+

**Efe Can Kara**

**2022481031**

**Potansiyometre Kontrollü Pomodoro Zamanlayıcısı**

**Proje kodları:**

// Gerekli kütüphanelerin tanımlanması

#include <Wire.h> // LCD için haberleşmeyi sağlayan I2C protokolü için gerekli kütüphane

#include <Adafruit\_LiquidCrystal.h> // Devrede kullanılan LCD'nin kullanımı için gerekli kütüphane

// Durum Sıraları Tanımlanması

const int POMODORO = 0;

const int SHORT\_BREAK = 1;

const int LONG\_BREAK = 2;

// Durum Sürelerinin Tanımlanması

const int POMODORO\_DURATION = 25 \* 60;

const int SHORT\_BREAK\_DURATION = 5 \* 60;

const int LONG\_BREAK\_DURATION = 15 \* 60;

// Arduino Pin Girişlerinin Tanımlanması

const int buzzerPin = 2;

const int potentiometerPin = A0;

Adafruit\_LiquidCrystal lcd\_1(0);

// Başlangıç Durumunun Belirlenmesi

int currentState = POMODORO;

long seconds = 0;

bool timerRunning = false;

int pomodoroCount = 0; // Pomodoro sayısını tutan değişken

void setup() {

lcd\_1.begin(16, 2);

pinMode(buzzerPin, OUTPUT);

}

void loop() {

lcd\_1.setCursor(0, 0);

lcd\_1.print(" ");

lcd\_1.setCursor(0, 1);

lcd\_1.print(" ");

lcd\_1.setCursor(0, 0);

switch (currentState) {

case POMODORO:

lcd\_1.print("Pomodoro");

break;

case SHORT\_BREAK:

lcd\_1.print("Kısa Mola");

break;

case LONG\_BREAK:

lcd\_1.print("Uzun Mola");

break;

}

lcd\_1.setCursor(0, 1);

lcd\_1.print(formatTime(seconds));

int potValue = analogRead(potentiometerPin);

int potThreshold = 500;

if (potValue > potThreshold) {

delay(50);

timerRunning = !timerRunning;

if (timerRunning) {

digitalWrite(buzzerPin, HIGH);

delay(50);

digitalWrite(buzzerPin, LOW);

}

while (analogRead(potentiometerPin) > potThreshold) {

delay(10);

}

}

if (timerRunning) {

delay(1000);

if (seconds >= getStateDuration(currentState)) {

seconds = 0;

switchState();

}

seconds += 1;

}

}

// Durum Süresini Almak İçin Fonksiyon

int getStateDuration(int state) {

switch (state) {

case POMODORO:

return POMODORO\_DURATION;

case SHORT\_BREAK:

return SHORT\_BREAK\_DURATION;

case LONG\_BREAK:

return LONG\_BREAK\_DURATION;

}

return 0;

}

// Durumu Değiştirmek İçin Fonksiyon

void switchState() {

digitalWrite(buzzerPin, HIGH);

delay(500);

digitalWrite(buzzerPin, LOW);

switch (currentState) {

case POMODORO:

pomodoroCount++;

if (pomodoroCount < 4) {

currentState = SHORT\_BREAK;

} else {

currentState = LONG\_BREAK;

pomodoroCount = 0; // 4 Pomodoro sonrası Long Break başlar, Pomodoro sayısı sıfırlanır.

}

break;

case SHORT\_BREAK:

currentState = POMODORO;

break;

case LONG\_BREAK:

currentState = POMODORO;

break;

}

}

// Zamanı Biçimlendirmek İçin Fonksiyon

String formatTime(long seconds) {

int minutes = seconds / 60;

int remainingSeconds = seconds % 60;

return String(minutes) + "m " + String(remainingSeconds) + "s";

}

**Projenin Drive Linki:**

https://drive.google.com/file/d/1juk3E2mNvLsJ--PmqegYRdz4S8zENxdC/view?usp=sharing

**Projenin Fotoğrafı (Tinkercad):**



