

IoT Sensor Project (Smart Plant Sensors)

sddec21-08

Team Leader - ztk@iastate.edu

Zach Kauffman - Software/Firmware

Mason Gil - Software

Daniel Phalen - Hardware

Thomas Smeed - Hardware

Walter Gilbert - Software

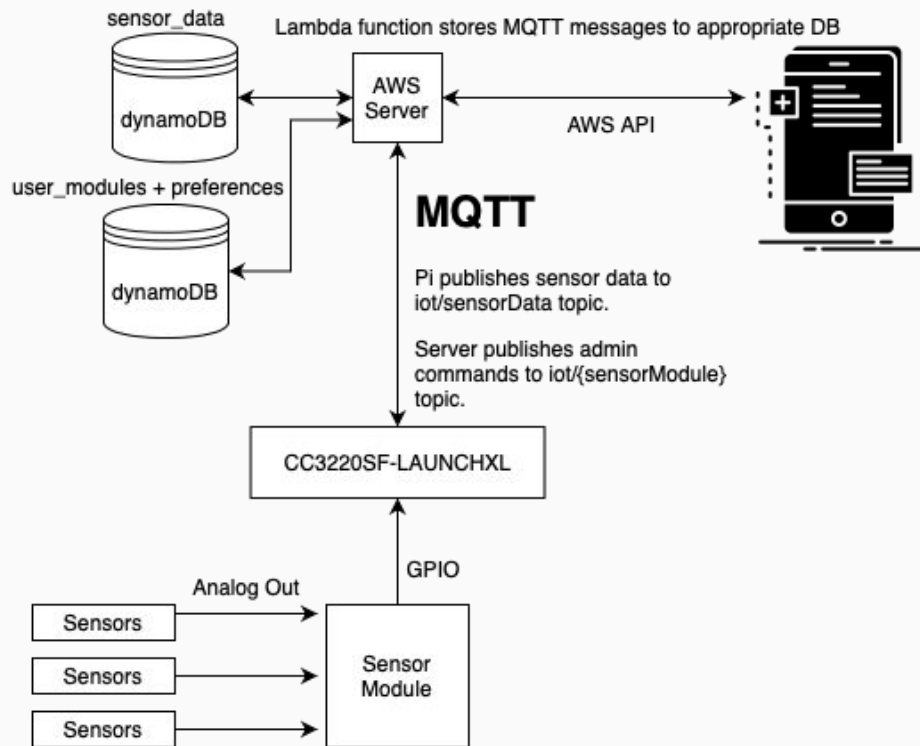
Advisor: Daji Qiao

Client: Mark Easley / Texas Instruments

Project Recap

Project Recap

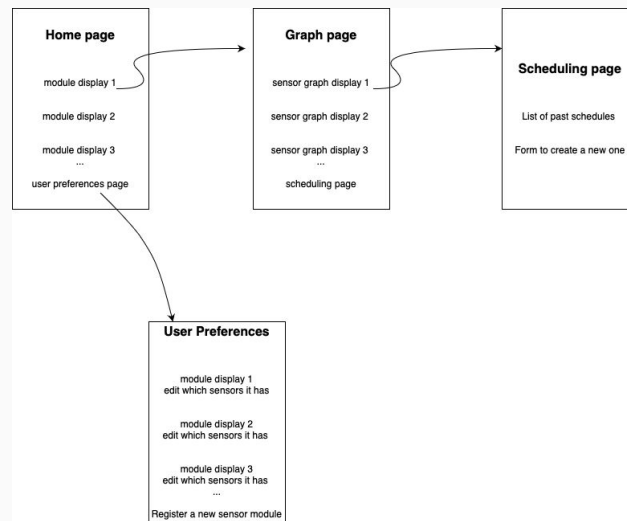
- Smart garden ecosystem with hardware and software components
- Daughter board that interfaces with sensors and talks to TI launchpad
- Web-based interface that allows users to monitor the status of their plants remotely
- Focus on modular design with arbitrary sensors and a flexible user preferences setup



On Connect

Grabs most recent DB value to display to user

Subscribes to iot/sensorData to refresh as new data comes in



Goals for the Semester

- Graph view for sensors
 - See how the temperature has been changing over time, watch moisture levels, etc.
- User preferences + alerts
 - Set up thresholds for sensor values and let user know when values are out of bounds
 - Visual indication on the board (simple LED to start)
- Daughter board and integration with gateway

Progress

Progress Since 9/23

- Finished up graph view
- User preference system is complete
- Alert system is in place, along with user schedules
- PCB is finished and ordered
- Testing sensors with the Launchpad
- Have components for the daughterboard

Graph view

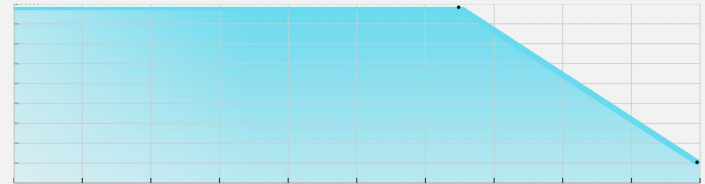
zk-garden-1

ztk@iastate.edu

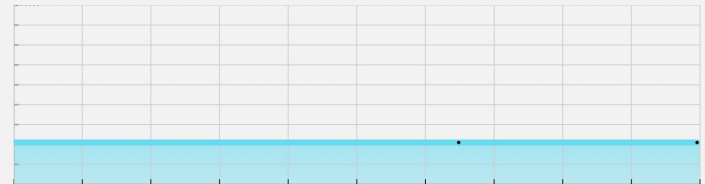
Temperature: 53 °
Moisture: 4 lb/ft³

 [VIEW SYSTEM](#)

Temperature



Moisture



[VIEW SCHEDULING](#)

Notification System

- Built on top of AWS's Simple Notification Service (SNS)
 - Easy to integrate
 - Flexible publish/subscribe architecture that meets our needs
 - Supports SMS + email alerts
 - Supports message filters (which we use to filter out only specific users)
- AWS Lambda function triggers whenever data comes in
 - Checks if the user who the data belongs to has any thresholds set up
 - Checks if the data is in/out of bounds with respect to those thresholds
 - Emails the user if it's not
 - Sends the board "status=good" or "status=bad" message

User is subscribed to the appropriate SNS topic (with the proper Message Attribute filter) when they first register with the site (with the email they registered with).

Amazon SNS > Topics > sensor-oob-alerts > Subscription: 6b145133-5c5d-4ef9-adf9-f520c774038c

Subscription: 6b145133-5c5d-4ef9-adf9-f520c774038c Edit Delete


Details

ARN arn:aws:sns:us-east-2:425047544756:sensor-oob-alerts:6b145133-5c5d-4ef9-adf9-f520c774038c	Status ✔ Confirmed
Endpoint ztk@iastate.edu	Protocol EMAIL
Topic sensor-oob-alerts	

Subscription filter policy Redrive policy (dead-letter queue)

Subscription filter policy
This policy filters the messages that a subscriber receives. [Info](#)

```
{
  "user": [
    "ztk@iastate.edu"
  ]
}
```


 **SensorAlerts** <no-reply@sns.amazonaws.com>
to me ▾
⋮

The following sensors are currently out of bounds:
moisture: 55 > 50
temperature: 71 > 65

--

If you wish to stop receiving notifications from this topic, please click <https://sns.us-east-2.amazonaws.com/unsubscribe.html?SubscriptionId=6b145133-5c5d-4ef9-adf9-f520c774038c>

Please do not reply directly to this email. If you have any questions or

 **SensorAlerts** <no-reply@sns.amazonaws.com>
to me ▾
⋮

The following sensors are currently out of bounds:
temperature: 72 > 65

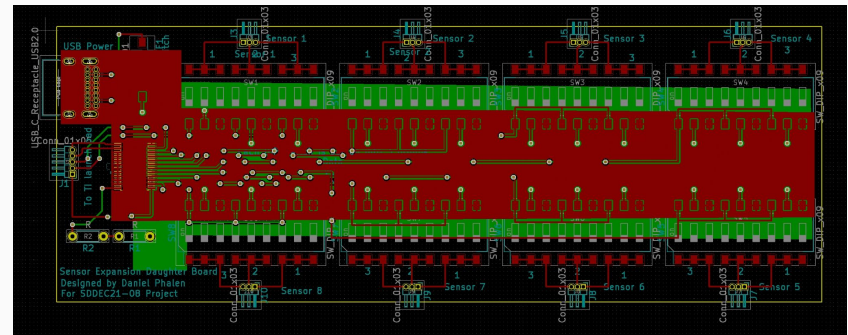
⋮

↩ Reply ➦ Forward

Challenges

Challenge: Flexibility on the firmware side

- Key part of project is flexibility. This means arbitrary sensors, arbitrary ports, etc.
- Software is easy to make flexible
- Hardware has flexibility built in
- How to make the main firmware control loop flexible enough that it knows what sensors are hooked up to it?

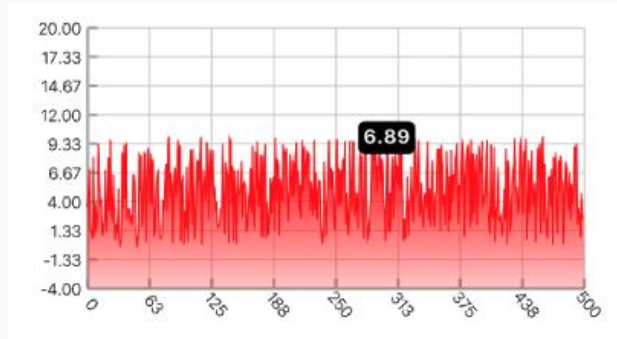


Solution

- To achieve firmware flexibility, utilize the internet connectivity of the TI Launchpad
- Empty MQTT request on boot
- AWS Lambda function sees the request, sends down a message containing a list of sensors `module_sensors`
- Control loop considers `module_sensors[0]` to be the sensor hooked up on the daughter board's Sensor 1 port, etc

Challenge: Working with software libraries

- Most graph libraries for React Native have poor documentation
- It took a few implementations of different libraries to get our graph functionality working
- There isn't a date picking library for React Native that supports web integration



Questions?

ztk@iastate.edu