EE445L – Lab11: Final Embedded Lab

Harley Ross and Dalton Altstaetter

5/2/14

OBJECTIVES

EE445L- Lab8: Requirements Document

Harley Ross and Dalton Altstaetter

3/6/14

1. OVERVIEW

1.1 Objectives:

Our team decided to do a project that involves the user interacting with the system using an accelerometer. The purpose of our project is to better develop our skills using user inputs to and changing our outputs based on those inputs.

1.2 Roles and Responsibilities:

Dalton will develop the GPIO files and timer interrupts. Harley will develop the finite state machine and the higher level functionality of the system. We will develop the hardware and PCB together to better understand how to design the software. The clients of our project are the TA's and judges of the Open House.

2. FUNCTION DESCRIPTION

2.1 Functionality:

The system will produce outputs on a LCD that are based on the inputs that the accelerometer receives from the user. The outputs of the LCD will change according to the accelerometers measurements in acceleration and angles.

2.2 Performance:

The system will have to measure angles and forces made by the user, and then change the output quickly enough for the user to react to those new outputs.

2.3 Usability:

The output interface will be a LCD screen. The inputs will be the measurements read from the accelerometer. The inputs and outputs for our design will vary based on the cost of each.

3. **DELIVERABLES**

   3.1 Reports:

   The reports we will write are the Schematic, PCB Layout, Measurement Data, Testing Procedures, Testing Data, and Analysis and Discussion.

   3.2 Outcomes:

   3.2.1 Objectives

   1-page requirements document

   3.2.2 Hardware Design

   Regular circuit diagram (SCH file)

   PCB layout and three printouts (top, bottom and combined)

   3.2.3 Software Design

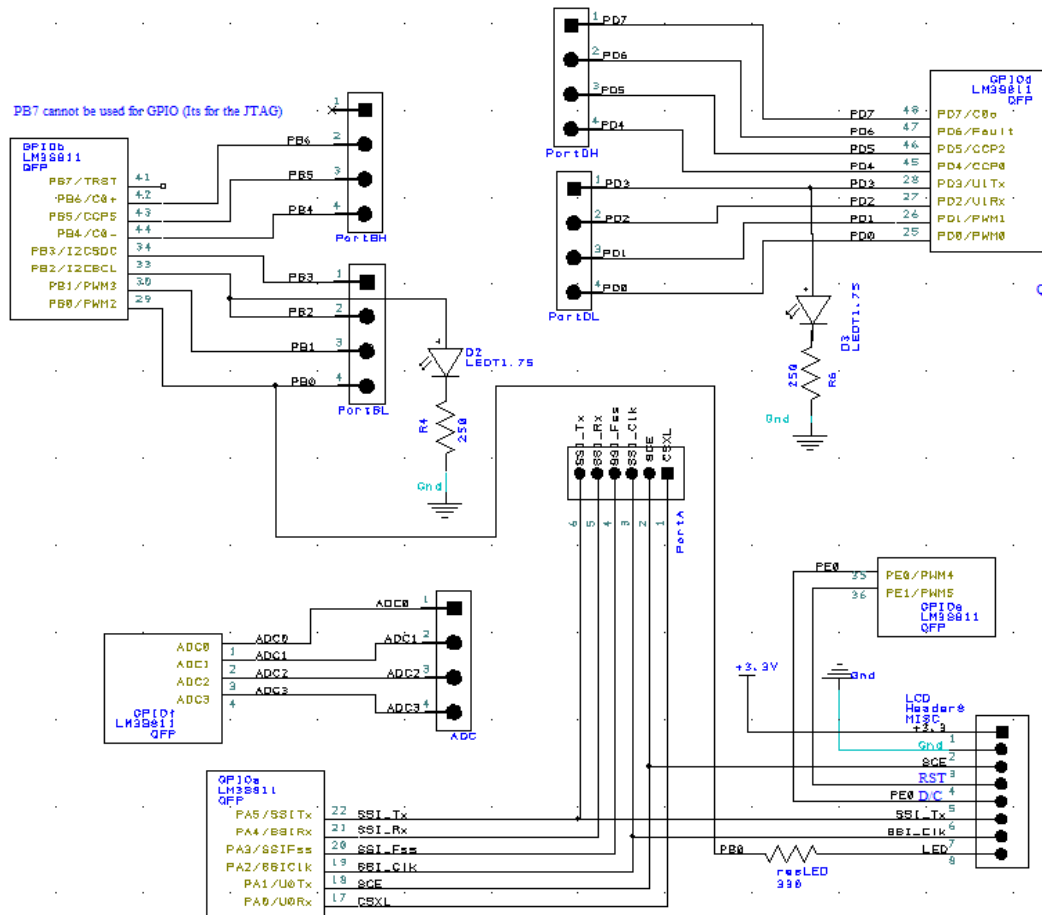   Include the requirements document (Preparation a)

   3.2.4 Measurement Data

   Give the estimated current (Procedure d)

   Give the estimated cost (Procedure e)

   3.2.5 Analysis and Discussion (none)

HARWARE DESIGN
SCH

PB7 cannot be used for GPIO (Its for the JTAG)

GPIOb
LM3S811
QFP

PB7/TRST	41
PB6/C0+	42
PB5/CCP5	43
PB4/C0-	44
PB3/I2CSDC	34
PB2/I2CBCL	33
PB1/PWM3	30
PB0/PWM2	29

PB6 — 2 — PortBH
PB5 — 3
PB4 — 4

PB3 — 1 — PortBL
PB2 — 2
PB1 — 3
PB0 — 4

D2 LEDT1.75
R4 25Ω
Gnd

PD7 — 1 — PortDH
PD6 — 2
PD5 — 3
PD4 — 4

PD3 — 1 — PortDL
PD2 — 2
PD1 — 3
PD0 — 4

GPIOd
LM3S811
QFP

PD7	48	PD7/C0o
PD6	47	PD6/Fault
PD5	46	PD5/CCP2
PD4	45	PD4/CCP0
PD3	28	PD3/U1Tx
PD2	27	PD2/U1Rx
PD1	26	PD1/PWM1
PD0	25	PD0/PWM0

D3 LEDT1.75
R6 25Ω
Gnd

Question About PE1, it is used as a reset for an input in the

SSI_Tx
SSI_Rx
SSI_Fss
SSI_Clk
SCE
CSXL
PortA

ADC0 — 1 — ADC
ADC1 — 2
ADC2 — 3
ADC3 — 4

GPIOi
LM3S811
QFP

ADC0	1
ADC1	2
ADC2	3
ADC3	4

PE0 — 35 — PE0/PWM4
PE1 — 36 — PE1/PWM5

GPIOe
LM3S811
QFP

+3.3V
Gnd

GPIOa
LM3S811
QFP

PA5/SSITx	22	SSI_Tx
PA4/SSIRx	21	SSI_Rx
PA3/SSIFss	20	SSI_Fss
PA2/SSIClk	19	SSI_Clk
PA1/U0Tx	18	SCE
PA0/U0Rx	17	CSXL

LCD
Header6
MI3C
+3.3

Gnd	1
SCE	2
RST	3
D/C	4
SSI_Tx	5
SSI_Clk	6
LED	7
	8

PE0
PB0
resLED
390

SOFTWARE DESIGN

Accelerometer

The accelerometer software uses the Serial Peripheral Interface (SPI) in mode SPO = 1, SPH = 1 at a SPI clock speed of 10 MHz to communicate with the 9-Degree of Freedom (DOF) LMS9DOF which has a 3-axis accelerometer, a 3-axis gyroscope, a 3-axis magnetometer, and a temperature sensor to communicate with our LM3S1968/LM3S811. The accelerometer software just used the accelerometer since the gyroscope data wasn't used to calculate orientation. The LMS9DOF stores the x, y, and z axis data in 3 separate 16-bit registers as a two's complement number. The accelerometer software continually reads these registers at a rate of 760 Hz as set by the initialization of the accelerometer and sends these three 16-bit registers to the microcontroller to be processed to find orientation. To find the orientation of the accelerometer (in fixed-point ranging from 00.00-90.00 degrees with a resolution of $\Delta = 0.01$) I used the C library function from "<math.h>" to call "atan2(y,$\sqrt{x^2+z^2}$appropriated (for my fixed-point math) to turn it into the appropriate angle, this was done for each axis. I then used these angles to set thresholds for the expected motions of the golf swing, the backswing, forward swing, and follow-through. Upon the end of the follow-through I captured the x and y axis accelerations (measured in units of "g's", earth gravitation units) and normalized them to convert them to vectors to pass to the LCD display, where it would then repeat the cycle for the next swing, if necessary.

Nokia_LCD

The software to run the Nokia LCD used SSI0 to initialize and send signals to the LCD. From there, bitmaps were created to print an outline of the boundaries for the miniature golf course. This outline was copied to a two-dimensional array that holds all of the boundary locations and ball locations. The ball two dimensional array was reprinted to the screen to show ball movement. The moveBall function called for an X and Y vector component values. From those values, the ball would move up or down by one depending on the Y component and right or left depending on the X component. The function would keep preprinting the movement components became zero. If the position of the ball was near a wall, the function would reverse the direction of the ball to replicate bouncing. The function would also detect the hole where the ball user wanted the ball to go. If the ball's location was near the hole, the LCD would print the next level and restart the balls starting position.

ANALYSIS AND DISCUSSION
Our end goal for the project was to combine the accelerometer software with the LCD to create a miniature golf game. Our team spilt the work up between the LCD and the accelerometer to get the work done faster. Upon merging our code, our team found bugs with the merging that lead to problems with the LCD. Our PCB design has only one SSI port so we planned on having to reinitialize the ports every time we needed to switch between the accelerometer and the LCD. Our LCD ended up being shorted when the LM3S811 chip was soldered on. Because of this, our team was not able to complete our project using the PCB.