

# Fire Engine

Time required: 45 minutes

Please read all the directions carefully before beginning the assignment.

- Comment your code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

---

## Understanding

Demonstrate understanding of:

**variables, constants, for and while loops**

---

## Knowledge Points

This program simulates the sound of a fire engine siren.

---

## Fire Engine Sire Simulation

The fire engine siren's low frequency sound is 500Hz, its high frequency sound is between 1500Hz. The siren sound is generated by repeating the following pattern: the low frequency sound amplifies to a high frequency sound in 1.5 seconds, and then drops back to the lower frequency in 3.5 seconds. Therefore, the fire engine siren sounds can be programmatically simulated as follows:

Set the low frequency to be 500Hz, then set the high frequency sound to be 1500HZ, repeatedly playing the buzzer in a range from 500Hz to 1500Hz and then back to 700Hz. The ratio of amplification time to the drop time is 1.5:3.5, which is 3:7, so the ratio of frequency amplification to the drop needs to be 7:3. By tuning the sound time and amplification vs. drop's amplitude, the fire engine siren is simulated.

### Why “repeat until frequency > 1500”, not “frequency = 1500”?

That is because in the example of simulating the fire engine sound effects, it is hard to define the sound frequency each time when it is increased from 500Hz to 1500Hz and what increment it should be each time the frequency is increased. If we set frequency=1500, the final frequency should reach 1500 so that it can break the loop, or the frequency will be increased again and again, making it hard to debug. So we use frequency > 1500, and

when the frequency is above 1500, the loop will be broken and the following program decreasing the frequency will be executed.

---

## Requirements

The program runs as shown.

---

## Tutorial Assignment

1. Start the Arduino IDE. Save the program as **FireEngine**
2. Complete and test the program as pictured with the requirements listed.

```

1 /**
2   @file    FireEngine.ino
3   @author  William A Loring
4   @version v1.0.0
5   @Revised 05/17/18   Created: 06/18/2021
6   @Description: Play fire engine sounds and move forward
7   */
8   // ***** DON'T CHANGE CODE BELOW ***** //
9   #include <MeMCore.h> // Include mBot library
10  MeRGBLed led(0, 30); // Create LED object
11  MeIR ir; // Setup IR remote object
12  MeBuzzer buzzer; // Setup buzzer object
13  MeDCMotor MotorL(M1); // MotorL is Left Motor
14  MeDCMotor MotorR(M2); // MotorR is Right
15  // Low frequency of fire engine
16  const int LOW_FREQUENCY = 500;
17  // High frequency of fire engine
18  const int HIGH_FREQUENCY = 1500;
19
20 void setup() {
21   led.setpin(13);
22   // Start listening to the ir remote
23   ir.begin();
24 }
25 // ***** DON'T CHANGE CODE ABOVE ***** //
26
27 void loop() {
28   // loop until remote button is pressed
29   if (ir.keyPressed(IR_BUTTON_UP)) {
30     fireEngine();
31   } else if (ir.keyPressed(IR_BUTTON_DOWN)) {
32     // Uncomment the following line
33     // when you complete the function
34     // policeCar();
35   }
36 }

```

3.

```

38 void fireEngine() {
39     MotorL.run(-127); // MotorL (Left) forward is -negative
40     MotorR.run(127);  // MotorR (Right) forward is +positive
41
42     // Play siren while loops 2x's
43     for (int i = 0; i < 2; i++) {
44         // Set frequency to low frequency
45         int frequency = LOW_FREQUENCY;
46
47         while (frequency < HIGH_FREQUENCY) {
48             buzzer.tone(frequency, 13);
49             frequency = frequency + 35;
50         }
51
52         frequency = HIGH_FREQUENCY;
53         while (frequency > LOW_FREQUENCY) {
54             buzzer.tone(frequency, 13);
55             frequency = frequency - 15;
56         }
57     }
58     MotorL.stop(); // Stop MotorL
59     MotorR.stop(); // Stop MotorR
60 }

```

4.

---

## Assignment

Start with your tutorial project and add the following.

Keep the fire siren. Add a police car sound or a different siren sound. The tutorial shows a policeCar function call which you can complete.

How to simulate police car sound effects: Low-frequency sound is set at 500Hz and high-frequency sound at 1500Hz. It takes 23 ms to raise a low-frequency sound to a high-frequency one and 23 ms to lower a high-frequency sound to a low-frequency one.

---

## Assignment Submission

- **All students** → Attach finished programs to the assignment in Blackboard.
- **In class assignment submission** → Demonstrate in person.

- **Online submission** → A link to a YouTube video recording showing the assignment placed in the submission area in BlackBoard.