Smart Fridge

Chase Breeden, Carter Crews, Justin James  
{charbree, carcrew, jusrjame}@ut.utm.edu

***Abstract*—If you have ever left your house wondering whether you left the refrigerator door open, had a roommate stealing your food/drinks, or just wanted to monitor the temperature in your fridge, then Smart Fridge is what you’re looking for. With less than $100, you can build your own Smart Fridge. The Smart Fridge uses a door sensor with two temperatures sensors mounted to a Raspberry Pi. This setup communicates with Amazon Web Services to monitor the fridge in real time and alert the owner if the temperature makes a dramatic change.**

# INTRODUCTION

## IOT

IOT stands for Internet of Things. It is a term used to describe a range of devices that are connected to the internet to be controlled remotely, like thermostats, garage door openers, and even toasters.[1]

## Smart Fridge

Our Smart Fridge is a monitoring tool for owners of refrigerators that wish to monitor their refrigerator’s operational status. This tool can be used to monitor multiple refrigerators in real time at the owner’s request.

# BACKGROUND

The idea originated from an issue at Carter’s home. Carter’s family has a refrigerator with a door that won’t always close. Sometimes this would happen when he was going to get a midnight snack, and by the time the morning came, all the food in the refrigerator would be ruined.

This led to the idea of creating a monitoring system to alert someone to the danger at hand. This way the issue could be corrected before disaster settled in. At it’s core, the goal was to help keep Carter out of trouble with his father.

The original idea was to use a Raspberry Pi to monitor the refrigerator, while using Amazon Web Services (AWS) to send notifications, and each Pi hosting it’s own web server that could be used to view the status of each device while on the local network.

Eventually, the concept was modified to rely more heavily on AWS. Instead of having each Pi act individually, we would create a system that would report their status to AWS IoT. AWS IoT would then act on this data, and send notifications as appropriate. We could then create a single web page that would display the status of all connected Raspberry Pis and host it on an EC2 server, so that the status of all connected refrigerators could be viewed from anywhere in the world!

# DEVELOPMENT TOOLS

## Python 3

Python 3 is an updated version of the Python programming language, released in 2008. While many of the languages features are the same, the way built-in objects such as strings and dictionaries work. [2] In this project, we were able to use a Python 3 script to setup communication between the sensors and the Raspberry Pi.

## Raspberry Pi 3

The Raspberry Pi is a credit card sized, fully functioning computer. This computer runs a Linux based operating system called Raspbian. We use this computer as a controller for the various sensors attached to it. It processes the data then sends it to AWS where the data is then distributed to the user either via text alerts or in real time via a web page.

These small computers have General Purpose Input and Output (GPIO) pins in order to interact with the sensors. The sensors are attached directly to the pins and are the state of the pins is read via python script.

## Uxcell MC 38 Door Sensor

The Uxcell MC 38 Door Sensor contains magnetic contacts that function as a reed switch. The circuit is closed when the magnetic contacts are close together and the circuit is open when the magnetic contacts are pulled apart from one another. The contacts are equipped with adhesive tape to easily attach the switch to almost any surface. While the door sensor is plug n’ play, it still requires a simple python script is used to utilize the switch and send data to the Raspberry Pi.

## DS18B20 Waterproof Temperature Sensors

The temperature sensors are digital sensors that have a range from -67 degrees Fahrenheit to 257 degree Fahrenheit and are accurate within half of a degree. Unlike the magnetic door sensors, the temperature sensors are not plug n’ play and require changes to the Raspberry Pi’s configuration file. However, like the magnetic sensors, the temperature sensors utilize a Python 3 script to communicate with the Raspberry Pi.

## Amazon Web Services

## In this project, we used AWS as a backend to receive the data from the Pis, and to process and act on this data. We used several services to turn this project into a reality, including IoT, EC2, and SNS.

## AWS IoT

## AWS IoT is a relatively new addition to the AWS portfolio of services. It used MQTT-based messages to communicate the status of several connected devices back to AWS. AWS then stores this reports in what’s called a device shadow. This device shadow serves as a place where the stored data can be read and acted upon from other services or SDKs for AWS.

## Within IoT, there is a rules engine that can act on data received from the device. In this case, we created a rule to monitor for special messages from the Pi indicating it was in an alarm state. At this point, the message is then forwarded to SNS, which will send a message to the every number on the subscription list.

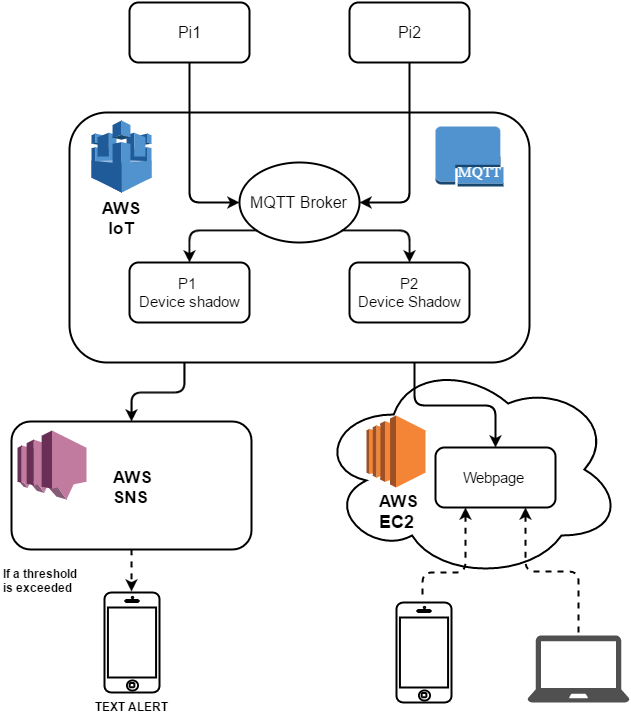
## Elastic Compute Cloud (EC2)

## EC2 is a platform for hosting virtual servers in the cloud. It is widely used as a cheap hosting platform for websites. For this project, we piggybacked on the EC2 server that Carter has to run his personal website. Setup is fairly simple and can be easily completed by following the documentation provided by Amazon.

## Simple Notification Service (SNS)

SNS is a service that is used to send all kinds of notifications, from text and email notifications, to push notifications to smartphones. In our implementation, we use SNS to send the user alerts when the temperature drops below a certain threshold, or if the doors are open for too long.

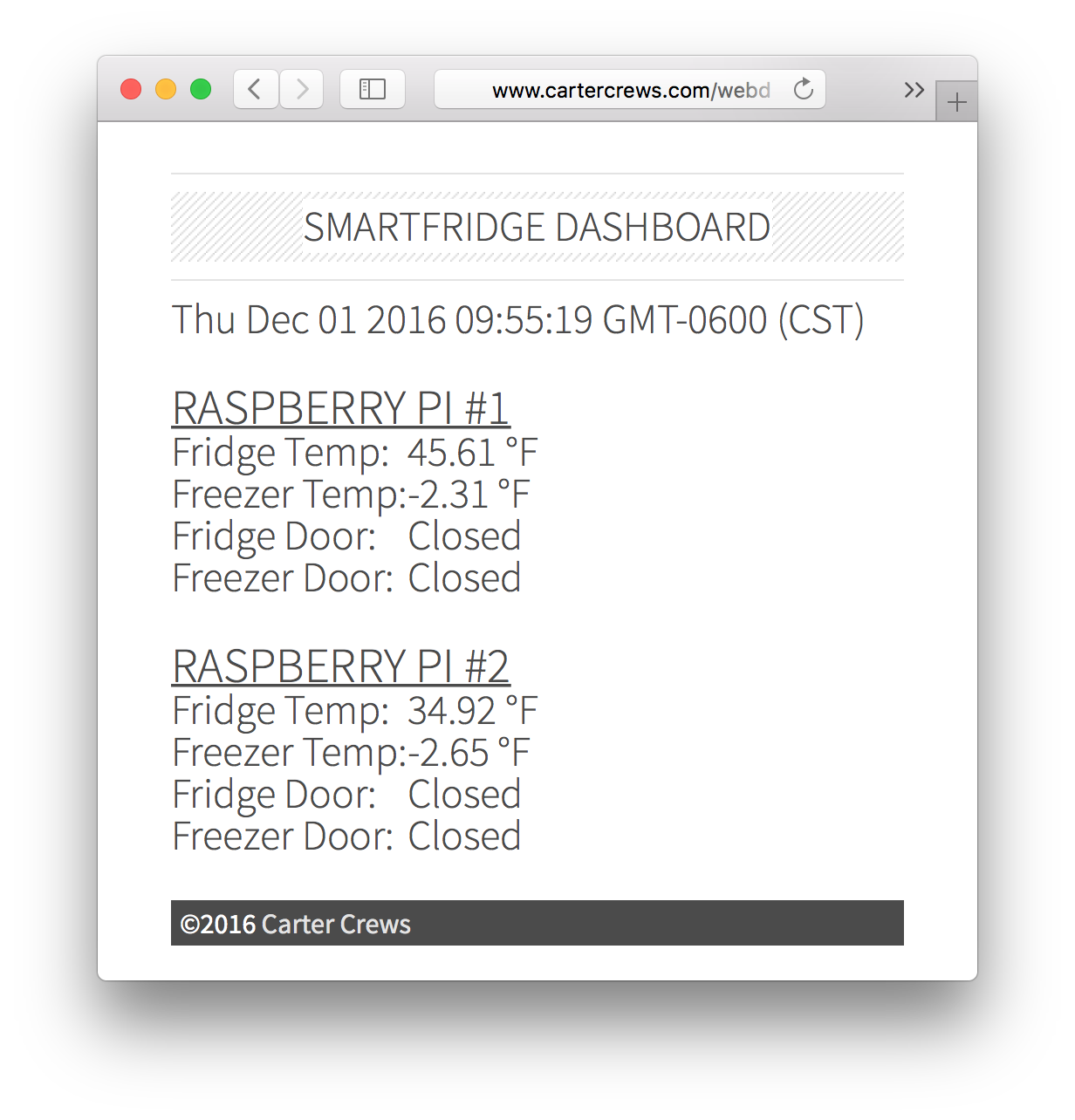
**Figure 1**



# FEATURES

This monitoring system constantly gathers temperature and door state information in order to be relayed to the user. The user can either check the status of his/her refrigerator in real time via a website or let the system send them an alert if an error state has been reached. An error state would be reached if the temperature exceeded a predefined limit or if the door is open for longer than it should be.

**Figure 2**



# SUMMARY

Smart Fridge evolved into a simple monitoring tool for owners’ to analyze their refrigerators’ performance. Smart Fridge proves that adding internet capabilities to a refrigerator can be done well under the market price of a commercial smart refrigerator. This allows the owner to prevent spoiling all of the contents in the refrigerators themself and will allow the owner to determine a precise timeline for when the issues occurred.

# FUTURE DEVELOPMENT

The beauty of the Raspberry Pi is that it is fully customizable and can do many things. The only creative limits the Pi has . However, we only have a few future plans in mind for our system.

*A. User Customization*

Allow the user to customize the limits for sensor alerts and also allow the user to customize alert frequency.

*B. Camera Integration*

Set up a feature that allows the user to arm the camera remotely in order to take a picture whenever the door sensor’s state changes to open. The picture will then be sent to the user via a SNS alert.

*C. Reworked Webpage*

Add a feature that shows a timelapse of temperature data such as a bar graph.

##### References

#### 2016. Internetsociety.Org. Accessed December 1 2016. <http://www.internetsociety.org/sites/default/files/ISOC-IoT-Overview-20151022.pdf.>

1. “Python 3.0 Release”. 2016 Python.Org. Accessed November 30 2016. <http://www.python.org/downoad/releases/3.0/>