

```
//feel free to modify this code as you see fit, but please make
//a habit of giving credit where credit is due.
//I modified this from code written by OFFSHOR3 that can be found at
// http://pastebin.com/gBxu4pDs
```

```
int ledPin = 1;
//the pin the has the led
int speakerPin = 2;
//the pin my speaker is attached to
```

```
// I defined the tempo in terms of the number of milliseconds in an eighth note,
// then defined the rest of the notes relative to the eighth note. This makes
// adjusting the tempo simple as I only need to change the eigh variable.
```

```
#define eigh 100
#define quar eigh*2
#define half quar*2
#define dothalf quar*3
```

```
// The notes are defined by their frequencies. I included an octave variable to make
// key changes easier. Increasing the oct value makes the key higher, while lowering oct will
// lower the key.
// Remember that oct needs to be positive, so to lower the key use decimals above zero. e.g.
// 0.5, 0.25, etc.
```

```
#define oct 4
```

```
#define a4 440*oct
#define aS4 466*oct
#define b4 493*oct
#define c5 523*oct
#define cS5 554*oct
#define d5 587*oct
#define dS5 622*oct
#define e5 659*oct
#define f5 698*oct
#define fS5 740*oct
#define g5 784*oct
#define gS5 831*oct
#define a5 880*oct
#define aS5 932*oct
#define b5 988*oct
```

```
#define c6 1047*oct
#define cS6 1109*oct
#define d6 1175*oct
#define dS6 1245*oct
#define e6 1319*oct
#define f6 1397*oct
#define fS6 1480*oct
#define g6 1568*oct
#define gS6 1661*oct
#define a6 1760*oct
```

```
void setup()
{
  pinMode(ledPin,OUTPUT);
  pinMode(speakerPin, OUTPUT);
}
```

```
void loop()
{
  up_theme();
}
```

```
void beep(unsigned char speakerPin, int frequency, long time)
{
  digitalWrite(ledPin, HIGH);

  int i;
  long delay_time = (long)(1000000/frequency);
  long loop_time = (long)((time*1000)/(delay_time*2));
  for (i=0; i<loop_time; i++)
  {
    digitalWrite(speakerPin,HIGH);
    delayMicroseconds(delay_time);
    digitalWrite(speakerPin,LOW);
    delayMicroseconds(delay_time);
  }

  digitalWrite(ledPin, LOW);

  delay(20);
}
```

```

void up_theme()
{
    //bar 4

    beep(speakerPin, f5, eigh);
    beep(speakerPin, a5, eigh);
    beep(speakerPin, f5, eigh);
    beep(speakerPin, e5, dothalf + quar);
    delay(eigh);

    beep(speakerPin, f5, eigh);
    beep(speakerPin, a5, eigh);
    beep(speakerPin, e5, eigh);
    beep(speakerPin, d5, quar);
    delay(eigh);

    //bar 7
    beep(speakerPin, d5, eigh);
    beep(speakerPin, f5, eigh);
    beep(speakerPin, d5, eigh);
    beep(speakerPin, c5, half);
    delay(eigh);

    beep(speakerPin, d5, eigh);
    beep(speakerPin, a5, quar);
    beep(speakerPin, g5, quar);
    delay(eigh);

    beep(speakerPin, d5, eigh);
    beep(speakerPin, a5, quar);
    beep(speakerPin, g5, quar);
    beep(speakerPin, f5, quar);
    beep(speakerPin, d5, dothalf+quar);
    delay(eigh);

    beep(speakerPin, f5, eigh);
    beep(speakerPin, g5, eigh);
    beep(speakerPin, f5, eigh);
    beep(speakerPin, e5, dothalf+quar);
    delay(eigh);

    //bar 14

```

```
beep(speakerPin, e5, eigh);  
beep(speakerPin, g5, eigh);  
beep(speakerPin, e5, eigh);  
beep(speakerPin, c5, quar);  
delay(eigh);
```

```
beep(speakerPin, c5, eigh);  
beep(speakerPin, e5, eigh);  
beep(speakerPin, c5, eigh);  
beep(speakerPin, aS4, quar);  
delay(eigh);
```

```
beep(speakerPin, aS4, eigh);  
beep(speakerPin, c5, eigh);  
beep(speakerPin, aS4, eigh);  
delay(quar);
```

```
beep(speakerPin, a4, quar);  
beep(speakerPin, aS4, quar);  
beep(speakerPin, c5, quar+eigh);  
beep(speakerPin, d5, eigh);  
beep(speakerPin, e5, quar);  
delay(quar);
```

```
beep(speakerPin, c5, quar);  
beep(speakerPin, d5, quar);  
beep(speakerPin, e5, quar+eigh);
```

```
}
```