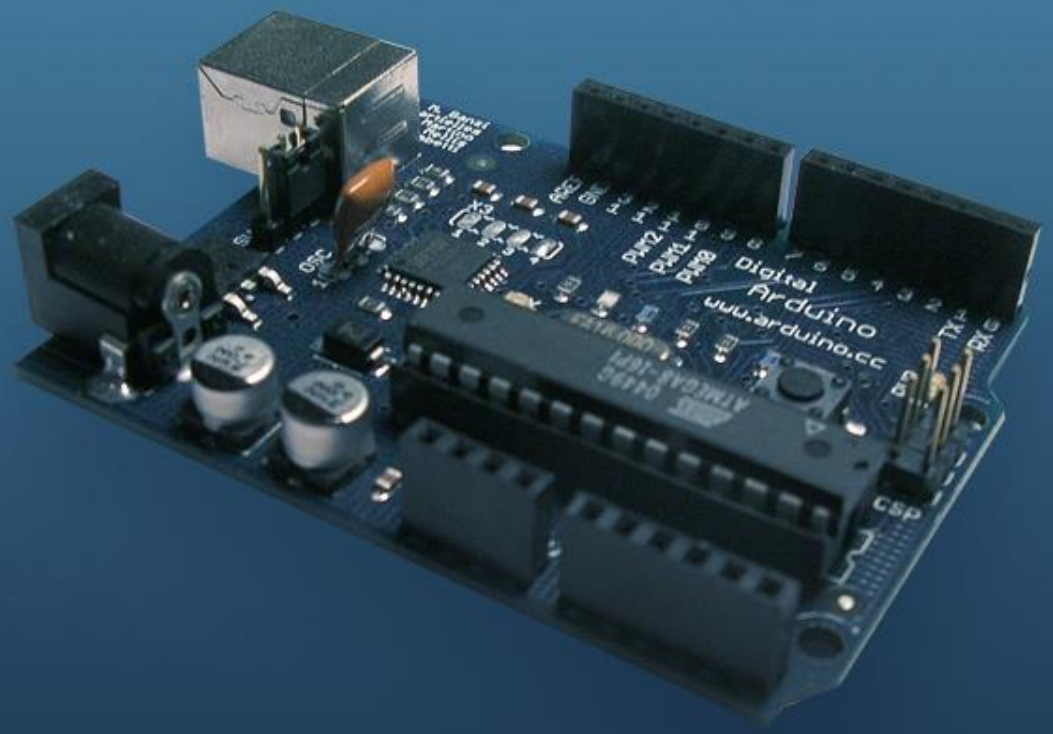
Arduino

Physical Computing I/O board



By/Mohamed Elsayed Mohamed

Content

- Introduction
- Arduino IDE
- Code Structure
- Variables Declaration
- Arithmetic Operators
- Control Statements
- Loops
- Functions
- I/O Instructions
- Serial Instructions
- Tasks

Introduction

What is Arduino?

- Arduino is an open-source computing platform based on a simple i/o board and a software development environment .

- Arduino can be used to develop embedded systems or prototypes of embedded systems so fast and easily.

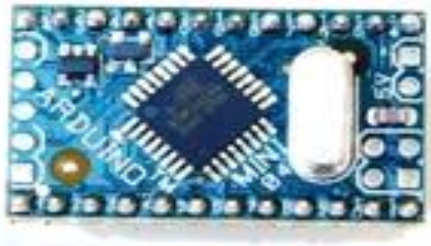
Introduction

Arduino boards

UNO

Mega

Arduino Nano



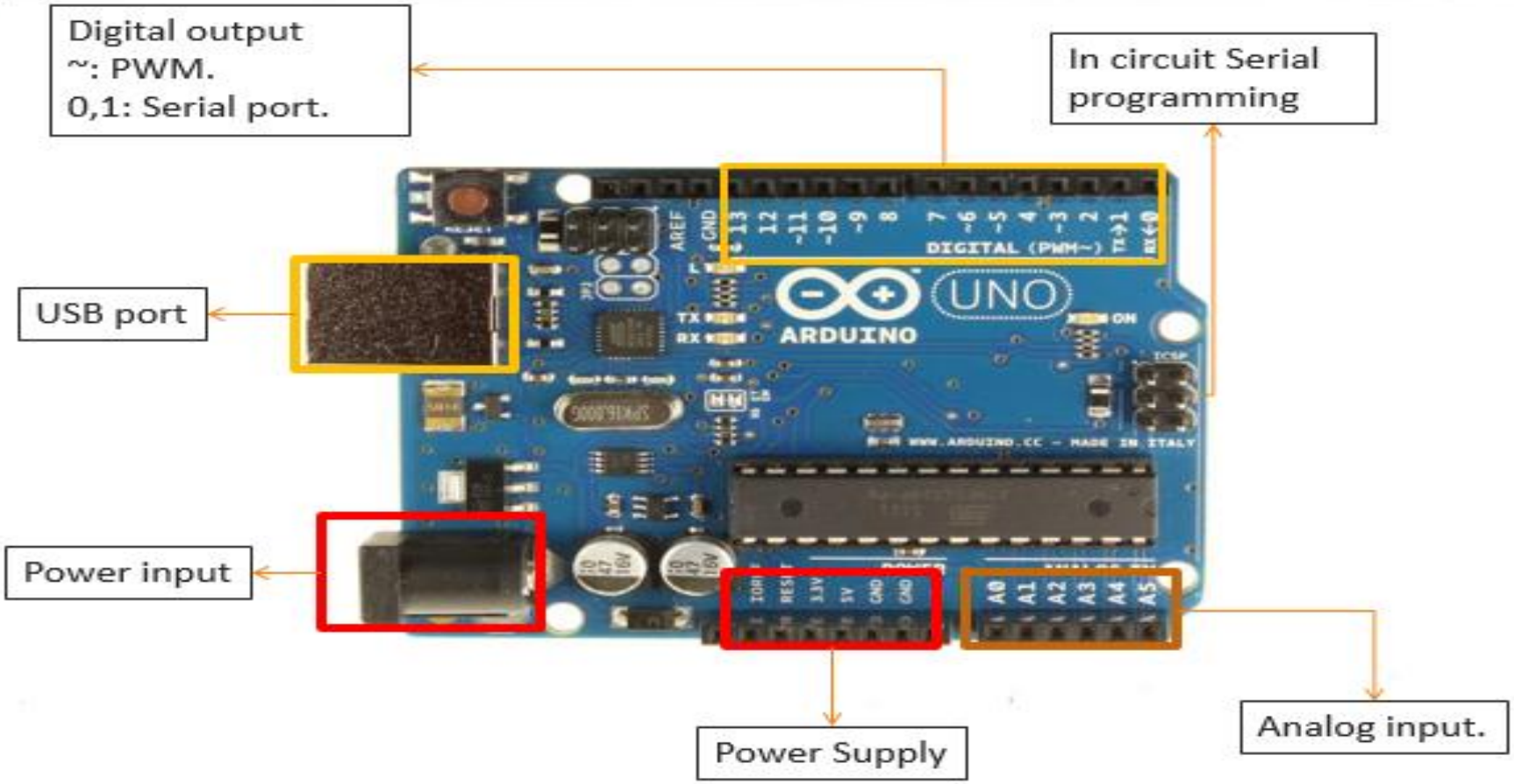
Arduino Mini

Arduino BT

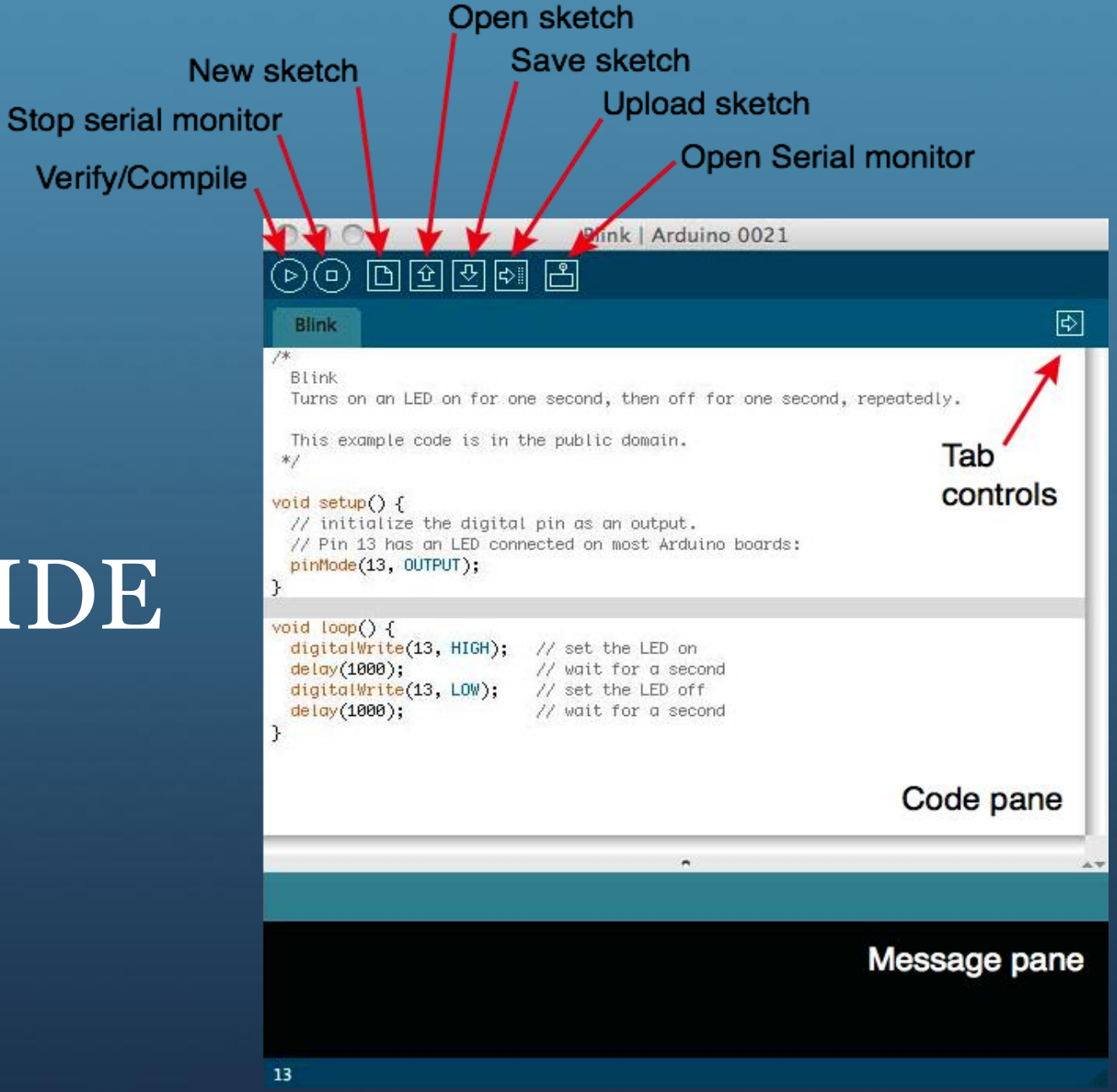
Other

Introduction

Arduino UNO



Arduino IDE


Open sketch
New sketch
Save sketch
Stop serial monitor
Upload sketch
Open Serial monitor
Verify/Compile
Blink | Arduino 0021
Blink
/*
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
}
Tab controls
Code pane
Message pane
13

Code Structure: Header

Blink | Arduino 0021

Blink §

```
/*
  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
}
```

Header provides information

Code Structure: setup function

Blink | Arduino 0021

Blink §

```
/*
  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
}
```

setup function is executed only once at the start

Code Structure: loop function

Blink | Arduino 0021

Blink §

```
/*
  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
}
```

loop function is repeated indefinitely

Code Structure

- Curly braces { }

```
type function()
{
  statements;
}
```

Define the beginning and the end of function and statement blocks

forgetting to end a line with a semicolon will lead to compilation error !!!

- Semicolon ;

Variable Declaration

Integer: used with integer variables
Ex: int x=1200;
Character: used with single character, represent value from -127 to 128.
Ex. char c=‘r’;
Long: Long variables are extended size variables for number storage, and store 32 bits (4 bytes), from -2,147,483,648 to 2,147,483,647.
Ex. long u=199203;
Floating-point numbers can be as large as 3.4028235E+38 and as low as -3.4028235E+38. They are stored as 32 bits (4 bytes) of information.
Ex. float num=1.291; [The same as **double** type]

Variable Scope

```
int value;                    // 'value' is visible
                              // to any function

void setup()
{
  // no setup needed
}

void loop()
{
  for (int i=0; i<20;)   // 'i' is only visible
  {                      // inside the for-loop
    i++;
  }
  float f;               // 'f' is only visible
}                        // inside loop
```

Global variable

Local variable

Arithmetic Operators

```
y = y + 3;
x = x - 7;
i = j * 6;
r = r / 5;
```

Compound Assignment

```
x ++      // same as x = x + 1, or increments x by +1
x --      // same as x = x - 1, or decrements x by -1
x += y    // same as x = x + y, or increments x by +y
x -= y    // same as x = x - y, or decrements x by -y
x *= y    // same as x = x * y, or multiplies x by y
x /= y    // same as x = x / y, or divides x by y
```

Comparison Operators

```
x == y    // x is equal to y
x != y    // x is not equal to y
x <  y    // x is less than y
x >  y    // x is greater than y
x <= y    // x is less than or equal to y
x >= y    // x is greater than or equal to y
```

Logical Operators

Logical AND:

```
if (x > 0 && x < 5)     // true only if both
                        // expressions are true
```

Logical OR:

```
if (x > 0 || y > 0)     // true if either
                        // expression is true
```

Logical NOT:

```
if (!x > 0)             // true only if
                        // expression is false
```

Control statements

- If statement

```
if (someVariable ?? value)
{
  doSomething;
}
```

If(x=10)

If(x==10)

Control statements

- If….else statement

```
if (inputPin == HIGH)
{
  doThingA;
}
else
{
  doThingB;
}
```

Control statements

- If….else statement

```
if (inputPin < 500)
{
  doThingA;
}
else if (inputPin >= 1000)
{
  doThingB;
}
else
{
  doThingC;
}
```

Loop statements

- For Loop

```
for (initialization; condition; expression)
{
  doSomething;
}
```

```
for (int i=0; i<20; i++)  // declares i, tests if less
{                         // than 20, increments i by 1
  digitalWrite(13, HIGH); // turns pin 13 on
  delay(250);             // pauses for 1/4 second
  digitalWrite(13, LOW);  // turns pin 13 off
  delay(250);             // pauses for 1/4 second
}
```

Loop statements

- While Loop

```
while (someVariable ?? value)
{
   doSomething;
}
```

```
While (someVariable < 200) // tests if less than 200
{
  doSomething;             // executes enclosed statements
  someVariable++;          // increments variable by 1
}
```

Loop statements

- do…while Loop

```
do
{
  doSomething;
} while (someVariable ?? value);
```

```
do
{
  x = readSensors();     // assigns the value of
                         // readSensors() to x
  delay(50);             // pauses 50 milliseconds
} while (x < 100);       // loops if x is less than 100
```

Functions

```
type functionName(parameters)
{
  statements;
}
```

```
int delayVal()
{
  int v;                    // create temporary variable 'v'
  v  = analogRead(pot);     // read potentiometer value
  v /= 4;                   // converts 0-1023 to 0-255
  return v;                 // return final value
}
```

Digital I/O instructions

- pinMode(pin,mode)

```
pinMode(pin, OUTPUT);    // sets ‘pin’ to output
```

- digitalRead(pin)

```
value = digitalRead(Pin);   // sets 'value' equal to
                            // the input pin
```

- digitalWrite(pin,value)

```
digitalWrite(pin, HIGH);   // sets 'pin' to high
```

Analog I/O instructions

- analogRead(pin)

```
value = analogRead(pin);  // sets 'value' equal to 'pin'
```

Analog pins don't need to be declared as INPUT or OUTPUT

The resulting value range from 0 to 1023

- analogWrite(pin,value)

```
analogWrite(pin, value);  // writes 'value' to analog 'pin'
```

Writing an analog value using (PWM) to PWM pins 3,5,6,9,10

The value can be between 0-255

Serial instructions

- Serial.begin(rate)

```
void setup()
{
  Serial.begin(9600);   // opens serial port
}                       // sets data rate to 9600 bps
```

- Serial.println(data)

```
Serial.println(analogValue);  // sends the value of
                              // 'analogValue'
```

Delay instructions

- delay(ms)

```
delay(1000);   // waits for one second
```

- millis()

```
value = millis();  // sets ‘value’ equal to millis()
```

Functions

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

  DashedLine();
  Serial.println("| Program Menu |");
  DashedLine();
}

void loop() {
}

void DashedLine()
{
  Serial.println("----------------");
}
```

Function is called here

Function is called again

Function is created here

Functions

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

  DashedLine();
  Serial.println("| Program Menu |");
  DashedLine();
}

void loop() {
}

void DashedLine()
{
  Serial.println("-----------------");
}
```

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

  // draw the menu box
  DashedLine(24);
  Serial.println("| Program Options Menu |");
  DashedLine(24);
}

void loop() {
}

void DashedLine(int len)
{
  int i;

  // draw the line
  for (i = 0; i < len; i++) {
    Serial.print("-");
  }
  // move the cursor to the next line
  Serial.println("");
}
```

Tasks

1) Execute an Arduino sketch to simply turn a led on and off, the led is connected to pin 13 and is blinked every second.

2) Execute an Arduino sketch to simply read a switch connected to pin 2 to control a led connected to pin 13.

3) Execute an Arduino sketch to simply brighten and dim a led connected to any PWM pins.

4) Execute an Arduino sketch that model traffic lightening system using red, yellow, and green leds .

5) Execute an Arduino sketch that use LDR sensor to control the lighting of a led connected to pin 13.

References

- Evans, B. (2011). Beginning Arduino Programming, Apress
- https://www.slideshare.net/avikdhupar/intro-to-arduino
- https://www.slideshare.net/xxahmedsakrxx/introduction-to-arduino
- www.Arduino.com

Thank you

☺