Project Proposal

Team Name: Honey Badgers

Team Member:
Thomas Dye
John Walter

Team Member Focus:
Team member duties are anticipated to include, but are not restricted to:

- Research
- Report writing
- Coding
- Testing

With a team comprised of two people, there is not going to be a clear division of duties this early into the project. We are working using a collaborative space, either online using Google Drive or offline at the UW1-321 lab space. As requirements emerge for work needing to be performed, either of us can be assigned to perform a task. Independently, we'll tackle any of the team member duties as necessary. Collaboratively, we'll brainstorm or do rapid prototyping of something in the lab. There is not any one particular task that is needed to be performed by an individual team member. Tasks will be assigned based on scheduling availability, and to some extent, interest in a particular job.

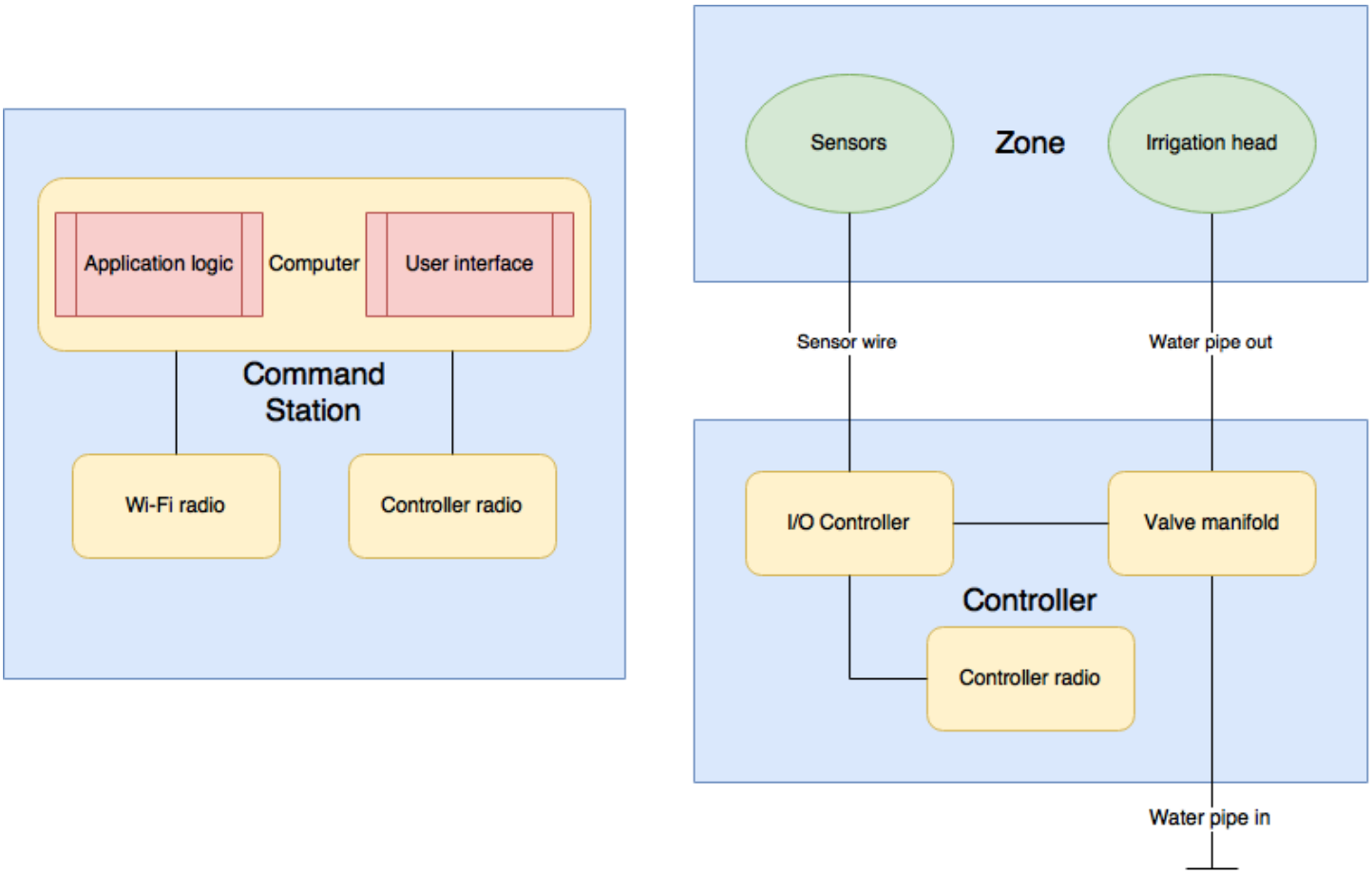
Project Description:
Our project is to create a "smart" irrigation system by using sensor analysis to control water flow to designated zones, in cooperation with UWB faculty member Rafael Machado De L Silva. Rafael has a similar project he wants created that has a high degree of feature overlap and has made hardware to accomplish this available to us.

Basic scenario:

- Sensors located in a designated area of land, hereby referred to as a "zone," report the current soil moisture level and other potential data to a controller.
- The controller wirelessly relays sensor data for a zone to a command station.
- The command station evaluates the sensor report. If soil moisture content has dipped below the zone's defined threshold, a watering schedule is considered.
    - The watering schedule may potentially be affected by the following conditions:
        - Time of day
        - Amount of direct sunlight being received
        - Wind, humidity, and temperature
        - Weather precipitation forecast
    - Depending on the types of sensors deployed, the command station will evaluate the zone's irrigation schedule.
        - Examples:
            - Watering in the early morning or late afternoon is preferred.
            - Hot, windy, and sunny weather conditions may require longer watering sessions.
            - Rain forecasted to occur may make additional watering unnecessary.
    - The command station creates or updates an irrigation plan and sends start / stop times to the controller.

- At the prescribed time, the controller opens and closes valves connected to piping to deliver water to zones.

The command station's evaluation of a zone's irrigation schedule will be augmented by user preference and the number (and type) of sensors in use. Our minimum goal for this iteration of the project is to collect soil moisture information as use that for evaluation. Additional sensor types and weather data analysis can be implemented as time allows.



The controller is directly connected to sensors via wire leads which are ideally collocated with irrigation pipes going to zones. The controller is remotely located outdoors. The command station communicates with the controller wirelessly over a long range radio link to periodically receive sensor data and to transmit updates to the irrigation schedule. The command station will provide a user interface to allow definition of zones and irrigation preferences. Our minimum goal is to provide an onboard lightweight webserver for user interaction (over TCP/IP) which connects to the command station's application logic. Weather forecast analysis would require an internet connection.

Some of this project's specialty hardware is sponsored by UWB through Rafael Machado De L Silva.

Sensor Shield (name and link):
Moisture sensor

- Overview
  - At minimum, we will use the moisture sensor included with the OSOYOO Sensor Kit for Arduino. https://www.amazon.com/OSOYOO-Modules-Mega2560-Raspberry-Learning/dp/B00WQY2704
  - Additional weather sensors to be provided by Rafael, with specifics not defined at this time.

Wireless Communication Shield (name and link):
Controller radio and accessories: Xbee PROS3B (XBP9B-DMWT-002 revF)
https://www.sparkfun.com/products/11634

- Overview
  - This long-range, low-power radio provides sufficient bandwidth to transmit raw sensor data and receive scheduling updates between the controller and command station.
  - The Arduino-based controller will use a single Arduino Shield Wireless SD https://www.arduino.cc/en/Main/ArduinoWirelessShield
  - The (likely) Raspberry Pi-based command station will also interface with this radio. For initial development, the radio is connected to a PC via the sparkfun Xbee dongle https://www.sparkfun.com/products/11697
  - This radio uses a robust communication protocol (IEEE 802.15.4), allowing radios to be configured to a private network using unique addresses. The radios themselves are self-contained embedded systems retaining their own configuration parameters.
  - 900 MHz radios were chosen for long range communication with high penetration without the need for a high gain antenna.
- Resources
  - Datasheet: https://cdn.sparkfun.com/datasheets/Wireless/Zigbee/90002173_N.pdf
  - Firmware utilities and API: https://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu
  - Drivers and Documentation: https://www.digi.com/support/productdetail?pid=5298

Other needed hardware components (name and link):
Arduino Mega 2560 https://www.arduino.cc/en/Main/ArduinoBoardMega2560

- Overview
  - Inexpensive central I/O device connecting the command station to the sensor and valve controller.
  - Has hardware support for XBee PRO series radios through the Arduino Shield Wireless SD.
  - 256KB of flash memory for backend I/O logic and scheduler.
- Resources:
  - Datasheet and software: https://www.arduino.cc/en/Main/ArduinoBoardMega2560

Command station

- Overview
  - This will temporarily be a Linux virtual machine running on commodity hardware with USB pass-through enabled.
  - At the conclusion of this project, this will likely be a Raspberry Pi or other similar mini-PC running a lightweight version of Linux and a webserver.
  - This device must be capable of running the Xbee radio and Wi-Fi simultaneously.

Plastic Water Solenoid Valve https://www.adafruit.com/products/997 (or similar)

- Overview
  - Either single solenoid controlled water valves, or multiple electronically controlled valves through a central manifold.
  - Specific hardware has not been identified at this time and will be procured under advisement by Rafael.

Estimated Time Schedule:

- 10/15/16: Project proposal
- 10/29/16: Xbee radio transmission between controller and command station. The controller can wirelessly send analog test data from sensor probes to a PC-based command station. The PC-based command station can instructions to the controller through a rudimentary functional API.
- 11/05/16: Final hardware requirements are identified and scheduled for procurement.
- 11/19/16: Basic functionality of the prototype is to be established
- 12/03/16: User interface prototype finished, final polish and testing scheduled
- 12/09/16: Prototype demo and handoff

Estimated Blocking Issues:

- Steady progress with development will be required throughout this project.
- The next meeting with Rafael to discuss sensor hardware has a dependency on getting the Xbee radio work finished.
- A PHP-based web interface, if we go that route for user interaction on the command station, will require learning PHP.