Senior Design: High Level Design Report
The Dream Team: Andrew Carter, Matthew Elliott, Andrew Harms, Benjamin Keller
Due: 11/01/07

Introduction

Education is of paramount concern in society today. While homeworks, quizzes, and tests are effective means to measure learning, a real-time feedback system would offer another, possibly better, aid to the instructor. We propose a wireless classroom response system. To make the system more accessible to a wide range of students, our system will be based on open standards and open-source technology to reduce cost of the system. For the design, we must consider specific wireless and portable power technologies.

Problem Statement and Proposed Solution

Each group of students learns at a different pace and responds positively to different teaching methods. Measuring student understanding allows an educator to tailor his or her own style to a particular group of students, improving the learning experience for everyone involved. In the typical classroom, this measuring is accomplished through periodic homework assignments and targeting single students with questions. Utilizing these methods exclusively, several lectures could pass before the teacher notices a gap in student understanding.

Currently available devices that attempt to fill this need are flawed in key areas. A commonly used solution, einstruction(tm)'s classroom performance system (CPS), suffers from slow response, non-rechargeable batteries, and an expensive recurring subscription fee. With this hole in the market, it is possible for a small engineering team to develop an inexpensive and effective product, allowing the teacher to modify each lecture in real time, no matter the size of the audience.

The proposed system will have a response pad for each student as well as a response receiver. The response pads will have a user input that allows the student to enter the information that the teacher requests. Our user interface will also convey information concerning battery and unit state, such as "on" and "charging". The response pad will send information to the response receiver wirelessly. All wireless communication from the response pads will be routed to a computer through a wired connection by the response receiver. The response receiver will have a user interface in order to display configuration modes. The computer will collect, analyze, and display the information. In addition to wireless communication, the response pads will also connect to personal computers to register for classes. The response pads will be powered by rechargeable batteries. The response receiver will be powered by the wall outlet.

System Description and Block Diagram

The student response system contains two main system types: the response pad and response receiver. The response pad consists of a microcontroller, wireless transceiver chip, battery, battery charger, wired communication controller, and a user interface. The response receiver consists of a microcontroller, wireless transceiver chip, power device, wired communication controller, and a user interface. The response receiver and pads communicate with computers via wired communication. The response pads and receiver communicate with each other wirelessly. Each IC block (wireless transceiver, battery charger, etc.) within the response system devices must communicate with a microcontroller. Our generic block diagram is included at the end of the document (Figure 1).

The microcontroller will manage all the components of the student response system. It orchestrates the radio transmitting and receiving performed by the wireless transceiver chips on the response pads and receiver. It will control the operation of the battery charger as well as wired data transfer from response pads and receiver to computer. The microcontroller also handles user input as well as user output transferred by the user interface. This block will also include writing a significant amount of software.

Wireless communication will be used to link each individual response pad to the response receiver.

A battery charger IC will be used to charge a battery pack that is built into each response pad. Power must also be provided to the response receiver.

Wired communication between each system device and computer will be implemented. Response pads will need to communicate with computers in order to register for classes in which the student response system is used. This interface will be used to power the battery charger. A wired interface will also be used by the response receiver to transmit student response data to the computer.

The user interface of the response pads will consist of two components. Users input information to be sent by the response pad to the response receiver and eventually to the instructor. The user interface will also display information back to the user such as information to be sent and modes of operation. Modes may include "on", "charging", "transmitting data" and "ready to transmit". User data will be entered via keypads. The response receiver will also utilize a simple user interface to output configuration modes such as "on" and "receiving data".

Inter-chip communication will be implemented in order that each IC can communicate with the microcontroller. In this way the microcontroller can manage all the chips in each response device. Also the battery charger will use inter-chip communication to interface with the wired communication controller.

Software must be written for the computer in order to interface with the response pads and response receiver through wired communication. The computer will act as a charging station for the response pads and allow them to register for classes online. The computer will be used to collect, analyze, and display student feedback.

System Requirements

The final system will consist of multiple response pads, a response receiver, and accompanying computer software for classroom integration. Each of these pieces is integral to our overall system.

The response pads are the student interface to the system. The student will interact with the response pads via the LCD screen and the keypad. The LCD will display pertinent information and the keypad will allow input into the device, either answers to questions that are collected in class or information required for registering the device. The response pads will also interact with the rest of the system via the wireless link which will use the Zigbee protocol. The device will use USB to interface to a PC that can be used for powering the device and connecting to a computer for registration and firmware updates. Since the device also needs to be portable, it will include a rechargeable battery. Finally, the response pad will contain a micro-controller that acts as the brains of the device controlling the other components.

The response receiver is the central receiver that collects the data from the response pads, processes the information, and sends it to the teacher. The response receiver must have a wireless communication link that interfaces with the response pads, so it will also use the Zigbee protocol. A USB interface will act as both the connection to the computer and the power source for the device. Finally, the response receiver will need a microcontroller to control all components.

The accompanying computer software is the teacher interface into the system. The data is made available to the computer from the response receiver for viewing and/or presenting by the teacher through a USB interface. The software will also be able to interface into PowerPoint, or an equivalent program, for presentation to the class.

Microcontroller

The microcontroller orchestrates all the operation in our device. We need our microcontroller to have sufficient clock speed to manage all the devices and operations. Being a portable system, the response pad's microcontroller should be power efficient. Since the microcontroller must be programmed, it's language should be versatile yet comprehensible and intuitive. Because of multiple peripherals, we need sufficient input/output ports on our controller. Since the response pad could go into production, the microcontroller should be relatively inexpensive in bulk. Lastly, the type of packaging available for the chip is a concern, primarily in prototyping. We need to be able to easily troubleshoot and test the chip during the design process. Requirements for the response receiver are not as stringent since it will not use battery power.

Wireless Communication

Since our response pad is portable, wireless communication to the response receiver is a logical choice. The wireless protocol should be power efficient yet have sufficient range for our application. It should also be easy to interface to our microcontroller (this is more of a final part decision than protocol decision, however). IC Packaging is more of a concern here, since there are serious RF design decisions based around packaging. The protocol should be robust to interference, such as competing protocols on the same frequencies. Lastly, cost is a factor for major production.

Battery Charger and Battery

For portable power, a battery will be used. The battery should hold the requisite amount of power, along with current draw ratings on the order of what we need. A circuit will need to manage battery charging. It should be relatively easy to interface to our microcontroller.

PC Communication

There needs to be communication to a computer, both on the response pad and response receiver. This PC communication should be easy to interface to our microcontroller. The protocol should also be widely available on computers. The speed of the interface is a concern, especially on the receiver end to process question responses.

User Interface

The user hardware interface must supply the end user with relevant information during usage. The response pad needs to have a power efficient user interface, while the response receiver can use more power if necessary. The form factor of the interface is important for the response pad, since it should be a hand held device.

Inter-chip Communication

Given the large number of subsystems, there should be an efficient inter-IC communication protocol. It should be fast enough to process student responses.

Computer

We will write software to collect, analyze, and display student response data on a computer. Analysis may include things like totals, averages, and standard deviation. Results may be displayed in numerical or graphical form.

Future Enhancement Requirements

The most significant future enhancement is a direct interface to PowerPoint, or an equivalent piece of presentation software. This type of interface would make it easy for professors to integrate the feedback provided by our system into their lectures.

High Level Design Decisions

A block diagram of our high level design decisions is available at the end of this document (Figure 2).

Microcontroller

For a microcontroller, we will use an 8-bit microcontroller. This will provide us with more than enough computing power and speed. There are multiple companies offering different options in terms of speed, I/O lines, and instruction memory. The ubiquitous Serial Peripheral Interface (SPI) is included in most microcontrollers, so inter-chip communication is easily accomplished. Lower power consumption microcontrollers usually sacrifice features, but we have found microcontrollers that are suitable for our project. Most, if not all, modern microcontrollers have a C compiler available for programming, which may prove to be cost prohibitive. A free version would be the optimal choice.

Considering the wireless interface chip vendors, an Atmel microcontroller combined with the Atmel Zigbee radio makes the most sense. Code examples are readily available for interfacing the chips. A free C complier is available (avr-gcc) [3]. The cost ranges from ~$5 to ~$15 for single microcontrollers, with good price breaks for bulk orders (Digikey).

Wireless Communication

For a wireless communication protocol, Zigbee is the clear choice. It was designed for low power consumption, but still has a modest transmission range of hundreds of meters [1]. Of course, this test is not in a building, but we can expect similar line-of-sight results. A whole host of Zigbee transceivers are available, varying in RF power, system interface, and form factor. Moreover, each of these companies provide ample documentation and example code, so help is available during implementation. The cost of the transceivers is based on the options included. Our project calls for a "bare bones" IC, and cost is less than $10 (Digikey search "Zigbee").

Going along with our microcontroller choice, we anticipate selecting the Atmel AT86RF230. The radio has excellent qualities: Receiver sensitivity down to -101 dBm; transmit power of 3 dBm (2 mW); sleep current of .1uA; and 17 mA current draw at maximum power output [2]. One issue with this choice is the only available package is QFN (Quad Flat Package, No Leads). While this will make it harder to solder [4], a surface mount component with leads can cause spurious RF problems. In single units, the AT86RF230 is less than $6 (Digikey search, "AT86RF230").

Battery Charger and Battery

For the battery, we are going to use the Nickel Metal Hydride (Ni-MH) technology. Rechargeable Ni-MH battery packs provide the best compromise between cost effectiveness and capacity. 5 and 3V battery packs can be purchased for around $10 (digikey). Battery charging IC's compatible with Ni-MH are generally available for around $10 or less. Maxim makes battery charging ICs, and includes plans for USB charging [5].

Wired Communication

For computer communication, USB is a logical choice. USB is available on every modern computer. While the protocol is quite cumbersome, we believe the combined efforts of each team will make this successful. USB will also allow us to charge the response pad through the computer. The programming involved with implementing USB will require significant effort. Programming the microcontroller for USB will be less of a problem, but writing a driver for the computer will be difficult. We will utilize the Human Interface Device (HID) class drivers. These are normally intended for peripherals like keyboards, but one can design a generic HID that can transfer data via USB from our response receiver to the computer. These HIDs are well defined and universal to all computers. We will use the well-documented libusb on Linux (USB API) to interface with our application software.

User Interface

The user interface for the response pads will consist of a 4x3 matrix keypad and an 8x2 character LCD along with an LED. These UIs are small (hand-held size), cheap, and easy to interface to the microcontroller. The keypad is basically 7 wires and we have already seen how to interface an LCD with a microcontroller. Students will be able to respond to multiple choice questions as well as enter numerical answers with the keypad. It will be familiar, and easy to use for students. LEDs and seven-segment LEDs will not be able to provide enough students with enough information, so we have chosen an LCD with the ability to display characters.

Inter-chip Communication

We have chosen SPI because it is utilized in all the ICs we intend to use.

Computer

We have chosen to interface our system with a Linux OS as opposed to Windows because it is open source. The massive documentation on the internet will assist with writing the software applications to manage data and driver we need to use USB. Using an open-source OS is also consistent with the open-source nature of our project. We want to leave the project open for future development, so leaving an open system is appealing. That being said, we hope to add functionality for Windows or Mac OS X for usability in applications here at Notre Dame if time allows.

Major Component Costs

Response Receiver

Microcontroller: $5
Radio: $5
USB controller: $6
Extraneous board components(R,L,C,connectors,LEDs, etc): $15

Response Pad

Microcontroller: $5
Radio: $5

USB controller: $6
Battery charger: $5
Battery: $10
Keypad: $5
LCD: $8
Extraneous board components(R,L,C,connectors,LEDs, etc): no more than $15

We anticipate building three response pads. Conservative cost estimates are $31 and $59 for the receiver and pad, respectively. Our boards will cost approximately $100 (two board runs) and our development kit will cost $200 (with a subsidy from Prof. Schafer). Therefore, the total cost estimate for our system is $508. If our estimates were not high enough, we will only fabricate two response pads.

Conclusions and References

Because student feedback is essential for an instructor to pass information to the students as efficiently as possible, many traditional methods have been employed for many years. These include homework assignments and answering specific questions from students. The technology available today also makes another solution possible. Real-time feedback of student understanding can be accomplished through a student response system. While solutions do exist, they have several drawbacks that have prevented widespread use. Some of these problems include cost to the students and the classroom interface for the instructor. Our solution solves the problem of creating a viable, efficient student response system while also reducing student cost and leaving the system open to future development and improvement. Open-source solutions can keep cost to a minimum by making the solution open to the public. It can also be improved by others who have free access to all documentation and plans.

References

[1] : http://www.freescale.com/files/rf_if/doc/app_note/AN2902.pdf
[2] : http://www.atmel.com/dyn/resources/prod_documents/doc5131.pdf
[3] : http://winavr.sourceforge.net/
[4] : http://www.sparkfun.com/commerce/present.php?p=SMD-HowTo-1
[5] : http://www.maxim-ic.com/appnotes.cfm/an_pk/3241

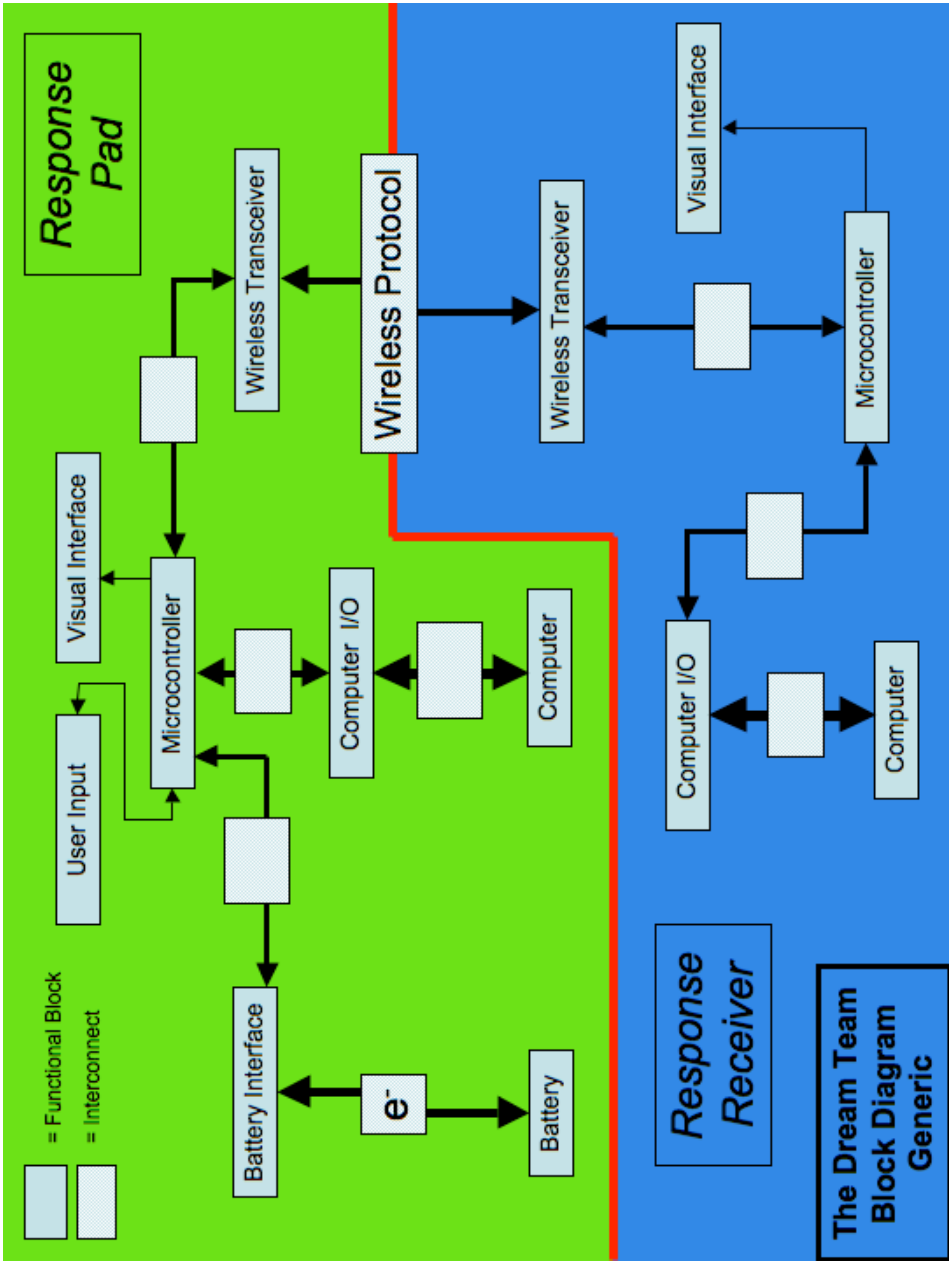

Figure 1: Generic Block Diagram

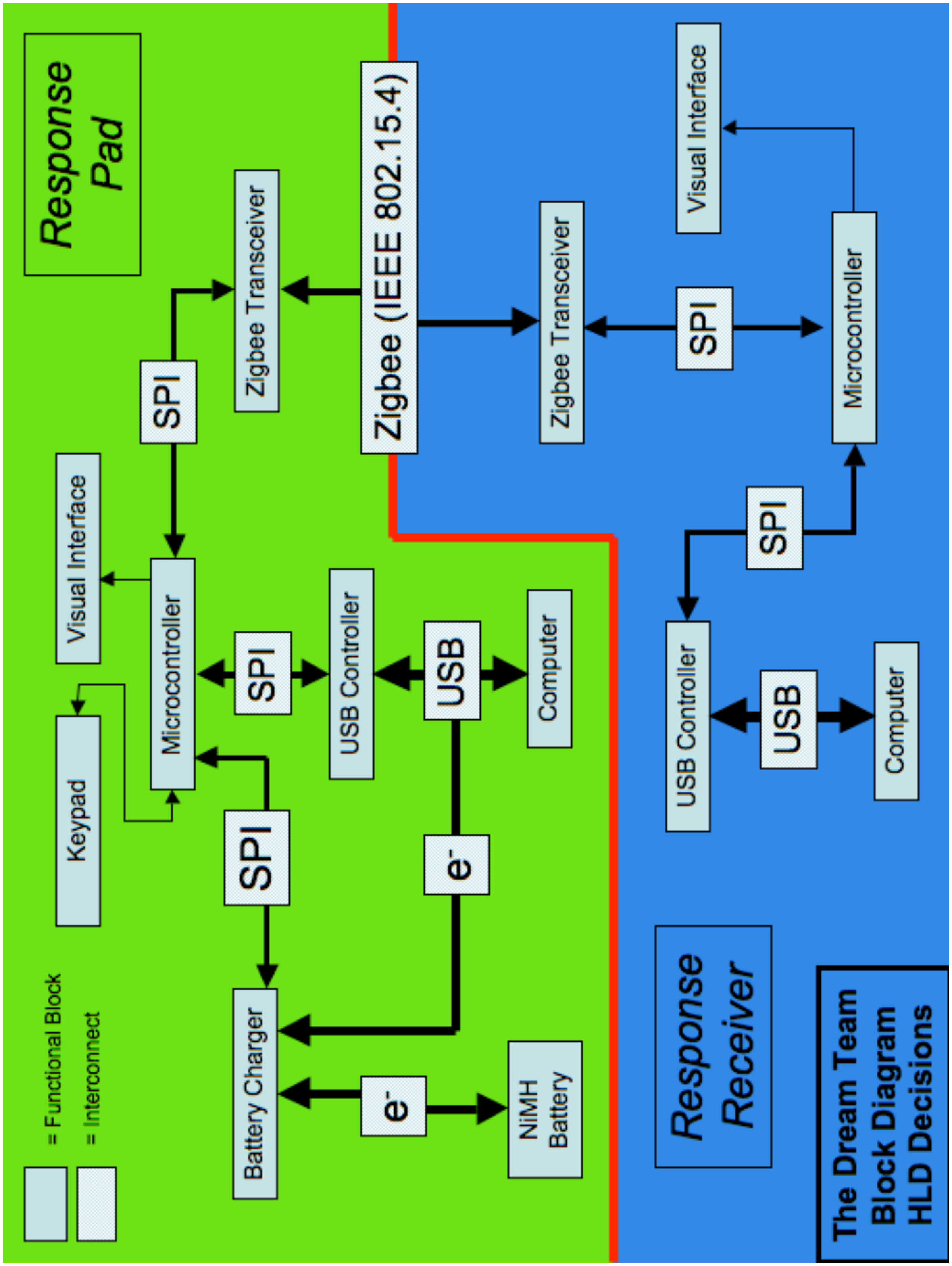

Figure 2: High Level Design Block Diagram