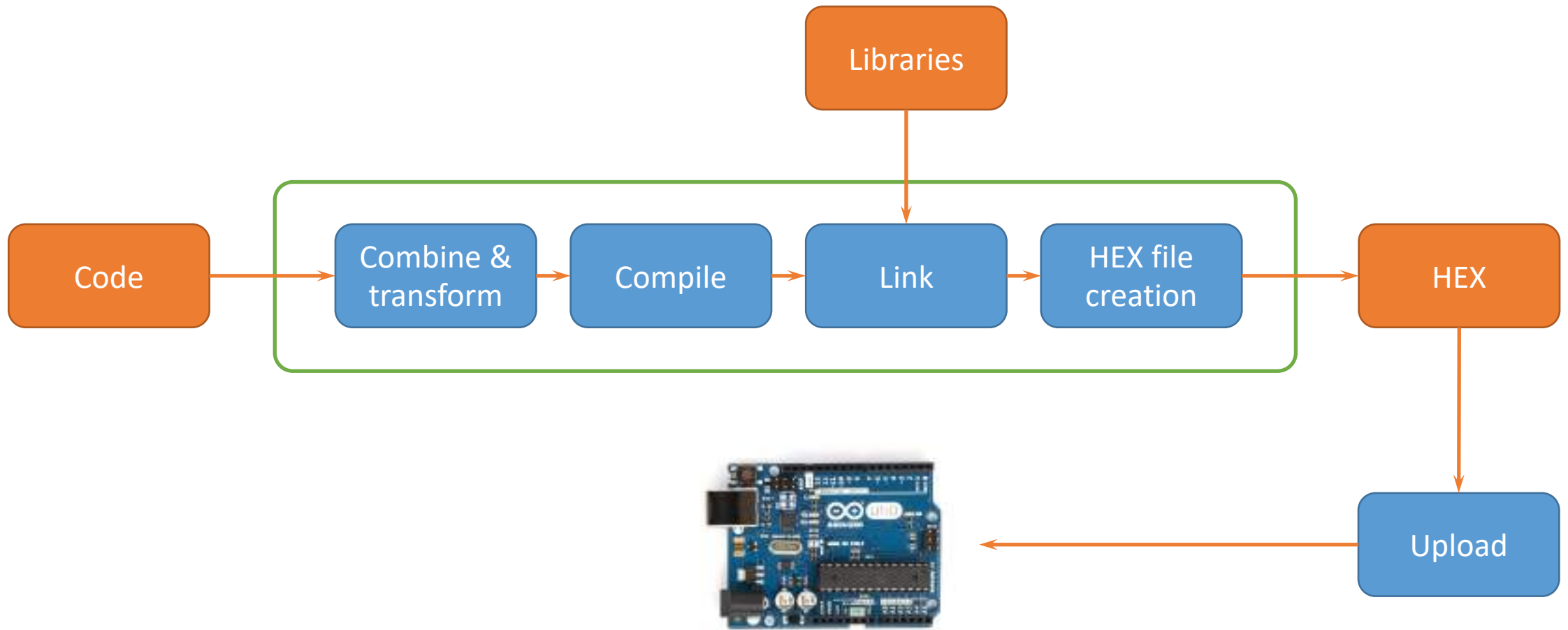


Arduino programs

- Arduino toolchain
- Cross-compilation
- Arduino sketches
- Sketch structure
- Pins
- Input and output
- Blink example

Arduino toolchain

Verify and upload



Arduino sketches

Arduino programs

- A program is called a sketch
- C++ program using Arduino library functions
- C++ is a superset of C
 - All C programs are legal C++

Object-oriented programming

- Organize your code through encapsulation
- Group together data and functions that are related
- User-defined type is specific to an application
 - `int` type has data (the number) and functions (+, -, *)

Classes and members

```
class X {  
public:  
    int m;  
    int mf(int v){int old=m; m=v; return old;}  
};  
  
X var;  
var.m=7;  
int z = var.mf(9);
```

Classes and members

- Declaration of a variable creates an object
- . operator used to access members
 - Data and functions
- Functions can be defined inside the class

Classes in libraries

- We will
 - not need to know much about classes
 - not define classes
 - use classes defined in libraries
- Examples
 - `Ethernet.begin(mac)`
 - `Serial.begin(speed);`
 - `client.print("Hello");`
 - `Serial.print("Hello");`

Sketch structure

setup() function

- A sketch does not have a `main()` function
- Every sketch has a `setup()` function
 - Executed once when Arduino is powered up
 - Used for initialization operations
 - Returns no value, takes no arguments

```
void setup( ) {  
    ...  
}
```

loop() function

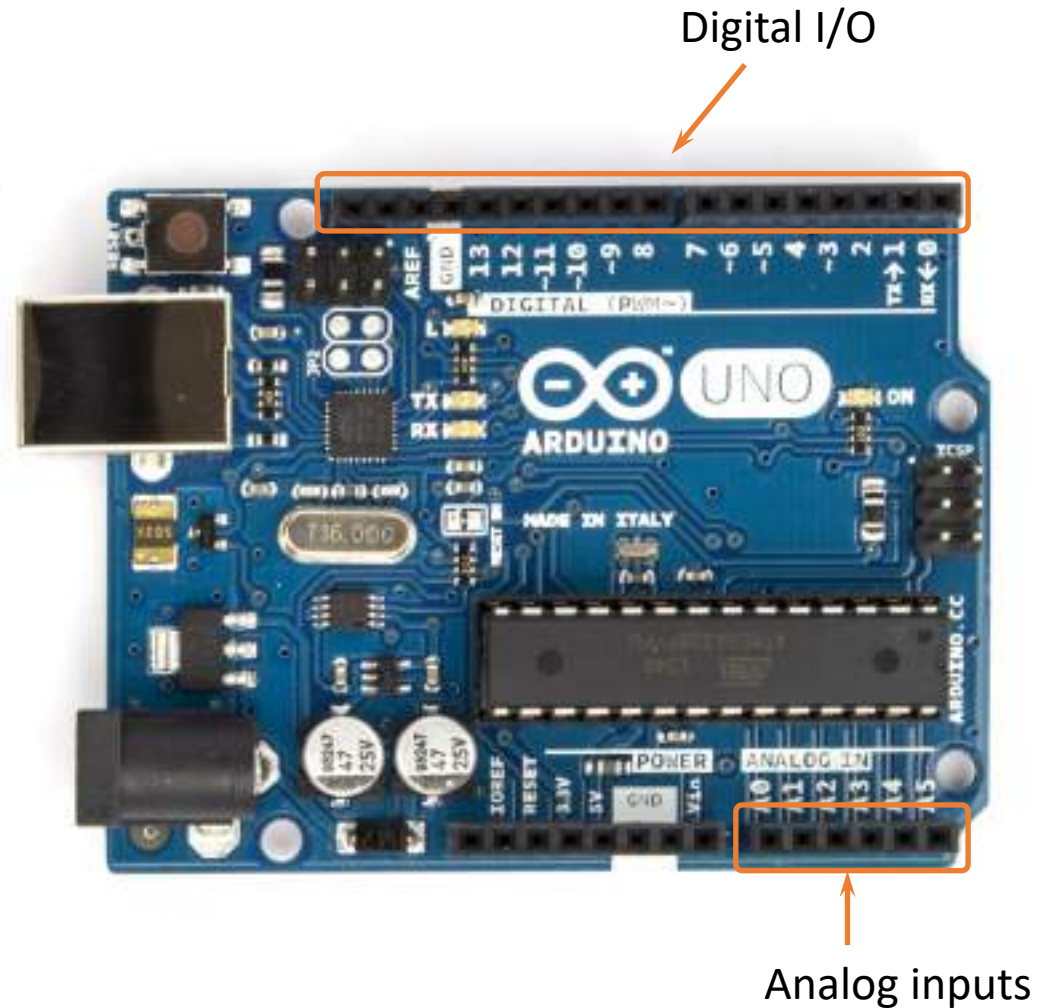
- Every sketch has a `loop()` function
 - Executed iteratively as long as the Arduino is powered up
 - `loop()` starts executing after `setup()` has finished
 - `loop()` is the main program control flow
 - Returns no value, takes no arguments

```
void loop( ) {  
    ...  
}
```

Pins

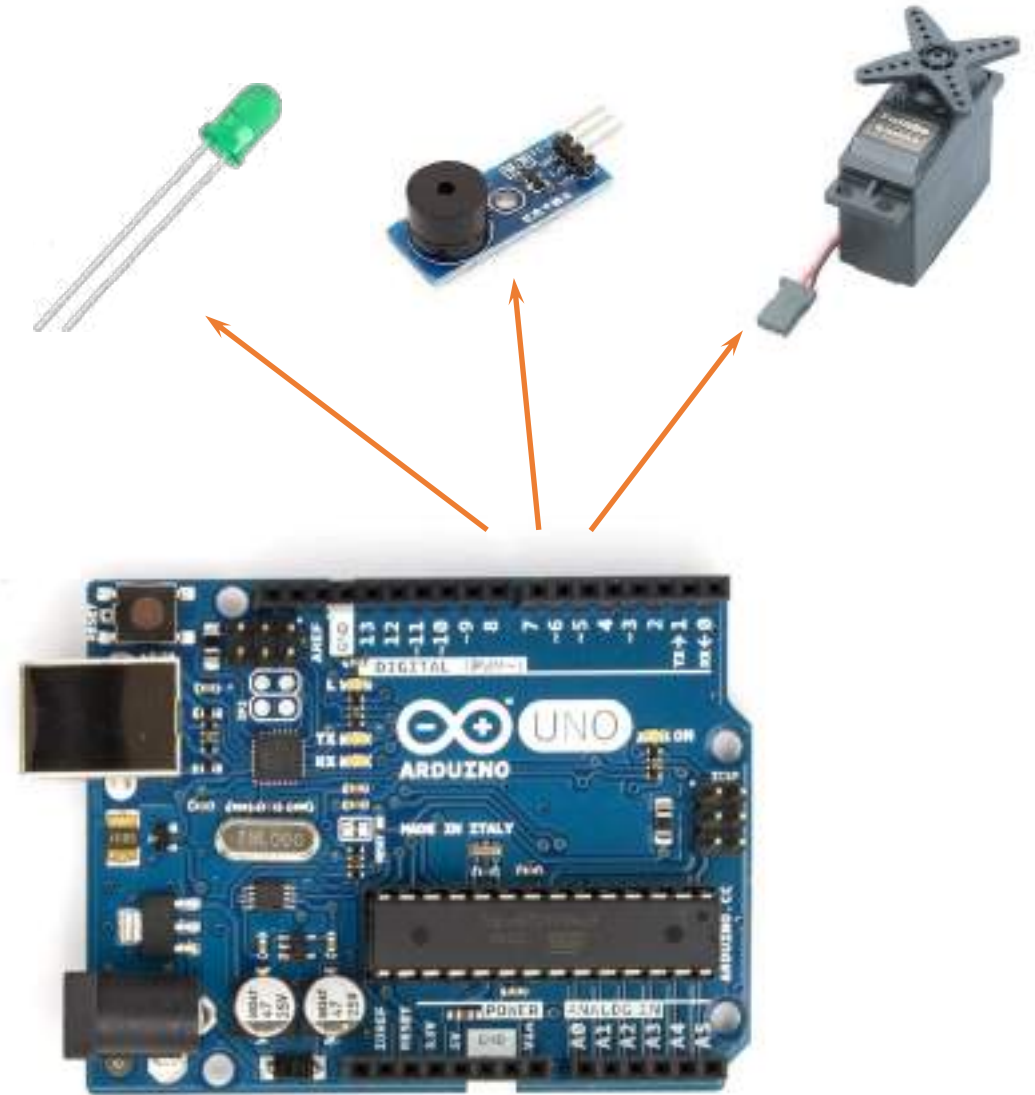
Pins

- Pins are wires connected to the microcontroller
- Pins are the interface of the microcontroller
- Pin voltages are controlled by a sketch
- Pin voltages can be read by a sketch



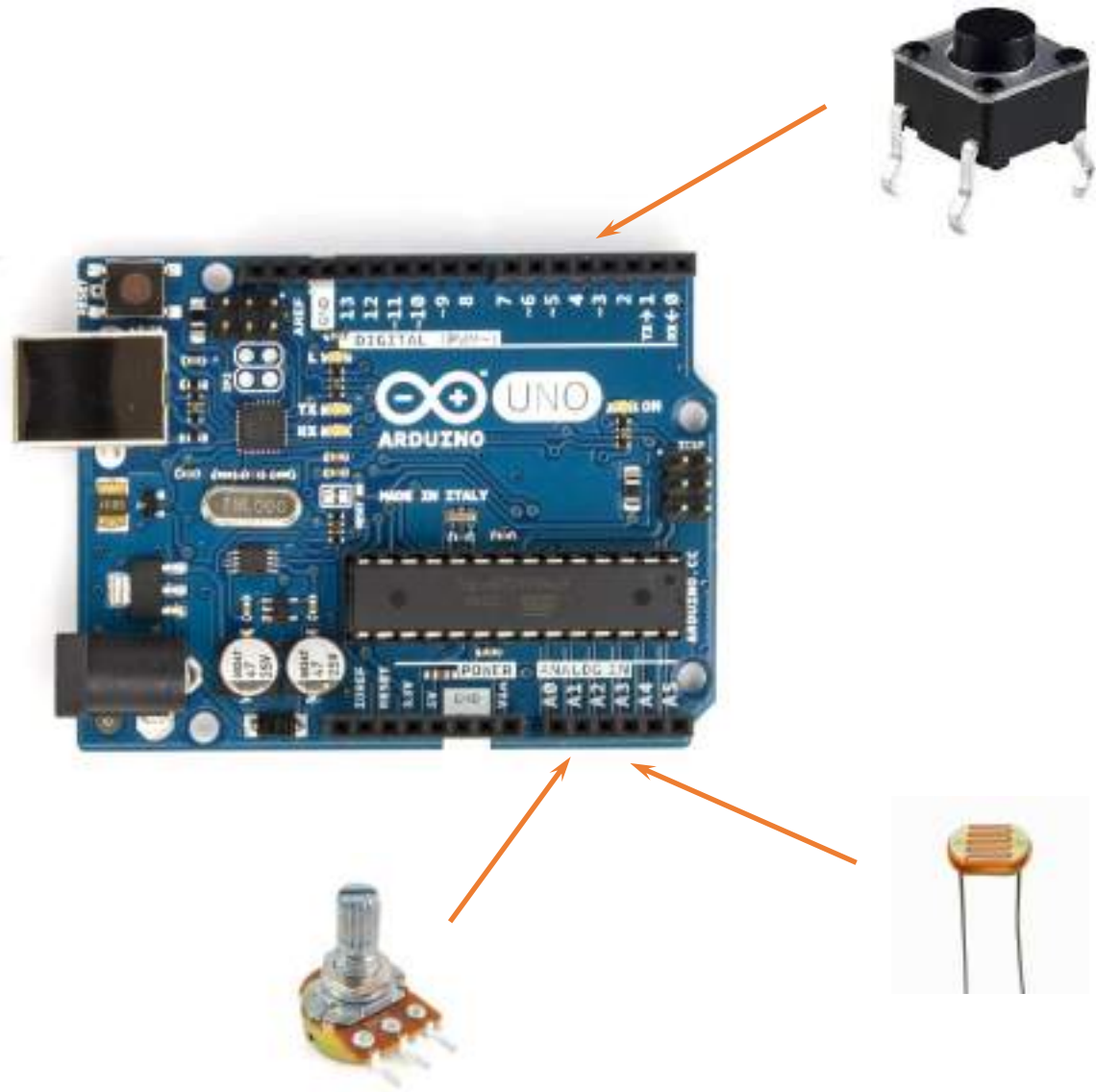
Output pins

- Output pins are controlled by the Arduino
 - Voltage is determined by your sketch
 - Other components can be controlled through outputs



Input pins

- Input pins are controlled by other components
 - Arduino reads the voltage on the pins
 - Allows it to respond to events and data



Digital vs. analog

- Some pins are digital-only
 - Read digital input, write digital output
 - 0 volts or 5 volts
- Some pins can be analog inputs
 - Can read analog voltages on the pin
 - Useful for analog sensors
- Analog-only pins are clearly labeled
- No pins can generate an analog output

Input and output

Input/output (I/O)

- These functions allow access to the pins

```
void pinMode(pin, mode)
```

- Sets a pin to act as either an input or an output
- `pin` is the number of the pin
 - 0–13 for the digital pins
 - A0–A5 for the analog pins
- `mode` is the I/O mode the pin is set to
 - `INPUT`, `OUTPUT`, or `INPUT_PULLUP`
 - `INPUT_PULLUP` acts as input with reversed polarity

Digital input

```
int digitalRead(pin)
```

- Returns the state of an input pin
- Returns either LOW (0 volts) or HIGH (5 volts)

- Example

```
int pinval;  
pinval = digitalRead(3);
```

- `pinval` is set to the state of digital pin 3

Digital output

```
void digitalWrite(pin, value)
```

- Assigns the state of an output pin
- Assigns either LOW (0 volts) or HIGH (5 volts)

- Example

```
digitalWrite(3, HIGH);
```

- Digital pin 3 is set HIGH (5 volts)

Analog input

```
int analogRead(pin)
```

- Returns the state of an analog input pin
- Returns an integer from 0 to 1023
- 0 for 0 volts, 1023 for 5 volts

- Example

```
int pinval;  
pinval = analogRead(A3);
```

- `pinval` is set to the voltage on A3 (0 for 0 volts, 1023 for 5 volts)
- The pin must be an analog pin

Blink example

Delay

```
void delay(msec)
```

- Pauses the program for msec milliseconds
- Useful for human interaction

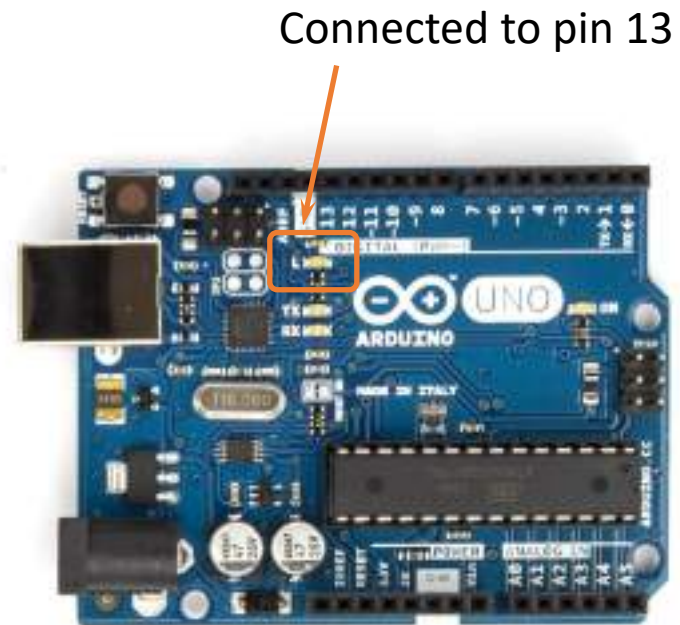
- Example

```
digitalWrite(3, HIGH);  
delay(1000);  
digitalWrite(3, LOW);
```

- Pin 3 is HIGH for 1 second

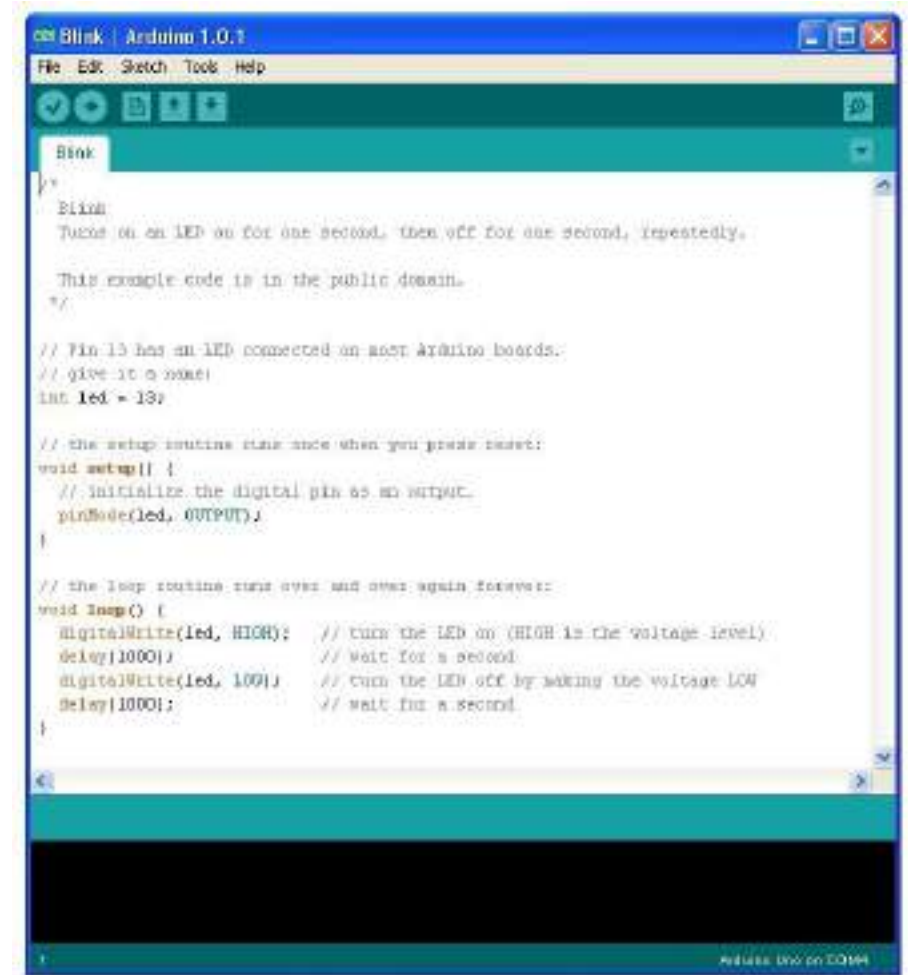
Blink example

- Blink is the generic simple example for embedded systems
 - Like “hello, world”
- File → Examples → 01.Basic → Blink
- Causes an LED to blink
- LED is built-in, so no wiring required



Blink sketch

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```



저번 주 실습 예제 설명

Blink example

1번 예제 - 방법1

```
int led_red = 0; // the red LED is connected to Pin  
0 of the Arduino
```

```
int i;
```

```
int states=1;
```

```
void blink_n_times(int n, int ms) {
```

```
    for(i=0;i<n;i++){  
        digitalWrite(led_red, HIGH);  
        delay(ms);  
        digitalWrite(led_red, LOW);  
        delay(ms);  
    }
```

```
}
```

```
void setup() {
```

```
    // set up all the LEDs as OUTPUT
```

```
    pinMode(led_red, OUTPUT);
```

```
}
```

```
void loop() {
```

```
    if(states == 1){
```

```
        blink_n_times(5,1000);
```

```
    states = 0;
```

```
}
```

```
}
```

1번 예제 - 방법2

```
int led_red = 0; // the red LED is connected to Pin 0 of  
the Arduino
```

```
int c = 0;
```

```
void blink_n_times(int n, int ms) {
```

```
    while(c<n){
```

```
        digitalWrite(led_red, HIGH);
```

```
        delay(ms);
```

```
        digitalWrite(led_red, LOW);
```

```
        delay(ms);
```

```
        c+=1;
```

```
    }
```

```
}
```

```
void setup() {
```

```
    // set up all the LEDs as OUTPUT
```

```
    pinMode(led_red, OUTPUT);
```

```
}
```

```
void loop() {
```

```
    blink_n_times(5,1000);
```

```
}
```

2번 예제

```
int led_red = 0; // the red LED is connected to Pin 0 of the Arduino
```

```
int i;
```

```
void blink_n_times(int n, int ms) {
```

```
    for(i=0;i<n;i++){
        digitalWrite(led_red, HIGH);
        delay(ms);
        digitalWrite(led_red, LOW);
        delay(ms);
    }
}
```

```
void setup() {
    // set up all the LEDs as OUTPUT
    pinMode(led_red, OUTPUT);
}
```

```
int n = 1;
```

```
int ms_initial = 6000;
```

```
int ms = 6000;
```

```
void loop() {
    blink_n_times(n,ms);
    n++;
    ms = ms_initial / n;
}
```

Lab

State change detection example

- Connect to Tinkercad at <http://tinkercad.com> and log-in.
- Go to Circuits and click “Create new circuit” button.
- Drag-and-drop the State change detection example from “Arduino” → “State change detection”.
- Examine the code and guess what it does.
- Start simulation to check if your guess was correct.
 - It counts the number of button press and turns on the built-in LED at every 4 presses.

Neopixel example

- Connect to Tinkercad at <http://tinkercad.com> and log-in.
- Go to Circuits and click “Create new circuit” button.
- Drag-and-drop the Neopixel example from “Arduino” → “Neopixel”.
- Examine the code and guess what it does.
- Start simulation to check if your guess was correct.
 - It sequentially turns on the Neopixel LEDs with randomly varying colors.

Neopixel with a button

- Combine “State change detection” and “Neopixel” to have your simulator do the following.
 - When your Arduino is turned on, the Neopixel is sequentially turned on with random colors.
 - Neopixel color changes when the number of button press increases (that is when the button status changes from “Released” to “Pushed”).