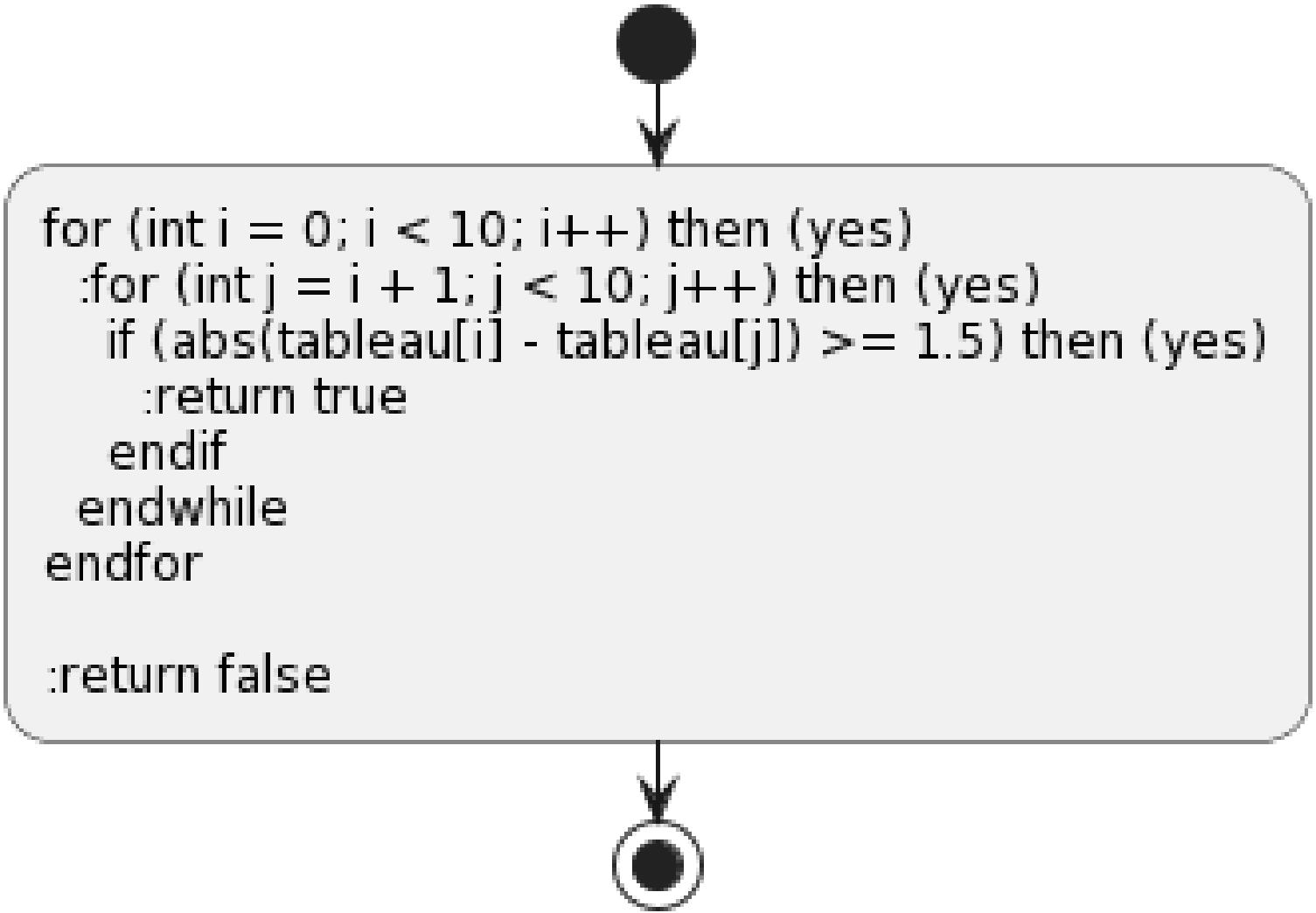
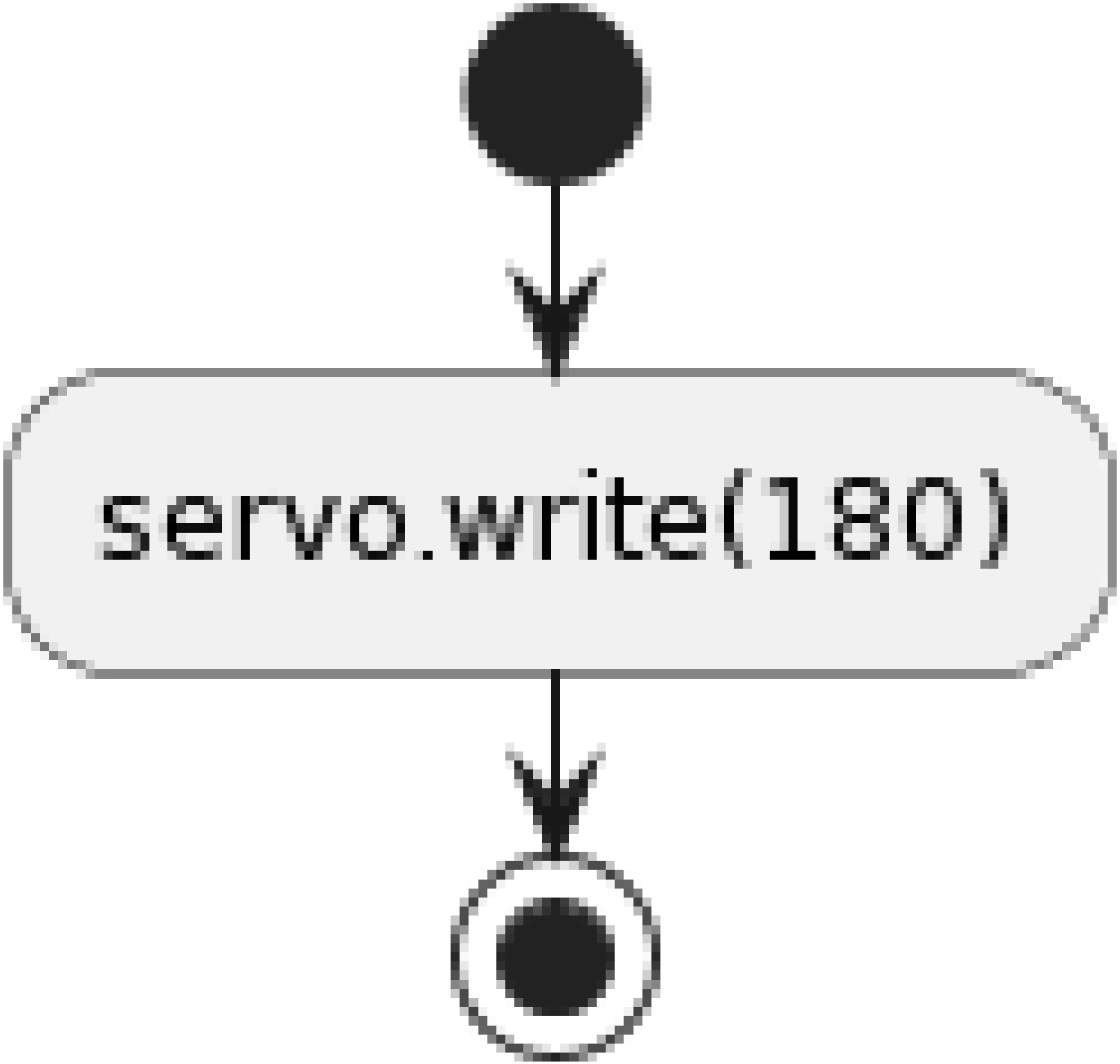
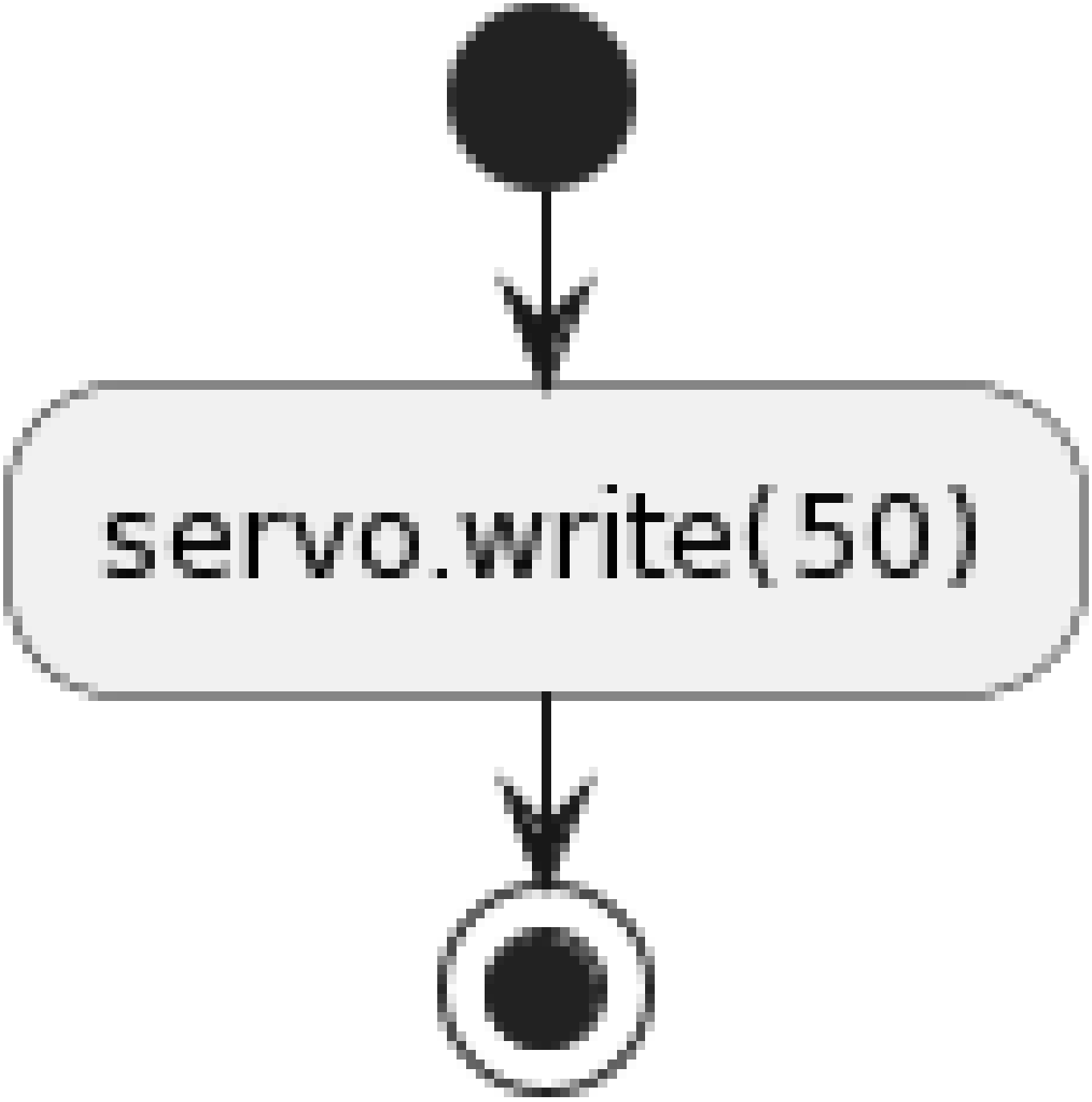
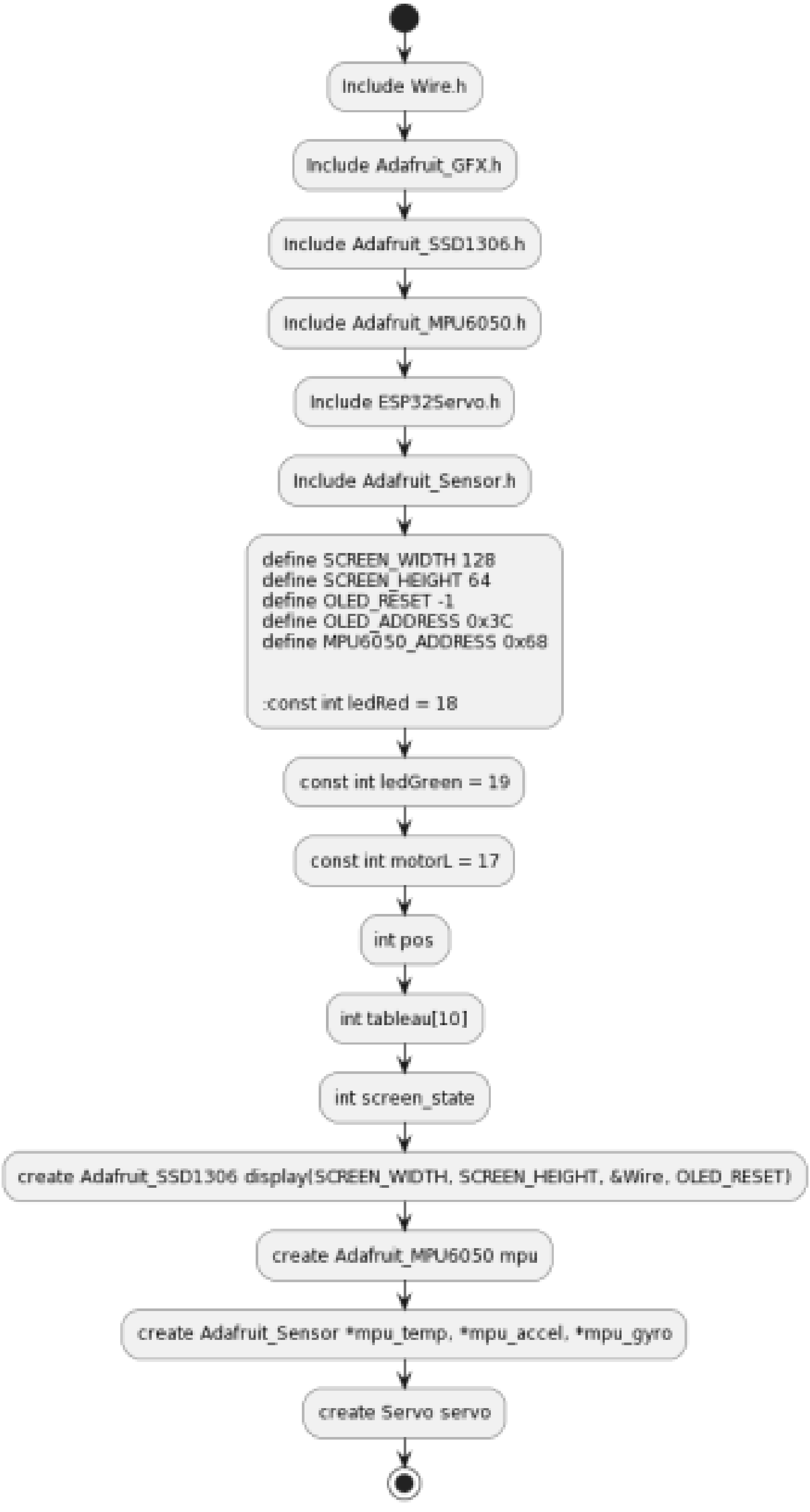
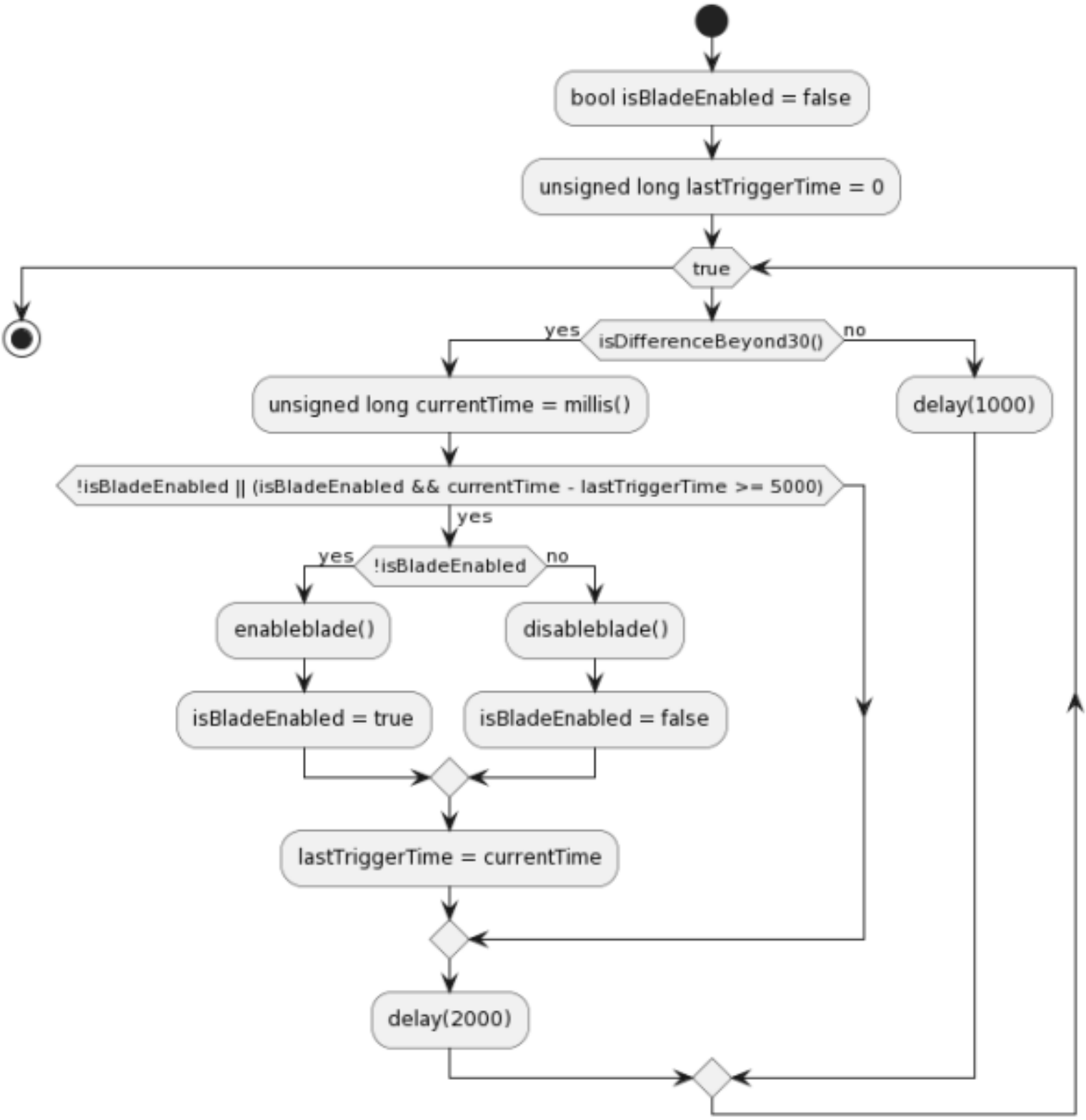
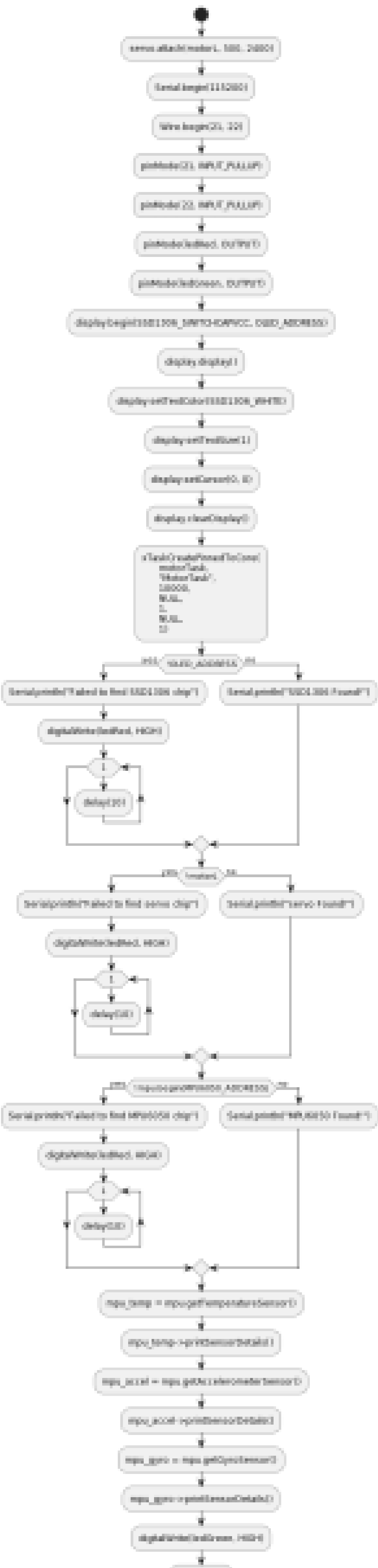

for (int i = 0; i < 10; i++) then (yes)
:for (int j = i + 1; j < 10; j++) then (yes)
if (abs(tableau[i] - tableau[j]) >= 1.5) then (yes)
:return true
endif
endwhile
endfor
:return false

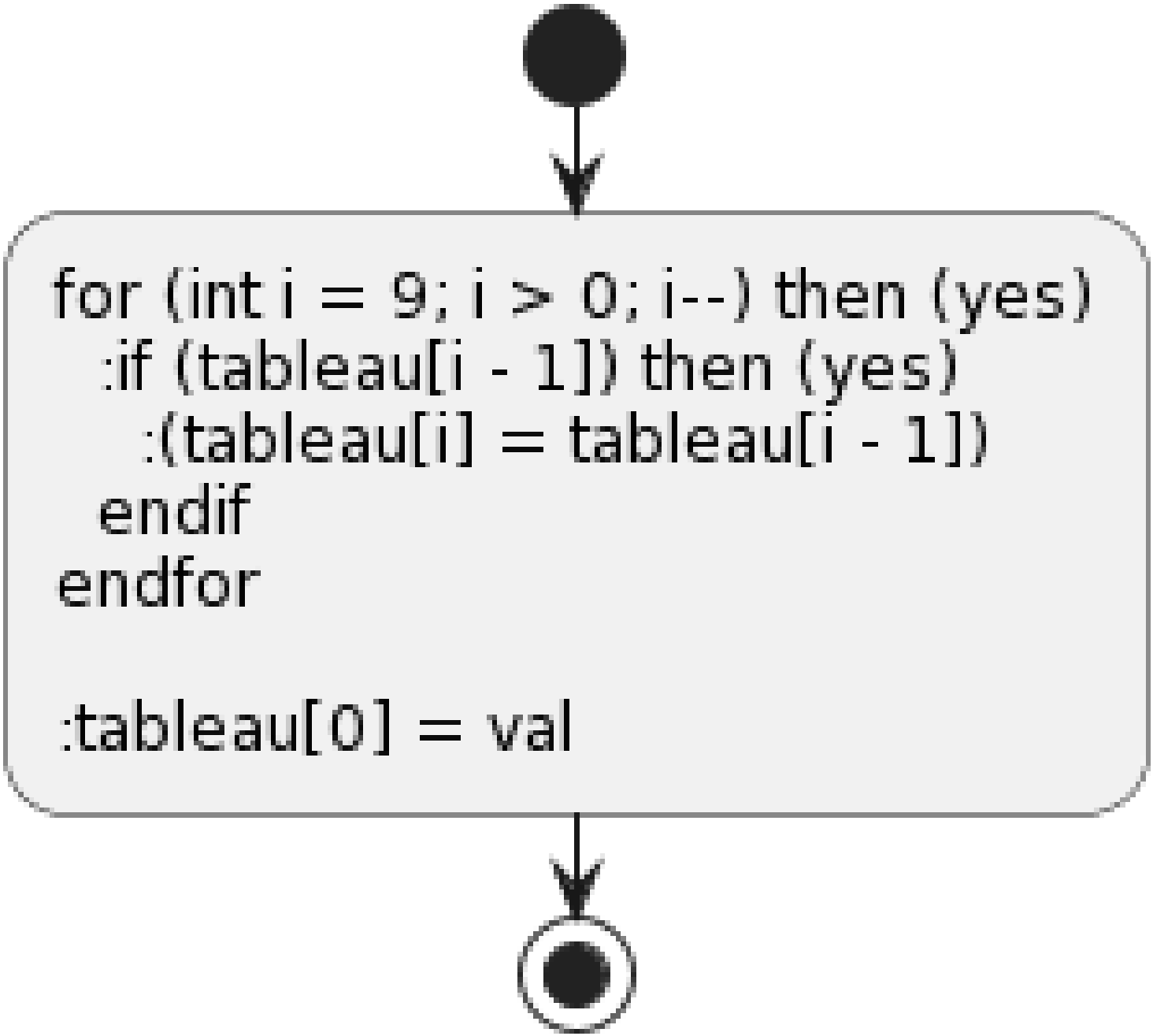

servo.write(180)

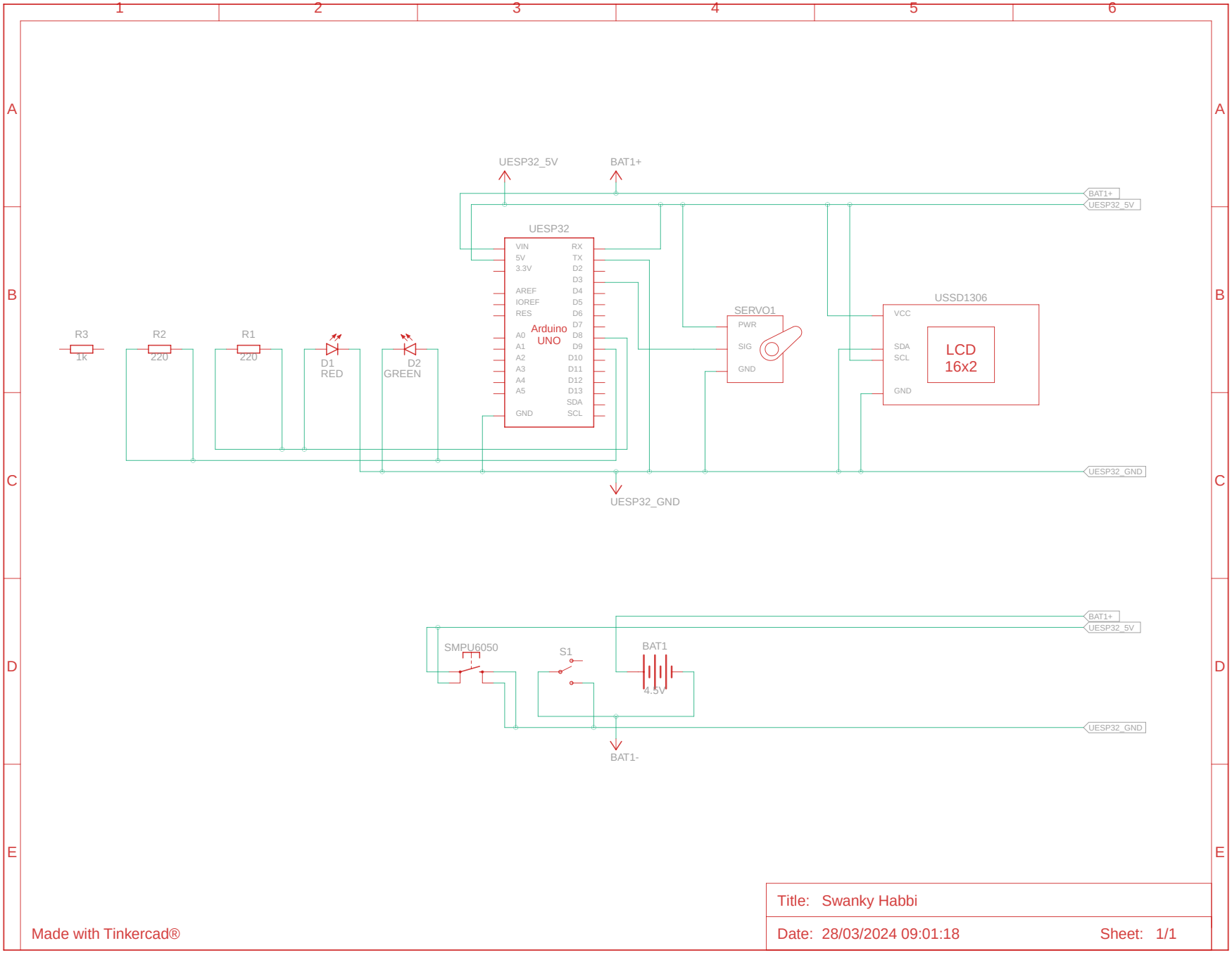

servo.write(50)

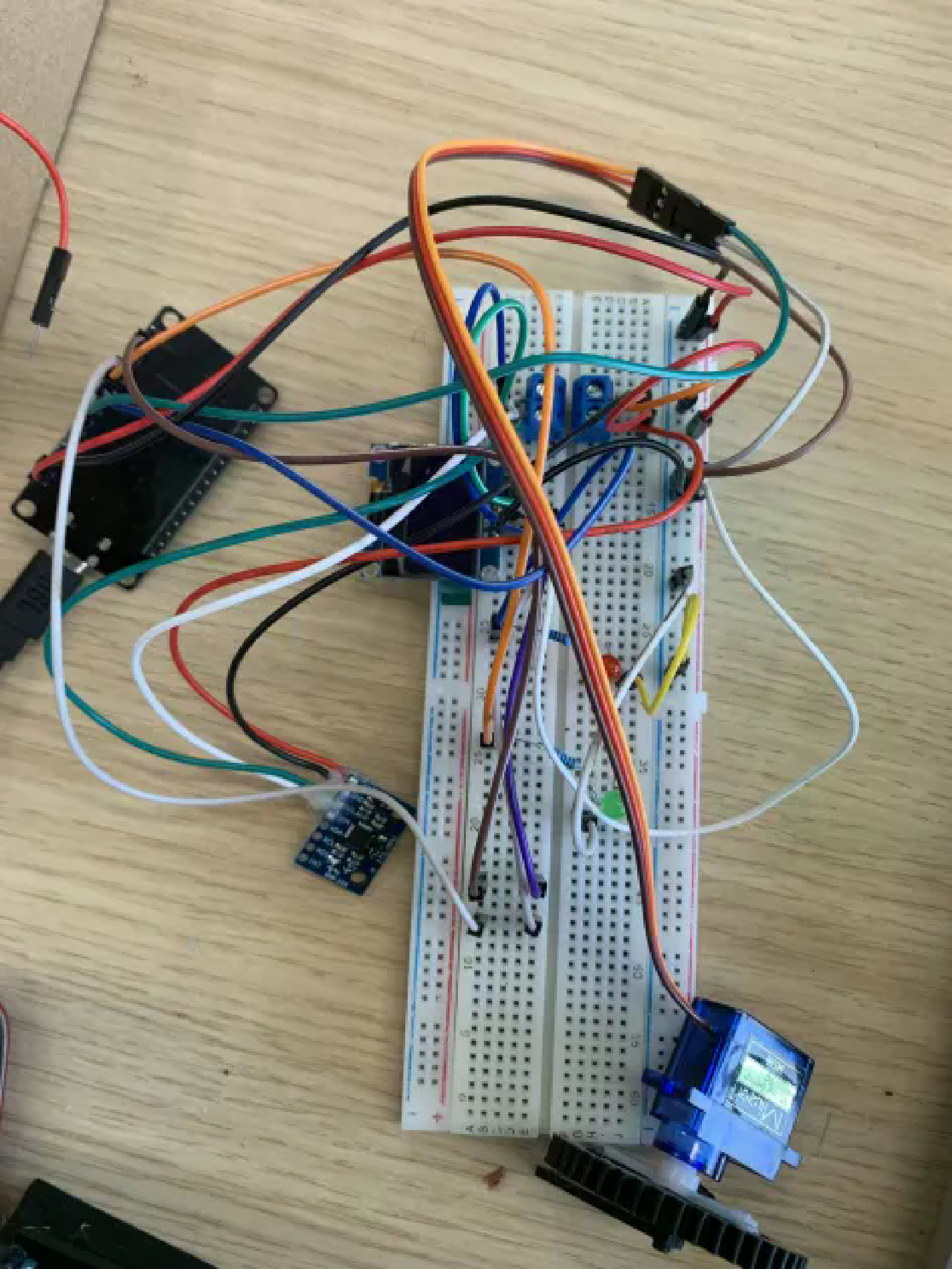

Include Wire.h
Include Adafruit_GFX.h
Include Adafruit_SSD1306.h
Include Adafruit_MPU6050.h
Include ESP32Servo.h
Include Adafruit_Sensor.h
define SCREEN_WIDTH 128
define SCREEN_HEIGHT 64
define OLED_RESET -1
define OLED_ADDRESS 0x3C
define MPU6050_ADDRESS 0x68
:const int ledRed = 18
const int ledGreen = 19
const int motorL = 17
int pos
int tableau[10]
int screen_state
create Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET)
create Adafruit_MPU6050 mpu
create Adafruit_Sensor *mpu_temp, *mpu_accel, *mpu_gyro
create Servo servo


bool isBladeEnabled = false
unsigned long lastTriggerTime = 0
true
isDifferenceBeyond30()
yes
no
unsigned long currentTime = millis()
delay(1000)
!isBladeEnabled || (isBladeEnabled && currentTime - lastTriggerTime >= 5000)
yes
!isBladeEnabled
yes
no
enableblade()
disableblade()
isBladeEnabled = true
isBladeEnabled = false
lastTriggerTime = currentTime
delay(2000)




for (int i = 9; i > 0; i--) then (yes)
:if (tableau[i - 1]) then (yes)
:(tableau[i] = tableau[i - 1])
endif
endfor
:tableau[0] = val


1
2
3
4
5
6
A
B
C
D
E
UESP32_5V
BAT1+
UESP32
VIN
5V
3.3V
AREF
IOREF
RES
A0
A1
A2
A3
A4
A5
GND
RX
TX
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
SDA
SCL
Arduino
UNO
R3
1k
R2
220
R1
220
D1
RED
D2
GREEN
SERVO1
PWR
SIG
GND
USSD1306
VCC
SDA
SCL
GND
LCD
16x2
UESP32_GND
SMPU6050
S1
BAT1
4.5V
BAT1-
Made with Tinkercad®
Title: Swanky Habbi
Date: 28/03/2024 09:01:18
Sheet: 1/1



```
const int button1 = 12;
const int button2 = 13;
const int LED1 = 2;
const int LED2 = 3;
const int LED3 = 4;
const int a = 7;
const int b = 6;
const int c = 5;
const int d = 11;
const int e = 10;
const int f = 8;
const int g = 9;
int counter = 0;
int ledStatus = 0;
int buttonState1 = 0;
int buttonState2 = 0;

void setup(){
  pinMode(button1, INPUT);
  pinMode(button2, INPUT);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);
  Serial.begin(9600);
}

void ledReset(){
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
}

void ledsBlink(){
  digitalWrite(LED1, HIGH);
  digitalWrite(LED3, HIGH);
  digitalWrite(LED2, HIGH);
  delay(500);
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  delay(500);
  digitalWrite(LED1, HIGH);
  digitalWrite(LED3, HIGH);
  digitalWrite(LED2, HIGH);
}

void clearDisplay(){
  digitalWrite(a, LOW);
  digitalWrite(b, LOW);
  digitalWrite(c, LOW);
  digitalWrite(d, LOW);
  digitalWrite(e, LOW);
  digitalWrite(f, LOW);
  digitalWrite(g, LOW);
}

void loop(){
  buttonState1 = digitalRead(button1);
  buttonState2 = digitalRead(button2);
  if (buttonState1 == HIGH) {
    ledReset();
    Serial.println("yes1");
    digitalWrite(13, HIGH);
    ledStatus++;

    switch (ledStatus){
      case 1:
        digitalWrite(LED1, HIGH);
```

```
        break;
      case 2:
        digitalWrite(LED1, HIGH);
        delay(500);
        digitalWrite(LED1, LOW);
        delay(500);
        digitalWrite(LED1, HIGH);
        break;
      case 3:
        ledsBlink();
        break;
      case 4:
        ledStatus = 0;
        break;
    }
  }
  if (buttonState2 == HIGH){
    Serial.println("yes2");
    digitalWrite(13, HIGH);
    clearDisplay();
    counter++;
    switch (counter){
      case 1:
        digitalWrite(a, HIGH);
        digitalWrite(b, HIGH);
        digitalWrite(c, HIGH);
        break;
      case 2:
        digitalWrite(a, HIGH);
        digitalWrite(b, HIGH);
        digitalWrite(g, HIGH);
        digitalWrite(c, HIGH);
        digitalWrite(d, HIGH);
        digitalWrite(e, HIGH);
        digitalWrite(f, HIGH);
        break;
      case 3:
        digitalWrite(a, HIGH);
        digitalWrite(b, HIGH);
        digitalWrite(g, HIGH);
        digitalWrite(e, HIGH);
        digitalWrite(d, HIGH);
        break;
      case 4:
        digitalWrite(f, HIGH);
        digitalWrite(b, HIGH);
        digitalWrite(g, HIGH);
        digitalWrite(c, HIGH);
        break;
      case 5:
        clearDisplay();
        counter = 0;
        break;
    }
  }
}
```