Subject موضوع الدرس Date التاريخ

Section (1)

C/C++ Review

* Variables:

How to declare a variable ⟹ (Data type Name ;)

Data types: int Long double float char Bool Array
2 Byte 4 Byte 4 Byte 4 Byte 1 Byte 1 Byte

(Note) The Previous sizes are in Arduino only.

EX: float a = 1.35;
int x = a;
when we Print the Value of (int x) it equals 1

EX: char b = 'a';
int c = b;
when we Print the Value of (int c) it will Prints the AscII code of ('a') which equals 97

* Arrays: Collection of similar data items stored at contiguos Memory locations.

to define Array ⟹ (Data type Name [ ] ;)
Size (Must be constant)

int A[5]; ⟹ (A[0], A[1], A[2], A[3], A[4])

Data type — Name — size

Subject موضوع الدرس

Examples of how we can initialize the array:

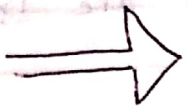
- int A[5] = {1,2,3,4,5};
- int A[] = {1,2,3}; ⟹ Size will equals the number of elements
- int A[5] = {1,2,3}; ⟹ A = {1,2,3,0,0};

* 2D - Array:

int X[3][3];

Number of Rows In Matrix

Number of Coulmns in matrix



X[0][0]	X[0][1]	X[0][2]
X[1][0]	X[1][1]	X[1][2]
X[2][0]	X[2][1]	X[2][2]

* If condition:
Used to decide if a certain statement or block of statements will be excuted or not.

```
If (condition)
{
   ---------
   ---------
}
else
{
   ---------
   ---------
}
```

Subject موضوع الدرس Date التاريخ

* Logical operators :

AND ⟶ &&

OR ⟶ ||

NOT ⟶ !

If A and B are two variables

(A && B) will be TRUE when Both A, B are true.
(A || B) will be TRUE when one of A or B are true.
(!A or !B) will be TRUE when the variable is False.

Ex : write an Embedded Prog. (0 ⟶ 10, Cold) , (10 ⟶ 20, Moderate)
(20 ⟶ 30, hot) for air temPrature.

```
float temP;

if (temP>0 && temP<=10)
  {
    Serial.PrintIn (" Cold ");
  }
else if (temP>10 && temP<=20)
  {
    Serial.PrintIn (" Moderate");
  }
else if (temP>20 && temP<=30)
  {
    Serial.PrintIn (" Hot");
  }
else {
      serial.Print In (" Error");
      }
```

Subject

* Loops
  - For loop
  - while loop
  - do... while loop

* For (initialization ; Condition ; Increment or decrement)

Ex:
```
int ledPins[3] = {13, 14, 15};
for (int i=0 ; i<3 ; i++)
{
  PinMode(ledPins[i], OUTPUT);
}
```
step ① → step ② step ③ step ④

Ex
```
int Pins[4] = {1, 2, 3, 4};
```
Input output Input Input

```
for(int i=0 ; i<4 ; i++)
{
  If (i==1)
  {
    PinMode(Pins[i], OUTPUT);
  }
  else
  {
    PinMode(Pins[i], InPuT);
  }
}
```

موضوع الدرس

Date التاريخ

We have 2 keywords helps us in Loops :

1) Break 2) continue

Break : ends the loop immediately when its encountered

Continue : skips the current iteration of the loop and continue the next iteration

Breaks

```
while (testExpression) {
    // codes
    if (condition to break) {
        break;
    }
    // codes
}
```

```
do {
    // codes
    if (condition to break) {
        break;
    }
    // codes
}
while (testExpression);
```

```
for (init; testExpression; update) {
    // codes
    if (condition to break) {
        break;
    }
    // codes
}
```

Continue

```
while (testExpression) {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
}
```

```
do {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
}
while (testExpression);
```

```
for (init; testExpression; update) {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
}
```

Ex: For loop Printing odd numbers from 0 → 10

```
for (int i=0; i<10; i++)
{
  If (i%2 == 0)
  {
    Continue;
  }
  Serial.Println(i);
}
```

---

* while: Allows you to specify that an action is to be repeated while the condition remains true

```
while (Condition)
{
  ----------
  ----------
}
```

Ex:

```
int i=0;

while (i<3)
{
  PinMode(led Pins[i], OUTPUT);
  i++;
}
```

i++; → Don't forget this increament to avoid going into infinite loop :)

Subject موضوع الدرس Date التاريخ

* do ... while : it's similar to while but it excuted at least once because the condition is tested at beginning.

```
do
{
  ---------
}
while (Condition);
```

this semicolon found only in (do...while) NoT in (while) statement

* Functions :

to declerate Function ⟹ (Return type Function name(Data type a, Data type b,))

Ex :

```
int Multiply (int a, int b)        ⟶ Function defenition
{
    int c = a*b;
    return c;
}

Void Setup()
{
  int x = 10;
  int y = 50;
  int z = Maltiply (x, y);         ⟶ Function Calling
}
```

Subject موضوع الدرس Date التاريخ

* Strings:
  its an array of characters

- String name = "Ahmed"; (Don't forget (" "))
  size = 5
- Char name[6] = "Ahmed"; ≡ {'A','h','m','e','d','\0'}
  ↑ null character

* to Put two strings after eachother:

String FName = "Ahmed";
String RName = "Ali";
Concateuation ⟹ (FName + RName) ⟹ AhmedAli
to Make space between the names:
FName + "_" RName ⟹ Ahmed Ali;

* to remove spaces before and after string we use Name.trim()

* to define Constants we have 2 choices:

  * Macros (out side functions)
    ** define PI 3.14

  * Constant (at any Place)
    Const float PI = 3.14;

* Macros are Better than Constants Because Constants stored in the memory but Macros are not.

Subject موضوع الدرس

Date التاريخ

we have 2 functions in Arduino :

① Void setup : Any config 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