

# YÖBİ4461 - Nesnelerin İnterneti

## Oyun Konsolu

### 1. Giriş

Projenin temel amacı, Arduino Leonardo ve Funduino Joystick Shield devre elemanlarını kullanarak Nesnelerin İnterneti (IoT) teknolojisiyle oyunlarda kullanılabilen bir oyun konsolu oluşturmaktır. Bu elemanlar aracılığıyla C programlama diliyle elde edilen veriler kullanarak bilgisayarın fare ve klavye işlevlerini kontrol etmek için kullanılacaktır. Bu şekilde, donanım ve yazılım arasında etkileşim sağlanarak oyun konsolu oluşturulacaktır. Ayrıca, proje kapsamında oluşturulan oyun konsolunun korunması ve estetik bir görünüme sahip olması için 3D yazıcıdan özel bir kap yapılmıştır. Bu kap, devre elemanlarının dış etkenlerden korunmasını sağlarken, aynı zamanda kullanıcılar için ergonomik ve şık bir tasarım sunmaktadır. 3D yazıcı teknolojisi kullanarak üretilen bu kap, projenin hem işlevselliğini hem de görselliğini artırarak, kullanıcı deneyimini en üst düzeye çıkarmayı hedeflemektedir.



## 2. Devre Elemanları

### 1. Arduino Leonardo:

Arduino Leonardo, mikrodnetleyici tabanlı açık kaynaklı bir geliştirme kartıdır ve özellikle USB iletişim yetenekleriyle öne çıkmaktadır. Bu özellikleri sayesinde, Arduino Leonardo, çeşitli projelerde geniş bir kullanım yelpazesi sunmaktadır. Projemizde Arduino Leonardo, hem donanım hem de yazılım tarafında kilit bir rol oynamaktadır. Bu mikrodnetleyici kartı, Funduino Joystick Shield ile birlikte kullanılarak oyun konsolunun kullanıcı girişlerini doğru ve hızlı bir şekilde algılamasını sağlar. Ayrıca, 3D yazıcıdan üretilen özel kap ile koruma altına alınarak, hem işlevsellik hem de estetik açıdan projenin tamamlanmasına katkıda bulunur. Arduino Leonardo'nun tercih edilme sebepleri ve kullanım alanları şu şekildedir:

- *Bilgisayar Fare ve Klavye İşlevlerini Kontrol Etmek:* Arduino Leonardo, yerleşik USB HID (Human Interface Device) desteği sayesinde, bilgisayara doğrudan bağlanarak fare ve klavye işlevlerini taklit edebilmektedir. Bu, bir oyun konsolunun temel işlevlerinden biridir ve joystick gibi giriş cihazlarından gelen verileri kullanarak bilgisayarı kontrol etmemizi sağlamaktadır.
- *Nesnelerin İnterneti (IoT) Uygulamaları:* Arduino Leonardo, çeşitli sensör ve modüllerle kolayca entegre edilebilmektedir. IoT projeleri için ideal bir platform sağlamaktadır. Oyun konsolunun IoT teknolojisiyle çalışabilmesi için bu entegrasyon oldukça önem arz etmektedir.

Projede kullanılma sebepleri arasında gelişmiş USB iletişimi ile Diğer Arduino modellerine kıyasla, Leonardo'nun USB HID desteği, onu fare ve klavye emülasyonu için ideal hale getirmektedir. Bu özellik, projede kullanılması planlanan joystick ve diğer giriş cihazlarının etkin ve doğru bir şekilde çalışmasını sağlamaktadır. Açık Kaynak ve kolay programlanabilir olması sayesinde Arduino Leonardo'nun açık kaynaklı yapısı ve geniş topluluk desteği, geliştirme sürecini kolaylaştırmaktadır. C programlama diliyle kolayca programlanabilir olması, projede ihtiyaç duyulan özel işlevlerin hızlı ve verimli bir şekilde gerçekleştirmektedir. Esneklik ve Çok Yönlülük yapısı, çeşitli sensörler ve modüllerle uyumlu çalışmasını sağlamaktadır. Bu esneklik, oyun konsolunun gelecekteki geliştirmeler ve iyileştirmeler için uygun bir platformdur.



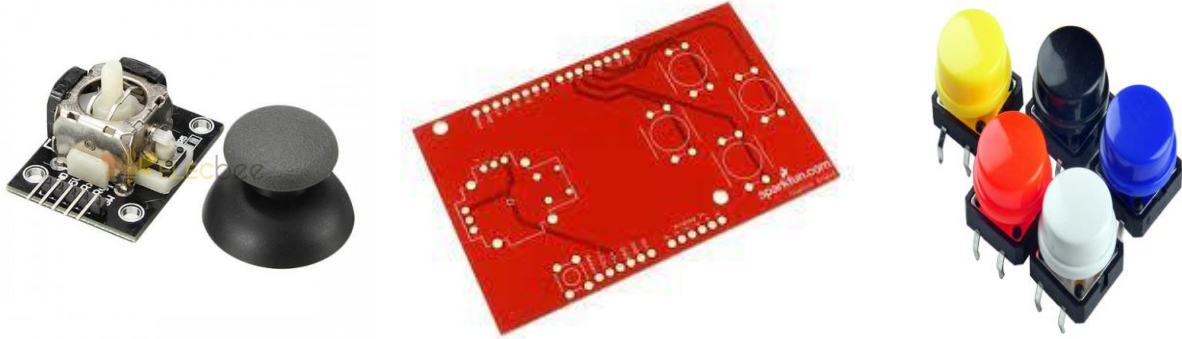
## 2. Funduino Joystick Shield:

Arduino tabanlı projelerde kolay ve hızlı bir şekilde joystick kontrolü sağlamak için tasarlanmış bir genişletme kartıdır. Oyun konsolu projelerinde kullanıcı girişlerini almak ve işlemek için ideal bir çözümdür. Funduino Joystick Shield, Arduino Leonardo ile birlikte kullanılarak oyun konsolunun kullanıcı girişlerini doğru ve hızlı bir şekilde algılamasını sağlamaktadır. Funduino Joystick Shield'in tercih edilme sebepleri ve kullanım alanları şu şekildedir:

- *Joystick Kontrolü:* Funduino Joystick Shield, üzerinde bulunan analog joystick ile kullanıcının yön ve hareket verilerini hassas bir şekilde algılayabilmektedir. Bu, oyun konsolumuzun temel işlevlerinden biridir ve kullanıcının bilgisayar oyunlarında yönlendirme yapabilmesini sağlamaktadır.
- *Butonlar, Ek Giriş ve Çıkışlar:* Shield üzerinde yer alan butonlar ve diğer giriş/çıkış birimleri, kullanıcı etkileşimini artırmakta ve çeşitli komutların verilmesini sağlamaktadır. Bu, oyun sırasında klavye işlevlerinin yerine getirilmesine imkan tanımaktadır.

Projede kullanılma sebepleri arasında Funduino Joystick Shield, Arduino Leonardo ile kolayca entegre edilebilmektedir. Bu özellik, projede hızlı ve sorunsuz bir kurulum sağlamaktadır.

Bu, kullanıcı deneyimini geliştirmekte ve oyun konsolunun kullanımını kolaylaştırmaktadır. Açık kaynaklı yapısı, geniş bir topluluk desteği sunmaktadır. Bu, proje geliştirme sürecinde karşılaşılan sorunların hızlı bir şekilde çözülmesini ve yeni özelliklerin eklenmesini kolaylaştırmaktadır.



### 3. Kodlar

```
#include "Keyboard.h"
#include "Mouse.h"
#define BUTTON_UP 2
#define BUTTON_RIGHT 3
#define BUTTON_DOWN 4
#define BUTTON_LEFT 5
#define BUTTON_E 6
#define BUTTON_F 7

int aKeyPressCount = 0;
int dKeyPressCount = 0;
bool spaceKeyPressed = false;
bool rKeyPressed = false;
int lastJoystickX = 0;
int lastJoystickY = 0;
const int deadzone = 3;
void setup() {
```

```

Keyboard.begin();
Mouse.begin();
Serial.begin(9600);
pinMode(BUTTON_UP, INPUT_PULLUP);
pinMode(BUTTON_RIGHT, INPUT_PULLUP);
pinMode(BUTTON_DOWN, INPUT_PULLUP);
pinMode(BUTTON_LEFT, INPUT_PULLUP);
pinMode(BUTTON_E, INPUT_PULLUP);
pinMode(BUTTON_F, INPUT_PULLUP);
}

void loop() {
    bool buttonUpState = (digitalRead(BUTTON_UP) == LOW);
    bool buttonRightState = (digitalRead(BUTTON_RIGHT) == LOW);
    bool buttonDownState = (digitalRead(BUTTON_DOWN) == LOW);
    bool buttonLeftState = (digitalRead(BUTTON_LEFT) == LOW);
    bool buttonEState = (digitalRead(BUTTON_E) == LOW);
    bool buttonFState = (digitalRead(BUTTON_F) == LOW);

    if (buttonUpState) {
        Keyboard.press('w');
    } else {
        Keyboard.release('w');
    }

    if (buttonRightState) {
        Keyboard.press('d');
        if (++dKeyPressCount == 2) {
            rKeyPressed = true;
            dKeyPressCount = 0;
        }
    }
}

```

```
    }  
  } else {  
    Keyboard.release('d');  
  }
```

```
if (buttonDownState) {  
  Keyboard.press('s');  
} else {  
  Keyboard.release('s');  
}
```

```
if (buttonLeftState) {  
  Keyboard.press('a');  
  if (++aKeyPressCount == 2) {  
    spaceKeyPressed = true;  
    aKeyPressCount = 0;  
  }  
} else {  
  Keyboard.release('a');  
}
```

```
if (buttonEState) {  
  Mouse.click(MOUSE_RIGHT);  
}
```

```
if (buttonFState) {  
  Mouse.click(MOUSE_LEFT);  
}
```

```
if (!buttonUpState && !buttonRightState && !buttonDownState && !buttonLeftState) {
```

```
if (spaceKeyPressed) {  
    Keyboard.write(' ');  
    spaceKeyPressed = false;  
}  
if (rKeyPressed) {  
    Keyboard.write('r');  
    rKeyPressed = false;  
}  
}
```

```
int joystickX = analogRead(A0);  
int joystickY = analogRead(A1);
```

```
joystickX = map(joystickX, 0, 1023, -20, 20);  
joystickY = map(joystickY, 0, 1023, -20, 20);
```

```
if (abs(joystickX) > deadzone || abs(joystickY) > deadzone) {  
    Mouse.move(joystickX, -joystickY, 0);  
    lastJoystickX = joystickX;  
    lastJoystickY = joystickY;  
} else {  
    Mouse.move(0, 0, 0);  
}
```

```
delay(10);  
}
```