Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, BC V5A 1S6

Re: ENSC 440 Design Specifications for EasyHome

Dear Dr. Rawicz,

In regards to the course requirement of ENSC 405W/440, please find attached the document outlining the functional specifications for EasyHome, a device for visually challenged individuals to control electronic/electrical devices at home.

This design specification document aims to lay out the functions that our product and each subcomponent must perform from proof of concept to the production stage. The high priority elements of the product have been drafted and the scope of the system's functionality has been defined. This document includes a detailed system overview, system design and test cases. The requirements established in this document will shape the development and design phase of the EasyHome system.

If you would like more information or have any questions regarding our proposal, you can contact me at 604-600-8496 or via e-mail at kamalk@sfu.ca.

Sincerely,

Kamal Kaur
Chief Executive Officer
StratOs Technologies

Enclosure: Design Specifications for EasyHome by StratOs Technologies

StratOs Technologies

Design Specifications-EasyHome

Team Members

Jayco Rimpillo

Jyotsna Jaswal

Kamal Kaur

Khalid Qahwash

Muhammad Bin Khalid

Submitted To

Mr. Steve Whitmore

Dr. Andrew Rawicz

School of Engineering Science

Simon Fraser University

Submitted On

March 30, 2017

Abstract

The purpose of this document is to specify the design requirements of EasyHome, a product to help visually disabled individuals in facing the everyday challenges at home. EasyHome will allow a visually disabled user to monitor and control home appliances including kitchen stove-top, door locks and body temperature measuring device from their phone.

EasyHome consists of a hub that will control all the devices, a user interface, internet cloud and peripheral devices. The user will be able to check the status of appliances with voice input. Through API commands, the hub queries the data from the switches and sensors, and format that information appropriately for use on the Internet cloud. Since the ZigBee standard is the one of the most common technologies used in wireless automation, we will use ZigBee as the primary wireless network for our smart devices.

Smart sensors and switches will be embedded into existing ordinary appliances to sense the state of the device. A first peripheral device designed as a part of EasyHome solution is an attachment to normal stovetop. This device will monitor temperature of the burners and deliver a voice notification when the user forgets to turn off the burner. The second peripheral device will be a system for detecting the status of door lock. The user will use the phone or web application to monitor and control door locking. EasyHome also includes a smart device to measure body temperature using audio input and output as well as mechanism for identifying colours of clothes. User Interface will also help user in detecting colours of objects in addition to controlling EasyHome peripheral devices.

Table of Contents

List of Figures

List of Tables

Glossary

API	Application Program Interface.
AWS	Amazon Web Services. An internet cloud service
Cloud	A way of sharing computing resources on the internet
EUI	Extended Unique Identifier
Hub	Primary gateway between the smart devices and the cloud
I2C	A bus for serial communication between devices
IR Sensor	A sensor that uses infrared to measure temperature
LED	A light emitting device that can be used to indicate events
OS	Operating system, manages all processes running on a device
UART	Universal asynchronous receiver/transmitter, a serial interface
USB	A popular standard that defines an interface between devices
WI-FI	A wireless communication standard, also called IEEE 802.11
ZigBee	A wireless communication specification based on IEEE 802.15.4

1. Background

Home automation has been present ever since the industrial revolution of the 1920's which brought up products such as the vacuum cleaner, and the washing machine. It was not until the early 2000's that 'smart' homes became feasible from a technological and economical point of view [1]. With the ever-expanding number of home automation solutions provided in the market today, one would ask why StratOs Technologies? The answer is simple, we aim at making smart homes a revolution for visually impaired people to address the issues faced by them at home and to make independent living easier for them. A revolution that would erase the hassle that was once encountered by blind people in determining what is the temperature of the oven, what stove top was turned on, checking if the lights are on or not, and even obtaining feedback about their body temperature.

2. Introduction

EasyHome is the product line developed by StratOs technologies which serves the purpose of enhancing the lives of visually impaired people at home. EasyHome consists of a series of component devices which obtains data, processes it, and transmit it across the network to our Hub and back to the user via our User Interface (UI).

Sensing is carried out to obtain and collect data from the component side to the microcontroller (Arduino) which further processes the data and sends it to the Xbee module which will send it over to the hub's Xbee module where further information is propagated across the platform to the cloud and finally to user's phone. This design specification document will explain in detail how our platform is implemented for each of our three applications: The Stove Top Detector, The Door Lock, and the smart Thermometer.

The three applications were chosen by our team as a starting point to provide aural feedback and control to the most essential home appliances where visually impaired people needed the most assistance with. We would then expand our techniques and the EasyHome product line to cover more home appliances in the future.

Figure 1 below presents the updated schematic of the latest version of our EasyHome product line.

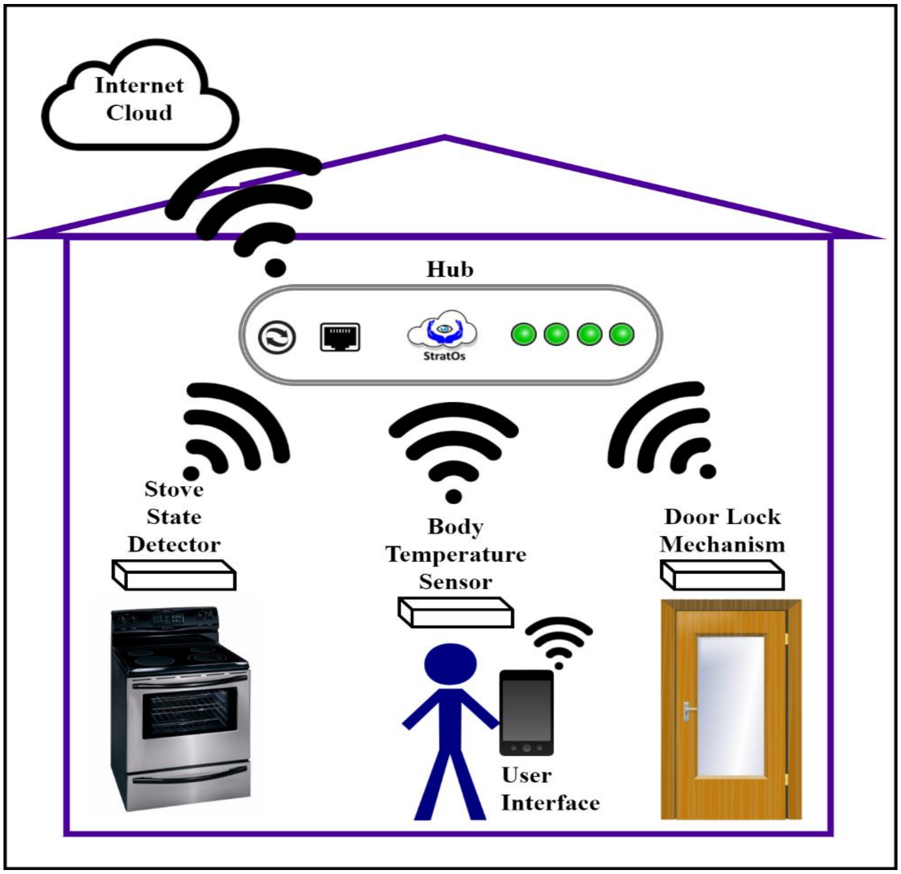

Figure 1: System Overview

As seen in the above figure, each appliance has its own sensor where 802.15.4 connectivity has been embedded upon by our engineering team using XBee modules embedded into the Arduino microcontroller from the component side and an Xbee module on the Raspberry Pi from the Hub's side. The cloud provides EasyHome connectivity to the internet using AWS cloud services. The User Interface would be designed to target blind people such that it provides on-demand feedback, monitoring and control of every single supported home appliance on the spot with minimal physical interaction. Finally, the hub enables communication between the appliances, the cloud and the UI using Raspberry

Pi microprocessor and Xbee modules which transmit and receive data from/to the Hub.

EasyHome is composed of multiple features which are run by sensors such as thermostats/thermocouples, magnetic open/close sensors, motion sensors, and other sensors in user's phone.

3. System Overview

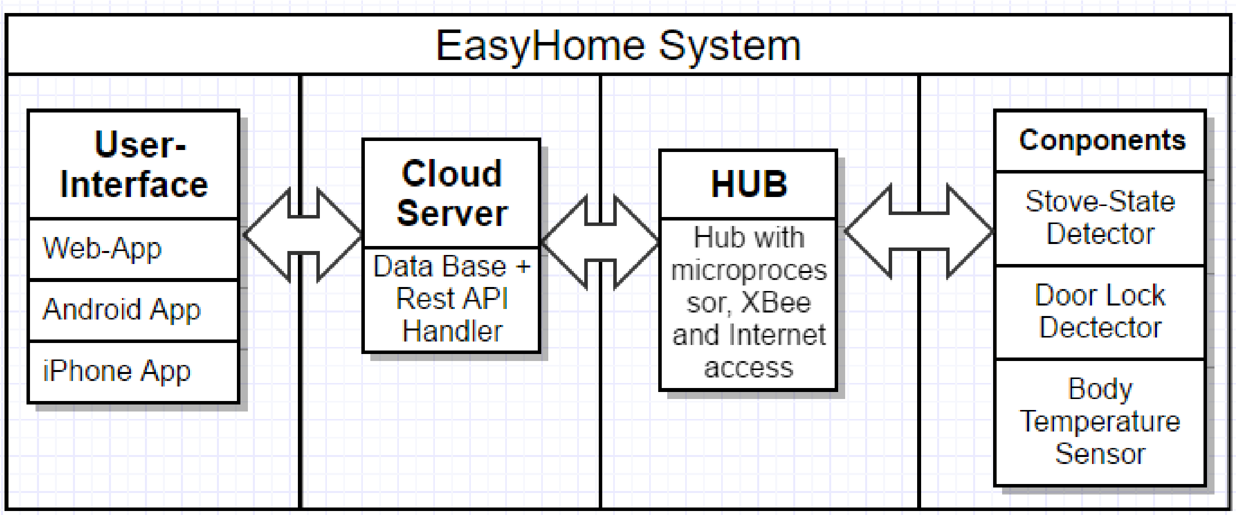

Figure 2: Communication Context Model of EasyHome System

Fig 2 describes the EasyHome system. EasyHome will have four different components that is User-Interface, Cloud Server, Hub and End-Components. They will all share two-way communication as shown in the above context model. In the following sections, the design of each system component has been described technically with process details.

3.1. Hub

The hub has two primary hardware, Microprocessor and the ZigBee Dongle. The main reasons behind choosing these components are justified as follows:

1. **Microprocessor:** in our system we required a processor that could interact with whole system and hold logic states at the same time. Hence, to handle the appropriate tasks the microprocessor was the basic component of need in the Hub hardware. The code architecture design plan is displayed in *Figure 3*.

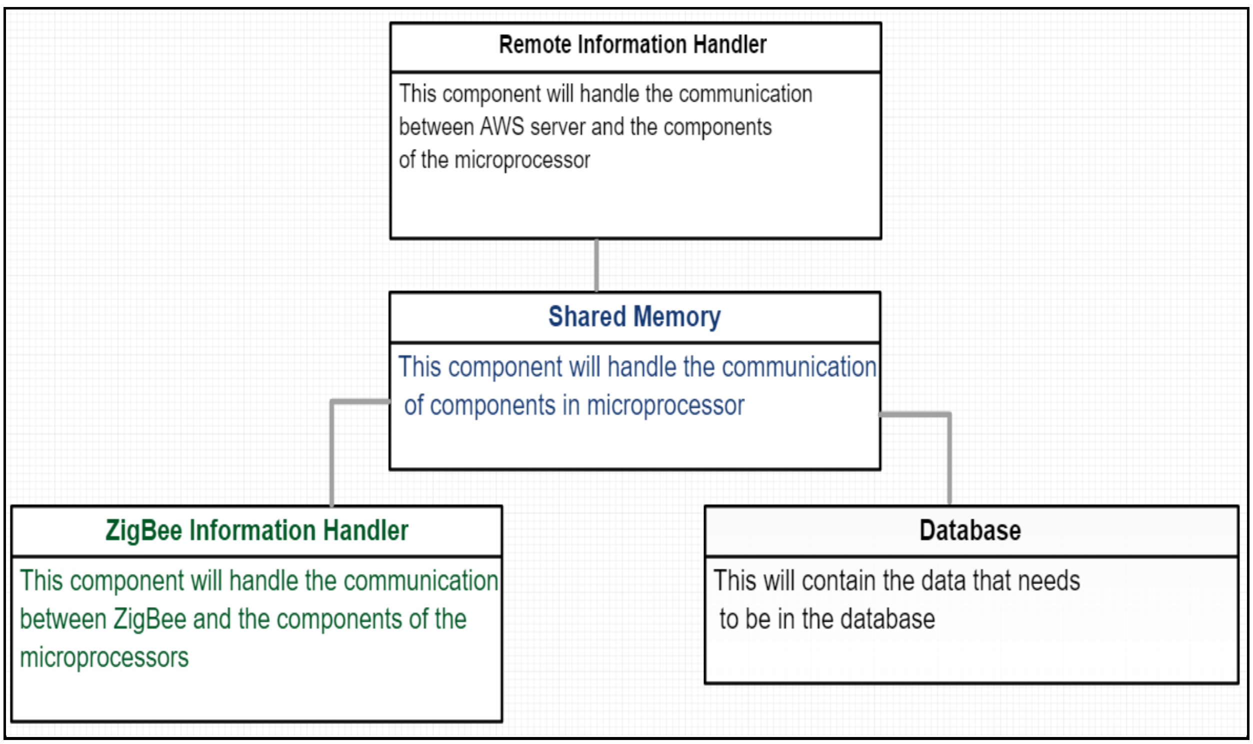

Figure 3: Context Model of Microprocessor

The Raspberry Pi 2B will be used to implement the role of the Hub. It has the required features needed to accomplish the desired tasks.

	Operating voltage:	5V
	Clock Frequecy:	900MHz
	Interfaces:	GPIOs, Serial UART, 4 USB ports
	Internet Support:	Wi-Fi Dongle, Ethernet port
	OS Support:	Unix-like

Table 1: Features of the Raspberry PI 2B

2. **Xbee:** The wireless communication is the key of our product. In today's technology, WiFi and Bluetooth communication consumes a lot of power. Hence, after doing research on the wireless modules available in the market, XBee technology seemed to fit properly in out hub as it can handle wireless communication and send that information to the microprocessor using least amount of power.

The design requirements for the Hub are described below.

3.1.1-I It shall have 4 LEDs that indicate the following:

1st LED= Power
2nd LED= Ethernet
3rd LED= ZigBee *tentative*

To indicate the state of the hub. This state indicator will be helpful in installation and troubleshooting purposes of our product. Any embedded system with state indicators are generally liked in market as it provides immediate feedback about the status of device.

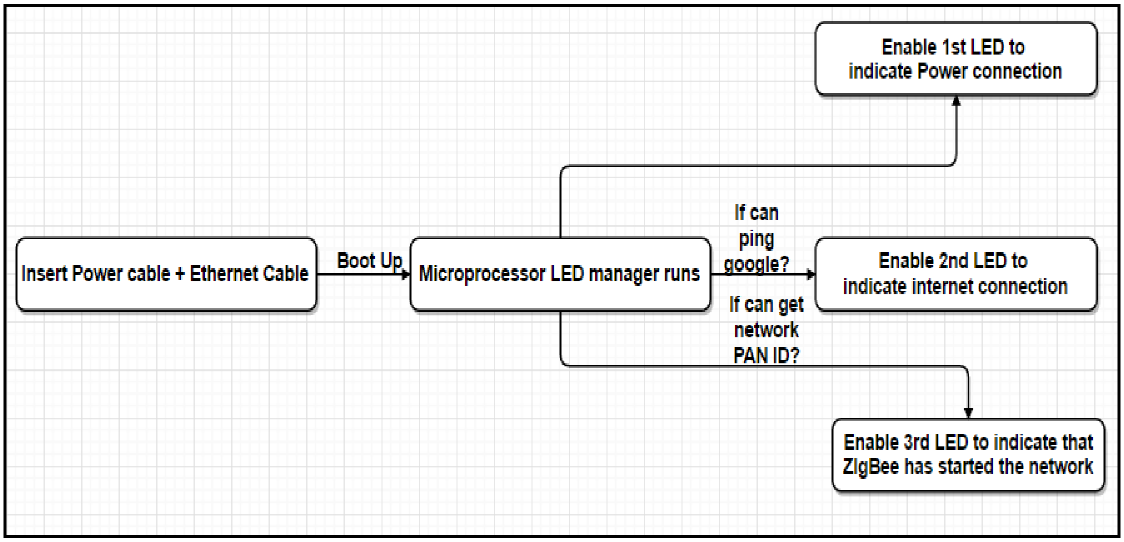

Figure 4: Process Model of LEDs to indicate Power, Internet and ZigBee Status

3.1.2-I It shall have the capability to connect to the internet and a mechanism to indicate to the technician that device has connected to internet, using 2nd LED (refer *Figure 4* for more complete flow).

3.1.3-I It shall be able to connect to at least 4 different EasyHome-components at a time to increase the user's control over the home appliances. This constraint is made to show the minimum number of EasyHome-Components that should be supported by our system. As per the cost of the product it is efficient to control at least four devices or else product will be considered more expensive with less use cases.

3.1.4-I It shall store the EUI numbers of the paired devices and they shall be deleted after de-provisioning. Provided range will be used to identify each device uniquely.

3.1.5-I It shall support a ZigBee Cluster and attribute an information exchange mechanism to support wireless communication between itself and the components. This feature has been chosen so that the integration with the Smart energy profile devices and HAN devices could be easily done as 90% of wireless market follow this criterion as well.

3.1.6-II It shall have a Cloud application that will allow the remote connection to the cloud for remote monitoring. The information that will be exchanged is listed as follows:

1. Online/Offline status of the hub. (online: when HUB can ping server).
2. The connectivity status of the components with the hub. Possible status can be Paired, not in network or Never Paired.

This monitoring information is important as it will provide more trust and visibility to our customer. More description is provided in section 3.5.

3.1.7-**II** It shall allow the local connection to User Interface for feedback. The information that will be exchanged is listed as follows:

1. Online/Offline status of the hub. (Online: when the internet application can ping the Hub).
2. The connectivity status of the components with the hub. Possible status can be Paired, not in network or Never Paired.

This use case is important for the consumers who have security issues and do not want any information to be transmitted through the cloud to avoid analogous attacks to their Internet connection.

3.1.8-**II** It shall be able to continuously monitor the components at the defined rate of the developer through configuration files (JSON). Monitoring the component information is a marketing strategy as the Government has been spending a lot of money in cutting the energy costs. If we monitor the component we can use that information and provide statistics in the future. This feature is also useful for the consumer as they can monitor their use of the component.

3.1.9-**III** It shall have a button that will be held for 5 seconds to reset the hub to default state. The pattern of LEDs will indicate if the reset was successful.

3.1.10-**III** It shall have an auto update property which updates the device through a remote update server. Every embedded system needs this feature as it provides the ability to debug the in-field problems encountered by consumers.

3.2. Stovetop Detector

The Stove State Detector has been chosen as one of the main components that can help the visually impaired. Its main purpose is to inform the user which stove top burners are turned on which makes it a useful cooking aid. It also promotes peace of mind as it can be used to check if the user accidentally left the stove on before leaving the house.

The Stove State Detector is composed of 3 main blocks: 1) The infrared (IR) temperature sensor, 2) the XBee module for Zigbee connectivity, 3) the Arduino microcontroller board.

1. **Infrared (IR) Temperature Sensor:** An Infrared (IR) temperature sensor will be used to detect the temperature of the stove. The IR sensor has been chosen instead of other types such as thermocouples and thermistors since it is the least obstructive. It senses the temperature without contacting the object which makes it safer to use in the kitchen environment. The IR sensor will be positioned above the stove and can be attached to the exhaust hood. Another advantage is that one IR sensor can cover multiple burners, while other sensors can only monitor one burner at a time.

2. **XBee Module:** The XBee module will take care of the wireless communication between the hub and the Stove Detector component. The module will handle the wireless Zigbee protocol needed to establish wireless connection to other XBee modules.

3. **Arduino Trinket Board:** The Arduino microcontroller board will act as the brain of the Stove Detector component. It gathers data from the IR sensor, processes that data and formats it into a data frame that can be sent to the XBee module for wireless transmission.

The Arduino Trinket board will be used as the microcontroller for the Stove Detector component. Its features are very similar to the Arduino Uno board but at a much smaller physical size which makes it a more favorable solution. The 3.3V variant has been chosen to allow it to run on a rechargeable battery.

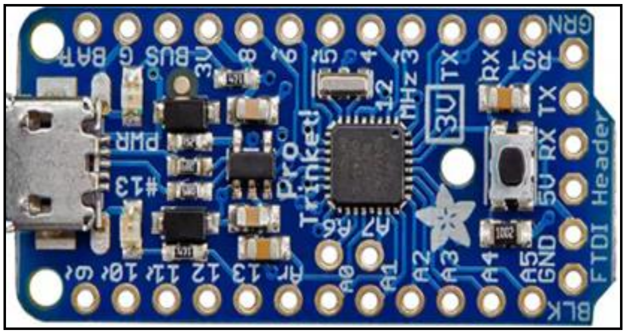

Figure 5: Arduino Trinket 3.3V 12MHz Board

The requirements for the Stove Detector and how they will be addressed in the design are explained below.

3.2.1-I It shall have the capability to maintain a secure connection to the Hub so that it can send feedback to the user.

The XBEE Series 2 module will be used to establish the Zigbee communication between the Stove Detector component and the Hub. It has been selected because there are open-source documentation and software libraries readily available online.

It will send or receive data to/from the Arduino through the serial interface pins (Tx and Rx). The hardware connections between the Arduino and the XBee are illustrated in *Figure* 6 and a basic explanation of the communication between XBee modules is shown.

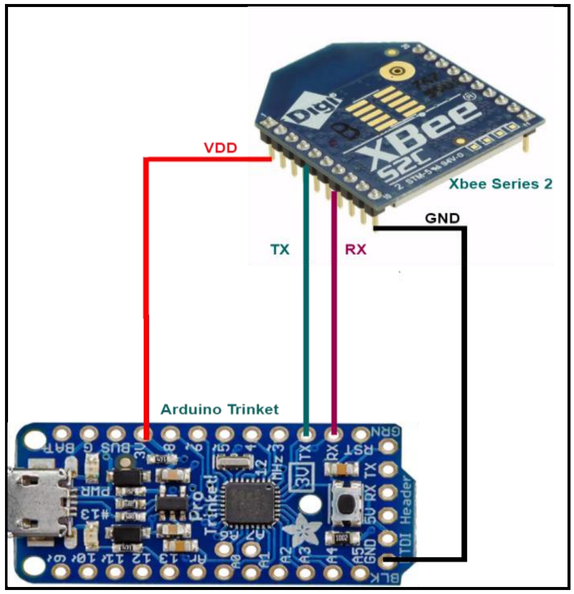

Figure 6: XBee and Arduino connections

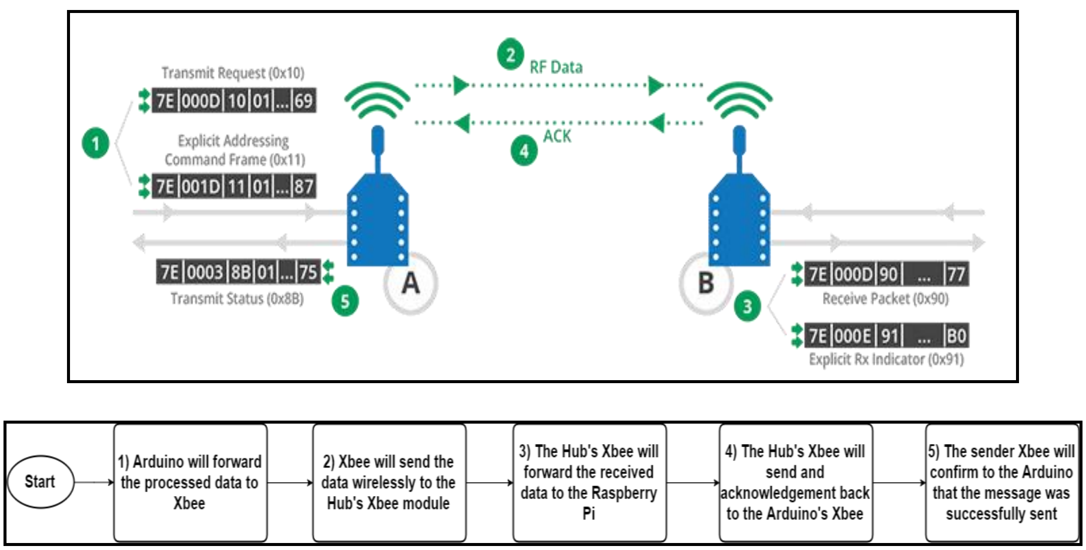

Figure 7 Xbee wireless communication process [2]

3.2.2-I It shall support a ZigBee Cluster and attribute information exchange mechanism to support wireless communication.

Although the Xbee modules can be set up such that they only work among the StratOS EasyHome Xbee products, they can also be configured to be compatible with existing Zigbee-enabled commercial products. By appropriately setting the Cluster ID, Endpoints, and Profile ID according to the Zigbee standard, it is possible to communicate with other smart products that are available in the market. *Figure 8* shows the fields that can be defined to support the ZigBee Standard.

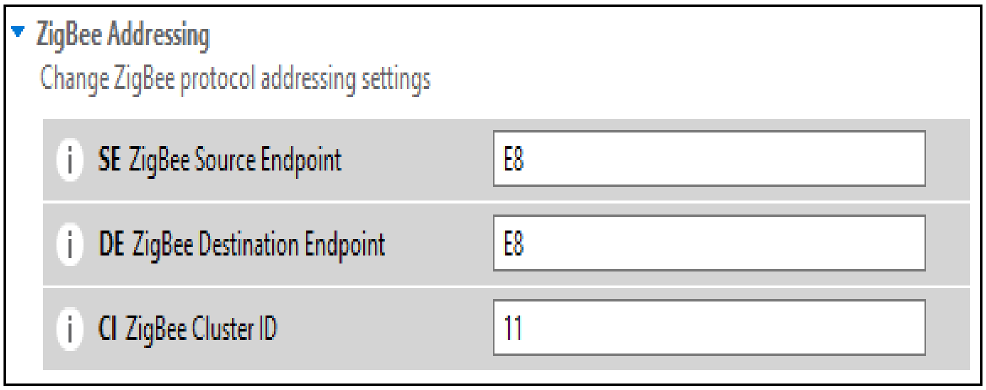

Figure 8: Xbee wireless communication process

3.2.3-II It shall have a state store mechanism that will store connectivity status so that it survives the power cycle.

3.2.4-III It shall have a connection state indication on the hardware.

To help the designers and technicians in troubleshooting the device, an LED will be used to indicate that the detector is transmitting or receiving information to the hub.

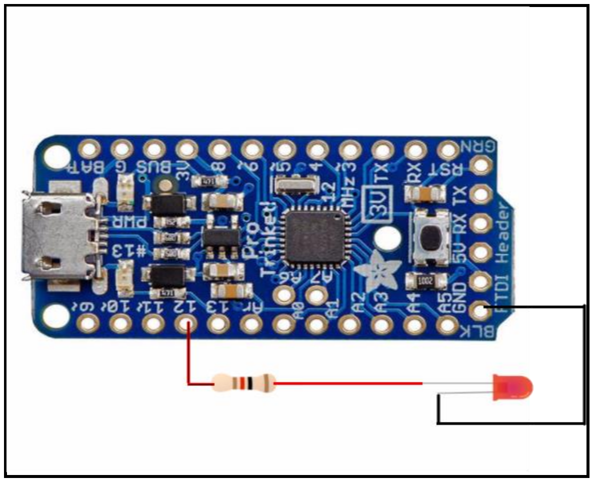

Figure 9: LED for indicating connection status

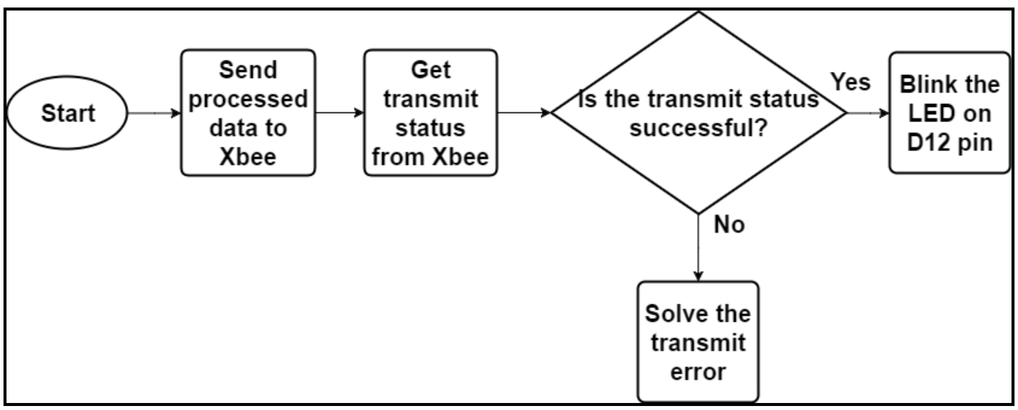

Figure 10: Algorithm for Status indication through LED

3.2.5-III It shall have the ability to precisely sense which of the stove top burners are currently turned on.

The Grid-EYE AMG88 will be used as the IR sensor for the Stove detector component. It has been chosen since it is the best solution to the problem. It has an array of sensors that can capture 8x8 pixels of thermal image which is enough resolution to distinguish between multiple burners present on the stove top. The IR sensor will be positioned above the stove and can be attached to the exhaust hood. This allows a non-contact measurement of the stove temperature which makes it safe to use in the kitchen environment. The sensor will send its data to the Arduino through I2C communication.

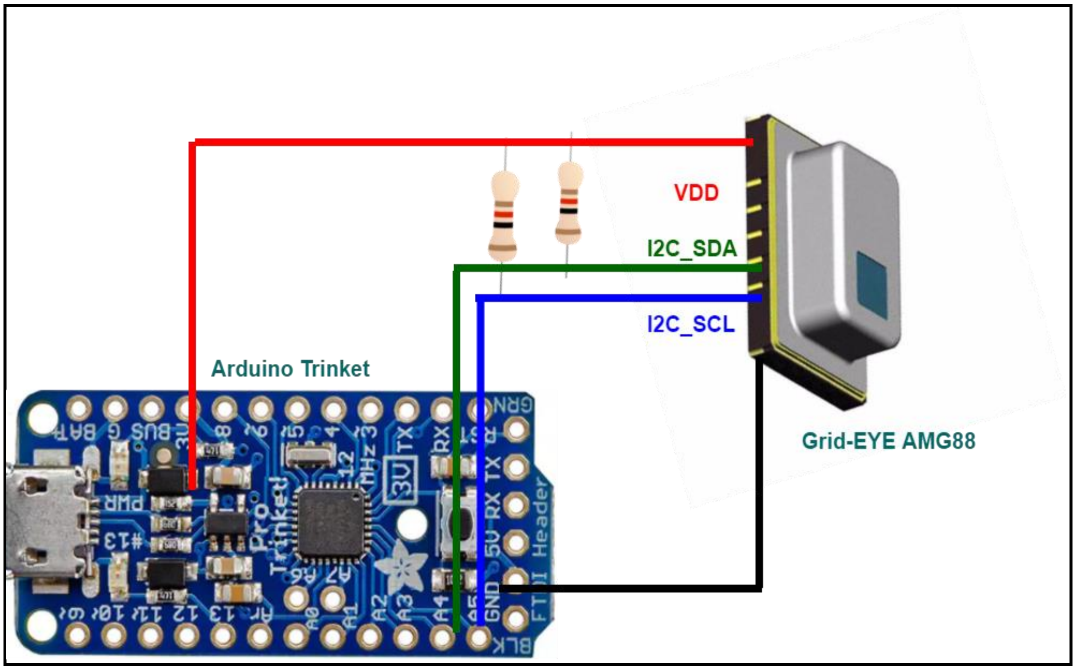

Figure 11: IR Sensor and Arduino connections

From the 8x8 pixels of temperature profile data, the location of the hot burners can be determined (Top left, top right, bottom left, bottom right). An example of its operation and the accompanied flowchart are illustrated below.

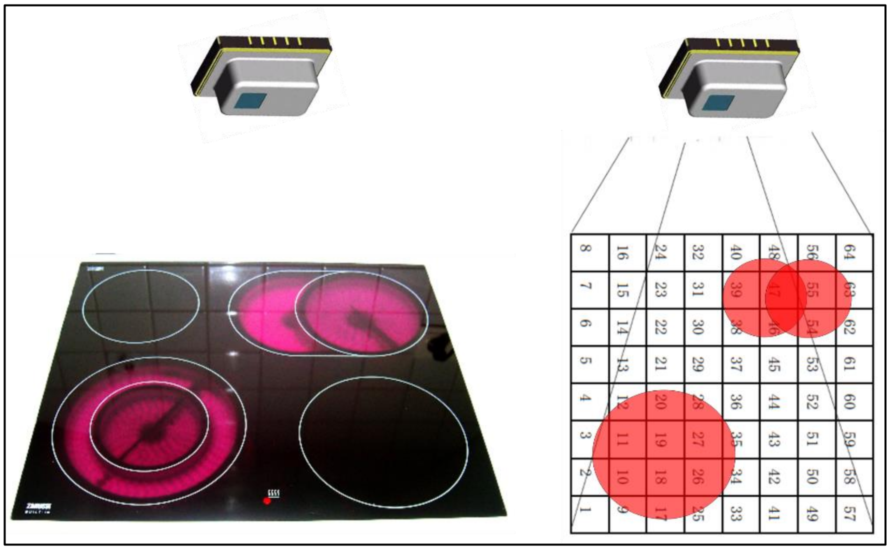

Figure 12: Example of IR Sensor temperature measurement

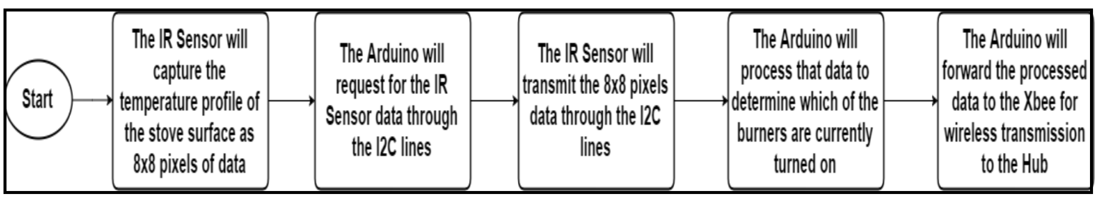

Figure 13: Algorithm for Stove surface temperature detection

3.2.6-III It will be powered using a rechargeable battery for a robust and reliable operation.

The Stove detector component will be powered using a 3.7V 1200mAh rechargeable Lithium Polymer (LiPo) battery. An MCP73831 miniboard will be attached to the Arduino Trinket to allow charging of the battery using power from the Arduino's microUSB port. Figure 14 below provides the connections between the Arduino and the battery charging circuitry.

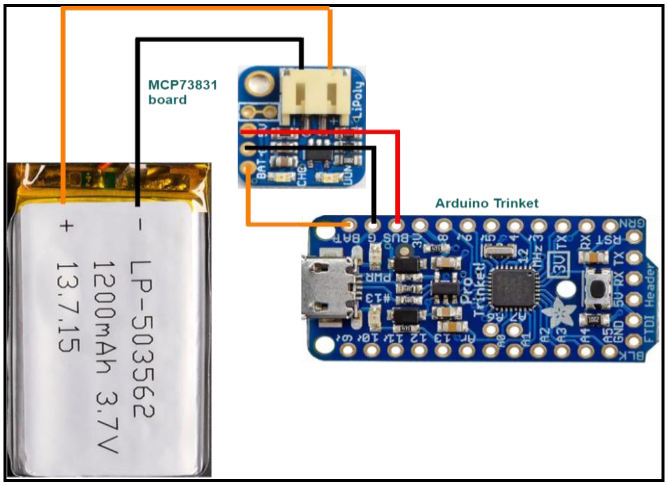

Figure 14: Battery and charging circuitry

Overall Materials for Implementing the Stove State Detector

An list of all the materials needed for the Stove Detector is presented below.

	Microcontroller Board	Zigbee Module	IR Sensor	Battery Charging Circuit	Rechargeable Battery
Device Name	Arduino Pro Trinket 3V Board	XBee Series 2	Grid-EYE AMG8834	Pro Trinket Li- Ion Battery Backpack	Lithium Ion Polymer Battery
Operating Voltage	3.3V (Accepts 5V from microUSB)	3.3V	3.3V	Input: 5V from microUSB Output: Up to 4.2V	3.7V to 4.2V
Max Current Draw (or Supply)	150mA regulator limit	45mA	4.5mA	(100mA charging current)	(1200mAh capacity)
Operating Frequency	12 MHz (Clock)	2.4 MHz (Wireless)	400kHz (I2C)		
Wired Communication Interface	UART, I2C, microUSB	Serial UART	I2C		

Table 2: List of Materials for the Stove State Detector

Overall Design for the Stove State Detector:

The overall system design for the Stove State Detector component is shown below. Note that only the important signals are present in the figure and not all connections are being shown.

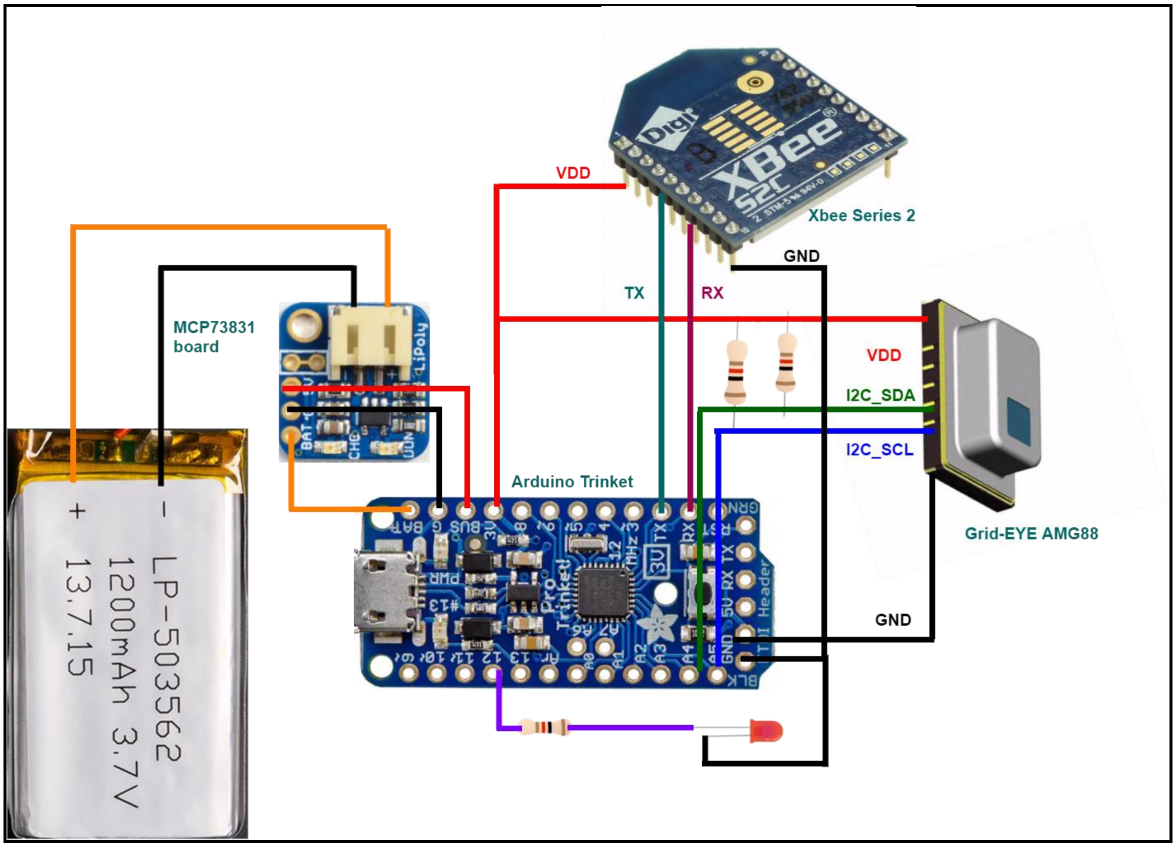

Figure 15: Overall Design for the Stove State Detector Component

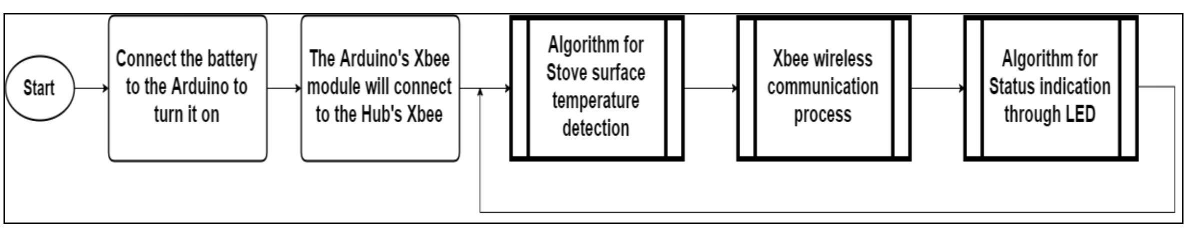

Figure 16: Overall high-level algorithm for the Stove State Detector Component

The overall high-level software solution that will be implemented on the Arduino is indicated above. Refer to the requirements in the previous pages for a more detailed explanation of each of the bolded blocks.

3.3. Door Lock Mechanism

3.3.1-**I** It shall be able to lock and unlock by utilizing our voice activated application.

After the command is sent to the UI, it is sent to the Hub which will send data to the Arduino where it is processed as lock/ unlock depending on the command. The Arduino will control the relay switch which control the magnetic door lock i.e. turn it on or off. The Arduino will confirm the execution and send an update to the XBee on the Hub which will send an update of the door lock's status on the UI.

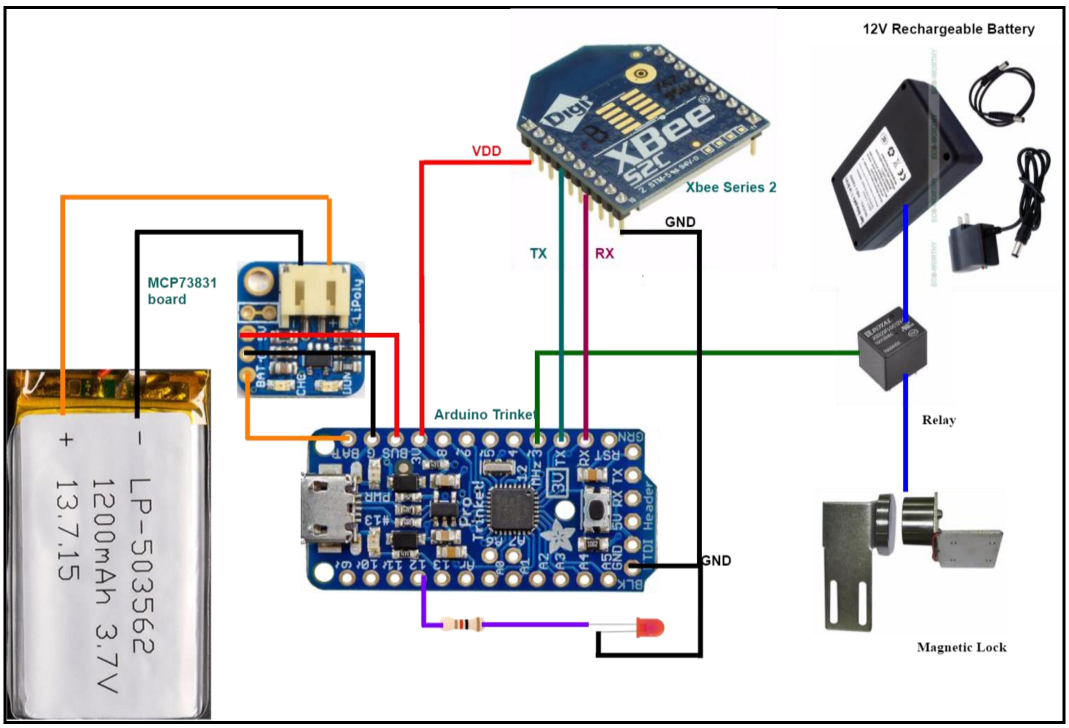

Figure 17: Overall Design for Door Lock Mechanism

3.3.2-**I** It shall have the capability to maintain a secure connection to the Hub. Refer to Section 3.2.1 for a more detailed explanation.

3.3.3-**II** It shall have the capability to send the status of the door to user. State of Relay will indicate if the door is locked electronically or not. The door lock's status update is sent to the UI in the same procedure stated in section 3.3.1 above.

3.3.4-**II** It shall have three LED lights for the following

1st LED= Locked

2nd LED= Zigbee

To indicate the status of hardware.

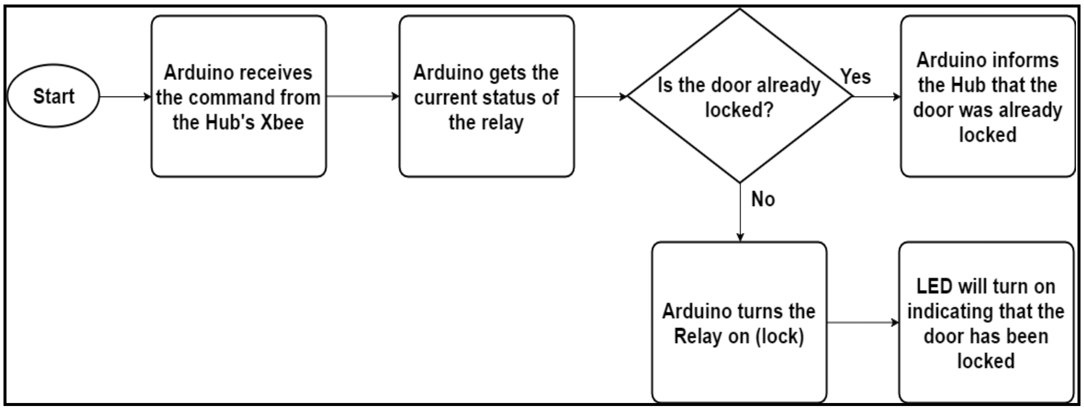

Figure 18: Algorithm for indicating door lock status

3.3.5-**III** It shall have a rechargeable battery as an alternative power source. Refer to Section 3.2.7 for a more detailed explanation.

3.3.6-III The Magnetic lock will have a rechargeable 12V battery to use when there is power outage for security.

Refer to Figure 17 for an illustration of how it is connected. The 12V battery will have a dedicated charger so that it will always have sufficient charge whenever a power outage occurs.

3.4. Body Temperature Sensor

Its main purpose is to read the temperature of human skin and inform the user via audio feedback through the phone. The body temperature sensor is composed of 3 main blocks: 1) temperature sensing Integrated circuit chip LM35 2) the Xbee module for Zigbee connectivity, 3) the Arduino UNO 3.3 V microcontroller board.

The LM35 IC sensor has been chosen instead of other types such as thermocouples and thermistors since it is most accurate. It senses the temperature of the surface it is in contact with an accuracy of 99% for temperatures between 40 -110 degree Celsius. The sensor will be cased inside a probe which can be touched under armpit for temperature sensing.

The Xbee module will take care of the wireless communication between the hub and the Body temperature sensor. component. The module will handle the wireless Zigbee protocol needed to establish wireless connection to other Xbee modules.

The Arduino microcontroller board acts as the brain of the Body temperature sensor. Arduino UNO has been chosen due to its low cost. It gathers data from the IC sensor, processes that data and formats it into a data frame that can be sent to the Xbee module for wireless transmission.

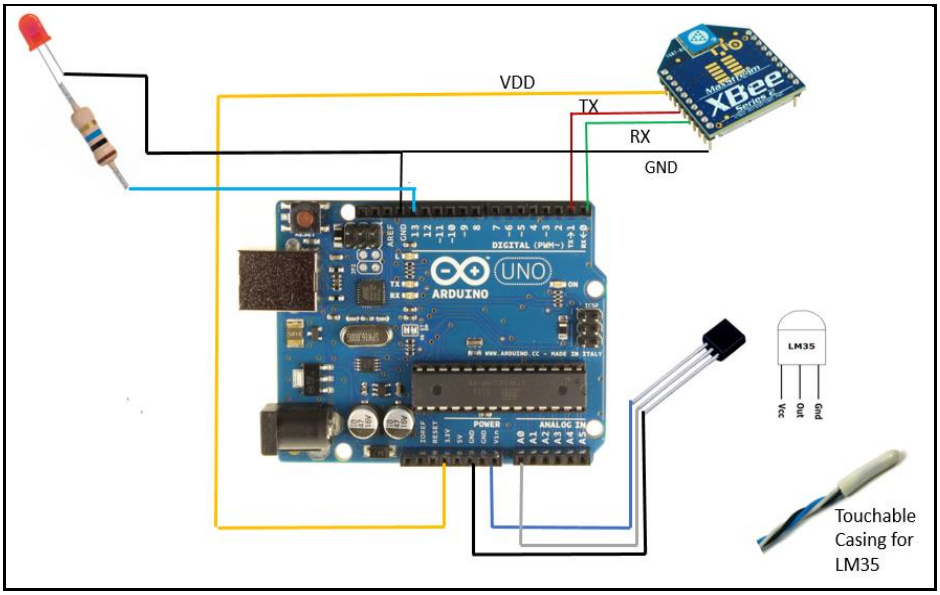

Figure 19: Overall Design for Body Temperature Sensor

The requirements for the Body temperature sensor and how they will be addressed in the design are explained below.

3.4.1-**I** It shall have the capability to maintain a secure connection to the Hub so that it can send feedback to the user.

Similar to Section 3.2.1

The XBEE Series 2 module will be used to establish the Zigbee communication between the body temperature sensor component and the Hub. It has been selected because it is consuming very low power, is an industry standard and there are open-source documentation and software libraries readily available online.

It will send or receive data to/from the Arduino through the serial interface pins (Tx and Rx).

3.4.2-**I** Onboard LEDs will provide connection state indication on the hardware. To help the designers and technicians in troubleshooting the device, an LED will be used to indicate that the detector is transmitting or receiving information to the hub.

Same as Section 3.2.2

3.4.3-**I** It shall support a XBee Cluster and attribute information exchange mechanism to support wireless communication.

3.4.4-**I** It shall have the capability to receive a request for temperature from the hub. Arduino Microcontroller will receive request via Xbee and read data from sensor.

3.4.5-**I** It shall have the capability to send the status information to the hub. Microcontroller will send the generated data to Xbee. This data will be sent to hub for further communication.

3.4.6-**I** It shall have the capability to refresh the data when the temperature of the body changes. The Microcontroller will keep on reading and reporting data until the application is still open on users phone with body temperature sensor screen open.

3.4.7-**II** It shall have the capability to detect temperature with more than 95% accuracy since the LM35 temperature sensors used for temperature detection have an accuracy of 99% for 40- 110 degree Celsius.

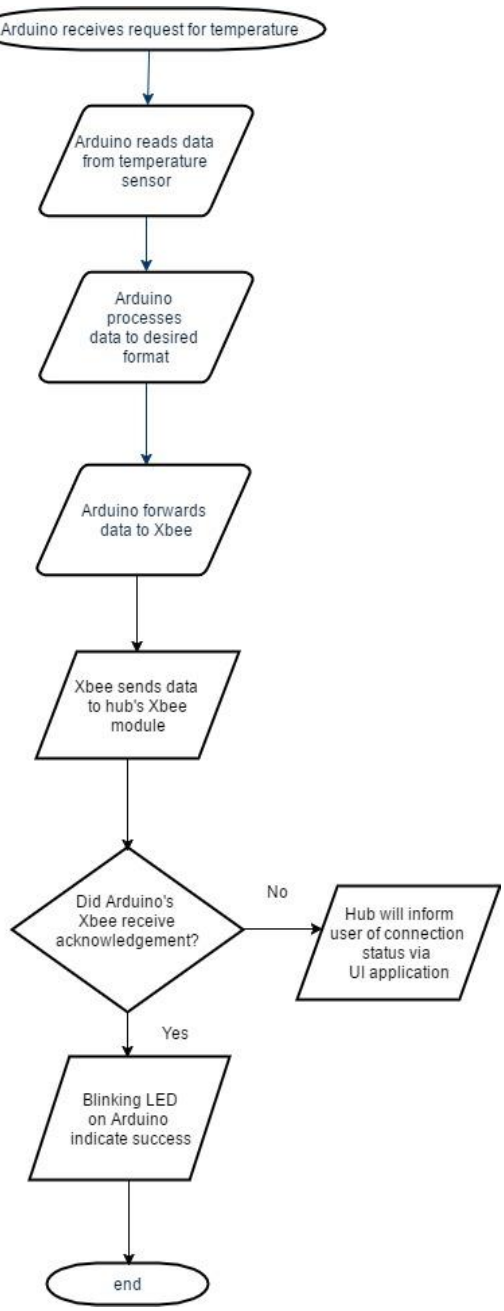

Figure 20: Algorithm for Body Temperature Sensing

3.5. Cloud Server

To provide the user ability to access and control the kitchen stove top and door lock remotely an Amazon AWS web server will be used. We chose Amazon AWS cloud server over other web servers because of its readily available feature IOT and Alexa Voice. AWS IOT and Alexa can be easily integrated with our product providing StratOs a choice of building upon AWS IoT instead of setting up cloud server from basics and integrating with 3rd party voice assistants.

Figure 21 displays that web server will be integrated with MongoDB database to store device, user and sub device information. MongoDb was chosen over relational databases because of its flexibility and faster query access. The server will have Node.js scripts running on it. It will also be exposed to REST APIs so that it can accept requests from User's phone, forwarding the requests to hub and handling communication from hub to user. Refer Figure 22 for detailed flow of information from UI and Hub to Web server.

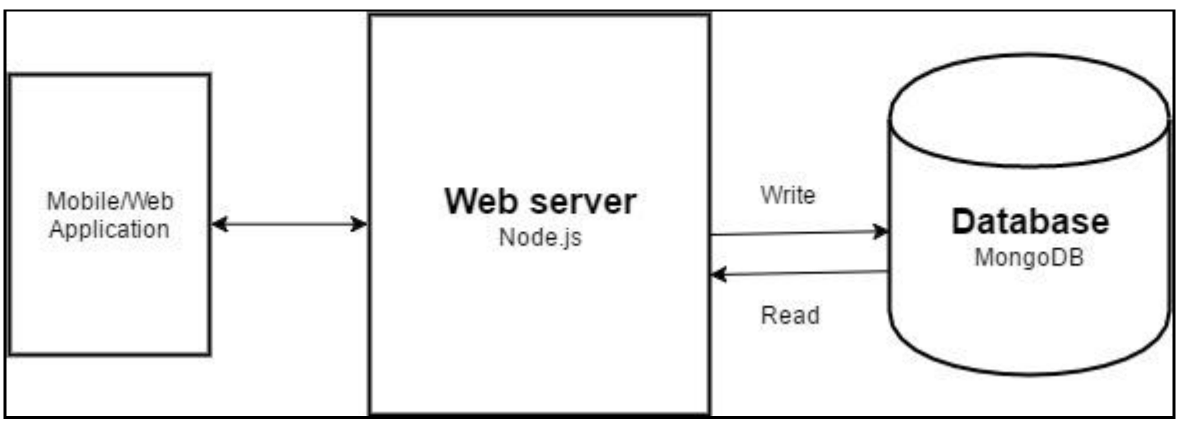

Figure 20: System Model of Cloud Server

3.5.1-I It shall authenticate with the hub using unique Install code on the device. The encryption certificate shall be saved in the hub memory and will serve for authentication every time a communication is attempted.

3.5.2-I It shall be integrated with MongoDB database and the cloud server should have the capability to store the user data. MongoDB's native Node.js driver shall be used to create the connection with the MongoDB server.

3.5.3-**I** It shall be able to store the user information, including: user email, password, address, phone number and the ID of the device purchased. insertOne method and the insertMany method shall be used to add documents to a collection in MongoDB. find method to issue a query to retrieve data from a collection in MongoDB.

3.5.4-**I** Several API calls shall be exposed to access the information from the server. Few examples are listed below:

- GET /user: It shall provide the list of the users in our database
- POST /user: It shall add new user to the system
- PUT /user: It shall modify the information of the existing user in the database
- DELETE /user: It shall delete the user from the database

More API will be introduced in the process of making data parsing easy to support the system integration with the User Interface.

3.5.5-**I** It shall run on AWS server.

3.5.6-**I** It shall be up and running for at least 95% of the year to provide users high availability and more reliability. This shall be achieved by having a failover server running at all times to serve as backup.

3.5.7-**II** It shall be able to store the history of a device up to a year. Device history will be saved on a separate collection on MongoDB and shall be accessible within the UI.

3.5.8-**III** It shall have a private key that will allow other users to authenticate. Any request from an IP without the clouds private key will be ignored and blocked from further attacks.

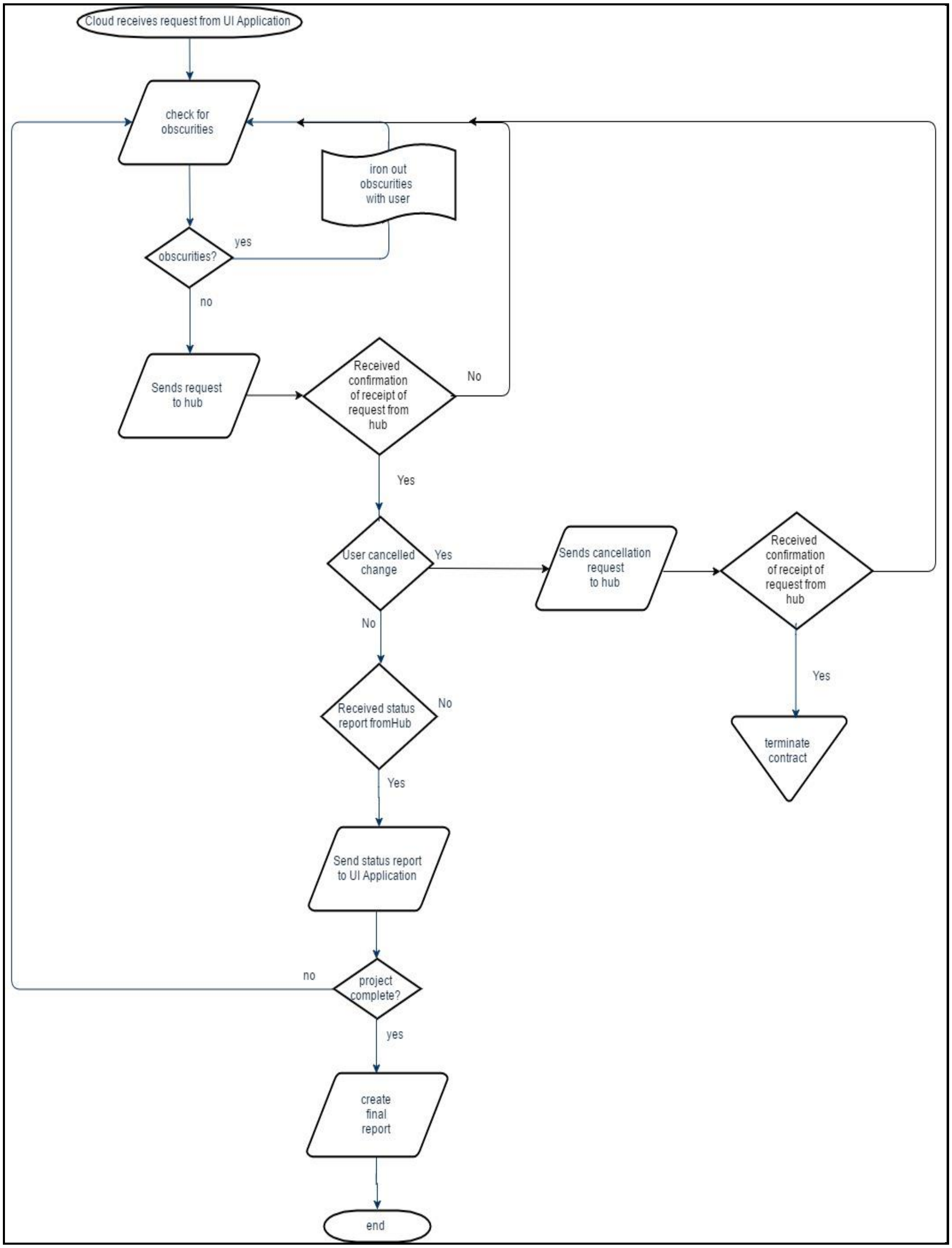

Figure 21: Operation Flow of the Cloud Server

3.6. User Interface- Stratos Application

3.6.1-**I** All the operations shall be supported using voice input and voice output. The app shall talk to the user using following keywords (the use case demonstration has been shown in Fig 23 and 24 respectively):

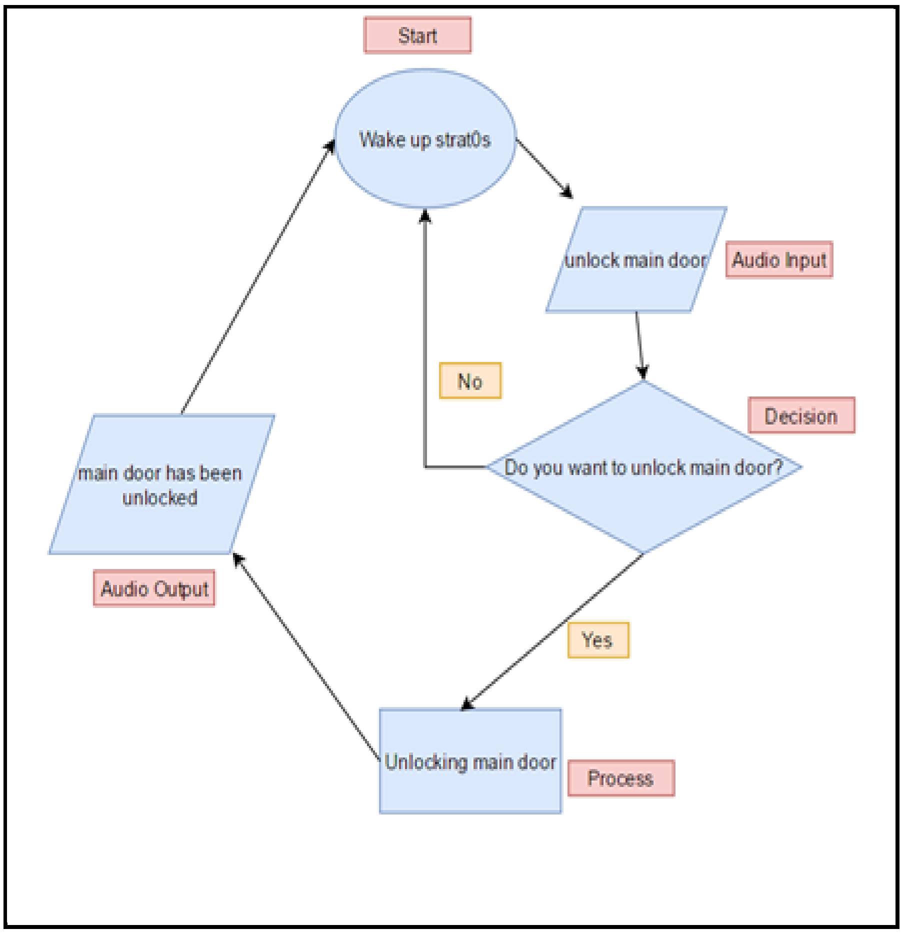

Figure 22: Sample UI process for unlocking main door

1. INFO This instruction shall read the customized information page of our application
2. PICTURE This instruction shall open the camera and inform the user that picture identification page is open and double tap to take a picture

3. STOVE This instruction shall read the temperature of the stove for the user

4. HEALTH This instruction shall read the temperature of the user's body

3.6.2 -I Each time user asks for status of connected devices, App will send API calls to cloud. Below is an example of listening intent, which will listen to user's command:

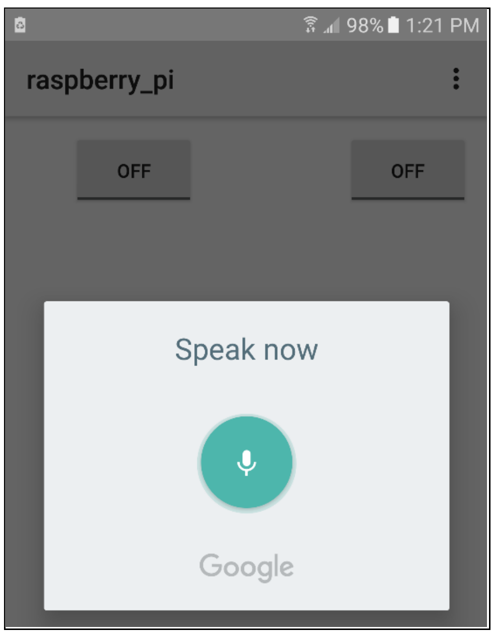

Figure 23: Example of User Interaction with UI

3.6.3-I The app shall have a feature to take an image and report the color and shape of the image. Color detection decreases the dependence of our users on others.

4. Non-Functional Requirement

4.1-**I** The cost of each product will be cheap to make it affordable for disabled people.

4.2-**I** All components will be covered in Lexan box case.

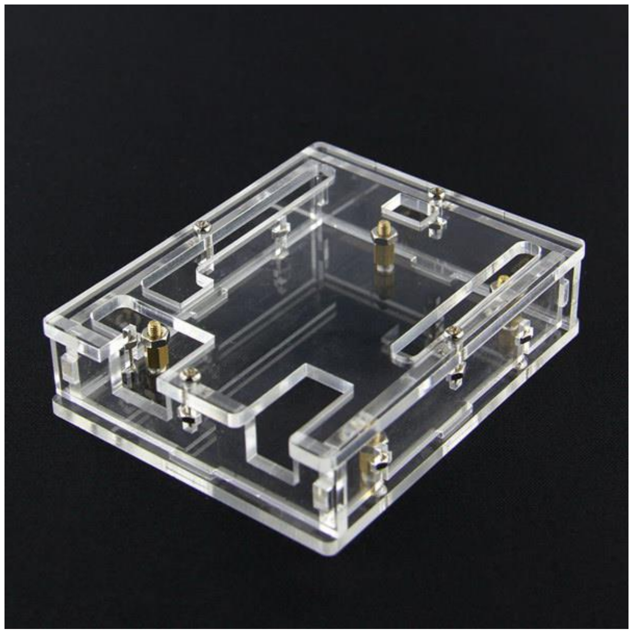

Figure 24: Sample Lexan casing for hardware[3]

4.3-**I** The electronic components of the device shall not cause interference with other devices.

4.4-**II** The casing of the product will be Nano micro porous for excessive heat dissipation.

4.5-**III** Size of hub's and other product's casing 10 x 10 x 10 cm (subject to change).

4.6-**III** All product casing will have polystyrene to avoid any damage if it fell from the wall or ceiling.

4.7-**III** All the product shall be powered via rechargeable batteries, so in case of emergency, such as electricity shortage the system is up and running.

4.8-**III** Battery should be easily accessible so user can change it by when a replacement is required.

4.9-**III** Micro-processors will be connected via Ethernet as well as a backup if WIFI is out of service.

5. Conclusion

In conclusion, our EasyHome platform is composed of a wide variety of requirements needed for its optimum operation. This includes our EasyHome Components, EasyHome Hub, EasyHome Cloud, and EasyHome UI. Components are the devices that take the input needed to provide feedback to the user through an XBee module, and our hub. The hub acts as the brain of our platform by establishing the communication between components and the UI.

The UI is the key element of StratOs Technologies' EasyHome products as it is what provides feedback and monitoring to the customer, especially when our target audience are visually impaired people. Feedback and monitoring is established through the UI as our components send data to our hub, which reroutes them to the user's device. The data appearing on the UI would be the component's current status, the appliance's state such as an oven's temperature, or a door's lock status, which would be read out loud to the user by the application on the smartphone.

Finally, the cloud acts as an alternative route for the component's data to direct to before heading back to the UI. This would increase the options through which feedback could be provided such as through the internet for remote access (outside the house), or simply to store data history on the cloud for future reference. Our team has provided a list of each of the requirement specifications needed for our platform with a set priority level to indicate its importance and necessity for our platform.

StratOs

Appendix- Test Plan

Our products are specially design for visually impaired people, the following critical test plan will ensure the high performance and usability by eliminating any error/bug. Each component will go through unit testing before integration. Once integration is successfully completed, system testing or User Interface will be tested to confirm its requirement functionality. Below is the example of critical test plan:

		Hub
Tests	Steps + Expected Results	Test Result (Pass/Fail)
Provide power supply to the hub	Raspberry Pi should turn on and 1st red LED should turn on	
WIFI Setup	Connect Ethernet cable to Hub. If a connection is made 2nd green LED light should turn on	
	Navigate to Raspberry Pi Settings and provide WIFI credentials. If WIFI is connected 3rd green LED light should turn on	
Secure connection between Hub and AWS	Login to AWS server and navigate to console. You should see a dot on today's date. That indicates that a connection has been established	
	Login in raspberry pi and navigate to strat0s folder and see if executable file is created and is running	
ZigBee Communication	Login to product microcontroller and navigate to strat0s folder. Executable file should be created. Use terminal to make sure it is running	
	If the executable file is running 4th blue LED should turn on	
Credential Certificates	Login to Hub microcontroller and navigate to strat0s folder and see	

	if public key, private & public key and certificate file is present	
		Cloud
Credential Certificates	Login to AWS cloud and make sure private & public key and certificate is created	
	Compare them with the one in Hub. They should be the same	
UI and device information exchange	The device and component information exchange after entering the specific user credentials should match with what user bought from StratOs	
		Stove
ZigBee Connection	Check if ZigBee wires are connected to GPIO pins as mentioned in developer user-manual	
ZigBee Communication	Login to microcontroller and navigate to strat0s folder. Executable file should be created. Use terminal to make sure it is running	
Sensor Connection	Check if all sensors wires are connected to GPIO pins as mentioned in developer user-manual	
Receiving data	Run sensor-test file and see if it outputs the temperature for all sensor. Run the test file on at-least three different temperature	
		Door Lock
ZigBee Connection	Check if ZigBee wires are connected to GPIO pins as mentioned in developer user-manual	
ZigBee Communication	Login to microcontroller and navigate to strat0s folder. Executable file should be	

	created. Use terminal to make sure it is running	
Magnetic lock Connection	Check if the magnetic lock wires are connected to GPIO pins, as mentioned in developer user-manual, using relay	
Receiving data	Run door lock-test file and see if it turns on and off the relay switch	
		Body Temperature
ZigBee Connection	Check if ZigBee wires are connected to GPIO pins as mentioned in developer user-manual	
ZigBee Communication	Login to microcontroller and navigate to strat0s folder. Executable file should be created. Use terminal to make sure it is running	
Sensor Connection	Check if the sensors wires are connected to GPIO pins, as mentioned in developer user-manual	
Receiving data	Put the probe under your armpit and run body temp-test file and see if it gives appropriate output	
		User Interface
This test shall only be performed when all above test is done and PASS. Also, this test should be done with all the end devices including body temperature and Stove. Install strat0s App on test device and connect your device with Charles.		
Login	Sign in using authentic credentials	
Wake up command	"wake up stat0s" should open the app and speaks "Hi, how can I help you today"	
User interaction (Successful)	Ask app to "unlock the door". It should ask for confirmation "Do you want to unlock the main door"	

	Say “No”. It should give feedback “ok, no action taken”	
	Now speak the command again and say “Yes” to confirmation. It should speak “alright, unlocking the main door”	
	The App should now get back to you with either a successful message	
User interface (Error)	Now repeat all method when hub is not connected to internet. Output should be “Error communicating with Hub. Please check internet connection”	
	Now repeat all method with no ZigBee connection. Output should be “Error communicating hub device”	
Checking API calls	Now open Charles and see if the API calls are made to AWS when users confirms an action	
	Check if the key in API matches with the user’s key. It should be same as each user will have individual key to access AWS	
Data confirmation	Check if the current state of device matches with cloud	
	If above fails, make sure Cloud and Hub test plan were done properly and all Passed. If still fails FLAG, it and report it to developers	

Test Case 01 – Stove Detector		
Actions/Steps	Expected Results	Test Result (Pass/Fail)
Wake up command "wake up stat0s" should open the App	No physical interaction required, and the application loads and informs the user that it has loaded.	

Voice Command the app to give feedback on stovetop status	The app will tell the user which burner is currently on aurally.	
Adjust the distance of the detector from the burner (20cm, etc.)		
3. Verify that the detector is accurately reporting the temperature threshold is exceeded		

Test Case 02 – Door Lock		
Actions/Steps	Expected Results	Test Result (Pass/Fail)
Wake up command "wake up stat0s" should open the App	No physical interaction required, and the application loads and informs the user that it has loaded.	
Voice Command the app to lock /unlock the door	If the door is already unlocked/lock. App should speak "door is already unlocked/locked"	
	If not, then it should lock/unlock the door as per user request	
Voice command to "status of the door"	It should provide with the current status of door	

Test Case 03 – Body Temperature		
Actions/Steps	Expected Results	Test Result (Pass/Fail)
Wake up command "wake up stat0s" should open the App	No physical interaction required, and the application loads and informs the user that it has loaded.	
Voice Command the app to "check body temperature"	After measuring the temperature from probe app will tell the user his/her body temperature	

User Interface Design Appendix

Table of Contents

List of Figures

List of Tables

1. Introduction

User interface is of utmost importance to StratOs technologies. We target customers with visual impairment and are thus in need of a simple, and practical interface to use our EasyHome product line. Our user interface requires minimal physical interaction and operates using aural communication through our smartphone application.

Furthermore, our EasyHome product line is designed to enhance the safety and security of visually impaired people at home. For instance, while designing the stove top detector, several safety factors were taken into consideration. This includes fire hazards, wiring issues, short circuits, proper grounding, and accurate measurement to avoid having wrong feedback of hot burners to the user which will be hazardous if left on for prolonged periods of time. We also designed our door lock component considering security issues that may arise, and tackled them accordingly by providing feedback of any unusual situations straight to the user through our UI app.

We aim at making our platform clear, simple, and easy to use by visually impaired people and easy to troubleshoot on behalf of the technician in case a fault occurs. This appendix will serve the purpose of providing a detailed explanation of our approach in designing a user interface for our EasyHome platform to meet the needs of visually impaired people. Moreover, it will cover testing mechanisms undertaken by our team to provide an accurate simulation of the challenges that a blind person might face while using our products in a day to day basis.

StratOs

2. User Analysis

Main users of our proposed product are visually disabled folks. In order to make our product safer and easier for our user we analysed our users based on several factors including knowledge required to use the proposed system or device, restrictions with respect to previous experiences and physical ability requirements to use EasyHome products.

- User knowledge needed to use EasyHome solution was analysed separately for user Interface i.e. Phone Application and hardware.
    - User Interface - Phone/ Web Application
        - User should have knowledge on how to install and use an application on their phone. A hard copy of instructions on how to install and get started with our application on Android and iOS will be included in braille print as well as available on our website.
        - User should be able to activate voice input on their phone. This requires knowledge of using vision accessibility feature Voiceover on iphone or TalkBack in Android. Based on surveys conducted prior to design process most of our subjects were already using assistant applications and similar techniques.
        - User should be able to login on a web browser. If our user is comfortable with using computer this shall be a very basic step. This part, however, is purely intended for users who use a smartphone other than iOS and Android.
    - Hardware - Hub and End components
        - User should be able to press on or off button on the hub and end components. Since our user is visually disabled they will require assistance from visually abled person for resetting hardware. Moreover assistance will be required in connecting end component devices to power supply, stovetop and door as well as connecting hub to power and internet if users moves residence. First time installation shall be provided by EasyHome technicians.

- While designing our hardware and User Interface we are trying to solve common problems with similar products and applications available in the market. While most applications currently available in the market require some kind of interaction with phone's screen we are attempting to eliminate need to touch phone screen completely.

- Usage of our product requires a user should not be mute or deaf since primary mode of communication is the audio feedback. Since our user is visually impaired, assistance of a visually abled person may be required at times.

StratOs

3. Technical Analysis

Hub

Discoverability	The hub will be placed close to the router in the user's house. There will not be much direct user interaction required after initial installation except resetting power switches and cables. The power button will be at the front of the device and shall have raised power symbol etched. The wiring from router to hub will follow color coding.
Signifier	Etched power symbol at the power button will make it possible for our user to turn power on/off since they can feel the power symbol and understand that this button controls power. Color coding the cables and ports will make it much more intuitive for the person assisting our blind user to reconnect the devices if required.
Feedback	The only feedback mechanism on Hub hardware is 3 LEDs which indicate the status of zigbee, ethernet and power. This will be of help for the installer or any helper of blind person.
Affordances	Standard Universal Micro USB and Ethernet Port will only be open ports on the Hub.
Mapping & Constraints	The 3 LEDs used for feedback will signify normal operation when on. This is an industry standard and thus is a logical constraint as the technician or user's helper can notice that device is not functioning normally when LEDs are off.

iOS and Android application

Discoverability	The mobile applications will be available in the app store for our users, they will also be installed during the installation process. Information on how to install the apps will be included in user manual as well as on our website. Since our application minimizes need for touching the screen it will be much easier for the user to discover new features using voice.

Feedback

All the interface control will be provided through voice. As our user is blind, we will design our app in a way that they will be able to use some keywords to interact with the UI. The figure below has an example of a feedback mechanism.

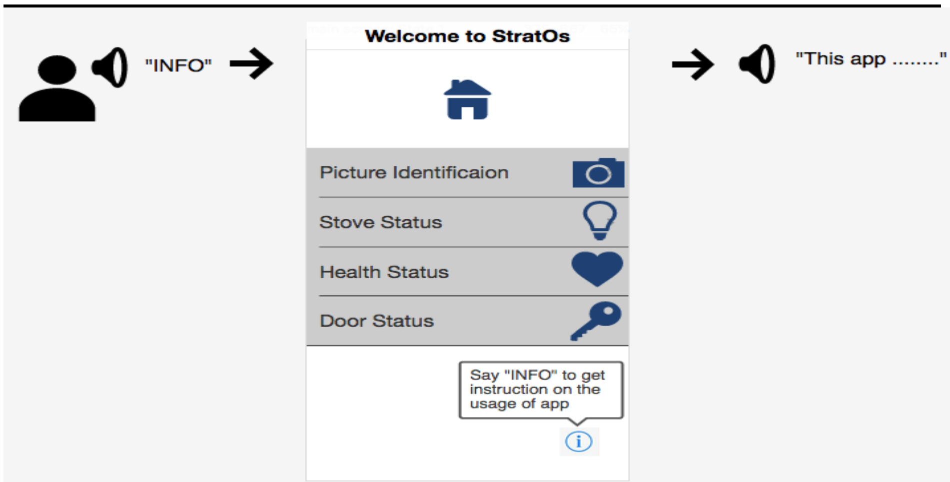

Figure 25: User Interface

Conceptual Model

Table 3:Conceptual model table

Components	Possible User voice Inputs/ corresponding Outputs	
Stove Detector	Stove /The Stove A is (on/off)	
Color Detector	Color/Take a Picture to clicking the mobile screen once	The colour is —- in the picture that you took
Door Lock	Door/ The door is (on/off)	
Body Temperature	Body temperature/ The body temperature is — degrees	
Info	The StratOs app is open. User can input following keywords in order to use the application: Stove Detector, Color, Door, Body Temperature	

StratOs

Affordances and Signifiers	User will get a predetermined feedback when they are asking control of a device that does not exist. Also voice options will only be provided for the devices configured. The voice options will inform user of possible actions for each device. Touch options will be disabled on the User Interface because our primary users are visually disabled people.	

Components

Discoverability	Stove Detector	The stove detector will be placed on top of stove and can be attached to the exhaust hood to detect stove status as precisely as possible.
	Door Lock	The door lock will be placed on top side of the door and door frame.
	Body Temperature Sensor	The body temperature sensor will be placed by the user at a location of their preference such that during usage they can touch the probe under their armpit.
Feedback	All the feedback will be provided for all the components through software application on user's phone. When an action is performed the user will be informed of success. Also in case of failure in performing some task user will be notified with appropriate message.	
Affordance	Stove Detector	User does not have any interaction with this component.
	Door Lock	User does not have any interaction with this component. Location of this component will provide a good indication of its action.
	Body temperature Sensor	The probe will be structured such that it makes more intuitive for user to touch it. Voice message on phone will provide clear instructions.
Signifiers, Constraints & Mappings	Since our user has minimal interaction will stove detector and door lock signifiers, constraints & mappings will exist mainly for body temperature in the form of voice messages properly	

instructing user to touch the probe under his armpit using the extendible cable.

4. Engineering Standards

- The Wi-Fi dongle attached to the Hub follows the WI-FI/IEEE 802.11 standard [4].
- For interoperability and scalability, the XBee modules used for the Hub and Components comply with the ZigBee/IEEE 802.15.4 specification [5].
- The hub's Micro-USB interface conforms to the USB2.0 standard [6] to guarantee safe power.
- The temperature sensors that will be used shall adhere to the Temperature Measurement Standards [7] to ensure accurate and reliable results.
- The magnetic door lock that will be used shall comply with PC95.1 standard [8] to assure safe user exposure to electromagnetic fields.
- The body temperature sensor shall comply with the IEC 60601-1-11:2015 standard [9] to verify safe operation of the medical electronic equipment for home environment use.
- The cloud server shall have security features outlined in AWS Cloud Security Resources [10].
- The hub and the components shall comply with the Canadian Electrical Code Part I [11] to ensure the safety of electrical equipment and their installation.
- The hub and the components shall comply with the Canadian Electrical Code Part II, NO. 61508-2:17 [12] for the functional safety of electronic devices.
- All materials shall be RoHS compliant [13] to reduce the exposure of the user to hazardous materials found in electronic products.
- The hub and the components' power circuitry shall conform to the IEEE- 1625-2008 standard [14] for the reliable operation of the system with rechargeable batteries.

5. System Test Plan

All our product will be tested thoroughly at each stage of development. It is to eliminate all errors and to ensure that our system is safe and secure as user is primarily dependent on the functionality of our product. To maintain high quality standard the system will undergoes testing process to make sure it is working as intended and meets all requirement specifications. Each component used in our system will undergo unit testing. Once unit testing is successful it will undergo integration testing and make sure all devices are communicated with each other. Once that is achieved, system testing will take place where the whole system will be tested as user's perspective, as a black box.

Phase 1 Unit Testing

Unit testing will be conducted on individual components of the system to determine each component is functional as follows:

End Components Unit Test Plan

To test the microcontroller, need to ensure,

- All sensor, magnetic lock and ZigBee is corrected to the right GPIO pins as mentioned in developer guide.
- ZigBee has been soldered to microcontroller, as per developer guide.
- ZigBee SDK and AWS SDK has been installed in stratOs folder.
- User's Private key, Public key and certificate has been placed in cert folder under stratOs, for authentication purpose.
- Respective components developer code file has been copied and executable file has been created.
- Run the excitable file and check if it is receiving data from Hub. It should display "Hello from Hub " and LED on HUB should turn on.

Hub Unit Testing Plan

- ZigBee SDK and AWS SDK has been installed in stratOs folder.
- User's Private key, Public key and certificate has been placed in cert folder under strat0s, for authentication purpose.
- Hub developer code file has been copied and executable file has been created.
- WIFI dongle is plugged in to Raspberry Pi.
- WIFI credential are saved in order for hub and cloud communication.
- Verify that WIFI and Ethernet LED turn on when internet is connected.

User Interface Unit Testing

- Wake up command “wake up strat0s” should open the app.
- Ensure Listening intent is working and app is able to detect when and what user is speaking.
- Ensure speaking intent is working and app is giving feedback to user’s input.
- Ensure word detection is working and app is able to differentiate between different verbs and nouns.
- Ensure “color detection” command open back camera, takes picture and detect the color of the object in the picture.

Phase 2 Integration Testing

Integration testing allows the opportunity to combine all of the units within a system and test them as a group. We will integrate all the hardware component and run the executable file in order to ensure that the data has been communicated as expected and the product is meeting all functional requirement. Once all the aforementioned unit testing is approved we will then integrate them one by one and test them to ensure they are working as expected.

This is achieved by the following steps:

- Ensure devices are turned on and all unit testing is approved.
- Run ZigBee-test file on ends component and make sure it is outputting “Hello from Hub”.
- Run hub-test file on Hub and make sure it outputs “Hello from end component” and “Hello from AWS”.
- If any of the above not worked as expected flag it and report it to developer.

Phase 3 System Testing

Critical Test Plan plays a vital role in our system Testing. Critical Test plan is similar to unit testing but cover more technical aspect. Critical test plan can be found in Design Spec document. Before doing system testing we need to make sure that each end device including stove, door lock, body temperature and most importantly User interface, Pass critical test Plan.

<u>Stove State Detector</u>

Placement of detector / Spatial & Physical:

Safety - location of the detector would not cause a fire/electrical hazard

- Detector positioned such that it does not fall/land onto the burner itself.
- The detector should not cause an obstacle hazard for the blind person.
- The detector should not cause a fire/electrical hazard
- Wires insulated and placed away from heat behind the stove.

* The user should not be exposed to harmful emissions from the components if the detector circuit accidentally shorts.
* Avoid macro shocks - which would occur if the grounded person touches a grounded wire - by having a grounded chassis to deviate the current away from the person and back to the device.

The figure below is an illustration of a macroshock scenario that would result if two leads are handled by a grounded person:

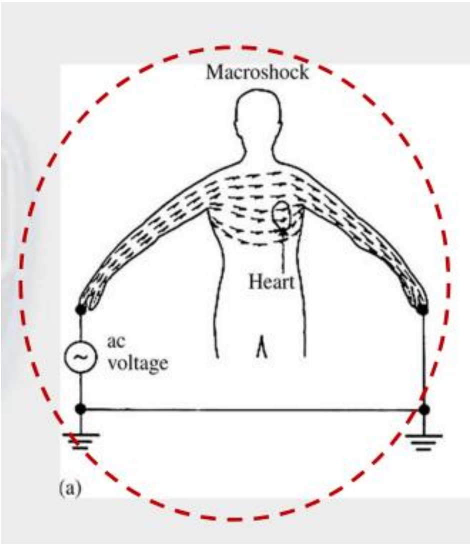

Figure 26: Definition of a macroshock [15]

Ease of use – it should be easy for user to handle the detector

* Wired properly and accurately to avoid intertwining of wires.
* Proper packaging of all the components into one container that fits just right.
* Battery should be a commonly found battery type.
* Battery should last for at least two weeks without charging.
* Battery should be easy to charge.

The aforementioned tests are example test cases for stove detector. Similar cases will be performed on all end points. The high performance and high quality of the overall system will be achieved by Unit testing, Integration Testing, Critical Test Plan and Test Cases.

Door Lock Mechanism

Safety - Door lock should not cause an obstacle hazard

- Test - Door lock should only try to lock when the door is stationary.
- Test - Door lock should be away from door knob.
- Test - Door lock should not interfere with normal mechanical door operation when not active.
- Test - Able to install the door lock in a firm and secure position.

Ease of use:

- Door lock should be able to act as a replacement for the manual mechanical lock mechanism.
- It should be easy to uninstall/replace from the inside without causing problems.

Security:

- Ensure the door lock mechanism is not faulty and provides the correct feedback each time.
- Ensure the cloud service is secured to avoid hacking which would leave the door lock vulnerable.
- Provide alerts to the UI in case multiple physical home intrusion attempts occurs.
- Ensure the door locks when commanded to lock and unlocks when commanded to unlock - this is done by accurate voice recognition algorithms-taking into account all sources of noise to the voice, different accents, and faulty microphones.
- In case the phone is stolen, the user should block the phone's access to the Hub from the Hub's ports.

6. Conclusion

Our EasyHome products are designed to enhance the lives of visually impaired people at home thus, an appropriate user interface that is simple, safe, and easy to handle is crucial to our solution. Given that one of our main products is closely related to stoves and ovens we have to consider safety in our design, that is, avoid the risk of fires, or short circuits that might be hazardous for a blind person living alone.

Moreover, our DoorLock mechanism has to be designed in a way that gives accurate information about the status of the door lock and respond to the user's commands on the spot as well. We have to consider security issues that might arise from our platform and tackle it accordingly as in the case of having a stolen phone or having the automatic lock not functioning properly. Another scenario that we need to consider is in the case of power outage in which case an alternate power supply is securely installed. At StratOs technologies, our goal is to deliver an easy to handle, aural communication system that's secure, and safe at the same time.

7. References

[1] Hendricks. D,(2014). The History of Smart Homes. [Online]. Available:http://www.iotevolutionworld.com/m2m/articles/376816-history-smart-homes.htm

[2] Digi International, Inc. (Digi), [Online]. Available: https://www.digi.com/resources/documentation/Digidocs/90001942-13/#concepts/c_transmit_receive_packet_exchange.htm%3FTocPath%3DXBee%2520API%2520mode%7CXBee%2520frame%2520exchange%7C_____2. [Accessed 28 March 2017].

[3] http://img1.banggood.com/images/2014/xiemeijuan/02/SKU206392/1. [Accessed 28 March 2017].

[4] IEEE Standards Association, "IEEE-SA -IEEE Get 802 Program - 802.11: Wireless LANs," [Online]. Available: http://standards.ieee.org/about/get/802/802.11.html. [Accessed 28 March 2017].

[5] ZigBee Alliance, "Standards: ZigBee Specification | ZigBee alliance," [Online]. Available: http://www.zigbee.org/download/standards-zigbee-specification/# [Accessed 28 March 2017].

[6] USB Implementers Forum, Inc., "USB.org - USB 2.0 Documents," [Online]. Available: http://www.usb.org/developers/docs/usb20_docs/usb_20_011317.zip [Accessed 28 March 2017].

[7] ASTM International, "Temperature Measurement Standards," [Online]. Available: https://www.astm.org/Standards/temperature-measurement-standards.htmL [Accessed 28 March 2017].

[8] SCC39/Technical Committee 95/C95.1 Revision Working Group, "IEEE SA - PC95.1 - Standard for Safety Levels with Respect to Human Exposure to Electric, Magnetic and Electromagnetic Fields, 0 Hz to 300 GHz," [Online]. Available: https://development.standards.ieee.org/get-file/PC95.1.pdf?t=14861000003 [Accessed 28 March 2017]

[9] International Electrotechnical Commission (IEC), "IEC 60601-1-11:2015 - Medical electrical equipment -- Part 1-11: General requirements for basic safety and essential performance -- Collateral standard: Requirements for medical electrical equipment and medical electrical systems used in the home healthca," [Online]. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=65529 [Accessed 28 March 2017].

[10] Amazon Web Services, "Cloud Security Resources," [Online].

Available: https://aws.amazon.com/security/security-resources/ [Accessed 28 March 2017].

[11] Canadian Standards Association, "C22.1-15 | C22.1 Canadian Electrical Code | ShopCSA," [Online].
Available: http://shop.csa.ca/en/canada/c221-canadian-electrical-code/c221-15/invt/27013892015 [Accessed 28 March 2017].

[12] Canadian Standards Association, "CAN/CSA-C22.2 NO. 61508-2:17 | Canadian Electrical Code Part II-General Requirements | ShopCSA," [Online]. Available: http://shop.csa.ca/en/canada/canadian-electrical-code-part-ii-general-requirements/cancsa-c222-no-61508-217/invt/27041522017 [Accessed 28 March 2017].

[13] European Union, "RoHS Compliance Guide: RoHS 6 Restricted Substances," [Online]. Available: http://www.rohsguide.com/rohs-substances.htm [Accessed 28 March 2017].

[14] Institute of Electrical and Electronics Engineers (IEEE), "IEEE SA - Standards Store | IEEE 1625-2008," [Online].
Available: http://www.techstreet.com/ieee/standards/ieee-1625-2008?gateway_code=ieee&vendor_id=4382&product_id=1588785#full [Accessed 28 March 2017].

[15] B. Gray, "Bmed-Instr Safety: Electrical Safety Set 1," 2017.