USC Viterbi School of Engineering

Ming Hsieh Department of Electrical Engineering

EE 459Lx - Embedded Systems Design Laboratory

EE 459 Project - Spring 2016
Irrigation Controller for Improved Water Conservation

Product Overview

Despite the predictions that 2016 may be a year of heavy rainfall in California, the issue of water conservation will continue to be one of paramount importance to all citizens. After several years of drought, in 2015 the governor requested that all users of water reduce their usage by 25% and in many areas the mandated reductions were even higher. To help understand where all the water at a typical residence was being used, last summer one of the instructors closely monitored the household water usage by recording the water meter numbers at the start and end of various events. The usage amounts were a bit surprising.

Typical shower	3.5 gallons
Dishwasher	10 gallons
Washing machine (full load)	40 gallons
Front and back yard sprinklers run for minimum allowed time	**400 gallons**

While it is true that "every drop counts", these results show that while conserving water inside the house (fixing dripping faucets, taking shorter showers, etc.) may help, it's not going to solve the problem. It's the water that is poured on lawns and landscaping that consume most of a homes water.

The help solve that problem, the project this semester is a residential or commercial irrigation controller that incorporates design features that will hopefully lead to reduced water consumption. The water conservation goal is achieved by two methods: sensors that help determine the amount of watering that needs to be done, and an improved user interface that makes it easier for users to adjust water usage.

Most low cost existing sprinkler controllers run as an open-loop system with no feedback to adjust the amount of water used based on recent rainfall, temperature, humidity, etc. This leads to water being wasted when the sprinklers activate when the ground is already sufficiently wet. The ability to adjust the water usage is a feature usually only found on higher priced models. A goal for this project is to incorporate this feature in a model in the price range most homeowners find acceptable.

A common problem with many irrigation controllers is that users find them difficult or inconvenient to make changes to the programming. One reason people give for not making adjustments to their sprinkler controller when the weather changes is that they don't know how it works. The design team should give careful consideration to the design of the user interface to make it easy for people to make adjustments. For example, an owner notices late one night that it's raining and realizes that the sprinklers are scheduled to run early the next morning. Many users would be unwilling to go outside in the rain to adjust the controller to not run the sprinklers. The product being developed here must have a way to make adjustments to the program that doesn't require going to the controller box and pressing switches.

Product Requirements

All teams must include in their product the basic requirements that are listed below. These can be be implemented in a variety of ways and it is up to the design team to decide how to include the features in their product. Design trade-offs in areas such as reliability, manufacturability, ease of use, cost, etc., should be analyzed to determine which is the best way to design the product. The product teams are encouraged to explore any designs that they can dream up and to go beyond these requirements in terms of both additional features and the quantity of each feature. Keep in mind the overall goal is to develop a commercially viable product which may or may not look like anything ever seen before on the market.

It is not required that every single feature that is claimed for the product be implemented in the prototype. Teams should plan to implement in their prototype about 80% of the features they claim for their product but should not make claims for product features that can not be implemented due to limitations on the available technology or the cost of the product. Teams are not required to implement every product feature to the same extent or quantity as would be required in the final product. However, the cost analysis of the final product should reflect any claims made for it, not just what was implemented in the prototype

Some of the requirement listed below have been added to the product simply to force teams to build a more challenging product that makes use of a certain amount of technology. In some cases teams may have to add features or capabilities to their product simply to meet these minimum requirements. This is to avoid a situation where, for example, if the class product was to build a residential burglar alarm, a team just paints a sign that says "Guard Dog on Duty" and says that's their product.

The main requirement for this irrigation controller is to include features that reduce water consumption. Two are mentioned above, sensing environmental conditions and easier programming, but teams are encouraged to add others if their market research indicates that these are technically feasible and would make the product more appealing to consumers. However the team must remember that each added feature will add to the product's cost so adding lots of interesting features may force the price above what many consumers would be willing to pay. Some possible design options can be found by studying the features that other manufacturers have included in similar devices.

You can incorporate any available technology you wish to use into the product (GPS, Bluetooth, RFID, WiFi, etc.), but any technology you use should meet two requirements.

- The cost of the product must reflect whatever technology you include in it.
- Any added technology must be used in some way. For example, don't add a Bluetooth interface to the product and claim it has "Bluetooth capability" but admit during your final presentation that you really never knew what you were going to use Bluetooth for in the product.

In the product requirements listed below, the terms "inputs" and "outputs" refers to whatever devices the product includes that senses conditions (the inputs) or does something to cause some action to happen (the outputs).

- **Irrigation Capability** - Many commercial sprinkler controllers can handle 6, 8, 12, or more control valves or "zones". The cost of the unit goes up as the number of valves it can handle increases. For this product, the number of zones the controller can handle should be determined by the market research and estimates of the cost of the device. The software should be written to support as many zones as is claimed for the product. However it is only required to implement in hardware the ability to turn **two** zones on and off. The ability of the prototype to control actual irrigation valves will be tested in the project lab.

  For however many zones your product is designed to support, the controller must have the ability to program each valve go through an on-off cycle at least three times in a 24-hour period. The controller needs to be able to keep track of the time of day much like a digital clock. It should also know what day of the week it is and allow programming that is different for each day of the week or that repeats after some number of days. Additional programming flexibility and features can be added at the discretion of the product design team.

- **Environmental Sensor** - At least two types of environmental condition must be measured and used by the controller for determining how much to water, or if at all. The conditions might include rainfall, humidity, temperature, moisture content of the ground, etc. At least one of the sensors must be connected to the controller over a remote link of some type. This can be wired or wireless but must transfer data using an protocol appropriate for this type of link. The remaining sensors can be either local or remote, whichever the product team feels is best. The sensor information can be analyzed over as long a time as the team feels is appropriate. For example if the sensor is detecting rainfall now, but there hasn't been any for the last two weeks, the controller may still decide to run the sprinklers for a short time.

- **Remote Control** - Most irrigation controllers are programmed from their front panel with buttons, knobs and various displays. Your controller needs to have this local programming ability but must also be programmable (to some extent) from a remote device. This could be a custom remote control like for a TV, a smart phone app, a web site, etc. As a minimum requirement, the remote device must have the ability to override the action of the controller so a user can prevent the irrigation cycle from happening if it has started to rain.
- **Immune to loss of power** - All configuration data for the operation of the device should also be retained through power outages. It should not be necessary to reprogram it simply because it suffered a power outage. The product's internal time-of-day clock must be able to continue to function during power outages so that it will know the correct time when the power is restored. Alternatively it might incorporate the ability to set itself to the correct time when power is restored. It is not required that the various outputs continue to operate if the product loses power, however once power is restored everything should resume normal operation without operator intervention.
- **Reliability** - If the product consists of multiple devices that communicate via a link of some sort the main device must be able to detect and notify the user when it can no longer communicate with a remote device. This might be due to a broken communications link, a dead battery, or the remote unit itself may have become defective. A user should not think that all is well with his product when in fact a remote unit is no longer working.
- **Security** - If the product uses wireless links between the controller and one or more remote units, the design must address the issue of how to prevent inadvertent interference with or from another similar unit. The product should only communicate with wireless devices that it is supposed to be associated with. Your controller should not be turning your neighbor's sprinklers on and off. Same applies to your controller in that it should not be affected by the operation of a neighbor's system, or by a hacker.
- **Innovative** - At least one of the input or output components of the system must be new to the market in that it can not be found in competing products that are currently on the market.

All teams are encouraged to go beyond these minimum requirements in order to make their product more attractive to consumers. Whenever possible, additional features selected for inclusion in the product should be included in the prototypes sufficiently to show that they can be implemented and would work as planned. Teams should always be aware of how additional features will affect the cost of their product and be prepared to justify the added cost. For example, a team may decide to add a module to their product to give it a wireless connection to the Internet so the user can control and monitor the actions of the device from a remote site. The cost of this module must then be factored into the cost of the product. Saying that you have added a $80 module to a project that is supposed to sell for $60 is not a good idea if you know your consumer will not pay the associated increase in price.