

IOT Development Final Project

Caoimhe Malone
X14447022
BSHC4 IOT
IOT Development
May 2018

Android Application

My android application lets the user start an alarm clock, access an information hub and view the current temperature and humidity.

The alarm clock is the main function, when the button is pressed it runs the Clock_Method in the python script. It allows the user to set a time for an alarm.

When the connect button is pressed it connects to AWS so that when a button is pressed a message is sent using AWS to the raspberry pi. To stop the sending of messages press the disconnect button. The AWS code can be found in the home.java file.

Java Code

The credentials and certificates for AWS are located in the java code.

My folder path:

iot_final_proj/aws-iot-device-sdk-java-samples/src/main/resources/com/aws-iot-sdk-samples.properties

Python Code

The AWSIoTMQTTClient code is located in the python file, listener.py. This code gets the message from AWS which was sent from the android application.

Camera_Method contains the python code for the alarm clock. Once the button in the android app is selected it tells the raspberry pi that the Camera_Method should be started. Once started the RGB LCD lights up and shows the time, temperature, humidity and date. This is set in line 210 of the listener.py script. Hold down the button for 6 seconds to start the alarm. You can select to turn the alarm off by turning the knob of the rotary sensor anticlockwise and then

selected the button or you can turn the alarm on by turning the knob clockwise and pushing the button. Next you will be asked if you want to set a previously saved time or select a new one. To set a previously saved alarm turn the knob clockwise and push the button, to set a new alarm time turn the knob anticlockwise and press the button. To set the hour turn the knob and press the button. And the same for the minute. The alarm is now set and will go off at the stated time. Once the alarm is triggered hold down the button to turn it off or press the button once to snooze the alarm for 60 seconds. When the snooze alarm goes off it will ask you if you are still in bed. Press the button to stop it.

The Temp_Method prints the temperature and humidity from the sensors into the command line.

Each method has its own thread. These threads are in the listener thread.

HTML Code

The code for the information centre can be found in the folder workspace. My directory: `iot_final_proj/workspace`
It was created and is hosted on cloud 9.

Webpage: <https://smart-clock-example-caoimhemalone.c9users.io/index.php>

Here you can view a livestream of skynews from YouTube, the current temperature, weather forecast, the time along with news headlines.

The background for the clock and the news headlines changes according to the time of day. This is set in the `inc_library.php` file. The code for grabbing the headlines is also in this file. The top 20 headlines from the news are shown. They take a few seconds to show up as they need to be grabbed from the techradar RSS feed.

The javascript code for the weather is set in `weather.js`.

In the `config.php` file you set your location and other bits of information like you would rather the degrees to be shown in Celsius rather than Fahrenheit, which rss feed you would like to grab information from, I have it set for the techradar rss feed.

`Index.php` is where all of the html page code is.

`jsclock.js` 'draws' the clock for us.

Error Handling

Error Handling is in place in the python script. If there is a problem with the IO pins in the grove pi an error will appear in the console. If a key is pressed while the python script is running a keyboard interrupt error will appear in the console and the python script will stop running.

Changes from the original proposal

The idea in my original proposal and what I have produced are slightly different but they still both have the same goal; making an alarm clock. Although the idea of the panning information on the lcd screen seemed like an easy thing to do when writing the proposal it turned out that it was very difficult. In fact making a smart clock with a raspberry pi surprised me by being very difficult. There aren't a lot of references online to it. There are many YouTube videos of people showing off how their clock works, much like the video that I made for this report, but none of those videos linked tutorials or showed how they actually made them. I had a basic idea of how I could make something like what I had stated in my proposal but what I had in my brain and the code and resources available just didn't match up so I had to take a different approach. My android app became the hub for starting the script for the alarm. The LCD screen became a vessel of displaying the information centre section and the alarm would be displayed on the RGB LCD screen instead of the LCD touch screen. I wrote and rewrote the code for this project many times and failing made me reevaluate what I was trying to do with it. The product that I have produced is slightly different to what I had intended to do but it still has the same idea at play.

References

Dexter Industries. (n.d.). Dexter Industries. [online] Available at: <https://studio.dexterindustries.com/cwists/preview/432x> [Accessed 3 May 2018].

GitHub. (2017). riquezjp/kitchenTV. [online] Available at: <https://github.com/riquezjp/kitchenTV> [Accessed 4 May 2018].

YouTube. (2016). Kitchen TV using Raspberry Pi. [online] Available at: <https://www.youtube.com/watch?v=U5OF6tn-Ccg&t=11s> [Accessed 4 May 2018].