



Embedded Systems **(EPM)**

Lecture (1),(2) Summary

In Arduino we have 2 fundamental functions:

1- void setup()

Any code we want to make in the program will run only one time.

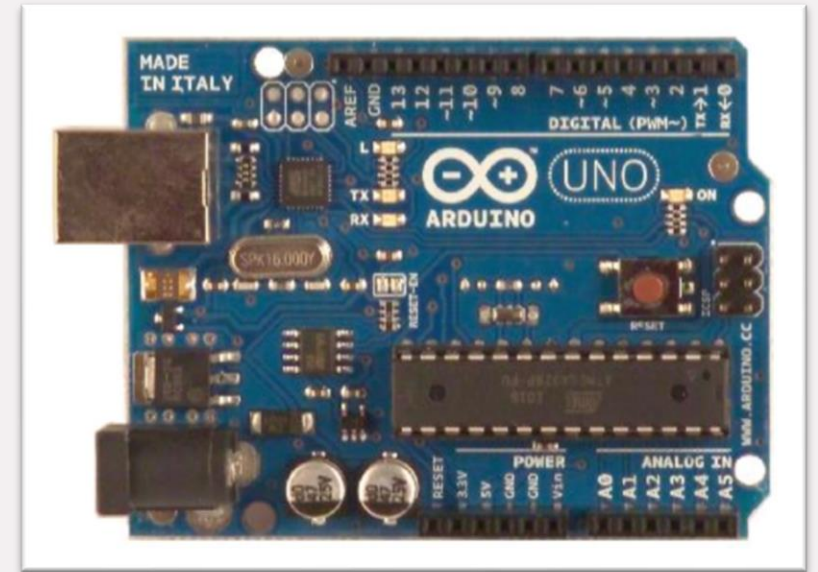
2- void loop()

It repeats the code which is inside it as well as the Arduino Running



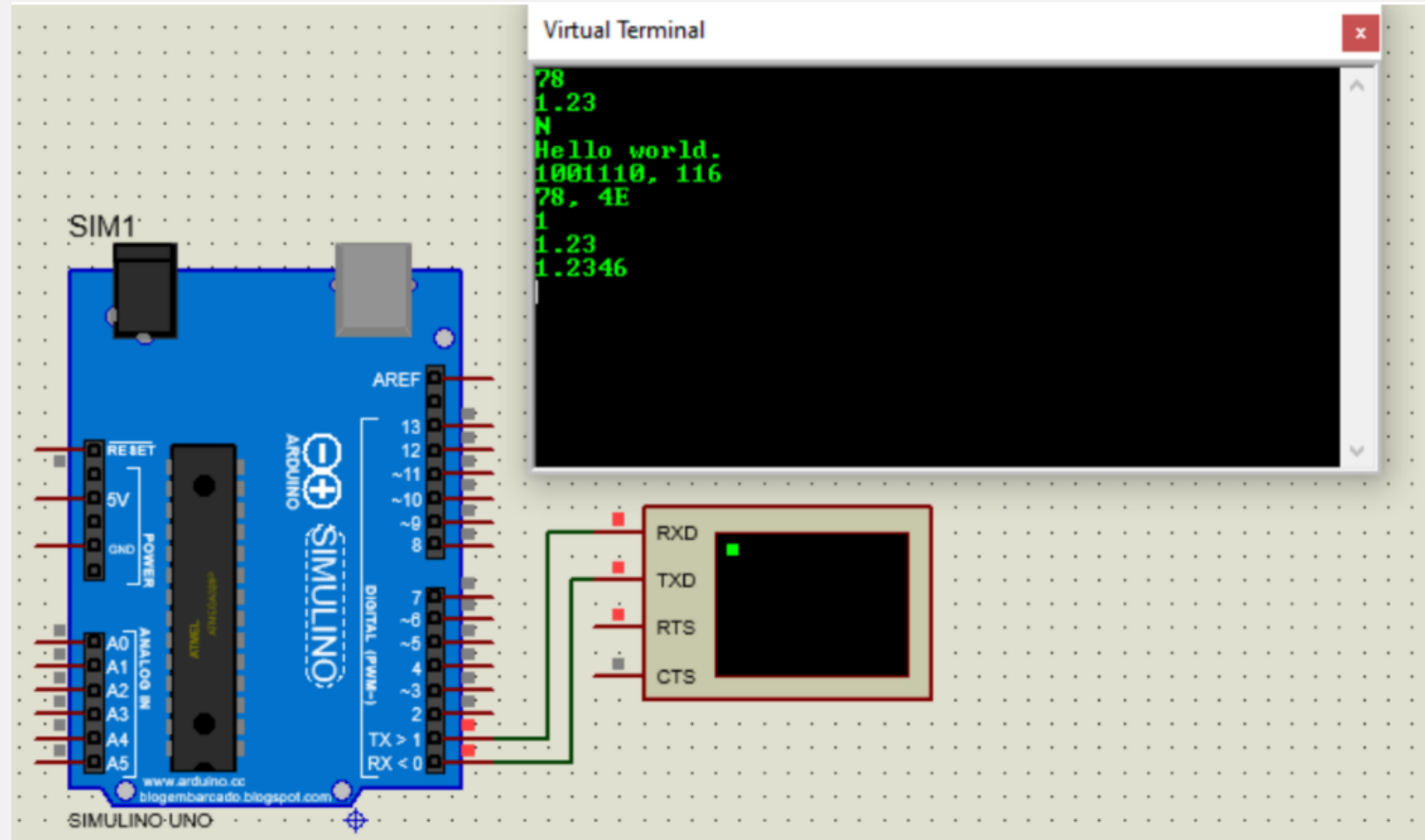
In this Course we will deal with Arduino UNO Board which is Characterized by:

- ATmega328P Microcontroller
- 5V (40 ma for I/O PINs)
- 14 Digital I/O (6 with PWM)
- 6 Analog IP (10 Bit ADC)
- Analog Comparator
- Flash Memory 32KB (0.5K for Arduino Boot-loader)
- SRAM (2KB)
- EEPROM (1KB)
- 16 MHZ
- (2) 8-Bit Real Time Timers , - (1) Watchdog Timer , - (1) 16-Bit Counter
- Communication SPI/I2C/Serial
- USB2Serail to Interface with Computer (Programming and Communication)
- Indicators (LEDs) for Serial Communication PINs (1,2) and PIN 13
- Standalone Power , - Reset Button



Serial Communication:

it makes us to communicate with the chip as we connect a keyboard and mouse to my computer



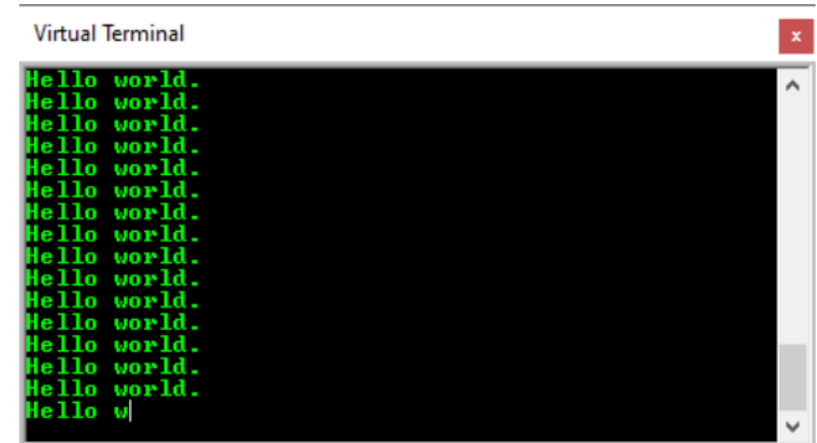
EX(1):

**we want to print Hello World and repeats it all time...
so we runs it in void loop.**

To print Hello World in one line every time we running we use (Serial.println) which is similar like (printf) and (\r\n) in c programming, that prints the word and go to the next line.

Serial.begin(9600): it means we set baud rate = 9600 bits/sec.

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.println("Hello world.");  
}
```



EX(2):

in this example we want to get a character from user and print its ASCII code by using Serial comm.

```
int incomingByte = 0; // for incoming serial data

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

void loop() {
  // send data only when you receive data:
  if (Serial.available() > 0) {
    // read the incoming byte:
    incomingByte = Serial.read();

    // say what you got:
    Serial.print("I received: ");
    Serial.println(incomingByte, DEC);
  }
}
```

Virtual Terminal

```
I received: 49
I received: 50
I received: 51
I received: 13
I received: 52
```

Functions used in Serial Communication:

<code>Serial.begin(9600) ;</code>	تشغيل الشاشة المتسلسلة مرة واحدة في البداية لاحظ: عند تشغيل السيريال لا يمكن استخدام الأطراف 0 و 1
<code>Serial.println(x);</code>	إظهار قيمة (x) على الشاشة ثم سطر جديد
<code>Serial.print("\t \n hello world");</code>	كتابة عبارة (string) \\t : tab \\n : new line
<code>while(!Serial.available()) {}</code>	طريقة لإيقاف الكود في انتظار قيمة من المستخدم
<code>while(Serial.available()==0) {}</code>	
<code>if (Serial.available()>0) {...}</code>	طريقة لاستقبال أي قيمة من المستخدم عبر الشاشة serial monitor المتسلسلة
<code>char x=Serial.read(); if (x=='y'){...}</code>	قراءة بايت من الشاشة المتسلسلة ككود ASCII لمعرفة الرقم المدخل اطرح 48 من القراءة
<code>Serial.parseInt();</code>	لقراءة قيمة ووضعها في متغير int
<code>Serial.parseFloat();</code>	لقراءة قيمة و وضعها في متغير float
<code>Serial.readString();</code>	لقراءة متغير (عبارة) و وضعها في String

Ex(3):

```
void setup() {  
    Serial.begin(9600); // opens serial port, sets data rate to 9600 bps  
}  
  
void loop() {  
    if(Serial.available())  
    {  
        Serial.read();  
        Serial.println(78);  
        Serial.println(1.23456);  
        Serial.println('N');  
        Serial.println("Hello world.");  
        Serial.print(78, BIN);  
        Serial.print(", ");  
        Serial.print(78, OCT);  
        Serial.println("");  
        Serial.print(78, DEC);  
        Serial.print(", ");  
        Serial.print(78, HEX);  
        Serial.println("");  
        Serial.println(1.23456, 0);  
        Serial.println(1.23456, 2);  
        Serial.println(1.23456, 4);  
    }  
}
```

Virtual Terminal



```
78  
1.23  
N  
Hello world.  
1001110, 116  
78, 4E  
1  
1.23  
1.2346
```


Ex(3):

Serial.println(78); : print 78 and go to next line

Serial.println(1.23456); : print 1.23456 and go to next line

Serial.println('N'); : print character N and go to next line

Serial.println("Hello World."); : print Hello World and go to next line

Serial.print(78, BIN); : print Binary value of 78

Serial.print(", "); : print ,

Serial.print(78, OCT); : print Octal value of 78

Serial.println(" "); : print space and go to next line

Serial.print(78, DEC); : print Decimal value of 78

Serial.print(", "); : print ,

Serial.print(78, HEX); : print Hexadecimal value of 78

Serial.println(" "); : print space and go to next line

Serial.println(1.23456 , 0); : print 1.23456 with 0 numbers after , and go to next line

Serial.println(1.23456 , 2); : print 1.23456 with 2 numbers after , and go to next line

Serial.println(1.23456 , 4); : print 1.23456 with 4 numbers after , and go to next line

Data Types:

int – 2 byte signed integer value, Range: -32768 to 32767

unsigned int – 2 byte unsigned integer value, Range: 0 to 65535

long – 4 byte signed integer value, Range: -2147483648 to 2147483647

unsigned long – 4 byte unsigned integer value, Range: 0 to 4294967295

**float /double – 4 byte real value, Range: -3.4028235E+38 to 3.4028235E+38
– Resolution: 3.4028235E-38**

boolean – 1 byte integer value false (0) or true (1)

char – 1 byte signed integer value, Range: -128 to 127

byte – 1 byte signed integer value, Range: 0 to 255

string – C++ class represents array of chars

void

- used by function to indicate no value is returned**
- used by pointers to indicate unknown data type**

Another Examples on C

```
float value = 1.1; <-- Global Variable
void setup()
{
    Serial.begin(9600);
}
void loop()    Every 100ms the program will run and (Variable) will
               decreament (Value --) before going into the conditions
{
    value = value - 0.1;
    if( value == 0)
        Serial.println("The value is exactly zero");
    else if(fabs(value) < .0001)(fabs) is a function which give absolute value
        Serial.println("The value is close enough to zero");
    else
        Serial.println(value);
    delay(100);
}
```

Using Arrays

`int inputPins[] = {2,3,4,5};` define this 4 pins as INPUT

`int ledPins[] = {10,11,12,13};`

`void setup()` define this 4 pins as OUTPUT

```
{
    for(int index = 0; index < 4; index++)
    {
        pinMode(ledPins[index], OUTPUT);
        pinMode(inputPins[index], INPUT);
    }
}
void loop()
{
    for(int index = 0; index < 4; index++)
    {
        int val = digitalRead(inputPins[index]);
        if (val == LOW)
        {
            digitalWrite(ledPins[index], HIGH);
        }
        else
        {
            digitalWrite(ledPins[index], LOW);
        }
    }
}
```

Using Strings

```
String text1 = "Welcome to ";
String text2 = " Arduino";
String text;
void setup()
{
    Serial.begin(9600);

    text = text1 + text2 + " board"; we add the strings to each other
    Serial.println(text);
    Serial.print("Length : ");
    Serial.println(text.length()); text.length() returns the string length

    for(int i=0;i<text.length();i++)
    {
        Serial.print(text[i]); Serial.print(" ");
    }
    Serial.println("");

    text = text.toUpperCase(); text.toUpperCase(), make all letters
    Serial.println(text);         UpperCase

    text = " hello ";
    text = text.trim(); text.trim(), removes the spaces before and
    Serial.println(text); after the string
}

void loop() {}
```