Lab 1: Number Systems and Introduction to Arduino IDE

I. Lab Environment

- Need a cardkey to access the lab and the building. The cardkey can be purchased at the bookstore for a non-refundable $10 fee.
- Need a csc account to access the course website and computers in the lab. Activate your csc account here: https://connex.csc.uvic.ca/portal/site/itsupport/page/88ff914f-369a-4f5c-9f02-a47c610745ad. Email itsupport@csc.uvic.ca if you have any difficulties to activate your account.
- ECS 249 is the only lab with hardware/software provided for this course. Suggest you start your assignments early.
- **"H"** drive is your network space on the CSC server. Store all your work in H drive and keep a back up copy using other portable devices (such as a flash memory or a floppy disk). Files stored in the C drive might be erased overnight.
- Printing credit can be purchased on line at https://www.csc.uvic.ca/PrintPagePurchasing/

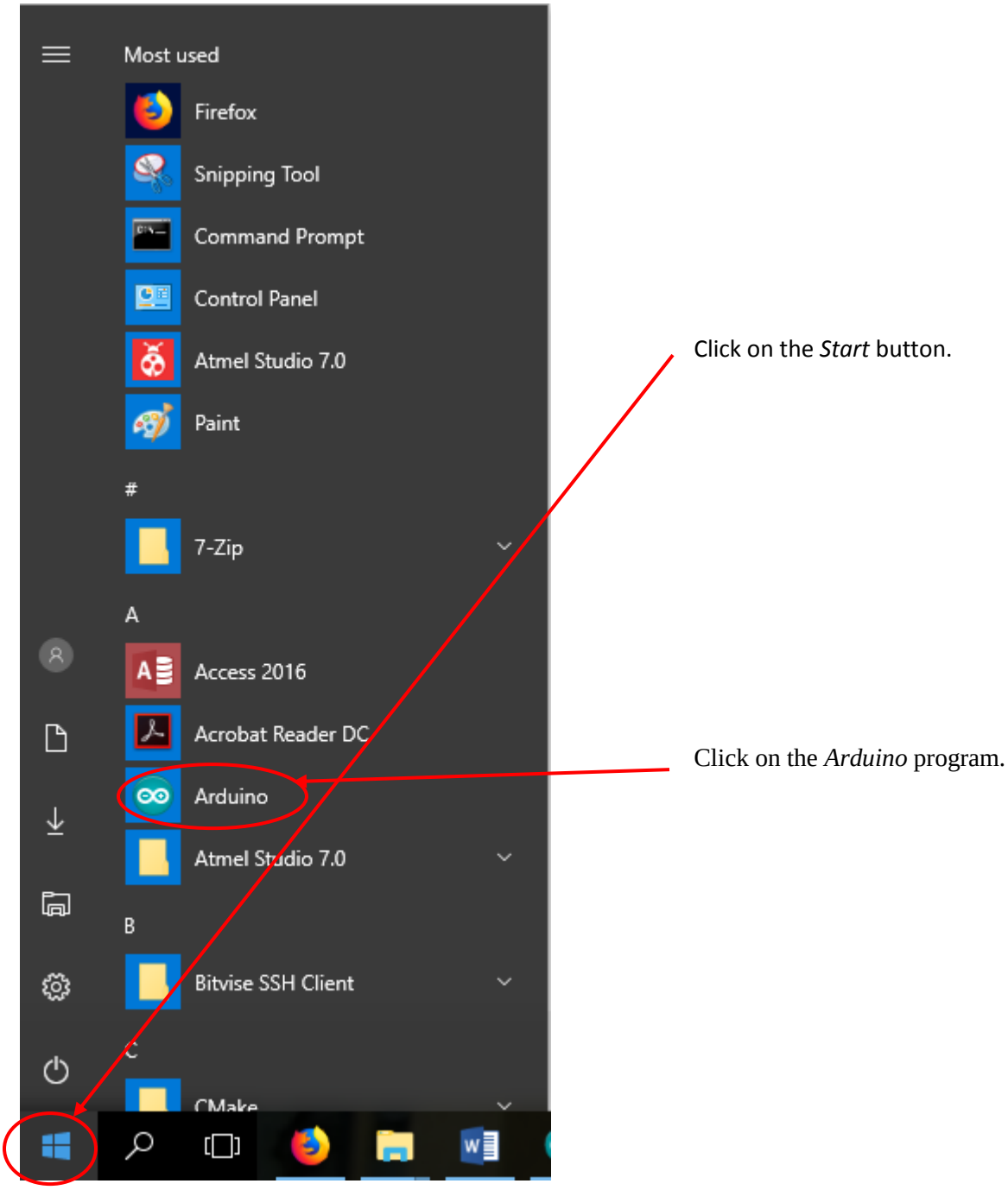
II. Number Systems

In the computer system, we need to know how many bits (storage space) are used. For example, let's count the number of students in the lab. In decimal number system, we just need two digits. In binary number system, how many bits do we need? Let's do counting up in decimal, binary and hexadecimal. Here is the conversion table:

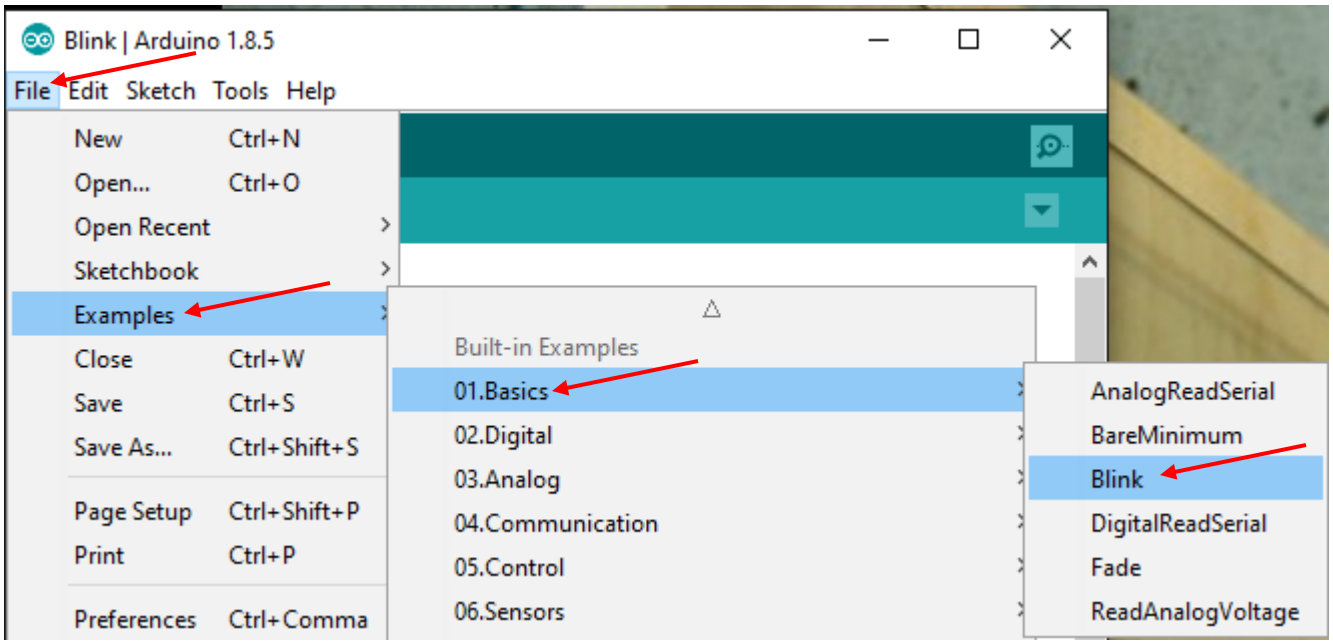
Decimal (2 digits)	Binary (5 bits)	Hexadecimal (2 digits)
0	0b00000	0x00
1	0b00001	0x01
2	0b00010	0x02
3	0b00011	0x03
4	0b00100	0x04
5	0b00101	0x05
6	0b00110	0x06
7	0b00111	0x07
8	0b01000	0x08
9	0b01001	0x09
10	0b01010	0x0A
11	0b01011	0x0B
12	0b01100	0x0C
13	0b01101	0x0D
14	0b01110	0x0E
15	0b01111	0x0F
16	0b10000	0x10
17	0b10001	0x11
18	0b10010	0x12
19	0b10011	0x13
20	0b10100	0x14
21	0b10101	0x15
22	0b10110	0x16
23	0b10111	0x17
24	0b11000	0x18

III. Test the Arduino Board - Introduction to Arduino IDE [1]

- Launch Arduino IDE: click on the *Start* button, then click on the *Arduino* program.



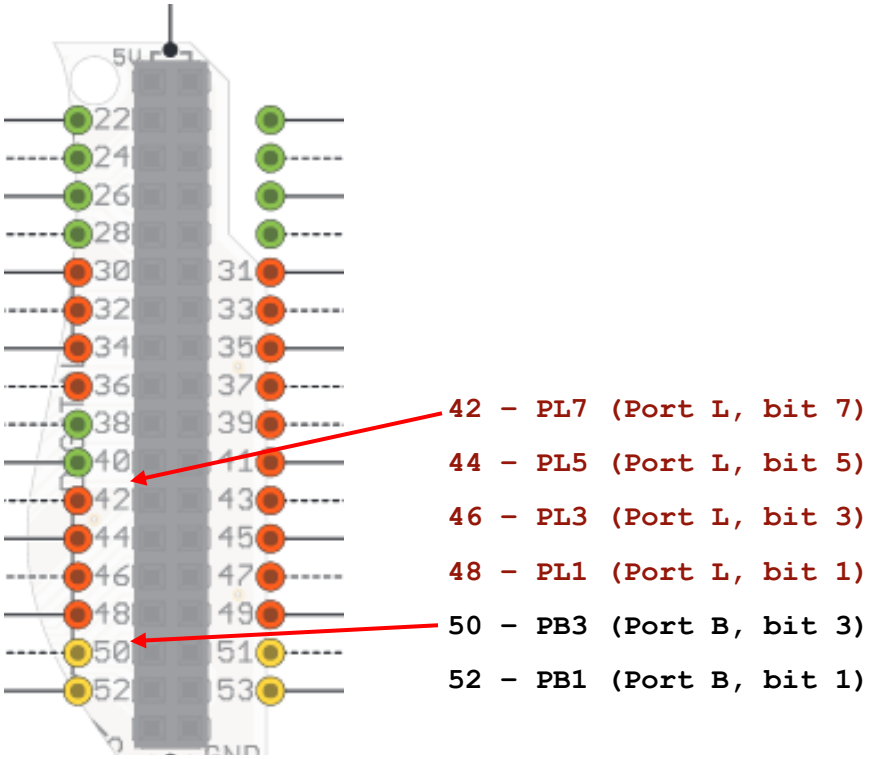
- The Arduino IDE is launched, go to the menu, click on *File -> Examples -> 01.Basics -> Blink.*



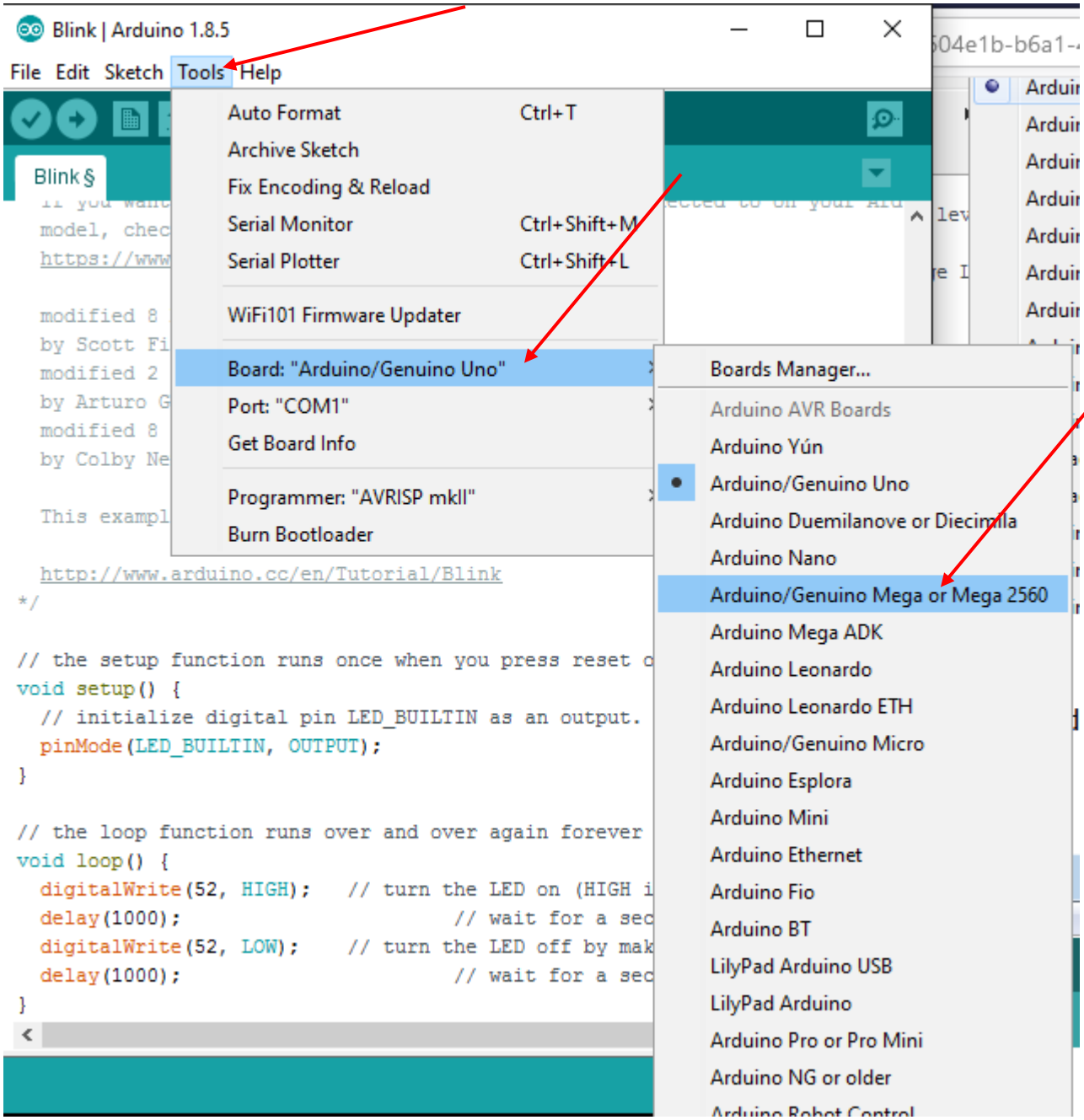
- Scroll down the source code and change *LED_BUILTIN* to 52:

```
  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.

  http://www.arduino.cc/en/Tutorial/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(52, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(52, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(52, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

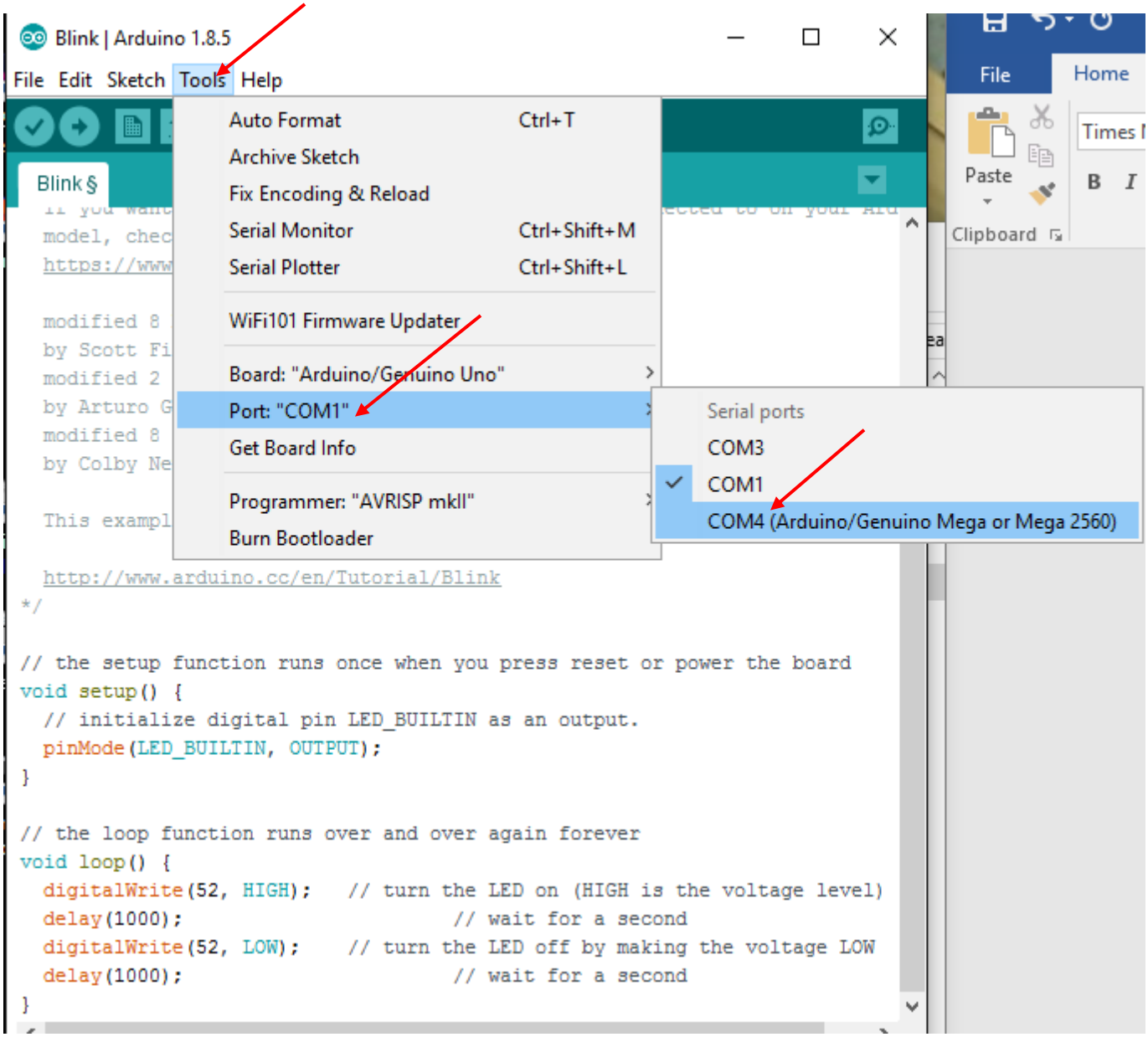
- Why change it to 52? It is the pin number on the Arduino board. You will learn more later in the course. To give you a brief explanation now, see the portion of a diagram of the Arduino board below. It shows the pins connected to the LED lights. Pin 52 controls the LED light at the bottom.



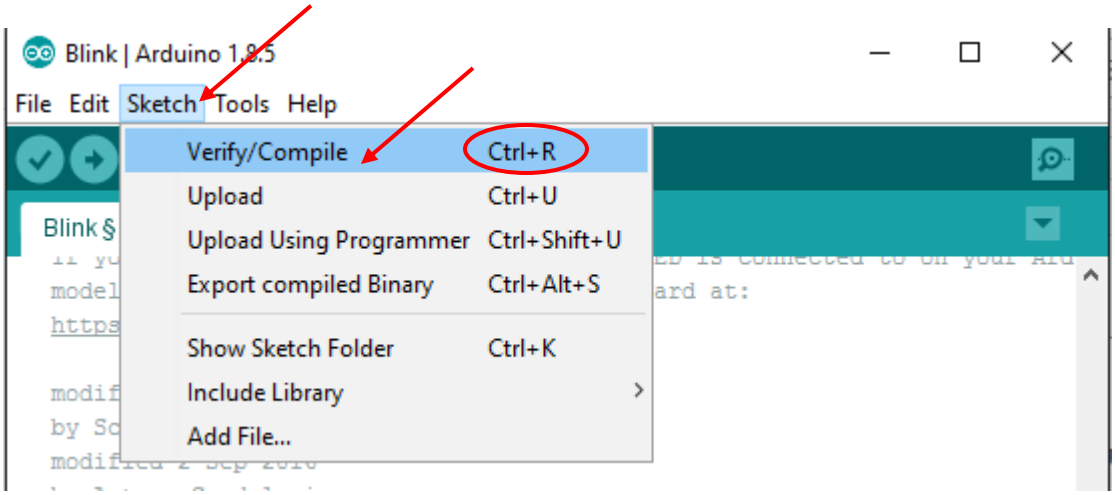
- In the Arduino IDE program, select the right board: go to the menu, click on *Tools →Board: "Arduino/Genuino Uno" → Arduino/Genuino Mega Or Mega 2560*.



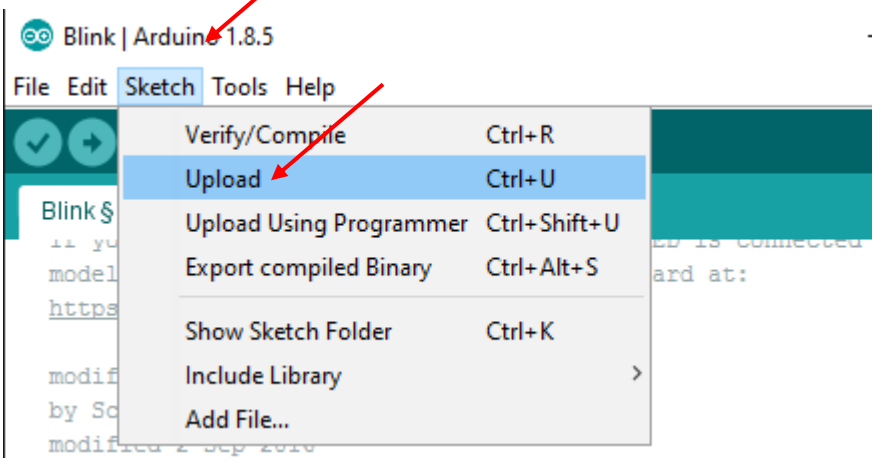
- Select the right port: go to the menu, click on *Tools →Port: "COM1" → COM4 (Arduino/Genuino Mega or Mega 2560).*



- Compile your code: go to the menu, click on *Sketch →Verify/Compile Ctrl+R.* You may use shortcut keys: press *Ctrl* key and *r* (lower case r) key at the same time.



- Upload your program to the board: go to the menu, click on *Sketch →Upload.* The LED light at the bottom blinks. Change 52 to other numbers (42, 44, 46, 48 or 50) and observe the LED lights.



IV. Exercises

1. Why do we use 5 bits in the table in part II? Can we use only 4-bits instead?
2. How many different values can be represented by 6 bits?
3. There are 190 students registered in CSc 230, what is the minimum number of bits needed to represent this number in binary (assume it is an unsigned number)?
4. Convert the following numbers:
    a. 0b1001101 (unsigned) to decimal
    b. 63 to binary and hexadecimal
5. Convert the following numbers:
    a. -63 to binary (using 2's complement notation) and then to Hex
    b. 0b101011 (2's complement) to decimal
    c. 0b001110 (2's complement) to decimal
6. What are the minimum and maximum values (in decimal) represented by a 4-bit binary number: a) as an unsigned number? b) as a signed number? How about 8 bits, 16 bits, k bits?

7. What is the result of bit-wise AND operation of: 0b10110010 with 0b11110000?
8. What is the result when bit-wise XOR and mask of 0b11111111 are applied on byte 0b10110100?
9. What is the mask to be used if we want to clear bits 2, 3, 5 and 7 of a byte?

Download lab1.txt. Write your answers and submit lab1.txt via conneX. You must click the "Submit" button. Write your name, student number at the top. Due at the end of your lab.

1. Adapted from the lab notes written by Dr. Bill Bird for csc 230 in the summer of 2018.