Affordable in home internet of things temperature sensor

Josiah Craw
35046080

October 14, 2019

Abstract

Contents

1 Introduction

The project is the design of a internet of things (IoT) temperature sensor. The intent of this product is to enable home renters to collect accurate temperature data as evidence that a home may be too cold in the winter. The solution must be low cost, have enough storage for a month of data collections and must be small and easy to hide in a home. This

2 Requirements

The main requirements for this project are that the design must be low cost and last between two and four weeks for both data collection and battery usage. The device should be able to connect to the internet, allowing the data to be accessed remotely as well as allowing the sensors to offload data to external storage.

2.1 RF Connectivity

The system should include some form of internet connection interface to allow the data to be put into a cloud storage system. A form of wireless communication between the modules and a phone, this could either be implemented with all modules communicating to each other and having a master node that connects to the users device or with every node connecting to the users device separately.

2.1.1 Zigbee vs WiFi vs Bluetooth Low Energy

The main competitors in the RF connectivity space is Zigbee, WiFi, and Bluetooth low energy. ZigBee is based on the IEEE 802.15.2 standard, this technology allows low power mesh network transmission at 2.4GHz frequency. This connectivity system is widely used in the IoT industry with standards such as light link being used to connect smart lighting products by manufactures like Phillips and Xiaomi. The next technology is WiFi the advantages of WiFi is the ability to use it to seamlessly access the wider internet, however the system is complicated as each unit would have to be completely independent, with each sensor in the network connecting directly to the users private network. WiFi is also uses a significant amount of power. Finally Bluetooth Low Energy (BLE) is power efficent, and can also connect directly to a users phone or computer with Bluetooth compatability.

2.2 Data Collection

Data collection is an important part of the sensor as the hardware must be able to store between two and four weeks to meet the specifications. All data collected will need to contain the sensor data, as well as a time component, this time component will most likely have to be sourced from a real time clock (RTC) as the specifications require operation even when internet is unavailable. This amount of data will most likely require some external storage in the form of EEPROM, electronically erasable read only memory. This data when offloaded into the cloud will need to be kept secure to avoid user data breaches. Different database and storage solutions will need to be considered for the cloud storage aspect of the solution.

2.3 Power Use

For this project the power use is very important as the system should be able to seamlessly record data for a complete two to four week period. This means that keeping power consumption is important in the success of the design.

3 Hardware

The chosen final specifications for the design were chosen to meet the requirements, focusing on the simple usability of the design as well as keeping the power consumption low enough so the device may run uninterrupted.

3.1 Processor

The chosen processor is from ST Microelectronics, specifically the STM32WB55CEU6. This MCU was chosen as it integrates both ZigBee and BLE support on one chip. This has two benefits, the first is cost as the unit integrates both the MCU and the transmit/receive functionality the cost is reduced as only one relativity affordable chip is required, with the chip costing $6.65 for a single unit and $3.76 each in 1,000 quantity all in USD.

This chip also has advantages for power consumption, as the package consists of two separate Arm cortex processors. The first is a Cortex M4 this is dedicated to running the application and is a very low energy part. The second is a Cortex M0+ this is dedicated to running the communication stack. The network stack in this processor is capable of calling interrupts, this means that the the low power M4 processor can be put into deep sleep mode to conserve power and can be woken by a central node over the network to process the current temperature then send it over the network before returning to deep sleep.

3.2 Temperature Sensor

3.3 Power Delivery

3.4 Modules

The design is to be modular to keep the production costs low as the entire system will have a low initial production run of 5,000 units and to keep the PCB and component assembly cost as low as possible each the main unit containing the STM MCU and the temperature will feature a six pin connector with Voltage in, voltage out, ground and USART as well as one GPIO pins for future expansion. These pins will interface with external battery pack or base station modules.

3.4.1 Data Collection Board

The data collection board is design to contain the STM processor as well as the temperature sensor. Alongside these two the board contains supporting peripherals such as power delivery components and the PCB antenna, with the required components. The PCB antenna is used to keep the unit affordable as extra costly external components are not required.

3.4.2 Battery Expansion Board

The battery board contains holder for two 18650 size lithium ion batteries, these batteries contain enough power to keep the data collection board operating for 4 weeks. The rest of the spare GPIO is routed out to five pins, with the GPIO, two USART and voltage out and the ground. This will allow the system it potentially allow other sensors to be added to the unit.

3.4.3 Base Station Expansion Board

This module is much more complicated than the battery board, this board uses a DC barrel jack to power it from the main supply. This board also contains an ESP32 module this module is integrated for the base station to allow WiFi connectivity. There is also EEPROM to allow for long term data storage on the board. The ESP32 is connected to the USART connections of the STM micro controller, in this configuration the STM will be running in a slave like manner, with is being used primarily as a transmitter for into the ZigBee network. The ESP32 will be running Amazon FreeRTOS this is based on the popular FreeRTOS platform.

3.5 Base Station

4 Firmware

4.1 Node

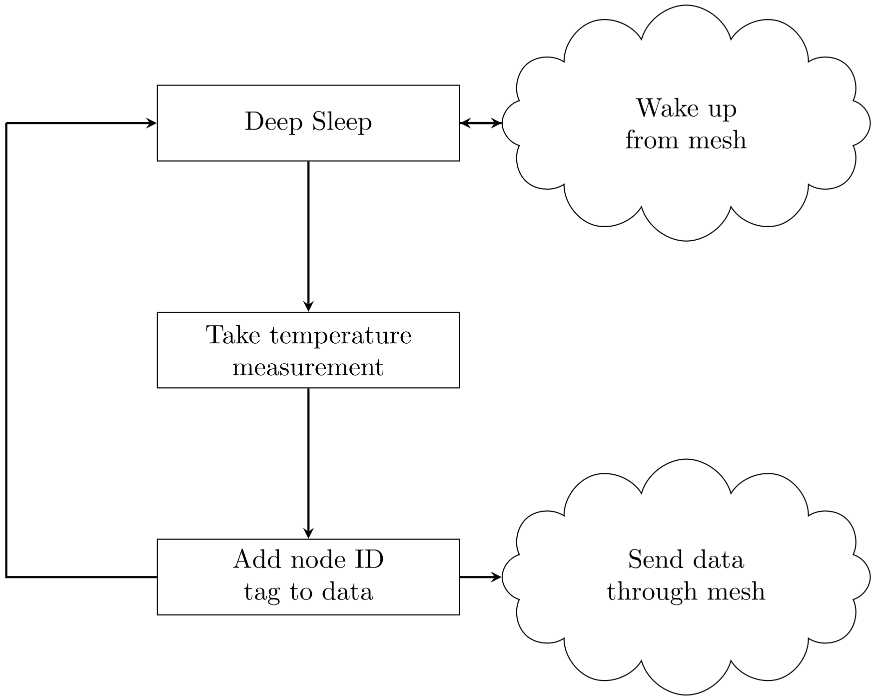

Figure 1: Individual node firmware block diagram

4.2 Base Station

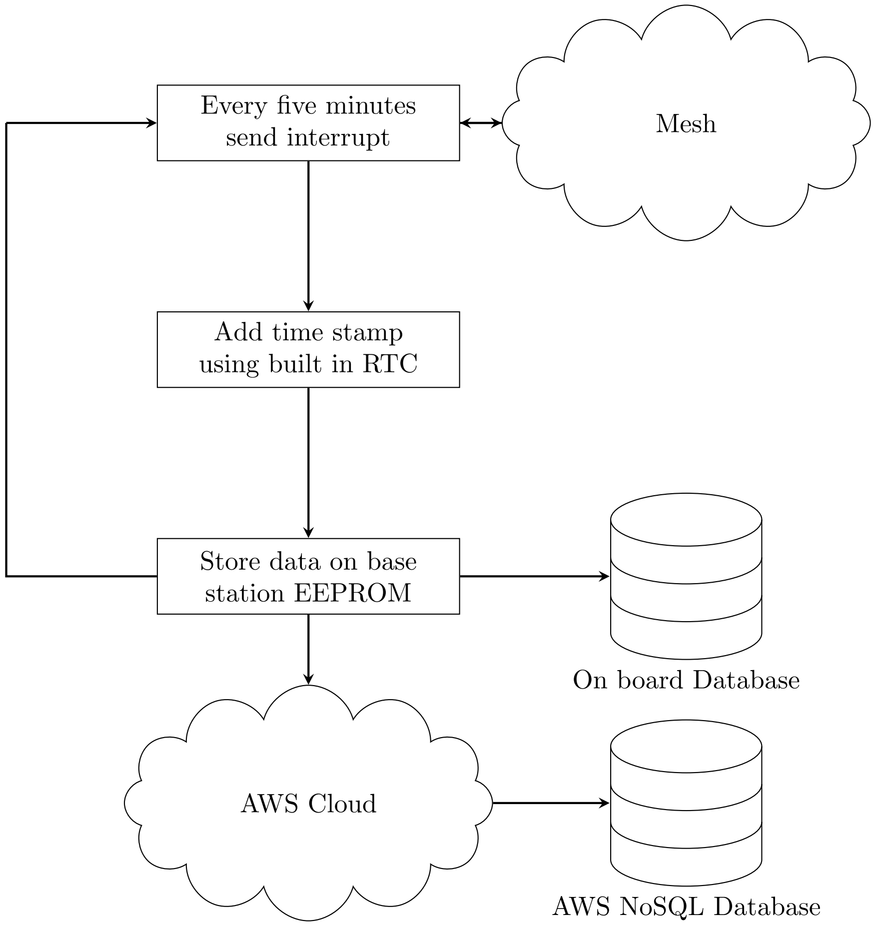

Figure 2: Base station firmware block diagram

5 Cloud System

The cloud system key component of the IoT platform and must be implemented securely, and seamlessly to aid in ease of use, the system must also meet the cost constraints as to ensure the product is competitive in price while still being able to offer a complete experience to the user. For these reasons the Amazon Web Services (AWS) platform has been chosen for the cloud platform due to the cost as AWS operates on a 'pay as you go' model with competitive prices allowing the project to require no investment in compute hardware infrastructure.

5.1 Amazon FreeRTOS

One of the key features provided by AWS for this project is the Amazon FreeRTOS. Amazon FreeRTOS is a real time operating system (RTOS) based on the open source FreeRTOS project. This software runs natively on the ESP32 used in the base station and allows the system to connect to the internet simply and securely. The main features of this is the secure connections to the AWS platform, using the MQTT platform for secure messaging to communicate with the AWS IoTCore product. This is provided in the AWS platform to manage and control devices. Another major feature for the system is the built in API for over the air (OTA) firmware updates. This is a powerful feature as it is very important for IoT devices to be able to be upgraded and features and security patches are essential to maintain a competitive IoT product.

5.2 Data Management and Security

The next key feature of AWS is the data management as using the AWS global platform allows the product to scale beyond one country with out increased cost and infrastructure upgrades. The security is also an important feature of the platform as maintaining data security can be expensive and as this is all contained within the cost of AWS this further reduces the cost of developing an in house system. The global system also allows the data to be made redundant across several data centres ensure potential customers data is not lost. All of the aspects combined show make the AWS platform a good option for the project as it significantly reduces cost compared to developing and maintaining the software and hardware support in house.

5.3 Over the Air updates (OTA)

5.4 Data Analysis and Big data applications

Finally, AWS provides cost effective compute systems, this means that the collected users data can be efficiently processed and evaluated to provide insights into the user and may be informative for the user such as a heat map for their house and a simple animation demonstrating the temperature of their house and rooms over the course of a day or month. This information may assist users in finding the issue in their homes thermal management eg. If one room in the home was getting cold with all the other room cooling after, the first room is most likely the culprit for the thermal issues in the home.

The computed data provided by many customers in a area could provide a effective data set for many applications such as potential power consumption statistics and room occupancy during different time based on the temperature within the room. This data could be analyzed in cloud to provide insights into the habits of customers. However despite the potential research and informational value of the data, user privacy and security would have to be maintained. This could be done by stripping any identifying information for the data before it is computed and added to a larger set of users data. This could be achieved in the cloud by extracting the users general data from their storage and processing it automatically before processing it in the main data processing system.

6 Future Expansion

Due to both the hardware modularity and software modularity with over the air updates, the design can be expanded and upgraded simply thorough either pushing updates to the software and firmware or providing add in modules. This allows the system to provide continuing support for features, keeping the users in the platform and potentially funding the ongoing costs of the cloud systems.

6.1 Humidity Sensors

A future module could be installed using the spare GPIO available on the data collection board expansion connector.

6.2 System API and existing system integration

A major feature for future implementation would be an API, this API could be used both on the external internet and on the internal users network. The API on the external network could be added using the AWS API Gateway. This would allow external API calls from systems that users use on the internet, allowing further development by other developers for the platform. Features the increase the value to consumers could be added, making this a valuable feature to implement. The internal API could be designed to run off of the ESP32 allowing the users other smart home products to interface with the system eg. A smart heating solution could interface with the product allowing the user to set heating thresholds and have the smart heater turn on when those limits are reached.