

AI Hydroponics Project Step-by-Step Guide

Things used in the project:

Hardware components:

Components	Quantity
Raspberry Pi	1
Total dissolved solid sensor, SEN0244	1
Water level sensor	1
Light sensor, SI1145	1
Analogue-to-Digital converter	1
Growing lights	4
Submersible water pump	1
Peristaltic pump	1
Container for water	1
Plant holder	12
UPVC pipes (0.50m)	3
UPVC pipes (0.20m)	4
Tee joints	4
90° joints	7

Software components:

App(Dashboard):

- IBM Node-RED

AI Voice Assistant:

- IBM Watson Assistant
- IBM Watson Speech-to-Text
- IBM Watson Text-to-Speech

Database:

- Erlang (Web server on Inets, Database on Mnesia)
- Jiffy (3rd party Erlang JSON parser library)
- EMQTT (3rd party Erlang MQTT client/broker library)

Setting up App(Dashboard):

1. Create an IBM Cloud account.
2. Create a node-red app instance on IBM Cloud.
3. Deploy it using Cloud Foundry.
4. Add the required libraries to the package.json file in the app's gitlab repository.

```
{
  "name": "node-red-app",
  "version": "1.1.3",
  "dependencies": {
    "node-red-dashboard": "3.1.7",
    "node-red-node-ui-microphone": "0.3.x",
    "node-red-contrib-mic": "0.0.1",
    "node-red-node-random": "0.4.0",
    "node-red-contrib-scx-ibmiotapp": "0.0.49",
    "node-red-contrib-cloudantplus": "2.0.5",
    "node-red-contrib-postgresql": "0.10.1",
    "node-red-contrib-httpauth": "1.0.12",
    "node-red-node-email": "1.15.1",
    "node-red-contrib-chat": "1.0.0",
    "@ibm-cloud/cloudant": "^0.0.25",
    "bcrypt": "^5.0.1",
    "body-parser": "1.x",
    "express": "4.x",
    "http-shutdown": "1.2.2",
    "ibm-cloud-env": "^0",
    "node-red": "^2.2.2",
    "node-red-contrib-ibm-db2": "0.x",
    "node-red-node-cf-cloudant": "0.x",
    "node-red-node-openwhisk": "0.x",
    "node-red-node-watson": "0.x",
    "node-red-nodes-cf-sqlldb-dashdb": "0.x"
  },
  "scripts": {
    "start": "node --max-old-space-size=160 index.js --settings ./bluemix-settings.js -v"
  },
  "engines": {
    "node": "14.x"
  }
}
```

5. Use the node-red-dashboard library to set up displays for the data and test it using the inject nodes.
6. Create a flow to allow custom type inputs using the form node.
7. Use the html template node to style the different display nodes with css.
8. Setup the connection to the database using HTTP nodes, and replace the inject nodes with inputs from the database.

Setting up AI Voice Assistant:

Getting the required credentials:

1. Create an IBM Cloud account
2. Create a Speech-to-Text instance on IBM Cloud

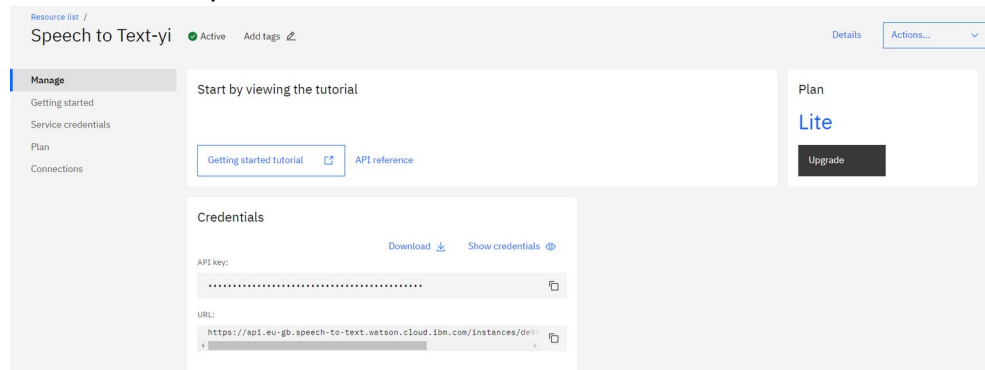


Fig x. Creating a Speech-to-Text instance

3. Create a Text-to-Speech instance on IBM Cloud
4. Create a Watson Assistant instance on IBM Cloud

Training Watson Assistant:

1. Create an assistant inside the Watson Assistant workspace

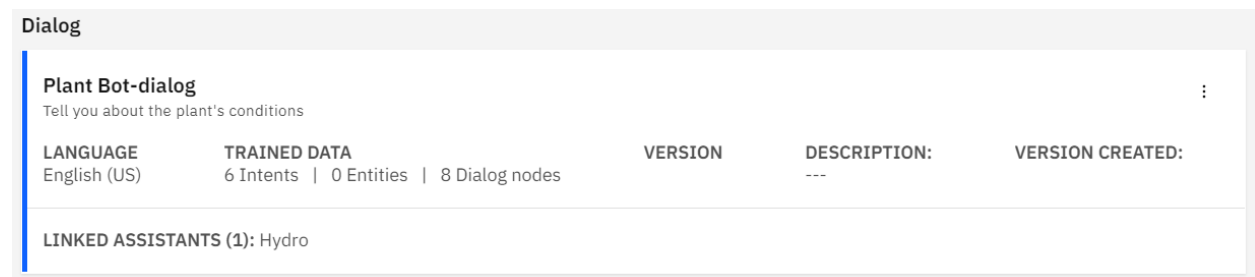


Fig x. Creating an assistant

2. Create required intents, i.e., water, nutrient, light, greetings, endings, condition etc.

<input type="checkbox"/> Intents (6) ↑	Description	Modified ↑↓	Examples ↑↓
<input type="checkbox"/> #condition	Ask about condition	8 days ago	7
<input type="checkbox"/> #endings		8 days ago	5
<input type="checkbox"/> #greetings		5 days ago	5
<input type="checkbox"/> #light	ask about light	8 days ago	9
<input type="checkbox"/> #moisture		8 days ago	6
<input type="checkbox"/> #nutrient	Ask about nutrients	8 days ago	6

3. Add user examples for each intent

<input type="checkbox"/>	is it bright enough
<input type="checkbox"/>	is it too dark
<input type="checkbox"/>	what is the light intensity

Fig x. User examples that indicate light intent

4. Add nodes in dialog section for each intent

5. Add response in each node

Assistant responds

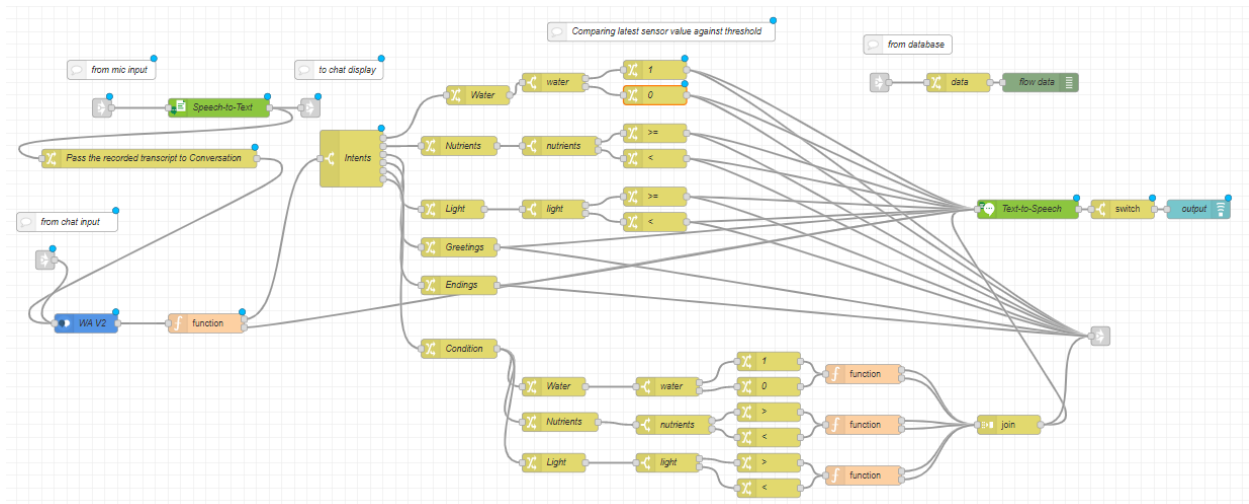
Text

^ v

The water level is adequate.

Fig x. Example response if moisture intent is detected

Integrating with Node-RED:



1. Install Watson Studio package on Node-RED
2. Add the Assistant, Speech-to-Text and Text-to-Speech node onto the Node-RED workspace
3. Configure the nodes by adding the respective credentials from IBM Cloud

✎ Name

🔑 API Key

✎ Service Endpoint

✎ Assistant ID

⌚ Timeout Period

☐ Switch on Debug

☐ Restart Dialog

☒ Return Context

☒ Return Alternate Intents

☒ Multiple Sessions

Fig x. Configuring Watson Assistant node on Node-RED

4. Configure connection between nodes by ensuring message received is in the correct format and circumventing undefined messages
5. Extract user intent and set up corresponding action
6. Connect to database to get latest sensor value and compare against optimal threshold
7. Add a microphone node and an output node so that users can use this feature on the dashboard

Setting up Database:

1. Set up the environment

Note that you will need to update lines 8, 9, 11 and 12 of the file ``mqtt/setup_mqtt.erl`` to reflect your MQTT broker configuration in the github repo before you can receive updates from hydroponic systems. If you are using the included Dockerfile, then most of this section can be skipped. Currently the Dockerfile has no support for the server to run over HTTPS so it is recommended to use the manual setup.

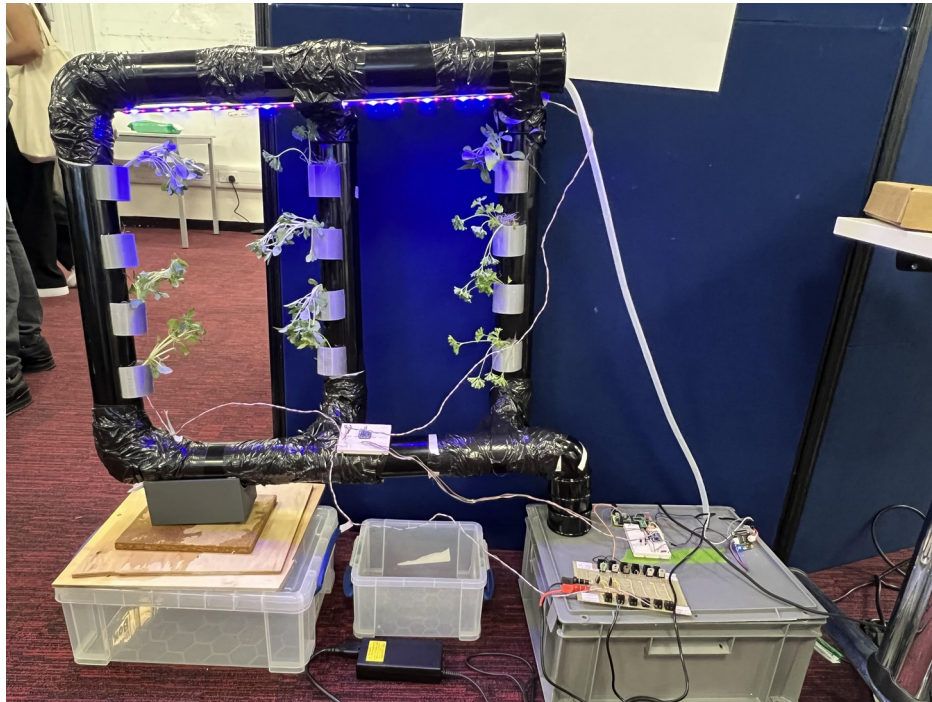
1. Clone the git repo into the /src folder on a cloud VM.
2. Run the setup scripts ``setup_users.sh`` & ``setup_mnesia.sh`` as root to create the filesystem & users.
3. If you are using HTTPS, run ``sudo iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-port 46972``, otherwise replace 443 with 80.
4. If you are using HTTPS, uncomment line 28 in ``/src/web/setup_web.erl`` and update lines 14, 15 and 16 to point to your ssl certificates. These ssl certificates should be accessible (read only/0440) by the user ``web-inets``.

2. Run the server

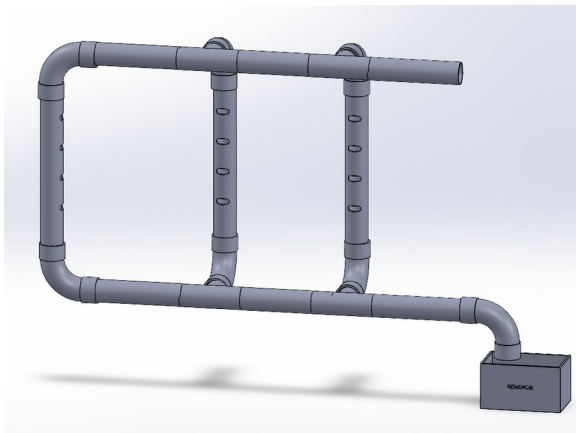
1. To run the server, run ``sudo /src/start.sh`` from anywhere. This will spawn the relevant applications to handle the database, web-server and mqtt server.
2. If you are using the Dockerfile, ensure you expose port 46972 as 80 to allow HTTP connections.

Setting up Hardware:

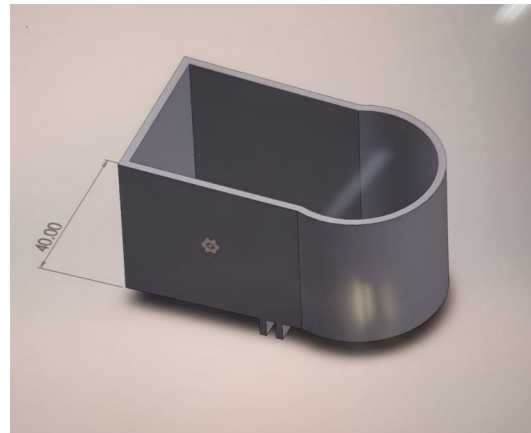
Setting up system's frame: **Physical Build**



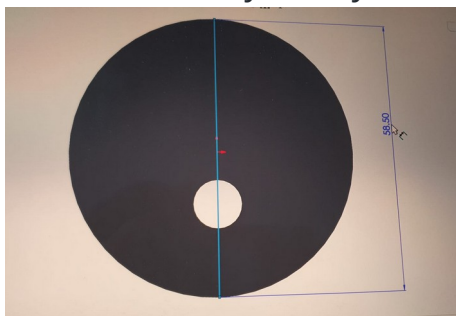
Picture of the Farm



Schematic for Physical System of the Farm



3D-Printed Plant Holder



3D-Printed Pipe Stopper



3D-Printed Water Level Barrier

1. Measure the overlap distance between the 90° joints & UPVC pipes and tee joints & UPVC pipes
2. Drill 4 rectangular holes with the size of 5cm x 5.1cm on each of the 0.5m UPVC pipes
3. Measure the diameter of 90° joint
4. Drill the hole with a diameter of 68mm on the water reservoir (the diameter of 90° joint)
5. Connect the joints with UPVC pipes according to the given configuration
6. 3D-print the plant holders as given in the figure and install on rectangular holes
7. 3D-print the pipe stopper given in the figure and install on the free side of UPVC pipes at the top
8. 3D-print the water level barrier to distribute the water flow evenly. Install this barrier at the joints on the top branch of the system. (The two tee joints and single 90° joint)
9. Install the submersible water pump inside the water reservoir. Open two little holes on top of the reservoir lid for this, one for the pump hose and the other for power cable of the motor. Connect the power cable to automation system and hose to the 3D-printed pipe stopper on the free side UPVC pipes at the top.
10. Install the water level sensor at the top the reservoir with electronics placed outside for connection to the main system.
11. Install the peristaltic pump with one hose inside the nutrient reservoir and the other inside the water reservoir. For this open another small hole on top of the water reservoir specifically for this hose.
12. Install the dissolved solid sensor inside the nutrient reservoir. Connect this to the outside electronics sharing the same hole as water sensor power cable.
13. Before starting the system, do a test run and check for leakages. If found, hot glue the affected area. (This mostly happens at joint merging points)
14. Fill your water & nutrient reservoirs, and plant your seeds in the 3D-printed plant holders
15. You are now ready to experiment with hydroponics. Good luck!

Setting up Raspberry Pi and Sensors:

**** insert circuit diagram**

- a. Connect a Raspberry Pi to a breadboard

- b. Connect an analogue-to-digital converter to the Raspberry Pi
- c. Connect the water level and total dissolved solid sensor to the analogue-to-digital converter by using long wires
- d. Connect the light sensor and analogue-to-digital converter to the Raspberry Pi by using I2C connection