

**ДОДАТОК А****Лістинг програми для керування ультрафіолетовою камерою для полімеризації**

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
const int motor1Pin = 9;
const int motor2Pin = 10;
const int ledPin = 5;
const int button1Pin = 2;
const int button2Pin = 3;
const int button3Pin = 4;
unsigned long startTime;
unsigned long countdownTime = 0;
unsigned long remainingTime = 0;
unsigned long lastInteractionTime = 0;
const unsigned long interactionTimeout = 10000;
enum State {
    WELCOME,
    COUNTDOWN,
    IDLE,
    RUNNING,
    STOPPED
};
State currentState = WELCOME;
void setup() {
    pinMode(motor1Pin, OUTPUT);
    pinMode(motor2Pin, OUTPUT);
```

```
pinMode(ledPin, OUTPUT);
pinMode(button1Pin, INPUT_PULLUP);
pinMode(button2Pin, INPUT_PULLUP);
pinMode(button3Pin, INPUT_PULLUP);
lcd.init();
lcd.backlight();
showWelcomeMessage();
startTime = millis();
}
void loop() {
  analogWrite(motor1Pin, 255);
  if (currentState == WELCOME) {
    if (digitalRead(button1Pin) == LOW || digitalRead(button2Pin) == LOW) {
      currentState = COUNTDOWN;
      lastInteractionTime = millis();
      updateLCD();
    }
  } else if (currentState == COUNTDOWN) {
    handleButtons();
    if (millis() - lastInteractionTime >= interactionTimeout) {
      currentState = IDLE;
      showIdleMessage();
    }
  } else if (currentState == IDLE) {
    if (digitalRead(button1Pin) == LOW || digitalRead(button2Pin) == LOW) {
      currentState = COUNTDOWN;
      lastInteractionTime = millis();
      updateLCD();
    }
  }
}
```

```

} else if (currentState == RUNNING) {
    handleRunningState();
} else if (currentState == STOPPED) {
    handleStoppedState();
}
if (digitalRead(button3Pin) == LOW && (currentState == COUNTDOWN ||
currentState == RUNNING || currentState == STOPPED)) {
    if (currentState == COUNTDOWN || currentState == STOPPED) {
        startTime = millis();
        currentState = RUNNING;
        digitalWrite(ledPin, HIGH);
        analogWrite(motor2Pin, 255);
        countdownTime = remainingTime;
    } else if (currentState == RUNNING) {
        remainingTime = countdownTime - (millis() - startTime);
        if (remainingTime < 0) remainingTime = 0;
        stopSecondSystem();
        currentState = STOPPED;
    }
    delay(200);
}
}

void handleButtons() {
    if (digitalRead(button1Pin) == LOW) {
        countdownTime += 10000;
        remainingTime += 10000;
        delay(200);
        updateLCD();
        lastInteractionTime = millis();
    }
}

```

```

}
if (digitalRead(button2Pin) == LOW) {
  if (countdownTime >= 10000) {
    countdownTime -= 10000;
    remainingTime -= 10000;
    if (remainingTime < 0) remainingTime = 0;
  } else {
    countdownTime = 0;
    remainingTime = 0;
  }
  delay(200);
  updateLCD();
  lastInteractionTime = millis();
}
}

void updateLCD() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Time: ");
  lcd.setCursor(6, 0);
  lcd.print((currentState == STOPPED) ? remainingTime / 1000 :
countdownTime / 1000);
  lcd.print(" s");
}

void showWelcomeMessage() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Hello, User!");
}

```

```
void showIdleMessage() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("User, are you");
    lcd.setCursor(0, 1);
    lcd.print("here?");
}

void handleRunningState() {
    if (millis() - startTime >= countdownTime) {
        stopSecondSystem();
        currentState = COUNTDOWN;
        updateLCD();
    } else {
        remainingTime = (countdownTime - (millis() - startTime)) / 1000;
        lcd.setCursor(6, 0);
        lcd.print(remainingTime);
        lcd.print(" s ");
    }
}

void handleStoppedState() {
    if (digitalRead(button1Pin) == LOW) {
        remainingTime += 10000;
        countdownTime = remainingTime;
        delay(200);
        updateLCD();
        lastInteractionTime = millis();
    }
    if (digitalRead(button2Pin) == LOW) {
        if (remainingTime >= 10000) {
```

```
    remainingTime -= 10000;
  } else {
    remainingTime = 0;
  }
  countdownTime = remainingTime;
  delay(200);
  updateLCD();
  lastInteractionTime = millis();
}
lcd.setCursor(6, 0);
lcd.print(remainingTime / 1000);
lcd.print(" s ");
}
void stopSecondSystem() {
  analogWrite(motor2Pin, 0);
  digitalWrite(ledPin, LOW);
}
```

## **ДОДАТОК Б**

**Демонстраційний матеріал у вигляді презентації**

