Digital Twin for Plant Health Monitoring

Rayan Contuliano Bravo

Université Libre de Bruxelles

06/03/2025

Motivation and Problem Statement

Problem Statement: Given a physical plant, develop a digital twin to monitor health, predict needs, and provide recommendations and actions.

- Improve decision making and prediction
- Constant monitoring
- Reduce costs
- Reduce human error

Formal Problem Description

Given a physical plant P evolving in a continuously changing environment E. The goal is to develop a digital twin D that is a dynamic, real-time virtual replica of the environmental context and state of the plant.[1]

$$E(t) = \{L(t), M(t), T(t), H(t)\}$$

- $L(t) \in [0, 65000]$: Light intensity measured in lux at time t
- $M(t) \in [0, 100]$: Soil Moisture measured in percentage at time t
- $T(t) \in [-40, 80]$: Temperature measured in degrees Celsius at time t
- $H(t) \in [0, 100]$: Humidity measured in percentage at time t

[1] This is a simplification, in practice, the environment can be characterized by many more variables.

Formal Problem Description (2)

The plant P can be characterized by its internal state $I(t)$, which evolves with t, which can be described by a set of internal variables.

$$I(t) = \{G(t), C(t)\}$$

- $G(t) \in [0, 100]$: Growth stage of the plant measured in percentage at time t
- $C(t) \in [0, 255]$: Global coloring of the plant measured in RGB values at time t

Formal Problem Description (3)

The digital twin D not only let us visualize the state of the plant and the factors influencing it, but also predict its future state and provide recommendations and actions to optimize its health.

Let $S(t)$ be the state of the system at time t as a function of the environment E and the plant's internal state I.

$$S(t) : E(t) \times I(t) \rightarrow \mathcal{S}$$

where $\mathcal{S}$ is the state space

System Design and Implementation

The digital twin system is composed of the following sensors :

- Temperature & Humidity Sensor DHT22
- Light Intensity Sensor Adafruit BH1750
- Soil Moisture Sensor Adafruit STEMMA

DHT22 [2]

- More accurate than DHT11
- Measures temperature from $-40°C$ to $80°C$ ($\pm 0.5°C$ error)
- Measures humidity from 0% to 100% ($\pm 2\%$ error)

This sensor will allow us to maintain the optimal temperature and humidity in the plant environment to keep it healthy.

Uses the GPIO commmunication protocol

[2] https://randomnerdtutorials.com/raspberry-pi-dht11-dht22-python/

BH1750 [3]

- Measures light intensity in LUX from 0 to $65k+$ lux

This sensor will allow us to keep an optimal light intensity for the plant. (photosynthesis)

Uses the I2C communication protocol

[3] https://learn.adafruit.com/adafruit-bh1750-ambient-light-sensor/python-circuitpython

STEMMA [4]

- Capacitive because resistive sensors will oxidize over time
- Give results from 200 (very dry) to 2000 (very wet)
- Gives the temperature ss well (+- $2C$ error)

Uses the I2C communication protocol

All the sensors can be used with their respective **adafruit** libraries in Python which make it easier to integrate into the system

[4] https://learn.adafruit.com/adafruit-stemma-soil-sensor-i2c-capacitive-moisture-sensor/overview

MQTT

Sensor data is sent to the Digital Twin system, which **processes it** and sends it to the MQTT broker. The MQTT broker manages the incoming messages and disseminates them to the interested subscribers (e.g., dashboard, database) leading to decoupling of components.

- **Decoupling of components**: Any subscriber can subscribe to data streams without needing direct connections to sensors or the database. If database goes offline the dashboard can still work.
- **Real-time Communication**: Pushing messages directly to subscribers as soon as they're available. **No periodic polling**.
- **Improved Scalability**: New features (like a second dashboard, smartphone notification app, cloud analytics, etc.) does not require changing the original publishers or redesigning the system.

Example sensor: Soil Moisture

- Very controllable sensor which is perfect for implementation and testing
- Make sure that the data flow is working correctly.
- Other sensors should not be too hard to integrate

Usage of mock data while waiting for the sensor to arrive

Status and Challenges

Status:

- First reading of the documentation
- Started the implementation of the system architecture
- Trying to mock the data while waiting for the sensor to arrive

Challenge:

- Integrating MQTT in the system

Future Step:

- Integrate the soil moisture sensor in the system
- Check the data flow from the sensor to the dashboard