
ARDUINO

MICROCONTROLLERS – Revision

- 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 .

MAIN TERMS USED IN MCU

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

MAIN TERMS USED 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. ADC channels are shared with PORT A of AVR.
- Motor Driver : It is an external driver circuit which acts as a bridge between IC and motors.

Development Boards

- With the microcontrollers done, lets move on to the next step.
- Microcontrollers give the signals to the actuators. How do we order them to give the signals?
- It is done by the use of development boards.

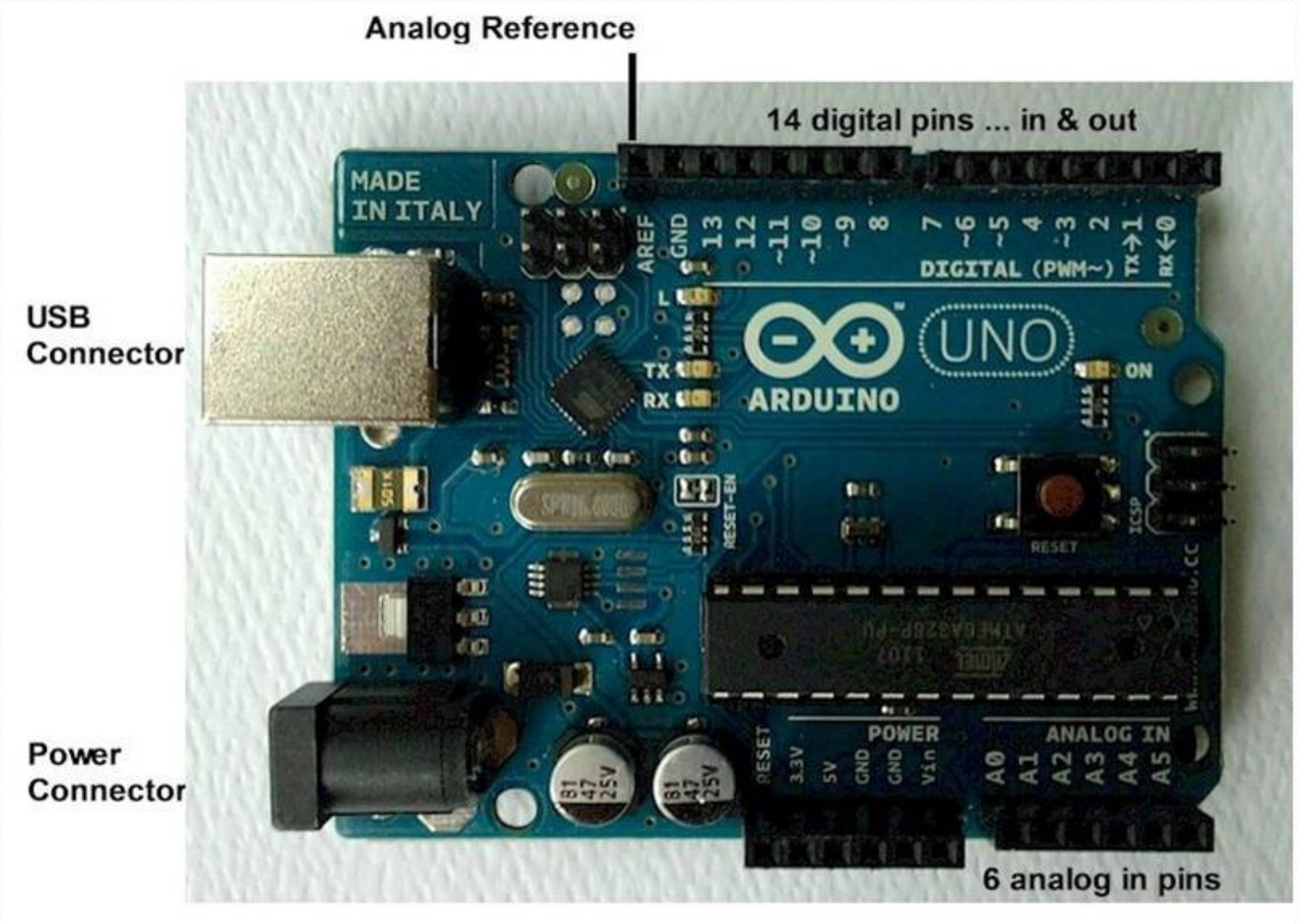
Development Boards

- Printed circuit boards that provide all the circuitry necessary for a useful control task like I/O circuit, clock generator, stored program etc.
- Here we are going to discuss about 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 seperately.
- It also features a USB interface allowing serial communication through a USB device, eliminating the need for a separate AVR programmer.

ARDUINO


Analog Reference
14 digital pins ... in & out
USB Connector
Power Connector
6 analog in pins
MADE IN ITALY
DIGITAL (PWM~)
ARDUINO
UNO
POWER
ANALOG IN

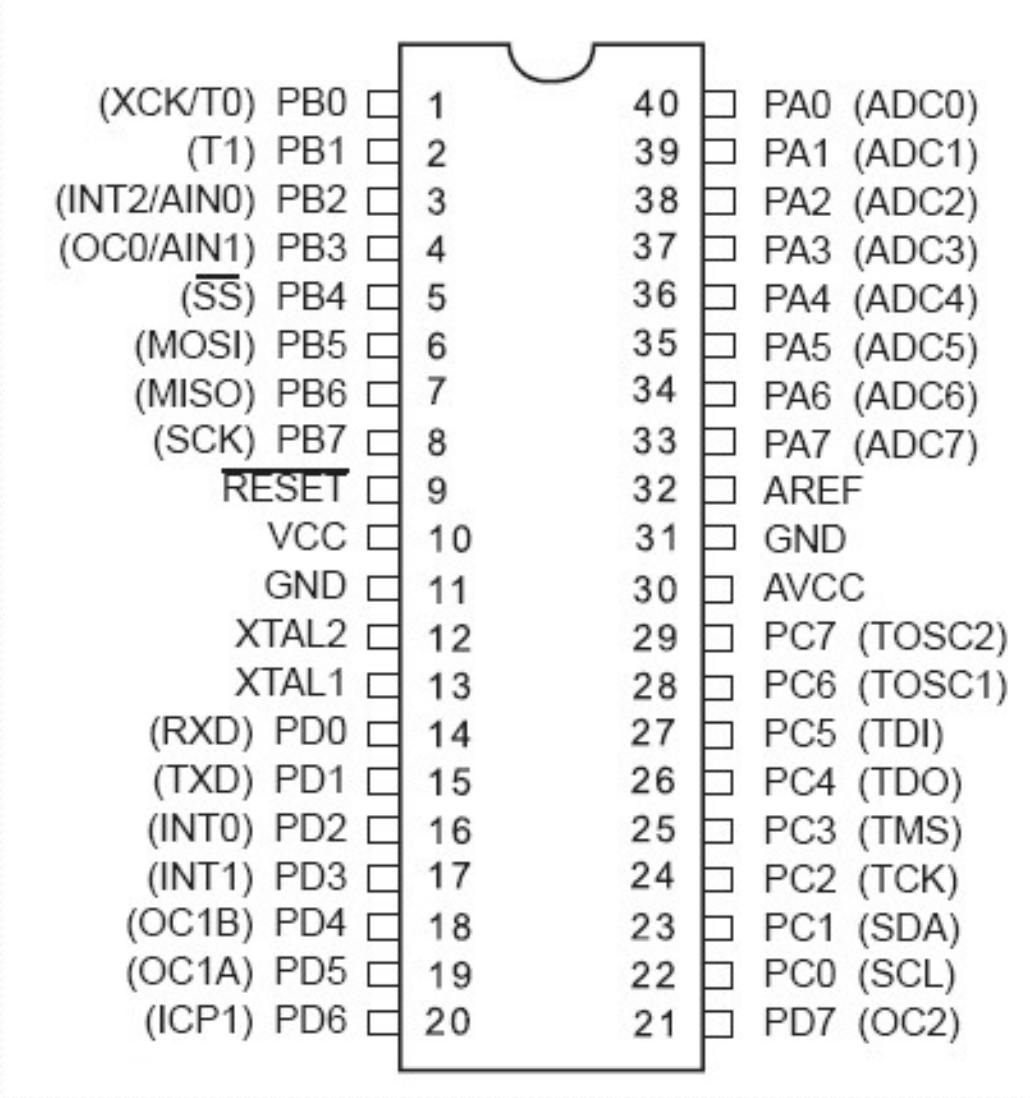
Why ARDUINO ?

- The main reason is its programming environment.
- It 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.

AVR

- AVR is a microcontroller manufactured by Atmel
- We use atmega16A microcontroller.
- 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)

Comparison between Arduino & AVR

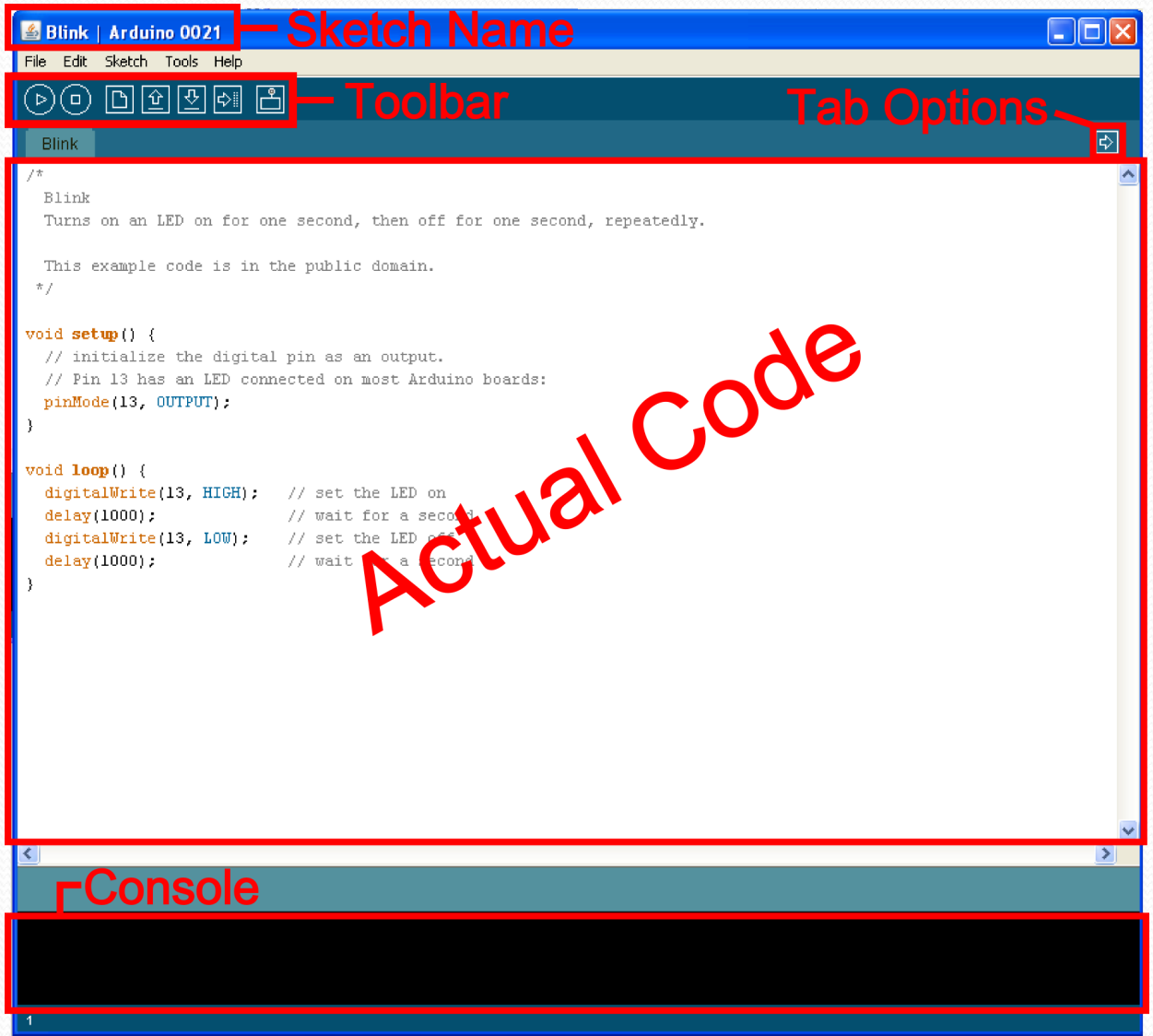
Arduino

- It is easier to use
- It requires little knowledge
- No need to know the inbuild circuits
- It is not as powerful as AVR
- Meant for an amateur

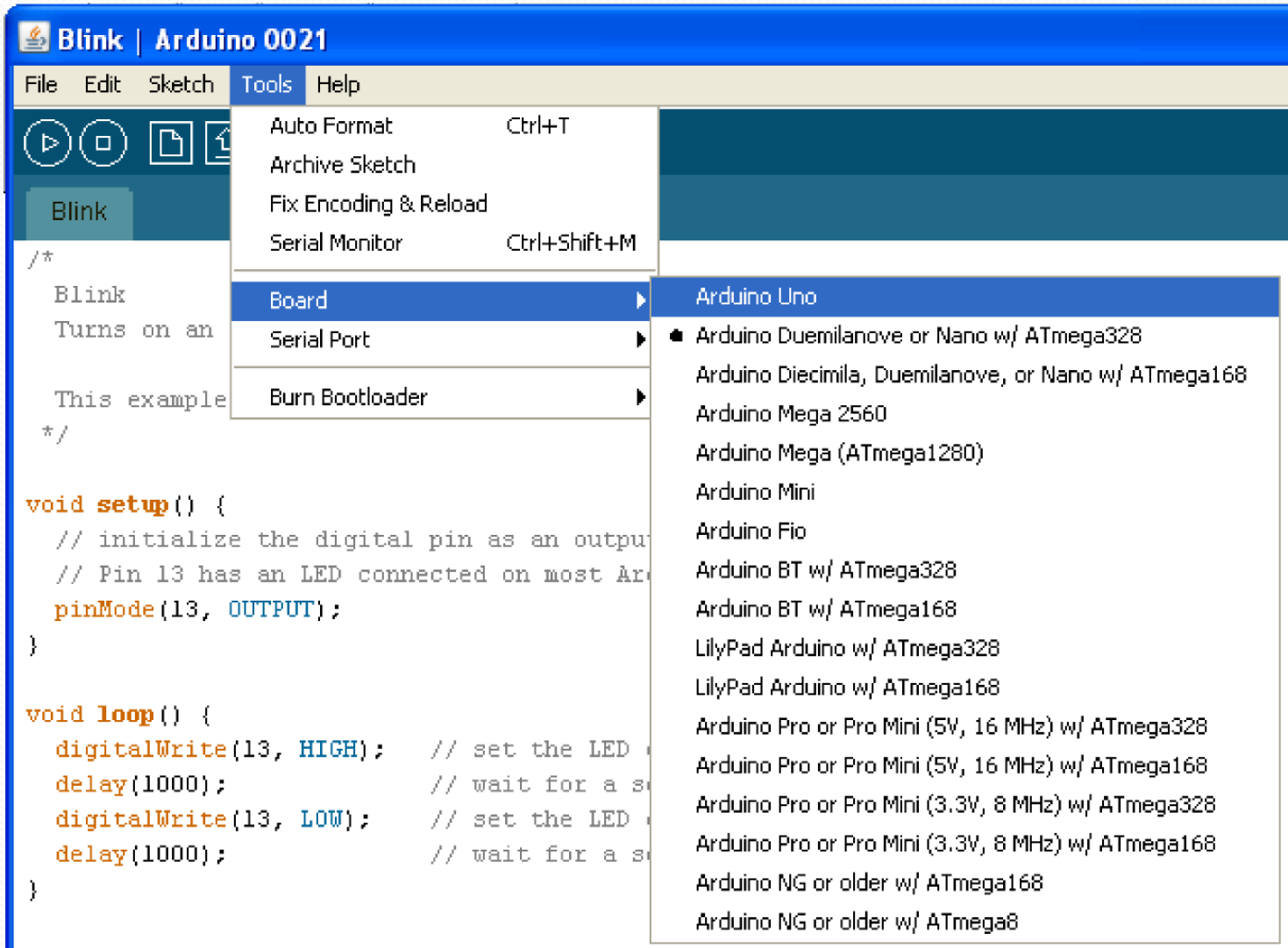
AVR

- It is more difficult to use
- It requires much knowledge
- One needs to know about how the internal circuits are implemented
- More powerful than Arduino
- Meant for a

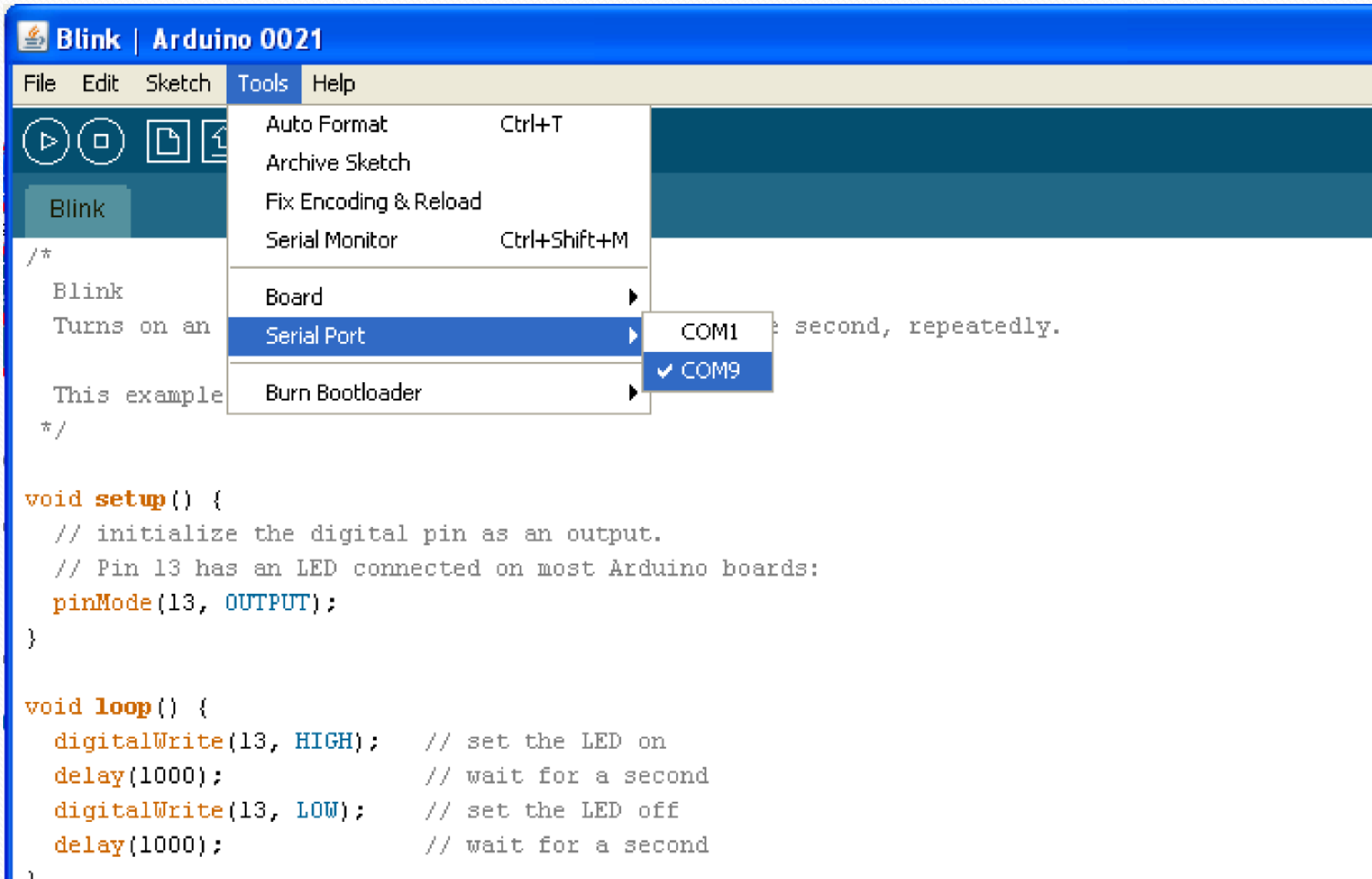
The Arduino Environment



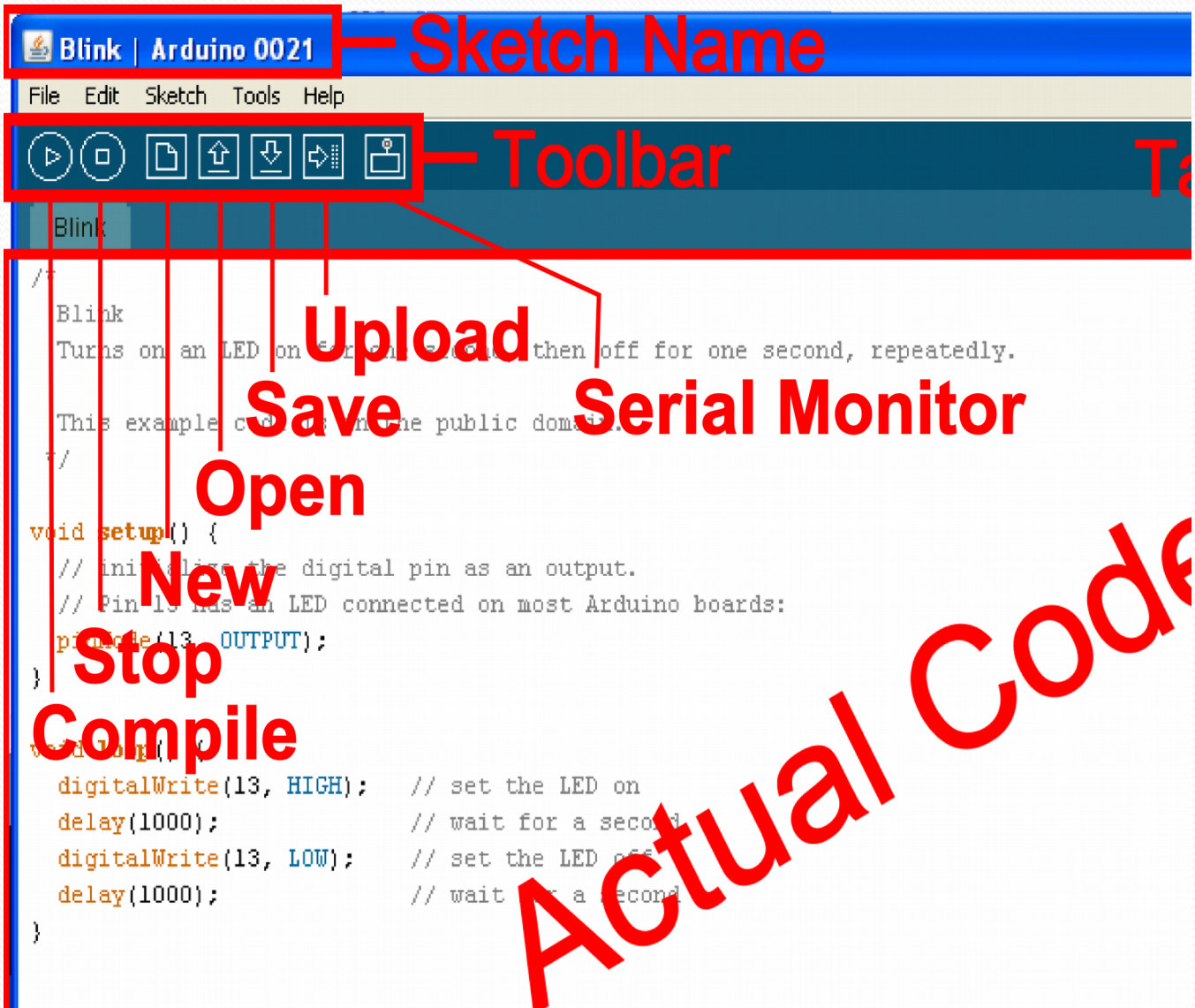
Board Type


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
}

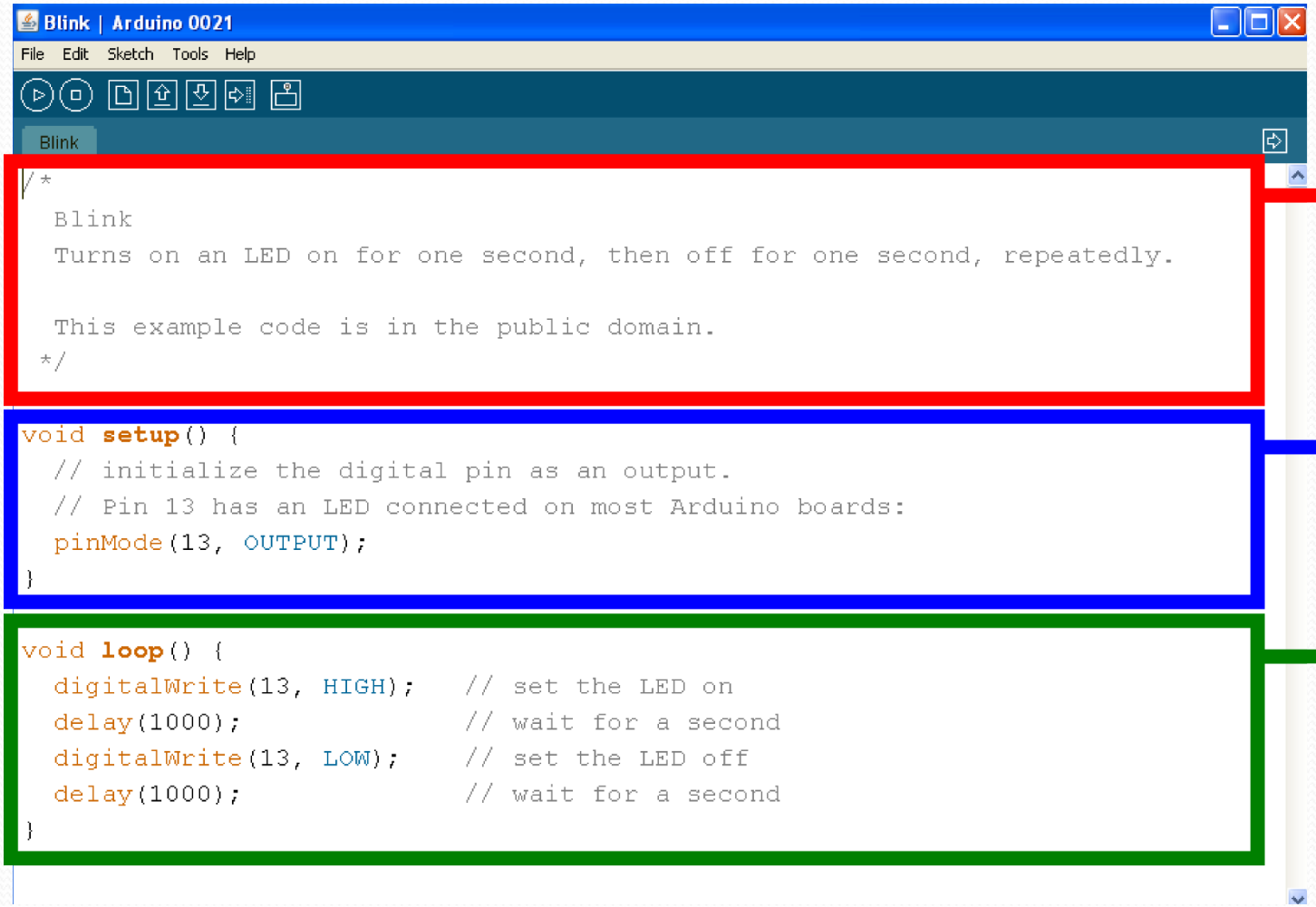
Serial Port / COM Port



The Environment



Parts of the Sketch



Comments / Explaining the game

Setup / Stretching or tying shoes

Loop / Playing the game

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

- The equals sign
- = is used to assign a value
- == is used to compare values

OPERATORS

- And & Or
- && is “and”
- || is “or”

Variables

- Basic variable types:

- Boolean
- Integer
- Character

Declaring Variables

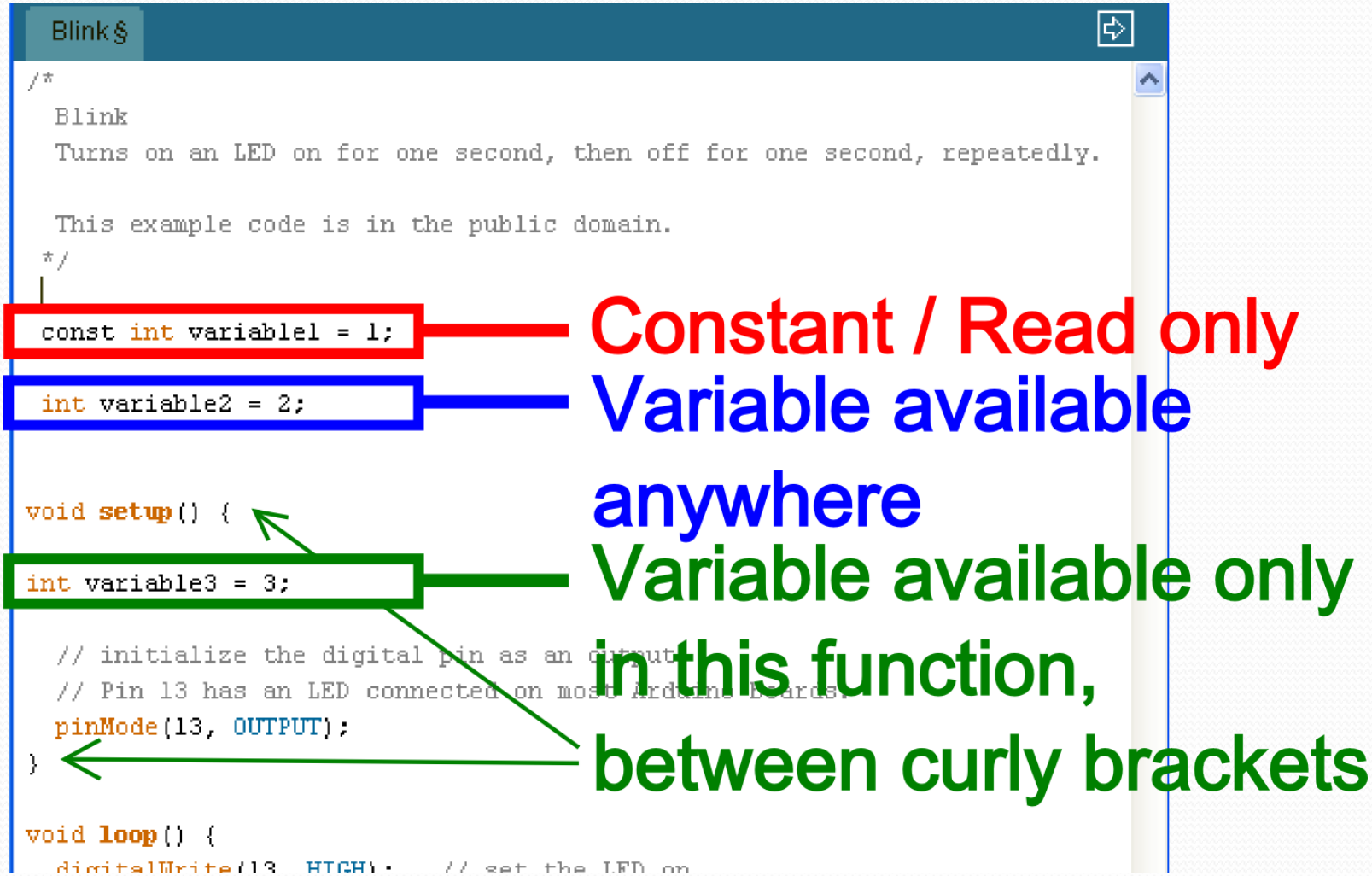
- Boolean: ***boolean variableName;***
- Integer: ***int variableName;***
- Character: ***char variableName;***
- String: ***stringName;***

Assigning Variables

- Boolean: ***variableName = true;***
- or ***variableName = false;***
- Integer: ***variableName = 32767;***
- or ***variableName = -32768;***
- Character: ***variableName =***

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

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

Setup

void setup ( ) {
pinMode (13, OUTPUT); }

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

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

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:
  Serial.begin(9600);
}
```

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

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);
  }
}
```

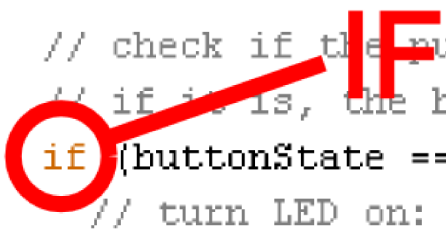
If Statement

If

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);
  }
}
```



Conditional

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);
  }
}
```

Conditional inside parenthesis, uses ==, <=, >= or ! you can also nest using && or ||

Statement(s)

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);
  }
}
```

Action that occurs if conditional is true, inside of curly brackets, can be anything, even more if statements

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

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

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 header

LOOP

void loop ( ) { }

- The “void” in the header is what the function will return (or spit out) when it happens, in this case it returns nothing so it is void

LOOP

void loop ( ) { }

- The “loop” in the header is what the function is called, sometimes you make the name up, sometimes (like loop) the function already has a name

LOOP

void loop ( ) { }

- The “( )” in the header is where you declare any variables that you are “passing” (or sending) the function, the loop function is never “passed” any variables

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
}
```

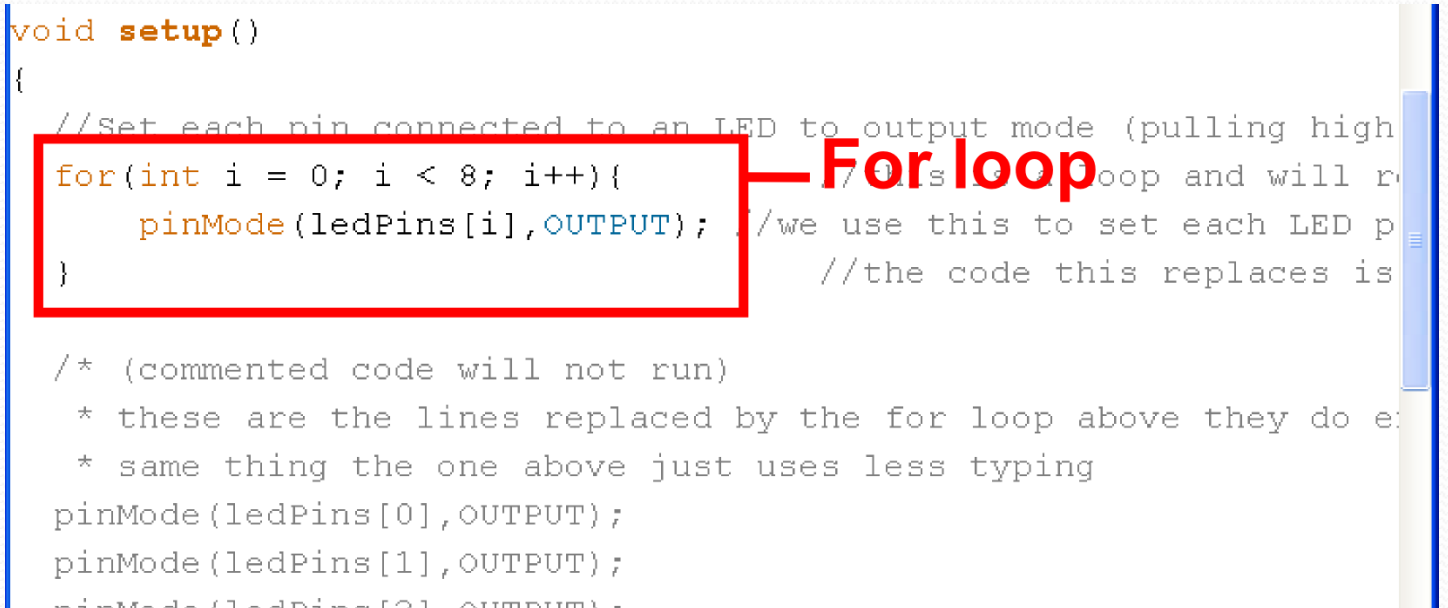
Loop body between curly brackets

Basic Repetition

- For
- While

Basic Repetition

```
for (int count = 0; count<10; count++)
{
//for action code goes here
//this could be anything
}
```



Basic Repetition

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

{

//for action code goes here

}

```
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 header

Basic Repetition

for (int count = 0; count<10; count++)
{
//for action code goes here
}

```
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

Basic Repetition

for (int count = 0; count<10; count++)
{
//for action code goes here
}

```
void setup()
{
  //Set each pin connected to an LED to output mode (pulling high
  for(int i = 0; i < 8; i++){                    //Declare a loop and the
      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);
```

Declare a variable and assign it a value

Basic Repetition

for (int count = 0; count<10; count++)
{
//for action code goes here
}

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

  /* (commented code will not run)
   * these are the lines replaced by the for loop above it does 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);
```

If this conditional is true do the code inside the curly brackets, if it's false the computer exits the for loop

Basic Repetition

for (int count = 0; count<10; count++)
{
//for action code goes here
}

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

  /* (commented code will not run)
   * these are the lines replaced by the for loop after the to 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);
```

Change variable so the computer isn't stuck inside for loop forever

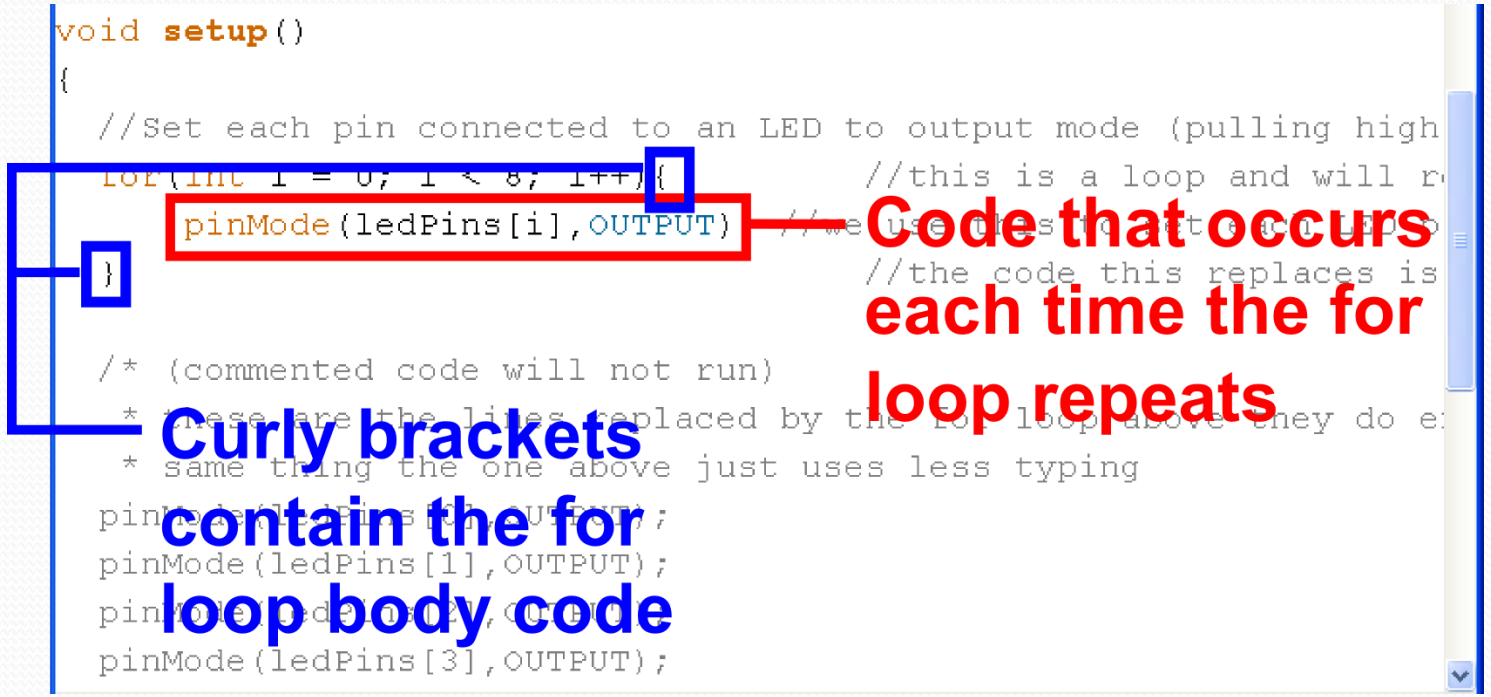
Basic Repetition

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

{

//for action code goes here

}



Basic Repetition

```
while ( count<10 )
{
//while action code goes here
//should include a way to change count
//variable so the computer is not stuck
//inside the while loop forever
}
```

Basic Repetition

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

Digital Read

```
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
}
```

Digital Write

void setup ( ) {

digitalWrite (12, HIGH); }

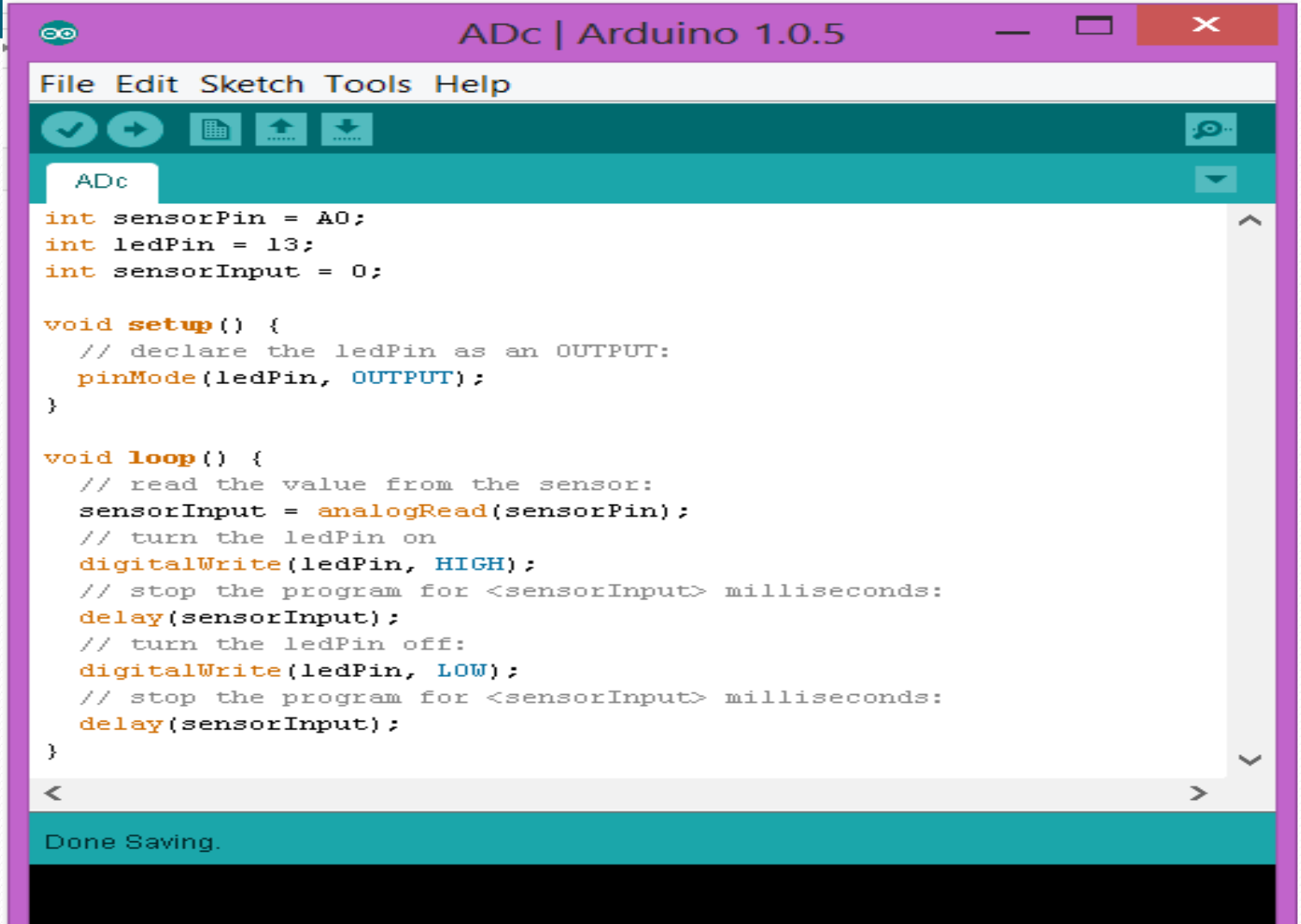
```
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
  Serial.begin(9600);
  digitalWrite(12, HIGH);
}
```

- digitalWrite is used to give output values to DIGITAL pins, i.e., HIGH voltage or LOW voltage.

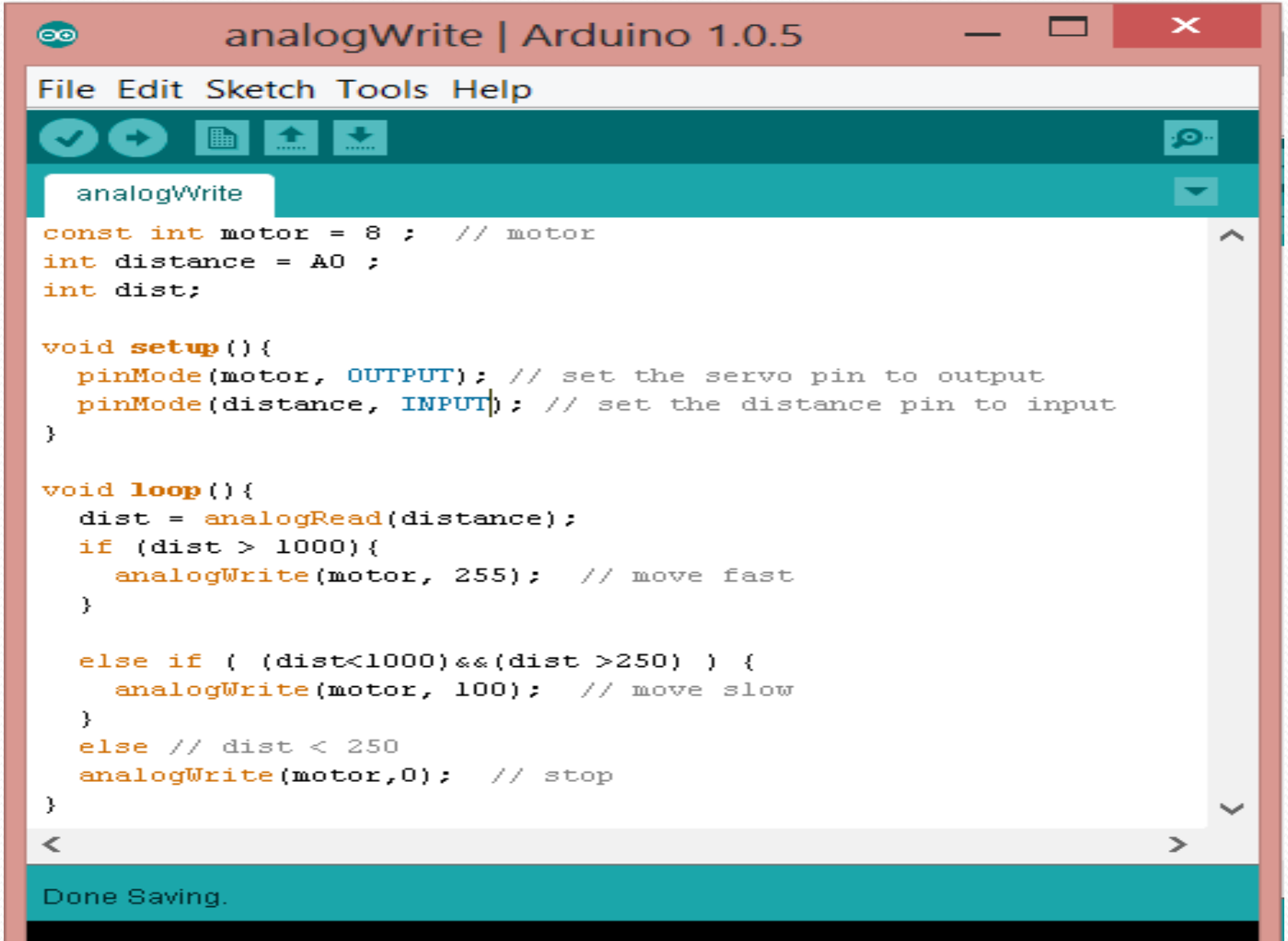
Analog to Digital Conversion in Arduino

- 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.

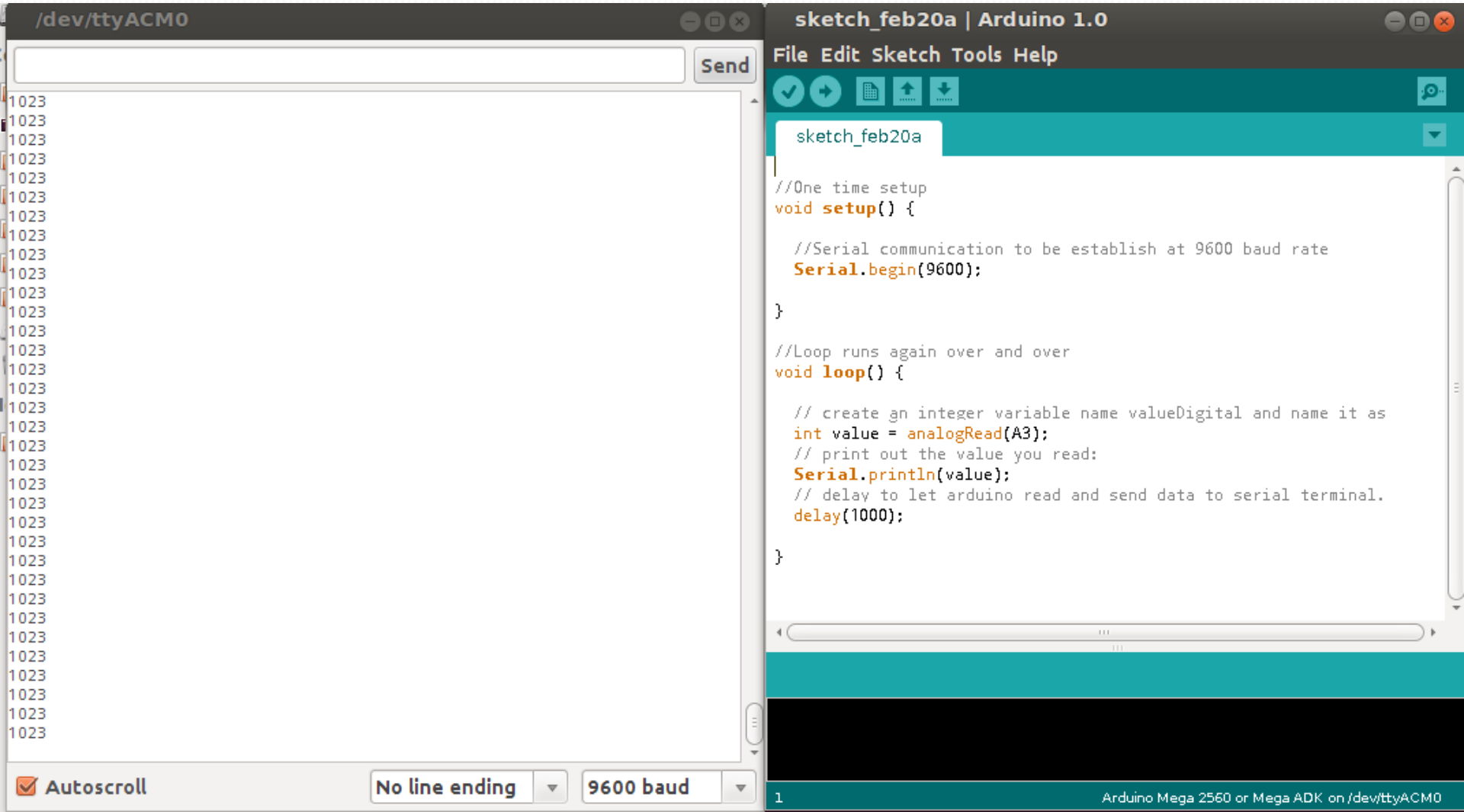
Analog to Digital Conversion in Arduino



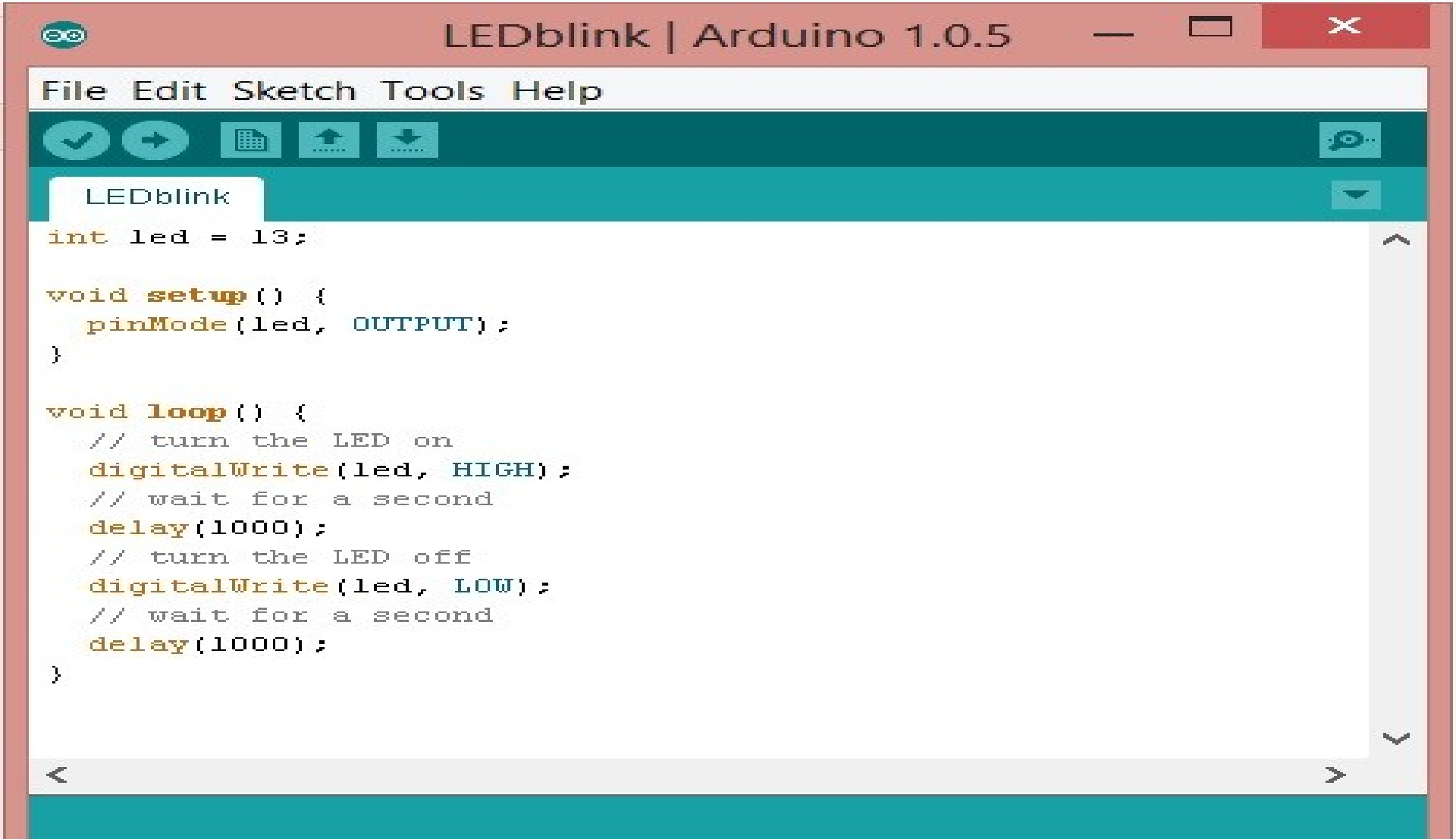
Analog Write



Showing readings on Serial Monitor



Program for blinking an LED



Questions ?

Thanks!