

## Gömülü Sistemler İçin Bilgisayar Tabanlı Grafıksel Kullanıcı Ara yüzü Tasarım Aracı Geliştirilmesi

*Development of Computer Based Graphical User Interface Design Tool For Embedded Systems*

Cihan YILDIRIM<sup>1</sup> , Veysel Gökhan BÖCEKÇİ<sup>1</sup> 

<sup>1</sup>Marmara Üniversitesi, Teknoloji Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, 34722, İstanbul, Türkiye

### Öz

Günümüzde kullanılmakta olan birçok elektronik eşya dokunmatik ekran özellikli bir yapıya sahip olarak kullanıcılara sunulmaktadır. Dokunmatik ve renkli grafik ekranlar ile ürünlerdeki kalite algısı yükseltilirken aynı zamanda işlevselliği de artırılmaktadır. TFT ekranlı ürünlerin kullanımındaki artışla birlikte maliyetlerde oluşan azalmalar, üreticileri TFT ekranlı ürün üretme konusunda daha çok teşvik etmektedir. Bununla birlikte özellikle gömülü sistemlerde TFT ekranlı ürünlerin AR-GE süreçleri oldukça zahmetlidir. Her ürün için ayrı görsel çalışmalar yapıyor olması ve bunların gömülü sisteme aktarılması projelerde adam-ay artışına neden olmaktadır. Bu alandaki yazılımsal AR-GE süreçlerini hızlandırmak için modüler ve kolay kullanıma sahip bir grafıksel kullanıcı ara yüzü (GUI) kütüphanesi kullanmak oldukça avantajlı olmaktadır. Yapılan çalışmada Excel makroları, VB Script kodları ve Python gibi yazılımlardan faydalanarak temelde bir Excel platformu üzerinde gömülü sistemlerde kullanılmak üzere TFT ekranlar için genel bir ekran tasarım yapısı oluşturulmuştur. Bu yapı sayesinde Excel üzerinde TFT ekranın içerik tasarımı sayfalar halinde tasarlanabilmekte ve gömülü sisteme aktarabilmek için gerekli binary dosyası üretilebilmektedir. Gömülü sistem tarafındaki yazılım ise PC ortamında oluşturulan ekran tasarımlarının bulunduğu binary dosyasını SPI veya UART hattından aldığı komutlar ile harici bir flash bellekten okuyarak ekranda gösterebilmektedir. Böylece TFT ekranlı bir ürün geliştirirken Ar-Ge sürecinde oluşabilecek birçok tasarım PC ortamında oluşturulup harici flash içerisine indirgenemediğinden gömülü sistemler için genel bir tasarım aracı elde edilebilmiştir. Oluşturulan yazılım elektromanyetik uyumluluk, yayılım ve iklimlendirme gibi çeşitli testleri başarıyla geçmiştir. Piyasadaki popüler olarak kullanılan yabancı menşeli alternatif ürünler ile kıyaslandığında ise üretim maliyetleri ve kullanım yeterlilikleri yönünden avantaj sağladığı görülmüştür.

**Anahtar kelimeler:** gömülü gui kütüphanesi, tft ekranlar, resim ölçeklendirme, alfa harmanlama, parlaklık ayarlama.

### Abstract

Many electronic goods in use today are offered to users with a touch screen-enabled structure. With touch and color graphic screens, the perception of quality in the products is increased while at the same time its functionality is increased. With the increase in the use of TFT screen products, the reduction in costs encourages manufacturers to produce TFT screen products more. However, the R&D processes of TFT display products, especially in embedded systems, are quite laborious. The fact that separate visual studies are carried out for each product and their transfer to the embedded system causes an increase in man-months in projects. It is very advantageous to use a modular and easy-to-use graphical user interface (GUI) library to accelerate software R&D processes in this area. In the study, a general screen design structure was created for TFT screens to be used in embedded systems on an Excel platform, by making use of software such as Excel macros, VB Script codes and Python. Thanks to this structure, the content design of the TFT screen on Excel can be designed as pages and the necessary binary file can be produced to transfer it to the embedded system. On the other hand, the software on the embedded system can display the binary file containing the screen designs created in the PC environment, by reading it from an external flash memory with the commands it receives from the SPI or UART line. Thus, a general design tool for embedded systems has been obtained, since many designs that may occur in the R&D process while developing a product with TFT screen can be created in the PC environment and reduced to external flash. The created software has successfully passed various tests such as electromagnetic compatibility, emission and air conditioning. When compared with alternative products of foreign origin, which are popularly used in the market, it has been observed that it provides an advantage in terms of production costs and usage capabilities.

**Keywords:** embedded gui library, tft screen, image scaling, alpha blending, opacity adjustment.

## I. GİRİŞ

Grafıksel kullanıcı arayüzü (GUI) insan makine etkileşimini görsel bileşenlerle kolaylaştıran yazılımlardır. Makineler ile iletişim aracı olarak kullanılan GUI yazılımları, bilgi ileten nesnelere ve kullanıcı tarafından gerçekleştirilebilecek eylemleri temsil eder. Kullanıcı bu nesnelere etkileşime geçtiğinde nesnelere genellikle renk, boyut veya görünüm olarak değişir.

İlk GUI prototip çalışmaları 1979'daki Xerox Palo Alto Araştırma Merkezine (PARC'a) dayanmaktadır. Apple şirketi PARC'ın araştırmalarına büyük ilgi duymuştur ve bu çalışmalardan ilham alan (Apple şirketi kurucu ortaklarından) Steve Jobs, attığı adımlarla kısa süre içinde Apple'in kişisel bilgisayarlarda GUI'leri etkin olarak kullanabilen ilk bilgisayar şirketi olmasında kilit rol oynamıştır [1].

O dönem Apple'ın rakibi IBM ise ilk kişisel bilgisayarını 1981'de piyasaya sürdüğünde Microsoft'un MS-DOS işletim sistemi ile çalışıyordu ve kullanıcıların bilgisayarda işlem yapmak için DOS komutlarını bilmesi ve bunları her seferinde elle girmesi gerekiyordu. Örneğin sabit diskteki bir dosyayı bir yerden başka bir yere taşımak için dosya adını ve taşınacak yolu içeren çeşitli DOS komutlarını girmek gerekiyordu. Günümüzde (GUI'lerin avantajı sayesinde) fare ile basitçe sürükle-bırak şeklinde yaptığımız bir işlemi o dönem DOS ortamında metinsel komutlarla yapmak oldukça zahmetli olmaktadır. Bundan üç yıl sonra Xerox'un da desteğiyle Apple, son derece başarılı Macintosh'unu çıkararak kişisel bilgisayarlarda grafiksel kullanıcı ara yüzlerinin (GUI'lerin) etkin bir şekilde kullanılabilmesini sağlamıştır [2].

1980'li yıllardan günümüze kadar olan süreçte insan makine etkileşiminde çok önemli rol oynayan GUI yazılımları büyük gelişmeler göstermiştir. DOS komut sistemli kullanımı son derece zahmetli işletim sistemlerinden, kullanıcının en az zahmetle en fazla işi yapabildiği kullanıcı dostu işletim sistemlerine kadar uzanan bu süreçteki gelişmelerde insan makine etkileşimi noktasında GUI yazılımlarının katkısı büyüktür.

Günümüzde GUI tasarımlarında modülerlik, platform bağımsız özellik, güvenilirlik, kolay yapılandırılabilirlik düşük maliyet, kişiselleştirilebilirlik, ergonomik yapı, düşük bellek kullanımı odaklanılan alanlar olarak dikkat çekmektedir [3-6].

Endüstriyel uygulamalarda GUI tasarımının yerel ve uzaktan görüntüleme ile kontrol özelliklerinin olması ihtiyaç duyulan kabiliyetler olarak öne çıkmaktadır. Örnek uygulamaları arasında güvenlik, yazılım güncelleme, enerji tüketimi tasarrufu bulunmaktadır [7-9].

GUI ile endüstriyel uygulamalar arasındaki etkileşimin verimini arttırmak için GUI performanslarını geliştirmeye yönelik otomatik test yazılımları da ayrıca geliştirilmektedir. Buradaki başlıca amaç zaman tasarrufu ve el ile test gibi personel gerektiren durumların önüne geçmektir [10].

Gelişen internet ve yazılım alt yapısıyla birlikte konuşma tanıma ve doğal dil işlemedeki güçlü ilerlemelerden yararlanarak yeni grafiksel kullanıcı ara

yüzlerinin her zamankinden daha sezgisel ve etkili olacağı ön görülmektedir [11]. Henüz İngilizce dili ile sınırlı olarak doğal dil işleme yönteminin kullanıldığı metni konuşmaya çeviren sayısal işaret işleme modülleri ile grafiksel kullanıcı arayüzleri geliştirilmektedir. İleride bu tip sistemlerin çoklu dil desteklerinin olacağı ve gerçek zamanlı olarak web tabanlı uygulamalarının da bulunacağı öngörülmektedir [12]. Görme engelliler bireyler için özellikle ses temelli GUI aygıtlarının geliştirilmesine yönelik alt yapı çalışmaları yapılmaktadır [13].

Gömülü sistemler bazında yeni nesil mikroişlemcilerin belirli bir kısmının ses ve görüntü işleme donanımları barındırma sebepleri geleceğin kullanıcı ara yüzlerinin ilerlediği noktayı daha iyi göstermektedir [14].

Günümüzde sanal gerçeklik (VR) ve artırılmış gerçeklik (AR) tasarımı, tamamen yeni bir endüstri olarak geleceğin kullanıcı ara yüzü zeminini oluşturmaktadır. Özellikle son birkaç yıldır VR ve AR pazarı Microsoft, Google, HTC, Facebook ve hatta Apple gibi büyük teknoloji şirketleri tarafından büyük ilgi görmektedir [15].

Yapay zekâ, istikrarlı bir şekilde büyümek ve değişen iş ortamına ayak uydurmak isteyen şirketler için lüks değil, olmazsa olmaz bir teknoloji haline gelmiştir. Amazon, Apple, Google, Microsoft vb. teknoloji devi şirketler kullanıcı ara yüzü dâhil birçok konuda insan makine etkileşiminde yapay zekâdan faydalanmaktadır [16].

### 1.1. Kullanıcı Ara yüzü Tasarımı

Kullanıcı ara yüzü (UI) tasarımı, bilgisayar, ev aletleri, mobil cihazlar ve diğer elektronik cihazlar gibi makineler ve yazılımlar için kullanıcı deneyimini en üst düzeye çıkarmaya odaklanan tasarımıdır. Kullanıcı ara yüzü tasarımının temel amacı, kullanıcı hedeflerini (kullanıcı merkezli tasarım) gerçekleştirme açısından kullanıcının etkileşimini olabildiğince basit ve verimli hale getirmektir.

### 1.2. Grafiksel Kullanıcı Ara yüzü (GUI)

Grafiksel kullanıcı ara yüzü (GUI) objeleri genellikle ikonlar, kursörler ve butonlardan oluşur. Bu grafiksel nesnelere bazen ses ya da görsele etkilerle zenginleştirilir. Saydamlaştırma ya da gölge düşürme gibi efektler sıklıkla kullanılan görsel efektler arasındadır [17].

### 1.3. Gömülü Sistemlerde GUI Tasarımlarının Önemi

GUI tasarımlarının PC üzerinden jenerik bir yapı ile oluşturulabilmesi ve mikrodenetleyicili gömülü sistem donanımı tarafına kolayca aktarılabilmesi, kod geliştiriciyi her yeni projede çekirdekte yürütülen düşük seviyeli yazılımlar ile çalışmaktan kurtaracağı için rahat ve hızlı bir ürün tasarım süreci elde edilmiş olur.

Jenerik tasarıma sahip GUI kütüphanelerinin kullanılmadığı TFT ekranlı AR-GE ürün çalışmalarının adam/ay süresi ve bunun işletmelere olan toplam maliyeti ile gelişmiş bir GUI tasarım aracı kullanılarak gerçekleştirilen projelerin tasarım süreci karşılaştırıldığında bu çalışma ile üreticinin ciddi maddi kazançlar elde ettiği görülmüştür.

Yüksek teknoloji ürünleri hızlı çıkarmanın işletmeye sağladığı maddi katma değerinin ve ürün kalitesindeki artışın getireceği rekabet gücünün yanı sıra, geliştirilen GUI kütüphanesinin sağladığı bazı yazılımsal kazanımlar sayesinde daha düşük kapasiteli işlemci ve hafıza birimleri kullanılarak daha yüksek kalitede ürünler çıkarabilmek mümkün olmaktadır. Bu sayede düşük donanım ve yazılım maliyetleri ile daha yüksek katma değere sahip ürünlerin geliştirilebilmesi mümkün olmaktadır.

## II. MATERYAL VE YÖNTEM

Yapılan çalışmanın PC ve gömülü sistem tarafı olmak üzere iki temel ayağı vardır. Gömülü sistem kısmı da kendi içinde gömülü donanım ve yazılım olarak iki ana başlık altında incelenecektir. PC tarafında Python ve Visual Basic Script dilleri kullanılmıştır. Ekran içerik listelerini oluşturmak için Excel 'den yararlanılmıştır. Visual Basic Script'i Excel'e gömülü olarak çalıştırılarak ekran verilerinin oluşturulmasında kolaylık sağlanmıştır. Excel'de gömülü olarak çalışabilme özelliğinden dolayı Visual Basic tercih edilmiştir. Python dili ise oldukça pratik kullanıma ve yüksek seviyeli bir dil yapısına sahip olduğu için tercih edilmiş olup bu dil ile kodlanan yazılım, gömülü sisteme aktarılacak olan binary (.bin) data dosyalarının oluşturulmasında ve resimlerin sıkıştırılmasında kullanılmıştır.

Gömülü sistemin yazılım tarafında C dili kullanılmış olup mikro denetleyicinin kod geliştirme ortamı olarak Atmel Studio IDE'si kullanılmıştır. C derleyicisi olarak

da maliyet unsuru göz önünde bulundurularak GCC C derleyicisi tercih edilmiştir. Gömülü sistemin donanım tarafında ise düşük maliyet ile alınabilecek maksimum performansa ve çevre birimlerinin yeterliliğine odaklanılarak ATSAMD51J20A mikrodenetleyicisi seçilmiştir.

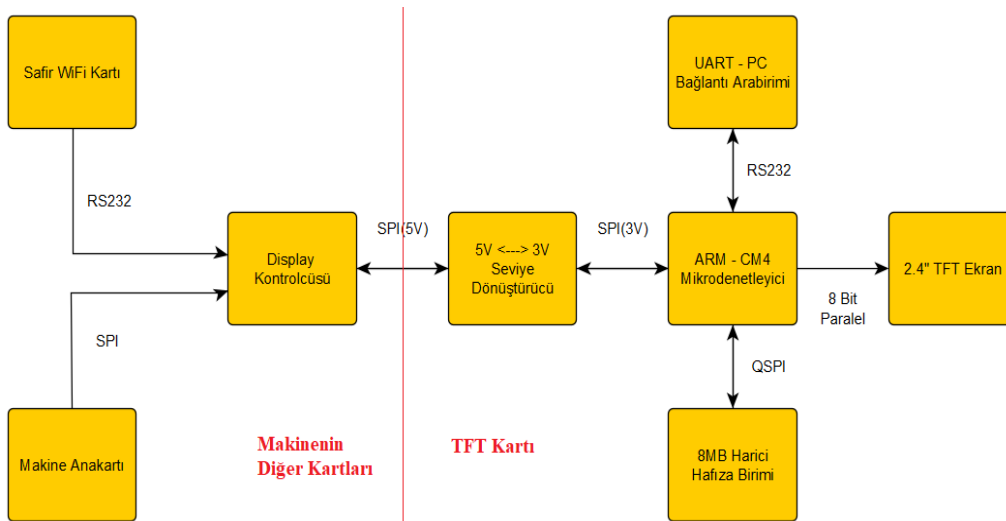
### 2.1. Gömülü Sistemin Genel Donanım Yapısı

Şekil 1'de projenin donanım yapısının blok diyagramı görülmektedir. Yapılan bu çalışma aynı zamanda uluslararası bir firmanın beyaz eşya makinelerinde kullanılacak şekilde tasarlanmıştır.

Devrede ARM Cortex-M4 (CM4) çekirdekli fiyat/performans yönünden başarılı sayılabilecek ATMEL marka bir MCU kullanılması uygun görülmüştür.

ATSAMD51J20A, 120Mhz saat frekansına sahip olup 256Kb Ram ve 1Mb dâhili Flash hafıza birimine sahiptir. TFT ekran uygulamalarında yüksek hızla birlikte yüksek Ram ve Flash hafıza ihtiyacı da büyük önem arz ettiği için CM4 serisi içerisinde bu yönden tatmin edecek uygun fiyatlı bir MCU seçilmeye özen gösterilmiştir.

Ayrıca ekran görsellerini ve bir takım ürüne özel dataları saklamak için de 8Mb harici seri hafıza entegrasyonu kullanılmıştır. MCU, seri haberleşme portu ile PC 'den aldığı ekran datalarını QSPI (Quad Serial Peripheral Interface) haberleşmesi ile harici hafıza birimine kaydeder. Uygulama yazılımı ise harici hafızadaki bu dataları okuyarak ekranı günceller. Devre ayrıca farklı MCU'lardan (Örneğin; Şekil 1 'deki Display kontrolcüsünden) komut alabilmek için SPI (Serial Peripheral Interface) portu üzerinden harici haberleşme sağlayabilmektedir. Çalışma için seçilen TFT ekran ise ILI9340 sürücülü 2.4 inch'lik bir TFT ekrandır. 320x240 çözünürlüğe sahip bu TFT ekran ile MCU 8080 paralel 8 bit veri yolu üzerinden haberleşmektedir.



Şekil 1. Gömülü TFT kartı blok şeması

## 2.2. Yazılım

Gömülü devre üzerindeki MCU, Atmel serisi ARM çekirdekli bir denetleyici olduğu için Atmel Studio IDE'si kullanılarak C dilinde kod geliştirilmiştir. Ücretsiz derleyici olması sebebiyle GCC C derleyicisi tercih edilmiştir. Mikrodenetleyici yazılımı uygulama kodu ve bootloader kodu olarak iki parçadan oluşmaktadır.

Bootloader yazılımı, harici ya da dâhili hafızadaki kodları güncellemek için kullanılır. PC üzerinde tasarlanan ekran dataları bu yazılım vasıtasıyla harici hafızaya kaydedilir. Daha sonra harici hafıza entegresindeki senaryoya göre UART veya SPI üzerinden başka kartlardan gelen komutlar doğrultusunda MCU uygulama yazılımı ekran görsellerini oluşturur.

Yazılımın ikinci ayağı olan PC yazılımı ise GUI ekran tasarımlarının kolay ve pratik bir şekilde yapılabilmesi ve bu ekran tasarımlarının elektronik devre tarafında işlenebilmesi için gerekli ikilik (binary) data dosyasının oluşturulmasını sağlamaktadır.

Görsellerin daha az yer kaplaması için Python ile bir sıkıştırma yazılımı geliştirilmiştir bunun hakkında detaylı bilgi bölüm 2.4 'de verilmiştir. Bu sıkıştırma yazılımının temel mantığı tekrar eden verilerin paketlenmesi üzerinedir. İkon, logo, resim vb. fazla data içeren objeler belirli bir yöntem ile sıkıştırılarak harici hafıza biriminde daha az yer kaplaması sağlanmıştır. Gömülü taraftaki yazılım ise bu dataları açarak ekrana gönderecek şekilde kodlanmıştır. Sıkıştırma yapılıp yapılmaması kullanıcının tercihine bırakılmıştır.

## 2.3. Yöntem

Projenin PC tarafında Excel uygulamasına entegre olarak çalışan Visual Basic Script'i ve bunlara girdi ve çıktılar sağlayan Python kodları yazılmıştır. TFT'de gösterilecek ekranların içerik bilgilerinin tamamı Excel ortamında makroların sağladığı kolaylıklar ile hazırlanabilmektedir. Ayrıca girdilerin tamamına yakını tablo formunda girildiği için özel bir GUI yazılımı yapmaya ihtiyaç duyulmamıştır.

Excel üzerinde ekran tasarımı yapılırken Tablo 1'deki gibi belli bir sıra takip edilmiştir. İlk etapta başlık bölgesi (Header Definations) tanımlanmıştır. Başlık bölgesinde ürüne özel versiyon numaraları, display tipi, boyutu, yerleşim şekli, ön tanımlı dil seçeneği ve toplam ekran sayısı gibi bilgiler yer almaktadır.

Buradaki ekran sayısından kasıt geliştirilmek istenen gömülü ürün için Excel üzerinde kaç farklı ekran tasarımı yapılacağı ile ilgilidir. Sonraki etapta ekran listesi (Screen List) tanımlanmıştır. Bu listede hangi ekranın hafıza biriminde hangi adreste yer alacağı ile ilgili bilgiler yer almaktadır. Bunu arka plan

(background) bilgilerinin yer aldığı bölgenin adres kaydı oluşturulmuştur. Ekranlardaki objelere geçmeden önce objelerin yerleşeceği sınırların adedi ve listesi yer almaktadır. Daha sonra sınır değer tanımlamalarının bulunduğu bölge tabloda yer almaktadır. Sınır değer tanımlamaları ekranda kayar menü objeleri kullanıldığında bu objelerin hangi çerçeve sınırları içerisinde var olacağını bildirmek için kullanılan tanımlamaları içermektedir. Sınır tanımlamalarının (Border Definations) ardından Objelerin ön tanımlı bilgileri yer alır. Bu ön tanımlı bilgiler objenin saydamlık değeri, boyut ölçeği ve diğer dil seçeneklerinin yer aldığı adres bölgesinin başlangıç adresi gibi bilgilerden oluşmaktadır.

Animasyon sayısı ve animasyon verileri opsiyonel olarak tanımlanmaktadır. Bu bölge tanımlanan ekrana özel varsa animasyon parametrelerinin saklandığı bölgedir. Animasyon yoksa buraya "0" değeri verilmektedir. Gömülü sistem SPI ya da UART üzerinden gönderdiği komutlar ile burada ekrana tanımlı animasyonları aktif ya da pasif edebilmektedir. CRC doğrulama kodu Animasyon tanımlamalarından sonra bir bayt olarak yerleştirilmiştir. Bu bir baytlık veri harici hafızaya o sayfanın ön tanım verilerinin doğru yerleşip yerleşmediğini kontrol etmek için kullanılmaktadır. Bu bölümden sonra objelerin farklı dillerdeki adreslerinin bulunduğu liste yer almaktadır. Seçili olan dile göre ilgili objenin dil seçeneğinin bulunduğu adrese ulaşılabilir. Objelerin dillere göre tipi, rengi, ekrandaki konumu, boyutu ve obje verilerinin bulunduğu bölgenin adresi sırası ile bu bölgede listelenmektedir. En son kısımda da arka plan objesinin detaylı bilgileri yer almaktadır. Tüm bu listenin özeti Tablo 1 'de görülmektedir. Excel içine eklediğimiz Visual Basic Scripti ile bin dosyası oluşturulur.

TFT ekranda objeleri kaydırma, sıkıştırma, saydamlığını değiştirme ve resim boyutlarını değiştirme gibi birtakım özelliklerden faydalanılmıştır. Tüm bu özellikler Tablo 1'de girilen obje ön tanımları ve gömülü sisteme seri veri yolu üzerinden gönderilen komutlar ile aktif pasif yapılabilmektedir.

## 2.4. Resim ve Metin Verilerinin Oluşturulması

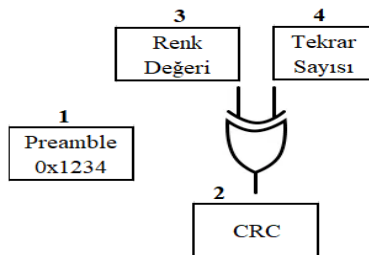
Her bir sayfayı Tablo 1 'deki yerleşim planına göre Excel üzerinden tamamlandıktan sonra en sona Text ve Image adında iki sayfa daha eklenmiştir. Burada tüm sayfalarda tanımlı metinsel ve resimsel objelerinin içeriği hakkında bilgiler bulunmaktadır. Python kodu yardımıyla metinsel (text) ifade listesini UTF-16 ve küçük sonlu (little endian) formata çevirerek metinsel içerik için ayrı bir ".bin" dosyası üretilmektedir. Resim (image) kısmında da görsellere göre dosya boyutunu ve sıkıştırılmış binary dosyaları üreten bir python kodu kullanılmaktadır. Resim verileri hafızadan yer kazanabilmek için sıkıştırılarak saklanabilmektedir.

**Tablo 1.** Harici hafıza organizasyon şeması

HARİCİ HAFIZA ORGANİZASYON ŞEMASI				
	Açıklamalar 1/2	Byte	Açıklamalar 2/2	Byte
BAŞLIK TANIMLARI	Major Parameter version	1	Object_0 Language_0 Data Address	4
	Mini Parameter version	1	...	...
	Micro Parameter version	1	...	...
	Product Type	1	...	...
	Display Width Value	2	Object_0 Language_k Data Address	4
	Display Height Value	2	...	...
	Display Type	1	...	...
	Display Position (Horizontal)	1	Object_n Language_0 Data Address	4
	Default Language (Language_0)	1	...	...
	Total Screen Count	1	...	...
EKRAN LİSTESİ	Screen_0 Address	4	...	...
	...	...	Object_n Language_k Data Address	4
	Screen_n Address	4	Object_0 Language_0 Type	1
SINIR DEĞER TANIMLARI	Background Address	4	Object_0 Language_0 Color	2
	Border Count	1	Object_0 Language_0 X Point	2
	Border_0 X	2	Object_0 Language_0 Y Point	2
	Border_0 Y	2	Object_0 Language_0 Data Length	4
	Border_0 W	2	Object_0 Language_0 Data Address	4
	Border_0 H	2	...	...
	Border_0 Object Count	1	Object_n Language_0 Type	1
	...	...	Object_n Language_0 Color	2
	...	...	Object_n Language_0 X Point	2
	...	...	Object_n Language_0 Y Point	2
OBJE ÖN TANIMLARI	Border_n X	2	Object_n Language_0 Data Length	4
	Border_n Y	2	Object_n Language_0 Data Address	4
	Border_n W	2	...	...
	Border_n H	2	...	...
	Border_n Object Count	1	...	...
	Object_0 Default Opacity Value	1	Object_0 Language_k Type	1
	Object_0 Default Scale Ratio	1	Object_0 Language_k Color	2
	Object_0 Default Address	4	Object_0 Language_k X Point	2
	...	...	Object_0 Language_k Y Point	2
	...	...	Object_0 Language_k Data Length	4
ANIMASYON TANIMLARI	Object_n Default Opacity Value	1	Object_0 Language_k Data Address	4
	Object_n Default Scale Ratio	1	...	...
	Object_n Default Address	4	Object_n Language_k Type	1
	Animation Count	1	Object_n Language_k Color	2
	Animation_0 Length	1	Object_n Language_k X Point	2
	Animation_0 Data	1	Object_n Language_k Y Point	2
	...	...	Object_n Language_k Data Length	4
	Animation_n Length	1	Object_n Language_k Data Address	4
	Animation_n Data	1	Background Type	1
	CRC	1	Background Color	2
ARKAPLAN TANIMLARI			Background X Point	2
			Background Y Point	2
			Background Data Length	4
			Background Data Address	4

## 2.5. Resim Sıkıştırma Algoritması

Resim ve ikon gibi çok sayıda pikselden oluşan objelerin hafıza biriminde az yer kaplaması için sıkıştırma yaparak boyutlarının küçülmesini sağlamak özellikle GUI uygulamaları gibi bol görsel içeren çalışmalarda oldukça faydalı olmaktadır.



Şekil 2. Resim sıkıştırma algoritması

Şekil 2'de uygulanan resim sıkıştırma algoritmasının yapısal blokları görülmektedir. Sıkıştırılmak istenen resimde ne kadar çok ardışık ortak renk varsa o kadar çok sıkıştırma yapılabilir. Sıkıştırma algoritması

kullanıldığında resim içeriğine bağlı olmak koşuluyla resmin data boyutlarında ortalama %30 veya daha fazla küçülme sağlandığı görülmüştür. Bu da harici hafızadan büyük oranda yer kazanılmasını sağladığı gibi bootloader üzerinden içerik güncellemesi yapılırken yükleme süresini ve indirilen resim datalarının boyutunu da aynı oranda küçültmektedir.

Tekrar eden renklerin renk sayısı ile tekrar sayısının özel veya kapısından geçirilmesi ile CRC doğrulama kodu elde edilir. Bu üç veri de 2 şer baytlık uzunluktadır. Öncesinde de senkronizasyon için başlangıç (preamble) amaçlı sabit bir veri (0x1234) gönderilir. Tüm veriler küçük sonlu (Little Endian) formatında tutulduğu için başlangıç (preamble) verisi 0x1234 olarak değil 0x3412 şeklinde saklanacaktır. Bu dört adet ikişer baytlık veri dizisi Şekil 2 'de verilen numara sırasına göre yerleştirilir. Toplam 8 baytlık veri dizisinin tekrarlı verilerinin yerine getirilerek yapılan sıkıştırma işlemine karşın gömülü yazılım tarafında da aynı işin tersten yapılması ile resmin orijinal hali tekrar elde edilir. Bu işlemin uygulandığı durumlarda en az 6 rengin tekrar şartı aranmaktadır. Bu da 12 bayt yapar. Dolayısıyla her sıkıştırma işleminde minimum 4 bayt kazanç elde edilmiş olur. Tekrar sayısı arttıkça elde edilen kazanç da aynı oranda artacaktır.

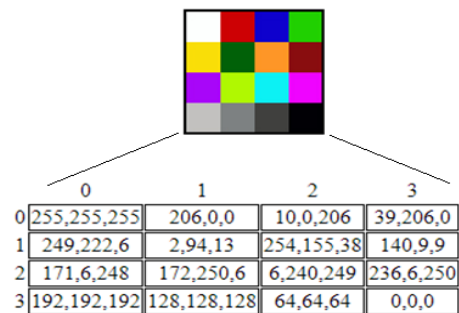
## 2.6. Objeye Boyutlarının Ölçeklendirilmesi

Gerek tasarımsal sebeplerden gerekse de animasyon içerisinde bazen objelerin boyutunu değiştirmek gerekebilir. Bir resmin boyutunu değiştirmek için birçok farklı yöntem kullanılabilir. Bu yöntemlerden bazıları aşağıdaki gibidir:

En yakın-komşu interpolasyonu, çift çizgili ve çift kübik algoritmalar, Sinc ve Lanczos yeniden örnekleme, kutu örnekleme, Mipmap, Fourier-transform metotları, Kenar yönelimli interpolasyon, Hqx, Vektörizasyon, Derin konvolüsyonel sinir ağları vb. birçok metot bulunmaktadır [18]. Bu çalışmada en yakın komşu algoritması tercih edilmiştir.

## 2.7. En Yakın Komşu Metodu

Kaynak resmi hedef boyuta ölçeklendirirken resmin değer kaybetmemesi için pikselleri atlamadan değerlendirmek gerekir. Örnek olarak Şekil 3 'deki 4x4, 16 piksellik görüntüyü bellek üzerinde aşağıdaki tabloda verilen değerler temsil eder.

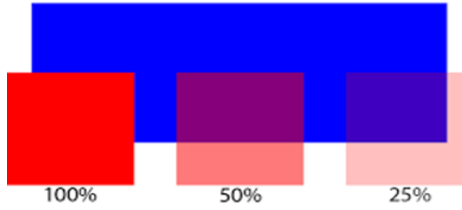


Şekil 3. Resmin piksel verileri ve komşuluk ilişkisi

Şekil 3'deki resmin 10x10 boyutunda bir resim olarak ölçeklendirilmesi istenseydi piksellerden bazılarının birden fazla kullanılması gerekmekeydi. Buradaki temel soru, hedeflenen ölçü kaynağın bir katı olmadığı için hangi piksellerin tekrar edeceğinin neye göre seçileceği sorusudur. Bu sorunun cevabı en yakın komşu algoritmasıdır ve en yakın komşu algoritması doğrusal enterpolasyona dayanır [19,20].

## 2.8. Alfa Harmanlama Metodu İle Opaklığın Ayarlanması

Objeye ile arka plan arasındaki görünürlük ilişkisi opasite, yani saydamlık değeri olarak tanımlanır. Opasite (Saydamlık) ayarlamak için uygulamada kullanılan metotlardan birisi de alfa harmanlama (alpha blending) metotudur. Yarı saydam bir ön plan rengini arka plan rengiyle birleştirme ve böylece ikisi arasında harmanlanmış yeni bir renk üretme işlemidir. Ön plan renginin saydamlığının derecesi tamamen saydamdan tamamen opak konuma kadar (0-255 arasında) ayarlanabilmektedir. Ön plan rengi tamamen saydamsa harmanlanan renk sadece arka plan rengidir. Ters durumda da sadece ön plan rengi görünürde olacaktır. Yarı saydamlık durumlarında renkler ağırlık oranlarına göre harmanlanacaktır. Şekil 4 'de opasite değişiminin arka planın görünürlüğüne olan etkisi görülmektedir.



Şekil 4. Opasite değişimi

Opasite değişimi uygulamasına ürün üzerinden ekran görüntüsü olarak örnek vermek gerekirse Şekil 5'i örnek olarak gösterilebilir. Şekil 5 'e baktığımızda "Yeni Kurulum" yazısı ve ikonuyla, "Eşleştirmeyi Sil" yazısı ve ikonu arasında ton farkı olduğu görülmektedir. Üst satırdaki yazı ve ikonunun saydamlık değeri %0 olduğu için yazı orijinal rengi olan beyaz renk ile siyah arka planı ezerek gözükmemektedir. Ortadaki yazı ve ikon ise %30 kadar saydamlaştırılmış ve orijinalde beyaz olan yazı ve ikon, siyah arka planı %30 oranında geçirdiği için griye çalan bir renk elde edilmiştir. Bu sayede yazı ve ikonlara görsel olarak aktiflik veya pasiflik etkisi kazandırılmıştır.



Şekil 5. Saydamlık değişiminin yapıldığı ekran görüntüsü

Alfa harmanlama (Alpha blending), bilgisayar grafiklerinde saydamlık efektlerine izin verir ve iki rengin dışbükey bir kombinasyonundan oluşur. Renk kodundaki alfa değeri 0.0 (tam saydam) ile 1.0 (tam opak) arasındadır. Ortaya çıkan rengin ifadesi aşağıdaki eşitliklerde gösterilmiştir.

$$out_A = src_A + dst_A (1 - src_A) \quad (1)$$

$$out_{RGB} = \frac{(src_{RGB} src_A + dst_{RGB} dst_A (1 - src_A))}{out_A} \quad (2)$$

**Eşitlik (2)** de  $dst_{RGB}$ , hedeflenen RGB değerini  $dst_A$ , hedeflenen alfa değeri  $src_{RGB}$ , kaynak RGB değerini  $src_A$ , kaynak alfa değerini  $out_{RGB}$ , çıkış RGB değerini  $out_A$ , çıkış alfa değerini göstermektedir. Hedef arka plan opaksa  $dst_A$  değeri 1 olur ve bunu üstteki denkleme girersek  $out_A$  değeri de 1 olur. Bu durumda renk çıktısını veren denklemin son hali aşağıdaki denklemde görülmektedir.

$$out_{RGB} = src_{RGB} src_A + dst_{RGB} (1 - src_A) \quad (3)$$

Alfa bileşeni, 32 bit RGBA 'da olduğu gibi kırmızı, yeşil ve mavi bileşenleri eşit şekilde karıştırmak için kullanılabilir veya alternatif olarak, spektral renk filtrelemesi için birincil renklerin her birine karşılık gelen üç alfa değeri belirtilebilir. Önceden karıştırılmış alfa kullanılırsa, yukarıdaki denklemlerin sadeleşmiş hali aşağıdaki gibi olur.

$$out_A = src_A + dst_A (1 - src_A) \quad (4)$$

$$out_{RGB} = src_{RGB} + dst_{RGB} (1 - src_A) \quad (5)$$

Çıkış rengini veren denklemin basitleştirilmiş hali **Eşitlik (5)** de gösterildiği gibidir [21].

## III. ELDE EDİLEN ÇIKTILAR VE ÖLÇÜMLER

Yapılan çalışmada gerek donanım olarak gerekse de yazılım metodolojisi olarak seri üretime uygun özelliklerde ve düşük maliyetlerde TFT ekranlı ürün tasarımı yapılmış ve geliştirilen GUI kütüphanesi tasarlanan donanımda çalıştırılmıştır. Yapılan tasarım Wi-Fi bağlantılı bir cihaz olduğu için elektromanyetik yayılım testlerine girmiştir. Elektromanyetik yayılım (EMI) sonuçları Tablo 2'de görülmektedir. Tablodan da görüleceği üzere farklı frekanslardaki yayılım değerleri güvenli limit olarak belirlenen 40dBuV/m'lik sınırın altında kalmaktadır [22,23]. Ayrıca gerçekleştirilen gömülü sistem, elektromanyetik uyumluluk ve iklimlendirme gibi çeşitli testleri başarıyla geçerek genel kullanıma uygun bir ürün olduğunu gerek yazılımsal gerekse de donanımsal olarak göstermiştir. Makine üzerinde çalıştırılan GUI kütüphanesi modüler yapısı sayesinde farklı cihazlarda da kullanılabilmiştir. Çalışma süresince geliştirilen sistem ile piyasada benzer amaçla kullanılan yabancı kaynaklı alternatif GUI araçlarına maddi kaynak

ayırmeden proje geliştirmek mümkün hale gelmiştir. Yapılan çalışma ile karşılaştırma yapabilmek için geliştirilen sisteme en yakın alternatif sayılabilecek TFT ekranlı GUI tasarım aracı olarak piyasada yer bulan bir firmanın 2.4" lik ve 4Mb hafızalı ürünü incelenmiş ve bu ürünün yapılan çalışmaya göre gerek yazılım olarak gerekse de donanım olarak çok daha düşük özelliklerde kaldığı görülmüştür [24].

Tablo 3 yapılan çalışmaya benzer özelliklere sahip popüler bir gömülü GUI kartı (sağ) ile bu makalede anlatılan çalışmada geliştirilen sistemin (sol) temel özelliklerinin karşılaştırılması görülmektedir.

Muadil ürün olarak belirlenen 2.4" TFT ekranlı üründe kırmızı ile boyanan özelliklerin projede yapılan çalışmaya göre çok daha düşük özelliklere sahip olduğu görülmektedir. Dolayısıyla gömülü sistemler için geliştirilen GUI kütüphanesi çalışmasının, bu

kütüphaneyi kullanan işletmeye gerek teknik olarak gerek maddi olarak gerekse de zaman ve insan kaynağı tüketimi olarak birçok avantaj sağladığı görülmüştür.

Piyasada sıklıkla tercih edilen ve tasarlanmak istenen gömülü sistem ile benzer özelliklere sahip bir ürünle yapılan karşılaştırma Tablo 3'de gösterilmektedir. Burada görüldüğü üzere muadil piyasa ürünü ile aynı ekran boyutundaki ve ona kıyasla iki misli (8Mb) hafızaya sahip olan bir gömülü sistemin (2020 yılı şartlarında) seri üretim yapılması durumunda 10\$ gibi bir maliyet ile üretilebildiği görülmüştür. Dolayısıyla yapılan bu çalışma ile daha düşük maliyetler ile daha nitelikli ve kaliteli ürünlerin elde edilmesi sağlanmıştır. Ayrıca yazılımsal olarak yapılan resim sıkıştırma özelliği sayesinde bellek tüketimi çok daha verimli kullanılmıştır. Yetenek olarak sağladığı katma değer ise ayrı bir kazanım olarak değerlendirilebilir.

**Tablo 2.** Makinenin yoğun çalışma anındaki elektromanyetik yayılım sonuçları

Frekans (MHz)	Tepe değeri (dBuV/m)	Limit (dBuV/m)	Marjin (dB)	Ölçülen Süre (mS)	Band Genişliği (kHz)	Yükseklik (cm)	Pol.	Azimet açısı (deg)
33.6114	30.06	40.00	9.94	1000	120.000	112.0	V	-92.0
44.9838	29.21	40.00	10.79	1000	120.000	109.0	V	68.0
79.8798	30.71	40.00	9.29	1000	120.000	104.0	V	-96.0
90.2468	30.43	40.00	9.57	1000	120.000	115.0	V	-113.0
113.1735	18.51	40.00	21.49	1000	120.000	193.0	V	-127.0

**Tablo 3.** Geliştirilen gömülü sistemin piyasadaki muadili ile karşılaştırılması

Nitelikler	Tasarlanıp gerçekleştirilen sistem	Piyasada yaygın kullanılan muadil sistem
<b>TFT Boyutu</b>	2.4"	2.4"
<b>Çözünürlük</b>	320*240	320*240
<b>Touch Panel</b>	Kapasitif	Rezistif
<b>Renk derinliği</b>	65K	65K
<b>Harici hafıza</b>	8MB	4MB
<b>RAM</b>	256KB	3.5KB
<b>MCU</b>	120Mhz - CM4	48MHz - CM0
<b>Fiyatı</b>	10\$	16.4\$

#### IV. TARTIŞMA VE DEĞERLENDİRME

Projede oluşturulan jenerik yapı sayesinde iş yükündeki artışa rağmen iş akış süreçlerinde bir gerileme oluşmadan düşük maliyet ile yüksek kalitede bir ürün geliştirme ortamı oluşturulabilmiştir. Böylece piyasada ücretli olarak servis edilen ve ciddi bir pazar hâkimiyeti kuran yabancı menşeli GUI araçlarına ücretsiz bir alternatif sağlanarak yerli sermayenin korunması ve bu konudaki yerli ürün açığının kapatılması sağlanmıştır.

Excel gibi bir ortamda (animasyonlar dâhil) tüm GUI tasarımlarının tablolar halinde oluşturulup script kodları vasıtasıyla "binary" verilere dönüştürülmesi ve harici bir hafızaya yüklenmesi sayesinde tasarımın herhangi bir gömülü sistemde kullanılabilmesi sağlanmıştır. Gömülü sistem için taşınabilir ve modüler yapıda geliştirilen yazılım seri portundan aldığı komutlara göre harici hafıza birimine sayfalar halinde

yüklenmiş ekranları çıkarabilmekte ve tanımlı animasyonları çalıştırabilmektedir. Üstelik oluşturulan mimari çoklu dil desteğini destekleyecek şekilde geliştirilmiştir. Tüm bunlar yapılan çalışmanın kolayca herhangi bir gömülü sisteme uyarlanabilir jenerik bir yapıda olmasını sağlamıştır.

Böylece piyasadaki birçok profesyonel GUI araçları ile rekabet edebilecek özelliklere sahip yerli bir alternatif ürün çalışması yapılmıştır. Fakat birçok üründe olduğu gibi bu çalışmada da eksik noktalar kalmıştır. İlerleyen süreçlerde bu eksikliklerinde zamanla giderilmesi planlanmaktadır. Bunlardan bazıları şu şekilde sayılabilir:

- Yönetimsel kolaylık sağlaması açısından projede UTF-16 tercih edilmesine rağmen hafıza maliyeti sağladığı için ileride iyileştirilmesi gereken kısımlardan birisi

olarak kodlama formatının UTF-8'e dönüştürülmesi şüphesiz daha faydalı olacaktır.

- İnternet bağlantısı doğrudan TFT kartı üzerine alınarak arada köprü olarak kullanılan display kartının yarattığı haberleşme gecikmelerinin ve oluşan ekstra üretim maliyetlerinin ortadan kaldırılması mümkün olabilir.
- PC tarafında Excel kullanımı yerine daha özgün bir GUI arayüz yazılımının oluşturulması tasarımları oluştururken daha büyük kolaylıklar sağlayabilir.

## V. SONUÇ

Profesyonel GUI araçlarıyla aynı kabiliyete sahip hızlı ve kolay kullanıma sahip genel bir GUI kitaplığı geliştirilmiş ve mikrodenetleyici tabanlı gömülü sistem donanımı üzerinde TFT ekranlı bir proje gerçekleştirilmiştir.

Bu çalışmada hem donanım hem de yazılım metodolojisi olarak seri üretime uygun ve düşük maliyetli TFT ekranlı ürün tasarımları tasarlanmış ve geliştirilen GUI kütüphanesi mikrodenetleyici tabanlı bir donanım üzerinde çalıştırılmıştır. Geliştirilen gömülü sistem EMI, EMC, iklimlendirme gibi testleri başarıyla geçmiştir. Yapılan çalışmayla karşılaştırma yapabilmek için geliştirilen sisteme en yakın alternatif olarak değerlendirilebilecek, piyasada sıklıkla kullanılan bir TFT ekranlı GUI aracı incelenmiştir.

Projenin en kapsamlı ve üzerinde en çok düşünülmesi gereken kısmı modüler ve jenerik bir yapı oluşturulması gerekliliğidir. Kullanıcı, bilgisayar programı ile TFT ekranda oluşturmak istediği bütün görselleri, animasyonları, hareketli ekranları ve bunların birbiri ile olan bağlantı senaryolarını tasarlayabilmeli ve bu çalışma sonucunda üretilen binary data dosyalarını seri port üzerinden harici hafızaya gönderebilmelidir. Bu sayede jenerik ve tekrar tekrar kullanılabilen (Reusable) bir yapı geliştirilmiş olur. Mikrodenetleyici tarafı harici hafızadaki görsel tasarımı, oluşturulan senaryolar kapsamında çalıştırarak son kullanıcının MCU yazılımı ile uğraşmadan (ya da istisnai durumlara göre çok az uğraşarak) pratik bir şekilde TFT ekran üzerinde proje geliştirebilmesi sağlanacaktır. Bu anlamda gerek yazılımsal gerekse donanımsal olarak kullanılan tüm materyaller 6. Bölümden itibaren detaylı olarak anlatılmaktadır.

## KAYNAKÇA

- [1] <https://www.sensomatic.com/chz/gui/history.html> (17.07.2020).
- [2] <https://uxdesign.cc/the-worlds-gone-flat-evolutions-in-interface-design-cb7ddd295f54> (17.07.2020).
- [3] Hu, X., Jiang, C., Zhang, W., Zhang, J., Yu, R., & Lv, C. (2010, December). An Event Based GUI Programming Toolkit for Embedded System. In *2010 IEEE Asia-Pacific Services Computing Conference* (pp. 625-631). IEEE.
- [4] Chen, Z. J., Zhang, Z. X., & Zhang, J. J. (2008, July). Design and implementation of video player system based on embedded system and Qt/E. In *2008 5th International Conference on Visual Information Engineering (VIE 2008)* (pp. 468-472). IET.
- [5] Haque, M. E., Islam, M. R., Rabbi, M. T. F., & Rafiq, J. I. (2019, December). IoT Based Home Automation System with Customizable GUI and Low Cost Embedded System. In *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)* (pp. 1-5). IEEE.
- [6] Tan, D. P., Chen, S. T., Bao, G. J., & Zhang, L. B. (2018). An embedded lightweight GUI component library and ergonomics optimization method for industry process monitoring. *Frontiers of Information Technology & Electronic Engineering*, 19(5), 604-625.
- [7] R. Filman, "Editor's Introduction: From Isolation to Universal Connectivity" in *IEEE Internet Computing*, vol. 2, no. 03, pp. 40-41, 2001. doi: 10.1109/MIC.2001.935175
- [8] Huo, P., Yang, F., Luo, H., Zhou, M., & Zhang, Y. (2019). Distributed monitoring system for precision management of household biogas appliances. *Computers and Electronics in Agriculture*, 157, 359-370.
- [9] Ayan, O., & Turkay, B. (2018, October). Improvement of Home Energy Awareness through the Designed Graphical User Interface. In *2018 6th International Conference on Control Engineering & Information Technology (CEIT)* (pp. 1-6). IEEE.
- [10] Ramler, R., Buchgeher, G., & Klammer, C. (2018). Adapting automated test generation to GUI testing of industry applications. *Information and Software Technology*, 93, 248-263.
- [11] <https://www.britannica.com/technology/graphical-user-interface> (17.07.2020).
- [12] Santra, S., Bhowmick, S., Paul, A., Chatterjee, P., & Deyasi, A. (2018, May). Development of GUI for text-to-speech recognition using natural language processing. In *2018 2nd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)* (pp. 1-4). IEEE.
- [13] Li, J., W. Tigwell, G., & Shinohara, K. (2021, May). Accessibility of High-Fidelity Prototyping Tools. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (pp. 1-17).
- [14] <https://developer.arm.com/solutions/machine-learning-on-arm/developer-material/how-to-guides/build-arm-cortex-m-voice-assistant-with-google-tensorflow-lite/single-page> (17.07.2020).
- [15] <https://www.toptal.com/designers/ui/vr-ar-design-guide> (17.07.2020).



- 
- [16] <https://medium.com/hackernoon/how-facebook-apple-microsoft-google-and-amazon-are-investing-in-ai-f58b5706e34a> (17.07.2020).
- [17] <https://www.computerhope.com/jargon/g/gui.htm> (17.07.2020).
- [18] Reddy, K. S., & Reddy, D. K. R. L. (2013). Enlargement of image based upon Interpolation Techniques. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(12), 4631.
- [19] <http://courses.cs.vt.edu/~masc1044/L17-Rotation/ScalingNN.html> (17.07.2020).
- [20] Rukundo, O., & Cao, H. (2012). Nearest neighbor value interpolation. *arXiv preprint arXiv:1211.1768*.
- [21] [https://en.wikipedia.org/wiki/Alpha\\_compositing#Alpha\\_blending](https://en.wikipedia.org/wiki/Alpha_compositing#Alpha_blending) (17.07.2020).
- [22] CISPR, CISPR 32:2015, second edition, "Electromagnetic compatibility of multimedia equipment – Emission requirements."
- [23] Code of Federal Regulations (CFR), CFR Title 47, FCC Part 15, Subpart B – Unintentional Radiators, Section 15.109, Radiated emission limits.
- [24] <https://nextion.tech/basic-series-introduction/> (17.07.2020).