ECE 4890 Spring 2013: Design Project RFPs

Revised 3/23/2013

Introduction

This document contains inputs received from the outside sponsors, faculty members, and other interested parties, with regard to senior design project ideas. The names of the submitters is available by request. When you have formed a possible design team, request project contact information from Dr. Wickert, and proceed to interview the submitter/customer for more details, and perhaps the formation of a preliminary requirements specification.

As more RFPs are submitted to me, they will be integrated into this document. Check back for additions and changes.

RFPs

Project 1: Hearing Assistance Device

Introduction

Currently an individual, who feels their hearing is in decline, must visit an Audiologist to have their hearing tested. An Audiologist will place the individual in a sound proof room, place a stereo headphone on the individual and send a sequence of tones to each ear of the headphone. The Audiologist will vary the magnitude (in dB) of these tones to determine the person's audio threshold at each test frequency.

A graph is produced which plots frequency on the x-axis and audio threshold (in dB) on the y-axis for each ear. This data can be used to design a hearing aid based on the individual's specific hearing needs or an audio compensated headphone for home use.

Product Description

This project will focus on an audio compensated headphone for personal use in watching internet TV or listening to internet radio.

The first task is to produce an interactive Windows GUI based program, which will measure a user's hearing from three hundred hertz to twenty kilohertz and will output a graph of the user's audio threshold for each ear. The user should be able to vary the test tone in 100 Hz steps. The user should be able to vary the magnitude of the test tone, within Federal regulations, to determine their audio threshold for that test tone. The interactive program will save and date the user's audio profile.

The second task is to generate a frequency compensated response, from current data or saved user audio profile, which should produce a flat ± 1 dB audio output response to the user over the three hundred hertz to twenty kilohertz audio band via the PC audio output port(s) without noticeable delay. Example, the user should be able to watch an Internet TV channel without noticing a disjointed connection between the audio output and the TV person talking.

Product Stretch Goals

A third task is to program a DSP or microcontroller device with the frequency compensated response generated in task two. The DSP or microcontroller device will be connected between the audio source (RCA or 3.5mm connections) and the headphones.

Requirements

- Determine the target PC's audio/sound card specifications.
- Determine if the required frequency range is obtainable using the target PC
- Project Sponsor(s) will determine if the target PC's audio/soundcard specifications are acceptable.
- Generate a Windows based GUI program to test an individual's audio threshold from 300 Hz to 20 kHz in 100 Hz steps. Subject to change based on the previous requirement results.
- The individual's audio threshold shall be measured in 1 dB steps.
- The GUI shall be able to display a graph of the individual's audio threshold for each ear on the same graph.
- The GUI shall be able to save an individual's audio threshold profile.
- The GUI shall be able to recall a saved individual's audio threshold profile.
- The GUI shall be able to produce a frequency compensation response, for use by the PC's audio/sound card, which will provide the tested individual a flat ± 1 dB audio output over the required frequency range.
- The GUI will be able to program the PC's audio/sound card with the frequency compensation response.

Constraints

The Team will need to research the audio frequency response of the PC's used at the University to determine if the required frequency range is obtainable.

The team will be provided with a stereo headphone for connection to the PC. All other required hardware should use existing University hardware.

Application

Improving hearing quality for individuals with minor hearing loss.

Areas of Engineering

Analog and digital signal processing, software development, and electronic circuit design.

What the Student Gains

Knowledge of consumer electronic product design and development.

Project 2: Seros Media Library

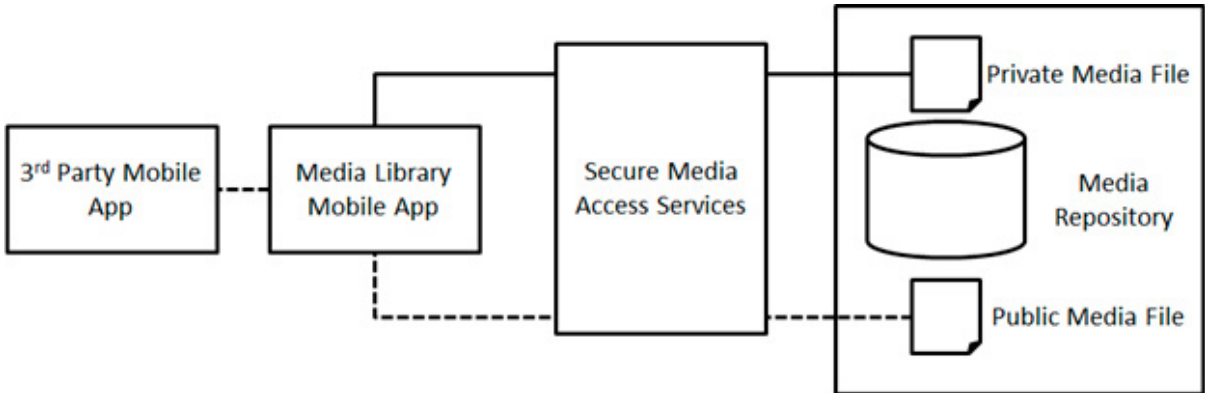
Problem Statement

Enterprises are struggling with the implications of the rapid rise of employees using personally owned mobile devices and applications to enhance productivity and job satisfaction. Most 3rd party apps are aimed at the end user rather than the enterprise, and they provide convenience features to integrate with social media, cloud storage solutions, and other apps on the device. For example, the native web browser on the device can open a presentation file on a web site and copy it to local storage or to another app that supports this file type. Once the file is saved, it is no longer in the control of the enterprise and could be viewed by unauthorized entities. As use of personal mobile devices and apps proliferate, the risk of sensitive data being leaked to unauthorized entities increases.

Within the enterprise, not all data is created equally. Some files, such as sales presentations, can be shown to multiple audiences while private/sensitive data, such as product plans, must only be accessible by authorized enterprise employees. Mobile Device Management (MDM) solutions provide management of features and data on the mobile device through features exposed by the mobile device operating system. Administrators can lock down the device or wipe the device to protect enterprise data. While these features provide strong controls, they can also prevent users from realizing the productivity and job satisfaction benefits of using mobile devices. A different solution is needed.

Product Required

A mobile app is required that provides users with the ability to view, edit, and save media files while also ensuring the files are only accessible by authorized users to protect private/sensitive data. Media file types includes, but is not limited to, documents, presentations, spreadsheets, video file, audio files, images, and PDFs. As shown in the figure, the mobile app would access the media repository though Secure Media Access Services. The lines show access paths and indicate that private/sensitive media files can be accessed by the app (the solid line) possessing the appropriate identity credentials, and public media files can also be accessed by the app and shared with other 3rd party apps on the mobile device (the dotted line). For example, a public presentation file can be downloaded and shared with a presentation program (e.g. Keynote or Kingsoft Office) or saved to the local storage.



The media access services would provide categorized lists of available files, options to search for files, and access to the media files. All features would be controlled by user identity to ensure users can only view and access files for which they are authorized. The mobile app must provide the ability to view common media file types. This app will also use a data classification assigned to the media file to control the actions users can take with the file.

Requirements

Requirements for the mobile app are as follows.

1. The mobile app shall provide capabilities to view media files including PDF files, video, audio, images, and at least one of the Microsoft Office formats (e.g. Word .doc or .docx, Excel .xls or .xlsx, or PowerPoint .ppt or .pptx).
2. The mobile app shall be written in Java and run on a mobile device running the most current version of the Android operating system.
3. The mobile app shall provide a list of media files available on a media server. For development and testing purposes, an Apache httpd web server or open source repository such as Apache Jackrabbit can be setup and used to access media files based on URLs.
4. Users must enter a username and password to access the app. This user identity will be sent to the web server to obtain files that the user is authorized to access.
5. The app shall show a list of files accessible by the logged in user.
6. When the user selects a file from the list, the file shall be pulled from the web server and rendered in the view.
7. If an accessed file has a public data classification, the user can view the file and save it to folders accessible by the mobile app.
8. If an accessed file has a private data classification, the user can view the file, but not save it to any folders.

Stretch Goals

The following items are stretch goals.

1. Support viewing of all Microsoft Office file formats.
2. Include an app for iOS platforms with an app written in Objective-C.
3. Build Secure Media Access Services to provide document lists in JSON or XML format instead of using a web server to access files.

Project 3: Seros Dashboard Creator

Problem Statement

Seros has created a Mobile Acceleration Platform (MAP) for both iOS and Android based mobile devices. One of MAP focus areas is the fast creation of mobile dashboards along with the ability to quickly connect with a variety of data sources. Dashboards for either the iOS or Android mobile devices are described by an XML document. The XML dashboard description documents are hosted on an Integration Server for download to a user's mobile device. Currently, the XML dashboard descriptions are created manually. Seros desires to have a Dashboard Creator Web Ap-

plication that provides a high degree of automation for creating new dashboard descriptions and updating existing dashboard descriptions. A mockup of what such a Dashboard Creator might look like is illustrated in the figure below.

Product Required

Seros requires a browser-based web application to automate the creation and editting of mobile dashboards and to automatically generate the XML-based dashboard description for subsequent use by the Integration Server.

Requirements

1. Create a web application similar to the mockup shown in the figure above. The mockup contains three different panels that must perform the following functions. Note the panels are described moving from left-to-right on the mockup.
   - Gallery Panel. In this panel, a library of dashboard gizmo images is displayed. The examples shown in the figure are a bar chart, line chart, pie chart, gauge/dial, process steps, and map. There are additional gizmos, such that this panel must have a scroll bar to allow the dashboard creator to scroll through the entire gizmo library. Each gizmo image can be "dragged and

dropped" into the Dashboard Panel.

- Dashboard Panel. This panel allows the "dragged and dropped" gizmos to be composed into a mobile dashboard. Each gizmo image must have the ability to be re-sized as required to fit the desired dashboard layout, and the dashboard panel must provide a realistic representation of how the dashboard will actually appear on the mobile device. The function of the Load, Preview, Layout, and Save buttons is described below.
  - Load Button: Loads a previously saved dashboard description XML file.
  - Preview Button: Shows the dashboard where functional JavaScript gizmos are displaying data read from the specified data source URL/URI. Note that this could be accomplished using a separate web application (see stretch goals below).
  - Layout Button: Select between formatting definitions unique to tablets and smartphones. For example, portrait and landscape for tablets or gizmo stacking for smartphones.
  - Save Button: Saves the dashboard configuration and properties specified by the Dashboard and Properties Panel as an XML-document compliant with an XSD schema. This includes the gizmo type, location and size; data URL/URIs, and color schemes.
- Properties Panel. This panel allows the properties to be defined and configured for the overall dashboard and for each selected gizmo that appears in the Dashboard Panel. It also allows for the security roles to be defined for the overall dashboard along with the appearance properties for the dashboard and gizmos. At a minimum, the dashboard creator must provide for all of the properties illustrated in the mockup figure.

2. The library of dashboard gizmos must come from the software libraries currently in use by Seros, and these libraries will be supplied upon acceptance of this RFP by a group of students.
3. The XML-based dashboard descriptions must comply with the XSD schema for dashboard descriptions. This schema will be supplied upon acceptance of this RFP by a group of students.
4. The web application must execute within the commonly used browsers such as FireFox, Internet Explorer, and Chrome.
5. The web application must provide a good user experience and be usable by people that don't have a deep technical background.

Stretch Goals

1. The Preview Button is considered a stretch goal
2. Save the dashboard description into a relational database where it can be subsequently transformed into an XML-file, JSON, or HTML5.

Preferred Implementation Technologies

HTML, CSS, and JavaScript with Java-based services as required. Additionally, any 3rd-party products/libraries must have a licensing agreement acceptable to Seros. Typically, Seros will not utilize products that do not have a "liberal license" policy, e.g., Lesser Gnu Public License (LGPL),

Apache Software Foundation, Massachusetts Institute of Technology (MIT), Eclipse, or Berkley. Note that Gnu GPL licenses are not acceptable.

Project 4: Sprinkler Control System

Introduction

This project involves remote access, monitoring and control using a PIC development board and some external electronics. Although the expressed purpose of this project is to develop the hardware and software for a sprinkler controller with 15 zones, the designer should consider modular solutions in order to easily adapt the design for other applications such as WiFi control of house lights and possibly to easily add modules for additional zone controls for future applications. Consider upgrade paths for BLE, Zigbee, ANT, ... sensors.

Project Description

1. Hardware is accessible over WiFi for status and control without compromising network security.
2. User interface application to run on either a Windows 8 PC or Android platform (choose one)
   - Provides mechanism to find, select and connect to WiFi networks.
   - Provides current hardware and software status including sensor states
   - Provides means of altering the program
3. Configure sprinkler cycle to begin at any of the following:
   - Water on either Even or Odd dates
   - Water every n days (n=0 to 7)
   - Water on any combination of (Sunday, Monday, Tuesday, … Saturday)
   - Water on command (now)
   - Water at set time and date
4. Each zone can be watered for a set duration or until a zone-specific logic signal is received, with a timeout. Zones are independent in that they can be set to overlap and each has its own termination condition and logic sensor. For example, zone1 may be set to run for 15 minutes on Tuesdays at 5am and zone9 may start (regardless of whether or not other zones are running) every other day at 5:05am and could be set to stop on the first of (a logic signal, 2 hours). This would allow a soil humidity (or other) sensor to be used to water any zone until enough water has been applied. In the case of sensor failure or other (unexpected) event, the 2 hour timeout would shut off the zone. For the purposes of this project, the zone controls should control LEDs and the logic signal should be some kind of manually operated switch.
5. There are two sensors which inform the entire system (not per zone) regarding
   - outside temperature (single point calibration at 32F (0C) is minimum requirement)
   - outside relative humidity (for the purpose of determining if it is raining)

Project Support

A PIC development board from Microchip will be provided (www.microchip.com). The team should select one which has WiFi and Eithernet capability as well as sufficient I/O for indicators and switches while remaining within the budgetary constraints. The budget for the project is $1,000 including any hardware (development kits/proto boards) or software. The prototype will go to the sponsor for use as a sprinkler system controller, including schematics and instructions for use as well as any source code; however, the design belongs to the team. Any intellectual property developed during the course of the project belongs to the team. Designs may be manufactured for sale to benefit the team as the team members see fit.

Project 5: Human-Powered BLE Pedometer

Problem Statement

The objective is to design and build a human-powered wireless pedometer that would measure activity, and optionally, heart rate and wirelessly transmit that data to a smartphone. The pedometer would use body heat (and/or optionally motion) to power the unit and Bluetooth Low Energy (BLE) to transmit the data. An optional display on the pedometer could display activity, power level, and time (synchronized from the smartphone). The smartphone (recommend iPhone) would require no hardware, but would require an application to collect, manage and display the incoming data.

Key Project Attributes

The project would utilize and demonstrate advances in energy harvesting and low power wireless communications and would require low power circuit design, wireless design, and software/firmware design.

Project 6: Wireless Shelf Display

Problem Statement

The objective is to design and build a shelf labelling system that uses long range battery-assisted passive UHF RFID to display product pricing and product information. The shelf unit would include an RFID chip to communicate with a host/reader system, a microcontroller and a low power display. The RFID chip could also be used to harvest energy from the RFID reader (and/or dedicated charger) field so that the product would not require traditional (non-green) batteries to power the display. A supercapacitor or other energy storage technology could be used. The reader system would send data to the display and collect diagnostic status from the display (i.e. error status and/or stored energy level/low battery).

Key Project Attributes

The project would utilize and demonstrate advances in energy harvesting and long range RFID and would require low power circuit design, wireless design, and software/firmware design.

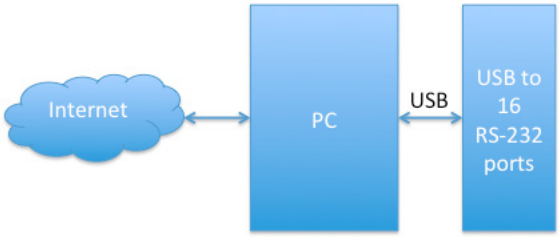
Project 7: Remote Operation of Battery Lab Thermal Chambers

Problem Statement

The High-Capacity Battery Research and Controls Laboratory (HCBRTL) at UCCS houses test and measurement equipment that can be used to gather data from battery cells, modules, and packs in simulated real-world operating environments (cf. http://mocha-java.uccs.edu/HCBRTL). There are presently nine environmental (thermal) chambers in the laboratory, which are used to keep batteries at a constant desired ambient temperature during testing.

Each of these chambers is presently operated manually using the front panel on the unit. This is not desirable because it requires that we travel to the laboratory to monitor performance and to change setpoints—we would also like the capability of changing the setpoints and monitoring chamber temperatures remotely. The chambers have serial RS-232 interfaces that can be connected to a supervisory computer to enable remote operation.

Therefore, we are seeking a solution as shown in the following block diagram:



The USB interface of a PC will connect to a commercial-off-the-shelf (COTS) USB-to-16-port-serial interface. The ports of this interface will be connected to individual thermal chambers. Software on the PC, written in C# (to be compatible with future plans for the laboratory), will monitor the thermal chamber operating points, and will command the thermal chambers to change their setpoints on user demand.

The PC will also host a web site where the present status of all thermal chambers is presented and continuously updated. The user can log in to this web site remotely, using authenticated login, and change the thermal chamber set points.

There are two major components to the design: (1) the Internet web-page interface to view current operating conditions and to change setpoints, and (2) the RS-232 chamber interface to query chamber temperatures and to change setpoints. This breakdown works naturally for a two-member team. For a three-member team, there is an additional requirement for an iPod Touch/iPhone HTML5 application that can monitor and operate the chambers.

Specifications

- The solution must work with Watlow 96 thermal-chamber controllers, Watlow F4 thermal-chamber controllers, and EZT-560i thermal-chamber controllers.
- The solution must work with up to 16 different thermal chambers simultaneously.
- The solution must be configurable with a simple text file (stating which ports are used, assigning names to chambers on each used port, identifying which thermal-chamber controller type

is associated with each port).

- Update rate for all monitored temperatures and setpoints must be at least 0.5 Hz.
- Internet access when changing setpoints and turning a chamber on/off must be secure. Network security must not be minimized.
- The code written to monitor and operate the temperature chambers must be written in C# and must operate on a Windows 7 operating system. The PC must use an Apache web server, and all hosted web pages must auto-refresh as data changes.
- The web interface must show all sixteen channels, names of chambers on each channel (if connected), present setpoint for each chamber, present operating temperature for each chamber, and must allow changing setpoints and turning chamber on and off.
- iPod Touch/iPhone application for optional third member: Must provide same functionality as web interface, but in an easier-to-use application.
- Stretch goal: Log error/alarm conditions when chamber is operating out of bounds.