# Problem Statement

Jack Neff CS444 Fall2017

October 12, 2017

---

**Abstract**

This system we seek to design is an internet-of-things device designed to independently manage gardens through collection and analysis of plant growth data. Automatic gardening systems are commonplace, but this system tryuly automates by learning from past experiences, and applying that information to future practices. We will be using some kind of microcontroller (more will be revealed when we meet the client) to control and monitor watering, temperature, and humidity. This central monitor will be linked via network to microcontrollers in the surrounding garden, which admisister the correct amount of water to each plant. The growth data can either be fed into the machine manually, or recorded by the networked microcontrollers and relayed to the central monitor. The end result should be a fully autonomous networked garden.

# 1 Problem Statement

We are designing a system that automatically nurses an array of plants. This problem, at a small scale, amounts to automatically tending a home garden, and learning from past data how to grow plants as large and healthy as possible. Unlike systems that operate on preset timers that need to be set by gardeners, what we are designing would analyze data obtained from its own measurements of plant growth and would learn to apply the correct amount of water for the right amount of time in any environmental state. This reduces to two main problems: the collection of data with the use of sensors, and the analysis of data with proprietary algorithms that guides the system to finding the "correct" thing to do. At a larger scale, this kind of system would be used on wineries, orchards, or farms to automatically tend to crops. It would reduce water use by avoiding overwatering or wasting of water.

This project, titled The Green Smart Gardening System, aims to monitor, analyze, and implement the findings of environmental factors and their relationship to plant growth. The goal here is to have multiple sensors that monitor humidity, temperature, and soil moisture, and plant growth, and to feed the data from those sensors into a central computer that analyzes it and returns the results to the main controller. In this way, the system learns what watering techniques work best under different conditions, and can calculate the ratio of resource expenditure to plant growth to factor in to its analysis. Monitoring lots of different environmental factors will be difficult, and it will require a lot of experimentation to make sure our sensors are picking up data correctly. We also need a way to reliably transfer data from those sensors to a remote device, and a way to prevent the sensors from dying for lack of charge. If we can, we will solar power them, likely with a backup battery.

To implement this, we will need to acquire one or more humidity, temperature, and moisture sensors that can remotely transmit data. We need to connect them all to a centralized microcontroller. The type of microcontroller is to be determined. Our clients have a list for us, from which we will select one that

is optimized to meet our computational requirements at a minimal power consumption. Data recieved by the microcontroller will be transmitted to a remote database storage, where an offsite computer will analyze the data, and then pass to the microcontroller an analysis that informs its function. In order to translate collected data into an action plan for the system, we will need to write algorithms that take environmental factors as variables, and solve for the amount of water to give to a plant or area.

Analyzing the data and coming to a conclusion about it will be difficult. We will not be able to test extensively unless we use a lot of equipment and plant a lot of pots, given the time constraint and the amount of time it takes to grow a plant. Hopefully we will be able to collect enough data to perfrom our own analysis and write a specified algorithm. If we are, writing the algorithm will take some statistics research and some preliminary botany research. We also need to design an database capable of storing the correct symbols, and a network that can transmit data from a garden to a computer over the internet.

The microcontroller will need internet capability, and preferably some file transfer protocol. A program will need to be written to recieve, assemble, send, and parse data. It will be reliable but as energy efficient as it can be. One piece of design I would like to implement is the capability of the microcontroller to read that a sensor's battery is low, and send that information to the remote database, where it provides the connected computer with an alert. We will go one of two ways on the microcontroller in terms of user interface: either it is a modem-like, out-of-sight box, that operates basically self-contained, or an ipad-style interface with a great deal of user interaction. On an ipad interface, we could provide users with a nice looking display of collected data, in graph form. Even if we use the modem design, this data could be displayed on the connected computer.

The binding principle behind this whole project is that it is fully automated and sustainable. It manages itself in terms of function and energy usage, and only needs to be interacted with to perform maintenance. To accomplish this, not only the microcontroller, but all of the sensors need to be solar powered. Also, the central monitor needs to be constantly recieving new information to adjust its settings. In order to analyze the data, it needs to be beamed to the cloud, since the microcontroller will not be able to handle analysis computationally, and is restricted to local data anyway. However, the sensors shouldn't all connect to the cloud. To make things more efficient, the sensors should send their data to the microcontroller, which connects to the internet and sends the data to the cloud. In return it recieves an updated, analyzed series of data packets that tell it what to do with the conditions specified. This data is then implemented by the microcontroller.