Technical Specification / Schedule
PC Control

Team 10: Team Toro
CSE 498, Collaborative Design

Stephanie Cook
Matthew Grabow
Daniel Fiordalis
Thomas Castellani

Department of Computer Science and Engineering
Michigan State University

Spring 2008
Project Overview

• Controller
  – Receives signals from USB Remote with information on desired watering and lighting behavior
  – Implements schedule by sending power to appropriate zones

• USB Remote
  – Sends signals to Controller wirelessly
  – Communicates with PC Control via USB

• PC Control
  – Create watering and lighting schedule
  – Clean, usable user interface
Architecture Illustrated

Team 10: Team Toro
Functional Specifications

• Controller Schedule stored in XML format
  – UI generates XML based on user input
  – XML packaged and sent to Controller for implementation
  – Advanced error handling for failed parsing

• User Interface allows for easier navigation
  – WPF animations
  – Fewer dialog box interruptions
  – Tabbed windows
    • Maximize use of screen real estate
    • Abstracts tools into logical categories
Functional Specifications

• Advisor scheduling
  – Implement hardware restrictions in software
  – Utilize environmental variables to develop schedule
  – Interface with internet to retrieve weather information

• USB Drivers
  – Reduce unnecessary polls to hardware
  – Add asynchronous functionality to driver
  – Improve handling of failed communications
System Components

• Hardware Platforms
  – Controller
  – USB Remote
  – PC

• Software Platforms / Technologies
  – Windows XP and Vista
  – Windows Presentation Foundation
  – Visual Studio 2008
  – .NET Framework v 3.5
  – C# 3.0
Architecture Illustrated
Architecture Illustrated
Architecture Illustrated

Team 10: Team Toro
Architecture Illustrated

- Scheduling Advisor

Team 10: Team Toro
Architecture Illustrated

- USB Communication
Risks

- **USB Drivers**
  - Ensure compatibility for Vista and XP
  - Purchased and studying *USB Complete* textbook

- **Porting Action Scripts functionality to C#**
  - Maintain developed functionality
  - Refine functionality with event handlers

- **Windows Presentation Foundation**
  - Develop aesthetically pleasing modern UI
  - Use Expression Blend to create UI
Project Schedule

February 1
- Outer window
- Settings
- Helper dialogs
- Initial WPF animations
- Some styling completed
- Gain understanding of USB Complete
- Break down driver code for understanding

February 8
- Save and load schedules into application (parsing only)
- Have interface successfully detect remote
- Images (logos, etc) placed and loaded into interface
- Write driver than can successfully communicate with application
Project Schedule

February 15

• Advisor taking input and returning text
• All dialogs present
• Basic templates applied to controls
• Detect OS for visual settings
• Interface to send information to USB device
• USB driver that’s discoverable and has minimal functionality

February 22

• Balloons for error or info messages where no click is necessary (ie when USB device successfully connected)
• Save location for weather
• Store list of favorite zips for weather (useful for contractors)
• Parsed schedule to be displayed in the schedule window
• USB to be able to manually turn on and off zones
• USB to be able to send/receive schedules
Project Schedule

February 29
- Imposing hardware/software restricts
- Manual mode
- Scheduling mode
- Rain sensor implementation
- Manual Scheduling

March 14
- Distinguish between lighting and watering
- Fully functional schedule
- All animations and styles finalized and consistent
- Plan video

March 21
- Debugging
- Video development

March 28
- Project video
Project Schedule

April 4
- Tweaking to desirable behavior
- Bug fixes

April 11
- Finish GUI
- Zero bugs
- Create installation package

April 18
- Prepare for Design Day

April 25
- Design Day