EVERYONIX TECHNOLOGIES

www.everyonix.com/blogs/

For everyonix use only. Outside use is not allowed

ARDUINO UNO R3

LIBRARY: -

No library or header file will be used to access these functions.

FUNCTIONS: -

These are all prebuild functions that you can use in your program to control your Arduino board. No header file is required to access these functions.

1. **pinMode()**:- This function is used to define the mode of Arduino pins i.e. input or output. This is used in setup() function of Arduino code.
   syntax:- pinMode(*pinNumber,Mode*);
   eg:- pinMode(13,OUTPUT);

2. **digitalWrite():-** This function is used to define the put the value to a arduino pin. The value can be either HIGH or LOW.
   syntax:- digitalWrite(*pinNumber,Value*);
   eg:- digitalWrite(13,HIGH);

3. **digitalRead():-** This function is used to read the value coming from external sensor or module to Arduino. A integer variable is declared first and then read value is stored in that.
   syntax:- int *variableName*;
   variableName = digitalRead(*pinNumber*);
   eg:- int var1;
   var1 = digitalRead(13);

4. **analogWrite():-** This function is used to write the analog value to a pin. The value can be any in between 0 to 255. 0 is 0V and 255 is 5V.
   syntax:- analogWrite(*pinNumber,Value*);
   eg:- analogWrite(A0,137);

5. **analogRead():-** This function is used to read the value coming from external sensor to Arduino. A floating variable is declared before and then read value is stored in that.
   syntax:- float *variableName*;
   variableName = analogRead(*pinNumber*);
   eg:- float var2;
   var2 = analogRead(A3);

6. **delay():-** This function is used to provide delay in execution of Arduino code. The delay is in milliseconds means 1000 means delay of delay of 1 second. 1000ms = 1s.
   syntax:- delay(*value*);
   eg:- delay(1000);

7. **delayMicroseconds():-** This function is used to provide delay in execution of Arduino code in microseconds.
   syntax:- delayMicroseconds(*Value*);
   eg:- delayMicroseconds(5);

8. **millis():-** This function is used to count the time from when the Arduino started executing its code. Usually the time count is in milliseconds.
   syntax:- unsigned long *variableName*;
   variableName = millis();
   eg:- unsigned long time;
   time = millis();

9. **micros():-** This function is used to count the time from when the Arduino started executing its code. Usually the time count is in microseconds.
   syntax:- unsigned long *variableName*;
   variableName = micros();
   eg:- unsigned long time;
   time = micros();

10. **pulseIn():-** This function is used to read the duration of pulse coming to a pin from external sensor. This function reads either the HIGH pulse or LOW pulse.
    syntax:- unsigned long *variableName*;
    variableName = pulseIn(*pinNumber,value*);
    eg:- unsigned long duration;
    duration = pulseIn(7,HIGH);

11. **shiftIn():**- This function is used to read the data serially which is coming from external modules and sensors in parallel form. i.e. parallel data is read serially. A shift register should be used to read such parallel data serially.
    syntax:- byte *variableName* = shiftIn(*dataPinNumber, clockPinNumber, bitOrder*);
    eg:- byte incoming = shiftIn(2, 8, MSBFIRST);

12. **shiftOut():-** This function is used to put the data serially through a single pin of Arduino to parallel form to any sensor or display device. A shift register should be used to put data serially to parallel form.
    syntax:- shiftOut(*dataPinNumber, clockPinNumber, bitOrder, byteValue*);
    eg:- shiftOut(2, 8, MSBFIRST, 10110011);

13. **abs():**- This function is used to get the absolute value of a number. If number is negative then it gives positive of that number and if number is positive then it gives the number as

it is.
syntax:- abs(x)
eg:- abs(-34) => result = 34

14. **map():**- This function is used to map the value from old lower and upper boundaries to new lower and upper boundaries.
    syntax:- *variableName* = map(*variableName, oldLower, oldUpper, newLower, newUpper);*
    eg:- value = map(value, 0, 1023, 0, 255);
    this will map value from 0-1023 range to 0-255 range and store the result in same variable value.

15. **max():**- This function will give the maximum of two numbers.
    syntax:- *variableName* = max(x,y);
    eg:- var3 = max(30,60); => result = 60

16. **min():**- This function will give the minimum of two numbers.
    syntax:- *variableName* = min(x,y);
    eg:- var4 = min(30,60); => result = 30;

17. **pow():**- This function will calculate the the power of a number.
    syntax:- *variableName* = pow(*base, exponent*);
    eg:- var5 = pow(2,10) => result = 1024

18. **sq():**- This function will calculate the square of a number.
    syntax:- *variableName* = sq(*number*);
    eg:- var6 = sq(3); => result = 9

19. **sqrt():**- This function will calculate the square root of a number.
    syntax:- *variableName* = sqrt(*number*);
    eg:- var7 = sqrt(25); => result = 5

20. **isAlpha():**- This function is used to check whether a character given is alphabet or not.
    syntax:- isAlpha(*char*);
    eg:- isAlpha(ch);

21. **isLowerCase():**- This function is used to check whether a character given is lower case or not.
    syntax:- isLowerCase(*Char*);
    eg:- esLowerCase(ch);

22. **isUpperCase():**- this function is used to check whether a character given is upper case or not.
    syntax:- isUpperCase(*Char*);
    eg:- isUpperCase(ch);

23. **isAscii():**- This function is used to check whether a character given is Ascii or not.
    syntax:- isAscii(*Char*);
    eg:- isAscii(ch);

24. **random():**- This function is used to get random number between a lower boundary(optional) and a upper boundary.
    syntax:- *variableName* = random(*min, max*);
    eg:- var8 = random(20,100);

These are some important function that will be used in Arduino code. However there are some other functions also which will be covered later in modules and sensors.