moisture | Exposition |
| *T. pulegioides* | Dry | Sunny |
| *M. ciliata* | Intermediate | Intermediate |
| *D. carthusianorum* | Intermediate | Intermediate |
| *G. sanguineeum* | Dry | Intermediate |
| *O. vulgare* | Dry | Intermediate |
| *A. trichomanes* | Wet | Shadow |

Because *T. pulegioides* grows on dry soils and in sunny environments it will be placed on the top of the wall. On the contrary, *A. trichomanes* grows in humid and shaded soils. For these reasons it will be placed on the bottom of the wall. It will probably be the most humid place in the module because of water percolation by gravity. Then, *M. ciliata* and *D. carthusianorum* grow in fresh soils thus they will be place on the bottom center of the wall. Finally, *G. sanguineeum* and *O. vulgare* grow better in dry soils so they will be placed on the upper center. Choices were made based on the potential distribution of the water in the wall. This distribution is presented in figure 8. The position of the plants is presented in figure 5.

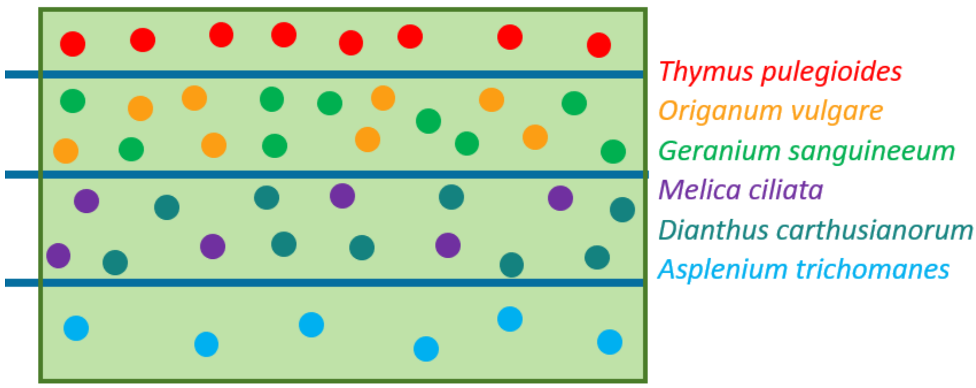



Figure 5: **Plants location in the wall.**

### 2.2.3 Water depletion (P) factor

In the soil, the maximum amount of water available for plants is called the total available water ($TAW$, in mm) [46]. It can be calculated with Equation 1. The $RAW$ is the fraction of the TAW that can be depleted from the root zone before moisture stress (reduction in evapotranspiration) occurs [2]. The RAW is specific to each plant. It is linked to the TAW by the p factor ($p$), as in Equation 2. The p factor can take values between 0.1 and 0.8. It is smaller for plants with high evapotranspiration.

$$TAW = (\theta_{FC} - \theta_{WP}) \cdot Z_r \qquad [mm] \tag{1}$$

With:

- $\theta_{FC}$: VWC at field capacity [cm$^3$/cm$^3$].
- $\theta_{WP}$: VWC at the wilting point [cm$^3$/cm$^3$].
- $Z_r$: rooting depth [mm].

$$RAW = p \cdot TAW \qquad [mm] \tag{2}$$

In the greenwall, the hypothesis was made that the p factor is the same for all plants. Indeed, the p factor of *M. ciliata*, a gramineous plant, can take values between 0.4 and 0.65 [2]. *T. pulegioides*, *O. vulgare* and *G. sanguineeum* are small plants with low evapotranspiration rate. Their p factor probably lies between 0.4 and 0.5 [2]. No data was found for *D. carthusianorum* and *A. trichomanes.* Their p factor was set to 0.5, since it is the p factor value that the FAO use the most when doing estimations [46]. Given the range of the known p factor and the standards of the FAO, this value of 0.5 is also the one chosen for the greenwall.

The difference between the moisture content at the field capacity and the RAW will be considered as the threshold below which the moisture content must never fall when irrigation is based on soil VWC measurements, to avoid water stresses.

### 2.2.4 Crop coefficient Kc

A crop coefficient (K$_c$) must be used in order to estimate the amount of water evapotranspired by the soil-plant system. It allows the conversion from the reference evapotranspiration ($ET_o$), that can be estimated using meteorological data, to the crop evapotranspiration ($ET_c$), in order to integrate the effects of characteristics that distinguish studied plants from grass [45]. ET$_c$ can be calculated with Equation 3.

$$ET_c = ET0 \cdot K_c \qquad [mm] \tag{3}$$

An experiment was designed to determine the Kc values of each plant. It consists in measuring the daily evapotranspiration of the plants for several days, and comparing it with the ET$_o$ estimate based on meteorological data. Evapotranspiration can be measured by assessing the water balance of a lysimiter on a 24 hours basis with the FAO method [43]. At the place where the wall should have been installed, shelves the height of the wall would have been erected. A roof would have been placed over them to insulate them from the rain. These shelves would have made it possible to place the plants in the conditions they would have faced on the wall. Two individuals of each species would have been placed at a height at which they would have been placed on the wall. The experiment would have last two weeks. Each morning the weight of the pots would have been measured with a balance. Then the pots would have been watered to come back to a given weight. Returning to the same weight each day eliminates the need to measure the moisture content of the pots. Pots without holes in the bottom would have been used to prevent drainage. Then the daily evapotranspiration could have been estimated by calculating the difference between the weight of the pot before and after the daily watering. The reference evapotranspiration would have been estimated using meteorological data from the weather station (see section Sensors). By comparing the measured evapotranspiration and the reference evapotranspiration, the K$_c$ of each plant could have be calculated using Equation 3.

Since the experiment was not conducted, tables were used to determine theoretical K$_c$ [8]. A K$_c$ value of 0.5 was found for *T. pulegioides*, *O. vulgare*, *D. carthusianorum* and *G. sanguineeum.* No values were found for *M. ciliata* and *A. trichomanes.* However, *M. ciliata* is a gramineous plant and has consequently a $K_c$ of approximately 0.8 [8]. Finally a constant crop coefficient of 0.6 was chosen for all the green wall. To estimate the $K_c$, the WUCOLS [36] method was initially considered, but in the end was not retained. Indeed, in this approach the $K_c$ is calculated from three other coefficients (plant density coefficient, plant species coefficient, microclimate coefficient), each of which has uncertainties. Moreover, this method is very well described for the region where it was developed (California), but much less so in Europe [24].

## 2.3 Substrate

A light, organic substrate was chosen to fill the wall. It consists of 10% expanded clay (8-16mm), 10% zinco (substrate for extensive green roofs), 20% pozzolana and 60% potting soil. It has several advantages. Firstly, it is poorly draining compared to other substrates, which minimizes water loss through drainage. Second, it has few large pores, which ensures the proper functioning of the humidity sensors (see section Sensors).

### 2.3.1 Measurements

First, the density at which the substrate would be packed had to be determined. This density had to be easy to apply, but also easily reproducible. The wall was filled carefully to obtain a homogeneous substrate density and thus have a representative measure of the entire wall. To achieve this, compaction was carried out with moderate force in every 10 cm layer. When the wall was filled, two samples of 200 cm$^3$ were taken with a Kopecky and then put in a 105 °C oven in order to evaporate all water. The bulk density $\rho_b$ was finally deduced using Equation 4. Unfortunately, due to the coronavirus crisis the samples couldn't be taken out of the oven.

$$\rho_b = \frac{M_f}{V} \qquad [g/cm^3] \tag{4}$$

With:

- $M_f$: the weight of the sample out of the oven $[g]$.
- $V$: the volume of the kopecky (200 cm$^3$).

Secondly, it was necessary to determine the VWC at field capacity ($\theta_{FC}$) and at the wilting point ($\theta_{WP}$). Four 100 cm$^3$ samples were put in a Richards chamber at a 300 hPa (pF 2.48) in order to measure $\theta_{FC}$. The water weight was calculated by subtracting the weights after the suction was applied (Equation 5) to the initial samples weights. The resulting VWC for the samples was between 0.274 and 0.297 cm$^3$/cm$^3$ (mean value of **0.285** cm$^3$/cm$^3$).

$$\theta_{FC} = \frac{(M_i - M_f)}{V \cdot \rho_{water}} \qquad [cm^3/cm^3] \tag{5}$$

Where :

- $M_i$: the weigth of the sample before going in the Richards chamber $[g]$.
- $M_f$: the weigth of the sample after going in the Richards chamber $[g]$.
- $\rho_{water}$; the density of water (1 g/cm$^3$).
- $V$: the sample volume (100 cm$^3$).

To determine $\theta_{WP}$, three samples were put in a Richards chamber at pF 4.2. The residual water content was then deduced using Equation 6. The results are not available due to the coronavirus outbreak.

$$\theta_{PF} = \frac{(M_i - M_f)}{V \cdot \rho_{water}} \qquad [cm^3/cm^3] \tag{6}$$

With:

- $M_i$ is the weigth of the sample before going in the Richards chamber $[g]$.
- $M_f$ is the weigth of the sample after going in the Richards chamber $[g]$.
- $\rho_{water}$ is the density of water (1 g/cm$^3$)
- $V$ is the sample volume (100 cm$^3$).

## 2.4 Hydraulic design

### 2.4.1 Pipes and pressure

A scheme of the green wall system hydraulic installation is presented in figure 6.

The reservoir has a length of 1,2 m and a width of 1 m. If it contains 1000 liters of water, the water height would be 83 cm and if it contains 800 liters then it would be 67 cm. It is approximated that the reservoir would be nearly full all the time and will have a volume oscillating between 800 and 1000 liters of water. If it goes below 800 some tap water will be added. The mean value is thus calculated and a water height of 75 cm is considered for the calculations. A ball valve is located at the outlet of the reservoir. This valve will be commanded by the code created for irrigation.

Concerning the pipes, the reservoir is connected to the green wall with a pipe of approximately 295 cm. This module was the nearest of the three to the reservoir because the humidity sensors available had smaller threads. The pipe has a diameter of 2 cm and is in rubber. To maximise the flow, it is better if the main pipe does not touch the soil before going into the walls but stay in height. When the main pipe arrives to the wall it is divided to provide water to the three irrigation pipes via some connections visible in figure 10 b.

With regard to the pressure at the exit of the irrigation pipes, it is approximated that it is equal to the atmospheric pressure, 1,01325 bar. But, it will be higher than that at the entrance of the main pipe, just at the outlet of the reservoir. There the pressure will be equal to the atmospheric pressure add up with the height of the water in the reservoir, 1,0868 bar if there is 75 cm water in the tank.

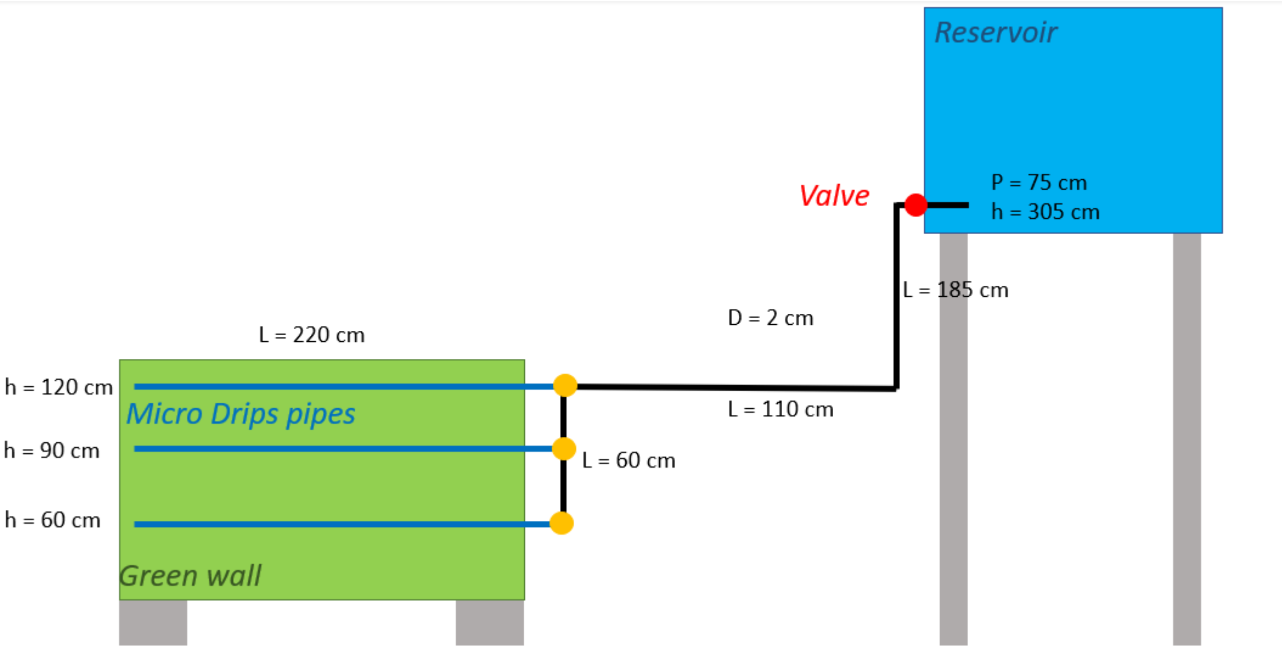



Figure 6: **Hydraulic design of the system**

### 2.4.2 Water flows

The initial plan was to test the water flow in real conditions at the three different heights in order to reproduce the real system before putting the irrigation pipes in the wall. The irrigation pipes would be put in a larger pipe that would lead to a little tank. This way the weight of the tank can be measured and the volume of irrigation water can be known. The ball valve would have been open during a specific time, five minutes for example, and weighted the amount of water corresponding to calculate the water flow. This would have been done for the three irrigation pipe conditions. Since the reservoir was never put on the scaffold these measurements could not be carried out.

Therefore, a theoretical calculation of the water flow is made to have an idea of what it would have been in reality. The aim is to obtain an order of magnitude not an exact value. In this regard, some hydraulic calculations based on a fluid and energy transfer course [38] were made. The calculations shown in the report are those accounting for the upper irrigation pipe but the principle is the same for the two other ones.

First, the water velocity is needed to deduce the water flow (equation 7).

$$Q = A \cdot v \tag{7}$$

Where :

- $Q$ is the volumetric flow rate $[m^3/s]$
- $A = \pi \cdot r^2 = 3,14 \cdot 10^{-4}$ is the area of the pipe $[m^2]$
- $v$ is the water velocity $[m/s]$

But this velocity is unknown. Therefore some iterations will be made to find this value. These iterations will be performed to validate the equation 8 following the law of conservation of mechanical energy.

$$\Delta K + g \cdot \Delta Z + \frac{\Delta P}{\rho} + W_f = 0 \tag{8}$$

Where :

- $K$ is the kinetic energy $[J]$
- $g$ = 9,81 is the gravitational force $[m/s^2]$
- $Z$ is the potential energy $[J]$
- $P$ is the gradient of pressure $[Pa]$
- $\rho$ = 1000 is the water density $[kg/m^3]$
- $W_f$ is the friction force $[N]$

This equation can be applied to a system if the entrance point and the end point are known (equation 9). Here, the entrance point (point 1) is located at the outlet of the reservoir just before the ball valve. The exit point (point 2) will be just at the end of a dripper in the middle of the wall in the upper pipe.

$$\frac{v_2^2 - v_1^2}{2g} + (z_2 - z_1) + \frac{P_2 - P_1}{\rho g} + \frac{W_f}{g} = 0 \tag{9}$$

Where :

- $v_2$ is unknown and is what we want to determine
- $v_1 = 0$ m/s

- $z_2 = 3{,}05$ m
- $z_1 = 1{,}2$ m
- $P_2 = 0$ m (relative pressure)
- $P_1 = 0{,}75$ m (relative pressure)

The only unknown left in equation 9, beside the water velocity, is $W_f$, the friction forces. These forces are composed of the distributed friction forces ($W_{f(distri)}$) and the singular friction forces ($W_{f(singu)}$) as shown in equation 10.

$$W_f = W_{f(distri)} + W_{f(singu)} \tag{10}$$

First, the distributed friction forces have to be determined. In this the Reynolds number has to be calculated (equation 11). After iterating, a value of approximately 8000 is found for this number. The Darcy friction coefficient $\lambda$ is determined from the Reynolds number and the roughness of the pipe via the Moody diagram. The roughness of the pipe is the roughness of rubber and is 25*$10^{-6}$ m [41]. When the $\lambda$ is obtained, the $Wf_{distri}$ can be found (equation 12).

$$Re = \frac{\rho v D}{\mu} \tag{11}$$

Where :

- $Re$ is the Reynolds number $[-]$
- $v$ is the unknown velocity $[m/s]$
- $D = 0{,}02$ is the pipe diameter$[m^2]$
- $\mu = 10^{-3}$ is the water viscosity $[kg/m.s]$

$$W_{f(distri)} = \lambda \cdot \frac{L}{D} \cdot \frac{v^2}{2} \tag{12}$$

Where :

- $\lambda$ is the Darcy friction coefficient $[-]$, determined from the Moody diagram
- L $= 4{,}05$ is the length of the pipe $[m]$

Second, the singular friction forces have to be calculated. Two techniques are possible : the first one calculates equivalent lengths ($L_{eq}$) for the singularities [38] and the second one calculates a singular friction force coefficient ($\zeta$) ; [53]. Here under the values of these coefficients and the number of each singularity can be found for the upper pipe. We then deduce the value of the singular friction forces (equation 13).

- Ball valve : $L_{eq} = 7{,}0104$ m, occurrence $= 1$
- 90 $^o$ elbow : $L_{eq} = 0{,}70104$ m, occurrence $= 2$
- Pipe separation : $\zeta = 1{,}5$ m$^{-1}$, occurrence $= 2$

$$W_{f(singu)} = \lambda \cdot \frac{\Sigma L_{eq}}{D} \cdot \frac{v^2}{2} + \Sigma\zeta \cdot \frac{v^2}{2} \tag{13}$$

All the parameters are known in the equation 9 beside the water velocity. Several iterations are operated until the equality of the equation 9 is verified. This fact is accepted when the equation equals less than 0,1 mm. When it is done, the value of the velocity is obtained and the water flow rate can be deduced with the equation 7.

The results are presented in figure 7. The flows obtained via calculation are compared with the tests in real conditions that team 2 conducted with porous tubes. Following the calculations, the water flow is higher when the tube is located lower. The cause is that the difference of height accelerates the fluid, more than the effect of slowing down because there is more friction (the length of the pipes being higher). It is interesting to notice that the calculations and the test have the same order of magnitude, this would indicate that the calculations were well done. The flow in the upper pipe is 0,120 l/s, in the middle pipe it is 0,124 l/s and in the bottom pipe it is 0,128 l/s. Finally, the average flow calculated is **0,124 l/s**, and has only a difference of $7{\cdot}10^{-3}$ l/s with the mean flow that team 2 obtained.

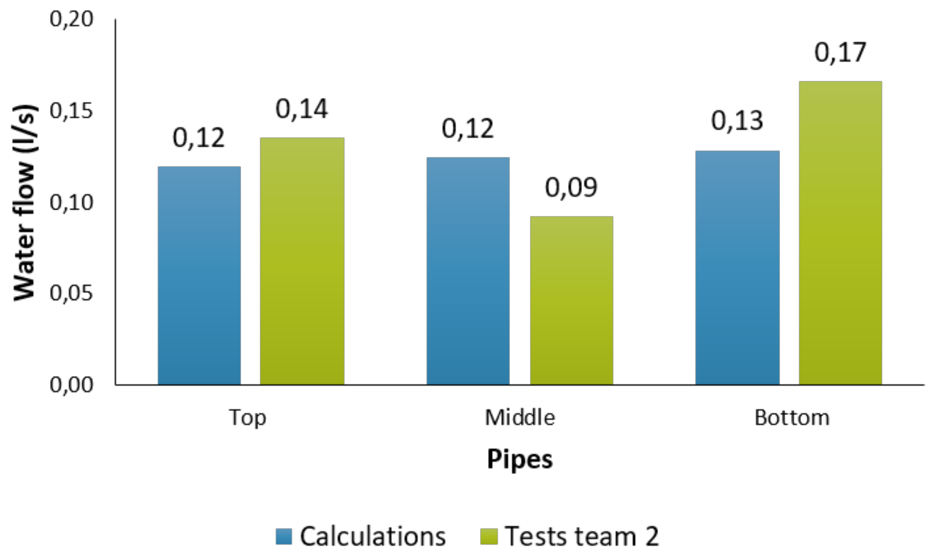



Figure 7: **Water flows for the three pipes according to calculations and to tests in real conditions**

The calculations have some limitations. First of all the pressure in the tank will not be constant since the water height will evolve with the rainwater and the volume taken for irrigation. In practice the water height in the reservoir should be controlled and if the volume is less than 800 liters then the calculations should be adapted in regard of the water height observed. The calculations for a water height of 35 cm have been made to see if the difference with the flow for a 75 cm height was significant. The difference of water flow in the upper pipe is $1{,}5{\cdot}10^{-3}$ l/s, that is really negligible.
Then, it is very difficult to know how the water distributes in the three pipes without doing some test in real conditions. For example, team 2 observed a smaller flow in the middle pipe. Therefore the mean flow should be take into account but it is necessary to remain very cautious when comparing it in the three irrigation pipes.
Concerning the length of the pipes, the diameters and the height of the different elements may contain some small errors because the entire hydraulic system could not be installed. Therefore not all measurements was made and some data are coming from hypothesis.
Furthermore, the singular friction forces are very difficult to estimate at the green wall level. Especially concerning the junctions between the three pipes and in the drippers selves. No data was found concerning these issues. The drippers and the junctions are therefore approximated the drippers by a

pipe separation. The difference of flow calculated for the upper pipe if we put 1 or 10 pipe separations is of 0,027 l/s and that does not change the order of magnitude of the flows. So the impact of this approximation is not major.
Finally, it could be interesting to compare in real situation which system have a greater flow between drippers and porous tubes.

### 2.4.3 Dripping system

The irrigation system in the wall consists of three irrigation pipes of approximately 2 meters long. It is *GARDENA Micro-Drip* pipes. They present a dripper every 10 cm. There are thus 20 drippers per pipe. Since they will be maximum eight plants per length of the wall it ensures that there will be a dripper not far from every plant.

It is advised to evaluate the uniformity of a mirco-drip irrigation system. In order to do that, it is necessary to measure the flow at the exit of some drippers of the wall. It should be done for at least 18 drippers and it would be convenient to measure the flow of seven drippers per pipe at different locations. The equation 14 is then used to calculate the uniformity. The aim is to have an uniformity near 90 %. Unfortunately these measurements could not be carried out because the hydraulic system was never properly installed.

$$U = \frac{F_{low}}{F_{moy}} \tag{14}$$

Where :

- $U$ is the uniformity $[-]$
- $F_{low}$ is the flow of the bottom quarter of the drippers $[m^3/s]$
- $F_{moy}$ is the mean flow of the drippers $[m^3/s]$

### 2.4.4 Water distribution in the wall

It is essential to have an idea of the water distribution in the wall because it conditions the distribution of plants and the interpretation of VWC measurements. However its estimation is complex because it depends on multiple factors: the location of the micro-drip pipes, the irrigation flow rate and frequency, the weather, the substrate, the bondary conditions*etc.* Hydrus software is often very useful for this kind of modeling.. An attempt was made with *Hydrus 1D* but unfortunately no results could be obtained because of the lack of information about the substrate hydraulic properties. An alternative would have been the use of *Hydrus 2D* but this softwarer require specific licenses.

A sketch of the probable water distribution was made in order to decide the location of the plants (figure 8). Two hypotheses are made to construct this graph :

- There are four input of water : rainfall at the top and the three pipes. At these locations the water content will therefore increase.
- The water content increases with depth because of gravity. Since the irrigation frequency is of 24 hours the water will have the time to flow down the wall. In a study of the impact of green walls on buildings in oceanic climate [13], this observation was clearly highlighted.

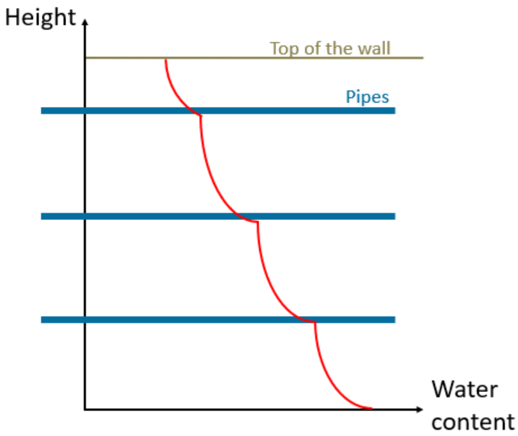



Figure 8: **Estimation the water content evolution in the wall**

## 2.5 Wall construction

### 2.5.1 In practice

The wall was built and filled on March 5 following the plan of Figure 3. First the module was placed on some concrete bricks so that the wooden structure is not in direct contact with the ground (Figure 9 a). Then it was filled with the substrate in layers of 10 centimeters. Moderate compaction was achieved using a broom after each layer in order to fit the experimental conditions and to prevent an excessive settling of the substrate (Figure 9 b).

((a)) Raising of the modules

((b)) Filling of the modules

Figure 9: **Construction of the walls**

Concerning the irrigation pipes and the humidity sensors, their emplacement is shown in Figure 2.

In a previous use of the walls, holes had already been drilled 37, 67 and 97 cm from the bottom to pass irrigation pipes. These holes were reused to place the drip irrigation pipes. The volume under each pipe can then be considered as a soil layer. The bottom layer is a larger, but it probably won't dry out because of the downward movement of water due to gravity. At the top of the wall there is a 13 cm thick layer above which there is no irrigation pipe. Indeed rainfall is expected to be sufficient to water this layer. In addition, thyme will be planted at this level. As mentioned before, it is a plant that tolerates dry conditions well. It therefore reduces the impact of periods without rain. The pipes are located in the middle of the wall (Figure 10 a) and all three are connected outside (Figure 10 b).

Concerning the humidity sensors, one was placed in each of the three layers. It is important to place the sensors in the driest part of the layers the most unfavourable conditions for plant growth. The driest parts are expected to be a the top of each layer because water should quickly go down with gravity (Figure 8). The volume of influence of the humidity probes extends over a radius of four centimeters around the probe. So they were placed at 5 centimeters under the pipes. The orientation of the probes complied with the manufacturer's recommendations. The thickness of the wall is very low ( 13 centimeters) so the probes were placed halfway between the metal plate and the back panel. The sensors have threads coming out of the module and connected to the AKUINO central (Figure 10 c).

((a)) Installation of the pipes in the wall. ((b)) Pipes connection. ((c)) Probes threads.

Figure 10: **Pipes and probes installation in the module.**

### 2.5.2 Further reflections on the wall design

It was initially envisaged to place a waterproof texile between the layers in the wall, in order to limit the movement of water between the layers, an therefore increase water distribution homogeneity. In the end, this was considered unhelpful since the non-uniformity of water distribution was taken into account in the choice of plants and their placement in the wall. For example, fern-type plants were placed at the bottom of the wall, where the water content is assumed to be highest.

The use of a geotextile can also help to contain root development, prevent their passage from one layer to another, and thus reduce competition between plants. This can also be used to protect pipes from clogging by roots. Once again, this system was not retained. First reason is that the irrigation system is not porous tubes but a drip system so the risk of clogging is lower. The passage of roots from one layer to another is also seen as an opportunity for better use of water resources, while ensuring that the planting density is not too high.

Another concern was to ensure that the substrate does not become saturated so as not to impact root respiration. Highly draining layers are sometimes inserted to ensure the percolation of excess water into the green walls. In the case studied here, no measures were finally taken given the highly infiltrating nature of the substrate used. In addition, water cannot accumulate at the bottom of the wall because it is pierced with numerous holes that allow excess water to drain away.

Finally, the impact of, the heating of the metal plate on the water distribution was studied. Indeed it can be important given that the wall is south-facing. The placement of an organic mulch to isolate the substrate from the plate was considered. However, due to the thinness of the wall, this is difficult to install. Moreover this solution is not durable. This problem should be studied in more detail. One avenue could be to use materials that are more durable than mulch, such as wood.

### 2.6 Coronavirus change : pots instead of walls

Due to the coronavirus crisis, the experiment on the walls had to be stopped.. Even though their construction had begun, it didn't serve in the end. Instead, it was decided that the project would consist of irrigating plants in a pot, putting the same sensors in the pots as there would have been in the walls. The pots were located on a balcony in Mont-Saint-Guibert. The weather station was placed there as well. A scheme of the new system is presented in Figure 11.

As shown on the figure 12, this was actually very different from the walls that were supposed to be used. This means that the equations used in the code need to be adapted (for instance $ET_0$ calculation) before being applied to an actual green wall.

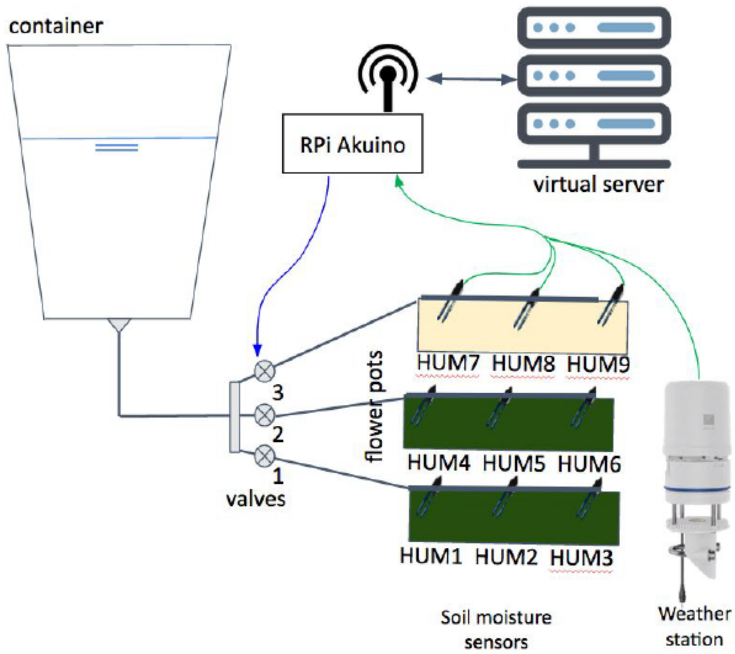



Figure 11: **New irrigated system.** Source: Christophe Dupriez.

Figure 12: **Pots used instead of the walls.** Source: Christophe Dupriez.

# 3 System characterisation and monitoring

## 3.1 Humidity sensors

### 3.1.1 Sensors monitoring and calibration

Soil volumetric water content (VWC) was measured with three EC-5 sensors (DECAGON®). In this work, they will be called HUM7, HUM8 and HUM9. Those sensors are capacity probes that measure VWC by measuring the dielectric permittivity ($\epsilon$) of the bulk soil([11]). They have an accuracy of 0.03 $cm^3/cm^3$. The placement of the sensor is shown in Figure 13.

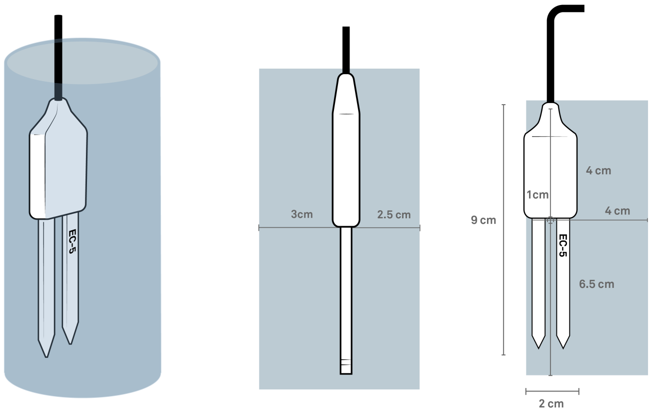



Figure 13: **Scheme of an EC5 probe.** Distances indicate the volume of influence of the sensor.

First of all, EC5 probes were calibrated in the wall substrate. Indeed, these are low-cost sensors that show considerable sensor-to-sensor variability [6]. If not appropriately accounted for, it can affect measurements accuracy. The calibration of the three sensors took place in the soil physics lab in Gembloux. The data were collected with a CR800 datalogger (CAMPBELL®). The sensors were powered with a 2.5 V power supply. First, the substrate was placed in an oven at 50 $^oC$ during 5 days, in order to dry it entirely. To reach a precise volumetric water content, a certain mass of water was added to a constant mass of dry soil. After mixing the soil, the samples were inserted in a cylindrical container of 14 cm in diameter and 25 cm in height. In the cylinder, the samples were eventually compacted to a certain volume to achieve a precise density. Indeed soil density affects measurements. The sensors were inserted vertically in the center of the container in order to keep the volume of influence of the probe undisturbed. Five volumetric water content were tested : 0 (dry soil), 15, 30, 45 and 100 (pure water) $cm^3/cm^3$. However, when the soil was completely dry (SWC of 0 %) the probes did not provide any measurement. This is why only four points were used to draw the calibration curves (Figure 14).

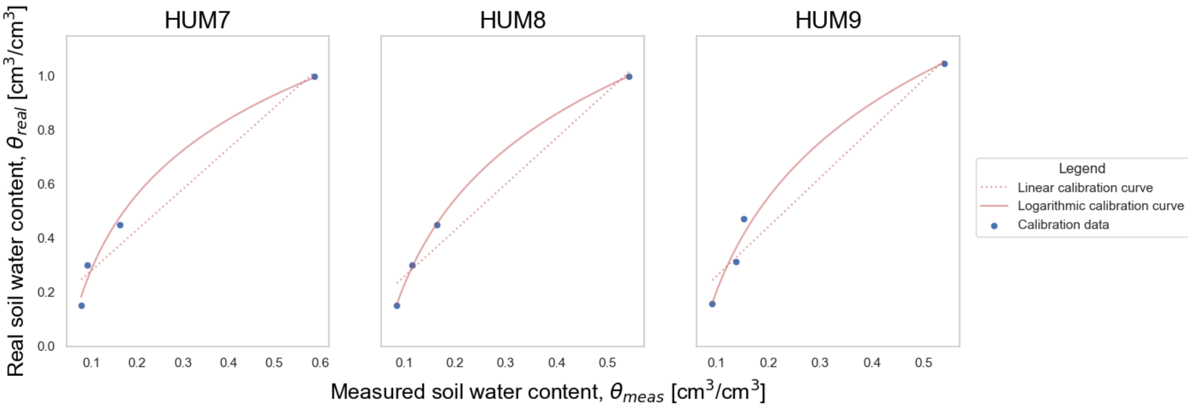



Figure 14: **Calibration curves of EC5 humidity sensors in the wall substrate.**

The curve that best fitted to data was found by the least-squares method. In the EC5 manual guide[1], a linear equation is proposed to convert sensor outputs to water content in potting soil. However Figure 14 shows that a logarithmic model fits the data better. Table 2 contains the results of the least-squares fit for the two types of models.

Table 2: **EC5 sensors calibration curves characteristics.** $\theta_{real}$ and $\theta_{meas}$ refer to real and measured volumetric water content, respectively.

| Sensor name | Linear model | $r^2$ | Logarithmic model | $r^2$ |
|---|---|---|---|---|
| HUM7 | $\theta_{real} = 1.5076 \cdot \theta_{meas} + 0.1289$ | 0.9602 | $\theta_{real} = 0.4019 \cdot ln(\theta_{meas}) + 1.2082$ | 0.988 |
| HUM8 | $\theta_{real} = 1.7106 \cdot \theta_{meas} + 0.0897$ | 0.9661 | $\theta_{real} = 0.457 \cdot ln(\theta_{meas}) + 1.2789$ | 0.9997 |
| HUM9 | $\theta_{real} = 1.7416 \cdot \theta_{meas} + 0.0741$ | 0.9519 | $\theta_{real} = 0.4808 \cdot ln(\theta_{meas}) + 1.3012$ | 0.9867 |

In pots, the composition of the substrate differs from that of the greenwall. In addition, the sensors are connected to a 5 V power supply, while the were calibrated for a 2.5 V power supply. The manufacturer provides a general quadratic equation to convert EC5 output (in Volts) into VWC when sensors are connected to a 5 V power supply[2]. However in the case of the pots, these equations gave aberrant results. As a last resort, Cédric Bernard's work was used to transform the sensor signals into VWC. Cédric Bernard used the commercial substrate 'Rockery Type Plants-Light' from Zinco. This substrate is composed of a mix of crushed brick, mineral aggregates, substrate compost and fibre materials. When the EC5s were calibrated, they were connected to 5V. Equation 15 is the calibration equation found by Cédric Bernard, that convert the raw signal ($V$, in volts), into SWC $\theta$.

$$\theta = \frac{0.3524 * V - 0.1544}{V - 0.3747} \qquad [cm^3/cm^3] \tag{15}$$

Figure 15 shows that the fitted quadratic model gives poorer results, compared to the equation found by Cédric Bernard. Cédric Bernard's calibration equation is probably incorrect in the case of the greenwall substrate, which differs slightly from the one used in his experiment. Other parameters, such as density, are most likely also different. Nevertheless, this equation was used, in order to communicate the results in explicit units (cm$^3$/cm$^3$) rather than in an analog signal (volts) which is difficult to interpret.

[1]METER. 2012. "EC-5 Manual Guide."
[2]LICOR. 2015. "Connecting and Programming the EC-5."

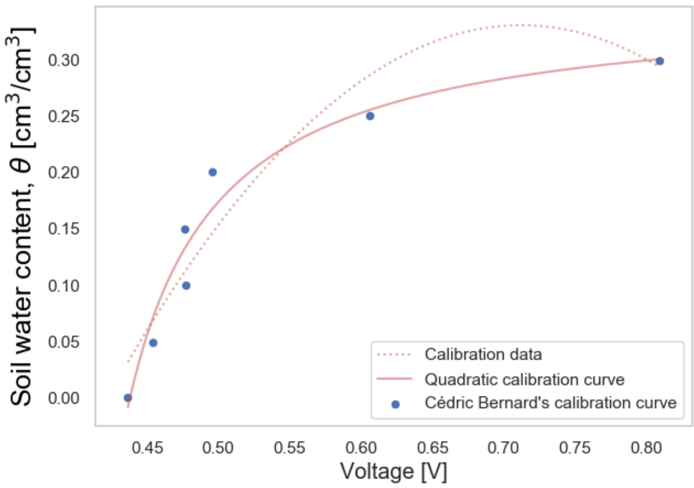



Figure 15: **Calibration curves of EC5 humidity sensors, in zinco substrate, for a power supply of 5 V.** Calibration data are those of Cédric Bernard.

Once the probes are calibrated, they are placed in the wall. The probes are inserted in holes in the wall at different heights (32, 57 and 82 cm with respect to the wall bottom). EC-5s are inserted vertically into the soil with the flat side placed at $90^o$ with respect to the ground surface. It allows to limit the accumulation of water on the probe's faces.

### 3.1.2 Temperature effects

EC5 sensor's outputs are sensitive to temperature [9]. According to the manufacturer (DECAGON®), this temperature sensitivity is not caused by the sensors themselves, but rather the electrical properties of the soil. Particularly, dielectric permittivity can be quite sensitive to temperature changes. Temperature compensation can be used to improve the performance of the EC5 probes to minimize the effects of substrate temperature on probe output [22]. Nemali al. (2007) found a correction coefficient of 0,003 V/°C [22]. However in the Application Note "Correcting temperature sensitivity of ECH2O soil moisture sensors", provided by the manufacturer [9], it is shown that the interaction between temperature and sensor output is much more complex. It highly depends on soil constituents and solute content. Indeed, the dielectric permitivitty of a soil is a complex quantity with real and imaginary components. The real component is the dielectric permittivity of the soil constituents and has a negative correlation with temperature. The imaginary component is related to dielectric losses and electrical conduction through the soil and has a positive correlation with temperature. In a soil, the opposing temperature sensitivities of the real and imaginary components of the dielectric permittivity can be seen as two opposing forces. In some soils, the real component takes over. Thus when the temperature increases it leads to a decrease in the VWC measured by the sensor. In other soils however, when an imaginary component dominates, an increase in temperature leads to an increase in the VWC. Eventually in some soils, the two components closely balance each other and there is no apparent temperature sensitivity in the VWC measurement. Therefore it is impossible to determine a generic correction factor for temperature that can be applied to all soils [9]. There are two different strategies to take the influence of temperature into account. If soil temperature is measured at the same location as the EC5 sensor, then a multiple regression strategy can be used to relate the true VWC to the measured VWC and temperature data. Another strategy must be used when no data are available. Both methods are described in details in the application note cited earlier in this section. In this project the data-based strategy was used.

The aim of the multilinear regression is to build a mathematical model as in Equation 16.

$$VWC_{corrected} = C_1 \cdot VWC_{meas} + C_2 \cdot T_{soil} + C_3 \qquad [cm^3/cm^3] \tag{16}$$

With:

- $VWC_{meas}$ : VWC measured by the EC5 sensor [$cm^3/cm^3$].
- $T_{soil}$ : soil temperature at the location of the sensor [°C].
- $C_1$, $C_2$, $C_3$: empirical coefficients determined by multiple regression on collected data.

In this project, no sensors were provided to measure soil temperature. It was therefore necessary to find a proxy for this variable. It was hypothesized that soil temperature is strongly correlated with air temperature which is measured by the weather station ATMOS41. In the future of this project it would be a good idea to place some temperature sensors in the wall itself. Some parrot sensors can be used for example.

First, the analysis requires the identification of three or more 24 hours periods. These periods must satisfy certain conditions : no precipitation and no irrigation in each 24 hours period, temperatures at the beginning and at the end are comparable, no anomalies in the data selected. The period chosen runs from April 2 to April 4 2020. During this period neither precipitation nor irrigation occurred. Secondly the hypothesis of correlation between the VWC and air temperature must be confirmed. The Pearson's correlation coefficient $r^2$, and the associated p-value, was calculated for each probe, using the Python 3.7 `pearsonr` function of the `scipy.stats` module, on data collected between April 2 and April 4. Then, VWC data had to be interpolated between endpoints of each 24 hours period. This was performed with Excel® 2016 software. To determine the real (non affected by temperature) SWC of the soil, drainage/evaporation are assumed constant over each period. Therefore VWC is interpolated as a straight line between couple of endpoints. The resulting linear equation was applied in every points between endpoints. This process had to be repeated for each 24 hour period. Finally, all 24 hours periods were combined and a multiple regression was performed with Excel. The operation was conducted for the three sensors.

## 3.2 Weather station

The weather station used for this project is the ATMOS41. This station permits to measure different parameters : solar radiation, precipitation, air temperature, barometric pressure, vapor pressure, relative humidity, wind speed, wind direction, maximum wind gust, lightning strikes, lightning distance, tilt. All sensors are integrated into a single, small form-factor unit. A scheme of the weather station used for the project is shown in Figure 16.

Solar radiation is measured by a pyranometer. The pyranometer used a silicon-cell sensor to measure the total incoming (direct and diffuse) solar radiation. The pyranometer is factory calibrated and the calibration value can be found on the interior of ATMOS41. This factor has already been added into the ATMOS41 so there is no need to do anything with it. The solar radiation is measured every 10 seconds and values are recorded instantaneous.

The anemometer measures wind speed. The space underneath the rain gauge is where the ATMOS41 measures wind speed.The ATMOS41 measures wind speed every 10 sec and keeps a running average of the last 10 measurements. If a measure is more than eight times the average, the measure is rejected.

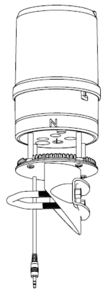
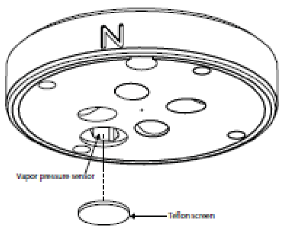


Figure 16: **Scheme of the weather station**

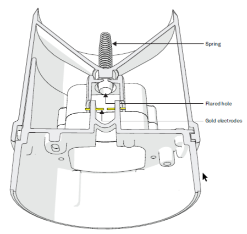
The pressure vapor sensor is located behind a teflon screen. This screen protects the sensor from liquid water and dust while allowing the water vapor to pass to the sensor. The sensor measures relative humidity, temperature and the vapor pressure. The location of the vapor sensor is presented in Figure 17.

The rain gauge measures the rainfall. During a rain event, the flared hole forms the rain into drops. These drops pass then by a drip counter. Because the flared hole forms a drop of a known size, the ATMOS41 counts the drops and calculate the water volume. The ATMOS41 counts the drop continuously and adds each drop to an accumulated total.

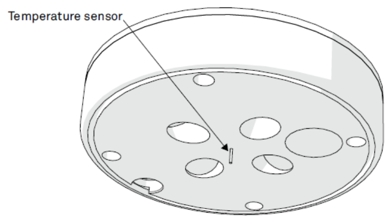
The measurements of the temperature is made in the center of the anemometer area thanks to a tiny temperature sensor called a thermistor. The station measures the air temperature with a high accuracy because solar radiation and wind speed are known. In fact, these two parameters determine the errors between measured air temperature and the actual air temperature. The ATMOS41 measures the air temperature once every 10 sec and records the instantaneous values.




((a)) Scheme of the vapor sensor




((b)) Scheme of rain gauge




((c)) Scheme of temperature sensor

Figure 17: **Schemes of the main sensors of the weather station**

# 4 Irrigation strategy

As mentioned previously, ensuring uniformity of water distribution throughout the wall is the main challenge. Water loss through drainage must also be avoided. Both originate from the downward movement of water in the wall under the effect of gravity. The drip system makes it possible to minimize water losses by frequently irrigating small quantities of water. Thresholds below which the VWC must not pass have been previously identified. In practice the objective is to keep the VWC away from these thresholds, trying to stay close to the VWC at field capacity. Moreover it was shown that if the substrate is irrigated while still wet, the uniformity is greater [28].

Second, additional challenges arise when irrigation is automated. Since the system is supposed to work autonomously, the program should be as resilient as possible. Whatever the situation or problem encountered, it must continue to supply water to the wall. The entire program (`wat3.py`) can be found in Appendix B. As a reminder, it is written in Python 2.7. All graphs presented in this report were built in the Spyder environment (Python 3.7).

## 4.1 Irrigation program

### 4.1.1 Global Scheme

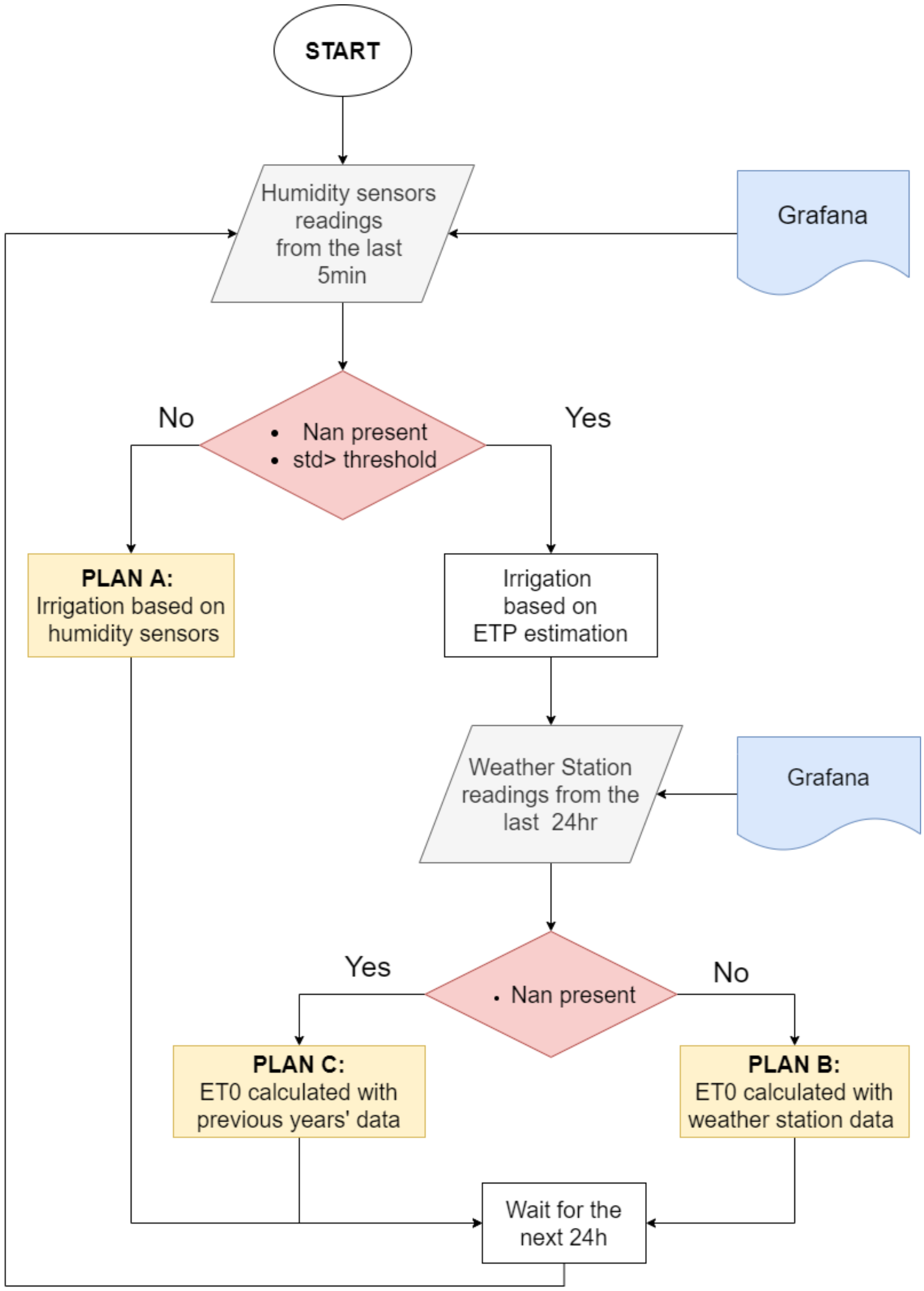



Figure 18: **Flowchart: Global Scheme**

It is important to remind that this irrigation plan was first built for the pots in Mont-Saint-Guibert. Modifications were then made so that it could also be applied to the greenwall. At the beginning of the program the user can specify which type of system he wants to irrigate ('pot' or 'greenwall'). If there is an adaptation of the code to make to fit the wall situation it will be written explicitly.

Figure 18 shows the global scheme of the decision-making process. It includes three different irrigation strategies (Plan A, Plan B and Plan C). These 3 are as follows, in order of priority:

- Plan A: Irrigation based on humidity sensors.
- Plan B: Irrigation based on weather station data.
- Plan C: Irrigation based on previous years' weather data.

Plan A is first in order of priority. It uses the VWC measurements from the humidity sensors to calculate the valve opening time. If the sensors are properly calibrated and the soil retention properties (VWC at field capacity and wilting point) are well characterized, this strategy is considered the most accurate. However, humidity sensors may encounter problems that make them unusable for a certain period of time. In this case, water needs must be estimated by other means. That is why Plan B and Plan C were constructed. Plan B establishes the water balance of the system to calculate the amount of water needed. It uses atmospheric variables, such as solar radiation, air temperature or vapor pressure measured by the weather station, to estimate water losses in the system by evapotranspiration (drainage is neglected). Rainfall is considered as the only water input. The water needs corresponds to the difference between ET (calculated with the Penman-Monteith equation) and rainfall. Plan C was design to address the possibility of a malfunction of the weather station. It uses atmospheric data from previous years to estimate the average evapotranspiration each day. It is the last resort. As it does not rely on sensors, it ensures a supply of water, even if all sensors are out of order.

The data are collected on a by the AKUINO central with a time step of one minute.

The irrigation time step is set to 24 hours. The main reason for this is the uncertainty about the time it takes for the valve to change from a fully closed state to a fully open state when irrigation is switched on, and from a fully open state to a fully closed state when irrigation is switched off. The duration of each of these processes is approximately 1 minute. The total absolute error on the valve opening and closing time is therefore 2 minutes. In the case of the pot, the water supply provides a flow rate of 1 l/h. If in 24 hours the volumetric water content of the soil in the pot decreases by 2 %, this is equivalent to a water loss of 0.460 l. To offset this loss, the valve must be kept open for 27.6 minutes. With an absolute error of 2 minutes, the relative error amounts to 7.2 %. This error is not negligible. If the irrigation time step is decreased, the absolute error will increase almost proportionally. Secondly, the estimation of the reference evapotranspiration ($ET_0$) by the FAO Penman-Monteith method on 24-hour time scale has been shown to give accurate results.

In the global scheme of the irrigation program (Figure 18), first step is to assess whether or not Plan A can be launched. First readings from the three humidity sensors over the last five minutes are collected via Grafana. In Plan A irrigation doses are calculated based on the mean SWC computed over 5 minutes for each sensor. Data from humidity sensors are considered usable if they meet two conditions. The first is the absence of NaN value. NaN values can be a sign of sensor malfunction or power supply problems. There are detected with the `isnan` function of the `math` module. The second is the limitation of the dispersion of the set of values for each sensor. Indeed, considering a mean VWC value implies that VWC is considered stationary over 5 minutes, or at least negligible compare to humidity sensors' uncertainty. As a reminder, the uncertainty of humidity sensors is 0.03 $cm^3/cm^3$.

Therefore, the standard deviation of each 5-minutes data series must be lower than this uncertainty. If the standard deviation is greater than this threshold, it is interpreted as an anomaly. Standard deviation $\sigma$ of each humidity sensor is computed as in Equation 17. The program runs Plan A if at least two of the three sensors meet these two conditions.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \qquad [cm^3/cm^3] \tag{17}$$

With:

- N: number of readings for a period of 5 minutes (5 whit data acquisition time step of 5 minutes).
- $x_i$: VWC reading of a sensor $[cm^3/cm^3]$.
- $\bar{x}$: mean VWC of a sensor $[cm^3/cm^3]$.

To test the hypothesis of a variation in water content lower than the accuracy of the humidity sensors for a period of 5 minutes, the water balance every 5 minutes will be calculated using data collected on the irrigated system. Water inputs correspond to rainfall and irrigation. The only water loss considered is evapotranspiration (drainage is neglected). Evapotranspiration (ET) is estimated with the FAO Penman-Monteith method on 5-minute time scale [44]. The use of this equation is explained in details in the description of Plan B, further in this report. Reference evapotransiration ($\mathrm{ET}_0$) is considered, as it is often the maximum value of ET encountered. The resulting variation of VWC $d\theta_{5min}$ for a period of 5 minutes is estimated using Equation 18.

$$d\theta_{5min} = \frac{R_{5min} + I_{5min} - ET_{0,5min}}{H} \qquad [cm^3/cm^3/5min] \tag{18}$$

With:

- $\mathrm{R}_{5min}$: rainfall depth for a period of 5 minutes $[cm/5min]$.
- $\mathrm{I}_{5min}$: irrigation depth for a period of 5 minutes $[cm/5min]$.
- $\mathrm{ET}_{0,5min}$: $\mathrm{ET}_0$ for a period of 5 minutes $[cm/5min]$.
- H: pot height $[cm]$.

If plan A can not be used (at least two humidity sensors do not meet conditions cited previously), Plan B will be considered (Figure 18). As a reminder, plan B uses meteorological data measured by the weather station to estimate the water balance of the system to be irrigated. To do that, data recorded by the weather station over the last 24 hours must be collected (solar radiation, air temperature, wind speed, vapor pressure, atmospheric pressure, rainfall). As the possibility of a malfunction of the atmospheric station cannot be excluded, it is necessary to evaluate the quality of the data collected. In Plan B only the absence of NaN values in data series will be checked, with the same method as in Plan A.

If the data needed to run Plan B is considered unusable, Plan C will be launched as a last resort. Then, regardless of the path taken, the program will wait for 24 hours before starting again.

### 4.1.2 Description of plan A: Irrigation based on humidity sensors.

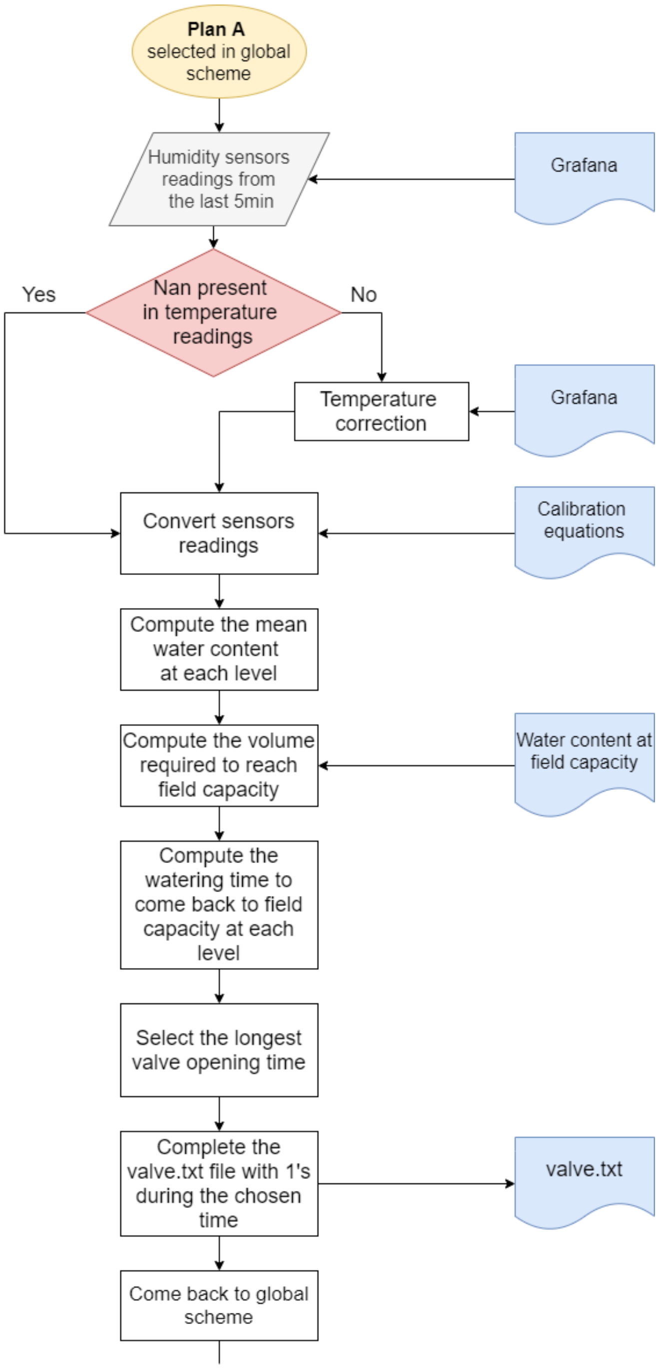



Figure 19: **Plan A: Irrigation based on humidity sensors**

Plan A is coded between lines 126 and 508 of the irrigation program (Appendix B). The program starts by collecting the measurements taken during the last 5 minutes by the three EC5 probes (HUM7, HUM8, HUM9) as well as the weather station sensor that measures air temperature (SDI7). This data is taken on Grafana, using the function `urlopen` of the module `urllib`, that allows to open a network object denoted by a URL (Grafana website) for reading.

First, the raw analog signal from the EC5 probes [Volts] must be converted into volumetric water content (VWC) [$cm^3/cm^3$] with the appropriate calibration equation. A function called `calib` was created to perform the conversion. The first argument is the humidity sensor readings array. The second specifies which calibration equation the user wants to use. Several calibration equations exists (generic calibration equation provided by the manufacturer, Cédric Bernard's calibration equation...). As mentioned previously, in this work it is the calibration equation of Cédric Bernard that will be used.

Secondly, humidity sensor temperature sensitivity is corrected. A function was created, that converts the uncorrected VWC into corrected VWC using equations found by multilinear regressions. The first argument is the uncorrected VWC array. A second argument specifies the name of the sensor considered (HUM7, HUM8 or HUM9). Indeed, coefficients of the multilinear regression differ between humidity sensors. However, if they are NaN values among air temperature readings, the correction is not made. This is to ensure that the code does not crash.

To calculate the water needs, the volume of the pot is divided into three zones, corresponding to the locations of the three humidity sensors. In each zone (index $i$), water needs are evaluated as the difference between the mean water content over the last five minutes ($\overline{\theta_i}$) and water content at field capacity ($\theta_{FC,i}$). As no indications were provided concerning retention properties of the pot substrate, field capacity was assessed using data collected. Automated processing algorhitms exist for estimating field capacity from VWC time series [5]. However such methods were not used in this work, as available data do not meet the conditions of application, and because they represent a substantial amount of work. Instead, the VWC at field capability in each zone was determined graphically, based on its definition, which is "the water quantity which a certain, initially saturated soil is still able to hold against gravity after 2–3 days" [39]. A huge amount of water were brought to the pot to saturate the soil on March 28. After three days where no irrigation and no rainfall occured, SWC at field capacity in each zone was assessed (Figure 20). Table 3 contains the obtained values.

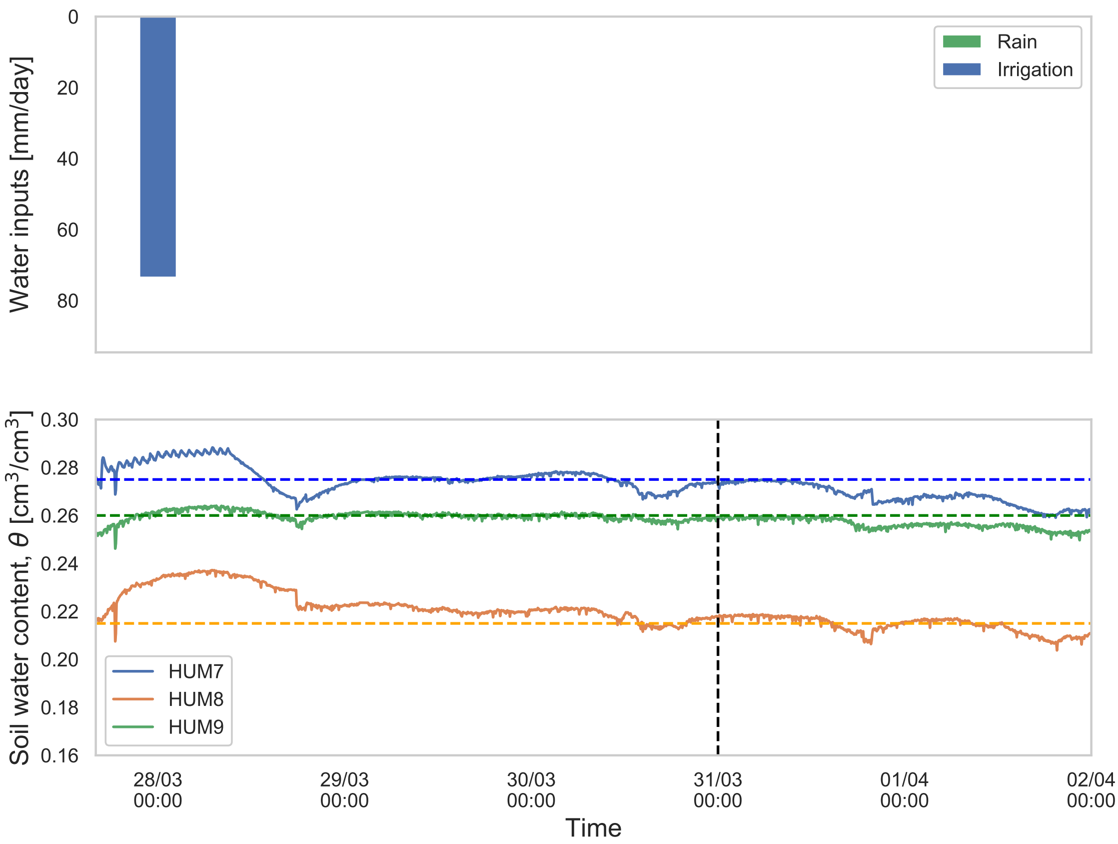



Figure 20: **Graphical evaluation of the volumetric water content at field capacity in the three zones of the pot**. HUM7, HUM8 and HUM9 refers to VWC in the three zones of the pot. The corresponding dotted lines represents VWC at field capacity.

Table 3: **Volumetric water content at field capacity in the three zones of the pot.**

| Sensor | HUM7 | HUM8 | HUM9 |
|---|---|---|---|
| $\theta_{FC}$ $[cm^3/cm^3]$ | 0.275 | 0.215 | 0.26 |

Given the dimensions of the pot (area: 1920 cm$^2$, height: 12 cm), the irrigation volume V$_i$ required to come back to field capacity in each zone can be computed by Equation 19.

$$V_i = (\theta_{FC,i} - \overline{\theta_i}) \cdot A_i \cdot H \qquad [cm^3] \tag{19}$$

With:

- $\theta_{FC(i)}$: VWC at field capacity in zone $i$ $[cm^3/cm^3]$.
- $\overline{\theta_{(i)}}$: mean VWC over the last five minute for in zone $i$ $[cm^3/cm^3]$.
- $A_i$: zone area (a third of the pot area) $[cm^2]$.
- $H$: the pot height $[cm]$.

In the case where the mean VWC is greater than VWC at field capacity, the irrigation volume is set to zero. Given the discharge of the irrigation pipe (1000 $cm^3/hr$), the valve opening $t_i$ necessary to come back to field capacity in each zone is computed with Equation 20.

$$t_i = \frac{V_i}{Q_i} \cdot 3600 \qquad [s] \tag{20}$$

With:

- $V_i$: irrigation volume necessary to come back to field capacity in each zone $[cm^3]$.
- $Q_i$: discharge in each zone (a third of the total discharge) $[cm^3/hr]$.

As the irrigation volume necessary to come back to field capacity is not identical in the three zones, the longest valve opening time must be selected to ensure that field capacity is reached in each zone. It reduces the risk of generating water stress in areas requiring more water, but this is likely to cause water loss through drainage in areas with lower water requirements. However, as a reminder, Plan A can be run even if one sensor is deemed unusable. If this is the case, the selection is only made between the two valid sensors. The instructions are transmitted to the valve by adding two lines to the file `valve.txt`, one with the opening time (now + 30 seconds to take into account a possible processing time) and the instruction "1" to trigger the opening of the valve, and the other with the closing time (now + 30 seconds + irrigation time) and the instruction "0" to close the valve. When writing the first line (opening of the valve command), the `valve.txt` file is open using the argument 'w', to erase the current file that had been sent the day before. To keep track of the fact that irrigation was provided by Plan A, a file called `history.txt` is also completed with the current time and the letter "A".

**GREEN WALL ADAPTATION**

The calculation of the irrigation time must take into account different parameter values in the case of the green wall. First the three zones (layers) have different dimensions in the green wall, while in the pot they are considered perfectly equivalent.

Secondly, the discharge vary between layers. The formula used is the same as for the pot (20). In this case, the choice of the longest irrigation time to irrigate the entire wall is all the more important, as the changes mentioned above are likely to result in greater heterogeneity in the water content values.

### 4.1.3 Description of plan B: Irrigation based on weather station data.

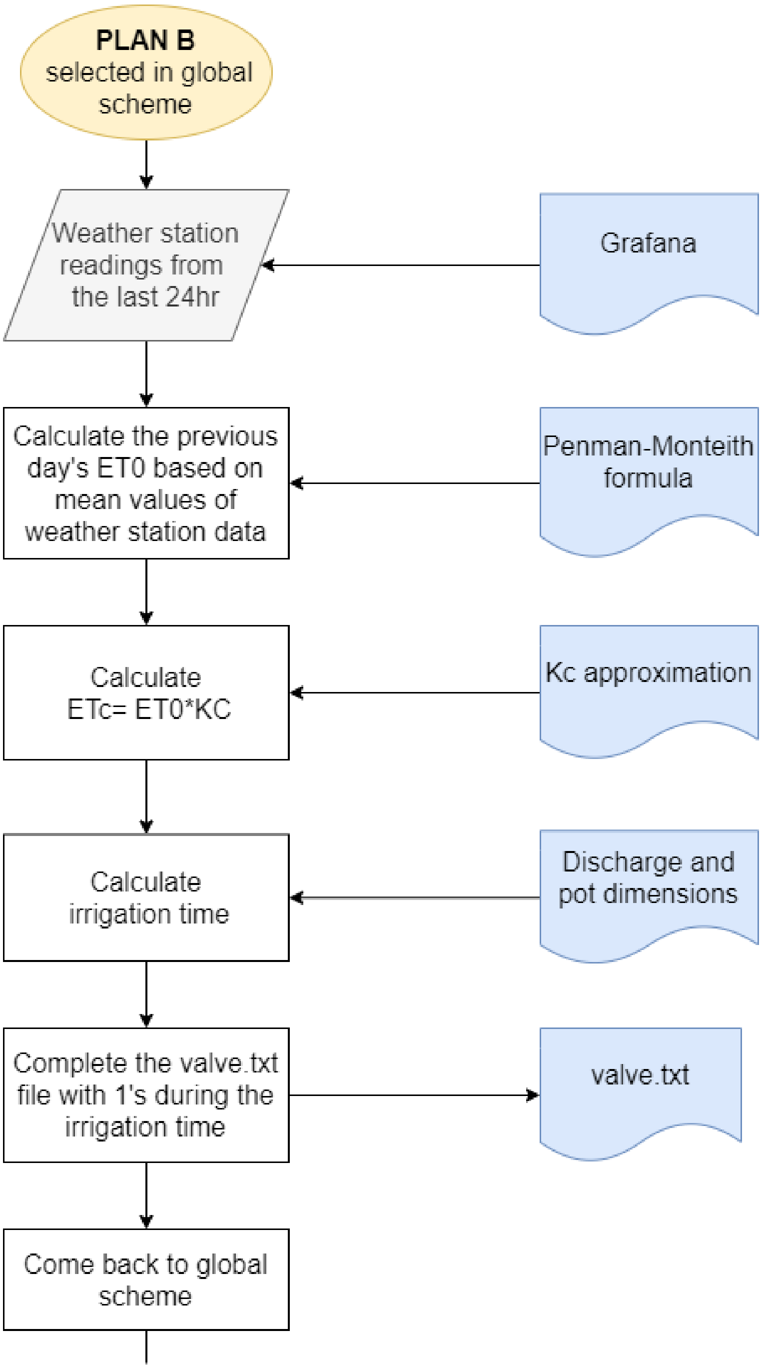



Figure 21: **Plan B: Irrigation based on weather station data**

Plan B is coded between lines 510 and 882 of the irrigation program (Appendix B). Plan B begins with the reading of the measurements taken by the weather station over the last 24 hours. Six sensors are necessary to establish the water balance of the system to irrigate (Table 4). This data is taken on Grafana website as in Plan A (function `urlopen` of the module `urllib`).

Table 4: **Sensors used in Plan B.**

| Name | Variable measured | Units |
|---|---|---|
| SDI0 | Solar radiation | W/m$^2$ |
| SDI1 | Rainfall | mm/h |
| SDI4 | Wind speed | m/s |
| SDI7 | Air temperature | °C |
| SDI8 | Vapor pressure | kPa |
| SDI9 | Atmospheric pressure | kPa |

In Plan B, two components of the water balance must be calculated. On the one hand, water inputs are computed by summing the rainfall of the last 24 hours. On the other hand, water losses are evaluated by calculating the amount of water evapotranspired during the last 24 hours, using the FAO Penman-Monteith formula (Equation 21) in order to calculate the reference evapotranspiration [42].

$$ET_0 = \frac{0.408\,\Delta\,(R_n - G) + \gamma\,\frac{900}{T+273}u\,(e_s - e_a)}{\Delta + \gamma\,(1 + 0.34\,u)} \qquad [mm] \tag{21}$$

With:

- $ET_0$: the reference evapotranspiration $[mm/day]$.
- $R_n$: the net radiation at the crop surface $[MJ/m^2/day]$.
- $G$: the soil heat flux density $[MJ/m^2/day]$.
- $T$: the air temperature at 2m height $[°C]$.
- $u$: the wind speed at 2m height $[m/s]$.
- $e_s$: the saturation vapour pressure $[kPa]$.
- $e_a$: the actual vapour pressure $[kPa]$.
- $\Delta$: the slope of saturation vapor pressure curve at air temperature T $[kPa/°C]$.
- $\gamma$: the psychrometric constant $[kPa/°C]$.

First, raw data must be converted into suitable variables for the Penman-Monteith equation. A first simplification can be made by assuming that the soil heat flux density ($G$) is equal to zero. Indeed, for a daily time step it is admitted that the quantity of heat infiltrating during the day is equal to the quantity of heat released at night. In addition, since the pot is not in continuity with the soil, conduction exchanges are probably very low.

Air temperature ($T$), wind speed ($u$) and vapor pressure ($e_a$) are averaged over 24 hours, as well as the atmospheric pressure ($P$) that is used to compute the psychrometric constant.

To convert the solar radiation measurements $[W/m^2]$ obtained with the SDI0 sensor into the net radiation at the crop surface $[MJ/m^2/day]$, the solar radiation values ($R_s$) obtained over one day are summed and then converted from $[W/m^2/day]$ to $[MJ/m^2/day]$ (Equation 22).

$$R_n = \frac{\sum (R_s \cdot 60)}{10^6} \qquad [MJ/m^2/day] \tag{22}$$

The saturation vapour pressure $e_s$ is not directly given by the weather station and must therefore be calculated with the Equation 23, using the mean air temperature $T$.

$$e_s = 0.6108 \cdot e^{\frac{17.27 \cdot T}{T+273.3}} \qquad [kPa] \tag{23}$$

The slope of saturation vapour pressure curve at air temperature $\Delta$ is obtained using Equation 24.

$$\Delta = \frac{4098 \cdot e_s}{(T+273.3)^2} \qquad [kPa/°C] \tag{24}$$

Finally, the psychrometric constant $\gamma$ is computed with Equation 25.

$$\gamma = \frac{C_p \cdot P}{\epsilon \cdot \lambda} = 0.665 \cdot 10^{-3} \cdot P \qquad [kPa/°C] \tag{25}$$

With:

- $P$: the mean atmospheric pressure over the last 24 hours $[kPa]$.
- $\lambda$: the latent heat of vaporisation ($\lambda = 2.45\ [MJ/kg]$).
- $C_p$: the specific heat at constant pressure ($C_p = 1.013 \cdot 10^{-3}\ [MJ/kg/°C]$).
- $\epsilon$: the ratio between molecular weight of water vapour and dry air ($\epsilon = 0.622$).

Once $\mathrm{ET}_0$ is known, the crop evapotranspiration $\mathrm{ET}_c$ can be computed using the crop coefficient $\mathrm{K}_c$ (equation 26). For salads, the FAO recommends using a Kc of 1 [45], however, as the rocket has been implanted as seed in the pots, this coefficient was reduced to 0.5 to take into account the lower transpiratory flux.

$$ET_c = K_c \cdot \frac{ET_0}{10} \qquad [mm/day] \tag{26}$$

Considering that the soil is at field capacity at the beginning of the experiment, the amount of water evapotranspired by the crop each day corresponds to the optimal daily water requirement for the development of that crop. Therefore, the amount of water to be irrigated each day is equal to the difference between $ET_c$ and the amount of water brought by rain over the last 24 hours. Knowing the flow rate and the dimensions of the pot it is possible to calculate the valve opening time needed to irrigate this precise volume of water. The irrigation time $t$ is given by Equation 27.

$$t = \frac{(ET_c - R) \cdot A}{Q} \cdot 3600 \qquad [sec] \tag{27}$$

With:

- $A$: the pot area $[mm^2]$.
- $ET_c$: the daily crop evapotranspiration $[mm/day]$.
- $R$: the daily rainfall $[mm/day]$.
- $Q$: the discharge in the irrigation pipe $[mm^3/hr]$.

In the same way as in Plan A, the instructions are transmitted to the valve by means of the file `valve.txt`. To keep track of the fact that irrigation was provided by Plan B, a file called `history.txt` is also completed, this time with the letter "B".

**GREEN WALL ADAPTATION**

As for plan A, some parameters will take different values depending on the case under consideration (pot or green wall). In the case of the green wall the cultural coefficient $K_c$ is 0.6 instead of 0.5 for the pot.
Again, the dimensions of the wall and the different discharges in each layer must be considered. In the case of the pot, the area influenced by precipitation is equal to the area undergoing losses by evapotranspiration. This is not the case for the greenwall and the formula to obtain the necessary irrigation time must therefore be adapted.

The surface influenced by rain is considered to be the top layer area ($A_rain$). However the surface undergoing losses by evapotranspiration is different. Indeed, the verticality of the surface of the greenwall must be taken into account. An approach consist in applying a weighting coefficient to $ET_c$. Coefficients of vertical wall varying from 0.76 to 1.46 were found [37]. However those are adapted to very specific situations. The approach used in this work was to consider that in addition to the top surface of the wall, the surface of all the metal cells containing the plants on the vertical wall were involved in evapotranspiration. The vertical evapotranspiration area for each layer $A_{ET,v(i)}$ is given by Equation 28.

$$A_{ET,v(i)} = A_{ET,tot} \cdot \frac{H_{(i)}}{H_{tot}} \qquad [cm^2] \tag{28}$$

With:

- $A_{ET,tot}$ is the total evapotranspiration area of the vertical surface of the wall $[cm^2]$. It is the sum of the surfaces of all the cells in the metal sheet.
- $H_{(i)}$ is the height of layer (i) $[cm]$.
- $H_{tot}$ is the total height of the wall $[cm]$.

It must be noticed that the evapotranspiration area of the top layer is the addition of the top surface of the wall and the vertical evapotranspiration surface. The irrigation volume required in each layer $V_i$ can then be obtained with Equation 32.

$$V_i = (\frac{ET_c}{10} \times A_{ET,v(i)}) - (\frac{R}{10} \times A_{rain}) \qquad [cm^3] \tag{29}$$

With:

- $ET_c$ is the daily crop evapotranspiration $[mm/day]$.
- $A_{ET(i)}$ is the evapotranspiration area of layer (i) $[cm^2]$.
- $R$ is the daily rainfall $[mm/day]$.
- $A_{rain}$ is the area influenced by rain $[cm^2]$. It is different from 0 $cm^2$ only for the top layer.

Finally the time $t_i$ for which the valve must remain open for each layer is given by Equation 30. This time also depends on the discharge, which will be different for the three layer heights considered (see section 2.4 Hydraulic design). The irrigation time selected will be the longest time to ensure that the plants are not under water stress in any of the layers.

$$t_i = \frac{V_i}{Q_i} \times 3600 \qquad [sec] \tag{30}$$

With:

- $V_i$ is the irrigation volume required in layer (i) $[cm^3]$
- $Q_i$ is the discharge in layer (i) $[cm^3/hr]$

#### 4.1.4 Description of plan C: Irrigation based on previous years' weather data.

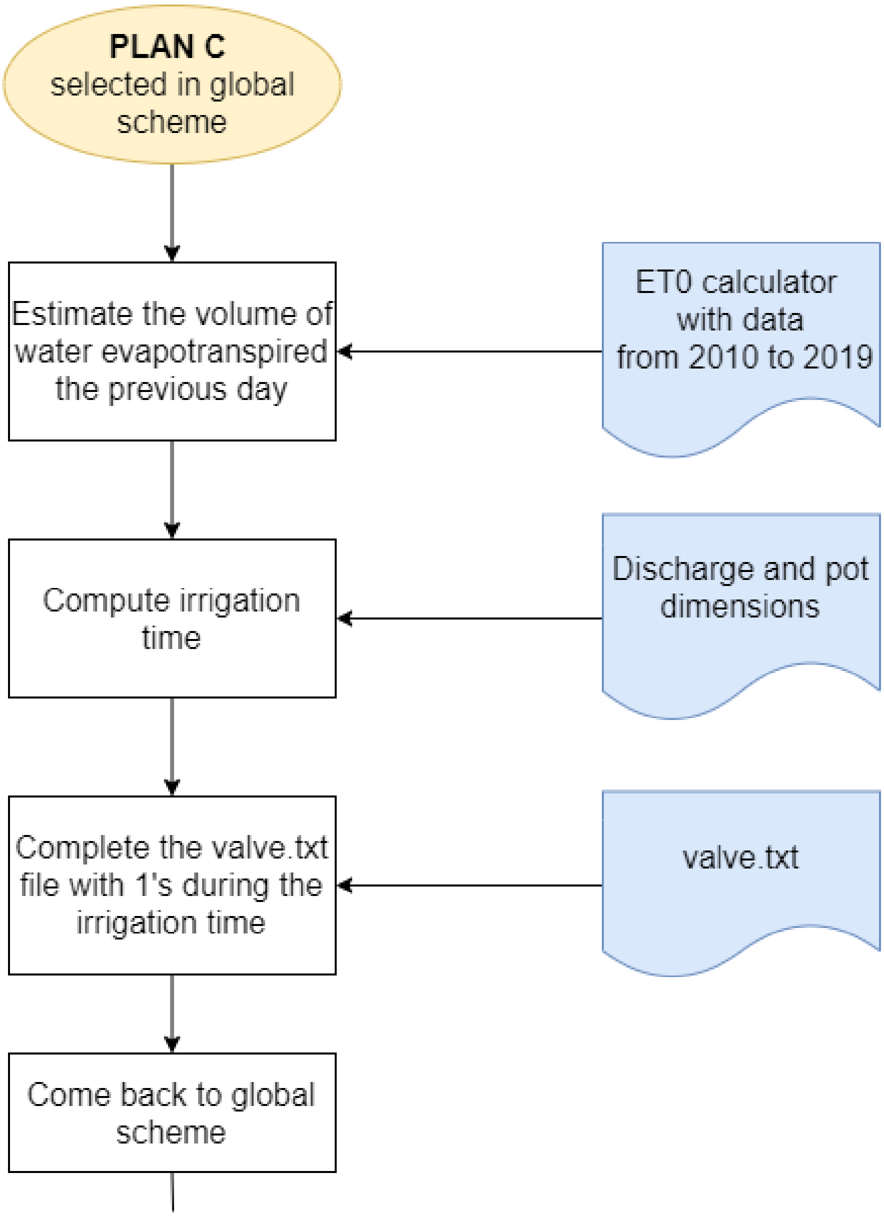



Figure 22: **Plan C: Irrigation based on previous years' weather data.**

Plan C was designed to provide irrigation when the first two strategies are subject to malfunction. It is coded between lines 884 and 996 of the irrigation program (Appendix B). It starts with the reading of the file `ET0_2010_2019.csv` that contains the 10 year average daily $ET_0$ in mm, calculated using meteorological data recorded at the weather station of Ernage (50.58°N, 4.69°E), near Gembloux. For each year between 2010 and 2019, the $ET_0$ was estimated using the $ET_0$ calculator software of the FAO. Then the daily average $ET_0$ was calculated and recorded in the `ET0_2010_2019.csv` file. Figure 23 shows the evolution of the average $ET_0$ between March 1 and May 31.

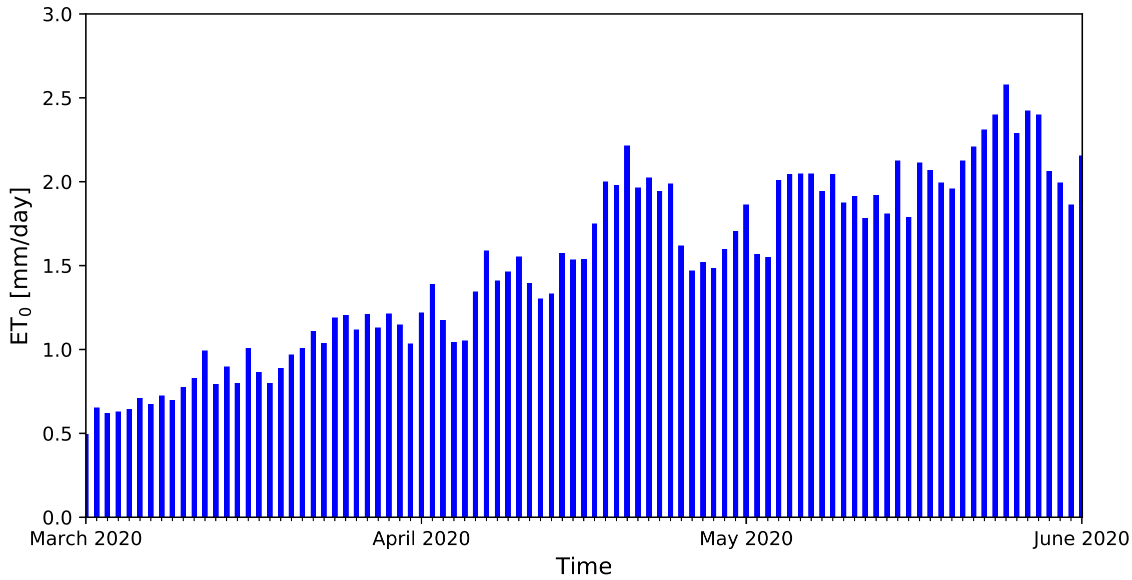



Figure 23: **10-year average daily ET$_0$, between March 1 and May 31.**

For Plan C, the crop evapotranspiration ($ET_c$) is calculated in the same way as for Plan B using Equation 26. Again the crop coefficient (K$_c$) was set to 0.5.

Given the highly intermittent nature of rainfall in Ernage, this variable was not averaged over 10 years. As a consequence, the equation used to calculate of the irrigation time $t$ (Equation 31) is a simplified version of the formula used in Plan B (Equation 27), as it does not take into account water inputs due to rainfall.

$$t = \frac{ET_c \cdot A}{Q} \cdot 3600 \qquad [sec] \tag{31}$$

With:

- $t$ is the irrigation time $[sec]$.
- $ET_c$ is the daily crop evapotranspiration $[mm/day]$.
- $A$ is the pot area $[mm^2]$.
- $Q$ is the discharge in the irrigation pipe $[mm^3/hr]$.

In the same way as in Plan A and Plan B, the instructions are transmitted to the valve by means of the file `valve.txt`. However, this time the instructions are written for the entire year. To keep track of the fact that irrigation was provided by Plan C, the file `history.txt` is filled with the letter "C".

**GREEN WALL ADAPTATION**

In Plan C, the code adaptations to greenwall are similar to Plan B except that no rainfall is taken into account. This simplifies Equation 32 as follows.

$$V_i = \frac{ET_c}{10} \times A_{ET,v(i)} \qquad [cm^3] \tag{32}$$

# 5 Results

## 5.1 Low variations of the volumetric water content over 5 minutes

This section presents the results of the estimation of the variation in VWC for a period of 5 minutes. As a reminder, in Plan A it was assumed that this variation was lower than the accuracy of the humidity sensors (0.03 $cm^3/cm^3$). VWC variations are estimated by calculating the difference between water inputs (irrigation and rainfall) and $ET_0$. Figure 24 shows the parallel evolution of water inputs, $ET_0$ and VWC variations over 5 minutes between April 23 and April 30, when Plan B was running. Water was brought in the pot by daily irrigation, but almost no rainfall occurred during this period. Figure 25 shows the parallel evolution of water inputs, $ET_0$ and VWC variations over 5 minutes between May 5 and May 9, when Plan A was running. Again water was only brought in the pot by daily irrigation, as no rainfall occurred. In both Figures 24 and 25 it clearly appears that the VWC variations are always lower than the EC5 uncertainty. Therefore the hypothesis seems valid. Moreover, the variations are even less than 0.02 $cm^3/cm^3$. In the data sheets of the humidity sensors supplied by the manufacturer, it is stated that in case of specific calibration of the sensors, the uncertainty equals 0.02 $cm^3/cm^3$. Therefore, this assumption remains valid even when the sensors are calibrated precisely.

However, it is important to point out the limitations of this test. Indeed, rainfall was very low throughout the duration of the pot irrigation experiments. The main water input is therefore irrigation. The intensity of irrigation is limited by the flow rate in the main irrigation pipe (1 l/hr). Given the surface area of the studied pot (1920 $cm^2$), and assuming a perfect distribution uniformity, the maximum irrigation intensity is 0.43 mm/5 min. If a rainfall of higher intensity occurs suddenly, it is not certain that the assumption is still valid. However, two elements tend to reinforce this hypothesis. Firstly, even in the case of a heavy rainfall, it usually comes gradually. Since it is the variation in VWC over a short period of time that is studied here, the impact should not be much greater than the abrupt opening of the valve. Second, it appears from Figures 24 and 25 that the 5 minutes VWC variations are mainly impacted by the $ET_0$, irrigation seems to cause very little variation. This would probably also be the case with a rainfall.

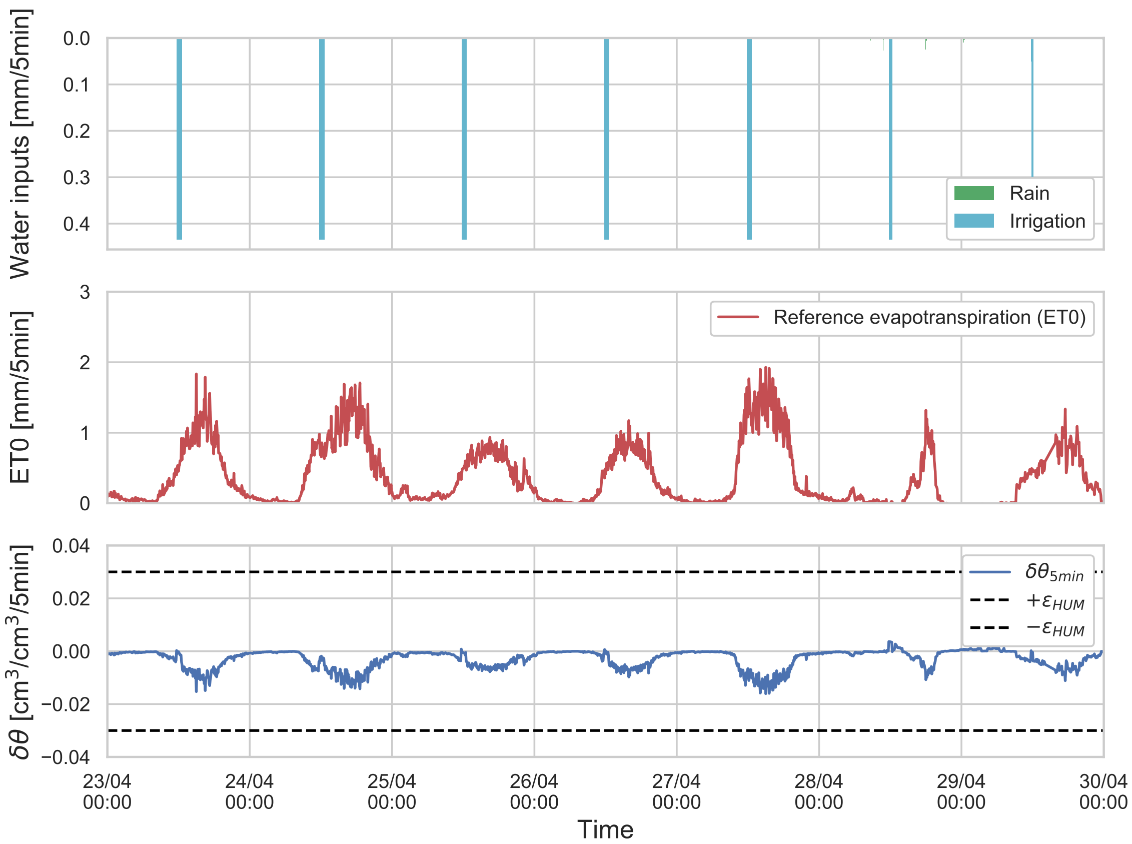



Figure 24: **Water inputs, ET$_0$ and VWC variations evolutions over 5 minutes between April 23 and April 30.** Irrigation is provided by plan B. $+\epsilon_{HUM}$ and $-\epsilon_{HUM}$ designate the upper and lower boundary of the uncertainty interval of the EC5 sensors.

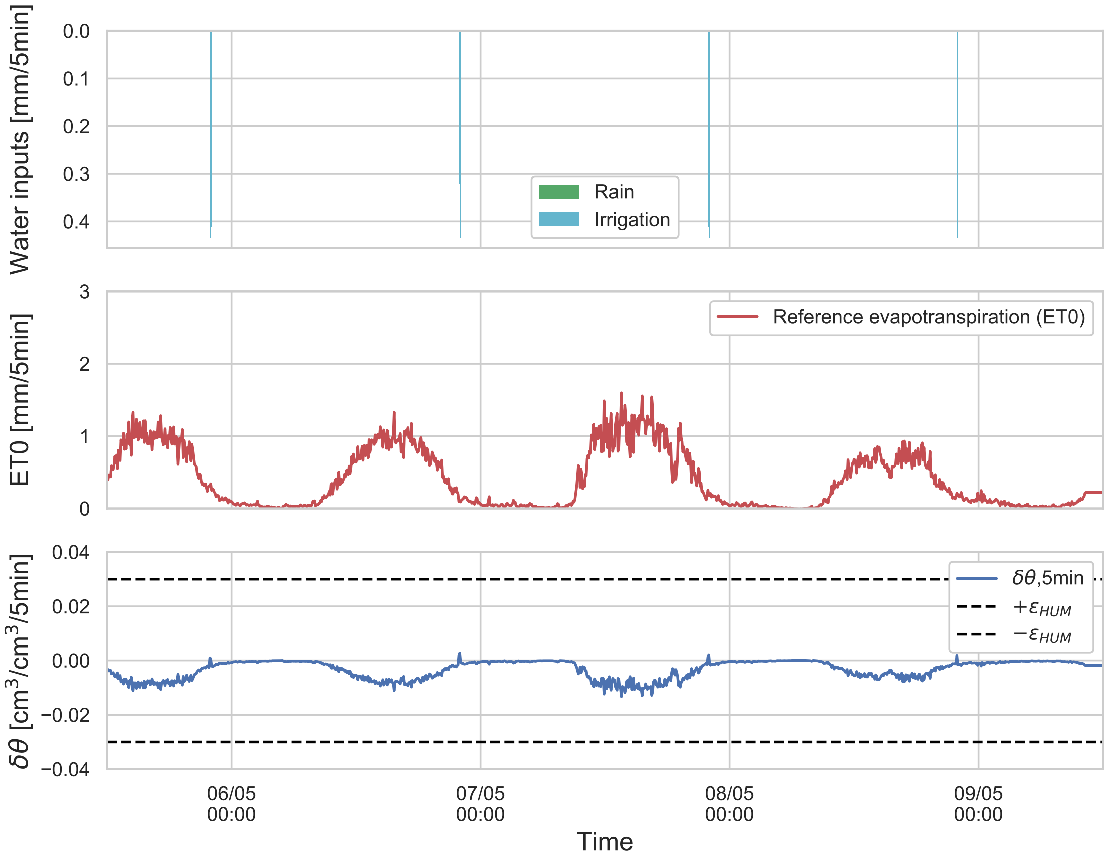



Figure 25: **Water inputs, $ET_0$ and VWC variations evolutions over 5 minutes between May 5 and May 9.** Irrigation is provided by plan A. $+\epsilon_{HUM}$ and $-\epsilon_{HUM}$ designate the upper and lower boundary of the uncertainty interval of the EC5 sensors.

## 5.2 Temperature correction

Figure 26 compares daily variations of air temperature and SWC measured by the three humidity probes (HUM7, HUM8 and HUM9). Daily cycles can be clearly identified for both variables. Moreover, it seems like SWC and air temperature evolve in opposite direction. Indeed when air temperature reaches a peak, SWC is at its lowest level. Inversely, when air temperature is at its lowest point, SWC reaches a local maximum at almost the same time.

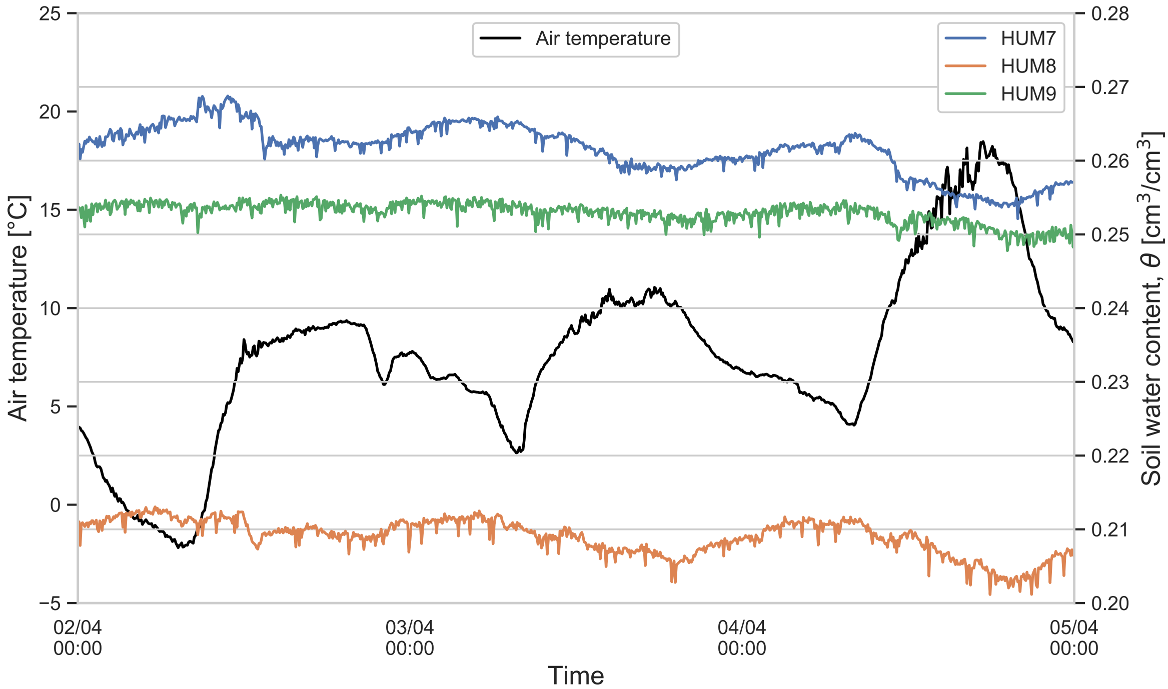



Figure 26: **Evolution of VWC and air temperature between April 2 and April 5.** Neither precipitation nor irrigation occured during this period.

Figure 27 shows the results of the correlation analysis between SWC measured by the EC5 probes (converted in $cm^3/cm^3$ with Cédric Bernad's calibration equation) and air temperature. The negative values of Pearson's coefficient of correlation ($r^2$) obtained for the three probes show a strong negative correlation. This suggests that the real component of dielectric permittivity (permittivity of soil constituents) dominates in this soil. Corresponding p-values are lower than 0.001. It suggests that the correlations are very highly significant. However care should be taken to interpet the p-values obtained. Indeed the calculation relies on the assumption that each dataset is normally distributed [17]. Given the fact that SWC is not stationary, the hypothesis of a normal distribution of the SWC is incorrect.

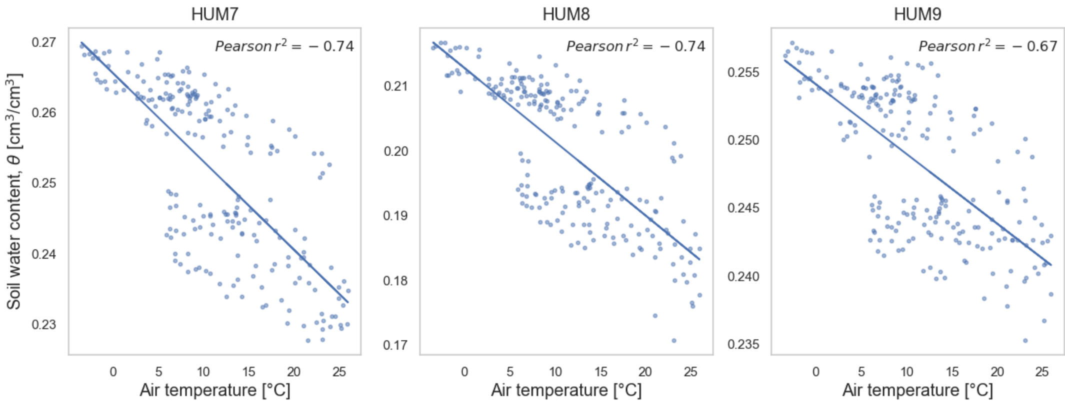



Figure 27: **Correlation between SWC and air temperature, for the three EC5 probes.**

Assuming the relationship between SWC and air temperature, a multiple regression was performed for each sensor. Resulting coefficient of Equation 16 are shown in Table 5.

Table 5: **Temperature correction: multiple regression coefficients for each humidity probe.**

| Sensor | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|
| HUM7 | 0.673 | 0.0002 | 0.084 |
| HUM8 | 0.033 | -0.0001 | 0.202 |
| HUM9 | 0.491 | -0.0001 | 0.129 |

Figure 28 shows the effect of temperature correction on the output of each EC5 sensor. It appears that the multilinear regression approach succeeds in decreasing the daily VWC fluctuations. Cyclical variations still persist. The reason for this is most likely the use of air temperature rather than soil temperature to correct VWC, as the matter is not available. Differences between air and ground temperatures are likely to exist. It becomes critical when water is brought to the surface (rainfall, irrigation). If the temperature of water is significantly different from the air temperature, then the soil temperature will be strongly affected. At that point it will probably depend much less on the air temperature. In this case the equations found are probably no longer valid. If future improvements are considered, the authors recommend :

1. The use of three sensors, such as thermocouples, to measure soil temperature close to the humidity probes.

2. To perform multilinear regression on a group of 24 hours periods that spans the range of expected VWC.

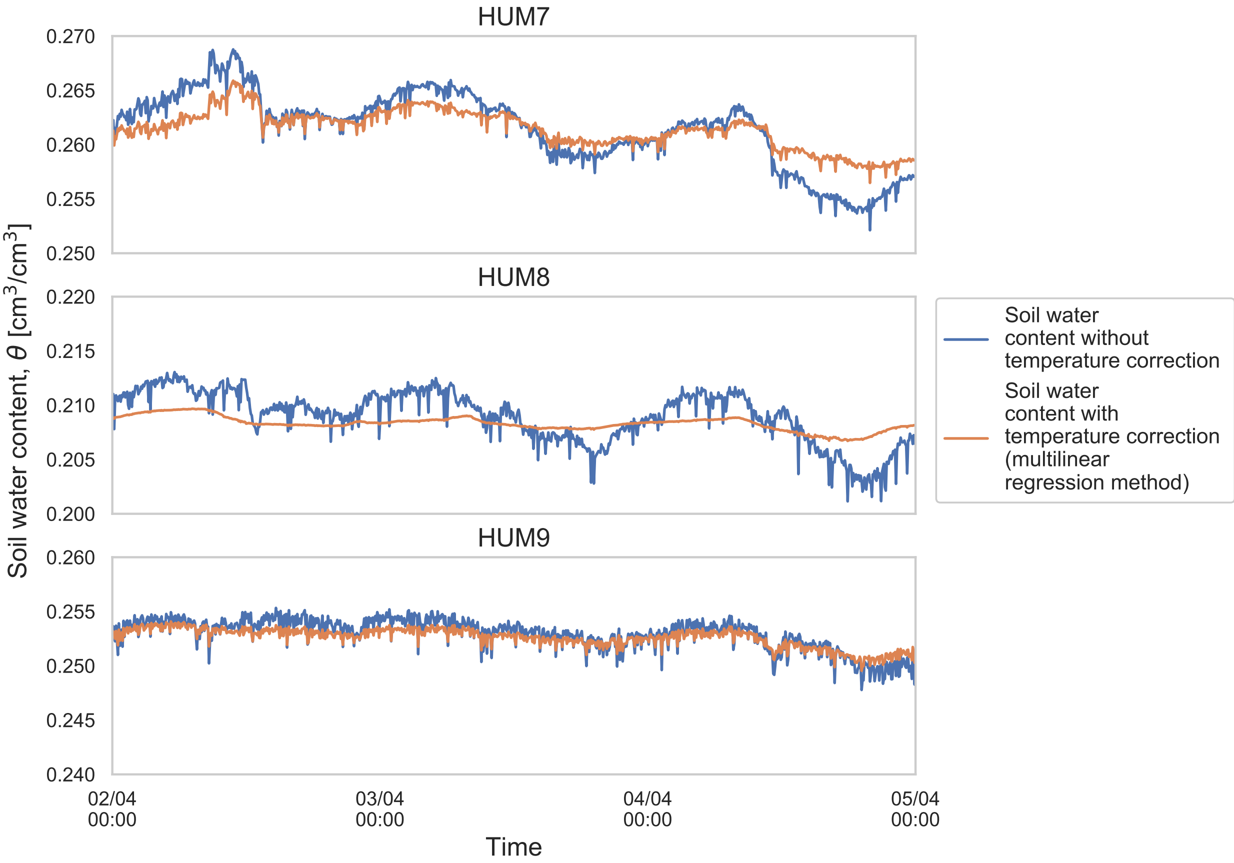



Figure 28: **Results of the temperature correction of VWC measured by the EC5 probes, using air temperature.**

## 5.3 Meteorological data

The meteorological variables were measured continuously throughout the irrigation experiment by the weather station. Air temperature, wind speed, rain, vapor pressure and solar radiation are used to calculate $ET_o$ (Equation 21). Figure 29 presents the evolution of those variables, as well as relative humidity, from 23 April to 7 May (period when most experiment occured). The daily cycle of solar radiation, air temperature and relative humidity is noticeable. Solar radiation is equal to zero during the night and increases during the day to reach 800 W/m$^2$. Air temperature and relative humidity have opposite variation. Regarding to the values of the data, wind speed taks values between 0 m/s and 4 m/s. The two periods are also characterized by small rain events comprised between 0 mm/h and 0.4 mm/h.

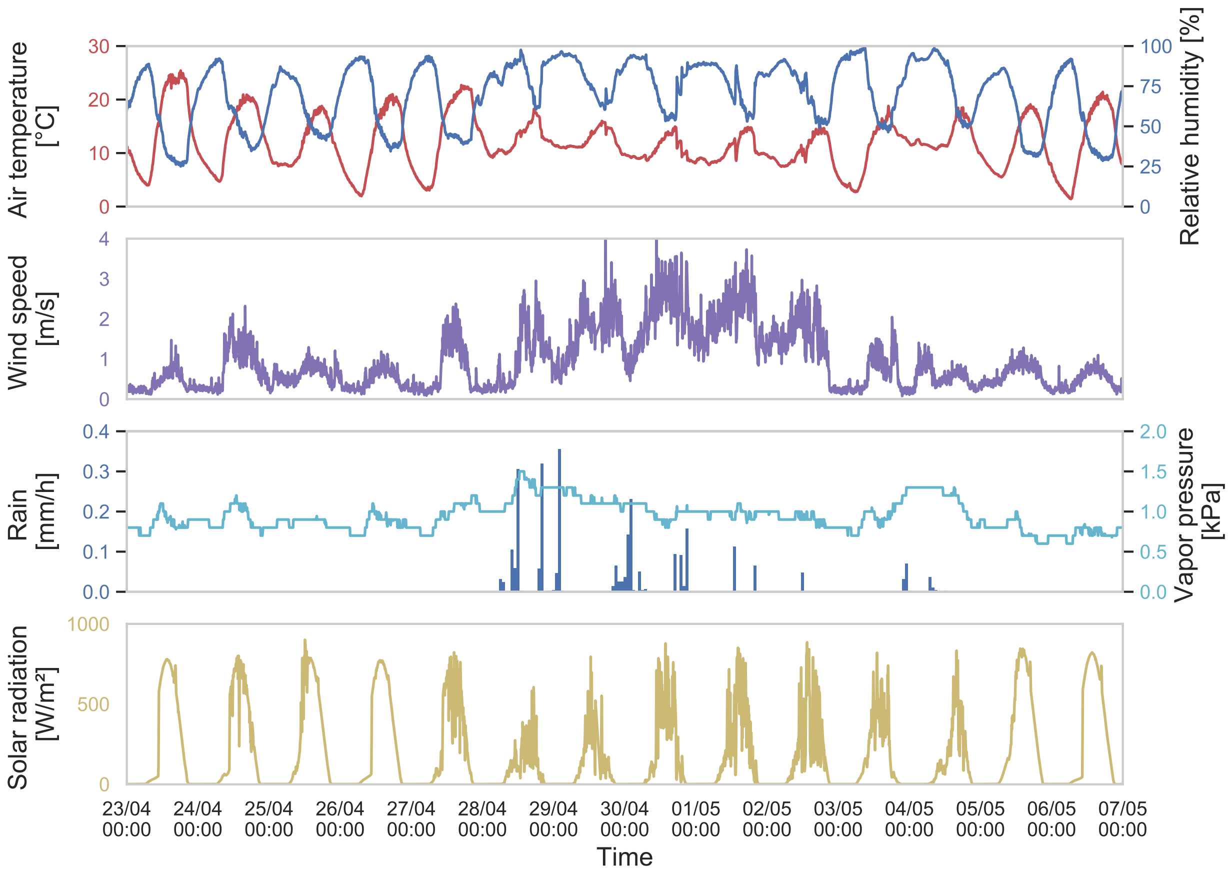



Figure 29: **Evolution of meteorological data, between April 23 and May 7, with an acquisition time step of one minute.**

## 5.4 Beta runs

The irrigation program was first implemented in parts. Each of the Plans (A, B and C) was implemented independently on the server, in order to ensure its proper functioning on the one hand, and to make modifications if necessary, based on the results, on the other hand.

### 5.4.1 Beta run of plan A

Figure 30 shows the evolution of VWC when Plan A was run for the first time, between May 6 and May 9. Irrigation took place every day at 11 pm. The addition of water generates little variation in the SWC on an hourly scale. Daily fluctuations due to temperature variations appear to be greater. This is probably due to low irrigation rates. Indeed, it can be seen on Figure 30 that irrigation doses never exceed 0.6 mm per 24 hours. It appears that soil humidity decreases steadily during between May 5 and May 9. The amount of water brought to the pot was therefore insufficient. The main reason of this is most likely the fact that the value of flow rate in the main irrigation pipe was set to 1.5 l/h. It was later realized that the flow rate was lower. It is actually around 1 l/h. Overestimating the flow rate leads to an underestimation of the time during which the valve must be kept open, and therefore to an irrigation dose that is too low. Secondly, this can also be due to the calculation of the volume of water necessary to reach field capacity, and thus to the estimation of VWC at field capacity in each zone. Those values have therefore been increased by one hundredth in order to correct the volume of water calculated (Table 6).

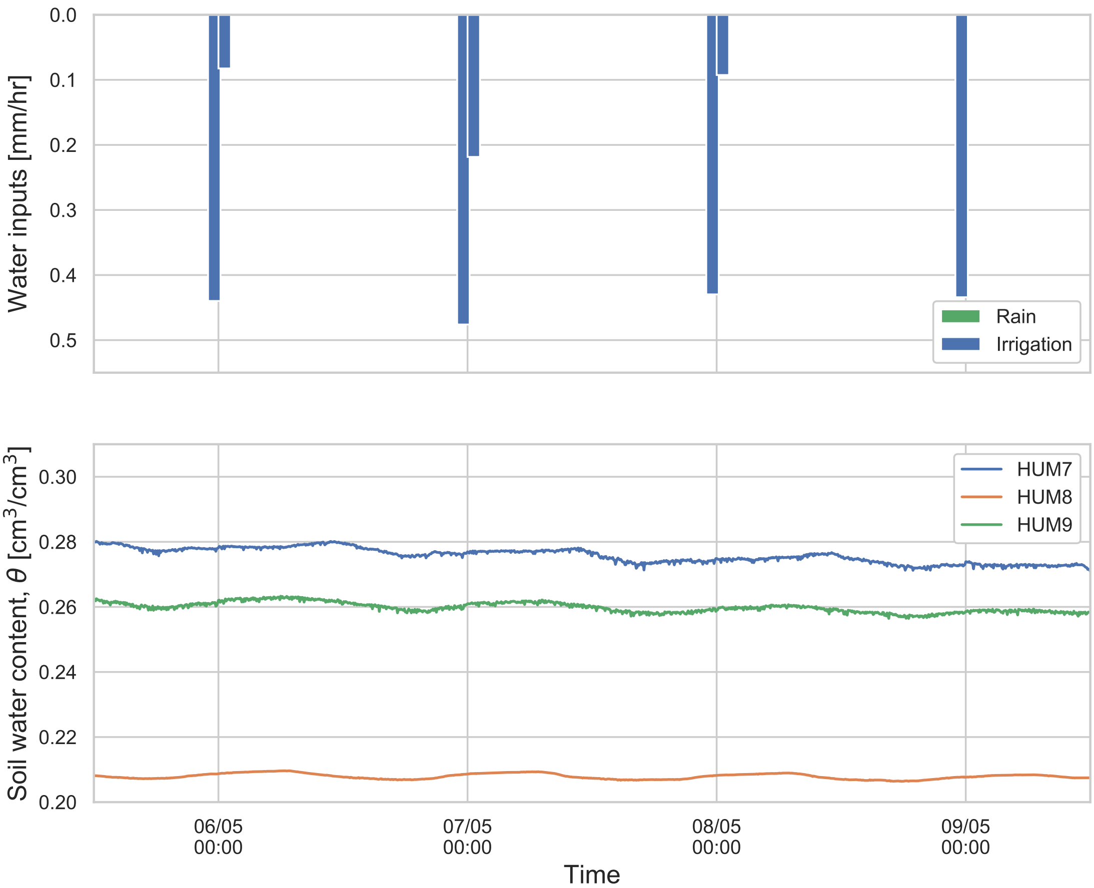



Figure 30: **Evolution of soil moisture in the three zones of the pot, during the beta run of Plan A.**

Table 6: **Re-estimation of volumetric water content at field capacity in the three zones of the pot, after running Plan A.**

| Sensor | HUM7 | HUM8 | HUM9 |
|---|---|---|---|
| $\theta_{FC}[cm^3/cm^3]$ before running Plan A | 0.275 | 0.215 | 0.26 |
| $\theta_{FC}[cm^3/cm^3]$ after running Plan A | 0.285 | 0.225 | 0.27 |

### 5.4.2 Beta run of plan B

The plan B was run independently for one week, from April 23 to April 30. Figure 31 shows the evolution of VWC, parallel to water supplies. Irrigation took place every day at 1 pm. First, the effect of the addition of water to the pot is noticeable in the first (HUM7) and the third (HUM8) zones. VWC starts to decrease slightly, then rise again to reach a peak. Those variations are probably due to the temperature of water. As it is likely to differ from the temperature of the air, the method used to correct temperature in this study is unable to correct the effect of water temperature in those zones. However, these variations are not observable in the second zone (HUM8). This difference remains unexplained. On a broader scale, an increase of soil water content through time clearly appears between April 23 and April 30, which may indicate that the irrigated volume is probably above the actual water needs of the plants. Irrigation doses reach 5 mm/day at the beginning of the period studied (between April 24 and April 29). The problem seems to come from the real evapotranspiration calculation. Indeed, the first value of Kc considered was 1, while the rocket had just been sown. A lower value of the crop coefficient should have been set, as a Kc of 1 is only appropriate for plants replanted at an advanced stage of development. As a result, it overestimated the amount of water evapotranspired and so the amount of water that should be put back into the system. The solution to correct the irrigated volume was to reduce the crop coefficient by half (0.5 instead of 1). The overestimating of the evapotranspiration can also be caused by the parameters used in the Penman-Monteith equation. For example, it is possible that the wind measured by the weather station is higher than the one actually present near the pots. Indeed, the balcony can reduce this win. It is interesting to note that in the last two days, the irrigation doses have been reduced. Indeed, Plan B takes into account the rain that fell the last 24 hours. It shows the correct functioning of the system and the sustainability of it (using every resource available).

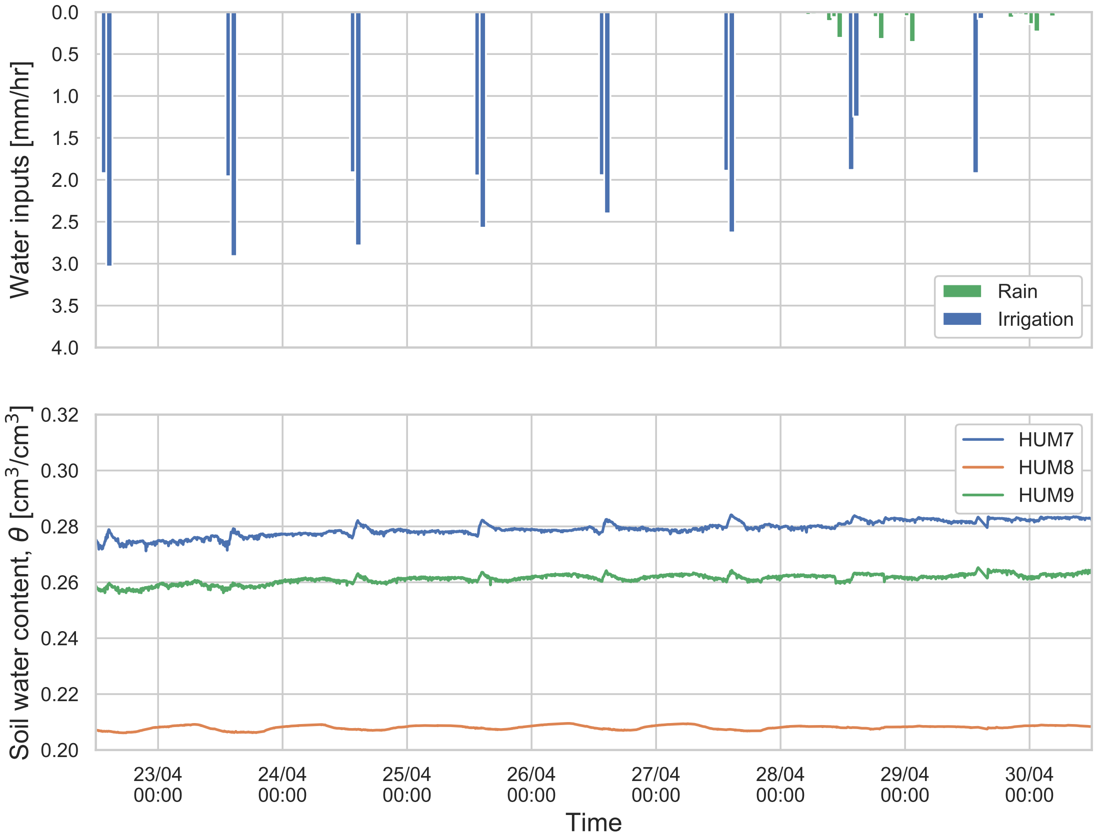



Figure 31: **Evolution of soil moisture in the three zones of the pot, during the beta run of Plan B.**

### 5.4.3 Beta run plan C

The plan C was just tested over one day, mainly because of a lack of time. Irrigation started a little before midnight of April 22. The beta run only served to ensure that the program was able to perform irrigation. This being said, its performances are expected to be a little worse than Plan B since it does not take rainfall into account and because the $ET_0$ is an approximation based on previous years. In Plan C too, the $K_c$ value was initially set to 1. Given the results of Plan B (see previous section), this value was reduced to 0.5. It could be interesting to test that plan on a longer period of time, to make sure that it does give reasonably good results.

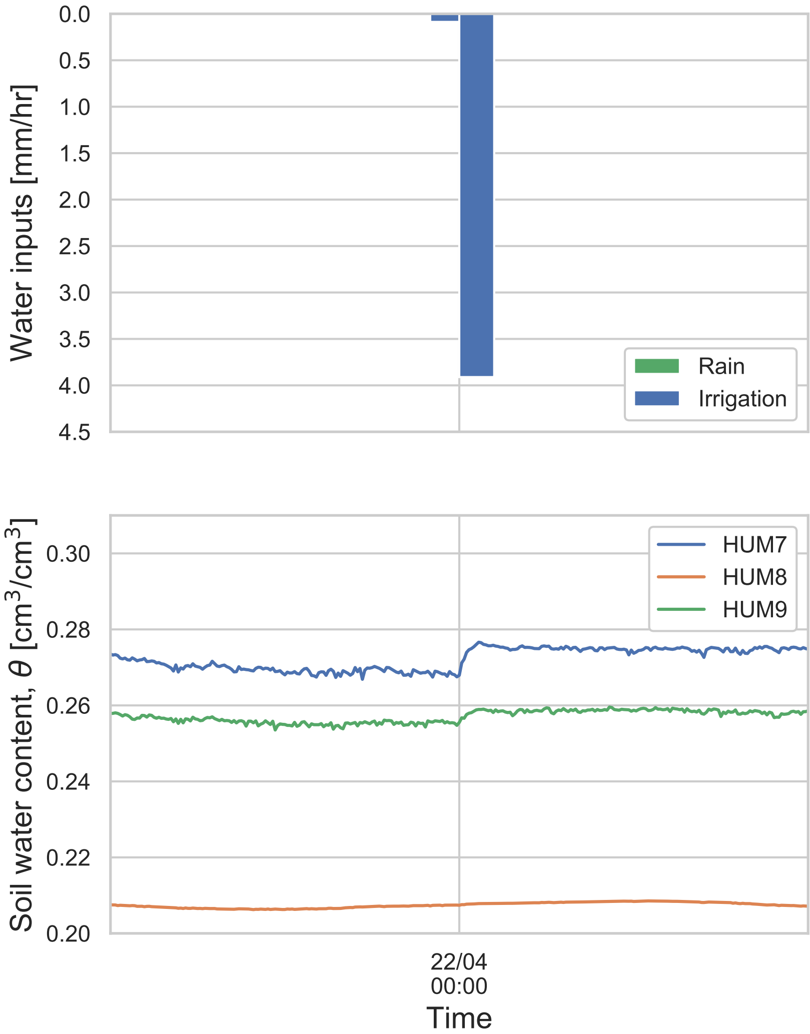



Figure 32: **Evolution of soil moisture in the three zones of the pot, during the beta run of Plan C.**

### 5.5 Global Scheme

The global scheme was tested between May 10 and May 14, taking into account the observations relating to each of the plans tested separately (see previous sections). Since the humidity probes have always met the conditions for the application of plan A, irrigation based on the water content measurements was carried out. Neither plan B nor plan C had to be used. Figure 33 shows the evolution of VWC, parallel to water supplies (expressed in mm per day). Dotted lines were added to the graph to show the value of VWC at field capacity considered in each zone. Between May 10 and May 14 the moisture content seems to increase slightly. At the beginning of this period it is slightly lower than the field capacity values in the first (HUM7) and third (HUM9) zones. In the second zone (HUM8), the difference is greater. At the end of the five days, the target field capacity values are slightly exceeded in the first and third zones, while it is not reached in the second zone. However, the common trend in all three areas is the increase in SWC. This study period is very short in order to be able to draw robust conclusions. However, the many uncertainties about the VWCs at field capacity in the different areas, but also about the actual flow through the gate, make it likely that irrigation rates still differ from the optimal rates.

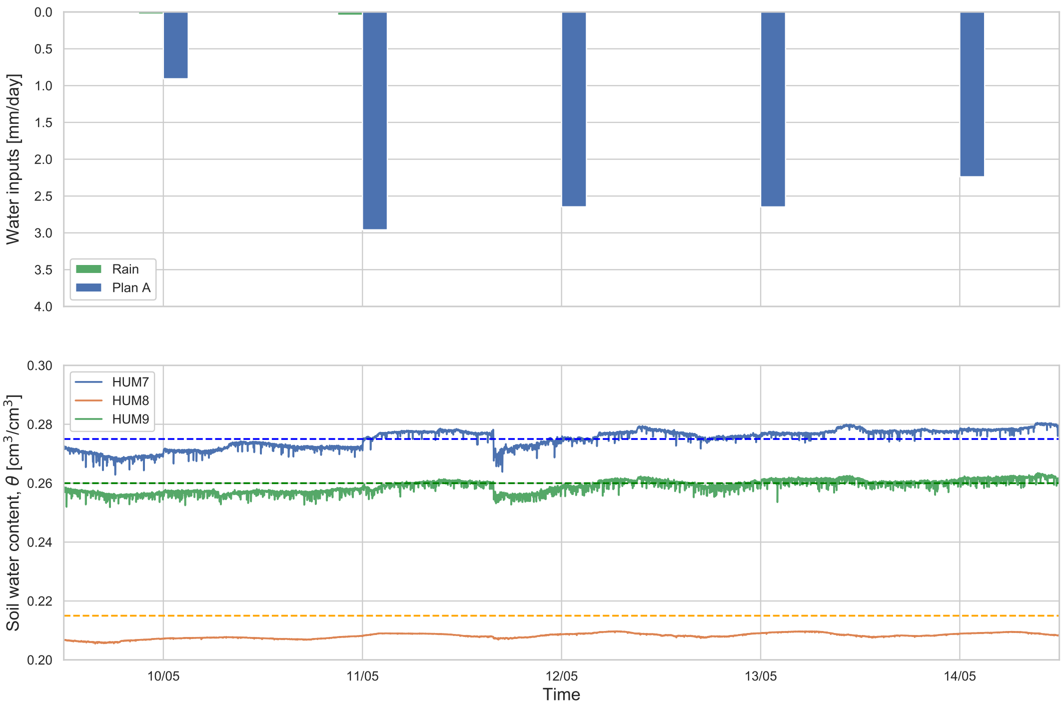



Figure 33: **Evolution of soil moisture in the three zones of the pot, during the test of the global scheme.** Dotted lines represent estimated VWC at field capacity in the three zones.

Since Plans B and C only depend on variables external to the system to be irrigated, the irrigation rates that would have been applied if these strategies had been used can be calculated. Figure 34 shows irrigation doses calculated by each Plan, between May 10 and May 14 (while the Global Scheme, thus Plan A, wan running, as discussed previously). It allows to compare the different plans. However, this does not determine which of the three plans provides the optimal water dose. It appears that the plan C is the one that provides the most constant irrigation doses. It also appears that plan B is nearer to plan A than plan C, as expected. Between May 11 and May 13, Plan A calculated the highest irrigation doses.

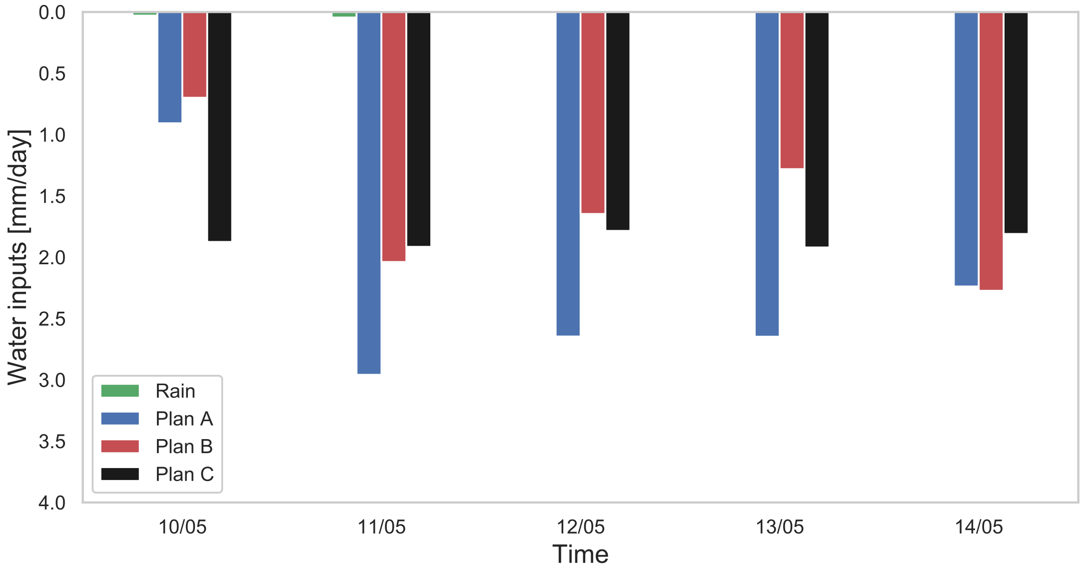



Figure 34: **Comparison of the three irrigation strategies (Plan A, Plan B, Plan C).**

# 6 Critical analysis

## 6.1 Global system

In this section some limitations of the method used are presented. If the project has to be continued we advice to take theses into account.

### 6.1.1 Construction

We wanted to put the sensors threads and the local server outside, near the reservoir and it could have been a problem in case of heavy rain.

We tried to keep a constant density in the wall while we constructed. And we based our probes calibrations, and measurements field capacity measurements on it. But we can not guarantee that the density was exactly the same.

No geotextile was put in the wall, we should therefore be careful to check if the competition between the plants is not too high. We want indeed to keep a neet equilibrium between the different species, to keep the wall a nice thing to look at.

### 6.1.2 Wall characterisation

We were not able to compute the water distribution in the wall with the help of *Hydrus* and had to made an approximation. We based our plants location on it so if our approximation was false it could have resulted in a higher mortality for the plants. Furthermore, we considered that the p factor was constant and we approximated his value as we did for the $K_c$. Some false values would have a quite negative effect in the computation of the plants water needs.

For the hydraulic calculations we made a lot of hypothesis when characterizing the pipes, the wall and the reservoir. Also, the friction forces due to the singularities where not found in the literature and where really difficult to estimate. A better technique would have been to do some tests of the water flows in the real conditions.

### 6.1.3 Sensors

When computing the water needs attention should be made to the fact that the wall is vertical. The insulation will therefore maybe be higher as well as the wind. Maybe the Penman-Monteith equation should be adapted.

The correction of the humidity data with the temperature should be evaluated on a long term period to determine if the method used seems correct. Furthermore, it is known that the air temperature differs from the wall temperature, it is thus advised to place some temperature sensors in the wall.

### 6.1.4 Code

The global code was run during several days, and the amount of water that would have been given by each plan was calculated. It appears that the plan C is the one with the slightest variations between days. It also appears that plan B is nearer to plan A than plan C, as expected. The fact that the VWC slightly increase during the Global Scheme running should be investigated.

For the Beta run of the plan B the irrigation happened every day at 1 pm. It seems not to be the best time to irrigate since the insolation is high at this hour and water from the pots will evaporate quicker.

## 6.2 Team organisation

In the beginning of the semester we saw each other during the course each Monday. And we did a to-do list for the week with tasks for everybody. It helped us to structure the work and to split in between Louvain-La-Neuve and Gembloux. We also did some practical tasks during the other days of the week (construction, calibration, tests for example). The presentations we had to do for the course also helped us to keep moving.

We distributed the tasks quite well within the group. In the beginning Maud was in charge of the plants, Matthieu tried to understand the code, Jonathan made some researches about ET0, and Adrien and Eloy had to investigate on and calculate the discharge in the pipe. As of Easter the project began to ask more time and we did a new task distribution. Mathieu and Adrien were in charge of the code while Jonathan, Eloy and Maud wrote the report.

In the beginning of the project it was quite difficult to understand all the different tasks we had. Furthermore everything was moving really slowly : the reservoir was not placed , neither was the hydraulic system. The understanding of the code and the beginning of adapting it took a while too.

## 6.3 Coronavirus modification

We established a weekly reunion every Wednesday to keep everybody informed of the latest improvements and to distribute some tasks. We also did some reunions to prepare the little presentations. Finally we had some meeting with Christophe and S. Garré to ask some questions and help us with the code. We manage to keep working at a good pace and without letting anyone behind.

Unfortunately the plants and the hydraulic system will never have been put in place, even though that would have been interesting. Doing the project in a more theoretical way was still interesting but not as motivating as before...

# 7 Conclusion

Currently, agriculture and irrigation have a significant impact on water resources and are key player in the management of the blue gold. The initial objective of the project was to design and implement an automated irrigation system for a living wall module of 1 m x 2.20 m in Gembloux. Green walls are aesthetic elements with multiple assets : contributing to air quality, maintaining biodiversity, temperature reduction, *etc.* This project and report gives enough information on the way to manage a green wall and is a first step in the spreading of this new technique.

The system used is composed of a green wall module filled with substrate and native plants (*Thymus p.*, *Origanum v.*, *Dianthus c.*, *Asplenium t.*, *Geranium s.* and *Melica c.*). An hydraulic system is coupled with the module and consist of a water tank located at three meters heights, some pipes and a dripping irrigation system. Some sensors where also used in order to characterise the system : a weather station above the module and three humidity sensors located at different heights in the wall. The humidity sensors measure the soil volumetric water content. These sensors are sensitive to temperature and a multilinear regression was made to take into account this parameter. The weather station measures and records different parameters such as wind speed, air temperature or solar radiation. The data recorded by the sensors are then used to irrigate. Concretely, the data generated by these sensors is send to a servor and a code passing through the same servor allows us to controle the opening and closing of a valve to start and stop the irrigation.

Due to the coronavirus crisis we were not able anymore to work on the green wall system. Some little pots were placed on a balcony in Mont-Saint-Guibert to allow us to run the irrigation code anyways.

Overall, three possibilities were found to irrigate, in the form of the three different plans. The first one, plan A, based on the sensors result, seems to be the most effective one, even though it might be interesting to try and quantify exactly how good this plan is compared to the other ones. The two other plans, based on ETP calculation with respectively direct meteorological data and past years meteorological data, revealed to be applicable too, as shown with their beta runs. This being said, it is obvious that the equations could and should be adapted when used for a real green wall and not for plants in a pot.

This project was a great way to mix up a lot of different knowledge we acquired during our first year of studies. It also allowed us to conduct the management of a whole and complex system. Furthermore, the coding part was a challenging one. We found on the whole the project really interesting even if we would have liked to see the real output of it and to conduct it until the end.

Some improvements can be made. Obviously, all the code was never applied to a real green wall, so it would be very interesting to compare the results when it comes to a green wall. Modifying the equations would most likely be needed. Besides, testing all three plans separately for a longer period of time could also help knowing which one should actually be used in a real situation. Concerning the wall design it would be interesting to compare it with the ones from the other groups and to come to find a way to optimize this design.

# 8 Bibliography

[1] Eleftheria Alexandri and Phil Jones. Temperature decreases in an urban canyon due to green walls and green roofs in diverse climates. 43:480–493, 2008.

[2] R Allen, Pereira L, D Raes, and Smith M. *Crop evapotranspiration - Guidelines for computing for crop water requirements - FAO irrigation and draingae papers*, chapter 8 : $ET_c$ under soil water stress conditions. FAO, 1998.

[3] R Allen, Pereira L, D Raes, and Smith M. *Crop evapotranspiration - Guidelines for computing for crop water requirements - FAO irrigation and draingae papers*, chapter 6 : $ET_c$ - Single crop coefficient (Kc)$.FAO,$1998.

[4] Francisca Amorim and Paulo Mendonça. Advantages and Constraints of Living Green Façade Systems. *International Journal of Environmental Science and Development*, 8(2):124–129, 2017.

[5] Eban Z. Bean, Ray G. Huffaker, and Kati W. Migliaccio. Estimating Field Capacity from Volumetric Soil Water Content Time Series Using Automated Processing Algorithms. *Vadose Zone Journal*, 17(1):180073, 2018.

[6] H. R. Bogena, J. A. Huisman, C. Oberdörster, and H. Vereecken. Evaluation of a low-cost soil water content sensor for wireless network applications. *Journal of Hydrology*, 344(1-2):32–42, 2007.

[7] Heye Reemt Bogena, Johan Alexander Huisman, Bernd Schilling, Ansgar Weuthen, and Harry Vereecken. Effective calibration of low-cost soil water content sensors. *Sensors (Switzerland)*, 17(1), 2017.

[8] BRL exploitation. Memento technique, irrigation des espaces verts. pages 27–29, 2010.

[9] Doug Cobos, Colin Campbell, and Decagon Devices. Correcting temperature sensitivity of ECH 2 O soil moisture sensors Strategy 1 : Multiple regression analysis. (1).

[10] Commission Européenne. L'infrastructure verte. pages 1–4, 2010.

[11] Decagon Devices. EC-5 Manual. pages 1–15, 2019.

[12] R. Djedjig, E. Bozonnet, and R. Belarbi. Experimental study of the urban microclimate mitigation potential of green roofs and green walls in street canyons. *International Journal of Low-Carbon Technologies*, 10(1):34–44, March 2015.

[13] Rabah Djedjig. Experimental study of green walls impacts on buildings in summer and winter under an oceanic climate. *Energy and Buildings*, page 9, 2017.

[14] John Dover. *Green infrastructure: incorporating plants and enhancing biodiversity in buildings and urban environments.*, page 175. Earthscan, August 2016.

[15] Michael F Goodchild. *Recent Advances in Geographic Information Science.*

[16] M.B. Kirkham. Field Capacity, Wilting Point, Available Water, and the Non-Limiting Water Range. *Principles of Soil and Plant Water Relations*, pages 101–115, 2005.

[17] C. J. Kowalski. On the Effects of Non-Normality on the Distribution of the Sample Product-Moment Correlation Coefficient. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 21(1):1–12, 1972.

[18] Jeremy T Lundholm. Green roofs and facades: a habitat template approach. *Urban Habitats*, 4(1):87–101, 2006.

[19] Maria Manso and João Castro-Gomes. Green wall systems: A review of their characteristics. *Renewable and Sustainable Energy Reviews*, 41(January 2018):863–871, 2015.

[20] Ugo Mazzali, Fabio Peron, Piercarlo Romagnoni, Riccardo M. Pulselli, and Simone Bastianoni. Experimental investigation on the energy performance of Living Walls in a temperate climate. *Building and Environment*, 64:57–66, June 2013.

[21] Alexandra Medl, Rosemarie Stangl, Silvia B. Kikuta, and Florin Florineth. Vegetation establishment on 'Green Walls': Integrating shotcrete walls from road construction into the landscape. *Urban Forestry & Urban Greening*, 25:26–35, July 2017.

[22] Krishna S. Nemali, Francesco Montesano, Sue K. Dove, and Marc W. van Iersel. Calibration and performance of moisture sensors in soilless substrates: ECH2O and Theta probes. *Scientia Horticulturae*, 112(2):227–234, 2007.

[23] Krishna S. Nemali and Marc W. Van Iersel. Physiological responses to different substrate water contents: Screening for high water-use efficiency in bedding plants. *Journal of the American Society for Horticultural Science*, 133(3):333–340, 2008.

[24] Hamideh Nouri, Simon Beecham, Ali Morad Hassanli, and Fatemeh Kazemi. Water requirements of urban landscape plants: A comparison of three factor-based approaches. *Ecological Engineering*, 57:276–284, 2013.

[25] Erica Oberndorfer, Jeremy Lundholm, Brad Bass, Reid R. Coffman, Hitesh Doshi, Nigel Dunnett, Stuart Gaffin, Manfred Köhler, Karen K. Y. Liu, and Bradley Rowe. Green Roofs as Urban Ecosystems: Ecological Structures, Functions, and Services. *BioScience*, 57(10):823–833, 2007.

[26] S. L. Pimm, C. N. Jenkins, R. Abell, T. M. Brooks, J. L. Gittleman, L. N. Joppa, P. H. Raven, C. M. Roberts, and J. O. Sexton. The biodiversity of species and their rates of extinction, distribution, and protection. *Science*, 344(6187), 2014.

[27] Thomas A.M. Pugh, A. Robert MacKenzie, J. Duncan Whyatt, and C. Nicholas Hewitt. Effectiveness of green infrastructure for improvement of air quality in urban street canyons. *Environmental Science and Technology*, 46(14):7692–7699, 2012.

[28] Luis Pérez-Urrestarazu and Miguel Urrestarazu. Chapter 2.2 - vertical greening systems: Irrigation and maintenance. In Gabriel Pérez and Katia Perini, editors, *Nature Based Strategies for Urban and Building Sustainability*, pages 55 – 63. Butterworth-Heinemann, 2018.

[29] Abdul M.A. Rahman, Foong S. Yeok, and Atikah F. Amir. The building thermal performance and carbon sequestration evaluation for Psophocarpus tetrogonobulus on biofaçade wall in the tropical environment. *World Academy of Science, Engineering and Technology*, 76(4):86–94, 2011.

[30] J-C Rameau, D Mansion, and G Dumé. *Flore Forestière Française, guide écologique illustré*, volume 1 : Plaines et collines. Institut pour le développement forestier, 1989.

[31] Bronwyn Rayfield, Jérôme Dupras, Xavier Francoeur, Maria Dumitru, Danielle Dagenais, Jérémie Vachon, and Alain Paquette. Spatial alternatives for Green Infrastructure planning across the EU: An ecosystem service perspective. *Landscape and Urban Planning*, 138(1):132–143, 2015.

[32] Richard Raymond and Laurent Simon. Biodiversité: Les services écosystémiques et la nature en ville. *Revue Forestiere Francaise*, 64(3):339–350, 2012.

[33] Marco Schmidt. Energy Saving Strategies Through the Greening of Buildings. 2002(September 2002):1–5, 2003.

[34] Karen C. Seto, Burak Güneralp, and Lucy R. Hutyra. Global forecasts of urban expansion to 2030 and direct impacts on biodiversity and carbon pools. *Proceedings of the National Academy of Sciences of the United States of America*, 109(40):16083–16088, 2012.

[35] T. Swoczyna, J. Mojski, A.H. Baczewska-Dabrowska, H.M. Kalaji, and N.I. Elsheery. Can we predict winter survival in plants using chlorophyll a fluorescence? *Photosynthetica*, 58:433–442, April 2020.

[36] University of California Cooperative Extension California Department of Water Resources. A Guide to Estimating Water Needs Of Landscape Plantings in California. 2000.

[37] P. M.F. van de Wouw, E. J.M. Ros, and H. J.H. Brouwers. Precipitation collection and evapo(transpi)ration of living wall systems: A comparative study between a panel system and a planter box system. *Building and Environment*, 126(September):221–237, 2017.

[38] M Vanclooster, M Javaux, and Y Bartosiewicz. Lbir1325a transferts de fluides et d'énergie pour les bioingénieurs. 2018.

[39] F. J. Veihmeyer and A. H. Hendrickson. Methods of measuring field capacity and permanent wilting percentage of soils. *Soil Science*, 68:75–94, 1949.

# 9 Webography

[40] Will Badr. 5 ways to detect outliers/anomalies that every data scientist should know (python code). https://towardsdatascience.com/5-ways-to-detect-outliers-that-every-data-scientist-should-know-python-code-70a54335a623, 2019. Consulté le 20/03/2020.

[41] Jacques Chaurette. Valeurs de rugosité de tuyaux. https://www.pumpfundamentals.com/download-free/pipe_rough_values-fr.pdf, 2003. Consulté le 26/04/2020.

[42] BEI ERE. Penman monteith et méthode fao. http://hmf.enseeiht.fr/travaux/bei/beiere/content/formule-de-penman-monteith, *n.d.*

[43] FAO. Chapter 1 - introduction to evapotranspiration. http://www.fao.org/3/x0490e/x0490e04.htm#determining%20evapotranspiration, *n.d.* Consulté le 15/03/2020.

[44] FAO. Chapter 4 - determination of eto. http://www.fao.org/3/X0490E/x0490e08.htm#TopOfPage, *n.d.* Consulté le 15/03/2020.

[45] FAO. Chapter 6 - etc - single crop coefficient (kc). http://www.fao.org/3/X0490E/x0490e0b.htm#chapter%206%20%20%20etc%20%20%20single%20crop%20coefficient%20(kc), *n.d.* Consulté le 16/03/2020.

[46] FAO. Chapter 8 - etc under soil water stress conditions. http://www.fao.org/3/x0490e/x0490e0e.htm#soil%20water%20availability, *n.d.* Consulté le 18/03/2020.

[47] Sara Garré, Mathieu Javaux, and Joost Wellens. Les techniques d'irrigation. https://www.fun-mooc.fr/courses/course-v1:ulg+108010+session02/info, 2020. Consulté le 15/05/2020.

[48] National Geographic. Disparition de la mer d'aral : les causes d'un désastre écologique. https://www.nationalgeographic.fr/environnement/2014/10/disparition-de-la-mer-daral-les-causes-dun-desastre-ecologique, *n.d.* Consulté le 10/04/2020.

[49] Anne-Laure Jaquemart and Charlotte Descamps. Flore écologique de belgique : banque d'images. https://aidealareussite.uclouvain.be/course/view.php?id=58, 2018. Consulté le 11/04/2020.

[50] United Nations. DÉclaration de rio sur l'environnement et le dÉveloppement principes de gestion des forÊts. https://www.un.org/french/events/rio92/rio-fp.htm, *n.d.* Consulté le 27/04/2020.

[51] ONU. 2,5 milliards de personnes de plus habiteront dans les villes d'ici 2050 (onu). https://news.un.org/fr/story/2018/05/1014202, *n.d.* Consulté le 25/04/2020.

[52] Ooreka. Jardinage : plantes. https://jardinage.ooreka.fr/plante/voir/, *n.d.* Consulté le 11/04/2020.

[53] Dessers Pierre. Les pertes de charge dans les tuyauteries initiation au calcul. https://www.econologie.com/fichiers/partager3/13172900930rmSDs.pdf, *n.d.* Consulté le 26/04/2020.

[54] SPGE. L'eau dans le monde. http://www.spge.be/fr/l-eau-dans-le-monde.html?IDC=1300, *n.d.* Consulté le 15/04/2020.

# 10 Appendix

## Appendix A : Description of the AKUINO central

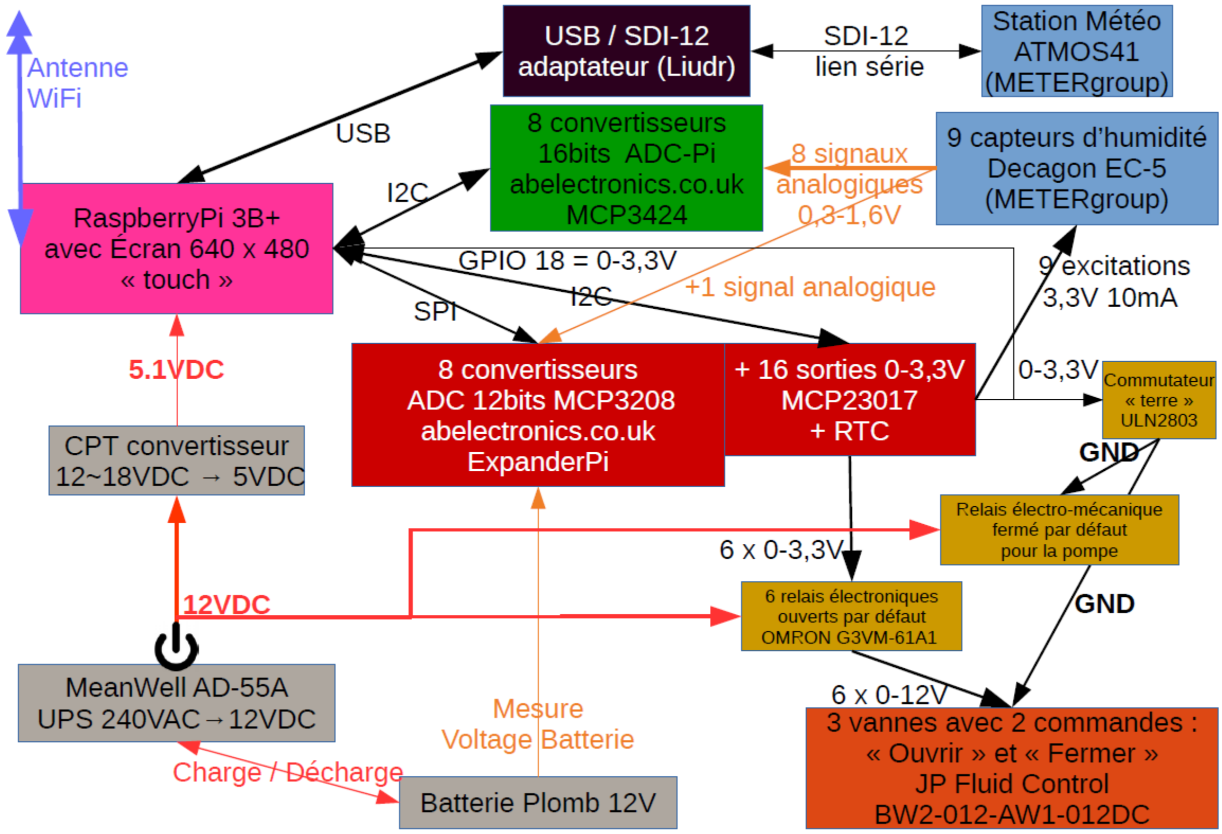



Figure 35: **Scheme of the green wall module and position of the dripping system and the EC-5 sensors.** Source: Christophe Dupriez.

## Appendix B : Python code (wat3.py)

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-

# ------------------------------------------------------------------------------------------
# DESCRIPTION: Automated irrigation program
# AUTHORS: Groupe 3
user = "GW3"
# Place where the code should run
#test = True      # True to run the code locally
test = False    # False to implement the code on the server

# ------------------------------------------------------------------------------------------
# /!\ PARAMETERS /!\:
# 1) Irrigation system:
irrig_syst = 'pot'           # Irrigation system on Christophe's balcony
#irrig_syst = 'greenwall'    # Greenwall

# ------------------------------------------------------------------------------------------
# PACKAGES

from datetime import datetime
import time
import json
import math
import os, sys
import socket
import traceback
import urllib2 as urllib
import csv

# ------------------------------------------------------------------------------------------
# 0) LINES OF CODE TO SET UP COMMUNICATION WITH THE SERVER AND THE SENSORS
# ------------------------------------------------------------------------------------------

# ------------------------------------------------------------------------------------------
# 0.1) Ensure to run in the user home directory

if test:
    host = "greenwall.gembloux.uliege.be"
else:
    host = "localhost"
    # Ensure to run in the user home directory
    DIR_BASE = os.path.expanduser("~")
    if not os.path.samefile(os.getcwd(), DIR_BASE):
        os.chdir(DIR_BASE)
    print(os.getcwd())

#
```

```
    ----------------------------------------------------------------------------------
# 0.2)Ensure to be the only instance to run
# Explanation: if another program is running, it gets killed and replaced by this one

    pid = str(os.getpid())
    _lock_socket = socket.socket(socket.AF_UNIX, socket.SOCK_DGRAM)

    try:
        _lock_socket.bind('\0' + user)
        print('Socket ' + user + ' now locked for process #' + pid)
        # Make the current pid available to be able to kill the process...
        open("pid.txt", 'w').write(pid)
    except socket.error:
        current = open("pid.txt", 'r').read()
        print(user + ' lock exists for process #' + current + " : may be you should ./clean.sh !")
        sys.exit()


# ----------------------------------------------------------------------------------
# 0.3) Handling time
# Explanation: EPOCH time is the number of seconds since 1/1/1970

def get_timestamp():
    return int(time.time())

# Transform an EPOCH time in a lisible date (for Grafana)
def formatDate(epoch):
    dt = datetime.fromtimestamp(epoch)
    return dt.isoformat()

# Transform an EPOCH time in a lisible date (for Grafana)
def formatDateGMT(epoch):
    dt = datetime.fromtimestamp(epoch - (2 * 60 * 60))  # We are in summer and in Belgium !
    return dt.isoformat()

delimiters = ' \t\n\r\"\''

# ----------------------------------------------------------------------------------
# 0.4) Getting the list of all available sensors

dataFile = None
try:  # urlopen not usable with "with"
    url = "http://" + host + "/api/grafana/search"
    dataFile = urllib.urlopen(url, json.dumps(""), 20)
    result = json.load(dataFile)
    #for index in result:
        #print(index)
except:
    print(u"URL=" + (url if url else "") + \
          u", Message=" + traceback.format_exc())
if dataFile:
    dataFile.close()

# ----------------------------------------------------------------------------------
```

```
# 1) IRRIGATION
# ---------------------------------------------------------------------------------------------------

# Scheme: collecting sensor readings, taking a decision to irrigate or not and sending the instructions to the valves
# Output: data file with one column with the Linux EPOCH time and valve state (0=closed, 1=opened)

# ---------------------------------------------------------------------------------------------------
# 1.1) Plan A : Irrigation based on humidity sensors readings
# ---------------------------------------------------------------------------------------------------

"""
sensors used:    - HUM7 : first humidity sensor [V]
                 - HUM8 : second humidity sensor [V]
                 - HUM9 : third humidity sensor [V]
                 - SDI7 : air temperature [ C ]
"""
print (
"""####################################
PLAN A
####################################"""
    )

while (True):

# ---------------------------------------------------------------------------------------------------
# 1.1.1) Reading humidity sensor values of the last 5 minutes (5 minutes of 60 seconds)

    dataFile = None
    try:  # urlopen not usable with "with"
        url = "http://" + host + "/api/grafana/query"
        now = get_timestamp()
        gr = {'range': {'from': formatDateGMT(now - (1 * 5 * 60)), 'to': formatDateGMT(now)}, \
              'targets': [{'target': 'HUM7'}, {'target': 'HUM8'}, {'target': 'HUM9'}, {'target': 'SDI7'}]}
        data = json.dumps(gr)
        #print(data)
        dataFile = urllib.urlopen(url, data, 20)
        result = json.load(dataFile)
        if result:
            #print(result)
            for target in result:
                # print target
                index = target.get('target')
                for datapoint in target.get('datapoints'):
                    value = datapoint[0]
                    stamp = datapoint[1] / 1000
                    #print(index + ": " + formatDate(stamp) + " = " + str(value))

    except:
```

```
        print(u"URL=" + (url if url else "") + \
              u", Message=" + traceback.format_exc())
    if dataFile:
        dataFile.close()

    # Build lists
    Vraw7 = []
    Vraw8 = []
    Vraw9 = []
    TempAir = []
    length_result = len(result[0].get('datapoints'))
    for i in range(0, length_result):
        Vraw7.append(result[0].get('datapoints')[i][0])
        Vraw8.append(result[1].get('datapoints')[i][0])
        Vraw9.append(result[2].get('datapoints')[i][0])
        TempAir.append(result[3].get('datapoints')[i][0])

    print '__________________________________'
    print 'Humidity sensor readings (raw values)'

    print 'HUM7 [V]:', Vraw7
    print 'HUM8 [V]:', Vraw8
    print 'HUM9 [V]:', Vraw9

#
    ---------------------------------------------------------------------------------------------
# 1.1.2) Choose to use Plan A or not
    # Conditions:    - No NaN values
    #                - No outliers
    #                - Standard deviation lower than humidity sensors uncertainty

    # ---------------------------------------------------------------------------
    # a) Parameters

    # # Minimal number of humidity sensors that meet the conditions
    NbHumMin = 2
    # Humidity sensor uncertainty[-]
    hum_uncert = 0.03

    # --------------------------------------------------------------------------
    # b) Check for NaN values

    # Find NaN values
    Vraw7_NaN = []
    Vraw8_NaN = []
    Vraw9_NaN = []
    TempAir_NaN = []
    for i in range(0, length_result):
        Vraw7_NaN.append(math.isnan(Vraw7[i]))
        Vraw8_NaN.append(math.isnan(Vraw8[i]))
        Vraw9_NaN.append(math.isnan(Vraw8[i]))
        TempAir_NaN.append(math.isnan(TempAir[i]))

    print '__________________________________'
    print 'Presence of NaN values'

    print 'HUM7:', Vraw7_NaN.count(True)
    print 'HUM8:', Vraw8_NaN.count(True)
    print 'HUM9:', Vraw9_NaN.count(True)

    # --------------------------------------------------------------------------
```

```
# c) Check for outliers (z-scores)

# ---------------------------------------------------------------------------
# d) Compute standard deviation

# mean function
def std(list_data):

    length_list = len(list_data)
    # mean
    mean = math.fsum(list_data)/length_list                  # Compute mean

    # standard deviation
    var = 0  # Initialize variance
    for j in range(0, length_list):
        var += (list_data[i] - mean) ** 2 / length_list  # Compute variance
    std = math.sqrt(var) / mean  # Compute standard deviation

    return std

std7 = std(Vraw7)
std8 = std(Vraw8)
std9 = std(Vraw9)
print '_________________________________'
print 'Standard deviation'

print 'Threshold [-]:', hum_uncert
print 'HUM7:', std7
print 'HUM8:', std8
print 'HUM9:', std9

# ---------------------------------------------------------------------------
# e) Results of the checks

# Check conditions for each sensor
conditionA = []                                        # List with 1 if OK and 0 if not OK
print '___________________________________'
print "Are humidity sensors' readings usable?"

# HUM7
if (
        all(x == False for x in Vraw7_NaN) and        # No NaN values
        (std7 < hum_uncert)                           # Standard deviation < threshold
        ):
    conditionA.append(1)
    print 'HUM7 can be used'
else:
    conditionA.append(0)
    print 'HUM7 can not be used'

# HUM8
if (
        all(x == False for x in Vraw8_NaN) and        # No NaN values
        (std8 < hum_uncert)                           # Standard deviation < threshold
):
    conditionA.append(1)
    print 'HUM8 can be used'
else:
    conditionA.append(0)
```

```
        print 'HUM8 can not be used'

    # HUM9
    if (
            all(x == False for x in Vraw9_NaN) and      # No NaN values
            (std9 < hum_uncert)                         # Standard deviation < threshold
    ):
        conditionA.append(1)
        print 'HUM9 can be used'
    else:
        conditionA.append(0)
        print 'HUM9 can not be used'

    # -------------------------------------------------------------------------
    # f) Choose to use humidity sensors or not

    if conditionA.count(1) >= NbHumMin:
        print("=> Plan A can be run")

# ----------------------------------------------------------------------------------------------
# 1.1.3) Convert analogous signal [Volts] into volumetric water content [cm3/cm3] (if Plan A chosen)

        def calib(Vraw, eq_type):
            '''
            calib converts the raw signal of an EC-5 sensor with an excitation voltage of 5V to volumetric soil moisture

            Input:
            ------
            V_raw: humidity sensor readings array                   [V]
            eq_type: specify which calibration equation is used
                - 'CB': C dric Bernard, 2018 - Zinco substrate
                  From: TFE of Cedric Bernard
                - 'meter_2.5V_scaled': METER group manual pot soil equation (scaled)
                - 'licor_5V': calibration equation for 5V excitation
                  From: LICOR, 8100_TechTip_EC-5_Probe_Connection_TTP24.pdf
            Output:
            ------
            VWC: volumetric water content array                     [cm3/cm3]
            '''
            if eq_type == 'CB':
                VWC = (0.3524 * Vraw - 0.1554) / (Vraw - 0.3747)
            elif eq_type == 'meter_2.5V_scaled':
                VWC = (8.5 * 0.1 * (Vraw / 2)) - 0.24
            elif eq_type == 'licor_5V':
                VWC = (-3.14E-07 * (Vraw / 1000) ^ 2) + (1.16E-03 * Vraw / 1000) - 6.12E-01
            else:
                VWC = (1.3 * (Vraw / 2)) - 0.348

            return VWC

        # Calibration equation used
        eq_type = 'CB'
        # Calculation
        VWC7 = []
        VWC8 = []
        VWC9 = []
```

```
        for i in range(0, length_result):
            VWC7.append(calib(Vraw7[i], eq_type))
            VWC8.append(calib(Vraw8[i], eq_type))
            VWC9.append(calib(Vraw9[i], eq_type))

        print'___________________________________'
        print'Volumetric water content: calibration equation'
        print 'HUM7 [cm3/cm3]:', VWC7
        print 'HUM8 [cm3/cm3]:', VWC8
        print 'HUM9 [cm3/cm3]:', VWC9

# -----------------------------------------------------------------------------------------------------
# 1.1.4) Correct temperature influence on raw signal (if Plan A chosen)
        # Condition: no NaN values in the humidity sensor temperature array
        if all(x == False for x in TempAir_NaN):

            def correctTemp(VWC, TempAir, sensor):
                '''
                correctTemp corrects VWC  with sensor temperature
                Source: Cobos, Doug, Colin Campbell, and Decagon Devices. n.d. Correcting  Temperature Sensitivity of
                        ECH 2 O Soil Moisture Sensors Strategy 1 : Multiple Regression Analysis,   no. 1.
                ------
                Input:
                VWC: humidity sensor readings array                     [V]
                TempAir: air temperature array                          [ C ]
                sensor: sensor name
                    - HUM7
                    - HUM8
                    - HUM9
                ------
                Output:
                VWC_corrected: corrected VWC array                      [cm3/cm3]
                ------
                Equation:
                VWC_corrected = C1* VWC + C2 * TempAir + C3
                '''

                if sensor =='HUM7':
                    C = [0.673092668753705, 0.000174348978687542, 0.0840794823052036]
                elif sensor =='HUM8':
                    C = [0.0326500322594179, -0.000130977585771525, 0.202491414445767]
                elif sensor =='HUM9':
                    C = [0.491143905041849, -0.0000577534352700035, 0.128819057287295]

                VWC_corrected = C[0] * VWC + C[1] * TempAir + C[2]

                return VWC_corrected

            # Calculation
            for i in range(0, length_result):
                VWC7[i] = correctTemp(VWC7[i], TempAir[i], 'HUM7')
                VWC8[i] = correctTemp(VWC8[i], TempAir[i], 'HUM8')
                VWC9[i] = correctTemp(VWC9[i], TempAir[i], 'HUM9')

            print'___________________________________'
            print'Volumetric water content: temperature correction'
```

```
            print 'HUM7 [cm3/cm3]:', VWC7
            print 'HUM8 [cm3/cm3]:', VWC8
            print 'HUM9 [cm3/cm3]:', VWC9

        else:
            print'__________________________________'
            print'Volumetric water content: temperature correction'
            print 'NaN detected : TempAir can not be used'


#
    ----------------------------------------------------------------------------------------------

# 1.1.5) Irrigation based on humidity sensors (if Plan A chosen)

        print'__________________________________'
        print'Irrigation'

        # calculate the average water content
        def mean(numbers):
            return float(sum(numbers)) / max(len(numbers), 1)

        theta_mean = []
        theta_mean.append(mean(VWC7))
        theta_mean.append(mean(VWC8))
        theta_mean.append(mean(VWC9))

        print 'Mean water content [cm3/cm3]:', theta_mean

        # Parameters
        if irrig_syst == 'pot':
            A_tot = 1920    # box area                          [cm2]
            H_tot = 12      # box height                        [cm]
            Q_tot = 1000    # discharge in the main pipe        [cm3/hr]

            # Heigth
            H7 = H_tot   # height of the first zone             [cm]
            H8 = H_tot   # height of the second zone            [cm]
            H9 = H_tot   # height of the third zone             [cm]
            H = [H7, H8, H9]   # zone height array              [cm]

            # Area of each box zone
            # Each sensor is expected to cover the same horizontal area
            A7 = A_tot/3   # height of the first zone           [cm]
            A8 = A_tot/3   # height of the second zone          [cm]
            A9 = A_tot/3   # height of the third zone           [cm]
            A = [A7, A8, A9]   # zone area array                [cm]

            # Discharge in pipes
            # Discharge is considered identical in the three zones
            Q7 = Q_tot/3   # discharge in the first zone        [cm3/hr]
            Q8 = Q_tot/3   # discharge in the second zone       [cm3/hr]
            Q9 = Q_tot/3   # discharge in the third zone        [cm3/hr]
            Q = [Q7, Q8, Q9]   # discharge array                [cm3/hr]

            # Water content at field capacity
            theta_fc7 = 0.285   # water content at field capacity in the first zone
                   [cm3/cm3]
            theta_fc8 = 0.225   # water content at field capacity in the first zone
                   [cm3/cm3]
            theta_fc9 = 0.27   # water content at field capacity in the first zone
                   [cm3/cm3]
```

```
        theta_fc = [theta_fc7, theta_fc8, theta_fc9]  # water content at field capacity array   [cm3/cm3]

    elif irrig_syst == 'greenwall':
        A_tot = 2200 * 13   # wall area                 [cm2]
        H_tot = 107         # wall height               [cm]

        # Heigth
        # The wall is divided into three layers
        H7 = 30     # height of the top layer           [cm]
        H8 = 30     # height of the intermediary layer  [cm]
        H9 = 37     # height of the bottom layer        [cm]
        H = [H7, H8, H9]      # layer height array      [cm]

        # Area
        A7 = A_tot  # area of the top layer             [cm2]
        A8 = A_tot  # area of the intermediary layer    [cm2]
        A9 = A_tot  # area of the bottom layer          [cm2]
        A = [A7, A8, A9]  # layer height array          [cm2]

        # Discharge in pipes
        Q7 = 432000          # discharge in the top layer            [cm3/hr]
        Q8 = 446400          # discharge in the intermediary layer   [cm3/hr]
        Q9 = 460800          # discharge in the bottom layer         [cm3/hr]
        Q = [Q7, Q8, Q9]     # discharge array                       [cm3/hr]

        # Water content at field capacity
        theta_fc7 = 0.285   # water content at field capacity in the first layer         [cm3/cm3]
        theta_fc8 = 0.285   # water content at field capacity in the intermediary layer         [cm3/cm3]
        theta_fc9 = 0.285   # water content at field capacity in the bottom layer         [cm3/cm3]
        theta_fc = [theta_fc7, theta_fc8, theta_fc9]  # water content at field capacity array   [cm3/cm3]

    print 'Water content at field capacity [cm3/cm3]:', theta_fc

    # Irrigation time => Water needs in the three layers
    time_irrig = []
    for i in range(0, len(theta_fc)):
        vol = (theta_fc[i] - theta_mean[i]) * A[i] * H[i]  # Irrigation volume [cm3]
        if vol < 0:
            time_irrig.append(0)
        else:
            time_irrig.append(int(vol / Q[i] * 3600))  # Irrigation time [s]
        del vol

    # Find the maximal irrigation time
    index_OK = [f for f, e in enumerate(conditionA) if e == 1]  # Index of the sensor that meet the conditions
    time_irrigOK = []                                           # Initialization
    for i in range(0, len(index_OK)):
        time_irrigOK.append(time_irrig[index_OK[i]])            # Irrigation time corresponding to sensor that meets the conditions

    index_max = time_irrigOK.index(max(time_irrigOK))           # Index of the maximal irrigation time

    # Irrigation
    timestamp = get_timestamp()
```

```
        # erase the current file and open the valve in 30 seconds
        open("valve.txt", 'w').write(str(timestamp + 30) + ";1\n")
        # append to the file and close the valve time_irrig later
        open("valve.txt", 'a').write(str(timestamp + 30 + time_irrigOK[index_max]) + ";0\n")
        print 'Open the valve for', time_irrigOK[index_max], 'seconds'

        # Processed finished
        print("valve.txt ready.")

        # Record action
        if os.path.isfile('history.txt'):  # If file history.txt already exists
            open("history.txt", 'a').write(str(timestamp) + ";A\n")      # Fill file
        else:  # If file history.txt does not exist
            file("history.txt", "w+")                                    # Create file
            open("history.txt", 'a').write(str(timestamp) + ";A\n")      # Fill file

    else:
        print("Go to plan B")

# -----------------------------------------------------------------------------------------
# 1.2) Plan B : Irrigation based on ET estimation
# -----------------------------------------------------------------------------------------
        print '####################################'
        print 'PLAN B'
        print '####################################'

        """
        sensors used:
        - SDI0 : solar radiation        [W/m2]
        - SDI1 : rain                   [mm/h]
        - SDI4: wind speed              [m/s]
        - SDI7 : air temperature        [ C ]
        - SDI8: vapor pressure          [kPa]
        - SDI9 : atmospheric pressure   [kPa]
        - SDI10 : relative humidity     [%]
        """

# -----------------------------------------------------------------------------------------
# 1.2.1) Reading sensor values of the last 24 hours (24 hours of 60 minutes of 60 seconds)

        dataFile = None
        try:  # urlopen not usable with "with"
            url = "http://" + host + "/api/grafana/query"
            now = get_timestamp()
            gr = {'range': {'from': formatDateGMT(now - (24 * 60 * 60)), 'to': formatDateGMT(now)},
                  'targets': [{'target': 'SDI0'}, {'target': 'SDI1'}, {'target': 'SDI4'}, {'target': 'SDI7'},
                              {'target': 'SDI8'}, {'target': 'SDI9'}, {'target': 'SDI10'}]}
            data = json.dumps(gr)
            # print(data)
            dataFile = urllib.urlopen(url, data, 20)
            result = json.load(dataFile)
```

```
            if result:
                # print(result)
                for target in result:
                    # print target
                    index = target.get('target')
                    for datapoint in target.get('datapoints'):
                        value = datapoint[0]
                        stamp = datapoint[1] / 1000
                        # print(index + ": " + formatDate(stamp) + " = " + str(value)
    )
        except:
            print(u"URL=" + (url if url else "") +
                  u", Message=" + traceback.format_exc())
        if dataFile:
            dataFile.close()

        # Build lists
        solRad = []
        rain = []
        windSpeed = []
        tempAir = []
        pressVap = []
        pressAtm = []
        humRel = []
        length_result = len(result[0].get('datapoints'))
        for i in range(0, length_result):
            solRad.append(result[0].get('datapoints')[i][0])
            rain.append(result[1].get('datapoints')[i][0])
            windSpeed.append(result[2].get('datapoints')[i][0])
            tempAir.append(result[3].get('datapoints')[i][0])
            pressVap.append(result[4].get('datapoints')[i][0])
            pressAtm.append(result[5].get('datapoints')[i][0])
            humRel.append(result[6].get('datapoints')[i][0])

        print '____________________________________________'
        print 'Sensor values'
        print 'Solar radiation [W/m2]:', solRad

#
    -----------------------------------------------------------------------------------------

# 1.2.2) Choose to use Plan A or not
# Conditions:   - No NaN values

        # ---------------------------------------------------------------------------
        # a) Parameters

        # --------------------------------------------------------------------------
        # b) Check for NaN values

        # Find NaN values
        solRad_NaN = []
        rain_NaN = []
        windSpeed_NaN = []
        tempAir_NaN = []
        pressVap_NaN = []
        pressAtm_NaN = []
        humRel_NaN = []
        for i in range(0, length_result):
            solRad_NaN.append(math.isnan(solRad[i]))
            rain_NaN.append(math.isnan(rain[i]))
            windSpeed_NaN.append(math.isnan(windSpeed[i]))
```

```
            tempAir_NaN.append(math.isnan(tempAir[i]))
            pressVap_NaN.append(math.isnan(pressVap[i]))
            pressAtm_NaN.append(math.isnan(pressAtm[i]))
            humRel_NaN.append(math.isnan(humRel[i]))

        print '____________________________________________'
        print 'Presence of NaN values'
        print 'Solar radiation:', solRad_NaN.count(True)
        print 'Rain:', rain_NaN.count(True)
        print 'Wind speed:', windSpeed_NaN.count(True)
        print 'Air temperature:', tempAir_NaN.count(True)
        print 'Vapor pressure:', pressVap_NaN.count(True)
        print 'Atmospheric pressure:', pressAtm_NaN.count(True)
        print 'Relative humidity:', humRel_NaN.count(True)

        # ---------------------------------------------------------------------------
        # c) Results of the checks

        # Check conditions for each sensor
        conditionB = []  # List with 1 if OK and 0 if not OK
        print '____________________________________'
        print "Are atmospheric sensors' readings usable?"

        # SDI0 : solar radiation
        if (
                all(x == False for x in solRad_NaN)  # No NaN values
        ):
            conditionB.append(1)
            print ' - SDI0 (solar radiation) can be used'
        else:
            conditionB.append(0)
            print ' - SDI0 (solar radiation) can not be used'

        # SDI4: wind speed
        if (
                all(x == False for x in windSpeed_NaN)  # No NaN values
        ):
            conditionB.append(1)
            print ' - SDI4 (wind speed) can be used'
        else:
            conditionB.append(0)
            print ' - SDI4 (wind speed) can not be used'

        # SDI7 : air temperature
        if (
                all(x == False for x in tempAir_NaN)  # No NaN values
        ):
            conditionB.append(1)
            print ' - SDI7 (air temperature) can be used'
        else:
            conditionB.append(0)
            print ' - SDI7 (air temperature) can not be used'

        # SDI8: vapor pressure
        if (
                all(x == False for x in pressVap_NaN)  # No NaN values
        ):
            conditionB.append(1)
            print ' - SDI8 (vapor pressure) can be used'
        else:
            conditionB.append(0)
            print ' - SDI8 (vapor pressure) can not be used'
```

```

        # SDI9 : atmospheric pressure
        if (
                all(x == False for x in pressAtm_NaN)  # No NaN values
        ):
            conditionB.append(1)
            print ' - SDI9 (atmospheric pressure) can be used'
        else:
            conditionB.append(0)
            print ' - SDI9 (atmospheric pressure) can not be used'

        # SDI10 : relative humidity
        if (
                all(x == False for x in humRel_NaN)  # No NaN values
        ):
            conditionB.append(1)
            print ' - SDI10 (relative humidity) can be used'
        else:
            conditionB.append(0)
            print ' - SDI10 (relative humidity) can not be used'

        # ---------------------------------------------------------------------------
        # d) Choose to use or not atmospheric sensors
        if all(x == 1 for x in conditionB):
            print("Plan B can be run")

# -----------------------------------------------------------------------------------------------
# 1.2.3) Convert variables for Penman equation
            print '___________________________________________'
            print 'Data conversion for Penman equation'

            # global radiation (SDI0): [W/m2] -> [MJ/m2/day]
            Rn = 0                                                           # Sum initialization
            Rn_list = []
            length_result = len(result[0].get('datapoints'))
            for i in range(0, length_result):
                Rn += 60 * solRad[i]                                         # Calculate sum [J/m2/day]
                                                                             # *60 to get the energy per minute [J/min]
            Rn = Rn / (1E06)                                                 # Convert units [MJ/m2/day]
            print'Rn =', Rn, 'MJ/m2/day'

            # wind speed (SDI4) : mean value over 24 hours [m/s]
            u_sum = 0                                                        # Sum initialization
            for i in range(0, length_result):
                u_sum += windSpeed[i]                                        # Calculate sum [m/s]
            u = u_sum / length_result                                        # Calculate mean [m/s]
            print'u =', u, 'm/s'

            # temperature (SDI7) : mean value over 24 hours [ C ]
            T_sum = 0                                                        # Sum initialization
            for i in range(0, length_result):
                T_sum += tempAir[i]                                          # Calculate sum [
```

```
 C ]
        T = T_sum / length_result                           # Calculate mean [ C ]
        print'T =', T, ' C '

        # actual vapor pressure (SDI8) : mean value over 24 hours [kPa]
        e_sum = 0                                           # Sum initialization
        for i in range(0, length_result):
            e_sum += pressVap[i]                            # Calculate sum [kPa]
        e_a = e_sum / length_result                         # Calculate mean [kPa]
        print'e_a =', e_a, 'kPa'

        # atmospheric pressure (SDI9) : mean value over 24 hours [kPa]
        p_sum = 0                                           # Sum initialization
        for i in range(0, length_result):
            p_sum += pressAtm[i]                            # Calculate sum [kPa]
        p = p_sum / length_result                           # Calculate mean [kPa]
        print'p =', p, 'kPa'

        # relative humidity (SDI10) : mean value over 24 hours [%]
        RH_sum = 0                                          # Sum initialization
        for i in range(0, length_result):
            RH_sum += humRel[i]                             # Calculate sum [%]
        RH = RH_sum / length_result                         # Calculate mean [%]
        print'RH =', RH, '%'

        # rain (SDI1) : [mm/hr] -> [mm]
        P = 0                                               # Sum initialization
        for i in range(0, length_result):
            P += rain[i] / 60                               # Calculate sum [mm]
        print'P =', P, 'mm'

# ----------------------------------------------------------------------------------------------
# 1.2.4) Calculate parameters of Penman equation

        """
        - e_sat: saturation vapour pressure [kPa]
        - gamma: psychrometric constant [kPa/ C ]
        - delta: slope of the vapour pressure curve [kPa/ C ]
        """
        print '____________________________________________'
        print 'Parameters of Penman equation'

        # saturation vapour pressure [kPa]
        e_sat = 0.6108 * math.exp((17.27 * T) / (T + 273.3))
        print'e_sat =', e_sat, 'kPa'
        # psychrometric constant [kPa/ C ]
        gamma = 0.665 * p * 1E-03
        print'gamma =', gamma, 'kPa/ C '
```

```
            # delta [kPa/ C ]
            delta = (4098 * e_sat) / (T + 237.3) ** 2
            print'delta =', delta, 'kPa/ C '

# ------------------------------------------------------------------------------------------
# 1.2.5) Estimate ET

            print '___________________________________________'
            print 'ET estimation of the previous day'

            # ET0
            cst = 900
            ET0 = (0.408 * delta * Rn + gamma * cst / (T + 273) * u * (e_sat - e_a)) / (delta + gamma * (1 + 0.34 * u))
            print 'ET0 =', ET0, 'mm/day'

            # Kc
            if irrig_syst == 'pot':
                Kc = 0.5    # cultural coefficient of rocket [-]
            elif irrig_syst == 'greenwall':
                Kc = 0.6    # global cultural coefficient of the wall [-]

            # Etc
            ET = Kc * ET0    # Daily evapotranspiration [mm/day]
            print 'ETc =', ET, 'mm/day'

# ------------------------------------------------------------------------------------------
# 1.2.6) Irrigation

            print'_________________________________'
            print'Irrigation'

            if irrig_syst == 'pot':
                # Parameters
                A_rain_tot = 1920   # rainfall area                                 [cm2]
                A_ET_tot = A_rain_tot   # ET area                                   [cm2]
                Q_tot = 1000    # discharge in the main pipe                        [cm3/hr]

                # Rainfall area
                A_rain7 = A_rain_tot/3  # rainfall area of the first zone           [cm2]
                A_rain8 = A_rain_tot/3  # rainfall area of the second zone          [cm2]
                A_rain9 = A_rain_tot/3  # rainfall area of the third zone           [cm2]
                A_rain = [A_rain7, A_rain8, A_rain9]  # zone rainfall area array    [cm2]

                # ET area
                A_ET7 = A_ET_tot / 3  # ET area of the first zone                   [cm2]
                A_ET8 = A_ET_tot / 3  # ET area of the second zone                  [cm2]
                A_ET9 = A_ET_tot / 3  # ET area of the third zone                   [cm2]
```

```
            A_ET = [A_ET7, A_ET8, A_ET9]  # zone ET area array                    [cm2]

            # Discharge in pipes
            # Discharge is considered identical in the three zones
            Q7 = Q_tot / 3  # discharge in the first zone                          [cm3/hr]
            Q8 = Q_tot / 3  # discharge in the second zone                         [cm3/hr]
            Q9 = Q_tot / 3  # discharge in the third zone                          [cm3/hr]
            Q = [Q7, Q8, Q9]  # discharge array                                    [cm3/hr]

        elif irrig_syst == 'greenwall':
            # Parameters
            A_rain_tot = 220*13    # rainfall area (top)    [cm2]
            A_cell = 10 * 10    # wall cell area    [cm2]
            cell_Nb = 26 * 10   # wall cell number    [cm2]
            A_ET_tot = A_rain_tot + A_cell*cell_Nb   # ET area    [cm2]
            H_tot = 107    # wall height    [cm]

            # Rainfall area
            A_rain7 = A_rain_tot      # rainfall area of the top layer    [cm2]
            A_rain8 = 0               # rainfall area of the intermediary layer    [cm2]
            A_rain9 = 0               # rainfall area of the bottom layer    [cm2]
            A_rain = [A_rain7, A_rain8, A_rain9]  # layer rainfall area array    [cm2]

            # Heigth
            # The wall is divided into three layers
            H7 = 30  # height of the top layer    [cm]
            H8 = 30  # height of the intermediary layer    [cm]
            H9 = 37  # height of the bottom layer    [cm]
            H = [H7, H8, H9]  # layer height array    [cm]

            # ET area
            A_ET7 = A_ET_tot * H7/H_tot     # ET area of the top layer    [cm2]
            A_ET8 = A_ET_tot * H8/H_tot     # ET area of the intermediary layer    [cm2]
            A_ET9 = A_ET_tot * H9/H_tot     # ET area of the bottom layer    [cm2]
            A_ET = [A_ET7, A_ET8, A_ET9]       # layer ET area array    [cm2]

            # Discharge in pipes
            Q7 = 432000  # discharge in the top layer    [cm3/hr]
            Q8 = 446400  # discharge in the intermediary layer
```

```
 [cm3/hr]
                Q9 = 460800  # discharge in the bottom layer [cm3/hr]
                Q = [Q7, Q8, Q9]  # discharge array [cm3/hr]

            # Irrigation time => Water needs in the three layers
            time_irrig = []
            for i in range(0, len(Q)):
                vol = ET/10 * A_ET[i] - P/10 * A_rain[i]                       # Irrigation volume [cm3]
                if vol < 0:
                    time_irrig.append(0)
                else:
                    time_irrig.append(int(vol / Q[i] * 3600))  # Irrigation time [s]
                del vol

            # Find the maximal irrigation time
            index_max = time_irrig.index(max(time_irrig))  # Index of the maximal irrigation time

            # Valve command
            timestamp = get_timestamp()
            # erase the current file and open the valve in 30 seconds
            open("valve.txt", 'w').write(str(timestamp + 30) + ";1\n")
            # append to the file and close the valve time_irrig later
            open("valve.txt", 'a').write(str(timestamp + 30 + time_irrig[index_max]) + ";0\n")
            print 'Open the valve for', time_irrig[index_max], 'seconds'

            print("valve.txt ready.")

            # Record action
            if os.path.isfile('history.txt'):  # If file history.txt already exists
                open("history.txt", 'a').write(str(timestamp) + ";B\n")  # Fill file
            else:  # If file history.txt does not exist
                file("history.txt", "w+")  # Create file
                open("history.txt", 'a').write(str(timestamp) + ";B\n")  # Fill file

        else:
            print("Go to plan C")

# ------------------------------------------------------------------------------------------------------
# 1.3) Plan C : Irrigation based on ET estimated with previous years' data
# ------------------------------------------------------------------------------------------------------
            print '####################################'
            print 'PLAN C'
            print '####################################'

            # Kc
            if irrig_syst == 'pot':
                Kc = 0.5  # cultural coefficient of rocket [-]
            elif irrig_syst == 'greenwall':
                Kc = 0.6  # global cultural coefficient of the wall [-]

            # Discharge and surfaces
            if irrig_syst == 'pot':
                # Parameters
```

```
            A_ET_tot = 1920   # ET area                                        [cm2]
            Q_tot = 1000    # discharge in the main pipe                       [cm3/hr]

            # ET area
            A_ET7 = A_ET_tot / 3  # ET area of the first zone                  [cm2]
            A_ET8 = A_ET_tot / 3  # ET area of the second zone                 [cm2]
            A_ET9 = A_ET_tot / 3  # ET area of the third zone                  [cm2]
            A_ET = [A_ET7, A_ET8, A_ET9]  # zone ET area array                 [cm2]

            # Discharge in pipes
            # Discharge is considered identical in the three zones
            Q7 = Q_tot / 3  # discharge in the first zone                      [cm3/hr]
            Q8 = Q_tot / 3  # discharge in the second zone                     [cm3/hr]
            Q9 = Q_tot / 3  # discharge in the third zone                      [cm3/hr]
            Q = [Q7, Q8, Q9]  # discharge array                                [cm3/hr]

        elif irrig_syst == 'greenwall':
            # Parameters
            A_cell = 10 * 10    # wall cell area                               [cm2]
            cell_Nb = 26 * 10   # wall cell number                             [cm2]
            A_ET_tot = 220 * 13 + A_cell*cell_Nb   # ET area                   [cm2]
            H_tot = 107    # wall height                                       [cm2]

            # Heigth
            # The wall is divided into three layers
            H7 = 30  # height of the top layer                                 [cm]
            H8 = 30  # height of the intermediary layer                        [cm]
            H9 = 37  # height of the bottom layer                              [cm]
            H = [H7, H8, H9]  # layer height array                             [cm]

            # ET area
            A_ET7 = A_ET_tot * H7/H_tot     # ET area of the top layer         [cm2]
            A_ET8 = A_ET_tot * H8/H_tot     # ET area of the intermediary layer [cm2]
            A_ET9 = A_ET_tot * H9/H_tot     # ET area of the bottom layer      [cm2]
            A_ET = [A_ET7, A_ET8, A_ET9]    # layer ET area array              [cm2]

            # Discharge in pipes
            Q7 = 432000  # discharge in the top layer                          [cm3/hr]
            Q8 = 446400  # discharge in the intermediary layer
```

```
[cm3/hr]
            Q9 = 460800  # discharge in the bottom layer [cm3/hr]
            Q = [Q7, Q8, Q9]  # discharge array [cm3/hr]

        # ET0 file of Ernage
        file = open("ET0_2010_2019.csv", "r")  # open the file
        reader = csv.reader(file, delimiter=";")  # file reading initialization

        # Skip the first two lines
        next(reader)
        next(reader)

        outfile = open('valve.txt', 'w')

        # Transform date into epoch time
        for row in reader:  # loop to go through the reader
            #print (row[0])  # display rows

            # Convert datetime into epoch
            hiredate = row[0]  # select date in the list
            pattern = '%d/%m/%Y %H:%M'  # date format
            epoch = int(time.mktime(time.strptime(hiredate, pattern)))  # convert date to epoch time
            #print epoch

            # Get ET0
            ET0 = float(row[1])      # Daily reference evapotranspiration        [mm/day]

            # Compute ETc
            ET = Kc * ET0            # Daily evapotranspiration                  [mm/day]

            # Irrigation time => Water needs in the three layers
            time_irrig = []
            for i in range(0, len(theta_fc)):
                vol = ET / 10 * A_ET[i]  # Irrigation volume                     [cm3]
                if vol < 0:
                    time_irrig.append(0)
                else:
                    time_irrig.append(int(vol / Q[i] * 3600))  # Irrigation time [s]
                del vol

            # Find the maximal irrigation time
            index_max = time_irrig.index(max(time_irrig))  # Index of the maximal irrigation time

            # open the valve
            outfile.write(str(epoch + 24 * 60 * 60) + ";1\n")
            # append to the file and close the valve the next day (+ 24*60*60) at 00:00 + time_irrig
            outfile.write(str(epoch + 24 * 60 * 60 + time_irrig[index_max]) + ";0\n")

        outfile.close()  # close valve.txt
        print("valve.txt ready.")
        file.close()  # close the file

```

```
            # Record action
            timestamp = get_timestamp()
            if os.path.isfile('history.txt'):    # If file history.txt already exists
                open("history.txt", 'a').write(str(timestamp) + ";C\n")       # Fill file
            else:                                       # If file history.txt does not exist
                file("history.txt", "w+")                                     # Create file
                open("history.txt", 'a').write(str(timestamp) + ";C\n")       # Fill file

    # Update nohup.out file
    sys.stdout.flush()

    # sleep for 24 hours (in seconds)
    time.sleep(24 * 60 * 60)
```