

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA TP.HCM**

ĐỒ ÁN CUỐI KỲ

Môn: Lập trình IoT



Lương Tấn Khang – 1612865

Phan Quốc Thắng – 1612891

Trương Văn Hoàng - 1612857

1. Giới thiệu ứng dụng

Hệ thống được thiết kế theo mô hình “**Plug and Play**” đơn giản, tức là người dùng không bị gò bó số lượng thiết bị IoT đặt dưới local.

Người dùng chỉ cần mua thiết bị về, connect đến đúng Wifi và dịch vụ MQTT, sau đó sử dụng Mobile App để quản lý từ xa.

Cách thức sử dụng Mobile App

a. Các màn hình

i. Loading và Home

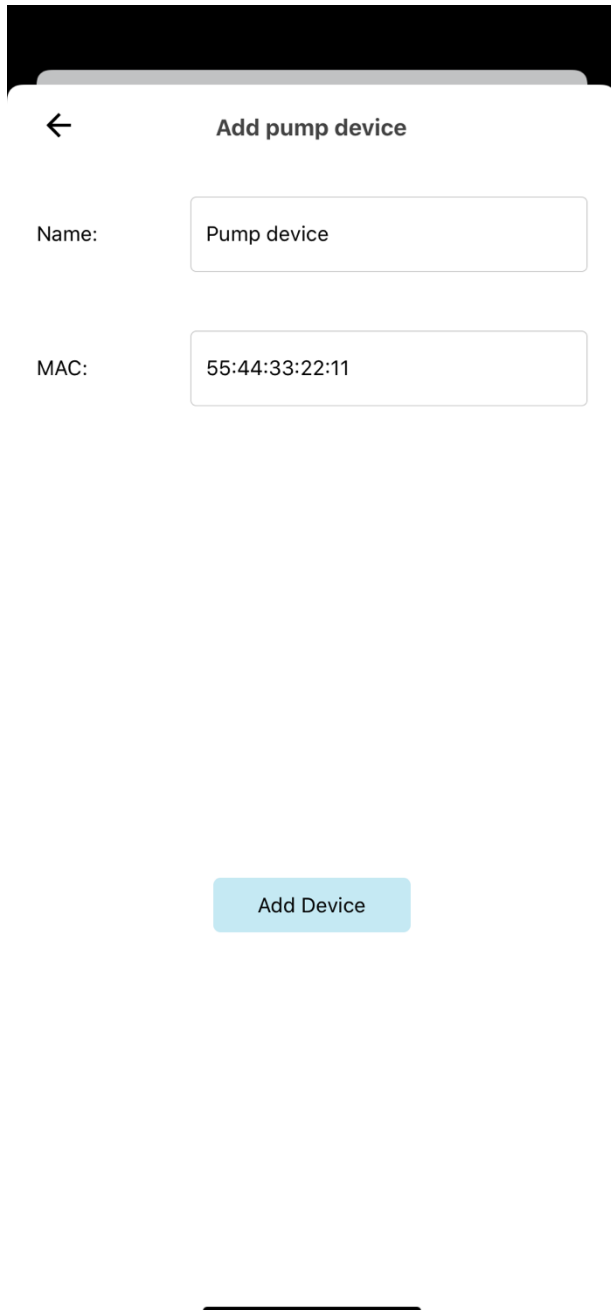


Có một nút bấm để kết nối ứng dụng với Server để đảm bảo chắc chắn việc kết nối đã được thiết lập. Sau khi đã thiết lập kết nối đến Server thì ứng dụng sẽ cho phép người dùng đi vào 2 nhánh chính đó là các thiết bị đo đặc số liệu nhiệt độ, độ ẩm đất, độ ẩm không khí và nhánh các thiết bị bơm nước

ii. List Measure và List Pump



Hai màn hình này thể hiện danh sách các thiết bị đã được kết nối. Người dùng có thể kết nối thiết bị thông qua địa chỉ MAC và loại thiết bị. Ngoài ra ứng dụng cho phép người dùng đặt tên thiết bị một cách tùy ý để dễ dàng cho việc quản lý cũng như theo dõi của người dùng.

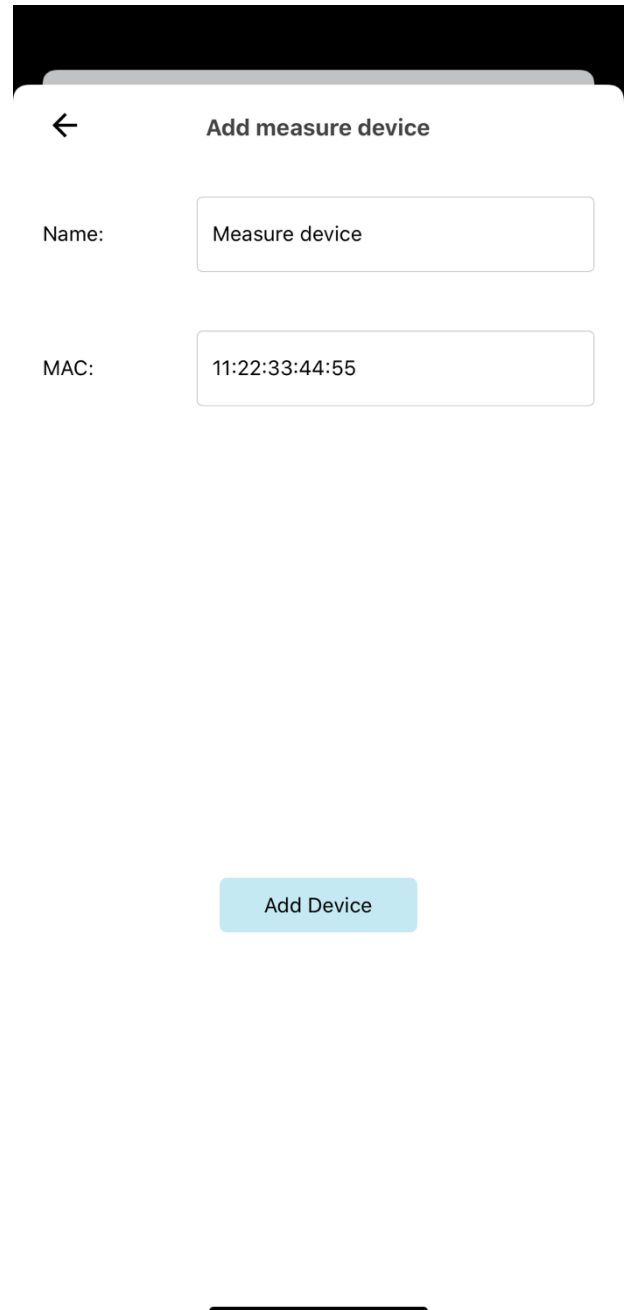


← Add pump device

Name: Pump device

MAC: 55:44:33:22:11

Add Device



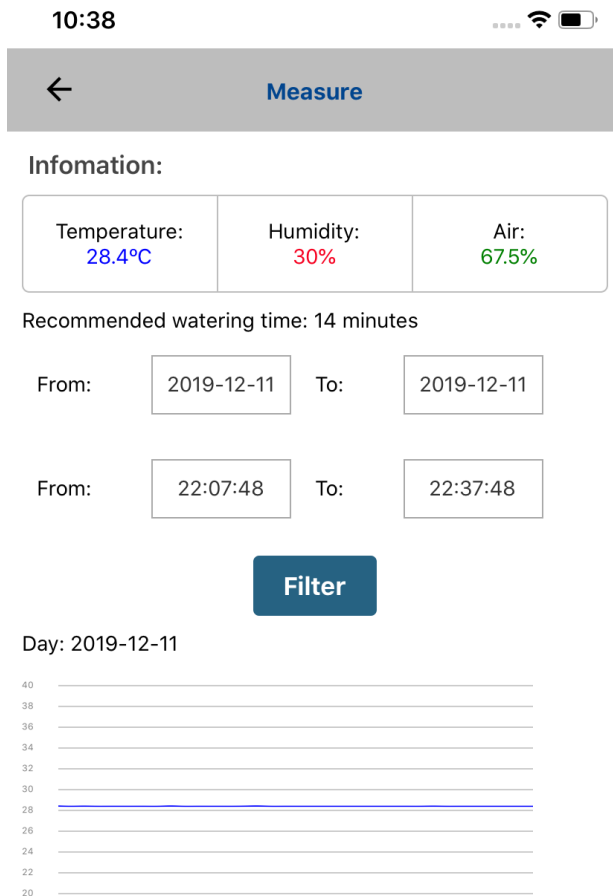
← Add measure device

Name: Measure device

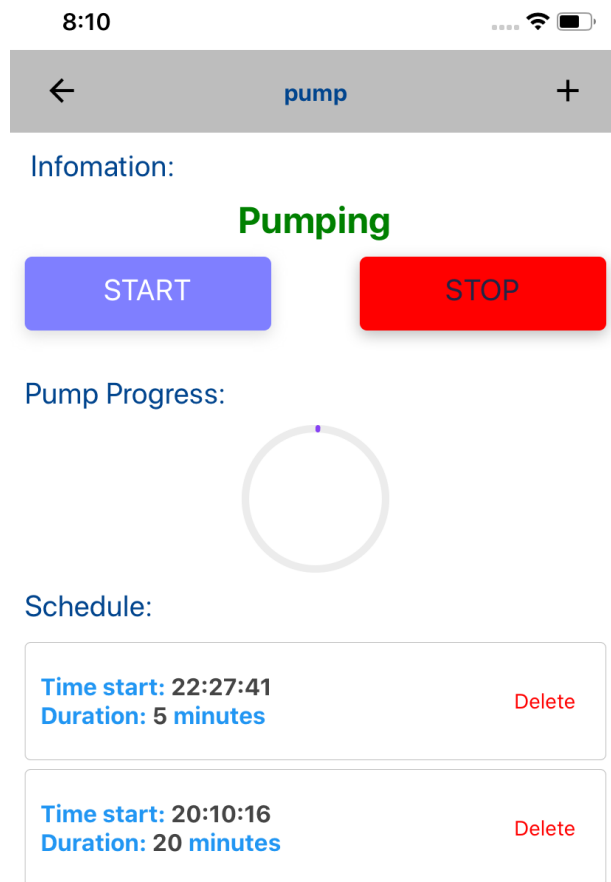
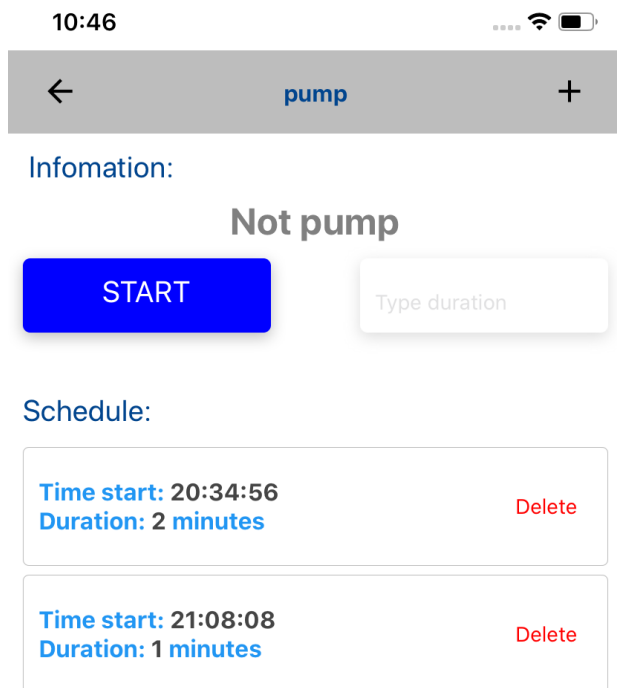
MAC: 11:22:33:44:55

Add Device

iii. Measure Detail và Pump Detail



Ở màn hình chi tiết về thiết bị đo, ứng dụng sẽ lấy dữ liệu về nhiệt độ, độ ẩm đất và độ ẩm không khí được lưu lại gần nhất thể hiện trên khu vực thông tin. Ngoài ra, thuật toán Fuzzy Logic cũng đưa ra cho chúng ta thời gian tưới nước dự kiến để người dùng có thể tham khảo. Người dùng có thể xem lại lịch sử dữ liệu đo được lưu trữ lại theo ngày và giờ, ở ví dụ bên hình thì dữ liệu vẽ trên biểu đồ đường sẽ là những số liệu đo được trong khoảng từ 22:07:48 tới 22:37:48 trong ngày 11/12/2019. Người dùng có thể tùy ý thay đổi ngày và giờ để hệ thống biểu diễn lên biểu đồ.



Ở màn hình chi tiết về máy bơm thì người dùng sẽ thấy được trạng thái của máy bơm là đang bật hay tắt, nếu thiết bị đang bật thì sẽ có một biểu đồ thể hiện mức độ hoàn thành của máy bơm (như hình bên cạnh).

Khi thiết bị đang bơm, người dùng có thể dừng việc bơm lại bằng cách nhấn vào nút **STOP** trên màn hình.

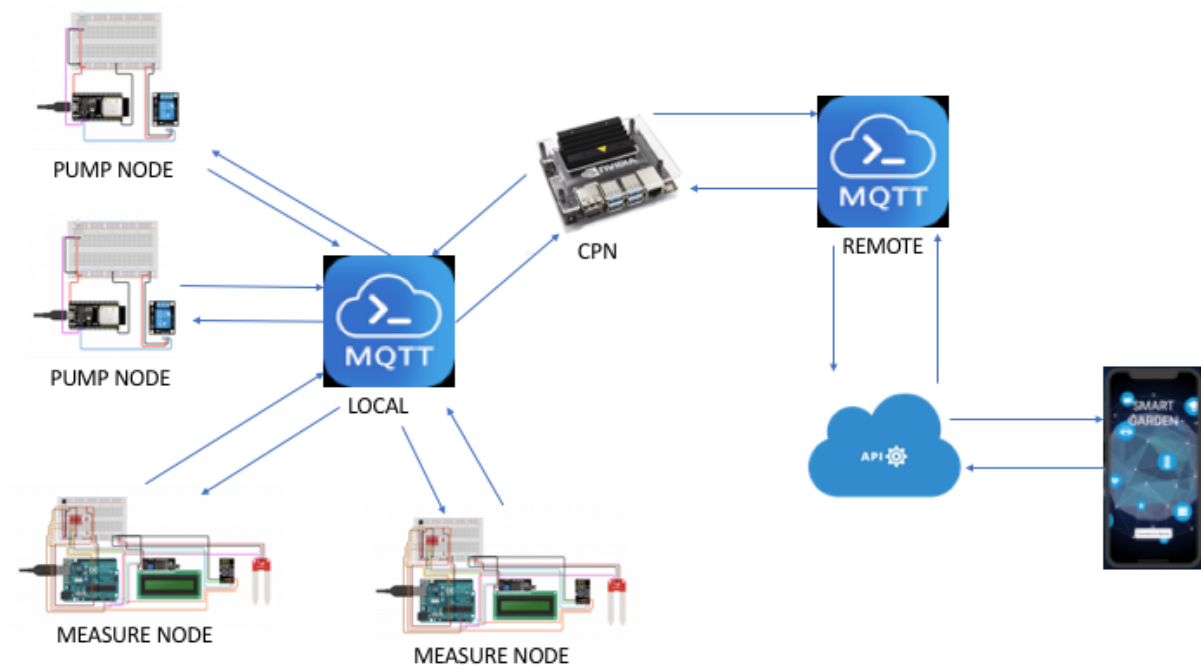
Ngoài ra, khi thiết bị đang ở trạng thái không bơm, người dùng có thể khởi động máy bơm theo thời gian được điền vào sau đó nhấn nút **START** trên màn hình.

Ứng dụng cho phép người dùng đặt lịch bơm theo từng ngày với thời gian bắt đầu bơm và thời lượng bơm tính theo phút. Ngoài ra người dùng có thể xoá lịch trình bơm nếu cần thiết.

The image shows a mobile application interface. At the top, there is a dark header with a back arrow and the text "Add schedule". Below this, the form has two input fields: "Choose time start:" with a value of "20:10:16" and "Duration: (minute)" with a value of "20". At the bottom of the form is a button labeled "Add Schedule". Below the form is a modal with two buttons: "Cancel" and "Confirm". At the very bottom, there is a time picker showing hours from 5 to 11 and minutes from 07 to 13, with "AM" and "PM" indicators. The time "8:10 PM" is currently selected.

Hour	Minute	Period
5	07	
6	08	
7	09	AM
8	10	PM
9	11	
10	12	
11	13	

2. Mô tả cách hoạt động của hệ thống



Quy trình hoạt động

- Hệ thống gồm:
 - Mobile App
 - Web API
 - Dịch vụ MQTT Remote
 - Database Postgres Remote
 - Central Processing Node
 - Dịch vụ MQTT Local
 - Measure Node
 - Pump Node
 - Databáe Postgres Local
- Cách hoạt động:
 - Mobile App sẽ cung cấp giao diện quản lý người dùng và chỉ kết nối đến Web API.

- Giữa Cloud và Local sẽ kết nối với nhau thông qua 1 dịch vụ MQTT trên internet (MQTT Remote)
- Ở local, chỉ có CPN được kết nối với internet và hoạt động như gateway, giao tiếp với web api thông qua MQTT remote.
- Các node ở local sẽ được kết nối với nhau thông qua 1 dịch vụ MQTT Local. Mỗi một Pump Node hay Measure Node sẽ được phân biệt qua MAC Address của module ESP8266.
- Ngoài ra, local side và remote side sẽ sử dụng 2 database Postgres riêng biệt nhau.
- Khi có 1 lệnh từ Mobile App gửi lên web api, web api sẽ xử lý và chuyển thành lệnh tương ứng đưa lên MQTT Remote, sau đó CPN sẽ nhận được lệnh này, xử lý và chuyển lên MQTT Local (nếu cần) để gửi đến các node tương ứng.

3. Chi tiết từng thành phần

a. Database

- Database Local

Database gồm có 2 bảng là Measures và Schedules.

Cấu trúc bảng Measure:

Field	Type
Mac	String
Time	String
Date	String
Temperature	Float
Humidity	Float
Air	Float
IsDeleted	Int

Cấu trúc bảng Schedules:

Field	Type
-------	------

Mac	String
Time	String
Duration	Int

- Database Remote

Database gồm 3 bảng Measures, Schedules và Devices

Cấu trúc bảng Measures tương tự như trên nhưng không có trường IsDeleted

Cấu trúc bảng Schedules tương tự như trên

Cấu trúc bảng Devices

Field	Type
Name	String
Mac	String
Type	String
Deleted	Int

b. Measure Node

- Node này sẽ bao gồm các module và cảm biến:

- NodeMCU Esp8266
- 2 Led Tín hiệu
- DHT22 sensor
- Moisture Humidity sensor

- Cách thức hoạt động:

Khi bật lên Node sẽ connect đến Wifi và MQTT mặc định, nếu không thành công thì đèn đỏ sẽ sáng ngược lại đèn xanh sẽ sáng.

Node sẽ đo các thông số môi trường gồm độ ẩm đất, độ ẩm không khí và nhiệt độ không khí mỗi giây. Sau 60 giây, sẽ tổng hợp và dùng phương thức chia trung bình để có được giá trị cuối cùng. Tiếp theo, Node sẽ publish lên MQTT Local, topic “/measure” với cấu trúc:

MAC Address|Temperature|Moisture Humidity|Air Humidity|

c. Pump Node

- Node này bao gồm module:

- NodeMCU ESP8266
- 2 LED tín hiệu
- Relay

- Cách thức hoạt động:

Khi bật lên Node sẽ connect đến Wifi và MQTT mặc định, nếu không thành công thì đèn đỏ sẽ sáng ngược lại đèn xanh sẽ sáng.

Node sẽ subscribe vào topic “/pump” ở MQTT local. Khi nhận được message từ topic này, Node sẽ xét xem có phải message đó được gửi tới mình không, sau đó sẽ tiến hành xử lý và thực hiện Đóng hoặc Ngắt relay. Message sẽ có dạng

MAC|{0,1}|Duration|

0 sẽ tương ứng với ngắt relay (Stop pump) và 1 sẽ tương ứng với kích relay (Start Pump).

Duration có đơn vị là phút.

d. Central Processing Node

CPN là một máy tính Jetson Nano, bao gồm 3 thành phần Measure Controller (MC), Pump Controller (PC) và CPN.

- Measure Controller

MC sẽ subscribe vào topic “/measure” ở MQTT Local, mỗi lần nhận được data từ Measure Node, MC sẽ cập nhật data này vào local DB và đánh dấu là data chưa được gửi.

- Pump Controller

Cứ mỗi 1 phút, PC sẽ quét bảng Schedule ở DB local, sau đó sẽ tiến hành publish vào topic “/pump” ở MQTT Local để ra lệnh cho Pump Node.

- CPN

- Cứ mỗi 10 phút, CPN sẽ quét các data trong bảng Measure ở DB local, sau đó publish vào topic “/data” trên MQTT remote và gửi hết những data được đánh dấu “IsDeleted = 0”, tiếp theo đánh dấu các data này là “IsDeleted = 1”
- CPN subscribe vào topic “/command” ở MQTT Remote.

Có 2 loại command: Pump và Schedule

- Command Pump sẽ có message đầy đủ để gửi đến Pump node. Khi nhận được message này, CPN sẽ publish vào topic “/pump” ở MQTT local và gửi message này lên đó cho Pump Node.
- Command Schedule sẽ có message chứa danh sách schedule của 1 Pump Node nào đó. Khi nhận được message này, CPN sẽ xoá hết schedule hiện tại của Node đó và cập nhật schedule mới.

e. *Web API và dịch vụ fuzzy logic*

○ Cách giao tiếp với CPN

WebAPI được viết bằng Golang.

WebAPI sẽ giao tiếp với hệ thống local qua MQTT Remote.

Ở đây, WebAPI làm 2 việc:

- Thứ nhất, publish vào topic “/command” để gửi lệnh Pump và Schedule xuống local.
 - Thứ hai, subscribe vào topic “/data” để nhận dữ liệu measure từ local.
- Fuzzy logic được viết bằng Python và có chức năng hỗ trợ tính toán thời gian cần bơm nước với data hiện tại cho người dùng. Input sẽ là nhiệt độ và độ ẩm đất, output sẽ là thời gian tưới đơn vị phút.

○ Bảng API

Tất cả các request phải đi kèm 1 token gắn ở header.

Method	API	Description
--------	-----	-------------

GET	/device?type=	Lấy danh sách các device dựa theo type
POST	/device?name=&mac=&type=	Tạo 1 device mới
DELETE	/device?mac=&type=	Xoá device
GET	/schedule?mac=	Lấy danh sách schedule của 1 Node dựa trên MAC address
POST	/schedule?mac=	Update toàn bộ schedule mới cho 1 Node
GET	/data?mac=&date_from=&date_to=&time_from=&time_to=	Lấy tất cả data của 1 Measure Node từ ngày date_from đến date_to trong khoảng thời gian time_from đến time_to
GET	/pump?mac=&type=&duration=	Gửi lệnh pump tới Node mac với duration đơn vị phút. Type thuộc {0,1}, 0 là dừng, 1 là bắt đầu.
GET	/last_data?mac=	Lấy data cuối cùng của 1 Measure Mac
GET	Port 5000, /?temperature=&moisture=	Lấy thời gian tưới nước được gợi ý từ fuzzy logic

f. Mobile App

Mobile App được viết bằng React Native.

g. *Fuzzy Logic*

▪ Thiết lập luật mờ

Luật mờ được thiết lập trên kinh nghiệm chuyên gia (ví dụ trong lĩnh vực nông học) và sử dụng các quy tắc suy diễn như :

“Nếu nhiệt độ là **nóng** và độ ẩm đất ở mức **khô** thì thời gian tưới nước cần phải **dài**”

“Nếu nhiệt độ là **mát** và độ ẩm đất ở mức **ướt** thì thời gian tưới nước cần phải **ngắn**”

Luật mờ	IF Nhiệt độ, [độ C]	AND Độ ẩm đất, [%]	THEN Thời gian tưới, [phút]
1	Rất nóng	Khô	Dài
2	Rất nóng	Ẩm	Trung bình
3	Rất nóng	Ướt	Ngắn
4	Rất nóng	Rất ướt	Ngắn
5	Nóng	Khô	Dài
6	Nóng	Ẩm	Trung bình
7	Nóng	Ướt	Ngắn
8	Nóng	Rất ướt	Ngắn
9	Bình thường	Khô	Dài
10	Bình thường	Ẩm	Trung bình
11	Bình thường	Ướt	Ngắn
12	Bình thường	Rất ướt	Rất ngắn
13	Mát	Khô	Dài
14	Mát	Ẩm	Trung bình
15	Mát	Ướt	Ngắn
16	Mát	Rất ướt	Rất ngắn
17	Lạnh	Khô	Dài
18	Lạnh	Ẩm	Ngắn
19	Lạnh	Ướt	Ngắn
20	Lạnh	Rất ướt	Rất ngắn

▪ Cài đặt

Sử dụng thư viện scikit-fuzzy

Có 3 bước để xây dựng hệ thống điều khiển mờ trong Python. Trước tiên phải import numpy, skfuzzy module và skfuzzy.control:

```
import numpy as np
import skfuzzy as fuzz
import skfuzzy.control
```

Bước 1: Khởi tạo các giá trị input và output

```
def make_variables(self):
    """
    Create input, output variables
    :return:
    """
    self.__temperature = skfuzzy.control.Antecedent(np.arange(self.__temperature_start, self.__temperature_stop), 'Temperature')
    self.__moisture = skfuzzy.control.Antecedent(np.arange(self.__moisture_start, self.__moisture_stop), 'Moisture')
    self.__time = skfuzzy.control.Consequent(np.arange(self.__time_start, self.__time_stop), 'Time')
```

Các giá trị đầu vào gồm self.__temperature và self.__moisture, một giá trị ngõ ra là self.__time. Temperature (°C) có phạm vi từ 0 - 45 và Moisture (%) có phạm vi từ 0 – 100, time (minutes) có phạm vi từ 0 – 30. Sử dụng ‘self’ vì nhóm sử dụng lập trình hướng đối tượng để xây dựng hệ thống điều khiển mờ.

Bước 2: Khởi tạo các hàm liên thuộc

```
def make_member_functions(self):
    """
    Create member functions
    :return:
    """
    self.__temperature['cold'] = fuzz.trapmf(self.__temperature.universe, [self.__temperature_low_start,
                                                                              self.__temperature_low_mid1,
                                                                              self.__temperature_low_mid2,
                                                                              self.__temperature_low_stop])
    self.__temperature['normal'] = fuzz.trimf(self.__temperature.universe, [
                                                                              self.__temperature_mod_start,
                                                                              self.__temperature_mod_mid,
                                                                              self.__temperature_mod_stop])
    self.__temperature['hot'] = fuzz.trapmf(self.__temperature.universe, [self.__temperature_high_start,
                                                                              self.__temperature_high_mid1,
                                                                              self.__temperature_high_mid2,
                                                                              self.__temperature_high_stop])
    self.__moisture['dry'] = fuzz.trapmf(self.__moisture.universe, [self.__moisture_low_start,
                                                                    self.__moisture_low_mid1,
                                                                    self.__moisture_low_mid2,
                                                                    self.__moisture_low_stop])
    self.__moisture['normal'] = fuzz.trimf(self.__moisture.universe, [
                                                                    self.__moisture_mod_start,
                                                                    self.__moisture_mod_mid,
                                                                    self.__moisture_mod_stop])
    self.__moisture['wet'] = fuzz.trapmf(self.__moisture.universe, [self.__moisture_high_start,
                                                                    self.__moisture_high_mid1,
                                                                    self.__moisture_high_mid2,
                                                                    self.__moisture_high_stop])
```

```

self.__time['very short'] = fuzz.trapmf(self.__time.universe, [self.__time_short_start,
                                                                self.__time_short_mid1,
                                                                self.__time_short_mid2,
                                                                self.__time_short_stop])

self.__time['short'] = fuzz.trapmf(self.__time.universe, [self.__time_nor_start,
                                                           self.__time_nor_mid1,
                                                           self.__time_nor_mid2,
                                                           self.__time_nor_stop])

self.__time['long'] = fuzz.trapmf(self.__time.universe, [self.__time_long_start,
                                                          self.__time_long_mid1,
                                                          self.__time_long_mid2,
                                                          self.__time_long_stop])

```

Temperature có 3 hàm liên thuộc: cold, normal, hot

Moisture có 3 hàm liên thuộc: dry, normal, wet

Time có 3 hàm liên thuộc: very short, short, long

Bước 3: Tạo bảng luật mờ

```

def make_rules(self):
    """
    Create Fuzzy Rules
    :return:
    """
    rule1 = skfuzzy.control.Rule(self.__temperature['cold'] & self.__moisture['dry'], self.__time['long'])
    rule2 = skfuzzy.control.Rule(self.__temperature['cold'] & self.__moisture['normal'], self.__time['short'])
    rule3 = skfuzzy.control.Rule(self.__temperature['cold'] & self.__moisture['wet'], self.__time['very short'])
    rule4 = skfuzzy.control.Rule(self.__temperature['normal'] & self.__moisture['dry'], self.__time['long'])
    rule5 = skfuzzy.control.Rule(self.__temperature['normal'] & self.__moisture['normal'], self.__time['short'])
    rule6 = skfuzzy.control.Rule(self.__temperature['normal'] & self.__moisture['wet'], self.__time['very short'])
    rule7 = skfuzzy.control.Rule(self.__temperature['hot'] & self.__moisture['dry'], self.__time['long'])
    rule8 = skfuzzy.control.Rule(self.__temperature['hot'] & self.__moisture['normal'], self.__time['short'])
    rule9 = skfuzzy.control.Rule(self.__temperature['hot'] & self.__moisture['wet'], self.__time['very short'])

```

4. Source code

[1] <https://github.com/ltkhang/IoT>