

Chapter - 4

Building IoT Application :-

* Introduction to Arduino - Arduino एक open source electronic platform है। जो Hardware और software दोनों का use करता है। ये beginners और experts दोनों के लिए suitable है। क्योंकि इसमें program करना easy होता है। और इसमें बहुत सारे sensor और module को जोड़ सकते हैं। Arduino को program करने के लिए आपको Arduino IDE (Integrated development Environment) की requirement होती है। जो की C, C++ जैसे language में लिखे गए code को compile और upload करता है। Arduino के साथ आप आसानी से project बना सकते हैं जैसे की automatic lights, robot etc। इसके लिए आपको basic electronics or programming की समझ होनी चाहिए।

* History of Arduino - Arduino को 2005 में develop किया गया था। जब interactive design institute IUREA के classroom में Italy में पहला Arduino board बनाया गया। Arduino एक open source microcontroller based board है। Arduino का नाम एक pub से लिया गया था। जो IUREA (Italy) में है। जिसका नाम Bar-de-si है। Arduino की development team के सदस्य Massimo banzi, david, gianluca etc थे।

* Why use Arduino -

1. Open Source - Arduino platform open source है। जिससे की arduino software आपको पूरी तरह फुल में मिलता है। और open source hardware मिलता है। Open source hardware means जिसका design सार्वजनिक (Publicly) रूप से available है। ताकि कोई भी इस design के आधार पर hardware का अध्ययन, संशोधन, निर्माण और बेच सके।

2. Cross-platform - Arduino IDE cross platform है। जिसका मतलब है की आप इसे window, MacOS, Linux अन्य OS पर भी run कर सकते हैं। जबकि अधिकांश microcontroller केवल window system को ही support करते हैं।

3. Inexpensive : Arduino board अन्य micro controller की तुलना में सस्ते हैं। Arduino R3 की कीमत itely में 1200 रुपये है। और arduino clone की कीमत 500 Rupee है।

4. Wide Variety - Arduino के बहुत सारे module उपलब्ध हैं। आप अपनी आवश्यकता अनुसार इनकी select कर सकते हैं।

Different type of Arduino board -

- Arduino Uno R3
- Arduino Nano
- Arduino Micro
- Arduino Leonardo
- Arduino Mega 2560 Rev 3
- Arduino Due
- Lilypad Arduino Board
- Arduino Bluetooth
- Arduino Diecimila
- Red board Arduino Board
- Arduino Robot
- Arduino Explorer
- Arduino Ethernet
- Arduino Zero
- Arduino Pro

* Arduino Uno : यह एक प्रकार का micro controller board है। जो ATmega 328P पर based है। और Uno एक itilian word है। जिसका अर्थ 1 है।

* What is Arduino Uno ATmega 328P - ATmega 328P एक प्रकार का signal chip micro-controller है। जो mega AVR family के अन्दर ATMEL के साथ बना है। Arduino Uno 8 bit RISC processor के साथ custom hardware architecture पर based है।
Analog pin - 6 होती है।
Digital pin - 14 होती है।

* Features of Arduino Uno board -

• Microcontroller	ATmega 328p
• Operating voltage	5V
• Input voltage (recommended)	7-12V
• Input voltage (limit)	6-20V
• Digital I/O Pin	14
• PWM digital I/O Pins	6
• Analog Input Pins	6
• DC current per I/O pin	20mA
• DC current for 3.3V pin	50mA
• Flash memory	32KB
• SRAM	2KB
• EEPROM	1KB
• Clock speed	16MHz
• LED - BUILTIN	13

* Power USB - Arduino board को आपके computer से USB cable का use करके operate किया जा सकता है। इसके A to B type USB का use किया जाता है।

* Power and Barrel connector - Arduino barrel jack से जोड़कर direct power supply से operate किया जा सकता है।

* Voltage Regulator - इसका काम Arduino board को फिर गर voltage को control करने के लिए किया जाता है। यह D. Voltage को manage करता है तथा 5V में convert करता है।

* Arduino Reset - इसका मतलब है आप arduino board को reset कर सकते हैं और और अपने Program को starting से start कर सकते हैं। इसके लिए आप board पर मौजूद reset button या reset pin का use कर सकते हैं।

* Crystal oscillator - इसका कार्य time को calculate करने के लिए किया जाता है।

* power pins -

1. Vin pins (voltage pins) - इस pin के जरिए Arduino board को external power से जोड़ा जा सकता है। इसे arduino board को run करने के लिए 9 Volt की requirement की होती है।

2. **5V pin (5 volt)** - इस Pin के जरूरत 5V input ली जा सकती है या फिर output भी दी जा सकती है। इसमें 5V से अधिक input voltage नहीं दी जा सकती। अलग 5V से ज्यादा होती है तो micro-controller खराब हो जाएगा या board ही जल जाएगा।
3. **3.3V pin** - 3.3V pin का use sensor और module के लिए किया जाता है।
4. **Analog input pin** - इस board में 6 analog input pin होती है। A0 से A5 तक जो analog signal को sensor से read कर सकती है और उन्हें digital value में convert कर सकती है जिससे micro-controller द्वारा इसे पढ़ा जा सके। इन pins को आवश्यकतानुसार digital pins में भी convert किया जा सकता है।
5. **ICSP pins (In-circuit serial programming)** - इसका use आप कर सकते हैं जब आप कोई ऐसा project बना रहे हों जिसमें आपको रोक से ज्यादा Arduino board की requirement हो। और आप चाहते हों कि board आपस में communication करे।
- * **Power LED indicator** - जब Arduino को power source से plug in तो इस LED indicator में light आ जाती है।

- * TX and RX LED - Board में दो level होते हैं
TX (Transmit) और RX (Receive)
TX LED serial data को send करते समय
blink करती है। तथा RX LED receive करते
समय blink करती है।
- * Digital pin - Arduino UNO board में 14 digital
input/output pin होती है।
इसमें 0 और 1 वाली pin से signal
transmit या receive होते हैं।
इसमें 3 ground pins होती है।
- * Arduino Nano - यह एक छोटा सा board होता है
जो ATmega328P या ATmega629
जैसे microcontroller पर based है। यह Arduino
Uno board के जैसा ही है। इस board में मुख्य
रूप से 8 Analog pins, 14 digital input/output
pins, 6 power pins and 2 reset pins होती है।
- * Arduino Micro - यह board मुख्य रूप से ATmega3204
based microcontroller पर based है।
इसमें 20 pins का set होता है।
- * Arduino die - यह Arduino board arm cortex
micro controller पर depend करता है।
इस board में 54 digital input/output pins तथा
12 analog pins होती है। यह board 3.3V पर
कार्य करता है।

* Arduino bluetooth atmega - यह मुख्य रूप से atmega 328P microcontroller पर based है। इस board की Arduino BT भी कहा जाता है। इस board में 16 digital pins तथा 6 analog pins होती हैं। इसमें arbuilt bluetooth होता है।

* Arduino Shields - यह एक Pre-built circuit board है जिसका use कई arduino board को connect करने के लिए किया जाता है। इस shield का use internet से जुड़ने, wireless communication करने, LED screen को control करने etc के लिए किया जाता है। ये shield विभिन्न प्रकार की हो सकती है। जैसे - GSM shield, Ethernet shield, LCD shield, Bluetooth shield etc.

* Introduction to Arduino IDE यह एक ऐसा software है जो developers को program code develop, run, test, debug करने की सुविधा प्रदान करता है। Arduino IDE एक open source software है जिसका use Arduino board के लिए program code लिखने तथा code को board पर upload करने के लिए किया जाता है। Arduino IDE में available text editor पर Arduino program code लिखा जाता है। इस program को .ino file extension के साथ save किया जाता है। इसमें लिखे गए code को sketch करता है। Arduino IDE पर मौजूद विभिन्न toolbar button आपको program को verify करने, upload करने, sketch बनाने, file को open करने, save करने और serial monitor open करने की अनुमति देता है। Arduino IDE में msg area आपको program को save और export करते समय program में मौजूद error को दिखाता है और आपको feedback देता है।

* Interface of Arduino IDE - आप Arduino software को उसकी official website www.arduino.cc पर जाकर download कर सकते हैं। यदि आप यह application window के लिए download करना चाहते हैं तो यह सुनिश्चित करें कि आपके पास window 8, 10 या 11 हो। इसमें सबसे ऊपर menu bar होता है जिसमें file, edit, sketch, tool, Help options होते हैं। Arduino IDE software में लिखे गए sketch को upload करने के लिए इसे Arduino board को IDE से जोड़ना होगा। Arduino IDE के विभिन्न भाग होते हैं -

1. Window Bar

Menu Bar

Shortcut buttons

Text editor

Output panel (Serial monitor)

Window Bar - Arduino IDE का window's bar current file का नाम और software version का नाम प्रदर्शित करता है।

Menu Bar - Arduino ID के menu bar में 5 menu होती हैं।

File menu

Edit menu

Sketch menu

Tools menu

Help menu

(i) File menu - इसकी command निम्न प्रकार है -

- For new sketch - $ctrl + N$
- New cloud sketch - $Alt + ctrl + N$
- Open - $ctrl + O$

• Sketchbook - यह book वर्तमान sketch को दिखाती है।

• Examples - इसके अन्दर कुछ basic sketch दिए गए हैं।

- Close - $ctrl + W$
- Save - $ctrl + S$
- Save as - $ctrl + shift + S$

• Preferences - ($ctrl + comma$), - इस command का use software की setting को change करने के लिए किया जाता है।

• Quit - $ctrl + Q$ (पूरे software को बन्द करने के लिए)

(ii) Edit Menu - इसकी command निम्न प्रकार है -

- Redo - $ctrl + shift + Z$
- Go to line - $ctrl + L$
- Copy for forum (markdown) - $ctrl + shift + C$
- Comment / uncomment - $ctrl + /$

• Increase indent - ($ctrl + +$) - इस command का use selected row के starting में space को add करने के लिए किया जाता है।

- Decrease indent (ctrl + -) - इस command की use space कम करने के लिए।
- Autoformat - ctrl + T
- Increase font size - ctrl + =
- Decrease font size - ctrl + -
- Find - ctrl + f
- Find Next - ctrl + g
- Find Previous - ctrl + shift + g
- Use selection for find - ctrl + E

(iii) Sketch menu -

- Verify / compile - (ctrl + R) - यह ^{sketch} code को compile करता है और कोई error मिलने पर उसे output panel पर प्रदर्शित करता है।
- Upload (ctrl + U) - यह arduino board में code को upload करता है।
- Upload Using Programmer (ctrl + shift + U) - यह programmer का use करके code को upload करता है जो tool menu में available है।
- Export compiled Binary (Alt + ctrl + S) - यह system में .hex file को save करता है।

- Show Sketch folder - (Alt + ctrl + K) - यह command folder को ^{open} show करती है।
Current sketch

- Include library - यह code की starting में # character को जोड़कर library को include करता है।

- Add file - यह command file को sketch में जोड़ती है। और नई file और नई file window में एक new tab में दिखाई देती है।

(iv) Tool menu -

- Auto format - - ctrl + T

- Archive sketch - यह command code को winrar format (.zip) पर में copy करता है।

- Serial monitor (ctrl + shift + M) - यह serial monitor data भेजने और प्राप्त करने की visual communication दिखाता है।

- Board - यह command Arduino board के प्रकार को select करती है।

- Port - यह उस board को select करता है जिससे Arduino connect किया जाता है।

- Burn Bootloader - यह Arduino board के bootloaders को burn करने के लिए use किया जाता है।

Command
Sketch

* Arduino Library - Arduino sketch की कार्यक्षमता को बढ़ाने के लिए library बहुत useful है। इसमें library की एक list है जिसे आप menu bar में sketch menu पर click करके library में जा सकते हैं और add कर सकते हैं।

* Making input output pins -

digitalRead and digitalWrite Command का use Arduino pin को input or output के रूप में select करने के लिए किया जाता है। ये commands case sensitive होती हैं तो इन्हे सही तरीके से लिखने की आवश्यकता होती है।

Example: digitalWrite (13, HIGH)

* How to select the board - sketch को upload करने के लिए आपको उस board को select करना होगा जिस पर आप कार्य करना चाहते हैं। इसमें com1, com2, com3, com4 या अन्य higher serial port होते हैं जो USB Port के Number होते हैं। Board और serial port दोनों को select करने के बाद verify button पर click करें और जैसे ही आप IDE software का uploading button पर click करेंगे वैसे ही recent Arduino module upload होना शुरू हो जाएगा और जब आप code upload करते हैं तो TX और RX LED blink करती हैं।

* Writing and editing codes in Sketch -

जब Arduino IDE open करते हैं तो editor window एक नया sketch शुरू करती है। नये sketch का नाम editor window पर sketch.mmmddxx format में show होता है। जहाँ mmm महीने के लिए 3 अक्षर का संक्षिप्त नाम है। और dd तारीख को represent करता है और x sketch नाम को unique बनाने के लिए एक अक्षर है।
जैसे - sketch-may2016.ino

इसमें हर function के लिए अलग-अलग color codes होते हैं जैसे function के नाम के लिए brown color और text के लिए blue color दिया जाता है जिससे syntax error find करना easy होता है।

* Arduino Sketch structure -

```
void setup() {  
  // put your setup code here, to run once:  
}
```

```
void loop() {  
  // put your main code here, to run repeatedly  
}
```

एक basic arduino sketch में setup और loop के function होते हैं। ये दोनों function default रूप

से arduino IDE window पर दिखाई देते हैं।
Setup के अन्दर लिखा गया code एक बार run होता है तथा loop के अन्दर लिखा गया code infinity times तक run होता है।

* What is function -

1. सभी function का एक unique name होना चाहिए।
जैसे - setup एक unique function है।
2. Function का नाम open और close होने वाले bracket "()" के द्वारा follow किया जाता है। इनके अन्दर कुछ हो भी सकता है और नहीं भी।
3. सभी function में एक written type होना चाहिए।
setup और loop function में void written type है।
4. एक function की body में एक opening और एक close bracket होता है। {}

Example

```
void setup () {  
  Serial.begin (9600);  
  Serial.print ("welcome to IoT");  
}  
void loop () {  
}
```

Sketch को देखने के लिए serial monitor window open करते हैं।

1. **Fault Finding** - Arduino board पर sketch upload करते समय ती यह सुनिश्चित करें की board को select किया गया है या नहीं। Board को select करने के बाद port check करें की वह select है या नहीं।

2. **Serial communication function** - serial.begin (9600) करने के लिए speed सुनिश्चित करता है। serial_monitor create

3. **Serial.print** - इस function का use serial port पर data print करने के लिए किया जाता है। यह integer, float, string को print कर सकता है।

4. **serial.print LN** - इस function के द्वारा किसी भी variable की value को next line में print करने के लिए किया जाता है।

Example - Serial.println ("value");

* **Embedded C language basics** - Embedded system एक ऐसा system है जिसमें hardware and software दोनों शामिल होते हैं जो एक विशेष कार्य को करने के लिए design किये जाते हैं। C language सबसे popular और व्यापक रूप से use की जाने वाली language है। यह syntax और semi syntax का use करते हैं जैसे - main function, data type, variable, loop function etc.

1 Variable declaration -

syntax -: datatype variable = value;

```
int a = 5;
```

```
float b = 4.6;
```

```
char f = "d";
```

2 Rule for naming variable -

(i) Variable को नाम alphabet या underscore से शुरू होना चाहिए।

(ii) इसके नाम में alphabet, digits का use किया जा सकता है लेकिन digits को starting में नहीं लिखा जा सकता।

(iii) इसमें कोई भी special character allow नहीं है। केवल underscore का use किया जा सकता है।

(iv) C language एक case sensitive language है।

(v) इसमें space का use नहीं किया जा सकता।

(vi) variable declare करते हैं समय reserved keyword का use नहीं किया जा सकता।

* Datatype - datatype का use किसी भी variable का type बनाने के लिए use किया जाता है।

1. Void - इसका use उस 3स. function में किया जाता है जो किसी भी value को return नहीं करता।
2. Int - यह -32767 to 32767 value तथा number को store करता है।
3. Unsigned int - यह 0 से 65535 तक value को number में store करता है।
4. long int - यह -2147483295 to 2147483295 तक value को store कर सकता है।
5. Float - Point की value store करने के लिए इसका use किया जाता है। यह data store करने के लिए 4 bit की storage लेता है।
6. Char - यह किसी character को store कर सकता है। यह store करने के लिए 1 Bytes की storage लेता है।
7. Unsigned Char - 0 से 255 तक character को store किया जा सकता है।
8. Boolean - यह datatype की value में से किसी एक को सही मानकर चलता है। यह data store करने के लिए 1 Byte की storage लेता है। इसकी value True or False होती है।

* Operators -

1. Arithmetic operators - Arithmetic operation करने के लिए इनका use किया जाता है।
जैसे - $+$, $-$, $*$, $/$, $\%$

Example -

```
void setup () {  
  Serial.begin (9600);  
  int a = 5;  
  int b = 6;  
  Serial.print (a+b);  
}
```

2. Comparison Operator - या दो से अधिक दो value की तुलना करने में किया जाता है। Comparison operator का use
जैसे - $>$, $<$, $>=$, $<=$, $==$, $!=$

Example -

```
void setup () {  
  Serial.begin (9600);  
  int a = 5;  
  int b = 6;  
  Serial.println (a > b);  
}
```

3. Logical operator - इसमें and, or या not का use करते हैं।

and = &&

or = ||

not = !

दो या दो से अधिक condition को compare करने के लिए logical operator का use किया जाता है।

```
Example - void setup() {  
    serial.begin(9600);  
    int a = 9;  
    int b = 6;  
    int c = 7;  
    serial.println(a > b && a > c);  
}
```

4. Increment and Decrement operator - किसी value में 1 value बढ़ाने के लिए increment operator (++) का use करते हैं। तथा किसी value में से 1 value घटाने के लिए decrement operator (--) का use किया जाता है।

```
Example - void setup() {  
    serial.begin(9600);  
    int a = 9;  
    a++;  
    serial.println(a);  
}
```

5. Assignment operator - value को assign करने के लिए assignment operator का use किया जाता है।

जैसे - =, +=, -=, *=, /=, %=

$a * = b = a = a * b$

```

Example- void setup () {
          serial.begin (9600);
          int a=9;
          int b=3;
          int a+=b;
          serial.println (a);
        }
  
```

6. Bitwise Operator - Bits पर operation करने के लिए इन operators का use किया जाता है
जैसे - &, ! (xor), ! (Not), ^, << (left shift), >> (Right shift)

(i) And (&) operator -

0	0	=	0
0	1	=	0
1	0	=	0
1	1	=	1

```

Example- void setup () {
          serial.begin (9600);
          int a=9;
          int b=3;
          serial.println (a & b);
        }
  
```

(ii) Or (|) operator -

1	1	1
1	0	1
0	1	1
0	0	0

```
Example- void setup () {  
    serial.begin (9600);  
    int a = 9;  
    int b = 3;  
    serial.println (a|b);  
}
```

(iii) Bitwise left shift (<<): value की left की ओर shift करने के लिए -

(iv) Bitwise Right shift (>>): value को right की ओर shift करने के लिए।

* Conditional statement and loops - किसी भी value को या किसी भी कार्य को condition के द्वारा करने के लिए इन statements का use किया जाता है।

1. If statements - Condition True होने पर statement को run करता है और condition False होने पर कुछ भी Print नहीं होता।

Syntax -

```
if (condition) {  
    body of statement
```

```
}
```

Example -

```
void setup () {  
    serial.begin (9600);  
    int a = 7;  
    int b = 5;  
    if (a > b) {  
        serial.print ("a is largest");  
    }  
}
```

2. If-else statement - जब if statement की condition True होती है तो इसका statement run करता है लेकिन यदि condition False होती है तो else statement run करता है।

```
Example - void setup () {  
    Serial.begin (9600);  
    int a = 7;  
    int b = 9;  
    if (a > b) {  
        Serial.print ("a is largest");  
    } else {  
        Serial.print ("b is largest");  
    }  
}
```

3. Else-if statement - एक से अधिक condition देने पर else if का use किया जाता है।

```
syntax - if (condition) {  
    body of statement  
} else if (condition) {  
    body of statement  
} else {  
    body of statement  
}
```

```
Example void setup () {  
    Serial.begin (9600);  
    int a = 7;  
    if (a > 0) {  
        Serial.print ("positive");  
    } else if (a < 0) {
```

```

        serial.print ("negative");
    } else {
        serial.print ("zero");
    }
}

```

Example- void setup () {
 serial.begin (9600);
 int a = 77;
 if (a > 80) {
 serial.print

Example Grad निकालने के लिए -

```

void setup () {
    serial.begin (9600);
    int a = 70;
    if (a > 80) {
        serial.print ("Grade A");
    } else if (a > 60) {
        serial.print ("Grade B");
    } else if (a > 40) {
        serial.print ("Grade C");
    } else {
        serial.print ("Fail");
    }
}

```

Loop in embedded in C language - statement को के लिए loop का use किया जाता है।
repeat करने

1. While loop - While loop एक conditional loop है जो Brackets के अन्दर code को execute करना तब तक जारी रखता है जब तक की गई condition false नहीं हो जाती।

Syntax - while (condition) {
statement block
}

Example - void setup() {
Serial.begin(9600);
int a = 1;
while (a <= 10) {
Serial.print(a);
a++;
}
} 1-10 तक counting के लिए

Example - 2 का table लिखने के लिए -

void setup() { Serial.begin(9600); int a = 1; while (a <= 20) { Serial.print(a); a = a + 2; or a + 2 = 2; } }	OR	void setup() { Serial.begin(9600); int a = 1; while (a <= 10) { Serial.print(a * 2); a++; } }
--	----	--

Question 1-10 तक की संख्या का sum print करें।

```
void setup() {
  Serial.begin(9600);
  int s = 0;
  int a = 1;
  while (a <= 10) {
    s = s + a;
    a++;
  } Serial.print(s);
}
```

2. do while loop - do while loop का कार्य while loop के समान है। do while loop के अन्दर की condition कम से कम एक बार execute अवश्य होगी। इसका कारण यह है की condition starting के ब्याय अंत में check होती है।

syntax -

```
do {
  statement block
} while (condition);
```

Example -

```
void setup() {
  Serial.begin(9600);
  int a = 1;
  do {
    Serial.println();
    Serial.print("welcome");
    a++;
  } while (a <= 10);
}
```

3. **for loop** - इसमें पहले condition check की जाती है इसके बाद कार्य किया जाता है। for loop को चलाने के लिए 3 statement की requirements होती हैं।

- (i) Initialization
- (ii) condition
- (iii) Increment

syntax - `for (start, stop, step) {
statement block
}`

Example - `void setup () {
Serial.begin (9600);
int i;
for (i=1; i<=10; i++) {
Serial.print (i);
}
}`

Question 1 से 50 तक sum संख्या निकालना -

```
void setup () {  
Serial.begin (9600);  
int i;  
for (i=10; i<=50; i+=2) {  
Serial.print (i);  
}  
}
```

Question

40-70 तक की संख्याओं के बीच 7 से विभाजित संख्या Print करें तथा sum को भी।

```
void setup () {
  Serial.begin (9600);
  int i;
  int s = 0;
  for (i = 40; i <= 70; i++) {
    if (i % 7 == 0) {
      s = s + i;
    }
  }
  Serial.print (s);
}
```

* Inbuilt function of digital Input/output -

1. digitalRead () - यह एक digital pin से value को read करती है। यह pin number को एक parameter के रूप में accept करता है। यह High या Low constant return करता है।

2. digitalWrite () - यह digital output pin के लिए एक High या Low value return करता है।

3. pinMode () - यह एक input या output के लिए pin को select करता है। आप इसमें pin number और input या output value parameter के रूप में पास कर सकते हैं।

4. shiftIn - यह data को Bytes में read करता है और एक pin से एक बार में 1 Byte read करता है।

5. shiftOut - यह data को एक Byte write करता है और एक pin से एक बार में 1 Byte write करता है।

6. tone() - यह एक pin से एक square wave पैदा करता है जिसका use buzzer या speaker में किया जाता है।

7. noTone() - यह wave को उत्पन्न होने से रोकता है।

Question for loop से factorial निकालो -

```
void setup () {  
  Serial.begin (9600);  
  int f=1;  
  int i;  
  for (i=1; i<=10; i++) {  
    f=f*i;  
  }  
  Serial.print (f);  
}
```

* Inbuilt Function for Analog input / output

1. `analogRead` - यह analog pin से value को read करता है।

2. `analogWrite` - इस function के द्वारा input के लिए use की जाने वाली voltage को set किया जाता है। इसे 5 प्रकार की voltage set की जा सकती है।

(i) `Default` - default रूप से input के लिए 5V या 3V को set किया जा सकता है।

(ii) `Internal` - 1.1V से 2.5V को set करने के लिए।

(iii) `Internal 1 volt` - केवल Arduino Mega^{board} पर 1V set करने के लिए।

(iv) `Internal 2 volt` - Arduino mega board के लिए 2V set करता है।

(v) `External` - 0 से 5V set करने के लिए।

* Inbuilt Function for Time -

1. `Delay()` - किसी भी कार्य को कुछ समय के लिए रोकने के लिए इसका use किया जाता है। यह milisecond में कार्य करता है।

$$1000 \text{ milisecond} = 1 \text{ second}$$

2. `delayMicrosecond` - `Microsecond` में किसी कार्य को रोकने के लिए।

3. `micros()` - `micros` के द्वारा यह पता लगाया जाता है कि Program कितनी देर में start होगा।

4. `millis()` - यह Arduino board start होने का Time `millisecond` में return करता है।

* Inbuilt function for Maths -

1. `abs()` - यह absolute value return करता है।

2. `Max()` - यह maximum value return करता है।

3. `min()` - यह Minimum value return करता है।

4. `pow()` - यह Power return करता है।

5. `sqrt()` - यह square return करता है।

* Inbuilt function for Character -

(i) `isAlpha` - यह check करता है कि variable के अक्षर alphabet है या नहीं।

2. `isAlphaNumeric` - यह check करता है कि variable में alphabet या numbers है या नहीं।
3. `isASCII` - यह check करता है कि character एक Ascii character है या नहीं।
4. `isDigit` - यह check करता है कि character एक Number है या नहीं।
5. `isGraph` - यह check करता है कि ASCII character print करने योग्य है या नहीं।
6. `isspace` - यह check करता है कि character में space है या नहीं।
7. `isHexadecimal digit` - यह check करता है कि character hexadecimal है या नहीं।
8. `isLowerCase` - यह check करता है कि character lower case है या नहीं।
9. `isUpperCase` - यह check करता है कि character Upper case है या नहीं।

* Inbuilt function for Random Number -

(1) random() - Randomly एक number find करने के लिए इसका use किया जाता है।

* Inbuilt function for Bits and Bytes -

1. Bit - 1 bit प्राप्त करने के लिए इसका use किया जाता है।

2. bitclear - यह एक numerical variable में से 1 bit को clear करने के लिए use किया जाता है।

3. bitRead - किसी number में से 1 bit को read करने के लिए।

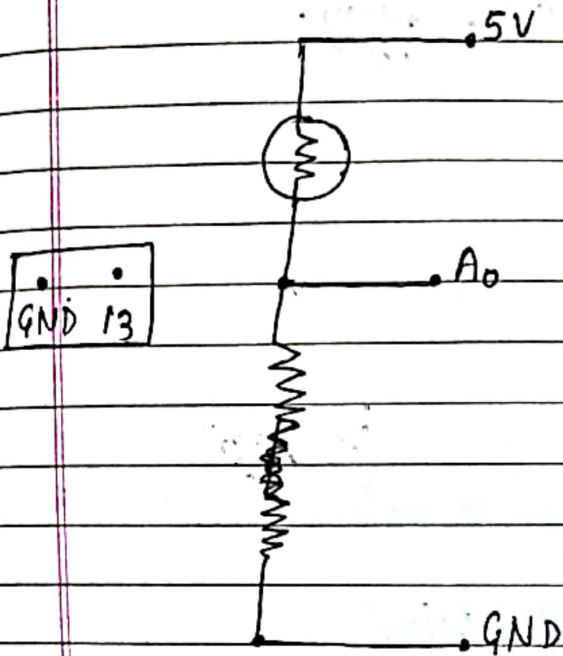
4. bitWrite - किसी भी variable का 1 bit लिखने के लिए।

* Blinking LED program -

```
void setup() {  
  int led = 5;  
  pinMode(led, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(led, HIGH);  
  delay(1000);  
  digitalWrite(led, LOW);  
  delay(1000);  
}
```

* LDR (Light dependent resistor)-



```

int sensorPin = A0;
int led = 13;
void setup ()
{
  Serial.begin (9600);
  pinMode (led, OUTPUT);
}
void loop ()
{
  int value = analogRead (sensorPin);
  if (value <= 100) {
    digitalWrite (led, HIGH);
    Serial.println (value);
    Serial.println ("led on");
    delay (1000);
  }
}

```

```
else {  
    digitalWrite (led, LOW);  
    serial.println ("led off");  
}
```

```
}}
```

• आवश्यक सामग्री -

- (i) Arduino Board
- (ii) LDR
- (iii) 10k Ohm Resistor
- (iv) LED
- (v) Bread Board or Jumper wires
- (vi) Circuit connection

* Circuit connection - (i) LDR का एक terminal 5V की Pin से connect करें।

(ii) LDR का दूसरा terminal resistor से connect करें।

(iii) Resistor का दूसरा terminal ground से connect करें।

(iv) LDR और Resistor के बीच वाले point को analog Pin A0 से connect करें।

(v) LED का एक सिरा ground से connect करें।

(vi) LED का दूसरा सिरा resistor से connect करें और resistor को Pin 13 से connect करें।

* कार्य प्रणाली - LDR का resistance प्रकाश की तीव्रता के according बदलता है। जब प्रकाश की intensity ज्यादा होती है तो इसका resistance कम हो जाता है। Arduino की analog pin LDR से voltage को मापती है। और इसे digital value 0-1023 में convert करती है। Arduino और LDR का use करके प्रकाश की तीव्रता के आधार पर LED को control किया जा सकता है।

* Ultrasonic sensor HC-SR04 -

```
const int trig = 9;  
const int echo = 8;  
long duration;  
int distance distance;  
void setup ()  
{  
  Serial.begin (9600);  
  pinMode (trig, OUTPUT);  
  pinMode (echo, INPUT);  
  pinMode (13, OUTPUT);  
}  
void loop ()  
{  
  digitalWrite (trig, LOW);  
  delayMicroseconds (2);
```

```
digitalWrite (trig, HIGH);  
delay microseconds (10);  
digitalWrite (trig, LOW);
```

```
duration = pulseIn (echo, HIGH);  
distance = duration * 0.0344 / 2;  
Serial.print ("distance");  
Serial.print (distance);  
Serial.println ("cm");  
if (distance <= 30) {  
    digitalWrite (13, HIGH);
```

```
} else {  
    digitalWrite (13, LOW);
```

```
}}
```

• आवश्यक सामग्री -

- (i) Arduino Board
- (ii) Ultrasonic sensor HC-SR04
- (iii) Bread Board
- (iv) Jumper wire

• Circuit connection -

- (i) VCC - Ultrasonic sensor के VCC pin को 5V से connect करते हैं।

- (ii) GND pin - इस sensor की ground pin को arduino की ground से connect करते हैं।
- (iii) Trig pin - sensor की trig pin को Arduino board के digital pin 9 से connect करते हैं।
- (iv) Echo - Ultrasonic sensor को Arduino की pin 8 से connect करते हैं।

कार्यप्रणाली :

- (i) Trig pin - यह एक trigger signal send करता है जो 10 microsecond की high pulse होती है।
- (ii) Echo pin - यह उस समय की गणना करता है जब signal किसी object से टकराकर वापस आता है।
- (iii) Duration - PulseIn function के माध्यम से echo pin पर High signals के time period को मापा जाता है।
- (iv) Distance calculation - $distance = duration * 0.0344 / 2$
यहाँ 0.0344 cm/ms sound waves की speed है।
और 2 से divide करना इसीलिए आवश्यक है क्योंकि signal को वस्तु तक जाने और वापस आने में time लगता है।

* PIR Sensor (Passive Infrared Sensor) -

```
void setup () {  
  Serial.begin (9600);  
  pinMode (13, OUTPUT);  
  pinMode (8, INPUT);  
}  
  
void loop () {  
  int value = digitalRead (8);  
  if (value == HIGH) {  
    digitalWrite (13, HIGH);  
    Serial.println ("hello, i found you");  
  }  
  else {  
    digitalWrite (13, LOW);  
    Serial.println ("I cannot find you");  
  }  
}
```

Components -

- (i) Arduino Board
- (ii) A. PIR sensor
- (iii) Jumper wires

• Connections -

- (i) VCC pin को 5V से जोड़े।
- (ii) Sensor के ground को Arduino board के ground से connect करें।
- (iii) Out pin को Arduino की digital pin 8 से connect करें।
- (iv) LED को pin 13 और ground से connect करें।

• Working

जब PIR sensor motion detect करता है तो serial monitor पर 'hello, i found you' print होता है और LED जल जाती है। अन्यथा 'i cannot found you' print होता है और LED बन्द हो जाती है।

* IR (Infrared) Sensor -

• Components - Arduino Board, IR Sensor, Jumper wire

• Connection -

- (i) VCC को 5V से connect करें।
- (ii) Sensor के GND को Arduino के GND pin से connect करें।
- (iii) Sensor की out pin को pin no 8 से connect करें।
- (iv) LED को Pin No 13 and ground से connect करें।

```
void setup () {  
    pinMode (13, OUTPUT);  
    pinMode (8, INPUT);  
}  
void loop () {  
    int value = digitalRead (8);  
    if (value == HIGH) {  
        digitalWrite (13, HIGH);  
    }  
    else {  
        digitalWrite (13, LOW);  
    }  
}
```

• Working process -

जब आप IR sensor के सामने कोई object लाते हैं तो sensor की output pin '0' return करेगी।
जब कोई object sensor के सामने नहीं होता तब output pin high होगी।

* IOT Project using Arduino and Bluetooth Module to control LED through Android App:-

- Components - Arduino Board, Bluetooth Module, Jumper wire, LED

- Connections - (i) Bluetooth की Vcc pin को 5V से connect करते हैं तथा ground pin को ground pin से connect करते हैं।
- (ii) Bluetooth की RX pin को Arduino की TX pin से connect करते हैं।
- (iii) Bluetooth की TX pin को Arduino की RX pin से connect करते हैं।
- (iv) LED को pin 13 तथा GND से connect करते हैं।

```
char data = 0;
void setup ()
{
  Serial.begin (9600);
  pinMode (13, OUTPUT);
}

void loop ()
{
  if (Serial.available () > 0)
  {
    data = Serial.read ();
    Serial.print (data);
    if (data == '1')
      digitalWrite (13, HIGH);
    else if (data == '0')
      digitalWrite (13, LOW);
  }
}
```

* DHT sensor -

- Components - DHT 11, Arduino board, Jumper wires
- Connections - (i) sensor की VCC pin को 5V से connect करते हैं।
(ii) sensor की Ground pin को Arduino की Ground pin से connect करते हैं।
(iii) sensor की data pin को pin 8 से connect करते हैं।

code

```
#include <dht.h>
int data = 8;
dht DHT;

void setup () {
  Serial.begin (9600);
}

void loop () {
  int value = DHT.read 11 (data);
  float t = DHT.temperature;
  float h = DHT.humidity;
  Serial.print ("Temperature = ");
  Serial.print (t);
  Serial.print ("C");
  Serial.print ((t * 9) / 5 + 32);
  Serial.print ("F");
```

```

serial.print ("Humidity =");
serial.print (h);
serial.println ("%");
delay (2000);
}

```

Working -

DHT 11 sensor एक लोकप्रिय sensor है जो Temperature और Humidity मापने के लिए use किया जाता है। यह sensor 0-50° Celsius तक Temperature माप सकता है तथा 20% से 90% तक Humidity माप सकता है।

* fading LED -

• PWM Pins -

analogWrite () एक pin पर analog मान set करता है।
analogWrite () sets an analog value on a pin

• syntax -

analogWrite (pin, value);

Value - 0-255 (integer)

0: 0% duty cycle (OFF)

64: 25% duty cycle

255: 100% duty cycle (ON)

127: 50% duty cycle

```
code 1 void setup () {  
  }  
  void loop () {  
    analogWrite (11, 0);  
    delay (1000);  
    analogWrite (11, 64);  
    delay (1000);  
    analogWrite (11, 127);  
    delay (1000);  
    analogWrite (11, 255);  
    delay (1000);  
  }  
}
```

```
code 2. void loop () {  
  int i;  
  for (i=0; i<=255; i+=5) {  
    analogWrite (11, i);  
    delay (10);  
  }  
  for (i=255; i<=0; i-=5) {  
    analogWrite (11, i);  
    delay (10);  
  }  
}
```

• Components -

- (i) Arduino Board
- (ii) LED
- (iii) Resistance
- (iv) Jumper wire
- (v) Breadboard

Construction :

- (i) LED की Pin # को Pin 11 से connect करते हैं।
यथा दूसरी Pin को ground से connect करते हैं।

Working :

Arduino के साथ एक LED को fade करनी की
Process fading LED कहा जाता है। इसमें LED की
चमक को धीरे धीरे बढ़ाया और घटाया जाता है।
यह प्रभाव PWM (Pulse width modulation) का
उपयोग करके प्राप्त किया जा सकता है।