User Manual

The user will need:

- A PC that has python version 3 installed (Mac/Linux/Windows) with Microsoft ODBC 17 Driver.
- Downloaded gui_code and src folders from our public GitHub repository: https://github.com/Willmish/comp0016-farmbeats
- A .env file containing the correct key for accessing the Azure database securely (This file is provided when setting up the database - refer to the deployment manual for more information).
- Internet connection.
- Raspberry Pi with a copied directory of the repository, installed dependencies, set up software with ssh, I2C, GPIO and PWM enabled and .env file containing unique device key for IoT Hub.

How to run the Raspberry Pi for data collection:

1. Change directories to the downloaded src folder on terminal.
2. Connect the device with the Raspberry Pi:
    0. Ensure the Raspberry Pi is connected to the internet with a microSD card containing a config file, allowing it to be connected to the internet.
    1. Type ssh pi@raspberrypi on your terminal. It should ask for a password - the default is 'raspberry', but this can be changed by the user.
    2. Once you are connected, create a folder called farmbeats and copy the current src folder from PC to Raspberry Pi.
    3. Once you are in the copied src directory of the Raspberry Pi, run python3 main.py to start the system for live data collecting and generating actuation.

How to run the GUI:

1. Change directories to the downloaded gui_code folder on the terminal.
2. Check that python 3 is installed on the PC by running:

    python3 --version

    If python 3 is installed, then the output would be Python 3.X.X where X could be any number.

3. Run the following command:

    python3 gui_main.py

    The default options page of the GUI should open.

If it does not open, there may be an issue with connecting to the database. Please ensure that the .env file is correct and that the Azure Database is set up properly by viewing it on your Azure portal account.

How to use the GUI:

1. To view a subsystem, click on a button showing the subsystem name. For example, if you want to view the current farm state for humidity, click on 'Humidity'.
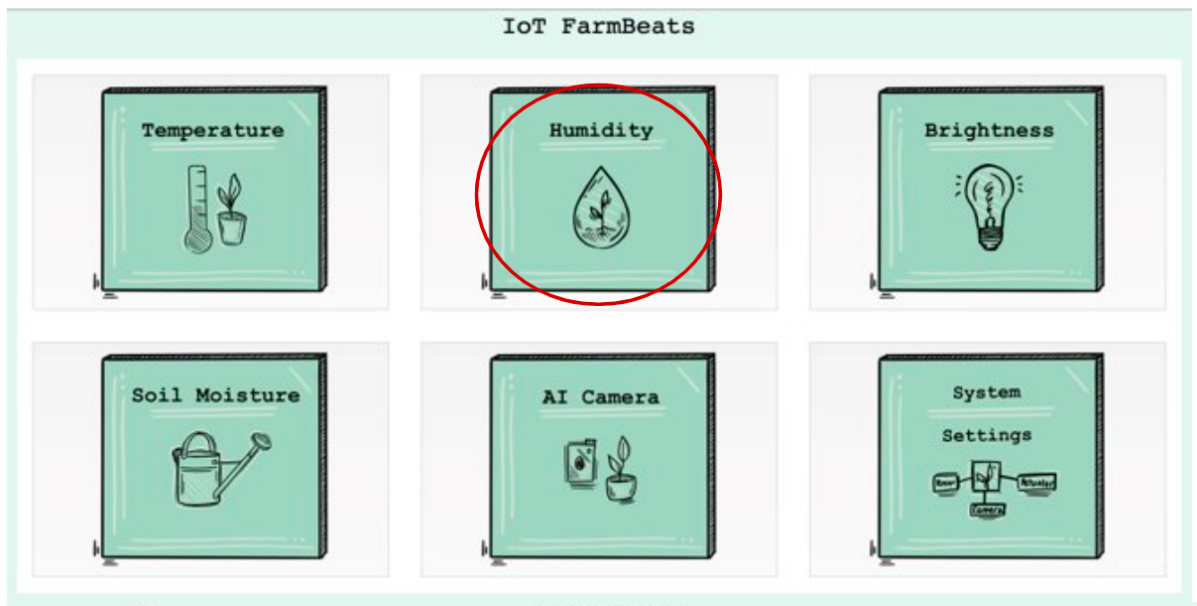
This page should open:
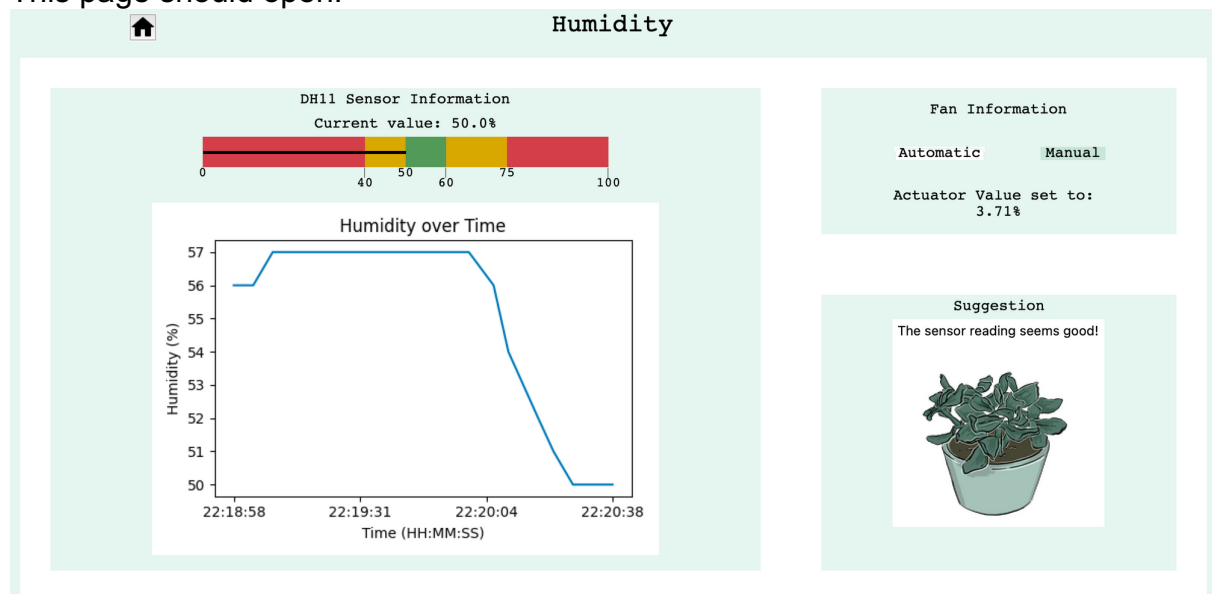
2. The user can now view the graphs, labels and scales for the subsystem. (Note, the manual mode has not been implemented, so the user cannot use the buttons).
3. To head back to the options page, click the home button.

4. To view the configuration file for the plant profile, press the 'System Settings' button on the options page, and this page should show up:
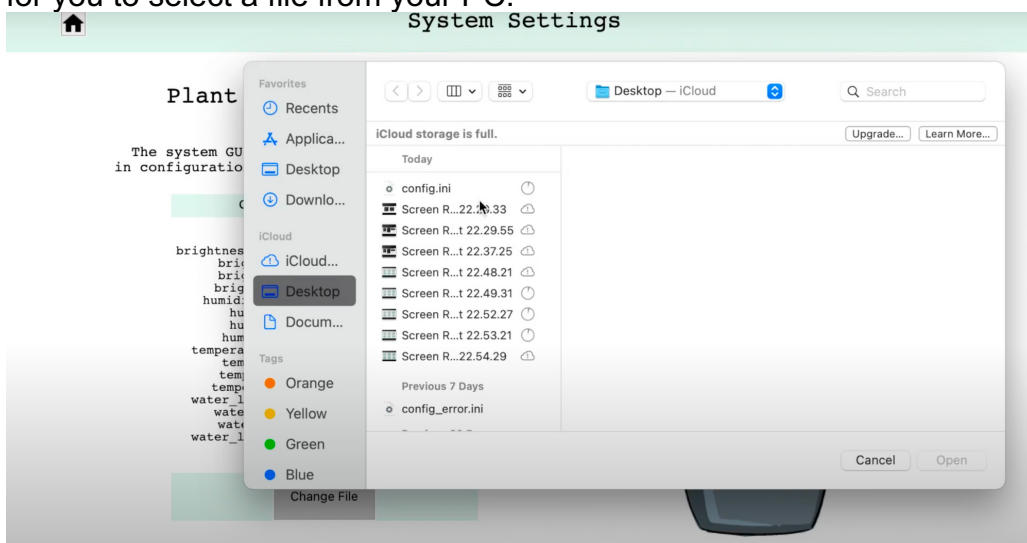

5. To change the file, click on 'Change File', and the following tab should open for you to select a file from your PC.

The file type must be .ini and the structure of the config file must be correct (see below), otherwise the GUI will not allow file change.
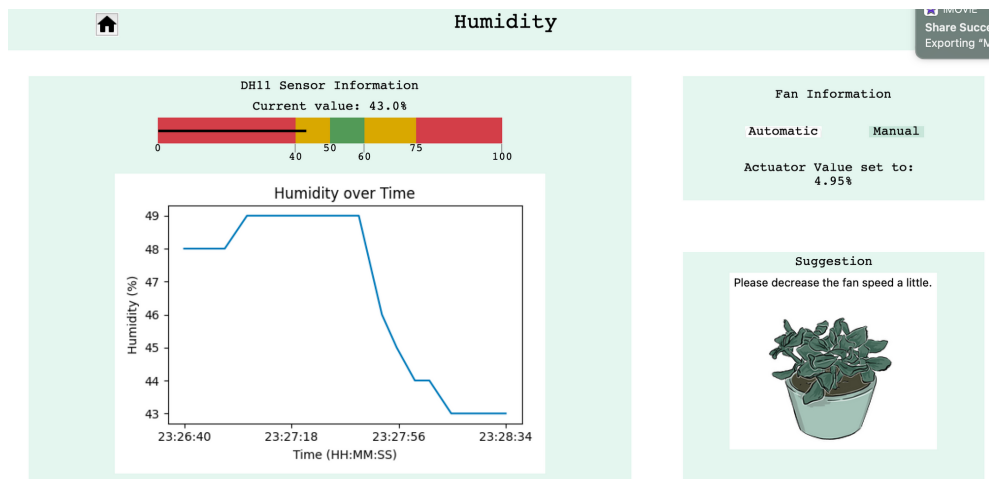

Explaining the config file:

The sensor scale of a subsystem profile page is based on a list of thresholds that allow the system to display whether or not the current condition is optimal (green) , poor (amber) or critical (red).

For example, for the humidity subsystem, the current sensor scale values show the upper bound to be 100 and the lower bound to be 0. These values are taken from humidity_bound = [0,100] from the config file. The other thresholds are stored in humidity_extr = [40, 50, 60, 75].

These values demonstrate:

- Lower critical range (red): between 0% and 40%.
- Lower poor range (amber): between 40% and 50%.
- Optimal range (green): between 50% and 60%.
- Upper poor range (amber): between 60% and 75%.
- Upper critical range (red): Between 75% and 100%

There are two thresholds lists for each of the four subsystems.
Configuration file example in .ini format:

```ini
[Plant Information]
name = New Plant
brightness_extr = [100, 520, 640, 1000]
brightness_bound = [0, 1200]
brightness_sensor_unit = lux
brightness_actuator_unit = %%
humidity_extr = [40, 55, 60, 90]
humidity_bound = [0, 100]
humidity_sensor_unit = %%
humidity_actuator_unit = %%
temperature_extr = [10, 16, 20, 30]
temperature_bound = [0, 60]
temperature_sensor_unit = °C
temperature_actuator_unit = °C
water_level_extr = [10, 18, 44, 70]
water_level_bound =  [0, 100]
water_level_sensor_unit = %%
water_level_actuator_unit = seconds
```

If a file of a different format is uploaded, it will not be accepted.