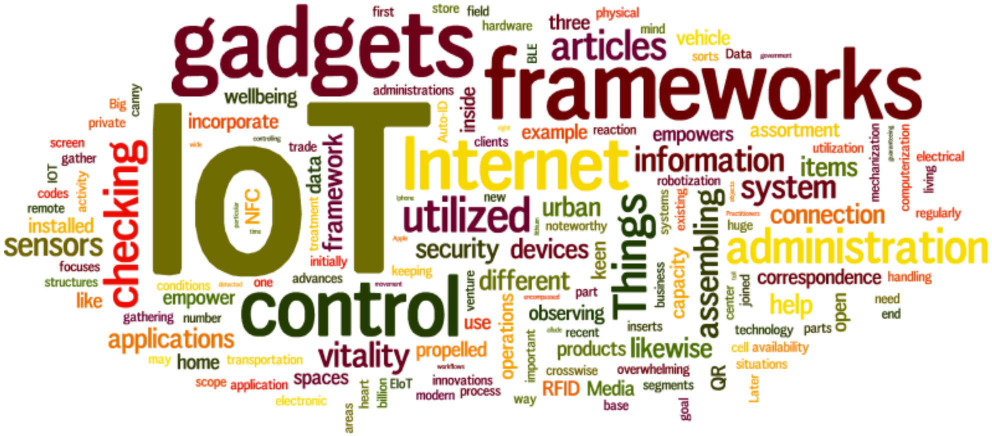
CS578: Internet of Things

Introduction to Arduino Programming



Dr. Manas Khatua

Assistant Professor, Dept. of CSE, IIT Guwahati

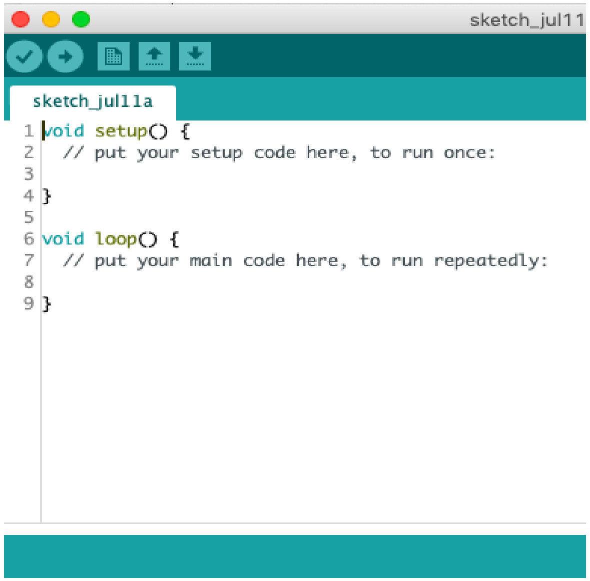
E-mail: manaskhatua@iitg.ac.in

"If you want peace of mind, do not find fault with others. Rather learn to see your own faults." – **Ma Sarada Devi**

Introduction

- The Arduino Software (IDE) allows you to write programs (i.e. **sketches**) and upload them to your board.

- A sketch is consists of two **mandatory** functions:
  - ✓ **Setup( )** -- it is executed **once**
  - ✓ **Loop( )** -- it is executed **repeatedly**



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

- Setup( ) is used for
  - ✓ initialization of serial communication
  - ✓ defining pinMode
  - ✓ declaring variables

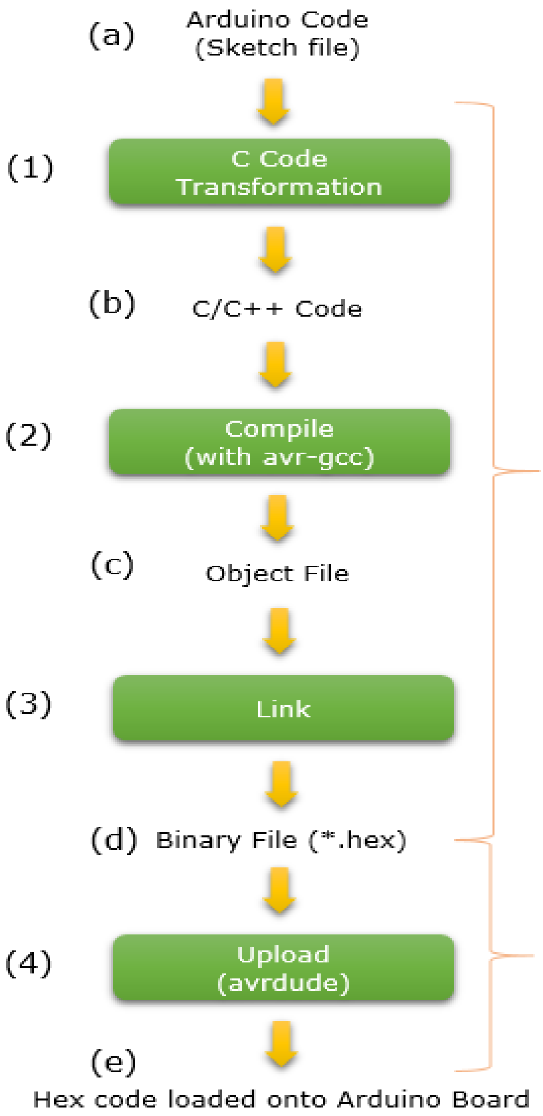
- Loop( ) is used for
  - ✓ writing the main code which has to execute continuously.
  - ✓ e.g. reading inputs from the sensors, triggering outputs to the external device, etc.

Cont…

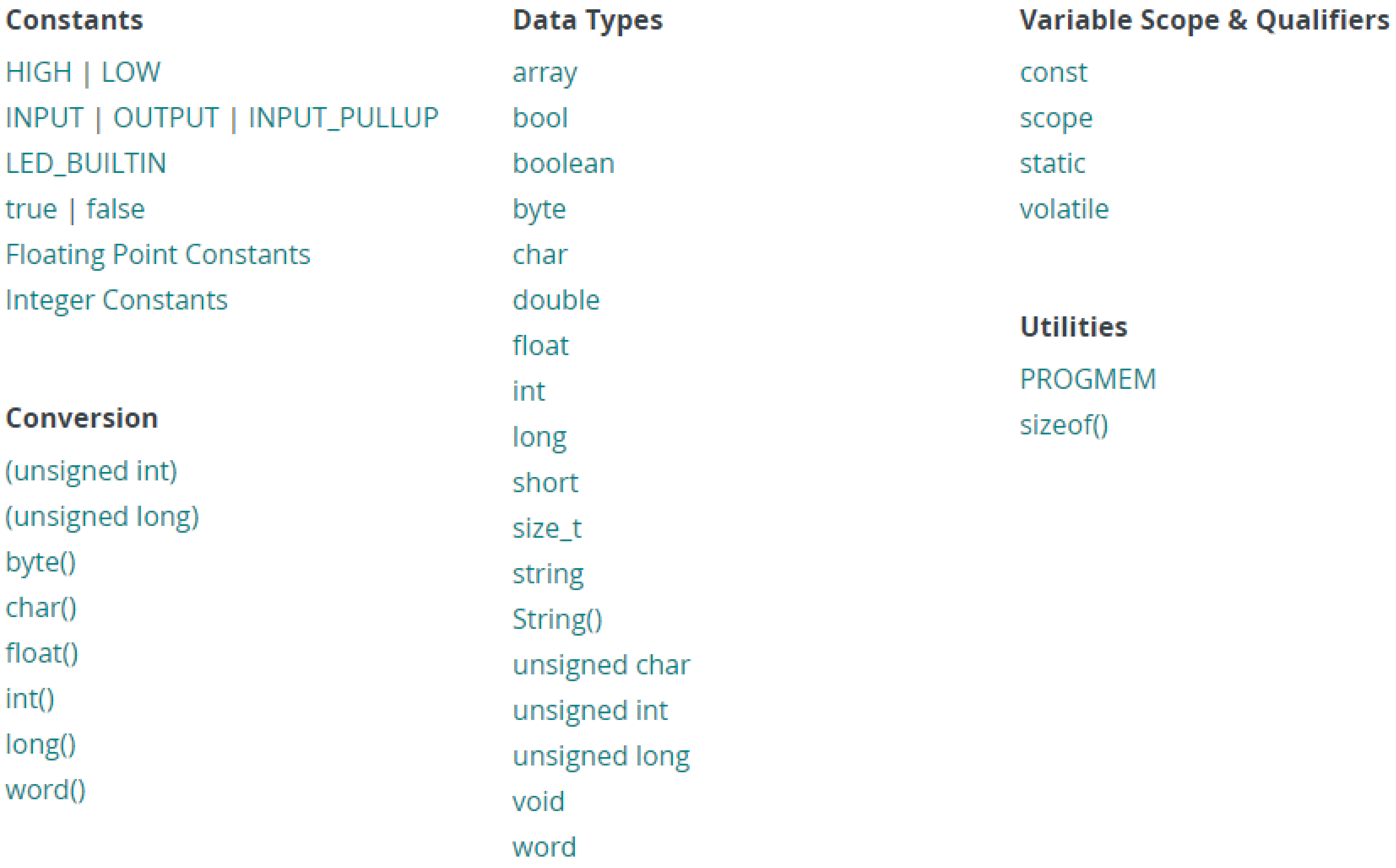
- **Sketches** are **compiled** by avr-gcc / avr-g++
  - It is based on C/C++ programming language

- So, the program syntax is almost similar to C/C++
  - Supported data types
  - Variables
  - Constants
  - Control structure
  - Looping structure
  - Arrays
  - Strings
  - Function

- One important extension is : **Arduino Libraries**
  - Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc.


(a) Arduino Code (Sketch file)
(1) C Code Transformation
(b) C/C++ Code
(2) Compile (with avr-gcc)
(c) Object File
(3) Link
(d) Binary File (*.hex)
(4) Upload (avrdude)
(e) Hex code loaded onto Arduino Board

Variables



Constants

HIGH | LOW

INPUT | OUTPUT | INPUT_PULLUP

LED_BUILTIN

true | false

Floating Point Constants

Integer Constants

Conversion

(unsigned int)

(unsigned long)

byte()

char()

float()

int()

long()

word()

Data Types

array

bool

boolean

byte

char

double

float

int

long

short

size_t

string

String()

unsigned char

unsigned int

unsigned long

void

word

Variable Scope & Qualifiers

const

scope

static

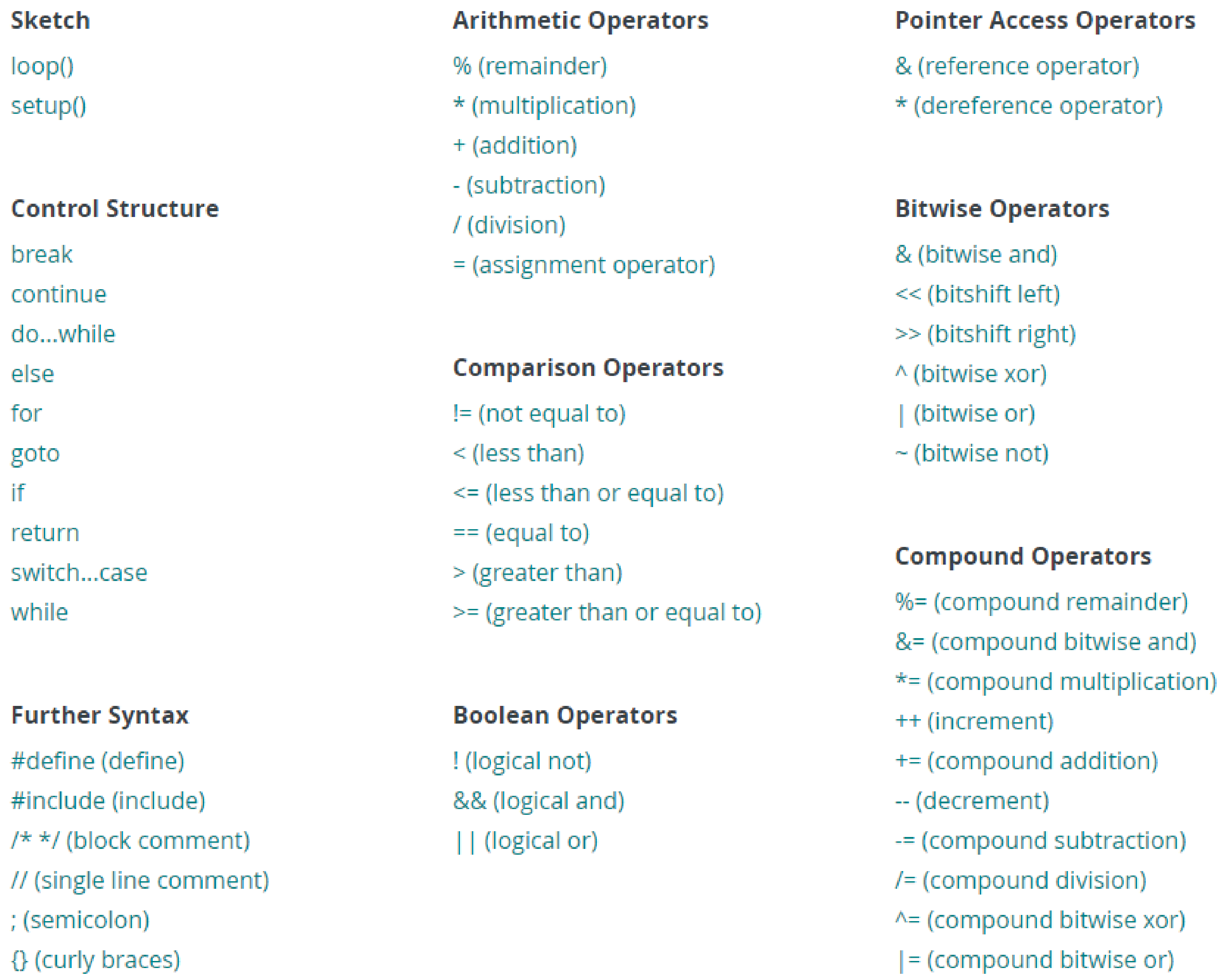
volatile

Utilities

PROGMEM

sizeof()

Operators & Structures



Sketch

loop()

setup()

Control Structure

break

continue

do...while

else

for

goto

if

return

switch...case

while

Further Syntax

#define (define)

#include (include)

/* */ (block comment)

// (single line comment)

; (semicolon)

{} (curly braces)

Arithmetic Operators

% (remainder)

* (multiplication)

+ (addition)

- (subtraction)

/ (division)

= (assignment operator)

Comparison Operators

!= (not equal to)

< (less than)

<= (less than or equal to)

== (equal to)

> (greater than)

>= (greater than or equal to)

Boolean Operators

! (logical not)

&& (logical and)

| | (logical or)

Pointer Access Operators

& (reference operator)

* (dereference operator)

Bitwise Operators

& (bitwise and)

<< (bitshift left)

>> (bitshift right)

^ (bitwise xor)

| (bitwise or)

~ (bitwise not)

Compound Operators

%= (compound remainder)

&= (compound bitwise and)

*= (compound multiplication)

++ (increment)

+= (compound addition)

-- (decrement)

-= (compound subtraction)

/= (compound division)

^= (compound bitwise xor)

|= (compound bitwise or)

Few Built-in Functions

https://www.arduino.cc/reference/en/

- **pinMode (pin, mode)**
  - It configures the specified pin to behave either as input or as output
  - By default the digital pins in Arduino function as input.

  - **pin**: is the number of the pin whose mode needs to be set
  - **mode**: can be INPUT, OUTPUT, INPUT_PULLUP.

```
pinMode(9,OUTPUT);
```

- **digitalReadPin(pin)**
  - Reads the value from a specified digital pin, either HIGH or LOW.

```
val = digitalRead(inPin);
```

- **digitalWrite(pin, value)**
  - Used for output by using the LOW/HIGH logic level (i.e. 0V / 5V)
  - **value**: LOW / HIGH

```
digitalWrite(10,HIGH);
```

- **analogRead(pin)**
  - Access and gets value from a particular Analog pin having 10-bit resolution (i.e. 10-bit ADC)
  - Returns: 0-1023 (integer)
  - Arduino UNO yields a resolution between readings of: 5 volts / 1024 units. It will map input voltages between 0 and the operating voltage(5V or 3.3V) into integer values between 0 and 1023.
  - The input range can be changed using analogReference()

```
val = analogRead(A3);
```

- **analogWrite(pin, value)**
  - Write the analog value (PWM wave) to a pin
  - **value:** it is the duty cycle value between 0 and 255 (as 6 pins).
  - Note: analogRead values go from 0 to 1023, analogWrite values from 0 to 255

```
analogWrite(9, val / 4);
```

Cont…

- **delay(ms)**
  - Pause the program for the amount of time (in millisecond) specified by **ms**

delay(1000); // wait for a second

- **Serial.begin(speed)**
  - It sets the **speed** in bps (baud rate) for serial data transmission from computer to Arduino board

Serial.begin(9600);

- **Serial.available ()**
  - Returns: the number of bytes (characters) available to read

if (Serial.available() > 0) { }

- **Serial.print(value)**
  - Print data to the serial port as human-readable ASCII text
  - Numbers are printed using ASCII character for each digit
  - Floats are printed as ASCII digits (upto 2 decimal places)
  - Bytes are send as a single character
  - Characters and Strings are sent as is.

Serial.print("I received: ");

- **Serial.print(value, format)**
  - The optional 2nd argument specifies the base (format) to use
  - **format:** BIN / OCT / DEC / HEX

Serial.print(i,DEC);
// Print Decimal value of number i

- **Serial.println(value) , Serial.println(value, format)**
  - Additionally it returns the number of bytes written

Cont…

- **Serial.read()**
  - Reads incoming serial data.
- **Serial.write(val) or .write(str) or .write(buf, len)**
  - Writes binary data to the serial port.
  - This data is sent as a byte or series of bytes; to send the characters representing the digits of a number use the print() function instead.
- **Trigonometry:**
  - **cos()**
  - **sin()**
  - **tan()**
- **Math:**
  - **abs()**
  - **max()**
  - **min()**
  - **pow()**
  - **sq()**
  - **sqrt()**
  - **random()**
  - **randomSeed()**

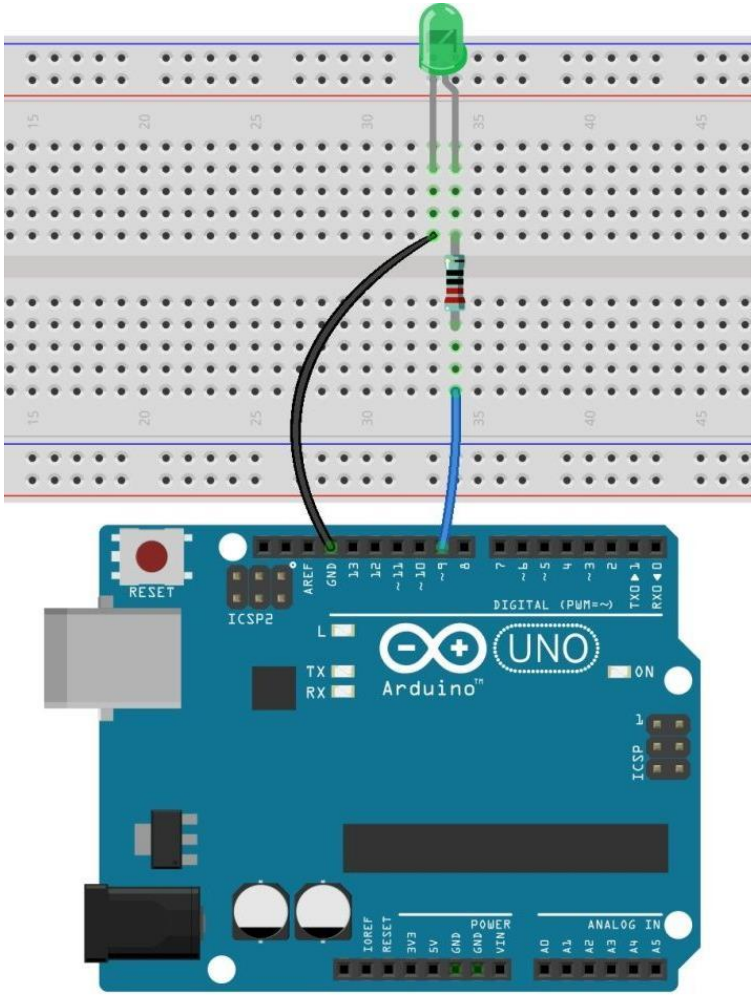
```
incomingByte = Serial.read();
```

```
Serial.write(45); // send a byte with the value 45
```

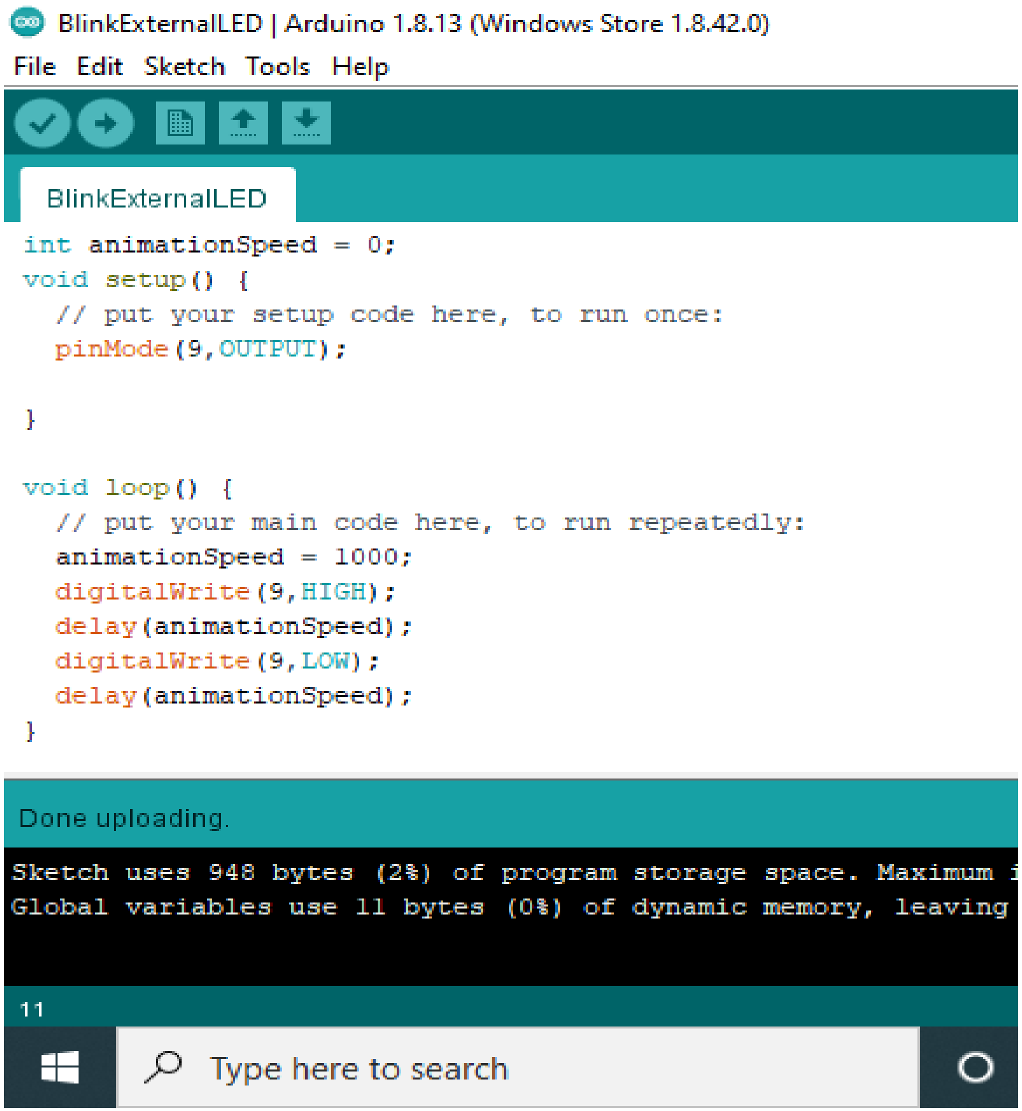
```
int bytesSent = Serial.write("hello");  //send the string "hello" and return the length of the string.
```

Programming Example 1: Blinking LED

- Requirements:
  - Arduino UNO
  - USB connector
  - Breadboard
  - LED
  - 1K Ohm resistor
  - Arduino IDE
- Connection:
  - Place the LED and resistor on breadboard
  - Connect the digital out pin with Anode, and ground with Cathode of the LED
  - Connect the bradboard power with Arduino
  - Connect the Arduino board with PC/Laptop
- Arduino Programming
  - Install IDE in PC/Laptop
  - Run the IDE
  - Select the Arduino board in IDE
  - Select the connected COM port
  - Start writing new sketch



Cont…


BlinkExternalLED | Arduino 1.8.13 (Windows Store 1.8.42.0)
File Edit Sketch Tools Help
BlinkExternalLED
11
Type here to search

```
int animationSpeed = 0;
void setup() {
  // put your setup code here, to run once:
  pinMode(9,OUTPUT);

}

void loop() {
  // put your main code here, to run repeatedly:
  animationSpeed = 1000;
  digitalWrite(9,HIGH);
  delay(animationSpeed);
  digitalWrite(9,LOW);
  delay(animationSpeed);
}
```

Done uploading.

```
Sketch uses 948 bytes (2%) of program storage space. Maximum i
Global variables use 11 bytes (0%) of dynamic memory, leaving
```

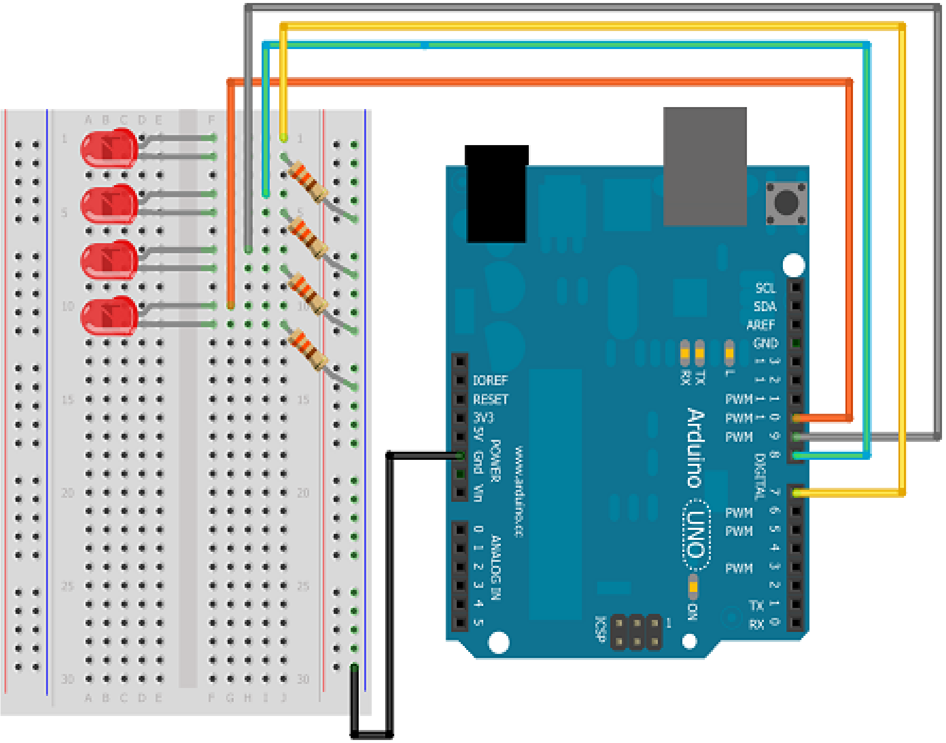
- Write this sketch as shown in the left figure
- Build this sketch and upload in Arduino board
- **OUTPUT**:
  - ➢ LED connected with digital pin 9 will periodically blink.

Example 2: Binary Counter in LED

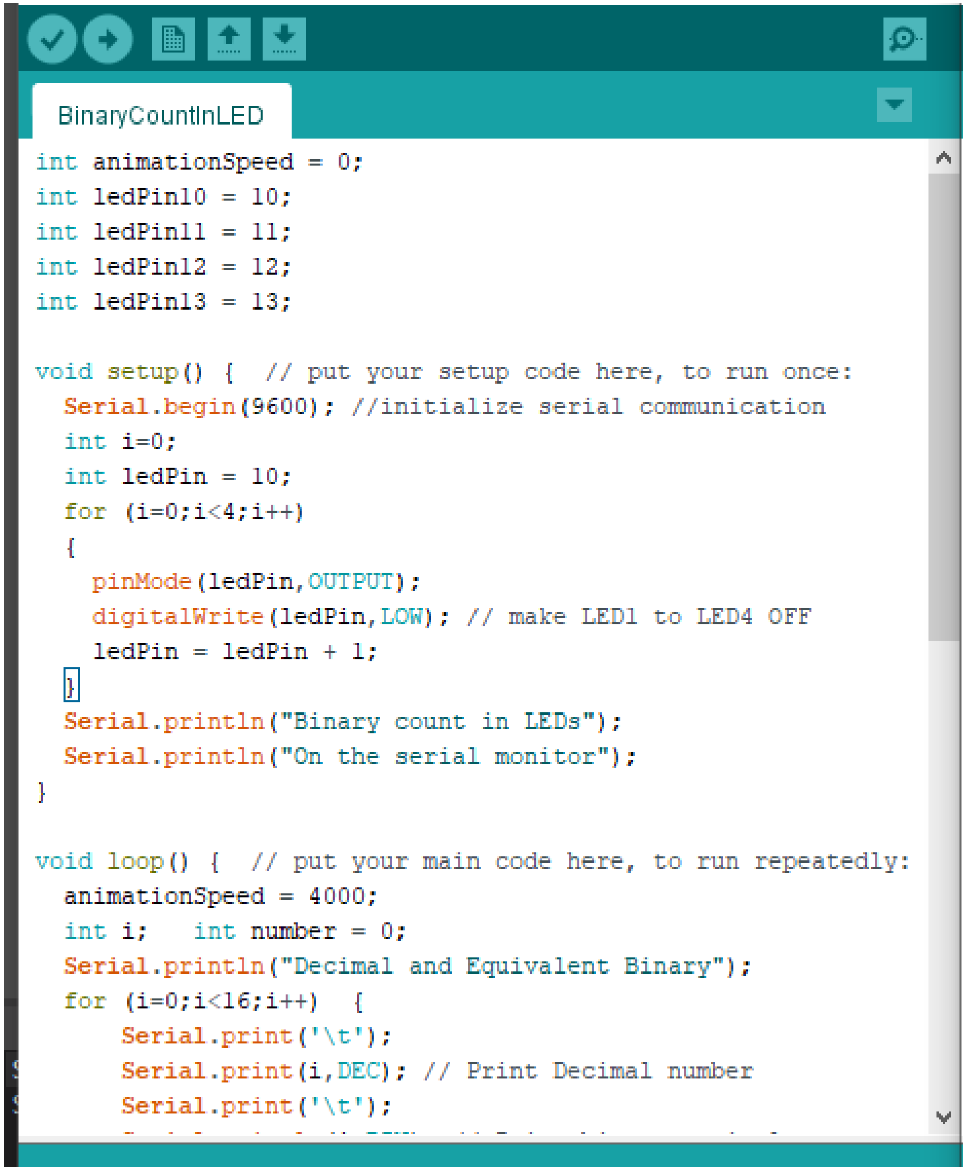
- Requirements:
  - Arduino UNO
  - USB connector
  - Breadboard
  - 4 piece LEDs
  - 4 piece 1K ohm resistor
  - Arduino IDE

- Connection:
  - Place the LED and resistor on breadboard
  - Connect the bradboard power with Arduino
  - Connect the LED with Arduino
  - Connect the Arduino board with PC/Laptop

- Arduino Programming
  - Install IDE in PC/Laptop
  - Run the IDE
  - Select the Arduino board in IDE
  - Select the connected COM port
  - Start writing new sketch



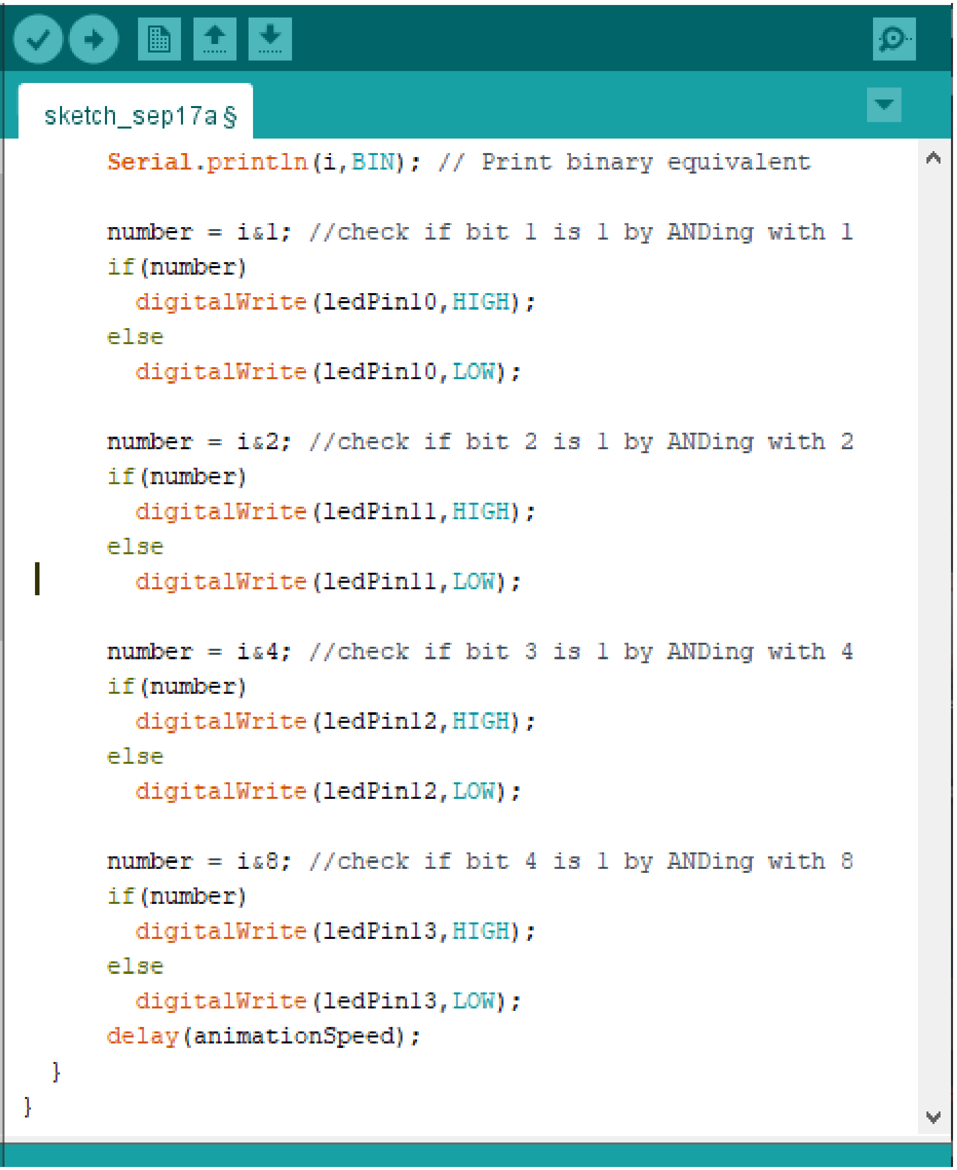
Sketch of Binary Counter



```
int animationSpeed = 0;
int ledPin10 = 10;
int ledPin11 = 11;
int ledPin12 = 12;
int ledPin13 = 13;

void setup() {  // put your setup code here, to run once:
  Serial.begin(9600); //initialize serial communication
  int i=0;
  int ledPin = 10;
  for (i=0;i<4;i++)
  {
    pinMode(ledPin,OUTPUT);
    digitalWrite(ledPin,LOW); // make LED1 to LED4 OFF
    ledPin = ledPin + 1;
  }
  Serial.println("Binary count in LEDs");
  Serial.println("On the serial monitor");
}

void loop() {  // put your main code here, to run repeatedly:
  animationSpeed = 4000;
  int i;   int number = 0;
  Serial.println("Decimal and Equivalent Binary");
  for (i=0;i<16;i++)  {
      Serial.print('\t');
      Serial.print(i,DEC); // Print Decimal number
      Serial.print('\t');
```



```
      Serial.println(i,BIN); // Print binary equivalent

      number = i&1; //check if bit 1 is 1 by ANDing with 1
      if(number)
        digitalWrite(ledPin10,HIGH);
      else
        digitalWrite(ledPin10,LOW);

      number = i&2; //check if bit 2 is 1 by ANDing with 2
      if(number)
        digitalWrite(ledPin11,HIGH);
      else
        digitalWrite(ledPin11,LOW);

      number = i&4; //check if bit 3 is 1 by ANDing with 4
      if(number)
        digitalWrite(ledPin12,HIGH);
      else
        digitalWrite(ledPin12,LOW);

      number = i&8; //check if bit 4 is 1 by ANDing with 8
      if(number)
        digitalWrite(ledPin13,HIGH);
      else
        digitalWrite(ledPin13,LOW);
      delay(animationSpeed);
  }
}
```

Demo on Binary Counter in LED

Live Demo

- See the live demo on
  - Connecting 4 LEDs with Arduino
  - Sketch writing, compiling, uploading and execution

18-09-2020 Dr. Manas Khatua 10

Lessons Learned

- ✓ What is Arduino Programming
- ✓ Syntax of Arduino Programming
- ✓ Supporting variable, structures, operators
- ✓ In-Built Arduino Function Library
- ✓ Programming example - LED blink
- ✓ Program and Demo on binary counter in LED

Thanks!