EPITA

ADA

ADArrose: Automatic sprinkler

Written by:
GISTRE - Thomas Delecroix
GISTRE - Cyril Etourneau
GISTRE - Louise Flick
GISTRE - Charles Villard

January 22nd, 2020

Contents

1 Project description

1.1 Introduction

This project intends to make an automated sprinkler based on a STM32F429 board. This project will use ADA language and SPARK verification system.

We all know the trouble to keep a plant. Especially in Paris, where the light is low and where our lives goes to 100km/h. We don't have time to water plants. On the eventuality that we manage to find the necessary time, we will most likely give it too much water or not enough and by the time we find out the plant will be dead. Using the secure ADA language and contract verification, we will be able to create an automatic sprinkler to keep our plants alive and have a beautiful green environment without worrying about it.

1.2 Features

Our automatic sprinkler comes with numerous features. First of all it can be set in different activity modes to best respond to your needs.

- Continuous mode:
  The automatic sprinkler is always active. Once every hour, the soil humidity around the plant is checked with a sensor. If the humidity percentage isn't high enough, the sprinkler will water the plant with the right amount of water to keep it well hydrated. Perfect for fragile plants.

- Economic mode:
  The automatic sprinkler is always active. When the luminosity is low, the soil humidity around the plant is checked once every hour with a sensor. If the humidity percentage isn't high enough, the sprinkler will water the plant with the right amount of water to keep it well hydrated. This is done to avoid losing water due to evaporation. Perfect to minimize the water consumption. Use on plants without special needs.

- Planned mode:
  The automatic sprinkler always active, but can only water the plant during the periods specified by the user. The humidity of the soil is checked at the start of the period and then once every hour. If the humidity percentage isn't high enough, the sprinkler will water the plant with the right amount of water to keep it well hydrated. A last check is done at the end of the period to ensure the well being of the plant. Useful for a tailored experience.

- Punctual mode:
  The automatic sprinkler does one humidity check when starting. If the humidity percentage isn't high enough, the sprinkler will water the plant with the right amount of water to keep it well hydrated. When the process is finished, the system signals to the user that it can be shut down. For those who want to keep some contact with their green friends. Use to reduce the electricity bill.

Numerous data sets about the plant are collected when the sprinkler is in the first 3 modes. The soil humidity and surrounding luminosity are recorded once every hour and the user can have access to these data over the last 24 hours.

Whenever something abnormal is detected in the plant consumption or in the environment, messages will warn him about the peculiar situation. If a problem arises in the system, the user will also be notified.

1.3 Tutorial

This tutorial presents the different steps to set up your system and prevent any problem.

- The system should come fully wired. If anything is amiss, refer to the electrical description of the system or contact our support service.
- Plug the hose to the end of the water pump.
- Bury the other end of the hose about 1cm under the soil at the foot of the plant.
- Fetch a container with enough capacity to manage multiple watering and fill it with water. A container with a cap and a hole for the hose is preferred to avoid getting anything in the water and to avoid evaporation.
- Put the water pump in the filled container.
- Setup the configuration file for the system in adarrose/src/config.ads. Be careful to fill all the fields and to check if the data is in the right unit and coherent. An error will be displayed at the start of the system otherwise. This configuration file is important in order to fit the watering to the needs of your plant as cactus and a tropical plant don't have the same watering needs. The system mode is set is in this file.
- Flash the card, boot the system and your setup is done!

You can now fully enjoy the ADArrose experience and let your plants continue their life worry free. The system will notify you if a problem arises through the screen. You can consult the data about your plant at any time as they are displayed live on the screen. Don't forget to update the date and hour whenever you restart the system.

2 Development process

2.1 High level requirements

- **REQ.1 Capacity.**
  The water container capacity must be known to notify the user when the system has pumped as much as contained in it.

- **REQ.2 Humidity detection.**
  The soil humidity must be recorded once every hour. If the soil humidity doesn't increase after the pump got activated an error must be reported.

- **REQ.3 Watering threshold.**
  A soil humidity threshold must be defined to be able to know when the plant needs water and water it.

- **REQ.4 Flooding.**
  Too much water in the soil must be detected and reported to the user.

- **REQ.5 Abnormal variation.**
  Any abnormal variation in water consumption of the plant must be detected to signal the problem.

- **REQ.6 Compute water volume.**
  The water volume to give to the plant must be computed from the hose diameter, the water pump debit and the soil humidity.

- **REQ.7 Time.**
  The internal clock of the system must be accurate to guarantee the behaviour of the planned mode.

- **REQ.8 Light.**
  The amount of light received by the plant must be recorded once every hour. If the plant doesn't get enough light over 24h, a notification is displayed.

- **REQ.9 Continuous mode.**
  If the system is set in continuous mode, the soil humidity is checked every hour. The plant is watered if the soil humidity is under the threshold defined.

- **REQ.10 Economic mode.**
  If the system is set in economic mode, the soil humidity is checked every hour. The plant is watered if the luminosity is low and the soil humidity is under the threshold defined.
- **REQ.11 Planned mode .**
  If the system is set in planned mode, the soil humidity is checked every hour. The plant is watered at the start and at the end of the active period if the soil humidity is under the threshold defined. The plant is also watered if the current time is in the active period and the hourly check signals that the plant needs water.
- **REQ.12 Punctual mode.**
  If the system is set in punctual mode, the soil humidity is checked once. The plant is watered if the soil humidity is under the threshold defined. A notification must be displayed signalling that the system is done once everything is in order.

2.2 Architecture

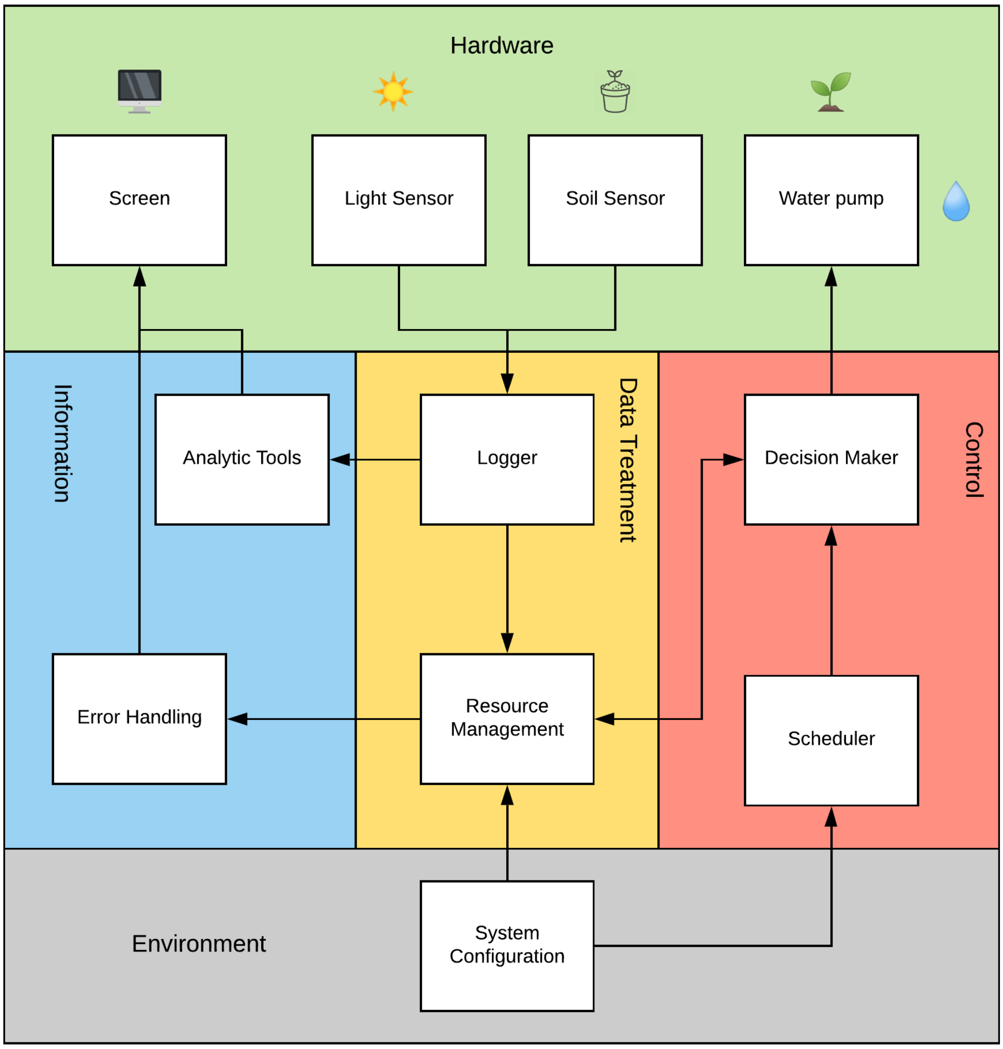


Figure 1: Software architecture

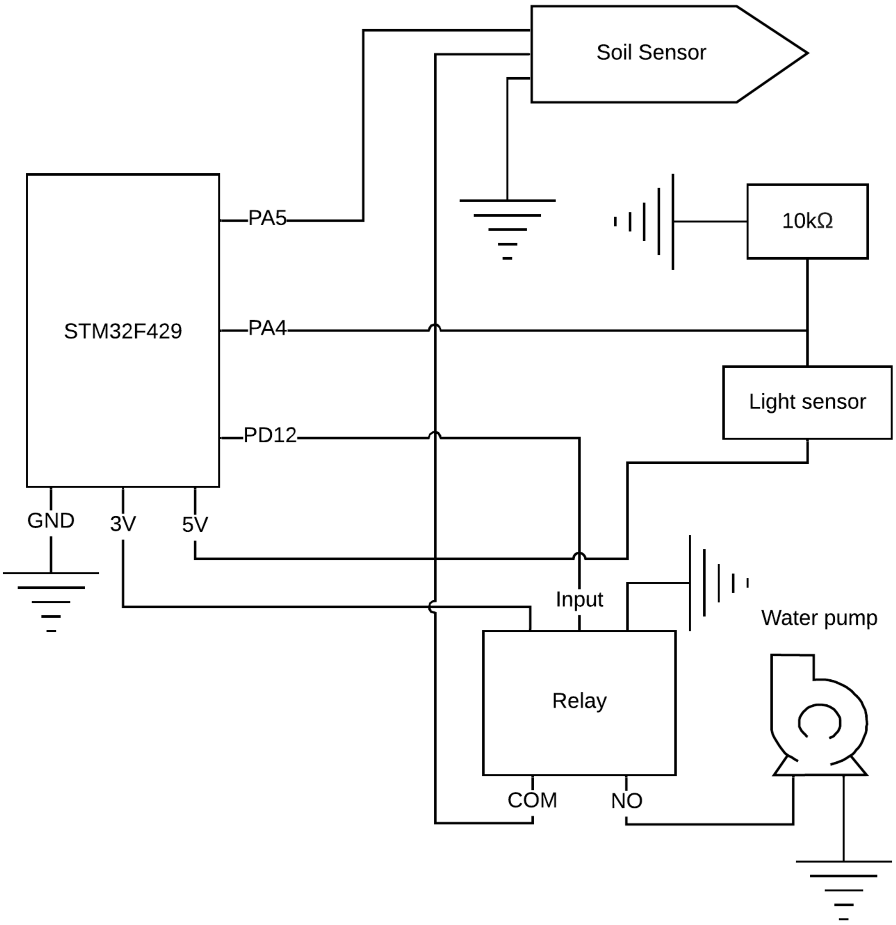


Figure 2: Electronic wiring schematic

2.3 Low level requirements

- **REQ.1.1 Container capacity.**
  The water container capacity must be defined in the configuration file and be high enough for a prolonged use of the system.

- **REQ.1.2 Error container.**
  If the water container capacity isn't defined or isn't enough, an error message must be returned to the user signalling that the value is undefined or not high enough.

- **REQ.1.3 Information container.**
  When a volume of water equal to the water container capacity has been pumped an information message must be returned to the user signalling that the water container must be refilled and the system restarted.

- **REQ.2.1 Error Humidity Detection.**
  If there is no soil humidity variation after the water pump was active, an error message must be returned to the user signalling that a problem occurred in the system and it must be checked.

- **REQ.2.2 Humidity record.**
  The soil humidity values must be recorded once every hour.

- **REQ.3.1 Watering threshold.**
  The watering threshold must be defined in the configuration file.

- **REQ.3.2 Error threshold.**
  If the water threshold isn't defined, an error message must be returned to the user signalling that the value is undefined.

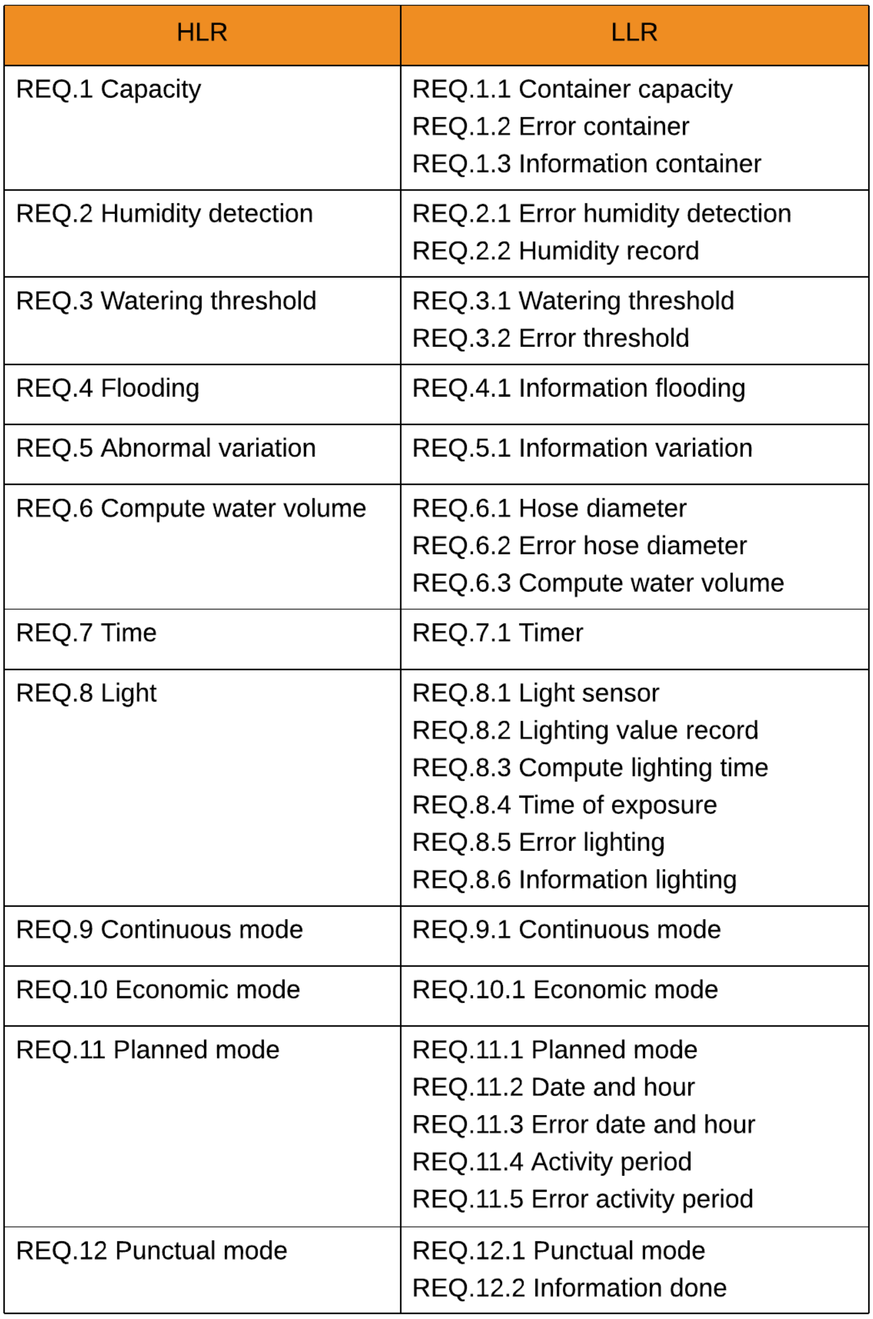
- **REQ.4.1 Information flooding.**
  If the soil humidity is 20% higher than the threshold defined, an error message must be returned to the user signalling that another water source is hydrating the soil.

- **REQ.5.1 Information variation.**
  If the water consumption varies in an abnormal way, an information message must be returned to the user signalling that there is a problem with the plant

- **REQ.6.1 Hose diameter.**
  The hose diameter must be defined in the configuration file.
- **REQ.6.2 Error hose diameter.**
  If the hose diameter isn't defined, an error message must be returned to the user signalling that the value is undefined.
- **REQ.6.3 Compute water volume.**
  The volume of water in cL shall be the difference between the threshold and the current soil humidity divided by 5 and rounded to the upper integer. The volume of the hose, which is pipe length times pipe diameter squared time Pi over 10, is then added to the computed volume.
- **REQ.7.1 Timer** The timer function using the internal clock of the card must work and be accurate.
- **REQ.8.1 Light sensor.**
  If the active mode is the punctual mode, no lighting value should be recorded.
- **REQ.8.2 Lighting value record.**
  The lighting values must be recorded once every hour in continuous, economic and planned mode.
- **REQ.8.3 Compute lighting time.**
  The lighting time of the plant over the last 24h must be computed.
- **REQ.8.4 Time of exposure.**
  The minimum time of exposure to sunlight for the plant must be defined in the configuration file.
- **REQ.8.5 Error lighting.**
  If the minimum time of exposure to sunlight for the plant is not defined, an error message must be returned to the user signalling that the value is undefined.
- **REQ.8.6 Information lighting.**
  If the exposure time over the last 24h is inferior to the to minimum time defined, an information message must be returned to the user signalling that the plant isn't exposed enough to the sunlight.

- **REQ.9.1 Continuous mode.**
  The plant is checked once every hour for watering.
- **REQ.10.1 Economic mode.**
  The plant is checked once every hour be should only be watered when the lighting value is under 10%.
- **REQ.11.1 Planned mode.**
  The plant is checked once every hour be should only be watered when the timer is in the period defined and at the start and end of the period.
- **REQ.11.2 Date and hour.**
  If the system is in planned mode, the date and hour of the day must be defined in the the configuration file.
- **REQ.11.3 Error date and hour.**
  If the date and hour is not defined, an error message must be returned to the user signalling that the values are undefined.
- **REQ.11.4 Activity period.**
  If the system is in planned mode, the activity period of the system must be defined in the the configuration file.
- **REQ.11.5 Error activity period.**
  If the activity period is not defined, an error message must be returned to the user signalling that the values are undefined.
- **REQ.12.1 Punctual mode.**
  The plant is checked only once and watered if needed.
- **REQ.12.2 Information done.**
  An information message must be returned to the user when the system has finished its tasks signalling that it can be shut down.

2.4 Traceability



HLR	LLR
REQ.1 Capacity	REQ.1.1 Container capacity REQ.1.2 Error container REQ.1.3 Information container
REQ.2 Humidity detection	REQ.2.1 Error humidity detection REQ.2.2 Humidity record
REQ.3 Watering threshold	REQ.3.1 Watering threshold REQ.3.2 Error threshold
REQ.4 Flooding	REQ.4.1 Information flooding
REQ.5 Abnormal variation	REQ.5.1 Information variation
REQ.6 Compute water volume	REQ.6.1 Hose diameter REQ.6.2 Error hose diameter REQ.6.3 Compute water volume
REQ.7 Time	REQ.7.1 Timer
REQ.8 Light	REQ.8.1 Light sensor REQ.8.2 Lighting value record REQ.8.3 Compute lighting time REQ.8.4 Time of exposure REQ.8.5 Error lighting REQ.8.6 Information lighting
REQ.9 Continuous mode	REQ.9.1 Continuous mode
REQ.10 Economic mode	REQ.10.1 Economic mode
REQ.11 Planned mode	REQ.11.1 Planned mode REQ.11.2 Date and hour REQ.11.3 Error date and hour REQ.11.4 Activity period REQ.11.5 Error activity period
REQ.12 Punctual mode	REQ.12.1 Punctual mode REQ.12.2 Information done

Figure 3: Traceability

3 Verification process

3.1 Test cases

3.1.1 HLR-test cases

- **TC.1.1 Error capacity.**
  Verify that the error message is shown on the screen when the capacity value is undefined.
- **TC.1.2 Information refill.**
  Verify that the information message is shown on the screen when the container has been emptied.
- **TC.2.1 Dry environment.**
  Verify that the humidity percentage is under 5% when the soil sensor is in a dry environment. (Air humidity)
- **TC.2.2 In water.**
  Verify that the humidity percentage is 100% when the soil sensor is in water.
- **TC.2.3 Error humidity.**
  Verify that the error message is shown on the screen when the humidity doesn't change after the pump was activated.
- **TC.3.1 Error threshold.**
  Verify that the error message is shown on the screen when the threshold value is undefined.
- **TC.4.1 Information flooding.**
  Verify that the information message is shown on the screen when the soil humidity is 20% higher than the threshold.
- **TC.5.1 Information no variation.**
  Verify that the information message is shown on the screen when the soil humidity doesn't change over 1 hour.
- **TC.5.2 Information high variation.**
  Verify that the information message is shown on the screen when the soil humidity drops more than 20% over 1 hour.

- **TC.6.1 Water volume.**
  Verify that the output water volume of the pump is the same as what was computed.
- **TC.6.2 Error hose diameter.**
  Verify that the error message is shown on the screen when the hose diameter value is undefined.
- **TC.7.1 Accuracy.**
  Verify that there is no time shift between the actual time and the internal clock of the system after the system ran for 48h.
- **TC.8.1 Night.**
  Verify that the light percentage detected is 0% during the night at midnight.
- **TC.8.2 Day.**
  Verify that the light percentage detected is 100% during the day at noon.
- **TC.9.1 Continuous mode**
  Verify that the humidity and light are recorded every hour and that the pump activates if needed at the same time.
- **TC.10.1 Economic mode**
  Verify that the humidity and light are recorded every hour and that the pump activates if needed at the same time only if the light value is under 10%.
- **TC.11.1 Error date and hour.**
  Verify that the error message is shown on the screen when the date and hour values are undefined.
- **TC.11.2 Error activity period.**
  Verify that the error message is shown on the screen when the activity period value is undefined.
- **TC.11.3 Planned mode**
  Verify that the humidity and light are recorded every hour and that the pump activates if needed at the same time only during the defined period and at the start and end of it.

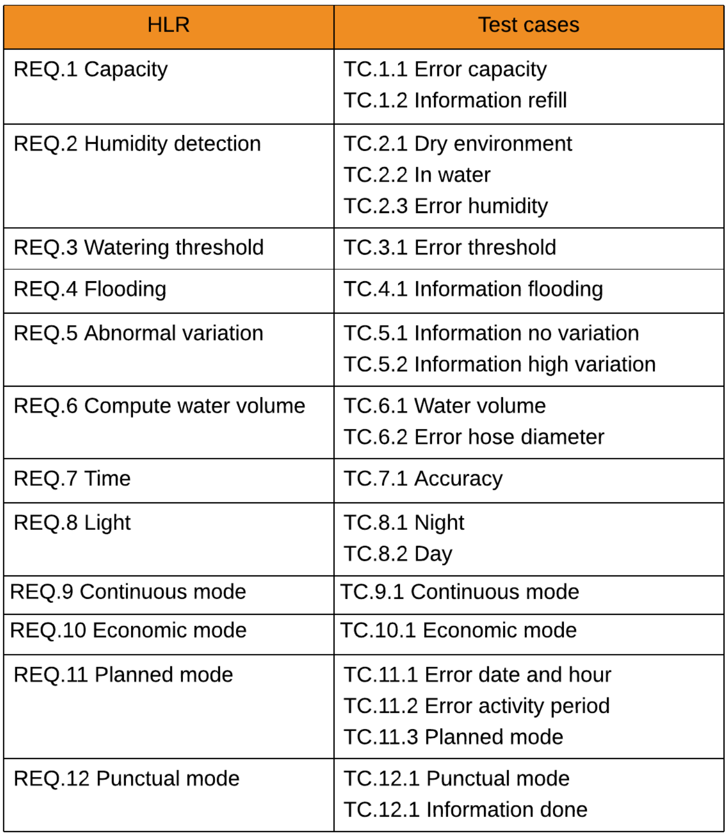
- **TC.12.1 Punctual mode**
  Verify that the pump activates only once if needed when the system is started.
- **TC.12.2 Information done.**
  Verify that the information message is shown on the screen when the system is done.

3.1.2 LLR-test cases

- **TC.1.1.1 ContainerAvailable.**
  Verify that Tank $>$ 10 when starting.
- **TC.1.3.1 Refill**
  Verify that the volume of water computed is added to the Watered variable after each watering.
- **TC.2.2.1 HumidityRecord.**
  Verify that the humidity values are recorded in the H_Vector vector.
- **TC.3.1.1 ThresholdAvailable.**
  Verify that Plant_Pot.Threshold $>$ 0 when starting.
- **TC.6.1.1 DiameterAvailable.**
  Verify that Pipe.D $>$ 0 when starting.
- **TC.6.3.1 ComputeWater.**
  Verify that the volume returned is 0 when there is no humidity change.
- **TC.8.1.1 PunctualLight.**
  Verify that no light values are recorded when the system is in punctual mode.
- **TC.8.2.1 LightRecord**
  Verify that the light values are recorded in the L_Vector vector.
- **TC.8.3.1 TimeExposure**
  Verify that the time exposure to light is added to the Light_Time variable after each record.
- **TC.8.4.1 MinimumTimeAvailable**
  Verify that Light_Time_Min $>=$ 0 when starting.

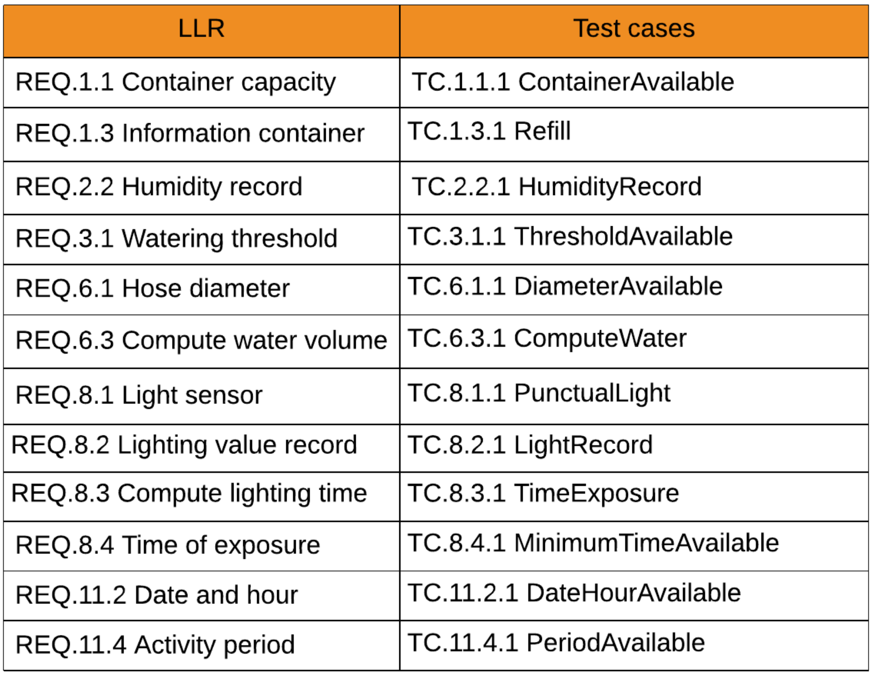
- **TC.11.2.1 DateHourAvailable**
  Verify that Date_Hour $>$ 0 when starting in planned mode.
- **TC.11.4.1 PeriodAvailable**
  Verify that Schedule.Start $>=$ 0 and Schedule.End $>=$ 0 when starting in planned mode.

3.2 Traceability



HLR	Test cases
REQ.1 Capacity	TC.1.1 Error capacity TC.1.2 Information refill
REQ.2 Humidity detection	TC.2.1 Dry environment TC.2.2 In water TC.2.3 Error humidity
REQ.3 Watering threshold	TC.3.1 Error threshold
REQ.4 Flooding	TC.4.1 Information flooding
REQ.5 Abnormal variation	TC.5.1 Information no variation TC.5.2 Information high variation
REQ.6 Compute water volume	TC.6.1 Water volume TC.6.2 Error hose diameter
REQ.7 Time	TC.7.1 Accuracy
REQ.8 Light	TC.8.1 Night TC.8.2 Day
REQ.9 Continuous mode	TC.9.1 Continuous mode
REQ.10 Economic mode	TC.10.1 Economic mode
REQ.11 Planned mode	TC.11.1 Error date and hour TC.11.2 Error activity period TC.11.3 Planned mode
REQ.12 Punctual mode	TC.12.1 Punctual mode TC.12.1 Information done

Figure 4: High level traceability



LLR	Test cases
REQ.1.1 Container capacity	TC.1.1.1 ContainerAvailable
REQ.1.3 Information container	TC.1.3.1 Refill
REQ.2.2 Humidity record	TC.2.2.1 HumidityRecord
REQ.3.1 Watering threshold	TC.3.1.1 ThresholdAvailable
REQ.6.1 Hose diameter	TC.6.1.1 DiameterAvailable
REQ.6.3 Compute water volume	TC.6.3.1 ComputeWater
REQ.8.1 Light sensor	TC.8.1.1 PunctualLight
REQ.8.2 Lighting value record	TC.8.2.1 LightRecord
REQ.8.3 Compute lighting time	TC.8.3.1 TimeExposure
REQ.8.4 Time of exposure	TC.8.4.1 MinimumTimeAvailable
REQ.11.2 Date and hour	TC.11.2.1 DateHourAvailable
REQ.11.4 Activity period	TC.11.4.1 PeriodAvailable

Figure 5: Low level traceability

3.3 Programming by contract

File **checkers.ads**

```
with Types; use Types;
with Errors; use Errors;
with Sensors; use Sensors;
with Configs; use Configs;

package Checkers is

   Init_H : Boolean := False;
   Last_H : Humidity;
   Tank_Current : Volume := 100.0;

   procedure Humidity_Checker_Before;

   function Humidity_Checker_After return Boolean -- True = Error
     with Pre => Last_H <= 100 and Last_H >= 0 and H <= 100 and H >= 0;

   procedure Tank_Update (V : Volume)
           with Pre => Tank_Current <= Tank and V <= Tank
           and Tank_Current >= V,
          Post => Tank_Current'Old <= Tank_Current;

   function Tank_Empty return Boolean
     with Post => Tank_Empty'Result = (Tank_Current = Volume(0));

   procedure Light_Check (L : Brightness; Min : Brightness)
     with Pre => L <= 100 and L >= 0 and Min <= 100 and Min >= 0;

end Checkers;
```

File **computes.ads**

```
with Ada.Real_Time; use Ada.Real_Time;
with Types; use Types;

package Computes is

   function Compute (Pip : Cylinder; H : Humidity; P : Plant;
        Pum : Debit) return Time_Span
     with Post => Compute'Result >= Seconds(0);

   function Compute_Time (Pum : Debit; Water : Volume) return Time_Span
     with Post => Compute_Time'Result >= Seconds(0);

   function Compute_Water (H : Humidity; P : Plant) return Volume
     with Pre => H <= 100 and H >= 0;

   function Compute_Pipe (Pip : Cylinder) return Volume
     with Pre => Float(Pip.L) <= Float'Last and Float(Pip.L) >= 0.0
     and Float(Pip.D) <= Float'Last and Float(Pip.D) >= 0.0,
          Post => Compute_Pipe'Result >= 0.0;

end Computes;
```

File **errors.ads**

```
package Errors is

   type String_Access is access constant String;

   Message_1 : aliased constant String := "I am Groot.";
   Message_2 : aliased constant String := "No Humidity Increase.";
   Message_3 : aliased constant String := "Plants needs water.";
   Message_4 : aliased constant String := "Please more light.";

   Message : Array (1 .. 4) of String_Access :=
     (
      1 => Message_1'Access,
      2 => Message_2'Access,
      3 => Message_3'Access,
      4 => Message_4'Access
     );
   Index : Natural := 1;

   procedure Set_Message (I : Natural)
     with Pre => Index <= 3 and Index >= 1,
          Post => I = Index;

   function Get_Message return String_Access;

end Errors;
```

3.4 Results

The system is fully efficient in its different activity modes. The data from the soil sensor and light sensor are recorded every hour and a nice graph of these data over the last 24 hours is outputted to the user.

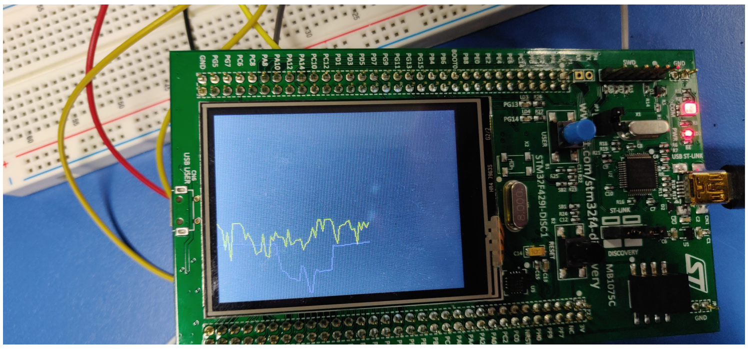
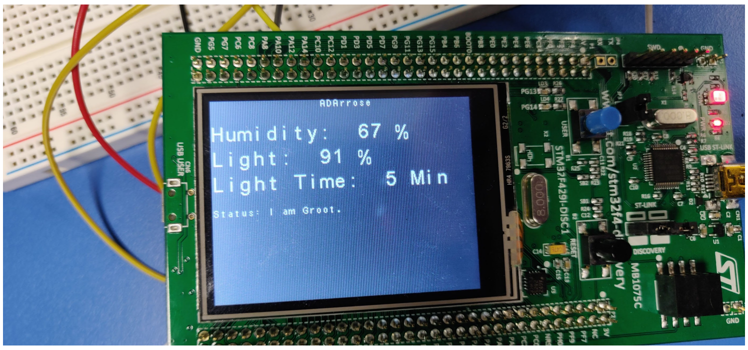


Figure 6: Light and humidity graph

The screen displays the current humidity and light percentage Touching the screen allows to switch between the graphical view and the current data view.



ADArrose

Humidity: 67 %

Light: 91 %

Light Time: 5 Min

Status: I am Groot.

Figure 7: Status with light and humidity percentage

Any error or information message is displayed at the bottom of the screen. This allows a permanent monitoring of the plant and any abnormal situation is reported.

When the humidity percentage isn't high enough, the pump is activated based on the system mode. The water volume is accurate. However, some of the water contained in the hose after the pump activation will drip into the plant pot. This additional water isn't enough to flood the pot and is negligible as it will increase the humidity of the soil of 1 percent maximum.

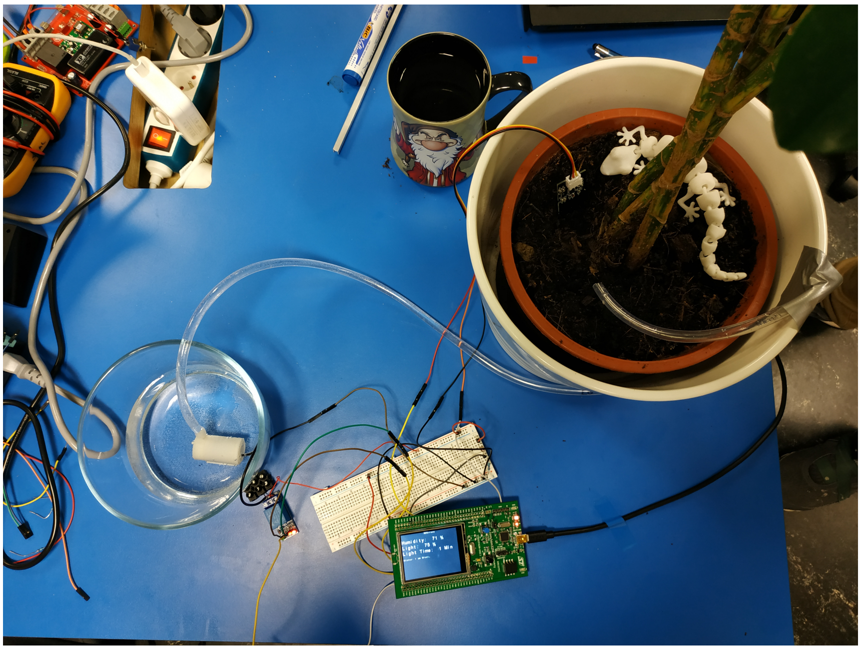
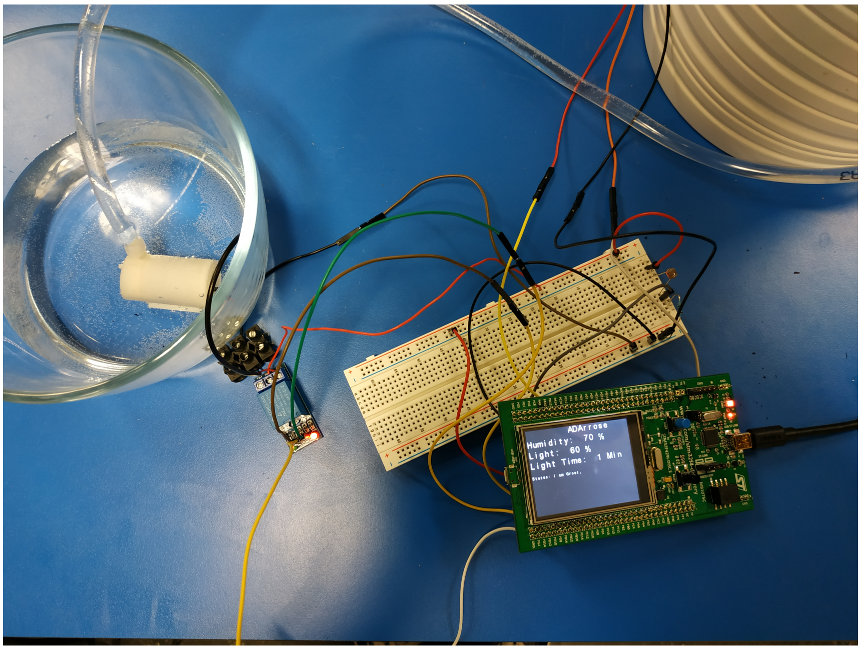


Figure 8: Full view of the system



```
ADArrose
Humidity:  70 %
Light:  60 %
Light Time:  1 Min
Status: I am Groot.
```

Figure 9: Pump and board

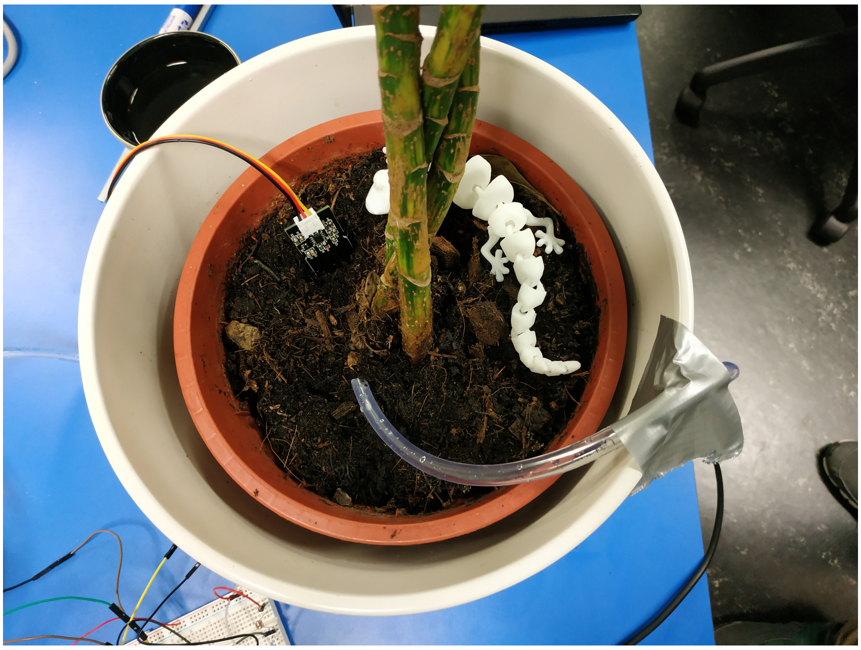


Figure 10: Pot with soil sensor and hose

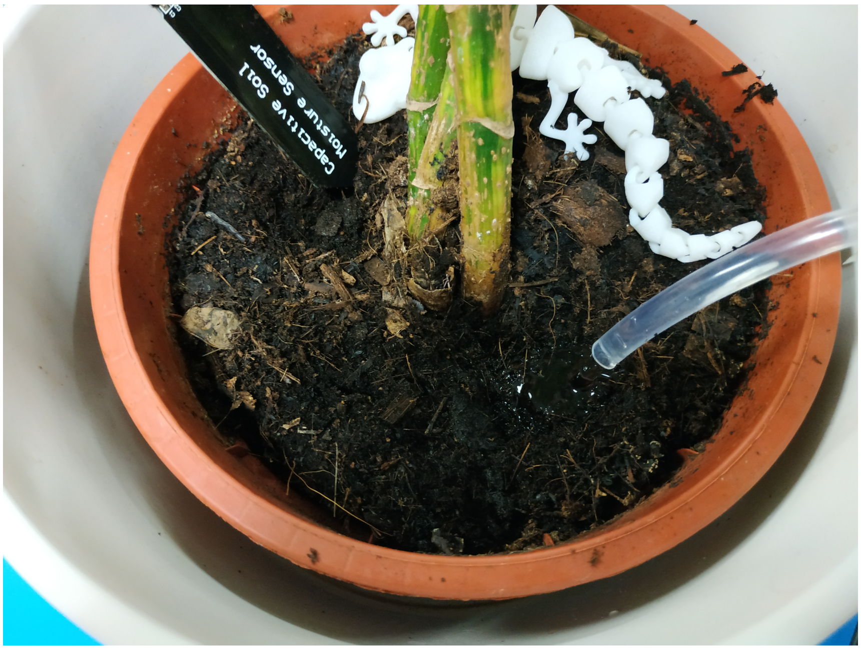


Figure 11: Pump activating

3.5 Future improvements

A lot of improvements can be implemented to upgrade the system. Being able to change the configuration parameter from the screen instead of having to change them in the configuration file is a big improvement. Moreover, by doing this, the system doesn't have to be restarted.

A button could also be added to signal when the water tank has been refilled. Like that the system doesn't have to be restarted after each refill.

The humidity and light needs of the different kinds of plant could also be listed in the system. Like that the user only has to select his plant to set up the system.

Other sensors could be added to give more information about the plant to the user. A heat sensor, for example would be interesting to upgrade the economic mode and provide a better experience.