UNIVERSITY OF KENT

Smart-Life

CO600 Group Project

Alexander Collins

Waleed Iqbal
Jung-Moo Lee
Igor Drobnica

Smart-Life

Alexander Collins Waleed Iqbal
Jung-Moo Lee Igor Drobnica

Abstract

The project we decided to embark on creating was an "Android Controlled Home Assistant". This system is similar in concept to the Alexa series of virtual assistant created by amazon and other competing brands, ours however differs in the respect that instead of voice-controlled activation being the only way to use the system, our system is controlled by our android application and voice control making It more flexible to the user. The hardware we have used is the Arduino, provided to us alongside other parts we ourselves have procured, using this combination of software and hardware our system can perform a variety of functions in the home such as temperature control, locking and unlocking doors and a variety of many other useful functions. Our device is also not limited to one house per person and a user can have a variety of these device installed in many different locations all selectable from our application. via our database which stores all the data necessary to allow the user flexibility in what they wish to do with the device.

1. Introduction

Interest for home assistants has increased over the years especially since voice recognition software has become easier and cheaper to produce, because of this many large companies have created a version of such a system, they all have one thing in common however, they store recorded data of voices around them, and some people do not like such a breach of privacy no matter how convenient the system is to use.

Because of this and with the aid of android technology and reliability we embarked to create a home assistant of our own custom design, one capable for filling this need in the market, our system uses a combination of the Arduino hardware board and its components and a simple android device to create a cheap affordable and most important reliable home assistant that can be controlled from your phone without the need of voice commands, in a way a form of universal remote control for your house, within this document we will explain the many features of the device and how they came to existence and even more how they work.

2. Background

The universal remote control is a concept that has existed all the way back to the 1970s and ours works in a similar idea but instead of small electronics and TV ours can control your house functions when correctly hooked up, this is done via the Arduino circuit board and upon installing this device into your house in a variety of ways you can then control these functions remotely via your smartphone.

The innovation that was the smartphone has become so ubiquitous currently that by creating an application that controls our home system onto the smartphone is both cost efficient and easy for users to understand making it far superior to other methods of control.

The idea of the smart home system idea was seen by many as a science fiction creation back in the 1970s and as such beyond the average person until now, with the combination of smartphone technology and printed circuit boards we can create this technology for all to use in a n affordable package.

Many commercial home systems use voice recognition technology to be both efficient and convenient, however this comes with an ethical concern, as such we do not use it for our system making it cheaper and more friendly to users young and old alike.

3. Aims

Our aim was to build an android application that links to an Arduino board within your house and from your application you can control every facet of your home from locks to alarms.

The system should allow users to control locks, set intruder alarms and control temperature to you home via all the onboard functions, these included a weather API to be able to see the latest weather forecast for your area.

To help the user better control their home we aimed to make our application easy to understand for the non-technical minded user so that our system is accessible to all.

4. Requirements

4.1 User

We came up with a few user requirements while we were on the prototype phase. As every system, it needs to have a user-friendly interface, with which the users can browse with no difficulty. With regards to the system itself we needed the user to be able to easily user the device as one would use a remote control but at the same time take security into consideration, we needed to juggle the ease of access the user will expect with the need for some form of verification, this will after all control your house locks, we eventually through user stories and debating among ourselves to go with a pin style security to make it easy to login each time and also remain secure.

4.2 Security

The issue of security was paramount as the user would be putting faith in our application to secure their home and its belongings, because of this we opted for a binary data file to be created on our server when the user inputs their details for registration, they will then be asked to create a pin to use henceforth, after this is done the user will only need to input their unique pin for access keeping their details and control remote secure.

The binary file is used to prevent hacking of the system it is both simple and reliable at this task.

5. Design

5.1 Initial System Design

Our initial design concept did not change much throughout the project, we decided upon android due to its ease of use and ubiquity and we had no knowledge on working with apple products, we decided upon using the dragon server due to needing a reliable and easily available storage for our database, the main change of design throughout the process was how we implemented the login details for users in this database within dragon, this was because we required help to access the server due to security concerns raised by the university, we did however work around this by encoding our login data into a small file which is generated upon registration.

5.2 Final System Design

Functions remained unchanged and only server issue where a major problem to be overcome, the final system design involved mainly functions and what we could include in absolute and what we could include if we had spare time.

The design also swapped regarding how best develop the interface with the initial design deemed to complicated and was then remade to be easier to understand and use for the user, this included new widgets being created for the application as our previous ones where too ambiguous.

5.3 Language and Environment

The program is coded using a combination of languages such as MySQL and java as well as c++ we also used the weka library, we used the android development kit to create our android application and eclipse to create the functions we needed in java for our Arduino side of the project, this was all tied together using MySQL in order to control the database and servers we used in the project, we used eclipse as we all have knowledge of this IDE and we had to learn how to use the android development kit from scratch but overall no problems was encountered in this regard making it easy for us to transition between languages and implement them all together.

6. Implementation

6.1 Servers

We used the server that is provided to us by the university to perform many of the functions that are needed to make this project a reality, we had issues at first attempting to access this in the way we wanted due to the security on the server that we could not change, however we did eventually find a work around with the help of our supervisor by putting some files within the server that our application could access, from there we could then use the server as we needed and with minimal lag in input.

The server was then used to create and store or database and hold users' details and other important data that could not be forgotten, the sensors used the server to slingshot data back to the phone at 10 second intervals allowing for a good use of the burglar alarm sensors and temperature control, we used ESP8266 as a network gateway between our hardware and phone application, because of this we have created a way to ensure the network is both secure and reliable.

Our Server uses http protocol, and as such we needed to use an API key to authenticate calls between the system, we used this to prevent any security issues via our server.

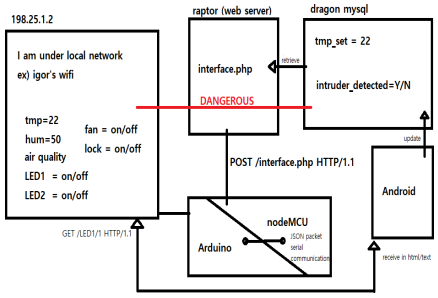

Figure 1 – How our server's function

6.2 Weather API

Our android application uses a weather API called "openweathermap" this API was free to use, and we found it both very reliable and well suited to the function we required of it, we use this for our current weather screen found inside temperature control on our application and works using a inputted postcode that the user can change via the setting page if the wish to do so.

The major advantage to this API over others we have seen is that it was easy to implement into the current scheme of our application and as such we did not need to do any elaborate changes to get the weather to work inside the temp page, because of this we could increase productivity by concentrating on other aspects that needed our attention such as the server implementation which was much harder to deal with overall.

6.3 Hardware

The Arduino hardware was the other side of our project and was vital to its success, we needed to ensure that implementing the software into the hardware was seamless and reliable to make sure that we could proceed with anything else.

The major issue with our hardware was power requirements, the functions we were asking for the Arduino to perform required more power than the board itself could provide, because of this we needed to attach an external battery to the hardware to solve this power issue, this turned out to work incredibly well and solved our power issue.

6.4 Locks

The ability to control the locks in your house is one of the fundamental aspects of this project and as such we put some time into making it very reliable and strong regarding its function, the Arduino is equipped with standard locks that can be fitted to doors and are controlled via the application.

The locks are not limited to a certain amount as you can add as many locks to the board as needed and all you need to do is increase the power supply which we made easier to do via small AA batteries so increasing in locks and power are cost effective.

The applications control itself are very easy to use and reliable, once you select the lock function from the menu you can switch which locks to control via the swiping left to right function, once selected you simply need to press the lock and unlock button for it to work, the delay is on average 3 seconds which is acceptable in this regard.

6.5 Primary Functions

The functions we added to our application are varied and useful for the average homeowner, these include:

Temperature control is used to allow the user to not only monitor the temp of their house but also adjust it to their liking, this is done with a combination of sensors to sense the temperature and a heat pad to heat up as well as a fan to cool down based upon what the user requires.

The way this works is that if the user wishes to cool down the room, they need only slide the bar in the temp control page to the corresponding temp they want, and the application will turn the fan or heat pad on or off based upon requirements that are needed to make this happen.

Upon reaching the correct temp required the sensors will monitor constantly for any changes and adjust and act regarding the feedback it receives, this makes the system completely automated for the user and works in the same way regarding air quality.

Air quality works in the same way regarding in principle as temp control does, in this case the sensors monitor air quality and return a value for the user to understand within the air quality page, after doing this the user can decide if they wish to activate the fan on the Arduino to improve air quality via the fan button which is toggled on and off from this page.

Intruder alarm is a very important function for our home software and as such we spent a lot of time deciding how best to implement such a feature, eventually we decided on using the sensors built of the Arduino to detect movement and feed this information to the user's phone via the application.

The user accesses the controls for this alarm in the application via the alarm button, once on here the user can select the alarm they wish, this only shows if they have multiple Arduino alarm for example, once selected the user needs to only push the alarm on button and the system is armed within seconds.

Upon being armed the system will monitor all movement via the sensors until triggered or turned off via the alarm control, once triggered the system will notify the users phone via a notification pop-up and will give the user the option to call the authorities or the user can ignore this, it is up to the user as the system will not automatically call the authorities for legal reasons.

Voice navigation system has been implemented to allow the user to navigate and use the functions via their voice if they wish to do so, they need to simply speak into the phone after pressing the voice command button and it will perform the function based on what they said, if will ask the user to confirm before doing this however to prevent mistakes.

Upon selecting the voice control function in the menu, the user can then press the voice command button and speak into the phone, if they say lock for example, they will be taken to the lock control page, they will then be able to control the locks as normal.

Voice control is not usable to activate the locks and other devices functions however as that would require storing a user's voice to improve recognition and we decided against that for ethical reasons, no personal data is stored whatsoever except a user's email and password.

6.6 Deep learning Features

The deep learning features of our application was implemented for the express purpose of improving certain aspects of our output to the user, for example when the user decides to lower the temperature via the temp control page the algorithm will use the data gathered from this input to determine what the desired average temp is for the user, a form of machine learning built into the back end of our software.

This was a very infant version of the deep learning feature we had hoped to introduce in the beginning of our project but due to time constraints and technical constraints we had to settle for a simple and minimalist deep learning algorithm.

This version we have installed in our project is limited and does not affect most of the functions of our project, because of this we had no need to inform the user of its results as it is all backend functions being used and no input from the user is needed.

A holdover of our plans for the deep learning feature can be seen by going to the settings page which shows we had more planned for it such as a greater learning ability to also predict what air quality the user prefers, and the system could adjust accordingly.

The deep learning feature uses standard algorithm techniques and was made to ensure that the feature does not hinder with the simple approach to control we wanted for the house system, we used the deep learning as a test bed early on to experiment with more complex techniques and overall are happy with what we have come up with.

The advantage to our deep learning algorithm is that it can easily be upgraded and improved with no problems if given enough time, due to this any future expansion of this project would be seamless and easy to implement.

6.7 Deep learning Results

The deep learning features are completely automated and as such the user does not need to interact with them, because of this we have not provided a way for the user to view any of its results and it remains a background running feature.

7. Testing

7.1 Software Testing

We ran testing of our software at every new addition to it, we did this via the agile methods taught to us last year and as such we encountered very few bugs and software development difficulties, this ensured every sprint was efficient and we encountered no setbacks from software.

The software testing was done by a combination of independent testing from one of us every addition of a function or by using automated testing classes, we did however not do this much as we had issue regarding hardware compatibility and due to only one of us having the hardware due to coronavirus restrictions, we needed to accommodate this fact.

The android development kit was the primary way we tested our software and allowed us to understand if we were missing anything for our project, an example of this is after testing our android application we decided we needed to make the UI simpler to understand for new users and as such redesigned it.

Overall software testing was extremely robust and allowed us to concentrate on more difficult testing such as hardware integration testing as we needed to ensure the software was tested with the hardware in mind for this project to work.

7.2 Hardware Testing

The hardware testing was the most complicated and difficult to achieve for many reasons, the main reason was that due to coronavirus and lockdown we could not share hardware as we normally would be able to do, because of this we allocated just one member to perform hardware testing with everyone else acting in support.

The main problem we came across during testing was in getting the power needed to use the hardware in the way we wanted, the board only has a small amount of onboard power and as such we required additional power to get it to function as we needed.

We solved this issue by attaching a small AA battery to the device and used it to supplement the power needs of the many functions we needed it to perform, this worked and allowed us to continue.

The main power issue was due to the fan and locks which would not work on onboard power alone, once we solved this issue, we also discovered through testing that a portable battery made even more powerful could allow the board to perform even more functions at a time allowing for a future expansion of the project if needed.

The other main problem we had during testing was how the hardware interacted with the dragon server, because of security of the server we had a large amount of trouble solving this issue but eventually settled on taking some data that we could not get onto the dragon server and moving it to the university's other server.

The testing we did for both servers proved extremely effective and we managed to ensure our project would be ready on time for evaluation, this could not have been done if we had not so extensively tested our hardware throughout the project.

The many bugs we came across along the way for this project was all minor and solved by the end of every sprint in which they were encountered, this allowed us to move forward to the next sprint without any hang-ups and greatly increased our productivity.

Overall, our testing methods was extremely robust, and we all took turns on 1 sprint each for testing, this allowed us to not only give people exposure to new problems but solved the issue of one person doing to much of a single objective every sprint.

8. Conclusion

To conclude, we built a system that allows the user to control their home in a great new way without the expensive and intrusive was that are currently available, this allows our smart life to answer the demand from a largely untapped market.

The objectives we set out to achieve at the beginning of our project was completed in six sprints, this was an extremely good achievement as all our objective was met and with the small time left over, we were able to test the software and hardware to an extremely high standard.

If we were to do this project again I believe improving our skills at algorithms would be a top priority as the time we spent learning and implementing these algorithms took us more then we anticipated however we took less time in some other areas so it evened out in the end for every sprint.

The teamwork and camaraderie during our development is one of the fundamental reasons we had so many successes throughout development and our diversity of skills and ideas contributed greatly to our project allowing us to complete it all to schedule.

Overall, the project can be viewed as a success as we achieved our objectives through the sprints and have left room for massive expansion of features for the project in the future, this will be explained next.

9. Future Work

Future enhancements could be made to our system, we could improve both the software and hardware side of the project, some examples are as follows:

Additional power supplies could allow more locks and fans as well as other devices to be controlled and theoretically is only limited by the power demands of these devices, this allows for good expansion in the future.

Alongside expanding already existing functions we can also add more for a user in newer versions via software updates and selling more parts to the user to add to their home system, by doing this the system could be upgraded for many years to come.

The system is very modular and as such future improvements can also reduce many functions for a user who wants an even cheaper system, for example if the just want automatic lock control they only need to buy the locks.

The software itself can be improved to accommodate greater functions in the future and the data mining algorithm with updates could one day be able to predict a user's habits to make life easier for them.

Overall, there is a great many future improvements and innovations that could be done to our system given more time, the way we designed the system will allow this for many years to come if wanted.

Acknowledgements

Our team would like to thank Daniele Soria for all the help they offered for the completion of this project.

Bibliography

Scott Campbell. (2015). *HOW TO SET UP THE DHT11 HUMIDITY SENSOR ON AN ARDUINO.* Available: https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/. Last accessed 22/03/2021.

Scott Campbell. (2015). HOW TO SET UP A 5V RELAY ON THE ARDUINO. Available: https://www.circuitbasics.com/setting-up-a-5v-relay-on-the-arduino/. Last accessed 22/03/2021.

Gowtham S. (2018). Serial Communication between NodeMCU and Arduino. Available: https://engineeringprojectshub.com/serial-communication-between-nodemcu-and-arduino/. Last accessed 22/03/2021.

https://arduinojson.org/. (2020). https://arduinojson.org/. Available: https://arduinojson.org/. Last accessed 22/03/2021.

Emmet. (2021). How to Install PHPMyAdmin on the Raspberry Pi. Available: https://pimylifeup.com/raspberry-pi-phpmyadmin/. Last accessed 22/03/2021.

https://randomnerdtutorials.com. (2021). ESP8266 NodeMCU HTTP GET and HTTP POST with Arduino IDE (JSON, URL Encoded, Text). Available: https://randomnerdtutorials.com/esp8266-nodemcu-http-get-post-arduino/. Last accessed 22/03/2021.

Appendices

Figure A.1 – Finished Design

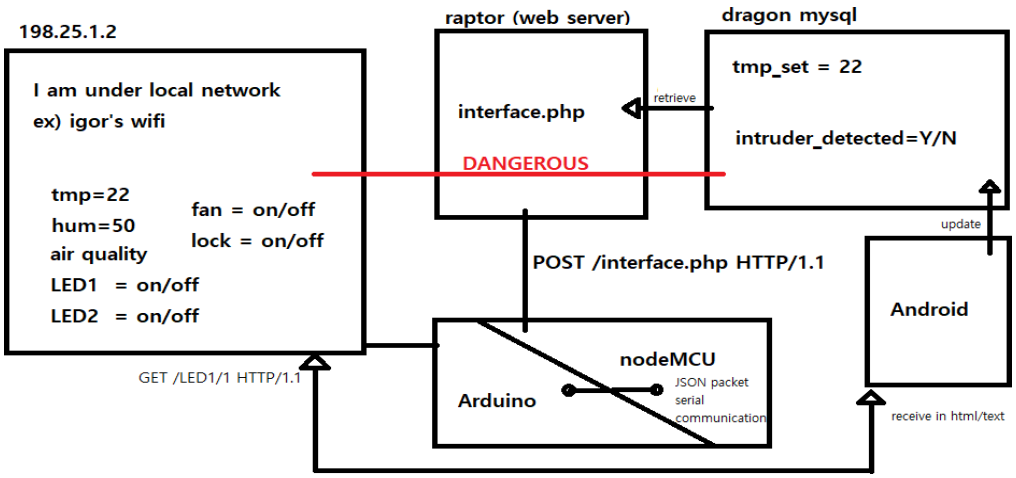

Figure A.2 – Assembled Hardware

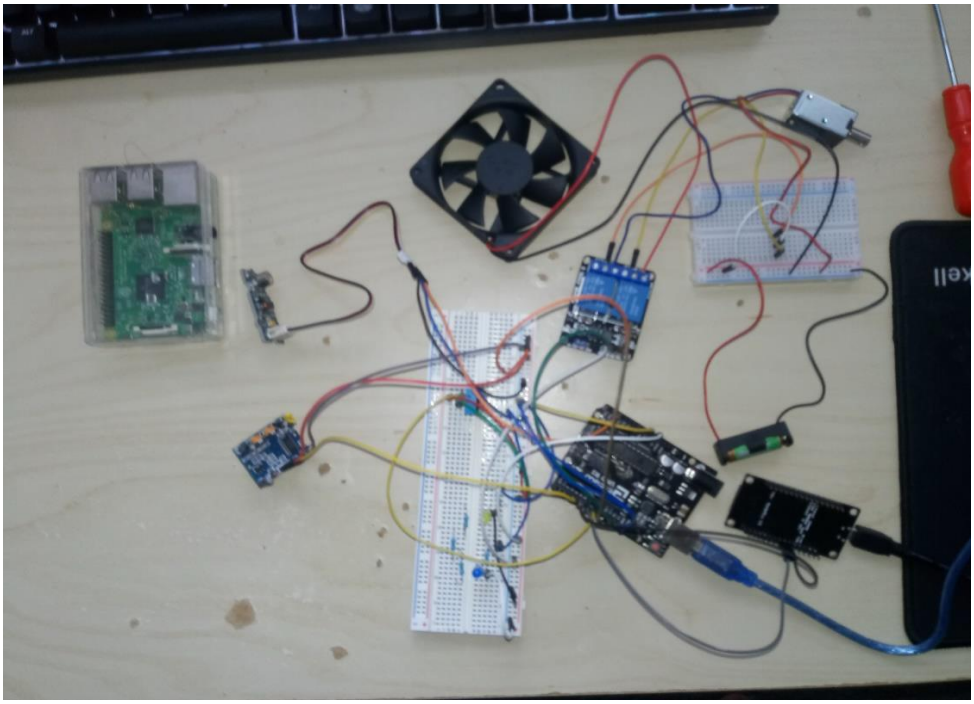

Figure A.3 – Initial Design

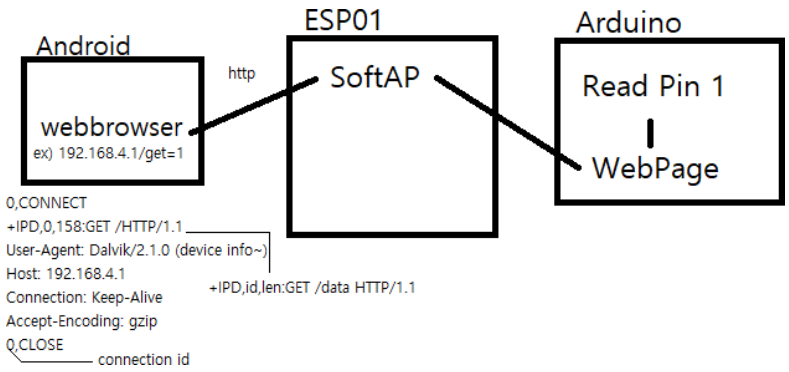

Figure A.4 – Login Design