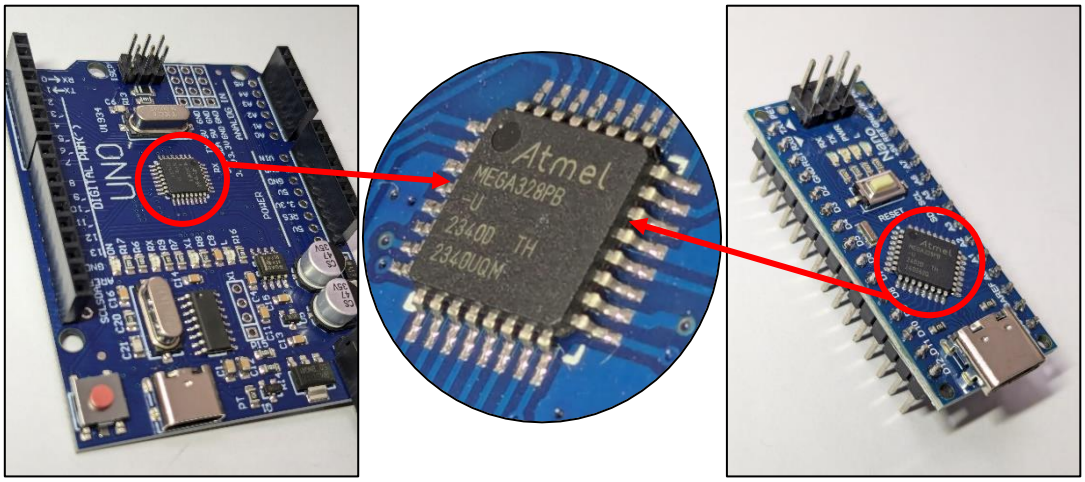
SYNACORP TECHNOLOGIES SDN. BHD. (1310487-K)
No.25 Lorong 1/SS3, Bandar Tasek Mutiara,
14120 Simpang Ampat, Penang, Malaysia.
T: +604.586.0026 F: +604.586.0026
WEBSITE: www.synacorp.my EMAIL: sales@synacorp.my

Using Compatible Arduino Uno R3 and Nano ATmega328PB model IC with Arduino IDE



Introduction:

Due to Arduino's open-source nature, many companies and individuals have created their own compatible boards. These boards often offer additional features or performance improvements while maintaining compatibility with the Arduino IDE.

Some popular Arduino board examples include:

- Arduino Uno Rev3
- Arduino Nano
- Arduino Mega

A). USB Driver Issues

The compatible Arduino models might use slightly different components to cut the product cost. Most manufacturers use the 3rd party USB to serial chip. This different chip requires different drivers to work. OS like Windows 10 and 11 will automatically install the driver once connected to the internet through Windows Update. Official Arduino Uno and Mega boards use the Atmel **ATmega16U2** chip, and the driver usually get installed automatically with the **Arduino IDE**.

Mostly used USB to Serial chip and driver:

- **ATmega16U2** (Atmel) - https://www.arduino.cc/en/software ***(Driver comes with IDE)***
- **CH340** Series (WCH) - https://www.wch-ic.com/downloads/CH341SER_ZIP.html
- **CP2102** (Silicon Labs) - https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers
- **FT232RL** (FTDI Chip) - https://ftdichip.com/drivers/vcp-drivers/
- **CH9102** (WCH) - https://www.wch-ic.com/downloads/CH343SER_ZIP.html

B). Type of IC model

Official **Arduino Uno Rev3** and **SMD** edition MCU boards launch with **ATmega328P**, this IC commonly comes in **DIP-28P3** and **SMD-32A** package. Some **Arduino Uno** and **Nano compatible boards** use the newer IC model such as **ATmega328PB** in **SMD package**.

This IC is not a drop-in replacement for **ATmega328P**, but a newer device. However, the functions are backward compatible with the existing **ATmega328** series functions according to **Atmel**. This newer model also comes with **four additional GPIO's Pin** functionalities.

C). Bootloader Issues

By using this **ATmega328PB** IC, user cannot flash the default **ATmega328P** based on **'Optiboot' bootloader**. This is because the **Signature bytes address ID** for both types of **ATmega328's** IC is different; the flashing process will be aborted. The programmer that **Arduino IDE** used, which is **AVRDUDE** will read the Signature byte ID to identify which IC being programmed or flashed. Since **ATmega328PB** never intended to being used with the official Arduino boards, they never released the official bootloader for this IC.

Boards	IC	Signature bytes address ID
*Standalone IC (no board)	ATmega328	0x1E9514
Official Uno Rev3 & Nano	ATmega328P	0x1E950F
Compatible Uno & Nano	ATmega328PB	0x1E9516

There are two custom bootloader that can be used with ATmega328PB based IC, the "**ATmega328PB with bootloader**" and "**MiniCore**". The "**ATmega328PB with bootloader**" are based on the official **Optiboot** bootloader, while the "**MiniCore**" are based on the newer **Urboot** bootloader. Both are tested with **Arduino IDE** version **1.8.19 (Legacy)** and 2.3.2 **(New)**. Before flashing the bootloader, **custom board installation** thru Arduino IDE must be done.

Differences	
ATmega328PB with bootloader	**MiniCore**
Based on Optiboot	Based on Urboot
No extra features (work as standard IC)	Unlock few additional IC extra features
Most compatible (direct port)	Actively in development
Only for ATmega328PB	All ATmega8, 48, 88, 168,328P, 328PB

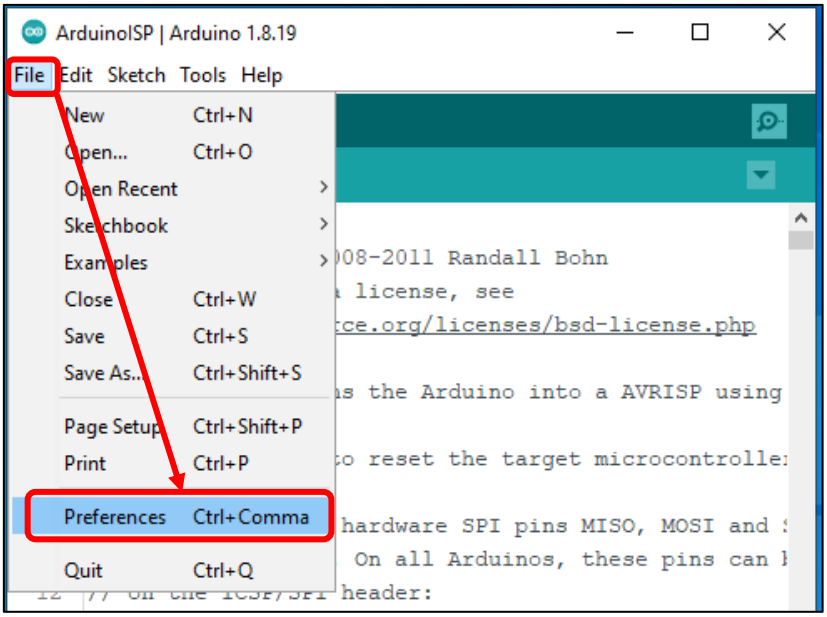
D). Flashing Bootloader, requirements

1. PC (with internet connection to download & install MiniCore)
2. Arduino Uno / Nano / Mega (any **Arduino** AVR base board **as Programmer**)
3. Arduino Uno / Nano (**to be flashed**, as a **Target** comes with **ATmega328PB**)
4. Jumper Wires - (via **ICSP** = Female to Female x 5pcs + Male to Female x 1pcs)
   (via **Digital + Power Pin** = Male to Male x 6pcs only)

E). Installing the Custom Board or Bootloader into Arduino IDE

1. Download and install the Arduino IDE program.
   (**Arduino IDE v1.8.19** **legacy** or **newer** **v2.3.2+**)

2. Once installed, **launch** the **Arduino IDE**. Navigate to **toolbar**, select **File** > **Preferences**.



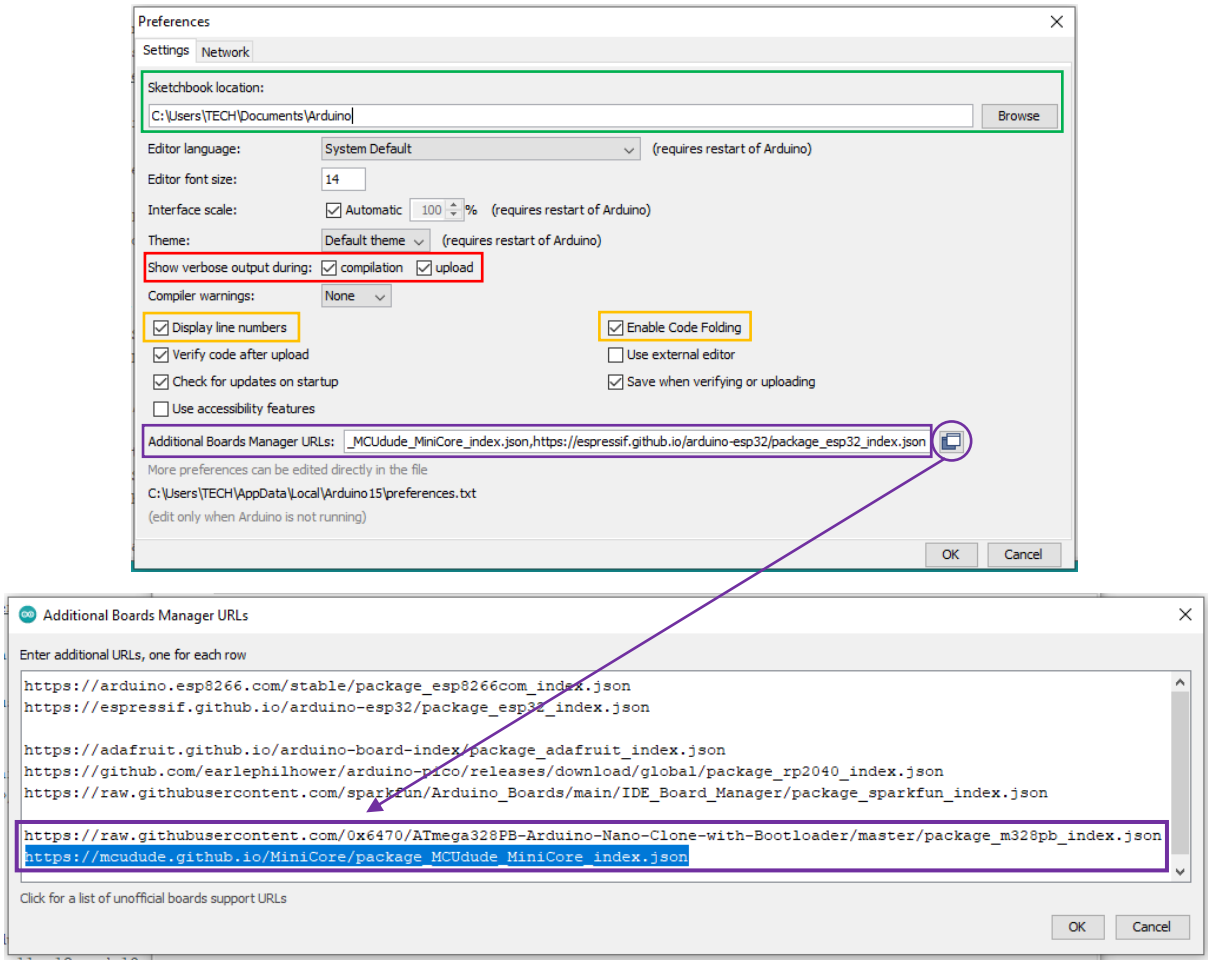
3. Once the **Preferences** tab shows up. Follow the setting from image attached below.
   a. Check the **Show verbose output** check box on **Compilation** and **Upload**. This option is good for troubleshooting.
   b. Click the icon next to **Additional Board Manager URLs**, paste the GitHub repo link and click on **OK**: -

      (i). **ATmega328PB with bootloader -**
      "https://raw.githubusercontent.com/0x6470/ATmega328PB-Arduino-Nano-Clone-with-Bootloader/master/package_m328pb_index.json"

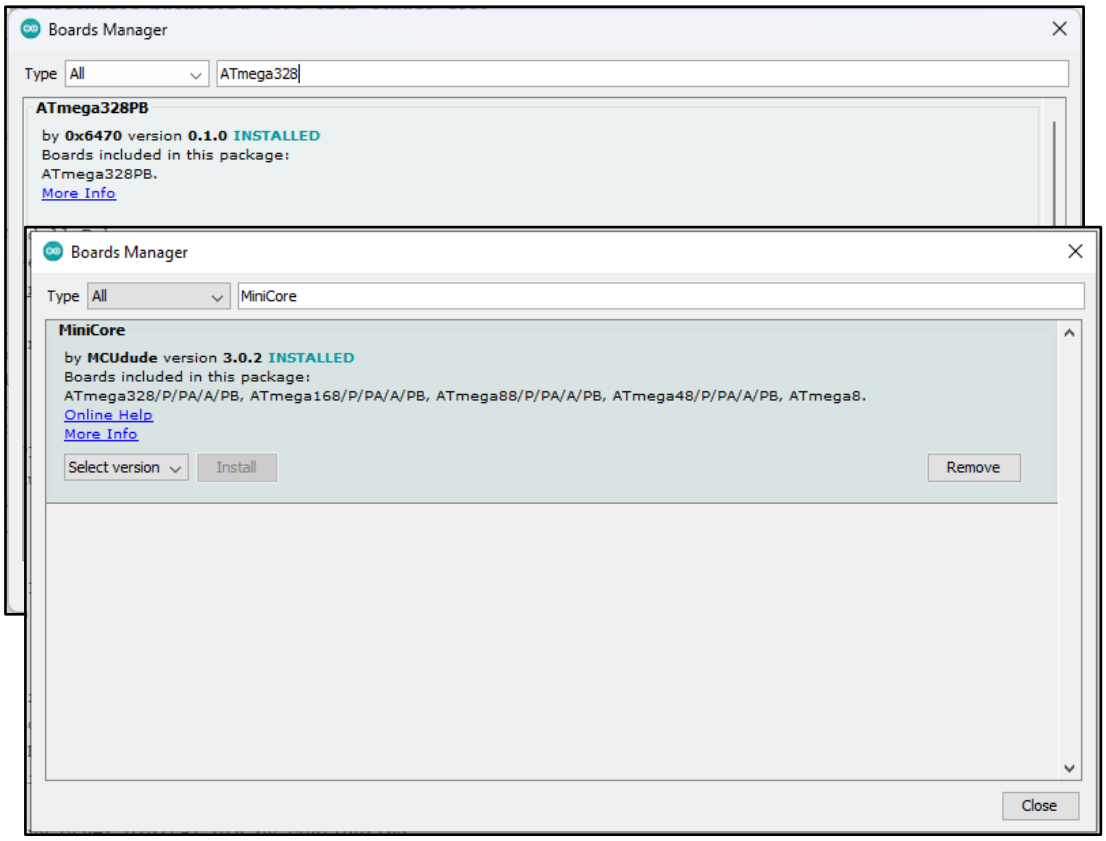
      or

      (ii). **MiniCore -**
      "https://mcudude.github.io/MiniCore/package_MCUdude_MiniCore_index.json"

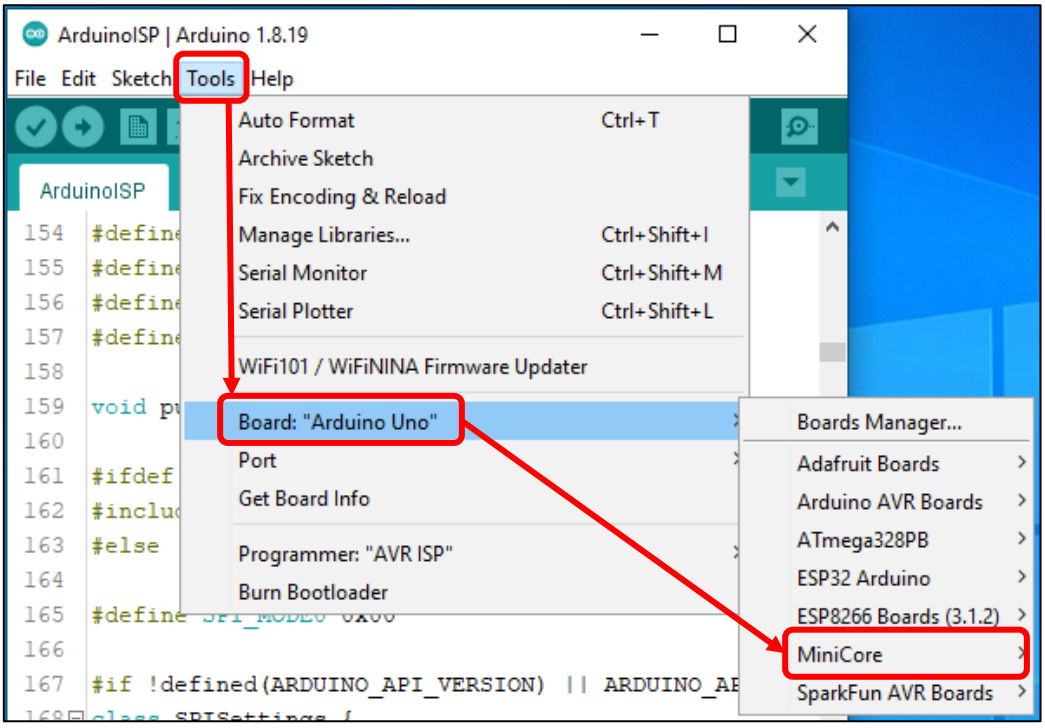
   c. **Display line numbers** and **Enable Code Folding** is optional, but it does help when the code is too long.
   d. **Sketchbook location** allows user to change the default compiled code location. Normally saved in user Document folder.



4. Once setting changes have been made, click on **OK** to close and save the **preferences**. It is recommended for users to restart the IDE when modifying the preferences parameter.
5. Make sure the PC is connected to the Internet. Then, click on the **toolbar > Tools > Boards: > Boards manager…**
6. New windows will pop-up (Boards Manager), on the search bar type in **“MiniCore”** or **“ATmega328PB”** and click on **Install** button. The board will begin download and get installed automatically.



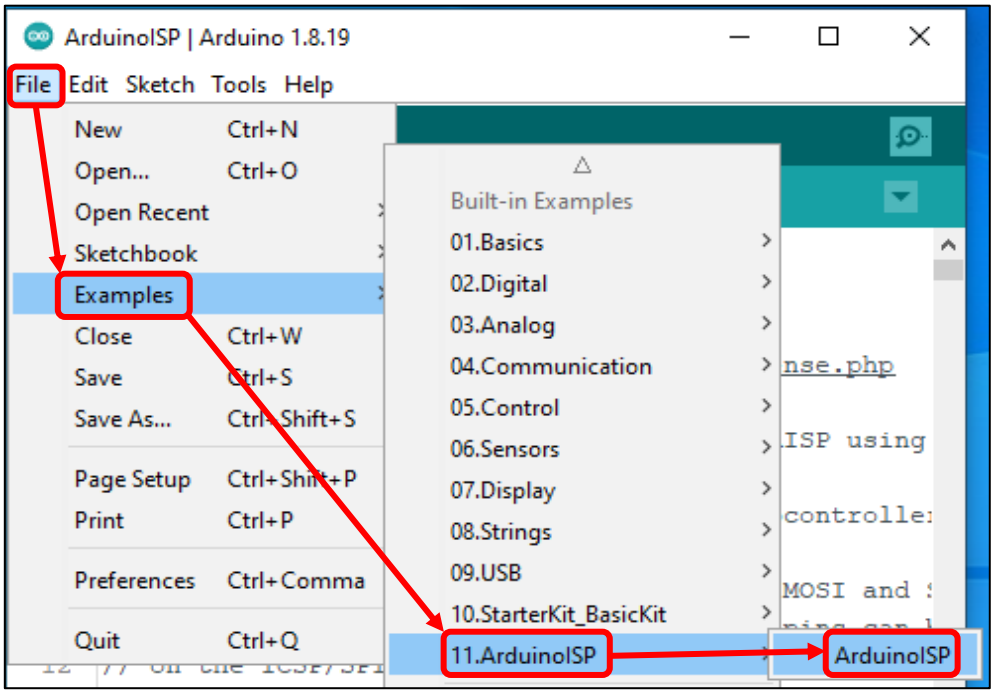
7. Once installed, close the **Boards Manager** window and **restart** the **Arduino IDE**. The **MiniCore** and **ATmega328PB** with bootloader option will appear on board selection menu. **Toolbar > Tools > Board: ….**



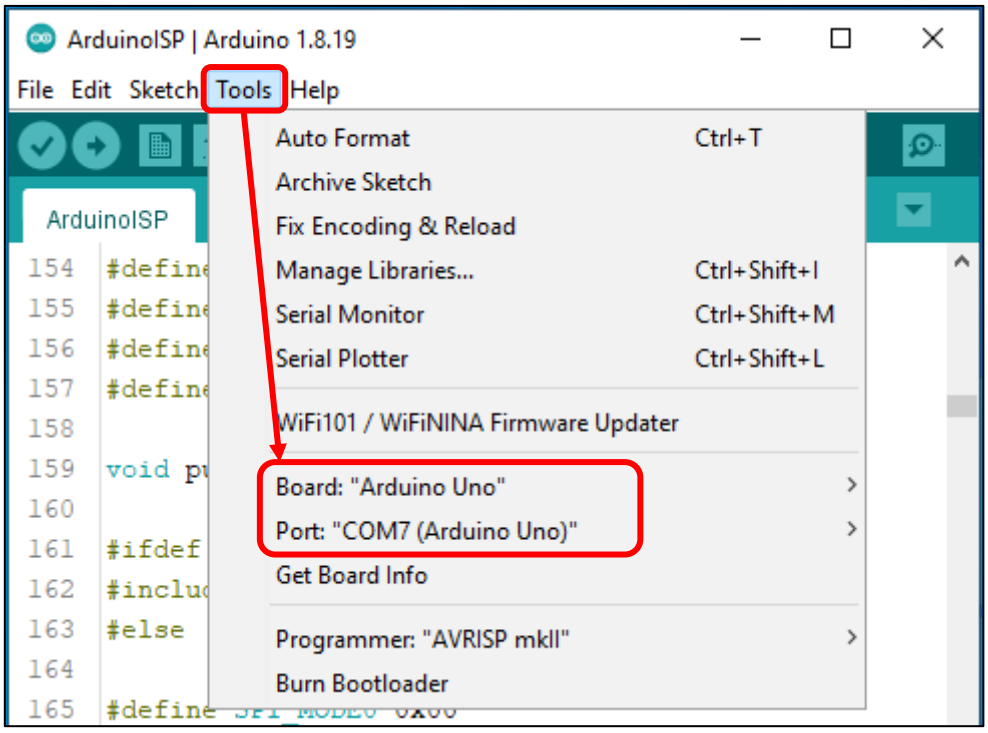
F). Wiring setup

Flashing bootloader requires 2 Arduino AVR board or AVR Programmer.

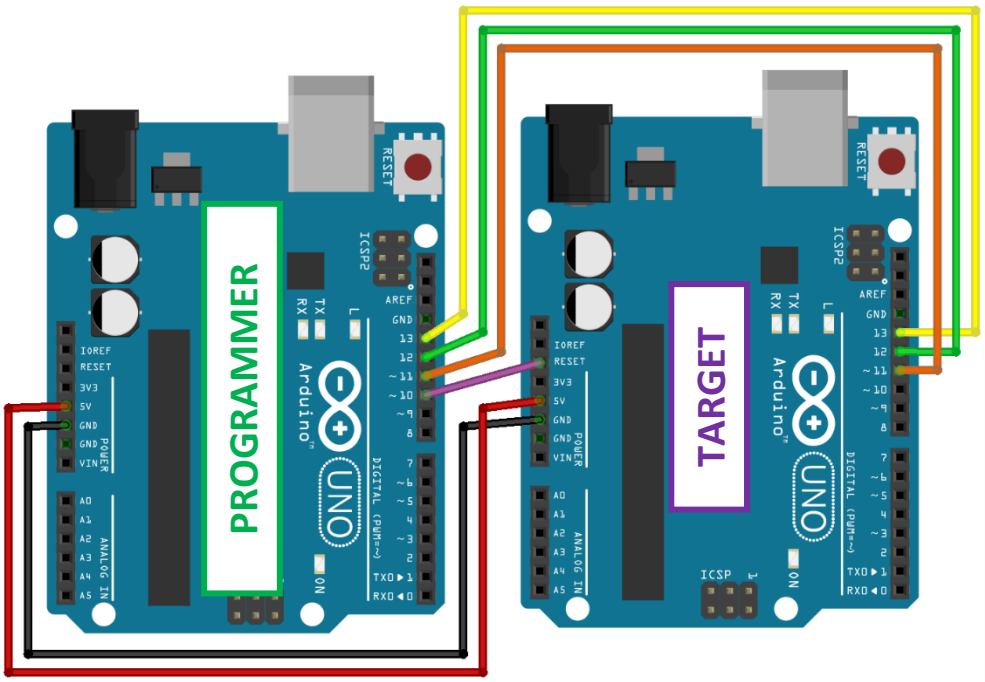
1. Upload the **ArduinoISP** code to the **'Programmer'** Arduino. Navigate to the **Toolbar > File > Examples > 11. ArduinoISP > ArduinoISP**.



2. In this guide the **programmer** Arduino used is **Arduino Uno**, so the board selection should be **Arduino Uno**.

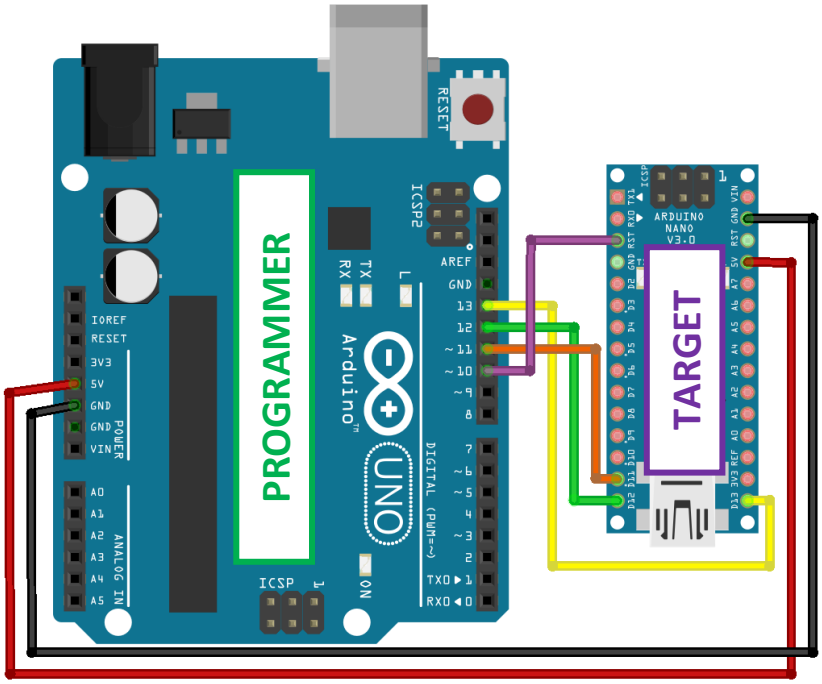


3. Once code uploaded to **Programmer** Arduino, there are 2 options for User to choose for Wiring both Arduino's.
    a. Wiring thru **Digital and Power Pin**
        i. Programmer = Uno **Target = Uno**



Example wiring between both Arduino Uno via Digital Pin

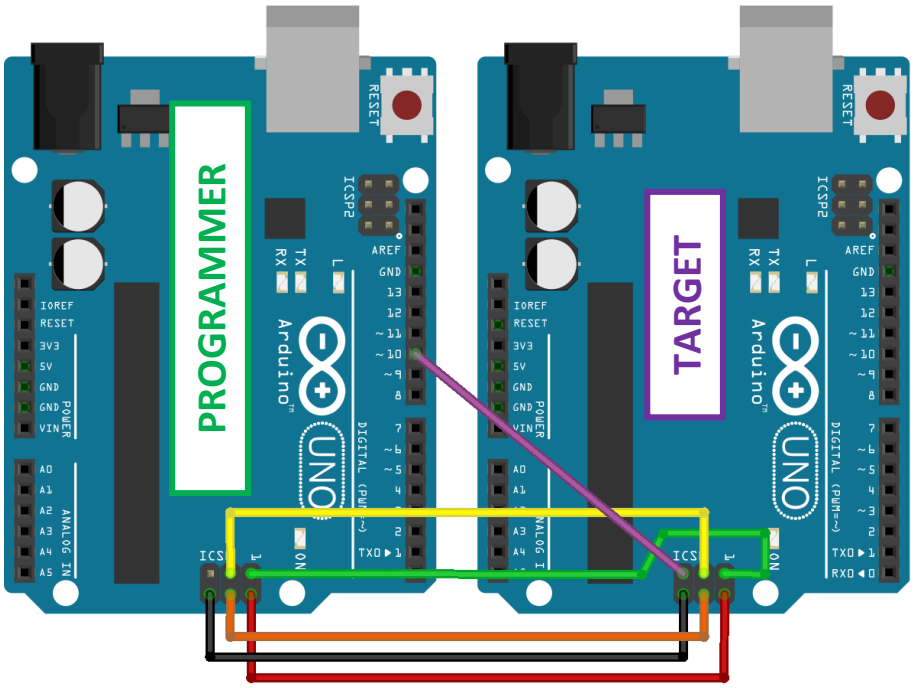
ii. Programmer = Uno **Target = Nano**



Example wiring between Arduino Uno and Nano via Digital Pin

b. Wiring thru **ICSP Header**

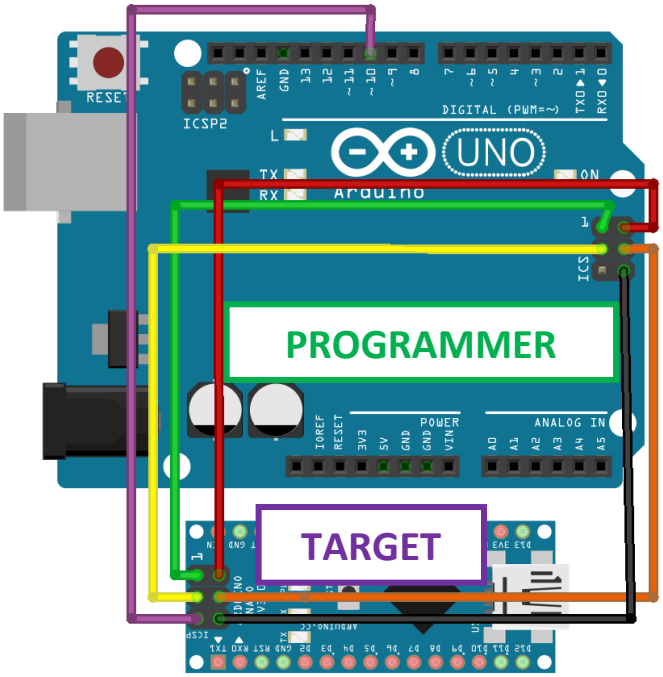
i. Programmer = Uno **Target = Uno**



Example wiring between both Arduino Uno via ICSP Pin

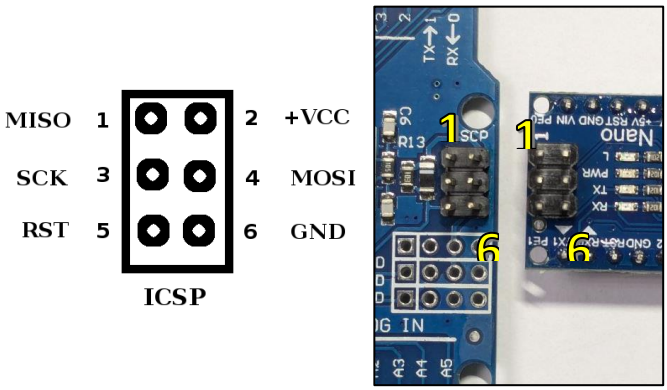
ii. Programmer = Uno

Target = Nano



Example wiring between Arduino Uno and Nano via ICSP Pin

4. Wiring table & ICSP Pinout

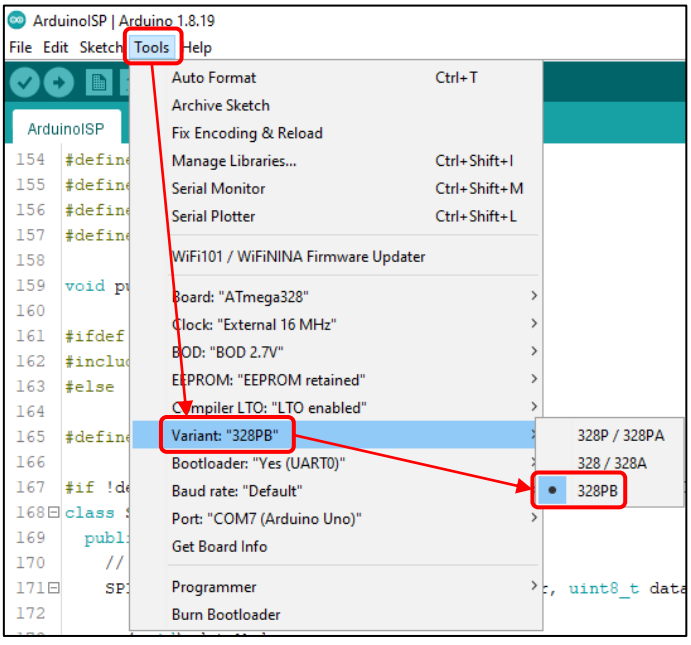


Programmer Board	Target Board
D10	RESET
COPI / MOSI / D11	COPI / MOSI / D11
CIPO / MISO / D12	CIPO / MISO / D12
SCK / D13	SCK / D13
5V	5V
GND	GND

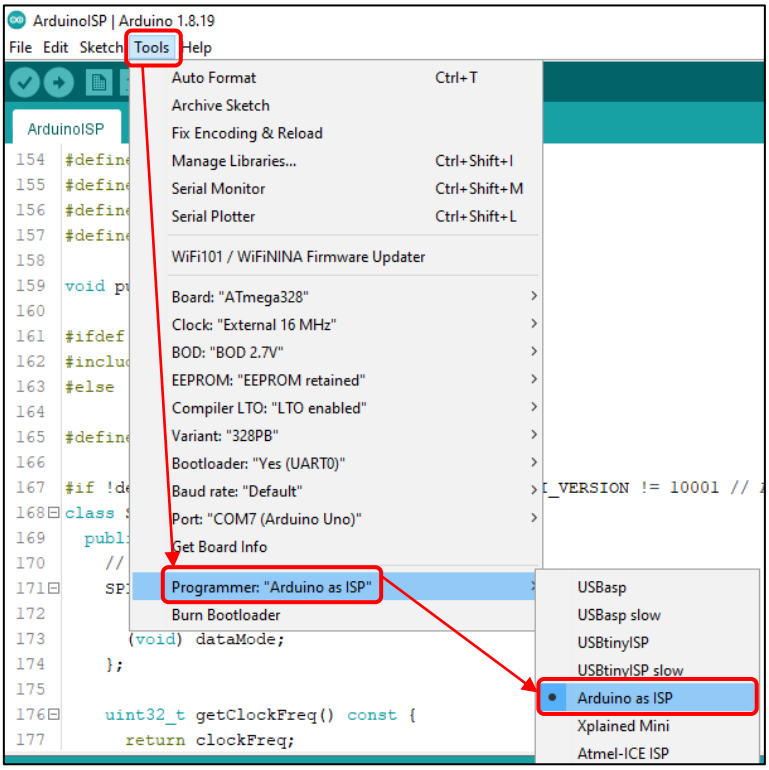
G-A). Flashing the MiniCore bootloader

Normally, flashing bootloader is only required when the bootloader is corrupted.

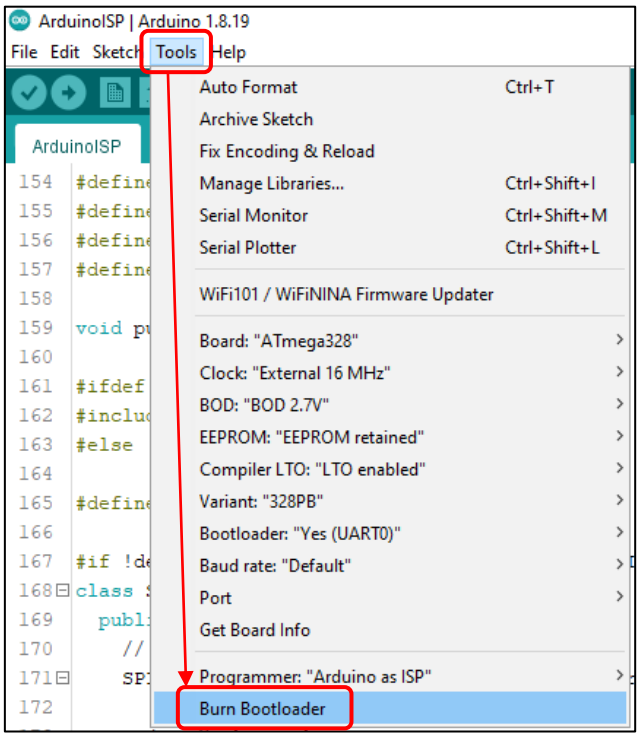
1. Once wiring setup is finished, select the **ATmega328** from Arduino IDE by navigate to **toolbar > Tools > Board: "…" > MiniCore > ATmega328**.
2. Follow the board setting as image below. **Tools > Variant: "328PB" >** select **328PB**.



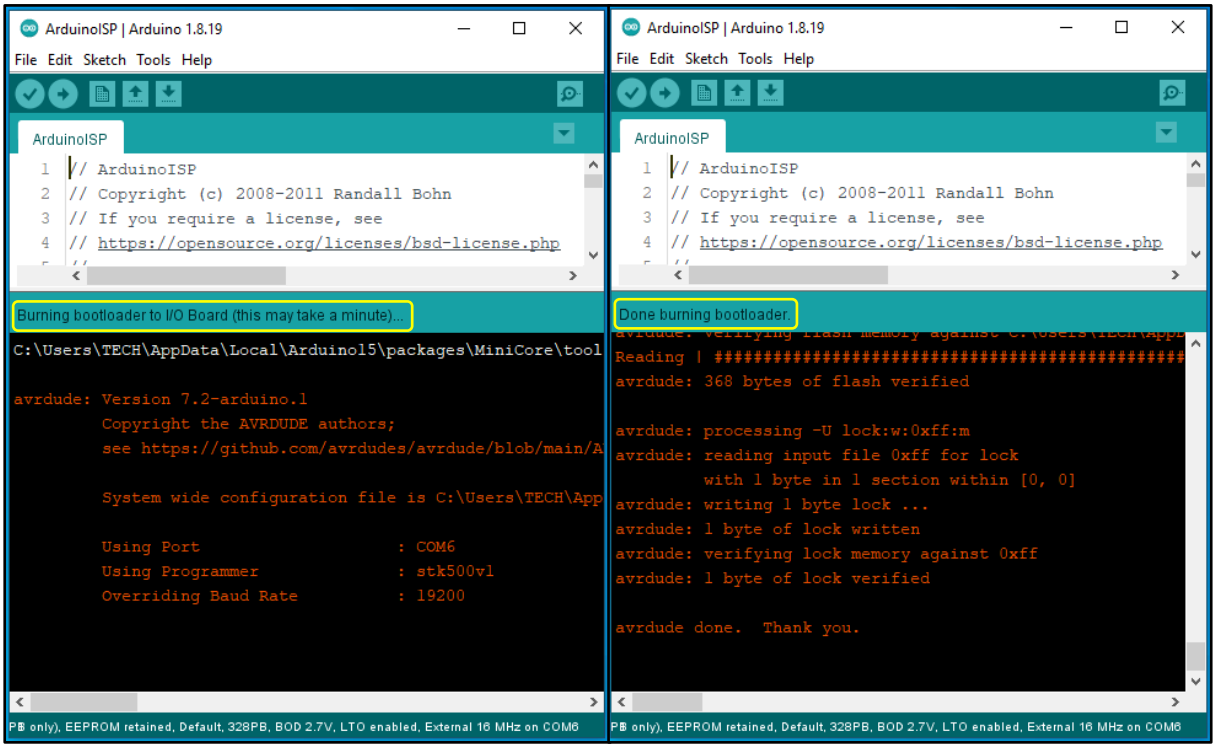
3. Then, again on the **toolbar** select **Tools > Programmer > Arduino as ISP**. Connect **only** the **programmer Arduino** to PC. The target Arduino will get supply power from the programmer and both boards should be powered ON.



4. Make sure to select correct **COM Port** (your **programmer** Arduino), click on **Burn Bootloader** to begin flashing process. Click on **Toolbar > Tools > Burn Bootloader**.

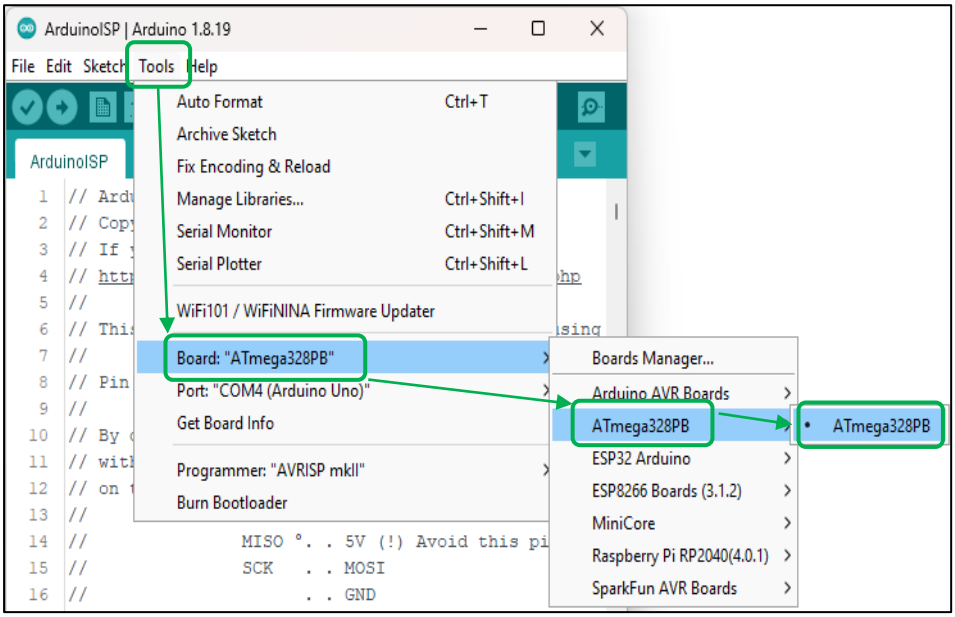


5. The flashing process should take around **3 to 10 seconds** to finish. Once finished, the status bar will show **"Done burning bootloader"**. **Done**! Disconnect both Arduino.

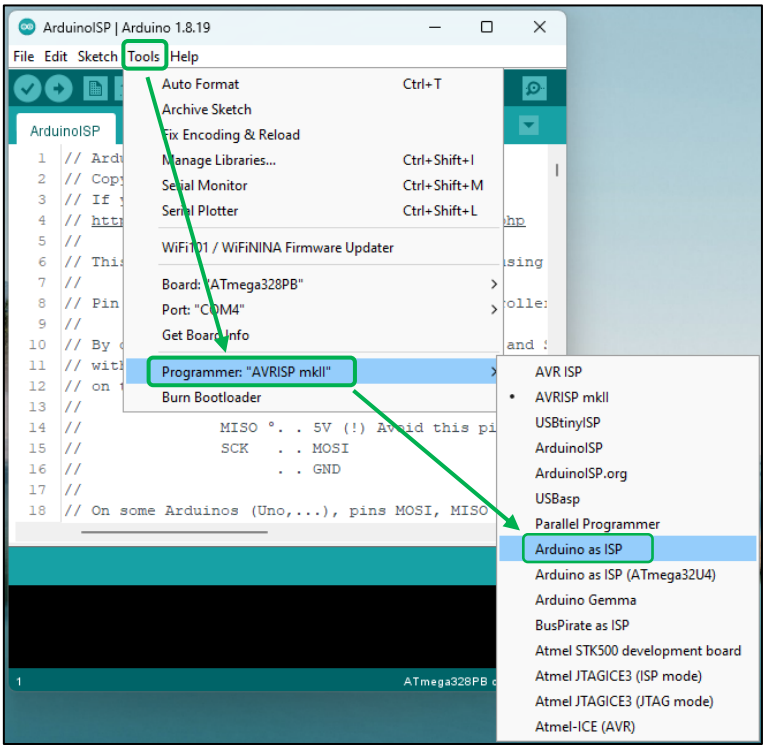


G-B). Flashing the ATmega328PB bootloader

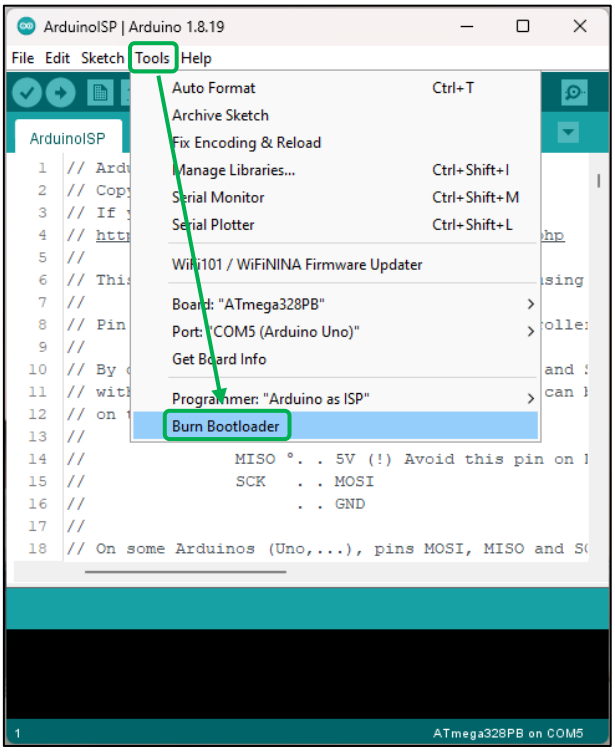
1. Once wiring setup is finished, select the **ATmega328** from Arduino IDE by navigate to **toolbar > Tools > Board: "…" > ATmega328PB > ATmega328PB.**



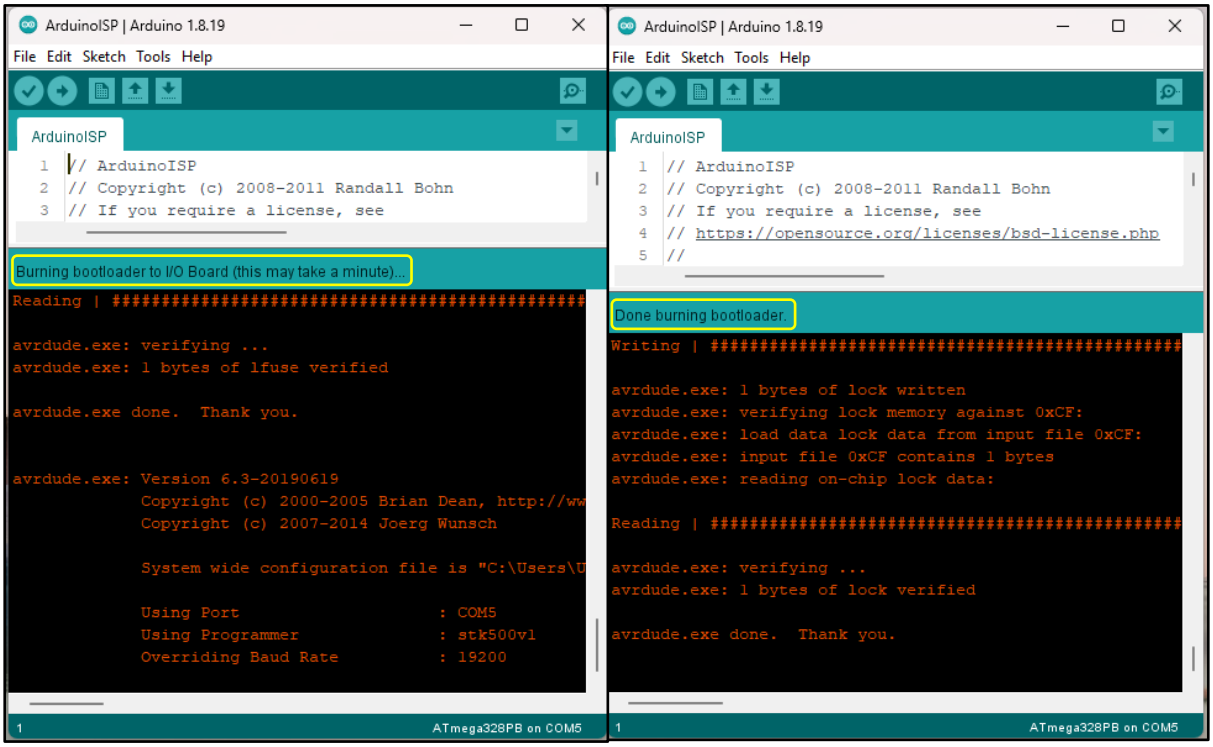
2. Then, again on the toolbar select **Tools > Programmer > Arduino as ISP**. Connect **only** the **programmer Arduino** to PC. The target Arduino will get supply power from the programmer and both boards should be powered ON.



3. Make sure to select correct **COM Port** (your **programmer** Arduino), click on **Burn Bootloader** to begin flashing process. Click on **Toolbar > Tools > Burn Bootloader**.



4. The flashing process should take around **3 to 10 seconds** to finish. Once finished, the status bar will show **"Done burning bootloader"**. **Done**! Disconnect both Arduino.

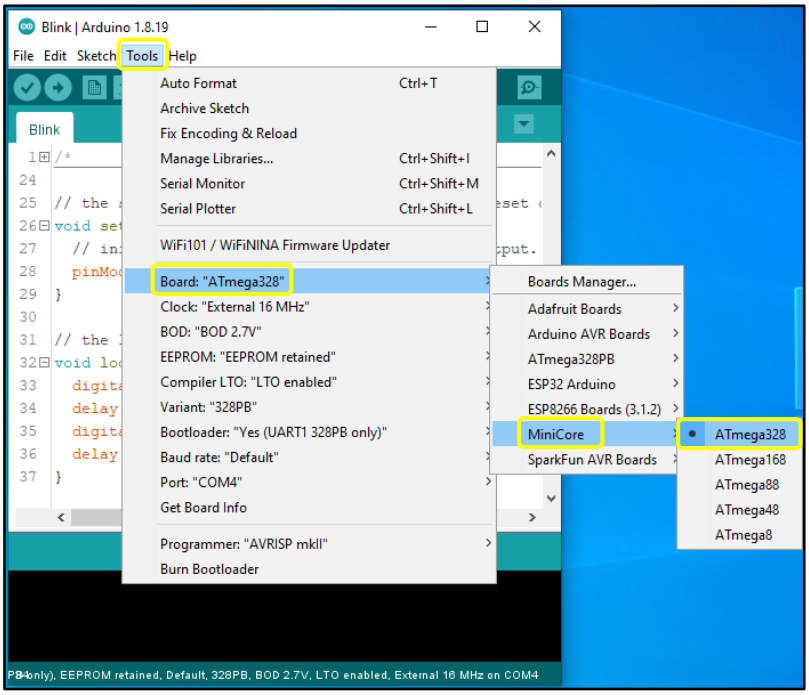


H). Uploading the code to the board that uses custom bootloader.

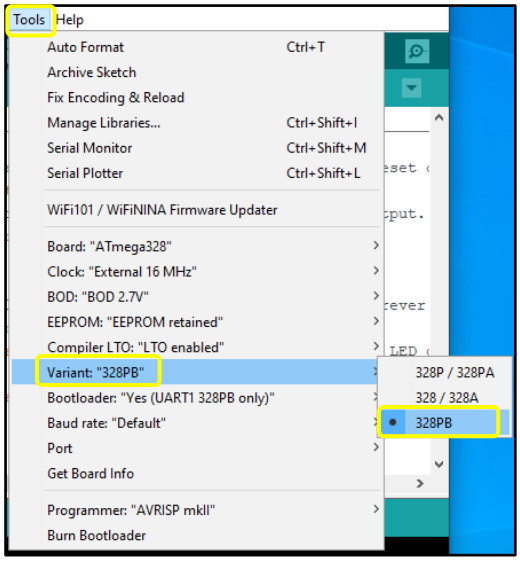
Based on which types of bootloaders that user flashed before, to upload the code user must select that type or it will fail.

MiniCore

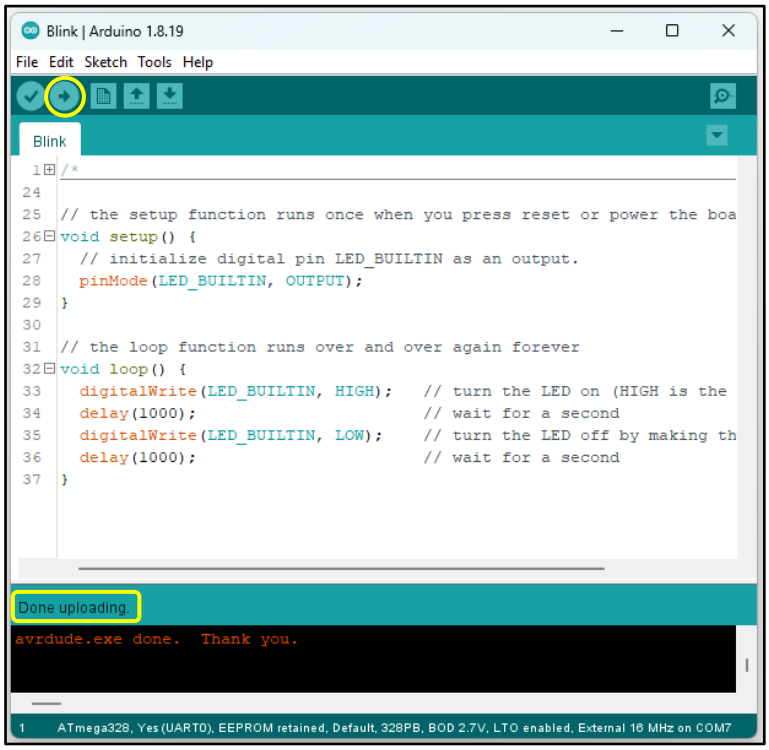
1. If **MiniCore** bootloader was flashed before, select the **Board: "…" > MiniCore > ATmega328"** and make sure the **Variant** is selected correctly as **"328PB"**.



2. Make sure to select the 328PB variant from the toolbar, **Tools > Variant: "…" > 328PB**. Make sure to select the correct **COM Port** for your Arduino, each PC assigns the COM port number differently.

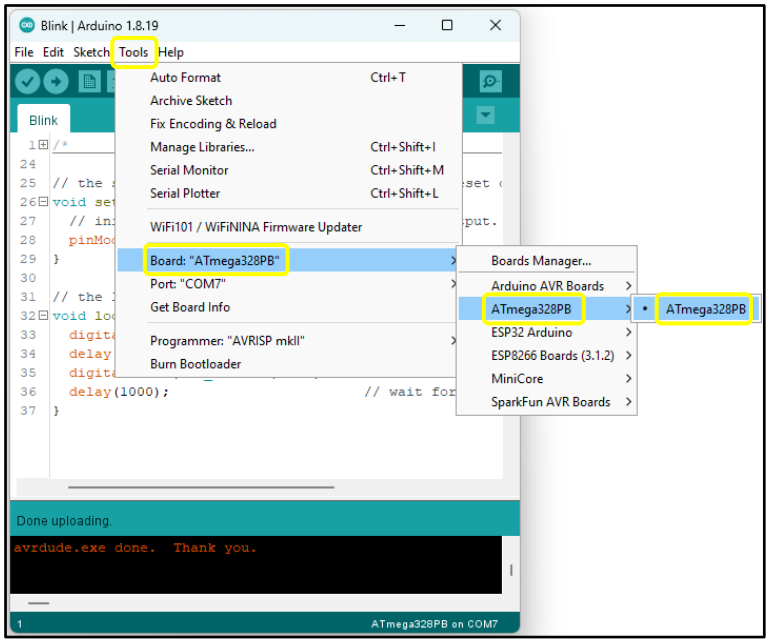


3. Click the **Upload** button as usual after selecting the board & COM port. Once code uploaded the Arduino IDE should display **Done Uploading**. Done! Your Arduino should run the latest code.

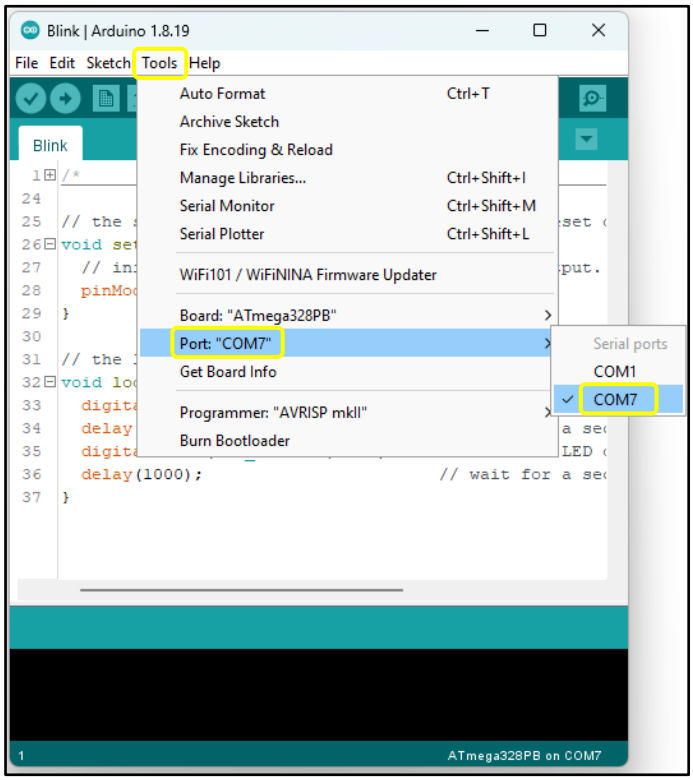


ATmega328PB

1. If **ATmega328PB** bootloader was flashed before, select the **Board: "…" > ATmega328PB > ATmega328PB.**



2. Make sure to select the correct **COM Port** for your Arduino, each PC assigns the COM port number differently.



3. Click the **Upload** button as usual after selecting the board & COM port. Once code uploaded the Arduino IDE should display **Done Uploading**. Done! Your Arduino should run the latest code.