Results for Part 1

In this section, the results of the above outlined procedures will be discussed. The results will be discussed in three individual sections, with a brief conclusion of the overall result of the project closing the section.

Arduino Results

The manner in which the hardware system was put together allowed for results to be gained incrementally, that is to say that once one peripheral had been attached, it was tested to ensure it was sending the correct data and then the next peripheral was attached and so on.

Arduino and WeMos

The WeMos unit was the first peripheral to be attached to the Arduino unit. It was decided that ensuring a stable WiFi connection could be achieved was imperative to the success of the project/ This stage was undertaken first to ensure that a connection could be achieved as well as having a good starting point for sending sensor readings to the database once they were connected.

To test that the system worked, the Arduino was given a string that it passed to the WeMos. If the WeMos received the string, it would display it in the Serial Monitor of the Arduino IDE. Once successful, this destination of the string was then changed to the database to test the WiFi connection and whether the data was held correctly in the database.

This result was important as it gave the team a solid platform to build upon. The WeMos also has a unique MAC address which is used by the database as the primary key for grouping and sorting data. This not only means that the MAC address cannot be accidentally overwritten or altered (it is stored in the WeMos firmware), but if the system ever experiences a loss of WiFi connection, a new WeMos unit can be swapped in without arduous refactoring of code.

Arduino and Sensors

The sensors selected for the project were chosen not only for their ease of use regarding system implementation, but also on how they returned data to the system. For example, the DHT11 sensor was used to collect temperature and humidity readings. Not only did this combine two sensors in one, it also returns readings in terms of degrees celsius which meant that accurate results were being obtained as soon as the sensor was connected properly. This scenario was similar for the other sensors chosen in the project.

Testing of the sensors followed the same process as testing the WeMos connection. The sensors would return values and these would be displayed in the Serial Monitor. This allowed the team to fine tune how they wanted the data to be displayed as well as test that it was held correctly in the database.

Arduino and JSON String

The team decided that instead of storing the sensor readings in separate columns within the database, a JSON string would be used to collate the data and be held in one column of the database. The thinking behind this was based around how efficient the system could be at sending data as well as making it easier to parse this data for use on the website. Rather than sending four separate pieces of data to the database all at slightly different times, a JSON string is created that takes all four data ‘chunks’ and appends them sequentially. Once all data chunks have been received, the JSON string is then sent to and held in the database. This means the website only has to access two columns in the table (the ID of the hardware system and the JSON string) and can then parse out the required data from the JSON string where needed. This means that all four sensor readings will be sent at the same time, removing any discreprancies in the times of the collected data.

Arduino and LED Display

Whilst it was accepted that the main use of the data collected was to power the creation of graphs and charts on the website, the team decided that being able to view real-time data via a hardware display might also be useful for fine tuning the placement of a subject plant or communicating whether the plant needed watering or not. In order to achieve this, the TM1638 module was selected because it features a display and switch buttons on the same unit. The switch buttons would allow different data to be read on the same display without the need for having unique displays dedicated to showing one value from a particular sensor.

The display itself is made up of eight 7-segment LEDs, a rather simple form of display but one that was perfect for displaying numbers, which would mainly be the datatype on show. However, the display can have problems displaying certain letters of the alphabet, ‘M’ and ‘W’ being prime examples. Despite this potential limitation, the team viewed this as a unique creative challenge. Four readings had to be displayed on the screen, namely temperature, humidity, light level and moisture level. To ensure that the user knows which sensor data is being displayed, the first LED displays a lowercase letter relating to the sensor type; ‘t’ for temperature, ‘h’ for humidity and ‘l’ for light. Displaying the moisture level was more problematic, in that while the light level figure could be displayed, it could not be labelled the same as the previous three sensor displays. To overcome this, the code was modified slightly so that the display reads ‘H20 PLS’ if the plant requires watering or ‘NO H20’ if watering isn’t required. Whilst the display isn’t uniform in how it communicates the real time data to the user,

the display device has been kept as the TM1638 module as it is simple to use, displays data accurately and draws very low power compared to more complex display modules. With the switch buttons being part of the module, this reduces the footprint of the hardware system as well as making the back end code a lot simpler.