**The Quiz Board**

Nathan Petersen | 07.26.2013

**What I learned:**

I learned many things while making this quiz board, but the most prevalent thing was how to teach other people about electronics and computer programming. This is a useful thing in life because computers and electronics are forever growing around us, so a person being able to use them is absolutely necessary. I also learned that computer software and electronic hardware can be used interchangeably. You can control electronics with code and then feed electronic signals back into the computer for it to make decisions. Since this was my first hybrid project (electronic circuits with a programmed microcontroller), I thought it was very neat that it’s fairly easy to combine the two aspects to create an excellent product.

**How it’s made:**

The quiz board has three parts to it: the wooden box, the electrical components, and the software to make it run. The wooden box was designed to be an elegant interface for the project. It was made using woodworking tools to construct the oak display board and the enclosure structure. The sides of the box were spray painted black to draw attention to the top display board. The electronic part of the project uses many different components soldered together on a custom designed circuit board. These components include switches, LEDs, a speaker, photodiode, resistors, transistors, and a battery. The software runs on an Arduino microcontroller which interfaces with the electronic circuitry. The software’s purpose is to turn the LEDs on and off based on the switch states and photodiode values. The end result is that the quiz can be used with an unlimited amount of questions and be programmed to display original audio-visual feedback for the user.

**Theoretical description:**

This quiz board solves the number one problem with all quiz boards: the fact that only a set number of questions can be asked. Therefore, when a user is done answering the questions, the board is rendered useless. To allow this quiz board to have the ability to ask an infinite amount of questions, it uses an optical card reader. Each question is printed on a paper card with holes punched in it that correspond to the answer. The quiz board uses binary encoding to convert three sensors into a 0-7 number input. The machine then reads what the answer should be based on the holes in the cards and then waits for the user to press an answer. If the input from the user is correct, the “win” LEDs are illuminated and a happy sound is played. When the user gets an answer wrong, the “lose” LEDs go on and a sad sound is played.

**Technical description:**

The quiz board has a microcontroller in it called an Arduino. This computer-like part controls all the buttons, LEDs, photodiodes, and speaker. It is programmed on a computer to run specific instructions. See code below. All the individual parts are wired to the inputs and outputs of the Arduino.

Speaker: The Arduino can only push so much power from its outputs so it’s necessary to use a transistor to amplify the signal to larger amplitudes which will result in a louder sound. The Arduino is wired through a resistor into the base of a NPN transistor. Then, the speaker is wired in series with another resistor from power to ground.

Win/Lose LEDs: The status LEDs use another transistor to amplify the signal, much like the speaker. The LEDs are all connected in parallel so that less voltage is needed to drive them (but more current). When current is run across an LED, the voltage experiences a drop. Since the board uses four LEDs, the last LED wouldn’t turn on if they were attached in series.

Switches: The switches serve as input to the Arduino. The Arduino can only read voltage inputs, so somehow it has to convert a pressed switch into a voltage signal and a not pressed switch into no voltage. This is done by using a voltage divider. When the switch is not pressed, the Arduino is forced to connect to ground through a resistor. Then, when the switch is pressed, 5V comes into the divider. The current takes the path of least resistance, so 5V is sent directly into the Arduino. This creates the 5V vs. 0V spread that is needed to detect a pressed switch.

Photodiode Sensors: These detect the holes in the cards by measuring light. These work on the same principle as the voltage divider that the switches use except the switches are replaced with an analog resistor, as opposed to digital. This creates a spread of voltage into the Arduino based on the photodiodes resistance.

LEDs for Sensors: These LEDs are connected independently to the power and ground. They are connected in parallel so that they are all lit at the same brightness and they don’t experience a voltage drop.

**Arduino Code:**

**#include "pitches.h"**

**//constant values the relate notes to frequencies.**

**// Example:**

**//** **#define NOTE\_A4 440**

**//#define NOTE\_AS4 466**

**//#define NOTE\_B4 494**

**int correctnotes[] = {NOTE\_C5, NOTE\_E5, NOTE\_G5, NOTE\_E5, NOTE\_G5, NOTE\_C6, NOTE\_G5, NOTE\_C6, NOTE\_E6, NOTE\_C6, NOTE\_E6, NOTE\_G6, NOTE\_E6, NOTE\_G6, NOTE\_C7, };**

**int incorrectnotes[] = {NOTE\_C3, NOTE\_FS2};**

**int incorrectnoteslength[] = {2.2, 2};**

**int firsttime = 1;**

**int threshold = 950;**

**int btn = -1;**

**//Digital pins**

**int ledlose = 8;**

**int ledwin = 7;**

**int ledsensor1 = 2;**

**int ledsensor2 = 3;**

**int ledsensor3 = 4;**

**int button1 = 9;**

**int button2 = 10;**

**int button3 = 11;**

**int button4 = 12;**

**int button5 = 13;**

**int speaker = 6;**

**//Analog pins**

**int sensor1 = A3;**

**int sensor2 = A2;**

**int sensor3 = A1;**

**void setup() {**

**playsound2();**

**Serial.begin(9600);**

**pinMode(ledlose, OUTPUT);**

**pinMode(ledwin, OUTPUT);**

**pinMode(ledsensor1, OUTPUT);**

**pinMode(ledsensor2, OUTPUT);**

**pinMode(ledsensor3, OUTPUT);**

**pinMode(sensor1, INPUT);**

**pinMode(sensor2, INPUT);**

**pinMode(sensor3, INPUT);**

**pinMode(button1, INPUT);**

**pinMode(button2, INPUT);**

**pinMode(button3, INPUT);**

**pinMode(button4, INPUT);**

**pinMode(button5, INPUT);**

**pinMode(speaker, OUTPUT);**

**}**

**void playsound0() { //lose sound**

**for (int thisNote = 0; thisNote < 2; thisNote++) {**

**// to calculate the note duration, take one second**

**// divided by the note type.**

**//e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.**

**int noteDuration = 1000/incorrectnoteslength[thisNote];**

**tone(speaker, incorrectnotes[thisNote],noteDuration);**

**// to distinguish the notes, set a minimum time between them.**

**// the note's duration + 30% seems to work well:**

**int pauseBetweenNotes = noteDuration \* 1.30;**

**delay(pauseBetweenNotes);**

**// stop the tone playing:**

**noTone(speaker);**

**}**

**}**

**void playsound1() { //win sound**

**for (int thisNote = 0; thisNote < (sizeof(correctnotes)/sizeof(\*correctnotes)); thisNote++) {**

**tone(speaker, correctnotes[thisNote], 40);**

**delay(40);**

**noTone(speaker);**

**}**

**}**

**void playsound2() { //startup sound**

**int upper = 1000;**

**int lower = 30;**

**for (int i = lower; i < upper; i+=2) {**

**tone(speaker, i, 10);**

**delay(1);**

**}**

**for (int i = upper; i > lower; i-=2) {**

**tone(speaker, i, 10);**

**delay(1);**

**}**

**}**

**void playsound3() { //showoff sound**

**int length = 75;**

**int btn = 0;**

**while (btn != 14) {**

**int rlo = random(200,1000);**

**int rhi = random(1250,2000);**

**tone(speaker, rlo, length);**

**delay(length \* 1.00);**

**tone(speaker, rhi, length);**

**delay(length \* 1.00);**

**btn = readbuttons();**

**}**

**}**

**void pulseleds() {**

**digitalWrite(ledwin, HIGH);**

**digitalWrite(ledlose, LOW);**

**delay(50);**

**digitalWrite(ledlose, HIGH);**

**digitalWrite(ledwin, LOW);**

**delay(50);**

**}**

**int readbuttons() {**

**int btn1 = digitalRead(button1);**

**int btn2 = digitalRead(button2);**

**int btn3 = digitalRead(button3);**

**int btn4 = digitalRead(button4);**

**int btn5 = digitalRead(button5);**

**int btn = 0;**

**if (btn1 == 1) {btn = btn + 1;}**

**if (btn2 == 1) {btn = btn + 2;}**

**if (btn3 == 1) {btn = btn + 4;}**

**if (btn4 == 1) {btn = btn + 8;}**

**if (btn5 == 1) {btn = btn + 16;}**

**return btn;**

**}**

**int readsensors() {**

**int svalue1 = analogRead(sensor1);**

**int svalue2 = analogRead(sensor2);**

**int svalue3 = analogRead(sensor3);**

**int answer = 0;**

**if (svalue1 < threshold) {answer += 1;}**

**if (svalue2 < threshold) {answer += 2;}**

**if (svalue3 < threshold) {answer += 4;}**

**return answer;**

**}**

**void keyboard() { //turns device into a keyboard if outside four buttons are pressed**

**delay(200);**

**int length = 10;**

**int btn = 0;**

**int sensor = 0;**

**int octave = 1;**

**while (btn != 14) {**

**btn = readbuttons();**

**sensor = readsensors();**

**octave = 1;**

**if (sensor == 4 || sensor == 5) {octave = 1;}**

**if (sensor == 7) {octave = 2;}**

**if (sensor == 3) {octave = 4;}**

**switch (btn) {**

**case 0:**

**noTone(speaker);**

**digitalWrite(ledwin, LOW);**

**digitalWrite(ledlose, LOW);**

**break;**

**case 1:**

**tone(speaker, 146 \* octave);**

**digitalWrite(ledwin, HIGH);**

**digitalWrite(ledlose, HIGH);**

**break;**

**case 2:**

**tone(speaker, 174 \* octave);**

**digitalWrite(ledwin, HIGH);**

**digitalWrite(ledlose, HIGH);**

**break;**

**case 4:**

**tone(speaker, 196 \* octave);**

**digitalWrite(ledwin, HIGH);**

**digitalWrite(ledlose, HIGH);**

**break;**

**case 8:**

**tone(speaker, 220 \* octave);**

**digitalWrite(ledwin, HIGH);**

**digitalWrite(ledlose, HIGH);**

**break;**

**case 16:**

**tone(speaker, 262 \* octave);**

**digitalWrite(ledwin, HIGH);**

**digitalWrite(ledlose, HIGH);**

**break;**

**}**

**}**

**}**

**void pitchbend() { //bends pitch based on user input**

**delay(200);**

**int startfreq = 440;**

**int inc = 0;**

**int upper = 2000;**

**int lower = 100;**

**int btn = 0;**

**while (btn != 14) {**

**btn = readbuttons();**

**if (btn == 0) {**

**noTone(speaker);**

**} else {**

**if (btn == 5) {inc += -1; delay(2);}**

**if (btn == 6) {inc += -1; delay(10);}**

**if (btn == 12) {inc += 1; delay(10);}**

**if (btn == 20) {inc += 1; delay(2);}**

**if (startfreq + inc > upper) {inc = upper - startfreq;}**

**if (startfreq + inc < lower) {inc = -1 \* (startfreq - lower);}**

**if (btn == 5 || btn == 6 || btn == 4 || btn == 12 || btn == 20) {tone(speaker, startfreq + inc);}**

**if (btn == 17) {inc = 0; delay(200);}**

**}**

**}**

**}**

**void loop() {**

**noTone(speaker);**

**if (firsttime == 1) {**

**firsttime = 0;**

**digitalWrite(ledwin, LOW);**

**digitalWrite(ledlose, LOW);**

**}**

**int btn1 = digitalRead(button1);**

**int btn2 = digitalRead(button2);**

**int btn3 = digitalRead(button3);**

**int btn4 = digitalRead(button4);**

**int btn5 = digitalRead(button5);**

**if (btn1 == HIGH) {btn = 1;} else**

**if (btn2 == HIGH) {btn = 2;} else**

**if (btn3 == HIGH) {btn = 3;} else**

**if (btn4 == HIGH) {btn = 4;} else**

**if (btn5 == HIGH) {btn = 5;} else {btn = -1;}**

**if (btn1 == HIGH && btn2 == HIGH && btn3 == HIGH && btn4 == HIGH && btn5 == HIGH) {playsound3();}**

**if (btn1 == HIGH && btn2 == HIGH && btn3 == LOW && btn4 == HIGH && btn5 == HIGH) {keyboard();}**

**if (btn1 == HIGH && btn2 == LOW && btn3 == LOW && btn4 == LOW && btn5 == HIGH) {pitchbend();}**

**digitalWrite(ledsensor1, HIGH);**

**digitalWrite(ledsensor2, HIGH);**

**digitalWrite(ledsensor3, HIGH);**

**int svalue1 = analogRead(sensor1);**

**int svalue2 = analogRead(sensor2);**

**int svalue3 = analogRead(sensor3);**

**int answer = 0;**

**if (svalue1 < threshold) {answer += 1;}**

**if (svalue2 < threshold) {answer += 2;}**

**if (svalue3 < threshold) {answer += 4;}**

**if (answer == 7) {pulseleds();}**

**digitalWrite(ledwin, LOW);**

**digitalWrite(ledlose, LOW);**

**if (btn == answer) {**

**digitalWrite(ledwin, HIGH);**

**playsound1();**

**}**

**if (btn != answer && btn != -1 && answer != 7) {**

**digitalWrite(ledlose, HIGH);**

**playsound0();**

**}**

**if (btn == -1) {digitalWrite(ledwin, LOW); digitalWrite(ledlose, LOW);}**

**Serial.print(svalue1);**

**Serial.print("\t");**

**Serial.print(svalue2);**

**Serial.print("\t");**

**Serial.print(svalue3);**

**Serial.print("\t");**

**Serial.print(answer);**

**Serial.print("\t");**

**Serial.print(btn);**

**Serial.**print**("\t");**

**Serial.println("");**

}