5-2022

Vinyl Player

Cooper Adams
Trinity University, cadams4@trinity.edu

Joshua Geer
Trinity University, jgeer@trinity.edu

Drew Gray
Trinity University, agray1@trinity.edu

TRINITY UNIVERSITY

Vinyl Player

ENGR-4367-1

Instructor: Dr. Nickels

Group P2: Cooper Adams, Joshua Geer, Drew Gray

5/16/2022

Table of Contents

Design Summary

The goal for our final project was to design and manufacture a record player capable of producing music using a 12 inch LP or 7-inch single vinyl record. When powered, the record player spins the vinyl at a specific speed, and using the needle sound vibrations are sent to an audio amplifier to output sound. A motor is used to control the desired speed and rotate the plate which the vinyl record is placed on. Figure 1 shows a simplified sketch of the needle cartridge system; the system consists of a needle and an attached magnet that is placed near a solenoid to generate an electrical signal based on the needle arm movement. The needle is placed in the grooves of the record causing the needle arm to vibrate, which produces an electrical signal in the solenoid due to the magnet.

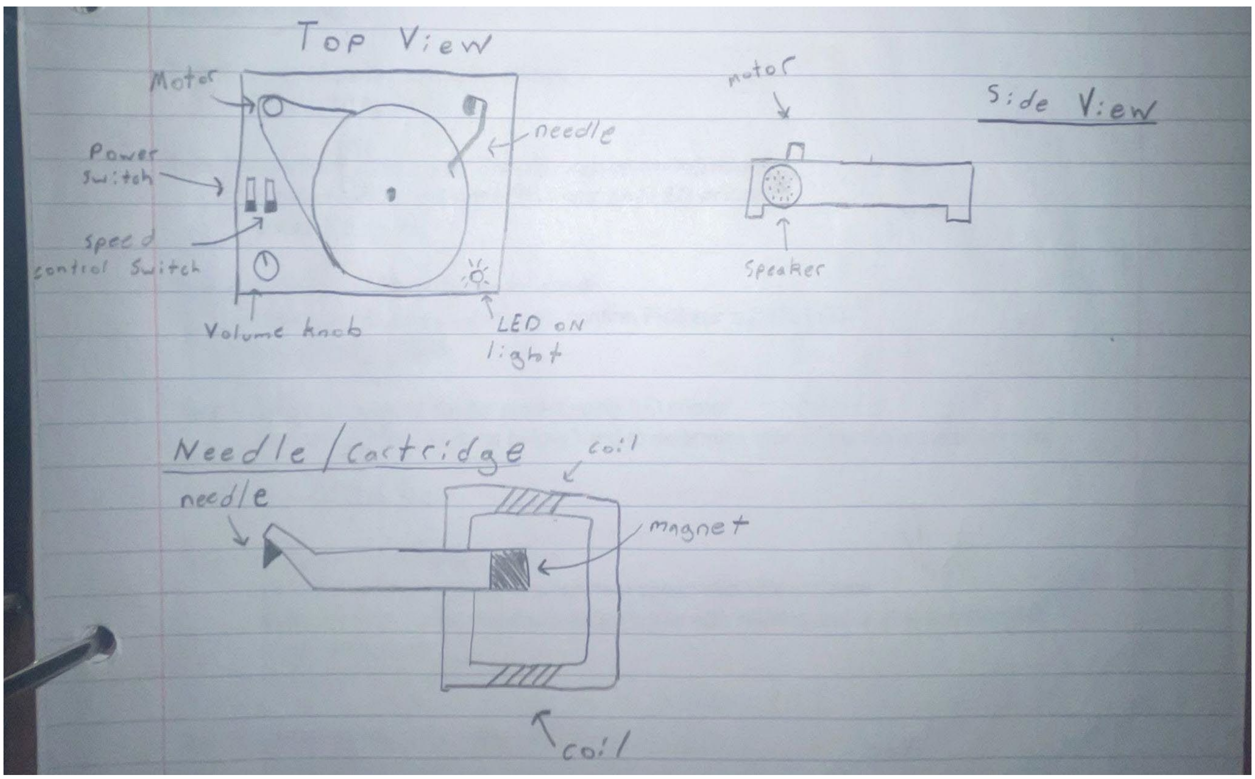

Figure 1: Initial sketches of design

The record player includes a power switch that turns on the record player when flipped and activates an LED indicator light to confirm the ON state of the device. Additionally, a speed control switch is implemented for the user to choose between low and high-speed modes (low speed is 33 ⅓ RPM for a 12-inch record and high speed is 45 RPM for a 7-inch record). Two LEDs are included to indicate to the user what speed mode is selected. To prevent the device from running continuously, a microphone is used to read sound from the speaker and shut off the device after a set time if no sound is being produced. Lastly, a volume control knob is included for the user to change the volume at which music is played.

System Details

The entire project is powered by two 12V DC batteries, one for the motor and PIC circuit, and one for the audio amplifier. Having two completely separate circuits prevents noise generated by the PIC or DC motor from picking up in the speaker, thereby increasing overall audio quality. A switch in the bottom left starts the record player, starting the DC motor and lighting up a red LED. The second switch functions as a speed switch, and when it is switched up, the slower motor setting runs (33 RPM), and a green LED lights up. When switched down, the motor switches to the faster speed (45 RPM), and the yellow LED lights up. A potentiometer volume knob is placed at the bottom right-hand side to allow one to adjust the level of the speaker. A 5V voltage regulator was used to step down the voltage for the audio sensor and PIC, ensuring that each component can operate in a safe voltage range. An audio sensor was included and drilled into the back right side of the player which picks up the sound of the speaker. When the record is done playing, the audio sensor sends a digital trip signal and stops the motor and lights, functioning as an auto-stop system. In order to restart the record player, one must flip the power switch off and back on. The final project can be seen in Figure 2 and 3.

The vinyl casing was designed in Fusion 360 and cut using the STEPCRAFT CNC. Three layers were cut from stock material that was originally from old shelves, allowing us to reduce our overall budget. Velcro strips were used to attach the layers to provide easy access to the electronics. The motor was designed offset to the spinning platter to promote an aesthetic appeal to the record player, as the user can see the belt moving between the motor and the platter. A 1 to 3 gear ratio was used so that a high RPM was not necessarily needed to power the player. The motor-to-belt connector was laser cut and glued together to get the best overall tolerance. The vinyl platter was also laser cut and glued together in several layers, to get a heavy platter that would improve stability. The platter was connected to a thrust bearing and spacer around the shaft, allowing for free spinning despite a substantial torque load. This setup is seen in Figure 4. The logic was programmed in PICBasic Pro using a 16F88 PIC microcontroller. The system flowchart and circuit diagram are seen in Figure 5 and 6.

Figures 2,3: Final Model of the Vinyl Player

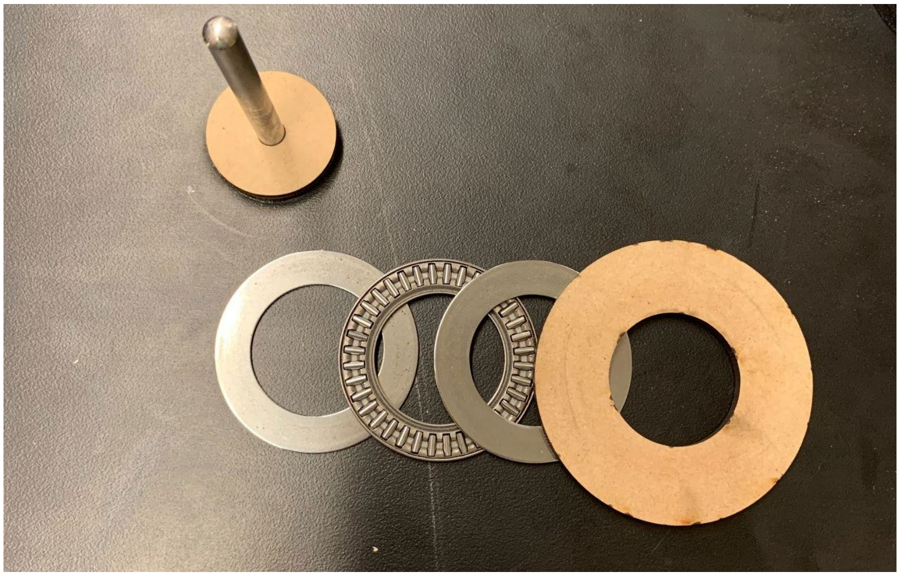

Figure 4: Thrust bearing and spacer placed under the vinyl platter to allow free movement

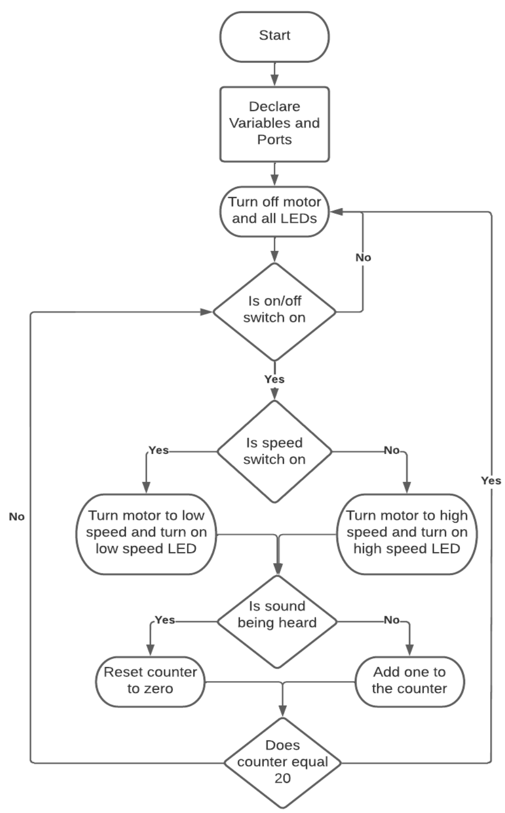

Figure 5: Program Flowchart of the Record Player

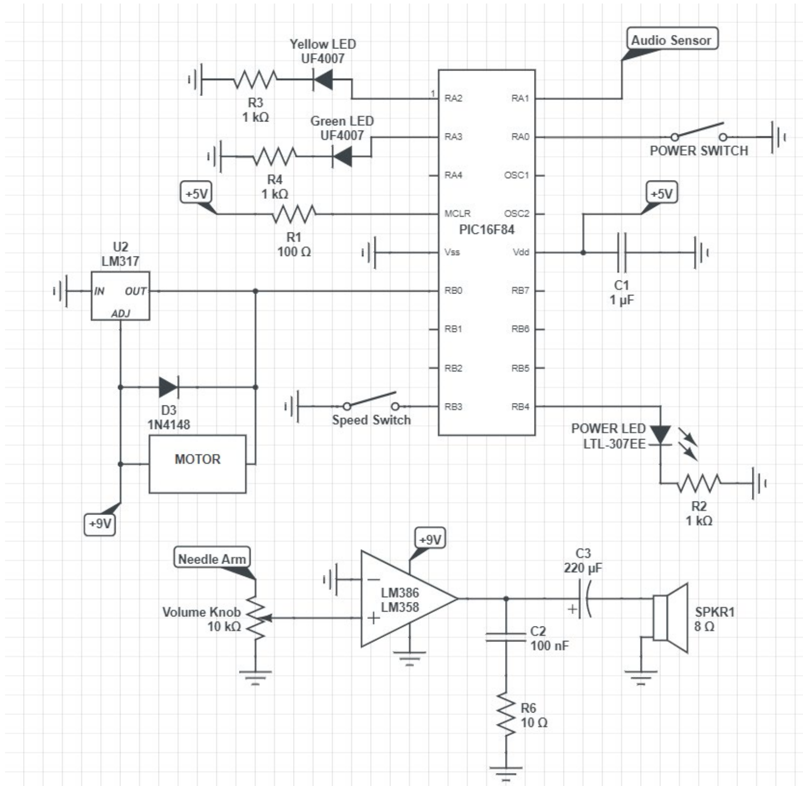

Figure 6: Circuit Diagram of the Record Player

Design Evaluation

Our final product was successful at including functional elements from all six required categories outlined in the final demonstration rubric. The finalized device was able to rotate the record at 2 predetermined speeds, the needle cartridge successfully generated a signal voltage, and that signal was sent to the audio amplifier and correctly altered to produce the desired sound using the speaker. Using the sound output of the speaker, the microphone was able to determine if music was being played and correctly shut off the motor when no sound was heard. The speed control switch, power switch, and volume control potentiometer worked exactly as desired.

Output display

The output display consisted of the power indicator LED and the two LEDs that indicate the speed mode. All three LEDs functioned correctly according to the user selected inputs. The power LED correctly lights up only when the power switch is set to ON mode. Additionally, the yellow LED lights up when the speed switch is set to low speed and the green LED lights up when the speed switch is set to high speed. Overall, the output displays functioned as designed in a reliable fashion, but with the minimal effort required to control LEDs using a microcontroller it likely will earn 10 out of 20 points for this rubric category.

Audio Output Device

Our audio output was the most important category for our project as it is required for a functional record player. While we initially considered fabricating our needle arm and cartridge, our final product used a prebuilt needle arm/cartridge and we were successful at producing a voltage signal from the record. The audio amplifier fulfilled our two requirements: to amplify the voltage signal being received from the needle cartridge and to control the volume of the music signal using a potentiometer. Additionally, the amplifier output was successfully sent to an external speaker to supply sound. The amplifier circuit does an acceptable job at producing a wave with minimal noise; however, the extreme upper and lower ranges of the music waveform still experience unwanted noise, so some further improvements to the circuit's signal filtering capabilities could be made. The needle and the audio amplifier required significant (but not substantial) external research to successfully implement so we believe 15 out of 20 points is a reasonable grade.

Manual User input

The user input consists of the power switch, speed control switch, and volume knob potentiometer, all of which were implemented correctly and reliably. The power switch activates the motor as well as the power indicator LED with no error in performance. The speed switch changes the motor speed with negligible delay time and our microcontroller activates the correct speed LED according to the position of the speed switch. The potentiometer effectively alters the volume of sound being output by the speaker.The needle arm could be considered a user input (it must be placed on the record to produce sound) and it was mounted securely with the ability to move along the grooves of the vinyl record with negligible unwanted movement. All user inputs were securely fastened to the top of our record player and simple enough for a new user to understand. Due to the reliability of the user inputs and the moderate amount of research and effort put into this category, we think 10-15 out of 20 points is a reasonable grade for this category.

Automatic Sensor

The microphone was our only automatic sensor and although it was successfully implemented it did not function exactly as anticipated. Our microphone was a pre fabricated component that was tuned to the desired sensitivity using its on board potentiometer. Tuning the microphone was a tedious process, as it needed to be sensitive enough to pick up quiet sounds produced by the speaker but not so sensitive that the motor sounds would be heard. While the microphone did successfully stop our motor if no sound was being produced, it did not always shut off after our exact setpoint time was reached. We wanted the motor to shut off after 6 seconds of no sound; however, due to either the code not being optimized or the microphone sensitivity not set perfectly, it would sometimes shut off sooner than the desired 6 seconds. Using the microphone required moderate research effort as it was not discussed in class, and while it didn’t function perfectly it would never shut off the motor while sound was being output by the speaker. We think our automatic sensor implementation is worth 10-15 out of 20 points based on the rubric.

Actuators, mechanisms, and hardware

The only actuator used in our final design was a DC motor which was successfully implemented and rotated the plate at the correct speed to turn the record. Our motor was controlled using hardware pulse width modulation, which utilizes prebuilt circuitry within the PIC16F88 to set certain pins to a constant pulse while the rest of the program is executed. This was essential to our project because when we initially used normal PWM it would not hold our specified motor speed constant when executing other lines of code. The motor was tuned to rotate the plate at 33 ⅓ RPM, but due to time constraints we did not construct a way to hold 7 inch records on our turntable so the high speed was not precisely tuned to 45 RPM but to roughly 40 BPM instead. In order to minimize the required torque significant research was done to select bearings and design the turntable so the record would spin fast enough. Overall, we think a score of 15 out 20 is reasonable for this category.

Logic, Processing, and Control

Our project made use of open loop control in order to set the motor to the desired speed and worked as desired. Using a continuous while loop, we checked the condition of each switch before setting the motor to the correct speed using hardware pulse width modulation. The program also includes a counter in a subroutine that checks our microphone condition. On every iteration of our main code, we check to see if sound is being played using the on chip analog to digital converter. If no sound is recorded a counter is incremented and the program is delayed for one second. If a sound is heard the counter is reset to zero. Once the counter reaches a specified value the motor and all LEDs correctly shut off to prevent continuous movement of the needle. Our Logic, processing, and control utilized HPWM which required significant outside research as well as some instructor assistance, so we believe 15 out of 20 points is reasonable.

Partial Parts List

Part/Component	Model Number	Price	Source
Audio sensor	TS-US-115-CA	$6.69	Amazon
Adjustable Voltage Regulator	LM317	Free?	Electronics Shop
Low Voltage Audio Power Amplifier	LM386	Free?	Electronics Shop
Voltage Regulator Transistor	7805CT	Free?	Electronics Shop
Needle and Cartridge	Fit 8” Ruby	$12.63	Walmart
12V Reversible Gear Head DC Motor	162191	19.95	Jameco

Lessons Learned

One of the biggest lessons we learned through this project was to purchase high-quality parts and components and research extensively before buying anything. One of the biggest problems we had towards the end of this project was having to deal with our lower-quality audio sensor. The way that this sensor was designed was it outputted a high voltage PWM wave when there was no sound being heard (around 5 volts, and a lower voltage PWM wave when sound was being heard (around 3 volts). Because this was not ideal for a PIC to digitally read we had to use analog to digital conversion, and find a suitable range of values to tell the pic whether the audio sensor was hearing sound or not. This in conjunction with the very finicky sensitivity potentiometer made this sensor very hard to work with and very unreliable. In the future, we would try to research more to find a sensor that produces a digital output and has a more reliable and predictable sensitivity potentiometer.

The next lesson we learned was that it was very important to plan out our time and be smart about how we allocate our time to complete tasks. Throughout this project, there were tasks that we assumed would take a much shorter amount of time but ended up taking longer than we thought, bleeding into the time we had set aside for other classes and projects. It is important to not underestimate the debugging process and the time it takes to understand why or why something is not working in a particular way. Additionally, it would have been more beneficial to distribute smaller tasks over a few days instead of putting large tasks in a couple of days. It was never easy to have to complete an entire portion of the code or build a working PCP board in one day. In the future, we will make sure to allocate our time differently.

Another lesson we learned was to test everything as you build and code it. There were many times when we had completed entire sections of code or built complete circuits without testing them and then wondered why they did not work the way we planned them to in the testing process. Because of this, we spent way more time debugging and rebuilding than we would have if we had built and coded in small sections to ensure that every subsection worked exactly how we designed it to. An important example of this was our use of the audio sensor. We had purchased this part and assumed we knew exactly how it worked without much testing. Towards the end of the project, we realized that this sensor was not completely suited for our purposes, but we had unfortunately run out of time to order a new one.

The last thing we learned from this project was the importance of splitting up tasks as there was always something to be doing. Many tasks in this project were individual (building a circuit, initial coding, etc.) and sometimes other tasks were put on hold until other tasks were completed. This sometimes resulted in members not being able to do anything during our designated project time. However, we realized towards the end that there is always research, testing, or some sort of productive activity that was available to do. Luckily, this was a very small problem within our group, but if we had realized that it was happening sooner we could have worked more efficiently and possibly gotten tasks done more quickly and effectively.

Appendix

PICBasicPro PIC16f88 Control Code

```
'***************************************************************
'*  Name    : Vinyl Control                                    *
'*  Author  : Joshua Geer, Cooper Adams, Andrew Gray           *
'*  Notice  : Copyright (c) 2022                               *
'*          : All Rights Reserved                              *
'*  Date    : 5/8/2022                                         *
'*  Version : 1.0                                              *
'*  Notes   : This code implements the vinyl control for our   *
'*          : constructed vinyl player                         *
'***************************************************************
'--Define configuration settings-------------------------------------------------
#CONFIG
      CONFIG _CONFIG1, _INTRC_IO & _PWRTE_ON & _MCLR_OFF & _LVP_OFF
#endconfig

'--Set internal oscillator frequency---------------------------------------------
define OSC 8
OSCCON.4 = 1
OSCCON.5 = 1
OSCCON.6 = 1 ' 8Mhz

'--Delacre Variables and Ports---------------------------------------------------
power switch    var PORTA.0
speed light LOW var PORTA.2 'yellow led
speed_light_HI  var PORTA.3 'green led

speed switch    var PORTB.3
power_led       var PORTB.4 'red led

check           var byte
counter         var byte
mic             var word
stoptime        var byte

'--Enable PORTB pull-ups---------------------------------------------------------
OPTION_REG = $7F

'--Set up ADCON1, digital converters, and I/O configuration----------------------
ADCON1 = %10000000 ' Right-justify results (lowest 10 bits)
ANSEL = %00000010
TRISB = %00001000
TRISA = %00000011

'--Set up other important parameters (hpwm and ADC configuration)----------------
define CCP1_REG PORTB
define CCP1 BIT 0
define ADC BITS        10      ' Set number of bits in result
define ADC CLOCK       3       ' Set clock source (3=rc)
define ADC_SAMPLEUS    50      ' Set sampling time in uS

'--Initialize Variables----------------------------------------------------------
counter = 0
stoptime = 10         'how many seconds of silence to turn off the motor
low speed light LOW
low speed light_HI
low power_led

'--Main Loop---------------------------------------------------------------------
while(1)
    if(power switch == 1 && counter < stoptime) then
        high power_led
```

```
        if(speed switch == 0)then                       'set speed based on speedswitch
            hpwm 1,162,1000                             'using hpwm which will run in
            high speed light LOW                        'the background and not stop
            low speed light HI                          'when the mic checks for sound
        elseif(speed switch == 1)then
            hpwm 1,200,1000
            low speed_light_LOW
            high speed_light_HI
        endif
    elseif(power_switch == 0 || counter >= stoptime) then 'if no sound plays for
two
        hpwm 1,0,1000                                   'seconds turn off motor and
        low power led                                   'and LEDs
        low speed light HI
        low speed light LOW
        while(power switch == 1)                        'turn powerswitch off and
        wend                                            'to rest code
        counter = 0
    endif

    adcin 1, mic                                'If audio sensor detects sound then

    check = mic/4                               'add one to counter and wait a 1 s
    if(check > 200 && counter < stoptime) then
        counter = counter + 1
        pause 1000
    elseif(check < 230) then
        counter = 0
    endif
wend
```