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

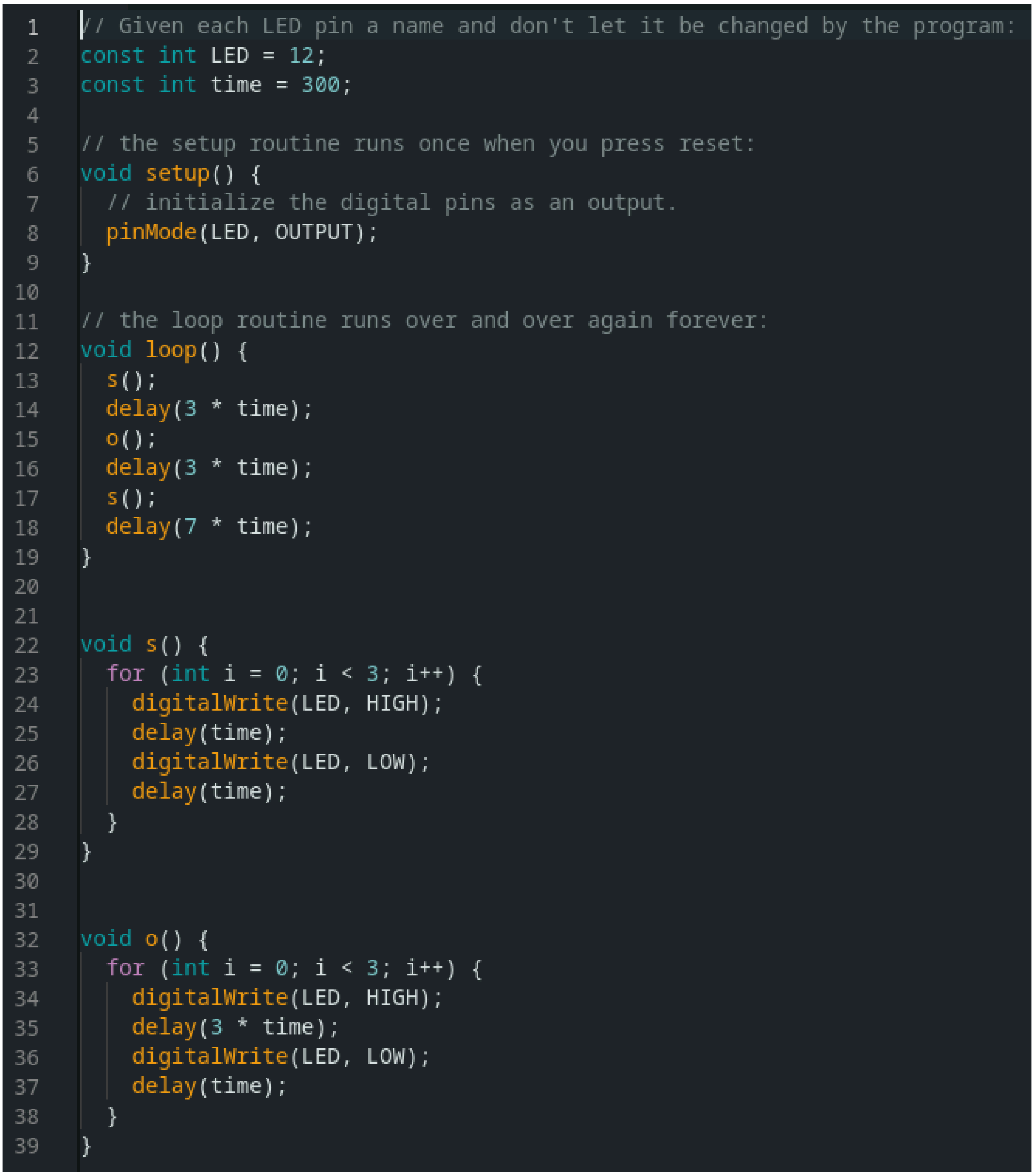
1a: Morse code uses 5 ”bits” for representing numbers 0-9, how many bits are needed if you would use binary? Why is this not possible in Morse code?

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.

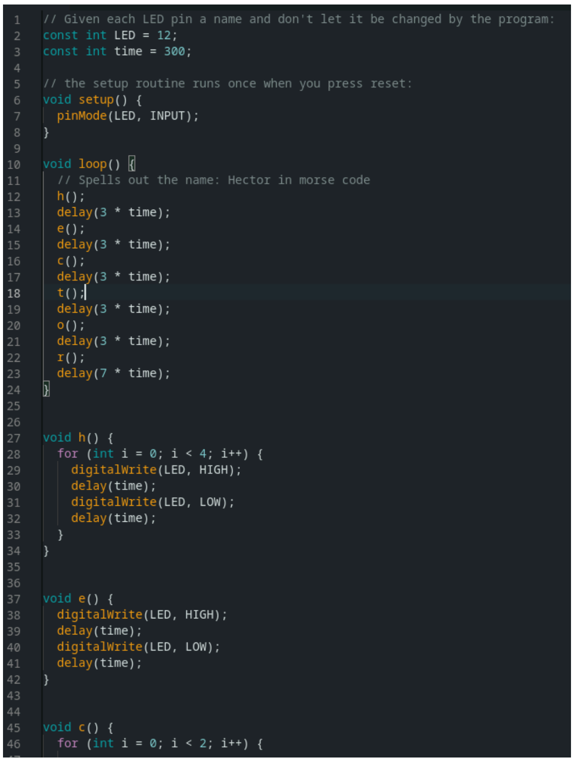


```
// Given each LED pin a name and don't let it be changed by the program:
const int LED = 12;
const int time = 300;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pins as an output.
  pinMode(LED, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  s();
  delay(3 * time);
  o();
  delay(3 * time);
  s();
  delay(7 * time);
}


void s() {
  for (int i = 0; i < 3; i++) {
    digitalWrite(LED, HIGH);
    delay(time);
    digitalWrite(LED, LOW);
    delay(time);
  }
}


void o() {
  for (int i = 0; i < 3; i++) {
    digitalWrite(LED, HIGH);
    delay(3 * time);
    digitalWrite(LED, LOW);
    delay(time);
  }
}
```

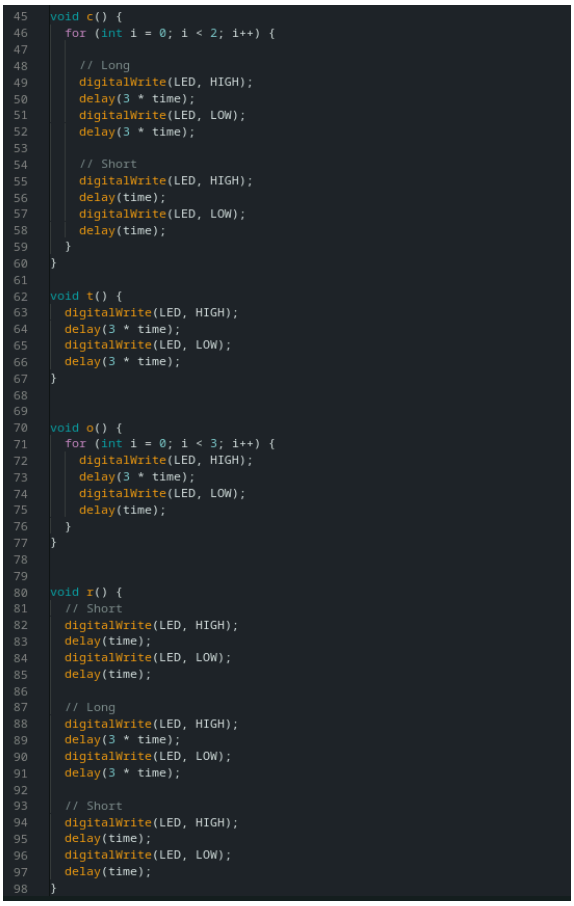
1d: Try setting ledPin = LED BUILTIN, what happens?

The light power is dampened and is much weaker.

1e: Update the program to Morse your name



```
// Given each LED pin a name and don't let it be changed by the program:
const int LED = 12;
const int time = 300;

// the setup routine runs once when you press reset:
void setup() {
  pinMode(LED, INPUT);
}

void loop() {
  // Spells out the name: Hector in morse code
  h();
  delay(3 * time);
  e();
  delay(3 * time);
  c();
  delay(3 * time);
  t();
  delay(3 * time);
  o();
  delay(3 * time);
  r();
  delay(7 * time);
}


void h() {
  for (int i = 0; i < 4; i++) {
    digitalWrite(LED, HIGH);
    delay(time);
    digitalWrite(LED, LOW);
    delay(time);
  }
}


void e() {
  digitalWrite(LED, HIGH);
  delay(time);
  digitalWrite(LED, LOW);
  delay(time);
}


void c() {
  for (int i = 0; i < 2; i++) {
```



```
void c() {
  for (int i = 0; i < 2; i++) {

    // Long
    digitalWrite(LED, HIGH);
    delay(3 * time);
    digitalWrite(LED, LOW);
    delay(3 * time);

    // Short
    digitalWrite(LED, HIGH);
    delay(time);
    digitalWrite(LED, LOW);
    delay(time);
  }
}

void t() {
  digitalWrite(LED, HIGH);
  delay(3 * time);
  digitalWrite(LED, LOW);
  delay(3 * time);
}


void o() {
  for (int i = 0; i < 3; i++) {
    digitalWrite(LED, HIGH);
    delay(3 * time);
    digitalWrite(LED, LOW);
    delay(time);
  }
}


void r() {
  // Short
  digitalWrite(LED, HIGH);
  delay(time);
  digitalWrite(LED, LOW);
  delay(time);

  // Long
  digitalWrite(LED, HIGH);
  delay(3 * time);
  digitalWrite(LED, LOW);
  delay(3 * time);

  // Short
  digitalWrite(LED, HIGH);
  delay(time);
  digitalWrite(LED, LOW);
  delay(time);
}
```

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

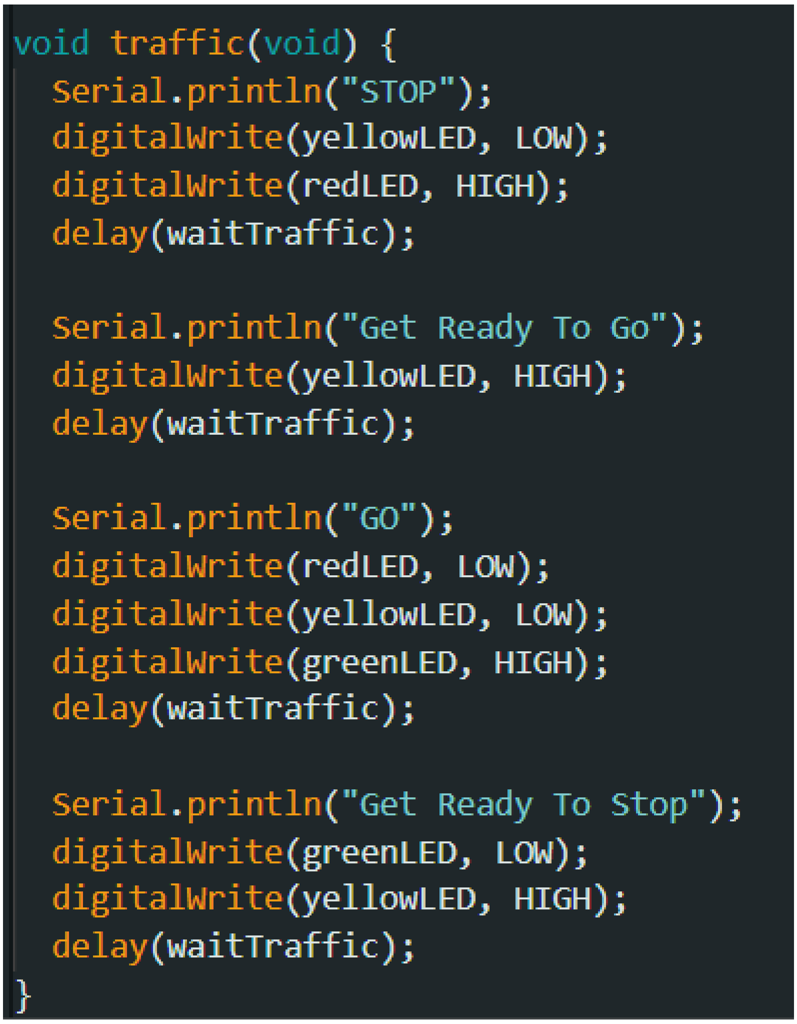
2a: What is 42%5?

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:

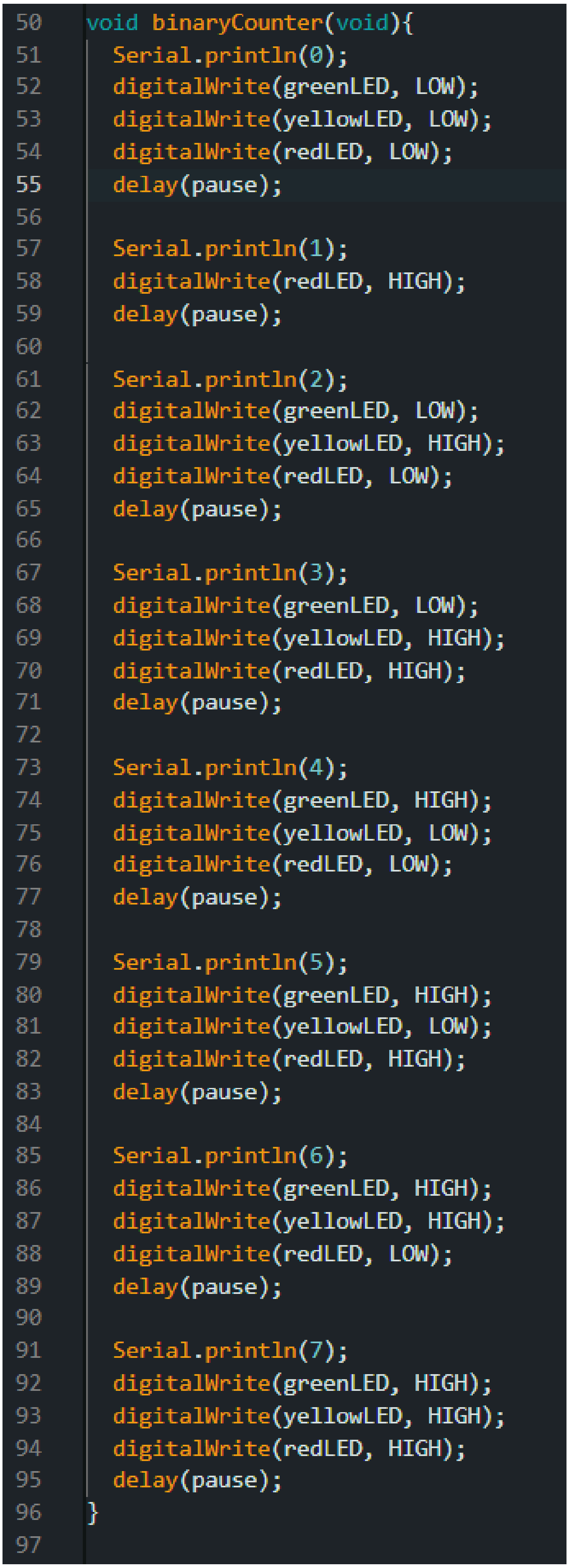
The traffic light is simple and goes through the 4 states on loop. The LEDs change as it prints to the serial monitor.



```
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.



```
void binaryCounter(void){
  Serial.println(0);
  digitalWrite(greenLED, LOW);
  digitalWrite(yellowLED, LOW);
  digitalWrite(redLED, LOW);
  delay(pause);

  Serial.println(1);
  digitalWrite(redLED, HIGH);
  delay(pause);

  Serial.println(2);
  digitalWrite(greenLED, LOW);
  digitalWrite(yellowLED, HIGH);
  digitalWrite(redLED, LOW);
  delay(pause);

  Serial.println(3);
  digitalWrite(greenLED, LOW);
  digitalWrite(yellowLED, HIGH);
  digitalWrite(redLED, HIGH);
  delay(pause);

  Serial.println(4);
  digitalWrite(greenLED, HIGH);
  digitalWrite(yellowLED, LOW);
  digitalWrite(redLED, LOW);
  delay(pause);

  Serial.println(5);
  digitalWrite(greenLED, HIGH);
  digitalWrite(yellowLED, LOW);
  digitalWrite(redLED, HIGH);
  delay(pause);

  Serial.println(6);
  digitalWrite(greenLED, HIGH);
  digitalWrite(yellowLED, HIGH);
  digitalWrite(redLED, LOW);
  delay(pause);

  Serial.println(7);
  digitalWrite(greenLED, HIGH);
  digitalWrite(yellowLED, HIGH);
  digitalWrite(redLED, HIGH);
  delay(pause);
}

```

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.

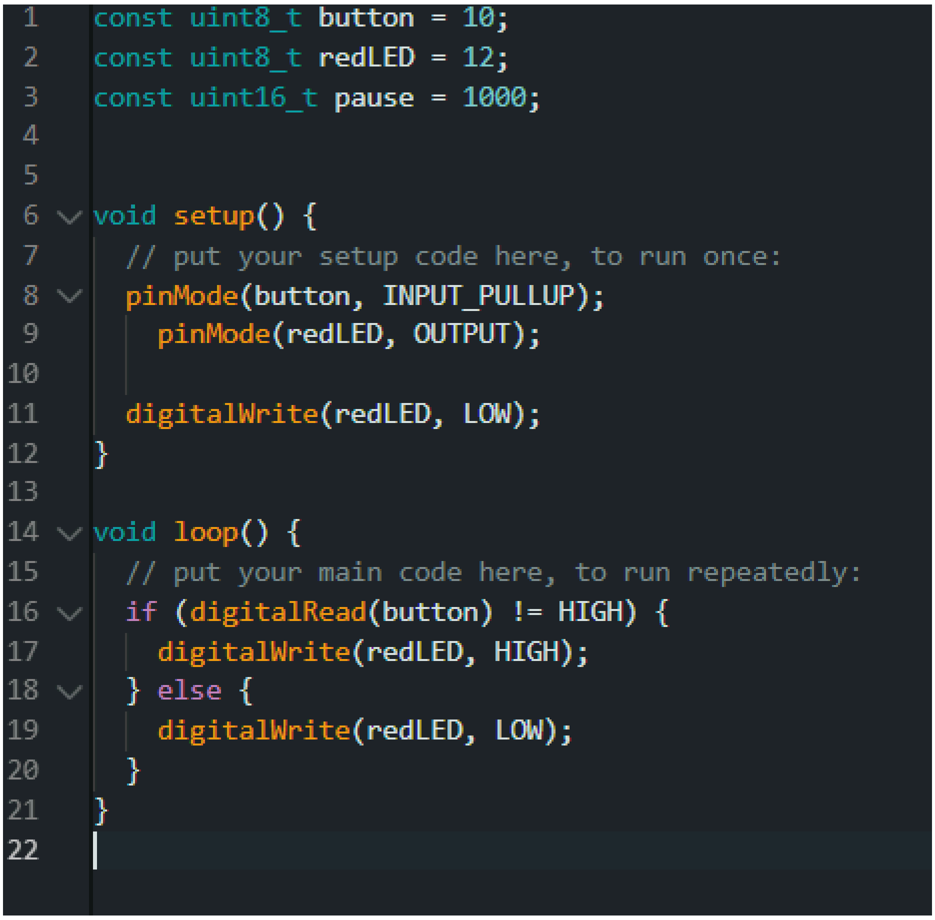
3b: What is the operator ! used for?

The '!' is the logical operator used for negation.

3c: Control the LED with the button

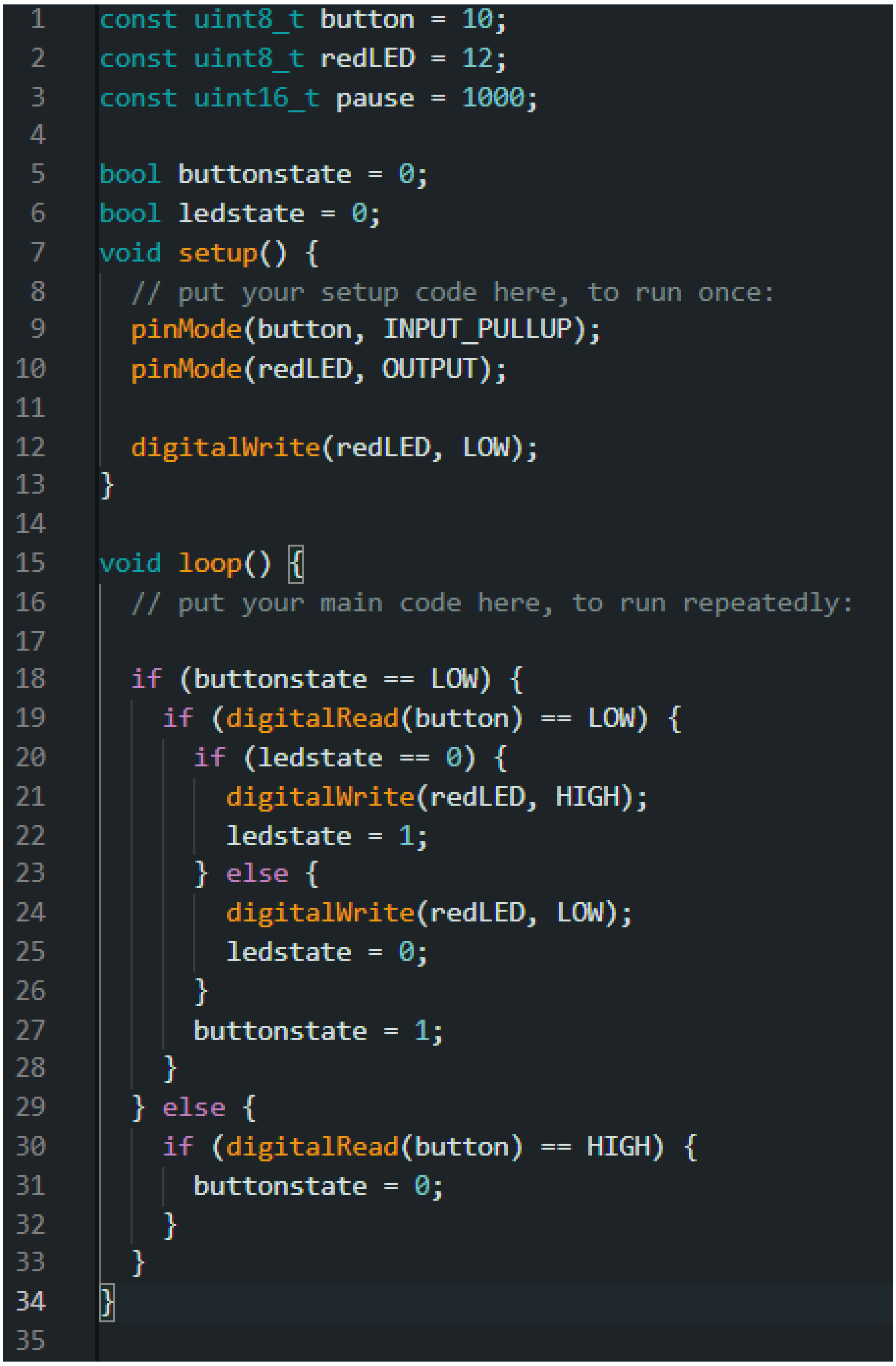
3c.1. While the button is pushed down the LED should be turned on.

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



```
const uint8_t button = 10;
const uint8_t redLED = 12;
const uint16_t pause = 1000;


void setup() {
  // put your setup code here, to run once:
  pinMode(button, INPUT_PULLUP);
    pinMode(redLED, OUTPUT);

  digitalWrite(redLED, LOW);
}

void loop() {
  // put your main code here, to run repeatedly:
  if (digitalRead(button) != HIGH) {
    digitalWrite(redLED, HIGH);
  } else {
    digitalWrite(redLED, LOW);
  }
}

```

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)

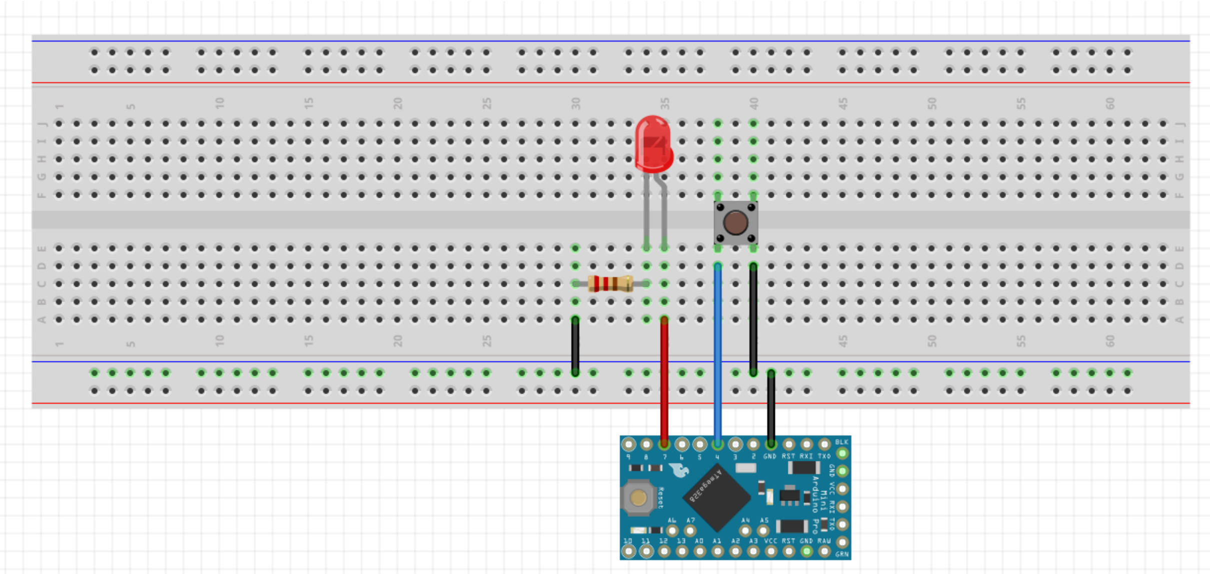


```
const uint8_t button = 10;
const uint8_t redLED = 12;
const uint16_t pause = 1000;

bool buttonstate = 0;
bool ledstate = 0;
void setup() {
  // put your setup code here, to run once:
  pinMode(button, INPUT_PULLUP);
  pinMode(redLED, OUTPUT);

  digitalWrite(redLED, LOW);
}

void loop() {
  // put your main code here, to run repeatedly:

  if (buttonstate == LOW) {
    if (digitalRead(button) == LOW) {
      if (ledstate == 0) {
        digitalWrite(redLED, HIGH);
        ledstate = 1;
      } else {
        digitalWrite(redLED, LOW);
        ledstate = 0;
      }
      buttonstate = 1;
    }
  } else {
    if (digitalRead(button) == HIGH) {
      buttonstate = 0;
    }
  }
}

```

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


1
5
10
15
20
25
30
35
40
45
50
55
60
J
I
H
G
F
E
D
C
B
A
9
8
7
6
5
4
3
2
GND
RST
RXI
TXO
BLK
GND
VCC
RXI
TXO
GRN
Atmega328
Reset
Arduino
Mini
Pro
A6
A7
A4
A5
10
11
12
13
A0
A1
A2
A3
VCC
RST
GND
RAW