**SIK Keyboard Code**

/\*\*

\* SparkFun Inventor's Kit Project

\* Keyboard Instrument

\* Date: March 29, 2016

\*

\* Description:

\* Use the soft touch potentiometer as a keyboard segmented into

\* 8 keys: C, D, E, F, G, A, B, C. When each key is pressed, the

\* corresponding note is played through a buzzer.

\*

\* Hardware Connections:

\* Arduino | Soft Pot | Buzzer

\* ---------------------------

\* 5V | pin 3 |

\* A0 | pin 2 |

\* GND | pin 1 |

\* 9 | | +

\* GND | | -

\*

\* You will also need to attach a 10k resistor from pin 2 to

\* pin 1 (GND) on the soft pot.

\*

\* License:

\* Public Domain

\*/

// Constants

const int SENSOR\_PIN = 0; // Analog input pin for soft pot

const int BUZZER\_PIN = 9; // PWM digital output pin for buzzer

const int DURATION = 10; // Time (ms) to play a note

// This function is run only once as soon as the Arduino boots

void setup()

{

// Set the buzzer pin as an output

pinMode(BUZZER\_PIN, OUTPUT);

}

// This gets run over and over right after the setup() function

void loop()

{

int sensorValue;

char note = 0;

int freq;

// Read the value (0 - 1023) from the ADC

sensorValue = analogRead(SENSOR\_PIN);

// Map the key pressed to a note

note = findNote(sensorValue);

// If it's a note, play it!

if ( note != 0 ) {

freq = getFrequency(note);

tone(BUZZER\_PIN, freq, DURATION);

delay(DURATION);

}

}

// Given an ADC value (0 - 1023), map it to a note

char findNote(int val)

{

// Return the note based on the key pressed

if ( (val > 10) && (val <= 160) )

{

return 'c';

}

if ( (val > 160) && (val <= 250) )

{

return 'd';

}

if ( (val > 250) && (val <= 350) )

{

return 'e';

}

if ( (val > 350) && (val <= 450) )

{

return 'f';

}

if ( (val > 450) && (val <= 560) )

{

return 'g';

}

if ( (val > 560) && (val <= 690) )

{

return 'a';

}

if ( (val > 690) && (val <= 850) )

{

return 'b';

}

if ( (val > 850) && (val <= 1023) )

{

return 'C';

}

// Return 0 to show that no key was pressed

return 0;

}

// Translate a note (a, b, c, d, e, f, g) to its frequency

int getFrequency(char note)

{

int i;

const int numNotes = 8; // number of notes we're storing

// Arrays containing our notes and frequencies

char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };

int frequencies[] = {262, 294, 330, 349, 392, 440, 494, 523};

// Step though the notes

for (i = 0; i < numNotes; i++) // Step through the notes

{

// If it matches a note in our list, return the frequency

if (names[i] == note)

{

return(frequencies[i]);

}

}

// If we looked through everything and didn't find a note,

// return 0, as we still need to return something.

return(0);

}