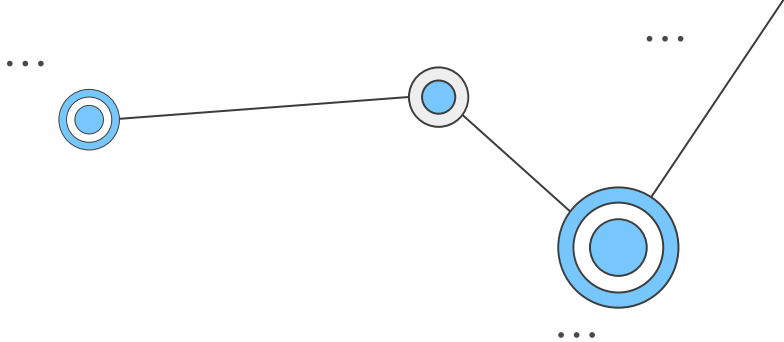


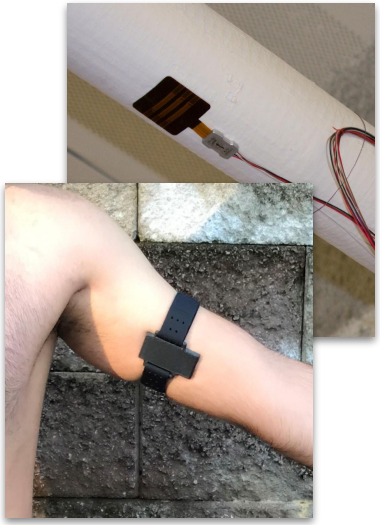
ThermaSENSE App

Ansh Gwash, Jack Gable, Yizhe Liu
CS 4624 Multimedia, Hypertext, and Information Access Capstone
Dr. Fox
Virginia Tech, Blacksburg, VA 24061
12/5/2022

Overview

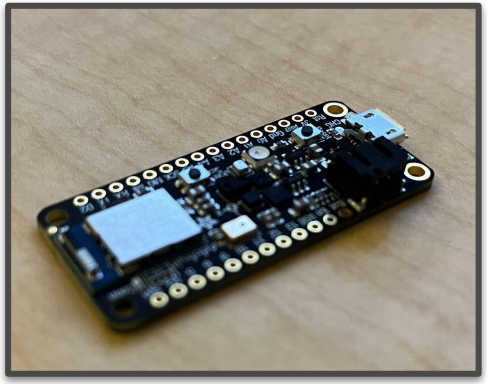
Project Recap

ThermaSENSE devices transmit data via **bluetooth**.
We need to **record** and **visualize** this data through an **app-to-cloud** pipeline.



Completed Work

Board Updates



Adafruit nrf52840 Sense

Represents a dummy ThermaSENSE device

Data Identifiers

Flexible data formats for the app to process

Buffer

Offline data stored until reconnected

Over the Air Updates

OTA updates for future firmware changes

Data Identifiers

```
|5473|26.659|23.071|28.084|63.751|101|0x0*
|6473|26.660|22.308|28.080|64.819|101|0x0*
|7475|26.662|23.590|28.082|65.613|101|0x0*
|8474|26.663|24.445|28.080|66.589|101|0x0*
|9474|26.665|23.804|28.079|65.704|101|0x0*
|10476|26.666|22.583|28.080|66.711|101|0x0*
|11475|26.668|23.315|28.081|66.833|101|0x0*
|12475|26.670|23.743|28.081|66.803|101|0x0*
|13477|26.672|23.346|28.085|66.376|101|0x0*
|14476|26.675|24.506|28.086|66.589|101|0x0*
|15477|26.677|24.292|28.088|66.772|101|0x0*
|16479|26.679|23.590|28.091|66.589|101|0x0*
|17479|26.682|24.384|28.091|66.559|101|0x0*
|18480|26.685|22.766|28.094|67.139|101|0x0*
|19482|26.687|23.743|28.097|68.604|101|0x0*
|28488|26.715|24.414|28.128|69.458|101|0x0*
```

```
0x0|15545|T1|20.542|Q1|0.000|T2|20.542|
Q2|0.000|HR|180.0|AC|5.0|OT|10.0|BT|92|
0x0|15577|T1|20.542|Q1|0.000|T2|20.542|
Q2|0.000|HR|180.0|AC|5.0|OT|10.0|BT|89|
0x0|15635|T1|20.542|Q1|0.000|T2|20.542|
Q2|0.000|HR|180.0|AC|5.0|OT|10.0|BT|
100|0x0|15666|T1|20.542|Q1|0.000|T2|
20.542|Q2|0.000|HR|180.0|AC|5.0|OT|
10.0|BT|86|0x0|15724|T1|20.542|Q1|
0.000|T2|20.542|Q2|0.000|HR|180.0|AC|
5.0|OT|10.0|BT|85|0x0|15755|T1|20.542|
Q1|0.000|T2|20.542|Q2|0.000|HR|180.0|
AC|5.0|OT|10.0|BT|81|0x0|15815|T1|
20.542|Q1|0.000|T2|20.542|Q2|0.000|HR|
180.0|AC|5.0|OT|10.0|BT|90|0x0|15875|
T1|20.542|Q1|0.000|T2|20.542|Q2|0.000|
HR|180.0|AC|5.0|OT|10.0|BT|88|0x0|
15934|T1|20.542|Q1|0.000|T2|20.542|Q2|
0.000|HR|180.0|AC|5.0|OT|10.0|BT|83|
0x0|15965|T1|20.542|Q1|0.000|T2|20.542|
Q2|0.000|HR|180.0|AC|5.0|OT|10.0|BT|92|
0x0|15997|T1|20.542|Q1|0.000|T2|20.542|
Q2|0.000|HR|180.0|AC|5.0|OT|10.0|BT|81|
0x0|16027|T1|20.542|Q1|0.000|T2|20.542|
Q2|0.000|HR|180.0|AC|5.0|OT|10.0|BT|79|
0x0|16084|T1|20.542|Q1|0.000|T2|20.542|
Q2|0.000|HR|180.0|AC|5.0|OT|10.0|BT|83|
0x0|16176|T1|20.542|Q1|0.000|T2|20.542|
Q2|0.000|HR|180.0|AC|5.0|OT|10.0|BT|94|
0x0|16207|T1|20.542|Q1|0.000|T2|20.542|
Q2|0.000|HR|180.0|AC|5.0|OT|10.0|BT|81|
0x0|16237|T1|20.542|Q1|0.000|T2|20.542|
Q2|0.000|HR|180.0|AC|5.0|OT|10.0|BT|90|
0x0|16267|T1|20.542|Q1|0.000|T2|20.542|
Q2|0.000|HR|180.0|AC|5.0|OT|10.0|BT|80|
```

```
0x0|15577|T1|20.542|Q1|0.000|T2|20.542|
Q2|0.000|HR|180.0|AC|5.0|OT|10.0|BT|89|
```

Offline Buffer

If the board cannot connect to bluetooth, store sensor data in buffer until connected.

Uses a circular queue to store sensor data, and empty the buffer once the board is connected to the app.

```
if (connected) // We are connected with the app
{
  //Serial.print("CONNECTED: ");
  if (first_connection){ // Upon first connection, allow 5 second wait for UART to turn on
    delay(5000);
    first_connection = false;
  }

  // SEND DATA FROM BUFFER TO bleuart.write()
  String message = "BAD VAL";
  while (buffer_pointer_front != -1) { // Modified circular queue to always empty buffer when possible
    message = buffer[buffer_pointer_front];
    if (buffer_pointer_front == buffer_pointer_rear){
      buffer_pointer_front = -1;
      buffer_pointer_rear = -1;
    }
    else if (buffer_pointer_front == BUFFER_NUM - 1) {
      buffer_pointer_front = 0;

    }
    else {
      buffer_pointer_front++;
    }
    //Serial.println("BUFFER PULL: " + message);
    bleuart.write(message.c_str(), message.length());

    // String strOTA = "---OTA TEST---";
    // bleuart.write(strOTA.c_str(), strOTA.length());
  }
```

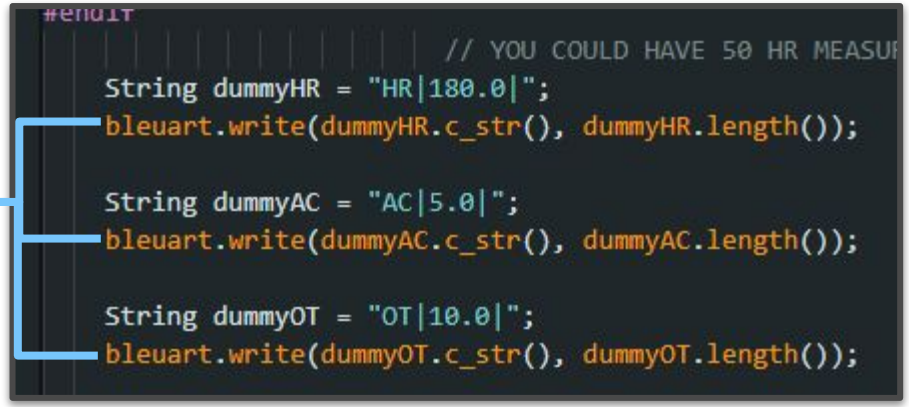
This code is run on constantly on the device, and unloading the buffer using bleuart.write()

Over the Air Updates

Firmware - Arduino sketches (programs) that can change in the future

Clients may need to update their devices' firmware in order to be compatible with the app and cloud storage. This update should be done through the app with Bluetooth, called OTA updates.

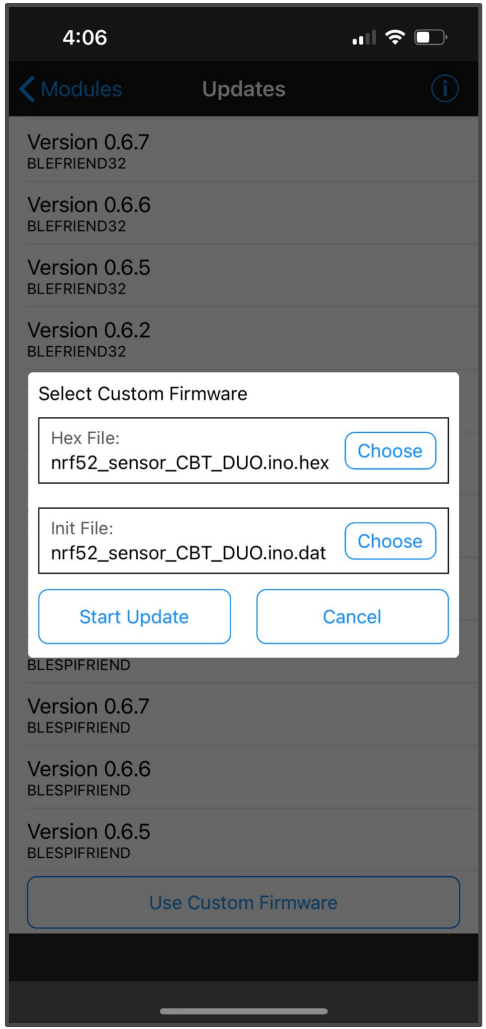
Examples of placeholder data



Over the Air Updates

How to send updates to Adafruit nrf52840 board? -
Not supported by Adafruit

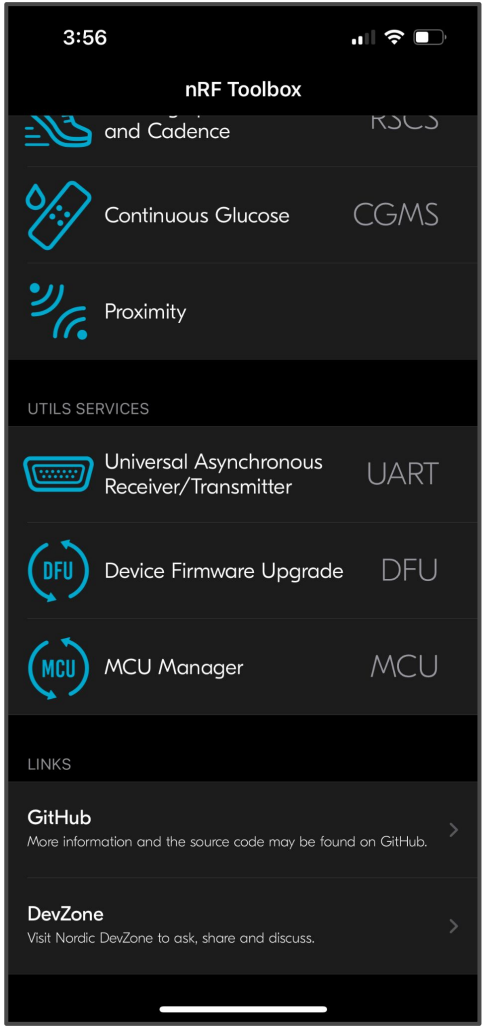
Arduino creates a .hex file and .dat file that can be sent as a custom firmware update through the Bluefruit Connect App (app created by Adafruit).



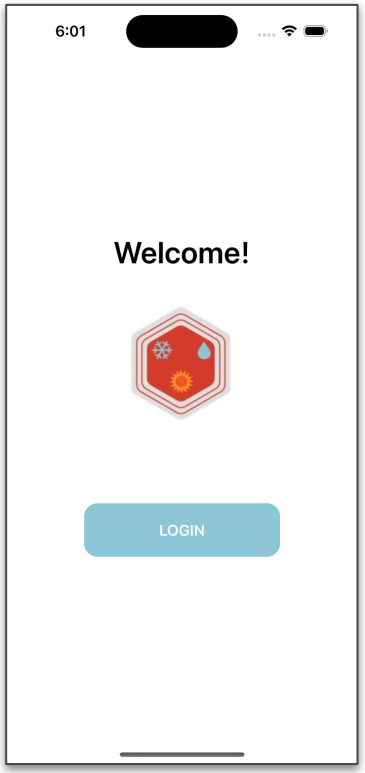
Over the Air Updates

How to send updates to Adafruit nrf52840 board? -
Not supported by Adafruit

Using a third-party nRF Toolbox to send the .zip file of the Arduino sketch directly to the board.



App Updates



ThermaSENSE App

Developed with Swift and Xcode

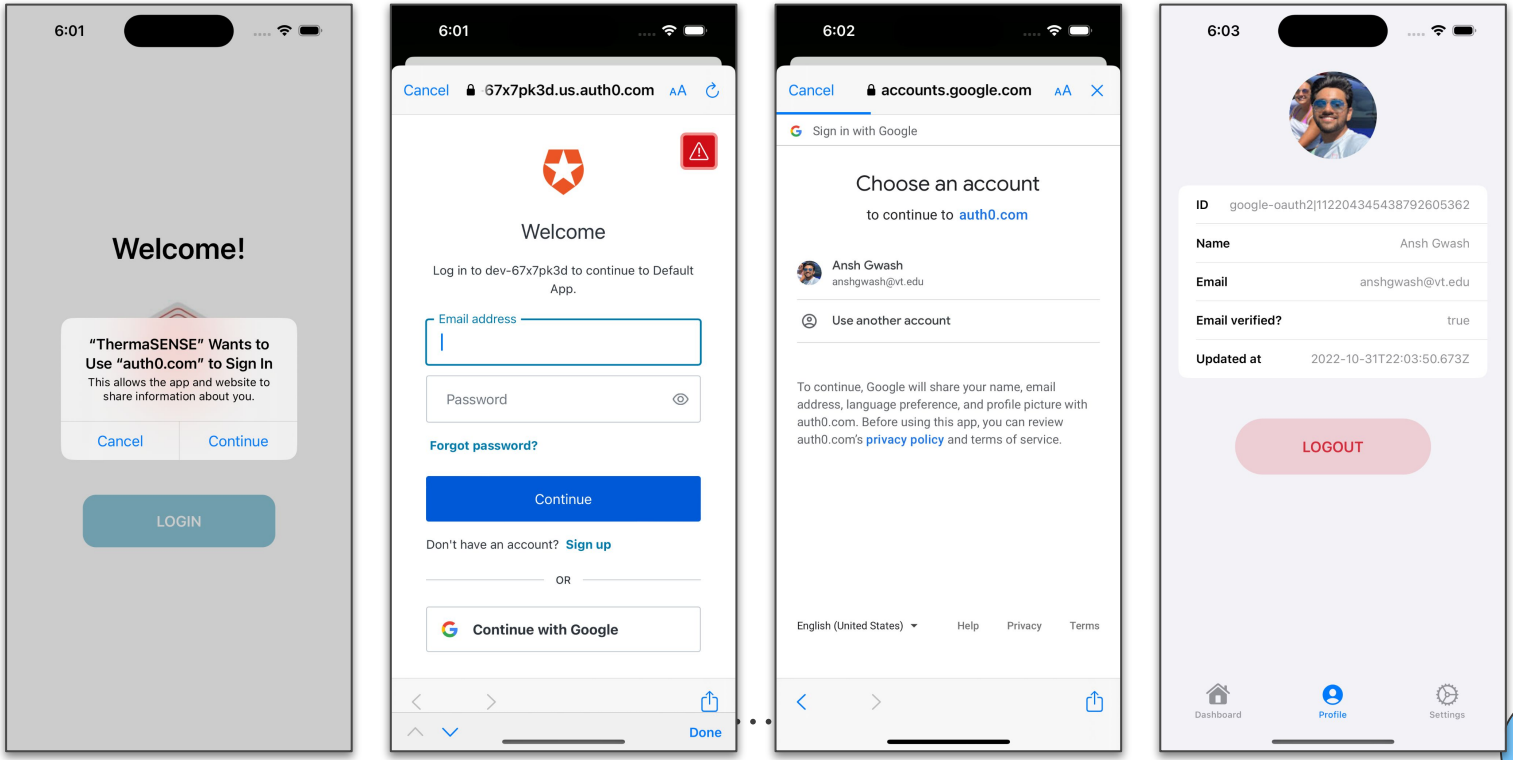
Login / Signup

Uses OAuth to authenticate users and let them sign in

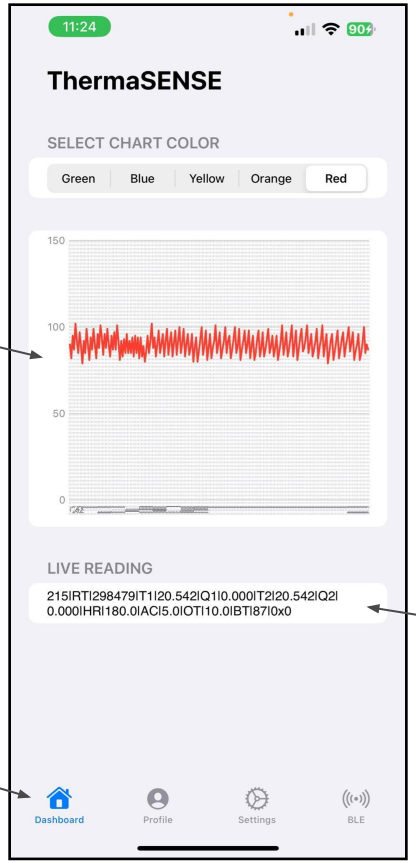
Dashboard

Visualizing the data in app based on the caught data and historical cloud data

Login / Signup



Dashboard



Live visualization of the sensor data

Live reading of the raw sensor data (This will be stored in the cloud)

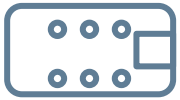
Toolbar at the bottom for navigation

AWS PROBLEMS

Conclusion

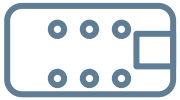
Final Week Tasks

- Documentation
- Finalize cloud connections
- Bring all builds into one app
- Walkthrough with client - how to update board code, app, and manage cloud



Future Work

- OTA firmware updates from iOS app
- Data visualization improvements (i.e. interactions with the graph, customized views)



Timeline/Milestones

11/2 - MVP	12/7 - Final Deliverables
• Account creation / login • Transmit refined BT data • Offline buffer • Data visualization in app • App dashboard	• Finish documentation • App integration • Cloud implementation • Final UI consistent with Web

Objective Key Results

Data Collection and App to Cloud Pipeline 96%	Data Processing and Visualization 100%	Cloud Integration 100%
• Transmits sensor data from device to iOS app to cloud • Future work required for OTA support	• Refine BT sensor data • Data visualization in the app via graphs • Pull historical data into app from cloud	• Store user data in Firebase • Admin account to view all users

Ali Roghani
ThermaSENSE
Founder and
Contact Person

References

- Core Bluetooth: https://developer.apple.com/documentation/corebluetooth
- OTA Update: https://www.arduino.cc/en/Tutorial/ota-getting-started
- AWS SimpleDB: https://aws.amazon.com/simpledb
- Swift Charts: https://developer.apple.com/documentation/charts

Dr. Edward Fox
CS 4624 Advisor