Project Plan

for

Sun SPOT Universal Monitoring System (SUMS)

Version 1.5
Prepared by Team Awesome
Texas Christian University
December 6, 2008

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
First Version	11/09/2008	Initial Version	1.0
Revised Version	12/06/2008	Revised to reflect instructor comments and feedback.	1.5

Approvals

Print Name	Title	Signature	Date Approved
Brian Freeman	Project Manager		
Lei Jin	Physical Components Lead		
Anh Pham	Applications Lead		
Jamie Phelps	Front End Lead		
Edward Tran	Back End Lead		

1. Introduction

1.1 Purpose

The Sun Small Programmable Object Technology (SPOT) Universal Mote System (SUMS) Project Plan will provide a definition of the project, including the project's goals and objectives. Additionally, the Plan will serve as an agreement between the following parties: Project Sponsor and Instructor (Dr. Donnell Payne), Project Team, and other personnel associated with and/or affected by the project.

The Project Plan defines the following:

- **Introduction**
  Purpose
  Background Information
  Project Approach
- **Project Goals and Objectives**
- **Project Overview**
  Scope Definition
  Project Assumptions
  Risk Assessment
- **Requirements Overview**
  Software Requirements
  Hardware Requirements
  Project Constraints
- **Project Management Approach**
  Deliverables and Milestones
  Project Schedule

1.2 Background Information

SPOTs are developed by *Sun Microsystems, Inc*. and can form a wireless sensor network (WSN). The project is to build a partially wireless system that allows users to monitor a variety of sensors on a computing platform and capture the real time data in multiple interfaces as they want.

The main hardware focus is the Sun SPOT Java Development Kits. For each development kit, there is a single base station that connects to computers via USB. The base station acts as a signal or data receiver for configured wireless motes. Each development kit also has two SPOTs with built in sensors. Additional external sensors may be connected to a SPOT. The base station and SPOTs have the capability to communicate with each other through radio signals.

The Sun SPOT Development Kit included the following software for development: NetBeans IDE 5.5, SPOT Manager, Apache Ant 1.6.5, and other supporting environmental setups such at the Java Runtime Environment and Standard Developer's Kit. We have found newer versions of the software are functional for SPOT development.

Currently the development environment and software has been set up. Experimentation with the SPOTs also has begun. As the project progresses, additional external sensors or servomotors may be needed for development of example SPOT applications. According to predicted facts and

constraints, the topology and structure of SUMS has been identified. The environment, interfaces, and languages that will be used in developing the system have also been determined, but remains volatile as domain exploration continues. These will be described in detail in either the following sections or in our Software Requirements Specifications.

1.3 Project Approach

This project will be rolled out in iterations. Each iteration will build upon its predecessor until final release.

Phase 1: Project Support Environment
Phase 2: Problem Exploration
Phase 3: Iteration #1
- 3.1 Documentation
- 3.2 Design
- 3.3 Implementation
- 3.4 QA/Testing

Phase 4: Iteration #2
- 4.1 Documentation
- 4.2 Design
- 4.3 Implementation
- 4.4 QA/Testing

Phase 5: Iteration #3
- 5.1 Documentation
- 5.2: Design
- 5.3: Implementation
- 5.4: QA/Testing

Phase 6: Iteration #4
- 6.1 Documentation
- 6.2: Design
- 6.3: Implementation
- 6.4: QA/Testing

Phase 7: Code freeze and review.
Phase 8: User documentation.
Phase 9: Release

2. Project Goals and Objectives

This project will proceed in two separate phases.

The primary goal of the SUMS system is to provide a system for monitoring sensor data of SPOT devices from a graphical console. Phase one will focus on this aspect of the SUMS project. We aim for this project to be a point from which future projects can be started. It will provide a basic solution that will aid the development of future projects by providing a supporting environment for users to monitor the SPOT devices in range.

Phase two of this project will be the development of two example applications that demonstrate the capabilities of SPOTs and how SUMS supports the development of those applications.

3. Project Overview

SUMS provides a solution for testing sensors while developing SPOT applications. There are three basic levels of the system within the system: Wide Area Mote Monitor (WAMM), Local Area Mote Monitor (LAMM), and the motes. The following describes the interactions of these systems and the figure on the next page diagrams the overall topology of the system. (See Figure 1.)

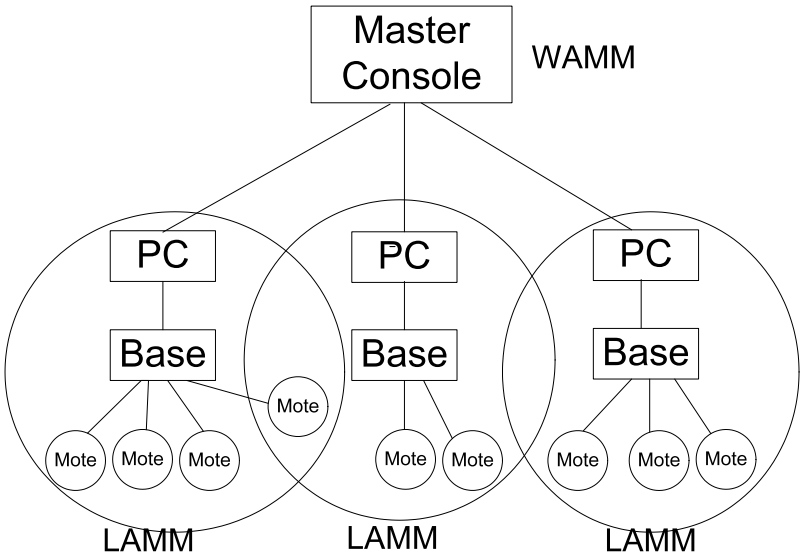

Figure 1

Motes are SPOTs programmed to connect to SUMS. Each mote will be programmed to send information and readings from one or more sensors. The motes connect to a LAMM's base station through radio communication using the 802.15.4 IEEE standard.

The LAMM monitors the local radio space of the base station. It interacts with connected motes in its radio space by specifying configuration information such as setting a sensor's sampling rate or having the mote filter any unwanted sensors. The LAMM communicates with the motes through the radio communication and with the WAMM through a network connection using sockets.

The WAMM is a supervisory system that monitors the overall topology of the entire system. It may display specific information and some sensor data of one or more connected motes in the system. The WAMM may also be referred to as the master console. The WAMM communicates with the LAMM through a socket network connection.

Along with the SUMS system, the project is to develop two approved SPOT applications. The applications will run independent of SUMS, but SUMS will be used to monitor the sensor data of the SPOTs as they perform their application.

3.1 Scope Definition

The project is the creation of SUMS. SUMS introduces a new level of testing and monitoring capabilities when developing applications on SPOTs. These capabilities include:

- Reading sensor values from motes
- Displaying sensor values from motes through LAMM and WAMM consoles
- Ability to log sensor data.
- Monitoring the overall topography of motes connected to SUMS

The project also includes two example applications to demonstrate the capabilities of SPOTs and SUMS.

3.2 Project Assumptions

The project makes the following assumptions:

- Project team members will be able to competently handle their responsibilities
- Project team members will be able to make the time to meet and communicate with each other outside of the given class time
- Project team members will continuously document their individual work through weekly activity logs allowing for the monitoring of the project's progress
- Users of SUMS will be competent in Java in order to properly program the motes and understand the project's documentation
- The hardware used in SUMS is fully functional and have not been compromised
- SPOT development kits will be provided by Dr. Payne and the TCU Computer Science department for the development of SUMS
- Additional sensors and equipment for example applications will be provided by Dr. Payne and the TCU Computer Science department
- A project support environment consisting of a server to hold project files and a network of computers will be provided by Dr. Payne and the TCU Computer Science department
- The TCU Computer Science department will be able to handle the future deployment of SUMS
- Future maintenance and user training will be handled solely through the developer and user guides provided with SUMS

3.3 Risk Assessment

The following is an overall risk assessment for the project. Risks will be continuously monitored and handled as needed.

Risk	Type	Severity / Probability	Mitigation Strategy
Member Turnover	Organizational	High / Low	Keep weekly activities logged to track member's responsibilities.
Management Change	Organizational	High / Low	Write an accurate and sponsor approved project plan and requirements document to alleviate management responsibilities from project sponsor.
Requirements Change	Project	High / Medium	Early documentation and SPOT exploration. New requirements will be documented as needed.
Unrealistic Schedule	Project	Medium / Low	Constant communication between sponsor, customer, and project manager to discuss project progress, short-term milestones, and long-term

			milestones.
Hardware Underperformance	Technological	Medium / High	Early documentation and SPOT exploration. Requirements may need to change as necessary.
Software Interface Incapability	Technological	High / Low	There are code demonstrations involving the interfacing in Java. If our first choice of development platform does not work, we can fall back on Java.
Loss of Brazos Server	Technological	High / Medium	It is a new server. Team members will have repositories of somewhat up to date versions for recovery use. Need to back up repository for offsite storage periodically using `svnadmin dump`.
Absence of team member commitment/attitude	People	High / Low	Motivation and persistence from other group members. If all else fails, using low marks and graduation status as motivation.
Absence of customer commitment / attitude	People	High / Low	Persistence from team members.
Low Team Communication	People	Medium / Medium	Regular meetings and weekly activity logs to keep everyone in check. As well as communication through e-mail.
Equipment Delays	Technological	Medium / Medium	Identify needed components and place orders as early as possible. Brainstorm alternate applications that can be accomplished with components on hand or readily available.
New Development Language	Technological/ People	High / Medium	Begin work with the development environment and language as soon as possible. Identify gaps in knowledge and possible resources to bridge those gaps as early as possible. Identify more experienced developers who may be willing to help.
Communication Between Disparate Programs (C#.NET and Java)	Technological / People	High / High	Identify and communicate any insurmountable obstacles early so that the project may fall back to Java development if necessary. Create data structures and streams that can be read and understood by both programs.

4. Requirements Overview

4.1 Software Requirements

- Windows XP or Vista
- Sun SPOT SDK (v4.0 - Blue SDK)
- Sun Java Runtime Environment (JRE) 1.6.0_07
- Java SE Development Kit 6
- Apache Ant 1.6.5
- NetBeans 5.5
- Sun SPOTManager

4.2 Hardware Requirements

- Sun SPOTs Development Kits (A standard development kit from Sun includes a base station and two SPOTs as well as a USB cable and software CD.)
- PC with USB port
- Network (WAMM and LAMMs should be in the same TCU subnet)
- Any sensors or other components required for applications (to be determined).

4.3 Project Constraints

4.3.1 Resource Constraints

- Budget. TCU Computer Science department will need to supply SPOTs and additional sensors.
- Availability of sensors and other required components.

4.3.2 Time Constraints

- Project must be completed by April 28, 2009. This is a firm date that is determined by the schedule of the course sequence. This deadline must be met by any means necessary.

4.3.3 Environmental Constraints

- WAMM and LAMM will operate within the TCU subnet. Any limitations imposed by TCU Technical Resources will apply to SUMS as well.

4.3.4 Hardware Constraints

- Processor speed and memory limits of SPOTs.
- Radio transmission speed and efficiency.

5. Project Management Approach

5.1 Deliverables and Milestones

5.1.1 Project Support Environment

Date: October 3, 2008

The Project Support Environment will consist of installation and study of all required working environments. It will also result in the creation of a website skeleton.

5.1.2 Project Plan v. 1.0

Date: November 9, 2008

The first version of the Project Plan will provide a sturdy starting point that will be continuously edited until the final draft is completed.

5.1.3 Project Plan Presentation

Date: November 11, 2008

The initial version of the Project Plan will be presented to the instructor, sponsor, and user.

5.1.4 Software Requirements Specification v. 1.0

Date: November 11, 2008

The first version of the Software Requirements Specification will be a solid foundation to allow for future iterations. These iterations will ultimately result in a final draft.

5.1.5 Software Requirements Specification Presentation
Date: November 13, 2008

The initial version of the Software Requirements Specification will be presented to the instructor, sponsor, and user.

5.1.6 Design Document Draft
Date: November 21, 2008

This milestone is a draft of the design documentation.

5.1.7 Communications Testing and Detail
Date: November 24, 2008

This milestone consists of determining the communication architecture for the project in complete detail. There will be testing to see if the methods used are the most efficient. There will also be a considerable amount of modeling of these systems as well. Also as part of this milestone, pulse width modulation with the Sun SPOTs will be explored, studied, and practiced.

5.1.8 Design Document v. 1.0
Date: December 6, 2008

The first version of the Design Document will outline the design for our project which will likely be expanded upon and further revised before release of a final draft.

5.1.10 Project Progress Presentation
Date: December 9, 2008

Team Awesome will present a state of the project report to all those interested or invested in the project. This will be an introduction to the problem, to the system and its functionality, and a discussion of the salient points of our software engineering process and design document.

5.1.11 1st Iteration
Date: January 19, 2009

The 1st iteration of the SUMS system. This will consist of allowing a SPOT to connect to a base station and send information up to the top tier of system.

5.1.12 2nd Iteration
Date: February 2, 2009

The 2nd iteration of the SUMS system. The first application will also be completed by this date.

5.1.13 3rd Iteration
Date: February 16, 2009

The 3rd iteration of the SUMS system.

5.1.14 Code Complete
Date: March 23, 2009

Code will be complete by this date. The second application will also be completed by this date.

5.1.15 Code Freeze
Date: April 14, 2009

Code changes will not be allowed after this date. The code on this date will be the code that is tested and documented. All critical bugs should have been identified and fixed by this date.

5.1.16 Student Research Symposium
Date: April 17, 2009

SUMS will be presented in the Student Research Symposium.

5.1.17 Testing Complete
Date: April 20, 2009

Testing will end on this date. Major problems with the system shall be fixed and tested.

5.1.18 User and Developer Guides
Date: April 21, 2009

Team Awesome will provide both a user and developer guide for the software system.

5.1.19 Final Presentation/All Deliverables Due
Date: April 28, 2009

The project will be completed at this point and all the deliverables turned in to the class instructor.

5.2 Project Schedule

See the attached Gantt Chart for a detailed project schedule.

5.3 Roles and Responsibilities

The roles and responsibilities presented here are very general so as to give a good idea as to who is responsible for the completion of certain tasks. These roles will evolve throughout the project. Also, if necessary, team members should be able to assist with parts different from those of their own. Testing of the software system and applications will be done by those who did not work on those parts. This will hopefully catch many of the errors that an actual user may encounter.

5.3.1 Project Manager
Brian Freeman

- Schedule milestones, work with Gantt chart schedule, inform team members of schedule changes
- Communicate with the customer/sponsor
- Assist the team with issues that may arise and maintain team relations
- Act as a back end programmer under the back end leader

5.3.2 Front End Lead
Jamie Phelps

- Provide primary vision for design user interface for the SUMS system
- Implement the user interface according to design
- Maintain the website
- Assist with problems arising with the server

5.3.3 Back End Lead
Edward Tran

- Ensure implementation of the base station and Sun SPOT subsystems of SUMS
- Act as primary programmer on the back end
- Provide technical expertise on SPOTs and feasibility estimates for schedule

- Assist the project manager in leading meetings

5.3.4 Physical Components Lead

Ancandy Jin

- Ensure safety and reliability of hardware
- Building components used in applications
- Maintain document templates and compilation
- Act as secretary for the group

5.3.5 Applications Lead

Anh Pham

- Provide primary vision for applications design
- Act as primary programmer on applications
- Work with physical components leader to ensure safety and reliability for applications hardware
- Maintain and edit project media (videos, screen capture videos, and photos)

The following is a responsibility matrix to further show the responsibilities of each member of the group.

Tasks | Names	Ancandy	Anh	Brian	Edward	Jamie
Project Plan	C	C	C	C	C
Software Requirements Specification	C	C	C	C	C
Design Document	C	C	C	C	C
1st Iteration of SUMS	A	A	C	R	C
Completion of 1st Application	R	C	A	A	A
2nd Iteration of SUMS	A	A	R	C	C
3rd Iteration of SUMS	A	A	C	C	R
Completion of 2nd Application	C	R	A	A	A
Code Completion of SUMS	A	A	R	C	C
Testing Complete	C	C	C	R	C
User Guide	C	C	C	C	C
Developer Guide	C	C	C	C	C
Activity Progress Reports (Logs)	C	C	C	R	C
Project Media	C	R	C	C	C
Project Website	C	C	C	C	R

Matrix Key:
A – approve: this person is expected to understand and approve solution to the task.
C – contribute: this person is expected to contribute in the completion of this task and approve of the solution to the task.
R – responsible: this person (only one) will ensure the completion of the task.