CS461 Capstone Design Document

December 1, 2017- Fall Term

Develop an Internet of Things Irrigation Valve

Prepared for

OSU |Openly Published Environmental Sensing Lab (OPEnS)

Chet Udell

Prepared by

Peter Dorich

CONTENTS

1 HUB DESIGN

The following three sections pertain to the hub section of the project. The hub portion of the project has been split into three different sections based on design. The first is the Adafruit control board, which will go over the orientation and design information of the board itself. This section will lay out the pin-out scheme on the board, and how different pieces of hardware will connect to it. This section pertains more to a hardware design, rather than software. The second section will highlight the flow of data in our system, specifically through the hub. And lastly, the third section will describe the different types of control modes that the user may activate on the irrigation valve.

1.1 Adafruit Control Board

The Adafruit 32u4 with long range radio will, firstly, require a power source to be powered on. Since this device will act as in intermediary between the computer and the irrigation valve, it can and will be located inside a building within a range of two to 26 kilometers from the field. With that in mind, it is far easier to implement a power supply that can draw electricity from the building, rather than a battery. Since the Adafruit 32u4 with LoRa requires five volts to run, we will use a five volt power supply that can plug into a wall socket. This implementation is ideal because the wireless hub will rarely have to be serviced. The five volt port is a micro-USB type B, and comes attached to the board. The next thing that needs to be implemented on this board is the Ethernet shield. This shield will allow connection between the system and the computer. Because most computers do not have LoRa built in, it is far easier and more efficient to implement a flow of data between the hub and computer on a network rather than local connection. In addition, this would allow remote control to be possible through the Internet. This physical connection also saves our development team from implementing things like passwords or authentication schemes. To improve our LoRa connection, a small antenna will be soldered onto the control board that will diminish the effects of Biomass blocking the signal. An important design constraint relates to memory. If the hub code is stored directly on the Adafruit 32u4, then if it were to power down unexpectedly, the volatile memory would disappear. The OpenS lab will provide software that saves all memory in non-volatile form.

1.2 Data Flow

All information that travels through our system will go through the hub. Controlling the flow of data becomes immensely important. The hub will use an existing framework that is used by the OPEnS lab. This framework currently works to organize all available soil data into a Google spreadsheet via the use of API's and the 32u4 board. The framework will need to be slightly modified in order to support multiple soil-moisture sensors and valves, as well as support for a different visual layout of our data. Currently, the framework uses an external API called PushingBox, which takes the soil data and inserts it into a Google Spreadsheet in an organized way. This API will continue to be used, however it has been discussed that the end goal for the data is not the document, but the web application. This API does not run on real time, but is capable of receiving an update every two minutes. Within the framework also exists Arduino sketches that allow the information to be pushed to PushingBox where it is logged in a document. This framework is a good start, but it lacks required functionality. For one, the current web application is deployed through a Google script, but in our implementation, it will be programmed with Java. This framework's implementation is achieved by editing a Google spreadsheet, but our implementation will use MQTT as broker between the web application and the hub. To enable the feather board for operation, the RH_RF95.h library must be included. In addition, the 32u4 Feather board must be setup with the appropriate MAC address, Static IP address, and the PushingBox device ID.

1.3 Control Modes

Now that the hardware and the software have been discussed, it should be noted what type of information will be exchanged, an how that relates to the valve control modes. The data being provided from the valve to the hub is the valve unique ID number, the soil-moisture content (VWC), and the state of the valve (open or closed). This amount of information is relatively low, to decrease overhead on the built in LoRa. The hub sends the web application the VWC content, the mode, the date, and the status duration of each of the moisture sensors and valves. When a valve's setting are being changed by a user on the web application, the hub will send the parameters to the valve with LoRa and Arduino sketches. The sketches will require the frequency of the radios, 915 MHz, and the Adafruit 32u4 feather board library, which will enable the use of the radio.

1.4 Conclusion