

IoT Project 1 - Deliverables

Project 1 Due: Friday 15th April 2016

**Student Name:** Brian McGowan

**Course:** Internet of Things

**Student ID Number:** S00165159

**Project Title**: Heating System Controller

**Background / Introduction**

Oftentimes it’s been the case that as you may not always be home whenever you planned to be. In such cases, the heating in the house may go on when it’s not needed, or by the time you get home the house is cold again. The aim of this project is to allow heating control remotely, to enable you to turn on or off via the web browser. And also as an added bonus, it enables the lazy of us to do it from our phones without having to get up.

**Requirements**

Requirements are:

* 1. Hardware:
     1. Arduino Board
     2. Relay Switch / Grove Kit
     3. 9V Battery / USB Power
     4. Wi-Fi Connection
  2. Software:
     1. Arduino IDE
     2. Custom REST API
     3. Wi-Fi Connection
  3. User:
     1. Ability to turn heating on
     2. Ability to turn heating off

**Methodology**

Configuring the hardware was a straight forward process. Firstly, configure the board to connect to the Wi-Fi network as per usual. Set up the relay sensor with the grove kit, on a digital port.

I needed to create a custom API for the YUN to operate my board Firstly I needed to create a web page for the index, turning the relay on and turning it off. This was created in HTML and CSS and will need to be installed on the board. Once installed the underlying Arduino code will need to interact with the web pages in the correct manner

Progress was tracked by firstly testing the web service was able to work correctly on my local machine, and then checking if it would work when deployed on the Arduino

**Success Criteria for Project Completion**

The relay switch will engage and dis-engage via push buttons on a web page installed on the Arduino sd card, accessed for the moment on an internal network via local ip address

**What were your main learnings in implementing this project?**

My initial attempt was to try and modify the Arduino API sketch so the website when installed could sit directly on top of it to interact with it via post, request and get patterns. However, due to limitations with the size of available memory on the board itself, and complications with the rest API code, this wasn’t possible. As recommended by Arduino, websites should be installed on the sd card on the board to be used. As the requests are made to the board’s api, it returns a status to the screen. At this point, I had my eureka moment of how to bring this forward. I implemented an ajax call via jQuery on the webpage of the board. The response of the request determined the data displayed on the screen, as well as the ability to use the on and off buttons, so you could easily check if the switch was on or off beforehand, a status check if you will. This opens up the possibilities of the board for me in a large way. I intend to modify the board’s api code to include data from a temperature sensor, and return this with the ajax calls that are made, so the web page can have more information displayed to the user to help with an informed decision.

**References**

Arduino Bridge Sketch <https://www.arduino.cc/en/Tutorial/Bridge>

Arduino Temperature Web Panel Sketch <https://www.arduino.cc/en/Tutorial/TemperatureWebPanel>