

ISI_NEM.py

```
1. from machine import Pin, I2C
2. import utime as time
3. from dht import DHT11
4. import utime
5.
6. from lcd_api import LcdApi
7. from pico_i2c_lcd import I2cLcd
8.
9. I2C_ADDR = 0x27
10. I2C_NUM_ROWS = 2
11. I2C_NUM_COLS = 16
12.
13. sda = Pin(20)
14. scl = Pin(21)
15.
16. i2c = I2C( , sda=sda, scl=scl, freq=200000)
17. lcd = I2cLcd(i2c, I2C_ADDR, I2C_NUM_ROWS,
18. I2C_NUM_COLS)
19. pin = Pin(18, Pin.OUT, Pin.PULL_DOWN)
20. sensor = DHT11(pin) # sensor olarak DHT11 tanımlanıyor
21. # landı. GP18'e bağlı.
22. lcd.move_to(0, 0) # kursor 0.sütun 0.satıra 51
```

```

23. lcd.putstr("DHT 11") # DHT 11 yaz
24. utime.sleep(2) # 2sn bekle
25. lcd.move_to(5, 1) # kursor 5.sütun 1.satıra
26. lcd.putstr("SENSOR OK") # Sensor OK yaz.
27. utime.sleep(2) # 2sn bekle
28. lcd.clear() # Lcd'yi sil.
29.
30. while True:
31.     sensor.measure() # ölçüm yap
32.     print("SICAKLIK: {}, NEM{}".format(sensor.
33.     temperature, sensor.humidity))
34.     time.sleep(1) # 1sn bekle
35.
36. lcd.move_to(0, 0) # kursor 0.sütun 0.satıra
37. lcd.putstr("SICAKLIK: {} NEM{}:".format(
    sensor.temperature, sensor.humidity))

```

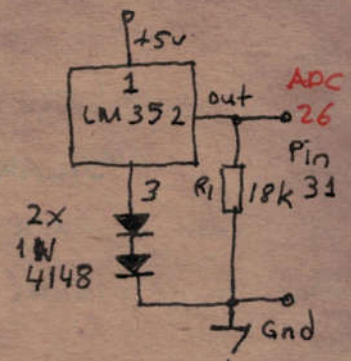
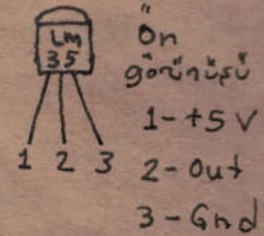
Yukarıdaki program DHT11 sensörü GP18'e bağlı olduğunda ve LCD-IZC'nin sda pini GP20'ye ve scl pininininde GP18'a bağlandığında önce LCD ekranında "DHT 11" ve alt satırında "SENSOR OK" yazdıktan sonra ekranda SICAKLIK: 30.3 ve alt satırda NEM: 32.0 yazar. Aynı anda shell kısmında da SICAKLIK:30.3, NEM: 32.0 yazar.

LM35-LS1.py

```

1. from machine import ADC
2. from machine import Pin
3. import time
4.
5. DIODE_OFFSET_VOLTAGE = 0.979
6. led_blue = Pin(2, Pin.OUT)
7. led_red = Pin(16, Pin.OUT)
8.
9. analogInputPin = ADC(26)
10.
11.
12. while True:
13.
14.     analogValue = ADC.read(26)
15.
16.     sensor_voltage = (analogValue / 65535) * 3.3
17.
18.     sensor_voltage = (sensor_voltage - DIODE_OFFSET_VOLTAGE)
19.     * 1000
20.     temperature = (sensor_voltage / 10) # °C 15151.
21.     time.sleep(1)
22.

```



```
23. print("SICAKLIK=", temperature, "C Derece")
24. print("\n") # sıcaklığı aynı satırda göster.
25.
26. if temperature < 30:
27.     led-blue.on()
28. else
29.     led-blue.off()
30. if temperature > 31:
31.     led-red.on()
32. else
33.     led-red.off()
```

Yukarıdaki program LM35 analog ısı sensörü LM35 bağlantısı kurularak out pini ADC 26 pin 31'e bağlanarak ve +5V ile beslenerek GP2'ye bağlı olan led-blue (mavi) ledi 30 derecenin altındaki sıcaklıklarda yakar. GP16'ya bağlı olan led-red (kırmızı) ledi 31 derecenin üstündeki sıcaklıklarda yakar. Shell kısmında ise SICAKLIK = 29.35174 C derece yazar.

RGB_Control.py

```
1. from machine import Pin
2. from mfrc522 import MFR522
3. import utime
4. # SCK=6, miso=4, mosi=7, cs=Sda=5, rst=22 (GPiN)
5. reader = MFR522(spi=0, sck=6, miso=4, mosi=
6. 7, cs=5, rst=22) # cs pini sda pini demektir
7. red = Pin(0, Pin.OUT) # R pini GP0'a baęli
8. green = Pin(1, Pin.OUT) # G pini GP1'e baęli
9. blue = Pin(2, Pin.OUT) # B pini GP2'ye baęli.
10.
11. print("RFID kartını yaklařtırın...")
12. print("")
13.
14. while True:
15.     reader.init() # MFR522 kullanıma hazırlanır.
16.     (stat, tag_type) = reader.request(reader.REQIDL)
17.     if stat == reader.OK:
18.         (stat, uid) = reader.selectTagSN()
19.         if stat == reader.OK:
20.             card = int.from_bytes(bytes(uid), "little",
21.             False)
22.             if card == 1942487079:
```

```

23. print("Card ID:" + str(card) + "ONAY: Yeşil Işık
24. yakıldı.")
25. red.value(0) # kırmızı ledi söndür
26. green.value(1) # Yeşil ledi yak.
27. blue.value(0) # Mavi ledi söndür.
28.
29. elif card == 3545020779:
30.     print("Card ID:" + str(card) + "ONAY: Mavi Işık
31.     yakıldı.")
32.     red.value(0) # kırmızı ledi söndür
33.     green.value(0) # Yeşil ledi söndür.
34.     blue.value(1) # Mavi ledi söndür.
35.
36. else: # Bu iki kartın dışında ise
37.     print("Card ID:" + str(card) + "BİLİNMEYEN
38.     KART! Kırmızı Işık Yakıldı")
39.     red.value(1) # kırmızı ledi yak
40.     green.value(0) # Yeşil ledi söndür.
41.     blue.value(0) # Mavi ledi söndür.

```

Yukarıdaki program RGB ledin R pini GP0'a, G pini GP1'e, B pini de GP2'ye bağlandıktan sonra MFRC522'nin pinleri olan MOSI GP4'e, MISO GP4'e, SDA GP5'e, SCK GP6'ya ve RST ise GP22'ye bağlanır. Shell kısmında "Card ID: 'den sonra numarası yazar. Eğer card ID biliniyorsa BİLİNMEYEN kart yazar. Kırmızı led yanar. Bilinirse yeşil led yanar. 56

LCD-saat.py

```
1. from machine import I2C, Pin
2. from ds1302 import DS1302
3. from pico-i2c-lcd import I2CLcd
4.
5. I2C_ADDR = 0x27
6. I2C_NUM_ROWS = 2
7. I2C_NUM_COLS = 16
8. # LCD'nin SDA pinini GP0'a ve SCL pinini de GP1'e bağlanır
9. i2c = I2C(0, sda=machine.Pin(0), scl=machine
10. .Pin(1), freq=400000)
11. lcd = I2CLcd(i2c, I2C_ADDR, I2C_NUM_ROWS,
12. I2C_NUM_COLS)
13. ds = DS1302(Pin(18), Pin(17), Pin(16))
14. # Rtc tipi DS1302, GP18=CLK, GP17=Data, GP16=Rst
15. ds.date_time() # Gerçek zaman tarih ve saat
16. # Alttaki satır önündeki # kaldırılarak çalışınca
17. # tarih saat set yapılmalıdır. Sonra tekrar # yazılır.
18. # ds.date_time([2024, 7, 30, 20, 44, 40, 0, 0])
19. # ds.hour()
20. print(ds.date_time()) # shelde tarih saati yaz.
21.
22.
```

```

23.
24. while True:
25.     (Y, M, D, day, hr, m, s) = ds.date_time()
26.     if s < 10:
27.         s = "0" + str(s)
28.     if m < 10:
29.         m = "0" + str(m)
30.     if hr < 10:
31.         hr = "0" + str(hr)
32.     if D < 10:
33.         D = "0" + str(D)
34.     if M < 10:
35.         M = "0" + str(M)
36.
37.     lcd.move_to(0, 0)
38.
39.     lcd.putstr("Time:" + str(hr) + ":" + str(s))
40.     lcd.move_to(0, 1)
41.     lcd.putstr("Date:" + str(D) + "/" + str(M) + "/" +
    str(Y))

```

Yukarıdaki program DS1302'nin CLK pini GP18'e, Date pini GP17 ve Rst pini de GP16'ya bağlandıktan sonra LCD ekranda DS1302'den gelen date_time() bilgisine göre Time: 21:24:58 ve alt satırda ise tarih bilgisi Date: 30/07/2024 şeklinde yazar. Sn bilgisi değişir. 58

main.py

RFID-RGB.py

```
1. from machine import Pin, SPI
2. from mfr522 import MFRC522
3. from time import sleep
4. 
5. 
6. buzzer = Pin(20, Pin.OUT) # cs=sda=5 (GPS)
7. reader = MFRC522(spi_id=0, sck=6, miso=4, mosi=7,
8. cs=5, rst=22)
9. red = Pin(10, Pin.OUT)
10. green = Pin(11, Pin.OUT)
11. blue = Pin(12, Pin.OUT)
12. 
13. print("RFID kartını yaklaştırmış...")
14. print("")
15. 
16. while True:
17.     reader.init()
18.     (stat, tag_type) = reader.request(reader.REQIDL)
19.     if stat == reader.OK:
20.         (stat, uid) = reader.SelectTagSN()
21.         if stat == reader.OK:
22.             card = int.from_bytes(bytes(uid), "little", False) 59
```

```
23.  
24. if card == 1942487079 :  
25.     print("CARD ID: + str(card) + "ONAY: Yeşil  
26.     Işık Yakıldı")  
27.     red.value(0)  
28.     green.value(1)  
29.     blue.value(0)  
30.     buzzer.on()  
31.     sleep(0.5)  
32.     buzzer.off  
33. elif card == 3545020779 :  
34.     print("CARD ID: + str(card) + "ONAY: Mavi  
35.     Işık Yakıldı")  
36.     red.value(0)  
37.     green.value(0)  
38.     blue.value(1)  
39.     buzzer.on()  
40.     sleep(0.5)  
41.     buzzer.off()  
42. else :  
43.     print("CARD ID: + str(card) + "BİLİNMEYEN  
44.     KART! Kırmızı Işık Yakıldı.")
```

45. red.value (1)
46. green.value (0)
47. blue.value (0)
48. buzzer.on()
49. sleep (1.5)
50. buzzer.off

Yukarıdaki program MFRC522 Rfid kart okuyucunun MOSI pinini GP4'ye, SDA pinini GP5'e, SCK pinini GP6'ya, MISO pinini de GP7'ye bağlanıp, buzzer GP20'ye ve RGB pinleri green (yeşil) GP11, red (kırmızı) GP10 ve en sonunda GP12 'de blue (mavi) 'ye bağlandığında tanımlanmış olan RFID card'lar ile yeşil led veya mavi led yanarak bip sesi çıkarır. Eğer tanımlanmamış RFID kart yada telefondaki NFC açık iken okuma yapılırsa kırmızı led yanar ve bip sesi çıkartır.

RGB_show.py

```
1. from machine import Pin
2. import utime
3.
4. led1 = Pin(10, Pin.OUT) # RGB kırmızı ledi
5. led2 = Pin(11, Pin.OUT) # RGB yeşil ledi
6. led3 = Pin(12, Pin.OUT) # RGB mavi led
7.
8. LED_state1 = True # LED durumu1 = 1
9. LED_state2 = False # LED durumu2 = 0
10.
11. while True:
12.     led1.value(LED_state1) # 1. durum kırmızı
13.     led2.value(LED_state2)
14.     led3.value(LED_state2)
15.     print("KIRMIZI RENK YANDI")
16.     utime.sleep(5)
17.
18.     led1.value(LED_state2) # 2. durum yeşil
19.     led2.value(LED_state1)
20.     led3.value(LED_state2)
21.     print("YESIL RENK YANDI")
22.     utime.sleep(5)
```

```
23.  
24. led1.value(LED_state1) # 3. durum sarı  
25. led2.value(LED_state1)  
26. led3.value(LED_state2)  
27. print("SARI RENK YANDI")  
28. utime.sleep(5)  
29.  
30. led1.value(LED_state2) # 4. durum mavi  
31. led2.value(LED_state2)  
32. led3.value(LED_state1)  
33. print("MAVI RENK YANDI")  
34. utime.sleep(5)  
35.  
36. led1.value(LED_state1) # 5. durum pembe  
37. led2.value(LED_state2)  
38. led3.value(LED_state1)  
39. print("PEMBE RENK YANDI")  
40. utime.sleep(5)  
41.  
42. led1.value(LED_state2) # 6. durum turkuaz  
43. led2.value(LED_state1)  
44. led3.value(LED_state1)  
45. print("TURKUAZ RENK YANDI")
```

```
46. utime.sleep(5)
47.
48. led1.value(LED_state1)
49. led2.value(LED_state1)
50. led3.value(LED_state1)
51. print("BEYAZ RENK YANDI")
52. utime.sleep(5)
```

Yukarıdaki program GP10 pinine bağlı red(kırmızı), GP11 pinine bağlı yeşil(green) ve GP12 pinine bağlı olan mavi(blue) RGB ledini 5'er saniye aralıklarla önce kırmızı, sonra sırasıyla yeşil, sarı, mavi, pembe, turkuaz ve en sonunda beyaz olarak yakar. Shell kumanda ise aşağıdaki gibi yazılar yazar

Shell X

```
KIRMIZI RENK YANDI
YESIL RENK YANDI
SARI RENK YANDI
MAVI RENK YANDI
PEMBE RENK YANDI
TURKUAZ RENK YANDI
BEYAZ RENK YANDI
KIRMIZI RENK YANDI
YESIL RENK YANDI
SARI RENK YANDI
```

```
1. from machine import I2C, Pin
2. from ds1302 import DS1302
3. from pico-i2c-lcd import I2cLcd
4. import utime
5.
6. led_red = Pin(10, Pin.OUT)
7. led_green = Pin(11, Pin.OUT)
8. led_blue = Pin(12, Pin.OUT)
9.
10. sıcaklik_sensory = machine.ADC(4)
11. sıcaklik_cevirme = 3.3/(65535)
12.
13. I2C_ADDR = 0x27 # LCD çalışmazsa 63 girin.
14. I2C_NUM_ROWS = 2 # LCD'nin satır sayısı
15. I2C_NUM_COLS = 16 # LCD'nin karakter sayısı
16.
17. i2c = I2C(0, sda=machine.Pin(0), scl=machine
18. Pin(1), freq=400000)
19.
20. ds = DS1302(Pin(18), Pin(17), Pin(16))
21. ds.date_time()
22. ## ds.date_time([2024, 7, 30, 20, 44, 40, 0, 0])
```

```

24. print (ds.date_time())
25.
26. while True:
27.     okunan_sicaklik = sicaklik_sensoru.read_u16()
28.     * sicaklik_cevirme
29.     sicaklik = 27 - (okunan_sicaklik - 0.706) / 0.0
30.     1722
31.     sicaklik = (sicaklik - 2)
32.     print (sicaklik) # sicaklik degerini yazdırır.
33.     utime.sleep(2) # 2 sn bekle
34.
35. (Y,M,D,day,hr,m,s) = ds.date_time()
36. if s < 10: # saniye 10'dan küçük ise
37.     s = "0" + str(s)
38. if m < 10: # dakika 10'dan küçük ise
39.     m = "0" + str(m)
40. if hr < 10: # saat 10'dan küçük ise
41.     hr = "0" + str(hr)
42. if D < 10: # Day of week 10'dan küçük ise
43.     D = "0" + str(D)
44. if M < 10: # Ay 10'dan küçük ise

```

```
45. M = "0" + str(M)
46.
47. lcd.move-to(0,0)
48. lcd.putstr("Time:" + str(hr) + ":" + str(m))
49. lcd.move-to(12,0)
50. lcd.putstr(str(sicaklik))
51. lcd.move-to(14,0)
52. lcd.putstr(" C")
53. lcd.move-to(0,1)
54. lcd.putstr("Date:" + str(D) + "/" + str(M) +
55. "/" + str(Y))
56. if sicaklik >= 28:
57.     led-red.value(0) # kırmızı ledi söndür
58.     led-green.value(1) # yeşil ledi yak
59.     led-blue.value(0) # mavi ledi söndür
60. if sicaklik >= 29:
61.     led-red.value(0) # kırmızı ledi söndür
62.     led-green.value(1) # yeşil ledi yak
63.     led-blue.value(1) # mavi ledi yak
64. if sicaklik >= 30:
65.     led-red.value = (1) # kırmızı ledi yak
66.     led-green.value(0) # yeşil ledi söndür
67.     led-blue.value(1) # mavi ledi yak.
```

68. **if** sıcaklık $\geq$ 31:

69. led-red.value(1) # kırmızı led'i yak

70. led-green.value(0) # yeşil led'i söndür

71. led-blue.value(0) # mavi led'i söndür

Yukarıdaki program LCD12C'nin sda pini GP0'a, scl pini ise GP1'e bağlandıktan sonra RGB led'in kırmızı olanı GP10'a, yeşili GP11'e ve mavisi ise GP12'ye bağlanınca LCD ekranında üst satırda "Time: 17:35:06" ve alt satırında ise "Date: 02/08/2024" şeklinde yazar. Aynı zamanda içindeki ADC(4)'e bağlı sıcaklık sensöründen aldığı sıcaklık değerini yazdıktan sonra "C" işareti yapar.

4x7segment_ds1302.py

```
1. from machine import Pin, I2C, ADC
2. import tm1637
3. import time
4. import ds1302
5.
6. adcPin = 4
7. sensor = machine.ADC(adcPin)
8. led_red = Pin(10, Pin.OUT)
9. led_green = Pin(11, Pin.OUT)
10. led_blue = Pin(12, Pin.OUT)
11. buzzer = Pin(20, Pin.OUT)
12.
13. mdisplay = tm1637.TM1637(clk=Pin(0), dio=Pin
14. (1))
15. ds = ds1302(Pin(18), Pin(17), Pin(16))
16. # DS1302'nin 18.pini CLK, 17.pini DATA ve 16.pini ise RST
17. # ds.date-time([2024, 8, 2, 8, 17, 15, 0]) # Tarih
18. # saat ayarı için # silinerek kaydedildikten sonra
19. # öñüne tekrar # konularak çalıştırılacaktır.
20. def ReadTemperature():
```

```

21. adc_value = sensor.read_u16()
22. volt = (3.3/65535)*adc_value
23. temperature = 27-(volt-0.706)/0.00172
24. return round(temperature,1)
25.
26. while True:
27.
28. (Y,M,D,day,hr,m,s) = ds.date_time()
29.
30. yil1 = 20
31. yil2 = 24
32.
33. mdisplay.numbers(D,M)#Displayde day:month
34. time.sleep(1) # 1sn bekle
35.
36. mdisplay.numbers(yil1,yil2)#Displayde 20:24
37. time.sleep(1) # 1sn bekle
38.
39. mdisplay.numbers(hr,m)#Displayde hour:minute
40. time.sleep(2) # 2 sn bekle
41.
42. temperature = ReadTemperature()

```

```
43. print(str(temperature)+" C")
44. time.sleep(1)
45.
46. if temperature >= 29: # yeşil renk yanar.
47.     led_red.value(0)
48.     led_green.value(1)
49.     led_blue.value(0)
50.
51. if temperature >= 30: # Mor renk yanar.
52.     led_red.value(1)
53.     led_green.value(0)
54.     led_blue.value(1)
55.
56. if temperature >= 31: # kırmızı renk yanar
57.     led_red.value(1)
58.     led_green.value(0)
59.     led_blue.value(0)
60.
61. if hr==10 and m==30: # saat 10:30 ise
62.     buzzer.value(1) # Buzzer'ı öttür
63.     time.sleep(2) # 2sn bekle
64.     buzzer.value(0) # Buzzer'i sustur.
65.     time.sleep(2) # 2sn bekle
```

66. buzzer.value(1) # Buzzer'i öttür.
67. time.sleep(2) # 2sn bekle
68. buzzer.value(0) # Buzzer'i sustur.

Yukarıdaki program 7 segmentli 4 digitten oluşan TM1637'nin clk pini GP0'a ve Dio pini de GP1'e bağlanınca, RGB pinleri kırmızı GP10, yeşil GP11, mavi GP12'ye bağliken digitlerde sırasıyla; önce gün:ay, 1sn sonra yıl1;yıl2 daha sonra saat:dakika gösterir. Bu sırada yıl,gün,ay bilgilerini ds1302 kartından okuyarak gösterir. Raspberry pi'co pin'in built-in olan ADC4 sıcaklık algılayıcısından aldığı sıcaklık (temperatura) değeriyle sorgulama yaparak sıcaklık 29°C'den büyükse RGB ledi yeşil, 30°C'den büyükse mor ve 31°C'den büyükse kırmızı olarak yanar. Eğer saat bilgisi "10" ve dakika bilgisi "30" ise bağlı olan buzzer (GP20) 1 dakika boyunca 2sn aralıklarla öter ve susar.

KEYPAD_sifre.py

```
1. from machine import Pin
2. import utime
3.
4. # keypad buton ve karakter haritası
5. matrix_keys = [ ['1', '2', '3', 'A'],
6.                  ['4', '5', '6', 'B'],
7.                  ['7', '8', '9', 'C'],
8.                  ['*', '0', '#', 'D']]
9. # Kablo bağlantı tanımlamaları
10. keypad_rows = [10, 9, 8, 7] # satırlar
11. keypad_columns = [6, 5, 4, 3] # sütunlar
12. # tuşlara basan kişi
13. guess = [ ]
14. col_pins = [ ]
15. row_pins = [ ]
16. # şifre bilgileri buradan değiştirilebilir.
17. secret_pin = ['1', '9', '2', '3']
18. # led ve role pinlerinin bağlantı noktaları.
19. led = Pin(19, Pin.OUT)
20. role = Pin(2, Pin.OUT, Pin.PULL_UP)
```

```

21. role.value(0) # role durdu
22. led.value(0) # led sönü
23. buzzer = Pin(20, Pin.OUT)
24.
25. for x in range(0, 4)
26.     row_pins.append(Pin(keypad_rows[x], Pin-
27.     OUT))
28.     row_pins[x].value(1)
29.     col_pins.append(Pin(keypad_columns[x], Pin
30.     IN, Pin.PULL_DOWN))
31.     col_pins[x].value(0)
32.     ##### scan keys ##### tarama yap
33.
34.     def scankeys():
35.
36.         for row in range(4):
37.             for col in range(4):
38.                 row_pins[row].high() # satırlar 1 yap.
39.                 key = None
40.
41.         if col_pins[col].value() == 1:

```

```
42. print ("Basılan Tuş:", matrix_keys[row][col])
43. key_press = matrix_keys[row][col]
44. buzzer.value(1) # Tuşa basılınca buzzer öter
45. utime.sleep(0.3) # 300 ms sn sıçrama için bekle
46. buzzer.value(0) # Elimizi çekince buzzer susar
47. guess.append(key_press)
48.
49. if len(guess) == 4:
50.     checkPin(guess)
51.
52.     for x in range(0, 4)
53.         guess.pop()
54.
55. row_pins[row].low() # satırları 0 yapar.
56.
57. ##### pinleri kontrol et #####
58. def checkPin(guess)
59.
60.     if guess == secret_pin: # Eğer şifre doğru.
61.
62.         print("Tebrیکler girilen şifre doğru...")
63.         led.value(0) # ledi söndür
64.         role.value(1) # roleyi çalıştır
```

```

65. utime.sleep(2) # 2sn bekle
66. role.value(0) # roleyi durdur.
67.
68. else :
69.     print("Yanlış şifre girdiniz!...")
70.     role.value(0) # roleyi durdur.
71.     led.value(1) # ledi yak
72.     utime.sleep(2) # 2sn bekle
73.     led.value(0) # ledi söndür
74.
75.     print("Şifreyi Giriniz...")
76.
77. while True:
78.     scankeys() # Tuşları devamlı tara

```

Yukarıdaki program keypad 4x4'ün satır pinleri sıra ile GP10, GP9, GP8, GP7'ye bağlandıktan sonra sütunlar GP6, GP5, GP4, GP3'e bağlandığında, GP19'da bağlı olan ledi ve GP2'de bağlı olan roleyi kontrol eder. Yani roleyi kontrol eder. Shell ekranında "Şifreyi giriniz" yazar. Eğer 1923 bilgisi yazılırsa "Tebrikler girilen şifre doğru...." yazar. Bu arada tuşlara basılırken GP20'ye bağlı buzzer öter. Eğer şifre yanlış girilirse shell ekranında "Yanlış şifre girdiniz!..." yazar.

7segment0k.py

```

1. from machine import Pin #
2. import utime #
3. led = Pin(1, Pin.OUT) #
4. buzzer = Pin(20, Pin.OUT) #
5. led.value(0) #
6. buzzer.value(0) #
7. #
8. segments = (8, 9, 4, 3, 2, 7, 6, 5)
9. # Ortak katod a, b, c, d, e, f, g, . sırasıyla
10.
11. patterns = [
12.     (1, 1, 1, 1, 0, 1, 1, 0), # 9
13.     (1, 1, 1, 1, 1, 1, 1, 1), # 8
14.     (1, 1, 1, 0, 0, 0, 0, 0), # 7
15.     (1, 0, 1, 1, 1, 1, 1, 1), # 6
16.     (1, 0, 1, 1, 0, 1, 1, 0), # 5
17.     (0, 1, 1, 0, 0, 1, 1, 1), # 4
18.     (1, 1, 1, 1, 0, 0, 1, 0), # 3
19.     (1, 1, 0, 1, 1, 0, 1, 1), # 2
20.     (0, 1, 1, 0, 0, 0, 0, 0), # 1
21.     (1, 1, 1, 1, 1, 1, 0, 1), # 0

```



```
22. ]
23.
24. pins = [Pin(seg, Pin.OUT) for seg in segments]
25.
26. display = Pin(13, Pin.OUT, value=0)
27.
28. def display_digit(digit):
29.
30.     pattern = patterns[digit]
31.
32.     for i in range(8):
33.         pins[i].value(pattern[i])
34.         # Display açılıyor
35.         display.value(1)
36.         # Display kapanıyor
37.         display.value(0)
38.         # Display test edildi
39. while True:
40.
41.     for i in range(10):
42.         start = utime.ticks_ms()
```

```
43. while utime.ticks_diff(utime.ticks_ms(),  
44. start) < 1000 : # 1000 msn geçtiğinde  
45. display_digit(i) # Displaye i değerini  
# 9'dan 0'a kadar gönder.
```

Yukarıdaki program 7 segment ortak katodlu bir display sırasıyla a = GP8, b = GP9, c = GP4, d = GP3, e = GP2, f = GP7, g = GP6 ve dot = GP5 bağlandıktan sonra katot pini 270-330Ω 1W'lık bir dirençle GP13 display pinine bağlanınca digitte sırasıyla 9, 8., 7, 6., 5, 4., 3, 2., 1, 0. şeklinde 9 dan 0'a doğru sayar.

IRNEC-7segment.py

```
1. import time
2. from machine import Pin, freq
3. from ir_rx.print_error import print_error
4. from ir_rx.nec import NEC_8
5. # Raspberry pi pin'ının on board ledi GP25'e bağlıdır
6. led = machine.Pin(25, machine.Pin.OUT)
7. # 7 segmentin ilk önce a,b,c,d,e,f,g sırasıyla
8. led-a = Pin(8, Pin.OUT)
9. led-b = Pin(9, Pin.OUT)
10. led-c = Pin(4, Pin.OUT)
11. led-d = Pin(3, Pin.OUT)
12. led-e = Pin(2, Pin.OUT)
13. led-f = Pin(7, Pin.OUT)
14. led-g = Pin(6, Pin.OUT)
15. led-ole = Pin(5, Pin.OUT)
16. # buzzer ve ir alıcı ayarlandı.
17. buzzer = Pin(20, Pin.OUT)
18. pin_ir = Pin(10, Pin.IN)
19. # 7 segment ortak katot displayde 1 gösterir
20. led-a.value(1)
```

21. led-b.value (1)

22. led-c.value (1)

23. led-d.value (1)

24. led-e.value (1)

25. led-f.value (1)

26. led-g.value (0)

27. led-role.value (0)

28. led.value (0)

29.

30. def decodeKeyValue (data):

31. if data == 0x19: # 0 için değerler

32. led-a.value (1)

33. led-b.value (1)

34. led-c.value (1)

35. led-d.value (1)

36. led-e.value (1)

37. led-f.value (1)

38. led-g.value (0)

39. return "0"

40. if data == 0x45: # 1 için değerler

41. led-a.value (0)

```
42. led-b.value(1)
43. led-c.value(1)
44. led-d.value(0)
45. led-e.value(0)
46. led-f.value(0)
47. led-g.value(0)
48. return "1"
49. if data == 0x46: # 2 için değerler
50.     led-a.value(1)
51.     led-b.value(1)
52.     led-c.value(0)
53.     led-d.value(1)
54.     led-e.value(1)
55.     led-f.value(0)
56.     led-g.value(1)
57.     return "2"
58. if data == 0x47: # 3 için değerler
59.     led-a.value(1)
60.     led-b.value(1)
61.     led-c.value(1)
62.     led-d.value(1)
```

```
63. led-e.value(0)
64. led-f.value(0)
65. led-g.value(1)
66. return "3"
67. if data == 0x44: # 4 için değerler
68.     led-a.value(0)
69.     led-b.value(1)
70.     led-c.value(1)
71.     led-d.value(0)
72.     led-e.value(0)
73.     led-f.value(1)
74.     led-g.value(1)
75.     return "4"
76. if data == 0x40: # 5 için değerler
77.     led-a.value(1)
78.     led-b.value(0)
79.     led-c.value(1)
80.     led-d.value(1)
81.     led-e.value(0)
82.     led-f.value(1)
83.     led-g.value(1)
84.     return "5"
```

85. **if** data == 0x43: # 6 için değerler

86. led-a.value(1)

87. led-b.value(0)

88. led-c.value(1)

89. led-d.value(1)

90. led-e.value(1)

91. led-f.value(1)

92. led-g.value(1)

93. **return** "6"

94. **if** data == 0x7: # 7 için değerler

95. led-a.value(1)

96. led-b.value(1)

97. led-c.value(1)

98. led-d.value(0)

99. led-e.value(0)

100. led-f.value(0)

101. led-g.value(0)

102. **return** "7"

103. **if** data == 0x15: # 8 için değerler

104. led-a.value(1)

105. led-b.value(1)

```
106. led_c.value(1)
107. led_d.value(1)
108. led_e.value(1)
109. led_f.value(1)
110. led_g.value(1)
111. return "8"
112. if data == 0x09: # 9 için değerler.
113.     led_a.value(1)
114.     led_b.value(1)
115.     led_c.value(1)
116.     led_d.value(1)
117.     led_e.value(0)
118.     led_f.value(1)
119.     led_g.value(1)
120.     return "9"
121. if data == 0x16: # * için değerler
122.     led_role.value(1) # role seçer.
123.     led.value(1) # Yeşil on board led yanar
124.     return "*"
125. if data == 0x00: # # için değerler
126.     led_role.value(0) # role bırakır.
```

```

127. led.value(0) # Yeşil on board led söner.
128. return "#"
129.
130. def callback(data, addr, ctrl)
131.     if data < 0:
132.         pass
133.     else:
134.         print(decodeKeyValue(data))
135.
136. ir = NEC_8(pin=ir, callback)
137.
138. try:
139.     while True:
140.         pass
141. except KeyboardInterrupt:
142.     ir.close()
143.

```

Yukarıdaki program ir alıcı GP10'a ile 7 segment led-
 leri sırayla a=GP8, b=GP9, c=GP4, d=GP3, e=GP2,
 f=GP7, g=GP6 ve role=GP5'e bağlandığında 7 seg-
 ment göstergede uk vericiden gönderilen kodlara göre
 rakam gösterir. Aynı anda aynı rakam shell'de görülür. 86

```
1. from machine import Pin
2. import time
3.
4. led-ileri = Pin(2, Pin.OUT)
5. led-geri = Pin(19, Pin.OUT)
6. # MOTORA ve MOTOR B
7. In1 = Pin(3, Pin.OUT)
8. In2 = Pin(4, Pin.OUT)
9. led-ileri.low()
10. #MOTORA ve MOTOR B
11. In3 = Pin(6, Pin.OUT)
12. In4 = Pin(7, Pin.OUT)
13. led-geri.low()
14. # ILERI
15. def move_forward():
16.     In1.high()
17.     In2.low()
18.     In3.high()
19.     In4.low()
20.     led-ileri.high()
21. # GERI
22. def move_backward():
```

```
23. In1.low()
24. In2.high()
25. In3.low()
26. In4.high()
27. led-geri.high()

28. # SAGA
29. def turn-right():
30.     In1.low()
31.     In2.low()
32.     In3.high()
33.     In4.low()

34. # SOLA
35. def turn-left():
36.     In1.high()
37.     In2.low()
38.     In3.low()
39.     In4.low()

40. # DUR
41. def stop():
42.     In1.low()
43.     In2.low()
```

```
44. In3.low()
45. In4.low()
46. led-ileri.low()
47. led-geri.low()
48.
49. while True:
50.     move_forward()
51.     print("ILERI")
52.     time.sleep(3)
53.     stop()
54.     print("DUR")
55.     time.sleep(3)
56.     move_backward()
57.     print("GERI")
58.     time.sleep(3)
59.     stop()
60.     print("DUR")
61.     time.sleep(3)
62.     turn_right()
63.     print("SAGA")
64.     time.sleep(3)
65.     stop()
66.     print("DUR")
```

```
67. time.sleep(3)
68. turn-left()
69. print("SOLA")
70. time.sleep(3)
71. stop()
72. print("DUR")
73. time.sleep(3)
```

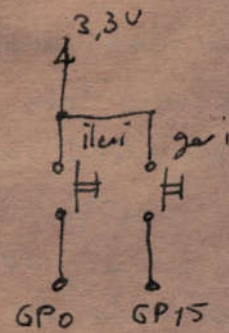
Yukarıdaki program led-ileri GP2'ye, led-geri GP19'a, L298 mini'nin Pinleri sırasıyla IN1 = GP3'e, IN2 = GP4'e, IN3 = GP6'ya ve IN4 = GP7'ye bağlandığı zaman 3'er saniye aralıklarla önce motorlar ileri döner, sonra durur, sonra motorlar geri döner ve durur. En sonunda sol motor (MotorB) ileri giderek sağa döner, durur sağ motor (MotorA) ileri giderek sola döner. Bu işlemler tekrar tekrar devam eder. Motorlar ileri giderken Yeşil led (led-ileri) yanar. Geri giderken ise kırmızı led (led-geri) yanar. Bu sırada shell kısmında aşağıdaki gibi durumlar yazdırılır.

```
GERİ
DUR
SAGA
DUR
SOLA
DUR
ILERİ
DUR
```

```

1. from machine import Pin
2. import time
3. buton_ileri = Pin(0, Pin.IN, Pin.PULL_DOWN)
4. buton_geri = Pin(15, Pin.IN, Pin.PULL_DOWN)
5. led_ileri = Pin(2, Pin.OUT)
6. led_geri = Pin(19, Pin.OUT)
7. # MOTORA (sağ motor)
8. In1 = Pin(3, Pin.OUT)
9. In2 = Pin(4, Pin.OUT)
10. # MOTORB (sol motor)
11. led_ileri.low()
12. In3 = Pin(6, Pin.OUT)
13. In4 = Pin(7, Pin.OUT)
14. led_geri.low()
15. # ILERİ
16. def move_forward():
17.     In1.high()
18.     In2.low()
19.     In3.high()
20.     In4.low()
21.     led_ileri.high()
22. # GERİ

```



23. **def** move_backward():

24. In1.low()

25. In2.high()

26. In3.low()

27. In4.high()

28. led-geri.high()

29. # SAGA

30. **def** turn_right():

31. In1.low()

32. In2.low()

33. In3.high()

34. In4.Low()

35. # SOLA

36. **def** turn_left():

37. In1.high()

38. In2.low()

39. In3.low()

40. In4.low()

41. # DuR

42. **def** stop():

43. In1.low()

```
44. In2.low()
45. In3.low()
46. In4.low()
47. led-ileri.low()
48. led-geri.low()
49.
50.
51. while True:
52.
53.     if buton-ileri.value() == 1:
54.         led-ileri.value(1)
55.         move-forward()
56.         print("MOTORLAR ILERI")
57.         time.sleep(0.3)
58.         led-ileri.value(0)
59.         stop()
60.
61.     if buton-geri.value() == 1:
62.         led-geri.value(1)
63.         move-backward()
64.         print("MOTORLAR GERI")
65.         time.sleep(0.3)
66.         led-geri.value(0)
```

67. stop()

68.

Yukarıdaki program daha önce L298-mini için bağlı olan In1 = GP3, In2 = GP4, In3 = GP6 ve In4 = GP7 pinlerine ek olarak buton-ileri GP0'a ve buton-geri GP15'e bağlandığında ve ileri butonuna basıldığında motor A ve motor B ileri döner. Shell kısmında DE MOTORLAR İLERİ yazar. Butondan elimizi çektiğimizde motorlar durur. Burada motorlar dönerken yeşil led (led-ileri) yanar. Geri butonuna basıldığında da motor A ve motor B geri döner. Shell kısmında ise MOTORLAR GERİ YAZAR. Motorlar geri dönerken kırmızı led (led-geri) yanar. Shell kısmı ise aşağıdaki gibi görülür.

MOTORLAR İLERİ
MOTORLAR İLERİ
MOTORLAR İLERİ
MOTORLAR GERİ
MOTORLAR GERİ

IR-L298mini.py

```
1. import time
2. from machine import Pin, freq
3. from ir_rx.print_error import print_error
4. from ir_rx.nec import NEC_8
5. # ILERİ ve GERİ ledleri tanımlıyoruz.
6. led-ileri = Pin(2, Pin.OUT)
7. led-geri = Pin(19, Pin.OUT)
8. # SAĞ ve SOL farları tanımlıyoruz
9. far-sol = Pin(1, Pin.OUT)
10. far-sag = Pin(13, Pin.OUT)
11. led = machine.Pin('LED', machine.Pin.OUT)
12. buzzer = Pin(20, Pin.OUT)
13. pin-ir = Pin(10, Pin.IN)
14. buzzer.value(0)
15. # MOTOR A girişleri
16. In1 = Pin(3, Pin.OUT)
17. In2 = Pin(4, Pin.OUT)
18. led-ileri.low()
19. # MOTOR B girişleri
20. In3 = Pin(6, Pin.OUT)
21. In4 = Pin(7, Pin.OUT)
22. led-geri.low()
23. # ILERİ
```

```
24. def move_forward():
25.     In1.high()
26.     In2.low()
27.     In3.high()
28.     In4.low()
29.     led-ileri.high()
30. #GERI
31. def move_backward():
32.     In1.low()
33.     In2.high()
34.     In3.low()
35.     In4.high()
36.     led-geri.high()
37. # SAGA
38. def turn_right():
39.     In1.low()
40.     In2.low()
41.     In3.high()
42.     In4.low()
43. # SOLA
44. def turn_left():
```

```
45. In1.high()
46. In2.low()
47. In3.low()
48. In4.low()
49. # dur
50. def stop():
51.     In1.low()
52.     In2.low()
53.     In3.low()
54.     In4.low()
55.     led-ileri.low()
56.     led-geri.low()
57.
58. def decodeKeyValue(data):
59.     if data == 0x19: # led'i kapat
60.         led.value(0)
61.         return "0"
62.     if data == 0x45: # led'i aç.
63.         led.value(1)
64.         return "1"
65.     if data == 0x46: # 2
66.         return "2"
```

```
67. if data == 0x47: #3
68.     return "3"
69. if data == 0x44: #4
70.     return "4"
71. if data == 0x40: #5
72.     return "5"
73. if data == 0x43: #6
74.     return "6"
75. if data == 0x7: #7
76.     return "7"
77. if data == 0x15: # Buzzeri öttür ve sustur.
78.     buzzer.value(1)
79.     time.sleep(0.5)
80.     buzzer.value(0)
81.     return "8"
82. if data == 0x09: #9
83.     return "9"
84. if data == 0x16: # Farları aç
85.     far_sol.value(1)
86.     far_sag.value(1)
87.     return "*"
98
```

```
88. if data == 0x0D: # Farları kapat
89.     far_sol.value(0)
90.     far_sag.value(0)
91.     return "#"
92. if data == 0x08: # Sola
93.     turn_left()
94.     return "LEFT"
95. if data == 0x1C: # Dur
96.     stop()
97.     return "OK"
98. if data == 0x5A: # Sağa
99.     turn_right()
100.    return "RIGHT"
101. if data == 0x52: # Geri
102.    move_backward()
103.    return "DOWN"
104. if data == 0x18: # İleri
105.    move_forward()
106.    return "UP"
107.
108.    return "ERROR"
109.
110. def callback(data, addr, ctrl):
```

```

111. if data < 0 :
112.     pass
113. else:
114.     print (decodeKeyValue(data))
115.
116. ir = NEC_8 (pin_ir, callback)
117. # ir.error_function (print_error)
118. # Yukarıdaki satır debug için hatayı gösterir.
119. try :
120.     while True:
121.         pass
122. except KeyboardInterrupt:
123.     ir.close()

```

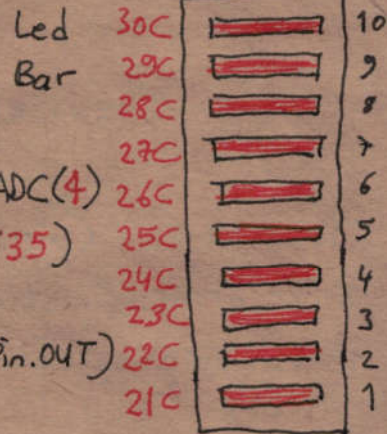
Yukarıdaki program ir_alıcı GP10'a bağlandığında L298 mini ise In1=GP3, In2=4, In3=6, In4=7'ye bağlıken sol far GP1'e, sağ far GP13'e bağlanarak; IR kumandadan ① tuşuna basılınca on board ledi yakar ② tuşuna basınca on board ledi söndürür. UP ▲ butonuna basılınca Motorlar ileri döner. DOWN ▼ butonuna basılınca motorlar geri döner. RIGHT ➡ tuşuna basılınca Motor B ileri döner. (sağa) LEFT ◀ tuşuna basılınca Motor A ileri döner (sola) ⑧ butonuna basılınca korna (buzzer) GP20 öter ve 1/2 sn sonra susar. * farları açar, # ise kapatır. 100

led_sicaklik.py

```

1. import machine
2. import utime
3.
4. sicaklik_sensory = machine.ADC(4)
5. sicaklik_cevirme = 3.3 / (65535)
6.
7. led = machine.Pin(25, machine.Pin.OUT)
8.
9. led21C = machine.Pin(2, machine.Pin.OUT)
10. led22C = machine.Pin(3, machine.Pin.OUT)
11. led23C = machine.Pin(4, machine.Pin.OUT)
12. led24C = machine.Pin(5, machine.Pin.OUT)
13. led25C = machine.Pin(6, machine.Pin.OUT)
14. led26C = machine.Pin(7, machine.Pin.OUT)
15. led27C = machine.Pin(8, machine.Pin.OUT)
16. led28C = machine.Pin(9, machine.Pin.OUT)
17. led29C = machine.Pin(10, machine.Pin.OUT)
18. led30C = machine.Pin(11, machine.Pin.OUT)
19. role = machine.Pin(25, machine.Pin.OUT)
20.
21. led.value(0)
22. role.value(0)

```



```
23,  
24. while True:  
25. okunan_sicaklik=sicaklik_sensoru.read_u16()*  
26. sicaklik_cevirme  
27. sicaklik = 27 - (okunan_sicaklik - 0.706) / 0.001721  
28. print(sicaklik)  
29. utime.sleep(1)  
30.  
31. if sicaklik > 21;  
32. led.off()  
33. led21C.on()  
34. led22C.off()  
35. led23C.off()  
36. led24C.off()  
37. led25C.off()  
38. led26C.off()  
39. led27C.off()  
40. led28C.off()  
41. led29C.off()  
42. led30C.off()  
43. role.off()  
44. if sicaklik > 22;  
45. led.off()  
46. led21C.on()
```

```
47. led22C.on()
48. led23C.off()
49. led24C.off()
50. led25C.off()
51. led26C.off()
52. led27C.off()
53. led28C.off()
54. led29C.off()
55. led30C.off()
56. role.off()
57. if sıcaklik > 23:
58.     led.off()
59.     led21C.on()
60.     led22C.on()
61.     led23C.on()
62.     led24C.off()
63.     led25C.off()
64.     led26C.off()
65.     led27C.off()
66.     led28C.off()
67.     led29C.off()
68.     led30C.off()
69.     role.off()
70. if sıcaklik > 24:
```

```
71. led.off()
72. led21C.on()
73. led22C.on()
74. led23C.on()
75. led24C.on()
76. led25C.off()
77. led26C.off()
78. led27C.off()
79. led28C.off()
80. led29C.off()
81. led30C.off()
82. role.off()
83. if sıcaklik > 25:
84.     led.off()
85.     led21C.on()
86.     led22C.on()
87.     led23C.on()
88.     led24C.on()
89.     led25C.on()
90.     led26C.off()
91.     led27C.off()
92.     led28C.off()
93.     led29C.off()
94.     led30C.off()
95.     role.off()
96. if sıcaklik > 26:
```

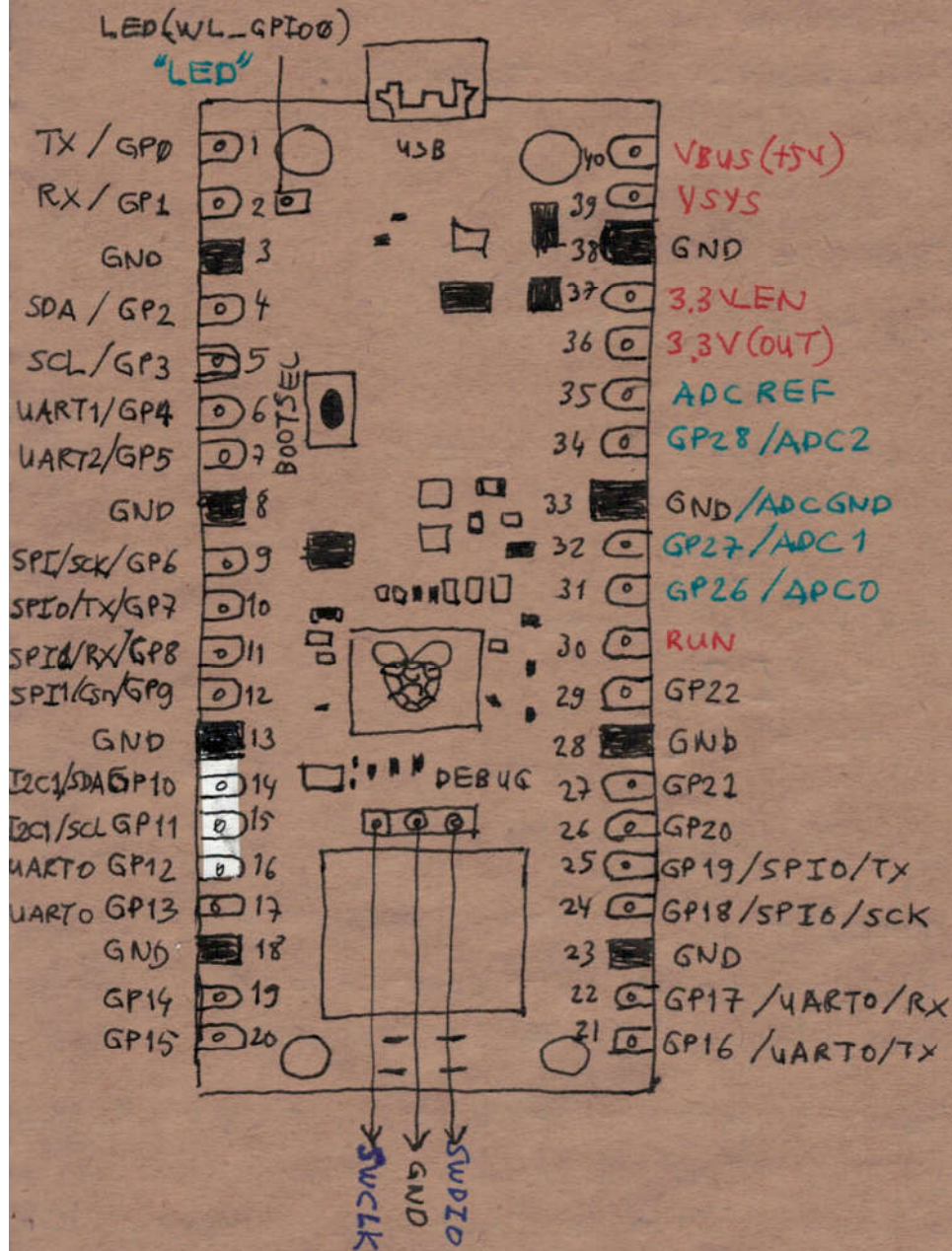
```
97.    led.off()
98.    led21C.on()
99.    led22C.on()
100.   led23C.on()
101.   led24C.on()
102.   led25C.on()
103.   led26C.on()
104.   led27C.off()
105.   led28C.off()
106.   led29C.off()
107.   led30C.off()
108.   role.off()
109.   if sıcaklik > 27:
110.       led.off()
111.       led21C.on()
112.       led22C.on()
113.       led23C.on()
114.       led24C.on()
115.       led25C.on()
116.       led26C.on()
117.       led27C.on()
118.       led28C.off()
119.       led29C.off()
120.       led30C.off()
121.       role.off()
122.   if sıcaklik > 28:
```

```
123. led.off()
124. led21C.on()
125. led22C.on()
126. led23C.on()
127. led24C.on()
128. led25C.on()
129. led26C.on()
130. led27C.on()
131. led28C.on()
132. led29C.off()
133. led30C.off()
134. role.off()
135. if sıcaklik > 29:
136.     led.off()
137.     led21C.on()
138.     led22C.on()
139.     led23C.on()
140.     led24C.on()
141.     led25C.on()
142.     led26C.on()
143.     led27C.on()
144.     led28C.on()
145.     led29C.on()
146.     led30C.off()
147.     role.off()
148. if sıcaklik > 30:
```

```
149. led.on()
150. led21C.on()
151. led22C.on()
152. led23C.on()
153. led24C.on()
154. led25C.on()
155. led26C.on()
156. led27C.on()
157. led28C.on()
158. led29C.on()
159. led30C.on()
160. role.on()
```

Yukarıdaki program Led Baragraf (10 led'li) üzerinde sıcaklık değeri gösterir. 1. bar 21°C , 2. bar 22°C , 3. bar 23°C 10 bar ise 30°C 'yi göstermek için kullanılmıştır. Raspberry pi pico içindeki led sıcaklığın 30°C ye ulaştığında çalışır. Ayrıca GPIO12 pinine bağlı olan bir role ile sıcaklık değerine göre herhangi bir ısıtıcı veya fan çalıştırılabilir. Örneğin sıcaklık 30°C 'nin üzerine çıkınca role çalışarak fanı çalıştırabilir. Böylece sıcaklık kontrol edilebilir. (oda sıcaklığı, kuluçka vb) Burada sıcaklık sensörü olarak ADC4 olarak tanımlanan ve 12 bit çözünürlük ile çalışan onboard ısı sensörü kullanılmıştır.

Raspberry Pi Pico W Pinout



tm1637-saat.py

```
1. from machine import Pin
2. from tm1637 import TM1637
3. import time
4.
5. display = TM1637 (clk=Pin(4), dio=Pin(5))
6. display = brightness(4)
7.
8. colon_state = True # : gösterildi
9.
10. while True:
11.     t = time.localtime()
12.     hour = t[3]
13.     minute = t[4]
14.
15.     display.numbers(hour, minute, colon=colon_
16.     state) # : işaretini tersine çevir
17.     colon_state = not colon_state
18.     time.sleep(0.5) # 500 ms bekle
```

Yukarıdaki program TM1637'nin clk pinini Rasp-
berry Pi'nin 4. pinine, dio pinini 5.
pinine bağlandığında local time bilgisini gösterir. 109

```

1. import network
2. import time
3. import ntptime
4. from machine import Pin
5. from tm1637 import TM1637
6.
7. # wifi Bilgileri
8. SSID = "FiberHGW-ZT39P-2.4GHz"
9. PASSWORD = "7zet9Nt1ck"
10.
11. # Donanım tanımları
12. tm = TM1637(cik = Pin(4), dio = Pin(5))
13. buzzer = Pin( , Pin.OUT) # Aktif buzzer
14.
15. # Wi-Fi bağlantısı
16. def connect_wifi():
17.     wlan = network.WLAN(network.STA_IF)
18.     wlan.activate(True)
19.     wlan.connect(SSID, PASSWORD)
20.
21.     timeout = 15
22.     for i in range(timeout):
23.         status = wlan.status()
24.         print(f"📶 Bağlantı denemesi {i+1} / 15 - Durum: {status}")

```

```
25. if wlan.isConnected()
26.     print("✓ Wi-Fi bağlı:", wlan.ifconfig())
27.     return True
28.     time.sleep(1)
29.
30.     print("X Wifi bağlantısı başarısız.")
31.     return False
32.
33. # Saat senkronizasyonu
34. def sync-time():
35.     try:
36.         ntp.time.settime()
37.         print("Saat senkronize edildi.")
38.         return True
39.     except Exception as e:
40.         print("Saat alınamadı:", e)
41.         return False
42. # Türkiye saati hesaplama
43. def get-local-time():
44.     t = time.localtime()
45.     hour = (t[3] + 3) % 24 # utc + 3
46.     minute = t[4]
47.     return hour, minute
48.
```

```

49. # Alarm tetikleme
50. def check_alarm(hour, minute):
51.     return hour == 16 and minute == 0
52. if connect_wifi():
53.     if sync_time():
54.         while True:
55.             hour, minute = get_local_time()
56.             print(f"Saat: {hour:02}:{minute:02}")
57.             tm.numbers(hour, minute)
58.
59.             if check_alarm(hour, minute):
60.                 print("Alarm! 16:00")
61.                 for _ in range(5): # 5 saniye bip sesi
62.                     buzzer.on()
63.                     time.sleep(0.5)
64.                     buzzer.off()
65.                     time.sleep(0.5)
66.                 time.sleep(10)
67.             else:
68.                 tm.show("FAIL")
69.         else:
70.             tm.show("NOWF")

```