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

استاد: اقای دکتر عباسی

نام آزمایش : پیانو

وسایل ازمایش: آردوینو (Arduino):

هر مدل آردوینو (مثل Arduino Uno) که دارای ورودی و خروجی دیجیتال باشد.

سنسورها یا کلیدها (Push Buttons):

تعداد ۴ کلید برای ورودی (برای A، B، C و D) که به ورودی‌های دیجیتال آردوینو متصل شوند.

بلندگو یا سرنگی (Buzzer):

برای تولید صدا که به پین Burr (پین 11) متصل می‌شود.

مقاومت‌ها:

برای استفاده در مدار کلیدها. معمولاً از مقاومت ۱۰ کیلو اهم برای مقاوم سازی ورودی‌های دیجیتال استفاده می‌شود.

سیم‌ها و بورد (Breadboard):

سیم‌ها برای اتصالات و بورد برای قرار دادن قطعات

هدف ازمایش: تولید صدا

توضیح کد: این کد برای یک میکروکنترلر آردوینو نوشته شده است و وظیفه آن تولید صداهای متفاوتی بر اساس ورودی‌های کلید یا سوئیچ‌های متصل به پورت‌های خاص است. به طور کلی، این کد به صورت زیر عمل می‌کند:

تعریف‌ها و مقادیر ثابت:

چند مقدار فرکانسی برای نت‌های موسیقی (C، D، E، F، G، A، B) توسط #define تعریف شده‌اند (اینها فرکانس‌های نت‌ها بر حسب هرتز هستند).

مقادیر عددی برای پایه‌های ورودی (A, B, C, D) و خروجی (Burr) تعریف شده‌اند.

تابع setup():

این تابع برای تنظیم تنظیمات اولیه پین‌ها استفاده می‌شود.

پین‌های مشخص شده با A, B, C, D به عنوان ورودی تعریف می‌شوند و با digitalWrite روی HIGH تنظیم می‌شوند که به معنای استفاده از مقاومت کششی داخلی آردوینو است.

تابع loop():

این حلقه‌ی اصلی برنامه است که به طور مکرر اجرا می‌شود.

برای هرپین ورودی (A, B, C, D)، بررسی می‌شود که آیا کلید متصل به آن پین فشار داده شده است (یعنی مقدار پین LOW شده است).

در صورت فشار داده شدن کلید:

تابع tone() فراخوانی می‌شود که فرکانس مربوط به کلید را در پین Burr که به یک بوق یا بلندگو متصل است، تولید می‌کند.

noTone(Burr) برای توقف تولید صدا زمانی که هیچ کلیدی فشار داده نشده است، استفاده می‌شود.

در نتیجه، این کد برای تولید صداهای متفاوت بر اساس فشار دادن کلیدهای متصل به آردوینو استفاده می‌شود. هر پایه‌ی ورودی به یک فرکانس خاص اختصاص داده شده، و بلندگو صدا را بر اساس آن تولید می‌کند

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

استاد: اقای دکتر عباسی

نام آزمایش : سنسور دما

هدف ازمایش: خواندن مقادیر از یک سنسور

وسایل ازمایش: آردوینو

هر مدل آردوینو (مثل Arduino Uno) که دارای ورودی‌های آنالوگ باشد.

سنسور آنالوگ:

می‌تواند هر نوع سنسور آنالوگ باشد (مثل سنسور نور، سنسور حرارتی، یا سنسور رطوبت) که خروجی آنالوگ تولید می‌کند. این سنسور باید به پین A0 متصل شود.

LED:

یک LED (دیود ساطع کننده نور) که به پین 13 متصل می‌شود.

مقاومت:

یک مقاومت (معمولاً 220Ω یا 330Ω) برای محدود کردن جریان LED. این مقاومت باید به LED متصل شود.

سیم‌ها و بورد (Breadboard):

سیم‌ها برای اتصالات و بورد برای قرار دادن قطعات

توضیح کد:

این کد برای پلتفرم آردوینو طراحی شده و هدف آن خواندن مقادیر از یک سنسور متصل به پین آنالوگ A0، نمایش آن در سریال مانیتور (Serial Monitor)، و کنترل یک LED متصل به پین دیجیتال 13 بر اساس آن مقادیر است. بیایید قسمت‌های مختلف کد را بررسی کنیم:

تعریف متغیرها:

sensor: برای نگهداری مقدار خوانده شده از سنسور.

led: شماره پین که LED به آن متصل است (پین 13).

تابع setup():

Serial.begin(9600): سرعت ارتباط سریال با سرعت 9600 بیت بر ثانیه را تنظیم می‌کند تا مقادیر در سریال مانیتور نشان داده شوند.

pinMode(led, OUTPUT): پین 13 را به عنوان خروجی تنظیم می‌کند تا بتواند LED را کنترل کند.

تابع loop():

sensor = analogRead(A0): مقدار آنالوگ از سنسور متصل به پین A0 خوانده می‌شود و در متغیر sensor ذخیره می‌شود.

Serial.print("sensor value="): متن “sensor value=” را به سریال مانیتور ارسال می‌کند.

Serial.println(sensor): مقدار خوانده شده از سنسور را به سریال مانیتور ارسال کرده و یک خط جدید شروع می‌کند.

delay(300): برنامه را به مدت 300 میلی‌ثانیه متوقف می‌کند تا خواندن‌ها خیلی سریع انجام نشوند.

شرایط if-else:

اگر مقدار سنسور بیشتر از 120 باشد، LED خاموش می‌شود (digitalWrite(led, LOW)).

در غیر این صورت، LED روشن می‌شود (digitalWrite(led, HIGH)).

با این ساختار، کد به طور مداوم مقدار سنسور را می‌خواند و اگر از یک مقدار خاص (120) بالاتر باشد، LED خاموش می‌شود و در غیر این صورت روشن باقی می‌ماند. به این ترتیب از LED می‌توان به عنوان یک ابزار بازخورد ساده بر اساس وضعیت سنسور استفاده کرد

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

استاد: اقای دکتر عباسی

نام آزمایش : فتوسل

هدف ازمایش: مقدار سیگنال را خوانده و سنسور نمایش میدهد

وسایل ازمایش: آردوینو

یک مدل از آردوینو مانند Arduino Uno.

نمایشگر 7 قسمتی (7-segment display):

برای نمایش اعداد 0 تا 9.

مقاومت‌ها:

برای هر بخش از نمایشگر. معمولاً از مقاومت‌های 220Ω یا 330Ω استفاده می‌شود.

سیم‌ها و بورد (Breadboard):

سیم‌ها برای اتصالات و بورد برای قرار دادن قطعات

:

توضیحات کد:

int sensor; // متغیر برای نگهداری مقدار خوانده شده از سنسور

int led = 13; // تعریف پین LED که به 13 متصل است

void setup() {

Serial.begin(9600); // شروع ارتباط سریال با سرعت 9600 بیت بر ثانیه

pinMode(led, OUTPUT); // تنظیم پین LED به عنوان خروجی

}

void loop() {

sensor = analogRead(A0); // خواندن مقدار آنالوگ از پین A0

Serial.print("sensor value="); // ارسال متن به سریال

Serial.println(sensor); // ارسال مقدار خوانده شده به سریال

delay(300); // تأخیر 300 میلی‌ثانیه

if(sensor > 120) { // اگر مقدار سنسور بزرگتر از 120 باشد

digitalWrite(led, LOW); // LED خاموش می‌شود

} else {

digitalWrite(led, HIGH); // LED روشن می‌شود

}

}

توضیحات کلیدهای کد:

تعریف متغیرها:

int sensor; برای ذخیره مقدار خوانده شده از سنسور استفاده می‌شود.

int led = 13; شماره پین LED مشخص می‌کند که به کدام پین آردوینو متصل است (در اینجا پین 13).

تابع setup():

Serial.begin(9600);: ارتباط سریال را با کامپیوتر در سرعت 9600 bps راه‌اندازی می‌کند. این امکان را فراهم می‌کند تا مقادیر را در سریال مانیتور مشاهده کنید.

pinMode(led, OUTPUT);: پین 13 را به عنوان خروجی تنظیم می‌کند تا بتواند LED را روشن یا خاموش کند.

تابع loop():

sensor = analogRead(A0);: مقدار آنالوگ از پین A0 خوانده می‌شود. این پین به سنسور متصل است.

Serial.print("sensor value=");: این خط یک متن برای مانیتور سریال می‌فرستد.

Serial.println(sensor);: مقدار دانلود شده را ارسال می‌کند و به خط جدید می‌رود.

delay(300);: تأخیر به مدت 300 میلی‌ثانیه بین خواندن‌ها.

سپس یک شرط if بررسی می‌کند:

اگر مقدار سنسور بیشتر از 120 باشد، LED را خاموش (digitalWrite(led, LOW);) می‌کند.

در غیر این صورت، LED را روشن (digitalWrite(led, HIGH);) می‌کند.

نکات مهم:

عملکرد کد: این کد به طور مداوم مقدار سیگنال خوانده شده از سنسور را نمایش می‌دهد و LED را بر اساس مقدار خوانده شده کنترل می‌کند. این طراحی می‌تواند در کاربردهای واقعی مانند سنسورهای نور یا سنسورهای دیگر استفاده شود.

تست و عیب‌یابی: اطمینان حاصل کنید که سنسور به درستی به پین A0 متصل شده است و مقدار اندازه‌گیری شده (سنسور) امیدوار است تا عملکرد صحیح کد را تضمین کند.

مقدار آستانه: مقدار 120 به عنوان یک آستانه برای خاموش و روشن کردن LED انتخاب شده است. می‌توانید این مقدار را بر اساس نیاز خود تغییر دهید

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

استاد: اقای دکتر عباسی

نام آزمایش : سون سگمنت

هدف ازمایش: نمایش اعداد1تا7 بر روی نمایشگر سون سگمنت

وسایل ازمایش: برد آردوینو

نمایشگر 7 قسمتی (7-Segment Display):

این نمایشگرها به‌صورت متداول در دو نوع آند مشترک (Common Anode) و کاتد مشترک (Common Cathode) عرضه می‌شوند. برنامه حاضر برای نوع خاصی طراحی نشده، اما می‌توانید با تنظیمات مناسب آن را به هر دو متصل کنید.

مقاومت‌ها:

معمولاً از مقاومت‌های 220Ω تا 330Ω برای محدود کردن جریان عبوری از LEDهای نمایشگر استفاده می‌شود که به جلوگیری از آسیب دیدن LEDها کمک می‌کند. برای هر بخش از نمایشگر، یک مقاومت به‌صورت سری وصل می‌کنید.

سیم‌های جامپر (Jumper Wires):

برای برقراری اتصالات بین قطع مختلف (آردوینو و نمایشگر) استفاده می‌شود.

بورد تست (Breadboard):

برای نصب قطعات و برقراری اتصالات بدون لحیم‌کاری مناسب است

توضیح کد: در کد آردوینو که شما ارائه داده‌اید، هدف آن نمایش اعداد 0 تا 9 بر روی یک نمایشگر هفت‌سگمنته است. بیایید عملکرد کد را مرحله به مرحله بررسی کنیم و نکات قابل توجه را توضیح دهیم.

توضیحات کد:

cpp

int sequence[][7] = {

{1, 0, 0, 0, 1, 1, 1}, // 0

{1, 1, 1, 0, 1, 1, 1}, // 1

{1, 0, 0, 1, 1, 1, 1}, // 2

{0, 1, 1, 0, 1, 1, 1}, // 3

{1, 0, 0, 1, 1, 1, 0}, // 4

{0, 0, 0, 1, 1, 1, 0}, // 5

{1, 1, 0, 0, 1, 1, 1}, // 6

};

void printNumber(int number) {

for (int i = 0; i < 7; i++) {

digitalWrite(1 + i, sequence[number][i]);

}

}

void setup() {

pinMode(7, OUTPUT);

pinMode(6, OUTPUT);

pinMode(5, OUTPUT);

pinMode(4, OUTPUT);

pinMode(3, OUTPUT);

pinMode(2, OUTPUT);

pinMode(1, OUTPUT);

}

void loop() {

for (int i = 0; i < 10; i++) {

printNumber(i);

delay(1000);

}

}

توضیحات کلیدی:

آرایه sequence:

این آرایه دو بعدی شامل الگوهای بیتی برای نمایش اعداد 0 تا 6 بر روی یک نمایشگر هفت‌سگمنته است. به یمن توالی داده شده برای هر عدد، برای هر سگمنت اطلاعاتی وجود دارد که مشخص می‌کند آیا آن سگمنت روشن (1) یا خاموش (0) باشد.

برای مثال، برای عدد 0، سگمنت‌های 1، 4، 5 و 6 روشن هستند و بقیه خاموش.

تابع printNumber(int number):

این تابع برای نمایش یک عدد بر روی نمایشگر طراحی شده است. با استفاده از یک حلقه، وضعیت سگمنت‌های مربوط به عدد ورودی (از آرایه sequence) تنظیم می‌شود.

برای هر عدد ورودی، هر پین آردوینو از 1 تا 7 با توجه به وضعیت مربوط به آن عدد روشن یا خاموش می‌شود.

تابع setup():

در این تابع، مشخص می‌شود که پین‌های 1 تا 7 به عنوان خروجی تنظیم می‌شوند. این پین‌ها به سگمنت‌های مختلف نمایشگر 7 سگمنته متصل هستند.

تابع loop():

در حلقه loop()، یک حلقه for از 0 تا 9 وجود دارد که از تابع printNumber برای نمایش هر عدد استفاده می‌کند.

با یک تأخیر 1000 میلی‌ثانیه بعد از هر نمایش، عدد بعدی بر روی نمایشگر نشان داده می‌شود.

نکات مهم:

محدودیت نمایش اعداد: توجه داشته باشید که آرایه sequence فقط اعداد 0 تا 6 را شامل می‌شود. در نتیجه، کد در حال حاضر برای اعداد 7، 8 و 9 کار نخواهد کرد. برای پشتیبانی از اعداد 7 تا 9 باید ردیف‌های بیشتری به آرایه sequence اضافه کنید.

تنظیمات سخت‌افزاری: اطمینان حاصل کنید که پین‌های آردوینو به درستی به سگمنت‌های مناسب در نمایشگر 7 سگمنته متصل شده‌اند. تمرکز بر روی اتصالات صحیح به جلوگیری از مشکل در نمایش عدد کمک می‌کند.

توجه به مقادیر digitalWrite: در digitalWrite(1 + i, sequence[number][i]);، با فرض اینکه پین‌های آردوینو از 1 تا 7 هستند، توجه کنید که 1 + i موجب انتخاب پین 1، 2 تا 7 می‌شود.

اگر در نظر دارید اعدادی که نمایش داده می‌شوند افزایش دهید (حتی تا عدد 9)، آرایه sequence باید به‌روزرسانی شود و برای هر عدد جدید الگوی بیتی مناسب تعیین گردد

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

استاد: اقای دکتر عباسی

نام آزمایش : فتوسل

هدف ازمایش: مقدار سیگنال را خوانده و سنسور نمایش میدهد

وسایل ازمایش: آردوینو

یک مدل از آردوینو Arduino Uno.

نمایشگر 7 قسمتی (7-segment display):

برای نمایش اعداد 0 تا 9.

مقاومت‌ها:

برای هر بخش از نمایشگر. معمولاً از مقاومت‌های 220Ω یا 330Ω استفاده می‌شود.

سیم‌ها و بورد (Breadboard):

سیم‌ها برای اتصالات و بورد برای قرار دادن قطعات

توضیح کد: این کد برای کنترل یک نمایشگر 7 سگمنت با استفاده از آردوینو طراحی شده است. هدف آن نمایش یک سری ارقام (احتمالاً 0 تا 6) بر روی نمایشگر 7 سگمنت است. بیایید قسمت‌های مختلف کد را بررسی کنیم:

تعریف آرایه دوبعدی sequence:

این آرایه مجموعه‌ای از 7 بیت برای هر عدد مشخص را نگه می‌دارد. هر بیت نشان‌دهنده روشن یا خاموش بودن بخشی از نمایشگر 7 سگمنت است.

هر سطر در آرایه به یک عدد خاص از 0 تا 6 مربوط می‌شود.

تابع printNumber(int number):

این تابع برای تنظیم پین های دیجیتال که به بخش‌های نمایشگر متصل هستند و از آرایه sequence استفاده می‌کند تا پیکربندی لازم برای نمایش عدد مشخص را ارسال کند.

حلقه for درون تابع، هر یک از 7 قطعه را تنظیم می‌کند تا به ترتیب آن‌ها را روشن یا خاموش کند (با استفاده از digitalWrite).

تابع setup():

هر یک از پین‌های 1 تا 7 به عنوان خروجی تنظیم می‌شوند، که نشان می‌دهد این پین‌ها برای کنترل بخش‌های نمایشگر استفاده خواهند شد.

تابع loop():

یک حلقه for درون تابع loop() طراحی شده است که اعداد از 0 تا 9 را چاپ می‌کند، اما در حقیقت فقط می‌تواند اعداد 0 تا 6 را پشتیبانی کند زیرا آرایه sequence فقط برای این اعداد تنظیم شده است.

هر عدد پس از نمایش برای یک ثانیه (1000 میلی‌ثانیه) بر روی نمایشگر باقی می‌ماند.

نکته:

این کد فعلی بیشتر از 6 عدد در sequence ندارد، بنابراین تلاش برای نمایش اعدادی چون 7، 8، و 9 کاری غیر منطقی با این تعریف می‌باشد مگر اینکه تعداد ردیف‌های sequence افزایش یابد و نمایش برای این اعداد تعریف شود.

اطمینان حاصل شود که پین‌های 1 تا 7 به درستی به بخش‌های مناسب در نمایشگر 7 سگمنت متصل شده‌اند، همچنین توجه به اختلافات میان آردوینو و مدل‌های مختلف نمایشگر که می‌تواند باعث اختلاف در نتیجه شود نیز مهم است