লেখক

সীমান্ত সরকার

হয়ে উঠো IoT Expert

(শূণ্য থেকে PYTHON + ESP32 দিয়ে ইলেক্ট্রনিক্স ও IoT শেখার কমপ্লিট গাইডলাইন)

**মুখ বন্ধ**

* + - ✅এই বইটি যাদের জন্য এবং যাদের জন্য নয়
    - ✅বইয়ের কোনো অংশ বুঝতে না পারলে কি করবো?
    - ✅কেন Micropython ও ESP32 দিয়ে IoT শেখা শুরু করবো?
    - প্রিরিকিউজিটস

**পার্ট – ১: আইওটি ও ইলেক্ট্রনিক্স কনসেপ্টস**

* **চ্যাপ্টার – শুণ্য IoT এর আদ্যপান্ত**
  + - ✅ইন্ডাস্ট্রিয়াল রিভুলেশন ও আইওটি
    - ✅আইওটি ডিভাইস
    - ✅আইওটি এর এপ্লিকেশন
    - ✅কেন IoT শেখা উচিত
    - বাংলাদেশ ও বিশ্বে IoT কীভাবে ব্যবহার হচ্ছে
* **চ্যাপ্টার – ১ IoT ডিভাইসের সব কিছু**
  + IoT আর্কিটেকচার
    - Perception layer
      * Sensor
      * Actuator
      * Relays
      * Microcontrolers
    - Network Layer
      * Communication Technology
      * Gateways
      * Cloud Platform
    - Application Layer
  + আইওটি এর 3V Data Model
  + IoT প্রোটোকলস
    - Perception Layer Protocols
    - Network Layer Protocols
    - Application Layer Protocol
  + IoT কমিউনিকেশন মডেল
    - Request & Response Model
    - Publisher-Subscriber Model
    - Push-Pull Model
    - Exclusive Pair
* **চ্যাপ্টার – ২ মাইক্রোকন্ট্রোলার ও সিনগ্যাল বোর্ড কম্পিউটার**
  + মাইক্রোকন্ট্রোলার - IoT এর ব্রেইন
    - মাইক্রোকন্ট্রোলার(MCU) কী
    - মাইক্রোকন্ট্রোলার এবং ডেভেলপমেন্ট বোর্ড
    - মাইক্রোকন্ট্রোলারের স্ট্রাকচার
  + ইন্টারফেসিং প্রোটোকলস
    - কমিউনিকেশন ইনটারফেস কি?
    - UART(Universal asynchronous receiver-transmitter)
    - SPIs (Serial Peripheral Interfaces)
    - I2C ( Inter-Integrated Circuit)
* **চ্যাপ্টার – ৩ বেসিক ইলেক্ট্রনিক্স**
  + মাল্টিমিটার
  + কারেন্ট,ভোল্টেজ,রেজিস্ট্যান্স
  + ডায়োড

**পার্ট – ২: প্রোগ্রামিং ও ফিজিক্যাল কম্পিউটিং**

* চ্যাপ্টার – ৪ শুরুর প্রস্তুতি
  + কেনা-কাটি
  + Install Python
  + ESP32 বোর্ড এর সাথে পরিচয়
  + Thorny Python IDE
    - Install
    - পরিচিতি
  + Flashing Micropython Firmware to ESP32
  + Testing in REPL
* চ্যাপ্টার – ৫ Python/Micropython Programming Basics
  + MicroPython বেসিকস Syntax,Script
    - Variable
      * Print
      * Input
      * Comments
    - Data Types
    - Operators
      * Mathematical Operators
      * Relational Operators
    - Conditional Statements
    - Loops
    - Function
    - Modules
    - Main & Boot File
* চ্যাপ্টার – ৬ Basic Projects with Micropython and ESP32
  + ✅Blinking an LED
  + ✅DHT22 Sensor
  + ✅Ultrasonic Sensor
  + ✅OLED Display
  + ✅Motor Driver: Control a DC Motor
  + ✅Push Button
  + ✅Buzzer
  + ✅Keypad
  + ✅Relay Module
  + ✅Potentiometer
  + LDR
  + ✅SD card Module
  + ✅LCD Display
  + (ADC)
  + Temperature sensor
  + Voltage sensor
  + Moisture Sensor

**পার্ট – ৩: আওটি ডেভেলপমেন্ট**

* চ্যাপ্টার – ৫
  + Web Servers and HTTP Client
* চ্যাপ্টার – ৬
  + MQTT Protocol
  + API
  + Thingspeak
  + IFTTT
* চ্যাপ্টার – ৭ Storing Data
  + Google Sheet
  + Database

**✅এরপর যা করবে**

মুখ বন্ধ

**এই বইটি যাদের জন্য এবং যাদের জন্য নয়ঃ**

মূলতো যাদের ইলেক্ট্রনিক্স,হার্ডওয়্যার,প্রোগ্রামিং নিয়ে আগ্রহ আছে বইটি তাদের জন্য লিখা। সিএসই, ইইই, ইসিই, মেকাট্রোনিক্স, রোবটিক্স এবং ইঞ্জিনিয়ারিং সাব্জেক্টে পড়ুয়া স্টুডেন্টরা এই বইটি থেকে উপকার পাবে বলে আমি আশাবাদী। এই বইটি ভার্সিটি লেভেলের স্টুডেন্টদের টার্গেট করে লিখা, যারা রিসার্চ বা থিসিসের কাজে আইওটি ব্যবহার করতে চাচ্ছে,আইডিয়া কম্পিটিশনের জন্য প্রোজেক্ট তৈরি করতে চাচ্ছে,ভবিষ্যতে আইওটি নিয়ে ক্যারিয়ার গড়তে চাচ্ছে।

যদি তুমি স্কুল-কলেজের বিজ্ঞানমেলার জন্য সহজ কিছু “ফান প্রোজেক্ট” তৈরির জন্য বইটি নিয়ে থাকো তবে তোমাকে আশাহত করে বলছি বইটি তোমার জন্য নয়।

যদি ইলেক্ট্রনিক্স-প্রোগ্রামিং নিয়ে পূর্ব অল্প বিস্তর ধারণা থাকে তবে এই বইটি পড়া তার জন্য কিছুটা সহজ হবে। যাদের একে বারেই এই নিয়ে ধারণা নেই তাদের জন্য কিছুটা কঠিন মনে হতে পারে তবে অসম্ভব কিছু নয়।

মনে রাখতে হবে,এটি কোনো গল্পের বই নয়। যেকোনো টেকনোলজি শেখার জন্য বই পড়া/টিউটোরিয়াল দেখার পাশাপাশি নিজে কোড করতে হয়।কোনো কিছু তুমি প্রথম বারেই বুঝে যাবে এমনটা প্রত্যাশা আমি করছি না, তুমিও করো না। বই পড়ার পাশাপাশি তোমাকে প্রচুর ইন্টারেনেটে ঘাটাঘাটি করতে হবে,প্রজেক্টগুলো নিজ হাতে বানিয়ে দেখতে হবে,কোনো কিছু কাজ না করলে এর সলিউশন খুঁজে বের করতে হবে।

মোট কথা বইটি পড়ার জন্য তোমাকে “প্রবলেম সলভার” মাইন্ডসেট নিয়ে আসতে হবে,তবেই বেস্ট আউটপুট পাওয়া সম্ভব।

**বইয়ের কোনো অংশ বুঝতে না পারলে কি করবো?**

যেকোনো টেকনোলজি শিখতে গিয়ে তুমি প্রথম বারেই সব বুঝে ফেলবে এমনটা আশা করা মোটেই ঠিক নয়,আমিও করছি না। কোড করতে গিয়ে তুমি ভুল করবে, ইরর খাবে।সব ঠিক-ঠাক থাকা সত্বেও প্রোজেক্ট কাজ করবে না,কয়েক ঘন্টা খুঁজা-খুঁজির পর বুঝতে পারবে কোনো একটা তার হয় তো লুজ ছিল!

যেকোনো ডেভেলপারের জন্য এই সব প্রবলেম নিত্যদিনের সঙ্গী।এই প্রবলেম সলভের কাজই ডেভেলপাররা করে থাকেন।তাই প্রত্যেক ডেভেলপারই একজন প্রবলেম সল্ভার।তোমাকেও এই মাইন্ড সেট বিল্ড করতে হবে।

কোথাও আটকে গেলে গুগল করো,ব্লগ পড়ো,ডকুমেন্টেশন ঘাটো, অন্য কারো প্রজেক্ট খুঁজে বের করার চেষ্টা করো,প্রশ্ন করো। কোড কপি করে চ্যাট জিপিটির কাছে দিয়ে বলো বুঝিয়ে দিতে,একবার না বুঝলে কয়েকবার চেষ্টা করো।খবরদার চ্যাট জিপিটি থেকে কোড কপি করবে না,কোনো টপিক বুঝার জন্য ব্যবহার করতেই পারো কিন্তু কোড কপি নয়।

তারপরও কোনো পার্ট বুঝতে না পারলে জাস্ট নোট করে সামনে এগিয়ে যাও।কোনো অংশে থেমে থেকো না,থেমে থাকা মানে পিছিয়ে পরা।এখন তুমি যা বুঝতে পারছো না অন্য কোনো কাজ করতে গিয়ে দেখবে হঠাৎ করে কন্সেপ্ট ধরে ফেলেছো। ট্রাস্ট মি! ইট ওয়ার্ক্স।

**Micropython কী ও কেন Micropython দিয়ে IoT শেখা শুরু করবো?**

Micropython আসলে Python 3 এর’ই একটি রুপান্তর তবে এটির টার্গেট ইউজ কেজ হচ্ছে মাইক্রোকন্ট্রোলার ও এম্বেডেড সিস্টেম। এটির মডিউলগুলো আরো Lighter (হাল্কা) ভার্সনে তৈরি করা হয় যেন মেমরি কম লাগে,যেহেতু মাইক্রোকন্ট্রোলারে মেমোরি কম থাকে।

পাইথনের সাথে মাইক্রোপাইথনের মূল পার্থ্ক্যটা হচ্ছে মাইক্রোপাইথন ডিজাইন করা হয়েছে Constrained Condition এ কাজ করার জন্য।আর তাই মাইক্রোপাইথনে পাইথনের মতো সকল স্ট্যন্ডার্ড লাইব্রেরী থাকে না বরং একটা ছোট সাবসেট থাকে।

কেন Micropython দিয়ে IoT শিখবো? এর উত্তর হচ্ছে- Micropython শেখা খুবই সহজ।

Micropython এর Syntax প্রায় সব দিক থেকেই Python এর মতোই।তাই তুমি যদি পাইথন আগে থেকে জেনে থাকো তাহলে এই বইয়ের মাইক্রোপাইথন নিয়ে বেসিক পার্টটুকু বাদ দিয়ে সরাসরি প্রোজেক্ট এর চ্যাপ্টারে চলে যেতে পারো।

অর্থ্যাত তুমি যদি শুধু python জেনে থাকো তবে ডিজিটাল ইলেক্ট্রনিক্স নিয়ে কাজ করা তোমার জন্য খুবই সহজ হয়ে যাবে। অন্যদিকে C++,যা দিয়ে সাধারণত Arduino,ESP তে কোড করা হয়, তা দিয়ে প্রোগ্রামিং করা বেশ কঠিন।

বর্তমানে Micropython ব্যবহার করে হবিস্ট,রিসার্চার এবং শিক্ষকরা কাজ করছেন এমন কি কমার্শিয়াল প্রোডাক্ট তৈরিতেও এটি ব্যবহার হচ্ছে।

ডেভেলপমেন্ট সেক্টরে কাজ শুরু করতে গেলেই আমাদের বেশির ভাগের মনে বার বার শুধু আসে “এই প্রোগ্রামিং ল্যাংগুয়েজটা ভালো হবে নাকি অন্যটা,এই টেকনোলজিটা শিখবো-নাকি ঐটা,এটার মতো মার্কেটে চাহিদা নাই-অইটাই সবাই চাচ্ছে”। এই সব অযথা চিন্তার লুপে পরেই দেখা যায় বেশির ভাগ স্টুডেন্টই সময় নষ্ট করে ফেলে,কাজের কাজ কিছুই হয় না।

একজন কাঠমিস্ত্রি ও লৌহমিস্ত্রি যখন কাজ করে তখন তারা অনেকগুলো টুল ব্যবহার করে,যেমনঃ হাতুরি। দুই জনের হাতুরি কিন্তু একই রকম না। লোহার কাজ করতে দরকার ভারী-মোটা হাতুড়ি আর কাঠের সুক্ষ কাজে দরকার ছোট হাতুড়ি।

প্রোগ্রামিং ল্যাংগুয়েজ একটা টুল মাত্র। যখন যে কাজের জন্য যেটা প্রয়োজন তখন যে টুল ব্যবহার করতে হবে।কি টুল ব্যবহার করলাম তার চাইতেও গুরুতবপূর্ণ হচ্ছে আমি সেটা কত ভালোভাবে ব্যবহার করলাম এবং কত ভালো প্রোজেক্ট তৈরি করলাম।

তাই মাইন্ডে ঘুরপাক খাওয়া সব অযথা চিন্তা বাদ দিয়ে নিজ হাতে কোডিং করা,প্রোজেক্ট তৈরিতে মন দাও।

**প্রিকিউজিটস**

|  |  |
| --- | --- |
| ESP32 DevKit V1 | ১ |
| Jumper Wire | – - - |
| Breadboard | ১ |
| LED (Mixed Color) | ১০ |
| DHT22 | ১ |
| Ultrasonic Sensor | ১ |
| Push Button | ৪ |
| Buzzer | ১ |
| Potentiometer | ১ |
| OLED Display | ১ |
| Motor Driver | ১ |
| MQ2 Gas Sensor | ১ |
| Relay Module | ১ |
| SD Card Module | ১ |
| Capacitive Soil Moisture Sensor | ১ |
| Motion Sensor | ১ |
| 4\*4 Matrix KeyPad | ১ |
| Resistor | ১০ |

তুমি আলাদা ভাবে এই সব কম্পোনেন্টস কিনতে পারো,অনলাইনে ইলেক্ট্রনিক্সের শপে পেয়ে যাবে।আর তুমি যদি চাও আরো সহজে IoT শিখতে তাহলে Bigganpur IoT Board টি কিনে নিতে পারো।এই বোর্ডটি এমন ভাবে ডিজাইন করা হয়েছে যাতে হার্ডওয়্যার এর কানেকশন দেয়া,আলাদা করে কম্পোনেন্ট কেনার ঝামেলা স্টুডেন্টদের নিতে না হয়,তারা যেন IoT এর কোডিং পার্টটুকুতে মনোযোগ দিতে পারে।

**পার্ট – ১**

**আইওটি ও ইলেক্ট্রনিক্স কনসেপ্টস**

**চ্যাপ্টার – শূণ্য**

**IoT এর আদ্যপান্ত**

**ইন্ডাস্ট্রিয়াল রিভুলেশন ও আইওটি**

“চতুর্থ শিল্পবিপ্লব” এই শব্দটা আজকাল আমরা প্রায় নিয়মিতই শুনতে পাই।বিশেষ করে কেউ যদি ভার্সিটি পড়ুয়া হয়ে থাকে এবং টেকনোলজি নিয়ে ঘাটা-ঘাটি করে তবে “চতুর্থ শিল্পবিপ্লবের জন্য আমাদের প্রস্তুত হতে হবে” এমন বক্তব্য সেমিনারগুলোতে গেলে প্রায়ই শুনা যায়।  
  
Internet of Things বা IoT হচ্ছে চতুর্থ শিল্পবিপ্লবের অনেকগুলো কম্পোনেন্টের মাঝে একটি যাতে করে মানুষের সরাসরি উপস্থিতি ছাড়াই ডাটা ইন্টারনেটের মাধ্যমে আদান-প্রদান করা হয়।যদি বইয়ের ভাষায় বলি, “IoT is an umbrella word for a set of hardware or devices that can communicate with one another(Sends Data) over the internet to form a system.”

IoT ডিভাইসগুলোর সাথে সাধারণত দুইটা কম্পোনেন্টস থাকেই –

১)সেন্সর(sensor) ২)একচুয়েটর(Actuator)

সেন্সর পরিবেশ থেকে বিভিন্ন ডাটা কালেক্ট করে।যেমনঃ টেম্পেরেচার,প্রেসার,হিউমেডিটি,কোন কিছুর উপস্থিতি ইত্যাদি বুঝার জন্য আলাদা সেন্সর পাওয়া যায়।

আর একচুয়েটর হলো ইলেক্ট্রনিকস কম্পোনেন্টস যা দিয়ে কোনো কিছু স্বয়ংক্রিয়ভাবে ঘুরানো বা যান্ত্রিকভাবে নিয়ন্ত্রণ করা হইয়।যেমনঃ মোটর,রোবটিক্স হ্যান্ড

IoT ডিভাইসের কাজ হচ্ছে সেন্সরের সাহায্যে পরিবেশ থেকে ডাটা নেয়া, কালেক্ট করা ডাটা নেটওয়ার্কের মাধ্যমে অন্য কোনো ডিভাইসে বা ডাটা বেজে পাঠানো,সেই ডাটা ইউজার কোনো ইন্টার্ফেইজ/সফটওয়্যার/ড্যাশবোর্ডের মাধ্যমে দেখবে এবং কোনো ইনপুট/কমান্ড দিবে যা অনুযায়ী একচুয়েটর কোনো কাজ করবে।

**আইওটি ডিভাইস**

আইওটি ডিভাইস দুই ধরণের হয়ঃ

১)প্রডাকশনের জন্য ব্যবহৃত ডিভাইসঃ

এই ডিভাইস গুলো বানিজ্যিক ব্যবহারের জন্য তৈরি করা বিক্রির উদ্দেশ্যে ।বেশির ভাগ সময়ই ডিভাইসগুলো যে ইন্ডাস্ট্রিতে ব্যবহার করা হবে ঐ এনভাইরনমেন্টের জন্য কাস্টম ভাবে তৈরি করা হয়। বিভিন্ন আইটি সার্ভিস প্রতিষ্ঠান ইন্ডাস্ট্রির জন্য ডিভাইসগুলো তৈরি করে দেয়। সরাসরি মাইক্রোকন্ট্রোলার ব্যবহার না করে কাস্টম CPU বা সার্কিটবোর্ড ব্যবহার করা হয়। ডিভাইসগুলো যেন সাইজে ছোট, দীর্ঘস্থায়ী হয় সেদিকে খেয়াল রাখা হয়।

২) ডেভেলপমেন্টের জন্য তৈরি ডিভাইসঃ

এই ডিভাইসগুলো সম্পূর্ণ শেখার স্বার্থে তৈরি করা হয়, বানিজ্যিক ব্যবহারের জন্য নয়। বিশ্ববিদ্যালয়ে থিসিস বা রিসার্চের কাজে কিংবা আইডিয়া কম্পিটিশনের জন্য যে প্রজেক্টগুলো তৈরি করা হয় তাও কিন্তু এই সেগমেন্টের আওতাভুক্ত। বিভিন্ন ডেভেলপার কিটস(Developer Kits) বাজারে পাওয়া যায়।শিক্ষার্থীরা এইগুলো দিয়ে শেখা শুরু করতে পারে। প্রোটোটাইপিং এর জন্য মাইক্রোকন্ট্রোলার(MCU যেমনঃ Arduino,ESP32), সিঙ্গেল বোর্ড কম্পিউটার(SBC যেমনঃ Raspberry PI), ব্রেডবোর্ড ইত্যাদি ব্যবহার করা হয়।

**এই বইয়ে আমরা যত প্রোজেক্ট করবো সবই ডেভেলপমেন্টের জন্য বা শেখার জন্য।**

**আইওটি এর এপ্লিকেশন**ব্যবহারের ভিত্তিতে আওইটি ডিভাইসগুলোকে মূলত চারটা গ্রুপে ভাগ করা হয়ঃ

১)কনজিওমার আইওটি (Consumer IoT): এই ডিভাইসগুলো সরাসরি কনজিওমাররা মার্কেট থেকে কিনবে এবং নিজে বা বাসা-বাড়িতে ব্যবহার করবে।যেমনঃ স্মার্ট স্পিকার, স্মার্ট হিটিং/কুলিং সিস্টেম,হোম অটোমেশন, রোবট ক্লিনার, পার্সোনাল হেলথ ট্রেকার-স্মার্ট ওয়াচ ইত্যাদি।

২) কমার্শিয়াল আইওটি (Commercial IoT): এই ডিভাইসগুলো অফিস বা কমার্শিয়াল বিল্ডিং-ব্যবসা প্রতিষ্ঠান,মার্কেটে ব্যবহার করা হইয়।যেমনঃ এটেন্ডেন্স সিস্টেম, মোশন ট্রেকার, সেইফটি ডিভাইস, ভেহিক্যাল ট্রেকিং ইত্যাদি।

৩) ইন্ডাস্ট্রিয়াল আইওটি (Industrial IoT): বড় ইন্ডাস্ট্রি মেমনঃ ফেক্টরি, ডিজিটাল এগ্রিকালচারে ব্যবহার করা হয়। প্রিডেক্টিভ মেইন্টেনেন্স, প্রোডাকশন মনিটরিং, সেইফটি মনিটরিং, টেম্পেরেচার-মায়েশ্চার কন্ট্রোলে এই ডিভাইসগুলো ব্যবহার করা হয়।

৪) ইনফ্রাস্ট্রাকচার আইওটি (Infrastructure IoT): যে ইনফ্রাস্ট্রাকচারগুলো সাধারণ মানুষ প্রতিদিন ব্যবহার করছে যেগুলো মনিটরিং করার জন্য ব্যবহার করা হয়।যেমনঃ স্মার্ট সিটি ,স্মার্ট পাওয়ার গ্রিড। স্মার্ট সিটিতে আওইটি ডিভাইসগুলো ব্যবহার করা হয় বিভিন্ন স্থানের ডাটা কালেক্ট করতে যাতে ট্রাফিং কন্ট্রোল করা যায়,পরিবেশ দূষণ কমানো যায়।

স্মার্ট পাওয়ার গ্রিডে লোড ডিমান্ড বুঝতে, গ্রিডের উপর চাপ কমাতে-খরচ কমাতে আইওটি ডিভাইসের মাধ্যমে ডাটা কালেক্ট করা হয়।

**কেন IoT শেখা উচিত**

এক বাক্যে এর উত্তর হচ্ছে,”কারণ এটা ভবিষ্যত”।এমন কী কেউ যদি প্রফেশনাল জীবনে এটা নিয়ে কাজ নাও করো তবে স্টুডেন্ট লাইফে অনেক ভালো লেভেলের প্রজেক্ট তৈরি করতে,থিসিসের-রিসার্চ পেপারের মান কয়েকগুণ বাড়িয়ে ফেলার জন্য হলেও তোমার এটা শেখা উচিত।তাছাড়া তুমি যদি এন্ট্রোপ্রেনোর হতে চাও, কোনো প্রবলেম সল্ভ করার জন্য সলিউশেন তৈরি করতে চাও বা শুধু মাত্র MVP(Minimal Viable Product) বানিয়ে কোনো বিজনেজ কম্পিটিশনে হাজার টাকা পুরস্কার এবং সিভি’তে লিখার মতো অসাধারণ কিছু করতে চাও তাহলে IoT তোমার জন্য!

IoT শুধু ডাটা নিয়ে কোনো ড্যাশবোর্ডে দেখিয়ে দেয়া পর্যন্তই সীমাবদ্ধ না।একবার ডাটা পেয়ে যাওয়ার পর সেই ডাটা দিয়ে তৈরি করতে পারবে মেশিন লার্নিং প্রজেক্ট,করতে পারবে থিসিস-রিসার্চ ওয়ার্ক।এর সাথে চলে আসে Digital Twin,Automation,Security ইত্যাদি বিষয়।

তুমি যে সাইন্সের যে সাব্জেক্টেই পড়ো না কেন,একাডেমিক কাজে তোমাকে লোকাল ডাটা কালেক করতে হবেই রিসার্চের জন্য।যত বেশি কোয়ালিটিফুল ডাটা পাওয়া যাবে তত সুযোগ থাকবে সেটা ভালো ভাবে এনালাইসিস করার।আর এই অটোমেশনের কাজ করতে পারে IoT।

**বাংলাদেশ ও বিশ্বে IoT কীভাবে ব্যবহার হচ্ছে**

তোমারা নিশ্চই মেডিকেল ভর্তি পরীক্ষার প্রশ্নপত্র ফাঁসের ঘটনার কথা শুনেছো।দেশের মেধাবীদের যোগ্যতা নির্ধারণের মতো গুরুতবপূর্ণ এই পরীক্ষায় কিছু অসৎ মানুষের জন্য হয়ে উঠেছে প্রশ্নবিদ্ধ এবং এটা একটা নেশনাল ডিজেস্টারও বলা চলে।

বার চেষ্টার পরও কর্তৃপক্ষ যখন প্রশ্ন ফাঁস ঠেকাটে ব্যার্থ হচ্ছিল তখন আমাদের দেশেরই এক প্রযুক্রি প্রতিষ্ঠান নিয়ে আসে “Smart Box” নামক একটা সলিউশন।

প্রশ্নপত্র স্মার্টবক্সে প্যাক করার পর তা আনলক করার ক্ষমতা থাকে নিয়ন্ত্রিত।বক্সগুলো খুলতে ব্যবহার করা হয় বিশেষ পিন নম্বর। যা কেন্দ্রগুলোতে পাঠানো হয় পরীক্ষা শুরুর আধাঘণ্টা কিংবা তার কম সময় আগে।এছাড়া থাকে বিশেষ অ্যালার্ম সিস্টেম। মাঝপথে কেউ প্রশ্নপত্রের বাক্স খুলতে চেষ্টা করলে সিগন্যাল যাবে নিয়ন্ত্রণ কক্ষে। ফলে প্রশ্ন ঢুকিয়ে একেকটি বক্স প্রস্তুত করা থেকে শুরু করে তা পরীক্ষার হলে খুলে নেওয়া পর্যন্ত প্রতি সেকেন্ডে রিয়েল টাইম মনিটরিং করার সুবিধা ও স্বচ্ছতা নিশ্চিত করা সম্ভব।

এই স্মার্টবক্স আর কিছুই না,একটা IoT ডিভাইস! বাংলাদেশে এমন অনেক IoT সলিউশন তৈরি করছে Bondstein Technologies Limited নামের এই বাংলাদেশী কম্পানিটি।তাঁদের যাত্রা শুরুর গল্পটাও কিন্তু অসাধারণ।গুগল করে এই কম্পানিটি সম্পর্কে আরো জানতে পারো।

তোমাকে যদি বলি বাংলাদেশেরই একটি প্রতিষ্ঠান তাঁদের IoT ডিভাইসের মাধ্যমে পবিত্র মক্কা নগরীর পানির সমস্যা সমাধান করেছে,তাহলে নিশ্চই অবাক হবে।ঠিক এই কাজটাই করেছে DataSoft।

রকমারি,উদ্ভাস-উন্মেষ,অন্য রকম পাঠশালার কর্ণধার সোহাগ ভাইকে তো চিনবেই।তুমি কি জানো সোহাগ ভাইয়ের মূল প্রতিষ্ঠান অন্যরকম গ্রুপের একটা বড় বিজনেজ হচ্ছে “অন্য রকম ইলেক্ট্রনিক্স (ORELCO)” এবং উনারা যানবাহন চুরি রোধের জন্য “প্রহরী” নামে Vehicle Tracking Device, ‌যা একধরণের IoT ডিভাইস, তৈরি করছে।

সরকার বাংলাদেশের কোন খাতে IoT ব্যবহার করতে চায় তা নিয়ে একটা ডিটেইল পরিকল্পনা প্রকাশ করেছে National Internet of Things Strategy Bangladesh নামে ২০২০ সালে।সেখানে Buildings and home management, Healthcare, Transportation, Industries, Agriculture, Infrastructure, Smart Cities, Power, Environment এই নয়টি খাত নিয়ে আলোচনা করা হয়েছে।গুগল করলেই এই রিপোর্টটি পেয়ে যাবে।

বাইরের বিশ্বের দিকে যদি তাকাই তাহলে Amazon এর Alexa কে তো চিনবেই

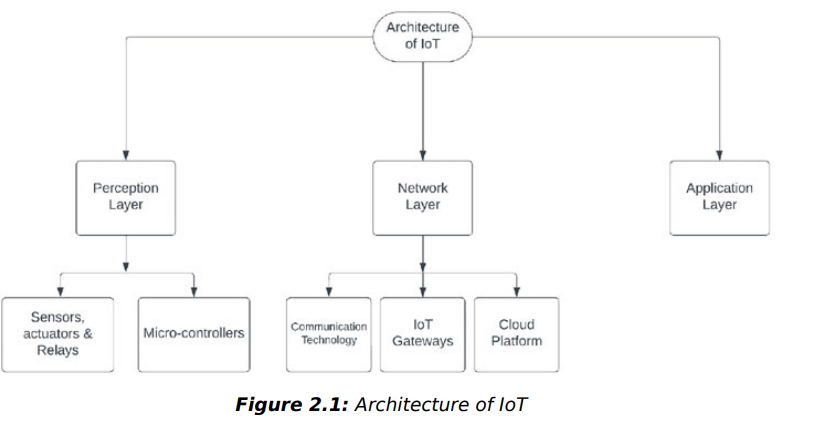
**চ্যাপ্টার – ১**

**IoT ডিভাইসের সব কিছু**

**IoT আর্কিটেকচার**

তোমার হাতের মোবাইল ফোনে তুমি যখন কথা বলো বা ইন্টারনেটে ব্রাউজিং করো মূহূর্তের মাঝেই হাজার-হাজার,লাখ-লাখ মেইল দূরে থাকা আরেকটি ডিভাইজের সাথে কিন্তু ডাটা আদান-প্রদান হয়ে যায় তাও আবার রিয়েল-টাইমে।কখনো কি ভেবে দেখেছো কোনো রকম ডাটা নষ্ট না হয়ে,রিয়েল-টাইমে তৎক্ষণাত কিভাবে এতো কিছু সম্ভব হয়। এর পেছনে কিন্তু জটিল একে আর্কিটেকচার কাজ করে।  
  
IoT এর আর্কিটেকচার হলো কিভাবে ফিল্ডে থাকা ডিভাইসগুলো(IoT Edge Device) থেকে ডাটা কালেক্ট হবে এবং কোনো ডাটা নষ্ট না হয়ে,সিকিউর ভাবে রিয়েল টাইমে ট্রান্সমিট হয়ে ক্লাউডে যাবে।

IoT আর্কিটেকচারের তিনটা মেইন পার্ট আছে -

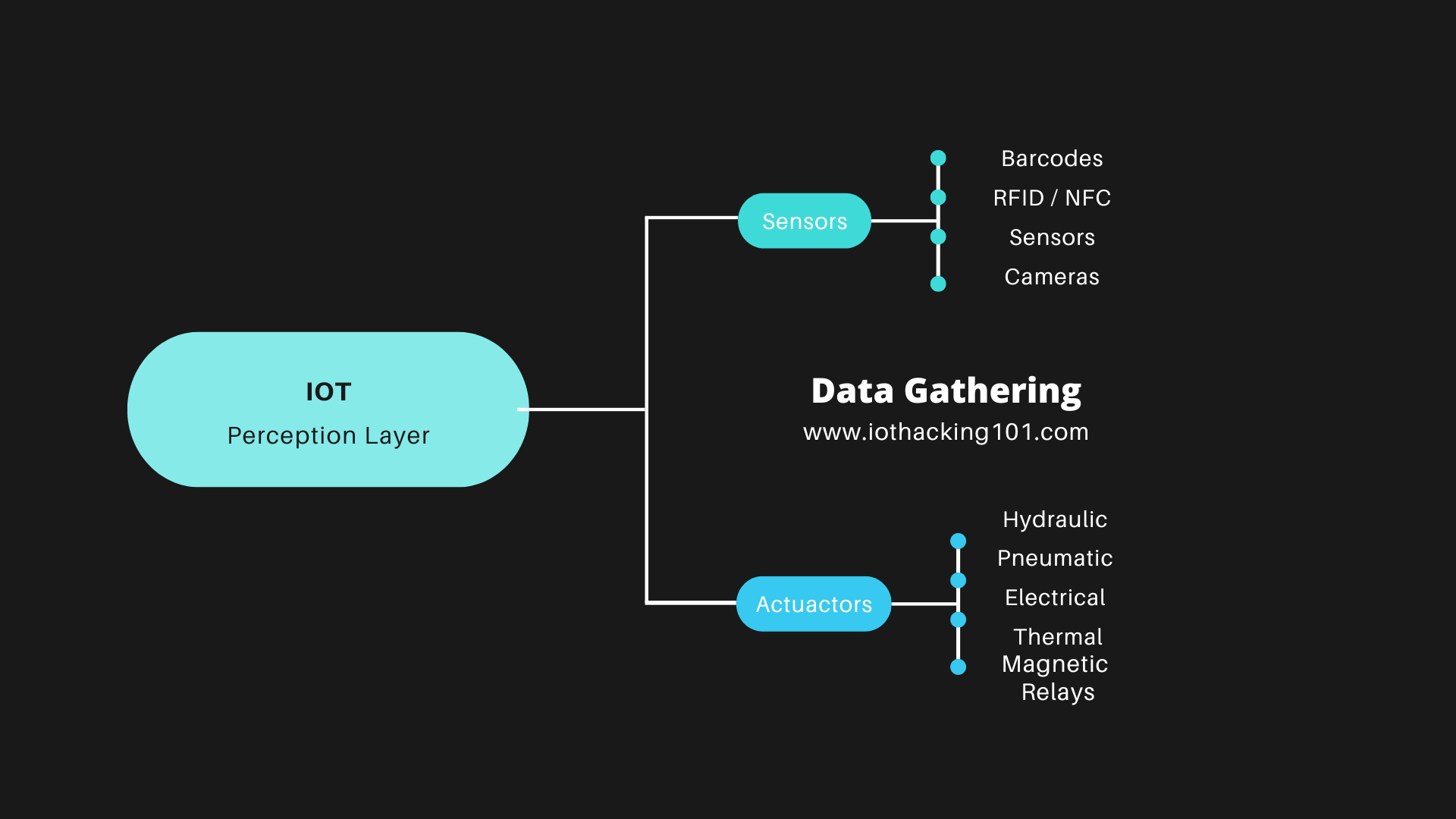


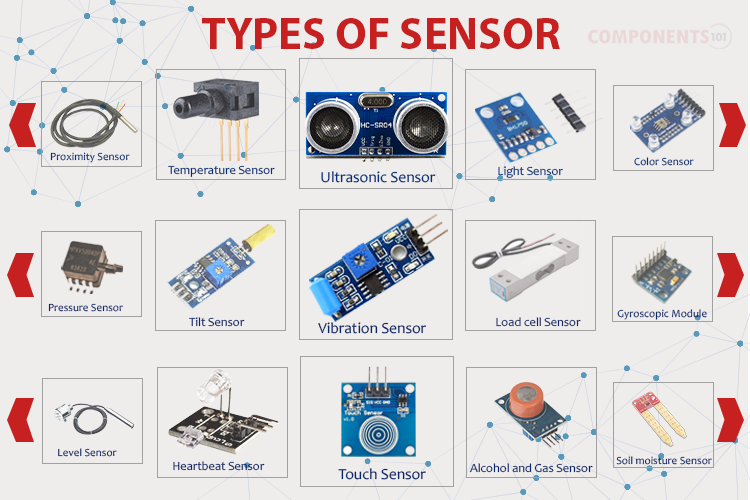
Perception Layer:

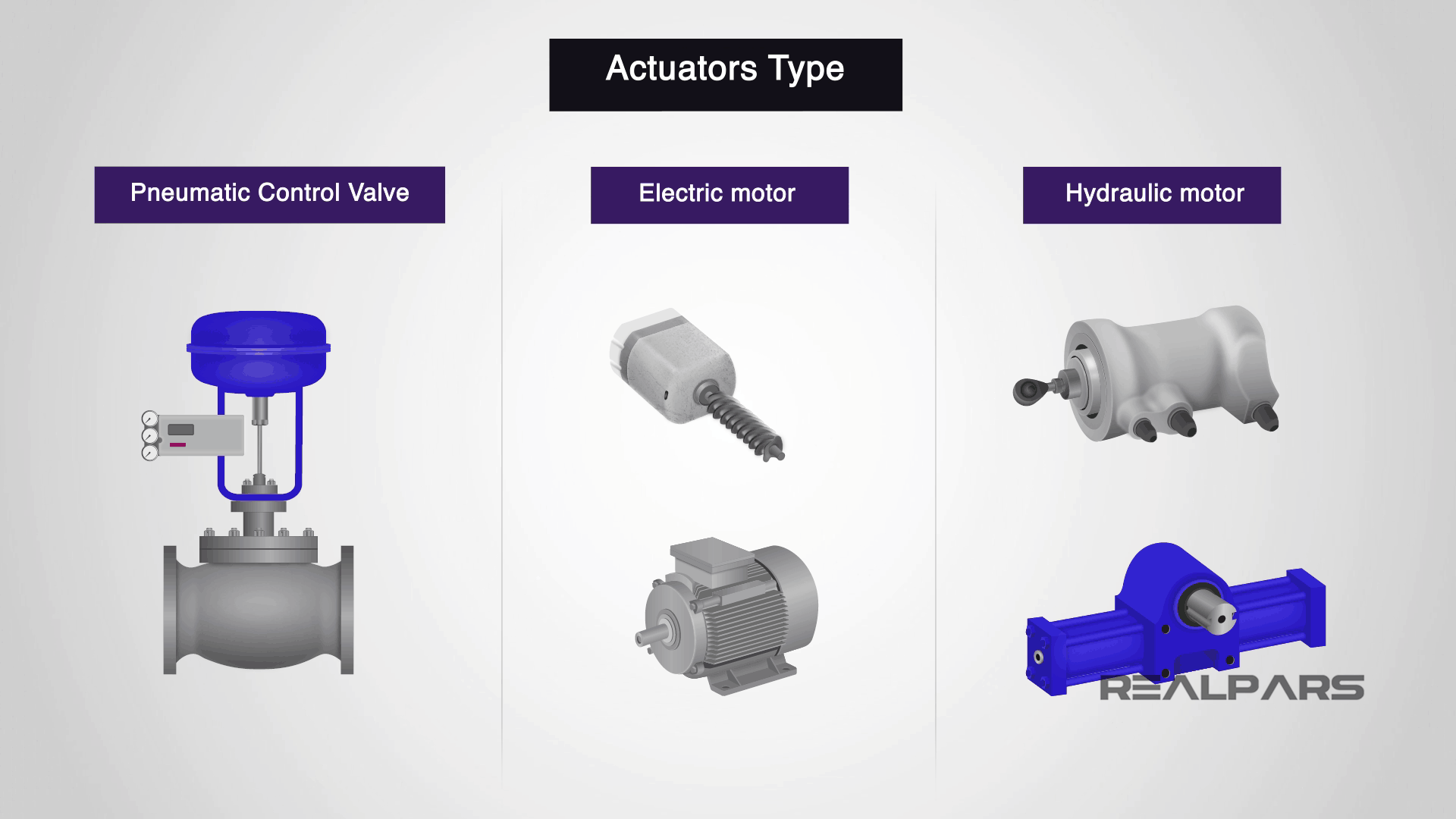
এটা হচ্ছে IoT এর ফিজিকাল বা হার্ডওয়্যার লেয়ার যেখানে থাকে সেন্সর(Sensor),একচুয়েটর(Actuator),মাইক্রোকন্ট্রোলার,রিলে(Relay)।

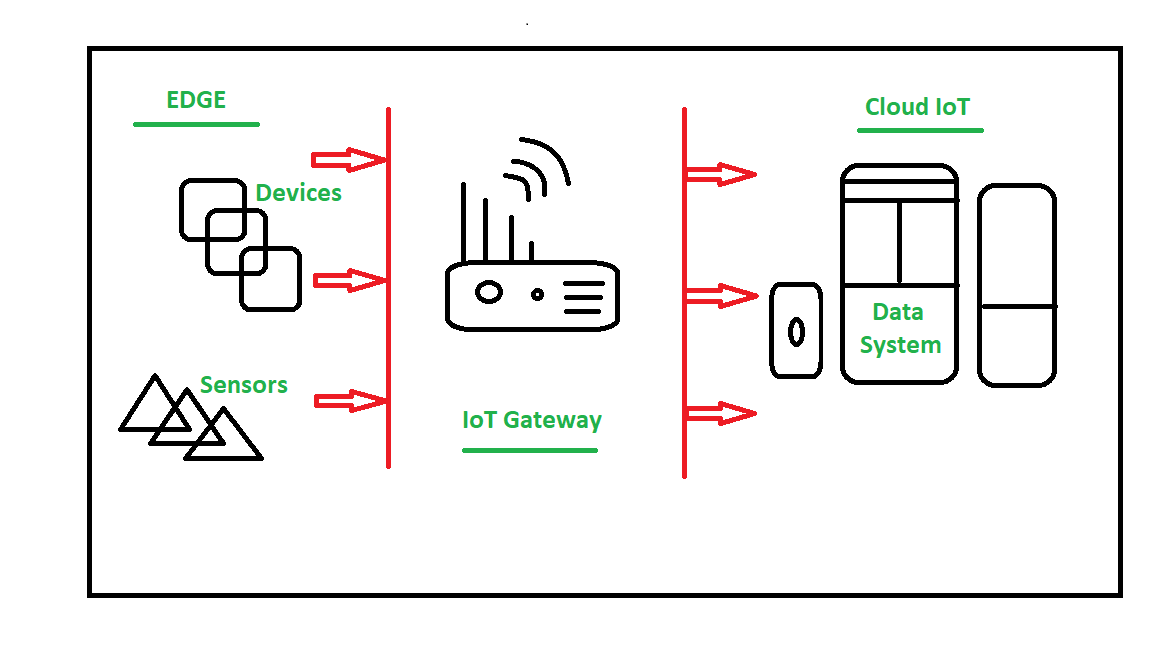
সেন্সরের কাজ হচ্ছে পরিবেশ থেকে ডাটা কালেক্ট করা।মানুষ যেমন কান,চোখ,জিহবা,নাক দিয়ে পরিবেশে কোনো কিছুর উপস্থিতি,শব্দ, স্বাদ, ঘ্রাণ অনুভব(Sense) করতে পায় তেমনি নির্দিষ্ট সেন্সর ইলেক্ট্রিক্যাল সিগন্যাল পাঠিয়ে মেশিনকে কোনো কিছুর উপস্থিতি/অনুপস্থিতি বুঝায়।

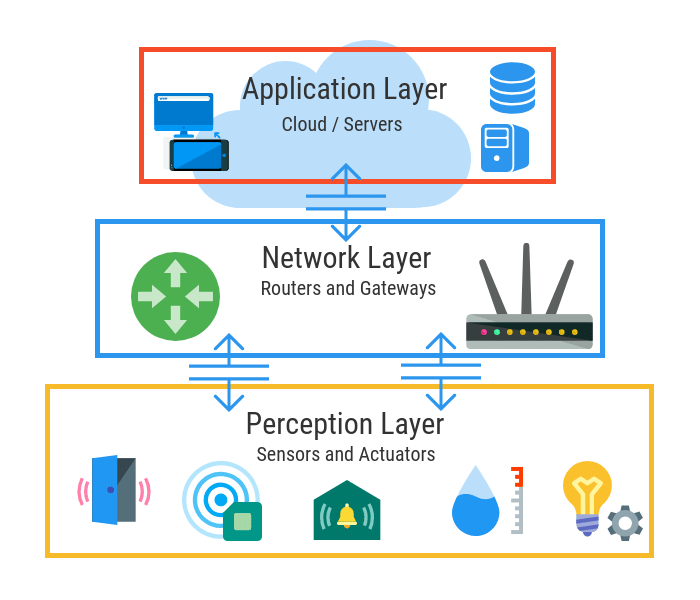
একচুয়েটরের কাজ হচ্ছে নির্দেশ অনুযায়ী কোনো নির্দিষ্ট কাজ করা।সেন্সর যেমন সেন্স করার কাজ করে,একচুয়েটর তেমনি পেশির মতো কাজ করে।

মাইক্রোকন্ট্রোলার হচ্ছে পুরো সিস্টেমের ব্রেইন,যা আমাদের দেয়া কোড এক্সিকিউট করে।  
রিলে হচ্ছে ইলেক্ট্রো-ম্যাগনেটিক সুইচ।সাধারণত আমরা ঘরে যে সুইচ ব্যবহার করি তা হাত দিয়ে অন-অফ করতে হয়।রিলের সুবিধা হচ্ছে এটা দিয়ে কোনো লোড অন-অফ করতে ফিজিক্যাল কন্টাক্ট প্রয়োজন হয় না,ইলেক্ট্রিক্যাল সিগন্যাল,যা মাইক্রোকন্ট্রোলার থেকে দেয়া হয়,তা দিয়ে অন-অফ করা যায়।   




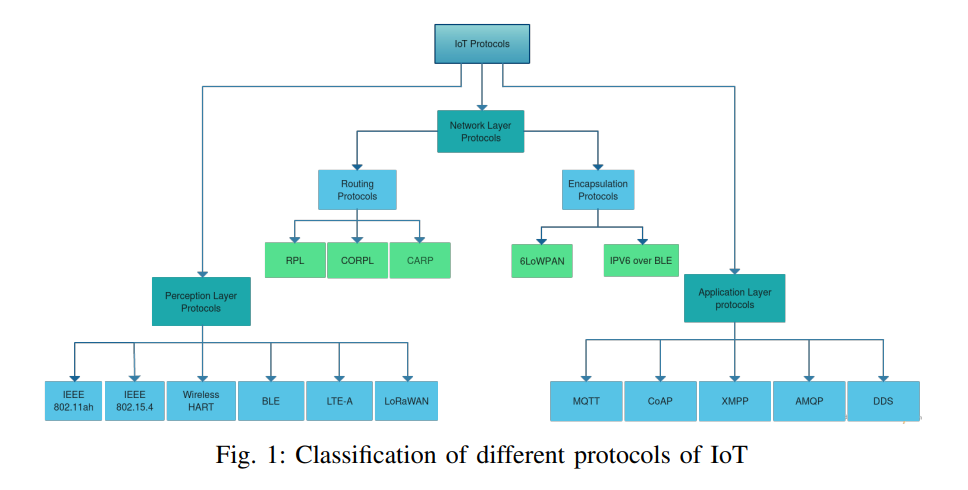


Network Layer:  
এই লেয়ারের কাজ হচ্ছে সেন্সর থেকে যে ডাটা সংগ্রহ করা হলো তা কানেক্টেড ডিভাইসগুলোর মাঝে আদান-প্রদান করা।এই লেয়ার হার্ডওয়্যার আর ক্লাউডের মাঝে সংযোগ তৈরি করে। IoT এর জন্য নেটওয়ার্ক লেয়ার খুবই গুরুত্বপূর্ণ কারণ এর উপর নির্ভর করে যেন ঠিক ভাবে যেন ডাটা ট্রান্সমিট।  
  
প্রতিটা IoT ডিভাইসের একটা ইউনিক আইডেন্টিফায়ার (Unique Identifier) থাকে যাতে করে একে আলাদা করা যায়,একে বলে IP Address। এছাড়াও আইপি এড্রেস কেমন হবে,কিভাবে ডাটা আদান-প্রদান হবে তা বিভিন্ন প্রোটোকল মেনে চলে যেমনঃ IPv4 , IPv6 ।  
  
ডাটা কত স্পিডে আদান-প্রদান হবে,কত দূরত্বে হবে ইত্যাদির উপর নির্ভর করে বিভিন্ন কমিউনিকেশন টেকনোলজি আছে।যেমনঃ Wi-Fi, Bluetooth, Zigbee, LoRaWAN,Cellular Networks ইত্যাদি।  
  
আইওটি গেইটওয়ে বিভিন্ন কমিউনিকেশন টেকনোলজির মাঝে মধ্যস্থতাকারী হিসেবে কাজ করে।গেইটওয়ের সাহায্যে device-to-device বা device-to-cloud কমিউনিকেশন সম্ভব হইয়।গেইটওয়ে কোনো হার্ডওয়্যার ডিভাইস কিংবা সফটওয়্যারও হতে পারে।  
  
  
আইওটি ডিভাইসগুলো এনার্জি মেনেজ করা একটা গুরুতবপূর্ণ কাজ।ডেভেলপমেন্টের সময় খেয়ার রাখতে হয় IoT ডিভাইসগুলো যেন ব্যাটারি থেকে খুবই কম পাওয়ার কনসাপশন করে অর্থ্যাত ডিভাইসগুলো Energy Constrained।কিন্তু ডিভাইসগুলো যদি সরাসরি ইন্টারনেট/ক্লাউডের সাথে কানেক্ট হয় তাহলে অনেক এনার্জির দরকার পরে কমিউনিকেশন টেকনোলজিগুলোর।  
  
এর সমাধান হচ্ছে ডিভাইসগুলো প্রথমে ছোট রেঞ্জের নেটওয়ার্ক টেকনোলজিগুলো যেগুলো খুবই কম এনার্জি কনসাপ্ট করে যেমনঃ Bluetooth,ZigBee এগুলোর মাধ্যমে গেটওয়্যার সাথে কানেক্ট হয়।গেইটওয়েগুলো এনার্জি নিয়ে কোনো লিমিটেশন নেই,এগুলো আইওটি ডিভাইসগুলোকে ইন্টারনেট/ক্লাউডের সাথে কানেক্ট করে দেয়।  
এই সবগুলো টপিক আমরা আইটি প্রোজেক্ট বানানোর সময় প্র্যাক্টিক্যালি দেখতে পাবো।  
  
**ক্লাউড কম্পিউটিংঃ**ক্লাউড কম্পিটিং বললেই আমাদের মাথায় আসে ক্লাউডে কোনো স্টোরেজ প্লেজ আমাদের ফাইল,সফটওয়্যার রাখার জন্য।আসলে স্টোরেজ ক্লাউড কম্পিউটিং একটা ছোট্ট অংশ মাত্র।সহজ ভাষায় ক্লাউড বলতে বুঝাচ্ছি সার্ভার যেখানে আমাদের এপ্লিকেশনটা রান হয়। ক্লাউড কম্পিটিং ভালো ভাবে বুঝতে হলে এর তিনটা টাইপ ভালো ভাবে বুঝতে হবেঃ SaaS(Software as a Service),PaaS (Platform as a Service) এবং IaaS(Infrastructure as a Service) ।  
  
মনে করো তুমি কোনো এক রেস্টুরেন্ট এর মালিক।তোমার রেস্টুরেন্ট এর সকল হিসাব-নিকাশ ম্যানেজ করার জন্য একটি সফটওয়্যার/অ্যাপ্লিকেশনের প্রয়োজন।তুমি সফটওয়্যারটা বানালেও।এখন এটা ওয়েবে ডিপ্লয় করতে হবে যেন যেকোনো জায়গা থেকে তুমি এটা চালাতে পারো।এখন তুমি এক কাজ করতে পারো,এই অ্যাপ্লিকেশন ডিপ্লয় করার জন্য প্রয়োজনীয় সার্ভার কম্পিউটার নিজে কিনে অফিসে বসাতে পারো,এর রানটাইম,অপারেটিং সিস্টেম,ডাটা প্রয়োজনীয় সব কিছু ম্যানেজ করার জন্য ডেভেলপার হায়ার করতে পারো।এই সব কিছুর জন্য তোমাকে এক আলাদা অফিস সেটআপ করতে হবে,যন্ত্রপাতি কিনতে হবে,ডেভেলপারের বেতন দিতে হবে মাসে মাসে। সমস্যাটা কি বুঝতে পারছো??  
  
হ্যাঁ,ঠিক ধরেছ। খরচ করতে হবে অনেক টাকা,থাকতে হবে টেকনোলজিতে দক্ষতা কিন্তু তারপরও এটা কিন্তু গুটি কয়েক বড় কম্পানি বাদে কারোর জন্যই সম্ভব না।এরই সমাধান আনে ক্লাউড কম্পিউটিং।  
  
 ধরো কোনো টেকনোলজি কম্পানি আছে যাদের কাজই হচ্ছে রেস্টুরেন্টের জন্য মেনেজমেন্ট সফটওয়্যার বানানো।তারা নিজেদের অ্যাপ্লিকেশন বানানো আছে,সার্ভার,ডেভেলপার এই সব কিছু তারাই ম্যানেজ করে।তুমি শুধু সাবস্ক্রিপশন কিনবে আর অ্যাপ্লিকেশন ব্যবহার করবে।তোমার মতো আরো হাজারো রেস্টুরেন্ট একই অ্যাপ্লিকেশনের সাবস্ক্রিপশন নিয়ে একাউন্ট খুলে ব্যবহার করবে,অ্যাপ্লিকেশন ডিজাইন,ম্যানেজের সব কাজ ঐ টেক কম্পানির।এই মডেলকে বলে SaaS(Software as a Service)।  
  
SaaS এর লিমিটেশন হচ্ছে এখানে অ্যাপ্লিকেশন আমরা নিজেরা বানাই না,তাই এর কাস্টমাইজেশন সম্ভব না।টেক কম্পানি যেভাবে দিবে তাই ব্যবহার করতে হবে।কিন্তু আমি যদি নিজে অ্যাপ্লিকেশন বানাই আর সেটা ডিপ্লয় করতে চাই কিন্তু সার্ভার সেটআপ,মেইনটেনেন্টস এর ঝামেলায় না যেতে চাই তাহলে আসে PaaS (Platform as a Service)।  
  
PaaS এ তোমাকে একটা রেডি-টু-ইউজ প্ল্যাটফর্ম দেয়া হবে যেখানে তুমি তোমার তৈরি অ্যাপ্লিকেশন ডিপ্লয় করতে পা্রবে,ডাটা মেনেজমেন্ট করতে পারবে। বাকি সব কিছু সার্ভার,রানটাইম,অপারেটিং সিস্টেম ইত্যাদি ম্যানেজ করার দায়িত্ব PaaS কম্পানির।তুমি শুধু মাস/বছর শেষে কম্পানি যা চার্জ করবে সে টাকাটা দিয়ে দিবে।  
  
IaaS(Infrastructure as a Service) আরেকটু অ্যাডভান্স ফেসিলিটি দেয়,এখানে তুমি সরাসরি Infrastructure-প্রয়োজনীয় হার্ডওয়্যারটাই ভাড়া নিতে পারবে।ধরো তোমার কোনো বড় কোনো মেশিন লার্নিং মডেল রান করতে হবে,এর জন্য ভালো কম্পিউটার দরকার যেটায় স্টোরেজ বেশি অন্যান্য কেপাবিলিটি বেশি।কিন্তু এখন নতুন কম্পিউটার কেনার টাকা নেই কিন্তু এমাজন,মাইক্রোসফট,গুগল ওদের কাছে কিন্তু অনেক পাওয়ারফুল কম্পিউটার,স্টোরেজ,নেটওয়ার্ক ইনফ্রাস্ট্রাকচার পরে আছে।তারা এই হার্ডওয়্যারগুলো ভাড়া দেয় যেখানে তুমি তোমার ভার্চুয়াল কম্পিউটার সেটআপ করতে পারবে।সেই ভার্চুয়াল কম্পিউটারের তোমার সুবিধা মতো স্টোরেজ নিতে পারবে,প্রয়োজনীয় সফটওয়্যার ইন্সটল করতে পারবে - একটি আসল কম্পিউটারে যা করা যায় তার সবকিছু ভার্চুয়াল ক্মপিউটারে করা সম্ভব।মাস শেষে ঠিক যতটুকু রিসোর্স তুমি ব্যবহার করেছে ঠিক তত টুকুর টাকাই তুমি পে করবে।  
  
এটাই হচ্ছে IaaS ।জনপ্রিয় IaaS এর মাঝে আছে Microsoft Azure, Amazon এর AWS ইত্যাদি  
  
Application Layer:  
হার্ডওয়্যার অর্থ্যাত perception layer থেকে ডাটা কালেক্ট হলো,network layer এর মাধ্যমে ডাটা ক্লাউডে পাঠিয়ে দিলাম।এবার এই ডাটাকে আইওটি ডিভাইস ব্যবহারকারীদের কাছে উপস্থাপনের পালা।ব্যবহারকারী মানে আমাদের কাস্টমারদের কাছে কিন্তু সরাসরি ডাটাটা(Raw Data) দিয়ে দিলে হবে না।  
  
তাদের কাছে সুন্দর ভাবে উপস্থাপন করতে হবে,একটি ইউজার ইন্টারফেইজ থাকতে হবে যেমনঃ ওয়েবসাইট,মোবাইল ডেক্সটপ অ্যাপ।গ্রাফ আকারে ডাটাগুলো দেখাতে হবে।ইউজার যদি কোনো একশন নিতে চায় তার জন্য বাটন থাকতে হবে।এই সবকিছুই হচ্ছে আইওটির শেষ লেয়ার- অ্যাপ্লিকেশন লেয়ারের কাজ।  
  
এই লেয়ারে যা যা থাকতে পারেঃ User Interface,Data Visualization and Analytics,Remote Control and Management,Notifications and Alerts,Security and Access Control,User Authentication and Identity Management,APIs (Application Programming Interfaces) ইত্যাদি।

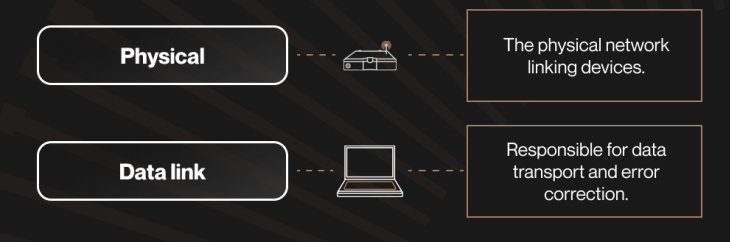


**IoT এর 3V Data Model**

আইওটি ডিভাসে যে ধরণের ডাটা সংগ্রহ করা হয় আর প্রসেস করে আদান-প্রদান করা হয় সে ডাটা প্রকৃতি কেমন এটা বুঝার জন্য 3V Data Model ব্যবহার করা হয়। 3V হচ্ছে তিনটা V:Volume, Velocity, and Variety। এই তিনটার উপর নির্ভর করে আইওটি ডিভাইসটা কত কমপ্লেক্স হবে।  
  
Volume: কি পরিমাণ ডাটা আদান-প্রদান হবে তার পরিমাণ।কানেক্টেট ডিভাইসের পরিমাণ আর সেন্সর দিন দিন যত বাড়বে ডাটার পরিমাণও তত বাড়বে।সেই অনুযায়ী ডাটা স্টোর করা,এনালাইসিস আর প্রসেস করাও জটিল হবে,ইনফ্রাস্ট্রাকচার উন্নত করতে হবে যাতে ইফিসিয়েন্সি ঠিক থাকে।  
  
Velocity: কত দ্রুত ডাটা আদান-প্রদান হবে।যত দ্রুততার সাথে রিয়েল টাইমে ডাটা-প্রদান করতে হবে আইওটি ডিভাইসগুলোকেও তত রিসপনসিভ আর ইফিসিয়েন্ট হতে হবে।  
  
Variety: কত ভিন্ন ধরণের ডাটা(Types এবং Formats) নিয়ে আইওটি ডিভাইস কাজ করবে।সেন্সরের উপর নির্ভর করে বিভিন্ন ধরণের ডাটা আসতে পারে।যেমনঃ টেক্সট,নাম্বার,অডিও,ভিডিও,টাইমস্টেম্প।বিভিন্ন ধরণের ডাটা হ্যান্ডেল করা কিংবা প্রসেস করার জন্য যথেষ্ট সক্ষমতা আইওটি ডিভাইসের থাকতে হবে।  
  
আইটি ডিভাইস ডেভেলপ করার সময় এই বিষয়গুলো খেয়াল রাখতে হবে যাতে করে ঠিক ভাবে ডাটা ম্যানেজ করা যায় সে সাথে বিজনেজ,ইন্ডাস্ট্রির জন্য ডাটাগুলো থেকে মূল্যবান ইনসাইট বের করে আনা যায়।আইওটির মূল লক্ষ্যই হচ্ছে বিজনেজের জন্য গুরুত্বপূর্ণ ইনসাইট বের করে আনা যাতে করে ম্যানেজমেন্ট ডাটা-ড্রিভেন সিদ্ধান্ত নিতে পারে,নিজেদের প্রফিট আরো বাড়াতে পারে।  
  
  
 **IoT প্রোটোকলস**

ইন্টারনেটে যুক্ত ডিভাইসগুলো নিজেদের মাঝে ডাটা আদান-প্রদান করতে অর্থ্যাত মেশিন টু মেশিন কমিউনিকেশনের জন্য কিছু নিয়ম এবং স্যান্ডার্ড আছে।এই স্যান্ডার্ড/নিয়ম-নীতিগুলোকে ইন্টারনেট প্রোটোকল(Internet Protocol - IP) বলে।  
  
IoT ডিভাইস/ক্লাইন্ট/নোডস(Nodes) গুলো প্রথমে কোনো বেইস ষ্টেশন/রাউটারস(Routers) বা গেইটওয়ের সাথে যুক্ত হয় এরপর গেইটওয়ের সাহায্যে ক্লাউডের সাথে যুক্ত হয়। এই পুরো প্রসেস এর মাঝে মেশিনগুলোর মাঝে কমিউনিকেশন বা ডাটা-আদান প্রদান করতে হইয়।হঠাৎ কমিউনিকেশনে কোনো সমস্যা হলে ডিভাইসগুলো কিভাবে তা সমাধান করবে তার জন্য একটি নির্দিষ্ট নিয়ম থাকতে হবে। এই নিয়মগুলোই হচ্ছে প্রোটোকলস।যেমনঃ ওয়েবসাইট ব্রাউজিং এর জন্য HTTP প্রোটোকল ব্যবহার করা হইয়।যখন কেউ ব্রাউজারে https://facebook.com/ লিখে সার্চ করে তখন সে আসলে ফেসবুকের সার্ভারের কাছে request পাঠায় facebook নামের একটা ওয়েব পেইজের জন্য।  
  
IoT প্রোটোকল কোনো একটি প্রোটোকল না।আইওটি ডিভাইসের তিনটা আলাদা লেয়ারের জন্য আলাদা আলাদা প্রোটোকল আছে।  


**Perception Layer Protocols:**এর মাঝে আছে Data Link Layer Protocols এবং Physical Layer। Physical Layer এ বিভিন্ন হার্ডওয়্যার কমিউনিকেশন ডিভাইসগুলো থাকে আর Data Link এর কাজ হচ্ছে ডাটাগুলো ঠিক ভাবে আদান-প্রদান করা,কোনো error আসলে সেটা হ্যান্ডেল করা।

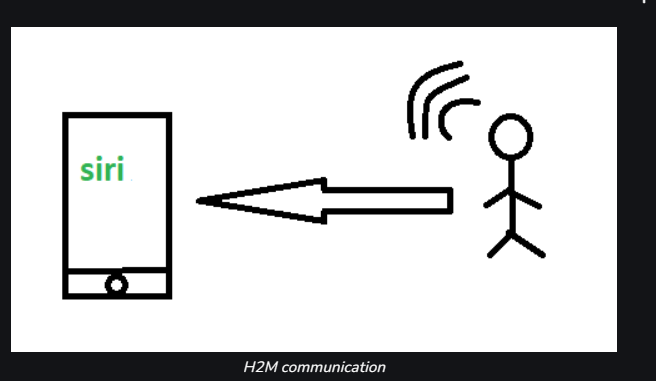


* Wi-Fi
* GSM
* Zigbee
* Z-wave
* BLE
* LoRa
* RF
* NB-IoT
* NFC

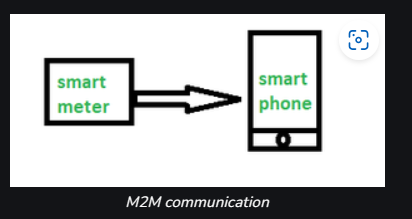
একটা বিষয় খেয়াল করো,এই লেয়ারে প্রোটোকল কি হবে তা কিন্তু হার্ডওয়্যার নির্ভর।যেমনঃ ESP32 তে Wi-Fi ,Bluetooth টেকনোলজি আছে।তাই আমরা যখন Wi-Fi ব্যবহার করবো তখন ঐ রিলেটেড প্রোটোকল নিয়ে কাজ করতে হবে।

**Network Layer Protocols:**  
এই লেয়ারে থাকে Routing Protocols আর Encapsulation Protocols । বিভিন্ন রাউটারের মাঝে রাউটিং কিভাবে সেই ইনফোরমেশনগুলো এক্সচেঞ্জ করে Routing Protocols আর Encapsulation Protocols এর কাজ হচ্ছে Data link layer protocol এর সাথে network layer protocol যুক্ত করা।  
  
**Application Layer Protocols:**  
এই প্রোটোকলগুলোর কাজ হচ্ছে মেশিনগুলোর মাঝে যেন একটিভ কমিউনিকেশন হয়,কোনো রকম ডাটা লস ছাড়াই।যখন ক্লাইন্ট সার্ভারের কাছে রিকোয়েস্ট পাঠায় তখন একটি সেশন,সার্ভার তখন রিকোয়েস্টের উত্তরে Respond করে।যতক্ষণ এই রিকোয়েস্ট-রেসপন্ড চলতে থাকে সেটাকে একটি Communication Session বলে।Application Layer Protocols এর কাজ হচ্ছে এই সেশনগুলো যেন ঠিক ভাবে হইয়।তাই একে Session Layer Protocols ও বলে।  
MQTT, HTTP,SMQTT, AMQP,CoAP, XMPP,DDS  
  
এই বইয়ের IoT Development পার্টে আইওটি প্রোজেক্ট বানানোর সময় MQTT প্রোটোকল নিয়ে আমরা বিস্তারিত শিখবো।   
  
  
 **কমিউনিকেশনের ধরণ**

আইওটিতে তিন ধরণের কমিউনিকেশন হতে পারেঃ-  
**১)Human to Machine (H2M) Communication:**   
এই ধরণের কমিউনিকেশনে মানুষ আইওটি ডিভাইস (মেশিন) এ কোনো ইনপুট দেয়।মেশিন সেই ইনপুট গ্রহণ করে কোনো সেন্সর দিয়ে এবং প্রসেস করে একচুয়েটর চলায় বা স্ক্রিনে টেক্সট আকারে রেসপন্স করে।যেমনঃ Facial recognition, Biometric Attendance system,voice recognition। এই ধরণের কাজে কিন্তু হার্ডওয়্যার এবং সফটওয়্যার একসাথে কাজ করে।

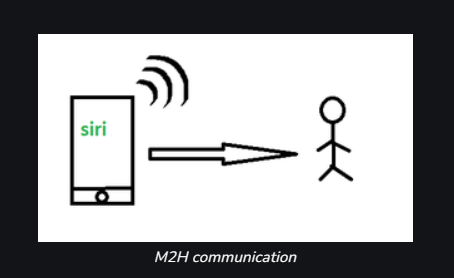
  
  
**২)Machine to Machine (M2M) communication:**

এই কমিউনিকেশনে ইনফোরমেশন আদান-প্রদান হয় দুই বা একাধিক মেশিনের মাঝে।এই কমিউনিকেশনে কোনো মানুষের হস্তক্ষেপ প্রয়োজন হয় না। M2M কমিউনিকেশন গুরুত্বপূর্ণ কারণ এর মাধ্যমে ইন্ডাস্ট্রিয়াল অটোমেশন সম্ভব।

মেশিনগুলো তারের সাহায্যে বা তার ছাড়া (Wireless) কানেক্টেড থাকতে পারে।

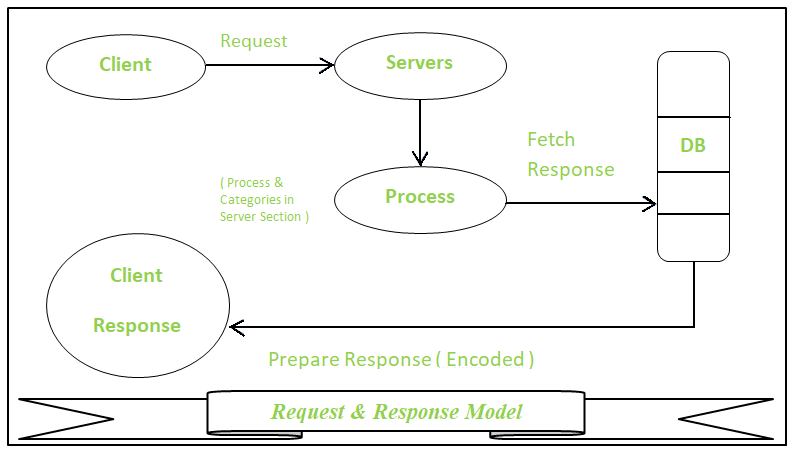
**৩)Machine to Human (M2H) communication:**

এখানেও মেশিন মানুষের সাথে ইন্টারেক্ট করে তবে পার্থক্য হচ্ছে মানুষ সরাসরি ইনপুট দেয় না,মেশিন কোনো ইনফরমেশনের সাপেক্ষে ট্রিগার হয় এবং মানুষকে গাইড করে যে কি করতে হবে।যেমনঃ Fire Alarms, Traffic Light,Fitness band, Health monitoring devices।



আইওটিতে কমিউনিকেশন মডেলগুলো ব্যবহার করা হয় যেন মেশিনগুলোর মাঝে ডাটার আদান-প্রদান ঠিক সময়ে,সঠিক ভাবে হতে পারে।  
 **IoT কমিউনিকেশন মডেল**  
কমিউনিকেশন মডেল টাইপসঃ  
**১)Request & Response Model –**এই মডেলে দুইটি জিনিস আছে Client এবং Server। ক্লাইন্টের যখন প্রয়োজন হয় শুধু তখনই সার্ভারের কাছে রিকোয়েস্ট পাঠায়।এই মডেলটি স্টেটলেস।সার্ভার রিকোস্টকে বিভিন্ন ক্যাটাগরীতে ভাগ করে এবং ক্যাটাগরী অনুযায়ী ডাটাবেজ থেকে ডাটা নিয়ে ক্লাইন্টকে রেসপন্স করে।উদাহরণঃ HTTP কমিউনিকেশন প্রোটোকলে এই Request & Response Model ব্যবহার করা হয়।আমরা ব্রাউজারে যত ওয়েবসাইট ভিজিট করি সবগুলোই এই মডেলে আমাদের ব্রাউজারে লোড হয়।

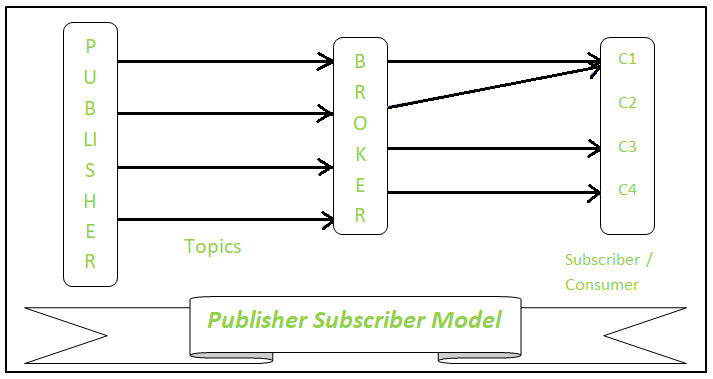
প্রোজেক্ট সেকশনে আমরা ESP32 দিয়ে সার্ভার বানাবো।তখন এই মডেলের প্র্যাকটিক্যাল ইমপ্লিমেন্টেশন দেখতে পাবে।



**২)Publisher-Subscriber Model –**

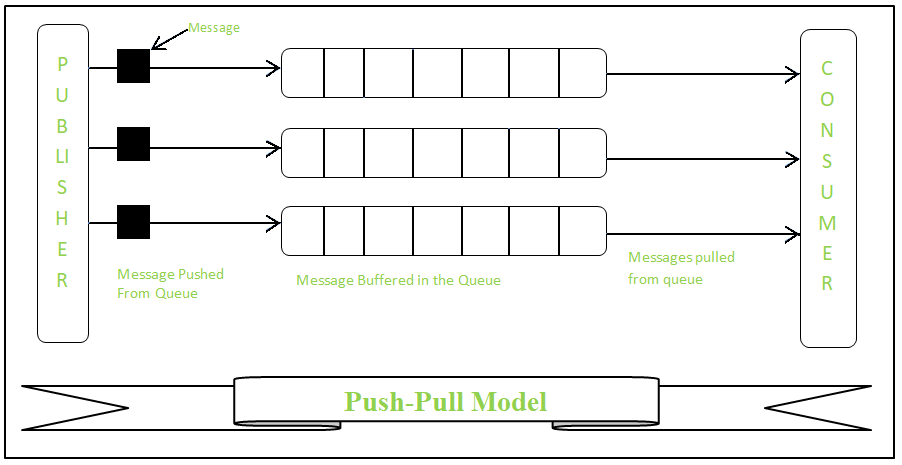
এই মডেলে চারটা বিষয় তোমাকে বুঝতে হবে- Publishers, Brokers, Subscriber এবং Topic

একটি সহজ উদাহরণ দিয়ে বুঝাই, ধরো তুমি ক্রিকেট খেলা খুব পছন্দ করো। একজন ইউটিউবার আছে যে খেলা নিয়ে নিয়মিত ভিডিও পাবলিশ করে।ঐ ইউটিউবার ফুটবল,ক্রিকেট,ভলিবল আরো অনেক খেলা নিয়ে ভিডিও পাবলিশ করে আলাদা আলাদা চ্যানেলে।কিন্তু তোমার শুধু ক্রিকেট নিয়ে আগ্রহ।তাই ক্রিকেট খেলার চ্যানেলে সাবক্রাইব করে রাখলে,যাতে নতুন ভিডিও আসলেই দেখতে পারো ।কিন্তু ব্যস্ততার কারণে কখন ভিডিও আসে খেয়াল রাখতে পারো না।তাই তোমার ছোট ভাইকে বলে রাখলে চ্যানেলে নতুন ভিডিও আসলেই তোমাকে জানায়।  
  
এখানে ইউটিউবার হচ্ছে Publishers,যে বিভিন্ন Topic এ ডাটা পাঠায়।উদাহরণে আলাদা আলাদা খেলাগুলো হচ্ছে এক একটি টপিক।Subscriber তার পছন্দের Topic এ,যে ডাটাগুলো পেতে চায়,সেটায় সাবক্রাইব করে।আর Broker এর কাজ হচ্ছে মধ্যস্থতাকারী (উদাহরণে তোমার ছোট ভাই) যারা কাজ হচ্ছে সব সময় সজাগ থাকা Subscriber যে টপিকে সাবক্রাইব করেছে Publishers সেই টপিকে ডাটা পাবলিশ করলো কি না।যদি ঐ টপিকে ডাটা পাবলিশ না করে অন্য টপিকে করে তাহলে সে Subscriber কে ডাটা পাঠাবে না।কিন্তু যদি সাবস্ক্রাইব করা টপিকে ডাটা পাবলিশ করে তাহলে Broker সঙ্গে সঙ্গে সেই ডাটা Subscriber কে পাঠিয়ে দিবে।উদাহরণঃ MQTT কমিউনিকেশন প্রোটোকল এই Publisher-Subscriber Model ব্যবহার করে।প্রোজেক্ট করার সময় এই মডেলের প্র্যাকটিক্যাল ব্যবহার দেখবো।



**৩)Push-Pull Model –**

এই মডেলে তিনটি বিষয় আছেঃ Data Publishers, Data Consumers, and Data Queues।



**৪)Exclusive Pair –**

এই মডেলে কমিউনিকেশন হয় বাই-ডাইরেকশনাল মোডে অর্থ্যাত সার্ভার এবং ক্লাইন্টের মাঝে ফুল-ডুপ্লেক্স কমিউনিকেশন হয়।সার্ভার এবং ক্লাইন্টের মাঝে সব সময় কানেকশন চালু থাকে ক্লাইন্ট রিকোয়েস্ট পাঠিয়ে সেটা বন্ধ না করে। উদাহরণঃ WebSocket এ এই ধরণের মডেল ব্যবহার করা হয়।



**মাইক্রোকন্ট্রোলার এবং ডেভেলপমেন্ট বোর্ড**

ডেভেলপমেন্ট বোর্ড বিভিন্ন কম্পানি বিভিন্ন নামে তৈরি করে।আবার এই ডেভেলপমেন্ট বোর্ডগুলো ফিচারও আলাদা আলাদা।যেমনঃ দুনিয়ায় সবচেয়ে পরিচিত ডেভেলপমেন্ট বোর্ড হচ্ছে আরডুইনো। এই টা কিন্তু আসলে ঐ কম্পানির নাম।তাদের তৈরি বিভিন্ন মডেলের ডেভেলপমেন্ট বোর্ড আছে পিন সংখ্যা,মেমোরি ইত্যাদি ভেদে যেমনঃ Arduino Uno, Arduino Nano, Arduino Mega।

আবার চাইনিজ কম্পানি Espressif Systems তাদের ডেভেলপমেন্ট বোর্ড ESP সিরিজ নিয়ে আসে।তাদের মূল আকর্ষণই ছিল বোর্ডে ইনবিল্ড ব্লু-টুথ আর ওয়াইফাই তাও আবার আরডুইনোর চাইতে কম দামে। ওদের আরো অনেক মডেল আছে বিভিন্ন ফিচার নিয়ে যা তাদের ওয়েবসাইটে গেলেই দেখতে পাবে।  
  
STMicroelectronics এর STM বোর্ডগুলোও অনেক বিখ্যাত।বর্তমানে চাইনিজ কম্পানি Seeed Studio এর Xiao Series অনেক সারা ফেলছে অনেক ছোট সাইজের মাইক্রোকন্ট্রোলারের জন্য,তারা AI (Artificial Intelligence) এবং হার্ডওয়্যারের ইন্ট্রিগেশন নিয়ে অনেক কাজ করছে।Seeed Studio তারা আমার-তোমার মতো Makers দের সাথেও কাজ করছে তাদের ডিজাইন করা প্রোডাক্ট বাজারে নিয়ে আসতে!  
  
আরেকটি বিখ্যাত কম্পানির নাম তোমরা অবশ্যই শুনে থাকবে- The Raspberry Pi Foundation। তাদের তৈরি Raspberry Pi সরাসরি একটি single-board computers (SBCs) অর্থ্যাত কমপ্লিট কম্পিউটার।পার্থক্য হচ্ছে এটা সাইজে ছোট,দাম কম এবং GPIO (general purpose input/output) pins থাকে যার সাহায্যে ইলেক্ট্রনিক্স কম্পোনেন্ট-সেন্সর,একচুয়েটরের সাথে ইনটারেকশন করা যায়। Raspberry Pi Pico W তে সরাসরি Micropython এ কোড করা যায়,IoT এর কাজগুলো করা যায়।

আমরা এই বইয়েESP32 DEVKIT V1 DOIT বোর্ড নিয়ে কাজ করবো কারণ আমাদের দেশে এই বোর্ডটা সব জায়গায় এভেলেবল এবং অন্য যেকোনো বোর্ড,যাতে Micropython এ কোড করা যায়,সেগুলোর চাইতে দাম কম।তুমি চাইলে Raspberry Pi Pico W বা ESP8266 ও ব্যবহার করতে পারো।সেক্ষেত্রে পিনগুলো ভিন্ন হবে,কোডে কিছু ছোট-খাটো পরিবর্তন করা লাগতে পারে। তবে চিন্তার কারণ নেই,এই সব কিছুই ইন্টারনেটে এভেলেবল,তোমাকে জাস্ট গুগল করতে হবে।  
  
আমাদের এই বইয়ের আসল লক্ষ্য IoT এর থিওরিগুলো বুঝা এবং প্রোজেক্টের মাধ্যমে এদের ইমপ্লিমেন্টেশনগুলো দেখা।কন্সেপ্টগুলো বুঝতে পারলে তুমি ভবিষ্যতে যেকোনো প্রোগ্রামিং ল্যাংগুয়েজ,মাইক্রোকন্ট্রোলার নিয়ে কাজ করতে পারবে।

নোটঃ অবশ্যই সবগুলো কম্পানি এবং তাদের বিভিন্ন মাইক্রোকন্ট্রোলার নিয়ে ঘাটাঘাটি করবে।তাদের মাইক্রোকন্ট্রোলার কোন প্রোডাক্ট/ইন্ডাস্ট্রিতে ব্যবহার হচ্ছে তা খোঁজে দেখবে।বিভিন্ন কম্পানির নিজস্ব কমিউনিটি আছে,বিভিন্ন রিসোর্স আছে শেখার জন্য,ইন্টারন্যাশনাল কম্পিটিশনও আয়োজন করে থাকে নিয়মিত।

**মাইক্রোকন্ট্রোলারের স্ট্রাকচার**

**(Microcontroller’s Structure)**

মাইক্রোকন্ট্রোলারের মেইন কম্পোনেন্টগুলো হচ্ছেঃ

১)Central Processing Unit (CPU):

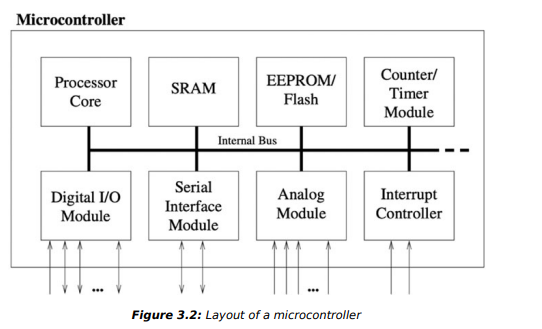
২)Memory:

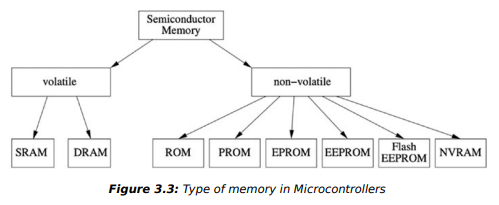
according to their function

* Register File
* Data Memory
* Instruction Memory

৩)Peripherals:

* I/O ports
* Analog I/O
* Interrupts
* Timers
* Communication Interfaces
* Debugging Unit

****

****

**Volatile memory:**

* **Static Random Access Memory (SRAM):**
* **Dynamic Random Access Memory (DRAM):**

**Non-volatile memory:**

**ROM:**

**PROM:**

**EPROM:**

**EEPROM:**

**Flash:**

**NVRAM:**

**কমিউনিকেশন ইনটারফেস কি?**

কমিউনিকেশন ইনটারফেসের মাধ্যমে মাইক্রোকন্ট্রোলার সেন্সর,মডিউল,ডিসপ্লে কিংবা অন্য আরেকটি মাইক্রোকন্ট্রোলারের সাথে যোগাযোগ করতে পারে।মাইক্রোকন্ট্রোলারের GPIO পিনের সাথে আমরা যত সেন্সর,মডিউল লাগাই সবগুলোই কোনো না কোনো কমিউনিকেশন ইনটারফেস ব্যবহার করে মাইক্রোকন্ট্রোলারের সাথে ডাটা আদান-প্রদান করার জন্য।এমন কি আমরা যখন কম্পিউটার থেকে USB তারের মাধ্যমে মাইক্রোকন্ট্রোলারে কোড আপলোড করি কিংবা Serial Monitor এ ডাটা দেখি তখনও কোনো না কোনো ইনটারফেস ব্যবহার করেই তা করি।এইগুলোও এক ধরণের কমিউনিকেশন প্রোটোকল’ই।  
  
ডাটা আদান-প্রদান কি ভাবে হবে এর উপর ভিত্তি করে ইনটারফেস বিভিন্ন প্রকারের হয়।যেমনঃ

Serial এবং parallel interface,

Sync or asynchronous interface,

bus or point-to-point (PTP) interface,

full-duplex interface or half-duplex interface,

master-slave interface or equal partners interface

আমরা তিনটি কমিউনিকেশন ইনটারফেস নিয়ে পড়বো যা Arduino এবং ESP32 তে পাওয়া যায়।এগুলো হচ্ছেঃ

১)UART (Universal asynchronous receiver-transmitter)

২)SPIs (Serial Peripheral Interfaces)

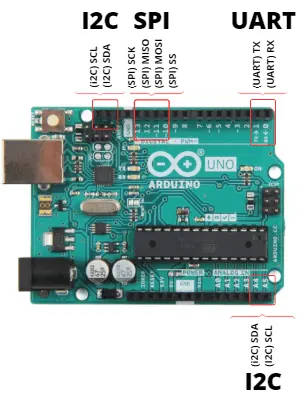
৩)I2C ( Inter-Integrated Circuit)

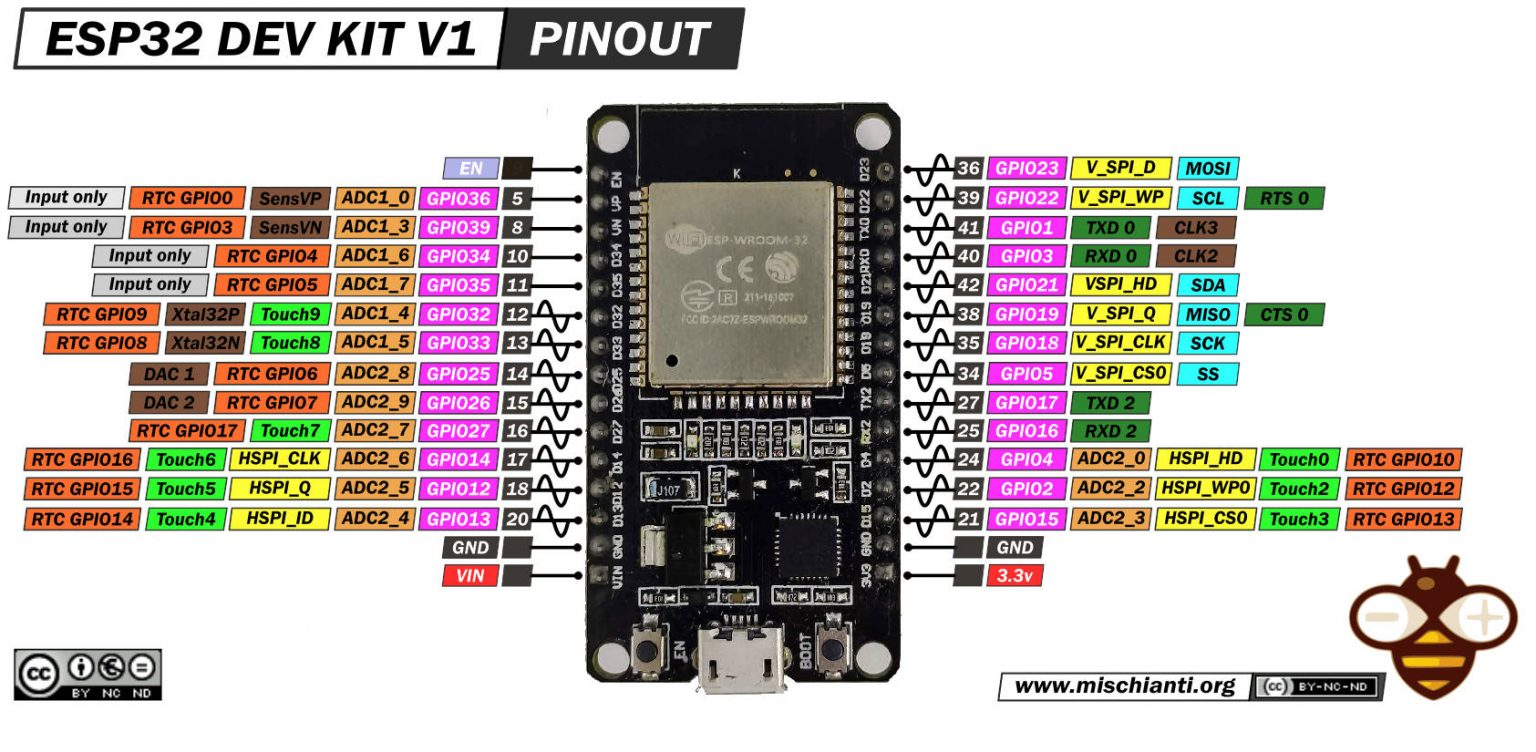
এছাড়াও কিছু গুরুত্বপূর্ণ ইনটারফেজ যা নিয়ে পরে পড়তে পারোঃ USART, CAN

**UART(Universal asynchronous receiver-transmitter)**

UART সরাসরি কমিউনেকশন প্রোটোকল না বরং এটি একটি কমিউনিকেশন মডিউল যার সাহায্যে asynchronous serial communication করা যায়। এই ইনটারফেসে দুইটি ইলিমেন্ট আছেঃ ১)UART Receiver এবং ২)UART Transmitter।

ট্রান্সমিটার এবং রিসিভার একটি নির্দিষ্ট স্পিডে ডাটা ট্রান্সফার করে।এই ডাটা ট্রান্সফার রেটকে বলে Baud Rate।ডাটা পাঠানো এবং রিসিভ করার জন্য ট্রান্সমিটার এবং রিসিভার উভয়কেই একই Baud Rate এ থাকতে হবে। সাধারণত Arduino এবং ESP32 তে 9600, 115200 Baud Rate ব্যবহার করা হয়।

UART মডিউলে দুইটি লাইনের মাধ্যমে ডাটা আদান-প্রদান হয়ঃ Transmit (TXD) এবং Receive (RXD) লাইন।



**SPIs (Serial Peripheral Interfaces)**

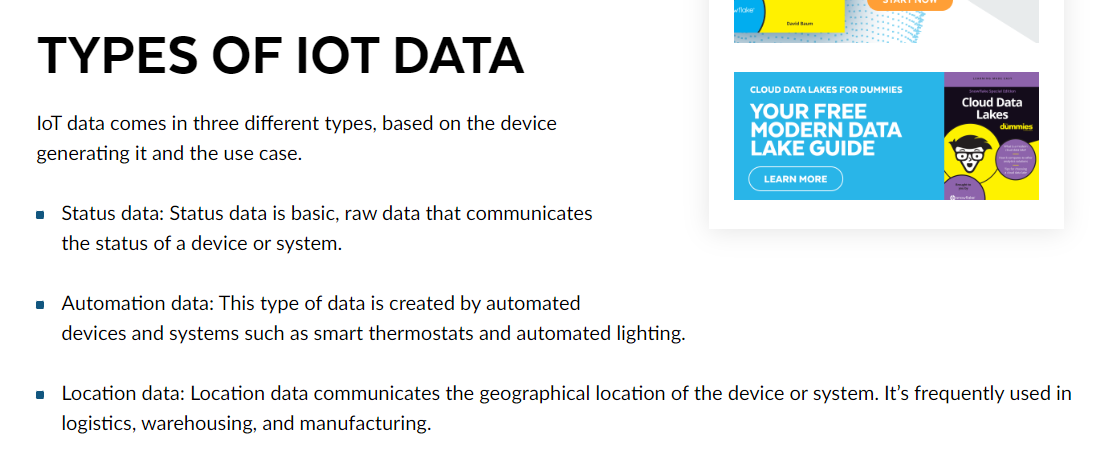
এটাও একধরণের synchronous serial communications protocols।এতে দুই ধরণের ডিভাইস থাকেঃ ১) Master এবং ২)Slaves ।মাস্টার হচ্ছে মেইন ডিভাইস(মাইক্রোকন্ট্রোলার) আর স্লেইভস হচ্ছে পেরিফেরাল ডিভাইস(সেন্সর,ডিসপ্লে,মডিউল)। একই মাস্টারের আন্ডারে একাধিক স্লেইভ ডিভাইস থাকতে পারে। SPI তে চারটি লাইন থাকেঃ

* MOSI (Master Out, Slave In) - এই লাইনের মাধ্যমে মাস্টার ডাটা ট্রান্সমিট করে স্লেইকে পাঠায়
* MISO (Master In, Slave Out) - এই লাইনের মাধ্যমে স্লেইভ ডাটা ট্রান্সমিট করে মাস্টারকে পাঠায়
* SCK (System Clock) - এই লাইনের মাধ্যমে মাস্টার ক্লক সিগন্যাল পাঠায়
* SS (Slave Select) - এই লাইনের মাধ্যমে মাস্টার স্লেইভ সিলেক্ট করে

**I2C ( Inter-Integrated Circuit)**

এটাও synchronous serial communication protocols।এই প্রোটোকল ব্যবহার করা হয় Integrated Circuits (ICs) এবং Microcontrollers (MCUs) এর ভেতর ডাটা আদান-প্রদান করতে।এখানে দুই ধরণের লাইন ব্যবহার করা হয় ডাটা আদান-প্রদানের জন্যঃ Serial Clock Pin (SCL) এবং Serial Data Pin (SDA)

বাজারে I2C Breakout Modules কিনতে পাওয়া যায় যা সরাসরি ডিভাইস যেমন LCD ডিসপ্লে এর সাথে যুক্ত করে ব্যবহার করা হয়।



**ইন্টারনেট কীভাবে কাজ করে?**

**ইন্টারনেট কী?**  
“The Internet is a global network of billions of computers and other electronic devices” অর্থ্যাৎ ইন্টারনেট হচ্ছে পুরো পৃথিবীজুড়ে ছড়িয়ে থাকা কম্পিউটার এবং ইলেক্ট্রোনিক ডিভাইসগুলো পরস্পরের সাথে সংযুক্ত নেটওয়ার্ক সমষ্টি। সহজ করে বললে পৃথিবীর সব কম্পিউটার/ইলেক্ট্রিক ডিভাইসকে তুমি যদি একে অপরের সাথে কানেক্ট করে দিতে পারো,সেটা হতে পারে তারের মাধ্যমে বা তারবিহীন(wireless), তাহলে এই গ্লোবাল স্কেলের নেটওয়ার্ক কানেকশনকে আমরা ইন্টারনেট বলি।

এই কানেকশন তৈরি করার মূল উদ্যেশ্য একটাই- ডাটা ট্রান্সমিট করা, একে-অপরের সাথে ডাটা আদান-প্রদান করা।  
এই ডাটা বিভিন্ন ফরমেটের হতে পারে - টেক্সট(.txt), ইমেজ( PNG,JPEG), অডিও(MP3,WAV), ভিডিও(MP4), JSON ইত্যাদি।

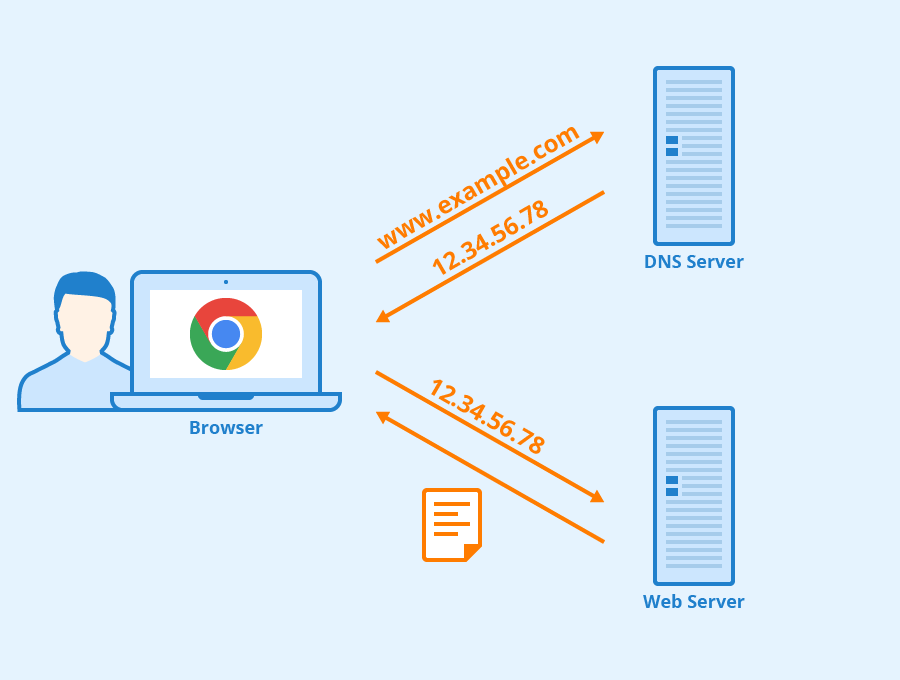
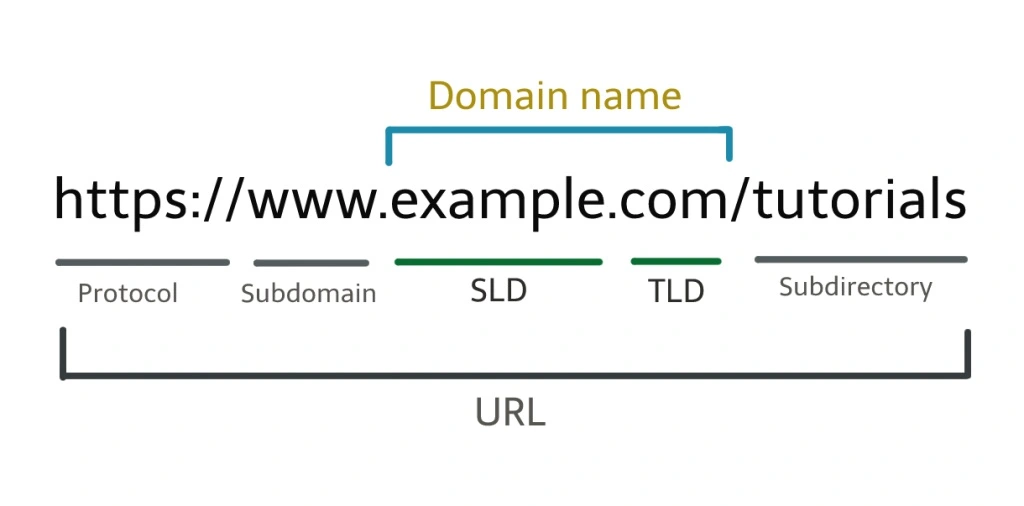
এখন তোমাড় মনে প্রশ্ন আসতে পারে,”আমি যদি আমার কম্পিউটার থেকে আমার বন্ধুর কম্পিউটার/ডিভাইসে ডাটা পাঠাতে চাই তাহলে পৃথিবীর কোটি কোটি কম্পিউটারের মাঝে কিভাবে ঠিক আমার ঐ বন্ধুর কম্পিউটারেই ডাটা’টা যাবে?”।  
এইখানেই আসে আইপি এড্রেস।  
  
**IP Address**

An IP address is a unique address that identifies a device on the internet or a local network. আইপি এড্রেস হবে প্রতিটি কম্পিউটার/মেশিনের জন্য একদম ইউনিক। পৃথিবীর প্রতিটি ডিভাইসের আলাদা আইপি এড্রেস আছে,তুমি যে ডিভাইসটি ব্যবহার করছো তারও আইপি এড্রেস আছে।এই আইপি এড্রেসের মাধ্যমে ইন্টারনেট বুঝতে পারে কোন ডিভাইস থেকে কোন ডিভাইসে ডাটা পাঠাতে হবে।  
  
আমরা যত ওয়েবসাইট ব্যবহার করি ঐ সবগুলোই কিন্তু কোনো না কোনো সার্ভার কম্পিউটারে রাখা হয়েছে।আমরা যদি ব্রাইজারে এই আইপি এড্রেস টাইপ করে ব্রাইজ করি তাহলেই কিন্তু ঐ ওয়েবসাইটে চলে যাওয়া যাবে।



এখন তুমি প্রশ্ন করতে পারো,” কিন্তু ভাই,আমি তো কখনই এমন ভাবে আইপি এড্রেস লিখে সার্চ করি নাই,তাছাড়া দুনিয়ার এতো ওয়েবসাইট,ডিভাইসের আইপি এড্রেস মনে রাখাও তো অসম্ভব,তাহলে কী ভাবে কাজ করে?”  
এর উত্তরে ডোমেন নেইম চলে আসে।  
  
**Domain Name**

তোমার ফোনে কিন্তু অসংখ্য মানুষের ফোন নাম্বার আছে।তুমি কিন্তু জানো সবার ফোন নাম্বার আলাদা করে মুখস্ত রাখা সম্ভব না।তাই আমরা কী করি,ফোনবুক নামে একটা সফটওয়্যার আছে সেখানে আমাদের পরিচিতদের ফোন নাম্বার তাদের নাম দিয়ে সেইভ করে রাখি।যখন আমরা তাদের কল করি তখন ঐ নামটা খুঁজে বের করি।সামনে নাম থাকলেও ভেতরে কিন্তু আসলে নাম্বারটাই আছে এবং ঐ নাম্বারের কল’টা যায়।  
  
একই রকমভাবে ডোমেইন নেইমও কাজ করে। যেমনঃ www.youtube.com এই ডোমেইন নেইম দিয়ে আমরা যখন সার্চ করি তখন এইটি DNS server এর মাধ্যমে ঐ আইপি এড্রেসটাকে খুঁজে নিয়ে আসে।DNS server এর কাজ হচ্ছে এই ডোমেইন নেইমগুলোকে মেনেজ করা। The Domain Name System (DNS) is the phonebook of the Internet.



Chapter -1 শুরুর প্রস্তুতি

Flashing Micropython Firmware to ESP32

অন্যান্য মাইক্রোকন্ট্রলার বোর্ডের মতো ESP32 তে ডিফল্ড ভাবে মাইক্রোপাইথন Flash/Upload করা থাকে না।তাই ESP32 তে MicroPython এ কোড করার জন্য সবার আগে Micropython Firmware টা Flash বা সহজ ভাষায় Upload করে নিতে হবে।

নোটঃ Micropython Flash করার মানে এই নয় যে আমরা MicroPython ছাড়া আর ESP32 তে কোড করতে পারবো না।এটা পুরাটা ড়িভার্সেবল।Arduino IDE থেকে আবারও C++ কোড আপলোড করলেই এই মাইক্রোপাইথন ফার্মওয়্যার চলে যাবে,C++ এ কোড করা যাবে আগের মতো।পরবর্তীতে আবারও MicroPython ব্যবহার করতে চাইলে আবারো Flash করে নিতে হবে।

Chapter-2 Python/Micropython Programming Basics

আমাদের এই বইয়ের মূল বিষয় IoT শেখা।পাইথন বা মাইক্রোপাইথন নিয়ে পুরো আলাদা একটি বই লিখে ফেলা সম্ভব।আমি ধরে নিচ্ছি তোমাদের মাইক্রোকন্ট্রোলারে প্রোগ্রামিং এর পূর্ব অভিজ্ঞতা আছে C++ এ।আমি শুধু এই বইতে C++ এর সাথে পাইথনের বেসিক যে কন্সেপ্টগুলোর পার্থক্য সেটি তুলে ধরছি সহজে প্রোগ্রামিং করার জন্য।

তুমি যদি কোনো দিন প্রোগ্রামিং নাও করে থাকো তাতেও কোনো সমস্যা নেই।প্রোগ্রামিং এর যে কন্সেপ্টগুলো জানলে যেকোন কাজ করে ফেলা যাবে তার সবগুলোই এখানে দিচ্ছি।তোমার কাজ হবে যেকোনো প্রোগ্রামিং চ্যানেল দেখে চট করে বেসিকগুলো হাতে-কলমে প্র্যাকটিস করে ফেলা,এক সপ্তাহ সময়ই যথেষ্ট এর জন্য।আর এখানে নোট হিসেবে লিখাগুলো তো থাকবেই,ভুলে গেলে চট করে দেখে ঝালাই করে নিবে।

**মাইক্রোপাইথন প্রোগ্রামিং**

মাইক্রোপাইথন এবং পাইথনের বেসিক সিন্ট্যাক্স একই।তাই তুমি যখন মাইক্রোপাইথন শিখছো ঠিক একই বেসিক তুমি পাইথনে কাজে লাগাতে পারো। ভালো হয় তুমি যদি পাইথন শেখার পেছনে আগে এক সপ্তাহ সমইয় দাও এবং কিছু প্রবলেম সলভিং করে পাইথনের উপর কিছুটা দক্ষ হয়ে নিতে পারো।

এই বইয়ে আমরা পাইথনের কন্সেপ্টগুলো দিবো,তোমাকে এইগুলো নিজ হাতে কোড করে ক্লিয়ার হয়ে নিতে হবে।চাইলে অনলাইন কোনো কোর্স ফলো করতে পারো।এই বইটির হার্ড কপি তোমার কাছে রেফারেন্স পয়েন্ট হিসেবে কাজ করবে।কোনো কিছু ভুলে গেলে বই থেকে ঝালিয়ে নিতে পারবে।

**Mathematical Operators**

পাইথনে সাধারণ হিসাব-নিকাশ করার জন্য যে অপারেটরগুলো ব্যবহৃত হয়। REPL এ বিভিন্ন ম্যাথ অপারেশন ট্রাই করে দেখো।

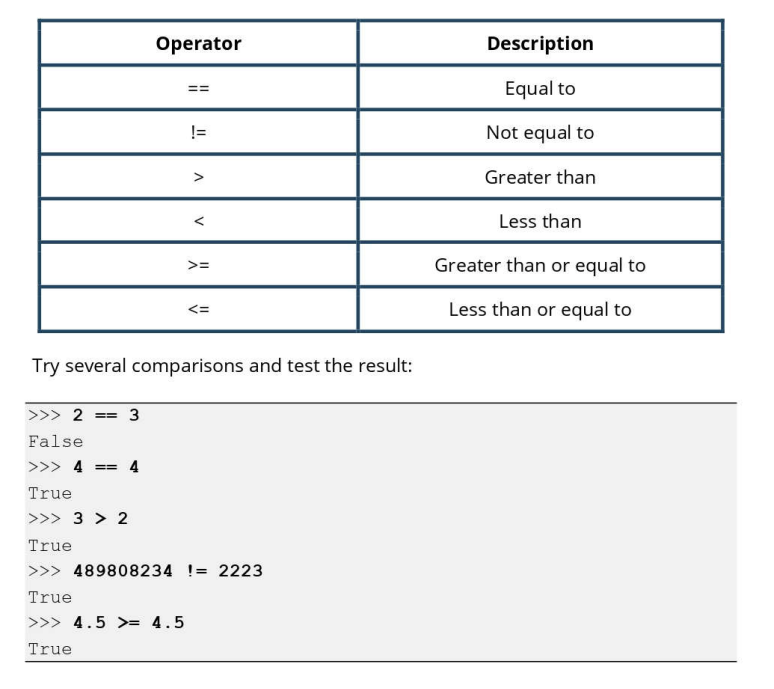
|  |  |
| --- | --- |
| অপারেটর | অপারেশন |
| + | যোগ |
| - | বিয়োগ |
| \* | গুণ |
| / | ভাগ |
| // |  |
| % | ভাগফল |



তুমি যদি আরো জটিল ম্যাথ অপারেশন যেমনঃ square root, logarithm, trigonometric ইত্যাদি অপারেশন করতে চাও তবে math মডিউল ব্যবহার করতে হবে।এই নিয়ে আমরা পরবর্তীতে শিখবো।

**Relational Operators**

এই অপারেটরগুলো দুইটি ভ্যালুর মাঝে তুলনা করতে ব্যবহার করা হয়।



**Print( ) ফাংশন**

আমরা এতক্ষণ কিন্তু কেবল REPL টার্মিনালে কিছু লিখেছে। সত্যতিকারে কোনো প্রোগ্রাম ফাইল লিখি নি।এবার একটি নতুন পাইথন ফাইল নাও, নাম দাও hello.py

এতে জাস্ট এই লিখাটা লিখো, কি লিখেছো তা নিয়ে চিন্তা করার দরকার নেই

এবার রান দাও।দেখ তোমার টার্মিনালে hello world দেখাচ্ছে। print( ) ফাংশনের সাহায্যে আমরা টার্মিনালে কোনো কিছু দেখাই।টার্মিনালের যা দেখাতে চাই তা প্যারেনথিসিসের ( ) – এর ভেতর লিখা হয়। প্রোগ্রামিং এ প্রথম বন্ধনীকে ( )প্যারেনথিসিস বলে।

যদি কোনো বাক্য হয় (string data) তবে অবশ্যই “ “ দিতে হবে। নাম্বারের বেলায় দরকার নেই।

একটা কন্সেপ্ট এখনই ক্লিয়ার করে নাও, যদি আমরা টার্মিনালে আউটপুট দেখাতে চাই তাহলেই কেবল print( ) ফাংশন ব্যবহার করি। যদি টার্মিনালের বদলে আমরা কোনো স্ক্রিনে(যেমনঃ LCD, OLED Display) তে আউটপুট দেখাতে চাই তবে তার জন্য যে ফাংশন সেটি ব্যবহার করবো,এই নিয়ে আমরা সামনে কাজ করবো।

**পাইথন সিনট্যাক্স**

সব ভাষারই কিন্তু একটা নিয়ম আছে। এই ধরো বাংলা ভাষার কথা, বাংলা ভাষায় কথা বলেত গেলে আমাদের কিছু নিয়ম মেনে কথা বলেত হয়,ঠিক তেমনি ইংরেজি ও অন্যান্য ভাষার জন্যও এটা সত্যি।ঠিক সেরকমই প্রোগ্রামিং ভাষারও নিয়ম আছে। এই নিয়মগুলোকেই আমরা সিনট্যাক্স বলি। সাইনট্যাক্স অনুযায়ী প্রোগ্রাম না লিখলে কোড কাজ করবে না, ইরর দিবে।

ভেরিয়েবল(Variables)

সহজ ভাষায় ভারিয়েবল হচ্ছে একটি খালি বক্স বা স্টোরেজ প্লেজহোল্ডার যাতে কোনো ডাটা (নাম্বার,টেক্সট) স্টোর করা হয়।যেমন নিচে 10 এই সংখ্যাটিকে আমরা X নামের একটি ভেরিয়েবলে স্টোর করলাম। আবার Y নামের আরেকটি ভেরিয়েবলে IoT টেক্সটিকে স্টোর করলাম।  
X = 10  
Y = “IoT”

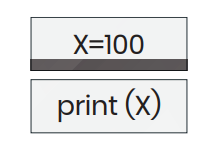
ভেরিয়েবলের মান(Value) সেট করতে assign বা সমান চিহ্ন(=) ব্যবহার করা হয়।

পাইথনে ভেরিয়েবল নেয়ার জন্য এটি কি টাইপের ভেরিয়েবল তা ডিক্লেয়ার করে নেয়ার প্রয়োজন পরে না।

**ভ্যারিয়েবল(Variable)**

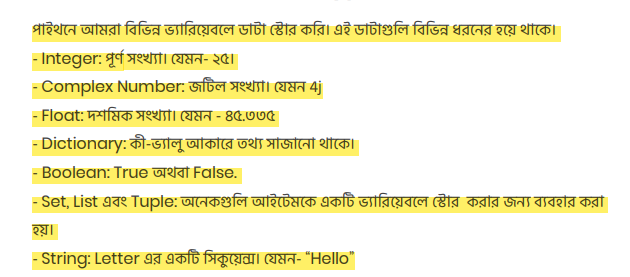
ভ্যারিয়েবল একটি খালি বক্সের মতো চিন্তা করতে পারো।এটি একটি নাম যা মেমোরির একটি লোকেশনকে নির্দেশ করে।আমরা ভেরিয়েবলের ভেতর ডাটা স্টোর করে রাখতে পারি।যেমনঃ নিচের ছবিতে X নামের ভ্যারিয়েবলের ভেতর 100 ডাটা আমরা স্টোর করেছি।

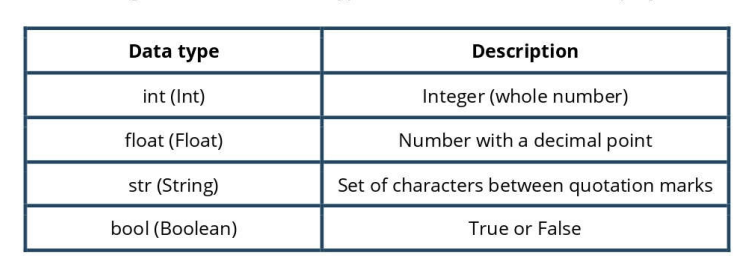
ভ্যারিয়েবলের ভেতর কোনো ভ্যালু স্টোর করতে এসাইন অপারেটর( = ) ব্যবহার করা হয়। ভ্যারিয়েবল কেস সেনসেটিভ অর্থ্যাত name আর Name কিন্তু এক নয়।



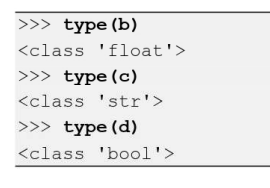
**ডাটা টাইপস(Data Types)**

পাইথন ভ্যারিয়েবলের ভেতর বিভিন্ন টাইপের ডাটা স্টোর করতে পারি

- 



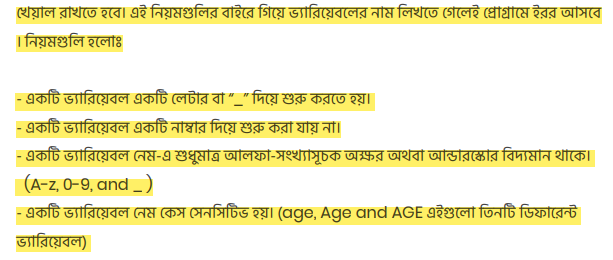
ভ্যারিয়েবল টাইপ দেখতে type( ) ফাংশন ব্যবহার করা হয়।

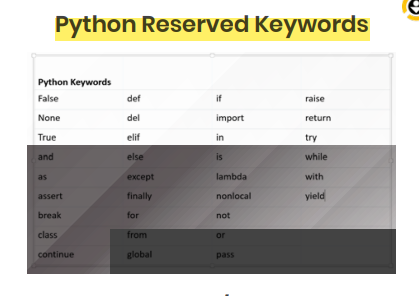


এবার একটা কাজ করো একটা ভ্যারিয়েবল ডিক্লেয়ার করে এর টাইপ বের করার প্রোগ্রাম লিখো নিচের ভাবে,দেখো আউটপুটে কি ডাটা টাইপ দেখায়।

কি? আউটপুট দেখা যায় না, তাই না? এমনটা কেনো হলো? পাইথন কি তাহলে আউটপুট দেয় নি? উত্তর হচ্ছে দিয়েছে কিন্তু তুমি তো আউটপুট টার্মিনালে দেখাতে বলো নাই,তাই সে টার্মিনালে কিছুই দেখাচ্ছে না।

এবার এটা ট্রাই করোঃ





**Comments**

অনেক সময় আমরা যে কোড লিখি তার পাশে কিছু নোট লিখে রাখার প্রয়োজন পরে, যাতে করে পরবর্তীতে আমরা বুঝতে পারি এই কোডের কাজ কি এবং অন্য কেউ আমাদের কোড পড়ে বুঝতে পারে। এই নোটগুলো পাইথন ইনটারপ্রেট করে না অর্থ্যাৎ আমরা যাই লিখি না কেন মেইন প্রোগ্রামে এগুলোর কোনো ফাংশনালিটি নেই।এদেরকে কমেন্ট বলে।



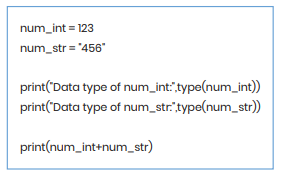
**Receiving Input**

প্রোগ্রামকে আরো ডায়নামিক করার জন্য আমরা ইউজার থেকে ইনপুট নিতে পারি।এটির জন্য input( ) ফাংশন ব্যবহার করা হয়।



**Data Type Conversion/Casting**

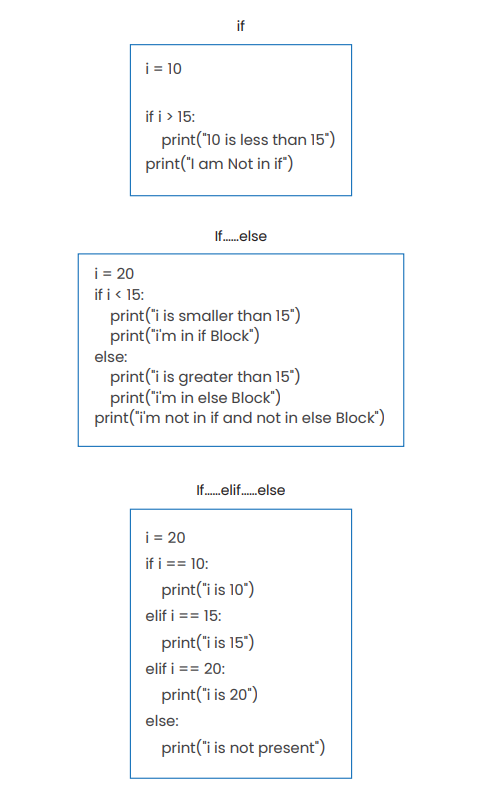
ডাটাটাইপ কনভাসর্ন বলতে এক টাইপের ডাটাকে অন্য টাইপে কনভার্ট করাকে বুঝায়। এর জন্য কিছু বিল্ট-ইন ফাংশন আছে। int(), float(), str() ব্যবহার করে টাইপ কাস্টিং করা হয়।

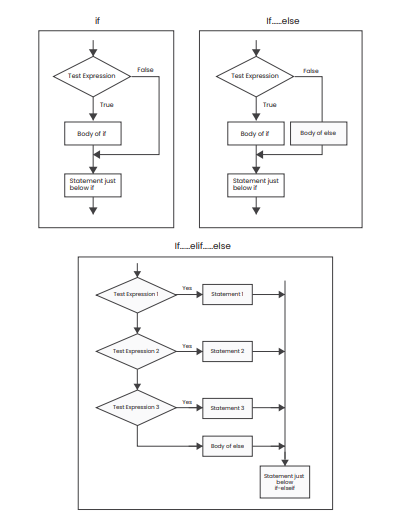


**Conditional Statements**

যখন কোনো ডিসিশন মেকিং এর প্রয়োজন হয় তখন কন্ডিশনাল স্ট্যাটমেন্ট ব্যবহার করা হয়। এর মাধ্যমে ঠিক হয় কোনো কন্ডিশন যদি True হয় তবে এর আন্ডারে যে কোড ব্লক থাকবে সেটি এক্সেকিউট হবে, আর যদি False হয় তবে এক্সেকিউট হবে না।

পাইথনে if, if-else, if-elif-else এভাবে তিনটি স্ট্রাকচারে শর্তগুলো সাজানো যায়,নিচের ফ্লো-চার্টগুলো দেখোঃ





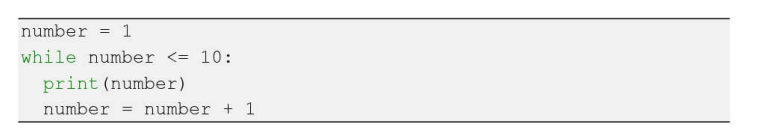
**Indentation**

Arduino IDE তে প্রোগ্রাম লিখার সময় কোড ব্লক বুঝাতে { } ব্যবহার করা হয়, সিনট্যাক্স এর শেষ বুঝাতে সেমিকোলন ( ; ) দেয়া হয়। পাইথনে এই ধরণের কোনো কিছু নেই। বরং কোনো কোড ব্লক বুঝাতে Indentation ব্যবহার করা হয়( চারটা স্পেস) আর প্রতি এক্সপ্রেশন এর শেষে কোলন : দিতে হয়।

****

**নোটঃ পাইথনে স্ট্যান্ডার্ড ইন্ডেন্টেশন এ চারটা স্পেস দেয়া হয়।কিন্তু মাইক্রোপাইথনে শুধুমাত্র দুইটি স্পেস দেয়া হয়। ইনডেনটেশন দিতে কীবোর্ডের Tab কী চাপো।**

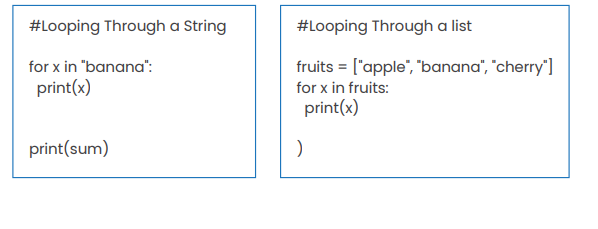
**Loop**

যখন একই ধরণের কাজ বার বার করার প্রয়োজন হয় তখন লুপ ব্যবহার করা ****

লুপ দুই ধরণেরঃ while loop আর for loop

While loop এ একটি কন্ডিশন দেয়া হয়, যতক্ষণ এই কন্ডিশন সত্য হবে ততক্ষণ এই লুপের মধ্যে থাকা কোড ব্লক চলতে থাকবে,যখনই কন্ডিশন মিথ্যা তখনই আর ঐ কোড ব্লক এক্সেকিউট হবে না।

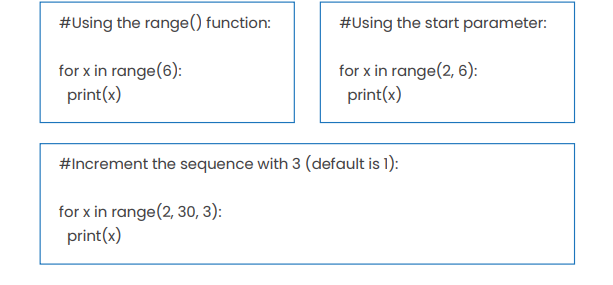
For loop ব্যবহার করা হয় কোনো সিকুয়েন্স যেমন লিস্ট, ডিকশনারি, স্ট্রিং ইত্যাদির উপর iterate করার জন্য।ফর লুপ এই সিকুয়েন্সের প্রত্যেকটা এলিমেন্টকে একবার করে ভিজিট করবে।



**For in range() function**

ফর লুপ যদি বলে দিতে চাই এক্সাক্টলি কতবার রান হবে বা কোনো রেঞ্জ এর ভেতর রান হবে তবে in range ব্যবহার করতে পারি।

যখন আমরা কোনো কোড ব্লককে নির্দিষ্ট সংখ্যক বার রিপিট করতে চাই তখন for loop ব্যবহার করা হয়।আর যখন কোনো কোড ব্লককে একটি নির্দিষ্ট কন্ডিশন মিথ্যা না হওয়া পর্যন্ত রিপিট করতে চাই তখন while loop ব্যবহার করা হয়।



**Function**

ফাংশন হচ্ছে একটি কোড ব্লক যা নির্দিষ্ট কোনো কাজ করে।যখন ঐ নির্দিষ্ট কাজটি করার প্রয়োজন হয় তখন ঐ ফাংশনকে কল করা হয় এবং কেবল তখনই ফাংশনের ফেতর থাকা কোড ব্লকটি রান হবে।

পাইথনের কিছু বিল্ট-ইন ফাংশন যেমনঃ print( ), input( ), max( ) আমরা অলরেডি ব্যবহার করেছি।

আমরা চাইলে নিজেরাও আমাদের প্রয়োজন মতো ফাংশন তৈরি করে নিতে পারি।আমাদের তৈরি ফাংশনগুলোকে বলা হবে user defined function।

ফাংশন লিখার জন্য def কীওয়ার্ডটি ব্যবহার করা হয়।

**Parameter/Argument**

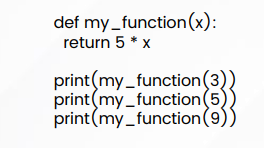
ফাংশন কল করার সময় আমরা এর ভেতর ডাটা পাস করে দিতে পারি।এই ডাটাগুলোকে বলা হয় প্যারামিটার বা আর্গুমেন্ট।আমরা চাইলে একের অধিক প্যারামিটারও দিতে পারি।

**Return Statement**

আমরা যদি চাই ফাংশনটিকে কল করার পর তারা আমাদের কিছু রিটার্ন দিবে সেক্ষেত্রে return statement ব্যবহার করা হয়।

যেমন ধরো max() ফাাংশনের কথা। এই ফাাংশনে যখন আমরা দুইটি সংখ্যা পাস করি, তখন এটি আমাদের এই দুইটি সংখ্যার মধ্যে যেটি বড় সেটি ফলাফল হিসেবে রিটার্ন করে। এই যে ফলাফল রিটার্ন করছে এই ঘটনাটাকে আমরা বলি ফাংশন আমাদের ভ্যালু রিটার্ন করছে।

যখন কোনো ফাংশনের ভ্যালু রিটার্ন করতে চাইবো তখন ঐ ফাংশনের ভেতর return লিখে ঐ ভ্যালুটি লিখে দিবো।



**Class এবং Object**

পাইথন একটি অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং ল্যাংগুয়েজ (Object-Oriented Programming – OOP)। সিম্পল ভাবে বললে এটা আলাদা কিছু না,জাস্ট প্রোগ্রাম লিখার একটি বিশেষ ধরণ। OOP বুঝতে গেলে দুইটা জিনিস সম্পর্কে জানতে হবেঃ class এবং object

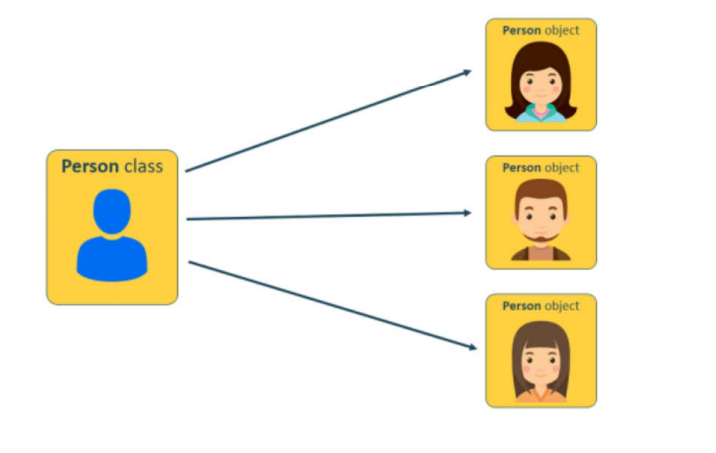
Class হলো প্রোগ্রামিং এর ব্লুপ্রিন্ট অবজেক্ট তৈরি করার জন্য।

ধরো আমরা কোনো এক ভার্সিটির স্টুডেন্টদের ডাটা তৈরি করবো যেখানে তাদের ডিটেইলস থাকবে। এখন একজন ভার্সিটি স্টুডেন্টদের ডিটেলস এ কি কি থাকতে পারে?

* স্টুডেন্টের নাম
* ডিপার্টমেন্ট
* ব্যাচ
* রোল
* হলের নাম ইত্যাদি

এখন তুমি যদি প্রত্যেক স্টুডেন্ট এর জন্য একটি করে ভেরিয়েবল নিয়ে কাজ করতে যাও তাহলে কিন্তু তোমার জন্য বেশ ঝামেলার কাজ হয়ে যাবে।যদি এমন হতো তুমি সব স্টুডেন্টদের জন্য একটা কমন ফর্ম তৈরি করলে আর যখন কোনো স্টুডেন্টের ডিটেলস লিখতে হবে তখন শুধু ঐ ফর্মেটের নামের পাশে ডিটেলস লিখে দিলে তাহলে তোমার জন্য কাজটা অনেক সহজ হয়ে যেত।

ঠিক এই ফর্মেট তৈরির কাজটাই করে class। আরেই class কে ব্যবহার করে যখন এক একটি স্টুডেন্টের ডিটেইলস লিখা হচ্ছে অখন ঐ স্টুডেন্টদের বলা হয় object.



এখন এই যে class এর মাঝে বিভিন্ন ভ্যারিয়েবল থাকতে পারে(যেমনঃ নাম,রোল,ব্যাচ…) আবার ফাংশনও থাকতে পারে(যেমন ধরো একটি ফাংশন আছে যাকে কল করলে স্টুডেন্ট ডেস্ক্রিপশন লিখে দেয় এক লাইনে)।

Class এর ভেতর থাকা ভ্যারিয়েবল গুলোকে আলাদা ভাবে ডাকা হয় অ্যাট্রিবিউটস (attributes) নামে আর ফাংশনগুলোকে ডাকা হয় মেথড (method) নামে।

**Modules**

মনে করো তোমার ভার্সিটির সকল স্টুডেন্ট, টিচার, কর্মচারীদের নিয়ে ডিটেলস লিখার জন্য আলাদা-আলাদা ক্লাস বানালে।সবার ডিটেইলস কিন্তু এক হবে না।তুমি করলে কি সবগুলো ক্লাস একটা পাইথন ফাইলে নিয়ে varsity\_person.py নামে সেইভ করে রাখলে।

এখন যখনই কোনো পারসোনের ডিটেইলস লিখার প্রয়োজন হয় তুমি জাস্ট ঐ varsity\_person.py ফাইলটা তোমার কোডে ইম্পোর্ট করে ফেল।

এই যে পাইথন ফাইল যাতে অনেকগুলো ক্লাস,ফাংশন,অ্যাট্রিবিউটস আগে থেকে লিখা থাকে যাতে পরবর্তীতে অন্য কোডে এইগুলো ব্যবহার করা যায় তাকেই মডিউলস বলে।একে লাইব্রেরি(library) নামেও ডাকা হয়।

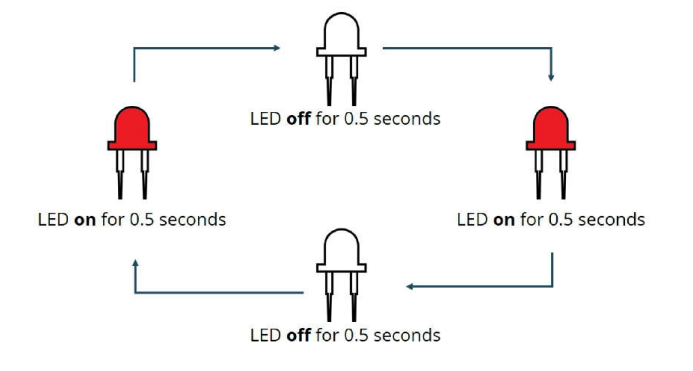
পাইথনে এমন অনেক মডিউল আছে ভিন্ন ভিন্ন কাজ করার জন্য।মাইক্রোপাইথনে যেহেতু আমরা মাইক্রোকন্ট্রোলারের লিমিটেড মেমোরি রিসোর্স নিয়ে কাজ করি তাই পাইথনের ঐ স্ট্যান্ডার্ড লাইব্রেরিগুলোর ছোট সাবসেট নিয়ে তৈরি লাইব্রেরি নিয়ে কাজ করতে হয়।

আবার মাইক্রোপাইথনে I/O পিন (Input/Output), নেটওয়ার্ক ইত্যাদি ইলেক্ট্রোনিক্স রিলেটেড আলাদা লাইব্রেরি আছে যা পাইথনে নেই।

থিওরি পড়ার পালা এখানেই শেষ! এখন আমরা শুরু করবো প্রোজেক্ট,কীভাবে প্রোগ্রামিং এর মাধ্যমে আমাদের চারপাশের পরিবেশের সাথে ইনটারেক্ট করা যায়। প্রোজেক্ট করতে গিয়ে আমরা ইলেক্ট্রনিক্স কম্পোনেন্ট সম্প্ররকে জানবো, শিখবো এইগুলো কী ভাবে কাজ করে।

**Blinking LED**

এই প্রোজেক্টে আমরা একটি এলইডি (LED) এর ব্লিংকিং দেখবো অর্থ্যাৎ এলইডি লাইটকে মাইক্রোকন্ট্রোলারের সাহায্যে কয়েক সেকেন্ড পর পর অন করবো আবার অফ করবো।



মাইক্রোকট্রোলারের সাহায্যে আমাদের চারপাশের পরিবেশের সাথে ইন্টারেক্ট করার জন্য মাইক্রোপাইথনে machine মডিউল আছে।এই মডিউলের অনেকগুলো ক্লাস আছে,তার মাঝে একটি হচ্ছে Pin যা দিয়ে মাইক্রোকন্ট্রোলারের ইনপুট/আউটপুট পিন (I/O pin) এর সাথে ইন্টারেক্ট করা যায়।

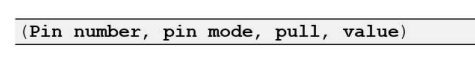
আমরা যেহেতু মাইক্রোকন্ট্রোলারের খুব লিমিটেড রিসোর্স নিয়ে কাজ করছি তাই সব machine মডিউলের সব ক্লাস ইম্পোর্ট করবো না।আমাদের শুধু Pin ক্লাস লাগবে সেটাই ইম্পোর্ট করবো।

time মডিউলের কাজ কি আশা করি নাম দেখেই বুঝতে পারছো।এর সাহায্যে-

* সময়, তারিখ জানা যায়
* কিছু সময়ের জন্য বিরতি (delay) দেয়া যায়
* প্রোগ্রাম কতক্ষণ ধরে চলছে জানা যায় ইত্যাদি

time মডিউলের sleep( ) মেথডের সাহায্যে কয়েক সেকেন্ডের জন্য বিরতি দেয়া যায়।কত সেকেন্ডের জন্য বিরতি দিতে চাও তা প্যারান্থিসিসের ভেতর লিখে দিলেই হচ্ছে।

এবার Pin class এর একটি অব্জেক্ট তৈরি করলাম led এবং Pin ক্লাস অব্জেক্ট তৈরির জন্য চারটা এট্রিবিউট নিতে পারে।



Pin Number: GPIO পিন, মাইক্রোকন্ট্রোলারের কত নাম্বার পিনের সাথে কম্পোনেন্টটা যুক্ত (এক্ষেত্রে LED)

Pin Mode: পিন ইনপুট কিংবা আউটপুট মুডে থাকতে পারে।

ইনপুট মুডে থাকার অর্থ ঐ পিন দিয়ে যুক্ত কম্পোনেন্ট থেকে সিগন্যাল মাইক্রোকন্ট্রোলারে আসবে,মাইক্রোকন্ট্রোলার সেটা রিড করবে।যেমনঃ সেন্সরের সিগন্যাল পিন, পুশ বাটন

Pin.IN

আউটপুট মুডে থাকার অর্থ যুক্ত কম্পোনেন্টটি আউটপুট ডিভাইস(যেমনঃ LED, actuator, motor)।মাইক্রোকন্ট্রোলার থেকে ঐ কম্পোনেন্টের পিনে সিগন্যাল যাবে,সেই অনুযায়ী ঐ কম্পোনেন্টটি কাজ করবে।

Pin.OUT

-এখন তুমিই বলো আমাদের এই প্রোজেক্টে পিন মুড কী হবে?

-হ্যাঁ,আউটপুট মুডে।

বাকি দুইটি প্যারামিটার Pull এবং value এর কোনো কাজ নেই এই প্রোজেক্ট,কোনো প্যারামিটার পাস না করা মানে ঐগুলো ডিফল্ট ভ্যালুতেই থাকবে।

এখন একটা জিনিস ঠান্ডা মাথায় বুঝার চেষ্টা করো,

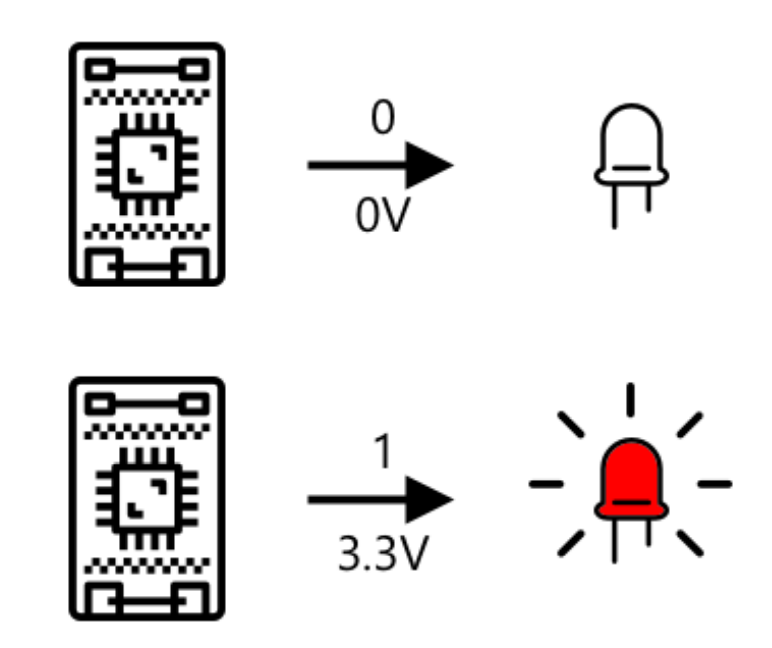
ধরো তুমি একটি মেশিন বানালে

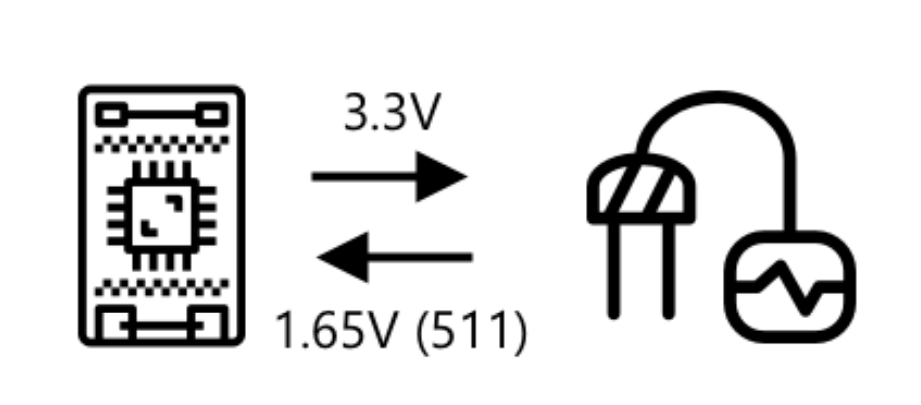
**LED on/off Using Push Button: Digital Input & Output**

যখন তুমি পুশ বাটনটা প্রেস করবে তখন বাটন থেকে একটি ইলেক্ট্রিক সিগন্যাল মাইক্রোকন্ট্রোলারের ঐ পিনে যাবে। যেহেতু এটি ডিজিটাল সিগন্যাল তাই এর মান(value) হয় 0 অথবা 1। ভোল্টেজের মান দিয়ে বলতে গেলে 0V কিংবা 3.3V।

প্রোগ্রামে আমরা বলে দিয়েছি যখন button এর ভ্যালু যখন 1 হবে তখন led এর ভ্যালুও 1 করে দিবে অর্থ্যাৎ এলইডি এর পাঁচ নাম্বার পিন দিয়ে একটি ইলেক্ট্রিক সিগন্যাল পাঠাবে যার ভোল্টেজ 3.3V, ফলে এলইডিটি জ্বলে যাবে।

আর যখন button এর ভ্যালু 0 হবে তখন led এর ভ্যালুও 0 হবে,এলইডিটি নিভে যাবে।





**API**

তুমি নিশ্চই গুগল ম্যাপ কখনও না কখনো ব্যবহার করেছো।দেশের সম্পূর্ণ রাস্তা-ঘাট,কোথায় কোন স্থাপনা আছে তা সব দেখা যায় গুগল ম্যাপে।এই ম্যাপ দেখতে কি গুগলকে তুমি টাকা দিচ্ছো? না,তাহলে এতো ভালো সার্ভিস তারা কিভাবে দিতে পারছে??

তুমি নিশ্চই বাংলাদেশের কোনো না কোনো ফুড ডেলিভারি সার্ভিস থেকে খাবার অর্ডার করেছো কিংবা রাইড শেয়ারিং সার্ভিস থেকে রাইড নিয়েছো। এই সার্ভিসগুলো অ্যাপে কিন্তু রিয়েল টাইমে ম্যাপে দেখা যায় তোমার অর্ডার করা খাবার বা রাইডটি এখন কোথায় আছে,তোমার কাছে আসতে কতক্ষণ লাগবে।  
  
একটু চিন্তা করো, গুগল ম্যাপ তৈরি করতে গুগলকে কিন্তু কোটি কোটি টাকা খরচ করতে হয়েছে।তাদের নিজস্ব স্যাটালাইট আছে তা দিয়ে রাস্তা-ঘাটের ম্যাপিং করতে পারে।কিন্তু আমরা যে ফুড ডেলিভারি সার্ভিস বা রাইড শেয়ারিং সার্ভিস ব্যবহার করছি তাদের কি স্যাটালাইট আছে? তারা কি ম্যাপ তৈরি করতে কোটি কোটি টাকা খরচ করেছে? উত্তর হচ্ছে-না।  
  
তাহলে তাদের অ্যাপে যে ম্যাপ দেখা যায় তা কি করে আসলো? উত্তর হচ্ছে- API

ঐ সার্ভিসগুলো গুগল ম্যাপের API ব্যবহার করছে তাদের অ্যাপে বিনিময়ে তাদের টাকা দিতে হচ্ছে গুগলকে।

API(Application Programming Interface) হচ্ছে কিছু প্রোগ্রামিং ফাংশন যেগুলো সফটওয়্যার ডেভেলপাররা তৈরি করে যার সাহায্যে তাদের সার্ভিস বা ডাটা অন্য কেউ ব্যবহার করতে পারে।

API ব্যবহার করার জন্য গুপন API Key থাকে।বিভিন্ন সার্ভিসের API ফ্রি হতে পারে আবার পেইড ও হতে পারে।যেমনঃ ধরো কোন ফুড ডেলিভারি সার্ভিস গুগল ম্যাপের ডাটা ব্যবহার করতে চায় তাদের নিজস্ব অ্যাপে।তখন ঐ সার্ভিস গুগলের কাছে যাবে এবং ম্যাপের জন্য পে করবে। এরপর গুগল তাদের কে একটি সিক্রেট API Key দিবে।

ঐ ডেলিভারি সার্ভিসের অ্যাপে এই API Key ব্যবহার করে গুগল ম্যা্পে ডাটার জন্য রিকোয়েস্ট পাঠাবে। গুগল ম্যাপ তখন ঐ রিকোয়েস্টের API Key ভেরিফাই করে দেখবে।যদি সব ঠিক থাকে তবে ডাটা দিয়ে দিবে।

সহজ ভাবে বললে এভাবেই কোনো API কাজ করে।

API কিন্তু খুবই পাওয়ারফুল একটি টুল।ধরো তুমি একটি ডিভাইস বানাতে চাচ্ছো যাতে আবহাওয়ার অবস্থা স্ক্রিনে দেখাবে।সব সময় তোমার কাছে কিন্তু সেন্সর থাকবে না,অনেক সময় সেন্সরের দাম অনেক বেশি হতে পারে,কিছু দিন ব্যবহার করার পর নষ্টও হয়ে যেতে পারে। তুমি যদি আবহাওয়ার ডাটা দেয় এমন কোনো সার্ভিসের API ব্যবহার করো তাহলেই কিন্তু সেন্সরের ঝামেলার সমাধান হয়ে যাচ্ছে।

আবার ধরো ফেসবুকের API ব্যবহার করে তোমার পেইজের ফলোয়ার সংখ্যার লাইভ আপডেট দেখাবে এমন ডিভাইস বানালে। একজন ডেভেলপার গিট হাব এর API ব্যবহার করে গিট হাবের কোড ম্যাপ দেখার ডিভাইস তৈরি করেছে।এটা উনি প্রোডাক্ট আকারে বিক্রর প্রস্তুতি নিচ্ছেন। এটা কিন্তু বেশ কুল! একটা প্রজেক্ট বিশেষ করে প্রোগ্রামারদের জন্য।

**JSON**

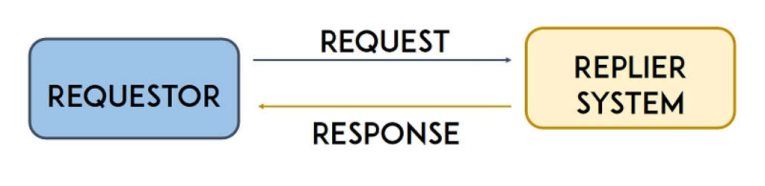
এটা তো বুঝলে, API এর মাধ্যমে আমরা ভিন্ন সফটওয়্যারের মধ্যে ডাটা আদান-প্রদান করে থাকি।চিন্তা করে দেখো তো, ভিন্ন ভিন্ন সফটওয়্যার ভিন্ন ভিন্ন টেকনোলজি,প্রোগ্রামিং ল্যাংগুয়েজ ব্যবহার করে তৈরি।এক প্রোগ্রামিং ল্যাংগুয়েজের সাথে অন্যটির সিনট্যাক্সে মিল নেই,ডাটা টাইপ ভিন্ন।তাহলে আমার সফটওয়্যার যদি পাইথনে লিখা হয় এবং ওয়েদার এর সফটওয়্যার যদি জাভাস্ক্রিপ্টে লিখা হয় তাহলে আমার সফটওয়্যার কি করে ঐ সফটওয়্যারে ডাটা বুঝবে?

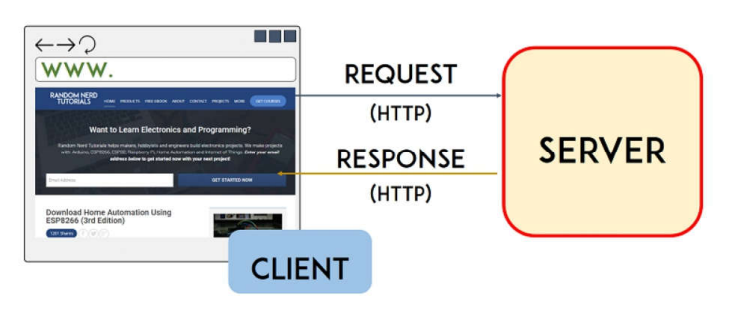
এই সমস্যার সমাধান করে JSON(JavaScript Object Notation)। JSON কে ডাটা আদান-প্রদানের জন্য গ্লোবাল স্ট্যান্ডার্ড ফরমেট ধরা হয়।অর্থ্যাৎ JSON ফরমেটের ডাটা যেকোনো প্রোগ্রামিং ল্যাংগুয়েজ বুঝতে পারে।

JSON এ ডাটা name/value প্যায়ার আকারে আসে(অনেকটা পাইথনের ডিকশেনারির মতো)। নেইম আর ভ্যালু কোলন( : ) দিয়ে আলাদা করা হয় কমা( , ) দিয়ে। দ্বিতীয় বন্ধনীর { } এর ভেতর

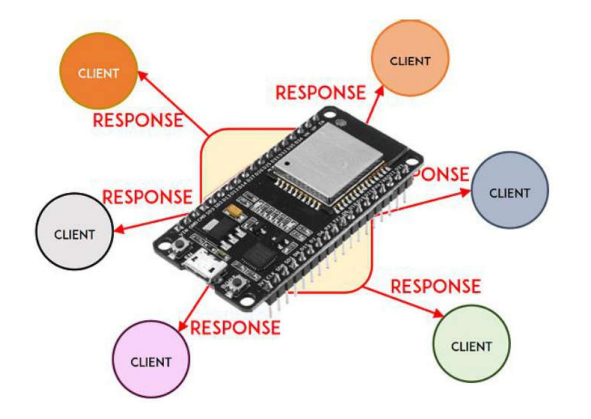
**HTTP Web Server**

পূর্বের চ্যাপ্টারে তুমি সার্ভার সম্পর্কে জেনেছো।সার্ভার হচ্ছে একটি মেশিন/কম্পিউটার যাতে কোনো ডাটা বা ওয়েবপেইজ স্টোর থাকে। ক্লাইন্ট ডিভাইস সার্ভারের কাছে Hyper Text Transfer Protocol(HTTP) এর মাধ্যমে রিকোয়েস্ট পাঠায়।সেই রিকোয়েস্টের জবাবে সার্ভারও HTTP এর মাধ্যমে রেসপন্স করে,তার ডাটা/ওয়েব পেইজ শেয়ার করে ক্লাইন্টের সাথে। এই ওয়েবপেইজ আমরা ব্রাউজারে দেখতে পাই।

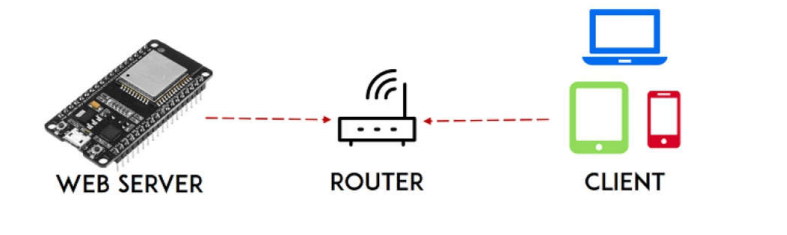
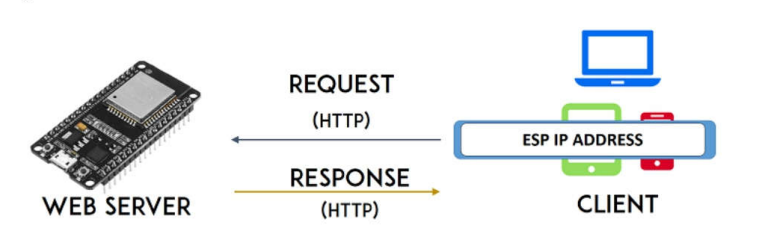




আমরাও ঠিক একই ছোট্ট একটি সার্ভার আমাদের লোকাল নেটওয়ার্কে(Local Network) তৈরি করবো ESP32 এর মাধ্যমে এই চ্যাপ্টারে।



লোকাল নেটওয়ার্কে কানেক্টেড থাকার অর্থ হলো আমাদের বাসায় যে রাউটার আছে তার সাথে আমাদের ESP32 কানেক্টেড হবে।একই রাউটারের সাথে কানেক্টেড অন্য ডিভাইস যেমনঃ স্মার্ট ফোন, ল্যাপটপ, কম্পিউটার ইত্যাদি যদি Wi-Fi কিংবা Ethernet এর সাথে কানেক্টেড থাকে তবে সেই ডিভাইস দিয়ে আমরা ESP32 এর IP Address ব্যবহার করে সার্ভারের কাছে HTTP রিকোয়েস্ট পাঠাতে পারবো।

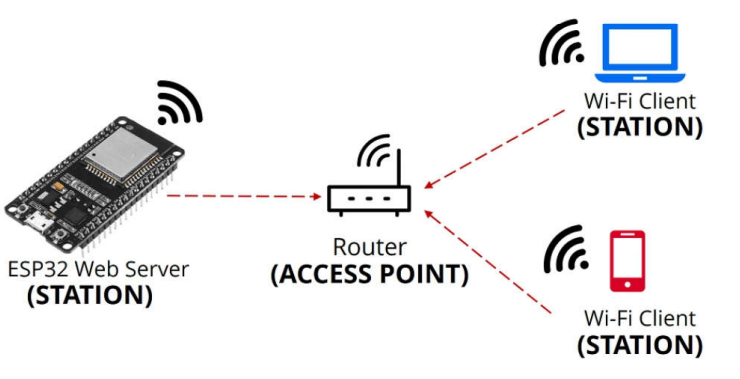
 

ESP32 সার্ভারে যে ওয়েবপেইজ আমরা স্টোর করে রাখবো তা আমাদের ক্লাইন্ট ডিভাইসের ব্রাউজারে দেখতে পারবো।

**Access Point Vs Station**

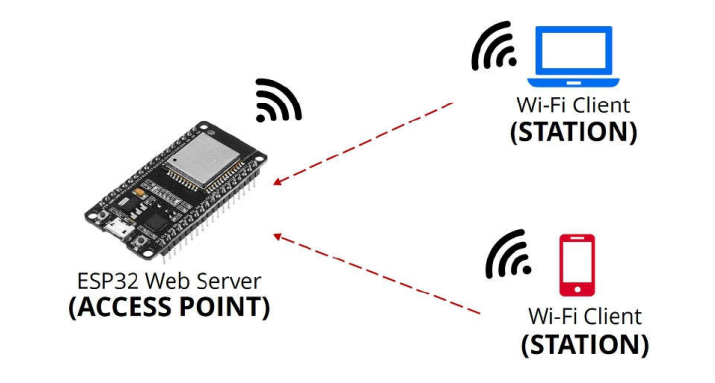
ESP32 কে ওয়েব সার্ভার হিসেবে দুই ভাবে ব্যবহার করা যায়।

১)Station Mode: এই মোডে ESP32 নিজে কোনো এক্সেস পয়েন্ট বা হটস্পট তৈরি করে না। বরং রাউটারের এক্সেস পয়েন্টের(Hotspot) সাথে যুক্ত হয়। সহজ ভাবে বললে,আমরা স্মার্ট ফোনে যেমন Wi-Fi এর SSID তে ক্লিক করে, পাসওয়ার্ড দিয়ে রাউটারের সাথে কানেক্টেট হই ঠিক তেমনি আমাদের ESP32 কেউ কানেক্টেড হতে হয়।



**Access Point Mode**

এই মোডে ESP32 রাউটারে সাথে যুক্ত হয় না বরং নিজেই হটস্পট বা এক্সেস পয়েন্ট তৈরি করে। অন্য ডিভাইস এই হটস্পটের সাথে Wi-Fi এর মাধ্যমে কানেক্টেড হয়। ESP32 এর এক্সেস পয়েন্টের ssid এবং পাসওয়ার্ড আমরা প্রোগ্রামিং করে সেট করে দিতে পারি।এক্ষেত্রে কোনো রাউটারের প্রয়োজন হয় না।যেহেতু ESP32 কোনো wired network যেমন রাউটারের সাথে যুক্ত হচ্ছে না তাই একে Soft-AP(soft Access Point) বলা হয়।



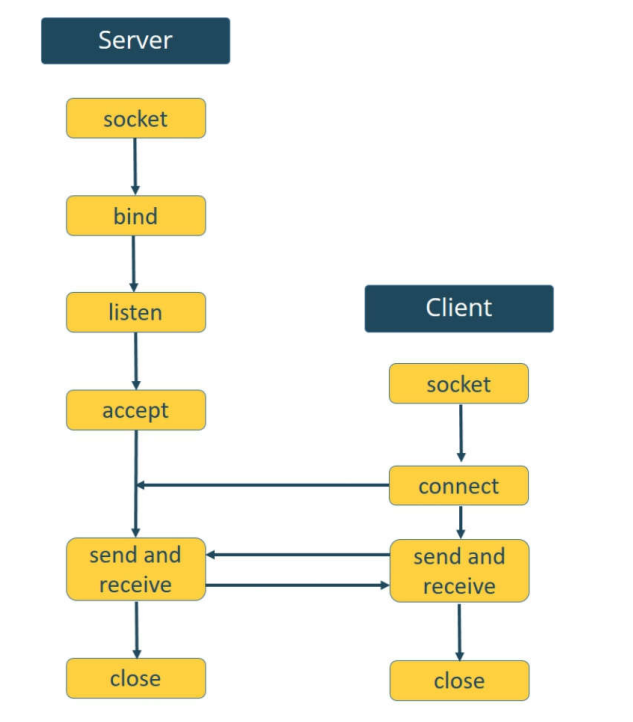
**Sockets**

প্রোজেক্ট শুরু করার আগে আমাদের আরেকটি ছোট্ট জিনিসের সাথে পরিচিত হতে হবে সেটি হচ্ছে web Sockets।

Sockets এর কন্সেপ্ট খুবই সিম্পল। ক্লাইন্ট-সার্ভার কানেক্টড তো হলো,সেটা স্টেশন মোড কিংবা এক্সেস পয়েন্ট মোড যেকোনো একটায়,এখন এদের মাঝে তথ্য আদান-প্রদান কি পদ্ধতিতে হবে?

এই তথ্য আদান-প্রদানের কাজটিই করে Sockets। Sockets এক ধরণের inter-process communication(IPC) প্রোভাইড করে,সহজ ভাষায় লোকাল নেটওয়ার্কে যুক্ত ডিভাইসগুলো কি ভাবে নিজেদের মাঝে ডাটা আদান-প্রদান করবে সেই প্রসেসটাই সহজ করে দেয়।

পাইথনে নিজস্ব sockets API আছে।মাইক্রোপাইথনে কাজ করার সময় এই API ব্যবহার করেই Sockets এর কাজ করা যাবে।



উপরের ডায়াগ্রাম দেখলে sockets কিভাবে কাজ করে তা ভালো ভাবে বুঝতে পারবে।সার্ভার সাইটে প্রথমে আমাদের “listenting sockets” তৈরি করতে হবে। এই listenting sockets এর কাজ সব সময় কান পেতে থাকা-কোনো ক্লাইন্ট কানেক্ট হতে চাচ্ছে কিনা।

listenting sockets এর জন্য socket module এ চারটি মেথড আছেঃ

* socket( )
* bind( )
* listen( )
* accept( )

স্টেপ-১) সকেট তৈরি করা socket.socket( ) এর মাধ্যমে এবং সকেট টাইপ বলে দেয়া।আমরা এখানে STREAM TCP সকেট ব্যবহার করছি।

স্টেপ-২) এখন এই সকেটকে একটি এড্রেসে বেঁধে দিব বা bind করে দিবো। এই জন্য bind( ) মেথডের ভেতর একটি টাপল ভেরিয়েবল দিতে হবে যার প্রথমে থাকবে IP Address এবং পরে Port Number: bind( (ip, port) )

স্টেপ-৩) এখন আমরা ক্লাইন্ট কানেকশনের জন্য সার্ভারকে তৈরি করবো listen( ) এর মাধ্যমে। এর মাঝে বলে দিবো সর্বোচ্চ কয়টি ডিভাইস কানেক্ট হতে পারবে।

স্টেপ-৪) while loop এর ভেতর কানেকশন রিকোয়েস্টের জন্য সার্ভার সব সময় কান পেতে থাকে(listen)।যখনই কোনো ক্লাইন্ড কানেকশন রিকোয়েস্ট পাঠায় তখনই accept( ) মেথড সেটি এক্সেপ্ট করে।

conn, addr = s.accept( )

conn ভেরিয়েবলের মধ্যে ডাটা স্টোর হয় যেগুলো সার্ভার-ক্লাইন্ট এর মাঝে আদান-প্রদান হবে। আর addr এর মাঝে ক্লাইন্টের আইপি এড্রেস স্টোর হয়।

স্টেপ-৫) recv( ) মেথডের মাধ্যমে ক্লাইন্ট থেকে ডাটা রিসিভ হয়। send( ) মেথডের মাধ্যমে সার্ভার থেকে ডাটা ক্লাইন্টকে পাঠানো হয়। সব শেষে close( ) মেথডের মাধ্যমে সকেটকে বন্ধ করে দেয়া হয়।

নোটঃ এই সব কিছু তোমার কাছে অনেক জটিল লাগতে পারে।চিন্তার কোনো কারণ নেই।কোনো কোড তোমাকে মুখস্থ করতে হবে না।জাস্ট প্রসেসটা বুঝার চেস্ট করো।

সেইম প্রসেস তুমি এলইডি লাইটের পরিবর্তে অন্য যেকোনো ইলেক্ট্রনিক্স ডিভাইস অন/অফ করতে পারো রিলে মডিউল ব্যবহার করে।

ওয়েব পেইজটি দেখতে আরো সুন্দর আর এডভান্স ফাংশনালিটি যুক্ত করতে CSS এবং JavaScript ব্যবহার করতে পারো।নিচের উদাহরণে এই প্রোজেক্টটাই CSS, JavaScript ব্যবহার করে দেখানো হলো।

<style> </style> ট্যাগের ভেতর আমরা CSS লিখি আর <script> </script> ট্যাগের ভেতর JavaScript।এই বইয়ে এগুলো নিয়ে আর ডিটেইলস লিখছি না। তুমি নিজ দায়িত্বে অল্প কিছু HTML, CSS, JavaScript শিখে নিবে।এগুলো শেখা খুবই সহজ,জাস্ট একটা সপ্তাহ টাইম দিতে হবে অনলাইনের কোনো কোর্সে।

**MQTT Protocol**

আমরা এতোক্ষণ শিখেছে HTTP এর মাধ্যমে কিভাবে ওয়েব পেইজ শেয়ার করা যায়,ESP32 কে station mode এবং access mode এর মাধ্যমে লোকাল নেটওয়ার্কে যুক্ত করেছি এবং কিভাবে web sockets এর মাধ্যমে লোকাল নেটওয়ার্কে যুক্ত ডিভাইসের মধ্যে ডাটা আদান-প্রদান করা হয়।

খেয়াল করে দেখো, এই সব কিছুই কিন্তু লোকাল নেটওয়ার্কে সীমাবদ্ধ।লোকাল নেটওয়ার্কের Wi-Fi রেঞ্জ যতটুকু ঠিক ততটুকুতেই তুমি ডাটা আদান-প্রদান করতে পারবে,এর বাইরে সম্ভব নয়।

কিন্তু IoT এর মূল কন্সেপ্টাই তো ছিল আমি পৃথিবীর যেকোনো প্রান্ত থেকে কমিউনিকেট করতে পারবো!সেটা কিন্তু এখনো করা হয় নি।

এই পার্টে আমরা দেখবো কি করে বিশ্বের যেকোনো প্রান্ত থেকে ডাটা আদান-প্রদান করা যায়।আর এটার জন্য আমাদের MQTT প্রোটোকল সম্পর্কে জানতে হবে।

Broker হিসেবে বিভিন্ন সার্ভিস ব্যবহার করা যায়।যেমনঃ Mosquito broker বেশ জনপ্রিয় ব্রোকার তবে এটা রাস্পবেরি পাই তে ইন্সটল করতে হয়। এছাড়া কিছু ক্লাউড ব্রোকার সার্ভিস আছে।আমরা ক্লাউড ব্রোকার সার্ভিস ব্যবহার করে একটি প্রোজেক্ট করবো।

MQTT communication project (local computer)

**Cloud**

আমরা যখন WLAN এর ভেতর কানেক্টেড ডিভাইসের মধ্যে ডাটা আদান-প্রদান করি তখন একটা নির্দিষ্ট সীমার মাঝেই কমিউনিকেশন সম্ভব হয়।পৃথিবীর যেকোনো প্রান্ত থেকে কমিউনিকেশন সম্ভব নয়।আবার এতে করে সিকিউরিটি রিস্কও আছে।

মনে রাখবে, তুমি যদি এমন আইওটি ডিভাইস তৈরি করতে চাও যা বিশ্বের যেকোনো জায়গা থেকে সব সময় এক্সেস করা যাবে তবে অবশ্যই তোমার ডিভাইসকে ক্লাউডের সাথে কানেক্টেড থাকতেই হবে। এখানে ক্লাউড সার্ভিস হিসেবে-

কোনো আইওটি সার্ভিস প্ল্যাটফর্ম হতে পারে,যেমন- ThinkSpeak,Ubedot

কোনো ক্লাউড বেইজড ডাটাবেজ হতে পারে,যেমন- MongoDB, Firebase

কিংবা কোনো কোনো ক্লাউড সার্ভিস প্ল্যাটফর্ম,যেমন- Microsoft Azure, Amazon AWS, Google Cloud Services

**ThinkSpeak**

ThinkSpeak একটি IoT সার্ভিস প্ল্যাটফর্ম। এর মাধ্যমে পৃথিবীর যেকোনো জায়গা থেকে ডাটা দেখা সম্ভব।সেই সাথে এই প্ল্যাটফর্ম পাসওয়ার্ড প্রোটেক্টেড,তাই ডাটার সিকিউরিটি থাকে।তাছাড়া ThinkSpeak এর মোবাইল অ্যাপ ThingView App এও ডাটা মনিটরিং করা যায়।

ThinkSpeak এর নিজস্ব broker আছে।তার মানে এই ব্রোকারের সাথেই MQTT এর মাধ্যমে কানেক্টেড হওয়া সম্ভব।ThinkSpeak এর মতো আরো অনেক আইওটি সার্ভিস প্ল্যাটফর্ম রয়েছে।যেমনঃ

**Database**

তুমি এখন বলতে পারো,” ভাইয়া, ThinkSpeak তো অন্যের বানানো প্ল্যাটফর্ম,তারা যে ভাবে প্ল্যাটফর্মটা ডিজাইন করে আমাকে তার মাঝেই কাজ করতে হয়,ওয়েব এড্রেসও তাদের। আমি কি করে এমন সফওয়্যার/অ্যাপ/ওয়েবসাইট বানাবো যাতে আমার আইওটি ডাটাগুলো দেখাবে যেমনটা কোনো কম্পানি তাদের আইওটি প্রোডাক্টের বেলায় করে থাকে?”

এর উত্তর হচ্ছে ডাটাবেজ। ডাটাবেজে আমাদের আইওটি ডাটাগুলো স্টোর হবে। এরপর তুমি যদি মোবাইল অ্যাপ বানাতে চাও কিংবা ওয়েবসাইট বানাতে চাও তবে ঐ অ্যাপ/ওয়েবসাইটকে ডাটাবেজের সাথে কানেক্ট করে দিলেই হচ্ছে।

অ্যাপ/ওয়েবসাইট বানাতে তুমি যেকোনো টেকনোলজিই ব্যবহার করতে পারো।বইয়ের এই শেষ চ্যাপ্টারে কিভাবে ডাটাবেজে ডাটা পাঠানো যায় সেটা দেখানো হবে।

**এরপর কী?**

এই বইয়ের প্রথম চ্যাপ্টারে আমরা আইওটির কনসেপ্টচুয়াল বিষয়গুলো জেনেছি, দ্বিতীয় চ্যাপ্টারে মাইক্রপাইথন ও ইলেক্ট্রনিক্স শিখে ছোট ছোট প্রজেক্ট করেছি, তৃতীয় চ্যাপ্টারে আগের শেখা সব কিছু ব্যবহার করে আমরা আইওটি প্রজেক্ট তৈরি করেছি।

কিন্তু আইওটি শেখা কিন্তু এখানেই শেষ নয়।লাস্ট প্রোজেক্টে আমরা কেবল ডাটা আদান-প্রদান দেখেছি।কিন্তু তুমি যখন কাউকে ডাটাগুলো দেখাবে তখন কিন্তু ডাটাবেজের ঐ ডাটা দেখালে হবে না।কারণ শুধু ডাটা কেবল কিছু সংখ্যা আর সংখ্যা দেখে কোনো ইনসাইট পাওয়া যায় না।আর তাই প্রয়োজন ডাটা ভিজুয়ালাইজেশন (Data Visualization)।

আবার এই ডাটা ভিজুয়ালাইজেশন(গ্রাফ,চার্ট), একশন বাটন, এলার্ম-নোটিফিকেশন তোমাকে কোনো না কোনো অ্যাপ্লিকেশনের ভেতর দেখাতে হবে,সেটা হতে পারে মোবাইল অ্যাপ কিংবা ওয়েবসাইট। মোবাইল অ্যাপের জন্য তুমি Flutter বা Android Studio ব্যবহার করতে পারো আর ওয়েবসাইটের বানানো বিভিন্ন পথ আছে।পাইথন দিয়ে ওয়েব ডেভেলপমেন্ট করতে চাইলে প্রথমে HTML, CSS, JavaScript শিখবে,এরপর পাইথনের যেকোনো ওয়েব ফ্রেমওয়ার্ক(Flask/FastAPI/Django) শিখবে।

হার্ডওয়্যার নিয়ে কাজ করার প্রবল ইচ্ছা তোমার মধ্যে যদি কাজ করে তবে আর দুইটা জিনিস শিখে নাও-

১)পিসিবি ডিজাইন(PCB Design): ব্রেডবোর্ড দিয়ে প্রোটোটাইপিং করা যায় কিন্তু প্রোজেক্টটি ব্যবহার উপযোগী হয় না।যদি তোমার বানানো প্রজেক্টকে প্রোফেশনাল লুক দিতে চাও তবে পিসিবি তে শিফট করো।পিসিবি জিজাইন করার জন্য অনেক সফটওয়্যার আছে।Easy EDA একটি ফ্রি অনলাইন সফটওয়্যার।পিসিবি তৈরি করা মোটেই জটিল কিছু না।একবার শিখে ফেললে তোমার ডিজাইন করা পিসিবি তুমি ঘরেই বানাতে পারো বা অনলাইন সার্ভিস থেকে প্রিন্ট করে নিয়ে আসতে পারো।অনলাইনে এই নিয়ে প্রচুর টিউটোরিয়াল পেয়ে যাবে।

২)থ্রি-ডি প্রিন্টিং(3D Printing): তোমার প্রজেক্টকে পরিপূর্ণ রূপ দিতে প্রয়োজন এনক্লোজার(Encloser)। Fusion 360৹ বা অন্য যেকোনো ডিজাইনিং সফটওয়্যার দিয়ে থ্রি-ডি ডিজাইনের কাজ চালিয়ে নিতে পারবে।দেশে অনেকে থ্রি-ডি প্রিন্টিং সার্ভিস দিয়ে থাকে, তাদের কাছ থেকে প্রিন্ট করে নিয়ে আসতে পারো।কিংবা তোমার ভার্সিটিতে FABLAB থাকলে ঐখানে গিয়েও কাজ করতে পারো।

বুঝতেই পারছো এটা এক লম্বা সফর। এই পর্যন্ত যদি তুমি আসতে পারো তাহলে তোমাকে কনগ্রেটস! তুমি ইঞ্জিনিয়ারিং স্টুডেন্ট হিসেবে নিশ্চয় বেশ কনফিডেন্ট ফিল করছো।এইবার এই স্কিলগুলো কাজে লাগাও, প্রোজেক্ট বানাও, রিসার্চে কাজ করতে চাইলে ভার্সিটির ফেকাল্টিদের সাথে কাজ করো। শুধু বাংলাদেশে নয় পুরো বিশ্বেই দক্ষ মানুষের অনেক চাহিদা রয়েছে। তোমার কাজগুলো সবার সামনে তুলে ধরো এবং অন্যকে সার্ভিস অফার করো। দেখবে তোমার ভার্সিটিতে তুমি কি ভাবে কি ভাবে যেন ফেমাস হয়ে গেছো!

Trust me! It Works!