Introduction to Arduino

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

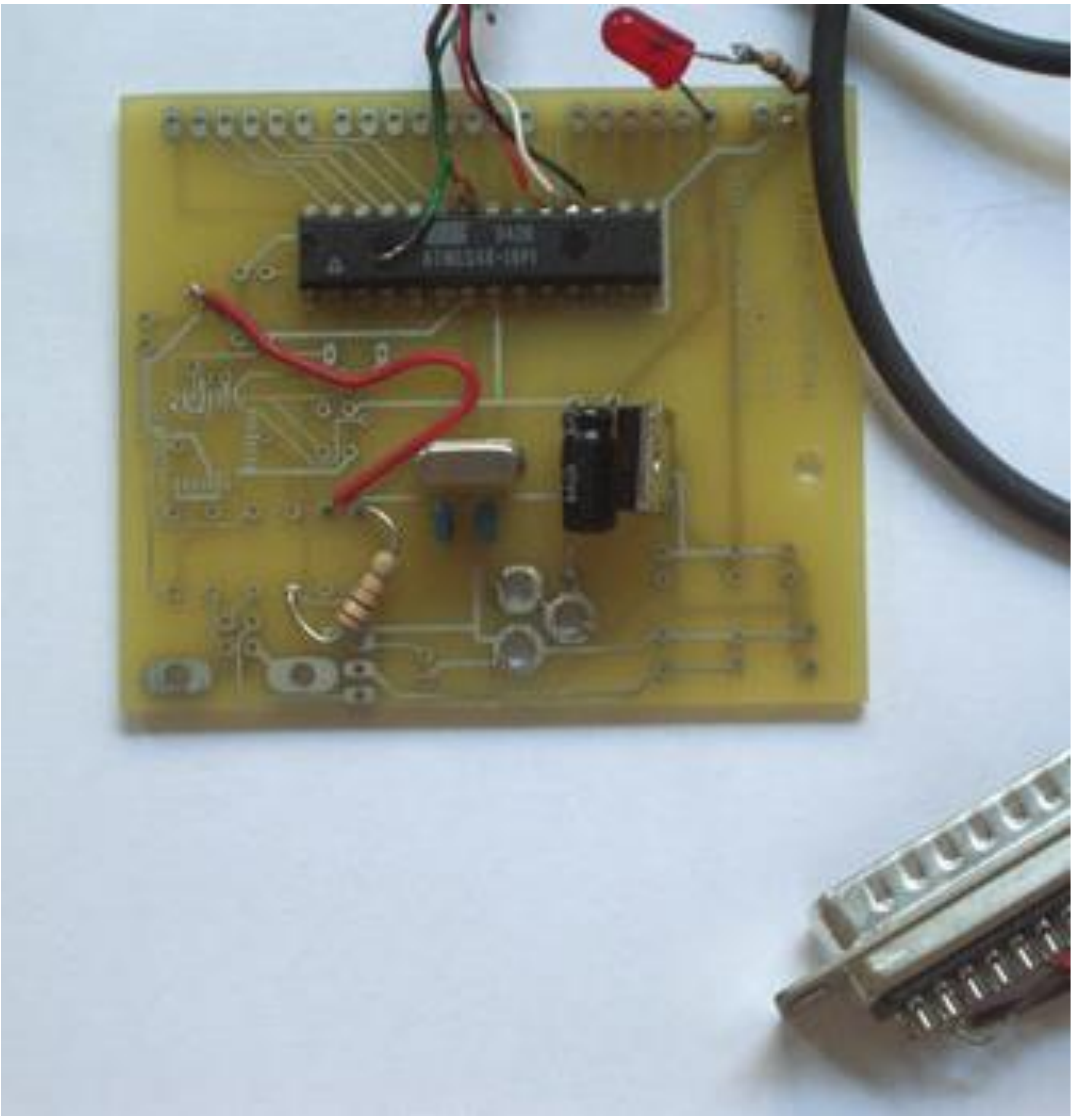
HYDERABAD

IIIT Hyderabad

Arduino...

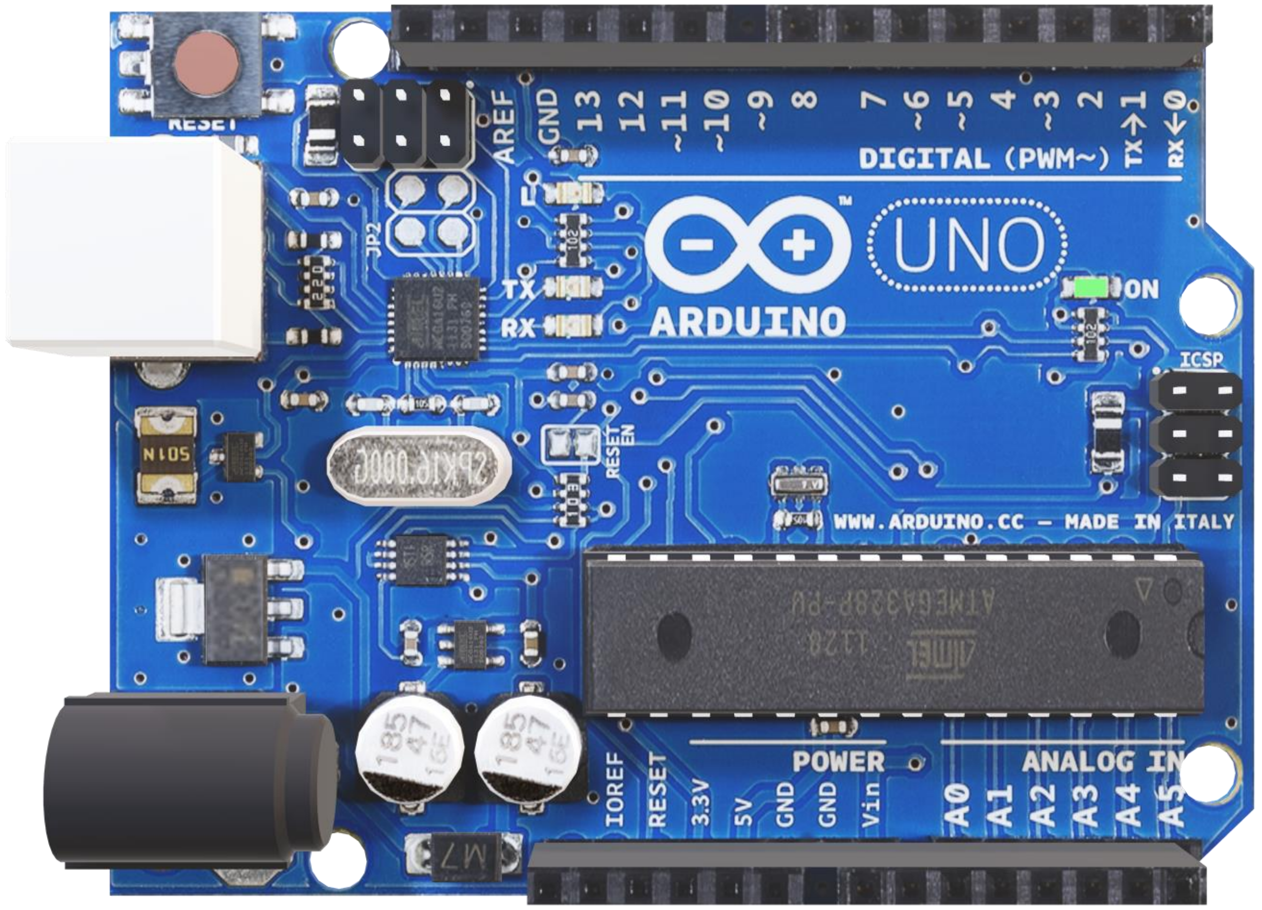
- is the goa-to gear for artists, hobbyists, students, and anyone with a gadgetry dream.

- rose out of another formidable challenge: how to teach students to create electronics, fast.



http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino

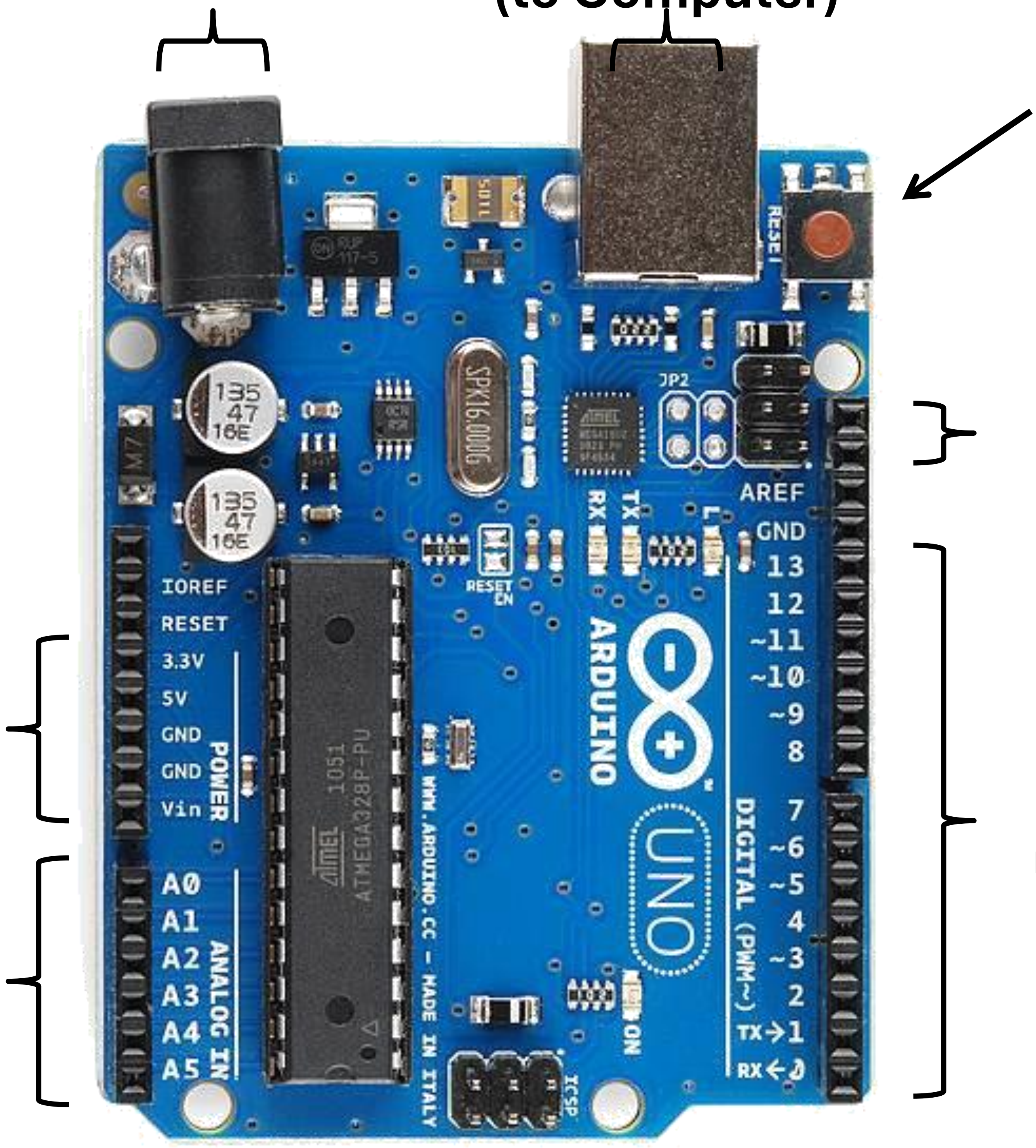
01 Basics of Arduino



- Arduino is a small controller that you can use to create all sorts of cool and fun projects.

- An Arduino is like your little smart friend. You write a set of instructions in a special language that it understands (code), and then you upload that code to the Arduino. Once it has the instructions, it starts doing whatever you told it to do.

- Let's say, we have to sense some movement. A motion sensor can be attached to it and it will send data to Arduino. Now, this data can be used to control a motor controlled door.



PWR IN

USB
(to Computer)

RESET

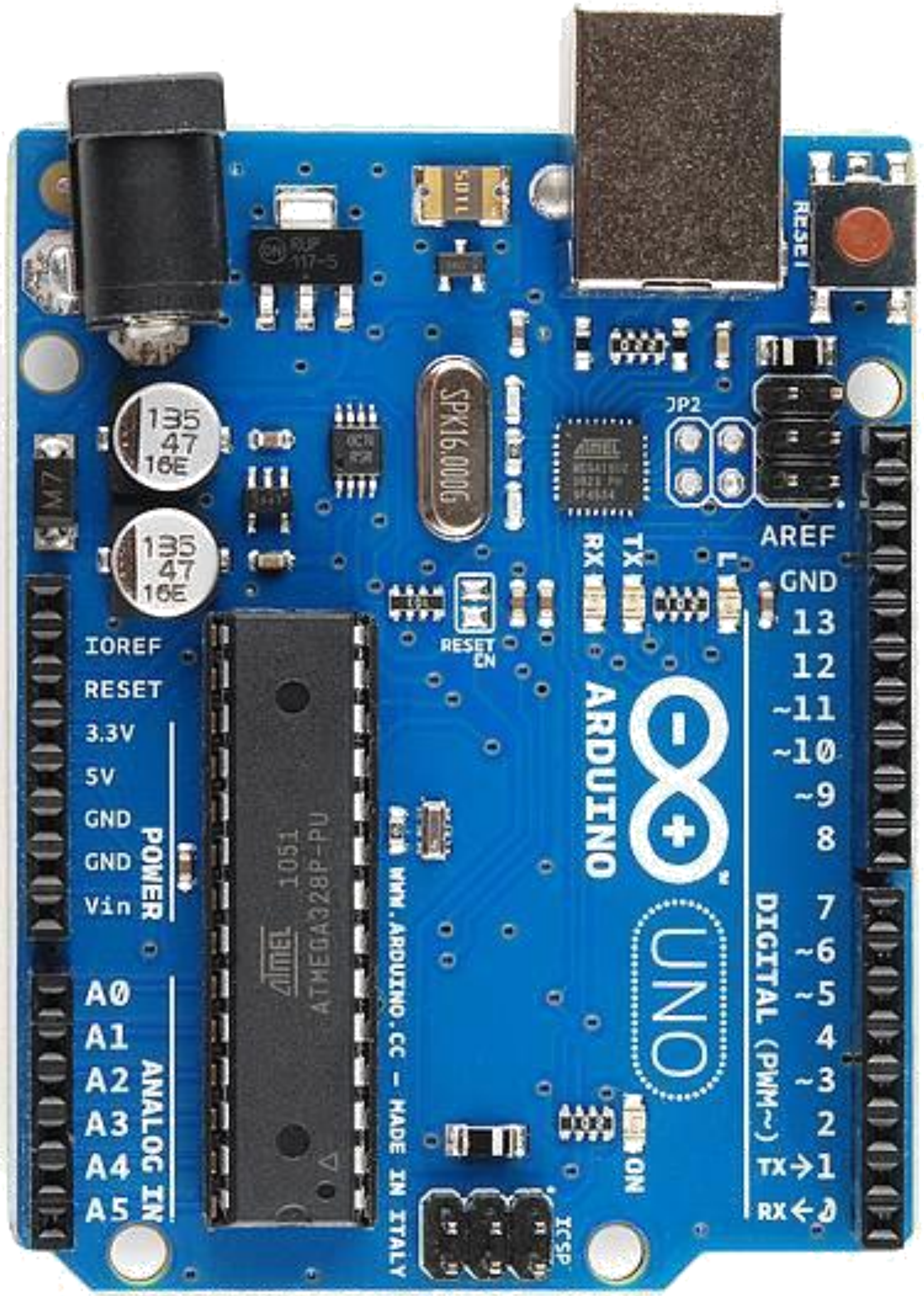
SCL\SDA
(I2C Bus)

POWER
5V / 3.3V / GND

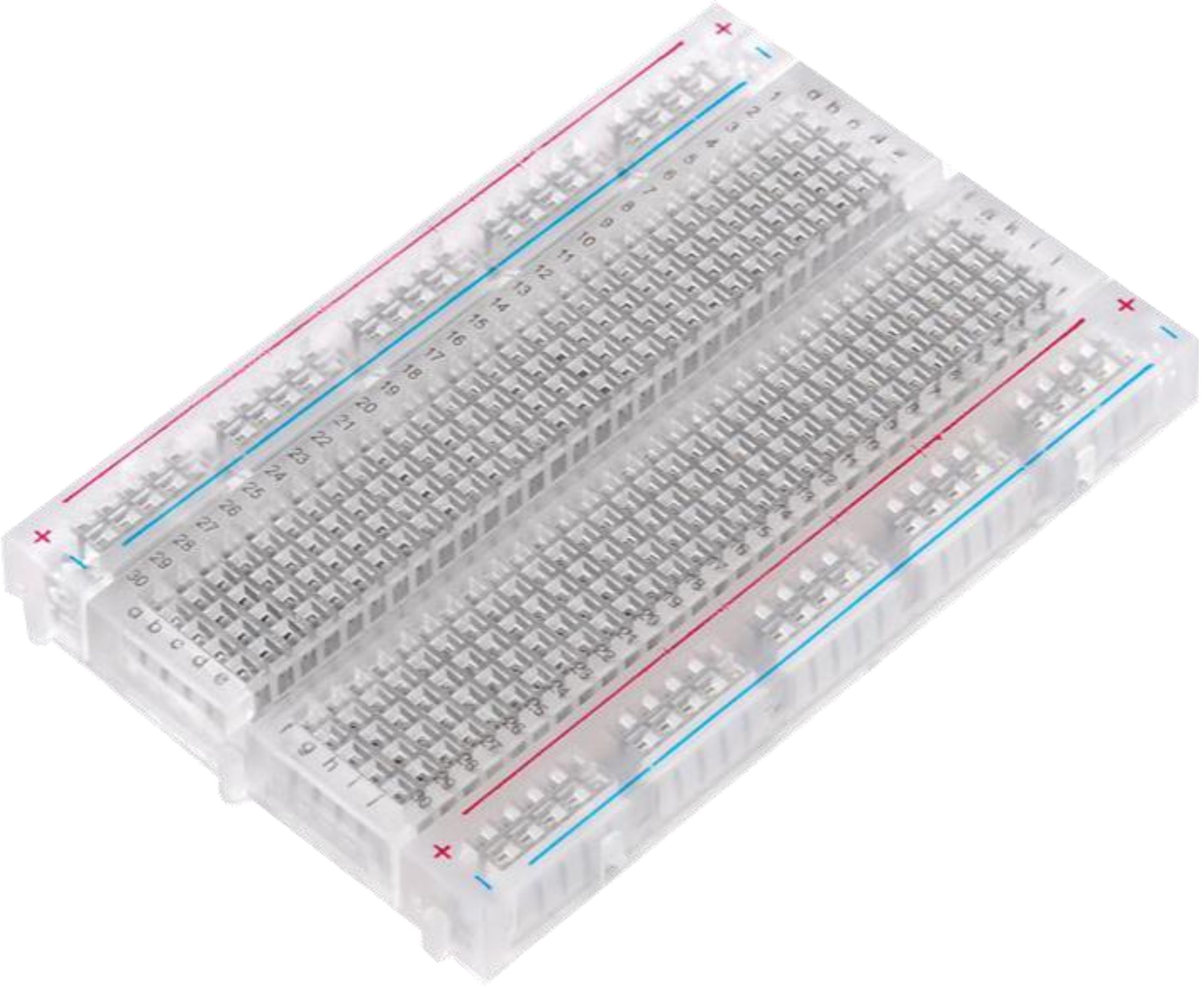
Digital I\O
PWM(3, 5, 6, 9, 10, 11)

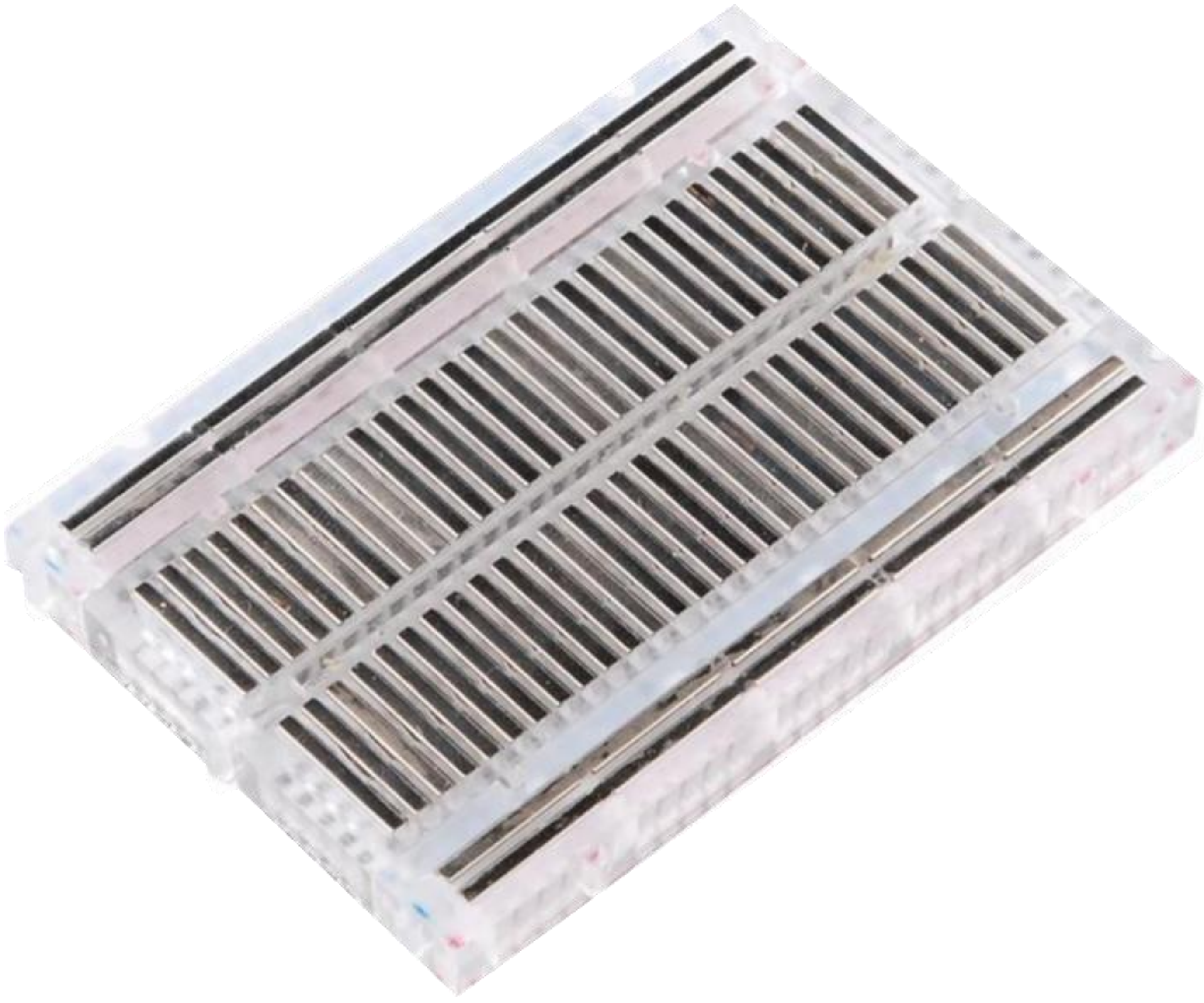
Analog
INPUTS

Go ahead and plug your board in!



What’s a Breadboard?

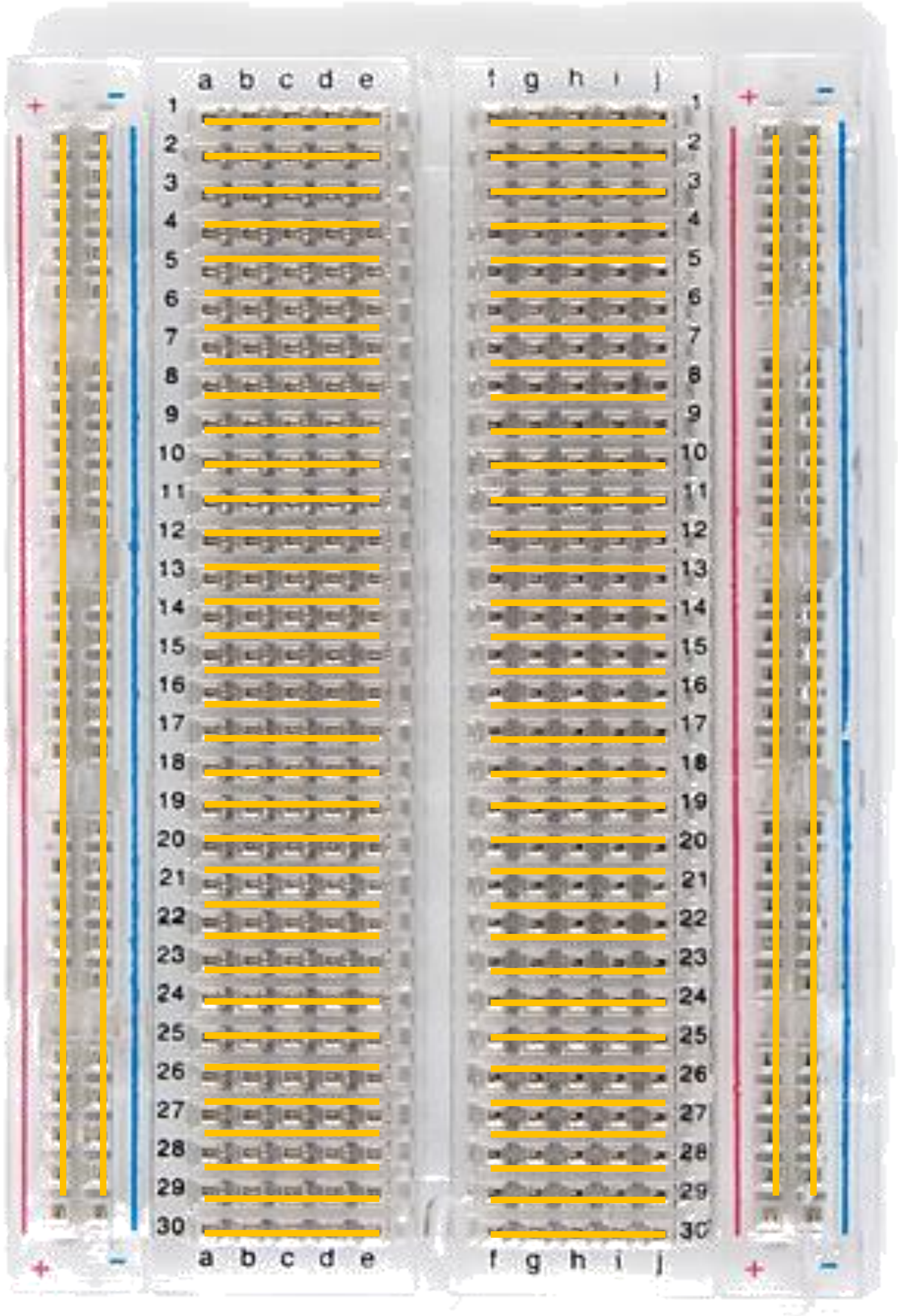




Solderless Breadboard

- Each row (horiz.) of 5 holes are connected.

- Vertical columns – called power bus are connected vertically



Concepts: INPUT vs. OUTPUT

Referenced from the perspective of the microcontroller (electrical board).

Inputs is a signal / information going into the board.

Output is any signal exiting the board.

Almost all systems that use physical computing will have some form of output

Concepts: INPUT vs. OUTPUT

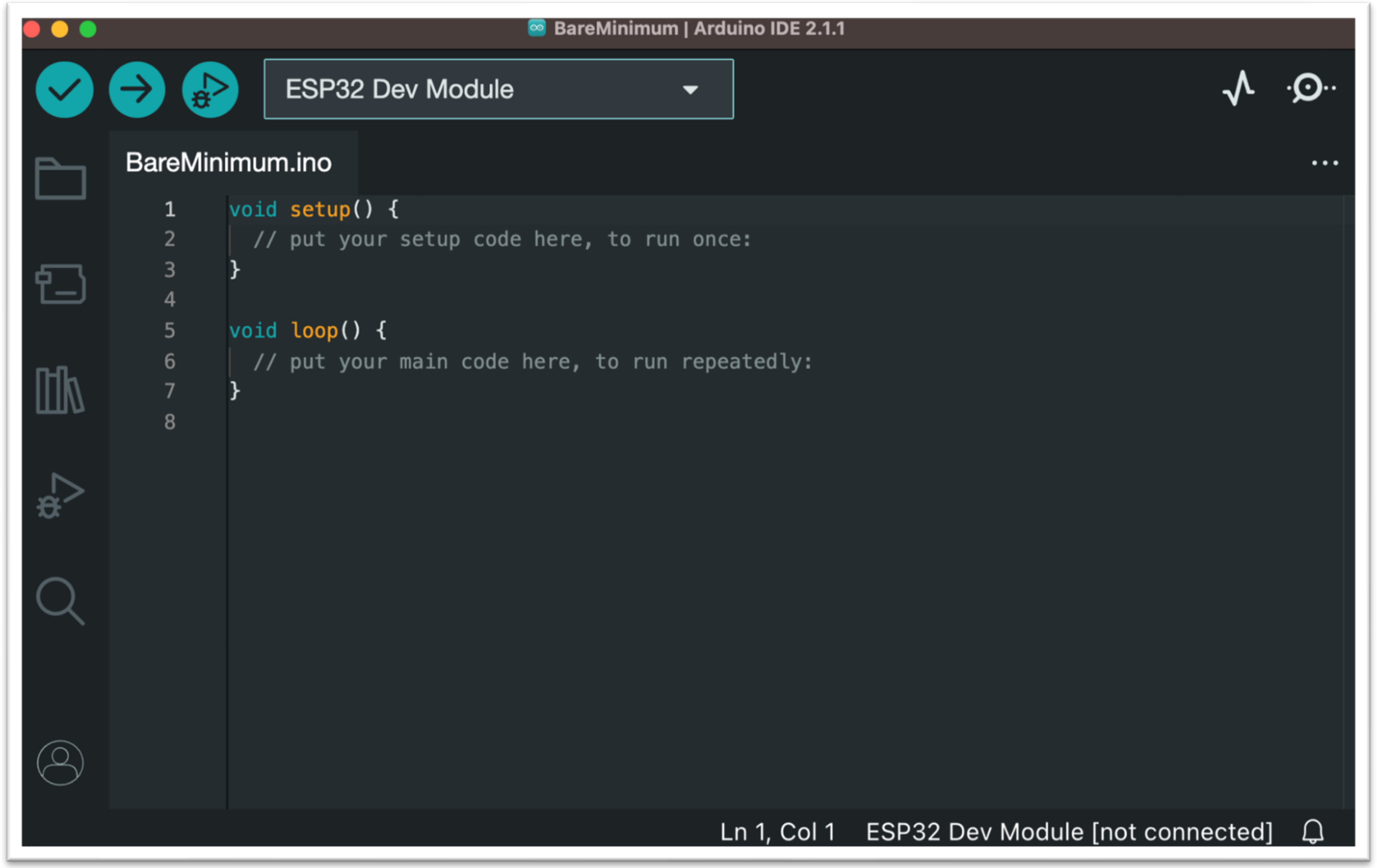
Referenced from the perspective of the microcontroller (electrical board).

Inputs is a signal / information going into the board.

Output is any signal exiting the board.

Examples: Buttons Switches, Light Sensors, Flex Sensors, Humidity Sensors, Temperature Sensors…	Examples: LEDs, DC motor, servo motor, a piezo buzzer, relay, an RGB LED

Arduino IDE



BareMinimum | Arduino IDE 2.1.1

ESP32 Dev Module

BareMinimum.ino

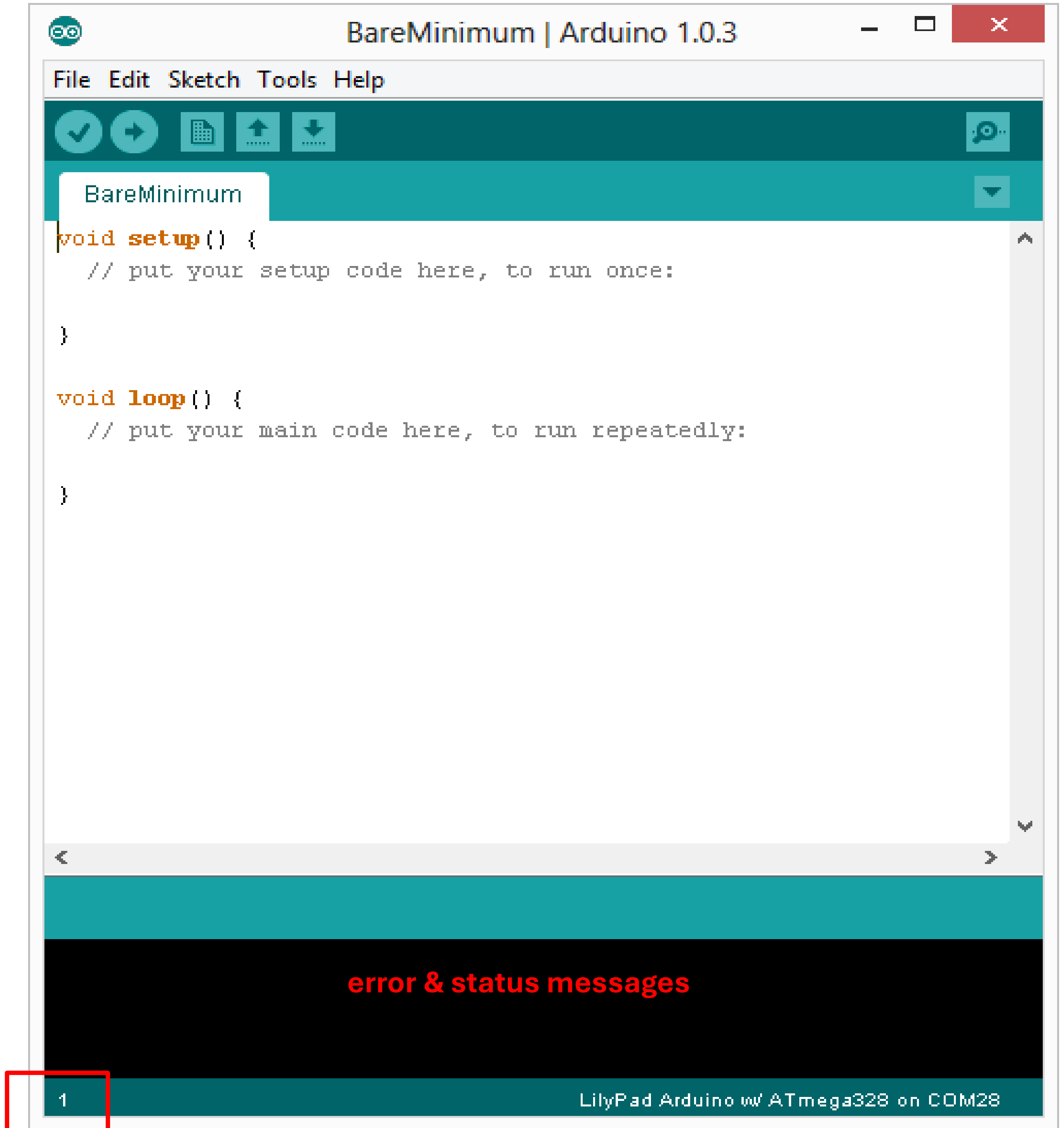
```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Ln 1, Col 1   ESP32 Dev Module [not connected]

IDE (Integrated Development Environment) is an official software by Arduino.cc used for writing, compiling, and uploading code to Arduino boards.

Arduino is similar to regular C++, with additional libraries for specific board functions.

Arduino
Integrated Development Environment (IDE)



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

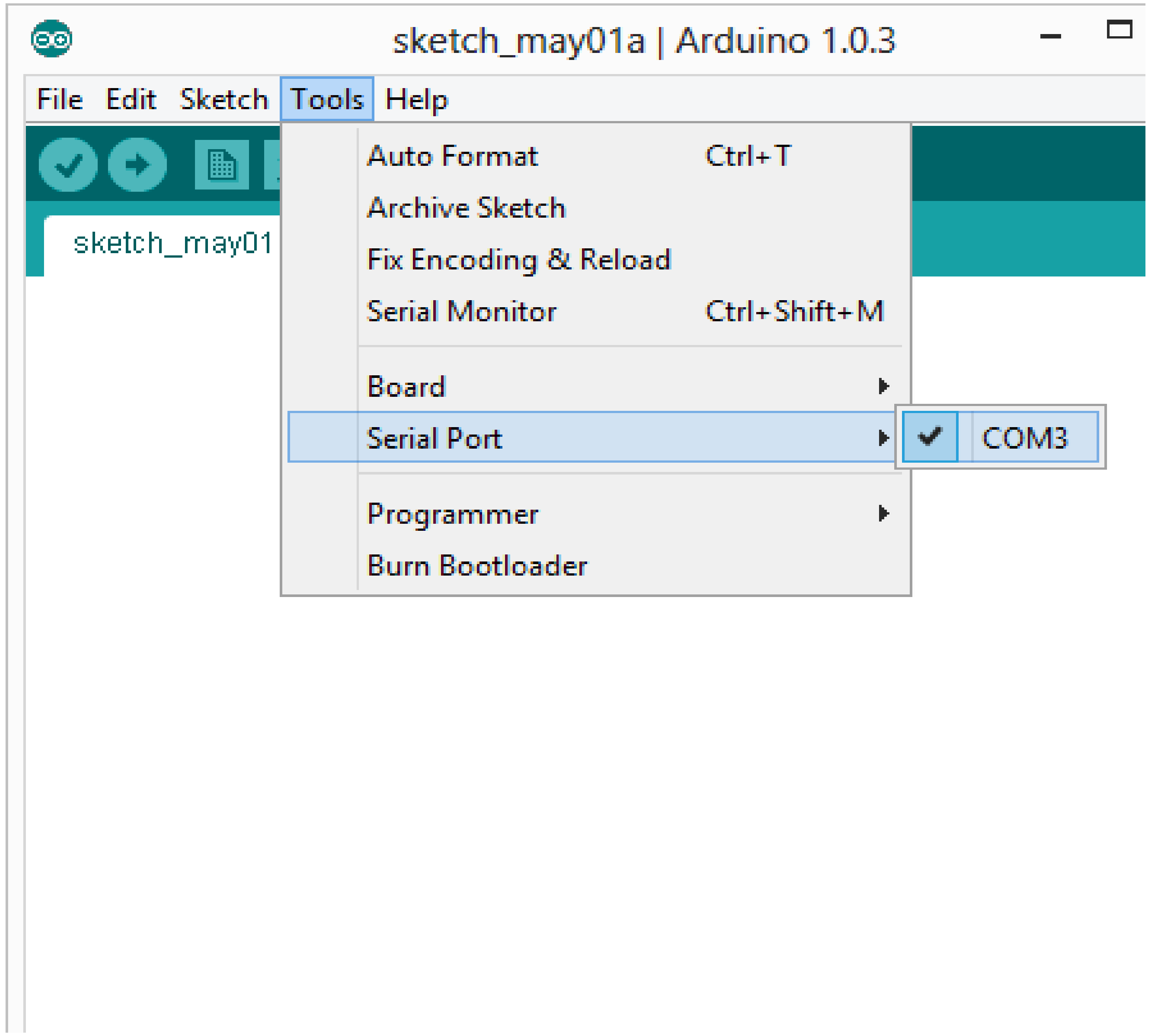
error & status messages

1 LilyPad Arduino w/ ATmega328 on COM28

Two required functions / methods / routines:

```
void setup()
{
        // runs once
}

void loop()
{
        // repeats
}
```

Settings: Tools → Serial Port

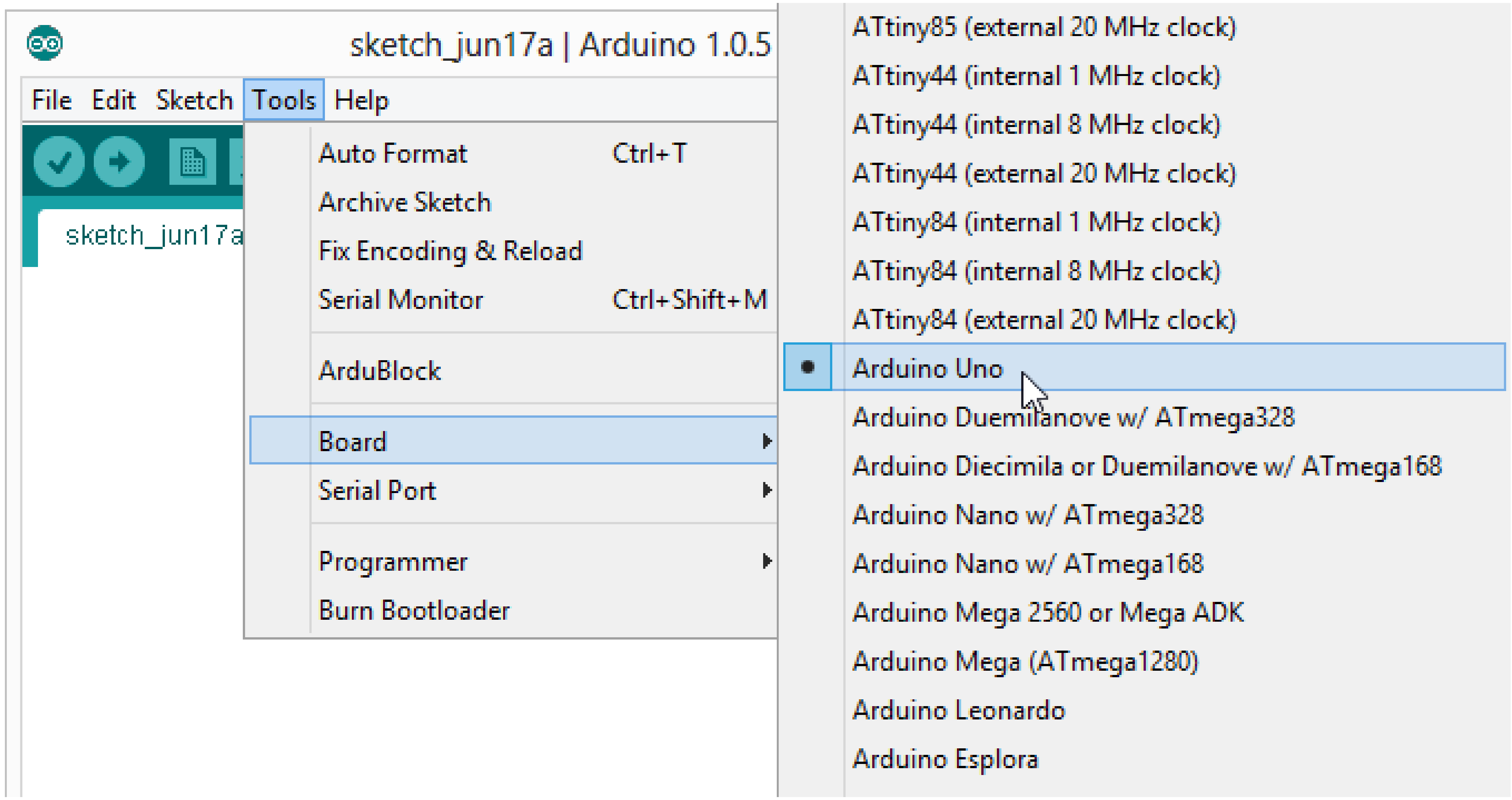


Serial Port ✓ COM3

- Your computer communicates to the Arduino microcontroller via a serial port → through a USB-Serial adapter.

- Check to make sure that the drivers are properly installed.

Settings: Tools → Board


sketch_jun17a | Arduino 1.0.5
File
Edit
Sketch
Tools
Help
sketch_jun17a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
ArduBlock
Board
Serial Port
Programmer
Burn Bootloader
ATtiny85 (external 20 MHz clock)
ATtiny44 (internal 1 MHz clock)
ATtiny44 (internal 8 MHz clock)
ATtiny44 (external 20 MHz clock)
ATtiny84 (internal 1 MHz clock)
ATtiny84 (internal 8 MHz clock)
ATtiny84 (external 20 MHz clock)
● Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Esplora

- Next, double-check that the proper board is selected under the Tools→Board menu.

Three commands to know...

- **pinMode**(pin, INPUT/OUTPUT);
-     ex: **pinMode**(13, OUTPUT);

- **digitalWrite**(pin, HIGH/LOW);
-     ex: digitalWrite(13, HIGH);

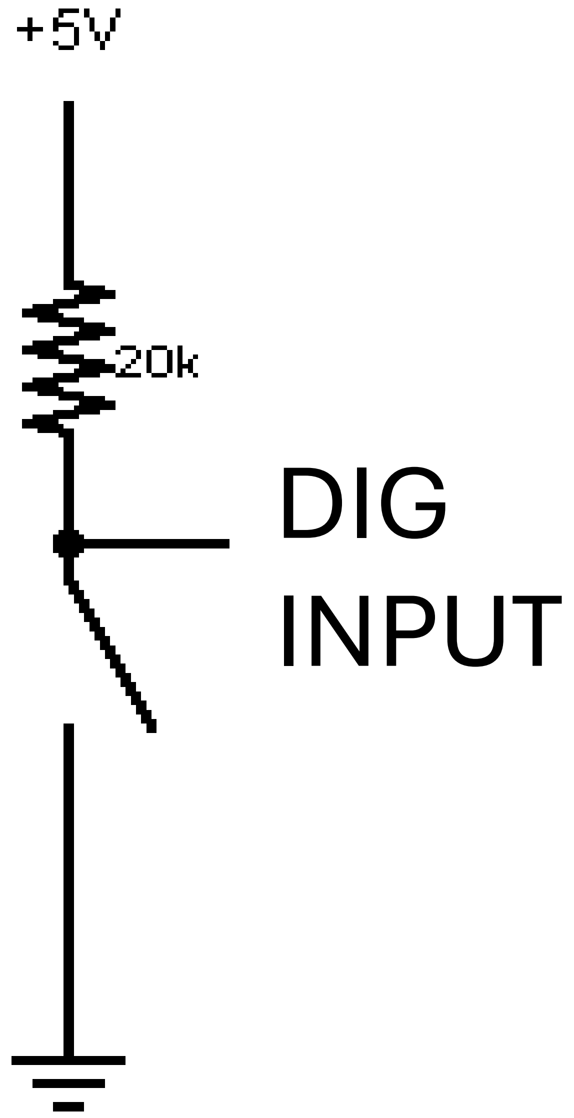
- **delay**(time_ms);
-     ex: delay(2500); // delay of 2.5 sec.

- **// NOTE: -> commands are CASE-sensitive**

Programming: Conditional Statements

`if()`

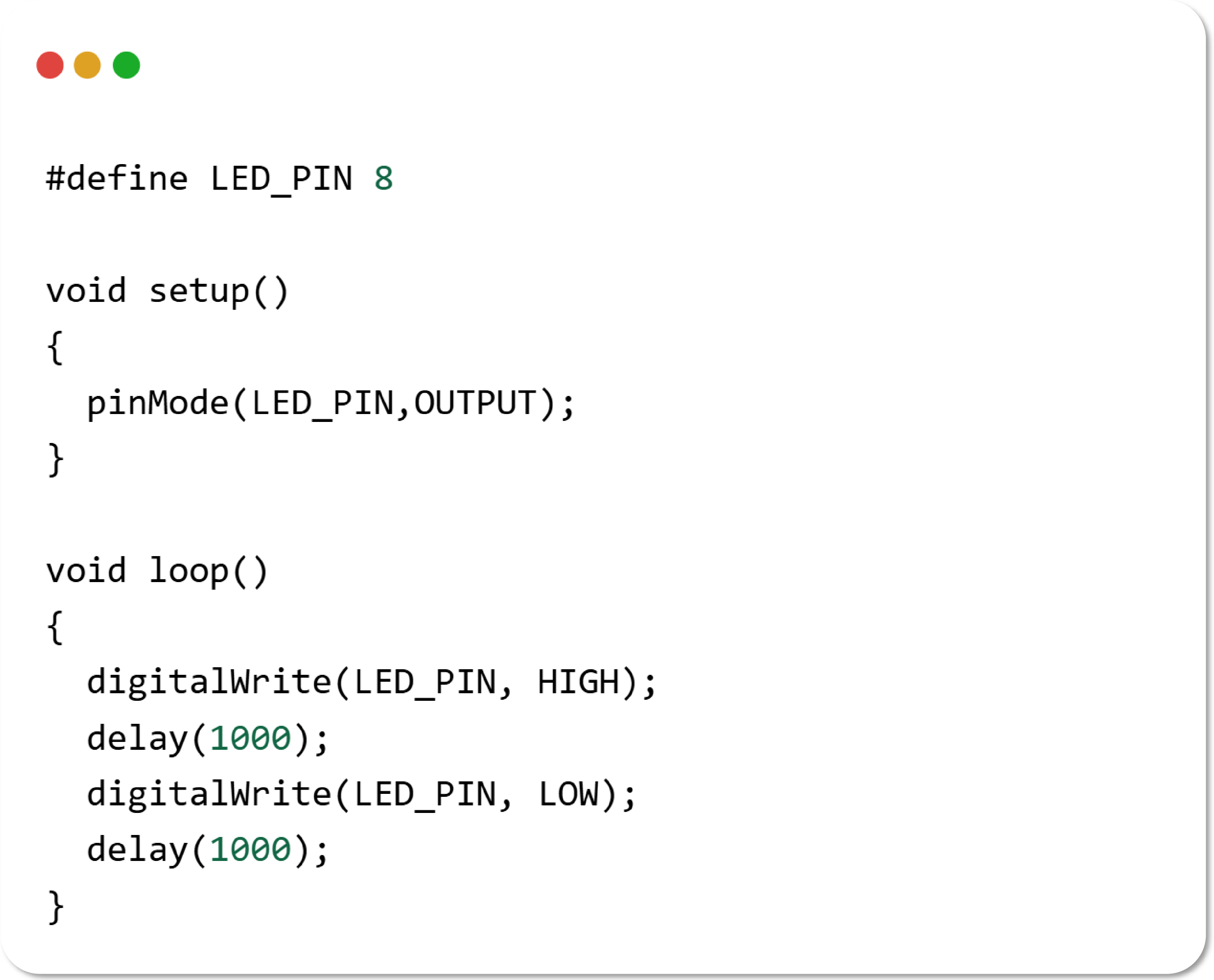
```
void loop()
{
    int buttonState =
digitalRead(5);
    if(buttonState == LOW)
    {    // do something
    }
    else
    {   // do something else
    }
```



Task 1: Blink the LED

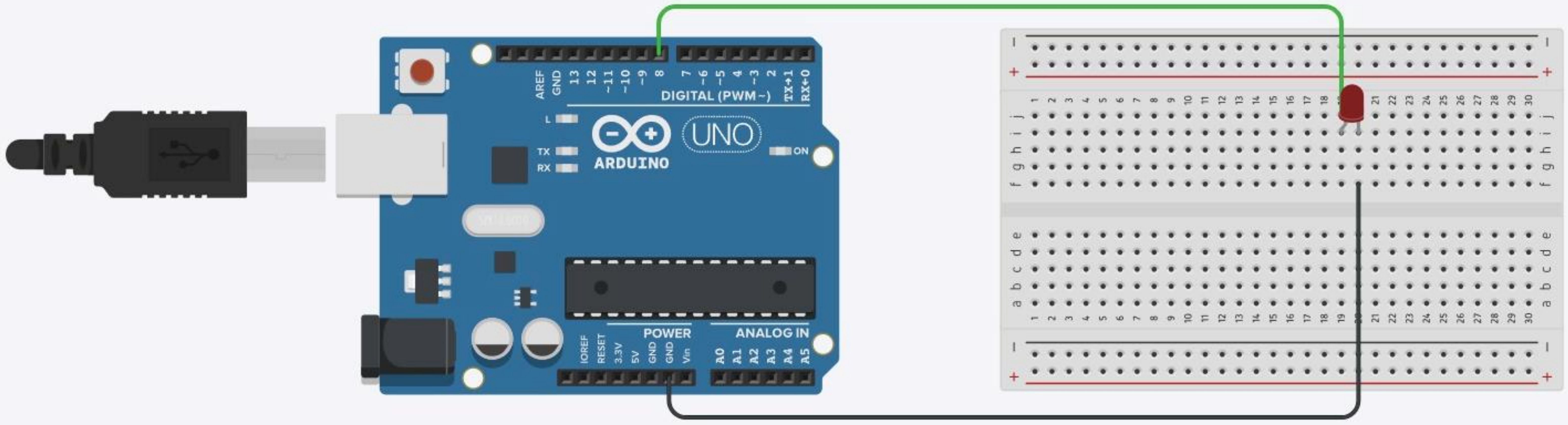
“Hello World” of Physical Computing

Let's explore how to make an LED blink using an Arduino. Here is the code you'll need to complete this task.



```
#define LED_PIN 8

void setup()
{
  pinMode(LED_PIN,OUTPUT);
}

void loop()
{
  digitalWrite(LED_PIN, HIGH);
  delay(1000);
  digitalWrite(LED_PIN, LOW);
  delay(1000);
}
```

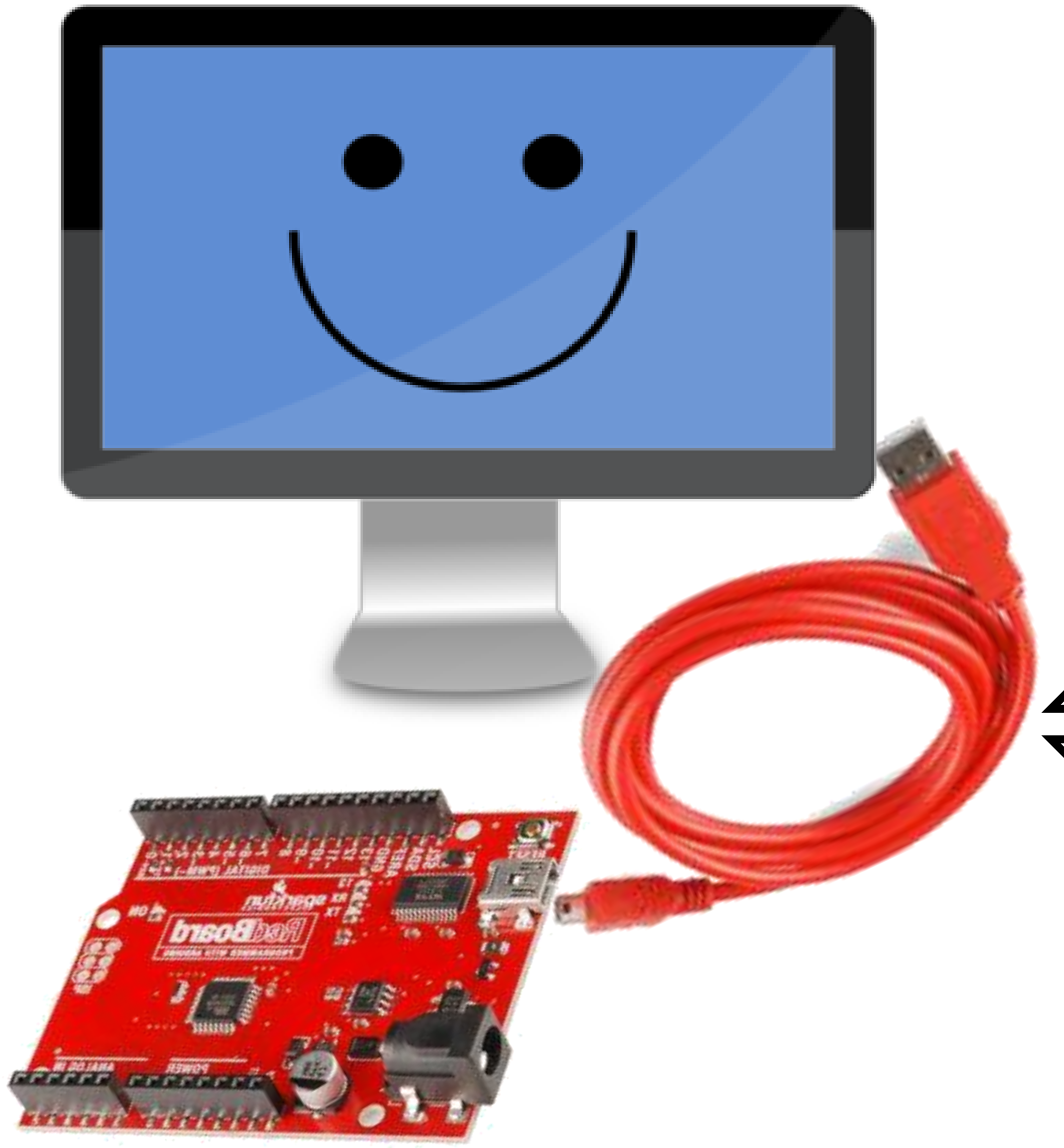
Task 1: Blink the LED



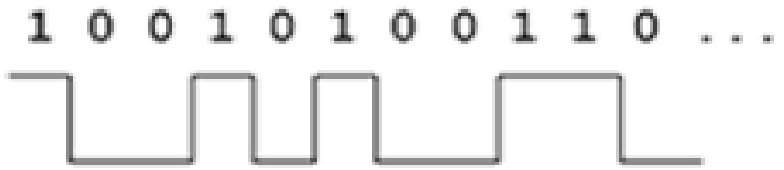
Connect VCC and GND through a resistor as shown

Using Serial Communication

Method used to transfer data between two devices.

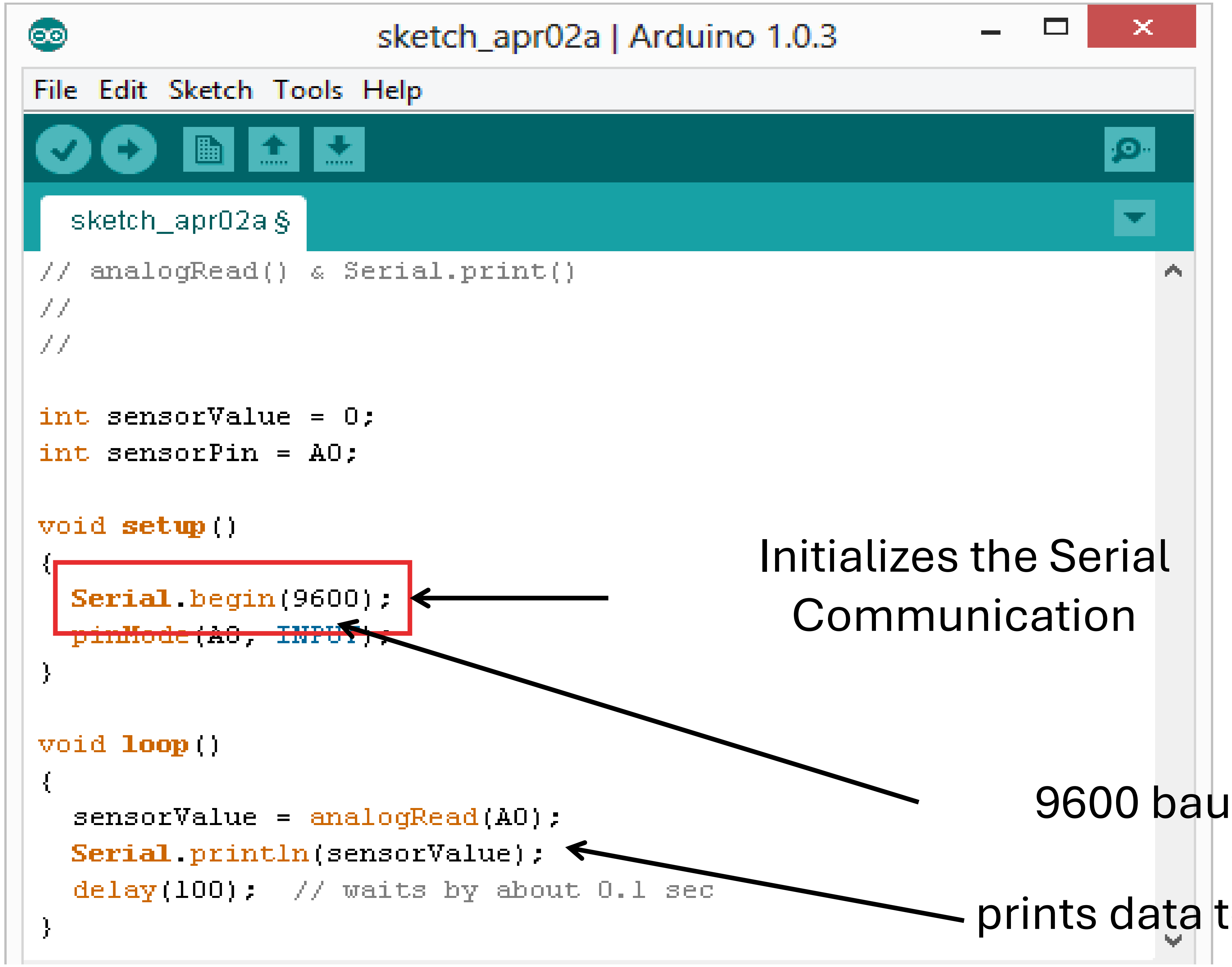


Data passes between the computer and Arduino through the USB cable. Data is transmitted as zeros (‘0’) and ones (‘1’) sequentially.



Arduino dedicates Digital I/O pin # 0 to receiving and Digital I/O pin #1 to transmit.

Serial Monitor & analogRead()



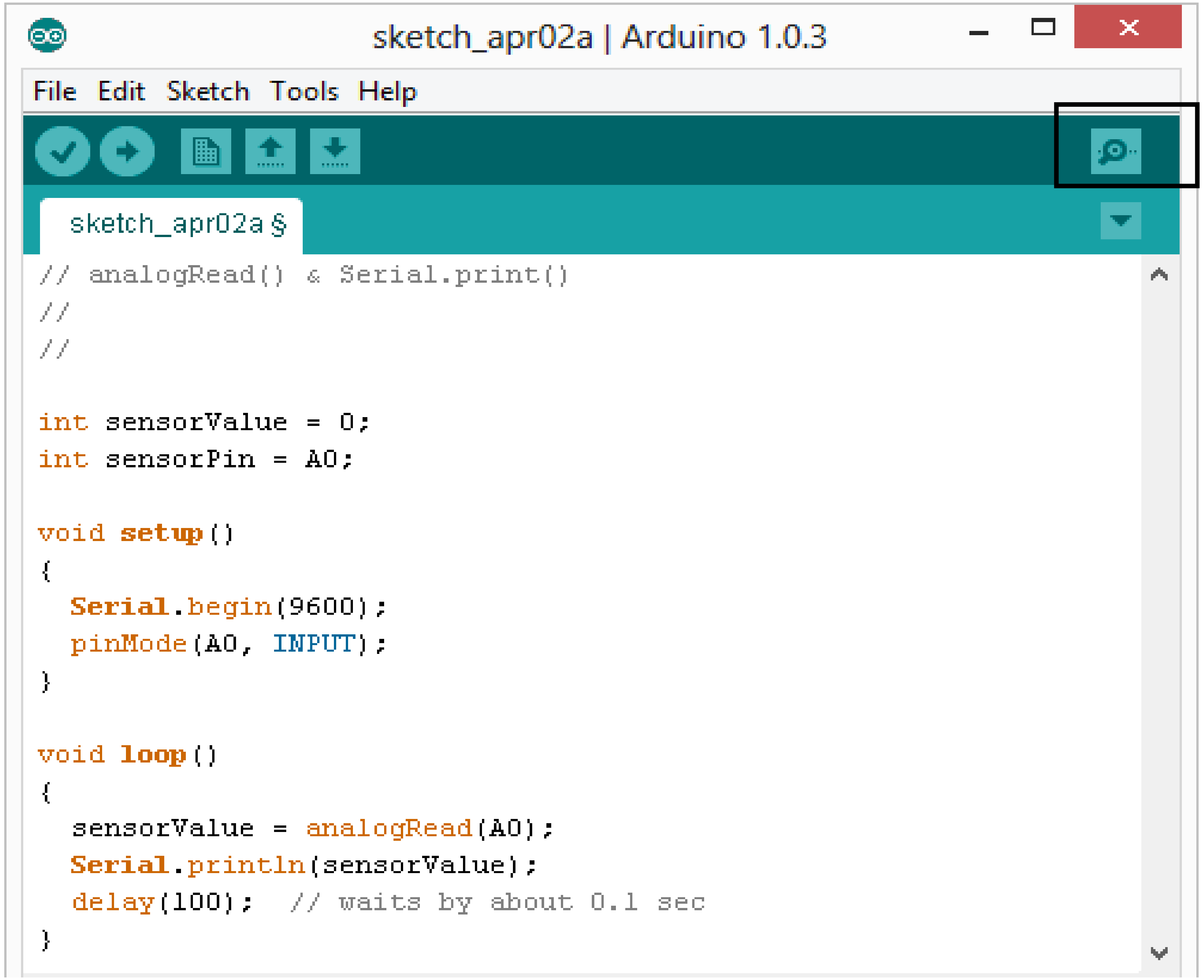
```
// analogRead() & Serial.print()
//
//

int sensorValue = 0;
int sensorPin = A0;

void setup()
{
  Serial.begin(9600);
  pinMode(A0, INPUT);
}

void loop()
{
  sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay(100);   // waits by about 0.1 sec
}
```

Initializes the Serial Communication

9600 baud data rate

prints data to serial bus

Serial Monitor & analogRead()

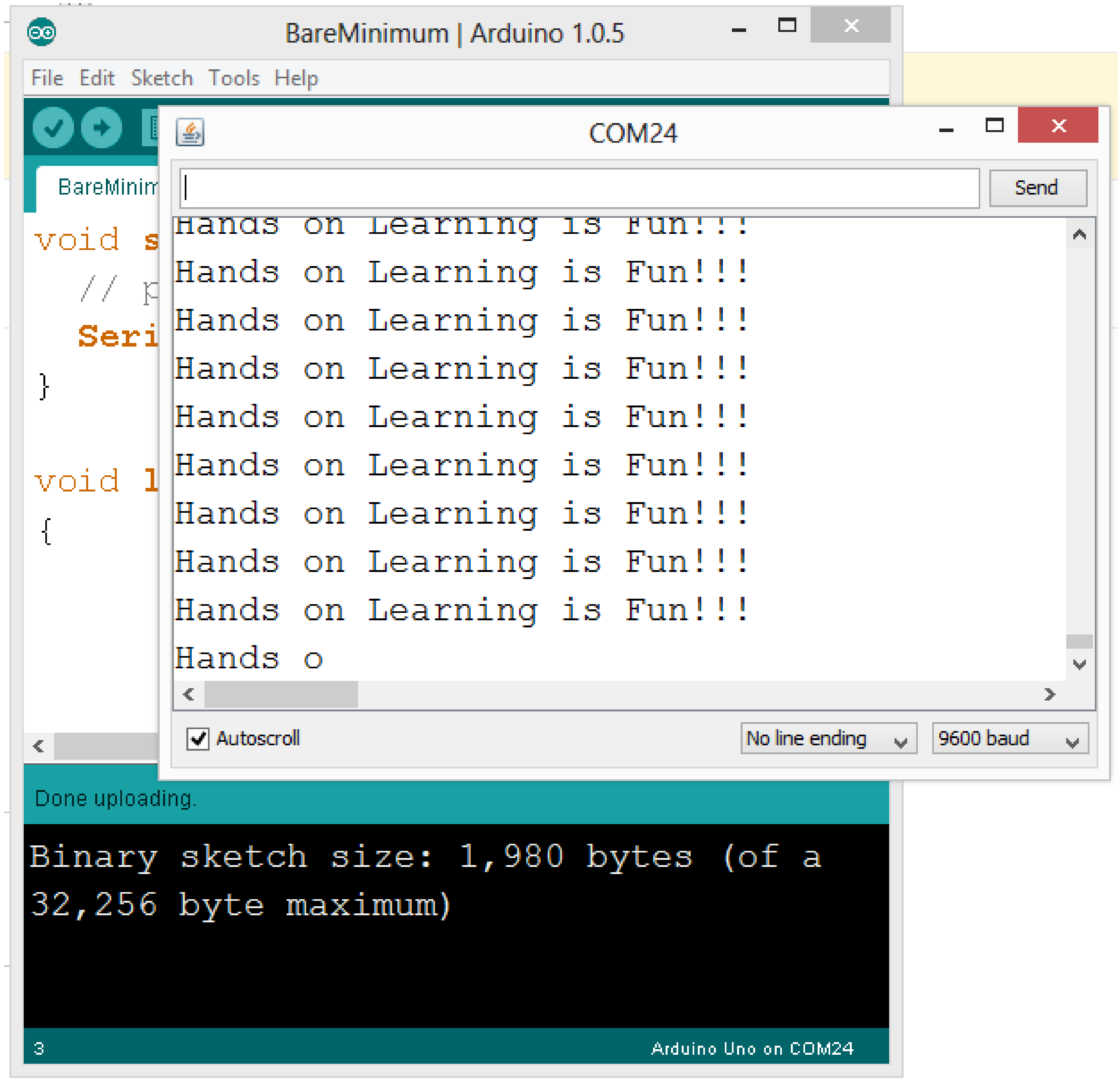


```
// analogRead() & Serial.print()
//
//

int sensorValue = 0;
int sensorPin = A0;

void setup()
{
  Serial.begin(9600);
  pinMode(A0, INPUT);
}

void loop()
{
  sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay(100);  // waits by about 0.1 sec
}
```

Opens up a Serial Terminal Window

Additional Serial Communication: Sending a Message

```
void loop ( )
{
 Serial.print(“Hands on “) ;
 Serial.print(“Learning ”) ;
•  Serial.println(“is Fun!!!”) ;


}
```



```
void s
  // p
  Seri
}

void l
{
```

```
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands on Learning is Fun!!!
Hands o
```

```
Done uploading.
```

```
Binary sketch size: 1,980 bytes (of a
32,256 byte maximum)
```