نام گزارشکار درس ریز پردازنده جلسه دوم نام و نام خانوادگی:مریم گروهی

استاد: اقای دکتر عباسی

نام آزمایش : پتانسیومتر

وسایل آزمایش: برد بورد – ال ای دی - مقاومت - سیم جامپر - برد آردوینو اونو – کابل متصل شدن برد آؤردوینو به سیستم توضیحات آزمایش:پایه مثبت ال ای دی وصل میکنیم به پایه 6 انالوگ پایه منفیش به مقاومت و مقاومت هم به زمین و توسط کدی که زدیم ال ای دی از 0 تا 255 به صورت تدریجی روشن و خاموش میشود.

هدف ازمایش : کنترل شدت نور

این کد مربوط به برنامه‌نویسی آردوینو است و برای کنترل شدت نور یک LED با استفاده از یک پتانسیومتر و همچنین ارسال داده‌های خوانده شده به سریال مانیتور استفاده می‌شود. بیایید بخش‌های مختلف این کد را تحلیل کنیم:

تعریف متغیرها

cpp

int ledpin = 9;

int val = 0;

ledpin شماره پین دیجیتالی است که به LED متصل است و به عنوان خروجی برای تغییر شدت نور آن استفاده می‌شود.

val مقداری است که از پتانسیومتر خوانده می‌شود.

تابع setup

cpp

void setup() {

pinMode(ledpin, OUTPUT); // تنظیم پین 9 به عنوان خروجی

Serial.begin(9600); // آغاز ارتباط سریال با نرخ انتقال 9600 بیت در ثانیه

}

pinMode(ledpin, OUTPUT) پین متصل به LED را به عنوان خروجی تنظیم می‌کند تا امکان تغییر شدت نور آن فراهم شود.

Serial.begin(9600) ارتباط سریال را برای ارسال اطلاعات با نرخ انتقال 9600 بیت در ثانیه تنظیم می‌کند.

تابع loop

cpp

void loop() {

val = analogRead(A0); // خواندن مقدار آنالوگ از پین A0 متصل به پتانسیومتر

analogWrite(ledpin, val / 4); // نوشتن مقدار به صورت PWM روی پین LED

Serial.print(val); // چاپ مقدار خوانده شده روی سریال مانیتور

}

analogRead(A0) مقدار آنالوگ را از پین A0 که به پتانسیومتر متصل است، می‌خواند. این مقدار بین 0 تا 1023 خواهد بود.

analogWrite(ledpin, val / 4) از ویژگی PWM (تعدیل پهنای پالس) برای تنظیم شدت نور LED استفاده می‌کند. مقدار PWM باید بین 0 تا 255 باشد، بنابراین مقدار خوانده شده با تقسیم بر 4 به این بازه نگاشت می‌یابد.

Serial.print(val) مقدار خوانده شده از پتانسیومتر را به پورت سریال (و در نتیجه به کامپیوتر) می‌فرستد تا در سریال مانیتور مشاهده شود.

عملکرد کلی:

این کد شدت نور LED را بر اساس موقعیت پتانسیومتر تنظیم می‌کند. حرکت پتانسیومتر باعث تغییر مقدار ولتاژ آنالوگی می‌شود که از طریق A0 خوانده می‌شود. این مقدار سپس به شدت PWM تبدیل شده و به LED فرستاده می‌شود.

می‌توانید تغییرات مقدار پتانسیومتر را در سریال مانیتور آردوینو مشاهده کنید، که به شما نشان می‌دهد چه مقدار آنالوگی خوانده می‌شود.

این برنامه به زبان ساده کنترل رابطه بین یک پتانسیومتر و یک LED را امکان‌پذیر می‌کند و برای آموزش مفاهیم پایه‌ای آردوینو بسیار مفید است

نام گزارشکار درس ریز پردازنده جلسه دوم نام و نام خانوادگی:مریم گروهی

استاد: اقای دکتر عباسی

pwm نام آزمایش :

Led هدف ازمایش : افزایش و کاهش روشنایی

تحلیل کد:

تعریف متغیر led:

int led = 9;

این خط یک متغیر به نام led تعریف می‌کند که شماره پین متصل به LED را مشخص می‌کند (در اینجا پین 9).

تابع setup():

void setup() {

pinMode(led, INPUT);

}

در اینجا pinMode(led, INPUT); پین 9 را به عنوان ورودی تنظیم می‌کند. این حالت نادرست است اگر بخواهید LED را کنترل کنید، زیرا باید پین به عنوان خروجی تنظیم شود. بایستی به صورت زیر باشد:

pinMode(led, OUTPUT);

تابع loop():

void loop() {

for (int i = 0; i < 255; i = i + 5) {

analogWrite(led, i);

delay(50);

}

for (int i = 255; i >= 0; i = i - 5) {

analogWrite(led, i);

delay(50);

}

}

افزایش روشنایی LED: حلقه اول با استفاده از for از 0 تا 255 (در حالت افزایش) می‌دود. در هر دور، analogWrite(led, i); مقدار i به LED ارسال می‌شود که به تدریج روشنایی آن را افزایش می‌دهد. LED در هر مرحله به مدت 50 میلی‌ثانیه روشن می‌ماند.

کاهش روشنایی LED: حلقه دوم مشابه حلقه اول است، اما از 255 تا 0 (در حالت کاهش) می‌دود، و باعث کاهش روشنایی LED می‌شود.

خلاصه عملکرد کد:

کد به‌طور کلی LED متصل به پین 9 را با افزایشی 5 واحدی روشن می‌کند تا به حداکثر روشنایی (255) برسد، سپس روشنایی را به طور مشابه با کاهشی 5 واحدی تا زمانی که LED خاموش شود کاهش می‌دهد. این چرخه ادامه می‌یابد

نام گزارشکار درس ریز پردازنده جلسه چهارم نام و نام خانوادگی:مریم گروهی

استاد: اقای دکتر عباسی

نام آزمایش : ادمک

هدف ازمایش : نمایش کاراکتر دلخواه

این کد مربوط به استفاده از یک LCD گرافیکی با کتابخانه LiquidCrystal در آردوینو است که برای نمایش کاراکترهای دلخواه و تعامل با یک پتانسیومتر طراحی شده است. در ادامه، به طور مفصل توضیح می‌دهم که هر بخش از کد چه کاری انجام می‌دهد:

کتابخانه‌ها و پین‌ها

cpp

#include <LiquidCrystal.h>

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

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

در اینجا از کتابخانه LiquidCrystal برای کنترل LCD استفاده شده است.

پین‌های مختلف آردوینو برای ارتباط با LCD تعریف شده‌اند:

rs, en, d4, d5, d6, d7 پین‌های آردوینو هستند که به ورودی‌های مختلف LCD متصل شده‌اند.

شیء lcd یک نمونه از کلاس LiquidCrystal است که به آردوینو اجازه می‌دهد با LCD ارتباط برقرار کند.

تعریف کاراکترهای سفارشی

cpp

byte frownie[8] = {

0b00000,

0b00000,

0b01010,

0b00000,

0b00000,

0b00000,

0b01110,

0b10001

};

byte armsDown[8] = {

0b00100,

0b01010,

0b00100,

0b00100,

0b01110,

0b10101,

0b00100,

0b01010

};

byte armsUp[8] = {

0b00100,

0b01010,

0b00100,

0b10101,

0b01110,

0b00100,

0b00100,

0b01010

};

در این بخش، سه کاراکتر سفارشی تعریف شده‌اند که از 8 بایت برای نمایش یک تصویر گرافیکی روی LCD استفاده می‌کنند:

frownie: یک صورت غمگین.

armsDown: یک کاراکتر با دست‌های پایین.

armsUp: یک کاراکتر با دست‌های بالا.

این تصاویر گرافیکی با استفاده از آرایه‌های بایتی (هر بایت یک ردیف از پیکسل‌ها را نمایش می‌دهد) تعریف شده‌اند.

تنظیمات اولیه

cpp

void setup() {

lcd.begin(16, 2); // راه‌اندازی LCD با اندازه 16x2

lcd.createChar(2, frownie); // ایجاد کاراکتر صورت غمگین

lcd.createChar(3, armsDown); // ایجاد کاراکتر دست‌های پایین

lcd.createChar(4, armsUp); // ایجاد کاراکتر دست‌های بالا

}

lcd.begin(16, 2) تنظیم می‌کند که LCD دارای 16 ستون و 2 ردیف است.

سه کاراکتر سفارشی که قبلاً تعریف کرده‌ایم، با استفاده از lcd.createChar() در حافظه LCD ذخیره می‌شوند تا بتوانیم در کد از آن‌ها استفاده کنیم.

حلقه اصلی (Loop)

cpp

void loop() {

int sensorReading = analogRead(A0); // خواندن مقدار از پتانسیومتر

int delayTime = map(sensorReading, 0, 1023, 200, 1000); // تطبیق مقدار خوانده شده با تاخیر

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

if (i % 2 == 0) {

lcd.setCursor(i, 0);

lcd.write(3); // نمایش کاراکتر دست‌های پایین

delay(300);

lcd.clear(); // پاک کردن صفحه

}

else {

lcd.setCursor(i, 0);

lcd.write(4); // نمایش کاراکتر دست‌های بالا

delay(300);

lcd.clear();

}

}

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

if (i % 2 == 0) {

lcd.setCursor(i, 1);

lcd.write(4); // نمایش کاراکتر دست‌های بالا

delay(300);

lcd.clear();

}

else {

lcd.setCursor(i, 1);

lcd.write(3); // نمایش کاراکتر دست‌های پایین

delay(300);

lcd.clear();

}

}

}

خواندن از پتانسیومتر:

analogRead(A0) مقدار ورودی آنالوگ از پتانسیومتر متصل به پین A0 را می‌خواند. این مقدار بین 0 تا 1023 خواهد بود.

analogRead(A0) مقدار ورودی آنالوگ از پتانسیومتر متصل به پین A0 را می‌خواند. این مقدار بین 0 تا 1023 خواهد بود.

تطبیق مقدار پتانسیومتر با تاخیر:

map(sensorReading, 0, 1023, 200, 1000) مقدار خوانده شده از پتانسیومتر را به یک بازه جدید (بین 200 و 1000 میلی‌ثانیه) تبدیل می‌کند تا سرعت حرکت کاراکترها تغییر کند.

نمایش کاراکترها و حرکت آن‌ها:

در ابتدا، یک حلقه for برای حرکت کاراکترهای دست‌های پایین و دست‌های بالا در ردیف اول (0) LCD وجود دارد.

در هر بار تکرار، به طور متناوب (زوج و فرد بودن مقدار i)، کاراکترهای armsDown (دست‌های پایین) و armsUp (دست‌های بالا) در موقعیت‌های مختلف LCD نمایش داده می‌شود.

پس از نمایش هر کاراکتر، صفحه پاک می‌شود (lcd.clear()).

حرکت کاراکترها در جهت مخالف:

سپس، همان فرآیند در ردیف دوم (1) LCD انجام می‌شود، اما این بار کاراکترها از سمت راست به چپ حرکت می‌کنند.

نکات:

کاراکترهای سفارشی: شما می‌توانید با تغییر بیت‌های داخل آرایه‌های frownie, armsDown, و armsUp، شکل‌ها و تصاویر مختلفی را نمایش دهید.

مقدار پتانسیومتر: این مقدار در تغییر زمان تأخیر بین حرکت کاراکترها استفاده می‌شود. با چرخاندن پتانسیومتر، سرعت نمایش کاراکترها تغییر خواهد کرد.

نتیجه نهایی:

این کد یک انیمیشن ساده از حرکت کاراکترهای دست‌های بالا و پایین را بر روی یک LCD 16x2 نمایش می‌دهد. سرعت این حرکت به مقدار پتانسیومتر متصل به پین بستگی دارد A0.

نام گزارشکار درس ریز پردازنده جلسه چهارم نام و نام خانوادگی:مریم گروهی

استاد: اقای دکتر عباسی

نام آزمایش : تاس الکترونیکی

هدف ازمایش : برای ایجاد یک بازی ساده شبیه به تاس

این کد به زبان برنامه‌نویسی Arduino نوشته شده است و عملکرد آن به شرح زیر است:

تحلیل کد:

تعریف کلید:

int key = digitalRead(A0);

که نمایش‌دهنده وضعیت پینی است که به عنوان ورودی استفاده شده است.

تابع setup():

void setup() {

pinMode(A0, INPUT);

for (int i = 1; i <= 6; i++) {

pinMode(i, OUTPUT);

}

randomSeed(analogRead(A0));

Serial.begin(9600);

}

تنظیم ورودی و خروجی: پین A0 به عنوان ورودی و پین‌های 1 تا 6 به عنوان خروجی تنظیم می‌شوند.

آماده‌سازی برای تولید عدد تصادفی: randomSeed(analogRead(A0)); مقدار اولیه تصادفی را با خواندن از A0 تنظیم می‌کند. اینکار باعث تولید اعداد تصادفی متفاوت در هر بار ریست می‌شود.

تنظیم ارتباط سریال: Serial.begin(9600); ارتباط سریال را با نرخ 9600 بیت در ثانیه آغاز می‌کند.

تابع loop():

void loop() {

if (key == 0) {

int randNum = random(1, 7);

for (int i = 1; i <= 6; i++){

digitalWrite(i, LOW);

}

Serial.println(randNum);

digitalWrite(randNum, HIGH);

delay(1000);

}

}

بررسی وضعیت کلید: اگر key برابر با 0 باشد (یعنی وضعیت پین A0 LOW باشد)، وارد حلقه می‌شود.

تولید عدد تصادفی: random(1, 7); یک عدد تصادفی بین 1 تا 6 تولید می‌کند.

غیرفعال کردن تمام پین‌های خروجی: تمام پین‌های 1 تا 6 به وضعیت LOW تغییر می‌کنند.

ارسال عدد تصادفی: عدد تصادفی تولیدشده از طریق ارتباط سریال چاپ می‌شود.

فعال کردن یک پین: پین با شماره‌ی randNum به وضعیت HIGH تغییر می‌کند.

توقف برنامه: با delay(1000); برنامه به مدت 1 ثانیه متوقف می‌شود.

نکات: