Πηγαίοι κώδικες από το Φύλλο Εργαστηριακής Άσκησης: **Ήχος** (και ολίγο φως)

Γνωριμία με το buzzer στην ΠΡΑΞΗ

ΠΡΩΤΟ ΒΗΜΑ:

```
void setup() {
   pinMode(8,OUTPUT);
 }
 void loop() {
  digitalWrite(8,HIGH);
  delay(200);
  digitalWrite(8,LOW);
  delay(200);
}
```

ΤΡΙΤΟ ΒΗΜΑ:

```
int i;
void setup() {
   pinMode(9,OUTPUT);
 }
 void loop() {
  for (i=0;i<=255;i=i+10)
     {
       analogWrite(9, i);
       delay(1000);
      }
}
```

ΠΕΜΠΤΟ ΒΗΜΑ:

```
int buzzer=8;      //set the digital IO pin to control the buzzer
void setup()
{
pinMode(buzzer,OUTPUT);    //set digital IO pin as output mode
}
void loop()
{
unsigned char i,j;    //Define variable
while(1)
{
for(i=0;i<80;i++)      //Make sound with same frequency
{
digitalWrite(buzzer,HIGH);     //Make sound
delay(1);                      //Delay 1ms
digitalWrite(buzzer,LOW);     //No sound
delay(1);                     //Delay 1ms
}
for(i=0;i<100;i++)               // Make sound with other frequency
{
digitalWrite(buzzer,HIGH);     //Make Sound
delay(2);                      //Delay2ms
digitalWrite(buzzer,LOW);      //No sounds
delay(2);                     //Delay 2ms
}
}
}
```

Σύνδεση με παλιότερες εφαρμογές μας

ΜΙΑ πιθανή λύση

```
int buzzer=8;
int zzz;

void setup()
{
pinMode(buzzer,OUTPUT);
}

void loop()
{
unsigned char i;

zzz=analogRead(A0);
if (zzz>100)
   {
     for(i=0;i<100;i++)
       {
        digitalWrite(buzzer,HIGH);
        delay(3);
        digitalWrite(buzzer,LOW);
        delay(3);
       }
    for(i=0;i<50;i++)
       {
        digitalWrite(buzzer,HIGH);
        delay(10);
        digitalWrite(buzzer,LOW);
        delay(10);
       }
   }
}
```

ΔΕΥΤΕΡΟ ΜΕΡΟΣ _ ΜΟΥΣΙΚΗ?

ΠΡΩΤΟ ΒΗΜΑ:

```
#define NOTE_C3  131
#define NOTE_D3  147
#define NOTE_G3  196
#define NOTE_A3  220
#define NOTE_B3  247
#define NOTE_C4  262

#define BUZZER_PIN 8

int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
};

// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
  4, 8, 8, 4, 4, 4, 4, 4
};

void setup() {

  // iterate over the notes of the melody:
  for (int thisNote = 0; thisNote < 8; 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 / noteDurations[thisNote];
    tone(8, melody[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(8);
  }
}

void loop() {
  // no need to repeat the melody.
}
```

ΔΕΥΤΕΡΟ ΒΗΜΑ: Μια πιθανή λύση

```
#define NOTE_C3  131
#define NOTE_D3  147
#define NOTE_G3  196
#define NOTE_A3  220
#define NOTE_B3  247
#define NOTE_C4  262

#define BUZZER_PIN 8

int melody[] = {

  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
};

// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
  4, 8, 8, 4, 4, 4, 4, 4
};

void setup() {

}

void loop() {

  // iterate over the notes of the melody:

  for (int thisNote = 0; thisNote < 8; 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 / noteDurations[thisNote];
    tone(8, melody[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(8);
     }
  delay (2000);
}
```

ΠΑΡΑΡΤΗΜΑ 1 ΡΟΖ ΠΑΝΘΗΡΑΣ

```
/*
***********************************************
 Public Constants
 ***********************************************
 */

#define NOTE_B0  31
#define NOTE_C1  33
#define NOTE_CS1 35
#define NOTE_D1  37
#define NOTE_DS1 39
#define NOTE_E1  41
#define NOTE_F1  44
#define NOTE_FS1 46
#define NOTE_G1  49
#define NOTE_GS1 52
#define NOTE_A1  55
#define NOTE_AS1 58
#define NOTE_B1  62
#define NOTE_C2  65
#define NOTE_CS2 69
#define NOTE_D2  73
#define NOTE_DS2 78
#define NOTE_E2  82
#define NOTE_F2  87
#define NOTE_FS2 93
#define NOTE_G2  98
#define NOTE_GS2 104
#define NOTE_A2  110
#define NOTE_AS2 117
#define NOTE_B2  123
#define NOTE_C3  131
#define NOTE_CS3 139
#define NOTE_D3  147
#define NOTE_DS3 156
#define NOTE_E3  165
#define NOTE_F3  175
#define NOTE_FS3 185
#define NOTE_G3  196
#define NOTE_GS3 208
#define NOTE_A3  220
#define NOTE_AS3 233
#define NOTE_B3  247
#define NOTE_C4  262
#define NOTE_CS4 277
#define NOTE_D4  294
#define NOTE_DS4 311
#define NOTE_E4  330
#define NOTE_F4  349
#define NOTE_FS4 370
#define NOTE_G4  392
#define NOTE_GS4 415
#define NOTE_A4  440
#define NOTE_AS4 466
#define NOTE_B4  494
```

```
#define NOTE_C5  523
#define NOTE_CS5 554
#define NOTE_D5  587
#define NOTE_DS5 622
#define NOTE_E5  659
#define NOTE_F5  698
#define NOTE_FS5 740
#define NOTE_G5  784
#define NOTE_GS5 831
#define NOTE_A5  880
#define NOTE_AS5 932
#define NOTE_B5  988
#define NOTE_C6  1047
#define NOTE_CS6 1109
#define NOTE_D6  1175
#define NOTE_DS6 1245
#define NOTE_E6  1319
#define NOTE_F6  1397
#define NOTE_FS6 1480
#define NOTE_G6  1568
#define NOTE_GS6 1661
#define NOTE_A6  1760
#define NOTE_AS6 1865
#define NOTE_B6  1976
#define NOTE_C7  2093
#define NOTE_CS7 2217
#define NOTE_D7  2349
#define NOTE_DS7 2489
#define NOTE_E7  2637
#define NOTE_F7  2794
#define NOTE_FS7 2960
#define NOTE_G7  3136
#define NOTE_GS7 3322
#define NOTE_A7  3520
#define NOTE_AS7 3729
#define NOTE_B7  3951
#define NOTE_C8  4186
#define NOTE_CS8 4435
#define NOTE_D8  4699
#define NOTE_DS8 4978

/*

  Melody
  Plays a melody
  circuit:
  - 8 ohm speaker on digital pin 8

*/

// #include "pitches.h"


#define BUZZER_PIN 8
#define REST 0

int melody[] = {
  REST, REST, REST, NOTE_DS4,
  NOTE_E4, REST, NOTE_FS4, NOTE_G4, REST, NOTE_DS4,
  NOTE_E4, NOTE_FS4,  NOTE_G4, NOTE_C5, NOTE_B4, NOTE_E4, NOTE_G4, NOTE_B4,
  NOTE_AS4, NOTE_A4, NOTE_G4, NOTE_E4, NOTE_D4,
  NOTE_E4, REST, REST, NOTE_DS4,

  NOTE_E4, REST, NOTE_FS4, NOTE_G4, REST, NOTE_DS4,
  NOTE_E4, NOTE_FS4,  NOTE_G4, NOTE_C5, NOTE_B4, NOTE_G4, NOTE_B4, NOTE_E5,
  NOTE_DS5,
  NOTE_D5, REST, REST, NOTE_DS4,
  NOTE_E4, REST, NOTE_FS4, NOTE_G4, REST, NOTE_DS4,
  NOTE_E4, NOTE_FS4,  NOTE_G4, NOTE_C5, NOTE_B4, NOTE_E4, NOTE_G4, NOTE_B4,

  NOTE_AS4, NOTE_A4, NOTE_G4, NOTE_E4, NOTE_D4,
  NOTE_E4, REST,
  REST, NOTE_E5, NOTE_D5, NOTE_B4, NOTE_A4, NOTE_G4, NOTE_E4,
  NOTE_AS4, NOTE_A4, NOTE_AS4, NOTE_A4, NOTE_AS4, NOTE_A4, NOTE_AS4, NOTE_A4,
  NOTE_G4, NOTE_E4, NOTE_D4, NOTE_E4, NOTE_E4, NOTE_E4
};

int durations[] = {
  2, 4, 8, 8,
  4, 8, 8, 4, 8, 8,
  8, 8,  8, 8, 8, 8, 8, 8,
  2, 16, 16, 16, 16,
  2, 4, 8, 4,

  4, 8, 8, 4, 8, 8,
  8, 8,  8, 8, 8, 8, 8, 8,
  1,
  2, 4, 8, 8,
  4, 8, 8, 4, 8, 8,
  8, 8,  8, 8, 8, 8, 8, 8,

  2, 16, 16, 16, 16,
  4, 4,
  4, 8, 8, 8, 8, 8, 8,
  16, 8, 16, 8, 16, 8, 16, 8,
  16, 16, 16, 16, 16, 2
};

void setup()
{
  pinMode(BUZZER_PIN, OUTPUT);
}

void loop()
{
  int size = sizeof(durations) / sizeof(int);

  for (int note = 0; note < size; note++) {
    //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 duration = 1000 / durations[note];
    tone(BUZZER_PIN, melody[note], duration);

    //to distinguish the notes, set a minimum time between them.
    //the note's duration + 30% seems to work well:
    int pauseBetweenNotes = duration * 1.30;
    delay(pauseBetweenNotes);

    //stop the tone playing:
    noTone(BUZZER_PIN);
  }
}
```