**Chapter 1: Introduction to IoT**

**1.1) Identify the requirements for the real-world problems**

**Introduction to Internet of Things**

Internet of Things (IOT) is the networking of physical objects that contain electronics embedded within their architecture in order to communicate and sense interactions amongst each other or with respect to the external environment. In the upcoming years, IOT-based technology will offer advanced levels of services and practically change the way people lead their daily lives. Advancements in medicine, power, gene therapies, agriculture, smart cities, and smart homes are just a very few of the categorical examples where IOT is strongly established.

Over 9 billion Things (physical objects) are currently connected to the Internet. As of now, in the near future, this number is expected to rise to a whopping 20 billion.



Figure 1.1: Introduction to IOT.

**There are four main components used in IOT:**

1. **Low-power embedded systems–**  
   Less battery consumption, high performance are the inverse factors play a significant role during the design of electronic systems.
2. **Cloud computing–**  
   Data collected through IOT devices is massive and this data has to be stored on a reliable storage server. This is where cloud computing comes into play. The data is processed and learned, giving more room for us to discover where things like electrical faults/errors are within the system.
3. **Availability of big data –**  
   We know that IOT relies heavily on sensors, especially real-time. As these electronic devices spread throughout every field, their usage is going to trigger a massive flux of big data.
4. **Networking-connection–**  
   In order to communicate, internet connectivity is a must where each physical object is represented by an IP address. However, there are only a limited number of addresses available according to the IP naming. Due to the growing number of devices, this naming system will not be feasible anymore. Therefore, researchers are looking for another alternative naming system to represent each physical object.

**There are two ways of building IOT:**

1. Form a separate internetwork including only physical objects.
2. Make the Internet ever more expansive, but this requires hard-core technologies such as rigorous cloud computing and rapid big data storage (expensive).

**IOT USAGE IN DEVICES**

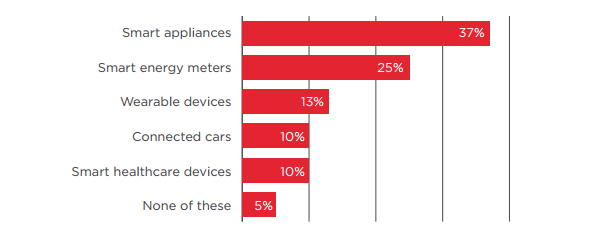


Figure 1.2: GSMA Report.

**Requirements of the real world solved by IOT**

1. **Ability to monitor security** of substations, as well as real-time data on electrical use, and report exceptions so they can be addressed in a timely manner. Ability to solve problems across silos in a utility where IT, Operations and Security don’t talk to each other. Facilitate communications about what’s important.
2. **Any information you need from a device** in order to perform a particular action. Examples provided:
   1. Re-closer on the distribution side of a power line that gets hit by lightning can be closed, checked and reopened by machine after reading the information on the site.
   2. Measure temperature and flow of a pipeline to ensure everything is working as expected or be notified is variances outside the norm are seen.
   3. Solar back-up to devices that may have power or battery issues.
3. **Ability to know how to fix your garage door** because the sensor can tell the company what’s wrong and they can tell you how to fix it. B2B example - a bio-lab is not aware of the volume of enzymes their clients still had on hand. If client ran out, they’d have to put a project on hold for a week or two while more enzymes we made and shipped. They now put one of our devices on every enzyme container so they know which scientist is using which enzyme and when supply is running low. Real-time stock updates enable new bio-lab to optimize the supply chain for their clients.
4. **Communication, collaboration, cohesion and unification of various objects.**Becoming more convenient and connected with the devices we use every day. How to collaborate with devices and work through technology.
5. **Streamlines efficiency and communication of information.**Sensors provide data all the time. Influences decisions by giving you real-time data. Sensors in stores and in manufacturing environments tell you exactly what’s going on and if something is out of the acceptable range you can correct it quickly. Emergency of smaller computers are enhancing communications. We’re taking commodity hardware and optimizing with sensors.
6. All technology starts as a novelty. Becomes a convenience. That’s where IOT is now. **Making it more convenient to control and monitor the 3D printer.** Not yet a pure utility (the end stage for a technology). Ultimately, going forward, things will be built with IOT as a core element, not a special feature. For us, the end point is the printer which can source content from the internet.
7. Industrial internet - **digitization can be applied to the decision making process.**Consumer and healthcare companies will have new apps with connected devices to help save lives and mitigate disasters (e.g., floods, earthquakes).
8. **All IOT solutions solve some problem** - some are smaller, some are bigger. It’s easy to make a lock to connect to the internet. We use cryptography, website and code so the lock doesn’t have to be connected to the internet. Our IOT is not connected to the internet, just the website. As such, the lock cannot be hacked.
9. **Simpler, less expensive home health monitoring** (e.g., scales, blood pressure) to prevent post-surgical events that require return visits to the hospital. Increase adoption and adherence to medication protocols. Opportunity to use data to predict what’s going to happen. Preventive and predictive healthcare.
10. A **connection between humans and computers.** Use Amazon Echo to get all his songs from Amazon Prime and play on demand. Links home management like garage door, lights and HVAC.
11. **Energy saving.** A lot of devices are left on overnight, or longer. Interact with buildings and homes to save energy.
12. **Health devices connected to smart phones diagnose health conditions quickly.** You can take pictures and obtain diagnostics to share with health professionals around the world. Enables the collection and sharing of data in an affordable way. Allows inventors to think about use cases. Digitizes the power grid. Play with how energy is being served. Every device in your house will give you an energy profile. Enterprises will benefit from the digitization of devices and enable the next wave of digitization.
13. Know people that are in the building and **have visibility into what’s going on**. We monitor several thousand conference rooms at Microsoft’s campus to determine if they’re occupied, if A/V is working, what devices people are using, scheduling, booking. We collect data in the cloud to analyze uptime and failure rates. We proactively monitor to see what’s going on. We have statistics about room use and occupancy that will inform and influence the design of the conference rooms on Microsoft’s new campus. In homes, our hub connects all light switches, thermostats, keypads, security system and provides statistics to the cloud so the homeowner can view a dashboard to see how the home is being used. Occupancy use data. We’ll be able to use predictive analytics to make suggestions on how to change the real-time lighting, temp, etc. for your home. Many more touch points - switches, mobile phone, devices, reporting to the cloud versus a single thermostat (Nest) - provide more data for analysis.
14. **Changed the brand cycle.**It used to be 18 to 24 months. Now you must be monitoring social networks to hear what customers are saying and address their concerns or leverage what they are seeing as most beneficial. As John Chambers says, 40% of companies won’t exist in 10 years if they’re not listening to, and responding to the needs of, their customers.
15. **Asset management** - how to engage information to run control systems. Understand the health of the asset producing the work. Know the health and diagnostics of the machine to reduce down time and proactively provide maintenance. Ability to tie the supply chain into the process and provide information back to manufacturing thus reducing costs and expense.
16. **Manufacturers using crowd sourcing to build out their manufacturing floor.** Consumer wireless routers are only secure for a couple of years. Consumer products have a short life expectancy with consumers. Whereas industrial companies need to have an ongoing relationship with their customers since they have service contract and the products often need ongoing service. In healthcare alone IOT has already made incredible contributions saving lives, giving doctors the ability to see a spectrum of health conditions across a large number of people. It will enable more self-care by patients. Clinical trials are now being based on data received from IOT devices thus accelerating time to market. Industrial is incredibly influential because of the buy in from so many big players like IBM, Cisco and GE.
17. **Enables people to try a new approach.** Automate and control things remotely in ways you couldn’t before. Opportunities differ by industry but every industry has many opportunities.
18. We’re at the very early stages but making progress every day. **Getting basic, real-time visibility into places where we haven’t had it before.** For example, we can see a pipeline every half mile and look at KPIs for variances rather than have a human out driving the line and taking measurements. We’ve figured out how to put predictive diagnostics in place. We’re creating a digital twin on the product based on historical performance so we can identify potential needs. IOT provides visibility and reliability where we’ve never had it in the past.

**1.2) Applications of IoT**

**1. IOT Applications – Wearable’s**

Wearable technology is a hallmark of IOT applications and probably is one of the earliest industries to have deployed the IOT at its service. We happen to see Fit Bits, heart rate monitors and smart watches everywhere these days.

One of the lesser-known wearable includes the Guardian glucose monitoring device. The device is developed to aid people suffering from diabetes. It detects glucose levels in the body, using a tiny electrode called glucose sensor placed under the skin and relays the information via Radio Frequency to a monitoring device.



**Figure 1.3:** Guardian glucose monitoring device.

## ****2. IOT Applications – Smart Home Applications****

When we talk about IOT Applications, Smart Homes are probably the first thing that we think of. The best example I can think of here is Jarvis, the AI home automation employed by Mark Zuckerberg. There is also Allen Pan’s Home Automation System where functions in the house are actuated by use of a string of musical notes. The following video could give you a better idea.

## ****3. IOT Applications – Health Care****

IOT applications can turn reactive medical-based systems into proactive wellness-based systems. The resources that current medical research uses, lack critical real-world information. It mostly uses leftover data, controlled environments, and volunteers for medical examination. IOT opens ways to a sea of valuable data through analysis, real-time field data, and testing. The Internet of Things also improves the current devices in power, precision, and availability. IOT focuses on creating systems rather than just equipment. Here’s how an IOT-enabled care device works.

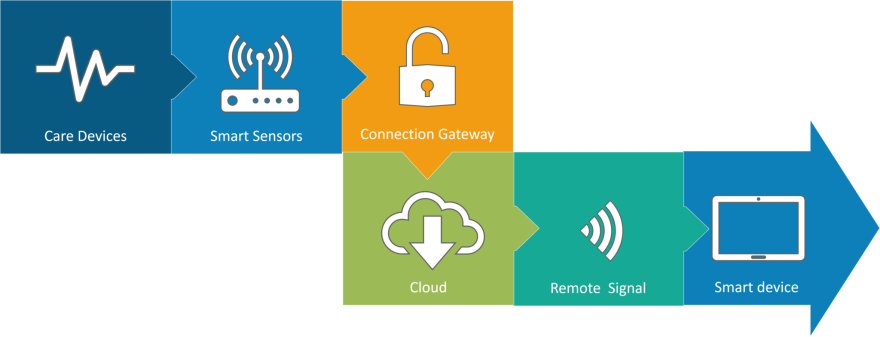


Figure 1.4: IOT enabled care device.

**4. IOT Applications – Smart Cities**

By now I assume, most of you must have heard about the term **Smart City**. The hypothesis of the optimized traffic system as I mentioned earlier, is one of the many aspects that constitute a smart city. The thing about the smart city concept is that it’s very specific to a city. The problems faced in Mumbai are very different than those in Delhi. The problems in Hong Kong are different from New York. Even global issues, like finite clean drinking water, deteriorating air quality and increasing urban density, occur in different intensities across cities. Hence, they affect each city differently. The Government and engineers can use IOT to analyze the often-complex factors of town planning specific to each city. The use of IOT applications can aid in areas like water management, waste control, and emergencies.



Figure 1.5: IOT based Smart City.

## ****5. IOT Applications – Agriculture****

Statistics estimate the ever-growing world population to reach nearly 10 billion by the year 2050. To feed such a massive population one needs to marry agriculture to technology and obtain best results. There are numerous possibilities in this field. One of them is the **Smart Greenhouse**. A greenhouse farming technique enhances the yield of crops by controlling environmental parameters. However, manual handling results in production loss, energy loss, and labor cost, making the process less effective. A greenhouse with embedded devices not only makes it easier to be monitored but also, enables us to control the climate inside it. Sensors measure different parameters according to the plant requirement and send it to the cloud. It, then, processes the data and applies a control action. 

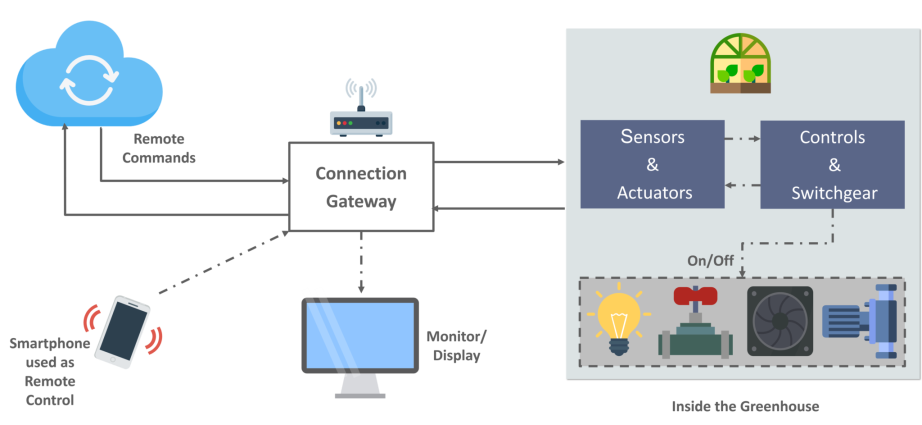


Figure 1.6: Smart Greenhouse.

**6. IOT Applications – Industrial Automation**

This is one of the fields where both faster developments, as well as the quality of products, are the critical factors for a higher Return on Investment. With IOT Applications, one could even re-engineer products and their packaging to deliver better performance in both cost and customer experience. IOT here can prove to be game changing with solutions for all the following domains in its arsenal.

* Factory Digitalization
* Product flow Monitoring
* Inventory Management
* Safety and Security
* Quality Control
* Packaging optimization
* Logistics and Supply Chain Optimization

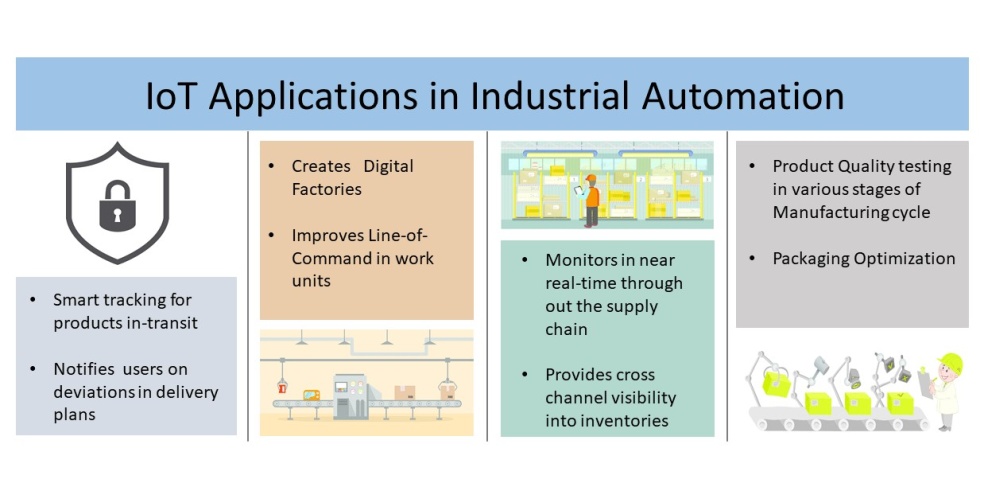


Figure 1.7: Industrial Automation.

**Chapter 2: Introduction to Colour Sorting and Count Machine**

**2.1) Problem definition**

Many times it is seen that in the industry, there is a need to sort products according to colour. For this purpose this project showcases the colour sorting machine. Along with the systematic colour sorting technique, it will also be able to count the number of items produced which is mostly done by other machines in the real world industries. This combination of sensors to implement the colour sorting machine as well as count the number of products that are produced will be a new addition to already existing inventions and can be used as a two in one machine instead of using many bulky, individual and complicated machines.

**2.2) Aims and Objectives**

* The main aim of this project is to sort products according to their colour.
* This will help in an organized segregation of products and simplify further processes involved in manufacturing products.
* In addition to colour sorting, a count of the products that are produced can also be implemented.
* This will ensure a compact method of colour sorting as well as counting the products.
* It is very useful for keeping track of products in large producing industries.

**2.3) Scope**

* Sort objects quickly according to colour.
* It reduces labour cost.
* It reduces manual work.
* It reduces time consumption
* By using IR sensor it can count the number of objects.

**2.4) Features**

* Provides a smooth conveyer for the objects.
* Simplifying the task such as recognizing or differentiating colours.
* Large amount of objects of different colours can be sorted quickly.
* Using servo motors the implementation becomes easy.

**Chapter 3: Review of Literature**

**Introduction to IoT**

The introduction to IoT and the main components of IoT is given in this site. **[1]**

**Applications of IoT**

This site gives the information about the applications i.e. which sectors IoT is used and how it eases a customer’s daily life. **[3]**

**Solving real world problems with IoT**

This site defines what the real world problems are and how it is solved by using IoT. **[2]**

**Introduction to colour sorting**

Why colour sorting machines are more effective than manual sorting mechanisms and how it is useful in industries with bulk production and in industries having repetitive actions is explained. Its applications areas are stated as well. This helped to formulate the problem definition and construct the aims and objectives for this project.

[Extract from site]

As the name suggests, colour sorting is simply to sort the things according to their colour. It can be easily done by seeing it but when there are too many things to be sorted and it is a repetitive task then automatic colour sorting machines are very useful.  These machines have colour sensor to sense the colour of any objects and after detecting the colour servo motor grab the thing and put it into respective box. They can be used in different application areas where colour identification, colour distinction and colour sorting is important. Some of the application areas include Agriculture Industry (Grain Sorting on the basis of colour), Food Industry, Diamond and Mining Industry, Recycling etc. The applications are not limited to this and can be further applied to different industries. **[4]**

**Components required for the implementation of project**

The components required for this project, i.e. Arduino board, 2 Hobbyist Servo Motors, Colour Sensor-TCS3200, power supply as well as the components required for construction of the prototype were given in this article. Along with that, the 3D modelling using Solidworks 3D modelling software was used to show the implementation of the various components, how they work and interact with every other component. The construction of the project as well as a demonstration of how the project works is given in this website.

[Extract from the site]

The working principle is as follows:

* Initially, the colored skittles which are held in the charger drop into the platform attached on the top servo motor.
* Then the servo motor rotates and brings the skittle to the colour sensor which detects its colour.
* After that the bottom servo motor rotates to the particular position and then the top servo motor rotates again till the skittle drop into the guide rail. **[5]**

**The YouTube video of colour sorting machine**

This is the YouTube video linked to the above website which gives us a demonstration of the project, the 3D modelling of the project, how the project is constructed and also how the code runs to give the desired output. **[6]**

**Aims and objectives of colour sorting machine**

This site gives the aims and objectives of a colour sorting machine as well as how the colour sorter is used to segregate items into separate bins. Also the use of colour sorting machine giving the example of its use in the field of candy industries is also explained. **[7]**

**Counting objects using IR Sensor**

How to count objects using IR Sensor, Code for IR Sensor with Arduino and how 7 Segment display is used in this project to display the count of objects is given in this site. **[8]**

**Code and connections for IR Sensor and theory on sensing of obstacles**

## The working of IR Sensor, its various components, its connections and how the code for counting objects using IR Sensor works is shown in this site. [9]

## Scope of colour sorting

## This site gives us ideas for the scope and applications of this project.

## [Extract from site]

## Applications of Arduino based Color Sorting Machine:

* The color Sorting Machine can be used for Industrial purposes, like sorting different industrial parts according to the colors.
* For sorting skittles, colored balls and M&Ms.
* Can be used in Automobile Industries. **[10]**

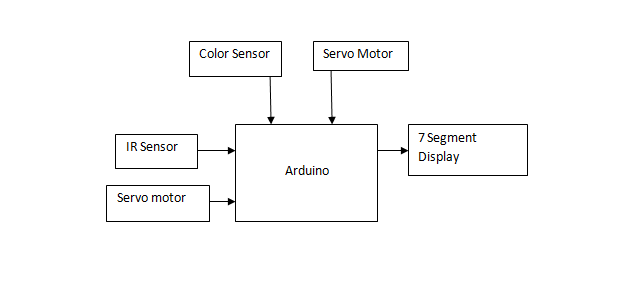
**Working of color sensor TCS3200**

How TCS3200 Color Sensor works, its various components, its pin diagram and its connections with Arduino board is shown on this site. **[11]**

**Chapter 4: System Description**

**4.1 Design**

**Block Diagram:**



Serial Monitor

Figure 4.1: Block Diagram of Color-Sorting Project

The above block diagram describes how the components of this IoT project will be connected. Two sensors will be used, namely the color sensor TCS3200 which will help detect color of the objects and an IR Sensor which will be used to count the number of objects. These two sensors will be used to accept input. The rotation of the objects towards the color sensor as well as dropping the object in its respective containers using a conveyer will be done using a Servo Motor. The output for the count will be displayed on a 7 segment display and the output for the Color Sensor will be displayed on the serial monitor.

**4.2 Hardware, Software and cloud platforms used**

**Components requirements:**

1. Arduino:

The Arduino MEGA 2560 is designed for projects that require more I/O lines, more sketch memory and more RAM. With 54 digital I/O pins, 16 analog inputs and a larger space for your sketch it is the recommended board for 3D printers and robotics projects. This gives your projects plenty of room and opportunities maintaining the simplicity and effectiveness of the Arduino platform. This document explains how to connect your Mega2560 board to the computer and upload your first sketch. The Arduino Mega 2560 is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards and running both online and offline. For more information on how to get started with the Arduino Software visit the Getting Started page.

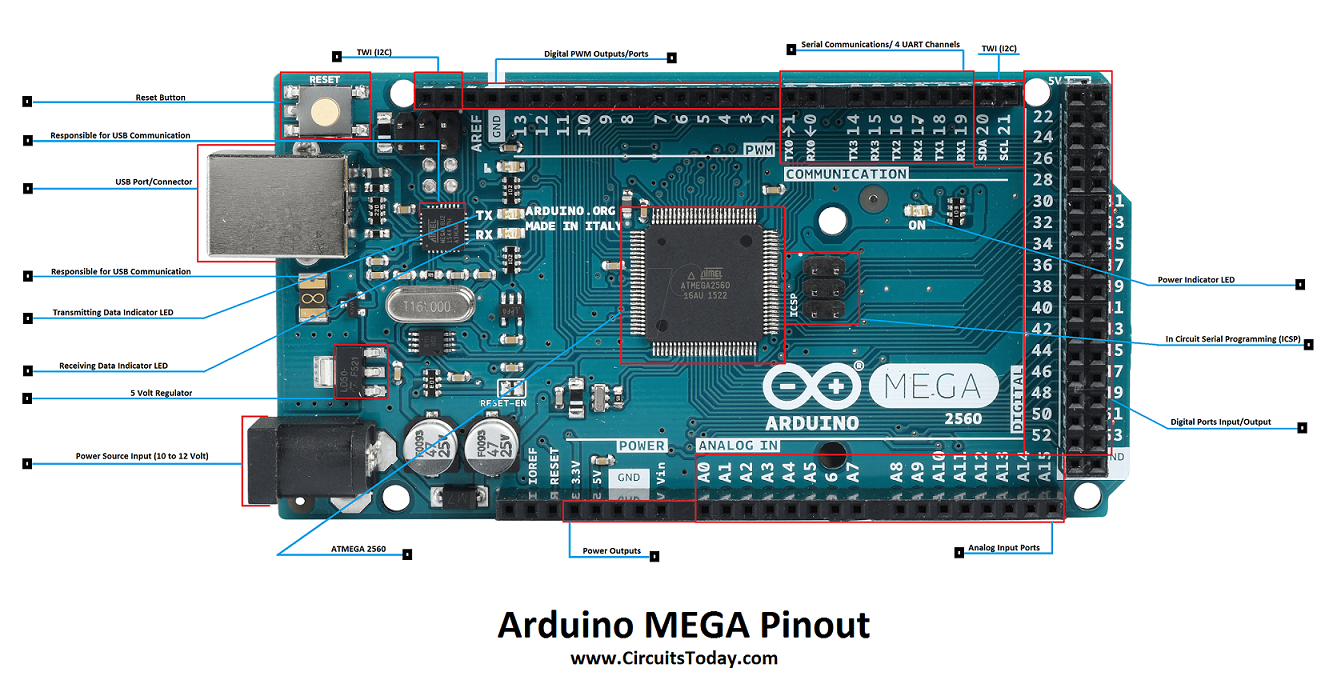
 

Figure 4.2: Arduino Mega 2560 hardware Figure 4.3: Arduino Software

2. TCS 3200 Color sensor:

TCS3200-DB Color Sensor Daughterboard is a complete color detector, including a TAOS TCS3200 RGB sensor chip, white LEDs, collimator lens, and standoffs to set the optimum sensing distance. The TCS3200 has an array of photodetectors, each with either a red, green, or blue filter, or no filter (clear). The filters of each color are distributed evenly throughout the array to eliminate location bias among the colors. Internal to the device is an oscillator which produces a square-wave output whose frequency is proportional to the intensity of the chosen color. The applications of color sensor are Test strip reading, sorting by color, Ambient light sensing and calibration, Color matching.



Figure 4.4: Color Sensor TCS3200

3. IR Sensor (LM393)

* Working voltage: 3 - 5V DC
* Output type: Digital switching output (0 and 1)
* 3mm screw holes for easy mounting
* Board size: 3.2 x 1.4cm [4]



Figure 4.5: IR sensor (LM393)

4. Servo motor:

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors. Servomotors are not a specific class of motor, although the term servomotoris often used to refer to a motor suitable for use in a closed-loop control system. Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing. The Specifications of servo motor are mentioned below

* Operating voltage: 4.8 V (~5V)
* Operating speed: 0.1 s/60 degree
* Stall torque: 1.8 kgf·cm
* Dead band width: 10 µs
* Temperature range: 0 ºC – 55 ºC [16]



Figure 4.6: Servo Motors

5. Fiber board:

Fiberboard (or fiberboard) is a type of engineered wood product that is made out of wood fibers. Types of fiberboard (in order of increasing density) include particle board or low-density fiberboard (LDF), medium-density fiberboard (MDF), and hardboard (high-density fiberboard, HDF). It is sometimes used as a synonym for particle board, but particle board usually refers to low-density fiberboard. Plywood is not a type of fiberboard, as it is made of thin sheets of wood, not wood fibers or particles. Fiberboard, particularly medium-density fiberboard, is heavily used in the furniture industry. For pieces that will be visible, a veneer of wood is often glued onto fiberboard to give it the appearance of conventional wood. In the packaging industry, the term "fiberboard" is often used to describe a tough Kraft-based paperboard or corrugated fiberboard for boxes. "Fiberboard" is also an intermediate product, an output of a pulp mill used as input for a paper mill.



Figure 4.7: Fiber board

6. Colored objects:

We will be using colored objects for sorting and testing the prototype.



Figure 4.8: Colored Objects

**4.3 Implementation Methodology**

We first observed the problem and planned to implement our project according to it. We also noted various additional things we can add into our project which can make the work of industry owners simple. To implement the color sorting machine and count we followed the following steps:

Step 1: We planned what all facilities are we going to provide the user and gathered all the required hardware and software according to it.

Step 2: Implementing one sensor at a time, we tested our code and saw if we got the desired output. Initially we started with easy to implement sensors like IR Sensor and then went onto implement the color sensor.

* IR Sensor: We have to implement one IR Sensor in this project to count the number of objects being produced. IR Sensor has 3 pins. We connect the IR Sensor to the Arduino Board. Then we burn our code to the Arduino board which executes our logic of counting the number of objects produced. That count is displayed on our serial monitor to keep track of the objects. According to our logic implemented,

0 objects → count is not affected.

1 or more object → the number of objects is counted.

* Color Sensor: We have to implement one color sensor in our project which will check the color of the object and help us in sorting objects produced in bulk. The code is run on Arduino board which executes the logic of how the objects are sorted. The sensor sorts 3 colors namely red, blue and green. Accordingly the object is sent to respective bin.

Logic:-

|  |  |  |
| --- | --- | --- |
| S2 | S3 | Color Detected |
| LOW | LOW | RED |
| HIGH | HIGH | GREEN |
| LOW | HIGH | BLUE |

Table 4.1: Logic Table for Color Sensor

Step 3: Building the Prototype:

* First we bought the fiber board and cut it into pieces of respective size.
* Then we decided the placement of the respective sensors.
* First we fixed the IR Sensor which would help us count the number of objects
* Then the color sensor was placed.
* We tested and inserted the Servo Motors for the conveyer.
* We kept respective bins for the colored objects.

**4.4 Hardware circuit diagram**

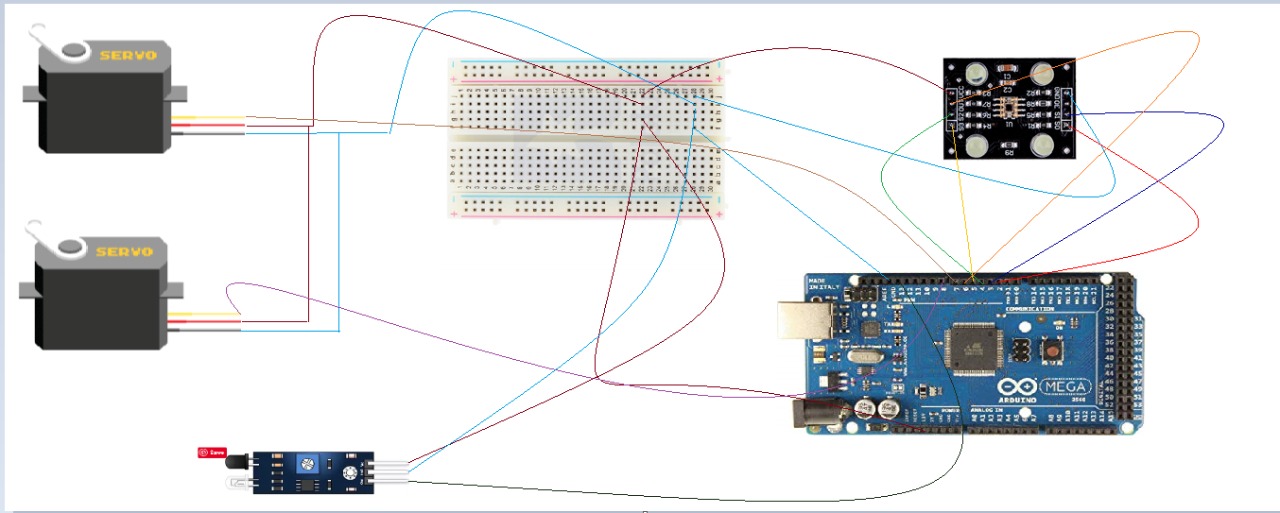
****

Figure 4.9: Circuit Diagram of Project.

**4.5 Code**

**Code for IR Sensor:-**

To count the number of objects using IR Sensor.

Display is shown on the serial monitor in arduino

Code for IR Sensor:

int LED = 13; // Use the onboard Uno LED

int obstaclePin = 7; // This is our input pin

int hasObstacle = HIGH; // HIGH MEANS NO OBSTACLE

int counter = 0;//initialize counter as 0

int i = 0;

void setup() {

pinMode(LED, OUTPUT);

pinMode(obstaclePin, INPUT);

Serial.begin(9600);

}

void loop() {

hasObstacle = digitalRead(obstaclePin); //Reads the output of the obstacle sensor from the 7th PIN of the Digital section of the arduino

if (hasObstacle == LOW) //LOW means something is ahead, so illuminates the 13th Port connected LED

{

Serial.println("Stop something is ahead!!");

counter = counter+1;

Serial.println(counter);

digitalWrite(LED, HIGH);//Illuminates the 13th Port LED

}

else

{

Serial.println("Path is clear");

digitalWrite(LED, LOW);

}

delay(200);

}

**Code for Color Sensor TCS320:-**

#include <Servo.h>  
#define S0 2  
#define S1 3  
#define S2 4  
#define S3 5  
#define sensorOut 6  
Servo topServo;  
Servo bottomServo;  
int frequency = 0;  
int color=0;  
int LED = 6;  
int isObstaclePin = 13;  // This is our input pin  
int isObstacle = HIGH;  // HIGH MEANS NO OBSTACLE  
int count ;  
void setup() {  
  pinMode(S0, OUTPUT);  
  pinMode(S1, OUTPUT);  
  pinMode(S2, OUTPUT);  
  pinMode(S3, OUTPUT);  
  pinMode(sensorOut, INPUT);  
  // Setting frequency-scaling to 20%  
  digitalWrite(S0, HIGH);  
  digitalWrite(S1, LOW);  
  topServo.attach(7);  
  bottomServo.attach(8);  
  Serial.begin(9600);  
   
}  
void loop() {  
   isObstacle = digitalRead(isObstaclePin);  
  if (isObstacle == LOW)  
  {  
    Serial.println("OBSTACLE!!, OBSTACLE!!");  
    count = count + 1 ;  
    Serial.println(count);  
    digitalWrite(LED, HIGH);  
     
  }  
  else  
  {  
    Serial.println("clear");  
    digitalWrite(LED, LOW);  
  }  
  delay(200);  
   
  topServo.write(36);  
  delay(3000);  
     
  color = readColor();  
  delay(50);    
  switch (color) {  
    case 1:  
    bottomServo.write(20);  
    break;  
    case 2:  
    bottomServo.write(90);  
    break;  
    case 3:  
    bottomServo.write(130);  
    break;  
    case 0:  
    break;  
  }  
  delay(5000);  
   
  topServo.write(210);  
  delay(2000);  
  color=0;  
}  
// Custom Function - readColor()  
int readColor() {  
  // Setting red filtered photodiodes to be read  
  digitalWrite(S2, LOW);  
  digitalWrite(S3, LOW);  
  // Reading the output frequency  
  frequency = pulseIn(sensorOut, LOW);  
  int R = frequency;  
  // Printing the value on the serial monitor  
  Serial.print("R= ");//printing name  
  Serial.print(frequency);//printing RED color frequency  
  Serial.print("  ");  
  delay(500);  
  // Setting Green filtered photodiodes to be read  
  digitalWrite(S2, HIGH);  
  digitalWrite(S3, HIGH);  
  // Reading the output frequency  
  frequency = pulseIn(sensorOut, LOW);  
  int G = frequency;  
  // Printing the value on the serial monitor  
  Serial.print("G= ");//printing name  
  Serial.print(frequency);//printing RED color frequency  
  Serial.print("  ");  
  delay(500);  
  // Setting Blue filtered photodiodes to be read  
  digitalWrite(S2, LOW);  
  digitalWrite(S3, HIGH);  
  // Reading the output frequency  
  frequency = pulseIn(sensorOut, LOW);  
  int B = frequency;  
  // Printing the value on the serial monitor  
  Serial.print("B= ");//printing name  
  Serial.print(frequency);//printing RED color frequency  
  Serial.println("  ");  
  delay(500);  
  if(R<G && R<G){  
    color = 1; // Red  
     
  }  
   
  else if(G<=R && G<B ){  
    color = 2; // Green  
     
  }  
   
   
  else if (B<R && B<G){  
    color = 3; // Blue  
     
  }  
  Serial.println(color);  
  return color;    
}

**4.6 Final Prototype**

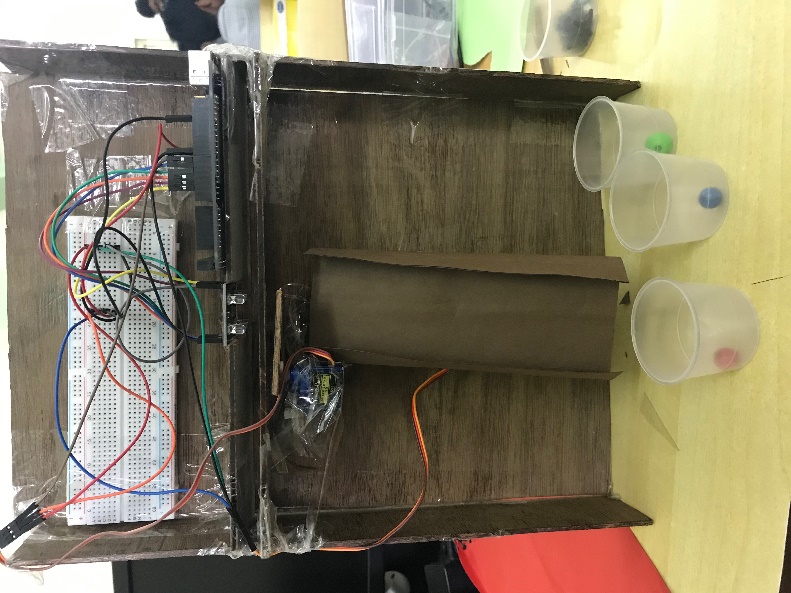
****

Figure 4.10: Final Prototype of Project

**4.7 Conclusion and Future scope of the project**

**Conclusion:**

The proposed project provides a color sorting machine with the ability to also count the number of objects produced. This helps us have a two in one machine and the working of this model is much easier as compared to other bulky industrial machines due to its features and compact size.

**Future Scope:**

* A robotic arm can be used to pick and place objects which can make the working of the project faster.
* The sensors can be modified to not only match the needs of color sorting but also distinguish between various other objects.
* It can be made more durable by using better quality materials.
* Large industries associated with color sorting can use this project to reduce bulky machinery in their workspace causing the working area to be more organized and provide larger amount of space for mobility.

**4.8 Constraints for real time deployment**

* Arduino Board: Even though this project could be implemented with Arduino Mega board, it is a small scale project. If this project is to be implemented on a large scale, the Arduino Mega board will fall short due to less number of pins and processing capabilities.
* IR Sensor: IR sensors used here have a small capability of detecting objects and they have a small range to detect objects in the environment. For large scale, IR sensors with a greater detection range and performance will have to be used.
* Cost: If this machine needs to be implemented in a real time industry, the net cost will be much higher.
* Sorting objects: In this project only particular objects can be sorted keeping in mind the size and other specifications of the objects.

**REFERENCES**

|  |  |
| --- | --- |
| **Sr.No** | **Link** |
| 1 | https://www.google.com/url?q=https%3A%2F%2Fwww.geeksforgeeks.org%2Fintroduction-to-internet-of-things-iot-set-1%2F&sa=D&sntz=1&usg=AFQjCNGsZEe2MVCdy-G8BpMEWwfbZvVNwg |
| 2 | https://www.google.com/url?q=https%3A%2F%2Fwww.linkedin.com%2Fpulse%2F20-real-world-problems-solved-iot-c-thomas-tom-smith-iii&sa=D&sntz=1&usg=AFQjCNGI9d3-Jx\_5c545-QNkD1w66AWALg |
| 3 | https://www.google.com/url?q=https%3A%2F%2Fwww.edureka.co%2Fblog%2Fiot-applications%2F&sa=D&sntz=1&usg=AFQjCNHAbcFe4eDCZ-Ce\_Rl6DakoSDVfpw |
| 4 | <https://circuitdigest.com/microcontroller-projects/arduino-color-sorter-machine-using-tcs3200-color-sensor> |
| 5 | <https://howtomechatronics.com/projects/arduino-color-sorter-project/> |
| 6 | <https://www.youtube.com/watch?v=g3i51hdfLaw> |
| 7 | <https://nevonprojects.com/iot-color-based-product-sorting-machine-project/> |
| 8 | <https://www.playembedded.org/blog/an-object-counter-using-an-ir-sensor-and-arduino/> |
| 9 | <http://surajpassion.in/ir-obstacle-sensor-with-arduino/> |
| 10 | <https://www.mepits.com/project/274/embedded-projects/arduino-based-color-sorting-machine> |
| 11 | <https://randomnerdtutorials.com/arduino-color-sensor-tcs230-tcs3200/> |

**APPENDIX**

**Assignment 1: Embedded Systems**

Embedded system is basically the study of how to setup a device that is hardware or software or both that is embedded in a larger system and is mostly a real time system. An embedded system usually consists of a microcontroller programmed to do a specific job. Internet of things is how these devices communicate with each other directly and indirectly to serve a specific purpose. Directly is when two devices or more talk peer to peer. And decide actions based on what the other device says. Indirect is when all of these devices are connected to a single node and the node receives and transmits signals to the devices and intercommunicate is thus established.

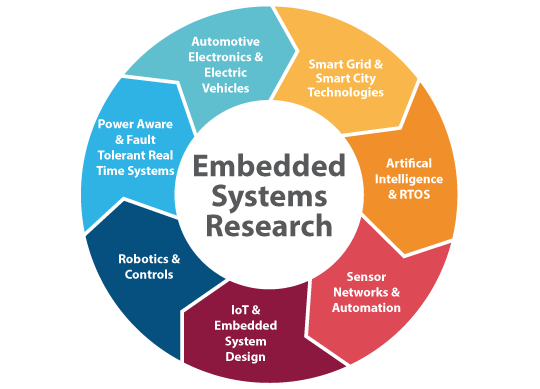


Figure a1.1: Embedded systems

**Core of Embedded Systems**

Embedded systems are domain and application specific and are built around a central core. The core of the embedded system falls into any of the following categories:

**General purpose and Domain Specific Processors**

Almost 80% of the embedded systems are processor/ controller based. The processor may be microprocessor or a microcontroller or digital signal processor, depending on the domain and application.

**Microprocessors:**

A microprocessor is a silicon chip representing a central processing unit. A microprocessor is a dependent unit and it requires the combination of other hardware like memory, timer unit, and interruptscontroller, etc. for proper functioning. Architectures used for processor design are Harvard or Von-Neumann.

**Microcontrollers:**

A microcontroller is a highly integrated chip that contains a CPU, scratch pad RAM, special and general purpose register arrays, on chip ROM/FLASH memory for program storage, timer and interrupt control units and dedicated I/O ports. Microcontrollers comprise the main elements of a small computer system on a single chip. They contain the memory, and IO as well as the CPU one the same chip. This considerably reduces the size, making them ideal for small embedded systems, but means that there are compromises in terms of performance and flexibility. Microcontrollers are often intended for low power and low processing applications, some microcontrollers may only use 4 bit words and they may also operate with very low clock rates - some 10 kHz and less to conserve power. This means that some MCUs may only consume a mill watt or so and they may also have sleep consumption levels of a few nano watts. At the other end of the scale some MCUs may need much higher levels of performance and may have very much higher clock speeds and power consumption. Texas Instrument’s TMS 1000 is considered as the world’s first microcontroller. Some embedded system application require only 8 bit controllers whereas some requiring superior performance and computational needs demand 16/32 bit controllers. The instruction set of a microcontroller can be RISC or CISC. Microcontrollers are designed for either general purpose application requirement or domain specific application requirement.

**CISC:**

Stands for "Complex Instruction Set Computing." This is a type of design in which the CISC architecture contains a large set of computer instructions that range from very simple to very complex and specialized. Though the design was intended to compute complex instructions in the most efficient way, it was later found that many small, short instructions could compute complex instructions more efficiently. This led to a design called Reduced Instruction Set Computing (RISC), which is now the other major kind of architecture. Intel Pentium processors are mainly CISC-based, with some RISC facilities built into them, whereas the PowerPC processors are completely RISC-based.

**RISC:**

RISC (reduced instruction set computer) is a [microprocessor](https://www.google.com/url?q=https%3A%2F%2Fwhatis.techtarget.com%2Fdefinition%2Fmicroprocessor-logic-chip&sa=D&sntz=1&usg=AFQjCNFF0vcH4WjjXmYSEWh_rv4Rqf0zXg) that is designed to perform a smaller number of types of computer [instruction](https://www.google.com/url?q=https%3A%2F%2Fwhatis.techtarget.com%2Fdefinition%2Finstruction&sa=D&sntz=1&usg=AFQjCNGVOUv1Srzm2mIXjKo_3qAWihOzEw)s so that it can operate at a higher speed (perform more millions of instructions per second, or [MIPS](https://www.google.com/url?q=https%3A%2F%2Fsearchitoperations.techtarget.com%2Fdefinition%2FMIPS-million-instructions-per-second&sa=D&sntz=1&usg=AFQjCNGEAWqMUQp-6hRHyI7YWahpjxDgrQ)). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the working more complicated and slower in operation.

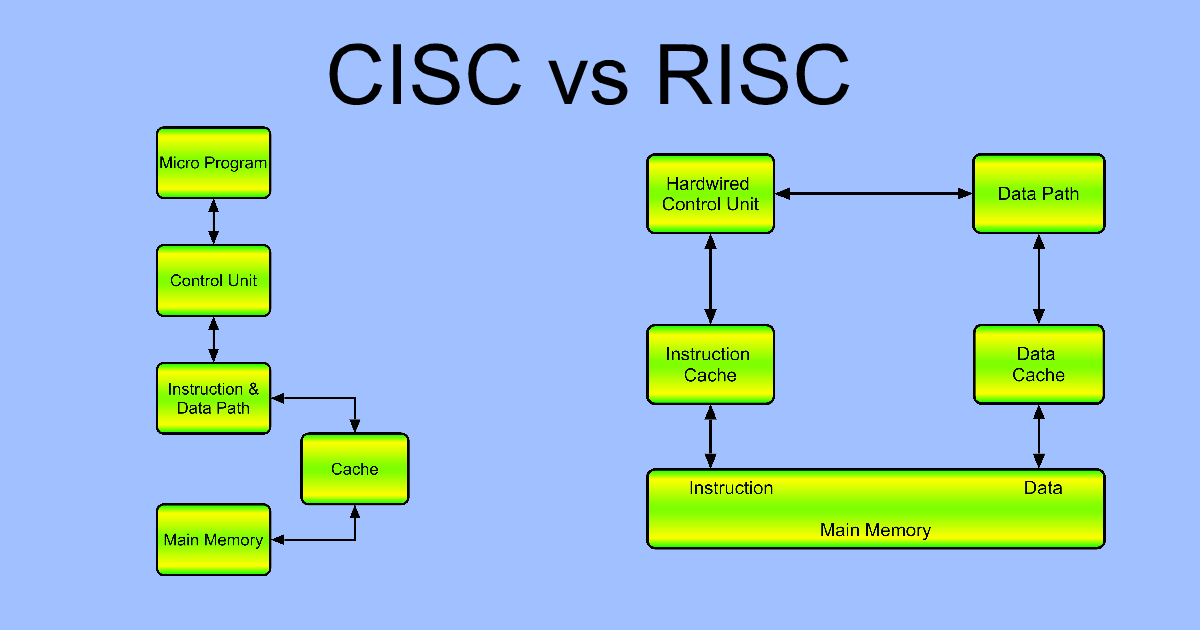


Figure a1.2: Block diagram of CISC and RISC.



Figure a1.3: CISC and RISC

# **ARM:**

An ARM processor is one of a family of [CPUs](https://whatis.techtarget.com/definition/processor) based on the [RISC](https://search400.techtarget.com/definition/RISC) (reduced instruction set computer) architecture developed by Advanced RISC Machines (ARM). ARM makes 32-bit and [64-bit](https://searchdatacenter.techtarget.com/definition/64-bit-processor) RISC [multi-core processors](https://searchdatacenter.techtarget.com/definition/multi-core-processor). RISC [processors](https://whatis.techtarget.com/definition/microprocessor-logic-chip) are designed to perform a smaller number of types of computer [instructions](https://whatis.techtarget.com/definition/instruction) so that they can operate at a higher speed, performing more millions of instructions per second ([MIPS](https://searchitoperations.techtarget.com/definition/MIPS-million-instructions-per-second)).  By stripping out unneeded instructions and optimizing pathways, RISC processors provide outstanding performance at a fraction of the power demand of [CISC](https://whatis.techtarget.com/definition/CISC-complex-instruction-set-computer-or-computing) (complex instruction set computing) devices. ARM processors are extensively used in consumer electronic devices such as [smart phones](https://searchmobilecomputing.techtarget.com/definition/smartphone), [tablets](https://searchmobilecomputing.techtarget.com/definition/tablet-PC), multimedia players and other mobile devices, such as [wearables](https://internetofthingsagenda.techtarget.com/definition/wearable-computer). Because of their reduced [instruction set](https://whatis.techtarget.com/definition/instruction-set), they require fewer [transistors](https://whatis.techtarget.com/definition/transistor), which enable a smaller die size for the integrated circuitry ([IC](https://whatis.techtarget.com/definition/integrated-circuit-IC)). The ARM processors smaller size reduced complexity and lower power consumption makes them suitable for increasingly miniaturized devices. The simplified design of ARM processors enables more efficient multi-core processing and easier coding for developers.

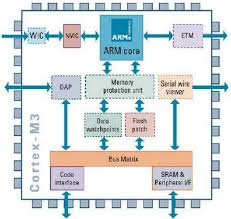


Figure a1.4: ARM Block Diagram



Figure a1.5: ARM processor

**ARM processor features include:**

* Load/store architecture.
* An [orthogonal](https://searchstorage.techtarget.com/definition/orthogonal) instruction set.
* Mostly single-cycle execution.
* Enhanced power-saving design.
* 64 and 32-bit execution states for scalable high performance.
* [Hardware virtualization](https://searchservervirtualization.techtarget.com/definition/hardware-virtualization) support.

**DSP:**

A digital signal processor (DSP) (or a [SIP block](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FSystem_in_package&sa=D&sntz=1&usg=AFQjCNEerBSYJiX1ClOhOURreKZRrhzwtA)) is optimized for the operational needs of [digital signal processing](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FDigital_signal_processing&sa=D&sntz=1&usg=AFQjCNEb2Ch_wVDUIEgZzBdJYbZQofZang). The goal of DSP is usually to measure, filter or compress continuous real-world [analog signals](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FAnalog_signals&sa=D&sntz=1&usg=AFQjCNEqtq8vYk1bJf4n4QiPPxXJLjEtMA). Most general-purpose microprocessors can also execute digital signal processing algorithms successfully, but may not be able to keep up with such processing continuously in real-time. Also, dedicated DSPs usually have better power efficiency, thus they are more suitable in portable devices such as [mobile phones](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FMobile_phone&sa=D&sntz=1&usg=AFQjCNHVnzoqi9CkdHgF5OKtP20se4J0dA) because of power consumption constraints.[[3]](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FDigital_signal_processor%23cite_note-schaum-2004-3&sa=D&sntz=1&usg=AFQjCNG9uozH4ACcu3NzzSBZT1WDhJtCFQ) DSPs often use special [memory architectures](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FMemory_architecture&sa=D&sntz=1&usg=AFQjCNFy9wT1PUWKauXvRqd-wQMuXCgQ-Q) that are able to fetch multiple data or instructions at the same time.

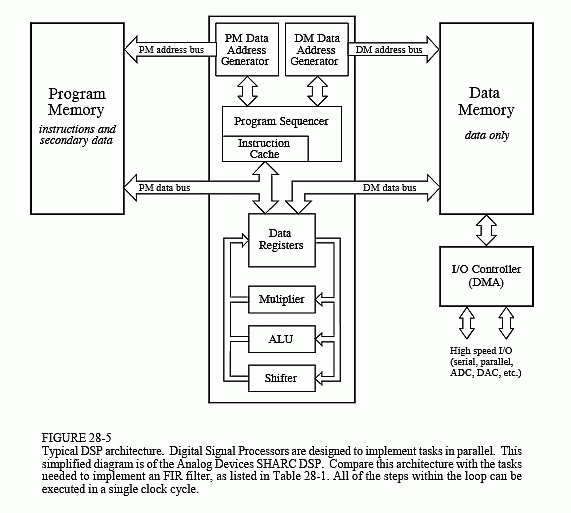


Figure a1.6: DSP Block diagram



Figure a1.7:  Digital Signal Processor

**Application Specific Integrated Circuits. (ASIC)**

ASICs are a microchip design to perform specific and unique applications.  Because of using single chip for integrates several functions there by reduces the system development cost. It helps in the design of smaller system with high capabilities or functionalities. The developers of such chips may not be interested in revealing the internal detail of it

**Programmable logic devices (PLD’s)**

A PLD is an electronic component. It used to build digital circuits which are reconfigurable.  A logic gate has a fixed function but a PLD does not have a defined function at the time of manufacture.  PLDs offer customers a wide range of logic capacity, features, speed, voltage characteristics. PLDs can be reconfigured to perform any number of functions at any time. A variety of tools are available for the designers of PLDs which are inexpensive and help to develop, simulate and test the designs.

      PLDs having following two major types.

**1) CPLD (Complex Programmable Logic Device):**

CPLDs offer much smaller amount of logic up to 1000 gates.

**2) FPGAs (Field Programmable Gate Arrays):**

It offers highest amount of performance as well as highest logic density, the most features.

**Commercial off-the-shelf components (COTs)**

A Commercial off the Shelf product is one which is used 'as-is'. The COTS components itself may be develop around a general purpose or domain specific processor or ASICs or a PLDs. The major advantage of using COTS is that they are readily available in the market, are chip and a developer can cut down his/her development time to a great extent. The major drawback of using COTS components in embedded design is that the manufacturer of the COTS component may withdraw the product or discontinue the production of the COTS at any time if rapid change in technology occurs.

DSP includes following key units:

Program memory: It is a memory for storing the program required by DSP to process the data.

Data memory: It is a working memory for storing temporary variables and data/signal to be processed.

Computational engine: It performs the signal processing in accordance with the stored program memory computational engine incorporated many specialized arithmetic units and each of them operates simultaneously to increase the execution speed. It also includes multiple hardware shifters for shifting operands and saves execution time.

I/O unit: It acts as an interface between the outside world and DSP. It is responsible for capturing signals to be processed and delivering the processed signals.

Examples: Audio video signal processing, telecommunication and multimedia applications.

**SoC:**

A system on a chip or system on chip is an [integrated circuit](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FIntegrated_circuit&sa=D&sntz=1&usg=AFQjCNEU9phlnsGBUcD1twJiETSUkA_11g) (also known as a "chip") that integrates all components of a [computer](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FComputer&sa=D&sntz=1&usg=AFQjCNHmXu32vKGipLKVFMDsrpSsRcLaqw) or other [electronic system](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FElectronics&sa=D&sntz=1&usg=AFQjCNFgXX3muXB6xusy_Zkz2imKP80gIw). These components typically (but not always) include a [central processing unit](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FCentral_processing_unit&sa=D&sntz=1&usg=AFQjCNHjIxTj_xKCXZCFvx_bd7IOjxWsAQ)(CPU), [memory](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FComputer_memory&sa=D&sntz=1&usg=AFQjCNFZ0PguVnikXA14_pRDnQ0P0mXbHw), [input/output](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FInput%2Foutput&sa=D&sntz=1&usg=AFQjCNHvy2tWR_pqzBiac0cYrQIfjt6ZlA) ports and [secondary storage](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FComputer_data_storage%23Secondary_storage&sa=D&sntz=1&usg=AFQjCNE86LT--WGjV2q9gKw4KkxAMk60aA) – all on a single [substrate](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FWafer_(electronics)&sa=D&sntz=1&usg=AFQjCNGW2TxcmHqyVjtcf3q8ere2j1kt5w) or microchip, the size of a coin.[[1]](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FSystem_on_a_chip%23cite_note-2&sa=D&sntz=1&usg=AFQjCNFpAqPPXZhtD98noBe0gIHVnjvrTA) It may contain [digital](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FDigital_signal_(electronics)&sa=D&sntz=1&usg=AFQjCNHYa6EcmN2y9LIombyti2aHPBz7Pw), [analog](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FAnalog_signal&sa=D&sntz=1&usg=AFQjCNFRlYrQ5uU_Pgp44MuET11TdKbjOw), [mixed-signal](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FMixed-signal_integrated_circuit&sa=D&sntz=1&usg=AFQjCNFO28CeutwlREifai-MrNPVenNHHg), and often [radio frequency](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FRadio_frequency&sa=D&sntz=1&usg=AFQjCNGshrVVm76ubvMxW4Dr4p-PFb8htQ) [signal processing](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FSignal_processing&sa=D&sntz=1&usg=AFQjCNGyCf7FUhiovT7u4Kc1pBvYjJBWhg) functions, depending on the application. As they are integrated on a single substrate, SoCs consume much less power and take up much less area than multi-chip designs with equivalent functionality. Because of this, SoCs are very common in the [mobile computing](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FMobile_computing&sa=D&sntz=1&usg=AFQjCNFUIVeZ0a5TKiyMy34yUC_agqSHNg) (such as in [Smart phones](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FSmartphone&sa=D&sntz=1&usg=AFQjCNHRPMCJ7dCbGEWM_ULYXvWqk6IXQQ)) and [edge computing](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FEdge_computing&sa=D&sntz=1&usg=AFQjCNEXrsTfWsOyZ5jUShlsjPDfpT8jcA) markets. Systems on chip are commonly used in [embedded systems](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FEmbedded_system&sa=D&sntz=1&usg=AFQjCNGsRXL96_F0W9nYPJTRG_fkn3RnLw) and the [Internet of Things](https://www.google.com/url?q=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FInternet_of_things&sa=D&sntz=1&usg=AFQjCNFuCONdv8SVMhiVP7aWsDOp_WcjBg).

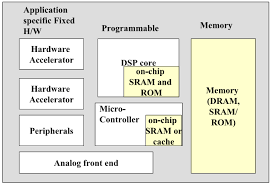


Figure a1.8: SoC Block Diagram.



Figure a1.9: SoC

**Assignment 2: Embedded System Tools**

**1. ARDUINO:-**

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. Arduino is an open-source electronics platform based on easy-to-use hardware and software. [Arduino boards](https://www.arduino.cc/en/Main/Products) are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the [Arduino programming language](https://www.arduino.cc/en/Reference/HomePage) (based on [Wiring](http://wiring.org.co/)), and [the Arduino Software (IDE)](https://www.arduino.cc/en/Main/Software), based on [Processing](https://processing.org/). Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of [accessible knowledge](http://forum.arduino.cc/) that can be of great help to novices and experts alike.

**Features of Arduino:-**

* **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
* **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
* **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
* **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
* **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the [breadboard version of the module](https://www.arduino.cc/en/Main/Standalone) in order to understand how it works and save money.



Figure a2.1: Arduino Software

Different Types of Arduino Boards:-

The list of Arduino boards includes the following such as

* Arduino Uno (R3): The Uno is a huge option for your initial Arduino. It consists of 14-digital I/O pins, where 6-pins can be used as PWM ([pulse width modulation](https://www.elprocus.com/pulse-width-modulation-pwm/) outputs), 6-analog inputs, a reset button, a power jack, a USB connection and more. It includes everything required to hold up the microcontroller; simply attach it to a PC with the help of a USB cable and give the supply to get started with a AC-to-DC adapter or battery.
* LilyPad Arduino: The Lily Pad Arduino board is a wearable e-textile technology expanded by Leah “Buechley” and considerately designed by “Leah and SparkFun”. Each board was imaginatively designed with huge connecting pads & a smooth back to let them to be sewn into clothing using conductive thread. This Arduino also comprises of I/O, power, and also sensor boards which are built especially for e-textiles. These are even washable!
* Red Board: The RedBoard Arduino board can be programmed using a Mini-B USB cable using the Arduino IDE. It will work on Windows 8 without having to modify your security settings. It is more constant due to the USB or FTDI chip we used and also it is entirely flat on the back. Creating it is very simple to utilize in the project design. Just plug the board, select the menu option to choose an Arduino UNO and you are ready to upload the program. You can control the RedBoard over USB cable using the barrel jack.
* Arduino Mega (R3): The Arduino Mega is similar to the UNO’s big brother. It includes lots of digital I/O pins (from that, 14-pins can be used as PWM o/ps), 6-analog inputs, a reset button, a power jack, a USB connection and a reset button. It includes everything required to hold up the microcontroller; simply attach it to a PC with the help of a USB cable and give the supply to get started with AC-to-DC adapter or battery. The huge number of pins makes this Arduino board very helpful for designing the projects that need bunch of digital i/ps or o/ps like lots buttons.
* Arduino Leonardo: The first development board of an Arduino is the Leonardo board. This board uses one microcontroller along with the USB. That means, it can be very simple and cheap also. Because this board handles USB directly, program libraries are obtainable which let the Arduino board to follow a keyboard of the computer, mouse, etc.

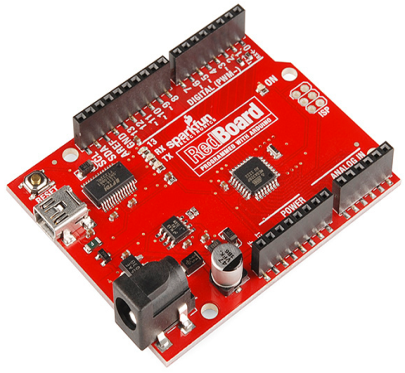
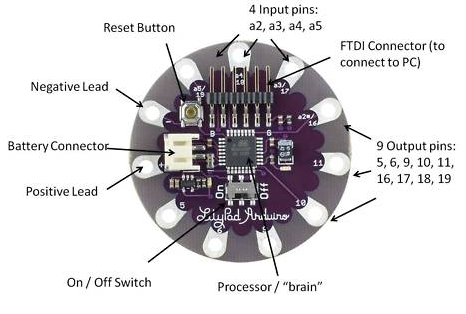
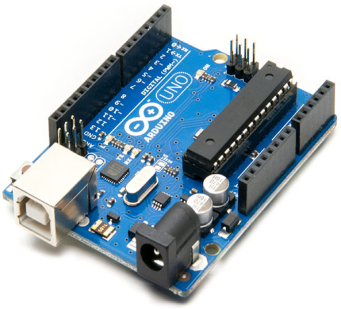


Figure a2.2: Arduino Uno (R3) Figure a2.3: LilyPad Arduino Figure a2.4: Red Board

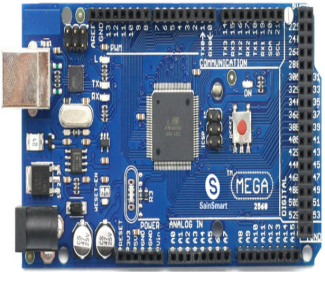
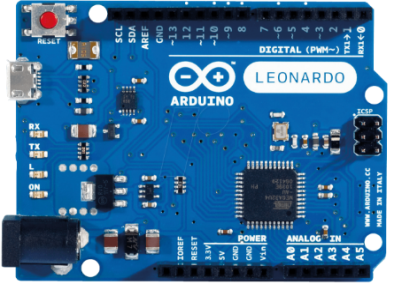
 

Figure a2.5: Arduino Mega (R3) Figure a2.6: Arduino Leonardo

**2. RASPBERRY PI:-**

The Raspberry Pi is a series of small [single-board computers](https://en.wikipedia.org/wiki/Single-board_computer) developed in the [United Kingdom](https://en.wikipedia.org/wiki/United_Kingdom) by the [Raspberry Pi Foundation](https://en.wikipedia.org/wiki/Raspberry_Pi_Foundation) to promote teaching of basic [computer science](https://en.wikipedia.org/wiki/Computer_science) in schools and in [developing countries](https://en.wikipedia.org/wiki/Developing_countries). The original model became far more popular than anticipated, selling outside its [target market](https://en.wikipedia.org/wiki/Target_market) for uses such as [robotics](https://en.wikipedia.org/wiki/Robotics). It does not include peripherals (such as [keyboards](https://en.wikipedia.org/wiki/Keyboard_(computing)) and [mice](https://en.wikipedia.org/wiki/Mouse_(computing))) or [cases](https://en.wikipedia.org/wiki/Computer_case). However, some accessories have been included in several official and unofficial bundles. **RASPBERRY PI**platform is most used after [ADRUINO](https://components101.com/microcontrollers/arduino-uno). Although overall applications of PI are less it is most preferred when developing advanced applications. Also the RASPBERRY PI is an open source platform where one can get a lot of related information so you can customize the system depending on the need. Here are few examples where RASPBERRY PI 3 is chosen over other microcontrollers and development boards:

1. Where the system processing is huge. Most ARDUINO boards all have clock speed of less than 100MHz, so they can perform functions limited to their capabilities. They cannot process high end programs for applications like Weather Station, Cloud server, gaming console etc. With **1.2GHz clock speed**and**1 GB RAM RASPBERRY PI** can perform all those advanced functions.

2. Where wireless connectivity is needed. RASPBERRY PI 3 has wireless LAN and Bluetooth facility by which you can setup WIFI HOTSPOT for internet connectivity. For **Internet of Things** this feature is best suited.

3. RASPBERRY PI had dedicated port for connecting touch LCD display which is a feature that completely omits the need of monitor.

4. RASPBERRY PI also has dedicated camera port so one can connect camera without any hassle to the PI board.

5. RASPBERRY PI also has PWM outputs for application use.

There are many other features like HD steaming which further promote the use of RASPBERRY PI.

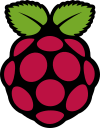
 

Figure a2.7: Raspberry Pi Logo Figure a2.8: Raspberry Pi

**Generations of released models:-**

| Family | Model | Form Factor | Ethernet | Wireless | GPIO | Released | Discontinued |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Raspberry Pi 1 | B | Standard  (85.60 × 56.5 mm) | Yes | No | 26-pin | 2012 | Yes |
| A | No | 2013 | Yes |
| B+ | Yes | 40-pin | 2014 |  |
| A+ | Compact  (65 × 56.5 mm) | No | 2014 |  |
| Raspberry Pi 2 | B | Standard | Yes | No | 2015 |  |
| Raspberry Pi Zero | Zero | Zero  (65 × 30 mm) | No | No | 2015 |  |
| W/WH | Yes | 2017 |  |
| Raspberry Pi 3 | B | Standard | Yes | Yes | 2016 |  |
| A+ | Compact | No | 2018 |  |
| B+ | Standard | Yes | 2018 |  |
| Raspberry Pi 4 | B (1GB) | Standard | Yes | Yes | 2019[[26]](https://en.wikipedia.org/wiki/Raspberry_Pi#cite_note-:4-26) |  |
|  |  |

Figure a2.1: Table for generations of Raspberry Pi

**3. ARM Cortex:-**

The ARM Cortex-M families are ARM microprocessor cores which are designed for use in microcontrollers, ASICs, ASSPs, FPGAs, and SoCs. Cortex-M cores are commonly used as dedicated microcontroller chips, but also are "hidden" inside of SoC chips as power management controllers, I/O controllers, system controllers, touch screen controllers, smart battery controllers, and sensors controllers. Though 8-bit microcontrollers were very popular in the past, Cortex-M has slowly been chipping away at the 8-bit market as the prices of low-end Cortex-M chips have moved downward. Cortex-M have become a popular replacements for 8-bit chips in applications that benefit from 32-bit math operations, and replacing older legacy ARM cores such as ARM7 and ARM9.



Figure a2.9: ARM Cortex

**4. INTEL GALILEO:-**

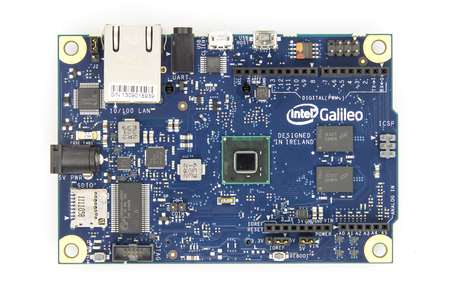
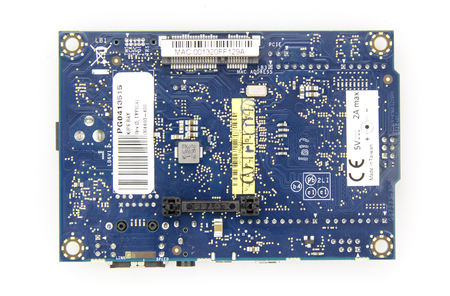
 

Figure a2.10: Intel Galileo Front Figure a2.11: Intel Galileo Back

Intel Galileo is the first in a line of Arduino-certified development boards based on Intel x86 architecture and is designed for the maker and education communities. Intel released two versions of Galileo, referred to as Gen 1 and Gen 2. These development boards are sometimes called "Breakout boards". Galileo is a microcontroller board based on the Intel® Quark SoC X1000 Application Processor, a 32-bit Intel Pentium-class system on a chip (datasheet). It’s the first board based on Intel® architecture designed to be hardware and software pin-compatible with Arduino shields designed for the Uno R3. Digital pins 0 to 13 (and the adjacent AREF and GND pins), Analog inputs 0 to 5, the power header, ICSP header, and the UART port pins (0 and 1), are all in the same locations as on the Arduino Uno R3. This is also known as the Arduino 1.0 pinout. Galileo is designed to support shields that operate at either 3.3V or 5V. The core operating voltage of Galileo is 3.3V. However, a jumper on the board enables voltage translation to 5V at the I/O pins. This provides support for 5V Uno shields and is the default behavior. By switching the jumper position, the voltage translation can be disabled to provide 3.3V operation at the I/O pins. Of course, the Galileo board is also software compatible with the Arduino Software Development Environment (IDE), which makes usability and introduction a snap. In addition to Arduino hardware and software compatibility, the Galileo board has several PC industry standard I/O ports and features to expand native usage and capabilities beyond the Arduino shield ecosystem. A full sized mini-PCI Express slot, 100Mb Ethernet port, Micro-SD slot, RS-232 serial port, USB Host port, USB Client port, and 8MByte NOR flash come standard on the board.

**5. DIFFERENT TYPES OF SENSORS:-**

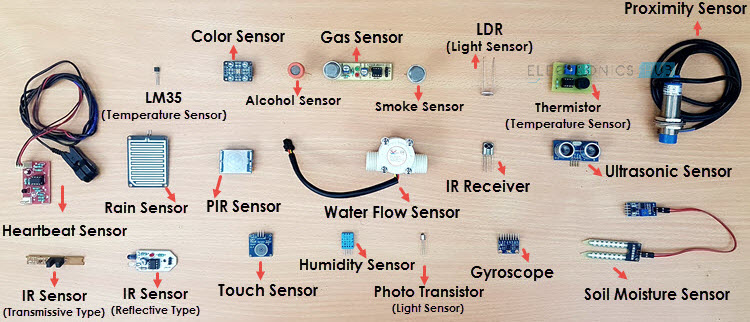


Figure a2.12: Different types of sensors.

A) Temperature Sensor

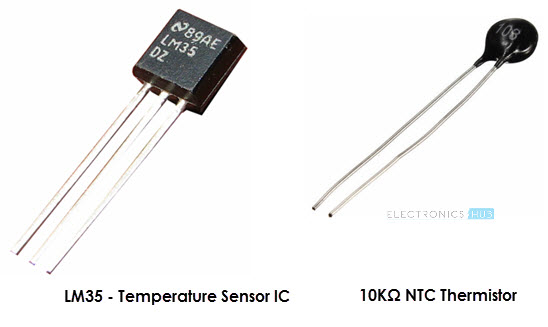


Figure a2.13: Temperature Sensor

One of the most common and most popular sensor is the Temperature Sensor. A Temperature Sensor, as the name suggests, senses the temperature i.e. it measures the changes in the temperature. In a Temperature Sensor, the changes in the Temperature correspond to change in its physical property like resistance or voltage. There are different types of Temperature Sensors like Temperature Sensor ICs (like LM35), Thermistors, Thermocouples, RTD (Resistive Temperature Devices), etc. Temperature Sensors are used everywhere like computers, mobile phones, automobiles, air conditioning systems, industries etc.

#### B) Proximity Sensors

#### **Types of Sensors Image 4**

Figure a2.14: Proximity Sensors

A Proximity Sensor is a non-contact type sensor that detects the presence of an object. Proximity Sensors can be implemented using different techniques like Optical (like Infrared or Laser), Ultrasonic, Hall Effect, Capacitive, etc. Some of the applications of Proximity Sensors are Mobile Phones, Cars (Parking Sensors), industries (object alignment), Ground Proximity in Aircrafts, etc.

#### C) Infrared Sensor (IR Sensor)

#### **Types of Sensors Image 5**

#### Figure a2.15: Infrared Sensor (IR Sensor)

IR Sensors or Infrared Sensor is light based sensor that are used in various applications like Proximity and Object Detection. IR Sensors are used as proximity sensors in almost all mobile phones. There are two types of Infrared or IR Sensors: Transmissive Type and Reflective Type. In Transmissive Type IR Sensor, the IR Transmitter (usually an IR LED) and the IR Detector (usually a Photo Diode) are positioned facing each other so that when an object passes between them, the sensor detects the object. The other type of IR Sensor is a Reflective Type IR Sensor. In this, the transmitter and the detector are positioned adjacent to each other facing the object. When an object comes in front of the sensor, the sensor detects the object. Different applications where IR Sensor is implemented are Mobile Phones, Robots, Industrial assembly, automobiles etc.

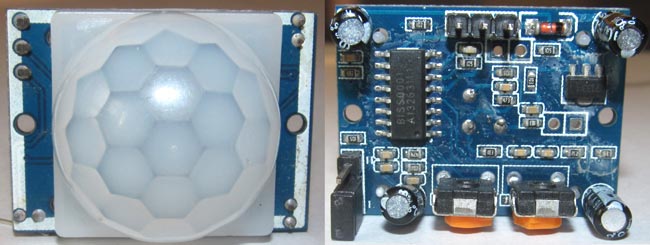
#### D) Ultrasonic Sensor

#### **Types of Sensors Image 6**

#### Figure a2.16: Ultrasonic Sensor

#### An Ultrasonic Sensor is a non-contact type device that can be used to measure distance as well as velocity of an object. An ultrasonic sensor works based on the properties of sound waves with frequency greater than human audible range. Using the time of flight of the sound wave, an Ultrasonic Sensor can measure the distance of the object (similar to SONAR). The Doppler Shift property of the sound wave is used to measure the velocity of an object.

### **E) PIR Sensor:**



#### Figure a2.17: PIR Sensor

PIR sensor stands for **Passive Infrared sensor.** These are used to detect the motion of humans, animals or things. We know that infrared rays have a property of reflection. When an infrared ray hits an object, depending upon the temperature of the target the infrared ray properties changes, this received signal determines the motion of the objects or the living beings. Even if the shape of the object alters, the properties of the reflected infrared rays can differentiate the objects precisely.