Arduino

Soumay Gupta
ECE, IInd yr.

Microcontrollers – Recap

- A microcontroller can be considered as a very small and simple version of a computer on a single IC which is used for a specific purpose.

- It has a CPU, flash memory ,RAM, EEPROM and many on- chip peripherals .

Terms in MCU

- Clock : Oscillating Signal with fixed time period
- Clock source : Generally Crystal Oscillators
- Flash memory : Stores the program
- SRAM : Static RAM. Volatile Memory.
- EEPROM : Electrically Erasable Programmable ROM. Permanent Memory.
- VCC : Power Supply
- GND : Ground

Terms in MCU

- PORT : The point where data internal to the MCU comes out. Simply, a combination of pins in an MCU is a PORT. A port contains 8 GPIO(General Purpose Input Output pins).
- ADC : ADC ports receive analog signals and convert them into digital number within certain numerical range depending on the resolution.
- PWM: Used to generate analog outputs depending on value of duty cycle instead of regular digital outputs by MCU.
- Motor Driver : It is an external driver circuit which acts as a bridge between IC and motors.

Development Boards

- Development Boards are used to control to the microcontroller.
- They are printed circuit boards that provide all the circuitry necessary for a useful control task like I/O circuit, clock generator, stored program etc.

- There are 2 types of development boards in particular.
  - 1. Arduino
  - 2. AVR

Arduino

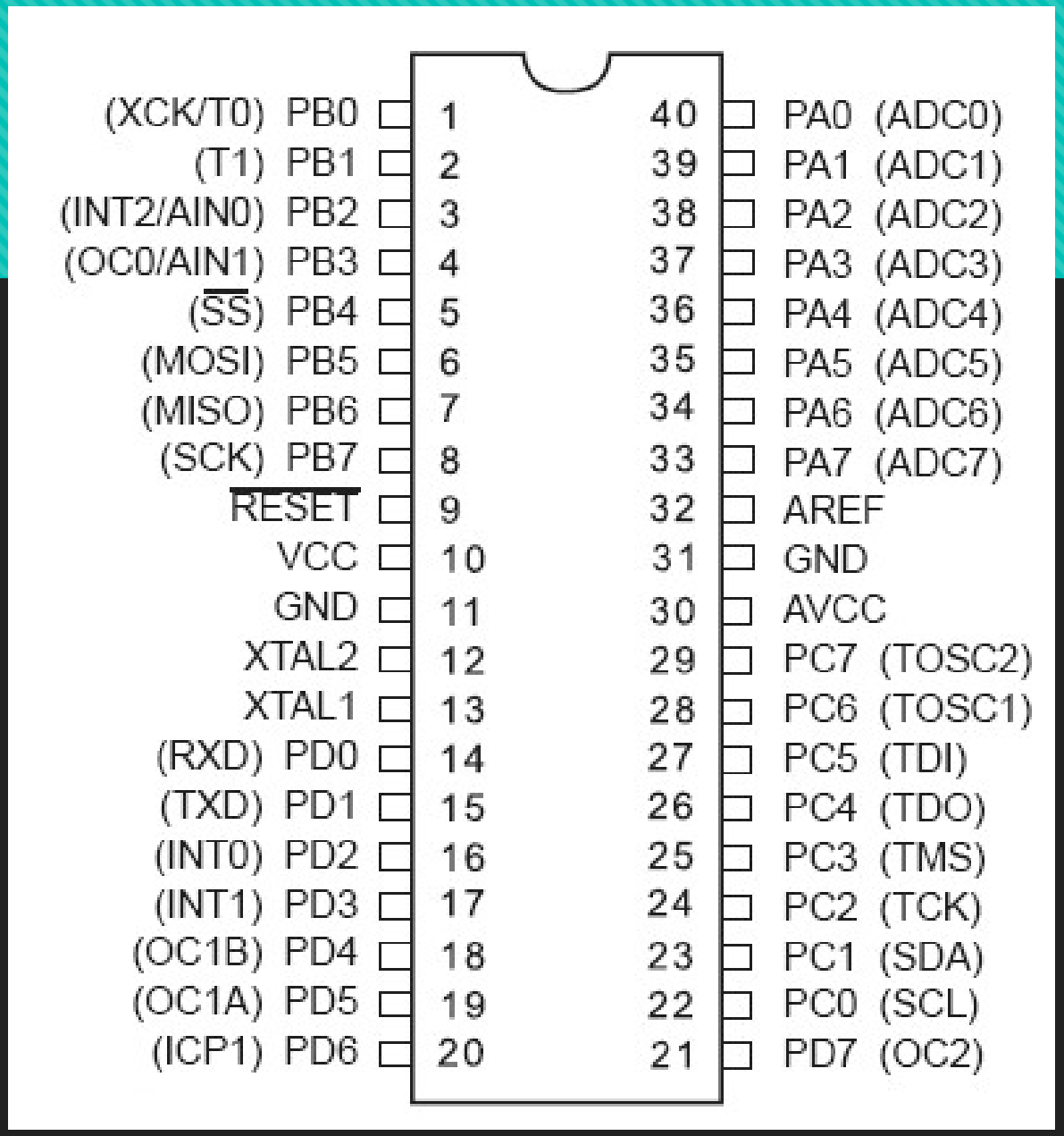
- Arduino is an AVR-based prototyping board with an emphasis on ease of use.
- It has separate pins for digital and analog purposes.
- There are 6 analog in pins and 14 digital in/out pins ground, Vcc …
- It consists of PWM output pins marked separately.
- It also features a USB interface allowing serial communication through a USB device, eliminating the need for a separate AVR programmer.


Analog Reference
14 digital pins ... in & out
USB Connector
Power Connector
6 analog in pins
MADE IN ITALY
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
RESET
ICSP
POWER
RESET
3.3V
5V
GND
GND
Vin
ANALOG IN
A0
A1
A2
A3
A4
A5

AVR

- AVR is a microcontroller manufactured by Atmel.
- Atmega 16A is used in AVR and Atmega 328 is used in Arduino.
- In this, communication between PC and MCU is done using a programmer.
- It consists of a on-board driver.

ATmega16A


(XCK/T0) PB0 1
(T1) PB1 2
(INT2/AIN0) PB2 3
(OC0/AIN1) PB3 4
(SS) PB4 5
(MOSI) PB5 6
(MISO) PB6 7
(SCK) PB7 8
RESET 9
VCC 10
GND 11
XTAL2 12
XTAL1 13
(RXD) PD0 14
(TXD) PD1 15
(INT0) PD2 16
(INT1) PD3 17
(OC1B) PD4 18
(OC1A) PD5 19
(ICP1) PD6 20
40 PA0 (ADC0)
39 PA1 (ADC1)
38 PA2 (ADC2)
37 PA3 (ADC3)
36 PA4 (ADC4)
35 PA5 (ADC5)
34 PA6 (ADC6)
33 PA7 (ADC7)
32 AREF
31 GND
30 AVCC
29 PC7 (TOSC2)
28 PC6 (TOSC1)
27 PC5 (TDI)
26 PC4 (TDO)
25 PC3 (TMS)
24 PC2 (TCK)
23 PC1 (SDA)
22 PC0 (SCL)
21 PD7 (OC2)

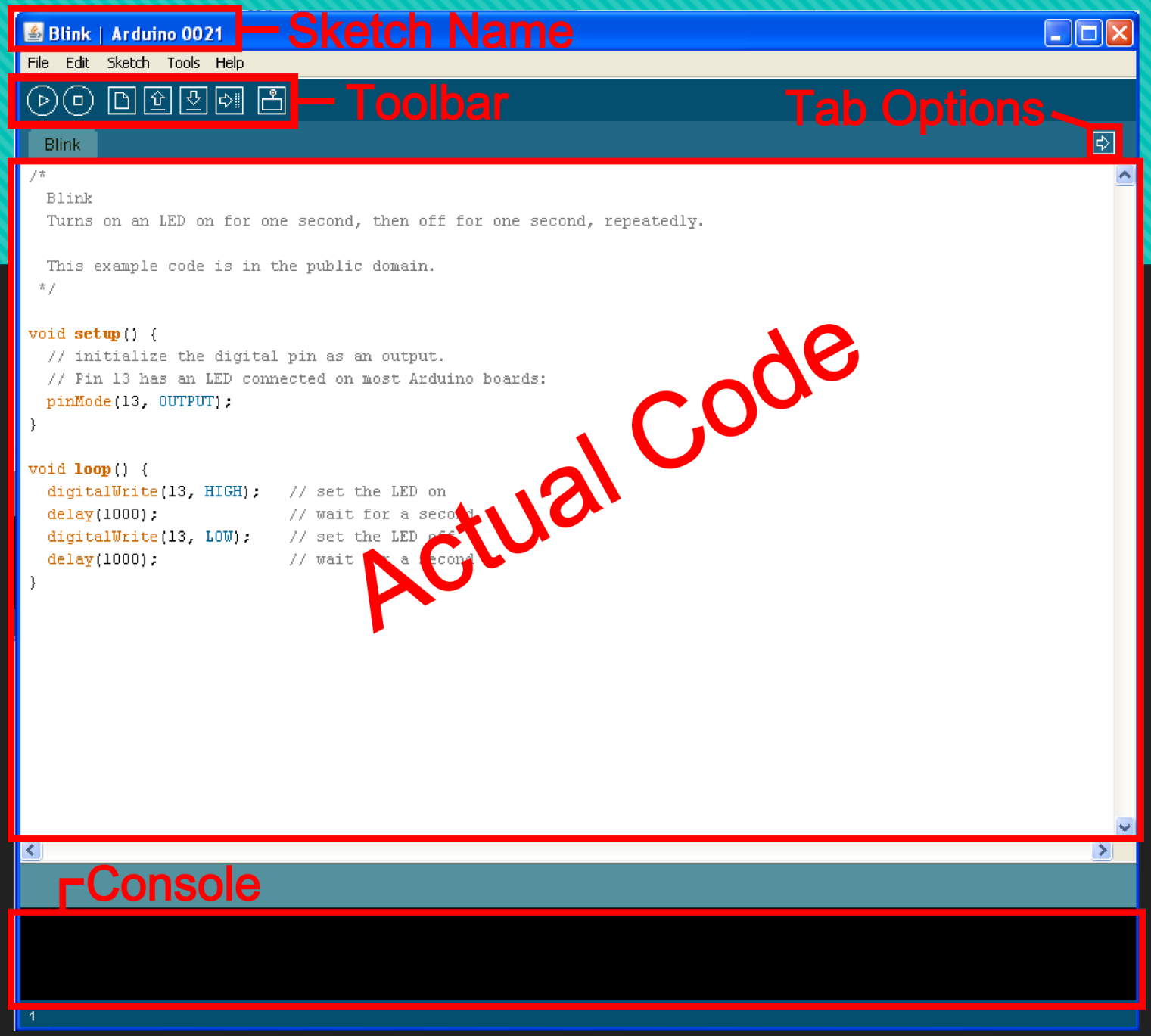
Arduino vs AVR

Arduino	AVR
It is easier to use	It is more difficult to use
It requires little knowledge	It requires much knowledge
No need to know the in-build circuits	One needs to know about how the internal circuits are implemented
It is not as powerful as AVR	More powerful than Arduino
Meant for an amateur	Meant for a professional

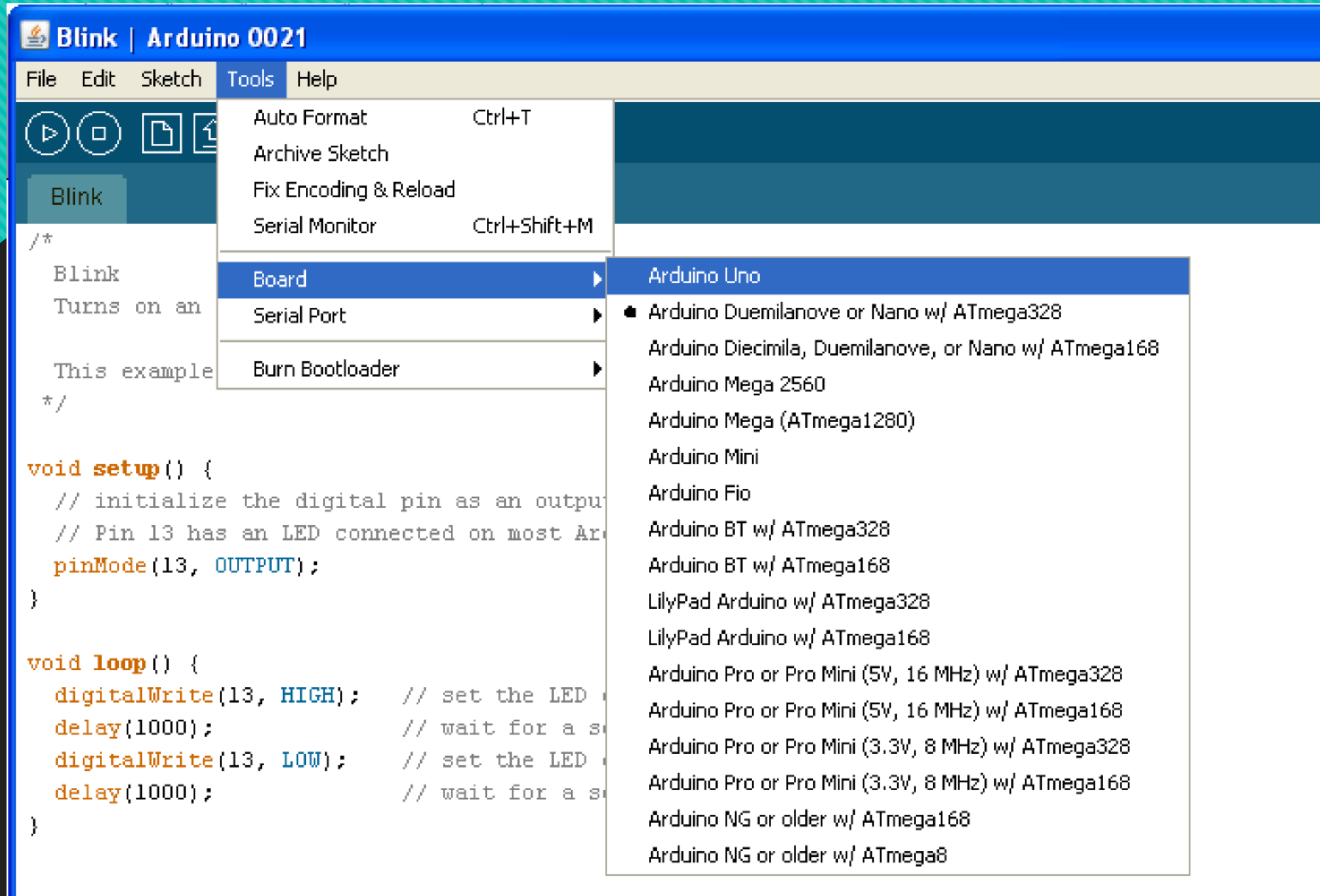
Why Arduino?

- The main reason is its programming environment.
- IDE includes many helpful libraries .
- While programming Arduino, you are least bothered with the internal hardware and registers of the microcontroller. All you do is call the functions written in the libraries (which are already provided)
- Generally used if you want to prototype your project very fast, and are not much concerned about the programming part.

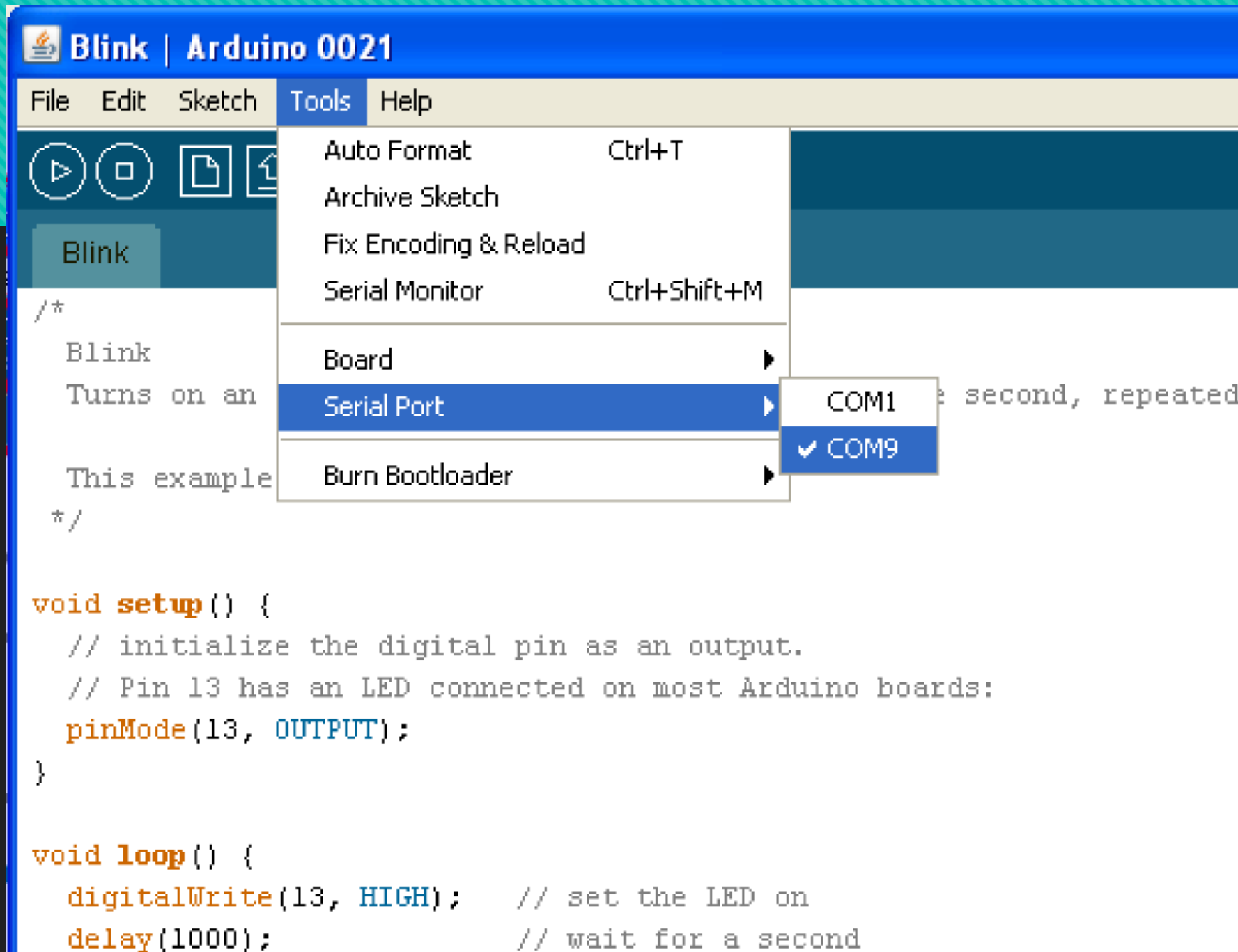
Arduino Environment



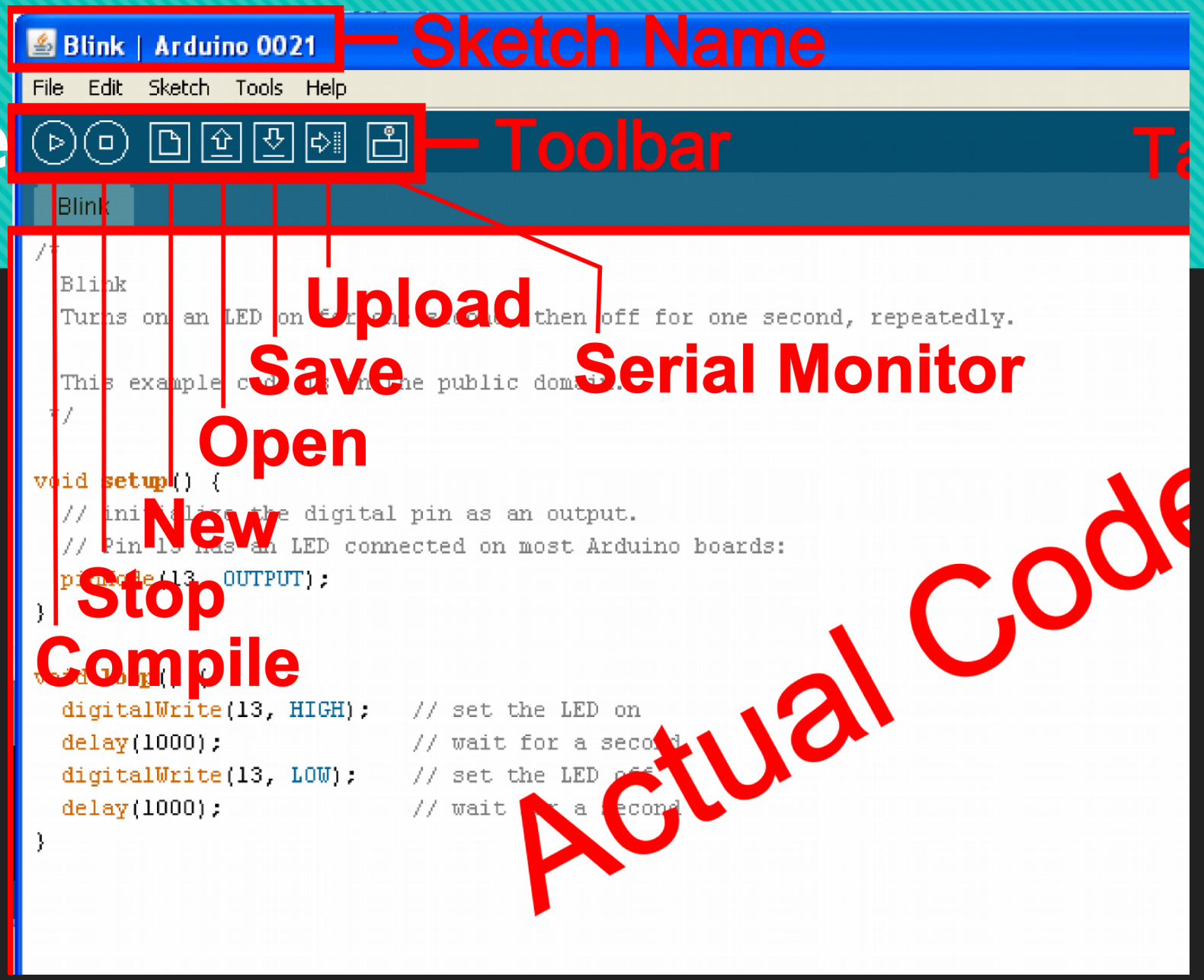
Board Type


Blink | Arduino 0021
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Burn Bootloader
Arduino Uno
Arduino Duemilanove or Nano w/ ATmega328
Arduino Diecimila, Duemilanove, or Nano w/ ATmega168
Arduino Mega 2560
Arduino Mega (ATmega1280)
Arduino Mini
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8
Blink
/*
Blink
Turns on an
This example
*/
void setup() {
// initialize the digital pin as an output
// Pin 13 has an LED connected on most Ar
pinMode(13, OUTPUT);
}
void loop() {
digitalWrite(13, HIGH); // set the LED
delay(1000); // wait for a s
digitalWrite(13, LOW); // set the LED
delay(1000); // wait for a s
}

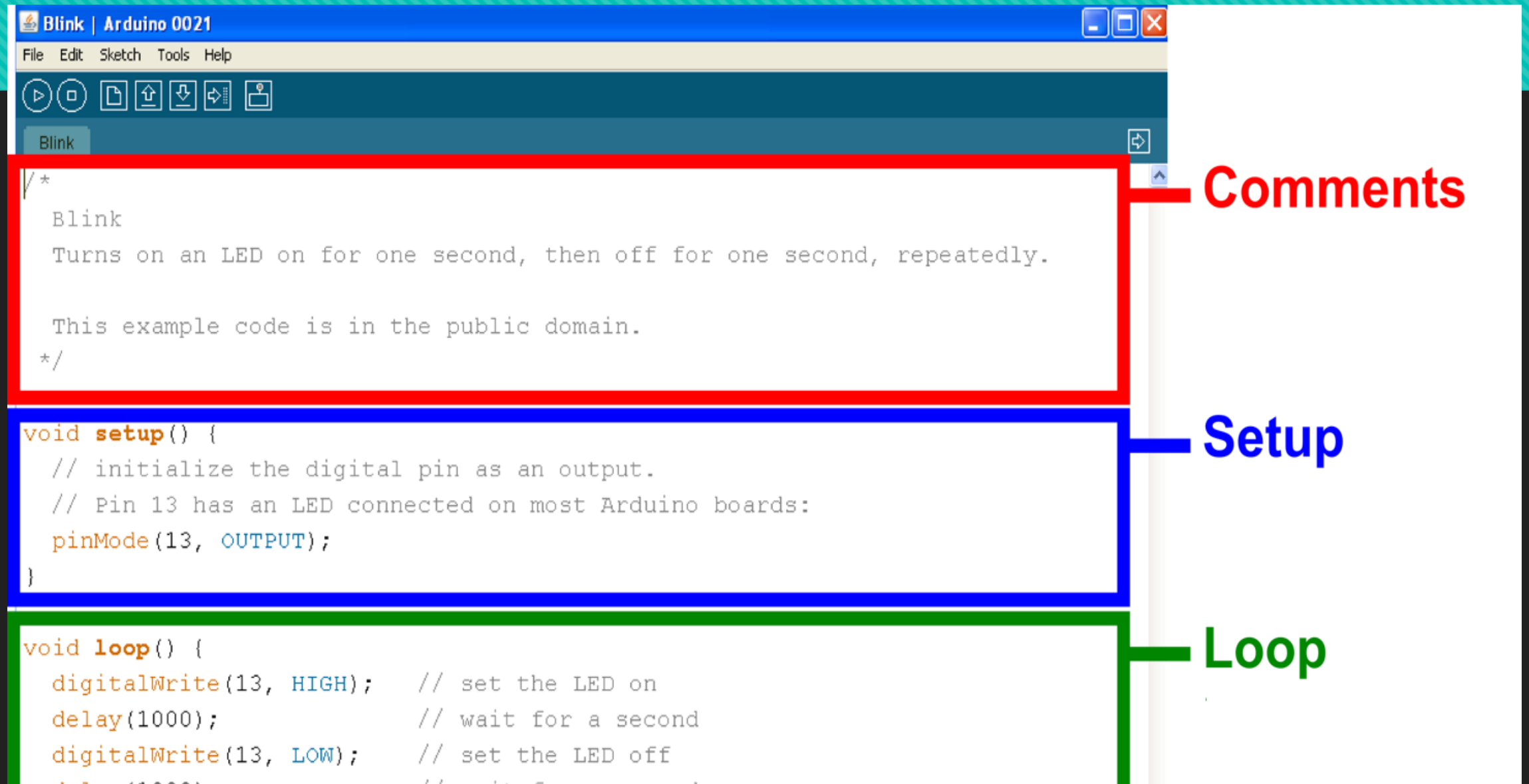
Serial Port/ COM Port



The Environme



Parts of a Sketch



Comments

- Comments can be inserted ANYWHERE
- Comments are created with // (single line) or /* and */ (multiple lines)
- Comments do not affect the code
- Help others to understand what a particular section of the code does

Operators

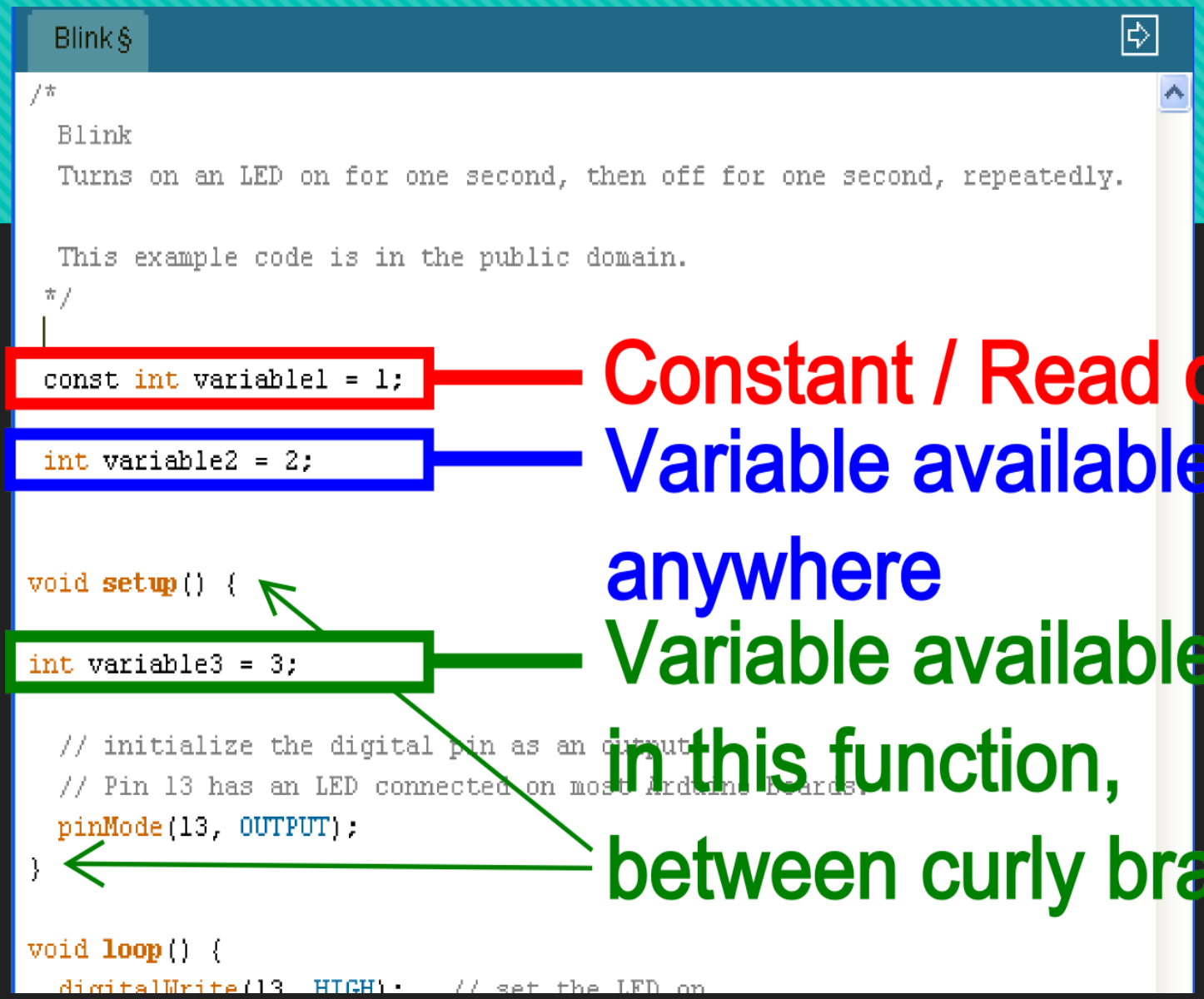
- = To assign a value
- <= , >= , == , ! To compare values
- && Logical And
- || Logical Or

Variables

- Boolean:
  - boolean foo = true;
- Byte, Short, Integer, Long:
  - int foo = 654;
- Float, Double
- String
  - String str = “yolo swag”;
- Character:
  - char hi = ‘A’;

Variable Scope

Where you declare your variables matters



Setup

void setup ( ) { }

```
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}
```

- The setup function comes BEFORE the loop function and is necessary for all Arduino sketches

Setup

void setup ( ) { }

```
void setup() {
  // Initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}
```

- The setup header will never change, everything else that occurs in setup happens inside the curly brackets

Setup

void setup ( ) {

pinMode (pin, mode); }

```
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}
```

- Outputs are declared in setup, this is done by using the pinMode() function
- This particular example declares digital pin #13 as an output, remember to use CAPS

Pin Declaration

```
1  int in1 = A0;
2  int in2 = 10;
3  int out = 6;
4
5  void setup() {
6    pinMode(11, OUTPUT);
7    pinMode(out, OUTPUT);
8    pinMode(in2, INPUT);   //can be omitted
9    pinMode(in1, INPUT);   //can be omitted
10   pinMode(7, INPUT);   //can be omitted
11 }
```

- Default mode is INPUT

Setup

void setup ( ) { Serial.begin(9600); }

```
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}
```

- Serial communication also begins in setup
- This particular example declares Serial communication at the standard baud rate of 9600.

Setup, Internal Pullup Resistors

void setup ( ) {
digitalWrite (12, HIGH); }

```
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
  digitalWrite(12, HIGH);
}
```

- You can also create internal pullup resistors in setup, to do so digitalWrite the pin HIGH

LOOP

void loop ( ) { }

Blink

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeat

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Loop

If Statement

if (this is true) { do this; }

```
void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

If Statement

Else

else { do this; }

```
void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

Else, optional

For Loop

for (int count = 0; count<10; count++) { }

```
void setup()
{
  //Set each pin connected to an LED to output mode (pulling high
  for(int i = 0; i < 8; i++){                    //this is a loop and will r
      pinMode(ledPins[i],OUTPUT);                //we use this to set each LED p
  }                                                //the code this replaces is

  /* (commented code will not run)
   * these are the lines replaced by the for loop above they do e
   * same thing the one above just uses less typing
  pinMode(ledPins[0],OUTPUT);
  pinMode(ledPins[1],OUTPUT);
  pinMode(ledPins[2],OUTPUT);
  pinMode(ledPins[3],OUTPUT);
```

For loop

While Loop

```
int count=0 ;
while ( count<10 )
{
    //looks basically like a “for” loop except the variable is declared before
    //and incremented inside the while loop
    … …
    … …
    count++ ;
}
```

While Loop

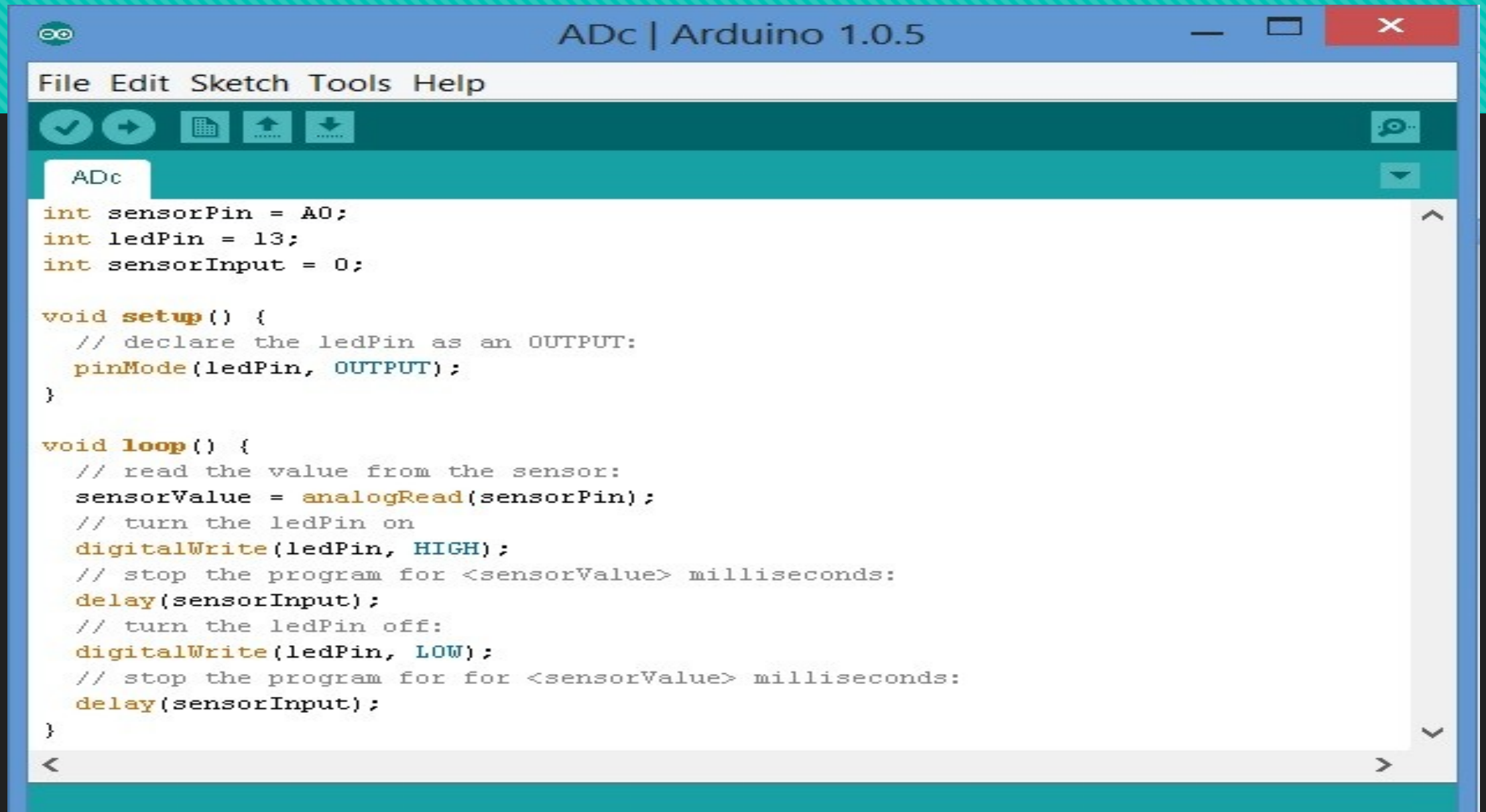
```
while ( digitalRead(buttonPin)==1 )
{
    //instead of changing a variable you just read a pin
    //so the computer exits when you press a button
    //or a sensor is tripped
    ... ...
    ... ...
}
```

Analog to Digital Conversion in Arduino

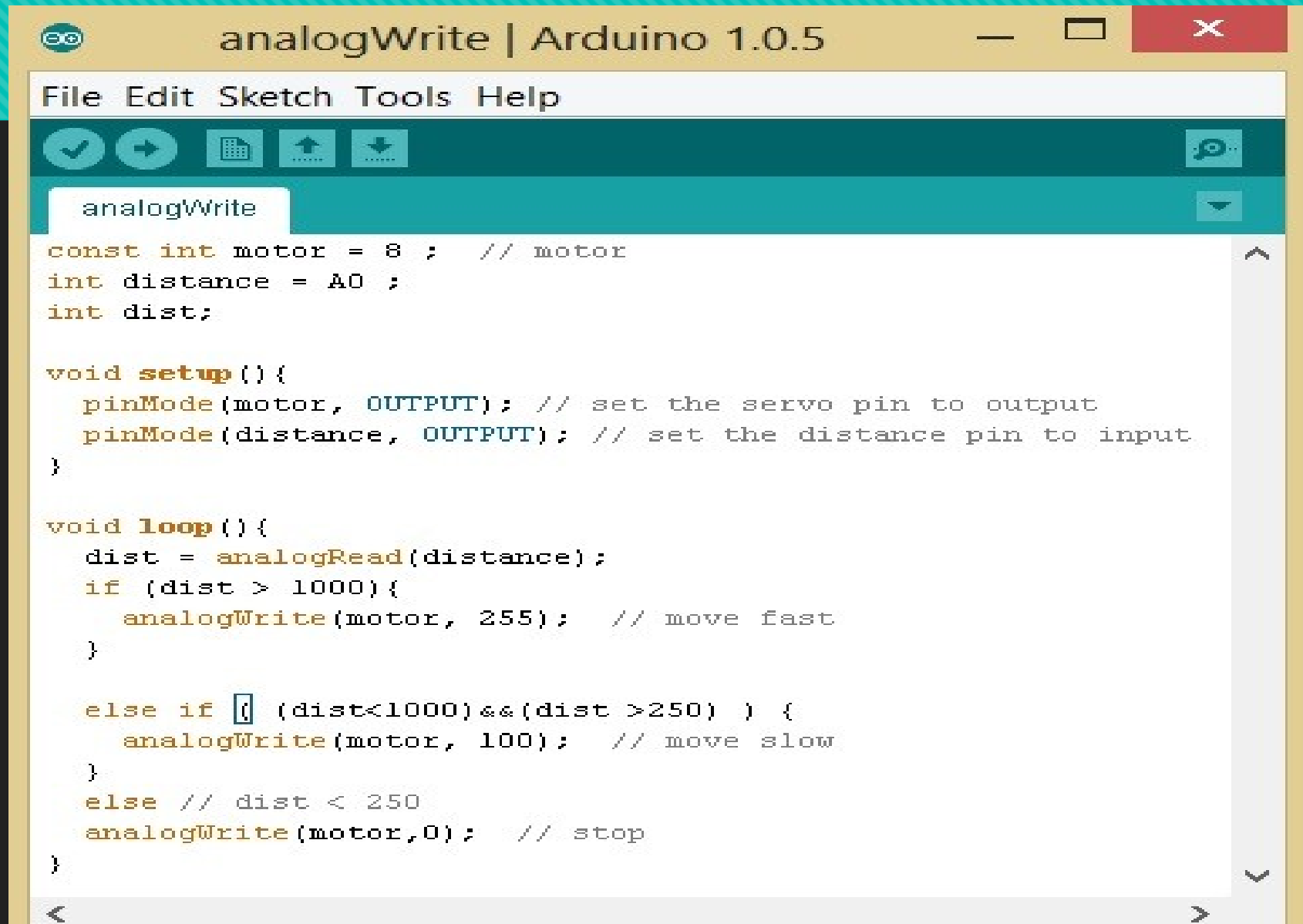
- Analog to Digital Conversion simply means you get an analog input and give a digital output.

- During the conversion, some error is introduced due to approximation of analog value to closest digital value.

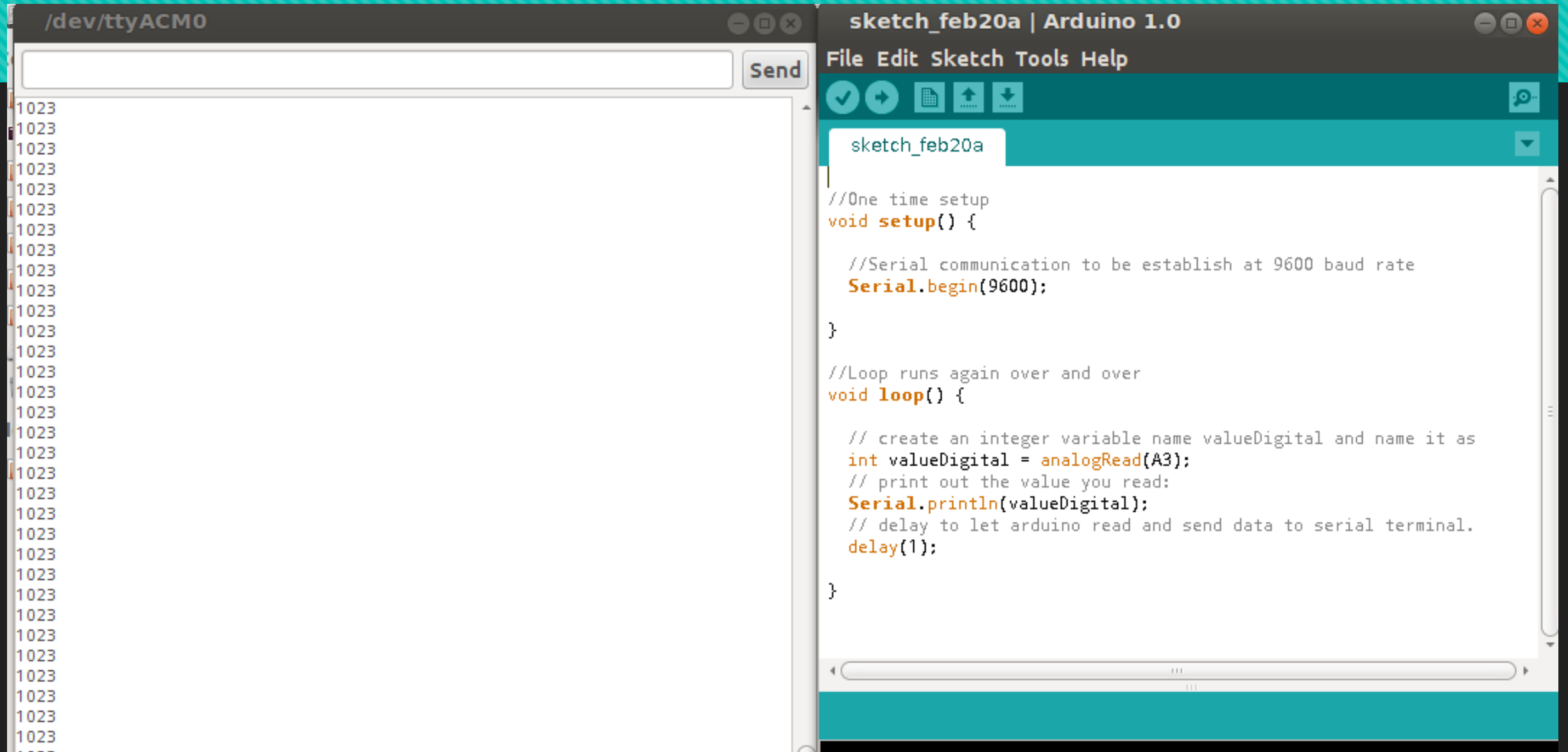
ADC in Arduino – analogRead()



AnalogWrite



Serial Monitor



Program for blinking an LED

LEDblink | Arduino 1.0.5

File Edit Sketch Tools Help

LEDblink

```
int led = 13;

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

void loop() {
  // turn the LED on
  digitalWrite(led, HIGH);
  // wait for a second
  delay(1000);
  // turn the LED off
  digitalWrite(led, LOW);
  // wait for a second
  delay(1000);
}
```

```
1  int led = 13;
2  long t = 0;
3
4  void setup() {
5    pinMode(led, OUTPUT);
6  }
7
8  void loop()
9
10   t = millis();
11   while (millis() < t + 1000) {
12     digitalWrite(led, HIGH);
13   }
14
15   t = millis();
16   while (millis() < t + 1000) {
17     digitalWrite(led, LOW);
18   }
19 }
```

Questions?

Thank You