

به نام خدا

عنوان آزمایش: نمایش کاراکترهای سفارشی بر روی LCD و کنترل سرعت انیمیشن با استفاده از پتانسیومتر و آردوینو.

هدف آزمایش: ایجاد و نمایش کاراکترهای سفارشی: یادگیری نحوه تعریف و نمایش کاراکترهای دلخواه (مانند قلب، چهره خندان و غیره) بر روی صفحه نمایش LCD.

کنترل سرعت انیمیشن با پتانسیومتر: یادگیری نحوه استفاده از پتانسیومتر برای خواندن مقادیر آنالوگ و تغییر سرعت انیمیشن نمایش داده شده روی LCD.

آشنایی با LCD کاراکتری: آشنایی با نحوه راه اندازی و استفاده از صفحه نمایش LCD کاراکتری ۱۶x2 با آردوینو.

شرح آزمایش

در این آزمایش، ما یک صفحه نمایش LCD کاراکتری را به آردوینو متصل می‌کنیم و از آن برای نمایش یک پیام ثابت و یک انیمیشن ساده استفاده می‌کنیم. انیمیشن ما شامل یک کاراکتر سفارشی است که دست‌های خود را بالا و پایین می‌برد و سرعت این انیمیشن با استفاده از یک پتانسیومتر کنترل می‌شود. با چرخاندن پتانسیومتر، سرعت انیمیشن افزایش یا کاهش می‌یابد.

توضیح کد:

:<LiquidCrystal.h>#include این خط، کتابخانه LiquidCrystal را برای کنترل LCD کاراکتری فراخوانی می‌کند.

:const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2; این خطوط، پین‌های اتصال LCD به آردوینو را تعریف می‌کنند.

:LiquidCrystal lcd(rs, en, d4, d5, d6, d7); یک شیء lcd از کلاس LiquidCrystal ایجاد می‌شود تا بتوان با LCD کار کرد.

:byte heart[8] = { ... }, byte smiley[8] = { ... }; این خطوط، آرایه‌هایی از بایت‌ها را تعریف می‌کنند که نمایانگر کاراکترهای سفارشی (قلب، چهره خندان و غیره) هستند. هر آرایه یک کاراکتر ۵x۸ پیکسلی را نشان می‌دهد.

:void setup() { ... } این تابع، یک بار در ابتدای اجرای برنامه اجرا می‌شود.

:lcd.begin(16, 2); این خط LCD را با ۱۶ ستون و ۲ ردیف راه اندازی می‌کند.

:lcd.createChar(0, heart);, lcd.createChar(1, smiley);, ... : این خطوط، کاراکترهای سفارشی را در حافظه CGRAM ال‌سی‌دی ذخیره می‌کنند و برای استفاده به آن‌ها یک شناسه (۰، ۱، ۲، ...) اختصاص می‌دهند.

:lcd.setCursor(0, 0); مکان نما را به موقعیت ۰، ۰ (بالا سمت چپ) منتقل می‌کند.

:lcd.print("I ");, lcd.write(byte(0));, lcd.print(" zahra ");, lcd.write((byte)1); این خطوط، یک پیام ثابت را به همراه کاراکترهای سفارشی قلب و چهره خندان روی LCD چاپ می‌کنند.

:void loop() { ... } این تابع، به طور مداوم پس از اجرای تابع setup تکرار می‌شود.

int sensorReading = analogRead(A0);
متغیر sensorReading ذخیره می‌کند. ولتاژ خوانده شده از پین A0 (پتانسیومتر) را می‌خواند و در

int delayTime = map(sensorReading, 0, 1023, 200, 1000);
تا ۱۰۲۳ به محدوده ۲۰۰ تا ۱۰۰۰ نگاشت می‌کند و در متغیر delayTime ذخیره می‌کند. این متغیر برای تعیین زمان تاخیر در انیمیشن استفاده خواهد شد.

lcd.setCursor(4, 1);
مکان نما را به موقعیت ۴,۱ (ردیف پایین، ستون ۵) منتقل می‌کند.

lcd.write(3);
اکتاراکتر سفارشی با شناسه ۳ (دست‌ها پایین) را روی LCD چاپ می‌کند.

delay(delayTime);
برنامه به مدت delayTime میلی ثانیه صبر می‌کند.

lcd.setCursor(4, 1);
مکان نما را مجدداً به موقعیت ۴,۱ منتقل می‌کند.

lcd.write(4);
اکتاراکتر سفارشی با شناسه ۴ (دست‌ها بالا) را روی LCD چاپ می‌کند.

delay(delayTime);
برنامه به مدت delayTime میلی ثانیه صبر می‌کند.

/ include the library code:

```
#include <LiquidCrystal.h>
```

```
// initialize the library by associating any needed LCD interface pin
```

```
// with the arduino pin number it is connected to
```

```
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
```

```
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

```
// make some custom characters:
```

```
byte heart[8] = {
```

```
0b00000,
```

```
0b01010,
```

```
0b11111,
```

```
0b11111,
```

```
0b11111,
```

```
0b01110,
```

```
0b00100,
```

```
0b10001};

byte armsDown[8] = {

    0b00100,

    0b01010,

    0b00100,

    0b00100,

    0b01110,

    0b10101,

    0b00100,

    0b01010};

byte armsUp[8] = {

    0b00100,

    0b01010,

    0b00100,

    0b10101,

    0b01110,

    0b00100,

    0b00100,

    0b01010

};

void setup() {

    lcd.begin(16, 2);

    // create a new character

    lcd.createChar(2, frownie);

    // create a new character

    lcd.createChar(3, armsDown);
```

```
// create a new character
lcd.createChar(4, armsUp);

// set the cursor to the top left

// Print a message to the lcd.}

void loop() {

    // read the potentiometer on A0:

    int sensorReading = analogRead(A0);

    // map the result to 200 - 1000:

    int delayTime = map(sensorReading, 0, 1023, 200, 1000);

    // set the cursor to the bottom row, 5th position:

    for (int i = 0; i <= 15; i++) {

        if (i % 2 == 0) {

            lcd.setCursor(i, 0);

            lcd.write(3);

            delay(300);

            lcd.clear();}

        else if (i % 2 != 0) {

            lcd.setCursor(i, 0);

            lcd.write(4);

            delay(300);

            lcd.clear();}}

    for (int i = 15; i >= 0; i--) {

        if (i % 2 == 0) {

            lcd.setCursor(i, 1);

            lcd.write(4);

            delay(300);

            lcd.clear();}}}
```

نحوه اتصال مدار:

پایه RS ال‌سی‌دی به پین ۱۲ آردوینو

پایه Enable ال‌سی‌دی به پین ۱۱ آردوینو

پایه D4 ال‌سی‌دی به پین ۵ آردوینو

پایه D5 ال‌سی‌دی به پین ۴ آردوینو

پایه D6 ال‌سی‌دی به پین ۳ آردوینو

پایه D7 ال‌سی‌دی به پین ۲ آردوینو

پایه VSS ال‌سی‌دی به GND آردوینو

پایه VDD ال‌سی‌دی به ۵V آردوینو

پایه V0 ال‌سی‌دی به وسط پتانسیومتر ۱۰ کیلو اهم (برای تنظیم کنتراست)

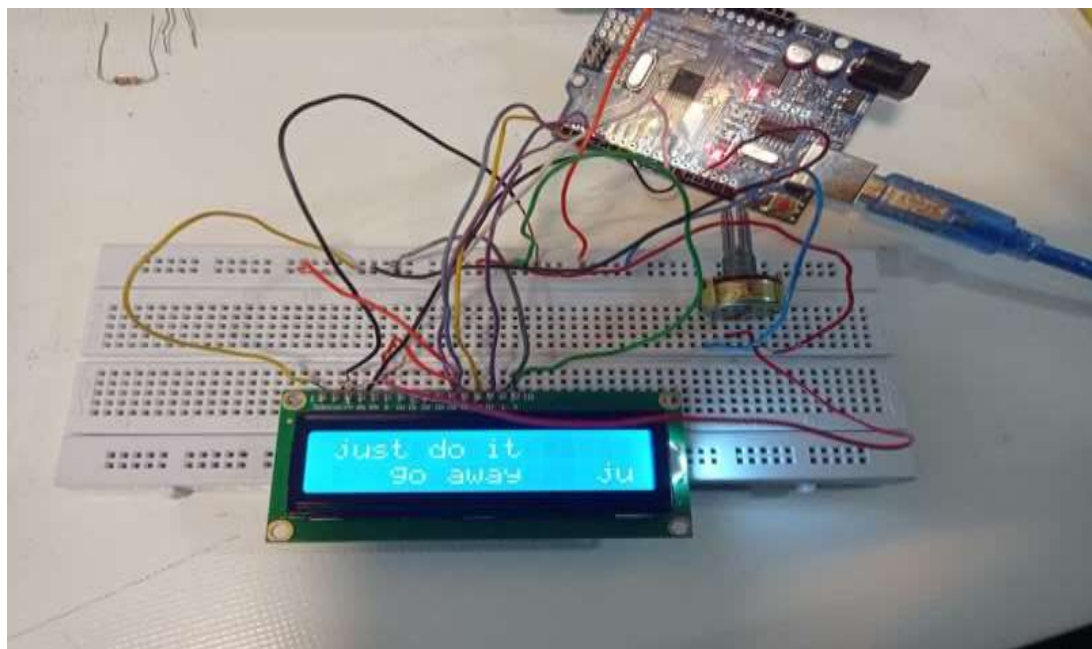
دو پایه کناری پتانسیومتر به ۵V و GND آردوینو

پتانسیومتر:

پایه وسط پتانسیومتر به پین A0 آردوینو

دو پایه کناری پتانسیومتر به ۵V و GND آردوینو

شماتیک مدار:



نتیجه گیری: در این آزمایش، با موفقیت توانستیم کاراکترهای سفارشی را بر روی LCD نمایش دهیم و با استفاده از پتانسیومتر، سرعت انیمیشن را کنترل کنیم. با چرخاندن پتانسیومتر، سرعت تغییر حالت کاراکتر (دست‌ها بالا و پایین) تغییر می‌کند.