

# Computer Architecture

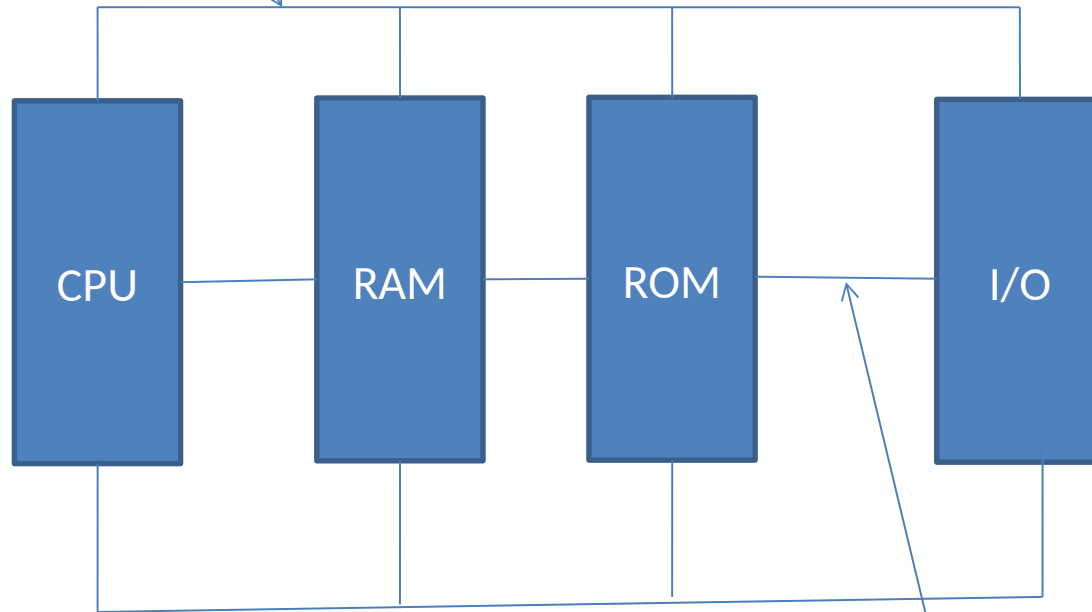
## The mainboard

# Lecture Objectives

Introduction to the Arduino Processors  
and the Integrated development platform

Preparation for the lab activities

Address BUS



Data BUS

Control BUS

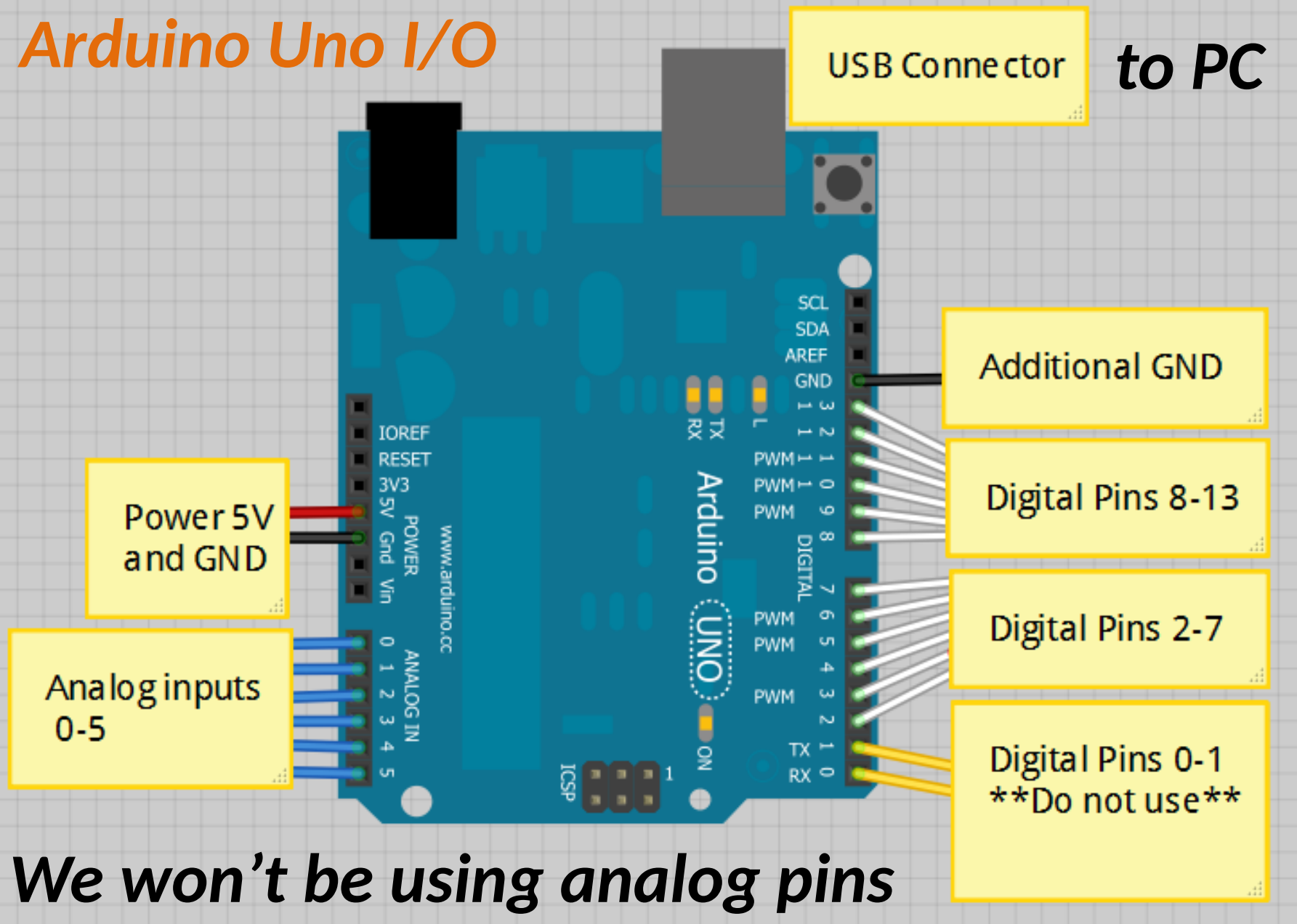
# Development Equipment

- Mainboard
  - Arduino Uno or Mega & PC USB cable
- Integrated Development Environment
  - Allows you to write and upload programs
- Breadboard
  - Allows components to be wired to the Arduino

# *The Mainboard: Arduino Uno*



# Arduino Uno I/O



*We won't be using analog pins*

# Atmel ATMega328 Processor

- CPU – 8-bit Atmel AVR (16MHz)
- FLASH – 32kBytes
- RAM – 2kBytes
- EEPROM – 1kBytes
- Input Output
  - Analog/Digital
  - PWM
  - Comms - Serial, I2C, SPI

# Arduino Mega

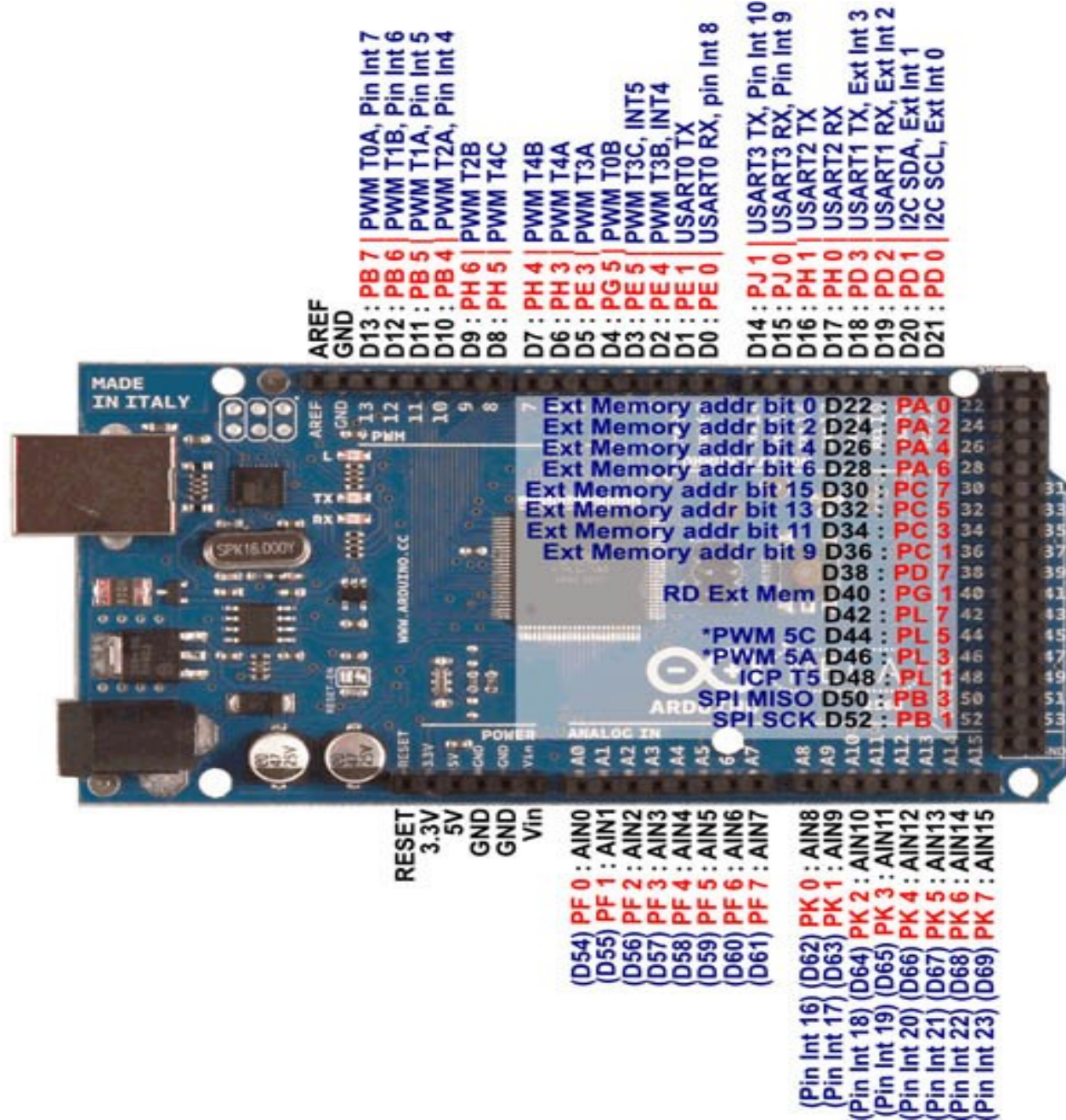


***The Mainboard:  
Arduino Mega***



# Microcontroller ATmega2560

Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	1 KB



RESET

3.3V

5V

GND

GND

Vin

(D54) PF 0 : AIN0  
(D55) PF 1 : AIN1  
(D56) PF 2 : AIN2  
(D57) PF 3 : AIN3  
(D58) PF 4 : AIN4  
(D59) PF 5 : AIN5  
(D60) PF 6 : AIN6  
(D61) PF 7 : AIN7

(Pin Int 16) (D62) PK 0 : AIN8  
(Pin Int 17) (D63) PK 1 : AIN9  
(Pin Int 18) (D64) PK 2 : AIN10  
(Pin Int 19) (D65) PK 3 : AIN11  
(Pin Int 20) (D66) PK 4 : AIN12  
(Pin Int 21) (D67) PK 5 : AIN13  
(Pin Int 22) (D68) PK 6 : AIN14  
(Pin Int 23) (D69) PK 7 : AIN15

AREF

GND

D13 : PB 7 | PWM T0A, Pin Int 7

D12 : PB 6 | PWM T1B, Pin Int 6

D11 : PB 5 | PWM T1A, Pin Int 5

D10 : PB 4 | PWM T2A, Pin Int 4

D9 : PH 6 | PWM T2B

D8 : PH 5 | PWM T4C

D7 : PH 4 | PWM T4B

D6 : PH 3 | PWM T4A

D5 : PE 3 | PWM T3A

D4 : PG 5 | PWM T0B

D3 : PE 5 | PWM T3C, INT5

D2 : PE 4 | PWM T3B, INT4

D1 : PE 1 | USART0 TX

D0 : PE 0 | USART0 RX, pin Int 8

D14 : PJ 1 | USART3 TX, Pin Int 10

D15 : PJ 0 | USART3 RX, Pin Int 9

D16 : PH 1 | USART2 TX

D17 : PH 0 | USART2 RX

D18 : PD 3 | USART1 TX, Ext Int 3

D19 : PD 2 | USART1 RX, Ext Int 2

D20 : PD 1 | I2C SDA, Ext Int 1

D21 : PD 0 | I2C SCL, Ext Int 0

Ext Memory addr bit 0 D22 : PA 0  
Ext Memory addr bit 2 D24 : PA 2  
Ext Memory addr bit 4 D26 : PA 4  
Ext Memory addr bit 6 D28 : PA 6  
Ext Memory addr bit 15 D30 : PC 7  
Ext Memory addr bit 13 D32 : PC 5  
Ext Memory addr bit 11 D34 : PC 3  
Ext Memory addr bit 9 D36 : PC 1

RD Ext Mem

\*PWM 5C

\*PWM 5A

ICP T5

SPI MISO

SPI SCK

D40 : PG 1

D42 : PL 7

D44 : PL 5

D46 : PL 3

D48 : PL 1

D50 : PB 3

D52 : PB 1

5V

D23 : PA 1 | Ext Memory addr bit 1  
D25 : PA 3 | Ext Memory addr bit 3  
D27 : PA 5 | Ext Memory addr bit 5  
D29 : PA 7 | Ext Memory addr bit 7  
D31 : PC 6 | Ext Memory addr bit 14  
D33 : PC 4 | Ext Memory addr bit 12  
D35 : PC 2 | Ext Memory addr bit 10  
D37 : PC 0 | Ext Memory addr bit 8  
D39 : PG 2 | ALE Ext Mem  
D41 : PG 0 | Wr Ext Mem  
D43 : PL 6  
D45 : PL 4  
D47 : PL 2  
D49 : PL 0  
D51 : PB 2  
D53 : PB 0  
\*PWM 5B  
T5 external counter  
ICP T4  
SPI MOSI  
SPI SS

# Integrated Development Platform

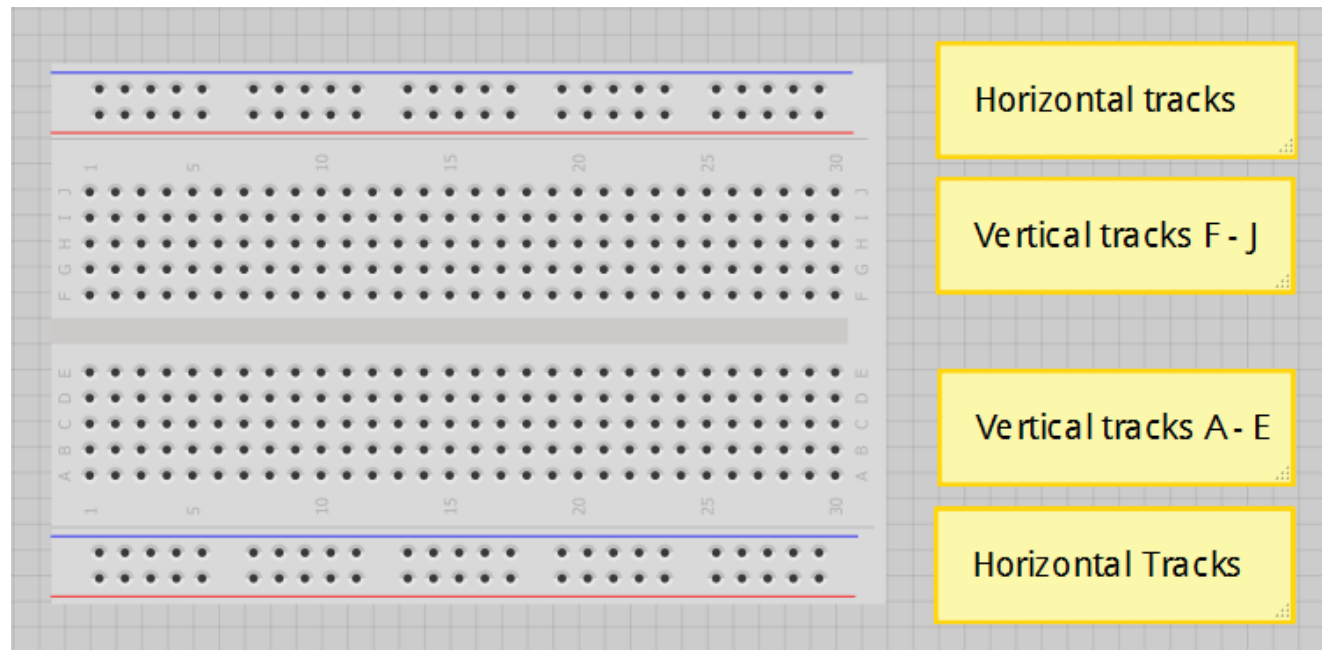
- Arduino has no operating system.
- Initialises it's hardware via it's BIOS.
- No direct user interaction.
- Requires a development platform.
- The development platform exists on a PC.

# Integrated Development Environment

- We need a PC based IDE to interact with Arduino
- This allows us to:
  - Write the software
  - Compile it into a binary image (.EXE)
  - Upload this to the Arduino
- This is programmed (Down loaded) into the FLASH memory ready for execution
  - Once done, it can operate independently

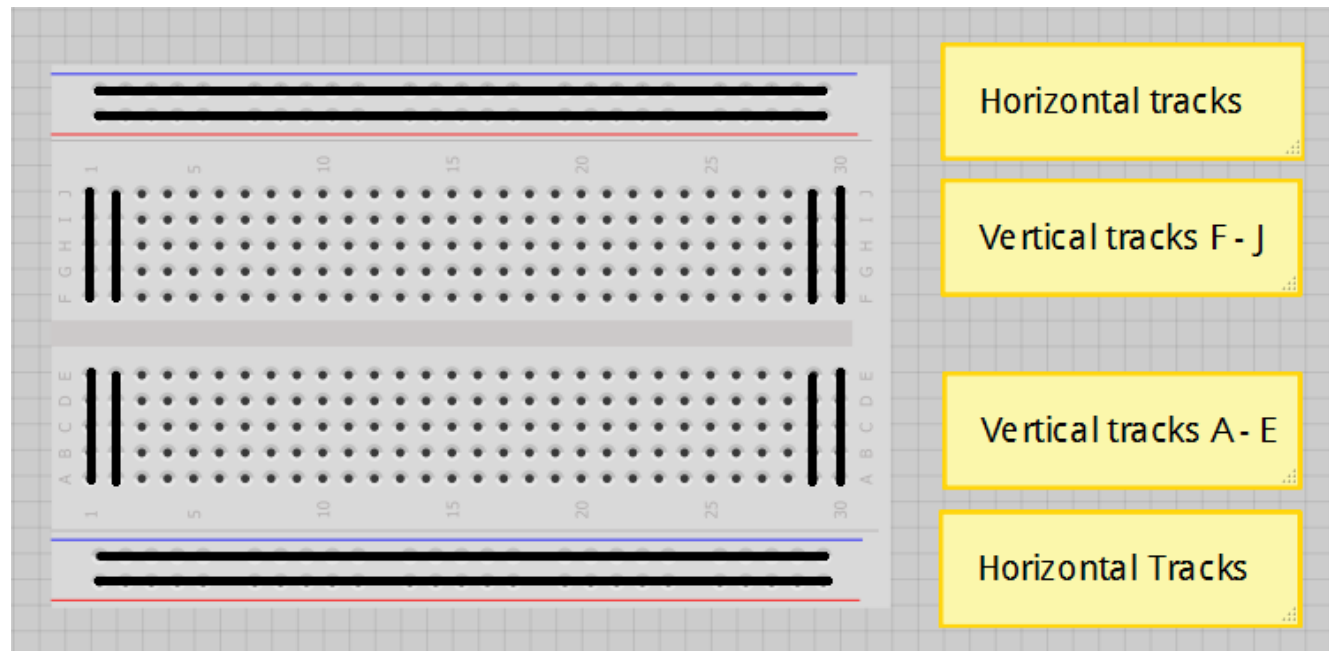
# Breadboards

- To wire components together easily
- Consist of pin sockets hardwired in rows

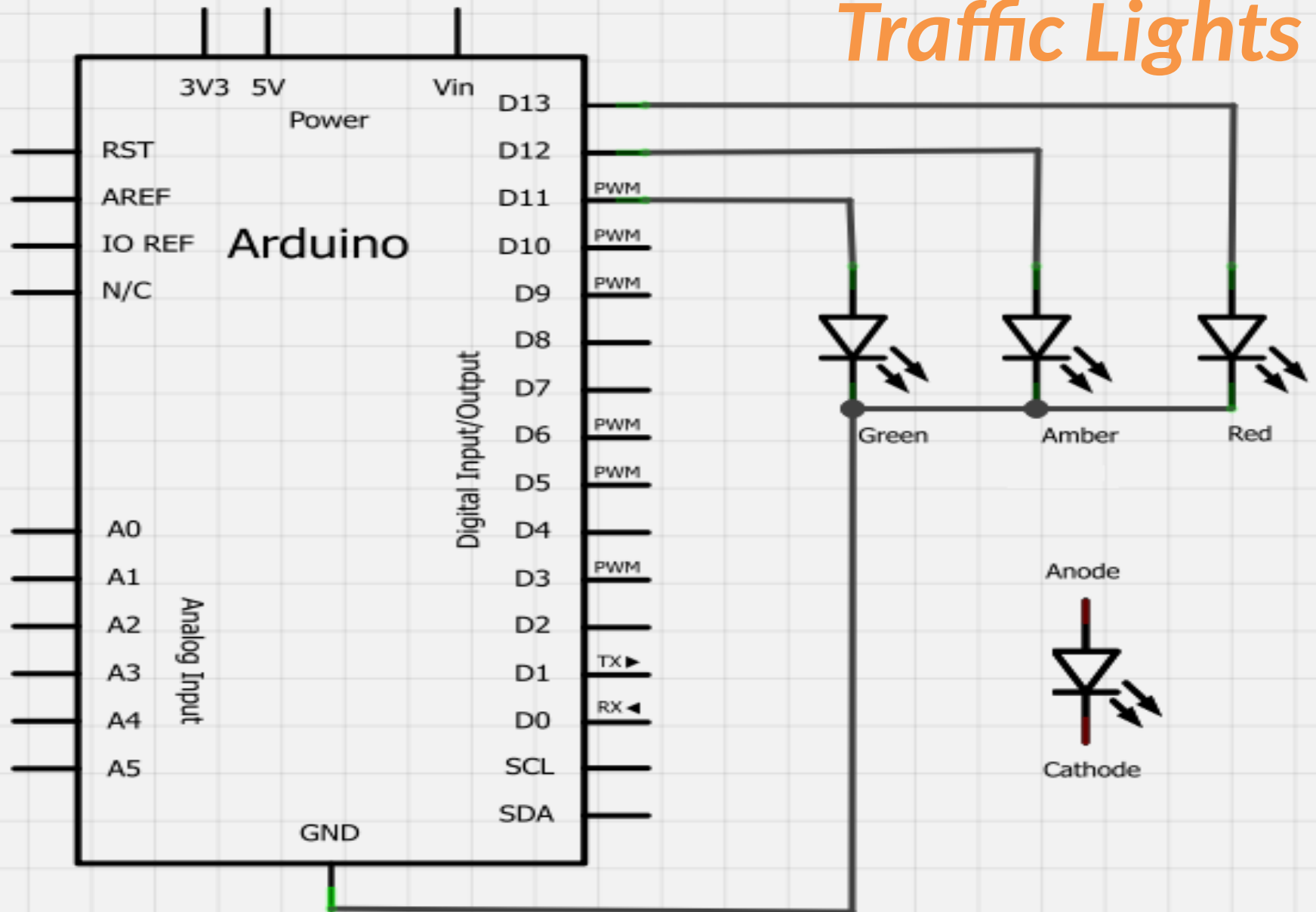


# Breadboards

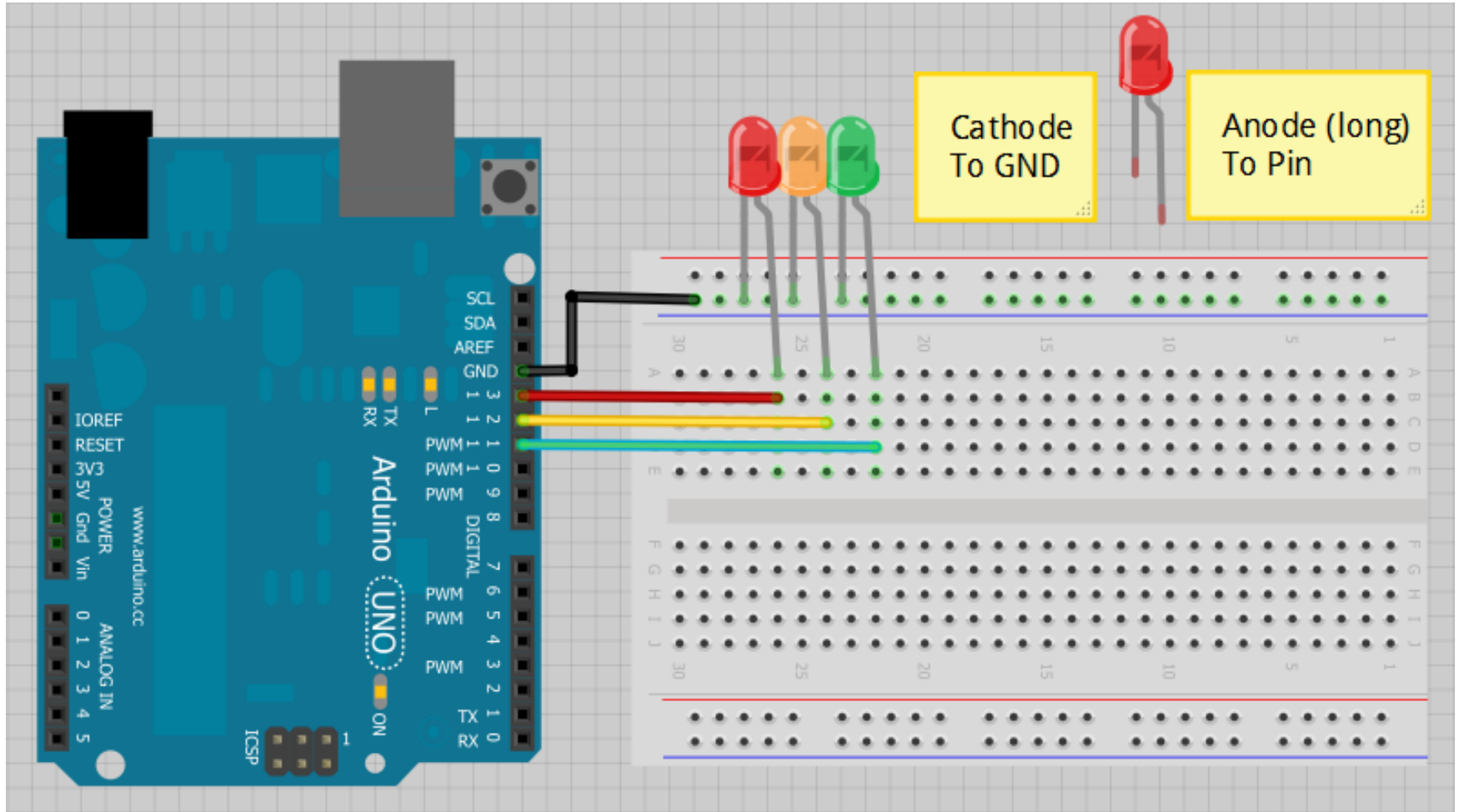
- To wire components together easily
- Consist of pin sockets hardwired in rows



# Traffic Lights



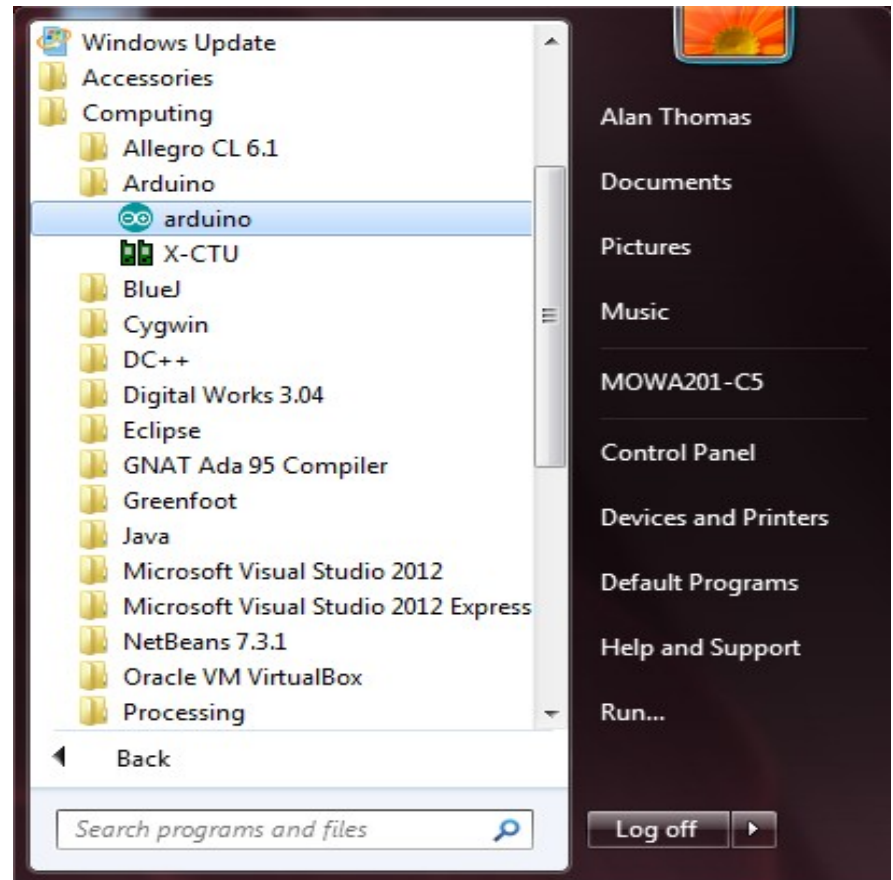
# Breadboards – Traffic Lights





# Arduino IDE

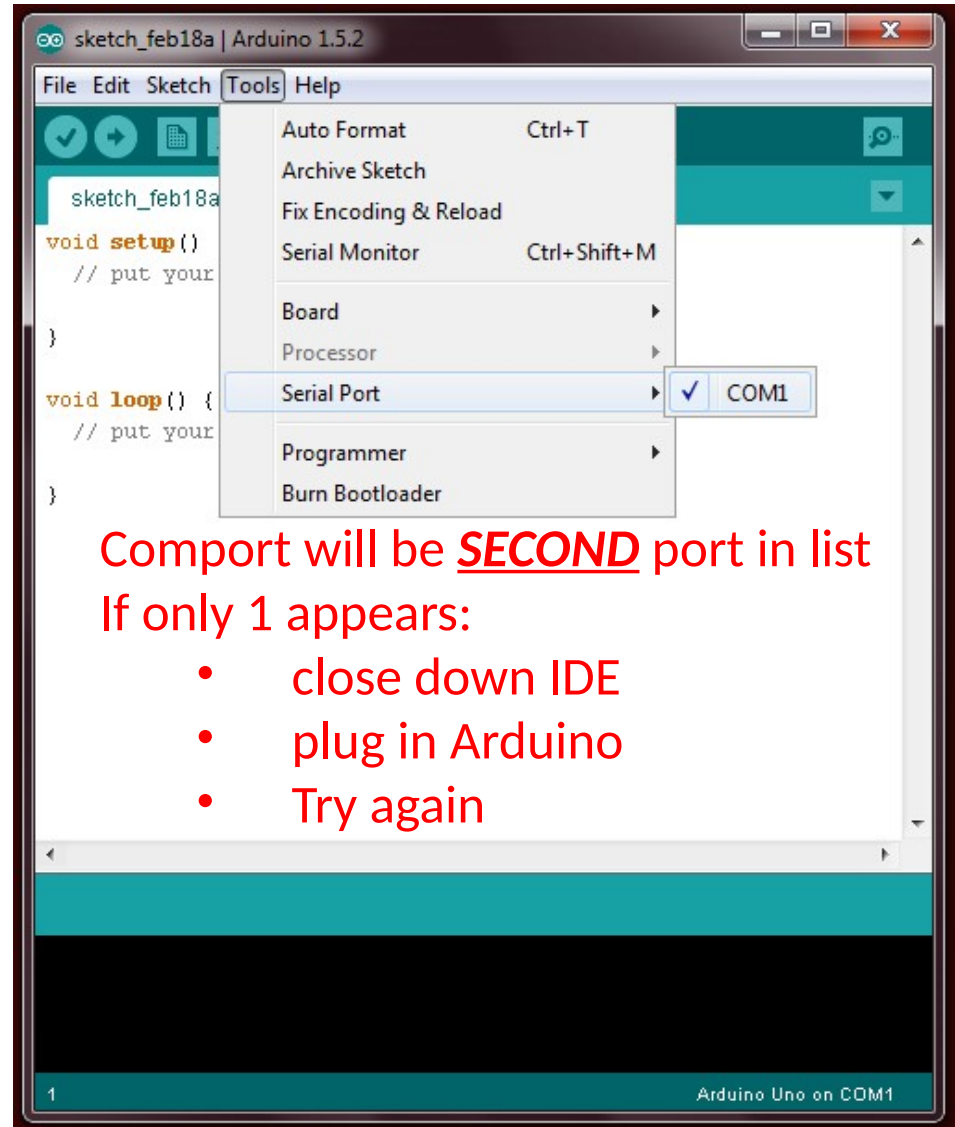
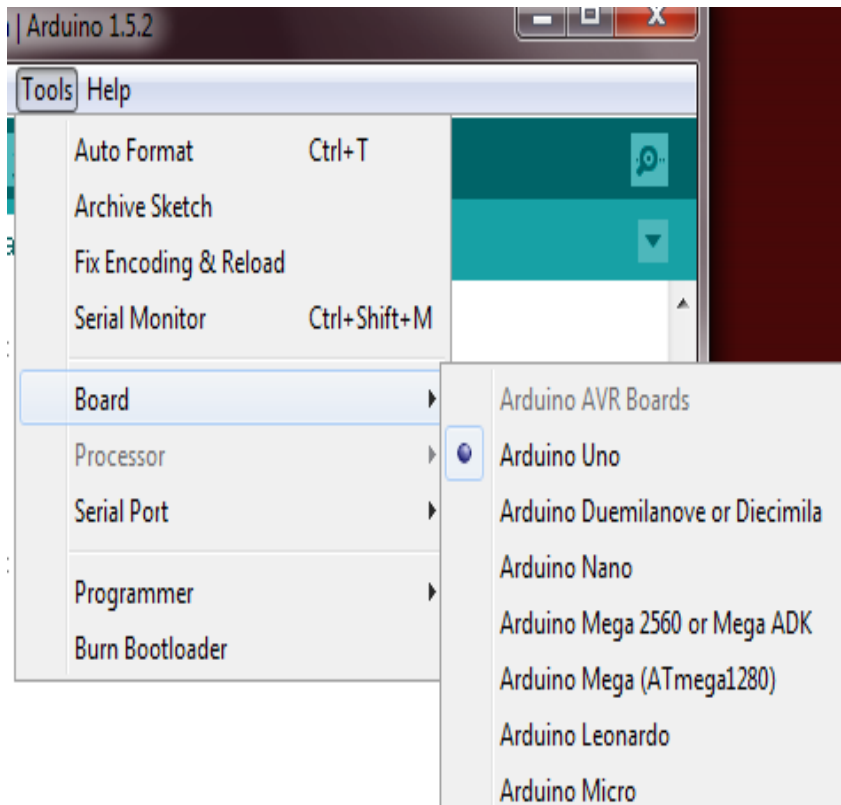
- Connect to PC using USB cable
- Launch the IDE:
  - Path:
    - All programs
    - Computing
    - Arduino
    - arduino



# 1. Configure IDE

1. Select Board

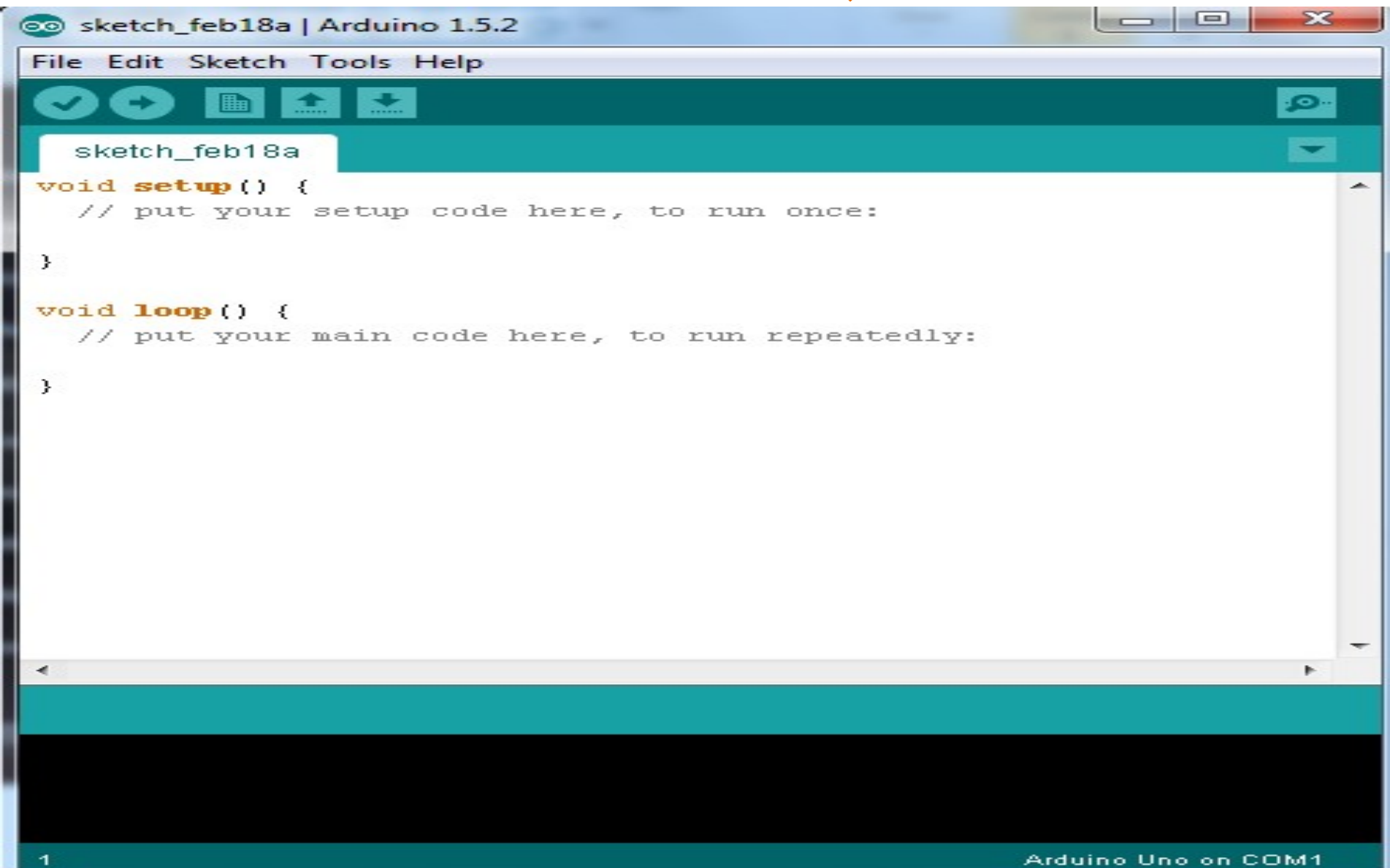
2. Select Serial Port



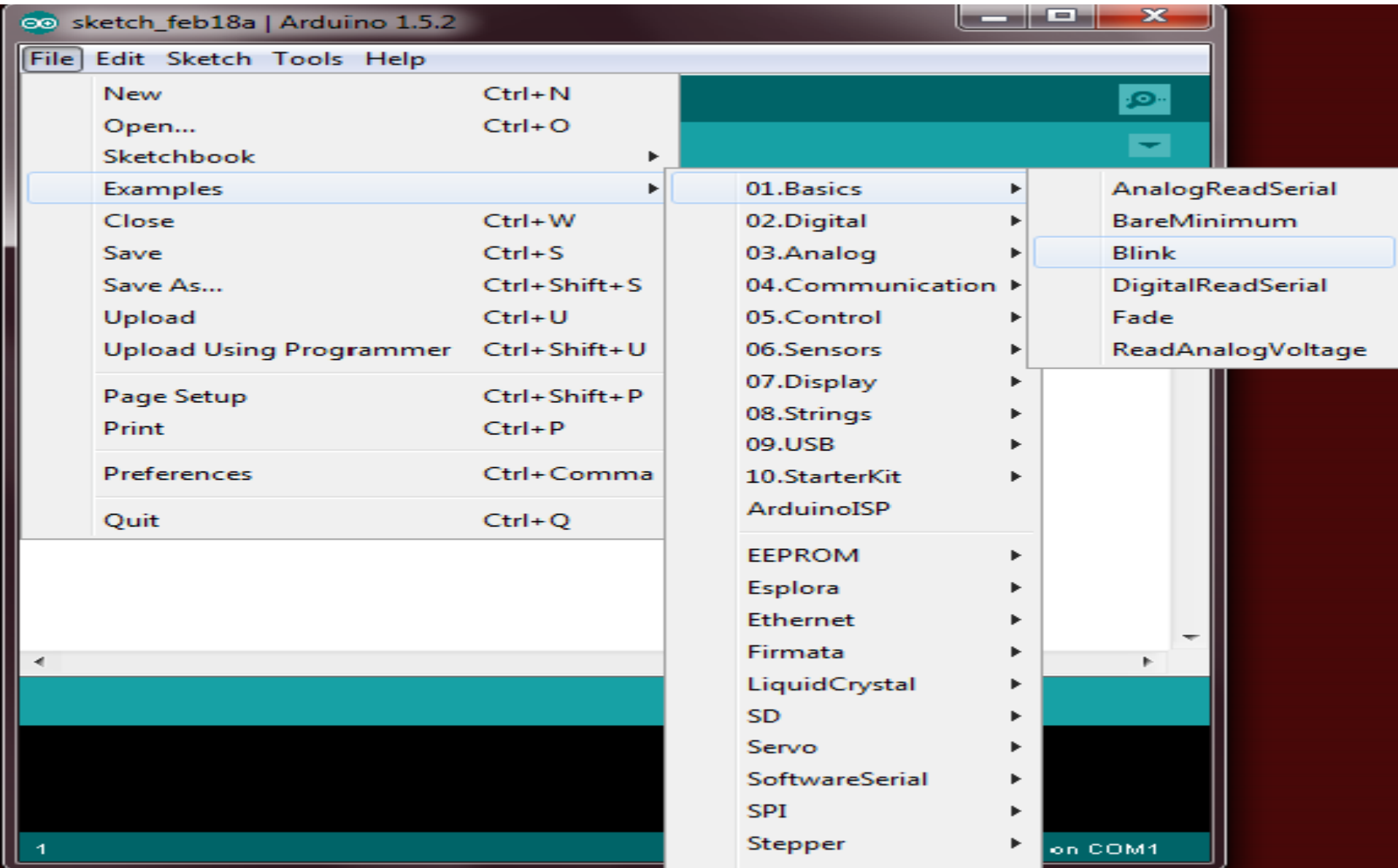
Comport will be **SECOND** port in list  
If only 1 appears:

- close down IDE
- plug in Arduino
- Try again

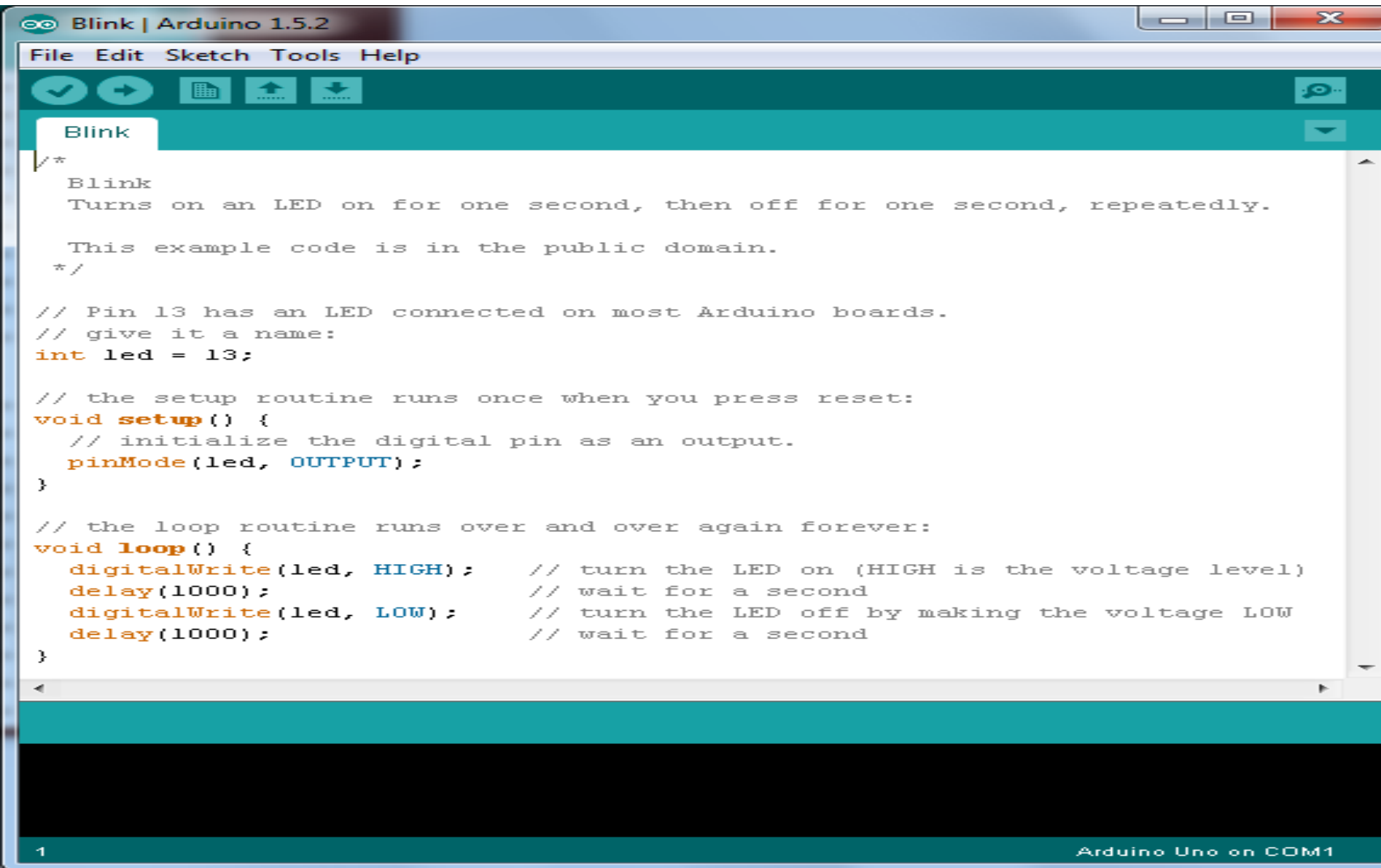
## 2. Write Code, or....



# 3. Load Example



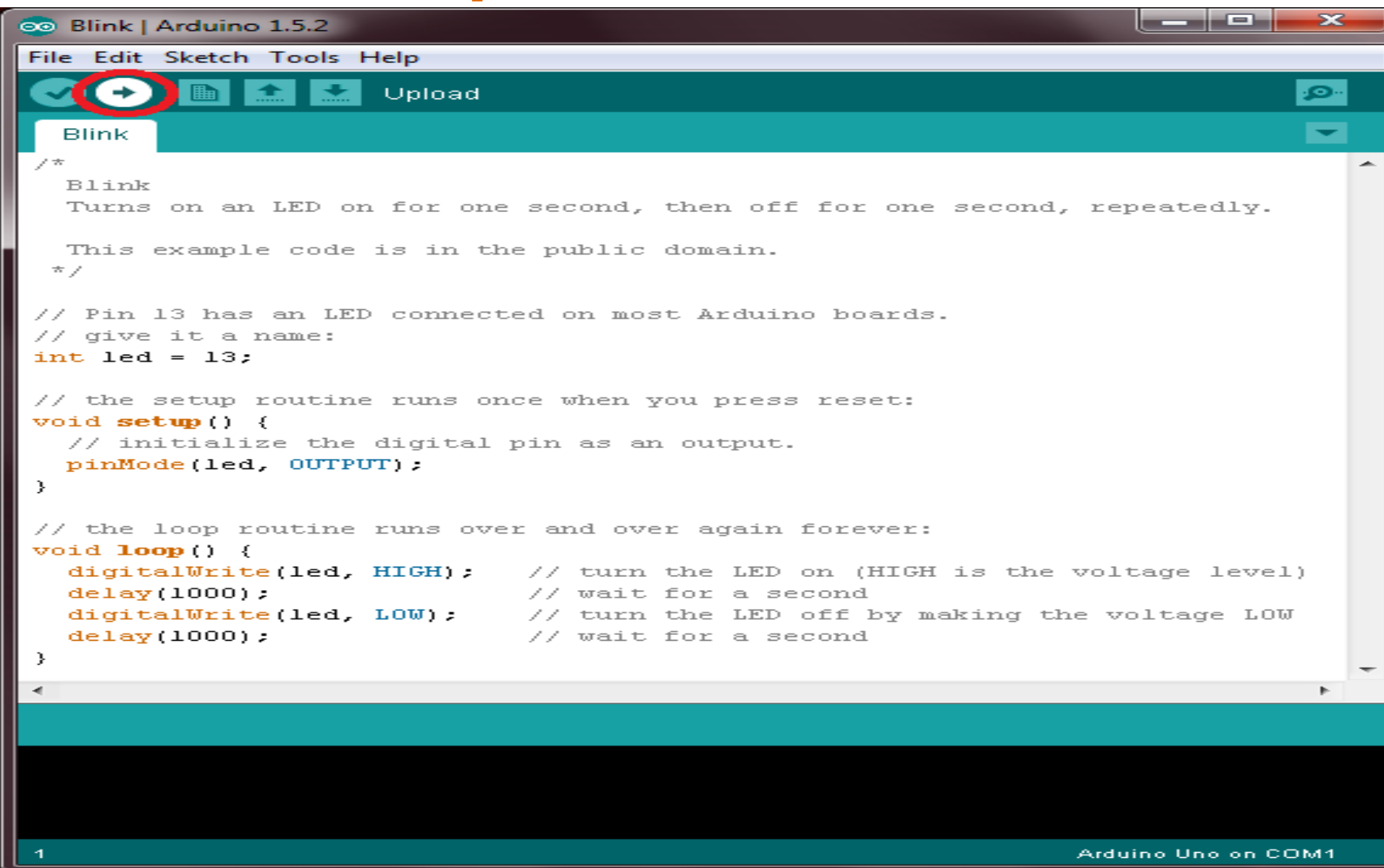
# 3a. Blink Example

The image is a screenshot of the Arduino IDE interface. The title bar at the top reads "Blink | Arduino 1.5.2". Below the title bar is a menu bar with "File", "Edit", "Sketch", "Tools", and "Help". Underneath the menu bar is a toolbar with icons for checking, running, serial monitor, and uploading/downloading. A tab labeled "Blink" is active. The main text area contains the following code:

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)  
  delay(1000);               // wait for a second  
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW  
  delay(1000);               // wait for a second  
}
```

At the bottom of the window, there is a status bar. On the left, it shows the line number "1". On the right, it says "Arduino Uno on COM1".

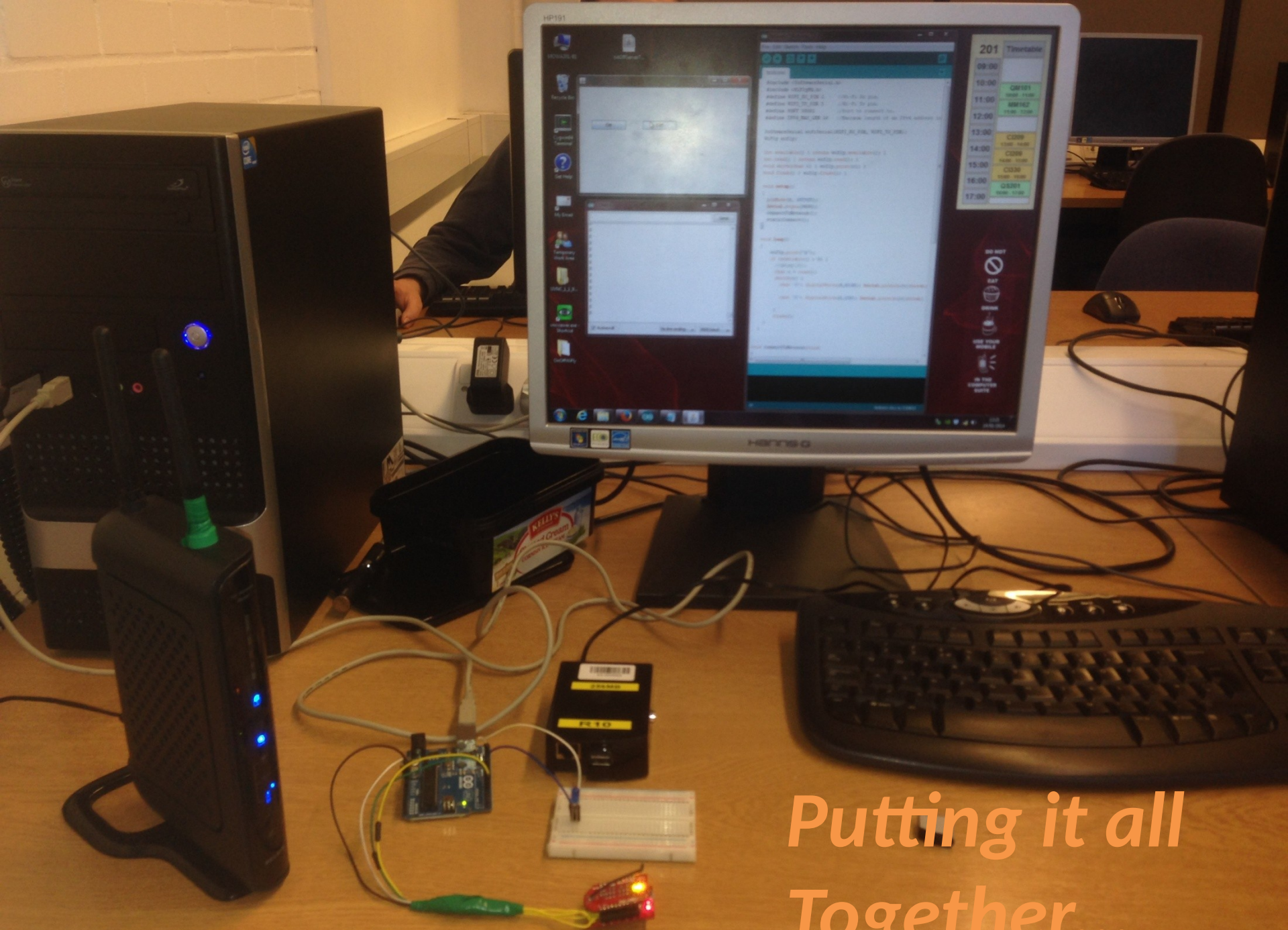
# 4. Upload to Arduino



# Lab activity 1 - Familiarisation

1. Obtain an Arduino dev. kit from the helpdesk
  - you will need your student card to sign it out
2. Upload the blink example
  - This flashes the on-board LED
3. Connect a single LED to pin 13 and GND
4. Using the PhysicalPixel example, modify your code so that you can turn the LED on and off from the computer keyboard.





Putting it all  
Together...



HP191

MOWA201-81

onOffServerT...

Recycle Bin

Cygwin64  
Terminal

Get Help

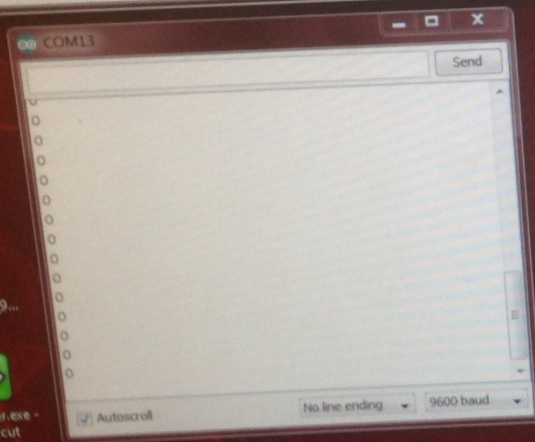
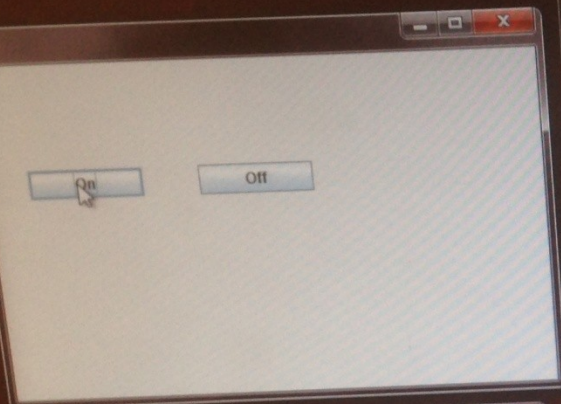
My Email

Temporary  
Work Area

UVNC 1.1.1...

vncviewer.exe -  
Shortcut

OnOffWifly



```
testconn [Arduino 1.5.2]
File Edit Sketch Tools Help

testconn

#include <SoftwareSerial.h>
#include <WiFlyHQ.h>
#define WIFI_RX_PIN 2 //Wi-Fi Rx pin.
#define WIFI_TX_PIN 3 //Wi-Fi Tx pin.
#define PORT 50001 //Port to connect to.
#define IPV4_MAX_LEN 16 //Maximum length of an IPv4 address in

SoftwareSerial wifiSerial(WIFI_RX_PIN, WIFI_TX_PIN);
WiFly wifly;

int available() { return wifly.available(); }
int read() { return wifly.read(); }
void write(char c) { wifly.print(c); }
void flush() { wifly.flush(); }

void setup()
{
  pinMode(8, OUTPUT);
  Serial.begin(9600);
  connectToNetwork();
  staticConnect();
}

void loop()
{
  wifly.print("A");
  if (available() > 0) {
    //delay(15);
    char c = read();
    switch(c) {
      case '0': digitalWrite(8,HIGH); Serial.println(c);break;
      case 'S': digitalWrite(8,LOW); Serial.println(c);break;
    }
    flush();
  }
}

void connectToNetwork(void)
{
  //
}
```

## 201 Timetable

09:00

10:00

11:00

12:00

13:00

14:00

15:00

16:00

17:00

QM101

10:00 - 11:00

MM162

11:00 - 12:00

CI209

13:00 - 14:00

CI209

14:00 - 15:00

CI330

15:00 - 16:00

QS201

16:00 - 17:00

DO NOT



EAT



DRINK



USE YOUR  
MOBILE



IN THE  
COMPUTER  
SUITE

12:20  
14/01/2014

Hanns.G



