ELEN90066 Embedded System Design Project

Hand held gaming console, firmware and game

Overview

During this semester long project you will design, build and test a hand held gaming console and write firmware for it.

The project will involve:

1. Circuit design and schematic capture
2. Design and layout of a Printed Circuit Board
3. Develop firmware
4. Hardware assembly
5. Debugging and testing
6. Application development
7. Product demonstration
8. Time management

Your console will have up to 9 push button inputs, an LCD screen and LEDs as outputs. The device will have firmware that implements at least one application for sketching. The sketch application permits the user to draw on the screen using 4 direction buttons. It is up to you to design any other applications you wish to implement.

During the semester you will be marked on your progress and at the end of semester you will present your completed console. Your marks will be based as follows:

1. Schematic Layout 20%
2. PCB Layout 40%
3. Quality of your code 20%
4. Functionality of your Console 20%
5. Bonus applications 10%

The project is worth 40% of your final mark and the exam is worth 60%.

Work individually

You are expected to do <u>part</u> of this project individually. It is very important that you understand all the complexities of creating an embedded system. It gives you the opportunity to take full ownership of the project and fully develop your skills in all aspects of embedded system design. Do not copy your work from others and do not leave your project on hard drives of shared computers, ensure that your files are saved in your private directory.

Time Management

Time management is a fundamental engineering skill. Developing hardware has many hard time constraints, and creating good hardware takes time. Hardware takes time to design, time to fabricate, time to assemble and time to debug. Producing hardware in the last minute does not work!

A mistake in the specification of the project can create a solution for the wrong problem (complete waste of time). A big mistake at the schematic layout level will result in time to find the problem when debugging, fix the problem in the schematic, redesign the PCB and rebuild your prototype (large waste of time). Making a mistake with software can usually be patched quickly and easily (small waste of time).

A further challenge is when a problem emerges you do not necessarily know at which process layer the problem is at (hardware design, fabrication, assembly or software) and that takes time to find. Sometimes the fix for a problem can be made at a different process layer (software patch for a hardware problem).

With hardware you must start early!

Deadlines

In this subject you are responsible for your time management. There are two unmovable deadlines:

End of week 7 you **must have your PCB submitted**. If you miss this date you will need to organise your own PCB fabrication (this will cost you about $40)

Last week of semester whatever you have completed is what will be marked **in your scheduled workshop class**. If you were not able to get your device to work you can still get partial marks for various sections of your circuit that are working.

Other Dates

Marking of your schematics begins **week 4** and can be marked up until **week 5** of semester. It is in your best interest to get your schematic marked before you start your PCB layout.

Marking of your PCB starts in **week 7** and can be marked until the last week of semester. It is in your best interests to get your schematic marked before you submit your PCB for fabrication.

Resources around marking time get very scarce, if you leave it to the last minute you will have to wait in a long line to get your work marked. Get in early!

Workshops

Workshops start in week 2. You must complete the Occupational Health and Safety (OHS) lab induction before you will be permitted into the labs.

Workshop resources are scarce and in this subject we endeavour not to waste any of it. You are allocated to one workshop session but you may go to any scheduled ELEN90066 workshop session if there is free space.

If you are scheduled for a workshop you have priority over other students. If there are not enough seats in the class, the demonstrators will ask non-allocated students to leave starting at the front right corner of the room (direction faced while seated) and heading across then back. If you are not scheduled for that class try to get the back left seat.

Functional Specification v1.0

The device will meet the following specifications

1. Power

1.1 The device will be powered from one 1.5v AA battery.

1.1.1 The battery holder will be mounted on the PCB.

1.2 The device will boost the battery voltage to 3.3v using a boost converter

1.3 The device will have a two pin 2.54mm header, one pin connected to the 3.3v line the other pin to ground to power during debugging.

1.4 The device will have a two pin 2.54mm header, one pin connected to the 1.5v line the other pin to ground to power during debugging.

1.5 The device will have a single ground pin connected to ground to facilitate the connection of the ground wire of the oscilloscope. This pin will be far away from any other pins such that the alligator clips do not risk shorting other components to ground.

1.6 A green LED will illuminate if the power is on.

2. Power (Bonus)

2.1 The device will have a soft power switch.

2.1.1 The device will power on when the power push button is pressed and the device is in the off state.

2.1.2 The device will power off when the power push button is pressed and the device is in the on state.

2.1.3 The device will use less than 1uA of standby current.

3. Push Button Inputs

3.1 The device will have up to 9 push buttons.

3.2 There will be one button for power.

3.3 There will be four buttons for left, right, up and down.

3.4 There will be at least one action button.

3.5 There will be one button to reset the microcontroller

3.6 There may be two more additional action buttons.

3.7 The buttons will be debounced using an RC circuit.

3.8 Multiple simultaneous button presses should be detectable.

3.9 Circuitry will be included to generate a single interrupt when any button is pressed.

4. Battery Voltage Level Input (Bonus)

4.1 An ADC channel will be connected to the battery to detect its voltage.

4.2 A red LED will illuminate if the power level is lower than 1.1 volts.

5. Touch Screen Input (Bonus)

5.1 The device may have a resistive touch screen.

5.2 The touch screen will be attached to the device using a 4 pin zif connector.

***Adding this onto the PCB makes it substantially harder to route and could cost you more marks than this bonus is worth. Only top students with spare time should attempt this. Touch screens must be purchased from Mouser (790-EATOUCH102-1) and connector (790-EAWF10004S). Bonus points are also available if demonstration of control is made using 2 potentiometers instead of the touch screen.

6. Display Output

6.1 The device will have an output consisting of a 102 x 64 pixel LCD screen.

6.2 The output will have a backlight.

6.3 The backlight will be variable brightness.

6.4 The device will have a green LED to indicate power output from the regulator.

6.5 The device will have a red LED controlled by the microcontroller.

7. Device Size and Layout

7.1 The device will be made of a PCB 78.5mm by 44mm (use the template provided).

7.2 The device will have an ergonomic layout of keys to make it easy to use.

8. Processing

8.1 The device will use an ATMEGA16L running at 8Mhz for processing.

8.2 The device will have 4Kbit of FRAM.

Layout Suggestion

Top Layer

- uC
- Battery Holder
- Circuitry
- JTAG header and other headers

Bottom Layer

- LCD
- Push Buttons
- LEDs

Components

You will be required to purchase a kit of parts, information on how is available on the subject LMS site.

The following components are included in your purchased kit for use in this project

0805 resistors 10k, 1k, 15, 30 and others

0805 capacitors 100nF

2 x 0805 capacitor 18pF

4 x 0805 capacitor 1uF

1 x 12010 capacitor 22uF

1 x AA Battery Holder

9 x SMD Tactile Push Button Switch

1 x Low Voltage Synchronous Boost Regulator SC120

1 x ATMEGA16 Low Voltage Microcontroller ATMEGA16L-8AU

1 x IND 4.7uH 1210

1 x LCD Screen

1 x LCD Backlight

1 x Inverting D flip flop 74AUC1G80

1 x 8 input Nand Gate 74AHC30

1 x 9 push buttons

1 x FRAM FM25L16B

2 x N Channel Mosfets 2N7002P

1 x 8Mhz Crystal Oscillator

The following components are included in your purchased kit for the programmer

1 x 7.3728Mhz Crystal Oscillator

1 x ATMEGA16 Low Voltage Microcontroller ATMEGA16L-8AU

1 x FT232RL USB to serial bridge

2 x 10k, 1 x 470R, 2 x 270R Resistor

1 x Ferrite Bead Inductor

1 x 10 pin Boxed Right Angle connector

1 x USB B Type Connector

3 x 0805 LED

4 x 1uF, 2 x 18pF, 7 x 100nF, 1 x 10nF capacitors

1 x PCB

Software Interface Specifications v1.0

1 Power Up

1.1 When the device powers up the screen will be cleared

1.2 If the device only has the sketching program it will load at boot up

1.3 The device will be fully booted within 2 seconds of boot up

2 Power Up Bonus

2.1 The device will boot up with the previous user program and state loaded

3 Sketching Program

3.1 At execution the screen will be cleared

3.2 At execution a single pixel 32 on the short edge and pixel 51 on the long edge will be activated.

3.3 When the up button is pressed the cursor moves one pixel up while leaving the previous activated pixels active.

3.4 When the left button is pressed the cursor moves one pixel left while leaving the previous activated pixels active.

3.5 When the right button is pressed the cursor moves one pixel right while leaving the previous activated pixels active.

3.6 When the down button is pressed the cursor moves one pixel down while leaving the previous activated pixels active.

3.7 When the clear button is pressed the screen is cleared and the cursor is moved back to the middle of the screen.

3.8 If the backlight is off and the back light button is pressed, the back light will illuminate.

3.9 When the back light button is pressed, the back light will reduce in brightness by 10%.

3.10 When the back light button is pressed and the back light is at 10% illumination the back light will be turned off.

4 Further programs and interfaces (Bonus)

4.1 If there is more than one program it is up to the programmer to work out a usable interface for selecting the programs and other controls that may differ to the specifications in 3.