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

استاد: اقای دکتر عباسی

نام آزمایش : میکروکنترلر

هدف ازمایش: مدار الکترونیکی را با استفاده از یک میکروکنترلر اردینو کنترل میکند

وسایل ازمایش: آردوینو

سنسور دما و رطوبت : این سنسور برای اندازه‌گیری دما و رطوبت استفاده می‌شود (DHT11)

نخ‌های اتصال و پروتوتایپ شیلد: برای متصل کردن سنسور به برد آردوینو.

منبع تغذیه: برای تأمین برق به آردوینو و دیگر دستگاه‌ها.

کامپیوتر: برای بارگذاری کد به آردوینو و مشاهده داده‌ها در سریال مانیتور

توضیح کد:

این کد به زبان برنامه‌نویسی آردوینو نوشته شده و یک مدار الکترونیکی را با استفاده از یک میکروکنترلر آردوینو کنترل می‌کند. بیایید این کد را مرحله به مرحله بررسی کنیم:

تعریف متغیر‌ها:

const int sw = 2; این خط نشان می‌دهد که یک کلید (سوئیچ) به پین شماره 2 آردوینو متصل است.

const int x = 0; و const int y = 1; نشان‌دهنده این هستند که دو محور ایکس و ایگرگ یک جوی‌استیک (یا سنسور دیگر) به پین‌های آنالوگ 0 و 1 متصل‌اند.

تابع setup():

pinMode(sw , INPUT\_PULLUP); پین شماره 2 را به عنوان ورودی با مقاومت کششی داخلی تعریف می‌کند. این باعث می‌شود که پین در وضعیت معمولاً بالا (HIGH) قرار بگیرد تا در صورت فشرده شدن کلید، به وضعیت پایین (LOW) تغییر کند.

Serial.begin(9600); این دستور سریال مانیتور را با نرخ انتقال 9600 بیت بر ثانیه آغاز می‌کند، که برای نمایش اطلاعات در سریال مانیتور نرم‌افزار آردوینو استفاده می‌شود.

تابع loop():

کد درون این تابع به صورت پیوسته اجرا می‌شود.

Serial.print("switch : "); و Serial.print(digitalRead(sw));: این دو خط وضعیت سوئیچ را می‌خوانند و در سریال مانیتور نمایش می‌دهند؛ 0 یا 1 بسته به وضعیت فشرده یا آزاد بودن کلید.

Serial.print(analogRead(x)); و Serial.print(analogRead(y));: این دستورات مقادیر خوانده شده از پین‌های آنالوگ 0 و 1 را می‌خوانند و مقادیر جوی‌استیک یا سنسورهای متصل را نمایش می‌دهند.

Serial.println(analogRead(y));: به طور خاص وضعیت محور y را همراه با چاپ خط جدید نمایش می‌دهد.

delay(500); برنامه را به مدت 500 میلی‌ثانیه متوقف می‌کند؛ یعنی نیم ثانیه میان هر سری از نمایش اطلاعات در سریال مانیتور تاخیر ایجاد می‌شود.

این کد وضعیت یک کلید و مقادیر آنالوگ از دو سنسور را به صورت دوره‌ای می‌خواند و در سریال مانیتور آردوینو چاپ می‌کند.

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

استاد: اقای دکتر عباسی

نام آزمایش : جوی استیک

هدف ازمایش: خواندن وضعیت سوئیچ

وسایل ازمایش: آردوینو- ماژول جوی استیک- نخ‌های اتصال- منبع تغذیه- سیم رابط

توضیح کد:

این کد آردوینو برای خواندن وضعیت یک سوئیچ و مقادیر دو محور از یک جوی‌استیک یا سنسور آنالوگ طراحی شده است. بیایید خطوط کد را به صورت جزئی توضیح دهیم:

تعریف متغیر‌ها:

const int sw = 2; این خط نشان می‌دهد که سوئیچ به پین دیجیتال 2 متصل است.

const int x = A0; و const int y = A1; نشان می‌دهند که محورهای X و Y جوی‌استیک به پین‌های آنالوگ A0 و A1 آردوینو متصل‌اند.

تابع setup():

Serial.begin(9600); برای آغاز ارتباط سریال با نرخ انتقال 9600 بیت بر ثانیه. این برای ارسال داده به سریال مانیتور استفاده می‌شود.

pinMode(y , INPUT); و pinMode(x , INPUT); تنظیم هر دو پین x و y به‌عنوان ورودی.

تابع loop():

Serial.print("switch : "); و Serial.print(digitalRead(sw));: این خطوط وضعیت سوئیچ را می‌خوانند و آن را در سریال مانیتور چاپ می‌کنند. چون sw به‌عنوان ورودی نه به‌طور پیش‌فرض و نه به‌طور آشکارا دارای مقاومت کششی نیست، احتمالاً باید به نحوی به‌درستی مدیریت شود.

Serial.print("VRx : "); و Serial.print(analogRead(x)); مقادیر مقادیر آنالوگ محور x را می‌خوانند و نمایش می‌دهند.

Serial.print("VRy : "); و Serial.println(analogRead(y)); مقادیر محور y را می‌خوانند و عبارت چاپی را با newline تکمیل می‌کنند.

delay(500); برای افزودن تاخیری به مدت 500 میلی‌ثانیه (نیم ثانیه) بین هر چرخه.

شرط‌های کنترل جهت:

if (analogRead(y) > 1000): اگر مقدار خوانده شده از محور y بیشتر از 1000 باشد، “right” (به سمت راست) در سریال مانیتور چاپ می‌شود.

if (analogRead(y) < 200): اگر مقدار خوانده شده از محور y کمتر از 200 باشد، “left” (به سمت چپ) چاپ می‌شود.

این به شما امکان می‌دهد تا زمانی که جوی‌استیک یا سنسور به سمت چپ یا راست حرکت داده می‌شود، به راحتی بفهمید کجا حرکت کرده است. مقادیر 1000 و 200 به عنوان حدود برای تشخیص جهت خاص y انتخاب شده‌اند.

استاد: اقای دکتر عباسی

نام آزمایش : جوی استیک

هدف ازمایش: خواندن وضعیت سوئیچ

وسایل ازمایش: آردوینو- ماژول جوی استیک- نخ‌های اتصال- منبع تغذیه- سیم رابط

توضیح کد:

این کد به زبان برنامه‌نویسی آردوینو نوشته شده و برای کنترل دما و رطوبت محیط به کمک سنسور DHT11 استفاده می‌شود. همچنین، بر اساس دمای محیط، یک سیستم سرمایش و گرمایش را کنترل می‌کند. بیایید قدم به قدم کد را مرور کنیم:

کتابخانه و تعریف‌ها

#include <DHT.h>: این خط کتابخانه‌ی DHT را برای کار با سنسورهای DHT11 یا DHT22 اضافه می‌کند.

#define DHTPIN 8 و #define DHTTYPE DHT11: پین متصل به سنسور DHT را مشخص می‌کند (پین 8)، و نوع سنسور استفاده شده DHT11 می‌باشد.

DHT dht(DHTPIN, DHTTYPE);: یک شیء dht از کلاس DHT ایجاد می‌کند که برای تعامل با سنسور DHT11 استفاده خواهد شد.

int cooler=6; و int heater=7;: مشخص می‌کند که دستگاه‌های خنک‌کننده و گرم‌کننده به ترتیب به پین‌های دیجیتال 6 و 7 متصل هستند.

تابع setup()

Serial.begin(9600);: آغاز ارتباط سریال با نرخ 9600 بیت بر ثانیه برای نمایش اطلاعات روی سریال مانیتور.

dht.begin();: آماده‌سازی سنسور DHT برای شروع به کار.

pinMode(cooler, OUTPUT); و pinMode(heater, OUTPUT);: تنظیم پین‌های سیستم سرمایش و گرمایش به عنوان خروجی.

تابع loop()

این تابع به طور مداوم اجرا می‌شود و وظایف زیر را به ترتیب انجام می‌دهد:

خواندن و نمایش رطوبت:

float humid = dht.readHumidity();: رطوبت محیطی را از سنسور می‌خواند.

این مقدار به صورت درصد در سریال مانیتور نمایش داده می‌شود.

خواندن و نمایش دما:

float temp = dht.readTemperature();: دمای محیطی را از سنسور می‌خواند.

این مقدار به درجه سانتی‌گراد در سریال مانیتور نمایش داده می‌شود.

کنترل سیستم سرمایش و گرمایش:

اگر دما بیشتر از 28 درجه سانتی‌گراد باشد (if(temp > 28))، پین مربوط به خنک‌کننده (پین 6) فعال (HIGH) و پین مربوط به گرم‌کننده (پین 7) غیرفعال (LOW) می‌شود.

اگر دما کمتر از 20 درجه سانتی‌گراد باشد (if(temp < 20))، پین مربوط به گرم‌کننده فعال و پین مربوط به خنک‌کننده غیرفعال می‌شود.

تاخیر:

delay(500);: تأخیر 500 میلی‌ثانیه ای بین نمایش اطلاعات و اقدامات کنترلی برای کاهش ازدیاد اطلاعات و اقدامات پی در پی.

بهینه‌سازی‌هایی که می‌توان انجام داد:

مطمئن شوید که سنسور DHT برای شرایط دمایی و رطوبت خاص به درستی کالیبره شده باشد.

بررسی کنید که سنسور در فاصله معقولی از دستگاه‌های کنترل دما و رطوبت قرار داشته باشد تا نتایج دقیق‌تری بدهد.

ممکن است بخواهید بین شرایط if مربوط به دما و else if استفاده کنید تا هر دو شرایط هم‌زمان پردازش نشوند و بهینه‌تر باشد.

این کد به طور خودکار دما را کنترل می‌کند، که برای سیستم‌های ساده مانند تهویه‌کننده هوا یا بخاری استفاده می‌شود.

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

استاد: اقای دکتر عباسی

نام آزمایش : میکروکنترلر

هدف ازمایش: مدار الکترونیکی را با استفاده از یک میکروکنترلر اردینو کنترل میکند

وسایل ازمایش: آردوینو

سنسور دما و رطوبت : این سنسور برای اندازه‌گیری دما و رطوبت استفاده می‌شود (DHT11)

نخ‌های اتصال و پروتوتایپ شیلد: برای متصل کردن سنسور به برد آردوینو.

منبع تغذیه: برای تأمین برق به آردوینو و دیگر دستگاه‌ها.

کامپیوتر: برای بارگذاری کد به آردوینو و مشاهده داده‌ها در سریال مانیتور

توضیح کد:

این کد به زبان برنامه‌نویسی آردوینو نوشته شده و برای کنترل دما و رطوبت محیط به کمک سنسور DHT11 استفاده می‌شود. همچنین، بر اساس دمای محیط، یک سیستم سرمایش و گرمایش را کنترل می‌کند. بیایید قدم به قدم کد را مرور کنیم:

کتابخانه و تعریف‌ها

#include <DHT.h>: این خط کتابخانه‌ی DHT را برای کار با سنسورهای DHT11 یا DHT22 اضافه می‌کند.

#define DHTPIN 8 و #define DHTTYPE DHT11: پین متصل به سنسور DHT را مشخص می‌کند (پین 8)، و نوع سنسور استفاده شده DHT11 می‌باشد.

DHT dht(DHTPIN, DHTTYPE);: یک شیء dht از کلاس DHT ایجاد می‌کند که برای تعامل با سنسور DHT11 استفاده خواهد شد.

int cooler=6; و int heater=7;: مشخص می‌کند که دستگاه‌های خنک‌کننده و گرم‌کننده به ترتیب به پین‌های دیجیتال 6 و 7 متصل هستند.

تابع setup()

Serial.begin(9600);: آغاز ارتباط سریال با نرخ 9600 بیت بر ثانیه برای نمایش اطلاعات روی سریال مانیتور.

dht.begin();: آماده‌سازی سنسور DHT برای شروع به کار.

pinMode(cooler, OUTPUT); و pinMode(heater, OUTPUT);: تنظیم پین‌های سیستم سرمایش و گرمایش به عنوان خروجی.

تابع loop()

این تابع به طور مداوم اجرا می‌شود و وظایف زیر را به ترتیب انجام می‌دهد:

خواندن و نمایش رطوبت:

float humid = dht.readHumidity();: رطوبت محیطی را از سنسور می‌خواند.

این مقدار به صورت درصد در سریال مانیتور نمایش داده می‌شود.

خواندن و نمایش دما:

float temp = dht.readTemperature();: دمای محیطی را از سنسور می‌خواند.

این مقدار به درجه سانتی‌گراد در سریال مانیتور نمایش داده می‌شود.

کنترل سیستم سرمایش و گرمایش:

اگر دما بیشتر از 28 درجه سانتی‌گراد باشد (if(temp > 28))، پین مربوط به خنک‌کننده (پین 6) فعال (HIGH) و پین مربوط به گرم‌کننده (پین 7) غیرفعال (LOW) می‌شود.

اگر دما کمتر از 20 درجه سانتی‌گراد باشد (if(temp < 20))، پین مربوط به گرم‌کننده فعال و پین مربوط به خنک‌کننده غیرفعال می‌شود.

تاخیر:

delay(500);: تأخیر 500 میلی‌ثانیه ای بین نمایش اطلاعات و اقدامات کنترلی برای کاهش ازدیاد اطلاعات و اقدامات پی در پی.

بهینه‌سازی‌هایی که می‌توان انجام داد:

مطمئن شوید که سنسور DHT برای شرایط دمایی و رطوبت خاص به درستی کالیبره شده باشد.

بررسی کنید که سنسور در فاصله معقولی از دستگاه‌های کنترل دما و رطوبت قرار داشته باشد تا نتایج دقیق‌تری بدهد.

ممکن است بخواهید بین شرایط if مربوط به دما و else if استفاده کنید تا هر دو شرایط هم‌زمان پردازش نشوند و بهینه‌تر باشد.

این کد به طور خودکار دما را کنترل می‌کند، که برای سیستم‌های ساده مانند تهویه‌کننده هوا یا بخاری استفاده می‌شود.