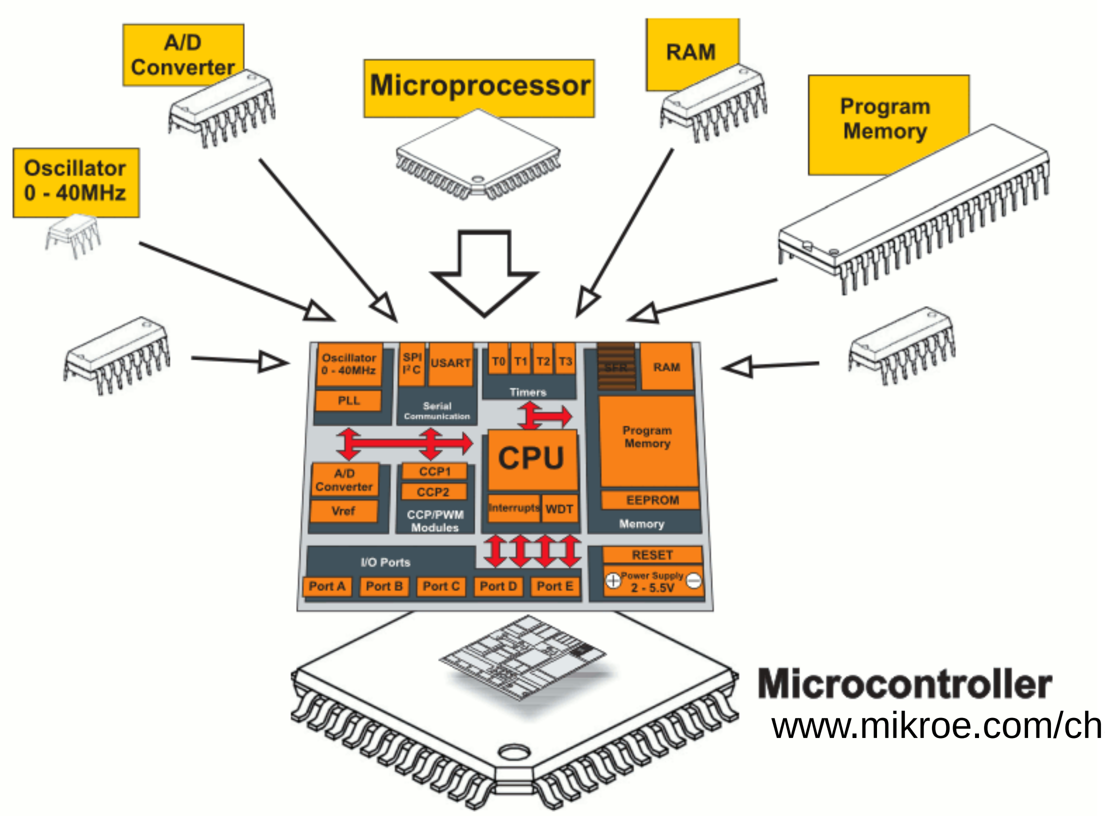
Arduino

What is a Microcontroller

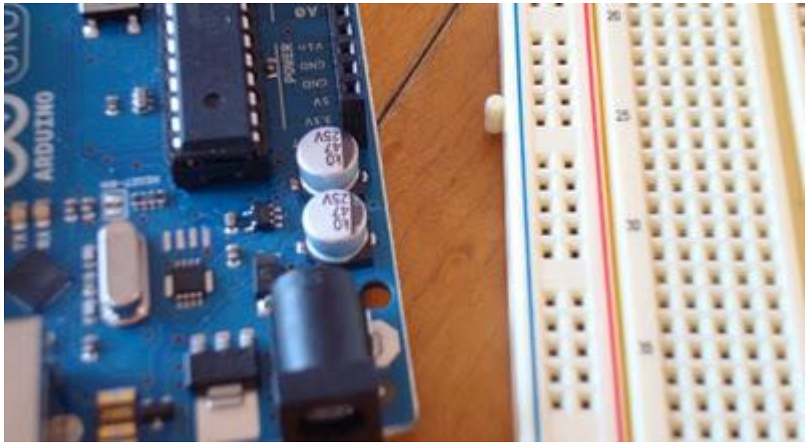


www.mikroe.com/chapters/view/1

Fig. 0-1 Microcontroller versus Microprocessor

- A small computer on a single chip
  - containing a processor, memory, and input/output
- Typically "**embedded**" inside some device that they control
- A microcontroller is often small and low cost
- [Examples]

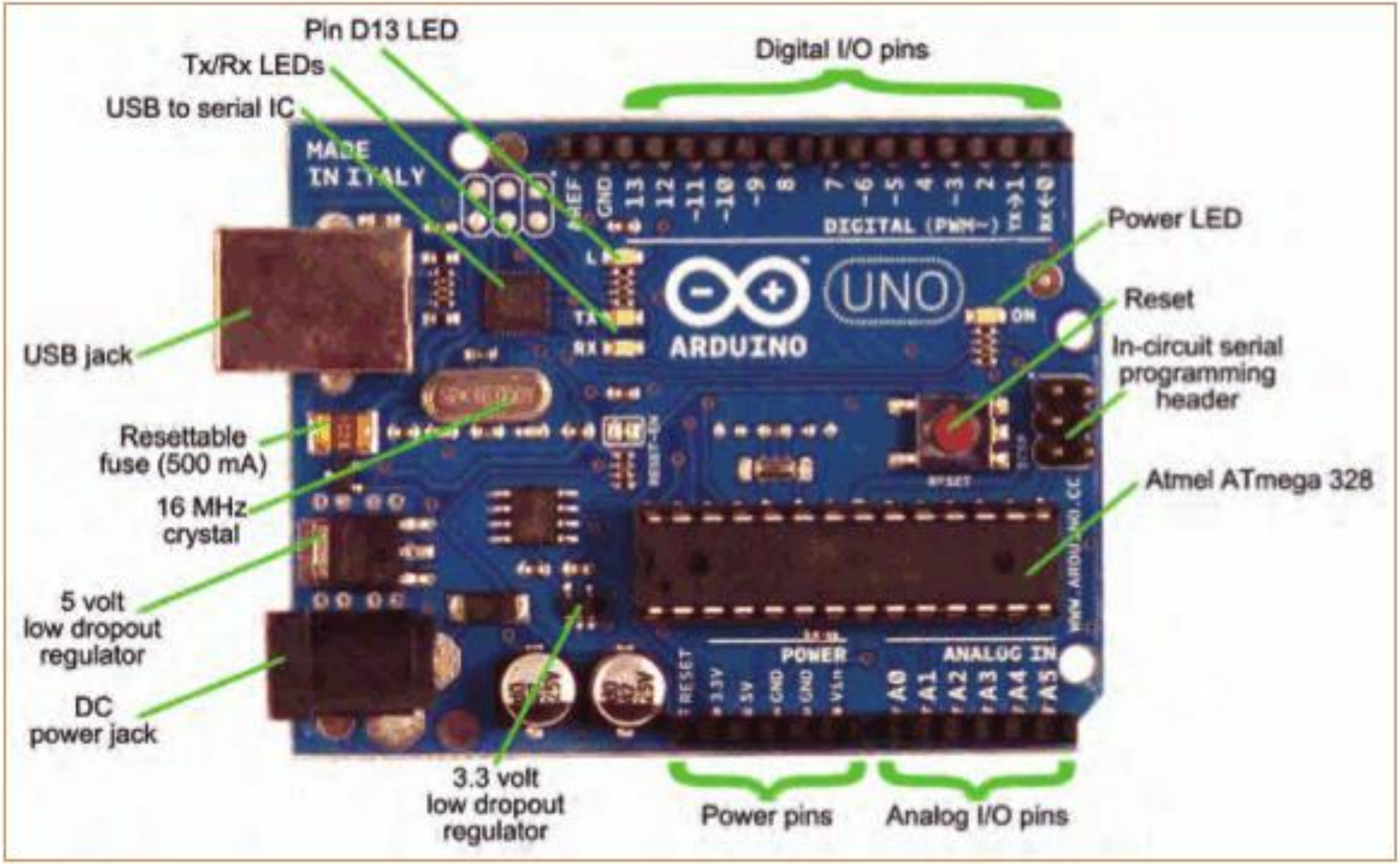
What is a Development Board



- A printed circuit board designed to facilitate work with a particular microcontroller.

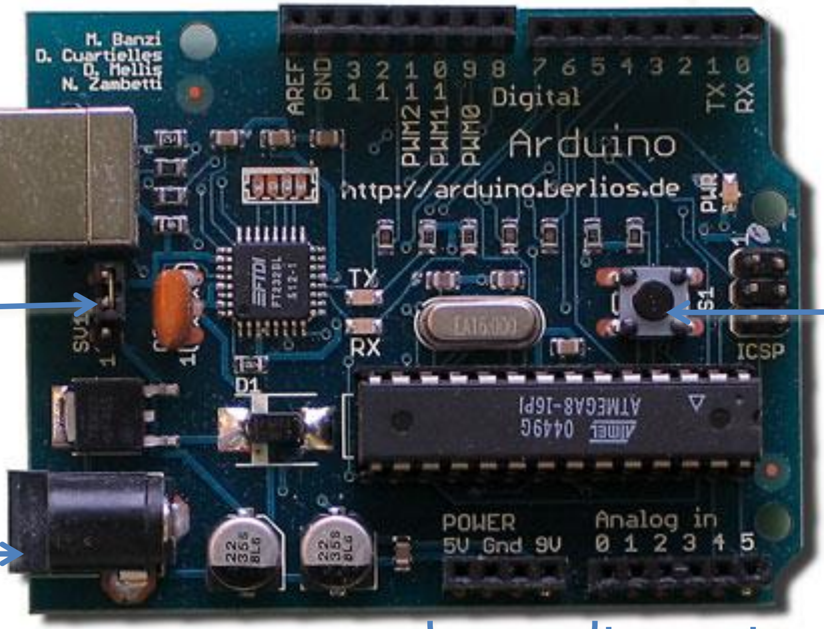
- Typical components include:
  - power circuit
  - programming interface
  - basic input; usually buttons and LEDs
  - I/O pins

The Arduino Development Board



Making-robots-with-arduino.pdf

The Arduino Development Board



Digital I/O (2 – 13)

Serial Transfer (0 -1)

USB (Data & Power)

Power Source Jumper

Reset

Alternate Power (9V)

5V / 9V / GND (x2)

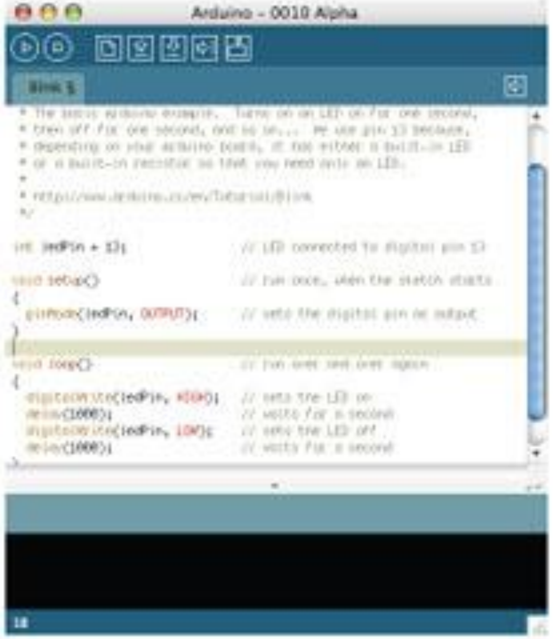
Analog Input (0 – 5)

What is the Arduino

The word "Arduino" can mean 3 things

A physical piece of hardware

A programming environment



```
 * The basic Arduino example.  Turns on an LED on for one second,
 * then off for one second, and so on...  We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;                // LED connected to digital pin 13

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()                     // run over and over again
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
}
```

A community & philosophy

Arduino playground

:: About the Arduino Playground ::

[illegible]

:: RoadMap: What Needs to be Done? ::

[illegible]

todbot.com/blog/bionicarduino

Getting Started

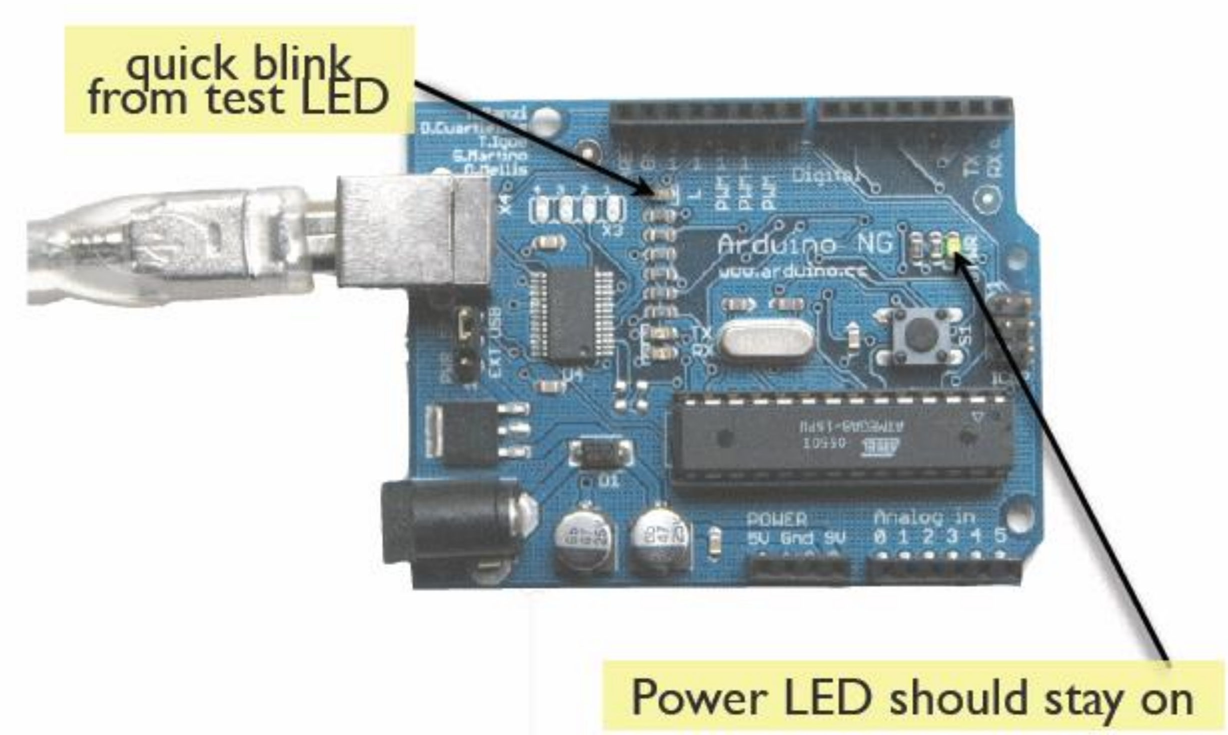
- Check out: http://arduino.cc/en/Guide/HomePage

1. **Download & install the Arduino environment (IDE)**

Software | Arduino

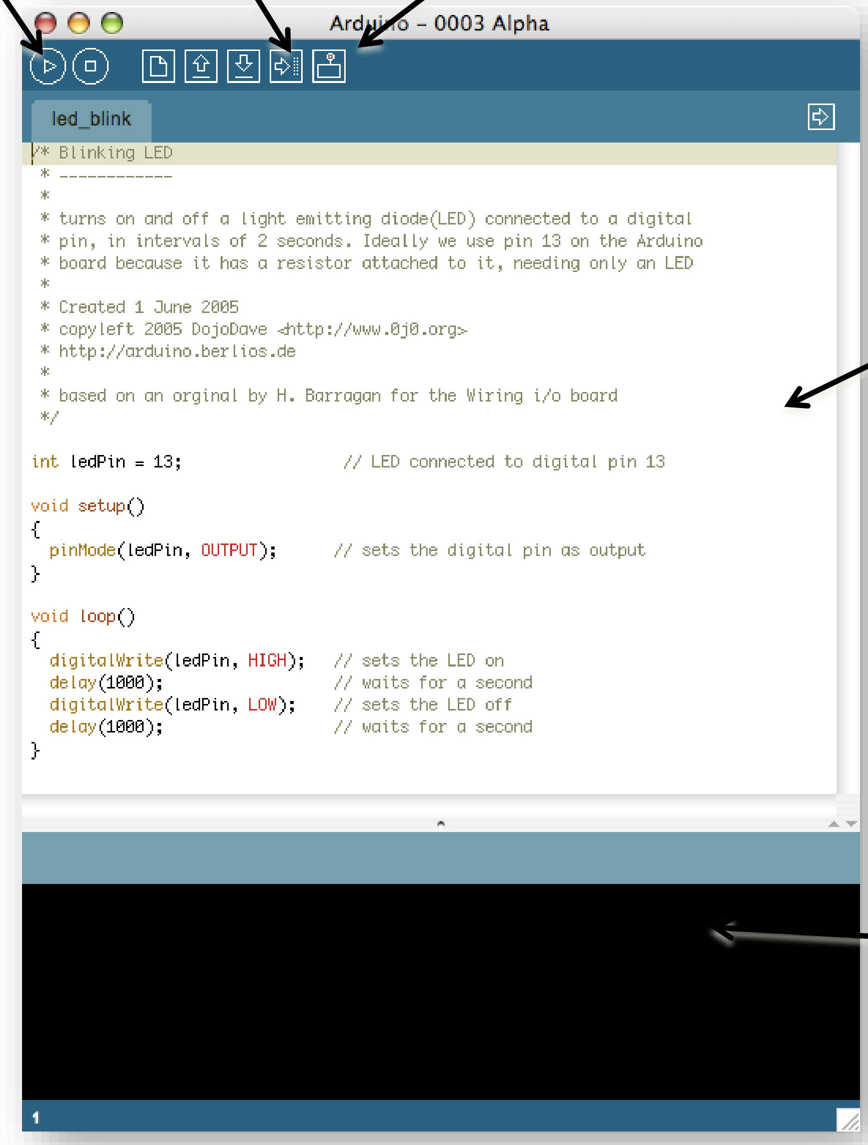
1. **Connect the board to your computer via the UBS cable**
2. **If needed, install the drivers (not needed in lab)**
3. **Launch the Arduino IDE**
4. **Select your board**
5. **Select your serial port**
6. **Open the blink example**
7. **Upload the program**

Try It: Connect the USB Cable



todbot.com/blog/bionicarduino

Arduino IDE Window



```
/* Blinking LED
 * ------------
 *
 * turns on and off a light emitting diode(LED) connected to a digital
 * pin, in intervals of 2 seconds. Ideally we use pin 13 on the Arduino
 * board because it has a resistor attached to it, needing only an LED
 *
 * Created 1 June 2005
 * copyleft 2005 DojoDave <http://www.0j0.org>
 * http://arduino.berlios.de
 *
 * based on an orginal by H. Barragan for the Wiring i/o board
 */

int ledPin = 13;                 // LED connected to digital pin 13

void setup()
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
}
```

Compile

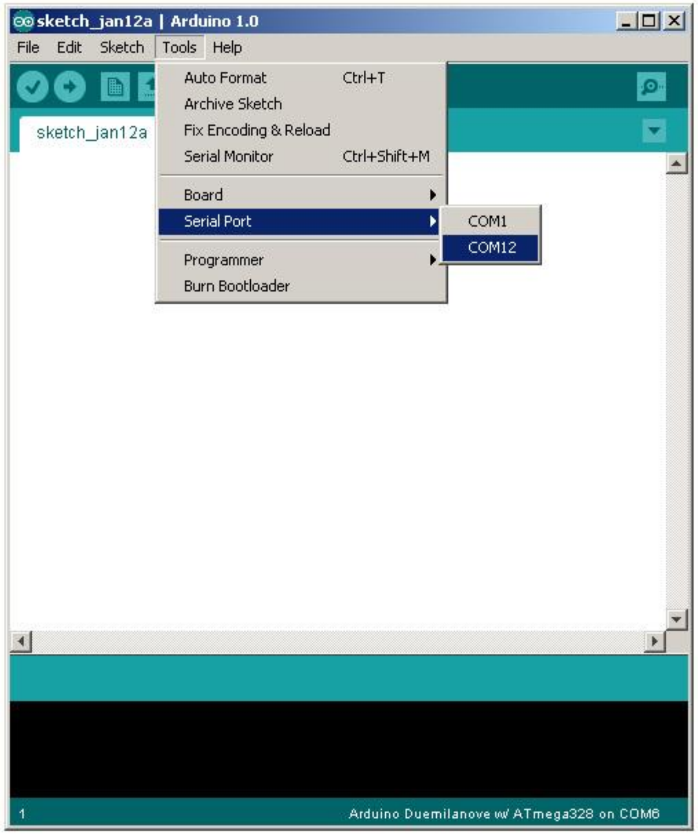
Upload to controller

Serial Monitor

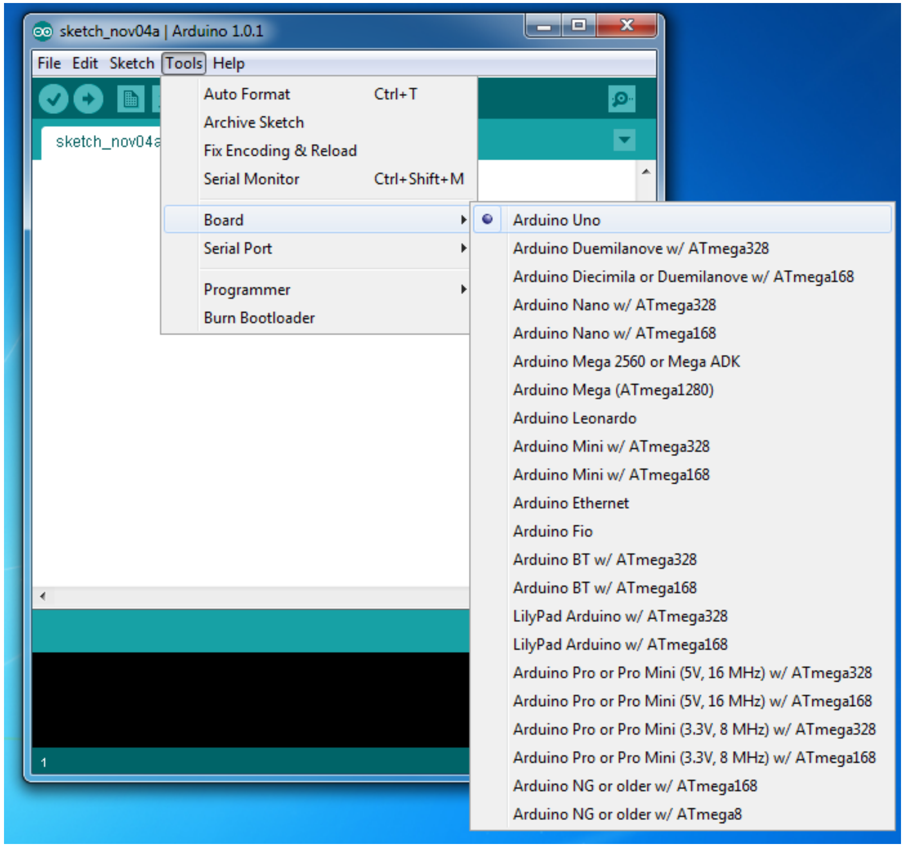
Code Editor

Text Output (Serial Data)

Select Serial Port and Board



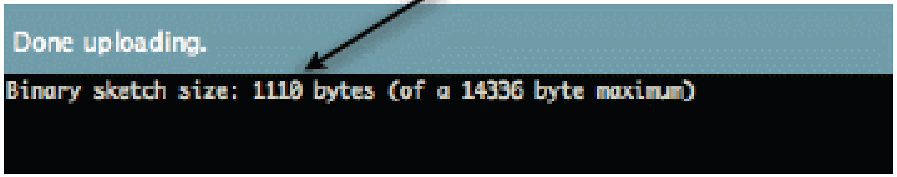
Arduino Duemilanove w/ ATmega328 on COM6


sketch_nov04a | Arduino 1.0.1
File
Edit
Sketch
Tools
Help
sketch_nov04a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader
● 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 Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
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
1

Status Messages

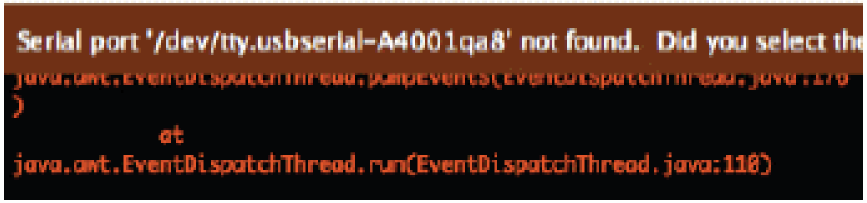
Size depends on
complexity of your sketch

Uploading worked



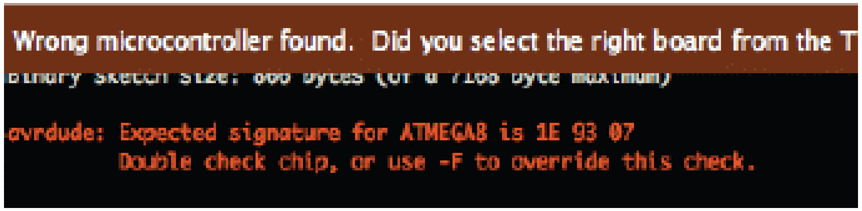
```
Done uploading.
Binary sketch size: 1110 bytes (of a 14336 byte maximum)
```

Wrong serial port selected



```
Serial port '/dev/tty.usbserial-A4001qa8' not found.  Did you select the
java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:170
)
            at
java.awt.EventDispatchThread.run(EventDispatchThread.java:110)
```

Wrong board selected



```
Wrong microcontroller found.  Did you select the right board from the T
Binary sketch size: 666 bytes (of a 7168 byte maximum)

avrdude: Expected signature for ATMEGA8 is 1E 93 07
         Double check chip, or use -F to override this check.
```

nerdy cryptic error messages

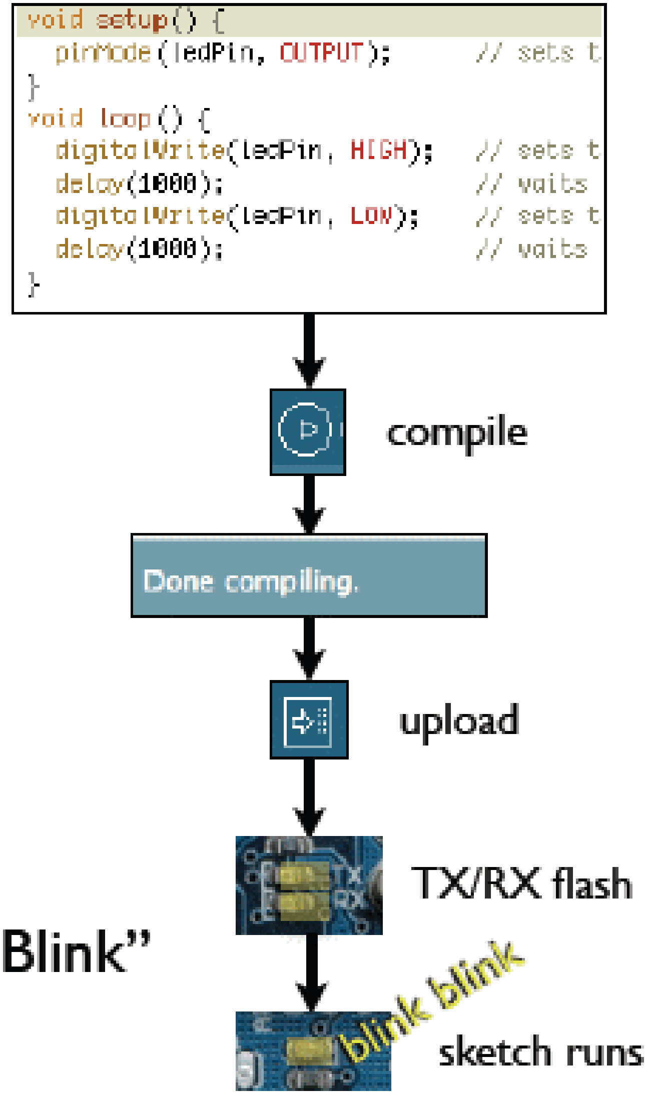
todbot.com/blog/bionicarduino

Using Arduino

- Write your sketch
- Press Compile button (to check for errors)
- Press Upload button to program Arduino board with your sketch

Try it out with the "Blink" sketch!

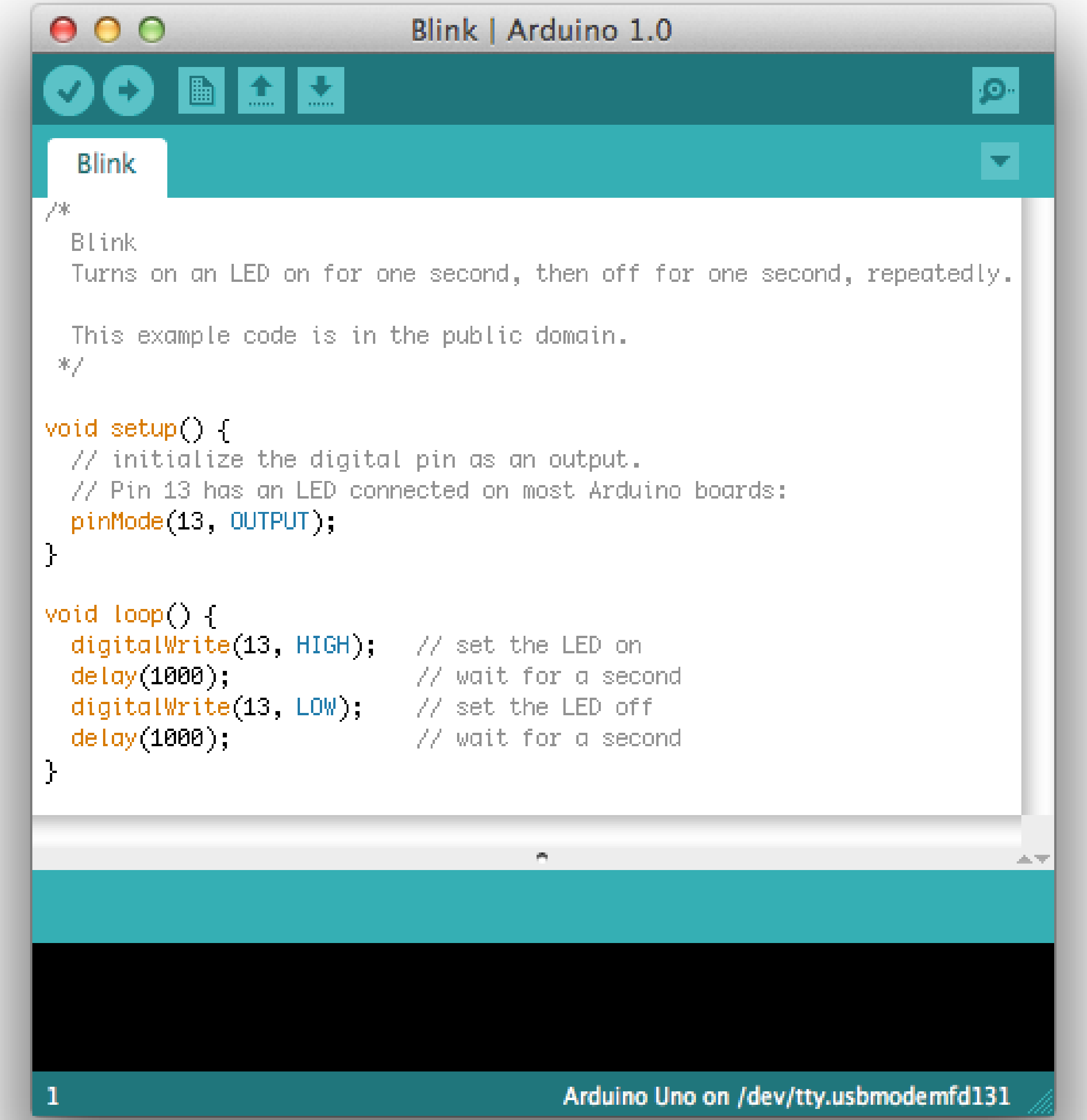
Load "File/Sketchbook/Examples/Digital/Blink"



```
void setup() {
  pinMode(ledPin, OUTPUT);      // sets t
}
void loop() {
  digitalWrite(ledPin, HIGH);   // sets t
  delay(1000);                  // waits
  digitalWrite(ledPin, LOW);    // sets t
  delay(1000);                  // waits
}
```

Done compiling.

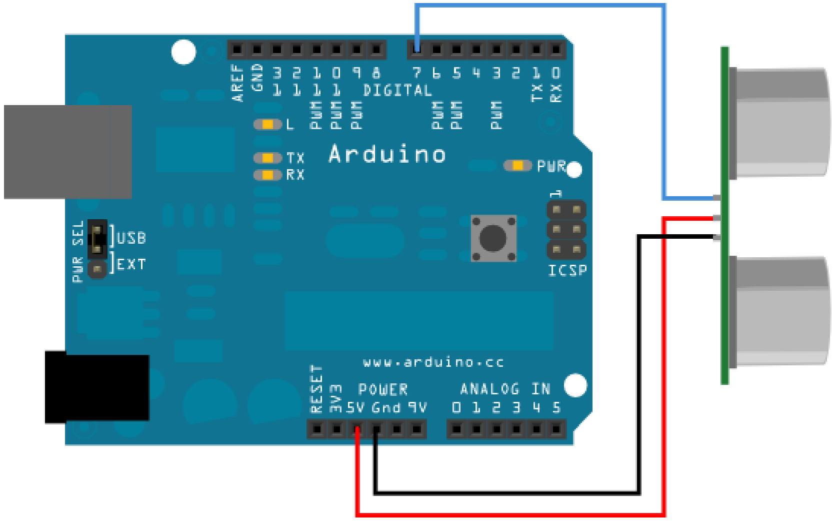
Our First Program



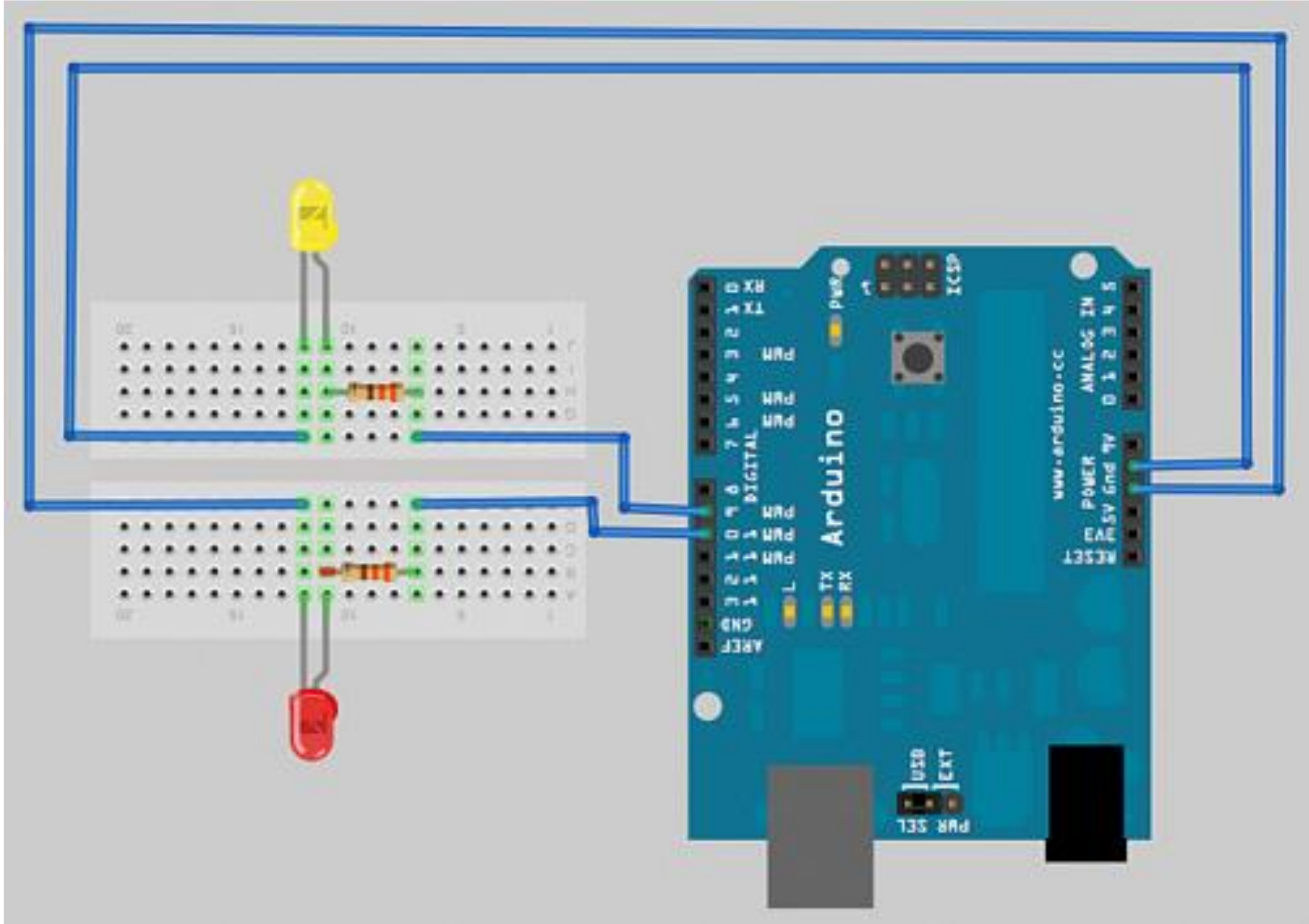
```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  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
}
```

Arduino Code Basics

- Commands and other information are sent to LED’s, motors and from sensors through **digital** and **analog** input & output pins

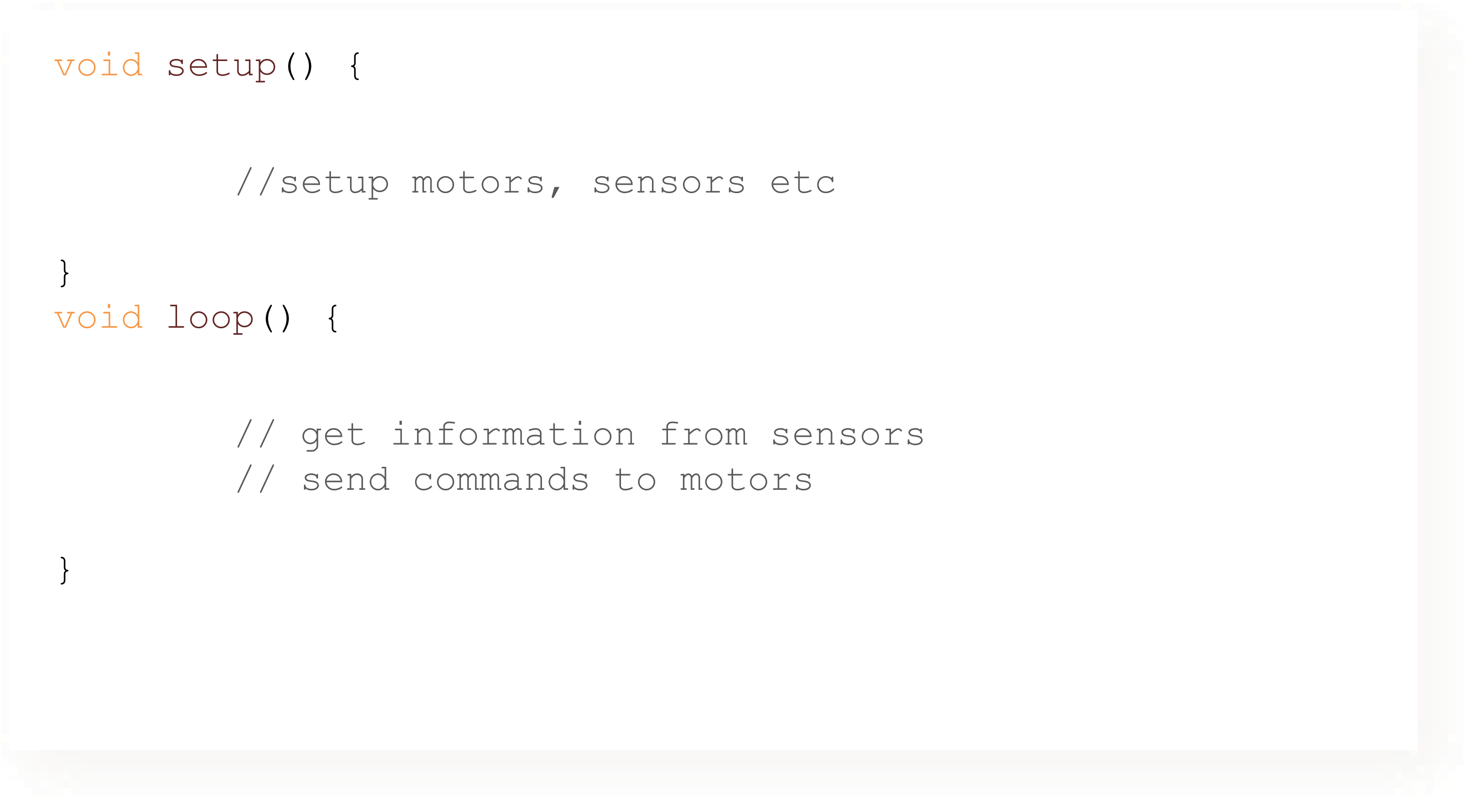


Setup - Adding an LED



Arduino Code Basics

Arduino programs run on two basic sections:

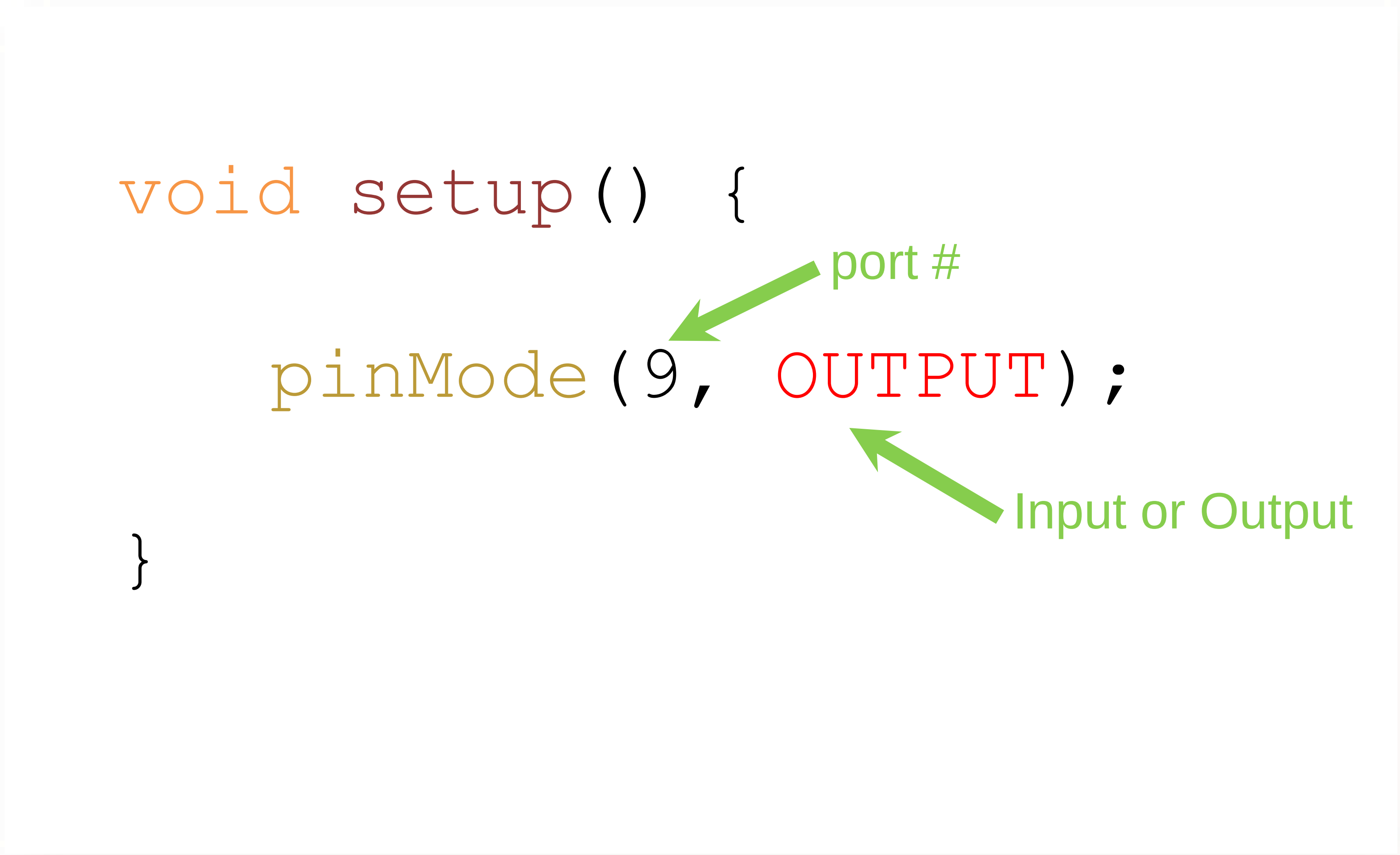


```
void setup() {


        //setup motors, sensors etc

}
void loop() {


        // get information from sensors
        // send commands to motors

}
```

SETUP

- The setup section is used for assigning input and outputs (Examples: motors, LED’s, sensors etc) to ports on the Arduino
- It also specifies whether the device is OUTPUT or INPUT
- To do this we use the command “pinMode”

SETUP



```
void setup() {

    pinMode(9, OUTPUT);

}
```

port #

Input or Output

http://www.arduino.cc/en/Reference/HomePage

LOOP

```
void loop() {

    digitalWrite(9, HIGH);
    delay(1000);
    digitalWrite(9, LOW);
    delay(1000);

}
```

Port # from setup

Turn the LED on
or off

Wait for 1 second
or 1000 milliseconds

TASK 1

- Using 3 LED’s (red, yellow and green) build a traffic light that
    - Illuminates the green LED for 5 seconds
    - Illuminates the yellow LED for 2 seconds
    - Illuminates the red LED for 5 seconds
    - repeats the sequence

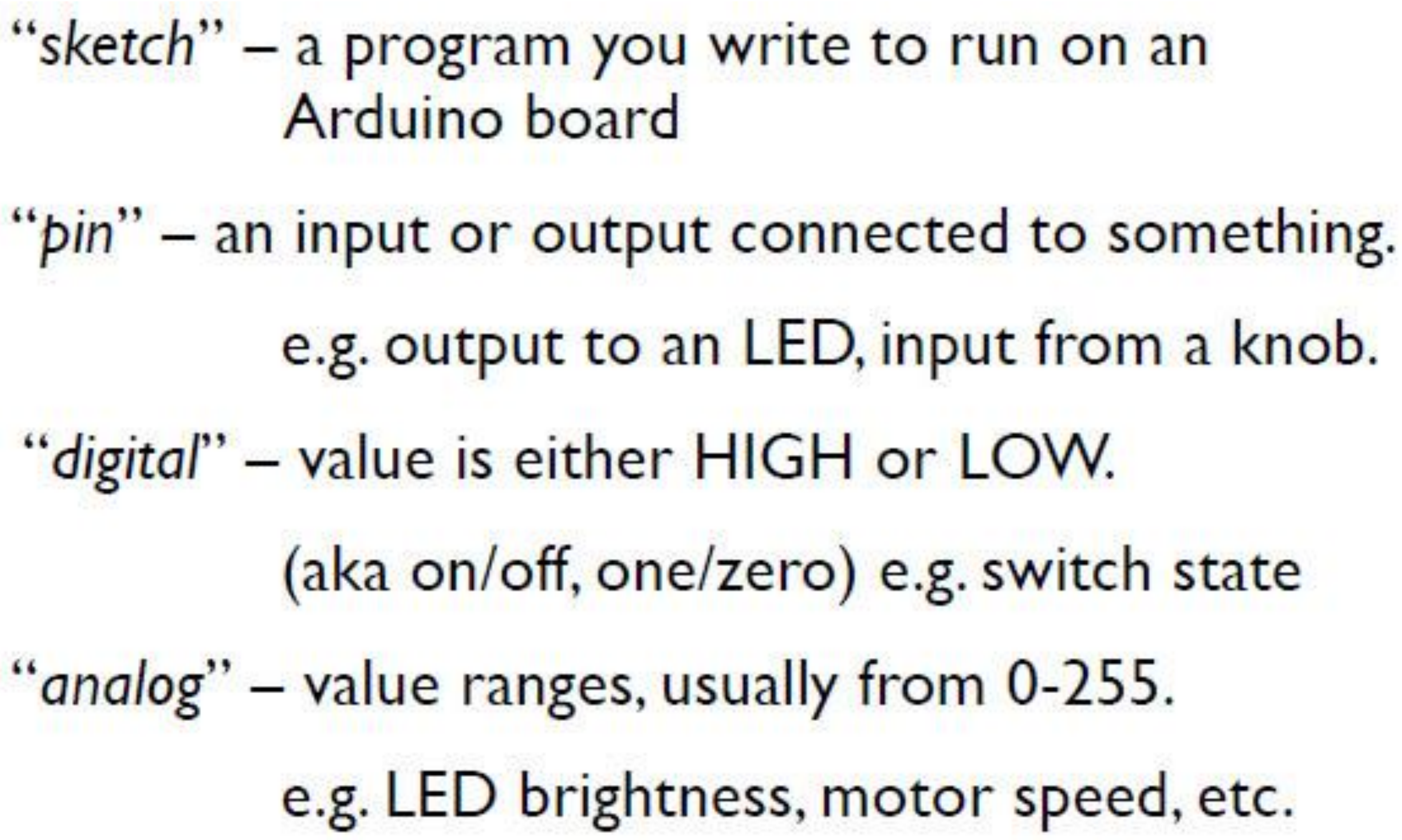
- Note that after each illumination period the LED is turned off!

TASK 2

- Modify Task 1 to have an advanced green (blinking green LED) for 3 seconds before illuminating the green LED for 5 seconds

NB: 1 Second = 1000 millisecond

Terminology



"*sketch*" – a program you write to run on an Arduino board

"*pin*" – an input or output connected to something.

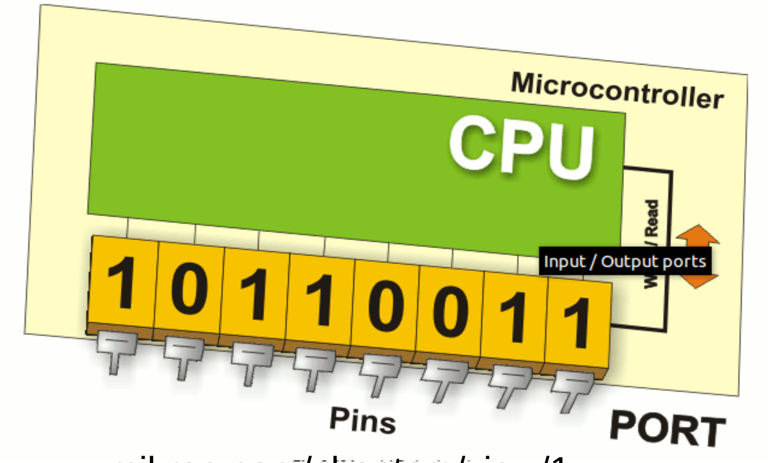
e.g. output to an LED, input from a knob.

"*digital*" – value is either HIGH or LOW.

(aka on/off, one/zero) e.g. switch state

"*analog*" – value ranges, usually from 0-255.

e.g. LED brightness, motor speed, etc.



www.mikroe.com/chapters/view/1

Digital I/O

`pinMode(pin, mode)`

Sets pin to either `INPUT` or `OUTPUT`

`digitalRead(pin)`

Reads `HIGH` or `LOW` from a pin

`digitalWrite(pin, value)`

Writes `HIGH` or `LOW` to a pin

Electronic stuff

Output pins can provide 40 mA of current

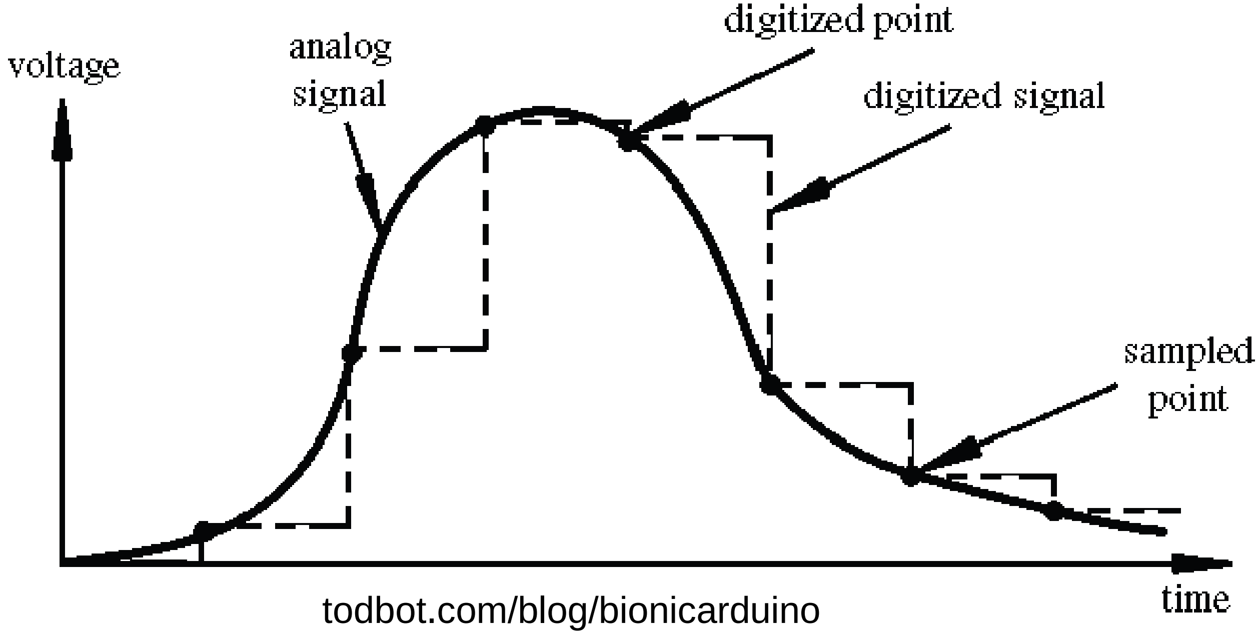
Writing `HIGH` to an input pin installs a 20KΩ pullup

Arduino Timing

- `delay(`*`ms`*`)`
  - Pauses for a few milliseconds
- `delayMicroseconds(`*`us`*`)`
  - Pauses for a few microseconds
- More commands: arduino.cc/en/Reference/HomePage

Digital? Analog?

- Digital has two values: **on** and **off**
- Analog has many (infinite) values
- Computers don’t really do analog, they ***quantize***
- Remember the 6 analog input pins---here’s how they work

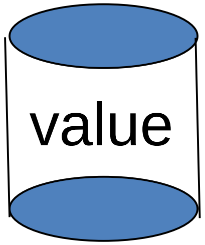


todbot.com/blog/bionicarduino

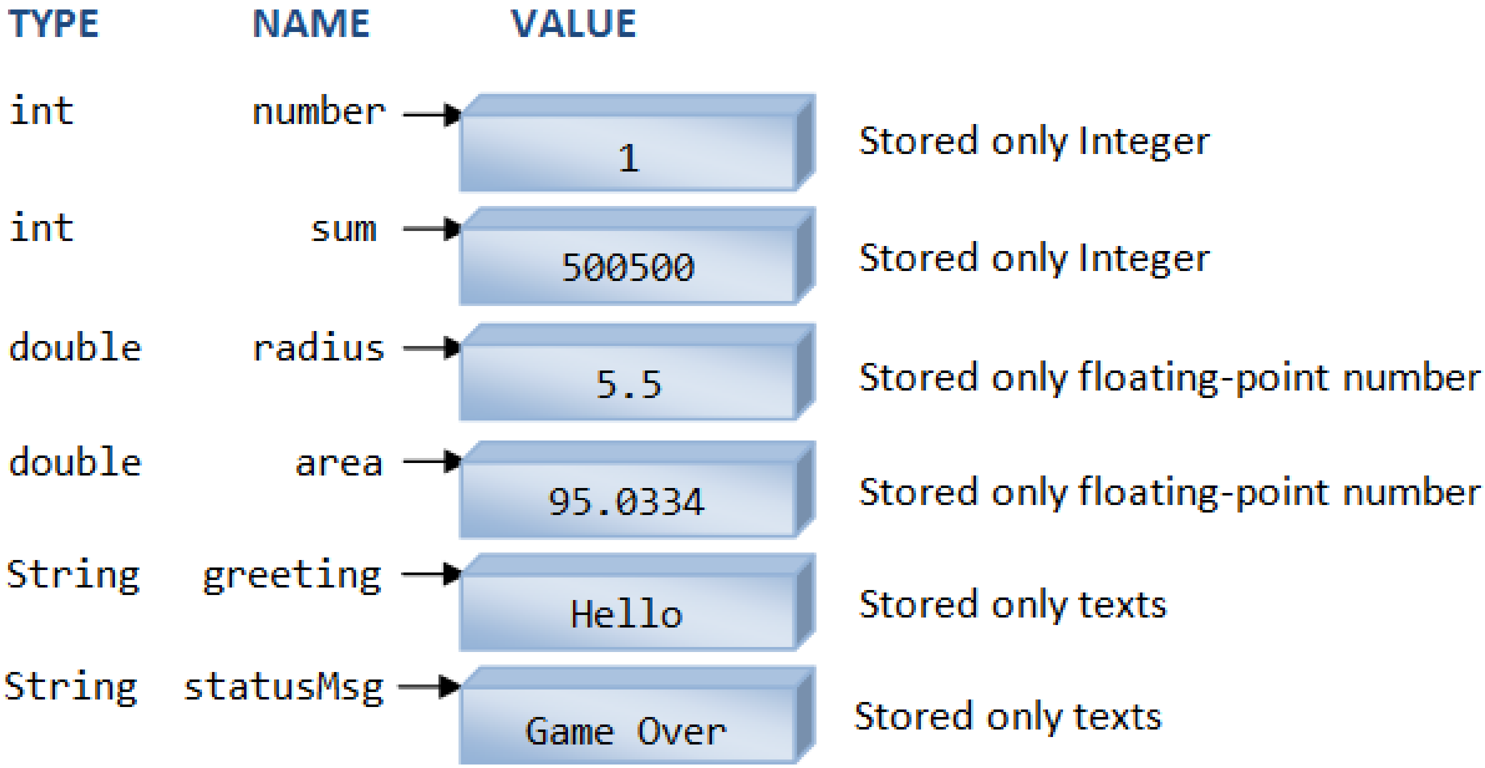
Arduino programming

Variables

- A variable is like “bucket”
- It holds numbers or other values temporarily



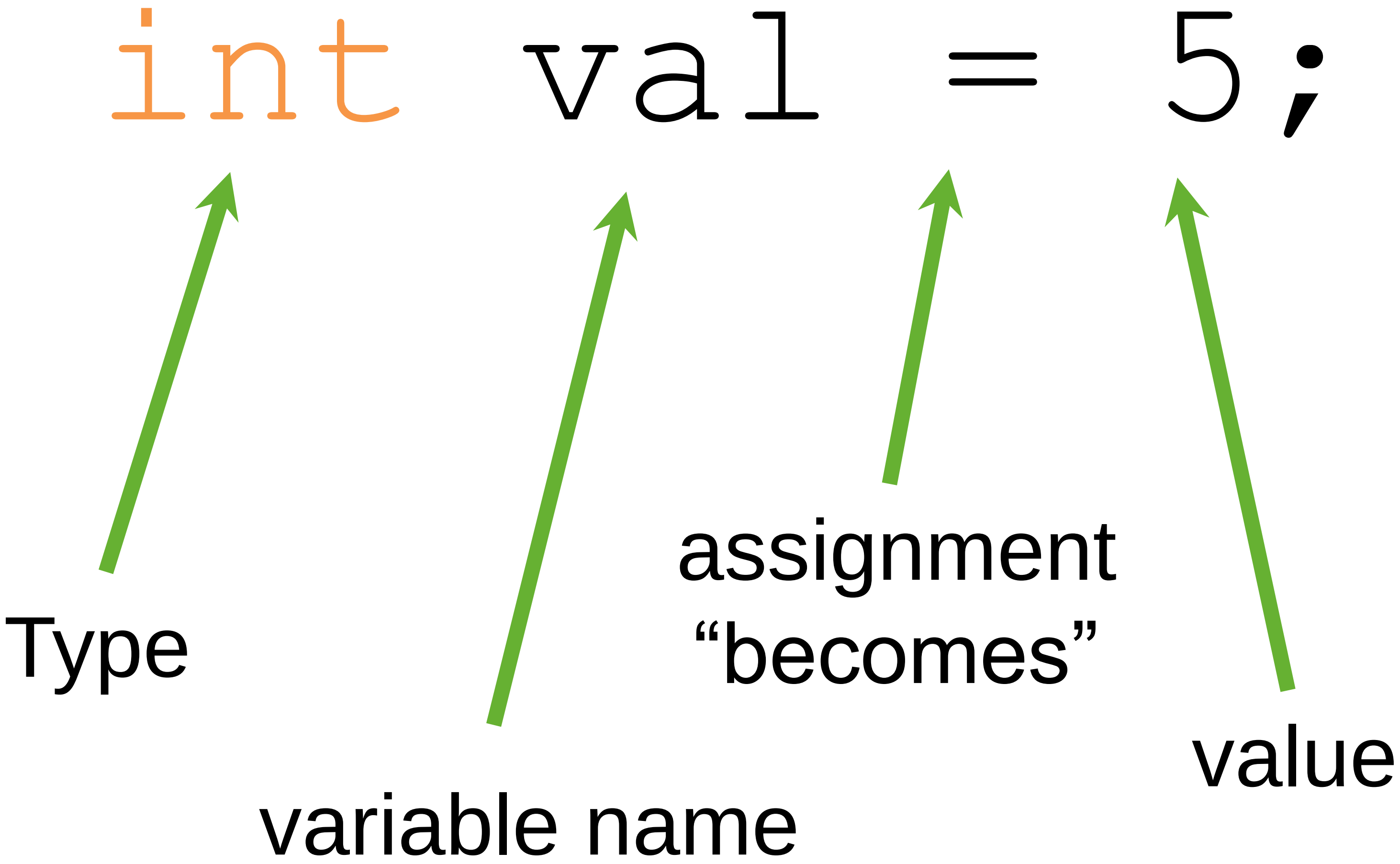
Variables



TYPE	NAME	VALUE	
int	number	1	Stored only Integer
int	sum	500500	Stored only Integer
double	radius	5.5	Stored only floating-point number
double	area	95.0334	Stored only floating-point number
String	greeting	Hello	Stored only texts
String	statusMsg	Game Over	Stored only texts

A *variable* has a ***name***, stores a ***value*** of the declared ***type***.

DECLARING A VARIABLE



```
int val = 5;
```

Type

variable name

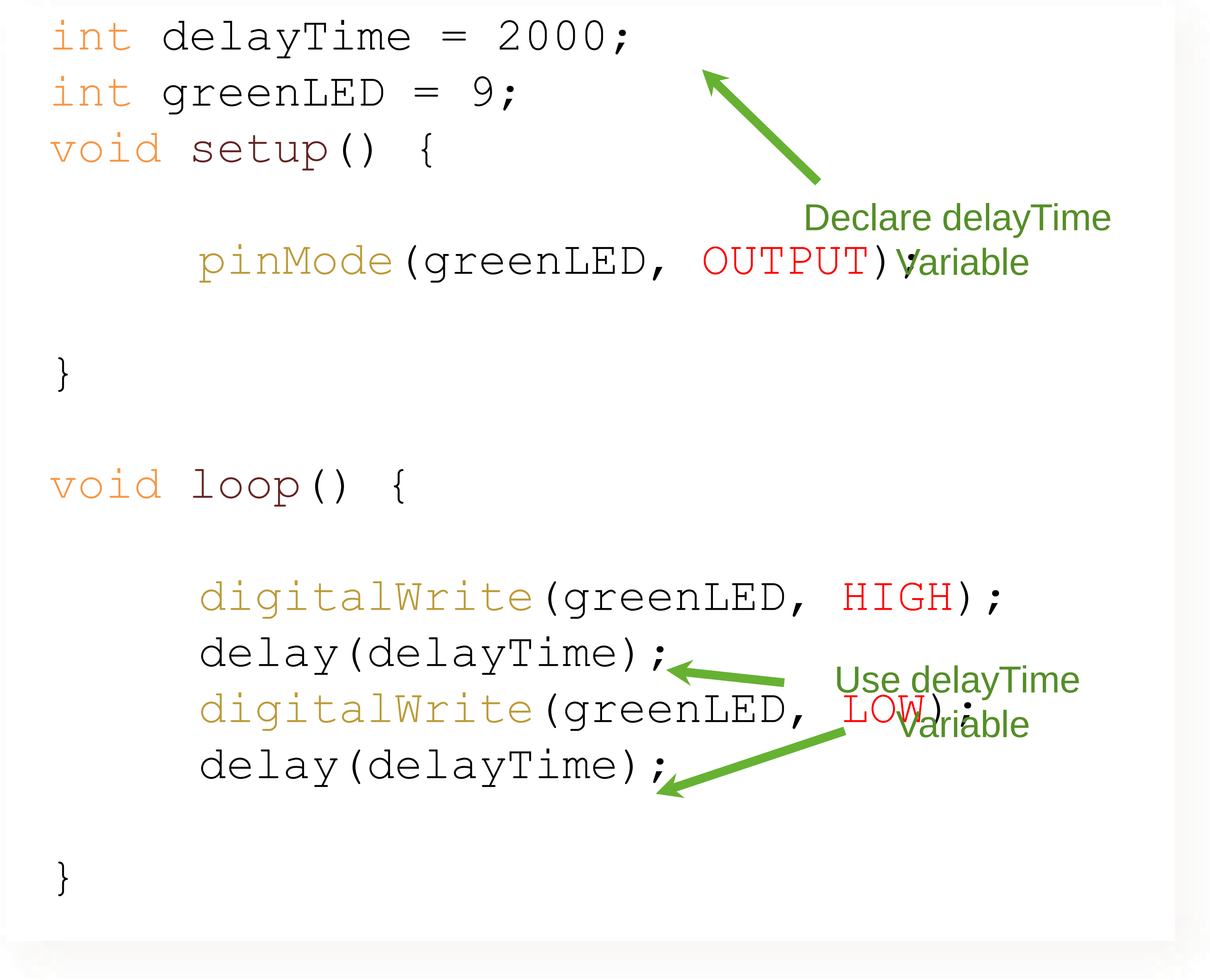
assignment
“becomes”

value

Task

- Replace all delay times with variables
- Replace LED pin numbers with variables

USING VARIABLES



```
int delayTime = 2000;
int greenLED = 9;
void setup() {

     pinMode(greenLED, OUTPUT);

}

void loop() {

     digitalWrite(greenLED, HIGH);
     delay(delayTime);
     digitalWrite(greenLED, LOW);
     delay(delayTime);

}
```

Declare delayTime Variable

Use delayTime Variable

Using Variables

```
int delayTime = 2000;
int greenLED = 9;

void setup() {
     pinMode(greenLED, OUTPUT);
}
void loop() {
     digitalWrite(greenLED, HIGH);
     delay(delayTime);
     digitalWrite(greenLED, LOW);
 delayTime = delayTime - 100;
     delay(delayTime);
}
```

subtract 100 from delayTime to gradually increase LED’s blinking speed

Conditions

- To make decisions in Arduino code we use an ‘if’ statement

- ‘If’ statements are based on a TRUE or FALSE question

VALUE COMPARISONS

GREATER THAN

a > b

GREATER THAN OR EQUAL

a >= b

LESS

a < b

LESS THAN OR EQUAL

a <= b

EQUAL

a == b

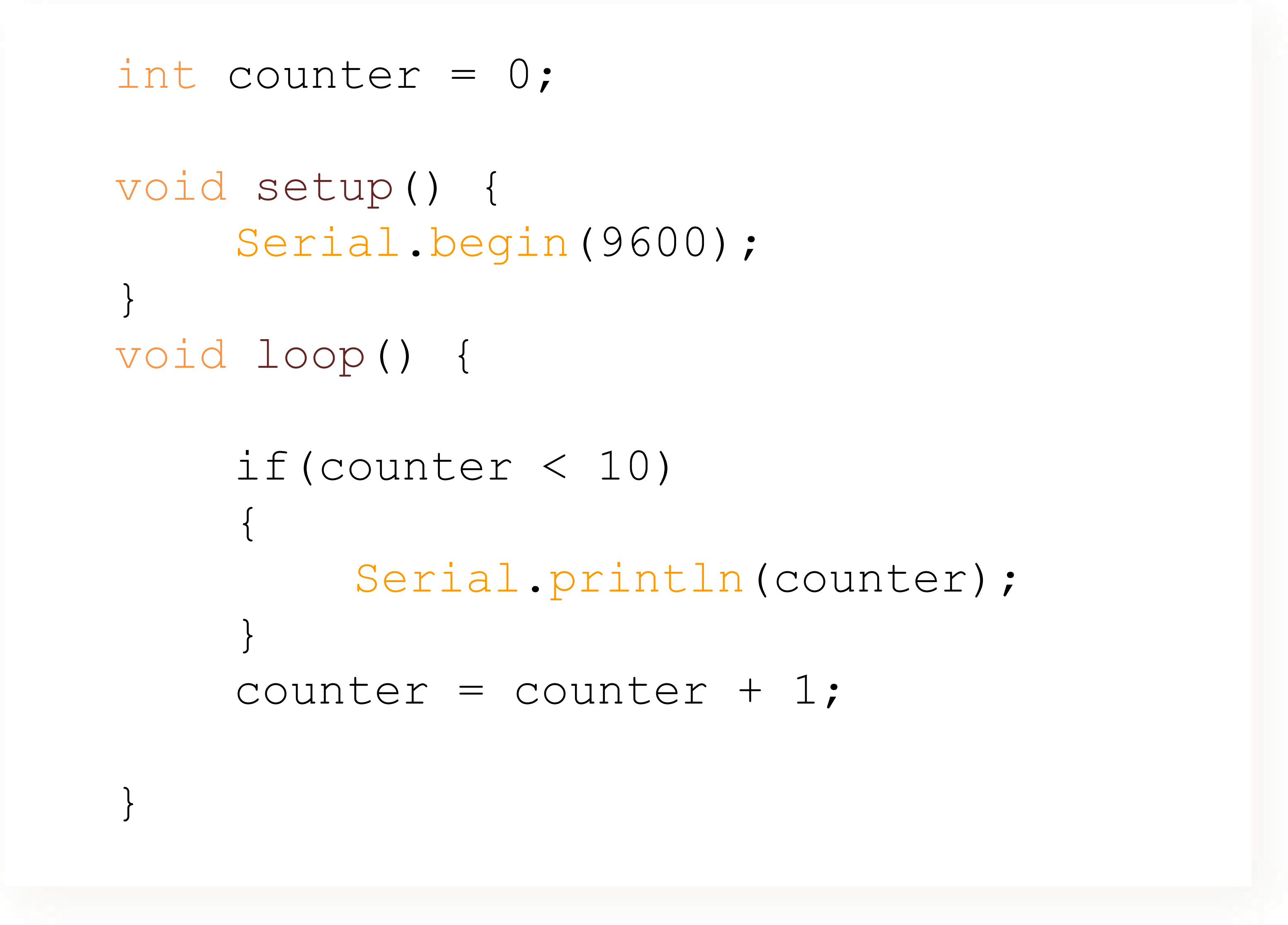
NOT EQUAL

a != b

IF Condition

```
if(true)
{
                  asdfadsf
    “perform some action”
}
```

IF Example



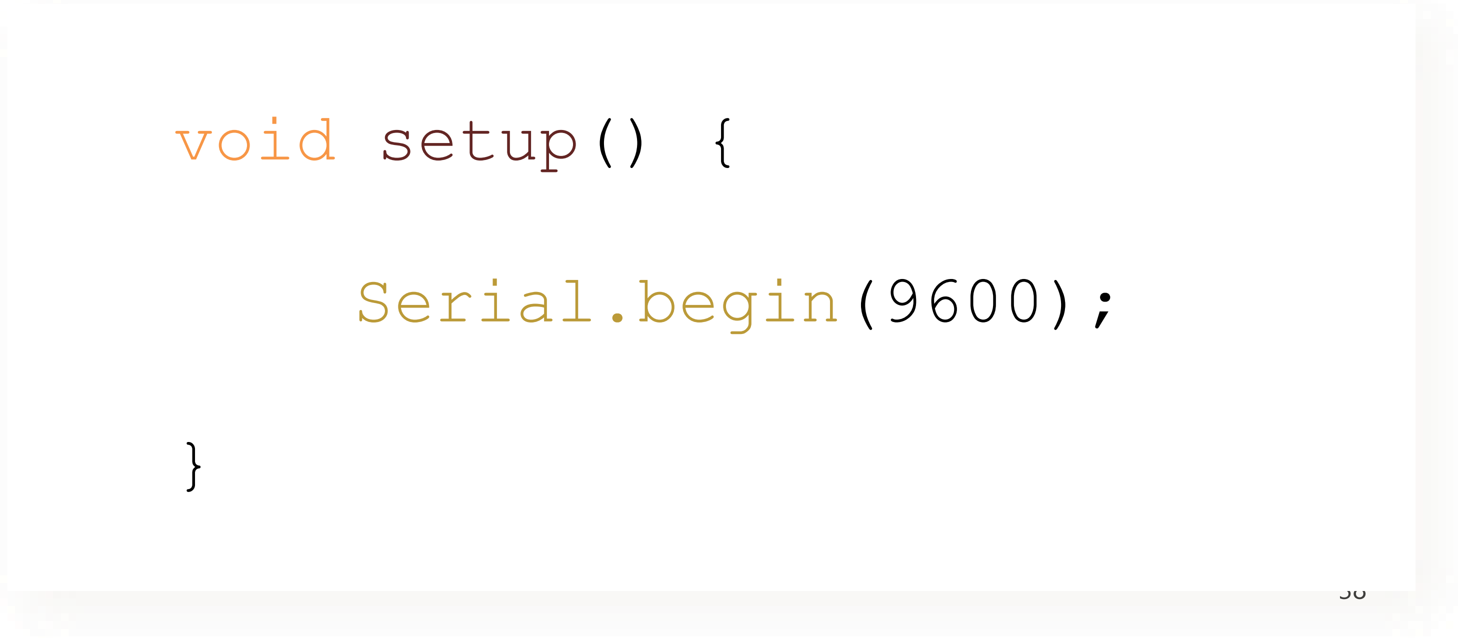
```
int counter = 0;

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

    if(counter < 10)
    {
        Serial.println(counter);
    }
    counter = counter + 1;

}
```

TASK

- Create a program that resets the delayTime to 2000 once it has reached 0

Input & Output

- Transferring data from the computer to an Arduino is done using **Serial Transmission**
- To setup Serial communication we use the following



```
void setup() {

    Serial.begin(9600);

}
```

Writing to the Console

```
void setup() {

     Serial.begin(9600);
 Serial.println(“Hello World!”);

}

void loop() {}
```

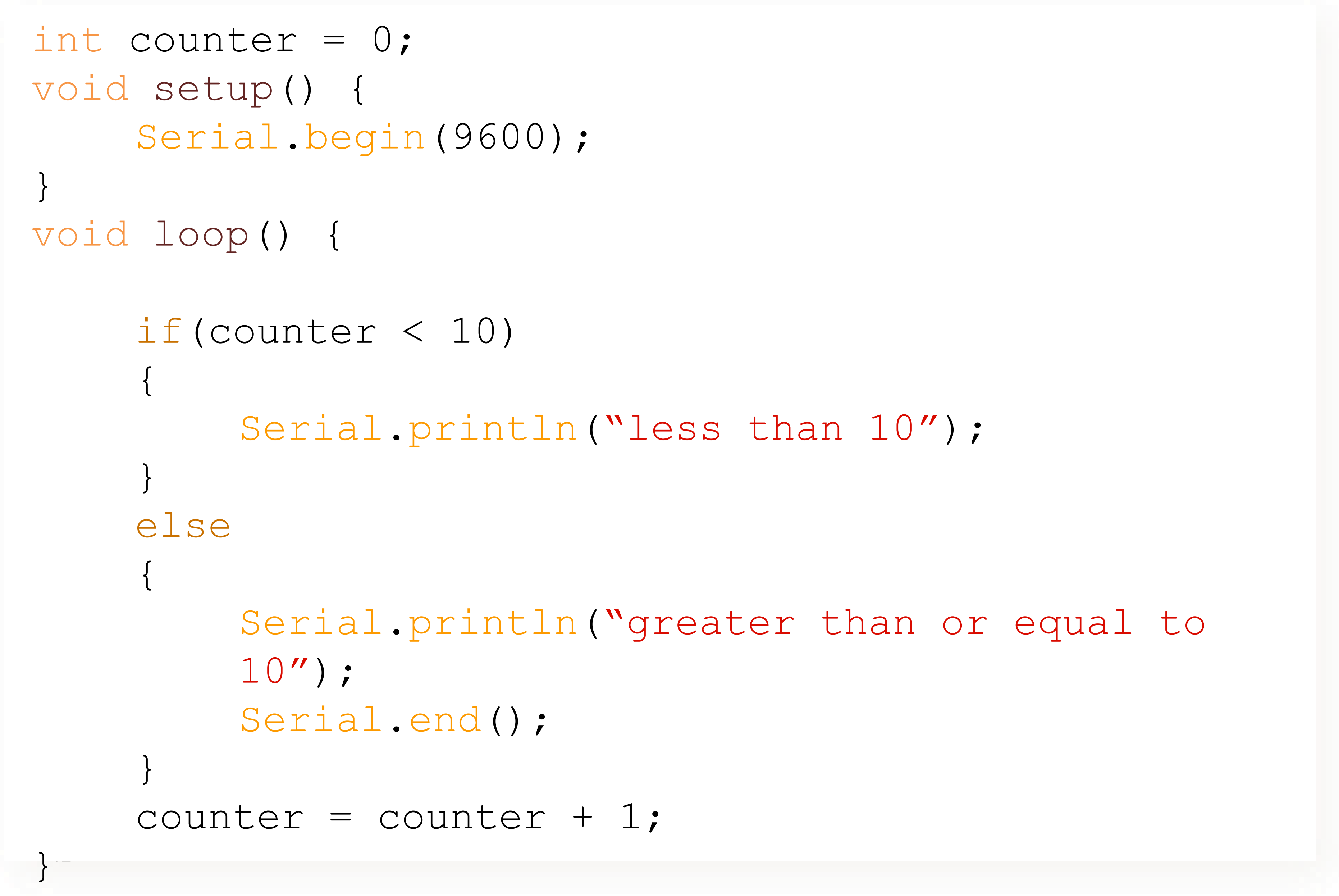
Task

- Modify your traffic light code so that each time a new LED is illuminated the console displays the status of the stoplight
- Example

IF - ELSE Condition

```
if( “answer is true”)
{
    “perform some action”
}
                        asdfadsf
else
{
    “perform some other action”
}
```

IF - ELSE Example



```
int counter = 0;
void setup() {
    Serial.begin(9600);
}
void loop() {

    if(counter < 10)
    {
        Serial.println(“less than 10”);
    }
    else
    {
        Serial.println(“greater than or equal to
        10”);
        Serial.end();
    }
    counter = counter + 1;
}
```

IF - ELSE IF Condition

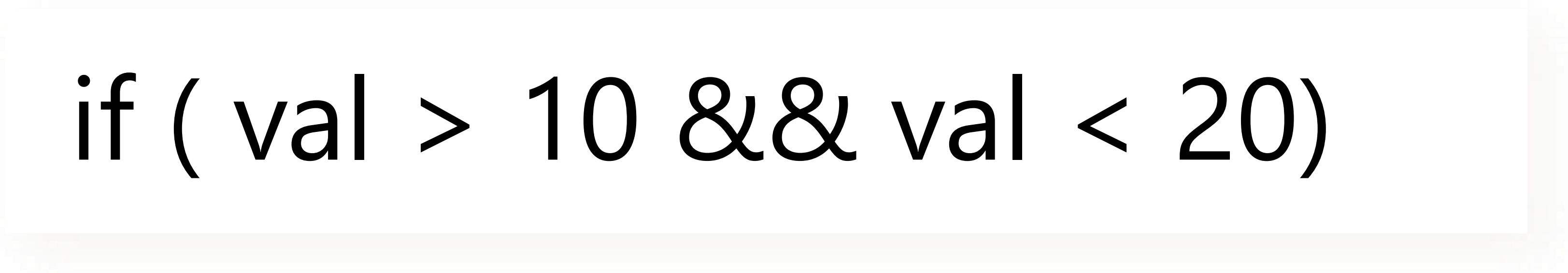
```
if( “answer is true”)
{
    “perform some action”
}
                 asdfadsf
else if( “answer is true”)
{
    “perform some other action”
}
```

IF - ELSE Example

```
int counter = 0;
void setup() {
     Serial.begin(9600);
}
void loop() {

     if(counter < 10)
     {
          Serial.println(“less than 10”);
     }
     else if (counter == 10)
     {
          Serial.println(“equal to 10”);
     }
     else
     {
          Serial.println(“greater than 10”);
          Serial.end();
     }
     counter = counter + 1;
}
```

BOOLEAN OPERATORS - AND

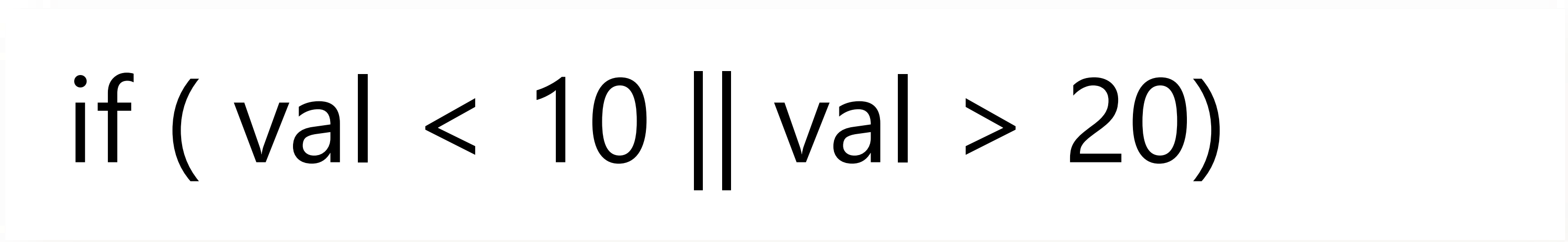
- If we want all of the conditions to be true we need to use ‘AND’ logic (AND gate)
- We use the symbols **&&**
  - Example



```
if ( val > 10 && val < 20)
```

BOOLEAN OPERATORS - OR

- If we want either of the conditions to be true we need to use ‘OR’ logic (OR gate)
- We use the symbols **||**
  - Example



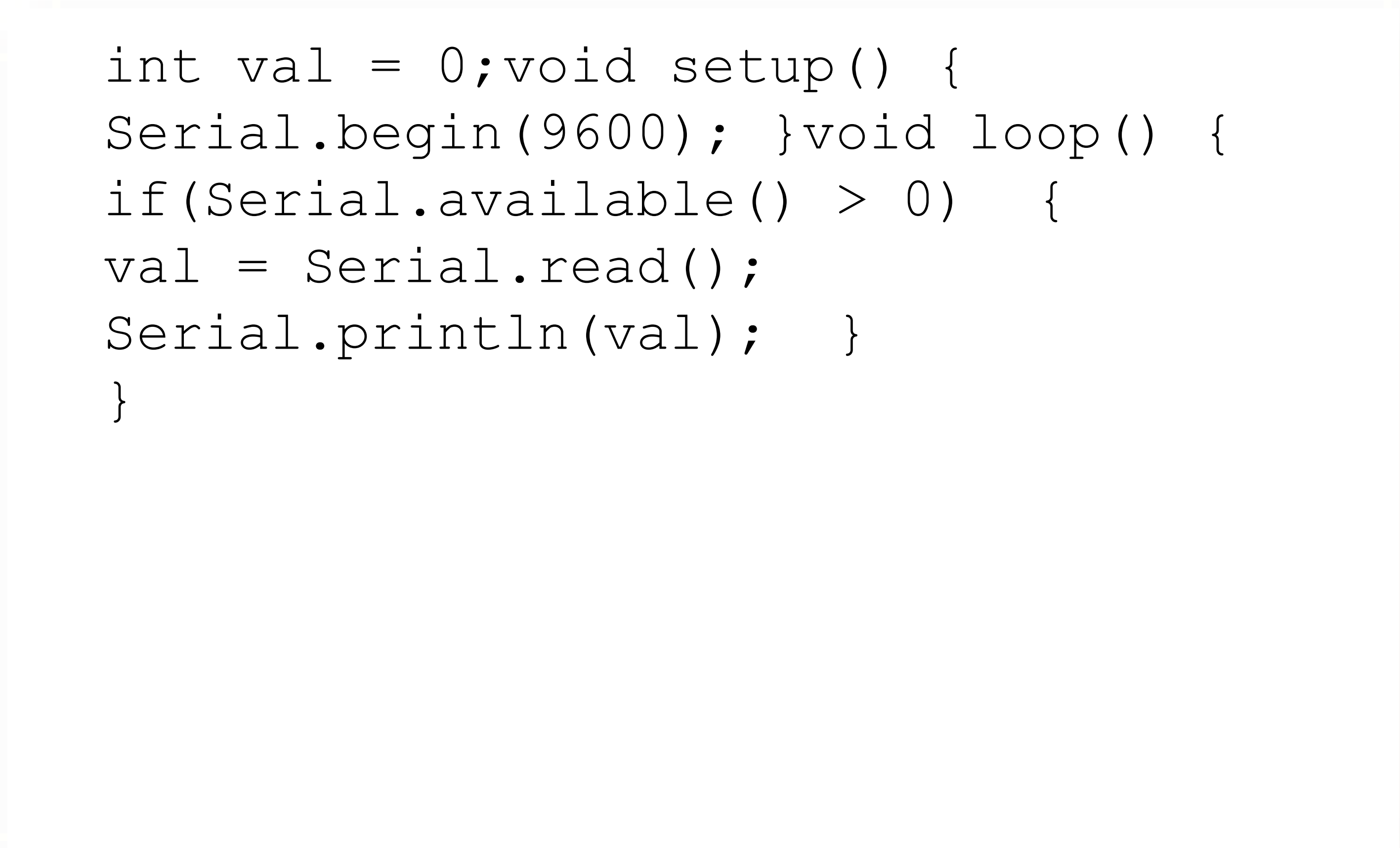
```
if ( val < 10 || val > 20)
```

TASK

- Create a program that illuminates the green LED if the counter is less than 100, illuminates the yellow LED if the counter is between 101 and 200 and illuminates the red LED if the counter is greater than 200

INPUT

- We can also use our Serial connection to get input from the computer to be used by the Arduino



```
int val = 0;void setup() {
Serial.begin(9600); }void loop() {
if(Serial.available() > 0)  {
val = Serial.read();
Serial.println(val);  }
}
```

Task

- Using input and output commands find the ASCII values of

#	ASCII	#	ASCII	@	ASCII	@	ASCII	@	ASCII	@	ASCII
1	49	6	54	a	97	h	104	o	111	v	118
2	50	7	55	b	98	i	105	p	112	w	119
3	51	8	56	c	99	j	106	q	113	x	120
4	52	9	57	d	100	k	107	r	114	y	121
5	53			e	101	l	108	s	115	z	122
				f	102	m	109	t	116		
				g	103	n	110	u	117		

Task

- Create a program so that when the user enters 1 the green light is illuminated, 2 the yellow light is illuminated and 3 the red light is illuminated

Create a program so that when the user enters ‘b’ the green light blinks, ‘g’ the green light is illuminated ‘y’ the yellow light is illuminated and ‘r’ the red light is illuminated

Task

- Light Mode Video
- Create a program that
  - when the user enters ‘1’ all lights turn on
  - when the user enters ‘2’ all lights flash
  - when the user enters ‘3’ lights cycle repeatedly
  - when the user enters ‘q’ or ‘e’ the lights turn off
  - when + or - is pressed the speed of the LED increase or decrease

BOOLEAN VARIABLES

```
boolean done = true;
```

TASK

- Write a program that asks the user for a number and outputs the number that is entered. Once the number has been output the program finishes.
  - EXAMPLE:

```
Please enter a number: 1 <enter>
The number you entered was: 1
```

TASK

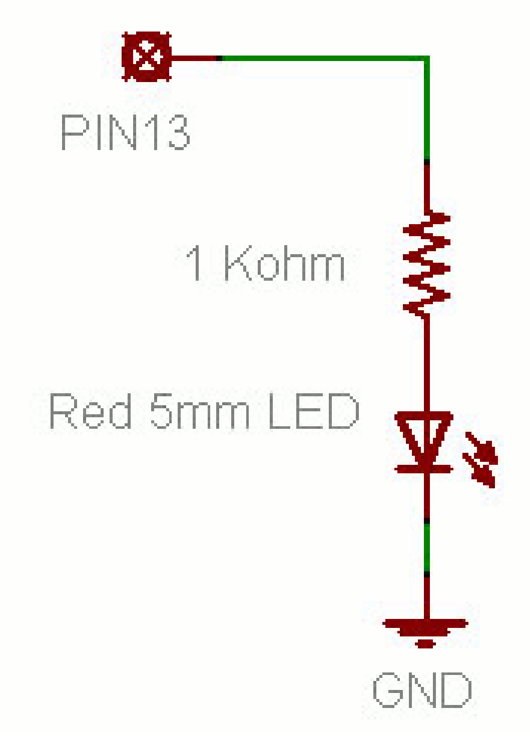
- Write a program that asks the user for a number and outputs the number squared that is entered. Once the number has been output the program finishes.

```
Please enter a number: 4 <enter>
Your number squared is: 16
```

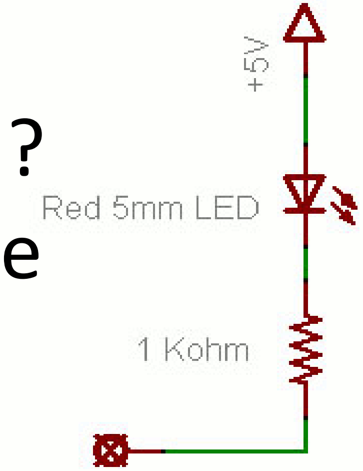
Christmas Light Assignment

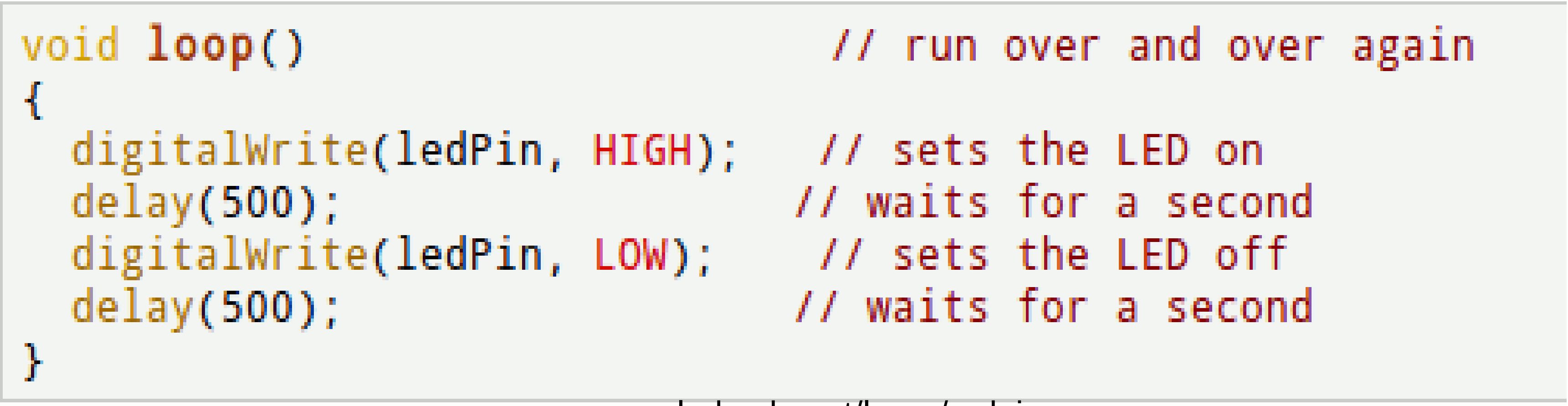
- Using at least 5 LEDs create a program that has at least 4 different modes
- Must use if statements and user input to switch modes
- Marks:
  - Creativity
  - Working user input
  - Four Working modes
  - Instructions

Putting It Together



- Complete the sketch (program) below.
- What output will be generated by this program?
- What if the schematic were changed? ➔





```
void loop()                        // run over and over again
{
  digitalWrite(ledPin, HIGH);    // sets the LED on
  delay(500);                   // waits for a second
  digitalWrite(ledPin, LOW);     // sets the LED off
  delay(500);                   // waits for a second
}
```

www.ladyada.net/learn/arduino

Good References

Getting Started with Arduino products | Arduino

www.ladyada.net/learn/arduino

www.EarthshineElectronics.com