Making Simple Devices Smart:Alarm Clock Coffee

Abdullah Almaayoof
Department of Computer Science
The George Washington University
Washington, DC, USA
almaayoof@gwu.edu

Saif Alzaabi
Department of Computer Science
The George Washington University
Washington, DC, USA
alzaabi@gwu.edu

***Abstract*—Our IoT coffee machine starts brewing coffee as soon as the alarm goes off, using edge computing with a Raspberry Pi and ultrasound sensors. The sound of the alarm is recorded and sent to a server, which checks for a match. If a match is found and a cup is present, a GPIO signal is sent to start brewing coffee. After the coffee is made, the ultrasound sensor is used to measure how long it takes for the user to take the coffee and this data is used to adjust the next brewing time. Our device will save time for the user while promoting sustainability by reducing waste.**

***Keywords*—IoT, Edge Computing, Coffee, Ultrasound, Smart Devices, Smart Home**

I. Introduction

In a world where automation is becoming increasingly important, the concept of an IoT coffee machine that can make coffee as soon as the alarm goes off is a promising idea. With this project, we aim to create a coffee machine that uses edge computing with a Raspberry Pi and ultrasound sensors to automate the coffee-making process. Our project involves recording the sound of the alarm, sending it to our Flask server for sound recognition using our Fast Fourier Transform Algorithm, checking for the presence of a cup using ultrasound sensors, and sending a GPIO Signal to start brewing coffee. We make sure that the coffee is fresh and not wasted by measuring the time it takes for the user to take the coffee. And By using edge computing, we aim to reduce latency and increase the efficiency of the coffee-making process. Ultimately, our project aims to save time for the user while promoting sustainability by reducing waste.

II. Background and Literature Review

We are developing this product to address the lack of smart functionality in everyday life items. A coffee machine is an essential appliance for many people, and it is time to make it smarter. With our product, users will have a more convenient and automated way of brewing their morning coffee as soon as the alarm goes off. By leveraging the power of edge computing with a Raspberry Pi and ultrasound sensors, our coffee machine can streamline the coffee-making process and save time for the user. Ultimately, we hope to create a smarter, more efficient, and sustainable way to enjoy coffee at home.

A. Hardware, firmware or products found in literature search

For our project, we utilized a combination of hardware and software components. The hardware includes a Raspberry Pi, ultrasound sensors, microphone, and temperature sensor. Most of these items were utilized in our IoT & Edge Computing course at George Washington University, except for the coffee machine motherboard. We collaborated with the machine shop at GWU to solder wires to the coffee machine motherboard, enabling us to trigger the coffee machine from the Pi.

In terms of software, we implemented a Fast Fourier transform algorithm to differentiate between different sound files and match the correct file with the alarm sound recording[1]. This allowed us to accurately detect when the alarm goes off and trigger the coffee-making process.

III. Methods used for IoT device implementation

Our IoT device is designed to detect the sound of an alarm and automatically start making coffee for the user. It takes in audio input from the surrounding environment and uses an ultrasound sensor to check for the presence of a coffee cup. If the sound recording matches the tone of the user's alarm, and a cup is present, the device sends a GPIO signal to turn on the coffee machine and start brewing coffee.

For our project, we adopted the IoT 5-layer model (Fig 1), which consists of the perception layer, network layer, middleware layer, application layer, and business layer. The perception layer involves the use of sensors to collect data from the environment, such as the sound of an alarm and the presence of a coffee cup. The network layer involves the communication between devices, such as the Raspberry Pi and server, to process and analyze data. The middleware layer is responsible for managing communication between devices and services, such as GPIO signal that triggers the coffee machine. The application layer provides the user interface for the device and allows users to interact with the coffee-making process. Finally, the business layer focuses on the financial aspects of the project, such as profitability and marketability.

We chose the IoT 5-layer model (Fig 1) because it provides a comprehensive framework for designing and implementing IoT devices. Our product is especially relevant as many people rely on coffee to kickstart their day. By leveraging the 5-layer model, we were able to build a device that seamlessly integrates with the user's morning routine while promoting sustainability by reducing waste.

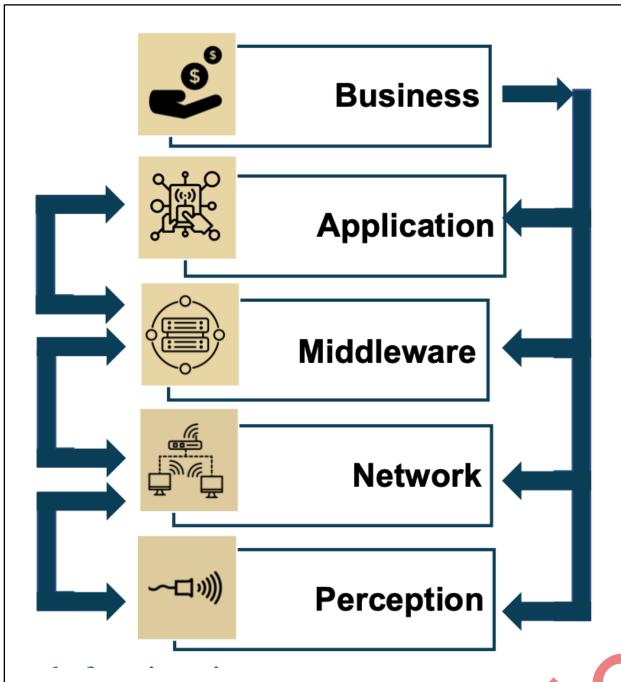

Fig. 1. IoT layer/architecture model we chose (Created by Professor Bulusu)

For the perception layer, we are using ultrasound sensors, a microphone, and a temperature sensor. For the network layer, we are using RESTful API calls between a Flask server in Python, an Android application, and the Raspberry Pi to process and analyze the data and decide whether to make coffee or not. The middleware layer is where the Pi communicates with the coffee machine to make coffee, using our RESTful API calls. The application layer is an Android application that users can access. Our product aims to provide a convenient way for coffee lovers to get their morning coffee without any hassle, and the coffee machine industry is highly competitive. Our product differentiates itself by incorporating IoT technology. Our business strategy includes marketing our product to coffee lovers who value convenience and sustainability. We plan to partner with coffee shops to promote our product and offer discounts to customers who bring their reusable cups. By incorporating sustainability into our business strategy, we hope to make a positive impact on the environment. Additionally, we plan to explore potential collaborations with coffee machine manufacturers to integrate our IoT technology into their products and offer smart coffee machines to consumers. This will not only generate additional revenue streams for our business but also promote the adoption of sustainable practices in the coffee industry.

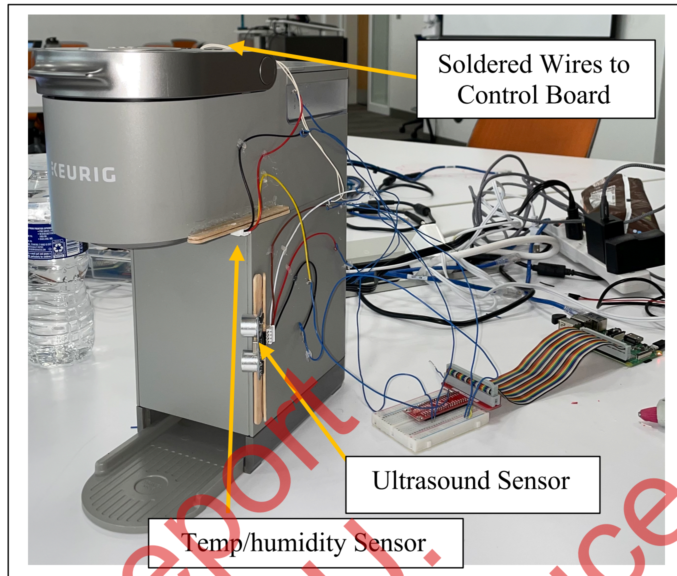

Fig. 2. Our IoT Coffee Clock Alarm Device

A. Sensors, Actuators, Microcontrollers and Single-board computer used

We are using Ultrasound sensors(HC-SR04), temperature/humidity sensor(TS0214D), and a microphone. Ultrasound sensors were chosen because they can detect the presence and proximity of objects such as the cup. We are using the microphone to detect the sound of the alarm, which has a specific tone. The temperature/humidity sensor was chosen because it can measure the temperature and humidity of the surrounding environment, which can be used to determine the temperature of the copy after its made. These sensors were selected based on their suitability for the specific use case and their compatibility with the Raspberry Pi platform.

B. Firmware development using Python and relevent libraries

1. Initialize the required libraries: Soundfile, PyAudio, SimpleAudio, sklearn and Librosa
2. Load the predetermined alarm sound file
3. Apply FFT to the alarm sound file
4. Select the most prominent range of frequencies in the alarm sound file (0 to 4000)
5. Start the audio stream
6. Take in 30-second intervals of the audio stream
7. Apply FFT to the intervals
8. Find the closest overlap of frequencies between 0 and 4000 with a step of 0.1 between FFTs
9. Calculate the mean squared error (MSE) of the frequencies in both graphs
10. If the MSE is less than 1.5e-7, turn on the coffee machine and make coffee for the user
11. If the MSE is greater than or equal to 1.5e-7, do nothing
12. Repeat steps 6-11 until the user turns off the device

C. Edge computing section

The edge compute component of our project is implemented using a moving average formula that can predict when the user will take their cup of coffee. This is done by using our ultrasound sensors to check how long it takes the user to take the cup after the coffee is made. And based on this information, the Pi applies the moving average formula to either delay or speed up the moment the coffee is made, so that the coffee is fresh when the user takes it.

This edge compute component is included in our perception layer of the IoT model. The perception layer is the layer that is responsible for gathering the data from its surroundings. In our project, we use the ultrasound sensors to gather information about if a cup is there or not, and if the coffee is made, how long it takes the user to take the coffee. This use of edge computing ensures that the data is gathered and processed locally, which reduces latency and improves the overall performance of our system.

IV. Discussion of your IoT Device and Results generated using the device

Initially, we planned to use a solenoid to press the coffee machine button to make coffee. However, after programming the solenoid and assembling everything, we found that it did not have sufficient power to press the button. As a result, we had to refine our approach and come up with a different solution. We decided to solder two wires to the control board of the coffee machine, which, when connected (i.e., closing the circuit), simulates the button being pressed.

To test the sensors, we calibrated them using a reference source and adjusted their parameters accordingly. We also conducted dry runs of our IoT device to ensure that all components were properly connected and communicating with each other. Additionally, we fine-tuned our final product by testing it in various scenarios and environments, making adjustments as necessary to optimize performance and functionality.

V. Conclusions, Summary, Future work

Through our work, we have concluded that the integration of IoT devices can greatly enhance the convenience and efficiency of everyday tasks such as making coffee. By using a combination of sensors, algorithms, and edge computing, we were able to create a smart coffee machine that is responsive to the user's needs and preferences.

One limitation of our project is that it currently relies on a pre-determined alarm sound to trigger the coffee-making process. However, in real-world scenarios, users may have different alarm sounds or may not use an alarm at all. To overcome this limitation, future iterations of the project could incorporate machine learning algorithms to detect and learn the user's specific alarm sound, or implement a voice-activated command system.

If our IoT device were to be scaled up or implemented outside of the classroom, it could greatly improve the efficiency of coffee-making in homes, offices, and other settings. To get there, we propose further development and refinement of the device, as well as marketing and partnerships with coffee machine manufacturers and retailers to incorporate this technology into their products.

In conclusion, the successful implementation of our IoT device has demonstrated the potential of this technology to revolutionize the way we approach everyday tasks. While there are still limitations and areas for improvement, we believe that with continued development and collaboration, smart coffee machines and other IoT devices can become a standard part of our daily lives.

Acknowledgment *(Heading 5)*

We would like to express our gratitude to our instructor, Professor Kartik Bulusu, for his guidance and mentorship throughout this project. We also thank Bakari Coulibaly, our friend, for his support and expertise in helping us troubleshoot technical electrical issues. In addition, we extend our appreciation to Jesse, the machinist at the George Washington University machine shop, for his assistance in soldering the wires we needed for our device. Without their support and advice, we could not have completed this project successfully.

References

[1] Sahai, Harshita. "Compare Two Different Audio in Python." OpenGenus IQ: Computing Expertise & Legacy, OpenGenus IQ: ComputingExpertise & Legacy, 21 Apr. 2022, https://iq.opengenus.org/compare-audio-python/

[2] Macdonald, Sandy. "Logging Raspberry Pi Environmental Data to Influxdb." Sandyjmacdonald,https://sandyjmacdonald.github.io/2021/12/29/logging-raspberry-pi-environmental-data-to-influxdb

[3] Maklin, Cory. "Fast Fourier Transform." *Medium*, Towards Data Science, 29 Dec. 2019, https://towardsdatascience.com/fast-fourier-transform-937926e591cb.