

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
SANAT VE TASARIM ANA SANAT DALI
MÜZİK VE SAHNE SANATLARI
TEZLİ YÜKSEK LİSANS PROGRAMI**

YÜKSEK LİSANS TEZİ

**GÖRME ENGELLİLER İÇİN MÜZİKSEL OKUMAYA
YÖNELİK BRAİLLE TABANLI DONANIMSAL
ARAYÜZ TASARIMI**

**BURAK AYAZ
17740002**

**TEZ DANIŞMANI
Doç. Dr. ARDA EDEN**

**İSTANBUL
2020**

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
SANAT VE TASARIM ANA SANAT DALI
MÜZİK VE SAHNE SANATLARI
TEZLİ YÜKSEK LİSANS PROGRAMI**

YÜKSEK LİSANS TEZİ

**GÖRME ENGELLİLER İÇİN MÜZİKSEL
OKUMAYA YÖNELİK BRAİLLE TABANLI
DONANIMSAL ARAYÜZ TASARIMI**

**BURAK AYAZ
17740002
ORCID NO: 0000-0001-9763-1439**

**TEZ DANIŞMANI
Doç. Dr. ARDA EDEN**

**İSTANBUL
2020**

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
SANAT VE TASARIM ANA SANAT DALI
MÜZİK VE SAHNE SANATLARI
TEZLİ YÜKSEK LİSANS PROGRAMI**

YÜKSEK LİSANS TEZİ

**GÖRME ENGELLİLER İÇİN MÜZİKSEL
OKUMAYA YÖNELİK BRAİLLE TABANLI
DONANIMSAL ARAYÜZ TASARIMI**

**BURAK AYAZ
17740002**

Tezin Enstitüye Verildiği Tarih: 05/10/2020

Tezin Savunulduğu Tarih: 28/09/2020

Tez Oy Birliği ile Başarılı Bulunmuştur

	Unvan Ad Soyad	İmza
Tez Danışmanı	: Doç. Dr. Arda Eden	
Jüri Üyeleri	: Doç. Dr. Koray Sazlı	
	Prof. Dr. Can Karadoğan	

**İSTANBUL
EYLÜL 2020**

ÖZ

GÖRME ENGELLİLER İÇİN MÜZİKSEL OKUMAYA YÖNELİK BRAİLLE TABANLI DONANIMSAL ARAYÜZ TASARIMI

Burak Ayaz

Eylül, 2020

Bu tez araştırması bünyesinde, görme engelli bireyler için, müziksel okumaya yönelik, müzik braille alfabesi ile uyumlu ve titreşim uyaranlı arayüze sahip “kendin yap” felsefesini temel alan düşük maliyetli bir araç tasarlanmıştır. Araç, mikrodenetleyici tarafından kontrol edilen ve braille alfabesindeki noktalara karşılık gelen titreşim motorlarından oluşan bir yapıya sahiptir. Her bir motor, ilgili parmağı titreşim ile uyararak, o parmağa karşılık gelen braille noktasının algılanmasını ve bu sayede müziksel dilin kullanıcıya aktarılmasını sağlamaktadır. Aracın tasarımında, mevcut ticari braille ürünlerinin tuş dizilimleri temel alınmış, böylelikle önceden edinilmiş kullanıcı alışkanlıklarının avantaja dönüştürülmesi hedeflenmiştir. Tasarlanan aracın, tonalite, ölçü rakamları, nota değer ve süreleri, sus işaretleri, oktav numaraları, ses değiştirici işaretler, ölçü işaretleri ve noktalar gibi temel müzik braille hücrelerini titreşim uyaranlı olarak yansıtmayı sağlanmıştır. Tasarlanan aracın optimum parametrelerinin ve başarımının ölçülmesi amacıyla iki deney grubu oluşturulmuş, birinci deneyde çeşitli motor titreşim süreleri ile hazırlanan sorular görebilen 40 kişi ile test edilmiş, çeşitli yaş, cinsiyet, ellilik ve hücreleri meydana getiren noktaların bitişik-ayrık olma durumuna göre algılama başarımı incelenmiştir. İkinci deneyde görme engelli, müzik braille bilen 8 kişi ile müzik braille testleri yapılmış ve vBraille’nin kabartma yazı ile okuma hızları karşılaştırılmıştır. Elde edilen sonuçlarda, güncel olarak Perkins Braille kullanan kişilerin kullanmayanlara göre vBraille’e daha çabuk adapte olduğu ve hızlı okuma yapabildiği görülmektedir.

Anahtar Kelimeler: Müzik Braille, Görme Engelli, Müzik, Arayüz Tasarımı, Titreşim.

ABSTRACT

A BRAILLE BASED MUSICAL READING INTERFACE FOR VISUALLY IMPAIRED

**Burak Ayaz
September, 2020**

Within this thesis research, a low-cost tool is designed for visually impaired individuals, based on the "do it yourself" philosophy for musical reading. Design is compatible with the music braille alphabet and having a vibration-stimulating interface. Design has a structure consisting of vibration motors that are controlled by the microcontroller which correspond to the dots in the braille alphabet. By stimulating the related finger with vibration, each motor ensures the perception of the braille point corresponding to that finger and thus transferring the musical language to the user. The design of the appliance is based on the key sequences of existing commercial braille products, thereby aiming to turn the previously acquired user habits into an advantage. The designed tool is provided to reflect the basic music braille cells with vibration stimuli such as key signature, time signature, note values and durations, rest signs, octave numbers, accidentals, barlines and dots. Two experimental groups were formed to measure the designed optimum parameters and performance. In the first experiment, with various motor vibration durations was tested with 40 sighted people, and its performance was examined according to various age, gender and the feature of being adjacent-discrete in continuing points. In the second experiment, music braille tests were done with 8 people who are visually impaired, knew music braille and the speed of reading of vBraille was compared with relief writing. In the results, it is seen that people who currently use Perkins Braille, adapt to vBraille and can read faster than those who do not.

Keywords: Music Braille, Visually Impaired, Music, Interface Design, Vibration.

ÖN SÖZ

Bu çalışmanın her aşamasında emeği olan; önerilerini, bilgisini ve kaynaklarını paylaşan, destek ve yardımını esirgemeyen değerli tez danışmanım ve hocam Sn. Doç. Dr. Arda Eden'e,

Çalışmanın başından itibaren olumlu yaklaşımları, çalışmanın şekillenmesi adına çeşitli yardım ve kaynaklar sunan ve tez jürisi olarak katılan hocam Sn. Doç. Dr. Koray Sazlı'ya,

Değerli anne ve babam Tülin ve Zafer Ayaz'a,
Teşekkür ederim..

İstanbul; Eylül, 2020

Burak Ayaz

İÇİNDEKİLER

ÖZ	iii
ABSTRACT	iv
ÖN SÖZ	v
İÇİNDEKİLER	vi
TABLOLAR LİSTESİ	viii
ŞEKİLLER LİSTESİ	ix
KISALTMALAR	xi
1. GİRİŞ	1
1.1. Braille Okuma Amaçlı Geliştirilmiş Araçlar	4
1.2. Araştırmanın Amacı.....	8
1.3. Araştırmanın Önemi	8
1.4. Problem Cümlesi	9
1.5. Alt Problemler	9
1.6. Sayıtlar.....	9
1.7. Sınırlılıklar.....	10
1.8. İlgili Araştırmalar	10
2. MÜZİK BRAILLE	14
2.1. Tonalite	14
2.2. Ölçü Rakamları	15
2.3. Oktav işaretleri.....	16
2.4. Ses Değiştirici İşaretler	17
2.5. Notalar ve Nota değerleri.....	18
2.6. Sus İşaretleri	18
2.7. Uzatma Bağı	19
2.8. Ölçü Çizgisi	19
3. MUSICXML	21
4. VBRAILLE	28
4.1. VBraille Tasarımı	28
4.2. Cihaz Maliyeti	32
4.3. Donanım	33
4.4. VBraille Dosya Biçimi	35
4.5. MusicXML-VBraille Kodu Dönüştürücü Yazılımı.....	38
5. YÖNTEM	40
5.1. Birinci Deney.....	40
5.2. İkinci Deney.....	41

6. BULGULAR VE YORUMLAR	42
6.1. Birinci Deney İncelemesi	42
6.1.1. Motor Sürelerindeki Doğru Cevaplarının Yaşa Göre İncelenmesi.....	43
6.1.2. Motor Sürelerindeki Doğru Cevaplarının Cinsiyete Göre İncelenmesi ..	52
6.1.3. Motor Sürelerindeki Doğru Cevaplarının Sağ ve Sol Ele Göre İncelenmesi	53
6.1.4. Testlerde Hücre Noktalarındaki Bitişik-Ayrık Durumunun İncelenmesi	53
6.2. İkinci Deney İncelemesi	56
7. SONUÇ	59
KAYNAKÇA	62
EKLER	64
ÖZ GEÇMİŞ	105

TABLÖLAR LİSTESİ

Tablo 1:	Braille Ürünleri Fiyat Listesi.....	7
Tablo 2:	Cihaz Maliyet Tablosu	32
Tablo 3:	VBraille Üzerinde Notaların Kodlanması	36
Tablo 4:	VBraille Üzerinde Nota Değerlerinin Kodlanması	36
Tablo 5:	VBraille Üzerinde Ses Değiştirici İşaretlerin Kodlanması.....	37
Tablo 6:	VBraille Üzerinde Nokta İşaretinin Kodlanması	37
Tablo 7:	Test Sonuçlarının Genel Raporu	42
Tablo 8:	Doğru Cevaplarının Cinsiyetlere Göre İstatistik Tablosu	52
Tablo 9:	Doğru Cevaplarının Sağ ve Sol Ele Göre İstatistik Tablosu	53
Tablo 10:	Noktaların Bitişik-Ayrık Durumunu İçeren Deney Soruları	54
Tablo 11:	Noktaların Bitişik-Ayrık Durumunu İçeren Deney Ortalamaları.....	55
Tablo 12:	İkinci Deney Grubu VBraille ve Kabartma Yazı Sonuç Tablosu	56

ŞEKİLLER LİSTESİ

Şekil 1:	Slate and Stylus.....	1
Şekil 2:	Solda Barbier Heceleme Tablosu ve Sağda 12 Hücreli Yazımı	2
Şekil 3:	Louis Braille 6 Hücreli Braille Alfabesi ve “a, b, c, 1, 2, 3” Yazımı	3
Şekil 4:	Perkins Brailler.....	4
Şekil 5:	Mountbatten Whisperer Writer	5
Şekil 6:	Hims Smart Beetle	5
Şekil 7:	Brailliant BI 14 Braille Display	6
Şekil 8:	Orbit Reader 20 Braille Notebook	6
Şekil 9:	Braille Okuma Vibrotaktıl Prototipi UbiBraille	11
Şekil 10:	Holibraille, Dokunmatik Ekranlı Mobil Cihazlar İçin Çok Noktalı Bir Vibrotaktıl Çıktı Sistemi.....	12
Şekil 11:	Holibraille, Farklı Materyaller Üzerinde Titreşim Ölçüm Grafiği	12
Şekil 12:	Vibraille Görseli.....	13
Şekil 13:	Müzik Notası ve Tek Satırlı Müzik Braille Yazımı.....	14
Şekil 14:	Tonalite	15
Şekil 15:	Ölçü Rakamları	16
Şekil 16:	Oktav İşaretleri.....	17
Şekil 17:	Ses Değiştirici İşaretlerin Braille Yazımı	18
Şekil 18:	Nota İsimleri, Değerleri ve Nokta.....	18
Şekil 19:	Sus İşaretleri.....	19
Şekil 20:	Uzatma Bağı.....	19
Şekil 21:	Boşluk İle Belirtilen Ölçü Çizgisi.....	20
Şekil 22:	1. ve 4. Noktalarla Temsil Edilen Noktalı Ölçü Çizgisi	20
Şekil 23:	Sol Anahtarında Dört Dörtlük “Do” Notasının Musicxml Üzerinde Gösterimi	23
Şekil 24:	MusicXML Döküman Özellikleri.....	24
Şekil 25:	MusicXML Müzikal Özellikler	25
Şekil 26:	MusicXML Üzerinde Ölçü Rakamı.....	25
Şekil 27:	MusicXML Üzerinde Anahtar	26
Şekil 28:	MusicXML Üzerinde Perde	26

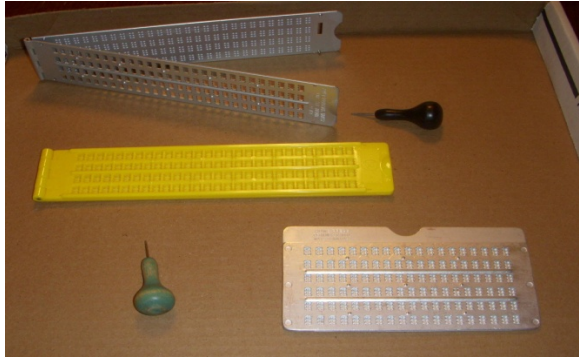
Şekil 29:	MusicXML Üzerinde Sus İşareti	27
Şekil 30:	Bir Braille Hücresi (Solda) ve VBraille Üzerinde Hücre Temsili (Sağda)	28
Şekil 31:	VBraille İlk Prototipi	29
Şekil 32:	RTV 2 Silikon Kalıp ve İçerisinde Poliüretan Reçine	29
Şekil 33:	İlk VBraille'in İç Tasarımı	30
Şekil 34:	İlk VBraille Filtre Sünger Yerleşimi.....	30
Şekil 35:	İlk VBraille İzolasyon Süngeri Yerleşimi.....	31
Şekil 36:	VBraille Motor Yerleşimi	31
Şekil 37:	VBraille 1. Test Cihazı	32
Şekil 38:	VBraille Donanım Şeması	34
Şekil 39:	VBraille Devresinin LED Aydınlatmalar İle Breadboard Üzerindeki Görünümü	34
Şekil 40:	VBraille Çalışma Adımlarını Gösteren Şema.....	35
Şekil 42:	Örnek Bir Egzinin Vbraille Transkodunda Gösterimi	38
Şekil 43:	MusicXML-VBraille Kodu Dönüşüm Algoritması	39
Şekil 43:	1000ms'deki Doğru Cevapların Saçılım Grafiği	44
Şekil 44:	500ms'deki Doğru Cevapların Saçılım Grafiği	45
Şekil 45:	400ms'deki Doğru Cevapların Saçılım Grafiği	46
Şekil 46:	350ms'deki Doğru Cevapların Saçılım Grafiği	47
Şekil 47:	300ms'deki Doğru Cevapların Saçılım Grafiği	48
Şekil 48:	250ms'deki Doğru Cevapların Saçılım Grafiği	49
Şekil 49:	200ms'deki Doğru Cevapların Saçılım Grafiği	50
Şekil 50:	150ms'deki Doğru Cevapların Saçılım Grafiği	51
Şekil 51:	Tüm Motor Sürelerinde Yaş-Doğru Oranı Grafiği	52

KISALTMALAR

ms	: Milisaniye
sn	: Saniye
mm	: Milimetre
Hz	: Hertz
GHZ	: Giga hertz
GB	: Gigabyte
gr	: Gram
L	: Sol
R	: Sağ
3D	: 3 Boyutlu
SPSS	: Statistical Package for the Social Sciences
XML	: Extensible Markup Language
DTD	: Document Type Definition
SGML	: Standard Generalized Markup Language
HTML	: HyperText Markup Language
USB	: Universal Serial Bus
LCD	: Liquid Crystal Display
LED	: Light Emitting Diode
TSD	: Touch Sense Display
MIDI	: Musical Instrument Digital Interface
GPS	: Global Positioning System
DAW	: Digital Audio Workstation
SD	: Secure Digital
SPI	: Serial Peripheral Interface
GND	: Ground
www	: World Wide Web
URL	: Uniform Resource Loader
ABD	: Amerika Birleşik Devletleri
₺	: Türk Lirası
€	: Euro
\$	: Dollar

1. GİRİŞ

Braille alfabesinin temelleri, Fransız ordusunda bir kaptan olan Charles Barbier'in talihsiz bir olay üzerine icat ettiđi ve "gece yazıları" olarak adlandırdıđı bir sisteme dayanmaktadır (AFB, [15.06.2020]). Savaş sırasında bir askerin yazılı bir mesajı okumak için fener yakması, askerlerin düşman tarafından fark edilmelerine ve öldürölmelerine sebep olmuştur. Barbier, Orada yaşadığı bu trajik olay üzerine askerlerin karanlıkta haberleşebilecekleri bir okuma-yazma sistemi icat etmiştir (Kent, [10.05.2020]). Barbier, yazı tahtaları (slate) ve noktaları oluşturacak kalemler (stylus) kullanarak kabartma yazı yazılmasını mümkün hale getirmiştir (Paths to Literacy, [15.16.2020]).



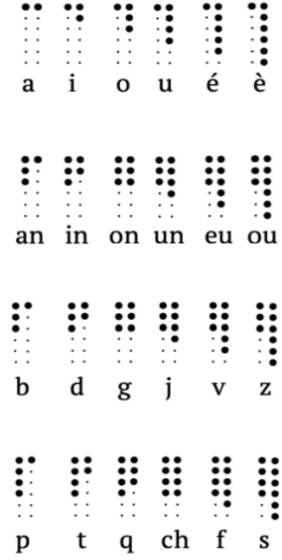
Şekil 1: Slate and Stylus

Wikipedia. "Slate and Stylus". https://en.wikipedia.org/wiki/Slate_and_stylus [6.5.2020]

O zamanlarda görme engelliler için kitaplar kabartma harflerle yazılmaktaydı. Öğrenciler harfleri dokunarak okumayı öğrenebilseler de bu süreç yavaş ve zor ilerliyordu (Bertman, [6.05.2020]).

Kaptan Barbier'in yazı sistemi kabartılmış baskıdan çok farklı, alfabenin hecelerini temsil eden hücre yapıları barındırmaktadır. Fransız dili sıkça kullanılan birçok harf kombinasyonu içerdiğinden, Barbier hücre temel yapısını altı yatay sıra ve altı dikey sütunla, toplam 36 farklı dikdörtgen hücre ile tasarlamıştır (Bertman, [6.05.2020]).

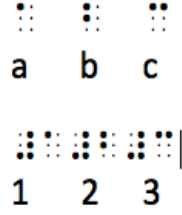
	1	2	3	4	5	6
1	a	i	o	u	è	é
2	an	in	on	un	eu	ou
3	b	d	g	j	v	z
4	p	t	q	ch	f	s
5	l	m	n	r	gn	ll
6	oi	oin	ian	ien	ion	ieui



Şekil 2: Solda Barbier Heceleme Tablosu ve Sağda 12 Hücreli Yazımı

1821 yılında Charles Barbier hem kendi alfabesinin eğitimini vermek hem de görme engelli gençleri ziyaret etmek amacıyla Paris Kraliyet Enstitüsü'ne gitmiştir. Barbier'in dersine katılan 12 yaşında görme engelli bir enstitü öğrencisi olan Louis Braille, gece yazılarının, karmaşık yapısının okumayı yavaşlattığını, bununla beraber noktalama, büyük harf ve rakamlar gibi eksiklerin olduğunu fark etmiş ve sistemi geliştirmek üzere çalışmalarına başlamıştır. Braille üç yıl sonrasında çalışmalarını tamamlamış ve 6 noktadan oluşan alfabesini, Barbier sistemindeki eksikleri tamamlayarak hazır hale getirmiştir. Braille'in bu alfabesi enstitüdeki öğrenciler tarafından yazılı kopyalar halinde çoğaltılmıştır (Kent, [10.05.2020]).

Braille karakterleri dikey bir dikdörtgen düzen üzerine dizilmiş, küçük kabartılmış noktalardan oluşmaktadır. Bu dikdörtgen içerisindeki noktalar ile altmış dört farklı kombinasyon oluşturulabilmektedir. Braille yazımında alfabe harfleri latin alfabesi olarak kullanılmakta ve her harf alfabetik sırayla uyumlu yazılmaktadır (AFB, [15.06.2020]).



Şekil 3: Louis Braille 6 Hücreli Braille Alfabeti ve “a, b, c, 1, 2, 3” Yazımı

İlk Türk Braille alfabesi 1920 yılında İzmir’de kabul edilmiş ve kullanılmaya başlanmıştır. Bu alfabe, eski Arap alfabesine karşılık kabartma sembollerin tespiti için meydana getirilmiştir. (MEB, 1975, 7). Braille yazımının Türkiye’de eğitimi de 1925 yılında açılan Görme Engelliler okulu ile başlamıştır. (MEB, 1991, X). Braille’ın Türkçe karakterlere uyarlanması ve günümüze kadar ulaşması 1951 yılında birçok ülkenin de katıldığı Unesco önderliğinde Paris’te “Kabartma Yazı Konferansı” sayesinde mümkün olmuştur (MEB, 1975, 7). Müzik Braille ilk olarak 1871 yılında Londra’da Dr. Armitage’in yaptığı çalışmalar sonunda yazılı bir kaynak haline getirilmiştir (MEB, 1991, IX). 1885 yılında Braille müzik uzmanlarından oluşan İngiliz, Alman ve Fransız bir topluluk Almanya’nın Köln kentinde bir araya gelerek birtakım işaret ve uygulama kurallarının bütünlüğünü sağlamak için bir komite kurmuş ve bu komite “Cologne Key” (Köln Anahtarı) isimli Braille müzik yazımını meydana getirmiştir (The Braille Authority of North America, 2016, XIX).

Müzik işaretlerinde birliğin sağlanmasını isteyen Amerikan Basımevi’nin dışişleri temsilcisi L. Reverat, 1927 yılında Paris’te bir Braille Müzik kongresi toplanması adına girişimde bulunmuştur. Bu kongre sonucunda 20 Nisan 1929 yılında Paris’te Fransa, İtalya, İngiltere, Almanya ve ABD’nin yanı sıra dokuz farklı Avrupa ve Güney Amerika ülkeleri temsilcileri verilen kararları aynen kabul ettiklerini bildirmişlerdir (MEB, 1991, IX). 22-29 Temmuz 1954 tarihleri arasında Paris’de gerçekleştirilen Braille Müzik İşaretleri Sistemi Kongresi Louis Rodenberg tarafından yürütülmüş ve 19 ülke katılım sağlamıştır. Bu kongrede çok az sayıdaki delege müzik yazımında bölüm bölüm yazım kuralını talep etmiştir. Ancak temsilcilerin çoğunluğu mürekkep baskıda dizek şeklinde yazılmış olan müziğin Braille yazıda ölçü ölçü yazılması kuralını desteklemiş ve bu kural tüm kongre tarafından kabul görmüştür. Bu kongrede önemli olan bir diğer nokta da, mürekkep baskıda yazılan tüm müzikal ifadelerin Müzik Braille’e çevrilmesi konusudur. Nota

işaretleri, uzatma bağı, portrenin altında veya üstünde bulunan köşeli parantezler ve çeşitli anahtar işaretleri ve dahası bu kongre süresince işlenmiş ve kural olarak belirlenmiştir. H.V. Spanner isimli temsilci bu kongrenin rapor taslağını hazırlamak ve yeni müzik işaretleri klavuzunu yazmakla görevlendirilmiştir. (MEB, 1991, X)

1956 yılında Braille Müzik Notasyonu Uluslararası El Kitabı, H. V. Spanner tarafından hazırlanmış ve Birleşmiş Milletler Eğitim, Bilim ve Kültür Örgütü tarafından desteklenen üçüncü Uluslararası Konferansın ürünü olarak ortaya çıkmıştır (The Braille Authority of North America, 2016, XIX).

1960'lı yılların başından itibaren The Braille Authority of North America (Amerikan Braille Otoritesi) müzikal ifadelerin Braille karşılığının genişletilmesi ve ihtiyaç duyulduğunda yeni işaret ve formatlar önermek için çalışmalara başlamıştır. (The Braille Authority of North America, 2016, XX).

1.1. Braille Okuma Amaçlı Geliştirilmiş Araçlar

Braille tarihinde bilinen en eski altı noktalı Braille yazımını kolaylaştıran araçtır. Parmak dizilimi ilgili noktalara karşılık gelecek şekilde yazım sağlar (Kent, [10.05.2020]). İlk Braille Daktilo 1892'de Frank Haven Hall tarafından üretilmiştir. Bu daktilo, Perkins Körler Okulu'nda marangozluk öğretmeni olan David Abraham 1951 tarafından geliştirilmiş ve Perkins Brailler adını almıştır (McGinnity, [14.03.2020]).



Şekil 4: Perkins Brailler

Wikipedia. “Perkins Braille”. https://en.wikipedia.org/wiki/Perkins_Brailler [6.05.2020]

Günümüze kadar gelen süreç içerisinde Braille okuma olanağı sunan birçok teknolojik araç geliştirilmiştir. Perkins Brailler tuş dizilimine sahip olan bir diğer

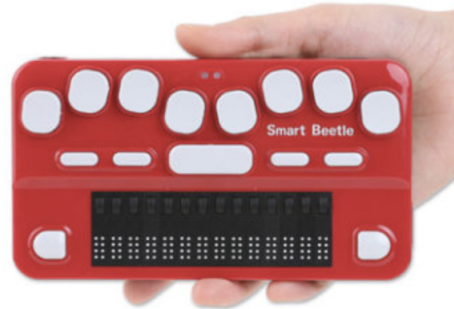
araç “Mountbatten Whisperer Writer” dır. Mountbatten Whisperer Writer, Braille eğitimi için kullanılabilecek bilgisayar bağlantılı çalışan bir araçtır (Humanware, [20.03.2020]).



Şekil 5: Mountbatten Whisperer Writer

Humanware. “Mountbatten Whisperer Writer”,. <https://store.humanware.com/hus/mountbatten-whisperer-writer.html> [6.05.2020]

Hims Inc. firması 1999 yılından beri Braille ve erişilebilir sağlık cihazları sağlayan küresel bir üretici ve distribütördür. ABD merkezi Teksas, Austin'de bulunmakta ve tüm ürünlerinin satış, destek ve onarımı Kuzey Amerika'ya hizmet vermektedir. (Hims, [20.03.2020]). Braille okuma imkanı sunabilen “smart screen” teknolojisini barındıran en uygun fiyatlı ürünü “Smart Beetle” 14 hücreli Braille okuma aracıdır. Usb ve Bluetooth üzerinden bağlantı sağlanabilmektedir (Hims, [20.03.2020]).



Şekil 6: Hims Smart Beetle

Hims. “Smart Beetle”, <https://www.hims-inc.com/product/smart-beetle/> [20.03.2020]

Aynı özellikleri barındıran farklı markaların ürünleri de bulunmaktadır. Humanware firmasının “Brailiant BI 14 braille display” ürünü, Smart Beetle ile aynı özellikleri barındırmaktadır. Farklı olan özellikleri ise tamamı ile dış tasarımları ve tuş yerleşimleridir (Hims, [20.03.2020]).



Şekil 7: Brailiant BI 14 Braille Display

Humanware. “Brilliant BI 14 Braille Display”. <https://store.humanware.com/hus/brailiant-bi14-braille-display.html> [6.05.2020]

En uygun fiyatlı ürün de Orbit firmasına ait “Orbit Reader 20”dir. Usb üzerinden bağlantı sağlayan ve 20 hücreli Braille okuma olanağı sunan bir araç olarak karşımıza çıkmaktadır (Flusoft, [20.03.2020]).



Şekil 8: Orbit Reader 20 Braille Notebook

Flusoft. “Orbit Reader 20 - Braille notebook”. <http://www.flusoft.de/produkte/orbit/index.html> [16.07.2020]

Bu bölümde adı geçen araçların güncel fiyatları ve bazı özellikleri Tablo 1’de belirtilmiştir.

Tablo 1: Braille Ürünleri Fiyat Listesi

Firma Adı	Ürün İsmi	Özellikler	Fiyat ¹
Flusoft	Orbit Reader 20 - Braille notebook	USB üzerinden bağlantı. 20 hücreli Braille okuma ve yazma aracı.	799.97 €
Perkins	Perkins Brailier	Kabartma yazı daktilosu.	\$810.00
HIMS	Smart Beetle	14 hücreli Braille okuma ve yazma aracı. USB ve Bluetooth üzerinden bağlantı.	\$995.00
Humanware	Brailliant BI 14 braille display	14 hücreli Braille okuma ve yazma aracı. USB ve Bluetooth üzerinden bağlantı.	\$995.00
Freedom Scientific	ELBraille 40 V	QWERTY klavye ve 40 hücreli Braille okuma ve yazma aracı. Bluetooth bağlantı özelliği. 1 GHZ işlemci. 128 GB hafıza. Dahili yazılım sayesinde günlük program yaratma.	\$2,695.00
HIMS	Actilino	16 hücreli Braille okuma ve yazma aracı. Bluetooth bağlantı özelliği.	\$2,795.00
Humanware	Mountbatten Whisperer Writer	Sesli okuma, daktilo yazımı ve Bluetooth ile cihazlara bağlantı.	\$2,850.00
HIMS	BrailleSense U2 MINI	14 hücreli Braille okuma ve yazma aracı. Dahili Windows işletim sistemi. 1 GHZ işlemci. 32 GB hafıza. Bluetooth bağlantı özelliği ve dahası.	\$2,995.00

¹ Tabloda adı geçen ürünlerin fiyatları, firmaların web siteleri üzerinden 8 Mayıs 2020 tarihinde alınmıştır.

HIMS	QBraille XL	QWERTY klavye ve 40 hücreli Braille okuma ve yazma aracı. Bluetooth bağlantı özelliği.	\$3,195.00
HIMS	BrailleSense U2	QWERTY klavye ve 40 hücreli Braille okuma ve yazma aracı. Bluetooth bağlantı. 1 GHZ işlemci. 32 GB hafıza. LCD ekran.	\$4,595.00

1.2. Araştırmanın Amacı

Bu araştırmanın amacı, görme engelli bireylerin müzik okuyabilmelerini sağlayacak, müzik braille alfabesini temel alan, titreşim uyaranlı bir arayüze sahip, düşük maliyetli bir araç tasarlamak ve tasarlanan aracın kullanılabilirliğine yönelik performans değerlerini ölçerek, aracın optimizasyonu için gerekli parametreleri tespit etmektir.

1.3. Araştırmanın Önemi

Bu araştırma, tasarlanan aracın:

- Braille alfabesini temel alması ve dolayısı ile yeni bir dil öğrenimi gerektirmemesi,
- Görme engelli bireyler tarafından sıklıkla kullanılan Tablo 1’de listelenmiş olan araçların veri girişi için oluşturulmuş tuş dizilimlerini temel alması ve bu araçların kullanılması sürecinde edinilmiş alışkanlıkların avantaja dönüştürülmesi,
- Hâlihazırda var olan ticari ürünler ile karşılaştırıldığında sahip olduğu düşük maliyeti,
- Tasarımına yönelik yöntem ve bilgi birikiminin paylaşılabirliği açısından açık kaynaklı olması,
- Yüksek seviyede teknik bilgi gerektirmeyen, kendin-yap felsefesini temel alması,
- MusicXML dosya biçimini destekleyen tüm nota yazım programları ile uyumlu çalışması,

- Ortaya koyacağı yöntem ve bilgi birikimi yönü ile benzer çalışmalara ışık tutabilecek bir niteliğe sahip olması açılarından önem taşımaktadır.

1.4. Problem Cümlesi

Braille alfabesini temel alan, titreşim uyaranlı arayüze sahip düşük maliyetli bir araç aracılığı ile görme engelli bireylerin müzik okumasını sağlamak mümkün müdür?

1.5. Alt Problemler

1. Tasarlanacak aracın maliyeti hâlihazırda var olan ticari ürünler ile karşılaştırıldığında ne orandadır?
2. Parmak uçlarına uygulanan titreşim dokunsal uyarıların algılanabilirlik oranı, motor titreşim sürelerine bağlı olarak ne şekilde değişmektedir?
3. Parmak uçlarına uygulanan titreşim dokunsal uyarıların algılanabilirlik oranı, yaşa bağlı olarak ne şekilde değişmektedir?
4. Parmak uçlarına uygulanan titreşim dokunsal uyarıların algılanabilirlik oranı, cinsiyete bağlı olarak değişiklik göstermekte midir?
5. Parmak uçlarına uygulanan titreşim dokunsal uyarıların algılanabilirlik oranı, bireyin ellilik² durumuna bağlı olarak ne şekilde değişmektedir ?
6. Hücreleri meydana getiren noktaların bitişik-ayrık olma durumunun, algılama başarımına etkisi ne şekilde değişmektedir ?

1.6. Sayıtlar

Araştırma çerçevesinde faydalanılan kaynakların geçerli ve güvenilir olduğu, deney gruplarına uygulanan testler sırasında deneklerin verdiği cevapların gerçeği yansıttığı sayıtlarından yola çıkılmıştır.

² Bireylerin sağlak ve solak olma durumu.

1.7. Sınırlılıklar

Araştırma çerçevesinde tasarlanan araç, altı noktalı Braille alfabesi ile sınırlandırılmıştır. Bununla birlikte, tasarlanan arayüzün doğası gereği çok satırlı Braille müzik yazıları bu araştırmanın kapsamı dışında tutulmuştur.

1.8. İlgili Araştırmalar

Araştırma çerçevesinde çeşitli tasarım, makale, proje ve patentler kaynak olarak değerlendirilmiş ve taranmıştır. Konu ile ilgili kaynakların büyük çoğunluğu yabancı dilde olması sebebiyle öncelikle kaynaklar Türkçeye çevrilmiş, ve araştırmanın konusu ile ilgili veriler değerlendirilmiştir. Konu ile ilgili literatürde geçen terimler orijinal dilinde (İngilizce) olarak kullanılmıştır.

Tüm bu çalışmalar tarihsel süreçte sıralamaya alındığında Braille yazımı ile ilgili ilk tasarımın Perkins Braille olduğu görülmektedir. Perkins Braille, görme engellilerin Braille hücrelerindeki kabartma yazılarını daktilo stili bir cihaz ile yazmalarını sağlamaktadır. Sahip olduğu arayüz ile 6 ve 8 hücreli tüm kombinasyonları yazabilmek mümkün hale gelmektedir. 1890'larda ilk tasarım Frank H. Hall tarafından daktilo biçiminde yapılmış olup ilerleyen süreçte daha fazla yeniliğe gidilerek 1930 yılından itibaren David Abraham tarafından geliştirilerek en kullanışlı ve dayanıklı halini almıştır (Howe Press & Perkins Braille, 2004). Araştırma bünyesinde tasarlanmış olduğumuz vBraille üzerinde bulunan titreşim motorlarının konumları, Perkins Braille'in tuş takımından yola çıkarak belirlenmiştir.

Titreşimli geri bildirim sistemi üzerine literatür araştırmasında karşılaştığımız en eski tarihli çalışma, 2007 yılında Carlos Blas tarafından görme engelli müzisyenlerin enstrümanlarını çaldıkları sırada müziği okumalarını kolaylaştırmak amacıyla ortaya konmuştur. "Inaudible MIDI interpretation device"³ Blas, bu çalışmada MIDI biçimdeki müziksel dili, Braille kabartmasına dönüştürebilen bir araç önermiş ve patentini almıştır. Tasarım, canlı performans sırasında, MIDI biçimindeki müziksel bilgiyi ilgili donanım ve yazılımlar aracılığı ile dönüştürerek, performansçının koluna takılan dokunsal bir Braille hücresi üzerindeki altı noktayı temsil eden

³ U.S. Patent no. 0,166,693 A1, 2007

hareketli pimleri tetiklemesi prensibine dayanmaktadır (Google Patents, [16.07.2020]).

2012’de Stephen Sophorn Lim’in “Touch sense device for the visually impaired⁴” isimli patent önerisi araştırmamızla ilişkilidir. TSD (Touch Sense Display) yenilikçi yaklaşımı, düşük kontrollü bir akım yaratan bir elektronik titreşim yöntemi ile geliştirilmesi için bir öneri yapılmıştır. Bu cihaz ile bilgisayar ortamında Braille alfabesini okumak, yazmak ve düzenlemek mümkün hale gelmektedir (Google Patents, [16.07.2020]).

Titreşimli geri bildirim sistemi üzerine yapılan bir diğer çalışma ise 2013 yılında, Hugo Nicolau tarafından gerçekleştirilen “UbiBraille” projesidir. Tasarım, geleneksel Braille alfabesinin, titreşim-dokunsal geri bildirim aracılığı ile okunması prensibine dayanmaktadır. UbiBraille, gerçekleştirilen alfabetik okumaya yönelik denemelerde büyük oranda başarı sağlamıştır (Hugo Nicolau, 2013, 1).

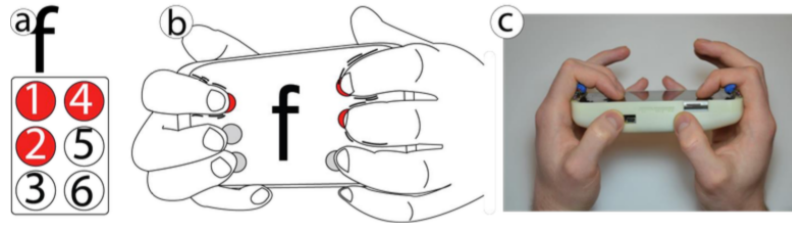


Şekil 9: Braille Okuma Vibrotaktıl Prototipi UbiBraille

Hugo Nicolau, João Guerreiro, Tiago Guerreiro, Luís Carriço “UbiBraille: Designing and Evaluating a Vibrotactile Braille- Reading Device.” **International ACM SIGACCESS Conference on Computers and Accessibility** (Bellevue, USA. 2013): 1.

Bir diğer çalışma, 2015 yılında Hugo Nicolau’nun tasarladığı “HoliBraille”, var olan dokunmatik mobil cihazlarla kullanılabilen ve Braille temelli girdi-çıkıtlı işlemlerini mümkün hale getirmek amacıyla tasarlanmıştır. HoliBraille, Braille öğrenmeye yönelik eğitim amaçlı ve görme engelli bireyler için bir iletişim aracı niteliğindedir (Hugo Nicolau, 2015, 1).

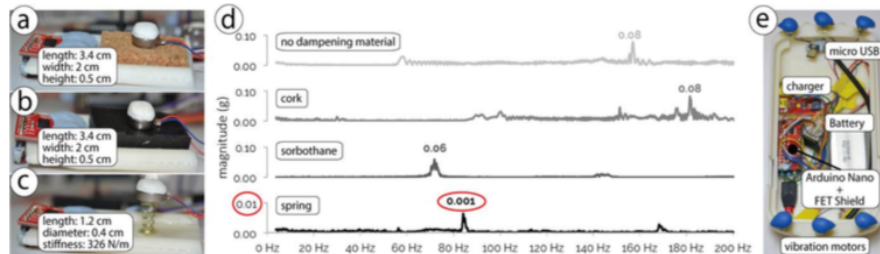
⁴ U.S. Patent no. 0,295,232 A1, 2012.



Şekil 10: Holibraille, Dokunmatik Ekranlı Mobil Cihazlar İçin Çok Noktalı Bir Vibrotaktıl Çıktı Sistemi

Hugo Nicolau, Kyle Montague, Tiago Guerreiro, André Rodrigues, Vicki L. Hanson “HoliBraille: Multipoint Vibrotactile Feedback on Mobile Devices.” **The 12th Web for All Conference** (Florence, Italy 2015): 1.

Çalışmada, maliyet, temin edinilebilirlik ve mobil ayarlara uygunluk adına seçilmiş olan, mantar, sorbothane ve yay gibi bir dizi farklı titreşim engelleyici malzemeler kullanılarak, araçtan geçen titreşim miktarını ölçmek için 500 Hz örnekleme frekansında ivmelenme okumaları toplanmıştır. 6 saniyelik ivme verileri kaydedilmiş ve bir Fast-Fourier Dönüşümü ile frekans alanına dönüştürülmüştür. Gerçekleştirilmiş olan testlerden elde edilen bulgular, vBraille tasarımı sürecinde bize önemli ipuçları sağlamıştır. HoliBraille çalışması ivmelenme ölçümlerden alınan sonuçlar Şekil 11’de gösterilmiştir.



Şekil 11: Holibraille, Farklı Materyaller Üzerinde Titreşim Ölçüm Grafiği

Hugo Nicolau, Kyle Montague, Tiago Guerreiro, André Rodrigues, Vicki L. Hanson “HoliBraille: Multipoint Vibrotactile Feedback on Mobile Devices.” **The 12th Web for All Conference** (Florence, Italy 2015): 2.

Çalışmadan elde edilen sonuçlar, ayrıca, sorbotenin, yalnızca ivmelenme frekansını azaltabildiği, yayın ise buna ek olarak ivmelenme genliğinde de 80 kat azalma sağlayabildiğini göstermektedir.

Titreşimli geri bildirim sistemi üzerine yapılan bir diğer çalışma 2017 yılında Luke Vincent Naman tarafından, Güney Kaliforniya Üniversitesi Keck Tıp Fakültesi ‘nin

düzenlemiş olduđu “USC Stevens Student Innovator Showcase Awards” öğrenci başlangıç projeleri (startup) yarışmasında geliştirilen “Vibraille”dir. Vibraille tek el kullanılarak okuma olanağı sağlayan, parmak ve bilek üstüne denk gelecek şekilde yerleştirilmiş motorların Braille alfabesine uygun biçimde titreşmesi prensibine dayalı bir araçtır. Arduino mikro denetleyicisi kullanılarak geliştirilmiş bu tasarım, Braille temelli rakam ve harflerin okunmasını mümkün kılmaktadır (USC Stevens Student Innovator, [14.03.2020]).



Şekil 12: Vibraille Görseli

Vibraille. <https://www.youtube.com/watch?v=KW5xf4pY1xU> [6.05.2020].

2018 yılında, Dominic R. Labbé tarafından patenti alınan (Portable device with virtual tactile keyboard and refreshable Braille display⁵) Braille tabanlı okuma-yazma aracı, değiştirilebilen çeşitli paletler ile dokunmatik bir ekran üzerinden yazım imkânı sunmaktadır. Portatif Braile cihazı, yenilenebilir Braile ekranı sayesinde dokunmaya duyarlı yüzeyden yazılan Braile verilerini alır ve aracın alt bölümünde bulunan, Braille hücrelerini taklit eden piezoelektrik noktalara bu verileri iletir. Mobil cihazlarla uyumlu olan bu araç engelli bireyler için de kullanım rahatlığı bakımından Bluetooth, ve gps gibi fonksiyonlar sunmaktadır (Google Patents, [16.07.2020]).

⁵ U.S. Patent no. 9,965,974 B2, 2018.

2. MÜZİK BRAILLE

Müzik braille, geleneksel müzik notasyonunun ifadelerine karşılık gelen braille hücreleri ve müzikal dilin bu biçimde ifade edilebilmesine yönelik oluşturulmuş birtakım kurallardan meydana gelmektedir. Braille müziğini biçimlendirmenin üç stili veya yöntemi vardır, bunlar:

- Alt alta yazılan ölçüler,
- Yan yana yazılan ölçüler,
- Tek satır halinde yazılan ölçüler

olarak belirlenmiştir (Nichols, 2010, 7).

Enstrümantal solo veya bir topluluktaki bir katılımcı için tek bir parçadan oluşan müzik eseri, tek satırlı formatta sunulmaktadır (The Braille Authority of North America, 2016, 171). Şekil 13’de, bir melodinin geleneksel müzik notasyonu ve tek satırlı Music Braille ile yazılmış biçimi görülmektedir.



Şekil 13: Müzik Notası ve Tek Satırlı Müzik Braille Yazımı

2.1. Tonalite

Müzik eseri boyunca ortak olarak kullanılan diyez ve bemol sayısını temsil etmektedir. (Nichols, 2010, 15). Tonalite, bir, iki veya üç ses değiştirici işaretten oluştuğunda, art arda karşılık gelen bemol veya diyez işaretlerle Braille olarak temsil edilir. Dört veya daha fazla ses değiştirici işaret yazılacağı zaman, işaret sayısı Braille rakamı ile ses değiştirici işaretten önce yazılır (The Braille Authority of North America, 2016, 61).

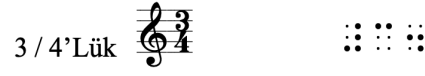
İki Diyez ton belirtimi		
İki Bemol ton belirtimi		
Üç Diyez ton belirtimi		
Üç Bemol ton belirtimi		
Dört Diyez ton belirtimi		
Dört Bemol ton belirtimi		

Şekil 14: Tonalite

2.2. Ölçü Rakamları

Ölçü rakamları, bir ölçüdeki vuruş sayısını ve her vuruşun nota değerinin ne olduğunu temsil eder. Örneğin, "4/4"lük ölçü rakamları, bir ölçü içerisinde dört vuruş olduğu anlamına gelmektedir ve her vuruşun dörtlük nota değeri vardır (Nichols, 2010, 14).

Müzik braille yazısında ölçü rakamları, 3-4-5-6 numaralı noktardan oluşan rakam işaretinin ardından, önce vuruş sayısını, ardından nota değerini gösteren braille hücreleri biçiminde yazılır. Nota değeri yazılırken dikkat edilmesi gereken nokta, bu değeri belirten sayının 2-3-5-6 numaralı noktaların bulunduğu alana, yani hücrenin alt kısmına yazıldığı halidir.



Şekil 15: Ölçü Rakamları

2.3. Oktav işaretleri

Braille müzik sisteminde, bir notanın hangi oktavda yer aldığını gösterebilmek amacıyla oktav işaretlerinden yararlanılmaktadır. Son derece ustalıkla tasarlanmış bu yöntem ile, bir notanın hangi oktavda olduğu belirtilirken dikkat edilmesi gereken bir dizi kural mevcuttur (Nichols, 2010, 10-11):

1. Oktav, ait olduğu notadan hemen önce gelir ve nota ile oktav işareti arasına başka hiçbir işaret gelmez (MEB ,1991, 33).
2. Bir müzik satırında ilk notadan önce daima bir oktav işareti gereklidir. (Nichols, 2010, 11).
3. İki nota arasındaki aralık ikili veya üçlü olduğunda oktav işareti gerekli değildir.
4. İki nota arasındaki aralık dördü veya beşli olduğunda yalnızca ikinci nota farklı bir oktavda ise bir oktav işareti gerekir.
5. İki nota arasındaki aralık altılı veya daha büyük olduğunda oktav işareti her zaman kullanılır (Nichols, 2010, 12).

6. Bir melodi içerisinde iki nota arasındaki aralık dördü aralıktan az ise ikinci notanın oktav işareti almasına gerek yoktur (MEB,1991, 33).



Şekil 16: Oktav İşaretleri

2.4. Ses Değiştirici İşaretler

Ses değiştirici işaretler, notadan önce ilgili notanın perdesini değiştirmek amacıyla yazılmaktadır (Nichols, 2010, 16). Genel olarak mürekkep baskıdaki ses değiştirici işaretler braille yazımında da her zaman kullanıldıkları yerlerde gösterilmektedir. Mürekkep baskıda olduğu gibi sadece aynı nota için konulan ses değiştirici işaret bir ölçü boyunca etkilidir (MEB,1991, 37).

Diyez	⠠⠠⠠
Bemol	⠠⠠⠠
Çift Diyez	⠠⠠⠠⠠
Çift Bemol	⠠⠠⠠⠠
Naturel	⠠⠠

Şekil 17: Ses Değiştirici İşaretlerin Braille Yazımı

2.5. Notalar ve Nota değerleri

“Music Braille Handout” kaynağında bahsedildiği üzere, müzikteki yedi nota “do, re, mi, fa, sol, la, si”, braille alfabe yazımındaki “d, e, f, g, h, i, j” harfleri ile aynı noktalarla gösterilmektedir. “Do” notasını temsil etmek için 1-4-5 noktaları kullanılır. “La” notasını temsil etmek için 2-4 noktaları kullanılır. “Si”de 2-4-5 noktaları ile yazılmaktadır (Nichols, 2010, 4-5). Müzik braille’de nota hücrelerine eklenen 6. nokta ile, dörtlük ve altmışdörtlük, 3. nokta ile ikilik ve otuzikilik, 3. ve 6. noktalar ile de birlik ve onaltılık notalar elde edilir (MEB,1991, 31). Sekizlik notalarda herhangi bir nokta bulunmamaktadır⁶ (Nichols, 2010, 8).

	C	D	E	F	G	A	B
Tam ve 16’lık	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠
Yarım ve 32lik	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠
Çeyrek ve 64’lük	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠
8’lik ve 128’lik	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠	⠠⠠⠠
Notaya eklenen nokta	⠠⠠	⠠⠠	⠠⠠	⠠⠠	⠠⠠	⠠⠠	⠠⠠

Şekil 18: Nota İsimleri, Değerleri ve Nokta

2.6. Sus İşaretleri

Sus işareti de müzik braille üzerinde mutlaka gösterilir. Ölçü rakamı ne olursa olsun mürekkep baskıda bir ölçülük sessizlik birlik sus ile gösterilir. Bunun haricinde, 4/2’lik ölçüde, çift birlik sus (double whole rest) kullanılabilir.

⁶ 128’lik notalar da aynı özelliğe sahiptir.

256’lık nota veya sus işaretini diğer değerlerden (örn.: 4’lük, 8’lik) ayırmak adına mutlaka bir değer işareti (value sign) kullanılır (The Braille Authority of North America, 2016, 50).

Tam ve 16’lık sus	⠠⠠
Yarım ve 32lik sus	⠠⠠
Çeyrek ve 64’lük sus	⠠⠠
8’lik ve 128’lik sus	⠠⠠
256’lık sus	⠠⠠⠠⠠
İki Tam Notalık Sus	
“Double Whole Rest”	a) ⠠⠠⠠
	b) ⠠⠠⠠⠠

Şekil 19: Sus İşaretleri

2.7. Uzatma Bağı

Uzatma bağı, iki notanın ilkinden hemen sonra yerleştirilir (The Braille Authority of North America, 2016, 81). İki aynı notayı bağlamak için 1-4 hücre işareti kullanılır (Nichols, 2010, 22).



Şekil 20: Uzatma Bağı

2.8. Ölçü Çizgisi

Her ölçü bitişi için ölçüyü belirtmek adına boşluk kullanılmaktadır (The Braille Authority of North America, 2016, 43). Ölçü bitişi bazı formatlarda notanın önüne boşluk bırakıldıktan sonra “1-2-3” noktalarının geldiği ve tekrar boşluk bırakıldığı bir işaret ile de gösterilmektedir.



Şekil 21: Boşluk İle Belirtilen Ölçü Çizgisi

Uzun ölçüleri kolay okumak, daha kısa bölümlere ayırmak amaçlı kullanılan noktalı veya kesikli ölçü çizgisi, boşluklar arasında “⋮” işaretiyle temsil edilir. (The Braille Authority of North America, 2016, 43).



Şekil 22: 1. ve 4. Noktalarla Temsil Edilen Noktalı Ölçü Çizgisi

3. MUSICXML

MusicXML, bir mzik notasyon ıktısının dijital olarak tercme ediliř hali ve yazılımlar arasında kullanılan bir kod dilidir. Mzikal bilgi, notasyon yazılımları, sıralayıcılar (sequencer) ve diğ er performans programları, mzik eđitimi programları ve mzik veri tabanları ile kullanılabilir olacak řekilde tasarlanmıřtır (MakeMusic Inc., 2017, 1). Bu durumda, MusicXML, mzikal notasyon ve vBraille arasındaki iletiřimi kurmak amacıyla kullandıđımız en nemli ara olarak karřımıza çıkmaktadır. MusicXML sayesinde bir mzik notasyonu programından alınan “.musicxml” formatlı dosya sayesinde mzik eserindeki tm detaylara ulařabilmekteyiz.

XML (eXtensible Markup Language), iřaretleme dillerinin temeli olan “SGML”den ortaya ıkan, basit ve ok esnek geniřletilebilir bir iřaretleme dilidir. XML, halen gncellenmektedir ve birok kiřinin dřndđ gibi HTML in yerini almak iin geliřtirilmemiřtir (Benz, 2003, 3).

XML temel olarak byk lekli elektronik yayıncılıkta kullanılmak zere tasarlanmıř ve eřitli verilerin web veya herhangi bir platform zerinden bilgi transferinin sađlanmasıda nemli bir rol oynamaya bařlamıřtır. Kısaca tanımlamak gerekirse, XML, HTML gibi iřaretleme etiketleri kullanan bir dil olmakla beraber, veriyi tanımlamak iin dizayn edilmiřtir. nceden belirtilmiř etiketler (tag) bulunmadıđı gibi kod yazan kiři kendi etiketlerini oluřturur. XML zerinde bir veriyi tanımlamak iin dokman tipi tanımlaması DTD veya XML Schema kullanmak gerekmektedir. Bu sebeple XML kodunun DTD veya XML Schema ieren, kendini tanımlayıcı bir yapıda olması gerekmektedir (Durmaz, [15.05.2020]).

XML kullanmak, dilin temel szdizimi (syntax) iin endiřelenmekten kurtulmayı sađlamakta, neyi nasıl temsil edeceđimize dair bize olanak sunmaktadır (MakeMusic Inc. , 2017, 6).

MusicXML'den nce, yaygın olarak desteklenen tek mzik notasyonu deđiřim biimi MIDI'dir. MIDI, sıralayıcılar (sequencer) gibi performans uygulamaları iin

günümüzde kullanılan en işlevsel formattır, ancak müzik notasyonu gibi detaylı bilgi gerektiren uygulamalar için yeterli değildir. MIDI, fa diyez ve sol bemol arasındaki fark gibi birçok bilgiyi barındırmamaktadır (MakeMusic Inc., 2017, 5).

Birçok XML kitabı, mümkün olduğunda, öğelerde (elements) özniteliklerden (attributes) ziyade anlambilimi (semantics) temsil etmeyi önermektedir. Gerçekte temsil ettiğiniz şeyin bir bölümden (part) daha fazlası olduğunu fark ederseniz, bir öğeyle hiyerarşik bir yapı oluşturabilirsiniz (MakeMusic Inc., 2017, 8).

Birçok müzik yazılım geliştirici firma MusicXML kullanmaktadır. MusicXML'i çeşitli yönleriyle kullanan firmalar:

- Nota düzenleme yazılımları: Finale, Sibelius, Dorico, MuseScore, capella, Encore, ve Overture.
- Web tabanlı müzik düzenleme siteleri: Noteflight, SmartMusic, Soundslice, Flat, Gustaf, ve Scorio.
- DAW yazılımları (Digital audio workstation): Cubase, Digital Performer, Logic, Reaper, ve Sonar.
- Yazılı müzik tarama programları: SmartScore, PhotoScore, SharpEye Music Reader, capella-scan, ScoreMaker, PlayScore, ve Audiveris.
- Tablatür düzenleyiciler (Tablature editors) Guitar Pro, Progression, Fandango, TablEdit, TaBazar, ve TuxGuitar.
- iOS notasyon programları hem iPhone hem de iPad için: Komp, Notion, Symphony Pro, ve SeeScore.
- Android nota programları: Ensemble Composer, Notate, ve NotateMe.
- Elektronik müzik yazılımları: Newzik, Blackbinder, ve OrganMuse.
- Müzikoloji yazılım ve araçları: music21 ve MelodicMatch (MakeMusic Inc., 2017, 7).

Şekil 23'te, MusicXML üzerinde temel müzikal ifadeleri tanımlamak adına, dördüncü oktavdaki birlik do notasının nasıl temsil edildiği gösterilmiştir.

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
"-//Recordare//DTD MusicXML 3.1 Partwise//EN"
"http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.1">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
      <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
  </part>
</score-partwise>

```

Şekil 23: Sol Anahtarında Dört Dörtlük “Do” Notasının MusicXML Üzerinde Gösterimi

Şekil 23'te tonalite, do (C) majör olarak görülmektedir, diyez veya bemol yoktur, bu yüzden beşliler (fifths) ögesi 0'dır. Eğer 2 diyeze sahip re (D) majör tonu olsaydı, beşliler ögesi 2 olarak yazılacaktı. MusicXML elementlerinden olan "Fifths", beşliler çemberindeki diyezler sırasında yukarıya giden belirteçlerinin ifade edilmesi amacıyla kullanılmaktadır. MusicXML diyez ve bemol için ayrı öğeler kullanmak yerine tek bir öğe ile standart ton belirteçlerini temsil etmemizi sağlar (MakeMusic Inc., 2017, 11). Tasarlanan aracın okuyabileceği biçimin oluşturulması sürecinde müzikal ifadelerin temsilleri MusXML üzerindeki satırlardan çekilmektedir. Bu sebeple dikkate alınan müzikal parametreler:

- 1) Ölçü rakamı,
- 2) Tonalite,
- 3) Oktav numarası,
- 4) Nota isimleri ve değerleri,
- 5) Ses değiştirici işaretler,
- 6) Sus işaretleri,
- 7) Uzatma bağı,
- 8) Nokta,
- 9) Ölçü çizgisi, gibi 9 adet temel müzikal bilgi vBraille için kullanılmıştır.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
"-//Recordare//DTD MusicXML 3.1 Partwise//EN"
"http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.1">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
```

Şekil 24: MusicXML Döküman Özellikleri

Şekil 24'te yer alan, MusicXML'de en üstte bulunan bu bölüm, MusicXML kullandığımızı, bölümlerin (parts) ölçüler (measures) içerdiği, kısmen bir müzik notasyonu kullandığımızı söylediğimiz yerdir. DTD için bir internet konumu (URL) içeren genel bildiri, yalnızca referans amaçlı kullanılmaktadır. MusicXML dosyalarını okuyan çoğu uygulama, kullanıcının cihazında MusicXML DTD'lerin yerel bir kopyasını yüklemek isteyecektir. “<score-partwise version="3.1">”, bir kök hücre türüdür. “<score-partwise>” elementi, ölçülerin, bölümlerden oluştuğu bir kısımdır (MakeMusic Inc., 2017, 11).

```
<measure number="1">

  <attributes>

    <key>

      <fifths>0</fifths>

    </key>
```

Şekil 25: MusicXML Müzikal Özellikler

“<measure number="1">” elementinde ilk bölümdeki ilk ölçü başlatılır. “<attributes>” Nitelikler (attributes) ögesi, bu bölümde takip edilen notaları ve müzik verilerini yorumlamak için gerekli bilgileri içerir.

MusicXML süreleri de kesirdir. Payda (denominator) nadiren değişmesi gerektiği için, bölümler ögesinde ayrı ayrı temsil edilir, böylece her bir nota ile yalnızca payın (numerator) ilişkilendirilmesi gerekir Bu, MIDI'de nota sürelerini temsil etmek için kullanılan şemaya benzemektedir. Anahtar ögesi (key), bir anahtar imzasını (key signature) temsil etmek için kullanılır (MakeMusic Inc., 2017, 12).

```
<time>

  <beats>4</beats>

  <beat-type>4</beat-type>
```

Şekil 26: MusicXML Üzerinde Ölçü Rakamı

MusicXML’deki zaman ögesi (time), bir eserin ölçü rakamını temsil eder. Ölçü rakamının vuruş ve vuruş tipi sırasıyla (beat and beat-type) olarak temsil edilir (MakeMusic Inc., 2017, 12).

```
<clef>  
  
  <sign>G</sign>  
  
  <line>2</line>
```

Şekil 27: MusicXML Üzerinde Anahtar

Şekil 27’de standart olarak kullanılan sol anahtarı, portenin ikinci satırındaki bir sol anahtarı (G clef) olarak temsil edilir. Bu bölüme kadar “</attributes>” nitelikler biter ve ilk nota başlatılır (MakeMusic Inc., 2017, 12).

```
<note>  
  
  <pitch>  
  
    <step>C</step>  
  
    <alter>1</alter>  
  
    <octave>4</octave>
```

Şekil 28: MusicXML Üzerinde Perde

MusicXML, perdeyi (pitch) üç bölüme ayırır: adım ögesi (A, B, C, D, E, F veya G), isteğe bağlı bir değiştirme (alter) ögesi (bemol için -1, diyez için 1) ve bir oktav ögesi, orta do (C) ile başlayan oktav için 4’tür. Değiştirme değerleri, çift bemol ve çift diyez için -2 ve 2 olarak kullanılabilir (MakeMusic Inc., 2017, 20). Şekil 28’de oktavdaki 4 sayısı, oktavın orta do ile başladığını gösterir. Dolayısıyla bu nota bir orta do’dur (MakeMusic Inc., 2017, 12).

```
<note>

  <rest measure="yes"/>

  <duration>2</duration>

  <voice>1</voice>

  <type>eighth</type>
```

Şekil 29: MusicXML Üzerinde Sus İşareti

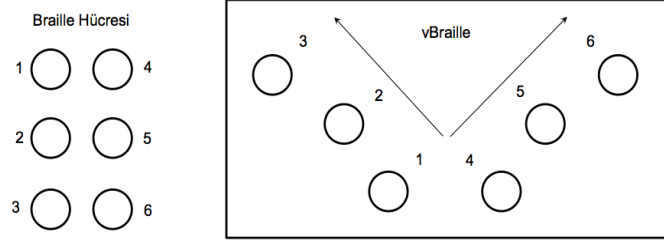
Sus işareti için, perde (pitch) ögesi, sus işareti yerine kullanılmaktadır (MakeMusic Inc., 2017, 20). Şekil 29’da sekizlik bir sus işareti gösterilmiştir. “<type>” yazım ögesini takiben, bir veya daha fazla nokta ögesi “<dot/>” ögesi, noktaları belirtmek adına kullanılır (MakeMusic Inc., 2017, 30). Bağlı notanın yerleşimi, “tie” ögesi ile belirtilir. Bağ ögesi, bir bağlantının başlangıç notası için bir başlatma (start) ve bitiş notası için bir durdurma (stop) türü barındırır. Bağ, bir önceki notaya <tie type = "stop">, sonraki notaya <tie type = "start"> olarak yerleştirilir (MakeMusic Inc., 2017, 20). Nota finali, “</note>” , ölçü finali, “</measure>” , bölüm finali, “</part>” , müzik notasyonunun finali de “</score-partwise>” ile ifade edilir (MakeMusic Inc., 2017, 13).

4. VBRAILLE

vBraille, HIMS ve benzeri araçların tuş dizilimini temel alan, titreşim tabanlı bir müziksel okuma aracıdır. Aracın tasarımı sırasında Tablo 1’de adı geçen mevcut ticari ürünlerin tuş dizilimleri temel alınmış, böylelikle önceden edinilmiş kullanıcı alışkanlıklarının avantaja dönüştürülmesi hedeflenmiştir. Bununla birlikte, sol ve sağ el baş parmakları altına yerleştirilmiş iki adet düğme aracılığı ile braille hücreleri arasında geçiş yapılması sağlanmıştır.

4.1. VBraille Tasarımı

Tasarım, braille hücrelerini meydana getiren noktaları temsil eden, sol elde ve sağ elde işaret, orta ve yüzük parmaklarını uyaran titreşim motorları, bir mikro denetleyici, bir SD kart okuyucu ve güç kaynağı gibi bileşenleri içeren bir muhafaza içerisine yerleştirilmiş donanım ve yazılımlardan oluşmaktadır.



Şekil 30: Bir Braille Hücresi (Solda) ve VBraille Üzerinde Hücre Temsili (Sağda)

VBraille’in poliüretan reçine ile üretilmiş bir gövdenin içerisine yerleştirilmiş ilk prototip tasarımı Şekil 31’de görülmektedir.



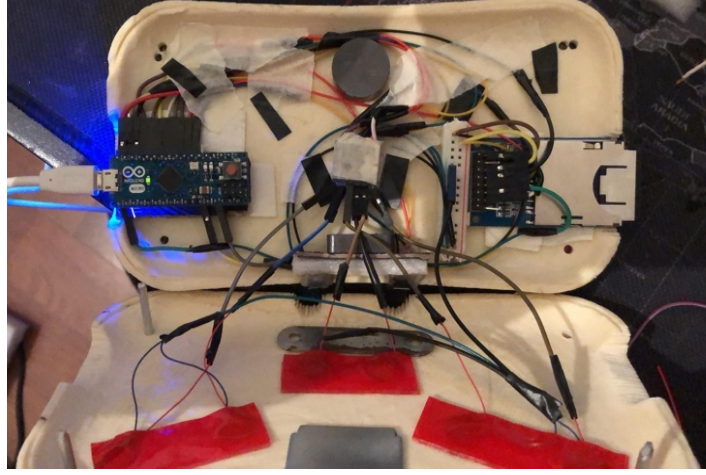
Şekil 31: İlk VBraille Prototipi

Gövde, herhangi bir ergonomik kaygı güdülmeden tasarlanmış olup, tamamen ilk denemelerin gerçekleştirilebilmesi amacı ile oluşturulmuştur. Poliüretan reçinenin şekillendirilmesi için RTV2 tipi silikon kalıplayıcıya ihtiyaç duyulmaktadır. Böylelikle bu kutunun tasarımı için hem silikon kalıplayıcının hazırlanması, hem de poliüretan kalıbın hazırlanması için iki aşamadan geçilmiştir.



Şekil 32: RTV 2 Silikon Kalıp ve İçerisinde Poliüretan Reçine

Poliüretan reçine ile yapılmış versiyonda titreşim motorları iki farklı yerleştirme yöntemiyle test edilmiştir. İlk yöntemde titreşim motorları bant üzerine üst kapağa değmeyecek biçimde, Şekil 33'deki gibi yerleştirilmiştir.



Şekil 33: İlk VBraille'in İç Tasarımı⁷

3 mm yüksekliğe sahip titreşim motorları, 5 mm derinlikteki parmak çukurlarına yerleştirildiğinde arta kalan boşluklar için, 3'er mm'lik filtre süngerleri yerleştirilmiştir. Bu tasarım ile gerçekleştirilen denemelerde, süngerlerin parmaklara iletilen titreşimin zayıflamasına neden olduğu görülmüştür.



Şekil 34: İlk VBraille Filtre Sünger Yerleşimi

Bu sebeple, sünger ile motorlar yer değiştirilerek yeniden denenmiş ve sünger materyalini izolasyon süngerleri olarak kullanmanın sert yapıya sahip olan filtre süngerine nazaran daha verimli olduğu gözlemlenmiştir. Gövde, titreşim rezonansını azaltmak adına orta kısımdan bir plastik kelepçe yardımıyla sıkıştırılmıştır. Aynı

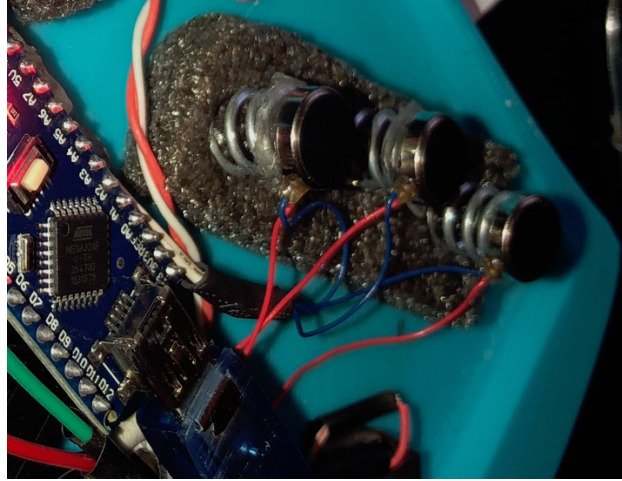
⁷ Kalın ve sağlam olması adına kırmızı çift taraflı silikon bantlar kullanılmış, motorların üst kapağa değmeden sabitlenmesini sağlamıştır.

şekilde titreşim motorlarının 45 derecelik açılarla yerleştirilmesi titreşim algılamada daha iyi netice verdiği görülmüştür.



Şekil 35: İlk VBraille İzolasyon Süngeri Yerleşimi

İkinci tasarıma ait prototip, önceki tasarıma göre daha hafif, ince ve parmak yerleşim noktaları daha yakın olacak şekilde bilgisayar üzerinde, Tinkercad programında tasarlanmış ve bir 3D yazıcı aracılığıyla basılmıştır. 3D materyalinin sağlam ve hafif olması amacıyla PLA Filament kullanılmıştır. Motorlar, HoliBraille (Hugo Nicolau, 2015) tasarımından elde edilen verilerden yola çıkılarak, sert sünger üzerine yay ve onun üzerine silikon bantlar ile yapıştırılarak yerleştirilmiştir.



Şekil 36: VBraille Motor Yerleşimi

Test prototipi ile yapılan rastgele denemeler, sünger ve yay sistemi sayesinde, motorların meydana getirdiği titreşimlerin gövdeye yayılmasını büyük oranda

engellediğini ve bunun sonucunda, titreşimlerin parmaklarda daha net algılanabildiğini göstermiştir.



Şekil 37: VBraille 1. Test Cihazı

4.2. Cihaz Maliyeti

Braille temelli okuma araçlarının yüksek maliyeti temel alınarak tasarlanan bu arayüz, uygun fiyata sahip parçalar bir araya getirilerek oluşturulmuştur. Tasarım, 1 adet Arduino Nano Klon (ATmega328 mikroişlemcili), 1 adet Arduino SD kart modülü, 6 adet 3x10mm titreşim motoru, 6 adet 3D yazıcı yayı, 2 adet 2 pinli buton (push-button), 1 adet açma-kapama anahtarı, 3D yazıcı ile yazılmış alt-üst PLA filament kapak, silikon bant, 2 filtre süngeri ve çeşitli kablolar içermektedir. Bazı parçalar daha uygun fiyatlı olması adına farklı mağazalardan satın alınmıştır. Cihaz maliyet tablosu Tablo 2’ de gösterilmektedir.

Tablo 2: Cihaz Maliyet Tablosu

Parça	Özellikler	Adet	Fiyat ⁸
Arduino Nano Klon	ATmega328 mikroişlemcili	1	29,29 TL**
Titreşim motoru	3mm x 10mm	6	25,84 TL **

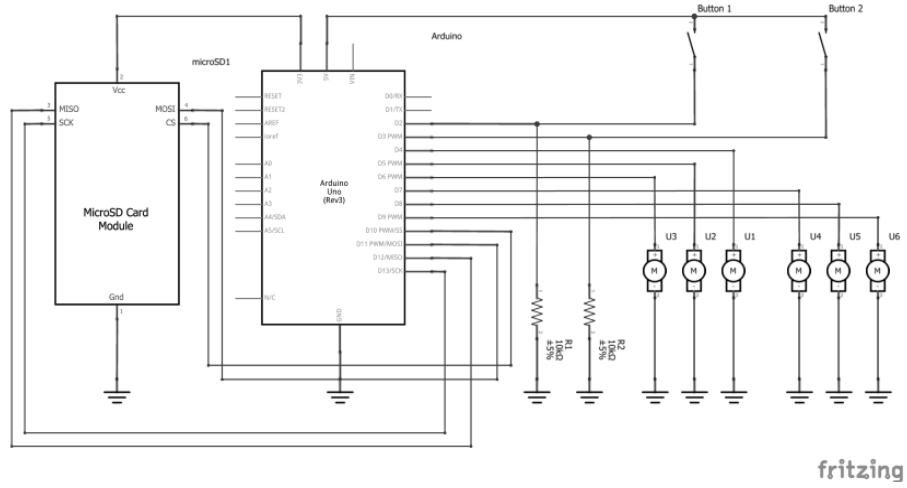
⁸ Tabloda adı geçen ürünlerin fiyatları, *Punto Copy (İstanbul) baskı merkezi, **roboshop.com, ***n11.com web siteleri üzerinden 12 Mayıs 2020 tarihinde alınmıştır.

Arduino SD kart modülü	3.3 ve 5volt'luk voltaj regülatörü	1	8,10 TL **
3D yazıcı yayı	1.2mm x 20mm Nikel (10'lu)	1	10,27 TL **
Buton	2 pinli 6x6x2mm Push Buton	2	0,56 TL **
PLA filament + baskı	108 gr + 2 parça baskı	1	70 TL*
Açma-kapama anahtarı	Tekli, dar, ışısız Anahtar Switch 3P	1	2,33 TL **
Filtre süngerı	10 PPI	1	26,95 TL ***
		Toplam:	173,34 TL

Tablo 2’de cihaz parçaları 12 Mayıs 2020 tarihi itibarıyla toplamda 173 TL’ye temin edilebilmektedir. Diğer ürünlerle karşılaştırmak istenildiğinde, Tablo 1’de yer alan en düşük fiyatlı Orbit Reader 20 Braille notebook (799.95 €) ile vBraille (173,34₺) arasında fiyat farkı olduğu görülmektedir.

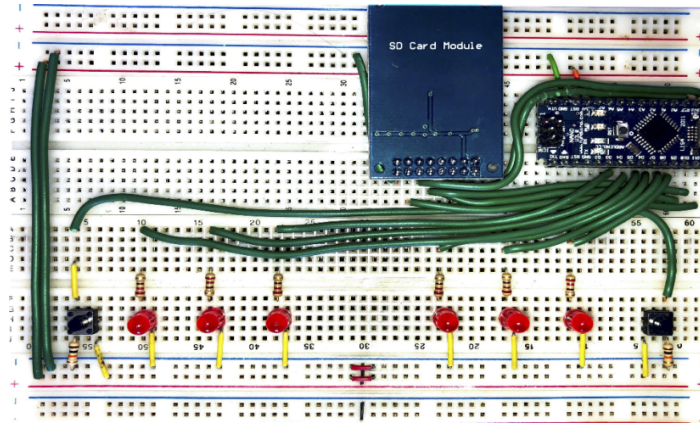
4.3. Donanım

vBraille donanımının merkezinde bir ATmega328 mikro denetleyicisi bulunmaktadır. vBraille dosyalarının saklanacağı SD kartı okumak amacıyla kullanılacak SD kart modülü, ATmega328 mikro denetleyicisi ile SPI protokolü üzerinden konuşmaktadır. Tasarımın titreşimli arayüzünü meydana getiren 6 adet titreşim motoru ve Braille hücreleri arasında gezinmek amacıyla kullanılan 2 adet buton, ilgili sayısal pinlere doğrudan bağlanmıştır. İki adet 10k’lık direnç ise, buton pinleri ile GND arasında pull-down görevi üstlenmektedir. Açma-kapama düğmesi mikroişlemciyi besleyen USB üzerindeki 5 volt’a bağlanmıştır. Devre elemanlarının bağlantı şeması Şekil 38’de görülmektedir.



Şekil 38: VBraille Donanım Şeması

Şekil 38’de yer alan breadboard üzerindeki devre bağlantıları, Şekil 39’daki şema ile aynı olmakla birlikte, burada titreşim motorlarının yerini LED aydınlatmaların aldığı görülmektedir. Bu devre, tasarım sürecinde programlama ve hata gidermek amacıyla kurulmuştur.

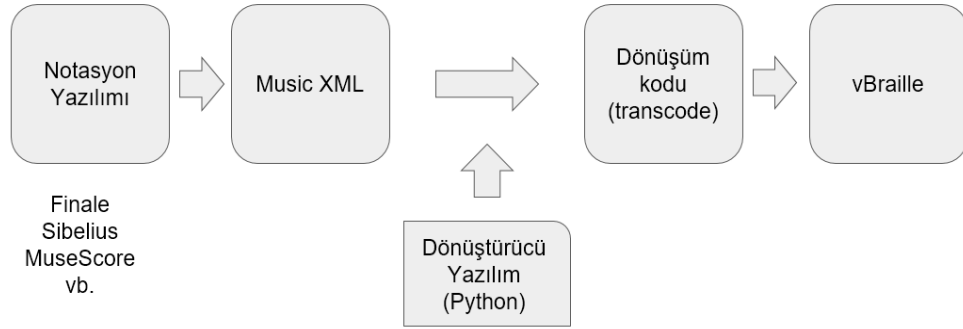


Şekil 39: VBraille Devresinin LED Aydınlatmalar İle Breadboard Üzerindeki Görünümü

VBraille Çalışma Adımları:

1. Sibelius, Finale, Musecore vb. bir müzik notasyon programı aracılığı ile yazılan müzik eseri MusicXML biçiminde kaydedilir.

2. Python ortamında geliştirilmiş bir yazılım ile MusicXML biçimindeki dosya ayrıştırılarak mikro denetleyicinin okuyacağı bir biçime .txt formatında bir vBraille koduna dönüştürülür.
3. Metin tabanlı bu yeni dosya bir SD kart üzerine kaydedilir.
4. SD kart vBraille'a yerleştirilir.
5. Mikro denetleyici, braille hücrelerini oluşturan noktalara karşılık gelen titreşim motorlarını doğru sıra ile çalıştırır.
6. Braille hücreleri arasında geçişler, ileri ve geri gezinti işlevi sağlayan düğmeler aracılığı ile gerçekleştirilir.
- 7.



Şekil 40: VBraille Çalışma Adımlarını Gösteren Şema

4.4. VBraille Dosya Biçimi

Tasarlanan vBraille kodu⁹ temel olarak düz bir metin dosyasıdır. Bir müzik braille hücrelerinin içeriği, toplamda 4 karakterden oluşan sembolik bir metin ile temsil edilmektedir. Karakter sayısının bu şekilde sabit tutulması ile, donanım tarafında, dosyanın ayrıştırılması açısından kolaylık sağlanması hedeflenmiştir. MusicXML dosyası içerisinden ayrıştırılan notasyona ait her bir içerik (sembol) için özel bir sembolik metin oluşturulur. Oluşturulan metnin 4 karakterden kısa olması durumunda metin, sonuna 1 veya daha fazla sayıda “_” karakteri eklenerek 4

⁹ VBraille açık kaynak kodu [23.08.2020]: <https://github.com/BurakAyaz/vBraille>

karaktere tamamlanır. Bütün bir vBraille dosyası, her bir sembolik metnin aralarında boşluk olmadan art arda yazılmasıyla oluşturulur.

Tonalite, zaman ifadesinden sonra gelmekte ve her bir beşliler çemberi ses değiştirici işaret ifadesi için birer sembol kullanılmaktadır. Beşliler Çemberi girdisi “#__” ifadesi ile kullanılmakta ve diyez için “>__” bemol için de “<__” yazılmaktadır. Kaç adet ses değiştirici işaret belirtilecekse adet sayısı kadar “+__” yazılmaktadır.

Ölçü rakamı, vuruş sayısı ve birim vuruş değeri de eklenerek elde edilen bir metin ile kodlanmaktadır. Metin sonuna eklenen 2 adet “__” karakteri ile kod 4 karaktere tamamlanır. Örneğin, 3/4’lük ölçü rakamı için vBraille kodu “>3__” devamında da “<4__” biçiminde oluşturulur.

Notalar, İngilizce ’deki sembollerinin küçük harfli karakterleri ile Tablo 3’deki gibi kodlanmaktadır. Sus işaretleri için de, İngilizce “rest” kelimesinin ilk harfi olan “r” karakteri seçilmiştir.

Tablo 3: VBraille Üzerinde Notaların Kodlanması

Do: c	Re: d	Mi: e	Fa: f	Sol: g	La: a	Si: b	Sus: r
-------	-------	-------	-------	--------	-------	-------	--------

Nota süreleri ise “0, 1, 2, 3” sayıları ile Tablo 4’teki şekilde ifade edilmektedir.

Tablo 4: VBraille Üzerinde Nota Değerlerinin Kodlanması

1’lik	16’lık	0
2’lik	32’lik	1
4’lük	64’lük	2
8’lik	128’lik	3

VBraille üzerinde bir nota, nota adı ve nota değerini gösteren iki karakter art arda getirilerek ve sonuna 2 adet “__” karakteri eklenerek kodlanır. Örneğin, 16’lık bir Re

notası için vBraille kodu “d0__”, 4’lük bir Sol notası ise “g2__” biçiminde olur. Benzer şekilde 2’lik bir sus, “r1__” biçiminde kodlanır.

vBraille üzerinde ses değıştirici işaretler ve nokta için seçilen karakterler Tablo 4’te görölmektedir. Kodlama, gerekli sayıda “_” karakteri eklenerek tamamlanır. Örneğin, diyez işareti için vBraille kodu “+__”, bemol için “-__” ve natürel için “n__” biçiminde oluşturulur. İki diyez veya iki bemol için kod iki kez tekrarlanır.

Tablo 5: VBraille Üzerinde Ses Değıştirici İşaretlerin Kodlanması

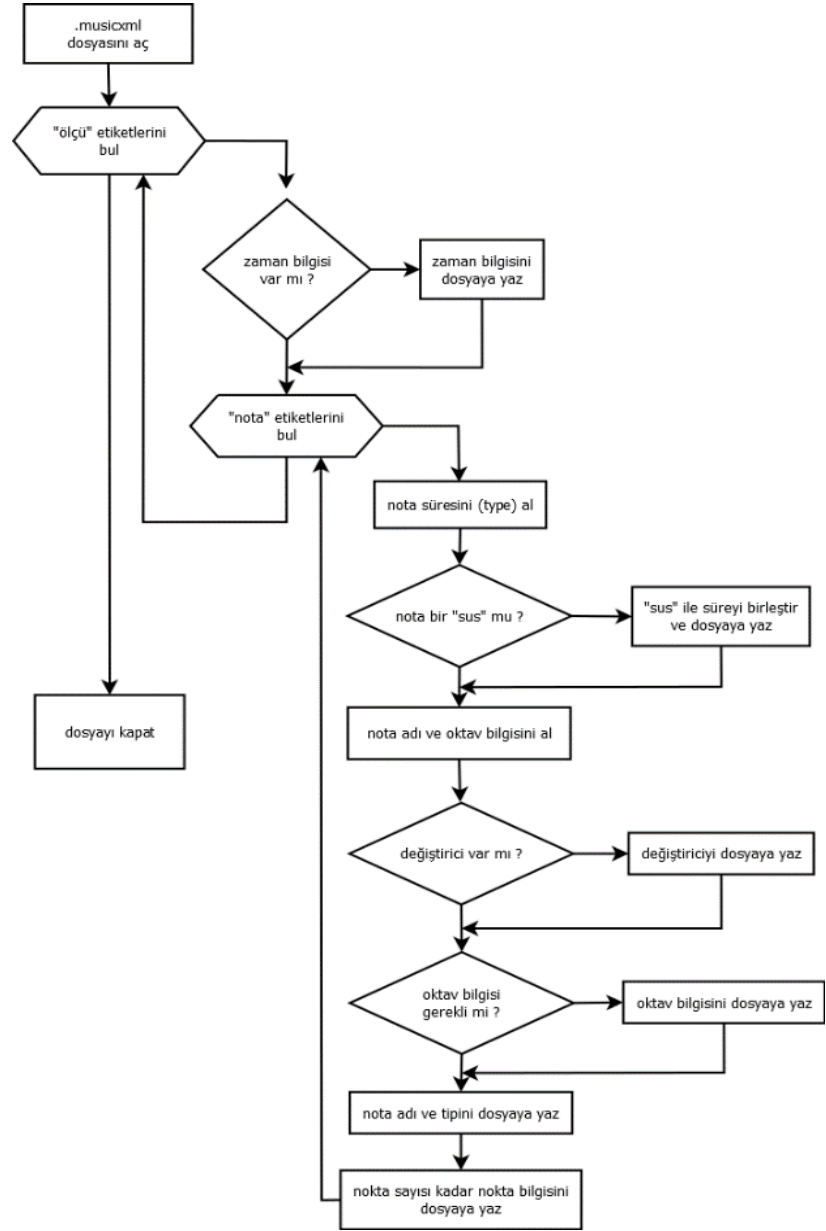
Natürel	n
Diyez	+
Bemol	-

Tablo 6: VBraille Üzerinde Nokta İşaretinin Kodlanması

Nokta	.
-------	---

Nota sürelerini uzatan nokta işaretleri “.___” biçiminde kodlanır. Ses değıştirici işaretler ile benzer şekilde, birden fazla nokta için bu kod aynı sayıda tekrarlanır. Müzik braille alfabesinin kurallarına uygun biçimde, bir notanın hangi oktavda bulunduğunun gösterilmesi gerektiği durumlarda oktav numarası, “o” karakteri, ilgili oktavın numarası ve 2 adet “_” karakteri eklenerek kodlanır. Örneğin, dördüncü oktav “o4__” biçiminde gösterilir.

MusicXML içerisinde “tie” komutu bulunursa satırda geçen “start” fonksiyonu nota öncesine bir adet 4. Hücre noktasını temsilen “^1__” bağ işaretini koymasını sağlar, devamında da uzayacağı notanın öncesine “^2__” işaretini koyarak ikinci sembolü oluşturur. Her ölçü sonunda tüm nota verileri girildikten sonra “|__” işareti ile ölçü kapandığı bilgisi verilir. VBraille, anahtar bilgisini, portede 2. çizgideki sol anahtarı olarak kabul etmektedir.



Şekil 42: MusicXML-VBraille Kodu Dönüşüm Algoritması

5. YÖNTEM

Titreşim uyaranlı arayüz için sol elde 1-2-3. hücre noktaları, sağ elde de 4-5-6. hücre noktalarından oluşan, önceki araçlara benzer bir dizilim kullanılmıştır. Titreşim engelleyici materyaller için HoliBraille (Hugo Nicolau, 2015, 1) isimli çalışmadaki verilerden yararlanılmıştır. Çalışma, birincisi görebilen bireyler ve ikincisi görme engelli bireylerin katıldığı 2 farklı deney içermektedir.

5.1. Birinci Deney

Birinci deney, tasarlanan aracın sahip olduğu titreşim uyaran arayüz aracılığı ile, braille alfabesinin okuyucuya iletilmesi sürecinde, optimum motor titreşim sürelerinin ve hücreleri meydana getiren noktaların bitişik-ayrık olma durumunun, algılama başarımına etkisinin belirlenmesi amacıyla, yaş, ellilik durumu ve cinsiyet gibi özellikler temel alınarak düzenlenmiştir. Bu deneye 20 kadın ve 20 erkek olmak üzere 10'lu yaşlardan 60'lı yaşlara toplam 40 görebilen kişi katılmıştır.

Testler, 16 adet birbirinden farklı braille hücresi içeren 3 adet soru kombinasyonu ile oluşturulmuş ve toplam 8 adet motor süresi olan: 1000 ,500 , 400, 350, 300, 250, 200, 150 milisaniyeleri ile oluşturulmuştur. Testte, her bir sorunun bir önceki soruya benzer olmaması için hücre sıraları değiştirilmiştir. Tüm yaşlar 10-20, 20-30, 30-40, 40-50, 50-60 olarak gruplanmıştır. Test yapılacak kişiye aracın çalışma prensibini anlaması için 2 dakika inceleme ve test etme zamanı verilmiştir.

Testlerdeki hücreler, (nokta sayısı-nokta sayısı) 1-0, 0-1, 1-1, 2-0, 0-2, 2-1, 1-2, 2-1, 2-2, 2-2, 2-2, 3-2, 2-3, 3-2, 2-2, 3-3 olacak şekilde, 16 adet sorulmuştur. Hücreleri oluşturan noktaların bitişik-ayrık olma durumunu incelemek için tüm testler, sol el üzerinde dört adet bitişik, dört adet ayrık ve aynı şekilde sağ el üzerinde de dört adet bitişik, dört adet ayrık olacak şekilde hazırlanmıştır. Soru üzerindeyken geri ve ileri butonları kullanıldığında (tekrar etme ya da atlama) ve yanlış telafuzlarda cevaplar yanlış olarak belirlenmiştir.

VBraille’de uygun motor sürelerini bulmak adına bu çalışmaya benzer araştırma ve projelerde kullanılan testler uygulanarak istatistiksel veriler elde edilmiştir. 40 kişilik görebilen kişiler arasındaki yaş, cinsiyet ve ellilik durumları ve doğru cevap ortalamaları arasındaki ilişkiler, çıkan sonuçların anlamlılığını belirlemek adına, “p” değeri, “<0.05” ise anlamlı, “>0.05” ise anlamsız olarak yorumlanmıştır. İstatistiksel analizler SPSS 22.0 istatistik paket yazılımı kullanılarak yapılmıştır. Testlerin incelenmesindeki ilk aşamada yaş ve motor süreleri verilerinin (değişkenlerin) normal dağılıma uygun olup olmadığını ortaya koymak amacıyla normallik testi olan “One sample Kolmogorov-Smirnov” uygulanmıştır. Bu test sonucunda sadece 150 milisaniye’de değişkenler normal dağılıma uygunluk gösterdiği için “Pearson Korelasyon Testi” uygulanmıştır. Diğer süreler Pearson varsayımlarını karşılamadığı için (değişkenler normal dağılım göstermemekte) “Spearman Korelasyon Testi” uygulanmıştır. Motor süreleri, hem cinsiyet, hem de ellilik durumuna göre incelendiğinde, (iki aritmetik ortalama arasındaki farkın anlamlılığını incelemek) 150 milisaniye normal dağılıma sahip olduğu için “Independent Samples T Test” uygulanmış olup, normal dağılıma uygunluk göstermeyen diğer süreler için de “Mann Whitney U Testi” uygulanmıştır.

5.2. İkinci Deney

İkinci deney, müzik braille bilen, 8 görme engelli kadın ve erkekten oluşan bir gruba uygulanmıştır. Bu deneyin amacı, basit bir müzik braille notasyonunun hem vBraille hem de kabartma yazıya göre ne sürede okunabileceğini belirlemektir. Tüm test boyunca, müzik braille hücreleri görme engelli kişilerden müzikal karşılık olarak seslendirilmeleri istenmiştir.

Test başlangıcında çeşitli müzik braille hücreleri içeren sorular 10 dakika boyunca 8 adet motor süresi olan: 1000 ,500 , 400, 350, 300, 250, 200, 150 milisaniyeleri ile sorulmuş ve kişinin algılamasında en rahat olabileceği hız belirlenmiştir. Belirlenen motor süresi ile 17 adet birbirinden farklı müzik braille hücresi içeren 6 farklı soru hem vBraille’de hem de kabartma yazıda görme engelli kişiye okutulmuş ve okuma hızlarını karşılaştırmak amacıyla süreler kronometre üzerinden not alınmıştır. Hücre hatalı söylendiğinde, tekrar edilmesi istenmiş ve doğru cevap bulunana kadar aynı işlem uygulanmıştır.

6. BULGULAR VE YORUMLAR

İlk deney grubundan alınan sonuçlar, yaş, cinsiyet, ellilik durumu ve hücreleri oluşturan noktaların bitişik-ayrık olma durumu çeşitli tablo ve istatistiksel verilerle gösterilmiş ve incelenmiştir. İkinci deney grubundan alınan sonuçlar ise, vBraille arayüzünün kabartma yazıya göre ne hızda okuma sağladığını belirlemek amacıyla, Perkins hakimiyeti, seçilen motor süresi ve yaş özelliklerini barındıran tablo beraberinde incelenmiştir.

6.1. Birinci Deney İncelemesi

Farklı motor sürelerinde gerçekleştirilen denemelerden alınan doğru cevap sayıları ve ortalamaları, yaş, cinsiyet ve ellilik durumu genel olarak Tablo 7’de görülmektedir.

Tablo 7: Test Sonuçlarının Genel Raporu

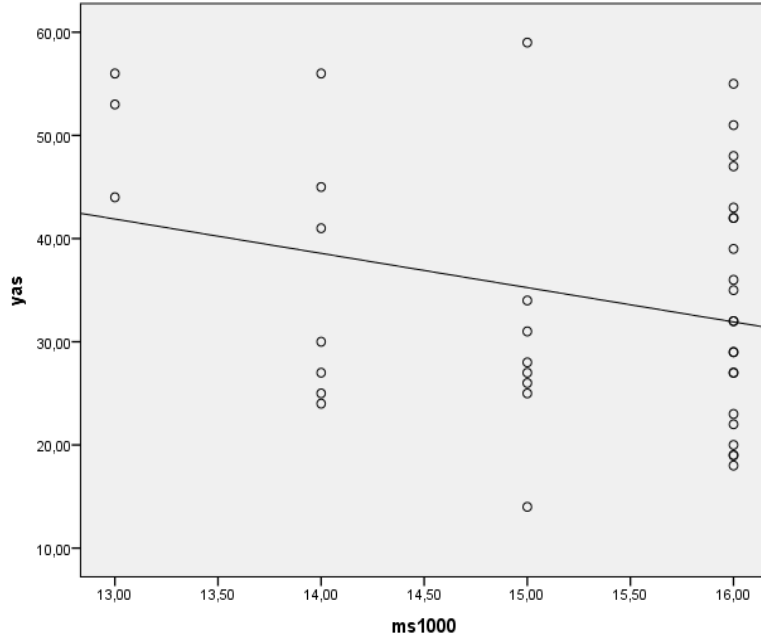
Yaş	Cinsiyet	El ¹⁰	1000	500	400	350	300	250	200	150	Genel Ort.
19	e	r	16	16	16	15	16	16	14	12	15.125
31	e	l	15	16	16	16	16	16	15	11	15.125
39	k	l	16	16	15	15	16	15	14	13	15
29	e	r	16	16	16	14	16	15	15	11	14.875
25	e	r	14	16	14	16	16	15	15	12	14.75
26	e	r	15	16	16	16	16	16	13	10	14.75
22	k	r	16	15	13	15	16	15	14	14	14.75
55	e	r	16	14	16	15	14	15	15	12	14.625
32	k	l	16	15	14	14	16	15	13	12	14.375
27	e	r	16	15	12	16	15	16	13	12	14.375
24	k	r	14	16	16	14	16	14	13	12	14.375
19	e	r	16	16	16	16	15	14	11	9	14.125
32	k	l	16	15	14	13	16	15	11	12	14
34	e	r	15	16	16	13	14	15	11	12	14

¹⁰ Harflerin ayırt edilebilmesi için ingilizce karşılıklarındaki baş harfler kullanılmıştır.

23	e	l	16	15	16	4	16	15	14	14	13.75
43	e	r	16	14	14	16	13	14	12	10	13.625
47	k	r	16	16	13	14	14	16	11	9	13.625
30	k	r	14	16	15	15	14	16	10	9	13.625
20	k	r	16	16	13	16	14	13	12	8	13.5
28	k	l	15	15	15	16	16	15	4	12	13.5
35	e	r	16	16	15	12	15	12	13	8	13.375
25	k	l	15	14	15	15	16	12	11	9	13.375
27	k	r	14	14	16	13	15	11	11	10	13
14	k	r	15	15	13	14	13	13	11	9	12.875
41	k	r	14	13	16	16	14	13	7	10	12.875
18	k	l	16	16	13	15	14	12	11	6	12.875
29	e	r	16	13	15	14	13	11	11	8	12.625
42	k	r	16	13	15	14	11	10	12	10	12.625
51	k	r	16	13	13	13	13	11	10	11	12.5
53	e	r	13	16	15	15	14	10	11	5	12.375
56	k	r	13	13	13	14	14	10	11	9	12.125
36	e	r	16	16	13	13	11	9	13	6	12.125
27	k	r	15	16	15	13	13	10	8	7	12.125
42	k	r	16	15	14	12	10	8	8	4	10.875
48	e	r	16	14	15	11	9	8	7	6	10.75
27	e	r	16	12	14	11	12	7	7	5	10.5
44	e	r	13	11	11	14	12	8	7	7	10.375
45	e	l	14	12	14	11	9	8	6	6	10
59	e	r	15	8	7	5	4	6	11	11	8.375
56	k	r	14	9	7	6	6	6	6	5	7.375
			15.2	14.5	14.1	13.5	13.6	12.4	11.1	9.45	

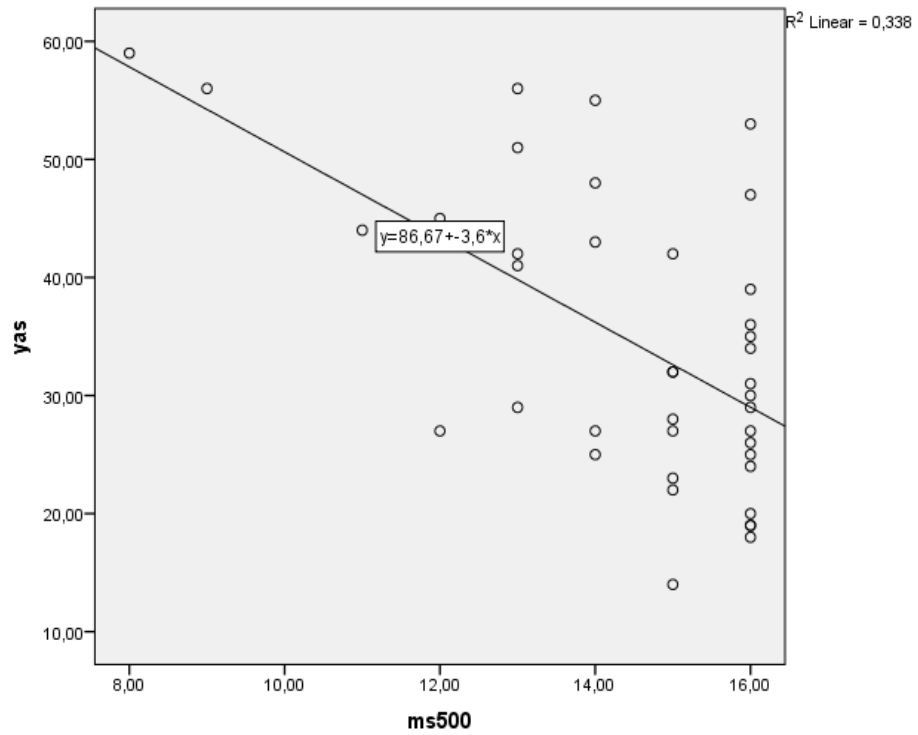
6.1.1. Motor Sürelerindeki Doğru Cevaplarının Yaşa Göre İncelenmesi

Tabloda koyu renkli hücreler doğru cevap sayılarının, kişilere (yatay) ve motor sürelerine (dikey) göre ortalamalarını gösmektedir. Her motor süresinde farklı yaşlarda verilen cevaplar istatistiksel olarak incelenmiştir.



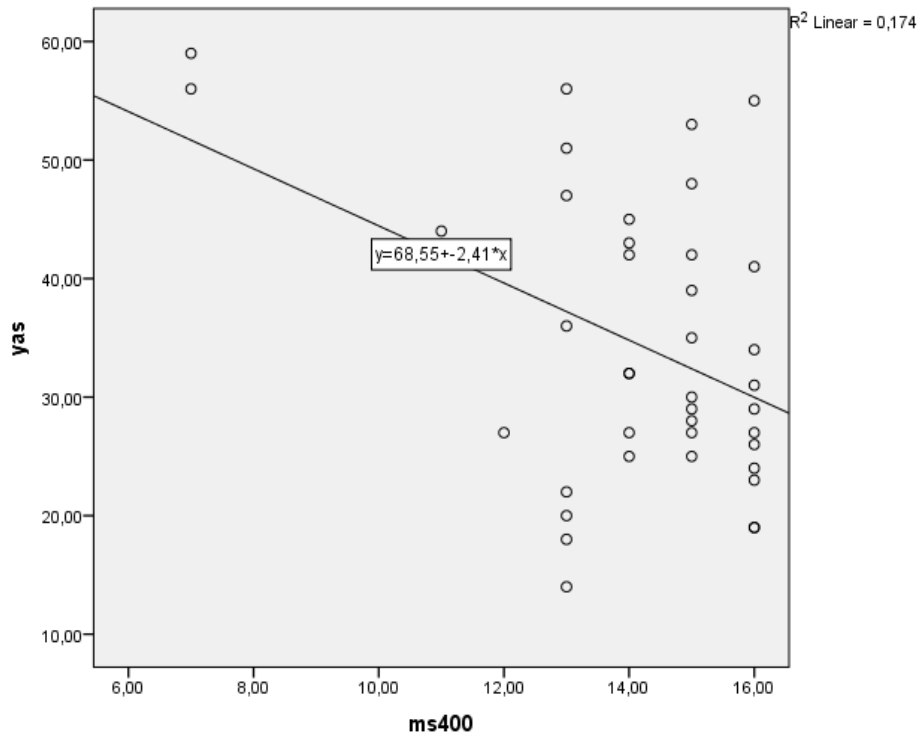
Şekil 43: 1000ms'deki Doğru Cevapların Saçılım Grafiği

1000 milisaniye'lik motor süresi için, doğru cevap sayıları 13-16 arasında değişmektedir. Dolayısı ile hata aralığı en fazla 3'tür. Tüm yaşlardaki doğru ortalaması da Tablo 7'de görüldüğü gibi 15,2'dir. 50-60 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 13, en yüksek de 16'dır. 40-50 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 13, en yüksek de 16'dır. 30-40 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 15, en yüksek de 16'dır. 20-30 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 14, en yüksek de 16'dır. 10-20 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 15, en yüksek de 16'dır. Genel olarak, araştırmaya katılan tüm bireylerin 1000 milisaniye üzerinde verdiği doğru yanıt sayıları fazladır. Aradaki ilişki istatistiksel olarak incelendiğinde, araştırmaya katılan tüm bireylerin yaşları arttıkça, 1000 milisaniye üzerinde verdiği doğru yanıt sayıları değişiklik göstermektedir; bu sebeple anlamlı değildir ($p=0,277$, $r=-0,176$) ($p>0,05$).



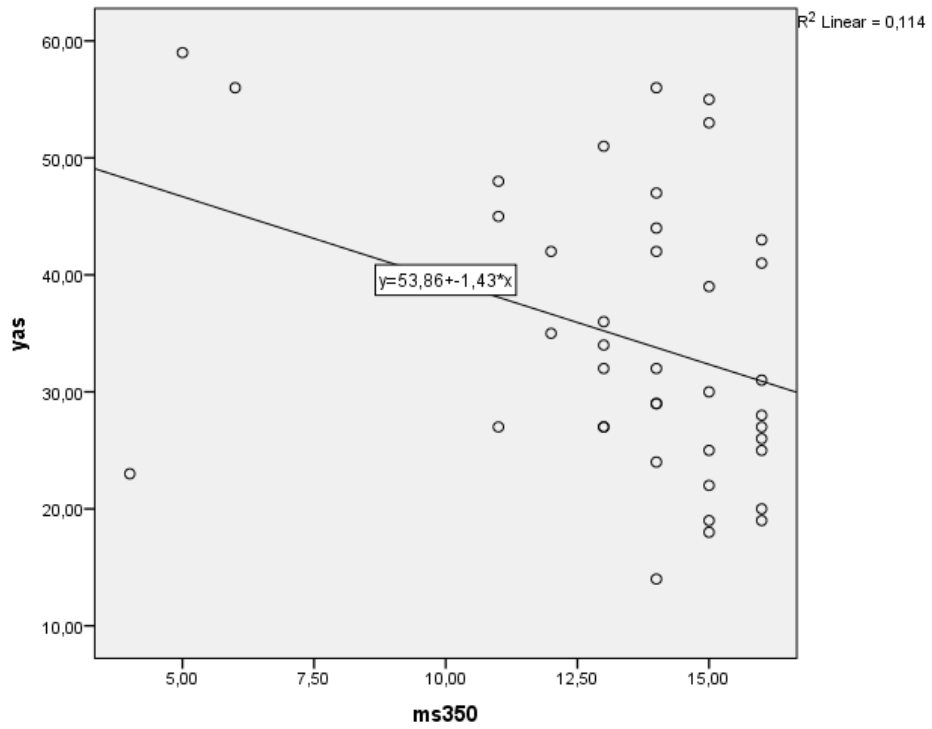
Şekil 44: 500ms'deki Doğru Cevapların Saçılım Grafiği

500 milisaniye'lik motor süresi için, doğru cevap sayıları 8-16 arasında değişmektedir. Hata aralığı en fazla 8'dir. Tüm yaşlardaki doğru ortalaması Tablo 7'de görüldüğü gibi 14,5'dir . 50-60 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 8, en yüksek de 16'dır. 40-50 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 11, en yüksek de 16'dır. 30-40 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 15, en yüksek de 16'dır. 20-30 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 12, en yüksek de 16'dır. 10-20 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 15, en yüksek de 16'dır. Aradaki ilişki istatistiksel olarak incelendiğinde, katılımcıların 500 milisaniye üzerinde verdiği doğru yanıt sayıları azalmaktadır ve aradaki ilişki istatistiksel olarak anlamlıdır ($p=0,002$, $r=-0,477$) ($p<0,05$).



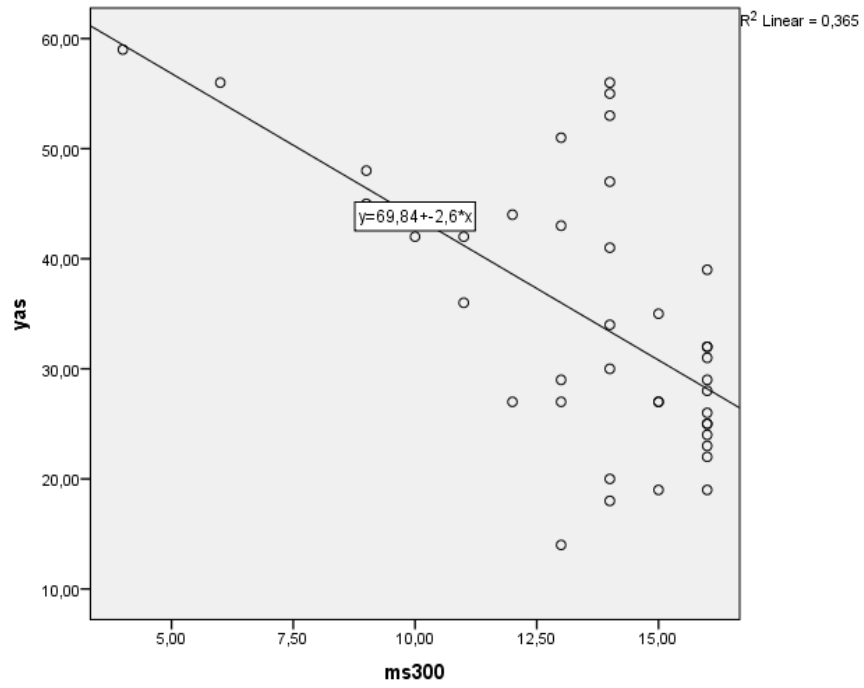
Şekil 45: 400ms'deki Doğru Cevapların Saçılım Grafiği

400 milisaniye'lik motor süresi için, doğru cevap sayıları 6-16 arasında değişmektedir. Hata aralığı en fazla 10'dur. Tüm yaşlardaki doğru ortalaması Tablo 7'de görüldüğü gibi 14,1'dir. 50-60 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 7, en yüksek de 16'dır. 40-50 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 11, en yüksek de 16'dır. 30-40 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 13, en yüksek de 16'dır. 20-30 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 12, en yüksek de 16'dır. 10-20 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 13, en yüksek de 16'dır. Araştırmaya katılan tüm bireylerin yaşları arttıkça, 400 milisaniye üzerinde verdiği doğru yanıt sayıları değişiklik göstermektedir; aradaki ilişki istatistiksel olarak anlamlı değildir ($p=0,110$, $r=-0,256$) ($p>0,05$).



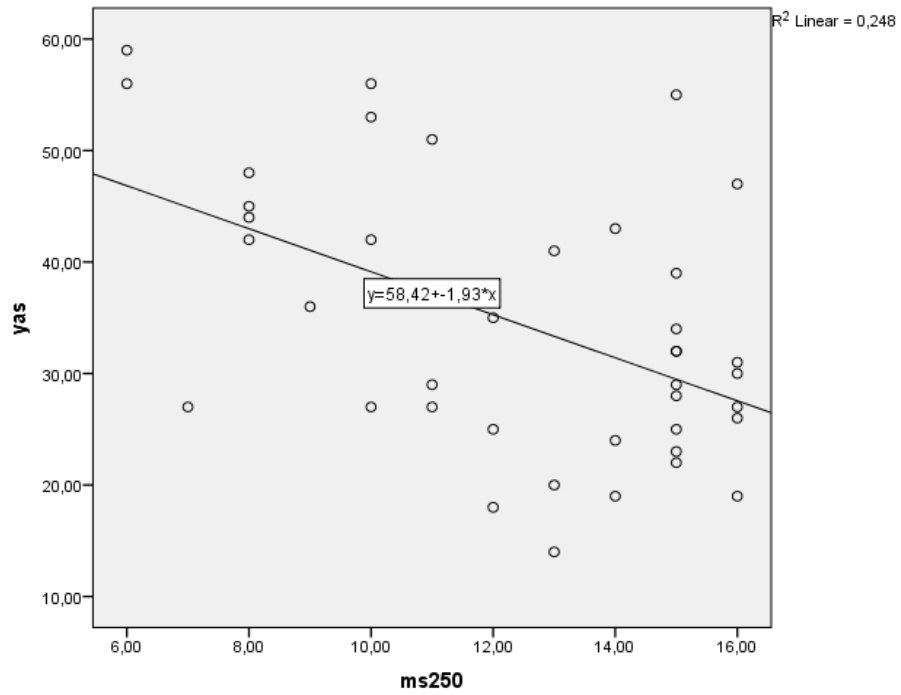
Şekil 46: 350ms'deki Doğru Cevapların Saçılım Grafiği

350 milisaniye'lik motor süresi için, doğru cevap sayıları 4-16 arasında değişmektedir. Hata aralığı 12'dir. Tüm yaşlardaki doğru ortalaması Tablo 7'de görüldüğü gibi 13,5'dir. 50-60 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 5, en yüksek de 15'dir. 40-50 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 11 en yüksek de 16'dır. 30-40 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 12, en yüksek de 16'dır. 20-30 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 4, en yüksek de 16'dır. 10-20 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 14, en yüksek de 16'dır. Aradaki ilişki istatistiksel olarak incelendiğinde, araştırmaya katılan tüm bireylerin yaşları arttıkça, 350 milisaniye verdiği doğru yanıt sayıları azalmaktadır; aradaki ilişki istatistiksel olarak anlamlıdır ($p=0,024$, $r=-0,357$) ($p<0,05$).



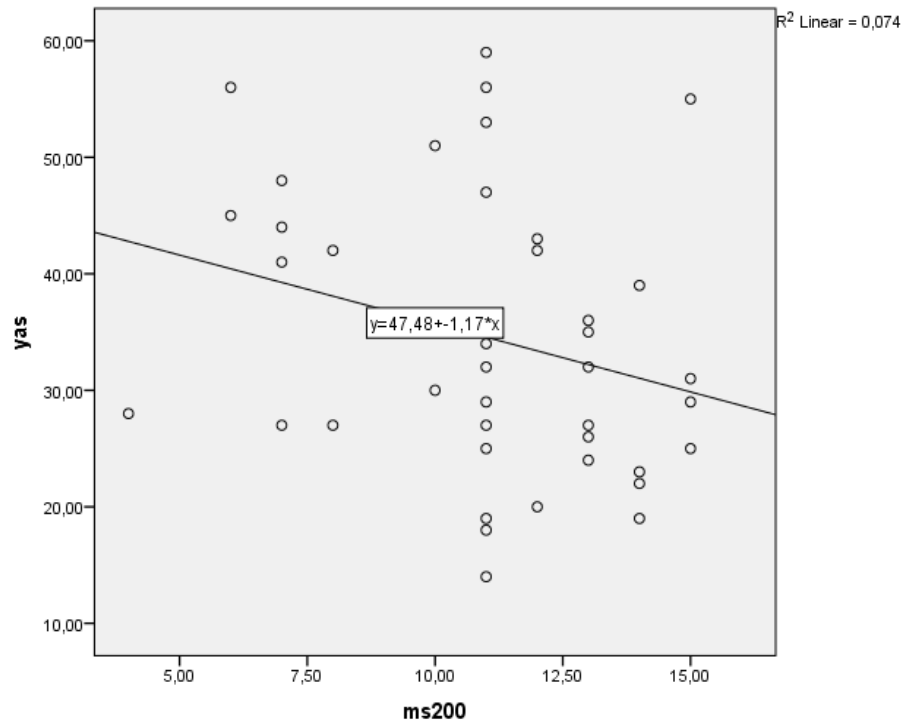
Şekil 47: 300ms'deki Doğru Cevapların Saçılım Grafiği

300 milisaniye'lik motor süresi için, doğru cevap sayıları 4-16 arasında değişmektedir. Hata aralığı en fazla 12'dir. Tüm yaşlardaki doğru ortalaması Tablo 7'de görüldüğü gibi 13,6'dır. 50-60 yaş grubunda katılan 6 katılımcının verdiği doğru cevaplar, en düşük 4, en yüksek de 15'dir. 40-50 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 9, en yüksek de 15'dir. 30-40 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 11, en yüksek de 16'dır. 20-30 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 12, en yüksek de 16'dır. 10-20 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 13, en yüksek de 16'dır. Genel olarak, araştırmaya katılan tüm bireylerin yaşları arttıkça, 300 milisaniye üzerinde verdiği doğru yanıt sayıları azalmaktadır; aradaki ilişki istatistiksel olarak anlamlıdır ($p=0,0$, $r=-0,545$) ($p<0,05$).



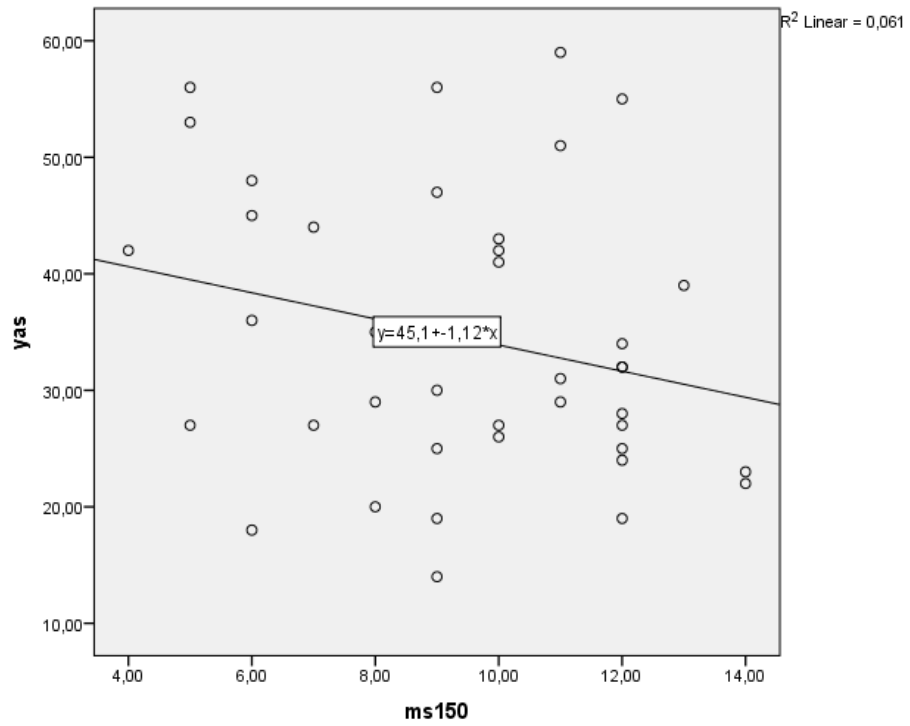
Şekil 48: 250ms'deki Doğru Cevapların Saçılım Grafiği

250 milisaniye'lik motor süresi için, doğru cevap sayıları 6-16 arasında değişmektedir. Hata aralığı en fazla 10'dur. Tüm yaşlardaki doğru ortalaması Tablo 7'de görüldüğü gibi 12,4'tür. 50-60 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 6, en yüksek de 15'dir. 40-50 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 9, en yüksek de 15'dir. 30-40 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 11, en yüksek de 16'dır. 20-30 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 12, en yüksek de 16'dır. 10-20 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 13, en yüksek de 16'dır. Araştırmaya katılan tüm bireylerin yaşları arttıkça, 250 milisaniye üzerinde verdiği doğru yanıt sayıları azalmaktadır; aradaki ilişki istatistiksel olarak anlamlıdır ($p=0,008$, $r=-0,416$) ($p<0,05$).



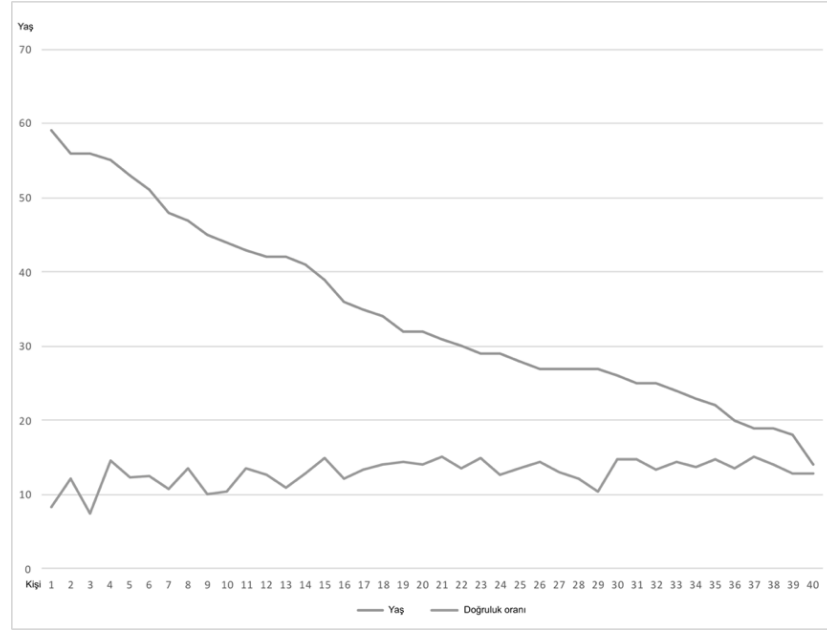
Şekil 49: 200ms'deki Doğru Cevapların Saçılım Grafiği

200 milisaniye'lik motor süresi için, doğru cevap sayıları 4-15 arasında değişmektedir. Hata aralığı en fazla 11'dir. Tüm yaşlardaki doğru ortalaması Tablo 7'de görüldüğü gibi 11,1'dir. 50-60 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 6, en yüksek de 15'dir. 40-50 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 6, en yüksek de 11'dir. 30-40 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 10, en yüksek de 14'tür. 20-30 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 4, en yüksek de 15'dir. 10-20 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 11, en yüksek de 14'tür. Araştırmaya katılan tüm bireylerin yaşları arttıkça, 200 milisaniye üzerinde verdiği doğru yanıt sayıları değişiklik göstermektedir; aradaki ilişki istatistiksel olarak anlamlı değildir ($p=0,054$, $r=-0,306$) ($p>0,05$).



Şekil 50: 150ms'deki Doğru Cevapların Saçılım Grafiği

150 milisaniye'lik motor süresi için, doğru cevap sayıları 4-14 arasında değişmektedir. Hata aralığı en fazla 10'dur. Tüm yaşlardaki doğru ortalaması Tablo 7'de görüldüğü gibi 9,45'tir. 50-60 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 5, en yüksek de 12'dir. 40-50 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 4, en yüksek de 10'dur. 30-40 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 6, en yüksek de 13'tür. 20-30 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 5, en yüksek de 14'tür. 10-20 yaş grubundaki katılımcıların verdiği doğru cevaplar, en düşük 6, en yüksek de 12'dir. Araştırmaya katılan tüm bireylerin yaşları arttıkça, 150 milisaniye üzerinde verdiği doğru yanıt sayıları değişiklik göstermektedir; aradaki ilişki istatistiksel olarak anlamlı değildir ($p=0,144$, $r=-0,235$) ($p>0,05$).



Şekil 51: Tüm Motor Sürelerinde Yaş-Doğru Oranı Grafiği

Şekil 51’de yaşlar ve doğru cevapları, kişi sayısı (yatay), yaş ve doğru sayısı (dikey) olarak gösterilmiştir. Grafikte, yaşa göre doğru cevap sayısı arasında ters bir orantı olduğu, doğru sayısının yaş azaldıkça arttığı görülmektedir.

6.1.2. Motor Sürelerindeki Doğru Cevaplarının Cinsiyete Göre İncelenmesi

Motor sürelerinde verilen doğru yanıtları, araştırmaya katılan bireylerin cinsiyetlerine göre incelendiğinde, aradaki farklılıklar çok düşük olmakla beraber istatistiksel olarak da anlamlı değildir ($p>0,05$).

Tablo 8: Doğru Cevaplarının Cinsiyetlere Göre İstatistik Tablosu¹¹

Süre	Erkek		Kadın		p
	Ortalama+SS	Standart Sapma	Ortalama+SS	Standart Sapma	
1000 ms	15,3	1	15,1	0,9	0,55*
500 ms	14,4	2,2	14,5	1,7	0,799*
400 ms	14,3	2,2	13,9	1,9	0,174*
350 ms	13,1	3,4	13,8	2,1	0,869*
300 ms	13,3	3,1	13,8	2,5	0,637*
250 ms	12,3	3,5	12,5	2,7	0,913*

¹¹ *Mann Whitney U Testi, **Independent Samples T Testi

200 ms	11,7	2,9	10,4	2,6	0,085*
150 ms	9,3	2,7	9,5	2,6	0,817**

6.1.3. Motor Sürelerindeki Doğru Cevaplarının Sağ ve Sol Ele Göre İncelenmesi

Deney yapılan 40 kişiden 31 kişi sağlak ve 9 kişi de solaktır. Motor sürelerinde verilen doğru yanıtları, deneye katılan bireylerin ellilik durumuna göre incelenmiştir.

Tablo 9: Doğru Cevaplarının Sağ ve Sol Ele Göre İstatistik Tablosu¹²

Süre	Sol El		Sağ El		p
	Ortalama	Standart Sapma	Ortalama	Standart Sapma	
1000 ms	15,4	0,7	15,1	1	0,629*
500 ms	14,8	1,2	14,3	2,1	0,812*
400 ms	14,6	1	13,9	2,3	0,607*
350 ms	13,2	3,8	13,5	2,6	0,83*
300 ms	15	2,3	13,1	2,8	0,012*
250 ms	13,6	2,5	12	3,2	0,19*
200 ms	11	3,7	11	2,5	0,779*
150 ms	10,5	2,9	9,1	2,5	0,162**

Motor sürelerinde verilen doğru yanıtları, deneye katılan bireylerin ellerine göre incelendiğinde aradaki farklılığın yalnızca 300 milisaniye’de istatistiksel olarak anlamlı olduğu görülmektedir ($p<0,05$). Aynı zamanda Tablo 9’a göre 40 kişilik erkek-kadın karışık testinde sol ve sağ el karşılaştırıldığında sol el üzerinden verilen cevapların sağ ele göre daha yüksek doğru oranına sahip olduğu görülmektedir.

6.1.4. Testlerde Hücre Noktalarındaki Bitişik-Ayrık Durumunun İncelenmesi

Bu bölümde, motor titreşim sürelerinin ve hücreleri meydana getiren noktaların bitişik-ayrık olma durumuna göre algılama başarımına etkisi incelenmiştir. Bu etkilerin belirlenmesi amacıyla, 1-2, 2-3, 4-5, 5-6 noktalarını içeren hücreler “bitişik”, 1-3, 4-6 noktalarını içeren hücreler de “ayrık” olarak tanımlanmıştır. Testlerde sol el için 4 bitişik 4 ayrık, aynı şekilde sağ el için de 4 bitişik 4 ayrık hücre noktalarını içeren sorular sorulmuştur.

¹² *Mann Whitney U Testi, **Independent Samples T Testi

Tablo 10: Noktaların Bitişik-Ayrık Durumunu İçeren Deney Soruları

Test Numarası	Braille Yazımı	Toplam Hücre Sayısı	Yazı Yazımı ¹³	Sol Bitişik	Sol Ayrık	Sağ Bitişik	Sağ Ayrık
1	⠠	1	1 Up				
2	⠠	1	Oct. 4				
3	⠠	2	8th A				
4	⠠	2	2	x			
5	⠠	2	Oct 5				x
6	⠠	3	Half D		x		
7	⠠	3	8th B			x	
8	⠠	3	8th E	x			
9	⠠	4	8th F	x		x	
10	⠠	4	Whole D		x	x	
11	⠠	4	Whole A	x			x
12	⠠	5	Whole G			x	
13	⠠	5	Whole C		x		
14	⠠	5	Whole E				x
15	⠠	5	8th Rest		x		x
16	⠠	6	Whole F				

Tablo 10’da görülen test soruları, 1 ile 6 noktaya denk gelen, vBraille’in içerisinde bulunan müzik braille hücreleri şekilde hazırlanmıştır. Her bir kategoride bitişik-ayrık durumu bulunan sorular “x” ile işaretlenmiştir. Bu durumda kategori için 4 tane soru temel alınmıştır.

¹³ Yazı yazımı bölümündeki ifadeler, müzik braille İngilizce kaynaklarında olduğu gibi kullanılmıştır.

Tablo 11: Noktaların Bitişik-Ayrık Durumunu İçeren Deney Ortalamaları

	Sol El			Sağ El		
Süre	Bitişik Ortalaması	Ayrık Ortalaması	Sol El Değişim ¹⁴	Bitişik Ortalaması	Ayrık Ortalaması	Sağ El Değişim ¹⁵
1000 ms	3,9	3,825	%1.92	3,675	3,925	%6.36
500 ms	3,8	3,725	%1.97	3,775	3,775	%0
400 ms	3,825	3,675	%3.92	3,725	3,45	%7.38
350 ms	3,6	3,8	%5.26	3,525	3,625	%2.75
300 ms	3,475	3,625	%4.13	3,375	3,4	%0.73
250 ms	3,5	3,425	%2.14	3,35	3	%10.44
200 ms	3,25	3,25	%0	2,775	2,8	%0.89
150 ms	3,2	2,925	%8.59	2,175	2,3	%5.43

Yapılan testlerdeki tüm cevaplar bitişik-ayrık olma durumuna göre tekrar incelenmiş ve Ek 1 ve Ek 2’deki cevap tabloları ortaya çıkmıştır. Tablo 11’de ortalama olarak yazılan değerler, Ek 1 ve Ek 2’de dikey sıralanan cevaplar üzerinden alınmıştır. Sağ ve sol el bitişik-ayrık ortalamaları arasında farklar vardır. Bu farkları belirtmek amacıyla bitişik ve ayrık ortalamalarının arasındaki değişim oranları hesaplanmış ve tabloya yazılmıştır. Tablo 11’e bakıldığında 1000 milisaniye’de sol el değişim oranı %1.92, sağ el değişim oranı %6.36’dır. 500 milisaniye’de sol el değişim oranı %1.97, sağ el %0’dır. 400 milisaniye’de sol el değişim oranı %3.92, sağ el %7.38’dir. 350 milisaniye’de sol el değişim oranı %5.26, sağ el %2.75’dir. 300 milisaniye’de sol el değişim oranı %4.13, sağ el %0.73’tür. 250 milisaniye’de sol el değişim oranı %2.14’tür, sağ el %10.44’tür. 200 milisaniye’de sol el bitişik-ayrık değişim oranı %0, sağ el %0.89’dur. 150 milisaniye’de sol el değişim oranı %8.59, sağ el %5.43’tür. Genel olarak bakıldığında, sol elde 1000-500-400-250-150 ms’de bitişik ortalaması ayrık ortalamasına göre yüksek, 350-300 ms’de ise düşüktür. Sağ elde 400-350-250 ms’de bitişik ortalaması ayrık ortalamasına göre yüksek, 1000-300-200-150 ms’de düşüktür. 200 ve 500 ms’de her iki el için bitişik-ayrık ortalamaları arasındaki değişim oranı %0 olmakla beraber hiçbir fark çıkmamıştır.

¹⁴ Sol el bitişik ve ayrık ortalamalarının arasındaki değişim oranı.

¹⁵ Sağ el bitişik ve ayrık ortalamalarının arasındaki değişim oranı.

6.2. İkinci Deney İncelemesi

İkinci deneyde, 17 hücre barındıran 6 farklı müzik braille testleri ilk önce vBraille sonra da kabartma yazı ile okutulmuştur. Sonuçlar, Perkins hakimiyeti ve süre farkları bakımından incelenmiştir. Birçok görme engellinin alışkın olduğu Perkins parmak hizalaması, vBraille titreşim motoru diziliminde kullanılması sebebiyle bu test için önem taşımaktadır. Bu nedenle, deneye katılan bireylerin Perkins kullanım sıklığı: güncel “iyi”, az “orta” ve çok az “kötü olarak belirlenmiştir. Kabartma yazıda sadece süre özelliği baz alınmıştır.

Tablo 12: İkinci Deney Grubu Vbraille Ve Kabartma Yazı Sonuç Tablosu

Yaş	Motor Süresi	Perkins	vBraille		Kabartma yazı		vBraille / Kabartma yazı
35	200ms	iyi	Test	Süre	Test	Süre	Süre Farkı
			1	1:10	1	0:40	+0:30
			2	1:00	2	0:25	+0:35
			3	0:44	3	0:23	+0:21
			4	0:46	4	0:20	+0:26
			5	0:35	5	0:19	+0:16
			6	0:42	6	0:26	+0:18
57	500ms	kötü	Test	Süre	Test	Süre	Süre Farkı
			1	2:42	1	0:35	+2:07
			2	2:10	2	0:22	+1:48
			3	1:58	3	0:17	+1:41
			4	1:38	4	0:18	+1:20
			5	1:35	5	0:15	+1:20
			6	1:38	6	0:17	+1:21
57	300ms	kötü	Test	Süre	Test	Süre	Süre Farkı
			1	3:00	1	0:34	+2:26
			2	2:02	2	0:26	+1:36

			3	1:52	3	0:23	+1:29
			4	1:16	4	0:22	+0:54
			5	1:38	5	0:17	+1:21
			6	1:12	6	0:23	+0:49
25	200ms	orta	Test	Süre	Test	Süre	Süre Farkı
			1	1:32	1	0:20	+1:12
			2	1:05	2	0:15	+0:50
			3	1:10	3	0:17	+0:53
			4	0:40	4	0:16	+0:24
			5	0:48	5	0:12	+0:36
			6	1:00	6	0:19	+0:41
42	250ms	kötü	Test	Süre	Test	Süre	Süre Farkı
			1	1:34	1	0:23	+1:11
			2	2:14	2	0:17	+1:57
			3	1:25	3	0:18	+1:07
			4	1:15	4	0:20	+0:55
			5	1:11	5	0:19	+0:52
			6	2:00	6	0:17	+1:43
47	250ms	iyi	Test	Süre	Test	Süre	Süre Farkı
			1	1:32	1	0:17	+1:15
			2	0:48	2	0:15	+0:33
			3	1:00	3	0:16	+0:44
			4	0:45	4	0:19	+0:26
			5	0:50	5	0:16	+0:34
			6	1:06	6	0:26	+0:40
32	200ms	iyi	Test	Süre	Test	Süre	Fark
			1	1:40	1	0:23	+1:17
			2	1:00	2	0:27	+0:37

			3	0:44		3	0:34	+0:10
			4	0:40		4	0:33	+0:07
			5	0:28		5	0:18	+0:10
			6	0:34		6	0:29	+0:05
32	300ms	orta	Test	Süre		Test	Süre	Fark
			1	1:38		1	0:39	+1:58
			2	1:30		2	0:35	+1:05
			3	0:46		3	0:25	+0:21
			4	0:37		4	0:20	+0:17
			5	0:46		5	0:18	+0:28
			6	0:50		6	0:23	+0:27

Tablo 12’de 8 görme engelli kişinin test özelliklerine bakıldığında yaş aralığı 25-57, motor süresi en düşük 200 en yüksek de 500ms olarak görülmektedir. VBraille ile kabartma yazı arasındaki en uzun süre farkı “2 dakika 26 saniye” ile 57 yaşında, 300ms motor süresinde ve Perkins’i çok az kullanmış bir kişide görülmektedir. Bir diğer taraftan, en kısa süre farkı “5 saniye” ile 32 yaşında, 200ms motor süresinde ve Perkins’i güncel olarak kullanan bir kişide görülmektedir. Alınan sonuçlar incelendiğinde, vBraille ile yapılan okumalardan alınan süreler kabartma yazı sürelerine göre çeşitlik göstermiş, hatta yakın süreler ortaya çıkmıştır.

7. SONUÇ

Araştırma bünyesinde, müziksel okumaya yönelik, kendin-yap felsefesine dayanan titreşim uyaranlı bir arayüz tasarlanmış, birinci deney grubu, görebilen 40 kişi üzerinde motor süre testleri yapılmış ve tüm motor sürelerinde verilen doğru cevapları yaş, cinsiyet, ellilik durumu ve hücrelerin bitişik ayrık olma durumu üzerinden incelenmiştir. İkinci deney grubunda ise 6 farklı müzik braille testleri ilk olarak vBraille, sonrasında da kabartma yazı ile okutulmuştur. Sonuçlar, Perkins hakimiyeti ve süre farkları bakımından incelenmiştir.

Tasarım, önceki araçlarda bulunan Bluetooth, GPS, sesli geribildirim gibi ek özellikler gerektirmemektedir. Bu sebeple, istenilen fonksiyonları yerine getirilebilecek parçalarla, arayüz “170 TL” civarında tasarlanabilmektedir. Tasarım, en düşük maliyetli ticari ürünlerle bile karşılaştırıldığında farkın 35 kat olması, oranın çok fazla olduğunu net bir şekilde göstermektedir.

Prototip üzerindeki motorların titreşim süreleri ile elde edinilen verilerde karşılaşılan en büyük sorunlardan birincisi, düşük fiyatlı motorların aynı şiddetle titreşim üretememesi olmuştur. Karşılaşılan bir diğer sorun ise, gövdeden uygun biçimde izole edilmemiş motorların meydana getirdiği titreşimlerin gövdeye yayılabilmesi ve algılamada karışıklığa yol açmasıdır.

Yapılan çalışmada, 1000 ms’den 300 ms’lik motor sürelerine kadar katılımcıların doğru cevap oranının yüksek olduğu, ancak, 250 ms’den itibaren yanlış cevap oranının arttığı görülmektedir. 300 ms, vBraille ile hücre okumada kullanılacak en optimum hız olduğu belirlenmiştir. 1000 ms için bir hücrenin tamamlanması 6 sn sürmekte, 150 ms’de ise 0,9 sn sürmektedir. Hücreleri hızlı okumak adına vBraille ile daha fazla pratik yapıldığında, doğruluk ortalamasının arttırılabileceği sonucuna varılmıştır.

Titreşim uyaranlı arayüzün motor süreleri ve yaş arasındaki ilişki incelendiğinde, 1000 ile 300 ms arasında yaş ile okuma hızı arasında bir düşüş neredeyse bulunmamaktadır. Fakat 250 ms’den itibaren 150 ms’ye kadar yaşlar arttıkça doğru

cevapları düşüş göstermektedir. Tüm motor süreleri ve yaşlarda genel ortalamaya bakıldığında, yaş arttıkça doğru cevap sayılarının azaldığı tespit edilmiştir.

Doğru cevap ortalamaları cinsiyetlere göre incelendiğinde, kadın ve erkeklerin aynı sorular üzerinde verdiği cevap ortalamaları birbirlerine çok yakındır. Sadece “200 ms” de kadınlar, erkeklere göre daha yüksek cevap vermiş ve “istatistiksel olarak anlamlı” ifadesine yakın sonuç elde edilmiştir. Fakat bu sonuç, genel tabloda sadece bir motor süresi üzerinden elde edildiği için, başarı oranlarının cinsiyete bağlı olarak değişiklik göstermediği anlaşılmıştır.

Katılımcıların ellilik durumu, motor süreleri üzerinden verilen cevaplar ile incelendiğinde, tüm hızlarda sol el sağ ele göre daha yüksek doğru cevaplara sahiptir. 1, 2 ve 3. Noktaları barındıran hücrelerin sol el üzerinden başlaması, konsantrasyonun sol elde daha fazla olması gibi bir durumu beraberinde getirmiştir. Bu sonuç beraberinde, titreşim dokunsal uyarıların algılanabilirlik oranı, bireyin ellilik durumuna bağlı olarak değişiklik göstermediği sonucu çıkarılmaktadır.

Hücre noktalarının bitişik-ayrık olma durumu, hücre algılama adına çeşitli sonuçlar vermiştir. Sadece “200 ve 500 ms” motor sürelerinde sağ ve sol el için değişim oranı “0”dır. Diğer tüm sürelerdeki bitişik-ayrık ortalamalarının değişim oranı çeşitlilik göstermekte fakat yüksek farklılıklar çıkmamaktadır. Bu veriler beraberinde, hücre noktalarının bitişik-ayrık olma durumunun parmak algılama başarımına etkisinde, “200 ve 500 ms” en ideal motor süreleri olarak belirlenmiştir.

Cihaz, görme engelli bireyler tarafından test edildikten sonra alınan sonuçlarda, güncel olarak Perkins Braille kullanan kişilerin vBraille’e daha çok hakim olduğu ve hızlı okuma yapabildiği görülmektedir. Ek 5 bölümündeki yorumlara bakıldığında, ilk müzik eğitimi için kullanılması ve notaya hızlı bir şekilde erişim kolaylığı sağlaması bakımından tercih edilebileceği yorumlar arasında sıklıkla söylenmiştir. Özellikle Perkins braille kullanan kişilere büyük oranda fayda sağlayacağı ve bu cihazdan yararlanarak kişinin kendi kabartma notasını kolaylıkla okuyabileceği sonucuna varılmıştır.

Gelinen son aşamada, uygun maliyetli tasarlanan bir araç ile ölçü rakamları, tonalite, nota değerleri-süreleri, sus işaretleri, oktav numaraları, ses değiştirici işaretler, ölçü işaretleri ve noktalar gibi temel müziksel bilgiler farklı kişilerde çeşitli sürelerde yüksek doğruluk oranları ile okunabilmektedir. Fakat, titreşim motorları arasındaki

mesafenin rastgele seçilmesi, titreşim motorlarının voltaj ve titreşim değerlerinin tutarsız olması ve gövde üzerinde motorların meydana getiren titreşimleri önleyen yay ve süngerlerin uygulanması gibi yöntemlerin hücre algılama oranında azalmaya neden olduğu sonucu çıkarılmaktadır. Tasarım ile yapılan testlerde, hücre algılamadaki doğru oranlarının yaş arttıkça azaldığı, ancak, cinsiyet ve ellilik durumuna göre bir değişiklik göstermediği belirlenmiştir. Aynı zamanda, hücre noktalarının bitişik-ayrık olma durumunun hücre algılamada bir sorun yaratmadığı bu tez çalışması sonucunda elde edilmiştir.

KAYNAKÇA

- AFB . “What is Braille” . <https://www.afb.org/blindness-and-low-vision/braille/what-braille> [23.08.2020]
- Durmaz, Aziz. “XML 'in tanımı ve Kullanımı Hakkında Bilgi” <http://www.csharpnedir.com/articles/read/?id=38&title=XML%20%27in%20tan%C4%B1m%C4%B1%20ve%20Kullan%C4%B1m%C4%B1%20Hakk%C4%B1nda%20Bilgi> [14.3.2020]
- Benz, Brian, John Durant. **XML Programming Bible** (Wiley Publishing, Inc., Indianapolis, Indiana. 2003): 3.
- Google Patents. “Inaudible midi interpretation device”. <https://patents.google.com/patent/US20070166693A1/en?q=Inaudible+midi+interpretation+device&oq=Inaudible+midi+interpretation+device> [16.07.2020]
- _____. “Portable Device With Virtual Tactile Keyboard and Refreshable Braille Display”. <https://patents.google.com/patent/US9965974B2/en?q=9965974> [16.07.2020]
- _____. “Touch sense device for the visually impaired.” <https://patents.google.com/patent/US20120295232A1/en> [16.07.2020]
- Flusoft. “Orbit Reader 20 - Braille notebook” <http://www.flusoft.de/produkte/orbit/index.html> [16.07.2020]
- Hims. “About Us”. <https://www.hims-inc.com/about-us/> [20.05.2020]
- _____. “Smart Beetle”. <https://www.hims-inc.com/product/smart-beetle/> [20.05.2020]
- Humanware. “Mountbatten Whisperer Writer”, <https://store.humanware.com/hus/mountbatten-whisperer-writer.html> [6.05.2020]
- _____. “Brilliant BI 14 Braille Display”, <https://store.humanware.com/hus/brilliant-bi14-braille-display.html> [6.05.2020]
- Kent, Deborah. **What is Braille?** https://books.google.com.tr/books/about/What_Is_Braille.html?id=DsnFtwAACAAJ&redir_esc=y (Enslow Publishers, Berkeley, USA. 2012). [10.05.2020]
- MakeMusic Inc. “MusicXML 3.1 Tutorial”. (2017, Aralık) https://wpmedia.musicxml.com/wp-content/uploads/2017/12/musicxml-tutorial.pdf?_ga=2.25273533.1225061397.1589396902-1866173376.1589396902 [10.05.2020]

- McGinnity, B.L., Seymour-Ford, J. and Andries, K.J. **Howe Press and the Perkins Braille**. <https://www.perkins.org/history/legacy/howe-press-perkins-braille> (Perkins History Museum, Perkins School for the Blind, Watertown, MA. 2004) [14.03.2020]
- MEB. **Braille Müzik İşaretleri Sistemi Klavuzu** (Milli eğitim Basımevi, İstanbul 1991): 5-33.
- _____. **Kabartma Yazı El Kitabı** (Körler Yazı El Kitabı, 1975): 7.
- Hugo Nicolau, João Guerreiro, Tiago Guerreiro, Luís Carriço “UbiBraille: Designing and Evaluating a Vibrotactile Braille- Reading Device.” **International ACM SIGACCESS Conference on Computers and Accessibility** (Bellevue, USA. 2013): 1.
- Hugo Nicolau, Kyle Montague, Tiago Guerreiro, André Rodrigues, Vicki L. Hanson “HoliBraille: Multipoint Vibrotactile Feedback on Mobile Devices.” **The 12th Web for All Conference** (Florence, Italy 2015): 1-2.
- Nichols, Sharon. **Music Braille Handout** (Texas School for the Blind & Visually Impaired Outreach Programs, USA. 2010): 4-14.
- Paths to Literacy. “Slate and Stylus”. <https://www.pathstoliteracy.org/slate-and-stylus> [23.08.2020]
- Bertman, Stephan. “Louis Braille and the Night Writer” <https://www.historytoday.com/louis-braille-and-night-writer> [6.05.2020]
- The Braille Authority of North America. **Music Braille Code 2015** (Louisville, Kentucky, 2016): 19-171.
- Wikipedia. “Louis Braille” . https://tr.wikipedia.org/wiki/Louis_Braille [6.05.2020]
- USC Stevens. “Vibraile”. <https://stevens.usc.edu/2017-usc-stevens-student-innovator-showcase-awards-30k-to-student-startups/> [14.03.2020]
- Wikipedia. “Perkins Braille”. https://en.wikipedia.org/wiki/Perkins_Braille [6.05.2020]
- Youtube. “Vibraile”, <https://www.youtube.com/watch?v=KW5xf4pY1xU> [6.05.2020]

EKLER

Ek 1. İlk deney grubu üzerinde uygulanan hücrelerin rakamla yazıldığı soru kağıdı.

EL	YAŞ	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1000ms T1																	
T1		1	5	2 4	1 2	4 6	1 3 5	2 4 5	1 2 4	1 2 4 5	1 3 5 6	2 3 4 6	1 2 3 5 6	1 3 4 5 6	1 2 3 4 6	1 3 4 6	1 2 3 4 5 6
500ms T2																	
T2		5	1	4 6	2 4	1 2	2 4 5	1 3 5	1 3 5 6	2 3 4 6	1 2 4	1 2 4 5	1 2 3 5 6	1 3 4 6	1 2 3 4 5 6	1 3 4 5 6	1 2 3 4 6
400ms T3																	
T3		1	5	1 2	2 4	2 4 5	4 6	1 3 5	1 2 4 5	1 2 4	1 3 5 6	1 2 3 5 6	2 3 4 6	1 2 3 4 6	1 3 4 5 6	1 2 3 4 5 6	1 3 4 6
350ms T1																	
T1		1	5	2 4	1 2	4 6	1 3 5	2 4 5	1 2 4	1 2 4 5	1 3 5 6	2 3 4 6	1 2 3 5 6	1 3 4 5 6	1 2 3 4 6	1 3 4 6	1 2 3 4 5 6
300ms T2																	
T2		5	1	4 6	2 4	1 2	2 4 5	1 3 5	1 3 5 6	2 3 4 6	1 2 4	1 2 4 5	1 2 3 5 6	1 3 4 6	1 2 3 4 5 6	1 3 4 5 6	1 2 3 4 6
250ms T3																	
T3		1	5	1 2	2 4	2 4 5	4 6	1 3 5	1 2 4 5	1 2 4	1 3 5 6	1 2 3 5 6	2 3 4 6	1 2 3 4 6	1 3 4 5 6	1 2 3 4 5 6	1 3 4 6
200ms T1																	
T1		1	5	2 4	1 2	4 6	1 3 5	2 4 5	1 2 4	1 2 4 5	1 3 5 6	2 3 4 6	1 2 3 5 6	1 3 4 5 6	1 2 3 4 6	1 3 4 6	1 2 3 4 5 6
150ms T2																	
T2		5	1	4 6	2 4	1 2	2 4 5	1 3 5	1 3 5 6	2 3 4 6	1 2 4	1 2 4 5	1 2 3 5 6	1 3 4 6	1 2 3 4 5 6	1 3 4 5 6	1 2 3 4 6

Ek 2. İlk deney grubu, 40 kişide sol ve sağ el bitişik-ayrık cevap ortalamalarının gösterildiği 1. tablo.

sol ayrık 1000	sol bitişik 1000	sağ ayrık 1000	sağ bitişik 1000	sol ayrık 500	sol bitişik 500	sağ ayrık 500	sağ bitişik 500	sol ayrık 400	sol bitişik 400	sağ ayrık 400	sağ bitişik 400	sol ayrık 350	sol bitişik 350	sağ ayrık 350	sağ bitişik 350
4	4	4	2	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	2	3	4	4	3	4	4	4	3	4	4	4	4
4	4	4	4	3	3	4	3	4	4	3	3	4	4	2	2
4	4	4	4	3	4	4	4	4	4	3	4	4	4	3	4
4	4	4	3	4	4	4	4	4	3	4	3	4	4	4	4
4	4	4	4	3	4	4	4	3	3	4	3	4	4	4	4
2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	3	4	4	4	1	3	3	3
4	3	4	3	4	3	3	4	4	4	3	4	4	4	3	4
4	4	4	4	4	3	4	4	4	4	3	3	4	3	4	4
4	4	4	4	4	4	4	4	4	4	3	3	2	4	3	4
4	3	4	4	4	4	4	4	3	4	4	4	4	3	4	4
4	4	4	4	4	4	4	4	4	4	2	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	2	4	4	3	4	4	3	4	4	4	4
4	4	4	4	4	4	4	3	4	4	2	4	4	4	2	3
4	4	4	4	4	4	3	4	4	4	4	4	4	3	4	4
4	4	4	4	4	4	4	4	4	4	3	3	4	3	3	4
4	4	4	4	4	3	3	4	4	3	4	4	4	3	4	3
4	4	4	4	4	4	4	4	4	4	4	3	4	3	4	4
4	4	4	4	3	4	4	4	4	4	3	4	4	4	3	4
4	3	4	4	4	4	4	4	4	4	4	4	3	3	3	3
4	4	3	4	4	4	4	4	4	4	4	4	4	4	4	4
2	4	4	4	4	4	4	4	3	4	4	4	4	4	4	3
4	4	4	3	4	4	4	4	4	4	4	4	4	4	4	3
3	4	4	3	4	4	3	4	4	4	4	4	4	4	4	3
4	4	4	2	4	4	4	4	4	4	4	4	4	4	3	3
3	4	4	4	4	3	4	4	3	4	3	3	4	4	4	3
4	4	4	3	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	3	3	3	4	1	3	4	4	2	4	4
4	4	4	4	4	4	4	2	4	4	3	4	4	4	4	3
3	4	4	4	4	4	4	4	4	4	4	4	4	2	3	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	2
4	4	3	4	3	3	2	3	1	4	1	3	3	3	3	1
4	4	3	4	2	4	3	3	1	4	1	3	3	3	3	2
4	4	4	2	4	3	4	3	4	4	4	4	4	4	4	4
4	3	4	4	4	4	2	4	3	4	4	4	4	2	4	4
4	4	4	4	4	4	4	4	4	4	2	4	4	4	4	4
3.825	3.9	3.925	3.675	3.725	3.8	3.775	3.775	3.675	3.825	3.45	3.725	3.8	3.6	3.625	3.525

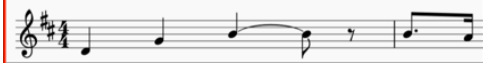
Ek 3. İlk deney grubu, 40 kişide sol ve sağ el bitişik-ayrık cevap ortalamalarının gösterildiği 2. tablo.

sol ayrık 300	sol bitişik 300	sağ ayrık 300	sağ bitişik 300	sol ayrık 250	sol bitişik 250	sağ ayrık 250	sağ bitişik 250	sol ayrık 200	sol bitişik 200	sağ ayrık 200	sağ bitişik 200	sol ayrık 150	sol bitişik 150	sağ ayrık 150	sağ bitişik 150
3	4	4	2	3	3	4	3	1	3	3	1	1	1	1	0
3	4	3	3	2	4	1	3	3	3	1	4	2	4	3	2
3	2	4	2	4	4	3	3	3	2	2	2	1	1	0	3
4	4	4	4	4	4	3	4	4	4	2	3	4	4	2	3
4	4	4	4	4	3	4	4	3	4	4	4	4	4	3	4
4	4	4	3	4	4	4	4	2	4	3	4	2	3	3	2
4	4	4	4	4	4	4	4	4	4	2	2	1	4	3	2
4	4	4	4	2	3	4	3	4	4	4	2	3	2	1	3
4	3	2	4	3	3	2	2	3	1	2	1	3	4	4	1
4	4	4	4	4	3	4	4	4	4	3	3	4	4	3	3
4	4	2	3	4	3	2	2	3	4	3	2	2	4	1	2
4	2	3	3	4	3	2	3	4	1	4	2	3	3	2	0
4	4	3	3	4	4	2	3	3	4	2	4	4	4	2	2
4	4	4	3	3	3	3	3	4	4	3	2	4	3	2	1
4	4	4	4	4	4	4	4	4	4	4	3	4	4	4	2
4	2	3	3	4	3	2	4	3	4	3	3	4	4	2	3
3	3	2	3	0	3	2	1	3	2	3	2	4	2	1	1
4	3	4	4	4	4	3	4	4	4	4	3	3	4	3	3
4	3	3	4	4	4	4	4	2	3	4	3	3	2	3	1
4	2	3	4	4	3	3	4	4	3	2	3	2	2	3	2
4	4	4	4	4	4	4	3	3	3	1	4	2	4	3	2
4	4	4	4	4	4	3	4	3	2	2	2	1	1	0	3
2	4	3	4	0	3	3	4	4	4	2	3	4	4	2	3
4	4	4	4	4	4	3	4	3	4	4	4	4	4	3	4
4	4	4	3	4	4	4	4	2	4	3	4	2	3	3	2
4	4	4	4	3	3	4	4	4	4	2	2	1	4	3	2
3	4	4	4	3	4	3	2	4	4	4	2	3	2	1	3
4	4	4	4	4	4	4	4	3	1	2	1	3	4	4	1
4	3	4	4	4	3	3	3	4	4	3	3	4	4	3	3
4	4	4	4	4	4	4	4	3	4	3	2	2	4	1	2
4	1	3	3	4	2	4	4	4	1	4	2	3	3	2	0
4	3	3	2	4	4	1	4	3	4	2	4	4	4	2	2
4	4	3	4	3	4	4	4	4	4	3	2	4	3	2	1
4	4	4	4	4	4	3	4	4	4	4	3	4	4	4	2
4	4	4	4	4	4	3	4	3	3	3	4	4	3	2	1
0	3	2	2	2	3	1	1	1	2	1	3	1	2	1	3
1	3	1	0	1	3	2	2	4	3	3	4	4	3	2	2
3	4	4	4	4	4	3	4	4	3	2	2	1	4	2	3
4	2	3	3	4	2	2	3	1	4	2	3	4	2	4	4
4	4	2	2	4	4	2	2	4	2	4	4	4	3	2	4
3.625	3.475	3.4	3.375	3.425	3.5	3	3.35	3.25	3.25	2.8	2.775	2.925	3.2	2.3	2.175

Ek 4. İkinci deney grubu hücrelerin rakamla yazıldığı test soru kağıdı.

+ _ + _ # _ > 4 _ < 4 _ o4 _ d2 _ g2 _ b2 _ ^1 _ ^2 _ b3 _ r3 _ |
 _ b3 _ . _ a0 _

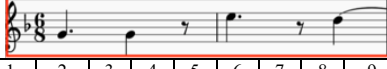
TEST 1



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
TEST 1 Cevapları	1 4 6	14 6	3 4 5 6	1 4 5 6	2 5 6	1 5 6	1 2 5 6	2 4 5 6	4	14	24 5	13 46	0	24 5	3	23 46	0

_ # _ > 6 _ < 8 _ o4 _ g2 _ . _ g2 _ r3 _ | _ o5 _ e2 _ . _ r3 _ d2 _ ^1 _ ^2 _

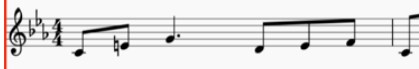
TEST 2



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
TEST 2 Cevapları	1 2 6	34 56	1 2 3	2 3 6	5	1 2 5 6	3	1 2 5 6	13 46	0	46	12 46	3	13 46	15 6	4	14

_ _ _ # _ > 4 _ < 4 _ o4 _ c3 _ n _ e3 _ g2 _ . _ d3 _ e3 _ f3 _ | _ c3 _

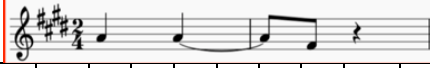
TEST 3



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
TEST 3 Cevapları	1 2 6	12 6	1 2 6	3 5 7	1 4 5	2 5 6	5	1 4 5	16	12 4	12 56	3	15	12 4	12 45	0	14 5

+ _ + _ + _ # _ > 2 _ < 4 _ o4 _ a2 _ a2 _ ^1 _ ^2 _ | _ a3 _ f3 _ r2 _ | _

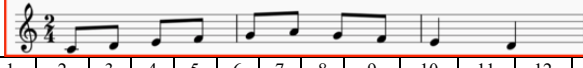
TEST 4



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
TEST 4 Cevapları	1 4 6	14 6	1 4 6	1 4 6	1 4 5	1 2 6	2 5 6	5	24 6	24 6	4	14	0	24	12 45	12 36	0

_ > 2 _ < 4 _ o4 _ c3 _ d3 _ e3 _ f3 _ | _ g3 _ a3 _ g3 _ f3 _ | _ e2 _ d2 _ | _

TEST 5



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
TEST 5 Cevapları	3 4 5 6	12	2 5 6	5	1 4 5	1 5	1 2 4	1 2 4	0	12 5	24	12 5	12 45	0	12 46	15 6	0

_ > 2 _ < 4 _ o5 _ g1 _ | _ c2 _ _ e2 _ | _ c3 _ c3 _ _ e3 _ e3 _ | _ g3 _

TEST 6



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
TEST 6 Cevapları	3 4 5 6	12	2 5 6	4 6	1 2 3 5	0	1 4 5 6	1 2 6	12 46	0	14 5	14 5	12 6	12 4	12 4	0	12 5

Ek 5. İkinci Deney Grubu vBraille Yorumları.

Aslı Şensoy:

Cihazın maddi açıdan uygun olması çok iyi bir şey. Çok güzel, başarılı bir cihaz. Hız arttırma ve azaltma tuşu eklenirse kişinin kendi hızını algılaması daha rahat olacaktır. İlköğretim düzeyinde çalışan öğrenci ve öğretmenler için çok öğretici bir cihaz olacaktır.

Arzu Şensoy:

Müzik notasyonu rahat bir şekilde okunuyor. Tasarımı da güzel.

Koray Sazlı:

Cihaz son derece fonksiyonel, ancak cihaz üzerinden parça icrası çok zor. Bir diğer yandan, kişiye noktaları titreşerek göstermesi çok yararlı bir yöntem, öğretme konusunda çok faydalı olacağını düşünüyorum.

Kişi hücreyi algılamada emin değilse vBraille üzerinden hız arttırma ve azaltma tuşu eklendiğinde hücreyi teyit etmek daha rahat olacaktır.

vBraille’de hücreleri tek tek okumak kabartma yazıya göre yavaş olduğunu gösterir. Hiçbir cihazın olmadığı aciliyet gerektiren nokta tanımlama cihazı olarak kullanılabilir, “fax cihazı gibi, acil ihtiyaçta camı kırın”.

Öneri:

Sol ve sağ el parmaklarını temsil eden noktaların arasında yarım saniye ya da 1 saniye arayla titremesi nota algılamayı daha rahat hale getirebilir.

Çağlar Arsu:

Cihaz üzerinden müzik okumak çok vakit alabilir.

Öneri: motorların daha büyük olması lazım, parmakların içeriyle girdiği bir tasarım olursa okumak daha rahat olur.

Mert Karagüzel:

Kabartma yazıda noktaları hemen algılamak nota okumada bir zorluk yaratmıyor. Fakat, vBraille kullanırken önce harf ya da rakam olarak düşünüyör sonra nota karşılığını söylüyörüm, bu da zamanı uzatıyor.

-vBraille kullanımın en önemli özelliği, kabartma yazıda yanlış çıkan ya da silinen noktaları vBraille’de çok net bir şekilde okuyacak olabilmektir.

Öneri: vBraille üzerinde tüm hücre noktaları bir anda titreyebilir. Eğer böyle bir şekilde çalışırsa Birlik fa ve ölçü işaretini ayırmak adına ölçü işaretinde parmakların hepsini iki kez aynı anda titremeyle vermek yararlı olabilir.

Selim Altınok:

-Braille yazım programları kullanan nota ya da yazı yazarlar vBraille'e daha aşina olabilir. Biz hücreyi dikey olarak gördüğümüz için yatay dizilim yeni bir öğrenme şekli oluyor, bu sebeple yavaş okuma yapıyoruz.

-vBraille titreşimleri sırayla olmasından ziyade hepsi birlikte titreyebilir.

Kerim Altınok:

-vBraille kullanırken önce harf ya da rakam olarak düşünüyor sonra nota karşılığını buluyorum, iki aşamalı olması zamanı uzatıyor.

Özgür Altınok:

-Takoz: 6 nokta üzerinde misketler aracılığıyla braille öğrenmek için kullanılan bir araçtı, 30 yıl önce ilk braille eğitimimde kullanmıştım. VBraille'i takoz kullanan kişilerin daha hızlı anlayabileceğini düşünüyorum.

-VBraille: öğrenim ve öğretim için ideal, yavaş da olsa okumak hiç de zor değil. Musicxml okuması çok önemli. Cihazı 1-2 saat daha fazla kullansam çok daha hakim olabilirim. Kabartma yazı yöntemi ile kesinlikle karşılaştırılmamalı.

Öneri: cihaz üzerinde birkaç tane cell olursa öncesi ve sonrasına atlama ve okuma için uygun olabilir. Buna bağlı ya da bununla birlikte çalışacak "jaws" uygulaması gibi notaları tek tek duyuran seslendiren bir sistem olabilir.

Ek 6. vBraille ve musxml2vBraille Kaynak Kodları

OneButton.h

```
// -----  
  
// OneButton.h - Library for detecting button clicks, doubleclicks and  
long  
  
// press pattern on a single button. This class is implemented for use  
with the  
  
// Arduino environment. Copyright (c) by Matthias Hertel,  
// http://www.mathertel.de This work is licensed under a BSD style  
license. See  
  
// http://www.mathertel.de/License.aspx More information on:  
// http://www.mathertel.de/Arduino  
  
// -----  
  
// 02.10.2010 created by Matthias Hertel  
// 21.04.2011 transformed into a library  
// 01.12.2011 include file changed to work with the Arduino 1.0  
environment  
  
// 23.03.2014 Enhanced long press functionalities by adding longPressStart  
and  
  
// longPressStop callbacks  
  
// 21.09.2015 A simple way for debounce detection added.  
// 14.05.2017 Debouncing improvements.  
// 25.06.2018 Optional third parameter for deactivating pullup.  
// 26.09.2018 Anatoli Arkhipenko: Included solution to use library with  
other  
  
// sources of input.  
// 26.09.2018 Initialization moved into class declaration.  
// 26.09.2018 Jay M Ericsson: compiler warnings removed.  
  
// -----  
  
#ifndef OneButton_h
```

```

#define OneButton_h

#include "Arduino.h"

// ----- Callback function types -----

extern "C" {
typedef void (*callbackFunction)(void);
}

class OneButton
{
public:
    // ----- Constructor -----
    OneButton();

    OneButton(int pin, int active, bool pullupActive = true);

    // ----- Set runtime parameters -----

    // set # millisec after safe click is assumed.
    void setDebounceTicks(int ticks);

    // set # millisec after single click is assumed.
    void setClickTicks(int ticks);

    // set # millisec after press is assumed.
    void setPressTicks(int ticks);

```

```

// attach functions that will be called when button was pressed in the
// specified way.

void attachClick(callbackFunction newFunction);

void attachDoubleClick(callbackFunction newFunction);

void attachPress(
    callbackFunction    newFunction); // DEPRECATED, replaced by
longPressStart,

                                // longPressStop and duringLongPress

void attachLongPressStart(callbackFunction newFunction);
void attachLongPressStop(callbackFunction newFunction);
void attachDuringLongPress(callbackFunction newFunction);

// ----- State machine functions -----

/**
 * @brief Call this function every some milliseconds for checking the
input
 * level at the initialized digital pin.
 */
void tick(void);

/**
 * @brief Call this function every time the input level has changed.
 * Using this function no digital input pin is checked because the
current
 * level is given by the parameter.
 */
void tick(bool level);

bool isLongPressed();

```

```

int getPressedTicks();

void reset(void);

private:

int _pin; // hardware pin number.

unsigned int _debounceTicks = 50; // number of ticks for debounce times.

unsigned int _clickTicks = 400; // number of ticks that have to pass by
                                // before a click is detected.

unsigned int _pressTicks = 1000; // number of ticks that have to pass by
                                // before a long button press is
detected

int _buttonPressed;

bool _isLongPressed = false;

// These variables will hold functions acting as event source.
callbackFunction _clickFunc = NULL;
callbackFunction _doubleClickFunc = NULL;
callbackFunction _pressFunc = NULL;
callbackFunction _longPressStartFunc = NULL;
callbackFunction _longPressStopFunc = NULL;
callbackFunction _duringLongPressFunc = NULL;

// These variables that hold information across the upcoming tick calls.
// They are initialized once on program start and are updated every time
the
// tick function is called.

int _state = 0;

unsigned long _startTime; // will be set in state 1

```

```
    unsigned long _stopTime; // will be set in state 2
};
```

```
#endif
```

OneButton.cpp

```
/**
 * @file OneButton.cpp
 *
 * @brief Library for detecting button clicks, doubleclicks and long press
 * pattern on a single button.
 *
 * @author Matthias Hertel, https://www.mathertel.de
 * @Copyright Copyright (c) by Matthias Hertel, https://www.mathertel.de.
 *
 * This work is licensed under a BSD style license. See
 * http://www.mathertel.de/License.aspx
 *
 * More information on:
 * https://www.mathertel.de/Arduino/OneButtonLibrary.aspx
 *
 * Changelog: see OneButton.h
 */

#include "OneButton.h"

// ----- Initialization and Default Values -----
```

```

/**
 * @brief Construct a new OneButton object but not (yet) initialize the IO
 * pin.
 */
OneButton::OneButton()
{
    _pin = -1;
    // further initialization has moved to OneButton.h
}

OneButton::OneButton(int pin, int activeLow, bool pullupActive)
{
    // OneButton();
    _pin = pin;

    if (activeLow) {
        // the button connects the input pin to GND when pressed.
        _buttonPressed = LOW;
    } else {
        // the button connects the input pin to VCC when pressed.
        _buttonPressed = HIGH;
    } // if

    if (pullupActive) {
        // use the given pin as input and activate internal PULLUP resistor.
        pinMode(pin, INPUT_PULLUP);
    } else {
        // use the given pin as input
        pinMode(pin, INPUT);
    }
}

```

```

    } // if
} // OneButton

// explicitly set the number of millisec that have to pass by before a
click is
// assumed as safe.
void OneButton::setDebounceTicks(int ticks)
{
    _debounceTicks = ticks;
} // setDebounceTicks

// explicitly set the number of millisec that have to pass by before a
click is
// detected.
void OneButton::setClickTicks(int ticks)
{
    _clickTicks = ticks;
} // setClickTicks

// explicitly set the number of millisec that have to pass by before a
long
// button press is detected.
void OneButton::setPressTicks(int ticks)
{
    _pressTicks = ticks;
} // setPressTicks

```



```

// save function for click event

void OneButton::attachClick(callbackFunction newFunction)
{
    _clickFunc = newFunction;
} // attachClick


// save function for doubleClick event

void OneButton::attachDoubleClick(callbackFunction newFunction)
{
    _doubleClickFunc = newFunction;
} // attachDoubleClick


// save function for press event
// DEPRECATED, is replaced by attachLongPressStart, attachLongPressStop,
// attachDuringLongPress,
void OneButton::attachPress(callbackFunction newFunction)
{
    _pressFunc = newFunction;
} // attachPress


// save function for longPressStart event

void OneButton::attachLongPressStart(callbackFunction newFunction)
{
    _longPressStartFunc = newFunction;
} // attachLongPressStart


// save function for longPressStop event

void OneButton::attachLongPressStop(callbackFunction newFunction)

```

```

{
    _longPressStopFunc = newFunction;
} // attachLongPressStop

// save function for during longPress event
void OneButton::attachDuringLongPress(callbackFunction newFunction)
{
    _duringLongPressFunc = newFunction;
} // attachDuringLongPress

// function to get the current long pressed state
bool OneButton::isLongPressed(){
    return _isLongPressed;
}

int OneButton::getPressedTicks(){
    return _stopTime - _startTime;
}

void OneButton::reset(void){
    _state = 0; // restart.
    _startTime = 0;
    _stopTime = 0;
    _isLongPressed = false;
}

/**
 * @brief Check input of the configured pin and then advance the finite
state

```

```

* machine (FSM).

*/

void OneButton::tick(void)
{
    if (_pin >= 0) {
        tick(digitalRead(_pin) == _buttonPressed);
    }
}

/**
 * @brief Advance the finite state machine (FSM) using the given level.
 */
void OneButton::tick(bool activeLevel)
{
    unsigned long now = millis(); // current (relative) time in msecs.

    // Implementation of the state machine

    if (_state == 0) { // waiting for menu pin being pressed.
        if (activeLevel) {
            _state = 1; // step to state 1
            _startTime = now; // remember starting time
        } // if

    } else if (_state == 1) { // waiting for menu pin being released.

        if ((!activeLevel) &&
            ((unsigned long)(now - _startTime) < _debounceTicks)) {
            // button was released too quickly so I assume some debouncing.

```

```

        // go back to state 0 without calling a function.
        _state = 0;

    } else if (!activeLevel) {
        _state = 2; // step to state 2
        _stopTime = now; // remember stopping time

    } else if ((activeLevel) &&
                ((unsigned long)(now - _startTime) > _pressTicks)) {
        _isLongPressed = true; // Keep track of long press state
        if (_pressFunc)
            _pressFunc();
        if (_longPressStartFunc)
            _longPressStartFunc();
        if (_duringLongPressFunc)
            _duringLongPressFunc();
        _state = 6; // step to state 6
        _stopTime = now; // remember stopping time
    } else {
        // wait. Stay in this state.
    } // if

} else if (_state == 2) {
    // waiting for menu pin being pressed the second time or timeout.
    if (_doubleClickFunc == NULL ||
        (unsigned long)(now - _startTime) > _clickTicks) {
        // this was only a single short click
        if (_clickFunc)
            _clickFunc();
        _state = 0; // restart.
    }
}

```

```

    } else if ((activeLevel) &&
               ((unsigned long)(now - _stopTime) > _debounceTicks)) {
        _state = 3; // step to state 3
        _startTime = now; // remember starting time
    } // if

    } else if (_state == 3) { // waiting for menu pin being released
finally.

        // Stay here for at least _debounceTicks because else we might end up
in
        // state 1 if the button bounces for too long.
        if ((!activeLevel) &&
            ((unsigned long)(now - _startTime) > _debounceTicks)) {
            // this was a 2 click sequence.
            if (_doubleClickFunc)
                _doubleClickFunc();
            _state = 0; // restart.
            _stopTime = now; // remember stopping time
        } // if

    } else if (_state == 6) {
        // waiting for menu pin being release after long press.
        if (!activeLevel) {
            _isLongPressed = false; // Keep track of long press state
            if (_longPressStopFunc)
                _longPressStopFunc();
            _state = 0; // restart.
            _stopTime = now; // remember stopping time
        } else {

```

```

        // button is being long pressed

        _isLongPressed = true; // Keep track of long press state

        if (_duringLongPressFunc)

            _duringLongPressFunc();

    } // if

} // if

} // OneButton.tick()

// end.

```

vBrailleDuino.h

```

#ifndef _vBrailleDuinoH
#define _vBrailleDuinoH

#include <Arduino.h>
#include <SD.h>

#define SD_SC_PIN 10

typedef struct dict {

    char id;

    int value;

} t_dict;

int getValueFromDict(char c, t_dict *d, int dictSize);

void updateLEDs(int veri, int delayTime, int updateType);

```

```

class VBD_Handler {
public:
    VBD_Handler(void);

    void Navigate(int buttonID);

    void Parse(char *data);

    void SetFile(File file);

    void SetDelayTime(int delayTime);

    void SetFilePos(int filePos);

    int GetFilePos();

    void SetFileSize(unsigned long fileSize);

    unsigned long GetFileSize();

    unsigned long filePos;

private:
    File _file;

    int _buttonID;

    long _filePos;

    char _dataRead[4];

    int _delayTime;

    unsigned long _fileSize;

    int _cellNo;

    int _route;

    int _cellStep;
};

#endif

```

vBrailleDuino.cpp

```
#include "vBrailleDuino.h"
```

```
//The binary format and LED order is as "321456"
```

```
t_dict noteNames[7]          = { {'c', B001110},  
                                  {'d', B001010},  
                                  {'e', B011100},  
                                  {'f', B011110},  
                                  {'g', B011010},  
                                  {'a', B010100},  
                                  {'b', B010110}  
                                };
```

```
t_dict noteValues[4]         = { {'0', B100001},  
                                  {'1', B100000},  
                                  {'2', B000001},  
                                  {'3', B000000}  
                                };
```

```
t_dict octaveNumbers[7]     = { {'1', B000100},  
                                  {'2', B000110},  
                                  {'3', B000111},  
                                  {'4', B000010},  
                                  {'5', B000101},  
                                  {'6', B000011},  
                                  {'7', B000001}  
                                };
```

```
t_dict rests[4]             = { {'0', B101100},  
                                  {'1', B101001},  
                                  {'2', B111001},
```



```

        {'3', B101101}
    };

t_dict accidentalsAndDot[4]    = { {'n', B001001},
                                     {'+', B001101},
                                     {'-', B011001},
                                     {'.', B100000}
    };

t_dict misc[2]                = { {'|', B111111},
                                     {'#', B100111}
    };

t_dict numbersTop[10]         = { {'0', B010110},
                                     {'1', B001000},
                                     {'2', B011000},
                                     {'3', B001100},
                                     {'4', B001110},
                                     {'5', B001010},
                                     {'6', B011100},
                                     {'7', B011110},
                                     {'8', B011010},
                                     {'9', B010100}
    };

t_dict numbersBottom[10]      = { {'0', B100011},
                                     {'1', B010000},
                                     {'2', B110000},
                                     {'3', B010010},
                                     {'4', B010011},

```

```

        {'5', B010001},
        {'6', B110010},
        {'7', B110011},
        {'8', B110001},
        {'9', B100010}
    };

    t_dict ties[2] = { {'1', B000100},
                       {'2', B001100}
    };

```

```

int uType=0;

```

```

/*****/
/*THIS IS THE HANDLER CLASS FOR VBD*/
/*****/

```

```

VBD_Handler::VBD_Handler() {
    _filePos = 0;
    _delayTime = 500;
    _route = 1;
}

```

```

void VBD_Handler::Navigate(int buttonID) {

    _buttonID = buttonID;
    switch(_buttonID) {

```

```

case 1:                                     //Left button clicked

    _cellStep = (_route==1) ? 8 : 4; // if previous route was right jump
2 cells back

    _filePos = _filePos - _cellStep;
    _filePos = _filePos < 0 ? 0 : _filePos;
    _cellNo = (_filePos/4)+1;
    Serial.print("CELL: ");
    Serial.print(_cellNo);
    Serial.print("  @");
    Serial.print(_delayTime);
    Serial.println("ms");
    Serial.println("-----");
    for (int i=0; i<4; i++) {
        _file.seek(_filePos + i);
        _dataRead[i] = _file.peek();
        Serial.print(_dataRead[i]);
    }
    Serial.println();
    Parse(_dataRead);
    _route = 0;
    break;

case 2:                                     //Right button clicked

    if (_route==0) _filePos = _filePos + 4; // if previous route was
left jump next cell

    _cellNo = (_filePos/4)+1;
    Serial.print("CELL: ");
    Serial.print(_cellNo);
    Serial.print("  @");
    Serial.print(_delayTime);

```

```

        Serial.println("ms");

        Serial.println("-----");

        for (int i=0; i<4; i++) {
            _file.seek(_filePos + i);
            _dataRead[i] = _file.peek();
            Serial.print(_dataRead[i]);
        }

        Serial.println();

        _filePos = (_filePos == _fileSize - 4) ? _filePos : _filePos + 4;
//if end of file stay there

        Parse(_dataRead);

        _route = 1;

        break;
    }
}

```

```

void VBD_Handler::Parse(char *data) {
    uint8_t ledState=0;
    switch(data[0]) {
        case 'c':
        case 'd':
        case 'e':
        case 'f':
        case 'g':
        case 'a':
        case 'b':
            ledState =   getValueFromDict(data[0], noteNames, 7)
                        + getValueFromDict(data[1], noteValues, 4);

            uType = 0;

            break;
    }
}

```

```

case 'r':
    ledState = getValueFromDict(data[1], rests, 4);
    uType = 0;
    break;

case 'o':
    ledState = getValueFromDict(data[1], octaveNumbers, 7);
    uType = 0;
    break;

case 'n':
case '+':
case '-':
case '.':
    ledState = getValueFromDict(data[0], accidentalsAndDot, 4);
    uType = 0;
    break;

case '>':
    ledState = getValueFromDict(data[1], numbersTop, 10);
    uType = 0;
    break;

case '<':
    ledState = getValueFromDict(data[1], numbersBottom, 10);
    uType = 0;
    break;

case '|':

```

```

        ledState = getValueFromDict(data[0], misc, 2);
        uType = 1;
        break;

    case '#':
        ledState = getValueFromDict(data[0], misc, 2);
        uType = 0;
        break;

    case '^':
        ledState = getValueFromDict(data[1], ties, 2);
        uType = 0;
        break;

    }
    updateLEDs(ledState, _delayTime, uType);
}

void VBD_Handler::SetFile(File file) {
    _file = file;
}

void VBD_Handler::SetDelayTime(int delayTime) {
    _delayTime = delayTime;
}

void VBD_Handler::SetFilePos(int filePos) {
    _filePos = filePos;
}

```

```

int VBD_Handler::GetFilePos() {
    return _filePos;
}

void VBD_Handler::SetFileSize(unsigned long fileSize) {
    _fileSize = fileSize;
}

unsigned long VBD_Handler::GetFileSize() {
    return _fileSize;
}

//Function to read from dictionary
int getValueFromDict(char c, t_dict *d, int dictSize) {
    int i, val;
    for (i=0; i<dictSize; i++) {
        if (d[i].id==c) {
            val=d[i].value;
            break;
        }
    }
    return (i<dictSize) ? val : -1;
}

//UPDATE LEDs
void updateLEDs(int cell, int delayTime, int updateType) {

```

```

uint8_t idx[6] = {2, 1, 0, 3, 4, 5};
uint8_t pins[6] = {6, 5, 4, 7, 8, 9};
uint8_t mask[6] = {32, 16, 8, 4, 2, 1};

switch (updateType) {

case 0:
    for (int i=0; i<6; i++) {
        if ((cell & mask[idx[i]]) == mask[idx[i]]) {
            digitalWrite(pins[idx[i]], HIGH);
            Serial.println(pins[idx[i]]-3);
            delay(delayTime);
            digitalWrite(pins[idx[i]], LOW);
        }
        else
            digitalWrite(pins[idx[i]], LOW);
    }
    break;

case 1:
    for (int i=0; i<6; i++) {
        if ((cell & mask[i]) == mask[i]) {
            digitalWrite(pins[i], HIGH);
            Serial.println(pins[i]-3);
        }
        else
            digitalWrite(pins[i], LOW);
    }
    delay (300);
    for (int i=0; i<6; i++) {

```



```

        digitalWrite(pins[i], LOW);
    }
}
}

```

vBrailleDuino.ino

```

#include <SPI.h>

#include <SD.h>

#include "vBrailleDuino.h"

#include "OneButton.h"

int intervalDelayTime = 1000;

uint8_t mode = 1; //mode is 1 for reading mode, 0 for file mode

String command;

OneButton button1(2, false);

OneButton button2(3, false);


VBD_Handler handler;

File file;

void setup() {
    button1.attachClick(btn1Click);
    button1.attachDoubleClick(btn1DoubleClick);
    button2.attachClick(btn2Click);
}

```

```

button2.attachDoubleClick(btn2DoubleClick);

for (int i=4; i<10; i++) {
    pinMode(i, OUTPUT);
}

Serial.begin(115200);

Serial.println("Initializing SD card ...");


if (!SD.begin(SD_SC_PIN)) {
    Serial.println("Initialization failed !!!");
    while(1);
}

else {
    Serial.println("SD card successfully initialized ...");
}


if (file) file.close();
if (file = SD.open("d1")) {
    Serial.print("Opened file: ");
    Serial.println("d1");
    handler.SetFile(file);
    handler.SetFileSize(file.size());
    unsigned long fs = handler.GetFileSize();
    Serial.print("File length: ");
    Serial.println(fs);
    Serial.print("Number of cells: ");
    Serial.println(fs/4);
    handler.SetFilePos(0);
}

```

```

else {
    Serial.println("File could not open ...");
}

handler.SetFile(file); //Bu if clause içinde olmalı sanki
}

void loop() {
    readSerialAndParseCommand();
    switch(mode) {
        case 0:
            break;
        case 1:
            checkButtons();
            break;
    }
}

void checkButtons() {
    button1.tick();
    button2.tick();
}

void btn1Click() {
    handler.Navigate(1); // 1 for reverse
}

void btn2Click() {
    handler.Navigate(2); // 2 for forward
}

```

```

}

void btn1DoubleClick() {
    delay(400);
    handler.SetFilePos(0);
    Serial.println("Set to beginning of file ...");
}

void btn2DoubleClick() {
    delay(400);
    handler.SetFilePos(handler.GetFilePos()-4);
    handler.Navigate(2);
}

void parseCommand(String cmd) {
    String part1, part2;
    part1 = cmd.substring(0, cmd.indexOf(" "));
    part2 = cmd.substring(cmd.indexOf(" ")+1);

    if(part1.equalsIgnoreCase("delay")) {
        int data = part2.toInt();
        handler.SetDelayTime(data);
        Serial.print("Delay time set to: ");
        Serial.println(data);
    }
    else if(part1.equalsIgnoreCase("goto")) {
        int data = part2.toInt();
        if (data<1 || data>(handler.GetFileSize()/4)) {
            Serial.println("Out of file size...");
        }
    }
}

```

```

else {
    handler.SetFilePos((data-1)*4);
    Serial.print("Cell position set to: ");
    Serial.println(data);
}
}

else if (part1.equalsIgnoreCase("open")) {
    if (file) file.close();
    if (file = SD.open(part2)) {
        Serial.print("Opened file: ");
        Serial.println(part2);
        handler.SetFile(file);
        handler.SetFileSize(file.size());
        unsigned long fs = handler.GetFileSize();
        Serial.print("File length: ");
        Serial.println(fs);
        Serial.print("Number of cells: ");
        Serial.println(fs/4);
        handler.SetFilePos(0);
    }

    else {
        Serial.println("File could not open ...");
    }

}

else {
    Serial.println("Command not identified...");
}
}

```

```

void readSerialAndParseCommand() {
  if(Serial.available()) {
    char c = Serial.read();

    if(c=='\n') {
      parseCommand(command);
      command = "";
    }
    else {
      command += c;
    }
  }
}

```

Musxml2vBraille.py

```
#!/usr/bin/python3
```

```
#####
```

```
# MusicXML to viBraille converter #
```

```
# by Arda Eden 2018 #
```

```
#####
```

```
import sys
```

```
import xml.etree.ElementTree as ET
```

```
if (len(sys.argv)<3):
```

```
    print("Not enough arguments...")
```

```

sys.exit()

isFirstNote = True

interval = 0

oldOct = 0

oldIndex = 0

tree = ET.parse(sys.argv[1])
root = tree.getroot()

#Necessary values for interval calculation
stepIndexes = {'c':0, 'd':1, 'e':2,
               'f':3, 'g':4, 'a':5, 'b':6}

#This is the dictionary for note steps conversion
noteSteps = {'C':'c', 'D':'d',
             'E':'e', 'F':'f',
             'G':'g', 'A':'a', 'B':'b'}

#This is the dictionary for note types conversion
noteTypes = {'whole'   :'0'      , 'half'    :'1',
             'quarter' :'2'      , 'eighth'  :'3',
             '16th'    :'0'      , '32th'   :'1',
             '64th'    :'2'      , '128th'  :'3'}

text_file = open(sys.argv[2],"w")

if (text_file):
    print('File opened without any problem ...\n')

```

```

for measures in root.findall('./part/measure'):

    #Check for the Key of music

    if(measures.find('attributes/key') is not None):

        tmpKeyNum = int(measures.find('attributes/key/fifths').text)

        if (tmpKeyNum != 0):

            if (tmpKeyNum < 0):

                tmpAccType = '-'

            elif (tmpKeyNum > 0):

                tmpAccType = '+'

        if (abs(tmpKeyNum)<4):

            for i in range(abs(tmpKeyNum)):

                print(tmpAccType + '___')

                text_file.write(tmpAccType + '___')

            else:

                print('#___')

                text_file.write('#___')

                print('>' + str(abs(tmpKeyNum)) + '___')

                text_file.write('>' + str(abs(tmpKeyNum)) + '___')

                print(tmpAccType + '___')

                text_file.write(tmpAccType + '___')

    # Check for the measure attributes

    if (measures.find('attributes/time') is not None):

        tmpBeats = measures.find('attributes/time/beats').text

        tmpBeatType = measures.find('attributes/time/beat-type').text

```



```

print('#__')
text_file.write('#__')
for i in range(len(tmpBeats)):
    print('>' + tmpBeats[i] + '__')
    text_file.write('>' + tmpBeats[i] + '__')
for i in range(len(tmpBeatType)):
    print('<' + tmpBeatType[i] + '__')
    text_file.write('<' + tmpBeatType[i] + '__')

for notes in measures.findall('note'):

    if (notes.find('type') is not None):
        noteType = noteTypes.get(notes.find('type').text)
    else:
        noteType = ''

    # If note is a rest, use "r" as the symbol
    if (notes.find('rest') is not None):
        print('r' + noteType + '__')
        text_file.write('r' + noteType + '__')

    #if not a rest keep on ...
    else:

        noteStep = noteSteps.get(notes.find('pitch/step').text)
        noteOct = int(notes.find('pitch/octave').text)

        #Get accidentals if exist.
        if (notes.find('accidental')) is not None:

```

```
acc = notes.find('accidental').text
```

```
if (acc=='flat-flat'):  
    for i in range(2):  
        print('-__')  
        text_file.write('-' + '__')
```

```
if (acc=='flat'):  
    print('-__')  
    text_file.write('-' + '__')
```

```
if (acc=='sharp'):  
    print('+__')  
    text_file.write('+' + '__')
```

```
if (acc=='double-sharp'):  
    for i in range(2):  
        print('+__')  
        text_file.write('+' + '__')
```

```
if (acc=='natural'):  
    print('n__')  
    text_file.write('n' + '__')
```

```
# !!! Note interval and octave code decision is done here !!!
```

```
# -----
```

```
# Create an octave code for the first note
```

```
if (isFirstNote is True):
```

```

        oldOct = noteOct

        oldIndex = stepIndexes.get(noteStep)

        print('o' + str(noteOct) + '__')

        text_file.write('o' + str(noteOct) + '__')

        isFirstNote = False

# Do the math and decide for octave codes :)
else:

    idx = stepIndexes.get(noteStep)

    o = noteOct - oldOct

    interval = abs((o * 7) + (idx - oldIndex)) + 1

    if ((interval >= 6) or ((interval in range(4,6)) and o !=
0)):

        print('o' + str(noteOct) + '__')

        text_file.write('o' + str(noteOct) + '__')

        oldIndex = idx

        oldOct = noteOct

# Create a note code in output

print(noteStep + noteType + '__')

text_file.write(noteStep + noteType + '__')

# Get the dot count and create a dot

# code in output for each <dot/> tag in note
notedotcount = len(notes.findall('dot'))

for i in range(notedotcount):

    print('.') + '__'

    text_file.write('.') + '__'

```

```
if (notes.find('tie') is not None):  
    myTie = notes.find('tie')  
    if(myTie.get('type') == 'start'):  
        print('^1__')  
        text_file.write('^1__')  
        print('^2' + '__')  
        text_file.write('^2__')  
    print('|' + '__')  
    text_file.write('|' + '__')  
text_file.close()
```

ÖZ GEÇMİŞ

Doğum Tarihi	28.02.1992	
Doğum yeri	İstanbul	
Eğitimi		
Yüksek Lisans	2017-	Yıldız Teknik Üniversitesi, Sosyal Bilimler Enstitüsü Müzik ve Sahne Sanatları
Lisans	2010-2015	İstanbul Bilgi Üniversitesi, Fen ve Edebiyat Fakültesi Müzik
Lise	2006-2010	Şenesenevler Lisesi
Çalıştığı Kurumlar:	2004-2020 2020-	Serbest Çalışan Govinet İnternet Hizmetleri