



EXPERIMENT NO: 01

Introduction to the microcomputer system

1.1 OBJECTIVE

This experiment aims to learn about the primary programming language of Arduino Uno and Arduino Uno itself. Also, know about how simple circuit works with Arduino Uno.

1.2 Pre-lab Preparation

- What is a variable in a programming language?
- What is the difference between the global variable and the local variable?
- Write (and run it in your house) a program that will add any two numbers, and it will show the result using a C programming language(Compiler or IDE: Code block or any other compiler)

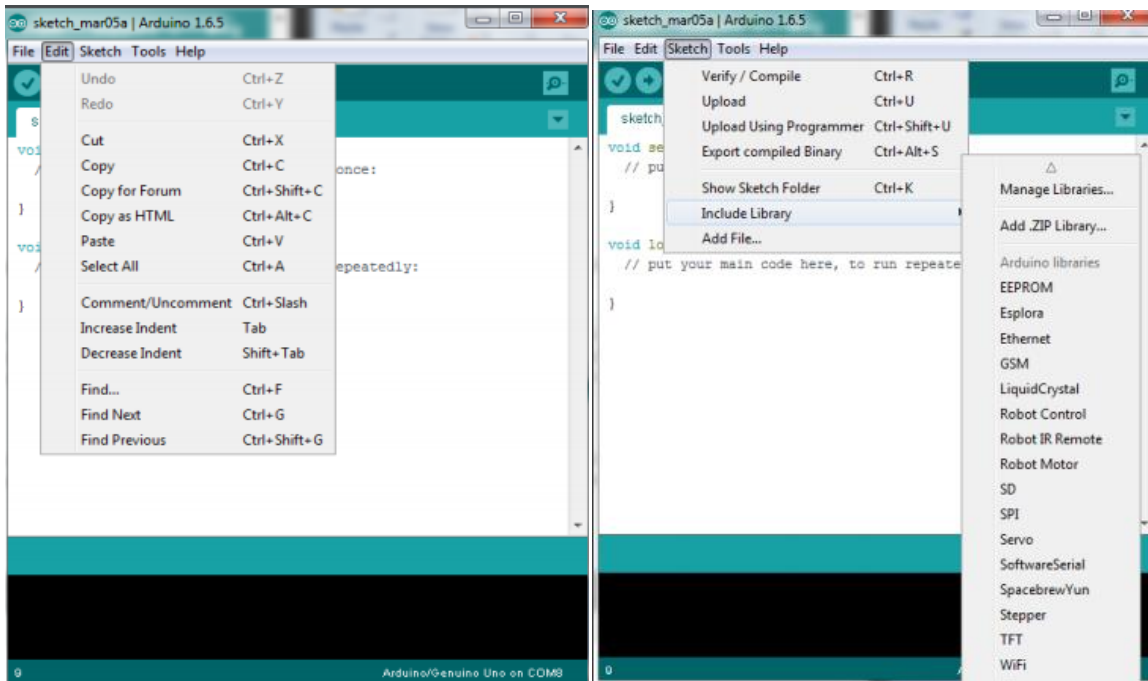
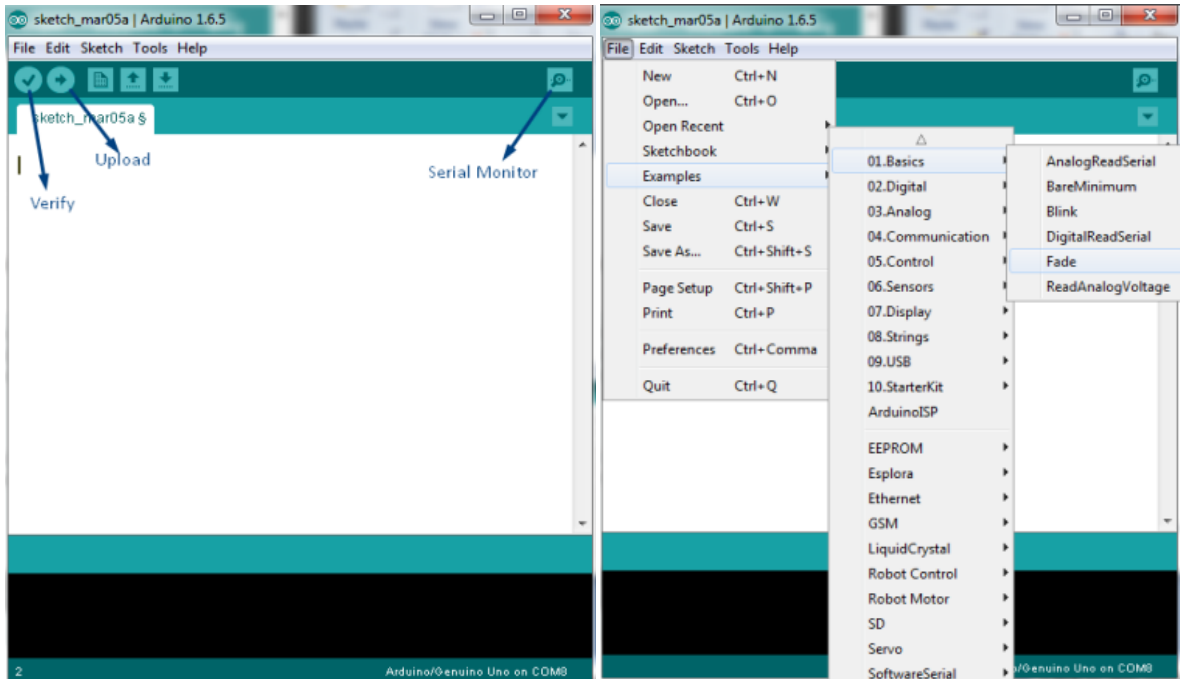
1.3 Equipment

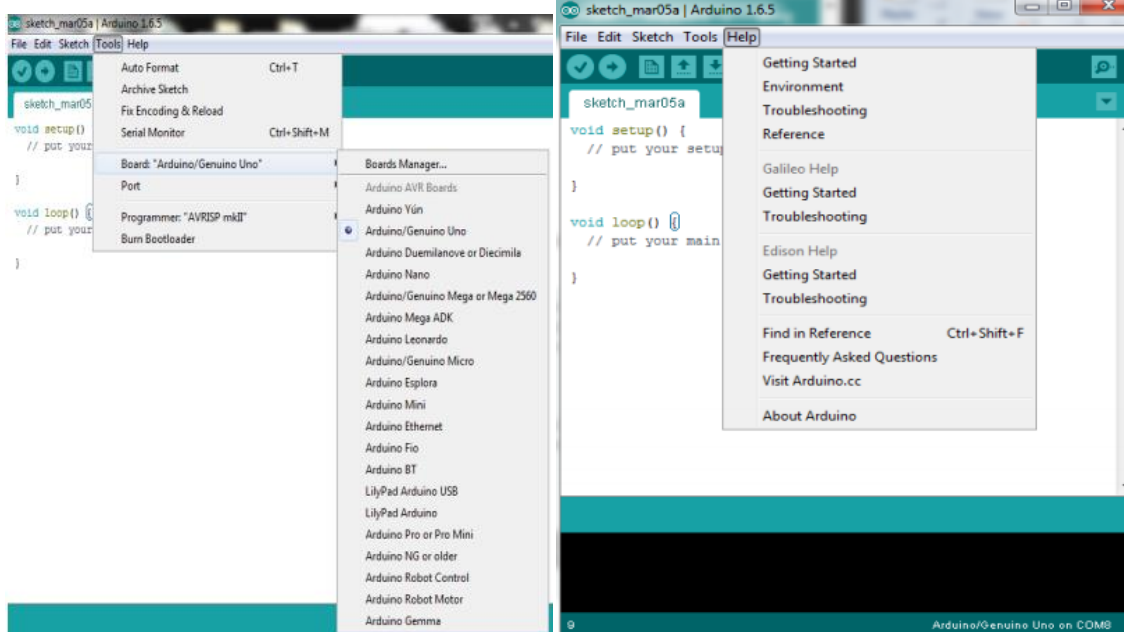
- Arduino Uno(Atmega 328 microcontroller)
- Breadboard
- 220-ohm resistor
- LEDs
- Wires

1.4 Theoretical background:

ATmega 328P microcontroller operates at 5V. The input voltage of Arduino Uno can be significantly from 6V-12V. Arduino itself can supply 5V and 3.3V.

- Total 20 GPIO(or input/output) pins, 14 digital pins, and 6 analog pins
- Output voltage is 0V or 5V(logic 0 or 1)
- Analog pins can read analog data and digital data
- Digital pins can read only digital data
- It is an 8-bit microcontroller
- It has 32 kB of flash memory
- It has 2kB of SRAM
- It is capable of reaching thorough-puts of 1 MIPS per MHz.





```
#include <stdio.h>
```

```
main()
```

```
{
```

```
printf("hello, world\n");
```

```
}
```

```
#include<stdio.h>
```

```
void main() {
```

```
float number1=5.05;
```

```
printf("Number is %f", number1);
```

```
}
```

The functions we are going to use are:

1. pinMode(pin, mode)

pin: the number of the pin whose mode you wish to set

mode: INPUT, OUTPUT

2. digitalWrite(pin, value)

pin: the PIN

value: HIGH or LOW

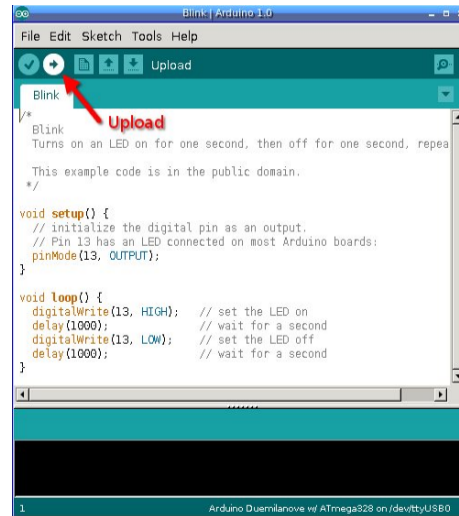
3. delay(ms)

ms: the number of milliseconds to pause

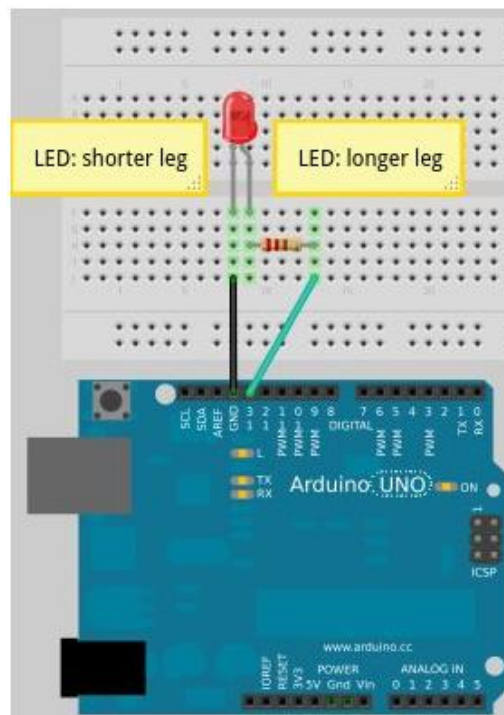
1.5 Procedure:

Step 1: Open the Arduino Uno IDE.

Step 2: Go To File → Example → Basic → Blink and click it. The code can be seen below.



Step 3: Now, build this circuit.



Step 4: Verify and upload this code and observe it.

Step 5: In the program, write “LED_BUILTIN” delete this line only and write here only 13

Step 6: Verify and upload the code and observe it.

Step 7: In the code, increase the delay time like:

`delay(2000);`

or

`delay(5000);`

and so on

Step 8: Verify and upload the code and observe it.

Step 9: In the code, decrease the delay time like:

`delay(200);`

or

`delay(500);`

and so on

Step 10: Verify and upload the code and observe it.

Step 11: Delete all the “delay()” and delete “digitalWrite(LED_BUILTIN, LOW);” line

Step 12: Verify and upload the code and observe it.

Step 13: Delete all the “delay()” and delete “digitalWrite(LED_BUILTIN, HIGH);” line

Step 14: Verify and upload the code and observe it.

1.6 Post Lab Work:

- Which IC is triggering the LED?
- If we increase the delay time of the task-1 program, what will happen?
- If we decrease the delay time, what will happen?
- If we want to “ON” the led nonstop, what will we have to do, or will the program change be changed?
- If we want to “OFF” the led nonstop, what will we have to do, or will they be the program change?