# Project Proposal

# ECE 445 Senior Design
# Fall 2012

Names: Jacob Clifton, David Mckiernan, Ryan Schmid

TA: Rajarshi Roy

Team 4 - Jump Trading Simulation Interface

# Table of Contents

## I. Introduction

1. Title

***Jump Trading Simulation Interface***

Currently, clinical simulators (full-body, robotic electromechanical devices) rely on computer-based input through a laptop computer. The computer interface is a complex series of screen-based inputs that are typically handled through keystrokes and the use of a mouse. This project was chosen to design a new user interface that is easier to use than the current model.

### 2. **Objectives**

*Goals:*

Our goal is to design a new user interface that is easier to use than the current model. The interface would allow the technician to manipulate key data quicker and more efficiently, thus allowing for a more realistic simulation experience. In addition, the technician would be better able to monitor student performance instead of staring at a screen.

*Functions:*

Connect clinical simulators to our interface to create an easier method for an operator to interact with the simulation software.

*Benefits:*

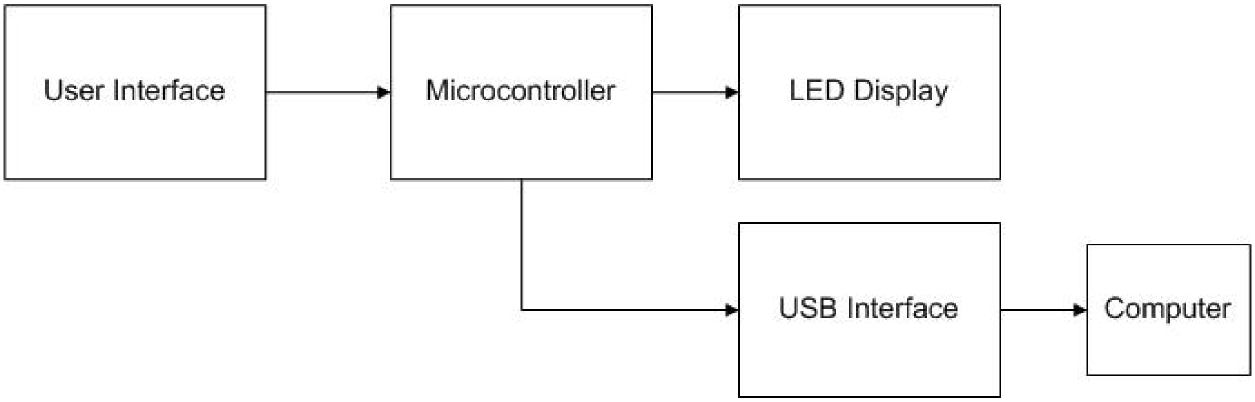
- Significantly easier for operator to change parameters of the simulation
- More intuitive than computer mouse
- Aftermarket so it can connect to any computer running the Laerdal software
- Allow the change of multiple parameter simultaneously
- Preset values will simplify the interface for the operator (resting state, tachycardia, etc.)

*Features:*

- LED display to show what each knob is controlling and display keypad entry
- Keypad for quick entry of parameters
- Toggle switches to select parameter to manipulate via knob or keypad
- Rotary knobs for sweeping parameters such as heart rate
- Buttons to execute programmed macros
- USB interface for easy connection to computer
- Runs on power provided by USB drive so there is only one cable to connect

## II. Design

1. Block Diagram



2. Block Descriptions

User Interface:

- Knobs, buttons, switches to specify parameters
- Communicates with the microcontroller

Microcontroller:

- Receives signals from the user interface converts to data output for PC
- Communicate with LED Display

LED Display:

- Display the parameter that is being modified
- Display keypad input

USB Interface:

- Provide a connection between the Microcontroller and PC
- Provides additional power for our system

3. Performance Requirement

Hardware

- Total current must be less than 500mA to allow USB to power device
- Switches, knobs, and buttons should accurately interact with the software
- LED display must refresh in less than 250ms

Software

- Response time from User interface to computer should be less than 500ms for real-time response
- LED display should display both the keypad entry and what each knob is controlling simultaneously
- Knobs should be able to sweep the entire range of values (heart rate 0-300bpm) but won't be sensitive enough to set a specific value because of the limitations of the PIC

## III. Verification

1. Requirements

User Interface:

- Needs to send Appropriate signals to the microcontroller for potentiometer, knobs, buttons, switches, and alpha numeric keypad
- Buttons need to be debounced
- Potentiometers need to have an output voltage from 0-5 volts

Microcontroller:

- Needs to accept signals from User Interface, LED Display, and USB Interface
- Needs to send appropriate signals to User Interface, LED Display, and USB Interface

LED Display:

- Needs to accept signals from the Microcontroller and display the correct representation on the screen

USB Interface:

- Needs to send signals from the Microcontroller to the computer and from the computer to the Microcontroller
- Needs to send the appropriate power to the Microcontroller, LED Display, and User Interface
- Needs to provide 500 mA of current for the entire system

2. Verification

User Interface:

- To ensure proper function of the buttons we will measure the output with an oscilloscope and ensure that there is no fluctuation in the output.
- Sweep the knobs over full range looking at output with a multimeter.  Values should increase linearly from 0-5 volts.
- We will need to verify that the 6 way switch output 6 specific values with a multimeter
- To verify the keypad we will need to ensure that each button outputs the correct voltage

Microcontroller:

- To test the microcontroller we will send basic high/low signals and have it process and output to LED's in order to test for correct functionality.
- Have a source sweep from 0-5 volts and change the brightness of the LED.
- To ensure correct functionality we will set the input parameters to known values and then ensure that output values match the expected output values.

LCD Display:

- To test we will hardcode characters and ensure that they display properly.

USB Interface:

- We will verify that the USB Interface is giving the correct amount of power to all the systems (e.g. the user interface needs 5 volts to operate correctly).  We can determine if it is functioning correctly simply by connecting a multimeter and determining if the output voltage is correct.

- We will need to determine that signals sent through the USB interface reach their destination and we can do this again with a multimeter. We can connect a signal on one side and determine if we see the appropriate output on the other side.

### 3. Tolerance Analysis

The potentiometers are the components that most affect the performance of the project. A noisy potentiometer will have an adverse effect on the parameters that we are trying set. For example, if one was to set the heart rate to 80 bpm and the signal was noisy, then the actual value to be displayed would fluctuate. We should be able to tune the value to +-2 units by changing the potentiometer (e.g. we want the heart rate to be 100 bpm so 98-102 bpm would be acceptable. If a specific value is needed the operator can simply use the numeric keypad). The potentiometer needs to operate in a range where 270 degree corresponds to the entire range of values.

## IV. Cost and Schedule

### 1. Cost Analysis

Labor:

| Member | $/hour | # of weeks | Hours/week | Total hours | Subtotal | (x2.5) |
|---|---|---|---|---|---|---|
| Jacob Clifton | $35 | 12 | 15 | 180 | $6,300 | $15,750 |
| Ryan Schmid | $35 | 12 | 15 | 180 | $6,300 | $15,750 |
| David Mckiernan | $35 | 12 | 15 | 180 | $6,300 | $15,750 |
| | | | | Total | $47,250 | |

Parts:

| Nomenclature | Part Number | Unit Price | Quantity | Total Price |
|---|---|---|---|---|
| PIC microcontroller | PIC16F887 | $2.63 | 1 | $2.63 |
| Linear 1K Potentiometer | 296UD102B1N | $1.60 | 1 | $1.60 |
| Push Button | PR141C1900 | $1.43 | 4 | $5.72 |
| LCD Screen | KS0108B | $19.95 | 1 | $19.95 |
| Rotary Switch 6 Pos | KC26A30.001NLS | $5.73 | 2 | $11.46 |
| USB 2.0 A Male to A Male | AE9930-ND | $3.19 | 1 | $3.19 |
| USB Connector | 292303-1 | $1.44 | 1 | $1.44 |
| Keypad -12 | COM-08653 | $3.95 | 1 | $3.95 |

Total - $49.94

## 2. Schedule

| Week | Date | Tasks | Team Member |
|---|---|---|---|
| 1 | 9/17 | Write proposal with team members | Jacob |
| | | Write proposal with team members | Ryan |
| | | Write proposal with team members | David |
| 2 | 9/24 | learn software for PIC and Laerdal software, ethical considerations | Jacob |
| | | Detailed electrical design, create parts list, ethical considerations | Ryan |
| | | PIC and perform simulations, ethical considerations | David |
| 3 | 10/1 | Completing tasks from previous week and preparing for design review | Jacob |
| | | Completing tasks from previous week and preparing for design review | Ryan |
| | | Completing tasks from previous week and preparing for design review | David |
| 4 | 10/8 | Begin coding the PIC | Jacob |

|  |  |  |  |
|---|---|---|---|
|  |  | Order Parts | Ryan |
|  |  | Work on setting up microcontroller | David |
| 5 | 10/15 | Continue coding and determine how to interface with the Laerdal software | Jacob |
|  |  | Assemble components on bread board | Ryan |
|  |  | Order a housing device for our system, and help with coding | David |
| 6 | 10/22 | Optimize code and continue working on software aspects of design | Jacob |
|  |  | Circuit debugging and help with software | Ryan |
|  |  | Create board layout and order PCB | David |
| 7 | 10/29 | Continue coding (Laerdal) and prepare for mock-up demo | Jacob |
|  |  | Continue coding (PIC) and prepare for mock-up demo | Ryan |
|  |  | Continue coding (Laerdal PIC) and prepare for mock-up demo | David |
| 8 | 11/5 | Prepare for presentation | Jacob |
|  |  | Prepare for presentation, assemble circuit on to PCB | Ryan |
|  |  | Prepare for presentation | David |
| 9 | 11/12 | Debugging code | Jacob |
|  |  | Debugging code | Ryan |
|  |  | Debugging PCB and getting everything within our housing | David |
| 10 | 11/19 | Enjoy Thanksgiving | Jacob |
|  |  | Eat into oblivion | Ryan |
|  |  | Enjoy Thanksgiving | David |
| 11 | 11/26 | Start writing final paper, prepare for final demo | Jacob |
|  |  | Work on presentation final touches, prepare for final demo | Ryan |
|  |  | Working on final paper and presentation, prepare for | David |

| | | | |
|---|---|---|---|
| | | final demo | |
| 12 | 12/3 | Final presentation, final demo, finish paper | Jacob |
| | | Final presentation, final demo, finish paper | Ryan |
| | | Final presentation, final demo, finish paper | David |