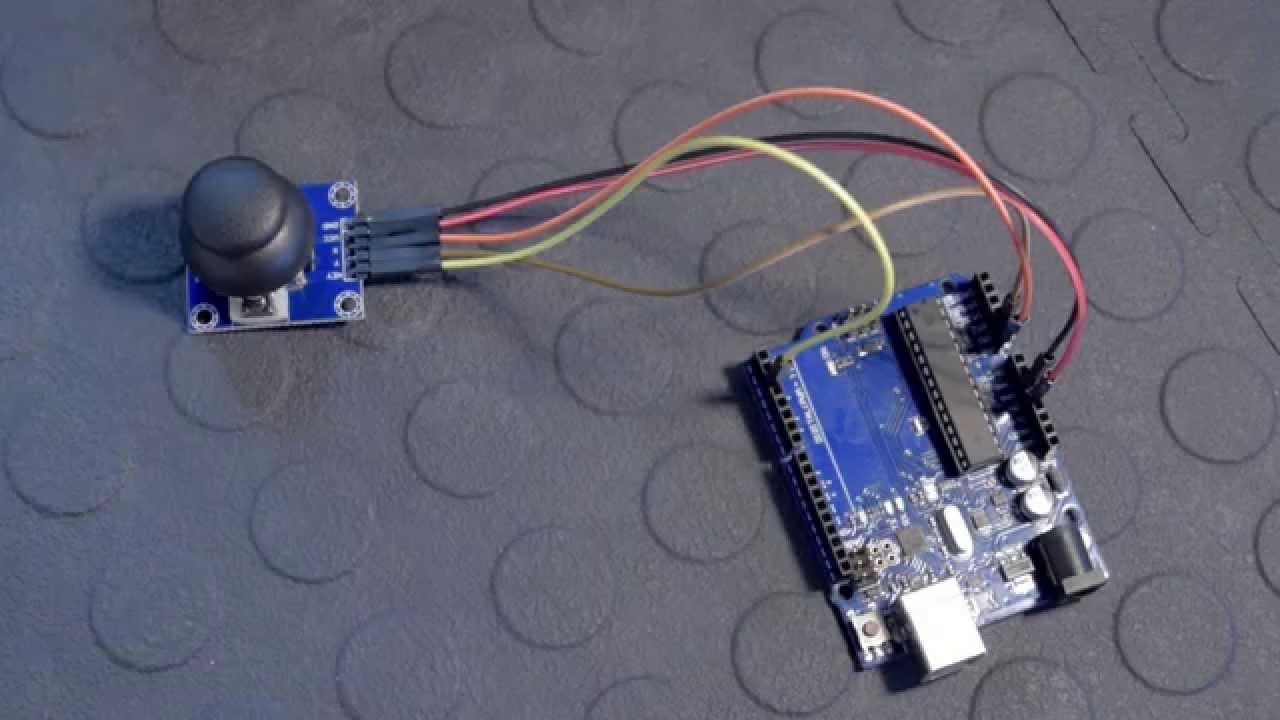
JOYSTİCK (ANALOG) YAPIMI

**Hazırlayanlar**

* **UTKU BAŞAR PAK**
* **BAHATTİN KARAGÜL**
* **OSMAN AYYILDIZ**
* **ENES TALHA ERGİN**



**Projenin Amacı**

Arduino ile joystick analog projesinin amacı, kullanıcının fiziksel hareketlerini dijital sinyallere çevirerek çeşitli elektronik cihazları kontrol edebilmektir. Bu projede, bir joystick'in iki eksenli (X ve Y) hareketleri, Arduino mikrodenetleyici tarafından analog sinyaller olarak algılanır ve işlenir. Arduino, bu sinyalleri kullanarak bağlı olan servo motorlar, LED'ler veya diğer çıkış cihazlarını kontrol eder. Örneğin, joystick'i sağa veya sola hareket ettirerek bir robotun yönünü değiştirebilir veya yukarı ve aşağı hareket ettirerek bir ışığın parlaklığını ayarlayabilirsiniz. Bu tür projeler, robotik kontrol, oyun konsolu arayüzleri veya kullanıcı dostu elektronik sistemler geliştirme gibi geniş bir uygulama yelpazesinde kullanılabilir.

**Kullanılan Bileşenler**

**JUMPER KABLOLAR**

**ANAHTAR**

**JOYSTİCK MODÜLLERİ**

**NOKİA 5110 EKRANI**

**ARDUİNO UNO**

**Projenin Çalışma Prensibi**

Arduino ile joystick analog projesinin çalışma prensibi, joystick'in hareketlerini algılayarak bunları dijital işlemlere dönüştürmeye dayanır. Joystick'in her iki ekseni (X ve Y), iki ayrı potansiyometre aracılığıyla hareketi tespit eder ve bu hareketi, değişen voltaj değerleri olarak analog sinyallere çevirir. Bu analog sinyaller, Arduino'nun analog giriş pinlerine gönderilir. Arduino, bu sinyalleri dijital veriler olarak okuyup işleyerek joystick'in yön ve hız bilgilerini belirler. Ardından, bu bilgileri kullanarak bağlı olan çıkış bileşenlerini (örneğin, bir LED'in parlaklığını ayarlamak veya bir servo motorun pozisyonunu değiştirmek gibi) kontrol eder. Böylece, joystick hareketleri doğrudan elektronik cihazların davranışını yönetir ve kullanıcıya fiziksel hareketlerle dijital kontrol imkanı sunar.

**Projenin Faydaları ve Uygulama Alanları**

### Faydaları:

* **Kullanıcı Dostu Kontrol**: Joystick'ler, basit ve sezgisel bir kullanıcı arayüzü sağlar, karmaşık sistemlerin kontrolünü kolaylaştırır.
* **Esneklik ve Uyarlanabilirlik**: Proje, çeşitli cihazların ve sistemlerin kontrol edilmesi için uyarlanabilir, böylece geniş bir yelpazede kullanım imkanı sunar.
* **Eğitim ve Öğrenme**: Elektronik ve programlama konusunda öğrenme ve öğretme sürecini destekler, temel kavramları pratik uygulamalarla pekiştirir.
* **Hızlı Prototipleme**: Projeler hızlı bir şekilde test edilip geliştirilebilir, yeni fikirlerin çabucak denenmesine olanak tanır.

### Uygulama Alanları:

* **Robotik ve Mekatronik**: Robotların ve mekatronik sistemlerin yönlendirilmesi ve kontrol edilmesi.
* **Oyun Konsolları**: Oyunlarda karakter veya araç kontrolü için joystick arayüzleri.
* **Yapay Uzuvlar ve Tıbbi Cihazlar**: Protez ve tıbbi cihazların kullanıcı tarafından hassas bir şekilde kontrol edilmesi.
* **Ev Otomasyonu**: Akıllı ev sistemlerinde ışıklar, perdeler veya diğer cihazların manuel kontrolü.
* **Sanat ve Etkileşimli Enstalasyonlar**: Dijital sanat eserlerinin ve interaktif enstalasyonların yönetimi ve manipülasyonu.

Bu proje, hem mühendislik hem de yaratıcı alanlarda geniş çapta faydalar ve uygulama fırsatları sunar.

ARDUİNO KODLARI

#include <LCD5110\_Graph.h>  //Ekran Kutuphanesi

#include <avr/pgmspace.h>  //PacMan oyunu Kutuphanesi

#define RST 3                //Ekran baglantilari

#define CE  4

#define DC  5

#define DIN 6

#define CLK 7

LCD5110 myGLCD(CLK, DIN, DC, RST, CE);  //Ekran Pinleri

extern uint8\_t SmallFont[];    //5110 ekraninda kullanacagimiz yazi fontunun kucuk olmasi

const uint8\_t pacman1[] PROGMEM={                   //Pixel koordinatlari

0x80, 0xE0, 0xF0, 0xF8, 0xFC, 0xFE, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x7E, 0x3E, 0x1C,

0x0C, 0x00, 0x00, 0x00, 0x1F, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF9,

0xF0, 0xE0, 0xC0, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x03, 0x07, 0x07, 0x0F,

0x0F, 0x0F, 0x0F, 0x0F, 0x0F, 0x07, 0x07, 0x03, 0x03, 0x00, 0x00, 0x00,

};

const uint8\_t pacman2[] PROGMEM={

0x80, 0xE0, 0xF0, 0xF8, 0xFC, 0xFE, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0xFE, 0x7C,

0x7C, 0x38, 0x20, 0x00, 0x1F, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xF9,

0xF9, 0xF0, 0xF0, 0xE0, 0xE0, 0xC0, 0x40, 0x00, 0x00, 0x00, 0x00, 0x01, 0x03, 0x07, 0x07, 0x0F,

0x0F, 0x0F, 0x0F, 0x0F, 0x0F, 0x07, 0x07, 0x03, 0x03, 0x01, 0x00, 0x00,

};

const uint8\_t pacman3[] PROGMEM={

0x80, 0xE0, 0xF0, 0xF8, 0xFC, 0xFE, 0xFE, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFE, 0xFE, 0xFC,

0xF8, 0xF0, 0xE0, 0x80, 0x1F, 0x7F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,

0xFF, 0xFF, 0xFF, 0xFF, 0xFB, 0xF9, 0x79, 0x19, 0x00, 0x00, 0x00, 0x01, 0x03, 0x07, 0x07, 0x0F,

0x0F, 0x0F, 0x0F, 0x0F, 0x0F, 0x07, 0x07, 0x03, 0x01, 0x00, 0x00, 0x00,

};

const uint8\_t pill[] PROGMEM={ //yem oldugu kisimi belirlemis

0x0E, 0x1F, 0x1F, 0x1F, 0x0E,

};

#define Width  84  //

#define Hight  48

#define Range  12

int FirstShotX , FirstShotY; //yakalama menzilini belirlemek

int PointX, PointY;

int delaytime;

void setup()

{

  /\*  graph kitapligini baslatma  \*/

  myGLCD.InitLCD();

  myGLCD.setFont(SmallFont);

  randomSeed(analogRead(0));

  /\* Joystick baglantilari \*/

  FirstShotX = analogRead(A0);

  FirstShotY = analogRead(A1);

  /\* Joystick merkezlemesi \*/

  PointX = 75;

  PointY = 20;

  delaytime = 50;

  myGLCD.print("- UTKUOYUNU -", CENTER, 0);

  myGLCD.print("PACMAN", CENTER, 20);

  myGLCD.print("Arduino", CENTER, 40);

  myGLCD.update();

  delay(3000);

  Serial.begin(115200);

}

void loop()

{

  uint8\_t\* bm;

  for (int i=-20; i<84; i++)

  {

    myGLCD.clrScr();

    // PacMan karakteri

    switch(((i+20)/3) % 4)

    {

      case 0: bm=pacman1;

              break;

      case 1: bm=pacman2;

              break;

      case 2: bm=pacman3;

              break;

      case 3: bm=pacman2;

              break;

    }

    int sensorValueX = (analogRead(A0) - FirstShotX)\*0.1 + 32;        //X ekseni aralik katsayisi

    int sensorValueY = (FirstShotY - analogRead(A1))\*0.06 + 14;       //Y ekseni aralik katsayisi

    myGLCD.drawBitmap(sensorValueX, sensorValueY, bm, 20, 20);

    // Yemi yedigimizde rastgele bir yere koymasi

TX:

    if((sensorValueX-5 <= PointX && PointX <= sensorValueX+15) && (sensorValueY-3 <= PointY && PointY <= sensorValueY + 20))

    {

       PointX = random(0, 80);

       PointY = random(0, 43);

       goto TX;

    }

    else

      myGLCD.drawBitmap(PointX, PointY, pill, 5, 5);

    int j, someInt, flag = 0;

    for(j=2; j<9; j++)

    {

      someInt = digitalRead(j);

      if(someInt == 0)

      {

        flag = 1;

        break;

      }

     }

     if(flag == 1)

     {

       Serial.println(i);

       switch(j)

       {

         case 2: Serial.println("--------> Button A");

                 delaytime = 50;

                 myGLCD.invert(true);

                 break;

         case 3: Serial.println("--------> Button B");

                 delaytime = 50;

                 myGLCD.invert(false);

                 break;

         default: break;

       }

       flag=0;

     }

    delay(delaytime);

    myGLCD.update();

}

,