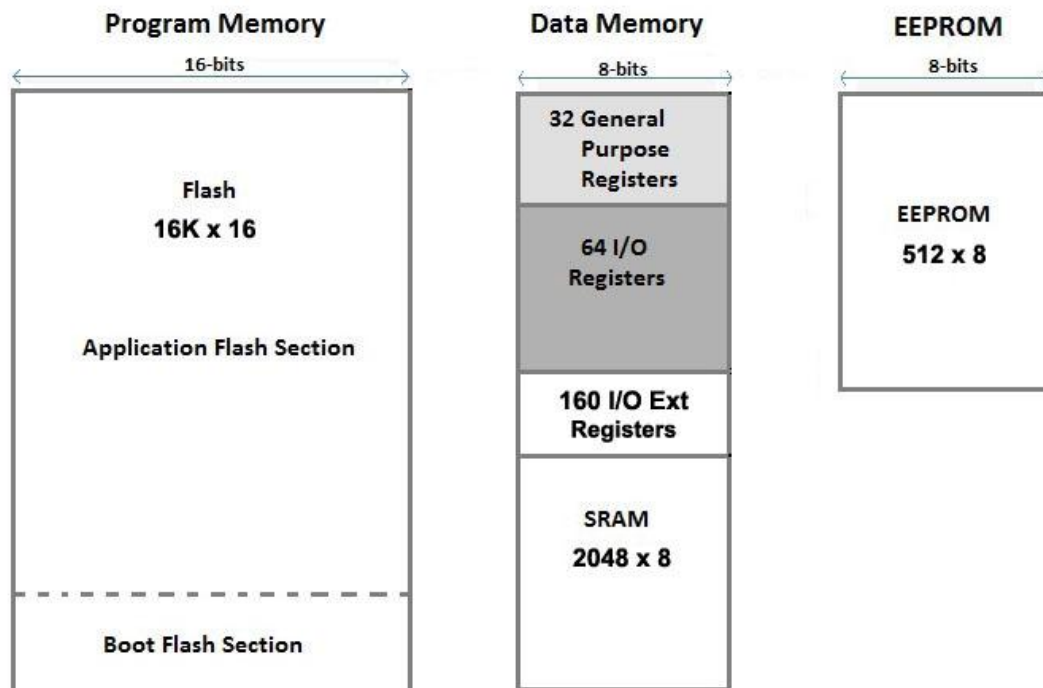


Arduino as ISP and Arduino Bootloaders

What makes an Arduino what it is? Many things, but one of the most important ones is the way every Arduino board is easily programmed with the Arduino Software (IDE). It is enough to connect it to the computer USB port and press the “Upload” icon to start a process that transfers your sketch into the Flash memory of the microcontroller.

This happens thanks to a special piece of code that is executed at every reset of the microcontroller and that looks for a sketch to be uploaded from the serial/USB port using a specific protocol and speed. If no connection is detected, the execution is passed to the code of your sketch.

This little (usually 512 bytes) piece of code is called the “Bootloader” and it is in an area of the memory of the microcontroller – at the end of the address space - that can’t be reprogrammed as a regular sketch and had been designed for such purpose.

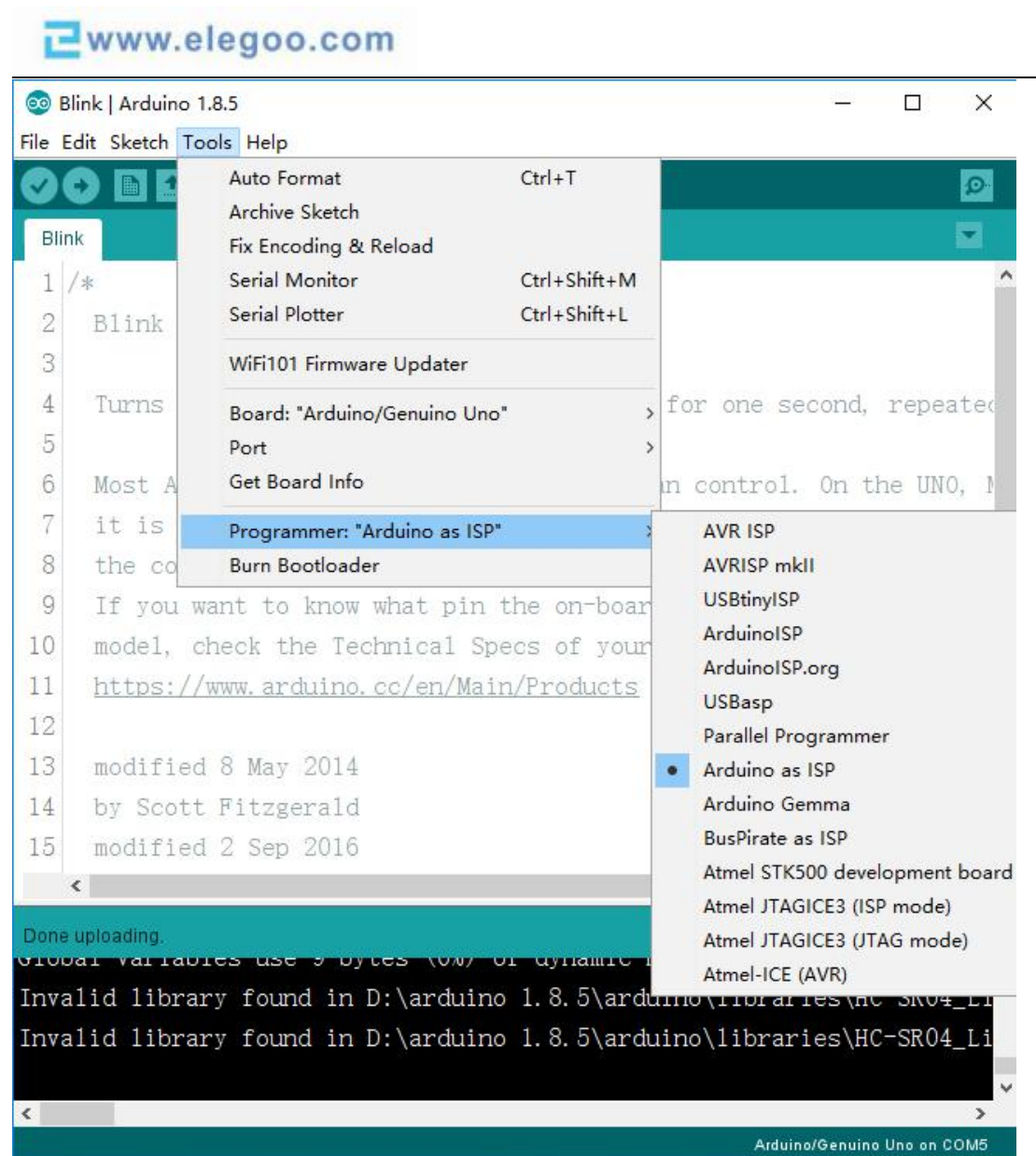


The Memory Map of an ATmega328P

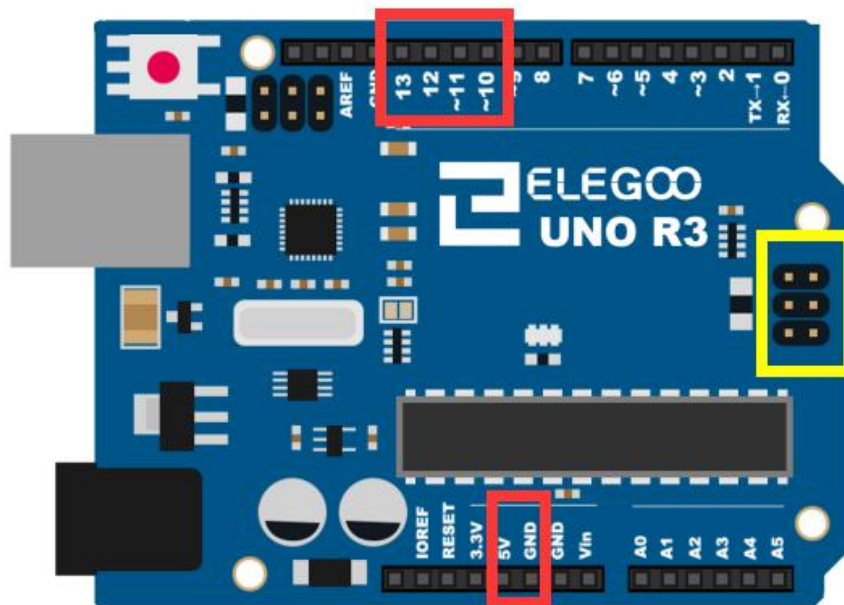
To program the bootloader and provide to the microcontroller the compatibility with the Arduino Software (IDE) you need to use an In-circuit Serial Programmer (ISP) that is the device that connects to a specific set of pins of the microcontroller to perform the programming of the whole flash memory of the microcontroller, bootloader included. The ISP programming procedure also includes the writing of fuses: a special set of bits that define how the microcontroller works under specific circumstances.

Use Arduino as ISP

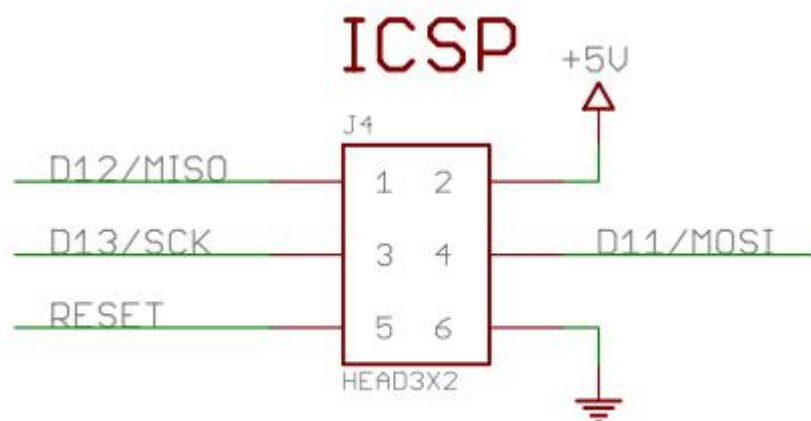
The whole process of loading the bootloader code and burning properly the fuses to make an ATmega microcontroller an “Arduino” is managed by the Arduino Software (IDE): it provides a specific menu item and allows you to use a variety of programming devices. Among the programmers, the “Arduino as ISP” is the cheapest and most practical solution to burn a bootloader on another Arduino board with ATmega, 32U4 or ATtiny.



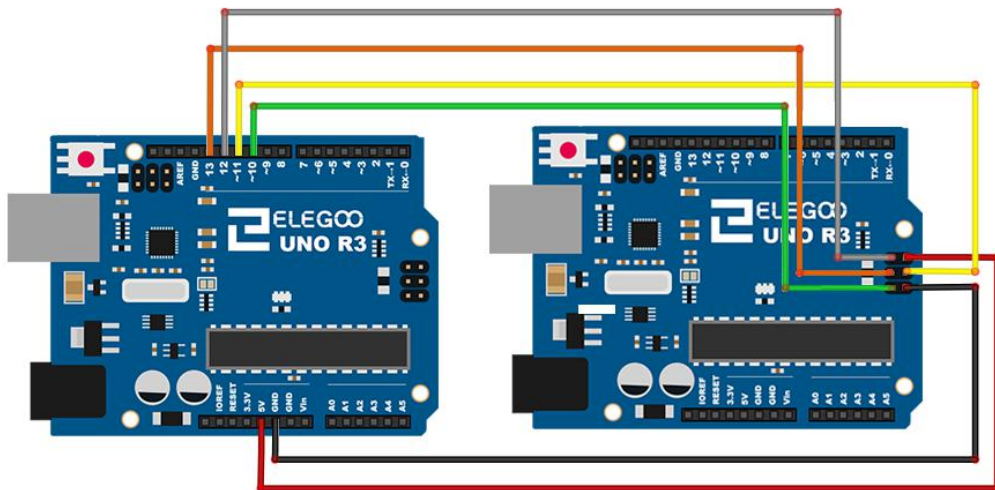
The programming process uses four data pins, VCC and GND. Three pins connect MISO, MOSI and SCK between the programming micro and the target micro, a fourth pin from the programming micro goes to the reset pin of the target.



These six pins are usually available in a connector labelled as “ISP” on the Arduino boards; this connector is used also by shields that rely on the SPI interface.



As you can imagine, you need to connect the programmer to the target board or chip with six jumpers. They can go from one ISP connector to the other one, with the exception of pin D10 from the programmer to the RESET of the target.



On some Arduino boards (Uno,...), pins MOSI, MISO and SCK are the same pins as digital pin 11, 12 and 13, respectively. That is why many tutorials instruct you to hook up the target to these pins. If you find this wiring more practical, have a define `USE_OLD_STYLE_WIRING`. This will work even when not using an Uno. (On an Uno this is not needed).

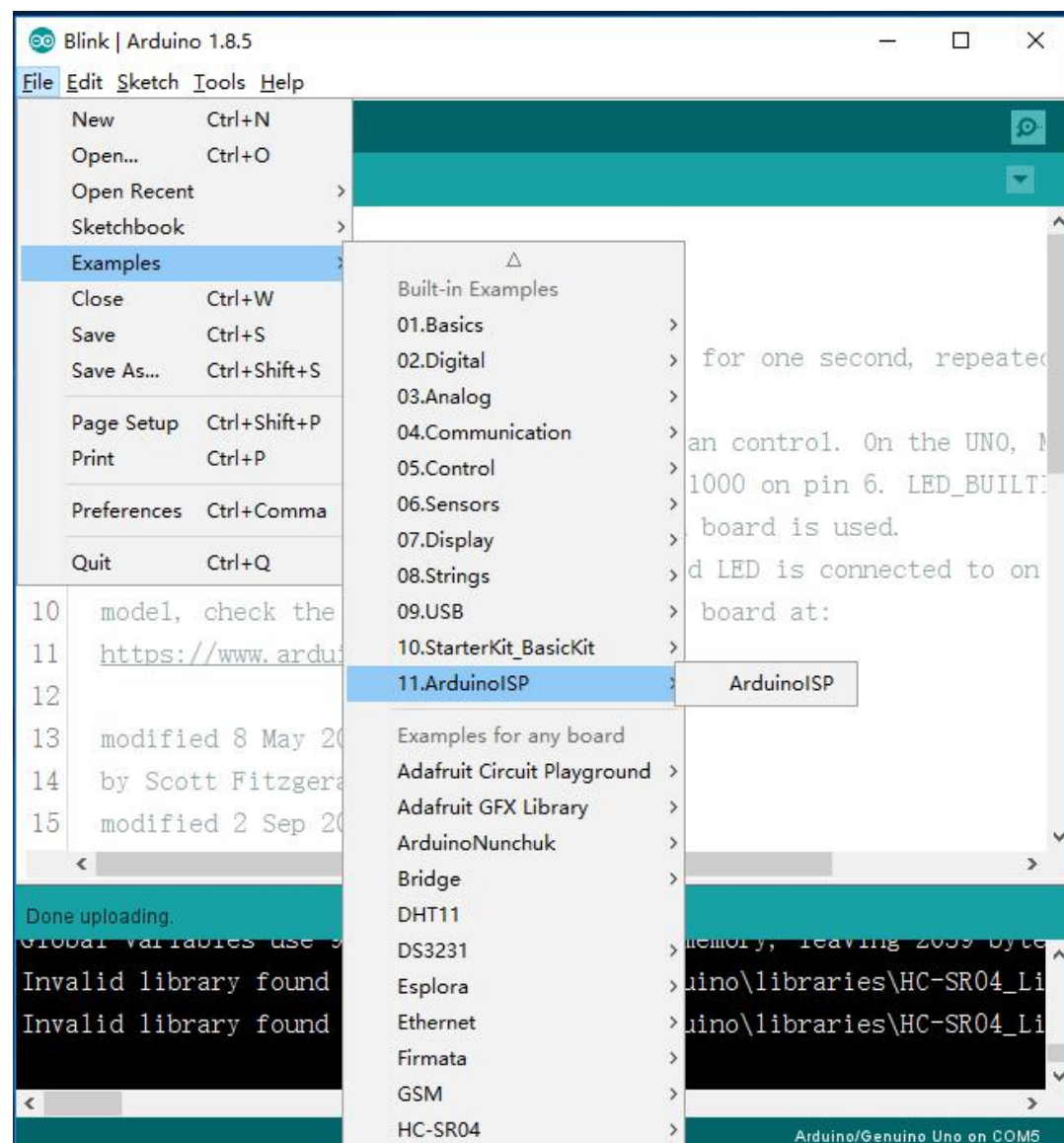
About voltages

The Arduino family of boards includes 5V and 3.3V devices. When using an Arduino that is not 5V tolerant (Due, Zero, ...) as the programmer, make sure to not expose any of the programmer's pins to 5V. A simple way to accomplish this is to power the complete system (programmer and target) at 3V3.

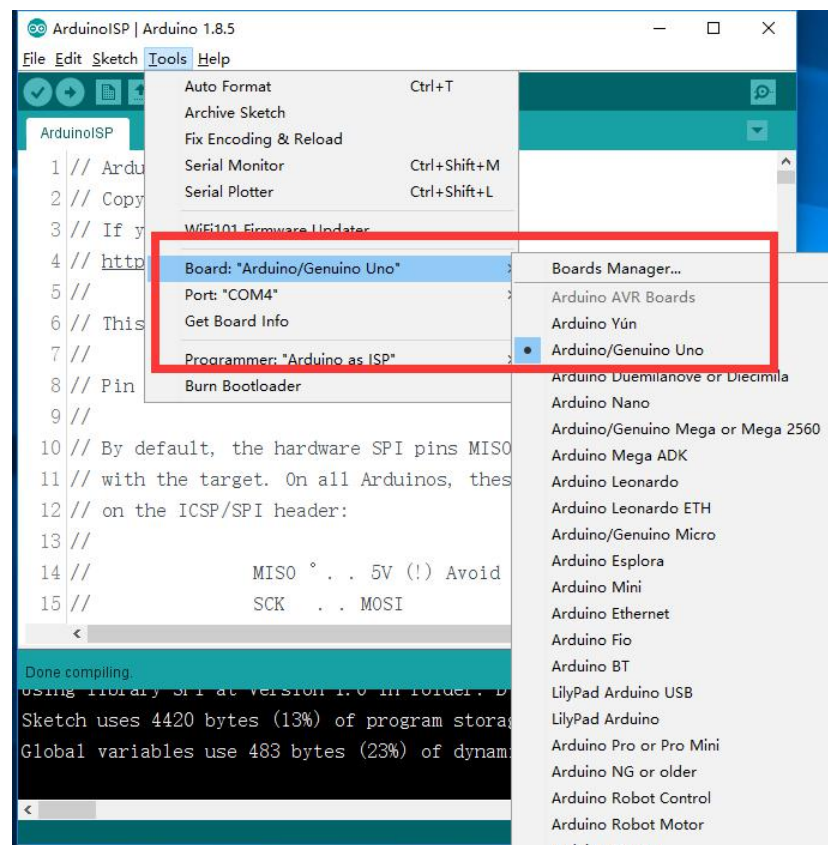
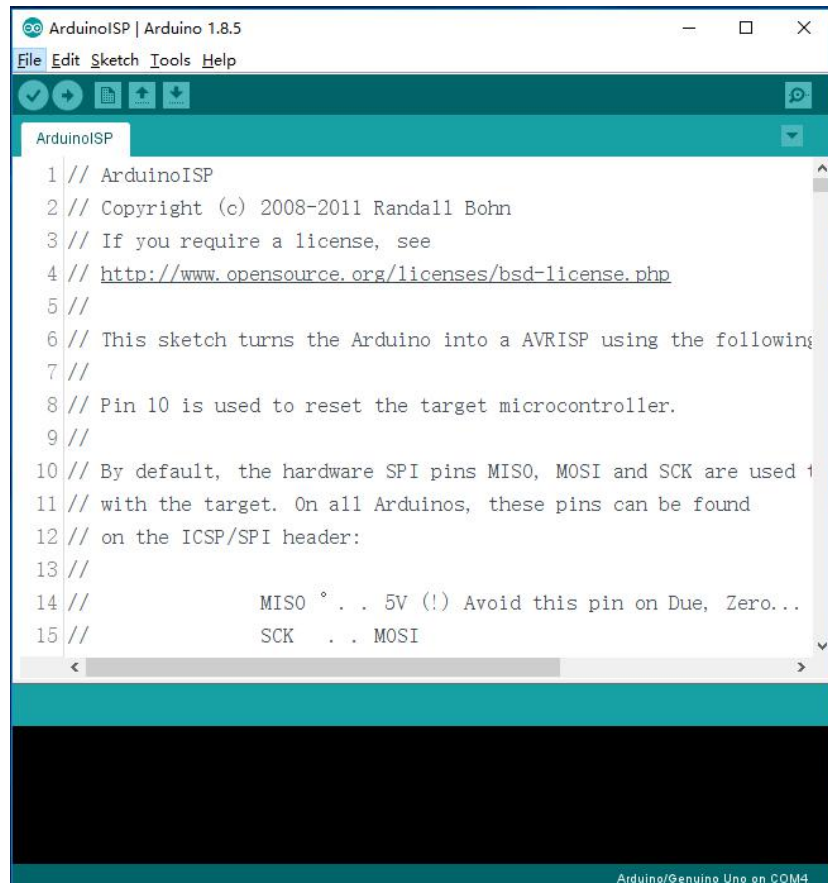
Note: Please do not connect to USB or power supply the boards while you set up the connections and wires. We also suggest that you first program the Arduino used as ISP programmer before you proceed with the wiring to the target board.

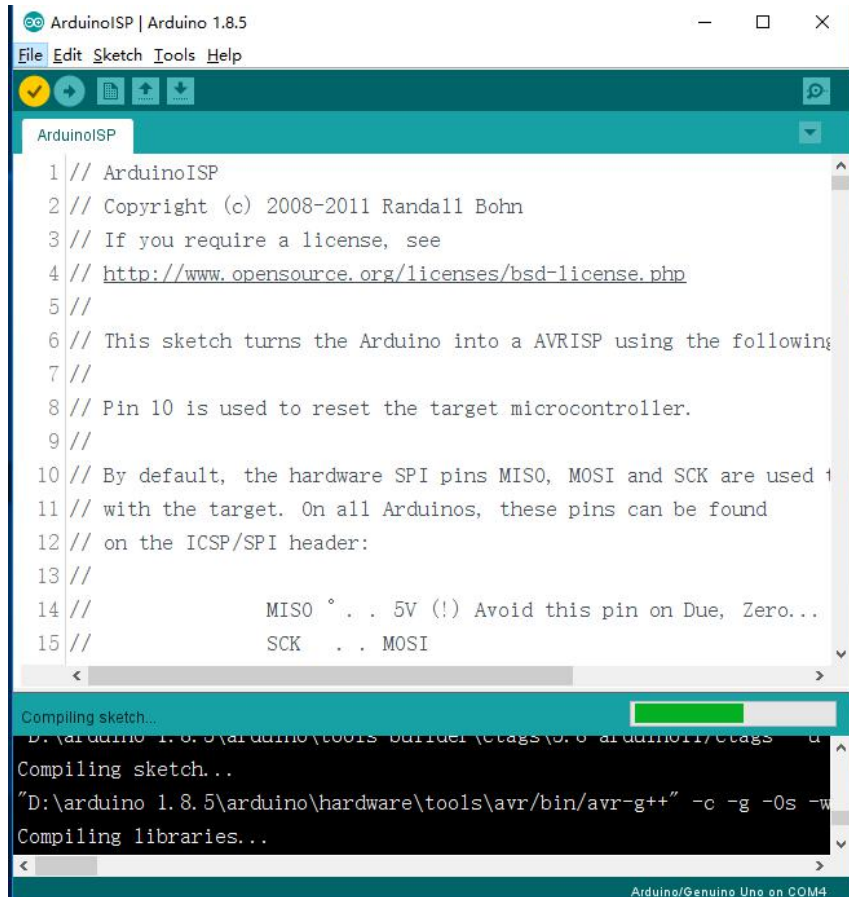
Load the sketch

The Arduino that you will use as programmer needs a specific sketch. You find it under Examples > 11. ArduinoISP > ArduinoISP .



To upload the sketch to your Arduino board – the one that you will use as the programmer – you need to select board type and port, then you can proceed as usual.





```

1 // ArduinoISP
2 // Copyright (c) 2008-2011 Randall Bohn
3 // If you require a license, see
4 // http://www.opensource.org/licenses/bsd-license.php
5 //
6 // This sketch turns the Arduino into a AVRISP using the following
7 //
8 // Pin 10 is used to reset the target microcontroller.
9 //
10 // By default, the hardware SPI pins MISO, MOSI and SCK are used
11 // with the target. On all Arduinos, these pins can be found
12 // on the ICSP/SPI header:
13 //
14 //           MISO  ° . . 5V (!) Avoid this pin on Due, Zero...
15 //           SCK   . . MOSI
  
```

Compiling sketch...

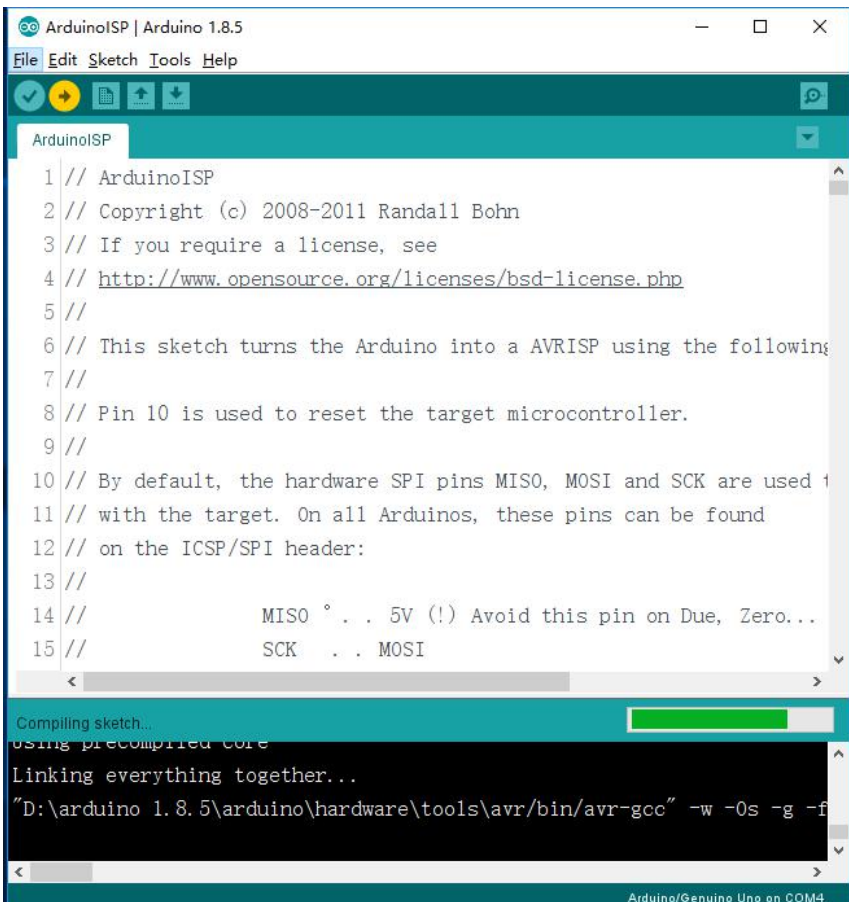
D:\arduino 1.8.5\arduino\tools\builder\ctags\5.8\arduino11\ctags -u

Compiling sketch...

"D:\arduino 1.8.5\arduino\hardware\tools\avr\bin\avr-g++" -c -g -Os -w

Compiling libraries...

Arduino/Genuino Uno on COM4



```

1 // ArduinoISP
2 // Copyright (c) 2008-2011 Randall Bohn
3 // If you require a license, see
4 // http://www.opensource.org/licenses/bsd-license.php
5 //
6 // This sketch turns the Arduino into a AVRISP using the following
7 //
8 // Pin 10 is used to reset the target microcontroller.
9 //
10 // By default, the hardware SPI pins MISO, MOSI and SCK are used
11 // with the target. On all Arduinos, these pins can be found
12 // on the ICSP/SPI header:
13 //
14 //           MISO  ° . . 5V (!) Avoid this pin on Due, Zero...
15 //           SCK   . . MOSI
  
```

Compiling sketch...

using precompiled core

Linking everything together...

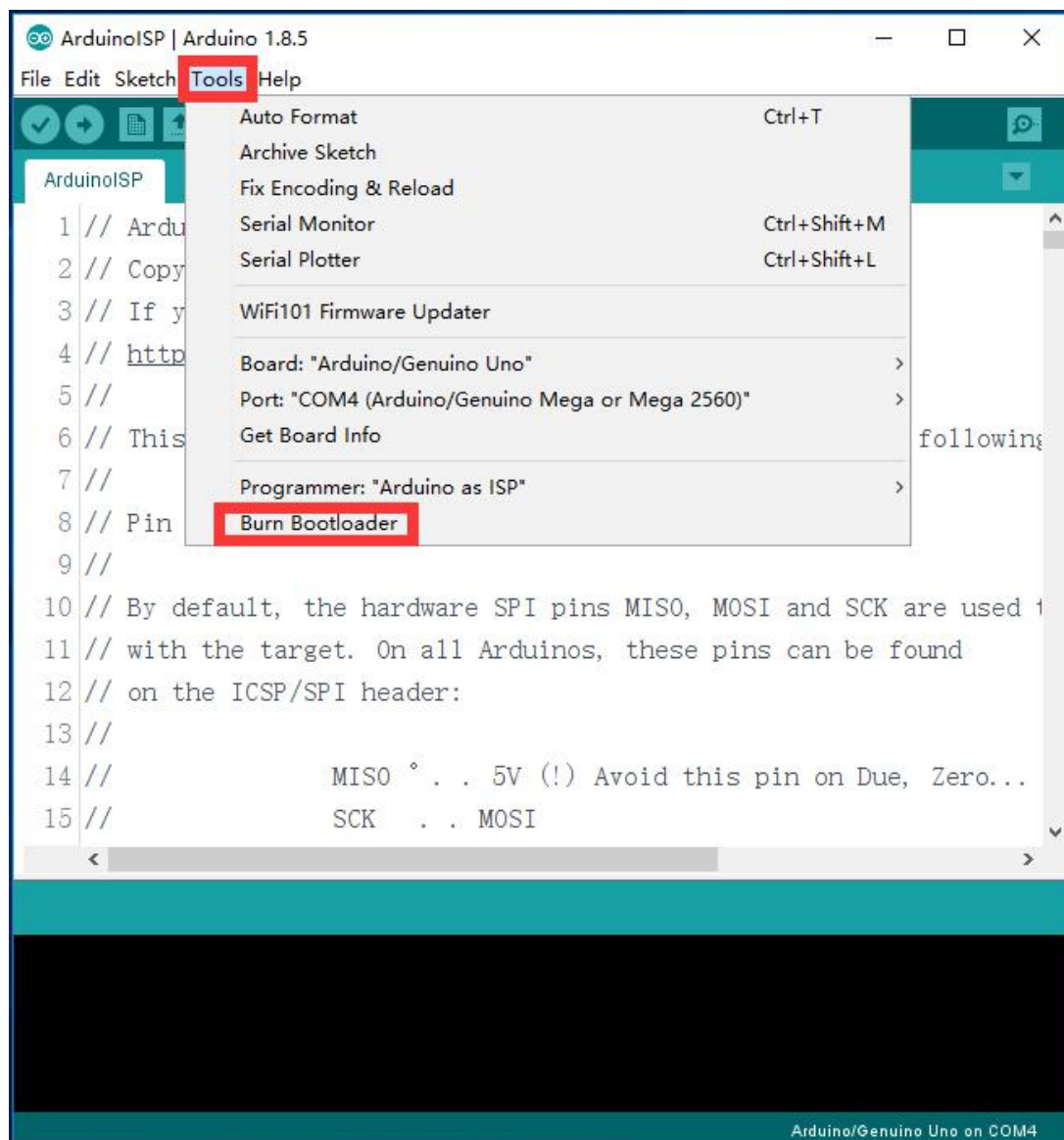
"D:\arduino 1.8.5\arduino\hardware\tools\avr\bin\avr-gcc" -w -Os -g -f

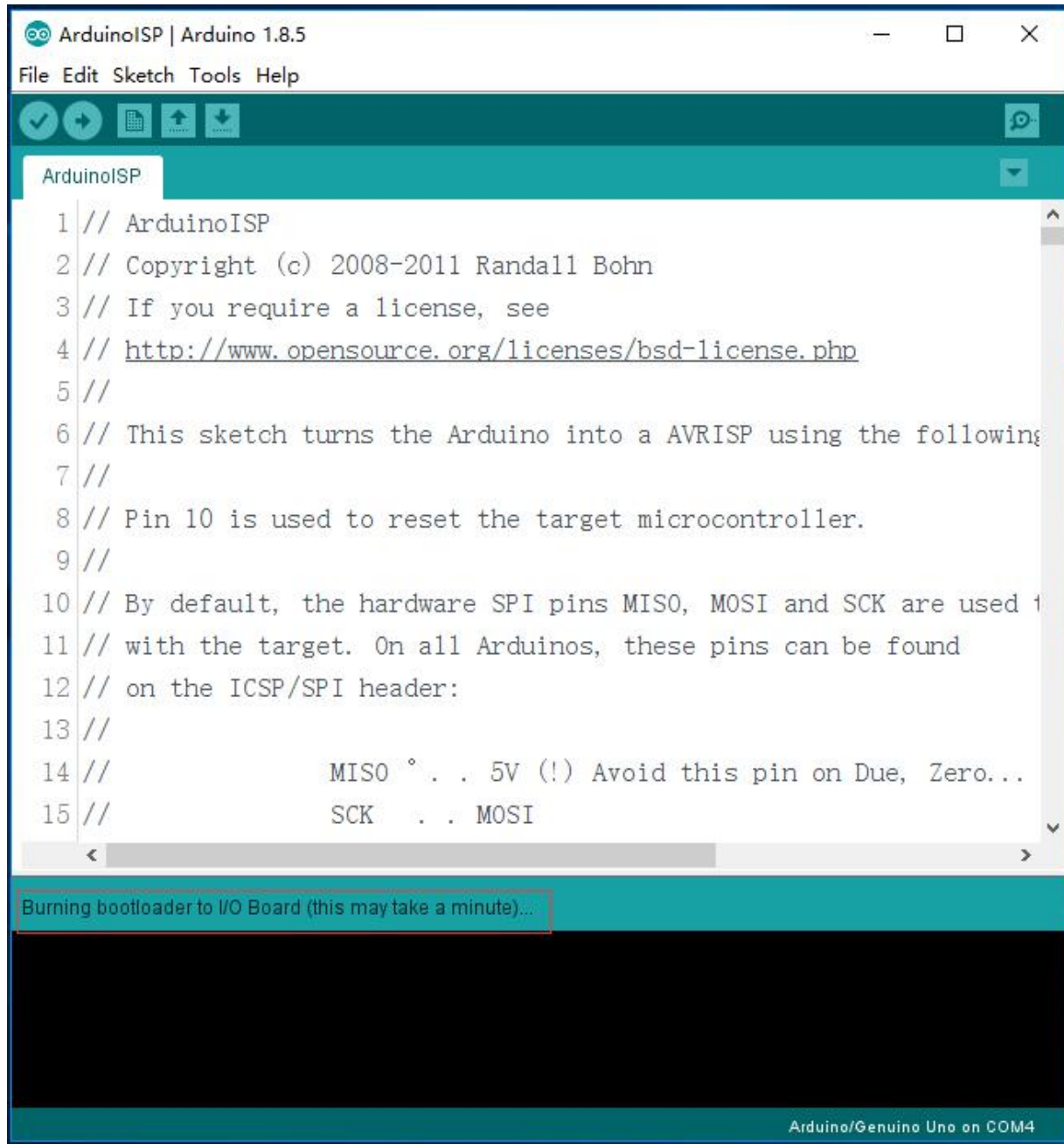
Arduino/Genuino Uno on COM4

Program the bootloader

If all the wires are set, you need to switch to the board type you want to program with the bootloader. This step is mandatory to select the proper bootloader code and the fuses configurations. The programming procedure checks the signature of the microcontroller before any writing action is taken, but many boards share the same microcontroller and each board has its own bootloader. The port remains the one of the ArduinoISP.

Choose “Burn bootloader” under tools and wait for the confirmation message in the lower part of the Arduino Software (IDE) interface.



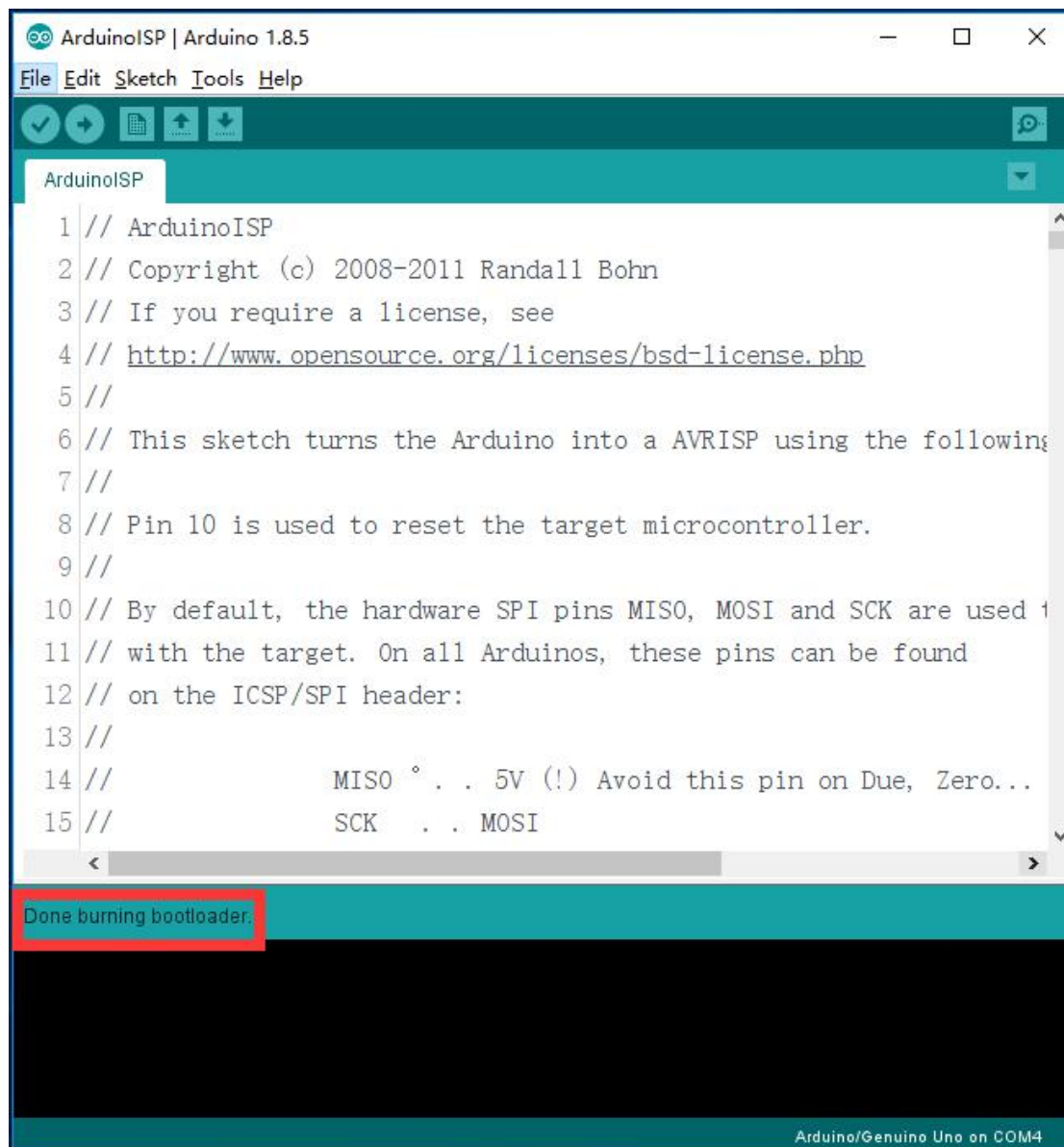


```
1 // ArduinoISP
2 // Copyright (c) 2008-2011 Randall Bohn
3 // If you require a license, see
4 // http://www.opensource.org/licenses/bsd-license.php
5 //
6 // This sketch turns the Arduino into a AVRISP using the following
7 //
8 // Pin 10 is used to reset the target microcontroller.
9 //
10 // By default, the hardware SPI pins MISO, MOSI and SCK are used to
11 // with the target. On all Arduinos, these pins can be found
12 // on the ICSP/SPI header:
13 //
14 //           MISO ° . . 5V (!) Avoid this pin on Due, Zero...
15 //           SCK   . . MOSI
```

Burning bootloader to I/O Board (this may take a minute)...

Arduino/Genuino Uno on COM4

Finally, arduino IDE shows "Done burning bootloader" as shown:



After the success, try the board again to see if it is functional.