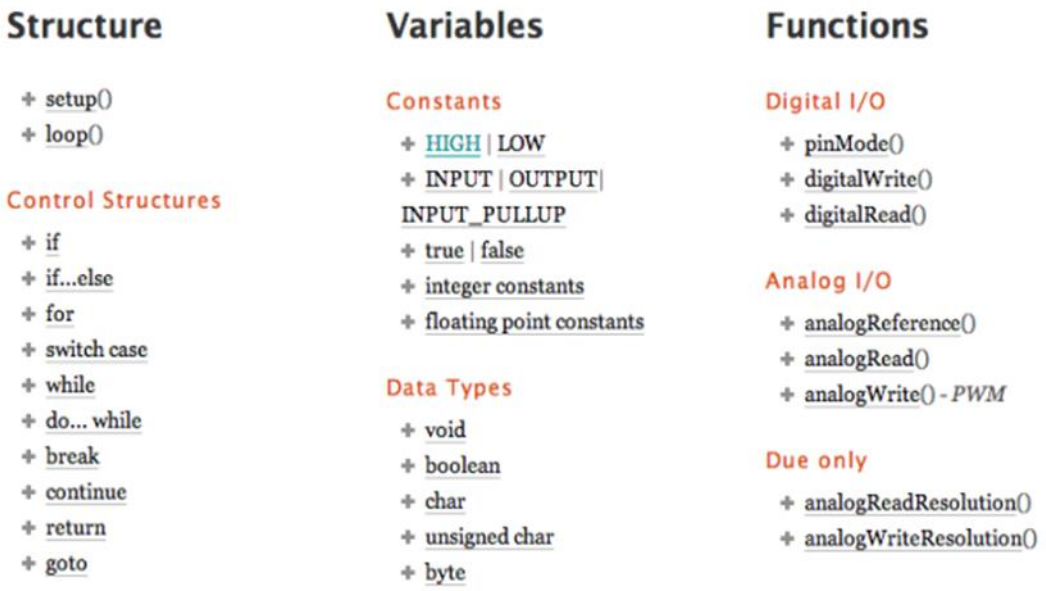
Microcontroller

Lecture 3

Arduino programming

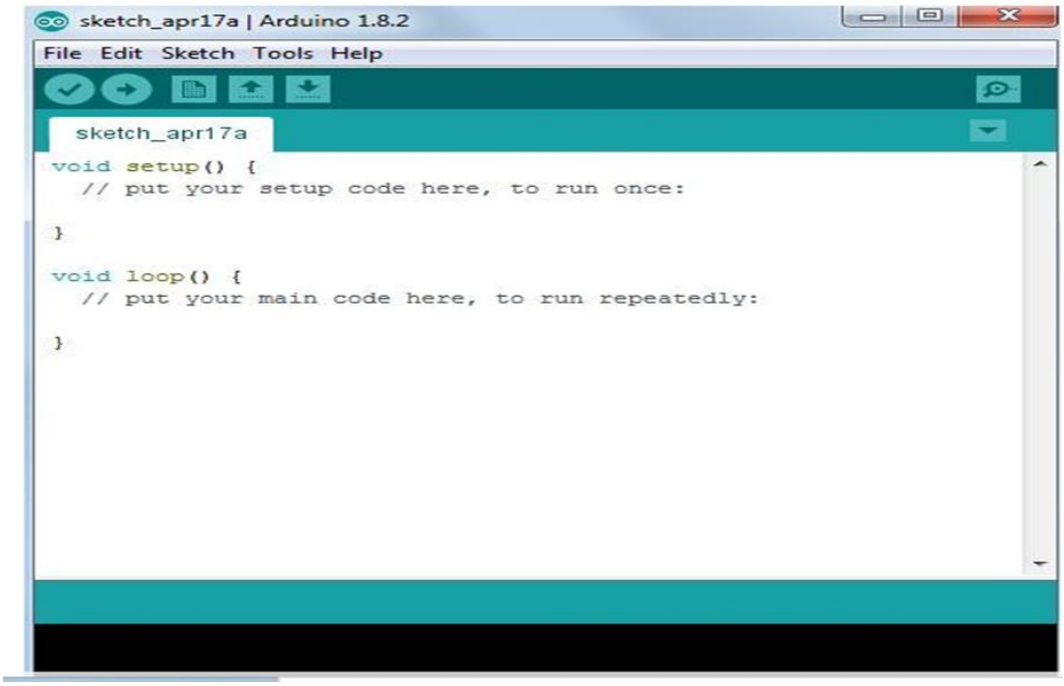
- The Arduino Programming Language is basically a framework built on top of C++.
- A program written in the Arduino Programming Language is called sketch. A sketch is normally saved with the .ino extension (from Arduino).
- The Arduino program can be divided in three main parts: Structure, Values (variables and constants), and Functions.



Structure

Software structure consist of two main functions :

- **Setup( ) function**
- **Loop( ) function**



Setup()

• The setup() function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc.

• The setup function will only run once, after each power up or reset of the Arduino board.

led | Arduino 1.8.13

File Edit Sketch Tools Help

led

```
const int ledpin=12;
const int button=2;
int state =0;
void setup() {
// put your setup code here, to run once:
pinMode(ledpin,OUTPUT);
pinMode(button,INPUT);
digitalWrite(ledpin,LOW);
}
void loop() {
```

Loop()

- After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

Variables

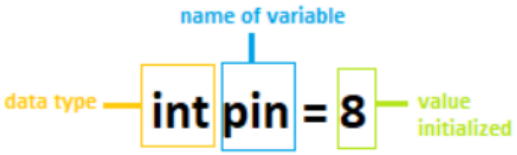
A variables allows you to name and store a value to be used in the future. For example, you would store data acquired from a sensor in order to use

it later. To declare a variables you simply define its type , name and initial value.

There are four common types of variables that are commonly used

```
**********
**********
int temperature =13;
float pin =13.2;
char message ='a';
bool pin = High;
**********
```

Consider the below example:




```
**********
**********
int    such as 5,10,1000
float   such as 5.1 ,10.2
char    such as A,B,F,H
bool    such as true/ false / high/ low /0 /1
**********
```

Constants

Pin level Constants The digital pins can take two value HIGH or LOW.

In Arduino, the pin is configured as INPUT or OUTPUT using the pinMode() function. The pin is further made HIGH or LOW using the digitalWrite() function.

HIGH

The board includes two types of voltage pins to provide HIGH value, which are listed below:

o 5V

o 3V

Some boards include only 5V pins, while some include 3.3V.

Some boards consist of both 5V and 3.3V pins.

For example, Arduino UNO R3. The pin configured as HIGH is set at either 5V or 3.3V.

The pins are configured at the 5V or 3.3V depending on:

- for voltage > 3.0V (presented at 5V pin)
- for voltage > 2.0V (presented at 3.3V pin)

LOW

The pin configured as LOW is set at 0 Volts.

The pins are configured at the 5V or 3.3V depending on:

- for voltage < 1.5V (presented at 5V pin)
- for voltage < 1V (presented at 3.3V pin)

Logical level Constants

The logical level constants are true or false.

The value of true and false are defined as 1 and 0. Any non-zero integer is determined as true in terms of Boolean language. The true and false constants are type in lowercase rather than uppercase (such as HIGH, LOW,

etc.).

Constant Keyword

The name **const** represents the constant keyword. It modifies the behavior of the variables in our program. It further makes the variable as 'read-only'.

The variable will remain the same as other variables, but its value cannot

be changed.

Example Code

const float pi = 3.14;

Example Code

const int a =2;

Functions

The functions allow a programmer to divide a specific code into various sections, and each section performs a particular task. The functions are created to perform a task multiple times in a program.

The function is a type of procedure that returns the area of code from which it is called.

For example, to repeat a task multiple times in code, we can use the same set of statements every time the task is performed.

Function Declaration

The method to declare a function is listed below:

- **Function return type**

We need a return type for a function. For example, we can store the return value of a function in a variable.

We can use any data type as a return type, such as float, char, etc.

- **Function name**

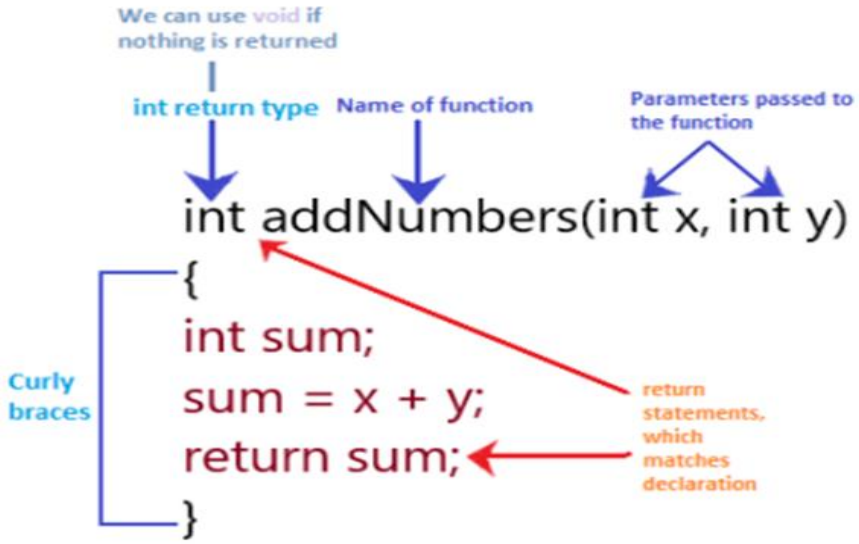
It consists of a name specified to the function. It represents the real body of the function.

- **Function parameter**

It includes the parameters passed to the function. The parameters are defined as the special variables, which are used to pass data to a function. The function must be

followed by parentheses () and the semicolon ; The actual data passed to the function is termed as an argument.

Example: Consider the below image:



pinMode()

[Digital I/O]

Description //Configures the specified pin to behave either as an input or an output.

Syntax// pinMode(pin, mode)

Parameters pin:// the Arduino pin number to set the mode of. mode: INPUT, OUTPUT, or INPUT_PULLUP.

Returns //Nothing