IoT Smoke Detector Using IBM Watson

Brandon Johns[1], Casey Delaney[2] and Peeradech Abhidechsakhulchai [3]

***Abstract*— This paper will go over the creation of an IoT Smoke Detector Using IBM Watson, as well as areas of difficulty that were encountered. Areas of difficulty include changing the cloud system, changing from Raspberry Pi to TI CC3220sf and changing the smoke detector sensor. At the same time, we also had issues with the CLI Docker which will be explained in the software slides. However, we were still able to connect the Sensor Tag to IBM Watson and receive data onto a graph.**

I. INTRODUCTION

An IoT Smoke Detector is a smoke detector device that goes beyond the traditional capabilities of your normal household smoke detector. A normal household smoke detector serves two purposes. The first purpose is detect when an imminent danger is present to your life, and the second purpose to to notify the nearby occupants of said imminent danger. Normally, the notification system is accomplished by broadcasting a loud, high-pitched periodic tone.

This system, although effective, lacks a very important characteristic. It is only effective at notifying people in the nearby vicinity. If the occupant were not in the house, then nobody would be able to notify the appropriate emergency crew. Potentially, this would result in the loss of a house, along with all belongings and pets trapped inside.

The goal of our device is to overcome this ancient technology by integrating IoT characteristics using IBM Watson. The device will detect if there is a present danger, and subsequent ally, it will send an alert to the users phone. This allows the person to be able to contact the appropriate services, even when they are not at home, which will ensure a higher chance of being able to save the premises.

*This work was supported by Professor KaiKai Liu

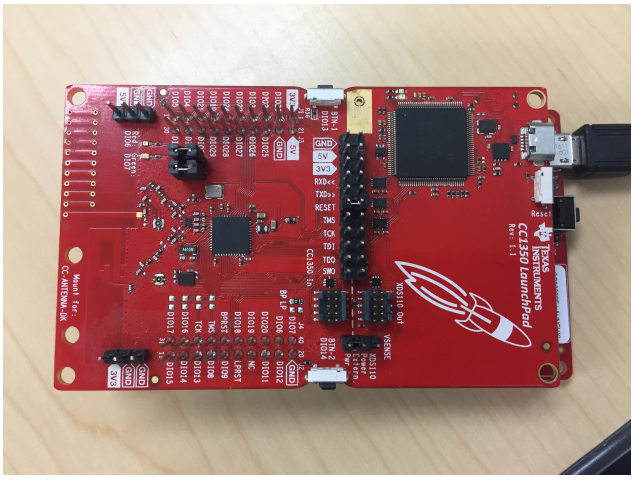
[1]B. Johns is with the Charles W. Davidson College of Engineering as a student in Computer Engineering at San Jose State University, San Jose, CA 95192, USA

[2]C. Delaney is with the Charles W. Davidson College of Engineering as a student in Computer Engineering at San Jose State University, San Jose, CA 95192, USA

[3]P. Abhidechsakhulchai is with the Charles W. Davidson College of Engineering as a student in Computer Engineering at San Jose State University, San Jose, CA 95192, USA

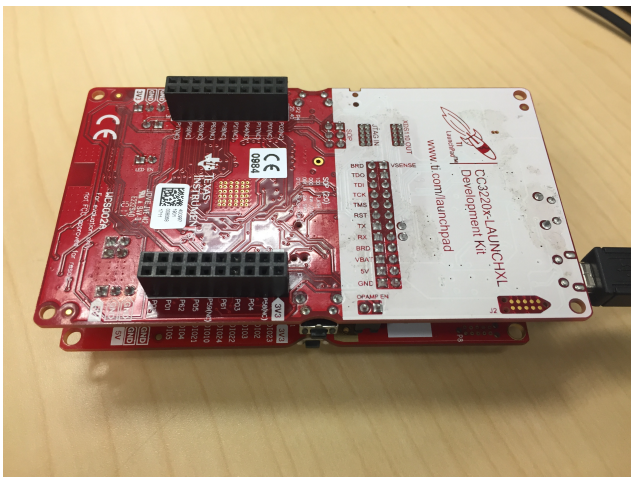
II. MATERIALS AND PURPOSE

A. CC1350 Launchpad



The purpose of the CC1250 Launchpad is to provide a connection to the sensor device. In our case, for the sensor, we stacked the CC28505 SensorTag with the Debuggger Devpack. In general, the CC1350 Launchpad provides limited functionality, but one thing it allows is to provide a connection via Bluetooth low energy. This connection is used via the 2.4-GHz protocol.

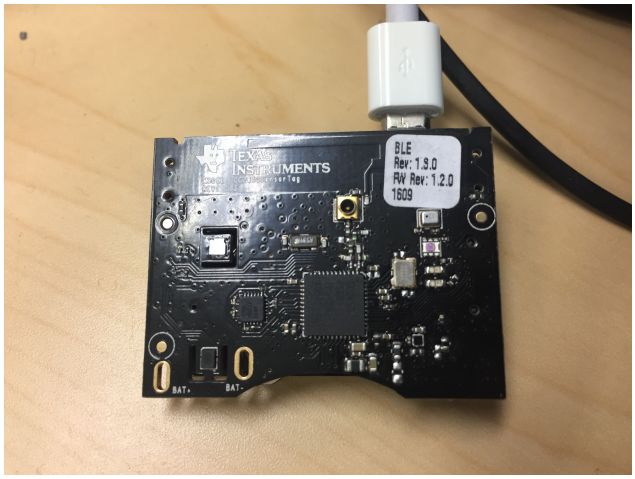
B. CC3220SF Launchpad



The CC3220SF Launchpad adds functionality onto the functionality of the CC1350 Launchpad. Because the CC1350 Launchpad does not have the ability to connect to WiFi, which would then enable a connection to the cloud, we must use the CC3220SF Launchpad to provide this functionality by stacking the two devices together. As seen in the image above, you can see that the CC3220SF

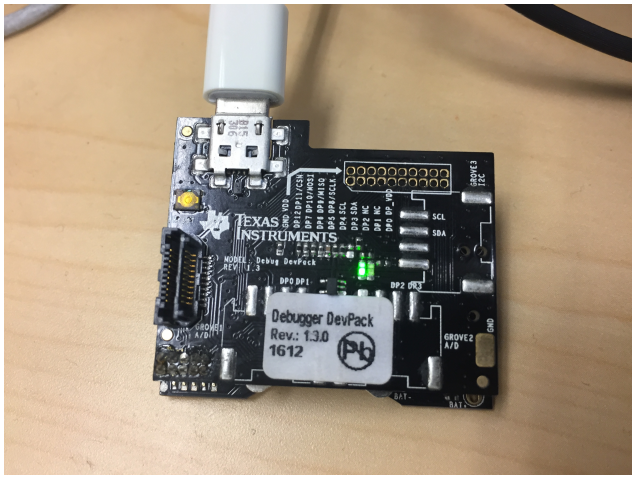
Launchpad is connected to the CC1350 Launchpad. This enables the user to have the functions of both a low energy Bluetooth connection as well as having the fucntion of connecting to a wifi source to access the cloud.

C. CC28505 SensorTag



The CC28505 SensorTag is the group's replacement for the Smoke Detector that we had initially tried to repurpose to have IoT capabilities. This device establishes a connection with the CC 3220SF, which will then relay the information through use of WiFi into IBM Watson.

D. Debugger Devpack



The Debugger Devpack was connected via stacking to the CC28505 SensorTag as depicted in the above image. It is difficult to see, but both of these devices are connected to each other. The Devpack allows access to the CC28505 SensorTag, and it is also the SensorTag's source of power.

III. Difficulties

A. Cloud System

The smoke detector was originally supposed to connect to Amazon Web Services as its main platform. After meeting with Professor Liu, a suggestion was made to switch to IBM Watson. As a group, our belief was that AWS would be simple and easy because of the amount of time spent learning about it earlier in the semester. It had never occurred that Watson was a capable Web Service. Professor Liu said connecting TI devices to Watson was fairly easy. At the time most of our homework in AWS, and that was our groups background. The assumption was that the homework could be implemented into the project. The idea of repurposing the first homework assignment into our project code seemed too good to pass up. The assignment taught how to connect the TI CC3220SF board to AWS, and the TI CC3220SF board was the main component of the smoke Detector.

However, the group decided to try a new platform, which created a new experience as well as understanding of various web services. At the time of building the smoke detector, the lectures on Watson had not been covered. All of the information for connecting to IBM Watson was found from information that the team gathered online. The user interface for IBM Watson was an easier platform for us to comprehend, although, the group still came across a fair amount of difficulties throughout the project. Because we did not learn about this service until the very end of the semester, it was difficult for us to troubleshoot our problems.

B. Raspberry Pi to TI CC 3220

The smoke detectors original plan was to incorporate a Raspberry Pi as the devices main platform instead of the TI CC3220SF. The debate went back and forth for the first month of this assignment. Eventually, it was decided on using the Pi, even though the original project submission said that a TI CC3220SF board would be used. There seemed to be endless resources online for a Raspberry Pi based Smoke Detector. The conclusion based on the resources online was that the Raspberry Pi would be a simpler project. As a group, there was a meeting with Professor Liu, and he explained how the TI CC3220SF board connects with the IBM Watson platform fairly easily.

Ultimately, the team decided to leave the Raspberry Pi project, and the TI CC3220SF was brought back into use again. The IT board has less documentation on it, and it is also is less user friendly than Raspberry Pi. Alternatively, Raspberry Pi had so many resources that it at the end of the day, the group decided on a challenge, which the TI board would deliver.

One of the challenges that the group encountered was that most of the tutorials online did not provide enough information, or the information was out of date. There was a hard time finding the required files and software for the board. Many tutorials required a level of background knowledge of the TI systems which our group did not have. A few days before the project was due, at the final steps of the getting our smoke detector to work, it was discovered that the TI board required Windows 10 Pro. The entire project was written on Windows 10 made finishing the project difficult. Transferring to Windows 10 Pro would mean that we would have to start the project over again. This code issues will be elaborated upon in the software presentation slides.

C. Sensors

The vision in the beginning was to incorporate a real smoke detector and modify it to connect to our Raspberry Pi or TI CC3220SF boards. The smoke detector was going to send alerts to mobile devices based on its detection of a fire. However, the smoke detector that was apart of the project did not send an analog or digital signal that could be read from the IT CC3220SF platform. The smoke detector required additional components that would increase the overall cost and complexity of the project. The project was quickly becoming much more complex and above the groups knowledge. All tutorials found online required many more components that simply could not be required. The goal was to use parts that Professor Liu, had in stock. Upon visiting Professor Liu in his office hours he made a suggestion and provided the CC28505 SensorTag.

Sequentially, the group chose the route of using a Sensor Tag device. The Sensor Tag was built with multiple functions; for example, humidity, acceleration, and temperature. The Sensor Tag is also a TI brand which made conversion easier for use with the IT CC3220SF board. The CC28505 board was able to send and receive information from IBM Watson. Once the sensor was found, the project began to quickly take form. The receiving and interpreting of data became a lot easier. There were also tutorials online with the Sensor Tag that helped us connect it to IBM Watson, unlike the smoke detector where the very general instructions left us confused.

IV. CONCLUSIONS

Overall, the team was able to successfully read data from the sensor (dubbed as our smoke detector) and send this data into IBM Watson. The difficulties that the team encountered was being able to successfully connect to CLI Docker for IBM Watson, but overall, the team was able to get 95% of the items to work correctly. The team also attempted to redo the project with the correct operating system, but this required restarting the project, of which we did not have enough time. Being that everyone on the team was very new to this stuff with extremely limited experience, we were happy with what we were able to accomplish.

ACKNOWLEDGMENT

This project was fully supported by San Jose State University, as well as the CMPE 189 professor, Kaikai Liu. All items purchased were sponsered by San Jose State University.