



CSLR61: EMBEDDED SYSTEMS LABs

Report

106119100 - Rajneesh Pandey



Date : 31/01/2022

CSLR61 : EMBEDDED SYSTEMS LAB-1

Roll no. : **106119100**

Name : **Rajneesh Pandey**

Section : **CSE-B**

1. Blink LEDs in alternate order – 1 and 3 together and 2 and 4 together.

```
/* 106119100 Rajneesh Pandey */
#include "mbed.h"

DigitalOut led1(LED1);
DigitalOut led2(LED2);
DigitalOut led3(LED3);
DigitalOut led4(LED4);

int main(){
    while (1){
        led1 = !led1;
        printf("Blink! LED1 is now %d\n", led1.read());
        led3 = !led3;
        printf("Blink! LED3 is now %d\n", led3.read());
        wait_ms(1000);
        led2 = !led2;
        printf("Blink! LED2 is now %d\n", led2.read());
        led4 = !led4;
        printf("Blink! LED4 is now %d\n", led4.read());
    }
}
```

arm MBED

Blinky Load demo Run + Add component

```

1  /* 106119100 Rajneesh Pandey*/
2
3 #include "mbed.h"
4
5 DigitalOut led1(LED1);
6 DigitalOut led2(LED2);
7 DigitalOut led3(LED3);
8 DigitalOut led4(LED4);
9
10 int main() {
11     while (1) {
12         led1 = !led1;
13         printf("Blink! LED1 is now %d\n", led1.read());
14         led3 = !led3;
15         printf("Blink! LED3 is now %d\n", led3.read());
16
17         wait_ms(1000);
18
19         led2 = !led2;
20         printf("Blink! LED2 is now %d\n", led2.read());
21         led4 = !led4;
22         printf("Blink! LED4 is now %d\n", led4.read());
23
24     }
25 }
26
27 }
```

Serial output

```

Blink! LED4 is now 0
Blink! LED1 is now 1
Blink! LED3 is now 1
Blink! LED2 is now 1
Blink! LED4 is now 1
Blink! LED1 is now 0
Blink! LED3 is now 0
Blink! LED2 is now 0
Blink! LED4 is now 0
Blink! LED1 is now 1
Blink! LED3 is now 1
Blink! LED2 is now 1
Blink! LED4 is now 1
Blink! LED1 is now 0

```

2. Blink LEDs – count from 1 to 15; if the board is counting odd value, wait for 1 sec, else wait for 2 sec.

```

3. /* 106119100 Rajneesh Pandey*/
4. #include "mbed.h"
5.
6. BusOut myleds(LED1, LED2, LED3, LED4);
7.
8. int main(){
9.     while (1){
10.         for (int i = 1; i < 16; i++){
11.             myleds = i;
12.             printf("%d ", i);
13.             if (i & 1){
14.                 printf("odd count\n");
15.                 wait(1);
16.             }
17.             else{
18.                 printf("even count\n");
19.                 wait(2);
20.             }
21.         }
22.     }
23. }
```

24.

arm MBED

The screenshot shows the arm MBED IDE interface. On the left is the code editor with the following C code:

```
1 /* 106119100 Rajneesh Pandey*/
2 #include "mbed.h"
3
4 BusOut myleds(LED1, LED2, LED3, LED4);
5
6 int main() {
7     while(1) {
8         for(int i = 1; i < 16; i++) {
9             myleds = i;
10            printf("%d ",i);
11            if(i&1){
12                printf("odd count\n");
13                wait(1);
14            }
15            else{
16                printf("even count\n");
17                wait(2);
18            }
19        }
20    }
21 }
```

In the center is a pinout diagram for a microcontroller board, showing various pins labeled GND, V_{IN}, V_{DD}, SCK, MISO, MOSI, SS, CS, SDA, SCL, I₂C, SPI, Serial, Analog In, and PWM. A red box highlights the Serial port. On the right is the serial output window displaying the following text:

```
Serial output
1 odd count
2 even count
3 odd count
4 even count
5 odd count
6 even count
7 odd count
8 even count
9 odd count
10 even count
11 odd count
12 even count
13 odd count
14 even count
```

3. Blink LEDs – for all composite number below 15.

```
4./* 106119100 Rajneesh Pandey */
5.#include "mbed.h"
6.
7.BusOut myleds(LED1, LED2, LED3, LED4);
8.
9.bool isComposite(int n)
10.{
11.    if (n == 1)
12.        return true;
13.    for (int i = 2; i < n; i++)
14.    {
15.        if (n % i == 0)
16.            return true;
17.    }
18.    return false;
19.}
20.int main()
21.{
```

```

22.     while (1)
23.     {
24.         for (int i = 1; i < 16; i++)
25.         {
26.             if (isComposite(i))
27.             {
28.                 myleds = i;
29.                 printf("%d is Composite Number\n", i);
30.                 wait(1);
31.             }
32.         }
33.     }
34. }
35.

```

arm MBED

Blinky with BusOut + Add component

Run

```

1 #include "mbed.h"
2
3 BusOut myleds(LED1, LED2, LED3, LED4);
4
5 bool isComposite(int n){
6     if(n==1) return true;
7     for(int i=2;i<n;i++){
8         if(n%i==0) return true;
9     }
10    return false;
11 }
12 int main() {
13     while(1) {
14         for(int i = 1; i < 16; i++) {
15             if(isComposite(i)){
16                 myleds = i;
17                 printf("%d is Composite Number\n",i);
18             }
19         }
20     }
21 }
22 }
23

```

Serial output

```

1 is Composite Number
4 is Composite Number
6 is Composite Number
8 is Composite Number
9 is Composite Number
10 is Composite Number
12 is Composite Number
14 is Composite Number
15 is Composite Number
1 is Composite Number
4 is Composite Number
6 is Composite Number
8 is Composite Number
9 is Composite Number

```

4. Blink LEDs – to count even numbers

```
5. #include "mbed.h"
6.
7. BusOut leds(LED1, LED2, LED3, LED4);
8.
9. int main(){
10.     while (1){
11.         for(int i = 0; i < 16; i++){
12.             if(i%2==0){
13.                 leds = i;
14.                 printf("count %d\n", leds.read());
15.                 wait(2);
16.             }
17.         }
18.     }
19. }
```

arm MBED

Blinky

Load demo

Run

+ Add component



```
1 #include "mbed.h"
2
3 BusOut leds(LED1, LED2, LED3, LED4);
4
5 int main(){
6     while (1){
7         for(int i = 0; i < 16; i++){
8             if(i%2==0){
9                 leds = i;
10                printf("count %d\n", leds.read());
11                wait(2);
12            }
13        }
14    }
15 }
```



Serial output

```
count 0
count 2
count 4
count 6
count 8
count 10
count 12
count 14
count 0
count 2
```

Date : 07/02/2022

CSLR61 : EMBEDDED SYSTEMS LAB-2

Roll no. : **106119100**

Name : **Rajneesh Pandey**

Section : **CSE-B**

1. Switch between hex counter and decade counter using switches. Display the current value of the counter with the help of 4 LEDs and the screen.
 - Libraries Used: InterruptIn, BusOut

```
#include "mbed.h"
BusOut leds(LED1, LED2, LED3, LED4);
InterruptIn swt(p5);

void hexCounter()
{
    while (1)
    {
        for (int i = 0; i < 16; i++)
        {
            leds = i;
            printf("Hex count : %d\n", i);
            wait(1);
        }
    }
}

void decCounter()
{
    while (1)
    {
        for (int i = 0; i < 10; i++)
        {
            leds = i;
            printf("Dec count : %d\n", i);
            wait(1);
        }
    }
}

int main()
{
    swt.rise(&hexCounter);
    swt.fall(&decCounter);
}
```

Output :

simulator.mbed.com/#user_1644227344419

arm MBED

Blinky with BusOut + Add component

```
1 #include "mbed.h"
2
3 BusOut leds(LED1, LED2, LED3, LED4);
4 InterruptIn swt(p5);
5
6 void hexCounter(){
7     while(1){
8         for(int i = 0; i < 16; i++) {
9             leds = i;
10            printf("Hex count : %d\n",i);
11            wait(1);
12        }
13    }
14 }
15 void decCounter(){
16     while(1){
17         for(int i = 0; i < 10; i++) {
18             leds = i;
19             printf("Dec count : %d\n",i);
20             wait(1);
21         }
22     }
23 }
24 int main() {
25     swt.rise(&hexCounter);
26     swt.fall(&decCounter);
27 }
```

Run

Switch (p5)

Serial output

```
Hex count : 1
Hex count : 2
Hex count : 3
Dec count : 0
Dec count : 1
Hex count : 0
Hex count : 1
Hex count : 2
Hex count : 3
```

2. Display the octal counter using the onboard LEDs using Ticker Object.
After 10 second, reset all the values of the onboard LED using Timeout Object.

- Libraries Used: TimeOut, Ticker

```
#include "mbed.h"

BusOut leds(LED1, LED2, LED3, LED4);
Ticker tck;
Timeout to;

int octCount = 0;

void octalCounter()
{
    leds = octCount;
    printf("counter %d\n", leds.read());
    octCount = (octCount + 1) % 8;
}

void reset()
{
    printf("resetting...");
    octCount = 0;
    leds = 0;
    printf("Value after resetting %d\n", leds.read());
}
```

```

int main()
{
    tck.attach(&octalCounter, 2);
    to.attach(&reset, 10);
}

```

simulator.mbed.com/#user_1644228553682

arm MBED

Blinky + Add component

Load demo Run

Serial output

```

counter 0
counter 1
counter 2
counter 3
counter 4
resetting...Value after resetting  0
counter 0
counter 1
counter 2
counter 3

```

3. Display the odd counter using the onboard LEDs and screen; After 10 seconds, reset all the values of the onboard LED and display even counter using Timeout Object. Give switch option to manually toggle between two counters.

Libraries Used: InterruptIn, Ticker, TimeOut, BusOut

```
#include "mbed.h"

BusOut leds(LED1, LED2, LED3, LED4);
Ticker tck;
Timeout to;
InterruptIn swt(p5);

int state = 1;
int odd = 1;

void Counter(){
    while(1){
        for(int i = 0; i < 16; i++){
            if (i % 2 == state){
                leds = i;
                if (state)
                    printf("Odd Counter %d\n", leds.read());
                else{
                    printf("Even Counter %d\n", leds.read());
                }
                wait(1);
            }
        }
    }
}
```

```

void Toggle(){
    state = !state;
    printf("Toggle using switch...\n");
    Counter();
}

void reset(){
    printf("resetting Odd Counter...\n");
    state = 0;
    leds = 0;
    tck.detach();
    printf("Starting Even Counter...\n");
    Counter();
}
void oddCounter(){
    leds = odd;
    odd = (odd+2)%16;
    printf("Odd Counter %d\n", leds.read());
}

int main()
{
    tck.attach(&oddCounter, 2.0);
    to.attach(&reset, 10);
    swt.rise(&Toggle);
}

```

simulator.mbed.com/#user_1644231511600

arm MBED

Blinky Load demo Run + Add component

```

1 #include "mbed.h"
2
3 BusOut leds(LED1, LED2, LED3, LED4);
4 Ticker tck;
5 Timeout to;
6 InterruptIn swt(p5);
7
8 int state = 1;
9 int odd = 1;
10
11 void Counter(){
12     while(1){
13         for(int i = 0; i < 16; i++){
14             if (i % 2 == state){
15                 leds = i;
16                 if (state)
17                     printf("Odd Counter %d\n", leds.read());
18                 else{
19                     printf("Even Counter %d\n", leds.read());
20                 }
21                 wait(1);
22             }
23         }
24     }
25 }
26
27 void Toggle(){
28     state = !state;
29     printf("Toggle using switch...\n");
30     Counter();
31 }
32
33 void reset(){
34     printf("resetting Odd Counter...\n");
35     state = 0;

```

The schematic view shows a digital breadboard with various components connected. A Push Button (p5) is connected to pin 21. A Serial port is connected between pins 10 and 11. A SPI port is connected between pins 12 and 13. An Analog In port is connected to pin 14. A 3.3V voltage source is connected to pin 15. A ground connection is present at pin 16.

Push Button (p5)

Serial output

```

Odd Counter 9
resetting Odd Counter...
Starting Even Counter...
Even Counter 0
Even Counter 2
Even Counter 4
Even Counter 6
Even Counter 8
Even Counter 10
Even Counter 12
Even Counter 14
Even Counter 0
Toggle using switch...
Odd Counter 1

```

Date : 14/02/2022

CSLR61 : EMBEDDED SYSTEMS LAB-3

Roll no. : **106119100**

Name : **Rajneesh Pandey**

Section : **CSE-B**

1. Increase the intensity of one LED while decreasing the other LED's intensity at the same unit. Display the intensity value of the LEDs.
 - Libraries Used: pwmout

Code :

```
#include "mbed.h"

PwmOut led1(p5);
PwmOut led2(p6);

int main(){
    led1 = 0.0;
    led2 = 1.0;
    while (true)
    {

        led1 = led1+0.1;
        led2 = led2-0.1;
        printf("LED1 is now %.2f\n", led1.read());
        printf("LED2 is now %.2f\n", led2.read());
        wait(1);
        if (led1 == 1.0)
            led1 = 0.0;
        if (led2 == 0.0)
            led2 = 1.0;
    }
}
```

Output :

The screenshot shows the arm MBED Mbed Simulator interface. On the left, there is a code editor window titled "Blinky" containing C code for a PWM LED blink example. In the center, there is a schematic diagram of a microcontroller board with various pins and components. Two blue LED components are connected to pins p5 and p6. On the right, there is a terminal window titled "Serial output" displaying the serial console output.

```
1 #include "mbed.h"
2
3 PwmOut led1(p5);
4 PwmOut led2(p6);
5
6 int main() {
7
8     led1 = 0;
9     led2 = 1;
10
11    while(1) {
12
13        led1 = led1 + 0.10;
14        led2 = led2 - 0.10;
15
16        printf("LED1 is now %.2f\n", led1.read());
17        printf("LED2 is now %.2f\n\n", led2.read());
18
19        wait(1);
20
21        if(led1==1.0)
22            led1 = 0;
23
24        if(led2 == 0.0)
25            led2 = 1;
26
27    }
28 }
```

Serial output

```
LED2 is now 0.50
LED1 is now 0.40
LED2 is now 0.40
LED1 is now 0.50
LED2 is now 0.30
LED1 is now 0.60
LED2 is now 0.20
LED1 is now 0.70
LED2 is now 0.09
```

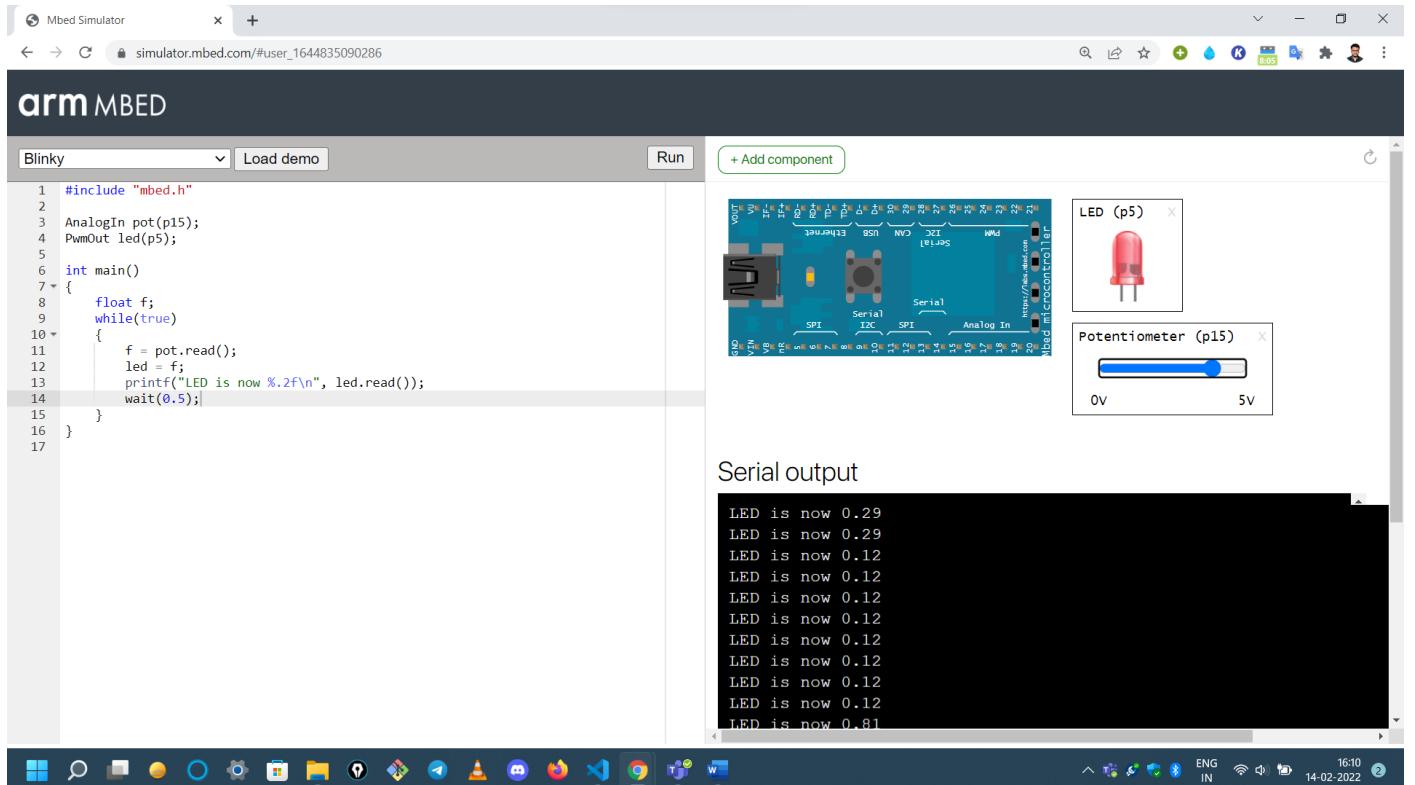
2. Interface potentiometer with mbed board and based on the value of potentiometer, adjust the intensity of an external LED.
- Libraries To Be Used: pwmout, analogin

Code:

```
#include "mbed.h"

AnalogIn pot(p15);
PwmOut led(p5);

int main(){
    float f;
    while (true)
    {
        f = pot.read();
        led = f;
        printf("LED is now %.2f\n", led.read());
        wait(0.5);
    }
}
```



3. Interface switch, 2 LEDs, potentiometer with the mbed board. If the switch is on, control the intensity of the LEDs using potentiometer (one LED should be increasing and other should be decreasing at the rate given via potentiometer). If the switch is off, blink LEDs one at increasing rate and other at decreasing rate (same rate) and loop back.

Code:

```
#include "mbed.h"

AnalogIn pot(p15);
PwmOut led1(p5);
PwmOut led2(p6);

InterruptIn swt(p7);

void on()
{
    float f;
    while(true){
        f = pot.read();
        led1 = f;
        led2 = 1-led1;
        printf("LED1 is now %.2f\n", led1.read());
        printf("LED2 is now %.2f\n", led2.read());
        wait(1);
        if (led1==1.0)
            led1 = 0.0;

        if (led2 == 0.0)
            led2 = 1.0;
    }
}

void off(){
    led1 = 0.0;
    led2 = 1.0;
    while (true)
    {
```

```

        led1 = led1+0.1;
        led2 = led2-0.1;
        printf("LED1 is now %.2f\n", led1.read());
        printf("LED2 is now %.2f\n", led2.read());
        wait(1);
        if (led1 == 1.0)
            led1 = 0.0;
        if (led2 == 0.0)
            led2 = 1.0;
    }
}

int main(){
    swt.rise(&on);
    swt.fall(&off);
}

```

Output :

The screenshot shows the arm MBED Mbed Simulator interface. On the left, the code for the 'Blinky' application is displayed. The hardware setup on the right shows a breadboard with two LEDs (p5 and p6) connected to pins 19 and 20 respectively, and a potentiometer (p15) connected to pin 1. The serial output window at the bottom shows the following text:

```

Serial output
LED2 is now 0.85
LED1 is now 0.15

```

Date : 28/02/2022

CSLR61 : EMBEDDED SYSTEMS LAB-4

Roll no. : **106119100**

Name : **Rajneesh Pandey**

Section : **CSE-B**

1. Write a program to increase the intensity of external LED. At the same rate, increase the sound intensity of speaker.

Libraries to be Used: pwmout

```
#include "mbed.h"

PwmOut led(p5);
PwmOut spker(p6);

int main()
{
    led = 0.0;
    spker = led;
    while (true)
    {

        led = led + 0.1;
        spker = led;
        printf("LED is now %.2f\n", led.read());
        printf("SPEAKER is now %.2f\n", spker.read());
        wait(1);
    }
}
```

Output :

The screenshot shows the Mbed Simulator interface with the following components:

- Code Area:** Displays the C code for the "Blink" example. The code initializes two PWM outputs, led(p5) and spker(p21), and enters a loop where it prints the current LED value and speaker volume to the serial port.
- Hardware Diagram:** Shows the STM32F103C8T6 microcontroller with various pins labeled. Components connected include an LED (p5) and a PWM Speaker (p21).
- Serial Output:** A text box showing the following serial data:

```
SPEAKER is now 0.50
LED is now 0.60
SPEAKER is now 0.60
LED is now 0.70
SPEAKER is now 0.70
LED is now 0.80
SPEAKER is now 0.80
LED is now 0.90
SPEAKER is now 0.90
LED is now 1.00
SPEAKER is now 1.00
```

2. Design an alarm system, which makes double beep sound for every 5s using a speaker.

Libraries To Be Used: pwmout, analogin

```
#include "mbed.h"

PwmOut speaker(p21);
Ticker tck;

void beepTwice()
{
    speaker = 0.0;
    speaker = 1.0;
    printf("Beep");
    wait(0.2);
    speaker = 0.0;
    wait(0.2);
    speaker = 1.0;
    printf("Beep\n");
    wait(0.2);
    speaker = 0.0;
}

int main()
{
    beepTwice();
    tck.attach(&beepTwice, 5);
}
```

Output :

The screenshot shows the Mbed Simulator interface. On the left, the code for a PWM speaker is displayed:

```
1 #include "mbed.h"
2
3 PwmOut speaker(p21);
4 Ticker tck;
5
6 void beepTwice()
7 {
8     speaker = 0.0;
9     speaker = 1.0;
10    printf("Beep");
11    wait(0.2);
12    speaker = 0.0;
13    wait(0.2);
14    speaker = 1.0;
15    printf("Beep\n");
16    wait(0.2);
17    speaker = 0.0;
18 }
19
20 int main()
21 {
22     beepTwice();
23     tck.attach(&beepTwice, 5);
24 }
```

The middle section shows the component library with a selected "Pwm Speaker (p21)" component. The right section shows the "Serial output" window displaying the text "BeepBeep" five times.

3. Have a switch, 2 LEDs and a speaker interfaced with mbed board. Upon switching on, the intensity of the sound should be increasing and upon switching off, should make the intensity of the sound decreasing. Proportionally increase or decrease the intensity of two LEDs.

```
#include "mbed.h"

PwmOut led1(p5);
PwmOut led2(p6);
PwmOut spker(p21);

InterruptIn swt(p7);

void on()
{
    while (true)
    {
        spker = spker + 0.1;
        led1 = spker;
        led2 = spker;
        printf("Speaker is now %.2f\n", spker.read());
        printf("Led1 is now %.2f\n", led1.read());
        printf("Led2 is now %.2f\n", led2.read());
        wait(2);
    }
}

void off()
{
    while (true)
    {
        spker = spker - 0.1;
        led1 = spker;
        led2 = spker;
        printf("Speaker is now %.2f\n", spker.read());
        printf("Led1 is now %.2f\n", led1.read());
        printf("Led2 is now %.2f\n", led2.read());
        wait(2);
    }
}
```

```
}
```

```
int main()
{
    spker = 0.5;
    swt.rise(&on);
    swt.fall(&off);
}
```

The screenshot shows the arm MBED online simulator interface. On the left, the code for the Blinky demo is displayed:

```
1 #include "mbed.h"
2
3 PwmOut led1(p5);
4 PwmOut led2(p6);
5 PwmOut spker(p21);
6
7 InterruptIn swt(p7);
8
9 void on()
10 {
11     while (true)
12     {
13         spker = spker + 0.1;
14         led1 = spker;
15         led2 = spker;
16         printf("Speaker is now %.2f\n", spker.read());
17         printf("Led1 is now %.2f\n", led1.read());
18         printf("Led2 is now %.2f\n", led2.read());
19         wait(2);
20     }
21 }
22 void off()
23 {
24     while (true)
25     {
26         spker = spker - 0.1;
27         led1 = spker;
28         led2 = spker;
29         printf("Speaker is now %.2f\n", spker.read());
30         printf("Led1 is now %.2f\n", led1.read());
31         printf("Led2 is now %.2f\n", led2.read());
32         wait(2);
33     }
34 }
35
36 int main()
37 {
38     spker = 0.5;
39     swt.rise(&on);
40     swt.fall(&off);
41 }
```

The right side of the interface shows the hardware setup. A breadboard diagram is at the top, followed by component placement: a Speaker (p21), LED (p5) (red), LED (p6) (blue), and a Switch (p7). Below the breadboard, the serial output window displays the following text:

Serial output

```
Led1 is now 0.60
Led2 is now 0.60
Speaker is now 0.50
Led1 is now 0.50
Led2 is now 0.50
Speaker is now 0.40
Led1 is now 0.40
Led2 is now 0.40
Speaker is now 0.50
Led1 is now 0.50
```

Date : 07/03/2022

CSLR61 : EMBEDDED SYSTEMS LAB-5

Roll no. : **106119100**

Name : **Rajneesh Pandey**

Section : **CSE-B**

1. Blink Led In Arduino UNO Simulator using Tinkercad

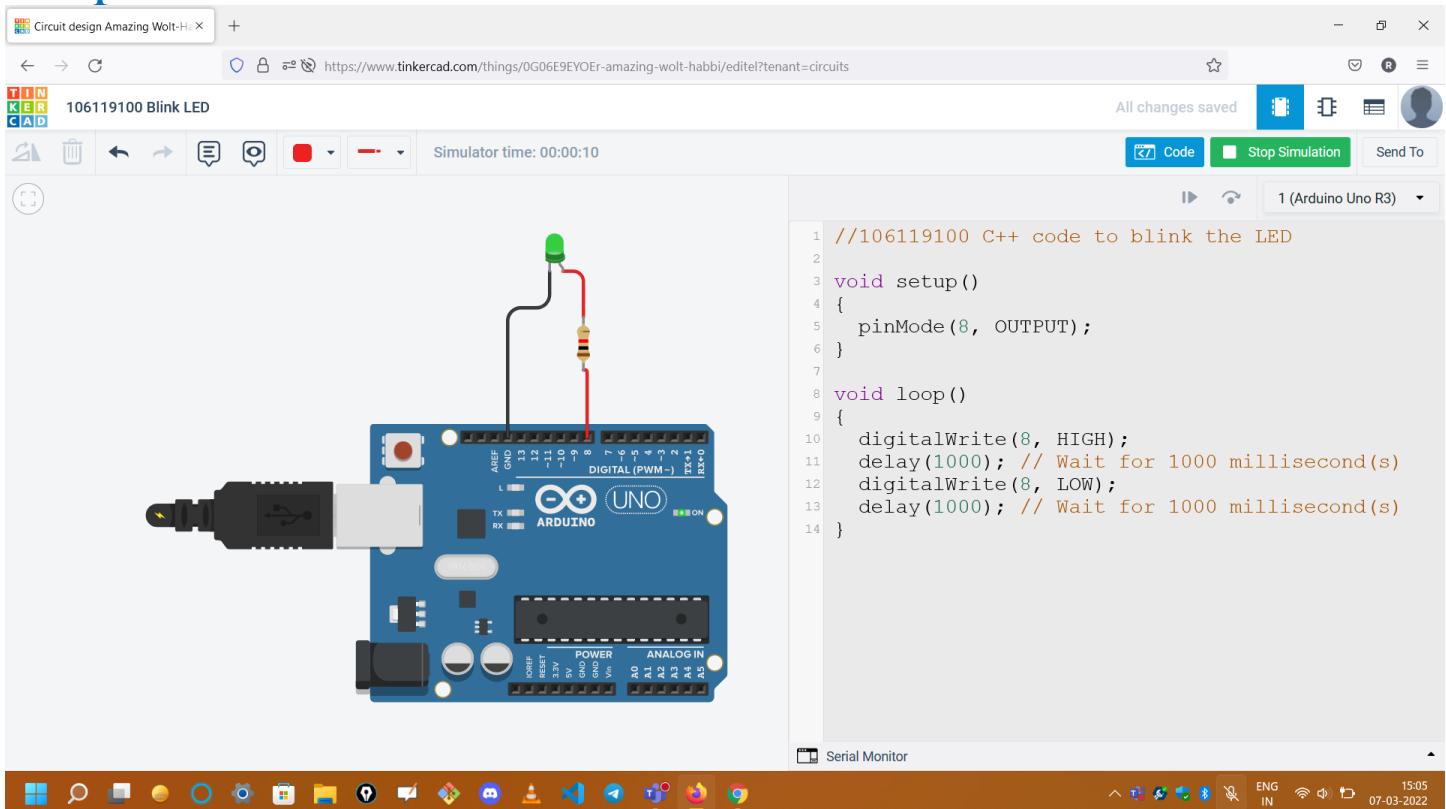
Code

```
// 106119100 C++ code to blink the LED

void setup(){
    pinMode(8, OUTPUT);
}

void loop(){
    digitalWrite(8, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(8, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
}
```

Output



2. Blink LEDs in alternate order – 1 and 3 together and 2 and 4 together.

Code:

```
// 106119100 C++ code to Blink Alternative LED
void setup(){
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
}
void loop(){
    digitalWrite(5, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, HIGH);
    digitalWrite(6, HIGH);
    delay(2000);
    digitalWrite(8, LOW);
    digitalWrite(6, LOW);
    digitalWrite(5, HIGH);
    digitalWrite(7, HIGH);
    delay(2000);
}
```

Output:

Circuit design Copy of 106119100

https://www.tinkercad.com/things/dMFjWHFTl6m-copy-of-106119100-wait-for-1-and-2-sec-for-odd-and-even-count-/edit#circuits

All changes saved

Code Stop Simulation Send To

1 (Arduino Uno R3)

Simulator time: 00:00:07

1 // 106119100 C++ code to Blink Alternative LED
2 void setup(){
3 pinMode(5, OUTPUT);
4 pinMode(6, OUTPUT);
5 pinMode(7, OUTPUT);
6 pinMode(8, OUTPUT);
7 }
8 }
9 void loop(){
10 digitalWrite(5, LOW);
11 digitalWrite(7, LOW);
12 digitalWrite(8, HIGH);
13 digitalWrite(6, HIGH);
14 delay(2000);
15 digitalWrite(8, LOW);
16 digitalWrite(6, LOW);
17 digitalWrite(5, HIGH);
18 digitalWrite(7, HIGH);
19 delay(2000);
20 }
21 }
22 }

Serial Monitor

16:01
07-03-2022

3. Blink LEDs – count from 1 to 15; if the board is counting odd value, wait for 1 sec, else wait for 2 sec.

Code:

```
// 106119100 C++ code to Wait for 1 and 2 sec for
// odd and even count
void setup()
{
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
}

void displayNumber(int n)
{
    digitalWrite(5, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(6, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(7, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(8, (n & 1) ? HIGH : LOW);
}

void loop()
{
    for (int i = 1; i <= 15; i++)
    {
        displayNumber(i);
        if (i & 1)
            delay(1000);
        else
            delay(2000);
    }
}
```

Output:

The screenshot shows a Tinkercad simulation for an Arduino Uno R3. The circuit consists of an Arduino Uno board connected to four red LEDs. The LEDs are connected in parallel, with their common ground connection tied to GND on the Arduino. The individual LED cathodes are connected to pins 5, 6, 7, and 8 respectively. The Arduino Uno board is shown with its standard pinout, including AREF, GND, 3.3V, 5V, POWER, GND, Vin, and ANALOG IN/A0-A5. A USB cable is connected to the Arduino. The code in the editor is as follows:

```
// 106119100 C++ code to Wait for 1 and 2 sec for
// odd and even count
void setup(){
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
}
void displayNumber(int n){
  digitalWrite(5, (n&1) ? HIGH : LOW);
  n >>= 1;
  digitalWrite(6, (n&1) ? HIGH : LOW);
  n >>= 1;
  digitalWrite(7, (n&1) ? HIGH : LOW);
  n >>= 1;
  digitalWrite(8, (n&1) ? HIGH : LOW);
}
void loop(){
  for (int i = 1; i <= 15; i ++){
    displayNumber(i);
    if(i&1)
      delay(1000);
    else
      delay(2000);
  }
}
```

The serial monitor window is visible at the bottom, showing the status bar with "Serial Monitor", "15:49", "ENG IN", "07-03-2022", and a battery icon.

4. Blink LEDs – for all composite number below 15.

Code:

```
// 106119100 C++ code to count composite number
void setup()
{
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
}
void displayNumber(int n)
{
    digitalWrite(5, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(6, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(7, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(8, (n & 1) ? HIGH : LOW);
}
bool isPrime(int n)
{
    if (n == 0 || n == 1)
        return 0;
    if (n == 2)
        return 1;
    for (int i = 3; i <= n; i++)
    {
        for (int fac = 2; fac <= n - 1; fac++)
        {
            if (i % fac == 0)
                return 0;
        }
    }
}
```

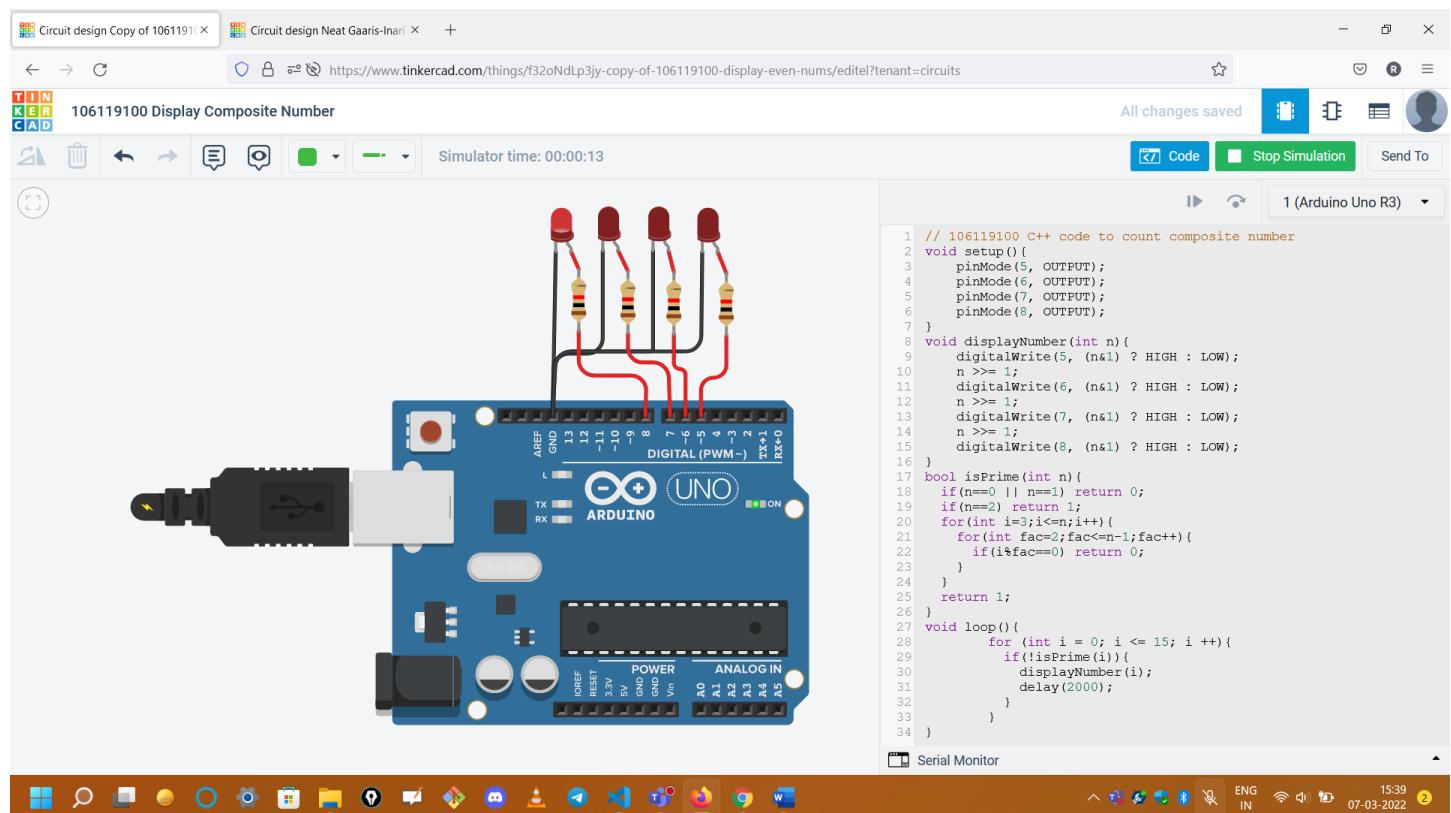
```

    return 1;
}

void loop()
{
    for (int i = 0; i <= 15; i++)
    {
        if (!isPrime(i))
        {
            displayNumber(i);
            delay(2000);
        }
    }
}

```

Output:



5. Blink LEDs – to count even numbers

Code:

```
// 106119100 C++ code to blink the LED

void setup()
{
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);
}

void displayNumber(int n)
{
    digitalWrite(5, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(6, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(7, (n & 1) ? HIGH : LOW);
    n >>= 1;
    digitalWrite(8, (n & 1) ? HIGH : LOW);
}

void loop()
{
    for (int i = 0; i <= 15; i += 2)
    {
        displayNumber(i);
        delay(2000);
    }
}
```

Output:

Circuit design Neat Gaaris-Inari

https://www.tinkercad.com/things/fyc0jYCuqv0-neat-gaarais-inari/edit#circuits

106119100 Display Even Nums

All changes saved

Simulator time: 00:00:11

Code Stop Simulation Send To 1 (Arduino Uno R3)

```

1 // 106119100 C++ code to blink the LED
2
3 void setup() {
4     pinMode(5, OUTPUT);
5     pinMode(6, OUTPUT);
6     pinMode(7, OUTPUT);
7     pinMode(8, OUTPUT);
8 }
9
10 void displayNumber(int n)
11 {
12     digitalWrite(5, (n&1) ? HIGH : LOW);
13     n >>= 1;
14     digitalWrite(6, (n&1) ? HIGH : LOW);
15     n >>= 1;
16     digitalWrite(7, (n&1) ? HIGH : LOW);
17     n >>= 1;
18     digitalWrite(8, (n&1) ? HIGH : LOW);
19 }
20
21 void loop() {
22     for (int i = 0; i <= 15; i += 2){
23         displayNumber(i);
24         delay(2000);
25     }
26 }

```

Serial Monitor

15:28 07-03-2022

Date : 28/03/2022

CSLR61 : EMBEDDED SYSTEMS LAB-6

Roll no. : **106119100**

Name : **Rajneesh Pandey**

Section : **CSE-B**

1. Interface an ultrasonic sensor and display the distance measured in a screen and glow red, blue and green led if the distance is close, moderate and far respectively. Play different tones for different cases.

Code

```
// 106119100 Lab6-1 Interface an ultrasonic sensor

#define F(x) x ? HIGH : LOW
const int red = 3;
const int blue = 4;
const int green = 5;
const int speaker = 10;
const int ping = 7;

void setup()
{
    // initialize serial communication:
    Serial.begin(9600);
    pinMode(red, OUTPUT);
    pinMode(blue, OUTPUT);
    pinMode(green, OUTPUT);
    pinMode(speaker, OUTPUT);
}

void displayLight(int code)
{
    digitalWrite(red, F(code == 0));
    digitalWrite(blue, F(code == 1));
    digitalWrite(green, F(code == 2));
}

void playTone(long duration, int freq)
{
    duration *= 1000;
    int period = (1.0 / freq) * 100000;
    long elapsed_time = 0;
```

```

while (elapsed_time < duration)
{
    digitalWrite(speaker, HIGH);
    delayMicroseconds(period / 2);
    digitalWrite(speaker, LOW);
    delayMicroseconds(period / 2);
    elapsed_time += (period);
}
}

void loop()
{
    long duration, inches, cm;
    pinMode(ping, OUTPUT);
    digitalWrite(ping, LOW);
    delayMicroseconds(2);
    digitalWrite(ping, HIGH);
    delayMicroseconds(5);
    digitalWrite(ping, LOW);
    pinMode(ping, INPUT);
    duration = pulseIn(ping, HIGH);
    inches = microsecondsToInches(duration);
    cm = microsecondsToCentimeters(duration);
    Serial.print(inches);
    Serial.print("in, ");
    Serial.print(cm);
    Serial.print("cm");
    Serial.println();
    if (cm < 50)
    {
        displayLight(0);
        playTone(300, 100);
    }
    else if (cm < 150)
    {

```

```
        displayLight(1);
        playTone(300, 200);
    }
    else
    {
        displayLight(2);
        playTone(300, 300);
    }
    delay(100);
}
long microsecondsToInches(long microseconds)
{
    return microseconds / 74 / 2;
}
long microsecondsToCentimeters(long microseconds)
{
    return microseconds / 29 / 2;
}
```

Output

Circuit design Frantic Amur-Esb x Muted

Circuit design Sizzling Bruticus x

https://www.tinkercad.com/things/bYCB0XWRnFa-frantic-amur-esboo/edit#tenant=circuits

All changes saved

Code Stop Simulation Send To

Simulator time: 00:00:21

Ultrasonic Distance Sensor
Name 1

```
56 {  
57   displayLight(0);  
58   playTone(300, 100);  
59 }  
60 else if (cm < 150)  
61 {  
62   displayLight(1);  
63   playTone(300, 200);  
64 }  
else  
65 {  
66   displayLight(2);  
67   playTone(300, 300);  
68 }  
69 delay(100);  
70 }  
long microsecondsToInches(long microseconds)  
{  
74   return microseconds / 74 / 2;  
75 }  
long microsecondsToCentimeters(long microseconds)  
{  
77   return microseconds / 29 / 2;  
78 }  
79 }
```

Serial Monitor

186in, 271cm
67in, 171cm
67in, 173cm
67in, 173cm
67in, 173cm
67in, 173cm
67in, 173cm
67in, 173cm

Send Clear

Circuit design Frantic Amur-Esb x Muted

Circuit design Sizzling Bruticus x

https://www.tinkercad.com/things/bYCB0XWRnFa-frantic-amur-esboo/edit#tenant=circuits

All changes saved

Code Stop Simulation Send To

Simulator time: 00:00:34.135

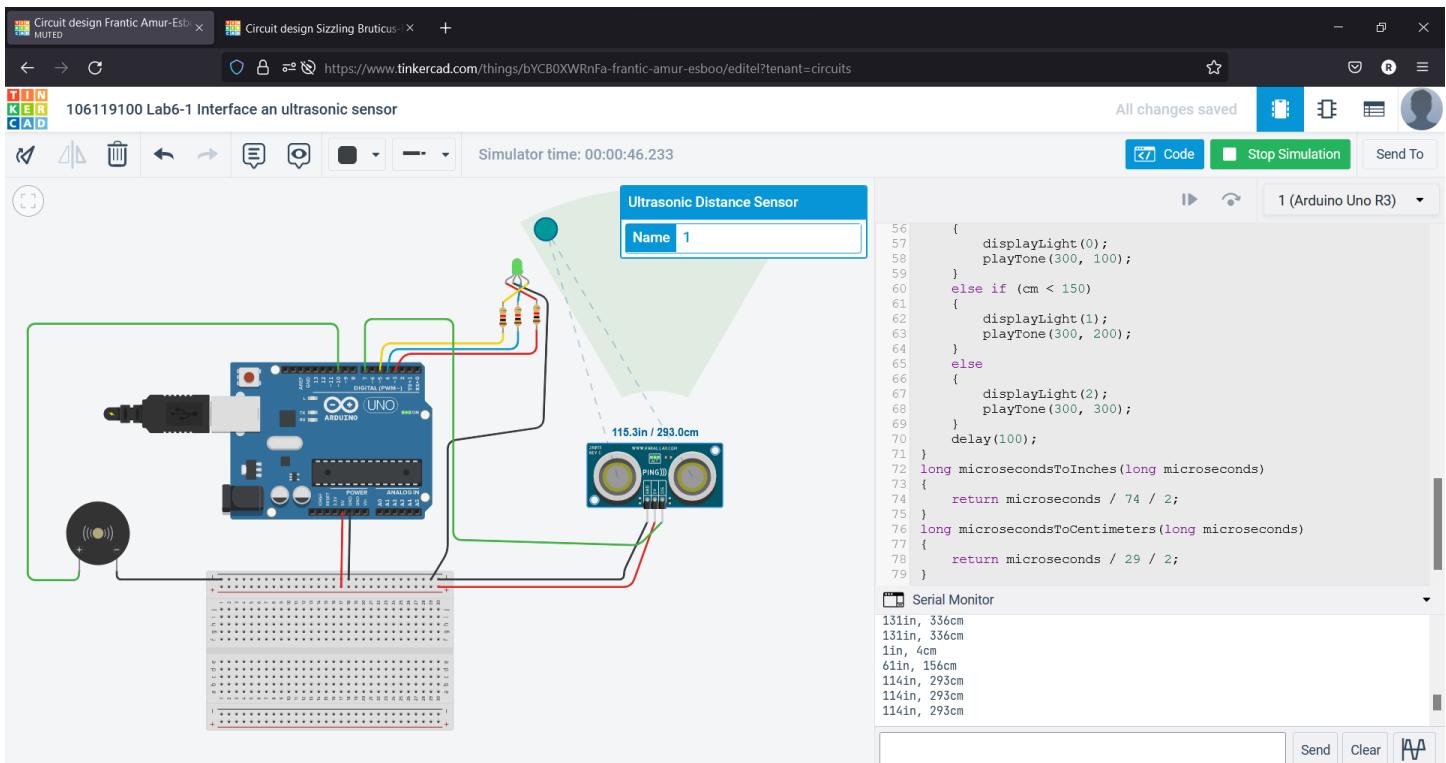
Ultrasonic Distance Sensor
Name 1

```
56 {  
57   displayLight(0);  
58   playTone(300, 100);  
59 }  
60 else if (cm < 150)  
61 {  
62   displayLight(1);  
63   playTone(300, 200);  
64 }  
else  
65 {  
66   displayLight(2);  
67   playTone(300, 300);  
68 }  
69 delay(100);  
70 }  
long microsecondsToInches(long microseconds)  
{  
74   return microseconds / 74 / 2;  
75 }  
long microsecondsToCentimeters(long microseconds)  
{  
77   return microseconds / 29 / 2;  
78 }  
79 }
```

Serial Monitor

67in, 173cm
67in, 173cm
67in, 173cm
131in, 336cm
131in, 336cm
131in, 336cm
131in, 336cm

Send Clear



2. Interface a PIR sensor with Arduino and blink a led if any movement is captured by the sensor and count the number of movements. Make a buzzer when movement detected

Code:

```

// 106119100 Interface a PIR sensor with Arduino
#define F(x) x ? HIGH : LOW
const int led = 4;
const int pir = 3;
const int speaker = 10;
int totalMovements = 0;
bool isMoving = false;
void setup()
{
    Serial.begin(9600);
    pinMode(led, OUTPUT);
    pinMode(pir, INPUT);
}

```

```

pinMode(speaker, OUTPUT);
}
void playTone(long duration, int freq)
{
    duration *= 1000;
    int period = (1.0 / freq) * 100000;
    long elapsed_time = 0;
    while (elapsed_time < duration)
    {
        digitalWrite(speaker, HIGH);
        delayMicroseconds(period / 2);
        digitalWrite(speaker, LOW);
        delayMicroseconds(period / 2);
        elapsed_time += (period);
    }
}
void loop()
{
    int pirStat = digitalRead(pir);
    if (pirStat == HIGH)
    {
        if (!isMoving)
        {
            playTone(300, 300);
            totalMovements += 1;
            Serial.print("Total movements: ");
            Serial.print(totalMovements);
            Serial.print("\n");
            isMoving = true;
        }
        digitalWrite(redLed, HIGH);
    }
    else
    {

```

```

    isMoving = false;
    digitalWrite(led, LOW);
}
}

```

Output:

The screenshot shows a Tinkercad workspace for a project titled "106119100 Lab6 - 2 Interface a PIR sensor with Arduino".

Circuit Diagram: On the left, a breadboard is populated with a USB cable, an Arduino Uno, a PIR sensor, and a speaker. Wires connect the PIR sensor's output to digital pin 3 on the Arduino, and the Arduino's digital pin 3 to the breadboard ground. The Arduino's digital pin 4 connects to a 10k pull-up resistor and then to the speaker's positive terminal. The speaker's negative terminal is connected to ground.

Code: In the center-right, the Arduino IDE code is displayed:

```

1 #define F(x) x ? HIGH : LOW
2
3 const int led = 4;
4 const int pir = 3;
5 const int speaker = 10;
6 int totalMovements = 0;
7 bool isMoving = false;
8
9 void setup()
10 {
11   Serial.begin(9600);
12   pinMode(led, OUTPUT);
13   pinMode(pir, INPUT);
14   pinMode(speaker, OUTPUT);
15 }
16 void playTone(long duration, int freq)
17 {
18   duration *= 1000;
19   int period = (1.0 / freq) * 100000;
20   long elapsed_time = 0;
21   while (elapsed_time < duration)
22   {
23     if (isMoving)
24       tone(speaker, freq);
25     else
26       noTone(speaker);
27     delayMicroseconds(period);
28     elapsed_time += period;
29   }
30 }
31
32 void loop()
33 {
34   if (digitalRead(pir) == F(isMoving))
35   {
36     isMoving = !isMoving;
37     playTone(1000, 1000);
38   }
39 }

```

Serial Monitor: On the right, the serial monitor window shows the output: "Total movements: 1", "Total movements: 2", "Total movements: 3", and "Total movements: 4".

Circuit design Copy of 106119101 x New Tab https://www.tinkercad.com/things/7nPPBoyQh49-copy-of-106119100-lab6-1-interface-an-ultrasonic-sensor/edite?tenant=circuits

All changes saved

106119100 Lab6 - 2 Interface a PIR sensor with Arduino

Simulator time: 00:00:55.199

Code Stop Simulation Send To 1 (Arduino Uno R3)

PIR Sensor
Name 1

```

1 #define F(x) x ? HIGH : LOW
2 const int led = 4;
3 const int pir = 3;
4 const int speaker = 10;
5 int totalMovements = 0;
6 bool isMoving = false;
7 void setup()
8 {
9   Serial.begin(9600);
10  pinMode(led, OUTPUT);
11  pinMode(pir, INPUT);
12  pinMode(speaker, OUTPUT);
13 }
14 void playTone(long duration, int freq)
15 {
16   duration *= 1000;
17   int period = (1.0 / freq) * 100000;
18   long elapsed_time = 0;
19   while (elapsed_time < duration)
20     ;

```

Serial Monitor

```

Total movements: 1
Total movements: 2
Total movements: 3
Total movements: 4
Total movements: 5
Total movements: 6
Total movements: 7
Total movements: 1
Total movements: 2

```

Send Clear