Medical Simulation Interface
Project Proposal

ECE445 Senior Design
Spring 2012

Tommy Shiou, Nick Cialdeli
TA: Justine Fortier
Project 28

Table of Contents

1.1 Introduction

1.1.1 Purpose

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 seeks to make the control of the simulators more efficient and accurate by redesigning the control interface.

1.1.2 Objectives

1.1.2.1 Goals

- Design a new control interface to more efficiently manipulate medical simulator values
- Communicate value changes with existing simulator software and hardware
- Display currently simulator values to user

1.1.2.2 Benefits

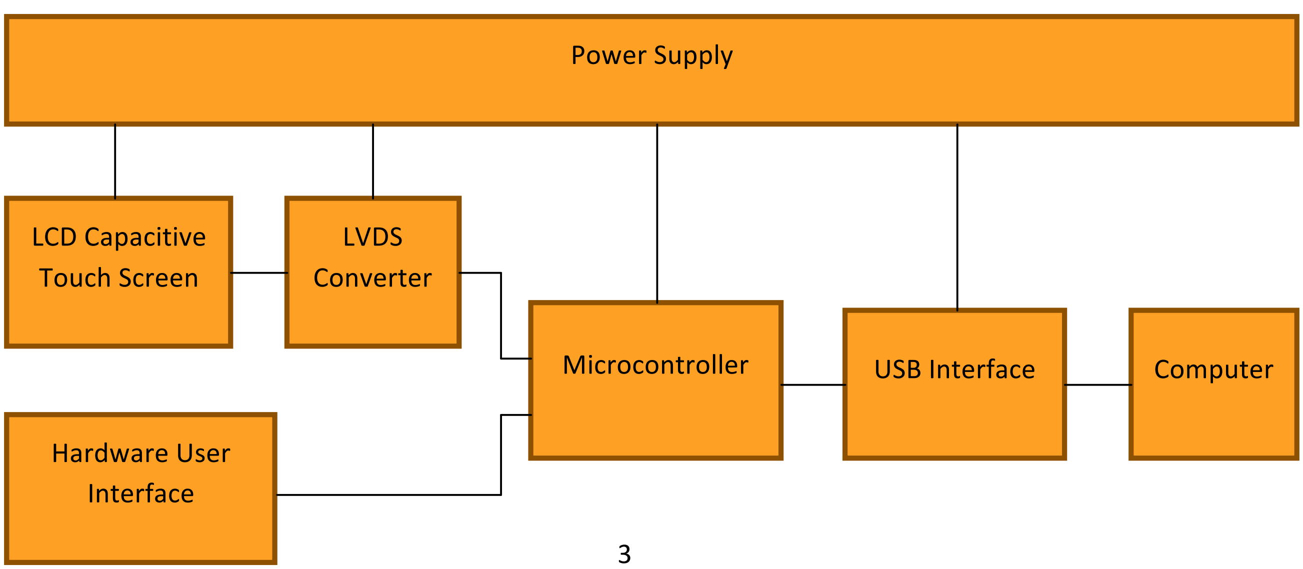
- More efficient control for quicker situational response
- Allows operator to focus on simulation instead of controls
- Easy visual access to current simulation values

1.1.2.3 Features

- Capacitive touch screen for intuitive display and control
- LCD screen to display simulation values
- USB communication with existing computer hardware
- Rotary dial to sweep parameters
- Hardware buttons with tactile feedback for mode control

1.2 Design

1.2.1 Block Diagrams



1.2.2 Block Descriptions

1.2.2.1 Power Supply

- Draws power from the USB interface
- Provides power for all modules

1.2.2.2 Hardware User Interface

- Includes rotary encoder dial for sweeping value changes
- Includes buttons for mode changes
- Communicates with microcontroller through GPIO

1.2.2.3 LCD Capacitive Touch Screen

- Takes data from the microcontroller and displays current values
- Communicates to the microcontroller of any value or mode changes
- Interfaces with the LVDS converter for microcontroller data

1.2.2.4 LVDS Converter

- Converts Low-voltage Differential Signaling communication interface from the touch screen and converts to data format readable by microcontroller HDMI port

1.2.2.5 Microcontroller

- Receives signals from the hardware and makes value adjustments to send to computer
- Receives signal from the touch screen and makes mode and value adjustments
- Sends values and data to display to LCD touch screen

1.2.2.6 USB Interface

- Provides communication between microcontroller and simulation computer

1.2.2.7 Computer

- Existing simulation computer

1.2.3 Performance Requirements

- Maximum power drawn from the USB port on the computer should be less than 2.5W
- Data communication from user input to simulation computer should be less than 250ms. Round trip display response to update screen data should be less than 500ms.
- Rotary encoder should have a resolution of at least 24 pulses per revolution

1.3 Verification

1.3.1 Requirements

1.3.1.1 Power Supply

- Must not overdraw power from the computer USB port
- Must provide enough power to all modules

1.3.1.2 Hardware User Interface

- Buttons must be debounced to avoid signal noise
- Buttons must output signal voltage within 0-5V
- Rotary encoder dial must output to grey code readable by the microcontroller GPIO

1.3.1.3 LCD Touch Screen

- Uses LVDS interface to communicate with microcontroller through the converter
- Receives touch input as an interface with microcontroller
- Displays simulation data from microcontroller

1.3.1.4 LVDS Converter

- Converts LVDS interface protocol over USB from the LCD touch screen
- Outputs to HDMI interface for communication with microcontroller

1.3.1.5 Microcontroller

- Outputs data convertible by LVDS converter
- Accepts signals from hardware user interface to make necessary simulation value adjustments
- Accepts touch screen data to change modes and data

1.3.1.6 USB Interface

- Send microcontroller signals to computer in order to interface with Laerdal software
- Provide necessary power to power supply

1.3.2 Verification Procedures

1.3.2.1 Power Supply

- Verify that the power supply will never draw more than 2.5 watts and 500 milliamps
- Verify that the power supply can supply up to 500 milliamps x 5 volts

1.3.2.2 Hardware User Interface

- Buttons will be monitored with an oscilloscope to ensure no signal noise or bouncing
- Verify all buttons operate in range 0-5 volts
- Monitor the rotary encoder using a logic analyzer to verify that it outputs grey code with at least 24 pulses per revolution

1.3.2.3 LCD Touch Screen

- Touch screen output can be monitored using a logic analyzer to ensure proper protocol
- Capacitive touch screen functionality will be tested through user trial
- Screen will be tested to display microcontroller's linux operating system

1.3.2.4 LVDS Converter

- The output will be tested with the microcontroller to ensure that the microcontroller can take control input from the touch screen and also output HDMI digital display data.

1.3.2.5 Microcontroller

- Output known digital display data from microcontroller HDMI port and verify it can be converted by the LVDS converter and display onto touch screen
- Verify that screen touch input can be read correctly to control microcontroller OS
- Use a power supply to verify that microcontroller GPIO pins are operating correctly and are readable by microcontroller operating system
- Verify state of buttons connected to GPIO pins are read correctly by the microcontroller OS

1.3.2.6 USB Interface

- Verify communication words are read and sent correctly to Laerdal software
- Verify that the USB interface can provide at least 2.5 watts and 250 milliamps of current at 5 volts to power supply

1.3.3 Tolerance Analysis

The tolerance of the touch screen should be within 5mm or else the touch input from the user will be ambiguous. We hope to use the 10” screen real estate efficiently so the smallest graphic button will be about 20mm x 20mm and assuming that a fingertip will not be larger than 10mm in diameter, this means that for accurate use the touch screen should be at least able to register the fingertip within 5mm on either side. We can test this by finding a specific point on the touch screen and have different people’s finger press on the point and monitor the registered position signal to see if the measured position is within 5mm of the desired point.

Power supplied to all components in the system should be around 5V because most modules uses USB power. This means that even at maximum current draw the power supply should be able to supply the necessary 5 volts with a tolerance of about 0.25 volts. This can be monitored by attaching a multimeter probe with history functionality (Labview) and ensure that the voltage supplied doesn’t drop below 4.75 volts and doesn’t go above 5.25 volts.

1.4 Cost and Schedule

1.4.1 Cost Analysis

1.4.1.1 Labor Cost

Member	$/hour	# of weeks	Hours/week	Total hours	Subtotal	(x2.5)
Thomas Shiou	$35	12	20	240	$8,400	$21,000
Nick Cialdella	$35	12	20	240	$8,400	$21,000
				480	$42,000	

1.4.1.2 Materials Cost

Item	Part Number	Unit Price	Quantity	Total Cost
10.1” LCD Touch Panel	776-MGG1010AU12	$155.36	1	$155.36

Rotary Encoder	COM-10982	$3.95	1	$3.95
Flatlink Transmitter	SN65LVDS93A	$12.04	1	$12.04
PanelBus DVI Receiver	TFP401A-EP	$21.60	1	$21.60
Raspberry Pi	RASPBRRY-MODB-512M	$35.00	1	$35.00
SD Card	MMBTF04GWBCA-QME00	$10.28	1	$10.28
				$238.23

1.4.2 Schedule

Week	Date	Task	Team Member
1	2/4	Write proposal with partner	Thomas Shiou
			Nick Cialdella
2	2/11	Finalize requirements with Jump Trading, contact Laerdal for software, prepare for mock design review	Thomas Shiou
			Nick Cialdella
3	2/18	Learn Laerdal software for interfacing with simulations, ethical considerations	Thomas Shiou
		Begin electrical hardware design, finalize parts list, ethical considerations	Nick Cialdella
4	2/25	Finish previous week's tasks, prepare for design review	Thomas Shiou
			Nick Cialdella
5	3/4	Master Laerdal software SDK	Thomas Shiou
		Display converter PCB layout, order board & parts	Nick Cialdella
6	3/11	Begin coding microcontroller, individual report	Thomas Shiou
		Mechanical housing solution complete, individual report	Nick Cialdella
7	3/18	Spring Break!	Thomas Shiou
			Nick Cialdella
8	3/25	Prepare for mock-demo, finish microcontroller coding	Thomas Shiou
		Prepare for mock-demo, assemble hardware components	Nick Cialdella
9	4/1	Prepare for mock-presentation, begin Laerdal software interfacing	Thomas Shiou
		Prepare for mock-presentation, ensure proper functionality, verify safety precautions with device	Nick Cialdella
10	4/8	Continue Laerdal software interfacing, run tests	Thomas Shiou

		Run tests, package device in housing	Nick Cialdella
11	4/15	Finish Laerdal software interfacing, endless testing	Thomas Shiou
		Endless testing, debug, perfection	Nick Cialdella
12	4/22	Demo and final presentation, final report document	Thomas Shiou
			Nick Cialdella
13	4/29	Final report, SUCCESS!	Thomas Shiou
			Nick Cialdella