

سنسور دما

سنسورهای DHTxx دارای دو مدل هستند:

DHT11

DHT22

- این دو مدل تا حدودی شبیه هم به نظر می‌رسند و پایه‌های خروجی یکسانی دارند، اما مشخصات آن‌ها با یکدیگر متفاوت است. در اینجا به مشخصات این دو مدل می‌پردازیم:
- مدل DHT22 گران‌تر است و با توجه به قیمت گران‌تر آن، مسلماً قابلیت‌های بهتری دارد. رنج دمایی این سنسور از 40°C تا 125°C و دقت آن $\pm 0.5^{\circ}\text{C}$ است.
- در حالی که رنج دمایی مدل DHT11 از 0°C تا 50°C و دقت آن $\pm 2^{\circ}\text{C}$ است. به علاوه سنسور DHT22، رنج اندازه‌گیری رطوبت بیشتری دارد. رنج این مدل از 0٪ تا 100٪ و دقت آن 2٪ تا 5٪ است، در صورتی که رنج رطوبت قابل اندازه‌گیری برای سنسور DHT11 از 20٪ تا 80٪ و دقت آن 5٪ می‌باشد. بنابراین سنسور DHT22 علاوه بر رنج دمایی، رنج رطوبت گسترده‌تری دارد.

مقایسه دو سنسور دما

DHT22	DHT11	
3V-5V	3V-5V	ولتاژ عملکرد
2.5mA	2.5mA	ماکزیمم جریان عملکرد
0-100% / 2-5%	20-80% / 5%	رنج رطوبت
40-80°C / ± 0.5°C-	0-50°C / ± 2°C	رنج دما
(خواندن در دو ثانیه) 0.5Hz	(خواندن در هر ثانیه) 1Hz	نرخ نمونه برداری
15.1mm x 25mm x 7.7mm	15.5mm x 12mm x 5.5mm	ابعاد بدنه
دقت بالا	قیمت فوق العاده کم	مزیت

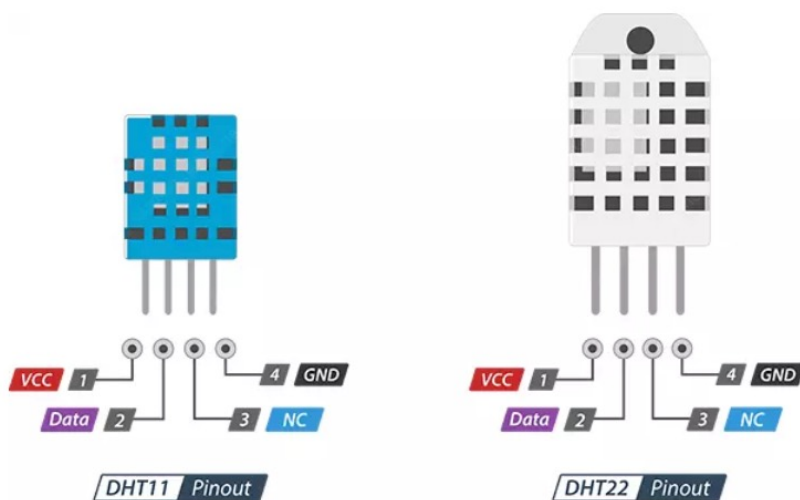
پایه های خروجی سنسور دما

VCC: این پین، ولتاژ تغذیه سنسور را فراهم می کند. اگر چه ولتاژ تغذیه در رنج 3.3V تا 5.5V می باشد، اما توصیه شده است که از ولتاژ 5V استفاده شود. در صورتی که ولتاژ تغذیه شما 5V است، می توانید سنسور را در فاصله 20 متری قرار دهید. اما اگر ولتاژ تغذیه شما 3.3V است، طول کابل نباید بیش از 1 متر باشد. در غیر این صورت ولتاژ افت می کند و باعث بروز خطا در اندازه گیری می شود.

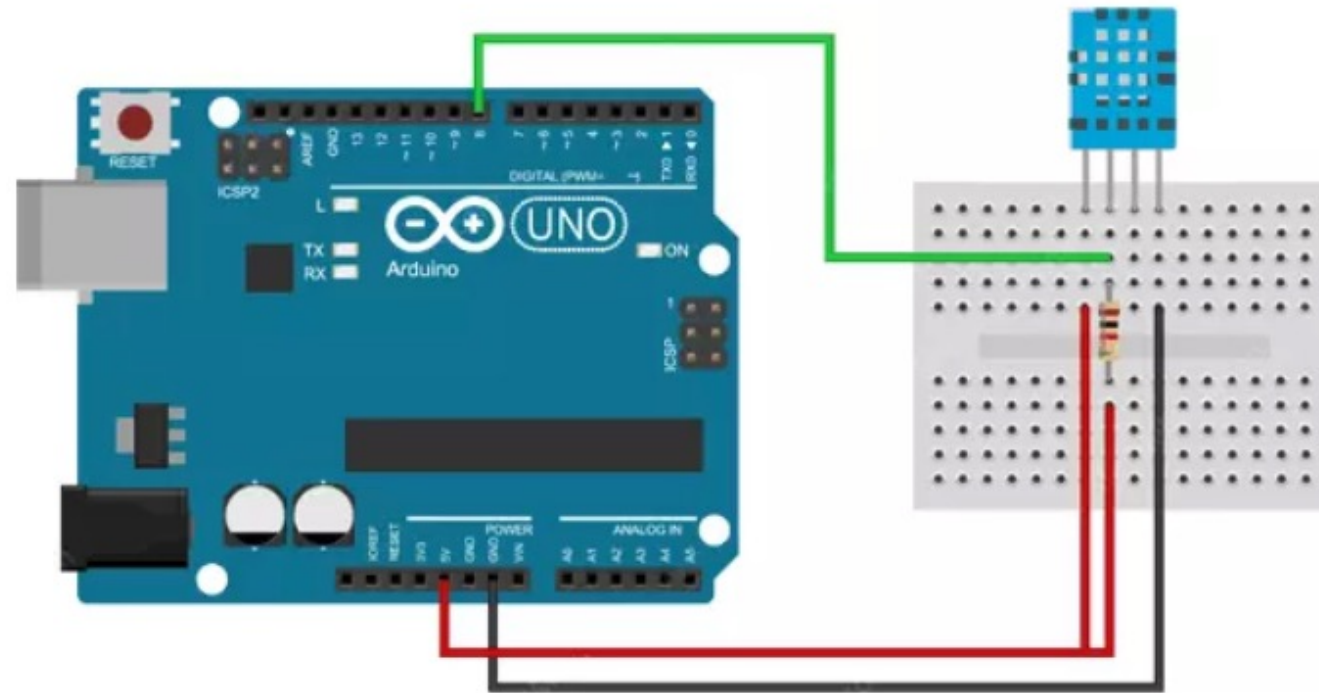
Data: این پین برای برقراری ارتباط سنسور و میکروکنترلر است.

NC: این پین متصل نمی شود.

GND: این پین به زمین آردوینو متصل می شود.



نحوه اتصال سنسور دما به آردینو



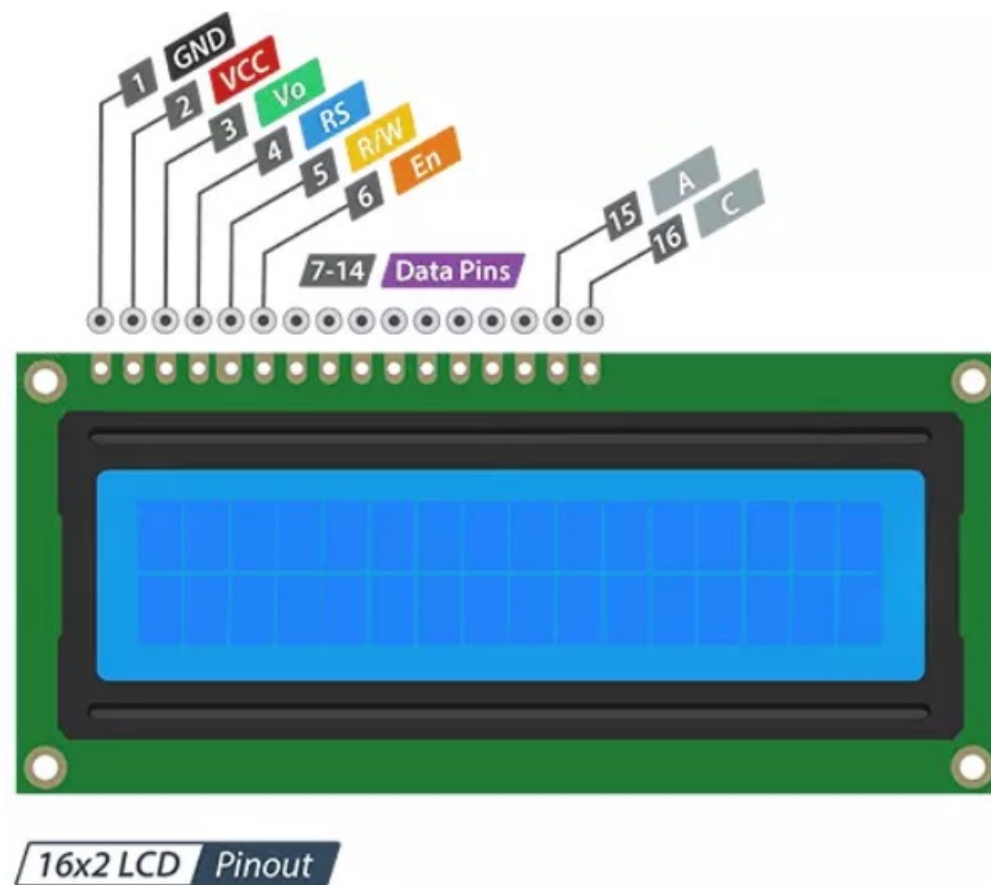
سیم‌کشی سنسور DHT11 به برد آردوینو UNO

ال سی دی کاراکتری

LCD های کاراکتری همان طور که از نام آن پیدا است، تنها برای نمایش پیام متنی و یا کاراکتر طراحی شده اند. این LCD دارای یک LED برای نور پس زمینه است که می تواند 32 کاراکتر اسکی را در دو ردیف 16 تایی نمایش دهد.



جانمایی پین های LCD



نحوه اتصال پین ها

- **پین GND :** به زمین آردوینو متصل می شود.

- **پین VCC :** تغذیه الکتریکی LCD است و به پین 5 ولت آردوینو متصل می شود.

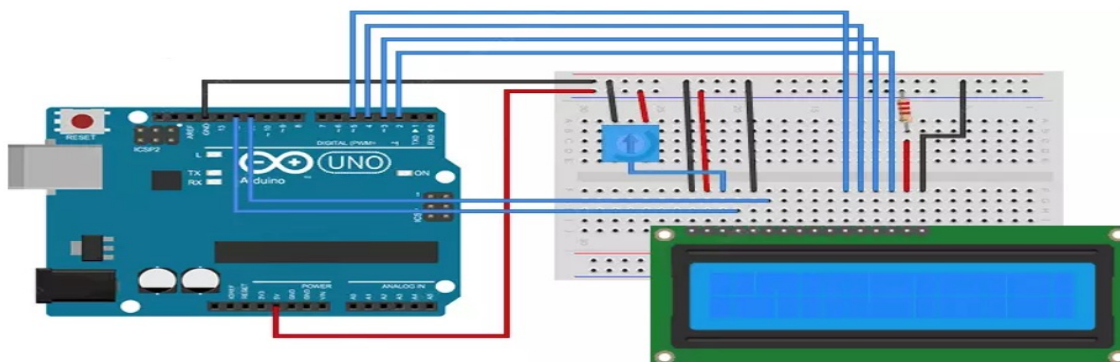
- **پین Vo :** جهت کنترل کنتراست Contrast و نور LCD مورد استفاده قرار می گیرد. با استفاده از یک تقسیم کننده ولتاژ و یک پتانسیومتر می توانید کنتراست را تنظیم کنید.

- **پین RS :** به آردوینو اجازه می دهد که به LCD اطلاع دهد که آیا در حال ارسال دیتا یا ارسال فرمان است. این پین اصولاً برای ایجاد تمایز بین فرمان و دیتا مورد استفاده قرار می گیرد. به عنوان مثال هنگامی که مقدار پین RS به صورت LOW باشد، در واقع شما در حال ارسال فرمانی از آردوینو به LCD هستید (فرمان هایی مانند دستور تنظیم مکان نما در یک مکان خاص، دستور پاک کردن صفحه نمایش، دستور رفتن به سمت راست صفحه نمایش و). اما زمانی که پین RS به صورت High باشد، شما در حال ارسال دیتا یا کاراکتر از آردوینو بر روی LCD هستید.

- **پین R/W :** بر روی LCD به شما نشان می دهد که آیا در حال نوشتن دیتا بر روی LCD هستید یا در حال خواندن دیتا از آن هستید. با توجه به اینکه در این پروژه از این LCD تنها به عنوان خروجی استفاده می کنیم، باید مقدار این پین را LOW تنظیم کنید. زمانی که این پین LOW باشد، LCD در مد نوشتن قرار می گیرد و شما می توانید به LCD دیتا ارسال کنید.

نحوه اتصال پین ها

- **پین E :** جهت فعال سازی LCD استفاده می شود. یعنی زمانی که این پین LOW باشد، در حقیقت LCD اهمیتی به مقادیر پین های RS، R/W و خطوط باس دیتا نمی دهد. اما با High شدن این پین، LCD اطلاعات دریافتی را پردازش خواهد کرد.
- **پین های D0-D7 (باس دیتا) :** برای انتقال دیتای 8 بیتی مورد استفاده قرار می گیرند که شما برای LCD ارسال می کنید. به عنوان مثال اگر بخواهید کاراکتر A را بر روی LCD مشاهده کنید، باید این هشت پین را با کد 0001 0100 (براساس جدول اسکی) تنظیم کنید. به این شکل LCD کاراکتر A را نمایش خواهد داد.
- **پین های A-K (آند و کاتد) :** برای کنترل نور پس زمینه LCD مورد استفاده قرار می گیرند.



دستورات مرتبط با LCD

- این کد، با فراخوانی کتابخانه LiquidCrystal آغاز می‌شود. آردوینو دارای کتابخانه‌ای با نام LiquidCrystal است که راه اندازی و برنامه ریزی ماژول LCD را به مراتب ساده‌تر می‌کند.
- `#include <LiquidCrystal.h>`
- شی LiquidCrystal را ایجاد می‌کند. این شی از 6 پارامتر استفاده می‌کند و مشخص می‌کند که کدام یک از پین‌های آردوینو به هریک از پین‌های RS، E و چهار پین دیتا D4 تا D7 متصل شده‌اند.
- `LiquidCrystal lcd(12, 11, 5, 4, 3, 2);`
- `lcd.begin(16, 2);`
- `lcd.clear();`
- `lcd.setCursor(0, 1);`
- `lcd.print(" LCD Tutorial");`

ادامه توابع ...

- **تابع home :** اگر می خواهید به سادگی و بدون پاک کردن صفحه، مکان نما را در گوشه بالا سمت چپ قرار دهید، از تابع `home()` استفاده کنید.
- **تابع blink و lcd.cursor :** اپلیکیشن های زیادی نظیر توربو C++ و یا ++notepad وجود دارند که در آن ها با فشار کلید `insert` مکان نما تغییر خواهد کرد. چیزی مشابه همین عملکرد را می توانید برای تغییر مکان نما LCD با استفاده از تابع `blink()` یا `lcd.cursor()` داشته باشید.
- **تابع blink ()** بلوک های چشمک زن از پیکسل های 8×5 را نمایش خواهد داد. در صورتی که تابع `lcd.cursor()` یک خط در زیر مکانی که قرار است گازاکتر بعدی نوشته شود، نمایش خواهد داد.
- **تابع noBlink () و lcd.noCursor ()** برای خاموش کردن مکان نمای چشمک زن می توانید از تابع `noBlink()` و برای مخفی کردن مکان نما نیز می توانید از تابع `lcd.noCursor()` استفاده کنید.
- **تابع lcd.scrollDisplayRight () و lcd.scrollDisplayLeft ()** با استفاده از این دو تابع می توانید محتوای نمایش داده شده بر روی LCD را به اندازه یک `space` به سمت راست و یا سمت چپ شیفت دهید. اگر می خواهید محتوای نمایش داده شده بر روی صفحه LCD را به صورت پیوسته به یک سمت حرکت دهید، کافی است این توابع را در یک حلقه `for` استفاده کنید.

```
#include <DHT11.h>

#define DHT11_PIN 8
DHT11 dht11(DHT11_PIN);

#include <LiquidCrystal.h>
#define RS 2
#define E 3
#define D4 4
#define D5 5
#define D6 6
#define D7 7
LiquidCrystal lcd(RS,E,D4,D5,D6,D7);

void setup(){
  lcd.begin(16, 2);
}
void loop()
{
  lcd.setCursor(0,0);
  lcd.print("Temp = ");
  lcd.print((int)dht11.readTemperature());
  lcd.print(" C");
  lcd.setCursor(0,1);
  lcd.print("Humidity = ");
  lcd.print((int)dht11.readHumidity());
  lcd.print(" %");
  delay(2000);
}
```