Jump Trading Medical Simulation Controller

Project Proposal

Jian Chen - Tanmay Mishra - Michal Rys

TA: Justine Fortier

ECE 445 Senior Design

Feb 6, 2013

INTRODUCTION

The Jump Trading Simulation and Education Center is a high-tech facility used train future medical doctors. The facility's simulators are currently controlled by software running on laptops. In order to make an adjustment to the simulation, the technician must tediously navigate the software's interface with a mouse and keyboard, making on-the-fly adjustments difficult. This project was selected to simplify such adjustments by designing a hardware interface into the system.

We are excited about this project because we found the subject matter, a biomedical device, to be a unique application of electronics. In addition, we found the significant software component to the project appealing since our group members have experience with implementing software solutions. This project gives us the real world experience of working with a customer, as well as the opportunity to conceptualize, design, and build a device that has potentially life-changing uses in the world.

OBJECTIVES

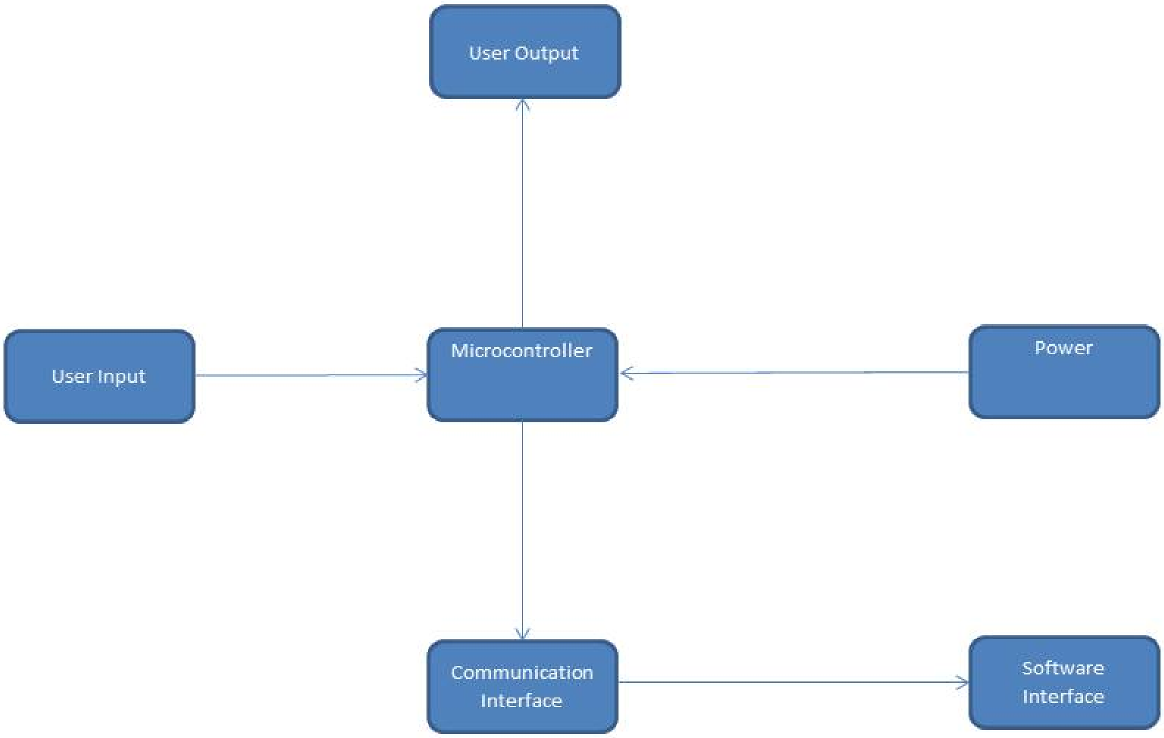
Benefits

- Provides an ergonomic solution to making on-the-fly adjustments
- Eliminates the use of software GUI as a form of making adjustments
- Maximises time spent on simulations, rather than making adjustments

Features (dependent on more details from Dr. Voznelik)

- Adjustment and display of heart rate
- Adjustment and display of blood pressure
- Adjustment and display of pulse
- Adjustment and display of additional parameters (as requested by customer)
- Retains settings after power cycling the device
- Interfacing with Laerdal (software used by Physicians)
- Allow to create and save presets (a possibility not yet corroborated by the customer)

DESIGN


User Output
User Input
Microcontroller
Power
Communication Interface
Software Interface

BLOCK DESCRIPTIONS

User Input:

- Capacitive touch surface, by using different finger gestures (horizontal/vertical swiping, multi-finger swiping, pinch/spread, and rotate), to adjust the value of parameters
- Communicates with the Microcontroller

Microcontroller:

- Receives signals from User Input
- Receives power from Power Unit
- Outputs relevant parameter values to User Output
- Outputs parameter values to Communication Interface

User Output:

- Displays the parameter being adjusted
- Considering 8-segment displays, led strips, lcd

Communication Interface:

- Delivers parameter values from microcontroller to software interface running on a computer
- Considering USB or wireless

Software Interface:

- Receive signals from Communication Interface and passes them to the Laerdal software
- Invokes software libraries from the Laerdal SDK
- Exact functionality dependent on tools available on SDK

Power:

- Power up Microcontroller and load up saved settings (if applicable) or load up default settings
- Considering usb or battery power

REQUIREMENTS

User Input:

The capacitive touch surface should be anti-scratch, sensitive enough to tell the difference of different finger gestures. It is also expected be to energy efficiency; could be powered by USB connection or battery. User input should have mechanism for creating presets

Microcontroller:

Microcontroller should in real time receive user inputs, display parameter on User Output, and send the parameters to Communication Interface (<150 ms). Microcontroller should be capable of retaining values after power down and power up of device, as well as retaining saved preset values.

User Output:

User Output should display signals from Microcontroller in real time (<200 ms from input time).

Power:

In order for usb power to be a viable option, entire system must use less than 2.5W (500 mA * 5V). If a battery-powered device is implemented, the device should work at least one month without battery replacement. However, this requirement depends on the specific use cases of the client. If the client requires high portability, the Communication Interface is going to be wireless. In this case, the device would be powered by 4 AA battries (2500mAh per battery). In the case that portability is not highly demanded, a USB cable would be used to power the device.

Communication Interface:

Device should be able to communicate with computer properly over the distance between the technician and computer. As mentioned with the power block, usb or wireless will be decided based on the use cases of the client.

Software Interface:

Should be able to interpret the signals from Communication Output, and communicate with the Laerdal software to make the adjustment. Should be able to make adjustment within 200ms of user’s input.

VERIFICATION

High Level:

When a user is using the User Input device (Capacitive Touch Screen), different gestures should be accurately mapped to the correspond parameter that is being adjusted. For instance, a swipe from left to right should correspond to the increase of the heart-rate parameter, while a swipe from right to left should be detected as a decrease in heart-rate. When adjusting a specific parameter, the real time value of the parameter should be accurately displayed on the display.

The adjusted parameters should also be accurately, through Communication Interface, transferred to Software Interface.

User Input:

Ensure that all types of gestures are being recognized by microcontroller. Ensure that latency is low (<100 ms) for delivery to microcontroller.

Power:

Ensure that sufficient power at proper voltages is being delivered to all components. If battery design chosen, run battery life tests to ensure device will run for at least a month under average usage (as defined by Jump Trading).

Microcontroller:

Ensure parameter values are being calculated correctly from input data. Ensure presets and persistent data are being saved correctly.

User Output:

Check if correct values are being displayed. Check if latency is sufficiently low (<200 ms)

Communication Interface:

Check if packets are correctly being sent and received. Check if latency is low (<100ms) for packet between microcontroller and computer.

Software Interface:

Properly interacts with Laerdal software to adjust simulation parameters. Exact tests will depend on functionality available through Laerdal SDK.

TOLERANCE ANALYSIS

Using the touch panel as user input runs the risk of accidental adjustments. For example, the technician might unintentionally introduce some upward motion into a horizontal swipe across the screen. We will design our device to maximise the tolerance of unintentional inputs from the user. We will first determine the average error a person tends to have when touching a screen, and keep these in mind during our implementation. After the design is complete, we will test the gesture tolerances by methodically performing them with a stylus.

COST

LABOR:

Member	$/hour	# of weeks	Hours/week	Total hours	Subtotal	(x2.5)
Jian	70	12	15	180	12,600	31,500
Tanmay	70	12	15	180	12,600	31,500
Michal	70	12	15	180	12,600	31,500

Total: $94,500

PARTS:

Parts	Unit Price ($)	Quantity	Total Price ($)
Microcontroller	10	1	10
Capacitive Touch Surface	50	1	50
USB 2.0 Cable	2	1	2
LCD Screen	20	1	20
Bluetooth module	20	1	20
Chassis	50	1	50

Total: $152

- **GRAND TOTAL = LABOR + PARTS = $94,652**

SCHEDULE

	Jian	Tanmay	Michal
2/3/2013	Confirm Jump Trading's requirement for the design	Work on project proposal	Work On project proposal
2/10/2013	Finalize options for Communication Device and Power	Finalize options of user input, order microcontroller and touch screen; start learning microcontroller programming	Confirm Jump Trading's requirements for the design, establish budget constraints with Jump Trading, acquire software SDK from Jump Trading; create repository for project code
2/17/2013	Testing input and power option on breadboard; design user output	Implement microcontroller code that reacts to touch input	Learn how to use Laerdal SDK, design software interface
2/24/2013	Implement and test Communication Device	Implement code that converts gestures into parameter values	Prepare for design review
3/3/2013	Prototype user output; debug and optimize hardware	Debug/Optimize microcontroller code	Implement software interface
3/10/2013	Design PCB	Preliminary design for Chassis	Debug/Optimize software interface
3/17/2013	Spring Break	Spring Break	Spring Break
3/24/2013	Fix issues in prototype device	Assemble PCB	Finalize Design of Chassis
3/31/2013	Assemble Device in Chassis	Ensure Completion	Fix remaining issues

4/7/2013	Fix Remaining Issues	Verification Testing	Tolerance Analysis
4/14/2013	Prepare Demo	Prepare Presentation	Prepare Paper
4/21/2013	Demo	Presentation	Final Paper
4/28/2013	Presentation	Final Paper	Check-In Supplies