ECS724 Interactive Digital Media Techniques · Autumn 2014

Mini-Assignment 1: Arduino

Bei Liang 140253278

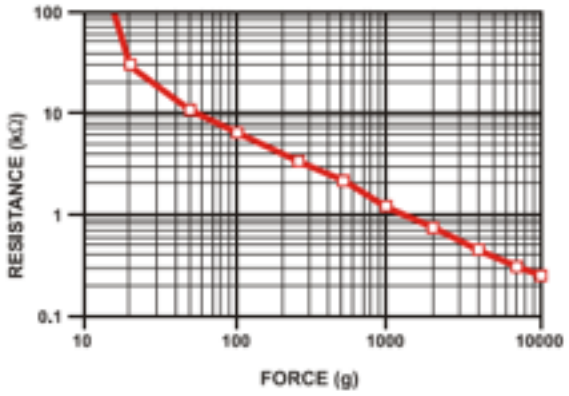
Step 1: Connect and read the FSR

I connect the circuit as instruction sheet writes. Since the voltage of FSR is proportional to the pressure on the sensor, I make the LED intensity to show the variance. Therefore, I declare the *ledPin* as an *OUTPUT*. After uploading, the results in the serial monitor become bigger and LED turns brighter when the pressure is higher, which shows the resistance of FSR is lower.

Answers of the Questions

1. According to the serial monitor, the sensor puts out a range of value from 0 to 245.
2. No, the values do not scale linearly with finger pressure. The force vs. resistance characteristic shown in the figure[1] below provides an overview of FSR typical response behaviour, which data is plotted on a log/log format. Then we can get a rough R_{FSR} based on the figure. So the value will be

$$V_{OUT} \approx \frac{V}{1 + R_{FSR}/R}$$



Step 2: Use the FSR value to create a tone

I think it has the same principle as Step 1, so I replace the LED part with a piezo. In order to cover two octave, I map the voltage range from 0-1023 to 220-880 and then use the *tone* function to generate tones.

Answers of the Questions

1. The speaker can produce a range of frequencies between 220Hz and 850Hz shown in the serial monitor.
2. It is difficult to play a melody on FSR, because we don't know the exact pressure to play the note we want.
3. We may use different buttons as a keyboard to make it easier to play.

STEP 3: Use the buttons to make a simple keyboard

I plan to play from C4 to G4 separately. I use different transistors connected with the buttons to change the voltage of each button, and the voltage can be shown in the

[1] https://www.sparkfun.com/datasheets/Sensors/Pressure/fsrguide.pdf

serial monitor. Then I use *if* structure to play different notes under different voltages by *tone* function.

Answers of the Questions
1. The instrument is easier to play than the one in Step 2.
2. More expressive.

Step 4: Add vibrato using the FSR
I take this step as a kind of combination of Step 2 and Step 3. I add another *if…else…* in every *if else* structure of Step 3, which makes FSR add vibrato to each note. For example, when I press a button which means C4(262Hz), the *tone* function will just play C4 if there is no pressure on FSR. Otherwise, I will map the value of voltage between 249Hz(-5%) and 275Hz(+5%), which is proportional to the pressure on the FSR.

Answers of the Questions
1. If there is no pressure on the sensor, the centre frequency will be played. If there is little pressure, it will bend the note down. The note will be upper along with the higher pressure.
2. In this way, we can play more vividly, just like play a stringed instrument.
3. We could attach the FSR sensor directly to the surface of buttons to make the instrument easier to play. Therefore we may play more sound effects on the keyboard.

Special Bonus Step: Play a sequence melody
I plan to play a melody of Sherlock Theme from BBC TV Series, so firstly I store the melody in an array and create a variable called *counter* to detect which note in the melody should be played. When the button is pressed, use a boolean called *playing* to toggle the *tone* function. If no button is pressed, add one to *counter* and return to zero when all the notes have been played.