

System & Unit Test Report

Tiny House Sensor Network
The Smart Sensor Network Team
30 November 2015

BACKEND (FIRMWARE & HARDWARE) SCENARIO - Sargis Yonan & Joseph Ou:

User Story 1 from Sprint 1: “As a user I want to be able to view house data”

User Story 2 from Sprint 2: “As a user, I want actual sensor data to be coming from tested or implemented sensors”

User Story 2 from Sprint 3: “As a user, I want to wirelessly access raw data so that I can access it remotely and then manipulate the machine”

Scenario: I want to check the status of my water heater in my sensor network, and change its setpoint temperature if it is under 100 degrees celsius

Example:

1. Follow the installation instructions on:
https://github.com/shivamndave/tiny_house/blob/master/docs/Installation/Firmware_%26_Hardware_Installation_Guide.pdf to setup up the smart controller environment for your specific microcontroller and needs
2. Once the system is installed and initialized, send a GET_STATUS message to the microcontroller to check the current temperature.
 - with XCTU, send the following 3 bytes (GET_STATUS_COMMAND, 0, '-'): → 0x33 0x00 0x2D
3. A large packet (usually over 40 bytes is returned). The packet will be in the form:
/current_temperature/current_state/setpoint/positiveOffset/negativeOffset/
The current temperature will be found in between the first and second forward slashes.
For example, you might receive the following packet: “/90/1/85/5/5/”, meaning the current temperature the sensor is reading is 90 degrees celsius, you are currently in state one, the cooling state, meaning the pin connected to heating element is off (0 Volts outputting through it) -- see: tiny_house/blob/master/src/firmware/Sensor_Driver/fsm.h for state definitions
4. To change the current setpoint, send the CHANGE_SETPOINT byte, followed by a byte representing the value in hex which is the desired setpoint, followed by the '-' delimiter. In this case, we'll set the new setpoint to 100 degrees celsius (Note: 100 as a decimal is hexadecimal 64)

→ 0xBB 0x64 0x2D

The acknowledgement 0xDC should be sent indicating the setpoint was changed successfully.

5. If you were to GET_STATUS now, you should receive the packet: “/93/2/100/5/5/”. This means that the current temperature is 93 degrees celsius, the heater is currently heating, the setpoint is set to 100 degrees celsius, and both the negative and positive offsets are 5 degrees. This indicates that the setpoint was changed because due to the current temperature being below the deadband, the heating element is triggered

UNFINISHED USER STORIES:

User story 2 is not complete. One task, in which we intended to have two (or multiple) Arduinos communicating wirelessly with the control board, was not accomplished.

FRONTEND (Dashboard)

User Story 2 from Sprint 1: “As a user, I want to view a basic web application dashboard to get an idea of how my house data will be displayed.”

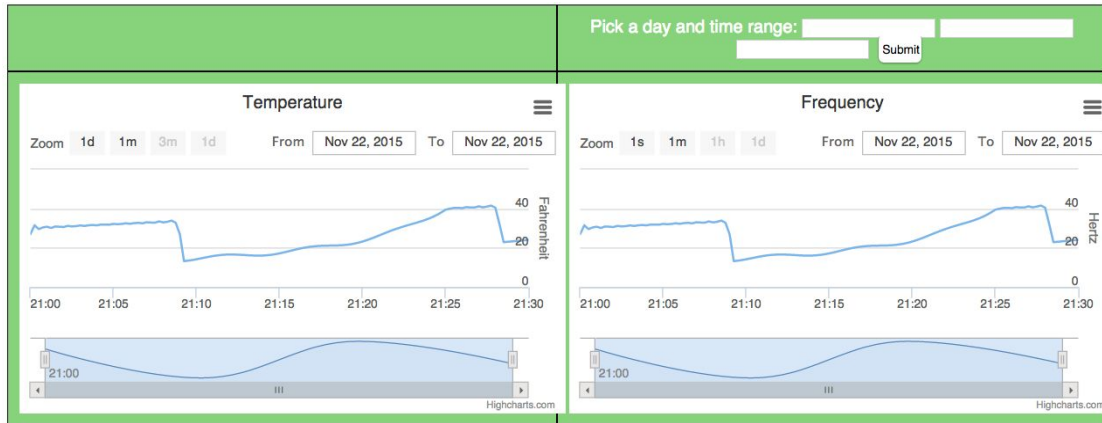
Scenario 1, Basic view of the tiny house html dashboard with no data:

1. Clone github repo at https://github.com/shivamndave/tiny_house
2. Visit app folder located at /src/app in the cloned repository directory
3. Open file named tiny.html

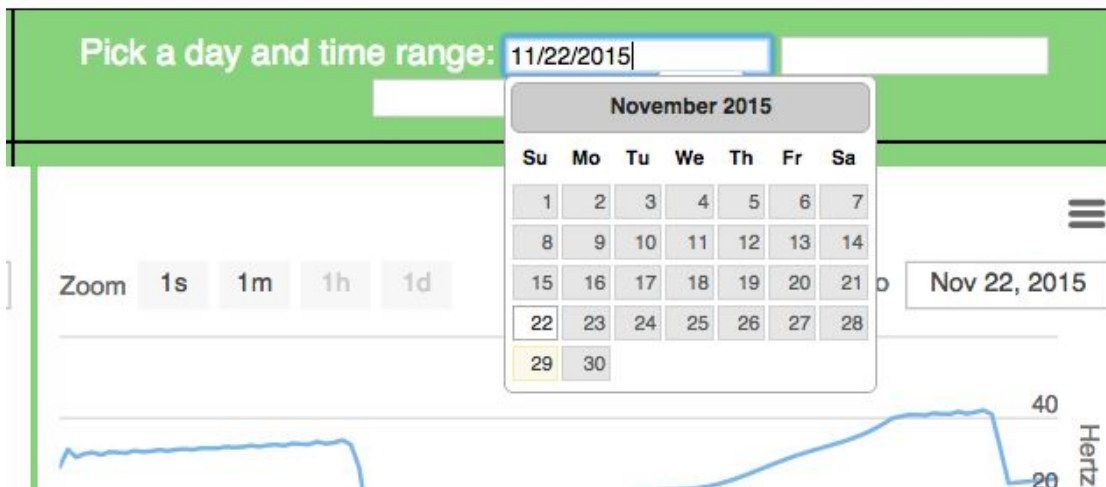
User Story 1 from Sprint 2 & 3: “As a user, I want an improved web application dashboard that shows sample or actual data.”

Scenario 1, Querying for an existing range that contains temperature and frequency data:

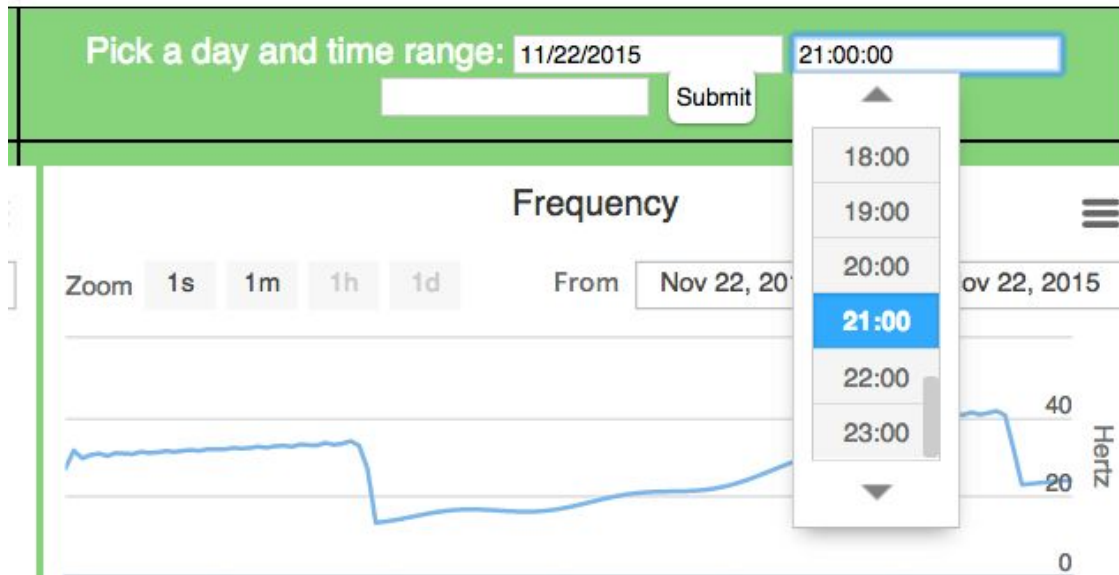
1. Prepare database (Use frontend database installation)
2. Clone github repo at https://github.com/shivamndave/tiny_house
3. Move the app folder located at /src/app in the cloned repository directory to the /var/www/html directory located on your virtual machine server
4. Open file named tiny.html through the url: “localhost:8080/app/tiny.html”
5. If the database has been properly setup the default sample data should be showcased from November 22, 2015 at approximately 21:00:00 until approximately 21:30:00



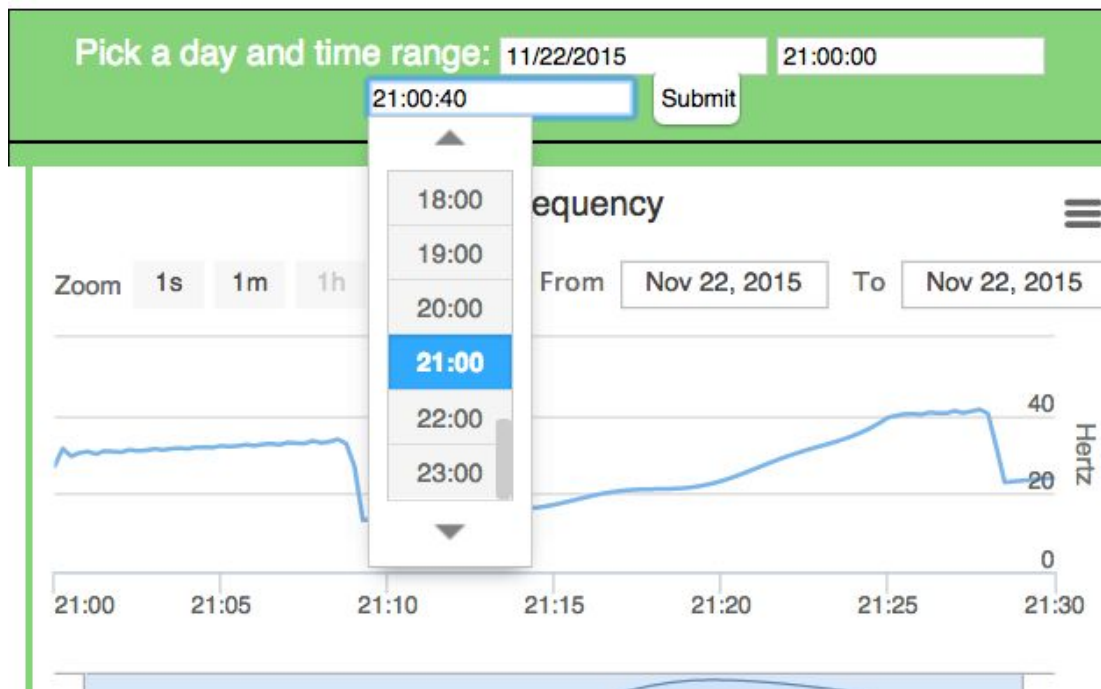
6. Next the user can query a specific time
 - a. First enter a day on which you want to query for, in this case enter “November 22, 2015” or “11/22/2015”



- b. Next enter a start time. The time is a 24 hour clock where 0:00 is 12am midnight (at the start of the day and end of previous day) and 24:00 is 12am midnight (at the end of the day and start of next day). In our case enter 21:00:00



- c. Next enter an end time, similar to the last step. In our case enter 21:00:40 to have a period of 21:00:00 to 21:00:40, approximately 40 seconds

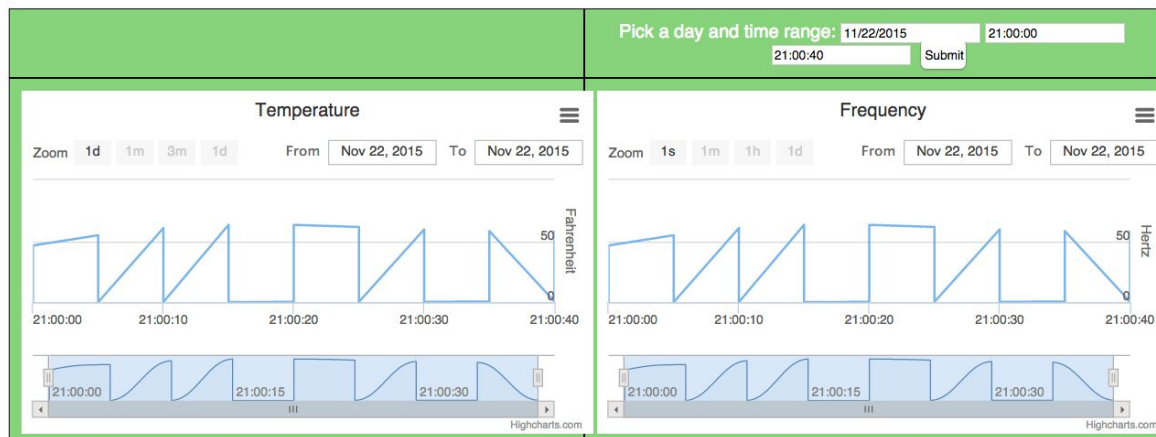


- d. Last press the submit button in order to query for November 22, 2015 from 21:00:00 to 21:00:40

Pick a day and time range: 11/22/2015		21:00:00
21:00:40		Submit

Frequency		≡
Zoom	1s 1m 1h 1d	From Nov 22, 2015 To Nov 22, 2015

7. The website will refresh to show data based on the query you gave, if the query was a viable time period

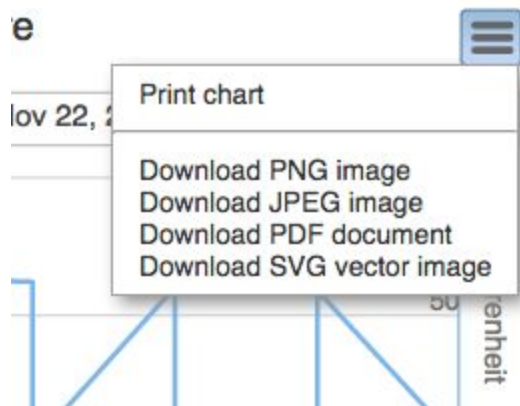


8. If the time period queried did not have any data (sensors under maintenance or other reason) than a message will appear letting you know of this.

Pick a day and time range: 11/2/2015 21:00:00 21:00:40 Submit	
Temperature sensors were under maintainance. Charts cannot be displayed	Frequency sensors were under maintainance. Charts cannot be displayed

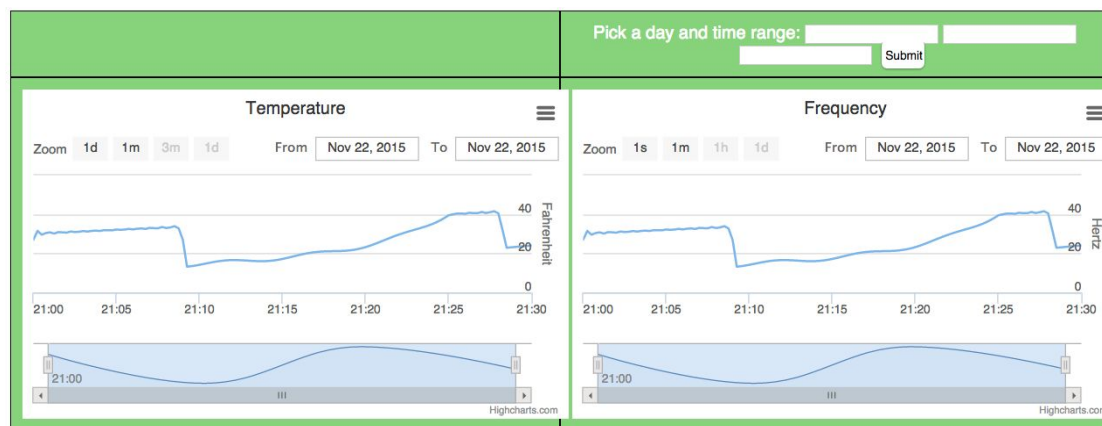
Optional Additional Feature Step:

9. Downloading the graphs can be accessed through the 3 line icon in the upper left. Select the type of file you would want the downloaded image to be in. Options range from PNG, JPEG, PDF, and SVG



Scenario 2, Querying for a non-existing range

1. Prepare database (Use frontend database installation)
2. Clone github repo at https://github.com/shivamndave/tiny_house
3. Move the app folder located at /src/app in the cloned repository directory to the /var/www/html directory located on your virtual machine server
4. Open file named tiny.html through the url: "localhost:8080/app/tiny.html"
5. If the database has been properly setup the default sample data should be showcased from November 22, 2015 at approximately 21:00:00 until approximately 21:30:00



6. Next the user can query a specific time (for images for these steps view step 5 on scenario 1, however enter information below instead)
 - a. First enter a day on which you want to query for, in this case enter "November 2, 2015" or "11/2/2015"
 - b. Next enter a start time. The time is a 24 hour clock where 0:00 is 12am midnight (at the start of the day and end of previous day) and 24:00 is 12am midnight (at the end of the day and start of next day). In our case enter 21:00:00
 - c. Next enter an end time, similar to the last step. In our case enter 21:00:40 to have a period of 21:00:00 to 21:00:40, approximately 40 seconds

- d. Last press the submit button in order to query for November 2, 2015 from 21:00:00 to 21:00:40
7. The time period queried does not have any data (sensors under maintenance or other reason) so a message will appear letting you know of this.

	Pick a day and time range: 11/2/2015 21:00:00 21:00:40 Submit
Temperature sensors were under maintainance. Charts cannot be displayed	Frequency sensors were under maintainance. Charts cannot be displayed