UNIVERSITY OF FLORIDA

Project #13

Department of Electrical and Computer Engineering

EEL 4914C Senior Design

Summer 2007

Final report

Guitar Hero Game Controller

Submitted by:

Carlo Pascoe

&

Michael Yip

Table of Contents

Project Abstract

The objective of this project was to create a device that will use a standard electric guitar as a game controller for a game system. Playing notes on the electric guitar act as if the user was using the game controller to play the series of Guitar Hero games. After ten weeks of painstaking labor the device works as expected. The final product is a device that takes in an electric guitar signal and outputs multiple signals to the game console in order to play the game. The remainder of this report describes how this was done.

Introduction

In the game Guitar Hero, players play the game as if they were a guitar player. The game displays notes on screen for a certain type of rock song. The players then use a game controller to press certain buttons that corresponds to the notes on the screen, which will then play the song. To make it even more realistic to playing an electric guitar, the company has created a controller that is like a small electric guitar shown in Figure 1. Fret buttons on this controller are like holding the notes on an electric guitar and another strum button is used to simulate the player strumming the guitar. Even with this game controller, users want more from the game. They want to be able to play their own electric guitar. So far in the game industry there isn't a device that will use a real electric guitar to play the game. Having sold over 10 million copies of the three Guitar Hero games, this device will have a large market.

Figure 1 - Guitar Hero Game Controller

Technical Objectives

In Guitar Hero players must press specific combinations of buttons, corresponding to notes in a song, at specific times, corresponding to the time the note is played in a song. In order to make our controller a reality we needed to analyze the frequency and amplitude of the analog signal produced by an electric guitar and then produce a set of signals that interface with a PlayStation II console and the Guitar Hero series of games.

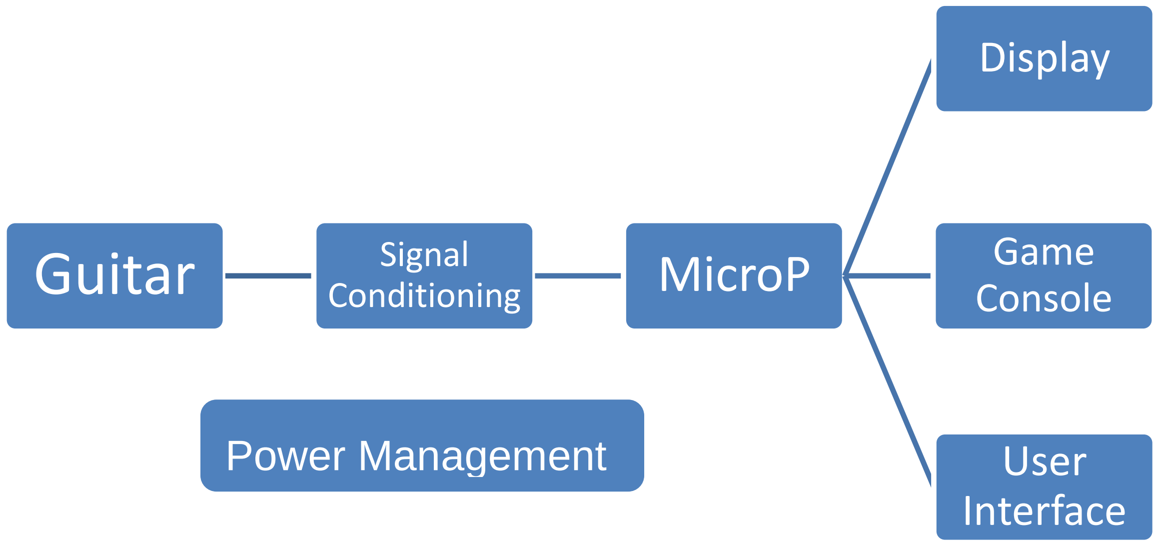
By analyzing the frequency we were able to determine the note played on the guitar. Since the analog signal produced by the guitar is on the order of 50mv to 100mv peak to peak and contains many resonant frequencies in addition to the frequency we want to detect we needed to amplify and filter the guitar signal before anything could be done with it. We used an op-amp with an approximate gain of 100 to amplify the signal. Since the frequencies we are interested in are between 50Hz and 1000Hz we used a low pass filter with a corner frequency at approximately 1 kHz to filter the signal. The frequency is determined by using an interrupt driven pulse accumulator feature on the PIC processor to count the number of high transitions in a specified amount of time (100ms). The TTL signal input of the pulse accumulator is produced with a MOSFET switch circuit and the filtered and amplified guitar signal.

By analyzing the relative amplitude of the guitar signal we were able to determine when a note is played. The output of a second amplifier circuit is connected to a peak rectifier circuit that will transition from less than 1V to some voltage above 3v when a note is played. The rectifier output is fed into a comparator that compares the rectifier voltage to a variable voltage that is controlled with a potentiometer. By detecting the transition of the comparator with a PIC we are able to determine when a note is played. The potentiometer allows us to adjust the sensitivity.

When a note is played, the determined frequency is compared to a table of values specific to the song that is currently being played and if it is within a range of values that corresponds to one of the five fret buttons of a standard Guitar Hero controller one of five output pins on the PIC is asserted. This button information is transmitted serially to the PlayStation II console. Since we do not know all of the codes that must be transmitted to the PlayStation in order for it to work and we had no way of determining all of the codes short of spending weeks analyzing the signals we decided to interface with a standard controller and use its serial transmitter, which already has all of the codes programmed, to interface with the console.

Concept Generation and Selection Matrices

Customer needs: **Reliable** **Low Cost**



Reference	T1	G1	F1	C1	MP1	U2	D1	P2
A	T1	G3	F1	C1	MP2	U3	D3	P2
B	T1	G1	F1	C1	MP1	U1,U2,U3	D1,D2	P2
C	T1	G2	F2	C2	MP3	U1	D3	P1
D	T1	G1	F2	C1	MP1	U2	D1,D2	P1,P2

Concepts	Transd.	Gain	Filter	Configuration	Micro P	User I	Display	Power Manag.
1	Guitar	Op Amp	RC	Single Ended	PIC	Button	LCD	Battery
2		BJT	Active	Differential	ATMEL	LED	LED	Power Supply
3		CMOS			TI	Pot.	Monitor	

Table 1- Concept Selection Matrix

In the Concept Generation Matrix, the customer needs for our device are reliability and price. The input transducer will be an electric guitar. The signal from the guitar will need to go through gain and filtering. There are several different approaches for the signal conditioning. There are many possibilities for the microprocessor stage, from a simple PIC processor to an advance TI microprocessor. From the processor, there will be a LCD display for the user and also a user interface. The output of the processor will go directly to communicating with the game console. In Table 2, several combinations of these options are chosen as a possibility for creating this device.

With these options chosen from the Concept Generation Matrix in Table 2, the Concept Selection Matrix in Table 3 is used to decide which concept is a possible solution. Our four concepts are compared to a reference concept, and then ranked. From the four concepts, Concept B seems to be the option that will work for our device.

Concept Screening Matrix						
Concepts A-F; compare with the reference concept . "+" implies better than reference, "0" equals reference, "-" less than reference.						
		Concept	**Concept**	**Concept**	**Concept**	**Reference Concept**
		A	**B**	**C**	**D**	
Customer/Market Needs	**Importance Rating (high=3, medium =2, low=1)**					
Reliable	3	0	0	-	+	0
Low Cost	2	-	+	-	-	0
						0
						0
Sum +'s		0	1	0	1	0
Sum 0's		1	0	0	0	0
Sum -'s		1	0	2	1	0
Net Score(+=+1,0=0,-=-1)		-1	1	-2	0	0
Rank		3	1	4	2	
Continue?			Yes			

Table 2- Concept Screening Matrix

System Level Presentation

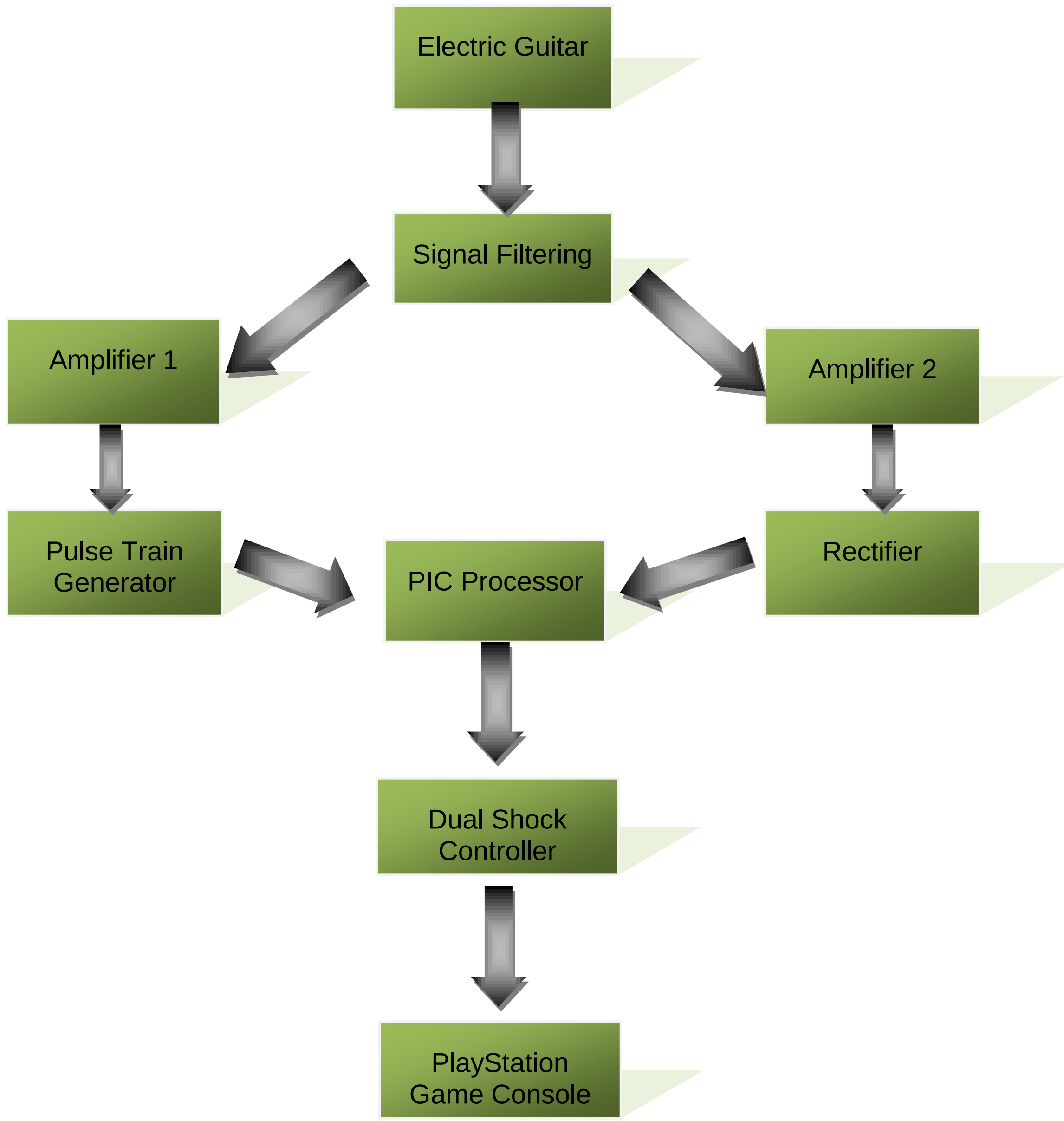


Figure 2 –System Level Design

Component Specifications:

- **Electric Guitar:** Brand: Squier, Body: Agathis. Neck: Maple, Machine Heads: Sealed die-cast, Frets: 22 Medium Jumbo, Bridge: Standard tremolo
- **Signal Filtering:** A simple RC Low-Pass filter places after the Guitar Input with a corner frequency at 90 Hertz.
- **Amplifier 1:** LM358 Operational Amplifier with a gain of 300 V/V which goes into a 2N7000 Power MOSFET to produce a TTL signals for the microprocessor.
- **Amplifier 2:** LM358 Operational Amplifier with a gain of 300 V/V which goes into a peak voltage rectifier. This signal then goes into a LM393 voltage comparator to compare the voltage to a potentiometer. The output of the comparator goes to the microprocessor.
- **PIC Processor:** An 18-pin Microchip PIC18F1320 8-bit microprocessor with optional A/D, ECCP module, USART communication, and timing modules. 8k bytes Flash, 256 bytes SRAM, and 256 bytes of EEPROM. Operating clock speed up to 8 MHz.
- **Standard Game Controller:** Analog controller compatible with PlayStation. Supports digital control, analog control, vibration function and pressure sensitive function.
- **PlayStation Game Console:** A computer entertainment system with DVD and CD playback. Digital Surround Sound and 2 memory card slots.

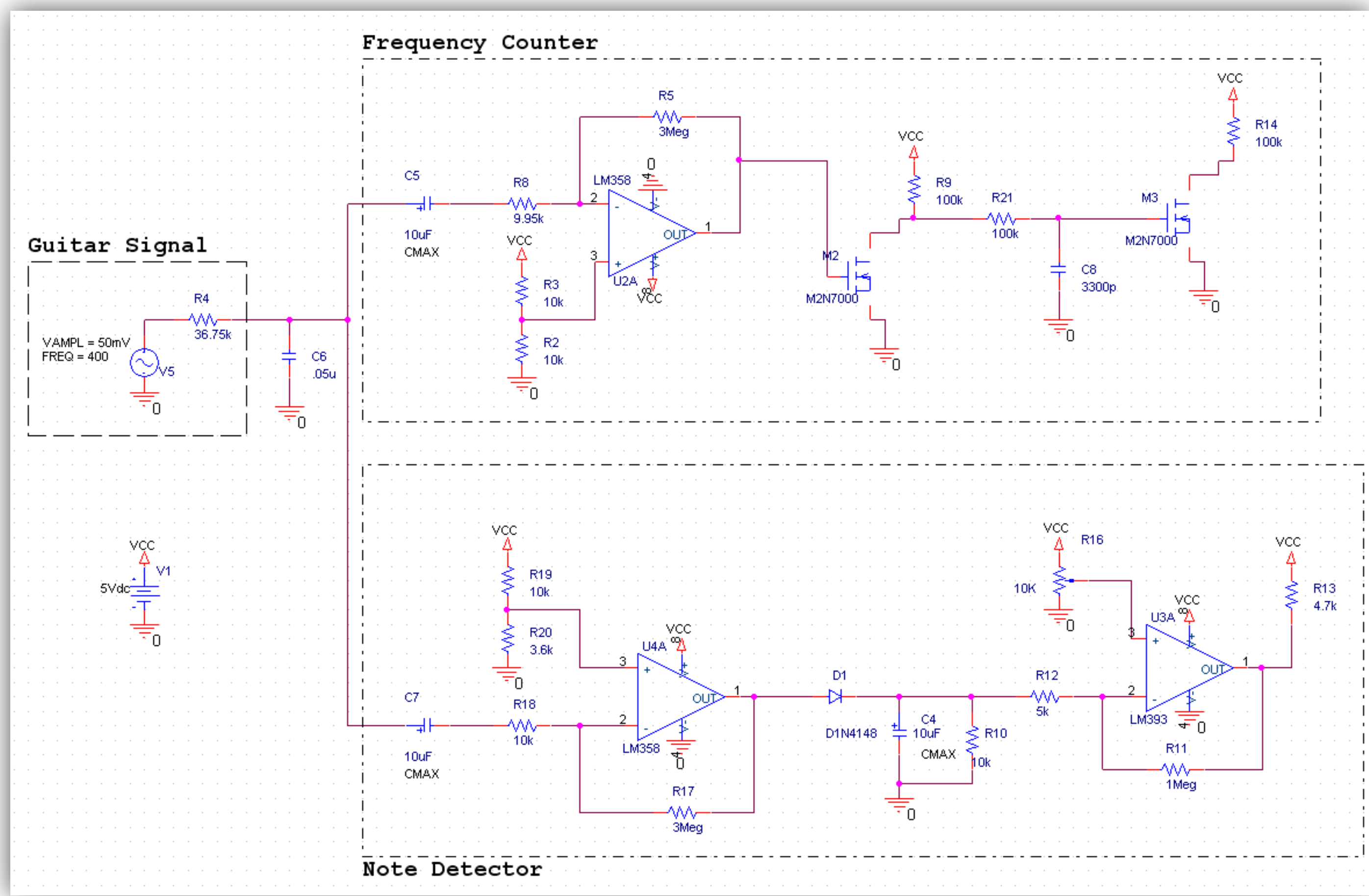


Figure 3 – Circuit Schematic for the Frequency Counter and Note Detector

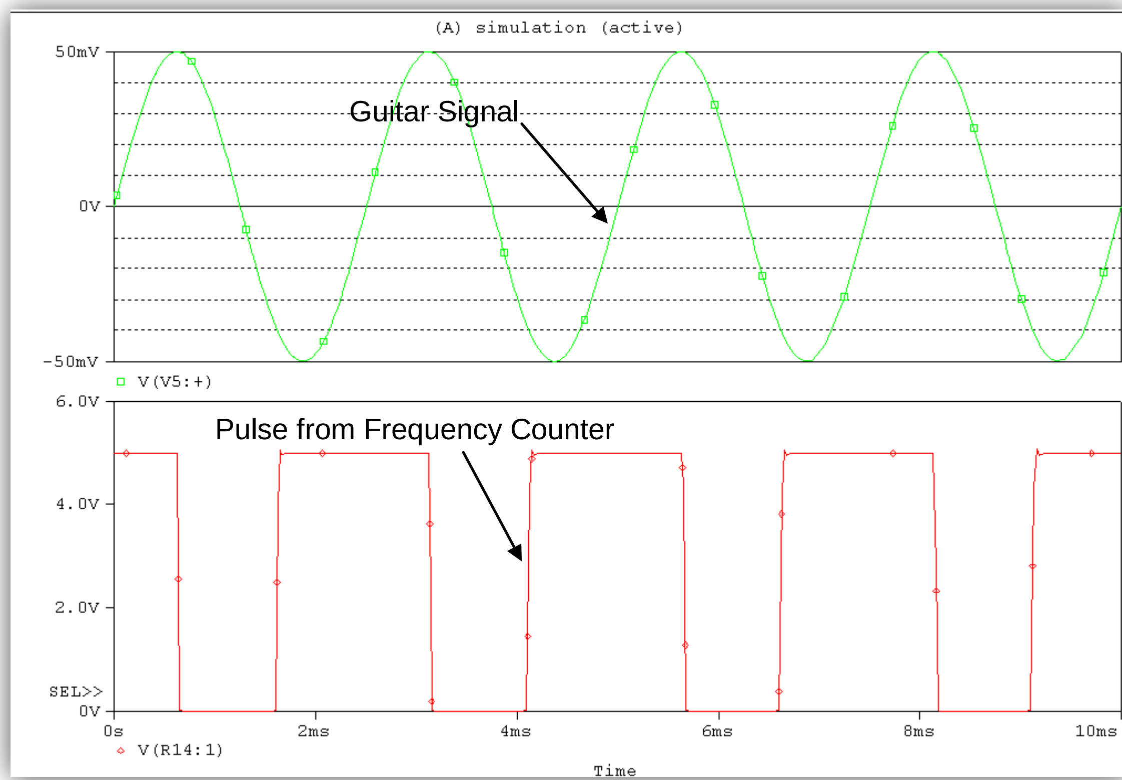


Figure 4 – Simulation of the Frequency Counter

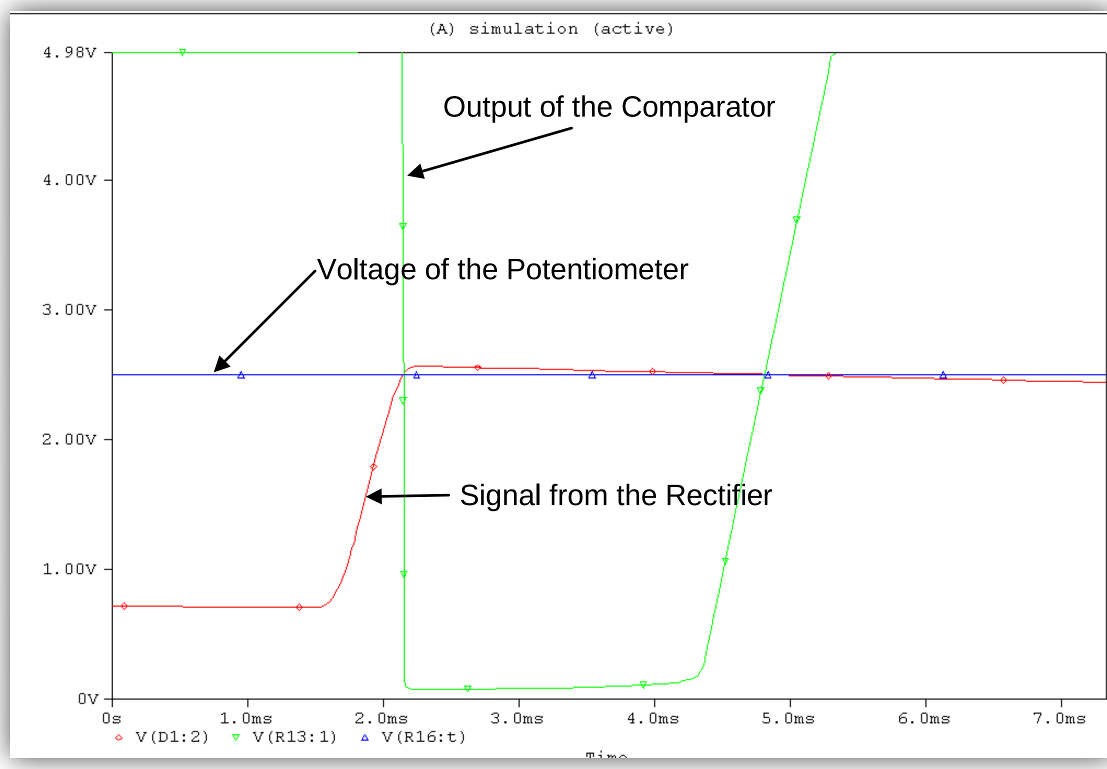


Figure 5 - Simulation of the Note Detector

Testing of Components:

- The electric guitar's signal was analyzed by the oscilloscope, and the input resistance of the guitar was found with a multi-meter.
- The amplifiers were built on a breadboard and tested with a function generator and oscilloscope for the correct gain.
- The microprocessor was tested with sample programs provided.
- The game controller's signal was also tested with the breadboard. Power MOSFETs were used to create a switch that could be triggered by the PIC microprocessor.

Final Product

The final outcome of our product is a device that can taking in a standard electric guitar signal and identify it in order to play the PlayStation game Guitar Hero. As shown in Figure 2, the system level design gives a look at the steps needed to make this device. In Figure 3, the circuit schematic gives a detail view of the two amplifiers. The final device uses the signal from the frequency detector circuit, shown in Figure 4, to identify the frequency with a pulse accumulator in the PIC processor. And to determine when to start counting these pulses, detector circuit was created, as shown in the simulation in Figure 5. The final product is able to detect when and what note the electric guitar is playing and then activate the appropriate button for the game.

The device is able to hit all of the buttons in the game with the current frequency programmed into the PIC processor. On the other hand, there are several ways to make this product better. First, a built in tuner can be made in the device that will help the user tune the electric guitar in order to play the notes for the game more accurately. Next, the program in the processor can be improved. A better sampling algorithm can help with the harmonics and any other problems. In the hardware side, a better type of filter for the signal will greatly help with the harmonics from the electric guitar. Another issue in making the device work better will be from the user side. Since both of us did not know how to play the guitar in the beginning, some of the playing problems can be contributed by us. Although the current performance of the device can play the game well, many ways can help improve upon this current device.

Business Case

Prototype Development Cost	Rate/unit	Qty	Cost
PlayStation II controller	14.95	1	14.95
LCD screen (2 lines)	0	1	0
5V Power supply	0	1	0
power connector	0.1	1	0.1
PIC18F1320-I/P	0	1	0
Enclosure	5	1	5
Potentiometer	0	2	0
2N7000	0	7	0
Resistor	0	24	0
Capacitor	0	7	0
LED	0	6	0
LM358N OpAmp	0	2	0
Diode	0	1	0
LM393N Comparator	0	1	0
Guitar Jack	1.5	1	1.5
PCB Board	0	1	0
SPST Push Button	0	1	0
Headers	0	34	0
Wires	0	34	0
wire connectors	0	34	0
18 Pin Socket	0.1	1	0.1
8 Pin Socket	0.1	2	0.2
Total			**21.85**

Table 3 – Prototype Development Cost

The total cost of our prototype is itemized in Table 3. Since we were able to procure most of the components from the lab the cost to us for our prototype was a relatively low $21.85. This cost dramatically increases when total parts, labor, and overhead are factored in. Table 4 below itemizes the estimated manufacturing costs for our Guitar hero controller. Rather than pay someone an hourly wage were our profit will be determined by how fast or slow that person works, we have decided that it would be a good idea to pay by the completed unit; paying $20 per unit incentivizes speedy completion and ensures that our profit margin per unit remains constant. Though there is no way to know for sure until we actually set up an operation, we estimate our overhead costs per unit to be $24. This $44 in addition to the $53.17 for parts brings the total cost per unit to $97.17. Since almost 80% of the $53.17 ($41.22) is from a single supplier there is a good chance that a more equitable deal could be brokered for a bulk sale thereby reducing the cost per unit.

Manufacturing Cost per Unit	Rate/unit	Qty	Cost
Labor (commission)	20 (per completed unit)	1	20
Insurance (workman's comp, etc)-20%	4	1	4
Overhead (equipment,expendables,etc)-100%	20 (per completed unit)	1	20
PlayStationII Duel shock controller	11.95 (for 100+)	1	11.95
LCD screen (2 lines)	5	1	5
5V Power supply	5	1	5
power connector	0.1	1	0.1
PIC18F1320-I/P	2.56 (for 100+)	1	2.56
Enclosure	5	1	5
Potentiometer	0.1	2	0.2
2N7000	0.07	7	0.49
Resistor	0.01	24	0.24
Capacitor	0.05	7	0.35
LED	0.1	6	0.6
LM358N OpAmp	0.1 (for 100+)	2	0.2
Diode	0.05	1	0.05
LM393N Comparator	0.17 (for 100+)	1	0.17
Guitar Jack	1	1	1
PCB Board	1793 (for 100)	1	17.93
SPST Push Button	0.15	1	0.15
Headers	1.90 (for 10 40Pin headers)	34	0.19
Wires (Multicolor Ribbon Cable - 34 core)	1.95 (for 1m)	34	1.33
wire connectors	.50 (for 40)	34	0.5
18 Pin Socket	0.08	1	0.08
8 Pin Socket	0.04	2	0.08
CD with instructions and Tabs for songs	free after rebate	0	0
Total			**97.17**

Table 4 – Manufacturing Cost

If we were to actually bring our product to market, we anticipate our known startup costs to be about $13,100 (these costs are itemized in Table 5. This total does not include unknown startup costs which include but are not limited to lawyers fees associated with creating a corporation or LLC and resolving any legal disputes with Sony, RedOctane, and record companies who own the right to the songs we are programming into our device. These costs can easily mach or far exceed the known startup costs listed in Table 5, so determining these costs will defiantly be a priority before moving forward with our business plan.

Start-Up Costs	
lawyers fees (Sony, record companies, RedOctane, etc…)	?
Tools (soldering -RoHS lead free compliant,clamps,cables)	1000
Programmers	100
NIH/FDA/ISO test and approval (model)	10000
Computer/Oscilloscope/Powersupplies	2000
Total	**13100**

Table 5 – Start-Up Cost

General Information	
Product life (years)	5
Production first year	1000
Production subsequent years	300
Sale price(excl. shipping)	160
Advertising per year	1000
Cost of Capital/discount rate	0.1

Table 6 – General Information

Analysis	**Year 0**	**Year 1**	**Year2**	**Year 3**	**Year 4**	**Year 5**
Sales	Initial Investment	$ 160,000	$ 48,000	$ 48,000	$ 48,000	$ 48,000
Production cost	Start-up	$ 98,170	$ 30,151	$ 30,151	$ 30,151	$ 30,151
Cash flow (sales-production cost)	$ (13,100.00)	$ 61,830	$ 17,849	$ 17,849	$ 17,849	$ 17,849
Net Present Value		**$ 85,949.53**				
Internal Rate of Return		**406%**				

Table 7 – Projected Return

Tables 6 and 7 show our business plan with a 5 year product life cycle. This plan assumes that the biggest demand will be in the first year of production with minimal demand in the subsequent years. We assume that by the 5th year the game technology will be obsolete either leading to the end of production or creation of a new generation product. Crunching the numbers estimates the net present value of our initial investment to be $85,949.53 and an internal rate of return of 406%. These numbers would justify the initial investment but they do not include the unknown startup costs or allow for the possibility that we have greatly underestimated demand. With over 10 million copies of the various Guitar Hero games sold there is defiantly a lot more potential customers than we have anticipated will purchase our product. If we were to actually bring our product to market we would sell preorders on our website to better gauge demand. If our preorders exceed the 1000 units anticipated in the first year then we will know we have greatly underestimated the demand and rush the production of more units. More units mean more revenue and therefore a greater rate of return. Another thing that will more than likely increase the number of units sold is to produce a unit that will interface with an Xbox game system rather than a PlayStation. The difference in production cost would be the difference between interfacing with an Xbox controller rather than a PlayStation controller and would be minimal. By selling both of these units at the same time we potentially double the number of units sold. Another avenue not discussed would be to sell our intellectual property to RedOctane for a hefty sum and let them control the production.

Study Organization

There haven't been any changes between our final and original Gantt chart. We have been able to meet the most of our deadlines listed in the chart. Deadlines that were not met were usually made up during the following week or the extended time allot in the chart.

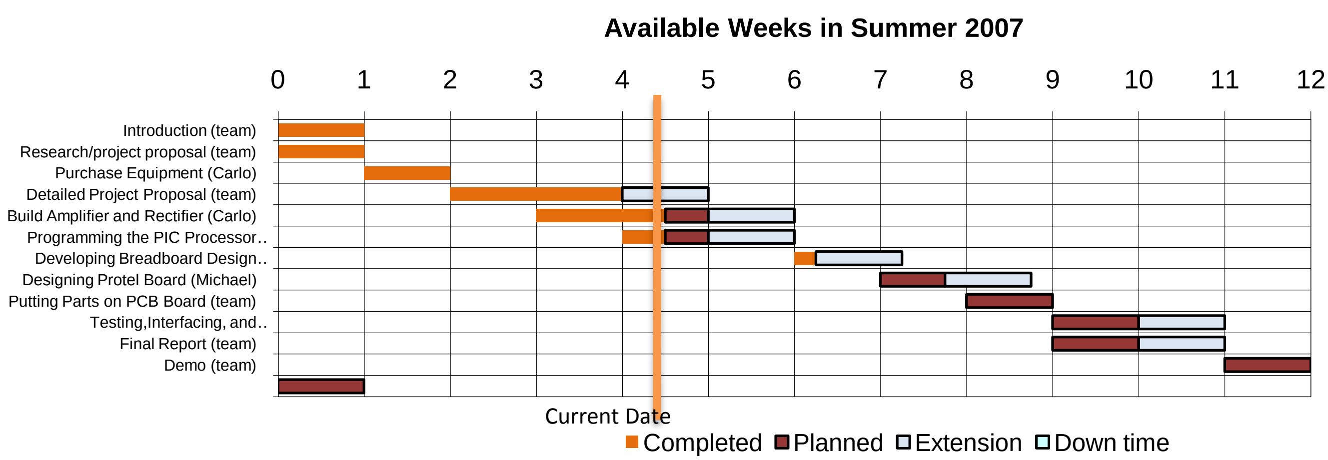


Figure 6 – Original Gantt Chart

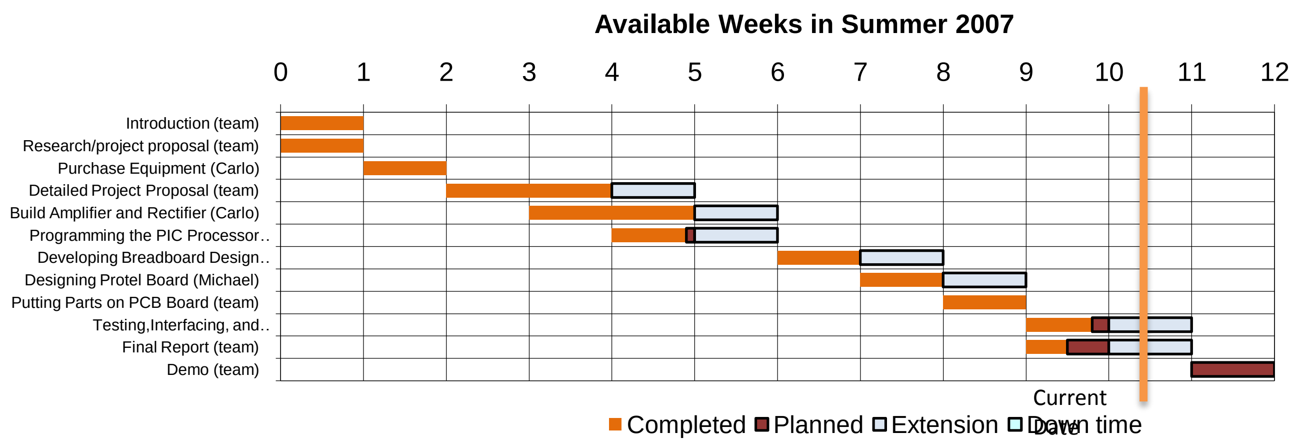
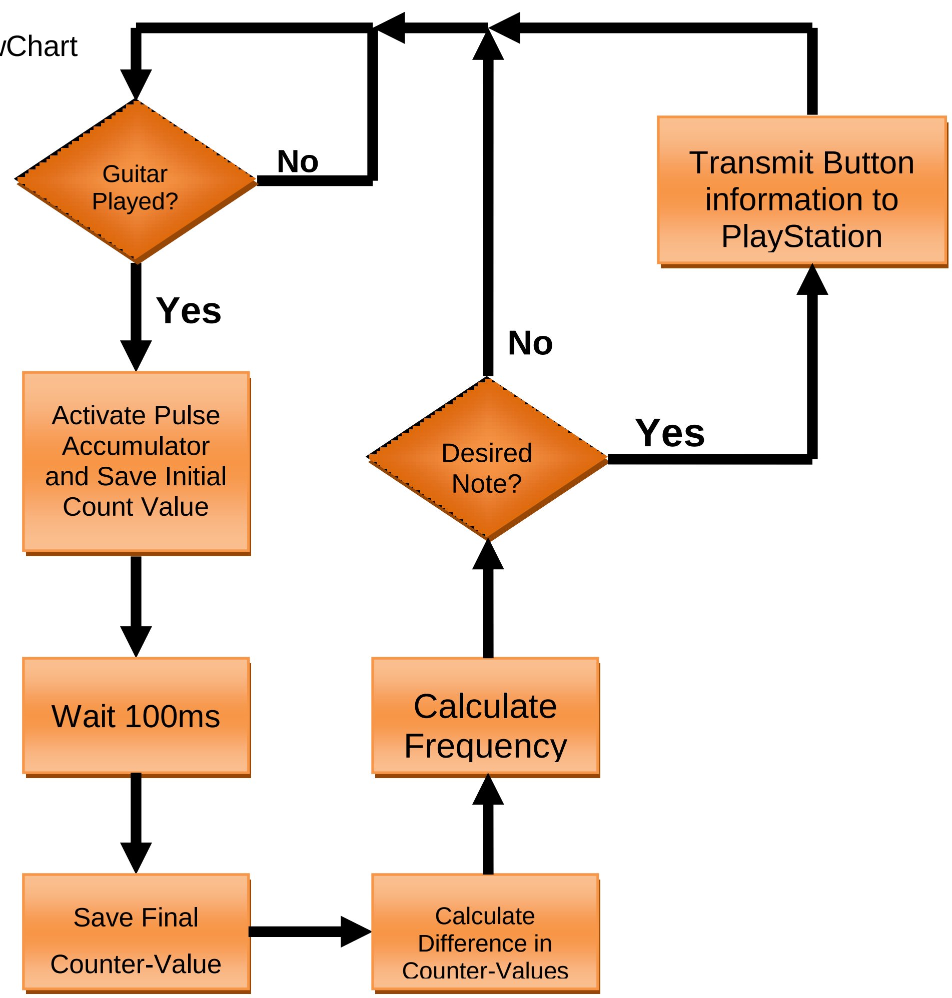


Figure 7 – Current Gantt Chart

References or Bibliography

APPENDIX

Source Code FlowChart



Source Code

```
DEFINE LCD_EBIT 4 'LCD enable bit
    OSCCON = %01100010 ' Set up internal oscillator

    ' Timer 0 Setup
    'T0CON.T08BIT = 0    ' 16-bit Timer
    'T0CON.T0PS2 = 0     ' PreScaler 1:2
    'T0CON.T0PS1 = 0
    'T0CON.T0PS0 = 0
     ADCON1 = ADCON1 | %00010000

    ' Counter Setup
    'T1CON.TMR1CS = 1    'External clock from pin RB6/PGC/T1OSO/T13CKI/P1C/KBI2
    'T1CON.TMR1ON = 1    'Enable Timer 1
    'T1CON.RD16 = 1      'Enable 16-bit read/write
    T1CON = %00000011

    CounterValueStart VAR WORD  ' Variable for Counter Start Value
    CounterValueEnd VAR WORD    ' Variable for Counter End Value
    CounterValue VAR WORD       ' Variable for Counter Value Diff
    SongSelectValue VAR WORD    ' Variable for Song Select

    TMR1H = 0                'Zero-Out Counter
    TMR1L = 0                'Zero-Out Counter

    TRISA.6 = 0              ' Green
    TRISA.7 = 0              ' Red
    TRISB.2 = 0              ' Yellow
    TRISB.7 = 0              ' Blue
    TRISB.3 = 0              ' Orange

    TRISB.0 = 1              ' Song Select port input

    TRISB.6 = 1              ' Set PortB.6 as the input signal
    TRISB.5 = 1

    Pause 500                ' Wait 500mS for LCD to startup
```

```
LCDOut $fe, 1            ' Clear LCD screen
Pause 1                  ' Wait 1 millisecond

SongSelectValue = 0      ' Initialize variable to zero

LCDOut $fe,$80           ' Move cursor to the beginning of the first line
LCDOut "Guitar Hero Controller" ' Display

Pause 2000               ' Wait 2 seconds
PORTA.6 = 0
PORTA.7 = 0
PORTB.2 = 0
PORTB.7 = 0
PORTB.3 = 0

LCDOUT $FE, $01          'Clear Screen

loop:  LCDOUT $FE, $80
    If SongSelectValue = 0 then
    LCDOUT "Smoke on the Water  "
    LCDOUT $FE, $C0
    LCDOUT "by Deep Purple      "
    endif
    IF SongSelectValue = 1 then
    LCDOUT "Shout at the Devil  "
    LCDOUT $FE, $C0
    LCDOUT "by Motley Crue      "
    endif

    'LCDOut $FE, $C0         'Move Cursor to first line
    'LCDOut "Frequency: ", DEC5 (CounterValue*10), " Hz"


    if portb.0 = 0 then
    SongSelectValue = SongSelectValue + 1
    if SongSelectValue = 2 then
    SongSelectValue = 0
```

```
        endif
 here:  if portb.0 = 0 then
        goto here
        ENDIF
        endif

        if portb.5 = 0 then         'If comparator goes low, then go to the
        LCDOUT $FE, $C0
        gosub counter               ' subroutine counter
        endif

        GoTo loop                   ' Do it forever
End
counter:
        PORTB.7 = 0
        PORTA.6 = 0
        PORTA.7 = 0
        PORTB.2 = 0
        PORTB.3 = 0
        CounterValueStart = (TMR1H << 8) | TMR1L   'Initial Count Value
        Pause 100
        CounterValueEnd = (TMR1H << 8) | TMR1L   'end Count Value
        CounterValue = CounterValueEnd - CounterValueStart
        TMR1H = 0                   'Zero-Out Counter
        TMR1L = 0                   'Zero-Out Counter

        if SongSelectValue = 0 then

        if ((CounterValue >= 14 & CounterValue <= 16) | (CounterValue >= 26 & CounterValue <= 33)) then
        PORTB.7 = 1
        PORTA.6 = 0
        PORTA.7 = 0
        PORTB.2 = 0
        PORTB.3 = 0
        endif
        if ((CounterValue >=34 & CounterValue <= 37) | CounterValue = 17) then
        PORTB.7 = 0
```

```
PORTA.6 = 1
PORTA.7 = 0
PORTB.2 = 0
PORTB.3 = 0
endif
if (CounterValue >= 18 & CounterValue <= 20 ) then
PORTB.7 = 0
PORTA.6 = 0
PORTA.7 = 1
PORTB.2 = 0
PORTB.3 = 0
endif
if (CounterValue >= 21 & CounterValue <= 23 ) then
PORTB.7 = 0
PORTA.6 = 0
PORTA.7 = 0
PORTB.2 = 1
PORTB.3 = 0
endif
if (CounterValue >=110 & CounterValue <=113 ) then
PORTB.7 = 0
PORTA.6 = 0
PORTA.7 = 0
PORTB.2 = 0
PORTB.3 = 1
ENDIF
ENDIF

IF SongSelectValue = 1 then

if ((CounterValue >= 14 & CounterValue <= 16) | (CounterValue >= 26 & CounterValue <= 33)) then
PORTB.7 = 1
PORTA.6 = 0
PORTA.7 = 0
PORTB.2 = 0
PORTB.3 = 0
endif
```

```
if ((CounterValue >=34 & CounterValue <= 37) | CounterValue = 17) then
PORTB.7 = 0
PORTA.6 = 1
PORTA.7 = 0
PORTB.2 = 0
PORTB.3 = 0
endif
if (CounterValue >= 18 & CounterValue <= 20 ) then
PORTB.7 = 0
PORTA.6 = 0
PORTA.7 = 1
PORTB.2 = 0
PORTB.3 = 0
endif
if (CounterValue >= 21 & CounterValue <= 23 ) then
PORTB.7 = 0
PORTA.6 = 0
PORTA.7 = 0
PORTB.2 = 1
PORTB.3 = 0
endif
if (CounterValue >=110 & CounterValue <=113 ) then
PORTB.7 = 0
PORTA.6 = 0
PORTA.7 = 0
PORTB.2 = 0
PORTB.3 = 1
ENDIF
ENDIF

PAUSE 300
RETURN
```