

Frog Microcontroller Trainer (FMT): Piezo- Adding Duration

Length of Time: 40 Minutes

Recommended for Beginner

Contents

- Overview
- Objectives
- Standards
- Materials
- Lesson Elements
- Resources



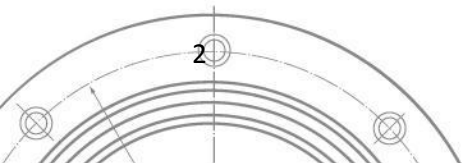
Overview:

- Students will be exploring the Piezo component of the Frog Microcontroller Trainer in this lesson.

Objectives:

Students will be able to:

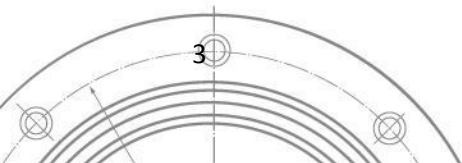
- *Use a sketch to create a constant sound with the Piezo*
- *Modify the sketch to send an S.O.S. signal using the sound created by the Piezo*





Computer Science Teacher Association Standards

- 1B-CS-02- Model how computer hardware and software work together as a system to accomplish tasks.
- 1B-AP-12- Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.
- 2-AP-16- Incorporate existing code, media, and libraries into original programs, and give attribution.
- 2-IC-21- Discuss issues of bias and accessibility in the design of existing technologies.



Materials:

- [Frog Microcontroller Trainer](#) from 1st Maker Space
- USB power cord
- Arduino IDE
- PC or Mac (Chromebooks if using Arduino Create for Education App)
- Introduction to Microcontroller Trainer Lesson (if needed as a resource)

Preparation:

- The teacher should ensure that students all have a compatible device and Arduino IDE installed prior to this lesson.

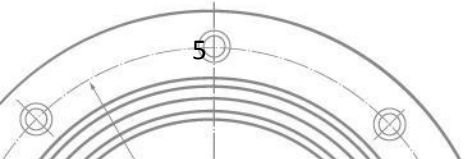
Background Information:

The Piezo buzzer is a speaker that uses the piezoelectric effect for generating sound. The mechanical motion is created by applying a voltage to a piezoelectric material, and this motion is typically converted into audible sound using diaphragms and resonators.

Lesson Elements													
Basic Part Identification– 5 minutes	Have students identify the Piezo on the Frog Microcontroller Trainer.												
Access 1MS Code – 10 minutes	Visit the 1st Maker Space website and access the library of code developed specifically for the Frog Microcontroller Trainer. Here, you will find a section with various code written for each major component of the trainer. The teacher will locate the sketch by clicking on 4.00- Using the Piezo. Explore our collection of Arduino projects tailored for the 1st Maker Space Microcontroller Trainer. Each project provides step-by-step guidance and includes ready-to-use code for direct input into the Arduino IDE. From basic LED tasks to advanced sensor activities, enhance your programming skills and harness the full potential of the 1MS Trainer with our interactive hub. <table><tr><td>Project 1.00 Blink In this project, you'll learn how to blink an LED! Read More →</td><td>Project 1.01 Blink x2 In this project, you'll learn how to blink more than one LED! Read More →</td><td>Project 1.02 Simple LED Chase In this project, you'll use all four LEDs to blink a chase pattern! Read More →</td><td>Project 1.03 Analog Write In this project, you will learn about a new function called analogWrite! Read More →</td></tr><tr><td>Project 1.04 Pulse LED In this project you'll learn how to use the analogWrite function to create a "pulsing" effect on an LED! Read More →</td><td>Project 2.00 Serial Printing In this project, you'll learn how to utilize the Arduino's Serial communication capabilities to print text data to your computer's Serial Monitor! Read More →</td><td>Project 2.01 Talking to the Board In this project, you'll learn how to create a two-way communication channel between your Arduino board and the Serial Monitor on your computer! Read More →</td><td>Project 3.00 Read Input In this project, you'll learn how to read the state of a digital input pin on the Arduino, specifically from a button connected to pin 1, and then print that state to the Serial Monitor! Read More →</td></tr><tr><td>Project 3.01 Blink an LED with a button In this project, you'll learn how to control an LED with a button! Read More →</td><td>Project 3.02 AND Logic In this project, you'll learn how to use logical AND (&&) to control an LED with two buttons! Read More →</td><td>Project 3.03 OR Logic In this project, you'll learn how to use logical OR () to control an LED with either button! Read More →</td><td>Project 4.00 Using the Piezo In this project, you'll learn how to generate a simple sound using a Piezo buzzer! Read More →</td></tr></table> Here you will find the sketch to share with students. Students can copy the code by simply clicking on the blue copy button.	Project 1.00 Blink In this project, you'll learn how to blink an LED! Read More →	Project 1.01 Blink x2 In this project, you'll learn how to blink more than one LED! Read More →	Project 1.02 Simple LED Chase In this project, you'll use all four LEDs to blink a chase pattern! Read More →	Project 1.03 Analog Write In this project, you will learn about a new function called analogWrite! Read More →	Project 1.04 Pulse LED In this project you'll learn how to use the analogWrite function to create a "pulsing" effect on an LED! Read More →	Project 2.00 Serial Printing In this project, you'll learn how to utilize the Arduino's Serial communication capabilities to print text data to your computer's Serial Monitor! Read More →	Project 2.01 Talking to the Board In this project, you'll learn how to create a two-way communication channel between your Arduino board and the Serial Monitor on your computer! Read More →	Project 3.00 Read Input In this project, you'll learn how to read the state of a digital input pin on the Arduino, specifically from a button connected to pin 1, and then print that state to the Serial Monitor! Read More →	Project 3.01 Blink an LED with a button In this project, you'll learn how to control an LED with a button! Read More →	Project 3.02 AND Logic In this project, you'll learn how to use logical AND (&&) to control an LED with two buttons! Read More →	Project 3.03 OR Logic In this project, you'll learn how to use logical OR () to control an LED with either button! Read More →	Project 4.00 Using the Piezo In this project, you'll learn how to generate a simple sound using a Piezo buzzer! Read More →
Project 1.00 Blink In this project, you'll learn how to blink an LED! Read More →	Project 1.01 Blink x2 In this project, you'll learn how to blink more than one LED! Read More →	Project 1.02 Simple LED Chase In this project, you'll use all four LEDs to blink a chase pattern! Read More →	Project 1.03 Analog Write In this project, you will learn about a new function called analogWrite! Read More →										
Project 1.04 Pulse LED In this project you'll learn how to use the analogWrite function to create a "pulsing" effect on an LED! Read More →	Project 2.00 Serial Printing In this project, you'll learn how to utilize the Arduino's Serial communication capabilities to print text data to your computer's Serial Monitor! Read More →	Project 2.01 Talking to the Board In this project, you'll learn how to create a two-way communication channel between your Arduino board and the Serial Monitor on your computer! Read More →	Project 3.00 Read Input In this project, you'll learn how to read the state of a digital input pin on the Arduino, specifically from a button connected to pin 1, and then print that state to the Serial Monitor! Read More →										
Project 3.01 Blink an LED with a button In this project, you'll learn how to control an LED with a button! Read More →	Project 3.02 AND Logic In this project, you'll learn how to use logical AND (&&) to control an LED with two buttons! Read More →	Project 3.03 OR Logic In this project, you'll learn how to use logical OR () to control an LED with either button! Read More →	Project 4.00 Using the Piezo In this project, you'll learn how to generate a simple sound using a Piezo buzzer! Read More →										



	Project Code: Copy <pre>//////////////////////////////////// // 4.00 - Using the Piezo byte piezoPin = 12; byte SW1 = 1; bool pressed = LOW; void setup() { pinMode(piezoPin, OUTPUT); pinMode(SW1, INPUT); } void loop() { if (digitalRead(SW1) == pressed) { digitalWrite(piezoPin, HIGH); delay(1); digitalWrite(piezoPin, LOW); delay(1); } } ////////////////////////////////////</pre> If you're copying and pasting the code, or typing from scratch, delete everything out of a new Arduino sketch and paste / type in the above text.
Connect the Trainer and Verify the Sketch – 15 minutes	Plug the Frog Microcontroller Trainer into your computer's USB port. Using the dropdown box near the top, select the Arduino Uno R4 Minima and appropriate COM port. Note: Mac computers may automatically select the port.



sketch_jan20a | Arduino IDE 2.3.7

File Edit Sketch Tools Help

✓ → ⚙

Select Board

sketch_jan20a

1 v

2

3

4 }

5

6 v

7

8 }

9 }

10

Arduino UNO R4 Minima
COM13

Unknown
COM6

Unknown
COM4

Unknown
COM7

Unknown
COM19

Unknown
COM8

Unknown
COM10

Unknown

Select other board and port...

to run once:

to run repeatedly:

Making sure the USB cable is connecting the FMT to the computer, verify the sketch by clicking on the check mark in the upper left-hand corner.

```
16
17 byte SW1 = 1;
18
19 bool pressed = LOW;
20
21 byte piezoPin = 12;
22
23 void setup() {
24   // The piezo is being written to.
25   pinMode(piezoPin, OUTPUT);
26   pinMode(SW1, INPUT);
27 }
28
29 void loop() {
30   if (digitalRead(SW1) == pressed) {
31     //           Hz   ms
32     BuzzPiezo(5, 1000);
33     BuzzPiezo(50, 1000);
34     BuzzPiezo(500, 1000);
35   }
36 }
37
```

Once verified, upload the sketch to the trainer by clicking the right-facing arrow. It will turn yellow.

```
16
17 byte SW1 = 1;
18
19 bool pressed = LOW;
20
21 byte piezoPin = 12;
22
23 void setup() {
24   // The piezo is being written to.
25   pinMode(piezoPin, OUTPUT);
26   pinMode(SW1, INPUT);
27 }
28
29 void loop() {
30   if (digitalRead(SW1) == pressed) {
31     //           Hz   ms
32     BuzzPiezo(5, 1000);
33     BuzzPiezo(50, 1000);
34     BuzzPiezo(500, 1000);
35   }
36 }
37
```

	If the code loaded correctly, your FMT should now be making a series of low, middle and high tones when pressing SW1.
Modify the Sketch – 25 minutes	Once students have successfully gotten their trainer to make the tones when SW1 is pressed, have students modify the sketch to send an S.O.S signal. (... - - - ...) Sample sketch below:  <pre>3_Adding_Duration_sos.ino 18 byte piezoPin = 12; 19 20 byte SW1 = 1; 21 bool pressed = LOW; 22 23 int shortBeep = 180; 24 int longBeep = 360; 25 int waitBeep = 200; 26 int waitLetter = 500; 27 int beepFreq = 500; 28 29 void setup() { 30 // The piezo is being written to. 31 pinMode(piezoPin, OUTPUT); 32 } 33 34 void loop() { 35 // Hz ms 36 37 while (digitalRead(SW1) != pressed) { 38 ; //Do Nothing 39 } 40 //S 41 BuzzPiezo(beepFreq, shortBeep); 42 delay(waitBeep); 43 BuzzPiezo(beepFreq, shortBeep); 44 delay(waitBeep); 45 BuzzPiezo(beepFreq, shortBeep); 46 delay(waitLetter); 47 48 //O 49 BuzzPiezo(beepFreq, longBeep); 50 delay(waitBeep); 51 BuzzPiezo(beepFreq, longBeep); 52 delay(waitBeep); 53 BuzzPiezo(beepFreq, longBeep); 54 delay(waitLetter); 55 56 //S 57 BuzzPiezo(beepFreq, shortBeep); 58 delay(waitBeep); 59 BuzzPiezo(beepFreq, shortBeep); 60 delay(waitBeep); 61 BuzzPiezo(beepFreq, shortBeep); 62 delay(waitLetter); 63 }</pre>

```

64 }
65
66 /*
67  This function will generate a certain tone for a certain time
68  | given a frequency in Hz and a duration in ms.
69  */
70
71 void BuzzPiezo(long frequency, long duration) {
72     long period = 1000 / frequency;
73     long cycles = duration / period;
74
75     for (long i = 0; i < cycles; i++) {
76         digitalWrite(piezoPin, HIGH);
77         delay(period / 2);
78         digitalWrite(piezoPin, LOW);
79         delay(period / 2);
80     }
81 }
82 //////////////////////////////////////////////////
83

```

Career Exploration:

- Students can choose from a list of careers related to soldering to learn more about. A good starting place is the [IDOE Career Explorer database](#).
- Another good resource for career information is the bureau of labor statistics

Practical Application:

Piezoelectric speakers are frequently used to generate sound in digital quartz watches and other electronic devices and are sometimes used as tweeters in less-expensive speaker systems, such as computer speakers and portable radios. They are also used for producing ultrasound in sonar systems.

Additional Resources:

- 1st Maker Space Frog Microcontroller Library
- 1st Maker Space Learn Arduino with the Frog Microcontroller

Performance Assessment/Check for Understanding :

- Was the student able to get their trainer to make a continuous sound?
- Was the student able to modify the sketch to make an S.O.S signal?

Additional Feedback: