SmartGrow
A Personalized Plant Monitor

Alexander Little and Martin Greenberg

Acknowledgements
Jorge Santiago-Aviles, Ani Nenkova, Jonathan M. Smith
Introduction and Abstract:

The robotic agriculture space has recently become a popular area for innovation. However, most existing systems in this space focus only on irrigation. Existing solutions for robotic agriculture are typically intended to be deployed on a large scale and are inappropriate for home-scale use. Finally, there are very few systems able to adapt to the particular plant being supported by the system. SmartGrow is an electronic plant ecosystem, designed for home use, to give the user an in-depth look at their plant’s health.

The system uses an Arduino Uno and its I²C bus to accurately measure factors that determine plant health such as soil moisture, temperature, and lighting. The Arduino automates some parts of the plant care, such as watering and lighting. The processing power of the SmartGrow system is provided by a Raspberry Pi 2. The Raspberry Pi is responsible for reading in the raw data from the Arduino, and then making decisions on what needs to be done for the plant.

Approach:

First, our group’s goal was to understand the existing systems, as there were many industrial scale systems in use today. We found that they could be grouped into three main categories, robotic watering systems, robotic horticulture systems, and robotic agriculture systems. The watering systems, such as “AccuRain”, “Growver” or “Droplet”, were mainly designed to cover a large-scale set-up of plants, and was not responsive to soil moisture. Instead they used a timer to water the whole area, typically a greenhouse of some sort and had a high cost on the scale of thousands of dollars. The robotic horticulture systems, such as the
Harvest Automation HV100 were made for mass harvesting of the plants, but little information is available on more advanced systems as they are custom-designed for individual companies and their details are not made public. Finally, the robotic agriculture systems were focused on one unique and specialized application. Specifically these systems could be used for pruning vines, grafting, spraying fertilizer on corn crops. In general, we found that most development in this space has been on robotic harvesting or irrigation, on a large commercial scale of plants. With the SmartGrow system we wanted to build a personalized plant-care system, built for more of a hobbyist gardener rather a commercial scale.

We wanted to build the platform to be able to support the growth of different types of plants. The system was designed to track metrics about plant and environment which would provide an accurate picture of the plant’s health. The four metrics that were originally planned to be used were: ambient temperature and humidity, soil temperature and moisture, ambient light levels and the plant’s level of photosynthetic activity. In order to track this, we would need sensors for each metric, specifically these five sensors: an ambient light sensor, an ambient temperature and humidity sensor, a soil probe for temperature and moisture sensing, a water level sensor for the reservoir, and an “Infragram” imaging sensor for photosynthetic activity. The first three sensors were sold by Adafruit, the Arduino distributor, while the soil probe and the camera were marketed by individual companies but made to be compatible with the Arduino Uno.

Once the plant was appropriately monitored, the SmartGrow system would also need to support the growth and health of the plant. The system would actively support the growth of the plant by: adding light by turning on a lamp if the environment is too dark, and by pumping water
from a reservoir into the soil if the soil is too dry. It would also passively support the growth of the plant by: warning the user if one of the metrics tracked exceeds a set threshold, and reminding the user to refill the reservoir when it is low. Specifically, the peristaltic water pump (DC motor) is controlled by Arduino using a Darlington driver, and the lamp is made switchable by a WeMo (internet-connected power switch) switch, which the Raspberry Pi can turn on and off. The notifications became text messages that were sent by the Raspberry Pi making REST calls to a web API, which would send the appropriate message, complete with data parameters.

Results/Measurements:

At the end of our project, we have a working prototype with a large majority of the goals accomplished. The Arduino collects sensor data and sends it to the backend on the Raspberry Pi, via a Bluetooth connection. Specifically, it collects soil humidity and temperature, ambient light, and water level all through its I²C bus. The active systems of watering and lighting are both functional, with the signals to their respective power drivers activating when appropriate. We were able to both physically see their usage, but also able to interpret system logs of correct behavior. The passive care was also successful as the Raspberry Pi would call an “If This Then That” web API to send out text messages to the user’s phone, which they would put into the user settings.

The component which we had to abandon was the Infragram sensor, as we were unable to produce useful plant health metrics from the device. Not only was the camera of a poor quality, but the data and algorithm which the company provided with the sensor did not seem to correspond to future plant health. The camera would be able to tell the differences between a
healthy and dead plant, yet it failed to produce any useful gradient of health between the two extremes.

**Ethical/Privacy considerations:**

The SmartGrow system does not have privacy concerns as it is a personal system, and would not transfer any data in between units. Ethically, if the system was given an incorrect plant profile, or its notifications not heeded, it could kill plants that the user planted; however, the whole goal and point of the system is to avoid this result.

**Discussion:**

Given more time and resources, a machine learning approach could be incorporated into the system. Rather than manually programming temperature/light thresholds for a particular plant, the system could learn the optimal thresholds by monitoring a plant’s photosynthetic activity using a set of trial thresholds, adjusting over time to learn the best temperature and sun exposure for any plant without requiring user intervention.
Sources Cited:


