A Framework of an IoT Testbed

Chad Workman
Electrical Engineering Department
University of Arkansas
Fayetteville, AR, USA
cmworkma@uark.edu

Gavin Tomlinson
Electrical Engineering Department
University of Arkansas
Fayetteville, AR, USA
gt012@uark.edu

Jia Di
Computer Science and Computer Engineering Department
University of Arkansas
Fayetteville, AR, USA
jdi@uark.edu

Xiaojiang Du
Electrical and Computer Engineering Department
Stevens Institute of Technology
Hoboken, NJ, USA
xdu16@stevens.edu

Qiang Zeng
Computer Science and Engineering Department
University of South Carolina
Columbia, SC, USA
zeng1@cse.sc.edu

***Abstract*—As Internet of Things (IoT) technology becomes more widespread and commonplace in homes， the efficiency of these devices using available bandwidth is becoming more of a concern, as the number of connected devices in a home increase drastically. If each device is controlled using a separate Application Programming Interface (API), the strain on a network will be much worse than it would if all these devices are controlled from a single point. This single point could handle all commands to and from the devices, thereby decreasing the network load. The framework of a testbed presented in this paper will allow developers to build an API around the devices included in the testbed. Then test their algorithms and other research methods from a remote location.**

I. Introduction

The Internet of Things (IoT) is a network of objects with sensors and the capability to connect to the internet. These devices communicate amongst each other to share information and their current state, as well as taking commands to perform certain actions. This technology has been quickly making its way into people's homes and many other places. With the drastic growth in the number of these devices for a variety of different commercial premises, there has been a push for a more interchangeable and reliable control system. Currently most devices meant for households have their own APIs and are controlled separately unless specifically purchased around a single control system.

Companies also use IoT devices to control their systems inside of manufacturing plants. Recently there have been advances in edge computing in IoT devices used in business settings. These devices are more similar to a network of computers rather than a single computer with many sensors. They are able to handle information on their own and send less data to the main control point, freeing up bandwidth for more devices on the same network [1][2]. This form of control network is seen as ideal for performance. However, this type of system is unreasonable for a household where the number of devices being used at the same time is much less as compared to an enterprise setting. The best control scheme for home is usually from a single access point, so that all devices are connected to the main API and can be seen and accessed easily from a remote location during their use. This scheme is much cheaper and simpler to install in homes. As edge computing is getting into households, this testbed being developed is a valuable tool for educational and development purposes where a single control point will perform adequately.

Currently many devices are controlled via an app that is made specifically for that device. This would lead to smart homes that require the user to have a different app for each individual device they own. This is not convenient, and the devices will not work effectively together as a smart home due to the disconnected nature of individual APIs. The testbed presented in this paper will supply a physical environment for APIs to be developed to control devices that do not work together currently. The array of devices provided will each have different control schemes, which will be used as a basis to build upon. Some of these devices will be controlled directly through a microcontroller. And others will be controlled through their API, while the main API will just act as the user inside of the app. In this way the user will have access to all the devices from a single point. This API would also allow simpler devices to be added.

II. Testbed components

The purpose of this testbed is to create an emulated smart home to test and evaluate how IoT devices communicate to help smart home users. Currently the focus is to create a testbed for remote users to use in the future with straightforward API development.

A smart home should hold devices that reduce the complexity of daily life. It is necessary to focus on testing devices that accomplish this goal. The devices selected for the testbed mainly center around automating simple yet required tasks in a home. These tasks would include locking/unlocking a door or turning off and on appliances. To simulate human movement through the testbed, developer-controlled robots will

978-1-6654-0652-9/22/$31.00 ©2022 IEEE

be deployed. The following lists details of these IoT devices and robots.

A. LoCoBots and Turtlebot

There are three rolling robots used in this testbed; two LoCoBots and one Turtlebot. The LoCoBots and Turtlebot are robots designed at Carnegie Mellon University and were purchased from Trossen robotics [3]. These robots sit on a wheeled chassis which is controlled by an Intel NUC computer running on Linux 16.04 [4]. To control these robots, the ROS and Pyrobot APIs are used. These interfaces allow the robot to sense and avoid obstacles as well as manipulating objects using the 5 degree of freedom grasping arm.

The purpose of these robots is to emulate human movements in the testbed environment. The API developer will access the robots via a server connected to the testbed Wi-Fi. This connection to the robots over Wi-Fi will be hosted through SSH and RealVNC, which will give the developer access to the Intel NUC on each robot. This method will also allow the developer to manipulate the testbed environment via the LoCoBot and Turtlebot arms.

B. Teckin Smart Plug

A smart plug is a very simple device used to turn on and off other devices. This smart plug shown in Figure 1 uses a microcontroller connected to a relay to control the higher voltage power supply. This relay is in turn controlled by a transistor and voltage divider circuit. By connecting the I/O of an ESP8266 microcontroller to the transistor circuit, it is possible to actuate the relay and control the smart plug [5]

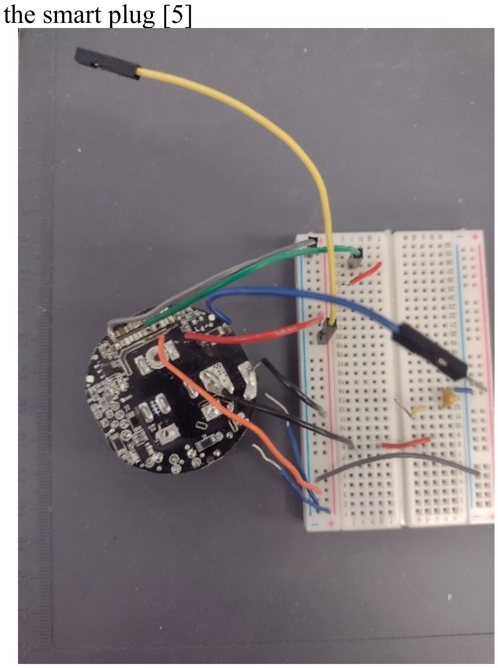

Fig. 1. Teckin smart plug test connection

The Teckin smart plug is a useful IoT device as it allows the developer to control non-networked appliances. The design of the smart plug is simple which makes it an ideal candidate for testing. These smart plugs were purchased as a pack of four which presented the opportunity to use one smart plug with the factory API and to modify another electronically to accept direct inputs. This will enable the developer to expand the capabilities of the testbed API by presenting different applications.

C. August 2nd Gen Smart Lock

The August smart lock is a device that takes the place of a conventional lock allowing for remote locking and unlocking of doors. To integrate this device into the network, an ESP8266 microcontroller was used to control a H-bridge motor driver as seen in Figure 2. This method allows for the user to precisely control the rotation speed and position of the smart lock [6].

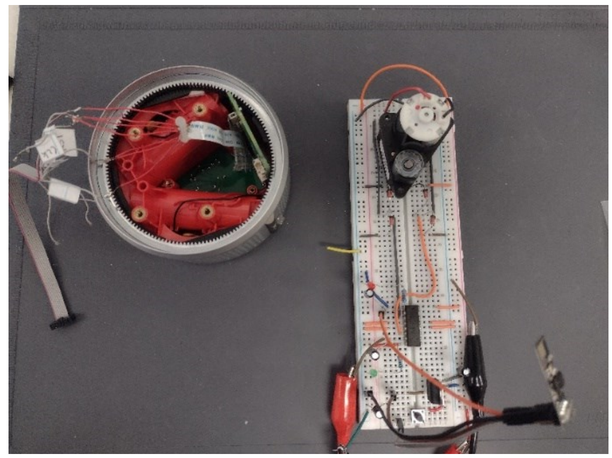

Fig. 2. August smart lock housing (left) lock motor control (right)

A smart lock will be a useful addition to this testbed as security is a common application of IoT devices. The August smart lock currently connects to an app on the homeowners' phones or can be networked with most other ecosystems. For this testbed two smart lock devices are included. Similar to the smart plug, this will allow the testbed to have a controlled smart lock with the original August API as well as a way to test the user API on a similar but much simpler device.

D. Wyze V2 Smart Camera

The Wyze V2 Smart Camera shown in Figure 3 is an affordable Wi-Fi accessible smart camera with its own control and streaming capabilities. The original functionality of this device does not allow for sufficient control over the camera feed as the only way to see the video is through the app. To make the video RTSP capable, the camera would have to use a different control system.

Fig. 3. Wyze V2 Smart Camera

The different control system came in the form of a new OS launched from the SD card slot. This slot was used during the manufacturing process to run testbench code and make modifications to the OS without having to reprogram the internal board. This means that that camera already has the capability to launch from the SD slot and a separate OS can be loaded onto the camera to offer the full functionality needed [7].

The added functionality allows the camera's program to be modified remotely. This will provide users with the capability to configure settings to their liking to fit the user API. The livestream output is now in a format that can be fed into a control algorithm used on the NAO robot described below.

E. *Robot lab NAO*

Fig. 4. NAO Smart Robot

NAO is a robot produced by SoftBank Robotics built for use in education and research, as shown in Figure 4. NAO is able to mimic the movements of a human around the smart home. Using the Wyze cameras, NAO will be able to detect its surroundings and move about the test area as a human would. During development of the API for the testbed, there will not always be a person to move about the testing area; therefore the NAO will fill that role and be able to trigger devices as it roams around.

NAO will be able to see its surrounding via the cameras located on its head and the livestream from the smart cameras like Wyze. NAO is controlled through a Python script and this control scheme will be accessible to the developer to change its behaviors [8].

As the development of this testbed progresses, there will be other devices connected to the ecosystem such as smart doorbells, thermostats, motion detectors, etc. Establishment of a foolproof method for connection is necessary before expanding the testbed, which is why these simple devices will be connected first. The next section will cover the implementation of this testbed.

III. Implementation

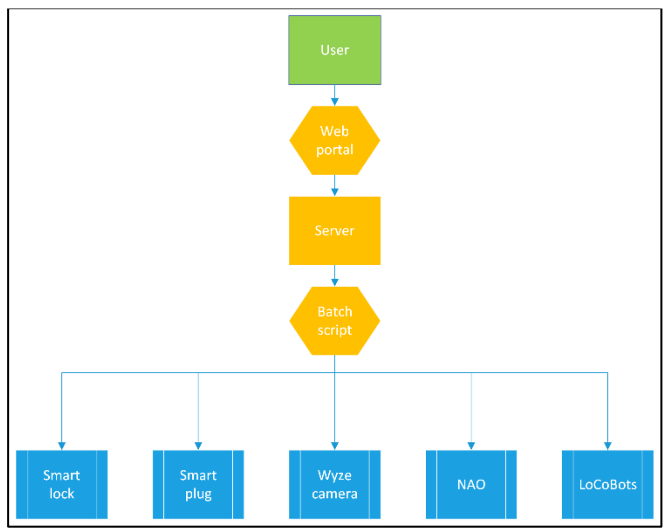

Fig. 5. IoT testbed ecosystem diagram

The architecture of this testbed is depicted in Figure 5. As discussed above, the purpose of this IoT testbed is to emulate a smart home environment so that an API can be created to increase connectivity and improve home services. Since this API is for external researchers and users to implement their algorithms and theories, this testbed is required to be accessible over the internet. To meet this requirement, a server and dedicated device network will be used. The user will access the server via a web portal and conduct the API research and development on the server.

The current framework of this testbed allows for more complex devices with proprietary APIs to be connected along with basic devices like the smart plug. The capability to have API enabled and non-API enabled devices on the same network will allow the user to create a versatile IoT API.

Once finished, the testbed will be a small environment with all of the devices located around the room in a similar way they would be in a smart home. There will be a charging station for NAO and the LoCoBots for periods when the testbed is not in use. Smart cameras will be placed in the top corners of the room looking down so the developers can see what is happening in the room and also for the NAO to see its surroundings. The room will be sectioned off with obstacles dividing the room into several regions where different devices will be stored. Each device along with the server will be connected to a single router. This will allow all of the devices to communicate with the developer from a remote location.

IV. Conclusion and Future Work

As previously stated, most of these devices had to be modified in order to connect to the server and give full control without the device manufacturer's own control system interfering. By completing this testbed, an API will be able to be developed which can configure a device being connected and be controlled from a single point easily, rather than console commands and running batch scripts. This form of single access point API will suit the needs of smart homes much more effectively than what edge computing does at this time. Development into the intercommunication of these devices will allow unassociated devices to work more seamlessly together, thereby enabling remote researchers to implement and evaluate their algorithms and theories. Future work includes:

- Server implementation and connections with all of the devices – setting up the server in Linux and making sure it can hold a stable connection with all of the devices in the system. This server will be the access point for remote developers to work on and store information between sessions;
- Ensuring all devices are operational with instructions on how to use the device if the control system is complicated or outdated;
- Room and obstacle course set up – this should be easily modifiable depending on the needs from the developer, e.g., the room can be divided up in a way for the NAO and other robots to navigate; and
- Stable livestream feed and functional obstacle detection from the livestream – this will be done in Python and will be used to guide the NAO bot around the room through multiple Wyze cameras.

Acknowledgement

This work is sponsored by the National Science Foundation Award CCRI-2016485.

References

[1] "Intelligen Power: How our IOT thinking advances power management," *Eaton*, 01-Jun-2019. [Online]. Available: https://www.eaton.com/in/en-us/company/news-insights/internet-of-things.html.

[2] A. K. Gupta and R. Johari, "IoT based Electrical Device Surveillance and Control System," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 2019, pp. 1-5, doi: 10.1109/IoT-SIU.2019.8777342.

[3] "LoCoBot – an open source low cost robot," *LoCoBot*. [Online]. Available: http://www.locobot.org/.

[4] "Locobot (PyRobot)," *ROS enabled research platform with 5 DOF manipulator*. [Online]. Available: https://www.trossenrobotics.com/locobot-pyrobot-ros-rover.aspx.

[5] The4, "Teckin SP10 Smart plug," *Teckin US*. [Online]. Available: https://www.teckinhome.com/collections/smart-plug/products/teckin-sp10-smart-plug?variant=41479981433054.

[6] "August® smart locks," *August Smart Locks*, 10-Aug-2021. [Online]. Available: https://www.augustsmartlocks.com/.

[7] EliasKotlyar, "Eliaskotlyar/Xiaomi-Dafang-Hacks," *GitHub*. [Online]. Available: https://github.com/EliasKotlyar/Xiaomi-Dafang-Hacks.

[8] "Nao the humanoid and programmable robot: Softbank Robotics," *SoftBank Robotics - Group*. [Online]. Available: https://www.softbankrobotics.com/emea/en/nao.