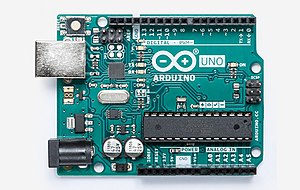
**آردوینو** (Arduino) یک پلتفرم سخت‌افزاری و نرم‌افزاری متن باز است. پلتفرم آردوینو شامل یک [میکروکنترلر تک‌بردی](https://fa.wikipedia.org/wiki/%D8%B1%DB%8C%D8%B2%DA%A9%D9%86%D8%AA%D8%B1%D9%84%E2%80%8C%DA%AF%D8%B1" \o "ریزکنترل‌گر) متن باز است که قسمت سخت‌افزار آردوینو را تشکیل می‌دهد. علاوه بر این، پلتفرم آردوینو یک IDE که به منظور برنامه‌نویسی برای بردهای آردوینو طراحی شده‌است و یک بوت لودر نرم‌افزاری که بر روی میکروکنترلر بارگذاری می‌شود را در بر می‌گیرد. پلتفرم آردوینو به منظور تولید سریع و ساده پروژه‌های سخت‌افزاری تعاملی و ساخت وسایلی که با محیط تعامل داشته باشند و البته اهداف آموزشی طراحی شده‌اند.



برد آردوینو می‌تواند مقادیر ورودی را از تعداد زیادی سنسور و کلید و… بخواند و بر اساس برنامه ای که درون آن بارگذاری شده‌است تصمیم بگیرد و خروجی خاصی که می‌تواند کنترل تعدادی لامپ، موتور و … را برای شما انجام دهد. آردوینو در سال ۲۰۰۵ به منظور ایجاد راهی ارزان و ساده برای برنامه‌نویسی اشیایی تعاملی ایجاد شد. آردوینو به همراه یک IDE ساده ارائه می‌شود که در رایانه‌های عادی قابل اجرا است که اجازهٔ برنامه‌نویسی به کمک [سی](https://fa.wikipedia.org/wiki/%D8%B3%DB%8C_(%D8%B2%D8%A8%D8%A7%D9%86_%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87%E2%80%8C%D9%86%D9%88%DB%8C%D8%B3%DB%8C)" \o "سی (زبان برنامه‌نویسی)) یا [سی++](https://fa.wikipedia.org/wiki/%D8%B3%DB%8C%2B%2B" \o "سی++) را برای آردوینو می‌دهد.

آردوینو، یک نرم‌افزار متن باز اختصاصی برای برنامه‌نویسی بردهای خود تهیه کرده‌است که به نام [نرم‌افزار آردوینو IDE](https://fa.wikipedia.org/wiki/%D9%86%D8%B1%D9%85%E2%80%8C%D8%A7%D9%81%D8%B2%D8%A7%D8%B1_%D8%A2%D8%B1%D8%AF%D9%88%DB%8C%D9%86%D9%88_(%D9%85%D8%AD%DB%8C%D8%B7_%D8%AA%D9%88%D8%B3%D8%B9%D9%87%E2%80%8C%DB%8C_%DB%8C%DA%A9%D9%BE%D8%A7%D8%B1%DA%86%D9%87)" \o "نرم‌افزار آردوینو (محیط توسعه‌ی یکپارچه)) شناخته می‌شود. نرم‌افزار آردوینو بدون نیاز به پروگرامر می‌تواند مستقیماً میکرو کنترلر شما را برنامه‌ریزی نماید. امکاناتی مانند ترمینال سریال نیز در داخل نرم‌افزار آردوینو قرارداده شده تا بتوانید به وسیله ارتباط سریال و USB برنامه‌هایی را که برای میکرو کنترلر می‌نویسید عیب‌یابی نمایید. همان‌طور که پیش تر اشاره شد، شیوه نگارش دستورها در این نرم‌افزار، تا حدود زیادی مشابه برنامه‌نویسی به زبان سی و سی++ است. (منبع ویکی پدیا)

* **How does Arduino Work?**

1. **Hardware (Arduino Board):**

The heart of Arduino is a microcontroller (e.g., **ATmega328** on the Arduino Uno board).

* + The board contains GPIO (General Purpose Input/Output) pins that can be used to interface with sensors (input devices) and control things like LEDs, motors, displays (output devices).
  + It can communicate via digital, analog, and serial signals, making it very versatile for embedded systems.

1. **Software (Arduino IDE):**
   * Arduino has its own development environment called **Arduino IDE** (Integrated Development Environment), where you write programs (called "sketches") using a simplified version of **C++**.
   * You write code on your computer, upload it to the Arduino board via a USB cable, and the board executes the code to control connected components (like reading from a sensor or turning on an LED).

* **Which Simulator is Best for This Project?**

Considering project involves: Reading sensor data (temperature and humidity), Displaying results on an LCD, Handling switches, LEDs, and real-time data changes,

**Proteus** would be the most efficient and powerful simulator for this. It allows us to work with real components virtually and gives us detailed control over sensors, displays, and alerts (LEDs). We can precisely test how our system will behave in real-time.

پروتئوس یکی از کارآمدترین و قدرتمندترین شبیه سازها برای این کار است. این به ما اجازه می دهد تا با اجزای واقعی به صورت مجازی کار کنیم و همچنین کنترل دقیقی بر حسگرها، نمایشگرها و هشدارها (LED) به ما می دهد. ما می توانیم دقیقاً آزمایش کنیم که سیستم ما در زمان واقعی چگونه کار می‌کند.

**Wokwi** is an online Arduino simulator that allows us to design, simulate, and test Arduino projects directly in our browser. It provides a virtual environment where we can connect various components like sensors, LEDs, LCDs, and switches without needing the physical hardware. It is excellent choice too.

* + **Real Hardware vs. Simulator:**

**Simulator Benefits:** Great for testing before working with real hardware, especially if we don't have access to physical components or if we are still learning.

**Real Hardware Benefits:** Better for real-world testing, ensuring our system works reliably. Simulators might not cover all edge cases or real-time delays, but real hardware will show the actual behavior.

In our case, we could start with **Proteus** to simulate and test our code and circuits, and later, if possible, move to **real hardware** to finalize the project for actual testing and deployment.

**-------------------------------------------------------------------------------------------------------------**

* + **Components:**

**1. Arduino Board:** The **Arduino Uno** is an excellent choice for our project:

Why Arduino Uno: It's the most commonly used board, easy to program, with enough I/O pins to connect all components.

**2. LCD Display: 16x2 LCD (with I2C interface)**:

**How to Use LCD in Arduino:** The LCD will be used to display sensor readings (like temperature and humidity) or predicted values.

We should connect the LCD to the Arduino using **GPIO pins** (or I2C for fewer wires).

In the **Arduino code**, we will use an **LCD library** to send data to the display (for example, showing sensor values or predictions).

**16x2 LCD**: This is the most commonly used LCD with two rows and 16 characters per row. It’s perfect for displaying short messages or data (e.g., “Temp: 25°C” and “Humidity: 60%”).

**I2C LCD**: A type of LCD that uses the I2C protocol to communicate with Arduino, which reduces the number of pins required for wiring.

**Why I2C**: Using an **I2C LCD** requires fewer pins (only 2 for communication—**SDA** and **SCL**) compared to a regular LCD, which needs many pins.

**Tip**: Get a 16x2 LCD with an I2C module pre-attached to make wiring easier.

**3. Sensors:**

**DHT22** is more accurate and has a wider range (temperature: -40 to +125°C, humidity: 0-100%).

**DHT11** is cheaper but has a smaller range and is less accurate.

**Recommendation**: Use **DHT22** if accuracy is important for your predictions; otherwise, **DHT11** is also fine for basic projects.

**4. LEDs (Light Emitting Diodes):**

Using **LEDs** to indicate different statuses (e.g., a warning when temperature exceeds a threshold, like 30°C).

**Quantity**: 2 or 3 LEDs are enough (e.g., one for temperature alert, one for humidity alert, and one for general status).

**Resistors for LEDs**: You will need 220Ω resistors for each LED to limit the current and prevent them from burning out.

**5. Switches:**

We will need **push buttons (switches)** to control our system, such as toggling between different display modes (e.g., switching between displaying current values and predicted values):

**Quantity**: 1 or 2 switches. For example, one switch to toggle between current and predicted temperature, and another to reset or turn the system on/off.

**6. Resistors:**

For switches We’ll need **pull-down resistors** (typically **10kΩ**) for each switch(1 resistor per switch) to ensure proper signal readings when the switch is not pressed.

**7. Power Supply:**

Powering Arduino: We could power the Arduino via USB or use a **9V battery** with a **DC adapter**.

**8. Wires and Breadboard:**

**Breadboard**: Using for easy prototyping and connecting components.

**Jumper Wires**: we’ll need enough jumper wires (male-to-male) to connect the Arduino to sensors, LCD, LEDs, and switches.

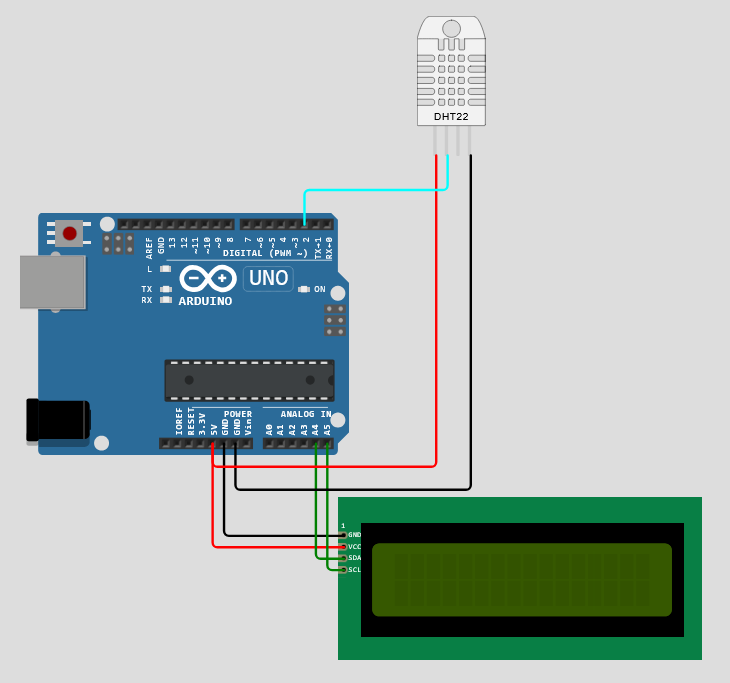
* **شروع عملی پروژه**

همان طور که پیش تر نیز گفته شد، برای انجام پروژه از سایت Wokwi استفاده کردیم. این سایت یک شبیه ساز عالی برای انجام پروژه است. در کنار پیاده سازی کد در یک محیط عالی، قابلیت های بسیار خوبی برای اجزای سخت افزاری مدار هم دارد. با نصب کتابخانه های مختلف و جستجوی اجزای متفاوت می‌توان مدار های خوبی را درست کرد.

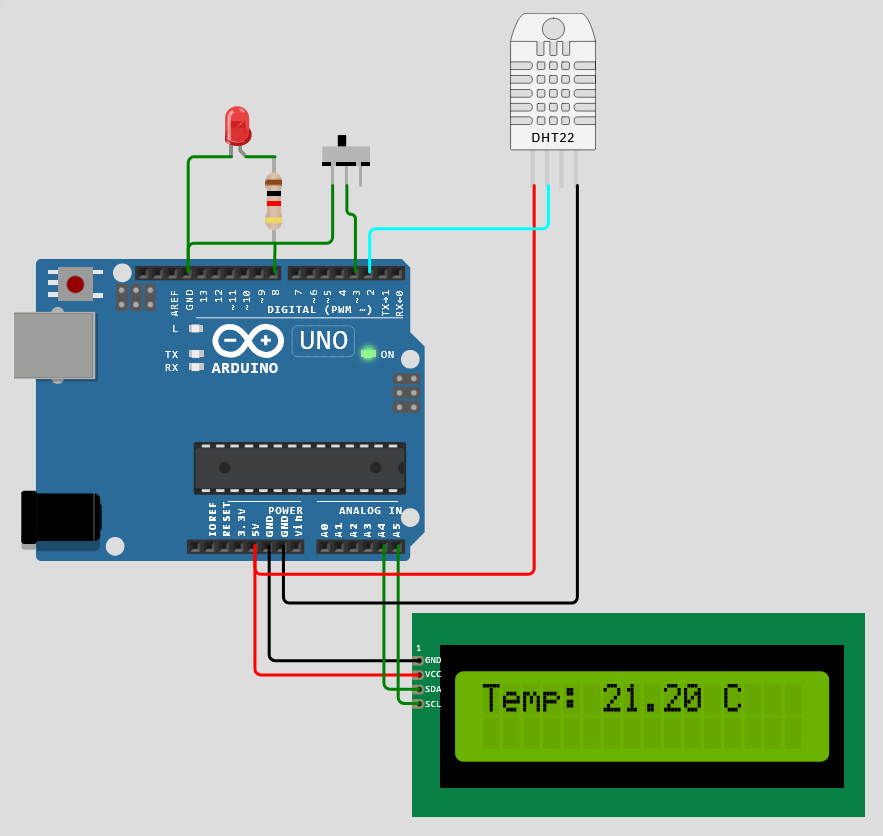
سعی کردیم کار هایی که برای انجام عملی پروژه پس از یک سری مقدمات جهت یادگیری، انجام داده ایم را به صورت گام های مشخص و واضح در این گزارش شرح دهیم:

1. ابتدا ساده ترین شکل مدار را می‌بندیم. بدین شکل که سنسور دما و رطوبت (DHT22) و نمایشگر (LCD 16\*2 (ICD)) که در واقع نمایشگری با دو لاین و 16 کاراکتر در هر لاین می‌باشد را به برد آردوینو وصل میکنیم.

* سنسور دما و رطوبت : شامل چهار پین ورودی/خروجی می‌باشد. دو پین مربوط به ولتاژ های بالا و پایین (VCC و GND) و دو پین با نام های SDA (Digital data pin (input/output)) که برای اتصال سنسور به مدار استفاده میشود و NC (Not connected) که به جایی وصل نشده است.
* نمایشگر: این عنصر هم چهار پین ورودی خروجی دارد. در کمار دو پین VCC و GND، دو پین دیگر دارد: SDA (I2C data line) که مربوط به داده است و SCL (I2C clock line) که مربوط به زمان است. این دو پین به A4 و A5 در پین های برد متصل شده‌اند.



1. مرحله بعد پیاده سازی کد مربوطه است.
2. در مرحله بعد بررسی شروط خاص را پیاده سازی میکنیم.
3. در یک مرحله خاص، عملیات های مربوط به پیش بینی را پیاده سازی میکنیم.
4. در نهایت یک سری جزیئات را نهایی میکنیم.



<https://wokwi.com/projects/421269680768459777>

پروژه‌ای که ما انجام دادیم، در مورد اندازه گیری دما و رطوبت محیط است. این پروژه دوم اصلی درس ساختار و زبان کامپیوتر ما میباشد.