



Karadeniz Teknik Üniversitesi

Mühendislik Fakültesi

Bilgisayar Mühendisliği

Bilgisayar Organizasyonu Laboratuvarı

Kesmeli (Interrupt) Giriş Çıkış Deneyi

Deney Sorumlusu: Dr.Ögr.Üyesi Selçuk CEVHER

Deney Grubu: A4

Öğrenci NO: 394791

Ad Soyad:Yasemin TOPCU

Öğrenci NO:394774

Ad Soyad:Ayşenur TAK

## Deneyin Amacı

Donanımsal kesmeli giriş çıkışın nasıl yapıldığını anlamak deneyin amacıdır.

Donanımsal kesmeli giriş çıkışta sistem, donanımsal arayüzlerden gelen sinyaller ile kesmeye uğrar ve normal yürütme akışı durarak gelen kesme sinyaline cevap verilir. Deneyimizde bu kesme türünü butonlarla sisteme sinyal vererek, var olan işleyişi durdurulup başka bir işlem yaptırma olarak gerçekleştirmeye çalıştık. Butonlardan gelen sinyallere göre belli bir ledin yanmasını veya sönmesini ve aynı zamanda da lcd ekran üzerinde o an ki durumun yazdırılmasını sağladık.

## Deneyin Yapılışı

Deneyde <https://wokwi.com/> adresinde yer alan online Arduino simülatöründe Arduino Uno simülatörünü kullandık. Burada sketch.ino, diagram.json ve pitches.h dosyalarına ilgili kodlarımızı yazdık.

sketch.ino dosyasına Arduino kartında çalışacak kodları yazacağız. Arduino bir mikroişlemci kartı olduğundan ana sistem bu kart içerisinde çalışacaktır.

diagram.json dosyasına eklenecek kod modüller arası bağlantıları içerecektir.

Bu deneyimizde eklediğimiz modüller:

3 adet Register

2 adet Led

2 adet Pushbutton

1 tane LCD 20x4

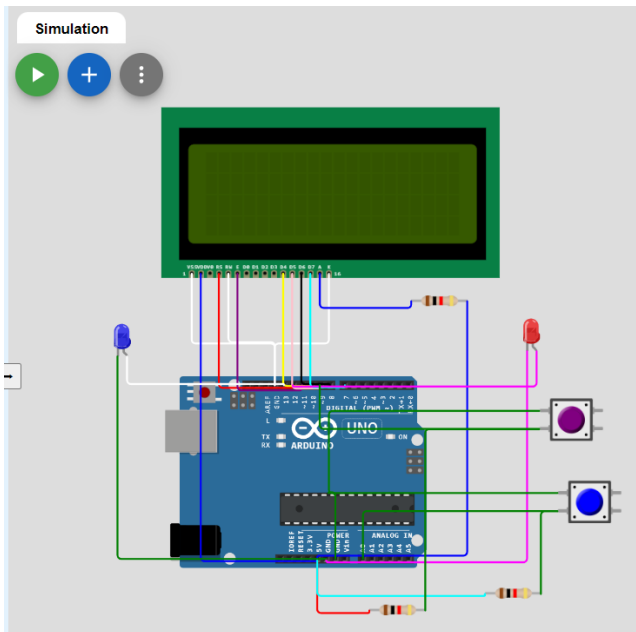
Simülatörümüzde bu modülleri eklediğimizde ve bağlantılarını yaptığımızda diagram.json dosyasını oluşturmuş oluyoruz. diagram.json dosyasındaki kodlar ile eklediğimiz modüllerin renklerini değiştirebiliyoruz. Bu deneyimizde butonların rengini mor ve mavi ,ledlerin rengini ise kırmızı ve mavi olarak kullandık.

diagram.json dosyasındaki kodlarımız aşağıdaki gibidir:

```
sketch.ino • diagram.json • pitches.h • libraries.txt • Library Manager
1 {
2   "version": 1,
3   "author": "Anonymous maker",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "wokwi-arduino-uno", "id": "uno", "top": 79.34, "left": -18.67, "attrs": {} },
7     { "type": "wokwi-led", "id": "led1", "top": 16, "left": -83.33, "attrs": { "color": "blue" } },
8     {
9       "type": "wokwi-led",
10      "id": "led2",
11      "top": 10.66,
12      "left": 346.67,
13      "attrs": { "color": "red" }
14    },
15    {
16      "type": "wokwi-resistor",
17      "id": "r1",
18      "top": -5.65,
19      "left": 240,
20      "attrs": { "value": "1000" }
21    },
22    { "type": "wokwi-resistor", "id": "r3", "top": 319.33, "left": 196, "attrs": {} },
23    {
24      "type": "wokwi-pushbutton",
25      "id": "btn1",
26      "top": 102.2,
27      "left": 374.4,
28      "attrs": { "color": "purple" }
29    },
30  ]
31 }
```

```
sketch.ino ● diagram.json ● pitches.h ● libraries.txt Library Manager ▼
30 {
31   "type": "wokwi-pushbutton",
32   "id": "btn2",
33   "top": 188.6,
34   "left": 393.6,
35   "attrs": { "color": "blue" }
36 },
37 { "type": "wokwi-lcd2004", "id": "lcd2", "top": -203.24, "left": -22.4, "attrs": { } },
38 { "type": "wokwi-resistor", "id": "r2", "top": 301.98, "left": 317.34, "attrs": { } }
39 ],
```

```
sketch.ino ● diagram.json ● pitches.h ● libraries.txt Library Manager ▼
40 "connections": [
41   [ "lcd2:VSS", "uno:GND.1", "white", [ "v73.28", "h90.01" ] ],
42   [ "lcd2:RW", "uno:GND.1", "white", [ "v71.94", "h52.18" ] ],
43   [ "lcd2:K", "uno:GND.1", "white", [ "v73.28", "h-57.32" ] ],
44   [ "lcd2:VDD", "uno:5V", "blue", [ "v301.95", "h122.51" ] ],
45   [ "lcd2:A", "r1:1", "blue", [ "v35.95", "h91.51" ] ],
46   [ "r1:2", "uno:5V", "blue", [ "v267.37", "h-152.34" ] ],
47   [ "lcd2:RS", "uno:12", "red", [ "v119.95", "h79.51" ] ],
48   [ "lcd2:E", "uno:11", "purple", [ "v122.61", "h67.35" ] ],
49   [ "lcd2:D4", "uno:10", "yellow", [ "v117.95", "h30.01" ] ],
50   [ "lcd2:D5", "uno:9", "pink", [ "v120.61", "h29.18" ] ],
51   [ "lcd2:D6", "uno:8", "black", [ "v115.95", "h30.35" ] ],
52   [ "lcd2:D7", "uno:7", "cyan", [ "v119.28", "h36.85" ] ],
53   [ "led1:A", "uno:13", "white", [ "v29.04", "h168" ] ],
54   [ "led2:A", "uno:12", "magenta", [ "v37.04", "h-251.34" ] ],
55   [ "led2:C", "uno:GND.2", "magenta", [ "v221.71", "h-208.67" ] ],
56   [ "btn1:2.1", "uno:9", "red", [ "h-244", "v-50.97" ] ],
57   [ "uno:5V", "r3:1", "red", [ "v56.86", "h51.34" ] ],
58   [ "r3:2", "btn1:2.1", "red", [ "v-188.63", "h132.33" ] ],
59   [ "btn1:1.1", "uno:GND.3", "red", [ "h-230.66", "v153.36" ] ],
60   [ "btn2:1.1", "uno:GND.3", "cyan", [ "h-236.01", "v68.03" ] ],
61   [ "btn2:2.1", "uno:A0", "cyan", [ "h-208.01", "v41.7" ] ],
62   [ "uno:5V", "r2:1", "cyan", [ "v38.86", "h174" ] ],
63   [ "r2:2", "btn2:2.1", "cyan", [ "v-85.28", "h18.99" ] ],
64   [ "led1:C", "uno: GND.2", "white", [ "v214.82", "h218.03" ] ],
65   [ "btn1:2.1", "uno:9", "green", [ "h-230.4", "v-47.8" ] ],
66   [ "btn1:1.1", "uno:GND.3", "green", [ "h-220.8", "v86.4", "h0", "v0", "h0" ] ],
67   [ "btn2:1.1", "uno:GND.3", "green", [ "h-240.41", "v-0.74", "h0" ] ],
68   [ "btn2:2.1", "uno:A0", "green", [ "h0" ] ],
69   [ "r2:2", "btn2:2.1", "green", [ "v-86.83", "h-1.74" ] ],
70   [ "r3:2", "btn1:2.1", "green", [ "v-190.58", "h4.4" ] ],
71   [ "led1:C", "uno:GND.2", "green", [ "v0" ] ]
72 ],
73 "dependencies": { }
```



ŞEKİL1

Oluşturulan diagram.json dosyasına göre yanda gördüğünüz Şekil1’deki bağlantılar kuruluyor ve dış cihazlar mikroişlemciye bağlanmış oluyor.

Sonrasında ise sketch.ino dosyasına kaynak kodlarımızı ekliyoruz.

```
sketch.ino • diagram.json • pitches.h • libraries.txt Library Manager ▼
1  #include <LiquidCrystal.h>
2  #include "pitches.h"
3
4  LiquidCrystal lcd(12, 11, 10, 9, 8, 7);
5  int mavi_led = 13;
6  int kirmizi_led = 12;
7
8  int mor_buton = 9;
9  int mavi_buton = A0;
10
11 //pin değişim kesmesi ayarlanıyor
12
13 void pciSetup(byte pin)
14 {
15     //enable pin
16     *digitalPinToPCMSK(pin) |= bit(digitalPinToPCMSKbit(pin));
17     //askıdaki tüm interruptları temizle
18     PCIFR |= bit(digitalPinToPCICRbit(pin));
19     //tekrar interruptları enable et
20     PCICR |= bit(digitalPinToPCICRbit(pin));
21 }
22
23 void setup()
24 {
25     //kullanılacak led ve buton tanımları
26     pinMode(mavi_led, OUTPUT);
27     pinMode(kirmizi_led, OUTPUT);
28     pinMode(mor_buton, INPUT);
29     pinMode(mavi_buton, INPUT);
30     //9 nolu pin için pci ayarlandı
31     pciSetup(mor_buton);
32     //A0 pini için pci ayarlandı
33     pciSetup(mavi_buton);
34 }
35
36 ~~~
40 ISR (PCINT0_vect)
41 {
42     if (digitalRead(mor_buton))
43     {
44         digitalWrite(mavi_led, HIGH);
45         lcd.begin(20, 4);
46         lcd.setCursor(3, 0);
47         lcd.print("mavi led yandı");
48     }
49 }
50
51 //Pin değişim kesmelerinin vektörleri
52 //kullanılacak pine göre vektör tanımlı yapılmalıdır
53 //Örneğin burada A0 pinini kullanmak için "PCINT1_vect" vektörü tanımlanır
54
55 ISR (PCINT1_vect)
56 {
57
58     if (digitalRead(mavi_buton))
59     {
60         digitalWrite(mavi_led, LOW);
61         lcd.begin(20, 4);
62         lcd.setCursor(3, 0);
63         lcd.print("mavi led sond");
64     }
65 }
```

```
sketch.ino • diagram.json • pitches.h • libraries.txt Library Manager ▼
67   }
68
69   void loop()
70   {
71     //mavi led 1 sn aralıklar ile yanıp söner
72     digitalWrite(kirmizi_led, !digitalRead(kirmizi_led));
73     delay(1000);
74   }
```

sketch.ino'daki kodlar mikroişlemcinin içinde çalışacak sistem kodlarıdır.

İlgili kütüphaneler dahil edildikten sonra butonlar, ledler ve lcd ekran tanımlanır.

pciSetup'da PCIFR |= bit(digitalPinToPCICRbit(pin)); kodu ilgili pinde bulunan askıdaki tüm interruptları temizlerken, PCICR |= bit(digitalPinToPCICRbit(pin)); kodu ise ilgili pinden gelecek interruptları enable eder.

setup fonksiyonunda butonları işlevleri tanımlanır.

ISR (PCINT0\_vect) ve ISR (PCINT1\_vect) vektör tanımlamaları kesme vektörleridir. Her kesme

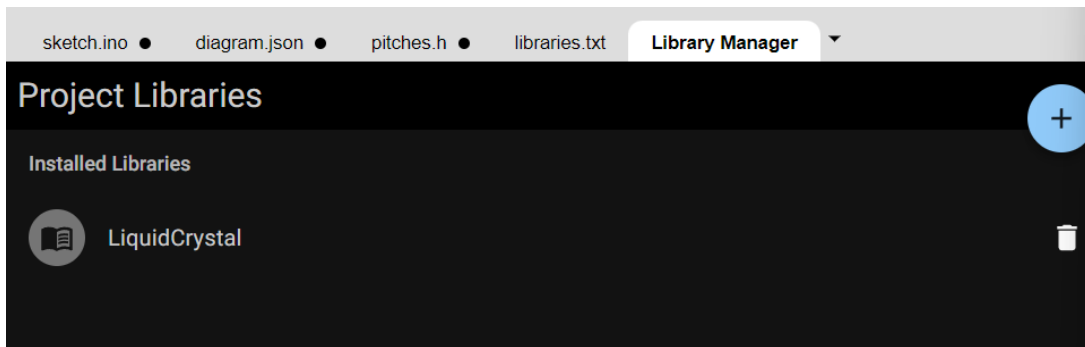
pininden gelecek sinyalde, ne yapılacağı tanımlanmaktadır. A0 pinine bağlı mavi butona

basıldığında mavi led söndürülür ve lcd ekrana "mavi led sondu" yazdırılır. D9 pinine bağlı mor

butona basıldığında ise mavi led high'a çekilir yani yanması sağlanır ve lcd ekrana "mavi led yandı" yazdırılır.

loop fonksiyonu Arduino'da sürekli çalışacak fonksiyondur. Burada 1 saniye aralıklar ile kırmızı ledin yanıp sönmelerini sağlayacak kodu yazdık.

sketch.ino dosyasında LiquidCrystal.h kütüphanesini kullandığımız için Library Manager kısmından bu kütüphaneyi ekledik. Eklenmiş şekli aşağıdaki gibidir.



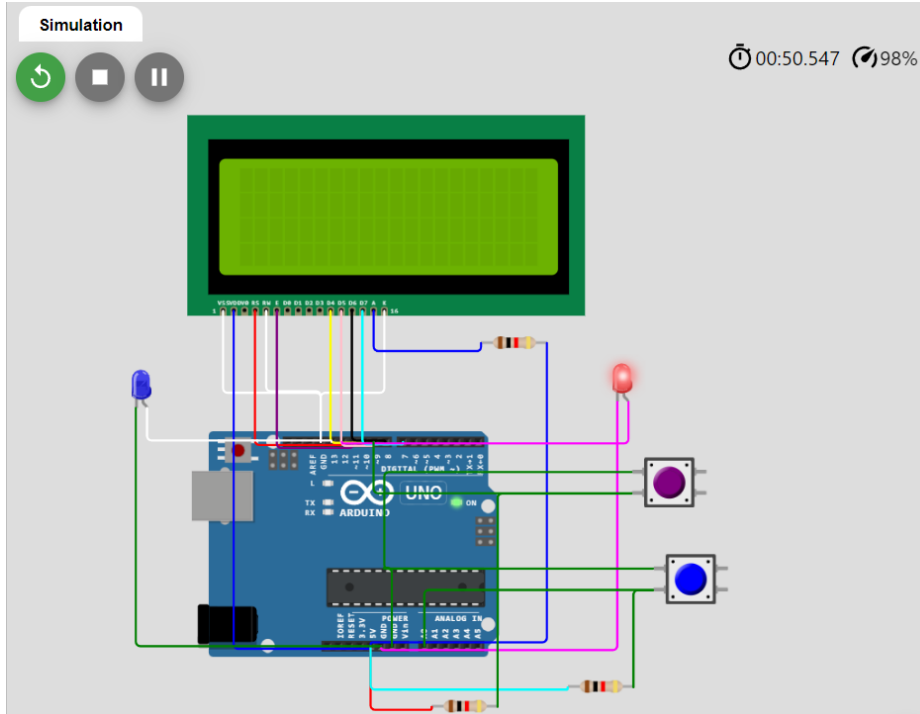
İlave olarak pitches.h dosyası ekliyoruz. Bu dosyanı içeriği aşağıdaki gibidir:

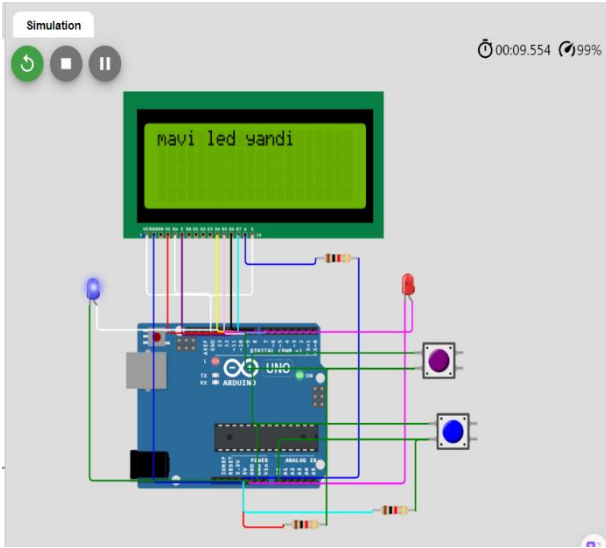
```
sketch.ino ● diagram.json ● pitches.h ● libraries.txt Library Manager ▼
1  /*****
2  | This file defines constants with the frequency
3  | of different musical notes.
4  | *****/
5
6  #define NOTE_B0  31
7  #define NOTE_C1  33
8  #define NOTE_CS1 35
9  #define NOTE_D1  37
10 #define NOTE_DS1 39
11 #define NOTE_E1  41
12 #define NOTE_F1  44
13 #define NOTE_FS1 46
14 #define NOTE_G1  49
15 #define NOTE_GS1 52
16 #define NOTE_A1  55
17 #define NOTE_AS1 58
18 #define NOTE_B1  62
19 #define NOTE_C2  65
20 #define NOTE_CS2 69
21 #define NOTE_D2  73
22 #define NOTE_DS2 78
23 #define NOTE_E2  82
24 #define NOTE_F2  87
25 #define NOTE_FS2 93
26 #define NOTE_G2  98
27 #define NOTE_GS2 104
28 #define NOTE_A2  110
29 #define NOTE_AS2 117
30 #define NOTE_B2  123
31 #define NOTE_C3  131
32 #define NOTE_CS3 139
33 #define NOTE_D3  147
34 #define NOTE_DS3 156
```

```
sketch.ino ● diagram.json ● pitches.h ● libraries.txt Library Manager ▼
34 #define NOTE_DS3 156
35 #define NOTE_E3  165
36 #define NOTE_F3  175
37 #define NOTE_FS3 185
38 #define NOTE_G3  196
39 #define NOTE_GS3 208
40 #define NOTE_A3  220
41 #define NOTE_AS3 233
42 #define NOTE_B3  247
43 #define NOTE_C4  262
44 #define NOTE_CS4 277
45 #define NOTE_D4  294
46 #define NOTE_DS4 311
47 #define NOTE_E4  330
48 #define NOTE_F4  349
49 #define NOTE_FS4 370
50 #define NOTE_G4  392
51 #define NOTE_GS4 415
52 #define NOTE_A4  440
53 #define NOTE_AS4 466
54 #define NOTE_B4  494
55 #define NOTE_C5  523
56 #define NOTE_CS5 554
57 #define NOTE_D5  587
58 #define NOTE_DS5 622
59 #define NOTE_E5  659
60 #define NOTE_F5  698
61 #define NOTE_FS5 740
62 #define NOTE_G5  784
63 #define NOTE_GS5 831
64 #define NOTE_A5  880
65 #define NOTE_AS5 932
66 #define NOTE_B5  988
67 #define NOTE_C6  1047
```

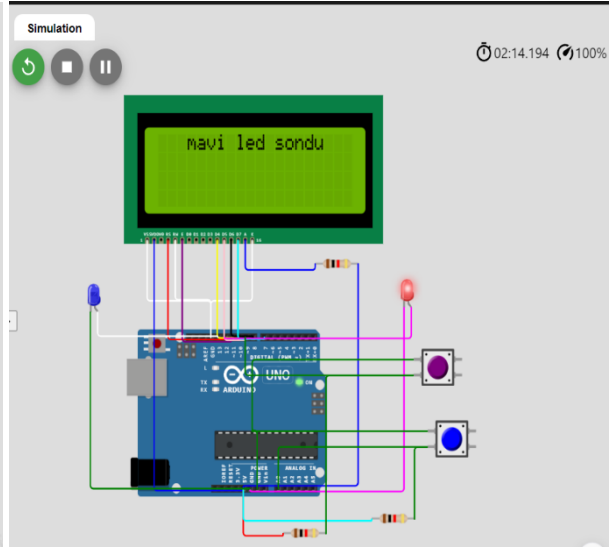
```
sketch.ino ● diagram.json ● pitches.h ● libraries.txt Library Manager ▼
62 #define NOTE_G5 784
63 #define NOTE_G55 831
64 #define NOTE_A5 880
65 #define NOTE_AS5 932
66 #define NOTE_B5 988
67 #define NOTE_C6 1047
68 #define NOTE_CS6 1109
69 #define NOTE_D6 1175
70 #define NOTE_DS6 1245
71 #define NOTE_E6 1319
72 #define NOTE_F6 1397
73 #define NOTE_FS6 1480
74 #define NOTE_G6 1568
75 #define NOTE_GS6 1661
76 #define NOTE_A6 1760
77 #define NOTE_AS6 1865
78 #define NOTE_B6 1976
79 #define NOTE_C7 2093
80 #define NOTE_CS7 2217
81 #define NOTE_D7 2349
82 #define NOTE_DS7 2489
83 #define NOTE_E7 2637
84 #define NOTE_F7 2794
85 #define NOTE_FS7 2960
86 #define NOTE_G7 3136
87 #define NOTE_GS7 3322
88 #define NOTE_A7 3520
89 #define NOTE_AS7 3729
90 #define NOTE_B7 3951
91 #define NOTE_C8 4186
92 #define NOTE_CS8 4435
93 #define NOTE_D8 4699
94 #define NOTE_DS8 4978
```

Simülasyonu başlattığımızda simülasyon boyunca aşağıdaki gibi kırmızı led yanmaktadır.





Şekil2



Şekil3

Mor butona bastığımızda şematik Şekil2'deki gibi oluyor.

Biz mor butona bastığımızda mavi ışık yanıp, ekranda mavi led yandı yazdırmış oluyoruz.

Mavi butona bastığımız zamanda (Şekil3) mavi ışık sönmüş oluyor ve ekranda mavi led söndü yazıyor.

Bu şekilde kesmeli giriş çıkış yapmış oluyoruz. Sistemin normal çalışmasında kırmızı led sürekli yanıp sönmektedir. Butonlara basıldığında sisteme normal çalışmasının dışında bir sinyal verilmiş oluyor. Sistem kesme geldiğinde normal süre gelen işini bırakıp kesmeye cevap verir. Burada mor butona basıldığında kesmenin emrettiği iş mavi butonun yanması ve lcd ekrana "Mavi led yandı" yazdırılmasıdır. Bu işlem yapıp normal işlemlere devam edilir yani kırmızı ledin belli aralıklarla yanıp sönmeye işine devam edilir. Burada süre gelen işlemin durdurulup kesmeye cevap verilmesi çok kısa süren bir işlem olduğundan biz normal gözlemimizde kırmızı ledin yanıp sönmesinde bir gecikme görmeyiz fakat aslında o işleme bir ara verilip mavi led ve lcd'ye yazma işlemi yapılır. Aynı şekilde mavi butona basıldığında da başka bir kesme daha verilir burada da mavi led söndürülür ve lcd ekrana "Mavi led sondu" yazdırılır.

## **Deneyden Öğrendiklerimiz**

Deneyde, Arduino üzerinde donanımsal kesmelerin nasıl kullanılacağını anlamak amacıyla verilen kodu simülasyon ortamında çalıştırdık. Kod, iki farklı düğme kullanılarak donanımsal kesmeleri tetikleyerek belirli kodda kesme vektörlerinde belirtilen olayları gerçekleştirmektedir. "pciSetup" fonksiyonu, donanımsal kesmelerin tanımlanması için gerekli kayıtları düzenlemekte ve bu sayede belirli bir pine bağlı kesmeleri etkinleştirmektedir. "ISR" (Interrupt Service Routine) fonksiyonları, kesme gerçekleştiğinde yapılacak işlemleri belirlemektedir. Deney, donanımsal kesmelerin, kullanıcının bir butona basarak kesme oluşturabileceği gibi bir örnek ile nasıl



kullanılabileceğini gösterdi. Ayrıca, LCD ekran ve LED kontrolü gibi uygulamalar ile Arduino programlamada donanımsal kesmelerin nasıl yapılabileceğini anlamamıza yardımcı oldu.