Cornell University
School of Engineering
Department of Electrical and Computer Engineering
ECE 6931, Final Report

Relaxation Gaming Wrap

Prepared By Members**:**
Wei Wang (ww387), Yuan Cui (yc969)
Consultant:
Bruce Land
Date of Submission: 20 May 2014

Abstract

The topic of our Master of Engineering project is a relaxation wrap aiming to come up with a solution that will relax gamer's hand and wrist without impacting gaming experience. As Electronic Sports is becoming increasingly popular nowadays. There are many professional gamers participating in various tournaments every. As a matter of fact, most professional games require a mouse as primary source of input. However, using mice, especially moving and clicking for an extended period of time, brings undesirable wrist conditions.

Traditionally, people use passive devices such as soft pads as a solution. However, this is not a good fit for rapid hands motion required by the enthusiasm nature of electronic sports. When gamers are under extensive stress, such a pad solution will prevent gamers from moving their hands freely and cannot massage their hands at all since the pad is pressed with high pressure. Thus, the design team is looking to find an active solution to mitigate gamer's health concern.

This project be based on microcontroller. This project will include development with respect to both hardware and software. During our design implementation, the design team will always keep in mind that gamer's convenience should always be considered instead of developing a pure massage machine. And the final goal of the design project is that a device that could provide sufficient treatment to user's hand and wrist should be provided. At the same time, gamer could also enjoy the stressful electronic sports without being negatively impacted.

Executive Summary

We are developing a solution to this issue using infrared ray and massage treatment system that is controlled by microcontroller. The system will be able to monitor real-time gamer's wrist condition and provide treatment without making noticeable interruptions so that the gaming competition experience is not affected. And detailed implementation will involve embedded hardware design and C/C++ coding through real time kernels. The final goal we want to achieve is that the relaxation will produce helpful treatment by IR or massage so that the carpals/pisiform and hand muscle never feel painful or stressed after hours of intensive mouse movements. Furthermore, this invention can also help people such as programmers and designers who uses mouse throughout working hours relax their wrist.

This report will cover all design details about this M.Eng project. A more detailed introduction including the related background information will be provided. And project plans and concerns will also be introduced in the first section. After that, high level analysis about how this project should be implemented will be shown. A block diagram illustrating all block information will be provided attached with explanation from view of circuit level schematic. Then, all components will be breakdown into parts associated with reason why it is chosen to be a fit for this design project.

The detailed design section will explain all design concerns, problem solving, implementations and debugging with diagram and data table. Both the hardware and software related issues will be discussed in this section. For example, this report shows all design restrictions on hardware while showing how the software code design will allow all functionalities to be achieved. In the end, results about this report will provide data to prove the functionalities of design initiative. Test data and diagrams will mainly be provided for sensing part and treatment part. Based on our findings, we will discuss whether our design is successful or not and our experience of learning on this M.Eng project. In addition, we will also show possible future improvements of this design project.

Acknowledgement

Wei Wang and Yuan Cui will take this opportunity to express our profound gratitude and deep regards to the Cornell University for support toward this degree program and resources for this engineering project. We would also like to take this opportunity to express a deep sense of gratitude to Bruce R. Land for his cordial support, valuable information and guidance, highly informative lectures, and constructive opinions, which helped the design team to have an in-depth understanding of engineering design and debug. His suggestions on design concepts, considerations and experiment methods ensured the project prototype to meet the functional and non-functional requirements of the design.

We are also obliged to staff members of ECE 5010, ECE6930, and ECE 6931 for providing valuable information about fourth year design procedure and making comments on the team's deliverables. We are truly thankful for their work of providing such useful courses during the period of this fourth year design project.

In the end, we thank Digi-Key and Amazon for their reasonable pricing and shipping services. We also thank our friends and classmates who have provided helps on purchasing and commenting on design process, without which this master of engineering design project would encounter more difficulties.

Wei Wang, Yuan Cui

Glossary

Embedded System: An embedded system is a computer system designed for specific control functions within a larger system, often with real-time computing constraints.

Microcontroller: A small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

ADC: Analog-to-digital converter, a device that converts the input's continuous physical quantity to a digital number that represents the quantity's amplitude.

Accelerometer: An accelerometer is a device that measures acceleration in all dimensions. However, the proper acceleration measured by an accelerometer is not necessarily the coordinate acceleration.

Pulse Width Modulation: Pulse-width modulation (PWM), or pulse-duration modulation (PDM), is a modulation technique that conforms the width of the pulse, formally the pulse duration, based on modulator signal information. Although this modulation technique can be used to encode information for transmission, its main use is to allow the control of the power supplied to electrical devices, especially to inertial loads such as motors.

APM: action per minute, used to compute how fast gamers are giving actions to the game controller such as keyboard and mouse.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Background

Electronic sport, also known as E-sport, is a form of competition where players compete with other players using electronics as a medium, mostly a computer, to achieve certain outcomes. It emulates sports because players are distinguished by their decision-making abilities and reflex while competing. This form of sports is becoming increasingly popular enabled by the development of electronics and technology in the past decade. More and more people, especially younger generations, are interested in E-sports because of its convenience to compete and intellectual stimulating experience. As a result, many electronic brands often sponsor E-sports tournaments for advertisement, and the prize pool for E-sports is increasing rapidly. For example, Valve hosted this year's Dota2 tournament for a total prize pool of 3 million dollars shared by top 8 teams. As the money begins to pour into E-sports, many people devote themselves as professional gamers whose job is to train and perform at their best level. [1]

Although E-sports are becoming increasingly professional, compared to normal professional sports, they are still lacking in some fields. For example, the players often do not have dedicated medical staff to care about their health. E-sports often impose significant health issues to their participants. Notable health concerns include poor vision resulting from long time of playing in front of a monitor and skin issues resulting from using a mouse extensively during completion.

1.2 Functional Requirement

To ensure the basic functionality of the design, certain requirements are needed and are described in the following table.

Table 1. Functional Requirement

Priority	**Name**	**Measure**	**Description**
Must	Pressure Sensing	Sensors	Sensors will be used to monitor the pressure from hand to table. The sensor should be sensitive

			enough to measure in time pressure information.
Should	Heating	By design	The heating will be considered to be a tentative treatment to help the blood flow. However, due to the ambiguity in various levels and method of heating, the choice of design may change during design process.
Must	Control Algorithm	Microcontroller	The control algorithm needs to determine the running state of the entire wrap system to ensure correct information is taken and the corresponding treatment orders are given.
Must	APM Count	Circuit	The APM, which stands for action per minute, should be collected correctly to find out how fast is the player moving his or her fingers.
Must	Movement Measurement	Circuit	The system should also be able to successfully measure how fast and where is the hand moving to gain more information about the hand's motion.
Must	Motor Control	Microcontroller	The motor, which is selected to be part of treatment method, has to be precisely controlled by the microcontroller.
Must	ADC Functionality	Microcontroller	The ADC functionality of the microcontroller must be properly

			programmed so that analog sensing information can be read correctly.
Should	Signal Integrity	Circuit	The circuit layout design must achieve satisfied level of signal integrity so that there is no error in communication.
Optional	PuTTy Communication	Microcontroller/ Communication Bus	PuTTy communication could be a useful tool for debugging and future software extension design.
Must	Power Consumption	By design	The overall power consumption must be limited to the level that an USB port could suffer, which is 5V and 500mA.
Optional	LCD Display	By design	Outputting information such as ADC reading could be displayed on a LCD for debugging purpose.

1.3 Non – Functional Requirements

Despite those functional requirements, there are also non-functional requirements that must be considered during the design of this enhanced photovoltaic charging system.

Table 2. Non-functional requirements

Priority	**Factor**	**Requirement**
Must	Location	The whole gaming wrap design must ensure all electronics must be smoothly attached to hand.
Must	Size	The size of final PCB that will be on hand should not exceed the size of hand.
Must	Operation	The gaming wrap must operate as expected without hurting gamer's gaming experience.

Should	Usability	Controller should provide recommendations about the hand's health condition.
Should	Cost	The cost of this project should be kept to a relatively low price as it will be considered to be a peripheral product to gaming health.
Must	Fault Protection	Although not expected to happen, the system may run into fault condition where units stop functioning as expectation. Under such circumstance, the whole system should output error message and change the state to idle.

1.4 Cost

The prototype cost will be significantly higher than the production cost as evaluation boards need to be purchased and there will be some waste of components due to potential failure of tests.

Table 3. Table of estimated cost

	Prototype	Hand-Assembled	Mass-Produced
Quantity	1	>=5	>=100
Estimated Cost per unit	$50	$20	$10

1.5 Block Diagram

To solve issues described in previous sections, we determined to design an embedded system using a real time kernel based on Atmel microprocessor. The project will mainly include 3 parts. The high level block diagram is shown below.

Figure 1. System Level Block Diagram

The sensor will be responsible for taking "user health condition input". It will measure the pressure, blood flow etc. to determine whether user need treatment or not. Then it will feed such information to the microprocessor (MCU) for decision making. And the MCU will perform real-time calculation to determine what the reaction should the treatment devices perform. We will build a prototype to prove the feasibility of this embedded system. For treatment testing approach, we will perform user feedback analysis to study the efficiency of the treatment we provided. The current design will use massage and infrared ray as treatment methods.

For the prototype, we expect all design blocks described above meet the specification. The sensors will generate readable digital/analog signals to the microprocessor. The microprocessor should output correct information corresponding to designated input. Furthermore, the treatment device should also respond correctly according to microprocessor output. We then could prove the functionality of the project is implementable. Moreover, for the final product, we will select test users who are exposed to the health risks mentioned previously and collect feedbacks from them. We expect high positive feedback rate to demonstrate that our design idea and implementation is successful.

1.6 Risks

When designing this gaming wrap system, the main risk and concerns are for safety and efficiency. For example, due to the danger of heating IR, which could be a potential risk for skin or eye, safety has to be considered for this design project.

Risk 1 (high priority): All electronics related safety concern.

Solution: consult with our advisor Bruce Land to look for feasible solution and the correct way to use all project related electronic devices such as IR emitter and vibration motors. So, there is no potential risk such as hurting user.

Risk 2 (high priority): Overall power consumption of the system may exceed the power availability of USB port.

Solution: consider the main energy consumption components that are not necessary in the system and research ways of having a sleep mode for these components and if there is no sleep mode, then these components would be discarded.

1.7 Budget

This Table of budge is only for estimation. Selection of devices might change during detailed design and will be explained further in following sections of this design report. [2]

Table 4. Budget information

Product Name	**Description**	**Source**
Atmega 1284p development board	A development board for Atmel Atmega 1284p microcontroller with easy access to pins and USB.	ECE 4760 Lab
Accelerometer	To measure how fast is the hand moving	Digi-Key
Pressure Sensor	Build by Carpal Tunnel pressure resistor.	ECE 4760 Lab
IR Transmitter	IR module to detect finger motion	ECE 4760 Lab
IR Emitter	Used to heat the surface skin of hand to help blood flow	ECE 4760 Lab
Vibration Motor	Provide vibration to hand dead skin area for treatment	Ebay
Miscellaneous	Small capacitors, resistors, wires, wire clippers, and other small materials for designing and testing the system	ECE 4760 Lab
Symposium poster	For presentation in the ECE Day	Mann Library

Table 5. Budget estimation (for development)

Product Name	**Qty**	**Unit Price**	**Required**
Atmega 1284p development board	1	$0.00	$0.00
Accelerometer	1	$13.99	$13.99
Pressure Sensor	1	$0.00	$0.00
IR Transmitter	1	$0.00	$0.00
IR Emitter	1	$0.00	$0.00
Vibration Motor	1	$3.99	$3.99
Miscellaneous	1	$10	$10
Symposium poster	1	$28	$28
Total Expense			**$55.98**

1.8 Project Plan

Table 6. Project Plan

		January				February				March				April				Hours			
#	Tasks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	Yuan Cui	Wei Wang	Total
1	Hardware Research	4	6																4	6	10
2	Hardware Try-out			6	6														6	6	12
3	Hardware Integration					2	2	2	2										4	4	8
4	Software Design							4	4	6	8	8	8						14	24	38
5	Software Verification										2	2	2	4					6	4	10
8	Hardware Software Integration													10	10	10			16	14	30
9	Demo Verification														6	10	10		10	16	26
10	Prototype Demo																4	10	4	10	14
	Total Hours																		64	84	148

2 High-Level Analysis

2.1 Detailed Block Diagram

Figure 2 below shows the high level block diagram of the relaxation gaming wrap with some details of device and circuit design. Since the goal of this project is to improve the gamer's health condition under extensive practice and without impacting their gaming experiences, so the idea is that we need proper sensing and treatment systems that are successfully controlled by the microcontroller. As mentioned in previous sections, the power supply will be 5Volts and we need to minimize the total power consumption. Various sensors are placed in circuits to monitor the gamer's operation. And the microcontroller will drive the treatment system including the IR emitter and vibration motor with correct driving circuits. The communication method is currently determined to be UART to USB for debugging purpose at this moment.

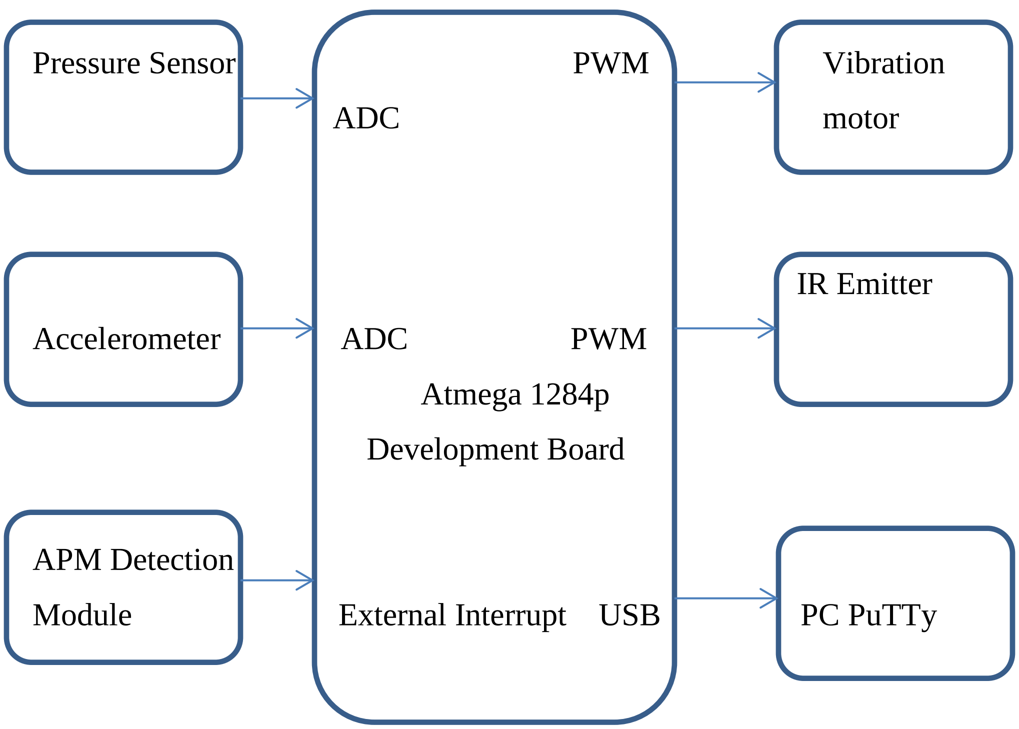

Figure 2. Detailed High Level Block Diagram

2.2 Components Breakdown

As indicated in previous paragraphs, there are various components and devices shown on the schematics that are not clearly explained their functionalities and purposes. The following sections of the report will clearly describe the reasons why such devices are chosen for this relaxation gaming wrap project design.

2.2.1 Microcontroller (Atmega 1284p)

Atmel Atmega 1284p is used to implement this project. The design team gained practical experience with development based on this microcontroller from ECE4760 course works. Thus, this microcontroller is more convenient for prototype purpose. On the other hand, it has all functionalities that will be used for this project design including analog to digital converter, pulse width modulation, and external or internal interrupt handling.

Atmega 1294p is a microcontroller that runs with 16MHz clock frequency. Thus, it will also provide sufficient speed for control algorithm that will be used. Moreover, the easier access of using real time kernel is also a plus for choosing this microcontroller because we are able to rune tasks in parallel to achieve real time monitoring and treatment. We have also gained practical experience with such a kernel from ECE 4760 taught by Bruce R. Land.

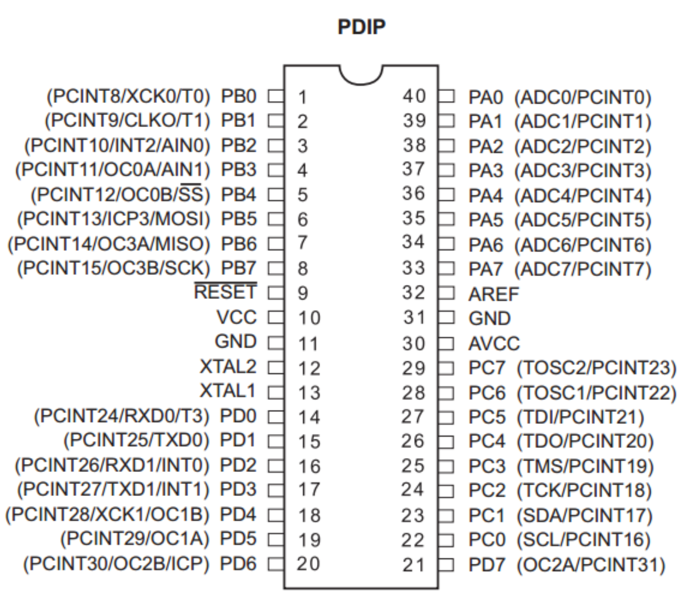

Figure . Pinout of Atmega 1284p

2.2.2 Accelerometer (MXR2999EL-B)

The MRX2999EL-B is a 0.5g plus or minus X and Y dimension analog accelerometer. It provides better than 1mg resolution with variable VDD fitting the range of 3VDC to 5.25VDC, where this design uses 5VDC for VDD. The temperature shift is negligible as it varies 0.4mg for very degree C. It also provides excellent linearity and the cost is reasonable. All those factors make the design team to select this module as the accelerometer for this project design. Below is the detailed block diagram of this accelerometer module.

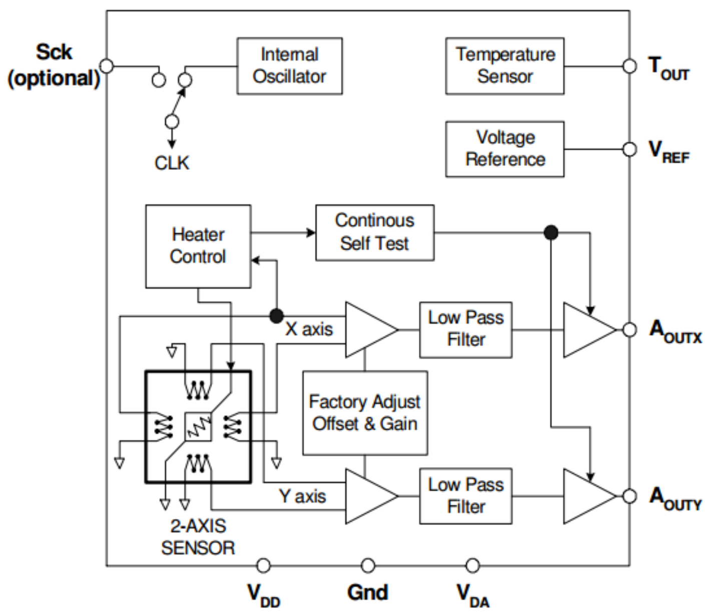

Figure 3. MXR2999E Functional Block Diagram

2.2.3 Vibration Motor (Parallex-28821)

After discussion with Bruce R. Land, we decided to use vibration motor as part of the solution for the treatment system that provides massage functionality. The motor of choice mainly follows the cost concern. With respect to various pricing reports, we finally determined to use the vibration motor from Parallex-28821, which is also commonly used for cell-phones.

This motor has a nominal voltage of 3.0VDC and 90mA current maximum, thus causing 270mW power at most. The rated speed is 9000 rpm and the starting voltage cannot be lower than 2.3VDC to allow proper usage of this motor. The connection is fairly simple in comparison to other devices in this project as only positive and negative pins are provided.

2.2.4 IR Emitter and Transceiver system (TSAL6400/LTE(R))

The Infrared ray is used for two purposes for this design project with different modules. The purposes are heating the corner of hand to increase the blood flow of the hand area of interest. On the other hand, the IR transceiver system will be responsible for collecting the rising edges caused by finger motions so that the microcontroller will take this information and convert it to APM value.

The TSAL6400 is used as a tentative solution for the IR heating purpose. This emitter will produce significant heat when the apply voltage is higher than 2.5VDC and burned out under 5VDC. For safety concern, as the output power of microcontroller pin is low enough that the connected TSAL6400 is unable to operate under unsafe circumstance; we decide to use this IR transmitter as a temporary solution for the design choice.

On the other hand, by suggestion from Bruce R. Land and course material of ECE 4760, we decided to use LTE4208/LTR4206 as the IR transceiver package. For more accuracy, they have to be setup under proper circuit environment and will be shown in detailed design section.

Figure 4. Picture Demo of Treatment Setup

2.2.5 USB Communication (FT232RL)

The FT232R is a serial UART to USB interface with single chip USB to asynchronous serial data transfer. The data transfer rates from 300 baud to 3M baud for RS422, RS485, and RS232. The supply voltage allows 3.3VDC to 5.25VDC and it is USB 2.0 fully compatible. This USB communication interface is chosen to be part of the ECE 4760 Atmega 1284p development board designed by Bruce R. Land.

With careful examination for this module’s datasheet and experience of using this interface module for various USB communication purpose, we make sure that this serial to USB transceiver only fits to the development board for ECE 4760 and for our own design. The figure below briefly describes the pinout and the functionality provided by this USB module.

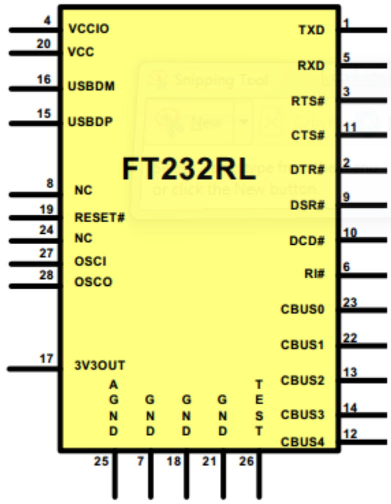

Figure 5. Pinout of FT232RL

2.2.6 Op-Amp (LM358)

Op-Amp is used in the IR transceiver circuit setup. The choice for this device is much simpler as there are lots of common choices. For example, the LM358 from Texas Instruments is a feasible solution as it provides large DC voltage gain about 100 dB and 1MHz unity Gain bandwidth. The VDD requirement is between 3VDC to 32VDC and drains less than 500 uA current. All these fits into the design requirement and the performance is satisfactory.

3 Detailed Design

In this section, both detailed hardware design and software design will be discussed. The hardware design part includes design of successfully and proper way of connecting components mentioned previously and the PCB layout design discussion for future product purpose. As mentioned previously, explanations about design details on various components such as sensors, microcontroller and communication will be provided too. Related register configuration for various hardware components will be shown in hardware design but not in software. On the other hand, the software design section will mainly focus on the coding of the Atmega 1294p microcontroller. Details about interrupt service routine, real time kernel and even breakdown to initialization, control logics and power saving related coding will be explained.

3.1 Detailed Hardware Design

3.1.1 External Interrupt

The external interrupt functionality of Atmega 1284p is used to collect user finger motions. We call it action per minute (APM) in professional gaming. The purpose of using this method is to count how rapidly users are using the mouse's click button. Thus, we want each finger click to trigger a counter programmed in the microcontroller to count it as an action and we will collect the data every 5 second to perform minute estimation. If such a value is high enough than a certain threshold, we consider the user to be rapidly playing games.

Such an interrupt method is achieved using the LTE4208 as IR transmitter and LTR 4206 as IR receiver. The circuit setup below took the reference from ECE 4760 labs that created by Bruce R. Land. [3] The receiver will take the reflection ray from the phototransistor and feed it to the op-amp. The op-amp will amplify this signal to the level that the microcontroller could take. Each finger click will cause receive to receive a ray that will be transferred to a high signal to the microcontroller. And a rising edge will be generated due to the rise time of the receiver and the Atmega 1284p will take it as a rising edge external interrupt.

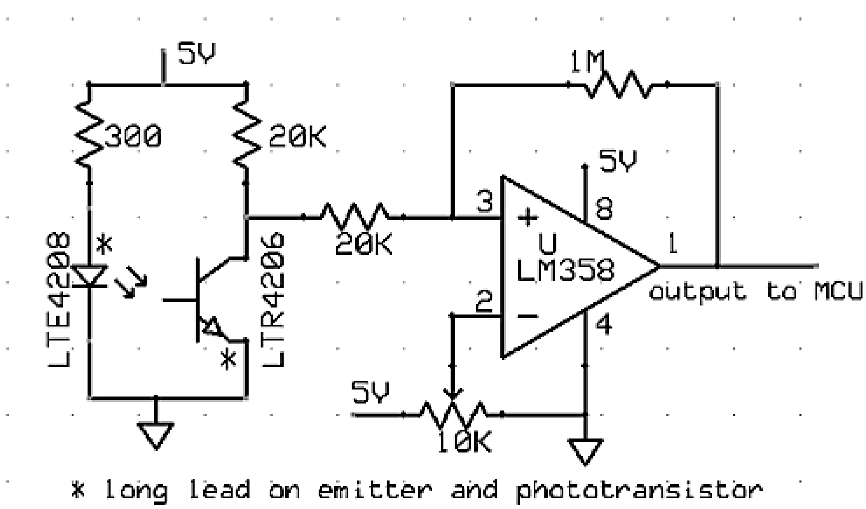

Figure 6. IR Transceiver Circuit Schematic [3]

The setup is mainly achieved modifying the EIMSK and EICRA register in the microcontroller. For our design purpose, we used PORTD0 and want to capture the rising edge. Thus, we programmed that EIMSK=1<<INT0; EICRA=3;

3.1.2 Analog Digital Converter

The Atmega 1284p has built in ADC that has 10-bit resolution, 200kHz maximum saping rate and noise canceler which makes it fits the design requirement. The first conversion takes about 25 cycles and the following will take 13 cycles. Thus, in code design, we have to consider this in timing requirement. The result of ADC is that ADC = Vin×1024/Vref. The 10 bit information will be stored in the ADCH and ADCL registers and the storing order is determined by the ADLAR. The trick part of ADC design is that you have to read the ADC registers and in order so that the ADC is not locked. We initialize the ADC by setting ADCSRA|=(1<<ADSC);, while(ADCSRA&(1<<ADSC)); will wait the conversion to finish and we read the ADCH to any variable into our code so that the ADC will perform the next sampling . [4]

In our design, we used 2.56VDC internal voltage reference as the maximum scale. We have verified that all our ADC related measurement will falls between the 0-2.56 VDC operation ranges. Thus, we put ADMUX = (1<<ADLAR)|(1<<REFS0)|(1<<REFS1)|; with associated ADC port number. The port number is set by the MUX numbers that can be found in the Atmega 1284 user manual.

3.1.3 Pulse Width Modulation

The microcontroller also has the ability of outputting pulse width modulation waveform. To achieve various voltage levels, the easiest way is to use PWM and low pass filter. The PWM can output square wave with various duty cycle by changing the compare and match register, which is OCR0A and OCR0B. Configurations are done by setting the TCCR0A, TCCR0B and TIMSK0 registers that sets the outputting clock frequency, overflow management and compare and match mode (whether clear or not). In addition, the direction of PORTB3 and PORTB4 are set to output as it is predefined as potential PWM output by the Atmel microcontroller.

One PWM output is connected to the motor directly as the vibration motor it has low pass functionality and various vibration levels can be achieved by different OCR0A values. On the other hand, the heating provided by the IR emitter is low passed by the heat capacity of the skin.

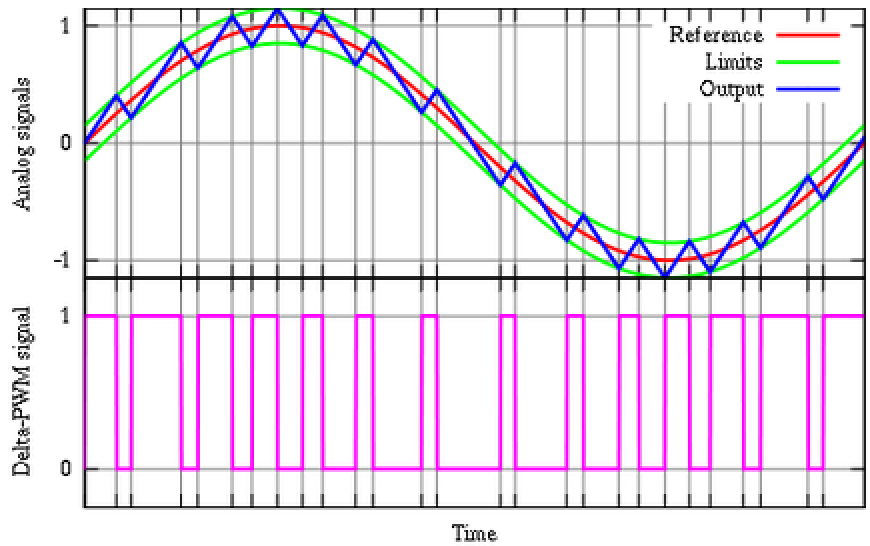

Figure 7. PWM Illustration

Code sample is shown as the following.

DDRB|=(1<<PINB4);DDRB|=(1<<PINB3);

TCCR0B=1; //clk by 1024 TIMSK0=(1<<TOIE0);

TCCR0A=(1<<COM0A0)|(1<<COM0A1)|(1<<COM0B0)|(1<<COM0B1)|(1<<WGM00)|(1<<WGM01); OCR0A=128;OCR0B=128;

3.1.4 Communication Design

In this design, for debugging purpose, we used serial communication between the microcontroller and the computer. On the computer side, we setup PuTTy to read the correct USB communication port from microcontroller. On the other hand, we have also initialized the microcontroller to be able to send serial information to the computer. We take the advantage of trtUart code sample from ECE 4760 website to establish such a communication protocol. And thanks to Bruce R. Land's hard work, the implementation using both the hardware and software encounters very little difficulty. From the UART code library, we simply need trt_uart_init(); stdout = stdin = stderr = &uart_str; to initialize the communication. Then, we can use the fprintf build-in fuction freely to output any message we want to the computer. Such as: FILE uart_str = FDEV_SETUP_STREAM(uart_putchar, uart_getchar, _FDEV_SETUP_RW);

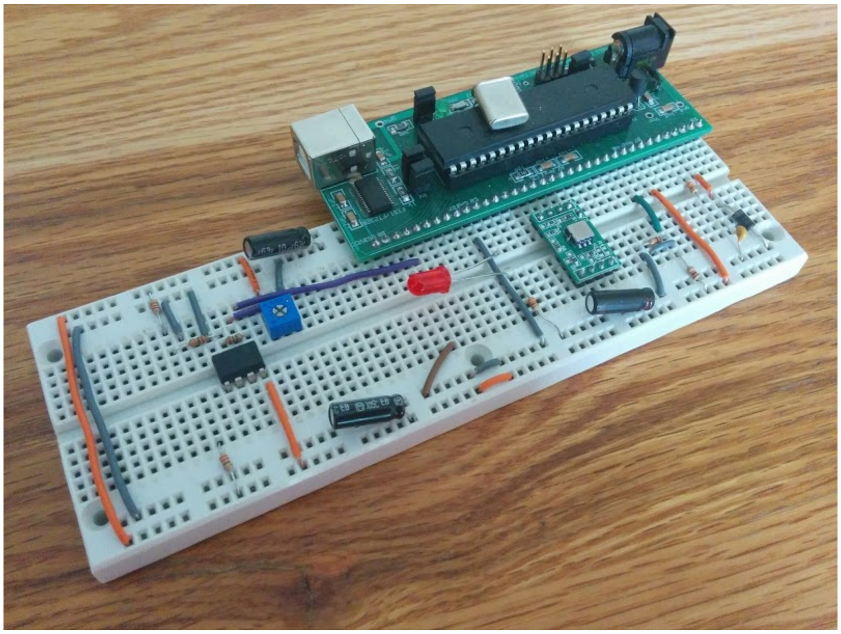

Figure 8. Overall Circuit Setup

3.2 Software Design

In this section, the programming for the microcontroller will be discussed. The programming for this controller can be broken down to three major concerns. The first one is the interrupt service routine, which explains how microcontroller responds to the interrupts. And the other one is the tiny real time kernel, which primarily discusses how those different tasks are running in parallel. In the end, we will have a brief discussion about current logic.

3.2.1 ISR Code Design

When interrupt happens for Atmel microcontroller, it will try to find the corresponding routine to execute the pre-defined tasks based on the interrupt vector table. The WINAVR library and compiler allow us to use the ISR library to program those tasks. Otherwise, if there is no such a routine, the microcontroller will hang at that interrupt and never exits. The whole project design will stuck.

After optimization and care examination for all functionalities, we finally decide to use two interrupts only for the APM count functionality. ISR (TIMER0_OVF_vect) and ISR (INT0_vect) are used. The first interrupt handles timer overflow interrupt so that we could increase our count scope to a larger time scale. We also set the OCR0A and OCR0B values in this interrupt routine as it is a safer that guarantees the successful of real time value assignment due to the highest priority of interrupt. On the other hand, the second interrupt will be trigger by any rising edge of PORTD0, which means we have detect one or more finger clicks. Thus, at this routine, we simply keep incrementing our APM counter and reset to 0 when necessary.

In addition, we have also put all variables involved in the interrupt as volatile so that the values are always read correctly because it will load value from memory instead of cache. That is because under some circumstance the value of some variable will be changed without acknowledging the CPU. There are also possibilities that the compiler is not aware of the interrupt service routine that it will try optimization to that variable so that the routine lose track of the correct value.

3.2.2 TRT Code Design

The tiny real time kernel is a multitasking kernel that allows several tasks to run in parallel. Although in most cases, writing small tasks and perform sequential running will be acceptable for a small microcontroller project, we desire all sensing information to be reported in real time. Thus, choosing an available kernel can be the best choice.

Tiny Real time was written by Dan Henriksson and Anton Cervin. The kernel is taught by Bruce R. Land in ECE 4760 lectures. Thanks to Bruce's brilliant teaching, we are able to use this kernel fluently.

To use the kernel, we need to create semaphores to protect shared variable and corresponding tasks with start and end time requirement. For example, trtCreateTask(get_ADC_info, 200, SECONDS2TICKS(0.1), SECONDS2TICKS(0.2),

&(args[1])); will create a ADC measurement task with 0.1 second timing restriction. And for that specific task, which is void get_ADC_info(void* args), we used a while(1) loop to make this program keep running. trtSleepUntil(trtCurrentTime()+SECONDS2 TICKS(0.2), trtCurrentTime()+SECONDS2TICKS(0.3)); will wake up the trt task after each time it finishes. And we program the content as described in the hardware design sections. We created several different tasks with similar timing restriction and initialize them at the same starting time. In this method, we achieve the real time monitoring for all sensing and treatment related functions. Thus, for the "main" of the program, we simply need a while(1){} and there is no need for any content.

Furthermore, to save power, we initially determine to sleep the CPU when not needed. However, due to the timing concern mentioned in section 3.1.2, the ADC will not perform correctly if we use the set_sleep_mode(SLEEP_MODE_IDLE); sleep_enable(); code provided by the tiny real time kernel. More investigation on this could be a possible future extension design.

3.2.3 Control Logic Design

After getting all the sensing information, we need to change the OCR0A and OCR0B to change the treatment value. Currently, it is achieved by a weighted function that considers all sensor inputs and tries to calculate a feasible treatment output value. The recent version uses: Treat_Value=(int)(1.35*(float)(pressure)+0.25*(float)(APM)+0.35 *(float)(acceleration)+19.3); with corresponding if-else statements to prevent value to overflow or underflow. (explain with data)

To be more specific, we first determine the value of input range then determine its coefficient. Since the most important feedback, we start with Treat_Value=1.45*pressure as the pressure value varies from 130 to 160 after certain conversion but our desire treatment value varies from 190 to 235. Thus we applied this ratio directly. After taking the pressure feedback into consider, we lowered its weighting factor because there are more aspects than represents gamer's current excitement. They are the acceleration of hand and APM value. The result section will show the measurement in details. Our initiative is to treat them equally important to reflect hand's motion. Thus, they have similar but less than pressure's weighting factor. In the end, with combination of all previously mentioned factors, we have achieved a result shown as below. All test cases done in real world scenario and input level differences are shown in appendix.

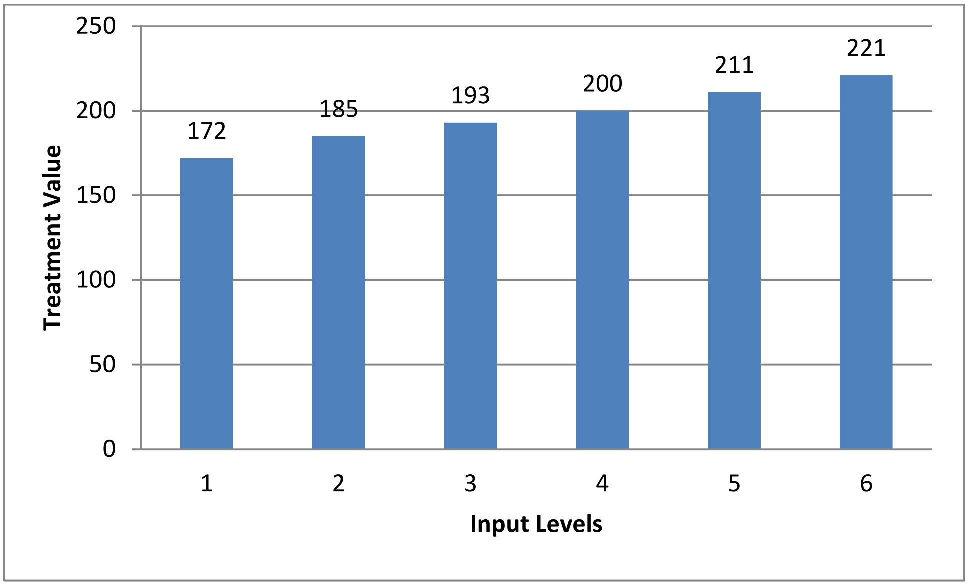

Figure 9. Treatment Value vs Input Conditions

State machine is not used for this prototype design as the only current available mode is the curing mode and it performs as expected to be. For future design, error mode, debugging mode, idle mode that can be used to save power and run self-calibration can also be implemented.

4 Experimental Results

In this section, various types of experiments are done to verify the functionality of this relaxation gaming wrap system design. There are three main concerns that need to be tested. The first is the APM test, which could verify if the interrupt method is feasible or not. Secondly, the PWM generation should also be tested to figure whether the microcontroller ports output sufficient power. In the end, we need to verify whether the ADC value conversion is successful or not. Other peripheral tests will also be mentioned in the rest of this section.

4.1 APM Measurement

The APM measurement is done through moving fingers and monitors the PuTTy feedback that counts the hand’s motion for every 5 seconds. In this test, we let one person to equip with the APM measuring tools and the other person runs a java based open source online APM collection software and compare our result to the computer PuTTy message. The PuTTy message figure will be shown in the ADC section.

Figure 10. APM Measurement Setup

The result for APM test is that we found out the interrupt usually creates 1.15 to 1.4 times more than software count. The figure of measurement results does not show a perfect linear relationship between estimated and real APM value. Thus, we need to down scale the measured value to an acceptable range. The cause for this issue is that multiple rising edges could be associated with single finger click. However, as this value is not crucial to treatment, such a scaling method is a feasible solution to obtain desired APM value.

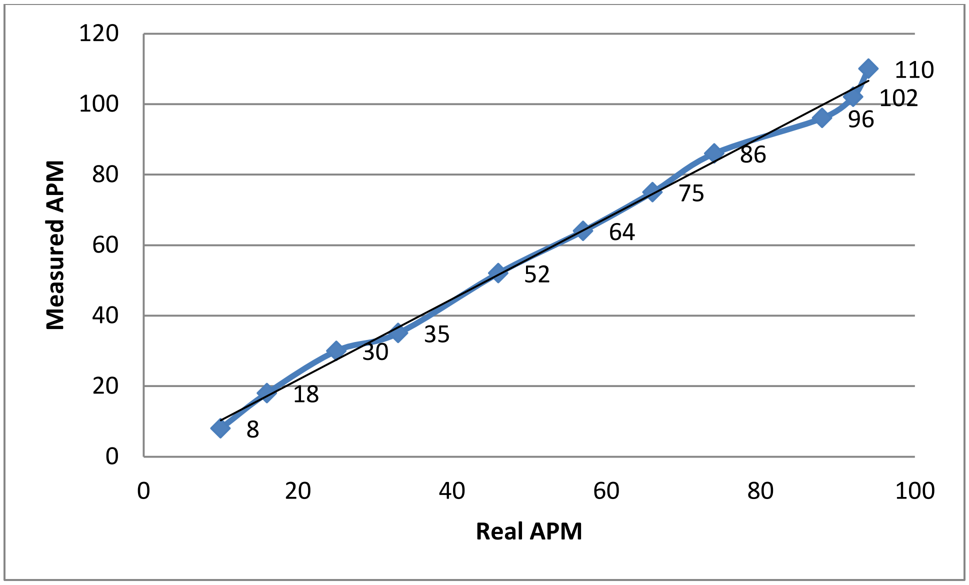

Figure 11. Comparison between Real APM and Measured APM

4.2 Treatment System Measurement

The treatment is mainly build by using the PWM output from microcontroller and we need to verify the driving capability from the Atmega 1284p because each port could only output 20mA × 5VDC = 100mW power. However, both the vibration motor and the IR emitter have higher nominal power ratings.

To begin with, we tested the vibration motor by various pre-set OCR0A values and observe the performance. For this test, we took the idea of direct digital synthesis from ECE 4760 lab works. [5] While the compare and match register is changing, the output pulse changes its duty cycle and finally to achieve control signal amplitude change. The values start from 50 to 255, which covers mostly the range of the PWM duty cycle. Our observation have found that it start generating output voltage around 100 and was too rapid at 255. Thus, the design team has determined 125 to 230 to be the feasible treating range for the vibration motor.

On the other hand, we have also tested the IR emitter. We found that only the maximum heating level produces observable heat to user's hand. Figure 12 will provide detailed test results by numbers. The temperature is tested by real life scenario that we put the wrap in between the temperature meter and the IR emitter and take the meter's readings. If the medical restriction requires more power output, we need to design new circuit to fit the IR heating purpose. Further research need to be done to determine the final heating solution.

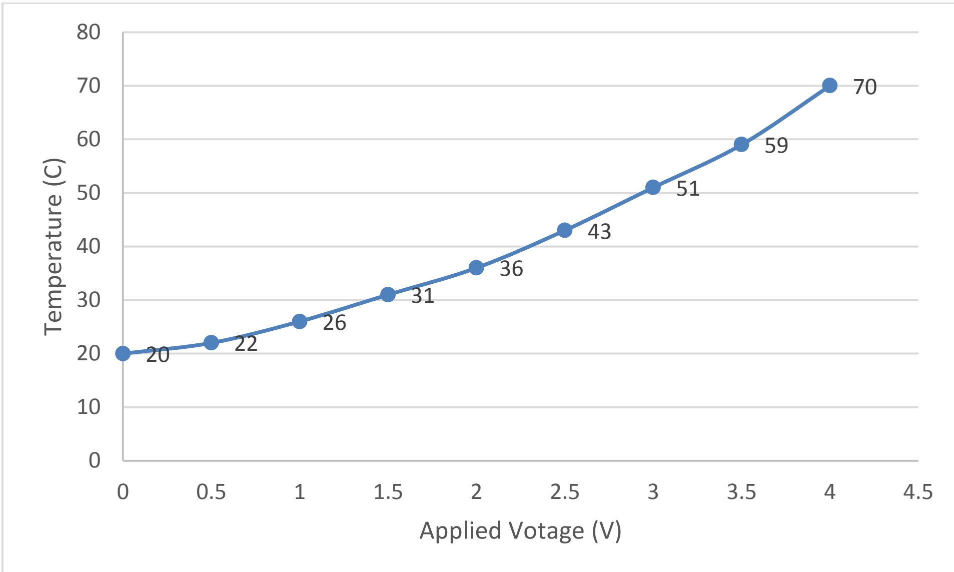

Figure 12. IR Emitter Temperature Curve

In addition, those high temperature cannot be driven by the Atmega 1284p pins as the power requirement is very high. Thus, there is no safety concern on temperature.

4.3 ADC Experiment

The ADC experiment will mainly include the test for analog accelerometer and pressure sensor, which provides variable voltage level. Accuracy will be shown with either comparing and contrast figures or tables. The result we collected mainly base on the input level we classify. The input levels are determined by real experience from gamers and we divide those level 6. This method is chosen because it is more considers the real world condition more on true human behaviour rather than fixed numbers.

4.3.1 Accelerometer Experiment

The accelerometer module does not require complicate hardware setup as it is already a development version by MEMSIC Inc. It provides VDD, GND, analog X, and analog Y pins directly. All we need is to test if it is a good fit for hand's acceleration.

We simply connect the power pins and feed X/Y to the ADC ports on microcontroller and also the oscilloscope. The analog X and Y direction output provides output range within 3.3VDC and 0VDC. Even though the maximum output exceeds our pre-set internal reference voltage, such a fast acceleration can hardly be achieved by gamers hands when using mouse as gaming input. We found the nominal output voltage range falls between 0.2VDC and 2.4VDC, which is a perfect fit for our ADC design.

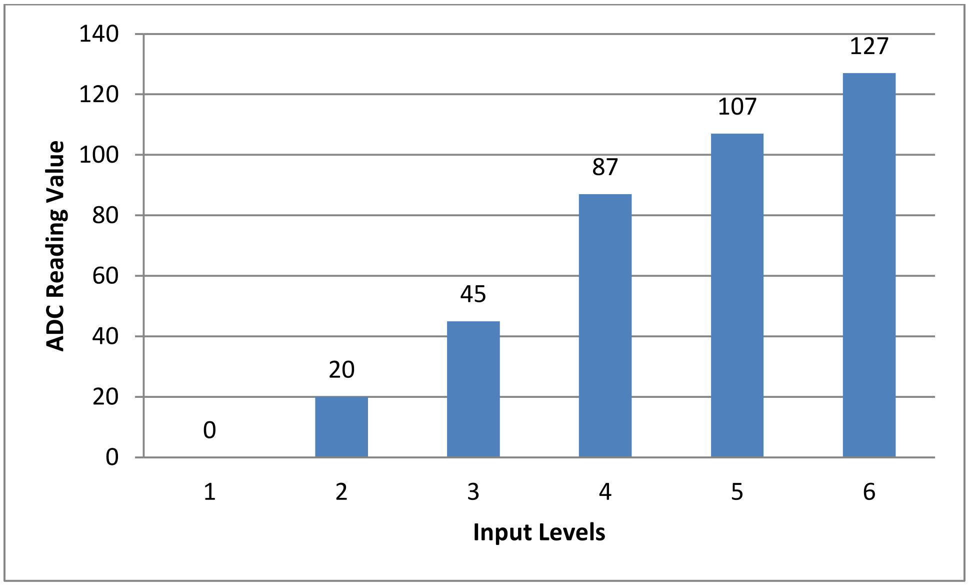

Figure 13. Acceleration Distribution on User's Input

4.3.2 Pressure Sensor Experiment

First of all, we ensure the pressure sensor's power consumption by adding large resistors in series and parallel. We begin testing the operating range of the pressure sensor first. The pressure sensor is basically a variable carpal tunnel resistor corresponds to different pressure input. With respect to this characteristic, we first measure the range of such a resistor to be 0 to 1000 Ohms. Then, we put it in parallel with a 1k resistor and then together in series with a 1k resistor. We feed the voltage across the touch resistor to the ADC. Thus, under 5VDC power supply, the output voltage to the microcontroller ADC will vary from 0VDC to 2.5VC due to the variable voltage divider we designed.

By various tests, we have verified that the ADC high bits register provides values from 10 to 255. And that meet our expectation for the ADC reading on the pressure sensor. The figure below shows the final PuTTy message. On that message, we could see that information about APM and ADC are successfully taken. And the treatment value is also correctly calculated by the microcontroller.

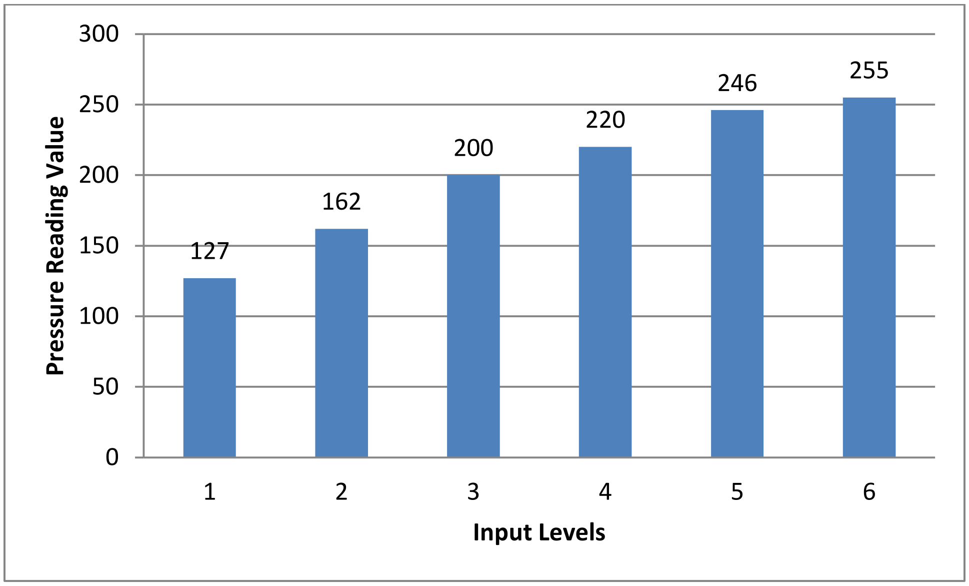

Figure 14. Pressure Distribution on User's Input

4.4 Overall Prototype Performance Test

After connection all parts together we start our integration tests to make sure the power consumption does not exceed the power limited mentioned in section 1, which is 2.5W. And we also want to find out the correctness of real time kernel to try if all sensing, control and treatment functions are running in real time. We mainly test the power consumption by measuring the output current and voltage at each port as it is far more significant that microcontroller power and USB communication power usage. We consume about 300mA current in total that falls within the desired range (motor and IR emitter drain at maximum current from each port). However, this limit is not critical as it is possible to double or triple the total power by connection more USB ports in parallel.

On the other hand, the code integration test encounters various bugs that not all parts are corresponding correctly. To fix those issues, we start adding components little by little and try to find the impact produced by each of them. In the end, all systems mentioned previously can run in parallel without interference from each other. Beside demo for ECE day and Bruce R. Land, we have also present our prototype to some gaming professionals and tested use by ourselves. Most people feel much relaxed after 4-5 hours of gaming with the designed relaxation gaming wrap. It supressed the phenomenon of red skin and wrist pain. However, in contrast, those people who involved this test used to suffer dead skin problem due to extensive gaming activities. The result shows that they all like this project idea and their hands feel much better after extensive gaming experience.

4.5 Test Result Discussion

First of all, by receiving mostly positive feedback from self-gaming test and other sample users that this design team has send, it can be shown that the prototype performs as expected to be. However, as the number of person that involved is very limited, there are still uncertainties whether this project design will fit for everybody. Thus, more tuning or design modifications may be needed when more sample test results can be collected.

On the other hand, as mentioned in previous sections, the treating IR emitter is only a tentative solution. More research and test should be done to figure out what is the optimal medical solution and electronic device for this project design. Indeed, some medical research has reported that improper usage of IR emitter could even hurt human's hand's health. And the size of this IR emitter is too big that will cause uncomfortable for long term use. Based on those factors, this IR transmitter should be replaced with other version in the future. And similar situation has also be reported with respect to vibration motors. To be truly convincing that this designed device could be beneficial to health, more rigorous tests on the treatment devices should be done.

5 Conclusion

With respect to analysis in previous sections, it can be clearly justified that the main purpose of this design is achieved. The design for health improvement system and the control system meet the requirement of the initial design and the experiment results are satisfactory. Through this design project, both Wei and Yuan gained practical engineering design experience and insight to electrical engineering. Moreover, potential upgrade for both software and hardware aspects are also discovered during design activities. The following sections will explain in details.

5.1 Design Conclusion

The overall goal of this design is to prove the idea for curing the gamer's hand is feasible and low cost. According to all the analysis on high-level block analysis, detailed software and hardware design, and those experimental results shown in previous sections, it can be found that this relaxation gaming wrap design is successful with respect to its sensing and treatment ability.

Moreover, it is proved that all three major blocks performs excellent by demo to Bruce R. Land that sensing, control and treatment systems are successfully designed. We have demonstrated that the pressure, hand's motions, and finger motions can be captured in real time and the microcontroller will respond to that information to make quick and accurate decision to the treatment system. And the treatment will provide satisfactory treating performance.

By self-gaming experience and feedback from gamer friends, we ensured that our project idea could truly benefit gamers' health. It is shown that the primary goal, which is to mitigate gamer's wrist and hand's stress without impact to gaming experience, has been achieved. We have also collected several suggestions and those will be discussed in future section.

5.2 Experience and Learning Discussion

Through this design project, both Wei and Yuan learned several useful engineering design concepts from experiments and suggestions from Bruce R. Land. And these knowledge and experience will benefit all members in this team during careers as electrical engineers. The main points of learning include what mistakes do engineers often make during engineering designs and what should be the engineering way to perform experiments.

First of all, the design team learned that a good engineer design should consider both user and engineer use. The initial design idea is too complex to be a feasible customer level product. That is because at the beginning of this design project, the team was only focusing on the engineering development version but totally ignoring the potential product that could be built from this design project. Thus, the first generations of this relaxation gaming wrap design had too many peripheral testing ports and plans for testing and making the design team unable to meet deadlines planed. To overcome this issue, the team had set down for a meeting deciding what are necessary and essential for testing and implementation and what should be the final product to truly push this product to the market.

On the other hand, the design team have also learned the individual tests for component guarantee nothing about integration test from Bruce R. Land. Before deploying all prototype components to the Atmega 1284p hardware, all those hardware performs as expected with ideal voltage and current sources. However, after the overall hardware setup the story of their functionality changes. We found various non-ideal phenomenon happened that prevents the prototype to be successfully built. Based on this lesson, the whole design team took advices from Bruce R. Land and start integrating few parts at a time and debugging from good design reference.

5.3 Discussion of Future Design and Implementation

During this design process, the design team have also discovered potential upgrade for this relaxation gaming wrap system design. The possible changes involve choice for microcontroller and building associated software to be more users friendly. To make this product perfect for market needs, those research and developments should be continued and those possible upgrades should be implemented even though the existing design is satisfactory.

First of all, the reasons why Atmega 1284p is used are that Atmel products are usually low cost, available anywhere and easy to program. In other words, it fits perfectly to small projects such as a single charging system and other student design projects as we rapid used in ECE 4760 course and have gained a lot of practical experience with Atmel products. However, the Atmel microcontrollers are generally not high power efficiency and do not have advanced chipset structures such as the Cortex-ARM series. Beside Atmel, better chips could also be found from manufactures such as NXP, STM, and Qualcomm and so on. Those manufactures provide better chipsets with faster processing speeds and lower power consumption. Thus, in future implementation, the microcontroller might be replaced with a better one if the cost function of the whole system design can be minimized.

In addition, the current design prototype is not user friendly enough. Since the USB communication is already setup on the hardware side, there is little difficulty to implement a user interface that allows users to store their personal data, and having easier access to the control of this gaming wrap. Moreover, servers can also be used to store user data online for better diagnostics purpose. If all factors mentioned previously that has the potential to be improved are achieved, the future of this product is truly promising.

References

[1] *DotA 2 News: Fear ruled out of EG due to an elbow injury, Mason to replace | GosuGamers*, <http://www.gosugamers.net/dota2/news/27233-fear-ruled-out-of-eg-due-to-an-elbow-injury-mason-to-replace> [accessed March 15, 2014]

[2] *Digi-Key*, <http://www.digikey.com/> [accessed May 10, 2012]

[3] *ece4760 Lab 4*, < http://people.ece.cornell.edu/land/courses/ece4760/labs/f2013/lab4.html> [accessed January 25, 2014]

[4] *Atmega 1284p Datasheet*, < http://www.atmel.com/images/doc8059.pdf> [accessed January 29, 2014]

[5] *ece476 Lab 2*, < http://people.ece.cornell.edu/land/courses/ece4760/labs/f2013/lab2.html> [accessed January 24, 2014]

Appendix

Appendix A – Code Samples

```
//-------------------------------------------------------//
//--------We love ECE 4760 and 5760------------//
//--------------------------------------------------------//
#define SEM_RX_ISR_SIGNAL 1
#define SEM_STRING_DONE 2 // user hit <enter>
#define SEM_PRESSURE 3
#define SEM_ACCELER 4 // user hit <enter>
#define SEM_APM 5
#define SEM_TREAT 6
#define SEM_MUTEX 7
#define SEM_ADC 8
#define F_CPU 16000000UL

#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <util/delay.h> // needed for lcd_lib
#include <inttypes.h>
#include <avr/eeprom.h>
#include <avr/sleep.h>
volatile int Treat_Value,Treat_IR, apm_period, apm_counter;
volatile long int apm_period_ovlf;
ISR (TIMER0_OVF_vect)
{
	OCR0A=(char)(Treat_Value);
	OCR0B=(char)(Treat_IR);
	apm_period_ovlf+=1;
	if(apm_period_ovlf==187500) {apm_period_ovlf=0;apm_counter=0;}
}
ISR (INT0_vect) // Get apm info
{
	apm_counter+=1;
   // apm_period=TCNT2 + apm_period_ovlf;
   // TCNT2=0;
   // apm_period_ovlf=0;
}
// ISR (TIMER2_OVF_vect) {
	// apm_period_ovlf = apm_period_ovlf + 256;
// }
```

```
#include "trtSettings.h"
#include "trtkernel_1284.c"
#include "trtUart.h"
#include "trtUart.c"
FILE uart_str = FDEV_SETUP_STREAM(uart_putchar, uart_getchar,
_FDEV_SETUP_RW);
int args[4] ;
#include "trtTasks.h"
#include "init.h"

int main(void)
{
	init();
	// set_sleep_mode(SLEEP_MODE_IDLE);
	// sleep_enable();
	while(1)
	{
		// sleep_cpu();
	}
}
```

Appendix B – Input Setup Ranges

Table 7. Input Setup Details

Level	APM	Pressure	Acceleration
0	0	127	0
1	30	129	5
2	55	140	35
3	60	195	65
4	79	231	97
5	94	255	127