

# **34338 Telecommunication programming projects with Arduino, Winter 2026**

**Group no: 3**

**Group members:**

**s235162 - Hector Foss  
s224761 - Nikolaj Nguyen**

**Date: 06-01-2026**

## Exercise 1 - Morse code

**1a:** Morse code uses 5 "bits" for representing numbers 0-9, how many bits are needed if you would use binary? Why is this not possible in Morse code?

**1b:** What is the value of a after the loop?

```
int a = 1; for(int i = 0; i < 5; i++){ a += a; }
```

The value of *a* would be 5, since the loop runs 5 iterations (from 0 to 4).

**1c:** Make a program that morses "SOS". You should follow the international requirement for morse code shown in figure 1.

```
1 // Given each LED pin a name and don't let it be changed by the program:
2 const int LED = 12;
3 const int time = 300;
4
5 // the setup routine runs once when you press reset:
6 void setup() {
7     // initialize the digital pins as an output.
8     pinMode(LED, OUTPUT);
9 }
10
11 // the loop routine runs over and over again forever:
12 void loop() {
13     s();
14     delay(3 * time);
15     o();
16     delay(3 * time);
17     s();
18     delay(7 * time);
19 }
20
21
22 void s() {
23     for (int i = 0; i < 3; i++) {
24         digitalWrite(LED, HIGH);
25         delay(time);
26         digitalWrite(LED, LOW);
27         delay(time);
28     }
29 }
30
31
32 void o() {
33     for (int i = 0; i < 3; i++) {
34         digitalWrite(LED, HIGH);
35         delay(3 * time);
36         digitalWrite(LED, LOW);
37         delay(time);
38     }
39 }
```

## 1d: Try setting ledPin = LED BUILTIN, what happens?

The light power is dampened and is much weaker.

## 1e: Update the program to Morse your name

```
1 // Given each LED pin a name and don't let it be changed by the program:
2 const int LED = 12;
3 const int time = 300;
4
5 // the setup routine runs once when you press reset:
6 void setup() {
7   pinMode(LED, INPUT);
8 }
9
10 void loop() {
11   // Spells out the name: Hector in morse code
12   h();
13   delay(3 * time);
14   e();
15   delay(3 * time);
16   c();
17   delay(3 * time);
18   t();
19   delay(3 * time);
20   o();
21   delay(3 * time);
22   r();
23   delay(7 * time);
24 }
25
26
27 void h() {
28   for (int i = 0; i < 4; i++) {
29     digitalWrite(LED, HIGH);
30     delay(time);
31     digitalWrite(LED, LOW);
32     delay(time);
33   }
34 }
35
36
37 void e() {
38   digitalWrite(LED, HIGH);
39   delay(time);
40   digitalWrite(LED, LOW);
41   delay(time);
42 }
43
44
45 void t() {
46   for (int i = 0; i < 2; i++) {
47
48     // Long
49     digitalWrite(LED, HIGH);
50     delay(3 * time);
51     digitalWrite(LED, LOW);
52     delay(3 * time);
53
54     // Short
55     digitalWrite(LED, HIGH);
56     delay(time);
57     digitalWrite(LED, LOW);
58     delay(time);
59   }
60 }
61
62 void t() {
63   digitalWrite(LED, HIGH);
64   delay(3 * time);
65   digitalWrite(LED, LOW);
66   delay(3 * time);
67 }
68
69
70 void o() {
71   for (int i = 0; i < 3; i++) {
72     digitalWrite(LED, HIGH);
73     delay(3 * time);
74     digitalWrite(LED, LOW);
75     delay(time);
76   }
77 }
78
79
80 void r() {
81   // Short
82   digitalWrite(LED, HIGH);
83   delay(time);
84   digitalWrite(LED, LOW);
85   delay(time);
86
87   // Long
88   digitalWrite(LED, HIGH);
89   delay(3 * time);
90   digitalWrite(LED, LOW);
91   delay(3 * time);
92
93   // Short
94   digitalWrite(LED, HIGH);
95   delay(time);
96   digitalWrite(LED, LOW);
97   delay(time);
98 }
```

## 1f: Finally, update the program to morse your name using for-loops

This is done in exercise 1e, which was the initial solution.

## Exercise 2 - More LEDs

### 2a: What is $42\%5$ ?

Modulo finds the remainder after dividing one number (dividend) by another (modulus). For  $42\%5$ , 5 and 40 are divisible leaving us with a rest product of 2.

$$42\%5=2$$

### 2b: Make a traffic light:

The traffic light is simple and goes through the 4 states on loop. The LEDs change as it prints to the serial monitor.

```
void traffic(void) {  
    Serial.println("STOP");  
    digitalWrite(yellowLED, LOW);  
    digitalWrite(redLED, HIGH);  
    delay(waitTraffic);  
  
    Serial.println("Get Ready To Go");  
    digitalWrite(yellowLED, HIGH);  
    delay(waitTraffic);  
  
    Serial.println("GO");  
    digitalWrite(redLED, LOW);  
    digitalWrite(yellowLED, LOW);  
    digitalWrite(greenLED, HIGH);  
    delay(waitTraffic);  
  
    Serial.println("Get Ready To Stop");  
    digitalWrite(greenLED, LOW);  
    digitalWrite(yellowLED, HIGH);  
    delay(waitTraffic);  
}
```

## 2c: Make a binary counter:

The binary counter works in the same way as the traffic light going through each state with a delay in between.

```
50 void binaryCounter(void){
51     Serial.println(0);
52     digitalWrite(greenLED, LOW);
53     digitalWrite(yellowLED, LOW);
54     digitalWrite(redLED, LOW);
55     delay(pause);
56
57     Serial.println(1);
58     digitalWrite(redLED, HIGH);
59     delay(pause);
60
61     Serial.println(2);
62     digitalWrite(greenLED, LOW);
63     digitalWrite(yellowLED, HIGH);
64     digitalWrite(redLED, LOW);
65     delay(pause);
66
67     Serial.println(3);
68     digitalWrite(greenLED, LOW);
69     digitalWrite(yellowLED, HIGH);
70     digitalWrite(redLED, HIGH);
71     delay(pause);
72
73     Serial.println(4);
74     digitalWrite(greenLED, HIGH);
75     digitalWrite(yellowLED, LOW);
76     digitalWrite(redLED, LOW);
77     delay(pause);
78
79     Serial.println(5);
80     digitalWrite(greenLED, HIGH);
81     digitalWrite(yellowLED, LOW);
82     digitalWrite(redLED, HIGH);
83     delay(pause);
84
85     Serial.println(6);
86     digitalWrite(greenLED, HIGH);
87     digitalWrite(yellowLED, HIGH);
88     digitalWrite(redLED, LOW);
89     delay(pause);
90
91     Serial.println(7);
92     digitalWrite(greenLED, HIGH);
93     digitalWrite(yellowLED, HIGH);
94     digitalWrite(redLED, HIGH);
95     delay(pause);
96 }
97 }
```

**2d: What could you have used the %-operator for in this exercise?**

We could have used the modulus with a value of 4 with a counter and then used the switch function.to determine the steps of the traffic light. Then at the end of each loop we could add 1 to the counter.

## Exercise 3

### 3a: What is the difference between pinMode(PIN, INPUT) and pinMode(PIN, INPUT PULLUP)?

- pinMode(PIN, INPUT) sets the pin as plain input.
- pinMode(PIN, INPUT PULLUP) also sets the pin as input but also enables a built-in ‘pull-up’ resistor’. The purpose of this is to avoid that an input is floating. Adding the pull-up will drive the input to a desired state, either high or low.

### 3b: What is the operator ! used for?

The ‘!’ is the logical operator used for negation.

### 3c: Control the LED with the button

#### 3c.1. While the button is pushed down the LED should be turned on.

You can use digitalRead(PIN) and digitalWrite(PIN, ) to read from the button pin and write to the LED pin

```
1  const uint8_t button = 10;
2  const uint8_t redLED = 12;
3  const uint16_t pause = 1000;
4
5
6  void setup() {
7      // put your setup code here, to run once:
8      pinMode(button, INPUT_PULLUP);
9      pinMode(redLED, OUTPUT);
10
11     digitalWrite(redLED, LOW);
12 }
13
14 void loop() {
15     // put your main code here, to run repeatedly:
16     if (digitalRead(button) != HIGH) {
17         digitalWrite(redLED, HIGH);
18     } else {
19         digitalWrite(redLED, LOW);
20     }
21 }
```

**3c.2. Make a latching button.** A latching button should change state if you push it. (Push to turn on LED. Push again to turn LED off)

```
1 const uint8_t button = 10;
2 const uint8_t redLED = 12;
3 const uint16_t pause = 1000;
4
5 bool buttonstate = 0;
6 bool ledstate = 0;
7 void setup() {
8     // put your setup code here, to run once:
9     pinMode(button, INPUT_PULLUP);
10    pinMode(redLED, OUTPUT);
11
12    digitalWrite(redLED, LOW);
13}
14
15 void loop() {
16     // put your main code here, to run repeatedly:
17
18     if (buttonstate == LOW) {
19         if (digitalRead(button) == LOW) {
20             if (ledstate == 0) {
21                 digitalWrite(redLED, HIGH);
22                 ledstate = 1;
23             } else {
24                 digitalWrite(redLED, LOW);
25                 ledstate = 0;
26             }
27             buttonstate = 1;
28         }
29     } else {
30         if (digitalRead(button) == HIGH) {
31             buttonstate = 0;
32         }
33     }
34 }
35
```

**3d: How often does your program check if the button has been pushed?  
Does this seem reasonable?**

It checks as fast as possible with no delay function. This is probably too fast and could be slowed down with a delay function or something with a similar function.

## Exercise 4: Fritzing

