Smart HomeKit-Enabled Peephole Camera

Emmanuel J. Theodore , Brendan J. Gibbons, , Luke J. Heinzmann, Cameron J. Dore, Saurav Basnet
Program of Electrical and Computer Engineering
Wentworth Institute of Technology
Boston, MA, 02115, USA
theodoree@wit.edu, gibbonsb@wit.edu, heinzmannl@wit.edu, dorec@wit.edu, basnets@wit.edu

It can be difficult to satisfy the need for smart home convenience devices without invasive procedures that would go against living contracts in an apartment, dormitory, or any rented vacant living area. Creating a device that allows for convenience and security is imperative. The objective of this project is to develop and test a Smart Peephole Door Camera to identify who is at the door. A small, portable 3D printed enclosure that houses a Raspberry Pi, portable battery, and camera fit over a door's peephole on the inside of the door through a non-invasive installation process via 3M Command Strips. This allows the resident to conceal the camera from any outside visitors. The device connects to the user's home network via Wi-Fi to allow access to Apple's HomeKit API to enable remote viewing from a compatible iOS device. In addition, an open-source platform provided by Apple on GitHub with the addition of the HomeBridge lightweight NodeJS server allows seamless integration between the smart peephole camera and the user's iOS device. 3D CAD files created on SolidWorks and Shapr3D were printed and tested through multiple design iterations to ensure that the design of the enclosure and the mounting methods meet the fundamental constraints of this project.

Keywords — HomeKit API, 3D Printing, SolidWorks, Raspberry Pi, shapr3D

I. Introduction

Video doorbells have been on the smart home market for a while now, and they are just starting to take off. Many consumers are interested in adding an extra layer of security to their front door with a video doorbell. While some consumers are homeowners and can make modifications to their front door, those who rent apartments and live in student dormitories are missing out due to restrictions on what they can change about their rented housing. The need for this project is to develop a similar product to a video doorbell that can be used in spaces that are not owned by the consumer. There are a few alternative options available on the market, although they may have an invasive installation process, or be missing certain features. The ideal design for this project is a camera that fits over the peephole indoors, can easily be removed for maintenance and recharging, as well as being implemented into Apple's HomeKit smart home platform for remote viewing.

There are already a few products available to the public for added security on a consumer's front door. For example, there is the Ring doorbell, which is mounted in place of a doorbell and features a video camera that records activity and notifies the user of visitors. In addition, modifications for renters and students have been made available, such as the Brinno Duo camera or the Ring peephole camera. While some of these devices meet the primary need of being installed on the inside, modifications to the door are still required. One modification that the Ring peephole camera requires is replacing the door's peephole with a provided lens, as well as needing specific door and peephole dimensions for the device to be compatible. In addition, the Brinno Duo mounts indoors but does not utilize any smart home platforms for the user to remotely monitor their door.

While the Ring and Brinno devices partially meet the constraints of this project, they do not go all the way. The Ring devices still have invasive installation requirements, and the Brinno Duo does not allow for remote monitoring using HomeKit. These devices are similar in functionality, but the project is tailored toward students and renters who cannot modify their door at all and want the remote connectivity that a HomeKit enabled camera offers. One primary reason for HomeKit connectivity being the main selling point is the human-centered design. Those with disabilities can utilize the group's device to make their home more accessible even if it is a simple fix such as checking who is at the door from anywhere using any Apple device allowing for control of HomeKit devices.

The paper is organized as follows. A design overview will mention the components used in this project, while hardware and software sections will highlight the process of designing the 3D-printed parts and software, respectively, needed to add the functionality of an Internet of Things (IoT) device. The results section will showcase how the interface looks on a device, and conclusions, further endeavors, and acknowledgments will be discussed.

II. Methods

A. Design Overview

Hardware and Software played an integral part in the design of the *Smart Peephole Camera*. To maintain cheaper manufacturing costs, 3D printers will be utilized for the non-electrical components, and open-source software will be modified to allow the electrical components to interact with each other wired and wirelessly. Aligned with the constraints of the project, the device must attach to the door through non-invasive measures and must be easy to install.

A Raspberry Pi (3 B+, raspberrypi.org) is the computer for the *Smart HomeKit-Enabled Peephole Camera*. The Raspberry Pi uses a 150-mm, 15-pin ribbon cable to connect the camera (#B07SN8GYGD, Arducam) to the Raspberry Pi. A 12,000-mAh portable battery pack was connected by a Micro-USB to USB-A cable from the battery to Raspberry Pi to power the Raspberry Pi and camera.

3D-printed components include an open-source design Raspberry Pi enclosure, which was printed in two parts that snap together to hold the Raspberry Pi; an attachable camera enclosure to secure the camera in place; and a door mount to secure the assembly to the door.

B. Fabrication and assemble

The Raspberry Pi is a very complex piece for an enclosure design. It has multiple dimensions due to the different heights of each port and circuit component. Also, since the Raspberry Pi is constantly monitoring the live video feed, the enclosure needed to have an adequate heat sink so the Raspberry Pi would not overheat. With all of this in mind, the group decided it would be easier to find an open-source design for the Raspberry Pi enclosure. The other 3D-printed components were designed by the group.

The camera has two 3D-printed parts. The first part is an attachment that houses the camera through a fiction fit. It has a cutout for the camera where it slightly protrudes, and it has an area to fit the ribbon cable (Fig. 1).

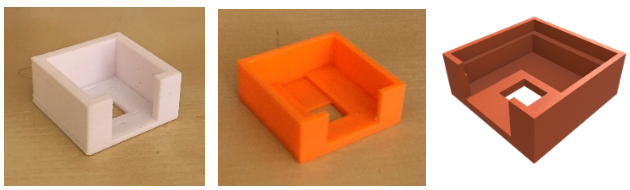

Fig. 1. Camera Enclosure. This is the design evolution (from left-to-right) from the first prototype to the final design. The camera fits into the width and length of the small box inside the enclosure. The height of the camera enclosure has decreased, and the opening for ribbon cable has increased. An inner, thicker shell has been added in the final design to obtain a tighter fit for the camera.

Once the camera is placed into the camera enclosure, it snaps onto the bottom of the Raspberry Pi enclosure. The camera has a larger ribbon cable, so the camera must fit around the top of the Raspberry Pi and halfway down the other side of the bottom Raspberry Pi enclosure. A component was added to the open-source design to allow for the camera to attach to the housing (Fig. 2).

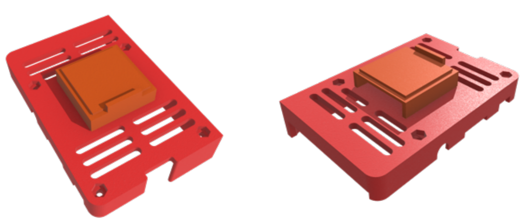

Fig. 2. Camera Enclosure Mount (orange). This is a top-side view (left) and a bottom-side view (right) of the camera enclosure mount that allows the camera enclosure attachment to snap onto the camera enclosure mount. It is attached to the bottom of the Raspberry Pi enclosure. To achieve the snapping locking mechanism, the length and width of the camera enclosure was designed to equal the length of width of the camera enclosure mount.

The door mount has many considerations in its design. It must fit the common diameter of a peephole. To account for 3D-printing errors, the diameter of the peephole must be 17 mm including tolerance; on the higher side of the 12 mm to 19 mm range of the average-sized peephole [1]. The door mount must hold the rest of the assembly and components, and the center of the circle on the door mount must align with the center of the camera on the attachable camera enclosure. Extensive measurements were conducted, and the prototype was drawn (Fig. 3).

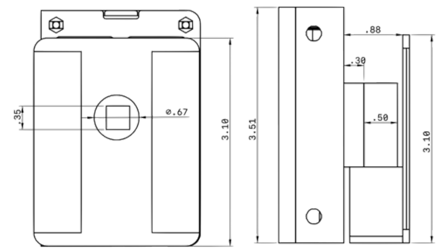

Fig. 3. Initial prototype and dimensions of the back-view perspective of the door mount (left) and side-view perspective of the assembly (right). The door mount contains an opening for a peephole, a square design, and an area for the Raspberry Pi enclosure to attach to the door mount. The assembly consists of the door mount, camera enclosure, camera enclosure slot, and an open-sourced Raspberry Pi enclosure to house the Raspberry Pi.

The group had to think of an innovative way to attach the *Smart HomeKit-Enabled Peephole Camera* to the door. The door mount, once created, was reduced in size and allowed for 3M Command Strips to add seamlessly in the design. The approximate dimensions for the average-size command strip were added, and the approximate depth of the command strips were considered so the command strips sit flush with the surface of the door mount (Fig 4).

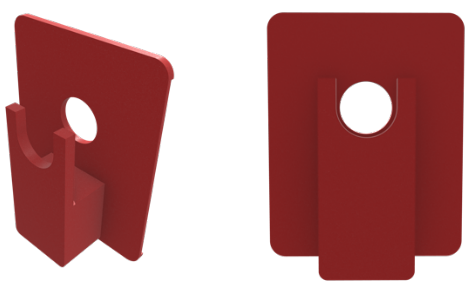

Fig. 4. Front-side view (left) and back-view (right) of the Door Mount. The command strips fit on the left-and-right side of the door mount to easily align the door mount with the peephole.

All the constraints were addressed, and the final design of the assembly was created (Fig. 5).

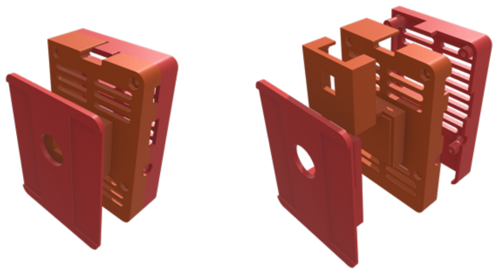

Fig. 5. Final design of assembly (left) and expanded view showing each component of the assembly (right).

C. Software Design

We used SolidWorks and Shapr3D in combination with 3D printers to prototype the modules of the design. SolidWorks and Shapr3D allow us to create the mockups and print them using a 3D printer.

There are many ways to create the software needed to make the smart peephole work. The best way would be to buy a developer licenses from Apple and use the HomeKit API (Application Programming Interface). This API is in the Swift language and would sit on the Raspberry Pi's software layer as shown in Fig. 6. As we did not have the resources to acquire this at the time and due to COVID-19, we used an API on GitHub instead. Homebridge is a lightweight NodeJS server that emulates the iOS HomeKit API used to create HomeKit compatible devices. Homebridge relies on languages such as GO, C, JavaScript, and Jason and affectively achieves the same HomeKit integration with no extra steps for the user.

The Camera module on the Raspberry Pi communicates via the Raspberry Pi to the HomeBridge/HomeKit API. This translates the video to the Home App on the iPhone which allows you to see the video. As seen on Fig. 6

The Raspberry Pi connects to Apple's HomeKit API to allow the camera to work with Apple's Home application on iOS and Mac devices.

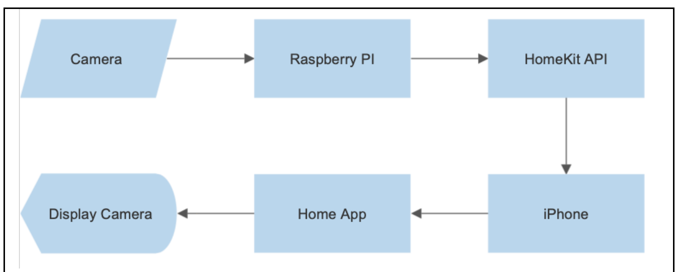

Fig. 6. Software Flow Chart. The hardware uses Apple's HomeKit API to connect to the iPhone over the internet through Apple's dedicated Home Application. The camera feed is displayed as a live stream through the user's device.

It is important when developing a web-enabled camera, especially one that will be in your home to have the most extreme security. As we are leveraging the Apple Ecosystem with home kit interaction, we can leverage Apple's HomeKit Secure video or HSV to securely and locally encrypted video stream that gets processed by Apple Hardware such as HomePod, Apple TV, and iPad devices before getting sent to iCloud for data storage and retrieval. As a result of this implementation, the encryption keys are stored locally on Apple devices with dedicated hardware security measures like secure boot and secure enclave to ensure integrity, confidentiality, and availability.

D. Fabrication and Assembly

The software utilized for this project was open-source software provided by Apple on GitHub. This software was loaded onto the Raspberry Pi to turn it into a HomeKit Home Hub. The camera attached to the Pi was then recognized as a smart home device and was able to be added to the HomeKit app using the same process to add new HomeKit certified accessories. In addition, the enclosure for the device was designed around the Raspberry Pi, 3D printed and assembled around the Raspberry Pi.

E. Testing Procedures

The various prototypes were tested by printing various iterations of the enclosure and determining which worked best for the overall design. The group used the on-campus 3D printing lab to rapidly prototype the different enclosure models for the project. In addition, the software installed on the Raspberry Pi was tested by adding the device to one of the group member's iPhones and ensuring the camera streamed to the device seamlessly through the Apple Home app.

III. Results

The design was successful in meeting the needs for a toolless installation on the inside of the door. Making use of the Apple HomeKit allows for seamless integration with the Apple Home Application. The device is concealed from the outside of the door. The use of HomeKit also allows for the Smart Peephole Camera to work with other smart home devices.

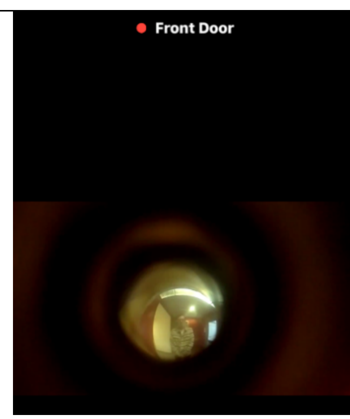

Fig. 7. Apple's Home Application showing Live View of Camera. The camera is showing a view of what the camera sees through the inside of the door's peephole. This view, from the infancy stages of the project, would have likely improved through the constant revisions of the door mount and camera enclosure.

IV. Conclusions and Future Directions

For this project, the group met the main goal which was to create a HomeKit enabled camera that is mounted on the inside of a door and utilized to monitor visitors. The camera has basic functionality where a livestream can be accessed utilizing a HomeKit enabled iOS device.

Throughout the duration of completing the project, the group came up with a variety of prototypes for the 3D-printed enclosure for the device. After testing each prototype, the group determined which changes should be made. The design was then updated and reprinted for additional testing. The assembly is significantly cheaper than the devices that are on the market today. The Raspberry Pi, camera, battery pack, and 3D-printed components are estimated to cost 50% less than the alternative devices sold on the market with a $199+ price tag.

While the main goal of the project was met, the group plans to add additional features to the device in the future. The group received additional funding from an advisor to purchase an Apple developer account. This additional funding will allow the group to implement additional features such as communicating with other HomeKit devices, storage of recorded video, or even two-way audio communication between the user and visitors at the door. In addition, the 3D printed enclosure would be adjusted as needed to provide a more streamlined design and decrease the overall cost of printing the enclosure. Some considerations around malicious use was explored, specifically around using the product on the outside of the door facing inward with a modified lens. Research into this led us to some issues that would require resolving before this modified version could be implemented. For one, the Raspberry pi, while generally energy-efficient still will require frequent recharges if not plugged in. As well as the HomeKit integration relies on Wi-Fi antennas that have a limited range. These issues combined with the likelihood of the victim opening their door and realizing a device had adhered to it untimely left us satisfied. Such modifications would become a new device internally and someone with the skillset to make those modifications could do so without this peephole project.

Acknowledgment

Thank you to Dr. Douglas Dow for providing the Raspberry Pi used in the prototyping of the Smart Peephole Camera. Free 3D Printing courtesy of the Wentworth Technology Sandbox.

References

[1] "PEEPHOLES & DOOR VIEWERS COMPARISON," *peepholesecurity.com*, 2009. [Online]. Available: http://peepholesecurity.com/Viewercomparison.pdf. [Accessed: 21-Apr-2020].

[2] S. Davidoff, M. K. Lee, C. Yiu, J. Zimmerman, and A. K. Dey, "Principles of Smart Home Control," Lecture Notes in Computer Science UbiComp 2006: Ubiquitous Computing, pp. 19–34, 2006.