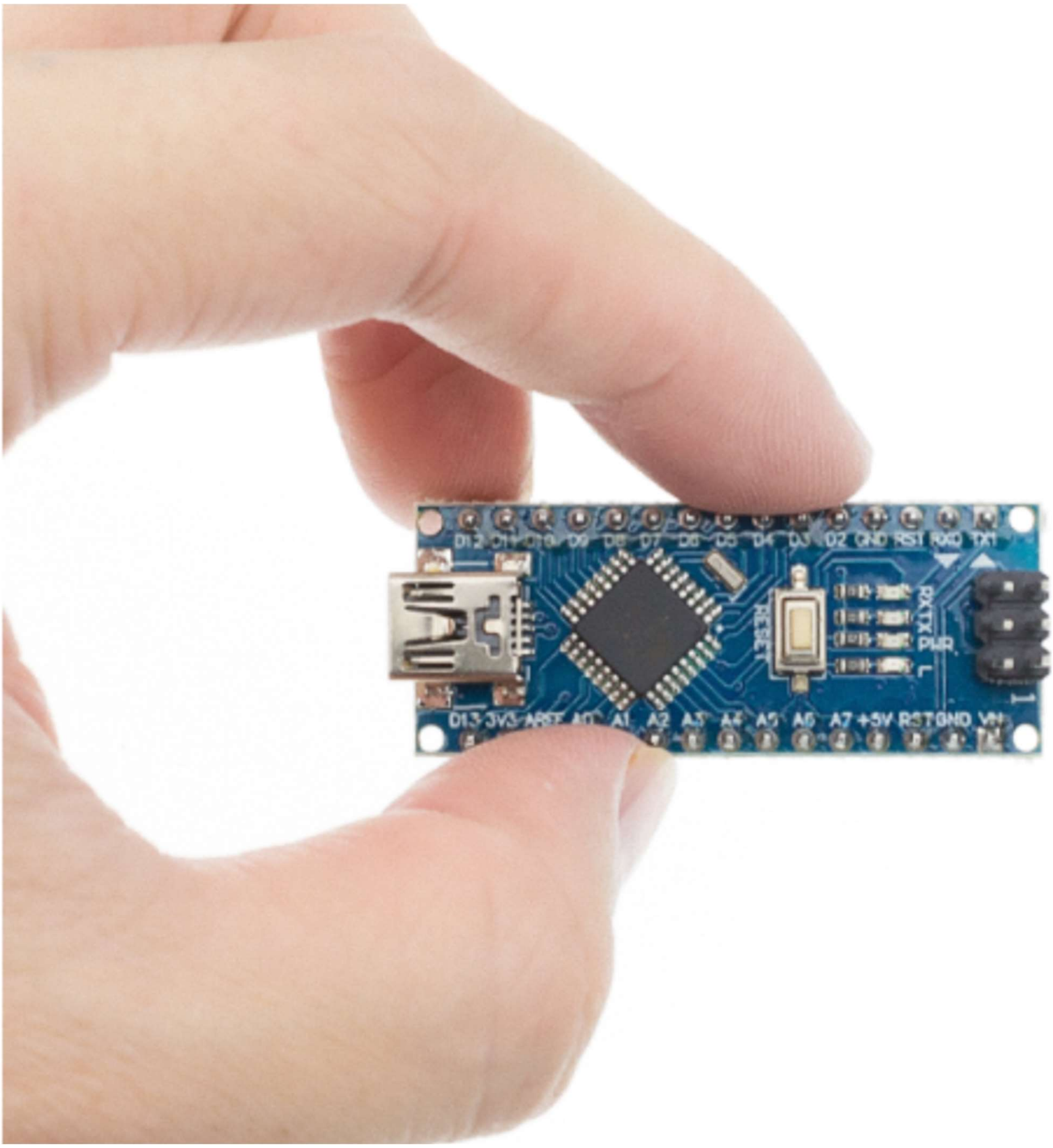
MAISON UP
for makers, engineers and beyond

Nano Board CH340/ATmega328P

GUIDE FOR ABSOLUTE BEGINEERS


D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RXD TX1
RESET
RX TX PWR L
D13 3V3 AREF A0 A1 A2 A3 A4 A5 A6 A7 +5V RST GND VIN

Contents

Chapter 1. Introduction to Arduino Nano

Welcome to the world of Arduino Nano, a versatile microcontroller board designed for a wide range of electronic and embedded projects. Whether you're a hobbyist, student, or professional, the Arduino Nano is here to empower your creativity. It's based on the reliable ATmega328P microcontroller, like the Arduino Uno, making it a robust tool for your projects.

Chapter 2. Key Features of Arduino Nano

Compact Size: Arduino Nano's small form factor is perfect for projects with tight spaces, making it ideal for size-constrained applications.

USB Connectivity: With a built-in USB interface, programming and power supply become hassle-free, eliminating the need for extra external components.

Digital and Analog Pins: The board provides a set of digital and analog pins that can interface with various sensors, actuators, and other components.

Arduino IDE Compatibility: Arduino Nano is fully compatible with the Arduino Integrated Development Environment (IDE), ensuring accessibility for beginners and experienced programmers.

Diverse Applications: Whether you're into robotics, home automation, data logging, or interactive art installations, Arduino Nano adapts to suit your needs.

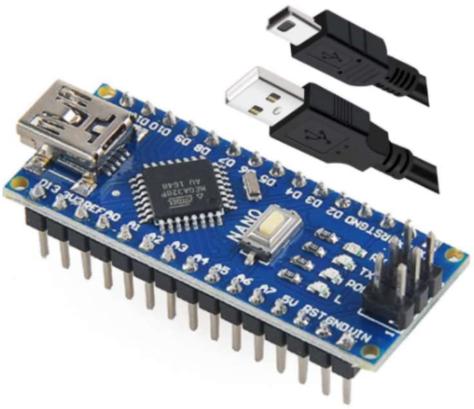
Expandable: Enhance your projects by expanding Arduino Nano with various shields and modules, unlocking even more capabilities.

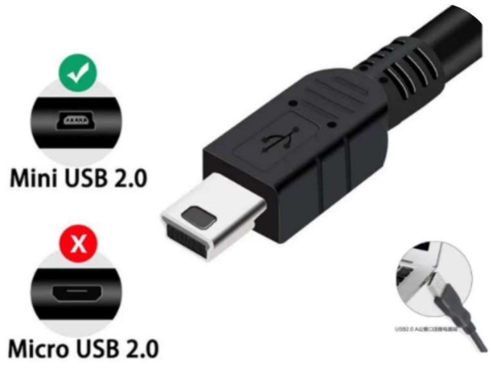
Chapter 3. Getting Started with Arduino Nano

Getting started is straightforward,

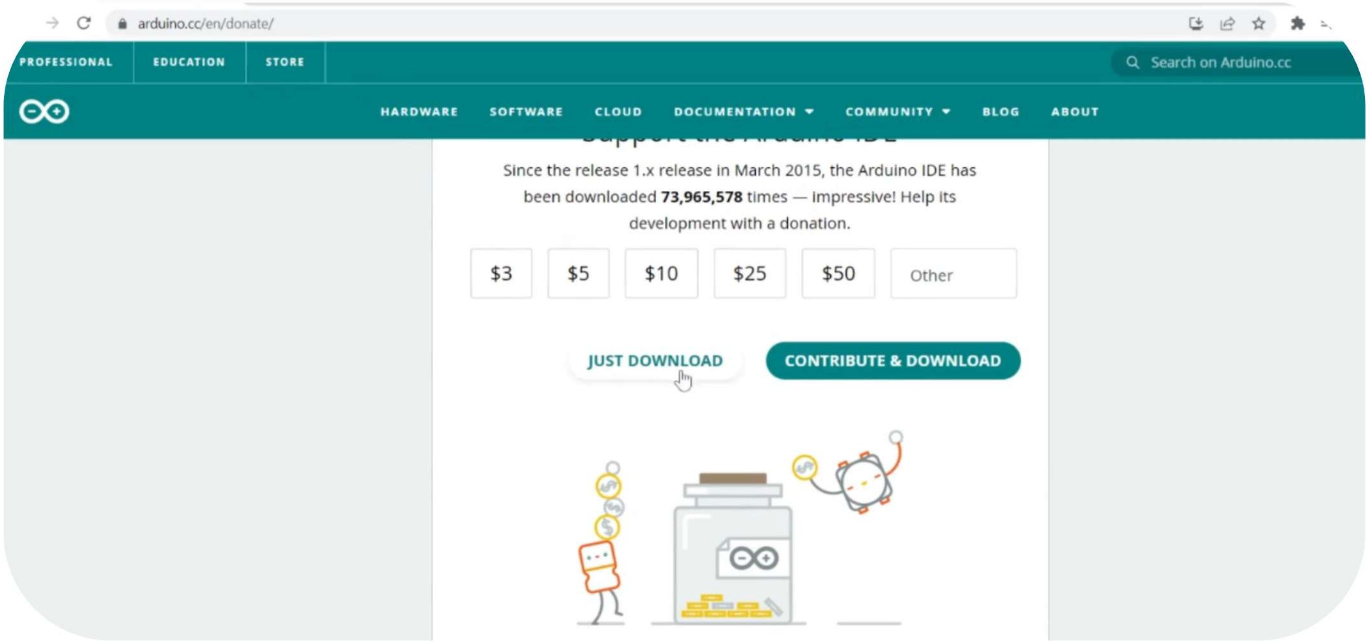
Unboxing and Inspection: Carefully unbox your Arduino Nano and inspect the board and accessories for any visible damage.

Please note that the Arduino Nano necessitates a USB 2.0 A to USB 2.0 Mini B cable for programming, which differs from the standard USB cables typically used for charging various devices, such as cameras, MP3 players, and gaming consoles. It's crucial to have a cable that supports both charging and data transmission functions to ensure seamless programming of your Nano Board.





Installing the Arduino IDE: Download the Arduino IDE from the official website and install it according to your operating system.



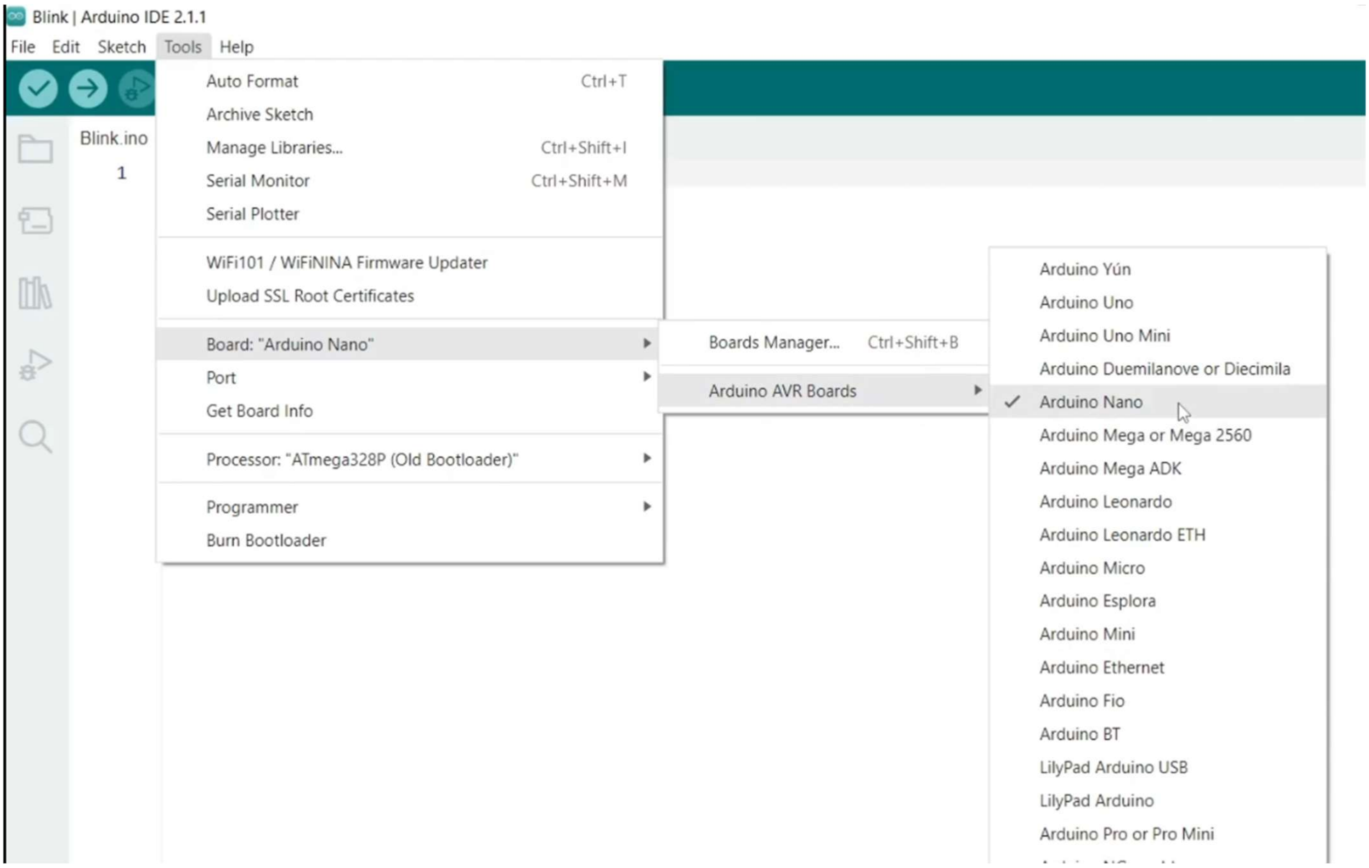
Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **73,965,578** times — impressive! Help its development with a donation.

$3 | $5 | $10 | $25 | $50 | Other

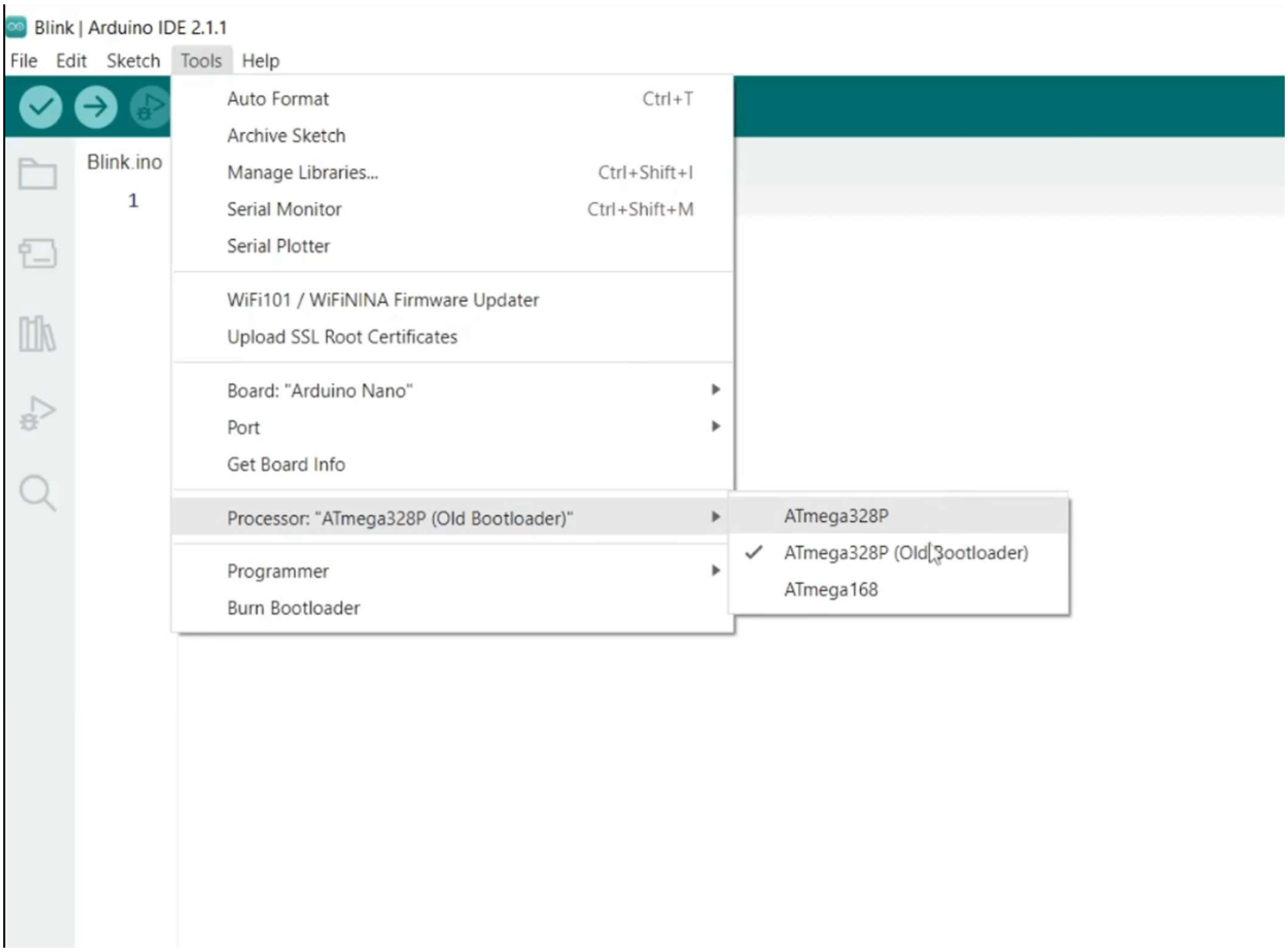
JUST DOWNLOAD | CONTRIBUTE & DOWNLOAD

Selecting the Arduino Nano Board:

Open the Arduino IDE, navigate to "Tools" > "Board," and choose "Arduino Nano" from the list.



Selecting the Processor: Open the IDE, In the 'Tools' menu, select the appropriate processor for your board. For most Nano boards, choose 'ATmega328P' or 'ATmega328P(Old Bootloader), depending on your specific board model. If you encounter difficulties in uploading code, you can try changing this setting.

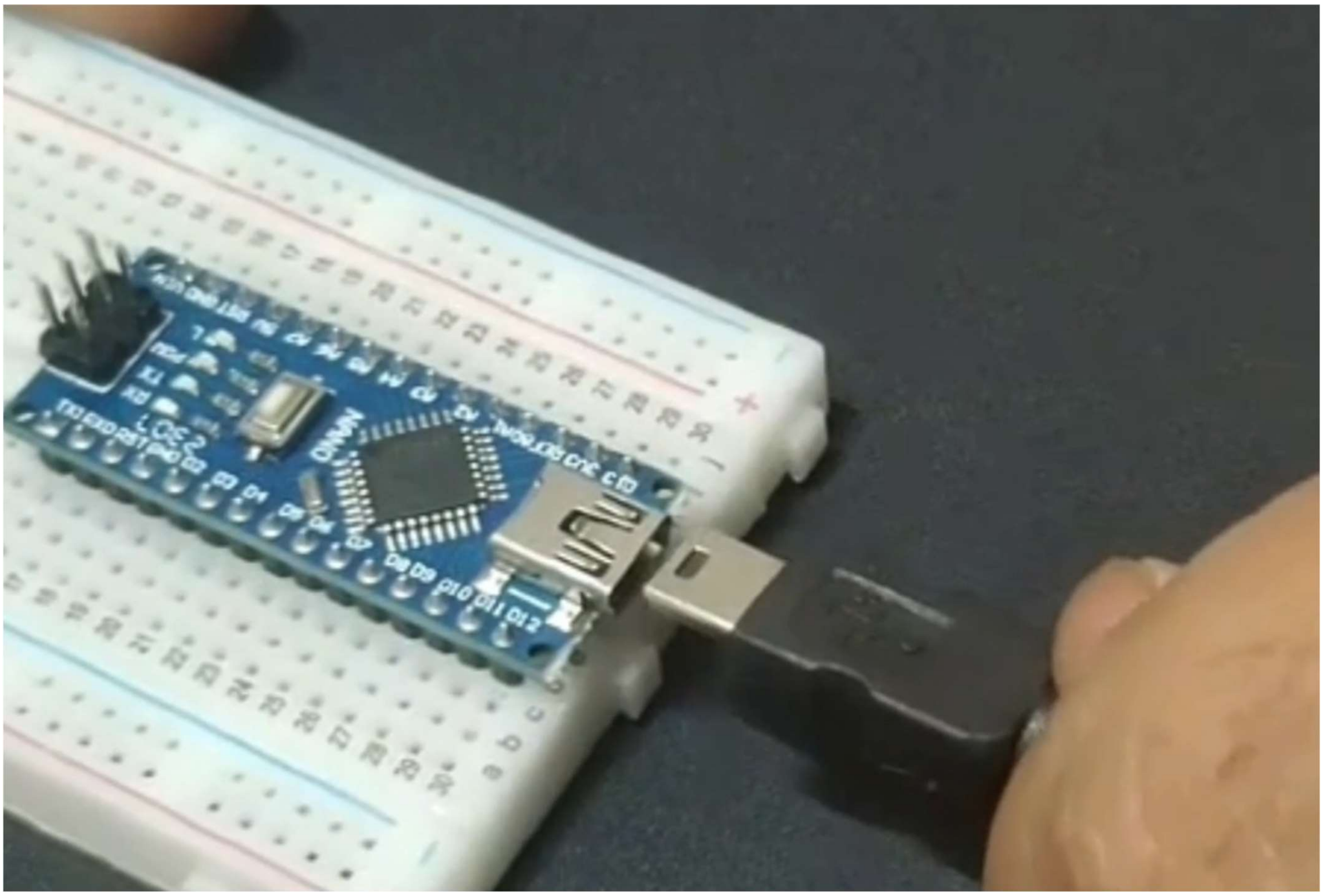


Board: "Arduino Nano"

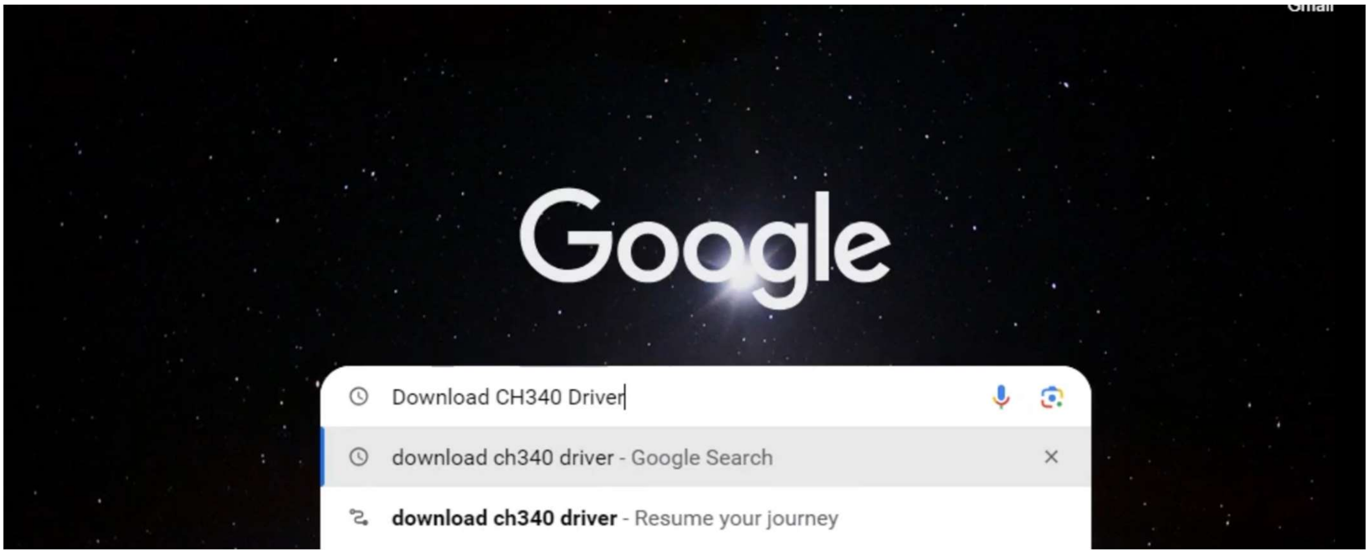
Processor: "ATmega328P (Old Bootloader)"

Blink | Arduino IDE 2.1.1
File
Edit
Sketch
Tools
Help
Blink.ino
1
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
Port
Get Board Info
Programmer
Burn Bootloader
ATmega328P
✓ ATmega328P (Old Bootloader)
ATmega168

Connecting the Arduino Nano: Plug your Arduino Nano into your computer using a USB cable. This powers the board and allows you to upload your programs.



Ensure you have the CH340 driver installed to enable successful programming of your Nano board. You can use your favourite web browser to search for 'Download CH340 Driver' and install it according to your PC's operating system. Please note that the Nano board will not work without this driver, and installing it is a mandatory step for successful programming.



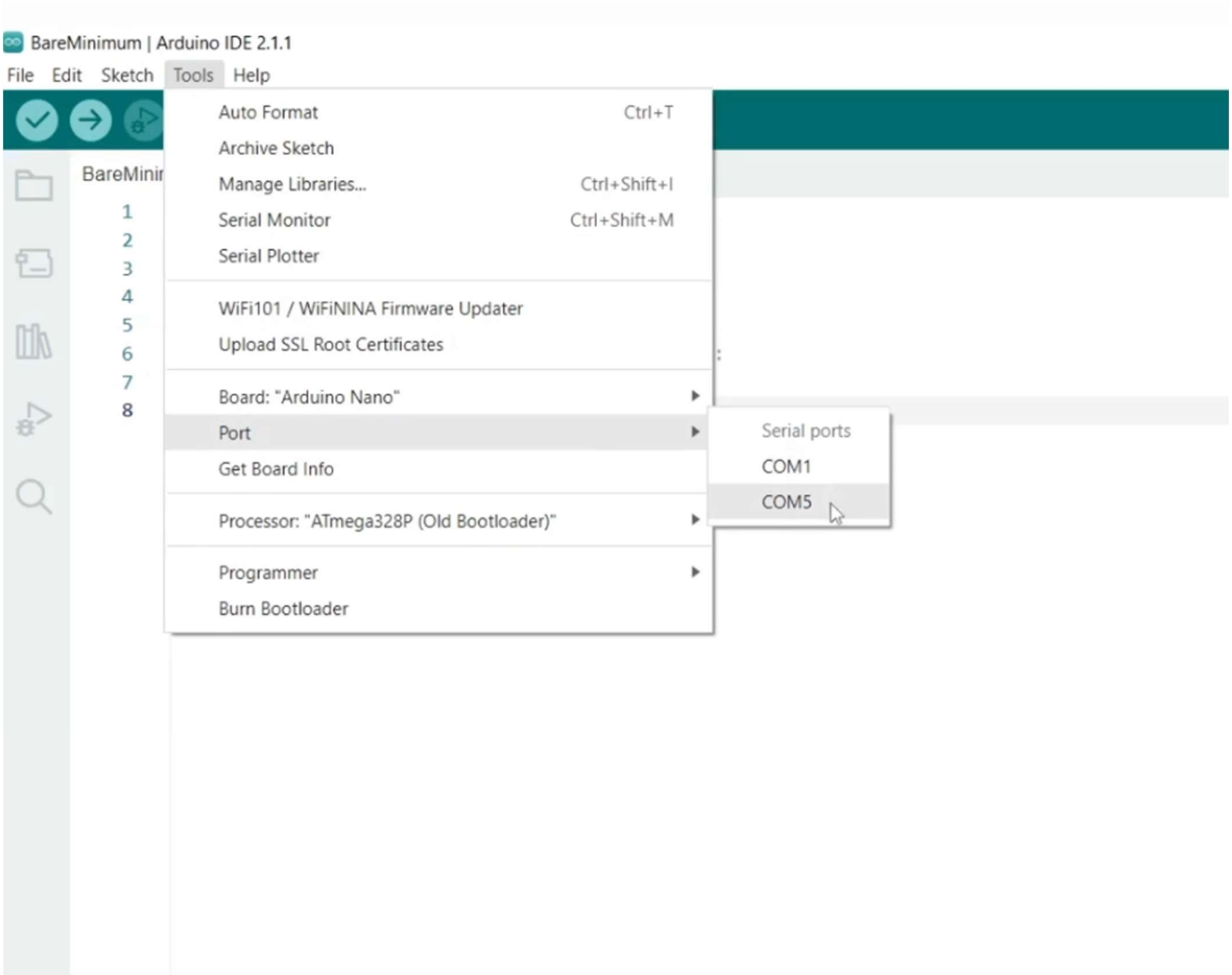
Download CH340 Driver

download ch340 driver - Google Search

download ch340 driver - Resume your journey

Choosing the COM Port:

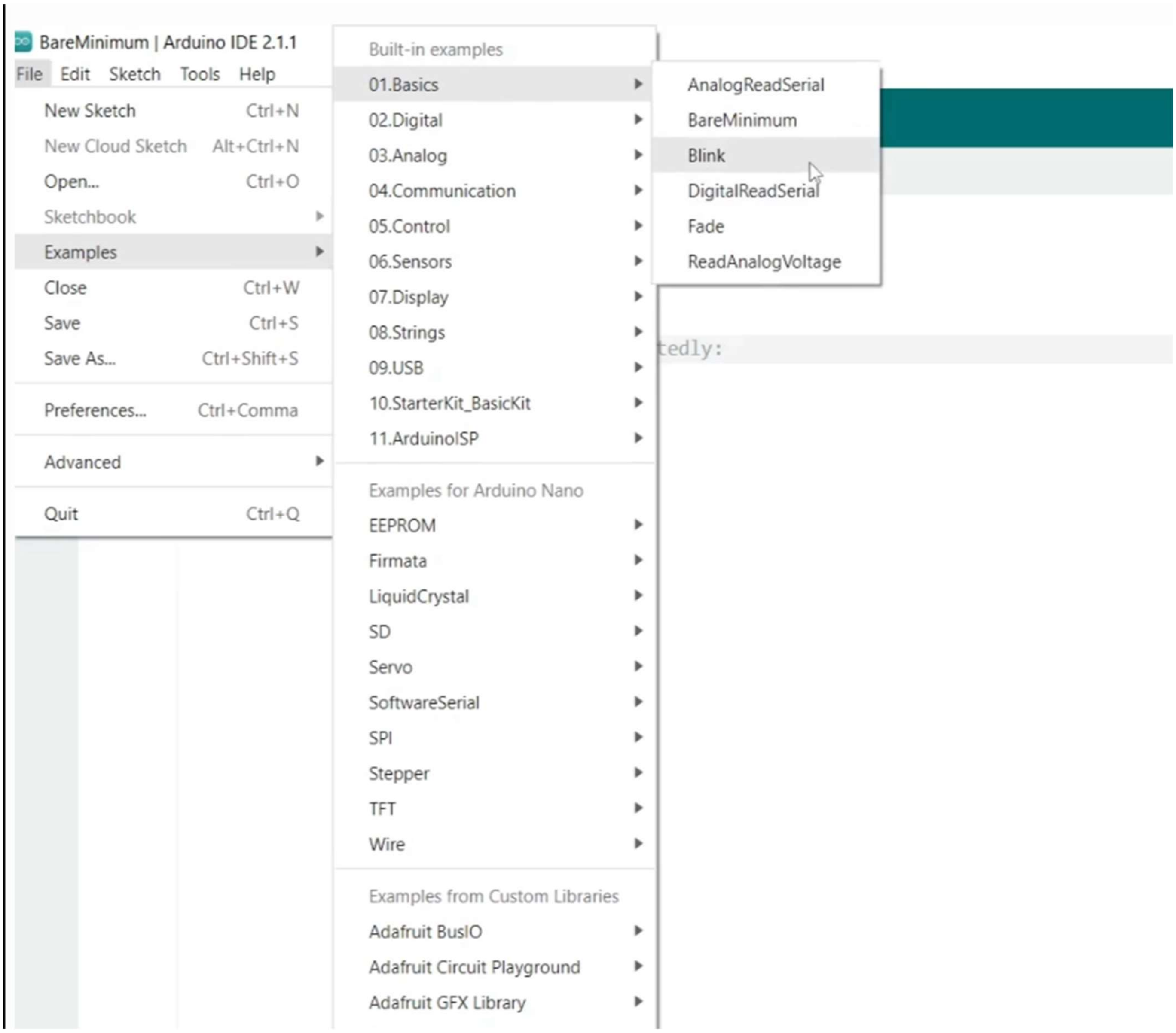
When selecting the COM port for your Arduino Nano, navigate to the 'Tools' menu and choose the COM port that corresponds to your board. If you're having trouble locating the COM port, you can try disconnecting and reconnecting the board to see it disappear and reappear in the list. Alternatively, you can access the Device Manager settings on your computer and scroll through the 'USB' section to find the 'CH340 Port'.


BareMinimum | Arduino IDE 2.1.1
File
Edit
Sketch
Tools
Help
BareMinir
1
2
3
4
5
6
7
8
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
Board: "Arduino Nano"
Port
Get Board Info
Processor: "ATmega328P (Old Bootloader)"
Programmer
Burn Bootloader
Serial ports
COM1
COM5

Uploading Your First Sketch: To upload your first sketch, let's start with a simple Blink code from the examples. Here's how to do it:

✓ In your Arduino IDE, navigate to 'File,' then 'Examples,' 'Basics,' and select 'Blink.'

✓ This will open the Blink sketch in a new window.

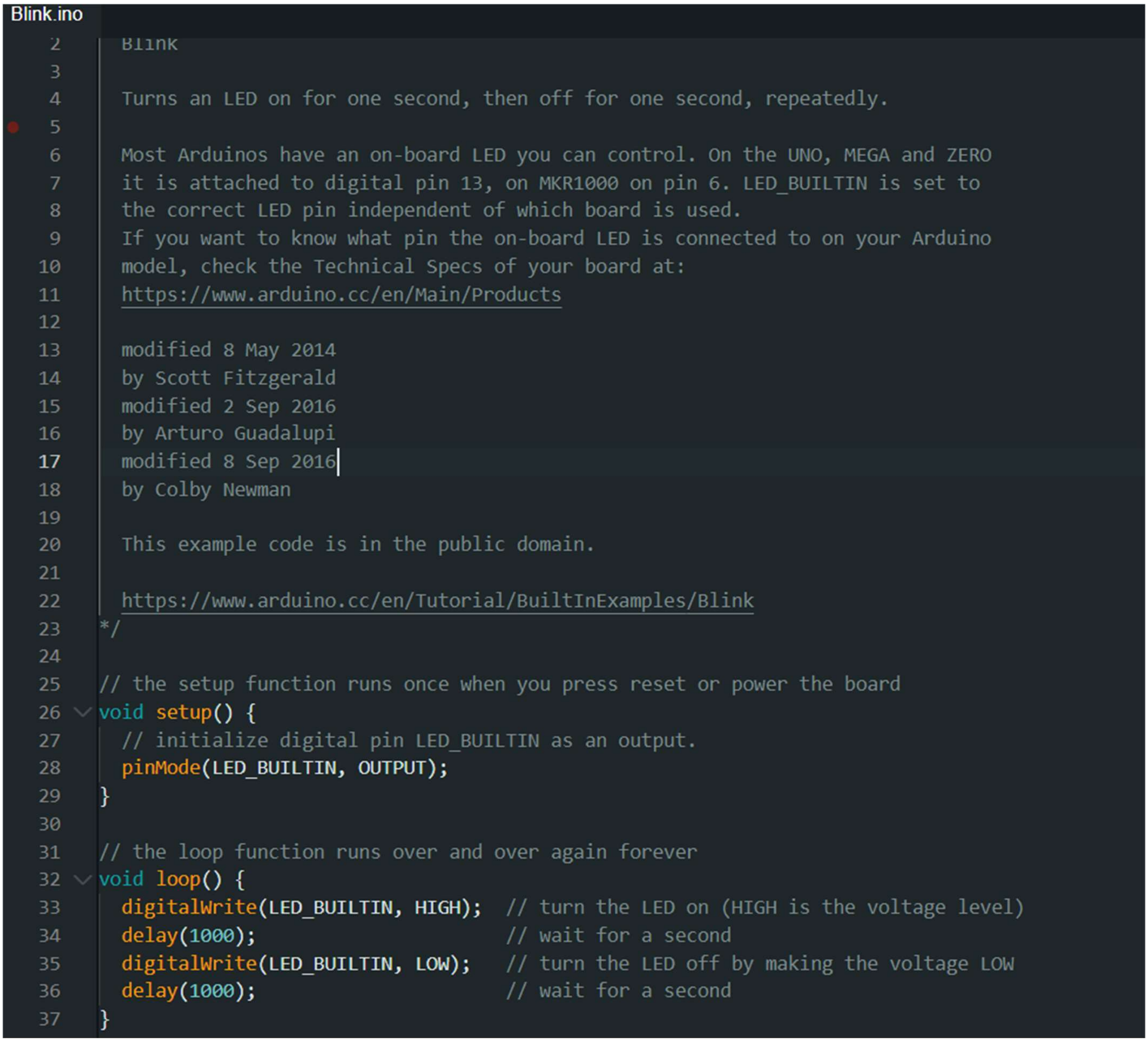


Understanding the Sketch Structure: Arduino sketches have two main functions: setup() and loop(). Use setup() for initialization and loop() for your main program logic.

The code begins with comments that explain what the program does and provide some additional information about the onboard LED on various Arduino models. It also mentions that the code is in the public domain.

The setup function is explained, stating that it runs once when you press the reset button or power the Arduino board. It initializes the built-in LED (referenced as LED_BUILTIN) as an output.

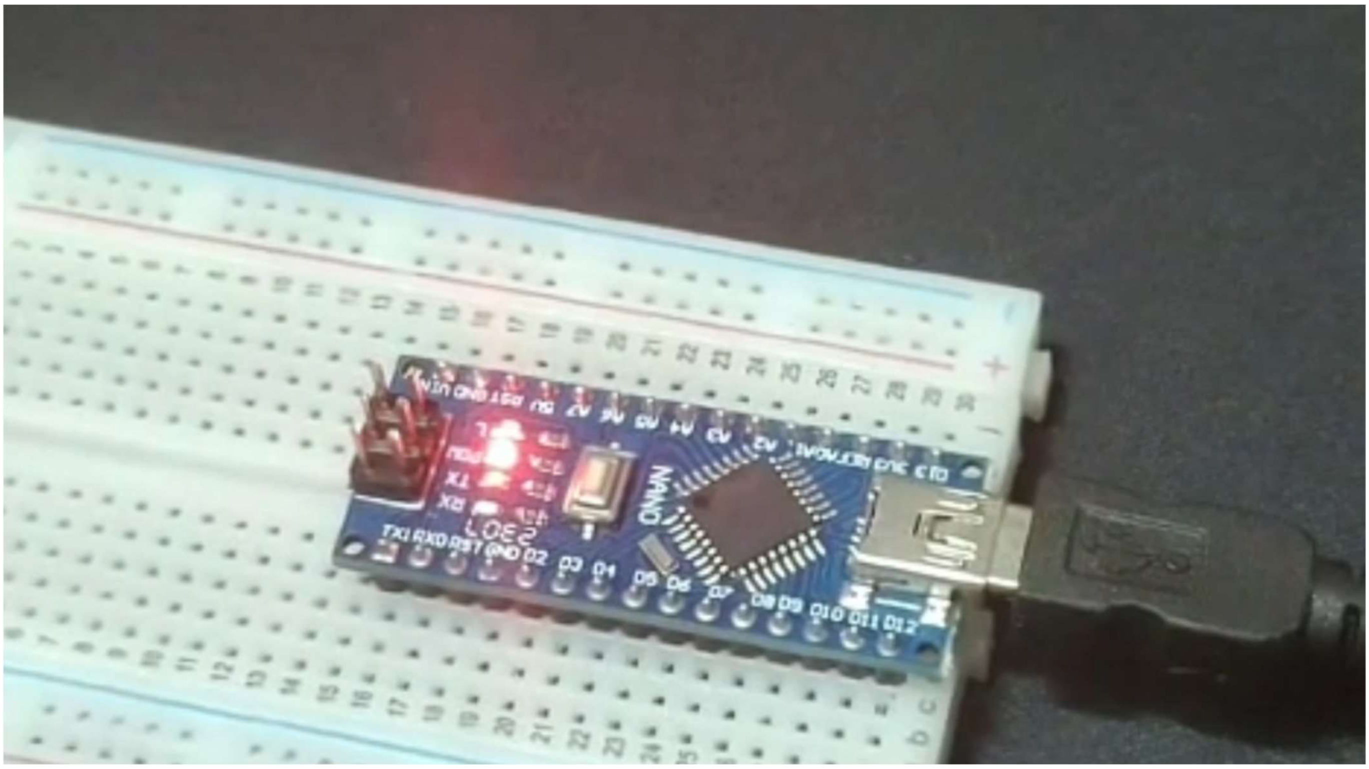
The loop function is explained as running repeatedly forever. It turns the LED on, waits for a second, turns it off, and waits for another second. This creates the familiar blinking effect.



```
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```

Uploading the Sketch: Click the "Upload" button in the top left of IDE to compile and upload your code. Once complete, your program is running on the Nano.

Now, you're ready to start experimenting and exploring Arduino Nano's potential.



Chapter 4. Power Sources and Voltage Requirements

Understanding power sources is crucial for Arduino Nano projects:

USB Power:

You can establish connection by simply plugging in the USB cable to your computer or any standard 5V charger. This powers the Arduino Nano and allows for both programming and operation.

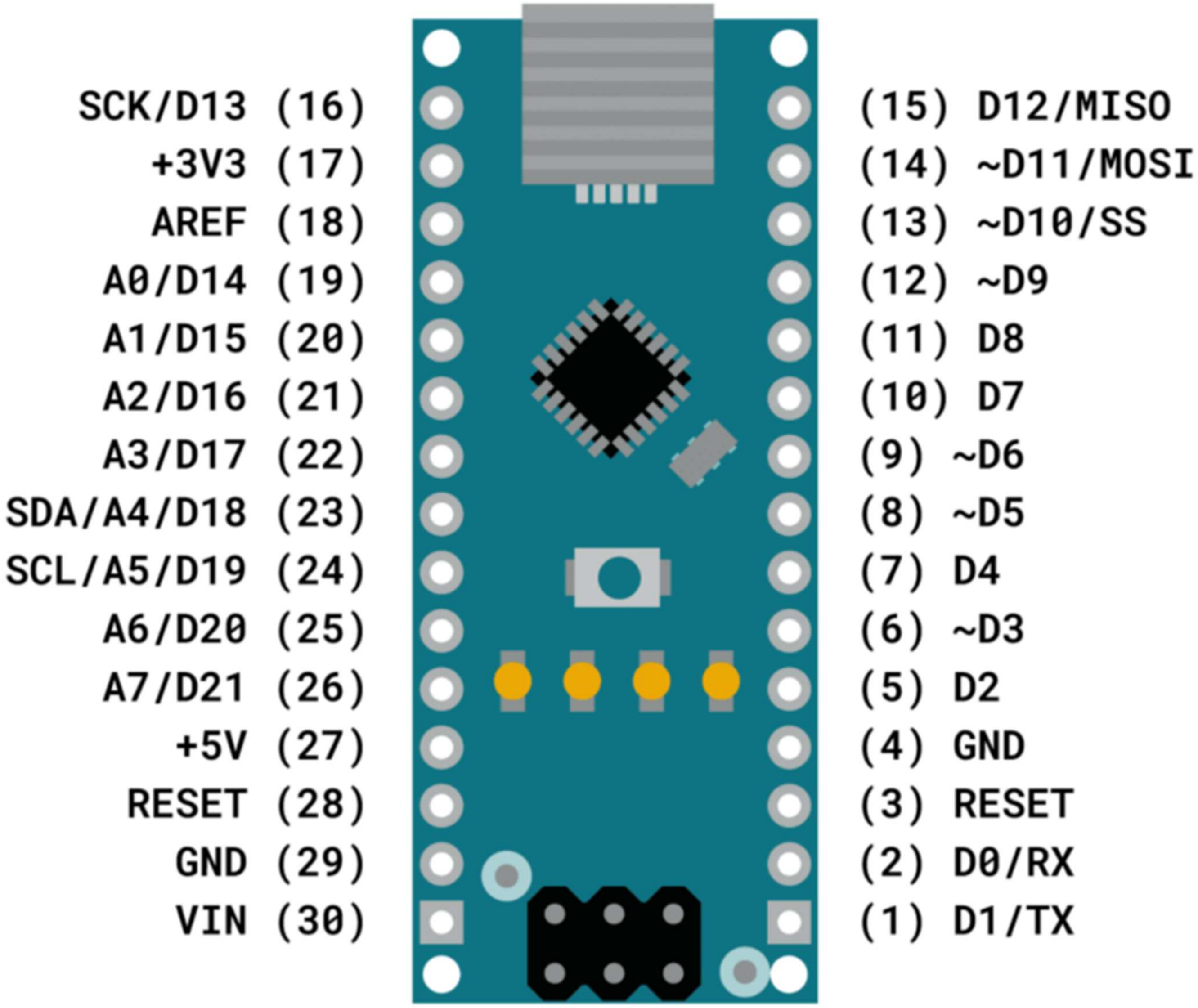
External Power Supply:

You can use batteries like or DC Power Supply on below 2 Pins to power your Nano Board as shown in the pin diagram below.

5V Pin: If you choose to power your Arduino Nano through the 5V Pin, it's critical to ensure that the DC supply voltage is exactly 5V. This pin is non-regulated, meaning it does not have a voltage regulator, and providing more than 5V can damage your board.

Vin Pin: Alternatively, you can use the Vin Pin, which can handle a voltage range of 7-12V. The Vin Pin is connected to the onboard voltage regulator, making it a regulated input. This means it can manage a broader range of voltages without risking damage to the board.

It's important to note the difference between these two options to avoid potential damage to your Arduino Nano. Be cautious about the voltage you apply to the 5V Pin to ensure safe and reliable operation.

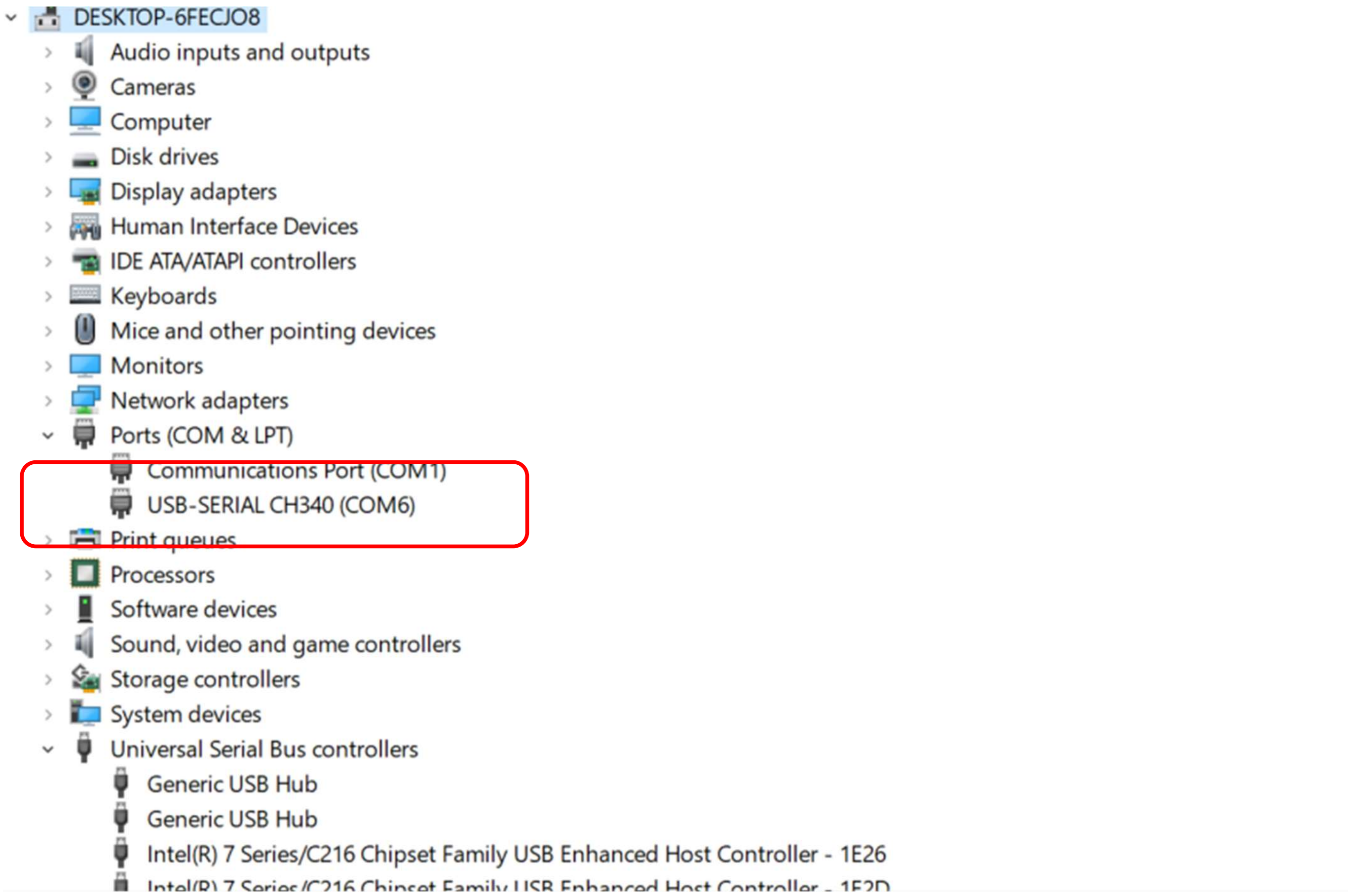


Understanding these power requirements is vital for ensuring the reliable and safe operation of your Arduino Nano projects. By choosing the right power source and managing voltage effectively, you can maximize the performance and longevity of your creations.

Chapter 5. Troubleshooting and Debugging

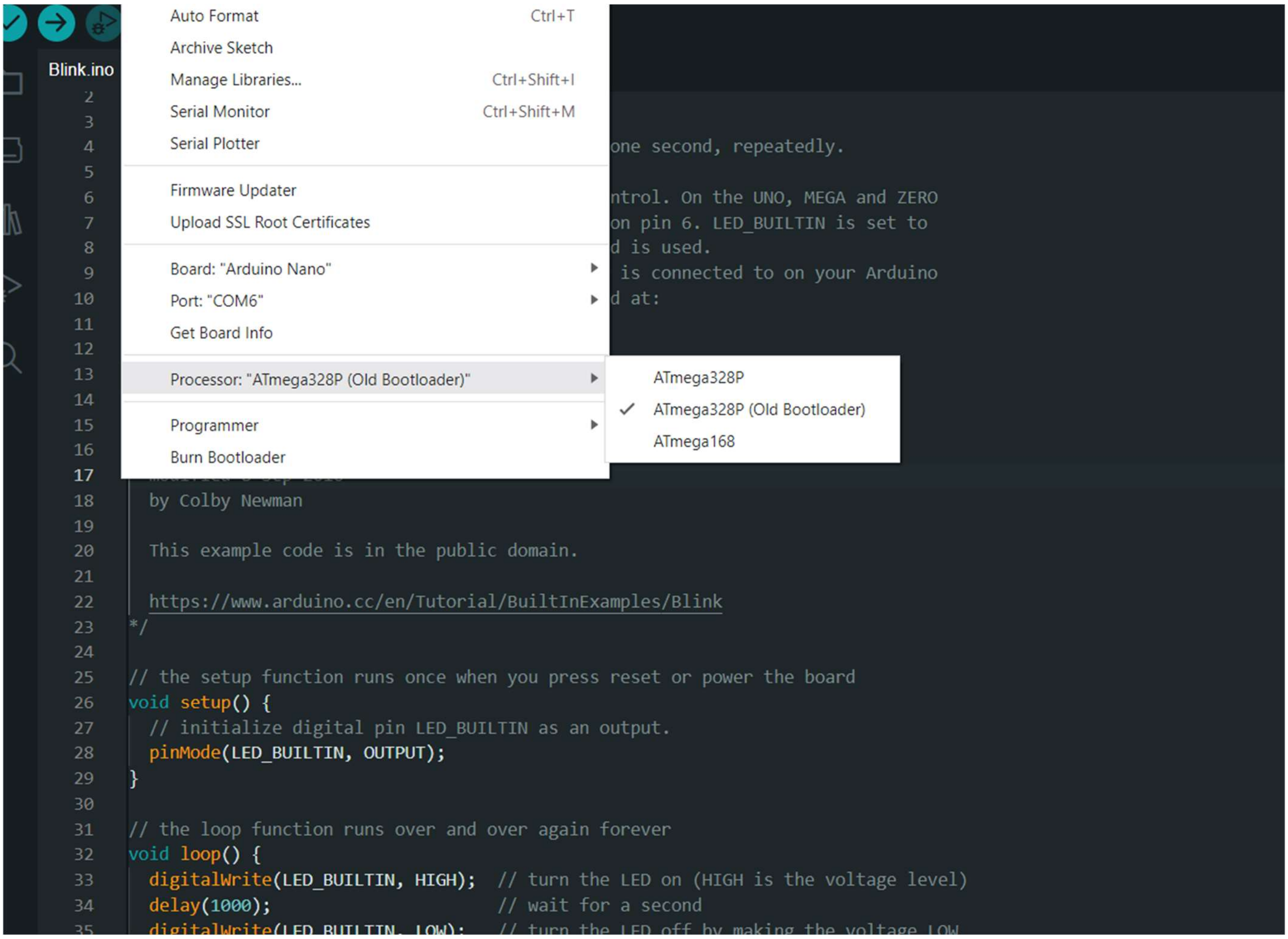
When troubleshooting and debugging issues with code uploads, follow these effective steps:

✓ **Verify CH340 Driver Installation:** First, ensure that the CH340 Driver is correctly installed on your computer and matches your PC's operating system. You can check this by listening for a connection sound when you connect your board to your computer or by confirming the presence of CH340 in the Device Manager when the board is connected.



DESKTOP-6FECJO8
- Audio inputs and outputs
- Cameras
- Computer
- Disk drives
- Display adapters
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
  - Communications Port (COM1)
  - USB-SERIAL CH340 (COM6)
- Print queues
- Processors
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers
  - Generic USB Hub
  - Generic USB Hub
  - Intel(R) 7 Series/C216 Chipset Family USB Enhanced Host Controller - 1E26
  - Intel(R) 7 Series/C216 Chipset Family USB Enhanced Host Controller - 1E2D

✓ **Check IDE Settings:** Double-check that your IDE settings are correct. Ensure that you have selected the appropriate board from the "Tools" menu under "Boards" (Arduino Nano) and that you've chosen the correct processor (ATmega328P or Old Bootloader). You can swap between these settings to confirm. Also, make sure the correct port is selected.



Board: "Arduino Nano"

Port: "COM6"

Processor: "ATmega328P (Old Bootloader)"

✓ ATmega328P (Old Bootloader)

```
one second, repeatedly.

ntrol. On the UNO, MEGA and ZERO
on pin 6. LED_BUILTIN is set to
d is used.
 is connected to on your Arduino
d at:

  by Colby Newman

  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
```

- ✓ **Inspect Connections:** Ensure that all connections are secure and correctly wired to prevent unexpected behaviours. Try disconnecting and reconnecting the USB connection before attempting to upload your code again.
- ✓ **Power Supply Verification:** Confirm that the power supply provides the correct voltage, as voltage fluctuations can lead to instability.
- ✓ **Avoid Pin and Port Conflicts:** Be cautious of pin and port conflicts that can cause problems. Ensure that no other programs or applications are running in the background that could conflict with the upload process.

Embrace Creative Problem-Solving: Be open to creative problem-solving when you encounter challenges in your projects.

Seek Community Support: If you're facing complex issues, don't hesitate to seek help from online resources and forums. Additionally, feel free to reach out to us at admin@maisonup.in or call us at 8800854049 for unconditional support.

These troubleshooting tips are valuable for overcoming challenges in your Arduino Nano projects, ensuring a smooth and successful development process.

Chapter 6. Safety Precautions

Prioritize safety to prevent accidents and protect both yourself and your equipment:

- ✓ Avoid static electricity by grounding yourself before handling components.
- ✓ Maintain a proper workspace with eye protection when necessary.
- ✓ Take precautions when soldering, including ventilation for fumes.
- ✓ Respect component ratings and pay attention to polarity.
- ✓ Handle batteries with care, following manufacturer guidelines.
- ✓ Disconnect power during wiring changes and component handling.
- ✓ Keep fire safety equipment nearby when soldering or working with high-power components.
- ✓ Consult datasheets for safety precautions and stay informed.

By following these safety guidelines, you can ensure a secure and productive Arduino Nano project experience.

Chapter 7. Inspiration for Arduino Projects

Finally, embrace your creative spirit. The Arduino Nano is a canvas for your imagination. Start with small projects, learn from the community, collaborate with others, and document your journey. The process of making and learning is as important as the result. So, experiment, enjoy, and keep exploring the endless possibilities with your Arduino Nano.