

The background features a series of parallel, slanted neon lines in shades of green, blue, and purple, creating a sense of depth and movement. On the left side, there is a vertical bar with a blue-to-purple gradient.

# **Gomulu Sistemler**

## **Dijital Saat Projesi**

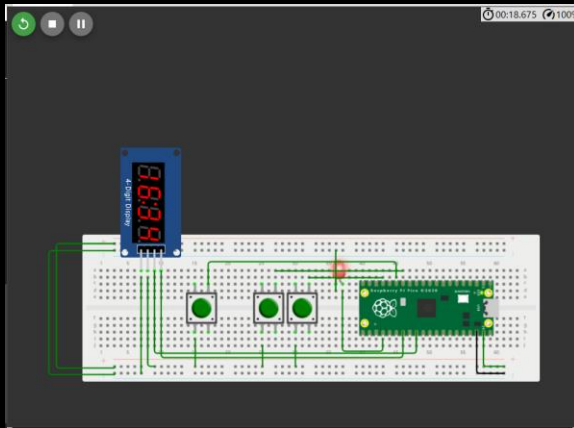
**Hazirlayan:**

**Ilyas Can Turali - 213405517**

# Projede Kullandigim Malzemeler ve Maaliyetleri

- BreadBoard / Jumpers / Leds - 50₺
- 4'lü 7-Segment TM1637 Kırmızı Display Modulu - 32,47₺
- Raspberry Pi Pico - 125₺
- Duracell Alkalın 9V Pil - 85,02₺
  
- **Toplam Maaliyet = 292,49₺**

# Projenin Planlama Evresi



Simulasyon'da BreadBoard üzerinde devre elemanlarının hazırlanması.

Kod'un yazılarak Simulasyon üzerinde test edilmesi

```
main.py  diagram.json  tm1637.py
1  import tm1637
2  from machine import Pin
3  from utime import sleep
4  import time
5
6  mydisplay = tm1637.TM1637(clk=Pin(21), dio=Pin(22))
7  brightness = 1
8
9  minute_btn_pin = Pin(11, Pin.IN, Pin.PULL_DOWN)
10 minute_prv_state = minute_btn_pin.value()
11
12 hour_btn_pin = Pin(10, Pin.IN, Pin.PULL_DOWN)
13 hour_prv_state = hour_btn_pin.value()
14
15 edit_btn_pin = Pin(11, Pin.IN, Pin.PULL_DOWN)
16 edit_mode = pin(14, Pin.OUT)
17
18 led_pin = Pin(25, Pin.OUT, Pin.PULL_DOWN)
19
20 debounce_delay = 50 # [ms] delay in milliseconds
21
22 def minute_btn_clicked(pin):
23     global minute_prv_state, h, m
24     minute_cur_state = pin.value()
25
26     # Check if the button state has changed
27     if edit_mode.value() == 1:
28         if minute_cur_state != minute_prv_state:
29             minute_prv_state = minute_cur_state
30             if minute_prv_state == 1:
31                 m += 1
32                 if m > 59:
33                     m = 0
34                     mydisplay.numbers(h, m)
35             else: minute_prv_state = 0;
36     else:
37         if minute_cur_state != minute_prv_state:
38             minute_prv_state = minute_cur_state
39             if minute_prv_state == 1:
40                 global brightness
41                 brightness = brightness + 1 if brightness < 7 else 1
42                 mydisplay.brightness(brightness)
43             else: minute_prv_state = 0;
44
45     sleep(debounce_delay / 1000) # Debounce delay
46
```

Projenin kodlanmasında **MicroPython** kullanıldı. TM1637 kutuphanesi eklenerek 7-Segment Display'in kontrolleri sağlandı.

7-Segmente DIO girişinden veri akışı sağlandı.

Projedeki Butonlar Saati, Dakikayı ve Parlaklığı ayarlamak için kullanıldı. Bir butona ise düzenleme işlevi atandı.

Butonlara interruptlar atanıp, kullanıldı.

# Projede Zorlandigim Kisimler ve Buldugum Cozumler

**Butonların Birden Fazla Çalışması:** Tek bir vuruşta buton birden fazla çalışıyordu. Bunu çözmek için butonun önceki değerini kaydettim ve önceki değer ile yeni değer eğer farklı ise Buton tetiklendi, ayrıca "Debounce delay" denilen gecikme eklendi böylece butonun tek seferde tetiklenmesi sağlandı.

# Projede Zorlandigim Kisimler ve Buldugum Cozumler

**Butonların Algılanmaması:** Eski deger, yeni degere atandiginda butonun birden fazla calismasi engellenmisti ancak sonraki yapilan tetiklemelerde yeni deger ve eski deger ayni oldugundan dolayi buton tekrar tetiklenemiyordu. "Else" alaninda eski degeri 0'a esitleyerek problem cozuldu.

Simulasyonda calisan bu kisim Gercek projede sikinti cikarmistir.

# Projede Zorlandigim Kisimler ve Buldugum Cozumler

**Performans Dusuklugu:** Raspberry Pi Pico'yu while dongusune sokunca yavaslama ve zamanda saptama gorunuyordu. Her dongu sonunda 250 ms uyutarak "sleep" bu sorun cozuldu.