Activity Monitor

version 0.1

Jason Poon

March 08, 2010

Contents

Welcome to Activity Monitor

Contents:

Project Proposal

MEMORANDUM

To: Ms. J. Pavelich, Prof. S. Gopalakrishnan

From: Jason Poon

Date: March 08, 2010

Re: EECE 496 - Project Proposal

Introduction

The project I would like to propose for EECE 496 is the implementation of a fall detection system. The purpose and significance of such a system are outlined below; in addition, details regarding the requirements and timeline are included.

Project Objectives

The purpose of this project will be to implement a fall detection system. With the usage of multiple sensor nodes places around one's body, it is the goal of the system to aggregate data from the various sensors and detect if a fall has taken place.

Project Significance

A fall detection system can prove very useful for senior citizens where a fall may be very dangerous. With the usage of a fall detection system, it is then possible to monitor one's activities and, in the case of a fall, automatically notify the authorities (e.g. caregiver) such that they can immediately respond to the situation. Such a system can also be applied any individual monitor their exercise routine and provide an analysis of the data.

Project Requirements

In order for this project to succeed, the following items will be required:

- **Sensor Module**
  - An accelerometer will be used to detect the motion of the user and to determine the current state in which the user is in (e.g. standing, sitting, falling, walking)
- **Wireless Sensor Network**
  - In order to coordinate data retrieved from the various sensors placed around the body, a wireless sensor network will be required.

Project Timeline

ID	Task Name	Start	Finish	Duration
1	**Setup**	**1/28/2010**	**2/5/2010**	**7d**
2	Selection of Operation System	1/28/2010	2/1/2010	3d
3	Installation and configuration of development environment	2/1/2010	2/5/2010	5d
4	**Design**	**2/1/2010**	**2/18/2010**	**14d**
5	Gain familiarity with OS API and hardware	2/1/2010	2/12/2010	10d
6	Software Architecture and Design	2/5/2010	2/18/2010	10d
7	**Development**	**2/10/2010**	**4/1/2010**	**37d**
8	Mote Communication	2/10/2010	2/23/2010	10d
9	Accelerometer Drivers	2/23/2010	3/8/2010	10d
10	Base Logic	3/8/2010	3/19/2010	10d
11	UI	3/19/2010	4/1/2010	10d
12	Testing	2/15/2010	4/2/2010	35d
13	**Documentation**	**3/1/2010**	**4/12/2010**	**31d**
14	Progress Report	3/1/2010	3/8/2010	6d
15	Final Report	4/1/2010	4/12/2010	8d

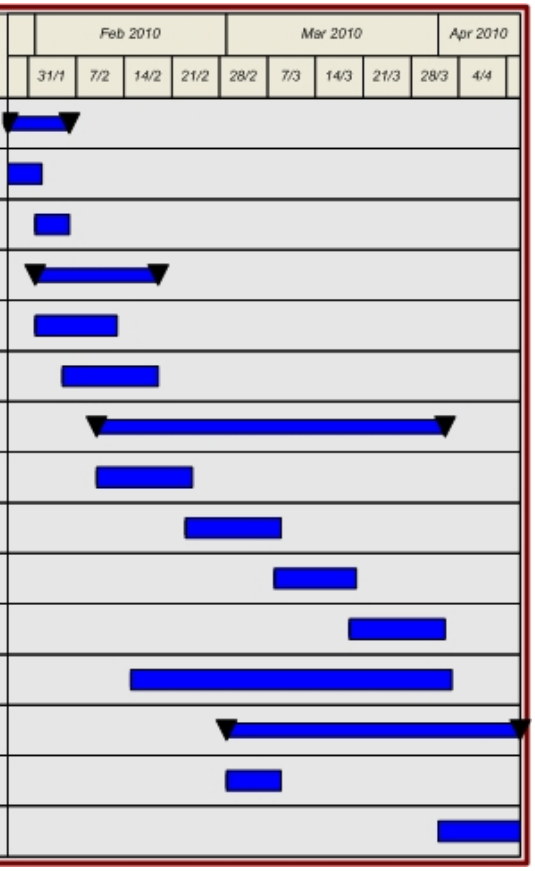

Figure 1 - Gantt chart outlining schedule for project

Conclusion

The included gantt chart will be used as a guideline for the scheduling of tasks. In following the gantt chart, the successful completion of a system able to detect an individual's current state should result. If you have any questions regarding this proposal, feel free to contact me via email at eece496@jasonpoon.ca.

Jason Poon

Status Report

Project: Activity Monitor

Name: Jason Poon

Student ID: 21736053

Technical Supervisor: Prof. S. Gopalakrishnan

MEMORANDUM

To: Ms. J. Pavelich

From: Jason Poon

Date: March 08, 2010

Re: EECE 496 - Project Status

Introduction

Under the guidance of Prof. Gopalakrishnan, I have been developing a system capable of monitoring various activities via several sensor nodes attached to the body. The goal of the system will be the ability to monitor an individual's activities and motions. With such a system, it can be used to detect harming motions such as falls where the system can notify the authorities (e.g. caregiver) where they can immediately respond to the situation.

The overall project progress is outlined below in three sections: work completed, work in progress, and work remaining.

Work Completed

- **Order/Receive Necessary Hardware Components**

  Hardware components required for this project include: wireless nodes, accelerometer sensors, and a USB debugger or flasher circuit.

  Several Firefly nodes, Firefly sensor boards (3-axis accelerometer, temperature sensor, light sensor, and microphone), and a USB debugger. While waiting for the parts to arrive, Prof. Gopalakrishnan was able to lend me a micaZ mote and USB flasher. The Firefly components took longer than expected to arrive which pushed the schedule back approximately two weeks.

- **Real-Time Operating System (RTOS)**

  Following an initial investigation of which RTOS to use, two serious contenders remained: Nano-RK and LiteOS. In the end, Nano-RK was chosen over LiteOS.

  The LiteOS installation process was considerably buggy and development on LiteOS was unfortunately heavily geared towards Windows. Nano-RK, on the otherhand, is very well documented. Although Nano-RK does not have a forum or mailing list to post questions, the project owners were very helpful. For instance, Anthony Rowe, one of the primary contributors to the Nano-RK project, quickly responded and assisted in solving a flashing issue I was experiencing with the Firefly nodes. He also provided me with write access to the Wiki and the SVN repository where I have since committed several patches to the Nano-RK project.

- **Development Environment Setup**

  No major difficulties arose while setting up my development environment. The combination of developing on a Linux machine and the quick-start guide from Nano-RK made the process very straight forward. The project source files are available publicly through Git at http://github.com/jpoon/eece496.

Work in Progress

- **Mote Communication**

  Nano-RK implements a multitude of networking protocols including: RT-Link, WiDom, and b-mac. After some research, RT-Link seemed most suitable for the activity monitor due to its ability to provide bounded end-to-end delay across multiple hops and collision-free operation. RT-Link also has greater battery performance in comparison to WiDom and b-mac.

  After fixing several build issues with the Nano-RK RT-Link example program, I have managed to get several Firefly nodes to communicate with each other. However, it will require considerably more time to fully understand the protocol and optimize it for use with the activity monitor.

- **Accelerometer Drivers**

  Software drivers are required to process the input received from the sensor boards (e.g. accelerometers). Although I currently have written basic software to read information from the sensor boards, I am currently facing an issue where I am unsure of the meaning behind the values returned from the sensors. One would assume that if the node was stationary, the (x,y,z) values for the accelerometer would be (0,0,0); however, the accelerometer returns seemingly random values of (133, 82, 200). This problem can be solved via a reference document for the Firefly sensor boards or hopefully through individuals who have previously worked with the sensor boards (e.g. the Nano-RK contributors).

Work Remaining

- **Base Logic**

  The base logic will process the information obtained from the various sensors and notify the user interface (UI) of the changes in a person's motion. The main difficulty I foresee with this stage of development is the profiling of users; different users will have different thresholds for the various activities. The activity monitor system should be intelligent enough to work on all individuals.

- **User Interface (UI)**

  The UI will be the last major component of the activity monitor. Not much thought has been put invested in designing the user interface. The complexity of the UI will depend on the time available once the rest of the software components have been completed. The goal is to implement a graphical interface that abstracts the inner-workings of the program. However, if time is a factor, the UI could be as simple as prompting the user via minicom (serial communication).

Conclusion

Although I am roughly two weeks behind schedule, since receiving the hardware components, I have made significant head-way in developing the system and am confident in the punctual completion of the activity monitor system.

Jason Poon

Real-Time Operating System

Comparison of various real-time operating systems.

LiteOS

URL: http://www.liteos.net/

Mailing Lists: Two mailing lists -- one for general users, other for developers

- buggy installation
- **development of operating system focused heavily to "windows" users**
    - java installer
- small but helpful dev community

Nano-RK

URL: http://www.nanork.org/

Mailing Lists: <None>

- supports many hardware platforms: Firefly, micaZ

Indices and tables

- *Index*
- *Module Index*
- *Search Page*