

T.C.
KARABÜK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ



AKILLI ORTAM DENETLEYİCİ RGB LED
TERMİNAL

BİTİRME TEZİ

Hazırlayan

MERT ÇOLAK 2015110226006

Tez Danışmanı

Yrd.Doç.Dr. İbrahim ÇAYIROĞLU

KARABÜK-2019

İÇİNDEKİLER

	<u>Sayfa</u>
İÇİNDEKİLER.....	1
ÖNSÖZ.....	3
ÖZET	4
TABLolar VE ŞEKİLLER.....	5
KISALTMALAR	6
BÖLÜM 1	7
GİRİŞ.....	7
1.1. GİRİŞ.....	7
1.2. AMAÇ	7
1.3. KULLANIM ALANI	7
BÖLÜM 2	8
ELEKTRONİK KART TASARIMI, ÜRETİLMESİ VE PROGRAMLANMASI	8
2.1. KULLANILAN ENTEGRELER VE KOMPONENTLER	8
2.1.1. ATMEGA328P-AU MİKRODENETLEYİCİ	8
2.1.2. CH340G (USB_TTL)	9
2.2. KART ÜRETİMİ.....	9
2.2.1. ŞEMA ÇİZİMİ	10
2.2.2. BASKI ÇİZİMİ	11
2.2.3. BASKI DEVRE YAPMA	11
2.3. ELEKTRONİK KARTIN PROGRAMLANMASI.....	102
2.3.1. BOOTLOADER YÜKLENMESİ	13
2.3.2. PROGRAMIN YÜKLENMESİ	13
2.3.3. ELEKTRONİK KARTIN İLETİŞİM PROTOKOLÜ	14
2.3.4. ELEKTRONİK KARTIN PROGRAMI.....	15

BÖLÜM 3	17
3.1. LCD ÇERÇEVE YAPIMI.....	17
3.1.1. LCD ÇERÇEVESİ MONTAJI.....	17
3.1.2. LEDLER İÇİN JAK VE KABLO BAĞLANTISI	17
BÖLÜM 4	18
4.1. BİLGİSAYAR PROGRAMI.....	18
4.1.1. PROGRAMIN KURULMASI	18
4.1.2. PROGRAMIN TANITILMASI	19
4.1.3. PROGRAM KODLARI	22
KAYNAKÇA	39
ÖZGEÇMİŞ	40

ÖNSÖZ

Öncelikle, projeme görüşleriyle yön veren ve desteğini hiç esirgemeyen, Yrd. Doç.Dr. İbrahim ÇAYIROĞLU hocama ve lisans eğitimim boyunca bizden desteklerini esirgemeyen bölüm hocalarıma teşekkür ederim. Tüm eğitim öğretim hayatım boyunca benden maddi ve manevi desteklerini eksik etmeyen aileme sonsuz teşekkürlerimi sunarım.

MERT ÇOLAK

ÖZET

Yaptığım bu projede akıllı ortam denetleyici rgb led terminal yapacağım. Genel amaç oyun ve film sektöründeki görselliği arttırmada kullanılan ambiyans aydınlatmalarla ilişkili. Proje de tasarlanan kart ve yazılım sayesinde bilgisayardaki görüntüyü analiz ederek monitörün arkasını aydınlatır. Bu bir monitörde olabilir, bir sinema sahnesi de. Böylelikle ekrandaki oyunu, filmi ledlerle ekrandan, dışarı taşıyan kendinizi daha etkin bir şekilde oyunun, filmin içindeymiş gibi hissetmenizi sağlayan bir aydınlatma uygulamasıdır. Sadece akıllı ambiyans yaratmayacak. Farklı modlarda eklenmiştir. Örnek vermek gerekirse yazılımsal ses analizi. Yazılımsal ses analizi sayesinde bilgisayara ek bir donanım takılmadan ses yazılımsal olarak analiz edilecek, kullanıcı müziği kulaklıkla dahi dinlese analiz edilerek müziğin şiddetine göre tepki verir. Diğer modlardan bahsedecek olursam flash, polis, ambulans, itfaiye, sabit renk gibi efektler eklenmiştir. uyku modu gibi değişik modlarda eklenmiştir. Mesela 120 dakikaya programı kurduğunuzda istediğinize göre 120 dakika sonunda tamamen kapanabilir veya 120 dakika dolmaya başladıkça yavaş yavaş söner ve kapanır.

TABLolar VE ŐEKİLLER

Sayfa

<i>Őekil 1: Akıllı ortam denetleyici rgb led terminal ile aydınlatılmış bir sahne.....</i>	<i>8</i>
<i>Őekil 2: Atmega328p-au pin yapısı.....</i>	<i>9</i>
<i>Őekil 3: Őema Tasarımı.....</i>	<i>10</i>
<i>Link 1: Elektronik Kart Őeması Ve Baskı Devresi.....</i>	<i>10</i>
<i>Tablo 1: Amper Deęerine Gre Devre Yolları Kalınlıęı Őeması.....</i>	<i>11</i>
<i>Őekil 4: Baskı Devre Yapma.....</i>	<i>12</i>
<i>Tablo 2: Arduino Uno ile bootloader yklenirken baęlanması gereken pin baęlantıları...</i>	<i>13</i>
<i>Őekil 5: Elektronik Kartın Program Kodları.....</i>	<i>15</i>
<i>Őekil 6: LED erevenin yapımı.....</i>	<i>17</i>
<i>Tablo 3: Programın Ana Blmleri.....</i>	<i>19</i>
<i>Őekil 7: Programdan bir grnt.....</i>	<i>20</i>
<i>Tablo 4: Program ierisinde yer alan Modların aıklaması.....</i>	<i>21</i>
<i>Őekil 8: Program Kodları.....</i>	<i>22</i>
<i>Link 2: Projenin alıřma videosu.....</i>	<i>38</i>

KISALTMALAR

Volt	V
Amper	A
Red Green Blue	RGB
Universal Serial Bus	USB
Light Emitting Diode	LED
Transistor To Transistor Logic	TTL
Basic Input/Output System	BIOS
Alf Ve Vegard RISC	AVR
Pulse Width Modulation	PWM
Bit Per Second	BPS
Frame Per Second	FPS

BÖLÜM 1

1.1. Giriş

Akıllı ambiyans aydınlatma, bulunduğu çevreyi veya ortamı o anki ses, müzik veya görüntü analizi yaparak ortamı aydınlatır. Ortama göre aydınlatma yapılarak ışığın tonu, rengi ve aydınlatma miktarı ayarlanabilir. Loş bir ortam veya aydınlık bir ortam seçilebilir. Böylelikle ortam ambiyansını hissetmeyi güçlendirir. Bir su sahnesinde veya bir alev sahnesinde ortam sahneye göre aydınlandığında sanki filmin içindeymişiz gibi hissetmemizi sağlar.

1.2. Amaç

Bir akıllı ortam denetleyici rgb led terminal tasarlamak ve yazılımıyla birlikte tüm bilgisayarla uyumlu çalışan bir yazılım yazarak akıllı ortam denetleyici projesi oluşturmak. Böylelikle her bilgisayara kolayca takılabilen çevresel bir donanım aksesuarı oluşturmaktır. Proje tasarlanırken seri üretim düşünülerek elektronik kart tasarlanmış ve yazılımın uyumluluğu düşünülmüştür. Buna göre her bilgisayarda bulunan haberleşme için USB kullanılmış. Yazılım ise C# ta yazılarak, çeşitli performans geliştirmeleri yapılarak çekirdek için özel olarak performans seçenekleri eklenmiş ve kurulabilir şekilde derlenmiştir. Böylece her kullanıcı bilgisayarı için istediği performansı seçerek rahatça kullanabilmektedir.

1.3. Kullanım Alanı

Sinema, konferans, gösterimler, oyuncu etkinlikleri gibi görüntü odaklı veya müzik odaklı, konser, parti gibi aydınlatma sağlanan her yerde kullanılabilir. Proje ambiyans yaratmak istenen her yer de kullanılabilir. Ve farklı modlar eklendiği için mesela uyku modu, karanlıktan korkan çocuklar için bile kullanılabilir. Belirli bir dakika sonunda aydınlatma kesilir veya istenildiği takdir de seçeneklerden yavaş yavaş karartma seçilebilir. Onun dışında acil bir durum için polis, itfaiye, ambulans gibi çeşitli modlar kullanılabilir. Kısacası bir çevresel donanım olarak bilgisayara takılabileceği gibi bir gösteride akıllı ortam aydınlatması için de kullanılabilir.



Şekil 1: Akıllı ortam denetleyici rgb LED terminal ile aydınlatılmış bir sahne

BÖLÜM 2

2.1. Kullanılan Entegreler Ve Komponentler

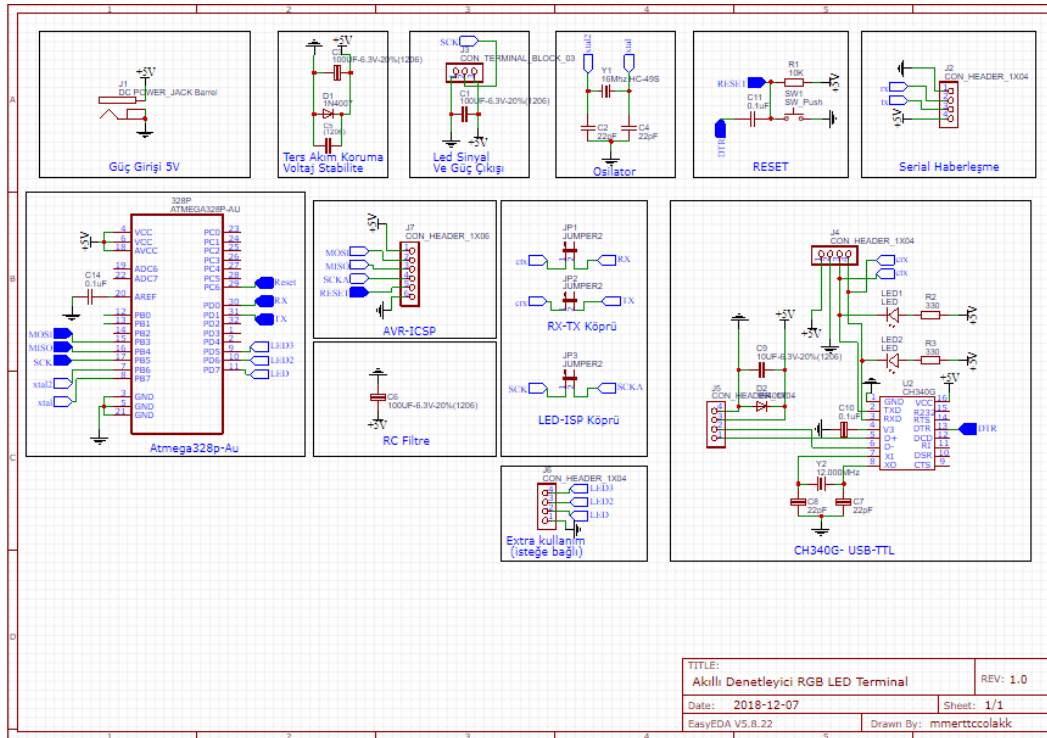
Kart tasarımında iki ana entegre bulunmaktadır. Bunlardan biri “atmega328p-au” mikro denetleyicisi. Diğeri ise “CH340G” entegresidir. Diğerkomponentler ise güç girişı için adaptör jak’ı, voltaj regülesi için regülatör, kısa devre ve güç dalgamanlasını önlemek için diyot ve kondansatörler bulunmaktadır.

2.1.1. Atmega328p-Au Mikro Denetleyici

Projenin kart üzerinde çalışan sistemin ana bölümüdür. “CH340G” gelen seri bilgileri işler ve programlanabilir olan “WS2812B” ledlerin konumunu belirler ve ona göre parlaklık, renk gibi verileri gönderir. Bu mikro denetleyici ürünü TQFP-32 kılıfta olup SMD yapısına sahip olduğu için dip kılıflara göre az yer kaplar. 20 MHz frekansı ile çalışan 8 bit

2.2.1. Şema Çizimi

Şema çizimini yaparken mikro denetleyicinin ve kullanacağım seri port entegresinin datasheet'ini kullanacağım. Daha önceden kullandığım kart tasarımlarımdan da örnek olarak şemayı parça parça modül olarak tasarlayacağım. Böylelikle ilk devremizi test ederken tek bir kart tasarlamak yerine, iki ayrı modülle test edebileceğiz. Örnek olarak 2 kart tasarlanacak ve bir sorun olursa tüm entegreler tüm çizimler yeniden çizilmeyecektir. En son devreler sorunsuz çalışır ise tek bir kart tasarlanarak tekrar basılacaktır. Buradaki amaç aslında versiyon geliştirmek içinde kullanılabilir. USB-TTL modülü sabitse diğer versiyonda mikro denetleyici değişirse prototipini test etmek hem daha ekonomik hem de daha hızlı olacaktır. Buradaki tasarladığım devrede bir sorun çıkmadı. Usb-ttl ve mikro denetleyici olarak tasarladığım iki modülde düzgün çalıştı. Böylelikle tek bir kart tasarımı yaparak projeyi bitirdim.



Şekil 3: Şema Tasarımı

2.2.2 Baskı Çizimi

Şemasını çizdiğimiz elektronik kartın, baskı devresi için devre yollarını çizdim bu bölümde. Burada dikkat edilmesi gereken nokta dip ve smd elemanlarını yerleştirirken yüzlere dikkat edilmesidir. Dip komponentler olduğu zaman kartın diğer yüzünde olduğu için entegreler simetri edilerek karta çizilir. Bir diğer noktada elemanların güç analizini yaparak mesela devremizde olan güç çekecek olan “şerit programlanabilir ws218b ledleri” gibi elemanların devre yollarını kalın tasarlamaktır.

Link 1: Elektronik Kart Şeması Ve Baskı Devresi

[“https://easyeda.com/mmerttccolakk/Ambilight-Project”](https://easyeda.com/mmerttccolakk/Ambilight-Project)

Bağlantı Yolu Genişliği	Taşınabilecek Maksimum Akım Değeri
0,2 mm	100 mA
0,5 mm	300 mA
1,0 mm	2,5 A
2,0 mm	5 A
3,0 mm	6 A
4,0 mm	7 A
5,0 mm	9 A

Tablo 1: Baskı devre yolları tasarlanırken standart bir bakır plaka için komponentlerin ve elemanların çekebileceği amper göz önüne alınarak devre yolunun genişliğine dikkat edilir.

2.2.2 Baskı Devre Yapma

Baskı devre yapmak için birçok yöntem vardır. Seri üretim için cnc veya lazerli bakır yol çiziciler vb. Ben baskı devreyi yaparken ütü baskı yöntemini kullandım. Baskı devreyi

izdiđim baskıyı aynalayarak pdf'i lazerli yazıcıyla kuşe kâğıdına ıktı aldım. Aynalamamın sebebi kâğıttan bakır yzeğe getiđinde ıktının simetrisi olmaktadır. Burada yazıcının tonerli olması nemli nk toner sıcaklıkla transfer olabilen bir madde. Ve biz bu maddeyi tlerek baskı devreye geiriyoruz. Bakır levhayı temizledikten sonra tnn en sıcak ayarında ıktı aldıđımız kâğıdı 10 dakika tlyoruz. Baskı devre, bakır plakaya ıktıktan sonra. Bakır levha zerindeki kâğıdı su ile yavaşa temizliyoruz, sadece toner kalacak şekilde. Daha sonra %20 Pehidrol, %80 Tuz Ruhu karışımına atarak bakır levhanın tonerli olmayan kısımlarının erimesini bekliyoruz. Yaklaşık 5 dakika sryor. Bylelikle devre yollarımız ıkmış oldu. Ardından pin deliklerini delerek, komponentleri yerleřtirip lehimliyoruz.



řekil 4: Baskı Devre Yapma

2.3. ELEKTRONİK KARTIN PROGRAMLANMASI

ncelikle kullandıđımız mikro denetleyici (atmega328p-au) ATMEL altyapısında olduđu iin ilk alıřtırma iin bootloader gereklidir. İlk alıřtırma iin bu zorunlu bir iřlemdir. Bootloader iřlemi tıpkı bilgisayarların BIOS'u gibi dřnlebilir. Mikro denetleyicinin temel fonksiyonlarını yerine getirmesi iin gereklidir. Bootloader iřlemi ISP protokol ile bařka bir ATMEL mikro denetleyici veya bir AVR programlayıcısıyla gerekleřtirilebilir.

Ben bir arduino kullanarak bu işlemi gerçekleştirdim. Bootloader işlemini gerçekleştirdikten sonra artık ttl seviye ile devremizde bulunan “CH340G” entegremiz sayesinde bir daha programlayıcı gerekmeden devre kendisini USB protokolünü kullanarak programlayabilecektir.

2.3.1. BOOTLOADER YÜKLENMESİ

Öncelikle arduino veya avr bağlantımızı şekil 5 te verilen şekilde ISP pin ve güç bağlantılarını yaptıktan sonra Arduino Programını IDE sini açıyoruz.

- Dosya > Uygulamalar > ArduinoISP seçip.
- Araçlar > Board > Arduino Uno
- Araçlar > Serial Port (Buradan hangi Com portundaysanız onu seçin)
- Araçlar > Programlayıcı > AVRISP mkII seçiniz.
- Araçlar > Önyükleyiciyi yazdır.

Böylelikle Bootloader işlemini tamamlamış olduk.

ARDUİNO UNO	MİKRODENETLEYİCİ
5V	VCC
GND	GND
10	RESET
11	MOSI
12	MISO
13	SCK

Tablo 2: Arduino Uno ile bootloader yüklenirken bağlanması gereken pin bağlantıları

2.3.2. PROGRAMIN YÜKLENMESİ

Kartımıza daha önceden yerleştirdiğimiz dahili olan “CH340G” entegresi olduğundan dolayı kartımıza Arduino IDE’yi açarak kolayca arduino uno programlıyormuş gibi programlayabiliriz. Kartımızı USB ile bilgisayarımıza taktığımızda “CH340G” entegresi RX-TX bağlantıları ile atmega328p-au ile veri transferi yaparken DTR-RESET bağlantısı ile de otomatik yazılım yükleme moduna almaktadır. Yazdığımız kodu kartımıza atmak

iin programımızı atıktan sonra, Aralar > Kart menüsünden Arduino UNO seeneğini tıklıyoruz. Daha sonra, yine Aralar menüsünden Port alt menüsü altında kartımızın baėlı olduėu portu seiyoruz. Bu port numarası, her bilgisayarda farklı olabilmektedir. Programımızı yazdıktan sonra kartımıza yüklemek istediėimizde, öncelikle “Kontrol Et” seeneğine tıklıyoruz. Program, yazdığımız kodu öncelikle bilgisayarımızda bir klasöre kaydetmemizi istiyor, daha sonra da yazdığımız kodu derleyerek herhangi bir hata varsa bu hatayı bize bildiriyor. Eėer yazdığımız kodda bir hata yoksa ve kartımız bilgisayarımıza USB ile baėlıysa, “Yükle” seeneğine tıklayarak kodumuzu kartımıza yükleyebiliyoruz.

2.3.3. ELEKTRONİK KARTIN İLETİŞİM PROTOKOLÜ

Elektronik kartın iletişimi “CH340G” entegresi ile bilgisayardan usb baėlantısıyla seri port aıp atmega328p-au baėlanmasıyla gerekleşir. Elektronik kartın programı yazılırken veri iletişim performansını ve yazılımın stabilitesini arttırmak amacıyla deėişkenler minumum seviyede kullanılmış. Ve üst deėer limitleri fazla olan int long vs deėişkenleri mümkün olduėu yerlerde byte ve sabit deėer yapısıyla tasarlanmıştır. Int ile 0-255 saydırmakla byte yapısında 0-255 saydırmak arasında performans aısından büyük bir fark bulunmaktadır. 1 Byte = 8 Bit deėerindedir. Kısaca programın iletişim protokolünden bahsedecek olursam. 60 Led’imiz varsa her biri için 3 byte gereklidir. Her byte led’in bir rengi için bir PWM deėerini temsil etmektedir. Bir led’te tüm renkleri oluştıurabilmek adına 3 ana renk bulunur. Bunlar kırmızı, yeşil ve mavidir. Bunlara ayrı, ayrı vereceėimiz pwm sinyali sayesinde bir led’te $0-255^3$ ten 16581375 farklı renk elde edilebilmektedir. Ledlerin konumu için ise farklı bir protokol geliştirerek konum bilgisi göndermedim. Eėer konum bilgisini de göndersemdim 3 byte yerine her seferinde 4 byte gönderecektim ve buda %25 performans kaybı demektir. Bilgisayar programında kaç adet led’imiz olduėunu kurulum sırasında bir kez söylüyoruz. Ve daha sonra her iki program her 3 byte da bir diėer led için bilgileri almaya başlıyor. Bu döngü her led için tekrarlandıktan sonra program başa dönüyor. Böylece 3 byte ile hem konum hem de renk bilgisi saėlanabiliyor. Programımız 115200 BPS de iletişim kuruyor. Saniyede 115200 bit transfer ediliyor.

Bir saniyede tüm led’lerin güncellenme sayısı:

60 led’imiz olduėuna göre $60 \cdot 3$ RGB için 180 byte

60*3*8 byte >> bit çevrimi 1440 bit

İletişim protokolümüz bant genişliği 155200 bps

Saniye tüm ledlerin güncellenme sayısı $155200/1440=80$ rps

Kısaca iletişim protokolümüz 60 tane rgb led'i konumuna ve rengine göre saniyede 80 defa güncelleyebiliyor.

2.3.4. ELEKTRONİK KARTIN PROGRAMI

```
//kütüphaneler
#include "FastLED.h"

//led tanımlamaları
#define DATA_PIN 13

//led pwm degiskenleri
byte red=0; byte green=0; byte blue=0;

//led aydınlık tanımlama eeprom üzerinde olacak bunlar
#define BRIGHTNESS 100
#define NUM_LEDS 64
#define NUM_BYTES (NUM_LEDS*3) // rgb 3 renk olduğu için led*3 byte oluşturuyoruz

//led sayi degisken ve byte sayi degisken
uint8_t led_say = 0;
uint8_t byte_say = 0;

//kaç adet led var tanımlıyoruz ve buffer tutuyoruz
CRGB leds[NUM_LEDS];
byte buffer[NUM_BYTES];

void setup() {
  //seri bağlantı kuruluyor
  Serial.begin(115200);
  Serial.setTimeout(500);
  //resetlenme ve ön bekleme ile güç kontrolü için
  delay(500);
}
```



```

//led katagorisi tanımlanıyor
FastLED.clear(true);
FastLED.addLeds<WS2812B, DATA_PIN, GRB>(leds, NUM_LEDS);
FastLED.setBrightness(BRIGHTNESS);

//başlangıç çalıştırma için rgb test yapılıyor
for( byte a = 0; a < 4; a++) {
  red=0; green=0; blue=0;
  if(a==0){red = 255;}else if(a==1){ green=255; }else if(a==2){blue=255;}else{red = 255;green=255;blue=255;}
  for( int i = 0; i < NUM_LEDS; i++) {
    leds[i]= CRGB(red,green,blue);
    FastLED.show();
    delay(5);
  }
  for( int i = NUM_LEDS; i > 0; i--) {
    leds[i] = CRGB::Black;
    FastLED.show();
    delay(3);
  }
}
FastLED.clear(true);
red=0; green=0; blue=0;
}

void loop() {
  if (Serial.available() >0){Serial.readBytes(buffer, NUM_BYTES);

  while (byte_say < NUM_BYTES){

    byte red   = buffer[byte_say++];
    byte green = buffer[byte_say++];
    byte blue  = buffer[byte_say++];
    leds[led_say++] = CRGB(red, green, blue);
  }

  FastLED.show();

  byte_say = 0;
  led_say = 0;
}
}

```

Şekil 5: Elektronik Kartın Program Kodları

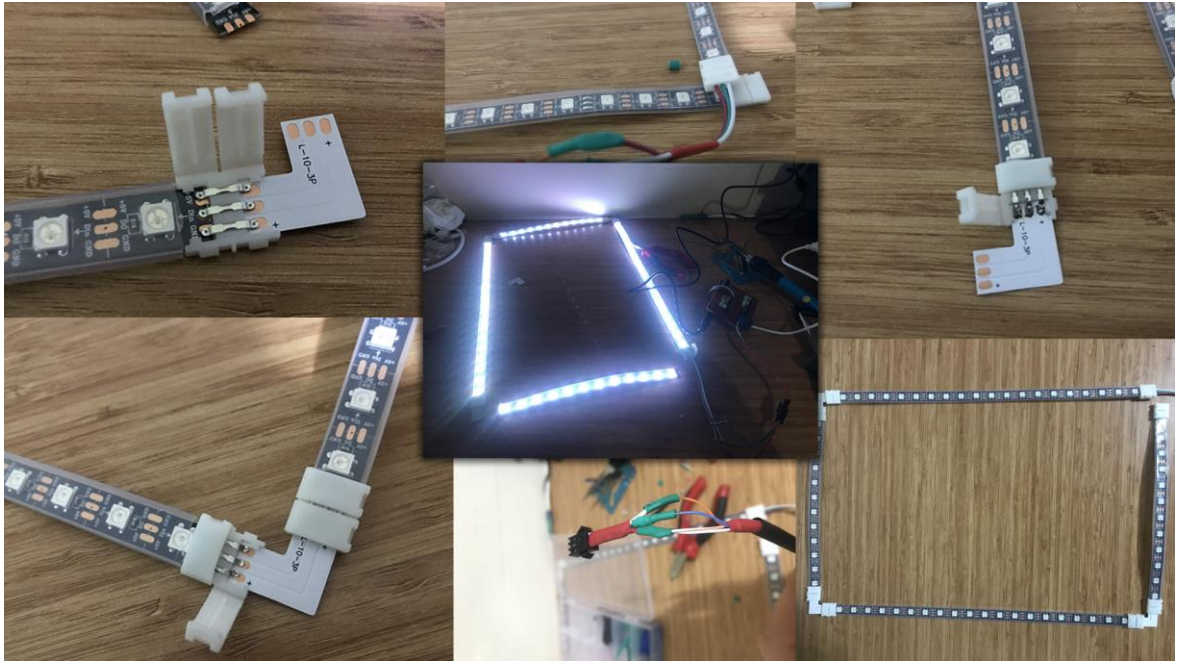
BÖLÜM 3

3.1. LCD ÇERÇEVE YAPIMI

LCD çerçeve yaparken, tasarlayacağımız çerçeve programlanabilir olan WS218B şerit ledlerden yapılmıştır. Kolaylıkla çerçevenin her bilgisayarın monitörüne, televizyona, sinema sahnesi vs uyumlu olması için modüler yapılmıştır. Ekleme aparatlarıyla sadece ledleri keserek araya 90 derece konektörler atılır.

3.1.1. LCD ÇERÇEVESİ MONTAJI

Ledler, ekranın ölçüsü alınarak yatayda ve dikeyde uygun boyda kesilir. Ardından 90 derecelik konektörler takılarak dikdörtgen biçimine getirilir. Burada dikkat edilmesi gereken nokta ledlerin akış yönüdür. Ledler üzerindeki oklara göre birbirine bağlanmalıdır. Çerçeve dizildikten sonra sıcak silikon ile bilgisayarın çerçevesine montajı yapılmıştır.



Şekil 6: LED çerçevenin yapımı

3.1.2. LED İÇİN JAK VE KABLO BAĞLANTISI

LCD çerçevesi tamamlandıktan sonra sıra LCD çerçevesinin elektronik karta bağlanmasında, bilgisayara sürekli olarak takılmayacağını düşünürsek veya taşınabilir bir bilgisayara takılabileceğini düşünürsek bu çerçeveye sabit bir kablo takmak doğru olmaz. Konektör tarzında kablo ayıraçları ile kablolar takılmıştır. Bir konektör LCD çerçeveye bağlanırken Bir çerçevede elektronik karta bağlanarak istenildiğinde bilgisayardan kart sökülüp başka bir bilgisayara takılabilir. Ve taşıma aşamasında kolaylıkta sağlar. Kablo konektörleri bağlanırken lehim kullanılarak zamanla oksitlenmesi engellendi daha sonra kablolar makaronla sarılıp ısıtıldı ve iyice sağlamlaştırıldı.

BÖLÜM 4

4.1. BİLGİSAYAR PROGRAMI

Bilgisayar programı Visual Studio programında C# programlama dili kullanılarak yazılmıştır. Kullanıcıların kolaylıkla kullanabilmesi için arayüz sadeleştirilmiş, ayarlar ve hesaplamalar otomatikleştirilmiştir. Program ilk açıldığında led sayılarını bir kez girip kaydet'e basmanız durumunda bir daha tekrar aynı ayarı yapmanıza gerek kalmamaktadır. Program içerisinde özel ayarları tutan bir veri tabanı eklenmiştir. Ayrıca her bilgisayarda kolaylıkla çalışabilmesi için RAM ihtiyacını düşük tutan (using, thread, private value gibi) fonksiyonlarla programlanmıştır. Programı ilk tasarladığımda saniyede iki kere görüntü analizi yaparken fonksiyon kullanımlarını ve çeşitli performans ayarlarını öğrendiğimde bu sayı saniyede dört yüz 'ün üzerine çıktı. RAM ise 1500MB tan 30MB düştü. Burada belirtmek istediğim konu iki kodda aynı işi yaparken aradaki kod farkının ne derece önemli olduğudur.

4.1.1. PROGRAMIN KURULMASI

Programın kurulması ve ayarlarının yapılması için ilk önce program açılır ve LCD çerçevesinin tek taraflı boyuna ve yatay ledleri sayılarak programın sol üstteki kısmına girilir ve "Ayarları Kaydet" butonuna tıklanır ve program yeniden başlatılır. Daha sonra bağlantı ayarları için elektronik kart bilgisayara takılır ve "Port" kısmından COM portu seçilerek

bağlana tıklanır. Eğer bilgisayarda başka bir cihaz bağlı değilse zaten tek bir port gözükecektir. Ama birden fazla cihaz takılıysa yanlış portu seçebilirsiniz. Eğer yanlış portu seçmiş iseniz LCD çerçeve bir tepki vermeyecektir. “Bağlantıyı Kes” deyip bir diğer portu seçin. Eğer doğru portu seçtiyseniz çerçeve kırmızı, yeşil, mavi ve beyaz olmak üzerine hızlıca yanıp söner. Bunun bir diğer nedeni de rgb ledlerin sağlamlık testini yapmaktır. Böylece kurulumda tamamlanmış olur. Bağlantı portunu değiştirmedığınız sürece port numarası sabit kalacaktır.

4.1.2. PROGRAMIN TANITILMASI

Program 4 ana bölümden oluşmaktadır. Bunlar aşağıdaki “Tablo 3”te gösterilmiştir.

Bölüm	Açıklama
İstatistik-Durum	FPS değeri vb. değerlerin istatistikleri
Led Ve Ekran Ayarları	LED sayısının girildiği ayarlar
Bağlantı Ayarları	Bağlantının sağlandığı bölüm
Modlar	Modların seçildiği ve ayarlandığı bölüm

Tablo 3: Programın Ana Bölümleri

- **İstatistik-Durum**

Bu bölümde programın LED’leri kaç kere güncellediği FPS değerinden görülebilir. Programın içerisine bir saat eklenmiştir ve bu saat son bir saniyede LED’lerin kaç kere güncellendiğini tutarak FPS değerini ölçmüş olur. Bir diğer değişkende kaç dakika kaldığıdır.

- **LED Ve Ekran Ayarları**

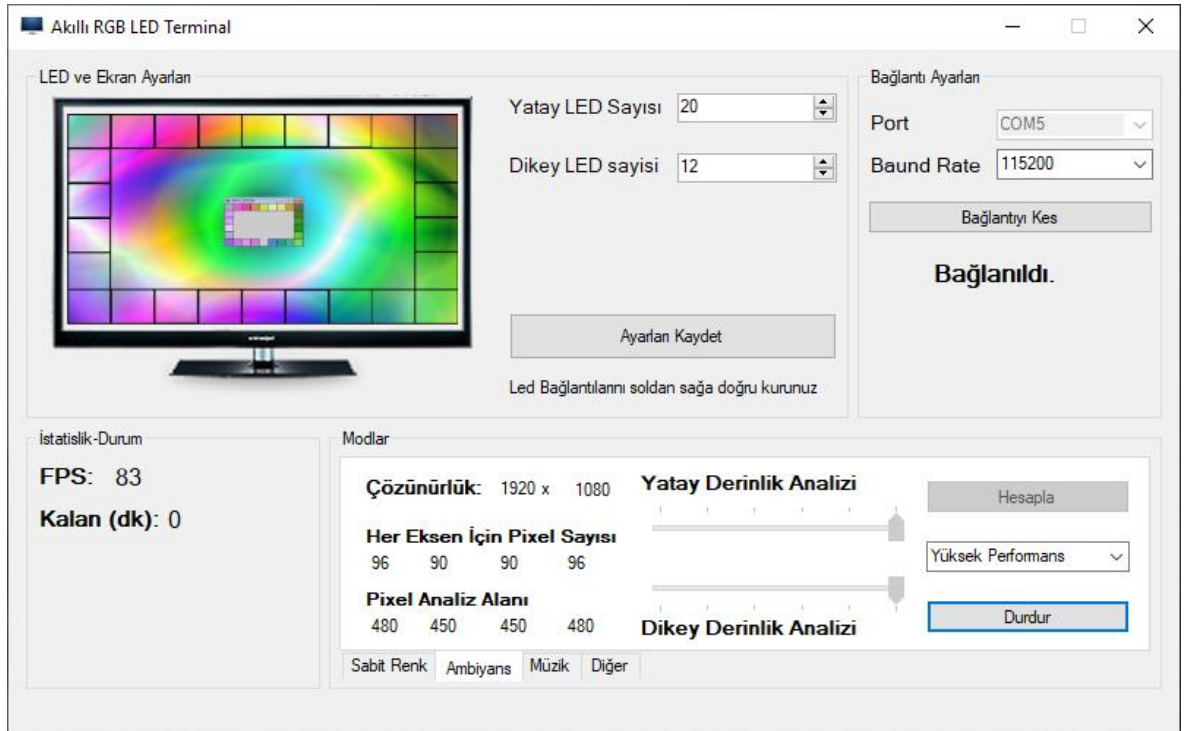
Bu bölümde programa LCD çerçevdeki yatay ve dikey LED sayıları girilerek kaydedilir.

- **Bağlantı Ayarları**

Bu bölümde seri port bağlantısı sağlanır. Aynı zamanda ilerde kart güncellenirse diye programa bant genişlik ayarı da konulmuştur. Buradaki bağlantı kontrolü aynı zamanda try/catch yapısı kullanılarak tasarlanmıştır. Ve bir timer ile sürekli bağlantı kontrol edilir. Port birden kapanır veya hata verirse programda herhangi bir donma, takılma olmaz. Program otomatik portu kapatır ve başlangıçtaki port seçim ekranına otomatik döner.

- **Modlar**

Modlar kısmında tab sekmesi kullanılarak bir grup içerisine birçok mod eklenmiştir. Bunlar aşağıdaki “Tablo 4” te gösterilmiştir.



Şekil 7: Programdan bir görüntü

Sabit Renk Modu

Bu mod 'ta bir renk paletinden bir renk seçilerek Tüm LED'lerin rengi değiştirilir.

Ambiyans MODU

Bu mod 'ta yatay ve dikey analiz katsayısı istenilirse arttırılır. Ve “Hesapla” butonuna basarak gerekli hesapların yapılması sağlanır. Ardından istenilirse Performans ayarı değiştirilir. Buradaki performans ayarı, çalışan fonksiyona çekirdek önceliğinin bir parametresidir. Yüksek Performans, Normal, Düşük Performans gibi parametreleri vardır. Artık “Çalıştır” butonuna tıklayarak ambiyans modunu aktif edebiliriz. Bu modu aktif ettiğimizde ekrandaki görüntüyü analiz edip ona göre dış çerçeveyi aydınlatacaktır.

Müzik Modu

Bu mod 'ta programda öncelikle müzik modunda hangi çıkış veya giriş cihazının referans alınacağı aygıt listesinden seçilir. Bu bir mikrofon veya hoparlörde olabilir. Daha sonra istenirse trackbar dan ne kadar yükseltileceği seçilir. Ardından performans ayarları istenirse seçilir. Ve en son müzik modunun hangi ritimde yansıtılacağı seçilir. Burada farklı efektler vardır. Mesela müziğin ritmine göre şerit ledlerin şiddeti sıralı olarak kayar veya tamamen müziğe göre renk şiddeti oluşturur.

Diğer Modlar

Bu bölümde bir çok efekt vardır. Mesela uyku modu programı 120 dakikaya kurup 120 dakika sonunda tamamen kapanmasını sağla biliriz. Veya 120 dakika boyunca yavaşça sönmesini de seçebiliriz. Onun dışında acil durumlar ve ortam

aydınlatması için polis, itfaiye, ambulans, flash, sakinleştirici gibi çeşitli efektlerde eklenmiştir.

Tablo 4: Program içerisinde yer alan Modların açıklaması

4.1.3. PROGRAM KODLARI

```
1 using AkilliAmbiyansRGBLED.Properties;
2 using System;
3 using System.Collections.Generic;
4 using System.ComponentModel;
5 using System.Data;
6 using System.Drawing;
7 using System.Linq;
8 using System.Text;
9 using System.Threading.Tasks;
10 using System.Windows.Forms;
11 using System.Threading;
12 using NAudio.CoreAudioApi;
13 using System.Diagnostics; //fps hesaplama
14
15 namespace AkilliAmbiyansRGBLED
16 {
17     4 basvuru
18     public partial class Form1 : Form
19     {
20         int genislik, yukseklik, x_bolunme, y_bolunme;
21         bool ilk_calistirma = false; bool suren_islem = false; bool ilk_calistirma_music = false; bool music_suren_islem = false; bool efekt_suren_islem = false;
22         bool ilk_calistirma_efekt = false; bool dinlendirme_mod = false;
23         byte yatay_derinlik_deger, dikey_derinlik_deger; byte yükseltme_degeri = 0; byte music_modu_degiskeni = 255; byte efekt_modu_degiskeni = 0;
24         byte saydir = 0; //sıralı music led için byte dizisini dizmek için kullanılıyor
25         Stopwatch zaman_say = new Stopwatch(); //uyku modu için global zaman sayma
26         //diziler byte tipinde oluşturuldu
27         Byte[] sayilar = new Byte[((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3];
28
29     1 basvuru
30     public Form1()
31     {
32         InitializeComponent();
33         Control.CheckForIllegalCrossThreadCalls = false;
34     }
35     //genel olarak işçiler tanımlanıyor
36     Thread fps_cekirdek;
37     Thread music_cekirdek;
38     Thread efekt_cekirdek;
39
40     1 basvuru
41     public Form1()
42     {
43         InitializeComponent();
44         Control.CheckForIllegalCrossThreadCalls = false;
45     }
46     //genel olarak işçiler tanımlanıyor
47     Thread fps_cekirdek;
48     Thread music_cekirdek;
49     Thread efekt_cekirdek;
50
51     1 basvuru
52     public void baglanti_ac()
53     {
54         //comboBox üzerinden port seçildi butonlar açılıp timer ve comboboxlar kapatılacak
55         baglan_b.Enabled = true;
56         portlar.Enabled = false;
57     }
58
59     0 basvuru
60     public void verigonder(Byte deger, Byte uzunluk)
61     {
62         if (serialPort1.IsOpen == true)
63         {
64             //serialPort1.Write(deger, 0, uzunluk);
65         }
66     }
67
68     1 basvuru
69     private void baglan_b_Click(object sender, EventArgs e)
70     {
71         baglan();
72     }
73
74     1 basvuru
75     private void Form1_Load(object sender, EventArgs e)
76     {
77         //Ayarlar yükleniyor
78         yatay_led.Value = Settings.Default.yatay;
79         dikey_led.Value = Settings.Default.dikey;
80     }
81 }
```

```

61         dikey_led.Value = Settings.Default.dikey;
62         //çözünürlük yükleniyor
63         genislik = Screen.PrimaryScreen.Bounds.Width;
64         yukseklik = Screen.PrimaryScreen.Bounds.Height;
65         //music ayarları
66
67         MMDeviceEnumerator enumerator = new MMDeviceEnumerator();
68         var devices = enumerator.EnumerateAudioEndPoints(DataFlow.All, DeviceState.Active);
69         ortam_muzik.Items.AddRange(devices.ToArray());
70     }
71
72     1 başvuru
73     private void portlar_SelectedIndexChanged(object sender, EventArgs e)
74     {
75         baglanti_ac();
76     }
77
78     1 başvuru
79     private void portlar_Click(object sender, EventArgs e)
80     {
81         if (portlar.Text == "Port Seçin")
82         {
83             portlar.Items.Clear();
84             for (int i = 0; i < System.IO.Ports.SerialPort.GetPortNames().Length; i++)
85             {
86                 portlar.Items.Add(System.IO.Ports.SerialPort.GetPortNames()[i]);
87             }
88         }
89
90     2 başvuru
91     public void baglan()
92     {
93         //port açılıp bağlanılıyor
94         if (serialPort1.IsOpen == false)
95         {
96             serialPort1.PortName = portlar.Text;
97             serialPort1.BaudRate = Convert.ToInt32(haberlesme_hiz.Text);
98             serialPort1.ReadTimeout = 500;
99             serialPort1.WriteTimeout = 500;
100         }

```



```

96
97         baglanti_bildirim.Text = "Bağlanılıyor...";
98     try
99     {
100         serialPort1.Open();
101         baglan_b.Text = "Bağlantıyı Kes";
102         periyodik_kontrol.Enabled = true;
103         baglanti_bildirim.Text = "Bağlandı.";
104
105     }
106     catch (Exception hata)
107     {
108         periyodik_kontrol.Enabled = false;
109         baglanti_bildirim.Text = "Hata:" + hata.Message;
110         serialPort1.Close();
111         baglan_b.Text = "Bağlan";
112         baglanti_bildirim.Text = "Bağlantı Yok";
113         baglan_b.Enabled = false;
114         portlar.Enabled = true;
115         haberlesme_hiz.Enabled = true;
116         portlar.Text = "Port Seçin";
117         portlar.Items.Clear();
118     }
119 }
120 else
121 {
122     serialPort1.Close();
123     baglan_b.Text = "Bağlan";
124     baglanti_bildirim.Text = "Bağlantı Yok";
125     periyodik_kontrol.Enabled = false;
126     baglan_b.Enabled = false;
127     portlar.Enabled = true;
128     haberlesme_hiz.Enabled = true;
129     portlar.Text = "Port Seçin";
130     portlar.Items.Clear();
131 }
132 }
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168

```

```

1 bagvuru
private void button8_Click(object sender, EventArgs e)
{
    if (colorDialog1.ShowDialog() == DialogResult.OK)
    {
        //renk seçildi ise renk kodu byte olarak rgb olarak aktarılır pwm
        button8.BackColor = colorDialog1.Color;
        byte red = colorDialog1.Color.R;
        byte green = colorDialog1.Color.G;
        byte blue = colorDialog1.Color.B;

        //progressbar rgb pwm değerleri yansıtılıyor
        red_prog.Value = Convert.ToInt32(red);
        green_prog.Value = Convert.ToInt32(green);
        blue_prog.Value = Convert.ToInt32(blue);

        //diziler byte tipinde oluşturuldu
        Byte[] sayilar = new Byte[((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3];

        //diziye rgb pwm değerleri byte cinsinden aktarılıyor
        for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
        {
            sayilar[i] = red;
            sayilar[i + 1] = green;
            sayilar[i + 2] = blue;
        }

        if (serialPort1.IsOpen == true)
        {
            serialPort1.Write(sayilar, 0, sayilar.Length);
            serialPort1.Write(sayilar, 0, sayilar.Length);
        }
    }
}

```

```

168
169 1 başvuru
170 private void Button1_Click(object sender, EventArgs e)
171 {
172     Settings.Default.dikey = Convert.ToInt32(dikey_led.Value);
173     Settings.Default.yatay = Convert.ToInt32(yatay_led.Value);
174     Settings.Default.Save();
175 }
176
177 1 başvuru
178 private void hesaplama_buton_Click(object sender, EventArgs e)
179 {
180     //cozunurluk
181     genislik_text.Text = genislik.ToString();
182     yukseklik_text.Text = yukseklik.ToString();
183     //bolunme pixel hesap
184     sol_bol.Text = (yukseklik / Convert.ToInt32(dikey_led.Value)).ToString();
185     ust_bol.Text = (genislik / Convert.ToInt32(yatay_led.Value)).ToString();
186     alt_bol.Text = (genislik / Convert.ToInt32(yatay_led.Value)).ToString();
187     sag_bol.Text = (yukseklik / Convert.ToInt32(dikey_led.Value)).ToString();
188     //bolunme deęişkenleri atanıyor
189     y_bolunme = Convert.ToInt32(sol_bol.Text);
190     x_bolunme = Convert.ToInt32(ust_bol.Text);
191     //her analiz için bir analiz alanı
192     sol_alan.Text = (yatay_derinlik.Value * Convert.ToInt32(sol_bol.Text)).ToString();
193     sag_alan.Text = (yatay_derinlik.Value * Convert.ToInt32(sag_bol.Text)).ToString();
194     alt_alan.Text = (dikey_derinlik.Value * Convert.ToInt32(alt_bol.Text)).ToString();
195     ust_alan.Text = (dikey_derinlik.Value * Convert.ToInt32(ust_bol.Text)).ToString();
196     //derinlik analizi için deęer byte deęerine aktarılıyor (daha düşük yer kaplasın diye)
197     dikey_derinlik_deger = Convert.ToByte(dikey_derinlik.Value);
198     yatay_derinlik_deger = Convert.ToByte(yatay_derinlik.Value);
199     //çalıştır butonu aktif ediliyor hesaplamalar yapıldı
200     calistir_buton.Enabled = true;
201 }
202
203 1 başvuru
204 private void ortam_muzik_SelectedIndexChanged(object sender, EventArgs e)
205 {
206     var device = (MMDDevice)ortam_muzik.SelectedItem;
207     int booster = (int)(Math.Round(device.AudioMeterInformation.MasterPeakValue * yukseltme.Value));
208     if (booster > 100) { booster = 100; }
209 }
210
211 1 başvuru
212 private void muzik_calistir_Click(object sender, EventArgs e)
213 {
214     if (ortam_muzik.Text == "Lütfen bir ses aygıtı seçin.")
215     {
216         MessageBox.Show("Lütfen bir ses aygıtı seçin.");
217     }
218     else
219     {
220         //ilk çalıştırma
221         if (!ilk_calistirma_music)
222         {
223             //iscilerin is parçacıkları tanıtıldı
224             music_cekirdek = new Thread(music_cekirdek_islemi);
225             //debug için işçi isimleri
226             music_cekirdek.Name = "music_iscisi";
227             //yukseltme.Value thread için deęişkene atanıyor
228             yukseltme_degeri = Convert.ToByte(yukseltme.Value);
229             //music modu belirleniyor
230             switch (music_modu.Text)
231             {
232                 case "Sabit Renk (Kırmızı)":
233                     music_modu_degiskeni = 0;
234                     break;
235                 case "Sabit Renk (Mavi)":
236                     music_modu_degiskeni = 1;
237                     break;
238                 case "Sabit Renk (Yeşil)":
239                     music_modu_degiskeni = 2;
240                     break;

```

```

230     {
231         case "Sabit Renk (Kırmızı)":
232             music_modu_degiskeni = 0;
233             break;
234         case "Sabit Renk (Mavi)":
235             music_modu_degiskeni = 1;
236             break;
237         case "Sabit Renk (Yeşil)":
238             music_modu_degiskeni = 2;
239             break;
240         case "Sabit Renk (Beyaz)":
241             music_modu_degiskeni = 3;
242             break;
243         case "Sabit Renk (Rastgele)":
244             music_modu_degiskeni = 4;
245             break;
246         case "Sırala (Kırmızı)":
247             music_modu_degiskeni = 5;
248             break;
249         case "Sırala (Yeşil)":
250             music_modu_degiskeni = 6;
251             break;
252         case "Sırala (Mavi)":
253             music_modu_degiskeni = 7;
254             break;
255         case "Sırala (Beyaz)":
256             music_modu_degiskeni = 8;
257             break;
258         case "Sırala (Rastgele)":
259             music_modu_degiskeni = 9;
260             break;
261         case "Sırala (Dağıt Şerit)":
262             music_modu_degiskeni = 10;
263             break;
264     }
265     //music çekirdek öncelikleri tanımlanıyor
266     switch (isci_music_oncealik.Text)
267     {
268         case "Düşük":
269             music_cekirdek.Priority = ThreadPriority.Lowest;
270     }
271     switch (isci_music_oncealik.Text)
272     {
273         case "Düşük":
274             music_cekirdek.Priority = ThreadPriority.Lowest;
275             break;
276         case "Ekonomi":
277             music_cekirdek.Priority = ThreadPriority.BelowNormal;
278             break;
279         case "Normal":
280             music_cekirdek.Priority = ThreadPriority.Normal;
281             break;
282         case "Performans":
283             music_cekirdek.Priority = ThreadPriority.AboveNormal;
284             break;
285         case "Yüksek Performans":
286             music_cekirdek.Priority = ThreadPriority.Highest;
287             break;
288     }
289     //thread içindeki sürdürülen işlem
290     music_suren_islem = true;
291     //işçiler çalıştırılsın
292     music_cekirdek.Start();
293     //butonlar
294     isci_music_oncealik.Enabled = false;
295     music_modu.Enabled = false;
296     ortam_muzik.Enabled = false;
297     yukseltme.Enabled = false;
298     muzik_calistir.Text = "Durdur";
299     //başlat durdur için bool
300     ilk_calistirma_music = true;
301     //muzik değişkenleri oluşturuluyor
302     var device = (MMDevice)ortam_muzik.SelectedItem;
303     int booster = (int)(Math.Round(device.AudioMeterInformation.MasterPeakValue * yukseltme.Value));
304     if (booster > 100) { booster = 100; }

```

```

305     else
306     {
307
308         //yasıyor ise durdurulacak
309         if (music_cekirdek != null && music_cekirdek.IsAlive) { music_cekirdek.Suspend(); }
310         //thread içindeki sürdürülen işlem
311         music_suren_islem = false;
312         //buton ayarları
313         isci_music_oncecik.Enabled = true;
314         music_modu.Enabled = true;
315         ortam_muzik.Enabled = true;
316         yukseleme.Enabled = true;
317         muzik_calistir.Text = "Çalıştır";
318         //başlat durdur için bool
319         ilk_calistirma_music = false;
320     }
321 }
322
323
324 1 başvuru
325 public void calistir_buton_Click(object sender, EventArgs e)
326 {
327     //ilk çalıştırma
328     if (!ilk_calistirma)
329     {
330         //iscilerin is parçacıkları tanıtıldı
331         fps_cekirdek = new Thread(fps_cekirdek_islemi);
332         //debug için işçi isimleri
333         fps_cekirdek.Name = "fps_iscisi";
334         //fps çekirdek öncelikleri tanımlanıyor
335         switch (fps_iscisi.Text)
336         {
337             case "Düşük":
338                 fps_cekirdek.Priority = ThreadPriority.Lowest;
339                 break;
340             case "Ekonomi":
341                 fps_cekirdek.Priority = ThreadPriority.BelowNormal;
342                 break;
343             case "Normal":
344                 fps_cekirdek.Priority = ThreadPriority.Normal;
345                 break;
346             case "Normal":
347                 fps_cekirdek.Priority = ThreadPriority.Normal;
348                 break;
349             case "Performans":
350                 fps_cekirdek.Priority = ThreadPriority.AboveNormal;
351                 break;
352             case "Yüksek Performans":
353                 fps_cekirdek.Priority = ThreadPriority.Highest;
354                 break;
355         }
356         //thread içindeki sürdürülen işlem
357         suren_islem = true;
358         //işçiler çalıştırılısın
359         fps_cekirdek.Start();
360         //butonlar
361         hesaplama_buton.Enabled = false;
362         yatay_derinlik.Enabled = false;
363         dikey_derinlik.Enabled = false;
364         calistir_buton.Text = "Durdur";
365         //başlat durdur için bool
366         ilk_calistirma = true;
367     }
368     else
369     {
370
371         //yasıyor ise durdurulacak
372         if (fps_cekirdek != null && fps_cekirdek.IsAlive) { fps_cekirdek.Suspend(); }
373         //thread içindeki sürdürülen işlem
374         suren_islem = false;
375         //buton ayarları
376         hesaplama_buton.Enabled = true;
377         yatay_derinlik.Enabled = true;
378         dikey_derinlik.Enabled = true;
379         ilk_calistirma = false;
380         calistir_buton.Text = "Çalıştır";
381         MMDeviceEnumerator enumerator = new MMDeviceEnumerator();
382
383         var devices = enumerator.EnumerateAudioEndpoints(DataFlow.All, DeviceState.Active);

```

```

377     MMDDeviceEnumerator enumerator = new MMDDeviceEnumerator();
378
379     var devices = enumerator.EnumerateAudioEndpoints(DataFlow.All, DeviceState.Active);
380     ortam_muzik.Items.Clear();
381     ortam_muzik.Items.AddRange(devices.ToArray());
382     ortam_muzik.Text = "Lütfen bir ses aygıtı seçin.";
383 }
384
385
386 1 başvuru
387 private void haberlesme_hiz_SelectedIndexChanged(object sender, EventArgs e)
388 {
389     haberlesme_hiz.Enabled = false;
390 }
391
392 1 başvuru
393 private void efekti_baslat_Click(object sender, EventArgs e)
394 {
395     if (efekt_modlari.Text == "Lütfen bir efekt seçin.")
396     {
397         MessageBox.Show("Lütfen bir efekt seçin.");
398     }
399     else
400     {
401         //ilk çalıştırma
402         if (!ilk_calistirma_efekt)
403         {
404             //iscilerin is parçacıkları tanıtıldı
405             efekt_cekirdek = new Thread(efekt_cekirdek_islemi);
406             //debug için işçi isimleri
407             efekt_cekirdek.Name = "efekt_iscisi";
408             //efekt modu belirleniyor
409             switch (efekt_modlari.Text){
410                 case "Polis":
411                     efekt_modu_degiskeni = 0;
412                     break;
413                 case "İtfaye":
414                     efekt_modu_degiskeni = 1;
415                     break;
416                 case "Ambulans":

```

```

411 case "İtfaye":
412     efekt_modu_degiskeni = 1;
413     break;
414 case "Ambulans":
415     efekt_modu_degiskeni = 2;
416     break;
417 case "Sakinleştirici":
418     efekt_modu_degiskeni = 3;
419     music_modu_degiskeni = 255;
420     dinlendirme_mod = false;
421     break;
422 case "Uyku":
423     efekt_modu_degiskeni = 4;
424     zaman_say.Stop();//zaman başlatılıyor
425     zaman_say.Start();//zaman başlatılıyor
426     music_modu_degiskeni = 255;
427     break;
428 case "Flash":
429     efekt_modu_degiskeni = 5;
430     break;
431 }
432 //fps çekirdek öncelikleri tanımlanıyor
433 switch (efekt_önceligi.Text)
434 {
435     case "Düşük":
436         efekt_çekirdek.Priority = ThreadPriority.Lowest;
437         break;
438     case "Ekonomi":
439         efekt_çekirdek.Priority = ThreadPriority.BelowNormal;
440         break;
441     case "Normal":
442         efekt_çekirdek.Priority = ThreadPriority.Normal;
443         break;
444     case "Performans":
445         efekt_çekirdek.Priority = ThreadPriority.AboveNormal;
446         break;
447     case "Yüksek Performans":
448         efekt_çekirdek.Priority = ThreadPriority.Highest;
449         break;
450 }

```

```

447         case Yüksek Performans :
448             efekt_cekirdek.Priority = ThreadPriority.Highest;
449             break;
450         }
451         //thread içindeki sürdürülen işlem
452         efekt_suren_islem = true;
453         //işçiler çalıştırılsın
454         efekt_cekirdek.Start();
455         //butonlar
456         efekti_baslat.Text = "Durdur";
457         //başlat durdur için bool
458         ilk_calistirma_efekt = true;
459     }
460     else
461     {
462         //yasıyor ise durdurulacak
463         if (efekt_cekirdek != null && efekt_cekirdek.IsAlive) { efekt_cekirdek.Suspend(); }
464         //thread içindeki sürdürülen işlem
465         efekt_suren_islem = false;
466         //buton ayarları
467         ilk_calistirma_efekt = false;
468         efekti_baslat.Text = "Çalıştır";
469     }
470 }
471 }
472
473 1 başvuru
474 private void efekt_modlari_SelectedIndexChanged(object sender, EventArgs e)
475 {
476     //uyku modu seçildiğinden uyku modunun seçenekleri açılacak
477     //seçilmediğinde ise boşuna seçenekler açık olmayacak
478     if (efekt_modlari.Text == "Uyku") {
479         uyku_suresi.Enabled = true;
480         uyku_sure_azalt.Enabled = true;
481         pwm_uyku.Enabled = true;
482     }else{
483         uyku_suresi.Enabled = false;
484         uyku_sure_azalt.Enabled = false;
485         pwm_uyku.Enabled = false;
486     }
487 }

```

```

482         uyku_suresi.Enabled = false;
483         uyku_sure_azalt.Enabled = false;
484         pwm_uyku.Enabled = false;
485     }
486 }
487
488 1 başvuru
489 private void periyodik_kontrol_Tick(object sender, EventArgs e)
490 {
491     if (serialPort1.IsOpen == false) { baglan(); }
492 }
493
494 1 başvuru
495 private void fps_cekirdek_islemi()
496 {
497     //suren islem durdurma baslatma için bool
498     while (suren_islem)
499     {
500         //fps hesaplama
501         Stopwatch watch = new Stopwatch();
502         watch.Start();
503         //ekran kopyalanıyor
504         Bitmap Screenshot = new Bitmap(genislik, yukseklik);
505         using (Graphics GFX = Graphics.FromImage(Screenshot))
506         {
507             GFX.CopyFromScreen(0, 0, 0, 0, Screenshot.Size, CopyPixelOperation.SourceCopy);
508         }
509
510         //diziler byte tipinde oluşturuldu
511         Byte[] gonderilen = new Byte[(((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3)];
512
513         //veri dizimi için sayaç ayarlanıyor
514         int sayac_anlik = 0;
515
516         //fonk degiskenleri tanımlanıyor
517         int alan = 0;
518         int red = 0;
519         int green = 0;
520         int blue = 0;
521
522         //pixel analizi yapılacak resim using kullanılarak seçilip ram tasarrufu yapılıyor
523         using (Bitmap myBitmap = new Bitmap(Screenshot))
524         {
525             //dikey soldan sağa pixel alanları gönderiliyor
526             for (int i = 0; i <= (genislik - x_bolunme); i += x_bolunme)
527             {
528                 //ilk degiskenler sıfırlanıyor ve dikey soldan sağa pixel alanları gönderiliyor
529                 alan = ((i + x_bolunme) - i) * (dikey_derinlik_deger);
530                 red = 0;
531                 green = 0;
532                 blue = 0;
533
534                 for (int c = 1; c <= dikey_derinlik_deger; c++)
535                 {
536                     for (int b = i; b <= (i + x_bolunme - 1); b++)
537                     {
538                         Color pixelColor = myBitmap.GetPixel(b, c);
539                         red += pixelColor.R;
540                         green += pixelColor.G;
541                         blue += pixelColor.B;
542                     }
543                 }
544
545                 red /= alan;
546                 green /= alan;
547                 blue /= alan;
548
549                 gonderilen[sayac_anlik] = Convert.ToByte(red);
550                 sayac_anlik++;
551                 gonderilen[sayac_anlik] = Convert.ToByte(green);
552                 sayac_anlik++;
553                 gonderilen[sayac_anlik] = Convert.ToByte(blue);
554                 sayac_anlik++;
555             }
556             //dikey aşağıya pixseller taranıyor
557             for (int i = 0; i <= (yukseklik - y_bolunme); i += y_bolunme)
558             {
559                 //ilk degiskenler sıfırlanıyor ve dikey aşağıya pixseller taranıyor
560                 alan = ((genislik) - (genislik - yatay_derinlik_deger)) * ((i + y_bolunme) - i);

```



```

556 {
557
558     //ilk degiskenler sifirlanıyor ve dikey aşağıya pixseller taranıyor
559     alan = ((genislik) - (genislik - yatay_derinlik_deger)) * ((i + y_bolunme) - i);
560     red = 0;
561     green = 0;
562     blue = 0;
563
564     for (int c = i; c <= (i + y_bolunme - 1); c++)
565     {
566         for (int b = (genislik - yatay_derinlik_deger); b <= (genislik - 1); b++)
567         {
568             Color pixelColor = myBitmap.GetPixel(b, c);
569             red += pixelColor.R;
570             green += pixelColor.G;
571             blue += pixelColor.B;
572         }
573     }
574     red /= alan;
575     green /= alan;
576     blue /= alan;
577
578     gonderilen[sayac_anlik] = Convert.ToByte(red);
579     sayac_anlik++;
580     gonderilen[sayac_anlik] = Convert.ToByte(green);
581     sayac_anlik++;
582     gonderilen[sayac_anlik] = Convert.ToByte(blue);
583     sayac_anlik++;
584 }
585
586 //sağ alttan sola doğru yatay ilerliyor aşağıdaki pixseller taranıyor
587 for (int i = 1920; i >= 5; i -= x_bolunme)
588 {
589     //ilk degiskenler sifirlanıyor sağ alttan sola doğru yatay ilerliyor aşağıdaki pixseller taranıyor
590     alan = ((i) - (i - x_bolunme)) * ((yukseklık) - (yukseklık - dikey_derinlik_deger));
591     red = 0;
592     green = 0;
593     blue = 0;
594
595     for (int c = (yukseklık - dikey_derinlik_deger); c <= (yukseklık - 1); c++)
596     {
597         for (int b = (i - x_bolunme); b <= (i - 1); b++)
598         {
599             Color pixelColor = myBitmap.GetPixel(b, c);
600             red += pixelColor.R;
601             green += pixelColor.G;
602             blue += pixelColor.B;
603         }
604     }
605     red /= alan;
606     green /= alan;
607     blue /= alan;
608
609     gonderilen[sayac_anlik] = Convert.ToByte(red);
610     sayac_anlik++;
611     gonderilen[sayac_anlik] = Convert.ToByte(green);
612     sayac_anlik++;
613     gonderilen[sayac_anlik] = Convert.ToByte(blue);
614     sayac_anlik++;
615 }
616
617 //dikey yukarı pixseller taranıyor
618 for (int i = 1080; i >= 5; i -= y_bolunme)
619 {
620     //ilk degiskenler sifirlanıyor ve dikey yukarı pixseller taranıyor
621     alan = ((yatay_derinlik_deger + 1)) * ((i) - (i - y_bolunme));
622     red = 0;
623     green = 0;
624     blue = 0;
625
626     for (int c = (i - y_bolunme); c <= (i - 1); c++)
627     {
628         for (int b = 1; b <= (yatay_derinlik_deger + 1); b++)
629         {
630             Color pixelColor = myBitmap.GetPixel(b, c);
631             red += pixelColor.R;
632             green += pixelColor.G;
633             blue += pixelColor.B;

```

```

631         red += pixelColor.R;
632         green += pixelColor.G;
633         blue += pixelColor.B;
634     }
635 }
636 red /= alan;
637 green /= alan;
638 blue /= alan;
639
640 gonderilen[sayac_anlik] = Convert.ToByte(red);
641 sayac_anlik++;
642 gonderilen[sayac_anlik] = Convert.ToByte(green);
643 sayac_anlik++;
644 gonderilen[sayac_anlik] = Convert.ToByte(blue);
645 sayac_anlik++;
646 }
647 //resim ram den siliniyor
648 if (myBitmap != null) { myBitmap.Dispose(); GC.Collect(); }
649 if (serialPort1.IsOpen == true) { serialPort1.Write(gonderilen, 0, gonderilen.Length); }
650 watch.Stop();
651 fps_degeri.Text = watch.Elapsed.Milliseconds.ToString();
652 }
653 }
654
655 1 başvuru
656 private void music_cekirdek_islemi()
657 {
658     //while ile sürekli işlem kılıyoruz
659     while (music_suren_islem)
660     {
661         try
662         {
663             //fps hesaplama
664             Stopwatch watch = new Stopwatch();
665             watch.Start();
666             var device = (MMDevice)ortam_muzik.SelectedItem;
667             int booster = (int)(Math.Round(device.AudioMeterInformation.MasterPeakValue * yükseltme_degeri));
668             if (booster > 100) { booster = 100; }
669             byte music_pwm = Convert.ToByte((booster * 255) / 100);
670
671             if (muzik_seviye_goster.Checked == true) { ortam_music_seviyesi.Value = booster; }
672             //music modu belirleniyor
673             switch (music_modu_degiskeni)
674             {
675                 case 0:
676                     //diziye rgb pwm değerleri byte cinsinden aktarılıyor
677                     for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
678                     {
679                         sayilar[i] = music_pwm;
680                         sayilar[i + 1] = 0;
681                         sayilar[i + 2] = 0;
682                     }
683                     if (serialPort1.IsOpen == true)
684                     {
685                         serialPort1.Write(sayilar, 0, sayilar.Length);
686                     }
687                     break;
688                 case 1:
689                     //diziye rgb pwm değerleri byte cinsinden aktarılıyor
690                     for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
691                     {
692                         sayilar[i] = 0;
693                         sayilar[i + 1] = 0;
694                         sayilar[i + 2] = music_pwm;
695                     }
696                     if (serialPort1.IsOpen == true)
697                     {
698                         serialPort1.Write(sayilar, 0, sayilar.Length);
699                     }
700                     break;
701                 case 2:
702                     music_modu_degiskeni = 2;
703                     //diziye rgb pwm değerleri byte cinsinden aktarılıyor
704                     for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
705                     {
706                         sayilar[i] = 0;
707                         sayilar[i + 1] = music_pwm;
708                         sayilar[i + 2] = 0;

```

```

708         sayilar[i + 2] = 0;
709     }
710     if (serialPort1.IsOpen == true)
711     {
712         serialPort1.Write(sayilar, 0, sayilar.Length);
713     }
714     break;
715 case 3:
716     //diziye rgb pwm deęerleri byte cinsinden aktarılıyor
717     for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
718     {
719         sayilar[i] = music_pwm;
720         sayilar[i + 1] = music_pwm;
721         sayilar[i + 2] = music_pwm;
722     }
723     if (serialPort1.IsOpen == true)
724     {
725         serialPort1.Write(sayilar, 0, sayilar.Length);
726     }
727     break;
728 case 4:
729     music_modu_degiskeni = 4;
730
731     //rastgele deęer çekiliyor true false için
732     Random rastgele = new Random();
733     int sayi = rastgele.Next(0, 10);
734
735     //diziye rgb pwm deęerleri byte cinsinden aktarılıyor
736     for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
737     {
738
739         if (sayi < 5) { sayilar[i] = music_pwm; } else { sayilar[i] = 0; }
740         sayi = rastgele.Next(0, 10);
741         if (sayi < 5) { sayilar[i + 1] = music_pwm; } else { sayilar[i + 1] = 0; }
742         sayi = rastgele.Next(0, 10);
743         if (sayi < 5) { sayilar[i + 2] = music_pwm; } else { sayilar[i + 2] = 0; }
744     }
745     if (serialPort1.IsOpen == true)
746     {
747         serialPort1.Write(sayilar, 0, sayilar.Length);
748         serialPort1.Write(sayilar, 0, sayilar.Length);
749     }
750     break;
751 case 5:
752     //ledler sıralı gidicek.kırmızı
753     music_modu_degiskeni = 5;
754     for (int i = sayilar.Length - 6; i >= 0; i -= 3)
755     {
756         sayilar[i + 5] = sayilar[i + 2];
757         sayilar[i + 4] = sayilar[i + 1];
758         sayilar[i + 3] = sayilar[i];
759     }
760     sayilar[0] = music_pwm;
761     sayilar[1] = 0;
762     sayilar[2] = 0;
763     if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
764     break;
765 case 6:
766     //ledler sıralı gidicek.yeşil
767     music_modu_degiskeni = 6;
768     for (int i = sayilar.Length - 6; i >= 0; i -= 3)
769     {
770         sayilar[i + 5] = sayilar[i + 2];
771         sayilar[i + 4] = sayilar[i + 1];
772         sayilar[i + 3] = sayilar[i];
773     }
774     sayilar[0] = 0;
775     sayilar[1] = music_pwm;
776     sayilar[2] = 0;
777     if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
778     break;
779 case 7:
780     //ledler sıralı gidicek.mavi
781     music_modu_degiskeni = 7;
782     for (int i = sayilar.Length - 6; i >= 0; i -= 3)
783     {
784         sayilar[i + 5] = sayilar[i + 2];
785         sayilar[i + 4] = sayilar[i + 1];
786

```

```

786         sayilar[i + 4] = sayilar[i + 1];
787         sayilar[i + 3] = sayilar[i];
788     }
789     sayilar[0] = 0;
790     sayilar[1] = 0;
791     sayilar[2] = music_pwm;
792     if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
793
794     break;
795 case 8:
796     //ledler sıralı gidicek. beyaz
797     music_modu_degiskeni = 8;
798     for (int i = sayilar.Length - 6; i >= 0; i -= 3)
799     {
800         sayilar[i + 5] = sayilar[i + 2];
801         sayilar[i + 4] = sayilar[i + 1];
802         sayilar[i + 3] = sayilar[i];
803     }
804     sayilar[0] = music_pwm;
805     sayilar[1] = music_pwm;
806     sayilar[2] = music_pwm;
807     if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
808
809     break;
810 case 9:
811     //ledler sıralı gidicek. rastgele
812     music_modu_degiskeni = 9;
813     for (int i = sayilar.Length - 6; i >= 0; i -= 3)
814     {
815         sayilar[i + 5] = sayilar[i + 2];
816         sayilar[i + 4] = sayilar[i + 1];
817         sayilar[i + 3] = sayilar[i];
818     }
819     //rastgele deger çekiliyor true false için
820     rastgele = new Random();
821     sayi = rastgele.Next(0, 10);
822
823     if (sayi < 5) { sayilar[0] = music_pwm; } else { sayilar[0] = 0; }
824     sayi = rastgele.Next(0, 10);
825     if (sayi < 5) { sayilar[1] = music_pwm; } else { sayilar[1] = 0; }
826
827     if (sayi < 5) { sayilar[2] = music_pwm; } else { sayilar[2] = 0; }
828
829     if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
830
831     break;
832 case 10:
833     //ledler sıralı gidicek. rastgele renklersıralı gönderilecek
834     music_modu_degiskeni = 10;
835     for (int i = sayilar.Length - 6; i >= 0; i -= 3)
836     {
837         sayilar[i + 5] = sayilar[i + 2];
838         sayilar[i + 4] = sayilar[i + 1];
839         sayilar[i + 3] = sayilar[i];
840     }
841
842     //if (saydir < 10) { sayilar[0] = music_pwm; saydir++; } else { sayilar[0] = 0; }
843     //if (saydir < 20 && saydir > 10) { sayilar[1] = music_pwm; saydir++; } else { sayilar[1] = 0; }
844     //if (saydir < 30 && saydir > 20) { sayilar[2] = music_pwm; saydir++; } else { sayilar[2] = 0; }
845     //if (saydir > 29) { saydir = 0; }
846
847     if (saydir < 10 && saydir >= 0) { sayilar[0] = music_pwm; saydir++; } else { sayilar[0] = 0; saydir++; }
848     if (saydir > 10 && saydir < 20) { sayilar[1] = music_pwm; saydir++; } else { sayilar[1] = 0; saydir++; }
849     if (saydir > 20 && saydir < 30) { sayilar[2] = music_pwm; saydir++; } else { sayilar[2] = 0; saydir++; }
850     if (saydir > 29) { saydir = 0; }
851     if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
852
853     break;
854 }
855 watch.Stop();
856 fps_degeri.Text = watch.Elapsed.Milliseconds.ToString();
857 }
858 catch { }
859 }
860 }
861 1 baguru
862 private void efekt_cekirdek_islemi()
863 {
864     //while ile sürekli işlem kılıyoruz

```

```

1 baguru
861 private void efekt_cekirdek_islemi()
862 {
863     //while ile sürekli işlem kılıyoruz
864     while (efekt_suren_islem){
865         //try{
866         //fps hesaplama
867         Stopwatch watch = new Stopwatch();
868         watch.Start();
869
870         switch (efekt_modu_degiskeni){
871         case 0:
872             efekt_modu_degiskeni = 0;
873             if (dinlendirme_mod)
874                 //diziye rgb pwm değerleri byte cinsinden aktarılıyor
875                 for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
876                 {
877                     sayilar[i] = 0;
878                     sayilar[i + 1] = 0;
879                     sayilar[i + 2] = 255;
880                 }
881             else
882             { //diziye rgb pwm değerleri byte cinsinden aktarılıyor
883                 for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
884                 {
885                     sayilar[i] = 255;
886                     sayilar[i + 1] = 0;
887                     sayilar[i + 2] = 0;
888                 }
889             }
890
891             if (dinlendirme_mod)
892             {
893                 dinlendirme_mod = false; //oluşturulan bytelar gönderiliyor
894                 if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
895                 Thread.Sleep(100);
896                 for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
897                 {
898                     sayilar[i] = 0;
899                     sayilar[i + 1] = 0;
900                     sayilar[i + 2] = 0;
901                 }
902                 if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
903                 Thread.Sleep(150);
904             }
905             else
906             {
907                 dinlendirme_mod = true;
908                 if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
909                 Thread.Sleep(100);
910                 for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
911                 {
912                     sayilar[i] = 0;
913                     sayilar[i + 1] = 0;
914                     sayilar[i + 2] = 0;
915                 }
916                 if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
917                 Thread.Sleep(150);
918             }
919             break;
920         case 1:
921             efekt_modu_degiskeni = 1;
922             if (dinlendirme_mod)
923                 //diziye rgb pwm değerleri byte cinsinden aktarılıyor
924                 for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
925                 {
926                     sayilar[i] = 255;
927                     sayilar[i + 1] = 255;
928                     sayilar[i + 2] = 255;
929                 }
930             else{ //diziye rgb pwm değerleri byte cinsinden aktarılıyor
931                 for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
932                 {
933                     sayilar[i] = 255;
934                     sayilar[i + 1] = 0;
935                 }
936             }
937         }
938     }
939 }

```

```

935     for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
936     {
937         sayilar[i] = 255;
938         sayilar[i + 1] = 0;
939         sayilar[i + 2] = 0;
940     }
941 }
942
943
944
945 if (dinlendirme_mod){
946     dinlendirme_mod = false; //oluşturulan bytelar gönderiliyor
947     if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
948     Thread.Sleep(100);
949     for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
950     {
951         sayilar[i] = 0;
952         sayilar[i + 1] = 0;
953         sayilar[i + 2] = 0;
954     }
955     if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
956     Thread.Sleep(150);
957 }
958 else{
959     dinlendirme_mod = true;
960     if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
961     Thread.Sleep(100);
962     for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3)
963     {
964         sayilar[i] = 0;
965         sayilar[i + 1] = 0;
966         sayilar[i + 2] = 0;
967     }
968     if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
969     Thread.Sleep(150);
970 }
971
972 break;
973 case 2:
974     efekt_modu_degiskeni = 2;
975
976     break;
977
978 case 2:
979     efekt_modu_degiskeni = 2;
980     if (dinlendirme_mod) { music_modu_degiskeni = 255; } else { music_modu_degiskeni = 0; }
981     //diziye rgb pwm değerleri byte cinsinden aktarılıyor
982     for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3){
983         sayilar[i] = 0;
984         sayilar[i + 1] = 0;
985         sayilar[i + 2] = music_modu_degiskeni;
986     }
987
988     if (dinlendirme_mod) { dinlendirme_mod = false; //oluşturulan bytelar gönderiliyor
989         if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
990         Thread.Sleep(50);
991     }else{
992         dinlendirme_mod = true;
993         if (serialPort1.IsOpen == true) { serialPort1.Write(sayilar, 0, sayilar.Length); }
994         Thread.Sleep(100);
995     }
996
997     break;
998
999 case 3:
1000     efekt_modu_degiskeni = 3;
1001     //dinlendirici mod
1002
1003     if (music_modu_degiskeni <= 0){
1004         dinlendirme_mod = true;
1005     }
1006     if (music_modu_degiskeni >= 255) {
1007         dinlendirme_mod = false;
1008     }
1009     if (dinlendirme_mod){
1010         music_modu_degiskeni += 1;
1011     }else { music_modu_degiskeni -= 1; }
1012
1013     //diziye rgb pwm değerleri byte cinsinden aktarılıyor
1014     for (int i = 0; i < ((Settings.Default.dikey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3){
1015         sayilar[i] = music_modu_degiskeni;
1016         sayilar[i + 1] = music_modu_degiskeni;
1017     }

```

```

1010         sayilar[i + 1] = music_modu_degiskeni;
1011         sayilar[i + 2] = music_modu_degiskeni;
1012     }
1013
1014     pwm_uyku.Value = music_modu_degiskeni;
1015     //oluşturulan byteler gönderiliyor
1016     if (serialPort1.IsOpen == true){serialPort1.Write(sayilar, 0, sayilar.Length);}
1017     break;
1018 case 4:
1019     efekt_modu_degiskeni = 4;
1020     //zaman bitene kadar
1021
1022     if (zaman_say.Elapsed.TotalMinutes < Convert.ToInt32(uyku_suresi.Value)) {
1023         //global bir değişken pwm olarak atanacak (music_modu_degiskeni = 255; case bölümüne yazıldı)
1024         //bu modu music_modu kullanılmadığı için music_degiskenini kullanacağız (pwm tesarrufu için)
1025         kalan_dk.Text = (Convert.ToInt32(uyku_suresi.Value) - zaman_say.Elapsed.TotalMinutes).ToString("0.###");
1026         if (uyku_sure_azalt.Checked == true) {
1027             //oran oranti ile sn hesapları ile gereken pwm bulunuyor
1028             music_modu_degiskeni = Convert.ToByte((((Convert.ToInt32(uyku_suresi.Value)*60) - zaman_say.Elapsed.TotalSeconds)*255)/ (Convert.ToInt32(uyku_suresi.Value) * 60));
1029         }
1030         else { music_modu_degiskeni = 255; }
1031         //diziye rgb pwm değerleri byte cinsinden aktarılıyor
1032         for (int i = 0; i < ((Settings.Default.dkey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3){
1033             sayilar[i] = music_modu_degiskeni;
1034             sayilar[i + 1] = music_modu_degiskeni;
1035             sayilar[i + 2] = music_modu_degiskeni;
1036         }
1037         pwm_uyku.Value = music_modu_degiskeni;
1038         //oluşturulan byteler gönderiliyor
1039         if (serialPort1.IsOpen == true){
1040             serialPort1.Write(sayilar, 0, sayilar.Length);
1041         }
1042     }
1043     else{
1044         kalan_dk.Text = "dk bitti.";
1045         //diziye rgb pwm değerleri byte cinsinden aktarılıyor
1046         for (int i = 0; i < ((Settings.Default.dkey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3){
1047             sayilar[i] = 0;
1048             sayilar[i + 1] = 0;
1049             sayilar[i + 2] = 0;
1050         }
1051         pwm_uyku.Value = music_modu_degiskeni;
1052         //oluşturulan byteler gönderiliyor
1053         if (serialPort1.IsOpen == true){
1054             serialPort1.Write(sayilar, 0, sayilar.Length);
1055         }
1056     }
1057     break;
1058 case 5:
1059     }
1060     break;
1061 case 5:
1062     efekt_modu_degiskeni = 5;
1063     byte gecici_pwm = 0; //ilk pwm
1064     byte bekleme = 5; //ilk bekleme süresi
1065     //do while ile sürekli dönen 2 farklı işlemi tek işleme düşürdük
1066     do{
1067         //diziye rgb pwm değerleri byte cinsinden aktarılıyor
1068         for (int i = 0; i < ((Settings.Default.dkey * 2) + (Settings.Default.yatay * 2)) * 3; i = i + 3){
1069             sayilar[i] = gecici_pwm;
1070             sayilar[i + 1] = gecici_pwm;
1071             sayilar[i + 2] = gecici_pwm;
1072         }
1073         //oluşturulan byteler gönderiliyor
1074         if (serialPort1.IsOpen == true){
1075             serialPort1.Write(sayilar, 0, sayilar.Length);
1076         }
1077         //yeni flash göndermeden önce bekleme süresi
1078         Thread.Sleep(bekleme);
1079         //bekleme ile aynı zamanda do while kontrol ediyor ikinci döngüden sonra bekleme arttırılıp
1080         //do whileden çıkması sağlanıyor
1081         if (bekleme > 19) { bekleme = 200; } else{
1082             gecici_pwm = 255; //ikinci pwm
1083             bekleme = 20; //ikinci bekleme süresi
1084         }
1085     } while (bekleme < 199);
1086     break;
1087
1088     watch.Stop(); // fps hesaplama için watch durduruluyor
1089     fps_degeri.Text = watch.Elapsed.Milliseconds.ToString(); //saniyedeki döngü sayısının ms cinsinden 1000ms bölüp fps (frame per second) buluyoruz
1090     //}catch { }
1091 }
1092 }
1093 }
1094

```

Link 2: Projenin çalışma videosu

“<https://www.youtube.com/watch?v=wWtOZFQ5inQ>”

KAYNAKÇA

- [1.] <https://docs.microsoft.com/en-us/dotnet/api/system.drawing.bitmap.setpixel?view=netframework-4.7.2>
- [2.] https://www.tutorialspoint.com/csharp/csharp_multithreading.htm
- [3.] <https://docs.microsoft.com/en-us/dotnet/api/system.threading.thread?view=netframework-4.7.2>
- [4.] <https://docs.microsoft.com/en-us/dotnet/api/system.io.ports.serialport?view=netframework-4.7.2>
- [5.] https://cdn.sparkfun.com/assets/c/a/8/e/4/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf
- [6.] <https://docs.microsoft.com/en-us/dotnet/api/system.windows.media.color.fromrgb?view=netframework-4.7.2>
- [7.] <https://github.com/rosterloh/ambilight>
- [8.] <https://github.com/mpolednik/script.kodi.hue.ambilight>
- [9.] <https://github.com/michaelficarra/ambilight>
- [10.] <https://cdn.sparkfun.com/datasheets/Dev/Arduino/Other/CH340DS1.PDF>
- [11.] <https://www.arduino.cc/en/tutorial/arduinoISP>

ÖZGEÇMİŞ



MERT ÇOLAK 1994'de VOLGOGRAD / RUSYA FED.'da doğdu; ilk, orta ve lise öğrenimini K.MARAŞ / Elbistan'da tamamladı; 2015 yılında Karabük Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği Bölümüne girdim halende bu bölümde eğitimime devam etmekteyim. Mekatronik Mühendisliğinin, kontrol ve yazılım alanları ile ilgilenmektedir. İş disiplinine sahip, verilen görevi zamanında yerine getiren, sosyal yönü kuvvetli, iletişim becerilerinde iyi bir kişiliğe sahiptir. İyi seviyede İngilizce bilmektedir.

İletişim Bilgileri

E-posta: info@mertcolak.com.tr

Website: mertcolak.com.tr

Diğer Bağlantılar:

<https://github.com/mmerttccolakk>

<https://easyeda.com/mmerttccolakk>

<https://tr.linkedin.com/in/mmerttccolakk>