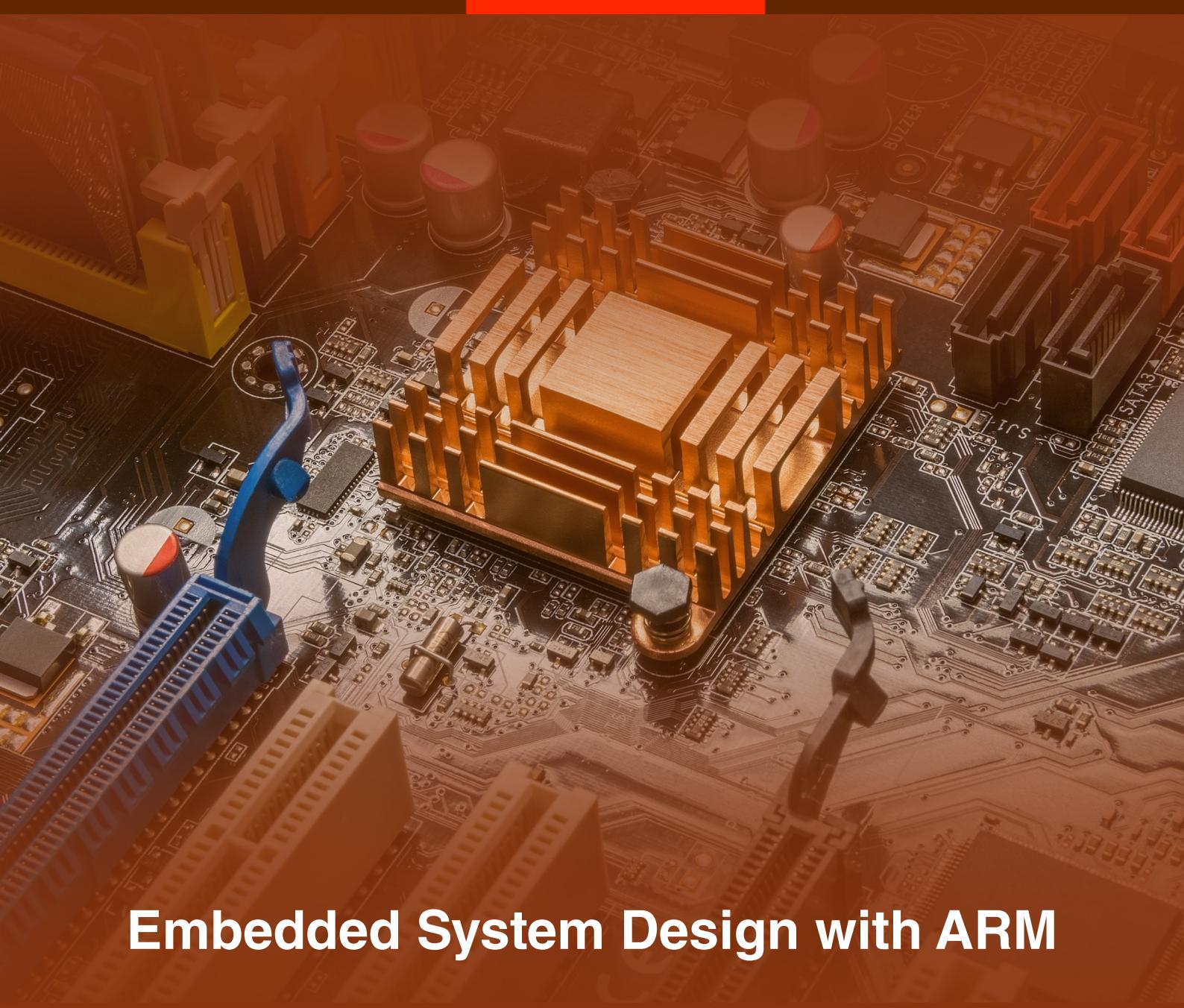


এমবেডেড সিস্টেম ডিজাইন উইথ ARM

বাংলা



Embedded System Design with ARM

Prof. Indranil Sen Gupta

Dr. Kamalika Datta

Computer Science and Engineering
IIT Kharagpur



Index

Sl. No	Topics	Page no
Week-1		
1.	এন্ডেড সিস্টেমের ভূমিকা	4
2.	এন্ডেড সিস্টেমগুলির ডিজাইনের বিবেচনা	16
3.	মাইক্রোপ্রসেসর এবং মাইক্রোকন্ট্রোলার্স	30
4.	আর্ম মাইক্রো কন্ট্রোলার শিল্পকলা (পর্ব 1)	45
5.	আর্ম মাইক্রো কন্ট্রোলার শিল্পকলা (পর্ব 2)	57
6.	আর্ম মাইক্রো কন্ট্রোলার শিল্পকলা (পর্ব 3)	69
Week-2		
7.	আর্ম ইন্ট্রাকশন সেট (পর্ব 1)	83
8.	আর্ম ইন্ট্রাকশন সেট (পর্ব 2)	95
9.	আর্ম ইন্ট্রাকশন সেট (পর্ব 3)	106
10.	STM32F401 নিউক্লিও বোর্ড বিষয়	116
11.	PWM এবং STM32F401 এর ইন্টারাপ্ট	129
Week-3		
12.	ডিজিটাল থেকে এ্যানলগ রূপান্তর	145
13.	এ্যানলগ থেকে ডিজিটাল রূপান্তর (পর্ব 1)	163
14.	এ্যানলগ থেকে ডিজিটাল রূপান্তর (পর্ব 2)	180
15.	আডিপুট ডিভাইস, সেন্সর এবং অ্যাক্টিউটর (পর্ব 1)	192
16.	আডিপুট ডিভাইস, সেন্সর এবং অ্যাক্টিউটর (পর্ব 2)	208
17.	আডিপুট ডিভাইস, সেন্সর এবং অ্যাক্টিউটর (পর্ব 3)	220
Week-4		

18.	মাইক্রোকন্ট্রোলার উন্নয়নের বোর্ড	231
19.	এমবেড সি প্রোগ্রামিং পরিবেশ	242
20.	Stm32f401 বোর্ডের সাথে ইন্টারফেসিং	255
21.	আরডুইনো ইউনের সাথে ইন্টারফেসিং	268
22.	ইন্টারফেসিং 7-বিভাগের Led এবং Lcd প্রদর্শন (পর্ব 1)	284
23.	ইন্টারফেসিং 7-বিভাগের Led এবং Lcd প্রদর্শন (পর্ব 2)	296

Week-5

24.	সিরিয়াল পোর্ট টার্মিনাল অ্যাপ্লিকেশন (শীতলতম)	310
25.	তাপমাত্রা সেন্সর সহ পরীক্ষা নিরীক্ষা	322
26.	Ldr লাইট সেন্সর সহ পরীক্ষা (পর্ব 1)	333
27.	Ldr লাইট সেন্সর সহ পরীক্ষা (পর্ব 2)	344
28.	স্পিকার সঙ্গে পরীক্ষা	354
29.	মাইক্রোফোনের সঙ্গে পরীক্ষা	369

Week-6

30.	কন্ট্রোল সিস্টেম ডিজাইন	381
31.	রিলে নিয়ে পরীক্ষা-নিরীক্ষা	396
32.	ডিসি মোটরের গতি নিয়ন্ত্রণের উপর পরীক্ষা-নিরীক্ষা	412
33.	একাধিক সেন্সর এবং রিলে পরীক্ষা	433

Week-7

34.	ইন্টারনেট অফ থিংস এর ভূমিকা	443
35.	জিএসএম এবং ক্লাউড	460
36.	হোম অটোমেশন সিস্টেমের ডিজাইন	476
37.	টাচ সেন্সর ব্যবহার করে একটি সাধারণ অ্যালার্ম সিস্টেমের	496

নকশা

Week-8

38.	অ্যাক্সিলোমিটার	508
39.	অ্যাক্সিলোমিটার ব্যবহার করে পরীক্ষা করুন	517
40.	ক্লটুথ ব্যবহার করে পরীক্ষা নিরীক্ষা	529
41.	গ্যাস সেন্সর নিয়ে পরীক্ষা নিরীক্ষা	559
42.	কোর্সের সংক্ষিপ্তকরণ	571

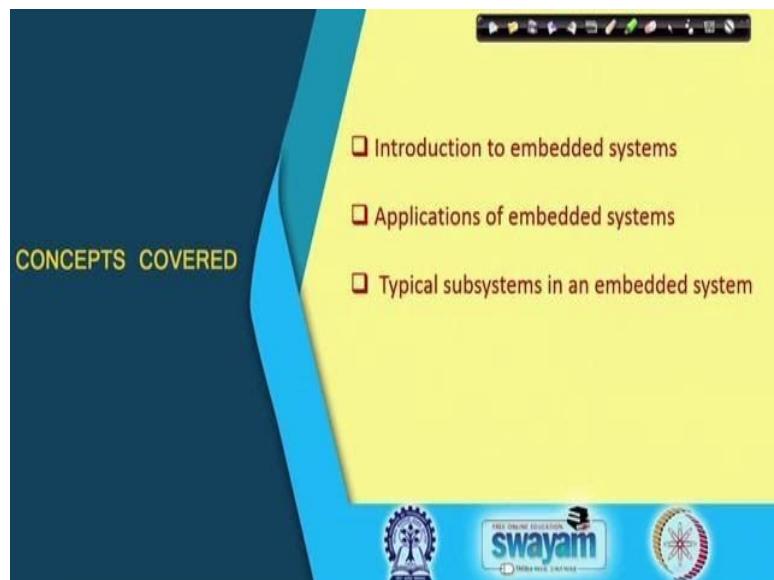
Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 01
Introduction to Embedded Systems

আমি এবং ডাঃ কমলিকা দত্ত যৌথভাবে পরিচালিত এই কোর্সে আপনাকে স্বাগত জানাই। এই কোর্সে আমরা এমবেডেড সিস্টেম(Embedded System) ডিজাইনের বিভিন্ন দিক সম্পর্কে কথা বলব, বিশেষত আমরা কয়েকটি নকশার উদাহরণে জোর দেব। demonstration এর বাহক হিসাবে আমরা মূলত দুটি পৃথক এমবেডেড সিস্টেম প্ল্যাটফর্মের সাথে কাজ করব; এর একটি ARM ভিত্তিক বোর্ডের উপর ভিত্তি করে এবং অন্যটি হবে একটি Arduino Uno ভিত্তিক বোর্ড।

এখন আসল পরীক্ষায় এগিয়ে যাওয়ার আগে আমাদের কেন একটি এন্ডেড সিস্টেমের প্রয়োজন, একটি এন্ডেড সিস্টেমের মূল বৈশিষ্ট্যগুলি কী এবং ডিজাইনের বিকল্পগুলি কী কী তার পিছনে কিছু তাত্ত্বিক পটভূমি (theoretical background) এবং কিছু অনুপ্রেরণা পাওয়া সর্বদা ভাল। সুতরাং, এই প্রসঙ্গে এই কোর্সের প্রথম বক্তৃতার শিরোনাম, এন্ডেড সিস্টেমগুলির পরিচিতি।

(Refer Slide Time: 01:40)



এখানে আমরা প্রাথমিকভাবে এন্ডেড সিস্টেম, এটি মূলত কী এবং কিছু সাধারণ অ্যাপ্লিকেশন এবং কোনও এমবেডেড সিস্টেমের মধ্যে আমরা সাধারণত কোন ধরনের সাধারণ সাবসিস্টেম গুলো দেখতে পাই তার বিষয়ে কথা বলব। এই কয়েকটি বিষয় যা আমরা এই কোর্সের অংশ হিসাবে আলোচনা করব।

(Refer Slide Time: 02:05)

Introduction

- We have been brought up in the age of computing.
- Computers are everywhere (some we see, some we do not see).
- Types of computers we are familiar with:
 - Desktops and Laptops
 - Servers
 - Mobile phones
- But there's another type of computing system that is often hidden.
 - Far more common and pervasive...
 - Hidden in the environment.

Embedded Systems

The slide features a video of a professor in the bottom right corner. The video player interface is visible at the top. The bottom of the slide has the 'SWAYAM' logo and the Indian emblem.

প্রথম যে বিষয়টির উপরে আমি জোর দিতে চাই তা হ'ল, আমরা খুব ভাগ্যবান যে আমরা জন্মগ্রহণ করেছি এবং কম্পিউটিংয়ের যুগে বেড়ে উঠেছি। আমাদের জন্মের পর থেকে অনেকেই চারপাশে কম্পিউটার সিস্টেম দেখেছি, তবে অবশ্যই 4 বা 5 দশক আগে দৃশ্যাটি এক রকম ছিল না।

ইলেকট্রনিক্স, কম্পিউটিং সিস্টেম, লো পাওয়ার ডিভাইসগুলির (low power devices) চমত্কার প্রসার ঘটেছে এবং কয়েক বছর ধরে বিভিন্ন ধরণের অগ্রগতি হয়েছে। প্রথম যে বিষয়টিটিতে আমি জোর দিতে চাই তা হ'ল কম্পিউটার সিস্টেম আজ সর্বত্র রয়েছে। আপনি যদি আপনার চারপাশে তাকান; আমি কারিগরি পরিবেশে কথা বলছি না, পরীক্ষাগারে নয়, আপনার কলেজেও নয় আপনার বাসভবনেও যখন আপনি নিজে ঘুরে থেকে জেগে ওঠেন।

আপনি যদি আপনার চারপাশে ঘুরে দেখেন তবে কয়েকটি কম্পিউটেশনাল ডিভাইসের কয়েকটি উদাহরণ পাবেন যা কোথাও বসে আছে। এটি এমন এক পরিবেশ যেখানে আমরা সকলেই বড় হয়েছি। গ্রিত্যাগতভাবে বলতে গেলে, আমরা এমন কম্পিউটারের সাথে পরিচিত হয়ে উঠলাম যেগুলি হয় ডেঙ্কটপ, ল্যাপটপ, সার্ভার ইত্যাদি যা কিছু দিক থেকে আরও প্রচলিত। ডেঙ্কটপ কম্পিউটারগুলি সাধারণত এমন মেশিন হয় যেখানে আপনি আপনার কম্পিউটিংটি শিখেছিলেন। আজ ল্যাপটপগুলি অনেক বেশি সাধারণ হয়েছে, আমাদের বেশিরভাগই ব্যক্তিগত ল্যাপটপ আছে। সুতরাং, ল্যাপটপগুলি ডেঙ্কটপগুলির আগে যে ভূমিকাটি ব্যবহার করত তার প্রতিস্থাপন করে। মোবাইল ফোনের বিষয়ে কথা বলা এটি একটি খুব কম মূল্যায়ন করা কম্পিউটিং ডিভাইস।

আজ যে কোনও মোবাইল ফোনের ভিতরে যা আছে, যদি আপনি 30 থেকে 35 বছর আগের কম্পিউটারগুলির গণ্য ক্ষমতা সম্পর্কে চিন্তা করেন তবে বিশ্বাস করুন যে কোনও মোবাইল ফোনের অভ্যন্তরে যে প্রক্রিয়াগুলি রয়েছে তার তুলনায় 1 মিলিয়ন গুণ বেশি শক্তিশালী আপনি বড়

কম্পিউটার সিস্টেমে দেখতে পাবেন। তবে মোবাইল ফোন আমরা কিছু নির্দিষ্ট অ্যাপ্লিকেশনের জন্য ব্যবহার করি, এটি ফোন হিসাবে ব্যবহারের জন্য, বার্তা প্রেরণ করে। এবং অবশ্যই, বিনোদনমূলক উদ্দেশ্যে আমরা ভিডিও দেখতে পারি, আমরা ইল্টারনেট ব্রাউজ করতে পারি, এবং এমন অনেকগুলি নতুন অ্যাপ্লিকেশন রয়েছে যা আপনি আজ মোবাইল ফোনে চালাতে পারেন।

তবে আমি যেটি বলতে চাই তা হ'ল আরও একটি ধরণের কম্পিউটিং সিস্টেম রয়েছে যা প্রায়শই আমাদের থেকে গোপন থাকে। লুকানো মানে আমরা সেই কম্পিউটিং সিস্টেমগুলি দেখতে পাচ্ছি না। যেমন আমরা জানি যে ডেস্কটপ গণনা করতে পারে, একটি ল্যাপটপ গণনা করতে পারে, একটি মোবাইল ফোনও কিছু দিক থেকে গণনা করতে পারে। তবে আপনি যদি একটি রেফ্রিজারেটরের দিকে তাকান, আমরা যদি আমাদের শীতাতপ নিয়ন্ত্রণ যন্ত্রটি দেখি তবে সেগুলি কি কোনও কম্পিউটিং মেশিনের মতো দেখাচ্ছে? না। তবে এই মেশিনগুলির ভিতরে কিছু কম্পিউটিং মস্তিষ্ক (computing brain) লুকিয়ে আছে, আমি এই লুকানো জিনিসকেই উল্লেখ করছি। এগুলি আমাদের প্রাত্যহিক জীবনে এবং সর্বাধিক প্রচলিত, কারণ এগুলি সর্বত্র রয়েছে এবং আমি যে বক্তব্যটি তৈরি করতে চাই তা হ'ল তারা যে পরিবেশের জন্য তৈরি হয়েছিল তাতে তারা লুকিয়ে আছে।

এখন, এই জাতীয় সিস্টেমগুলি প্রতিহ্যগতভাবে এমবেডেড সিস্টেম হিসাবে উল্লেখ করা হয়। এর অর্থ এগুলি কম্পিউটিং সিস্টেম, তবে সেগুলি পরিবেশের মধ্যে এস্বেড করা রয়েছে। পরিবেশ শব্দটি দ্বারা আমরা পরিপার্শ্বিকতা বা আসলে সেই দৃশ্যটি বোঝায় যার জন্য সিস্টেমটি নকশা করা হয়েছিল। আমি আবার একটি শীতাতপ নিয়ন্ত্রণ যন্ত্রের উদাহরণ নিতে পারি। একটি AC machine সাধারণত একটি ঘরের ভিতরে রাখার কথা। এখন এই কম্পিউটিং সিস্টেমটি যে কোনও AC machine এর অভ্যন্তরে বসে আছে, সেই কম্পিউটিং সিস্টেমের জন্য AC machine টি পরিবেশ। এটি সেই AC machine এর বাইরে কারও সাথে যোগাযোগ করে না।

এটি এসি মেশিন নিয়ন্ত্রণের জন্য দায়ী, আপনি আপনার রিমোট কন্ট্রোলটিতে যা চাপ দিচ্ছেন তার প্রতিক্রিয়া জানাতে দায়বদ্ধ, আপনাকে AC machine কে কিছু কমান্ড প্রদান করা হবে ইত্যাদি। এটি সেই ধরণের কমান্ড যা অভ্যন্তরীণ কম্পিউটারগুলি মেশিন প্রতিক্রিয়া জানাতে দায়বদ্ধ।

(Refer Slide Time: 07:47)

What are Embedded Systems?

- Computers are embedded within other systems:
 - What is "other systems"? – Hard to define.
 - Any computing system other than desktop / laptop server.
 - Typical examples:
 - Washing machine, refrigerator, camera, vehicles, airplane, missile, printer.
 - Processors are often very simple and inexpensive (depending on application of course).
 - Billions of embedded system units produced yearly, versus millions of desktop units.

এছেড়ে সিস্টেমগুলির বিষয়ে আবার কথা বলছি, এই এমবেডেড সিস্টেমগুলি কী কী? আমি ইতিমধ্যে একটি উদাহরণ দিয়েছি। সুতরাং, এখন এটি আপনার কাছে কিছুটা স্পষ্ট হওয়া উচিত। এছেড়ে করা সিস্টেমগুলি এমন কম্পিউটার যা তারা অন্যান্য সিস্টেমে এছেড়ে থাকে। এখন ডানদিকে কিছু ছবি দিয়েছি। আপনি এই উদাহরণগুলি দেখুন, এখানে আপনি একটি ওয়াশিং মেশিন দেখেন, এটি একটি ফ্রিজ, এটি একটি লেজার প্রিন্টার, এটি একটি ডিজিটাল ক্যামেরা, এটি একটি অটোমোবাইল এবং এখানে আপনি একটি মিসাইল দেখতে পাচ্ছেন যা আকাশের মধ্য দিয়ে উড়ে যাচ্ছে। এটি এমবেডেড সিস্টেমগুলির সমস্ত উদাহরণ, এগুলি অবশ্যই মোটামুটি শক্তিশালী কম্পিউটিং সিস্টেম বা তাদের অভ্যন্তরে ডিভাইসগুলি রয়েছে, এই কম্পিউটিং সিস্টেমগুলির শক্তি নির্দিষ্ট প্রয়োগের উপর নির্ভর করে।

কিছু অ্যাপ্লিকেশনের জন্য আপনার খুব বেশি গণনার শক্তি প্রয়োজন হয় না, তবে ক্ষেপণাস্ত্রের মতো ব্যবস্থার জন্য প্রচুর কম্পিউটেশনের প্রয়োজন রিয়েল টাইমে। ক্ষেপণাস্ত্রটি তার ট্রায়েল ট্র্যাজেক্টোরিতে (trajectory) প্রবাহিত হওয়ার সাথে সাথে পরিবেশ থেকে ক্রমাগত প্রতিক্রিয়া পাওয়া যায় এবং কিছু সংশোধনমূলক পদক্ষেপ গ্রহণ করা দরকার যা ক্ষেপণাস্ত্রটি সঠিক পথ অনুসরণ করতে পারে এবং একটি নির্দিষ্ট উপায়ে লক্ষ্যবস্তুতে আঘাত করতে পারে।

সুতরাং, এটি পরিবেশের উপর নির্ভর করে। এখন এই "অন্যান্য সিস্টেমগুলি" যেকোন কিছুতে উল্লেখ করতে পারে, খুব নির্দিষ্ট উপায়ে এটি নির্ধারণ করা খুব শক্ত। তবে বিস্তৃতভাবে আপনি বলতে পারেন যে এই অন্যান্য সিস্টেমগুলি কোনও কম্পিউটিং সিস্টেম, যা আমাদের প্রচলিত কম্পিউটার যেমন ডেস্কটপ, ল্যাপটপ, সার্ভার ইত্যাদি নয়।

যে কোনও কম্পিউটিং সিস্টেম যা এর মধ্যে একটি নয়, তারা কোথাও বসে পরিবেশের সাথে কাজ করছে, সেগুলি এমবেডেড সিস্টেম হিসাবে শ্রেণীবদ্ধ করা যেতে পারে। কিছু সাধারণ উদাহরণ এখানে দেওয়া হল। এবং আবার ঠিক যেমনটি আমি আবেদনের উপর নির্ভর করে বলেছি, প্রসেসরটি খুব

সহজ এবং সাধারণ হতে পারে। কারণ অনেকগুলি ডিভাইস রয়েছে, আমাকে আবার একটি উদাহরণ দেওয়া যাক। মনে করুন আপনি একটি মাইক্রোওয়েভ ওভেন কিনতে বাজারে গিয়েছেন, সেগুলি দামের জন্য ২,০০০ টাকার মধ্যে পাওয়া যায়। ৩০০০ এবং আরও বেশি।

তারা তুলনায় বেশ সম্ভা। মাইক্রো ওভেনের ভিতরে এক্সেড প্রসেসরও রয়েছে। এখন যদি আমি বলি যে আমার একটি মাইক্রোওয়েভ ওভেনের ভিতরে বসে থাকতে খুব শক্তিশালী পেন্টিয়াম প্রসেসর (Pentium Processor) প্রয়োজন, সেই পেন্টিয়াম প্রসেসরের দাম নিজেই সমস্ত আনুষাঙ্গিক এবং পেরিফেরাল সহ ৫০০০ ডলার হবে, তবে এটি অর্থনৈতিকভাবে টেকসই হবে না। অ্যাপ্লিকেশনগুলির উপর নির্ভর করে তাদের খুব সম্ভা হতে হবে।

বর্তমানে প্রচুর সংখ্যক এক্সেড সিস্টেম সারা বিশ্বে বিভিন্ন অ্যাপ্লিকেশনগুলিতে ব্যবহৃত হচ্ছে, কোটি কোটি প্রচলিত কম্পিউটিং ইউনিট। সুতরাং, এই জাতীয় এক্সেড কম্পিউটিং ডিভাইসগুলির সংখ্যা আজ প্রচলিত কম্পিউটিং ডিভাইসের তুলনায় অনেক বেশি।

(Refer Slide Time: 12:13)

Common Features of Embedded Systems

- They are special-purpose or single-functioned.
 - Executes a single program, possibly with inputs from the environment.
 - Imagine a microwave oven, a washing machine, an AC machine, etc.
- Tight constraints on cost, energy, form factor, etc.
 - Low cost, low power, small size, relatively fast.
- They must react to events in real-time.
 - Responds to inputs from the system's environment.
 - Must compute certain results in real-time without delay.
 - The delay that can be tolerated depends on the application.

The slide also features images of a microwave oven, a washing machine, and an air conditioner. At the bottom, there are logos for 'swayam' and 'FREE ONLINE EDUCATION'.

কিছু সাধারণ বৈশিষ্ট্য রয়েছে যা বেশিরভাগ এক্সেড সিস্টেমে উপস্থিত রয়েছে, আসুন আমরা সেগুলির কয়েকটি দেখি। প্রথম জিনিসটি হ'ল এগুলি বিশেষ উদ্দেশ্য বা একক কার্যকরী। এয়ার কন্ডিশনার মেশিনের মতো, ভিতরে একটি প্রসেসর রয়েছে, সেই প্রসেসরের একমাত্র উদ্দেশ্য হ'ল AC machine টি সঠিকভাবে কাজ করছে কিনা তা নিশ্চিত করা এবং অন্য কিছুই নয়। সুতরাং, এটি আপনাকে কিছু গণিত ইত্যাদিতে গণনায় সহায়তা করার চেষ্টা করছে না।

সুতরাং, এর অর্থ এটির বিশেষ উদ্দেশ্য। এটি সাধারণত যে অ্যাপ্লিকেশনের জন্য নির্মিত হয়েছিল তার সাথে সম্পর্কিত একটি একক প্রোগ্রাম সম্পাদন করে এবং এটি পরিবেশ থেকে ইনপুট গ্রহণ করতে পারে।

আপনি একটি এসি মেশিনের কথা ভাবেন; পরিবেশ থেকে প্রাপ্ত ইনপুটগুলির অর্থ, ব্যবহারকারী দূরবর্তী নিয়ন্ত্রণের মাধ্যমে কিছু কমান্ড প্রেরণ করতে পারে। এসি মেশিন নিজেই এর ভিতরে কিছু সেন্সর থাকতে পারে, বর্তমান ঘরের তাপমাত্রা কী, বর্তমান আর্দ্রতা স্তরটি কী তা বুঝতে পারে। তদনুসারে এটি খুব সর্বোত্তম উপায়ে ভিতরে বিভিন্ন সাবসিস্টেম সক্রিয় করতে পারে।

আপনি কোনও ফ্রিজের কথা ভাবেন সেখানে সাধারণত কিছু কন্ট্রোল প্যানেল থাকে। আপনি সেখানে তাপমাত্রা এবং অন্যান্য কারণগুলি সেট করতে পারেন। এটি আবার অ্যাপ্লিকেশনটির উপর নির্ভর করে, পরিবেশের মাধ্যমে আপনি প্রসেসরের কাছে কিছু কমান্ড প্রেরণ করতে পারেন, এবং প্রসেসর সেই অনুযায়ী প্রতিক্রিয়া জানাবে। মাইক্রোওয়েভ ওভেন, ওয়াশিং মেশিন, এসি মেশিনের মতো কয়েকটি উদাহরণ আমি দিয়েছি।

এবং এই এলেক্ট্রোলিটিক বেশিরভাগ সিস্টেমে ব্যয়, শক্তি এবং ফ্যাট্টের ফর্মের উপর খুব শক্ত প্রতিবন্ধকতা রয়েছে। প্রসেসরের স্বল্প ব্যয় করতে হবে অন্যথায় সিস্টেমটি অর্থনৈতিকভাবে টেকসই হবে না। প্রসেসরের অবশ্যই পাওয়ার সোর্স থেকে বা ব্যাটারি যেখানেই এটি কাজ করছে সেখানে খুব বেশি শক্তি খরচ করবে না। কারণ প্রসেসরটি আপনার কাছ থেকে লুকানো রয়েছে এবং এটি প্রত্যাশিত যে ব্যবহারকারীর শক্তি সম্পর্কে খুব বেশি চিন্তা করা উচিত নয়। ফর্ম ফ্যাট্টের; আপনি আজকাল এমন একটি এসি মেশিনের কথা ভাবেন যা খুব স্লিপ্প, এখন এর অভ্যন্তরে এমন কোনও কম্পিউটার রাখার সামর্থ নেই যা খুব ভারী এবং বড় কারণ এটি আপনার সিস্টেমকেও বিশাল এবং বিশাল করে তুলবে।

সুতরাং, সিস্টেমের উপর আবার নির্ভর করে আপনার কম্পিউটিং সিস্টেমটি অবশ্যই খুব কমপ্যাক্ট তৈরি করা উচিত এবং কোনও প্রশংসনীয় ক্ষেত্র বাড়ানো ছাড়াই অভ্যন্তরে ফিট হওয়া উচিত। সুতরাং, স্বল্প ব্যয়, স্বল্প শক্তি, ছোট আকার, তুলনামূলক দ্রুত এই বৈশিষ্ট্যগুলির কয়েকটি এবং অন্য একটি গুরুত্বপূর্ণ বিষয় হল এটির বাস্তব সময়ের প্রতিক্রিয়া হওয়া উচিত। আপনি এসি মেশিনের কথা ভাবেন, হঠাতে কোনও কারণে ঘরের তাপমাত্রা বেড়ে গেছে। তাপমাত্রা হ্রাস করতে এটি উচ্চ শক্তিতে সংক্ষেপকটি (compressor) চালু করতে হবে। সুতরাং, এটি অবশ্যই বাস্তব সময়ে এই বাহ্যিক ইনপুটগুলির প্রতিক্রিয়া জানাতে পারে, এটি 10 মিনিটের পরে ভালভাবে প্রতিক্রিয়া জানাতে পারে না।

রিয়েল টাইম মানে, যখনই কোনও ইনপুট আসে, নির্দিষ্ট সময়ের মধ্যে সিস্টেমের সেই ইনপুটটির প্রতিক্রিয়া জানানো উচিত। বেশিরভাগ এলেক্ট্রোলিটিক সিস্টেম রিয়েল টাইমে সাড়া দেয়। তারা সিস্টেমের পরিবেশ থেকে ইনপুট নেয় এবং গণনা পরিচালনা করার পর সহনীয় বিলম্ব প্রয়োগ থেকে প্রয়োগের ক্ষেত্রে পরিবর্তিত হয়। সুতরাং, এগুলি কয়েকটি সাধারণ বৈশিষ্ট্য।

(Refer Slide Time: 17:19)

Typical Design Constraints

- Low Cost
 - A sophisticated processor can increase the cost of the embedded system.
- Low Energy Consumption
 - Many embedded systems operate on battery.
- Limited Memory
 - Typically constrained to a finite and small amount of memory.
- Real-Time Response
 - Most embedded systems are used for controlling some equipment.
 - Must generate response within a specified time.

FREE CINEMA EDUCATION
swayam
THEATRE HOME & OFFICE

এখন, পাশাপাশি ডিজাইনের কিছু বাধাও রয়েছে; এর কয়েকটি বেশ সূস্পষ্ট। প্রসেসরের স্বল্প দামের জিনিস হওয়া উচিত। আপনার এটিকে খুব ব্যবহৃত করা উচিত নয় কারণ এটি আপনার সিস্টেমকে এস্বেডেড করে তুলবে, এটি বেশ ব্যবহৃতও করে তুলবে। আপনি যেমন একটি মেশিনের ভিতরে একটি খুব পরিশীলিত প্রসেসর ব্যবহার করতে পারবেন না; মাইক্রোওয়েভের মতো আপনি পেন্টিয়ামের মতো একটি পরিশীলিত প্রসেসর রাখতে পারবেন না, কারণ পেন্টিয়ামের দাম নিজেই বেশ বেশি।

স্বল্প জ্বালানি খরচ হল অন্য বৈশিষ্ট্য যা বেশ সাধারণ। এমনকি এমন মেশিনগুলির জন্যও যা বৈদ্যুতিক শক্তি থেকে চালিত হয় যেমন একটি ফ্রিজের মতো আপনি যদি এটি লাগিয়ে রাখেন এবং চলে যান তবে আশা করা যায় যে এটি খুব কম শক্তি ব্যয় করবে। এবং দ্বিতীয়ত, অনেক গ্যাজেট রয়েছে যা ব্যাটারিতেও পরিচালনা করে। তাদের জন্য স্পষ্টতই, শক্তি খরচ একটি বড় সমস্যা। এবং এই প্রসেসরগুলি খুব সহজ, কারণ তাদের কাছে মেমরির পরিমাণও খুব কম। সুতরাং, আপনি যেই প্রোগ্রাম লিখুন না কেন, আপনি যেই গণনা করেন না কেন তাদের অবশ্যই এই সীমিত মেমরির জায়গার ভিতরে ফিট করতে হবে। আমি উল্লেখ করেছি অনেক অ্যাপ্লিকেশন রিয়েল টাইম প্রতিক্রিয়া দাবি করে; একটি নির্দিষ্ট সময়ের মধ্যে, সিস্টেমের ইনপুটগুলিতে সাড়া দেওয়া উচিত।

(Refer Slide Time: 19:09)

How to define an Embedded System?

- It is a microcontroller-based system that is designed to control a function or range of functions, and is not meant to be programmed by the end user.
 - The user may make choices concerning the functionality but cannot change them.
 - The user cannot make modifications to the software.
 - Can you "program" your washing machine or refrigerator or car?
 - Not today ... but not very sure of the near future.

এখন, আমরা এতদূর যা কিছু কথা বলেছি তার ভিত্তিতে একটি এন্ডেড সিস্টেমকে কীভাবে সংজ্ঞায়িত করতে পারি? ভাল এন্ডেড সিস্টেমকে আলগাভাবে একটি মাইক্রোকন্ট্রোলার ভিত্তিক সিস্টেম হিসাবে সংজ্ঞায়িত করা যেতে পারে। এটি এন্ডেড সিস্টেমে সত্য। ভিতরে থাকা প্রসেসরের দাম কম হতে হবে এটি কম শক্তি হতে হবে, এটি আকারে ছোট হতে হবে। মাইক্রোকন্ট্রোলার নামে কিছু প্রসেসর রয়েছে যা আমরা এই কোর্স জুড়ে আলোচনা করব, তারা এই সমস্ত মানদণ্ডকে সন্তুষ্ট করে।

এন্ডেড বেশিরভাগ সিস্টেমে তাদের ভিতরে মাইক্রোকন্ট্রোলার থাকে, তারা একক চিপে সম্পূর্ণ কম্পিউটারের মতো। আপনার যা কিছু প্রয়োজন, প্রসেসর, মেমরি, আইও সবকিছু একক চিপের ভিতরে থাকে এবং তাই এটি খুব কম ব্যয় করে। সুতরাং, এটি একটি মাইক্রোকন্ট্রোলার ভিত্তিক সিস্টেম হওয়া উচিত এবং সাধারণ উদ্দেশ্য নয়। এটি একটি ফাংশন বা ফাংশনগুলির সীমিত সেটগুলির একটি ব্যাপ্তি নিয়ন্ত্রণ করতে হবে। এবং আরেকটি বিষয় হ'ল এটি ব্যবহারকারীর দ্বারা প্রোগ্রাম করা নয়। একটি ল্যাপটপ বা একটি ডেস্কটপের মতো আপনি কোনও প্রোগ্রাম লিখতে পারেন, ভালভাবে আপনি কোনও সংখ্যার ফ্যাক্টরিয়াল গণনা করতে, এন সংখ্যার সমষ্টি ইত্যাদির জন্য একটি প্রোগ্রাম লিখতে পারেন তবে যখনই আপনি আপনার এসি মেশিনের মতো কোনও ডিভাইসে কোনও এমবেডেড সিস্টেমের কথা ভাবেন, আপনি সাধারণত আপনার প্রতিদিনের গণনার জন্য এটি ব্যবহার করেন না, আপনি নিজে এটি প্রোগ্রাম করতে পারবেন না। এটি আপনার কাছে কারখানা থেকে প্রোগ্রামযুক্ত হয়ে এসেছে, আপনি প্রোগ্রামটি পরিবর্তন করতে পারবেন না, এটি একটি স্থির প্রোগ্রাম সিস্টেম, আপনি কেবল সেই প্রোগ্রামটিই ব্যবহার করছেন।

ব্যবহারকারী উদাহরণস্বরূপ, দূরবর্তী নিয়ন্ত্রণের মাধ্যমে ইনপুট দিতে পারে, তবে কার্যকারিভা পরিবর্তন করতে পারে না। উদাহরণস্বরূপ আপনি কোনও এসি মেশিনকে ক্রম হিটারের মতো অন্য কোনওটিতে পরিবর্তন করতে পারবেন না। ব্যবহারকারী সাধারণত স্টেওয়্যারটিতে কোনও পরিবর্তন করতে পারবেন না, তবে অবশ্যই আজকাল সিস্টেমগুলি আরও পরিশীলিত হয়ে উঠছে; আপনি একটি

সেট টপ বক্স মনে করেন যা আপনি টিভি সেটগুলির সাথে ব্যবহার করেন। সেট টপ বক্সটিতে এমবেডেড সিস্টেম রয়েছে, তাদের ভিতরে বেশ শক্তিশালী প্রসেসর রয়েছে। এখন কখনও কখনও আপনি লক্ষ্য করেছেন যে কিছু প্রোগ্রাম দেখার সময় বা আপনি যখন এটি চালু করছেন, এটি আপডেট ডাউনলোড করে বলে। ঠিক আছে যে সেট সফটওয়্যারটি সেট টপ বক্সের ভিতরে চলছে সেগুলি পরিষেবা সরবরাহকারী পর্যায়ক্রমে আপডেট হয়। সেটি সেট টপ বক্সের ভিতরে সরবরাহ করা হয়। ঠিক যেমন আপনার স্ট্যান্ডার্ড অপারেটিং সিস্টেমগুলির মতো Windows প্রায়শই ইনস্টল করার জন্য কিছু আপডেট প্রয়োজন।

তার অর্থ, আপনার বিদ্যমান স্ট্যান্ডার্ডটিকে কিছুটা অর্থে আরও উন্নত করতে আপনার কিছু সংশোধন প্রয়োজন। সুতরাং, সাধারণত আপনি নিজের ওয়াশিং মেশিন বা রেফ্রিজারেটর বা একটি গাড়ি প্রোগ্রাম করতে পারবেন না, তবে আগামী পরবর্তী প্রজন্মের এন্ডেড সিস্টেমগুলি আসার সাথে আপনি সেগুলিও প্রোগ্রাম করতে সক্ষম হতে পারেন। আপনি যখন কোনও যান চালনা করেন তখন আপনি এটি প্রোগ্রাম করতে পারেন এবং এটি আপনার প্রয়োজন অনুসারে কাস্টমাইজ করতে পারেন, তবে আজ হিসাবে সিস্টেমগুলি ব্যবহারকারীদের কাছে খুব সীমিত পছন্দ নিয়ে আসে; ব্যবহারকারীর নির্দিষ্ট কিছু জিনিস থাকতে পারে তবে সবকটিই নয়।

এটি একটি এমবেডেড সিস্টেমের স্ফিয়াটিক ডায়াগ্রাম। এন্ডেড কম্পিউটারটি মাঝখালে বসে আছে, যার কয়েকটি হার্ডওয়্যার, মাইক্রোকন্ট্রোলার এবং কিছু প্রোগ্রাম রয়েছে যা এর স্ট্যান্ডার্ডটিতে চলছে। এটি কিছু সেন্সরের মাধ্যমে ইনপুট নেয় এবং এটি কিছু আউটপুটের মাধ্যমে কিছু নিয়ন্ত্রণ করতে পারে এবং সাধারণত কিছু ব্যবহারকারীর ইন্টারফেস যেমন কিছু স্যুইচ, কিছু ছেট কীবোর্ড থাকে যেমন কিছু টাচ সেন্সর সম্ভবত রিমোট কন্ট্রোল থাকে, এগুলি সমস্ত ব্যবহারকারীর ইন্টারফেস এবং এটিতেও থাকতে পারে অন্যান্য সাবসিস্টিমে কিছু লিঙ্ক। এটি একটি এন্ডেড সিস্টেমের একটি সাধারণ স্ফিয়াটিক।

(Refer Slide Time: 24:09)

- What embedded system is not ...
 - A microprocessor sitting inside a traditional computing system (like desktop, laptop, server, etc.).
- It is actually:
 - A microprocessor used to control another piece of technology (dedicated, and not general-purpose).
 - For low cost, microcontrollers that are typically used are single-chip devices containing processor, memory, and I/O interfaces.

এখন, এমবেডেড সিস্টেমটি কী নয়? আচ্ছা এটি কোনও মাইক্রোপ্রসেসর নয় যা এর ভিতরে বসে প্রতিযোগিত কম্পিউটিং সিস্টেম। আমরা এটিকে কম্পিউটার বলি, আমরা এটিকে এমবেডেড সিস্টেম বলি না। হ্যাঁ এটি একটি মাইক্রোপ্রসেসর যা অন্য মেশিনের মতো এসি মেশিন, মাইক্রোওয়েভের মতো, ফ্রিজের মতো আরও কিছু প্রযুক্তি ব্যবহার করতে ব্যবহৃত হয়, এটি এমবেডেড সিস্টেম।

এটি অন্য কোনও সিস্টেমের মধ্যে এস্বেড করা রয়েছে। এখন যেমন আমি স্বল্পমূল্যের মাইক্রোকন্ট্রোলারদের জন্য বলেছিলাম যে তারা একক চিপ ডিভাইস। আপনি দেখুন এখানে আমি একটি মাইক্রোকন্ট্রোলারের একটি ছবি দেখিয়েছি যা PIC পরিবারের অন্তর্গত। এটি একটি PIC মাইক্রোকন্ট্রোলার, এটি একটি Layout Diagram যেখানে প্রসেসর, মেমরি, I/O সার্বিসেস সমন্বয় কিছু একই চিপের ভিতরে এস্বেড করা থাকে।

(Refer Slide Time: 25:17)

Applications of Embedded Systems

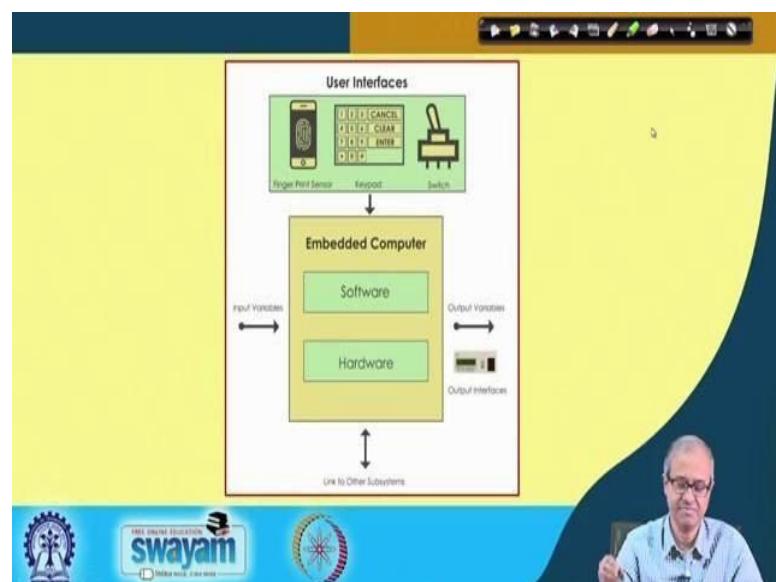
- *Limited by imagination.*
 - Consumer Segment: Refrigerator, washing machine, A/C machine, camera, microwave oven, TV, security system, etc.
 - Office Automation: Printers, Fax machines, photocopying machines, scanners, biometric scanner, surveillance camera, etc.
 - Automobiles: Air bags, anti-lock braking system (ABS), engine control, door lock, GPS system, vehicular ad-hoc network (VANET), etc.
 - Communication: Mobile phones, network switches, WiFi hotspots, telephones, MODEM, etc.
 - Miscellaneous: Automatic door locks, automatic baggage screening, surveillance systems, intelligent toilet, etc.

এখন এমবেডেড সিস্টেমগুলির কয়েকটি অ্যাপ্লিকেশন সম্পর্কে কথা বলছি, আপনি সেগুলির কয়েকটি সংক্ষিপ্ত করতে পারেন, তবে সেগুলি কল্পনা দ্বারা সীমাবদ্ধ, আপনি যা কল্পনা করতে পারেন তার কোনও সীমা নেই। এবং এই জাতীয় প্রয়োগের সংখ্যা দিন দিন কেবল বাড়বে। আজ আমরা আইওটি সম্পর্কে কথা বলছি, আগামীকাল আইওটি সর্বত্র থাকবে, সম্ভবত সেই সমস্ত ডিভাইসগুলির কিছু আপনার শরীরে থাকবে, আপনি সেই ডিভাইসগুলি আপনার সাথে বয়ে নিয়ে যাবেন।

ଗ୍ରାହକ ହିସେବେ ରେଫିଜ୍ୟାରେଟର, ଓ୍ୟାଶିଂ ମେଶିନ, ଏସି ମେଶିନ, କ୍ୟାମେରା ସମ୍ପକେ ଭାବତେ ପାରେନ, ଏହି ଇତିମଧ୍ୟେ ଆମରା କଥା ବଲେଛି । ଏମନକି ଆପନାର ଅଫିସେ ଆପନି ପ୍ରିନ୍ଟାର ଏବଂ ଫ୍ୟାକ୍ରୁ ମେଶିନ, ଫଟୋକପି ମେଶିନ, ସ୍କ୍ୟାନାର, ବାୟୋମେଡ଼ିକ ସ୍କ୍ୟାନାର, ନଜରଦାରି କରାର ଜନ୍ୟ କ୍ୟାମେରା ବ୍ୟବହାର କରେନ, ଏଗ୍ରଲ ଏମବେଡେଡ ସିଲ୍ଟେମେର ଉଦାହରଣ । ଆପନି ଅଟୋମୋବାଇଲଗ୍ନିଲିର କଥା ଭାବେନ; ଆଜ ସମସ୍ତ ଅଟୋମୋବାଇଲ କିଛୁଟା ଅର୍ଥେ ବୁନ୍ଦିମାନ, ଏମନ କଯେକଟି କମ୍ପିୟୁଟେଶନାଲ ଡିଭାଇସ ରଯେଛେ ଯେଗ୍ରଲି ଅଟୋମୋବାଇଲଗ୍ନିଲିର ଅଭ୍ୟନ୍ତରୀଣ କାଠାମୋକେ କାର୍ଯ୍ୟକର ଉପାୟେ ବ୍ୟବହାର କରାର ଚେଷ୍ଟା କରେ ।

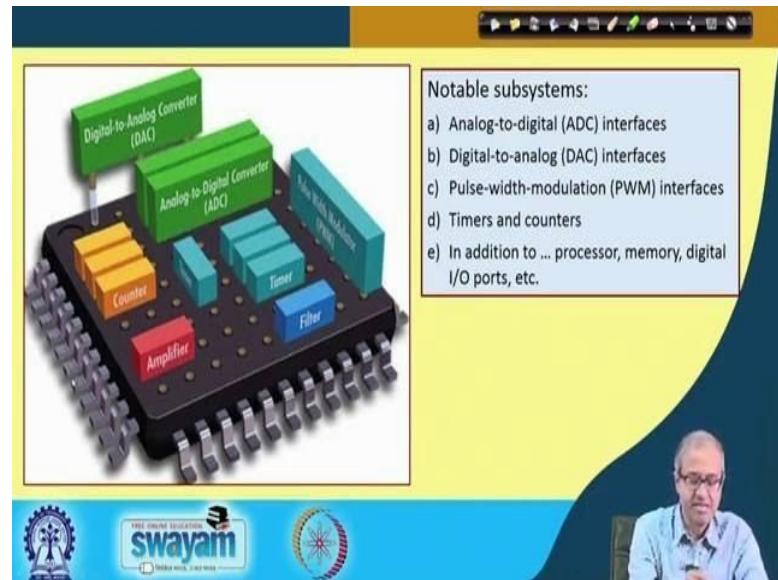
সুতোঁ, এয়ারব্যাগগুলি, অ্যান্টি-লক ব্রেকিং সিস্টেম, ইঞ্জিন নিয়ন্ত্রণ, দরজা লক, জিপিএস, এগুলি এখানে এষ্বেডেড প্রসেসরের মাধ্যমে নিয়ন্ত্রিত হয়। যোগাযোগের জন্য মোবাইল ফোন সবচেয়ে বড় উদাহরণ; আপনি যে নেটওয়ার্ক সুইচগুলি যোগাযোগের রাউটারগুলি, স্যুইচ, ওয়াই-ফাই হটস্পটস, টেলিফোনগুলির জন্য ব্যবহার করেন সেগুলির প্রত্যেকটির অভ্যন্তরে প্রসেসর রয়েছে তারপরে আপনার কাছে স্বয়ংক্রিয় দরজা লাকিং সিস্টেম থাকতে পারে আমরা এটিকে অনেকগুলি স্থান দেখি; বিমানবন্দর এবং রেল স্টেশনগুলিতে স্বয়ংক্রিয় ব্যাগেজ স্ক্রিনিং, নজরদারি সিস্টেম, বুদ্ধিমান ট্যুলেট, যেখানে ট্যুলেটের অভ্যন্তরে এমন কিছু এমবেডেড সিস্টেম রয়েছে যা ব্যবহারকারীদের প্রয়োজনগুলিতে স্বয়ংক্রিয়ভাবে সাড়া দিতে পারে এগুলি এমবেডেড সিস্টেমের উদাহরণ।

(Refer Slide Time: 27:29)



আবার সেই চিত্রটি নিয়ে কথা বলার সাথে সাথে দেখবো এন্ডেড সিস্টেমটি দেখতে এটির মতোই। আমাদের কিছু হার্ডওয়্যার সহ একটি এন্ডেড কম্পিউটার থাকতে হবে, এটি করার জন্য আপনাকে কিছু স্ট্রাইচের লিখতে হবে এবং অবশ্যই এটি ব্যবহারযোগ্য করে তোলার জন্য একটি উপযুক্ত ইউজার ইন্টারফেস থাকতে হবে। এবং ব্যবহারকারীর ইন্টারফেসটি যথেষ্ট সহজ হতে হবে, এটি একটি সম্পূর্ণ বড় কীবোর্ড হওয়া উচিত নয় যেখানে আপনি কিছু এবং সবকিছু টাইপ করতে পারেন। এটি একটি নির্দিষ্ট অ্যাপ্লিকেশনের জন্য খুব নির্দিষ্ট হবে।

(Refer Slide Time: 28:02)



এটি ঠিক একটি কার্টুনের মতোই আমি দেখিয়ে যাচ্ছি। এই আয়তক্ষেত্রাকার বাক্সটি একটি প্রসেসর। প্রসেসরের ভিতরে আপনি অনেক কিছুই দেখতে পাবেন। আপনি ডিজিটাল থেকে অ্যানালগ রূপান্তরকারী দেখতে পারেন, আপনি এনালগ-ডিজিটাল রূপান্তরকারী, কাউন্টার, টাইমার, পালস উইদ মড্যুলেটর(analog to digital converter, counters, timers, pulse width modulator), এতগুলি সাবসিস্টিমে দেখতে পারেন। প্রসেসর, মেমরি, IO পোর্ট(processor, memory, IO ports) ইত্যাদির পাশাপাশি একটি সাধারণ মাইক্রোকন্ট্রোলার যা আজ একটি এন্ডেড সিস্টেমের হস্য গঠন করে তাতে ডিজিটাল ইন্টারফেসের সাথে অ্যানালগ থাকতে পারে, কারণ বাইরের পৃষ্ঠিবীর বেশিরভাগ ইনপুট প্রকৃতিতে অ্যানালগ, এগুলি অবিরত, যেমন তাপমাত্রা, আর্দ্রতা, চাপ ইত্যাদি।

এটির সাথে আমরা এই বক্তৃতার শেষে এসেছি। আমরা আমাদের প্রবর্তী বক্তৃতায় আমাদের আলোচনা চালিয়ে যাব।

ধন্যবাদ।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 02
Design Considerations of Embedded Systems

আমাদের পূর্ববর্তী বক্তৃতায় আমরা এন্ডেড সিস্টেম (Embedded System)টি আসলে কী আর কী নয় তা নিয়ে কথা বলেছি। এই বক্তৃতায় আমরা এন্ডেড সিস্টেম ডিজাইনের কিছু নকশা বিবেচনা সম্পর্কে কথা বলব। যখন আমরা একটি এন্ডেড সিস্টেম ডিজাইন করি তখন আমাদের কোন জিনিসগুলি দেখতে হবে এবং ডিজাইনের বিবেচনা এবং অন্যান্য ট্রেড অফ(trade off)গুলি কী যা আপনার জন্য উচিত। এই বিষয়গুলি আমরা এই বক্তৃতায় বলব।

(Refer Slide Time: 00:52)

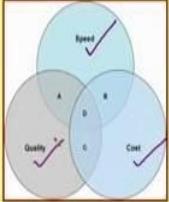


আমরা ডিজাইনের চ্যালেঞ্জগুলি সম্পর্কে ব্যাপকভাবে কথা বলব, এবং আমরা আরও গুরুত্বপূর্ণ কিছু ডিজাইনের ট্রেড অফগুলি(trade off) বোঝার চেষ্টা করব। ডিজাইন ট্রেড অফ মানে কিছু পরম্পরাবরোধী parameter থাকতে পারে; আপনি কোনও একটি parameter উন্নত করতে চেষ্টা করতে পারেন তবে এটি অন্য কোনও parameterকে হ্রাস করতে পারে। আপনি একসাথে সবকিছু উন্নত করতে পারবেন না, একে trade off বলা হয়। এবং এমন কিছু ব্যয় মেট্রিক রয়েছে যা nonrecurring unit cost matrix এর মতো বেশ গুরুত্বপূর্ণ। ঠিক আছে, আমরা এই সম্পর্কে কথা বলবো।

(Refer Slide Time: 01:31)

Design Challenges

- Primary design goal:
 - An implementation that realizes the desired functionality.
- The main design challenge is ...
 - To simultaneously optimize several design metrics.
 - Often mutually conflicting.
- What is a design metric?
 - Some feature of an implementation that can be measured and evaluated.

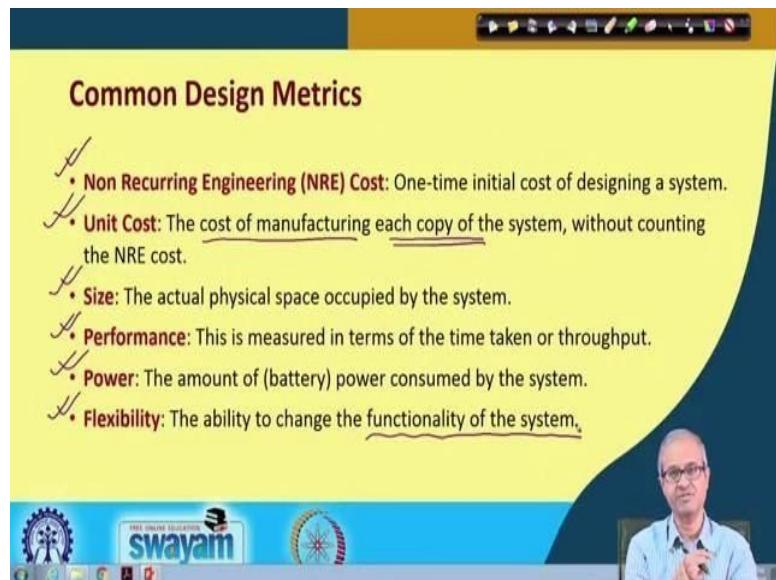



আসুন প্রথমে ডিজাইন চ্যালেঞ্জ সম্পর্কিত কিছু বিষয় সন্ধান করি। আপনি যখন এমবেডেড সিস্টেম ডিজাইন করছেন, আপনার প্রাথমিক ডিজাইনের লক্ষ্যটি এমন একটি সিস্টেম ডিজাইন করা হবে যা পছন্দসই কার্যকারিতাটি উপলব্ধি করে। মনে করুন, আপনি একটি ওয়াশিং মেশিনের জন্য একটি এমবেডেড সিস্টেম তৈরি করার চেষ্টা করছেন, সুতরাং আপনার এমবেডেড সিস্টেমটি সেই পরিবেশে সঠিকভাবে কাজ করতে সক্ষম হওয়া উচিত। এটি একটি সঠিক উপায়ে ওয়াশিং মেশিনকে (washing machine) নিয়ন্ত্রণ করতে সক্ষম হওয়া উচিত, যাতে ব্যবহারকারী হিসাবে আপনি পারফরম্যান্সে খুশি হন।

সুতরাং, কাঞ্জিত কার্যকারিতা হ'ল মূলমন্ত্র, এমন কিছু বাস্তবায়ন যা ডিজাইনের কার্যকারিতা উপলব্ধি করে। তবে এই বাস্তবায়নটি পেতে আপনার কিছু নকশার চ্যালেঞ্জ থাকতে পারে। আমরা বিভিন্ন ডিজাইন মেট্রিকগুলির সাথে কথা বলব এবং আপনাকে সেগুলির কয়েকটি অনুকূলিত (optimise) করতে হতে পারে। পছন্দ করুন, ব্যবহারকারী বলতে পারে যে আমি এমন কিছু চাই যা স্বল্প ব্যয়বহুল, আকারে ছোট, খুব শক্তিশালী এবং খুব জোরালো। তবে আপনি দেখুন যে আপনি ব্যয়টি হ্রাস করার চেষ্টা করছেন, স্বাভাবিকভাবেই performance ও হ্রাস পাবে, কর্তৃরতাও আপস করতে হতে পারে। সুতরাং, তা বিশ্বের সেরা হতে পারে না; এটিকে ট্রেড অফ(trade off) হিসাবে উল্লেখ করি। এগুলি বেশিরভাগ ক্ষেত্রে পারস্পরিক বিরোধপূর্ণ।

আমি একটি ছোট উদাহরণ দিয়েছি যেখানে তিনটি বৃত্ত তিনটি parameter বা ডিজাইনের মেট্রিকগুলি উল্লেখ করে: গতি, ব্যয় এবং গুণমান। ভাল, বেশিরভাগ ডিজাইনে, আমরা এই তিনটিরই উন্নতি করতে চাই, তবে আমি যেমন বলেছিলাম যে এই তিনটি প্রায়শই পরস্পরবিরোধী হয়। আপনি যদি ব্যয়টি হ্রাস করার চেষ্টা করেন তবে গতি এবং মানের উপর আপোষ করতে হবে, ইত্যাদি।

(Refer Slide Time: 04:26)



আসুন আমরা এমবেডেড সিস্টেম ডিজাইনের সাথে সম্মত কিছু সাধারণ নকশার মেট্রিকগুলির বিষয়ে কথা বলি। প্রথম গুরুত্বপূর্ণ parameterটিকে নন রিকারিং ইঞ্জিনিয়ারিং(non-recurring engineering) ব্যয় বলা হয়, এটি এক ধরণের প্রাথমিক ব্যয়।

মনে করুন আপনার কাছে কোনও নকশা ধারণা রয়েছে, আপনি সিস্টেমটি তৈরি করতে যাচ্ছেন। তবে আপনি উত্পাদন প্রক্রিয়া শুরু করার আগে আপনাকে আপনার কারখানায় এমন কিছু মেশিন ইনস্টল (install) করতে হবে যা আপনাকে উৎপাদনে সহায়তা করবে। এবং সেই প্রাথমিক ইনস্টলেশনটিতে(installation) কিছু প্রাথমিক ব্যয় হবে; এটিকেই অপুনরাবৃত্তিয়(non recurring) ব্যয় হিসাবে উল্লেখ করা হয়। এটি শুরুতে একবারে প্রয়োজনীয় হবে। আমাদের যখন সেই কাঠামো প্রস্তুত হয়ে যায়, আপনি যতগুলি চান সিস্টেম তৈরি করতে পারেন; আপনি যতগুলি চান 10, 20, 100, 1000 তৈরি করতে পারেন। non recurring ব্যয় হল এক সময়ের প্রাথমিক খরচ।

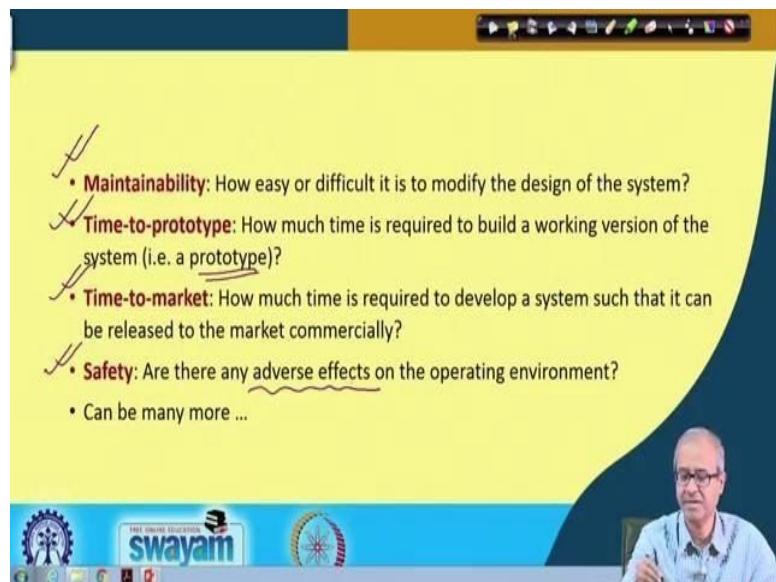
এবং একবার আপনি এই NRE ব্যয়টি আচ্ছাদিত হয়ে গেলে ইউনিট ব্যয়ের বিষয়ে কথা বলতে হবে। আমাদের সেই কাঠামো থাকার পরে, সিস্টেমের প্রতিটি অনুলিপি তৈরির জন্য ব্যয় কী, যা আপনি ইউনিট ব্যয় হিসাবে সংজ্ঞায়িত করেন। ইউনিট ব্যয় কাঁচামাল, শ্রমের ব্যয় এবং এর জন্য আরও বিবেচনা করবে। আকারটি একটি গুরুত্বপূর্ণ parameter যা আমি আগে বলেছিলাম। এছেড়ে বেশিরভাগ সিস্টেমে খুব ছোট ফর্ম ফ্যাক্টরের(form factor) সাথে খাপ খাইয়ে নেওয়া দরকার, যাতে এটি পরিবেশ (environment) এর অভ্যন্তরে সুন্দরভাবে ফিট করতে পারে যার জন্য এটি বোঝানো হচ্ছে।

পারফরম্যান্স আবার প্রয়োগের উপর নির্ভর করে। কিছু প্রয়োগের জন্য আপনি পারফরম্যান্স সম্পর্কে খুব সচেতন নাও হতে পারেন, আপনার খুব কম কম্পিউটেশনাল সংস্করণ, তবে কিছু অ্যাপ্লিকেশন রয়েছে যেখানে পারফরম্যান্স গুরুত্বপূর্ণ। আপনার অবশ্যই পারফরম্যান্স নিশ্চিত হওয়ার

দরকার। বিদ্যুৎ খরচ, এটি গুরুত্বপূর্ণ। আমরা আজ যে সিস্টেমগুলি দেখি, সেগুলির বেশিরভাগগুলি ব্যাটারিতে চালিত হয়। তাদের খুব কম শক্তি গ্রহণ করা প্রয়োজন। ভাল, আমাদের মোবাইল ফোন এব্রেডেড সিস্টেমের খুব ক্লাসিক একটা উদাহরণ।

Flexibility মানে সিস্টেমের কার্যকারিতা পরিবর্তন করার ক্ষমতা। সিস্টেমটি প্রাথমিকভাবে নির্দিষ্ট প্রয়োগের জন্য তৈরি করা হয়েছিল। আপনি কি এটি অন্য কোনও অ্যাপ্লিকেশনের জন্যও ব্যবহার করতে পারেন, তাই এটি কী সংশোধন করা সম্ভব? ঠিক, যদি সেই flexibility থাকে তবে সম্ভবত দ্বিতীয় বিকাশের জন্য আপনার মোট ব্যয় অনেক কম হবে। সুতরাং, flexibility ও একটি বিষয় যা বিবেচনা করা দরকার।

(Refer Slide Time: 07:57)



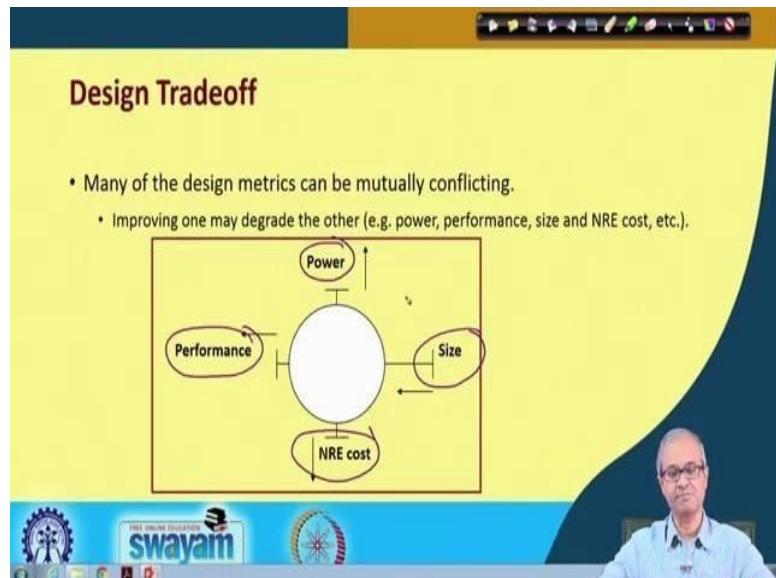
রক্ষণাবেক্ষণ বলছে যে আমি আরও বড় ব্যবস্থার অংশ হিসাবে ইতিমধ্যে একটি এমবেডেড সিস্টেম কিনেছি। বলছি যে আমি একটি এসি মেশিন কিনেছি। আগামীকাল সংস্থাটি বলছে যে আমরা একটি উন্নত এসি মেশিন নিয়ে এসেছি যার একটি ব্লুটুথ ইন্টারফেস (bluetooth interfacing) রয়েছে, যা আপনার মোবাইল ফোন এবং কম্পিউটার এবং ল্যাপটপের সাথে ইন্টারফেস করতে পারে। সুতরাং, আপনি এমনকি মোবাইল ফোনে কিছু অ্যাপ্লিকেশন ইনস্টল করে আপনার ল্যাপটপ থেকে এসি মেশিনটিও নিয়ন্ত্রণ করতে পারেন।

সুতরাং, পুরাণো সিস্টেমের পক্ষে কী এই নতুন প্রযুক্তির সাথে খাপ খাইয়ে নেওয়া সম্ভব? আপনি কি এই নকশাটি সংশোধন করতে পারেন যাতে এই যুক্ত কার্যকারিতাটি সংযুক্ত করা যায়? এটিই আমরা রক্ষণাবেক্ষণের দ্বারা বোঝাতে চাই। তবে অবশ্যই, রক্ষণাবেক্ষণ কেবলমাত্র একটি সীমিত পরিমাণে সম্ভব। আমি যে উদাহরণটি উল্লেখ করেছি কেবল তার জন্য, ব্লুটুথ (bluetooth) থাকার জন্য

আপনার কাছে একটি ক্লটুথ ইন্টারফেস থাকা দরকার। সুতরাং, এটি যদি প্রথম স্থানে না থাকে তবে আপনার কাছে ক্লটুথ থাকতে পারে না।

তারপরে হল প্রোটোটাইপ করার সময়। প্রোটোটাইপ(prototype) নামে পরিচিত সিস্টেমটির প্রথম কার্যকরী সংস্করণটি তৈরি করতে কত সময় প্রয়োজন? প্রথম সংস্করণটি প্রস্তুত হতে কত সময় প্রয়োজন যাতে আপনি পরীক্ষা করতে পারেন? এবং প্রোটোটাইপিংয়ের(prototyping) পরে, সময়ে বাজারে আসে। বাজারে বিক্রি হতে পারে এমন একটি তৈরি পণ্য তৈরি করতে কত সময় প্রয়োজন? ঠিক আছে, প্রোটোটাইপিং এবং সমাপ্ত পণ্য দুটি সম্পূর্ণ ভিন্ন জিনিস। সমাপ্ত পণ্য স্থায়িত্ব, সমাপ্তি, মানের সবকিছু থাকতে পারে। তবে অবশ্যই, অনেক অ্যাপ্লিকেশনের জন্য সুরক্ষা একটি গুরুত্বপূর্ণ সমস্যা। আপনাকে অবশ্যই নিশ্চিত হতে হবে যে সেই সিস্টেমটি ব্যবহারের কারণে পরিবেশের উপর কোনও বিরূপ প্রভাব রয়েছে কিনা। এবং এই জাতীয় আরও অনেক কিছুই থাকতে পারে, আমি কেবল তাদের কয়েকটিকে তালিকাভুক্ত করেছি।

(Refer Slide Time: 10:15)



Design Tradeoff

- Many of the design metrics can be mutually conflicting.
 - Improving one may degrade the other (e.g. power, performance, size and NRE cost, etc.).

Power

Performance

Size

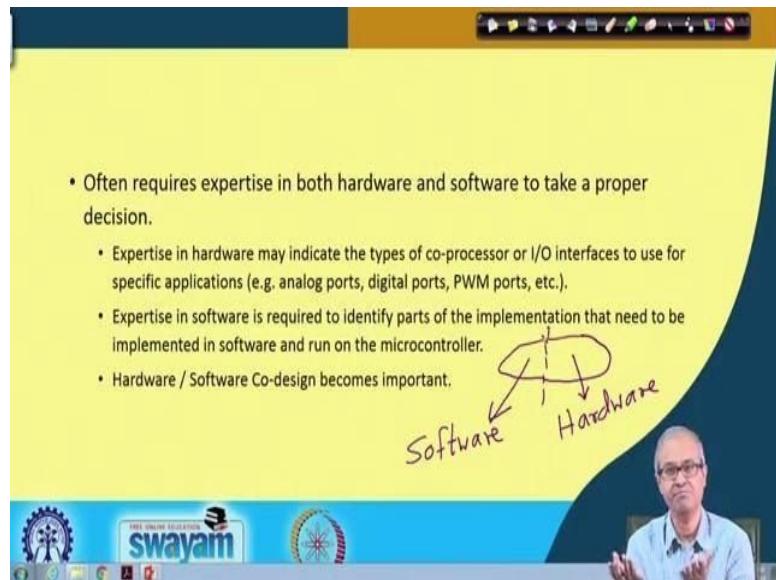
NRE cost

FREE ONLINE EDUCATION
swayam

ডিজাইন ট্রেড অফ (design trade off) সম্পর্কে কথা বলার জন্য, বেশ কয়েকটি নকশার প্রয়োজনীয়তা থাকতে পারে যা পারস্পরিক বিরোধী। এখানে এই চিত্রটিতে আমি এই জাতীয় চারটি মেট্রিক, বিদ্যুত ব্যবহার, পারফরম্যান্স, পুনরাবৃত্তি ব্যয় এবং আকারটি দেখানো হচ্ছে। এখন, এই চারটি পারস্পরিক বিরোধপূর্ণ হতে পারে; আপনি যদি একটি টানেন, অন্যগুলি বিপরীত দিকে টানা হবে, এর অর্থ আপনি যদি কোনওটির উন্নতি করতে চান তবে অন্যরা অবনমিত হতে পারে। সুতরাং, আপনাকে একটি সামগ্রিক দৃষ্টিভঙ্গি নিতে হবে। এটি ডিজাইনারের কাজ। আপনি একবারে সমস্ত কিছু উন্নত করতে পারবেন না।

উদাহরণস্বরূপ, খুব স্লিম ল্যাপটপগুলি তৈরি করা হচ্ছে। ভাল আপনি আশা করেন না যে এই ল্যাপটপগুলি সমস্ত দিক থেকে শক্তিশালী হয়ে উঠবে। এটিকে স্লিম করতে কোথাও কোথাও কিছু আপস করা হয়েছে। এগুলি ডিজাইন ট্রেড অফের (design trade off) উদাহরণ।

(Refer Slide Time: 11:32)



এবং আরেকটি গুরুত্বপূর্ণ বিষয় হ'ল ডিজাইন ট্রেড অফ সম্পর্কে কথা বলা, এন্ডেড সিস্টেমগুলির জন্য আপনার প্রশিক্ষণের জন্য বিভিন্ন ধরণের পেশাদার প্রয়োজন। আপনি প্রচলিত কম্পিউটিং বা প্রতিহ্যবাহী কম্পিউটিং পরিবেশ সম্পর্কে চিন্তা করেন, যেখানে আপনার ইঞ্জিনিয়ারদের একটি সেট রয়েছে, যারা হার্ডওয়্যার বিকাশ করে, পেন্টিয়াম প্রসেসরটি (pentium processor) ইন্টেল(intel) দ্বারা অত্যন্ত দক্ষ হার্ডওয়্যার ইঞ্জিনিয়ারদের একটি সেট দ্বারা তৈরি করা হয়েছিল। আরও কিছু গ্রন্থ রয়েছে যারা সেই কম্পিউটার সিস্টেমে সফটওয়্যার তৈরি করে যারা এই চিপগুলি ব্যবহার করে।

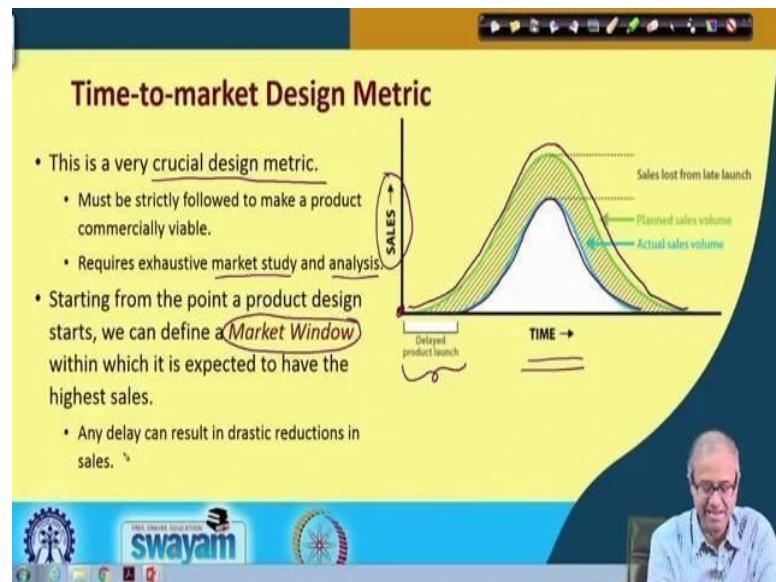
উদাহরণস্বরূপ, মাইক্রোসফ্ট (microsoft)এমন কিছু সফটওয়্যার ইঞ্জিনিয়ার আছেন যারা অপারেটিং সিস্টেমটি বিকশিত করেন, যারা মাইক্রোসফ্ট অফিসের(microsoft office) মতো ব্যাবহারগুলি(utility) বিকাশ করে এবং আরও অনেক কিছু সুতরাং, হার্ডওয়্যার বিশেষজ্ঞ(Hardware expert) আছে, আছে সফটওয়্যার বিশেষজ্ঞ(software expert)। তাদের ভূমিকা কমবেশি একে অপরের থেকে স্বতন্ত্র। হার্ডওয়্যার বিশেষজ্ঞের(hardware expert) সফটওয়্যারটি খুব ভালভাবে জানা উচিত নয় এবং সফটওয়্যার বিশেষজ্ঞদের কেবল হার্ডওয়্যার সম্পর্কে খুব কমই জানেন। তবে এখন আপনি যখন একটি এমবেডেড সিস্টেম ডিজাইন করছেন, আপনি একটি খুব ছোট সিস্টেম সম্পর্কে কথা বলছেন যেখানে হার্ডওয়্যার এবং সফটওয়্যার উভয়ই থাকবে। এবং ডিজাইন ট্রেড অফ (Design trade off) সম্পর্কে কথা বলার জন্য, উভয়ই বলতে হবে।

সুতরাং, এখন আপনার হার্ডওয়্যারটি জানার দক্ষতা থাকা দরকার, কীভাবে এটি ব্যবহার করবেন এবং কীভাবে কার্যকর উপায়ে এটি প্রোগ্রাম করবেন তা স্টেওয়্যার। সুতরাং, হার্ডওয়্যার এবং স্টেওয়্যার উভয়ই দক্ষতার প্রয়োজন এটি আমি এখানে বলতে চেয়েছিলাম। কোন নির্দিষ্ট অ্যাপ্লিকেশনগুলির জন্য আপনার কী ধরণের কোপ্রোসেসর(coprocessor) বা আইও ইন্টারফেসের(I/O interface) প্রয়োজন হবে তা সনাত্ত করতে আপনার হার্ডওয়্যারের দক্ষতার প্রয়োজন হতে পারে, কারণ আপনার বিভিন্ন পছন্দ থাকতে পারে। কিছু অ্যানালগ পোর্ট(Analog port), ডিজিটাল পোর্ট(Digital Port), পালস উইথ মডুলেশন(Pulse Width Modulation) পোর্ট থাকতে পারে তবে কোনটি নির্দিষ্ট অ্যাপ্লিকেশনের জন্য সেরা হবে? সুতরাং, আপনি যদি হার্ডওয়্যার বিশেষজ্ঞ না হন তবে আপনি এখানে সত্যিই কোনও ভাল সিদ্ধান্ত নিতে পারবেন না।

একইভাবে, আপনারও স্টেওয়্যার বিষয়ে দক্ষতা থাকা দরকার, কারণ আপনাকে সিদ্ধান্ত নিতে হবে যে স্টেওয়্যারে আপনাকে প্রয়োগের কোন অংশগুলি প্রয়োগ করা দরকার এবং কোন অংশটি ইতিমধ্যে কিছু হার্ডওয়্যার দ্বারা প্রয়োগ করা হয়েছে। আপনি এখন হার্ডওয়্যার সফটওয়্যার কো-ডিজাইন নামে পরিচিত এমন কিছু সম্পর্কে কথা বলছেন, যার অর্থ আপনি উভয় হার্ডওয়্যার এবং স্টেওয়্যার সাবসিস্টেমটিকে (subsystem) একসাথে ডিজাইন করছেন।

ধরুন আপনার ডিজাইন করার জন্য একটি বৃহত সিস্টেম রয়েছে। আপনাকে সিদ্ধান্ত নিতে হবে আমি ভাল করেই একটি সীমাবদ্ধতা তৈরি করেছি, এর একটি অংশ স্টেওয়্যারে প্রয়োগ করা হবে, এটি একটি মাইক্রোকন্ট্রোলারের প্রোগ্রাম(microcontroller program) হিসাবে চলবে এবং অন্য অংশটি কিছু বিশেষায়িত হার্ডওয়্যার সার্কিট(special hardware circuit) দ্বারা প্রয়োগ করা হবে। সুতরাং, যেখানে এই সীমানাটি হবে ডিজাইনের ট্রেড অফের বিষয়, ডিজাইনার সিদ্ধান্ত নেবেন কীভাবে এই বিন্দুযুক্ত লাইনটি স্থানান্তর করবেন, যাতে কাঞ্চিত পারফরম্যান্স এবং অন্যান্য মানদণ্ড সর্বোত্তম সম্ভাব্য উপায়ে সন্তুষ্ট থাকে।

(Refer Slide Time: 15:29)



এখানে একটি গুরুত্বপূর্ণ বিষয় রয়েছে, সময় ডিজিটাল মেট্রিকের জন্য। আসুন এটি ব্যাখ্যা করার চেষ্টা করা যাক। কোনও সংস্থা দেখুন যখনই এটি কিছু পণ্য উত্পাদন করে চূড়ান্ত লক্ষ্য হ'ল এর থেকে কিছু লাভ অর্জন করা। সুতরাং, বিপণন যে কোনও সংস্থার জন্য সবচেয়ে গুরুত্বপূর্ণ সমস্যা। পণ্যের গুণমান গৌণ হয়ে যায়। কখনও কখনও অনেক সংস্থা পণ্যটি বাজারে আনার জন্য পণ্যের মানের সাথে আপস করে, যাতে তারা এ থেকে আরও বেশি লাভ অর্জন করতে পারে, বাজারে যাওয়ার সময়টি একটি অত্যন্ত গুরুত্বপূর্ণ ডিজাইন মেট্রিক।

পণ্যটি বাণিজ্যিকভাবে কার্যকর করার জন্য এটি অবশ্যই কঠোরভাবে অনুসরণ করা উচিত। আপনি যদি কোনও পণ্য চালু করতে বিলম্ব করেন তবে আপনার বড় ক্ষতি হতে পারে। হতে পারে আপনার কিছু প্রতিযোগী ইতিমধ্যে একটি অনুরূপ পণ্যের সাথে বাজারে এসে পড়েছে এবং লোকেরা প্রাকৃতিকভাবে তাদের কাছ থেকে কিনবে আপনার কাছ থেকে নয়। এবং এর জন্য খুব সতর্কতার সাথে বাজার গবেষণা এবং বিশ্লেষণ প্রয়োজন। এখন, আমরা এটি এর মতো একটি চির দিয়ে চিত্রিত করছি। এখানে এক্ষেত্রে অক্ষগুলি সময় দেখায় এবং y- অক্ষগুলি বিক্রয় দেখায়।

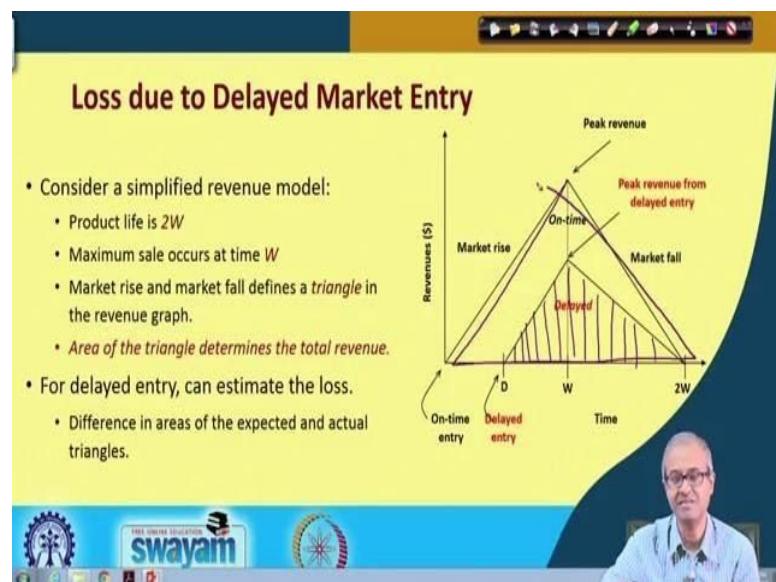
ধরুন আপনার উত্সটি সেই পয়েন্টটি নির্দেশ করে যেখানে আপনি বাজারে পণ্যটি নিয়ে আসবেন বলে আশা করা হচ্ছে। সুতরাং, যখন কোনও পণ্য বাজারে আসে তখন কী হয়, ভাল ব্যবহারকারী বা গ্রাহকদের পণ্য সম্পর্কে কিছুটা জানতে কিছু সময় প্রয়োজন। সময়ের সাথে ধীরে ধীরে বিক্রয় বাড়ে। এবং একটি পয়েন্ট থাকবে যেখানে এই বিক্রয় পরিপূর্ণ হবে এবং সেই বিন্দু ছাড়িয়ে আবার এই বিক্রয়টি নিচে নামা শুরু করবে। এবং একটি পয়েন্ট থাকবে যেখানে পণ্যটি অপ্রচলিত হয়ে উঠবে, কেউই এখন এটি কিনছে না। এটি একটি সাধারণ বক্ররেখা(curve), বিক্রয় বনাম সময়।

এবং ধরুন পণ্য লক্ষে দেরি হচ্ছে। সুতরাং, এই বাঁকটির পরিবর্তে, আপনি এখন এই অভ্যন্তরীণ বক্ররেখা অনুসরণ করছেন। প্রাথমিক দেরি হয়েছিল। সুতরাং, আপনি দেখতে পাবেন যে বাঁকটি

আবার একই রকম হবে, তবে আপনি যে পিক বা সর্বাধিক বিক্রয় অর্জন করতে পারেন তা হ্রাস পাচ্ছে। এই বক্ররেখার নীচে থাকা মোট অঞ্চলটি আপনার মোট রাজস্বকে বোঝায় যে আপনি কতটা মোট বিক্রয় করতে পেরেছেন। বক্ররেখার অধীনে মোট অঞ্চলটি সেই সময়ের মধ্যে আপনার মোট বিক্রয়কে নির্দেশ করবে। বক্ররেখার নীচে এই বক্ররেখাটি মার্কেট উইンドো (market window) হিসাবে সংজ্ঞায়িত করা হয়। যে কোনও বিলম্বের ফলে বিক্রয় কমে যেতে পারে।

আসুন আমরা কিছুটা আসন্ন (Approximation) সহ একটি সাধারণ গণনা করি। ভালভাবে বক্ররেখাটি দেখতে এটির মতো, তবে প্রথম অনুমানের জন্য আমরা ধরে নিতে পারি যে এটি সরল রেখা, আপনি ধরে নিতে পারেন যে এটি সরল রেখা, এটিও একটি সরল রেখা।

(Refer Slide Time: 19:10)



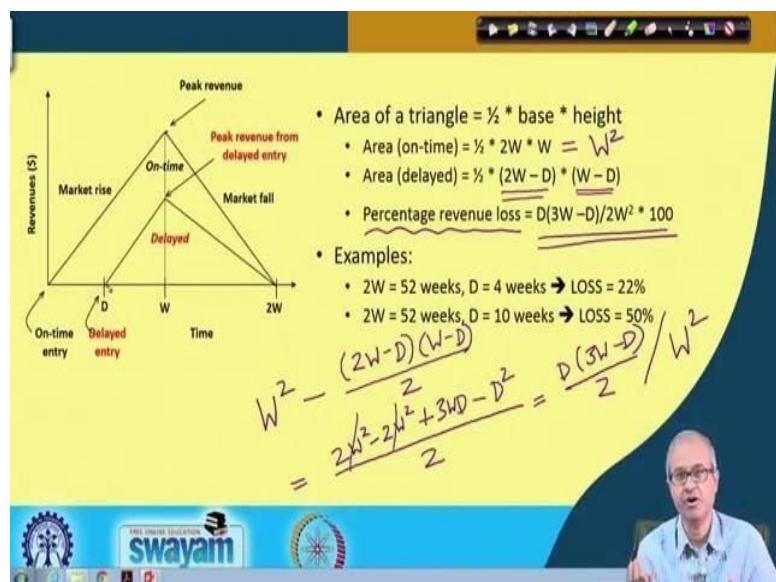
এই অনুমানের সাথে আসুন আমরা বাজারে প্রবেশ দেরি হলে আমাদের কতটা ক্ষতি হতে পারে বলে অনুমান করার চেষ্টা করা যাক। আমরা সরলীকৃত রাজস্বের মডেল হিসাবে বিবেচনা করেছি, আপনি আপনার নির্দিষ্ট ক্ষেত্রে আরও বাস্তবসম্মত যা বোঝাতে পারেন তা বোঝাতে আপনি খুব সহজেই এটি পরিবর্তন করতে পারেন। এখানে আমরা কী ধরে নিচ্ছি যে এই গ্রাফটি দেখায় যে পণ্য জীবন $2W$ । সূতরাং, 0 পর্যন্ত $2W$ পর্যন্ত, এটি আপনার মোট পণ্য জীবন। $2W$ এর পরে কেউ আপনার পণ্য কিনবে না। এবং আমাদের ধরে নেওয়া যাক সর্বাধিক বিক্রয় এর মাঝামাঝি সময়ে ঘটে, W সময়ে বিক্রয় সর্বাধিক হয়ে যায়। বাজার বৃদ্ধি এবং বাজার পতন আমরা এটির সরল রেখাগুলির প্রায় অনুমান করি যার অর্থ আমরা একটি গ্রিভুজাকার কাঠামো সংজ্ঞায়িত করি।

এবং যেমনটি আমি উল্লেখ করেছি যে এই গ্রিভুজের অধীনে মোট ক্ষেত্রটি আপনার উৎপন্ন মোট আয় নির্ধারণ করবে। এখন, যদি আমাদের একটি গ্রিভুজ থাকে তবে আপনি কীভাবে গ্রিভুজটির ক্ষেত্রফল

পরিমাপ করবেন তা জানেন। এখন, দেরিতে প্রবেশের জন্য যেমনটি আমি পূর্বের চিত্রটিতে বলেছিলাম, ধৰন এগুলি দিয়ে অনেক বেশি বিলম্ব হয়েছে। সুতরাং, এখন, আপনার ত্রিভুজটি এই বিন্দু D থেকে এখন থেকে শুরু হবে, তবে এটি আবার একই পয়েন্টে পোঁছে যাবে কারণ বাজারে গতিশীলতার উপর নির্ভর করে সেই পণ্টটির নির্দিষ্ট পয়েন্টের আগ্রহের বাইরে চলে যেতে শুরু করবে।

সুতরাং, পতন একই পয়েন্টে ঘটতে শুরু করবে W। সুতরাং, এরপরে আবার এটি $2W$ পর্যন্ত অব্যাহত থাকবে যার বাইরে আর সেই পণ্টটির কোনও আগ্রহ থাকবে না, তবে আপনার প্রাথমিক বিলম্বটি আসলে বাম প্রাণ্টি বা বাম প্রাণ্টকে স্থানান্তরিত করছে আপনার ত্রিভুজের ডানদিকে, এটি ঘটছে। প্রত্যাশিত এবং আসল ত্রিভুজগুলির ক্ষেত্রে পার্থক্য থাকবে। আপনার আসল ত্রিভুজটির মতো অঞ্চলটি কেবলমাত্র এটির অনেক বেশি হবে তবে আপনার প্রত্যাশিত অঞ্চলটি ছিল পুরো বৃহত্তর ত্রিভুজের ক্ষেত্র। বিলম্বিত প্রবর্তনের কারণে এ ক্ষেত্রে এটি স্বাভাবিকভাবেই অনেক কম।

(Refer Slide Time: 21:46)



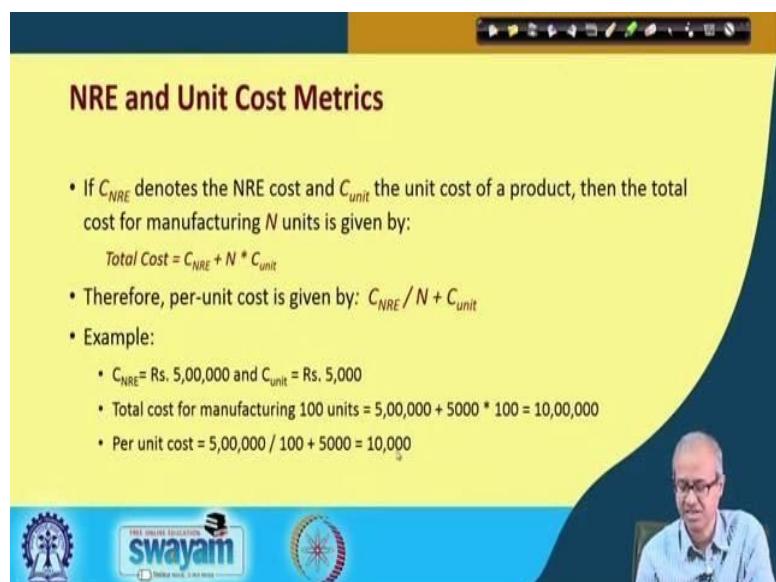
আসুন একটি গণনা করা যাক। আমরা বাম দিকে একই প্লট দেখাই। আমরা জানি যে একটি ত্রিভুজের ক্ষেত্রফল অর্ধেকের উচ্চতা হিসাবে সংজ্ঞায়িত হয়। অন-টাইমের(ON time) ক্ষেত্রে, এর অর্থ, বৃহত্তর ত্রিভুজ অঞ্চলটি (larger triangle area) বেসের(Base) অর্ধেক হবে, যা $2W$ এবং উচ্চতা W আমাদের ধরে নেওয়া যাক এটি একটি সমবাহু ত্রিভুজ, এটি 45 ডিগ্রি ঢালে বেড়ে চলেছে। সুতরাং, উচ্চতা ধরে নেওয়া যাক এটিও W । অঞ্চলটি W^2 হয়।

দেরি হওয়ার জন্য আপনার বেসটি $2W - D$ হয় এবং আপনার উচ্চতা যেমন $W - D$ হয়ে যায় ততই বিলম্ব হওয়ায়। ক্ষেত্রের গণনা দেখানো হয়েছে।

শতাংশের রাজস্ব ক্ষতির গণনাও দেখানো হয়।

উদাহরণ হিসাবে ধরুন, আপনার উইন্ডোর আকারটি আমি বছর = 52 সপ্তাহ। যদি আপনার বিলম্ব 4 সপ্তাহ হয় তবে আপনার ক্ষতি 22% হয়ে যায়। তবে আপনি যদি 10 সপ্তাহ দেরি করেন তবে আপনার ক্ষতি 50% এর মতো বড় হয়ে যায়। সুতরাং আপনি দেখতে পাচ্ছেন যে আপনার ক্ষতি খুব দ্রুত বৃদ্ধি পায় এবং এই বিলম্ব কোনও পণ্যের আর্থিক সাবলীলতার জন্য খুব ব্যয়বহুল হয়ে যায়। এটি একটি খুব গুরুত্বপূর্ণ বিবেচনা।

(Refer Slide Time: 25:28)



NRE and Unit Cost Metrics

- If C_{NRE} denotes the NRE cost and C_{unit} the unit cost of a product, then the total cost for manufacturing N units is given by:
$$\text{Total Cost} = C_{NRE} + N * C_{unit}$$
- Therefore, per-unit cost is given by: $C_{NRE} / N + C_{unit}$
- Example:
 - $C_{NRE} = \text{Rs. } 5,00,000$ and $C_{unit} = \text{Rs. } 5,000$
 - Total cost for manufacturing 100 units = $5,00,000 + 5000 * 100 = 10,00,000$
 - Per unit cost = $5,00,000 / 100 + 5000 = 10,000$

এখন, NRE এবং ইউনিট ব্যয় নিয়ে কথা বলছি, যদি C_{NRE} অ-পুনরাবৃত্ত প্রকৌশল ব্যয়কে বোঝায় এবং C_{unit} ইউনিট ব্যয়কে চিহ্নিত করে। তারপরে পণ্যের এন ইউনিট তৈরির জন্য মোট ব্যয় হবে $C_{NRE} + N * C_{unit}$ । এখন, আপনি যদি প্রতি ইউনিটের জন্য আমাদের প্রতি কত ইউনিট, প্রতি ইউনিট খরচ নিয়েছে তা গণনা করার চেষ্টা করেন, আপনি মোট ব্যয়টিকে N দ্বারা ভাগ করে নিন, সুতরাং এটি $C_{NRE} / N + C_{unit}$ ।

আপনার NRE যদি খরচ হয় পাঁচ লক্ষ টাকা এবং ইউনিট ব্যয় Rs. 5000, তারপরে 100 ইউনিট উৎপাদন করার জন্য মোট ব্যয় হয় ১০ লক্ষ টাকা। সুতরাং, প্রতি ইউনিট ব্যয় হবে Rs. 10000

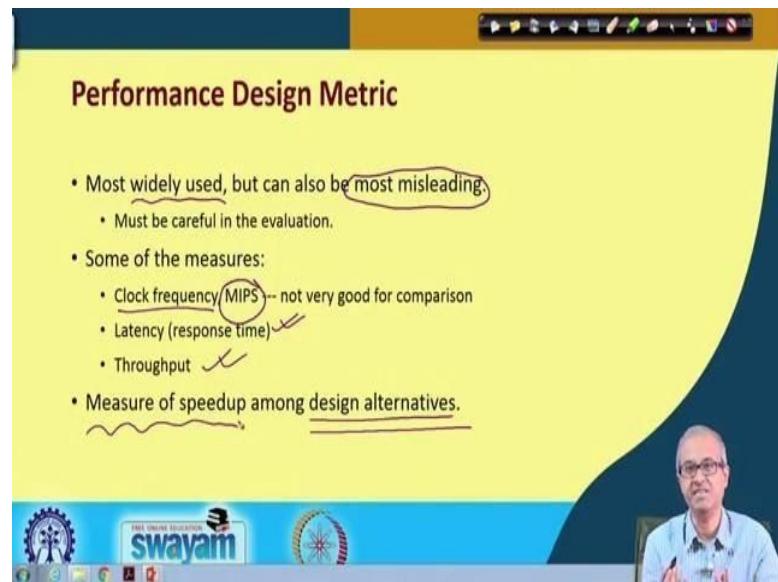
(Refer Slide Time: 27:04)

- We can compare technologies by cost:
 - Choice A: $C_{NRE} = \text{Rs. 20,000}$, $C_{unit} = \text{Rs. 8,000}$ ✓
 - Choice B: $C_{NRE} = \text{Rs. 4,00,000}$, $C_{unit} = \text{Rs. 3,000}$ ✓
 - Choice C: $C_{NRE} = \text{Rs. 10,00,000}$, $C_{unit} = \text{Rs. 8,000}$ ✓
- Of course, time-to-market cost must also be considered.

একটি খুব সাধারণ গণনা আমি কেবল চিত্রিত করছি। ধরুন আপনার কয়েকটি পছন্দ আছে; এমন কোনও পণ্য উত্পাদন করতে চান যে আপনি দেখতে পান যে বিভিন্ন ধরণের সরঞ্জাম আপনি কিনে নিতে পারেন them এবং উপকরণগুলির প্রাথমিক ব্যয়ও বেশ আলাদা। ধরা যাক আপনার প্রথম পছন্দের জন্য তিনটি পছন্দ রয়েছে নন-পুনরাবৃত্তি ব্যয় 20000, দ্বিতীয়টি 4 লক্ষ, তৃতীয়টি 10 লক্ষ। এবং তিনটি ক্ষেত্রে ইউনিট ব্যয় এই এবং এটি এখান থেকে আসছে।

সুতরাং, আপনি এন এর কিছু নির্দিষ্ট মানের জন্য প্রতি ইউনিট ব্যয় গণনা করতে পারেন; আমি আপনার জন্য অনুশীলন হিসাবে এটি ছেড়ে। ধরুন আমি বলি আমি 1000 ইউনিট পণ্য তৈরি করতে চাই, তবে এই সূত্রটি ব্যবহার করে আপনি চয়েস এ, চয়েস বি এবং চয়েস সি(choice A, choice B, and choice C) এর জন্য প্রতি ইউনিট ব্যয় গণনা করুন; তাদের মধ্যে কোনটি ভাল। কেবল আপনাকে বাজারের দামের সময়কেও বিবেচনা করতে হবে না, কারণ এটির অর্থ ব্যয়ের দিক থেকে একটি পছন্দ ভাল হতে পারে তবে আপনি সেই মেশিনটি শুরু করতে দীর্ঘ সময় নিজেন, কারণ ইনস্টলিং(installing), প্রশিক্ষণ এবং কেবল সেই মেশিনটিই ব্যবহার করা হচ্ছে আরও অনেক সময় প্রয়োজন যা আপনাকেও বিবেচনা করতে হবে। সুতরাং কেবল ব্যয়ই নয়, বাজারজাত করারও সময়।

(Refer Slide Time: 28:46)



এবং এর সাথে আপনার কাছে কিছু পারফরম্যান্স ডিজাইনের মেট্রিক রয়েছে যা বিবেচনা করা দরকার। সাধারণত, এগুলি প্রসেসরের পারফরম্যান্স মূল্যায়নের জন্য প্রতিহিগতভাবে(traditionally) ব্যবহৃত হয়। এম্বেডেড সিস্টেমের জন্য আপনি একই বিবেচনা ব্যবহার করতে চেষ্টা করতে পারেন তবে কয়েকটি জিনিস আপনার মনে রাখা দরকার। এই পারফরম্যান্স ডিজাইনের মেট্রিকগুলি আপনাকে জানায় যে কোন প্রসেসর কিছু ক্ষেত্রে অন্যের চেয়ে দ্রুত faster এগুলি বেশ বিস্তৃতভাবে ব্যবহৃত হয় তবে এটিও সত্য যে আপনি যদি এই মেট্রিকগুলির অর্থ বুঝতে না পারেন তবে সেগুলি সবচেয়ে বিভ্রান্তিকর হতে পারে।

বেশিরভাগ কম্পিউটার নির্মাতারা বেঞ্চমার্কিং(benchmarking) এবং গতির প্রতি কিছু পরিসংখ্যান উদ্ভৃত করে গ্রাহকদের বিভ্রান্ত করার চেষ্টা করে, যদি আপনি আরও গভীর খনন করেন এবং বুঝতে চেষ্টা করেন, তবে আপনি প্রকৃতপক্ষে অনুভব করবেন যে আপনার নির্দিষ্ট পরিবেশের জন্য সেই সংখ্যাগুলি কোনও অর্থহীন নয়। আপনার অন্য কিছু প্রয়োজন হতে পারে যা আপনার আবেদনের জন্য সবচেয়ে উপযুক্ত হবে। সুতরাং, আপনাকে অবশ্যই মূল্যায়নে খুব সতর্কতা অবলম্বন করা উচিত।

আবার ব্যবহৃত কিছু সাধারণ ব্যবস্থাগুলি প্রদর্শিত হয়। ক্লক ক্লিকোয়েন্সি(clock frequency), ভাল আপনি দেখতে পাচ্ছেন যে কেউ 1 গিগাহার্জ (1 GHz), 2 গিগাহার্জ (2GHz), 3 গিগাহার্জ(3 GHz) বলছে দ্রুত ক্লক ক্লিকোয়েন্সি মানে কি দ্রুত কম্পিউটার? অগত্যা। আরও অনেকগুলি জিনিস রয়েছে যা একটি কম্পিউটার সিস্টেমের আসল কর্মসূচি নির্ধারণ করে। MIPS এর মতো কিছু ব্যবস্থা রয়েছে; এটিও কিছু বোঝায় না। এগুলি কী ধরণের নির্দেশনা? সহজ নির্দেশনা বা জটিল নির্দেশাবলী? আপনার MIPS-র চিত্র তাদের মধ্যে মারাত্মকভাবে পরিবর্তিত হবে। সুতরাং, এটি কোনও ন্যায্য পরিমাপ দেয় না। কেবলমাত্র কিছু নির্দিষ্ট অ্যালগ্রিদমে সম্পর্কিত, আপনি আমার অ্যালগ্রিদমে চালাতে কত সময় নিচ্ছে তা পরিমাপ হওয়া উচিত। এটি এমন কিছু যা আপনার মূল্যায়নে সর্বাধিক উপকারী হবে।

বিলম্বিত প্রতিক্রিয়া সময়: আপনি আউটপুট ফিরে পেতে কত সময় পরে আপনি একটি ইনপুট দিতে? এটি অবশ্যই অনেক রিয়েল টাইম অ্যাপ্লিকেশনগুলির জন্য একটি ভাল পরিমাপ হতে পারে। এছেড়ে সিস্টেমের জন্য থ্রুপুট খুব ভাল নাও হতে পারে। এর অর্থ হ'ল সংখ্যার গণনা যা ইউনিট সময় অনুসারে বহন করা যায়। সাধারণত, আমরা প্রচলিত কম্পিউটার সিস্টেমগুলির জন্য থ্রুপুট সম্পর্কে কথা বলি, তবে একটি এমবেডেড সিস্টেমের জন্য যা খুব নির্দিষ্ট অ্যাপ্লিকেশনটির জন্য ব্যবহৃত হয়, আমরা সাধারণত থ্রুপুট(throughput) সম্পর্কে কথা বলি না।

এবং যদি আপনার কাছে বেশ কয়েকটি ডিজাইনের বিকল্প রয়েছে, আপনার গতির তুলনা করার একটি উপায় আপনার উচিত। এই মূল্যায়নগুলি সহজ নয়। সর্বোত্তম উপায় হ'ল আপনি যে আসল কম্পিউটিং মেশিনে চালাতে চান সেই অ্যাপ্লিকেশনটি চালানো এবং এটি দেখতে যে কতটা সময় লাগে তা। নির্মাতারা যাই বলুক না কেন আপনি এগুলিকে এক চিমটি লবণের সাথে নিয়ে যান, তাদের 100% বিশ্বাস করবেন না।

এটির সাথে আমরা এই বক্তৃতাটির শেষে এসেছি যেখানে আমরা এছেড়ে সিস্টেমের নকশায় প্রাসঙ্গিক কিছু ডিজাইন মেট্রিক এবং ব্যয় এবং উপর্যুক্ত ক্ষেত্রে কিছু বিষয় সম্পর্কে বললাম।

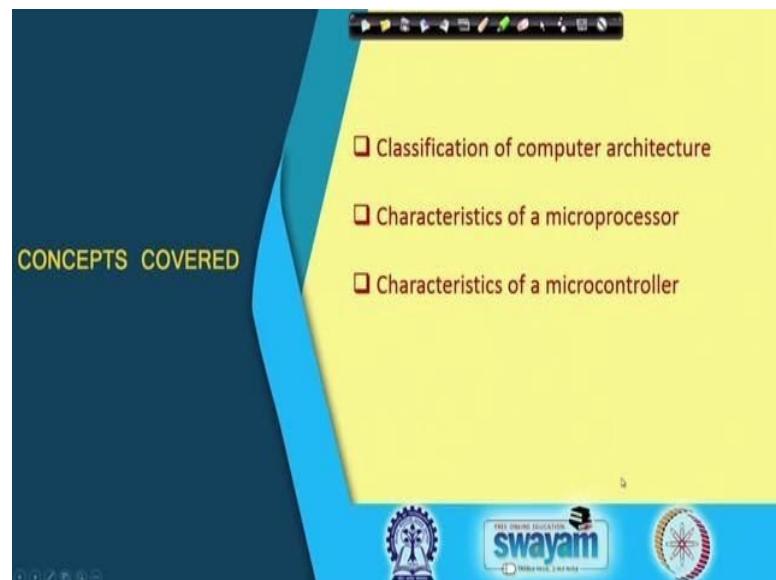
ধন্যবাদ।

Embedded System Design with Arm
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 3
Microprocessors and Microcontrollers

এই বক্তৃতাতে আমরা মাইক্রোপ্রসেসর (Microprocessor), মাইক্রোকম্পিউটার (microcomputer) এবং মাইক্রোকন্ট্রোলারগুলির (microcontroller) কয়েকটি প্রাথমিক ধারণা সম্পর্কে কথা বলব কারণ এটি যে কোনও এন্ডেড সিস্টেমের পিছনে থাকা মন্ত্রিক বা প্রসেসিং শক্তি। এইগুলির মধ্যে প্রধান পার্থক্যগুলি কী তা জানা দরকার।

(Refer Slide Time: 00:49)

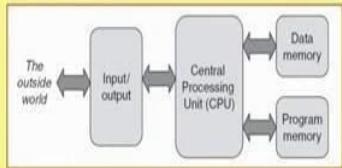


আমরা কম্পিউটার আর্কিটেকচারের (Computer Architecture) কিছু শ্রেণিবিন্যাসের কথা বলব, তারপরে মাইক্রোপ্রসেসর এবং মাইক্রোকন্ট্রোলারগুলির প্রধান বৈশিষ্ট্য উল্লেখ করবো।

(Refer Slide Time: 01:06)

Basic Operation of a Computing System

- The central processing unit (CPU) carries out all computations.
 - Fetches instructions from the program memory and executes it; may require access to data in data memory.
- The input/output block provides interface with the outside world.
 - Allows users to interact with the computing system, and also observe the output results.



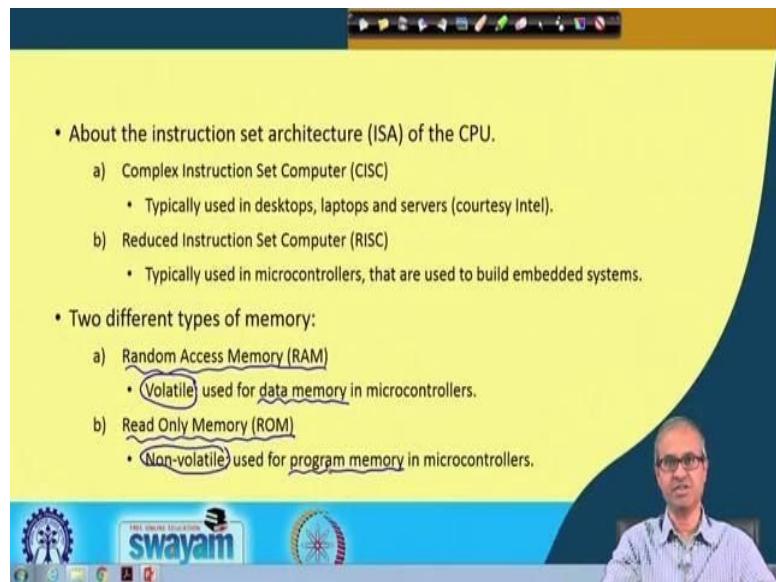
The diagram illustrates the basic operation of a computing system. It shows a flow of data between the outside world, an input/output block, a central processing unit (CPU), and two types of memory: data memory and program memory. The outside world interacts with the input/output block. The input/output block interacts with both the CPU and the data memory. The CPU interacts with both the data memory and the program memory. The program memory also interacts with the CPU.

একটি কম্পিউটার সিস্টেম কীভাবে মূলত কাজ করে তা দিয়ে বক্তব্য শুরু করি। একটি কম্পিউটার সিস্টেমের ব্যাপারে, আপনি জানেন যে CPU নামে একটি কিছু রয়েছে যা সমস্ত গণনা করে। আপনি এই স্কিম্যাটিক ডায়াগ্রাম(Schematic Diagram) দেখতে পাচ্ছেন, CPU মাঝখানে বসে আছে এবং কিছু memory রয়েছে যেখানে প্রোগ্রাম (Program) এবং ডেটা(Data) সংরক্ষণ করা হয়। CPU কী করে? এটি প্রোগ্রাম memory থেকে একের পর এক নির্দেশনা আনে, এগুলি সম্পাদন করে এবং কার্যকর করার সময় এটি ডেটা memory এর মধ্যে কিছু ডেটা স্থানান্তর করতে পারে, কোনও ডেটা পড়ে বা ডেটা memoryতে ফিরে ডেটা লেখা যেতে পারে।

প্রসেসিংয়ের পাশাপাশি, কম্পিউটার সিস্টেমটি প্রায়শই বাইরের বিশ্বের সাথে ইন্টারফেস বা যোগাযোগ করা প্রয়োজন। বাইরের পৃথিবী সাধারণত ব্যবহারকারী, যিনি একটি ডেস্কটপ বা ল্যাপটপের মতো প্রচলিত সিস্টেমগুলির জন্য একটি কম্পিউটার সিস্টেম ব্যবহার করছেন। IO সাবসিস্টেম ব্যবহারকারীদের কম্পিউটিং সিস্টেমের সাথে পারস্পরিক ক্রিয়া(Interact) করার অনুমতি দেয়।

বাইরের পৃথিবী সর্বদা একজন হতে পারে না, এটি পরিবেশ হতে পারে, এটি কিছু তাপমাত্রা, চাপ, আর্দ্রতা - পরিবেশের কিছু parameter অনুভব করে এবং সেগুলি হ'ল ইনপুট। এবং একইভাবে এটি কিছু সংশোধনমূলক পদক্ষেপ নিতে পারে যেমন এটি একটি হিটার চালু করতে পারে, compressor চালু করতে পারে, এবং আরও; এগুলি আউটপুট ডিভাইস।

(Refer Slide Time: 03:08)



instruction সেট আর্কিটেকচার সম্পর্কে কথা বলা এটি এমন এক উপায় যা আপনি কম্পিউটার সিস্টেমগুলিকে শ্রেণিবদ্ধ করতে পারেন। instruction সেটগুলির ক্ষেত্রে বিস্তৃত শ্রেণিবিন্যাসের উপর নির্ভর করে CPU কোন ধরণের নির্দেশনা(instruction) কার্যকর করতে সক্ষম? আমরা কম্পিউটারগুলিকে CISC আর্কিটেকচার বা RISC আর্কিটেকচার হিসাবে শ্রেণিবদ্ধ করতে পারি। কমপ্লেক্স ইন্ট্রাকশন সেট কম্পিউটার, এর সংক্ষিপ্ত রূপ সিআইএসসি। এই জাতীয় CISC আর্কিটেকচারের সবচেয়ে সাধারণ উদাহরণ হ'ল প্রসেসরের ইন্টেল শ্রেণি(Intel class of Processor) যা ডেস্কটপ, ল্যাপটপ এবং সার্ভারের বাজারে প্রাধান্য দেয়।

প্রসেসরের 90% এরও বেশি ইন্টেল বা ইন্টেলের(Intel) তৈরি কিছু ক্লোন দ্বারা তৈরি করা হয়। এগুলিকে CISC বলা হয়, প্রচুর flexibility এবং বৈশিষ্ট্য সহ নির্দেশটি মোটামুটি জটিল হতে পারে।

reduced instruction set computers বা RISC নামে আরও একটি দর্শন আছে। এখানে instruction সেটটি খুব সহজ করে দেওয়া হয়েছে, এবং তাই হার্ডওয়্যারে কম্পিউটার সিস্টেমটি প্রয়োগ করা অনেক সহজ। কিছু দিক থেকে instruction কার্যকর করার ক্ষেত্রে এটি আরও কার্যকর। আজ আমরা যে মাইক্রোকন্ট্রোলারগুলি দেখি তারা বেশিরভাগই RISC আর্কিটেকচারের ভিত্তিতে, কারণ মূলত এই কারণেই, তাদের কার্যকর করা সহজ। কেবল মাইক্রোকন্ট্রোলারই নয়, কিছু স্মার্ট বেশিরভাগ আধুনিক প্রসেসরগুলি instruction কার্যকরকরণের RISC দর্শনের উপর ভিত্তি করে কারণ এটি নির্দেশনা execution ইউনিটকে আরও দৃঢ় করে তোলে।

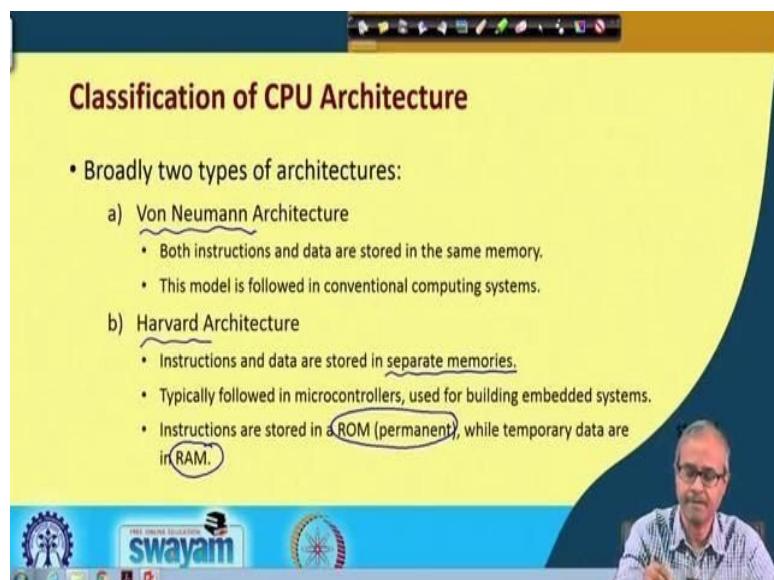
সুতরাং, এমনকি আমি যেসব ইন্টেল প্রসেসরগুলি নিয়ে কথা বলছি সেগুলি CISC ভিত্তিক; অভ্যন্তরীণভাবে এই জটিল instruction সরল instruction বা মাইক্রো(Micro) নির্দেশাবলীতে বিভক্ত হয়ে যায়, এগুলি দেখতে আরও একটি RISC নির্দেশাবলীর মতো দেখা যায়, যা নিয়ামক ব্যবহার করে দক্ষতার সাথে সম্পাদন করা হয়। memory সিস্টেমের বিষয়ে কথা বলতে বলতে এখানে দুটি ধরণের

memory থাকতে পারে; যার মধ্যে এলোমেলো random access memory আছে যেখানে উভয়ই পড়তে এবং লিখতে পারা যায় এবং অন্যটি কেবল তখনই memory যখন আপনি স্থায়ীভাবে কিছু সংস্করণ করেন আপনি কেবল সেগুলি থেকে পড়তে পারবেন।

RAM সাধারণত volatile, volatile অর্থ তারা শক্তিটি চালু হওয়ার সময় কেবলমাত্র মানগুলি বজায় রাখে, আপনি বিদ্যুৎ সুষৃষ্টি অফ করার সাথে সাথে ডেটা নষ্ট হয়ে যায়। মাইক্রোকন্ট্রোলার জন্য RAM গুলি সাধারণত ডেটা মেমরিতে ডেটা সংস্করণ করতে ব্যবহৃত হয়, তবে আপনি যখন প্রোগ্রাম memory-র কথা বলছেন সেই স্থানটি আপনি যেখানে কোনও প্রোগ্রাম সংরক্ষণ করছেন সেখানে এই memory সাধারণত অস্থিতিশীল (non volatile); যার অর্থ এটি একটি ROM, কারণ আপনি একবার বিদ্যুৎ বন্ধ করলেও প্রোগ্রামটি নষ্ট হবে না তা এটি সংরক্ষণ করে।

বেশিরভাগ মাইক্রোকন্ট্রোলারগুলিতে প্রোগ্রাম memory এর একটি ক্ষেত্র থাকে যা ROM ব্যবহার করে প্রয়োগ করা হয়। তবে আজকাল এমন কিছু মাইক্রোকন্ট্রোলার রয়েছে যা কিছু স্ল্যাশ memory এর মতো বিকল্প প্রযুক্তি ব্যবহার করে, এটি non volatile যেখানে আপনি কিছু প্রোগ্রাম সংস্করণ করতে পারেন, আপনি চাইলে প্রোগ্রামটিও পরিবর্তন করতে পারেন তবে আপনি যদি বিদ্যুৎ বন্ধ করে দেন তবে প্রোগ্রামটি ধ্বংস হয় না।

(Refer Slide Time: 07:28)



Classification of CPU Architecture

- Broadly two types of architectures:

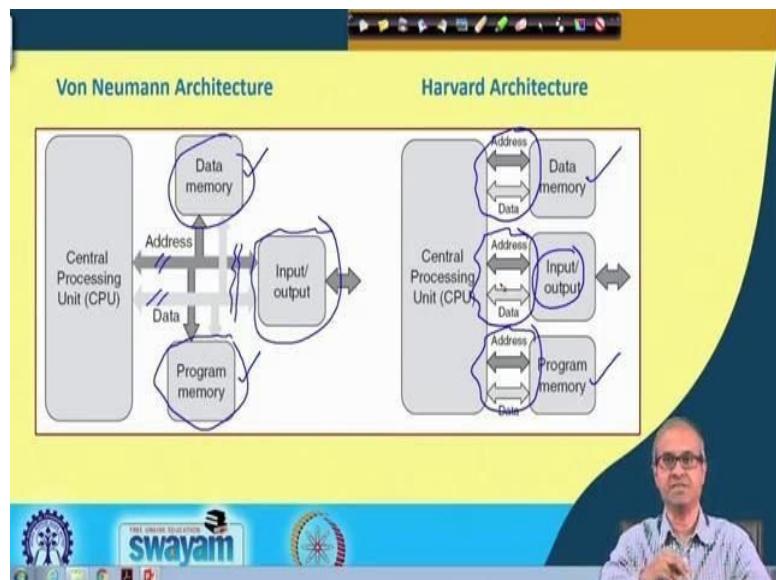
- Von Neumann Architecture
 - Both instructions and data are stored in the same memory.
 - This model is followed in conventional computing systems.
- Harvard Architecture
 - Instructions and data are stored in separate memories.
 - Typically followed in microcontrollers, used for building embedded systems.
 - Instructions are stored in a ROM (permanent), while temporary data are in RAM.

CPU আর্কিটেকচারের ক্ষেত্রে আমরা এখন পর্যন্ত instruction সেট আর্কিটেকচারের কথা বলছি, এখন এখানে CPU কীভাবে কাজ করে সে সম্পর্কে এখানে একটি বিস্তৃত শ্রেণিবিন্যাস রয়েছে। আমরা ভন নিউম্যান (Von Neumann) এবং হার্ড্রড আর্কিটেকচারের (Harvard architecture) হার্ড্রড শ্রেণির কথা বলি। মূল পার্থক্যটি হ'ল ভন নিউম্যান আর্কিটেকচারে আমাদের একক memory রয়েছে যেখানে

instruction এবং ডেটা উভয়ই সংরক্ষণ করা হয়। আপনি আমাদের চারপাশে যে প্রচলিত কম্পিউটার সিস্টেম দেখেন তা বেশিরভাগ ভন নিউমান আর্কিটেকচারের উপর ভিত্তি করে।

তবে হার্ভার্ড আর্কিটেকচার(architecture) ধারণাগতভাবে দুটি পৃথক memory রয়েছে; একটিতে প্রোগ্রাম সংযোগ করা হয়, অন্যটিতে আপনি ডেটা সংযোগ করা হয়। বেশিরভাগ মাইক্রোকন্ট্রোলারই এই নীতি অনুসরণ করে; তাদের আলাদা memory আছে। সুতরাং, instruction সাধারণত কোনও ROM এ সংরক্ষণ করা হয় যখন কোনও RAM এ ডেটা সংরক্ষণ করা হয়।

(Refer Slide Time: 08:47)



চিত্রগতভাবে ভন নিউমান এবং হার্ভার্ডের architecture গুলি এটির মতো দেখানো যেতে পারে। ভন নিউমান আর্কিটেকচারে আপনি দেখতে পাচ্ছেন এখানে ডেটা এবং program memory রয়েছে; যদিও আপনি এগুলিকে পৃথক হিসাবে দেখিয়ে চলেছেন, কিন্তু CPU একই address ব্যবহার করে তাদের একই ডেটা বাসের(Data Bus) মাধ্যমে প্রবেশ(Access) করছে।

সুতরাং, CPUর ক্ষেত্রে এটি প্রদর্শিত হবে যে memory একই, তবে বাস্তবে আমরা এই চিত্রটি যেমন দেখায় ঠিক তেমন আলাদা আলাদা জায়গায় আলাদা করে রাখতে পারি। তবে CPU সম্পর্কিত যেভাবে ঠিকানাগুলি উত্পন্ন হয় তা একীভূত করা হয়, একই address এবং ডেটা লাইনগুলি প্রোগ্রামের পাশাপাশি ডেটা স্থানান্তর করতে ব্যবহৃত হয়। এবং IO ডিভাইসগুলিও একই বাসের মাধ্যমে সাধারণত সংযুক্ত থাকে। ক্লিপক(Typical) ভন নিউম্যান আর্কিটেকচারটি দেখতে কেমন লাগে। অবশ্যই, পরিশীলিত Architecture রয়েছে যেখানে সমান্তরালভাবে ডেটা স্থানান্তর করার জন্য একাধিক বাস থাকতে পারে।

হার্ডওয়ার্ড আর্কিটেকচারে পৃথক ডেটা ও প্রোগ্রাম memory রয়েছে এবং address এবং ডেটা বাসগুলি সম্পূর্ণ পৃথক। সুতরাং, আপনি সমান্তরালভাবে কোনও প্রোগ্রাম আনার সময় আপনি ডেটা memoryতে কিছু ডেটা আনতে বা লিখতেও পারেন। এবং কিছু architecture এ IO ডিভাইসগুলি পৃথক address এবং ডেটা বাসের মাধ্যমে সংযুক্ত থাকে।

অসুবিধাটি হ'ল বাহ্যিক ডিভাইসগুলির ইন্টারফেস করার জন্য আপনার প্রচুর বাস(BUS), প্রচুর IO পিনের প্রয়োজন, তবে সুবিধাটি হল আপনি দ্রুত এবং সমান্তরাল ডেটা স্থানান্তর বৈশিষ্ট্যগুলি পান।

(Refer Slide Time: 10:56)

What is a Microprocessor?

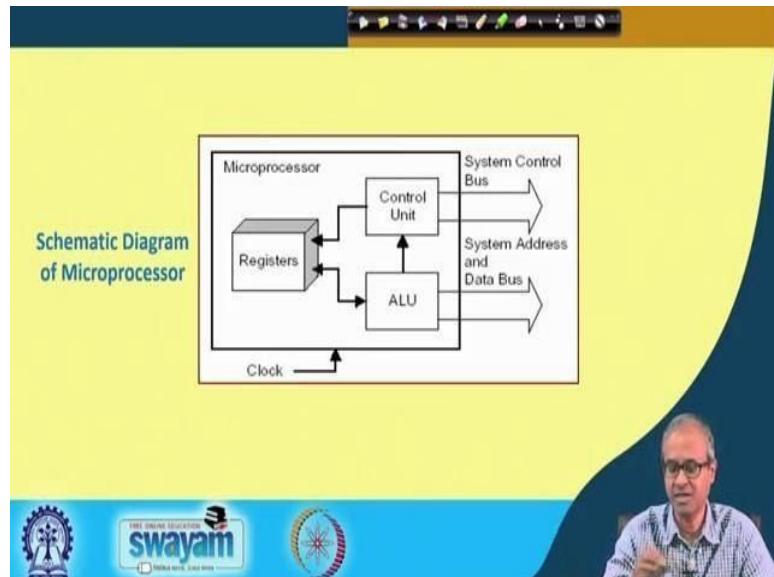
- It is basically the entire CPU fabricated on a single chip.
 - Consists of a set of registers to store temporary data.
 - Consists of an arithmetic logic unit (ALU), where all arithmetic and logical computations are carried out.
 - Consists of some mechanism to interface external devices (memory and I/O) through buses (address, data and control).
 - Consists of a control unit that synchronizes the operation.

আসুন এখন একটি মাইক্রোপ্রসেসরের আসি। মাইক্রোপ্রসেসর কী? আচ্ছা আমরা দেখেছি একটি CPU কি। মাইক্রোপ্রসেসর একটি সাধারণ শব্দ, যার অর্থ একটি CPU একক ইন্টিগ্রেটেড সার্কিট বা আইসি চিপের মধ্যে বসানো। অর্ধপরিবাহী প্রযুক্তির অগ্রগতির সাথে আগে CPU সার্কিটগুলি খুব বড় এবং বিশাল ছিল; তারা আরও ছোট এবং ছোট হতে শুরু করেছে এবং এখন তারা একক চিপের ভিতরে ফিট করতে পারে। উদাহরণস্বরূপ, আমরা যদি পেন্টিয়াম CPU চিপটি(Pentium CPU Chip) দেখি তবে চিপের অভ্যন্তরে কোটি কোটি বেসিক উপাদান বা ট্রানজিস্টর(Transistor) রয়েছে।

মাইক্রোপ্রসেসরের ভিতরে জিনিসগুলি সাধারণত কী থাকে? রেজিস্টারগুলির একটি সেট রয়েছে, কিছু সাধারণ কিছু বিশেষ যা অস্থায়ীভাবে ডেটা সংরক্ষণের জন্য ব্যবহৃত হয়। একটি ALU আছে যেখানে সমস্ত ডেটা প্রসেসিং করা হয়, উভয় **গাণিতিক**(arithmetic) অপারেশন এবং লজিক্যাল(Logical) অপারেশন এবং memory IO ডিভাইসগুলিকে ইন্টারফেস করার জন্য কিছু ব্যবস্থা রয়েছে। কিছু বাহ্যিক বাস এবং নিয়ন্ত্রণ ব্যবস্থা রয়েছে যার মাধ্যমে আপনি মাইক্রোপ্রসেসরের সাথে memory বা

IO ডিভাইসগুলি সংযুক্ত করতে পারেন। এবং অবশ্যই, একটি কন্ট্রোল ইউনিট রয়েছে যা সিস্টেমের মন্তিষ্ঠ হিসাবে বিবেচিত হতে পারে, যা মাইক্রোপ্রসেসরের পুরো অপারেশনকে নিয়ন্ত্রণ করে, এটি সমস্ত অভ্যন্তরীণ ক্রিয়াকলাপকে সিঞ্চনাইজ(Synchronise) করে। মাইক্রোপ্রসেসরটি মূলত এটি।

(Refer Slide Time: 12:44)



পর্যায়ক্রমে আপনি দেখতে পাচ্ছেন যে এগুলি মূল উপাদান, আপনি এখানে রেজিস্টারগুলি দেখতে পারেন, আপনি ALU দেখতে পারেন, আপনি নিয়ন্ত্রণ ইউনিট দেখতে পারেন। এবং বাহ্যিকভাবে কয়েকটি বাস, অ্যাড্রেস(address) এবং ডেটা বাস(Data Bus) রয়েছে এবং সেখানে নিয়ন্ত্রণের বাস থাকতে পারে, বহিরাগত memory এবং IO ডিভাইসগুলির সাথে ইন্টারফেস করার জন্য address এবং ডেটা বাস(Data Bus) ব্যবহার করা হবে। এবং কন্ট্রোল বাসে অন্যান্য সংকেত যেমন read, write, enable, বিভিন্ন ধরণের interrupt এবং অন্যান্য জিনিস থাকবে, ধীর memoryগুলিকে ফাঁকে ফাঁকে দেওয়ার জন্য কিছু সংকেত থাকবে। অন্যান্য অনেক সংকেত রয়েছে যা নিয়ন্ত্রণের উদ্দেশ্যে ব্যবহৃত হয়।

(Refer Slide Time: 13:30)

What is a Microcomputer?

- It is a computer system built using a microprocessor.
- Since a microprocessor does not contain memory and I/O, we have to interface these to build a microcomputer.
- Too complex and expensive for very small and low-cost embedded systems.

The diagram illustrates the internal structure of a microcomputer. At the center is the Micro-processor. It is connected to three main components: Memory, I/O Ports, and a Control Bus. The Memory and I/O Ports are connected to the Micro-processor via Data Buses. The Micro-processor is connected to the Memory via an Address Bus and to the I/O Ports via a Control Bus. The Control Bus also connects the Micro-processor to the Memory.

এখন, মাইক্রো কম্পিউটারটি(Micro Computer) কী তা দেখি। মাইক্রোকম্পিউটার একটি কম্পিউটার যা একটি মাইক্রোপ্রসেসরের চারপাশে নির্মিত হয়। আপনি আমাদের ডেঙ্কটপের দিকে তাকান, আপনি আমাদের ল্যাপটপের দিকে তাকান, একটি পেন্টিয়াম রয়েছে যা আমরা আমাদের আধুনিক ডেঙ্কটপ এবং ল্যাপটপের অভ্যন্তরে ইন্টেল আই 3, আই 5, আই 7 ক্লাস প্রসেসর(Intel i3,i5,i7 class processor) সম্পর্কে কথা বলি। এগুলি মূলত মাইক্রোপ্রসেসর এবং ডেঙ্কটপ বা ল্যাপটপের অভ্যন্তরে কেবল মাইক্রোপ্রসেসরই থাকে না, memory রয়েছে, অন্যান্য ডিভাইস রয়েছে, একটি ডিস্ক ড্রাইভ রয়েছে কিছু অন্যান্য ইন্টারফেস রয়েছে কীবোর্ড এবং মাউস সংযোগের জন্য, উদাহরণস্বরূপ, প্রিন্টারগুলি যা আমাদের সম্পূর্ণ করে তোলে কম্পিউটার। এটি একটি মাইক্রো কম্পিউটার, এটি একটি মাইক্রোপ্রসেসর ব্যবহার করে নির্মিত একটি কম্পিউটার সিস্টেম।

এখন, একটি মাইক্রোপ্রসেসরে মূলত কিছু রেজিস্টার, ALU এবং নিয়ন্ত্রণ থাকে; এটিতে কোনও memory বা IO-র কোনও সুবিধা নেই। একটি কম্পিউটার সিস্টেম তৈরি করতে আমাদের এই সমস্ত ডিভাইসকে মাইক্রোপ্রসেসরের সাথে ইন্টারফেস করতে হবে। তবে, সমস্যাটি হ'ল যেহেতু আপনাকে মাইক্রোপ্রসেসরের চারপাশে এতগুলি জিনিস ইন্টারফেস করতে হবে, সিস্টেমটি জটিল, ভারী এবং ব্যবহৃত হয়ে ওঠে। এছাড়াও এটি উল্লেখযোগ্যভাবে উচ্চ শক্তি গ্রহণ করে। যদি আপনি এটি ডেঙ্কটপ বা ল্যাপটপের মতো উচ্চ পারফরম্যান্স অ্যাপ্লিকেশনে(High performance Application)র জন্য ব্যবহার করেন তবে এটি ঠিক আছে তবে এন্ডেড হওয়া সিস্টেমগুলির জন্য নয়। অতি স্বল্প শক্তি, বহনযোগ্যতা, ছোট আকারের এই বৈশিষ্ট্যগুলি বেশ প্রয়োজনীয়।

আপনি দেখতে পাচ্ছেন এমন একটি মাইক্রো কম্পিউটার একটি মাইক্রোপ্রসেসরের চারপাশে নির্মিত, আপনার memory রয়েছে, আপনি IO পোর্ট(Port) এবং এই IO পোর্টগুলির(Port) মাধ্যমে আপনি বেশ কয়েকটি IO ডিভাইসের সাথে ইন্টারফেস করতে পারেন।

(Refer Slide Time: 15:50)

এখন আসুন মাইক্রোকন্ট্রোলারের ব্যাপারে আসি যা এন্ডেড সিস্টেমগুলির হৃদয়। মাইক্রোকন্ট্রোলার কী? একটি মাইক্রো কম্পিউটার আমরা বলেছি, এটি একটি মাইক্রোপ্রসেসরের চারপাশে নির্মিত কম্পিউটার সিস্টেম। এখন, যদি পুরো মাইক্রো কম্পিউটারটি আমরা সংকুচিত করতে পারি এবং একটি একক চিপের ভিতরে রাখতে পারি তবে আমি একটি মাইক্রোকন্ট্রোলার পাই।

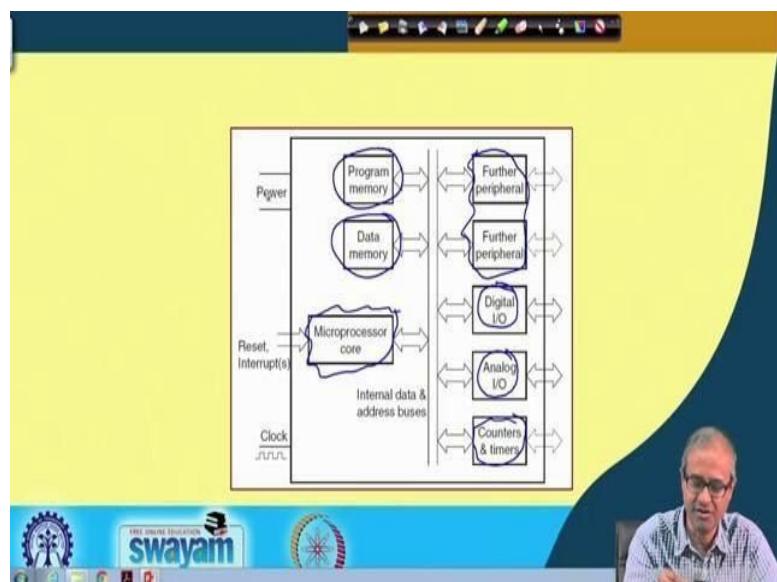
যেহেতু আমি সমস্ত কিছু একক চিপে রাখছি, মোট চিপ অঞ্চলটি প্রসেসর, memory ইত্যাদি দ্বারা ভাগ করে নিতে হবে সুতরাং, একটি মাইক্রোকন্ট্রোলার মূলত একটি চিপে একটি কম্পিউটার, কারণ এটি একটি চিপে থাকে এটি তুলনামূলকভাবে খুব কম ব্যবহৃত এবং সম্ভা। এটি ছোট, এটি খুব কম শক্তি ও খরচ করে।

প্রকৃতপক্ষে এগুলি এন্ডেড সিস্টেম ডিজাইনে খুব বেশি ব্যবহৃত হয়। আপনি যদি এই পরিকল্পনার দিকে তাকান তবে উপস্থিত বৈশিষ্ট্যগুলি কী কী তা বুঝতে পারেন আপনি দেখতে পাবেন যে এখানে CPU, রেজিস্টার, আগ্রহিত এবং নিয়ন্ত্রণ রয়েছে। কিছু memory অবশ্যই আছে, আপনার খুব বড় memory থাকতে পারে না, কিছু memory রয়েছে যা আকারে ছোট। IO ডিভাইস সংযোগের জন্য কিছু সুবিধা থাকতে পারে। কিছু টাইমার এবং কাউন্টার রয়েছে যা অনেকগুলি অ্যাপ্লিকেশনের জন্য প্রয়োজনীয়, সেখানে interrupt সার্কিট রয়েছে।

এবং এনালগ পোর্টগুলির(Analog ports) মতো আরও অনেক সুবিধা রয়েছে। বহিরাগত বিশ্ব থেকে যে সংকেতগুলি আসছে তা বেশিরভাগই সেগুলি অবিচ্ছিন্ন বা প্রকৃত অ্যানালগ, সেগুলি ডিজিটাল নয়। সুতরাং, আমাদের যদি চিপটিতে অন্তর্নির্মিত ডিজিটাল কনভার্টারের(Internal Digital convertor) সাথে এক ধরণের অ্যানালগ থাকে তবে এই জাতীয় ডিভাইসগুলি ইন্টারফেস করা খুব সুবিধাজনক হয়ে ওঠে। সাধারণত এই সমস্ত সুবিধা চিপের ভিতরে feed করা হয়। মাইক্রোকন্ট্রোলাররা সাধারণত কিছু ডেটাতে কাজ করে যা কিছু ইনপুট পোর্টের মাধ্যমে করা হয়; পোর্টগুলির মাধ্যমে বাইরে থেকে আসা ডেটা এবং এখানে একটি প্রোগ্রাম চলছে।

এবং আমি যেমন বলেছি যে তাদের এনালগ পিন, টাইমার, কাউন্টার এবং অন্যান্য ইউটিলিটি সার্কিটরি(Analog Pin, Timer, Counter and other utility Circuitry) রয়েছে। উদাহরণস্বরূপ, আপনার একটি Pulse Width Modulation সার্কিট থাকতে পারে যা খুব কার্যকর কারণ আমরা পরে দেখব।

(Refer Slide Time: 18:59)



এটি একটি মাইক্রোকন্ট্রোলারের একটি সামান্য বিশদ চিত্র। এখানে বেসিক মাইক্রোপ্রসেসর রয়েছে, রেজিস্টার, ALU এবং নিয়ন্ত্রণ, কিছু প্রোগ্রাম memory আছে, প্রোগ্রাম সংরক্ষণ করার জন্য একটি memory রয়েছে, আপনার ডেটা (RAM) সংরক্ষণ করার জন্য একটি memory রয়েছে এবং কিছু ইনপুট আউটপুট পিন থাকতে পারে। আপনি দেখতে পাচ্ছেন যে কয়েকটি IO ডিজিটাল হতে পারে, কিছু অ্যানালগ(Analog) হতে পারে; এখানে কাউন্টার এবং টাইমার থাকবে এবং পরিশীলিত ধরণের ইন্টারফেসও থাকতে পারে যেমন সিরিয়াল IO ইন্টারফেস, Pulse width modulated ইন্টারফেস(Interface), interrupt ইত্যাদি। এই সমস্ত সুবিধা আজ সাধারণ মাইক্রোকন্ট্রোলারগুলিতে সরবরাহ করা হয়, যাতে আপনি প্রায় কোনও প্রকারের জন্য এটি ব্যবহার করতে পারেন এষ্বেডেড

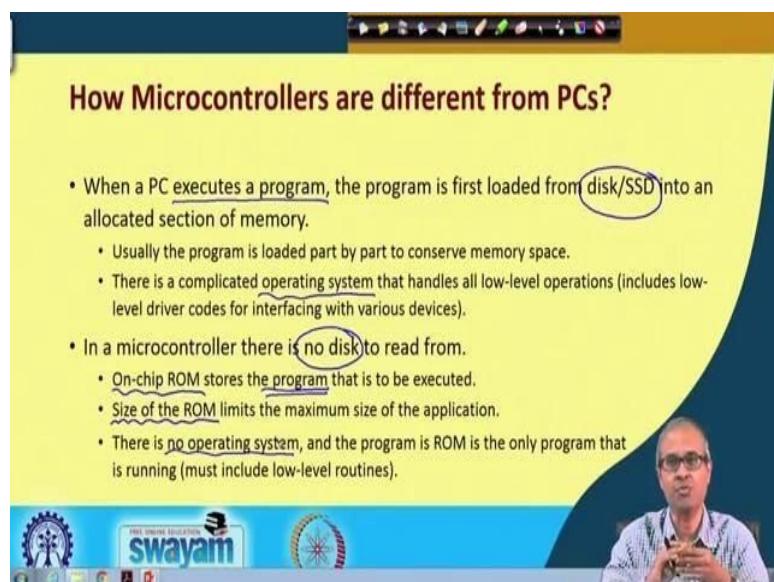
সিস্টেম অ্যাপ্লিকেশনগুলির এছাড়াও আপনার কাছে power supply, reset, interrupts, clock ইত্যাদি রয়েছে।

(Refer Slide Time: 20:07)



এগুলি আপনি দেখতে পাচ্ছেন কিছু সাধারণ মাইক্রোকন্ট্রোলার যা PIC নামে একটি সংস্থা তৈরি করে। এখন, তুলনায় আপনি পুরানো মাইক্রোপ্রসেসরগুলির একটি Motorola 68000 দেখতে পান। এটি একটি 48 পিনের চিপ(48 Pin Chip), আকারে এত বড়। তুলনায় ক্ষুদ্রতম PIC প্রসেসর আপনি দেখতে পাচ্ছেন এমন একটি মুদ্রার আকারের।

(Refer Slide Time: 20:53)



এখন, মাইক্রোকন্ট্রোলাররা তাদের নিজস্বতায় কম্পিউটার। PC হ'ল একটি মাইক্রোপ্রসেসরের নির্মিত কম্পিউটার। তাহলে, মূল পার্থক্যগুলি কী কী? এখনও কিছু পার্থক্য আছে; যখন কোনও PC প্রোগ্রামটি চালায়, সমস্যাটি কোথা থেকে আসে?

এগুলি প্রথমে হার্ড ডিস্ক বা ফ্ল্যাশ memory প্রযুক্তির উপর ভিত্তি করে সলিড স্টেট ড্রাইভে(Solid state drive) সংরক্ষণ করা হয়, সেখান থেকে প্রোগ্রামটি মেমরিতে লোড হয়ে যায় এবং memory থেকে একে একে চালিত করার নির্দেশাবলী। একটি সাধারণ কম্পিউটার সিস্টেম এইভাবে কাজ করে। উইন্ডোজ বা লিনাক্সের(Windows or Linux) মতো অপারেটিং সিস্টেম রয়েছে, তারা মোটামুটি জটিল প্রোগ্রাম। প্রোগ্রামটি কার্যকর হওয়ার সময় তারা নিম্ন স্তরের সমস্ত ক্রিয়াকলাপ পরিচালনার জন্য দায়বদ্ধ। যখনই কোনও ইনপুট আউটপুট অপারেশন হয় এটি অপারেটিং সিস্টেম, এতে চালকরা IO অপারেশনগুলি যন্ত্র নেওয়ার জন্য আহ্বান জানানো হয়।

তবে একটি মাইক্রোকন্ট্রোলার একটি খুব ছোট সিস্টেম। আপনি একটি ডিস্ক রাখার সামর্থ্য রাখতে পারবেন না, সবকিছু সেই মাইক্রোকন্ট্রোলারের মধ্যেই থাকবে। একটি কম্পিউটার সিস্টেমে আপনি ডিস্ক থেকে আপনার যে কোনও প্রোগ্রাম লোড করতে পারেন এবং এটি চালাতে পারেন, তবে এসি মেশিনের অভ্যন্তরে বসে থাকা একটি মাইক্রোকন্ট্রোলার কেবল সেই প্রোগ্রামটি চালাবে যা এসি মেশিনকে(AC Machine) নিয়ন্ত্রণ করার উদ্দেশ্যে বানানো। সুতরাং, একটি সাধারণ মাইক্রোকন্ট্রোলারে চিপের অভ্যন্তরে একটি ছোট্ট ROM থাকে, এটি স্থির প্রোগ্রামটি সংরক্ষণ করবে, প্রোগ্রামটি পরিবর্তন করা যাবে না। লক্ষ্য করার বিষয়টি হ'ল যে ROM সরবরাহ করা হয় তার সর্বোচ্চ আকারটি মাইক্রোকন্ট্রোলারের উপর চালানো প্রোগ্রামের আকার এবং জটিলতার সীমাবদ্ধ করে।

এবং কোনও অপারেটিং সিস্টেম নেই, যখনই সিস্টেমে পুনরায় সেট করা হবে পাওয়ার রমে (Power Rom) সঞ্চিত প্রোগ্রামটি চলতে শুরু করে। এগুলি মাইক্রোকন্ট্রোলার এবং PC-র মতো একটি সাধারণ কম্পিউটিং সিস্টেমের মধ্যে প্রধান পার্থক্য।

(Refer Slide Time: 23:35)

Where are Microcontrollers Used?

- Typically in applications where processing power is not critical.
 - Modern-day household can have 10 to 50 such devices embedded in various devices and equipments.
- One-third of the applications are in the office automation segment.
- Another one-third are in consumer electronics goods.
- Rest one-third are used in automotive and communication applications.

মাইক্রোকন্ট্রোলারগুলি অ্যাপ্লিকেশনগুলিতে ব্যবহৃত হয় যেখানে প্রসেসিং শক্তি সমালোচনামূলক নয়, আপনার খুব উচ্চ প্রসেসিং শক্তি প্রয়োজন হয় না বরং আপনার প্রয়োজন কম শক্তি, ছোট আকার, কম ব্যয় এবং এই জাতীয় বৈশিষ্ট্য। আপনি আধুনিক দিনে সাধারণত এমন অনেকগুলি ডিভাইস সন্ধান করেন, আপনি যে কোনও সরঞ্জাম কিনুন এবং ইনস্টল করুন না কেন এর অভ্যন্তরে কিছু এমবেডেড মাইক্রোকন্ট্রোলার থাকবে।

সুতরাং, আমি ইতিমধ্যে অফিস অটোমেশন সেগমেন্ট, consumer ইলেকট্রনিক্স, মোটরগাড়ি, যানবাহন, বিমান, রকেট স্পেসগ্যাস্ট্র, যোগাযোগ এই ডিভাইসগুলি যে কোনও জায়গায় খুঁজে পাই সে সম্পর্কে ইতিমধ্যে অনেক অ্যাপ্লিকেশন বলেছি।

(Refer Slide Time: 24:26)

Evolution of Microcontrollers

- Microcontroller evolved from a microprocessor-based board-level design to a single chip in the mid-1970's.
 - As the process of miniaturization continued, all of the components needed for a controller were built into a single chip.
- In the mid-1980's, microcontrollers got embedded into a larger ASIC (Application Specific Integrated Circuit).
 - Microcontrollers are fabricated as a module inside a larger chip.

এখন, বিবর্তনের কথা বলছি, 1970 সালের পর থেকে মাইক্রোকন্ট্রোলারের প্রথম প্রজন্মের বিকাশ শুরু হয়েছিল। একটি মাইক্রোকন্ট্রোলার ছিল যা ইন্টেল(Intel) দ্বারা বিকাশ করা হয়েছিল, এটি ছিল 8051, এটি এখনও ব্যবহার করা হচ্ছে, তবে শক্তি, **flexibility** এবং উন্নতবন্নী সম্পর্কে কথা বলার সাথে, 8051 এর একটি পুরানো ধরণের আর্কিটেকচার রয়েছে, এতে খুব বেশি **flexibility** নেই।

৮০ এর মাঝামাঝি থেকে শুরু করে প্রক্রিয়াটি অব্যাহত রয়েছে, আমরা এখন যা দেখছি তা হ'ল মাইক্রোকন্ট্রোলাররা বড় বড় চিপগুলির মধ্যে এন্সেড হয়ে চলেছে। এগুলিকে অ্যাপ্লিকেশন নির্দিষ্ট সংহত সার্কিট বলা হয়। এই কোর্সের অংশ হিসাবে আমরা যে মাইক্রোকন্ট্রোলারটি প্রদর্শন করব এটির মতো, এই মাইক্রোকন্ট্রোলার বোর্ড যা এসটি মাইক্রো ইলেক্ট্রনিক্স থেকে আসে, এটির ভিতরে এআরএম কর্টেক্স এম 4 চিপ(ARM Cortex M4 Chip) রয়েছে, এখন আরএম কর্টেক্স এম 4(ARM Cortex M4 Chip) কেবল এআরএম প্রসেসর(ARM Processor) নয়, কেবল একটি মাইক্রোকন্ট্রোলার নয়, তবে অন্যান্য অনেকগুলি জিনিস একই চিপে একীভূত হয়।

(Refer Slide Time: 26:03)

Advantages of using microcontrollers

- Fast and effective
 - The architecture correlates closely with the problem being solved (control systems).
- Low cost / Low power
 - High level of system integration within one component.
 - Only a handful of components needed to create a working system.
- Compatibility
 - OpCodes and binaries are the SAME for all 80x51 / ARM / PIC variants.

মাইক্রোকন্ট্রোলারদের সুবিধাগুলির সংক্ষিপ্তসার হিসাবে, তারা ক্রত, তারা তাতে কিছু নির্দিষ্ট অ্যাপ্লিকেশন সমাধানের দৃষ্টিকোণ থেকে কার্যকর। এটি অবশ্যই স্বল্প ব্যয়বহুল কারণ এটি যদি উচ্চ ব্যয় হয় তবে এটি পণ্যের ব্যয়ও বাড়িয়ে তোলে, এটি অবশ্যই কম বিদ্যুত গ্রহণ করবে, তাই আপনি জানেন যে আপনার বৈদ্যুতিক বিলটি হিট করা উচিত নয়।

এবং সামঞ্জস্যতা খুব গুরুত্বপূর্ণ। সাধারণত আপনি দেখতে পাবেন যে মাইক্রোকন্ট্রোলাররা নির্দিষ্ট কিছু সংস্থা থেকে আসে; আপনি বলতে পারেন এগুলি মাইক্রোকন্ট্রোলারের পরিবার। পিআইসি, এআরএম এগুলি কয়েকটি সাধারণ উদাহরণ। এখন, আপনি যদি পরিবারের জুড়ে মাইক্রোকন্ট্রোলারদের দিকে নজর দেন তবে তারা সমস্ত নির্দেশের সাথে সামঞ্জস্য রেখেছেন যাতে আপনি একবার কোনও সদস্যের

জন্য স্টেওয়্যার তৈরি করেন এবং আপনি আরও ভাল প্রসেসরে চলে যান, একই কোড খুব অল্প সংশোধন করে চালানো বা কার্যকর করতে পারে। এটিকে সামঞ্জস্যতা বলা হয় এবং রফ্ফণবেক্ষণের ক্ষেত্রে এটি অত্যন্ত গুরুত্বপূর্ণ।

এটির সাথে আমরা এই বক্তৃতার শেষে এসেছি। পরবর্তী বক্তৃতায় আমরা কীভাবে মাইক্রোকন্ট্রোলারদের(Microcontroller) ARM পরিবারের বিষয়ে বিশেষভাবে প্রসেসিং দ্রুততর করতে পারি সে সম্পর্কে কিছুটা আলোচনা শুরু করব। আমরা বিশেষভাবে দেখতে পাব যে মাইক্রোকন্ট্রোলারদের ARM পরিবারের অভ্যন্তরে কী কী বৈশিষ্ট্য রয়েছে এবং কেন তারা আলাদা, কেন লোকেরা এগুলি এত ব্যাপকভাবে ব্যবহার করছে।

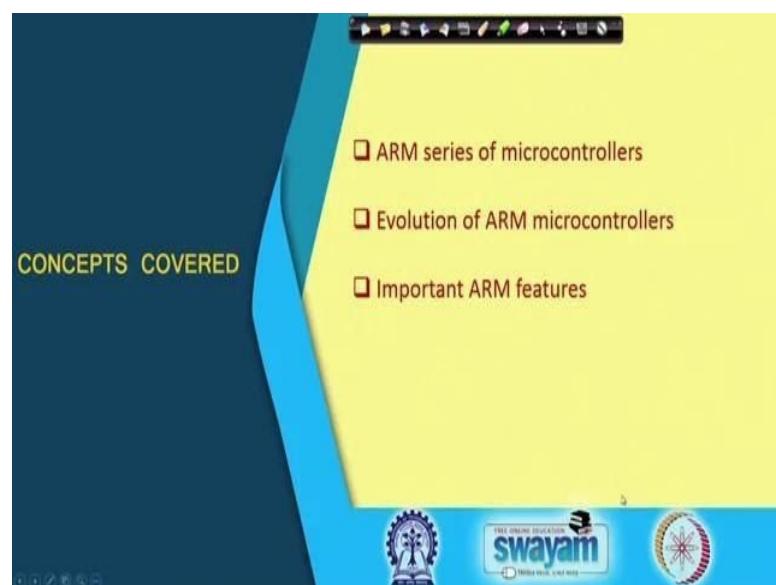
ধন্যবাদ।

Embedded System Design with ARM Prof. Indranil Sengupta Department of Computer Science and Engineering Indian Institute of Technology, Kharagpur

Lecture – 04 Architecture of ARM Microcontroller (Part I)

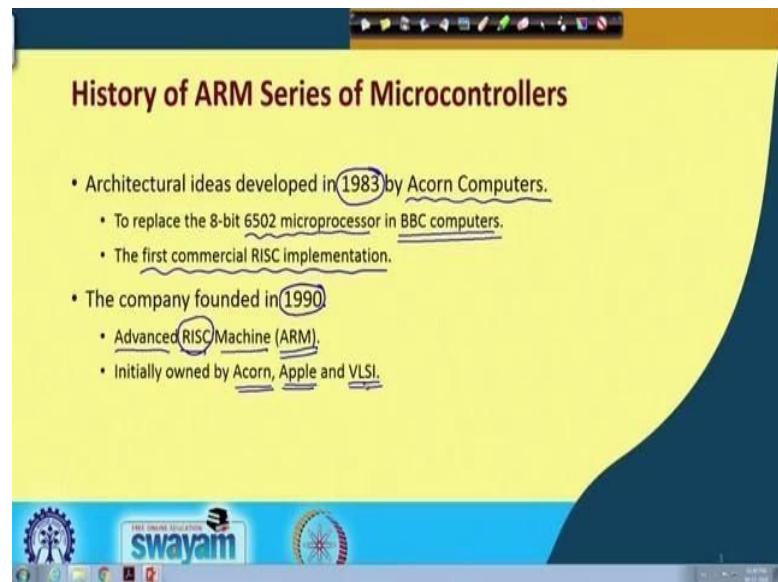
এই বক্তৃতায় আমরা ARM মাইক্রোকন্ট্রোলারদের উপর আমাদের আলোচনা শুরু করব। তাদের আর্কিটেকচার কেমন, তাদের নির্দিষ্ট বৈশিষ্ট্যগুলি কী এবং মাইক্রোকন্ট্রোলারগুলির পূর্ববর্তী প্রজন্মের থেকে তারা কীভাবে আলাদা। এই বক্তৃতার বিষয়টি ARM মাইক্রোকন্ট্রোলারের আর্কিটেকচার(microcontroller architecture), এটি বক্তৃতার প্রথম অংশ।

(Refer Slide Time: 00:41)



এই বক্তৃতাতে আমরা মাইক্রোকন্ট্রোলারদের ARM সিরিজ, তারা কীভাবে বিকশিত হয়েছে এবং কয়েকটি গুরুত্বপূর্ণ আর্কিটেকচার(architecture) এর বৈশিষ্ট্য সম্পর্কে কিছু সাধারণ ধারণা তৈরি করবো।

(Refer Slide Time: 00:57)



আসুন আমরা ইতিহাসটি খতিয়ে দেখি। মাইক্রোকন্ট্রোলারদের এই ARM শ্রেণিতে যে architecture এর ধারণাটি বিকশিত হয়েছে তা 1983 সালে অনেক আগেই বিকশিত হয়েছিল Acorn computers নামে একটি সংস্থা ছিল যা এই ধরণের ধারণাগুলি বিকাশ ঘটায় প্রথম। এখন এই ধারণাগুলি কিছুটা আলাদা ছিল কারণ তারা RISC আর্কিটেকচার ধারণার ভিত্তিতে আর্কিটেকচারাল আইডিয়া বিকাশ করতে শুরু করেছিল। এবং সেই সময়ে Mostek নামে একটি সংস্থা থেকে 6502 নামে একটি খুব জনপ্রিয় মাইক্রোপ্রসেসর ছিল যা BBC মাইক্রো নামে পরিচিত একটি খুব জনপ্রিয় মাইক্রো কম্পিউটারে ব্যবহৃত হয়েছিল।

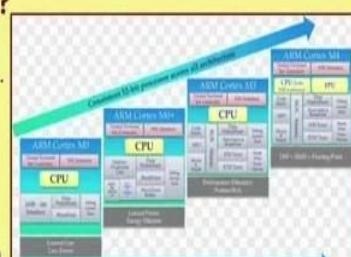
এই প্রথম প্রয়াসটি ছিল আরও শক্তিশালী প্রসেসরের দ্বারা সেই প্রসেসরের প্রতিস্থাপন করা, যা BBC মাইক্রোকে আরও দ্রুত এবং আরও শক্তিশালী করে তুলবে। এর ফলে প্রথম বাণিজ্যিক RISC বাস্তবায়ন হয়েছিল। এটিকে ARM বলা হত না, তবে এটি আরএম আর্কিটেকচারে বিকশিত হয়েছিল। একটি সংস্থা ছিল যা শেষ অবধি 1990 সালে প্রতিষ্ঠিত হয়েছিল। ARM নামটি অ্যারডভান্সড আরআইএসসি(Advanced RISC) মেশিনের সংক্ষিপ্ত নাম। সুতরাং, আপনি নিজে দেখতে পাবেন RISC শব্দটি এন্ডেড হয়েছে।

ARM আর্কিটেকচার মূলত আরআইএসসি আর্কিটেকচারাল ধারণা থেকে ধার করা। প্রাথমিকভাবে এই সংস্থাটি ARM যৌথভাবে অ্যাকর্নের মালিকানাধীন এবং মালিকানাধীন ছিল যা ছিল এর সূচক, অ্যাপলও(Apple) সেখানে ছিল এবং VLSI নামে আরও একটি সংস্থা ছিল। এই তিনটি সংস্থা একত্রিত হয়ে ARM নামে এই নতুন সংস্থা গঠন করে।

(Refer Slide Time: 03:37)

Why do we talk about ARM?

- One of the most widely used processor cores.
- Some application examples:
 - ARM7: iPod
 - ARM9: BenQ, Sony Ericsson
 - ARM11: Apple iPhone, Nokia N93, N100
 - 90% of 32-bit embedded RISC processors till 2010.
- Mainly used in battery-operated devices:
 - Due to low power consumption and reasonably good performance.



90% of 32-bit embedded RISC processors till 2010.

এখন ARM সম্পর্কে এত আকর্ষণীয় কী? কেন আমাদের ARM সম্পর্কে বিশেষভাবে কথা বলতে হবে? আপনি এই কোর্সে জিজ্ঞাসা করতে পারেন আমরা এম্বেড থাকা সিস্টেমগুলি শেখানোর জন্য কেন বিশেষত বাহিনী হিসাবে ARM ব্যবহার করার চেষ্টা করছি। কারণটি হ'ল ARM অনেকগুলি অ্যাপ্লিকেশনে ক্রমবর্ধমানভাবে ব্যবহৃত হচ্ছে, তারা মাইক্রোকন্ট্রোলারগুলির মধ্যে সবচেয়ে জনপ্রিয় বিভাগ যা এম্বেডড সিস্টেম অ্যাপ্লিকেশনগুলিতে গুরুত্ব সহকারে ব্যবহৃত হয়।

ଆସନ୍ତି ଆମରା କ୍ୟେକଟି ଉଦାହରଣ ନିଇ ଯା ସମ୍ପର୍କେ ଆପନାରା ସବାଇ ଜାନେନ୍। ଅୟାପଲ ଥେକେ ଯେ ଆଇପଦଗୁଲିତେ ଆପନି ଗାନ ଶୁଣନ୍ତେ ପାରବେନ ତାର ଭିତରେ ଏକଟି ARM ପ୍ରସେସର ଛିଲ Benq, Sony Ericsson ଏଗୁଲି ଖୁବ ପରିଚିତ ସଂସ୍ଥାଗୁଲି ଯାରା ଟିଭି ସେଟ ଏବଂ ଅନେକଗୁଲି ଅଡ଼ିଓ ଭିଜ୍ୟୁଲ୍ ସରଙ୍ଗାମ ପ୍ରକ୍ରିୟାଗୁଲିର ପ୍ରତ୍ୟେକଟିର ଭିତରେ ARM ପ୍ରସେସର ରଖେଛେ। ମାଧ୍ୟମରେ ତାରା ARM 9 ବ୍ୟବହାର କରା ଶୁଳ୍କ କରେ, ତବେ ପରବର୍ତ୍ତୀକାଳେ ତାରା ARM ପ୍ରସେସରେ ପରବର୍ତ୍ତୀ ସଂକ୍ଷରଣେ ତାଦେର ଆପଣେଇ କରେ। ଆମରା ସକଳେଇ ଅୟାପଲ ଆଇଫୋନଟିର ସାଥେ ପରିଚିତ ଏବଂ ଖୁବ ଜନପିଯ ନୋକିଆ ଫୋନେର କ୍ୟେକଟିର ମଧ୍ୟେ ARM 11 ପ୍ରସେସର ରଖେଛେ।

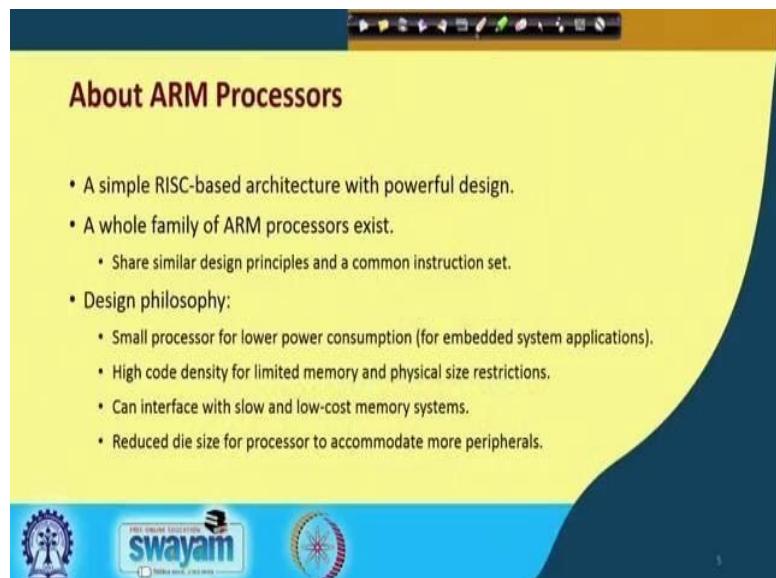
এখন এটি পরিসংখ্যানের বেশ পুরানো অংশ। 2010 অবধি, সমস্ত ওরুতর এন্ডেড থাকা অ্যাপ্লিকেশনগুলির 90% এর ভিতরে ARM প্রসেসর ছিল। আপনি যখন এমবেডেড সিস্টেম অ্যাপ্লিকেশন সম্পর্কে কথা বলবেন, তখন অ্যাপ্লিকেশনটির উপর নির্ভর করে প্রসেসরের কাছ থেকে আপনার কতটা পাওয়ার প্রয়োজন তা স্থির করে নিন। এটি যদি খুব সাধারণ অ্যাপ্লিকেশন হয় তবে আপনার ARM এর মতো শক্তিশালী প্রসেসরের দরকার নেই। আপনি 8-বিট পিআইসি মাইক্রোকন্ট্রোলার ব্যবহার করতে পারেন।

ARM প্রসেসরগুলি সাধারণত 32 bit এবং তার বেশি হয়। সুতরাং, আপনি যখন ARM প্রসেসরটি ব্যবহার করেন যখন আপনার যুক্তিসঙ্গতভাবে শক্তিশালী গণনার সক্ষমতা প্রয়োজন যা আপনার এন্ডেড

করা সিস্টেমের হৃদয়কে পরিণত করে। এখন আর একটি বিষয় হ'ল ARM প্রসেসরগুলির খুব কম বিদ্যুত খরচ এবং অবশ্যই, যুক্তিসংগতভাবে ভাল পারফরম্যান্স রয়েছে। এই বিদ্যুতের কম ব্যবহারের কারণে এগুলি ব্যাটারিচালিত ডিভাইসে খুব বেশি ব্যবহৃত হয়। মোবাইল ফোনের মতো অনেকগুলি ব্যাটারি চালিত ডিভাইস রয়েছে।

আপনি যদি এই চিত্রটি দেখেন তবে এটি কয়েক বছর ধরে ARM ভিত্তিক প্রসেসর এবং ASIC-এর বিবরণ দেখায়।

(Refer Slide Time: 08:09)



প্রথম জিনিসটি হ'ল, ARM মূলত একটি আরআইএসসি ভিত্তিক আর্কিটেকচার। আরআইএসসি আর্কিটেকচার কী তা আমরা কিছু বিশদে যাব। এটি প্রচলিত মাইক্রোকন্ট্রোলারের বিপরীতে কিছু উন্নত architecture এর ধারণা নিয়েছে যেগুলির খুব আদিম ধরণের ডিজাইন ছিল।

আমি 8051 এর কথা বলেছিলাম এটি একটি খুব জনপ্রিয় মাইক্রোকন্ট্রোলার, তবে আর্কিটেকচার অনুযায়ী এটি বেশ আদিম ছিল, এটি কোনও ধরণের architecture এর বর্ধন বা উন্নত বৈশিষ্ট্য ব্যবহার করেনি। যেমনটি আমি আপনাকে বলেছিলাম যে ARM প্রসেসরটি কেবল একটি নয়, তবে পুরো প্রসেসরের পুরো পরিবার এবং সর্বাধিক ওরুস্পূর্ণ বিষয়টি পশ্চাদপদ সামঞ্জস্য বজায় রাখার জন্য এই সমস্ত ভাগ মূলত একটি সাধারণ নির্দেশিকা সেট।

অবশ্যই প্রজন্মগুলিতে কিছু অতিরিক্ত নির্দেশ যুক্ত করা হয়েছে, তবে পুরাণো নির্দেশাবলীও রয়ে গেছে, যেমন একটি প্রবীণ প্রজন্মের জন্য তৈরি একটি প্রোগ্রাম প্রজন্মের জন্যও বেশ ভালভাবে চলতে পারে। এখন এখানে নকশার দর্শনটি ছিল অবশ্যই আমাদের একটি ছোট প্রসেসরের প্রয়োজন

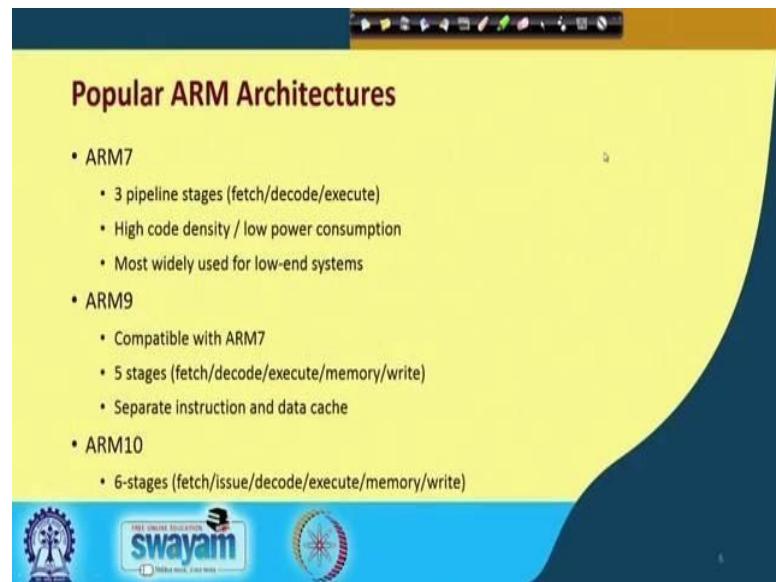
যাতে আমাদের কম বিদ্যুতের খরচ হয় এবং এন্ডেড থাকা সিস্টেমগুলির প্রয়োগের জন্য ব্যবহার করা যায়।

সুতরাং, প্রসেসরের আকার ছোট হওয়া উচিত, এটি কম শক্তি এবং উচ্চ কোডের ঘনত্ব গ্রহণ করা উচিত। আপনি মাইক্রোকন্ট্রোলারগুলিতে দেখতে পেয়েছেন যে প্রোগ্রাম মেমরি এবং ডেটা মেমরি সমন্বয়ের ভিতরে রয়েছে। রিয়েল এস্টেটের ঘাটতি রয়েছে; আপনি খুব বড় মেমরি ভিতরে রাখতে পারবেন না।

সুতরাং, আপনি যে প্রোগ্রামটি চালাতে পারেন তার আকারের সর্বাধিক সীমা রয়েছে। আসুন আমরা বলি যে প্রোগ্রামটির মেমরির আকার 100 কিলোবাইট। আমি যাই লিখি না কেন এটি অবশ্যই এই 100 কিলোবাইটের মধ্যে ফিট করে। সুতরাং, যদি আমার নির্দেশিকা সেটটি সমর্থন করে যে এই 100 কিলোবাইটে, আমি আমার কোডটি খুব সুন্দরভাবে প্যাক করতে পারি, যাতে আমি কোনও ধরণের প্রতিযোগিতামূলক আর্কিটেকচারের তুলনায় আরও কার্যকারিতা বাস্তবায়িত করতে পারি যেখানে একই জিনিসটি বাস্তবায়নের জন্য আরও অনেক মেমরি প্রয়োজন। আমি যা বলতে চাইছি তা হ'ল, ধরুন আমার একটি অ্যাপ্লিকেশন X রয়েছে যা আমি প্রচলিত আর্কিটেকচারে প্রয়োগ করতে চাই সম্ভবত এটির জন্য প্রয়োজন হবে 120 কিলোবাইট তবে আরআইএসসি ARM আর্কিটেকচারে আমি এটি 100 কিলোবাইটের মধ্যে ফিট করতে পারি।

এটিকে হাইড কোড ঘনত্ব(high code density) বলা হয়। কিছু নির্দেশাবলীর বৈশিষ্ট্য রয়েছে যা আমাদের প্রয়োজনীয় নির্দেশাবলীর সংখ্যা হ্রাস করতে দেয়। এটি সীমিত মেমরি এবং চেহারার আকারের সীমাবদ্ধতার সুবিধা নিতে পারে। এবং অবশ্যই, এখানে ইন্টারফেস(interface) অনেক flexibility রয়েছে। আমরা বিভিন্ন ধরণের মেমরি সিস্টেমের সাথে ইন্টারফেস করতে পারি, খুব ধীর বা তুলনামূলকভাবে। এবং অবশ্যই, ছাঁচের আকার হ্রাসের অর্থ হ'ল আপনি যখন চিপটি আসলে বানাচ্ছেন সেই সিলিকনের আকার খুব ছোট, যাতে আপনি যখন ASIC বিকাশ করবেন তখন এটির খুব ছোট অংশটি দখল করবে। সুতরাং, আপনি অবশিষ্ট স্থানটি আরও অনেক কিছুতে রাখতে পারেন; ASIC কে খুব শক্তিশালী করতে অতিরিক্ত কার্যকারিতা সহ।

(Refer Slide Time: 12:40)



জনপ্রিয় কিছু ARM আর্কিটেকচার এখানে দেখানো রয়েছে। অনেকগুলি রয়েছে, আমি কেবল তিনটি দেখছি, ARM 7, 9, 10 সেখানে 11 এবং এর বাইরে রয়েছে। ARM 7 এর 3 টি পাইপলাইন পর্যায় রয়েছে, আমরা পরে পাইপলাইন সম্পর্কে কথা বলব। এখন পাইপলাইন পর্যায়ের মূলত নির্দেশাবলী কীভাবে কার্যকর হয়।

এখানে 3 টি ধাপ রয়েছে: fetch, decode, execute। এটি উচ্চ কোড ঘনত্ব এবং কম বিদ্যুত ব্যবহার সমর্থন করে। আপনার যেকোন জায়গায় খুব বেশি পাওয়ারের ARM প্রসেসরের দরকার নেই, একটি মোবাইল ফোনের ভিতরে আপনার যা প্রয়োজন সম্ভবত কোনও ফ্রিজের ভিতরে আপনার প্রয়োজন হবে না, আপনার খুব সাধারণ ধরণের গণনা প্রয়োজন। ARM 9 এ প্রথম জিনিসটি হ'ল এগুলি সমস্ত পশ্চাদপটে সামঞ্জস্যপূর্ণ তবে পাইপলাইন ধাপগুলি 5 এ বাড়ানো রয়েছে; fetch, decode, execute, memory, write। এবং ক্যাশে মেমরিই ধারণাটি এসেছিল এবং এখানে একটি পৃথক নির্দেশিকা ক্যাশে এবং পৃথক ডেটা ক্যাশে রয়েছে

ARM 7-এ নির্দেশনা এবং ডেটা উভয়ই একই মেমরিতে ছিল। সুতরাং, এটি একটি ভন নিউমান আর্কিটেকচারের মতো ছিল, তবে ARM 9 থেকে আর্কিটেকচারটি হার্ডার্ড আর্কিটেকচারের দিকে পরিবর্তন শুরু করেছিল। ARM 10 এ চলে যাওয়া, মূল পার্থক্যটি ছিল পাইপলাইনটিকে ইস্যু নামক আরেকটি স্তর যুক্ত করে আরও বাড়ানো হয়েছিল।

এইভাবে বেসিক আর্কিটেকচারটি novel আর্কিটেকচারাল ধারণাগুলি যুক্ত করে প্রসেসরটিকে আরও শক্তিশালী এবং দ্রুততর করে গড়ে তোলা শুরু করে।

(Refer Slide Time: 14:42)

	ARM 7 (1995)	ARM9 (1997)	ARM10 (1999)	ARM11 (2003)
Pipeline depth	3-stage	5-stage	6-stage	8-stage
Typical clock frequency (MHz)	80	150	260	335
Power (mW/MHz)	0.06	0.19	0.50	0.40
Throughput (MIPS/MHz)	0.97	1.1	1.3	1.2
Architecture	Non Neumann	Harvard	Harvard	Harvard
Multiplier	8 x 32	8 x 32	16 x 32	16 x 32

এই টেবিলটি 4 টি ARM পরিবারের সদস্যদের ARM 7, 9, 10 এবং 11 এর মধ্যে একটি দ্রুত তুলনা দেয় আপনি প্রথম চালু হওয়ার বছরটি দেখতে পারেন। প্রথম জিনিসটি পাইপলাইন গভীরতা; গভীরতার অর্থ পাইপলাইনের কতগুলি স্তর রয়েছে।

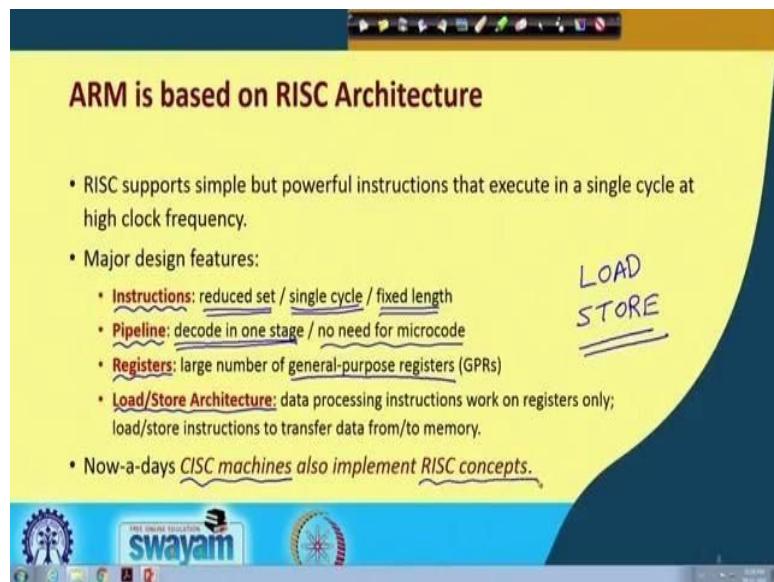
সুতরাং, আমরা কার্যকরভাবে কিছুটা কার্যকর করতে পাইপলাইনে ধাপের সংখ্যা বাড়িয়ে দিচ্ছি। কখনও কখনও প্রসেসরের গতি নির্ধারণ করা হয় যে আমরা কত ক্লক তৈরি করতে পারি। ARM 7 এ তখন 80 মেগাহের্টজ তখন 150, 260, 335 এবং আরও কিছু ছিল। সুতরাং, ক্লকের ক্রিকোয়েন্সিগুলি বাড়ছে, প্রসেসরগুলি দ্রুততর হচ্ছে। বিদ্যুৎ খরচ হ'ল ক্লকের ক্রিকোয়েন্সিরও একটি পরিমাপ, তত দ্রুত ক্লক এ বিদ্যুতের খরচ হবে।

সুতরাং, আপনার ক্লকের ক্রিকোয়েন্সি সম্পর্কিত বিদ্যুৎ খরচ অনুমান করা উচিত, কারণ প্রতিটি মাইক্রোকন্ট্রোলারের কাছে বেশ কয়েকটি অনুমোদিত ক্লকের ক্রিকোয়েন্সি থাকে। এটি আপনি কী ক্লকের ক্রিকোয়েন্সি চান তা নির্ভর করে, আপনি যদি একটি নিম্ন ক্লকের ক্রিকোয়েন্সি নিয়ে পরিচালনা করতে পারেন এবং আপনার উদ্দেশ্যটি ভালভাবে পালন করতে পারেন তবে আপনি নিম্ন শক্তি গ্রহণ করবেন। সুতরাং, মাইক্রোকন্ট্রোলারগুলিতে বিদ্যুত ব্যবহার সাধারণত মেগাওয়ার্টজ প্রতি মিলিওয়াট দ্বারা পরিমাপ করা হয়।

মাইক্রোকন্ট্রোলারগুলিতে আপনি প্রতি মেগাহের্টজ প্রতি সেকেন্ডে মিলিয়ন নির্দেশাবলীর মাধ্যমে ঝর্পুট পরিমাপ করেন। ARM 7 ভন নিউম্যানের উপর ভিত্তি করে তৈরি হয়েছিল, তবে পরবর্তীকালে হার্ডওয়ার আর্কিটেকচারের দিকে অগ্রসর হয় এবং প্রসেসরের অভ্যন্তরে একটি ওণক মধ্যে একটি বিল্ট থাকে।

প্রথম দুটি প্রজন্মের মধ্যে একটি 8×32 ছিল, যেখানে পরবর্তী দুটি প্রজন্মের জন্য একটি 16×32 গুণক ছিল কারণ অনেকগুলি অ্যাসিকেশনএ প্রায়শই গুণ অপারেশন প্রয়োজন। সুতরাং, যদি কোনও হার্ডওয়্যার গুণক অন্তর্নির্মিত থাকে তবে এটি অপারেশনটিকে বেশ উল্লেখযোগ্যভাবে গতি দেয়।

(Refer Slide Time: 18:13)



লক্ষ্য করার বিষয়টি হ'ল ARM আরআইএসসি আর্কিটেকচারের উপর ভিত্তি করে। RISC কিছু architecture এর বৈশিষ্ট্যের উপর ভিত্তি করে।

এই architecture এর বৈশিষ্ট্যগুলি এরকম। নির্দেশাবলীর প্রতি হিসেবে সংখ্যায় কম। নির্দেশাবলী সহজ যাতে সমস্ত নির্দেশাবলী একটি চক্র কার্যকর করা যেতে পারে। এগুলি সমস্ত নির্দিষ্ট দৈর্ঘ্যের যাতে নির্দেশের ডিকোডিং খুব সহজ হয়ে যায় এবং নিয়মকের জন্য হার্ডওয়্যার খুব সহজ হয়ে যায়। তারপরে এখানে পাইপলাইন সম্পর্কে আমরা পরে দেখতে পাব যে সমস্ত আধুনিক দিনের প্রসেসরের একটি পাইপলাইনে সাধারণত নির্দেশাবলী কার্যকর করা হয়। এখন যদি নির্দেশাবলী সহজ হয় তবে তাদের দৈর্ঘ্য নির্দিষ্ট করা হয়, তবে নির্দেশের ডিকোডিং খুব সহজ হয়ে যায়। আপনি নিজেই এক পর্যায়ে ডিকোড করতে পারেন, আপনাকে আবার নির্দেশের দিকে নজর দিতে হবে না এবং এই নির্দেশটি কী ছিল তা জানার চেষ্টা করতে হবে না।

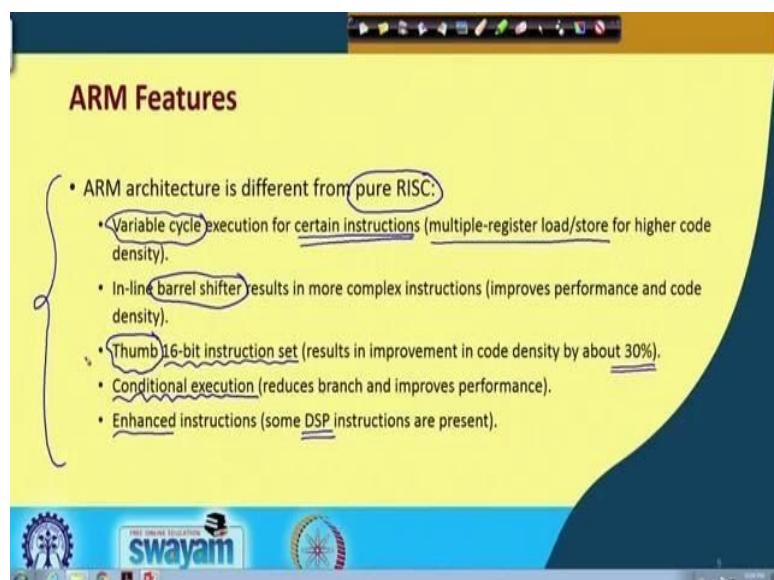
মাইক্রোপ্রোগ্রামিংয়ের কোনও প্রয়োজন নেই যা জটিল নির্দেশ সেট কম্পিউটারগুলির জন্য একটি আদর্শ। আপনি হার্ডওয়্যারে নিয়ন্ত্রণ ইউনিটটি সরাসরি প্রয়োগ করতে পারেন, এটি আরও দ্রুত হয়ে যায় এবং আপনি এটি একটি উচ্চতর ক্লকে চালাতে পারেন। RISC আর্কিটেকচারের একটি বৈশিষ্ট্য হ'ল সাধারণত 32 বা ততোধিক সংখ্যক সাধারণ উদ্দেশ্য নিবন্ধের সংখ্যা খুব বেশি। সিআইএসসি এর

বিপরীতে খুব কম বিশেষ উদ্দেশ্যে নিবন্ধক রয়েছে যেখানে প্রোগ্রাম কাউন্টার, স্ট্যাক পয়েন্টার, বেস রেজিস্টার ইত্যাদির মতো অনেকগুলি বিশেষ special purpose register রয়েছে। এবং অন্য একটি গুরুত্বপূর্ণ বিষয় হ'ল আরআইএসসি একটি লোড / স্টোর আর্কিটেকচারের উপর ভিত্তি করে, যার অর্থ রেজিস্টার এবং মেমরির মধ্যে ডেটা স্থানান্তর করার জন্য দায়বদ্ধ কিছু লোড এবং স্টোর নির্দেশ রয়েছে। অন্যান্য সমস্ত নির্দেশাবলী কেবল রেজিস্টারগুলিতে কাজ করে, তারা মেমরি অ্যাক্সেস করে না।

এই ধরণের নির্দেশিকা সেটকে কথনও কথনও লোড স্টোর আর্কিটেকচার বলা হয়, যেখানে কেবল লোড এবং স্টোর নির্দেশাবলী মেমরি অ্যাক্সেস করে এবং অন্যান্য সমস্ত নির্দেশাবলী কেবল রেজিস্টারে কাজ করে। এখন আমি আগেই বলেছি যে এমনকি আজকের (CISC) মেশিনগুলি যেমন মাইক্রো প্রোগ্রামিং ব্যবহার করে তাদের মেশিনগুলির ইন্টেল শ্রেণির মতো, তারা এই জটিল নির্দেশাবলীকে কিছু ধরণের মাইক্রোপ্রগ্রামগুলিতে অনুবাদ করে যা আরআইএসসি নির্দেশাবলীর চেয়ে বেশি দেখায়।

সুতরাং, তারা কিছু উপায়ে (RISC) ধারণাটিও বাস্তবায়িত করে, তারা একটি প্রাথমিক অনুবাদ করে যার পরে তারা স্ট্যান্ডার্ড আরআইএসসি কৌশল(standard RISC techniques) ব্যবহার করে কার্যকর করে। এখন ARM সম্পর্কে কথা বলছি, ভাল যদিও ARM নামটিতে এই RISC রয়েছে এই মাঝারি RISC-র সংক্ষিপ্ত রূপ,

(Refer Slide Time: 22:13)



সত্যি কথা বলতে ARM কোনও বিশেষ আরআইএসসি আর্কিটেকচার নয়। কিছু বৈশিষ্ট্য রয়েছে যা আর্কিটেকচারে প্রবর্তিত হয়েছিল কারণ এটি এন্ডেড সিস্টেম অ্যাপ্লিকেশনগুলিতে খুব দরকারী, যা আরআইএসসি বৈশিষ্ট্য নয়। কিছু পার্থক্য নিম্নরূপ: নির্দিষ্ট নির্দেশাবলীর জন্য কার্যকর করার জন্য পরিবর্তনশীল সংখ্যক ক্লকের চক্র প্রয়োজন। আরআইএসসির কথা বলার সময়, আমি বলেছিলাম যে

সমস্ত নির্দেশাবলী একক ক্লকের চক্রের মধ্যে কার্যকর করা উচিত, তবে ARM-এ কিছু নির্দেশাবলী আরও জটিল হতে পারে, এর জন্য একাধিক ক্লক প্রয়োজন হতে পারে।

একটি সহজ উদাহরণ একাধিক রেজিস্টার লোড স্টোর। সাধারণত আমরা একটি রেজিস্টারে মেমরি থেকে একটি মান লোড করি, তবে ARM আপনাকে উল্লেখ করতে দেয় যে লোড হওয়া মানটি লোড হবে এটি আমাদের 4 টি রেজিস্টার। সুতরাং, এই 4 টি নিবন্ধে লিখতে আপনার 4 টি ক্লক পাল্স(clock pulse) ব্যবহার হবে। এটি একটি চক্রের মধ্যে করতে পারবেন না, যেমন একাধিক ডেটা স্থানান্তর নির্দেশাবলী ARM-এ সমর্থিত। এবং ব্যারেল শিফটার নামে একটি জিনিস রয়েছে যা একটি খুব সাধারণ architecture এর ধারণা। এটি এমন একটি হার্ডওয়্যার যা একক চক্রে খুব দক্ষতার সাথে একাধিক বিট স্থানান্তরিত করতে দেয়। এই ব্যারেল শিফটারটি ARM আর্কিটেকচারের অংশ এবং এমন অনেকগুলি নির্দেশ রয়েছে যা সরাসরি এই ব্যারেল স্থানান্তর ক্ষমতাটি ব্যবহার করে। আমি একটি উদাহরণ দিতে চাই। ধরুন এখানে একটি ADD নির্দেশনা রয়েছে যা এতে 2 টি রেজিস্টার যুক্ত হয়েছে, আসুন আমরা r2 এবং r3 বলি। এটি যোগ করে, তবে আমি এটিও বলতে পারি যে আপনি 4 অবস্থানের দ্বারা বামে আর 2 এবং আর 3 যুক্ত করেছেন। 4 পজিশনে বাম শিফটার(barrel shifter) দ্বারা সম্পন্ন হবে, এটিতে কোনও অতিরিক্ত সময় লাগবে না, সেই একক ক্লকের চক্রে সবকিছু করা যায়।

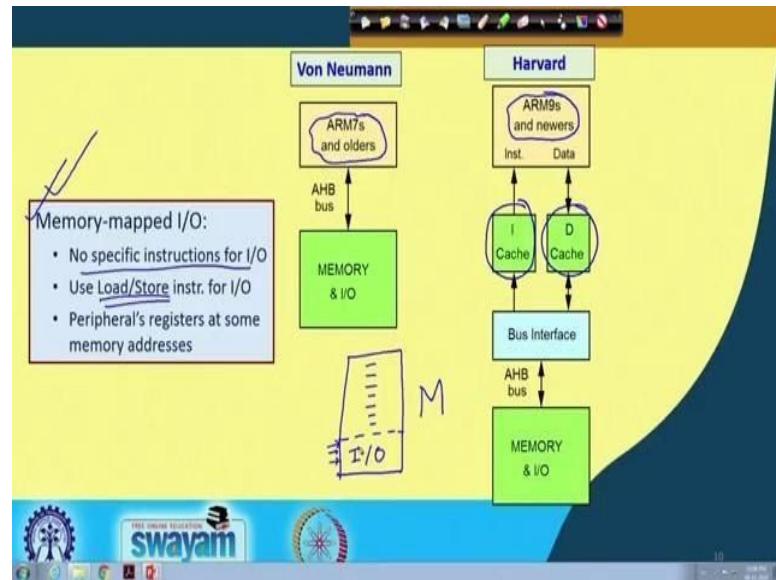
ব্যারেল শিফটারের উপস্থিতির কারণে এই ধরণের শিফট এবং পরিচালনা সংক্রান্ত ধরণের নির্দেশাবলী সম্ভব। এবং অন্য বৈশিষ্ট্যটি হ'ল আপনি থাস্ম মোডে(thumb mode) ARM সংজ্ঞিত(configure) করতে পারেন। থাস্ম মোড হ'ল ARM নির্দেশাবলীর একটি উপসেট, যা 16-বিট মোডে কাজ করে। সাধারণত ARM প্রসেসরগুলি 32-বিট প্রসেসর হয় তবে এমন অনেক অ্যাপ্লিকেশন থাকতে পারে যেখানে আপনার সেই শক্তি প্রয়োজন হয় না, আপনার আরও সহজ শক্তি প্রয়োজন। আপনার কাছে থাস্ম ইন্ট্রাকশন সেট থাকতে পারে যা মূলত একটি 16-বিট নির্দেশিকা সেট(instruction set)।

যদি আমরা ছোট নির্দেশাবলী ব্যবহার করি তবে এটি মোট কোডের আকারকে সংক্ষিপ্ত করে তুলতে পারে। আপনার কোডের ঘনত্ব আরও উল্লত করতে পারে। এবং আরও একটি আকর্ষণীয় বৈশিষ্ট্য রয়েছে যা আমরা এ সম্পর্কে বিস্তারিত আলোচনা করব, যাকে বলা হয় শর্তাধীন বাস্তবায়ন execution আপনি বলতে পারেন যে আপনি এই 2 টি সংখ্যা যুক্ত করেছেন শর্তযুক্ত 0 flag সেট করা আছে। প্রচলিত প্রসেসরগুলিতে যদি 0 flagটি সেট করা থাকে তবে আপনি JZ বা JNZ ধরণের নির্দেশনা রাখতে পারেন, একটি জাম্প করলে তারপর এটি 0 নয় কিনা তা পরীক্ষা করে দেখুন; তার অর্থ, আপনার অনেক জাম্প নির্দেশিকা দরকার।

তবে আপনার যদি শর্তসাপেক্ষ নির্দেশনা থাকে, যেমন আপনি বলেছেন যে 0 flag টি সেট করা থাকে তবে আপনি jump নির্দেশ পুরোপুরি এড়িয়ে চলেছেন। নির্দেশাবলীর সংখ্যাও হ্রাস পায়। এবং অবশ্যই, কিছু বৃদ্ধি আছে যেমন গুন ও যুক্ত করুন, এই ধরণের নির্দেশাবলী নির্দেশ সেটটিতে যুক্ত করা হয়েছে।

কারণ ARM আসল আরআইএসসি(RISC) থেকে কিছুটা বিচ্যুত হয়েছে, তবে এখনও এটি বেশিরভাগ আরআইএসসি(RISC)-এর উপর ভিত্তি করে তৈরি মোটাগুটি শক্তিশালী প্রসেসর, কেবলমাত্র বেশ কয়েকটি ক্ষেত্রে অবশ্যই খুব ভাল কারণেই এটি বিচ্যুত হয়।

(Refer Slide Time: 27:14)

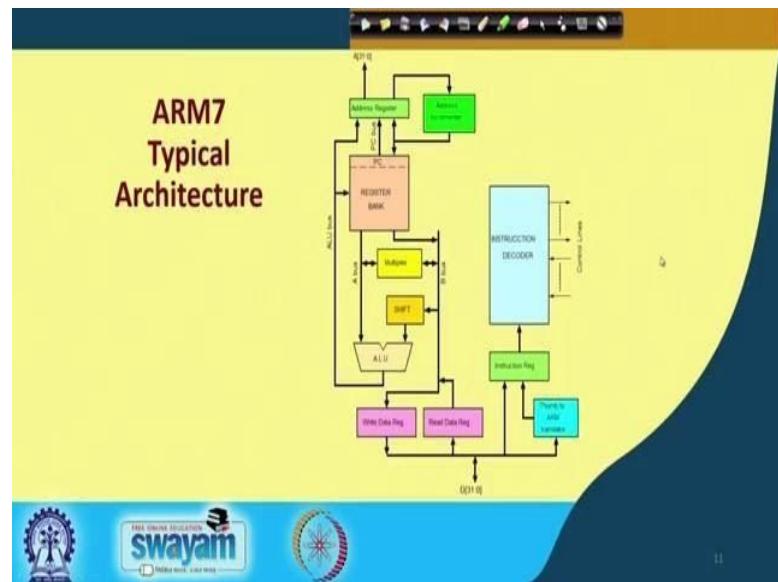


আপনি ইতিমধ্যে এই ভন নিউমান এবং হার্ডি আর্কিটেকচার সম্পর্কে কথা বলছেন যা আপনি ইতিমধ্যে জেনেছেন, এই ARM 7 এবং এমনকি পুরানো প্রসেসরগুলি ভন নিউমানের উপর ভিত্তি করে তৈরি হয়েছিল, সেখানে একটি একক মেমরি ছিল। পরবর্তী প্রসেসরের 2 টি পৃথক মেমরি, নির্দেশ মেমরি এবং ডেটা মেমরি রয়েছে এবং এর ভিতরেও রয়েছে নির্দেশ এবং একটি নির্দেশ ডেটা ক্যাশ। এখন আর একটি বৈশিষ্ট্য হ'ল ARM প্রসেসরগুলির ইনপুট / আউটপুট জন্য আলাদা নির্দেশনা নেই, তারা মেমরি mapped IO নামে কিছু ব্যবহার করেন।

আমাদের বলার মতো এটি হল মোট মেমরি অঞ্চল, এটি আপনার মেমরি। মেমোরির একটি অংশ রয়েছে যা আপনি আইও ডিভাইসগুলির জন্য সংরক্ষণ করেন। সাধারণত আপনি যখন মেমোরি অ্যাক্সেস করেন আপনি এখানে ডেটা সঞ্চয় করেন তবে আপনি যখন এই অঞ্চলে কিছু এড্রেস অ্যাক্সেস করার চেষ্টা করছেন তখন ডিকোডিং সার্কিটের স্বয়ংক্রিয়ভাবে মেমরির পরিবর্তে আইও পোর্টগুলি অ্যাক্সেস করবে।

তবে মেমরি এবং আইও অপারেশনের জন্য একই ডিকোডার থাকবে। লোড স্টোর নির্দেশাবলী সাধারণত মেমরি এবং রেজিস্ট্রারের মধ্যে ডেটা স্থানান্তর করতে ব্যবহৃত হয়, একই লোড স্টোর নির্দেশাবলী আইও পোর্ট থেকে রিড বা আইও রাইট এর জন্য ব্যবহৃত হবে। ইনপুট এবং আউটপুট জন্য পৃথক নির্দেশাবলী নেই। ব্যবহৃত address টি IO ডিভাইসের সাথে সম্পর্কিত address হবে।

(Refer Slide Time: 29:08)



এটি আদর্শ ARM আর্কিটেকচার, আমি এটির একটি স্ল্যাপশট দেখাতে চেয়েছিলাম। আপনি দেখতে পাচ্ছেন আপনার রেজিস্টার ব্যাংক রয়েছে, এখানে আমাদের গাণিতিক যুক্তি রয়েছে। একটি ডেটা এখানে সরাসরি রেজিস্টার ব্যাংক থেকে আসছে এবং অন্য একটির জন্য আপনি দেখতে পাচ্ছেন যে এখানে একটি ব্যারেল শিফটার বসে আছে।

সুতরাং, অন্যান্য ডেটা স্থানান্তরিত হতে পারে এবং তারপরে ALUতে প্রয়োগ করা যেতে পারে বা এটি ছাড়াও আসতে পারে, কোনও স্থানান্তর ছাড়াই। গুণক এখানে বসে আছে; যখনই আপনার এই গুণক হার্ডওয়্যারটির প্রয়োজন হয় আপনি এটিকে একটি গুণে বাড়িয়ে আনতে পারেন। আরও কিছু নির্দেশাবলীর বৈশিষ্ট্য রয়েছে। এখানে একটি অ্যাড্রেস রেজিস্টার, অ্যাড্রেস ইনক্রিমেন্টার রয়েছে, সুতরাং এগুলি অ্যাড্রেস বাস এবং এখানে মেমরির সাথে ইন্টারফেস করার জন্য ডেটা বাস রয়েছে।

ତବେ ଏର ଭିତରେ ଆକର୍ଷଣୀୟ, ଏଥାନେ ଏକଟି ଗୁଣକ ରଯେଛେ, ଏଥାନେ ଏକଟି ବ୍ୟାରେଲ ଶିଫ୍ଟୋର ରଯେଛେ ଯା ALUର ଆଗେ ବସେ ଆଛେ । ଏଟି କିଛୁ ନିର୍ଦ୍ଦେଶାବଳୀର ବାସ୍ତବାୟନକେ ଥୁବ ଦର୍ଶକ କରେ ତୋଳେ ।

সুতৰাং, এই সঙ্গে আমৱা এই বক্তৃতা শেষে আসছি। আমৱা এই আলোচনাটি পৱন্তী কয়েকটি বক্তৃতাগুলিৰ উপৱে চালিয়ে যাব যেখানে আমৱা ARM নিৰ্দেশিকা সেটটিতে আৱও কিছু অতিৰিক্ত বৈশিষ্ট্য এবং ARM আর্কিটেকচাৰেৰ দিকে নজৱ রাখব।

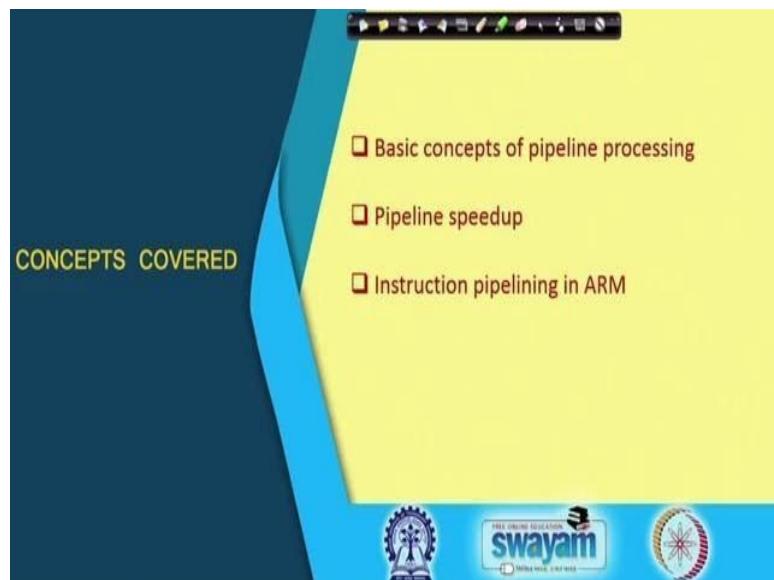
ধন্যবাদ।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 05
Architecture of ARM Microcontroller (Part II)

আমরা ARM মাইক্রোকন্ট্রোলারগুলির সম্পর্কে আলোচনা চালিয়ে যাচ্ছি।

(Refer Slide Time: 00:32)



এই বক্তৃতাটিতে, আমরা মূলত(mainly) ARM প্রসেসরে(processor) উপস্থিতি পাইপলাইন বৈশিষ্ট্যগুলি নিয়ে কথা বলব। যেহেতু পাইপলাইনের ধারণাটি আপনার কাছে নতুন হতে পারে, তাই আমি পাইপলাইনের প্রাথমিক ধারণাটি ব্যাখ্যা করার জন্য কিছুটা সময় ব্যয় করব। তারপরে আমি খুব সংক্ষেপে বলব যে ARM প্রসেসরের পাইপলাইনটি কেমন দেখাচ্ছে এবং এটি থেকে কী কী লাভ হবে বলে আশা করা হচ্ছে, পাইপলাইন কেন ব্যবহার করবেন, আপনার কী কী সুবিধা রয়েছে।

(Refer Slide Time: 01:16)

What is Pipelining?

- A mechanism for overlapped execution of several input sets by partitioning some computation into a set of k sub-computations (or stages).
 - Very nominal increase in the cost of implementation.
 - Very significant speedup (ideally, k).
- Where are pipelining used in a computer system?
 - Instruction execution: Several instructions executed in some sequence.
 - Arithmetic computation: Same operation carried out on several data sets.
 - Memory access: Several memory accesses to consecutive locations are made.

আমরা প্রাথমিক প্রশ্নটি দিয়ে শুরু করি পাইপলাইনিং কী? মূলত পাইপলাইনিং হ'ল ওভারল্যাপেড এক্সিকিউশন(overlapped execution) এর একটি প্রক্রিয়া। আপনি যখন বলেন আমরা কিছু নির্দিষ্ট কাজ সম্পাদন করছি, সাধারণত আমরা একটি কাজ করি, এটি সম্পূর্ণ করি এবং তারপরে পরবর্তী কাজটি শুরু করি। পাইপলাইনে ধারণাটি হ'ল আপনি প্রথম কাজ শেষ করার আগেই আমরা দ্বিতীয় কাজটি শুরু করি, এমনকি দ্বিতীয় কার্য শেষ করার আগেই আমরা তৃতীয় কার্য শুরু করি। সুতরাং, ওভারল্যাপযুক্ত ফ্যাশনে টাঙ্কগুলির বিভিন্ন অংশ সমান্তরালভাবে সম্পাদন করা যেতে পারে।

আমরা এমন একটি গণনা বিভাজন করি যা উপ-গণনা(sub-computation) বা পর্যায়গুলির সংখ্যাতে বহন করা হয়। তারপরে খুব উল্লেখযোগ্য গতি বাড়ানো সম্ভব, আমরা সর্বাধিক k পর্যন্ত দেখতে পাব। তবে খুব মজার বিষয় হ'ল আমরা k এর একটি উপাদান দ্বারা ব্যয় বাড়িয়ে দিচ্ছি না। ঠিক আছে, আমাদের কাছে k সংখ্যক প্রসেসর(processor) থাকতে পারে স্বাভাবিকভাবেই আমরা k গতির গতি বাড়িয়ে দেব। আমরা এটা করছি না। খুব নামমাত্র ব্যয়বৃদ্ধি দ্বারা আমরা k গতিবেগ প্রায় অর্জন করছি।

বর্তমান প্রসঙ্গে আমরা নির্দেশ কার্যকর করার কথা বলছি, তবে গাণিতিক ক্রিয়াকলাপে, ডেটা গেটের স্মৃতি অ্যাক্সেস(memory access of data vector) ইত্যাদির সময় প্রসেসিংয়ের অন্যান্য ডোমেনগুলিতেও পাইপলাইনিং খুব কার্যকরভাবে ব্যবহার করা যেতে পারে তবে বর্তমান প্রসঙ্গে, যেহেতু নির্দেশ কার্যকর করা হয় কেবলমাত্র আমরা যা নিয়ে উদ্বিগ্ন তা হল আমরা অন্যান্য বিষয়গুলির সম্পর্কে খুব বেশি বিশদে যাব না।

(Refer Slide Time: 04:18)

A Real-life Example

- Suppose you have built a machine M that can wash (W), dry (D), and iron (R) clothes, one cloth at a time.
 - Total time required is T .
- As an alternative, we split the machine into three smaller machines M_W , M_D and M_R , which can perform the specific task only.
 - Time required by each of the smaller machines is $T/3$ (say).

For N clothes, time $T_1 = N \cdot T$

For N clothes, time $T_2 = (2 + N) \cdot T/3$

$$\frac{(2+N)T}{3} \approx \left(\frac{NT}{3}\right)$$

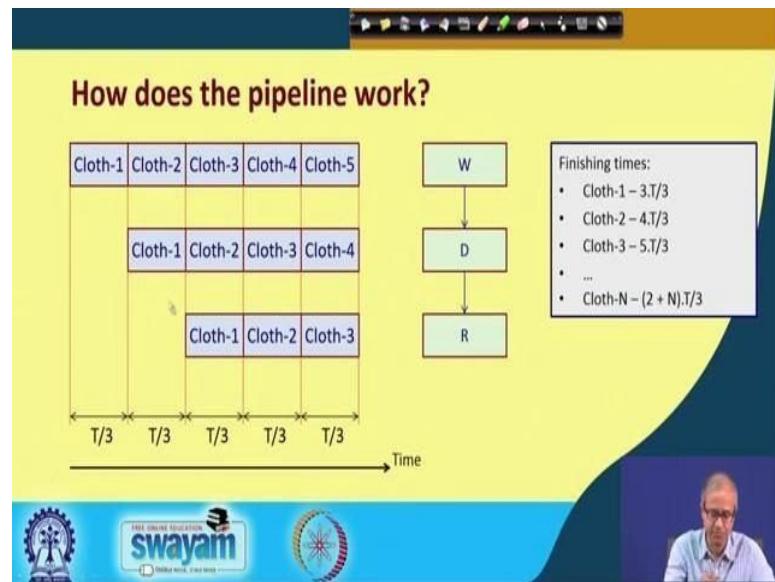
আসুন প্রথমে ধারণাটি বরাবর চেষ্টা করি। আমরা বাস্তব জীবনের উদাহরণ নিই, কম্পিউটারের সাথে এর কোনও যোগসূত্র নেই। ধরা যাক আমরা কিছু কাপড় ধূতে চাই। ধূক্তন আমি এমন একটি মেশিন M তৈরি করেছি যা একের পর এক তিনটি জিনিস ধূয়ে ফেলতে, শুকনো এবং ইস্তি(iron) করতে পারে। আমি মেশিনটিকে একটি কাপড় দেব, এটি এটিকে ধূয়ে ফেলবে, শুকিয়ে যাবে, ইস্তি করে দেবে এবং কাপড়টি ইস্তিযুক্ত আকারে আউটপুট দেয়। আমি ধরে নিই যে পুরো জিনিসের জন্য মোট সময় প্রয়োজন T ।

চিত্রগতভাবে এটিকে এর মতো দেখায়। আমার কাছে আমার মেশিনটি ধোয়া, শুকানো এবং ইস্তি করে, মোট সময় নেয় T । সুতরাং, আমার যদি N টি কাপড় থাকে, তবে মোট সময় প্রয়োজন হবে $N \times T$

এখন ধরে নেওয়া যাক যে একটা খুব জটিল মেশিনের পরিবর্তে যা কিছু করতে পারে, আমাকে এই মেশিনটিকে তিনটি ছোট ছোট টুকরো করে নিতে হবে। তিনটি সাধারণ মেশিন কেবল একটি ধূতে পারে, একটি কেবল শুকিয়ে যেতে পারে, এবং একটিতে কেবল ইস্তি করতে পারে। আসল মেশিনের তুলনায় স্বাভাবিকভাবে এটি তিনবার ব্যয় করা হবে না, এটি সম্ভা হবে। আসুন আরেকটি অনুমান করা যাক: প্রথম দিকে গোটা সময় T ছিল, আসুন আমরা এই সময়ের জন্য প্রতিটি হিসাবে $T/3$, $T/3$ এবং $T/3$ রাখি, যাতে মোট সময়টি এখনও T থাকে।

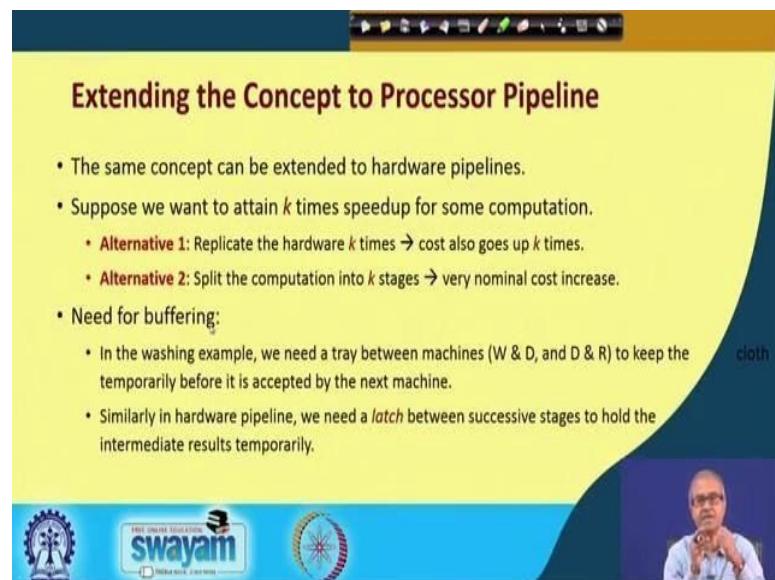
আমরা পরবর্তী ম্লাইডে দেখতে পাব যে এন সংখ্যার পোশাকের প্রক্রিয়াজাতকরণের জন্য মোট সময় প্রয়োজন কেবলমাত্র $(2 + N)T/3$ । যদি এন খুব বড় হয় তবে এই $2T$ উপেক্ষা করা যেতে পারে এবং আপনি এটির কাছাকাছি যেতে পারেন $NT/3$ সুতরাং, আমি একটি 3 সময়ের গতি পেয়েছি, তবে আমি 3 গুণ বেশি অর্থ প্রদান করিনি, আমি সহজ মেশিন ব্যবহার করছি। এটি পাইপলাইনের ধারণ।

(Refer Slide Time: 07:11)



আসুন এই চিত্রটি একটি অ্যানিমেটেড আকারে দেখি। প্রথম কাপড় ধোয়ার জন্য আসে, এটি $T / 3$ সময় লাগবে। এটি ধোয়া দিয়ে শেষ হওয়ার পরে, এই কাপড়টি শুকানোর জন্য যেতে পারে। এবং যখন কাপড় -1 শুকানোর জন্য এসেছে, মেশিন ডাক্ত আবার ফাঁকা। কাপড় 2 দিয়ে শুরু করতে পারেন। পরের ধাপে, কাপড় -1 আসে ইঞ্জিনের জন্য, কাপড় -2 আসে শুকানোর জন্য, কাপড় -3 ওয়াশিংয়ের জন্য এবং আরও অনেক কিছু। সুতরাং, আপনি দেখুন সমস্ত মেশিন সমস্ত ভরে যাওয়ার পরে, প্রতিটি $T / 3$ বারের জন্য একটি কাপড় বের হয়ে আসবে। আগে প্রতিবার একটি কাপড় বের হয়ে আসত টি সময়ে। সুতরাং, আমি কার্যকরভাবে 3 বার গতি পাচ্ছি। পাইপলাইনের পিছনে এটি প্রয়োজনীয় ধারণা।

(Refer Slide Time: 08:44)



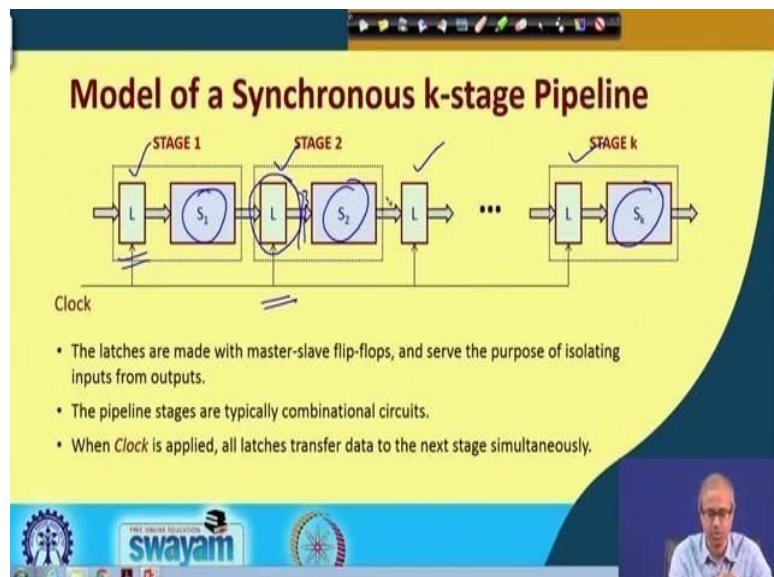
প্রসেসরের পাইপলাইনে ধারণাটি প্রসারিত করে আমরা এখন একটি প্রসেসরের গতি বাড়াতে চাই। মনে করুন, আমার একটি CPU আছে এবং আমি এটিকে কে গুণ আরও দ্রুত করতে চাই। আমার দুটি বিকল্প আছে। আমি এই CPU এর কপি(Copy) কিনতে পারি এবং সমন্বন্ধে কে নির্দেশাবলী চালাতে পারি। বিকল্পভাবে, আমি পাইপলাইনের কে-পর্যায়ে executionকে ভাগ করতে পারি। নির্দেশগুলি

কেবল পাইপলাইন ফাংশানে চালানো হবে যেমন ধোয়া, শুকানো, ইন্সি করার বিষয়টি আমি উল্লেখ করেছি। এছাড়াও আপনি এটিতে কে গতির দ্রুতগতি আশা করতে পারেন।

বিকল্প 1 তে ব্যাও কে সময় বাড়িয়ে দেবে, কারণ আপনাকে এই সিপিইউ-এর কপিগুলি কিনতে হবে। বিকল্প 2 হ'ল গণনা কে K-পর্যায়ে বিভক্ত করা; এখানে আপনি কে দ্বারা ডিভাইসটির ওপর করছেন না, বরং আপনি কে কে টুকরো করে বিভক্ত করছেন যার ফলে ব্যয় খুব নামমাত্র বেড়েছে।

তবে এটি করার জন্য আপনার কিছু বাফারিং দরকার। পোশাকের আগের উদাহরণটি ধরুন। যখন প্রথম কাপড় ধোয়া শেষ হয়, আপনি এটি শুকানোর জন্য দিতে চান। আপনাকে অবশ্যই এটি মাঝখানে কোথাও রেখে দিতে হবে, যাতে আপনি ধোয়ার জন্য পরবর্তী কাপড়টি গ্রহণ করতে পারেন। মেশিনগুলির মধ্যে অবশ্যই একটি বাফার(Buffer) বা ট্রে(Tray) থাকতে হবে, এগুলি এমন কিছু যা বাফারিং প্রয়োজনীয়তা। একটি নির্দেশিকা পাইপলাইনের জন্যও আমাদের পর্যায়েগুলির মধ্যে কিছু রেজিস্টার বা ল্যাচ প্রয়োজন, যা পূর্ববর্তী পর্যায়ে ফলাফল অস্থায়ীভাবে সংরক্ষণ করা হবে, যা পরবর্তী পর্যায়ে প্রক্রিয়াজাতকরণের জন্য ব্যবহৃত হবে। আপনি যদি এই রেজিস্টারগুলি ব্যবহার না করেন, তবে পূর্ববর্তী পর্যায়ে গিয়ে এই মানটি সংশোধন করতে পারেন, সুতরাং পরবর্তী পর্যায়ে এটির কারণে কিছু ভুল গণনা হয়ে যেতে পারে।

(Refer Slide Time: 11:05)



একটি k -স্টেজ পাইপলাইন সাধারণভাবে দেখায়। S_1, S_2 ল্যাচ সহ S_k অবধি সংখ্যা রয়েছে বা প্রতিটি জোড়া স্তরের মধ্যে নিবন্ধন করুন। যখনই কোনও পর্যায় তার গণনা সম্পূর্ণ করে, এটি এই ল্যাচটিতে(latch) ডেটা সংরক্ষণ করবে এবং এটি পরবর্তী প্রতিযোগিতা থেকে ডেটাটিকে তার ইনপুট ল্যাচে(input latch) নিয়ে যাবে। যদি এই ল্যাচটি না থাকে, তবে পরবর্তী গণনা চলাকালীন এই ইনপুটটি পরিবর্তন হতে চলেছে। সুতরাং, এই গণনাটি ভুল হয়ে যেতে পারে। এটা এভাবে কাজ করে।

(Refer Slide Time: 11:51)

Speedup and Efficiency

Some notations:

- τ : clock period of the pipeline
- t_i : time delay of the circuitry in stage S_i
- d_L : delay of a latch

Maximum stage delay $\tau_m = \max \{t_i\}$
 Thus, $\tau = \tau_m + d_L$

Pipeline frequency $f = 1 / \tau$

- If one result is expected to come out of the pipeline every clock cycle, f will represent the maximum throughput of the pipeline.

FREE ONLINE EDUCATION
swayam

আসুন আমরা সাধারণ অর্থে পাইপলাইনের গতি এবং দক্ষতার দ্রুত গণনা করি। এর আগে আমরা ধোয়ার জন্য গণনা দেখিয়েছি। আসুন আমরা ধরে নেই τ হল পাইপলাইনের ক্লক পিরিযড(Clock period), যার অর্থ, প্রতিটি τ সময়ের ডেটা এক স্তর থেকে অন্য পর্যায়ে চলে আসে। এবং t_i হল S_i পর্যায়ে যে সার্কিট রয়েছে তার জন্য সময় বিলম্ব। এবং ল্যাচগুলির বিলম্বটি d_L হবে।

সর্বোচ্চ পর্যায়ের বিলম্ব কী হবে, দেখুন এটি T_1 , এটি T_2 , এটি T_3 । সুতরাং, পাইপলাইনের ধীরতম পর্যায়টি নির্ধারণ করবে যে আমরা সর্বোচ্চ গতিটি কী দিয়ে আমরা ডেটা স্থানান্তর করতে পারি, কারণ এই ধীরতম ধাপটি বাধা হয়ে দাঁড়াবে। সুতরাং, $\max \{t_i\}$, আসুন আমরা এটাকে, সর্বাধিক বিলম্ব বলি। এবং এটিতে আমাদের ল্যাচ বিলম্ব d_L ও যুক্ত করতে হবে। সুতরাং, আপনি clock period τ এর সাথে $+ d_L$ করবেন।

এখন, পাইপলাইন ফ্রিকোয়েন্সি(pipeline frequency) $1 / \tau$ হবে। আপনি যদি প্রতি clock এ একটি ফলাফল বেরিয়ে আসার প্রত্যাশা করেন তবে f হবে পাইপলাইনের সর্বাধিক থ্রুপুট(throughput)। এই অনুমানের সাথে আসুন আমরা দ্রুত গণনা করার চেষ্টা করি।

(Refer Slide Time: 14:02)

T_k = \underbrace{[(k-1) + N] \cdot \tau}_{=} \quad (k-1) \tau \text{ time required to fill the pipeline}
$$1 \text{ result every } \tau \text{ time after that} \rightarrow \text{total } N \cdot \tau$$
- For an equivalent non-pipelined processor (i.e. one stage), the total time is
$$T_1 = N(k \cdot \tau) \quad (\text{ignoring the } \underbrace{\text{latch overheads}}_{})$$
- Speedup of the k-stage pipeline over equivalent non-pipelined processor:
$$S_k = \frac{T_1}{T_k} = \frac{N \cdot k \cdot \tau}{k \cdot \tau + (N-1) \cdot \tau} = \frac{N \cdot k}{k + (N-1)}$$

A box on the right says 'As N → ∞, S_k → k'. The slide has a yellow header and a blue footer with the Swayam logo."/>

আমরা ডেটার N সেটগুলি প্রসেস করার জন্য মোট সময় গণনা করি। তাও আমার ক্লক পিরিয়ড(Clock Period)। পাইপলাইনটি পূরণ করার জন্য এখন $K - 1$ টি ক্লক প্রয়োজন। K -পর্যায় রয়েছে, সুতরাং আমার $K-1$ টি ক্লক(clock) দরকার এমন পর্যায়ে পৌছাতে যেখানে সমস্ত K -পর্যায় কোনও কিছুর উপর কাজ করছে। এর পরে প্রতিবার সময় একটি নতুন ফলাফল উত্পন্ন হবে। সুতরাং, $(k - 1) \times \tau$ পাইপটি পূরণ করার প্রাথমিক সময় হবে এবং তারপরে আউটপুটগুলি উত্পন্ন করার জন্য এই $N \times \tau$ । এটি N ডাটা সেটগুলি প্রসেস করার মোট সময় হবে।

এখন, যদি আমাদের কাছে সম্পরিমাণ নন-পাইপলাইনযুক্ত প্রসেসর(non pipelined processor) থাকে তবে আপনি যদি সময়ের জন্য ল্যাচ বিলম্বকে উপেক্ষা করেন তবে মোট সময়টি $N \times k \times \tau$ হিসাবে অনুমান করা যায়। সুতরাং, একটি পাইপলাইনে আমরা যে স্পিডআপ S_k পাছি তা হ'ল এটি। এন খুব বড় হয়ে ওঠার সাথে সাথে এই S_k , k হচ্ছে। সুতরাং, আপনি পাইপলাইনটি প্রক্রিয়াজাত করছেন এমন প্রচুর সংখ্যক ডেটার জন্য, আপনার গতিপথ k পর্যায়ের সংখ্যার কাছাকাছি থাকবে। এটি একটি গুরুত্বপূর্ণ ফলাফল।

(Refer Slide Time: 16:21)

• Pipeline efficiency:

- How close is the performance to its ideal value?

$$E_k = \frac{S_k}{k} = \frac{N}{k + (N - 1)}$$

• Pipeline throughput:

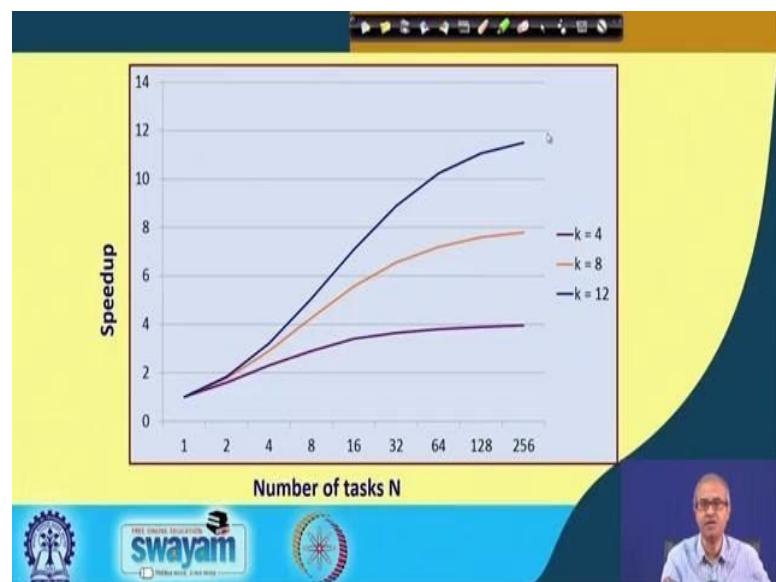
- Number of operations completed per unit time.

$$H_k = \frac{N}{T_k} = \frac{N}{[k + (N - 1)] \cdot \tau}$$


এখন, আমরা পাইপলাইন দক্ষতা সংজ্ঞায়িত আরও একটি শব্দ আছে; পারফরম্যান্স আদর্শ মানের কাছাকাছি কর্ত। ঠিক আছে, এই S_k টি আমরা কেবল গণনা করেছি, তাই যখন পাইপলাইন সর্বাধিক দক্ষতার সাথে পরিচালিত হয়। যদি আমি এটি কে দ্বারা বিভক্ত করি, কে বাতিল করতে পারে, তাই আমি একটি ফ্যাক্টর পাই। এটি আসল পাইপলাইন দক্ষতা হিসাবে সংজ্ঞায়িত করতে পারি। সুতরাং, এটি কখনই 100% হবে না, সম্ভবত এটি 90% দক্ষতায় কাজ করছে।

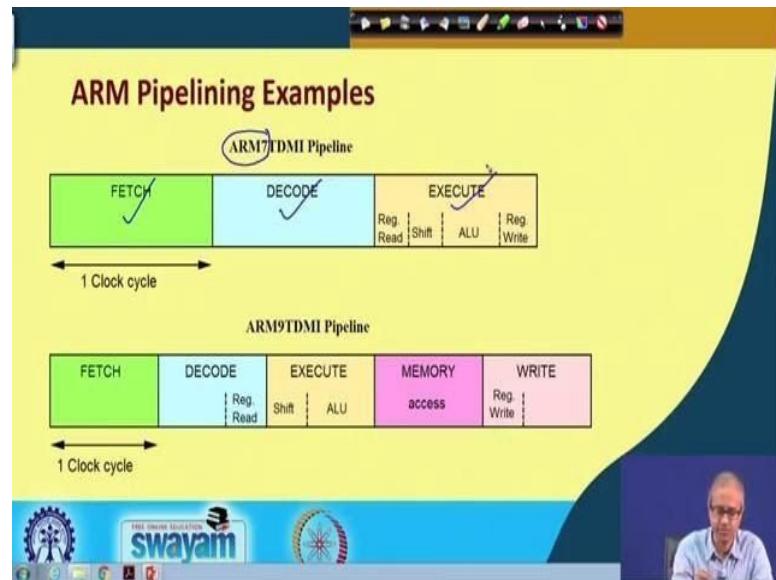
এবং অন্য একটি শব্দ যা অবশ্যই বিদ্যমান, এই প্রেক্ষাপটে তেমন গুরুত্বপূর্ণ নয় তাকে পাইপলাইন থ্রুপুট(throughput বলে; ইউনিট সময় অনুসারে সম্পূর্ণ অপারেশন সংখ্যা। মোট অপারেশন সংখ্যা N । আপনি যদি এটি ভাগ করেন তবে আপনি একটি অভিব্যক্তি পান, এটি পাইপলাইন থ্রুপুট(Pipeline throughput)।

(Refer Slide Time: 17:48)



এটিতে আমি খুব সাধারণ প্লটটি প্রদর্শন করছি, কে এর বিভিন্ন মানের জন্য কার্য বনাম গতিসম্পন্ন নম্বর N। আসুন আমরা K = 4 বলি; আপনি দেখেন যে কাজের সংখ্যা বৃদ্ধি পায়, গতিবেগ বৃদ্ধি পায় এবং স্তরগুলি খুব কাছাকাছি হয়ে যায়। K = 8 এর জন্য, এটি স্তরটি 8 এর খুব কাছাকাছি চলে যায়; ইত্যাদি। এখানে আমি 256 পর্যন্ত দেখিয়েছি। আসলে কী হচ্ছে আপনি কিছুটা ধারণা পাচ্ছেন।

(Refer Slide Time: 18:35)



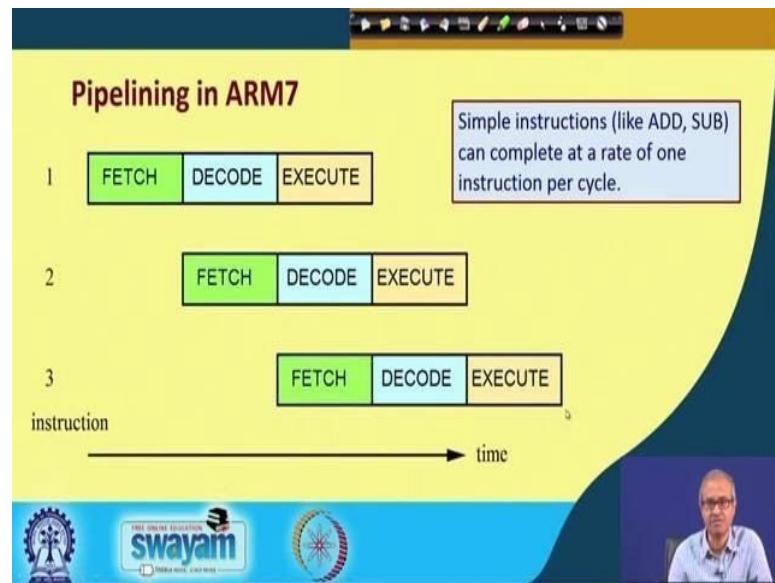
এখন, ARM-এ এসে আমি বিশদে যাচ্ছি না কারণ এটি এমন কোর্স নয় যেখানে আমি কম্পিউটার আর্কিটেকচার শেখাচ্ছি বরং আমি আপনাকে বলার চেষ্টা করছি যে ARM নির্দেশিকা পাইপলাইন ব্যবহার করে। আমার কাছে যদি K-স্টেজ পাইপলাইন থাকে তবে আমি কে গতির গতিবেগ আশা করতে পারি।

ARM 7 আর্কিটেকচারে এটি একটি বিশেষ প্রসেসর টিডিএমআই। তিনটি পর্যায় রয়েছে, fetch, decode এবং execute। অন্য সব কিছু ঠিকঠাক থাকলে, নির্দেশ কার্যকর করার ক্ষেত্রে আমরা প্রায় 3 গুণ স্পিডআপ(speed-up) পাব বলে আশা করা হচ্ছে।

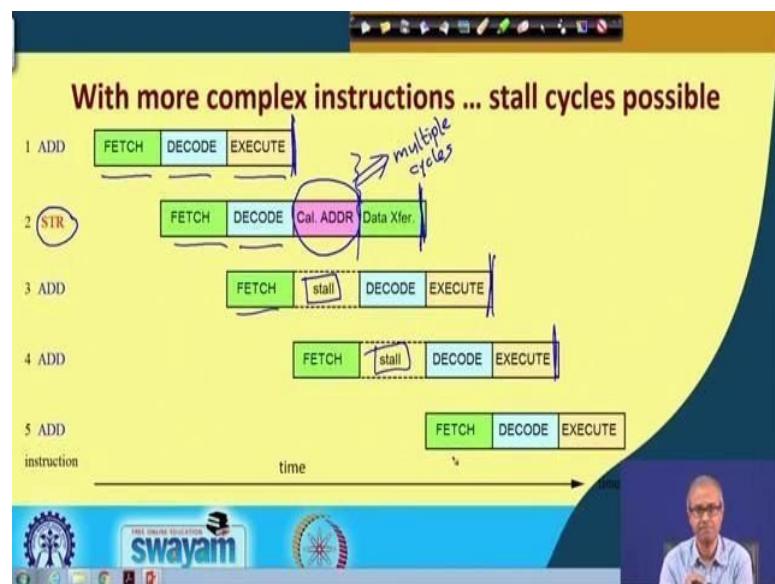
একইভাবে ARM 9 এর 5 স্টেজের পাইপলাইন, fetch, decode, execute, memory, write রয়েছে। আমরা কিছু ছোট জিনিস দেখতে পারি। ডিকোডের মধ্যে, নিবন্ধকের মানগুলি যা যা রেজিস্টারগুলির প্রয়োজন হয় তা পড়তে হয়। execute এর সময় ব্যারেল শিফটারটিও (Barrel Shifter) কাজ করছে, ALU অপারেশনগুলি চালিয়েছে, মেমরির অ্যাক্সেস(Memory access) এখানে চালিত হয়; লেখার সময়, ফলাফলগুলি রেজিস্টার ব্যাঙ্কে(Register bank) ফিরে লিখিত হয়, তাই এটি এখানে করা হয়। আর ARM 7 এর জন্য এগুলি সবই কার্যকর করা হয়েছিল। রেজিস্টার রিড(Register read), শিফট(Shift), এলইউ(ALU), রেজিস্টার রাইটিং(Register writing) সবকিছুই এক্সিকিউটে সম্পন্ন হয়েছিল। তবে ARM 9-এ, আপনি পাইপলাইনটিকে আরও বিস্তৃত এবং আরও flexible করে তুলছেন।

সুতরাং, এখানে স্পিডআপ(speed-up) সর্বোচ্চ 5 হতে পারে।

(Refer Slide Time: 20:33)



(Refer Slide Time: 21:15)



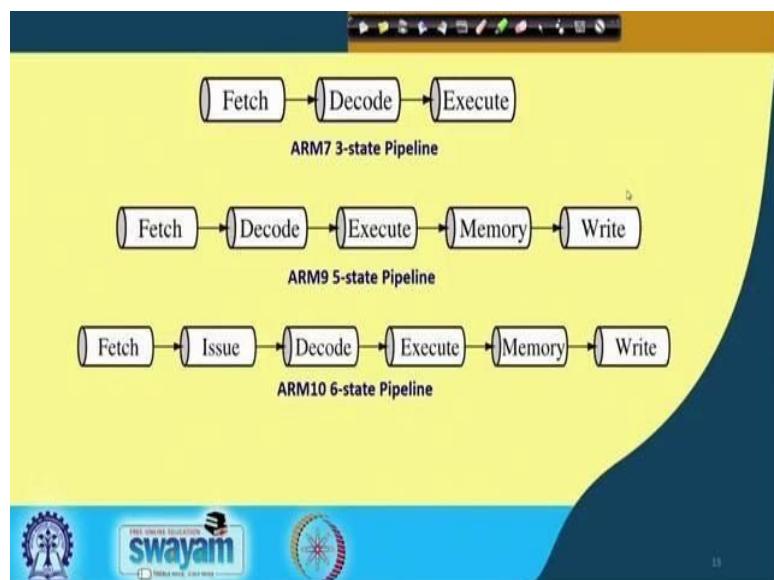
আমি এখানে কেবল একটি বিষয় উল্লেখ করতে চাই তা হ'ল কে-এর এই গতিপথটি আমি একটি আদর্শ গতি বলছি। পাইপলাইনটি যখন তার পুরো গতিবেগে পরিচালিত হয় তখন আপনি যে স্পিডআপটি পেতে পারেন তা কিন্তু কখনও কখনও কোনও কারণে আপনি পাইপলাইন পুরো গতিতে পরিচালনা করতে পারবেন না। এই ক্ষেত্রে, স্পিডআপটি কে-এর চেয়ে কম হয়ে যাবে। আমি একটি উদাহরণ দিচ্ছি। ধরা যাক, কিছু নির্দেশাবলী কার্যকর করা হয়েছে এবং আসুন আমরা বলি যে এগুলি সমস্ত ADD নির্দেশনা রয়েছে এবং একটি জটিল মাল্টি রেজিস্টার স্টোর নির্দেশ রয়েছে। সুতরাং, এটির জন্য একাধিক clock cycle প্রয়োজন।

ধারণাটি হ'ল সাধারণত এক ক্লকের মধ্যে সবকিছু শেষ হয়। তবে SRT নির্দেশের জন্য কী হতে পারে যে এই গোলাপী রঙের বাক্সটি আসলে একাধিক চক্রের প্রয়োজন হতে পারে। ওটার মানে কি? একাধিক চক্রের অর্থ এখানে আপনি পরবর্তী নির্দেশটি ডিকোড করতে পারবেন না, যদি না এই নির্বাহ শেষ হয় আপনি এটিকে ডিকোড(decode) করতে দেনপারবেন না। সুতরাং, এখানে কিছুটা বিলম্ব হবে কারণ

এটি একাধিক চক্রের প্রয়োজন। এই ধরনের বিলম্ব স্টল হিসাবে উল্লেখ করা হয়; আমরা তাদের স্টল চক্র(stall cycle) বলি।

স্টল চক্র মানে কিছু চক্র নষ্ট হয়। আপনি এখানে দেখুন প্রথম নির্দেশ সমাপ্ত হয়েছে; দ্বিতীয় নির্দেশ এখানে শেষ হয়েছিল, তৃতীয় নির্দেশ এখানে, চতুর্থ নির্দেশ এখানে। তবে এই বিলম্বের কারণে এই নির্দেশটি এখানে শেষ করার কথা ছিল তবে এটি বিলম্বিত হয়েছিল। কেবল এটিই নয়, পরবর্তী সমস্ত নির্দেশাবলী বিলম্বিত হয়েছিল এবং এর মধ্যে এই জাতীয় অনেক নির্দেশাবলী থাকতে পারে। সুতরাং, এই জাতীয় প্রতিটি নির্দেশের জন্য কিছু স্টল চক্র প্রবেশ করানো হবে। এবং একবার কোনও স্টল(stall) আসার পরে এই স্টলটি পরবর্তী নির্দেশাবলী দ্বারা চালিত হবে যতক্ষণ না সেই নির্দেশ পাইপটি না বের হয়। এই জাতীয় ক্ষেত্রে পাইপলাইনের সর্বাধিক পরিচালন গতি ধীর করতে পারে।

(Refer Slide Time: 24:07)

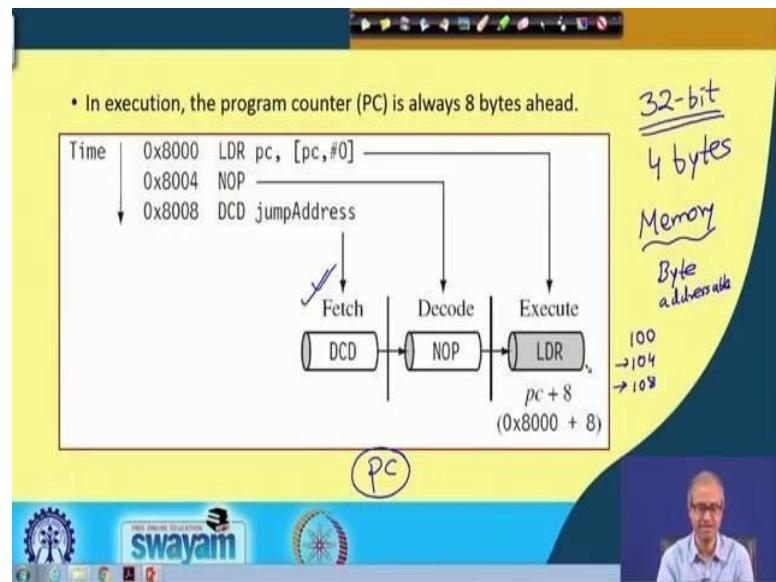


আমরা ARM 7, ARM 9 এবং ARM 10 পাইপলাইনের পার্থির চেয়ের দৃষ্টি দিয়েছি।

পাইপলাইন বিপত্তি নামে কিছু রয়েছে। আপনি পরবর্তী নির্দেশাবলী feed করতে পারেন কিনা তা এখানে ডেটা নির্ভরতা থাকতে পারে। এখানে অনেকগুলো আর্কিটেকচারাল সমস্যা রয়েছে যা পাইপলাইনটিকে তার সর্বোচ্চ গতিতে পরিচালিত করতে বাধা দিতে পারে।

স্টল নির্দেশাবলী এর কারণে ঘটতে পারে, একটি উদাহরণ একটি জটিল নির্দেশনার কারণে আমি দিয়েছিলাম তবে এর অন্যান্য কারণও থাকতে পারে। ডেটা হ্যাজারস নামে পরিস্থিতি রয়েছে, কাঠামোগত বিপত্তি হতে পারে, নিয়ন্ত্রণের ঝুঁকিও থাকতে পারে। বিভিন্ন ক্রমানুসারে পরিচালিত ক্রিয়াকলাপগুলির কারণে, এবং কিছু পদক্ষেপের মতো জাম্প(JUMP) এবং ব্রাঞ্ছগুলির(branch) জন্য আপনাকে স্টল চক্র সঞ্চাবেশ করতে হতে পারে। এই সমস্তগুলি পাইপলাইন সর্বাধিক ক্লকের ক্রিকোয়েন্সিতে পরিচালনা করতে বাধা দেয়।

(Refer Slide Time: 25:39)



আর একটি জিনিস বলি, এই ARM 7 ধরণের আর্কিটেকচারে একটি 3-স্টেজ(stage) পাইপলাইন রয়েছে। আসুন আমরা যখন বলি যে কোনও নির্দেশ কার্য সম্পাদনকারী পর্যায়ে পৌছে যায়, তখন একটি জিনিস আপনার মনে রাখবেন আমি আপনাকে সমস্ত নির্দেশাবলী 32-bit নির্দেশাবলী বলেছিলাম। এবং আপনার memory, byte addressable। সূতরাং, যদি প্রথম নির্দেশটি মেমোরি লোকেশন 100 এ সঞ্চিত থাকে তবে পরবর্তী নির্দেশটি মেমোরি অবস্থান 104 এ সংরক্ষণ করা হবে, পরবর্তী অবস্থানটি 108 অবস্থানের মধ্যে সংরক্ষণ করা হবে, কারণ প্রতিটি নির্দেশের জন্য 4 byte প্রয়োজন হবে।

মূল বক্তব্যটি হ'ল কিছু নির্দেশ কার্যকর করা হলে PC সর্বদা 8 byte এগিয়ে থাকবে। আপনি এটিতে 4 যুক্ত করুন কারণ আপনি এটি আনবেন। প্রতিটি নির্দেশ মেমোরি ঠিকানাতে 4 যোগ করা হবে। এবং PC হ'ল একটি বিশেষ রেজিস্ট্রার যা সর্বদা পরবর্তী নির্দেশাবলীর ঠিকানা সংগ্রহ করার জন্য সঞ্চয় করে। আপনি যখন এই নির্দেশনাটি আলচেন, এটি বর্তমান নির্দেশের প্লাস 8 এর PC হবে, কারণ বর্তমান নির্দেশটি এখানে রয়েছে। আমরা যদি এতে 4 যোগ করি তবে আমরা পরবর্তী নির্দেশনাটি পাব; যদি আমরা এটিতে 8 যোগ করি তবে আমরা পরবর্তী নির্দেশিকার পরবর্তী পাবো। এখানে আপনি সর্বদা পরবর্তী নির্দেশের পরবর্তী সন্ধান করছেন কারণ তিনটি ধাপ রয়েছে যার জন্য PC সর্বদা 8 বাইট এগিয়ে থাকে।

কারণ আপনি যখন এখানে ইতিমধ্যে দু'বার বর্ধিত PC চালাচ্ছেন এটি সর্বদা 8 byte এগিয়ে থাকে, যাতে আপনি আনার সময় সেই অনুযায়ী সামঞ্জস্য করতে পারেন এবং সেই অনুযায়ী আনতে হবে। এটির সাথে আমরা এই বক্তৃতার শেষে এসেছি। পরবর্তী বক্তৃতায়, আমরা সংগঠনটি, বিভিন্ন execution এর পদ্ধতি এবং এই জাতীয় নিবন্ধনুক্তির বিষয়ে ARM এর কিছু অন্য বৈশিষ্ট্যগুলি দেখব। এটি আমরা পরবর্তী বক্তৃতায় আলোচনা করব।

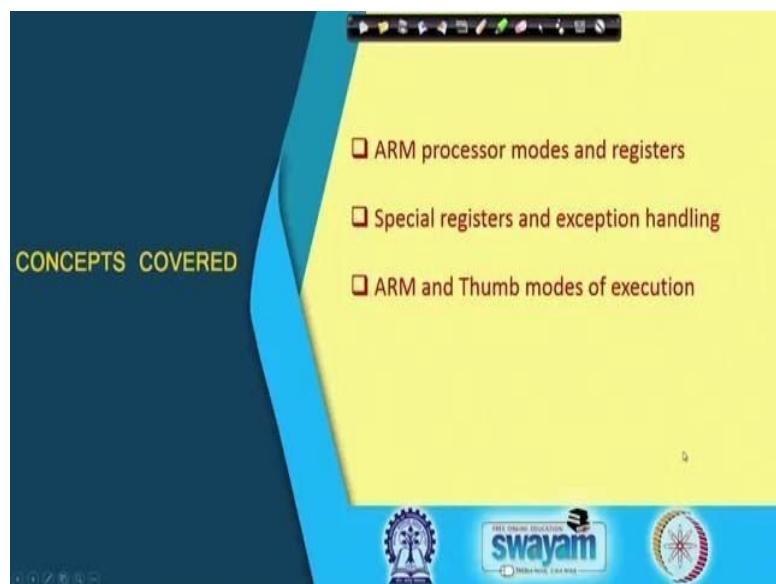
ধন্যবাদ।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 06
Architecture of ARM Microcontroller (Part III)

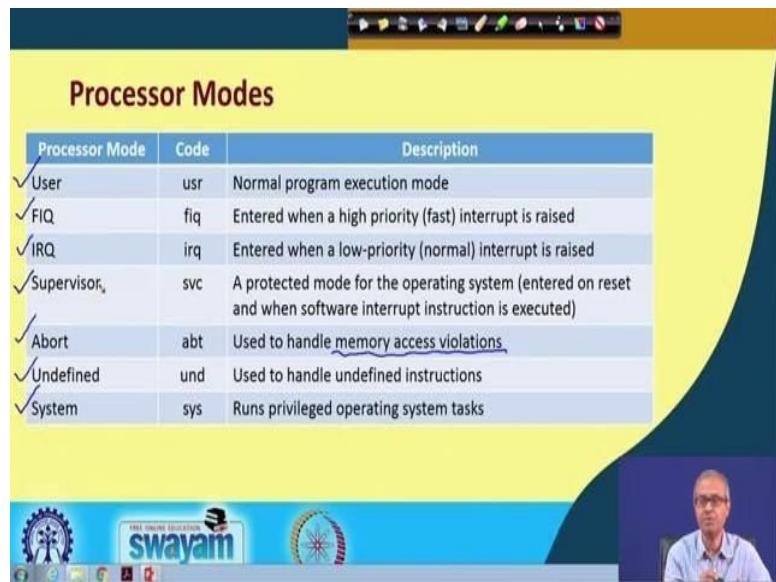
আমরা আমাদের আলোচনা চালিয়ে যাচ্ছি। এই বক্তৃতায় আমরা ARM মাইক্রোকন্ট্রোলার ARCHITECTURE সম্পর্কে বলব, এর তৃতীয় অংশ।

(Refer Slide Time: 00:30)



এই বক্তৃতাটিতে আমরা মূলত ARM আর্কিটেকচারে প্রসেসরের মোড(Processor Mode) এবং REGISTERগুলি সম্পর্কে কথা বলব। কিছু special REGISTER রয়েছে, আমরা এই special REGISTER এর প্রয়োজনীয়তা এবং ভূমিকাগুলি বলার চেষ্টা করব; ব্যতিক্রম হ্যান্ডলিং সম্পর্কে কিছু আছে এবং এটি সম্পর্কে খুব সংক্ষেপে EXECUTION কার্যকর করার থাম্ব মোড বলবো। আবার এর খুব বেশি বিশদে যাব না কারণ এই কোস্টি প্রাথমিকভাবে বিভিন্ন সিস্টেম ডিজাইনের demonstration এর জন্য ব্যবহৃত হয়েছিল, তবে architecture সংক্রান্ত কিছু প্রাথমিক ধারণা শেখা আপনাকে সর্বদা আরও ভাল ডিজাইনার হওয়ার ক্ষেত্রে সহায়তা করবে। আর্কিটেকচারাল ধারণাগুলিতে এমবেডেড সিস্টেম(Embedded System) ডিজাইনের প্রাথমিক ব্যাকগ্রাউন্ডের কয়েকটি আপনাকে দেওয়ার চেষ্টা করার মূল উদ্দেশ্য এটি।

(Refer Slide Time: 01:33)



Processor Mode	Code	Description
✓ User	usr	Normal program execution mode
✓ FIQ	fiq	Entered when a high priority (fast) interrupt is raised
✓ IRQ	irq	Entered when a low-priority (normal) interrupt is raised
✓ Supervisor	svc	A protected mode for the operating system (entered on reset and when software interrupt instruction is executed)
✓ Abort	abt	Used to handle <u>memory access violations</u>
✓ Undefined	und	Used to handle undefined instructions
✓ System	sys	Runs privileged operating system tasks

প্রসেসর মোড দিয়ে শুরু করা যাক। নির্ধারিত সময়ে কার্যকর করার সময় ARM প্রসেসর, 7 টি মোডের মধ্যে একটি হতে পারে। এই টেবিলটি এই 7 টি প্রসেসরের মোডের সংক্ষিপ্তসার জানায়।

কোনও প্রোগ্রাম কার্যকর করার সময় ব্যবহারকারী মোডটি সাধারণ মোড। যখন কোনও প্রোগ্রাম সাধারণত চালিত হয়, আমরা বলি যে সিস্টেমটি ব্যবহারকারী মোডে রয়েছে। এই হল ব্যবহারকারী মোড এর সংক্ষিপ্ত বিবরণ। প্রসেসরগুলিতে আপনি এখন জানেন, আমাদের বাহ্যিক ডিভাইসগুলি থেকে INTERRUPT আসতে পারে। যদি এটি আসে, প্রোগ্রামটি কার্যকর করছে যা স্থগিত করা হবে, আমাদের কিছু INTERRUPT হ্যান্ডলিং(Handling) কৃটিনে যাবো, সেই কৃটিন চালাতে হবে এবং তারপরে আবার ফিরে এসে INTERRUPT প্রোগ্রামটি আবার শুরু করতে হবে। এআরএমে দুটি ভিন্ন স্তরের বিপ্লিত interrupt processing অনুমোদিত, একটিকে উচ্চ PRIORITY বলা হয় বা অন্যটিকে সাধারণ PRIORITY বলা হয়।

এই FIQ উচ্চ প্রায়োরিটি মোড(PRIORITY Mode)। যখন উচ্চ প্রায়োরিটি (HIGH PRIORITY) র ইন্টারেটা (INTERRUPT) সক্রিয় হয় এবং স্বীকৃত হয় তখন FIQ মোডে প্রবেশ করা হয়। লক্ষ্য করার বিষয়টি হল যখন কোনও উচ্চ PRIORITY র INTERRUPTকে প্রক্রিয়া করা হচ্ছে, যদি এর মধ্যে কিছু কম PRIORITY INTERRUPT আসে; তাদের স্বীকৃতি দেওয়া হবে না, তাদের এড়ানো হবে। সুতরাং, উচ্চ PRIORITY INTERRUPTকে নিম্ন PRIORITY র তুলনায় প্রকৃত উচ্চ PRIORITY দেওয়া হবে।

আর IRQ হল low priority বা normal priority interrupt।

একটি সুপারভাইজারি মোড রয়েছে যা বেশিরভাগ আধুনিক প্রসেসরের রয়েছে, কিছু INSTRUCTION রয়েছে যেমন সুপারভাইজারি কল বা ট্র্যাপ বা SWI সফটওয়্যার �INTERRUPT

দেয়। বিভিন্ন নাম রয়েছে, তবে এই INSTRUCTIONগুলির উদ্দেশ্য হল এই নির্দেশ অপারেটিং সিস্টেমে নিয়ন্ত্রণ স্থানান্তর করতে দেয়। ঠিক আছে এমন একটি কম্পিউটারে যেখানে একটি অপারেটিং সিস্টেম রয়েছে, এই নির্দেশাবলী প্রসেসর মোড়টিকে ব্যবহারকারীর মোড থেকে সুপারভাইজার মোডে পরিবর্তন করার অনুমতি দেয় এবং তারপরে একটি subroutine call control সুপারভাইজার বা অপারেটিং সিস্টেমে ঝাঁপ দেবে।

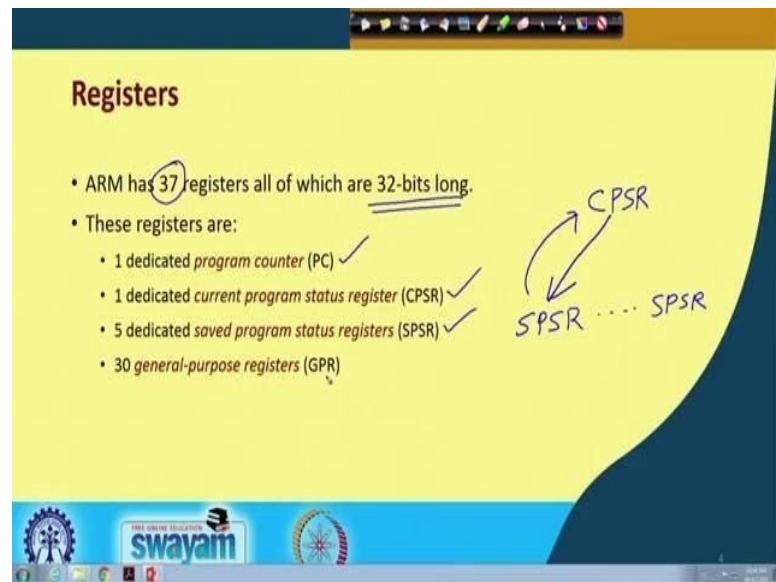
এখন, SUPERVISORY মোডটি svc। এটি একটি সুরক্ষিত মোড এবং আপনার বাস্তবায়নে যখন কোনও অপারেটিং সিস্টেম থাকে তখনই এই মোডটি প্রয়োজন। সমস্ত এলেড থাকা সিস্টেম অ্যাপ্লিকেশনটিতে আপনার এটির প্রয়োজন হবে না, তবে এমন সিস্টেমে যেখানে সত্যিই একটি অপারেটিং সিস্টেম রয়েছে এবং আপনার এই ব্যবস্থা কার্যকর করার মোডটি রাখতে আপনার কিছু সুরক্ষা দরকার।

আপনি যে memory protection রাখতে চান সেই ক্ষেত্রে ABORT হল প্রিষ্ঠিক মোড। আপনি দেখতে চান যে যখন কোনও প্রোগ্রাম কার্যকর হচ্ছে, আপনি কেবল মেমরির এই অঞ্চলটি অ্যাক্সেস করবেন। যদি দুর্ঘটনাক্রমে আপনার প্রোগ্রামটি অনুমোদিত সীমা ছাড়িয়ে কোনও স্থূতি অবস্থান অ্যাক্সেস(region of memory access) করার চেষ্টা করে, তবে এই ABORT বিস্তৃত হওয়া উত্পন্ন হবে এবং সংশ্লিষ্ট প্রসেসরের মোডটিকে ABORT মোড বলে; মেমরি অ্যাক্সেস লঙ্ঘন পরিচালনা করার জন্য প্রসেসরটি abort মোডে যায়।

এবং অপরিজ্ঞাত আসলে এমন কিছু যা ভবিষ্যতের বিস্তারের জন্য রাখা হয়। ভাল কিছু নির্দেশ opcode সংজ্ঞায়িত করা হয় নি, তাই তারা অপরিজ্ঞাত। সুতরাং, যদি ভুল করে আপনি এমন একটি নির্দেশনা কার্যকর করার চেষ্টা করছেন যেটির opcode সংজ্ঞাত, আপনি একটি সংজ্ঞাত বিস্তৃত পেয়েছেন এবং প্রসেসরের মোড সংজ্ঞাত স্থিতি und স্ট্যাটাসএ চলে যাবে।

এবং এমন উদাহরণ রয়েছে যেখানে আপনি কিছু সুবিধাযুক্ত অপারেটিং সিস্টেমের কাজ চালাতে চান এবং তার জন্য আপনি সাধারণত এই সিস্টেম মোডে যান। সিস্টেম মোড এবং SUPERVISORY মোড খুব বেশি সম্পর্কিত। আমি এর বিশদে যাচ্ছি না কারণ এমবেডেড সিস্টেম ডিজাইনের জন্য এগুলি সত্যিই প্রয়োজন হয় না। ARM এই সমস্ত মোডকে সমর্থন করে। ARM এর architecture বৈশিষ্ট্যের দিক থেকে যথেষ্ট FLEXIBLE। আমি পূর্বে উল্লিখিত হিসাবে এটি সাধারণ উদ্দেশ্য REGISTER এর একটি বড় সেট আছে।

(Refer Slide Time: 07:25)



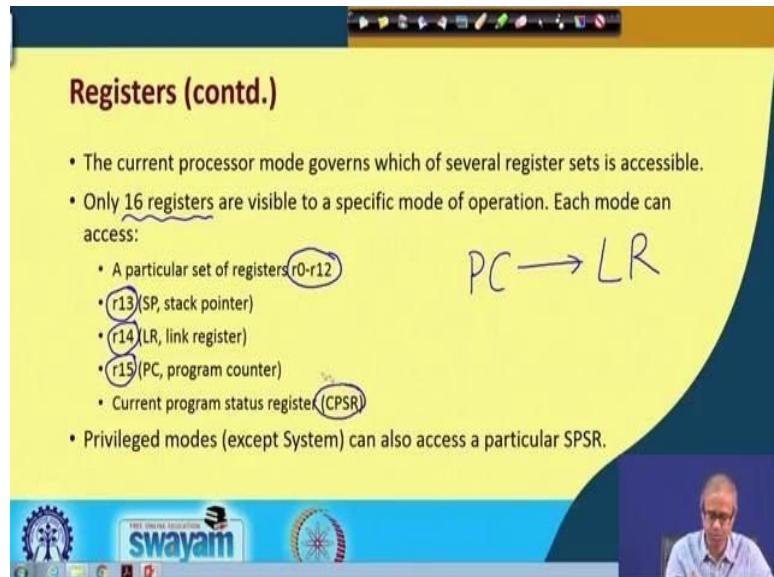
মোট 37 টি রেজিস্ট্রার রয়েছে। এই সমস্ত 32 bit দীর্ঘ। REGISTER গুলি সম্পর্কে কথা বলছি, এখানে 7 টি REGISTER রয়েছে যার সুনির্দিষ্ট উদ্দেশ্য রয়েছে; একটি PC। তারপরে CPSR রয়েছে। আপনারা যারা 8085 এর মতো কিছু মাইক্রোপ্রসেসরের সাথে পরিচিত তাদের জন্য বলি যে তারা জনবেনকভিশানাল ফ্ল্যাগ(CONDITIONAL FLAG) নামে কিছু আছে। যখনই কিছু নির্দেশাবলী কার্যকর করা হয় CONDITIONAL FLAG সেট করা বা পুনরায় সেট করা হয়; এখানে শূন্য FLAG, বহনকারী FLAG, overflow FLAG ইত্যাদি রয়েছে। এটি একটি FLAG REGISTER-এর মতো। এটিতে বেশ কয়েকটি FLAG রয়েছে যা প্রসেসরের মোড বা নির্দেশনা কার্যকরকরণের উপর নির্ভর করে সেট করা থাকে।

এবং সেখানে 5 টি সেভড প্রোগ্রাম স্ট্যাটাস REGISTER রয়েছে। এখন এই ধারণাটি আকর্ষণীয়। একটি CPSR রয়েছে এবং বেশ কয়েকটি SPSR রয়েছে। আপনি একটি প্রচলিত প্রসেসরে দেখতে পাবেন যখনই কোনও INTERRUPT আসে, আমরা সাধারণত কী করব? আমরা স্ট্যাকের সমস্ত REGISTER সংরক্ষণ করি যা status REGISTER এবং FLAG গুলি অন্তর্ভুক্ত করতে পারে, INTERRUPT প্রাপ্ত হ্যান্ডলারের কাছে যেতে পারে, সমস্ত কিছু শেষ করতে পারে, ফিরে আসি, সমস্ত REGISTER এবং স্থিতি REGISTER পুনরুদ্ধার করে এবং পুনরায় প্রয়োগ শুরু করে, তবে ARM এ স্ট্যাকের জন্য কোনও সমর্থন নেই, স্ট্যাকে(stack) বা স্ট্যাক থেকে পুশ বা পপ করার কোনও নির্দেশ নেই।

ধরুন যে কোনও প্রোগ্রাম চলমান স্ট্যাটাসটি CPSR-এ সংরক্ষিত আছে, সেখানে একটি INTERRUPT এসেছে। CPSR বিষয়বস্তু একটি SPSR অনুলিপি করা হবে, তারপরে INTERRUPT প্রাপ্ত হ্যান্ডলার চালিত হবে, ফিরে আসার আগে SPSR এর বিষয়বস্তু CPSR-এ ফিরিয়ে দেওয়া হবে। যেহেতু কোনও স্ট্যাক নেই সেহেতু CPSR থেকে মানটি পুনরুদ্ধার করা এবং তারপরে আবার এটিকে পিছনে সরিয়ে নেওয়া ব্যবহারকারীদের দায়িত্ব। এবং অবশ্যই, সেখানে 30 টি

সাধারণ উদ্দেশ্যে general purpose REGISTER রয়েছে; এগুলির নাম সাধারণত r0 থেকে r29 করা হয়।

(Refer Slide Time: 10:39)



Registers (contd.)

- The current processor mode governs which of several register sets is accessible.
- Only 16 registers are visible to a specific mode of operation. Each mode can access:
 - A particular set of registers (r0-r12)
 - r13 (SP, stack pointer)
 - r14 (LR, link register)
 - r15 (PC, program counter)
 - Current program status register (CPSR)
- Privileged modes (except System) can also access a particular SPSR.

PC → LR

এখন, প্রসেসরের মোডের উপর নির্ভর করে আপনি নির্ভর করতে পারেন যে এর মধ্যে কতগুলি REGISTER আপনি অ্যাক্সেস করতে পারবেন। 16 টি REGISTER সাধারণত একটি নির্দিষ্ট অপারেশন মোডে দৃশ্যমান; এই 16 টি REGISTER নীচে রয়েছে।

r 0 থেকে r12 এ 13 টি general purpose REGISTER রয়েছে। কোনও স্ট্যাক নেই, তবে r13 স্ট্যাক পয়েন্টার হিসাবে চিহ্নিত করা হয়েছে। আপনি যদি স্ট্যাকের দ্বারা একটি স্ট্যাক নিজেই প্রয়োগ করতে চান তবে আপনি r13 ব্যবহার করতে পারেন। আর r14 link REGISTER। প্রচলিত প্রসেসরের জন্য subroutine call instruction এর জন্য, PCর মান স্ট্যাকের মধ্যে push করা হয় তবে এখানে কোনও স্ট্যাক নেই। সুতরাং, লিঙ্ক REGISTER নামে একটি বিশেষ REGISTER রয়েছে, PCর মান লিঙ্ক রেজিস্টারে(link register) অনুলিপি করা হয়। যখনই সাবরুটিন কল(sub routine) আসে এবং আপনি যখন সাবরুটিন থেকে ফিরে আসছেন তখন যা কিছু হয় তারপরেই লিঙ্ক REGISTERটি PCতে অনুলিপি করা হয়।

PC, r15 হিসাবে অ্যাক্সেস করা হয়েছে। এবং তারপরে আমরা CPSR করব। আমি পরে বিভিন্ন REGISTERগুলির সাথে একটি চিত্র প্রদর্শন করব, কীভাবে তারা বিভিন্ন মোডে বিদ্যমান।

(Refer Slide Time: 12:29)

General-purpose Registers

- 6 data types are supported (signed/unsigned)
 - 8-bit byte, 16-bit half-word, 32-bit word
- All ARM operations are 32-bit.
 - Shorter data types are only supported by data transfer operations.

32-bit word

8-bit Byte

16-bit Half word

32-bit word

swayam

FREE ONLINE EDUCATION

সাধারণ উদ্দেশ্যে REGISTER ভূক্তির ক্ষেত্রে যেখানে আপনি কিছু ডেটা সংয় করার আশা করছেন, আমি আপনাকে বলেছি যে REGISTER-গুলি সমস্ত 32 bit আকারের। তবে সমর্থিত ক্রিয়াকলাপগুলির ক্ষেত্রে, আপনার কাছে 8-bit ক্রিয়াকলাপ, 16-bit বা অর্ধ word বা পুরো 32-bit word থাকতে পারে।

আপনি যখন অর্ধ WORD এর ক্রিয়াকলাপগুলি ব্যবহার করছেন, তখন নিম্ন 16 টি bit ব্যবহৃত হয় এবং বাইট ক্রিয়াকলাপের জন্য সর্বশেষ 8 bit ব্যবহৃত হয়। এবং এগুলি গাণিতিক ক্রিয়াকলাপগুলির জন্য ব্যবহৃত হয় না, কেবল ডেটা ট্রান্সফার অপারেশনের জন্য কারণ সমস্ত গাণিতিক অপারেশনগুলি কেবলমাত্র 32 bit আকারের হয়। আপনি যখন এক স্থান থেকে অন্য জায়গায় ডেটা স্থানান্তর করছেন, আপনি 8 bit বা 16 bit বা 32 bit স্থানান্তর করতে পারেন। মেমরি থেকে আপনি 8 টি bit ডেটা একটি রেজিস্টারে স্থানান্তর করতে পারেন, এটি শেষ 8 bit ইত্যাদিতে যাবে

(Refer Slide Time: 13:58)

Current Program Status Register (CPSR)

31 30 29 28 27 26

8 7 6 5 4 3 2 1 0

N Z C V Q

I F T M M M M

mode bits

Thumb state

FIQ disable

IRQ disable

overflow

carry/borrow

zero

negative

swayam

FREE ONLINE EDUCATION

CPSRতে কী রয়েছে? এটিতে বিভিন্ন FLAG রয়েছে। এর দ্বারা চিহ্নিত একটি ওভারফ্লো FLAG রয়েছে V, বিয়োগের জন্য একটি carry FLAG C আছে; আপনি এটিকে borrow হিসাবে ডাকেন, একটি শূন্য FLAG Z আছে, এটি ফলাফল শূন্য বা ননজারো কিনা তা পরীক্ষা করে, এবং সাইন স্ল্যাগ N। এছাড়াও প্রসেসরটি অনেকগুলি মোডে থাকতে পারে।

আমি বলেছিলাম 7 টি মোড আছে তবে ভবিষ্যতের সম্প্রসারণের সাথে রাখতে ARM 5 bit স্টোর মোড(store mode) রাখে। 5 বিটে আপনি 32 টি মোড পর্যন্ত সমর্থন করতে পারেন। bit 5 টি thumb state বোঝাতে ব্যবহৃত হয়। তারপরে 6 এবং 7 bitগুলি high priority interrupt কে INTERRUPT দেওয়ার আগে এবং স্বাভাবিককে নিষ্ক্রিয় করার জন্য এবং সক্ষম করার জন্য। এই bitগুলি 1 তে সেট করা থাকলে এগুলি অক্ষম করা হয়; যদি 0 থাকে তবে তারা সক্ষম হয়। এর মধ্যে থাকা অন্য bitগুলি অব্যবহৃত।

(Refer Slide Time: 15:26)

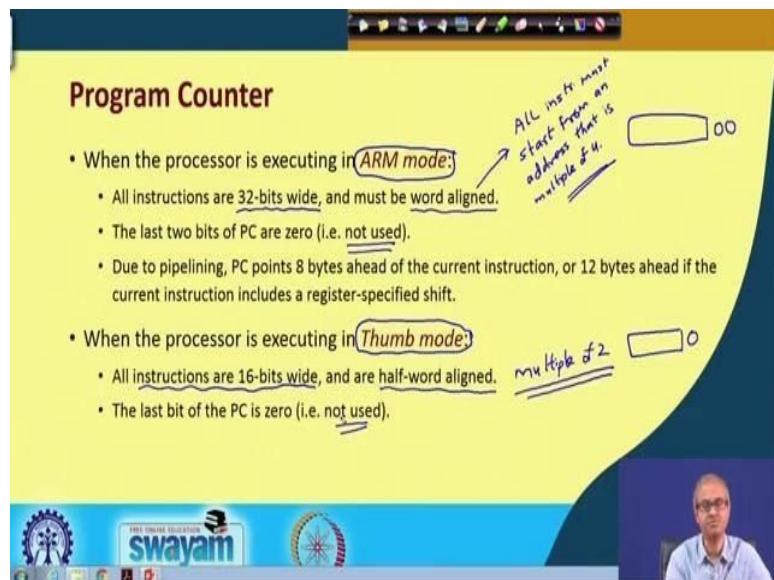
special REGISTER সম্পর্কে কথা বলছি, কেবল আবার দেখুন। এটি r15 বা program counter। আপনি জানেন যে PC সর্বদা সঠিক নির্দেশ কার্যকর করতে মেমরির প্রবর্তী নির্দেশের ঠিকানা রাখে। সুতরাং, গন্তব্য REGISTER PC হলে এর অর্থ সেখানে jump হচ্ছে। লিংক REGISTER সাবটাইন কল নির্দেশের জন্য r14 ব্যবহার করা হয়।

ARM-এ সাবটাইন কল নির্দেশকে বিএল, শাখা এবং লিঙ্ক বলা হয়। এটি কোথায় শাখা বানাবে তা নির্দিষ্ট করে। ব্রাঞ্চিংয়ের আগে PC-র বর্তমান মান এলআর (আর 14) এ সংরক্ষণ করা হবে এবং তারপরে নির্দিষ্ট গন্তব্যটি PC-তে লোড হবে। আপনি যখন ফিরে আসবেন, এলআর এর মানটি PC-তে ফিরে আসবে। সুতরাং আপনি যেখান থেকে চলে গেছেন সেখান থেকে আপনি EXECUTION পুনরায় শুরু করবেন।

r13 স্ট্যাক পয়েন্টার হিসাবে ব্যবহার করা যেতে পারে। আপনি যদি সফ্টওয়্যার স্ট্যাক(software stack) প্রয়োগ করতে চান, আপনি এটি করতে পারেন।

এবং CPSR কার্যকর হয়ে যাওয়া প্রোগ্রামের সাথে বর্তমান স্ট্যাটাস রেজিস্ট্রার(status register) ধারণ করে এবং যখনই কিছু ব্যতিক্রম বা INTERRUPT আসে তখন CPSR বিশেষ বা সংরক্ষিত প্রোগ্রামের একটিতে SPSR নিবন্ধিত হয়। তারা সিপিএসআরের একটি অনুলিপি রাখবে। আপনি যখন ব্যতিক্রম রুটিন থেকে ফিরে আসেন SPSR আপনার এক্সিকিউশন পুনরায় শুরু করার আগে CPSR ব্যাক করা উচিত।

(Refer Slide Time: 17:45)



PC সম্পর্কে কথা বলি, EXECUTION কার্যকর করার দুটি পদ্ধতি রয়েছে; একটি হ'ল ARM মোড, একটি হ'ল থাম্ব মোড(thumb mode)। আসুন এখানে পার্থক্য পরিষ্কার করুন। আমরা প্রথমে এক্সিকিউশনের ARM মোড সম্পর্কে কথা বলি যা ডিফল্ট মোড। ARM মোড বলছে যে সমস্ত নির্দেশাবলী 32 বিট প্রস্তু এবং তাদের WORD প্রাণ্তিক করা হয়েছে। WORD প্রাণ্তিককরণের অর্থ সমস্ত নির্দেশাবলীর একটি মেমরি ঠিকানা(Memory address) থেকে শুরু হওয়া উচিত যা 4 এর কিছু অধিক।

এটি করার প্রয়োজনের বিষয়ে আমি বিশদে যাচ্ছি না। আপনি যখন মাইক্রোকন্ট্রোলারের সাথে মেমরিটিকে ইন্টারফেস করেন তখন মেমোরি ইন্টারলিভিং(interliving) নামে একটি ধারণা আসে। যদি আপনার 4-উপায় অন্তঃলিখন থাকে, তবে 4 টির অধিক এমন ঠিকানা থেকে শুরু করে টানা 4 বাইটগুলি এক ক্লক চক্রে স্থানান্তরিত হতে পারে। তবে যদি এটি না হয় তবে আপনার জন্য 2 CLOCK চক্র প্রয়োজন হবে। সুতরাং, আপনার স্মৃতি স্থানান্তর ধীর হয়ে যাবে। দ্রুত মেমরি অ্যাক্সেস পেতে, ARM

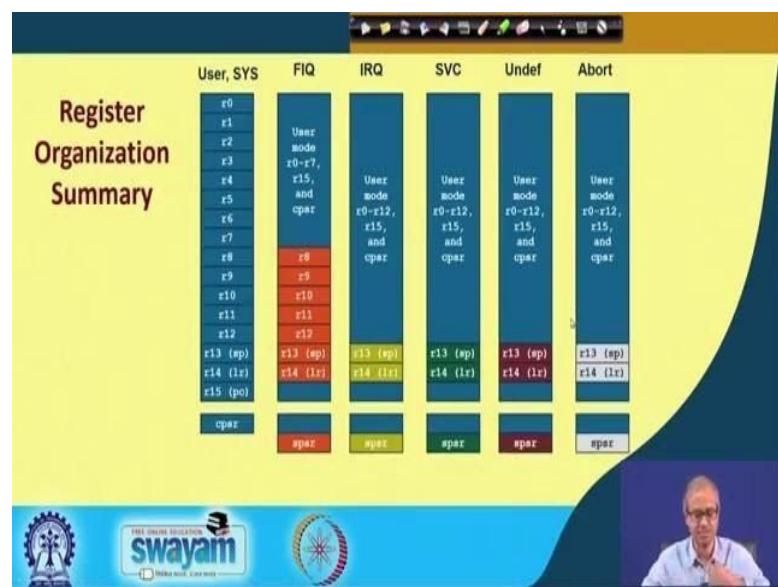
WORD alignment-র উপর জোর দেয় এবং 4 এর একাধিক মানে শেষ দুটি bit 00 হয় আমাদের নির্দেশের ঠিকানাটি রাখা আশা করা যায়, সর্বদা সর্বশেষ 2 bit 0 হবে।

সুতরাং, PC-র শেষ দুটি বিটগুলি আসলে ARM মোডে ব্যবহৃত হয় না, কারণ তারা সর্বদা থাকবে ০। এছাড়াও, PC পাইপলাইন পর্যায়ের সংখ্যার উপর নির্ভর করে 8 byte এগিয়ে বা 12 নির্দেশ করতে পারে।

আপনার যখন 32 bit প্রসেসিংয়ের সম্পূর্ণ শক্তি প্রয়োজন হয় না তখন কার্যকর করার আরও একটি মোড থাকে। হতে পারে আমরা খুব সহজ অ্যাপ্লিকেশনটিতে ARM প্রসেসর ব্যবহার করছি, যেখানে ছোট 16 bit নির্দেশ রয়েছে, কিছু নির্দেশের সাবসেট রয়েছে। এখানে থাক্স মোডে, নির্দেশাবলী 16-bit প্রশস্ত; এর অর্থ, 2 byte এবং সেগুলি অর্ধেক WORD এর সাথে সংযুক্ত অর্থ নির্দেশিক ঠিকানাগুলি 2 এর অধিক হয়।

সুতরাং, PCতে সর্বশেষ বিট 0 হয় যা ব্যবহৃত হয় না। এটি RM মোড এবং থাষ্ব মোডের মধ্যে প্রধান পার্থক্য। ARM মোড হল একটি সুপারসেট যেখানে সমস্ত নির্দেশ 32 টি bitগুলিতে এনকোড(encode) করা হয়, থাষ্ব মোড এটির একটি উপসেট যেখানে আপনি সমস্ত নির্দেশাবলী 16 বিটকে সংক্ষিপ্ত করে দেন যার কারণে কোডের ঘনত্ব বেশি হবে। ছোট অ্যাপ্লিকেশনগুলির জন্য, আপনার বাস্তবায়নের জন্য খুব সম্ভা পদ্ধতি থাকতে পারে। সুতরাং, আপনার ছোট সিস্টেমে প্রয়োজন এমন ক্ষেত্রে থাষ্ব মোড ব্যবহার করা হয়।

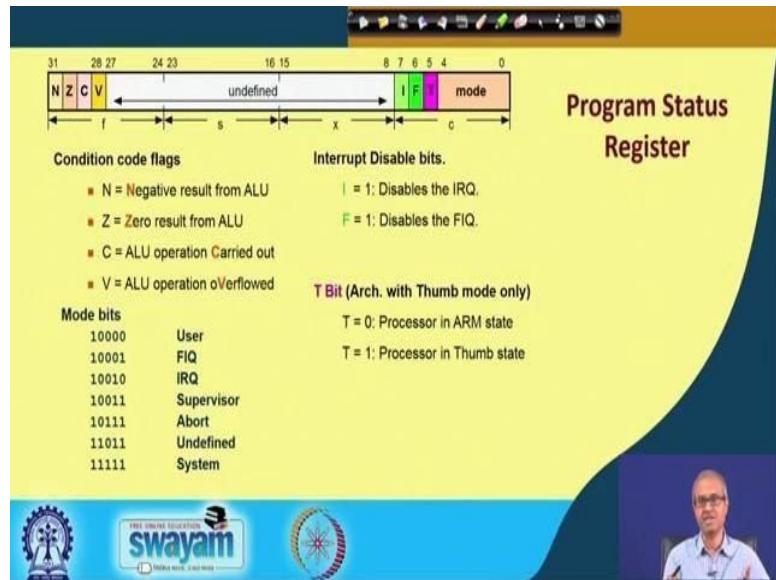
(Refer Slide Time: 21:50)



এই চিত্রটি আপনাকে বিভিন্ন মোডে REGISTERগুলি সম্পর্কে একটি ওভারভিউ দেয়। ব্যবহারকারী বা সিস্টেম মোডে, আপনি 13 টি REGISTER থেকে r0 থেকে r12, তারপরে r13, r14, r5, এসভিভি, লিঙ্ক, PC, CPSR অ্যাক্সেস করতে পারবেন। যে 4 টি SPSR রয়েছে তারা অন্য 5 টি মোডের সাথে

মিল রাখে। যদি interrupt FIQ আসে তবে CPSR এই SPSRটিতে অনুলিপি হয়ে যায় এবং এই r0 থেকে r7 FIQ তে থাকবে, তবে অন্যান্য REGISTERগুলি একটি FIQ -এর সাথে সুনির্দিষ্ট হবে। আপনি যদি ফিরে আসেন তবে এগুলি আবার পরিবর্তন হবে এবং IRQ r0এ r12 থেকে পুরো জিনিসটি আছে, কেবল r13 r14 FIQ- এর সাথে সুনির্দিষ্ট। SVC এবং ABORTর জন্য অনুরূপ।

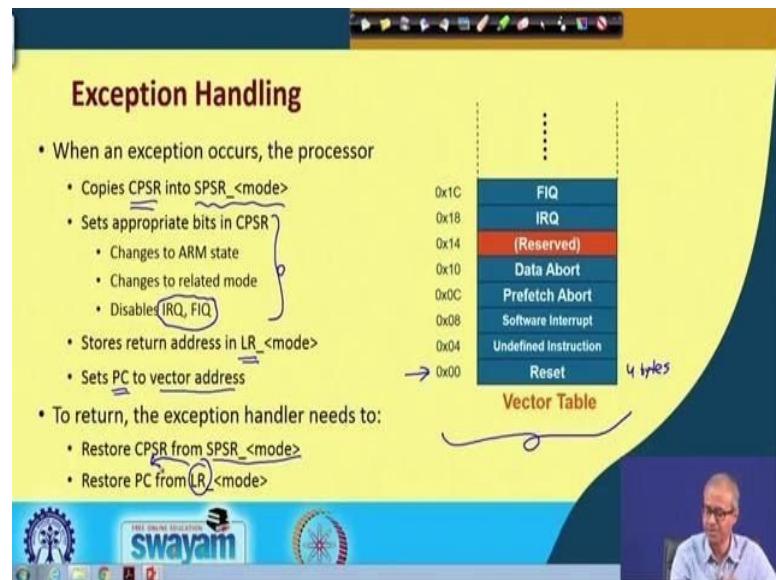
(Refer Slide Time: 23:36)



এখন আরও একবার CPSR নিয়ে কথা বলার বিষয়টি আবার আমি CPSRর আগে বলেছি। শেষ 5 টি bit, এখন নির্দিষ্ট বিট সংমিশ্রণগুলি এখানে দেখানো হয়েছে। এই 7 টি মোডের জন্য, মোড বিট সংমিশ্রণগুলি এই 10000 অর্থ ব্যবহারকারী মোডের মতো সংজ্ঞায়িত করা হয়েছে, 10001 হল FIQ মোড এবং এই জাতীয়

অন্যান্য FLAG বিটগুলি প্রদর্শিত হয়েছে।

(Refer Slide Time: 24:38)



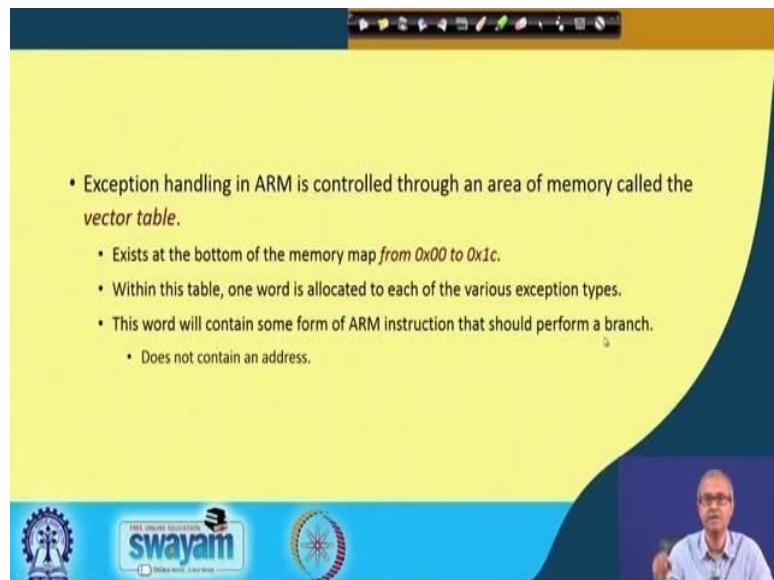
ব্যতিক্রম হ্যান্ডলিং সম্পর্কে কথা বলা, এর মতো একটি প্রক্রিয়া অনুসরণ করা হয়; ভেস্টের টেবিল নামক একটি টেবিলের মধ্যে একটি INTERRUPT ভেস্টের রয়েছে। এই ভেস্টের টেবিলটি ঠিকানা 0 থেকে উপর্যুক্ত, 0x00 মানে হেক্সাডেসিমাল 00; এগুলি হেক্সাডেসিমাল সংখ্যা 004, 008, 00C। এই প্রতিটি এন্ট্রি 4 বাইট হয়।

যখনই কোনও ব্যতিক্রম ঘটে তখন CPSR সংশ্লিষ্ট SPR-এ অনুলিপি করা হবে। আপনি জানেন যে 5 টি SPSR নির্ভর করে ব্যতিক্রম প্রকারের উপর নির্ভর করে CPSR সংশ্লিষ্ট এসপিএসআরে অনুলিপি করা হবে।

এবং CPSR সম্পর্কিত bitগুলি আপনি কোন মোডে রয়েছেন তার উপর নির্ভর করে সেট করা হবে আর INTERRUPT হ্যান্ডলিং করার সময় নেক্সেড INTERRUPT যাতে না আসে সেজন্য, INTERRUPT নিষ্ক্রিয় করা যায়। তারপরে ক্রিয়ে আসার পূর্বে রিটার্নের ঠিকানাটি সেই মোডের সাথে সংশ্লিষ্ট LR রেজিস্টারে সংরক্ষণ করতে হবে, বিভিন্ন মোডের জন্য আলাদা আলাদা এলআর REGISTER রয়েছে এবং PC ভেস্টেরের ঠিকানা দিয়ে লোড করা হয়। আপনি যদি এইটিতে জাম্প লেখেন তবে কেরতের জন্য ব্যতিক্রম হ্যান্ডলার বিপরীত কাজটি করবে। এটি SPSR CPSR, এলআর PCতে স্থানান্তরিত হবে তা আবার লোড হবে।

লক্ষ্য করার বিষয়টি হ'ল এখানে এন্ট্রিগুলি হ'ল এগুলি INTERRUPT হ্যান্ডলারের ঠিকানা নয় বরং এগুলি নির্দেশাবলী। সাধারণত এগুলি হ'ল জাম্প বা কল ধরণের নির্দেশাবলী। সুতরাং, আপনি যখন এখানে আসবেন, এখান থেকে রুটিনে লাফিয়ে উঠবে যা INTERRUPT হ্যান্ডল করছে। সুতরাং, এগুলি সমস্তই একটি WORD এর INSTRUCTION ভেস্টের সারণীতে সংরক্ষণ করা হয়।

(Refer Slide Time: 28:58)



এটি আমি ইতিমধ্যে উল্লেখ করেছি। ভেস্টের টেবিলটি 0x00 থেকে 0x1c ঠিকানা থেকে সঞ্চয় করা হয়। প্রতিটি ব্যতিক্রম প্রকারের জন্য একটি WORD বরাদ্দ করা হয় এবং আমি যেমন বলেছি, এই WORDটিতে কিছু নির্দেশ থাকবে; কিছু জান্স, কিছু ভাঁঁ নির্দেশ। এটি সম্পর্কিত INTERRUPT হ্যান্ডলারের শাখা করবে। এটিতে কোনও ঠিকানা নেই।

(Refer Slide Time: 28:27)

	ARM (cpsr T = 0)	Thumb (cpsr T = 1)
Instruction size	32-bit	16-bit
Core instructions	58 ✓	30 ✓
Conditional execution	Most ✓	Only branch instructions ✓
Data processing instructions	Access to barrel shifter and ALU	Separate barrel shifter and ALU instructions
Program status register	Read-write in privileged mode	No direct access
Register usage	15 GPRs + pc	8 GPRs + 7 high registers + pc

আমি ARM এবং থান্স নির্দেশ সম্পর্কে বলছি, একটি দ্রুত তুলনা। ARM মোডে CPSRতে টি বিটটি 0 এবং থান্স মোডের জন্য 1 সেট করা হয় ARM মোডে, 32 বিট নির্দেশাবলী রয়েছে; থান্স মোডে 16 বিট নির্দেশাবলী রয়েছে। প্রধান নির্দেশাবলীর সংখ্যা 58 এবং 30। Conditional execution এমন একটি বৈশিষ্ট্য যেখানে কিছু শর্ত সত্য বা মিথ্যা কিনা তার উপর নির্ভর করে কিছু নির্দেশ কার্যকর করা হবে।

ARM-এ প্রায় সমস্ত নির্দেশ শর্ত কার্যকর করার পক্ষে সমর্থন করে তবে থাস্বতে কেবল শাখার নির্দেশের জন্য আপনি শর্তাদি পরীক্ষা করতে পারেন, তবে অন্যান্য নির্দেশ শর্তসাপেক্ষ কার্যকরকরণকে সমর্থন করে না। ARM মোডে ডেটা প্রক্রিয়াকরণের নির্দেশাবলী ব্যারেল স্থানান্তরকে সমর্থন করে। তবে থাস্বের মধ্যে এই জাতীয় নির্দেশাবলী নেই, এখানে পৃথক শিফট নির্দেশাবলী রয়েছে, এবং পৃথক ALU নির্দেশাবলী রয়েছে।

ARM মোডে প্রোগ্রামের স্থিতি REGISTER, আপনি সুবিধাযুক্ত মোডে read write করতে পারেন, তবে থাস্ব মোডে আপনি প্রোগ্রামের স্থিতি REGISTERটিতে অ্যাক্সেস করতে পারবেন না। আর ARM মোডে REGISTERগুলির জন্য, আপনি 15 GPR, r 0 থেকে r 14 এবং PC ব্যবহার করতে পারেন, তবে থাস্বটিতে আপনি 8 GPR, 7 টি হাই REGISTER এবং PC ব্যবহার করতে পারেন। হাই REGISTER হল হাই অর্ডার REGISTERগুলির 16 bit। কিছু নির্দিষ্ট REGISTER আছে যা অ্যাক্সেস করা যায়।

(Refer Slide Time: 31:01)

What is Conditional Execution?

- It controls whether or not the CPU will execute an instruction.
- Most instructions have a condition attribute that determines whether it will be executed based on the status of the condition flags.
 - Prior to execution, the processor compares the condition attribute with the condition flags in CPSR.
 - If they do not match, the instruction is not executed.
- Example:

Z

MOVEQ, r1,#0 (if zero flag is set, then r1 = 0)

- The condition attribute (here EQ) is suffixed to the instruction mnemonic.

শর্তসাপেক্ষ কার্যকর, আমি এখানে একটি উদাহরণ নিতে পারি। শর্তসাপেক্ষ কার্যকরকরণ সিপিইউ বর্তমান নির্দেশকে কার্যকর করবে কিনা তা নিয়ন্ত্রণ করে। এখন ARM-এর বেশিরভাগ নির্দেশনা একটি CONDITIONAL বৈশিষ্ট্য নির্দিষ্ট করার অনুমতি দেয়। আমি এই মত একটি উদাহরণ নির্দেশ গ্রহণ করি। সাধারণত যদি এই EQ ব্যবহার না করেন, কেবল এই EQ নেই তা কল্পনা করুন, তারপরে MOV r1, # 0 এর অর্থ হল, r1 0 তে সূচনা হয়েছে এখন যদি শূন্য FLAGটি সেট করা থাকে, তবে কেবল আপনি মুভ করবেন, অন্যথায় আপনি মুভ করবেন না।

সুতরাং, এই শর্তটি নির্দিষ্ট CONDITIONAL উপর নির্ভর করে কার্যকর করা হবে। শর্তসাপেক্ষ EXECUTION কার্যকর করা হয়। আমরা পরে দেখব যে এই ধরণের শর্ত কার্যকরকরণ ব্যবহার করে

আমাদের অনেক ক্ষেত্রে শাখার নির্দেশনা রোধ করতে দেয়। এটি কম সংখ্যক নির্দেশাবলী সহ কোড ঘনস্বকে উচ্চ করে তোলে।

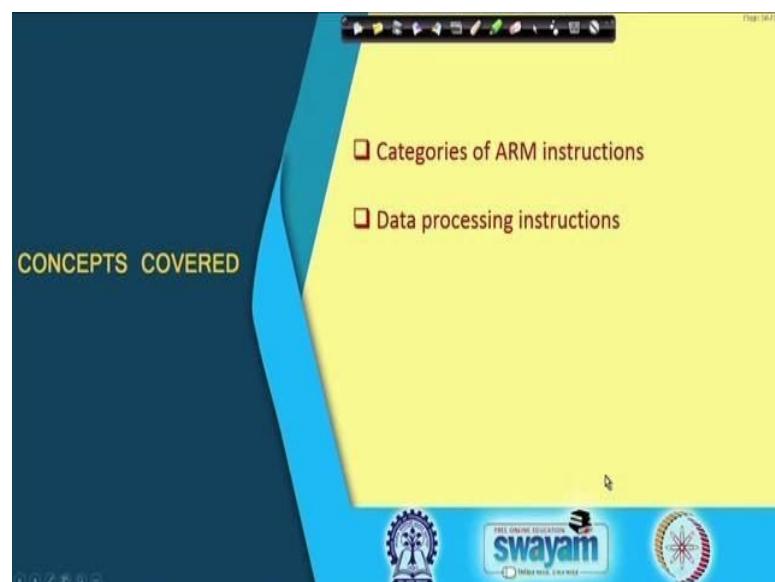
এই সমস্ত কিছু সহ, আমরা ARM আর্কিটেকচারের বিষয়ে আমাদের সংক্ষিপ্ত আলোচনাটি বন্ধ করছি। পরবর্তী বক্তৃতার পরে, আমরা ARM INSTRUCTION সেট এবং এআরএমের অন্যান্য বৈশিষ্ট্যগুলি সম্পর্কে কিছু দিকে নিয়ে যাব যা আমরা প্রদর্শিত প্রকৃত পরীক্ষায় সহায় হবে। আমরা প্রদর্শিত সমস্ত demonstration কিছু ARM বোর্ডের উপর ভিত্তি করে তৈরি করবো এবং কয়েকটি পরীক্ষা-নিরীক্ষাও আমরা আপনাকে আরডিনো বোর্ড(Arduino board দেখাব।

ধন্যবাদ।

Lecture – 07 ARM Instruction Set (Part I)

আমরা এখন ARM INSTRUCTION সেটটিতে খুব সংক্ষিপ্ত আলোচনা শুরু করব। এই কোর্সের পরবর্তী অংশে, আমরা ARM ভিত্তিক এবং আরডিনো(Arduino) ভিত্তিক বোর্ডগুলিতে প্রচুর DEMONSTRATION প্রদর্শন করব। সেখানে আমরা আমাদের প্রোগ্রামগুলি একটি high level language এ লিখব, তবে এখানে এই এবং অন্যান্য কয়েকটি বক্তৃতায় আমরা ARM প্রসেসরের নিম্ন স্তরের বৈশিষ্ট্যগুলি নিয়ে আলোচনা করব, এবং তার জন্য আমাদের **অ্যাসেম্বলি ভাষা** (assembly language) সম্পর্কে একটি প্রাথমিক ধারণা থাকতে হবে যা ARM আর্কিটেকচার সরবরাহ করে।

(Refer Slide Time: 01:07)



এখানে আমরা বিভিন্নভাবে ARM নির্দেশাবলী এবং বিশেষত ARM নির্দেশ সেট আর্কিটেকচার দ্বারা সরবরাহ করা ডেটা প্রক্রিয়াকরণের(Data processing) নির্দেশাবলীর বিভিন্ন বিভাগের বিষয়ে কথা বলব।

(Refer Slide Time: 01:27)

The ARM Instruction Set

- ARM instruction can be categorized into three groups:
 - a) Data processing instructions
 - Operate on values in registers
 - b) Data transfer instructions
 - Move values between registers and memory
 - c) Control flow instructions
 - Change the value of the program counter (PC)

ARM INSTRUCTION সেট সম্পর্কে কথা বলার সময়, একটি এন্ডেড সিস্টেম বিকাশ বা ডিজাইন করার প্রয়োজন, এখানে একটি হার্ডওয়্যার প্ল্যাটফর্ম, একটি মাইক্রোকন্ট্রোলার ভিত্তিক সিস্টেম থাকবে এবং এর জন্য আপনাকে স্টেওয়্যার লিখতে হবে।

বর্তমানে আমাদের বেশিরভাগ সি বা পাইথন বা জাভাতে(C,Python,Java) কিছু high level language সফটওয়্যারটি বিকাশ করা হয়েছে তবে প্রসেসরের আরও ভাল হতে পারে এমন একটি জ্ঞাত সিদ্ধান্ত নিতে প্রসেসরের নিষ্প স্তরের বৈশিষ্ট্যগুলি জানা সর্বদা ভাল, একটি নির্দিষ্ট অ্যাপ্লিকেশন জন্য। এই অনুপ্রেরণার সাথে আমরা আপনাকে ARM প্রসেসরের assembly language এর বৈশিষ্ট্যগুলির সংক্ষিপ্ত পরিচিতি দিচ্ছি যা ARM প্ল্যাটফর্মের মাধ্যমে সরবরাহ করা হার্ডওয়্যার বৈশিষ্ট্যগুলির প্রতিচ্ছবি।

স্পষ্টভাবে যেকোন নির্দেশের সেটকে বিভিন্ন গ্রুপ বা বিস্তৃত বিভাগে শ্রেণিবদ্ধ করা যেতে পারে। ARM-এ আমাদের এই বিভাগগুলি ডেটা প্রসেসিং(data processing), ডেটা স্থানান্তর(data transfer) এবং নিয়ন্ত্রণ প্রবাহ(control flow) হিসাবে সংজ্ঞায়িত করা যাক।

আধুনিক মাইক্রোকন্ট্রোলারদের (microcontrollers) বেশিরভাগ হ্যাভার্ড আর্কিটেকচারের(Harvard Architecture) উপর ভিত্তি করে তৈরি করা হয়, যেখানে আপনি আলাদা প্রোগ্রাম মেমোরি এবং একটি পৃথক ডেটা মেমরি পাবেন। প্রোগ্রামটি গঠনকারী নির্দেশাবলী প্রোগ্রাম মেমোরিতে সংরক্ষণ করা হবে, যখন সমস্ত অস্থায়ী ডেটা যা ডেটা মেমরিতে সংরক্ষণ করা হবে। নির্বন্ধগুলির মধ্যে এই পিসি একটি রেজিস্টার যা প্রোগ্রাম মেমোরিতে পরবর্তী INSTRUCTION-এর ঠিকানা নির্দেশ করবে। মেমরি থেকে যথনই কোনও নতুন নির্দেশনা আনা হয়, এটি PC-তে সঞ্চিত ঠিকানা থেকে তা আনা হবে।

আমি যখন ডেটা প্রসেসিংয়ের নির্দেশাবলীর কথা বলছি তখন আমরা ডেটাতে বিভিন্ন পার্টিগণিত (arithmetic) এবং লজিক অপারেশন(logic operation) পরিচালনার কথা বলছি। এখন, ARM

আর্কিটেকচারের আরআইএসসি দর্শনের উপর ভিত্তি করে, বেশিরভাগ ARM বৈশিষ্ট্যগুলি আরআইএসসি ভিত্তিক। এখানে সমস্ত ডেটা প্রসেসিংয়ের (Data Processing)নির্দেশাবলী কেবল রেজিস্টারে চালিত হয়; এর অর্থ, যখন আমি adding বলি তখন আমরা দুটি নিবন্ধের বিষয়বস্তু যুক্ত করব এবং ফলাফলটি আবার অন্য রেজিস্টারে সংরক্ষণ করব। সুতরাং, MEMORY মোটেও এখানে ছবিতে আসছে না।

এখন, যখন আমরা ডেটা স্থানান্তর(Data Transfer) সম্পর্কে কথা বলি সেখানে বিকল্প রয়েছে। আমরা একটি রেজিস্টার থেকে অন্য রেজিস্টারে স্থানান্তর করতে পারি বা আমরা রেজিস্টার থেকে ডেটা মেমরি বা ডেটা মেমরিতে রেজিস্ট্রেশন করতে পারি। এই বিকল্পগুলি সমস্ত সেখানে রয়েছে এবং এগুলি ডেটা স্থানান্তর নির্দেশাবলী গঠন করবে। এবং, নিয়ন্ত্রণ প্রবাহ নির্দেশাবলী হ'ল সেইগুলি যা কোনও প্রোগ্রামের প্রয়োগের ক্রমকে পরিবর্তিত করে। সাধারণত আমি যেমন বলেছিলাম PC পরবর্তী নির্দেশগুলি কার্যকর করার নির্দেশ করবে, কারণ নির্দেশাবলীর প্রয়োগ হ'ল PC-র মান বৃদ্ধি পাবে। তবে, যখনই আপনার কিছু jump বা subroutine call র মতো কিছু নির্দেশাবলী আসে, হঠাৎ সেই ক্রমটি বিন্ধিত হয়, এবং আপনি অন্য কোনও ঠিকানায় যাবেন এবং সেখান থেকে আপনি আপনার নির্দেশাবলী আনতে শুরু করবেন।

এটি নিয়ন্ত্রণ প্রবাহের পরিবর্তনের দ্বারা বোঝানো হয় এবং এই জাতীয় নির্দেশকে নিয়ন্ত্রণ প্রবাহের INSTRUCTION হিসাবে অভিহিত করা হয়। মূলত নিয়ন্ত্রণ প্রবাহ নির্দেশাবলী হ'ল সেগুলি যা কোনওভাবে PC-র মান পরিবর্তন করে যাতে নির্দেশ কার্য সম্পাদনের স্বাভাবিক ক্রমটি পরিবর্তিত হয়।

(Refer Slide Time: 06:17)

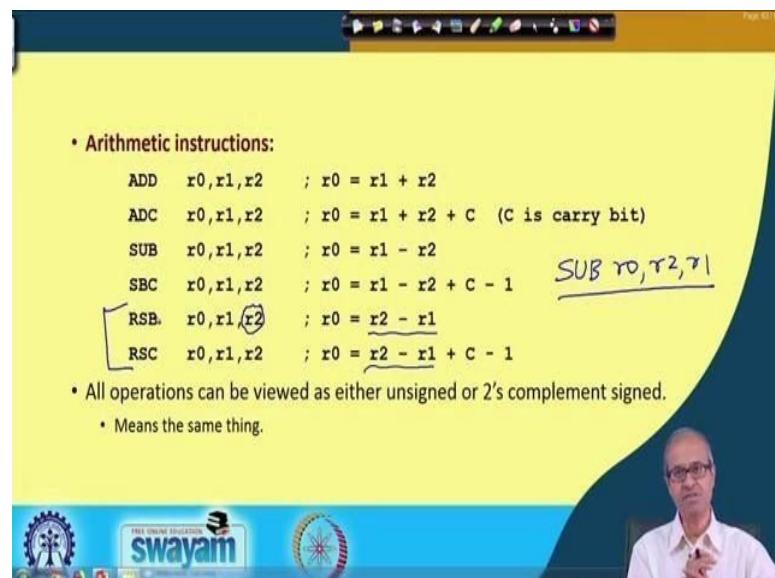
আমরা প্রথমে তথ্য প্রক্রিয়াকরণের নির্দেশাবলী সম্পর্কে কথা বললাম যা ARM INSTRUCTION-এর উপস্থিত রয়েছে। এখন, ARM নির্দেশে প্রথম জিনিসটি লক্ষ্য করে দেখুন যে রেজিস্টারগুলি সমস্ত 32-bit আকারের রয়েছে, প্রসেসরের অভ্যন্তরীণ ALU 32-bit সংখ্যায় অপারেটিংয়ের ক্ষমতা রাখে। সুতরাং, যে

সমস্ত অপারেশন পরিচালিত হচ্ছে সেগুলি 32-bit আকারের। আপনি রেজিস্টারগুলিতে এই অপারেশনগুলি পরিচালনা করতে পারেন, বা অপারেটরগুলির মধ্যে কয়েকটি স্থির হতে পারে - এগুলিকে আক্ষরিক বা তাত্ত্বিক মান বলা হয়। আমি বলতে পারি কোনও রেজিস্টারের সামগ্রীতে 10 মান যোগ করুন। সেই 10 হ'ল ফ্রেকের মতো যা আক্ষরিক বা তাত্ত্বিক মান বলা হয়, সেই মান 10 নির্দেশের অংশ হিসাবে নির্দিষ্ট করা হয় এবং তাকে তাত্ত্বিক অপারেন্ট বলা হয়।

গণনার পরে ফলাফলটিও একটি 32-বিটের ফলাফল এবং এটি কোনও নির্দিষ্ট নিবন্ধে সংরক্ষণ করা হবে। অবশ্যই একটি ব্যতিক্রম আছে; একটি বিশেষ গুণিত নির্দেশ রয়েছে যেখানে ফলাফলটি 64-বিট সংখ্যা হিসাবে সঞ্চিত রয়েছে। এখন, আপনি জানেন যে যখনই আমরা দুটি এন-বিট সংখ্যাকে গুণ করি ফলাফল 2^n বিট হতে পারে। সুতরাং, আপনি যখন দুটি 32-বিট সংখ্যার গুণ করবেন তখন ফলাফলটি সর্বোচ্চ আকারে 64-বিট হতে পারে। একাধিক নির্দেশের একটি সংক্ষরণ রয়েছে যেখানে ফলাফলটি দুটি রেজিস্টার বা 64-বিটে সংরক্ষণ করা যেতে পারে।

এই বক্তৃতাগুলিতে আমরা নির্দেশের এনকোডিং(Encoding) সম্পর্কে আলোচনা করব না; তার অর্থ, নির্দেশের শব্দের বিট(bit)গুলি নিবন্ধগুলি নির্দিষ্ট করে কীভাবে; আমরা আর্কিটেকচারের সেই বিশদে যাব না, তবে মূলত কী ধরণের নির্দেশাবলীর সমর্থন করা হয় সেগুলি আমরা আলোচনা করব।

(Refer Slide Time: 08:55)



প্রথমে arithmetic instruction সম্পর্কে কথা বলা যাক। সর্বাধিক প্রাথমিক নির্দেশাবলী হ'ল সংযোজন এবং বিয়োগফল। ARM বিভিন্ন সংযোজন এবং বিয়োগ নির্দেশের বিকল্প সরবরাহ করে। কী ধরণের নির্দেশাবলী রয়েছে তা আমাদের দেখতে দিন। প্রথমে একটি সহজ যোগ নির্দেশ, এখানে ভাল উদাহরণ হিসাবে আমি তিনটি রেজিস্টার দেখিয়েছি $r0, r1, r2$, তবে আপনার এখানে যে কোনও রেজিস্টার

থাকতে পারে কেবলমাত্র r0, r1, r2 নয়। প্রথমটি গন্তব্য উপস্থাপন করে, দ্বিতীয় এবং তৃতীয়টি দুটি উৎস অপারেশনকে নির্দেশ করে।

সুতরাং, আমি যখন ADD r0, r1, r2 লিখি তখন r1 এবং r2 এর মান যুক্ত হবে এবং ফলাফলটি r0 এ সংরক্ষণ করা হবে। ক্যারি সহ একটি সংস্করণ যুক্ত রয়েছে যা সাধারণত বহু নির্ভুল গাণিতিক পরিচালনা করতে ব্যবহৃত হয়। আপনি যখন দুটি 8-বিট সংখ্যা যুক্ত করছেন, আপনি প্রথমে দুটি 32-বিট সংখ্যা যুক্ত করুন, যদি পরবর্তী একটি 32-বিট সংখ্যা থাকে তবে সেই carryর সাথে যুক্ত হবে। সুতরাং, আপনার নির্দেশনার একটি এডিসি সংস্করণ থাকা উচিত, এটি ক্যারি সহ যুক্ত করা হয়।

তারপরে একটি সাধারণ SUB নির্দেশ রয়েছে এটি হল r1 - r2, ফলাফলটি r0 এ চলে যাচ্ছে। আবার বহু-নির্ভুল গাণিতিকের জন্য বিয়োগের ক্ষেত্রে একই ধরণের ধারনা রয়েছে। একটি BBC সংস্করণ রয়েছে যেখানে carry flag ব্যবহার করা হয়। আসল গণনাটি এইভাবে করা হয়: r1 - r2 + C - 1। এটি বিয়োগ অপারেশনের সময় borrow গ্রহণ করে নেয়। এখন, বিয়োগের নির্দেশাবলীর দুটি প্রকরণ রয়েছে, যেখানে অপারেটরগুলির ভূমিকা বিপরীত। SUB নির্দেশনার মতো আমরা r1 - r2 বলছি। এখন, আমি যদি বলি RSB, এর বিপরীত বিয়োগের অর্থ দাঁড়ায় r2 - r1। একইভাবে, borrow সহ একটি সংস্করণ রয়েছে, ক্যারি সহ বিপরীত বিয়োগ; এটি r2 - r1 + C - 1।

আপনি কেন দুটি ধরণের বিয়োগের নির্দেশের প্রয়োজন তা জিজ্ঞাসা করতে পারেন। আমি এই নির্দেশটি সর্বদা SUB r0, r2, r1 হিসাবে লিখতে পারি। এই ARM নির্দেশনাটি দ্বিতীয় অপারেন্টকে আরও সহজ উপায়ে নির্দিষ্ট করার অনুমতি দেয়। আপনার যদি বিয়োগের বিপরীত সংস্করণ থাকে তবে সেই flexible operand টি বা সাধারণ ডেটা থেকে বিয়োগ করা যেতে পারে। দ্বিতীয় অপারেন্ট কী ধরণের flexibility দেখাবে তা আমরা দেখব, তবে বিয়োগের বিপরীত (reverse version of subtract), এই FLEXIBLE অপারেন্ট এর সাথে একটি সাধারণ অপারেন্ট যোগ বা FLEXIBLE অপারেন্ট থেকে একটি সাধারণ অপারেন্ট বিয়োগ করা যেতে পারে, উভয় বিকল্পই সরবরাহ করা হয়।

এবং, এই বিয়োগের নির্দেশাবলীতে আপনি এই 32-বিট সংখ্যাগুলি স্বাক্ষরবিহীন হিসাবে বা 2 এর পরিপূরক স্বাক্ষরযুক্ত সংখ্যা হিসাবে দেখছেন। এখন, বাইনারি সংযোজন(binary addition) এবং বিয়োগের ক্ষেত্রে এই দুটি পৃথক করে না। যেভাবে পাটিগণিত(arithmetic) এখানে পরিচালিত হয় তা একই রকম।

(Refer Slide Time: 13:21)

কিছু বিটওয়াইজ লজিকাল নির্দেশাবলী রয়েছে। AND, ORR এবং EOR এর মতো নির্দেশাবলী রয়েছে যা নির্দিষ্ট অপারেশনগুলিতে বিটওয়াইজ লজিকাল অপারেশন করে।

এছাড়াও আরও একটি নির্দেশ রয়েছে যা বিটওয়াইজ(bitwise) স্তরে সমর্থিত, এটি সংক্ষেপে BIC বা বিট ক্লিয়ার ইন্সট্রুকশন(bit clear instruction) বলে পরিচিত। আসল অর্থ হ'ল রেজিস্টার r1 হল r2 এর পরিপূরক সহ অ্যান্ডেড; আপনি একটি অপারেশন এ একটি AND করেন। সুতরাং, যে r2 মধ্যে যা বিট 1, আপনি যদি না করেন তবে সেই বিটগুলি 0 হয়ে যাবে; আপনি যদি আর 1 দিয়ে করেন এবং সেই সাথে সংযুক্ত b1 এর বিট 0 হয়।

সুতরাং, আর 2-এ যেখানেই আপনার 1 রয়েছে r1 এর সাথে সম্পর্কিত বিটগুলি 0 করা হবে; তার মানে সেগুলি সাফ হয়ে যাবে। সুতরাং, এটিকে কিছুটা পরিষ্কার নির্দেশও বলা হয়।

(Refer Slide Time: 15:33)

- Register-register move instructions:
 - ✓ `MOV r0,r2 ; r0 = r2`
 - ✓ `MVN r0,r2 ; r0 = not r2`
- MVN is the acronym for "move negated"
- Each 1-bit in r2 clears the corresponding bit in r0.
- In the instruction encoding, the first operand r1 is not specified, as these are unary operations.

তারপরে সরানোর নির্দেশগুলি(register move instructions) নিবন্ধ করার জন্য আপনার কাছে কিছু রেজিস্টার(register) রয়েছে। এটিও আমরা ডেটা ম্যানিপুলেশন বিভাগের অধীনে সংজ্ঞায়িত করছি। রেজিস্টার থেকে রেজিস্টার মুভ(move register to register) করতে এখানে দুই ধরণের নিবন্ধক রয়েছে যা সমর্থিত, একটি move instruction। আপনি দেখতে পাচ্ছেন যখন আমি রেজিস্টার মুভ করতে move রেজিস্টার বলছি, আমার কেবল দুটি অপারেশন নির্দিষ্ট করি। আমি যখন এমওভি r0 বলি, r2 এর অর্থ r2 এর মান r0 এ অনুলিপি করা হয়। অন্য সংক্ষরণ move negated আছে; এর অর্থ, যদি আমি এমভিএন আর০ বলি, r2 এখানে এই 2 টি পরিপূরক হবে এবং r0 তে স্থানান্তরিত হবে।

অনেকগুলি অ্যাপ্লিকেশন রয়েছে যেখানে আমাদের 32-বিট সংখ্যার ঋণাত্মক মান প্রয়োজন। সেখানে আপনি এই এমভিএন নির্দেশ ব্যবহার করতে পারেন। এখন, এখানে এনকোডিংয়ে(encoding) আমি বলেছিলাম আপনার তিনটি রেজিস্টার দরকার নেই। এই আর 1 যা আমরা আগে উল্লেখ করছি, মাঝের অপারেন্ড যা এখানে প্রয়োজন হয় না, কেবল দুটি অপারেন্ড প্রয়োজন।

(Refer Slide Time: 17:13)

• Comparison instructions:

✓ CMP r1,r2 ; set cc on (r1 - r2)	Z S V
✓ CMN r1,r2 ; set cc on (r1 + r2)	
✓ TST r1,r2 ; set cc on (r1 <u>and</u> r2)	$0 \oplus 0 = 0$
✗ TEQ r1,r2 ; set cc on (r1 <u>xor</u> r2)	$1 \oplus 0 = 0$

• All these instructions affect the condition codes (N, Z, C, V) in the current program status register (CPSR).

• These instructions do not produce result in any register (r0).

$0 \oplus 1 = 1$
 $1 \oplus 0 = 1$

বিভিন্ন compare instruction আছে। আপনি CPSR স্মারণ করতে পারেন যে আমরা আমাদের আগের বক্তৃতাগুলিতে আলোচনা করেছি কিছু condition flag রয়েছে। আপনি স্মারণ করতে পারেন যে সেখানে জেড FLAG ছিল, সেখানে একটি এন FLAG ছিল, সেখানে একটি ভি FLAG ছিল। এই FLAGগুলি আসলে কিছু গণিতিক বা লজিক অপারেশনের ফলাফলের উপর নজর রাখে। আপনি যখন দুটি সংখ্যা যুক্ত করবেন তখন এই FLAGগুলি ফলাফলটি 0 ছিল কিনা, ফলাফলটি ধনাত্মক বা ঋণাত্মক ছিল কিনা, সেই অপারেশনের কারণে কোনও ওভারফ্লো(overflow) হয়েছিল কি না ইত্যাদি ট্র্যাক করে রাখবে।

কখনও কখনও আপনার কেবল দুটি সংখ্যার তুলনা করে সিদ্ধান্ত নেওয়া দরকার। আপনি আসলে সংযোজন বা বিয়োগফল করছেন না এবং ফলাফলগুলি কোথাও সংরক্ষণ করছেন না, কেবল আপনাকে দুটি মান তুলনা করতে হবে। তুলনা নির্দেশাবলী একটি হেস্ট উপলব্ধ। আপনি CMP r1,r2 বলতে পারেন; এর অর্থ, অভ্যন্তরীণভাবে আপনি r1 - r2 করছেন এবং ফলাফল 0, ঋণাত্মক বা ধনাত্মক কিনা তা আপনি পরীক্ষা করছেন, ততক্ষণে FLAGগুলি সেট করা হবে।

ঋণাত্মক তুলনা করার আরও একটি সংস্করণ রয়েছে; আসলে এর অর্থ আপনি r1 + r2 করেন এবং তারপরে FLAGগুলি সেট করুন। এখানে এই নির্দেশাবলী একটি রেজিস্টার কোন ফলাফল উত্পাদন করে না; এটি আপনার মনে রাখা উচিত। এটি কেবলমাত্র শর্তের FLAG সেট করে যাতে আপনি পরে ফলাফলটি condition FLAG-এর উপর নির্ভর করে ব্যবহার করতে পারেন।

একইভাবে, পরীক্ষার ধরণের তুলনা রয়েছে। এই তুলনাটির অর্থ আপনি আর 1 এবং আর 2 এর লজিকাল এবং বিটওয়াইজ এ্যান্ড করছেন; তারপরে আপনি ফলাফলটি 0 বা ননজিভে হিসাবে পরীক্ষা করছেন এবং তারপরে FLAGগুলি সেট করুন। একইভাবে, আপনি সমতার জন্য পরীক্ষা, TEQ। সমতা অর্থ exclusive; আপনি যদি 0 এবং 0 এর exclusive OR নেন তবে ফলাফলটি 0 হয়, যদি

আপনি 1 এবং 1 এর exclusive OR নেন, তবে এটি 0 হয় তবে আপনি যদি 0 এবং 1 বা 1 এবং 0 এর exclusive OR নেন তবে এটি 1।

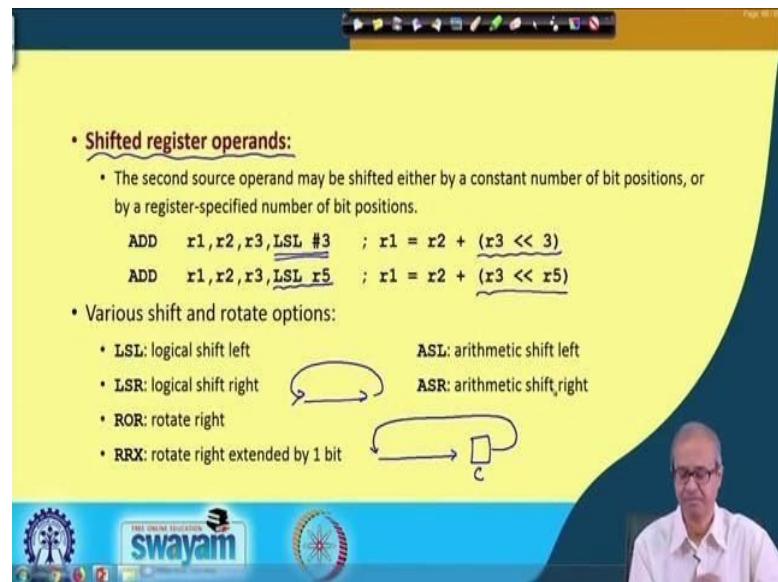
সুতরাং, EXCLUSIVE OR এর ফলাফল আপনাকে জানাবে যে বিটগুলি সমান বা সমান নয়। সুতরাং, আপনি যদি পুরো 32-বিট সংখ্যার XOR নেন এবং ফলাফলটি 0 দেখতে পান, এর অর্থ এই যে দুটি সংখ্যা সমান, সমস্ত বিটগুলি 0 এর XOR মান দিচ্ছে, এজন্য ফলাফলটি 0 হয়।

(Refer Slide Time: 20:53)

তাত্ক্ষণিক ক্রিয়াকলাপ নির্দিষ্ট করার উপায় রয়েছে। যেমনটি আমি বলেছিলাম যে আপনি কিছু তাত্ক্ষণিক ডেটা নির্দিষ্ট করতে পারেন, আমি এই হ্যাশ প্রতীক দিয়ে এটি লিখতে পারি। আমি তাত্ক্ষণিক অপারেন্ট হিসাবে দ্বিতীয় অপারেন্টটি ব্যবহার করতে পারি। অ্যাম্পারস্যান্ট নির্দেশ করে যে আমি যে নম্বরটি নির্দিষ্ট করছি তা হেক্সাডেসিমাল(hexadecimal)।

তাত্ক্ষণিক মানগুলির প্রতি ক্ষেত্রে যে নাম্বারটি উপস্থাপন করতে পারেন তার সর্বাধিক মানের উপর কিছু বাধা রয়েছে। পরিসীমাটি সাধারণত 0 থেকে 255 হয়, আপনি এটি 2's complement number হিসাবেও বিবেচনা করতে পারেন আপনি একটি ঋণাত্মক মানও দিতে পারেন। এবং আরও একটি সুবিধা আছে, নির্দেশে অতিরিক্ত 4 বিট রয়েছে যেখানে আপনি উল্লেখ করতে পারেন যে এই অপারেন্ট মানটি ঘোরানো হবে এবং তার অর্থ 32-বিট সীমার মধ্যে এই 8 বিটগুলি এখানে বা এখানে বা এখানে অবস্থিত হতে পারে। আপনি আপনার প্রয়োজন অনুসারে আপনার নম্বরটি তৈরি করতে পারেন এবং তারপরে আপনি ADD, SUB ইত্যাদি করতে পারেন

(Refer Slide Time: 23:33)



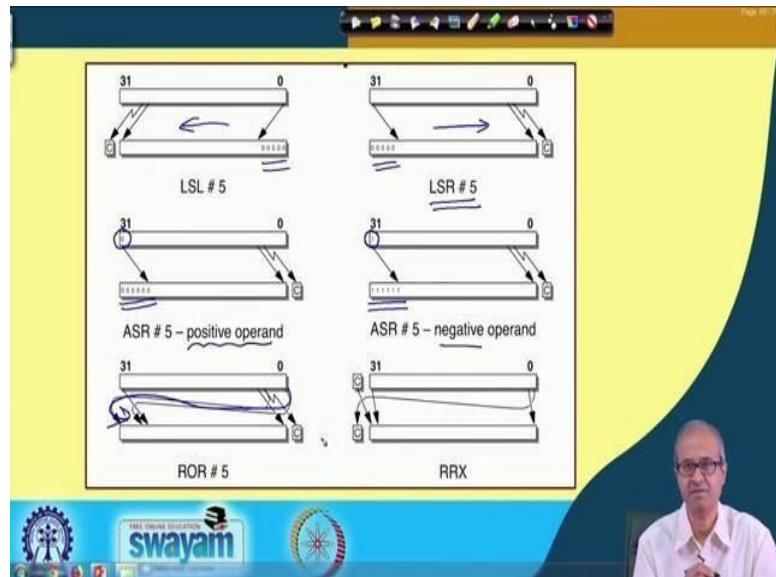
আমি দ্বিতীয় অপারেন্ডে কিছুটা FLEXIBLE, তার কথা বললাম, আপনি দেখুন এখানে স্থানান্তরিত এই রেজিস্টার অপারেন্ডটি এসেছিল। ARM আর্কিটেকচারে বলা হয়েছিল সেখানে ব্যারেল শিফটার রয়েছে। দ্বিতীয় অপারেন্ডটি প্রিস্থিকভাবে ব্যারেল শিফটারের মধ্য দিয়ে যায়, আপনি এটিকে স্থানান্তর করতে পারেন এবং তারপরে কিছু গাণিতিক বা লজিক অপারেশন করতে পারেন। assembly language level এ এটি নির্দিষ্ট করতে পারেন: r1, r2, r3, LSL #3 যোগ করুন। চতুর্থ প্যারামিটারে আপনি বলছেন লজিকাল শিফটটি বাম দিকে 3; তার মানে দ্বিতীয় অপারেন্ডটি তিনটি অবস্থানের দ্বারা বামে স্থানান্তরিত হবে এবং তারপরে r2 এ যুক্ত হবে। আপনি এটি একটি ফ্র্যাক্ষন হিসাবে নির্দিষ্ট করতে পারেন বা আপনি কিছু রেজিস্টার, লজিকাল শিফট(logical shift) r5 দ্বারা বামে নির্দিষ্ট করতে পারেন; r5 এর মান যাই হোক না কেন বহু বিট স্থানান্তরিত হবে।

আপনি এখানে দেখুন আমাদের একটি FLEXIBLE উপায়ে দ্বিতীয় অপারেন্ড নির্দিষ্ট করার সুবিধা রয়েছে। হয় সাধারণ ফর্মে বা কোনও স্থানান্তরিত ফর্মের কারণেই এই বিপরীত বিয়োগের সুবিধাটিও সেখানে।

এবং, কেবল যৌক্তিক শিফটটি(logical shift) বামে নয়, বিভিন্ন শিফট এবং আবর্তিত নির্দেশাবলী বা বিকল্পগুলি পাওয়া যায়; লজিকাল শিফট বাম, লজিকাল শিফট ডান, ডানে ঘোরান। ডানদিকে ঘোরানো মানে যখন আপনি ডানদিকে সরিয়ে ফেলবেন তখন শেষ বিটটি আবার বেরিয়ে আসবে আবার নিবন্ধকের অভ্যন্তরে আসবে। ডান বর্ধিত ঘোড়ানোর অর্থ আপনি যে নিবন্ধটি ডানদিকে ঘূরছে তার ক্যারি FLAG থাকবে। এই বিটটি এই ক্যারি FLAGটিতে যাবে এবং বহনকারী FLAG এতে ঘোরানো হবে, এ কারণেই এটি 33-বিট ঘূর্ণনের মতো, এটি 1 বিট দ্বারা প্রস্তারিত বলে। এবং পাটীগানিত শিফট লেফট(arithmetic shift left) লজিকাল শিফট লেফট(logical shift left) হিসাবে একই, কোন পার্থক্য নেই। এবং, গাণিতিক শিফট রাইটের(arithmetic shift right) অর্থ হল আপনি ডান

স্থানান্তরিত করার সময় এটি যদি ঋণাত্মক সংখ্যা হয় তবে 1 টি সবচেয়ে উল্লেখযোগ্য বিট সাইডে যুক্ত হবে যদি এটি ধনাত্মক সংখ্যা 0 যোগ করা হবে তবে এগুলি বিভিন্ন বিকল্প হবে।

(Refer Slide Time: 26:17)



এই চিত্রটিতে এগুলির কয়েকটি দেখানো হয়েছে। দেখুন যদি 5 টি লজিকাল শিফট লেফট বলা হয় তবে মূল রেজিস্টারটি 5 টি অবস্থানের দ্বারা বামে স্থানান্তরিত হবে এবং 0 এর ডানদিকে যুক্ত হবে। আপনি যদি 5 টি অবস্থানের দ্বারা লজিকাল শিফট রাইটকে একইভাবে বলে থাকেন তবে আপনি ডান স্থানান্তর করবেন এবং 0 টি যুক্ত হবে।

যদি আমি arithmetic shift right বলে থাকি তবে সংখ্যাটি ইতিবাচক হয় যার অর্থ এমএসবি 0 ছিল তবে 0 টি যুক্ত হবে তবে সংখ্যাটি যদি ঋণাত্মক ছিল যার অর্থ সর্বাধিক তাঁৎপর্যপূর্ণ বিট (most significant bit) 1 ছিল তবে বাম দিকে 1 যুক্ত হবে। এবং, আমি যা বলেছি তা ঘূরিয়ে নিয়ে যা যা ঘটে তা এখানে ভিতরে চুকবে। এই হল বিভিন্ন বিকল্প।

(Refer Slide Time: 27:23)

• Multiplication instruction

MUL r1, r2, r3 ; r1 = (r2 x r3) [31:0]

- Only the least significant 32-bits are returned.
- Immediate operands are not supported.

• Multiply-accumulate instruction:

MLA r1, r2, r3, r4 ; r1 = (r2 x r3 + r4) [31:0]

- Required in digital signal processing (DSP) applications.
- Multiplication with 64-bit results is also supported.

Page 10/11

FREE ONLINE EDUCATION
swayam

কিছু গুণের নির্দেশও রয়েছে। MUL r1, r2, r3। আপনি দুটি সংখ্যা আর 2, আর 3 গুন করেন এবং আপনি কেবল ফলাফলের 32-বিট নেন কারণ গুণ ফলাফল 64-বিট হতে পারে তবে আপনি higher order bit গুলি উপেক্ষা করেন তবে অনুমান করতে পারেন যে ফলাফলটি 32 বিটের মধ্যে ফিট হয়ে যাবে এবং আপনি কেবল শেষ 32 বিট নিয়ে যান এবং গুণে তাত্ত্বিক ক্রিয়াকলাপ ব্যবহার করা যায় না।

ডিজিটাল সিগন্যাল প্রসেসিং(digital signal processing) অ্যাপ্লিকেশনগুলির জন্য আরও কার্যকর যে আরও একটি নির্দেশনা রয়েছে, এটিকে বলা হয় গুণ এবং জমা করা। সেখানে আপনাকে অবিচ্ছিন্নভাবে কোনও কোনও সংখ্যার সাথে কোনও সংখ্যাকে গুণিত করতে হবে এবং অবিচ্ছিন্নভাবে অন্য মান যুক্ত করতে হবে।

এটির সাথে আমরা এই বক্তৃতার শেষে আসছি যেখানে আমরা কয়েকটি arithmetic এবং logical নির্দেশাবলী সম্পর্কে কথা বললাম যা ARM নির্দেশ সেট আর্কিটেকচার(set architecture) দ্বারা উপলব্ধ এবং সমর্থিত। পরবর্তী বক্তৃতায়, আমরা তথ্য স্থানান্তর নির্দেশাবলী সম্পর্কে কথা বলতে পারব; রেজিস্টার এবং মেমরির মধ্যে ডেটা স্থানান্তর করতে ব্যবহার করা যেতে পারে এমন বিভিন্ন ধরনের ডেটা ট্রান্সফার নির্দেশাবলী কি কী।

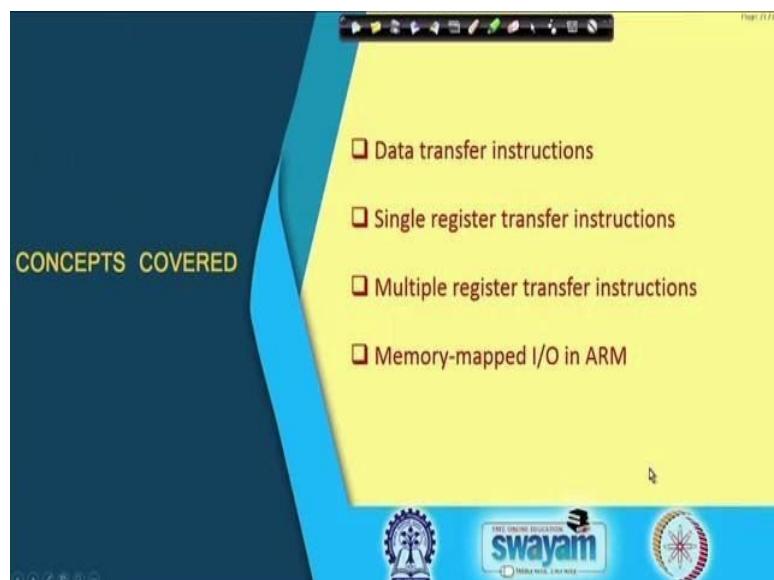
ধন্যবাদ।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 8
ARM Instruction Set (Part II)

শেষ বক্তৃতায় আমরা ARM INSTRUCTION সম্পর্কে আলোচনা করেছিলাম যার সাহায্যে আপনি কিছু গাণিতিক এবং যুক্তি(LOGIC) অপারেশন করতে পারেন। এখন এই বক্তৃতাটিতে আমরা আরও INSTRUCTION এর দিকে নজর রাখব যা আপনাকে **রেজিস্টার(Register)** এবং **মেমরি(Memory)** এর মধ্যে ডেটা TRANSFER করতে দেয়।

(Refer Slide Time: 00:44)



এই বক্তৃতাটিতে আমরা প্রাথমিকভাবে **রেজিস্টার(Register)** এবং **মেমরি(Memory)**র মধ্যে ডেটা স্থানান্তর নির্দেশ(**DATA TRANSFER INSTRUCTION**) সম্পর্কে কথা বলব এবং আপনি দেখতে পাবেন যে এমন INSTRUCTION রয়েছে যা একটি একক REGISTER এর মান এবং একাধিক REGISTER এর মান TRANSFER করতে পারে। এবং আমরা শেষে ARM ইনপুট / আউটপুট ক্রিয়াকলাপ সম্পর্কে কথা বলব যা memory mapped IO নামে পরিচিত।

(Refer Slide Time: 01:18)

• ARM instruction set supports three types of data transfers:

- Single register loads and stores
 - Flexible, supports byte, half-word and word transfers
- Multiple register loads and stores
 - Less flexible, multiple words, higher transfer rate
- Single register-memory swap
 - Mainly for system use (for implementing locks)

semaphore

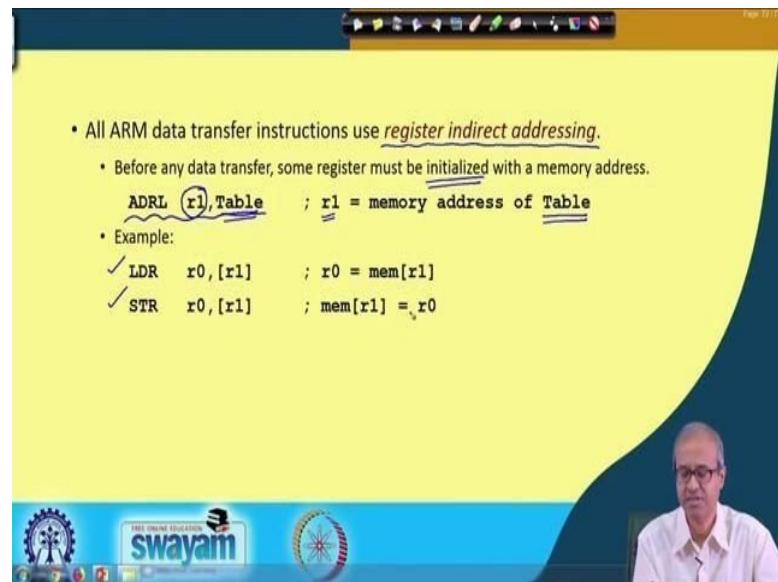
আসুন ডেটা ট্রান্সফার নির্দেশ (DATA TRANSFER INSTRUCTION) যা ARM আর্কিটেকচার দ্বারা সমর্থিত রয়েছে তা দেখুন। মূলত তিনি ধরণের ডেটা ট্রান্সফার করার অনুমতি রয়েছে। এখানে আমরা মূলত রেজিস্টার এবং MEMORYর কথা বলছি। আপনি একক REGISTERক লোড রাখতে পারেন এবং এটি MEMORY থেকে সঞ্চয় করতে পারেন, আপনি মানটি একটি একক রেজিস্টারে লোড করতে পারেন, লোডের নির্দেশনা রয়েছে; স্টোর বলতে কোনও রেজিস্টার থেকে আপনি সেই ফলাফলটি কিছু MEMORYর স্থানে লেখেন।

একক বৈকল্পিক রেজিস্টার(optinal register) রয়েছে যেখানে একটি মান লোড করা বা সঞ্চয় করা হয়; অথবা আপনার একাধিক REGISTER লোড এবং স্টোর করার বিকল্প থাকতে পারে। আপনি যেমনটি বলতে পারেন যে একটি নয়, চারটি রেজিস্টার লোড করা হবে, বা আমাদের চারটি REGISTER একের পর এক MEMORYতে লেখা থাকবে বলে দিন। আপনি একই নির্দেশ তা নির্দিষ্ট করতে পারেন। এছাড়াও একটি খুব বিশেষ ধরণের নির্দেশনা রয়েছে যেখানে একটি রেজিস্টার-MEMORY SWAP অপারেশন অনুমোদিত; অদলবদল (swap)মানে একটি একক নির্দেশনায় একটি রেজিস্টার এবং একটি MEMORY অবস্থানের মান বিনিয়ন।

এই ধরণের নির্দেশনাটি সাধারণ পরিস্থিতিতে সাধারণত অপারেটিং সিস্টেম দ্বারা কার্যকর নয়, এটি অপারেটিং সিস্টেম দ্বারা বিভিন্ন ধরণের লক প্রয়োগের জন্য ব্যবহৃত হয়। একটি খুব সাধারণভাবে ব্যবহৃত লক রয়েছে যাকে বলা হয় semaphore এবং এই ধরণের রেজিস্টার-মেমরি স্বাপ নির্দেশ (REGISTER-MEMORY SWAP INSTRUCTION) এই সেমাফোরগুলি(semaphores) বাস্তবায়নের জন্য খুব কার্যকর।

তবে সাধারণ অ্যাপ্লিকেশন বিকাশের জন্য, আমাদের এই ধরণের নির্দেশিকা ব্যবহার করার দরকার নেই।

(Refer Slide Time: 03:32)



প্রথম বিষয় লক্ষণীয় হল ARM এ ডেটা ট্রান্সফার নির্দেশ(DATA TRANSFER INSTRUCTION) রেজিস্টার অপ্রত্যক্ষ ADDRESS (register indirect addressing) ব্যবহার করে। এটি হল, আপনাকে অবশ্যই কোনও রেজিস্টার ব্যবহার করতে হবে MEMORY অবস্থানের দিকে নির্দেশ করতে এবং সেই নিবন্ধের মাধ্যমে আপনি লোড এবং স্টোর করছেন। আপনি লোড এবং স্টোর ব্যবহার করে MEMORY অ্যাক্সেস করতে পারার আগে, আপনাকে কোনও MEMORYর অবস্থানের address দিয়ে কিছু রেজিস্টার শুরু করতে হবে। একটি নির্দেশনা ADRL আছে। এখনে আপনি কিছু রেজিস্ট্রার নির্দিষ্ট করেছেন, r1 বলুন এবং আপনি কিছু MEMORY অবস্থান, সারণি উল্লেখ করেন। assembly language এ, আপনি absolute address নির্দিষ্ট করেন না; আপনি তথ্য সম্পর্কিত কিছু লেবেল নির্দিষ্ট করেন।

তো, কী হবে? এই ডেটা আইটেম সারণীর সাথে সম্পর্কিত **মেমরি অ্যাড্রেস**(MEMORY ADDRESS)টি r1 এ লোড হবে। একবার এটি হয়ে গেলে, আপনি লোড রেজিস্টার এবং এলডিআর, এসটিআর তে নিবন্ধের INSTRUCTION সংরক্ষণ করতে পারেন। LDR r0, [r1] এর অর্থ **মেমরি**(MEMORY)র অবস্থান যার r1 এ **অ্যাড্রেস**(ADDRESS)গুলি r0 এ স্থানান্তরিত হবে। এসআরটিতে, কেবল বিপরীত; **মেমরি**(MEMORY)র অবস্থান যার **অ্যাড্রেস**(ADDRESS)টি r1 এ রায়েছে সেগুলি r0 থেকে মান পাবে। এই ধরণের লোড(Load) এবং স্টোর(Store) অপারেশনগুলি সমর্থিত।

(Refer Slide Time: 05:24)

• Single register loads and stores

- The simplest form uses register indirect without any offset:

```
LDR r0,[r1] ; r0 = mem[r1]
STR r0,[r1] ; mem[r1] = r0
```

- An alternate form uses register indirect with offset (limited to 4 Kbytes):

```
LDR r0,[r1,#4] ; r0 = mem[r1+4]
STR r0,[r1,#12] ; mem[r1+12] = r0
```

- We can also use auto-indexing in addition:

```
LDR r0,[r1,#4]① ; r0 = mem[r1+4], r1 = r1 + 4
STR r0,[r1,#12]② ; mem[r1+12] = r0, r1 = r1 + 4
```

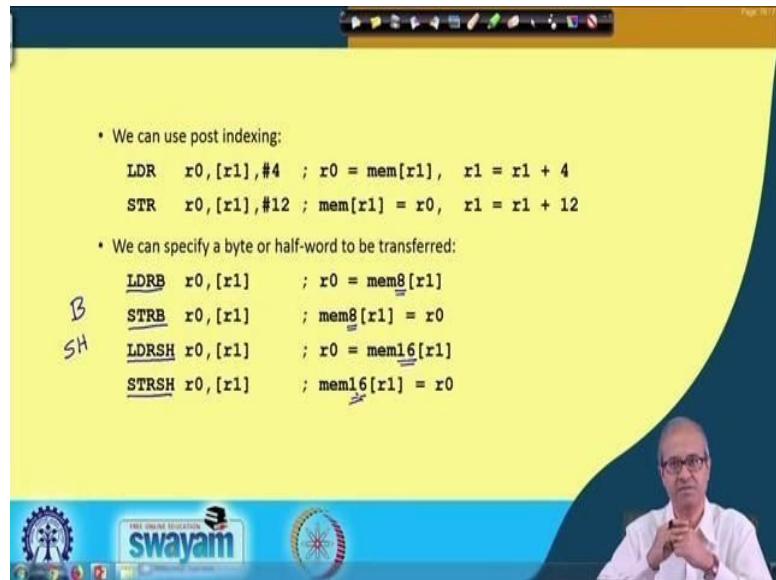
এটি আমরা একক রেজিস্টার লোড এবং স্টোর(unit register load and store) বলি, যেখানে আমরা একবারে একটি রেজিস্টার লোড এবং সঞ্চয় করি। এখন যেমন আমি বলেছি ARM একাধিক রেজিস্টার ব্যবহার করে অন্যান্য বিভিন্ন বিকল্পকে সমর্থন করে, যা আমরা আরও পরে আলোচনা করব। তবে এর আগেও একক রেজিস্টার লোড এবং স্টোরগুলিতে (unit register load and store)আমাদের বিভিন্ন বিকল্প থাকতে পারে।

প্রথম বিকল্পটি সরল **রেজিস্টার ইনডিরেক্ট**(register indirect)। আমরা অফসেটের(offset) সাথে অপ্রত্যক্ষভাবে **রেজিস্টার** (register) রাখতে পারি, যেমন আমরা বর্তমান নির্দেশের বিষয়ে একটি অফসেট মান নির্দিষ্ট করতে পারি। এই অফসেটটি 4 কিলোবাইটের মধ্যে সীমাবদ্ধ, কারণ নির্দেশ প্রদত্ত স্থানটি 12 বিট এবং $12 \times 4 = 48$ কিলোবাইট নির্দিষ্ট করতে পারি। বর্গকার বন্ধনীটির মধ্যে আপনি কেবল **রেজিস্টার** (REGISTER) নির্দিষ্ট করে দিজ্জেন, এখানে আপনি এর মতো কিছু তাত্ক্ষণিক ডেটাও নির্দিষ্ট করে দিজ্জেন; [R1, # 4]। এটার মানে কি? এই রেজিস্টারটির (REGISTER)মান এই অফসেটে(offset) যুক্ত করা হয় এবং তারপরে সেই **অ্যাড্রেস**(ADDRESS)র সাথে সম্পর্কিত MEMORYর অবস্থানটি রেজিস্টারে লোড করা হয়। একইভাবে রাখার জন্য।

মুতরাঃ, কেবল register indirect না, আমি একটি অফসেটও নির্দিষ্ট করতে পারি যা আমাদের লোডিং এবং স্টোরিং থেকে চূড়ান্ত **মেমরি অ্যাড্রেস**(MEMORY ADDRESS) তৈরি করতে নিবন্ধে যুক্ত

করা হবে। তৃতীয় বিকল্পও আছে। অফসেটের মাধ্যমে অপ্রত্যক্ষভাবে রেজিস্টার ব্যবহারের পাশাপাশি আপনার স্বয়ংক্রিয় সূচক থাকতে পারে। অটো(Auto) সূচকের অর্থ, এমন অনেকগুলি অ্যাপ্লিকেশন রয়েছে যেখানে আপনি MEMORY-র অবস্থানগুলির ব্লক থেকে একের পর এক ডেটা লোড বা সংগ্রহ(Data load or store) করতে চান। যে রেজিস্টারটি কিছু MEMORY অবস্থানের দিকে ইঙ্গিত করছিল তা একটি TRANSFER সম্পন্ন হওয়ার পরে স্বয়ংক্রিয়ভাবে বৃদ্ধি পেয়েছে। এবং আমি কীভাবে এটি নির্দিষ্ট করব? আমি এটি দ্বারা এটি নির্দিষ্ট প্রতীক! ব্যবহার করবো।

(Refer Slide Time: 09:23)



কখনও কখনও আপনার 12 টির চেয়ে অন্য কোনও মান দ্বারা রেজিস্টার বাড়ানোর প্রয়োজন হতে পারে, আপনি সেই ক্ষেত্রে নির্দেশের এই প্রকরণটি ব্যবহার করতে পারেন। এবং অন্যটির বিষয় হল আমরা এ পর্যন্ত যা কিছু আলোচনা করেছি আমরা ধরে নিচ্ছি যে পুরো 32-bit word টি স্থানান্তরিত হয়েছে। তবে এমন অনেকগুলি অ্যাপ্লিকেশন রয়েছে যেখানে আপনার পুরো 32-bit ডেটার প্রয়োজন হয় না; 8 বিট বা 16 বিটের ডেটা যথেষ্ট হতে পারে। এগুলিকে byte বা অর্ধেক word বলা হয়। লোড এবং স্টোরের কয়েকটি বিভাগ রয়েছে যা বাইট এবং অর্ধ word এর সাথে কাজ করে। বাইটটি B দ্বারা নির্দেশিত হয় এবং অর্ধ word টি SH দ্বারা নির্দেশিত হয়।

(Refer Slide Time: 11:45)

• Multiple register loads and stores

- ARM supports instructions that transfer between several registers and memory.
- Example: $r2-r5$

```

LDMIA r1, {r3, r5, r6} ; r3 = mem[r1]
; r5 = mem[r1+4]
; r6 = mem[r1+8]

```

A: After
B: Before

- For LDMIB, the addresses will be $r1+4$, $r1+8$, and $r1+12$.
- The list of destination registers may contain any or all of $r0$ to $r15$.

• Block copy addressing

- Supported with addresses that can increment (I) or decrement (D), before (B) or after (A) each transfer.

এখন, আসুন একাধিক REGISTERভুক্ত লোড স্টোরের ধরণে। এখানে আমরা বলছি যে ARM নির্দেশিকা সেটটিতে বেশ কয়েকটি রেজিস্টার এবং MEMORYর মধ্যে ডেটা TRANSFER করার সুবিধা রয়েছে। আপনি স্মারণ করিয়ে দিয়েছিলেন যে আমরা আগে উল্লেখ করেছি যে কয়েকটি সুবিধা রয়েছে যেখানে ARM, RISC আর্কিটেকচার থেকে বিচ্যুত হয়।

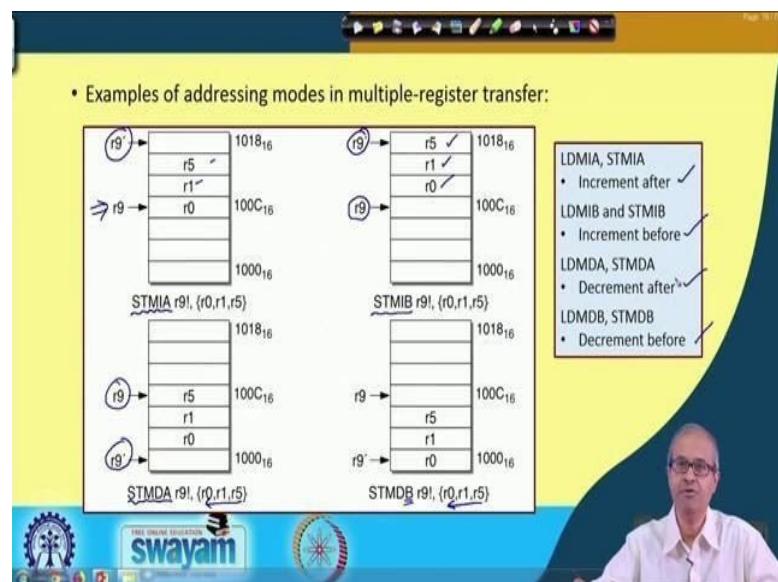
RISC আর্কিটেকচারের একটি দর্শন হ'ল সমস্ত নির্দেশ সমান আকারের হওয়া উচিত। এগুলি খুব সহজ হওয়া উচিত, তাদের একটি একক চক্রের মধ্যে সম্পাদন করতে সক্ষম হওয়া উচিত, তবে আপনি যখন একাধিক WORD স্বভাবতই **ট্রান্সফার (TRANSFER)** করেন তখন আপনি একটি চক্রের মধ্যে execution কার্যকর করতে পারবেন না; একাধিক চক্রের প্রয়োজন হবে কারণ আপনি একাধিক **মেমরি ওয়ার্ড(MEMORY WORD)** লোড বা সঞ্চয়(load or store) করছেন। সুতরাং, এখানেই আমরা থাঁটি RISC ধারণাটি থেকে কিছুটা বিচ্যুত করছি তবে এটি একটি খুব শক্তিশালী নির্দেশ। অনেক অ্যাপ্লিকেশনে আপনি এটি ব্যবহার করতে পারেন। আপনি এই নির্দেশের একটি প্রকরণ দেখতে পান LDMIA, অন্য সংস্করণ LDMIB রয়েছে। A এবং B অক্ষরগুলি যথাক্রমে পরে এবং পূর্বে হয়।

দেখুন কীভাবে এই INSTRUCTION নির্দিষ্ট করা আছে। সাধারণত লোড রেজিস্টারে দুটি অপারেশন থাকে। তবে এখানে প্রথমটি $r1$ । ধারণা করা হয় যে প্রথম অপারেন্টটিতে **মেমরি অ্যাড্রেস(MEMORY ADDRESS)** রয়েছে। এখানে আমরা বর্ণিত যে বর্গাকার বন্ধনীটি দিচ্ছি না, এবং curly বন্ধনী মধ্যে দ্বিতীয় অপারেন্টে আমরা রেজিস্টারগুলির একটি সেট নির্দিষ্ট করছি। এখন আপনি উল্লেখ করতে পারেন এমন অনেকগুলি উপায় রয়েছে। যদি রেজিস্টার (REGISTER)গুলি ক্রমাগত সংখ্যায় থাকে তবে আমরা এই $r2-r5$ এর মতো লিখতে পারি; এর অর্থ $r2, r3, r4, r5$ । অন্যথায় আপনি কমা দ্বারা $r3$,

r5,r6 হিসাবে পৃথক করতে পারবেন এটি A এর অর্থ আপনি প্রথমে লোড করুন এবং তারপরে **অ্যাড্রেস (ADDRESS)**টি বাড়িয়ে নিন।

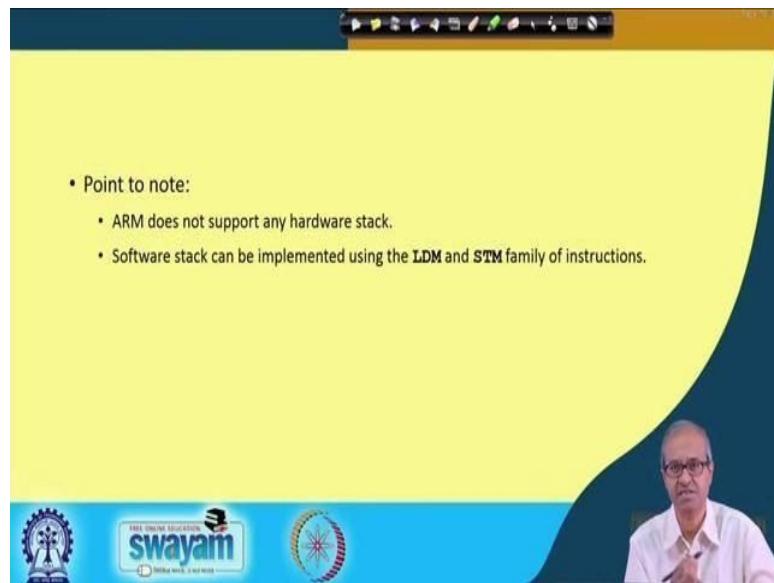
LDMIB এর আরও একটি প্রকরণ রয়েছে। B বা তার আগে আপনি প্রথমে এটি 4 দ্বারা বাড়িয়ে নিন এবং তারপরে লোড করুন। গন্তব্য নিবন্ধের তালিকায় আমরা 16 টি REGISTERকের যে কোনওটি ব্যবহার করতে পারি। এই B এবং এ ছাড়াও, আমি এবং D বিকল্পগুলিও ব্যবহার করতে পারি। আমি ইনক্রিমেন্টের জন্য দাঁড়িয়েছে, এবং D হ্রাস হ'ল। এই সমস্ত বিকল্প উপলব্ধ।

(Refer Slide Time: 16:02)



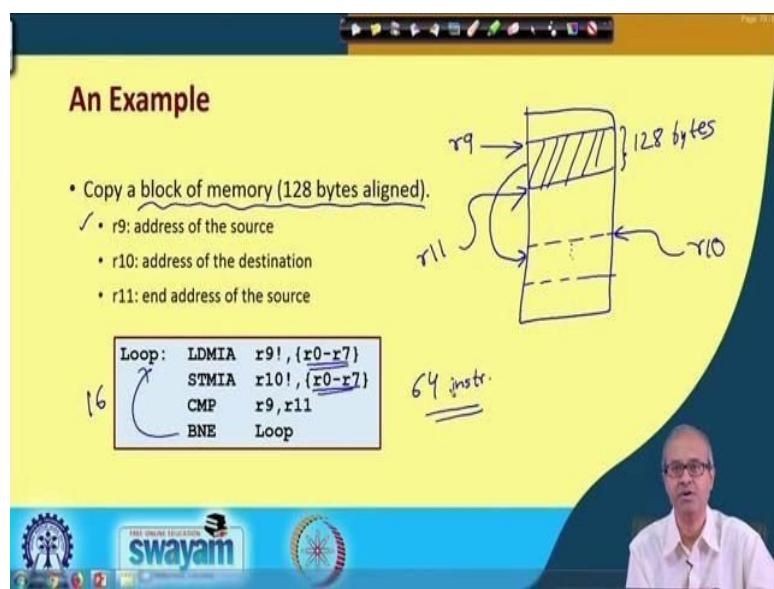
আসুন এখানে কিছু উদাহরণ দেখুন। আমি এখানে প্রথম উদাহরণটি দেখছি তার পরে স্টোর মেমরি(MEMORY)র অবস্থান বৃদ্ধি। সুতরাং, এখানে এই r9 এখানে আমি এটি দিচ্ছি! sign মানে এই r9 বর্ষিত হবে; অটো ইনডেক্সিং অটো ইনক্রিমেন্ট(Auto indexing auto incrementation)। একইভাবে, অন্যান্য উদাহরণ।

(Refer Slide Time: 18:58)



একটি আকর্ষণীয় বিষয় হল ARM আর্কিটেকচারে কোনও স্ট্যাক নেই, যা অন্যান্য সমসাময়িক আর্কিটেকচারে এত জনপ্রিয়। অবশ্যই, একটি প্রোগ্রামার একটি স্ক্রিপ্টওয়্যার স্ট্যাক বজায় রাখতে পারে, তবে কোনও স্পষ্টত পুশ বা পপ INSTRUCTION নেই যা সাধারণত প্রচলিত কম্পিউটার আর্কিটেকচারের নির্দেশ সংস্থার একটি অংশ হয়। এলডিএম এবং এসটিএম(LDM and STM) INSTRUCTION স্ট্যাক প্রয়োগের জন্য ব্যবহার করা যেতে পারে; আমরা পরে কিছু উদাহরণ দেখতে পাব।

(Refer Slide Time: 19:38)



আসুন 128 MEMORY অবস্থানের একটি স্লক অনুলিপি করার একটি উদাহরণ নিতে পারি। ধরুন আমার একটি অ্যাপ্লিকেশন রয়েছে যেখানে MEMORYতে আমার একটানা 128 বাইট রয়েছে।

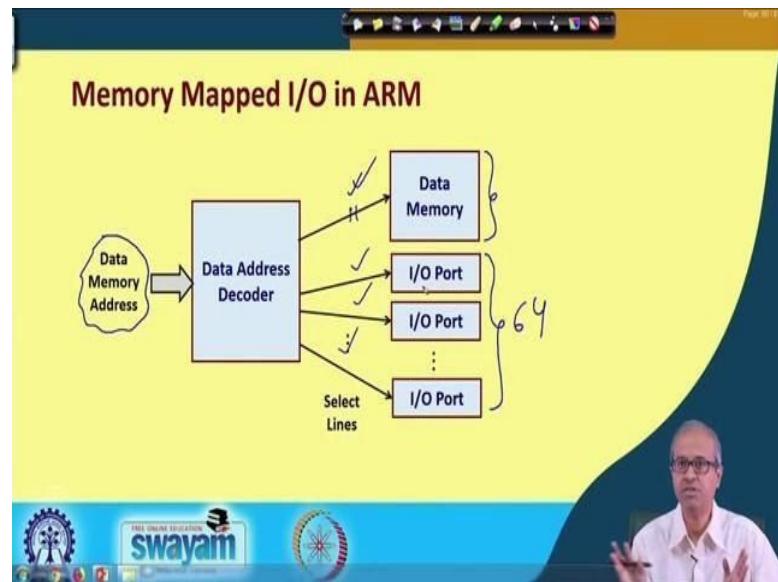
আমি এই পুরো ব্লকটি অন্য কোনও MEMORY লোকেশনে TRANSFER করতে চাই। আমাদের এটি এখানে TRANSFER করতে চাই। এখন একক REGISTER(REGISTER)র চালগুলি ব্যবহার করে, আমি কী করতে পারি? আমি একটি লুপ লিখতে পারি। আমি শুরুর ADDRESS দিয়ে কিছু REGISTER শুরু করি এবং আমি এটি 128 বার করি - প্রাথমিক ব্লক থেকে একটি নম্বর লোড করুন, এটি এখানে সঞ্চয় করুন; পরের নম্বরটি লোড করুন, এটি পরবর্তী স্থানে সংরক্ষণ করুন; তৃতীয় নম্বরটি লোড করুন, এটি তৃতীয় স্থানে সংরক্ষণ করুন এবং আমি পরীক্ষা করে দেখি যে সমষ্টি 128 নম্বর ডেটা স্থানান্তরিত হয়েছে কিনা।

এখন এখানে আমাকে 128 বারের জন্য লুপ করতে হবে। তবে আমি যখন এই জাতীয় একাধিক REGISTER কে TRANSFER নির্দেশ ব্যবহার করি তখন বলা যায়, আমার কাছে আমার 8 টি REGISTER আছে। যদি আমি একসাথে 8 টি REGISTER লোড করি এবং একসাথে 8 টি REGISTER সঞ্চয় করি; তার মানে আমি এক বারে 8 বাইট TRANSFER করছি। লুপের বিচারে আমার ক্ষেত্রে লুপ করতে হবে? $16 \times 8 = 128$ হিসাবে কেবলমাত্র 16 বার। সুতরাং, কার্যকর করা INSTRUCTION-এর সংখ্যা মারাঞ্চকভাবে কম হবে। এটি সামগ্রিক প্রয়োগের সময়ও হ্রাস করবে কারণ প্রতিটি নির্দেশ কার্যকরণের(execution) নিজস্ব ওভারহেড থাকে।

সুতরাং, আপনি যদি একক নির্দেশের অংশ হিসাবে একাধিক MEMORY অবস্থান TRANSFER করেন তবে মোট ব্লক চক্র (clock cycle) সংখ্যা হ্রাস পাবে। প্রোগ্রামটি দেখানো হয়েছে।

এটি আরও দ্রুত হবে কারণ আমরা লুপ প্রতি 4 টি নির্দেশ সহ 16 বার লুপ করছি। আগের ক্ষেত্রে আপনি **128 x 4 = 512 নির্দেশ**(INSTRUCTION)-এর প্রয়োজন ছিল, তবে তুলনায় আপনি এখানে কেবলমাত্র $16 \times 4 = 64$ INSTRUCTION কার্যকর করতে হবে।

(Refer Slide Time: 24:25)

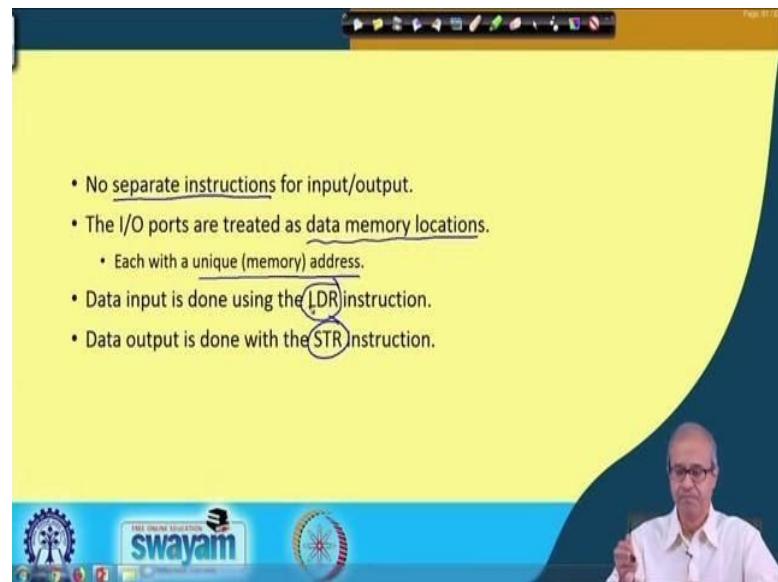


শেষ পর্যন্ত একটি বিষয় আমরা আলোচনা করতে চাই, একটি MEMORY mapped IO নামে একটি ধারণা আছে যা প্রচুর কম্পিউটার সিস্টেমে প্রচলিত।

প্রাথমিক ধারণাটি এরকম। যখনই কিছু ডেটা অ্যাক্সেস (Data Access)করা হয়, তখন প্রসেসরের মাধ্যমে ডেটা MEMORY ADDRESS উত্পন্ন হয়। একটি ডিকোডার রয়েছে যা এই ADDRESSটি ডিকোড করে এবং এটি সম্পর্কিত MEMORYর অবস্থানটি নির্বাচন করে। সাধারণত ডেটা MEMORY অ্যাক্সেস করা হয়, তবে MEMORYটিতে mapped IO আমরা অতিরিক্তভাবে যা বলছি তা হ'ল IO port গুলি MEMORYর অবস্থান হিসাবেও বিবেচিত হয়।

আসুন একটি উদাহরণ নিই, ধরুন এখানে 6 টি IO port রয়েছে। আমরা ধরে নিচ্ছি যে 64 টি মেমরি (MEMORY)র অবস্থানও রয়েছে। ADDRESSর উপর নির্ভর করে একই ডিকোডারটি(decoder) ডেটা মেমরি(MEMORY)র **অ্যাড্রেস**(ADDRESS)গুলির মধ্যে একটি বা এই পোর্টগুলির মধ্যে একটি নির্বাচন করবে। সুতরাং, আইও পোর্ট বা আইও ডিভাইসগুলিকে মেমরি(MEMORY) অবস্থান হিসাবে বিবেচনা করা হয়। এটি memory mapped IO র পিছনের মূল ধারণা। ইনপুট এবং আউটপুট করার জন্য আলাদা কোনও নির্দেশনা নেই।

(Refer Slide Time: 26:12)



ইনপুট এবং আউটপুট সম্পর্কিত আলাদা কোনও নির্দেশনা নেই যা প্রথাগত প্রসেসরের মতো এখানে রয়েছে 8051 মাইক্রোকন্ট্রোলার(microcontroller) বা 8085 মাইক্রোপ্রসেসর(microprocessor) যেখানে আপনি IN এবং OUT নির্দেশিকা পাবেন যা প্রাথমিকভাবে ইনপুট এবং আউটপুট জন্য বোঝানো হয়।

আইও পোর্টগুলি ডেটা MEMORYর অবস্থান হিসাবে বিবেচিত হয়, যেখানে তাদের প্রত্যেকেরই একটি স্বতন্ত্র ADDRESS থাকবে। আপনি যখন কোনও ইনপুট ডিভাইস থেকে কিছু ডেটা পড়তে চান, আপনি লেড নির্দেশ ব্যবহার করবেন; আপনি যখন কোনও আউটপুট ডিভাইসে কিছু ডেটা লিখতে চান আপনি স্টোর নির্দেশ ব্যবহার করবেন। ARM এটি পরিচালনা করে। এটি কিছু flexibility সরবরাহ করে, কারণ যখন আপনি MEMORYর অবস্থানগুলিতে অ্যাক্সেস করছেন তখন নির্দেশিকা সেটগুলির অনেকগুলি নমনীয়তা রয়েছে। এখন আপনি যদি নিজের আই ও ডিভাইসগুলিকে MEMORYর অবস্থান হিসাবেও ব্যবহার করেন তবে আইও ডিভাইসগুলিতে অ্যাক্সেস করার সময়ও একই ধরণের flexibility আপনি ব্যবহার করতে পারেন।

এটির সাথে আমরা এই বক্তৃতাটির শেষে এসেছি। পরবর্তী বক্তৃতায় আমরা নিয়ন্ত্রণ প্রবাহ নির্দেশ (CONTROL FLOW INSTRUCTION) সম্পর্কে কথা বলব যা ARM নির্দেশিকা সেট-এ পাওয়া যায়।

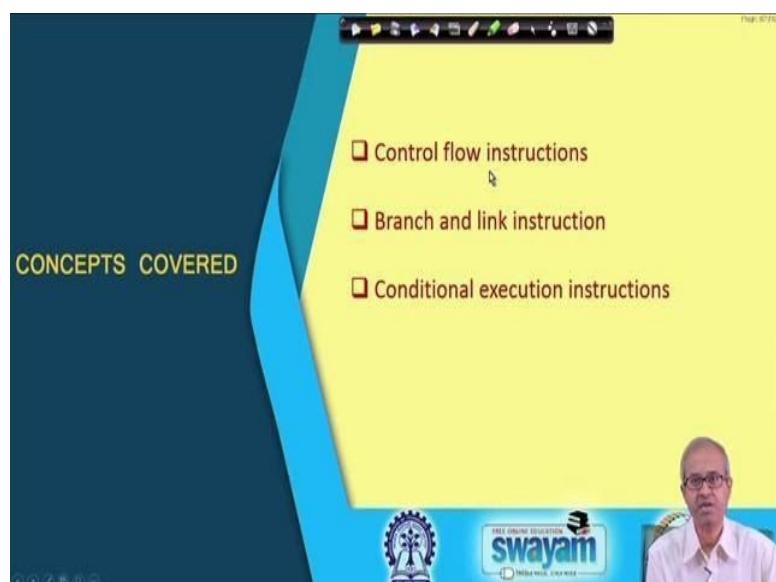
ধন্যবাদ।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 09
ARM Instruction Set (Part III)

আমরা এখন ARM এর **কন্ট্রোল প্রবাহ নির্দেশ (control flow instruction)** নিয়ে আলোচনা করব।

(Refer Slide Time: 00:31)



এটা হল **ARM নির্দেশাবলী**(instruction set) সম্পর্কে বক্তৃতা সিরিজের তৃতীয় অংশ। এখানে আমরা control flow instructions সম্পর্কে কথা বলবো। বিশেষত আমরা ব্রাঞ্চ এবং লিঙ্ক নির্দেশ (branch and link instruction) এর দিকে নজর রাখব যা প্রাথমিকভাবে সাবরুটিন কল(subroutine call) পরিচালনা করার জন্য ব্যবহৃত হয় এবং আমরা শর্তসাপেক্ষ কার্যকর প্রয়োগের বিষয়ে কথা বলব যা এই বিভাগ নিয়ন্ত্রণ প্রবাহের (control flow) সাথে সম্পর্কিত বলে বিবেচনা করতে পারেন।

(Refer Slide Time: 00:58)

(c) Control Flow Instructions

- These instructions change the order of instruction execution.
 - Normal flow is sequential execution, where PC is incremented by 4 after executing every instruction.
- Types of conditional flow instructions:
 - ✓ Unconditional branch
 - ✓ Conditional branch
 - ✓ Branch and Link
 - ✓ Conditional execution

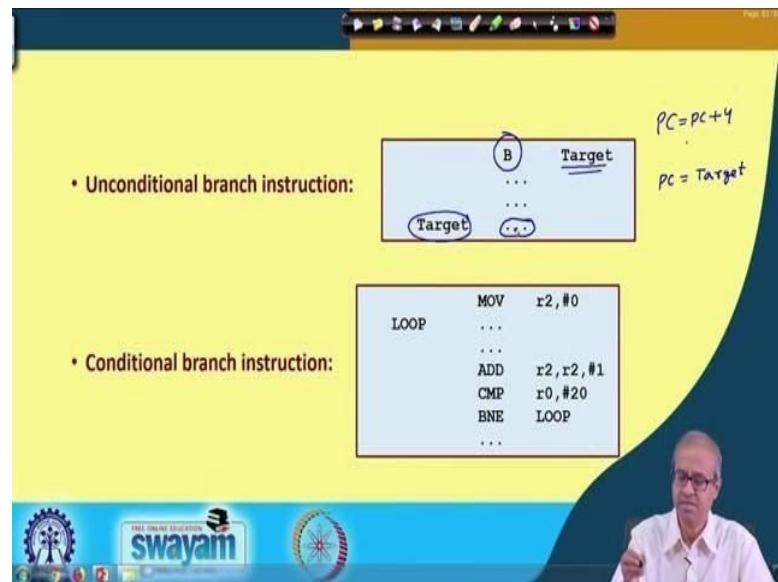
Diagram: A vertical stack of four horizontal lines representing memory. A dashed arrow labeled 'PC' points to the top line. Another dashed arrow labeled 'PC = PC+4' points to the bottom line.

swayam

কন্ট্রোল প্রবাহ নির্দেশ (control flow instruction) এর দিকে তাকানো যাক। যেমনটি আমি আগেই বলেছি, **কন্ট্রোল প্রবাহ নির্দেশ (control flow instruction)** কার্যকর করার ক্রম পরিবর্তন করে। execution কার্যকর করার প্রবাহে, নির্দেশাবলী সাধারণত মেমরিত একের পর এক সংরক্ষণ করা হয়, PC পরবর্তী নির্দেশ (INSTRUCTION) কার্যকর করার জন্য INSTRUCTION গ্রহণ করে। যখনই একটি INSTRUCTION কার্যকর হয়, আপনি PCকে 4 দ্বারা বৃদ্ধি করেন, যাতে PC পরবর্তী নির্দেশের দিকে INSTRUCTION পেতে পারে। প্রতিটি নির্দেশের আকার 32 বিট আকারের হয়, তাই আপনাকে এটি 4 বাইট দ্বারা বাড়াতে হবে।

এখন নিয়ন্ত্রণ প্রবাহে, এই PC, 4 দ্বারা বাড়ানো হচ্ছে না, বরং আপনি অন্য কোনও মান দিয়ে PC আপডেট করছেন। সুতরাং, পরবর্তী INSTRUCTION অন্য কোথাও থেকে হবে। নিয়ন্ত্রণ প্রবাহের নির্দেশের ধরণের ক্ষেত্রে, নিঃশর্ত শাখা(unconditional branch), শর্তাধীন শাখা(conditional branch), শাখা (branch) এবং লিঙ্ক এবং কন্ডিশনাল এক্সিকুশান নির্দেশ (conditional execution instruction) থাকতে পারে। নির্দেশাবলী এই 4 বিভাগে ব্যাপকভাবে হতে পারে।

(Refer Slide Time: 02:35)



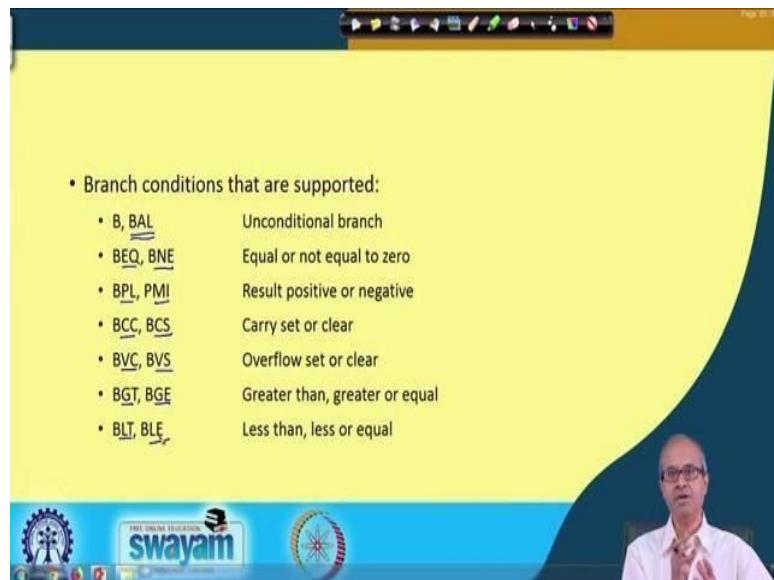
আসুন আমরা এক করে এই নির্দেশাবলী দেখি। প্রথমটি হল নিঃশর্ত শাখার INSTRUCTION। শর্তহীন শাখায়, আমাদের একটি শাখা INSTRUCTION রয়েছে যার mnemonic হল কেবল B। assembly language এ আমরা কেবল B টাগেট বলি।

সুতরাং এর অর্থ এই নির্দেশের পরে আমরা পরবর্তী নির্দেশিকায় যাব না, বরং পরবর্তী নির্দেশিকাটি এখানে লক্ষ্যমাত্রায় থাকবে। PCটি যেভাবে বাড়ানো হবে তা হল $PC = PC + 4$ না করার পরিবর্তে, আমরা যা বলছি তা হল আমরা $PC = \text{টাগেট}$ করব, যাতে পরের INSTRUCTIONটি প্রাপ্ত হবে এই INSTRUCTIONটি এখানে সংরক্ষণ করা হবে। একে বলা হয় নিঃশর্ত শাখা INSTRUCTION এর অর্থ এই যে আপনি যখনই এই বি INSTRUCTION পাবেন তখন সর্বদা এই নতুন ঠিকানায় একটি নিয়ন্ত্রণ স্থানান্তর থাকবে।

এখন আপনি কিছু শর্তের উপর নির্ভর করে নিয়ন্ত্রণের এই স্থানান্তরও পেতে পারেন। এগুলিকে বলা হয় শর্তাধীন শাখা INSTRUCTION। এখানে আমি শাখা যদি না সমান (বিএনই) নামে একটি শর্তযুক্ত শাখা INSTRUCTION দেখিয়েছি। আসুন এই সহজ কোড বিভাগটি দেখুন; আপনি বুঝতে পারবেন এটি কি করছে। এখানে আমরা কিছু রেজিস্টার আর 2 থেকে 0 সূচনা করছি, তারপরে কিছু নির্দেশাবলী এখানে কিছু গণনা চলছে। তারপরে আমরা আর 1 তে 1 যুক্ত করছি এবং ফলাফলটি আর -2-এ পুনরায় সংরক্ষণ করা হবে।

তারপরে আমরা তুলনা করছি যে আর 2, 20 হয়ে যাচ্ছে কিনা, যদি এটি 20 না হয় তবে আমরা লুপটিতে ফিরে যাই। আমরা এটি পুনরাবৃত্তি করি এবং একবার এটি 20 হয়ে যাওয়ার পরে, এটি বেরিয়ে আসে এবং পরবর্তী নির্দেশাবলীর সাথে চালিয়ে যায়। এর অর্থ এটি আমরা প্রয়োগ করেছি এমন একটি সাধারণ লুপ যা 20 বার নির্দেশাবলীর একটি সেট পুনরাবৃত্তি করবে। আপনি এই জাতীয় শর্তসাপেক্ষ লুপটি প্রয়োগ করতে এই শর্তাধীন শাখা INSTRUCTION ব্যবহার করতে পারেন।

(Refer Slide Time: 05:42)



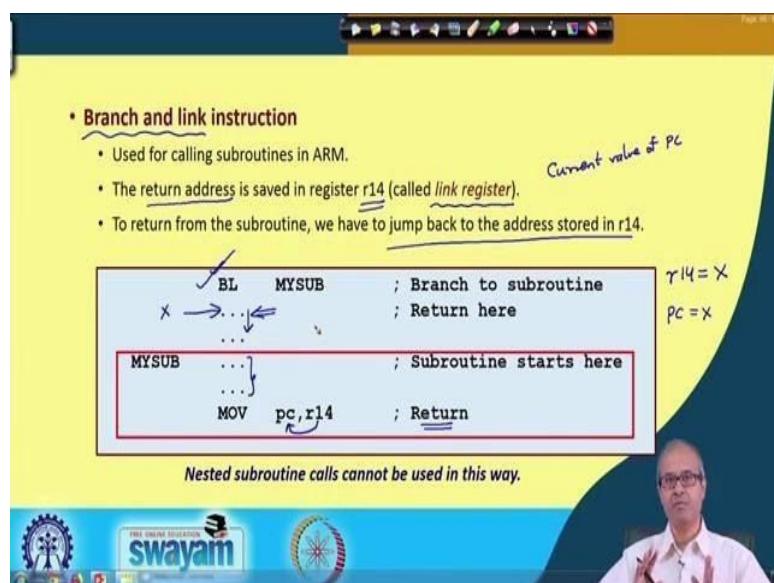
• Branch conditions that are supported:

- B, BAL Unconditional branch
- BEQ, BNE Equal or not equal to zero
- BPL, BMI Result positive or negative
- BCC, BCS Carry set or clear
- BVC, BVS Overflow set or clear
- BGT, BGE Greater than, greater or equal
- BLT, BLE Less than, less or equal

এই তালিকাটি দেখায় কেবল বিএনএই(BNE) নয়, বিভিন্ন শর্তের বিভিন্ন সম্ভাবনা রয়েছে।

আপনি কিছু গণনা বা তুলনা করতে পারেন, এবং এর ভিত্তিতে আপনি অন্য কোথাও লাফিয়ে পড়বেন কিনা সে বিষয়ে সিদ্ধান্ত নিতে পারেন।

(Refer Slide Time: 06:54)



• Branch and link instruction

- Used for calling subroutines in ARM.
- The return address is saved in register r14 (called link register). *Current value of PC*
- To return from the subroutine, we have to jump back to the address stored in r14.

BL MYSUB ; Branch to subroutine
X → ... ; Return here
...
MYSUB ... ; Subroutine starts here
...
MOV pc, r14 ; Return

Nested subroutine calls cannot be used in this way.

সাবরুটিন কল সম্পর্কে কথা বলছি, অনেকগুলি INSTRUCTION সেট আর্কিটেকচারে, CALL এর মতো কিছু INSTRUCTION রয়েছে। একটি কল INSTRUCTION রয়েছে যা আপনি সাবরুটিনকে কল(subroutine call) করতে ব্যবহার করেন। আপনি যখন সাবরুটিনকে সাধারণত কল করেন তখন কি হয়? PCটি স্ট্যাকের মধ্যে লোড করা হয়, আপনি সাবরুটিন (Subroutine) জাম্প দেন, সাক্ষুটাইন কার্যকর হয়ে যায়, সাব্রোটিনের শেষ INSTRUCTIONটি রিটার্নের

INSTRUCTION হবে। রিটার্ন কি করবে? এটি স্ট্যাক থেকে শেষ উপাদানটি পপ করবে, এটি এটি PCতে লোড করবে, যার অর্থ আপনি যেখানে কল INSTRUCTIONটি কার্যকর করা হয়েছিল সেই প্রোগ্রামে ফিরে আসবেন। এর অর্থ পরবর্তী INSTRUCTION, কারণ PC সর্বদা পরবর্তী নির্দেশকে ধারন করে।

এইভাবে জিনিসগুলি একটি প্রচলিত প্রসেসরের মাধ্যমে পরিচালিত হয়, তবে এআরএম-এ কোনও কল INSTRUCTION নেই বরং শাখা এবং লিঙ্ক নামে একটি INSTRUCTION রয়েছে, সংক্ষেপে বিএল(BL)। আমি আপনাকে যা বলেছিলাম এখানে কোনও স্ট্যাক নেই। এখানে একটি বিশেষ রেজিস্টার আর 14 রয়েছে যা লিঙ্ক রেজিস্টার বলে পরিচিত। রিটার্ন address মানে PCর বর্তমান মান। যখনই কোনও INSTRUCTION কার্যকর করা হয় PC অবিলম্বে 8 দ্বারা বাড়ানো হয়। সুতরাং, এটি পরবর্তী নির্দেশের দিকে INSTRUCTION করছে, এটিই ফেরতের ঠিকানা (return address)।

রিটার্ন এড্রেস(return address) লিঙ্ক রেজিস্টারে(link register) সংরক্ষণ করা হবে এবং তারপরে আপনি সাবরোটিনে ঝাঁপিয়ে পড়ুন। এবং যখন আপনি আবার ফিরে যাওয়ার চেষ্টা করছেন, আপনাকে r14-তে সঞ্চিত ঠিকানায় ফিরে যেতে হবে। এআরএম সাবরুটিন কল / রিটার্ন এভাবে পরিচালনা করা যায়। আসুন এখানে একটি সাধারণ উদাহরণ গ্রহণ করা যাক; খুব ছোট একটি কোড বিভাগ দেখানো হয়েছে। মনে করুন কোনও প্রোগ্রামে আপনি কোথাও একটি সাবরুটাইন কল INSTRUCTION বিএল দিয়েছেন এবং এটি যে সাবরুটিন আপনি কল করছেন।

আমাদের MYSUB হল সাবরুটিনের লেবেল যা আপনি কল করছেন এবং এই লাল বাক্সটি সাবরুটিনের বডিকে INSTRUCTION করে। এই সাবরুটিনের অভ্যন্তরে কিছু নির্দেশাবলীর ব্যবস্থা থাকবে তবে সাবরোটিনে ঝাঁপ দেওয়ার আগে কী হবে? লিঙ্ক রেজিস্টার r14 এ, এই পরবর্তী নির্দেশিকার ঠিকানাটি বলি এটি(X), এই X সংরক্ষণ করা হবে। বর্তমান PC যদি X হয় তবে X, X r14-এ সঞ্চিত হয়ে যাবে এবং তারপরে এটি MYSUB এ শাখা করা হবে। MYSUB এ বেশ কয়েকটি নির্দেশাবলী থাকবে, এটি কার্যকর হবে। শেষে যখন এটি ফিরে আসতে চায়, এটি কিছু এমওভি নির্দেশাবলী কার্যকর করে। এমওভি কেমন? PCতে MOV r14।

মূলত আপনি এই জায়গায় ফিরে যাচ্ছেন এবং আপনি এখান থেকে পুনরায় execution র কাজ শুরু করছেন এইভাবে আপনি সাবরুটিন (subroutine) কল প্রয়োগ করছেন এবং এআরএম-এ (ARM) ফিরে আসছেন। তবে এখানে স্পষ্টভাবে উল্লেখ করা হয়েছে যে আপনি এইভাবে নেস্টেড সাবরুটিন কল(Nested Subroutine Call) বা পুনরাবৃত্তি বাস্তবায়ন করতে পারবেন না, যা এত সাধারণ; আপনি এখানে দেখুন ফেরতের ঠিকানা r14 সংরক্ষণ করার জন্য আপনার কেবলমাত্র একটি রেজিস্টার রয়েছে।

(Refer Slide Time: 12:05)

• We can use software stack to save/restore the return address and registers.
• An example showing nested subroutine calls and return.

BL MYSUB1
...
MYSUB1 STMFD r13!, {r0-r2, r14}
BL MYSUB2 ←
...
LDMFD r13!, {r0-r2, pc}
MYSUB2 ...
MOV pc, r14

Nested subroutine calls can be used.

এখানে আমি আবার একটি সাধারণ উদাহরণ দেখাব যেখানে আমরা একটি স্ট্যাকের ব্যবহার চিত্রিত করছি। আমি বলেছিলাম যে আর্ম একটি স্ট্যাক সমর্থন করে না, তবে আমরা উপলভ্য নির্দেশাবলী ব্যবহার করে স্ট্যাকের প্রয়োগ করতে পারি। আসুন আমরা ধরে নিই যে আমাদের একটি স্মৃতি অবস্থান রয়েছে যা আমরা এখানে প্রয়োগ করতে বা স্ট্যাক করতে চাই। আমি বলেছি $r13$ একটি নিবন্ধ যা আপনি সাধারণত স্ট্যাক পয়েন্টার হিসাবে ব্যবহার করেন। আমাদের বলুন যে আমরা $r13$ ব্যবহার করছি, এগুলি কিছু মেমরি অবস্থান এবং $r13$ এখানে INSTRUCTION করছে। এই প্রোগ্রামটিতে আমি আমার মূল প্রোগ্রামটি থেকে কী চিত্রিত করছি, আমি বিএল মাইএইচ 1 এ কল করছি।

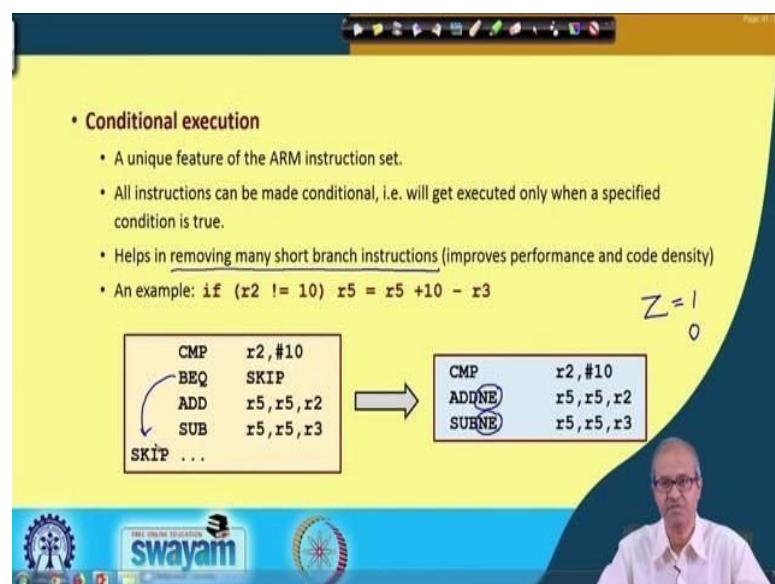
এটি MYSUB1 এর বডি এবং MYSUB1 থেকে, আমি আবার অন্য সাবরুটিন MYSUB2 কল করছি। এটি একটি নেস্টেড কল, দুটি স্ট্যাকের বাসা আমি প্রদর্শন করছি। নেস্টিংয়ের প্রথম স্তরে, কেবলমাত্র আমার $r14$ মানটি যেন হারিয়ে না যায় তা নিশ্চিত করার জন্য, আমরা একাধিক রেজিস্টার স্টের INSTRUCTION STMFD ব্যবহার করছি। সুতরাং, এটি কি করে? এটি মূলত $r14$ এবং কিছু অন্যান্য নিবন্ধগুলি সংরক্ষণ করে যা আমার প্রয়োজন হতে পারে। এই তিনটি এবং এই আর 14 লিঙ্ক রেজিস্টারটি এখানে $r13$ স্ট্যাকের মধ্যে সংরক্ষণ করা হয়। তারপরে আপনি MYSUB2 শাখায় শাখা এবং লিঙ্ক করুন।

এখন আপনার আর 14 টি ওভারলাইট হয়ে গেছে, এতে এই নির্দেশের ফেরতের ঠিকানা থাকবে। MYSUB2 কার্যকর হয়ে যায় এবং এটি হয়ে গেলে আপনি $r14$ এ সঞ্চিত ঠিকানায় ফিরে আসেন। সুতরাং, আপনি এখানে ফিরে যান এবং এখান থেকে পুনরায় কার্যকর করা শুরু করুন। এখন শেষে আপনার একটি ম্যাটিং লোড একাধিক রেজিস্টার লোড INSTRUCTION রয়েছে, আপনি কী

করবেন? আপনি আর 13 থেকে আবার লোড করুন; এই r13 সেখানে INSTRUCTION করছে। এখানে আপনি r0 থেকে r2 লোড করছেন, এবং শেষটি PCতে লোড হবে।

সুতরাং, এই আর 14 মালটি স্ট্যাকে যা কিছু সংরক্ষণ করা হয়েছিল তা এখন PCতে লোড হবে, যার অর্থ এটি স্বয়ংক্রিয়ভাবে এখানে ফিরে আসবে। এই জাতীয় একাধিক নিবন্ধের লোড এবং স্টোর(Load and Store) নির্দেশাবলী আপনি নেস্টেড সাবরুটিন কল(Nested Subroutine call) বা এমনকি পুনরাবৃত্তি প্রয়োগ করতে ব্যবহার করতে পারেন।

(Refer Slide Time: 15:41)



ARM নির্দেশাবলী(instruction set) একটি খুব গুরুত্বপূর্ণ বৈশিষ্ট্য হ'ল শর্তসাপেক্ষ এক্সিকুশন (execution)। এটি একটি অনন্য বৈশিষ্ট্য যা আমরা সাধারণত অন্যান্য নির্দেশাবলী সেটগুলিতে দেখতে পাই না। এআরএমের ডিজাইনের খুব সাবধানে এই নির্দেশাবলীর সেটটি বের করে নিয়েছেন। এটি বলে যে নির্দেশাবলী শর্তযুক্ত করা যেতে পারে; শর্তসাপেক্ষ অর্থ তারা হয় execution কার্যকর করতে পারে বা তারা কার্যকর করতে পারে না। আমি প্রতিটি নির্দেশের সাথে কিছু শর্ত নির্দিষ্ট করতে পারি, এমন একটি ক্ষেত্র থাকবে যেখানে কিছু শর্ত নির্দিষ্ট করা হয়েছে; যদি এই শর্তটি ধরে থাকে তবে তা কার্যকর করা হয়; অন্যথায় এটি কার্যকর করা হয় না।

একটি উদাহরণ নেওয়া যাক। ধরুন হাই লেভেল ভাষা (high level language) এ আমি এরকম কিছু বাস্তবায়ন করতে চাই। এখানে আমার একটি সমাধান আছে। ধরুন আমি এর মতো বাস্তবায়ন করি আমাকে 10 দিয়ে r2 পরীক্ষা করতে হবে, এটি 0 এর সমান বা 0 এর সমান নয় কিনা আমার প্রয়োজনীয়তাটি বলে যদি এটি 0 এর সমান হবে না, তবে এই গণনাটি করুন। তো, আমি কী করছি? আমি আর 2 এর সাথে 10 এর সমান হয় তবে আমার এটি করার কথা নয়;

যদি এটি সমান হয় তবে আমি এই SKIP শাখা করছি। আমি এই দুটি INSTRUCTION-এ এড়িয়ে চলেছি তবে এটি যদি সমান না হয় তবে আমি এই ADD ও SUB দিয়ে যাচ্ছি।

কেবলমাত্র আমিই বলতে চাই যে, এখানে আমি একটি শাখার নির্দেশের প্রয়োজন, তবে যদি আমি শর্তসাপেক্ষে বিভিন্ন ক্লিয়ের ব্যবহার করি তবে আপনি দেখতে পাচ্ছেন যে আমরা এখানে কী করেছি? আমরা আবার আর 2 এর সাথে 10 এর সাথে তুলনা করছি পরবর্তী ADD ও SUB নির্দেশিকায় আমরা শর্তসাপেক্ষ পোস্টফিক্সের(Postfix) সমান নয়; এর অর্থ, যদি না হয় তবে আপনি এইগুলি করেন।

সুতরাং, আমি যখন ADDNE বলি তখন কম্পিউটারটি 0 FLAG টি পরীক্ষা করে; যদি এটি 0 হয় তবে এটি কার্যকর করুন; একইভাবে যদি FLAG 0 হয়, তবে এটি কার্যকর করুন। এই শর্তযুক্ত বিভিন্ন নির্দেশাবলী একটি শাখার নির্দেশ (INSTRUCTION) এড়াতে সহায়তা করে। এটি অনেকগুলি ছোট শাখা মুছে ফেলতে সহায়তা করে। একটি সাধারণ কোডে আপনি এরকম অনেকগুলি দৃষ্টান্ত খুঁজে পাবেন। এই শর্তসাপেক্ষ নির্দেশাবলী আপনার কোডকে আরও খাটো এবং আরও দক্ষ করে তুলতে এই জাতীয় অনেক শাখার নির্দেশাবলী দূর করতে সহায়তা করতে পারে। এটিই প্রধান সুবিধা।

(Refer Slide Time: 20:18)

Postfix	Condition
CS	Carry set
EQ	Equal (zero set)
VS	Overflow set
GT	Greater than
GE	Greater than or equal
PL	Plus (positive)
HI	Higher than
HS	Higher or same (i.e. CS)

Postfix	Condition
CC	Carry clear
NE	Not equal (zero clear)
VC	Overflow clear
LT	Less than
LE	Less than or equal
MI	Minus (negative)
LO	Lower than (i.e. CC)
LS	Lower or same

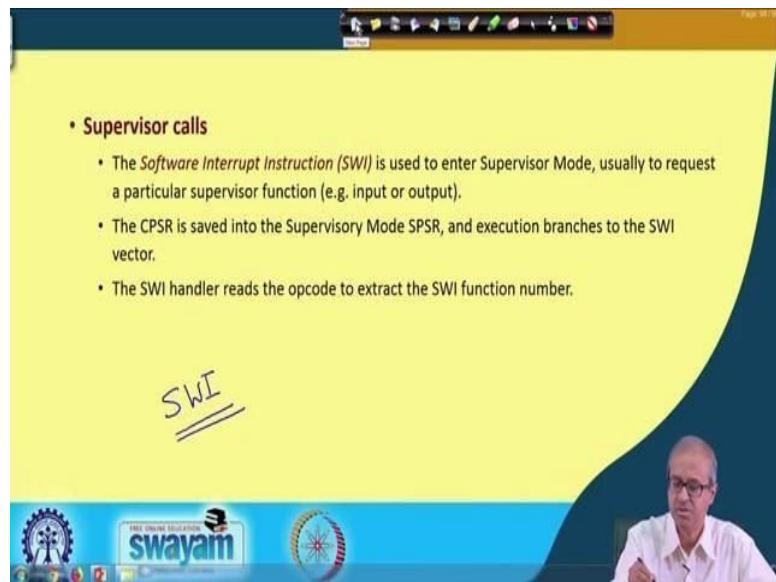
এখন উপরোক্ত বিভিন্ন নির্দেশাবলী পোস্টফিক্স(postfix) এখানে প্রদর্শিত হয়।
একটি নির্দেশের অংশ হিসাবে এখানে 4 টি বিট সংরক্ষিত রয়েছে যা শর্তটি উল্লেখ করবে।

(Refer Slide Time: 21:13)

আসুন আরেকটি উদাহরণ নিই। এখানে আমরা একসাথে দুটি শর্ত পরীক্ষা করছি। এটি প্রচলিত পদ্ধতি, আপনি প্রথমে আর 1 এবং আর 3 তুলনা করুন; যদি তারা সমান না হয় তবে আপনি সরাসরি এড়িয়ে যেতে পারবেন; তার মানে, আপনার এটি করার দরকার নেই। যদি সরাসরি এটি SKIP -এ আসার সমান না হয়, তবে আপনি পরবর্তী শর্তটি r5 এবং r6 পরীক্ষা করে দেখুন। যদি সেগুলি সমান না হয় তবে আপনি SKIP-এ যান অন্যথায় আপনি যুক্ত করেন।

তবে শর্তসাপেক্ষে বিভিন্ন, দ্বিতীয় তুলনা নির্দেশিকা আপনি শর্তযুক্ত করতে পারেন। সুতরাং, আপনি এখানে দেখুন আপনি দুটি শাখার INSTRUCTION অপসারণ করছেন। আপনার কোডটি এত বেশি দক্ষ হয়ে উঠছে। শর্তসাপেক্ষ কার্যকর নির্দেশাবলীর সংযোজন, লোকেরা কোডগুলি যেভাবে লিখতে পারে তাতে এটি একটি নতুন মাত্রা যুক্ত করেছে। কোডটি খুব কমপ্যাক্ট এবং দক্ষ হতে পারে, এবং কার্যকর করার ক্ষেত্রেও কম সময় লাগবে কারণ নির্দেশের সংখ্যা কম রয়েছে।

(Refer Slide Time: 23:14)



আরও একটি INSTRUCTION রয়েছে যা PC-র মানও পরিবর্তন করে। অবশ্যই, এছেড়ে থাকা সিস্টেম ডিজাইনের দৃষ্টিকোণ থেকে এটি এতটা গুরুত্বপূর্ণ নয়। এগুলিকে সুপারভাইজারি কল বলা হয়। সক্ষওয়্যার ইন্টারপেট বা SWI নামে একটি INSTRUCTION রয়েছে, এটি সাবরুটিন (subroutine) কলের মতো। এটি কিছু নিবন্ধক r14 এ রিটার্ন(return) ঠিকানা সংরক্ষণ করে একটি সাক্ষটিনও ঝাঁপিয়ে দেবে, তবে মূল পার্থক্য হল এই লাফটি করার সময় এটি প্রসেসরের মোডটি সুপারভাইজার মোডেও পরিবর্তন করবে। এই নির্দেশাবলী সাধারণত এমন একটি পরিবেশে ব্যবহৃত হয় যেখানে অপারেটিং সিস্টেম রয়েছে এবং আপনি অপারেটিং সিস্টেম থেকে কিছু পরিষেবা অনুরোধ করার চেষ্টা করছেন। আপনি SWI কল দিন, এটি ওএস(OS) লাফ দেয় এবং তদারকি মোডে পরিবর্তন করে। সাধারণত যখন ওএস কার্যকর করে, প্রসেসর মোডটি **সুপারভাইসারি**(supervisory) হয়, যা স্বয়ংক্রিয়ভাবে ঘটে। আমরা এর খুব বেশি বিশদে যাচ্ছি না।

আমরা সংক্ষেপে দেখেছি, বিভিন্ন ধরণের নির্দেশাবলী যা ARM-এ রয়েছে এবং অনন্য বৈশিষ্ট্য রয়েছে। বিশেষত আমি এই বক্তৃতায় শর্তসাপেক্ষ কার্যকর করার কথা বলেছি। এর আগে আপনি গাণিতিক এবং যুক্তি নির্দেশের শর্তাবলী দেখেছেন যখন আপনি নির্বাহ করছেন, দ্বিতীয় অপারেন্টেটি অনেক নমনীয় উপায়ে নির্দিষ্ট করা যেতে পারে। এই নমনীয় পদ্ধতিগুলির সুবিধা গহণের পরে আপনি একাধিক নিবন্ধক স্থান্তর নির্দেশাবলী ইত্যাদির সাহায্যে আপনার কোডটি আরও খাটো করে তুলতে পারেন।

আমি যেমন বলেছি যে এই কোর্সে আমরা কয়েকটি মাইক্রোকন্ট্রোলার বোর্ডের উপর ভিত্তি করে প্রচুর পরীক্ষামূলক demonstration দেখব। পরবর্তী বক্তৃতায় আমরা আপনাকে এমন একটি বোর্ডের সাথে পরিচয় করিয়ে দেব যার ভিত্তিতে অনেক পরীক্ষা-নিরীক্ষা প্রদর্শিত হবে। আমরা বোর্ডগুলির মূল বৈশিষ্ট্যগুলি, বিভিন্ন ধরণের সংযোজক এবং কীভাবে তাদের ব্যবহার করব সেগুলি নিয়ে আলোচনা করব এবং পরবর্তীকালে তারা কীভাবে ব্যবহার করা হয় তা বাস্তবে আমরা দেখতে পাব।

ধন্যবাদ।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 10
About the STM32F401 Nucleo Board

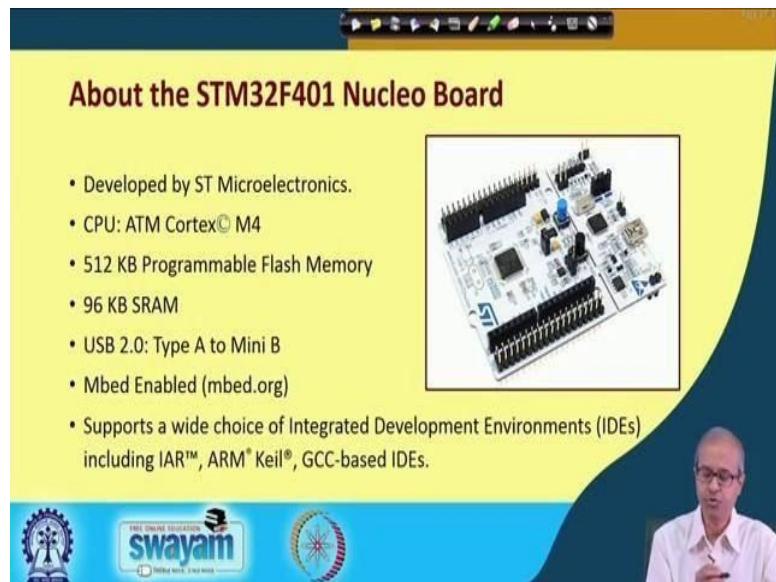
এই বক্তৃতাতে একটি অতি জনপ্রিয় মাইক্রোকন্ট্রোলার (microcontroller) বোর্ডের সাথে পরিচয় করিয়ে দেব যা এন্ডেড সিস্টেম (embedded system) ডিজাইনে ব্যবহৃত হয়। বোর্ডের নাম STM32F401।

(Refer Slide Time: 00:42)



এই বক্তৃতায় বোর্ড সম্পর্কে কয়েকটি উল্লেখযোগ্য বৈশিষ্ট্য জানাব।

(Refer Slide Time: 01:10)



সুতরাং, এই STM32F401 একটি বোর্ড যার ছবিটা এখানে দেখা যাচ্ছে। এটি ST microelectronics নামে একটি সংস্থা তৈরি করে। এই বোর্ডে অনেকগুলি ছোট ছোট উপাদান রয়েছে। এখানে আপনি এখানে একটি বড় আইসি চিপ(IC chip) দেখতে পারেন, এটি প্রকৃত মাইক্রোকন্ট্রোলার যা ভিতরে বসে আছে এবং আরও অনেকগুলি আনুষাঙ্গিক রয়েছে।

আপনি এখানে উল্লেখ করা মূল বৈশিষ্ট্যগুলি দেখুন: এই বোর্ডটি এসটি মাইক্রো ইলেক্ট্রনিক্স(ST micro-electronics) একটি সংস্থা তৈরি করেছে। তারা চিপটি তৈরি করে না, তারা বোর্ড তৈরি করে, এবং CPU যেটি এই মিডল চিপটি ব্যবহার করছে যা এখানে দেখানো হয়েছে, এটি ARM Cortex M4। এটি ARM প্রসেসর যা ব্যবহৃত হয় এবং প্রসেসরের নাম ARM Cortex M4।

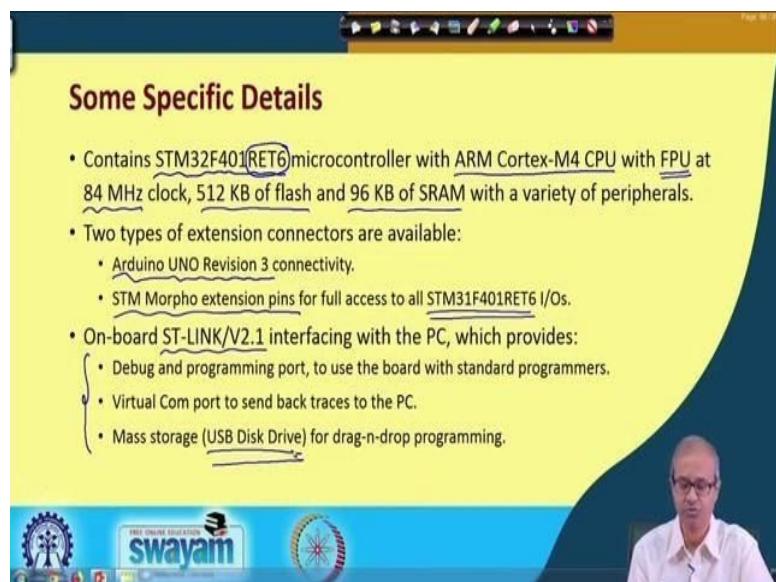
এই ARM CORTEX M4 বোর্ডের বৈশিষ্ট্যটি হ'ল ভিতরে 512 কিলোবাইট প্রোগ্রাম মেমরি রয়েছে। তারা প্রোগ্রাম মেমরি হিসাবে ফ্ল্যাশ মেমরি (Flash Memory) ব্যবহার করেছে, তারা কোনও ROM ব্যবহার করেনি কারণ একটি ROM প্রোগ্রামিং করা বেশ ঝামেলার। ফ্ল্যাশ মেমরি(Flash Memory) fly এ প্রোগ্রাম করা যেতে পারে, আপনি এটি ডাউনলোড করতে পারেন, আপনি ফ্ল্যাশ এ সঞ্চয় করতে পারেন। এবং এটি একটি অবিচ্ছিন্ন মেমরি, এমনকি আপনি পাওয়ার স্যুইচ অফ(Power switch off) করে প্রোগ্রামটি অদৃশ্য হবে না। এবং কিছু RAM - random access memory রয়েছে, যা আপনি ডেটা সঞ্চয় করতে ব্যবহার করতে পারেন। এটির মধ্যে 96 কিলোবাইট রয়েছে।

এখানে একটি USB ইন্টারফেসও রয়েছে যার মাধ্যমে আপনি এই বোর্ডটি PC বা ডেস্কটপ বা ল্যাপটপের মতো অন্যান্য ডিভাইসের সাথে সংযুক্ত করতে পারেন এবং এই বোর্ডটি mbed করা আছে। ঠিক আছে, mbed প্ল্যাটফর্মের মতো; যদি কোনও বোর্ড mbed সংস্করণ করা থাকে তবে mbed এর অধীন উপলব্ধ সমস্ত সরঞ্জাম এই বোর্ডের সাথে ব্যবহার করা যেতে পারে।

তাই, mbed এমন সরঞ্জামগুলির একটি সেট যা প্রাথমিকভাবে এন্ডেড সিস্টেম (embedded system) বোর্ডগুলির প্রস্তুতকারীদের জন্য বোঝানো হয়; তারা অনেকগুলি ইউটিলিটি তৈরি করেছে যা প্রক্রিয়াটিকে খুব সহজ করে তোলে। এবং, এমন কিছু বৈশিষ্ট্য রয়েছে যা সংহত **ইন্টিগ্রেটেড ডেভেলপমেন্ট**(integrated development) পরিবেশের বিস্তৃত পছন্দকে সমর্থন করে।

বেশ কয়েকটি সফটওয়্যার ভিত্তিক ইন্টারফেস রয়েছে যার মাধ্যমে আপনি এই বোর্ডগুলি অ্যাক্সেস করতে পারবেন, আপনি এই বোর্ডগুলি প্রোগ্রাম করতে পারেন, এবং আরও। এটি আমরা পরে দেখতে পাব কারণ আমাদের পরীক্ষায় আমরা mbed.org- এ উপলব্ধ সরঞ্জামগুলি ব্যবহার করব। সেখানে আমরা কীভাবে ইন্টারফেস করব, কীভাবে ব্যবহার করব এবং কিভাবে এই বোর্ডটি প্রোগ্রাম করব তা আমরা দেখব।

(Refer Slide Time: 04:42)



এখন, কিছু নির্দিষ্ট বিশদ এখানে উল্লেখ করা হয়েছে। এই বোর্ডে আমি যেমন উল্লেখ করেছি যে আমাদের কাছে STM32F401 মাইক্রোকন্ট্রোলার রয়েছে, সেখানে একটি সংস্করণ RET6 রয়েছে। এতে অন্যান্য অনেক কিছুর সাথে ARM CORTEX M4CPU রয়েছে। একটি FPU রয়েছে যা 84 মেগাহার্টজ ঘড়ি(clock), ফাংশনাল প্রসেসিং ইউনিট, 512 কিলোবাইট ফ্ল্যাশ, 96 কিলোবাইট SRAM, এবং প্রচুর I/O PORT চলছে। এবং অন্যান্য পেরিফেরালগুলি(peripherals) সরবরাহ করা হয় যা **এন্ডেড সৃষ্টি**(embedded DEVELOP)কে খুব সহজ করে তোলে।

এবং, বোর্ডের অংশ হিসাবে উপলব্ধ যে দুটি প্রকারের এক্সটেনশন সংযোজক (EXTENSION CONNECTOR) রয়েছে। একটি হ'ল আরডুইনো(Arduino) UNO রিভিশন 3 সামঞ্জস্যপূর্ণ। আপনি

দেখতে পাচ্ছেন যে আজ এস্বেডেড সিস্টেম ডিজাইনের উন্নয়নের জন্য আরডুইনো(Arduino) একটি খুব জনপ্রিয় প্ল্যাটফর্ম। আরডুইনো গণনাগতভাবে খুব সহজ, ARMএর মতো শক্তিশালী নয়। তবে, কারণ আপনি যখন কোনও এআরএম প্রসেসরের আশেপাশে কিছু **সৃষ্টি** (DEVELOP) করছেন তখন আরডুইনো এত জনপ্রিয়, এটি আরডিনো ভিত্তিক সংযোগগুলি তৈরি করতে হবে এমন আরও কিছু সাবসিস্টেম থাকতে পারে। যাতে এই বোর্ডটি একটি আরডুইনো ধরণের **সংযোগ** (CONNECTOR) সরবরাহ করে।

এছাড়াও, এই STM মরফো এক্সটেনশন(Morpho extension) পিনগুলি আপনাকে বোর্ডের ভিতরে বসে থাকা এসটিএম মাইক্রোকন্ট্রোলারের সমস্ত ইনপুট আউটপুট পিনগুলিতে আপনাকে সম্পূর্ণ অ্যাক্সেস(Access) সরবরাহ করে। কিছু ইনবিল্ট ইন্টারফেস সরবরাহ করা হয়েছে যা আপনাকে বেশ কয়েকটি স্ট্যান্ডার্ড ইন্টিগ্রেটেড ডেভলপমেন্ট এনভায়রনমেন্ট(Standard integrated environment) সাথে ইন্টারফেস করতে দেয় যা আপনি বাইরে থেকে কিছু ডিবাগার(Debugger) চালাতে পারবেন, যদি আপনার বোর্ড সঠিকভাবে কাজ না করে তবে আপনি এই ডিবাগারগুলি(Debugger) ব্যবহার করে বোর্ডটি পরীক্ষা বা নির্ণয় করতে পারেন।

আপনি ভার্চুয়াল সিরিয়াল পোর্ট বা ভার্চুয়াল কম পোর্ট ব্যবহার করতে পারেন যার মাধ্যমে আপনি কিছু ডেটা আপনার PCতে ফেরত পাঠাতে পারেন এবং আপনার PC স্ক্রিনে প্রদর্শন করতে পারেন। এবং, ঠিক যখনই আপনি PC বা ল্যাপটপে কোনও জিনিস অধিরহন(mount) করে দেখেন যে ডিভাইসটিকে ড্রাইভ হিসাবে স্থাপন করা হয়েছে, আপনি এই নিউলিও বোর্ডটি যখন USB সংযোজকের মাধ্যমে কোনও PCতে সংযুক্ত করেন যা একটি USB ডিস্ক ড্রাইভ(Disk drive) হিসাবেও মাউন্ট(mount) হয়ে যায়। আপনি যখন এই বোর্ডটি প্রোগ্রাম করবেন আপনি সেই প্রোগ্রামটিকে ডিস্ক ড্রাইভ আইকন(Disk drive icon)টিতে টানুন এবং ফেলে দিন, সেই প্রোগ্রামটি স্বয়ংক্রিয়ভাবে ডাউনলোড হয়ে যাবে।

এই ড্রাইভারটি স্বয়ংক্রিয়ভাবে পরিবেশের অংশ হিসাবে ইনস্টল করা আছে। সুতরাং, এটি বিকাশকারী(developer) দের পক্ষে খুব সহজ হয়ে যায়। এই সমস্ত জিনিস আমরা পরে দেখব যখন আমরা আপনাকে প্রদর্শন(demonstration) করবো।

(Refer Slide Time: 08:25)

- Input/Output
 - 50 general-purpose I/O pins with external interrupt capability.
 - 12-bit ADC with 16 channels.
 - 7 general-purpose timers.
 - 2 watchdog timers.
 - USB 2.0 interface.
 - And several others ...
- User LED (LD2)
- Two push buttons: USER and RESET.

বাহ্যিক ইন্টারফেস সম্পর্কে খুব দ্রুত সংক্ষিপ্তসার এখানে দেখানো রয়েছে। এখানে প্রচুর পরিমাণে আইও PORT সরবরাহ করা হয়, 50 সাধারণ-উদ্দেশ্য আইও পিন সরবরাহ করা হয়। এখন, একটি ভাল বিষয় হ'ল এই সমস্ত আইও পিনগুলি বহিরাগত বিন্দিত উৎসগুলির সাথে সংযোগ করতেও ব্যবহার করা যেতে পারে। কিছু INTERRUPT রয়েছে যা আমরা পরে আলোচনা করব, তবে বিষয়টি হ'ল আমরা যদি এমন একটি অ্যাপ্লিকেশন বাস্তবায়ন করতে চাই যেখানে কিছু বাহ্যিক ডিভাইসগুলি INTERRUPTগ্রহণ করছে সেখানে আপনি এই 50 টি পিনের সাথে সংযোগ করতে পারেন।

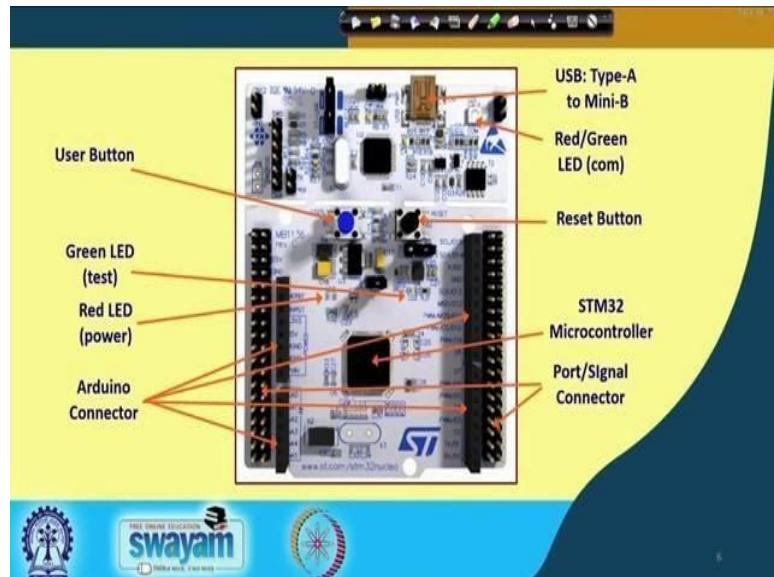
এটি কিছু পূরানো প্রসেসরের বিপরীতে; আসুন আমরা 8085 বলি যা আপনারা অনেকেই হ্যাত জানতে পারবেন যাঁর কিছু উত্সর্গীকৃত বিন্দিত পিন ছিল। সেখানে TRAP, RST 7.5, 6.5, 5.5 এবং আইএনটিআর নামে কিছু পিন ছিল; পাঁচটি ইন্টারাপ্ট (INTERRUPT) লাইন রয়েছে যা উত্সর্গীকৃত। আপনি যখন একটি ইন্টারাপ্ট (INTERRUPT) ডিভাইসটি সংযোগ করতে চেয়েছিলেন তখন আপনাকে এই পাঁচটি লাইনের একটিতে সংযোগ করতে হয়েছিল। তবে, এখানে আপনি যে কোনও জায়গায় সংযোগ করতে পারেন, কিছু পিন আপনি সাধারণ উদ্দেশ্য হিসাবে ব্যবহার করতে পারেন IO আপনি ইন্টারাপ্ট (INTERRUPT) হিসাবেও ব্যবহার করতে পারেন।

এবং, আরেকটি বিষয় হ'ল ডিজিটাল কনভার্টারের সাথে এনালগের একটি নির্মাণ রয়েছে। এঙ্গেড থাকা সিস্টেম ডিজাইনের জন্য A / D converter গুলি খুব গুরুত্বপূর্ণ কারণ আপনি যে বহিরাগত প্যারামিটারগুলি বুঝতে চান সেটি প্রকৃত অ্যানালগ। তারা ক্রমাগত বিভিন্ন প্যারামিটারগুলি ভোল্টেজে অনুবাদ করা যায়। সেগুলি ইন্টারফেস করার জন্য যদি আমাদের বোর্ডে A / D converter থাকে তবে এটি খুব সুবিধাজনক হয়ে ওঠে।

বোর্ডের ভিতরে একটি 12-বিট A/D converter রয়েছে এবং আপনি এটিতে 16 টি ইনপুট ডিভাইস সংযোগ করতে পারেন। বোর্ডে বেশ কয়েকটি টাইমার সরবরাহ করা হয়েছে। এর মধ্যে এই হল 7 টা, তাদের মধ্যে দুটো ওয়াচডগ টাইমার(Watchdog timer) এবং আরু উল্লেখ করেছি একটি USB 2.0

ইন্টারফেস রয়েছে। এটি প্রধানত PC বা ল্যাপটপের মতো একটি হোস্ট সিস্টেম(Host System) সাথে সংযোগ করতে ব্যবহৃত হয়, কিছু LED রয়েছে, কিছু পুশ বোতামের স্যুইচ রয়েছে।

(Refer Slide Time: 11:13)



এখানে আপনার বোর্ডের একটি ছবি রয়েছে। এখন, আমি আপনাকে এই বোর্ড দেখাতে চাই।

(Refer Time: 11:25).

(Refer Slide Time: 11:55)



এটি হল আমরা যে নিউক্লিয়ার বোর্ড(Neuclear Board) কথা বলছি। আপনি যদি এর দুটি দিক দেখতে পান তবে অনেক সংযোজক (CONNECTOR) দেখতে পাবেন। কিছু কালো ছোট আকারের সংযোজক (Connector) রয়েছে এবং অভ্যন্তরীণ অংশে আপনি দেখতে পাচ্ছেন, এটি হল আড়তুইলো

ব্যবহারযোগ্য সংযোজক (compatible connector)। আপনি যদি কোনও আরডুইনো বোর্ডের দিকে লক্ষ্য করেন তবে তাদের এর মতো সঠিক আকারের সংযোজক(CONNECTOR) থাকবে, আপনি সেখান থেকে সরাসরি এটির সাথে একটি কেবল সংযোগ করতে পারেন। তবে, আপনি যদি উভয় পক্ষের অন্যান্য পিনগুলি উপলব্ধ দেখতে পান তবে পিনগুলি আপনাকে বোর্ডে অভ্যন্তরীণ মাইক্রোকন্ট্রোলার তৈরি করছে এমন সংকেত সরবরাহ করে। এই সমস্ত সংকেতগুলি এই সিগন্যাল পিনগুলি থেকে অ্যাক্সেস করা যায়, বা এটির একটি ছোট উপসেট এই আরডুইনো ইন্টারফেস পিনগুলির মধ্যে পাওয়া যায়, আপনি সেখান থেকেও এটি পেতে পারেন।

আপনি দেখতে পাচ্ছেন যে এখানে একটি USB ইন্টারফেস রয়েছে, এখান থেকে আপনি একটি USB কেবল সংযুক্ত করতে পারেন। এবং আপনি মাইক্রোকন্ট্রোলার চিপটি এখানে বসে দেখতে পাচ্ছেন এবং অন্যান্য আনুষাঙ্গিক জিনিস রয়েছে। এখন আসুন দেখুন এই বিভিন্ন সাবসিস্টেমগুলিতে কী রয়েছে।

আপনি যদি এখানে দেখেন তবে এটি একই বোর্ড যা আমি দেখিয়েছি। এই বোর্ডে এই কালো সংযোজকগুলি(Black Connector) হ'ল আমরা বলছিলাম যে আরডুইনো সংযোজক(CONNECTOR)। এখন, এই বোর্ডে আপনি দেখতে পাবেন যে কয়েকটি লেবেল A5, A4, A3, A2, A1, A0 দেওয়া হয়েছে এটি এনালগ ইনপুট পোর্ট; A/D CONVERTOR আমি আপনাকে বলেছিলাম যে আপনি এই PORTগুলির সাথে সরাসরি একটি এনালগ ইনপুট সংযোগ করতে পারেন। এখানে কিছু ভোল্টেজ উত্স রয়েছে যা 5 ভোল্ট, 3.3 ভোল্ট, গ্রাউন্ড এগুলি উপলব্ধ এবং বিভিন্ন ইনপুট আউটপুট পোর্ট পিনগুলি এখানে উপলব্ধ। এর মধ্যে কয়েকটি হ'ল ডিজিটাল ডেটা পিন, তাদের মধ্যে কয়েকটি **পালস উইথ মডুলেশান**(PULSE WIDTH MODULATION) পিন; এগুলি আমরা পরে দেখব।

এখানে, এসটিএম 32 মাইক্রোকন্ট্রোলার পিনগুলি পাওয়া যায় যা সরাসরি এর থেকে আসে। সমস্ত PORT পিন এখানে উপলব্ধ। A, B, C তিনটি PORT রয়েছে; এগুলি 16-বিট পোর্ট এবং আরও অনেকগুলি সিগন্যাল লাইন রয়েছে।

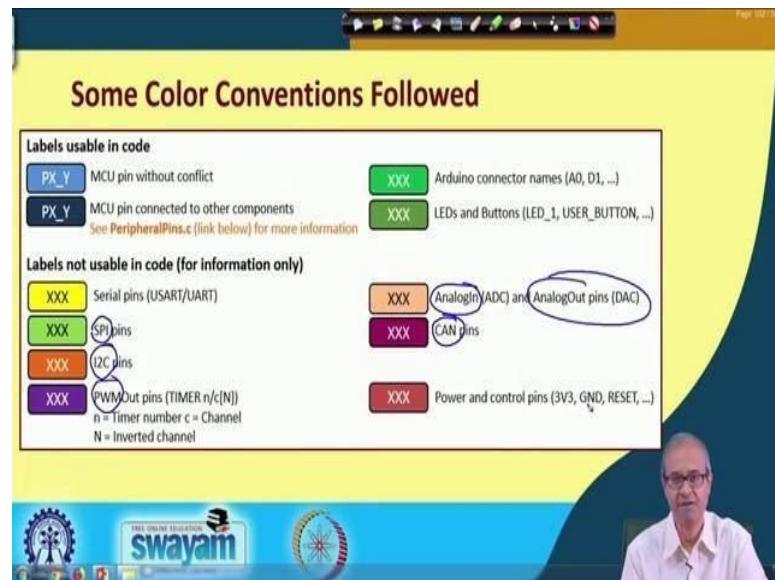
এছাড়াও আপনি শীর্ষে দেখতে পাচ্ছেন আমাদের কাছে USB ইন্টারফেস রয়েছে, এখান থেকে আপনি USB কেবলটি এটি PC-র সাথে সংযুক্ত করতে পারবেন, এখানে একটি রিসেট(Reset) বোতাম রয়েছে আপনি মেশিনটি পুনরায় সেট করতে পারেন। এবং ইঙ্গিতের জন্য কয়েকটি ছোট LED সংযুক্ত রয়েছে, একটি এলইডি এখানে রয়েছে, একটি LED এখানে রয়েছে, একটি LED এখানে রয়েছে এবং কয়েকটিতে বোতাম স্যুইচও রয়েছে।

রিসেট বোতাম বাদে অন্য একটি পুশ স্যুইচ(Push Switch) রয়েছে যা আপনি কোনও প্রোগ্রামে ব্যবহার করতে পারেন। আমরা একটি ছোট প্রোগ্রাম লিখতে পারি এবং সেখান থেকে ইনপুট নিতে পারি। এখন কথাটি হ'ল আপনি যখনই এস্বেডেড অ্যাপ্লিকেশনটি **গঠন** (develop)করতে এই বোর্ডটি ব্যবহার করছেন তখন আপনাকে এই বোর্ডের সাথে কিছু সংযোগ করতে হবে। আপনাকে অবশ্যই বুঝতে হবে যে এই সংযোজক (CONNECTOR) গুলির উপরে কোন সংকেত লাইনগুলি উপলব্ধ কেননা আপনাকে

এই সংযোজক (CONNECTOR) গুলির মাধ্যমে আপনার সমস্ত সংযোগ তৈরি করতে হবে। তাদের আরডুইনো পিনগুলির সাথে তাদের সংযুক্ত থাকা সহজ হবে কারণ বেশিরভাগ সময় আপনার কেবল কয়েকটি সংযোগের প্রয়োজন হবে, তবে এর বাইরে আপনার এখানে বিশদ সংযোজক(CONNECTOR) পিনগুলি ব্যবহার করতে হতে পারে।

আসুন আমরা দ্রুত এই পিনগুলি দেখতে পারি যে কিছু ধরণের পিন রয়েছে।

(Refer Slide Time: 16:47)



এর আগে কিছু কালার কনভেনশান (color convention) রয়েছে যা সেই পিনের বিবরণে অনুসরণ করা হয়। আপনি দেখতে পারেন এমন বিভিন্ন রঙ দেখুন, তারা বিভিন্ন শ্রেণীর পিনগুলি নির্দেশ করে। এই ক্লিয়েল রঙের মতো সংঘাত ছাড়াই মাইক্রোকন্ট্রোলার পিনগুলি নির্দেশ করে; দ্বন্দ্ব ছাড়াই তারা ব্যবহারকারীর কাছে উপলব্ধ, তারা অন্য কোথাও সংযুক্ত নেই। তবে, এই ধূসর বর্ণগুলি হল অন্যান্য উপাদানগুলির সাথে সংযুক্ত কিছু পিন। হয়তো আপনি বোর্ডে দেখেছেন কিছু এলএভি রয়েছে, কিছু সুইচ রয়েছে, তাই অনেক কিছুই আছে সম্ভবত এই পিনগুলির মধ্যে কিছু অভ্যন্তরীণভাবে সেই ডিভাইসগুলির সাথে সংযুক্ত রয়েছে। সুতরাং, আপনি যখন বাইরে থেকে এগুলি ব্যবহার করেন তখন আপনার সতর্ক হওয়া উচিত।

এছাড়াও এই হলুদগুলি কিছু সিরিয়াল পিন উল্লেখ করে যা থাকে **USART - উনিভারসাল সিরিয়াল অ্যাসিস্টনাস রিসিভার ট্রান্সমিটার** (universal serial asynchronous receiver transmitter)।

তারপরে SPI পিন, I2C পিন, PWM পিন এবং এই আরডুইনো CONNECTOR পিনগুলি ইতিমধ্যে উল্লেখ করেছি। অবশ্যই এই বোর্ডে কোনও অ্যানালগ আউটপুট (Analog output) পিন নেই, কেবল এনালগ ইনপুট পিন রয়েছে এবং সেখানে CAN পিন নামে কিছু আছে।

আমরা দেখতে পাচ্ছি যে এই সংক্ষিপ্ত শব্দটির অর্থ SPI, I2C, CAN এবং অবশ্যই, এখানে কিছু পাওয়ার পিন রয়েছে 3.3 ভোল্ট, 5 ভোল্ট, গ্রাউন্ড।

(Refer Slide Time: 18:50)

The Nomenclatures

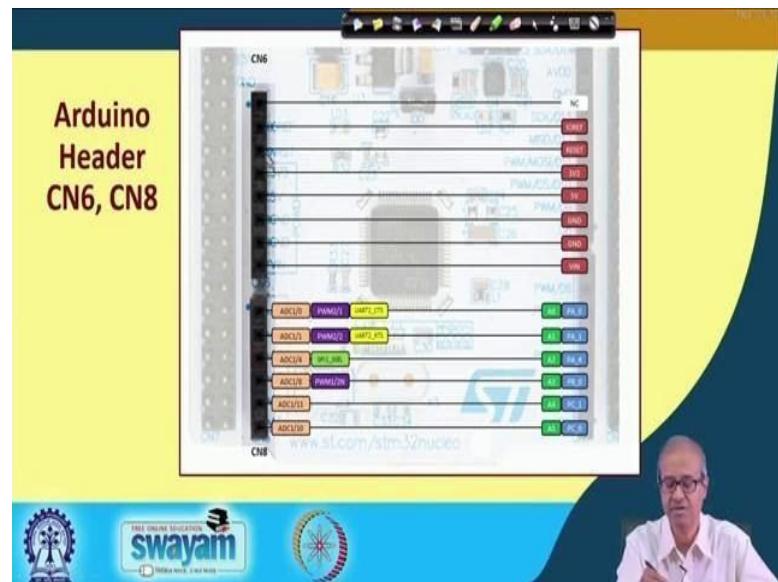
- SPI: Serial Peripheral Interface
- I2C: Inter-Integrated Circuit
 - Serial communication bus, widely used for attaching lower-speed peripheral ICs to microcontrollers over short distance.
- PWM: Pulse Width Modulation
- CAN: Controller Area Network
 - A robust vehicle bus standard, designed to allow microcontrollers and devices to communicate with each other without a host computer.

তবে, প্রথমে নামটি দেখুন। SPI এর অর্থ সিরিয়াল পেরিফেরিয়াল ইন্টারফেস(Serial Peripheral Interface)। আপনি যখনই চান কিছু সিরিয়াল যোগাযোগ অ্যাপ্লিকেশন গঠন (DEVELOP) করতে এই পিনগুলি ব্যবহার করতে পারেন।

I2C একটি স্ট্যান্ডার্ড বাস, এর অর্থ ইন্টার-ইন্টিগ্রেটেড সার্কিট(Inter integrated circuit)। এটি একটি সিরিয়াল যোগাযোগ বাস যা খুব ঘন দূরস্থে কিছু কম গতির ডিভাইসগুলিকে একটি মাইক্রোকন্ট্রোলারের সাথে সংযুক্ত করতে ব্যবহৃত হয় frequently এন্ডেডে সিস্টেম অ্যাপ্লিকেশনটির জন্য এই I2C ইন্টারফেসটি খুব কার্যকর হতে পারে, যদি আমাদের কাছে এমন কোনও ডিভাইস থাকে যা I2C সামঞ্জস্যপূর্ণ থাকে তবে আপনার এই I2C সংযোগ থাকতে হবে।

PWM এর অর্থ **পালস উইথ মডুলেশান**(Pulse Width Modulation), এটি আমরা দেখব। **পালস উইথ মডুলেশান**(Pulse Width Modulation) বাহ্যিক ডিভাইসগুলি নিয়ন্ত্রণ করার একটি খুব সুবিধাজনক উপায়। এবং কন্ট্রোলার এরিয়া নেটওয়ার্ক (Controller Area Network)এর সংক্ষিপ্ত রূপ CAN।

(Refer Slide Time: 20:40)

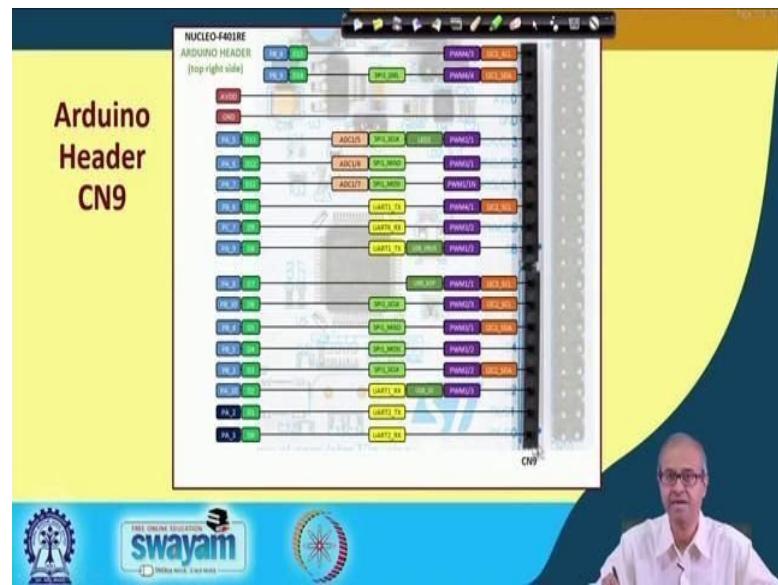


আসুন আমরা ইতিমধ্যে বোর্ডে দেখেছি এমন কিছু সংযোজক (CONNECTOR) যাকে খুব দ্রুত দেখি। তাদের অনেকগুলি খুব ছোট আপনি সম্ভবত খুব পরিষ্কারভাবে পড়তে পারবেন না তবে আমি আপনাকে বলছি। বোর্ডের বাম দিকে এইগুলি আড়ডুইনো সংযোজক (CONNECTOR)গুলি গীচেরটিকে CN 8 বলা হয়, উপরেরটিকে CN 6 সংযোজক (CONNECTOR)বলা হয়।

CN 8-তে আপনি দেখতে পাবেন যে এনালগ ইনপুট পোর্ট রয়েছে, তার মধ্যে ছয়টি পিনের কয়েকটি মাল্টিফাংশনাল(Multifunctional) সরবরাহ করা হয়েছে। এগুলিকে হয় এনালগ পিন হিসাবে ব্যবহার করা যেতে পারে অথবা এগুলি PWM পিন হিসাবে ব্যবহার করা যেতে পারে বা তারা UART পিন হিসাবে ব্যবহার করা যেতে পারে। যেমনটি আমি আপনাকে ARM-এ বলেছিলাম দর্শনের বিষয়টি হ'ল তারা বিভিন্ন ফাংশনের জন্য ডেডিকেটেড পিন ব্যবহার করেন না, একই পিন আপনি বিভিন্ন অ্যাপ্লিকেশনের জন্য ব্যবহার করতে পারেন। এবং এগুলি আসলে এই পোর্টগুলির সাথে A0, A1, A4, B0, B1, C0 সংযুক্ত।

একইভাবে, উপরের লাইনগুলি প্রাথমিকভাবে বিদ্যুৎ সরবরাহের লাইন, 5 ভোল্ট, গ্রাউন্ড, 3.3 ভোল্ট, পুনরায় সেট করা ইত্যাদি। এগুলি হ'ল আরডুইনো সংযোজক(CONNECTOR) প্রথম সেট।

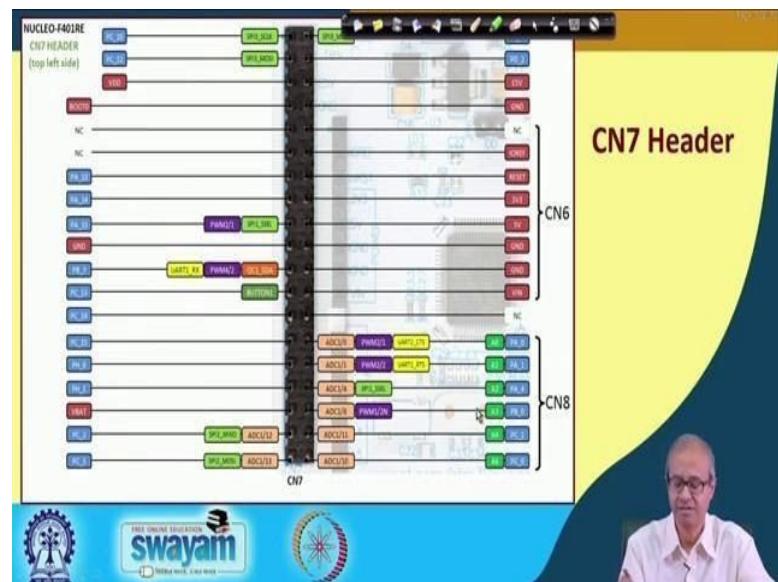
(Refer Slide Time: 22:25)



এবং, আরডুইনো CONNECTORগুলির দ্বিতীয় সেটটি যে ডানদিকে প্রদর্শিত হবে তাকে CN9 CONNECTOR বলা হয়, এটি একটি বৃহত আকারের সংযোজক। এতে আপনি দেখতে পাচ্ছেন এমন অনেকগুলি সংকেত রায়েছে যেমন PWM PORTগুলি, I2C পোর্টগুলি, এগুলি USB, UART পিন, এগুলি কিছু অতিরিক্ত এনালগ ইনপুট পিন এবং এগুলি কিছু পোর্ট পিন।

সুতোং, আপনি যখন অ্যাপ্লিকেশন তৈরি করছেন তখন এই পিনগুলি আপনি ব্যবহার করবেন।

(Refer Slide Time: 23:39)



এই বিস্তারিত পিনগুলি যা বোর্ডের ভিতরে বসে মাইক্রোকন্ট্রোলার থেকে আসে। বাম দিকে এটিকে CN7 বলা হয়, এটি হ'ল বড সংযোজক (CONNECTOR), আপনি যে পিনগুলি দেখেছেন এবং এখানে পোর্ট

A, পোর্ট B, পোর্ট C - সেখানে তিনটি 16-বিট পোর্ট রয়েছে যা মাইক্রোকন্ট্রোলারে পাওয়া যায়। এই সংযোজক (CONNECTOR)গুলিতে সমষ্টি 48 পিন উপলব্ধ।

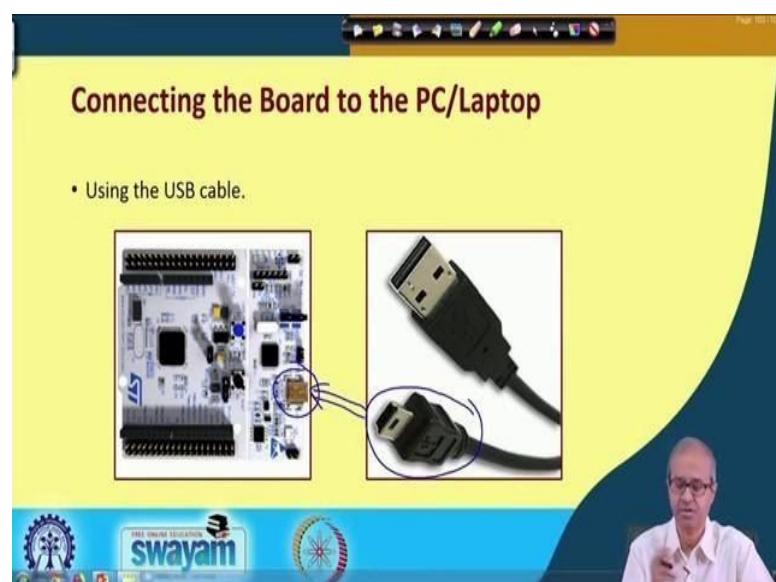
এছাড়াও অন্যান্য অনেক জিনিস যেমন বিদ্যুৎ সরবরাহ ইত্যাদি।

(Refer Slide Time: 24:36)



একইভাবে, ডানদিকে অন্য PORTটি এটি CN 10। এখানে আপনার কাছে Port B, Port A, PWM PORT, ADC PORT, SPI, I2C এবং আরও অনেকগুলি PORT রয়েছে। আপনার ব্যবহার এবং সংযোগের জন্য যা কিছু দরকার তা আপনি এই CONNECTOR গুলো থেকে ব্যবহার করতে পারেন।

(Refer Slide Time: 25:13)



আপনার এরকম একটি USB কেবল দরকার, সংক্ষিপ্ত দিকটি এখানে সংযুক্ত হবে এবং অন্য পাশটি একটি স্ট্যান্ডার্ড টাইপ C সংযোজক হবে, এটি আপনার PC বা ল্যাপটপে যাবে। আমরা এইভাবে

পরীক্ষাটি করবো। PCতে, আমরা অ্যাপ্লিকেশনটি গঠন (DEVELOP) করব, আমরা প্রোগ্রামটি লিখব, আমরা কোডটি সংকলন করব, এবং এই বোর্ডের সাহায্যে আমরা সমস্ত ডিভাইস, সেন্সর, **একচুয়েটার (actuator)**এর সাথে সমস্ত কিছুতে ইন্টারফেস করব। এবং, কোড লেখার পরে আমরা বোর্ডে কোডটি ডাউনলোড করব এবং আপনি যা ডাউনলোড করেছেন তা চলতে শুরু করবে।

এটির সাথে আমরা নিউক্লিয়ার বোর্ড(Neuclear Board)র এই খুব সংক্ষিপ্ত পরিচিতির শেষে এসেছি। এই কোর্সের অংশ হিসাবে, আমরা আপনাকে কিছু প্রদর্শন (demonstration) দেখাব। তবে, আপনি যদি সত্যিই এন্ডেড থাকা সিস্টেম ডিজাইনটি হাতে-কলমে(Hand-on) পদ্ধতিতে শিখতে আগ্রহী হন তবে আমরা দৃঢ় ভাবে সুপারিশ করি যে এই বোর্ডগুলির কয়েকটি সংগ্রহ করা উচিত, এবং আপনি যে পরীক্ষাগুলি প্রদর্শন বা প্রদর্শিত হোক তা আপনাদের অবশ্যই নিজেরাই করা উচিত।

আমরা আমাদের আলোচনা চালিয়ে যাব। আমাদের পরবর্তী বক্তৃতায় আমরা এই বোর্ডের কয়েকটি I/O পোর্ট বিশদ সম্পর্কে কথা বলব যা আমরা এখনই দেখেছি, বিশেষত আমরা **পালস উইথ মডুলেশান(PULSE WIDTH MODULATION)** PORTগুলি, ইন্টারাপ্ট(INTERRUPT) লাইনগুলি এবং এগুলি সম্পর্কে কথা বলব। এবং এরপরে আমরা এনালগ ইনপুট(Analog Input), আউটপুট(Output) এবং কিছু অন্যান্য অ্যাপ্লিকেশন এবং প্রদর্শন(demonstration)দেখব।

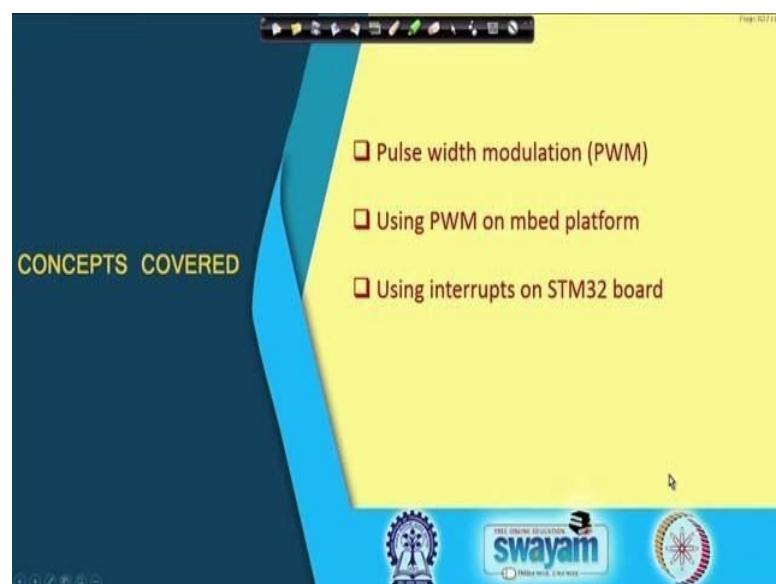
ধন্যবাদ।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 11
PWM and Interrupt on STM32F401

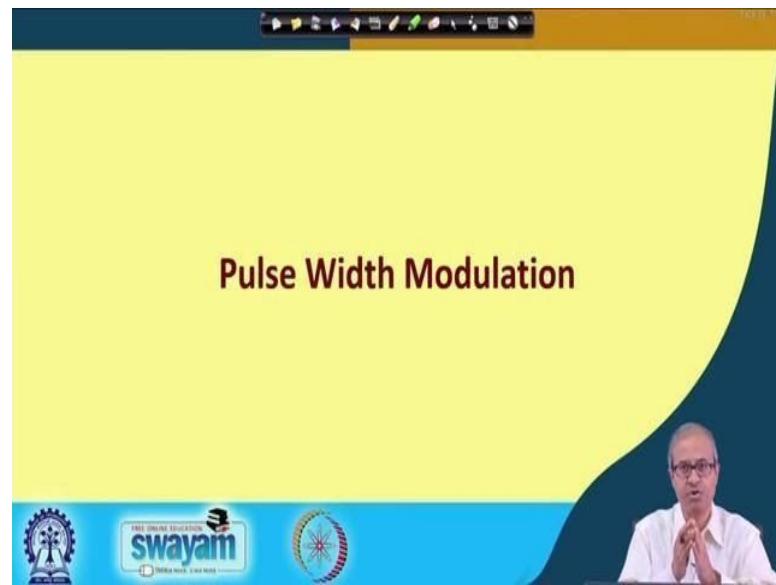
এই বক্তৃতাতে, আমরা এমন কিছু অনন্য ইনপুট -আউটপুট বৈশিষ্ট্য সম্পর্কে কথা বলব যা আপনি আমাদের শেষ বক্তৃতা STM 32 এর বোর্ডে দেখেছিলেন।

(Refer Slide Time: 00:43)



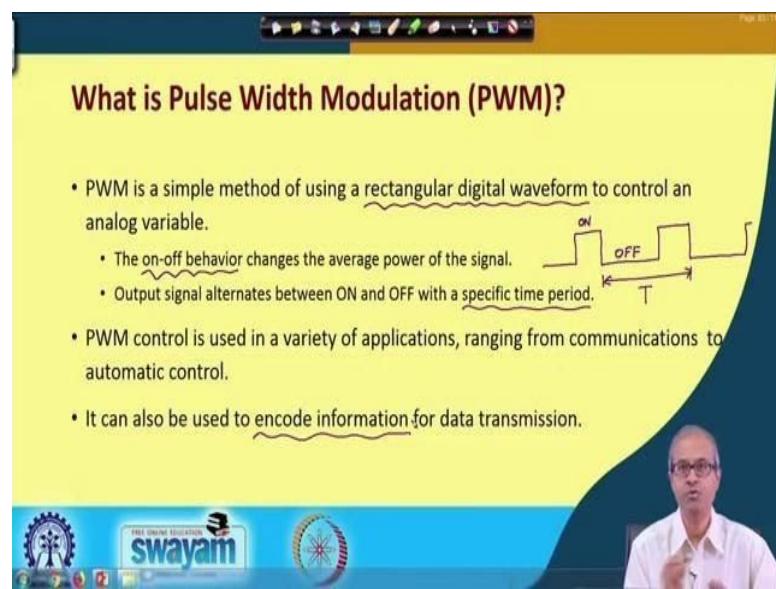
এই বক্তৃতায় আমরা প্রথমে **পালস উইথ মডুলেশন** (pulse width modulation) সম্পর্কে কথা বলব যাকে সংক্ষেপে আমরা PWM বলি, তারপরে দেখবো কীভাবে নির্দিষ্ট বোর্ডে এই PWM ব্যবহার করতে পারি এবং শেষ পর্যন্ত দেখবো কীভাবে এই বোর্ডে ইন্টারাপ্ট (INTERRUPT) ব্যবহার করতে হয়।

(Refer Slide Time: 01:08)



প্রথমে **পালস উইথ মডুলেশান (PULSE WIDTH MODULATION)**সম্পর্কে কথা বলা যাক। এটি এমন একটি কৌশল যেখানে আপনি কোনও ধরণের ডিজিটাল মানের সমতুল্য অ্যানালগ (analog) মানের মধ্যে এনকোড (encode) করতে পারেন, তবে সরাসরি নয়; এটিকে এনকোড(ENCODE) করার একটি পরোক্ষ উপায় আছে। এই বক্তৃতাটি এগিয়ে যাওয়ার সাথে সাথে আমরা এটি বু�তে পারবো।

(Refer Slide Time: 01:40)



প্রথমে আমাদের **পালস উইথ মডুলেশান (PULSE WIDTH MODULATION)**কী তা দেখে নিতে হবে। প্রথম জিনিসটি হ'ল PWM ব্যবহার করে আমরা একধরণের আয়তক্ষেত্রাকার ডিজিটাল তরঙ্গরপ তৈরি করতে পারি এবং একটি নির্দিষ্ট সময়কাল T থাকবে।

PWM আমাদের এমন এক ধরনের বৈশিষ্ট্য আনতে সহায়তা করে যা আমরা এনকোড(ENCODE) করার চেষ্টা করছি এমন কিছু প্যারামিটারের সাথে সমানুপাতিক, এটি অন-অফ(On-Off) বৈশিষ্ট্য থেকে আসে। যখন এই তরঙ্গরপ্তি বেশ মানে থাকে তখন আমরা বলি এটি ON আছে এবং যখন এটির মান কম থাকে তখন আমরা বলি যে এটি OFF রয়েছে। আগ্রহের বিষয়টি হল আমাদের ওয়েভফর্মটি কত ক্ষণ ON থাকে এবং এটি কতক্ষণ OFF থাকে। নির্দিষ্ট সময়কাল সহ আউটপুট সিগন্যাল ON এবং OFF সময়কালের মধ্যে পুনরাবৃত্তি হবে; অবশ্যই, এই ON এবং OFF সময়টিও একটি বিশেষ কেসের জন্য স্থির করা হবে।

PWM বিভিন্ন অ্যাপ্লিকেশনে ব্যবহার করা যেতে পারে। আমরা ডেটা যোগাযোগের জন্য এবং নিয়ন্ত্রণ অ্যাপ্লিকেশনগুলির জন্যও ব্যবহার করতে পারি, যা আমরা এমবেডেড সিস্টেম ডিজাইনের জন্য আরও আগ্রহী হতে চাই। এবং যোগাযোগ অ্যাপ্লিকেশনগুলির জন্য আপনি PWM-ও কিছু তথ্য এনকোড (ENCODE) করতে ব্যবহার করতে পারেন, তবে এখানে আলোচনার সুযোগের জন্য আমাদের এটি জানতে হবে না।

(Refer Slide Time: 03:46)

How it works?

- The period is normally kept constant, and the pulse width (or ON time) is varied.
- Duty Cycle:** It is defined as the proportion of time the pulse is ON, expressed as a percentage.

$$\text{Duty Cycle} = (\text{pulse ON time}) / (\text{pulse period}) * 100\% = t_{on} / T * 100\%$$

Average value

"on" time, t_{on}

Period, T

t_{on}

T

Page 38/111

FREE ONLINE EDUCATION
swayam

এখন, আসুন দেখা যাক PWM ঠিক কীভাবে কাজ করে। এই তরঙ্গরপ্তি যা PWM দ্বারা উত্পাদিত হচ্ছে সময়কাল স্থির রাখা হয়। একবার আমরা এই পর্যাকাল T নির্দিষ্ট করে দিলে, এটি অবিচলিত থাকবে। এখন, প্যারামিটারের উপর নির্ভর করে আমরা ENCODE করতে চাই এমন কিছু আছে যা আমরা আলাদা করার চেষ্টা করছি। এটি **পালস টাইথ** (pulse width) বা **অন টাইম** (ON time)। আমরা এটিকে t_{on} বলি; এই t_{on} এমন একটি জিনিস যা আমরা এনকোড (ENCODE) করতে চাইছি এমনভাবে তা প্যারামিটারের অনুপাতে পরিবর্তিত হয়।

এই বিষয়ে আমরা এই আয়তক্ষেত্রগ্রাকার তরঙ্গকারকের(Rectangular waveform) **ডিউটি সাইকেল (DUTY CYCLE)**নামে পরিচিত এমন কিছু সংজ্ঞা দিই। এটা পালস এর সময় এর সাথে সমানুপাতিক। সুতরাং, **ডিউটি সাইকেল(DUTY CYCLE)** T মোট সময় হিসাবে নির্ধারিত হয় যার জন্য পালস এর মোট পর্যায়কাল 100 দিয়ে গুণ হয়।

(Refer Slide Time: 05:51)

• Whatever duty cycle a PWM has, there is an average value, as indicated by the dotted line.

- If the ON time is small, the average value is low; if it is large, the average value is high.
- By controlling the duty cycle, we can control the average value.

• Average value of the signal = $\frac{1}{T} \int_0^T f(t) dt = t_{on}V_H + (1 - t_{on})V_L$

• In general, V_L is taken as 0V for ease of calculation.

- Average value becomes $t_{on}V_H$

PWM-এর সাথে লক্ষ্য রাখার আরেকটি বিষয় হ'ল যে কোনও duty cycle থাকুক না কেন আপনি কিছু গড় মান নির্ধারণ করতে পারেন। তরঙ্গক্রিয়ের গড় বা DC মান কত? আপনারা যারা ফুরিয়ার ট্রান্সফর্ম জানেন, তাদের জন্য যদি আপনি এই ওয়েভফর্মটির ফুরিয়ার ট্রান্সফর্ম নেন তবে আপনি 0 স্তরে কিছু প্রশস্ততা (amplitude) পাবেন যা DC উপাদান হয়ে যাবে।

আসুন ধরে নিই যে এই ভোল্টেজটি 0 ভোল্ট থেকে কিছুটা সর্বাধিক পর্যন্ত পরিবর্তিত হচ্ছে আসুন আমরা 5 ভোল্ট বলি। গড় মান মাঝখানে এই বিন্দুযুক্ত রেখা দ্বারা দেখানো হয়। এই গড় মানটি pulse কর সময় চালিত হয় তার উপর নির্ভর করে। আরও বেশি সময় ধরে এই বিন্দুযুক্ত রেখাটি উপরে উঠতে থাকবে, তবে **পালস(pulse)** গুলি খুব সংকীর্ণ হলে এই বিন্দুযুক্ত রেখাটি নীচে সরবে।

মূলত **ডিউটি সাইকেল (DUTY CYCLE)**টি সামঞ্জস্য করে আমরা তরঙ্গক্রিয়ের গড় মান সমন্বয় করছি। আপনি যদি কেবল একটি অবিচ্ছেদ্য অংশ নেন এবং গড় নেন তবে আপনি সময়ের সময়কালে তরঙ্গক্রিয়ের গড় মান গণনা করতে পারেন সুতরাং, 0 থেকে T সময়কালে অবিচ্ছেদ্য, আসুন আমরা এক পর্যায়কাল $f(t) dt$ বলি। আপনি যদি গণনা করেন তবে আপনি এর মতো মান পাবেন, যেখানে V_H উচ্চ

মানের ভোল্টেজ, V_L হল ভোল্টেজের নিম্ন স্তর। এটি t_{on} পিরিয়ডের জন্য উচ্চ, এটি $t_{off} = 1 - t_{on}$ পিরিয়ডের জন্য কম হবে, কারণ আমরা T দিয়ে ভাগ করছি।

বেশিরভাগ ক্ষেত্রে নিম্ন স্তরের ভোল্টেজকে 0 ভোল্ট হিসাবে নেওয়া হয়। সুতরাং, $V_L = 0$ হলে দ্বিতীয় পদটি অদ্যশ্য হয়ে যাবে। সুতরাং, আপনার গড় মান কেবল t_{on} হবে। V_H । সুতরাং, t_{on} সামঞ্জস্য করে গড় মান অন পিরিয়ড বা অপ্রত্যক্ষভাবে duty cycle এর সমানুপাতিক হয়ে উঠবে। যদি পর্যায়কাল স্থির থাকে, তবে duty cycle t_{on} এর সাথে সমানুপাতিক হয় যা ঘূরে দেখা যায় গড় মানের সাথে সমানুপাতিক।

(Refer Slide Time: 08:50)

How to Extract the Average Value?

- The average value can be extracted from the PWM stream using a low-pass filter.
- If the PWM frequency and the values of R and C are appropriately chosen, V_{out} becomes an analog output.
- Can be used in place of a digital-to-analog converter.

V_{in} ————— R ————— V_{out}
 C

In practice, the filter is not always required.
 Many physical systems have response characteristics that act like low-pass filters.

Page 30/111

এখন, এই PWM ওয়েভফর্ম(Waveform) থেকে, আমরা কীভাবে গড় মানটি বের করতে পারি? সবচেয়ে সহজ উপায় হল আপনি লো পাস ফিল্টার (low pass filter) ব্যবহার করতে পারেন। সবচেয়ে সহজ লো পাস ফিল্টার(low pass filter) হল একটি RC সার্কিট, সিরিজের রেজিস্ট্যান্স (series resistance) এবং সমান্তরালে ক্যাপাসিট্যার(parallel capacitor)। আপনি যদি একটি পালস ওয়েভফর্ম(pulse waveform) প্রয়োগ করেন তবে আপনি একটি আউটপুট ভোল্টেজ পাবেন যা DC মান হবে; প্রদত্ত RC সময় ধ্রুবক একটি উপযুক্ত উপায়ে চয়ন করা হয় যে উপাদানগুলির সর্বনিম্ন ফ্রিকোয়েন্সি কোনও **অ্যাটিন্যুয়েশন**(attenuation) ছাড়াই পাস (pass) করবে, উচ্চতর ফ্রিকোয়েন্সি ক্ষীণ হবে। সুতরাং, আপনি আউটপুটে ভোল্টেজের প্রায় DC মান পাবেন।

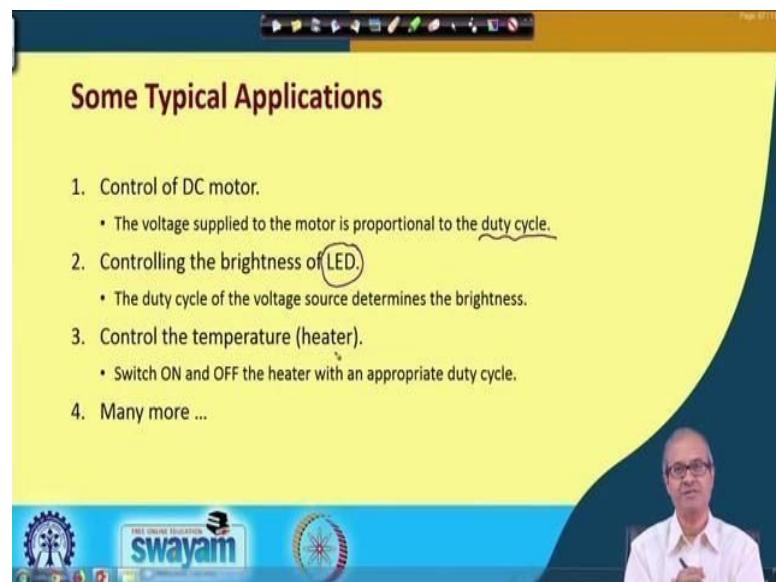
আকর্ষণীয় বিষয়টি লক্ষ্য করুন যে PWM সহ এই জাতীয় সার্কিটটি আপনি ডিজিটাল থেকে এনালগ কনভার্টারের(Digital to Analog convertor) জায়গায় ব্যবহার করতে পারেন। ডিজিটাল থেকে এনালগ কনভার্টার যা আমরা পরে আলোচনা করব তা একটি সার্কিট যা একটি ডিজিটাল ইনপুট নেয়

এবং আউটপুটে আনুপাতিক এনালগ ভোল্টেজ উত্পাদন করে। সুতরাং, ডিজিটাল ইনপুটটি নিয়ন্ত্রণ করে আমরা আউটপুট এনালগ ভোল্টেজ সামঞ্জস্য করতে পারি।

এখন, PWM আপনাকেও অনুন্নত বৈশিষ্ট্য সরবরাহ করে, যেমন আপনি যদি ওয়েভফর্মটির ডিটেটি চক্রটি পরিবর্তন করতে পারেন যা আপনি বলতে পারেন যে এটি একরকম ডিজিটাল ইনপুট, আপনি t_{on} adjust করছেন, এবং আপনি আউটপুটে একটি ভোল্টেজ পেয়ে যান যা t_{on} এর আনুপাতিক। সুতরাং, এই সার্কিটটি ডিজিটাল থেকে এনালগ কনভার্টারের মতো কিছু কাজ করে, ইনপুটটিতে ডিজিটাল মান প্রয়োগ করার পরিবর্তে আমরা PWM ওয়েভফর্মের pulse width ব্যবহার করতে পারি।

লক্ষ্য করার বিষয়টি হ'ল, ধৰন আমি কোনও ডিভাইস নিয়ন্ত্রণ করতে এই সার্কিটটি ব্যবহার করি। ঠিক আছে, এটি একটি হিটার হতে পারে; এটি হালকা হতে পারে, যেখানে আমি বিদ্যুৎ সরবরাহের জন্য একটি pulse তরঙ্গনপ্রয়োগ করছি। বেশিরভাগ সার্কিটগুলির অভ্যন্তরে অন্তর্নির্মিত রোধের (built in resistance) এবং ক্যাপাসিট্যান্স প্রভাব থাকে। এবং ইনপুট পর্যায়ে প্রায়শই অন্তর্নির্মিত লো পাস ফিল্টার(built in low pass filter) থাকে। সুতরাং, বেশিরভাগ ক্ষেত্রে এই লো পাস ফিল্টারটি প্রায়শই প্রয়োজন হয় না। এই PWM পালস ওয়েভফর্মটি আপনি যে ডিভাইসটি নিয়ন্ত্রণ করতে চান তাতে সরাসরি প্রয়োগ করতে পারেন। এবং ইনপুট পর্যায়ে লো পাস ফিল্টার স্বয়ংক্রিয়ভাবে গড় মানটি বের করবে এবং আপনি যা চান তার গড় মানের উপর ভিত্তি করে।

(Refer Slide Time: 11:39)



Some Typical Applications

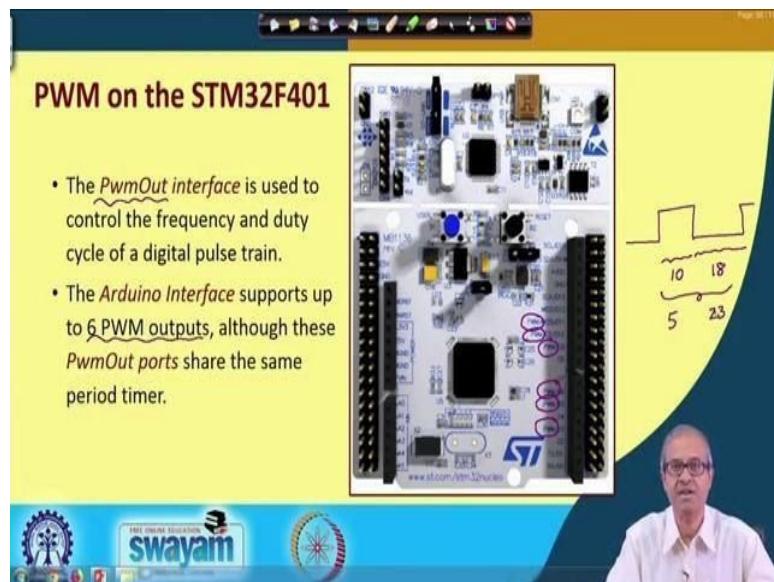
1. Control of DC motor.
 - The voltage supplied to the motor is proportional to the duty cycle.
2. Controlling the brightness of .
 - The duty cycle of the voltage source determines the brightness.
3. Control the temperature (heater).
 - Switch ON and OFF the heater with an appropriate duty cycle.
4. Many more ...

PWM এর কিছু সাধারণ অ্যাপ্লিকেশন এখানে তালিকাভুক্ত করা হল। মনে করুন আপনি একটি DC মোটর নিয়ন্ত্রণ করার চেষ্টা করছেন। এর জন্য প্রায়শই একটি নিয়ন্ত্রণ সংকেত রয়েছে যার সাহায্যে আপনি মোটরটিতে যে বিদ্যুৎ সরবরাহ করছেন তা সামঞ্জস্য করতে পারেন।

আপনি যদি PWM তরঙ্গপটি সরাসরি প্রয়োগ করেন তবে গড় মানটি হবে সমমানের বিদ্যুৎ সরবরাহ। সুতরাং, **ডিউটি সাইকেল**(DUTY CYCLE টি সামঞ্জস্য করে আপনি কার্যকরভাবে বিদ্যুৎ সরবরাহকে সামঞ্জস্য করছেন, যার ফলে মোটরের গতি সামঞ্জস্য করা হচ্ছে। এটি আপনি মনে করেন এমন একটি অ্যাপ্লিকেশন।

হিটারের মতো গ্যাজেটটি ভাবুন বা এমনকি আপনি মাইক্রোওয়েভ ওভেন সম্পর্কে ভাবতে পারেন। আপনি যখন মাইক্রোওয়েভ ওভেনের শক্তি সামঞ্জস্য করেন আসলে কী ঘটে? এখানে একটি PWM মেকানিজম রয়েছে যা মাইক্রোওয়েভ প্রক্রিয়া ON এবং OFF করবে এবং **ডিউটি সাইকেল**(DUTY CYCLE)টি সামঞ্জস্য করা হবে। স্বয়ংক্রিয় হিটার নিয়ন্ত্রণ সিস্টেমের ক্ষেত্রেও একই অবস্থা। আপনি হিটারটি ON এবং OFF করুন, আপনি হিটারটি ON করার সময়টি PWM ওয়েভফর্মের **ডিউটি সাইকেল**(DUTY CYCLE) এর সাথে আনুপাতিক হবে। এবং অন্যান্য অনেক অনুরূপ অ্যাপ্লিকেশন রয়েছে যা আপনি ভাবতে পারেন।

(Refer Slide Time: 13:30)



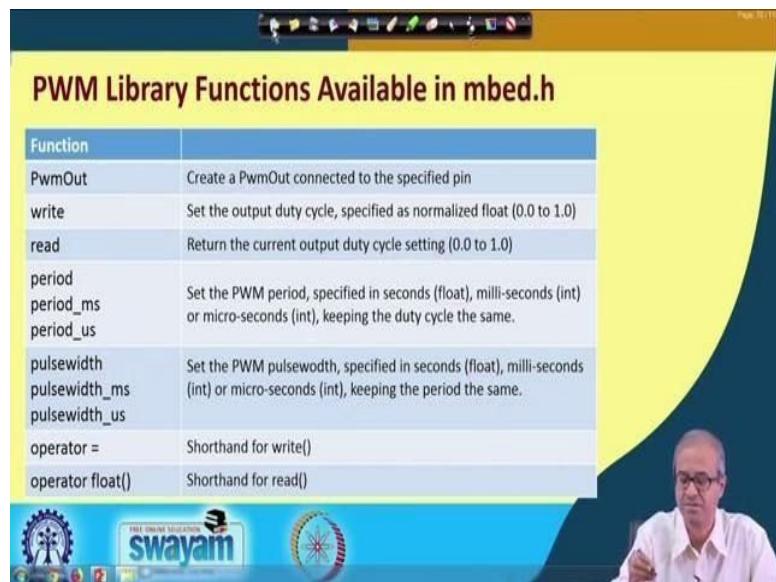
আপনি যদি এসটিএম 32 এফ 401 বোর্ডটি মুরে দেখেন, আপনি যদি আরডুইনো সংযোজকের দিকে তাকান তবে দেখতে পাবেন বেশ কয়েকটি PWM পিন রয়েছে। এখানে ছয়টি PWM পিন রয়েছে। এখানে একটি স্ট্যান্ডার্ড PWM আউট ইন্টারফেস(OUT Interface) রয়েছে যা আমরা এমবেড নামে একটি ধরণের সফটওয়্যার ডেভলপমেন্ট প্ল্যাটফর্ম ব্যবহার করব, এটি আমরা পরে দেখব। এমবেডের অধীনে এই PWM আউট একটি স্ট্যান্ডার্ড লাইব্রেরি যা সরবরাহ করা হয়, এটি আমাদের PWM ব্যবহার করে এমন অ্যাপ্লিকেশনগুলি বিকাশে সহায়তা করবে।

এখন, এই ইন্টারফেসটির কিছু ফাংশন রয়েছে যা উত্পাদিত PWM ওয়েভফর্মের ডিউটি চক্রটি সামঞ্জস্য করতে বলা যেতে পারে। আরডুইনো ইন্টারফেসটি 6 টি PWM আউটপুট সমর্থন করে তবে আপনার যদি এই বিশদ পিনগুলি অ্যাক্সেস করে থাকে তবে আরও অনেক কিছু রয়েছে। সুতরাং, আপনি 6 টিরও বেশি PWM আউটপুট পিন রাখতে পারেন।

মূলত এই PWM সংকেতগুলি ON এবং OFF কিছু টাইমার ব্যবহার করে তৈরি করা হয়। আপনি আউটপুটটি 1 এ রাখতে পারেন, একটি মান দিয়ে টাইমার লোড করতে পারেন এবং একটি ঘড়ি দিয়ে টাইমার গণনা নিচে নেমে আসবে, আপনি 0 না হওয়া পর্যন্ত অপেক্ষা করুন। উদাহরণস্বরূপ, এই অন পিরিয়ডটি 10 এর একটি গণনা মান উপস্থাপন করতে পারে এবং এটি অফ পিরিয়ড 18 এর একটি গণনা মান উপস্থাপন করতে পারে 18 সুতরাং, আমি সেই উপর্যুক্ত মান দিয়ে কাউন্টার বা টাইমারটি লোড করব এবং এটি 0 পর্যন্ত পোঁচানো অবধি গণনা চলতে দেব, এবং এটি 0 এ পোঁচানোর সাথে সাথে আমি এর মান পরিবর্তন করব সংকেত এবং এটি 10, 18, 10, 18, 10, 18 পুনরাবৃত্তি করে।

আমি যখন DUTY CYCLEটি পরিবর্তন করতে চাই তখন মোট আমি 28 টি রাখব তবে কেবল আমি প্রথম অংশটি পরিবর্তন করব। ধরুন আমি প্রথম অংশটি 5 করলাম দ্বিতীয় ভাগটি স্বয়ংক্রিয়ভাবে 23 এ সামঞ্জস্য হবে। এই পরিমাণটি 28 হবে কারণ সময়কাল একই থাকবে। DUTY CYCLEটি সামঞ্জস্য করার সাথে সাথে এই সমন্বয়গুলি স্বয়ংক্রিয়ভাবে সম্পূর্ণ হয়।

(Refer Slide Time: 16:19)



Function	
PwmOut	Create a PwmOut connected to the specified pin
write	Set the output duty cycle, specified as normalized float (0.0 to 1.0)
read	Return the current output duty cycle setting (0.0 to 1.0)
period	Set the PWM period, specified in seconds (float), milli-seconds (int) or micro-seconds (int), keeping the duty cycle the same.
period_ms	
period_us	
pulsewidth	Set the PWM pulsewidth, specified in seconds (float), milli-seconds (int) or micro-seconds (int), keeping the period the same.
pulsewidth_ms	
pulsewidth_us	
operator =	Shorthand for write()
operator float()	Shorthand for read()

আসুন PwmOut লাইব্রেরির অংশ হিসাবে উপলব্ধ ফাংশনগুলি দেখুন। এই PwmOut বর্গে এটি একটি অবজেক্ট ওরিয়েন্টেড ধারণা। এটি এমন একটি শ্রেণী যা একটি নির্দিষ্ট পিনের সাথে সংযুক্ত একটি

PwmOut ইন্টারফেস তৈরি করবে, কারণ আরডুইনো ইন্টারফেসে দেখা যাবে এমন অনেকগুলি পিন রয়েছে যা D1, D2, D3 ইত্যাদি বলে PWM / ডি 3। আমি কিছু উদাহরণ নেব।

রাইট অ্যান্ড রিড নামে কিছু ফাংশন রয়েছে। রাইটিং ফাংশনটি ব্যবহার করে আপনি তরঙ্গন্তপের DUTY CYCLE নির্দিষ্ট করতে পারেন। DUTY CYCLEটি 0 থেকে 1 পর্যন্ত ভগ্নাংশ হিসাবে সুনির্দিষ্ট করা হয়েছে যার অর্থ এটি 100 দ্বারা গুণিত হয় না, কেবলমাত্র মোট সময়কাল দ্বারা বিভাজিত সময়কালে এটি ভগ্নাংশ হিসাবে নির্দিষ্ট করা হয়। আপনি যদি এটি 0.5 হিসাবে লিখেন তবে এটি সমাপ্ত ও অক্ষ পরিয়ডের হবে।

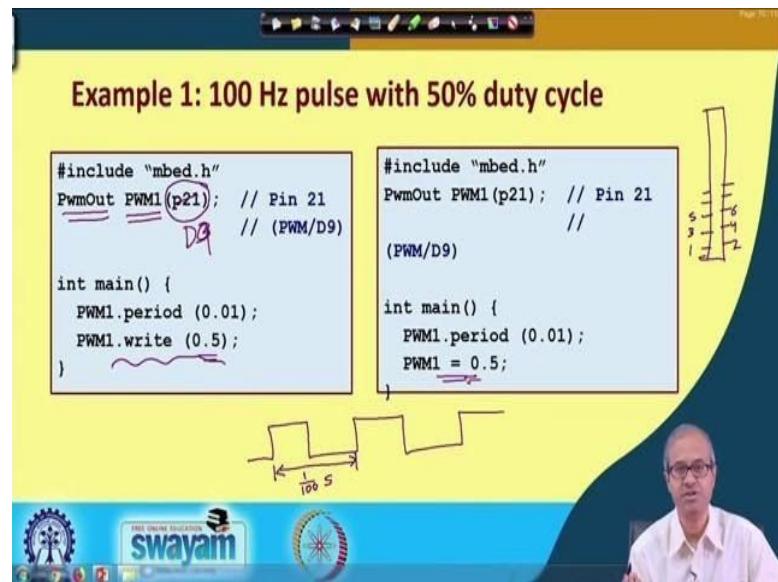
একইভাবে, আপনি DUTY CYCLE এর বর্তমান মানটি পড়তে পারেন। মনে করুন, কোনও প্রোগ্রামে আপনি যে ডিউটি চক্রটি নির্ধারণ করেছিলেন তা মনে নেই, আপনি এই ফাংশনটি পড়তে এবং এটি জানতে পারেন। এবং, আমরা সময়কাল নির্ধারণ করতে পারি; তিনটি ফাংশন উপলব্ধ: পিরিয়ড, পিরিয়ড_ms, পিরিয়ড_us প্রথমটি সময়কাল সেকেন্ডে নির্দিষ্ট করে, দ্বিতীয়টি মিলিসেকেন্ডে নির্দিষ্ট করে, তৃতীয়টি মাইক্রোসেকেন্ডে। লক্ষ্য করার একমাত্র পয়েন্ট হ'ল আপনি যখন এটি সেকেন্ডে নির্দিষ্ট করেন, আপনার প্যারামিটারটি একটি **ফ্লোটিং পয়েন্ট নাম্বার** (floating point number) হবে, আপনি 2.5 লিখতে পারবেন। তবে, আপনি যখন মিলিসেকেন্ড বা মাইক্রোসেকেন্ডে নির্দিষ্ট করবেন তখন প্যারামিটারগুলি পূর্ণসংখ্যা হবে। সুতরাং, কোনও ভগ্নাংশের অংশটিকে আপনার এটি মনে রাখা উচিত নয়।

ডিউটি সাইকেল(DUTY CYCLE) T লিখে আপনি নিজের সময়সীমা নির্ধারণ করতে পারেন।

ডিউটি সাইকেল(DUTY CYCLE)T নির্দিষ্ট করার পরিবর্তে নিজের সাথে নিয়ন্ত্রণ রাখার আরও একটি উপায় রয়েছে; আপনি PULSE WIDTH t_{on} নির্দিষ্ট করতে পারেন। এখানে আপনি সেকেন্ড, মিলিসেকেন্ড বা মাইক্রোসেকেন্ডে নির্দিষ্ট করতে পারেন। আবার সেকেন্ডে এটি ভাসমান পয়েন্ট হবে, মিলিসেকেন্ড এবং মাইক্রোসেকেন্ড এটি পূর্ণসংখ্যা হবে।

এখন, লেখার জন্য একটি শর্টকাট আছে। আপনি যদি সরাসরি বস্তুর নাম কোনও কিছুর সমান, যা লেখার সমতুল্য। আমরা কিছু উদাহরণ নেব।

(Refer Slide Time: 19:39)



মনে করুন, আমরা 50% duty cycle সহ 100 হার্জে pulse তৈরি করতে চাই। 100 হার্জ এর অর্থ কী? 100 হার্জ মানে $1/100$ সেকেন্ড $= 0.01$ সেকেন্ড সময়কাল হবে। আমি দুটি বিকল্প কোড দেখছি এই এমবেডেড। h header টি অন্তর্ভুক্ত করতে হবে কারণ আমি যে ফাংশনগুলি ব্যবহার করছি তার মধ্যে অনেকগুলি এখানে অন্তর্ভুক্ত রয়েছে। এই PwmOut হল PWM অবজেক্ট যা আমি আপনাকে বলেছিলাম। আমাদের এই শ্রেণীর উদাহরণ তৈরি করতে হবে, আপনি এমন একটি বস্তু তৈরি করেন যা আপনি এটিকে PWM 1 বলেছেন এবং P 21 পিন।

আপনি বিস্তারিত সংযোগকারী যেখানে আপনি দেখতে পাচ্ছেন যে উভয় পক্ষের মধ্যে একটি সংযোগকারী থাকবে সঙ্গে P 21 জানতে হবে। পিনগুলি **1, 2, 3, 4, 5, 6** এরকম নম্বর কনভেনশনযুক্ত। 21 নাম্বার পিনটি PWM port। যদি আরডুইনো সংযোগকারী(Connector) ব্যবহার করেন তবে এটিতে PWM / D 3 লেখা আছে দেখা যাবে। সুতরাং, P 21 এর পরিবর্তে আপনি সরাসরি **D3** ও লিখতে পারেন।

মূল ফাংশনে এটি আমি লিখছি একটি C কোড। আমি দুটি ফাংশন পিরিয়ডকে কল করছি এবং লিখছি, 0.01 এর অর্থ 100 হার্জ এবং আমি duty cycle 0.5 হিসাবে উল্লেখ করছি।

দ্বিতীয় কোডটি একই কোড যেখানে কল লেখার পরিবর্তে আমি শর্টকাট ব্যবহার করছি। আপনি যদি সরাসরি $PWM1 = 0.5$ লেখেন তবে এর অর্থ আসলে আমরা **ডিট্যু সাইকেল**(DUTY CYCLE)টি লিখছি 0.5, এটি কেবল একটি শর্টকাট। এভাবেই আপনি PWM তরঙ্গক্রম তৈরি করতে পারেন।

(Refer Slide Time: 22:14)

Example 2: 1 KHz pulse with 80% duty cycle

```
#include "mbed.h"
PwmOut PWM1(p21);      // Pin 21
                      // (PWM/D9)

int main() {
    PWM1.period_ms(1);
    PWM1 = 0.8;
}
```

A diagram of a square wave pulse. The period of the pulse is labeled as 1 msec.

FREE ONLINE EDUCATION
swayam
THINK BIG, DO BIG

অন্য একটি উদাহরণ নিন। এটি 80% DUTY CYCLE সহ 1 KHz pulse। এখানে তরঙ্গরপ্তি এরকম কিছু দেখাবে। 1 KHz মানে সময়কাল 1 মিলিসেকেন্ড এবং 80% DUTY হবে। আপনি যেভাবে PWM পিনটি নির্দিষ্ট করেন ঠিক সেভাবে আপনি সময়কালটি মিলি সেকেন্ডে নির্দিষ্ট করেন। এবং DUTY CYCLE 0.8। এটি এর মতো একটি তরঙ্গরপ্তি তৈরি করবে।

এটি আমরা পরে অংশে দেখতে পাব।

(Refer Slide Time: 23:12)

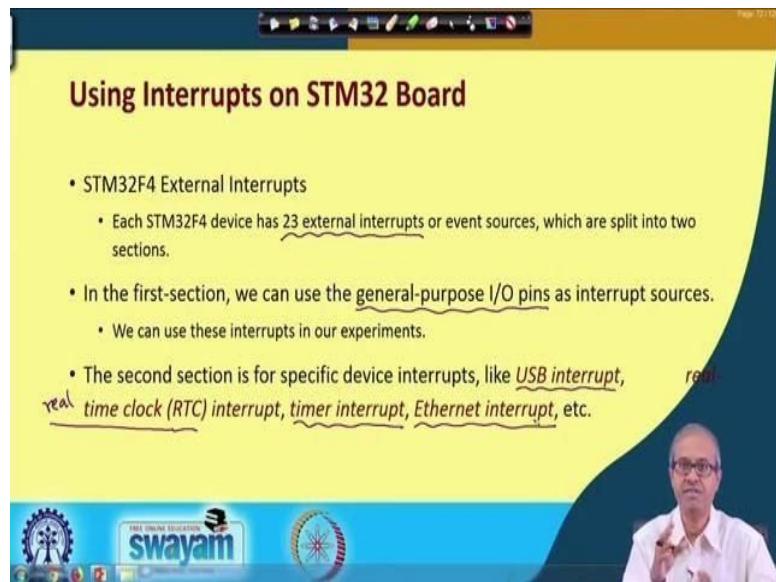
Interrupt on the STM32 Board

FREE ONLINE EDUCATION
swayam
THINK BIG, DO BIG

এখন, আসুন এসটিএম 32 বোর্ডে পাওয়া যায় এমন ধরনের ইন্টারাপ্ট(interrupt) কী তা দেখি। অনেক অ্যাপ্লিকেশনগুলিতে আপনি যেখানেই কিছু ডিভাইসকে হ্যান্ডেল করছেন সেখানে কিছু ডিভাইস রয়েছে

যা বাইরে থেকে কিছু বিস্তৃত সংকেত তৈরি করতে পারে। আমি উল্লেখ করেছি যে এআরএম প্রসেসরে অনেক বিস্তৃত উত্স রয়েছে; বাস্তবে, যে কোনও পোর্ট লাইন আপনি ইন্টারাপ্ট(interrupt) ইনপুট হিসাবে ব্যবহার করতে পারেন।

(Refer Slide Time: 23:39)



আসুন আমরা আরও কিছু বিশদ খতিয়ে দেখি।

বোর্ডের STM32F4 সাধারণ পরিবারের জন্য, 23 টি বাহ্যিক ইন্টারাপ্ট(interrupt) রয়েছে যা সমর্থিত। এই 23 বাহ্যিক ইন্টারাপ্ট(interrupt) দুটি বিভাগে বিভক্ত। প্রথম বিভাগে আমরা সাধারণ উদ্দেশ্যে আইও পিনগুলি পোর্ট A, পোর্ট B, পোর্ট C নিয়ে কথা বললাম যার বিষয়ে ইন্টারাপ্ট(interrupt) ইনপুট হিসাবে ব্যবহার করা যেতে পারে। প্রকৃত পরীক্ষায় আপনি এই পিনগুলি কেবলমাত্র interrupt উত্সগুলিতে সংযোগ করতে ব্যবহার করবেন।

তবে আরও একটি বিভাগ রয়েছে যেখানে কিছু নির্দিষ্ট ডিভাইস ইন্টারাপ্ট(INTERRUPT) সৃষ্টি করে। USB ইন্টারফেস যেমন একটি ইন্টারাপ্ট(INTERRUPT) সৃষ্টি করতে পারে ঠিক তেমনি রিয়েল টাইম ক্লক যা দিনের সময় বজায় রেখে একটি ইন্টারাপ্ট(interrupt) তৈরি করতে পারে। আপনি যে টাইমারটি ব্যবহার করছেন তা একটি ইন্টারাপ্ট(interrupt) তৈরি করতে পারে, যদি ইথারনেটের মতো নেটওয়ার্ক ইন্টারফেস থাকে যেখানে ইন্টারফেস(interface) জেনারেট করতে পারে, তবে ডিভাইসের নির্দিষ্ট ইন্টারাপ্ট(interrupt) দ্বিতীয় বিভাগে আসে। তবে প্রথম বিভাগটি আরও সাধারণ। আমাদের পরীক্ষায় আমরা প্রথম বিভাগটি ব্যবহার করব।

(Refer Slide Time: 25:17)

General-Purpose I/O (GPIO) Pin Interrupt Lines

- The 16 interrupt lines are referred to as Line0 to Line15.
 - Three 16-bit GPIO ports: PA, PB and PC.
 - Line0 is connected to PA0, PB0, and PC0.
 - Line1 is connected to PA1, PB1, and PC1.
 - Line2 is connected to PA2, PB2, and PC2, and so on.
- We cannot use PAx, PBx, PCx as interrupt sources together for the same value of x.

FREE ONLINE EDUCATION
swayam

প্রথম বিভাগে এগুলিকে সাধারণ উদ্দেশ্যে I/O বা GPIO বিনিয়িত লাইন বলা হয়। এই বিভাগের অধীনে বিস্তৃতভাবে আমরা বলি যে এখানে 16 টি interrupt লাইন রয়েছে। এই বিনিয়িত রেখাগুলি লাইন 15 পর্যন্ত উল্লেখ করা হয়। আপনার মনে রাখতে হবে যে এই এআরএম বোর্ডে আমরা ব্যবহার করছি, সেখানে তিনটি সাধারণ উদ্দেশ্যে I/O PORT রয়েছে, তাদের প্রত্যেককেই 16-বিট, এগুলিকে পোর্ট A, পোর্ট B এবং পোর্ট C বলা হয়। এগুলি পোর্ট A0 থেকে পোর্ট A 15 পর্যন্ত সংখ্যার, B0 থেকে 15, পোর্ট C 0 থেকে 15।

লক্ষ্য করার বিষয়টি হ'ল এই 16 টি interrupt লাইনগুলি স্বাধীনভাবে সমস্ত পোর্ট লাইনের সাথে সংযুক্ত নয়, লাইন 0 PA0, PB0 এবং PC0 এর সাথে সংযুক্ত রয়েছে। সুতরাং, যখনই আপনি লাইন 0 ব্যবহার করার চেষ্টা করছেন, আপনি নিজের interrupt ইনপুটের PA0 বা PB0 বা PC0 এর সাথে সংযুক্ত করতে পারেন; এর অর্থ হ'ল আপনি PA0 এবং PB0 কে দুটি পৃথক বিনিয়িত ইনপুট হিসাবে ব্যবহার করতে পারবেন না কারণ এগুলি প্রকৃতপক্ষে একই লাইনকে interrupt দেওয়ার ক্ষেত্রে বোঝায়।

একইভাবে লাইন 1 PA 1, PB 1, PC 1 এর সাথে সংযুক্ত রয়েছে। লাইন 2 PA 2, PB 2, PC 2 ইত্যাদির সাথে সংযুক্ত থাকে। যেমনটি আমি বলেছি আপনি এক্সট্রেট ইনপুট হিসাবে x এর একই মানের জন্য PAx, PBx, PCx ব্যবহার করতে পারবেন না কারণ PAx, PBx, PCx সমস্ত একই interrupt ইনপুট লাইন এক্সকে উল্লেখ করে। এটি আপনার মনে রাখতে হবে। আপনি একবারে ব্যবহার করতে পারবেন এমন 16 টি interrupt উভয় রয়েছে তবে সংযোগের জন্য আপনি ব্যবহার করতে পারেন সমস্ত 48 পোর্ট ইনপুট।

(Refer Slide Time: 27:28)

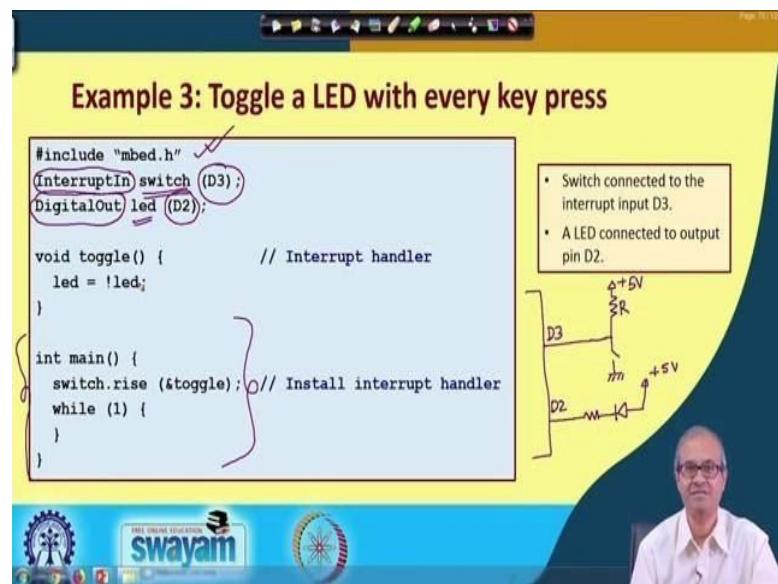
Function	Description
InterruptIn ✓✓	Create an InterruptIn connected to the <u>specified pin</u>
int read() ✓✓	Read the input, represented as 0 or 1 [int]
operator int()	Shorthand for read()
void rise (* func)	Attach a function to call when a rising edge occurs on the input
void enable_irq()	Enable IRQ ✓✓
void disable_irq()	Disable IRQ ✓✓

বাধা দেওয়ার জন্য লাইব্রেরির ফাংশন যা আমাদের কাছে পাওয়া যায় সেগুলি নীচে রয়েছে। ইন্টারাপ্ট(interrupt) ক্লাস, আপনি এই ধরণের একটি অবজেক্ট তৈরি করেন। আপনি যে পিনটি ঠিক PWM এর মতো সংযোগের চেষ্টা করছেন তা এখানেও নির্দিষ্ট করে দিন। এবং এটি উপলব্ধ ফাংশনগুলি। পর্তন মানে আপনি ইন্টারাপ্ট(interrupt) ইনপুটটির অবস্থানটি বর্তমানে 0 এবং 1 হয় কিনা তা পড়তে পারবেন। আপনি যদি রিড কল করেন, এটি পূর্ণসংখ্যার 0 এবং 1 এর মধ্যে আসে।

আপনি যদি তৈরি করছেন এমন বস্তুর নাম ব্যবহার করেন তবে এটি রিড শর্টকাট। রাইজ(RISE) নামে একটি ফাংশন রয়েছে, যেখানে আপনি যে প্যারামিটার দেন তা কোনও ফাংশনের পয়েন্টার; এর অর্থ যদি ইন্টারাপ্ট(interrupt) ইনপুটটিতে স্বয়ংক্রিয়ভাবে কোনও উত্থান ঘটে থাকে তবে সেই ফাংশনটি কল করা হবে। এটি আপনার বিন্দিত পরিষেবা সাবরুটাইন বা বিন্দিত হ্যান্ডলারের মতো মূলত কাজ করে।

আপনি যখন কোনও সি প্রোগ্রাম লিখছেন আপনি একটি নির্দিষ্ট ইন্টারাপ্ট(interrupt) ইনপুট জন্য নির্দিষ্ট করতে পারেন এটি আপনার ইন্টারাপ্ট(interrupt) হ্যান্ডলার হবে। এবং, আপনি সক্ষম করুন_irq, অক্ষম_irq ফাংশন কল করে ইন্টারাপ্ট(interrupt) সক্ষম এবং অক্ষম করতে পারেন। এটি আপনি চাইলে ইন্টারাপ্ট(interrupt) সক্ষম এবং অক্ষম করবে।

(Refer Slide Time: 29:04)



আসুন একটি সহজ উদাহরণ দেখুন। বাম দিকে আমি একটি সাধারণ উদাহরণের কোডটি ইন্টারপ্রেটেগুলি ব্যবহার করে দেখছি। মনে করুন এটি আপনার মাইক্রোকন্ট্রোলার বোর্ড। আসুন D3 লাইনে বলি আমি একটি সুইচ সংযুক্ত করেছি। সাধারণত আমরা একটি সুইচ এর সাথে সংযোগ করি; আমরা একটি প্রতিরোধ সংযোগ করি, তারপরে আপনি এই জাতীয় একটি সুইচ সংযুক্ত করুন, অন্যদিকে আমরা 5 ভোল্ট সংযোগ করি এবং এখানে আমাদের প্রতিরোধ আছে। যদি এই সুইচটি খেলা থাকে, উচ্চ ভোল্টেজ বা লজিক 1 D 3 এ আসবে, এটি টিপলে এটি স্লভাগে ছোট হয়ে যাবে, এবং 0 আসবে।

এবং, D 2 এর অন্যদিকে আমরা একটি এলইডি সংযুক্ত করছি। আমরা একটি প্রতিরোধ সংযোগ করছি, তারপরে আপনি একটি LED সংযুক্ত করছেন, তারপরে আমরা এটিকে 5 ভোল্টের সাথে সংযুক্ত করছি। সুতরাং, যখনই D 2 0 হয়, তখন একটি তড়িৎ প্রবাহিত হবে এবং এলইডি আলোকিত হবে, D 2 যদি 1 হয় তবে কোনও প্রবাহিত প্রবাহ থাকবে না কারণ উভয় পক্ষই একই ভোল্টেজে থাকবে, এবং LED আলোকিত হবে না। এইভাবে আমি সংযোগগুলি তৈরি করেছি। আমি যা চাই তা হ'ল প্রতিবার এই সুইচটি টিপলে LED এর স্বল্পন্ধ অবস্থা পরিবর্তন করা উচিত; তার মানে, অফ অন অন অফ এরকম।

প্রোগ্রামে, যথারীতি আমরা এই এমবেড। এইচ অন্তর্ভুক্ত করেছি, তারপরে আমরা ইন্টারাপ্ট ইন (interrupt IN) টাইপের একটি বস্তু তৈরি করেছি, আমরা নাম সুইচ দিয়েছি, আমরা এটি D 3 এর সাথে সংযুক্ত করেছি। D 3 হ'ল পিন নম্বর যার সাথে আপনি ইন্টারাপ্ট(interrupt) ইনপুট সংযুক্ত করেছেন। এবং এই হিসাবে একটি ডিজিটাল আউটপুট পিন হয়। এটির জন্য আমরা আরও একটি অবজেক্ট ব্যবহার করি, এটি ডিজিটাল আউট বলে। সমস্ত ডিজিটাল আউটপুট পোর্ট লাইনগুলির জন্য,

আমরা এই ধরণের ডিজিটাল আউট(Digital OUT) টাইপটি ব্যবহার করি এবং এই সংযোগটির নাম এবং এখানে আমি পোর্ট নম্বর D 2 ব্যবহার করেছি।

এটি আমার মূল কাজ। এখানে আমরা প্রথমে এই উৎপাদনের জন্য একটি কল করেছি, সুইচ হ'ল আপনি তৈরি করেছেন। Switch.rise এর অর্থ আমি এই ফাংশনটি অবজেক্ট সুইচের সাথে যুক্ত করছি এবং আমি টগল(Toggle) এর address দিয়েছি; টগল(Toggle) একটি ফাংশনের নাম। এর অর্থ যখনই ইন্টারাপ্ট(interrupt) ইনপুটের স্বয়ংক্রিয়ভাবে ফাংশন টগল কল হয়ে উঠবে এবং এর পরে একটি ডামি লুপ থাকবে যেখানে আপনার মূল প্রোগ্রামটি অপেক্ষা করছে

প্রতিবার একটি সুইচ প্রেসের অর্থ একটি ইন্টারাপ্ট(interrupt) আসছে কী হবে?

টগল কেবল led != করবে led 0 থেকে, এটি 1 হয়ে যাবে; এটি যদি 1 হয় তবে এটি 0 হয়ে যাবে

এটির সাথে আমরা এই বক্তৃতাটির শেষে এসেছি। এখানে আমরা আপনাকে PWM এবং এসটিএম 32 বোর্ডে ইন্টারাপ্ট(interrupt) ইন্টারফেস এবং সেগুলি কীভাবে ব্যবহার করব সে সম্পর্কে বলার চেষ্টা করেছি।

ধন্যবাদ।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 12
Digital to Analog Conversion

আপনি যখন এমবেডেড সিস্টেম ডিজাইন(Embedded system design) এবং প্রয়োগ করার চেষ্টা করছেন, আপনাকে বাইরের বিশ্বের সাথে ইন্টারফেস(interface) করতে হবে; সেখানে বিভিন্ন ধরনের সেন্সর(sensors) থাকবে, বিভিন্ন অ্যাকচুয়েটরস(actuators) থাকবে। সুতরাং যে ভোল্টেজ গুলো বাইরে থেকে আসছে, সেগুলো খুব কমই ডিজিটাল হয়, বেশিরভাগই এনালগ। একইভাবে যখন আপনি অ্যাকচুয়েটর গুলোকে সক্রিয় করছেন, আপনাকে প্রায়ই সেগুলোর জন্য এনালগ সিগন্যাল দিয়ে দিতে হবে। এইজন্য আপনার মাইক্রোকন্ট্রোলার প্রসেসিং ইউনিট(microcontroller processing unit)র সাথে ইন্টারফেস করার জন্য দরকার, এমন যন্ত্র, যাকে বলে ডিজিটাল টু এনালগ কনভার্টার(digital to analog converter) এবং এনালগ টু ডিজিটাল কনভার্টার(analog to digital converter)।

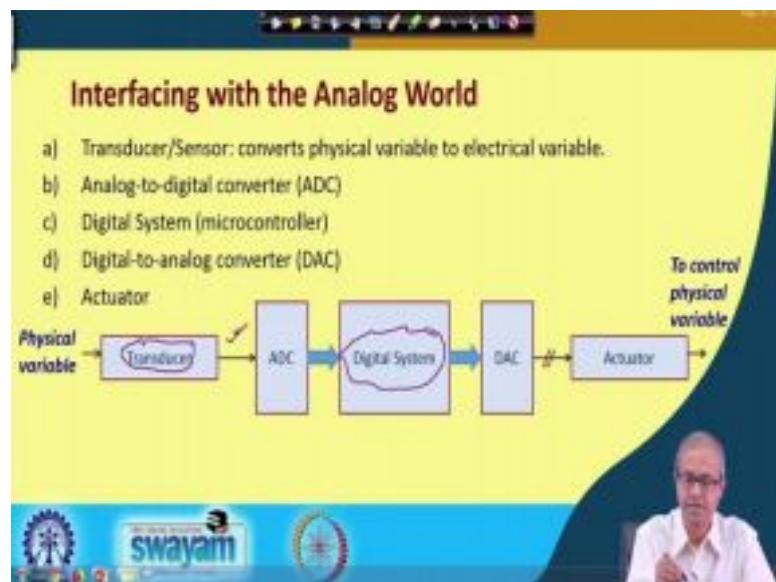
এই বক্তৃতার বিষয় হলো ডিজিটাল থেকে এনালগ রূপান্তর(conversion)। এখানে আমরা কিভাবে ডিজিটাল থেকে এনালগ করা যায় তার বিভিন্ন পদ্ধতি নিয়ে পড়াশোনা করব।

(Refer Slide Time: 01:15)



এই বক্তৃতায় আমরা ডিজিটাল থেকে অ্যানালগ রূপান্তরকরণের কয়েকটি প্রাথমিক ব্যাপারে কথা বলব এবং এই জাতীয় ডি/এ রূপান্তরকারী (D/A converter)গুলিকে বাস্তবায়নের জন্য আমরা দুটি ভিন্ন ভিন্ন ডিজাইনের স্টাইল নিয়ে আলোচনা করব।

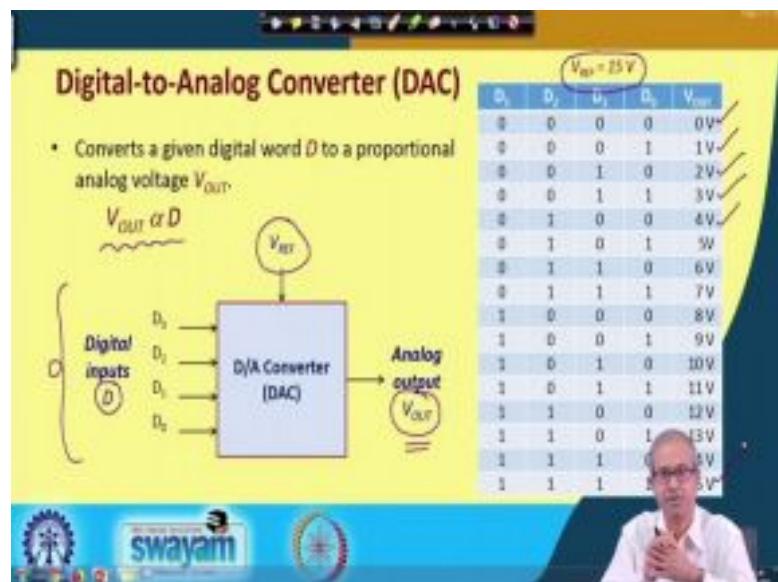
(Refer Slide Time: 01:37)



এনালগ শব্দটির সাথে পরিচিত করার জন্য, আমি একটি স্কিম্যাটিক ডায়াগ্রাম(schematic diagram) দেখাচ্ছি। মাইক্রোকন্ট্রোলার টি এখানে আছে, এটাই প্রকৃত প্রসেসিং ইউনিট(processing unit) বা ডিজিটাল সিস্টেম(digital system)। ফিজিক্যাল ভ্যারিয়েবেল(physical variable) যেটা আপনি বাইরে থেকে বুঝতে পারছেন, সেখানে আপনার কোন একটি ধরনের সেন্সর(sensor) ব্যবহার করা উচিত, এটিকে কিছু ক্ষেত্রে ট্রান্সডিউসার(transducer) বলে যা ভোল্টেজে পরিবর্তন করে। এবং একটি A/D কনভার্টার(converter) এই ভোল্টেজকে ডিজিটাল ইনপুটে রূপান্তর করে আর এই মাইক্রোকন্ট্রোলার ডিজিটাল ইনপুট কে সরাসরি পড়তে পারে।

এবং একইভাবে যখন এটি একচুয়েটর(actuator)কে সক্রিয় করতে চায় তখন এটি একটি ডিজিটাল আউটপুট দেয় যেটা D/A রূপান্তরের মাধ্যমে ভোল্টেজে পরিবর্তিত হয়; এটা একটা এনালগ ভোল্টেজ যেটা একটি অ্যাকচুয়েটর(actuator)-এ পাঠানো হয়। এইগুলি একটি সাধারণ এমবেডেড সিস্টেম(embedded sysytem)র প্রাথমিক উপকরণ।

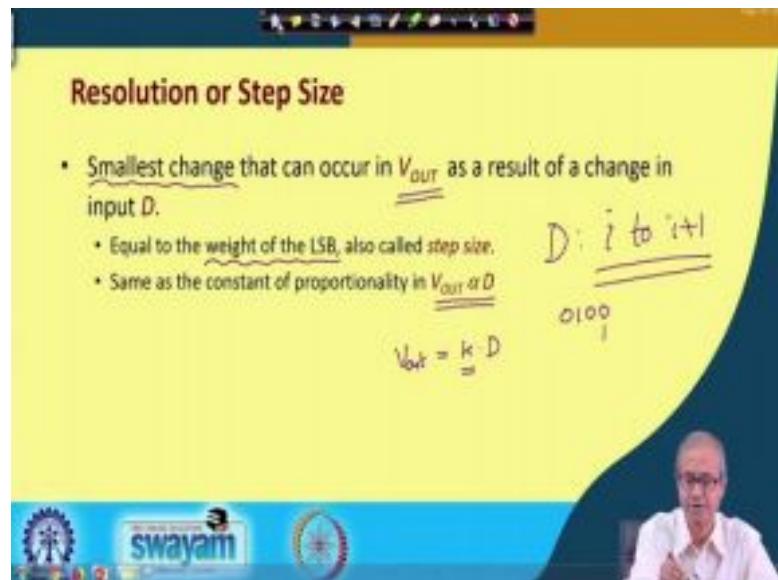
(Refer Slide Time: 02:57)



একটা ডিজিটাল টু এনালগ কনভার্টার বা D/A কনভার্টার(converter) কি? এই স্লাইড ডায়াগ্রাম(block diagram) টা দেখুন। ইনপুটে আমরা একটা ডিজিটাল ওয়ার্ড(word) প্রয়োগ করি। এই উদাহরণে, আমি দেখাচ্ছি একটা 4 বিট ওয়ার্ড(word) D0 D1 D2 D3. আমরা এটাকে D বলি; এটা হল ওই সংখ্যাটির ডেসিমাল(decimal) সমতুল্য। আর আউটপুটে, আমরা একটি এনালগ ভোল্টেজ উৎপন্ন করি। এনালগ ভোল্টেজ মানে একটি অবিচ্ছিন্ন ভোল্টেজ যেটা আমার প্রয়োগ করা ইনপুট এর ডিজিটাল মানের সাথে সমানুপাতিক হবে। সুতরাং, এই V_{OUT} হবে D এর সাথে সমানুপাতিক, এবং এই D/A কনভার্টার(D/A converter)-এ প্রায়ই, আমরা কিছু রেফারেন্স(reference) ভোল্টেজ প্রয়োগ করি, যেগুলো নির্ধারণ করে দেয় V_{OUT} এর সর্বোচ্চ মান কত হবে।

এই টেবিলে আমি একটি খুব ছোট উদাহরণ দিয়েছি। আমি ধরে নিলাম রেফারেন্স(reference) ভোল্টেজ হল 15 volts; এটি একটি 4 বিট D/A কনভার্টার(converter)। সুতরাং, এখানে হবে 0 0 0 0 থেকে 1 1 1 1, অথবা 16 টি কম্বিনেশন(combination), এবং আউটপুট ভোল্টেজ টি এই রকম হতে পারে। 0 0 0 0 মানে হল আউটপুট 0 volt, 1 ভোল্ট, 2 ভোল্ট, 3 ভোল্ট, এভাবে 15 ভোল্ট পর্যন্ত। আউটপুট ভোল্টেজ টি ইনপুট ডিজিটাল ওয়ার্ড(input digital word)-এর ডেসিমাল সমতুল্যের সমানুপাতিক হবে। এই ক্ষেত্রে D- এর প্রতিটি বৃক্ষির জন্য আউটপুট 1 ভোল্ট করে বাড়বে।

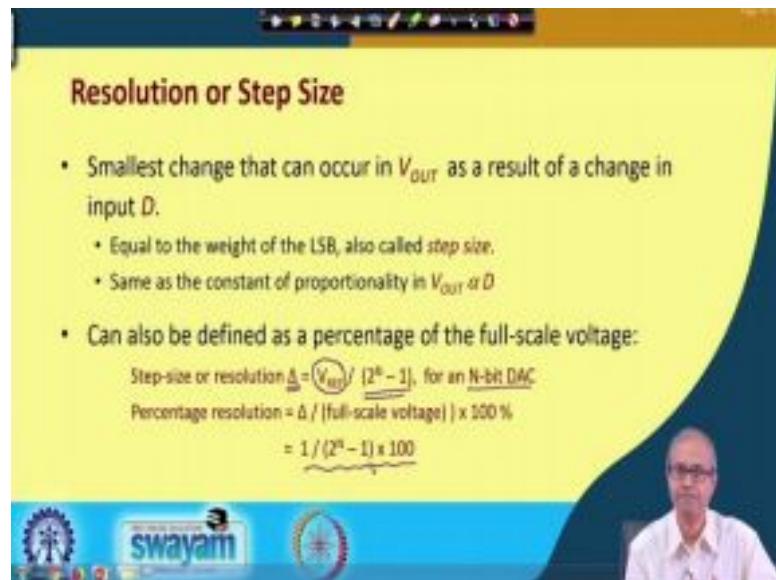
(Refer Slide Time: 04:53)



D-র প্রতি একক মান বৃদ্ধির জন্য আউটপুট-র এই বৃদ্ধি কখনও কখনও রেজোলিউশন(resolution) বা D/A কনভার্টার(converter)-এর স্টেপ সাইজ(step size) বলা হয়। রেজোলিউশন এর সংজ্ঞা কিভাবে বলবেন ? এটা হল ক্ষুদ্রতম পরিবর্তন যা D-র পরিবর্তনের উপর নির্ভর করে আউটপুট ভোল্টেজ এ হয়। D হলো একটি ডিজিটাল ইনপুট(digital input)। তাই, আমরা D-তে 0,1,2,3,4 প্রয়োগ করছি। যখনই D- এর মান যে কোন ভ্যালু(value) i থেকে পরবর্তী ভ্যালু $i+1$ এ পরিবর্তিত হয়, আউটপুট এর এনালগ ভোল্টেজের পরিবর্তন V_{OUT} হলো স্টেপ সাইজ(step size) বা রেজোলিউশন। কখনো কখনো আমরা এটিকে লিস্ট সিগনিফিকেন্ট বিট(least significant bit) এর ওয়েট(weight) হিসেবে বলতে পারি কারণ, যখন লিস্ট সিগনিফিকেন্ট বিট(least significant bit) পরিবর্তন হয়, তার মানে হল 1 এর বৃদ্ধি।

সুতরাং, V_{OUT} হল D-এর সমানুপাতিক। স্টেপ সাইজ(step size) হল সমানুপাতিকতার ধ্রুবক। সুতরাং, যে মানই আপনি D- তে প্রয়োগ করছেন, সেই স্টেপ সাইজ(step size) D-র সাথে গুন করে আপনি প্রকৃত ভোল্টেজ পাবেন।

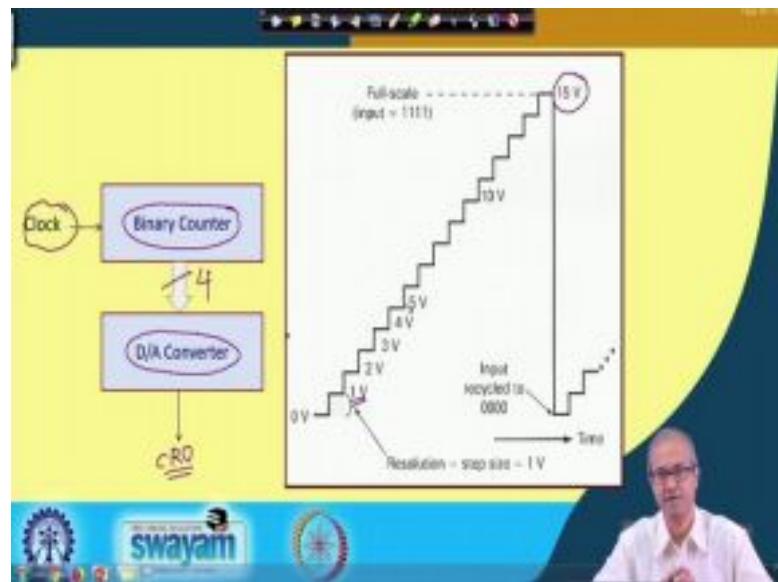
(Refer Slide Time: 06:37)



রেজিলিউশন(resolution) কে ফুল স্কেল ভোল্টেজ(full scale voltage) এর শতাংশ হিসেবে সংজ্ঞায়িত করা যায়। স্টেপ সাইজ(step size) বা রেজিলিউশন(resolution) যদি আপনি একে Δ বলেন, এটাকে ফুল স্কেল ভোল্টেজ V_{ref} কে D/A কনভার্টার এর সমস্ত স্টেপ দিয়ে বিভক্ত করে এর সংজ্ঞা বলা যায়। যেমন একটি ফোর বিট কনভার্টার(converter)-এ আপনি 0 0 0 0 থেকে 1 1 1 1 মানে 15 অবধি যেতে পারেন। একটি N বিট কনভার্টার-এ আপনি $2^N - 1$ অবধি গুনতে পারবেন। সুতরাং, মোট ভোল্টেজ কে স্টেপ সংখ্যা দিয়ে ভাগ করলে আমরা প্রতিটি স্টেপ(step)-এর উচ্চতা বা রেজিলিউশন পাই। এভাবে আপনি একটি N -bit D/A কনভার্টার এর স্টেপ সাইজ(step size) বা রেজিলিউশন সংজ্ঞায়িত করতে পারেন।

আমরা যদি একে শতাংশ-এ প্রকাশ করতে চাই, আমরা কি করব? আমরা এই Δ কে ফুল স্কেল ভোল্টেজ(full scale voltage) V_{ref} দিয়ে বিভাজিত করব। এটা হবে 1 ভাগ $(2^N - 1)$ গুণিতক 100. এটাই হলো রেজুলেশনের(resolution) শতাংশে প্রকাশ। যদি আপনাকে বলা হয় একটি N বিট D/A কনভার্টার(converter) এর রেজিলিউশন(resolution) শতাংশ- এ নির্ণয় করতে, আপনি এই ফর্মুলাটি(formula) ব্যবহার করবেন।

(Refer Slide Time: 08:03)



ধৰন আমাদেৱ এইৱকম একটি D/A কনভাৰ্টাৰ(conveter) আছে, এবং D/A কনভাৰ্টাৰ এৱ ইনপুটে আমাদেৱ একটি বাইনারি কাউন্টাৰ(binary counter) আছে, যেটা উপৱেৱ দিকে গণনা কৱে এবং আমৱা একটি ক্লক(clock) একটানা প্ৰয়োগ কৱি। কাউন্টাৰ(counter) টি 0,1, 2, 3, 4 এভাৱে উপৱেৱ দিকে গণনা কৱে চলে, আমৱা বলতে চাই এটি একটি 4 বিট D/A কনভাৰ্টাৰ। সুতৰাং, এখানে 4 টি লাইন আছে, আৱ এটা একটা 4 বিট কনভাৰ্টাৰ।

যেভাৱে ক্লক(clock) টি গণনা কৱে চলেছে তাতে কাউন্টাৰ(counter) টা শূন্য থেকে 15 অৰ্দি একটানা যাবে এবং তাৱপৰ আবাৱ 0 তে ফিৱে যাবে। আমি যদি 4 বিট D/A কনভাৰ্টাৰ(conveter) এ তৱঙ্গ আকাৱ পৰ্যবেক্ষণ কৱি, তাহলে আমি এইৱকম তৱঙ্গ আকাৱ দেখব। আউটপুটটি শূন্য থেকে 15 অৰধি ধাপে ধাপে যাবে এবং এটি তাৱপৰ 0 তে ফিৱে যাবে, এবং আবাৱ 1 2 3 4 এইভাৱে যাবে।

V_{ref} এৱ উপৱ নিৰ্ভৱ কৱে স্টেপ-এৱ উচ্চতাৱ সামঞ্জস্য কৱা হয়। যদি V_{ref}=15V হয়; তাৱ মানে, ফুল স্কেল ভোল্টেজ(full scale voltage) হল 15 V তখন আপনাৱ স্টেপ-এৱ উচ্চতা হল 1 ভোল্ট।
(Refer Slide Time: 09:45)

Types of D/A Converter

- We shall discuss two different designs of digital-to-analog converters.
 - a) Weighted resistor type DAC
 - b) Resistive ladder type DAC
- The first type is easier to analyze, while the second type is more practical from the point of view of implementation.

এখন বিভিন্ন ধরনের D/A কনভার্টার(conveter) এর ব্যাপারে বলছি. আমি প্রথমে বলেছিলাম যে আমরা সবচেয়ে সাধারণ 2 ধরনের D/A converter এর ব্যাপারে কথা বলব। একটাকে বলে Weighted resistor ধরনের, আর একটাকে বলে resistive ladder ধরনের। এখন বিষয়টা হলো প্রথম ধরনেরটা তৈরি করা ও বোৰা সহজ। কিন্তু এটার ডিজাইন(design)-এ কিছু সমস্যার ব্যাপার আছে, আমরা দেখব, দ্বিতীয় ধরনেরটা তৈরি করা সহজ, কিন্তু এর সার্কিট(circuit) একটু বেশি জটিল।

(Refer Slide Time: 10:25)

Weighted Resistor Type DAC

- For an n -bit DAC, it consists of n different resistance values of magnitudes $R, 2R, 4R, \dots, 2^{n-1}R$ respectively.
- The resistances help in generating currents inversely proportional to their magnitudes.
- The total current is added up by an operational amplifier, and is converted to the voltage output V_{out} .

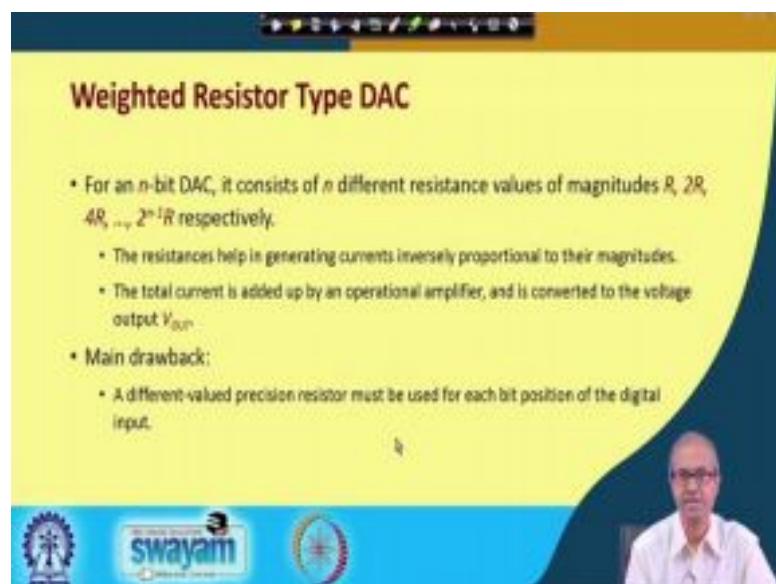
আমরা প্রথমে ওয়েটেড রেজিস্টার(weighted resistor) ধরনের D/A converter দিয়ে শুরু করি। আসুন আমরা একটি n -বিট D/A কনভার্টার এর ডিজাইন(design) বিবেচনা করি; সাধারণভাবে

এখানে n সংখ্যক বিট আছে যেগুলো ইনপুটে আসছে। প্রথম কাজটি হলো যে এই ধরনের D/A কনভার্টার গঠন করা, আমাদের n টি বিভিন্ন রোধের মান দরকার, এবং এটাই এই ধরনের D/A কনভার্টার উৎপাদন করার অন্যতম মুখ্য সমস্যা। আপনি দেখছেন, আপনি খুবই সঠিকভাবে 1, 2, বা তিনটি রোধের মান উৎপাদন করলেন, কিন্তু আমি যদি বলি আপনার 16 টি ভিন্ন রোধের মান দরকার, তখন আপনাকে তাদের প্রত্যেকটিকে খুবই সঠিকভাবে উৎপাদন করতে হবে, যেটা অনেক বেশি কঠিন হয়ে পড়বে। এই পদ্ধতিতে n টি ভিন্ন রোধের মান ব্যবহার করার দরকার পড়ে আর এদের মান হবে $R, 2R, 4R$ এর কিছু base মান; তার মানে দুইয়ের গুণিতক; এটা $2^{n-1} R$ অবধি বেড়ে যাবে।

বিভিন্ন রোধের মানগুলি প্রকৃতপক্ষে কিছু তত্ত্বিক উৎপাদনের জন্য দায়ী। আমরা ধরে নিলাম এখানে আপনার কাছে একটি রোধ R আছে এবং আমরা একটি ভোল্টেজ প্রয়োগ করলাম। ধরুন আমি এই পয়েন্টটা গ্রাউন্ড(ground) করলাম, সুতরাং এর মধ্যে দিয়ে যে তত্ত্বিক প্রবাহিত হবে তার মান হবে V/R . যদি আমি n টি ভিন্ন রোধের মান ব্যবহার করি, সে ক্ষেত্রে বিদ্যুতের মান রোধের মান দ্বারা নির্ধারিত করা যাবে, R এর জন্য এর কিছু মান থাকবে, $2R$ এর জন্য বিদ্যুতের পরিমাণ অর্ধেক হবে, $4R$ এর জন্য তারও অর্ধেক হবে, আর এভাবেই চলবে। বিদ্যুৎ এর মান গুলো 2 এর গুণিতক-এ পরিবর্তিত হবে।

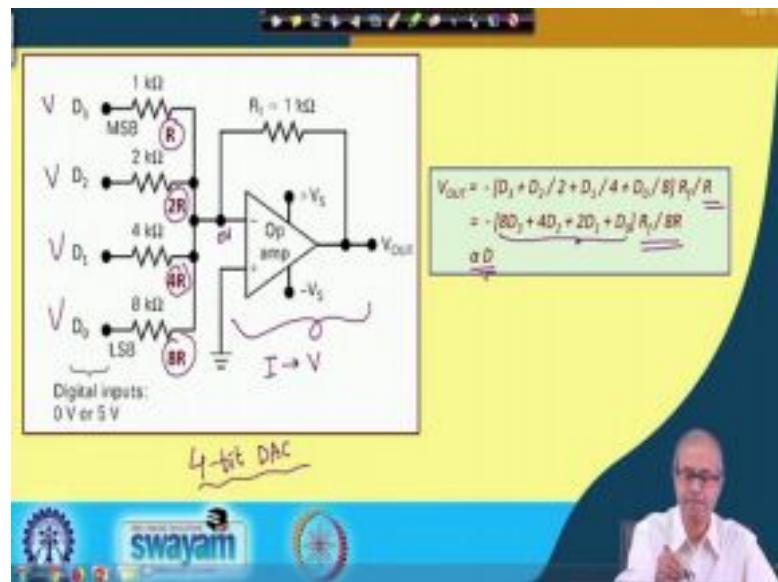
আমার কাছে যদি যে বিদ্যুৎ অবশ্যে বেরোচ্ছে সেটা গণনার কোন পদ্ধতি থাকে, তবে সেটা আপনাকে চূড়ান্ত DAC আউটপুট বলে দেবে, যদি আপনি একে ভোল্টেজে পরিবর্তন করতে পারেন।

(Refer Slide Time: 12:51)



প্রধান সমস্যাটা আমি আগেই বলেছি, আপনাকে এতগুলো রোধ ব্যবহার করতে হবে।

(Refer Slide Time: 12:57)

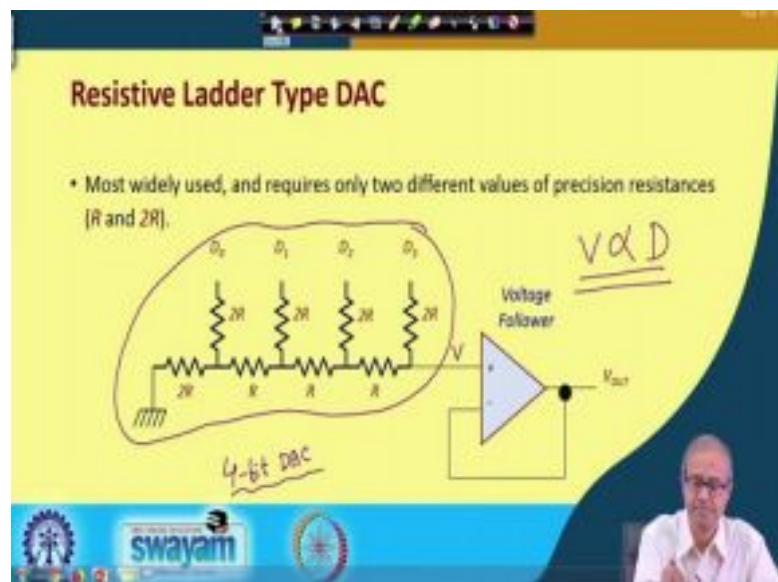


এটা এরকম দেখতে লাগে। আমি একটা ওয়েটেড রেজিস্টার(weighted resistor) ধরনের ফোর বিট D/A কনভার্টার(converter) দেখাচ্ছি। আপনি দেখছেন, আমরা এই মানগুলি ব্যবহার করছি R , $2R$, $4R$, $8R$ - উদাহরণ হিসেবে আমি এগুলো দেখাবো এক কিলো ওহম(ohm), দু কিলো ওহম, 4 কিলো ওহম এবং 8 কিলো ওহম। ডিজিটাল ইনপুট গুলো এভাবে প্রয়োগ করা হয়: $D_0 D_1 D_2 D_3$, সবচেয়ে কম গুরুত্বপূর্ণ বিটটি নিষ্পত্ত রোধের সাথে যুক্ত করা হয়। আউটপুট এর দিকে আমরা অপারেশনাল এমপ্লিফায়ার(opereational amplifier) ব্যবহার করছি। কিভাবে অপারেশনাল এমপ্লিফায়ার কাজ করে তার বিশদ বিবরণে আমি যাচ্ছি না, কিন্তু এই এমপ্লিফায়ার মূলত বিদ্যুৎকে তার সমানুপাতিক ভোল্টেজে পরিবর্তন করে।

সুতরাং যে বিদ্যুৎ তাদের মধ্য দিয়ে প্রবাহিত হবে, সেগুলো হলো V / R , $V / 2R$, $V / 4R$, এবং $V / 8R$. V_{OUT} এর জন্য মানটি দেখানো হল। পরিষ্কার বোৰা যাচ্ছে V_{OUT} হল D এর সমানুপাতিক।

এইভাবে ওয়েটেড রেজিস্টার(weighted resistor) D/A কনভার্টার কাজ করে। এটা বোৰা প্রকৃতপক্ষে খুবই সহজ। প্রতিটি রোধ একটি করে বিদ্যুৎ উৎপাদন করে যেটা ওই নির্দিষ্ট বিট টির মানের সমানুপাতিক এবং এই সমস্ত বিদ্যুৎকে অপারেশনাল এমপ্লিফায়ার(operational amplifier) যোগ করে আর ভোল্টেজ V_{OUT} এ পরিবর্তিত করা হয়।

(Refer Slide Time: 15:45)



এখন আমরা resistive ladder ধরনের কথায় আসছি। এখানেও আমি একটা ক্ষেত্রে D/A কনভার্টার(converter) এর ডায়াগ্রাম(diagram) দেখাবো। এখানে আপনি যেটা প্রথমে লক্ষ্য করবেন সেটা হল প্রয়োজনীয় রোধের সংখ্যা এখানে আরো বেশি। আগের পদ্ধতিতে, আমাদের চারটি রোধের দরকার ছিল, কিন্তু এখানে আপনার আটটি রোধের দরকার। ভালো জিনিসটা হল এই যে রোধের মানগুলি দুই ধরনের, R এবং $2R$, আপনাকে রোধের অনেক ধরনের বিভিন্ন মান ব্যবহার করার দরকার নেই।

এটা এরকম ব্যাপার নয় যে এই সার্কিট যে বিদ্যুৎ উৎপন্ন করে সেটাকে ভোল্টেজে ক্রপান্তরিত করতে হবে; বরং এটা সরাসরি ডিজিটাল ইনপুটের সমানুপাতিক হারে ভোল্টেজ উৎপন্ন করে। আউটপুটে আবার, আমরা অপারেশনাল এমপ্লিফায়ার(operational amplifier) সার্কিট ব্যবহার করছি, কিন্তু আপনি দেখতে পাবেন, এই op-amp কে আমরা অন্যভাবে যুক্ত করছি। একে বলে voltage follower সার্কিট। এটা যে কোন কিছুকে যে কোন কিছুতে ক্রপান্তরিত করে না, ইনপুট ছিল ভোল্টেজ এটা ভোল্টেজই থাকে, কিন্তু এটা প্রকৃতপক্ষে বাফার(buffer) হিসেবে কাজ করে, যাতে যথন অন্য কোন সার্কিটে আউটপুটটি যুক্ত থাকে তখন সেই সার্কিট টি এই D/A কনভার্টার সার্কিটের আউটপুট-কে ব্যাহত না করে।

(Refer Slide Time: 17:47)

Calculation

- Let us calculate the voltages at the opamp input when exactly one of the D_i inputs is at 1 (say, $+V$ volts).
- Case 1: Input is 1000

আমরা op-amp এর ইনপুটে ভোল্টেজ হিসাব করি। আমরা ধরে নিছি কোন একটি মুহূর্ষে একটি ইনপুট 1 এ আছে, অন্য তিনটি ইনপুট হলো 0। আমরা এই গুলো ধরে নিয়ে শুরু করছি। প্রথম ক্ষেত্রে আমরা ধরলাম যে ইনপুট হলো 1 0 0 0; মানে শুধু একটি বিট হলো 1 আর অন্য বিট গুলি 0। এখন দেখি এখানে কি হচ্ছে। সার্কিট টা এইরকম দেখতে হয়। এই সার্কিটে সবচেয়ে ডান দিকের বিট টি হল সবচেয়ে গুরুত্বপূর্ণ বিট(most significant bit) আর অন্য তিনটি হল নিম্নতর গুরুত্বপূর্ণ বিট, এগুলো সবই 0।

আমরা এখন এই সার্কিটটিকে বিশ্লেষণ করব। আপনি দেখুন এখানে দুটি রোধ আছে $2R$ এবং $2R$, এরা গ্রাউন্ড(ground) এর সাথে parallel-এ যুক্ত আছে। এটি একটি একক রোধ যেটি গ্রাউন্ডের সাথে যুক্ত আছে, তার তুল্য। এইভাবে আমরা সরলীকরণ করব।

(Refer Slide Time: 19:17)

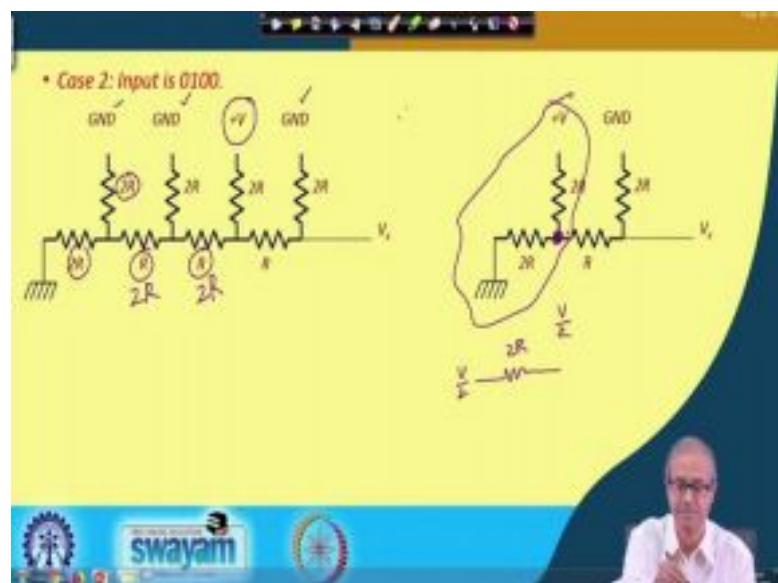
Calculation

- Let us calculate the voltages at the opamp input when exactly one of the D_i inputs is at 1 (say, $+V$ volts).
- Case 1: Input is 1000.

এটা R হয়ে যাবে, তখন সেই R এবং এই R series-এ আসবে এবং সোটি রোধ আবার 2R হয়ে যাবে। সুতরাং, এই 2R এবং এই 2R আবার parallel এ আসবে, ও সেটা R হয়ে যাবে। সেই R এবং এই R series এ আবার 2R হয়ে যাবে, এই 2R ও 2R parallel-এ R হয়ে যাবে। এই R ও এই R মিলে 2R হয়ে যাব। সুতরাং, অবশ্যে, আপনি এই ধরনের একটি সার্কিট পাবেন। এইটি সমতুল্য সার্কিট --- 2R যুক্ত থাকে $+V$ এর সাথে এবং এই দিকটি সমতুল্য ভাবে 2R যুক্ত থাকে গ্রাউন্ডের সাথে।

এটি voltage divider সার্কিট। আউটপুট ভোল্টেজ $V_o = 2R / (2R + 2R) = V / 2$.

(Refer Slide Time: 20:39)

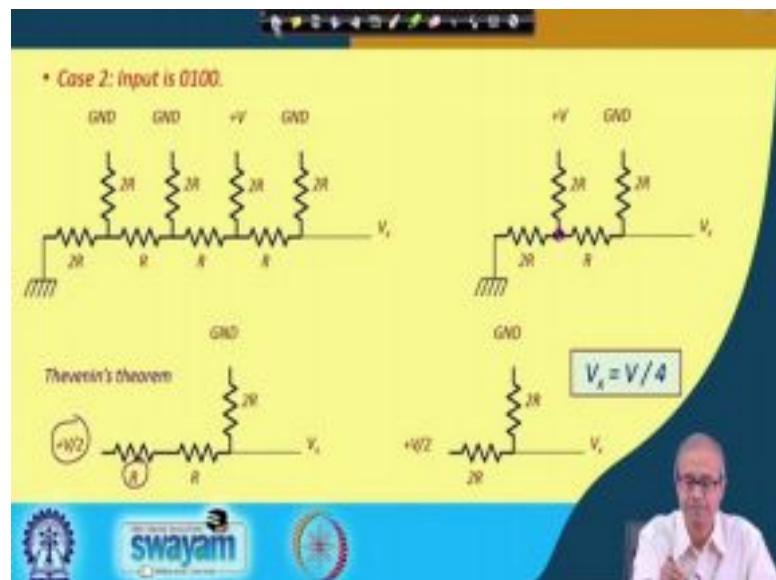


আমরা ধরে নিতে পারি এই দ্বিতীয় বিট টি হল 1। তার মানে, আমি এখানে একটা $+V$ প্রয়োগ করছি, অন্য তিনটি গ্রাউন্ড করা আছে। আগের ক্ষেত্রে যা করেছি সেই একই যুক্তি অনুসরণ করে, এই দুটি রোধ series এ আছে। তারা হল R, ওই R এবং এই R series এ 2R হয়। আবার এই 2R ও এই 2R parallel এ হয় R . R এবং R series- এ থাকলে 2R হয়। সুতরাং, আমরা একটি জায়গায় আসছি যেখানে আমাদের সমতুল্য সার্কিটটি এই রকম দেখতে লাগ-এ। আমার এখানে $+V$ আছে, 2R এখানে আছে, বামদিকে সমতুল্য রোধ হল 2R. দেখুন আমি এখানে এদেরকে parallel এ যুক্ত করতে পারবো না কারণ এদের একটি গ্রাউন্ড এর সাথে যুক্ত আছে এবং অন্যটি যুক্ত আছে $+V$ এর সাথে। আগের ক্ষেত্রগুলিতে উভয়ই গ্রাউন্ডের সাথে যুক্ত ছিল, সে কারণেই আপনি parallel combination ফর্মুলা ব্যবহার করতে পেরেছে।

এখানে আমরা Thevenin-এর উপপাদ্য প্রয়োগ করব। Thevenin- এর উপপাদ্য বলছে যদি আমাদের এইরকম একটা সার্কিট থাকে তবে আমরা এই পয়েন্টে দেখি। পুরো সার্কিট টা একটা ভোল্টেজের উৎস(voltage source) দিয়ে, যে ভোল্টেজ এখানে আসছে, সেটা দিয়ে প্রতিস্থাপিত করা যাবে; আপনি

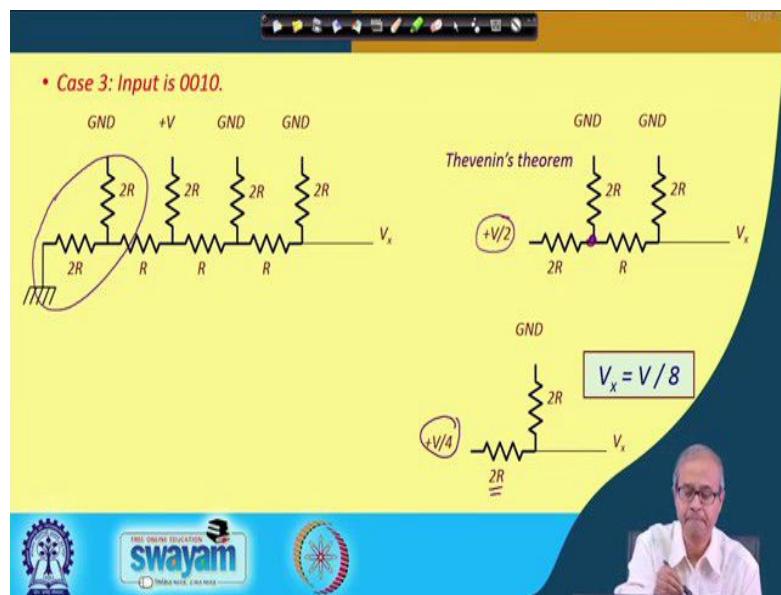
সার্কিটের অবশিষ্ট অংশ ভুলে যান, সার্কিটের এই অংশ ভুলে যান, শুধুমাত্র এই অংশটি। সুতরাং, ভোল্টেজ টি হবে $V/2$, এই ভোল্টেজকে 0 ধরে নিয়ে, আপনি একে গ্রাউন্ড যুক্ত করুন এবং সমতুল্য দেখুন; এটা হল $2R$ । সুতরাং, আপনি ধরে নিতে পারেন একটি ভোল্টেজ $V/2$ একটি রোধ $2R$ এর সাথে series এ থাকবে।

(Refer Slide Time: 23:03)



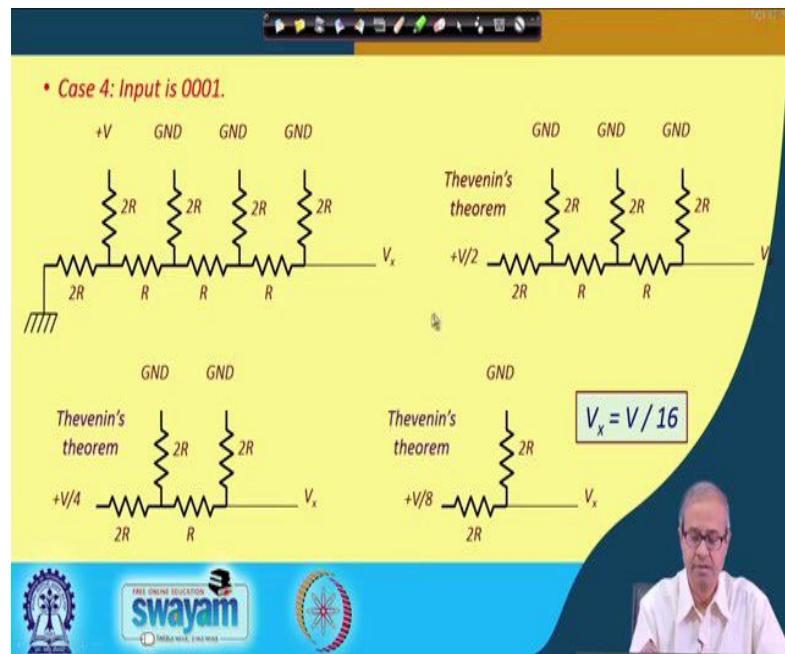
সুতরাং R এবং R series-এ এলে $2R$ হয়, এবং $2R$ ও $2R$ parallel-এ; এখানে এটা $V/2$ হয়। এইটা এইরকম কিছু একটা হবে: $V/2$ এখানে, $2R$ এখানে, এবং $2R$ এখানে গ্রাউন্ড এর সাথে যুক্ত। আবার এটি একটি voltage divider সার্কিট, একটি দিক $V/2$ আর অন্য দিক গ্রাউন্ড। সুতরাং, আউটপুট হবে $V/4$. আগের ক্ষেত্রে যেখানে MSB হল 1, এটা ছিল $V/2$ । এখন যখন দ্বিতীয় MSBও 1, এটা হল $V/4$; এটা অর্ধেক হয়ে গেল।

(Refer Slide Time: 24:11)



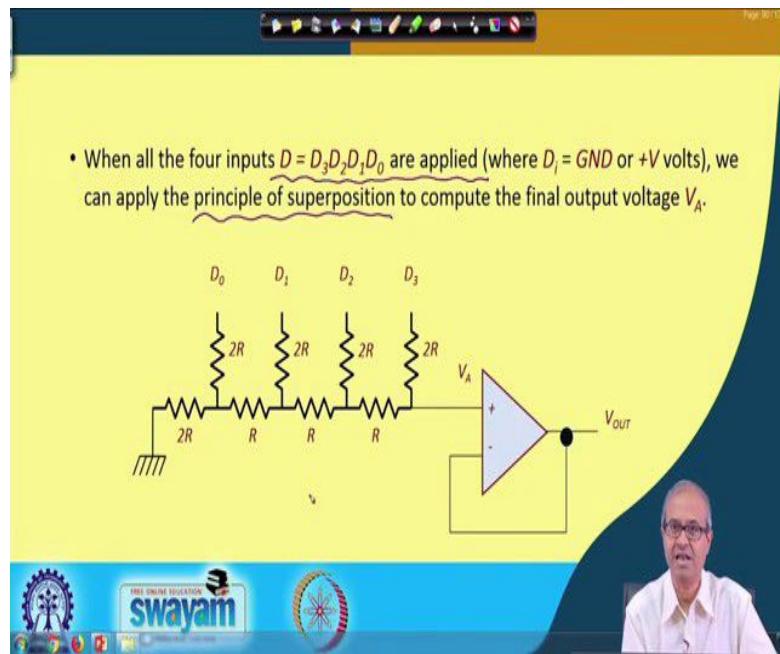
এখন পরবর্তী বিটের দিকে তাকাই। তাহলে যখন আমি 0 0 1 0 প্রয়োগ করি তখন কি হয়? একইভাবে এখানে প্রথমে দুটি রোধ থাকে, আপনি parallel -এ যুক্ত করতে পারেন এবং তারপর একটি series-এ যুক্ত করুন, তাহলে এর সমতুল্য সার্কিট পাবে। সুতরাং আপনার কাছে এখানে $V/2$ আছে এবং আপনি Thevenin এর উপপাদ্য প্রয়োগ করে এটিকে আবার যুক্ত করতে পারেন। এই পয়েন্টে আপনি এখানে $V/2$ পাবেন, এবং $2R$ এখানে আর আপনি আবার Thevenin- এর উপপাদ্য প্রয়োগ করুন।
 এখানে আপনি $V/4$ ও $2R$ পাবেন। অবশেষে আপনি এই ধরনের একটি সার্কিট দেখবেন; আবার একটি resistance divider $V/4$, $2R$, $2R$ থেকে গ্রাউন্ড। সুতরাং, ভোল্টেজ হবে $V/8$ । তাহলে আপনি দেখছেন $V/2$, $V/4$, $V/8$; এটা ক্রমশ অর্ধেক হয়ে যাচ্ছে। আমরা এখন দেখি যে যেখানে ইনপুট হল 0 0 0 1 সেখানেও একই জিনিস হচ্ছে কি না।

(Refer Slide Time: 25:19)



এখানে প্রথম পদক্ষেপ থেকেই আপনাকে Thevenin এর উপপাদ্য প্রয়োগ করতে হবে কারণ একদিকে গ্রাউন্ড আর অন্যদিকে $+V$ আছে। সুতরাং, আপনি এখানে $V/2$ প্রয়োগ করুন ও একে প্রতিস্থাপিত করতে $2R$ প্রয়োগ করুন, আপনি এ ধরনের একটি সার্কিট পাবেন। এখানে আপনি আবার Thevenin এর উপপাদ্য প্রয়োগ করুন, আপনি $V/4$ এবং $2R$ পাবেন। আপনি আবার Thevenin এর উপপাদ্য প্রয়োগ করলে $V/8$ এবং $2R$ পাবেন, আপনি দেখবেন V_x হবে $V/16$ । তো, আপনি দেখতে পাচ্ছেন যখন বিট গুলি MSB থেকে LSB এর দিকে যাচ্ছে, প্রতিটি বিটের মান হয়ে যাচ্ছে $V/2$, $V/4$, $V/8$ এবং $V/16$ ।

(Refer Slide Time: 26:19)



এখন প্রশ্ন হলো যখন চারটি বিটের সবকটাই একসাথে প্রয়োগ করা হয়, তখন কি হবে? যখন এই বিট গুলো একসাথে প্রয়োগ করা হয়, তার জন্য সার্কিট তত্ত্ব(theory) principle of superposition নামের একটি অতি পরিচিত উপপাদ্য আছে. এটা বলে যে যখন লিনিয়ার(linear) সার্কিটে অনেকগুলি ইনপুট প্রয়োগ করা হয়; রোধ যখন linear, তখন আউটপুট হবে যেখানে আপনি ইনপুট গুলি একটি একটি করে প্রয়োগ করছেন তাদের আউটপুট গুলির সমষ্টির সমান। সুতরাং আপনি যাই প্রয়োগ করুন না কেন আপনি কল্পনা করতে পারেন যে কোন একটি সময় একটি করে ইনপুট প্রয়োগ করা হচ্ছে, আপনি হিসাব করলেন, সমস্ত ভোল্টেজকে যোগ করলেন আর এই চূড়ান্ত ভোল্টেজটা পাবেন। principle of superposition-এর মাধ্যমে আপনি এই সার্কিট কে ব্যবহার করে চূড়ান্ত গণনা টি করতে পারবেন যেটা আপনাকে চূড়ান্ত আউটপুট দেবে।

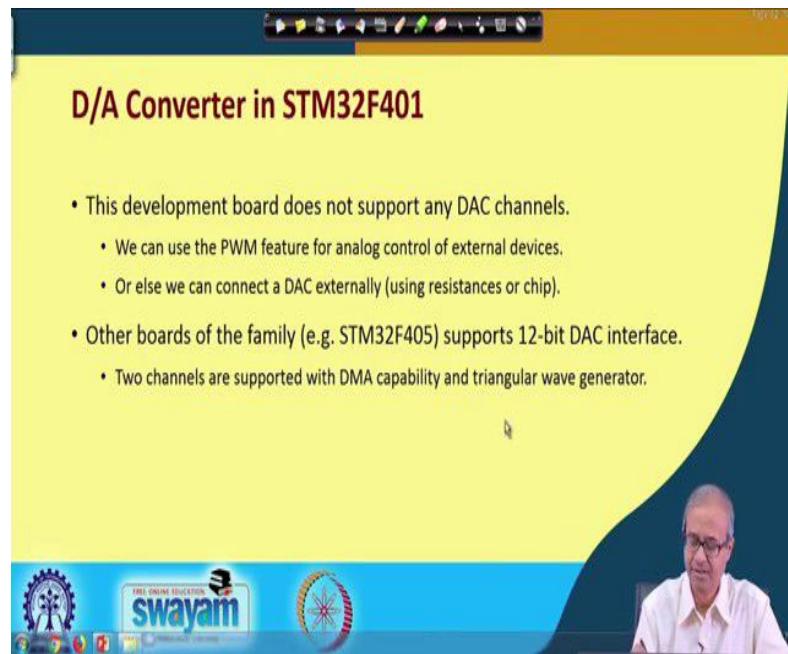
(Refer Slide Time: 27:23)

আপনি এই ধরনের একটি ফল পাবেন।
পরিষ্কার দেখা যায়, V_A হল D এর সমানুপাতিক।

(Refer Slide Time: 28:11)

সুতরাং, সাধারণভাবে N -বিট D/A কনভার্টার(converter) এর জন্য, আপনি যদি k তম বিট টি দেখেন, সেটির অবদান সাধারণত $V / 2^{N-k}$ হয়।

(Refer Slide Time: 28:33)



লক্ষণীয় বিষয় হল STM32F01 বোর্ড(board) যেটা আমরা ব্যবহার করছি, D/A কনভার্টার(converter) এর জন্য তার কোন সাপোর্ট নেই, কিন্তু আমি বলেছি যে বোর্ড টিতে আগেই PWM ক্ষমতা আছে এবং PWM ব্যবহার করে আমরা D/A রূপান্তরের মতই কিছু করতে পারি। এছাড়াও কিছু ক্ষেত্রে আছে যেখানে 12-বিট D/A কনভার্টার(converter) বোর্ডে তৈরি করা থাকে।

এর সাথেই আমরা ডিজিটাল থেকে এনালগ রূপান্তরকরণ(conversion) পদ্ধতি সংক্রান্ত বক্তৃতাটির শেষে এসে পড়েছি। পরবর্তী বক্তৃতা, আমরা এর বিপরীত এনালগ থেকে ডিজিটাল রূপান্তরকরণ(conversion) নিয়ে আলোচনা করব।

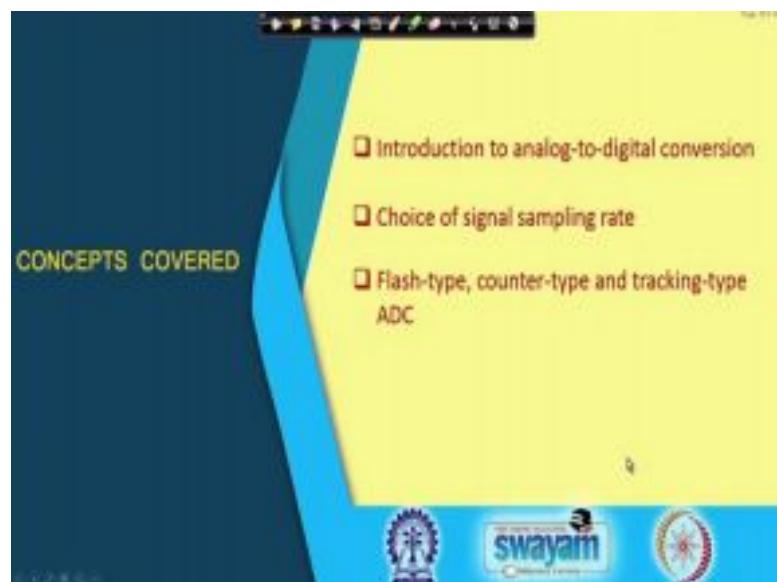
ধন্যবাদ।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 13
Analog to Digital Conversion (Part I)

এই বক্তৃতায়, আমরা এনালগ থেকে ডিজিটাল রূপান্তরকরণ(conversion) নিয়ে আলোচনা করব। আপনি মনে করতে পারবেন যে আমাদের পূর্ববর্তী বক্তৃতায় আমরা এর বিপরীত প্রক্রিয়া, মানে কিভাবে একটি ডিজিটাল value কে একটি সমতুল্য ও সমানুপাতিক এনালগ value তে রূপান্তরিত করা যায়, তা নিয়ে আলোচনা করেছিলাম।

(Refer Slide Time: 00:42)



এই বক্তৃতা আমরা প্রথমে A/D রূপান্তরকরণ এর প্রাথমিক ব্যাপারগুলো নিয়ে আলোচনা করব, এরপর সেখানে থাকবে signal sampling rate, আমরা সেটা নিয়ে খুবই সংক্ষেপে কথা বলব, এবং আমরা কিছু পরিবর্ত্ত A/D converter design দেখব।

(Refer Slide Time: 01:06)

Introduction

- An analog-to-digital converter (ADC) takes an analog input V_A as input, and generates a digital word D as output, where $D \propto V_A$
- Various types of ADC are possible:
 - a) Flash-type ADC
 - b) Counter-type ADC
 - c) Tracking-type ADC
 - d) Successive approximation ADC
 - e) Dual-slope ADC

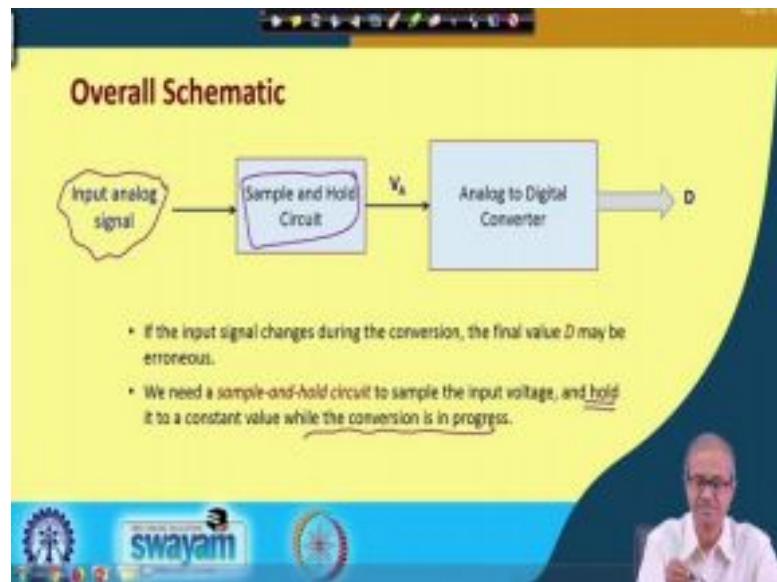
$V_A \rightarrow \boxed{\text{ADC}} \Rightarrow D$

Swayam

আসুন আমরা A/D converter এর প্রাথমিক সূচনা এবং এর বিভিন্ন ধরন, যেগুলো সম্ভব, সেগুলো নিয়ে আলোচনা শুরু করি। একটা এনালগ থেকে ডিজিটাল converter যেটাকে আপনি সংক্ষেপে A/D converter বলেন, সেটা একটি এনালগ ভোল্টেজ V_A কে ইনপুটে নেয় এবং আউটপুট-এ একটি ডিজিটাল মান(value) D দেয়, যেখানে D হল V_A এর সমানুপাতিক।

A/D converter এর বিভিন্ন ধরনের ডিজাইন(designs) সম্ভব, তাদের কয়েকটি এখানে তালিকাভুক্ত করা হলো--- **flash, counter, tracking, successive approximation** এবং **dual slope**. আমাদের বক্তৃতাগুলিতে আমরা প্রথম চার ধরনের নিয়ে আলোচনা করব, শেষেও কয়েকটি নিয়ে আলোচনা করবো না. প্রথম চারটি সাধারণত বেশি ব্যবহৃত হয়।

(Refer Slide Time: 02:17)

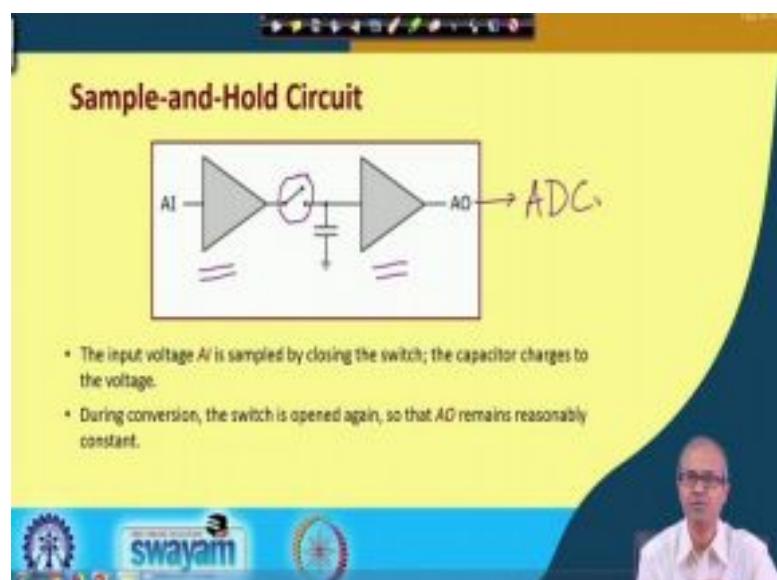


বিভিন্ন ধরনের A/D রূপান্তরকরণ(conversion) নিয়ে আলোচনা শুরু করার আগে, আমরা একটু বৃহত্তর ছবিটির দিকে তাকাই, আমরা এই A/D রূপান্তরকরণ করে কি লাভ করতে চাইছি। কিছু এনালগ সিগন্যাল(signal) আছে, যেটা বাইরে থেকে আসে, এটা একটা কণ্ঠস্বর হতে পারে, বা তাপমাত্রার তরঙ্গরূপ(waveform) হতে পারে। ধরুন আমরা বললাম, একটি শিল্প কারখানায় আমরা কিছু control circuitry প্রয়োগ করার চেষ্টা চালাচ্ছি, সেখানে আমরা একটি চুলা(oven) বা কোন কারখানার তাপমাত্রা, চাপ, আর্দ্রতা নিরীক্ষণ করছি। ওই variable গুলি সারাক্ষণ ধরে নিরীক্ষণ ও অনুভূত করা হয়। এটা একটি তরঙ্গের মতো হবে, যা উপর নীচে পরিবর্তিত হচ্ছে। এটা দরকারী নয় যে তারা একই থাকবে। আর এখানে আপনার A/D converter আছে যেটা ডোল্টেজকে, যেটা অনুভূত মানের সমতুল্য, তাকে সমানুপাতিক ডিজিটাল আউটপুটে রূপান্তরিত করে।

কিন্তু আপনি দেখতে পাচ্ছন যেখানে আরেকটি কার্যকরী ব্লক(block) আছে, যাকে বলে sample and hold circuit. এখন আমাদের sample and hold circuit দরকার কেন? কারণটা খুবই সোজা, ধরুন আমার A/D converter একটি রূপান্তর করতে 5 মিলি সেকেন্ড সময় নেয়। আমার ইনপুট তরঙ্গরূপ(waveform) আসছে, আমি এটা পড়ছি আর রূপান্তর করছি -- এই পদ্ধতিটা 5 মিলি সেকেন্ড সময় নেয়।

এখন ধরুন, আমার ইনপুট তরঙ্গ রূপ(waveform) খুব দ্রুত পরিবর্তন হচ্ছে, এমনকি 5 মিলি সেকেন্ড সময়ের মধ্যেও ইনপুট তরঙ্গ রূপ এর মান পরিবর্তন হচ্ছে। সুতরাং, কি হবে? রূপান্তরের সময় যদি ইনপুট নিজেই পরিবর্তিত হয়, তাহলে চূড়ান্ত ফলটিও ভাস্তু হবে। আমি আশা করি, যখন A/D converter গণনা করছে, তখন ইনপুটটি একটি স্থিতিশীল মান(value) ধরে রাখবে। Sample and hold circuit টি ঠিক তাই করে; এটি একটি নির্দিষ্ট সময়ে ইনপুট টির একটি নমুনা নেয় এবং যতক্ষণ A/D conversion পদ্ধতিটি চলছে ততক্ষণ সেই মানটি ধরে রাখে। এটিই sample and hold circuit-এর মূল উদ্দেশ্য।

(Refer Slide Time: 05:04)

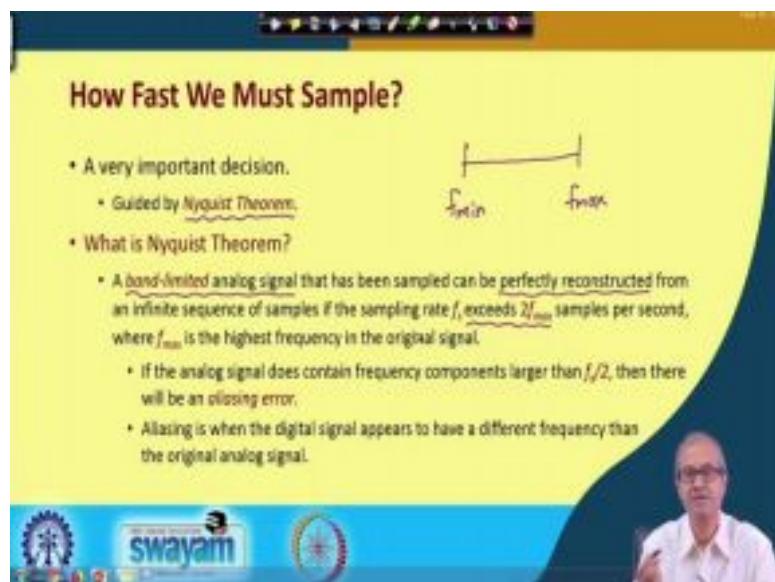


আমি সার্কিট ভায়াগ্রাম(circuit diagram)-এর বিশদ বর্ণনায় যাচ্ছি না যে sample hold circuit এর ভেতরে কি থাকে, কিন্তু পরিকল্পিতভাবে(schematically) আমি আপনাকে প্রাথমিক building block গুলি দেখাচ্ছি। দুটো buffer আছে, এটা এনালগ ইনপুট, এটা আপনার এনালগ আউটপুট, আরেকটি সুইচ(switch) আছে যেটা বৈদ্যুতিন ভাবে(electronically) নিয়ন্ত্রণ করা যায়।

যখনই সুইচ বন্ধ থাকবে তখন এই ক্যাপাসিটর(capacitor)টি সেই ভোল্টেজের চার্জ(charge) পাবে। সুইচটি কিছু সময়ের জন্য বন্ধ থাকবে, যেটা ক্যাপাসিটরের চার্জ পাওয়ার জন্য যথেষ্ট, এরপর সুইচটি আবার খোলা হবে। এখন ক্যাপাসিটরের ডিসচার্জ(discharge) হওয়ার কোন পথ

নেই কারণ এটি একটি buffer, এই buffer- এর ইনপুট রোধ সাধারণত খুব বেশি. সুতরাং, ক্যাপাসিটরটি ডিসচার্জ করতে পারে না। এনালগ আউটপুটটি আপনি আপনার **A/D converter** এ পাঠাতে পারেন। যতক্ষণ **A/D রূপান্তর(conversion)** চলছে ততক্ষণ এই ডোল্টজটির একটি স্থির লেভেল(level) বজায় রাখা হয়।

(Refer Slide Time: 06:34)



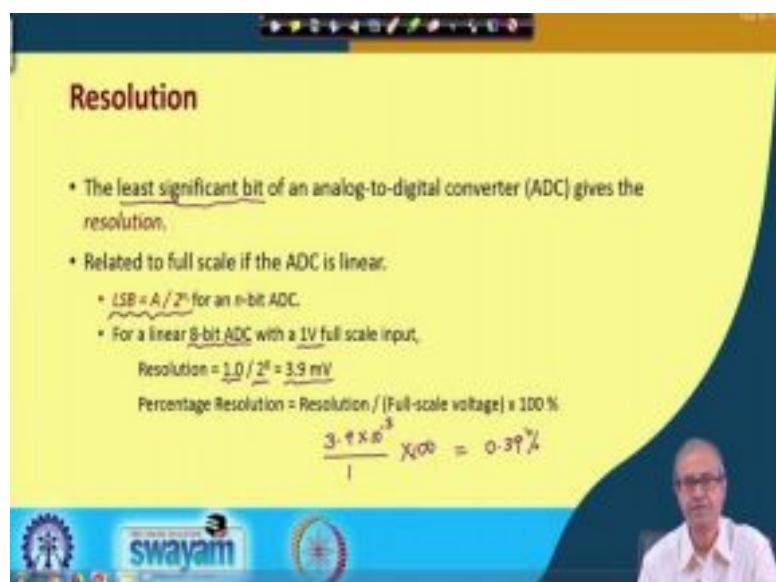
পরবর্তী প্রশ্ন হল **sampling** এর **rate** কত হওয়া উচিত? আমরা কি প্রতি সেকেন্ডে নমুনা নেব, না প্রতি মিলি সেকেন্ড-এ, না প্রতি মাইক্রো সেকেন্ড-এ, কোনটা সবচেয়ে ভালো **sampling rate** হবে। এখন **sampling rate** নির্ভর করে ইনপুট সিগন্যালের(signal) বৈশিষ্ট্যের উপর এবং এখানে একটা ভালো উপপাদ্য আছে সেটাকে বলে Nyquist উপপাদ্য যা আমাদের এই বিষয়ে একটি গাইডলাইন(guideline) দেয়। Nyquist উপপাদ্য বলছে যে একটি সীমিত band-এর এনালগ সিগনাল(signal), যার মানে এমন একটি সিগন্যাল যার কম্পাক্ষণগুলি একটি নির্দিষ্ট সীমায় আছে, ধরি আমরা বলছি f_{\min} থেকে f_{\max} .

ধরুন আমরা এটা **sampling** করছি, সিগন্যাল(signal) টা যদি $2f_{\max}$ এর থেকে বেশিরে **sampling** করা হয়, তবে গ্রহণ কেন্দ্র(receiving end) সম্পূর্ণভাবে পুনর্গঠন করা যাবে। যদি আপনি এটা করেন, তবে গ্রহণ কেন্দ্র(receiving end) আপনি সমস্ত কম্পাক্ষণ গুলি সঠিকভাবে পুনর্গঠন করতে পারবেন। এটাই Nyquist উপপাদ্য বলছে এবং এটা আপনাকে

প্রয়োজনীয় sampling rate কত হবে সে সম্পর্কে একটি গাইডলাইন দেবো তাই f_{\max} এর মান কত হবে, এটা অবশ্যই ইনপুট সিগন্যাল(signal) এর উপর নির্ভর করো।

আপনি যদি এই শর্ত না মানেন, সেক্ষেত্রে আপনার গ্রহণকেন্দ্রিত signal এর কম্পাক্ষগুলি সম্বন্ধে ভুল অনুমান পেতে পারে। আমি বিস্তৃত বর্ণনায় যাচ্ছি না, কিন্তু এটাই sampling এর ফলের সম্বন্ধে প্রাথমিক ধারণা।

(Refer Slide Time: 08:55)



আরেকটি জিনিস আছে যাকে বলে A/D converter এর রেজুলেশন(resolution) , ঠিক যেমন D/A রূপান্তরকরণে আমাদের রেজুলেশন(resolution) ছিল। A/D converter এও আমাদের রেজুলিউশন(resolution) আছে, যেটার মান least significant bit থেকে পাওয়া যায়। এনালগ ইনপুট এর ন্যূনতম পরিবর্তন যার জন্য ডিজিটাল আউটপুট এ +1 পরিবর্তন হয়, তাকে রেজুলেশন(resolution) বলে।

যদি A/D converter এর বৈশিষ্ট্য রৈখিক(linear) হয় তা হলে এটি ইনপুট-এর পরিবর্তনের সাথে সাথে রৈখিক ভাবে পরিবর্তিত হয়। LSB এর রেজোলিউশন(resolution) কে A হিসেবে সংজ্ঞায়িত করা হয়, A হল full scale voltage ডাগ 2^n .

আমরা একটি উদাহরণ নিই। ধরুন আমার একটি A/D converter আছে আর আমার full scale voltage হল 1 volt. সুতরাং রেজিলিউশন(resolution) হবে $1V / 2^8 = 3.9 \text{ mV}$. এটাকে 100 দিয়ে গুন করলে শতাংশ হিসেবেও প্রকাশ করা যায়। এইভাবে আপনি রেজিলিউশন ও পার্সেন্টেজ রেজিলিউশন(percentage resolution) গণনা করতে পারেন।

(Refer Slide Time: 11:18)

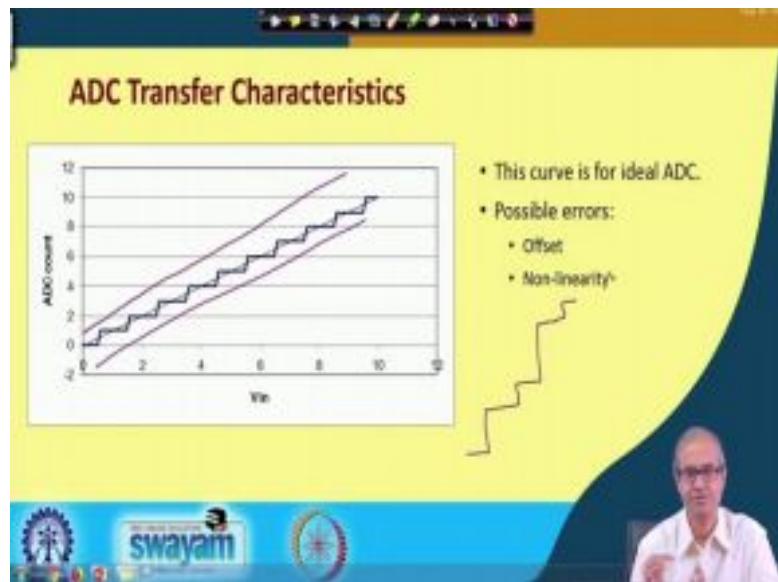
Conversion Time and Bandwidth

- How often can a conversion be done?
 - A few μs to a few ms depending on the technology.
- Input bandwidth
 - Maximum input signal bandwidth
 - Sample and hold input circuitry
 - Conversion frequency

Conversion time এবং bandwidth হল অন্য দুটো প্যারামিটার(parameter)। একটি A/D converter একটি ইনপুট এনালগ ভোল্টেজকে ডিজিটাল সমতুল্য-এ রূপান্তর করতে যতটা সময় নেয়, তাকে বলে Conversion time. কনভার্টার(converter)-এর প্রকার-এর উপর নির্ভর করে এটি খুব দ্রুত হতে পারে, এটা কয়েক ন্যানো সেকেন্ড সময় নিতে পারে, অথবা এটি যদি বেশ মন্ত্র হয় তবে কিছু মিলিসেকেন্ড সময় লাগতে পারে।

ইনপুট bandwidth হল ইনপুটটির সমর্থনযোগ্য সর্বোচ্চ কম্পাক্ষ, এটা সার্কিটের দুটি ডিম্ব অংশের উপর নির্ভর করে। একটি হল sample and hold circuit টা কতটা দ্রুত এবং আরেকটি হলো A/D converter টা কতটা দ্রুত।

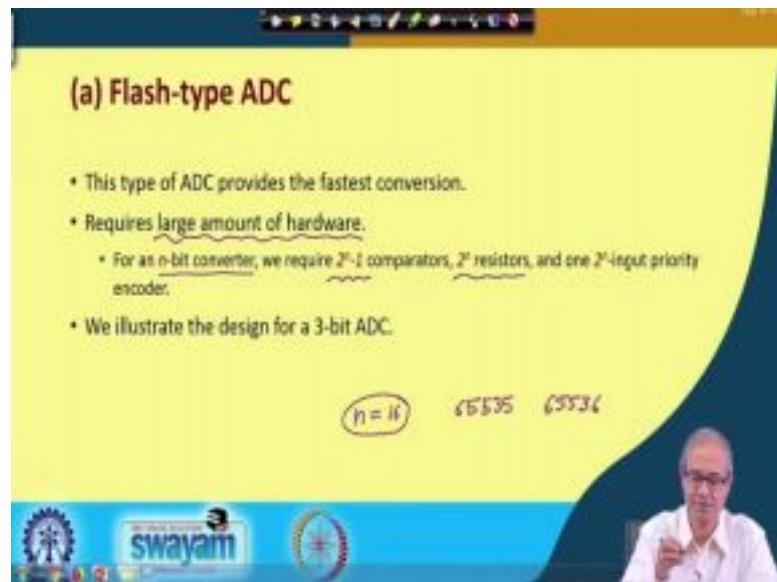
(Refer Slide Time: 12:17)



ADC-এর transfer বৈশিষ্ট্য হিসেবে বলা যায়, যতটা ইনপুট-এর পরিবর্তন হয় ততটাই D/A converter এর মত ধাপে ধাপে বৃদ্ধি হয়। এটা এভাবে বাড়তে থাকে, এই curveটি হল আদর্শ case এর জন্য কারণ এটি একটি সরলরেখা(straight line), কিন্তু সাধারণভাবে এখানে offset error বা nonlinearity error থাকে। Offset error মানে হল এটির পরিবর্তে আপনি আপনার তরঙ্গনূপ, হয় এইরকম দেখবেন অথবা এই রকম দেখবেন।

Nonlinearity মানে ধাপটির উচ্চতা সমান নাও হতে পারে, একটি ধাপ হয়তো দেখলেন খুব বড়, দ্বিতীয় ধাপ হয়তো ছোট, তৃতীয় ধাপ হয়তো বড়, চতুর্থ ধাপ হয়তো ছোট, একেই বলে Nonlinearity. প্রকৃতপক্ষে বাস্তবায়নের সময় এই ধরনের ডুল থেকে যেতে পারে।

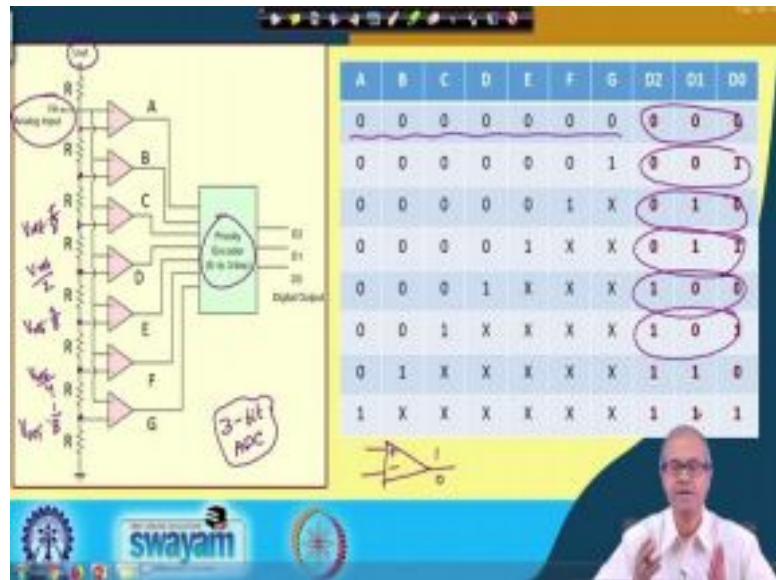
(Refer Slide Time: 13:13)



এখন আমরা বিভিন্ন ধরনের A/D converter ও তারা কিভাবে কাজ করে তা দেখব। প্রথম ধরনের A/D converter হল flash ধরনের A/D converter। এখানে কিন্তু flash memory কে বোঝানো হয়নি, flash মানে খুবই দ্রুত, এটি সবচেয়ে দ্রুত ধরনের A/D কন্পান্তরকরণ(conversion)। এই ধরনের converter এর অপূর্ণতা হলো এতে প্রচুর হার্ডওয়ার লাগে, কত বেশি লাগে? আমরা একটি অনুমান করি, ধরলাম আমরা একটি n -bit A/D converter ডিজাইন(design) করতে চাই।

ধরন বললাম $n=16$ । আমার 2^n-1 টি comparator লাগবে, $n=16$ এর জন্য, এটা হবে 65536। আর আমাদের একটা বিশাল বড় priority encoder দরকার যাতে 65536 টি ইনপুট থাকবে, যেটা বাস্তবে তৈরি করা অসম্ভব। এইজন্য আপনি শুধুমাত্র ছোট flash A/D converter তৈরি করতে পারবেন, n এর বড় মানের(value) জন্য এটা অবাস্তব। আমরা দেখব flash A/D converter কেমন দেখতে এবং এটা কিভাবে কাজ করে।

(Refer Slide Time: 15:09)



এখানে বাম দিকে আমি একটা 3 বিট **flash A/D converter** এর ছবি(diagram) দেখছি।
প্রথম ধাপে একটি রেফারেন্স(reference) ভোল্টেজ V_{ref} ব্যবহার করা হয় আর গ্রাউন্ডের
সাথে অনেকগুলি রোধ যুক্ত থাকে। এই রোধগুলি বিভিন্ন রেফারেন্স(reference) ভোল্টেজ দিয়ে
সাহায্য করে।

সবচেয়ে কমটির ক্ষেত্রে নিচে R ও উপরে $7R$ রোধ থাকে। সুতরাং এখানে কি ভোল্টেজ হবে? V_{ref} . $R / 8R$. এটা $V_{ref} / 8$ ভোল্টেজ দেবে। আপনি যদি পরেরটিতে যান এটি হবে $2R$ ও এটি হবে $6R$. সুতরাং, $2R$ কে ভাগ করছি $(2R+6R)$ দিয়ে। সুতরাং, এটা হল $V_{ref} / 4$. তারপর আপনি
এইটা দেখুন, এটা হবে $V_{ref} . 3 / 8$ এবং এভাবে চলতে থাকবে।

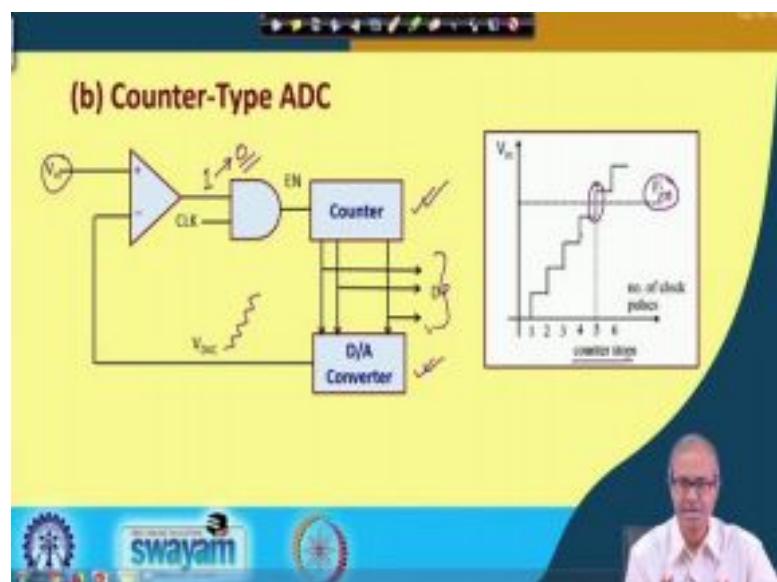
আপনি দেখুন এটা অনেকগুলি রেফারেন্স(reference) ভোল্টেজ দেয়, যেগুলি পরস্পরের সম
দূরত্বে থাকে। এবং এখানে অনেকগুলি ভোল্টেজ **comparator** থাকে। ধরুন আমার এই
ধরনের + এবং - **comparator** আছে, আমার দুটি ইনপুট, আমি দুটি ভোল্টেজ প্রয়োগ
করছি। যদি + ইনপুটটি - ইনপুট এর থেকে বড় হয়, তখন আউটপুটটি 1 হবে; যদি - টি বড়
হয়, তবে আউটপুট 0 হবে, এইভাবে **comparator** কাজ করে। সুতরাং এনালগ ইনপুট
ভোল্টেজ V_{in} এর উপর নির্ভর করে, A B C D E F G এই মানগুলো পরিবর্তন হয়।

priority encoderটি এই **binary combination**কে সমতুল্য ডিজিটাল আউটপুট-এ

রূপান্তরিত করো।

আমরা আউটপুটে প্রয়োজনীয় ডিজিটাল মান(value)টি পাই, এই বিলম্বিত(delay) হল comparator এর বিলম্ব(delay) ও priority encoder এর বিলম্বের(delay) যোগফল। এতে কোথাও কোন পুনরাবৃত্তি(iteration) বা loop নেই, তাই এই ধরনের converter খুবই ঢুত, কিন্তু আমি আপনাকে যেমন বলেছি, এতে প্রচুর hardware এর প্রয়োজন হয়।

(Refer Slide Time: 19:23)



আমরা ব্যবহারিক দিক থেকে তৈরি করা সম্ভব এমন কিছু দেখি। আপনি দেখবেন সেই দিক থেকে D/A converter সহজতর, শুধুমাত্র কিছু রোধ লাগে। আপনি 10-বিট, 12-বিট, 14-বিট, 16-বিট D/A converter খুব সমস্যা ছাড়াই তৈরি করতে পারবেন।

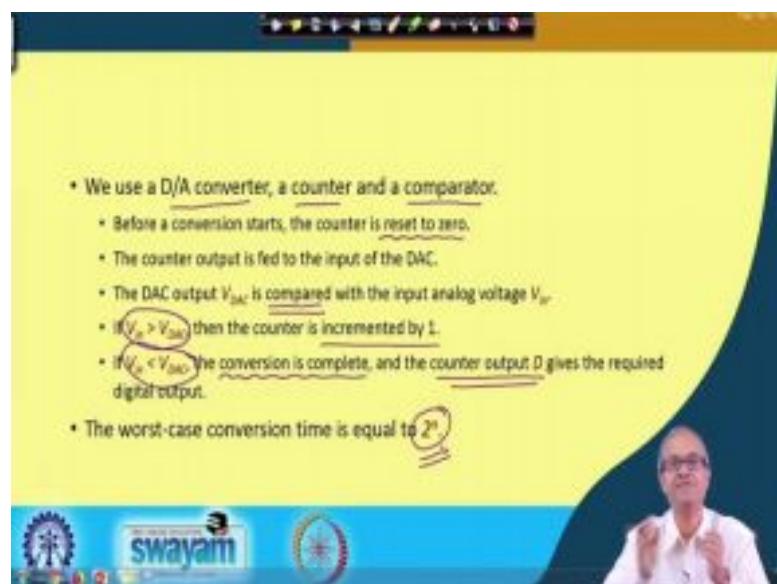
এখন যে পদ্ধতিটি নিয়ে আমরা কথা বলবো, তাতে একটি D/A converter কেন্দ্রীয়(central) block হিসেবে ব্যবহৃত হয়, আর এটা ব্যবহার করে আমরা একটি A/D converter তৈরি করার চেষ্টা করছি। এই পদ্ধতিকে বলে counter type A/D converter। আমাদের এখানে একটি D/A converter আছে, এই D/A converter এর ইনপুট আসে একটি binary counter থেকে। এটা 0, 1, 2, 3, 4 এইভাবে গণনা করে আর এই counter সব সময় গণনা করে না, শুধুমাত্র যখন clock আসে, তখনই শুধু গণনা করে।

এবং এই **clock** আসবে যদি অন্য ইনপুটটি 1 হয়, এটা একটা **AND gate** যদি অন্য ইনপুট 0 হয় তবে **clock** আসবে না। যতক্ষণ অন্য ইনপুট টি 1, এই **counter** গণনা করতে থাকে। এই **counter**টি কে আপনি V_{DAC} ভাবতে পারেন। এই **D/A converter** এর আউটপুট এর কি হবে? এটিও ধাপে ধাপে আস্তে আস্তে বাড়বে।

যখন **counter**টি বাড়তে থাকে, তখন **comparator** এর ইনপুটটিও বাড়তে থাকে, এবং একটি পয়েন্ট আসবে যখন - ইনপুটটি এনালগ ইনপুট ভোল্টেজকে অতিক্রম করবে। আর যখনই এটা অতিক্রম করবে, এর মান হবে 0 এবং **counter**টি গণনা করা বন্ধ করবে। এই ছবিটিতে আমরা ঠিক এই জিনিসটা দেখিয়েছি, ধরলাম এইটা আমাদের V_{in} এখানে আপনি এই **dotted** লাইনটা প্রয়োগ করছেন, এবং এই **counter**টি 0 থেকে শুরু করে ক্রমশ বাড়তে থাকে এবং যখনই এটা অতিক্রম করে তখন **counter** থেমে যায়।

counter এর মানটি হবে **digital** আউটপুট। এইভাবে আপনি একটি **D/A converter** ব্যবহার করে একটি **A/D converter** তৈরি করছেন। যেহেতু আপনি এখানে একটি **binary counter** ব্যবহার করেছেন তাই একে বলে **counter type AD converter**.

(Refer Slide Time: 22:12)

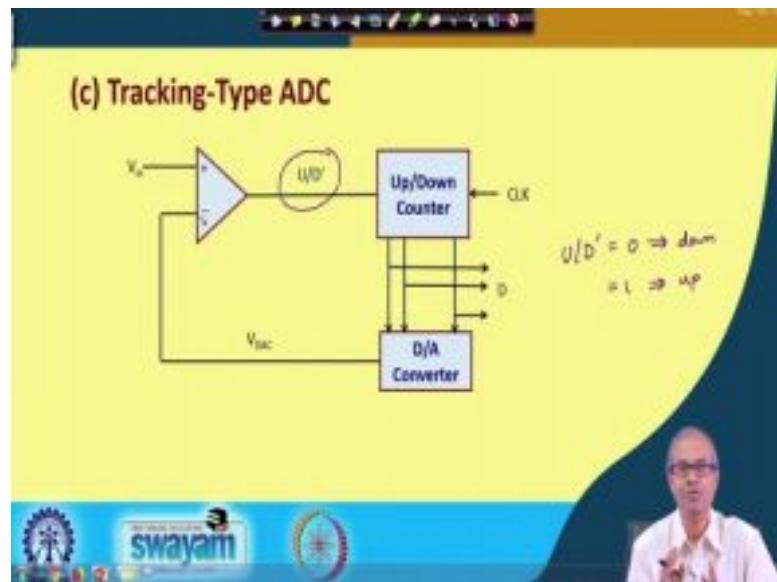


আমরা এখানে ঠিক কি করছি, তা সংক্ষেপে বলার জন্য, প্রতিটি ধাপ লিখে রাখা হল।
আপনার দরকার একটি D/A converter, একটি counter, একটি ভোল্টেজ comparator.
এই রূপান্তর শুরুর আগে counterটি 0 তে রাখা হয়, এবং আমরা দেখেছি counter এর
আউটপুটটি D/A converter এর ইনপুটে যায়।

D/A converter এর আউটপুট comparator এর - ইনপুটে পাঠানো হয়, এবং এনালগ
ভোল্টেজ ইনপুট comparator এর + ইনপুটে যুক্ত করা হয়। সুতরাং আমরা তুলনা করছি,
যদি আপনি দেখেন ইনপুট ভল্টেজটি DAC আউটপুট এর থেকে বড় তখন counter টি 1
বাড়ানো হয়; যার মানে AND gateটি সক্ষম হয়। কিন্তু যখনই এর উল্টাটা হয়, DA
converter এর আউটপুটটি V_{in} কে অতিক্রম করে, তার মানে রূপান্তরকরণ সম্পূর্ণ হয়েছে,
আমরা থামি, এবং counter এর আউটপুটটি হলো A/D converter এর প্রয়োজনীয়
আউটপুট।

এই পদ্ধতির সমস্যা হল, সবচেয়ে খারাপ case- এর ক্ষেত্রে একটি n-bit counter এর জন্য
আমাকে 2^n ধাপের সবকটিই V_{in} খোজার জন্য গণনা করতে হতে পারে। সুতরাং সবচেয়ে
বেশি clock pulse যা লাগতে পারে তা হল 2^n . কিন্তু এটা যেহেতু খুবই সরল আমরা এটা দিয়ে
খুব বড় converter তৈরি করতে পারি।

(Refer Slide Time: 24:00)

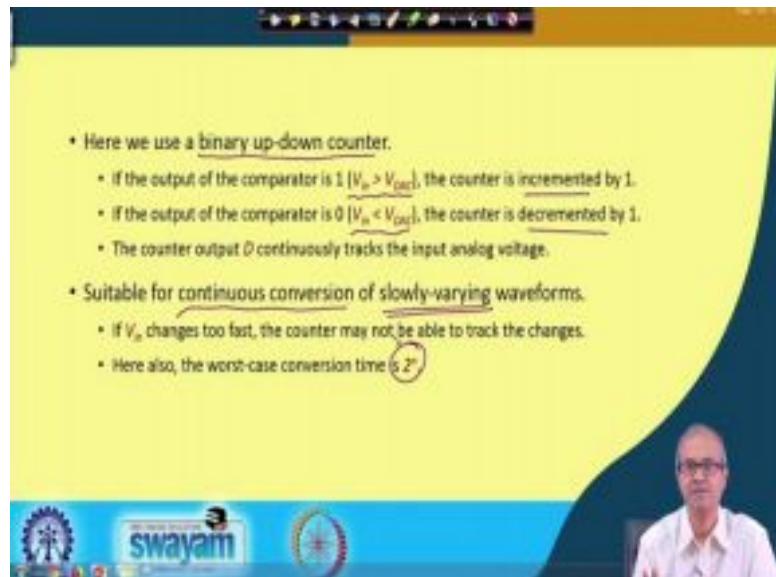


একটু সামান্য পরিবর্তন করে আপনি এটাকে আরো উন্নত করতে পারেন। এখানে আমরা tracking type A/D converter এর কথা বলব। আপনি দেখছেন সার্কিটটি প্রায় একইরকম কিন্তু আমাদেরকে ওই AND gateটি অপসারণ করতে হবে, এবং binary up counterটি up/down counter দিয়ে প্রতিস্থাপিত করতে হবে। Up/down counterটি একটি binary counter যা উপরে 0, 1, 2, 3, 4 অথবা নিচে 4, 3, 2, 1 অবধি গণনা করতে পারে। এখন আমরা কি করে জানব এটা উপরের দিকে গণনা করছে না নিচের দিকে? এখানে U/D' নামের একটি control ইনপুট আছে, যদি এটা 0 হয় তবে counterটি নিচের দিকে গণনা করবে, যদি এটা 1 হয় তবে counterটি উপরের দিকে গণনা করবে।

এখানেও একই নীতি মানা হয়, আপনি counter কে শূন্য থেকে শুরু করুন আর তারপর comparator টি D/A converter এর আউটপুট V_{in} এর সাথে তুলনা করবে। যদি ইনপুটটি এর থেকে বড় হয়, তা হলে U/D' আউটপুট 1. সুতরাং counterটি উপরের দিকে গণনা করবে, যদি D/A converterটি বড় হয়, তাহলে counterটি নিচের দিকে গণনা করবে। সুতরাং, D/A converter এর আউটপুট ইনপুট ভল্টেজকে track করবে। এখানে আপনি ধরে নিতে পারেন কোন sample and hold circuit নেই। ইনপুট তরঙ্কারটি আস্তে আস্তে পরিবর্তন হচ্ছে এবং D/A converter এর আউটপুট এটাকে track করার চেষ্টা করছে, এটা

ইনপুটটি কোন দিকে পরিবর্তন হচ্ছে তার ওপর নির্ভর করে হয় বাড়ছে অথবা কমছে আর এটা ক্রমেই হয়ে চলেছে; এইভাবে tracking AD converter কাজ করে।

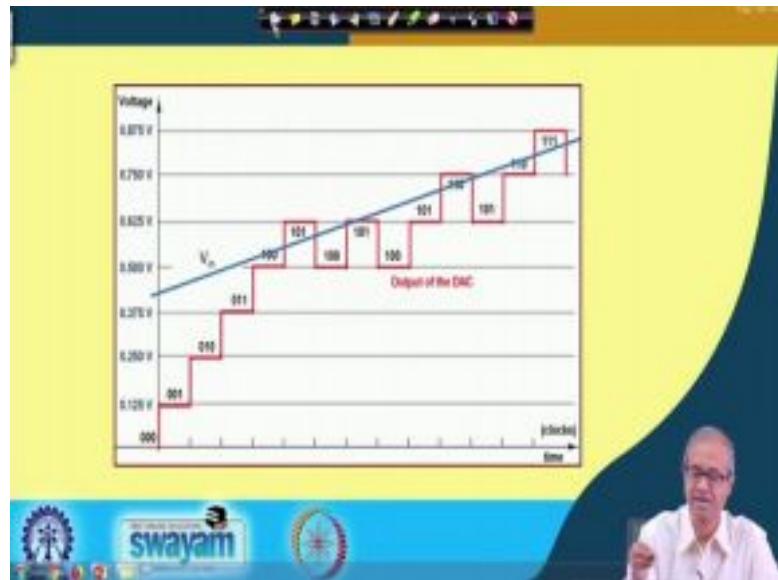
(Refer Slide Time: 26:00)



এখানে আমরা একটি **binary up down counter** ব্যবহার করছি, যদি **comparator** এর আউটপুট 1 হয়; তার মানে V_{in} হল V_{DAC} এর থেকে বড়, আমরা **counter**টি বাড়াই, যদি এটা বিপরীত হয় আমরা **counter**টি কমাই। সুতরাং, এখানে রূপান্তর সম্পূর্ণ হয়ে গেছে বলে কিছু নেই, এই রূপান্তর ক্রমে হয়েই চলেছে আর আমরা ইনপুট ভল্টেজকে **track** করেই চলেছি। সেই সমস্ত ক্ষেত্রে যেখানেই ইনপুট ভল্টেজ খুবই আস্তে পরিবর্তন হচ্ছে এবং আমাদের ইনপুটটি সারাক্ষণ **track** করার দরকার সেখানে আপনি এই ধরনের **AD converter** ব্যবহার করতে পারেন।

এটা খুব আস্তে পরিবর্তনকারী তরঙ্গাকারের একটানা রূপান্তরের জন্য উপযুক্ত, কিন্তু সবচেয়ে বাজে ক্ষেত্রে রূপান্তর এখনো 2^n কারণ ইনপুট খুব বেশি হয় তাহলে V_{in} এ পৌঁছতে আপনাকে 2^n ধাপের সমস্ত ধাপ গণনা করতে হবে। সবচেয়ে বাজে ক্ষেত্রের(case) **delay complexity** আপনি এখানে উল্লিখিত করছেন না।

(Refer Slide Time: 27:09)



ইনপুটটি কিভাবে track করা যায় সেটা দেখাতে, ধরুন আমি এই নীলরেখাটা দেখাচ্ছি, এই নীল রেখাটাকে আমরা আমাদের ইনপুট তরঙ্গাকার বললাম।

এবং এই লাল রেখাটি D/A converter-এর আউটপুটকে track করছে। আপনি দেখছেন এটা কিভাবে পরিবর্তন হয়, এটা বেড়ে যাচ্ছে এখানে এটা অতিক্রম করছে। সুতরাং পরবর্তী ধাপ হল এটা নিচের দিকে গণনা করছে। পরবর্তী ধাপে এটা দেখা যায় যে V_{in} আবার বাড়ছে, এটা আবার নেমে যাবে, এটা আবার বাড়বে, আবার বাড়বে, তারপর নামবে, বাড়বে, বাড়বে। আর আপনি দেখবেন DAC আউটপুটটি ইনপুট তরঙ্গাকার যখন পরিবর্তন হচ্ছে তাকে track করার চেষ্টা করছে, এটিই tracking type A/D converter এর প্রধান বৈশিষ্ট্য।

এর সাথে আমরা এই বক্তৃতাটির শেষে এসে পড়েছি। এখানে আমরা তিনটি ভিন্ন ধরনের(type) A/D রূপান্তরের ডিজাইন(design) নিয়ে আলোচনা করলাম. Flash type, counter type and tracking type. পরবর্তী বক্তৃতায় আমরা অন্য একটি ধরনের A/D converter নিয়ে কথা বলব, যেটা প্রকৃতপক্ষে counter type বা tracking type এর তুলনায় দ্রুততর। এটা ভাবা যায় যে n-বিট রূপান্তরের জন্য আপনার শুধুমাত্র n সংখ্যক clock pulse-ই দরকার এবং যেটা ব্যবহারিক দিক থেকে একদম ঠিক; একটি 16-বিট AD converter এর জন্য আপনার শুধুমাত্র 16 টি clock pulse লাগবে, 65000 টি clock pulse লাগবে না। এইটা

আমরা আমাদের পরবর্তী বক্তৃতায় আলোচনা করব।

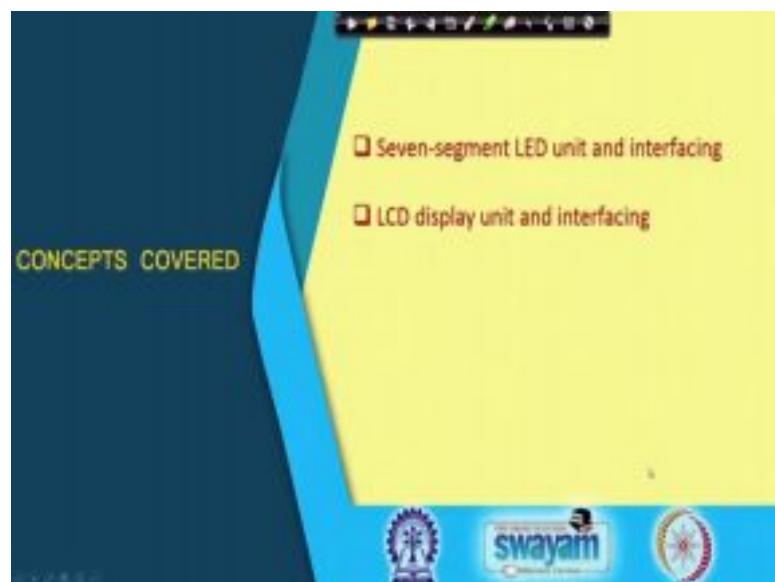
ধন্যবাদ।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 14
Analog to Digital Conversion (Part II)

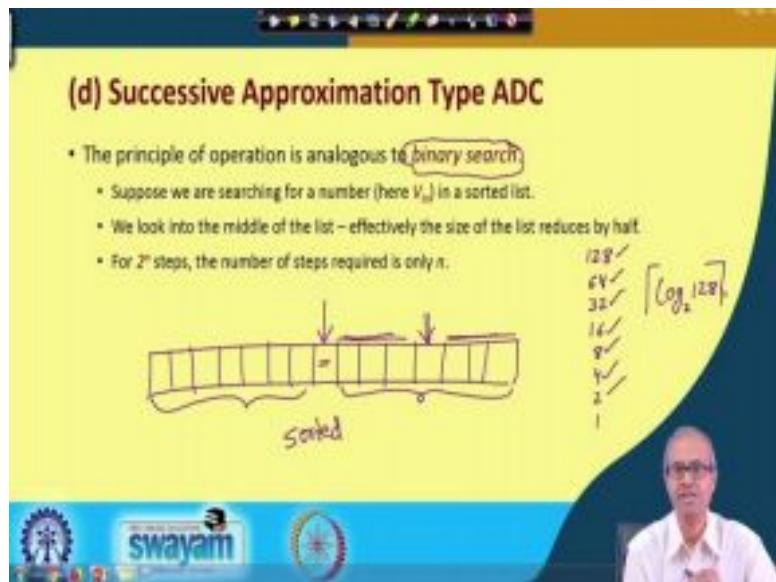
আমরা এনালগ থেকে ডিজিটাল রূপান্তর নিয়ে আলোচনা চালিয়ে যাব।

(Refer Slide Time: 00:27)



এই বক্তৃতায় আমরা প্রথমে আরেক ধরনের A/D converter নিয়ে কথা বলব, যেটা counter type অথবা tracking type converter এর মতই, যেগুলো নিয়ে আগের বক্তৃতায় আলোচনা করেছিলাম, কিন্তু রূপান্তরের সময় হিসাবে অনেক কার্যকরী। এবং ARM development বোর্ড-এ (board), যেটা আমরা ব্যবহার করব তাতে কি ধরনের A/D রূপান্তরের সুবিধা থাকে, এই নিয়ে আমরা আলোচনা করব।

(Refer Slide Time: 01:03)



A/D রূপান্তরের এই পদ্ধতিকে বলে successive approximation type A/D converter.

এর কার্যপ্রণালীর নীতির binary search এর সাথে সাদৃশ্য আছে। আপনাদের মধ্যে যারা প্রোগ্রামিং(programming) এর সাথে পরিচিত, যাদের data structure এর প্রাথমিক জ্ঞান ও বোধগম্যতা আছে, আপনারা হয়তো binary search algorithm পড়েছেন।

binary search কি, এটা যারা জানে না তাদের জন্য আমি এটা বলতে চাই। ধরুন আমার কাছে একটি সংখ্যার তালিকা আছে যেটা array তে রাখা আছে, এবং এই সংখ্যাগুলি হয় মানের উর্ধ্বক্রম অনুসারে বা অধঃক্রম অনুসারে সাজানো থাকে।

এখন, আমি একটি সংখ্যা খুঁজতে চাই। আমি একটি সংখ্যা, ধরলাম 45, উপস্থিত কি না, খুঁজে বের করতে চাই। এখন এই array টি যদি ক্রমানুসারে না থাকে, তাহলে আমাকে এটা প্রথম থেকে শেষ অব্দি খুঁজতে হবে। কিন্তু যেহেতু এটা ক্রমানুসারে আছে আমি একটা খুব বুদ্ধিমান কৌশল ব্যবহার করতে পারি। আমি array এর মাঝখান থেকে শুরু করতে পারি; আমি তুলনা করলাম যে মাঝের সংখ্যাটি, আমি যে সংখ্যা খুঁজতে চাই তার থেকে বড় না ছোট। যদি মাঝের সংখ্যাটি এর থেকে ছোট হয়, তার মানে, আমার সংখ্যাটি অবশ্যই তান দিকে থাকবে, যদি এটা অন্যরকম হয়, তাহলে এটা অবশ্যই বাম দিকে থাকবে।

সুতরাং, আপনি দেখছেন একবার তুলনা করে আমি তালিকাটির আকার(size) অর্ধেক করে

ফেললাম। আমি যে অর্ধেকটি নির্বাচন করেছি তার পুনরাবৃত্তি করছি। ধরুন এটা ডান দিকের অর্ধে আছে। আমি আবার মাঝখানে খুঁজলাম, এই অনুসন্ধান এর উপর নির্ভর করে আমাকে হয় বাম দিকে অথবা ডান দিকে দেখতে হবে; আমি এই প্রক্রিয়াটির পুনরাবৃত্তি করছি। ধরুন আমার array তে 128 টি সংখ্যা আছে। একটি তুলনার পর আমি তালিকাটি অর্ধেক কমিয়ে নিয়েছি, এটা 64 হয়েছে। আরেকবার খোঁজার পরে, এটা 32 হয়, আরো একবার খোঁজার পর এটা 16 হয়, এইভাবে এটা 8 4 2 এবং 1 হয়। অবশেষে, যখন শুধুমাত্র একটি সংখ্যা থাকে, তখন আমি বলতে পারি যে এটা, সেখানে আছে কি নেই। সুতরাং, সাত বার তুলনা করা দরকার। 7 কি? এটা $\log_2 128$ ছাড়া, আর কিছুই নয়।

এই **successive approximation technique** হল A/D conversion এর জন্য **hardware** এ **binary search** প্রয়োগ করার মত।

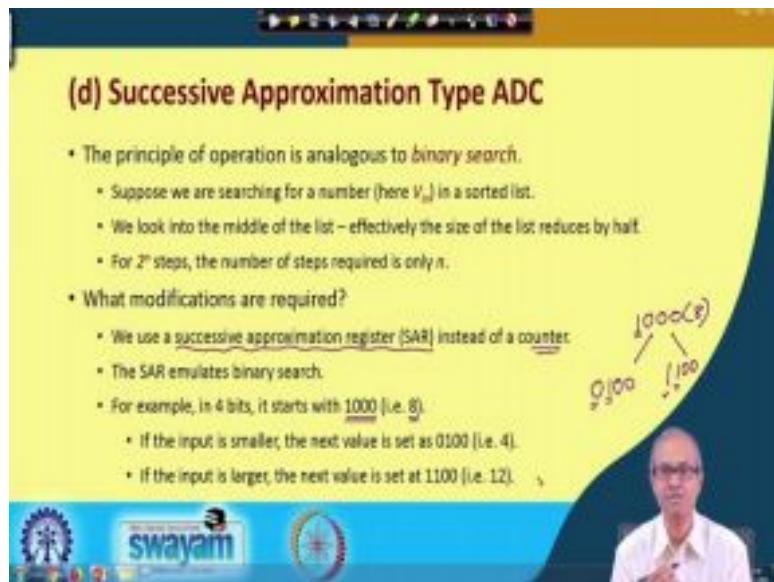
(Refer Slide Time: 04:33)

(d) Successive Approximation Type ADC

- The principle of operation is analogous to **binary search**.
- Suppose we are searching for a number (here V_m) in a sorted list.
- We look into the middle of the list – effectively the size of the list reduces by half.
- For 2^n steps, the number of steps required is only n .
- What modifications are required?
- We use a successive approximation register (SAR) instead of a counter.
- The SAR emulates binary search.
- For example, in 4 bits, it starts with 1000 (i.e. 8).
 - If the input is smaller, the next value is set as 0100 (i.e. 4).
 - If the input is larger, the next value is set at 1100 (i.e. 12).

আপনি দেখছেন 128 এর জন্য $\log_2 128$ টি তুলনা দরকার। সুতরাং, যদি 2^n সংখ্যক সংখ্যা থাকে, তাহলে আমার দরকার হবে $\log_2 2^n = n$. আমার দরকার n সংখ্যক ধাপ। সুতরাং, যদি আমি এটা প্রয়োগ করতে পারি, আমার শুধুমাত্র n সংখ্যক **clock pulse** লাগবে, 2^n সংখ্যক **clock pulse** নয়।

(Refer Slide Time: 05:17)



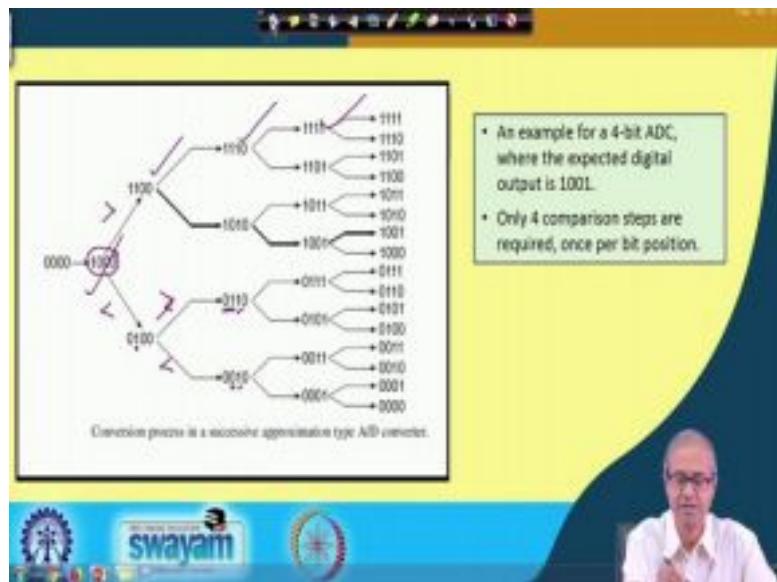
আমরা counter এর পরিবর্তে successive approximation register ব্যবহার করিব।

একটি 4-বিট A/D converter বিবেচনা করুন।

ধরুন বললাম আমরা এই register টা 1 0 0 0 দিয়ে শুরু করলাম। এই 1 0 0 0 মানে 8, যেটা 0 থেকে 15 এর প্রায় মধ্যবর্তী। আমি মধ্যবর্তী(middle) পয়েন্ট(point) টি দিয়ে শুরু করলাম, আমি তুলনা করবো যে এটা বড় না ছোট। ধরুন আমি বললাম যে ইনপুটটি ছোট; তখন এটা বাম অর্ধে 0 থেকে 7 এর মধ্যে হবে। 0 থেকে 7 এর মধ্যবর্তী পয়েন্টটি কি? 4. সুতরাং, 4 কি? 4 হল 0 1 0 0. এবং যদি এটা বড় হয় তখন ডানদিকে হবে; যার মানে, 9 থেকে 15, যার মধ্যবর্তী পয়েন্ট হল 12, 12 হল 1 1 0 0.

সুতরাং আমরা প্রক্রিয়াক্ষে কি করছি? আমরা 1 0 0 0 দিয়ে শুরু করেছিলাম। যদি এটা এর থেকেও কম হয় তাহলে এই 1 কে 0 করা হয় আর পরবর্তী বিট কে 1 করা হয়, কিন্তু যদি এটা অন্যরকম কিছু হয় তবে এই 1 টি 1ই থাকে, আমি এটা পরিবর্তন করি না, কিন্তু এই দ্বিতীয় বিটটি কে আমি 1 করছি। আমি এটাই বারবার করছি এবং এটা binary search কে অনুকরণ করে। এটাই successive approximation register এর ভূমিকা।

(Refer Slide Time: 07:07)



আমি যে ছবিটা(diagram) দেখাচ্ছি আপনি দেখুন। বামদিকে আমি এখনই যেটা বললাম ঠিক সেটা উদাহরণ সহকারে দেখাচ্ছি। SAR-এ আমরা 1 0 0 0 দিয়ে শুরু করি, আমরা এনালগ ইনপুট ভল্টেজ এর সাথে তুলনা করি যে এটা ছোট না বড়।

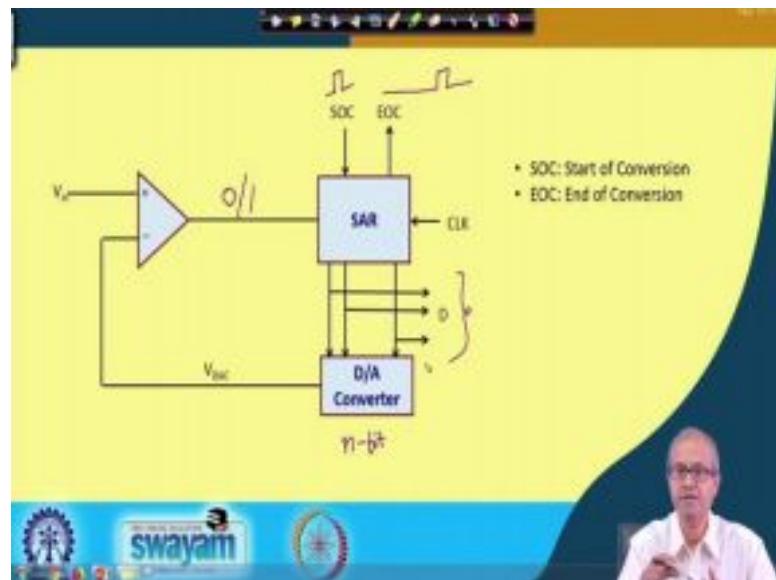
যদি ইনপুট টা SAR আউটপুট এর থেকে ছোট হয়; তখন আমি এখানে ঘাবা। আমি এই বিট(bit) টা কে 0 করলাম, পরের বিট(bit) টা কে 1 করলাম: সুতরাং 0 1 0 0. যদি এটা অন্যরকম হতো, আমি একে 1 1 0 0 করতাম, এবং আমি এই পদ্ধতির পুনরাবৃত্তি করছি। আপনি দেখুন আমি এটাও খেয়াল রাখছি যে এটা ছোট না বড়। যদি এটা ছোট হয়, শেষ যে বিট(bit) টা কে আমি 1 করেছি, এটাকে 0 করব, আর পরবর্তী বিট(bit)টা কে 1 করব।

কিন্তু অন্যদিকে যদি এটা বড় হয়, আমি পরিবর্তন করছি না, পরবর্তী বিট(bit) টা কে আমি 1 করছি। এবং এই পদ্ধতি বারবার হবে। সব সময় আমি পরবর্তী বিটের দিকে তাকাই, কাছের (immediate) বিট(bit) টা কে আমি 0 তে পরিবর্তন করছি আর পরবর্তী বিট(bit) টা কে 1 মান দিচ্ছি, অথবা আমি বর্তমান বিট(bit) টা কে বিরক্ত করছি না শুধুমাত্র পরবর্তী বিট(bit) টা কে 1 এ পরিবর্তন করছি।

সুতরাং, এইভাবে আমি একটা, দুটো, তিনটে, চারটি তুলনা করছি আর অবশেষে আমি A/D converter এর আউটপুট, সেটা যাই হোক না কেন, তার ফলাফল পাচ্ছি। সুতরাং এখানে

তুলনা করার জন্য শুধু মাত্র চারটি ধাপ দরকার।

(Refer Slide Time: 08:57)



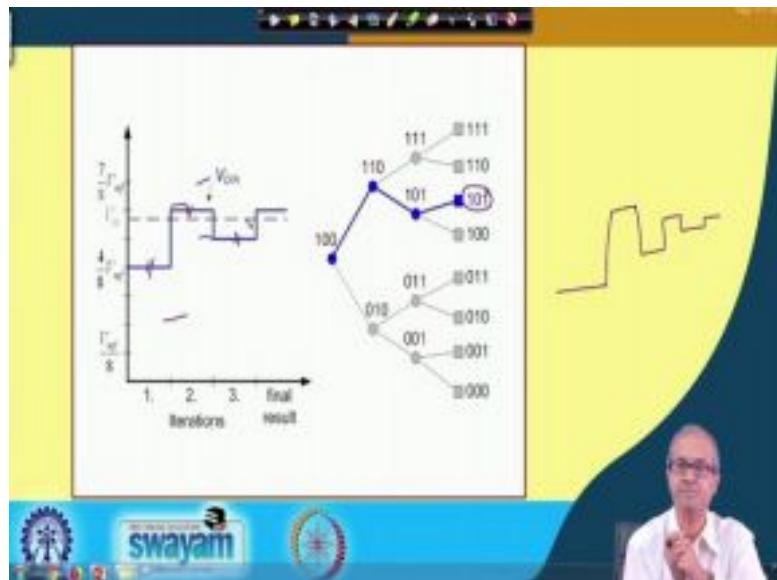
Block diagram এর দিক থেকে দেখলে successive approximation A/D converter কে এইরকম দেখতে লাগে। এখনো এতে একটা Comparator এবং একটা D/A converter আছে, অনেকটা tracking type ADC এর মত; শুধুমাত্র up down counter টি successive approximation register দিয়ে প্রতিস্থাপিত করা হয় এবং এটিই প্রকৃতপক্ষে যুক্তিটা প্রয়োগ করে, যেটা আমরা comparator এর আউটপুট এর উপর নির্ভর করে ০ বা 1 বলেছিলাম সেই সিদ্ধান্ত নয়।

আর যখনই আপনি রূপান্তরের পদ্ধতি শুরু করবেন, সেখানে সাধারণত দুটি ইন্টারফেস(interface) থাকে, SOC এবং EOC. যদি আপনি SOC তে একটি পালস(pulse) প্রয়োগ করেন তবে রূপান্তর শুরু হয়ে যাবে এবং আভ্যন্তরীণভাবে সেখানে 8 টি clock pulse থাকবে। ধরুন আপনি একটি n-বিট converter ব্যবহার করছেন। আপনার একটি n-বিট এর D/A converter দরকার, এই EOC সিগন্যাল(signal) টি n টি clock pulse-এর পর উঁচুতে রাখা হবে। সুতরাং, বাইরের যে কোন যন্ত্র(device) জানবে যে আমার A/D রূপান্তরটি সম্পূর্ণ হয়েছে। এখন আমি D এর মান(value) টি পড়তে পারি। এটা এইভাবে কাজ করো। এই

ধারণাটা খুবই সরল, কিন্তু খুবই নমনীয় ও দ্রুত; এতে শুধুমাত্র n সংখ্যক **clock pulse**

দরকার হয়।

(Refer Slide Time: 10:21)



3-বিট converter এর একটি ছোট উদাহরণ এখানে দেয়া হলো। দেখুন যেভাবে এই binary search এগিয়ে চলে, তার সাথে কিভাবে D/A converter এর আউটপুট পরিবর্তন হয়? আপনি মধ্যেখান দিয়ে শুরু করুন। এইটা আমার মধ্যবর্তী পয়েন্ট(point), আপনি হয় এদিকে যাবেন অথবা এইদিকে, ধরলাম আমাকে উপরের দিকে যেতে হবে। সুতরাং এইটি আমার পরবর্তী পয়েন্ট(point)। পরবর্তী ধাপে আপনি হয় এইখানে উঠে যাবেন অথবা এই দিকে, ধরলাম আপনি এইভাবে এখানে যাবেন। আপনি মধ্যবর্তী পয়েন্ট(point) দিয়ে শুরু করুন, তাহলে আপনি হয়তো এখানে যাচ্ছেন, তারপর আপনি এখানে যাচ্ছেন, তারপর হয়তো আমরা এইভাবে এখানে যাচ্ছি; আপনি এই পয়েন্টটিতে একত্রিত হচ্ছেন।

1 0 1 এই ইনপুট-এর জন্য, আমি এখানে একটি খুব ছোট উদাহরণ দিয়েছি। এটা ওই পয়েন্ট(point) মিলিত হচ্ছে, কিন্তু যদি আপনি D/A converter-এর আউটপুট তরঙ্গাকারে(waveform) দেখেন, তরঙ্গাকারটা এই রকম দেখতে লাগবে।

(Refer Slide Time: 11:39)

এখন ARM microcontroller এর ব্যাপারে বলতে হলে, কি কি ধরনের A/D conversion আছে? আপনি দেখবেন D/A রূপান্তর সহজ, D/A রূপান্তর করতে কিছু রোধ লাগবে, কিন্তু A/D রূপান্তর কঠিনতর। A/D রূপান্তর করতে কিছু বিশেষ circuitry দরকার লাগে।

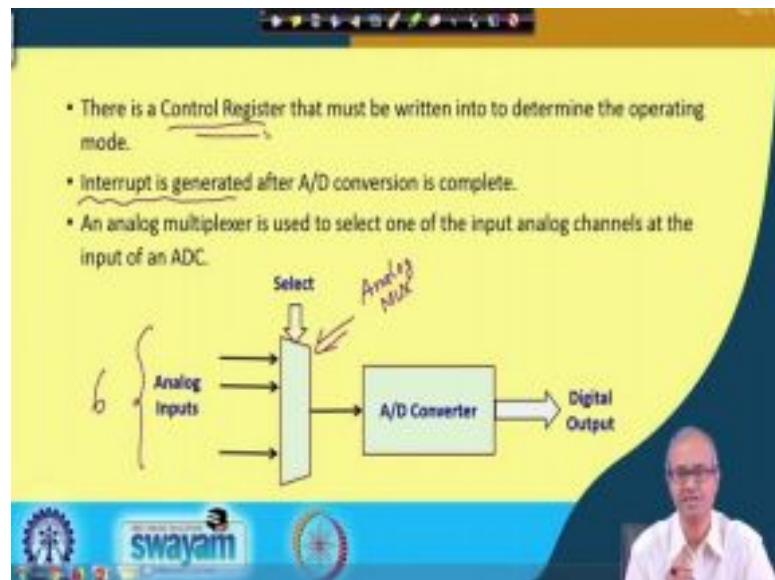
আসুন আমরা পুরোনো প্রজন্মের কিছু ARM এর দিকে দেখি। ARM7 হল একটি আগের প্রজন্মের microcontroller. এতে একটি অন্তর্নির্মিত(built in) 10 বিট(bit) successive approximation A/D converter ছিল। আগে successive approximation technique প্রয়োগ করা হয়েছিল কারণ, এটা দক্ষ, এতে কম hardware লাগে।

এখন, যেহেতু এটা একটা 10 বিট কনভার্টার(coverter), যদি আমি একটা 3.3 volt সাপ্লাই(supply) কে রেফারেন্স(reference) ভোল্টেজ এর সাথে যুক্ত করি, তাহলে প্রতিটি ধাপের উচ্চতা হবে full scale voltage ভাগ $(2^{10} - 1)$. আপনি গণনা করলে দেখবেন এটা 3.23 millivolt হয়। এটা বলে দেয় আপনি কতটা যথাযথভাবে পরিমাপ করতে পেরেছেন। এই resolution হল নির্ভুলতার একটি পরিমাপ।

Processor এরমধ্যে এরকম দুটি ADC module আছে, এদেরকে বলে ADC0 এবং ADC1. ADC0 তে 6 টা channel আছে এবং ADC1 এ 8 টা channel আছে। 6 টা channel মানে আপনি 6 টা ইনপুট রূপান্তরের জন্য যুক্ত করতে পারেন; 8 টা channel মানে আপনি 8 টা

ইনপুট যুক্ত করতে পারেন। এবং **clock** টির সর্বোচ্চ কম্পাক্ষ ছিল 4.5 MHz.

(Refer Slide Time: 13:57)



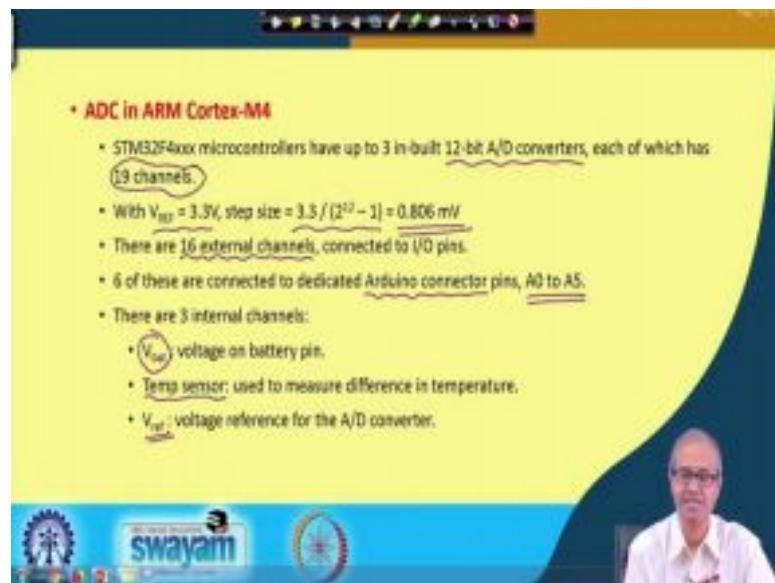
এখন আমরা দেখি এই channel গুলো কিভাবে কাজ করে। এখানে আপনার একটি A/D converter আছে আর এই যে block যেটা আমি দেখাচ্ছি সেটা হল এনালগ মাল্টিপ্লেক্সার(analog multiplexer)। আপনি জানেন একটা digital multiplexer কি; যদি অনেকগুলি ইনপুট থাকে, তাহলে select line এর উপর ভিত্তি করে একটি ইনপুট নির্বাচন করা হয়। Analog multiplexer হল একই রকম, প্রার্থে টা হলো ইনপুটগুলি হল এনালগ ডোল্টেজ; আপনি digital select input line এ কি প্রয়োগ করছেন তার উপর নির্ভর করে, কোন ইনপুটটি আউটপুট এ পাঠানোর জন্য নির্বাচিত হল।

এখন ADC0 তে 6 টা ইনপুট channel আছে। যথাযথভাবে নির্বাচনের মাধ্যমে, এদের মধ্যে কোন একটিকে রূপান্তরের জন্য নির্বাচন করতে পারেন। এদের মধ্যে অনেকগুলিকে একই সাথে ব্যবহার করে, কিন্তু আপনাকে একটির থেকে আরেক টিতে খুব তাড়াতাড়ি switch করতে হবে; আপনি একটি channel রূপান্তর করলেন, তারপর দ্বিতীয় channel রূপান্তর করলেন, তারপর তৃতীয় channel রূপান্তর করলেন। আবার প্রথমটিতে ফেরত পেলেন।

এখন, এটা আপনি Nyquist উপপাদ্য, যেটা আমি বলেছিলাম, তার জন্য সম্ভবত আবার ব্যবহার করতে পারবেন। A/D converter টি সম্ভবত 1 MHz গতিতে কাজ করছে, কিন্তু আপনার ইনপুট তরঙ্গাকারের(waveform) বৈশিষ্ট্যের উপর নির্ভর করে আপনার হয়তো ওই ধরনের sampling rate লাগবে না। হয়তো আপনার sampling rate শুধুমাত্র 1 KHz হবে। সুতরাং, অনেকগুলি চ্যানেল(channel) কে একই A/D converter এ মাল্টিপ্লেক্স(multiplex) করতে হবে এবং আপনি তাদেরকে একই সাথে ব্যবহার করতে পারেন। এখানে ওইরকম দুটি চ্যানেল(channel) আছে ADC0 এবং ADC1, তারা যথাক্রমে 6 টি এবং 8 টি channelকে সহায়তা(support) করে।

A/D রূপান্তর সম্পূর্ণ হওয়ার পরে একটি interrupt signal উৎপন্ন হয়। এটা এভাবে কাজ করে। এবং একটি control register আছে যেটা যথাযথভাবে শুরু করতে হবে। এটিই ARM7 এ ছিল।

(Refer Slide Time: 16:07)

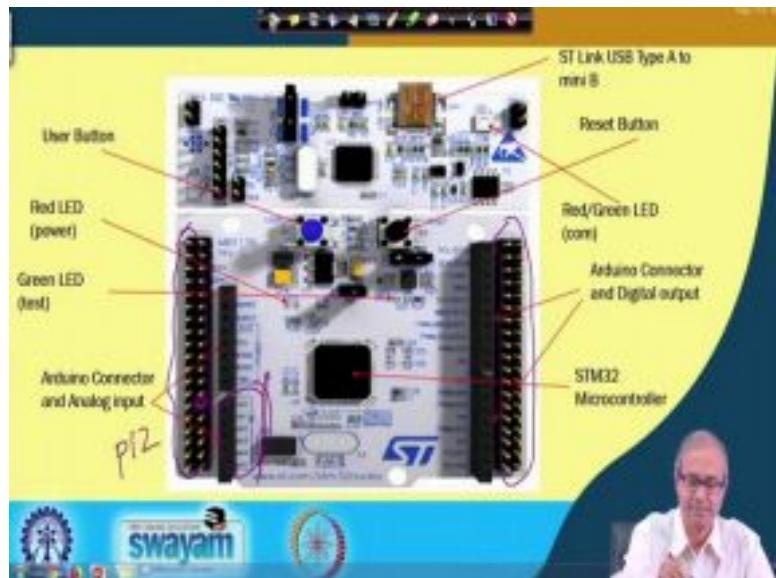


এখন যে বোর্ড(board) টা আমরা ব্যবহার করব, STM32, তার ভিত্তি হলো RM Cortex-M4 microcontroller. এই বোর্ড(board) তিনটি অন্তর্নির্মিত(built-in) 12- বিট A/D converter আছে এবং এদের প্রতিটি 19 টি করে channel কে সহায়তা দিতে পারে; তার মানে,

analog multiplexer এতে এতগুলি ইনপুট রয়েছে। 12- বিট এর জন্য আপনার রেফারেন্স(reference) হল 3.3 volt, আপনার প্রতিটি ধাপ এর উচ্চতা আরো কম হবে। সুতরাং আপনার নির্ভুলতা আরো বেশি হবে, 0.8 millivolts.

এখানে 16 টা বহিরাগত channel আছে যা ইনপুট/ আউটপুট pin সাথে যুক্ত এবং এই 16 টা channel- এর মধ্যে 6 টা Arduino connector pin-এর উপর উপলব্ধ। এগুলোকে বলে A0 থেকে A5. এবং এছাড়া 3 টি অভ্যন্তরীণ channel রয়েছে, যেগুলো system এর স্বাস্থ্য নিরীক্ষণ করার জন্য। এই 3 টি অভ্যন্তরীণ channel যুক্ত থাকে battery-র ভোল্টেজের সাথে, যদি বোর্ড(board) টি battery তে চলে তখন আপনি battery-র ভোল্টেজ দেখতে পাবেন, কিছু বোর্ডে অনুর্নিমিত তাপমাত্রা sensor থাকে। আপনি বোর্ডের তাপমাত্রাও দেখতে পারবেন আর এই মুহূর্তের রেফারেন্স(reference) ভোল্টেজও দেখতে পারবেন। এই সমস্ত অভ্যন্তরীণ জিনিস যা আপনি নিরীক্ষণ করতে পারেন; তাদেরকে অভ্যন্তরীণ channel বলে।

(Refer Slide Time: 17:59)



এখন, আবার এই বোর্ড(board) ফেরত আসছি, দেখুন এই Arduino connector-এ, 6 টা এনালগ ইনপুট লাইন(line) এখানে উপলব্ধ, কিন্তু আরো অনেক এনালগ ইনপুট লাইনকে

সহায়তা করা হয়। যদি আপনি আরো চান তাহলে আপনাকে সেগুলো এই extension pin গুলো থেকে পেতে হবে। STM 32 extension pin গুলোতে, signal pin গুলি আপনার কাছে উপলব্ধ, কিন্তু এখানে শুধুমাত্র Arduino compatible connector pin গুলি আপনার কাছে উপলব্ধ।

যখন পরীক্ষা-নিরীক্ষার জন্য আপনার A/D converter দরকার যেটা আমরা দেখাবো, আমরা খুব বেশি channel ব্যবহার করব না। আমরা Arduino input connector এর সাথে সরাসরি যুক্ত করতে পারি। সেই ক্ষেত্রে ওই pinগুলোকে আমরা A0 A1 A2 A3 A4 A5 নামে অভিহিত করতে পারি। আমরা pin সংখ্যাও দিতে পারি; এগুলো হবে 1 2 3 4 5 6 7 8 9 10 11 12. আমরা এগুলোকে P12 বলবো।

সুতরাং, আপনি সেগুলোকে নাম দিয়ে বা pin সংখ্যা দিয়ে ডাকতে পারেন। আপনি যখন কোন প্রোগ্রাম(program) লিখবেন, আপনি ওই গুলোর মধ্যে যেকোনো নাম ইচ্ছামত ব্যবহার করতে পারেন।

এর সাথেই আমরা A/D conversion নিয়ে আলোচনার শেষে এসে পড়েছি। আগেই আমরা D/A conversion দেখেছি। পরবর্তী বক্তৃতাগুলিতে, আমরা কিছু সাধারণ sensors এবং actuators নিয়ে কথা বলব যেগুলো embedded system এর অ্যাপ্লিকেশন গুলিতে খুবই গুরুত্বপূর্ণ, এবং তাদের অনেকগুলি আমরা হাতে-কলমে(hands-on) session এ প্রদর্শন করে দেখাবো। সেগুলো ব্যবহারের আগে, sensor কি, তারা কিভাবে কাজ করে এবং তাদের কিভাবে microcontroller এর সাথে interface করা যায়, সেই সম্বন্ধে প্রাথমিক ধারণা থাকা খুবই ভালো।

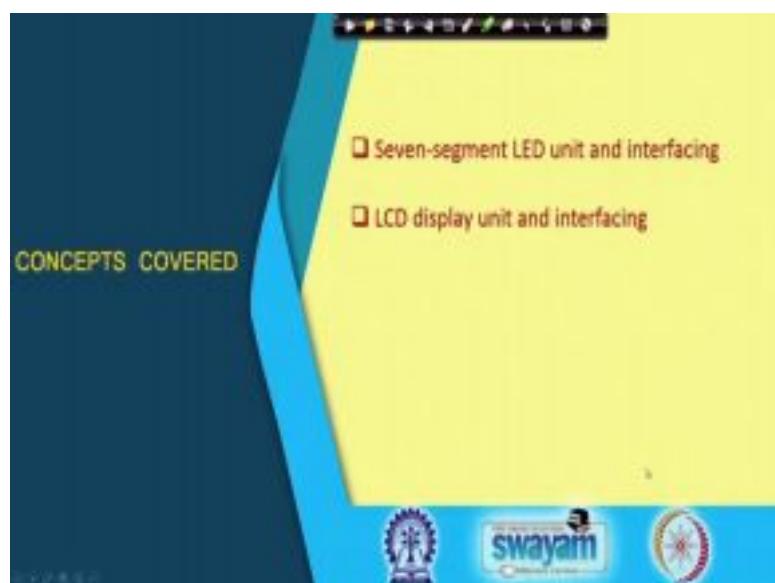
ধন্যবাদ।

**Embedded System Design with ARM Prof.
Indranil Sengupta Department of Computer
Science and Engineering Indian Institute of
Technology, Kharagpur**

**Lecture – 15 Output Devices, Sensors and
Actuators (Part I)**

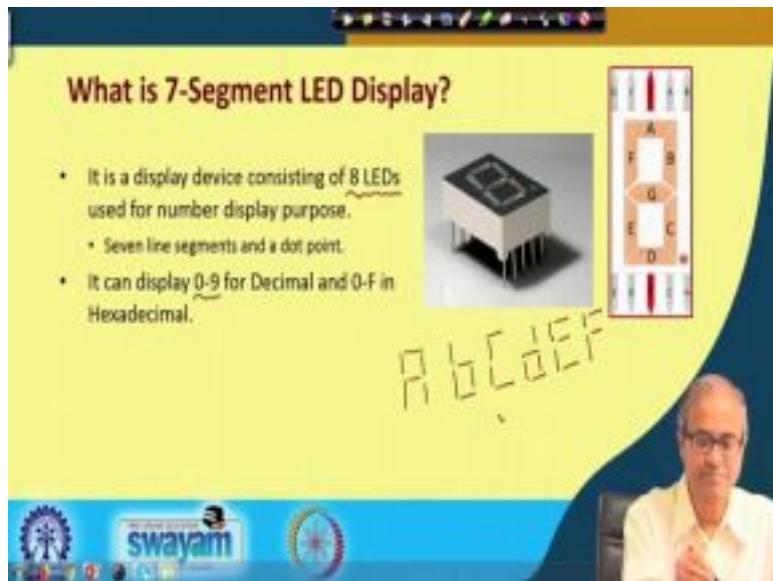
এই বক্তৃতায় আমরা বিভিন্ন আউটপুট ডিভাইস গুলোর উপর একটি সংক্ষিপ্ত আলোচনা শুরু করব, সেনসর্স(sensors) এবং একচুয়াটরস(actuators) যেগুলো বিভিন্ন পরীক্ষা এবং প্রদর্শনের জন্য প্রয়োজন হবে, আমরা আপনাকে stm32 এবং আরডুইনো বোর্ড ব্যবহার করে দেখাব।

(প্লাইড সময় 00:45 দেখুন)



এই বক্তৃতায় আমরা দুটি বিভিন্ন আউটপুট ডিভাইস এবং তাদের বৈশিষ্ট্য সম্বন্ধে কথা বলব। একটা হল আপনাদের খুবই পরিচিত সেভেন সেগমেন্ট এলইডি ডিসপ্লে ইউনিট, সেটা আপনারা অনেকেই জানেন। আর দ্বিতীয়টি হল লিকুইড ক্রিস্টাল ডিসপ্লে(Liquid Crystal Display) বা এলসিডি ডিসপ্লে ইউনিট(LCD display Unit)।

(প্লাইড সময় দেখুন 01.07)



আসুন আমরা 7 সেগমেন্ট এলইডি ডিসপ্লে ইউনিট দিয়ে শুরু করি। 7 সেগমেন্ট এলইডি ডিসপ্লে ইউনিট কি? আপনারা অনেকেই হয়তো এই ধরনের যন্ত্রের সাথে আগেই পরিচিত আছেন। এটি একটি ডিসপ্লে যন্ত্র(display device) যাতে আটটি লাইট ইমিটিং ডায়োড(Light Emitting Diode) আছে, যেগুলো নির্দিষ্ট ভাবে সংজ্ঞিত। এদের মধ্যে সাতটি এলইডি এমনভাবে সাজানো আছে যা ইংরেজি এইট(8) এর মত দেখতে এবং একটি ডট(dot) বিন্দু আছে। এই ধরনের ডিসপ্লে গুলি, এর বিভিন্ন অংশগুলোকে যথাযথভাবে সংক্রিয় করার মাধ্যমে, 0 থেকে 9 পর্যন্ত সংখ্যাসূচক অক্ষর প্রদর্শনের জন্য খুবই দরকারি, যেগুলো সবই লাইট ইমিটিং ডায়োড বা এলইডি(LED), আর এরা একটি বারের আকারের আকৃতির হয়। এগুলোকে জ্বালিয়ে বা নির্বাচিত ভাবে না জ্বালিয়ে আপনি যেকোন সংখ্যাসূচক অক্ষরকে দেখাতে পারেন।

শুধু মাত্র 0 থেকে 9 সংখ্যা-সূচক অক্ষরগুলিই নয়, আপনি A, B, C, D, E, F এই হেক্সাডেসিমাল অক্ষর-গুলিও দেখাতে পারেন। যেমন আপনি A দেখাতে পারেন এইভাবে, লোয়ার কেস b দেখাতে পারেন এইভাবে, এইভাবে C দেখাতে পারেন, এভাবে আবার d দেখাতে পারেন লোয়ার কেস(lower case)- এ, E দেখানো যেতে পারে এভাবে, আর F এভাবে দেখানো যেতে পারে। সুতরাং 0 থেকে 9 ছাড়াও, আপনি হেক্সাডেসিমাল অক্ষর-গুলিও দেখাতে পারেন। সেই জন্য হেক্সাডেসিমাল ডিসপ্লে যন্ত্র হিসেবে 7 সেগমেন্ট ইউনিটগুলি খুবই গুরুত্বপূর্ণ।

এইটা এইরকম দেখতে লাগে, যে, এখানে সাতটা অংশ আছে যাদের নাম্বার দেয়া হয়েছে A,B,C,D,E,F আর আছে একটা ডট(dot) বিন্দু। একটা সাধারণ ডিসপ্লে ইউনিট কি রকম দেখতে লাগে; এখানে দশটা পিন আছে, 5 টা পিন আছে এইন্দিকে, 5 টা পিন আছে ওইন্ডিকে, আর এই দশটা পিনের সাথে প্রত্যেকটার নিজস্ব অংশের সংযোগ আছে, এছাড়াও আছে উভয় দিক থেকেই উপলব্ধ একটা সাধারণ টার্মিনাল।

(স্লাইড সময় দেখুন: 03:27)

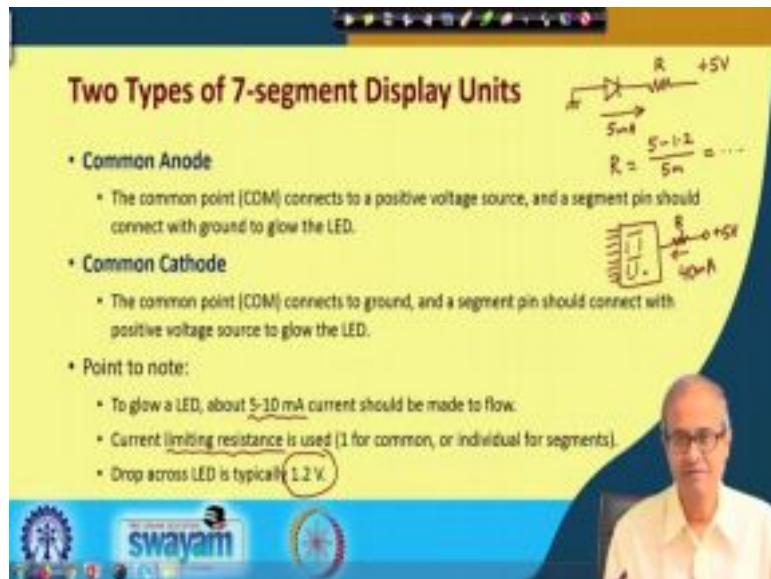
Two Types of 7-segment Display Units

- **Common Anode**
 - The common point (COM) connects to a positive voltage source, and a segment pin should connect with ground to glow the LED.
- **Common Cathode**
 - The common point (COM) connects to ground, and a segment pin should connect with a positive voltage source to glow the LED.

এখন কিভাবে এই এলইডি(LED) গুলো অভ্যন্তরীণভাবে যুক্ত রয়েছে তার উপর নির্ভর করে এই 7 সেগমেন্ট ডিসপ্লে গুলো দুই ধরনের হতে পারে, যয় সাধারণ অ্যানোড অথবা সাধারণ ক্যাথোড। সাধারণ অ্যানোড বলতে বুঝায় যে অ্যানোড টার্মিনাল(terminal) গুলো সব একসাথে যুক্ত রয়েছে এবং এটাই আপনার সাধারণ পয়েন্ট(point)।

কিন্তু এটা যদি অন্যভাবে বলা হয়, মানে আপনার কাছে এধরনের এলইডি(LED) আছে আর ক্যাথোড টার্মিনাল(terminal) গুলি সব একসাথে যুক্ত আছে এবং এটাই সাধারণ টার্মিনাল, আপনি তখন এটাকে সাধারণ ক্যাথোড ধরনের ডিসপ্লে(display) বলছেন। সুতরাং, কীভাবে তারা অভ্যন্তরীণভাবে সংযুক্ত রয়েছে তার উপর নির্ভর করে আপনার কাছে দুটি বিভিন্ন ধরণের ডিসপ্লে ইউনিট(display unit), সাধারণ অ্যানোড বা সাধারণ ক্যাথোড থাকতে পারে।

(প্লাইড সময় দেখুন : 04:47)



এখন লক্ষণীয় বিষয় হল, যখন আপনি একটা সেভেন সেগমেন্ট ডিসপ্লে ইউনিট(seven segment display unit) দেখছেন, তখন 5 থেকে 10 মিলি অ্যাম্পিয়ার বিদ্যুৎ এলইডি(LED) দিয়ে প্রবাহিত হওয়া দরকার, তবেই এটা সম্পূর্ণ তীব্রতায় জ্বলবে। এটা ন্যূনতম পরিমাণ বিদ্যুৎ যা প্রবাহিত করতেই হবে এবং আমরা সাধারণত কিছু বিদ্যুতের প্রতিরোধী রোধ ব্যবহার করি, যা একটি নির্দিষ্ট এলইডি(LED)তে মোট বিদ্যুত প্রবাহকে সীমাবদ্ধ করে। আরেকটি লক্ষণীয় বিষয় হলো, যখন এলইডি(LED) জ্বলছে, তখন এটি forward biased, আর এর দুই পাশের ভোল্টেজের প্রভেদ হল সাধারণত 1.2 ভোল্ট; কিছু এলইডির ক্ষেত্রে, এটা আরো বেশি হতে পারে।

যদি আমরা একটি নির্দিষ্ট এলইডি(LED)-র দিকে তাকাই, তাহলে আমরা বলতে চাই যে, আপনার একটি বিদ্যুৎ প্রতিরোধী রোধ এটার সাথে সিরিজে থাকা দরকার। এবং ধরন বলা যাক একদিকে আমি পাঁচ ভোল্ট, আর অন্যদিকে আমি গ্রাউন্ড(ground) এর সাথে যুক্ত করলাম, এবং আমি চাই একটি তড়িৎ প্রবাহ যেমন, 5 mA যেন এর মধ্য দিয়ে প্রবাহিত হয়। তখন এই রোধের মানটি খুব সহজেই গণনা করা যায়, এভাবে $(5V - 1.2V) / 5mA$. এটা আপনাকে রোধের মান বলে দেবে।

কিছু ক্ষেত্রে যখন আপনার সেভেন সেগমেন্ট ডিসপ্লে ইউনিট(seven segment display unit) থাকে, সেক্ষেত্রে তার মধ্যে আটটি এলইডি থাকে। সবচেয়ে ভালো হয় যদি একটি করে রোধ এই টার্মিনাল গুলির প্রতিটি তে থাকে, তাহলে এই সেগমেন্ট গুলোর প্রতিটিতে বিদ্যুৎ প্রবাহ 5 mA তে ঠিক করা যায়, কিন্তু পদ্ধতিটা সহজ করার জন্য, যেটা আপনি করতে পারেন তাহলে এই আটটি রোধ ব্যবহার করার পরিবর্তে একটি ছোট রোধ ব্যবহার করা, ধরন একে বলা হলো সাধারণ অ্যানোড, একে আপনি 5V এর সাথে যুক্ত করুন এবং যে সেগমেন্ট(segment) টা স্বালিয়ে দেখাতে চান সেটিকে গ্রাউন্ড এর সাথে যুক্ত করুন।

তখন এই রোধই নির্ধারণ করবে মোট বিদ্যুৎ কতটা প্রবাহিত হচ্ছে। সবচেয়ে থারাপ ক্ষেত্রে, যখন সব কটা, মানে এই আটটা সেগমেন্টই জ্বলছে, এই বিদ্যুৎ এই 8 টি ভাগে ভাগ হচ্ছে। যাতে সর্বমোট 40 mA বিদ্যুৎ কে আপনি এই রোধের মধ্যে দিয়ে প্রবাহিত করতে দেন, এবং সেই অনুসারে আপনার রোধের মান নেয়া উচিত। এভাবে আপনি রোধের মান কত হবে তা নির্বাচন করতে পারেন।
(প্লাইড সময় দেখুন: 07:49)

A Typical Interface

- Consider a common anode display unit.
- If a segment line is set to GND, it will glow.
- To display **0** we apply $A = B = C = D = E = F = GND$, and $G = Vcc$.
- To display **5** we apply, $A = C = D = F = G = GND$, and $B = E = Vcc$.
- To display **F** we apply $A = E = F = G = GND$, and $B = C = D = Vcc$.
- To display arbitrary characters, the seven segment lines can be driven from microcontroller port lines.
- For common cathode display, the convention will be just the reverse.
- GND for OFF, Vcc for ON.

swayam

যখন আপনার কাছে এইরকম একটি ডিসপ্লে ইউনিট(display unit) আছে তখন আমরা ধরে নিতে পারি যে আমার একটি সাধারণ অ্যানোড ডিসপ্লে আছে যেখানে সাধারণ টার্মিনাল(terminal) 5 V এর সাথে যুক্ত করা হয়েছে। সংশ্লিষ্ট বিভাগ টিকে আলোকিত করতে গেলে সেগমেন্ট পিন (segment pin) টি গ্রাউন্ডের সাথে সংযুক্ত থাকতে হবে।

ধরুন আমরা বলতে চাই যে আমি 0 (জিরো) অক্ষরটি দেখাতে চাই। তো 0 কি? 0 দেখতে এরকম লাগে। শুধুমাত্র G ছাড়া সমস্ত সেগমেন্টই জ্বলবে। শুধুমাত্র G সংযুক্ত থাকবে Vcc এর সাথে, এছাড়া আর সমস্ত সেগমেন্ট (segment) গ্রাউন্ড এর সাথে যুক্ত থাকবে। আমরা একইভাবে বলতে পারি, যখন আপনি 5 দেখাতে চাইবেন, 5 সংখ্যাটি অনেকটি এই রকম। তাই, B আর E সেগমেন্ট(segment) গুলি জ্বলবে না। সুতরাং B আর E কে আপনি Vcc এর সাথে যুক্ত করুন, অন্যদেরকে গ্রাউন্ড এর সাথে যুক্ত করুন।

একইভাবে, F সংখ্যাটির জন্য B,C,D জ্বলবে না। এইভাবে আপনি B,C,D কে Vcc এর সাথে যুক্ত করুন। যদি এটা সাধারণ ক্যাখোড হত তবে নিয়মটা উল্টো হত।

(লাইড সময় দেখুন: 09:55)

What is Liquid Crystal Display (LCD)?

- It is a type of display screen used in numerous applications.
- Very low power consumption as compared to LED.
- LCD units are very thin and is composed of several layers.
 - Two polarized panels, with a liquid crystal solution sandwiched between them.
 - Light is projected through the layers and is polarized as it passes, thereby producing the visual image.

The diagram illustrates the internal structure of an LCD unit, showing the following layers from left to right: Backplane, Polarizing Filter (P), Liquid Crystal (LC), Polarizing Filter (P), and a front glass. The backplane contains a Microcontroller (MCU) and various electronic components. The liquid crystal layer is the central component where light passes through polarized filters to create the visual image.

Swayam

সুতরাং এইভাবে আপনি একটি সেভেন সেগমেন্ট এলইডি ডিসপ্লে(seven segment LED display)কে মাইক্রোকন্ট্রোলার-এ পাঠাতে পারেন। আমি আপনাদেরকে ইন্টারফেসিং(interfacing) এর প্রাথমিক উপায়টি বলে দিয়েছি। এখন আমরা আসবো লিকুইড ক্রিস্টাল ডিসপ্লে ইউনিট(liquid crystal display unit), যা আজকাল খুবই জনপ্রিয়। অনেক ডিসপ্লে যন্ত্রে, আমরা এল সি ডি ডিসপ্লে ইউনিট দেখতে পাই। আমাদের মোবাইল ফোন, আমাদের ল্যাপটপ স্ক্রিন, টাচ স্ক্রিন এবং যেকোন জায়গায় নিয়ন্ত্রণ সরঞ্জাম থেকে শুরু করে অসংখ্য অ্যাপ্লিকেশনগুলিতে এলসিডি ডিসপ্লে ইউনিট(LCD display unit)গুলি ডিসপ্লে(display) স্ক্রিন হিসাবে ব্যবহৃত হয়।

আগেকার দিনের কিছু এলসিডি(LCD) স্ক্রিন রং দেখাতে পারতো না; সে গুলোকে বলা যায় একরঙা(Monochrome) স্ক্রিন, তারা শুধুমাত্র সাদা আর কালোই দেখাতে পারে। এই এল সি ডি স্ক্রিনের মুখ্য সুবিধা হল যে এল ই ডি(LED) র তুলনায় তারা খুবই কম পাওয়ারে চলে। আমি আপনাদের বলেছিলাম প্রতিটা এলইডি কে জ্বালাতে গেলে, 5 mA করে বিদ্যুৎ দরকার, কিন্তু এল সি ডির ক্ষেত্রে বিদ্যুতের চাহিদা হয় মাইক্রো অ্যাম্পিয়ার এ। আর, যেভাবে এলসিডি ইউনিট গুলো তৈরি হয়, তা তে সেগুলি ফর্ম ফ্যাক্টরের(Form factor) দিক থেকেও খুবই পাতলা।

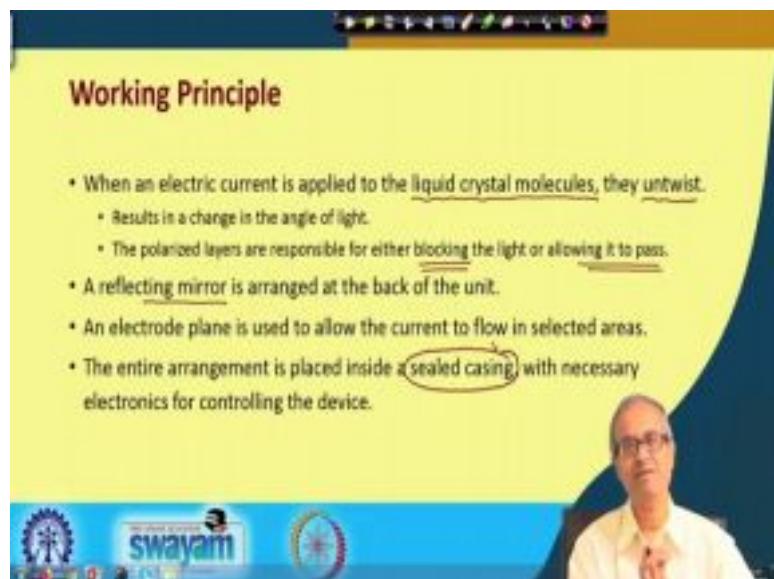
এলসিডি(LCD) কিভাবে তৈরি হয়, আমি তার বিস্তারিত বিবরণে যাচ্ছি না। এটুকুই শুধু বলতে চাই যে, এলসিডির মধ্যে কিছু সংখ্যক আস্তরণ থাকে, যেগুলো একসাথে চেপে আটকানো থাকে। কেন্দ্রীয় পদার্থটি একটি স্তর, যেটা এখানে মাঝখানে দেখানো হয়েছে, এটা একটি তরল ক্রষ্টাল(crystal) স্তর। এটিতে একটি বিশেষ উপাদান রয়েছে যাকে বলে তরল ক্রষ্টাল; প্রয়োগকৃত ভোল্টেজের উপর নির্ভর করে এটি হয় আলোকে এর মধ্য দিয়ে যেতে দেয় অথবা এটি আলোককে বাধা দেয়। এটি সাধারণত সমবর্তিত(polarized) হয়, এবং সেখানে কিছু কাঁচ উভয় পাশেই থাকে, এপাশে একটা কাঁচের স্তর, ওপাশে আর একটা কাঁচের স্তর, আর তরল ক্রিস্টাল এদের মধ্যখানে চেপে আটকানো থাকে। এখন আসুন আমরা এটাকে এলসিডির 7 সেগমেন্ট ডিসপ্লে(7 segment display of LCD) হিসেবে ধরি, সেখানে একটা ইলেকট্রোড (electrode) থাকে, যা এখানে একটি 7 সেগমেন্ট ডিসপ্লের আকারের হবে, এই ভিন্ন ভিন্ন ইলেকট্রোড(electrode)গুলোতে বিদ্যুৎ প্রবাহিত করলে তারা সক্রিয় হবে।

প্রয়োগ-কৃত তড়িৎ প্রবাহের উপর নির্ভর করে, আলোর কিছুটা মেরু করণ(polarization) হবে যেটা এর

মধ্যে দিয়ে প্রবাহিত হবে এবং এই মেরুকরণের উপর নির্ভর করে কিছুটা আলো আটকানো হবে আর কিছুটা আলো এর মধ্যে দিয়ে চলে যাবে; পিছনে একটি আয়না রয়েছে, যা আলো কে প্রতিবিস্থিত করে যাতে এটি দৃশ্যমান হয়। এছাড়া ভালো তীব্রতার জন্য প্রায়ই একটি এলইডি ব্যাকপ্লেন(backplane) থাকে। যেখানে পিছনের দিকটিতে এক ধরণের উজ্জ্বলতা থাকে যাতে ডিসপ্লে যান্টি(vehicle) খুব স্পষ্ট হয়ে যায়।

সুতরাং এখানে আমি এটি খুব সংক্ষেপে উল্লেখ করেছি। এটি বেশ কয়েকটি স্তর নিয়ে গঠিত, সেখানে 2 টি মেরুকৃত(polarized) প্যানেল রয়েছে যার মধ্যে তরল ক্রিস্টাল চাপে আটকানো আছে, যেমন টা আমি দেখিয়েছি। এবং আলো প্রজেক্ট করা হলে, যখন আলো একটি রঙিন এলসিডির স্তরগুলির মধ্য দিয়ে প্রবাহিত হয়, তখন তারা কোনও উপায়ে মেরুকৃত হয়, যা রঙগুলি উপস্থাপন করে। এটি একটি অত্যন্ত পরিশীলিত প্রকৌশল (engineering) যাতে এটি করা হয়, যাতে আলো নির্দিষ্ট রঙের বৈশিষ্ট্য নিয়ে আসে।

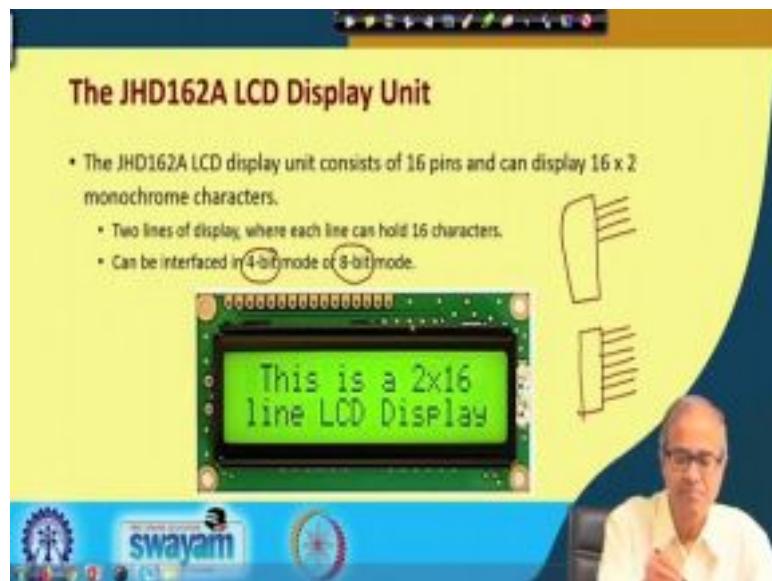
(ম্লাইড সময় দেখনুন: 14:05)



যখন কিছুটা বিদ্যুৎ কোন একটি ইলেকট্রোড(electrode) এর মধ্যে দিয়ে প্রবাহিত হয়, এই বিদ্যুৎ তরল ক্রিস্টাল অনুগুলিকে প্রভাবিত করে। তারা তাদের অভিমুখী করণ(orientation) পরিবর্তন করে এবং তাদের এই অভিমুখী করণ(orientation) পরিবর্তনের উপর নির্ভর করে যে, মেরুকৃত(polarized) আলো তাদের ভেতর দিয়ে যাবে কি যাবে না। এটি আলোর কৌণিক পরিবর্তন করে। সুতরাং আপনি হয় এই আলোর রেখা কে একটি নির্দিষ্ট সেগমেন্ট(segment)-র মধ্য দিয়ে যেতে দেবেন অথবা এটিকে আটকে দেবেন। এভাবে একটি সেগমেন্ট(segment) হয় জ্বলবে অথবা জ্বলবে না।

সুতরাং, আপনি সরাসরি তরল ক্রিস্টালের মধ্যে দিয়ে বিদ্যুৎ পাঠাচ্ছেন না, এটি পরোক্ষভাবে ঘটছে এবং আমি যেমন বলেছিলাম পিছনে আছে একটি প্রতিবিষ্প সৃষ্টিকারী আয়না ও ইলেকট্রোড তল (electrode plane) আর পুরো ব্যবস্থাটি একটি সিল করা আবরণের ভিতরে স্থাপন করা হয়। আপনি যখন একটি এলসিডি ডিসপ্লে ইউনিট দেখেন, তখন আপনি পুরোটাই একটা সুন্দর আবরণের ভিতরে দেখেন, খুবই পাতলা এই স্তর গুলি তৈরি করা হয় এবং সেই মসৃণ ফ্রেমের ভিতরে রাখা হয়।

(স্লাইড সময় দেখুন: 15:19)

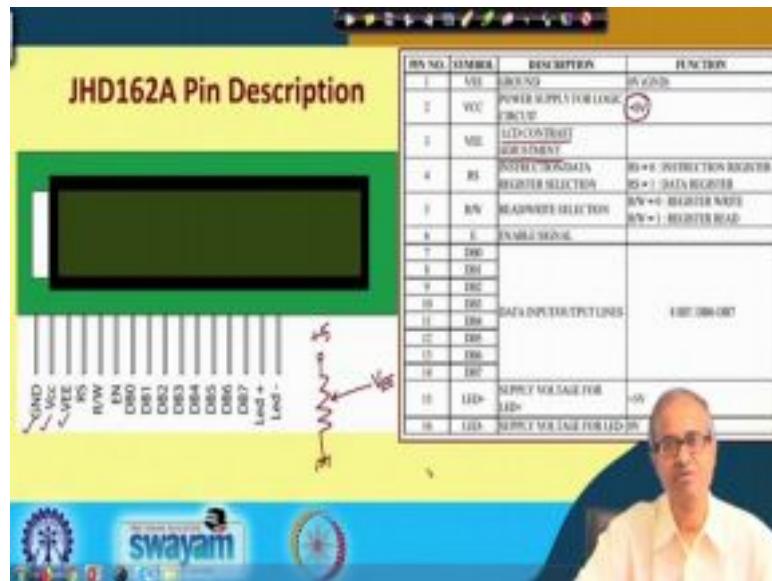


আসুন আমরা একটি নির্দিষ্ট এলসিডি(LCD) যন্ত্র কে বিবেচনা করি যা আমাদের প্রদর্শনের অংশ হিসেবে দেখানো হবে, এটি একটি খুবই সুবিধাজনক আড়তপুট যন্ত্র, যা আলফানিউমেরিক(alphanumeric) অক্ষর দেখাতে ব্যবহার করা যেতে পারে। আলফানিউমেরিক মানে কিন্তু শুধুই নাম্বার নয়, সঙ্গে বর্ণমালার অন্যান্য অক্ষর যেমন, এ বি সি ডি ই এফ জি এইচ সবই। পাট নম্বরটি হল JHD162A আর এই হল এটার ছবি, দেখতে পাচ্ছেন, একদিকে কিছু কানেক্টর(connector) পিন আছে, সর্ব মোট ষোলটা পিন দেওয়া আছে এবং ডিসপ্লে ইউনিট(display unit) টা দুই সারি(line) অক্ষর দেখাতে পারে, যেগুলোতে প্রতিটায় ষোলটি করে অক্ষর আছে।

এটি একটি 16×2 ডিসপ্লে ইউনিট, আর এতে দুধরনের মোড আছে. 4 বিট মোড-এ আপনি মাইক্রোকন্ট্রোলারের সাথে কেবল মাত্র 4 টি ডাটা লাইন(data lines) সংযুক্ত করবেন, এবং 8 বিট মোড-এ আপনি মাইক্রোকন্ট্রোলার এর সাথে 8 টি ডাটা লাইন সংযুক্ত করবেন। এখানে 16 টি পিন এর মধ্যে, 8 টি হল ডাটা লাইন. সুতরাং, আপনি যদি ওদের মধ্যে আট টি কে মাইক্রোকন্ট্রোলার এর সাথে যুক্ত করেন, তাহলে ডাটা পাঠানো সহজ হয়ে যাবে, কারণ সমস্ত অক্ষর সাধারণত ASCII এর মতো কিছু অক্ষর কোডে এনকোড(encode) করা থাকে, সেগুলি 7 বিট বা 8 বিট হয়।

ফ্লিপ(flip) দিকে আপনাকে আটটি তার সংযুক্ত করতে হবে. সেজন্য, তার-এর সংখ্যাও বেশি কারণ এই যন্ত্রটি 4 বিট মোড(mode)-এর বিকল্প সরবরাহ করে, যেখানে আপনি শুধুমাত্র 4 টি বিট ডাটা লাইন(data lines) মাইক্রোকন্ট্রোলার এর সাথে যুক্ত করতে পারেন, এবং মাইক্রোকন্ট্রোলারটি দুটি পর্যায়ে 8 বিট ডাটা পাঠিয়ে থাকে, প্রথমে 4 বিট আর তারপর আর 4 টি বিট। যদি আপনি এমন ড্রাইভার(driver) ব্যবহার করেন, যেটা 4 বিট মোড কে সাহায্য করে, তাহলে আপনার চিন্তিত হবার কোন দরকার নেই, আপনি শুধুমাত্র 4 টি লাইনের সাথে যুক্ত করুন এবং ড্রাইভার(driver) টি অবশিষ্ট কাজ করে দেবে।

(স্লাইড সময় দেখুন: 17:53)



এখন আমরা পিন গুলোর বিবরণ দেখি; এখানে 16টা পিন আছে. ডানদিকে পিনগুলোর সংক্ষিপ্ত বিবরণ উল্লেখ করা আছে। প্রথম দুটো হল গ্রাউন্ড(ground) এবং পাওয়ার সাপ্লাই, সাধারণত গ্রাউন্ড টি 0V এর সাথে আর পাওয়ার সাপ্লাই 5V এর সাথে যুক্ত থাকে। তৃতীয় ইনপুট হল VEE, এটি আপনি, ভোল্টেজ প্রয়োগ করে ডিসপ্লে তে বৈপরীত্যের(contrast) মাত্রার সামঞ্জস্য আনতে পারেন যাতে এটা, বৈপরীত্য সমন্বয়ের(contrast adjustment) কাজে ব্যবহার করতে পারেন।

আপনি এইভাবে সংযোগ করুন, আপনি একটি পরিবর্তনশীল রোধ, যাকে পোটেনশিওমিটার বলে, ব্যবহার করুন। একদিকে আপনি 5V সংযুক্ত করুন, আর অন্য দিক গ্রাউন্ড এর সাথে যোগ করুন, আর এটাই পোটেনশিওমিটার, এই ট্যাপিং(tapping) পয়েন্টটা একটা নব ঘূরিয়ে বা যে স্ক্রু(screw) টি আছে সেই স্ক্রুটিকে ঘূরিয়ে পরিবর্তন করা যেতে পারে এবং মাঝের পয়েন্টটা VEE এর সাথে যুক্ত করুন। উপযুক্তভাবে পোটেনশিওমিটারটির সামঞ্জস্য এনে, আপনি 0 হতে 5V এর মধ্যে যে কোনো ভোল্টেজ VEE তে প্রয়োগ করতে পারেন, আপনার দেখার কোণের উপর নির্ভর করে আপনি সেই অনুযায়ী বৈসাদৃশ্যটির সামঞ্জস্য করতে পারেন।

ওখানে একটা পিন আছে, যাকে বলে RS, এটি রেজিস্টার সিলেক্ট(register select) বোায়। আপনি যখন ডিসপ্লে ইউনিট এ কিছু ডেটা বা নির্দেশ পাঠান, তখন এটা হয় সেই ডেটা যা আপনি দেখতে চান, অথবা কিছু ক্ষেত্রে এটা শুধু ডেটা নয়, আপনি ডিসপ্লে ইউনিটকে বলছেন যে দেখো, আমরা অক্ষর গুলিকে 8 বিট মোড(mode) বা 16 বিট মোড এ দেখাতে চাই। এটি নির্দেশ বা ডেটা হতে পারে; এই RS আসলে আপনাকে জানায় আপনি আসলে কী লিখছেন।

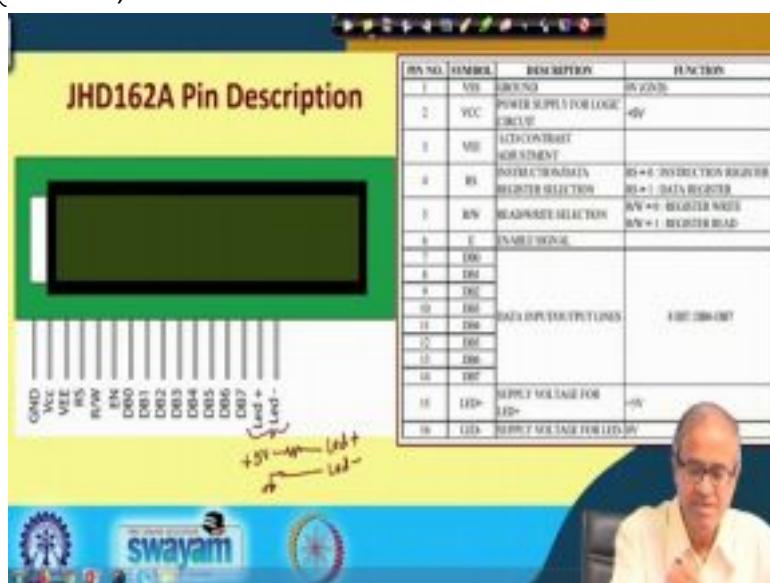
যখন আপনি মাইক্রোকন্ট্রোলারের সাথে সংযোগ করছেন, কিছু ক্ষেত্রে মাইক্রোকন্ট্রোলার নির্দেশ পাঠাবে; আবার কিছু ক্ষেত্রে এটা ডেটা পাঠাবে। এই RS পিন কে মাইক্রোকন্ট্রোলারের কিছু ডেটা লাইন(data lines) এর সাথে যুক্ত করতে হবে, যাতে এটাকে প্রয়োজনমতো 0 বা 1 এর সাথে যোগ করা যায়। এর পরের পিন টি হলো read/write পিন, এটা বলে দেয় যে আপনি এলসিডি ইউনিটে লিখছেন, না, এলসিডি ইউনিট থেকে

কোন অবস্থা সংক্রান্ত তথ্য(status information) পড়ছেন। যদি এটা 0 হয়, এটার মানে write, আর যদি এটা 1 হয়, এটার মানে হবে read।

আমাদের এই ইন্টারফেসিং(Interfacing) প্রয়োগে(application), আমরা শুধুই লিখব, আমরা অবস্থা সংক্রান্ত কোন তথ্য পড়বো না। আমরা এটাকে সরাসরি গ্রাউন্ড এর সাথে যুক্ত করতে পারি, আপনি এটাকে 0 রাখতে পারেন, যাতে এটা সবসময় রেজিস্টার write মোড এ থাকে। পরের পিন টি হলো একটি enable signal, আপনি দেখবেন এই enable signal ডিসপ্লে ইউনিটকে সক্রিয় করবে। যখনই এটাকে সক্রিয় করবেন তখনই এটা কাজ করবে। সাধারণত এই enable পিন টি মাইক্রোকন্ট্রোলার এর সাথে যুক্ত থাকে, যাতে যখনই এটা ডেটা পাঠাতে চাইবে, এটা একে সক্রিয় করবে এবং তখন এটা ডেটা টি লিখবে।

যদি এটা 1 থাকে, তখন এটা সক্রিয় হবে, যদি এটা 0 হয়, তখন এটা নিষ্ক্রিয় থাকবে। এর পর আসছে আটটি ডেটা লাইন DB0 থেকে DB7। যখন আপনি এই 8 বিট মোডে ইন্টারফেসিং করছেন, তখন আপনি মাইক্রোকন্ট্রোলারের পোর্ট এর 8 টা লাইন এই 8টা পিনের সাথে যুক্ত করবেন, কিন্তু যখন আপনি 4 বিট মোড ব্যবহার করবেন তখন শুধু মাত্র চারটি লাইন যোগ করতে হবে DB4, DB5, DB6 এবং DB7, আপনি কেবল উপরের ক্রমের চারটি লাইন মাইক্রোকন্ট্রোলার এর সাথে যুক্ত করবেন; এবং আমি যেমন বলেছি মাইক্রোকন্ট্রোলার একবারে 2 টি চক্র(cycle), 4 বিটে ডেটা আউটপুট(output) দেয়।

(স্লাইড সময় দেখনুন: 22:33)

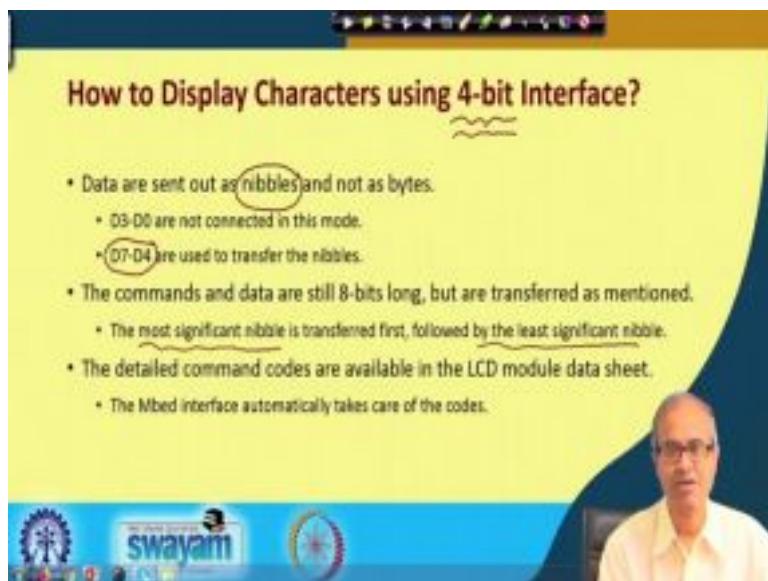


আর শেষ দুটি পিন হলো এলইডি+ (LED+) এবং এলইডি-, এগুলো উজ্জ্বলতার মাত্রা নিয়ন্ত্রণ করে। আপনি এলইডি+ এবং এলইডি- এর মাঝে একটি ভোল্টেজ যুক্ত করতে পারেন যাতে ডিসপ্লেটি উজ্জ্বল হয়, সাধারণত আপনি 5V নিয়ে কি করেন, আপনি একটি রোধ যুক্ত করতে পারেন, এটি আপনি এলইডি+ ইনপুটে যুক্ত করতে পারেন এবং আরেকটি ইনপুট আপনি এলইডি- এ যুক্ত করতে পারেন, এটা আপনি গ্রাউন্ড করতে পারেন, এভাবে আপনি সংযুক্ত করতে পারেন।

এটি আপনাকে বিভিন্ন পিনগুলি এবং ঠিক কীভাবে আপনি সেগুলি সংযুক্ত করতে পারবেন সে সম্পর্কে

আপনাকে জানায়।

(ম্লাইড সময় দেখন: 23:31)

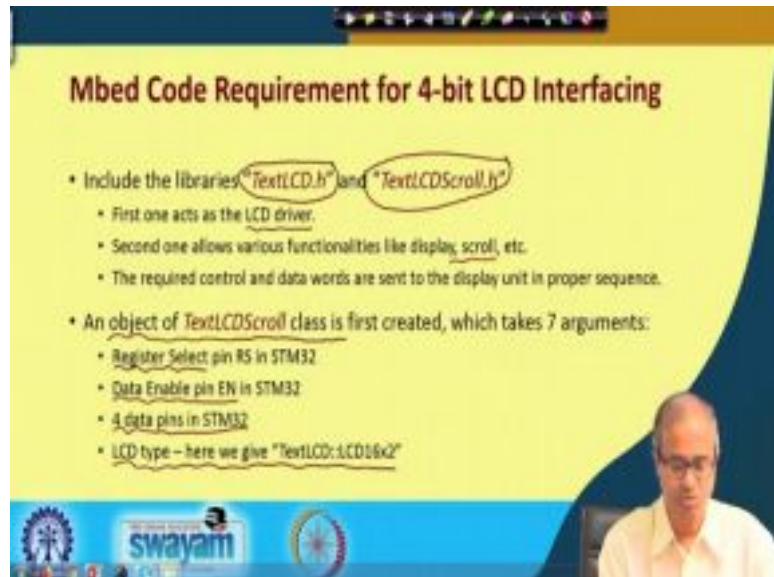


এখন আমরা ইন্টারফেসিং(interfacing) এর ব্যাপারে বলব; আমাদের পরীক্ষায় আমরা সহজ করার জন্য 4 বিট মোড ব্যবহার করব, কারণ শুধু 4 টি ডেটা লাইন(data lines) যুক্ত করতে হবে। এখন, আমি আপনাদের বলেছি 4 বিট মোড, ডেটা ডিসপ্লে যন্ত্রে একসাথে 4 বিট করে লেখা হয়, 4 বিট কে একসাথে নিবল(nibble) বলে। সুতরাং ডেটাকে নিবল(nibble) হিসেবে পাঠানো হয়।

DB7 থেকে DB4 মাইক্রোকন্ট্রোলার এর সাথে যুক্ত করা হয়, কিন্তু কমান্ডগুলি(commands) এবং ডেটা, যেগুলোকে লিখতে হবে, সেগুলো প্রকৃতপক্ষে 8 বিট দীর্ঘ, আর তাই তাদেরকে দুইটি চক্র(cycles) লেখা হয়। প্রথমে উচ্চতর অর্ডার এর 4 টি বিট পাঠানো হয়, পরে নিম্নতর অর্ডারের 4 টি বিট পাঠানো হয়। STM32 বোর্ড দিয়ে, আমরা একটি প্রোগ্রামিং ব্যবস্থাপনা ব্যবহার করব যাকে বলে এম্বেড(mbed)।

এম্বেড (mbed) প্রোগ্রামিং ব্যবস্থাপনাটি C তে আমাদের নিজস্ব অ্যাপ্লিকেশন তৈরিতে সাহায্য করে, আর এটা অনেক লাইব্রেরী ফাংশন ও ড্রাইভার দিয়ে দেয়, যেটা আমরা ব্যবহার করতে পারি। সেই লাইব্রেরী ফাংশনে, এই 4 বিট ইন্টারফেসিংয়ের যন্ত্র নেওয়া হবে এবং প্রোগ্রাম লেখার সময়, আমাদের কিছু নিয়ে উদ্বিগ্ন হতে হবে না, ড্রাইভার স্বয়ংক্রিয়ভাবেই এই সবের যন্ত্র নেবে; তার মানে, প্রথমে উচ্চতর অর্ডারের ফোর বিট লেখা, এরপর তার নিচের 4 বিট লেখা এবং তারপর তার পরেরটা এভাবে, যখন নির্দেশ লিখতে হবে, কখন সেটা লিখতে হবে সেটা ডাইভার(driver) ই ঠিক করে নেবে।

(ম্লাইড সময় দেখন: 25:31)

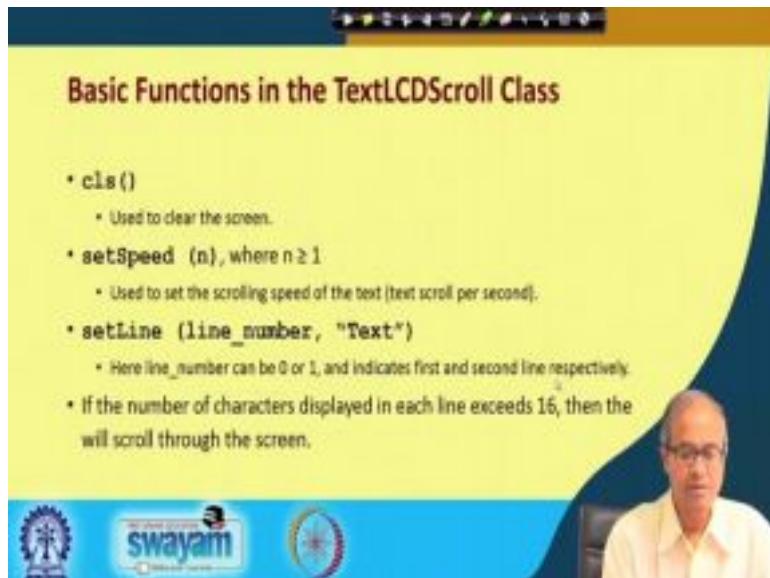


আমরা দুটো লাইব্রেরী ব্যবহার করব `TextLCD` and `TextLCDScroll`. প্রথম লাইব্রেরীটা হলো প্রকৃতপক্ষে `LCD driver`, এটি নিম্নতর লেভেলের ডেটা পার্থক্যে এবং প্রতিবার ফোর বিট করে যে নির্দেশ দেওয়া হয়, তার দায়িত্বে থাকে, দ্বিতীয়টি, আপনাকে ডিসপ্লেতে স্ক্রোলিং (scrolling) বৈশিষ্ট্যগুলি দিয়ে দেয়, আপনি যদি একটি সারিতে 16 টিরও বেশি অক্ষর দেখানোর চেষ্টা করেন, তবে ডিসপ্লেটি স্বয়ংক্রিয়ভাবে বাম থেকে ডানে স্ক্রোল(scroll) করবে।

আমি একটা উদাহরণ দেখাবো; আমরা প্রথমে `TextLCDScroll` class এর অবজেক্ট তৈরি করব, এটা এমন একটা অবজেক্ট তৈরি করবে যেটা এলসিডি ইউনিটকে চিহ্নিত করবে, এটাই হলো অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর ধারণা. যথন আপনি অবজেক্টের(object) ব্যাপারে বলবেন, সাতটি জিনিস আপনাকে বলতে হবে, কারণ আপনি দেখতে পাচ্ছেন এই এলসিডি ইউনিট(LCD unit) দিয়ে আপনাকে মাইক্রোকন্ট্রোলার এর সাথে কিছু সংযোগ করতে হবে, সংযুক্ত করার জন্য এই সাতটি জিনিস দরকার. এই জিনিসগুলি কি?

প্রথমে আপনাকে রেজিস্টার সিলেক্ট ইনপুট(register select input) সংযুক্ত করতে হবে। আমি বলেছিলাম আপনাকেই এটা নির্দেশ না ডেটা সেটা নির্বাচন করতে হবে। এরপর ডেটা enable, আপনাকে ডিসপ্লে ইউনিট টি সক্রিয় করতে হবে। তারপর চারটি ডেটা লাইন, এই নিয়ে ছাটা হল, অবশ্যে এখানে আসবে কি ধরনের এলসিডি, আমাদের বলতে হবে কি ধরনের এলসিডি ডিসপ্লে ইউনিট(LCD Display unit) সেখানে আছে। আপনি দেখবেন সেখানে অন্যান্য ডিসপ্লে ইউনিটও আছে, যেটা হলো 16 বাই 4, সেখানে চারটে লাইন আছে, কিন্তু যে ডিসপ্লে ইউনিট টা আমরা এখানে ইন্টারফেসিং(interfacing) করছি, তাতে শুধু দুটি লাইন আছে।

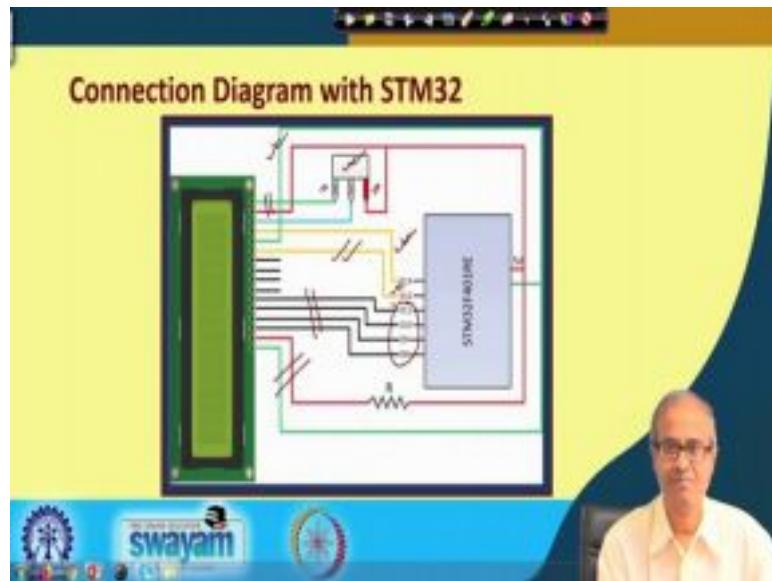
(স্লাইড সময় দেখুন: 27:47)



এই TextLCDScroll ক্লাস যেসব ফাংশনসকে(functions) সহায়তা করে সেগুলো দেখানো হয়েছে। `cls` এর মানে হলো স্ক্রিন পরিষ্কার করা, এটাকে ফাঁকা করে দেয়া হয়। `setSpeed(n)`, যেখানে `n` হল অখণ্ড সংখ্যা, যার মান একের সমান বা বেশি, মানে আপনি যখন স্ক্রিনে স্ক্রোলিং(scrolling) করছেন, এই স্ক্রোলিং এর গতি কত? এক মানে প্রতি সেকেন্ডে একটি করে অক্ষর, দুই মানে হবে প্রতি সেকেন্ডে দুইটি করে অক্ষর, তিন মানে প্রতি সেকেন্ডে প্রথম তিনটি অক্ষর, এভাবে আপনি স্ক্রোলিং(scrolling) এর স্পিড নিয়ন্ত্রণ করতে পারবেন।

`setLine` হল আরেকটি ফাংশন, যাতে আপনি বলতে পারেন কোন পাঠ্য আপনি দেখাতে চান আর কোন লাইনে আপনি দেখাতে চান। লাইন নাম্বার শূন্য বা এক হতে পারে, শূন্য মানে প্রথম লাইন, এক মানে দ্বিতীয় লাইন। এবং আমি আগে যেমন বলেছি, যদি যে পাঠ্যটি আপনি এক লাইনে ডিসপ্লে করতে চাহিলেন সেটি যোল অক্ষরের থেকে বড় হয়, তাহলে ওই পাঠ্য স্বয়ংক্রিয়ভাবেই স্ক্রোল করতে আরম্ভ হয়ে যাবে।

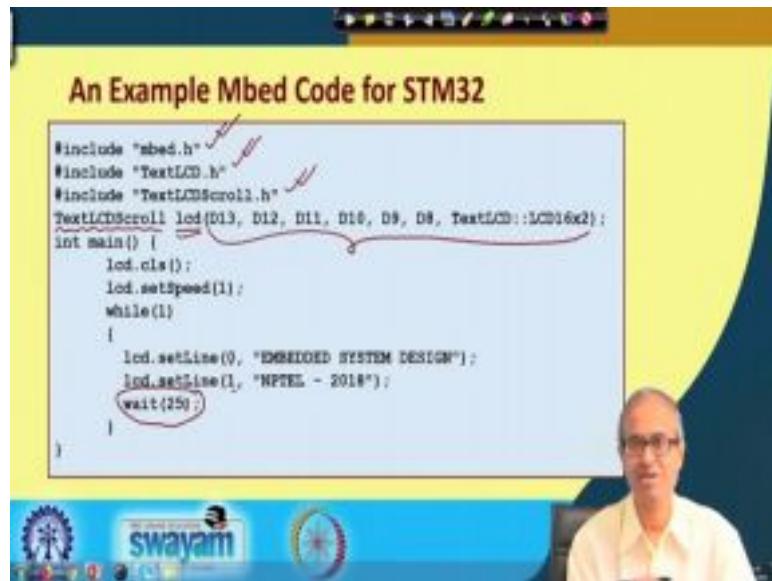
(স্লাইড সময় দেখুন: 28:53)



এখানে আমি STM32 মাইক্রোকন্ট্রোলার বোর্ড দিয়েএকটি খুব সাধারণ ইন্টারফেস(interface) দেখাচ্ছি। আপনি দেখুন এখানে আমি LED+ এবং LED- কানেকশন গুলি যুক্ত করছি। এখানে প্রথম দুটি হল পাওয়ার সাপ্লাই এবং গ্রাউন্ড। আপনি এখানে দেখুন, আমি একদিকে পোটেনশিওমিটার ব্যবহার করছি, অন্যদিক টা গ্রাউন্ড-এর সাথে যুক্ত করছি, আর মধ্যখালে 5 volt যুক্ত করছি, আমি VEE টার্মিনাল এর সাথে যুক্ত করছি বৈপরীত্য নিয়ন্ত্রণের জন্য।

তখন আমরা পাই রেজিস্টার সিলেক্ট(registers select) এই হলুদ লাইনটা, যেটা মাইক্রোকন্ট্রোলারের কোন একটি পিনের সাথে যুক্ত, এই সময় read/write গ্রাউন্ড-এর সাথে স্থায়ীভাবে যুক্ত, কারণ আমরা শুধুই লিখছি(writing), তখন এখানে আমাদের কাছে enable আছে। Enable-টি মাইক্রোকন্ট্রোলারের একটি লাইনের সাথেও যুক্ত, যেহেতু মাইক্রোকন্ট্রোলারকে এটি সক্রিয় করতেই হবে। এরপর উচ্চতর অর্ডার-এর 4 টি লাইন মাইক্রোকন্ট্রোলারের পিনগুলোর সাথে যুক্ত থাকে। এইভাবে মাইক্রোকন্ট্রোলারের সাথে ডিসপ্লে ইউনিট এর সংযোগ করতে হয়।

(ম্লাইড সময় দেখুন: 30:11)



আর আমি একটা নমুনা mbed কোড কিছু অক্ষর প্রদর্শন করার জন্য দেখাচ্ছি. আপনি দেখুন আপনাকে প্রথমে কিছু header অন্তর্ভুক্ত করতে হবে, কারণ আপনি bed প্রোগ্রামিং ব্যবহার করছেন, তাই mbed.h হল বাধ্যতামূলক। আপনাকে এটা অন্তর্ভুক্ত করতেই হবে এবং এছাড়া দুটি অতিরিক্ত লাইব্রেরী আছে, আপনি ব্যবহার করছেন TextLCD এবং TextLCDScroll। আমি যেমন বলেছিলাম, আপনাকে TextLCDScroll এর একটি object তৈরি করতে হবে, আপনি এর নাম দিন এলসিডি, আর এই হল সাত টা জিনিস যা আপনাকে নির্দিষ্ট করে দিতে হবে।

এখানে আপনি নির্দিষ্ট করে বলুন কোন পিন ওলো আপনি যুক্ত করছেন. প্রথমটি হলো রেজিস্টার সিলেক্ট(register select); যার মানে আপনি রেজিস্টার সিলেক্ট(register select) কে STM32 D13 পিন-এর সাথে যুক্ত করছেন। তার পরেরটা হল enable, এটি আপনি STM32 D12 পিন-এর সাথে যুক্ত করছেন। এরপর হলো চারটি ডেটা লাইন, যেগুলো আপনি D11, D10, D9, D8 এর সাথে যুক্ত করছেন আর অবশিষ্ট টি বলে দেবে এইটা কি ধরনের ডিসপ্লে ইউনিট(display unit)? এবং মূল প্রোগ্রামে, আপনি দেখবেন এটা খুবই সোজা, আমরা এটাকে প্রথম cls ফাংশন(function) বলছি।

আমাদেরকে এই অবজেক্টের(object) নাম দিতে হবে, এলসিডি ডট ফাংশনের(LCD dot function) নাম, এটা এভাবেই ডাকা হয়. সুতরাং, ডিসপ্লেটি ফাঁকা করা থাকে তারপর আমি স্ক্রোলিং(scrolling) এর গতি প্রতি সেকেন্ডে একটি করে অক্ষরে সেট(set) করছি. এরপর একটি লুপ(loop)-এ, যেখানে এক(1) মানে একটি অনন্ত(indefinite) লুপ. আমরা প্রথম লাইনে কিছু দেখাচ্ছি, দ্বিতীয় লাইনে কিছু আমরা এমবেডেড সিস্টেম ডিজাইন (EMBEDDED SYSTEM DESIGN) প্রদর্শন করেছি, যেটা কি না 16 অক্ষরের থেকে বড়। সুতরাং স্ক্রোলিং এখানে আস্তে হবে, প্রতি সেকেন্ডে একটি করে অক্ষর, দ্বিতীয় লাইন NPTEL-2018, এটি স্থির, 16 অক্ষরের কম, তাই স্ক্রোলিং(scrolling) হবে না।

আর এখানে, এটির প্রকৃত কোন প্রয়োজন নেই, আমি 25 সেকেন্ড অপেক্ষা করছি. এইভাবে আপনি সংযোগ করতে পারেন, আর এলসিডি তে STM32 মাইক্রোকন্ট্রোলার বোর্ডের সাথে সংযুক্ত করে প্রোগ্রাম করতে পারেন. এটা কিভাবে করতে হয়, এই ধারণা আপনাকে দেবে। এর সাথেই আমরা এই বক্তৃতাটির শেষে

এসে পড়েছি। আমরা অন্য কিছু সেনসর্স(sensors), একচুয়াটরস(actuators), এবং আউটপুট যন্ত্রের ব্যাপারে পরবর্তী বক্তৃতা গুলিতে কথা বার্তা চালিয়ে যাব।

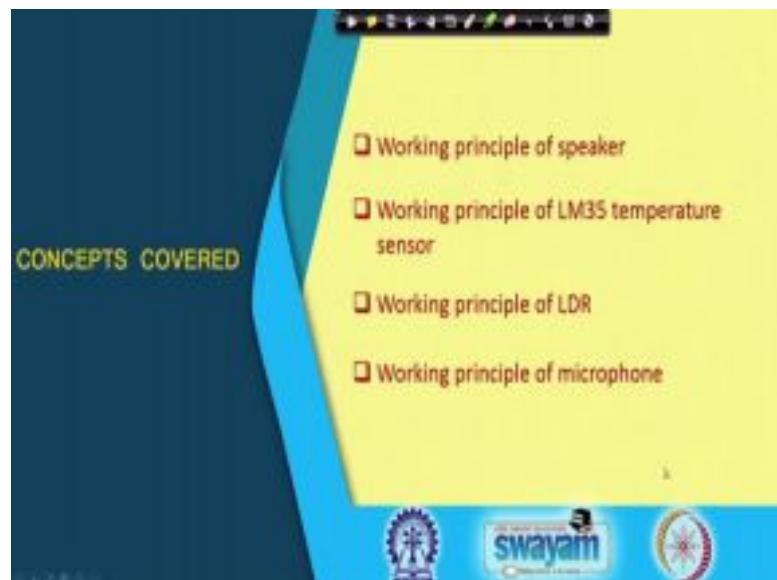
ধন্যবাদ।

Embedded System Design With ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 16
Output Devices, Sensors and Actuators (Part II)

এই বক্তৃতায় আমরা কিছু sensor নিয়ে কথা বলব যেগুলি আমরা interfacing এবং কিছু embedded system applications তৈরিতে ব্যবহার করতে পারি।

(Refer Slide Time: 00:41)



এখানে আমরা সংক্ষেপে কিছু sensor নিয়ে কথা বলব যেমন speaker, temperature sensor, LM35, light dependent resistor, microphone যেগুলো আমরা প্রদর্শনের অংশ হিসেবে দেখাবো। আমরা বিস্তারিত বর্ণনায় ঘাব না, কিন্তু খুব সংক্ষেপে কার্যপ্রণালী বলবো ঘাতে আপনি কিভাবে জিনিসটি কাজ করছে তা বুঝতে পারবেন।

প্রথমে আমরা speaker এর ব্যাপারে বলব।

(Refer Slide Time: 01:13)

What is a Speaker?

- It is an electro-acoustic transducer, which converts an electrical audio signal into a corresponding sound.
- It consists of the following:
 - A permanent magnet that is fixed.
 - A movable electromagnet attached to a diaphragm.
- How it works?
 - When a current flows through the coil of the electromagnet, a magnetic field is produced in the direction determined by the polarity of the input signal.
 - The two magnets either attract or repel.

Swayam

Speaker এমন একটি যন্ত্র যা দিয়ে কিছু শব্দ উৎপন্ন করা যায়।

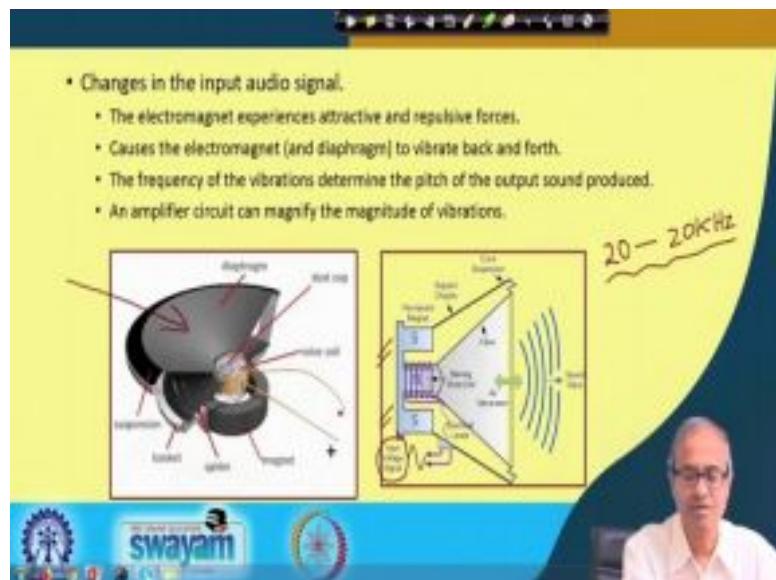
Speaker হল একটি **electro-acoustic transducer**, যার মানে এটা একটা বৈদ্যুতিক শব্দের signalকে সংশ্লিষ্ট শব্দে রূপান্তরিত করে; তার মানে আপনি একটি বৈদ্যুতিক তরঙ্গাকার signal ইনপুট হিসেবে দেন এবং speaker টি একটি শব্দ আউটপুট হিসেবে উৎপন্ন করে। এখন, speaker এর ভিতরে কি আছে? speaker-এর ভেতরে একটি স্থায়ী চুম্বক এবং একটি অস্থাবর বৈদ্যুতিন চুম্বক(electromagnet) আছে।

একটি বৈদ্যুতিন চুম্বক(electromagnet) কি? বৈদ্যুতিন চুম্বক(electromagnet) কিছু ferromagnetic উপাদান, যার চারদিকে একটি কুণ্ডলী(coil) থাকে, তাই দিয়ে তৈরি হয়; যখনই এই কুণ্ডলী টির ভেতর দিয়ে বিদ্যুৎ প্রবাহিত হয় তখন ওই উপাদানটি চুম্বক এ পরিণত হয়। ধারণাটি এই রকম। একটি স্থায়ী চুম্বক থাকে এবং ferromagnet-এর চারিপাশের কুণ্ডলীটি অস্থায়ী, যখনই এর মধ্যে দিয়ে বিদ্যুৎ প্রবাহিত হবে তখনই এটা চুম্বক হয়ে যাবে। সুতরাং, কি ধরনের বিদ্যুৎ আমরা প্রবাহিত করছি তার উপর নির্ভর করে, হয় এটা স্থায়ী চুম্বকটি দ্বারা আকর্ষিত হবে অথবা স্থায়ী চুম্বকটি থেকে বিকর্ষিত হবে, এবং সেখানে একটি কম্পন হবে এটাই প্রাথমিক ধারণা।

যখন বৈদ্যুতিন চুম্বকের কুণ্ডলীর ভেতর দিয়ে বিদ্যুৎ প্রবাহিত হয় তখন উৎপন্ন চৌম্বক

ক্ষেত্রটির(magnetic field) জন্য এই উপাদানটি চুম্বক-এ পরিণত হয় এবং চুম্বক দুটি একে অপরকে হয় আকর্ষণ করে অথবা একটি বিকর্ষণ শক্তির জন্য তারা একে অপরের থেকে দূরে সরে যায়। কোন দিকে বিদ্যুৎ প্রবাহিত হবে তা বিদ্যুতের মেরুর উপর নির্ভর করে।

(Refer Slide Time: 03:17)



এই ছবিটার দিকে আপনি একটু তাকালে দেখতে পাবেন পিছনে একটি স্থায়ী চুম্বক দেখানো হয়েছে। এবং কুণ্ডলীটিতে, কুণ্ডলী একটি মধ্যবর্তী point এ পেঁচানো থাকে এটা স্থায়ী চুম্বকটিতে যুক্ত থাকতে পারে, অথবা এটা পৃথক অস্থাবর(movable) অংশও হতে পারে, এভাবে এখানে দুটি ভিন্ন ধরনের speaker আপনি তৈরি করতে পারেন।

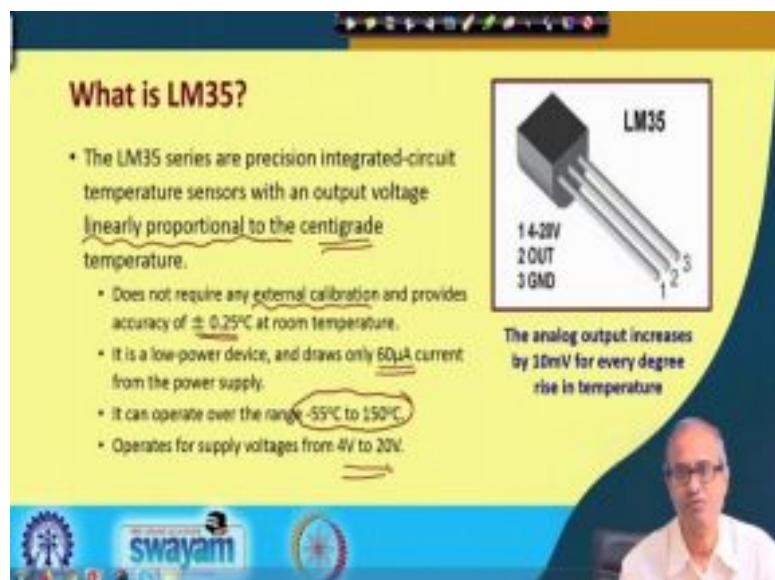
এবং যখন ইনপুট ভোল্টেজ সিগনালের উপর নির্ভর করে বিদ্যুৎ প্রবাহিত হয়, তখন কিছু বিদ্যুৎ কুণ্ডলীর মধ্যে প্রবাহিত হয়। সুতরাং, এই কুণ্ডলী আকর্ষণ বা বিকর্ষণ বলের উপর নির্ভর করে সামনে যায় বা পিছনে আসে এবং একটি পাতলা diaphragm কুণ্ডলীর সাথে যুক্ত থাকে। আপনি নিশ্চয়ই এইরকম diaphragm যেটা উপরে আছে, একটি speaker এর মধ্যে দেখেছেন -- একটি পাতলা নরম উপাদান যেটা সামনে-পিছনে নড়তে শুরু করে এবং একটি কম্পন হয়।

এখন যদি সেই কম্পন আমাদের কানের শোনার ক্ষমতার কম্পাঙ্কের পরিসরে(range) থাকে

তাহলে আমরা একটি শব্দ শুনবো। এখন আপনি জানেন সাধারণ শব্দের পরিসর হল 20 Hz থেকে 20 KHz, মানুষের কান এই কম্পাক্ষ পরিসরে শুনতে সক্ষম। যদি এমন কোন কম্পাক্ষ থাকে যার জন্য diaphragm টি এই পরিসরে কম্পিত হয়, আমরা সেই শব্দ শুনতে পারব; এটা এই ভাবে কাজ করে।

বৈদ্যুতিন চুম্বক আর diaphragmটি উপরে নিচে নড়তে থাকবে, একটি কম্পন হবে এবং কম্পনের কম্পাক্ষটি একটি শব্দ উৎপন্ন করবে। বেশিরভাগ audio system-এ একটি ভালো, খুব কম noise-এর amplifier থাকে তাতে আমরা খুব পরিষ্কার ও সুন্দর শব্দ পাই।

(Refer Slide Time: 05:45)



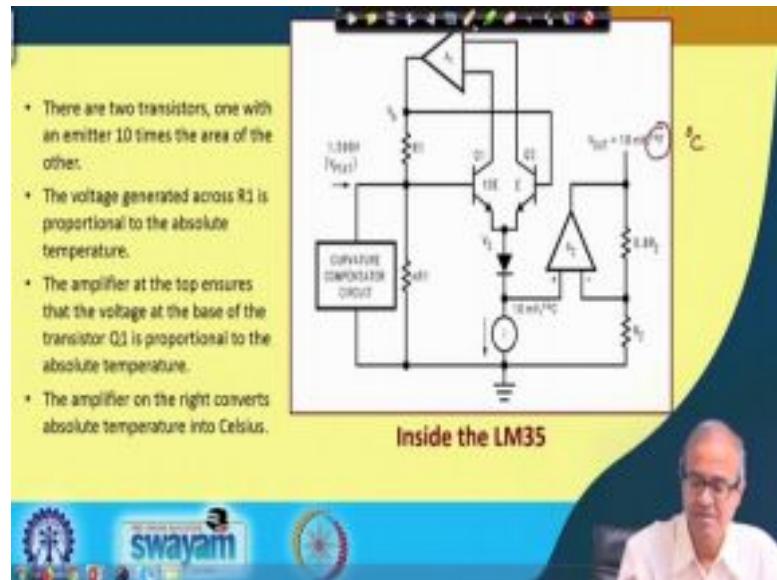
এর পর আমরা এক ধরনের temperature sensor নিয়ে কথা বলবো। এটা একটা খুবই প্রচলিত temperature sensor যা আমরা অনেকেই পরীক্ষা-নিরীক্ষার(experiment) জন্য ব্যবহার করি; একে বলে LM35। এই LM35 খুবই সুবিধাজনক temperature sensor; এটা এই রকম দেখতে লাগে। এখানে 3টা pin আছে; pin সংখ্যা 1 আর pin সংখ্যা 3 হল power supply pins. Pin সংখ্যা 3কে আপনি ground এ আর pin সংখ্যা 1কে আপনি ধনাত্মক(positive) power supply এ যুক্ত করুন; LM 35-এর ধরনের উপর ভিত্তি করে এই power supply voltage 4 থেকে 20 volt অবধি পরিবর্তিত হতে পারে।

এবং মধ্যবর্তী pin সংখ্যা 2 কিছু analog ভোল্টেজ উৎপন্ন করবে যা বাইরের ambient temperature এর সমানুপাতিক হবে। দুই ধরনের LM35 পাওয়া যায়, এদের মধ্যে কিছু ডিগ্রী সেন্টিগ্রেড বা সেলসিয়াসে তাপমাত্রার সমানুপাতিক ভোল্টেজ উৎপন্ন করে, আরেক ধরনের টা ডিগ্রী ফারেনহাইটে দিতে পারে।

আমি যেমন বললাম এই ধরনের sensor-এ আউটপুট ভোল্টেজ তাপমাত্রার সমানুপাতিক হয়। যে ধরনের sensor আমি আপনাকে দেখাচ্ছি, তাতে এটা তাপমাত্রার ডিগ্রী সেন্টিগ্রেডের সমানুপাতিক হবে। এবং সমস্ত circuitry এর মধ্যে তৈরি করা আছে, বাইরের থেকে কোন ধরনের calibration এর দরকার নেই। এবং এর নির্ভুলতা(accuracy) $+/- 0.25^{\circ}\text{C}$ ঘেটা বেশিরভাগ এপ্লিকেশন(application)-এই যথেষ্ট। এবং এটা খুবই কম শক্তি(power) ব্যবহার করে, operation-এর সময় 60 মাইক্রো অ্যাম্পিয়ার(micro ampere) বিদ্যুৎ। সুতরাং, এটি খুব কম শক্তির(low power) যন্ত্র, এবং এটা বিভিন্ন তাপমাত্রার পরিসরে কাজ করতে পারে -55 থেকে +155 ডিগ্রী সেলসিয়াস। power supply-এর পরিসর হল 4 থেকে 20 volt.

আরেকটি লক্ষণীয় বিষয় হলো যে মধ্যবর্তী পয়েন্ট(point) টা একটা analog output ভোল্টেজ উৎপন্ন করছে এবং এটা সমানুপাতিক। এখন সমানুপাতিকতার ধ্রুবক(constant of proportionality) কি? সমানুপাতিকতার ধ্রুবক মানে হল এমন যে প্রতি ডিগ্রি সেন্টিগ্রেড তাপমাত্রার বৃদ্ধির জন্য 10 millivolts করে আউটপুট ভোল্টেজ বৃদ্ধি পাবে, এইভাবে এই যন্ত্র তৈরি করা হয়েছে।

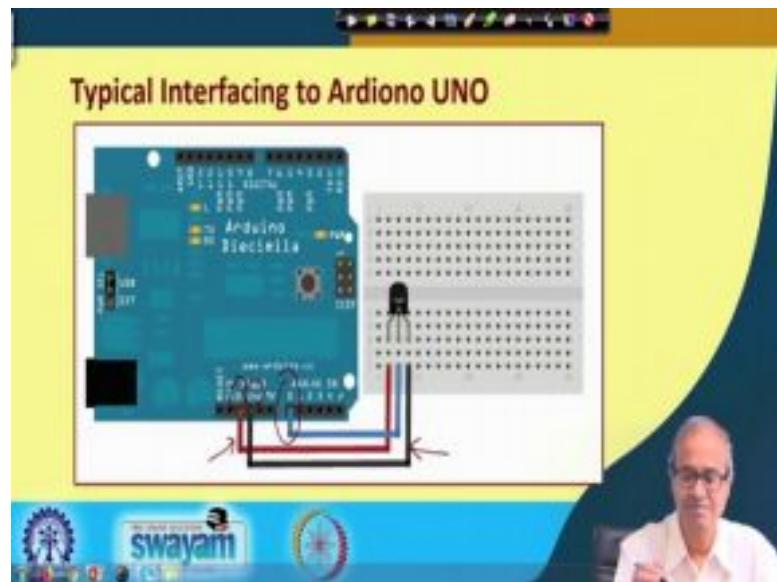
(Refer Slide Time: 08:41)



অভ্যন্তরীণভাবে আমি বিস্তৃত বর্ণনায় যাচ্ছি না, এই যন্ত্র এই রকম দেখতে লাগে, আপনি দেখুন দুটি **transistor**, কিছু **diode**, কিছু **amplifier**, একটি সারাংশণের **বিদ্যুৎ উৎস** আছে। এবং এই দুটি **transistor** খুব অদ্ভুত ধরনের, এদের একটির **emitter** আরেকটির 10 গুণ বড়। আপনি দেখবেন, আমি এই circuit কিভাবে কাজ করে তার বর্ণনায় যাচ্ছি না, কিন্তু ধারণা টা হলো এরকম, যে বিদ্যুৎ এই **transistor** এর মধ্য দিয়ে প্রবাহিত হচ্ছে, ambient তাপমাত্রার উপর তার অনেকটা নির্ভরতা আছে, এইভাবে তাপমাত্রার sensing করা হয়।

এবং এই **2 transistor** এর মধ্য দিয়ে যে বিদ্যুৎ প্রবাহিত হচ্ছে তাদের পার্থক্যকে বৃদ্ধি করা হয় এবং এই রোধ **R1** এর দু'ধারের ভোল্টেজ, হল প্রকৃতপক্ষে পরম তাপমাত্রা(**absolute temperature**) এর সমানুপাতিক। এবং অন্যদিকে এই পরম তাপমাত্রার মান(**absolute temperature value**), অন্য কিছু circuitry-এর মধ্য দিয়ে যে ভোল্টেজ আছে তাকে একটি ভোল্টেজে রূপান্তরিত করা হচ্ছে যেটা ডিগ্রী ফারেনহাইট অথবা ডিগ্রি সেন্টিগ্রেডের সমানুপাতিক করা হয়। সুতরাং, আমি আপনাকে যেমন বলেছিলাম LM35 যন্ত্র টি দুই ধরনের হয়, এক ধরনের টা আপনাকে ডিগ্রি সেন্টিগ্রেডের সমানুপাতিক আউটপুট ভোল্টেজ দেয় আরেকটি আপনাকে ডিগ্রি ফারেনহাইট এর সমানুপাতিক দেয়।

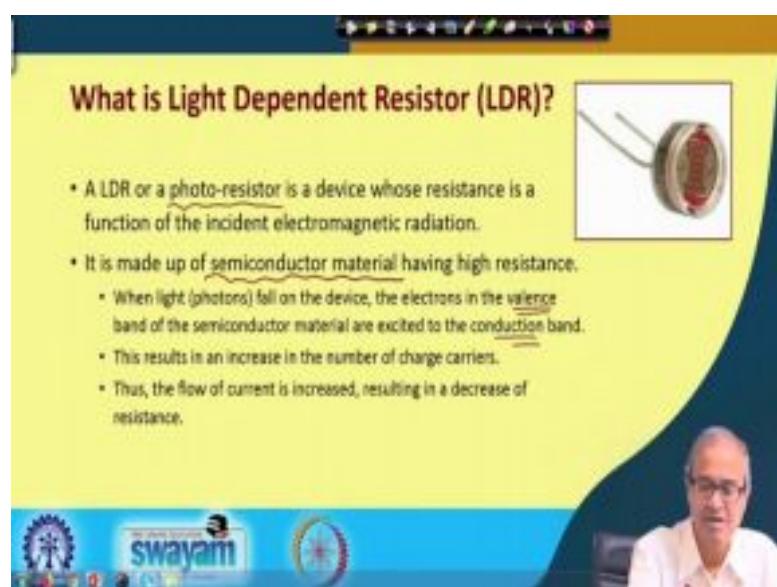
(Refer Slide Time: 10:17)



কিন্তু এটা ব্যবহার করা খুবই সহজ। এখানে আমি একটা ছবি দেখাচ্ছি যাতে একটা LM35 এর সাথে Arduino UNO বোর্ড(board) যুক্ত করার চেষ্টা করছি। সবচেয়ে বামদিকের terminal টা আমরা 5V এর সাথে যুক্ত করছি, এটা সেই টার্মিনাল যেটা 5V উৎপন্ন করছে, এটাকে আমরা সরাসরি 5V এর সাথে যুক্ত করছি, একদম ডান দিকের তারটি এই কালো টি ground এর সাথে যুক্ত থাকে।

এবং এই মধ্যবর্তী টি, যেটির তাপমাত্রার সমানুপাতিক **analog output উৎপন্ন করার কথা,**
সেটি একটি port লাইনের সাথে যুক্ত থাকে। আপনি দেখুন আমরা একটি এনালগ ইনপুট এর সাথে যুক্ত করেছি, কারণ এটি একটি এনালগ ভোল্টেজ। আপনি অন্তর্নির্মিত(built-in) A/D converter এর মাধ্যমে এনালগ ভোল্টেজ এর মান পড়তে পারবেন, আপনি ডাটা (data) পড়তে পারবেন এবং আপনি গণনা করতে পারবেন যে বাইরে প্রকৃত তাপমাত্রা কি।

(Refer Slide Time: 11:23)



এখন আমরা একটি sensor এর ব্যাপারে কথা বলব, যাকে বলে **light dependent resistor**

(LDR) যা আলোকে অনুভব(sense) করতে পারে। LDR একটি খুব ছোটো যন্ত্র(device)। এটা

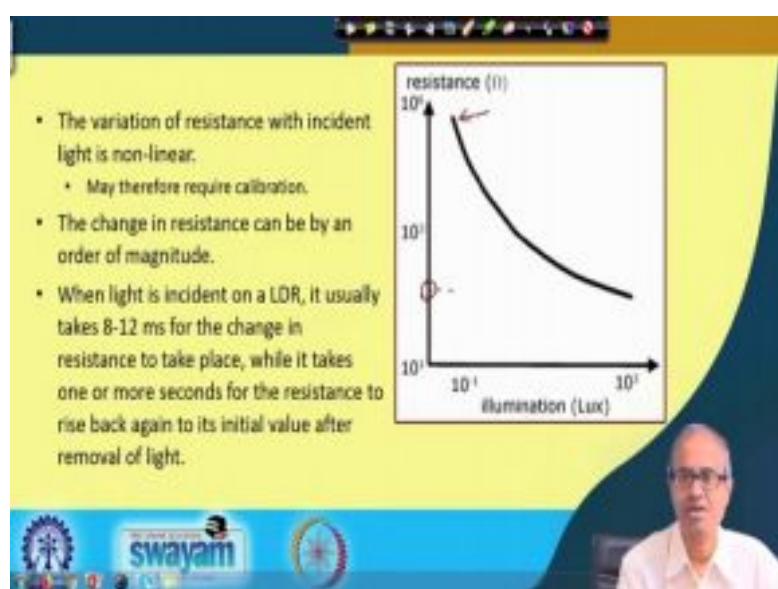
এরকম দেখতে লাগে, এটা একটি 2 terminal বিশিষ্ট যন্ত্র এবং এখানে একটা স্বচ্ছ জানালা(transparent window) থাকে, উপরে আপনি দেখতে পারবেন এর ভিতরে এই ধরনের zigzag pattern আছে, একটি LDR এ রকম দেখতে লাগে। এই LDR কে কখনো photo resistor বলে; এটা এক ধরনের রোধ যার মান(value) এর উপর আপত্তি আলোর উপর নির্ভর করে পরিবর্তিত হয়। এভাবে LDR কাজ করে। কিন্তু আলো আপত্তি হল তার উপর নির্ভর করে রোধ টি পরিবর্তিত হয়। আপনি রোধ টা পরিমাপ করতে পারেন যা আপনাকে আপত্তি আলোর তীব্রতার ব্যাপারে একটা ধারণা(idea) দেবে।

অ্যান্টরীণভাবে এটা কিছু **semiconductor** উপাদান(material) দিয়ে তৈরি, কারণ

semiconductor উপাদান কিছু **photo resistive** ধর্ম(property) আছে। যখন

আলো পতিত হয়, তখন প্রতিরোধতা(resistivity) পরিবর্তিত হয়। যখন আলো বা photons যন্ত্রটির উপর পতিত হয় তখন যন্ত্রটির ভিতরে থাকা electronগুলি একদিক থেকে আরেক দিকে যায়, সাধারণত তাদের valence bond বলে, তারা conduction bandএর দিকে যায়। যদি তারা valence band থেকে conduction band এর দিকে যায়; তার মানে, electronগুলি আরো বেশি নড়াচড়া করে যার ফলে রোধ কমে যায় কারণ electron এর প্রবাহ বিদ্যুৎ প্রবাহের সৃষ্টি করে।

(Refer Slide Time: 13:23)



অন্য একটি ধর্ম(property) হল যত ওজ্জ্বল্য(illumination) বৃদ্ধি পায় তত exponentially

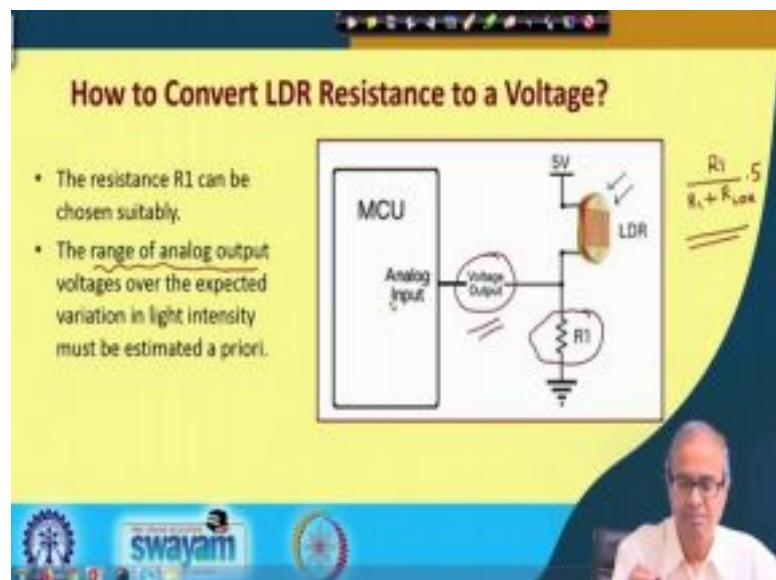
রোধ কমো LM35 এর মত এটা linear ঘন্ট নয়; রোধটা তীব্রতার মাত্রার সমানুপাতিক নয়,

এটা এক ধরনের exponential বক্ররেখার(curve) মত চলে।

আরেকটি লক্ষণীয় ব্যাপার হলো এই পার্থক্যটা খুবই তাৎপর্যপূর্ণ হতে পারে। আপনি দেখুন এই জাতীয় বক্ররেখায়(curve) আমাদের রোধ হয় mega ohm-এর ক্রমে(order), এবং এই point এ আমাদের রোধ হবে কয়েকশো ohm এর ক্রমে(order)। সুতরাং mega ohm থেকে কয়েকশ �ohm একটা বিশাল পার্থক্য। যদি আপনার রোধ-এর এই পরিবর্তন বোঝার(sense) জন্য উপযুক্ত circuitry থাকে, আপনি খুব নিখুঁত ও বিশ্বস্ত ভাবে আলোর তীব্রতার স্তর বুঝতে পারবেন।

আরেকটি লক্ষণীয় বিষয় হলো এই ঘন্ট গুলো কিন্তু ভীষণ দ্রুত নয়; এগুলি আউটপুট উৎপন্ন করতে কিছু সময় নেয়। যেমন যখন আলো পড়ে তখন এই ঘন্টগুলি সাড়া দিতে ও আউটপুট দিতে সাধারণত 10 মিলি সেকেন্ডের মত সময় নেয়। আবার যখন আলো সরিয়ে নেয়া হয়, আপনি এটিকে অন্ধকার জায়গায় রাখেন, এটার রোধ কিছু সেকেন্ড সময় নেয় পূর্বের উচ্চ রোধ-এর অবস্থায়(high resistance state) ফেরত যেতে।

(Refer Slide Time: 15:23)

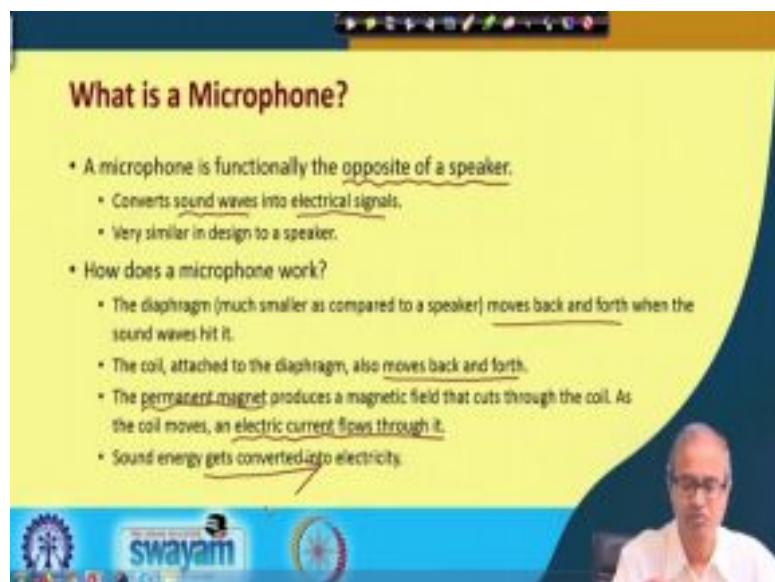


এখন interfacing এর ব্যাপারে বলা যায়, সাধারণত আমরা এক ধরনের resistance divider circuit ব্যবহার করি। যেমন আমরা একটি LDR ব্যবহার করতে পারি, আমরা একটি

রোধ R1 কে 5V supply-এর সাথে ব্যবহার করতে পারি, আমরা একে microcontroller-এর একটি এনালগ ইনপুট pin-এ পাঠাতে পারি।

সুতরাং, আপনাকে প্রথমে প্রত্যাশিত পরিবেশে এনালগ ইনপুট ভোল্টেজের পরিসর(range) খুঁজতে হবে। আলোর তীব্রতা একটি নির্দিষ্ট মানের নিচে গেছে কিনা তা দেখার জন্য আপনি হয়তো LDR ব্যবহার করবেন। সুতরাং, আপনাকে একটি পরীক্ষা করতে হবে এবং দেখতে হবে অঙ্ককার অবস্থায় এবং আলোকিত অবস্থায় রোধ এর মান কত। সুতরাং, এই R_{LDR} এর মান পরিবর্তিত হচ্ছে। সেই ভাবে আপনাকে R1 এর মান পছন্দ করতে হবে, যাতে ভোল্টেজের আউটপুটের পরিবর্তন তাঁপর্যপূর্ণ হয়। আপনি মাইক্রোকন্ট্রোলার এর এই এনালগ ইনপুটটি পড়তে পারবেন এবং সেটা নিয়ে কি করবেন তা নিয়ে সিদ্ধান্ত নিতে পারবেন।

(Refer Slide Time: 17:03)



এখন, শেষ যে ধরনের sensor নিয়ে আমরা এই বক্তৃতায় বলবো তা হল মাইক্রোফোন(microphone)। Microphone এক ধরনের শব্দ sensor, কিছু শব্দ উৎপন্ন করা হয়, সেটা microphone এ গ্রহণ করা হয় এবং এটিকে বৈদ্যুতিন আকার দেওয়া হয়।

সুতরাং, microphone কি? কার্যকারিতার দিক থেকে অভ্যন্তরীণভাবে এটা speaker-এর মতই, কিন্তু এটা একদম বিপরীত ভাবে কাজ করে। একটা speaker একটি বৈদ্যুতিক signalকে শব্দে রূপান্তরিত করে; microphone শব্দকে বৈদ্যুতিক signal এ **রূপান্তরিত করে।** Designটাও পুরো একই রকমের; এখানেও একটি diaphragm রয়েছে। কিন্তু, microphone গুলো speaker এর থেকে অনেক ছোট হয়, speaker সাধারণত আকারে বড় হয়।

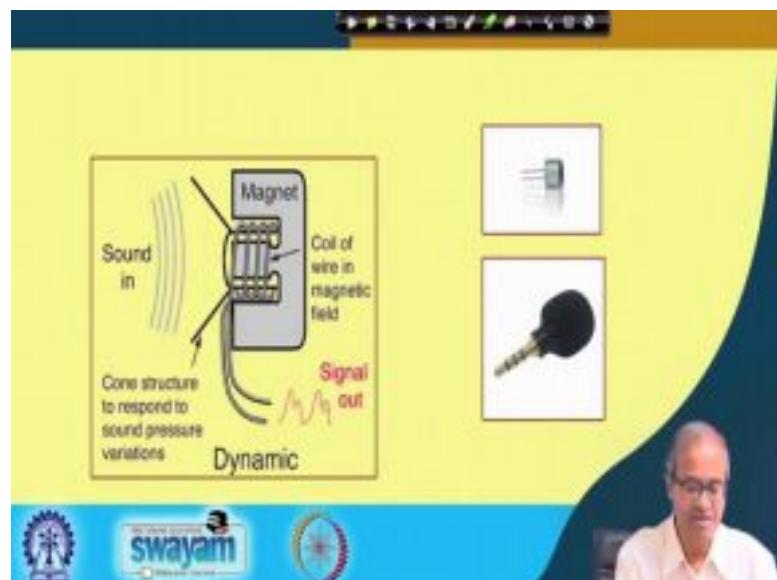
এর বিপরীতে microphone আকারে খুবই ছোট হয়; এখানে এর ডিতরে একটি খুব ছোট

diaphragm থাকে। স্থায়ী চুম্বকের চারপাশে diaphragm টির চারপাশে একটি কুণ্ডলী(coil) থাকে, যখন আপনি কথা বলেন diaphragm টি কম্পিত হয়, এখন আপনি কোন বিদ্যুৎ কুণ্ডলীর(coil) ডের দিয়ে পাঠাচ্ছেন না বরং আপনি শব্দের প্রতিক্রিয়ায় diaphragmটি কম্পিত হতে দেন।

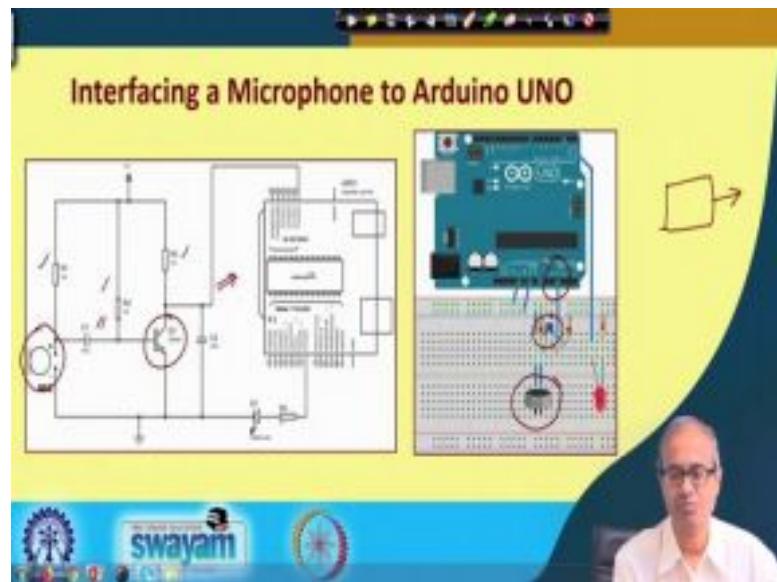
যখনই কুণ্ডলী টি(coil) একটি চুম্বকীয় মাধ্যমে কম্পিত হয়, তখনই একটি ভোল্টেজ কুণ্ডলীতে আবেশিত হবে(will be induced) এবং আপনি ভোল্টেজটি মাপতে পারবেন।

এটা প্রকৃতপক্ষে এভাবে কাজ করে। diaphragm টি আগে পিছে নড়াচড়া করে, diaphragm এর সাথে যে কুণ্ডলীটি থাকে সেটিও আগে পিছে নড়াচড়া করবে। যেমন ভাবে একটি speaker এ একটি স্থায়ী চুম্বক থাকে তেমনি কুণ্ডলী(coil) টির আশেপাশে একটি চৌম্বক ক্ষেত্র(magnetic field) থাকবে। যখনই কুণ্ডলী টা নড়ে তখনই এর মধ্যে দিয়ে তড়িৎ প্রবাহিত হয়, যে কারণে একটি ভোল্টেজ আবেশিত(induced) হবে। তাই, শব্দ শক্তি বৈদ্যুতিক শক্তিতে রূপান্তরিত হয়। এইভাবে একটা microphone কাজ করে।

(Refer Slide Time: 19:51)



(Refer Slide Time: 20:43)



এখন interfacing অতটা কঠিন নয়। আমি একটি সাধারণ interface দেখাচ্ছি, এর আরো সহজ উপায়ও আছে। আপনি দেখুন এটি একটি microphone এর প্রতীকী চিত্র(symbolic diagram), এখানে আমি আপনাকে দেখাচ্ছি একটা transistor level amplifier যাতে কিছু রোধ(resistances) ও কিছু capacitance আছে। যখনই আপনি কথা বলছেন শব্দ ভোল্টেজ-এ রূপান্তরিত হচ্ছে, যেটা 0 থেকে 5 ভোল্ট এর মধ্যে থাকে। এখানে সাধারণ একটি interface দেখান হয়েছে যাতে আপনি microphone টি দেখতে পাবেন এবং অন্য circuitry আপনি এখানে দেখবেন, এটি আপনার transistor এবং আপনি এটি Arduino UNO এর সাথে যুক্ত করেছেন।

এখন আমি আপনাকে বলব, যখনই আপনি পরীক্ষা-নিরীক্ষার জন্য একটা এই ধরনের microphone কিনতে চাইবেন, আপনি দেখবেন কিছু জিনিস উপলব্ধ আছে যাতে আগে থেকেই এই ধরনের circuitry এর মধ্যে তৈরি করা আছে। যদি আপনি এটা কিনে থাকেন আপনার আর এই circuit গঠন করার দরকার নেই, সরাসরি এই ইউনিট(unit) টা আপনাকে একটা ভোল্টেজ দেবে যেটা শব্দের সমানুপাতিক।

এর সাথেই আমরা বক্তৃতাটির শেষে এসে পড়েছি। পরবর্তী বক্তৃতায় আমরা এই ধরনের আরও কিছু যন্ত্র নিয়ে কথা বলব যেগুলো আমরা আপনাকে কিছু interfacing পরীক্ষা দেখানো বা প্রদর্শন করানোর জন্য ব্যবহার করব।

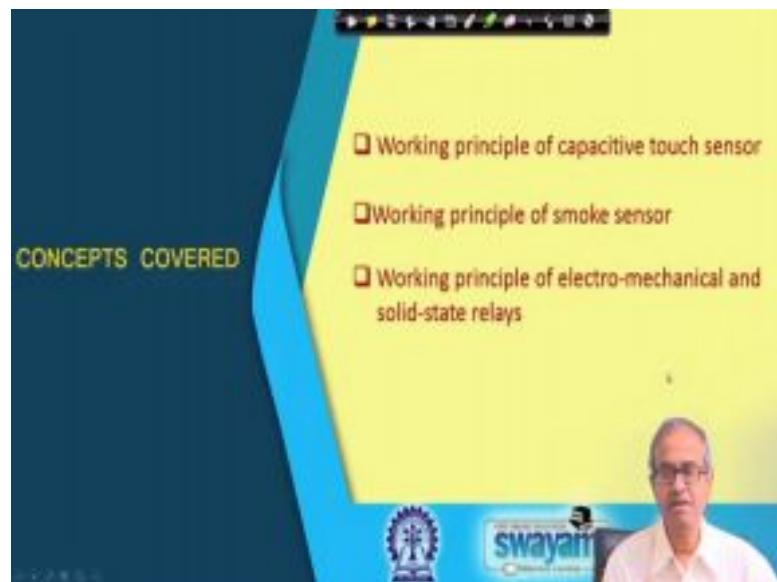
ধন্যবাদ।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 17
Output Devices, Sensors and Actuators (Part III)

আমরা আমাদের **sensor** এবং **actuator** নিয়ে আলোচনা চালাতে থাকবো।

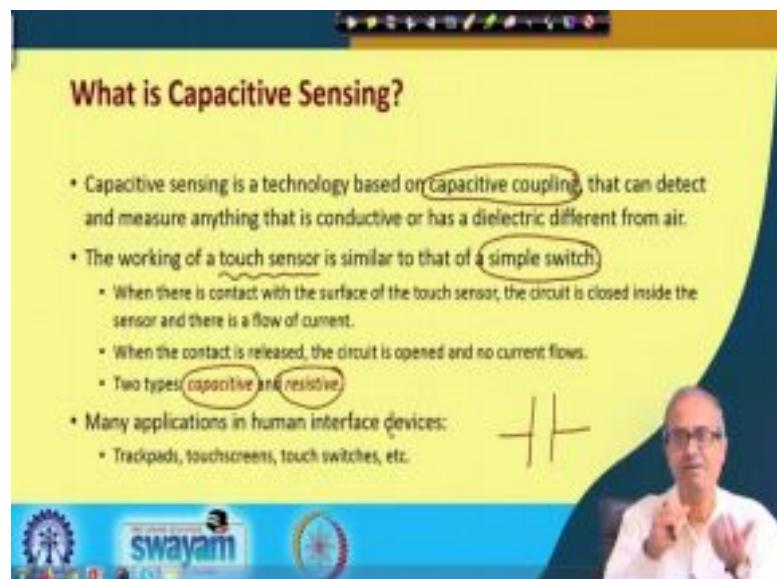
(Refer Slide Time: 00:29)



বিশেষত, আমরা **capacitive touch sensor** ইনপুট যন্ত্র হিসেবে, **smoke sensor** ইনপুট যন্ত্র হিসেবে আলোচনা করবো এবং তারপর আমরা **বিভিন্ন ধরনের relay** নিয়ে আলোচনা করব।

প্রথমে আমরা **capacitive touch sensor** নিয়ে কথা বলব যেটা **physical touch** চিহ্নিত করতে পারে। আমরা সবাই অনেক ধরনের touch-activated যন্ত্রের সাথে পরিচিত, যেমন আমাদের মোবাইল স্পর্শ সংবেদনশীল(touch sensitive)। আমরা আমাদের হাতের আঙুল দিয়ে স্ক্রিনে navigate করি; এছাড়াও অন্য অনেক ইনপুট যন্ত্র আছে, যেখানে আপনি স্ক্রিনটা ছুঁয়ে ইনপুট দিতে পারেন; সেখানে আলাদা কোন **keyboard** নেই। এটা একটা technology-- capacitive touch sensing ব্যবহার করে আমরা এই ধরনের sensing device তৈরি করতে পারি।

(Refer Slide Time: 01:29)



এখন, প্রথমে আমরা capacitive sensing কি এই নিয়ে সব কিছু জানবো। এটা capacitive coupling এর উপর নির্ভরশীল একটি technology. Capacitance কি? যখন সমান্তরাল প্লেট(plate) এর মত দুটি পদার্থ(material) কে একে অপরের কাছে নিয়ে আসা হয়, তখন capacitance তৈরি হয়। যখন আপনি আপনার আঙুল কে একটি পদার্থের কাছে আনেন, তখন আপনার আঙুলে থাকা আর্দ্ধতার জন্য ওই পদার্থ এবং আপনার আঙুল এর মধ্যে capacitance তৈরি হয়, কারণ আপনার দেহটি সুপরিবাহী। সুতরাং, যদি আপনি আপনার আঙুল আনেন সেটাও capacitance এর মানকে(value) প্রভাবিত করবে। touch sensor যেটা capacitive sensing দিয়ে তৈরি হয়, সেটা একটা switch এর মত। যখনই আপনি স্পর্শ করবেন কিছু সুইচ(switch) বন্ধ(closed) হয়ে যাবে, যখন আপনি ওই স্পর্শ দূরে সরিয়ে নেবেন সুইচ খুলে যাবে(open)।

সুতরাং, একটা সাধারণ push button সুইচ এর পরিবর্তে, আপনি একটি touch ধরনের সুইচও ব্যবহার করতে পারেন। এবং বিস্তারিত ভাবে বলতে গেলে এই ধরনের touch sensors গুলো capacitive বা resistive হতে পারে। অবশ্যই, capacitive touch sensor গুলো কর্মসূচিতার দিক থেকে অনেক বেশি নমনীয় ও ভালো হয়। আমি আপনাকে যেমন বলেছিলাম অনেক applications এ আমরা touch sensor ব্যবহার করি; যেমন track pads. বেশিরভাগ laptop-এই আলাদা mouse থাকে না; একটা সমতল থাকে, যাতে আঙুল সরিয়ে আপনাকে screen এ cursor টা সরাতে হয়; একে বলে trackpad।

(Refer Slide Time: 03:25)

Principle of Operation

- The capacitance of a parallel plate capacitor = $\epsilon_0 \cdot \epsilon_r \cdot A / d$, where ϵ_0 is the permittivity of free space, ϵ_r is the relative permittivity of the dielectric material, A is the area of the plates, and d is the distance between them.
- The capacitance will increase if a conductive object touches or approaches the sensor electrode.

একটা parallel plate capacitor কেমন দেখতে লাগে এই ছবি(diagram)টি তা দেখায় ।

এখানে একটি plate আছে, যার ক্ষেত্রফল (area) হলো A, এখানে দুটো সমান্তরাল plate আছে, তাদের দূরত্ব হল d, এদের মধ্যে একটি উপাদান থাকে, যাকে বলে dielectric, যার dielectric ধ্রুবক আছে।

এখন, ϵ_0 হল শূন্য স্থানের(free space) permittivity, এবং ϵ_r হলো দুটো প্লেটের(plate)

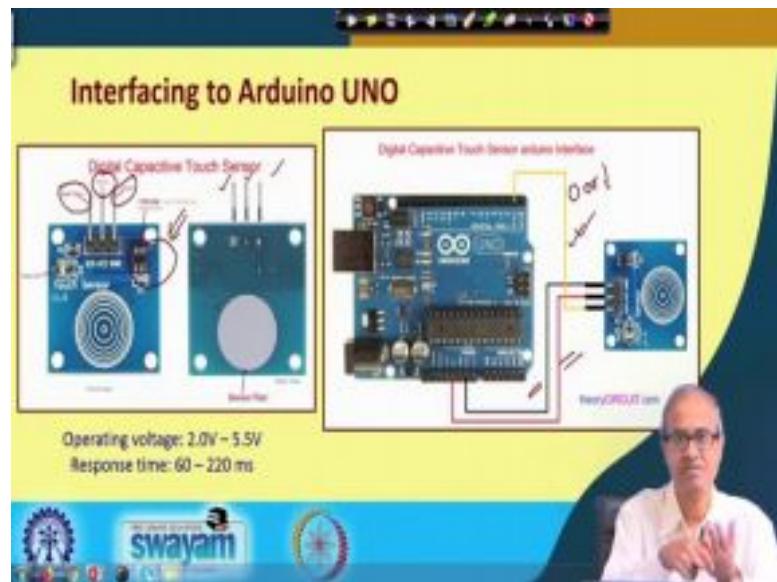
মধ্যেখনের permittivity । যখন আপনি এর কাছে একটি আঙুল নিয়ে আসেন, আপনার আঙুলের মধ্যে দিয়ে একটি coupling হবে। যদি আপনি আঙুল রাখেন, এই দুটো plate, capacitive effect এর মাধ্যমে একটা coupling পাবে। একটা touch sensor মূলত এইভাবে কাজ করে।

সুতরাং যখন কোন বস্তু স্পর্শ করে বা এমনকি আপনি আঙুল এর খুব কাছে নিয়ে আসেন

capacitance এর মান বৃদ্ধি পায়। যদি আপনার কাছে capacitance এর পরিবর্তন

চিহ্নিতকরণের উপযুক্ত circuitry থাকে, আপনি touch টা চিহ্নিত করতে পারবেন।

(Refer Slide Time: 05:19)



এক ধরনের যন্ত্র যেটা আমরা পরীক্ষা-নিরীক্ষার জন্য ব্যবহার করব, সেটা এরকম দেখতে।

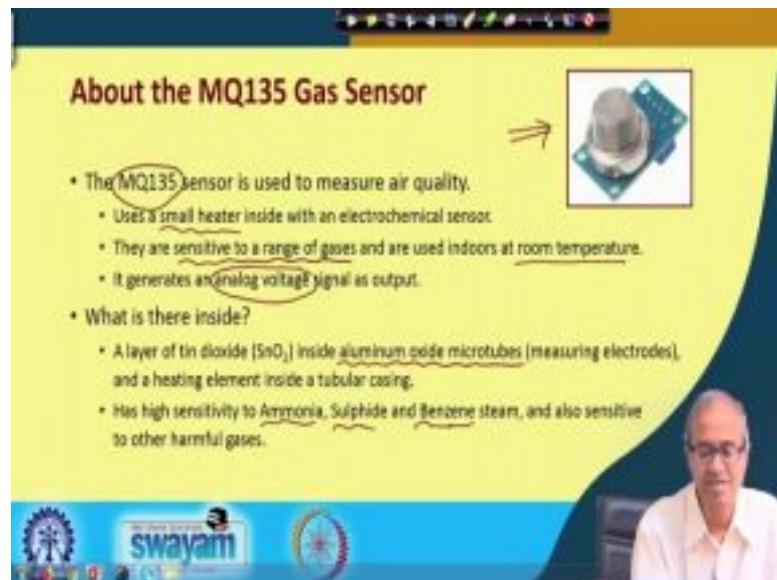
আপনি দেখবেন এখানে কিছু বৃত্তাকার pattern আছে, যেটা আপনার sensor pad; যদি আপনি আপনার আঙুল এই বৃত্তাকার অঞ্চলের উপরে রাখেন, তাহলে capacitor টির মান(value) পরিবর্তন হবে। আর এই বোর্ডে(board) কিছু বৈদ্যুতিন circuitry থাকে। আপনি দেখুন এখানে একটি ছোট IC আছে যার মধ্যে sensing এবং conditioning circuitry থাকে। সুতরাং interfacing টি খুব সহজ হয়ে যায়; এখানে 3টি pin আছে --- একটা হল +5 ভোল্ট পাওয়ার সাপ্লাই(power supply), অন্যটা হল ground এবং তৃতীয়টি হল এনালগ সিগন্যাল আউটপুট(analog signal output); আপনার স্পর্শের উপর নির্ভর করে, যে আউটপুট ভোল্টেজ আপনি পড়বেন, সেটা কত।

যখন এইরকম sensor কে আপনি microcontroller board-এ interfacing করছেন,

এটা খুব সোজা হয়। আপনি signal আউটপুটটি একটি port pin-এর সাথে যুক্ত করেছেন। এখন এই port pin কিন্তু এনালগ আউটপুট pin নয়, এটা একটা ডিজিটাল আউটপুট পিন(pin)। এর 0 বা 1, মানে আপনি একে হয় স্পর্শ করেছেন না করেননি।

এরপর আমরা smoke sensing নিয়ে কথা বলবো। এমন অনেক application আছে, যেখানে আপনাকে sensor install করতে হবে। উদাহরণ স্বরূপ, আপনার বাড়িতে আপনি cylinder-এর LPG gas-এ কোন ছিদ্র আছে কিনা দেখার জন্য একটা sensor install করতে চান। sensorটি LPG gas-এ কোন ধোঁয়া নিগতি হলে জানাতে পারে, সেখানে কোন এক ধরনের sensor থাকা দরকার।

(Refer Slide Time: 07:26)

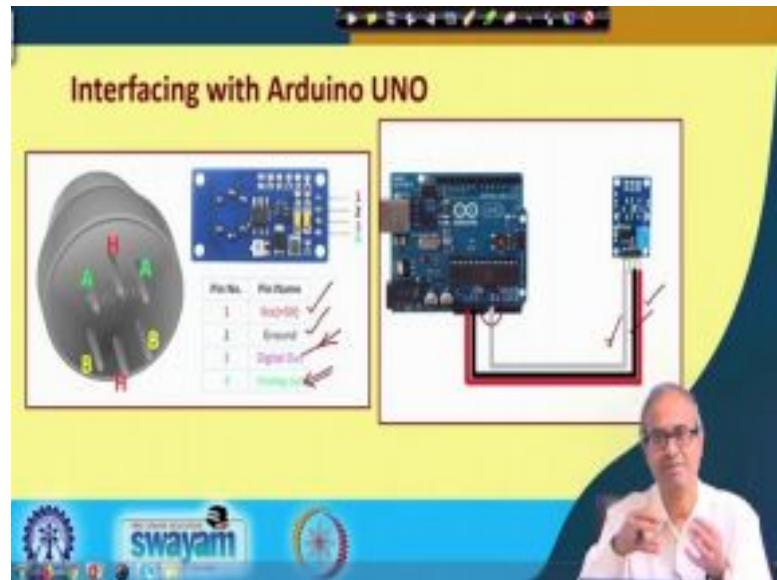


আপনি এখানে ডানদিকে একটা sensor দেখতে পাবেন। এই পরীক্ষাটিতে আমরা আপনাকে এ ধরনের sensor দেখাবো। আপনি দেখবেন এখানে একটা ছোট mesh ধরনের mask রয়েছে, এতে আপনাকে গ্যাস দিতে হবে, আপনি যে পরিবেশে গ্যাসটা অনুভব করছেন(sensing) সেই পরিবেশে এটা রাখতে পারেন। আপনি আপনার শ্বাস পরীক্ষা করতে পারেন; আপনি এটার উপর শ্বাস ছাড়তে পারেন। এটা পরীক্ষা করতে পারেন যে আপনার শ্বাস যেটা বেরিয়ে আসছে, তাতে অ্যালকোহলের বাঞ্চ আছে কি নেই। এবং এই sensor, যেটা আমরা ব্যবহার করব, এর নাম হলো MQ135.

এখানে অভ্যন্তরীণভাবে অনেক কিছু আছে। এখানে খুব ছোট সাইজের একটি বৈদ্যুতিক heater আছে, যেটা ভেতরের পদার্থটিকে গরম করে। এবং কিছু aluminium oxide micro tubes থাকে, এর ভেতরে খুব পাতলা tube থাকে, এবং মাঝখানে একটি heater এর কুণ্ডলী(coil) থাকে। সুতরাং, যখন এটা গরম হয়, ধোঁয়াটা এই শ্রেণীবদ্ধ তল(graded surface) দিয়ে ঢোকে, আর এই tube গুলির সাথে বিক্রিয়া করে। এখানে tin oxide এর একটা পাতলা স্তর থাকে, যেটা micro tubes এর ভেতরে থাকে এবং প্রধান sensor হিসেবে কাজ করে।

Tin oxide এর উপাদান(material) এর ধর্ম এটি যে ধরনের গ্যাসের সংস্পর্শে আসছিল তার উপর ভিত্তি করে তারতম্য হয়। এই ধরনের sensor অ্যামোনিয়া(ammonia), sulphide, benzene ও অ্যালকোহল এবং অন্যান্য শক্তিকর গ্যাসে সাড়া দেয়(respond), কিন্তু গ্যাসের প্রকারের উপর নির্ভর করে, আউটপুট ভোল্টেজ-এ কতটা পরিবর্তন হবে, এটা এক গ্যাস থেকে আরেক গ্যাসে আলাদা হয়।

(Refer Slide Time: 09:58)



যেমন আমি এই যন্ত্রটা দেখাচ্ছি, এটাতে চারটি pin আছে; একটা হল 5 volt power supply, ground, আর এই analog out. এখানে গ্যাসের উপর ভিত্তি করে একটা continuous voltage পাবেন, কিন্তু আপনি আরও একটি digital out signal পেতে পারেন, এটা আপনাকে বলে দেবে যে গ্যাস আছে কি নেই। এটার একটি সর্বোচ্চ মাত্রা আছে, এটা ঠিক করা যায়(can be set), এর উপর ভিত্তি করে আপনি হয় 0 অথবা 1 পাবেন।

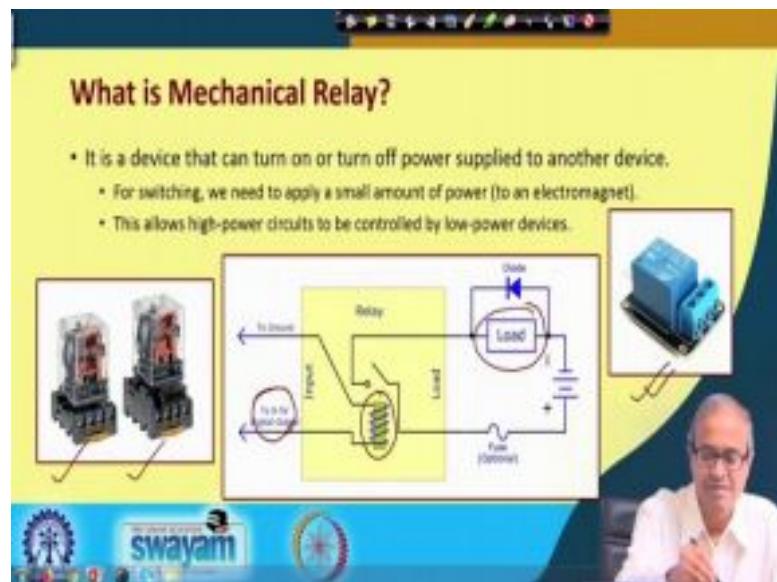
যদি আপনার digital out না দরকার থাকে, শুধু analog out চান, তাহলে আপনি শুধু তিনটি পিন(pin) যুক্ত করুন analog out টি আপনি কোন একটি এনালগ ইনপুট পিন, 5 volts এবং ground এর সাথে যুক্ত করতে পারেন।

এই circuit- এতে প্রয়োজনীয় সব ধরনের বৈদ্যুতিন circuit এর মধ্যে থাকে, সুতরাং যখন আপনি এটা interface করতে যান, সেটা ব্যবহারকারীর পক্ষে খুব সহজ হয়। সেটাই এর বড় সুবিধা।

এখন, আমরা actuator এ আসি। আমরা শুধুমাত্র বিভিন্ন sensor নিয়ে কথা বলেছিলাম যেখান থেকে আমরা ডাটা(data) পড়তে পারি, কিন্তু এখন আমরা কিছু যন্ত্রকে নিয়ন্ত্রণ করতে চাই। আমরা একটি heater কে on বা off করতে চাইতে পারি, আমরা আমাদের AC

যন্ত্রিকে বা যে কোন যন্ত্র, যা আপনি ভাবতে পারেন, on বা off করতে চাইতে পারিব। এই যন্ত্রগুলি বেশিরভাগই উচ্চ ক্ষমতাসম্পন্ন (high power) যন্ত্র, আপনি এগুলো সরাসরি, আপনার মাইক্রোকন্ট্রোলার থেকে ভোল্টেজ এর মাধ্যমে নিয়ন্ত্রণ করতে পারবেন না। সুতরাং, আপনার প্রয়োজন কোন এক ধরনের যন্ত্র যা উচ্চক্ষমতা সম্পন্ন বৈদ্যুতিক বা circuit line কে switch করতে পারে, এগুলিকে relay বলে; ওই ধরনের অ্যাপ্লিকেশনের জন্য আপনার relay দরকার।

(Refer Slide Time: 12:25)



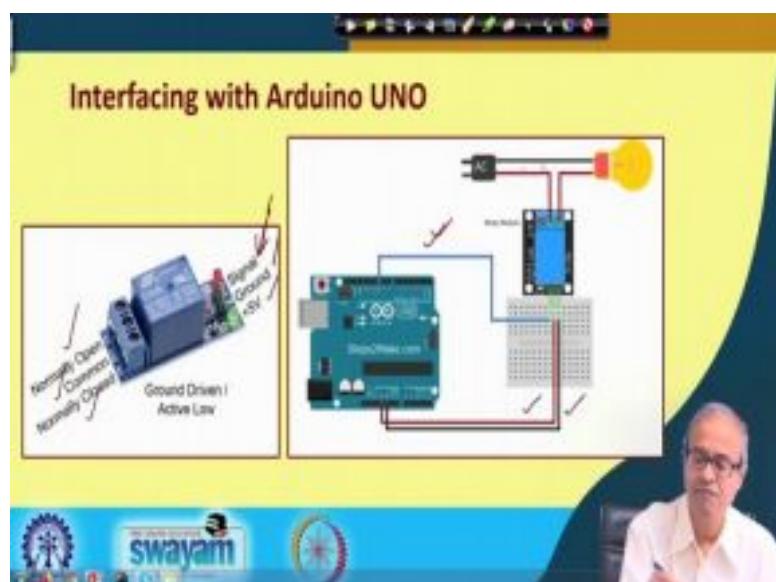
Relay দুই ধরনের হতে পারে, mechanical এবং solid state. প্রথমে আমরা mechanical relay নিয়ে কথা বলব। কখনো এগুলিকে electromechanical relayও বলে, কারণ আপনি একটি mechanical যন্ত্র নিয়ন্ত্রণ করছেন, বৈদ্যুতিক signal দিয়ে। বামদিকে আপনি দেখবেন এই ধরনের কিছু electromechanical relay-র ছবি, যেখানে একটি স্বচ্ছ উপাদানের (material) মধ্যে দিয়ে আপনি ভিতরে কি আছে তা দেখতে পাবেন। এদের অনেকগুলি সীল করা থাম-এ পাওয়া যায়, আপনি ভিতরে দেখতে পারবেন না, যেমন একদম ডানদিকের টির মত। এই ধরনের relay আমরা আমাদের পরীক্ষায় ব্যবহার করব।

এখন অভ্যন্তরীণভাবে আপনি দেখবেন এখানে কি আছে। এর ভিতর একটি ছোট তড়িৎ-চৌম্বক (electromagnet) আছে এবং একটি সুইচ (switch) আছে যাতে একটি অস্থায়ী terminal spring-loaded সুইচ (switch) আছে। যখন তড়িৎ-চৌম্বক সক্রিয় হয়, এটা চুম্বকে পরিণত হয়, সুইচটি (switch) আকর্ষিত হয় এবং circuitটা বন্ধ হয় এবং যদি আপনি বিদ্যুৎ টা

সরিয়ে নেন তবে circuit টা আবার খুলে যায়। এখন এই সার্কিট যেটা খুলে যায় ও বন্ধ হয়, সেটা একটা উচ্চ ক্ষমতার (high power) circuit, এটা উচ্চ লোড (high load) চালাতে ব্যবহৃত হতে পারে।

কিন্তু অন্যদিকে switch টি টানা ও ছাড়ার জন্য (pull and release), এই যে বিদ্যুৎ, যেটা আপনি তড়িৎ চুম্বক (electromagnet) এর মধ্য দিয়ে পাঠাচ্ছেন, এটি একটি খুব কম ক্ষমতার circuit হতে পারে, এটা 5 volt power supply তেও কাজ করতে পারে। আপনি একটা খুব কম ক্ষমতার (power) signal ব্যবহার করছেন একটা উচ্চতর ক্ষমতার circuit নিয়ন্ত্রণ করার জন্য। এই দুটো circuit পরস্পর থেকে পৃথক থাকে, তারা সরাসরি যুক্ত থাকে না; এই দুটো circuit এর মধ্যে কোন বাস্তবিক যোগাযোগ নেই।

(Refer Slide Time: 14:23)



নিয়ন্ত্রণের জন্য আপনি একটি পয়েন্ট (point) ground-এ যুক্ত করেন, 5 volt, আর এখানে আপনি একটি signal প্রয়োগ করছেন এটা on এবং off করার জন্য। দুই ধরনের relay পাওয়া যায়, একটিকে বলে ground driven, মানে, যদি আপনি একে 0 করেন, এটা on হবে। আর আরেক ধরনের টি ground driven নয়, এটা voltage driven, যখন একে 1 করবেন, এটা on হবে। আর আউটপুট এর দিকে আপনি দেখবেন সেখানে তিনটি terminal দেওয়া আছে, এগুলোকে বলে normally open (NO), normally closed (NC), এবং the common terminal.

NO মানে সাধারণভাবে open terminal এবং যখন এই সুইচ(switch) open থাকে, common শুলি সাধারণত অভ্যন্তরীণ ভাবে open হয়। সুতরাং, যখন সুইচ(switch) open হয় তখন এটা open থাকে, আর যখন সুইচ(switch) closed হয় তখন এটা closed থাকে, কিন্তু NC টা বিপরীত হয়। যদি আপনি NC এবং common এর মধ্যে একটি যন্ত্র যুক্ত করেন, তাহলে যখন switch open হয় তখন এই circuit সাধারণ ভাবে closed থাকে, কিন্তু যখন আপনি switchটা সক্রিয় করবেন, তখন circuit টা open হবে।

আমি এই ধরনের module দিয়ে একটা খুব সাধারণ interface দেখিয়েছি। একদিকে আপনি এটা Arduino board, Vcc, ground দিয়ে নিয়ন্ত্রণ করছেন আর একটি digital port pin দিয়ে আপনি relay নিয়ন্ত্রণ করছেন। এবং অন্যদিকে আমি সাধারণ ভাবে open connection ব্যবহার করছি। এটা আমি একটা circuit এ যুক্ত করছি, ধরুন আমরা বললাম একটি bulb কে AC mains এর সাথে।

যখন আমি এটা সক্রিয় করছি এবং relay টা on হচ্ছে, তখন bulb টা জ্বলবে, আর যদি off হয় তবে bulbও off হবে, এটা এভাবে কাজ করে। এই interface টা খুবই সাধারণ, আমরা পরে এ ধরনের interfacing এর পরীক্ষা দেখব। এখন relay এর আরেকটি version আছে যেটা তড়িৎ চুম্বক-এর mechanically সুইচটি(switch) টানা ও ছাড়ার উপর নির্ভর করে না, এগুলিকে বলে solid state relay.

(Refer Slide Time: 16:58)

What is Solid-state Relay?

- It is an electronic switching device that switches ON or OFF when a small external voltage is applied across its control terminals.
- It consists of:
 - A sensor that responds to an appropriate input (control signal).
 - A solid-state electronic switching device that switches power to the load circuitry.
 - A coupling mechanism to enable the control signal to activate this switch.
- Similar to electromechanical relay in functionality, but no moving parts.
 - They use semiconductor devices like thyristors and power transistors.
 - Currents of 100's of amperes can be switched.

Solid state মানে হল সবকিছু একটি chip-এর ভিতরে তৈরি, **semiconductor** যন্ত্র দিয়ে।

এটা আপনি একটা বৈদ্যুতিন (electronic) switching যন্ত্র বলতে পারেন, তড়িৎ চুম্বকীয় relay এর মত, এখানে কোন mechanical গতিশীল (moving) অংশ নেই। এখানে আপনার switching এর জন্য উচ্চক্ষমতাসম্পন্ন semiconductor যন্ত্র লাগবে যেমন thyristor অথবা power transistor, তারা খুব উচ্চ (high) ভোল্টেজে খুব বেশি বিদ্যুৎকে switch করতে পারে। উদাহরণস্বরূপ বলা যায়, কয়েকশো ampere এর বিদ্যুৎকে switch করা যায়।

ভেতরে কিছু বৈদ্যুতিন circuitry থাকবে, যা কিছু উপযুক্ত ইনপুটকে প্রতিক্রিয়া জানাবে যা control 0 বা 1 প্রেরণ করবে, এবং এই thyristor বা power transistor যেটাই থাকুক, তাকে switch on বা off করবে, এবং সেটাই যে যন্ত্রকে নিয়ন্ত্রণ করা হচ্ছে, সেটিকে power on বা off করবে।

(Refer Slide Time: 18:17)



এইভাবে solid state relay কাজ করে। এগুলি solid state relay এর কিছু ছবি। আপনি দেখুন এটি হলো মধ্যবর্তীটি, এটা বলে যে এটা 24 থেকে 380 volts AC এর মধ্যে কাজ করে। এরা আকারে খুবই ছোট কারণ semiconductorটি যে solid state device দিয়ে তৈরি, তারাও খুবই ছোট। বামদিকে আপনি যেটা দেখতে পাচ্ছেন সেটা আকারে 4 সেন্টিমিটার হয়।

এর সাথেই আমরা বিভিন্ন ধরনের আউটপুট যন্ত্র যেমন LED ও LDR, বিভিন্ন ধরনের sensor ও actuator নিয়ে আমাদের আলোচনার শেষে এসে পড়েছি। এখন প্রকৃত হাতে-কলমে ও প্রদর্শনীর session গুলির সময়, আমরা আপনাকে এইসব sensor ও actuator দেখাবো। কিভাবে ব্যবহার করতে হয়, কিভাবে interface করতে হয় ও কিভাবে microcontroller এ program লিখে এদের আপনার ইচ্ছা মত নিয়ন্ত্রণ করতে হয়। আমাদের পরবর্তী বক্তৃতা গুলিতে আমরা এই নিয়ে আলোচনা করব।

ধন্যবাদ।

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Lecture – 18
Microcontroller Development Boards

এই কোর্সের(course) ঢার নম্বৰ সপ্তাহে আপনাকে স্বাগত। এই সপ্তাহে আমি আপনাকে বিভিন্ন microcontroller board এর কাছে নিয়ে যাবো, বিশেষত দুটি বোর্ড(board) আমরা আলোচনা করব; একটি STM board ও অন্যটি Arduino board. এবং আমি দেখাবো কিভাবে আপনি এই দুটি বোর্ড(board) ব্যবহার করে program করতে পারেন।

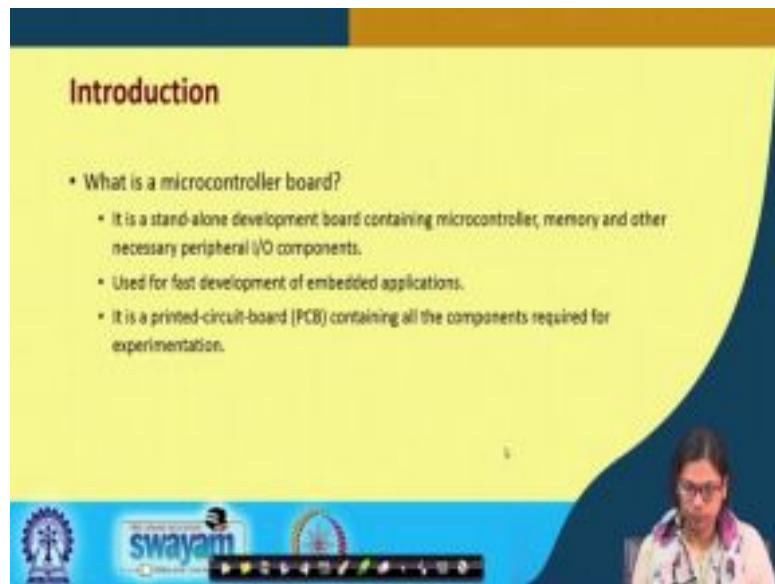
এছাড়াও অন্য development board হয়, কিন্তু এই কোর্সে আমরা এই দুটি নির্দিষ্ট বোর্ড(board) কে বিবেচনা করব এবং এই দুটি বোর্ডের বিভিন্ন পরীক্ষার ব্যাপারে আমরা আলোচনা করব।

(Refer Slide Time: 01:09)



এই বক্তৃতায় এই রকমের ধারণা গুলি নিয়ে আলোচনা হবে। আমি microcontroller board নিয়ে বলবো, আমরা ইতিমধ্যেই জানি microcontroller কি। আমি দুটো বোর্ড নিয়ে আলোচনা করব; একটি হল STM32F401 board আরেকটা হল Arduino UNO. এই কোর্সে আমরা যেসব পরীক্ষা করব তা মূলত এই দুটি বোর্ডের উপর ভিত্তি করবে।

(Refer Slide Time: 01:44)



এখন আমি microcontroller board কি, এ ব্যাপারে বলব। এর সংজ্ঞা হিসেবে বলা যায় যে এটি একটি স্বতন্ত্র(standalone) development board যাতে microcontroller, memory এবং অন্যান্য প্রয়োজনীয় peripheral I/O অংশবিশেষ থাকে। এটা কেন ব্যবহার হয়? এটা ব্যবহৃত হয় embedded applications দ্রুত গঠনের জন্য। এবং এটা সাধারণত printed circuit board এর উপর তৈরি হয় যাকে PCB বলে, যার মধ্যে পরীক্ষার জন্য প্রয়োজনীয় সব ধরনের অংশ থাকে।

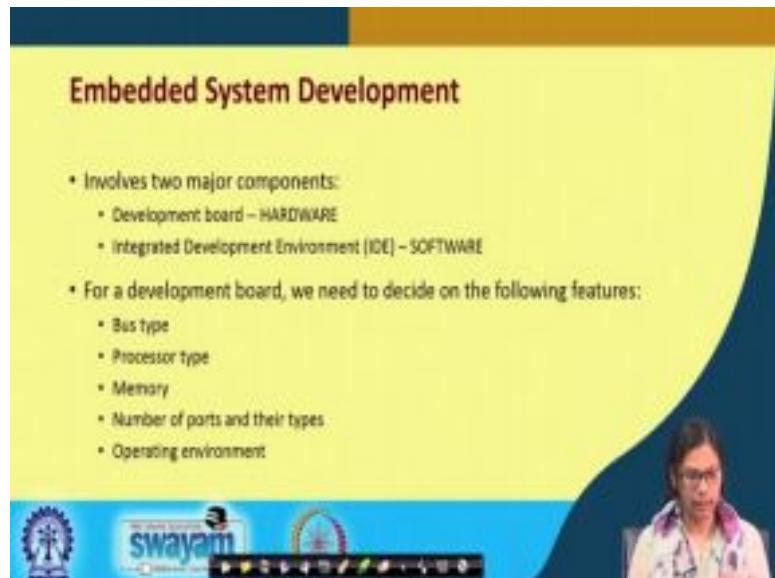
Microcontroller board গুলো ব্যবহৃত হয় embedded applications দ্রুত গঠনের জন্য। আজকালকার দিনে আমরা অনেক embedded applications দেখি। যদি আপনি একটা ছোট উদাহরণের কথা ভাবেন, যেমন smoke detector, এটা কি? এটা এমন একটা যন্ত্র যেটা ধোঁয়া সনাক্ত করে, যে এটা কোন একটি নির্দিষ্ট ঘর বা জায়গায় আছে কিনা। সে ক্ষেত্রে আপনার কি প্রয়োজন? আপনার বিভিন্ন ইনপুট আউটপুট যন্ত্র দরকার।

আরো নির্দিষ্টভাবে আপনার একটা microcontroller board দরকার যেটা দিয়ে আপনি কিছু analog মান(value) পড়ার জন্য একটি sensor যুক্ত করবেন, যেটা ঘরের ডের ধোঁয়ার পরিমাণ, কোন একটা digital মানে রূপান্তর করবে। এবং এরপর আমরা কিছু operation করব এটা খুঁজতে যে ঘরের মধ্যে ধোঁয়া আছে কি নেই। সেই শনাক্তকরণের উপর ভিত্তি করে বিভিন্ন রকমের কাজ করা যেতে পারে; একটা কাজ হলো যে একটা buzzerকে on করতে পারেন, এবং তাতে আপনি শব্দটা পাবেন এবং বুঝতে পারবেন।

Microcontroller boardগুলো এত ক্ষমতা সম্পন্ন হয় যে সমস্ত ইনপুট আউটপুট port এর

সাথে দেয়া থাকে, যাতে এটাকে আপনি সরাসরি sensor-এর সাথে যুক্ত করতে পারেন, বা **buzzer** টির সাথে, যাতে সম্পূর্ণ যন্ত্রটি তৈরি হয়।

(Refer Slide Time: 04:48)



যখন আমরা এই **embedded system** তৈরি নিয়ে কথা বলব, এটাতে দুটি প্রধান অংশ থাকে। একটি হল development board, যেটা প্রযোজনীয় hardware, এবং আরেকটি হল Integrated Development Environment, তাকে আমরা IDE বলি, সেটা হল software.

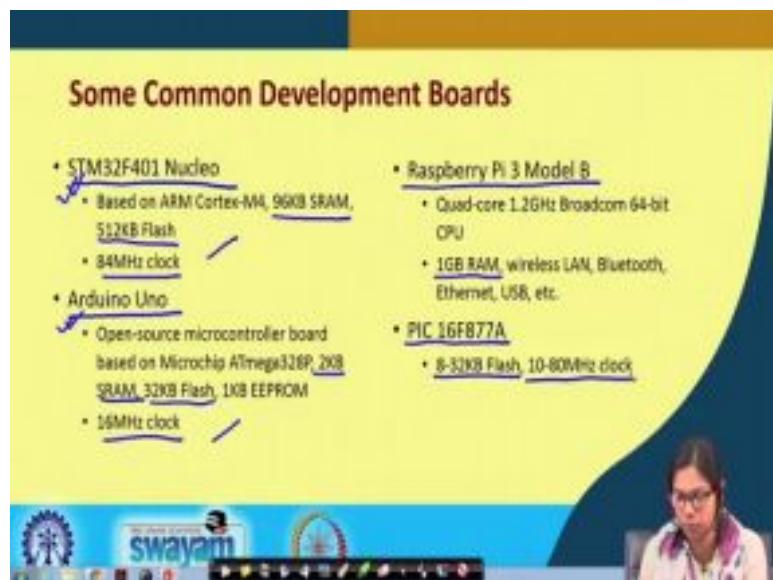
সুতরাং, যখন আমরা একটি system এর design করার কথা ভাবছি, এই দুটি জিনিস দরকার। আমাদের একটা hardware দরকার সেটা board টা, কিন্তু এটা আমরা কিভাবে program করব? আমাদের এর সাথে কিছু software জড়িত থাকা দরকার সেটা দিয়ে আমরা এটা program করতে পারব। আমরা এছাড়াও দেখব বিভিন্ন ধরনের কি কি IDE আছে যেটা আমরা এই দুটি board এর জন্য ব্যবহার করব। এখন যখনই আমরা একটি system কে design করতে চেষ্টা করব, আমরা সবদাই ভাববো কি ধরনের development board আমাদের ব্যবহার করা উচিত।

সে দিক থেকে দেখলে, কি development board ব্যবহার করব তা নিম্নলিখিত বৈশিষ্ট্য গুলোর উপর নির্ভর করে; একটি হলো bus ধরনের, আরেকটি processor ধরনের। কি ধরনের processor আমরা ব্যবহার করছি? Memory; memory হল অন্যতম গুরুত্বপূর্ণ parameter, যেটা যেকোনো ধরনের design এ দরকার, কারণ যদি আপনি একটি খুব বড়

program ৱার্থতে চান আৱ আপনার সীমাবদ্ধ memory থাকে, তাহলে আপনাকে ভাবতে হবে সেটা আপনি কিভাবে কৱিবেন।

সুতৰাং, যখন আপনি একটা **embedded system** তৈরি কৱিছেন, **port** সংখ্যা এবং তাদেৱ প্ৰকাৱ ছাড়াও একটা খুব গুৰুত্বপূৰ্ণ জিনিস যেটা জানতে হবে সেটা হল **memory**. আমাদেৱ ইনপুট আউটপুট **port** দৱকাৱ; আমাদেৱ এনালগ **port** ও দৱকাৱ। আমাদেৱ ঠিক কৱিতে হবে কি ধৱনেৱ **port** ওই নিৰ্দিষ্ট board-এতে থাকে ও operating environment কি হবে। কিছু development board এ আমৱা কিছু integrated development environment ব্যবহাৱ কৱি, কিন্তু কিছু ক্ষেত্ৰে compiler online-এ উপলব্ধ থাকে। শুধুমাত্ৰ প্ৰয়োজন হল যেন আপনার internet connection থাকে কাৱণ আপনাকে online compiler ব্যবহাৱ কৱিতে হবে। Development board এ আপনার code কে compile কৱিতে আপনার এই নিৰ্দিষ্ট software দৱকাৱ।

(Refer Slide Time: 08:07)



এগুলি কিছু সাধাৱণ development board. আমৱা ব্যবহাৱ কৱিব **STM32F401 Nucleo board**, এৱে ভিত্তি হলো ARM Cortex-M4, এবং এতে 96 KB SRAM আছে। এতে থাকবে 512 KB Flash memory, clock speed হবে 84 MHz. আৱেকটি জনপ্ৰিয় board হল **Arduino UNO**, যেটা open source microcontroller board, এৱে ভিত্তি হলো Microchip ATmega328P; এতে 2 KB SRAM আছে ও 32 KB হল Flash, এতে আৱও আছে 1 KB EEPROM. এতে একটি 16 MHz clock আছে।

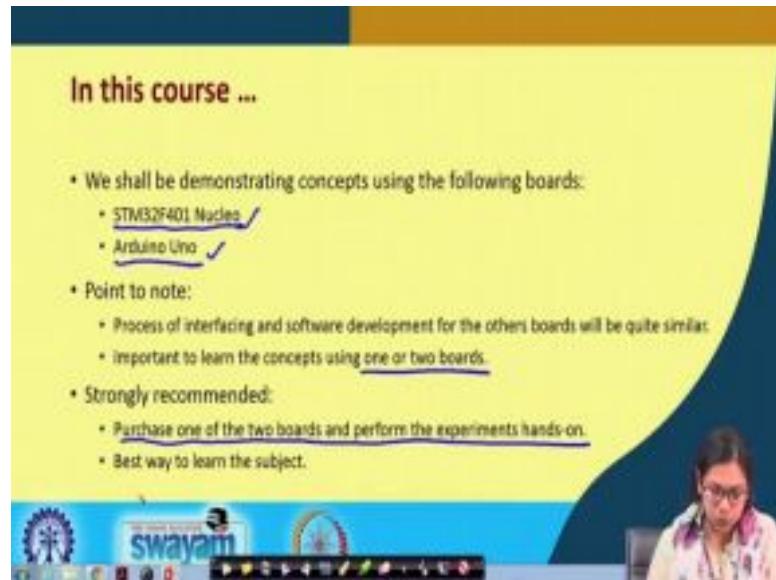
আপনি যদি STM board ও Arduino UNO এর বৈশিষ্ট্য গুলো দেখেন, আপনি পরিষ্কার বুঝবেন STM হল Arduino এর থেকে অনেক ক্ষমতা সম্পন্ন। এতে মেমোরি হল 96 KB 2 KB এর তুলনায়, 512 KB হল flash 32 KB flash এর তুলনায়, clock speed হল 84 megahertz 16 megahertz clock speed এর তুলনায়।

এখন আমার আপনাকে একটা গুরুত্বপূর্ণ জিনিস বলা দরকার। যে application গুলো দেখানো হবে সেটা Arduino UNO এবং STM board দুটোতেই run করানো যায়। আমাদের তৈরির জন্য যেকোনো একটি বোর্ড(board) আমরা ব্যবহার করতে পারি।

কিন্তু যখন আপনি খুব বড় একটি application এর কথা চিন্তা করছেন বা তৈরি করতে চেষ্টা করছেন, যাতে অনেক বেশি memory দরকার, সেই ক্ষেত্রে STM board কে পছন্দ করা হয়। এখন, আরো বেশি ক্ষমতা সম্পন্ন board আছে যেমন Raspberry pi 3 model B, যাতে আছে quad core 1.2 GHz Broadcom 64-bit CPU, এর সাথে এতে আছে 1 GB RAM, wireless LAN, Bluetooth, Ethernet ও USB. আপনি দেখছেন একে computer হিসেবে ব্যবহার করা যেতে পারে। এতে খুব উন্নত ধরনের বৈশিষ্ট্য আছে যাতে sophisticated application development তৈরি করা যায়, যদিও এইনিদিষ্ট course টির ক্ষেত্রে আমরা Raspberry Pi ব্যবহার করব না।

আরেকটি microcontroller board হল PIC microcontroller board. 16F877A হল এর অন্যতম একটি মডেল(model); এর বৈশিষ্ট্য এইরকম। এতে 8 থেকে 32 kb অবধি flash আর clock speed হবে 10 থেকে 80 MHz. সুতরাং, যখন আপনি খুব সহজ ধরনের কোন application তৈরীর কথা ভাববেন, আপনি PIC based microcontroller systems ব্যবহারের কথা ভাববেন।

(Refer Slide Time: 12:53)



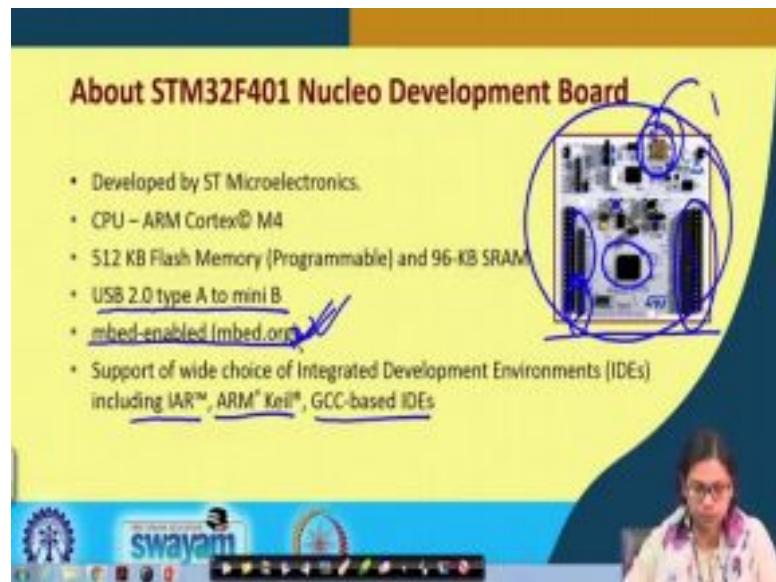
In this course ...

- We shall be demonstrating concepts using the following boards:
 - STM32F401 Nucleo
 - Arduino Uno
- Point to note:
 - Process of interfacing and software development for the other boards will be quite similar.
 - Important to learn the concepts using one or two boards.
- Strongly recommended:
 - Purchase one of the two boards and perform the experiments hands-on.
 - Best way to learn the subject.

যেমন আমি আপনাকে বলেছি, এই নির্দিষ্ট কোর্স-এ আমরা **নিম্নলিখিত board** গুলোর ব্যাপারে দেখাবো। প্রথমটি হলো **STM32F401 Nucleo board** এবং আরেকটি হলো **Arduino UNO**। এখন **interfacing** এর পদ্ধতি ও অন্য **board** এর জন্য **software development** প্রায় একই রকম হবে। যেহেতু আমরা **STM board** ও **Arduino UNO** নিয়ে আলোচনা করছি, যদি আপনি অন্য কোন **board** এ **interface** করতে চান তাহলেও পদ্ধতি আলাদা হবে না।

অন্তত একটি বা দুইটি **board** নিয়ে ধারণাটি শেখা গুরুত্বপূর্ণ, যাতে আপনি প্রাথমিক পদ্ধতি জানতে পারেন। এই কোর্সের(course) জন্য আপনাকে পরামর্শ হলো অন্তত একটি বা দুটি **board** কিনুন এবং পরীক্ষাগুলি করুন। আমরা এই নির্দিষ্ট কোর্সটিতে অনেক পরীক্ষা হাতে-কলমে করব। আপনার এগিয়ে যাওয়া উচিত এবং এই বোর্ডগুলির কমপক্ষে একটি দ্রয় করা উচিত এবং আপনি আমাদের সাথে পরীক্ষাগুলি করতে পারেন।

(Refer Slide Time: 14:53)

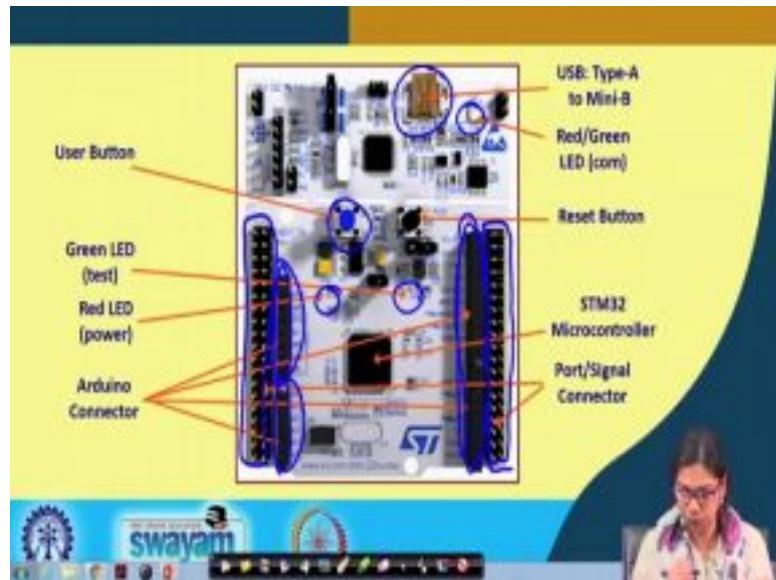


এখন আমি STM32F401 Nucleo development board নিয়ে এগিয়ে যাব। এই বোর্ড(board)টা আমরা ব্যবহার করব। এতে কিছু digital pin আছে, কিছু analog pin, একটা power, এটা একটা USB connection যার মাধ্যমে আপনি USB port কে যুক্ত করতে পারবেন। এই বোর্ড(board) ST microelectronics নামের একটি কোম্পানি তৈরি করেছে। এই নির্দিষ্ট বোর্ডের মধ্যে থাকা CPU হলো ARM Cortex-M4; এতে 512 KB programmable flash memory এবং 96 KB SRAM আছে। আমি আপনাকে আগে যেমন বলেছি যে এটি হল USB connector. এখানে আছে USB 2.0 port.

এখন থেকে আপনার PC কে যুক্ত করতে এই USB টা দরকার। এই নির্দিষ্ট বোর্ডটি(board) mbed-enabled. সূতরাং, আমরা যেটা করি, আমরা একটা ওয়েবসাইটে যাই mbed.org, আমি আপনাকে এটার প্রকৃত URL টা দেব, সেখানে আমরা code লিখব এবং code টা কে এই নির্দিষ্ট বোর্ডের রাখবো। এবং আপনি এই IO pin গুলো দিয়ে যুক্ত করতে পারেন বা আপনি এনালগ pin দিয়ে যুক্ত করতে পারেন, আপনি অন্যান্য নানা রকম কাজও করতে পারেন।

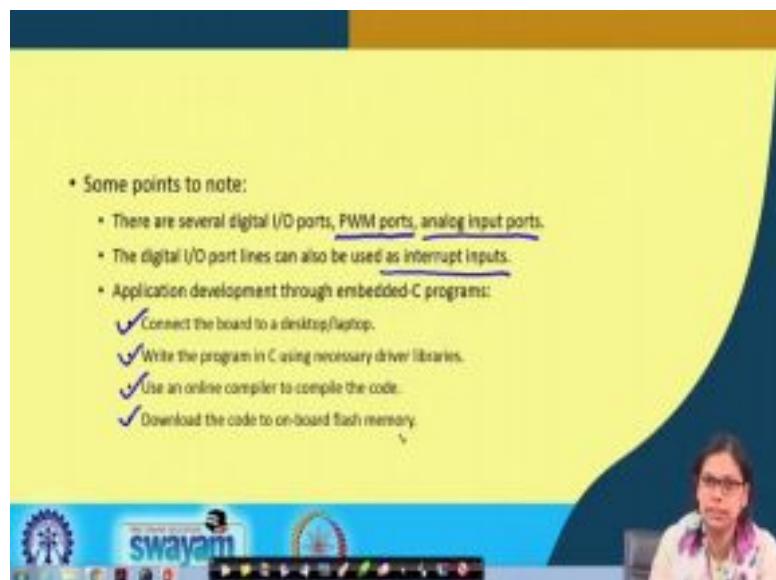
এই নির্দিষ্ট কোর্সে(course) আমরা online mbed compiler ব্যবহার করেছি। এখানে বিস্তৃত ধরনের IDE রয়েছে, সূতরাং আপনি ARM Keil বা IAR সহ GCC নির্ভর যে কোনো IDE ব্যবহার করতে পারেন।

(Refer Slide Time: 17:49)



এইটা এই board এর ছবি। এখানে একটা নীল রঙের button আছে যেটা user button, এটা হল reset button যা যন্ত্রিকে reset করতে ব্যবহার করা হয়। আমি এই connector টা নিয়ে কথা বলছিলাম, type A থেকে mini B, আর এখানে একটা লাল/সবুজ LED আছে। যখনই আপনি এই USB এর মাধ্যমে একটি code এখানে রাখবেন, আপনি দেখবেন এই লাল/সবুজ LED টা জ্বলবে। এগুলি Arduino connectors এবং এখানে অন্য port রয়েছে। আপনি এই port গুলি দেখছেন; এই port গুলি মূলত signal connector যা আমরা ব্যবহার করতে পারি। আপনি দেখবেন এই connector গুলি ছাড়াও অন্য অনেক connector আছে যা আমরা ব্যবহার করতে পারি, কিন্তু আমাদের পরীক্ষা-নিরীক্ষায় আমরা মূলত এই Arduino connector গুলো ব্যবহার করেছি।

(Refer Slide Time: 19:59)



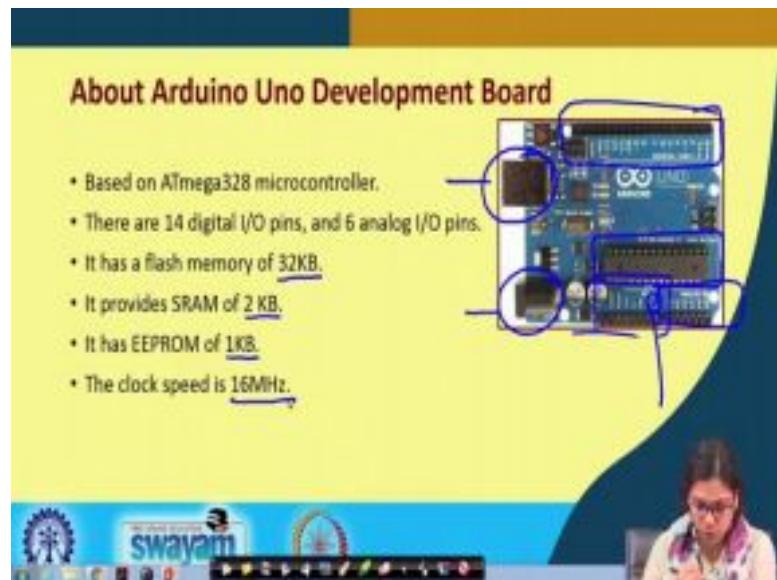
আমরা জানি কয়েকটি digital IO port আছে, এখানে PWM বা Pulse Width Modulation enabled port ও আছে, আর আছে analog input port. Digital I/O port line গুলোকে

interrupt input হিসেবে ব্যবহার করা যেতে পারে; এগুলো পূর্বেই আলোচিত হয়েছে।

আমি আগেই USB connector দেখিয়েছি যেটা দিয়ে আপনাকে desktop বা laptop এর সাথে যুক্ত করতে হবে, এরপর আপনাকে id টা open করতে হবে আর আপনাকে program লিখতে হবে।

আপনি উপস্থিত প্রয়োজনীয় driver library গুলো ব্যবহার করে C তে program লিখবেন, এরপর online compiler ব্যবহার করে আপনি code টা compile করতে পারেন। আপনি code টা compile করলেই এই নির্দিষ্ট code টা download হয়ে যাবে, এটা আপনি আপনার download folder-এ খুঁজে পাবেন। এরপর আপনি ওই নির্দিষ্ট codeকে copy করতেও device এ রাখতে পারবেন। আমি আপনাকে এই নির্দিষ্ট জিনিস টি দেখাবো যেটা আমি এইমাত্র বললাম। তবে আপনি যখন এই নির্দিষ্ট STM বোর্ড ব্যবহার করছেন তখন এই নিম্নলিখিত পদক্ষেপগুলি অনুসরণ করা দরকার।

(Refer Slide Time: 22:01)

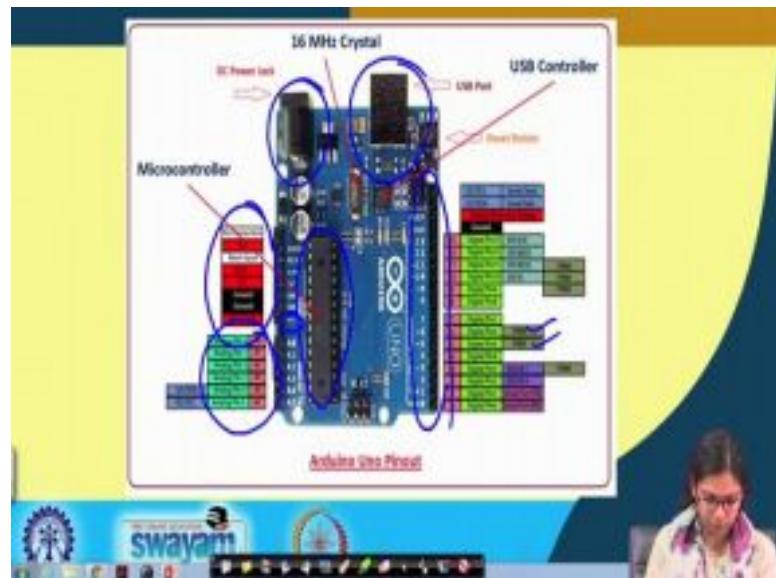


এখন আমরা Arduino UNO development board-এ যাব। এটি হলো Arduino UNO development board; আপনি দেখতে পারেন এগুলি হল digital এবং PWM ports, এটি analog port, এটি power এর জন্য, এটি হল microcontroller chip. আপনি এটিতে power দেওয়ার জন্য USB connector ব্যবহার করতে পারেন, অথবা একটা 9-volt ব্যাটারি ব্যবহার করেও আপনি এতে power দিতে পারেন।

Arduino UNO development board টা ATmega 328 microcontroller এর উপর ভিত্তি করে বানাগো। এতে চোদ্দোটা ডিজিটাল ইনপুট আউটপুট pin আছে, এদের প্রতিটিকে input বা output pin হিসেবে program করা যায়। আর এতে ষেল্টা এনালগ ইনপুট pin আছে, এর

flash memory 32 KB, এটা 2KB SRAM দেয়, এতে EEPROM আছে 1KB, আর এর clock speed 16 MHz. এইগুলি এই Arduino board এর বৈশিষ্ট্য।

(Refer Slide Time: 23:53)



আমি আপনাকে যেমন বলেছিলাম এই USB port দিয়ে আপনি PC-এর USB portকে যুক্ত করতে পারবেন। এইটা microcontroller যা আমরা ব্যবহার করছি, এগুলি analog pin, এগুলো power এর জন্য। আর আপনি যদি দেখেন এগুলো digital pin, কিন্তু কিছু digital pin আবার PWM, এই PWM কি এটা আমরা নির্দিষ্ট সময়ে দেখব।

আপনি দেখবেন একটা DC power jackও থাকে যেটা দিয়ে আপনি পুরো board-এ power দিতে পারবেন।

(Refer Slide Time: 25:02)

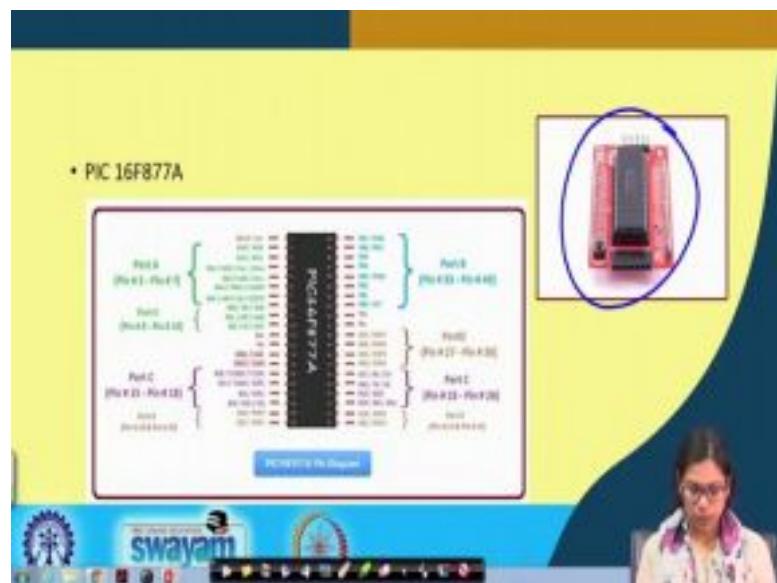


আমি আপনাকে আগে যেমন বলেছি এছাড়াও অন্য development boardও থাকে, এদের একটি হল

Raspberry Pi 3 model যেটা আরো পরিশীলিত(sophisticated)। এতে দুটো USB port থাকে; এটা হল network এর জন্য, এগুলো হল audio ও video.

এটিতে একটা HDMI port ও আছে, একটা micro USB power point আছে, এটি হলো display port. আপনি একটা micro SD এই পিচনের দিকে রাখতে পারেন, এটা bluetooth সক্ষম(enabled), এটি হলো CPU, memory, এবং এই pin গুলো port হিসেবে ব্যবহার করা যেতে পারে।

(Refer Slide Time: 26:02)



PIC boardএ সাধারণত এই pin গুলি থাকে আর এগুলো তাদের কিছু বৈশিষ্ট্য। এটা হল port A; port A হল pin সংখ্যা 2 থেকে pin সংখ্যা 7, port E হলো pin সংখ্যা 8 থেকে 10, এইভাবে। pin গুলির প্রত্যেকটির কিছু নির্দিষ্ট কার্যকারিতা আছে এবং এই নির্দিষ্ট pin-এর বিভিন্ন বৈশিষ্ট্য আছে। এটা একটা ছবি যা PIC 16F877A microcontroller দেখাচ্ছে, এটা সাধারণত এই রকম দেখতে লাগে।

এই pin গুলো দিয়ে আপনি যুক্ত থাকতে ও programming করতে পারেন।

আমরা আসলে এই বক্তৃতাটির শেষে এসে গেছি আর আমরা দুটো প্রয়োজনীয় board নিয়ে আলোচনা করেছি; একটা হল STM board আরেকটা হল Arduino UNO. কিন্তু এছাড়াও অন্য board কেমন দেখতে সেটা মোটামুটি ভাবে আপনাকে বলেছি। পরবর্তী সপ্তাহে আমি আরো বিশদে STM board কে নিয়ে মনোনিবেশ করব।

ধন্যবাদ।

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Lecture – 19
Mbed C Programming Environment

পৰবৰ্তী বক্তৃতায় স্বাগতম। এই বক্তৃতায় আমি mbed C Programming Environment নিয়ে কথা বলবো। আমি mbed C Programming Environment নিয়ে শুরু করবো আৱ আমি আপনাকে দেখাবো প্ৰকৃতপক্ষে কিভাবে আপনি এই environment ব্যবহাৰ কৰে program কৰতে পাৰেন।

(Refer Slide Time: 00:53)



যেমন আমি বলেছি, আমি mbed C environment এৱং সূচনা কৰাৰ এৰং আমি আপনাকে program গুলি তৈৱি, compile ও run কৰাৰ ধাপ গুলি দেখাবো।

(Refer Slide Time: 01:07)

What is Mbed C?

- It is an embedded software development platform.
 - Users can develop the software using C using microcontroller board specific library.
 - The program is compiled and downloaded onto the development board.
- We shall explain the Mbed-C platform using the example of STM32F401 Nucleo ARM Development Board.

প্রথমত, Mbed C কি? এটা একটি embedded software development প্ল্যাটফর্ম যা ব্যবহার করে ব্যবহারকারীরা C দিয়ে সফটওয়্যার(software) তৈরি করতে পারে, microcontroller board specific library দিয়ে program টা development board এ compile এবং download করা হয়। আমরা STM32F401 Nucleo ARM Development Board এর উদাহরণ ব্যবহার করে platform টা ব্যাখ্যা করব। এই নির্দিষ্ট board টা ব্যবহার করতে গেলে আপনাকে register করতে হবে, এবং আপনাকে কিছু কার্যকারিতা ইত্যাদি, যেগুলো সেখানে আছে সেগুলো দেখতে হবে।

(Refer Slide Time: 02:19)

Requirements to Begin

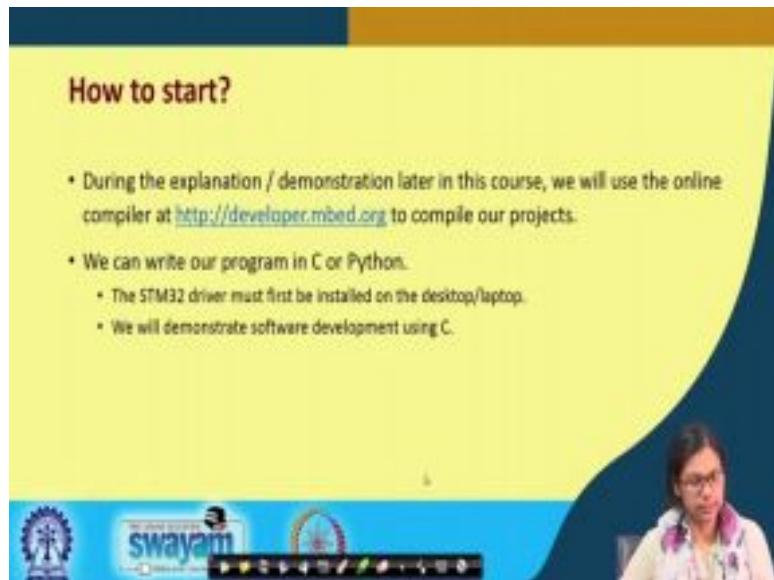
- STM32F401 development board and driver
- USB mini to USB Type B connector
- Development environment (IDEs or [mbed account](#))

The development environment runs on a desktop / laptop.

প্রথম প্রয়োজনীয়তা হল যে আপনার এই বোর্ডটা(board) ঠিক জায়গাতে driver install করা অবশ্য থাকতে হবে আপনার USB-mini থেকে USB-typeB connector দরকার। আপনি এই অংশটা এখানে যুক্ত করবেন এবং এইটা আপনার PC বা ল্যাপটপে যুক্ত হবে। প্রথমত এইটা আপনার

করা দরকার। এবং development environment এর জন্য আমি যেমন বলেছি আমরা একটা online compiler ব্যবহার করব যা হল mbed. সুতরাং আপনাকে mbed এ একটা একাউন্ট বানাতে হবে।

(Refer Slide Time: 03:24)



এখন, কিভাবে শুরু করব? এই কোর্সের পরবর্তীতে প্রদর্শনের সময় আমরা **online compiler** ব্যবহার করব যেটা আমাদের **project** গুলি **compile** করার জন্য <http://developer.mbed.org> এ উপস্থিত। আমরা আমাদের program C বা python এ লিখতে পারি, কিন্তু বিশেষ ভাবে আমরা C তে লিখব। STM32 driver টি প্রথমে desktop বা laptop এ install করতে হবে, এরপর আমরা এটা দিয়ে software development প্রদর্শন করব। অনুগ্রহ করে মনে রাখবেন যে USB cable লাগবে, যেখানে online compiler পাওয়া যায় আপনাকে সেখানে যেতে হবে এবং নিশ্চিত করতে হবে যে STM driver টা install করা আছে।

(Refer Slide Time: 04:26)

Installation of STM32 Driver

- To work with STM32 kit we need to install STM32 driver.
- STM driver can be downloaded from:
<http://www.st.com/en/development-tools/stsw-link009.html#getsoftware-scroll>
- For downloading you may need to provide basic details: e-mail, name etc.
- Driver will be downloaded as a ZIP bundle.
- Extract the downloaded zip file and install `dpinst_x86.exe` for 32-bit machine or `dpinst_x64.exe` for 64-bit machine.

STM driver-এর installation যথেষ্ট সোজা যেটা আপনাকে করতে হবে সেটা হল আপনাকে এই নির্দিষ্ট URL-টিতে যেতে হবে, এটা download করতে হবে, আপনাকে কিছু প্রাথমিক বিবরণ দিতে হবে। তখন driver টি একটা zip bundle হিসেবে download হবে, download করা zip ফাইলটা extract করতে হবে এবং 32-bit machine-এর জন্য এই নির্দিষ্ট ফাইলটি install করতে হবে, অথবা যদি এটা 64-bit machine হয় তবে এইটা download করতে হবে।

এইটা একটা গুরুত্বপূর্ণ পদক্ষেপ; এই নির্দিষ্ট যন্ত্রটা ব্যবহার করার আগে আপনাকে driver গুলি যথাযথভাবে install করতে হবে।

(Refer Slide Time: 05:37)

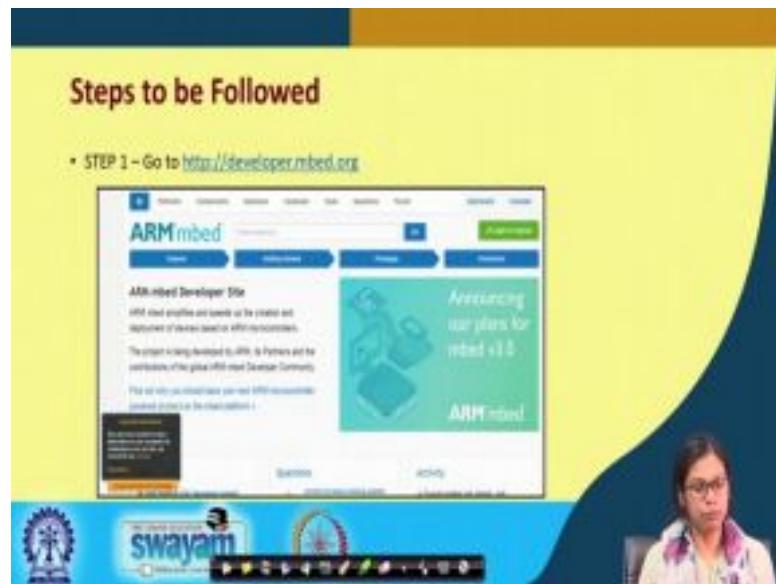
Checking the Installation

- Connect the STM32 kit to the desktop/laptop.
- Go to the device manager.
- Go to Ports (COM & LPT).
- Check which port number is used by STM kit.

In the following example, COM11 port is used.

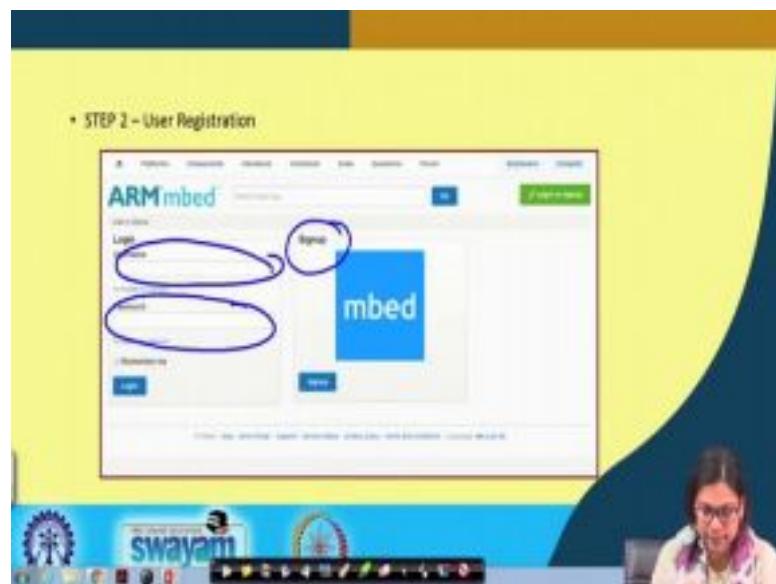
এরপর installation টা পরীক্ষা করব; যদি এটা আগেই হয়ে যায় এই STM32 kit টি desktop বা laptop এ যুক্ত করবো। আপনি device manager-এ যাবেন ও port সংখ্যা পরীক্ষা করবেন। যখনই driver টি সঠিকভাবে install হয়ে যাবে সেটা এখানে দেখানো হবে।

(Refer Slide Time: 06:40)



এখন, আর কি কি পদক্ষেপ নিতে হবে? প্রথম পদক্ষেপ হল আপনাকে developer.mbed.org এ যেতে হবে।

(Refer Slide Time: 06:54)



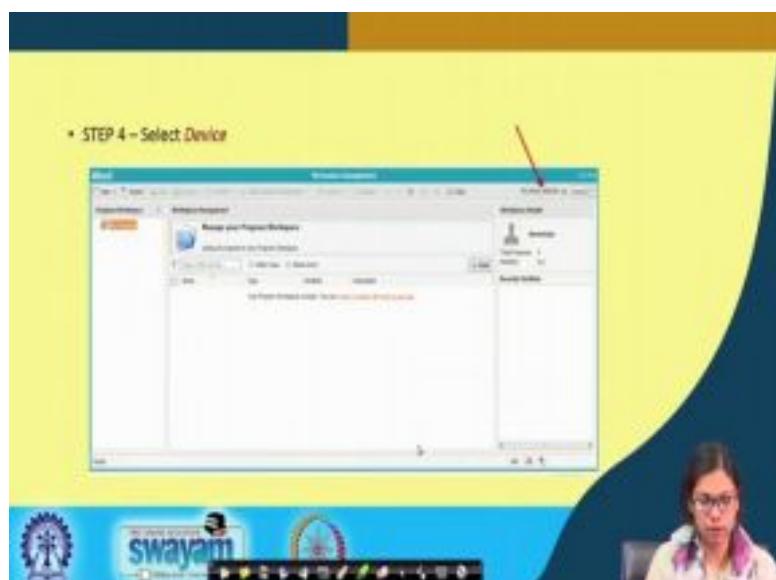
এরপর আপনি দেখবেন user name ও password লেখার একটা জায়গা আছে; অবশ্যই আপনাকে প্রথমে এটার জন্য signup করে নিতে হবে। এটা একটা সহজ পদ্ধতি যা আপনাকে মেনে চলতে হবে।

(Refer Slide Time: 07:20)



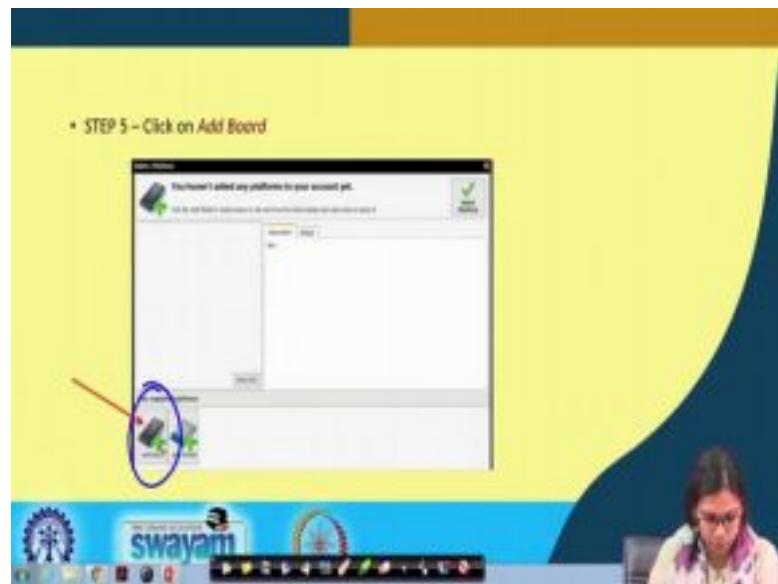
Login করার পর, আপনি আপনার বিবরণ দিন। আপনি এই রকম একটা screen পাবেন, যেখানে আপনি দেখবেন কিছু লেখা আছে, যেটা compiler; compiler এর উপর click করুন।

(Refer Slide Time: 07:46)



এবং আপনি এগিয়ে চলুন। প্রথমবার compiler এর উপর click করার পর, যারা করছে তারা এই সমস্যায় পড়বে, যেখানে লেখা আছে No Device Selected. হয় এটা STMF401RE অথবা অন্য কোন যন্ত্র। এরপর আপনাকে ওই যন্ত্র নির্বাচন করতে হবে।

(Refer Slide Time: 08:16)



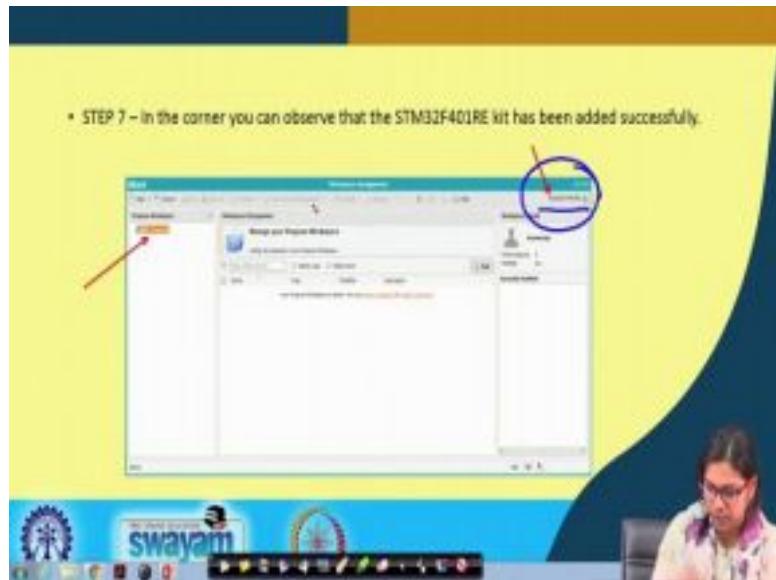
পরবর্তী পদক্ষেপ যেটা আপনাকে করতে হবে: Add board এ আপনি click করুন।

(Refer Slide Time: 08:35)



এরপর আপনি NucleoF401RE নির্বাচন করুন; এটাই সেই board যেটা আমরা ব্যবহার করছি। সুতরাং, আপনি যে board ই ব্যবহার করুন না কেন, আপনাকে ওই নির্দিষ্ট board টা নির্বাচন করতে হবে।

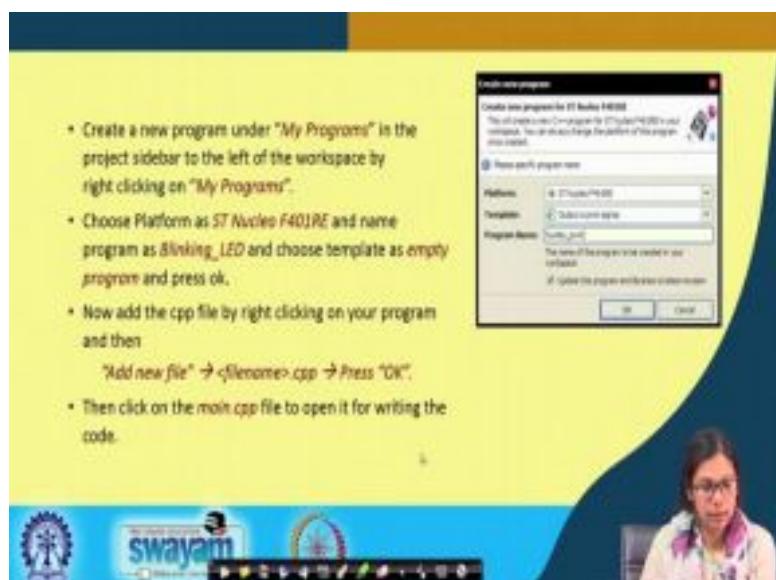
(Refer Slide Time: 09:00)



আমরা ওই board টা নির্বাচন করি ও এগিয়ে চলি। এবং যখনই আপনি এটা করবেন, দেখবেন ওই নির্দিষ্ট board নির্বাচিত আছে, আগে এটা দেখাচ্ছিলো no device selected. এখন, নিম্নলিখিত পদক্ষেপগুলো গ্রহণ করে আপনি দেখবেন NucleoF401RE নির্বাচিত হয়েছে।

আমরা এগিয়ে চলি; যখনই আপনি এই নির্দিষ্ট screen টা দেখবেন, তখন আপনিও আমার program গুলিও দেখতে পাবেন এবং আপনি দেখবেন যে এটা প্রায় ফাঁকা, কিন্তু আপনি যদি program লিখতে থাকেন এটা দেখবে। আপনাকে আমি আমার account দেখালে দেখবেন এখানে অনেক program লেখা আছে।

(Refer Slide Time: 09:46)

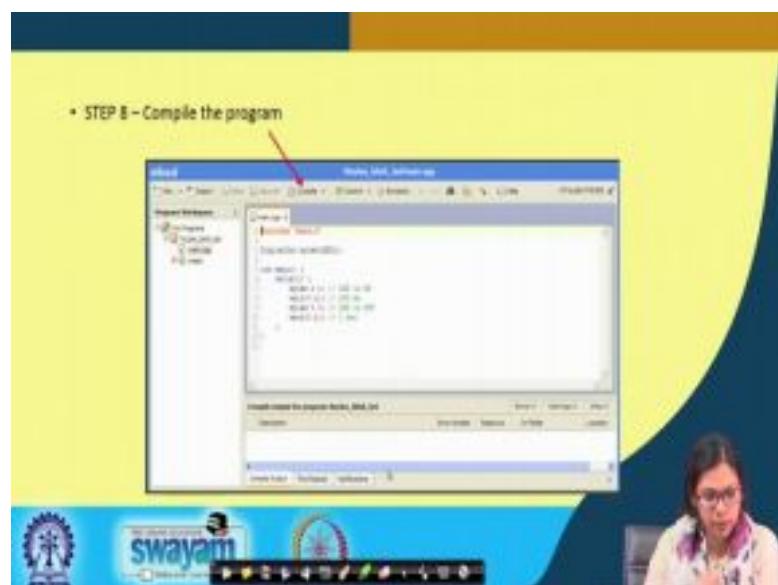


প্রথমে পদক্ষেপ হল 'my programs' এর মধ্যে একটা নতুন program, এটা project slide bar এ workspace এর বাম দিকে MyPrograms এ right click করে তৈরি করা হয়, এরপর আপনি একটি নতুন তৈরি করুন, program এর নাম দিন এবং একটি ফাঁকা template তৈরি করে, ok

টিপুন।

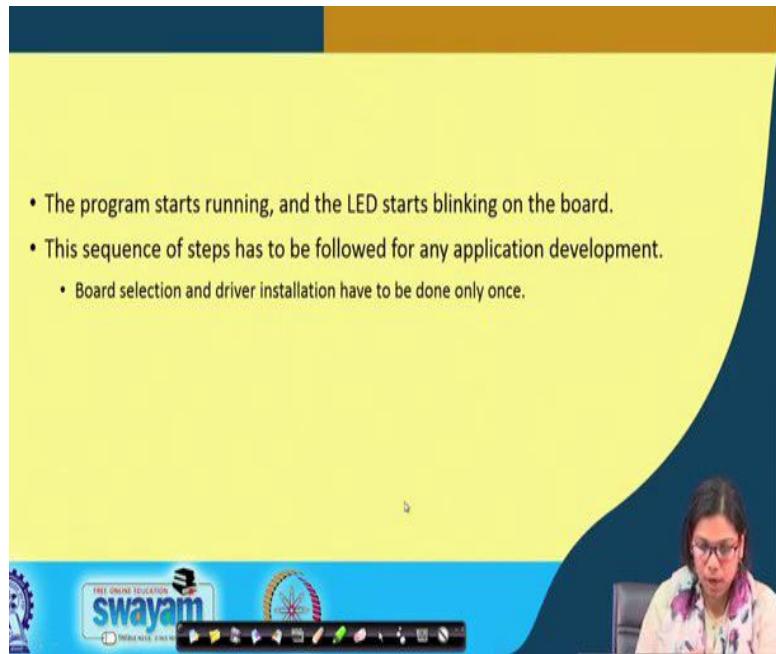
আপনি প্রথম প্রোগ্রাম(program) টি লেখার পর এই পদক্ষেপগুলি অনুসরণ করা দরকার। এখন আপনাকে কি করতে হবে? আপনাকে cpp file টা প্রোগ্রামটার ওপর right click করে যোগ করতে হবে ও একটি নতুন filename দেবেন। আমি এটাকে আরো কিছু slide শেষ করে এই tool এ নিয়ে যাব।

(Refer Slide Time: 10:50)



সুতরাং, এখানে আপনার main.cpp আছে। যখন আপনি এখানে click করবেন আপনি এই main.cpp program দেখবেন।

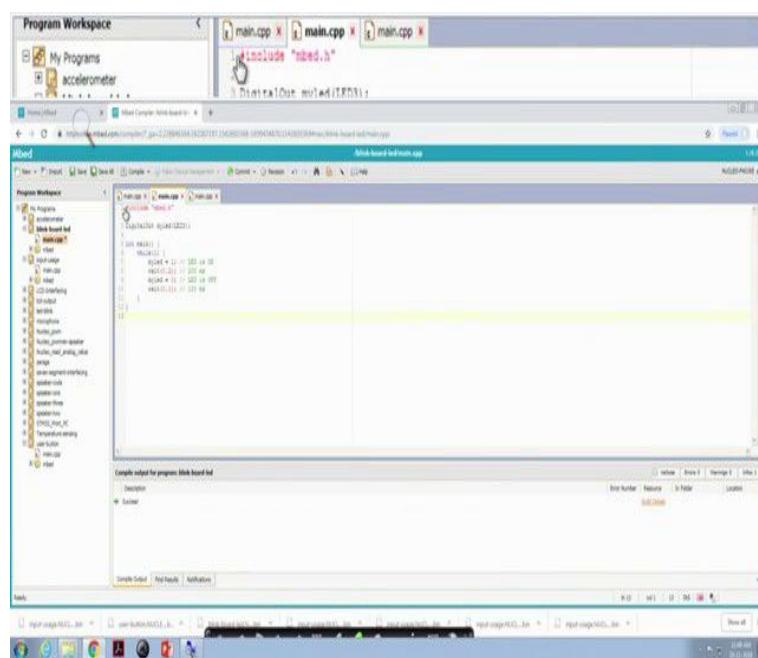
(Refer Slide Time: 11:2)



আর তারপর আপনাকে program টা compile করতে হবে। compile button টা এখানে উপরে দেখানো হয়েছে; আপনি compile button এ click করলে এটি compile হবে। আপনার compile করা হলে code টা download হবে, এটাকে Download folder এ save করে রাখবেন।

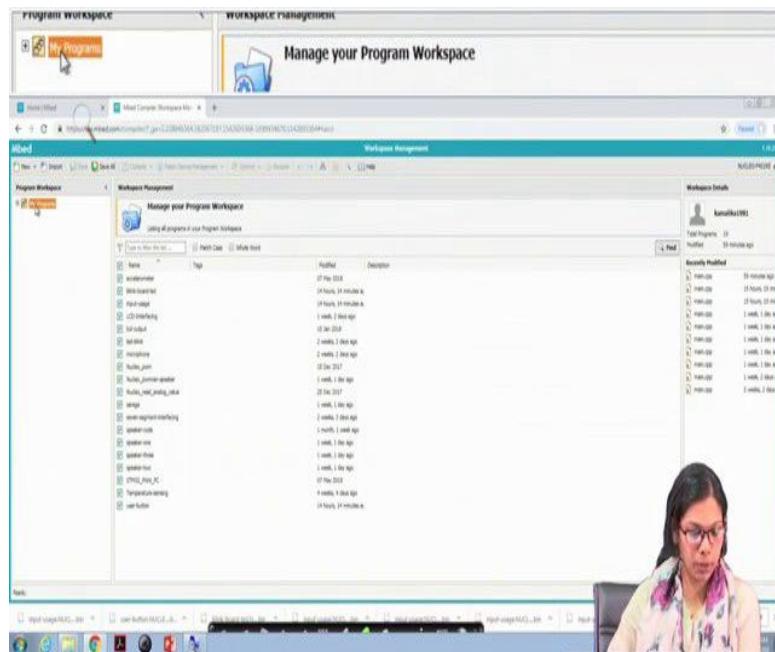
প্রোগ্রাম(program) টা run করতে শুরু করে এবং LED board-এর উপর স্বলতে-নিভতে শুরু করে। Application development এর জন্য এই পদক্ষেপগুলো অনুসরণ করতে হবে। board selection ও driver selection একবারই করতে হবে।

(Refer Slide Time: 12:31)



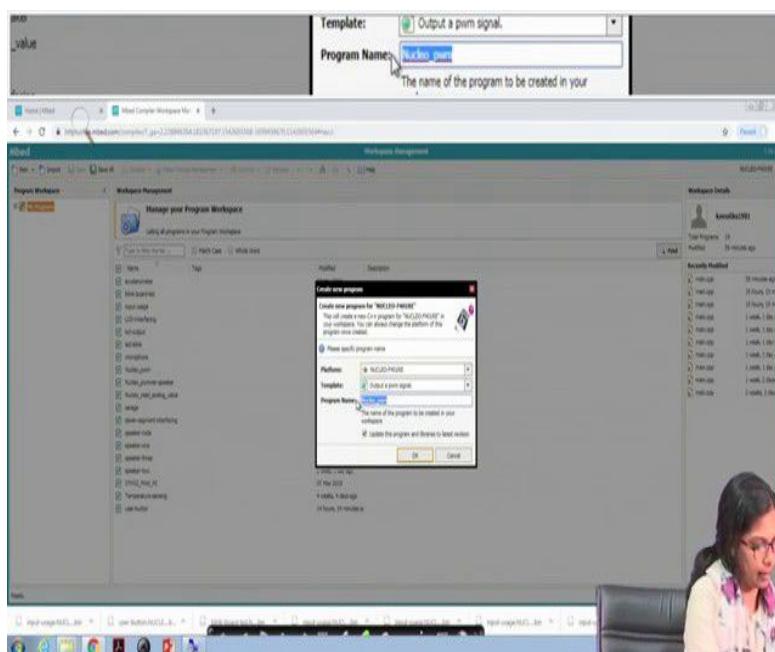
আপনি দেখুন এটা environment; এই সহজ program টি আমি দেখাচ্ছি। সুতরাং প্রথম পদক্ষেপের জন্য যা প্রয়োজনীয় তা আমি আপনাকে বলেছি।

(Refer Slide Time: 13:18)



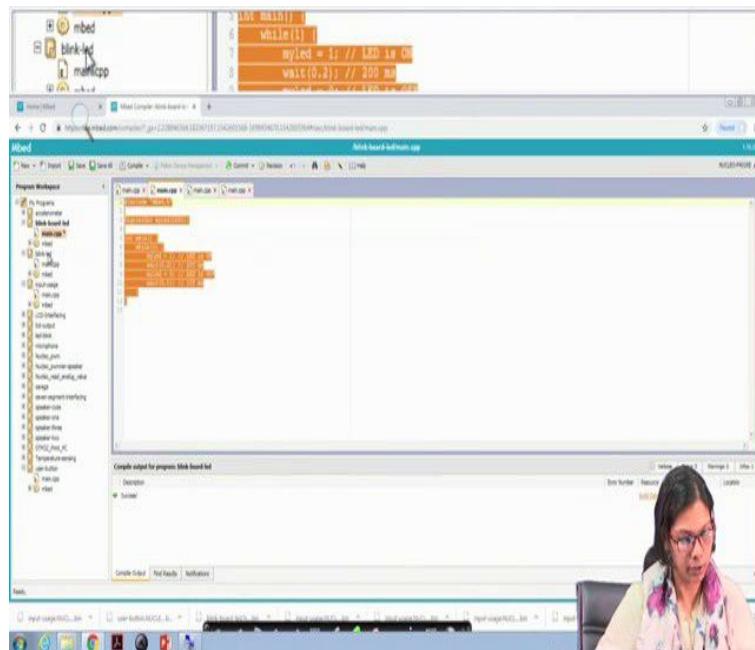
এইটা আমার program, আপনাকে main program-এ right click করতে হবে।

(Refer Slide Time: 13:23)



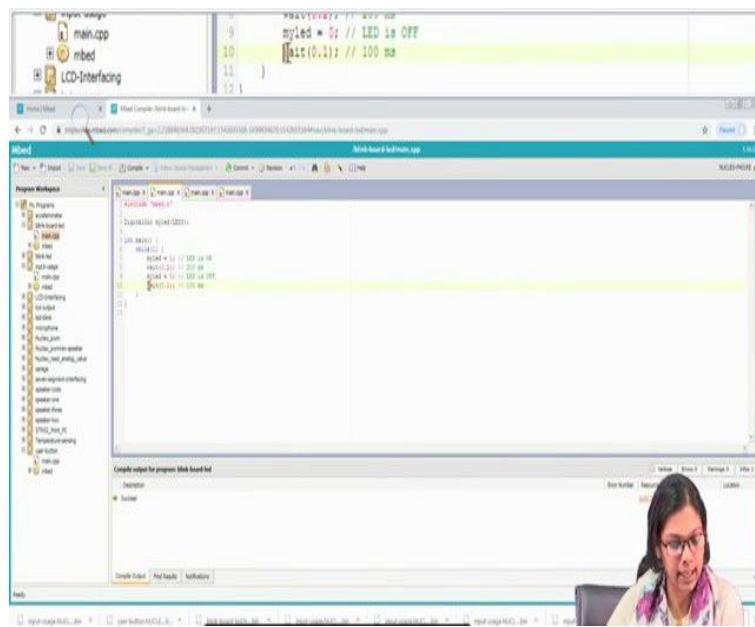
এবং এরপর আপনি দেখবেন এই ধরনের কিছু আসবে, আর আপনাকে এই program এর একটা নাম দিতে হবে, যেমন ধরলাম blinkLED এবং আমি Ok তে click করলাম।

(Refer Slide Time: 13:47)



যখন আমি এটা করলাম, আপনি এই নামটাতে double click করুন, আর আপনি কিছু sample code দেখুন। আমরা এই sample code টা কেটে নেব এবং এখনে নিজেদের code লিখব।

(Refer Slide Time: 14:17)



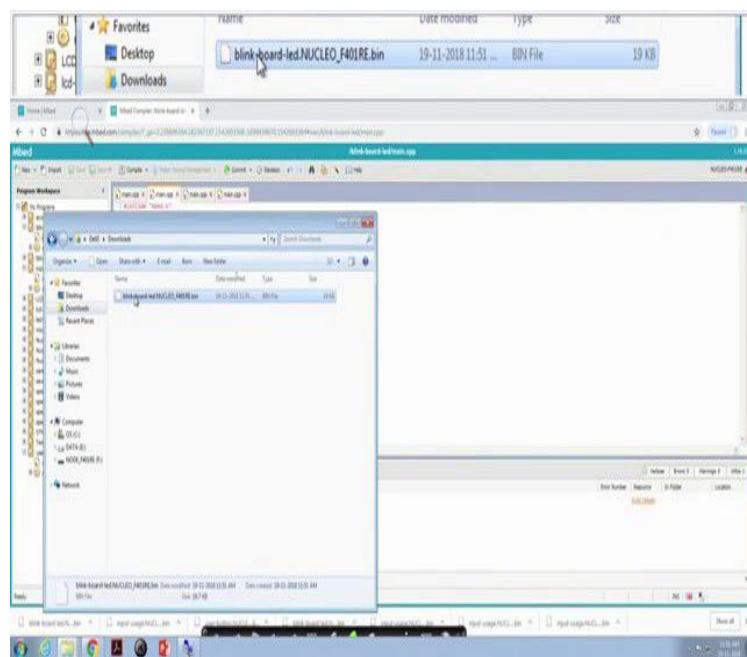
আমি এখনে click করলাম; এটা main.cpp যেমন আমি আপনাকে বলেছিলাম। আমি এটা save করব। এটা #include mbed.h -- এটা header file, আপনাকে এটা include করতে হবে। তারপর পরবর্তীতে আপনি যেটা করবেন সেটা হল DigitalOut myLED (LED3), এটা অন্যতম output pin. ওই STM board এ LED-র নাম হল LED3. প্রোগ্রামটি (program) ধারাবাহিকভাবে

চলছে ।

তো, এটা একটা সহজ code যা আমি লিখেছি। পরের slide-এ আমরা এই বিষয়ে আরো বিশদে দেখব।

এখন, আমি compile করার জন্য এই compile button ব্যবহার করব, এখন আপনি দেখতে পারছেন code টি compile হয়ে গেছে আর আমি আপনাকে সেটা বলছি, এটা Download folder এ save হবে ।

(Refer Slide Time: 16:16)



আপনি এটা copy করুন অথবা এখন থেকে cut করুন, এবং আপনি দেখতে পারবেন বামদিকে board টা আছে যেটা PC তে mount করা আছে; আপনি এখানে যাবেন আর paste করবেন।

যদি আপনি এই ধরনের কিছু করেন তাহলে board এর উপরের LED টা জ্বলতে শুরু করবে।

ধন্যবাদ।

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Lecture – 20
Interfacing with STM32F401 Board

এই বক্তৃতায় আমরা আপনাকে দেখাবো কিভাবে আমরা STM32F401 board কে Interface করতে পারি। আমি আপনাকে তিনটে program উদাহরণস্বরূপ দেখাবো যাতে আমি onboard LED, onboard switch ব্যবহার করব, আর আমি বিভিন্ন পিনের(pins) ইনপুট/ আউটপুট ব্যবহার দেখানোর জন্য একটা external switch যুক্ত করবো।

(Refer Slide Time: 01:03)



(Refer Slide Time: 01:35)

Introduction

- Three example programs shall be demonstrated on the STM32F401 development board.
- Here, we do not interface any external devices.
- We use the I/O facilities available on-board (like switches and LED), and simple external input.
- The examples:
 - Blink a LED on and off with specified on and off durations.
 - Depending on a user input on a port line, turn on or off a LED.
 - Use the on-board switch to turn on or off a LED.

এই টিনটে প্রোগ্রামের(program) উদাহরণ STM32F401 development board এ দেখানো হবে।

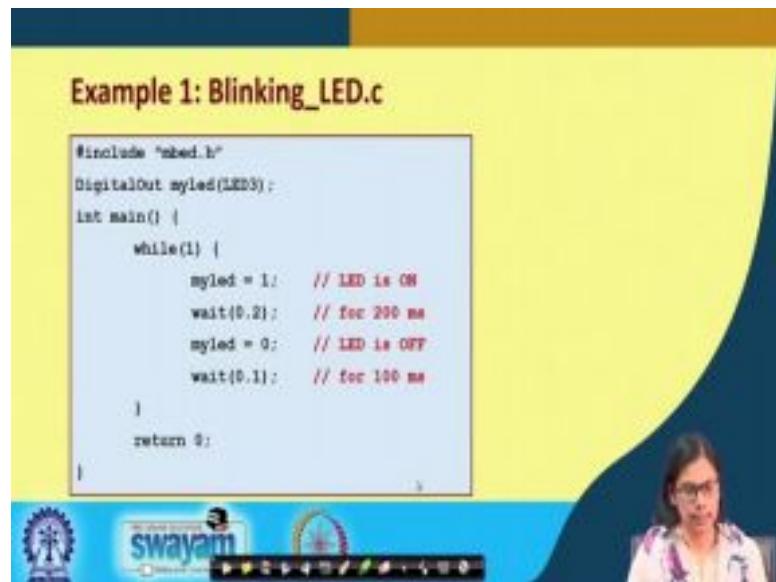
(Refer Slide Time: 01:48)



আমি আপনাকে এই board টা দেখাতে চাই। আপনি এই onboard LEDটা দেখতে পাচ্ছেন। আরো একটি onboard LED আছে, এগুলি হল user button ও reset button, এটা একটা লাল ও সবুজ LED, আমরা যখন প্রোগ্রামটা(program) রাখবো এই LED টা জ্বলবে। আর, এইগুলি ইনপুট আউটপুট digital pins যেগুলো ইনপুট pin অথবা আউটপুট pin হিসেবে ব্যবহার করা যায়।

উদাহরণগুলি এই রকম হয়। onboard LED টা কিছুক্ষণের জন্য জ্বলবে, আর কিছুক্ষণের জন্য বন্ধ হবে। এটা প্রথমটা । পরেরটায়, port line তে user input এর উপর নির্ভর করে LED টা জ্বলবে। ধরলাম LED টা জ্বলছে এবং যখন আমি switch টিপলাম, এটা অফ(off) হয়ে গেল অথবা উল্টোটা। LED টি off থাকবে আর যখন আমি switch টিপবো, LED টা on হবে।

(Refer Slide Time: 04:00)



আসুন আমরা এগিয়ে চলি। এটা আমাদের প্রথম program. আসুন দেখি এই program প্রকৃতপক্ষে কিভাবে কাজ করে। প্রথমটা হল mbed.h যা যেকোনো প্রোগ্রামে বাধ্যতামূলক, এটি header file. পরেরটা হল DigitalOut.

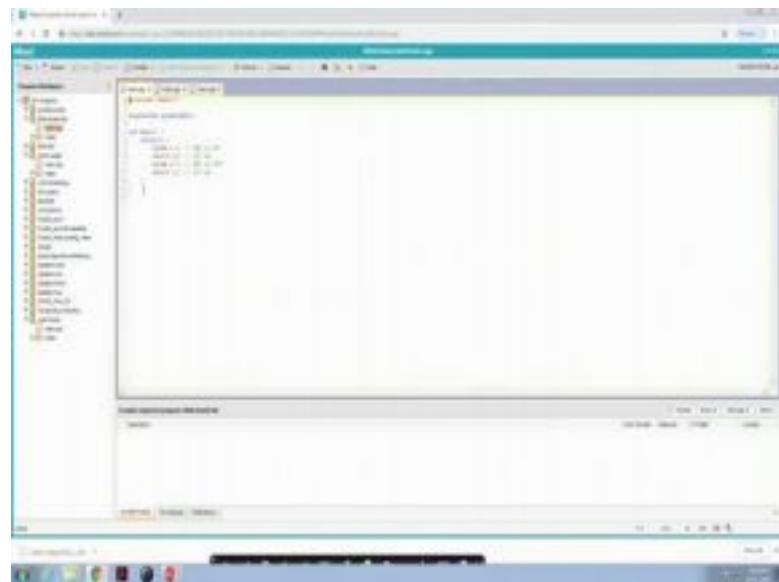
আমরা ব্যবহার করছি onboard LED LED3, আমরা LED টাকে digital আউটপুট হিসেবে ব্যবহার করছি ও এর নাম লিখছি myled. এছাড়াও অনেক port আছে, আপনি সেগুলো DigitalOut হিসেবে ব্যবহার করতে পারেন।

Main function থেকে execution শুরু হয়। এখানে main এর মধ্যে একটি while loop আছে। সাধারণত আপনি দেখবেন embedded application এ সাধারণত কি করা হয়, এটা কিছু ইনপুট পড়বে আর সেই ইনপুট এর উপর নির্ভর করে, এটা কোন কাজ করবে এবং সেটা হতেই থাকবে।

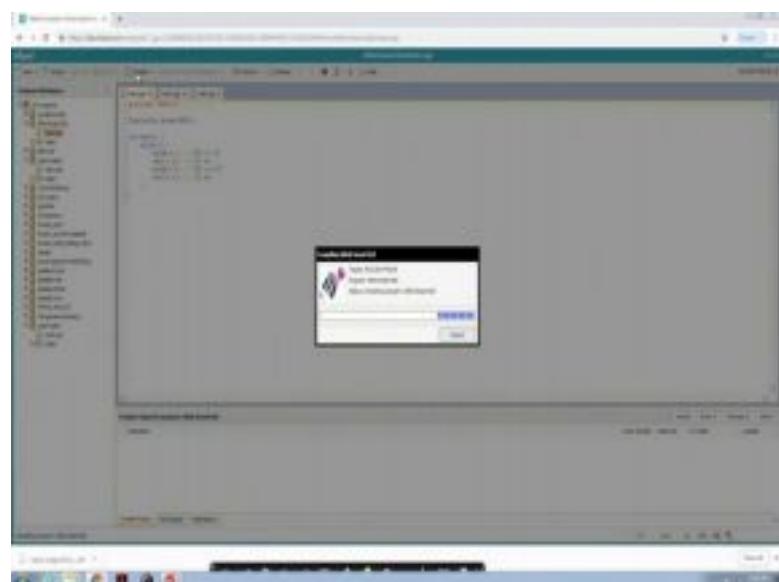
আমি এটা while(1) করে রেখেছি, কারণ আমি চাই এই জিনিসগুলো(set of things) সব সময় ঘটতে থাকুক। এই পদ্ধতিটা infinite loop এ চলতে থাকে।

এখন আমি এই নির্দিষ্ট code টি আপনাকে দেখাবো। আপনি দেখুন আমি আগেই USB port টি আমার PC এর সাথে যুক্ত করেছি। এবং আমি সেই codeটা, যা আমি আপনাকে দেখিয়েছি, সেইটা, আমি এখন online mbed compiler টি ব্যবহার করে সেখানে রাখবো।

(Refer Slide Time: 07:43)

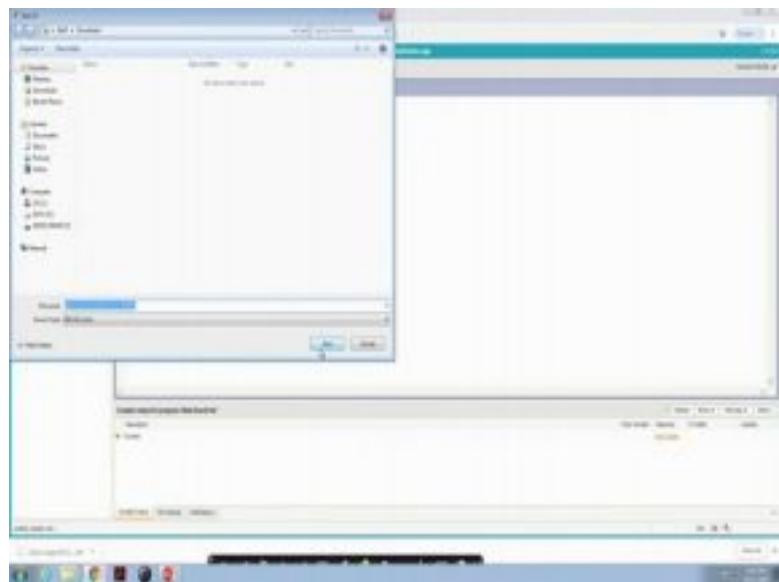


(Refer Slide Time: 07:51)

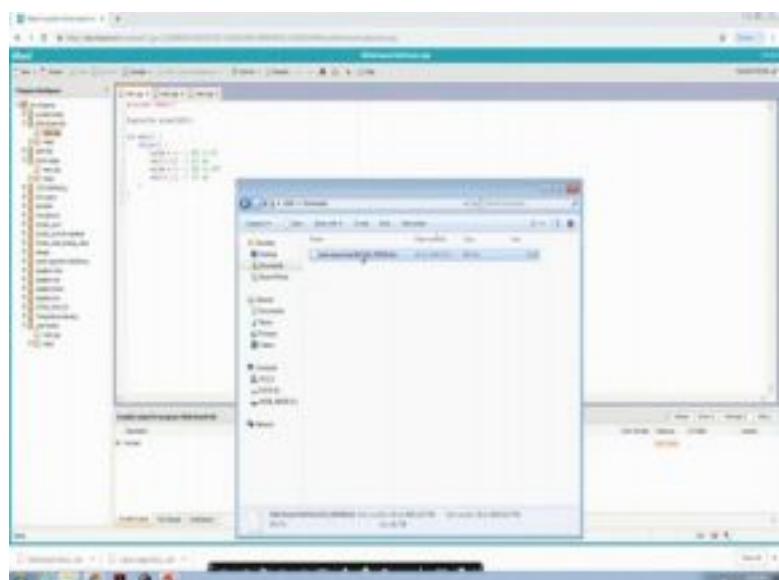


সুতোঁ, আমরা এখানে কি করছি? আমি code টা compile করছি, codeটা আমি আগেই লিখেছি, এটা extract করছি, আর Download folder এ save করছি।

(Refer Slide Time: 08:00)

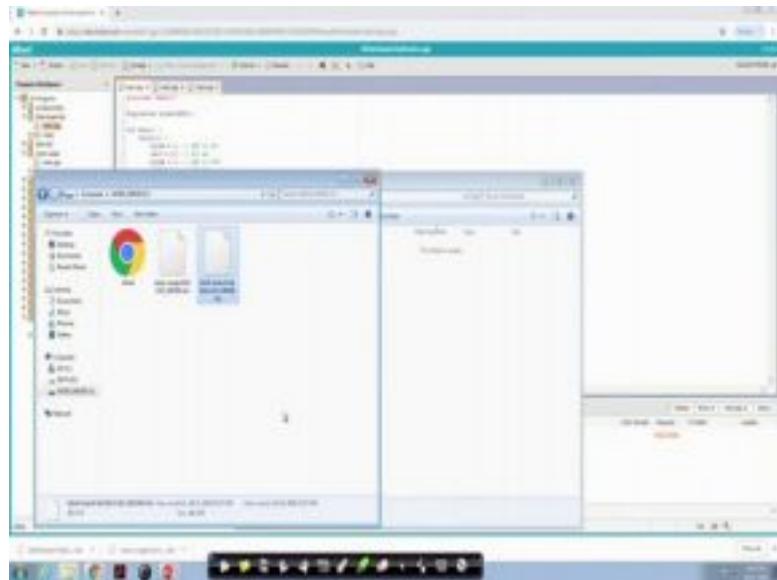


(Refer Slide Time: 08:06)



আমি Download folder খুলবো, এটা নেব, copy করব এবং এই নির্দিষ্ট board এ আমি এটা paste করব।

(Refer Slide Time: 08:14)



এখন, আপনি দেখুন paste করার পর কি হচ্ছে ? এটা LED3, এটা আসলে জ্বলছে নিভছে। আমি delay টা পরিবর্তন করব, আপনি মনোযোগ দিয়ে দেখুন আর আমি শুধু delay টা পরিবর্তন করব; ধরে নিলাম LED টা 2 সেকেন্ড জ্বলবে এবং 1 সেকেন্ড off থাকবে।

আমি code টা আবার compile করব, save করব এবং তারপর এটাকে আমি nucleo board এ copy করব, আমি code টা এই বোর্ড(board) রাখবো। এখন আপনি দেখুন LED টি 2 সেকেন্ড জ্বলবে আর 1 সেকেন্ড off থাকবে, এটা আবার দু সেকেন্ড জ্বলবে আর 1 সেকেন্ড off থাকবে। সুতরাং, সেই জ্বলা-নেভার(blinking) গতি পরিবর্তিত হয়েছে।

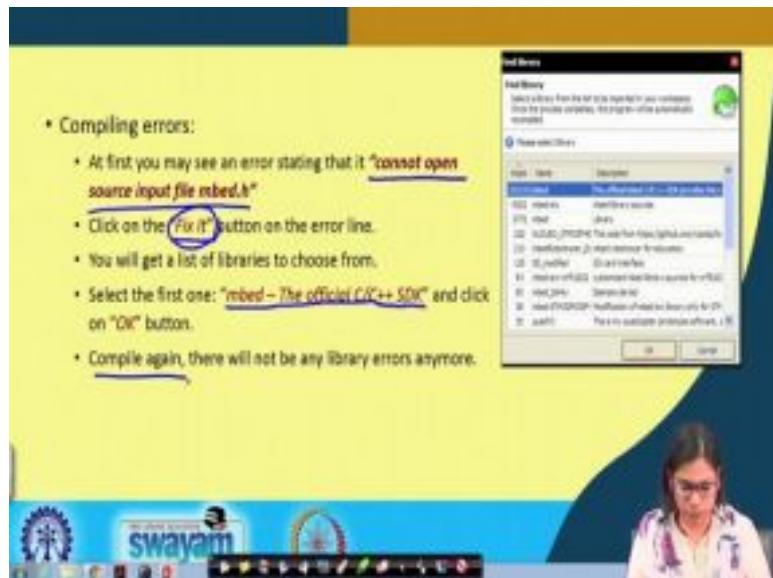
আপনি কিভাবে onboard LED ব্যবহার করবেন, তার একটি উদাহরণ দেখালাম।

(Refer Slide Time: 10:23)



সুতৰাং, program এ ক্ষেত্ৰত আসছি, এটা ছিল program এবং আপনি সেই অনুসারে পরিবৰ্তন কৰতে পাৰেন।

(Refer Slide Time: 10:32)

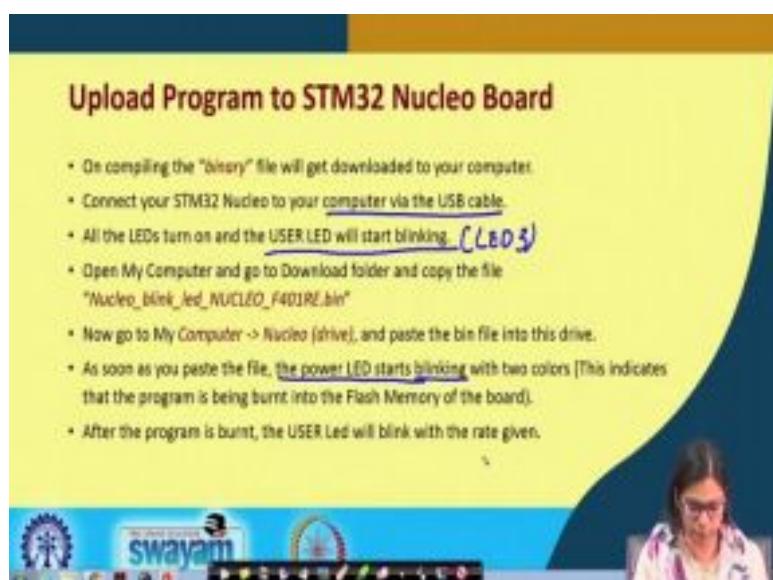


যখন আপনি এটা প্ৰথমবাৱ কৰবেন তখন এই ধৰনেৰ ভুল হতে পাৰে। আপনি এই ভুলটা দেখতে পাৱছেন, ভুলটা হলো আপনি ইনপুট file embed.h এৱে **উৎস(source)** খুলতে পাৱছেন না।

আপনাকে যেটা কৰতে হবে, সেটা হল FixIt এ click কৰা।আপনি এই buttonটা নিচেৰ কোনে দেখবেন, এৱে উপৰ click কৰবেন আৱ পছন্দ কৰাৰ জন্য লাইভেৱিল একটা তালিকা পাৱেন। এই ক্ষেত্ৰে আপনাকে প্ৰথমটা নিৰ্বাচন কৰতে হবে, যেটা হল mbed, যা official C C++ SDK.

আপনি মনে ৱাখবেন যখন আপনি প্ৰথমবাৱ কৰছেন, আপনি সন্তুষ্ট এ ধৰনেৰ সব ভুল পাৱেন।

(Refer Slide Time: 12:00)



আপনি এখন কিভাবে programটা upload করবেন? Binary file টা compile করলে এটা computer এ download হবে। তখন STM Nucleo board কে আমরা computer বা laptop এর সাথে USB cable দিয়ে যুক্ত করব, আর সমস্ত LED on থাকবে, user LED টি জ্বলতে-নিভতে থাকবে।

এটা আপনার প্রোগ্রাম আর এখন আপনি MyComputer Nucleo drive এ যান ও bin file টি এই drive এ paste করবেন, যেটা আমি আপনাকে দেখিয়েছি। যখনই আপনি ওই file টি paste করবেন, আপনি দেখবেন ওই power LED জ্বলতে-নিভতে থাকবে। যদি আপনি লক্ষ্য না করে থাকেন তাহলে পরবর্তী program এ আমি আপনাকে দুটো রং দিয়ে আপনাকে দেখাবো। প্রাথমিকভাবে এটা বুরা যায় যে এই প্রোগ্রামটি(program) flash memory তে burn করে রাখা আছে। আর program টি burn হওয়ার পরে user LED জ্বলতে-নিভতে থাকবে।

(Refer Slide Time: 13:37)



এখন, আমি দ্বিতীয় উদাহরণে চলে যাব। এই পরীক্ষায় আমরা microcontroller এ একটা ডিজিটাল ইনপুট(digital input) দিতে চেষ্টা করব। আমরা ব্যবহার করব একটা switch, একটা matrix switch যাতে একটা পৃথক switch ব্যবহার করব এবং LED3কে আবার জ্বালিয়ে-নিভিয়ে(blinking) প্রতিক্রিয়া পরীক্ষা করব। আমরা এখনকার জন্য বাইরের LED ব্যবহার করব না।

আমাদের একটা ছোট তারের সুতোর দরকার হবে আর আপনি এটা 5V supply ও ground কে board এর সাথে যুক্ত করুন। board এর একটি input port এর input করার জন্য আমি matrix switch থেকে পৃথক switch টি নেব।

(Refer Slide Time: 14:50)

Create the Program

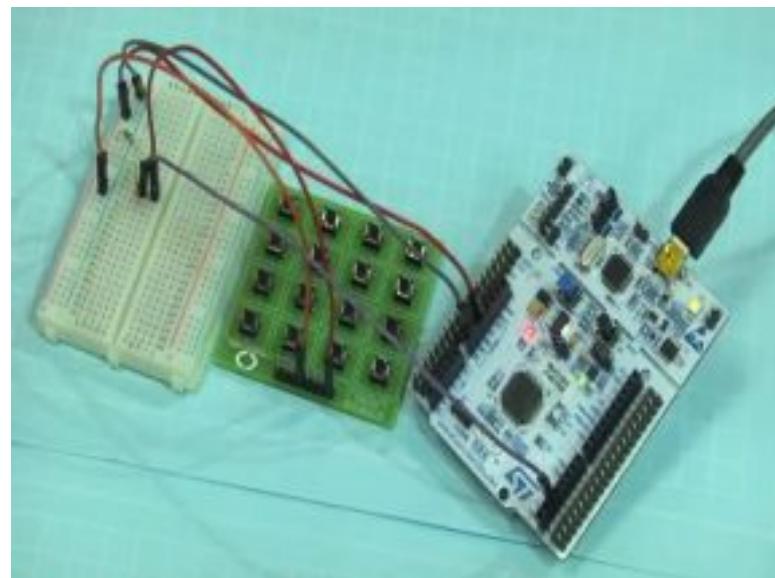
- Following previous steps, create a Program named "DigitalIOTest" and add a "Digital_IOTest.cpp" file to the Program folder.
- Now write the code as shown.

```
#include "mbed.h"
DigitalIn myInput(D2);
DigitalOut myLED(LED3);
int main() {
    while(1) {
        if(myInput) { myLED = 1; }
        else{ myLED = 0; }
    }
    return 0;
}
```

Digital_IOTest.cpp

Input টা নির্দিষ্ট করার জন্য এখানে আমাদের DigitalIn করতে হবে। D2 হবে ডিজিটাল ইনপুট port আর board এর উপরের LED, সেই LED3 হবে ডিজিটাল আউটপুট। প্রোগ্রামটি **স্ব-বর্ণনামূলক(self explanatory)**।

(Refer Slide Time: 16:31)

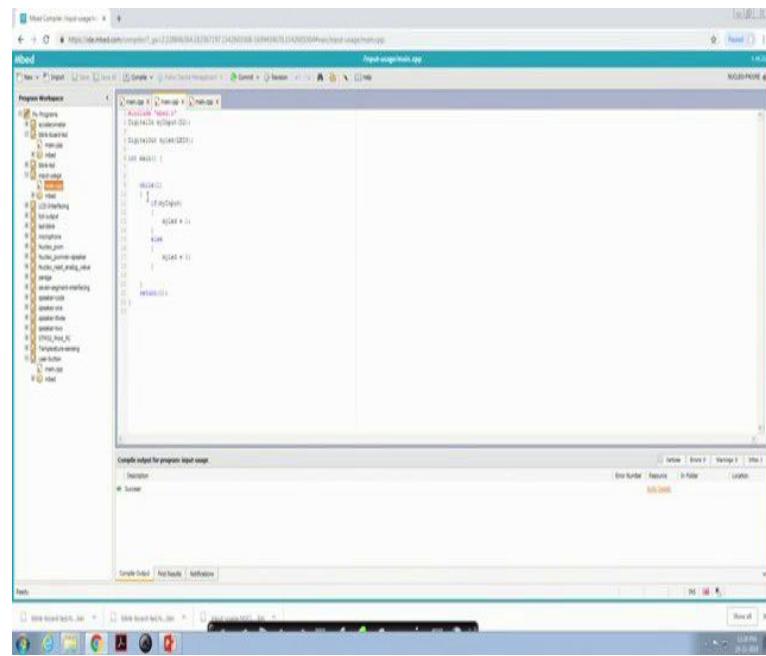


এখন, আমরা code কিভাবে কাজ করে দেখব। এটা আমার switch. আমি কি করেছি সেটা এখন আপনাকে বলব। আপনি দেখবেন এটা row 1, ও এইটা column 1.

Switch-এর একটি প্রান্ত resistance হয়ে Vcc-র সাথে যুক্ত হয়, port-এর অপর প্রান্ত ground এর সাথে যুক্ত হয়। এবং এই সাধারণ point থেকে আমরা port D2-তে input নেবো।

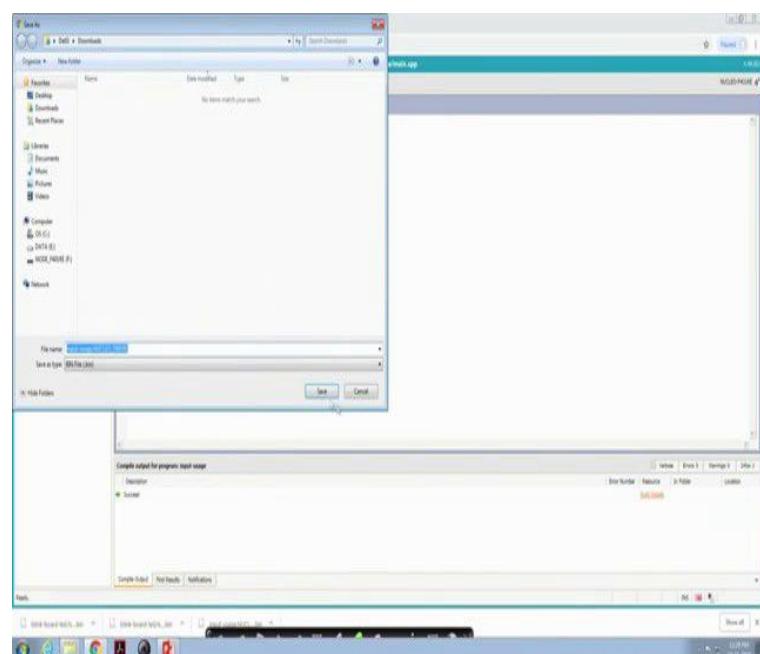
সুতরাং প্রাথমিকভাবে এটা কি মান(value) নেবে? এটা 5V এর কাছে মান নেবে।

(Refer Slide Time: 19:36)

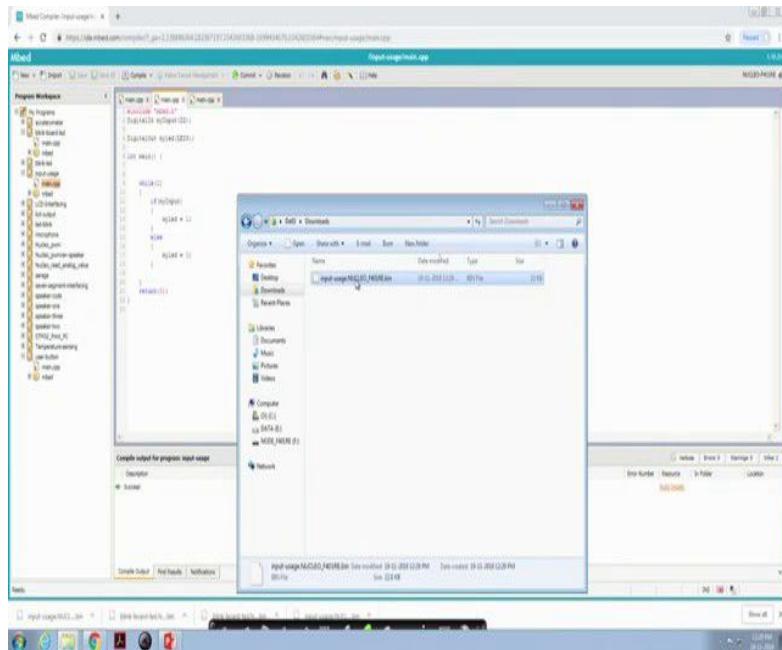


এটা আমার কোড(code) যেটা আমি এক্সুনি আপনার সাথে আলোচনা করলাম।

(Refer Slide Time: 19:53)



আমি এটাকে compile করব, download করব, ও paste করব।
(Refer Slide Time: 19:59)



যখন paste করব, আমি এই board টা আবার আপনাকে দেখাবো, কারণ তখন আপনি দেখতে পাবেন এই নির্দিষ্ট LED টা জ্বলতে-নিভতে থাকবে। key press করার উপর নির্ভর করে LED টি জ্বলবে নিভবে।

(Refer Slide Time: 21:36)

Typical Usage of Digital I/O

- Digital I/O can be used for interfacing external devices some of which we will see in the course of the lectures and demonstration.
 - Switching based on activity (like interrupts, etc.).
 - Controlling electro-mechanical devices.
 - Taking sensor inputs.
 - Communicating other similar or dissimilar devices
 - many more..

সুতরাং, এই digital I/Oকে বাইরের যন্ত্র interface করার জন্য ব্যবহার করা যেতে পারে, তাদের মধ্যে কিছু কিছু আমরা বিভিন্ন বক্তৃতায় ও প্রদর্শনীতে দেখব।

(Refer Slide Time: 22:23)

Example 3: Testing the USER_BUTTON

- Here we will be using the *USER* button available on the board to control a *DigitalOut* to the user LED on board.
- Please note that the tact switch on the board by default sends out a V_{cc} signal (i.e., logic high) and when pressed sends out a ground signal (i.e., logic low).



শেষ উদাহরণটা হল user button পরীক্ষা করা। আমরা এটাও দেখেছি এই নির্দিষ্ট বোর্ডে(board) একটি onboard user button থাকে। এখানে আমরা ওই user button টি board এর উপর user LED নিয়ন্ত্রণ করতে ব্যবহার করব।

আপনি নিচয়ই লক্ষ্য করেছেন এই কৌশলী switch, যেটা বোর্ডে(board) আছে, সাধারণত একটা V_{cc} signal দেয়। সুতরাং, যদি এটাকে না টেপা হয়, এটা 1 signal দেবে, আর টিপলে এটা ground signal দেয়; সেটা হবে logic 0.

(Refer Slide Time: 23:35)

Test_USER_BUTTON.cpp

```
#include "mbed.h"
DigitalIn myButton(USER_BUTTON);
DigitalOut out(LED3);
int main(){
    while(1){
        if(!myButton){      //when button pressed it sends 0
            out = 1;
        }
        else{              //when not pressed it sends 1
            out = 0;
        }
    }
    return 0;
}
```



এখন আমরা প্রথম প্রোগ্রাম দেখব, যেটা স্বায়থ্যমূলক।

এখন, এটা user button, LED টা জ্বলছে না কিন্তু এটা টিপলে এটা জ্বলছে। আমি আবার টিপলে,

এটা জ্বলছে, ছেড়ে দিলে, এটা জ্বলছে না।

সুতরাং, এই নির্দিষ্ট বক্তৃতায় আমরা এই পরীক্ষাগুলো আপনাকে দেখালাম।

ধন্যবাদ।

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Lecture – 21
Interfacing with Arduino Uno

21 তম বক্তৃতায় স্বাগত। এই বক্তৃতায় **Arduino UNO**, যেটা খুব জনপ্রিয় **board**, এর সাথে আমরা কিভাবে **Interface** করতে পারিতা আমি আপনাকে দেখাবো। এর আগের বক্তৃতা গুলিতে আমি আপনাকে নির্দিষ্ট ভাবে দেখিয়েছি আমরা কিভাবে **STM board** ব্যবহার করব এবং এখানে আমি যে পদক্ষেপগুলো **Arduino Uno board** ব্যবহার করতে প্রয়োজন হয়, সেগুলো যথেষ্টই সোজা, তা দেখাবো। প্রথমে আমি আপনাকে এই **board** এর বৈশিষ্ট্যগুলো দেখাবো এবং তারপর কি কি পদক্ষেপ **Arduino** এর সাথে **interface** এর জন্য দরকার তা দেখাবো।

(Refer Slide Time: 01:08)



আমি আপনাকে যেমন বললাম, আমি **Arduino UNO board** এ যেসব **pin** আছে তাদের বর্ণনা দেবো এবং **LED** এর জ্বলা-নেড়া(blinking) উদাহরণ তৈরি করব। যদিও এই পরীক্ষায়

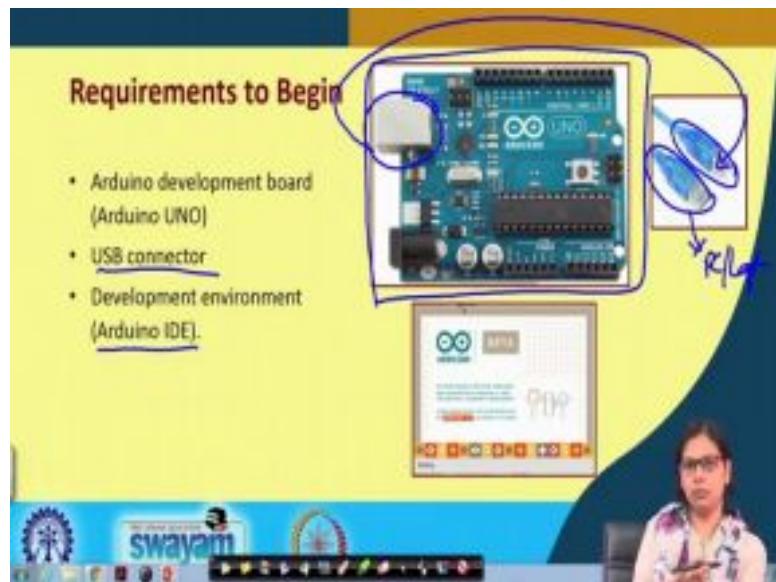
আমি LED এর সাথে interface করবো না, কিন্তু পরবর্তী slide গুলিতে আপনি দেখতে পাবেন আমরা কিভাবে board টির সাথে **interface করি**। আমি নির্দিষ্টভাবে Arduino UNO board ব্যবহার করার প্রয়োজনীয় ধাপগুলি আপনাকে দেখাবো।

(Refer Slide Time: 01:41)



সুতরাং, Arduino কি? এটা একটা **computer hardware & software** এর **open source** কোম্পানি। এটা **বিভিন্ন embedded systems board** ও আনুষঙ্গিক জিনিস ডিজাইন(design) ও উৎপাদন করে। এবং এদের মধ্যে খুব জনপ্রিয় একটি বোর্ড(board) হল **Arduino UNO**, যা সারা দুনিয়াব্যাপী বিভিন্ন ছোট ছোট **embedded system project** ডিজাইন(design) করার জন্য ছাত্র সম্প্রদায় ব্যবহার করে। এই **Arduino UNO** কে এই **নির্দিষ্ট site** থেকে **download** করা যায়, আপনি এখানে সরাসরি যেতে পারেন এবং ওই **নির্দিষ্ট site** থেকে **download** করতে পারেন।

(Refer Slide Time: 02:26)



সুতরাং, এর প্রয়োজনীয়তা কি কি? আমরা আগেই STM32 board এর কি প্রয়োজনীয়তা তা দেখেছি। এখন, Arduino Board এর জন্য এটা প্রায় একই রকম। এটা নির্ভর করে আমরা কিভাবে interface করছি তার উপর, কিন্তু সহজতম পদ্ধতিটা হল আপনি **USB connector** এর মাধ্যমে এটা আপনার **PC**-র সাথে যুক্ত করুন। এই jack টা এখানে যুক্ত করা হলো এবং এই jack টা আপনার PC বা laptop এর সাথে যুক্ত করা হবে। এখানে Arduino IDE ব্যবহৃত হয়, এটা এইরকম দেখতে লাগে, যেটা পূর্বে বর্ণিত website থেকে download করা যাবে।

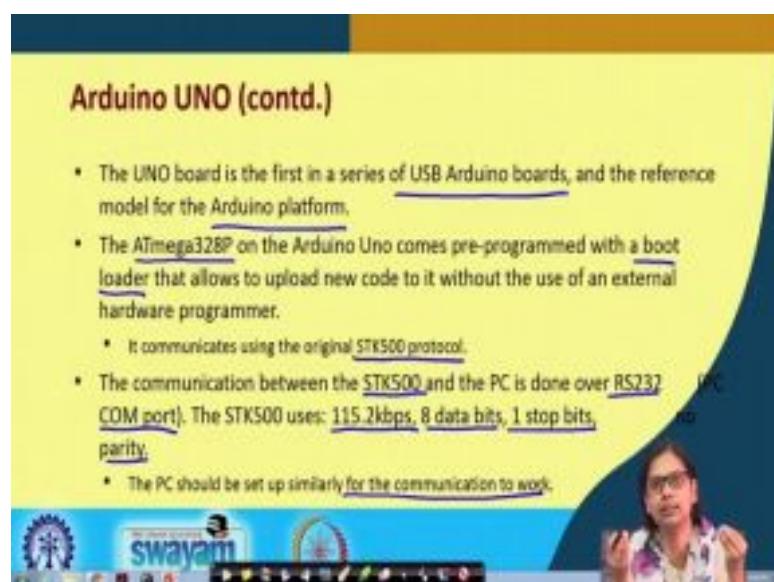
(Refer Slide Time: 03:24)



এটি হলো Arduino UNO, যা বহুব্যবহৃত open source microcontroller board, যেটা ATmega328P microcontroller এর উপর ভিত্তি করে এবং Arduino দ্বারা তৈরি 14 টা ডিজিটাল ইনপুট আউটপুট পিন(pin) উপলভ্য, যার সবগুলিই Arduino সামঞ্জস্যপূর্ণ(compatible) পিন(pin)।

সুতরাং, 14 টা ডিজিটাল pin ও 6 টা analog pin আছে এবং সেগুলো এই typeB USB cable এর মাধ্যমে Arduino IDE দ্বারা programmable, যা আমি আগেই দেখিয়েছি এটি USB cable বা একটি বাহ্যিক 9 ভোল্ট(volt) ব্যাটারি দ্বারা চালিত হতে পারে। ধরুন আমরা বললাম আপনি ইতিমধ্যে প্রোগ্রামটি(program) রেখে দিয়েছেন এবং আপনি যে কোনও PC-র সাহায্য ছাড়াই এটি চালাতে চান। সুতরাং, আপনি এতে কিভাবে শক্তি(power) দেবেন? এই বাইরের battery ব্যবহার করে আপনি সেটা করতে পারেন।

(Refer Slide Time: 04:40)

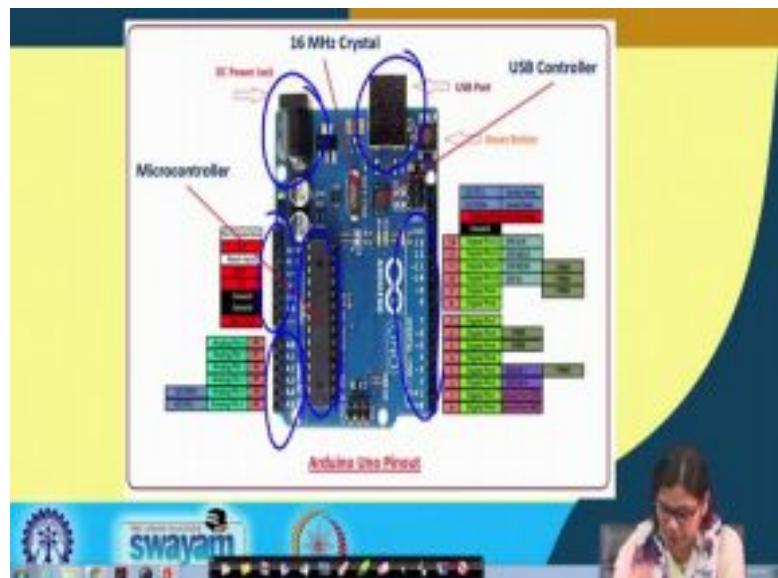


এইটি USB Aduino board গুলির সিরিজের মধ্যে প্রথম বোর্ড(board)। পূর্ববর্তী বোর্ড(board) গুলিতে আপনাকে এটিকে USB-র মাধ্যম ছাড়া, অন্য কোনও উপায়ে রাখার দরকার ছিল। কিন্তু, যখন USB সামঞ্জস্যপূর্ণ বোর্ডগুলি(board) আসে, তখন Arduino UNO হল প্রথম এই ধরণের board এবং Arduino platform-র জন্য reference model. Arduino UNO-তে ATmega 328P একটি বুট লোডার(boot loader) সহ প্রিপ্রোগ্রামযুক্ত (preprogrammed) হয়ে আসে যা কোনও বাহ্যিক হার্ডওয়্যার প্রোগ্রামার(hardware programmer) ব্যবহার না করেই এতে নতুন কোড(code) আপলোড(upload) করতে দেয়। সুতরাং, আমাদের কোন বাহ্যিক হার্ডওয়্যার

প্রোগ্রামার(hardware programmer) দরকার নেই; বরং যদি আমরা inbuilt boot loader টা সংযুক্ত করি, তা ইতিমধ্যেই সেখানে আছে।

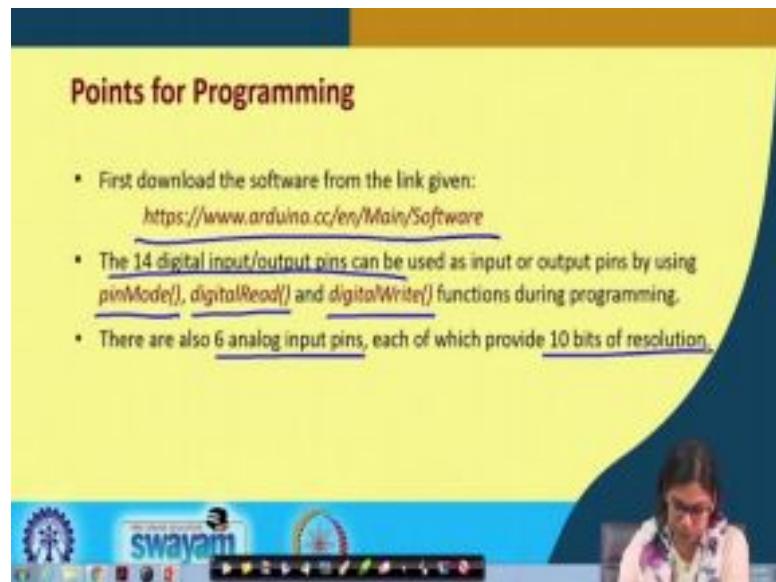
এটি মূল STK500 প্রোটোকল(protocol) ব্যবহার করে যোগাযোগ করে, এটি একটি প্রোটোকল(protocol) যা এই যোগাযোগের জন্য ব্যবহৃত হয়। এটা এবং PC-র মধ্যে যোগাযোগ RS232-এর মাধ্যমে সম্পন্ন হয় এবং এই STK500টি 115.2 kbps, 8 বিট(bit) ডেটা, 1 stop বিট এবং parity ব্যবহার করে। যোগাযোগের কাজ করার জন্য PC একইভাবে সেট আপ(set up) করা উচিত। যখন 2 টি যন্ত্র একে অপরের সাথে যোগাযোগ করে, যদি কোনও যন্ত্র উচ্চ হারে(rate) চালিত হয় অন্য যন্ত্র সেই হারে না চালায়, অন্য যন্ত্রটি যোগাযোগ করছে কী হারে এই যন্ত্রটিকে অবশ্যই তা জানতে হবে। মূত্রাঃ, এই উদ্দেশ্যে এটি প্রয়োজন।

(Refer Slide Time: 00:34)



আপনি দেখতে পারেন এটি সামগ্রিক বোর্ড(board)। আমি ইতিমধ্যে এই power jack টি সম্পর্কে আলোচনা করেছি, এটি USB controller এবং USB port যার মাধ্যমে এটি সংযুক্ত হতে পারে। এটি ATmega microcontroller, এটি ডিজিটাল পাওয়ার সাপ্লাই পিন(digital power supply pins), এণ্ডলি এনালগ পিন(analog pins) এবং এণ্ডলি ডিজিটাল ইনপুট আউটপুট পিন(digital input output pins)। আমরা যখন এই নির্দিষ্ট বোর্ডের(board) সাথে পরীক্ষা করবো তখন আমরা এণ্ডলি সন্ধান করব। আমরা বিশেষভাবে এই বিশেষ পিনগুলির(pin) বিভিন্ন দিকগুলো অনুসন্ধান করব।

(Refer Slide Time: 07:24)



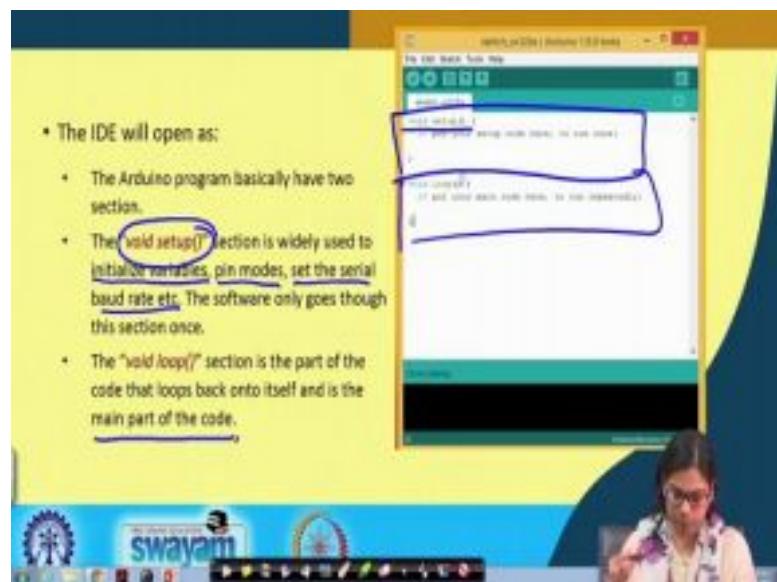
Arduino ব্যবহার করে প্রোগ্রামিংয়ের(programming) জন্য আপনার কেন পয়েন্টগুলি মনে রাখা দরকার? প্রথম এবং সর্বাগ্রে বিষয় হল আপনাকে IDE স্টেওয়্যার(software)টি ডাউনলোড(download) করতে হবে। এবং 14টা ডিজিটাল ইনপুট আউটপুট পিনগুলি (digital input output pins) ব্যবহার করার জন্য, আমাদের নিম্নলিখিত কার্যকারিতাগুলি(functions) জানতে হবে: একটি হল spin mode, অন্যটি ইনপুট পড়ার জন্য digital read, port টিতে মান(value) লেখার জন্য digital write।

এছাড়াও 6 টি অ্যানালগ ইনপুট পিন(analog input pins) রয়েছে যার প্রতিটি 10 বিট রেজোলিউশন(resolution) সরবরাহ করে। আমরা ইতিমধ্যে অ্যানালগ থেকে ডিজিটাল রূপান্তরকারীটির(converter) রেজোলিউশন এবং অন্যান্য দিকগুলি সম্পর্কে আলোচনা করেছি। সুতরাং, এটিতে 10 বিট রেজোলিউশন(resolution) আছে।

(Refer Slide Time: 08:22)



একবার ডাউনলোড(download) হয়ে গেলে ডাউনলোড করা zip টি বের করুন এবং arduino.exe তে ক্লিক করুন, তারপরে arduino IDE খুলবে যা এটার মতো দেখাচ্ছে।
 (Refer Slide Time: 08: 48)

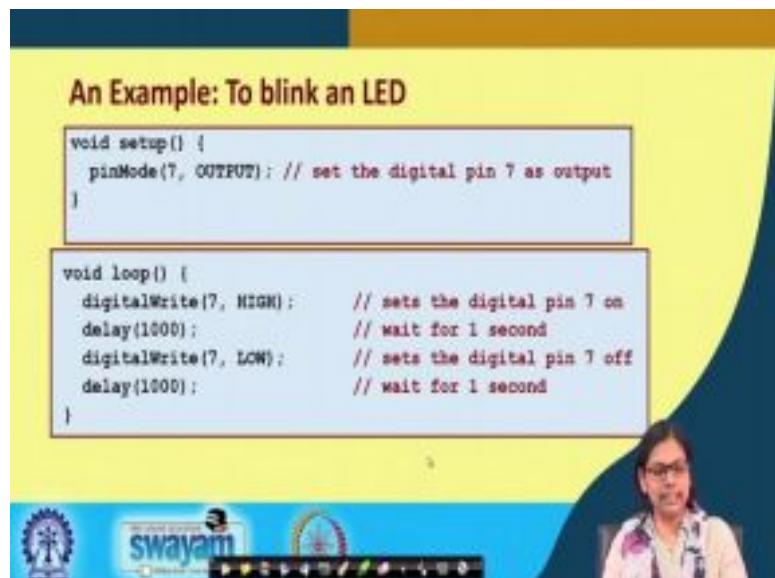


আপনি ইতিমধ্যে IDE ইনস্টল(install) করার কাজটি করেছেন যা মোটামুটি মেজা, এরপর যখন আপনি এটি খুলবেন IDE টা এইভাবে খুলবে। এটি সেটআপ(setup) পর্ব এবং এটি লুপ(loop)। সেটআপ(setup) পর্যায়ে, আমরা এখানে যে সেটআপ(setup) কোডটি(code) লিখি তা কেবল একবার চালানো হয়। সেটআপ(setup) বিভাগটি ব্যাপকভাবে ভেরিয়েবলগুলি(variables) সূচনা করতে, পিনের মোডগুলি(pin modes) সম্পর্কে উল্লেখ করতে এবং সিরিয়াল বাড রেট(serial baud rate) ইত্যাদি নির্ধারণের জন্য ব্যবহৃত হয়। আমরা যখন GSM বা Bluetooth- এর মতো অন্যান্য যোগাযোগের যন্ত্রগুলি ব্যবহার করতে পারি, আপনাকে এখানে সেই হারে নির্দিষ্ট করতে হবে।

সফ্টওয়্যারটি(software) কেবল একবার এর মধ্য দিয়ে যায়। void loop বিভাগটি কোডের(code) সেই অংশ যা নিজের দিকে ফিরে আসে এবং এটি কোডের(code) মূল অংশ। আমরা বলতে পারি যে, আপনি যে কোডটি(code) লিখেছেন, তাতে LED কিছু সময়ের জন্য জ্বলতে থাকবে।

কোডটির(code) সেই অংশটি এই void loop পর্যায়ে(phase) রাখা হবে।

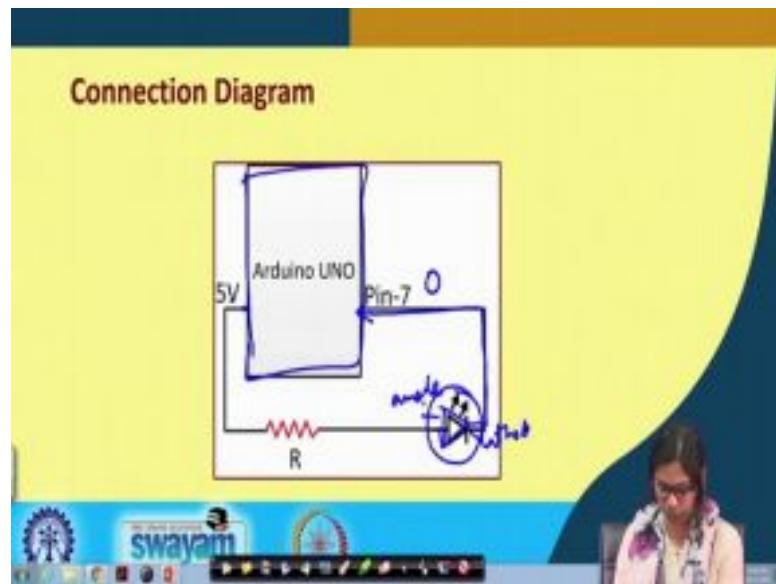
(Refer Slide Time: 10:26)



এখন, আমি একটি LED জ্বলার-নেতৃত্বের জন্য একটি ছোট উদাহরণ গ্রহণ করব। আপনি mbed compiler টিতে দেখেছেন যে আমাদের একটি header ফাইল অন্তর্ভুক্ত করতে হবে, আমরা এখানে কী প্রয়োজন তা আমরা দেখতে চাই। এখানে যে প্রধান পর্যায়গুলি প্রয়োজন তা দুটি, একটি সেটআপ পর্ব(setup phase) এখানে আমি মূলত সূচনা (initialization) করছি, আমরা পিন মোড সেট করছি(setting pin mode), Arduinoর পিন 7 কে output হিসাবে সেট করছি। আমরা এই বিশেষ বিবৃতি ব্যবহার করে এটা করছি।

তারপরে void loop আসে। আমরা digitalWrite নামে একটি ফাংশন(function) ব্যবহার করছি, যেখানে আউটপুট পোর্ট(output port) 7 আমরা high করছি। তারপরে আমরা এক সেকেন্ডের বিলন্ন(delay) দিচ্ছি, এটি মিলিসেকেন্ডে নির্দিষ্ট করা হয়েছে। 1000 মিলিসেকেন্ডে 1 সেকেন্ড। তারপরে আমরা একই পিন(pin) 7 থেকে low পর্যন্ত কমিয়ে একটি digitalWrite করি এবং তারপরে আমরা আবার একই 1000 মিলিসেকেন্ড বা 1 সেকেন্ডের জন্য বিলন্ন(delay) করি এবং এটি লুপে(loop) চলতে থাকে।

(Refer Slide Time: 12:27)



এটি একটি সাধারণ পিন ডায়াগ্রাম(pin diagram)। আমরা এখানে onboard LED ব্যবহার করছি না; বরং আমরা একটি বাহ্যিক LED ব্যবহার করছি। এটি Arduino UNO board, এবং এটি 5V supply এবং এটি এলইডি(LED)। এটি এনোড(anode) এবং এটি ক্যাথোড(cathode)। সুতরাং, একটি রোধের মাধ্যমে অ্যানোড প্রান্তটি 5V-র সাথে সংযুক্ত থাকে এবং ক্যাথোডের প্রান্তটি সরাসরি 7 নম্বর পিনের(pin) সাথে সংযুক্ত থাকে। এটিই সার্কিট ডায়াগ্রাম(circuit diagram), কখন LED জ্বলে উঠবে? যেহেতু এটি 5V-র সাথে সংযুক্ত, যখন এই পিনটিতে 0 ইনপুট থাকবে, তখনই কেবলমাত্র এই LED জ্বলতে থাকবে।

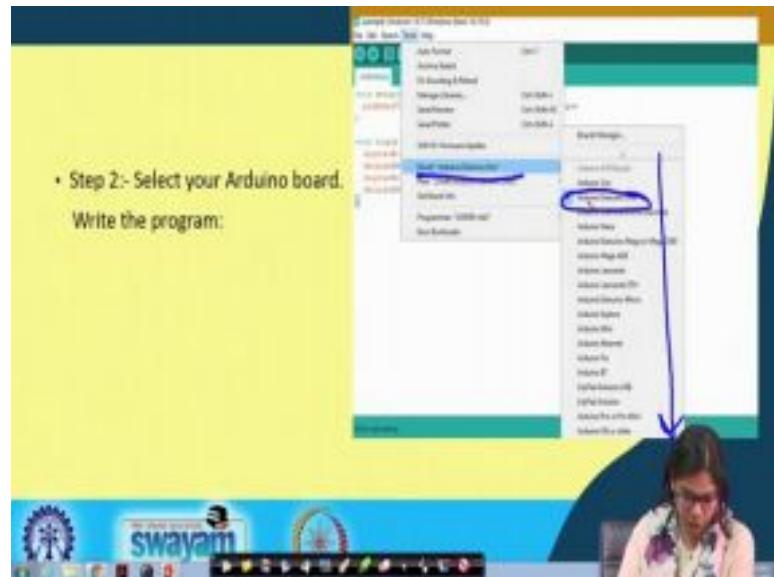
(Refer Slide Time: 13:27)



প্রয়োজনীয় জিনিসগুলি করার জন্য এখানে প্রয়োজনীয় পদক্ষেপগুলি কী কী? এখানে আমরা এই পিন

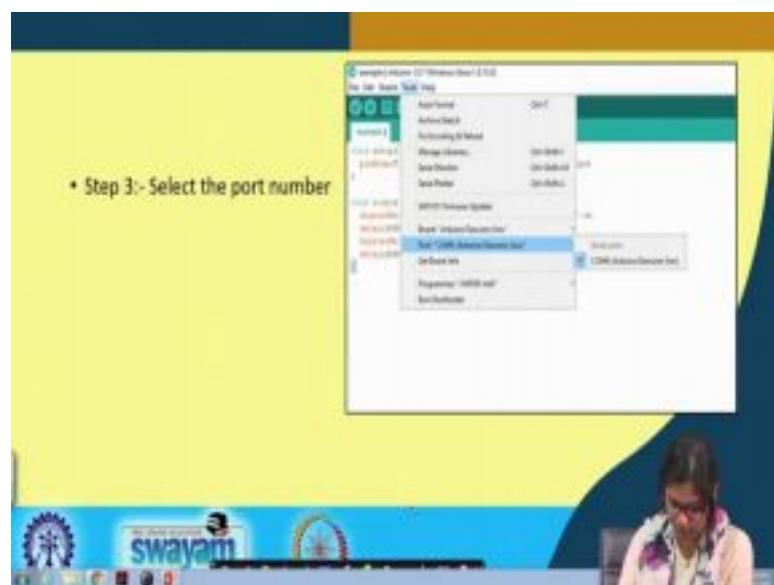
মোড (pin mode) 7 টি আউটপুট হিসাবে নির্বাচন করি, আমরা প্রোগ্রামটি লিখি এবং তারপরে আমরা Arduino board টি নির্বাচন করি।

(Refer Slide Time: 13:47)



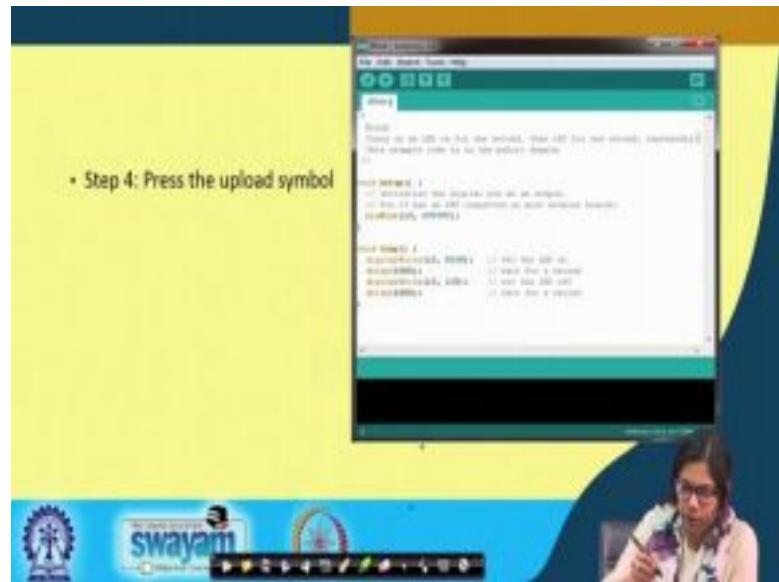
সুতরাং, বিশেষভাবে এই তালিকা থেকে আপনাকে নির্দিষ্টভাবে Arduino UNO board টা ব্যবহার করতে হবে।

(Refer Slide Time: 14:09)



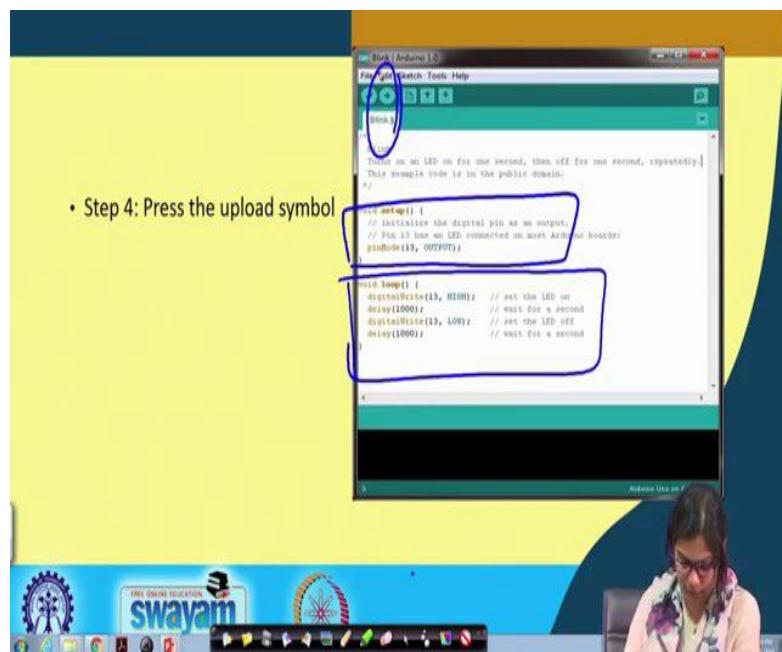
এরপর port সংখ্যা নির্বাচন করুন; আমরা এই নির্দিষ্ট UNO এর জন্য এই COMM নির্বাচন করতে পারি।

(Refer Slide Time: 14:25)



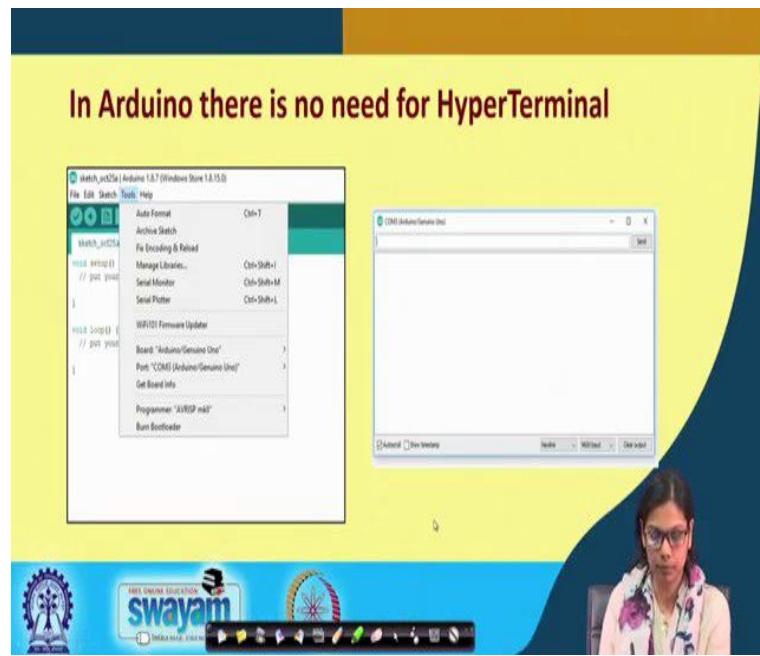
এবং তারপর আমরা এই upload button টিপে upload করি।

(Refer Slide Time: 14:34)



সুতরাং, এটি setup phase, এটা loop phase আর আমরা upload button টিপে program টি upload করব।

(Refer Slide Time: 14:54)



Hyper terminal সম্পর্কে আমরা একটি গুরুত্বপূর্ণ কথা বলিনি যা আমরা পরের সপ্তাহে আলোচনা করব। Arduino-র একটি ভাল বৈশিষ্ট্য রয়েছে যে এটির জন্য কোনও বাহ্যিক hyper terminal-র প্রয়োজন হয় না, এটি সেখানে অন্তর্নির্মিত(in-built)। আপনি tools এ যান এবং তারপরে আপনি এই port এ ক্লিক করতে পারেন এবং আপনি এটি পাবেন। আপনি যত ধরনের তথ্য(data) চান তা আপনি এই port-এ দেখতে পাবেন।

আমরা পরে এটি খতিয়ে দেখব, এটি Arduino-র অন্যতম বৈশিষ্ট্য। সুতরাং, এই Arduino board সম্পর্কে এটিই সমস্ত, আপনি কীভাবে Arduino board-এর সাথে interface করবেন, আপনার সেই ID দরকার, আপনার কোডটি(code) লিখতে হবে, কোডটি দুটি পর্যায়ে যায় এবং শেষ পর্যন্ত আপনাকে কোডটি আপলোড(upload) করতে হবে, এটি আপলোড(upload) হয়ে যন্ত্রটিতে যায় এবং তারপরে এটি সম্পন্ন হয়।

আজ আমি আপনাকে এখন দেখাব যে আমরা ইতিমধ্যে আলোচনা করেছি দুটি ধরণের board-র সাথে কীভাবে একটি LED সংযুক্ত করব।

দুটি বোর্ডই কিছু বৈশিষ্ট্য পেয়েছে, যা আমরা ইতিমধ্যে দেখেছি। এখন আমরা ইন্টারফেসিং(interfacing) পরীক্ষায় যাব। আমি ইতিমধ্যে আলোচনা করেছি যে কীভাবে onboard LED জ্বলে উঠবে। এখন আমি STM বোর্ডের সাথে Arduino বোর্ডের সাথে breadboard ব্যবহার করে অন্য একটি LED সংযুক্ত করব।

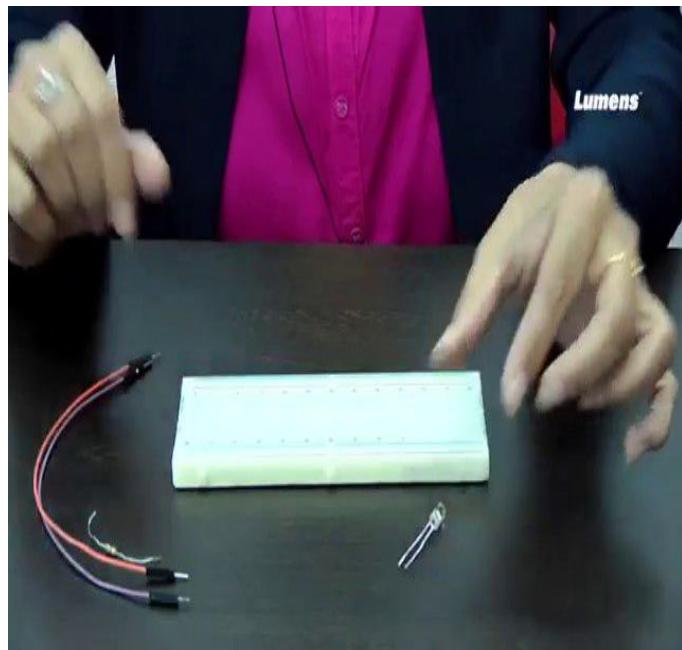
(Refer Slide Time: 17:25)



আমি প্রথমে breadboard নিয়ে কথা বলব। এটি এমন একটি breadboard যার মাধ্যমে আমরা বিভিন্ন ছোট ছোট ইন্টারফেসিং যন্ত্র(interfacing device) সংযোগ করতে পারি এবং আমরা ছোট পরীক্ষাও করতে পারি। এই লাইনগুলি অনুভূমিকভাবে(horizontally) সংযুক্ত। এখানে আমরা দেখতে পাচ্ছি এমন দুটি লাইন রয়েছে, এগুলি উল্লম্বভাবে(vertically) এক সাথে সংযুক্ত।

সুতরাং, যে কোনও বিন্দু(point) থেকে আপনি Vcc বা ground নিতে পারেন এবং তারপরে আপনি প্রয়োজনমত অন্যান্য পিনগুলি ব্যবহার করতে এবং সংযুক্ত করতে পারেন। আবার একটি জিনিস আপনাকে মনে রাখতে হবে যে এখানে একটি কাটা আছে। সুতরাং, এটি সংযুক্ত, তবে এটি এটির সাথে সংযুক্ত নয়। সুতরাং, এটি একটি পৃথক সংযোগ, এটি একটি পৃথক সংযোগ। আপনি যদি পুরো জিনিসটি সংযুক্ত করতে চান তবে আপনাকে এখান থেকে এই পর্যন্ত একটি তারের সংযোগ করতে হবে। আমরা এই breadboard টি ব্যবহার করে LED নিয়ে প্রথম পরীক্ষাটি দেখব।

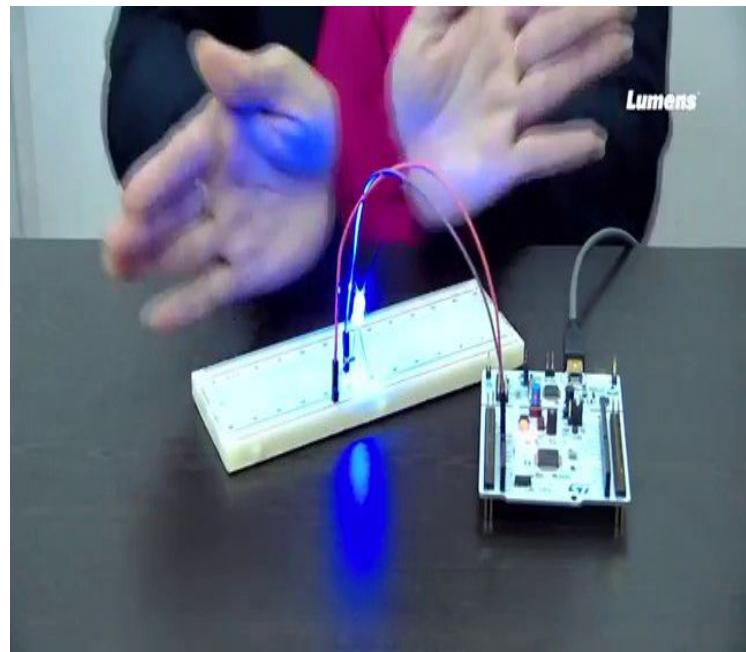
(Refer Slide Time: 19:05)



আমাদের কিছু jumper-র তার এবং একটি resistor-র প্রয়োজন। প্রথমত, এটি LED। আপনি যদি এই LED টি দেখতে পান তবে আপনি এটি বুঝতে পারেন যে এর একটি পা দীর্ঘ এবং অন্যটি ছোট। আমি ইতিমধ্যে বক্তৃতায় এটি নিয়ে আলোচনা করেছি, তবে এর অর্থ কী, এটি হল এনোড(anode) এবং এটিই ক্যাথোড(cathode)। সুতরাং, লম্বা পা টি এনোড এবং সংক্ষিপ্ত পা টি ক্যাথোড এবং বিদ্যুৎ সর্বদা এনোড থেকে ক্যাথোডে প্রবাহিত হয়।

আমি প্রথমে যা করব তা হল আমি এই LED টি STM board-র সাথে সংযুক্ত করব। এবং আমি প্রথমটি যা করব তা হল প্রথমে যাচাই করে দেখব যে এই LED ঠিকঠাক কাজ করছে কিনা। আপনি কিভাবে এটি পরীক্ষা করবেন? আপনি যা করতে পারেন, একটি রোধের মাধ্যমে অ্যানোডকে আপনি VCC-এর সাথে এবং ক্যাথোডকে আপনি সরাসরি ground এর সাথে সংযোগ করতে পারেন এবং আমরা দেখতে পাই যে LED জ্বলছে কিনা।

(Refer Slide Time: 20:34)



সুতরাং, STM board-র সাথে প্রথমে এই পরীক্ষাটি করি। এটি আমার এনোড(anode), এটি ক্যাথোড। সুতরাং, আমি এটি এভাবে breadboard-এ রেখেছি। তারপরে একটি রোধের মাধ্যমে, আমি breadboard এর এই প্রান্তে সংযোগ স্থাপন করব এবং এই প্রান্তটি থেকে উল্লম্বভাবে সংযুক্ত, আমি এটি Vcc তে রাখব। সুতরাং, এখানে Vcc পিনটি(pin) চতুর্থ পিন, তাই আমি এটি সেখানে রেখেছি। এবং এটি এই বিন্দু(point) থেকে, এটি হল এনোড, আমি এটি ground-এ সংযুক্ত করব এবং আমি দেখতে পাই যে LED জ্বলজ্বল করছে। আমি যদি এটি বাইরে নিয়ে যাই তবে এটি জ্বলছে না। এখন আবার এটি এখানে রাখব, এটি আবার জ্বলজ্বল করছে।

সুতরাং, যদি আপনি বুঝতে পারেন যে এখন আপনি এটি তৈরি করতে পারেন যে হ্যাঁ এই LED টি একটি কার্যক্রম LED। এখন আমি যা করব তা হ'ল আমি অ্যানোড(anode)কে Vcc-র সাথে সংযুক্ত করব তবে এখন আমি গ্রাউন্ড(ground)টি ডিজিটাল পিন(pin) ডি2 এর সাথে সংযুক্ত করব। আমি এই বোর্ড(board) একটি কোড(code) রাখবো। কোডটি(code) রাখার জন্য, যে সমস্ত পদক্ষেপগুলি অনুসরণ করা দরকার তা আমরা ইতিমধ্যে দেখেছি।

এখন কি হচ্ছে আপনি কি দেখতে পাচ্ছেন? LED টি 1 সেকেন্ডের জন্য জ্বলজ্বল করছে এবং এটি 2 সেকেন্ডের জন্য বন্ধ রয়েছে, আপনি দেখুন এটি 1 সেকেন্ডের জন্য জ্বলজ্বল করছে এবং এটি 2 সেকেন্ডের জন্য বন্ধ রয়েছে এবং এটি পুনরাবৃত্তি করছে।

একক এলইডি ব্যবহার করে আপনি এইভাবে এই STM board-র সাথে প্রোগ্রামিং(programming) করতে পারেন। আপনি LED দিয়ে বিভিন্ন ধরণের পরীক্ষা-নিরীক্ষাও করতে পারেন, আপনি আরও কয়েকটি LED'র সাথে সংযুক্ত করলে, আপনি সেভাবে একটি counter তৈরি করতে পারেন। সুতরাং,

আপনি এখানে অন্য একটি LED রাখতে পারেন এবং counter টি কী হবে? Counter টি প্রথমে 00 তারপর 01 এবং 10 এবং তারপরে হবে 11. এটি চিরতরে পুনরাবৃত্তি করতে পারে।

এখন আমি Arduino board-র সাথে একই পরীক্ষা করবো। কোডটি(code) আমি ইতিমধ্যে আলোচনা করেছি। এখন আমি প্রথমে Arduino board-র সাথে সংযোগটি করব, এটি Arduino UNO বোর্ড(board)। একইভাবে এই সংযোগটি সোজা। এখন আমি এই অংশটি Vcc-র সাথে এবং এই অংশটি পিন(pin) ডি2 এর সাথে সংযুক্ত করব। এখন আমি আবার Arduino editor ব্যবহার করব, যা আমরা কোডটি(code) রাখার জন্য ব্যবহার করেছি।

এখন কোডটি(code) রেখে দেওয়া হয়েছে এবং এটি একই জিনিসটি করছে, এটি 1 সেকেন্ডের জন্য চালু এবং এটি 2 সেকেন্ডের জন্য বন্ধ রয়েছে।

সুতরাং এই পরীক্ষায় আমরা STM ও Arduino UNO উভয় বোর্ডের(board) সাথে একটি LED সংযুক্ত করেছি। আরও কয়েকটি সংখ্যক LED এর সংযোগ স্থাপনের জন্য আপনি অন্যান্য উদাহরণ ব্যবহার করে দেখতে পারেন এবং তারপরে আপনি কোনও উপায়ে বা অন্য কোনওভাবে একটি LED counter তৈরি করতে পারেন।

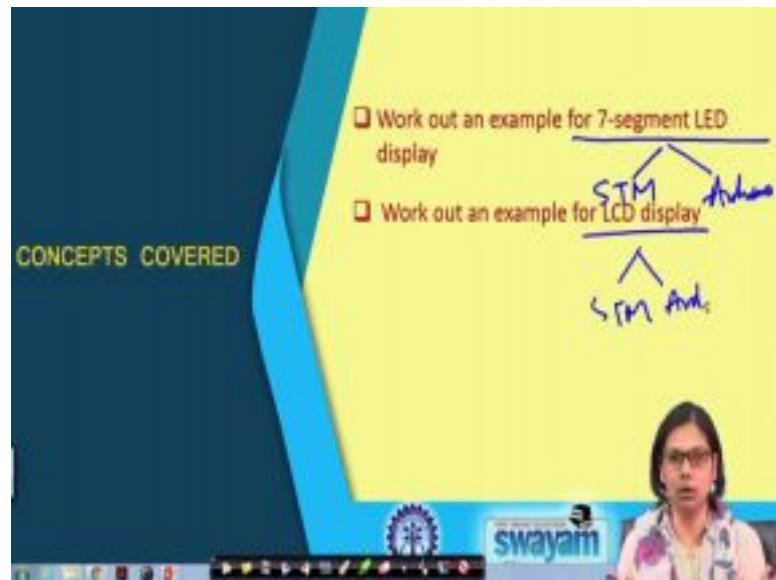
ধন্যবাদ।

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Lecture – 22
Interfacing with 7-Segment LED and LCD Displays (Part I)

22 তম বক্তৃতায় স্বাগতম, এই বক্তৃতায় আমরা 7-সেগমেন্টের (segment) LED এবং LCD ডিসপ্লে (display) ইন্টারফেস (interface) করব। আমরা ইতিমধ্যেই 7 সেগমেন্ট ডিসপ্লে (7 segment display), LCD, তারা কীভাবে কাজ করে তা নিয়ে আলোচনা করেছি। এবং, আমরা ইতিমধ্যেই দুটি বোর্ড, Arduino board এবং STM board সম্পর্কে আলোচনা করেছি। এই বিশেষ বক্তৃতায়, আমি আপনাকে সার্কিট ডায়াগ্রাম (circuit diagram) এবং 7 সেগমেন্ট ডিসপ্লে (7 segment display) ও LCD ইন্টারফেস (interface) করার জন্য প্রয়োজনীয় কোডটি (code) দেখাব।

(Refer Slide Time: 01:11)



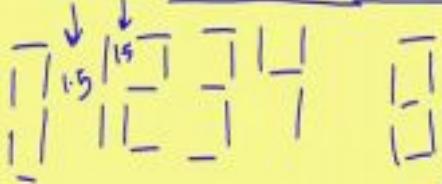
আমি আপনাকে 7 segment display টির একটি কাজের(workout) উদাহরণ দেখাব এবং আমি আপনাকে একটি LCD display টির একটি কাজের(workout) উদাহরণ দেখাব। আমি উভয় STM এবং Arduino-র জন্য দেখাবো।

(Refer Slide Time: 01:50)

Example 1

- Interface a 7-segment display unit to the STM32 board, and display the characters 0 to 9 cyclically with time delay of 1.5 seconds.

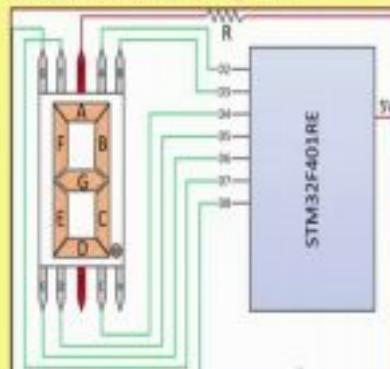
- Connect the segments A to G to the data output pins D2 to D8 on the Arduino connector.



আমি প্রথম যে উদাহরণটি দেখাব তা হল STM32 বোর্ডে 7-সেগমেন্ট ডিস্প্লে ইউনিট (segment display unit)কে interface করার জন্য এবং একটি চক্রীয় ফ্যাশনে (cyclic fashion) 0 থেকে 9 পর্যন্ত অক্ষরগুলি (characters) 1.5 সেকেন্ডের দেরিতে (delay) প্রদর্শন করার জন্য। এবং আমি এই STM board টি ব্যবহার করে ডেটা আউটপুট পিনের (pin) A থেকে G পর্যন্ত অংশ (segment) Arduino connector টির D2 থেকে D8 এর উপর সংযুক্ত করব।

(Refer Slide Time: 03:10)

Example 1: Connection Diagram

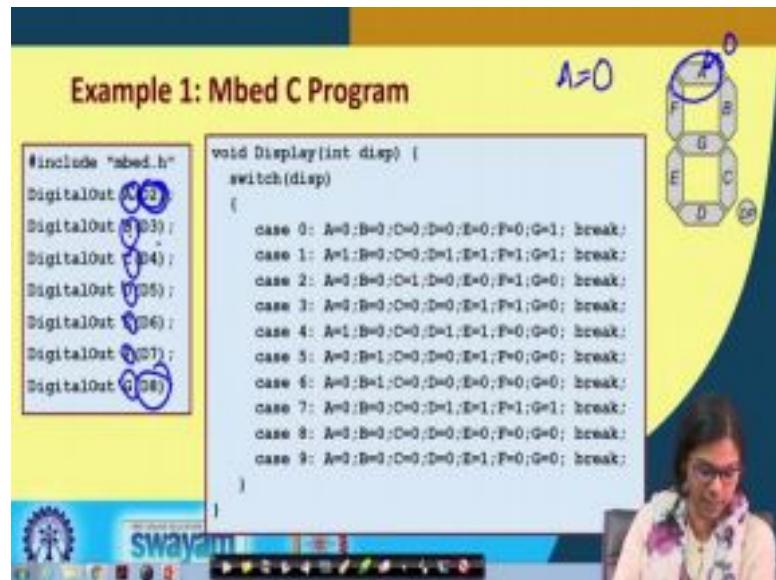


এটি সাধারণ সার্কিট ডায়াগ্রাম। আপনি দেখতে পাচ্ছেন এটি common port. এটি একটি সাধারণ anode 7 segment display, যেখানে common port টি রোধের মাধ্যমে 5V এবং অন্যান্য সমন্ত্বনা port গুলির সাথে সংযুক্ত থাকে, যা A, B, C, D, E, F, এবং G, এই STM board-র D2 থেকে D8 pin এর সাথে সংযুক্ত রয়েছে।

আপনি সংযোগটি দেখতে পারেন: A D2-এর সাথে সংযুক্ত, B D3-এর সাথে সংযুক্ত, C D4 এর সাথে

সংযুক্ত, D D5 এর সাথে সংযুক্ত, E D6 এর সাথে সংযুক্ত, F D7 এর সাথে সংযুক্ত, এবং G D8 এর সাথে সংযুক্ত রয়েছে। আমরা এইভাবে সংযোগটি করেছি, আপনার পক্ষে একই সংযোগ তৈরি করার মতো কোনও কঠোর নিয়ম নেই, তবে আমরা এইভাবে এটি সংযুক্ত করেছি।

(Refer Slide Time: 05:06)



এখন, আমরা mbed C code টা দেখব। যেমনটি আমি আপনাকে বলেছি, আমরা A, B, C, D, E, F এবং G এর সাথে সংযুক্ত সেগমেন্টগুলি (segment) থেকে শুরু করে D2 থেকে D8 উপর্যুক্ত পিনগুলি (pin) ব্যবহার করেছি এবং একইভাবে আমরা ডিজিটাল আউটপুট পোর্ট এর জন্যও এই নামগুলি দিয়েছি।

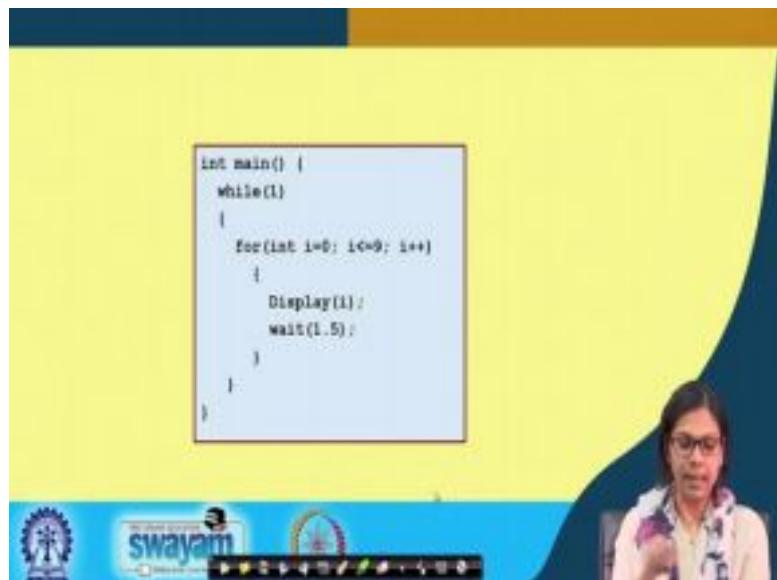
এখন, এটা একটা common anode display; কখন এই অংশটা (segment) জ্বলবে? সাধারণ point টা Vcc-তে যুক্ত থাকে, যখন আমরা এই লাইনে 0 মান দেব এই অংশটা (segment) তখন জ্বলবে। সুতরাং, D2 কে A এর মধ্যে দিয়ে পাঠাতে হবে, A তে 0 পাঠালে শুধুমাত্র এই নির্দিষ্ট অংশটা জ্বলে উঠবে। সুতরাং, জ্বলার জন্য আপনাকে 0 পাঠাতে হবে, কারণ এটি হলো একটি common anode; যদি এটা common cathode হত, আপনাকে উল্টোটা করতে হতো।

এখন, 9 টি কেস(case) রয়েছে কারণ আমি আপনাকে যেমন বলেছি, আমি 0 থেকে 9 অবধি দেরি(delay) করে দেখব। আসুন প্রথমে কেস 0 বিবেচনা করা যাক, 0 কী? আপনি দেখতে পাচ্ছেন যে কতগুলি বিভাগ(segment) আলোকিত হবে, G ব্যতীত সমস্ত বিভাগগুলি। সুতরাং, G বিভাগের মান 1 হবে আর অন্য সব বিভাগ যাদের আপনি জ্বালাতে চান, তাদেরকে একটি 0 মান পাস করতে হবে। একইভাবে এটি আমি A0, B0, C0, D0, E0, F0 করেছি, তবে G হল 1।

একইভাবে, 1 এর জন্য, আপনার B ও C এর জ্বলা দরকার। সুতরাং **B ও C তে একটি 0** মান থাকা উচিত অন্য গুলিতে 1 মান থাকা উচিত। সুতরাং আপনি অন্যগুলোর জন্যও সেটি করতে

পারেন। **মুতৰাং, এটা সেই কোড(code) যা আপনাকে লিখতে হবে।**

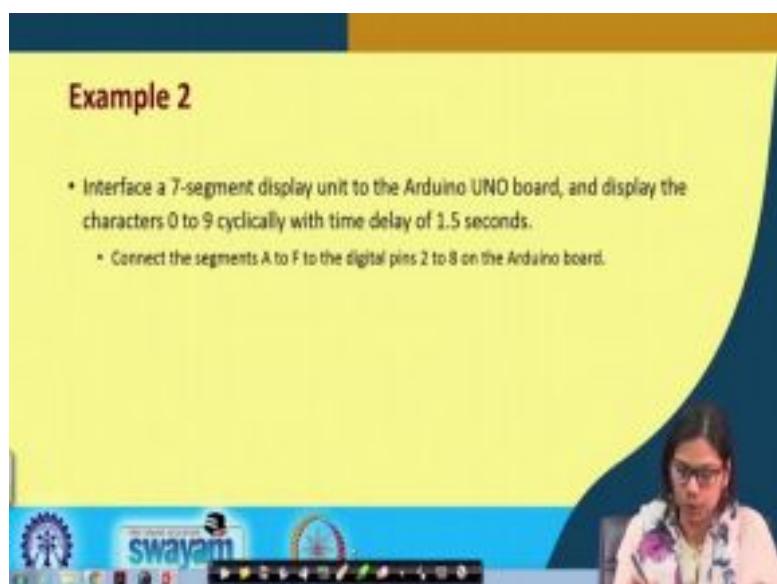
(Refer Slide Time: 07:47)



while(1) এর মধ্যে যে for loop আছে তাতে আমরা i এর মান 0 থেকে 9 অবধি পরিবর্তন করি। আমরা সংখ্যাটি দেখানোর জন্য Display(i) কে call করছি। তারপর 1.5 সেকেন্ডের একটি বিলশ্ব(delay) আছে।

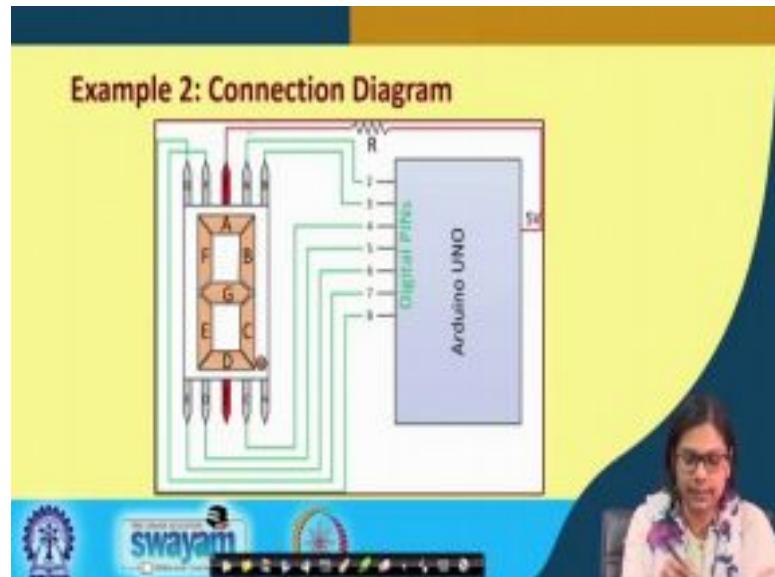
এটা একটা অবিচ্ছিন্ন loop এ চলতে থাকে।

(Refer Slide Time: 09:33)

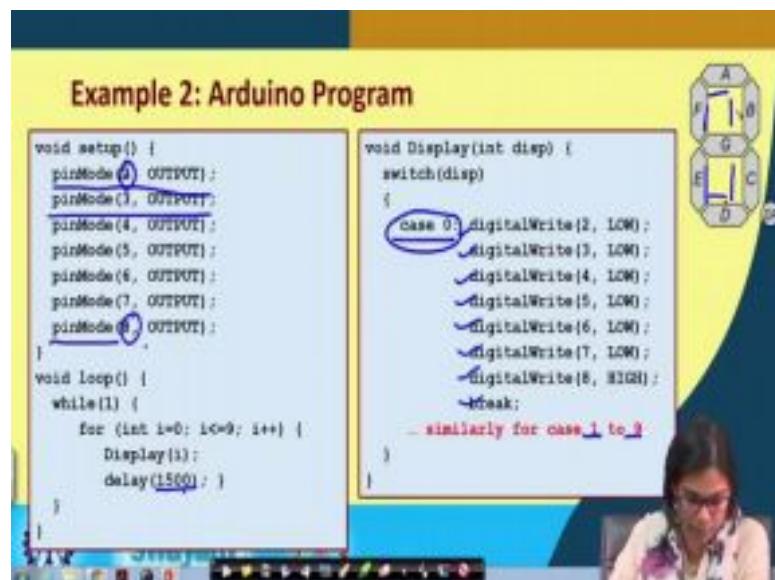


পরবর্তী উদাহরণটা একই ধরনের, কিন্তু Arduino board দিয়ে। আমাদের আগের উদাহরণটা ছিল STM board দিয়ে, এটা Arduino board দিয়ে। আপনাকে আবার এখানে এই অংশটা A থেকে F হয়ে G পর্যন্ত এবং Arduino board এ D2 থেকে D8 পর্যন্ত digital pin যুক্ত করতে হবে।

(Refer Slide Time: 10:18)

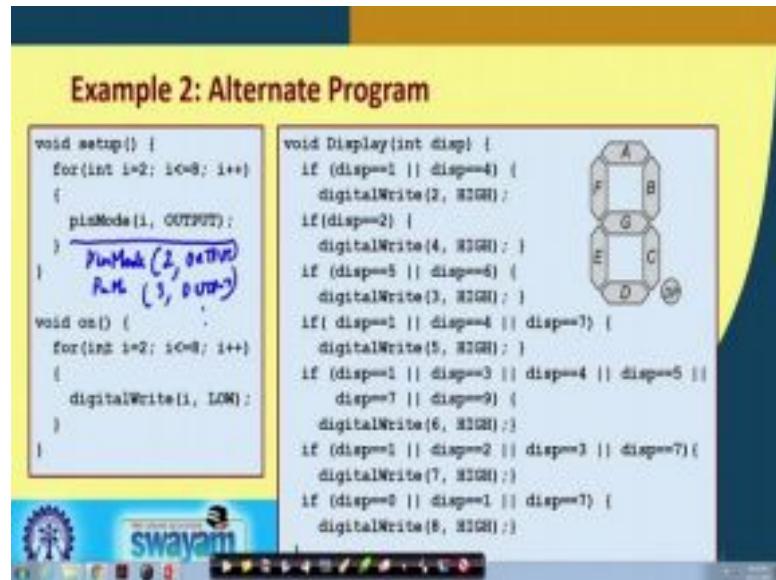


আমরা circuit diagram টা আবার দেখবো, যা প্রায় একই ধরনের।
(Refer Slide Time: 10:56)



এখানে, code টা দেখুন। setup পর্বে (phase), আমাদেরকে প্রতিটি pin এর জন্য pin mode বলতে হবে, আমরা pin 2 থেকে pin 8 পর্যন্ত ব্যবহার করছি। সুতরাং, আমাদের এই pin গুলোর প্রতিটির জন্য ও সব output এর জন্য pin mode বলতে হবে। এবং loop পর্বে, আমরা কি করছি? আমরা আগের code এ যা করেছি এখনো অনেকটা তাই করছি।

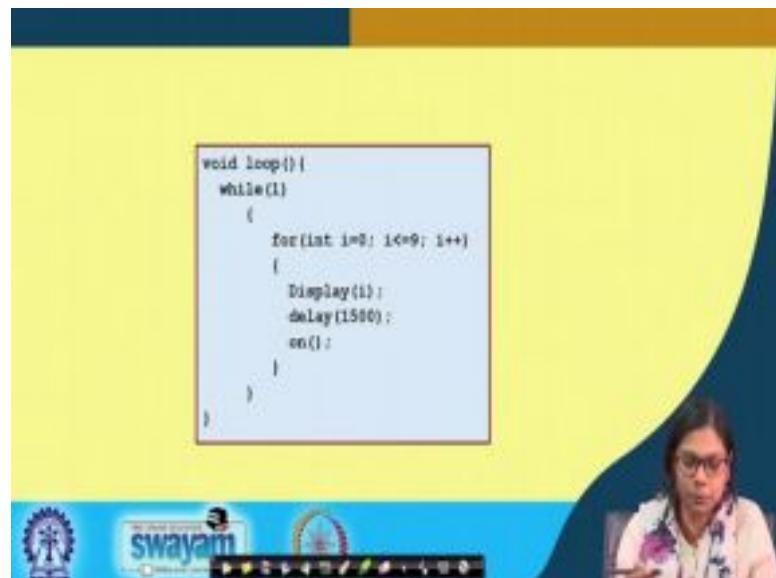
(Refer Slide Time: 13:05)



একটি দক্ষ এবং বিকল্প প্রোগ্রাম এর মতো কিছু হতে পারে। `for i=2; i<=8; i++`, আমরা পিনের মোডগুলি সেট করছি। **একইভাবে পরবর্তীতে for লুপের জন্য, আমরা 8 টি ডিজিটাল আউটপুট পিনগুলি LOW তে শুরু করছি।**

ডিসপ্লে ফাংশনে(display function) সংখ্যার উপর নির্ভর করে, আমরা কয়েকটি port pinগুলিতে HIGH লিখছি।

(Refer Slide Time: 16:48)



সুতরাং এখানে, loop বিভাগে আমরা পূর্ববর্তী কোডের(code) সাথে খুব অনুরূপ কিছু করি। কোডটি স্ব-বর্ণনামূলক(self-explanatory)।

(Refer Slide Time: 17:48)

Example 3

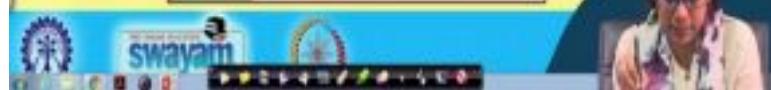
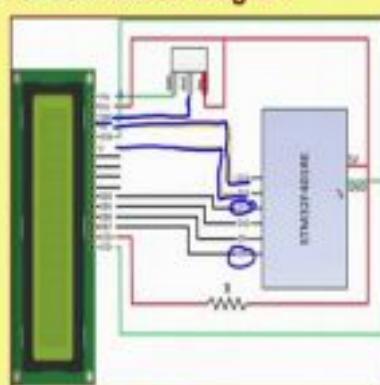
- Interface a LCD display unit to the STM32 board in 4-bit mode, and display a fixed message on the screen.
 - Connect the four data lines to D8 to D11 on the Arduino connector.
 - Connect the register select (RS) input to D13.
 - Connect the enable (E) input to D12.



এখন আমি একটি LCD unitকে STM32 boardটিতে 4-বিট(bit) মোডে(mode) ইন্টারফেস(interface) করব এবং LCDতে একটি নির্দিষ্ট বার্তা(fixed message) প্রদর্শন করব। আমরা Arduino সংযোগকারী টিতে 4 টি লাইন, D8-থেকে D11-এতে সংযুক্ত করিব। এবং আমরা এই register select টিকে D13 ইনপুট পিনের(pin) সাথে সংযুক্ত করিব। এবং আমরা এই enable ইনপুটটিকে D12 এর সাথে সংযুক্ত করিব। এই সংযোগগুলি STM32 এর জন্য প্রযোজনীয়। সংযোগটি এভাবে চলে।

(Refer Slide Time: 18:30)

Example 3: Connection Diagram



আপনি যদি LCDটি দেখেন, এটিতে এই পিনগুলি রয়েছে। সংযোগগুলি দেখানো হয়েছে। Read/write স্থায়ীভাবে ground-র সাথে সংযুক্ত, কারণ আমরা এখানে কেবল লিখছি।

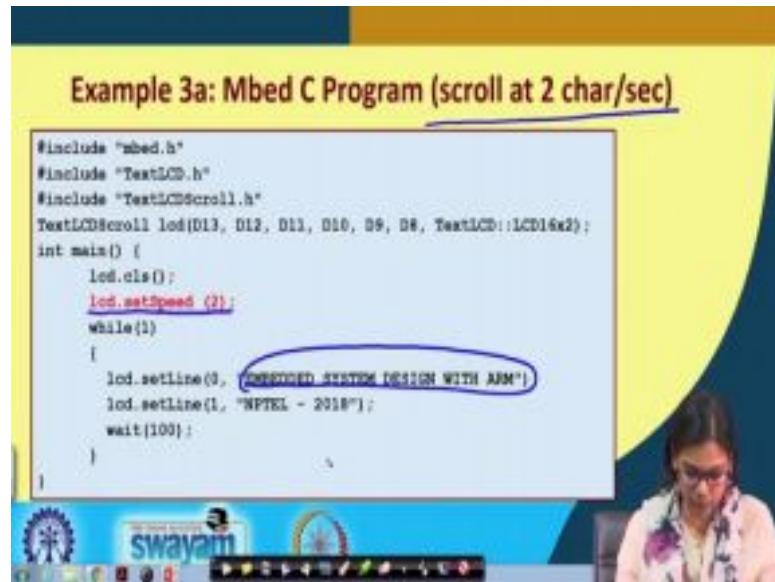
(Refer Slide Time: 20:37)

```
#include "mbed.h"
#include "TextLCD.h"
#include "TextLCDScroll.h"
TextLCDScroll lcd(D13, D12, D11, D10, D9, D8, TextLCD::LCD16x2);
int main() {
    while(1)
    {
        lcd.setLine(0, "EMBEDDED SYSTEM");
        lcd.setLine(1, "NPTEL - 2018");
        wait(20);
    }
}
```

এখন, আমরা এখানে mbed প্রোগ্রাম দেখব। শুধুমাত্র mbed.c কে include করা ছাড়াও, আমাদের অন্যান্য header fileগুলো include করা দরকার। LCD interfacing করার জন্য যে দুটো header file লাগবে সেগুলো হলো TextLCD.h এবং TextLCDScroll.h, আর আমাদেরকে TextLCDScroll নামের একটা function ব্যবহার করতে হবে। আমরা lcd নামের একটি object তৈরি করব ও সংযোগের জন্য data লাইনগুলিকে নির্দিষ্ট করব। এটা data bit line এর জন্য এবং কি প্রকারের LCD আমরা ব্যবহার করব? LCD16x2 দুটি লাইনের প্রতিটিতে 16 টি করে অক্ষর(character) দেখাতে পারে।

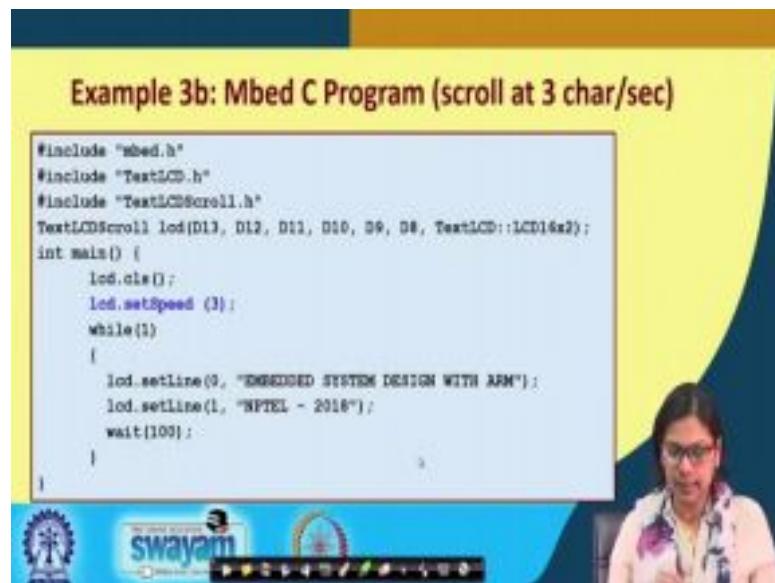
পরেরটি while(1) এর মধ্যে থাকা main program. আমরা LCD এর দুটি লাইনে দুটো character strings দেখাচ্ছি, 0 মানে প্রথম লাইন আর 1 মানে দ্বিতীয় লাইন। এরপর একটি 20 সেকেন্ডের বিলম্ব(delay) আছে।

(Refer Slide Time: 22:33)



এটি পরবর্তী প্রোগ্রাম, যা মেটামুটি অনুকরণ, তবে এখানে কেবলমাত্র পার্থক্যটি হ'ল, যে scrolling-এর গতিটি আমরা উল্লেখ করছি। আমরা উল্লেখ করছি যে string টি 16 টি অক্ষরে ফিট(fit) না হলে প্রতি সেকেন্ডে 2 টি করে অক্ষর স্ক্রোল(scroll) করা হবে।

যদি পার্থক্যটি 16 টির চেয়ে কম অক্ষরের বা 16 অক্ষরের সমান হয় তবে এটি স্ক্রোলিং(scrolling) ছাড়াই প্রদর্শিত হবে। তবে এটি এর চেয়ে বেশ হলে নির্দিষ্ট পার্থক্যটি স্ক্রোলিং(scrolling) হবে।
 (Refer Slide Time: 23:44)



এটি অন্য একটি অনুকরণ প্রোগ্রাম, আমাদের স্ক্রোল(scroll) গতি প্রতি সেকেন্ডে ঠিক 3 টি করে অক্ষর।
 (Refer Slide Time: 24:07)

Example 4

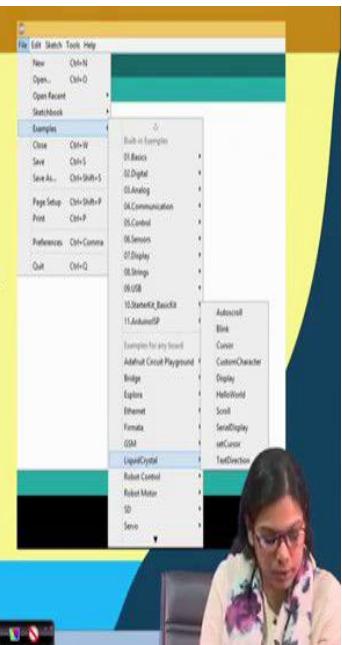
- Interface a LCD display unit to the Arduino UNO board in 4-bit mode, and display a fixed message on the screen.
 - Connect the four data lines to pins 8 to 11.
 - Connect the register select (RS) input to pin 13.
 - Connect the enable (E) input to pin 12.



এরপরে আমরা দেখাবো, আমরা কীভাবে একই LCD টিকে 4-বিট মোডে(mode) Arduino UNO বোর্ডের(board) সাথে ইন্টারফেস(interface) করব এবং স্ক্রিনে(screen) একটি নির্দিষ্ট বার্তা প্রদর্শন করব। সুতরাং, আমরা এখানে খুব বেশি জিনিস নিয়ে খেলছি না। সংযোগগুলি দেখানো হয়।

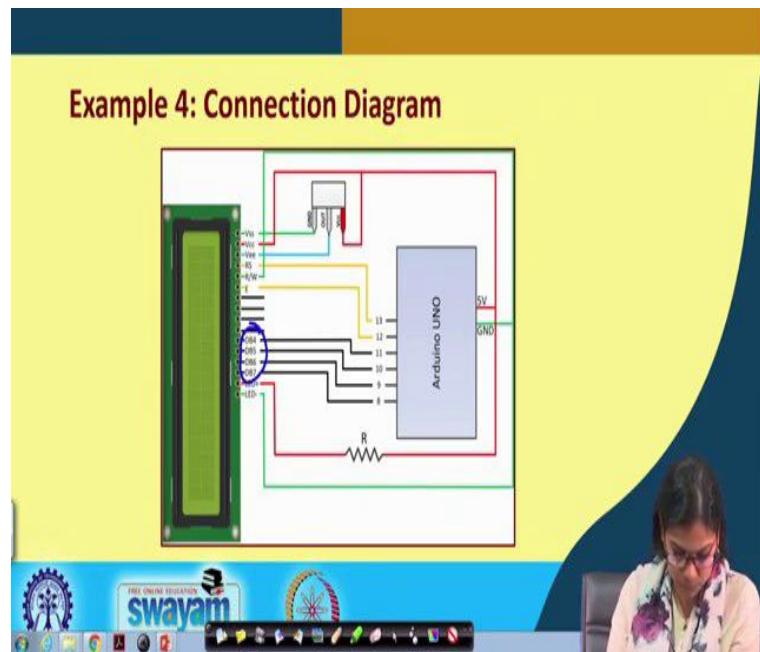
(Refer Slide Time: 24:49)

- In arduino *TextLCD* function is already imported it can be directly accessible in the example section.
- Open arduino IDE and follow the steps : >Examples-> LiquidCrystal->Choose any Example (Hello World).
- Modify the program according to your need.
- The description of the connection will load by default.

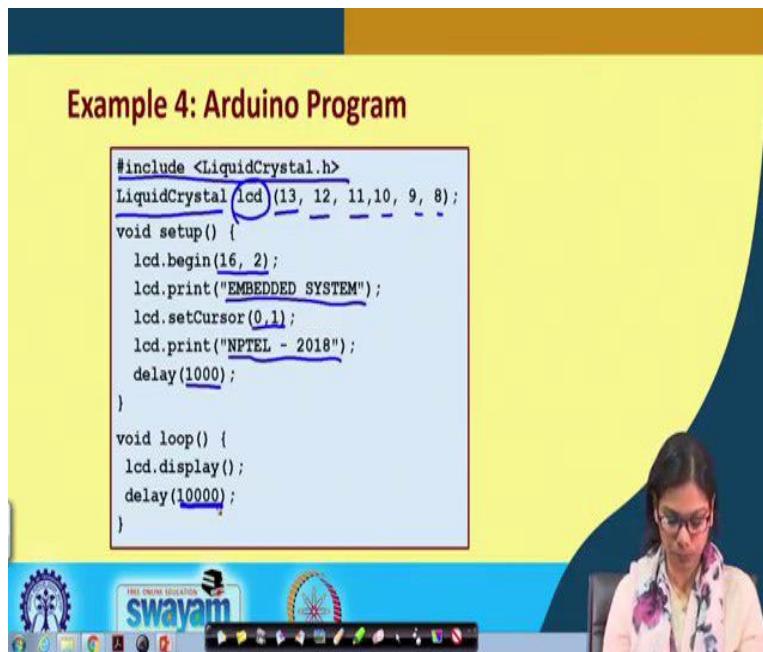


একইভাবে, আপনি ইতিমধ্যে দেখেছেন যে আগের STM-এর code-র জন্য, আমরা কয়েকটি header ফাইল ব্যবহার করছিলাম। Arduino-র জন্য আমরা একইভাবে কয়েকটি header ফাইল অন্তর্ভুক্ত করেছিলাম যা নিম্নলিখিত ক্রমে(order) আপনাকে করতে হবে, আমরা জানি এই *textLCD* function টি ইতিমধ্যে আমদানি করা হয়েছে, উদাহরণ বিভাগে এটি সরাসরি উপলব্ধ করা যেতে পারে।

(Refer Slide Time: 25:56)



এটিই সংযোগ ডায়াগ্রামটি, যা পূর্বেরটির সাথে একই রকমের।
(Refer Slide Time: 26:29)



এটাই code. সুতরাং, এখানে ওই নির্দিষ্ট header file LiquidCrystal.h কে অন্তর্ভুক্ত করতে হবে। LiquidCrystal class এর জন্য, আপনি একটি object lcd তৈরি করবেন। এবং সংযোগের জন্য pin সংখ্যাগুলো বলবেন। এই setup পর্বে, আপনি প্রাথমিকভাবে নিম্নলিখিত কাজ করবেন। lcd.begin বলে দেয় যে এটা একটা 16 x 2 প্রদর্শন(display)। lcd.print lcd.setCursor(0,1) দিয়ে, string EMBEDDED SYSTEM টি বলে দেয়। **cursorটি এখন এই অবস্থানে চলে যাবে**

এবং এটি NPTEL - 2018 প্রিন্ট করবো। এবং আমরা 1 সেকেন্ডের বিলম্ব দিই এবং একইজিনিস

পুনরাবৃত্তি হয়।

সুতরাং, আমরা এই বক্তৃতাটির শেষে এসে পড়েছি। এই বক্তৃতায় আমি দুটো output যন্ত্রের ব্যাপারে আলোচনা করেছি, একটি হল 7 segment, অপরটি হল LCD. এবং, আমি আপনাকে Arduino আর STM উভয় দিয়েই circuit diagram দেখিয়েছি, এবং এই দুটি যন্ত্র ব্যবহার করে আপনি কিভাবে কোন পাঠ্য প্রদর্শন করাতে পারেন তা দেখিয়েছি। আমরা এগিয়ে চলবো এবং পরবর্তী বক্তৃতা গুলিতে আরো কিছু interfacing এর পরীক্ষা দেখাবো।

ধন্যবাদ।

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya
Lecture – 23
Interfacing with 7-Segment LED and LCD Displays (Part II)

এই বক্তৃতায় আমরা আপনাদের কে এলসিডি(LCD) এর সঙ্গে 7 সেগমেন্ট(segment)

এলইডি(LED)-এর ইন্টারফেসিং প্রদর্শন করাব। আমরা বোর্ডের দুটি সেট নিয়ে আপনাদের কে পরীক্ষা গুলি প্রদর্শন করাব। আমরা প্রথমে আপনাদের কে মূলতঃ 7 সেগমেন্ট(segment) এলইডি এর ইন্টারফেসিং এর ওপর পরীক্ষা-নিরীক্ষা দেখাবো। আমি ইতিমধ্যে আলোচনা করেছি, 7 সেগমেন্ট(segment) ডিসপ্লে কি, এটি কী ভাবে কাজ করে। এখানে দুপ্রকারের 7 সেগমেন্ট(segment) ডিসপ্লে(display) আছে একটি সাধারণ অ্যানোড(anode), অন্যটি সাধারণ ক্যাথোড।

এই পরীক্ষায় আমরা সাধারণ অ্যানোড(anode) 7 সেগমেন্ট(segment) ইউনিট নেব। আমরা একটি সাধারণ অ্যানোড(anode) ডিসপ্লে(display)তে একটি সাধারণ পয়েন্ট এর সঙ্গে একটি প্রতিরোধক এর মাধ্যম এ Vcc এর সঙ্গে সংযুক্ত করব এবং যে কোনো এলইডি কে জ্বালাতে আমাদের কে একটি 0 পাস করাতে হবে। এই পরীক্ষা তে আমাদের কে যেটা করতে হবে সেটা হল কোন উপায় এলইডি কে জ্বালানো, আমরা 0 থেকে শুরু করব এবং এটা 1 হবে তারপর 2 এইভাবে 9 পর্যন্ত, এবং আবার 0 তে ফিরবে। সুতরাং এটা এক প্রকার কাউন্টার যেটা 0 থেকে 9 পর্যন্ত গণনা করবে এবং আবার 0 তে ফিরবে।

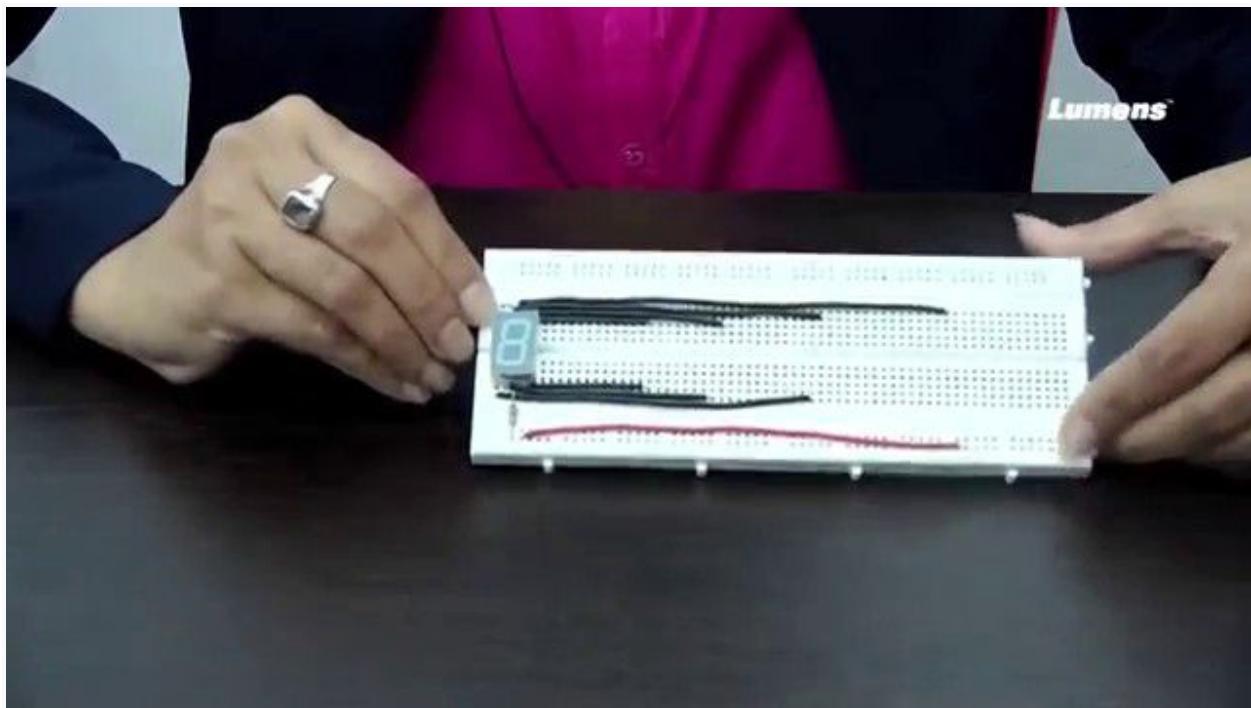
(Refer Slide Time: 02:27)



সুতৰাং দেখুন এটা একটি 7 সেগমেন্ট(segment) ডিসপ্লে, আপনি অবশ্যই এটা অনেক ক্ষেত্রে দেখবেন। আমরা দেখি এর কার্যকারিতা কি বা এই সেগমেন্ট(segment) গুলি কি। সুতৰাং এগুলো হচ্ছে 7 সেগমেন্ট(segment) ডিসপ্লে এর সেগমেন্ট(segment)।

আপনারা এই পিনগুলো দেখুন। এটা একটি সাধারণ অ্যানোড(anode) 7 সেগমেন্ট(segment), আমরা এই বিশেষ সাধারণ পয়েন্ট থেকে একটি প্রতিরোধক এর মাধ্যম এ Vcc এর সাথে সংযুক্ত করব।

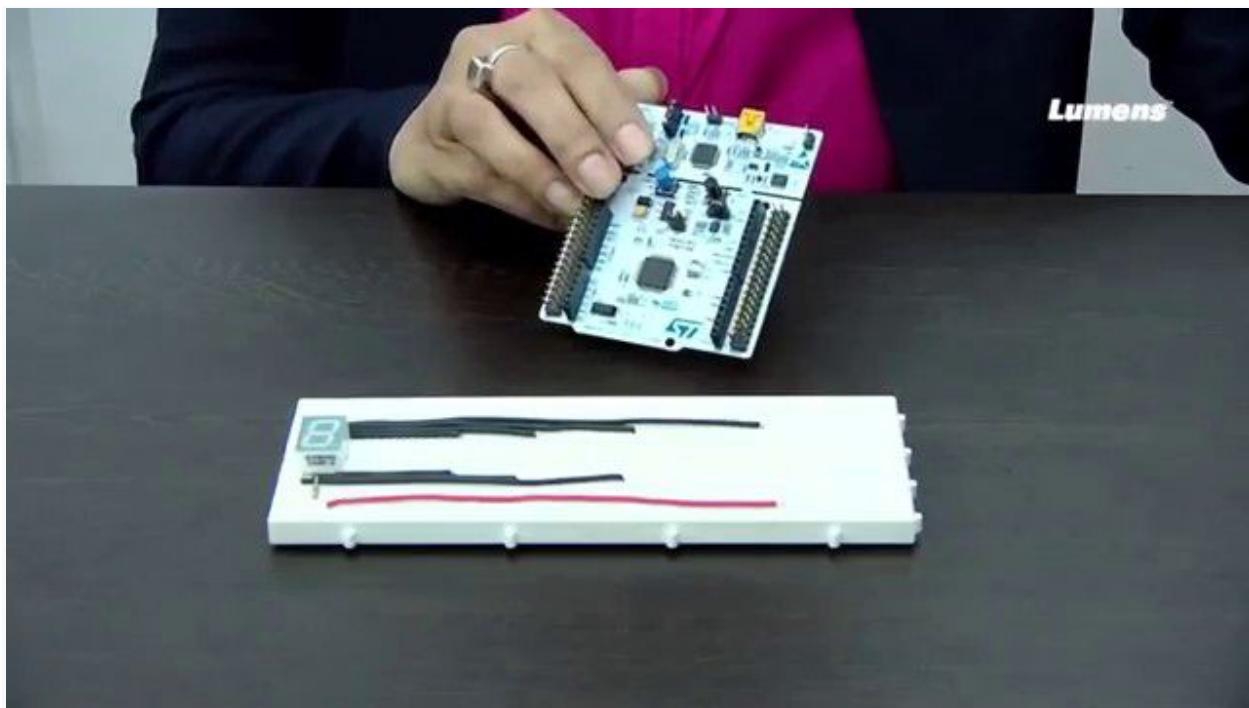
(Refer Slide Time: 03:57)



সুতরাং প্রথমেই আমি আপনাদের কে STM বোর্ডের সঙ্গে সংযোগ দেখাবো। এবং সাধারণ পয়েন্ট থেকে একটি প্রতিরোধক এর মাধ্যম এ আমি এটাকে Vcc এর সঙ্গে সংযুক্ত করব। এছাড়া আমি সেগমেন্ট(segment) ওলো কে উভয় বোর্ড এর কিছু ডিজিটাল আউটপুট পিন এর সঙ্গে সংযুক্ত করব।

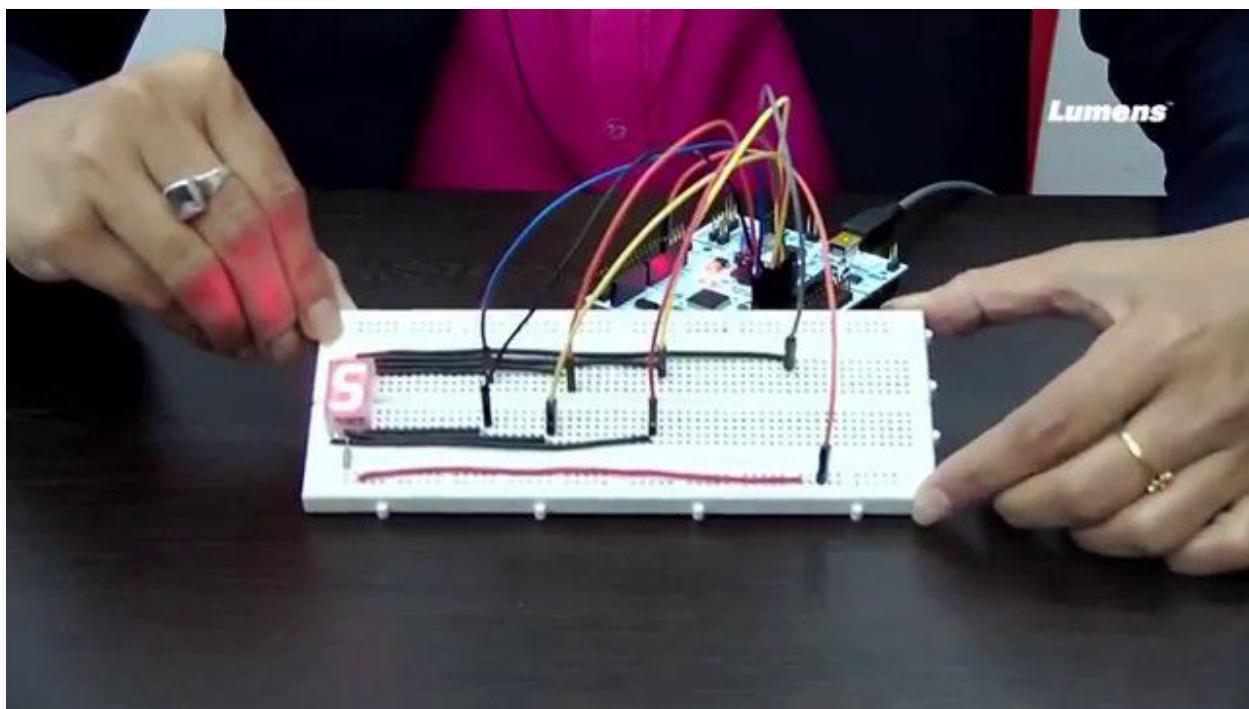
আমি ইতিমধ্যে আপনাদের কে বলেছি যে আমি মূলতঃ D2 থেকে D8 পর্যন্ত পিন ব্যবহার করব এবং সাধারণ পয়েন্টটি Vcc এর সাথে সংযুক্ত হবে এবং তারপরে আমি কোডটি লিখব যেখানে প্রতিটি সেগমেন্ট(segment) 1 সেকেন্ড এর চেয়ে সামান্য কিছুটা বেশি ছ্বলবে এবং তারপরে এটি পরবর্তী একটি-তে চলে যাবে এবং এই ভাবে চলবে। সুতরাং, আপনাকে এভাবে সংযোগ তৈরি করতে হবে। প্রথমে আমাকে এটি এসটিএম(STM) বোর্ডের সাথে সংযুক্ত করতে দিন।

(Refer Slide Time: 05:18)



এটা আমার STM বোর্ড; আমি যেমন বলেছিলাম সেভাবে এটাকে D2 থেকে D8 সংযুক্ত করব।

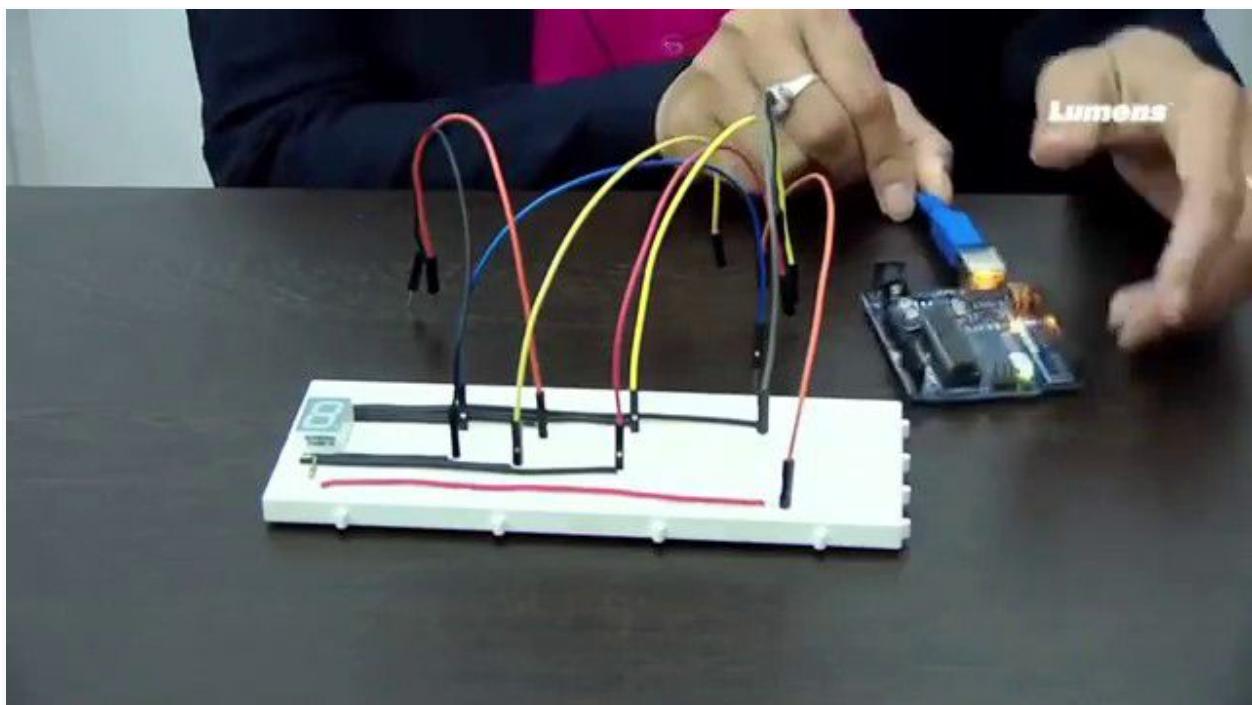
(Refer Slide Time: 05:31)



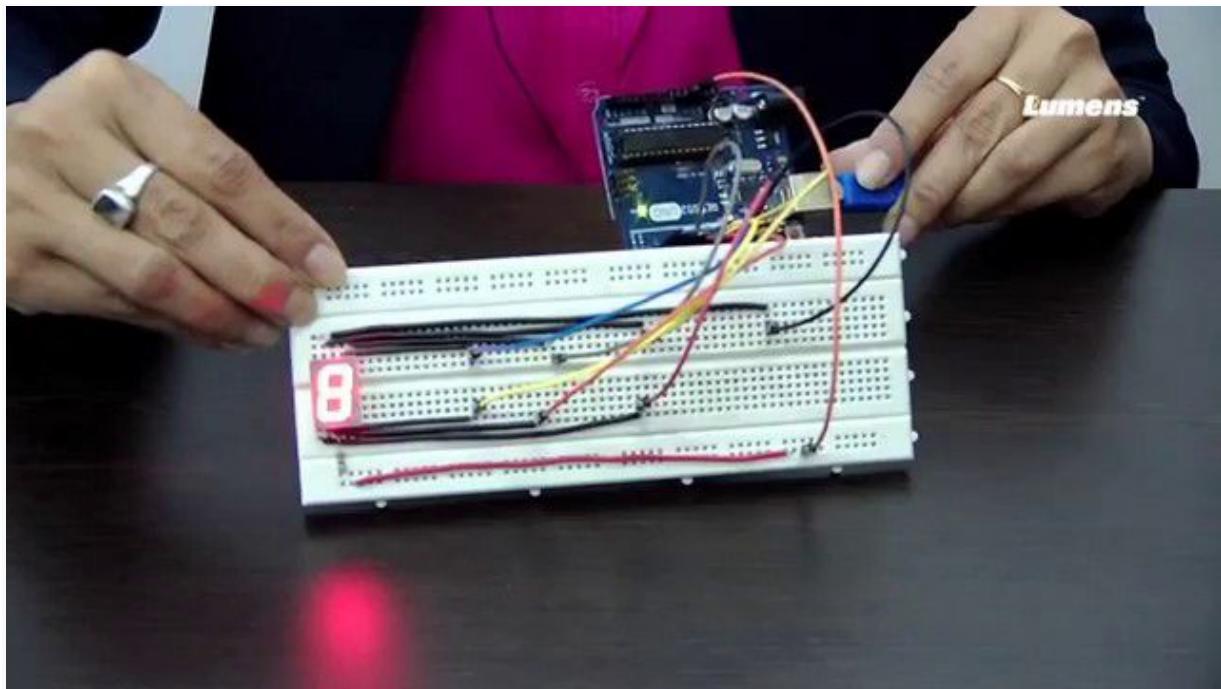
এটাকে D3 তে সংযুক্ত করব, এরপর A থেকে D2, B থেকে D3, C থেকে D4, D থেকে D5, E থেকে D6, F থেকে D7, G থেকে D8 সংযুক্ত করব। সাধারণ পয়েন্ট থেকে একটি প্রতিরোধক এর মাধ্যমে এটা Vcc সঙ্গে সংযুক্ত হবে। এখন আমি কোডটিকে STM বোর্ডে ফেলবো।

এখন আপনারা দেখতে পাবেন কি হচ্ছে। অংকগুলি একটার পর একটা প্রদর্শিত হচ্ছে। In এখন আমি Arduino board এর সাথে একই কাজ করব।

(Refer Slide Time: 09:33)



(Refer Slide Time: 09:40)



প্রথমে আমাকে সংযুক্ত করতে দিন, এটা হল VCC, এবং এরপর আমি d2 থেকে d8 সংযুক্ত করব।

এখন এই সংযোজকটি কে সংযুক্ত করব এবং কোডটি কে স্থাপন করব। এখন দেখুন আমি এই Arduino বোর্ডে কোডটি রেখেছি, এসটিএম বোর্ডের জন্য আমি একই কোড লিখেছি এবং এটি জ্বলজ্বল করছে।

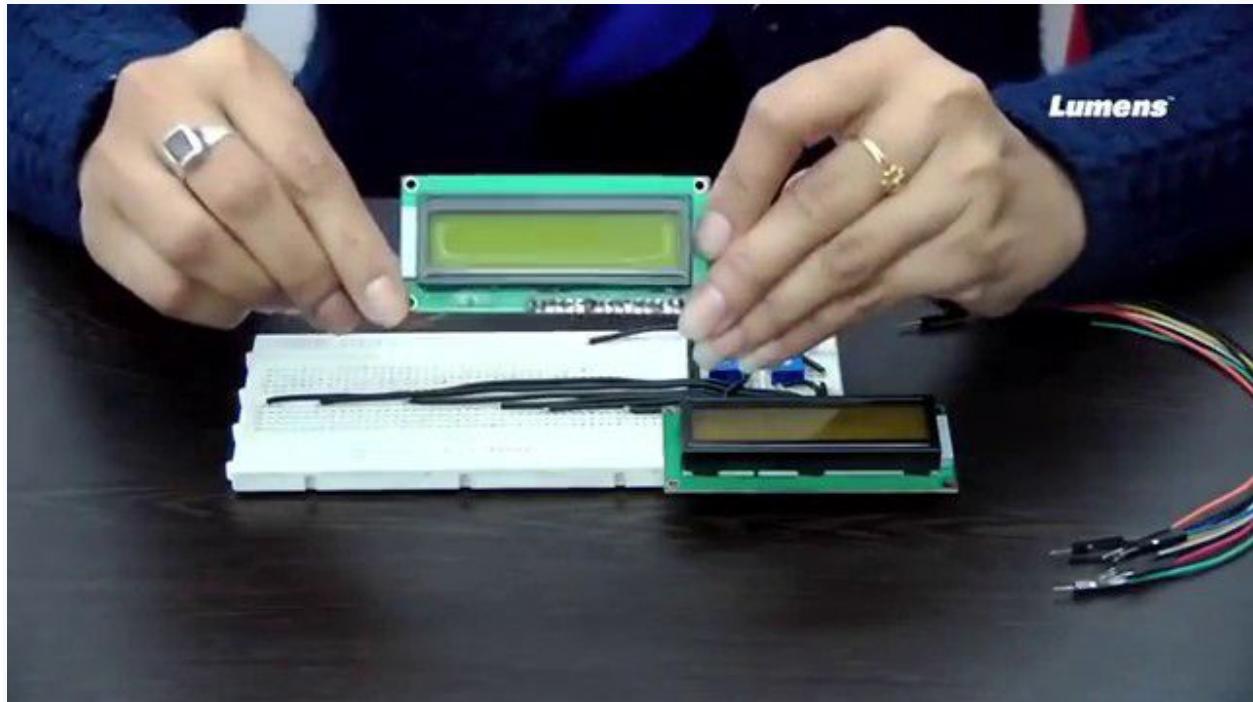
এরপরে আমি আপনাদের কে দেখাচ্ছি যে আমরা কী ভাবে এলসিডি ইন্টারফেস করতে পারি, একই এলসিডি যা আমরা ইতিমধ্যে আলোচনা করেছি।

এখন এই নির্দিষ্ট এলসিডি প্রতিটি লাইনে 2 টি পার্থ্য প্রদর্শন করতে পারে, প্রতিটি লাইনে সর্বাধিক 16 টি অক্ষর প্রদর্শন করতে পারে এবং পার্থ্যটি 16 টি অক্ষরের বাইরে গেলে এটি স্লোল করে। আমরা দুটি প্রতিনিধি বোর্ড নিয়ে এটা কীভাবে করব তা নিয়েও আলোচনা করেছি, একটি হল এসটিএম বোর্ডের সাথে এবং অন্যটি হল Arduino বোর্ড।

উভয় বোর্ডের জন্য আমাদের কয়েকটি জিনিস মনে রাখতে হবে। এখানে কিছু লাইব্রেরি ফাংশন রয়েছে যা আমি ইতিমধ্যে আলোচনা করেছি, সেই ফাংশনগুলি আপনাকে অন্তর্ভুক্ত করতে হবে। আপনি কোডটি লেখার সময়

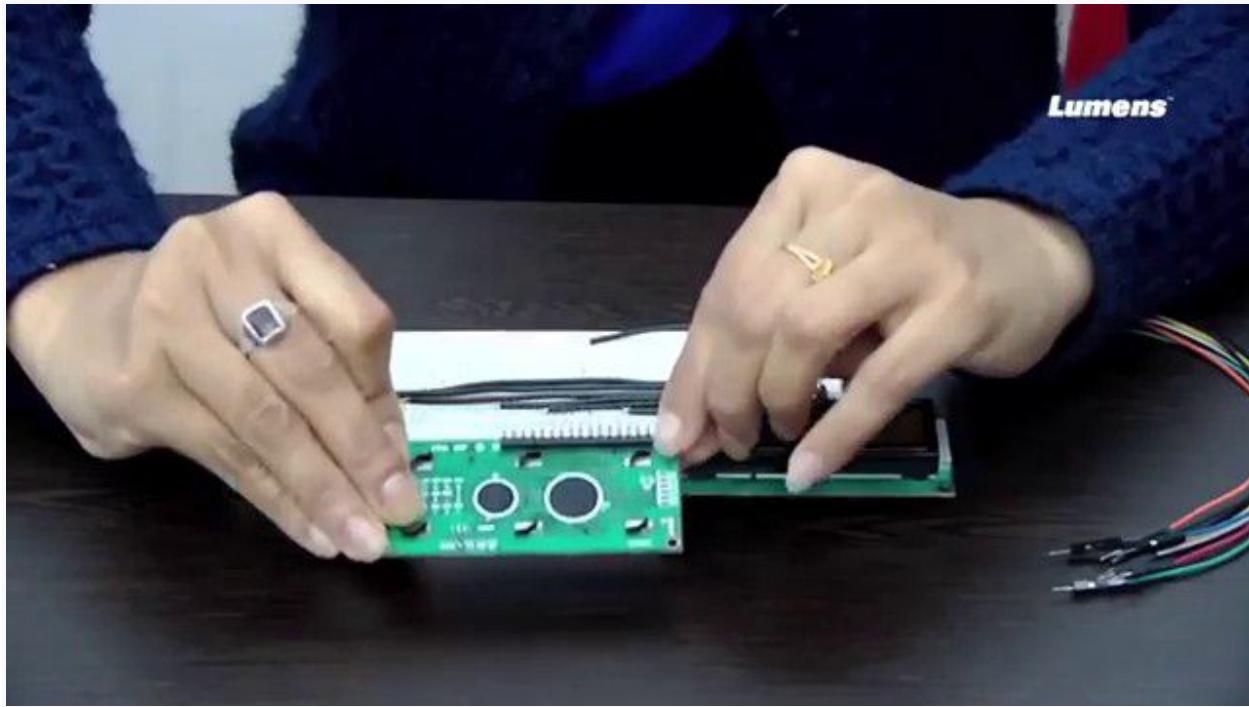
আপনাকে অবশ্যই নিশ্চিত করতে হবে যে আপনি ইতিমধ্যে এই ফাংশনগুলি অন্তর্ভুক্ত করেছেন। দুটি ফাংশন LCDdisplay এবং LCDscroll, আমরা ইতিমধ্যে আলোচনা করেছি।

(Refer Slide Time: 15:52)



এটি হল LCD এখন আমরা ইন্টারফেসিং করব; এটি 16 পিন পেয়েছে।

Refer Slide Time: 116:07)

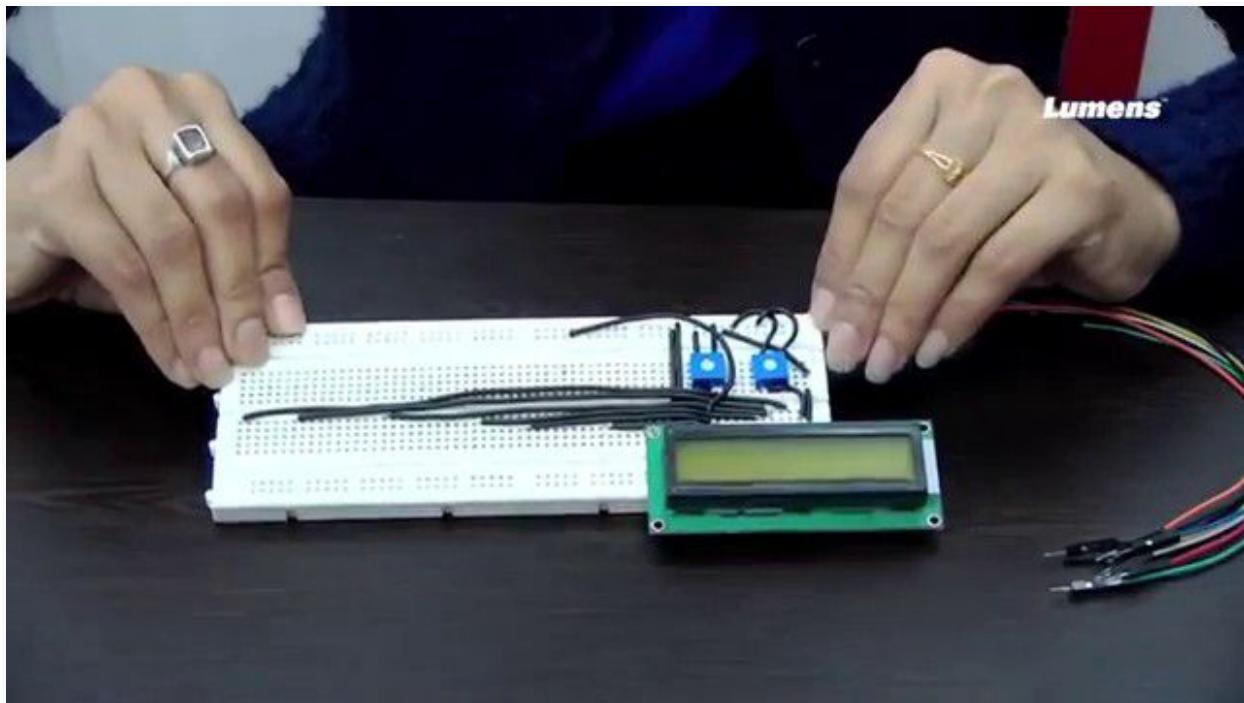


এগুলি মূলতঃ পিন। সুতরাং এই নির্দিষ্ট পিনটি 1 নম্বর, এরপর 2 নম্বর,, তারপর 3 নম্বর এবং এই ভাবেই 16 নম্বর পর্যন্ত।

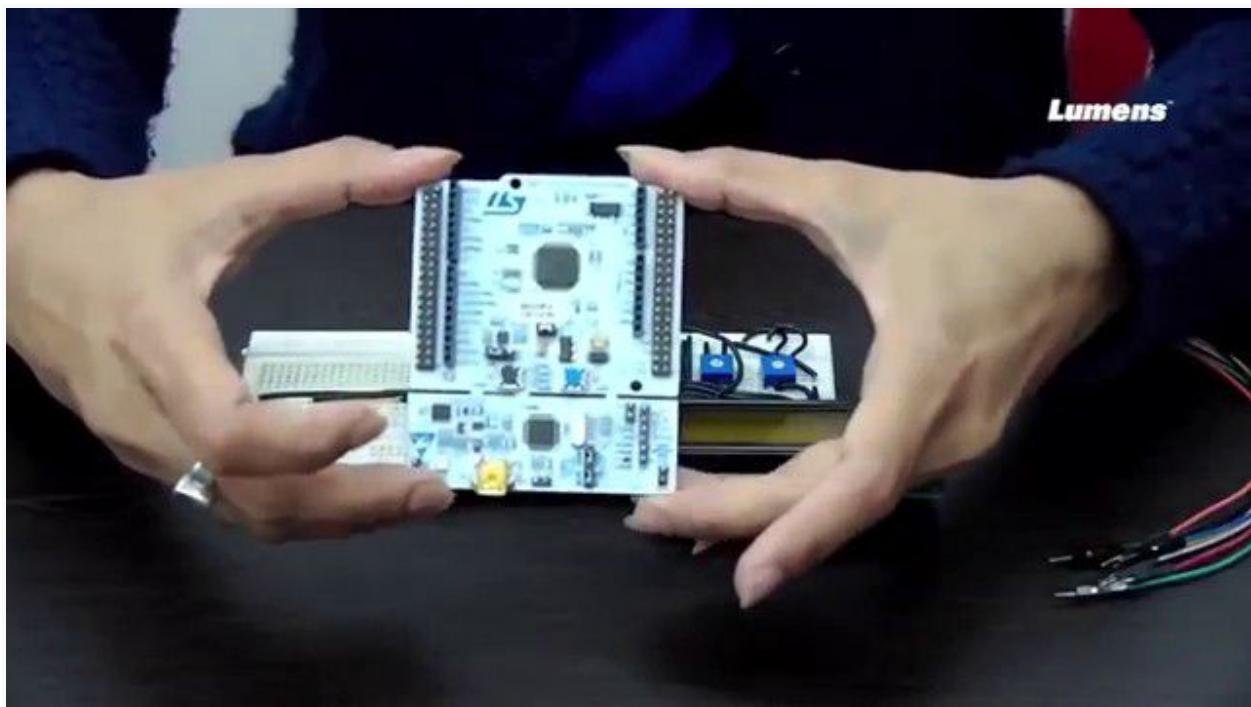
আমি ইতিমধ্যে সমস্ত পিন সম্পর্কে আলোচনা করেছি, তবে খুব সংক্ষেপে আমাকে বলতে দিন যে প্রথম পিনটি গ্রাউন্ডের সাথে সংযুক্ত থাকবে, পরের পিনটি Vcc সাথে সংযুক্ত হবে যেটা 5V, পরের পিন হ'ল কন্ট্রাস্ট কন্ট্রোল যা আমরা একটি পোটেনশিওমিটার ব্যবহার করে সংযোগ করব। এই পোটেনশিওমিটারের একটি প্রান্ত Vcc এর সঙ্গে সংযুক্ত হবে, অন্যটি গ্রাউন্ড এর সঙ্গে সংযুক্ত হবে এবং 3 নাম্বার পিন থেকে সংযোগ স্থাপন করব।

4 নম্বর পিনটি হ'ল প্রতিরোধক বা register যেটা d3 পিন এর সঙ্গে সংযুক্ত। এখন আমরা এইভাবে সংযোগটি করেছি, আপনারা বলুন সংযোগটি পরিবর্তন করতে চান কিনা এবং আপনারা এটিকে অন্য যে কোনও পিনের সাথে সংযুক্ত করতে চাইলে আপনাদের তা করার স্বাধীনতা রয়েছে। পরের টি হল read/write, এই ক্ষেত্রে আমরা শুধু লিখছি এবং তাই আমরা এটাকে গ্রাউন্ড এর সঙ্গে সংযুক্ত করেছি। পরবর্তী পিনটি হল enable যা d12 এবং তারপরে পরপর 4 বিট এসটিইএম(STM) বোর্ডের d11, d10, d9 এবং d8 পিনগুলির সাথে যুক্ত থাকে।

(Refer Slide Time: 18:13)



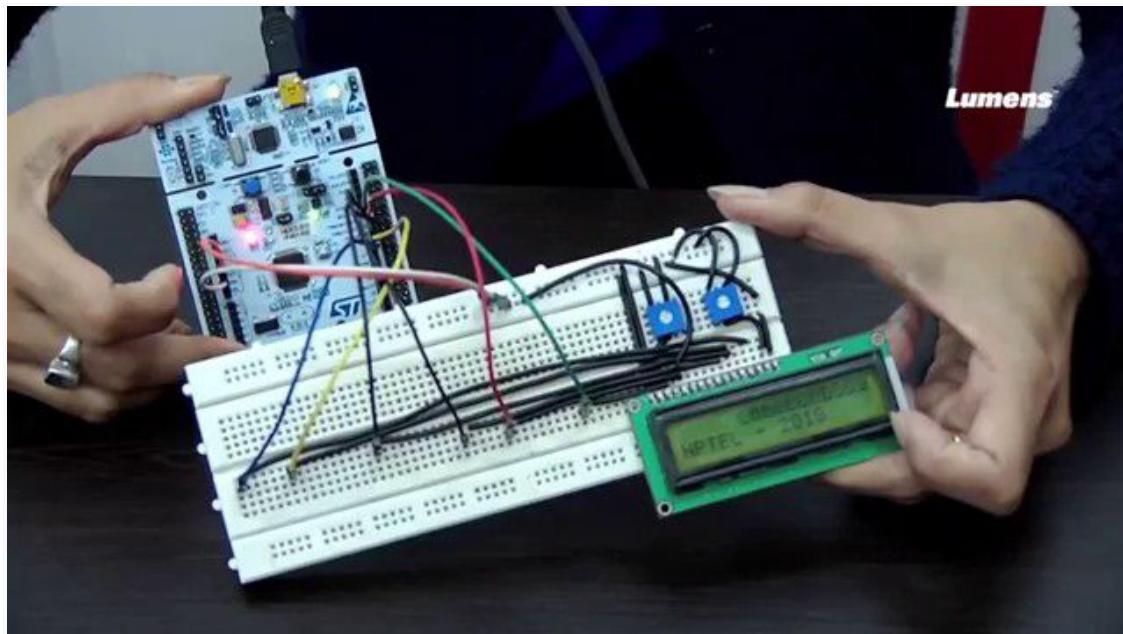
(Refer Slide Time: 18:37)



যেমন আমি বলেছিলাম, প্রথমে আমি আপনাদের কে এসটিএম(STM) বোর্ড দেখাবো; এই বোর্ডে আমরা সংযোগ গুলি করব এবং আপনারা ইতিমধ্যে এই বোর্ডের পিন গুলি সম্পর্কে জেনেছেন। সুতরাং এটি হল Vcc,

গ্রাউন্ড, এবং তারপর এই বিশেষ বোর্ডের অন্যান্য পিন গুলিও রয়েছে। আমি এগুলোকে একটার পর একটা সংযুক্ত করব।

(Refer Slide Time: 19:08)



প্রথমে আমাকে এখানে প্রথম পিনটি গ্রাউন্ড এবং অন্যটি Vcc সঙ্গে সংযুক্ত করতে দিন।

যেমনটা আমি বলেছিলাম এলসিডির প্রথম পিনটি গ্রাউন্ড এর সাথে সংযুক্ত, পরের পিনটি Vcc এর সাথে সংযুক্ত, পরবর্তী পিনটি এই পোটেনশিওমিটার এর মাধ্যমে সংযুক্ত রয়েছে।

এখন, পরবর্তীটি প্রতিরোধক বা resister হিসাবে d13 এর সঙ্গে সংযুক্ত করা হবে। d12 কে read/write এর সঙ্গে সংযুক্ত করা হবে; মূলতঃ read/write সরাসরি গ্রাউন্ড d12 এর সঙ্গে সংযুক্ত। Enable পিনটি d12 এর সঙ্গে সংযুক্ত। d3 কে প্রতিরোধক বা register হিসেবে নিন, এবং এটা হল enable পিন, এবং এই চারটি পিন মূলত ডেটা পিন।

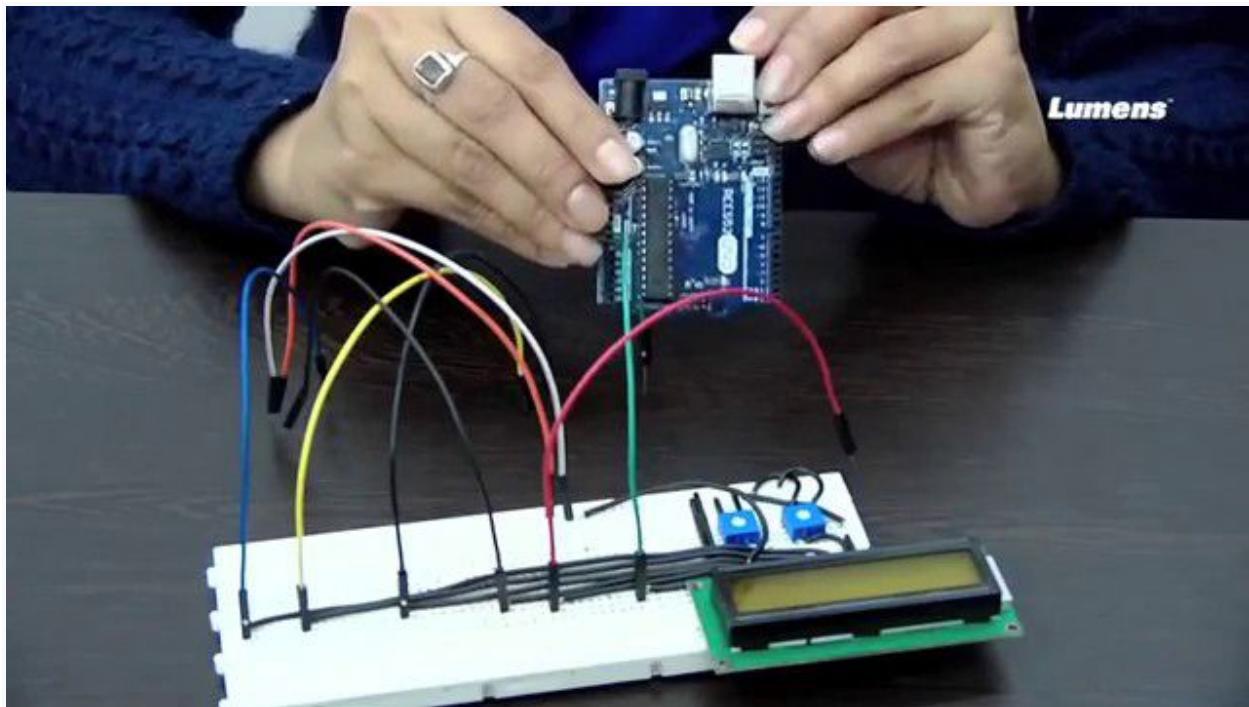
আমরা এই এলসিডির(LCD) সমস্ত পিন, পিন 1 থেকে শুরু করে পিন 16 পর্যন্ত বিভিন্ন পিনের সাথে এই এসটেম(STM) বোর্ডে এবং অভ্যন্তরীণভাবে গ্রাউন্ড Vcc কে সংযুক্ত করেছি। স্লাইডে ইতিমধ্যে আপনাদের কে দেখানো সার্কিট ডায়াগ্রামটি অনুসরণ করুন যেখানে আমরা ঠিক করেছি যে নির্দিষ্ট পিনটি এই নির্দিষ্ট বোর্ডের কোন পিনের সাথে সংযুক্ত হবে।

এবং ক্যাথোড সরাসরি গ্রাউন্ডে সংযুক্ত, এটি সংযোগ সম্পর্কে সমস্ত কিছু। আপনি যদি আপনার শেষে এটি তৈরি করতে চান তবে আপনার নিষ্পত্তিত জিনিসগুলির প্রয়োজন; আপনার নিষ্পত্তি জাম্পারের তারগুলি প্রয়োজন, আপনার পটেনশিওমিটার প্রয়োজন, আপনার প্রয়োজন লেলসিডি(LCD) এবং অবশ্যই এই বোর্ড। এখন আমি যে কোডটি করব যা আমি ইতিমধ্যে লিখেছি, এটি আমি এই বোর্ডে ফেলব। পার্থ্যটি "embedded system NPTEL 2018" প্রদর্শিত হচ্ছে।

এখন, আমি কেবল এই পার্থ্যটি অন্য কোনও পার্থ্যে পরিবর্তন করব, এবং আমাকে কী হবে তা দেখতে দিন। আমি একটি পার্থ্য "embedded system design with ARM" প্রবেশ করিয়েছি যেটাতে অক্ষরের সংখ্যা 16 এরও বেশি হওয়ায় পার্থ্যটি স্ক্রোল করছে। এছাড়াও একটি গুরুত্বপূর্ণ বিষয়ও রয়েছে, আপনি এটির গতিটি দেখুন। এখানে আমরা এক সেকেন্ডে দুটি অক্ষর স্ক্রোল করিয়েছি। এখন আমরা এটিও করতে পারি যে এক সেকেন্ডে 3 টি অক্ষর স্ক্রোল হয়ে যাবে। এখন আমরা এটি দেখতে পাই, আমি অন্য কোডটি ফেলবো না যেখানে আপনি দেখবেন হঠাতে এটির গতি পরিবর্তন হবে, আপনি এটি দেখতে পাচ্ছেন?

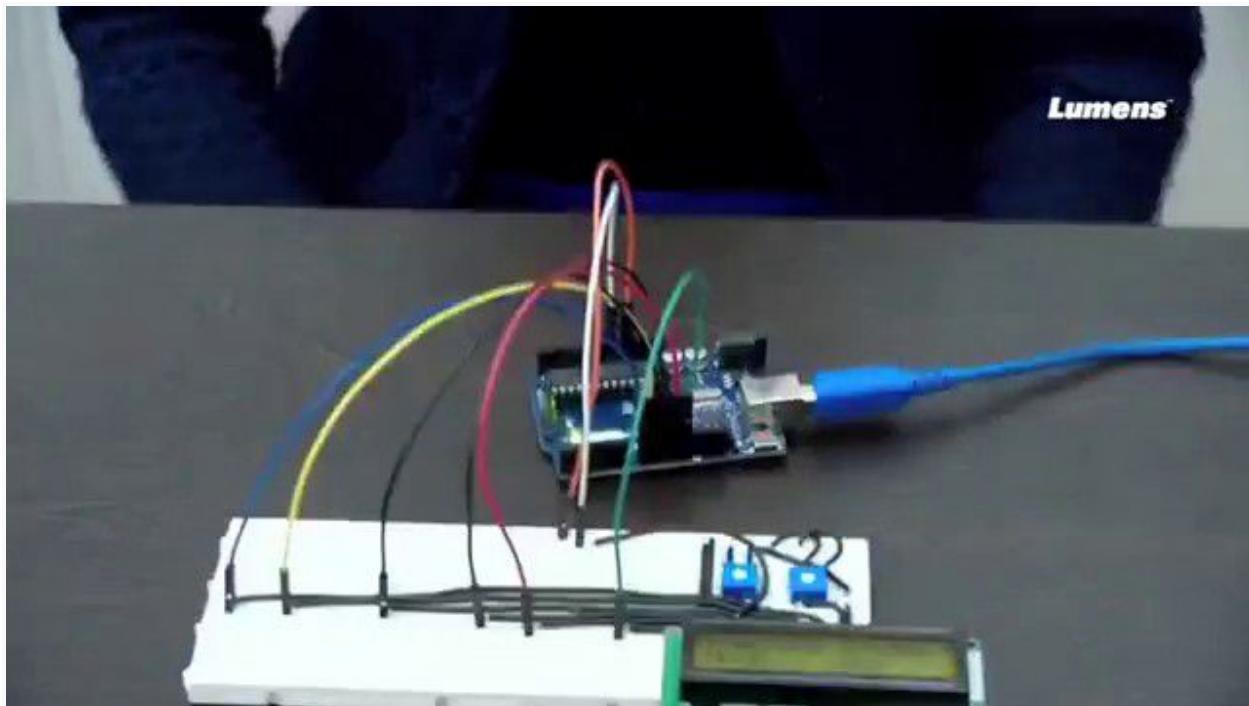
এখন, আপনারা দেখতে পাচ্ছেন যে এটি আগেরটির তুলনায় কিছুটা দ্রুত চলছে। সুতরাং, এইভাবে আপনারা এই নির্দিষ্ট লেলসিডিতে বিভিন্ন রকম পরিবর্তন করতে পারেন, আমি আপনাদের কে এখন Arduino বোর্ডের সাথে সংযোগটিও প্রদর্শন করাতে চাই।

(Refer Slide Time: 27:49)



সুতোঁ, এটি Arduino বোর্ড যা আমি ইতিমধ্যে বিশদ আলোচনা করেছি। সুতোঁ, এখন, আমি একই ধরনের সংযোগ করব।

Refer Slide Time: 28:01)



এখন আমি আপনাদের কে দেখাবো এটা কিভাবে Arduino কোডের সঙ্গে কাজ করে, ইতিমধ্যে আমরা দেখেছি Arduino কোড কিভাবে লেখা হয়। এখন আপনি কোডটি দেখুন যা আমরা ইতিমধ্যে ফেলেছিলাম এবং এখন এটি "embedded system NPTEL 2018" প্রদর্শন করছে।

এখন আমরা বিভিন্ন ধরণের সেক্ষরকে কীভাবে ইন্টারফেস করব তা নিয়ে আমরা এগিয়ে যাব। একবার আমরা যখন সেক্ষরগুলিকে ইন্টারফেস করি মূলত আমরা যা করছি তা হ'ল আমরা পরিবেশ থেকে কিছু প্যারামিটার পেয়ে যাব এবং আমাদের কে সেই প্যারামিটারগুলির ডিজিটাল মান প্রদর্শন করা দরকার। সুতরাং, আপনি আপনার প্রদর্শনের জন্য হয় 7 সেগমেন্ট(segment) ব্যবহার করতে পারেন বা আপনি এটি এলসিডি(LCD) ব্যবহার করে করতে পারেন।

আমরা যে সমস্ত পরীক্ষা-নিরীক্ষা করব সেগুলির জন্য আমাদের কিছু প্রদর্শনের প্রয়োজন হবে; আমরা এই দুটি আডিটপুট ডিভাইস ব্যবহার করব।

ধন্যবাদ

Embedded System Design with Arm
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya
Lecture – 24
Serial Port Terminal Application (Coolterm)

বক্তৃতা 24 এ আপনাদের কে স্বাগত। বক্তৃতায় সেন্সর(snsor) ওলি নিয়ে কাজ করতে যাবার আগে আমাদের কে অবশ্যই জানতে হবে আমার সেন্সরটি আসলে কী মান উৎপন্ন করছে। এই বিষয়ে, আমরা কুলটার্ম(CoolTerm) নামে একটি সিরিয়াল পোর্ট টার্মিনাল অ্যাপ্লিকেশন (serial port terminal application) নিয়ে আলোচনা করব। এছাড়াও অন্যান্য প্রয়োগ রয়েছে, তবে আমরা এই বিশেষ বক্তৃতায় কুলটার্ম(CoolTerm) বিবেচনা করেছি।

(Refer Slide Time: 01:01)

CONCEPTS COVERED

- About CoolTerm**
- Configuration**
- Demonstration**

At the bottom of the slide, there are three logos: the Indian Space Research Organization (ISRO) logo, the Swayam logo (which includes the text 'FREE ONLINE EDUCATION' and 'SWAYAM'), and the Ministry of Human Resource Development (MHRD) logo.

এই বক্তৃতায় আমি কুলটার্ম(CoolTerm) এবং অবশ্যই, এর প্রদর্শন সম্পর্কে কথা বলব।

Introduction

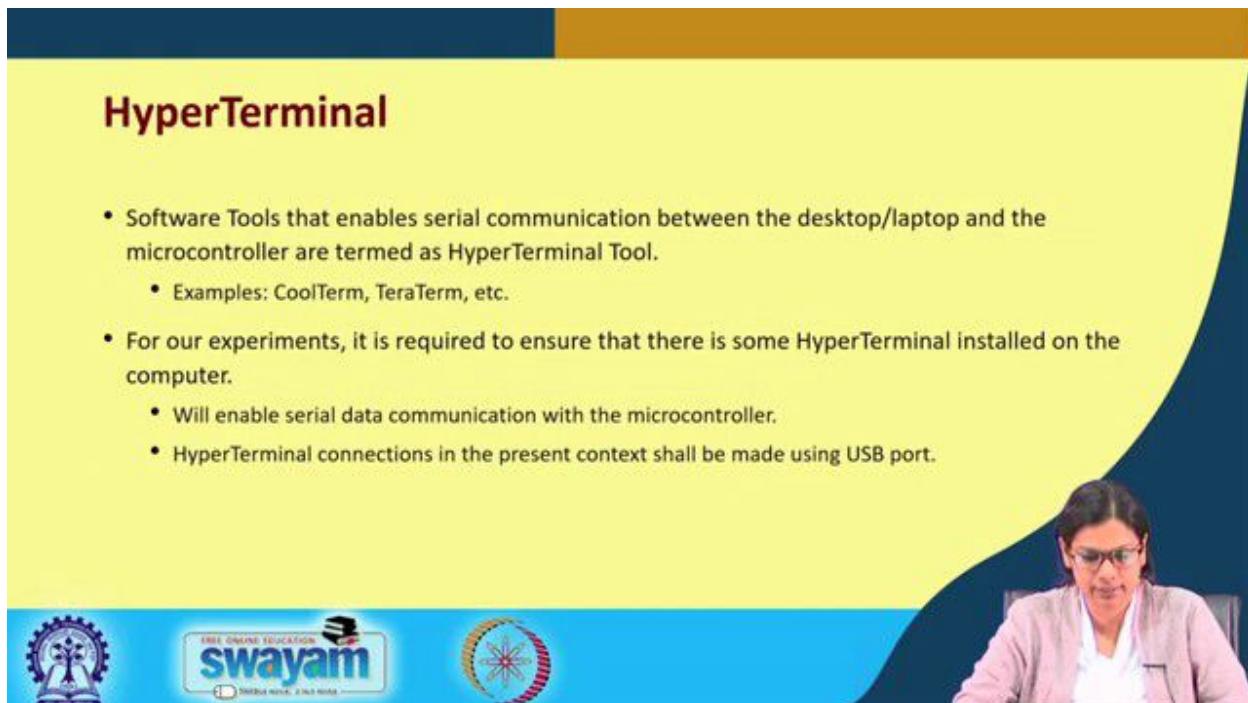
- Sensor data depends on the connection as well as on noise so we need to check the value of sensor.
- The value must be printed on the screen.
- To print the value on the PC screen we need a communication channel.
- A serial communication link between the PC and microcontroller is used to transfer data from microcontroller to PC.
- Software tools such as *CoolTerm*, *TeraTerm*, etc. are used to establish the serial communication and to print the data on the PC screen.



যখন আমরা আরডুইনো(arduino) বোর্ড ব্যবহার করি তখন একটি সিরিয়াল মনিটর(serial monitor) থাকে। আমাদের কেবল সেই সংযোগটি সরবরাহ করা প্রয়োজন; সিরিয়াল সংযোগে আমাদের সেই বন্তি তৈরি করতে হবে এবং সেই বন্তির মাধ্যমে আমরা এটি সরাসরি মুদ্রণ করতে পারি, তবে এসটিএম(STM) বোর্ডে এটি উপলব্ধ নেই। সুতরাং, সেই উদ্দেশ্যে আপনাকে কিছু হাইপার টার্মিনাল(hyper terminal)এমানটি মুদ্রণ করতে হবে। এই সেন্সরটির ডেটা(data) সংযোগের পাশাপাশি নয়েজের(noise) উপরও নির্ভর করে তাই আমাদের সেন্সরের মান পরীক্ষা করতে হবে। সুতরাং, সঠিক সময়ে প্রতিটি পয়েন্টে, সেন্সরটির কাছ থেকে আমরা কী মান পাই তা আমাদের যাচাই করা উচিত। মানটি পর্দায় মুদ্রিত হয়। মানটি কম্পিউটারের পর্দায় মুদ্রণ করতে আমাদের একটি সংযোগকারী মাধ্যমের প্রয়োজন। যোগাযোগ চ্যানেলটি অবশ্যই STM বোর্ড থেকে PC তে তৈরি করতে হবে। PC এবং মাইক্রোকন্ট্রোলারের মধ্যে একটি সিরিয়াল যোগাযোগের লিঙ্ক(link), মাইক্রোকন্ট্রোলার(microcontroller) থেকে পিসি(PC)তে ডেটা(data) স্থানান্তর করতে ব্যবহৃত হয়, যেটা আমরা করতে যাচ্ছি। আমরা ইউএসবি কেবল ব্যবহার করে মাইক্রোকন্ট্রোলার সংযোগ করব এবং তারপরে আমরা পিসি(PC)র সাথে একটি সংযোগ স্থাপন করব এবং তারপরে পিসি স্ক্রিন(PC screen)এ আমরা মানটি মুদ্রণ করব।

কুলটার্ম(CoolTerm) বা টেরাটার্ম(Teraterm) এর মতো স্ক্রিপ্টওয়্যার সরঞ্জামটি সিরিয়াল যোগাযোগ স্থাপনের জন্য এবং স্ক্রিনে ডেটা(data) মুদ্রণের জন্য ব্যবহার করা যেতে পারে। আমাদের এটি দেখতে দিন।

(Refer Slide Time: 02:53)



HyperTerminal

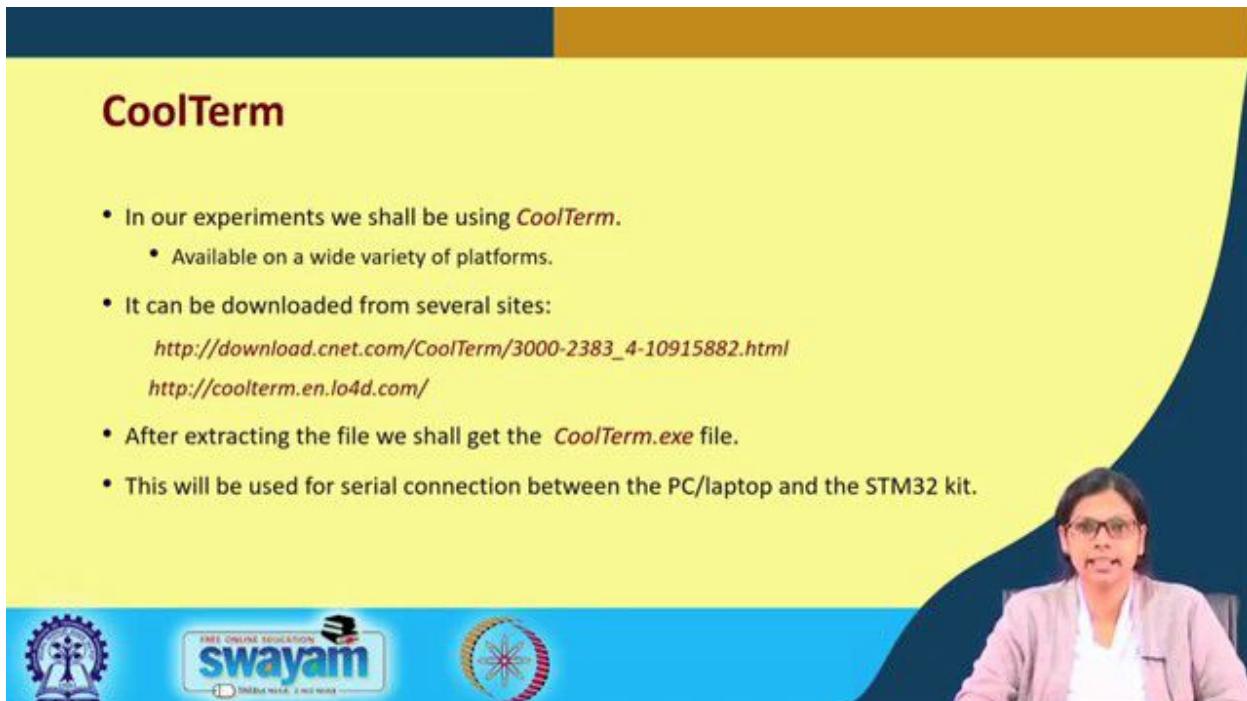
- Software Tools that enables serial communication between the desktop/laptop and the microcontroller are termed as HyperTerminal Tool.
 - Examples: CoolTerm, TeraTerm, etc.
- For our experiments, it is required to ensure that there is some HyperTerminal installed on the computer.
 - Will enable serial data communication with the microcontroller.
 - HyperTerminal connections in the present context shall be made using USB port.

হাইপার টার্মিনাল(hyper terminal) কি? এটি এমন একটি স্ক্রিপ্টওয়্যার(software) সরঞ্জাম যা ডেস্কটপ(desktop) বা ল্যাপটপ(laptop) এবং মাইক্রোকন্ট্রোলার(microcontroller) এর মধ্যে সিরিয়াল(serial) যোগাযোগ সম্ভব করে। এগুলির কয়েকটি উদাহরণ, একটি হ'ল কুলটার্ম, একটি হ'ল টেরাটার্ম, আমরা কুলটার্মের সাথে কাজ করব। সুতরাং আমাদের পরীক্ষার জন্য, এটি কম্পিউটারে একটি হাইপার টার্মিনাল ইনস্টল করা আছে তা নিশ্চিত করা প্রয়োজন। এটি মাইক্রোকন্ট্রোলার(microcontroller) এর সাথে সিরিয়াল ডেটা(serial data) যোগাযোগকে সম্ভব করবে এবং বর্তমান প্রসঙ্গে এই হাইপার টার্মিনাল(hyper terminal) সংযোগটি এই ইউএসবি(USB) পোর্টটি ব্যবহার করে তৈরি করা হবে।

আমরা এখানে মূলতঃ যা করছি তা হ'ল আমরা আমাদের পিসি(PC)তে ইউএসবি(USB) ব্যবহার করে প্রথমে মাইক্রোকন্ট্রোলার(microcontroller) ডিভাইস(device)টিকে যুক্ত করব। তারপরে আমরা একটি হাইপার টার্মিনাল(hyper terminal) ব্যবহার করব, যা আমরা পোর্টের সাথে

মাইক্রোকন্ট্রোলার(microcontroller) দ্বারা সংযুক্ত পোর্টের সাথে একই বাড রেট দিয়ে সেট করব। আমাদের পোর্টটি পরীক্ষা করা দরকার কারণ অনেকগুলি পোর্ট থাকতে পারে।

(Refer Slide Time: 04:21)



CoolTerm

- In our experiments we shall be using *CoolTerm*.
- Available on a wide variety of platforms.
- It can be downloaded from several sites:
 - http://download.cnet.com/CoolTerm/3000-2383_4-10915882.html
 - <http://coolterm.en.lo4d.com/>
- After extracting the file we shall get the *CoolTerm.exe* file.
- This will be used for serial connection between the PC/laptop and the STM32 kit.

বিভিন্ন সাইট(site) থেকে CoolTerm ডাউনলোড করা যেতে পারে। আমি এই 2 টি প্রতিনিধি সাইট এখানে দিয়েছি, এগুলি অন্য জায়গা থেকেও পাওয়া যায়। ফাইলটি extract করার পরে আমরা এই CoolTerm.exe ফাইলটি পেয়ে যাব এবং এটি পিসি বা ল্যাপটপ এবং STM32 কিটের মধ্যে মিরিয়াল যোগাযোগের জন্য ব্যবহৃত হবে।

(Refer Slide Time: 05:03)

Settings on CoolTerm

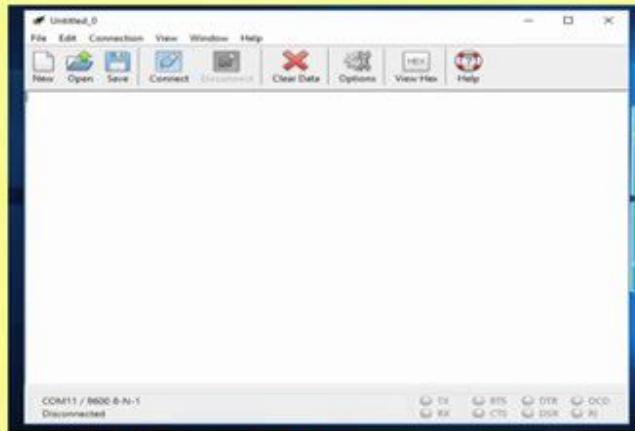
- For proper data communication, proper settings have to be made on the HyperTerminal software.
 - We shall explain the steps for CoolTerm.
- The default serial configuration in CoolTerm is 9600 bauds, 8-bit data.
- If you have not set any bauds rate in the program, then no need to change on this. Else it can be set to match with the program.



এখন, CoolTerm সেট করার জন্য হাইপার টার্মিনাল(hyper terminal) সফ্টওয়্যার(software)টিতে সঠিক সেটিংস(setting) তৈরি করতে হবে। ডিফল্ট(default) সিরিয়াল(serial) কনফিগারেশন(configuration) টি হল 9600 বাড রেট(baud rate) 8 বিট(8 bits)। আপনি যদি এটি পরিবর্তন করতে চান তবে এটি করতে পারেন। এই কুলটার্ম(CoolTerm) ব্যবহার করার আগে এই সেটিংটি অবশ্যই করা উচিত।

(Refer Slide Time: 05:45)

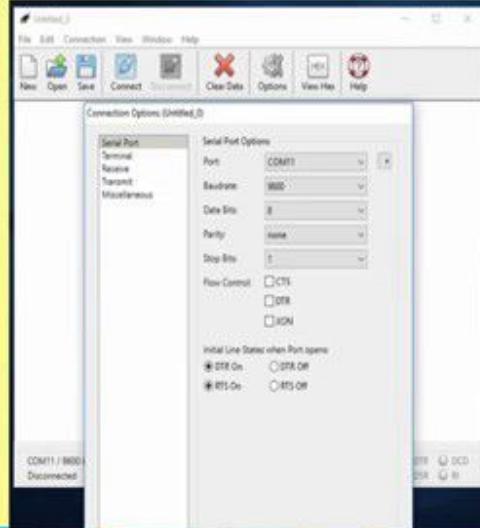
Connect



এখন একবার আপনি কুলটার্ম(CoolTerm) খুললে আপনি এর মতো একটি পর্দা পাবেন এবং তারপরে আপনাকে এই বিকল্পটিতে যেতে হবে।

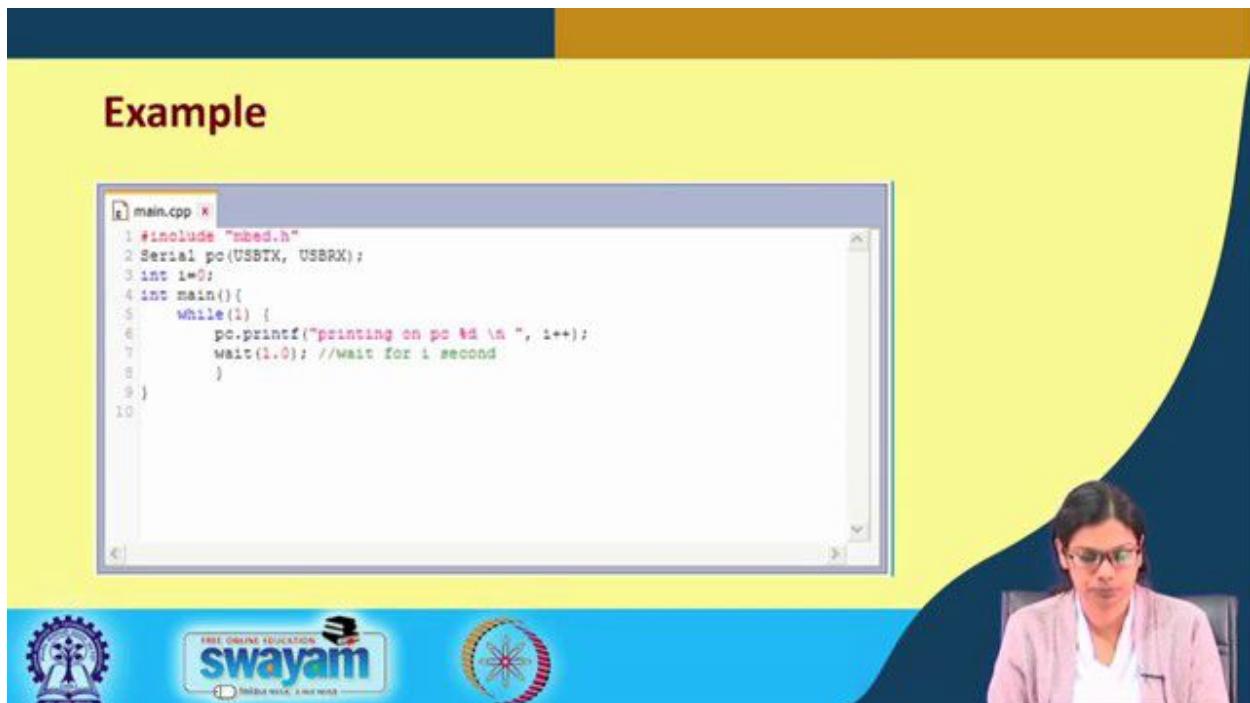
(Refer Slide Time: 05:57)

Configuration



আপনি একবার এই বিকল্পে যান আপনি এখানে এই পোর্ট com11 পাবেন। সুতরাং, এই উদাহরণে এটি 11 নাম্বার, তবে এটি আপনার মেশিনে কোন পোর্টের সাথে সংযুক্ত রয়েছে তার উপর নির্ভর করে। সুতরাং, দয়া করে নিশ্চিত হয়ে নিন যে আপনি এটি পরীক্ষা করেছেন এবং সেই অনুসারে সেই নির্দিষ্ট কম পোর্টের সাথে সংযোগ স্থাপন করেছেন এবং এটি 9600 এবং কনফিগারেশনের জন্য টিপুন। কুলটার্ম(CoolTerm) ব্যবহার করার আগে এই কনফিগারেশনটি নিশ্চিত করে নিন।

(Refer Slide Time: 06:27)



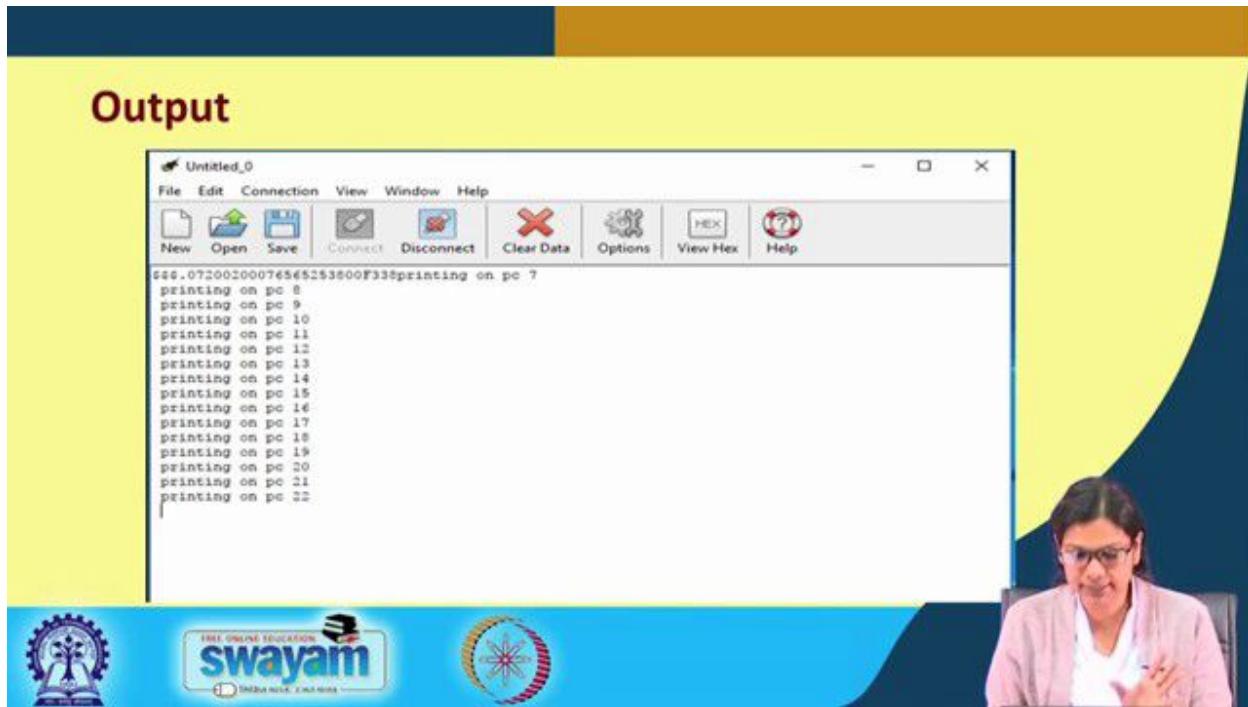
The image shows a computer monitor with a yellow background. On the screen, there is a terminal window titled 'main.cpp' containing the following C++ code:

```
1 #include "mbed.h"
2 Serial pc(USBTX, USBRX);
3 int i=0;
4 int main(){
5     while(1) {
6         pc.printf("printing on pc %d \n ", i++);
7         wait(1.0); //Wait for 1 second
8     }
9 }
```

Below the terminal window, there is a blue banner with the 'SWAYAM' logo and the text 'FREE ONLINE EDUCATION'. To the right of the banner, a video feed of a woman with glasses and a pink jacket is visible, likely the instructor.

এখন, এটি একটি উদাহরণ যা আমরা মুদ্রণ করব। আমরা এই সপ্তাহে প্রবর্তী বক্তৃতা গুলিতে এমন একটি সেন্টেরের সাথে এটি সংযুক্ত করেও আপনাদের কে দেখাব।

(Refer Slide Time: 06:49)



আপনি একবার একটি কোড লিখে এবং সিরিয়াল যোগাযোগ তৈরি করার পরে, সিরিয়াল যোগাযোগ এর এই বিষয়টিকে সিরিয়াল PC USBTX এবং USBRX ব্যবহার করে তৈরি করা হবে এবং এই ক্ষেত্রে, আমরা কেবল মাইক্রোকন্ট্রোলার থেকে কিছু মান মুদ্রণ করছি। তবে সাধারণ ক্ষেত্রে আমরা কী করব? আমরা সেন্সর এর মান মুদ্রণ করাব। এই ক্ষেত্রে, আউটপুটটি কেবল বৃদ্ধি পাচ্ছে।

এখন আমি আপনাকে কনফিগারেশন(configuration)টি দেখাব। আমি ইতিমধ্যে কুলটার্ম ইনস্টল করেছি এবং এটি এভাবেই চলে। মূলত আপনি এটি সংযোগ বিচ্ছিন্ন করতে পারেন এবং তারপরে আপনি আবার এটি সংযোগ করতে পারেন। সুতরাং, মাত্র এক সেকেন্ড, আমি প্রথমে সংযোগটি করব।

(Refer Slide Time: 08:35)

```

main.cpp  main.cpp  main.cpp

#include "mbed.h"

Serial pc(USBTX, USBRX);

AnalogIn analog_value(A0);

int main()
{
    float max_v;
    while(1) {
        max_v = analog_value.read();
        pc.printf("Printing the value of LOB =%f",max_v);
        wait(0.1);
    }
    return(1);
}

```

Compile output for program: STM32_read_analog

Description	Error Number	Resource	In Folder	Location
Statement is unreachable 'return();'	0	main.cpp		Line: 20, Col: 5
Success!	0			1.10.22.0

আমি কেবল এটি সংকলন করব এবং তারপরে আমি আপনাদের কে ইতিমধ্যে জানিয়েছি, আপনারা কীভাবে কোডটি ফেলবেন। আমি ডাউনলোড ফোল্ডারে (download folder) এ যাচ্ছি, যেখানে কোড উপলব্ধ রয়েছে, যা আমি এটি এতে ফেলব।

(Refer Slide Time: 09:29)

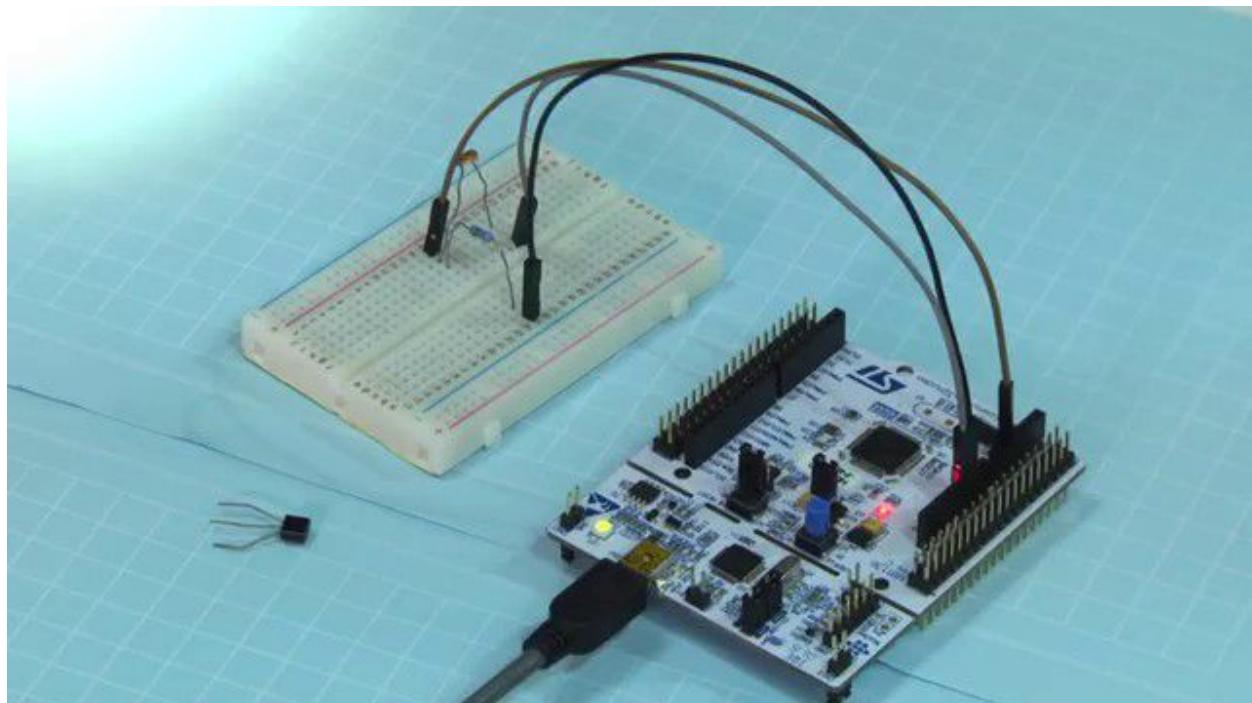
```

Printing the value of LOB =0.994999
Printing the value of LOB =0.954097
Printing the value of LOB =0.913475
Printing the value of LOB =0.874331
Printing the value of LOB =0.855434
Printing the value of LOB =0.815559
Printing the value of LOB =0.045177
Printing the value of LOB =0.941270
Printing the value of LOB =0.912943
Printing the value of LOB =0.852747
Printing the value of LOB =0.823487
Printing the value of LOB =0.837407
Printing the value of LOB =0.852747
Printing the value of LOB =0.814559
Printing the value of LOB =0.957143

```

আমি সংযোগ বিচ্ছিন্ন করব এবং তারপরে আবার সংযোগ করব। এখানেই আপনি দেখতে পাবেন যে আমরা LDR টির মান মুদ্রণ করেছি, আমি আপনাকে যে সংযোগটি করেছি তা আপনাকে দেখাব। এটি LDR থেকে কিছু মান মুদ্রণ করছে; এটি অ্যানালগ(analog) পিন থেকে কিছু মান পাঞ্চে এবং ডিজিটাল রূপান্তরণের সাথে অ্যানালগ(analog) এর পরে এটি ডিজিটাল মান দেখাচ্ছে। এখন, আমি আপনাকে এখানে সংযোগ চিত্রটি দেখাব।

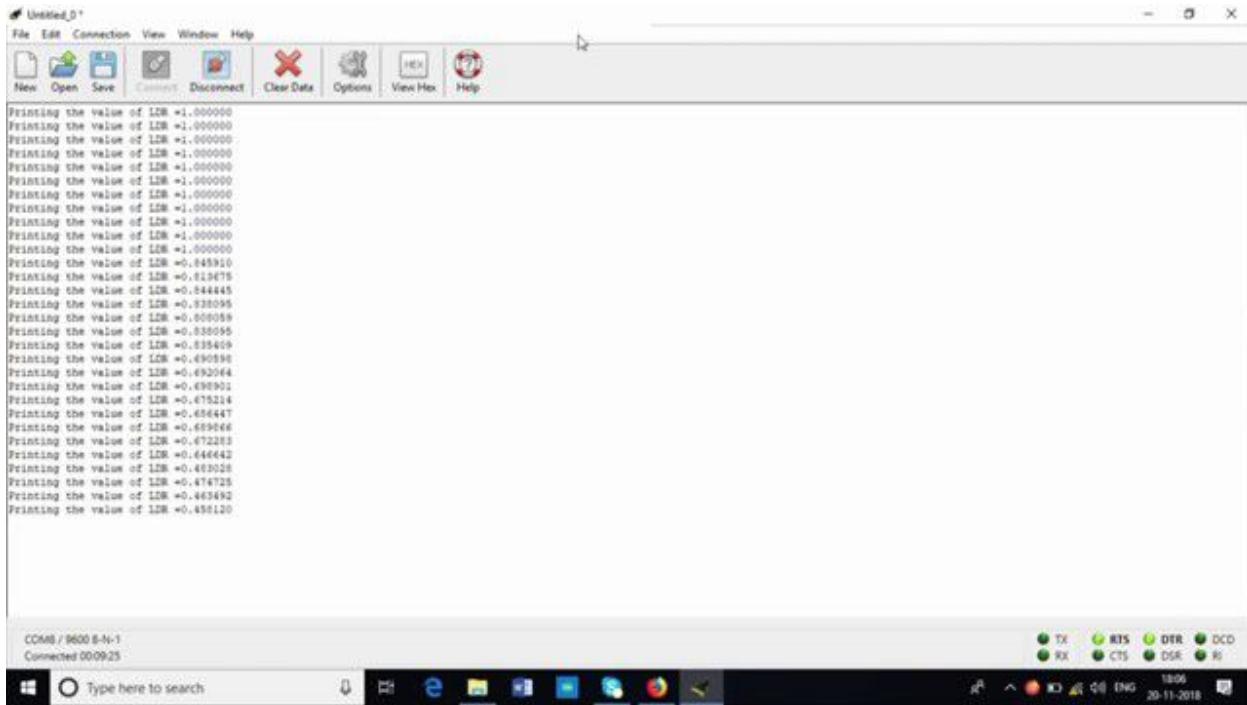
(Refer Slide Time: 10:09)



এটি LDR; আমি সংযোগ তৈরি করেছি। পরে যখন আমি LDR সম্পর্কে বিস্তারিত আলোচনা করেছি। LDR এর একটি বিন্দু Vcc এর সাথে সংযুক্ত থাকে, একটি প্রতিরোধের মাধ্যমে অন্য একটি বিন্দু ground এর সাথে সংযুক্ত থাকে। এটি আমরা যা করেছি এবং আমরা দেখেছি যে আমরা কী মান পাঞ্চ।

আপনি দেখতে পাচ্ছেন যে এদিক থেকে আমরা এই A1 পিনে অ্যানালগ(analog) ইনপুট পেয়েছি। সুতরাং, আমরা এখন থেকে A1 পিনে এনালগ ইনপুট পেয়েছি, আমি মূলতঃ যা করব, এখন আমি এই LDR টিতে এই উজ্জ্বল আলোটি ধীরে ধীরে রাখব। আমি আলোটি সরিয়ে নেব এবং আমি আপনাকে পর্দাটি প্রদর্শন করব, যেখানে এখন মানটি পরিবর্তন হয়।

(Refer Slide Time: 11:23)



আমি আপনাদের কে পর্দাটি দেখাবো, আমি ডেটা সাফ করে দিয়েছি। এখন দেখুন, এটি 1 মুদ্রণ করছে। ডিজিটাল আউটপুটের সর্বাধিক মান 0 থেকে 1। সুতরাং, এটি এখন 1 টি মুদ্রণ করছে ধীরে ধীরে, আমি আলোটি সরিয়ে নেব। এখন দেখুন মূল্য কি মুদ্রণ হয়? এখন এটি 0.8 মুদ্রণ মুদ্রণ করছে। সুতরাং, 1 থেকে মান 0.8 এ হ্রাস পেয়েছে। এখন আমি সেখানে আমার হাতটি একবার রাখব, আপনারা দেখুন এটি কী মান মুদ্রিত হচ্ছে। এখন এটি 0.6 মুদ্রণ হচ্ছে। সুতরাং, আস্তে আস্তে মান হ্রাস পাচ্ছে, এখন আমি আরও কিছুটা হাত দিচ্ছি। এখন, এটি 0.4 এ আসছে এবং অবশেষে, আমি এখানে আমার হাত রেখেছি এবং এখন আপনারা দেখতে পাচ্ছেন যে এটি 0.2 এ নামিয়ে আনা হয়েছে।

সুতরাং, আমরা মাইক্রোকন্ট্রোলারের সাথে সংযোগ স্থাপন করতে এবং এই PC তে মাইক্রোকন্ট্রোলার থেকে প্রাপ্ত কিছু মান প্রদর্শন করতে আমরা কুলটার্ম(CoolTerm) ব্যবহার করেছি। তার আগে আমি আপনাকে কেবল এটির জন্য কোডটি দেখাবো, আপনি ইতিমধ্যে দেখেছেন যে, কোডটি আপনাকে কীভাবে ডাউনলোড(download) করতে হবে, কীভাবে আপনাকে STM বোর্ডে কোডটি রাখতে হবে। এখন এখানে, প্রথমে আপনি দেখুন, প্রথম জিনিস যেটা আমাদের করতে হবে সিরিয়াল PC USBTX এবং USBRX সংযোগটি এবং এখানে আমরা একটি এনালগ মান পড়ছি। আমরা পরবর্তী সপ্তাহগুলিতে দেখতে পাব যে আমরা বিভিন্ন সেন্সর সহ অনেকগুলি অ্যানালগ(analog) মান ব্যবহার করব বা পড়ব। সেন্সরের আউটপুটগুলি দেখতে আমাদের কুলটার্মের(Coolterm) প্রয়োজন হতে পারে।

ଧନ୍ୟବାଦ।

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institutes of Technology, Meghalaya
Lecture – 25
Experiment with Temperature Sensor

25 তম বক্তৃতায় স্বাগতম। এই সপ্তাহে আমরা STM বোর্ডের সাথে বিভিন্ন সেন্সর(sensor)কে ইন্টারফেস(interface) করব। এই বিশেষ বক্তৃতায় আমরা LM35 তাপমাত্রা সংবেদককে ইন্টারফেস করব; LM এর অর্থ হলো লিনিয়ার মনোলিথিক(Linear Monolithic)। আমরা এই LM35 তাপমাত্রা সংবেদকের(sensor) বৈশিষ্ট্যগুলি খতিয়ে দেখি এবং তারপরে আমরা এসট্রিম(STM) বোর্ডের মাধ্যমে এই বিশেষ ডিভাইস(device)টিকে ইন্টারফেস করতে ব্যবহৃত কোড(code) অনুসারে সার্কিট(circuit) ডায়াগ্রাম(diagram)টি দেখব এবং তারপরে আপনাকে এই পরীক্ষাটি প্রদর্শন করব।

(Refer Slide Time: 01:23)

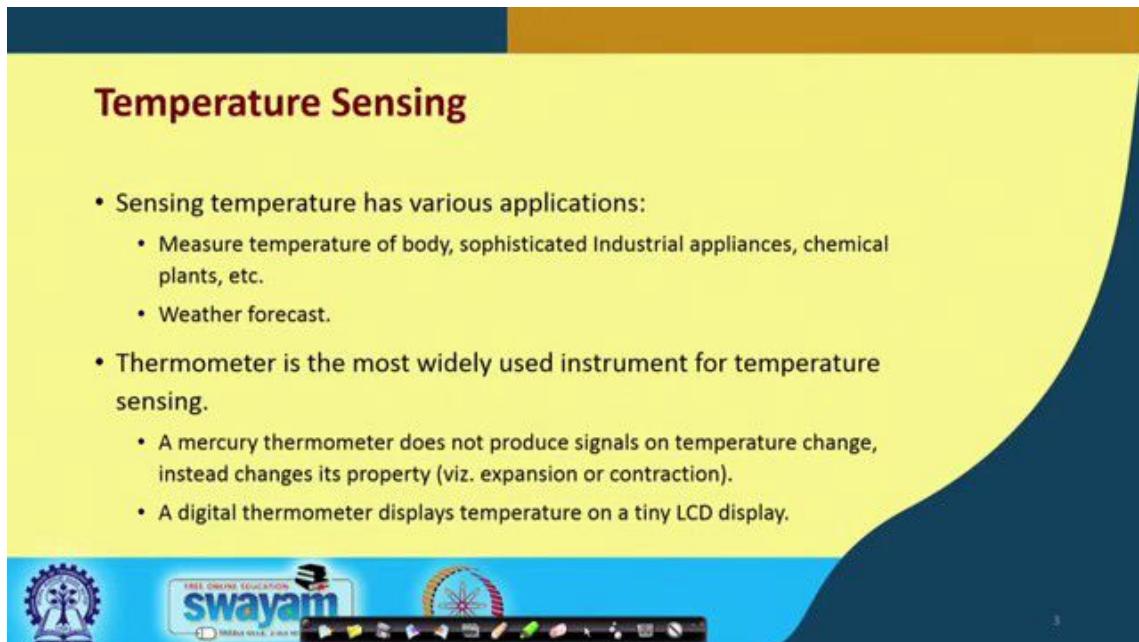
CONCEPTS COVERED

- ❑ Interfacing LM35 to STM32 board
- ❑ Calibration and program code
- ❑ Demonstration

FREE ONLINE EDUCATION **swayam** Empowering Minds, Enriching Lives

আমরা LM35 কে এসটিএম(STM) 32 বোর্ডের সাথে ইন্টারফেস করব এবং কীভাবে ক্রমাঙ্কন করব তাও আপনাকে দেখায়, কারণ যে কোনও সেন্সর(sensor)কে কাজ করার জন্য কোনও ধরণের ক্রমাঙ্কন প্রয়োজন, যেমন প্রোগ্রাম কোড (program code)) এবং প্রদর্শন।

(Refer Slide Time: 01:49)



Temperature Sensing

- Sensing temperature has various applications:
 - Measure temperature of body, sophisticated Industrial appliances, chemical plants, etc.
 - Weather forecast.
- Thermometer is the most widely used instrument for temperature sensing.
 - A mercury thermometer does not produce signals on temperature change, instead changes its property (viz. expansion or contraction).
 - A digital thermometer displays temperature on a tiny LCD display.

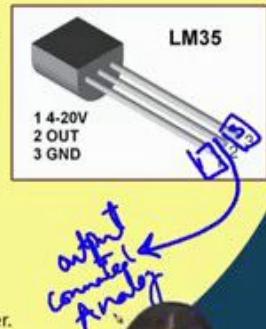
সুতরাং, তাপমাত্রা সংবেদনশীল কি? তাপমাত্রা একটি পারিপার্শ্বিক স্থিতিমাপ; সেন্সিং(sensing) তাপমাত্রার বিভিন্ন প্রয়োগ আছে। এটি কোনও শরীরে তাপমাত্রা পরিমাপ করতে পারে, এটি পরিশীলিত শিল্প সরঞ্জামগুলিতেও ব্যবহৃত হয়, এটি রাসায়নিক কারখানাতে ব্যবহৃত হয়, যেখানে তাপমাত্রা পর্যবেক্ষণ করার প্রয়োজন হয় এবং সেই তাপমাত্রার উপর নির্ভর করে নির্দিষ্ট ডিভাইস(device)গুলি চালু বা বন্ধ করার প্রয়োজন হতে পারে, আবহাওয়ার পূর্বাভাসে ইত্যাদি। আপনি যদি ভাবেন যে আমরা কীভাবে তাপমাত্রা পরিমাপ করি তবে তাপমাত্রা মাপার জন্য থার্মোমিটারই সর্বাধিক ব্যবহৃত উপকরণ।

পারদ থার্মোমিটার তাপমাত্রা পরিবর্তনে সিগন্যাল(signal) তৈরি করে না, পরিবর্তে এটি তার ধর্ম পরিবর্তন করে যেমন এটি প্রস্থারিত বা সংকুচিত করতে পারে - এইভাবে পারদ থার্মোমিটার কাজ করে। অন্যদিকে, একটি ডিজিটাল থার্মোমিটার একটি ছোট এলসিডি(LCD)তে তাপমাত্রা প্রদর্শন করে; এলসিডি কী তা আমরা দেখেছি। ডিজিটাল থার্মোমিটারে, তাপমাত্রা একটি ক্ষুদ্র এলসিডি(LCD)তে প্রদর্শিত হয়।

(Refer Slide Time: 03:29)

LM35 Temperature Sensor

- Precision integrated-circuit temperature device with an output voltage linearly proportional to the temperature in degrees Celsius (or Fahrenheit).
- Linear characteristic: $+10 \text{ mV}/{}^\circ \text{C}$ change in output.
- Rated for full -55°C to 150°C range.
- How to use the LM35?
 - Connect power supply (say, $+5\text{V}$ and GND) to pins 1 and 3.
 - Measure the analog voltage output on pin2.
 - We can connect pin 2 to one of the analog input pins of the microcontroller.



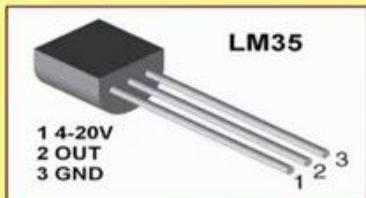
LM35 হ'ল একটি লিনিয়ার মনোলিথিক(Linear Monolithic) তাপমাত্রা সংবেদক, এটি জাতীয় সেমিকন্ডাক্টর(National Semiconductor) দ্বারা বিকাশিত একটি ইন্টিগ্রেটেড সার্কিট(Integrated Circuit)। এটি একটি সঠিক ইন্টিগ্রেটেড সার্কিট তাপমাত্রা ডিভাইস যার সাথে আউটপুট ভোল্টেজ লাইন ডিগ্রি সেলসিয়াস বা ফারেনহাইটের তাপমাত্রার সাথে সমানুপাতিক। এর অর্থ হ'ল আউটপুট(output) ভোল্টেজ ডিগ্রি সেলসিয়াস তাপমাত্রার সাথে রেখিক সমানুপাতিক। আপনি যদি এই রেখিক বৈশিষ্ট্যগুলি মনে করেন তবে এটি এমন হয় যে তাপমাত্রায় প্রতি ডিগ্রি সেলসিয়াস বৃদ্ধির জন্য 10 মিলিভোল্ট পরিবর্তন হবে। LM35 এর পূর্ণ স্কেল পরিসীমা হল -55 থেকে $+150$ ডিগ্রি সেন্টিগ্রেড।

আমরা এই LM35 কীভাবে ব্যবহার করব? LM35 এ একটি সমতল প্রান্ত থাকে এবং তারপরে একটি অর্ধ বৃত্তাকার প্রান্ত থাকে, এই 1 নম্বর পিনটি ভোল্টেজ পিন, এইটি Vcc এর সাথে সংযুক্ত হওয়া উচিত, 3 পিন নম্বর অবশ্যই গ্রাউন্ড(ground) এর সাথে সংযুক্ত থাকতে হবে। আউটপুট ভোল্টেজ 2 নম্বর পিন থেকে পাওয়া যাবে। পিন 1 Vcc এর সাথে সংযুক্ত হওয়া উচিত, পিন 3 অবশ্যই গ্রাউন্ড(ground) এর সাথে সংযুক্ত থাকতে হবে এবং এই মাঝের বিন্দু থেকে আমরা অ্যানালগ ইনপুট(analog input) পোর্টের সাথে সংযোগ করব।

(Refer Slide Time: 07:17)

Reading Analog Value from LM35

- Connect the two extreme terminals to power (say, 5V) and GND.
 - The middle terminal will produce a voltage proportional to the temperature.



```
#include "mbed.h"
AnalogIn sensor(A1);
...
float p; ✓
p = sensor.read();
...
pc.printf ("\n Value read: %f", p);
```

A0, A2, A3, A4
0 - 10V



আমরা কীভাবে অ্যানালগ(analog) মানটি পড়ব? অ্যানালগ(analog) মানটি পড়ার জন্য প্রথমে আমাদের LM35 কে গ্রাউন্ড এবং VCC এর সাথে সংযুক্ত করতে হবে, এবং মধ্যবর্তী টার্মিনালটি তাপমাত্রার সাথে আনুপাতিক ভোল্টেজ তৈরি করবে। এমবেড(mbed) C কোড(code) ব্যবহার করতে আপনাকে এই হেডার(Header) ফাইল(file)টি অন্তর্ভুক্ত করতে হবে তারপরে আমাদের অ্যানালগ(analog) পিনটি একটি ইনপুট অ্যানালগ(input analog) পিন হিসাবে শুরু করতে হবে। বোর্ডের A1 পিনটিকে এখানে অ্যানালগ(analog) পিন হিসাবে বিবেচনা করা হয়, আমরা একটি ফ্লোট(float) ভেরিয়েবল(variable) p নিই কারণ এটি 0 থেকে 1 অবধি মান পাবে। sensor.read পোর্ট A1 এর মাধ্যমে মানটি পড়বে এবং আপনি pc.printf ব্যবহার করে মুদ্রণ করতে পারেন। সুতরাং, এগুলি কোডের কয়েকটি লাইন যা অ্যানালগ পোর্ট(analog port) A1 থেকে সেন্সর মানটি পড়তে প্রয়োজন হবে। পাশাপাশি A0, A2, A3, A4 এবং A5 এর মতো অন্যান্য অ্যানালগ পোর্টগুলি রয়েছে।

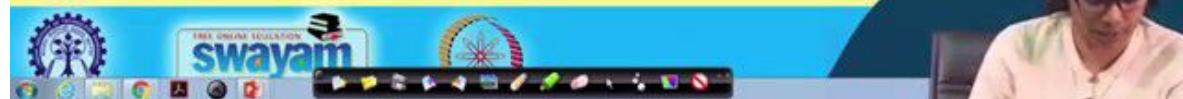
(Refer Slide Time: 09:37)

Calibration

- LM35 output voltage increases linearly with increase in temperature.
 - 10 mV increase for every $^{\circ}\text{C}$ rise in temperature.
 - For $k^{\circ}\text{C}$, the output voltage will be $10k \text{ mV}$.
 - If maximum temperature is 50°C , maximum output voltage will be 0.5 V ($10 \times 50 = 500$).
- How to compute the temperature in $^{\circ}\text{C}$?
 - The read() function in AnalogIn class returns a fraction between 0 and 1.
 - Apply a voltage 0.5V directly to the analog input pin, and measure the value printed by the program (suppose, the value printed is P_{max}).
 - For an unknown temperature, if the value printed is P , the temperature value can be calculated as

$$T = 50 / P_{\text{max}} * P$$

$$50^{\circ}\text{C} \quad 1^{\circ}\text{C}$$



এখন, আমরা কীভাবে ক্রমাঙ্কন করব? ধরুন ঘরের বর্তমান তাপমাত্রা x । বর্তমান তাপমাত্রার জন্য, আপনি কী মান পাচ্ছেন? যদি আমরা জানি যে এখন তাপমাত্রাটি হল, ধরুন 20 ডিগ্রি সেন্টিগ্রেড, আমি আমার অ্যানালগ আউটপুট(analog output)টিতে নির্দিষ্ট মান পাচ্ছি, তবে আমি যদি x মান পায়, তবে তাপমাত্রাটি কী হবে? ক্রমাঙ্কন করার এটি একটি উপায়। আসুন আমরা দেখি যে আমরা এই নির্দিষ্ট কোডটিতে কী করেছি। আমি ইতিমধ্যে উল্লেখ করেছি LM35 আউটপুট ভোল্টেজ(output voltage) তাপমাত্রার সাথে রৈখিকভাবে বৃদ্ধি পায়। তাপমাত্রায় প্রতি ডিগ্রি সেন্টিগ্রেড বৃদ্ধির জন্য 10 mV বৃদ্ধি রয়েছে। সুতরাং, k ডিগ্রি সেন্টিগ্রেড বৃদ্ধির জন্য, আউটপুট ভোল্টেজ হবে $10k \text{ mV}$ । যদি তাপমাত্রা সর্বোচ্চ হয় তবে ধরুন তাপমাত্রা 50 ডিগ্রি সেন্টিগ্রেড এবং সর্বাধিক আউটপুট ভোল্টেজটি 0.5 ভোল্ট। সুতরাং, $10 \times 50 \text{ mV} = 500 \text{ mV}$ । এখন আমরা কীভাবে ডিগ্রি সেন্টিগ্রেড তাপমাত্রা গণনা করব? আসুন এটি দেখতে দিন, AnalogIn ক্লাসে read ফাংশনটি 0 এবং 1 এর মধ্যে একটি ভগ্নাংশ প্রদান করে। আমি ইতিমধ্যে জানিয়েছি যে আমরা অ্যানালগ ইনপুট(analog input) পিনের সাথে সরাসরি একটি ভোল্টেজ 0.5 ভোল্ট প্রয়োগ করতে পারি এবং তারপরে আমরা প্রোগ্রাম(program) দ্বারা মুদ্রিত মানটি পরিমাপ করি। মনে করুন, যে মানটি print করা হয়েছে তা হ'ল P_{max} . এর অর্থ, 0.5 ভোল্টের জন্য আমরা যে মানটি পাচ্ছি তা হল P_{max} । তারপরে একটি অজানা তাপমাত্রার জন্য, যদি মুদ্রিত মান P হয় তবে তাপমাত্রার মান $50 / P_{\text{max}} * P$ হিসাবে গণনা করা যায়।

(Refer Slide Time: 12:55)

Calibration

- LM35 output voltage increases linearly with increase in temperature.
 - 10 mV increase for every $^{\circ}\text{C}$ rise in temperature.
 - For $k^{\circ}\text{C}$, the output voltage will be $10k\text{ mV}$.
 - If maximum temperature is 50°C , maximum output voltage will be 0.5 V ($10 \times 50 = 500$).
- How to compute the temperature in $^{\circ}\text{C}$?
 - The `read()` function in *AnalogIn* class returns a fraction between 0 and 1.
 - Apply a voltage 0.5V directly to the analog input pin, and measure the value printed by the program (suppose, the value printed is P_{max}).
 - For an unknown temperature, if the value printed is P , the temperature value can be calculated as

$$T = 50 / P_{max} * P$$

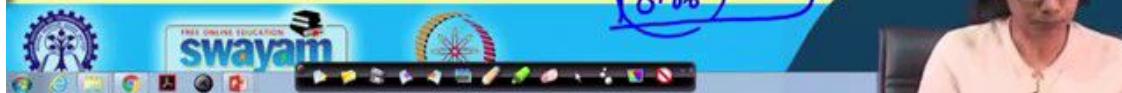
$$\begin{aligned} P_{max} &= 50 \\ 1 &= \frac{50}{P_{max}} \\ P &= \frac{P_{max} \times P}{50} \end{aligned}$$



(Refer Slide Time: 13:39)

- Alternate way of computing:
 - Suppose the room temperature is 20°C
 - Using Coolterm, we run the program and suppose we find the value printed as 0.06
 - For any unknown temperature, if the value printed is P , the temperature in $^{\circ}\text{C}$ will be $20P / 0.06$

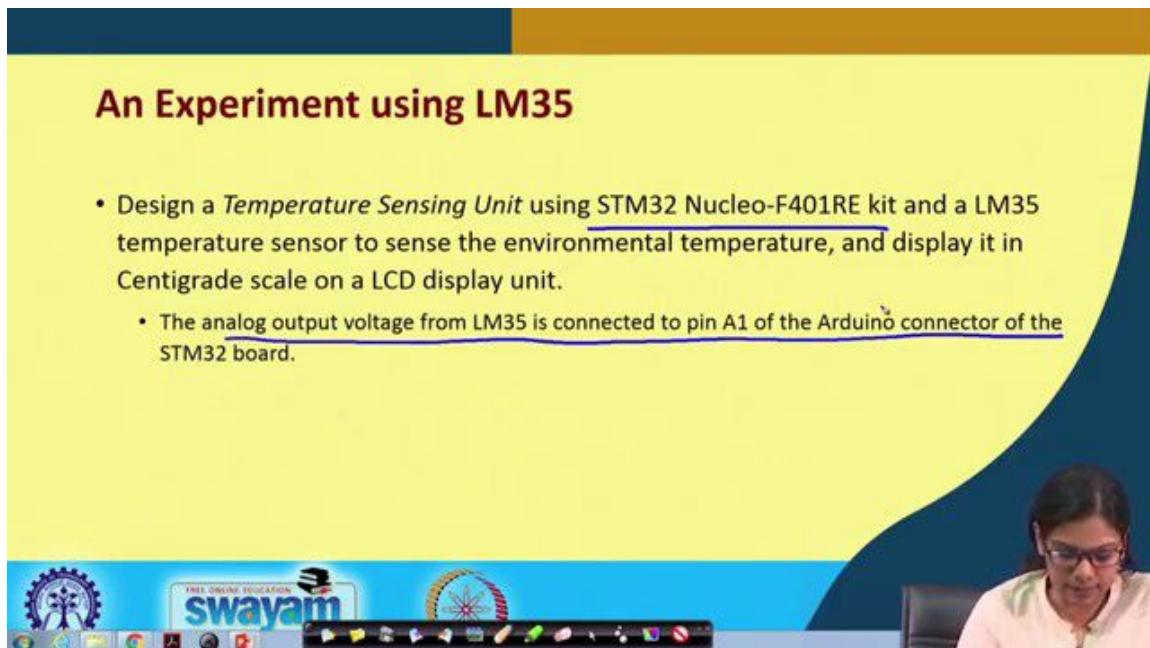
$$\begin{aligned} 0.06 &= 20^{\circ}\text{C} \\ 1 &= \frac{20^{\circ}\text{C}}{0.06} \\ P &= \frac{20^{\circ}\text{C} \times P}{0.06} \end{aligned}$$



আমরা কম্পিউটিংয়ের(computing) যে বিকল্প উপায় ব্যবহার করেছি সেটি আমাকে দেখাতে দিন।
মনে করুন, ঘরের তাপমাত্রা 20 ডিগ্রি সেন্টিগ্রেড। আমরা দেখব যে ভোল্টেজ আউটপুট থেকে আমরা কী মান পাচ্ছি। মনে করুন, ঘরের তাপমাত্রা 20 ডিগ্রি সেন্টিগ্রেড, আমরা ইতিমধ্যে কুলটার্ম(CoolTerm) সম্পর্কে আলোচনা করেছি; মনে করুন আমরা মানটি 0.06 হিসাবে পাই।

সুতরাং মূলতঃ, যদি এটি 0.06 হয় তবে তাপমাত্রা 20 ডিগ্রি সেন্টিগ্রেড। যদি আমরা মানটি P পাই, তবে তাপমাত্রা $20 / 0.06 * P$ হবে। সুতরাং, আমরা আগে ঘরের বর্তমান তাপমাত্রাটি ব্যবহার করে গণনা করেছি, যা আমরা সেই নির্দিষ্ট সময়ে পেয়ে যাচ্ছিলাম।

(Refer Slide Time: 15:15)

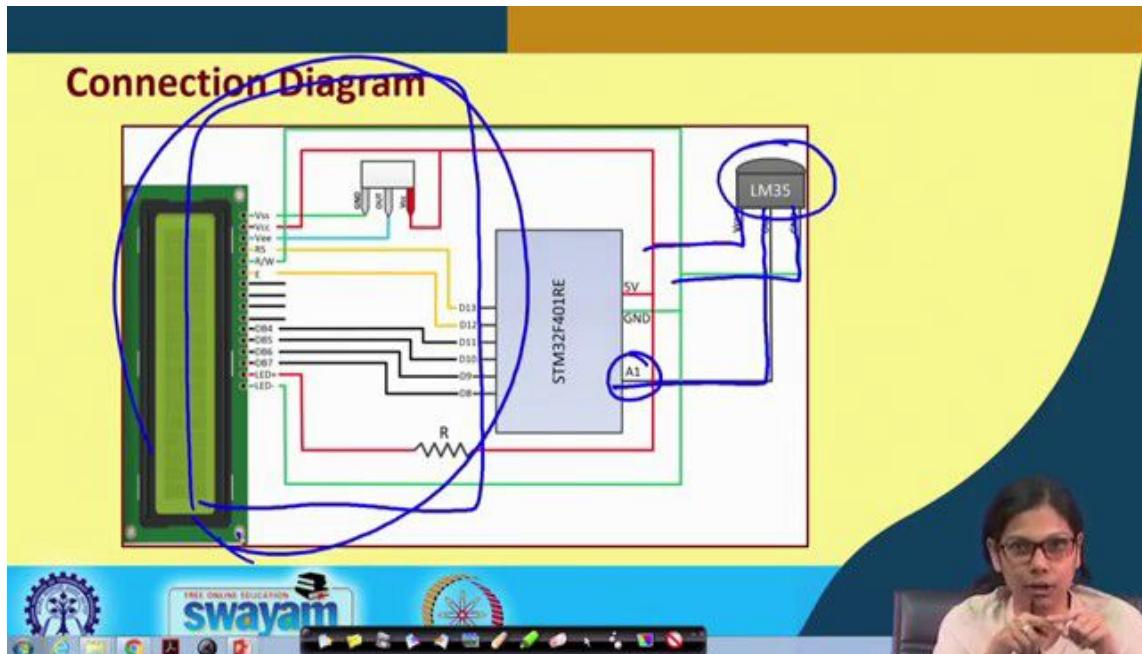


An Experiment using LM35

- Design a *Temperature Sensing Unit* using STM32 Nucleo-F401RE kit and a LM35 temperature sensor to sense the environmental temperature, and display it in Centigrade scale on a LCD display unit.
 - The analog output voltage from LM35 is connected to pin A1 of the Arduino connector of the STM32 board.

আসুন পরীক্ষাটি দেখি। আমি LM35 সম্পর্কে আলোচনা করেছি, কীভাবে আমরা LM35 কে STM এর সাথে সংযুক্ত করব এবং নির্দিষ্ট port থেকে আমরা কীভাবে একটি অ্যানালগ(analog) মান পড়ি এবং কীভাবে আমরা ক্রমাঙ্কণ করি। এখন আমরা পরিবেশগত তাপমাত্রা অনুধাবন করতে এবং এটি একটি এলসিডি ডিসপ্লে ইউনিটে(LCD display unit) সেন্টিগ্রেড স্কেলে প্রদর্শন করতে STM Nucleo-F401RE কিট(kit) এবং LM35 তাপমাত্রা সেন্সর ব্যবহার করে একটি তাপমাত্রা সংবেদক ইউনিট ডিজাইন করব। LM35 থেকে অ্যানালগ আউটপুট ভোল্টেজটি STM বোর্ডের পিন A1 এর সাথে সংযুক্ত। আসুন এখন সার্কিট ডায়াগ্রামটি খতিয়ে দেখি।

(Refer Slide Time: 16:13)



আমৱা মূলত D8 থকে D13 পর্যন্ত এলসিডি ডিসপ্লে(LCD display)টিৱ বিভিন্ন পিনেৱ সাথে সংযোগ কৱছি, আমৱা এখানে যা কৱতে যাচ্ছি তা হ'ল, আমৱা এই LM35 VCC আউটপুটটিকে A1 পিনেৱ সাথে সংযুক্ত কৱছি। এটি সংযোগ এবং সংযোগেৱ এই অংশটি সম্পর্কে সব কিছু যা আমৱা ইতিমধ্যে আলোচনা কৱেছি। সুতৰাং, এটি তাপমাত্ৰা অনুধাৰণ কৱবে, ক্ৰমাঙ্কনটি কৱবে এবং এই এলসিডিতে তাপমাত্ৰা প্ৰদৰ্শন কৱবে। আমৱা এই উদাহৰণটি প্ৰদৰ্শন কৱব, তবে এৱ আগে আমি আপনাকে কোডটি দেখাৰ যা সাৰ্কিট তৈৱিৱ পৱে কাৰ্যকৰ কৱা দৱকাৰ।

(Refer Slide Time: 17:23)

Mbed Program Code

```
#include "mbed.h"
#include "string.h"
#include "TextLCDScroll.h"

TextLCDScroll lcd (D13, D12, D11, D10, D9,
                    D8, TextLCD::LCD16x2);
Serial pc (USBTX,USBRX);
AnalogIn sensor(A1);
```

```
int main(){
    int val=297; // after calibration
    char buf[20];
    float p;
    lcd.setLine(0, "Temp in Degree C");
    while(1) {
        p = sensor.read();
        sprintf(buf, "%4.2f", p*val);
        lcd.setLine(1, buf);
        wait(1);
    }
}
```

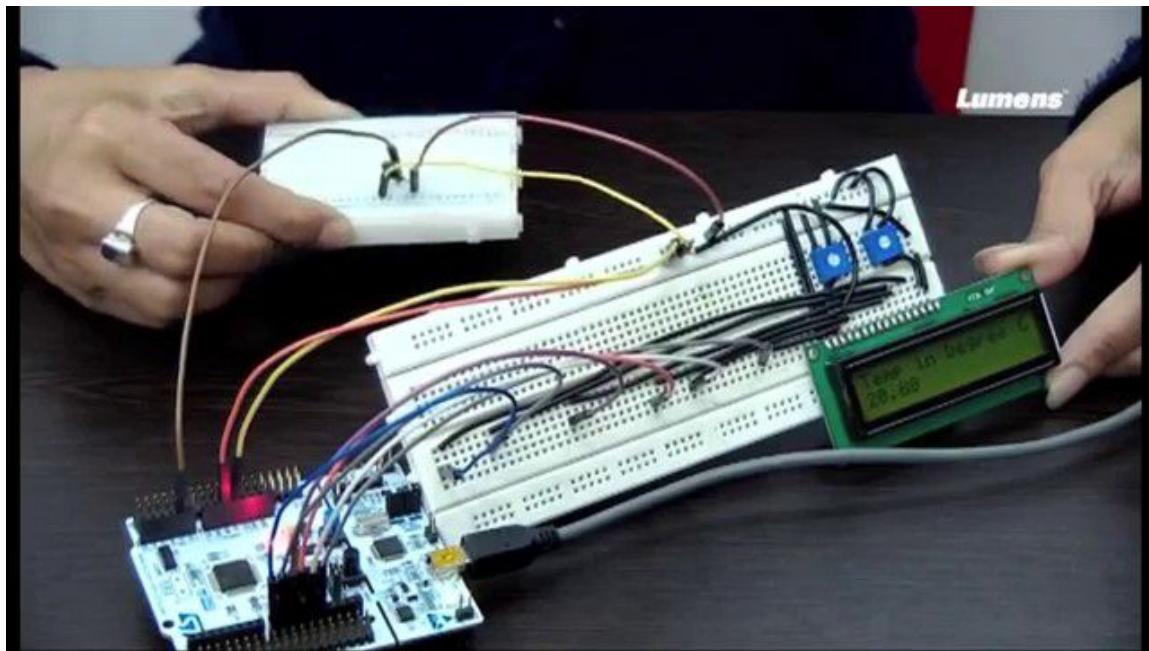


সুতোঃ, এটি এমবেড প্রোগ্রাম কোড(mbed program code), আমরা জানি যে আমাদের TextLCDScroll ব্যবহার করতে হবে। আমরা এখানে সিরিয়াল(serial) USBTX RX ব্যবহার করি, এটি সিরিয়াল(serial) যোগাযোগের জন্য যা আমরা করি যে আমরা এটি হাইপার টার্মিনাল(hyper terminal) এর যে কোনওটিতে প্রদর্শন করতে পারি, যা আমরা ইতিমধ্যে আলোচনা করেছি। এটি সেন্সর(sensor) এর নামে অ্যানালগ ইনপুট মান, আমরা পোর্ট এ 1 এর মাধ্যমে পড়ছি এবং এটি ক্রমাঙ্কের পরে, আমরা এর মতো একটি মান পাই ... মানটি 297।

আমরা এখানে একটি অক্ষর বাফার(buffer) নিই এবং আমরা 0 তম লাইনটি "Temp in Degree C" তে সেট করছি। প্রথমে আমরা while লুপের মধ্যে sensor.read ব্যবহার করে মানটি পড়ছি এবং তারপরে আমরা p * val গণনা করছি। p এর যে মানটি আমরা পেয়েছি সেই মানটি 297 দিয়ে গুণ করছি এবং আমরা এটি এই বাফার(buffer) এ সংরক্ষণ করছি, এবং শেষ পর্যন্ত আমরা এটি পরবর্তী লাইনে প্রদর্শন করছি। আমরা 1 সেকেন্ডের জন্য অপেক্ষা করছি এবং আবার আমরা একই জিনিস বারবার করছি। সুতোঃ এখন, আমি ইতিমধ্যে পুরো কোডটি আলোচনা করেছি যে কীভাবে আমরা এটির ইন্টারফেস(interface) করব এবং আমরা এটি কীভাবে প্রদর্শন করব। এখন আমরা আপনাকে এই নির্দিষ্ট কোডটির প্রদর্শনের জন্য পরীক্ষা-নিরীক্ষা দেখাচ্ছি, যেখানে আমরা এই LM35-টি STM-এর সাথে ইন্টারফেস করব এবং এলসিডি(LCD)তে তাপমাত্রা প্রদর্শন করব। আমি ST মাইক্রো ইলেক্ট্রনিক্স পোর্ট(micro electronics port) এর সাথে LM35 সংযুক্ত করব এবং তারপরে আমি বর্তমান তাপমাত্রা প্রদর্শন করব।

এই ঘরের বর্তমান তাপমাত্রা কত ছিল? আমি আপনাকে বলি এখানে শিলংয়ের তাপমাত্রা বেশ কম; বাইরের তাপমাত্রা বর্তমানে 17 ডিগ্রি। সুতরাং, আসুন দেখুন আমাদের সেন্সিং ইউনিট(sensing unit)এ এখানে কী আসছে।

(Refer Slide Time: 21:01)



সুতরাং, এটি তাপমাত্রা সংবেদক, এই তাপমাত্রা সেন্সরটি LM35 তাপমাত্রা সংবেদক। এই বাম পিনের সমতল প্রান্তটি 5 ভোল্ট VCC এর সাথে সংযুক্ত হওয়া উচিত। এই LM35 এর ডান প্রান্তটি গ্রাউন্ড(ground) এর সাথে সংযুক্ত হবে এবং মধ্যবর্তী অবস্থান থেকে এটি অ্যানালগ পোর্ট(analog port) A1 এর সাথে সংযুক্ত হবে। এই অ্যানালগ পোর্ট(analog port) টি অভ্যন্তরীণভাবে একটি AD কনভার্টারের সাথে সংযুক্ত। কুলটার্ম(CoolTerm) এর মতো যে কোনও সিরিয়াল যোগাযোগ সফটওয়্যার(software) ব্যবহার করে আপনি ডিজিটাল মানটি দেখতে পারেন। সুতরাং এখন, আমি সংযোগটি করছি।

সুতরাং, আমরা ইতিমধ্যে ক্রমাঙ্কনটি সম্পন্ন করেছি এবং আপনি ইতিমধ্যে জানেন যে এই নির্দিষ্ট সেন্সর(sensor) LM35 এর ভোল্টেজের 10 মিলিভোল্ট পরিবর্তন তাপমাত্রার 1 ডিগ্রি পরিবর্তনের সমান হবে। সুতরাং, আমরা সেই অনুযায়ী ক্রমাঙ্কনটি সম্পন্ন করেছি এবং এখন আমি কোডটি ফেলব, যা এই নির্দিষ্ট ঘরের বর্তমান তাপমাত্রা প্রদর্শন করবে। সুতরাং, আমাকে এটি সংযোগ করতে দিন। যে তাপমাত্রাটি এটি দেখাচ্ছে, আপনি এটি দেখতে পারেন, ডিগ্রি সেন্টিগ্রেডের তাপমাত্রা 18.20... 19.40। আমাকে তাপমাত্রাটি কিছুটা বাড়াতে দিন এবং মানটি কী আসে তা দেখুন।

আপনি দেখতে পাবেন যে এটি 21.60 হয়ে গেছে। আপনি যদি এই নির্দিষ্ট সেন্সরটিকে কিছুটা ঘষে থাকেন তবে মান পরিবর্তন হয় বা তাপমাত্রার পরিবর্তন দেখতে আপনি এর সামনে সামান্য কিছুটা ফুঁক(blow) মারতে পারেন। এটি LM35 এর সাথে একটি সহজ পরীক্ষা, যেখানে এটি তাপমাত্রাটি পড়ছে এবং এটি প্রদর্শন করছে। আপনি দেখতে পাবেন যে LM35 এর সাথে সংযোগটি সহজ সরল।

মোদা কথাটি হল আপনার এই ক্রমাঙ্কনটি দরকার, ক্রমাঙ্কন পদক্ষেপটি খুব গুরুত্ব পূর্ণ। এই ক্রমাঙ্কন পদক্ষেপে, আপনাকে সেন্সরটির বর্তমান মানটি পড়তে হবে, এটি কী দিচ্ছে এবং আপনি এটি কুলটার্ম(CoolTerm) এ দেখতে পাবেন এবং তারপরে সেই মানটির পরিপ্রেক্ষিতে এই নির্দিষ্ট ঘরের বর্তমান তাপমাত্রাটি কী। তদনুসারে, আপনি এই নির্দিষ্ট দৃশ্যটি অন্তর্ভুক্ত করতে আপনার কোড পরিবর্তন করতে পাবেন। LM35 এর বিভিন্ন অন্যান্য প্রয়োগ রয়েছে, আমরা আপনাকে একটি জিনিস দেখিয়েছি, যা আপনি অন্তর্ভুক্ত করতে পাবেন এবং এটি অন্য পরীক্ষাগুলির জন্য করতে পাবেন।

ধন্যবাদ

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institutes of Technology, Meghalaya
Lecture – 26
Experiment with LDR Light Sensor (Part I)

বক্তৃতা 26 এ স্বাগতম। এই বক্তৃতায় আমরা এসটিএম(STM) বোর্ডের সাথে এলডিআর(LDR) স্টারফেস(Interface) করব। আমরা প্রথমে এলডিআর(LDR) কী তা খতিয়ে দেখব। এবং এলডিআর(LDR) দিয়ে পরীক্ষাটি সম্পাদন করতে সার্কিট ডায়াগ্রামটি অবশ্যই দেখব। আমরা এলডিআর(LDR) নিয়ে দুটি পরীক্ষা-নিরীক্ষা করব।

(Refer Slide Time: 01:05)

CONCEPTS COVERED

- ❑ LDR interface circuit
- ❑ Experiment for sensing ambient light level and changing intensity of a LED (using PWM)
- ❑ Demonstration

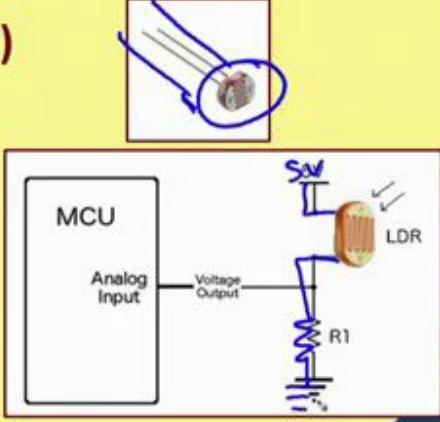
FREE ONLINE EDUCATION
swayam
Swayam is a free online education platform.

আমি যেমন বলেছি, আমরা এলডিআর(LDR) ইন্টারফেসিং সার্কিট(interfacing circuit) নিয়ে আলোচনা করব। এই পরীক্ষায় আমরা পরিবেষ্টনের আলো স্তরকে সংবেদন করব। এবং, আমরা PWM পোর্টের মাধ্যমে একটি এলইডি(LED)এর তীব্রতা পরিবর্তন করব।

(Refer Slide Time: 01:31)

Light Dependent Resistor (LDR)

- A LDR is a variable resistor (a passive component) whose resistance value changes depending upon the amount of light falling on it.
 - More the amount of light, less will be the resistance and vice versa.
 - The variation of resistance with light intensity is non-linear.
- A simple resistance divider can generate an analog voltage that depends on R_{LDR} .



FREE ONLINE EDUCATION
swayam

সুতরাং একটি আলো নির্ভর রোধ কি? এটি একটি পরিবর্তনশীল প্রতিরোধক যা একটি প্যাসিভ(passive) উপাদান, যার প্রতিরোধের মান তার উপর পড়া আলোর পরিমাণের উপর নির্ভর করে পরিবর্তিত হয়। যখনই আলো এর উপর পড়ে তখনই এর প্রতিরোধের মান পরিবর্তন হয়। আলোর পরিমান যত বেশি হবে প্রতিরোধ তত কমবে এবং তিনিমূলীয় হবে। সুতরাং, যদি আরও আলো ডিভাইসে পড়ে, তবে এর প্রতিরোধ ক্ষমতা হ্রাস পায় এবং যদি কম আলো পড়ে ; এর অর্থ, যদি এটি অন্ধকার হয়, তবে প্রতিরোধের মানটি বেশি হবে।

আলোর তীব্রতার সাথে এই প্রতিরোধের বৈচিত্র্য অ-রৈখিক(non-linear); আমরা যেমন LM35 এর ক্ষেত্রে দেখেছি যে 10 mV পরিবর্তনের সাথে সাথে তাপমাত্রায় এক ডিগ্রি সেন্টিগ্রেড বৃদ্ধি পায়। সুতরাং, একটি লিনিয়ার(linear) পরিবর্তন ছিল; এখনে সেটা নেই। আলোর তীব্রতার সাথে প্রতিরোধের প্রকরণ(variation)টি রৈখিক(linear) নয়। একটি সাধারণ প্রতিরোধের বিভাজক একটি অ্যানালগ(analog) ভোল্টেজ(voltage) তৈরি করতে পারে যা এই এলডিআরটির প্রতিরোধের উপর নির্ভর করে।

এটি একটি সাধারণ ডায়াগ্রাম(diagram)। এটি একটি LDR এবং আপনি দেখতে পাচ্ছেন যে একটি port 5V এর সাথে যুক্ত হয় অন্য একটি port প্রতিরোধের মাধ্যমে গ্রাউন্ড(ground) এ সংযুক্ত থাকে। আমরা যেখান

থেকে ভোল্টেজ আউটপুট নিছি তা এখান থেকে। আমরা অ্যানালগ(analog) ইনপুট(input)টির সাথে সংযোগ স্থাপন করছি এভাবে সংযোগটি চলে।

(Refer Slide Time: 03:55)

LDR Interfacing

- First test the LDR by measuring its resistance.
 - Consider for example, $R_{LDR} = 150 \text{ k}\Omega$ (no light), and $R_{LDR} = 10 \text{ k}\Omega$ (light)

$$V = IR$$

$$I = V / (R_{LDR} + R1)$$

$$V_{out} = I * R1$$

$$= V * R1 / (R_{LDR} + R1)$$

$$= 5 * R1 / (R_{LDR} + R1)$$

এখন, আসুন এই LDR ইন্টারফেসিংটি দেখুন। আমাদের প্রথমে LDR প্রতিরোধের পরিমাপ করে পরীক্ষা করা দরকার। এই ঘরে দেখুন একটি পরিবেষ্টিত আলো রয়েছে, অন্য ঘরে আলোর তীব্রতা ভিন্ন হতে পারে।

সুতরাং, আমরা এটি বিবেচনা করি, যখন আলো না থাকে তখন এলডিআর(LDR)টির প্রতিরোধ ক্ষমতা 150 কিলো ওহম(kilo ohm) হয় এবং যখন আলো থাকে এটি 10 কিলো ওহম(kilo ohm) হয়, আসুন আমরা যা পাওয়া তা বলি। তবে আপনাকে মাল্টিমিটার(multimeter) ব্যবহার করে পড়তে হবে। আমাদের বলুন যে আমরা এটি করেছি এবং এই পরিসরে আমরা একটি মান পেয়েছি।

যেমন দেখানো হয়েছে, এলডিআর(LDR) থেকে আউটপুট(ouput) ভোল্টেজ(voltage) হবে $5 * R1 / (R_{LDR} + R1)$ ।

এই সমীকরণের ভিত্তিতে, আমরা ইতিমধ্যে করা পূর্ববর্তী গণনার উপর নির্ভর করে কীভাবে এই নির্দিষ্ট $R1$ কে নেব তা খতিয়ে দেখব।

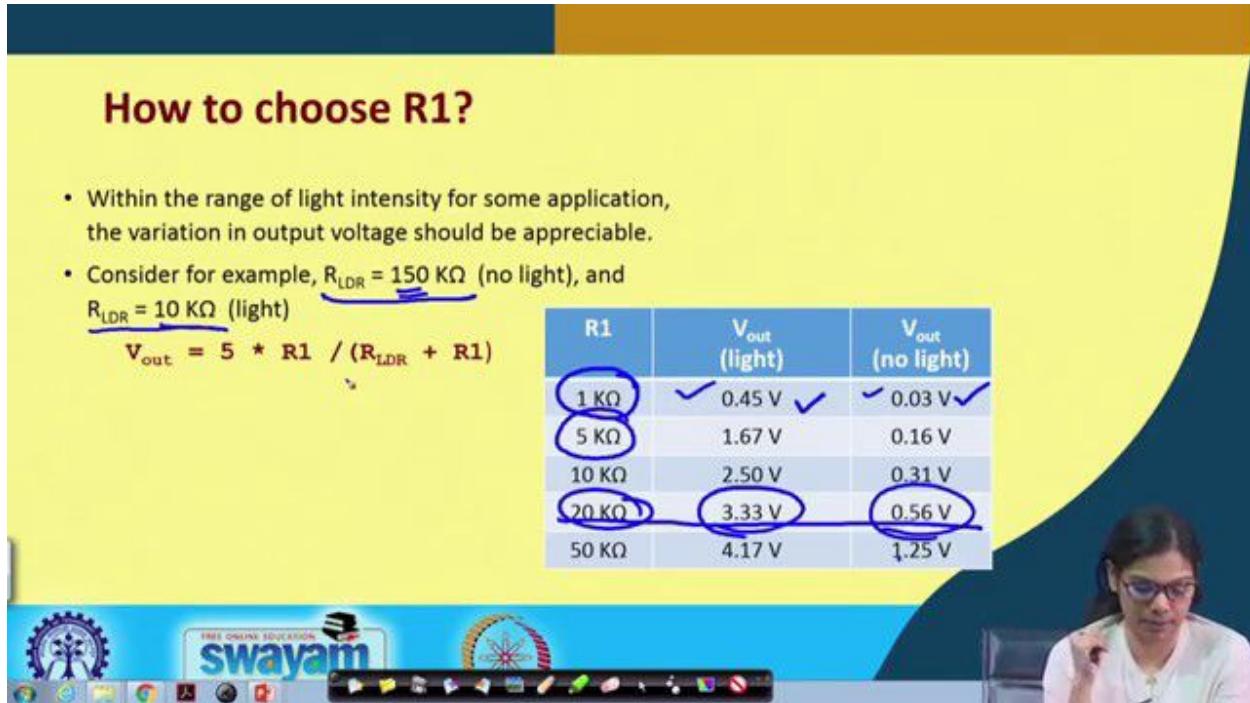
(Refer Slide Time: 06:25)

How to choose R1?

- Within the range of light intensity for some application, the variation in output voltage should be appreciable.
- Consider for example, $R_{LDR} = 150 \text{ k}\Omega$ (no light), and $R_{LDR} = 10 \text{ k}\Omega$ (light)

$$V_{out} = 5 * R1 / (R_{LDR} + R1)$$

R1	V_{out} (light)	V_{out} (no light)
1 k Ω	0.45 V	0.03 V
5 k Ω	1.67 V	0.16 V
10 k Ω	2.50 V	0.31 V
20 k Ω	3.33 V	0.56 V
50 k Ω	4.17 V	1.25 V

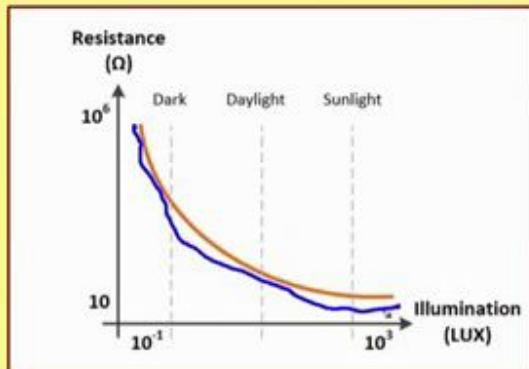


সুতরাং, কিছু প্রয়োগের জন্য আলোর তীব্রতার পরিসীমাটির মধ্যে আউটপুট ভোল্টেজ (output voltage) এর প্রকরণটি প্রশংসনীয় হওয়া উচিত, কারণ আমি চাই যে প্রকরণটি আরও বেশি হোক। যখন আলো থাকে এবং যখন আলো থাকে না তখন আমাদের R1 কে এমন ভাবে বেছে নেওয়া দরকার যে এটি বেশ দৃশ্যমান হয়।

উদাহরণস্বরূপ বিবেচনা করুন যখন আলো থাকে না তখন RLDR 150 কিলো ওহম (kilo ohm) হয় এবং যখন আলো থাকে তখন 10 কিলো ওহম (kilo ohm) হয়। R1 যখন 1 কিলো ওহম (kilo ohm) হয় তখন আমরা আলোর জন্য Vout এবং আলো না থাকার জন্য Vout এর গণনা করার জন্য এটি এই নির্দিষ্ট সূত্রে রেখেছি। মানগুলি প্রদর্শিত হয়। এর বিভিন্ন মানের জন্য অনুকূপ গণনা দেখানো হয়েছে।

(Refer Slide Time: 09:07)

Nonlinear Variation of Resistance with Light



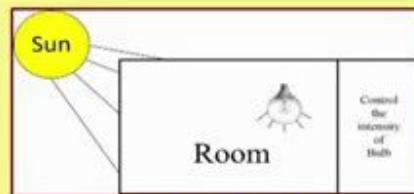
FREE ONLINE EDUCATION
swayam



আপনি আলোর সাথে প্রতিরোধের অ-রৈখিক(non-linear) প্রকরণ দেখতে পাচ্ছেন। যখন এটি অন্ধকার হয় তখন এর প্রতিরোধ ক্ষমতা বেশি থাকে এবং যখন আলো থাকে এর প্রতিরোধ ক্ষমতা হ্রাস পায়।

(Refer Slide Time: 09:41)

The Experiment



- Create a “*Room Light Controller*” using LDR. Sense the ambient light and adjust the intensity of the room light accordingly.
 - For demonstration, we use a LDR circuit connected to the analog input pin A1 of the Arduino interface.
 - Instead of a bulb (in a room), we use a LED that is connected to the PWM digital output pin D3.
 - The duty cycle of the PWM signal is varied depending on the level of light.
 - In the experiment, we define three levels with corresponding duty cycles 1.0 (fully on), 0.8 (somewhat dim), and 0.0 (off).
 - The LED glows when D3 is at 0.

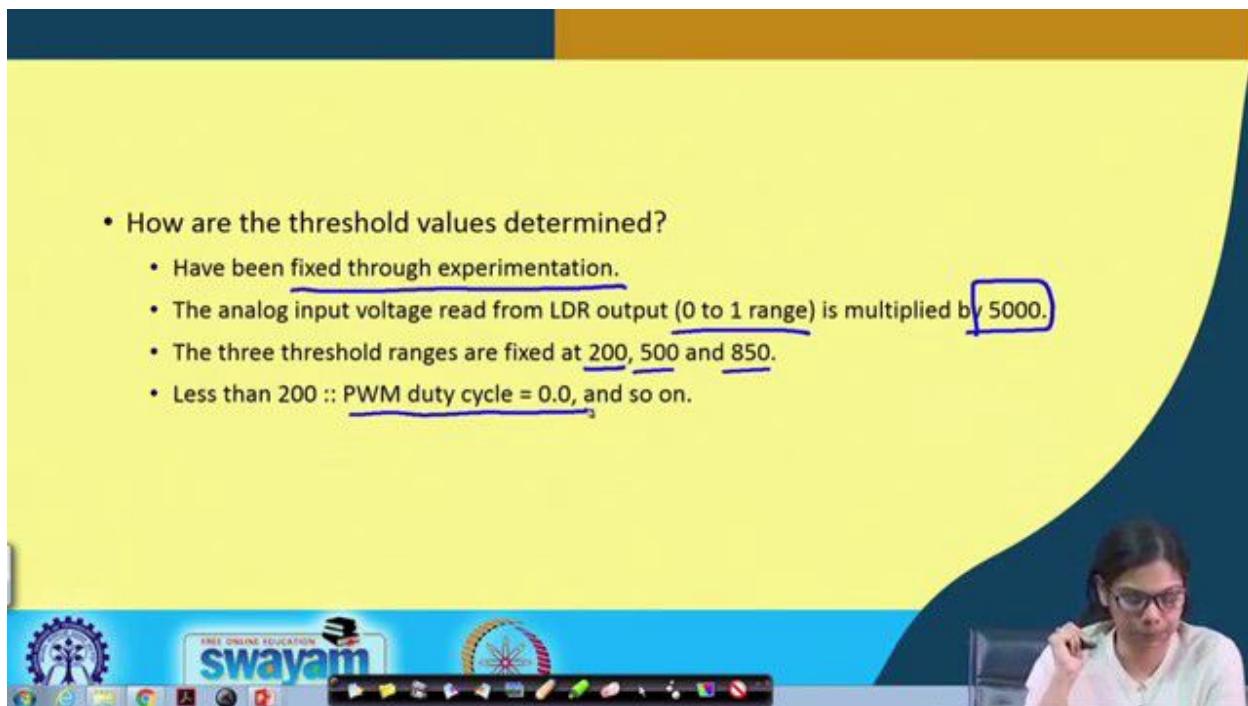


আমরা আজ যে পরীক্ষাটি করবো তা হ'ল, আমরা LDR ব্যবহার করে একটি রুম লাইট কন্ট্রোলার(room light controller) তৈরি করব। আমরা পরিবেষ্টিত আলো অনুধাবন করব এবং সেই অনুযায়ী ঘরের আলোর তীব্রতা সামঞ্জস্য করব। সুতরাং, আমরা এখানে কিভাবে অনুকরণ(simulation) করব? সিমুলেশন(simulation) এর উদ্দেশ্যে আমরা কেবল একটি LDR ব্যবহার করেছি। আমরা একটি বাহ্যিক LDR ব্যবহার করেছি যা এই STM বোর্ডের সাথে সংযোগ স্থাপন করবে এবং পরিবেষ্টিত আলোর উপর নির্ভর করে আলো কিছুটা উজ্জ্বল হয় এলইডি(LED) পুরোপুরি জ্বলবে, যদি আলো খুব উজ্জ্বল হয়, তবে এলইডি(LED) জ্বলবে না এবং যদি সামান্য আলো হয় এলইডি(LED) জ্বলে উঠবে, তবে খুব হালকা এবং এরকমই হবে।

আসুন আমরা মূলতঃ যা করেছি তা প্রদর্শনের জন্য দেখতে দিন। আমরা অ্যানালগ(analog) ইনপুট(input) পিন A1 এর সাথে সংযুক্ত একটি LDR সার্কিট(circuit) ব্যবহার করি। সুতরাং, আমি যেমন বলেছিলাম একটি বাইরের পরিবর্তে আমরা একটি এলইডি(LED) ব্যবহার করছি যা PWM ডিজিটাল আউটপুট(output) পিন D2 এর সাথে সংযুক্ত। আমরা ইতিমধ্যে PWM পিনটি কী, PWM পিনের সাথে সংযুক্ত ফাংশন(function) গুলি কী, আমরা সেই PWM পিনের মাধ্যমে কী করতে পারি তা নির্দিষ্টভাবে আলোচনা করেছি। এই ক্ষেত্রে আমরা যে কোনও পচন্দসই তীব্রতায় এলইডি(LED) জ্বালাতে PWM পিনগুলির মধ্যে একটি ব্যবহার করব। সিগন্যাল(signal) এর শুল্ক চক্র(duty cycle) টি আমাদের আলোর স্তরের উপর নির্ভর করে পরিবর্তিত হয়।

পরীক্ষায় আমরা 1, 0.8 এবং 0 এর যথাযথ চক্রের সাথে তিনটি স্তর নির্ধারণ করি; এই তিনটি জিনিস আমরা ব্যবহার করেছি। D3 পোর্ট লাইন 0 হলে এলইডি(LED) জ্বলে উঠবে।

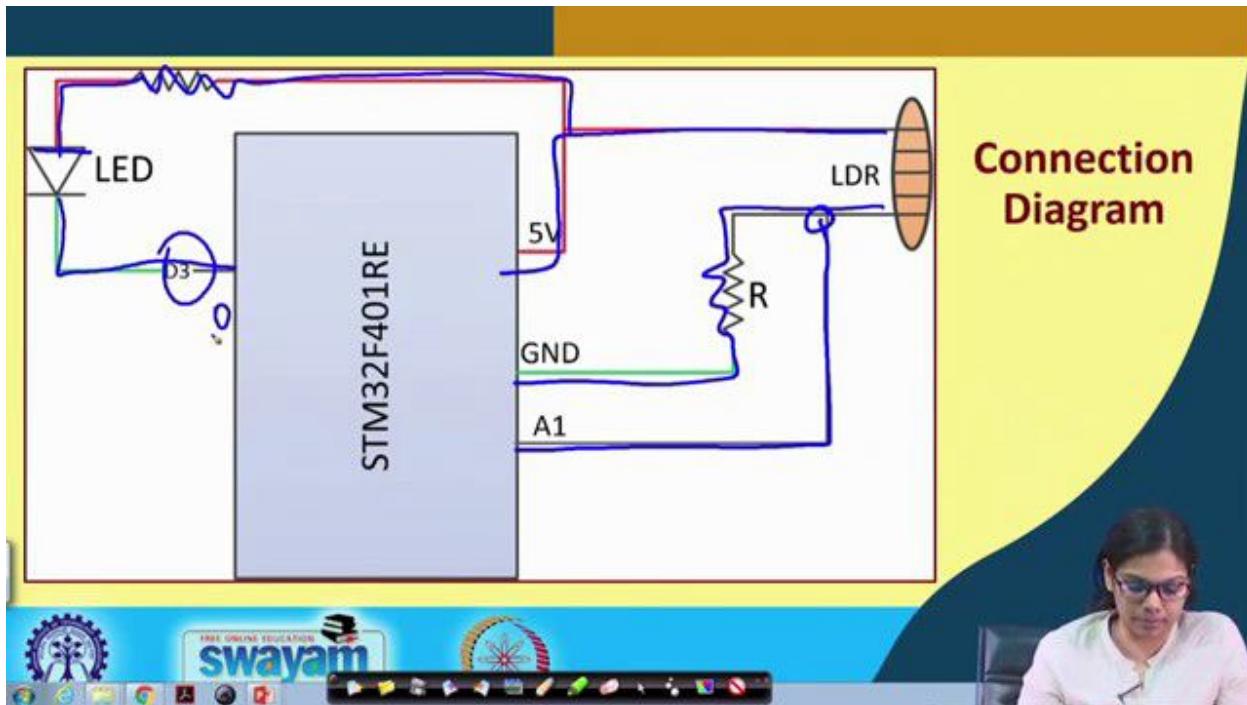
(Refer Slide Time: 12:01)



- How are the threshold values determined?
 - Have been fixed through experimentation.
 - The analog input voltage read from LDR output (0 to 1 range) is multiplied by 5000.
 - The three threshold ranges are fixed at 200, 500 and 850.
 - Less than 200 :: PWM duty cycle = 0.0, and so on.

প্রাণ্তিক মানগুলি(threshold values) কীভাবে নির্ধারণ করা হয়? এগুলি পরীক্ষার মাধ্যমে স্থির করা হয়েছে। আমরা একটি পরীক্ষা করেছি এবং সম্পর্কিত মানগুলি খুঁজে পেয়েছি। LDR আউটপুট(output) থেকে নেওয়া অ্যানালগ(analog) ইনপুট(input) ভোল্টেজ(voltage)টি 0 থেকে 1 এর মধ্যে রয়েছে, যা 5000 দ্বারা গুণিত হয়েছে,, যাতে আমরা কোনটা আলোকিত এবং কোনটা আলোকিত নয় তা খুঁজে পেতে পারি। তিনটি প্রাণ্তিক পরিসীমা(range) 200, 500 এবং 850 এ স্থির করা হয়েছে। এই তিনটি পরিসীমা আমরা রেখেছি; এবং 200 PWM এরও কম হলে শুষ্ক চক্র 0 হবে, এবং যদি এটি 500 এবং 850 হয় তবে আমরা সেই অনুযায়ী পরিবর্তন করব।

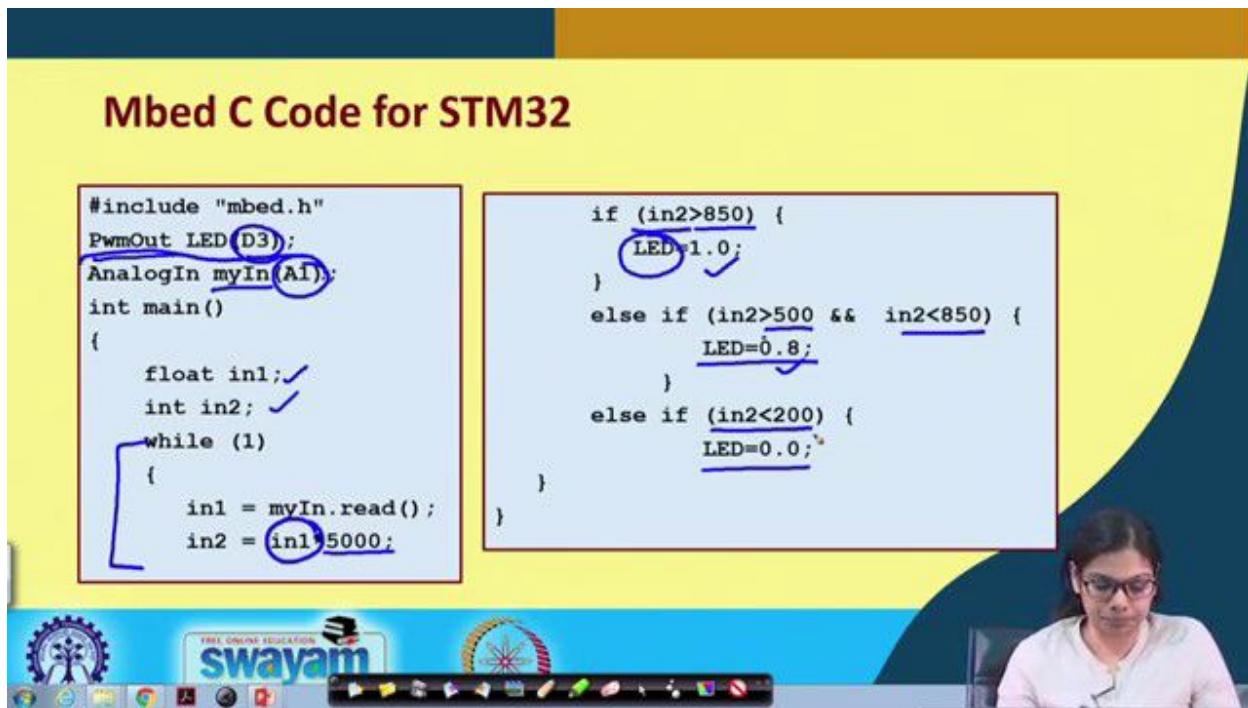
(Refer Slide Time: 13:13)



এটি হ'ল সার্কিট ডায়াগ্রাম(circuit diagram)। এটি একটি একক(single) LDR, একটি প্রান্ত 5 ভোল্টের সাথে সংযুক্ত। একটি প্রতিরোধের মাধ্যমে LDR এর আর একটি প্রান্ত গ্রাউন্ড(ground) এর সাথে সংযুক্ত। এবং আমরা যে অ্যানালগ(analog) মানটি A1 বিল্ড থেকে নিচ্ছি এবং আমি যেমন বলেছিলাম এলইডি(LED) একটি বাল্ব(bulb) কে সিমুলেট(simulate) করে। LED টির জন্য এটি হ'ল আনোড যা প্রতিরোধের মাধ্যমে 5 ভোল্টের সাথে সংযুক্ত থাকে এবং ক্যাথোডটি পোর্ট D3 এর সাথে সংযুক্ত থাকে। এই মানটি যদি 0 হয় তবে LED জ্বলবে।

এখন আমাদের কোড(code) এ আসতে দিন।

(Refer Slide Time: 14:33)



Mbed C Code for STM32

```
#include "mbed.h"
PwmOut LED(D3);
AnalogIn myIn(A1);
int main()
{
    float in1;
    int in2;
    while (1)
    {
        in1 = myIn.read();
        in2 = in1*5000;
    }
}

if (in2>850) {
    LED=1.0;
}
else if (in2>500 && in2<850) {
    LED=0.8;
}
else if (in2<200) {
    LED=0.0;
}
```

কোডটি স্ব-বর্ণনামূলক।

while(1) লুপের মধ্যে আমরা কী করছি? আমরা myIn.read () ফাংশনটি দিয়ে পড়ছি, তারপরে আমি এটি 5000 দিয়ে গুণ করছি। যদি এই মানটি 850 এর বেশি হয়, তবে আমি সেই এলইডি(LED) টি অন(on) করছি; যদি এটি 5000 এরও বেশি এবং 850 এরও কম হয় তবে আমি এটি আলাক্ষি তবে পুরো তীব্রতার সাথে নয় (ডিউটি(duty) চক্র 0.8 সহ); এবং যদি এটি 200 এরও কম হয় তবে আমি এটিকে বন্ধ করব। সুতরাং, এগুলি তিনটি স্তর যা আমরা এই এলইডি(LED) এর জন্য তৈরি করেছি।

আমরা কি আউটপুট(output) করছি? আমরা এই এলইডি(LED) এ আউটপুট(output) করছি যা PWM আউটপুট। তিনটি স্তরের জন্য উজ্জ্বলতা আলাদা হবে।

এখন, আমি আপনাকে এলডিআর(LDR) দিয়ে পরীক্ষাটি দেখাব। এই পরীক্ষায় আমি দেখাব যে, পরিবেষ্টনের আলোর উপর নির্ভর করে আমরা দেখতে পাব যে আমরা কীভাবে একটি এলইডি(LED) এর তীব্রতা পরিবর্তন করতে পারি।

আমরা একটা LDR এবং একটা LED নিয়েছি। এলইডি(LED)টি পরিবেষ্টিত আলোর উপর নির্ভর করে তিনটি স্তরে আলোকিত হবে।

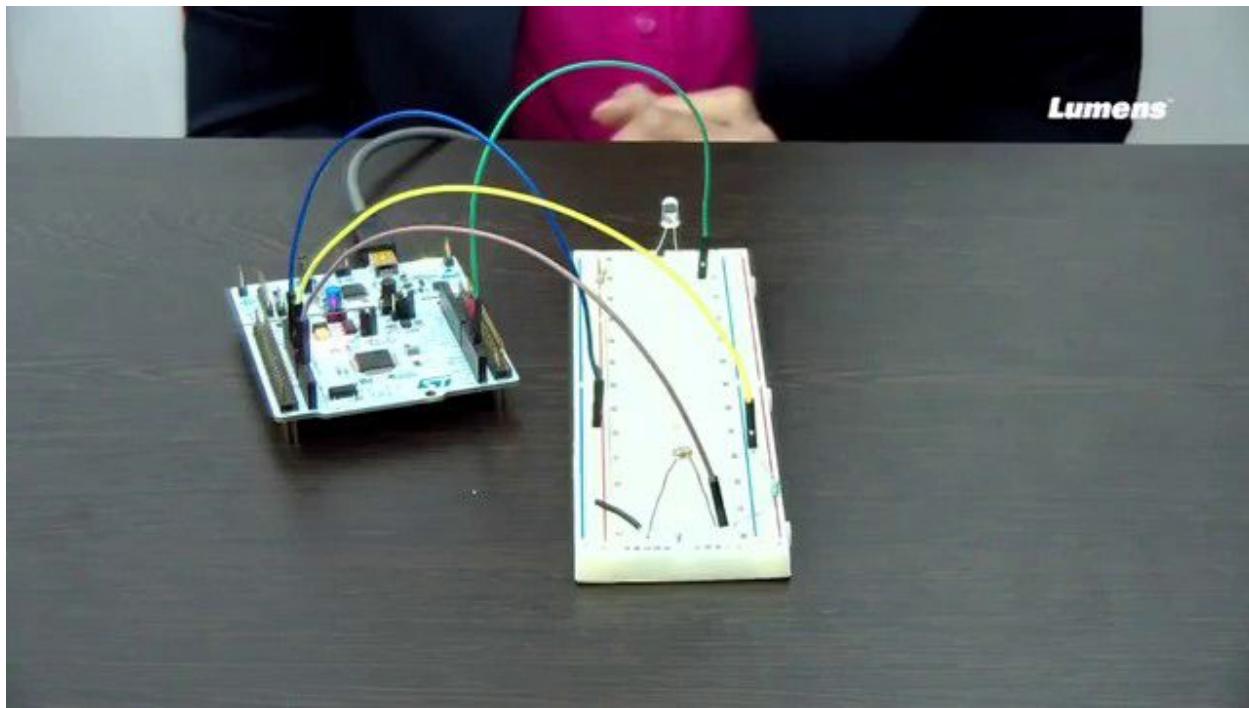
(Refer Slide Time: 09:15)



সুতরাং এটি হচ্ছে একটি LDR এবং এটি হচ্ছে একটি মাল্টিমিটার(multimeter)। এখন আমরা দেখতে পাব যে আমরা যখন এই নির্দিষ্ট পরিবেষ্টিত আলো ব্যবহার করে সংযোগ স্থাপন করব তখন এটি কী মান দেবে। এটা কি মান দিচ্ছে? এটি প্রায় 29 এর কাছাকাছি দিচ্ছে। আমি আমার হাত রাখব, আপনি দেখুন প্রতিরোধকের 130 বা এরকম কিছুতে পরিবর্তিত হয়েছে, তারপরে, আবার আমি সরিয়ে নেব এটি 29 এ স্থির হবে।

এখন, আমি এই ঘরের এই পরিবেষ্টিত আলোকে কিছুটা স্লান করে দেব। এখন আপনি দেখতে পাচ্ছেন যে প্রথমদিকে যখন পুরো আলো ছিল তখন আমি মানটি 29 পাঞ্চিলাম এবং এখন আমি যখন আমার হাতটি রাখি তখন আমরা একটি মান পেয়ে যাচ্ছি 41.7। এভাবেই প্রতিরোধের মান পরিবর্তন হয়।

(Refer Slide Time: 22:27)



সমস্ত সংযোগ করার পরে, কোড(code)টিকে আমি ফেলবো যা আপনার সাথে ইতিমধ্যে আলোচনা করেছি। এখন এলডিআর(LDR) পড়ার সাথে এলইডি(LED)টির তীব্রতা পরিবর্তিত হয়।

এই পরীক্ষায় আমরা মূলতঃ আপনাকে দেখিয়েছি আপনি কীভাবে LDR কে এসটিএম(STM) বোর্ডের সাথে সংযুক্ত করবেন এবং পরিবেষ্টনের আলোকের উপর নির্ভর করে আপনি কীভাবে বাস্তবে কোনও ধরণের অপারেশন(operation) করতে পারবেন।

ধন্যবাদ।

Embedded System Design with Arm
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya
Lecture – 27
Experiment With LDR Light Sensor (Part II)

বক্তৃতা 27 এ স্বাগতম। পূর্ববর্তী বক্তৃতায় আমি এলডিআর(LDR) সম্পর্কে আলোচনা করেছি এবং আপনাকে দেখিয়েছি যে কীভাবে পরিবেষ্টিত আলোকের উপর নির্ভর করে আমরা একটি এলইডি(LED) এর তীব্রতা পরিবর্তন করতে পারি। এই পরীক্ষায় আমরা আবার এলডিআর(LDR) ব্যবহার করব, আমরা আরও একটি পরীক্ষা করবো যেখানে আমরা এই ঘরে উপস্থিত ব্যক্তির সংখ্যা গণনা করব। অনুমান করা হয় যে এখানে একটি প্রবেশ পয়েন্ট(point) রয়েছে এবং একটি আলাদা প্রস্থান পয়েন্ট রয়েছে। সুতরাং, একটি দরজা থেকে আগত ব্যক্তি ভিতরে আসবে এবং সেই ব্যক্তিটি অন্য দরজা থেকে বেরিয়ে যাবে।

(Refer Slide Time: 01:19)

CONCEPTS COVERED

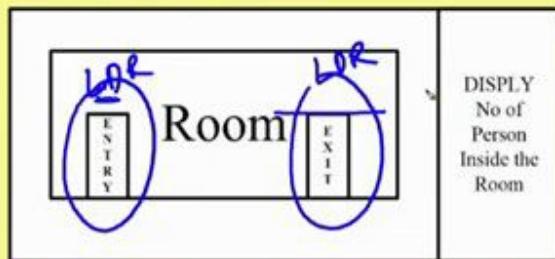
- Experiment of object counting using LDR light sensor
- Demonstration

FREE ONLINE EDUCATION **swayam** THE SWAYAM VISION

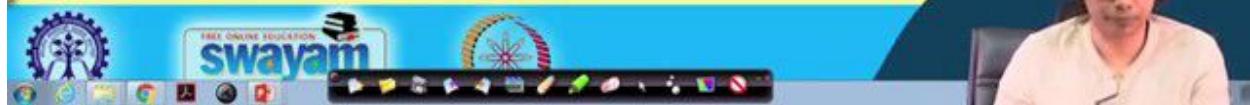
(Refer Slide Time: 01:29)

The Experiment

- Design a counter (using LDR and LED's), which will display the number of persons present in a room on a 7-segment LED display



- [Hint: Use two LDRs to detect direction of motion i.e., whether a person is entering or leaving]



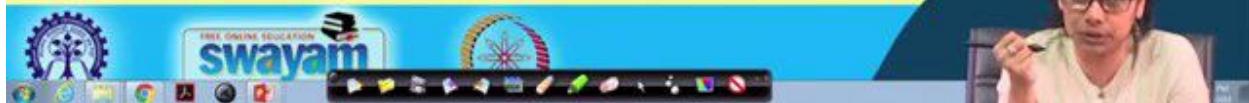
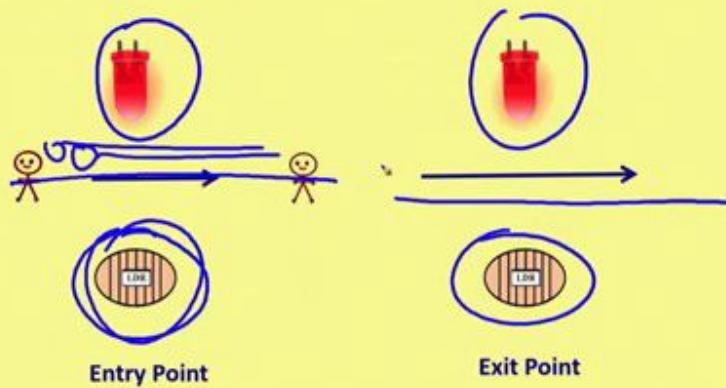
সুতরাং, এটি প্রবেশ পয়েন্ট, এটি প্রস্থান পয়েন্ট। এই প্রবেশের পয়েন্টে একটি এলডিআর(LDR) থাকবে এবং এই প্রস্থানস্থলে অন্য এলডিআর(LDR) থাকবে এবং একটি ঘরে উপস্থিত ব্যক্তির সংখ্যা একটি 7-সেগমেন্ট ডিসপ্লে(7-segment display) তে প্রদর্শন করবে।

এটি এলসিডি(LCD) তে প্রদর্শন করার পরিবর্তে, এই পরীক্ষায় আমরা এটি 7-সেগমেন্ট ডিসপ্লে(7-segment display) তে প্রদর্শন করব। কোনও ব্যক্তি প্রবেশ করছে বা চলে যাচ্ছে তার গতির দিক সনাক্ত করতে আমরা 2টি এলডিআর(LDR) ব্যবহার করব।

তারা কীভাবে প্রবেশ করছে তা আমাদের বিবেচনায় নেওয়ার দরকার নেই, কেবলমাত্র ব্যাপারটি হ'ল যদি এটি এই এলডিআর(LDR) টিকে অতিক্রম করে তবে কোনও ব্যক্তি প্রবেশ করেছে, যদি এটি এই এলডিআরটি অতিক্রম করে তার মানে ব্যক্তিটি বেরিয়ে যাচ্ছে।

(Refer Slide Time: 02:37)

The Arrangement



এভাবেই চলছে সাজানো ব্যবস্থা।

যখনই এখান থেকে কোনও বাধা আসবে; তার মানে, একজন ব্যক্তি প্রবেশ করেছে। এখান থেকে যখনই কোনও বাধা আসবে তখনই একজন ব্যক্তি বেরিয়ে যাচ্ছেন। সুতরাং, এখানে 2 টি পরিস্থিতি আছে। অন্যান্য অনেক প্রয়োগের জন্যও এলডিআর ব্যবহার করা যেতে পারে; আপনি একটি বোতলে প্রবেশ করা ট্যাবলেটগুলির সংখ্যা গণনা করতেও এটি ব্যবহার করতে পারেন। সুতরাং, সেক্ষেত্রে আপনাকে একই ধরণের ব্যবস্থা করতে হবে। আপনি এই এলডিআর(LDR) টিকে টিউব(tube) এর সামনের প্রাণ্টে রাখতে পারেন এবং আপনি ট্যাবলেটগুলিকে এভাবে একের পর এক করে রাখতে পারেন এবং প্রতিবার কোনও বাধা হলে কাউন্টার(counter) টি বাড়বে।

এখানে আমরা একটি ব্যক্তি কাউন্টার(person counter) বাস্তবায়িত করছি।

(Refer Slide Time: 04:15)

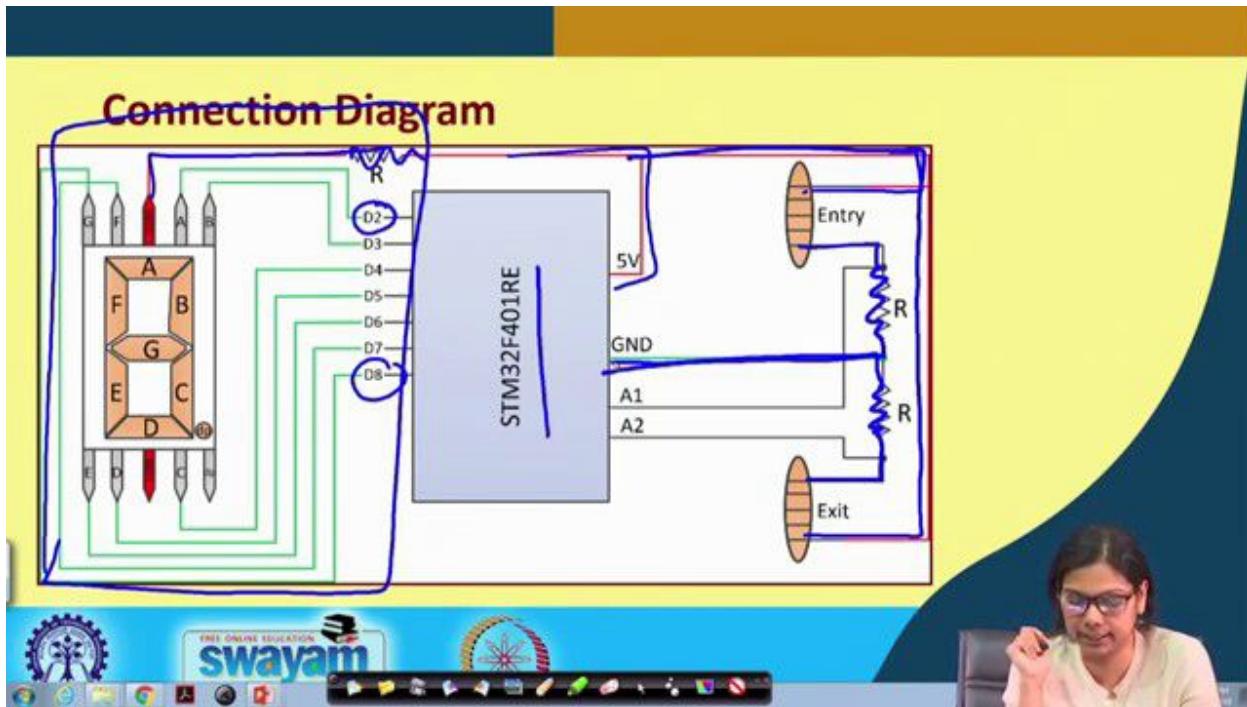
Basic Concept

- a) Initialize counter to 0.
- b) If the light intensity read by LDR from entry point decreases, increment the counter.
- c) If the light intensity read by LDR from exit point decreases, decrement the counter.



এখানে মূল ধারণাটি কী? আমরা 0 তে একটি কাউন্টার শুরু করি যখনই কোনও বাধা আসে তখন আলোর তীব্রতা হ্রাস পায় যা কাউন্টারকে বাড়িয়ে তোলে। প্রবেশ পয়েন্ট থেকে যদি এলডিআর(LDR) মান এর কোনও পরিবর্তন ঘটে থাকে, প্রতিবার এটি ঘটলে আমরা কাউন্টারটি বাড়ায়। যদি প্রস্থান স্থান থেকে এলডিআর দ্বারা পড়া আলোক তীব্রতা আবার হ্রাস পায় তবে আমাদের কাছে আরও একটি পয়েন্ট রয়েছে যা প্রস্থান পয়েন্ট; যদি এটি প্রবেশ পয়েন্ট থেকে হয় তবে এটি বৃদ্ধি পায়; যদি এটি প্রস্থানস্থান থেকে হয় তবে কাউন্টারটি হ্রাস পায়।

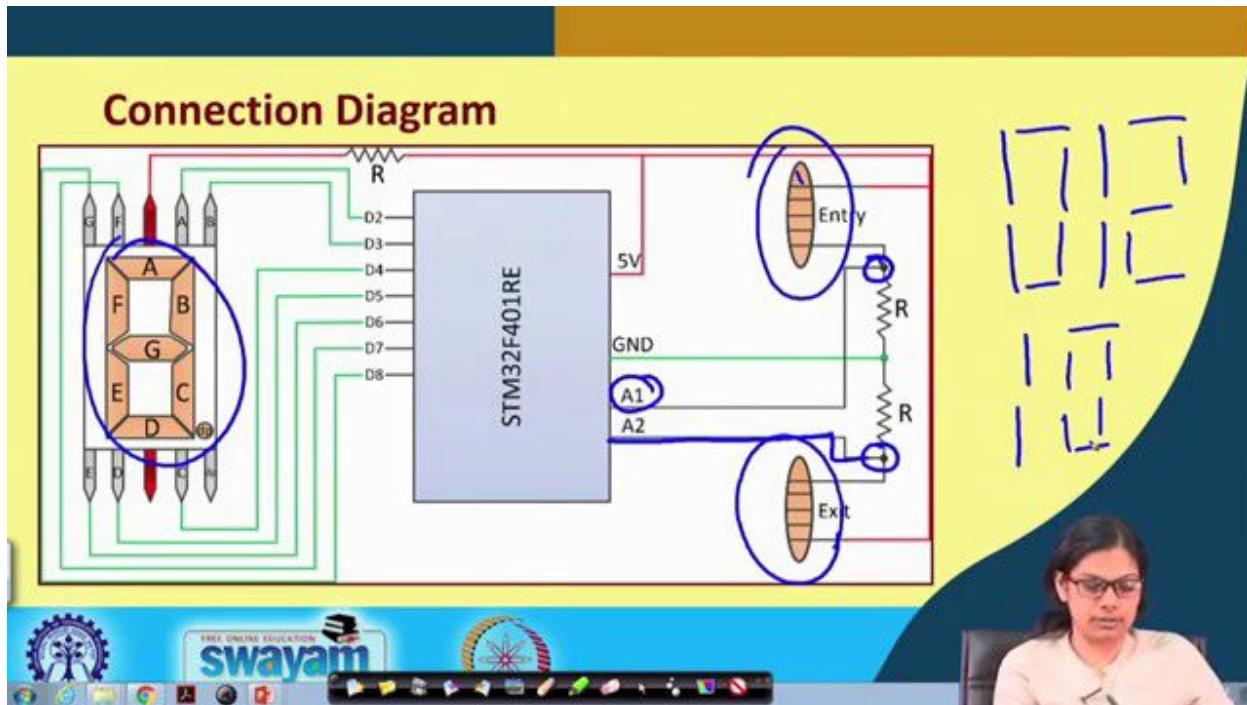
(Refer Slide Time: 06:21)



এটি সংযোগ ডায়াগ্রাম(diagram) যা সহজ-সরল। সংযোগ ডায়াগ্রাম(diagram) এর এই অংশটি আমরা ইতিমধ্যে আলোচনা করেছি, যেখানে এই ডিসপ্লে(display)-এর সমস্ত বিভাগগুলি(segments) পিন নম্বর D2 থেকে এসটিএম(STM) এর D8 এ সংযুক্ত রয়েছে। একটি প্রতিরোধের মাধ্যমে সাধারণ বিলু থেকে, সাধারণ আনোড়টি 5 ভোল্টের সাথে সংযুক্ত থাকে। যখনই কোনও বাধা থাকে তখনই এই প্রদর্শনটিতে একটি নির্দিষ্ট ঘরে উপস্থিত ব্যক্তির সংখ্যা প্রদর্শন করে।

এখন আসুন আমরা এই এলডিআর(LDR)টির সংযোগটি দেখি; এলডিআর(LDR) এর এক প্রান্তটি সরাসরি 5 ভোল্টের সাথে সংযুক্ত থাকে। এবং অন্য প্রান্ত প্রতিরোধের মাধ্যমে গ্রাউন্ড(ground) এ সংযুক্ত করা হয়।

(Refer Slide Time: 07:53)



প্রবেশ পয়েন্ট থেকে এটি পোর্ট(port) A1 এর সাথে সংযুক্ত রয়েছে। এবং প্রস্থান স্থান থেকে এটি পোর্ট A2 এর সাথে সংযুক্ত, এটি সংযোগ সম্পর্কে সমস্ত।

এখন যখনই এই নির্দিষ্ট এলডিআর(LDR)টিতে কোনও বাধা আসবে তখন কী হবে? এই প্রদর্শনটি 1, 2, 3, ইত্যাদি বর্ধিত হবে। এখন কেউ বাইরে বেরলে গণনা হ্রাস পাবে।

(Refer Slide Time: 09:29)

The Mbed C Code

```
#include "mbed.h"
#include "string.h"
DigitalOut A(D2);
DigitalOut B(D3);
DigitalOut C(D4);
DigitalOut D(D5);
DigitalOut E(D6);
DigitalOut F(D7);
DigitalOut G(D8);
AnalogIn Entry(A1);
AnalogIn Exit(A2);
```

```
void Display(int disp){
switch(disp)
{
    case 0: A=0;B=0;C=0;D=0;E=0;F=0;G=1; break;
    case 1: A=1;B=0;C=0;D=1;E=1;F=1;G=1; break;
    case 2: A=0;B=0;C=1;D=0;E=0;F=1;G=0; break;
    case 3: A=0;B=0;C=0;D=0;E=1;F=1;G=0; break;
    case 4: A=1;B=0;C=0;D=1;E=1;F=0;G=0; break;
    case 5: A=0;B=1;C=0;D=0;E=1;F=0;G=0; break;
    case 6: A=0;B=1;C=0;D=0;E=0;F=0;G=0; break;
    case 7: A=0;B=0;C=0;D=1;E=1;F=1;G=1; break;
    case 8: A=0;B=0;C=0;D=0;E=0;F=0;G=0; break;
    case 9: A=0;B=0;C=0;D=0;E=1;F=0;G=0; break;
}
}
```

A B C D E F



কোডটি আবার স্ব-ব্যাখ্যামূলক। আমাদের দুটি অ্যানালগ(analog) পোর্ট(port) রয়েছে, একটি A1, অন্যটি A2।

(Refer Slide Time: 10:45)

```

int main()
{
    float en, ex;
    int per=0;
    while(1) {
        en = Entry.read(); ✓
        ex = Exit.read(); ✓
        en=en*5000; ✓
        ex=ex*5000; ✓
        if (en<200)
        {
            per=per+1; ✓
            Display(per); ✓
            wait(1); ✓
        }
        else if(ex<200)
        {
            per=per-1; ✓
            Display(per); ✓
            wait(1); ✓
        }
        else if (per<=0 || per>9)
        {
            per=0; ✓
            Display(per); ✓
            wait(1); ✓
        }
        else
        {
            Display(per); ✓
            wait(1); ✓
        }
    }
}

```

এখন আসুন মূল প্রোগ্রাম(program)টি দেখি। প্রাথমিক ভাবে আমরা ব্যক্তির সংখ্যা 0 করেছি। while(1) লুপ(loop) এর মধ্যে, Entry.read() প্রবেশ পয়েন্টটির জন্য কিছু মান দেবে এবং Exit.read() প্রস্থান স্থান থেকে মানটি দেবে। উভয় মান 5000 দিয়ে গুণিত হয়।

যদি "en" 200 এর চেয়ে কম হয়, তবে ব্যক্তির সংখ্যা 1 দ্বারা বৃদ্ধি করা হয়। যদি প্রতিরোধ 200 এর চেয়ে কম হয় তবে একটি বাধা হয় এবং আমরা এটি বৃদ্ধি করেছি, আমরা এটি Display ফাংশন(function) এ প্রেরণ করি এবং 1 সেকেন্ডের জন্য অপেক্ষা করি। অন্যথায় আমাদের প্রস্থান স্থান থেকেও পরীক্ষা করতে হয়। যদি "ex" 200 এর কম হয়, তবে ব্যক্তির গণনা 1 হ্রাস পাবে।

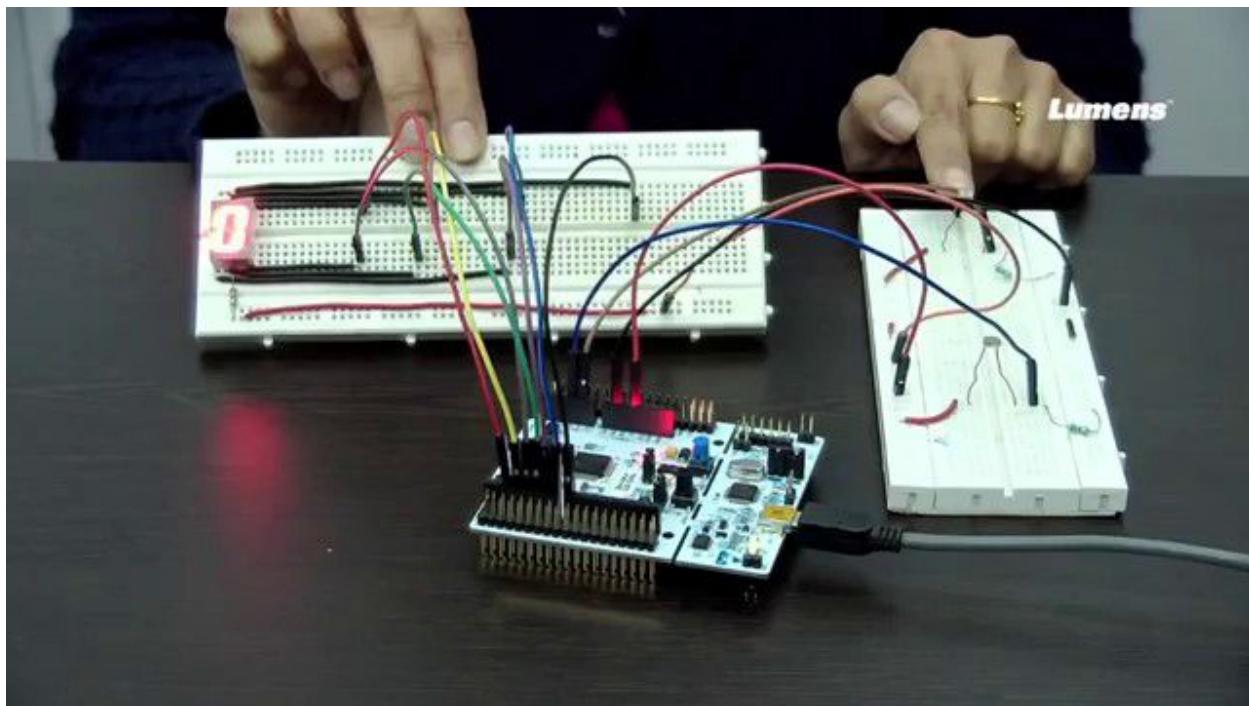
এর পরে আমরা আপনাকে একই কোডটির প্রদর্শনটি দেখাবো যা আমি এখনই আলোচনা করেছি।

আমি এখন আপনাকে সেই পরীক্ষাটি দেখাচ্ছি যাতে আপনি কোনও নির্দিষ্ট ঘরে উপস্থিত ব্যক্তির সংখ্যা গণনা করতে পারেন।

আপনার কাছে 2 টি এলডিআর(LDR) রয়েছে একটি প্রবেশ পয়েন্টে এবং অন্য একটি প্রস্থান পয়েন্টে। যখনই কোনও বাধা পাবে তখন কাউন্টারটি বাড়বে যদি কেউ প্রবেশ পয়েন্টটি অতিক্রম করে, এবং যদি কেউ প্রস্থান

পয়েন্টটি অতিক্রম করে তবে সেই নির্দিষ্ট সংখ্যাটি হ্রাস পাবে। যে কোনও নির্দিষ্ট সময়ে কোনও কক্ষে উপস্থিত ব্যক্তির সংখ্যা একটি 7 সেগমেন্ট ইউনিট (7 segment unit) এ প্রদর্শিত হবে।

(Refer Slide Time: 16:13)



এভাবে আপনাকে 2 টি এলডিআর(LDR) লাগাতে হবে; আসুন আমরা ধরি এটি প্রবেশ পয়েন্ট এবং এটি একটি প্রস্থান পয়েন্ট। এখন আমি এটি 7 সেগমেন্টের এলইডি (7 segment LED) তে প্রদর্শিত করব, আমরা ইতিমধ্যে 7 সেগমেন্ট ডিসপ্লে(7 segment display) ইউনিটের সাথে কীভাবে সংযোগ স্থাপন করব তা জানি। সমস্ত সংযোগ তৈরি করা হল।

সুতরাং, এটি আমার প্রবেশ এবং এটি আমার প্রস্থান। আসুন দেখুন এটি কীভাবে কাজ করে। আমাকে এটি সংযোগ করতে দিন এবং কোডটি ফেলতে দিন; প্রাথমিকভাবে এটি 0 গননা প্রদর্শিত হয়। এখন যেমন আমি বলেছিলাম যে এই এলডিআর(LDR)টি আমার প্রবেশ বিন্দু, আসুন দেখি আমি বাধা দিলে কী ঘটছে। এটি 1 দ্বারা বৃক্ষি পেয়েছে। আমি আরও একটি বাধা তৈরি করব, এটি 2 এ বৃক্ষি পেয়েছে, অন্য বাধায় 3 এ পরিণত হয়েছে।

আসুন দেখা যাক, প্রস্থান স্থানে যদি কোনও বাধা থাকে তবে কি ঘটবে, এটি 2 হয়, তারপরে এটি 1 হয়ে যায়। আমি আবার কিছু প্রবেশ করব। এটি সঠিকভাবে কাজ করে।

আমি আশা করি এলডিআর(LDR) নিয়ে এই পরীক্ষাটি আপনাদের সকলের কাছে পরিষ্কার।

ଧନ୍ୟବାଦ ।

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya
Lecture – 28
Experiment with Speaker

বক্তৃতা 28 এ স্বাগতম। এই বক্তৃতায় আমরা আপনাকে স্পিকার(Speaker) এর সাথে কিছু পরীক্ষা-নিরীক্ষা দেখাবো। আমরা এসটিএম(STM) বোর্ডের সাথে স্পিকার(Speaker) কে ইন্টারফেস(interface) করব এবং 4টি বিভিন্ন ধরণের সুর তৈরি করব। প্রথমত, আমরা সংযোগ ডায়াগ্রাম(diagram)টি কেন্দ্র হবে তা খতিয়ে দেখব। PWM পোর্ট(Port) আমরা ইতিমধ্যে 3 সপ্তাহে বিস্তারিত আলোচনা করেছি। আমি ইতিমধ্যে আলোচিত কিছু ফাংশন(function) ব্যবহার করব, যা আমি স্পিকারের সাথে ব্যবহার করব এবং তারপরে আমি আপনাকে দেখাবো কীভাবে আমরা কোড(code)টি পরিবর্তন করে স্পিকারের মাধ্যমে বিভিন্ন ধরণের সিগন্যাল(signal), বিভিন্ন ধরণের শব্দ তৈরি করতে পারি।

(Refer Slide Time: 01:14)

CONCEPTS COVERED

- ❑ Speaker interface circuit
- ❑ Experiments using speaker
- ❑ Demonstration

The slide is part of the Swayam platform, as indicated by the logo at the bottom right.

প্রথমত, আমরা সার্কিট(circuit)টি দেখাব এবং তারপরে আমরা স্পিকারের সাথে বিভিন্ন পরীক্ষা-নিরীক্ষা করব এবং অবশেষে আমি প্রদর্শন করব।

(Refer Slide Time: 01:27)

CONCEPTS COVERED

- ❑ Speaker interface circuit
- ❑ Experiments using speaker
- ❑ Demonstration

swayam

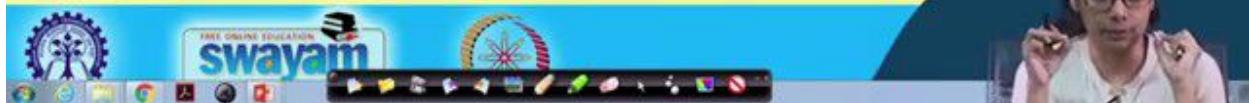
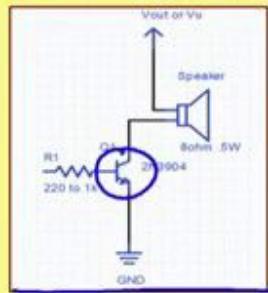
সুতরাং, কিভাবে একটি স্পিকার(speaker) কাজ করে? এটি ইতিমধ্যে আগের সপ্তাহে আলোচনা করা হয়েছে। একটি স্পিকার বৈদ্যুতিক শক্তিকে যান্ত্রিক শক্তিতে রূপান্তর করে কাজ করে। যান্ত্রিক শক্তি বায়ুকে সংকুচিত করে এবং গতিকে শব্দ শক্তিতে রূপান্তর করে।

এর অভ্যন্তরে স্থায়ী চৌম্বক রয়েছে এবং একটি কয়েল(coil) এবং ডায়াগ্রাম(diagram) রয়েছে যার মাধ্যমে এটি একরকম শব্দ উত্পন্ন করে।

(Refer Slide Time: 02:21)

How to Interface a Speaker?

- Any waveform in the audio frequency range from an output port can drive the speaker.
 - May require an amplifier circuit to generate adequate power for the electromagnet.
- A simple transistor based amplifier circuit:



আমরা কীভাবে স্পিকার(speaker) কে ইন্টারফেস(interface) করব? আপনি যদি কোনও আউটপুট(output) পোর্ট থেকে অডিও(audio) ফ্রিকোয়েন্সি(frequency) রেঞ্জের যে কোনও তরঙ্গক্ষেত্রে কথা ভাবেন তবে স্পিকারটি চালনা করতে পারে তবে কখনও কখনও আমরা দেখতে পাই যে আমাদের স্পিকারের অভ্যন্তরে থাকা বৈদ্যুতিন চোক্সকটির জন্য পর্যাপ্ত শক্তি উত্পন্ন করার জন্য একটি পরিবর্ধক(amplifier) সার্কিট(circuit) এর প্রয়োজন হতে পারে।

এটি একটি সাধারণ সার্কিট ডায়াগ্রাম (circuit diagram), একটি সাধারণ ট্রানজিস্টর(transistor) ভিত্তিক এমপ্লিফায়ার সার্কিট(amplifier circuit)। যদিও আমাদের ক্ষেত্রে, আমরা কোনও ধরণের সার্কিট(circuit) ব্যবহার করি নি, তবে আপনি কোনও স্পিকারের ইন্টারফেস করতে গেলে এই জাতীয় সার্কিট(circuit) ব্যবহার করা প্রয়োজন হতে পারে।

(Refer Slide Time: 03:29)

- Small speakers (e.g. 8 ohms or less) can also be driven directly from the port lines of a microcontroller.
 - Of course, sound output will be low.



আমরা একটি ছোট স্পিকারকে ইন্টারফেস করেছি, যা আমরা এটিকে সরাসরি একটি পোর্ট লাইনে সংযুক্ত করেছি। আমরা একটি PWM পোর্ট লাইনে সংযোগ করেছি। এগুলি স্পিকারের দুটি প্রান্ত; একটি গ্রাউন্ড সংযুক্ত, একটি PWM-out এ সংযুক্ত হবে। অবশ্যই, আমরা একটি খুব উচ্চ স্তর এর শব্দ তৈরি করতে সক্ষম হব না, তবে শব্দটি একটি যুক্তিসংজ্ঞত স্তর এর যা আপনি শুনতে পারেন।

(Refer Slide Time: 04:23)

Experiment 1

- Produce a fixed tone on the speaker by generating a square wave of a particular frequency, and use it to drive the speaker.
 - Audible sound is in the frequency range 20 Hz to 20 KHz.
 - In this experiment, we generate a tone with frequency 250 Hz.
 - We can use a pulse width modulated (PWM) port to generate the speaker control signal, by specifying:
 - a) The time period of the generated waveform (i.e. 1 / frequency)
 - b) The duty cycle of the generated waveform : $\frac{T_{ON}}{T_{ON} + T_{OFF}}$



যেমনটি আমি আপনাকে বলেছি আমরা কয়েকটি পরীক্ষা-নিরীক্ষা করব।

প্রথম পরীক্ষায়, আমরা একটি নির্দিষ্ট ফ্রিকোয়েন্সি(frequency) এর বর্গাকার তরঙ্গ(square wave) তৈরি করে স্পিকারের উপর একটি স্থির সূর তৈরি করব, যা আমরা স্পিকারটি চালনার জন্য এটি ব্যবহার করব। আমরা সকলেই জানি যে শ্রবণযোগ্য ফ্রিকোয়েন্সি পরিসীমা 20 হার্টজ(hertz) থেকে 20 কিলোহার্টজ(kilohertz) হয়।

এই পরীক্ষায় আমরা 250 হার্টজ ফ্রিকোয়েন্সি উত্পন্ন করি যা আমাদের কাছে শ্রবণযোগ্য। নিম্নলিখিতগুলি উল্লেখ করে আমরা স্পিকার নিয়ন্ত্রণ সংকেত(control signal) তৈরি করতে PWM পোর্টটি ব্যবহার করতে পারি। আমাদের উত্পন্ন ওয়েবফর্ম(waveform) বা তরঙ্গন্ত্রের সময়কাল এবং তরঙ্গন্ত্রের ডিউটি সাইকেল(duty cycle) উল্লেখ করতে হবে।

(Refer Slide Time: 05:43)

The PwmOut API :: Relevant Member Functions

- PwmOut (PinName pin)
 - Create a PwmOut connected to the specified pin.
- void write (float value)
 - Set the output duty cycle, specified as a percentage.
- void period (float seconds)
 - Set the PWM period, specified in seconds.

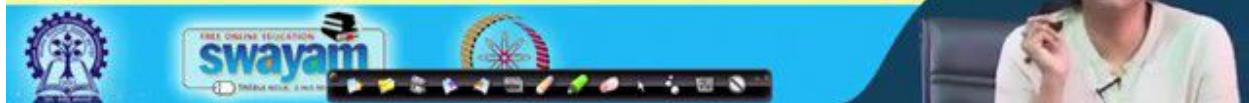
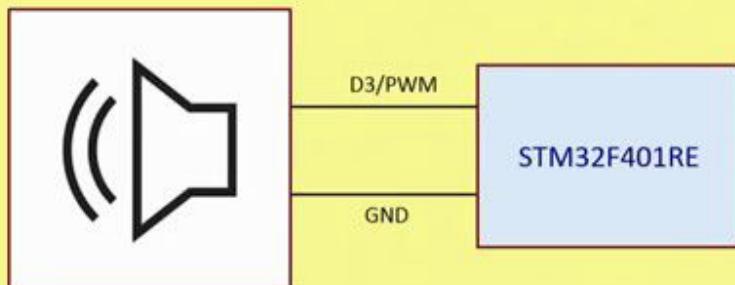


এই ফাংশনগুলি আমরা ইতিমধ্যে পূর্ববর্তী সপ্তাহগুলিতে আলোচনা করেছি। PwmOut নির্দিষ্ট পিনের সাথে সংযুক্ত একটি PwmOut সংকেত তৈরি করে।

void write() এ, এটি শতাংশ হিসাবে নির্ধারিত আউটপুট ডিউটি সাইকেল(output duty cycle)সেট করবে, শতাংশের সময় এটি চালু আছে।। একইভাবে, void period() PWM পিরিয়ড(period) সেকেন্ডে সেট করতে পারে। আরও অনেক ফাংশন(function) রয়েছে, যা বিস্তারিত ভাবে আগে আলোচনা করা হয়েছে।

(Refer Slide Time: 07:14)

Connection Diagram – STM32



এখন, এটি সার্কিট ডায়াগ্রাম, আপনি দেখতে পাবেন যে আমি যখন প্রদর্শন করব সহজভাবে আমি স্পিকারের এক প্রান্তি গ্রাউন্ড(ground) এ এবং অন্য প্রান্তি PWM পোর্টের সাথে সংযুক্ত করব যা D3।

(Refer Slide Time: 07:44)

Mbed program for Speaker (beep)

```
#include "mbed.h"
DigitalOut speaker(D3);
int main() {
    while(1) {
        speaker = 0;
        wait (0.002);           // Wait 2 msec
        speaker = 1;
        wait (0.002);           // Wait 2 msec
    }
}
```

This program does not use PWM. It simply outputs 0 and 1 alternately, with time period 4 msec, i.e. frequency of 250 Hz.



এখন, আমি আপনাকে বিভিন্ন প্রোগ্রাম(program) দেখাবো। প্রথম প্রোগ্রাম(program) টি 250 হার্টজ ফ্রিকোয়েন্সি(frequency)-এর একটি নির্দিষ্ট সূর তৈরি করবে। আমরা 0.002 সেকেন্ডের ওয়েট(weight) সহ

স্পিকার আউটপুট 0 দিছি যা 2 মিলিসেকেন্ড, এবং স্পিকার 0,002 সেকেন্ডের ওয়েট(weight) সহ 1 এ সেট করা হয়েছে, যা 2 মিলিসেকেন্ড। এই কোডটি ফ্রিকোয়েন্সি 250 হার্টজ(hertz) স্কোয়ার ওয়েভ(square wave) উত্পন্ন করে। আমরা এই বক্তৃতার শেষে এই পরীক্ষাটি প্রদর্শন করব।

(Refer Slide Time: 09:28)

Experiment 2

- Generate a two-frequency tone on the speaker, alternating between 333 Hz and 455 Hz; playing each frequency tone for 0.5 second.
 - Play 333 Hz tone for 0.5 second (time period = 3.0 msec).
 - Play 455 Hz tone for 0.5 second (time period = 2.2 msec).
 - Repeat the process in a loop.

$$\frac{1}{333} = 3 \text{ ms}$$

$$\frac{1}{455} = 2.2$$

এই পরীক্ষায়, আমরা স্পিকারের উপর 2-ফ্রিকোয়েন্সি সূর উত্পন্ন করব, প্রতি 0.5 সেকেন্ডে বদলে বদলে একবার 333 হার্টজ(hertz) এবং একবার 455 হার্টজ(hertz)। যদি এটি 333 হার্টজ(hertz) হয় তবে সময়কাল 1 333 দ্বারা বিভক্ত হবে; 1 কে 333 দ্বারা বিভক্ত করলে 3 মিলিসেকেন্ড আসে।

455 হার্টজ(hertz) জন্য, 1 কে 455 দ্বারা বিভক্ত করলে 2.2 মিলি সেকেন্ড আসে।

(Refer Slide Time: 10:39)

Mbed program for Speaker (Ambulance Sound)

```
#include "mbed.h"
PwmOut mypwm(D3);

int main() {
    mypwm.period_us(3000);
    mypwm.pulsewidth_us(1550);
    while(1) {
        mypwm.period_us(3000);
        mypwm.pulsewidth_us(1550);
        wait(0.5); // Play tone-1 for 0.5 sec
        mypwm.period_us(2200);
        mypwm.pulsewidth_us(100);
        wait(0.5); // Play tone-2 for 0.5 sec
    }
}
```

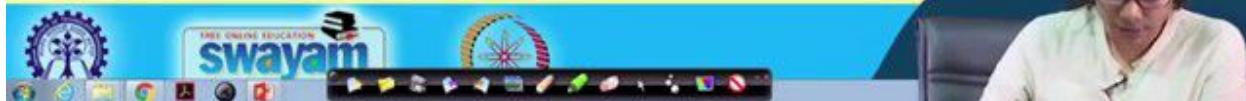


Let us see the code. কোডটি দেখুন। The code is self-explanatory. কোডটি স্ব-বর্ণনামূলক।

(Refer Slide Time: 12:34)

Experiment 3

- Generate the 12 musical notes of an octave on the speaker.
 - Start with the base frequency of 300 Hz.
 - The frequency of the next note is generated by multiplying with 12^{th} root of 2 (i.e. 1.0595).
 - After completing a cycle, it will repeat.



পরবর্তী পরীক্ষায় আমরা স্পিকারের উপর একটি অষ্টাভের 12 টি সংগীত নোট তৈরি করব। আমরা বেস ফ্রিকোয়েন্সি (base frequency) দিয়ে শুরু করি, ধরুন 300 হার্টজ এবং তারপরে পরবর্তী সূরটির ফ্রিকোয়েন্সি 2 এর 12 ম মূল দিয়ে গুণ করে তৈরি করা হয় যা 1.0595। এটি পুনরাবৃত্তি হয়।

Refer Slide Time: 13:26)

Mbed program for Speaker (Sa Re Ga Ma)

```
#include "mbed.h"
PwmOut speaker(D3);

int main() {
    float freq, period;
    int i;
    speaker.write(0.5); // Set duty cycle to 0.5
    while (1) {
        freq = 300.0; // Start with 300 Hz
        for (i=0; i<12; i++) {
            period = 1.0 / freq;
            speaker.period(period);
            wait(0.5);
            freq = freq * 1.0595; // Multiply by 12th root of 2
        }
        wait(1);
    }
}
```

এই কোডটিও সহজ-সরল।

(Refer Slide Time: 15:58)

Experiment 4

- Here the successive notes of a musical piece is played in a repetitive fashion.



পরবর্তী পরীক্ষায়, আমরা একটি সংগীত অংশের নোটগুলি বাজাতে যাচ্ছি।

আমরা কয়েকটি নোট (ফ্রিকোয়েন্সি) নির্বাচন করেছি যা আমরা নির্দিষ্ট সময়ের জন্য বাজিয়ে নির্দিষ্ট সূর তৈরি করব।

(Refer Slide Time: 15:33)

Mbed Program for Speaker (Musical Tone)

```

#include "mbed.h"

PwmOut speaker(D3);

int main() {
    float freq, period;
    int i;
    speaker.write(0.5);
    // Set duty cycle to 0.5
    while (1) {
        freq = 262.0;
        period = 1.0 / freq;
        speaker.period(period);
        wait(0.5);
        freq = 262.0;
        period = 1.0 / freq;
        speaker.period(period);
        wait(0.5);
        freq = 294.0;
        period = 1.0 / freq;
        speaker.period(period);
        wait(0.5);
        freq = 329.0;
        period = 1.0 / freq;
        speaker.period(period);
        wait(0.5);
        freq = 349.0;
        period = 1.0 / freq;
        speaker.period(period);
        wait(0.5);
        freq = 330.0;
        period = 1.0 / freq;
        speaker.period(period);
        wait(0.5);
        wait(0.5);
    }
}

```

সুতরাং, এটি মূলতঃ এই সমস্ত যা স্পিকার সম্পর্কে আপনাকে বলেছি এবং তারপরে আমি এগিয়ে যাব এবং আমি আপনাকে দেখাব আমি কীভাবে এটি ইন্টারফেস(interface) করব, আমি কীভাবে এই বিশেষ 4 টি ইন্টারফেস করব, স্পিকার এবং মাইক্রোকন্ট্রোলার বোর্ড ব্যবহার করে 4 টি কোড এর সরণিলি। সুতরাং এখন, আমি আপনাকে স্পিকারের 4 টি কোড দেখাব যা আমি এখন আলোচনা করেছি ঠিক আছে।

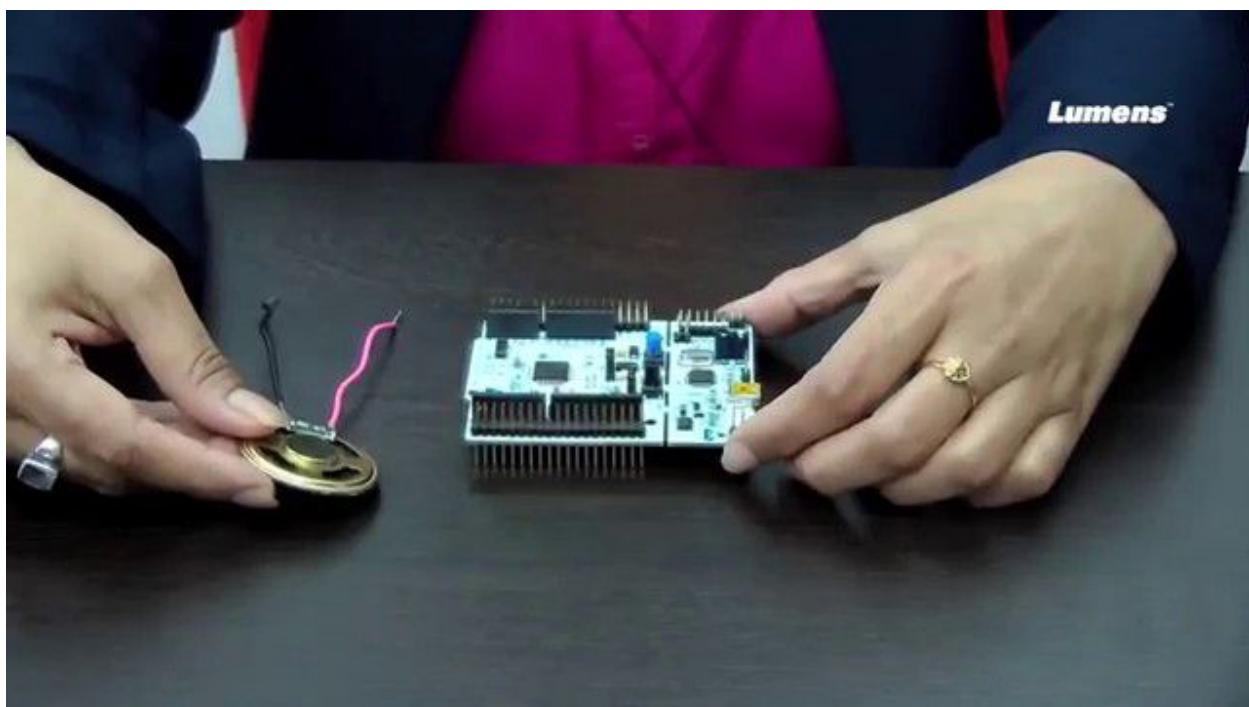
সুতরাং, ওই 4 টি কোড কী করছে? প্রথম কোডে, আমরা কেবল দীর্ঘ সময়ের জন্য কিছু এলোমেলো বীপ(beep) শব্দ করব। পরের কোডটি কি করছে? এটি সাইরেনের মতো শব্দ করছে। তারপরে পরের কোডটি 12 টি অক্ষোব্রে একে একে বাজে।

আসুন সার্কিটটি খতিয়ে দেখা যাক।

(Refer Slide Time: 19:06)

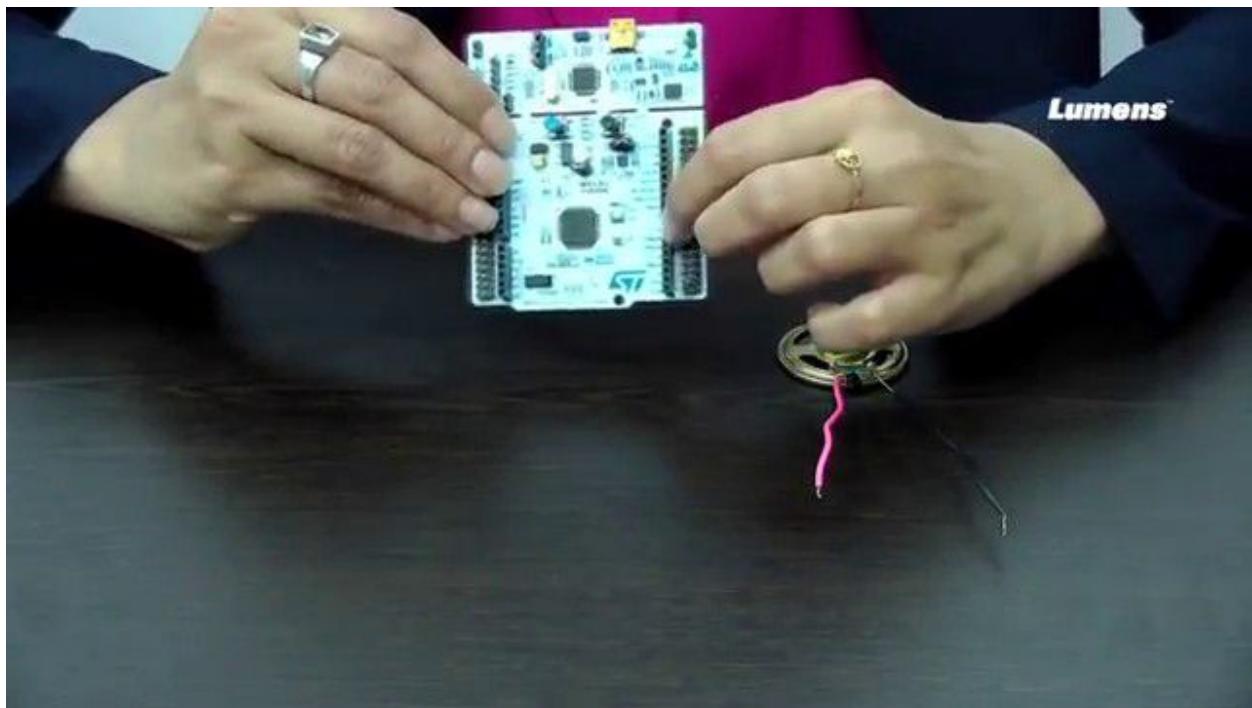


(Refer Slide Time: 19:11)



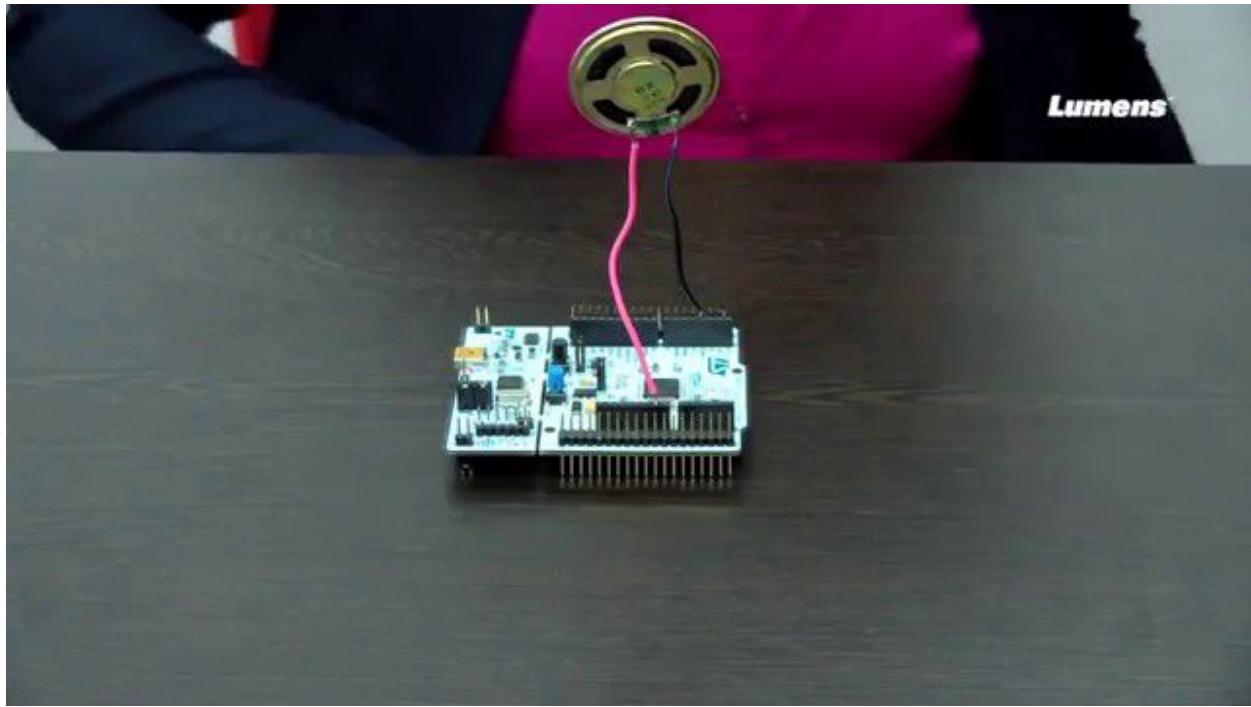
এটি এমন একটি স্পিকার যা আমরা আলোচনা করেছি। আমি যা করবো তা হল এর এক প্রান্তিটি গ্রাউন্ড(ground) এ এবং অন্য প্রান্তিটি PWM পোর্ট এ সংযুক্ত করব।

(Refer Slide Time: 19:47)



বোর্ডে, অনেকগুলি PWM পোর্ট রয়েছে, যার মধ্যে আমি D3 পোর্টটি ব্যবহার করব। তারপরে আমি চারটি কোড ফেলবো যা আমি বলেছিলাম। আমাকে এখন স্পিকারটি সংযোগ করতে দিন।

(Refer Slide Time: 20:38)



আমি এক প্রান্ত গ্রাউন্ড(ground) এ এবং অন্য প্রান্তটি D3 এর সাথে সংযুক্ত করব।

প্রথম কোডটি একটি নির্দিষ্ট শব্দ উত্তপ্ত করবে যা আপনি শুনতে পারবেন।

আমি এখন দ্বিতীয় কোডটি ফেলবো। মূলতঃ এটি একটি সাইরেনের শব্দ।

এখন, আমরা তৃতীয়টির সাথে এগিয়ে চলি, যা মূলতঃ সা রে গা মা শব্দ।

এখানে পরবর্তী, আমি শেষ কোডটি ফেলবো যা হ্যাপি বার্থডে(happy birthday) সুরটি বাজায়।

ধন্যবাদ।

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya
Lecture – 29
Experiment with Microphone

বক্তৃতা 29-এ স্বাগত। এই বক্তৃতায় আমি আপনাকে একটি মাইক্রোফোন(microphone) দিয়ে কিছু পরীক্ষা-নিরীক্ষা দেখাবো। মাইক্রোফোন(microphone) কী করে তা আমরা জানি।আমি আপনাকে দুটি পরীক্ষা দেখাব। একটি পরীক্ষায় এটি তালি(clap)র সংখ্যা প্রদর্শন করবে।

(Refer Slide Time: 00:55)

CONCEPTS COVERED

- ❑ Microphone interfacing
- ❑ Experiments using microphone
- ❑ Demonstration

SWAYAM

FREE ONLINE EDUCATION

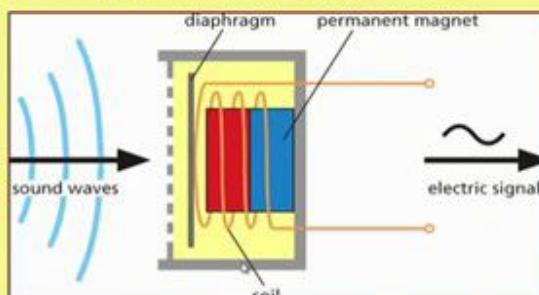
SWAYAM

SWAYAM

(Refer Slide Time: 01:11)

What is a Microphone?

- A microphone is a type of transducer, that converts acoustical energy (sound waves) into electrical energy (the audio signal).



The diagram illustrates the internal mechanism of a microphone. It shows a cross-section of the device with a diaphragm at the top, a permanent magnet in the middle, and a coil at the bottom. Sound waves enter through the diaphragm, causing it to vibrate. This vibration is converted into a corresponding motion of the coil within the magnetic field of the permanent magnet. The resulting electrical signal is then output from the microphone.

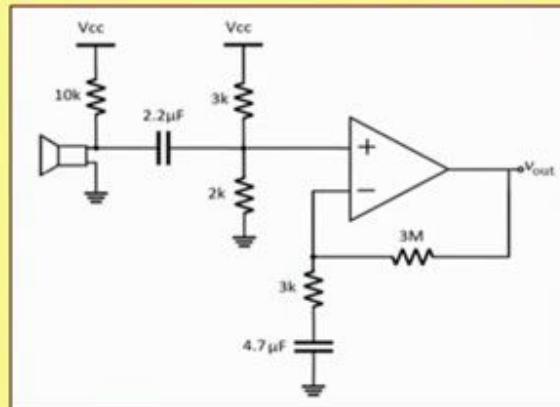
SWAYAM FREE ONLINE EDUCATION



মাইক্রোফোন(Microphone) হ'ল এক ধরণের ট্রান্সডুসার(transducer) যা শব্দ শক্তিকে বৈদ্যুতিক শক্তিতে রূপান্তর করে। আমরা দেখতে পাচ্ছি যে এখানে শব্দ তরঙ্গ আসছে এবং এখানে একটি ডায়াফ্রাম(diaphragm) এবং একটি স্থায়ী চুম্বক রয়েছে। এখানে কিছু অপারেশন(operation) ঘটে এবং এই শব্দ তরঙ্গ গুলি এখানে আসলে একরকম বৈদ্যুতিক সংকেত তৈরি করে।

(Refer Slide Time: 01:51)

Typical Microphone Interfacing Circuit

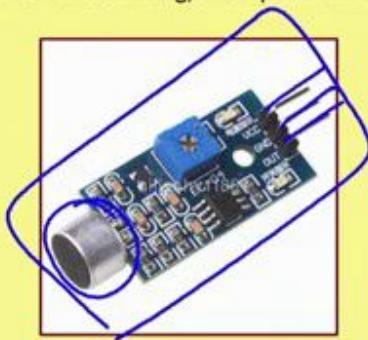


এটি একটি সাধারণ মাইক্রোফোন ইন্টারফেসিং সার্কিট(Interfacing Circuit)। আপনি যদি এটি দেখেন এখানে গ্রাউন্ড(ground) এবং অন্যান্য সমস্ত জিনিসগুলির সাথে কিছু প্রতিরোধের মাধ্যমে বেশ কয়েকটি সংযোগ রয়েছে, তবে সাধারণত যা করা হয় তা হল এখানে দেখানো ইন্টারফেসিং(interfacing) এর পুরো সেটটি আসলে এই নির্দিষ্ট চিপ(chip) এর ভিতরে রাখা হয়েছে।

(Refer Slide Time: 02:13)

Microphone Module with Interface

- For ease of interfacing, microphone with driver circuit is available as a module.



- Can work with power supply of 3.3V to 5.0V.
- Digital output (0 or 1), depending on whether sound is detected or not.
- Can be directly interfaced with the microcontroller.

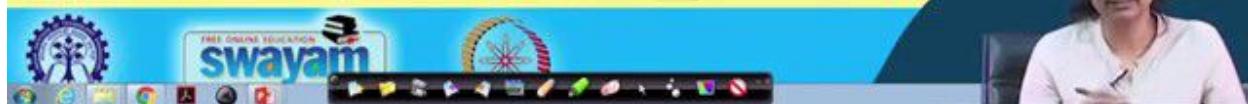


সুতরাং, এটি মাইক্রোফোন(microphone), আপনি যদি একটি একক মাইক্রোফোন(microphone) ব্যবহার করেন তবে আপনাকে এটি সংযোগের জন্য পূর্ববর্তী স্লাইড(slide) এ যেমন দেখিয়েছি তেমনি এটি সার্কিটের সাথে সংযুক্ত করতে হবে। তবে আপনি যদি এই নির্দিষ্ট মাইক্রোফোনটি ব্যবহার করেন তবে এটি পুরো মডিউল(module) হিসাবে ড্রাইভার সার্কিট(driver circuit) র সাথে আসে। এবং একটি Vcc, GND এবং একটি আউটপুট পিন সংযোগ রয়েছে। এটি 3.3 ভোল্ট(volt) থেকে 5 ভোল্টের(volt) পাওয়ার সাপ্লাইয়ের(power supply) সাথে কাজ করতে পারে এবং শব্দটি সনাক্ত হয়েছে কিনা তা নির্ভর করে ডিজিটাল(digital) আউটপুট(output) 0 বা 1 হয়। আমরা এটিকে সরাসরি মাইক্রোকন্ট্রোলারের সাথে সংযুক্ত করতে পারি।

(Refer Slide Time: 03:22)

Experiment 1

- Implement a clap switch, where a LED can be turned ON using the sound from a clap.
 - The microphone circuit is interfaced to the AnalogIn pin A1.
 - The AnalogIn object converts the input voltage to floating-point numbers from 0.0 to 1.0.
 - Through experimentation, we can determine the threshold value, which would depend on the noise of the environment.
 - The microphone module used, however, generates a 2-level digital output.
 - A LED is connected to the DigitalOut pin D2, where the LED glows if the D2 pin is at 0.
 - The LED turns on for 3 seconds, and then turns off.

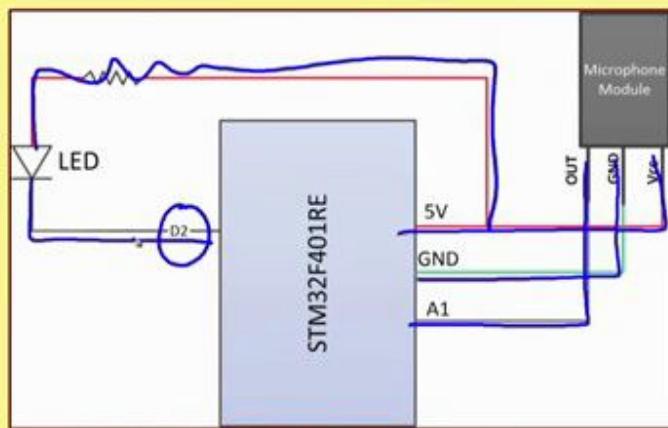


শীঘ্রই আমি পরীক্ষায় আসব কারণ এখানে সংযোগটি মোটামুটি সহজ। আমি একটি তালি সুইচ(switch) বাস্তবায়ন করব, যেখানে একটি তালি থেকে শব্দ ব্যবহার করে একটি LED চালু বা বন্ধ করা যায়। এখানে মাইক্রোফোন(microphone) সার্কিট(circuit)টি প্রথমে অ্যানালগ পিন A1 এর সাথে ইন্টারফেস(interface) করা হয়, এই AnalogIn অবজেক্ট(object)টি ইনপুট(input) ভোল্টেজ(voltage)কে 0 এবং 1 এর মধ্যে ভগ্নাংশ মানে রূপান্তর করে, আমরা ইতিমধ্যে এটি দেখেছি। এবং পরীক্ষার মাধ্যমে যখন আপনি তালি দিয়ে যে মানটি পাবেন তা দিয়ে আমরা এই প্রান্তিক মানটি নির্ধারণ করতে পারি। এটি অবশ্যই ব্যবহৃত মাইক্রোফোন মডিউল পরিবেশের নয়েজ(noise) এর উপর নির্ভর করবে।

একটি এলইডি(LED) ডিজিটাল(digital) আউট(out) পিন D2 এর সাথে সংযুক্ত থাকে, যেখানে D2 পিন 0 তে থাকলে LED আলোকিত হয়। যখনই এটি তালি দিয়ে শব্দ সনাক্ত করে, আমরা 3 সেকেন্ডের জন্য LED চালু করি এবং তারপরে এটি বন্ধ হয়ে যায়।

(Refer Slide Time: 04:51)

Connection Diagram – STM32



এটি সংযোগ ডায়াগ্রাম(giagram)। সুতরাং, এই OUT অ্যানালগ(analog) পিন A1 এর সাথে সংযুক্ত, এই GND গ্রাউন্ড(ground) এর এর সাথে সংযুক্ত এবং এটি 5 ভোল্টের সাথে সংযুক্ত। এবং এই এলইডি(LED)টির জন্য অ্যানোড(anode)টি প্রতিরোধের মাধ্যমে Vcc এর সাথে সংযুক্ত করা হয় এবং ক্যাথোড(cathode)টি পিন D2 এর সাথে সংযুক্ত থাকে।

(Refer Slide Time: 05:23)

Mbed program for Microphone

```
#include "mbed.h"
AnalogIn analog_value(A1);
DigitalOut led(D2);
int main() {
    float meas;
    led = 0
    while(1) {
        led = 1;
        meas = analog_value.read();
        if (meas >= 0.7) {
            led = 0;
            wait(3);
            led = 1;
        }
    }
}
```



এটি হ'ল প্রোগ্রাম(program) যা স্ব-ব্যাখ্যামূলক।

(Refer Slide Time: 06:55)

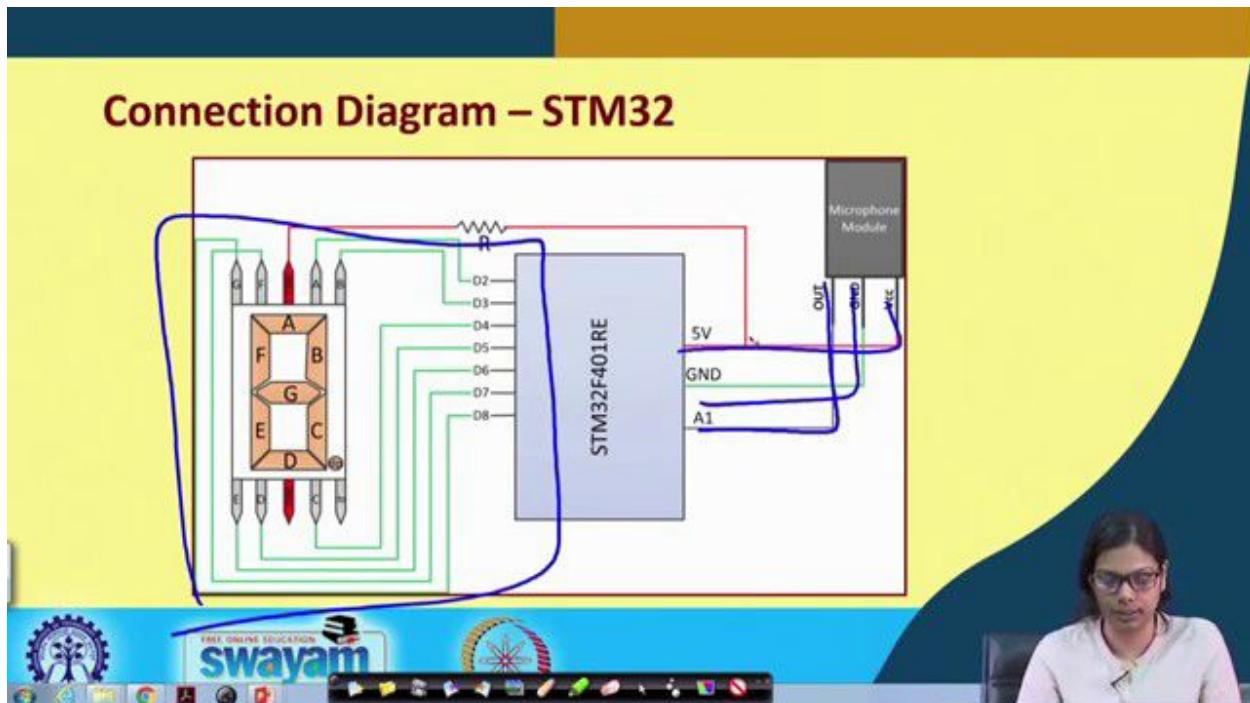
Experiment 2

- Interface a microphone and a 7-segment LED display unit to the microcontroller, and display the number of claps on the 7-segment display.



পৰেৱে পৱীক্ষায় আমৱা এই মাইক্ৰোকোনটি এবং একটি 7 সেগমেন্ট ডিসপ্লে(7 segment display) টি ইন্টাৱেক্ষেস কৱি এবং 7 সেগমেন্টে ডিসপ্লে(7 segment display) তে তালিৱ সংখ্যা প্ৰদৰ্শন কৱি। যথনই আমি একটি তালি দেব তখন 7 সেগমেন্ট ডিসপ্লে(7 segment display) এৱে মান বাড়বে এবং এটি যথন 9 এ পৌছবে তখন আবাৱ 0 তে ফিৱে আসবে।

(Refer Slide Time: 07:41)



এটি সাৰ্কিট ডায়াগ্ৰাম, যা সহজ সৱল।

(Refer Slide Time: 08:04)

Mbed program for Microphone (Clap Count)

```
#include "mbed.h"
DigitalOut A(D2);
DigitalOut B(D3);
DigitalOut C(D4);
DigitalOut D(D5);
DigitalOut E(D6);
DigitalOut F(D7);
DigitalOut G(D8);
AnalogIn val(A1);

void Display(int disp) {
    switch(disp)
    {
        case 0: A=0;B=0;C=0;D=0;E=0;F=0;G=1; break;
        case 1: A=1;B=0;C=0;D=1;E=1;F=1;G=1; break;
        case 2: A=0;B=0;C=1;D=0;E=0;F=1;G=0; break;
        case 3: A=0;B=0;C=0;D=0;E=1;F=1;G=0; break;
        case 4: A=1;B=0;C=0;D=1;E=1;F=0;G=0; break;
        case 5: A=0;B=1;C=0;D=0;E=1;F=0;G=0; break;
        case 6: A=0;B=1;C=0;D=0;E=0;F=0;G=0; break;
        case 7: A=0;B=0;C=0;D=1;E=1;F=1;G=1; break;
        case 8: A=0;B=0;C=0;D=0;E=0;F=0;G=0; break;
        case 9: A=0;B=0;C=0;D=0;E=1;F=0;G=0; break;
    }
}
```



The video shows a woman with glasses and a white shirt, speaking on a video call. The background is a blue and yellow wall. The video is from the Swayam platform, as indicated by the logo and interface elements at the bottom of the screen.

এখন, এই কোডটি সহজ সরল।

(Refer Slide Time: 08:30)

```
int num=0;
int main()
{
    float meas;
    while(1) {
        meas = val.read();
        if (meas >= 0.7) {
            num++;
            wait(0.1);
        }
        if (num > 9) {
            num=0;
        }
        Display(num);
    }
}
```



The video shows a woman with glasses and a white shirt, speaking on a video call. The background is a blue and yellow wall. The video is from the Swayam platform, as indicated by the logo and interface elements at the bottom of the screen.

এই সময়ে while(1) লুপটি, আমরা পোর্ট A1 এর মাধ্যমে পড়ি, মানটি যদি 0.7 এর সমান বা বেশি হয় তবে সংখ্যা বৃদ্ধি পায়। আমরা 0.1 সেকেন্ডের জন্য অপেক্ষা করি এবং তারপরে আমরা আবার চালিয়ে যাব।

সুতরাং, এখন আমরা মাইক্রোফোন ব্যবহার করে পরীক্ষাগুলি প্রদর্শন করব।

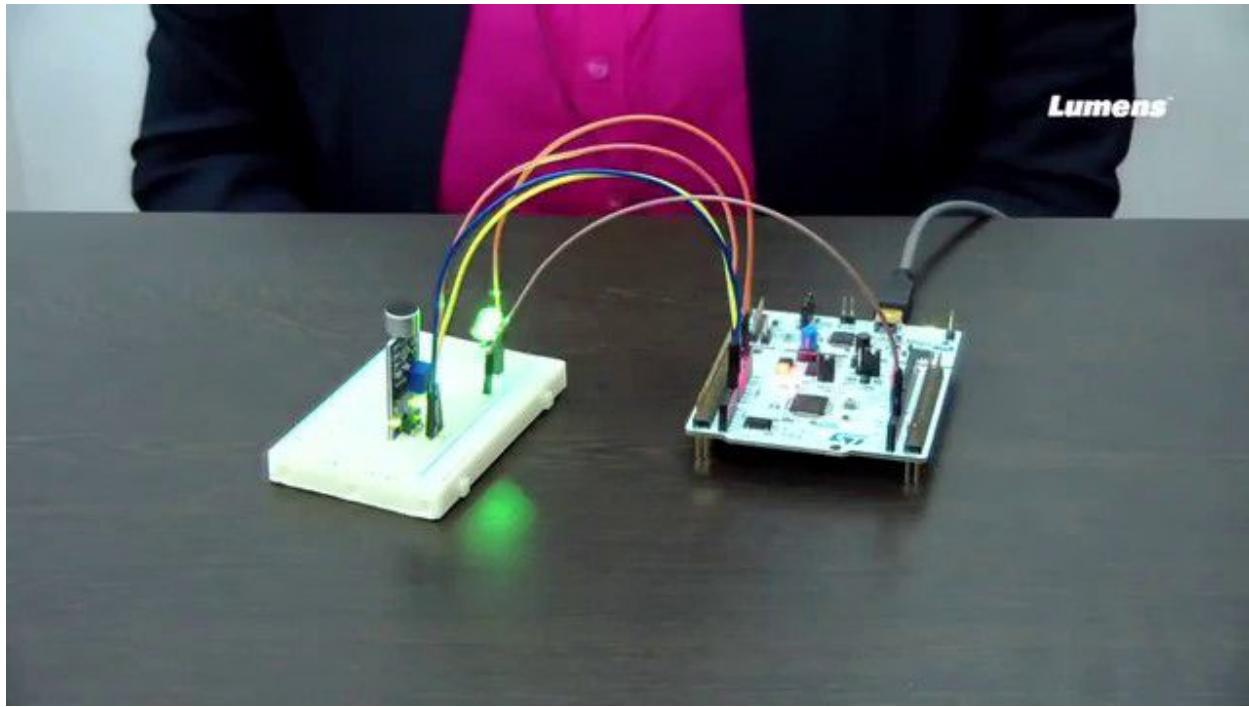
আমি এখন আপনাকে এই মাইক্রোফোনটি ব্যবহার করে এবং এসটিএম(STM) বোর্ড সহ পরীক্ষাগুলি দেখাব।

(Refer Slide Time: 11:39)



এটি মাইক্রোফোন। এটি Vcc, GND এবং OUT তিনটি লাইন পেয়েছে। আমরা সংযোগগুলি তৈরি করি।

(Refer Slide Time: 12:52)



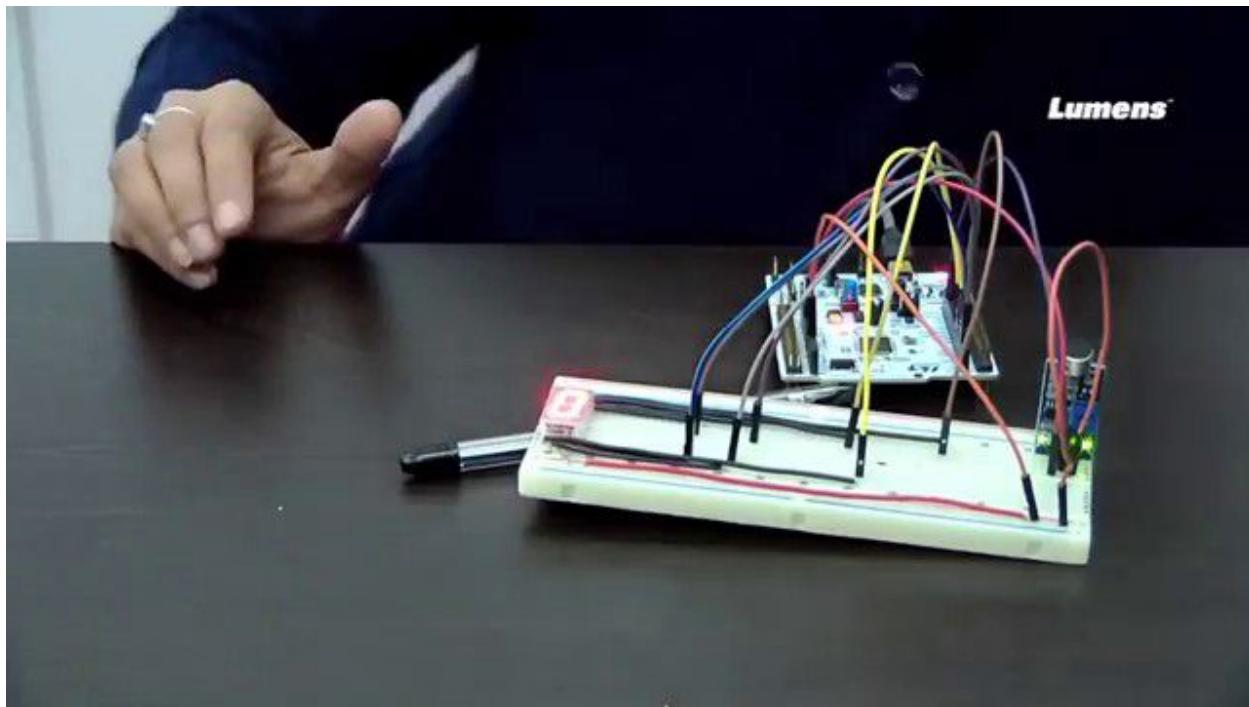
আমাকে এই এলইডি(LED) টি সংযোগ করতে হবে। আমি অ্যানোড(anode) টি 3.3 ভোল্টের সাথে সংযুক্ত করছি এবং ক্যাথোড(cathode) টি পোর্ট D2 এর সাথে সংযুক্ত করছি। মাইক্রোফোনে তিনটি পিন হল VCC, GND এবং OUT1 | OUT অ্যানালগ পোর্ট A2 এর সাথে সংযুক্ত।

আমি এখন এই বোর্ডে কোডটি ফেলবো।

আমি তালি দেব। আপনি দেখুন যখন আমি তালি দেব এলইডি(LED) টি 3 সেকেন্ডের জন্য জ্বলিবল করে। আমি আরও একবার এটা করব। পরীক্ষাটি খুব সহজ সরল আমাদের কেবল এটি নিশ্চিত করতে হবে যে সংযোগগুলি ঠিক আছে এবং তারপরে শব্দের শব্দের উপর উপর নির্ভর করে LED জ্বলে উঠবে। তাহলে, এই ছোট জিনিসটির প্রয়োগ কোথায়? আপনি এমন কোনও নোটিশ বোর্ডের কথা ভাবতে পারেন যেখানে রাত্রিকালে আপনাকে আলো দিতে হবে না। কেউ যখন নোটিশ বোর্ডের কাছে আসে আপনি আসলে একটি তালি দেবেন এবং সেই সাথে নোটিশ বোর্ডের অভ্যন্তরে একটি আলো জ্বলতে থাকবে।

মাইক্রোফোন ব্যবহার করে পূর্ববর্তী পরীক্ষার ধারাবাহিকতায়, আমি আপনাকে আরও একবার পরীক্ষা-নিরীক্ষা করে দেখাব যেখানে আমি ইতিমধ্যে 7 সেগমেন্ট ডিসপ্লে(7 segment display) এর জন্য সমস্ত সংযোগ তৈরি করেছি। Let us see here. আমাদের এখানে দেখতে দিন।

(Refer Slide Time: 19:11)



যখনই আমি তালি দেব তখন তালিগুলির সংখ্যা এই 7 সেগমেন্ট ডিসপ্লে(7 segment display) তে প্রদর্শিত হবে এবং যখন তালি আবার 9 এ পৌঁছবে তখন পুনরায় তা 0 তে সেট হবে।

ধন্যবাদ।

Embedded System Design with ARM

Prof. Indranil Sengupta

Department of Computer Science and Engineering

Indian Institute of Technology, Kharagpur

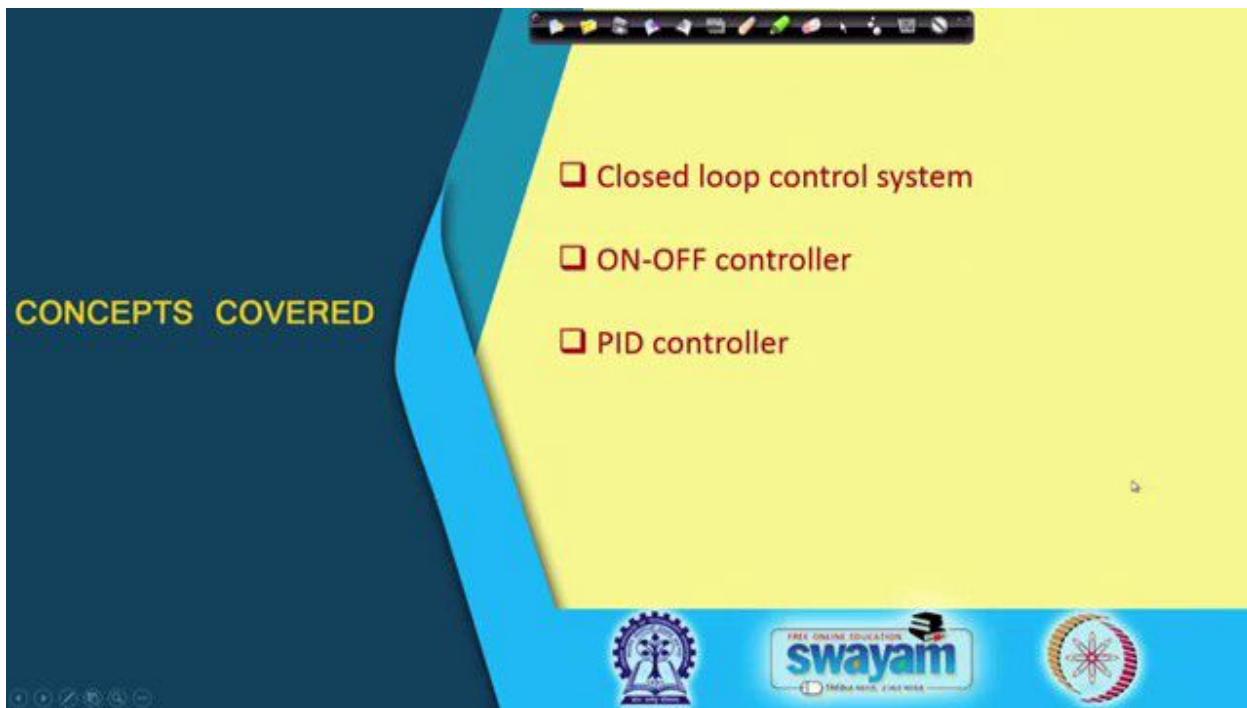
Lecture – 30

Design of Control System

আপনি বিভিন্ন প্রদর্শিত পরীক্ষাগুলি দেখেছেন যা এখনও অবধি আমরা আলোচনা করেছি। উদাহরণস্বরূপ, আপনি একটি আলো চালু বা বন্ধ করতে পারেন, আপনি একটি এলইডি(LED) জ্বালাতে পারেন, আপনি অনেক কিছুর উপর নির্ভর করে এটি বন্ধ করতে পারেন। এখানে অনেক ধরণের সেন্সর(sensor) রয়েছে, অ্যাকচুয়েটর(actuator) ব্যবহার করে আপনি বাইরে থেকে কিছু প্যারামিটার(parameter) পড়তে পারেন এবং আউটপুট ডিভাইস(output device) ব্যবহার করে আপনি কিছু তথ্য বাইরে আউটপুট হিসাবে দিতে পারেন।

এখন এই বক্তৃতায়, আমরা কন্ট্রোল সিস্টেম(Control System) নামক কিছু সম্পর্কে সবিস্তারে বলব। আমরা কন্ট্রোল সিস্টেম(Control System)সম্পর্কে প্রাথমিক ধারণা এবং এটি এম্বেডেড সিস্টেম(embedded system) ডিজাইন করতে কেন গুরুত্বপূর্ণ তা বিবেচনা করব।

(Refer Slide Time: 01:15)



এই বক্তৃতায় আমরা বিশেষত ক্লোজড-লুপ কন্ট্রোল সিস্টেম(closed-loop control system), এবং ওএন-অফ কন্ট্রোল(ON-OFF control) এবং PID কন্ট্রোল এর মতো বিভিন্ন ধরণের কন্ট্রোল সিস্টেম(control system) এর কথা বলব।

আমি এখানে আপনাকে বলি যে ধরন আমি একটি মাইক্রোকন্ট্রোলার(microcontroller) দিয়ে একটি হিটার(heater) কে ইন্টারফেস(interface) করছি। আমাকে গরম লাগলে এটি বন্ধ করে দিচ্ছি, যদি আমার আবার ঠাণ্ডা লাগে তবে আমি এটি চালু করি। তবে, ধরন আমি এই প্রক্রিয়াটি স্বয়ংক্রিয় করতে চাই, আমি আমার সিস্টেমটি এমনভাবে ডিজাইন করতে চাই যে যখনই তাপমাত্রা একটি নির্দিষ্ট স্তরের নীচে নেমে আসে, ধরন 25 ডিগ্রি সেন্টিগ্রেড, আমার হিটারটি চালু হবে, যদি এটি তার উপরে থাকে তবে বন্ধ হবে।

তবে, সেই প্রক্রিয়াটির জন্য সেখানে একটি প্রতিক্রিয়া থাকতে হবে। আমি কেবল আমার হিটারকে নিয়ন্ত্রণ করতে সক্ষম হওয়াই যথেষ্ট নয়, আমার বর্তমান তাপমাত্রার মানও পড়তে সক্ষম হওয়া উচিত। আমার তাপমাত্রা কম বা উচ্চতর কিনা তা স্বয়ংক্রিয়ভাবে আমার জানা উচিত, এটি চিত্রের মধ্যে আসা একটি লুপ কন্ট্রোল(loop control) সিস্টেমের ধারণা।

(Refer Slide Time: 02:41)

Introduction

- In many embedded system applications, we have to sense the value of some external parameter and take corrective actions to maintain it within acceptable limits.
 - Temperature of an oven, speed of a motor, etc.
 - Essentially a control system.
- Two types of control systems:
 - a) Open loop: where there is no feedback with respect to the measured value.
 - b) Closed loop: more sophisticated, corrective actions applied with feedback.

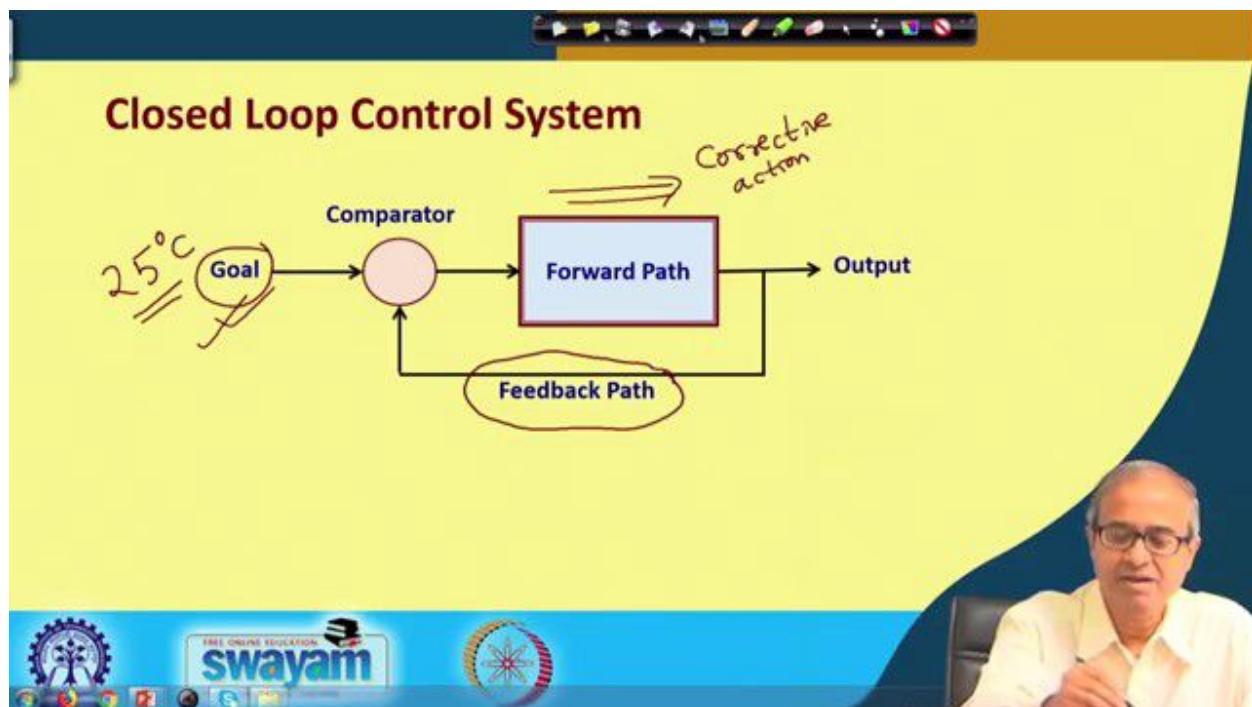
আমরা যা বলছি তা হ'ল আমি এখনই যেটা সম্পর্কে বললাম সেরকম অনেক এমবেডেড সিস্টেম অ্যাপ্লিকেশন(embedded system application) রয়েছে। উদাহরণস্বরূপ, আমি প্যারামিটার(parameter) গুলি তাপমাত্রা, আর্দ্রতা হিসাবে গ্রহণ করেছি, এটি যে কোনও কিছু হতে পারে। এখন আপনি যে মান পেয়েছেন তার উপর নির্ভর করে আপনি একটি সংশোধনমূলক পদক্ষেপ নিতে পারেন। কেন? এটি গ্রহণযোগ্য সীমাতে বজায় রাখতে।

যেমন বলা যাক, তাপমাত্রার জন্য আপনি বলতে পারেন যে আমার গ্রহণযোগ্য সীমা 25 থেকে 27 ডিগ্রি সেলসিয়াস। যদি আপনি দেখতে পান যে তাপমাত্রা এর নিচে নেমে যাচ্ছে, আপনার হিটারটি চালু করা উচিত। যদি আপনি দেখতে পান যে তাপমাত্রা এর উপরে যাচ্ছে, যদি শীতাতপ নিয়ন্ত্রণ ব্যবস্থা থাকে তবে আপনার এটি চালু করা উচিত। আমি এটি সেন্সিং এবং প্রতিক্রিয়া ধারণা সম্পর্কে বলছি। আপনি ভাবতে পারেন এমন অনেকগুলি অ্যাপ্লিকেশন(application) থাকতে পারে; ঘরের বা একটি ওভেনের তাপমাত্রা, মোটরের গতি ইত্যাদি।

এগুলি কন্ট্রোল সিস্টেম(control system) এর উদাহরণ; বিস্তৃতভাবে বলতে গেলে কন্ট্রোল সিস্টেম(control system) গুলি ওপেন লুপ(open loop) বা ক্লোজড লুপ(closed loop) হতে পারে। ওপেন লুপ মানে কোনও প্রতিক্রিয়া(feedback) প্রক্রিয়া নেই; আপনি কেবল হিটার চালু বা বন্ধ করে দিচ্ছেন, তবে

আপনি তাপমাত্রার মান পড়ছেন না। ক্লোজড লুপ(closed loop)টির অর্থ আপনি কেবল নিয়ন্ত্রণ করছেন তা নয়, একটি প্রতিক্রিয়া(feedback) প্রক্রিয়াও রয়েছে। স্বাভাবিকভাবেই, এটি আরও পরিশীলিত(sophisticated) কারণ প্রতিক্রিয়ার ভিত্তিতে আপনি কিছু সংশোধনকারী সংকেত(corrective signal) পাঠাতে পারেন। আমাদের এখানে বলা যাক, এই সীমার মধ্যে তাপমাত্রা বজায় রাখতে আপনাকে সেই অনুযায়ী হিটার নিয়ন্ত্রণ পাঠাতে হবে। এটি হল প্রাথমিক ধারণা।

(Refer Slide Time: 05:09)

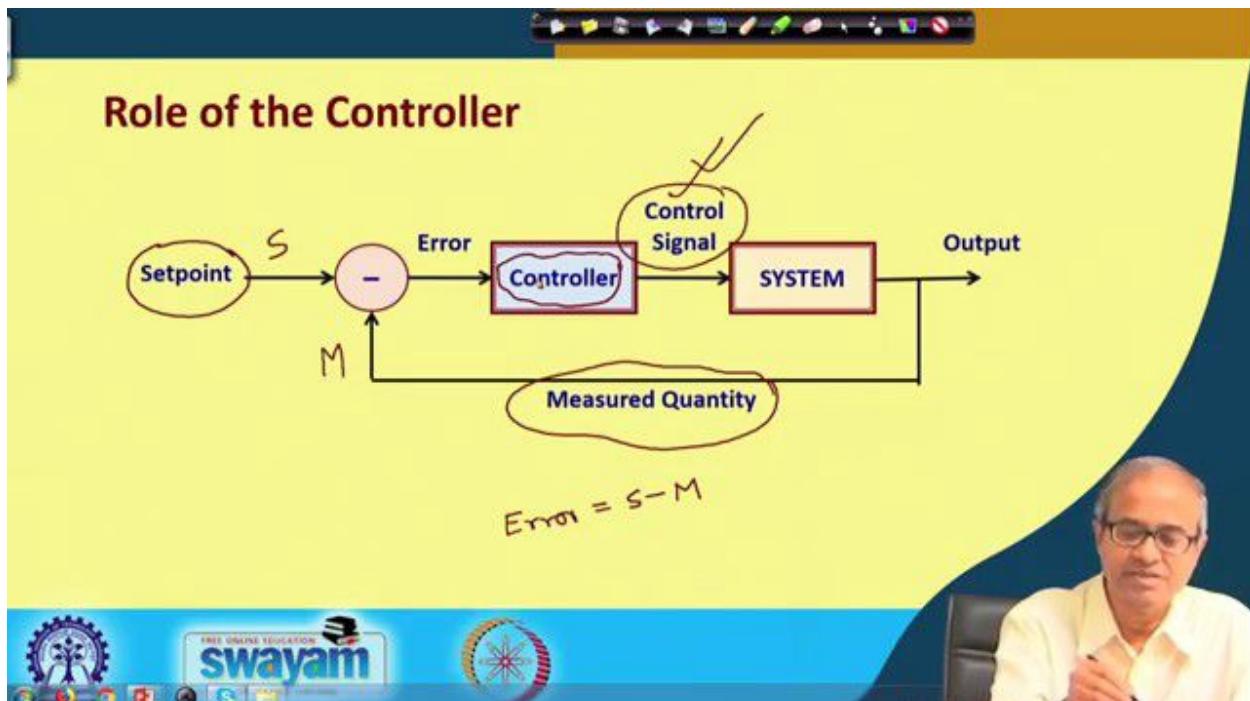


আসুন আমরা একটি সাধারণ ক্লোজড লুপ(closed loop) কন্ট্রোল সিস্টেমের একটি চিত্র দেখি। এই চিত্রটিতে কী রয়েছে? এখানে আমরা একটি লক্ষ্য দিয়ে শুরু করি, লক্ষ্যটির অর্থ আমরা কোনও নির্দিষ্ট স্তরে প্যারামিটার(parameter) বজায় রাখতে চাই; তাপমাত্রার জন্য আমরা বলি, এটি 25 ডিগ্রি সেলসিয়াস।

এবং আউটপুট থেকে আমরা বলতে পারি যে আমরা যে ঘরে বসে আছি, সেখানে একটি প্রতিক্রিয়া পথ থাকবে; তার মানে, কিছু সেন্সর থাকবে, যা তাপমাত্রাটি পড়বে এবং এটি কিছু প্রতিক্রিয়ার মান ফেরত পাঠাবে।

একটি তুলনাকারী বা কম্পারেটর(comparator) থাকবে, যা কিনা এই প্রতিক্রিয়াটি যা আপনি যা পড়ছেন তা নির্ধারিত লক্ষ্যের চেয়ে কম বা বেশি কিনা তা যাচাই করবে। তার উপর নির্ভর করে আপনাকে সামনের পথ ধরে কিছু প্রকার সংশোধনমূলক পদক্ষেপ পাঠাতে হবে যাতে আউটপুট প্যারামিটারের মানের নির্ধারিত লক্ষ্যের যতটা সম্ভব কাছাকাছি হয়। এটি হল ক্লোজড লুপ কন্ট্রোল সিস্টেম (closed loop control system) থাকার সামগ্রিক উদ্দেশ্য।

(Refer Slide Time: 06:49)

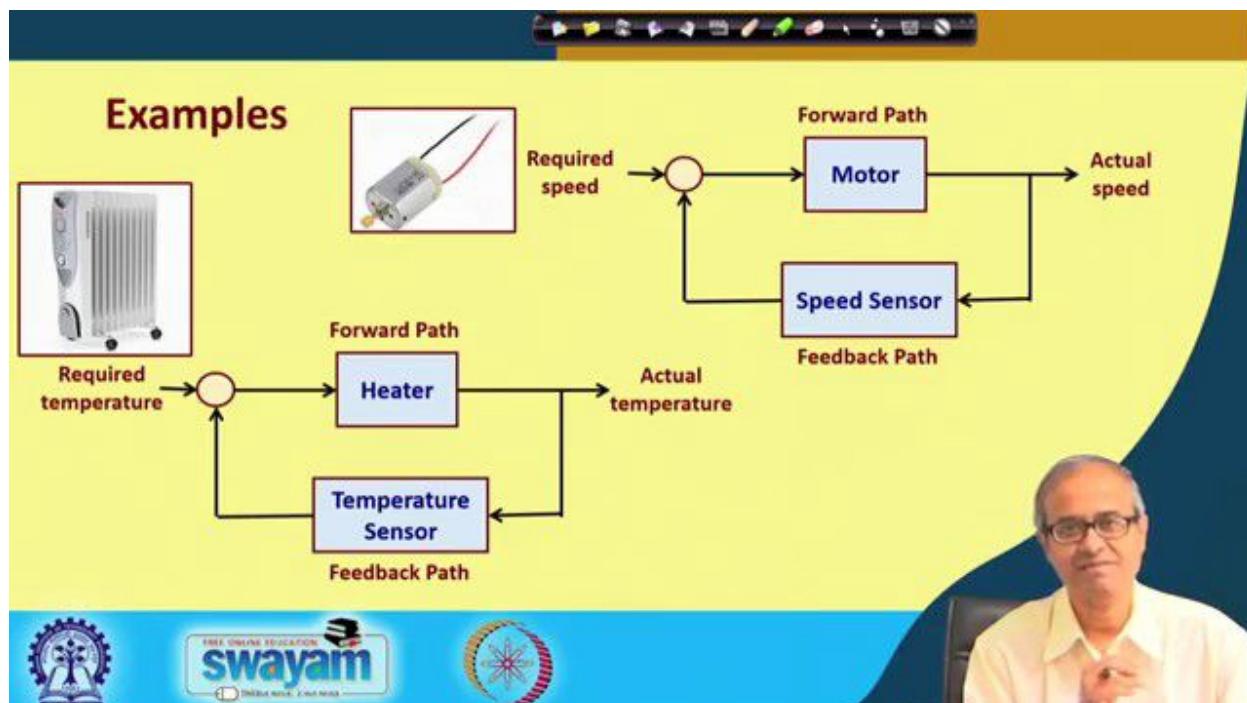


আসুন চিত্রটি আরও কিছুটা প্রসারিত করা যাক। আসুন আমরা এখানে লক্ষ্যটি নিয়ে কথা বলি, আমরা এটিকে সেট পয়েন্ট বলছি ... একই জিনিসকে আমরা একটি স্থর নির্ধারণ করছি, যেখানে ফিজিক্যাল(physical) প্যারামিটারের মান বজায় রাখতে আমরা কোনও সেন্সরের মাধ্যমে পাওয়া আউটপুটটি পরিমাপ করছি।

আসুন আমি বলতে পারি যে আমার সেট পয়েন্টটি S, আমার পরিমাপ করা মানটি M, ত্রুটি(error) হবে S-M। যদি তারা সমান হয় তবে এটি ঠিক আছে, কারণ আমার আউটপুট ঠিক বিন্দুতে রয়েছে। তবে ত্রুটি(error)র উপর নির্ভর করে এটির চেয়ে কম বা বেশি হলে এখানে একটি কন্ট্রোলার থাকবে, যা সিদ্ধান্ত নেবে যে কীভাবে আমার সিস্টেমের জন্য একটি কন্ট্রোল সিগন্যাল(control signal) বা নিয়ন্ত্রণ সংকেত তৈরি করা যায়।

সুতরাং, এখানে আমার সিস্টেমটি আমার হিটার বা এসি মেশিন বা যা কিছু নিয়ন্ত্রণ করার চেষ্টা করছি। এটি একটি মাইক্রোকন্ট্রোলার যা নিয়ন্ত্রণকারী প্রক্রিয়া হিসাবে কাজ করবে যা একটি বুদ্ধিমান সিদ্ধান্ত নেবে, কীভাবে কন্ট্রোল সিগন্যাল(control signal) বা নিয়ন্ত্রণ সংকেত তৈরি করা যায় যাতে আমার আউটপুট আমার সেট পয়েন্টের যতটা সম্ভব কাছাকাছি হয়। এখানেই নিয়ামক বা কন্ট্রোলার(controller) এর ভূমিকা আসে। কন্ট্রোলারের বিভিন্ন বিকল্প নকশা থাকতে পারে আমরা ভাবতে পারি।

(Refer Slide Time: 08:35)



প্রথমে কিছু উদাহরণ দেখি। এটি রুম হিটারের খুব সাধারণ উদাহরণ।

মনে করুন আমার কাছে একটি রুম হিটার রয়েছে যেখানে আমি প্রয়োজনীয় তাপমাত্রা নির্ধারণ করেছি। এখানে আমি ধরে নিছি যে আমি এমন জায়গায় আছি যা বেশ শীতল, এর অর্থ আপনি যদি এমন কোনও জায়গায় থাকেন যা গরম থাকে তবে আপনি এই উদাহরণটির প্রশংসা করতে পারবেন না, ধরুন যে আমি খুব শীতকাল এমন একটি জায়গায় অবস্থান করছি।

সুতরাং, আমার একটি রুম হিটার দরকার। প্রতিক্রিয়া পথে একটি তাপমাত্রা সেন্সর থাকবে। প্রকৃত তাপমাত্রা সংবেদন করা হবে, তুলনাকারী(comparator) একটি পার্থক্য খুঁজে পাবে এবং সামনের পথে কন্ট্রোলার একটি হিটার নিয়ন্ত্রণ করবে। অন্য একটি উদাহরণ নিন। ধরুন আমি কোনও DC মোটরের গতি নিয়ন্ত্রণ করতে

চাই। কিছু প্রয়োজনীয় গতি নির্দিষ্ট করা আছে, ধৰন প্রতি মিনিটে 100 পাক(revolution) বা 100 আরপিএম(RPM)।

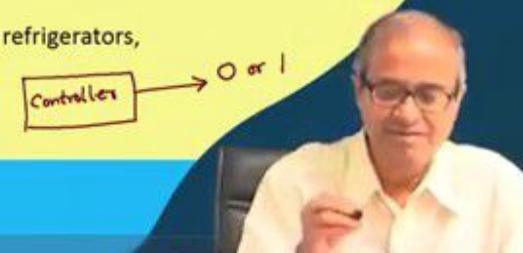
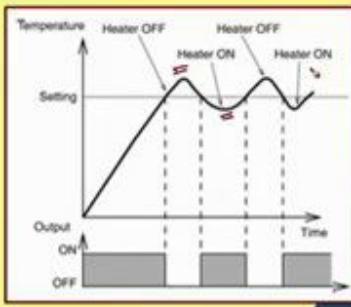
প্রতিক্রিয়া পথে একটি স্পিড সেন্সর(speed sensor) থাকবে, এটি কোনভাবে গতিটি সংবেদন করবে। পার্থক্যের উপর নির্ভর করে কন্ট্রোলার মোটর টিকে চালিত করার জন্য বিদ্যুৎ সরবরাহ সক্রিয় করবে, যাতে গতি বজায় থাকে।

(Refer Slide Time: 10:03)

Choice of Controller Type

a) ON-OFF Controller:

- An ON-OFF controller is the simplest type of controller, where the control signal has only two levels.
 - For example, in a heater, if the actual temperature is less than the set temperature, the heater is turned ON; otherwise, it is turned OFF.
 - This type of controller is inexpensive, but often causes oscillation in the output variable.
 - It is often used in simple appliances such as oven, iron, refrigerators, etc. where oscillations can be tolerated.



কন্ট্রোলার(controller) বা নিয়ন্ত্রকের ভূমিকার কথা বললে বিভিন্ন ধরণের কন্ট্রোল সিস্টেম(control system) থাকতে পারে। আসুন অন-অফ(ON-OFF) কন্ট্রোলার(controller) নামে পরিচিত এই সাধারণ ধরণের কন্ট্রোলার(controller) বা নিয়ন্ত্রক দিয়ে শুরু করা যাক। অন-অফ(ON-OFF) কন্ট্রোলার মানে পরিমাপ করা মানটি পূর্ব নির্ধারণের চেয়ে কম বা বেশি কিনা তার উপর নির্ভর করে আমরা হয় ডিভাইস(device) টি চালু বা বন্ধ করব।

যেমনটি আমি বলেছি এটি সাধারণতম নিয়ন্ত্রক(controller), যেখানে নিয়ন্ত্রণ সংকেত(control system) টির মাত্র 2 টি স্থান থাকে, যদি আপনি বলেন যে এখানে আমার কাছে কন্ট্রোলার রয়েছে। সুতরাং, কন্ট্রোলার একটি একক আউটপুট(output) উত্পন্ন করবে, যা 0 বা 1, 0 মানে বন্ধ এবং 1 মানে চালু। তবে

এটি আউটপুট(output) ভেরিয়েবল(variable) এ দোলন(oscillation) সৃষ্টি করতে পারে। কীভাবে? আপনি এই চিত্রটি ডানদিকে দেখতে পাচ্ছেন। এখানে, আমি যে লেখচিত্র দেখছি তা হ'ল তাপমাত্রা বনাম সময় লেখচিত্র। ধরুন আমি যে ঘরে বসে আছি, আমি তাপমাত্রাটি পরিমাপ করছি এবং আমার একটি নির্দিষ্ট তাপমাত্রা রয়েছে, মনে করুন 27 ডিগ্রি সেলসিয়াস।

যখন তাপমাত্রা 27 এর কম হয়, এটি আমার কন্ট্রোল আউটপুট(control output); হিটার চালু আছে। সুতরাং, তাপমাত্রা আস্তে আস্তে উপরে উঠবে, এখন এই সেট স্তরটি অতিক্রম করার সাথে সাথেই আমি হিটারটি বন্ধ করে দিই। এখন আপনি দেখুন আমরা হিটারটি বন্ধ করার সাথে সাথেই তাপমাত্রা হঠাতে হ্রাস করতে শুরু করে না কারণ হিটারটি ইতিমধ্যে অব্লজ্বল করছে। সুতরাং, তাপমাত্রা কিছু সময়ের জন্য এখনও বাড়বে তবে এরপরে এটি আরও বৃদ্ধি পাওয়া বন্ধ করবে।

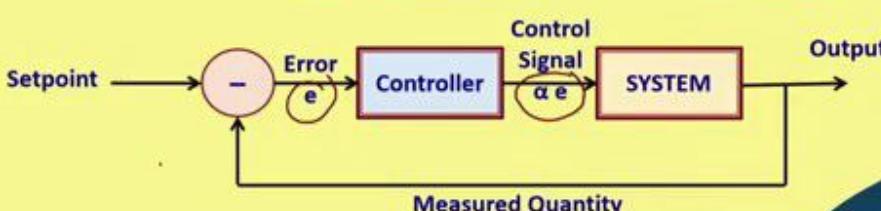
সুতরাং, এখানে এর মতো একটি ওভারশট(overshoot) হবে এবং এরপরে আপনি যখন হিটারটি বন্ধ করবেন তখন তাপমাত্রা আবার ধীরে ধীরে কমতে শুরু করবে। পরে, আপনি আবার হিটারটি চালু করুন, আবার হিটারটির কুণ্ডলীটি গরম হতে কিছুটা সময় নেবে। সুতরাং, এটি আবার নীচে চলে যাবে এবং আবার এই প্রক্রিয়া পুনরাবৃত্তি হবে। সুতরাং, আপনি দেখতে পাবেন যে এখানে উপরে-নীচে একটি নিরন্তর দোলন(oscillation) থাকবে, উপর নীচে, উপর নীচে এটি চলতে থাকবে, এমন নয় যে এটি একটি নির্দিষ্ট স্তরে খুব স্থিতিশীল থাকবে, এটি 26, 28, 26, 28 এরকম কিছু ঘটবে।

রুম হিটিংয়ের(room heating) মতো অ্যাপ্লিকেশনগুলির জন্য এই ধরণের একটি কন্ট্রোলার(controller) বা নিয়ামক বেশ ভাল, যেখানে ছোট ডিগ্রি পার্থক্য এ সমস্যা করে না। তবে, কিছু অ্যাপ্লিকেশন রয়েছে যেখানে আপনাকে প্যারামিটারটি খুব কাছাকাছি নিয়ন্ত্রণ করতে হয়, সেখানে আপনি এই জাতীয় সরল নিয়ামক(simple controller) ব্যবহার করতে পারবেন না।

(Refer Slide Time: 13:19)

b) Proportional (P) Controller:

- The control signal is set to be *proportional to the difference* between the actual output and the setpoint (i.e. the error).
- Need to find out the value of constant of proportionality.
 - Tuning the controller is a hard job.
- Typically, a P controller *decreases response time*, but increases *overshoot*.



FREE ONLINE REGULATION
swayam



আপনার কাছে এমন কিছু থাকতে হবে যা কিছুটা পরিশীলিত(sophisticated)। আমরা সমানুপাতিক নিয়মক(proportional controller) বা P কন্ট্রোলার(P controller) নামক কিছু ব্যবহার করতে পারি।

এটি কি করে? এখানে আমার কন্ট্রোলার ডিজিটাল সিগন্যাল(digital signal) তৈরি করছে না, তবে একটি অ্যানালগ(analog) সিগন্যাল তৈরি করছে যা আমার ত্রুটি(error)র সাথে সমানুপাতিক(proportional)। আমি যদি আমার ঘরের তাপমাত্রা দেখি এবং আমার সেট করা তাপমাত্রা একেবারে আলাদা হয়, আমি হিটারকে একটি উচ্চ ভোল্টেজ প্রেরণ করেছি। যদি আমি দেখি যে আমার পার্থক্যটি খুব ছোট, আমি হিটারকে একটি কম ভোল্টেজ প্রেরণ করেছি যাতে হিটারটি হঠাৎ খুব গরম না হয় এবং সেখানে একটি ওভারশট(overshoot) হয়, তবে পার্থক্যটি খুব বড় হলে আমি একটি বড় ভোল্টেজ প্রেরণ করেছি। সুতরাং, হিটারটি দ্রুত উত্তপ্ত দেয় যাতে এটি খুব দ্রুত সেট মানটি(set value) অর্জন করে।

আবার সেই একই চিত্রটি দেখছি, যদি আপনার ত্রুটি(error) e হয় আপনার কন্ট্রোল সিগন্যাল(control signal) e এর সমানুপাতিক হবে। সুতরাং, যদি $e = 0$ হয়, তবে আপনি কোনও সংশোধনকারী সিগন্যাল(corrective signal) প্রেরণ করছেন না। স্তরটি সেখানে যাই হোক না আপনি সেটি ছেড়ে দিন, এটি এইভাবে কাজ করে।

এখন অন-অফ(on-off) কন্ট্রোলারের তুলনায় আপনি এই সমানুপাতিক কন্ট্রোলারের থেকে কী কী সুবিধা পাচ্ছেন? ক্ষটি(error) বড় হলে আপনি একটি বৃহত্তর সংশোধনকারী সংকেত প্রেরণ করুন যাতে ক্ষটি(error) খুব দ্রুত হ্রাস করা যায়। সুতরাং প্রতিক্রিয়া সময় হ্রাস পেয়েছে, তবে ওভারশট(overshoot) এখনও যেমন রয়েছে তেমন উপরে যেতে পারে এবং এটি আবারও নামতে পারে, এই ধরণের ওভারশটিং এখনও থাকবে। কারণ, দেখুন আপনি এখনও একটি নিয়ন্ত্রণ সংকেত প্রেরণ করছেন যা প্যারামিটারটির মান এক উপায়ে পরিবর্তন করবে এবং আপনি তাত্ক্ষণিকভাবে এটি বন্ধ করতে পারবেন না, এটির জন্য কিছুটা সময় লাগবে এবং কিছুটা ওভারশট হবে।

(Refer Slide Time: 15:53)

c) **Proportional-Derivative (PD) Controller:**

- To reduce the overshoot, we can take into account how fast we are approaching the setpoint.
 - We add D control in addition to P control.
- D is estimated as the difference between the current measure and the previous measure.
- PD controllers are slower than P controllers, but generates less oscillation, and smaller overshoot/ripple.
- Drawback:**
 - Output is close to the setpoint, and so the error is very small.
 - Errors add up over time; we can define the integral (I) of the error:
$$\Sigma_{\text{time}} (\text{setpoint} - \text{output})$$

সুতরাং, আপনি পরবর্তী পদক্ষেপে যা করবেন তা হ'ল আপনি এটিতে আরও একটি ফ্লেবর(falvor) যুক্ত করুন, একটি ডেরিভেটিভ(derivative) ধারণা(concept) সমানুপাতিক এবং ডেরিভেটিভ কন্ট্রোলার(derivative controller) যুক্ত করুন, কারণ আমি উল্লেখ করেছি যে সমানুপাতিক নিয়মক এ ওভারশট করার সমস্যা ছিল।

ডেরাইভেটিভ(derivative) অংশেও আপনি লক্ষ্য রাখছেন, আপনি কতটা দ্রুত বিন্দুটির দিকে এগিয়ে চলেছেন এই বিষয়টিও লক্ষ্য রাখছেন। ডেরাইভেটিভ(derivative) এর অর্থ পার্থক্য; পরিমাপক ভেরিয়েবল(measured variable) এর সর্বশেষ মান বিয়োগ বর্তমান মান। সুতরাং,

ডেরাইভিটিভ(derivative) বর্তমান পরিমাপ এবং পূর্ববর্তী পরিমাপের মধ্যে পার্থক্যের একটি পরিমাপ ছাড়া কিছুই নয়, আপনি ক্রমাগত আউটপুট ভেরিয়েবল(output variable) টি অনুধাবন(sense) করুন, আপনি দেখবেন যে গতবারের তাপমাত্রাটি 21 ছিল এখন এটি 23 হল। সুতরাং, এটি 2 ডিগ্রি বৃদ্ধি পেয়েছে; তার মানে, খুব দ্রুত বাড়ছে। সুতরাং, আপনার এমন হারে হ্রাস করা উচিত যাতে ওভারশট(overshoot) না হয়।

এখানে আপনি যে নিয়ন্ত্রণ সংকেত(control signal) তৈরি করছেন তা ক্ষতি(error)র সাথে সমানুপাতিক হবে এবং এটি পার্থক্যের সমানুপাতিকও হবে; এর অর্থ, আপনি পূর্ববর্তী মান থেকে বর্তমান মানের বিয়োগ পরিমাপ করেছেন, এটি আপনার ডেরাইভিটিভ(derivative)। সুতরাং, এটি অবশ্যই এই উভয়ের সাথেই সমানুপাতিক হবে, সমানুপাতিকতা ধ্রুবক(proportionality constant) দুটি ক্ষেত্রে পৃথক হতে পারে কারণ, আপনি চাইলে কমবেশি করতে পারেন যাতে আপনি অনুপাতকে আরও বেশি গুরুত্ব দেবেন বা ডেরাইভিটিভ(derivative)কে আরও বেশি গুরুত্ব দেবেন।

ওভারশট(overshoot) হ্রাস করার ক্ষেত্রে PD কন্ট্রোলার আরও ভাল কাজ করে কারণ আপনি যেমন লক্ষ্যটিতে দ্রুত এগিয়ে যাচ্ছেন, আপনি কন্ট্রোল সিগনাল(control signal) কে হ্রাস করুন যাতে ওভারশট কম হয়।

Refer Slide Time: 18:37)

c) Proportional-Derivative (PD) Controller:

- To reduce the overshoot, we can take into account how fast we are approaching the setpoint.
 - We add D control in addition to P control.
- D is estimated as the **difference** between the current measure and the previous measure.
- PD controllers are slower than P controllers, but generates less oscillation, and smaller overshoot/ripple.
- **Drawback:**
 - Output is close to the setpoint, and so the error is very small.
 - Errors add up over time; we can define the integral (I) of the error:
$$\sum_{\text{time}} (\text{setpoint} - \text{output})$$

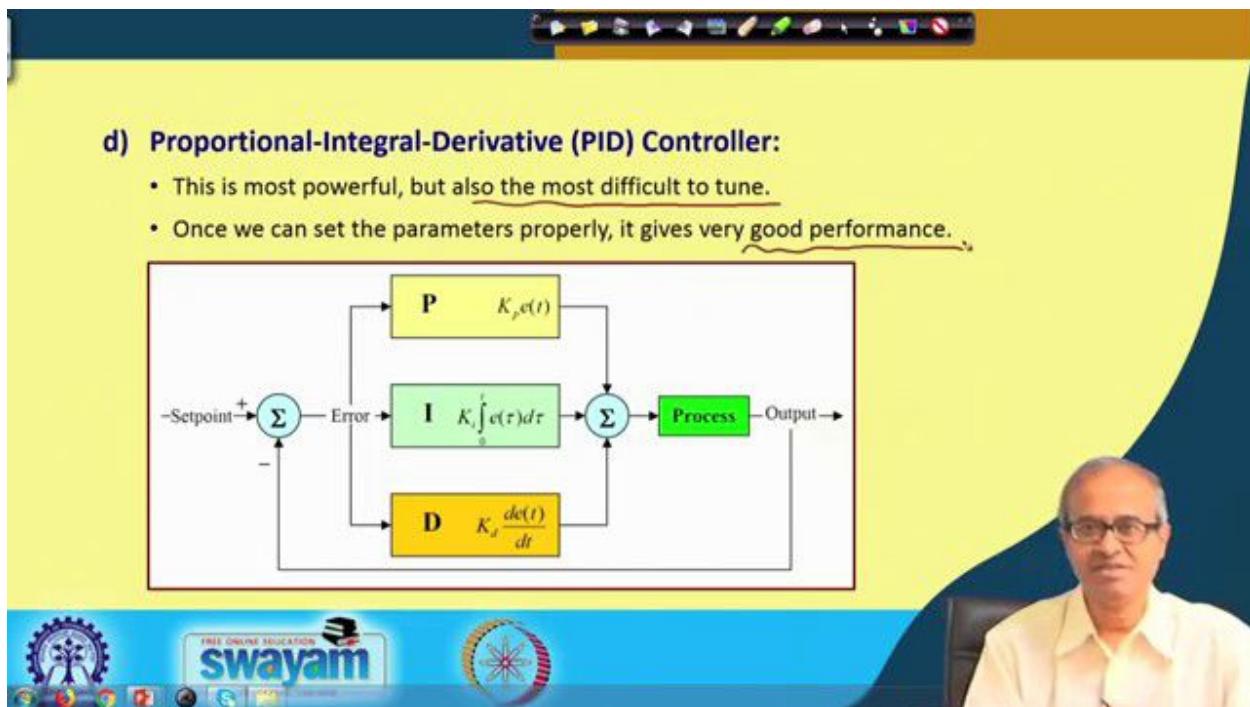
$$I = \frac{e_1 + e_2 + e_3 + \dots}{\text{Time}}$$

তবে একটা জিনিস হল; এই PD কন্ট্রোলারগুলি P কন্ট্রোলারের চেয়ে ধীর(slow), কারণ আপনি ইচ্ছাকৃতভাবে ক্রপান্তর হার কমিয়ে দিচ্ছেন। এটি যদি কাছাকাছি হয় তবে আপনি এটিকে আরও ধীর করুন এবং এজন্য এটি কম হয়, তবে দোলন(oscillation) কম হবে এবং ওভারশটও কম হবে। তবে একটি অসুবিধা হ'ল PD কন্ট্রোলে আউটপুটটি সেট পয়েন্টের কাছাকাছি হয়ে যায়; এর অর্থ, ক্রটি(error)টি ছোট, তবে ক্রটি(error)গুলি সময়ের সাথে সাথে জমা হতে থাকে। আমরা যদি অল্প পরিমাণে ক্রটির জন্য অনুমতি দিই তবে এই ক্রটিগুলি যুক্ত হতে থাকবে।

সুতরাং, আপনাকে একটি তৃতীয় প্যারামিটারও বিবেচনা করতে হবে তা হল তা ইন্টিগ্রাল(integral)।

ইন্টিগ্রাল(integral) এর জন্য গাণিতিক ধারণাটি হল একটি নির্দিষ্ট সময়ের মধ্যে যুক্ত করা, আপনি যদি কেবল ক্রটিগুলি যুক্ত করেন তবে এটি ইন্টিগ্রাল(integral) এর মান নির্ধারণ করবে। সুতরাং, আপনি একটি তৃতীয় প্যেন্ট যুক্ত করতে পারেন, যেখানে সময়ের সাথে এই ক্রটি সংযুক্ত হতে থাকে, এটি আপনাকে সময়ের সাথে কতটা ক্রটি জমা করছে তা দেবে। আপনি এই তৃতীয় শব্দ(term) টিকে আপনার নিয়ন্ত্রণ অভিযন্ত্রিতে(control expression) যুক্ত করুন, একটি ক্রটি এর সাথে সমানুপাতিক হবে যেটা আপনার P, একটি ডেরিভেটিভের সাথে সমানুপাতিক হবে যেটা হলো D, এবং এখানে একটি একত্রিত ক্রটির সমানুপাতিক হবে যা I; 3 টি একসাথে নিয়ে PID কন্ট্রোলার(PID Controller) বলে।

(Refer Slide Time: 20:21)



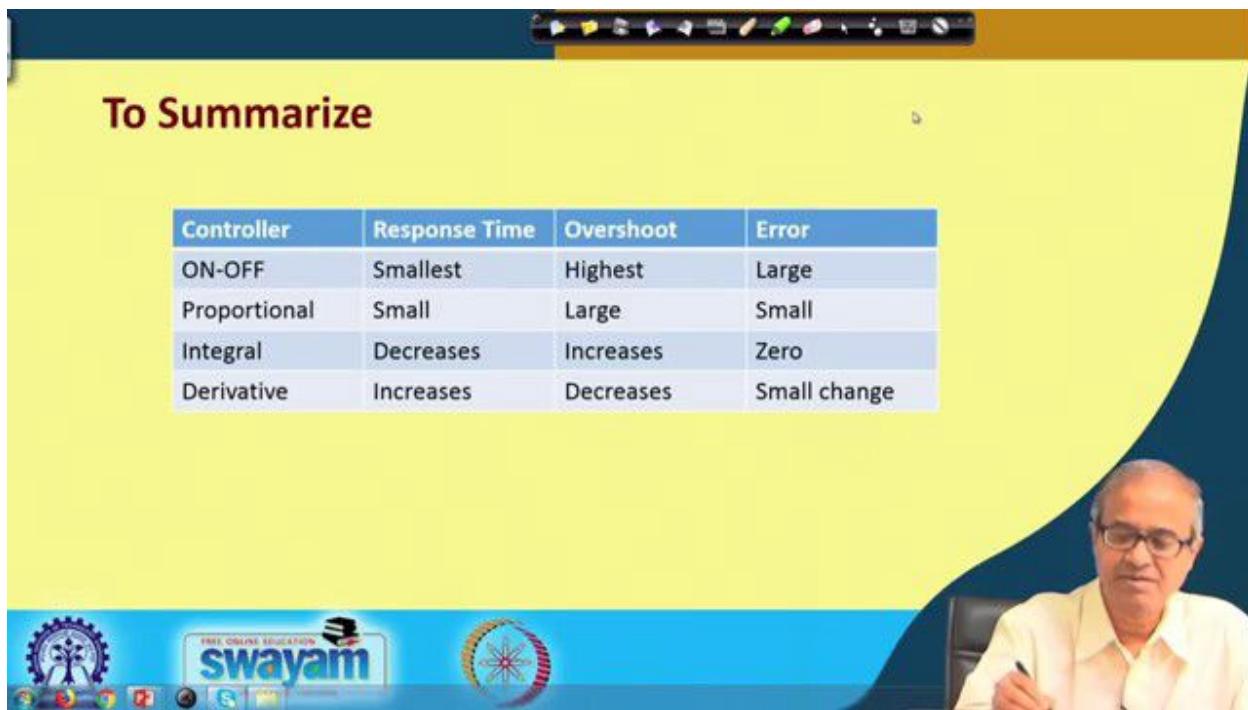
এটি PID কন্ট্রোলার(PID Controller)। এটি স্পষ্টতই সবচেয়ে শক্তিশালী, তবে এখন আমাদের কাছে কমবেশি করার মতো ৩ টি আলাদা জিনিস রয়েছে, একটি নির্দিষ্ট প্রয়োগের জন্য সমানুপাতিকতার ৩ টি ফ্র্যাক কী হবে তা একটি কঠিন প্রশ্ন। আপনার কাছে সেট পয়েন্ট রয়েছে, আপনার পরিমাপের ব্যবস্থা রয়েছে, আপনার কাছে নিয়ামক(controller) এর একটি সমানুপাতিক অংশ, ইন্টিগ্রাল অংশ, ডেরিভেটিভ অংশ রয়েছে, আপনি এই তিনটি যুক্ত করে নিন। K_p , K_i এবং K_d সমানুপাতিকতার তিনটি পৃথক ফ্র্যাক থাকতে পারে এবং আপনি নিয়ন্ত্রণ করার যে চেষ্টা করছেন সেই প্রক্রিয়ার জন্য আপনি একটি একত্রিত কন্ট্রোল সিগনাল(control signal) তৈরি করুন। এখন আবার, আমি আপনাকে বলেছিলাম যে এটি কম-বেশি করা সবচেয়ে কঠিন, তবে আপনি একবার এটি কমবেশি করার পরে এটি সবার মধ্যে সেরা পারফরম্যান্স দেবে।

PID কন্ট্রোলার এই অর্থে এবং এমবেডেড সিস্টেম(embedded system) এর প্রয়োগ গুলিতে ভাল, কারণ আপনি কিছু সরঞ্জাম নিয়ন্ত্রণের বিষয়ে কথা বলছেন এবং এগুলি সবই ফিডব্যাক কন্ট্রোল সিস্টেম(Feedback control system)। আপনি রেফ্রিজারেটরের মতো কিছু প্রয়োগের জন্য নিয়ন্ত্রণের উপযুক্ত মোড়টি বেছে নিতে পারেন, যার জন্য অন-অফ(ON-OFF) নিয়ন্ত্রণ(control) যথেষ্ট ভাল। তবে আপনি আরও পরিশীলিত(sophisticated) অ্যাপ্লিকেশনগুলিতে PD কন্ট্রোল বা PID কন্ট্রোল রাখতে পারেন, যেখানে আউটপুটটিতে ক্রটি নির্দিষ্ট সীমা ছাড়িয়ে যেতে পারে না।

(Refer Slide Time: 22:05)

To Summarize

Controller	Response Time	Overshoot	Error
ON-OFF	Smallest	Highest	Large
Proportional	Small	Large	Small
Integral	Decreases	Increases	Zero
Derivative	Increases	Decreases	Small change



টেবিলটি পার্থক্যগুলির সংক্ষিপ্তসার প্রকাশ করে।

তবে এই সমস্ত জিনিসগুলির সত্যই প্রয়োজন নেই, আপনি কোনও প্রয়োগকে নিয়ন্ত্রণ করার চেষ্টা করলে প্যারামিটার সম্পর্কে আপনাকে আরও ভালভাবে বুঝতে হবে এবং তারপরে আপনি নিয়ন্ত্রণের উপর্যুক্ত পদ্ধতি নির্বাচন করতে পারেন। সরলতার জন্য আমরা যে পরীক্ষাগুলি প্রদর্শিত করব সেগুলিতে আমরা অন-অফ(ON-OFF) ধরনের নিয়ন্ত্রণ ব্যবহার করব, কারণ প্রোগ্রাম(program)টি দেখানো সহজ এবং লেখাও সহজ। কারণ, আপনি দেখতে পান যে আপনি অন-অফ(ON-OFF) বা PID কন্ট্রোল ব্যবহার করছেন তা সম্পূর্ণ আপনার স্ট্রাকচার(software) এর উপর নির্ভর করে, এবং হার্ডওয়্যার(hardware) এর সাথে কিছুই করার নেই।

এটির সাথে আমরা এই বক্তৃতাটির শেষে এসেছি, যেখানে আমরা কন্ট্রোল সিস্টেমগুলি সম্পর্কে বিশেষত ফিডব্যাক কন্ট্রোল সিস্টেম(feedback control system) এবং ক্রটি ও অন্যান্য পছন্দসই বৈশিষ্ট্যগুলি হ্রাস করার জন্য কন্ট্রোল সিগন্যাল(control signal) উত্পন্ন করার কয়েকটি প্রচলিত(standard) উপায় নিয়ে আলোচনা করেছি।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture – 31
Experiments with Relay

এই বক্তৃতাটিতে, আমরা আপনাকে রিলে(relay) ব্যবহার করে প্রত্যক্ষ কিছু পরীক্ষা করে দেখাব। রিলে(relay) কী তা আপনি আগেই দেখেছেন। রিলে এমন একটি ডিভাইস(device) যা অনেক কম ভোল্টেজ(voltage) ব্যবহার করে মাইক্রোকন্ট্রোলারের(microcontroller) নিয়ন্ত্রণের মাধ্যমে উচ্চতর পাওয়ার(power) সার্কিট(circuit) এ পরিবর্তিত হতে পারে।

(Refer Slide Time: 00:47)

CONCEPTS COVERED

- Relay interfacing
- Experiments with relay interface
- Demonstration

FREE ONLINE EDUCATION
swayam
THEORY HOME WORK

আমরা মাইক্রোকন্ট্রোলার বোর্ডগুলির সাথে রিলে ইন্টারফেসিংয়ের(interfacing) বিষয়ে কথা বলব এবং আমরা আপনাকে কিছু পরীক্ষা-নিরীক্ষা ও প্রদর্শন(demonstration) দেখাব।

(Refer Slide Time: 00:59)

About the SRD-05DC-SL-C Relay

- Can be controlled using a 0V – 5V digital signal.
- The output load can be 10A, 250VAC.
- It is an electro-mechanical relay, with two states.

Normally Closed (NC)
Common (C)
Normally Open (NO)

NC
COM
NO

Relay: Is normal Signal = 0V

NC
COM
NO

Relay: Is triggered Signal = 5V

FREE ONLINE EDUCATION
SWAYAM

আমাদের প্রদর্শনী তে আমরা যে ধরণের রিলে ব্যবহার করব তা হ'ল SRD-05C-SL-C। আপনি দেখতে পাচ্ছেন এটি রিলে বোর্ডের একটি চিত্র যা আমরা ব্যবহার করব। আপনি লক্ষ করতে পারেন যে এই পরীক্ষায় যে বিশেষ রিলে আমি দেখাবো তা একটি একক রিলে ডিভাইস(single relay device), তবে এমন বোর্ডও পাওয়া যায় যেখানে 2 টি বা 4 টি রিলে ডিভাইস একটি একক বোর্ডে সংহত করা হয়।

সুতরাং একই মাইক্রোকন্ট্রোলার থেকে একাধিক বৈদ্যুতিক সরঞ্জাম নিয়ন্ত্রণ করা যায়, এমন প্রয়োগও থাকতে পারে। সেক্ষেত্রে আপনি সেই বোর্ডগুলির মধ্যে একটি ব্যবহার করতে পারেন। তবে এখানে আমরা একটি একক সরঞ্জামই নিয়ন্ত্রণ করব তাই আমি একক রিলে মডিউল(module) টি ব্যবহার করছি। এখন এই রিলে সম্পর্কে বলছি, আপনি একদিকে দেখুন, আপনি এই সিগন্যাল(signal) টি মাইক্রোকন্ট্রোলারের সাথে সংযুক্ত করুন। আপনি দেখতে পাচ্ছেন যে এখানে একটি GND রয়েছে, এখানে একটি +5 ভোল্ট রয়েছে, যা এই ডিভাইসটিকে বিদ্যুৎ সরবরাহ করে এবং সেখানে একটি তৃতীয় পিনে একটি ডিজিটাল সিগন্যাল(digital signal) রয়েছে যার সাহায্যে আপনি রিলে চালু বা বন্ধ করতে পারেন।

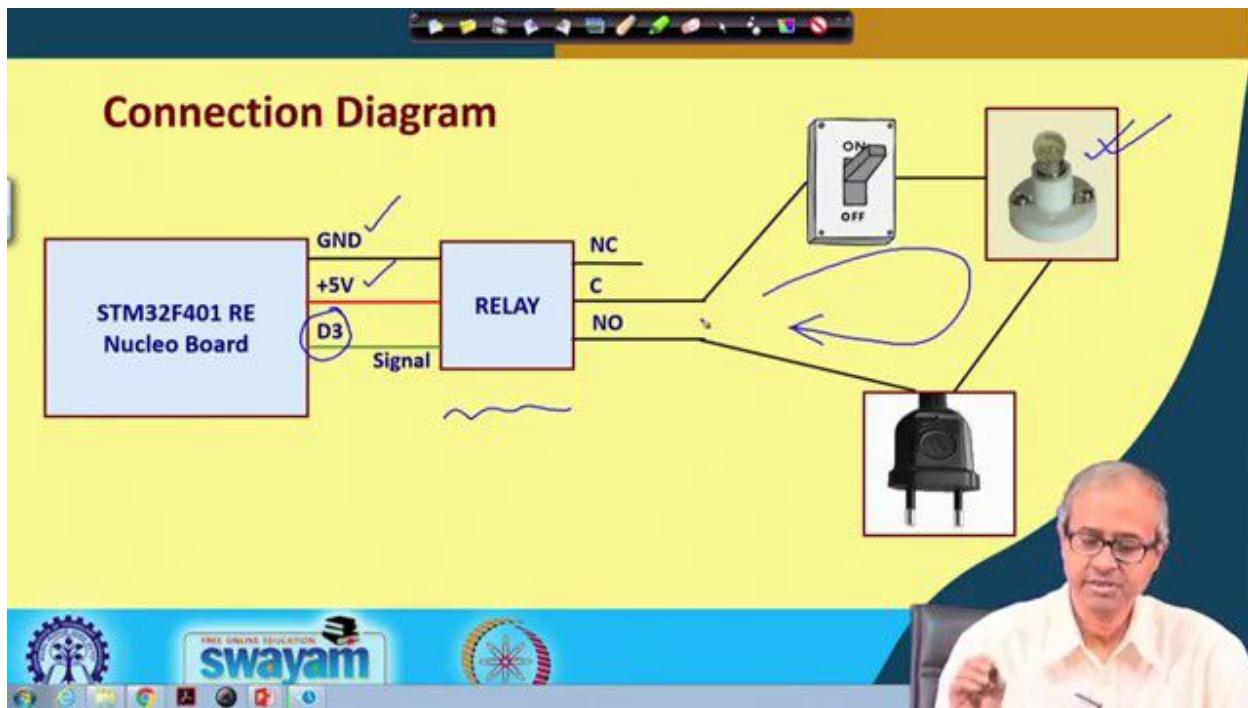
এই সিগন্যাল(signal) টি 0 থেকে 5 ভোল্টের পরিসরে কাজ করতে পারে, 0 এর অর্থ রিলে বন্ধ থাকবে, আপনি 5 ভোল্ট প্রয়োগ করলে রিলে চালু হবে। এবং আউটপুট(output) এর দিকে, আপনার একটি উচ্চ শক্তি সম্পর্ক ডিভাইস সংযোগ করার কথা, এটি 10 এমপিয়ার(ampere) এবং 250 ভোল্ট পর্যন্ত এসি(AC) পাওয়ার(power) সাপ্লাই(supply) হতে পারে, আপনি আউটপুট দিকটি সুইচ অন(switch ON) করতে পারেন। লক্ষ্য করার বিষয়টি হ'ল এই ডিভাইসটি কোনও সলিড স্টেট রিলে(solid state relay) নয়, বরং এটি একটি ইলেক্ট্রোম্যাকানিকাল রিলে(electromechanical relay)। সুতরাং, আপনি যখন কারেন্ট(current) প্রয়োগ করেন তখন ভিতরে একটি তড়িৎ চৌম্বক থাকে, যা চৌম্বকীয় হয়। একটি ধাতব স্যুইচ রয়েছে, যা আকর্ষিত হয় এবং যদি কোনও কারেন্ট(current) না থাকে তবে স্প্রিং লোডিং(spring loading) ব্যবহার করে এটি বন্ধ হয়ে যায়।

আমরা যখন রিলে চালু এবং বন্ধ করি তখন আপনি একটি শব্দ টাক টাক টাক শব্দে পান। স্যুইচটি সংযুক্ত হবে এবং এইভাবে সংযোগ বিচ্ছিন্ন হবে। আউটপুট এর দিকে আপনি দেখতে পাচ্ছেন যে 3 টি সংযোগকারী(connector) রয়েছে, একটি normally closed (NC), একটি common (COM), যা গ্রাউন্ড(ground) এবং normally open (NO) এর সাথে সংযুক্ত থাকে। সুতরাং, আপনি যখন কোনও ডিভাইস সংযুক্ত করবেন তখন আপনি সেগুলি COM এবং NO এর মধ্যে অথবা COM এবং NC এর মধ্যে সংযুক্ত করবেন।

পার্থক্যটি হ'ল আমরা যদি NO ব্যবহার করি তখন রিলে চালু হয় না; তার মানে, সিগন্যাল বা সংকেতটি 0, এই সার্কিটটিও উন্মুক্ত থাকবে এবং কোনও বিদ্যুৎ প্রবাহ থাকবে না, তবে আমরা যখন 5 ভোল্টের সংকেত(signal) প্রয়োগ করব তখন এই সার্কিট(circuit)টি চালু হবে। তবে NC পিলের জন্য এটি একদম বিপরীত হয়। আপনি যখন সিগন্যালে 0 ভোল্ট প্রয়োগ করছেন; এর অর্থ, রিলে বন্ধ রয়েছে, তবে আউটপুট দিকে সার্কিটটি চালু থাকবে। তবে, আপনি রিলে চালু করলে সার্কিট বন্ধ হয়ে যাবে; তার মানে, একদম বিপরীত ক্রিয়া। চিত্রগতভাবে এটি চিত্রিতে দেখানো হয়েছে।

এটি রিলের স্বাভাবিক অবস্থা, যেখানে এই ইনপুট সিগন্যালটি আপনি প্রয়োগ করছেন যা 0 ভোল্টে রয়েছে। আপনি NC সংযোগের জন্য দেখতে পাচ্ছেন যে এটি অভ্যন্তরীণভাবে স্বাভাবিক অবস্থায় সংযুক্ত রয়েছে, তবে NO এর জন্য এটি উন্মুক্ত। কিন্তু, রিলে যখন ট্রিগার(trigger) হয়, মানে এই ইনপুট সিগন্যালটি 5 ভোল্ট হয় তখন আপনি দেখবেন বিপরীতটি ঘটে। সংযোগটি অস্থির হয়ে যায় এবং NO সংযোগটি COM এর সাথে সংযুক্ত হয়। সুতরাং, আমরা হয় NO এবং COM এর মধ্যে, বা NC এবং COM এর মধ্যে সংযোগ স্থাপন করব।

(Refer Slide Time: 05:27)



এটি সেই সংযোগ ডায়াগ্রাম সম্পর্কেই যেখানে আমরা মাইক্রোকন্ট্রোলারের সাথে রিলে সংযুক্ত করব; ডেমনস্ট্রেশন(demonstration) এর জন্য আমরা STM32 বোর্ড ব্যবহার করব। এটি রিলেটির ব্লক ডায়াগ্রাম(block diagram) যেটি আমি মাঝখানে দেখাচ্ছি। আপনি যখন এটি এসটিইএম(STM) বোর্ডের সাথে সংযুক্ত করবেন তখন আপনার গ্রাউন্ড(ground) সংযোগের প্রয়োজন হয়, আপনার একটি +5 ভোল্টের প্রয়োজন এবং আপনার একটি সিগন্যাল(signal) এর প্রয়োজন।

এই পরীক্ষায়, আমি সিগন্যাল ফিডিং(signal feeding) এর জন্য ডেটা(data) লাইন D3 ব্যবহার করব এবং প্রদর্শনের উদ্দেশ্যে, আমরা একটি ছোট এলাইডি(LED) বাল্ব(bulb) ব্যবহার করব, যা রিলে(relay)র নিয়ন্ত্রণে চালু এবং বন্ধ হবে। আমরা NO ব্যবহার করব, মানে রিলেটি যখন সুইচ অন থাকবে না বাল্বটি তখন জ্বলবে না।

এটি সেই সার্কিট যার মাধ্যমে বিদ্যুৎ প্রবাহিত হবে। আপনি দেখতে পাচ্ছেন এখানে একটি বৈদ্যুতিক পাওয়ার(power) কর্ড(cord) রয়েছে, যা মেইন(main)গুলির সাথে সংযুক্ত হবে, এখানে বাল্ব(bulb) টি রয়েছে এবং একটি বাহ্যিক সুইচ(switch) রয়েছে। অবশ্যই আমাদের এই পরীক্ষায় সুইচটির প্রয়োজন হবে না, এই সুইচটি সর্বদা চালু থাকবে। যখনই রিলে চালু হবে, একটি বিদ্যুৎ প্রবাহিত হবে এবং বাল্বটি জ্বলবে; এবং রিলে

বন্ধ হয়ে গেলে, কোনও বিদ্যুৎ থাকবে না এবং বাল্বটি বন্ধ হয়ে যাবে। সুতরাং, এটি হল সংযোগ ডায়াগ্রাম(diagram) যা আমরা ব্যবহার করব।

(Refer Slide Time: 06:57)

Experiment 1

- Interface an electric bulb through relay, and switch it ON and OFF under program control through digital output port.
 - The relay is controlled by the digital output line D3 of the STM32 board.
 - The program turns ON the bulb for 2 seconds, turns it OFF for 5 seconds, and repeats it in a loop.

এখন, আসুন প্রথম পরীক্ষায় আসি। আমরা কিছু সময় অন্তর রিলে চালু এবং বন্ধ করব। আপনি যদি এই পরীক্ষার বিবৃতিটি দেখেন তবে আমরা বৈদ্যুতিন বাল্বটি ইন্টারফেসিং(interfacing) করবো, আমরা ইতিমধ্যে দেখেছি যে কীভাবে আমরা সার্কিট(circuit) টি ইন্টারফেসিং(interfacing) করব, এবং এটি চালু এবং বন্ধ করব।

আমরা ইতিমধ্যে সার্কিট ডায়াগ্রামে দেখেছি যে রিলেটি STM বোর্ডের আউটপুট(output) লাইন D3 এর সাথে সংযুক্ত রয়েছে এবং প্রোগ্রামটি এমনভাবে লেখা হয়েছে যাতে রিলে 2 সেকেন্ডের জন্য চালু হবে এবং 5 সেকেন্ডের জন্য বন্ধ থাকবে, এবং এই প্রক্রিয়া অনিদিষ্টকালের জন্য পুনরাবৃত্তি হবে। আপনাকে ডেমনস্ট্রেশন(demonstration) দেখানোর আগে, আমি প্রথমে আপনাকে কোডটি দেখাবো, এটি অর্জন করতে যে এমবেড(mbed) C কোড(code) টি আমরা লিখেছি।

(Refer Slide Time: 07:59)

Mbed C Code

```
// Turn on a bulb for 2sec and off for 5sec repeatedly through a
// relay, which is controlled by digital port line D3.

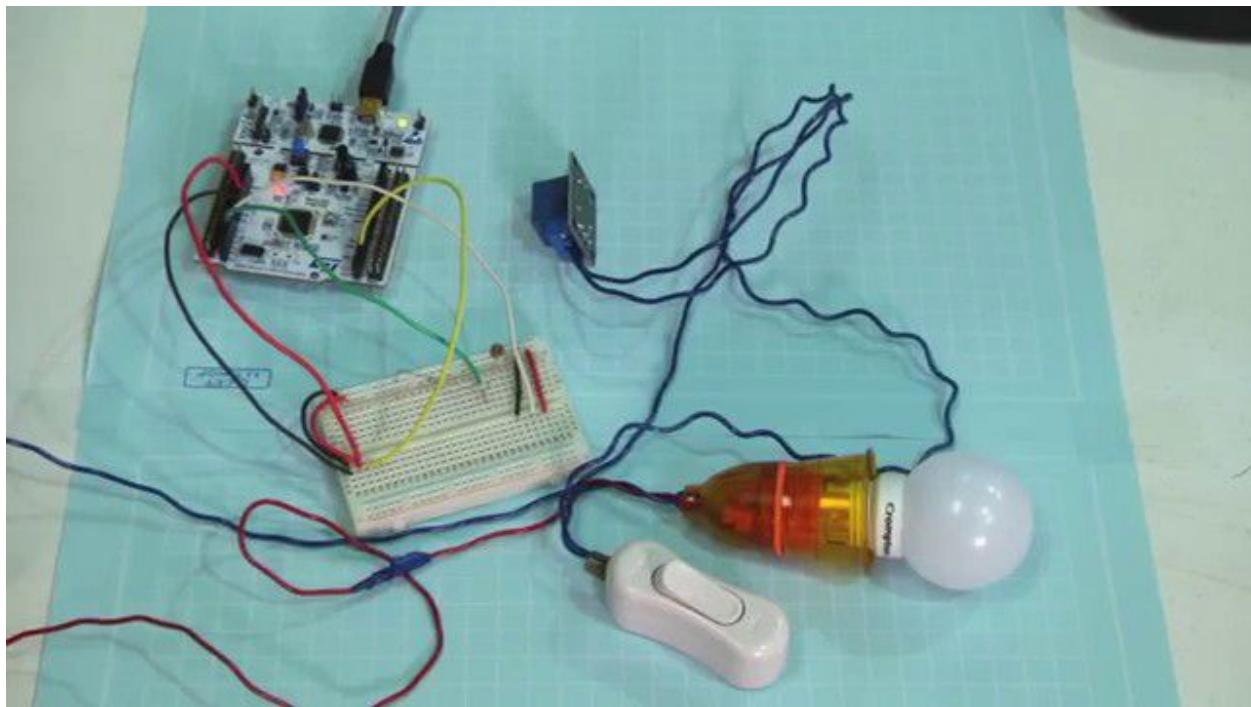
#include "mbed.h"
DigitalOut relay (D3);
int main() {
    while(1) {
        relay = 1;      // Turn ON bulb, and wait for 2 sec
        wait(2.0);
        relay = 0;      // Turn OFF bulb, and wait for 5 sec
        wait(5.0);
    }
}
```



এটি এমন কোড যা মোটামুটি সহজ সরল। যদি আপনি কোডটি লক্ষ্য করেন তবে দেখবেন আমরা এই mbed.h হেডার(header) টিকে অন্তর্ভুক্ত করেছি এবং D3 কে আমরা একটি ডিজিটাল আউট পিন হিসাবে ঘোষণা করেছি। কারণ, আমরা এখানে সেই আউটপুট পিনের একটি সাধারণ ডিজিটাল আউটপুট মোড(digital output mode) ব্যবহার করছি এবং আমরা এই পিনটিকে "relay" বলছি, আপনি এখানে যে কোনও নাম দিতে পারেন। এটি আমাদের মূল ফাংশন(main function) যা একটি ক্রমাগত(continuous) while লুপে(loop) এ চলে।

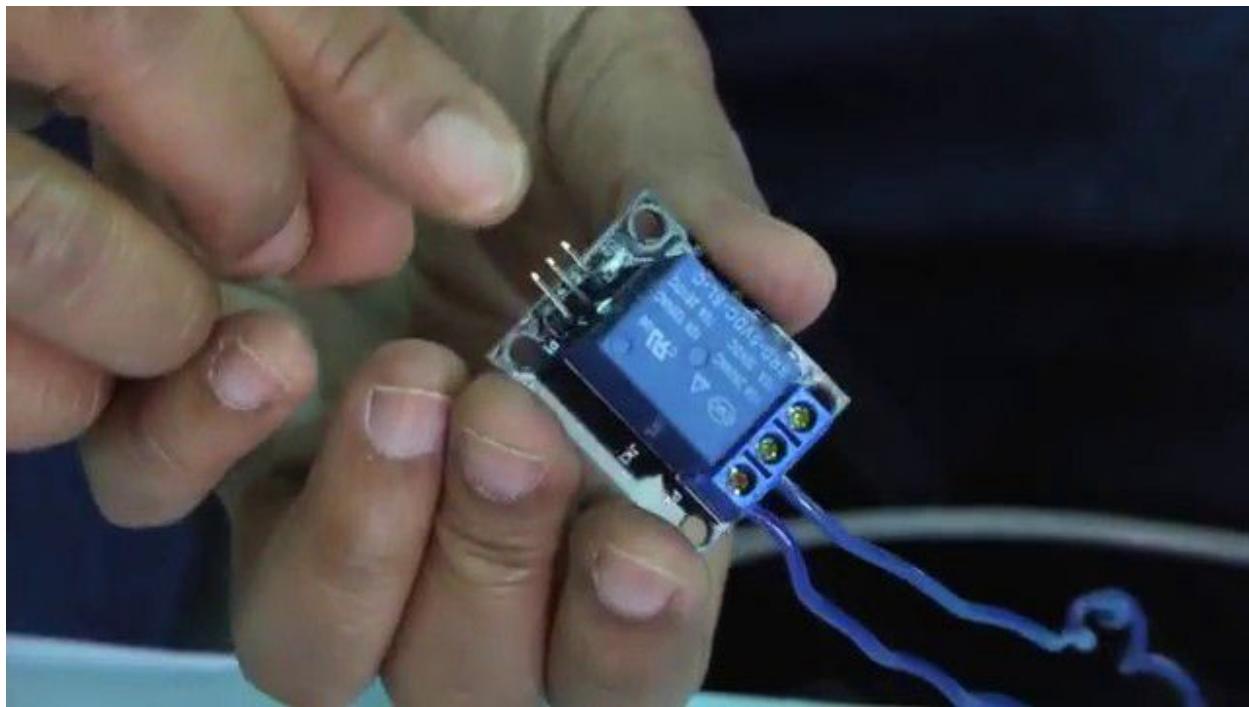
যখন relay তে 1 আউটপুট দেয়, তখন সার্কিটটি চালু হবে এবং বাল্টি 2 সেকেন্ডের জন্য জ্বলতে থাকবে। relay=0 এর অর্থ আবার, সার্কিটটি বন্ধ হবে এবং 5 সেকেন্ড অপেক্ষা করবে। সুতরাং, পুনরাবৃত্ত(repeated) while লুপে(loop) আপনি বাল্টি চালু করুন, 2 সেকেন্ড অপেক্ষা করুন, বাল্টি বন্ধ করুন, 5 সেকেন্ড অপেক্ষা করুন। আসুন এখন আমরা প্রদর্শন টা দেখি।

(Refer Slide Time: 09:23)



এখানে আপনি এই সার্কিটটি দেখুন যা আমি দেখাচ্ছি। আপনি দেখতে পাচ্ছেন এটি আপনার STM মাইক্রোকন্ট্রোলার(microcontroller) বোর্ড এবং আমি যে বাল্বটি দেখিয়েছি তার সার্কিট। এখানে একটি বৈদ্যুতিক পাওয়ার (power) ইনলেট(inlet) রয়েছে যা বৈদ্যুতিক শক্তির উত্স AC 220 ভোল্টের সাথে সংযুক্ত থাকে, এটি একটি সুইচ রয়েছে যা আমি স্থায়ীভাবে চালু রাখছি এবং এটি একটি ছোট এলাইডি(LED) বাল্ব(bulb) যা আমি ব্যবহার করছি এবং এটিই হল রিলে মডিউল যেটা আমরা ব্যবহার করছি।

(Refer Slide Time: 09:53)



আপনি একদিকে দেখুন যে আপনি এই রিলে মডিউলটি NO এবং COM সংযোগগুলির সাথে সংযুক্ত করেছেন এবং অন্যদিকে আপনি দেখেছেন তিনটি পিন রয়েছে। এই পিনগুলি হল signal, Vcc এবং GND; এই ৩টি পিন সংযোগ করতে হবে। এখন এই ব্রেডবোর্ড(breadboard), আমি ইতিমধ্যে এই জাতীয় সংযোগগুলি করেছি। ডানদিকে আমি কেবল এই রিলেটিকে প্লাগইন(plug in) করব।

সুতরাং আমি এই রিলেটিকে এই ব্রেডবোর্ড(breadboard) প্লাগইন(plug in) করেছি, আমাকে একটি শক্ত সংযোগ তৈরি করতে দিন ... হ্যাঁ এখানে একটি আলগা সংযোগ রয়েছে আমাকে এটি ঠিক করতে দিন। সুতরাং এখন, আমি এই প্রোগ্রাম(program)টি সংকলন(compile) করি, আপনি এটি ডিফল্টেরপে(by default) সংরক্ষণ(save) করুন। এটি ডাউনলোড(download) ফোল্ডারে(folder) সংরক্ষিত(save) হয়ে যাবে এবং আপনি এটিকে অনুলিপি(copy) করে F401 ড্রাইভে(drive) রাখুন(paste করুন), প্রোগ্রামটি ডাউনলোড হয়ে গেছে।

এখন, আমি বিদ্যুৎ শক্তি চালু করি। আপনি দেখুন কি ঘটে; বাল্পটি 2 সেকেন্ডের জন্য অন(ON) হবে এবং 5 সেকেন্ডের জন্য অফ(OFF) হবে, এটি আবার 2 সেকেন্ডের জন্য অন(ON) হবে 5 সেকেন্ডের অফ(OFF) হবে, এটি 2 সেকেন্ডের জন্য আবার অন(ON) হবে 5 সেকেন্ডের জন্য অফ(OFF) হবে। সুতরাং, এক্ষেত্রে আপনি দেখতেই পান রিলে ইন্টারফেসিং মোটামুটি সহজ। আসুন আমরা আবার আমাদের উপস্থাপনা(presentation) এবং পরবর্তী পরীক্ষায় ফিরে আসি।

(Refer Slide Time: 12:35)

Experiment 2

- Interface an electric bulb through relay, and vary its brightness under program control through PWM port.
 - The circuit connection remains the same, but now the digital port D3 is used with PWM control.
 - By changing the duty cycle, the brightness of the bulb can be controlled.

D3 Avg value of duty cycle

আপনি অবশ্যই লক্ষ্য করেছেন যে বাল্বটি যখন চালু এবং বন্ধ হয় তখন একটি শ্রবণ যোগ্য শব্দ আসে যা ইঙ্গিত করে যে রিলে(relay) সুইচটি চালু এবং বন্ধ হচ্ছে। পরবর্তী পরীক্ষায় আপনি এটি আরও স্পষ্টভাবে শুনতে পাবেন কারণ, এটি আরও দ্রুত চালু এবং বন্ধ করা হবে।

দ্বিতীয় পরীক্ষায় আসছি, ইন্টারফেস(interface)টি অনেকটা একই রকম, আমরা আবার একই সার্কিট ব্যবহার করে রিলে(relay) দিয়ে বাল্বটি সংযুক্ত করছি, আমরা এখনও রিলেটি D3 পোর্ট(port) টিতে সংযুক্ত করছি, তবে পূর্ববর্তী পরীক্ষায় আমরা ডিজিটাল নিয়ন্ত্রণ(digital control) ব্যবহার করছিলাম, হয় 0 বা 1, কিন্তু এই পরীক্ষাতে D3 তে একটি অবিচ্ছিন্ন পালস ট্রেন(pulse train) প্রেরণের জন্য PWM নিয়ন্ত্রণ ব্যবহার করা হচ্ছে।

সুতরাং, রিলে নির্দিষ্ট সময়ের জন্য চালু করা হবে এবং নির্দিষ্ট সময়ের জন্য বন্ধ করা হবে, এবং এটি বারবার ঘটবে। এবং ডিউটি সাইকেল(duty cycle) টি নিয়ন্ত্রণ করে বাল্বের উজ্জ্বলতা পরিবর্তন করা যাবে, কারণ আপনি জানেন যে PWM এর ভোল্টেজ আউটপুটটির গড় মান ডিউটি সাইকেল(duty cycle) এর সমানুপাতিক হবে।

সুতরাং, আমরা রিলেতে প্রয়োগ হওয়া ডিউটি সাইকেল(duty cycle) এর ভোল্টেজের গড় মান পরিবর্তন করব এবং তাই বাস্তুটি ডিউটি সাইকেল(duty cycle) এর আনুপাতিক ভাবে পরিবর্তিত হবে।

(Refer Slide Time: 14:33)

Mbed C Code for STM32

```
#include "mbed.h"

PwmOut bulb (D3);      // D3 is set as a PWM controlled digital output
int main() {
    bulb.period (0.02);    // PWM period 20 msec

    while(1) {
        bulb.write (1.0);    // duty cycle = 1.0;
        wait (3.0);          // wait for 3 second

        bulb.write (0.8);    // duty cycle = 0.8;
        wait (3.0);          // wait for 3 second
    }
}
```

এই প্রোগ্রামে(program), কেবল প্রদর্শনের জন্য, আমরা mbed.h কে অন্তর্ভুক্ত করেছি, তবে এখন আমরা এই D3 কে PwmOut হিসাবে ঘোষণা করেছি এবং "bulb" নাম দিয়েছি। এখন, মূল ফাংশনে(main function) আমরা কয়েকটি কাজ করেছি। এখন, শুরুতে, এই PwmOut এর জন্য, আপনি ইতিমধ্যে জানেন যে কিছু ফাংশন রয়েছে যেগুলি আমরা পরিয়ড(period) সেট করতে, ডিউটি সাইকেল সেট করতে ইত্যাদির জন্য ব্যবহার করতে পারি।

প্রথমে আমরা সময়কাল(period) কে 0.02 সেকেন্ড সেট করি যার অর্থ 20 মিলি সেকেন্ড। এর অর্থ 50 হার্টজ। সুতরাং, প্রতি সেকেন্ডে রিলে 50 বার চালু এবং বন্ধ হবে। আমি এটি 50 রেখেছি; লক্ষ্য করার বিষয়টি হ'ল আপনি যান্ত্রিক রিলে ব্যবহার করার কারণে, আপনার এই ফ্রিকোয়েন্সি(frequency)টি খুব বেশ দ্রুত হতে পারে না, কারণ রিলেটি অন এবং অফ হওয়ার সময় পাবে না।

প্রোগ্রামটি কিছুক্ষণ while লুপে(loop) অব্যহত থাকে, আমরা পুনরাবৃত্ত ফ্যাশনে(repetitive fashion) একে একে নির্দিষ্ট কিছু কাজ করেছি। bulb.write() একটি ফাংশন, যেখানে আপনি ডিউটি

সাইকেল(duty cycle) টি সেট করতে পারেন। প্রথমে আমরা ডিউটি সাইকেল(duty cycle) এ 1 সেট করছি, যার অর্থ এটি ক্রমাগত 1 হবে, এটি কখনও 0 হবে না, এটি সর্বোচ্চ উজ্জ্বলতা এবং আপনি 3 সেকেন্ড অপেক্ষা করুন। তারপরে আমরা ডিউটি সাইকেল(duty cycle) টি 80% করি। সুতরাং, কিছুটা কম, তারপরে আবার 3 সেকেন্ড অপেক্ষা করুন এবং এটি আমরা পুনরাবৃত্তি করব।

(Refer Slide Time: 16:37)

```
bulb.write (0.6); // duty cycle = 0.6;
wait (3.0); // wait for 3 second

bulb.write (0.4); // duty cycle = 0.4;
wait (3.0); // wait for 3 second

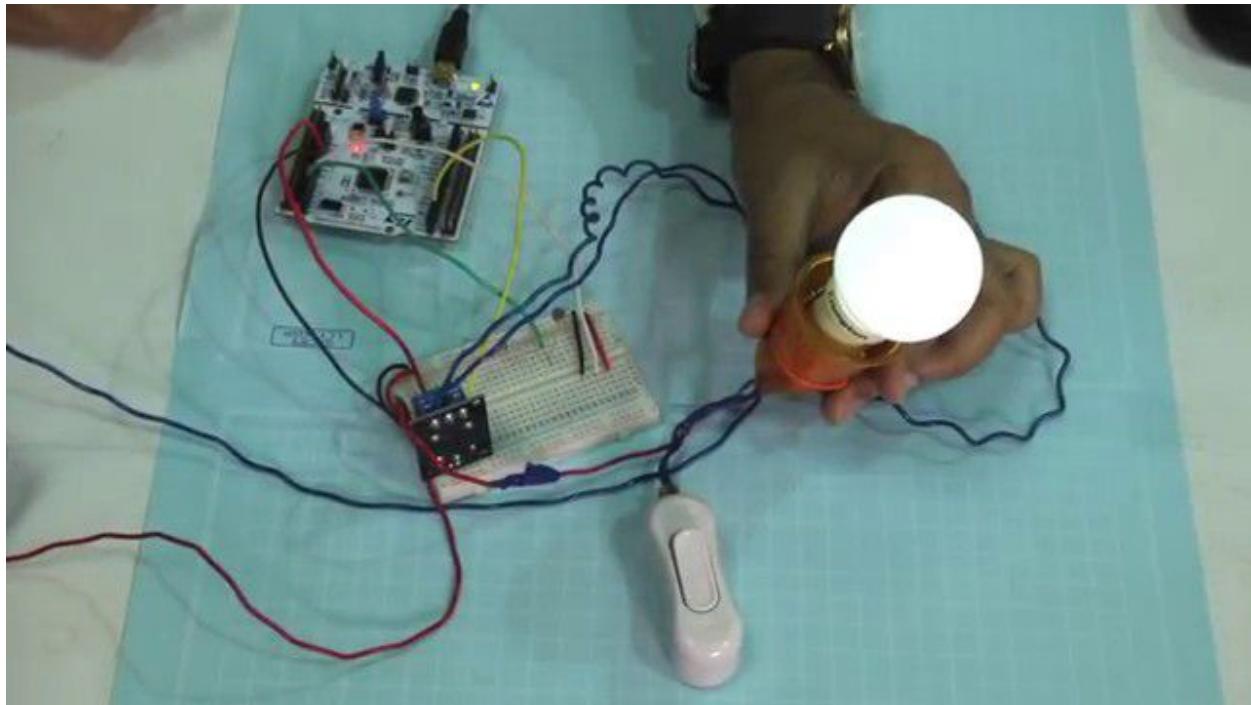
bulb.write (0.2); // duty cycle = 0.2;
wait (3.0); // wait for 3 second

bulb.write (0.0); // duty cycle = 0.0;
wait (3.0); // wait for 3 second
}
```

তারপরে আমরা ডিউটি সাইকেল(duty cycle) কে 0.6 করে আবার 3 সেকেন্ডের জন্য অপেক্ষা করি, তারপরে 0.4 করে 3 সেকেন্ডের জন্য অপেক্ষা করি, 0.2 করে 3 সেকেন্ডের জন্য অপেক্ষা করি এবং শেষ পর্যন্ত আমরা সম্পূর্ণ বন্ধ করব।

সুতরাং, একটি লুপে আমরা দেখব বাল্লের উজ্জ্বলতা ধারাবাহিকভাবে ক্রমান্বয়ে পরিবর্তিত হবে। আসুন আমরা প্রদর্শনটি দেখি।

(Refer Slide Time: 17:17)



এটি আমাদের দ্বিতীয় পরীক্ষা। আসুন আমরা আবার এই প্রোগ্রামটি সংকলন(compile) করি, আমি যে প্রোগ্রামটি দেখিয়েছি একই প্রোগ্রামটি এখানে রয়েছে। আমরা এটি সংরক্ষণ(save) করছি, আমরা এটি কপি(copy) করছি, আমরা এটি পেস্ট(paste) করেছি এবং নতুন প্রোগ্রামটি ডাউনলোড(download) হচ্ছে।

এখন দেখুন কি হয়। প্রতি সেকেন্ডে রিলে 50 বার চালু এবং বন্ধ হয়। আপনি শব্দটি খুব স্পষ্টভাবে শুনতে পাচ্ছেন, এবং আপনি দেখতে পান যে 3 সেকেন্ড অন্তর বাল্বের উজ্জ্বলতা পরিবর্তিত হচ্ছে। এটি সর্বোচ্চ উজ্জ্বলতা, তারপরে এটি 0.8, এটি 0.6, এটি 0.4, এটি 0.2 খুব হালকা এবং তারপরে সম্পূর্ণ বন্ধ; এবং এই চক্রটি পুনরাবৃত্তি হয়।

এই সাধারণ পরীক্ষাটি আপনাকে প্রকৃতপক্ষে দেখায় যে আমরা কীভাবে কোনও ডিউটি সাইকেল(duty cycle) এবং PWM নিয়ন্ত্রিত ফ্যাশনের সময়কাল এর জন্য কোনও বাল্ব বা কোনও সার্কিট চালু এবং বন্ধ করতে পারি। সুতরাং, আসুন আমাদের আলোচনা চালিয়ে যায়।

(Refer Slide Time: 18:57)

Experiment 3

- Interface an electric bulb through relay, and turn it ON or OFF depending on ambient light as sensed using a LDR.
 - The relay is connected through digital output pin D3 as usual.
 - A LDR light sensing circuit, that generates an analog output voltage, is connected to the analog input pin A1.

FREE ONLINE EDUCATION **swayam**

এখন, আমরা আপনাকে শেষ কোডটির জন্য প্রদর্শন করব, এটি একই রিলে সার্কিট, তবে আমরা এখানে আরও একটি সংযোজন করেছি, আমরা একটি LDR সংযুক্ত করেছি।

আমাদের সার্কিটটি এরকম; রিলে(relay) এর মাধ্যমে বৈদ্যুতিন বাল্ব এর সংযোগ যা ইতিমধ্যে রয়েছে, রিলে এখনও পিন D3 এর সাথে সংযুক্ত রয়েছে এবং সেখানে একটি LDR লাইট সেন্সিং ইউনিট(light sensing unit) রয়েছে, যা আমরা অ্যানালগ ইনপুট পিন A1 এর সাথে সংযুক্ত করেছি।

রিলে সার্কিটটি একই, এবং আমরা এই অতিরিক্ত সার্কিট ব্যবহার করেছি। এখানে, আমরা একটি LDR এবং সম্ভাব্য বিভাজক হিসাবে যুক্ত অন্য একটি প্রতিরোধের ব্যবহার করেছি।

এখন এই আউটপুট ভোল্টেজটি আমরা STM32 বোর্ডের অ্যানালগ ইনপুট পিন A1 এর সাথে সংযুক্ত করেছি। এখন কোডটি দেখুন, আমরা কী করেছি।

(Refer Slide Time: 20:37)

```

#include "mbed.h"

PwmOut bulb (D3); // D3 is set as a PWM controlled digital output
AnalogIn ldr (A1); // LDR circuit output is connected to pin A1
int main()
{
    float light;
    int value;
    bulb.period (0.02);
    while(1) {
        light = ldr.read();
        value = light * 5000;
        if (value > 4000)
            bulb.write (0.0); // OFF
        else
            bulb.write (1.0);
    }
}

```

Mbed C Code

এখন, মূল প্রোগ্রামটিতে আমরা পূর্ববর্তী পরীক্ষার মতো সময়কালটিকে 20 মিলিসেকেন্ডে সেট করছি। এখন while লুপের মধ্যে, আমরা এলডিআর(LDR) থেকে আলোর মান পড়ছি; মানটি 0 এবং 1 এর মধ্যে একটি ভগ্নাংশ। যাতে আপনি এটি তুলনা করতে পারেন, তার জন্য এটিকে পূর্ণসংখ্যার মান পর্যন্ত স্কেল করতে, আমরা এটি 5000 দ্বারা গুণ করব এবং "value" নামক একটি ভেরিয়েবলে(variable) সংরক্ষণ করব।

এখন, পরীক্ষার মাধ্যমে আমরা 4000 একটি উপযুক্ত প্রাণ্তিক মান হিসাবে খুঁজে পেয়েছি। সুতরাং, 4000 এরও বেশি এর অর্থ হল আমাদের পরিবেষ্টনে পর্যাপ্ত আলো রয়েছে এবং আমাদের আলোটি বন্ধ করতে হবে। আপনি কোনও বাড়ির কথা ভাবেন, যখন পর্যাপ্ত আলো থাকে তখন আলোর স্যুইচ অন(switch ON) করার কোনও মানে হয় না। সুতরাং, আমরা ডিউটি চক্র(duty cycle) টি 0.0 তে সেট করেছি, যার অর্থ আলোটি স্যুইচ অফ(switch OFF) করা আছে।

তবে এটি যখন 4000 এরও কম হয়, এর অর্থ অন্ধকার রয়েছে। এখন, আমাদের আলোর স্যুইচ অন(switch ON) করতে হবে। সুতরাং, আমরা ডিউটি চক্র(duty cycle) টি 1.0 তে সেট করেছি, যার অর্থ সর্বাধিক উচ্চতা এবং এই প্রক্রিয়া পুনরাবৃত্তি হয়। এটার জন্য আমরা আপনাকে প্রদর্শন(demonstration)টি দেখায়। আমরা একই কোড দেখছি যেটা আমরা স্লাইডে দেখেছিলাম। আমাকে প্রথমে এই কোডটি সংকলন(compile) করতে দিন, এটি সংরক্ষণ(save) করুন এবং তারপর এটি অনুলিপি(copy) করুন, নিউক্লিয়ো বোর্ড(nucleo board) এটি পেস্ট(paste) করুন।

এখানে রিলে সার্কিট ছাড়াও এখন আপনি দেখতে পাচ্ছেন যে আমরা এখানে একটি এলডিআর(LDR) সংযুক্ত করেছি। আপনি দেখতে পাচ্ছেন এটি LDR, এবং এখানে একটি প্রতিরোধ(resistance) আছে।

সুতরাং, LDR এবং প্রতিরোধ একসাথে সংযুক্ত হয়েছে, এলডিআর এর এক প্রান্তটি 5 ভোল্টের সাথে সংযুক্ত, এবং প্রতিরোধের অন্য প্রান্তটি গ্রাউন্ড(ground) এর সাথে সংযুক্ত, এবং এলডিআর এর মাঝখান থেকে এই সবুজ তারটি অ্যানালগ ইনপুট পিন A1 এর সাথে সংযুক্ত হয়েছে।

আপনি দেখুন এখন যথেষ্ট আলো আছে। সুতরাং, বাল্বটি চালু নেই। এখন আমি LDR টিপলে, আপনি দেখুন বাল্বটি চালু হচ্ছে; তার মানে, অন্ধকার আছে। আমি আবার আমার হাতটি সরিয়ে ফেললাম বাল্বটি আবার বন্ধ হল, আমি এলডিআর টিপছি; তার অর্থ অন্ধকার এবং আলোটি চালু হল।

এটি একটি খুব সাধারণ পরীক্ষা যা আপনাকে দেখায় যে কীভাবে কোনও বৈদ্যুতিক সরঞ্জাম চালু এবং বন্ধ করতে প্রোগ্রাম নিয়ন্ত্রণের অধীনে এলডিআর(LDR) ব্যবহার করা যেতে পারে। এই পরীক্ষাটি আপনি খুব সহজেই কোনও ধরণের হোম অটোমেশন সিস্টেমের সাথে সম্পর্কিত করতে পারেন, যেখানে আপনি বাস্তৱের মতো কিছু বৈদ্যুতিক সরঞ্জাম বন্ধ করতে চান, পারিপার্শ্বিক আলোর মানের উপর ভিত্তি করে বাল্ব একটি বাস্তব উদাহরণ। এখানে আমি আলোর উদাহরণ দিয়েছি; তাপমাত্রার মতো আরও অন্য কিছু থাকতে পারে, যদি তাপমাত্রা খুব বেশি হয়ে যায়, আপনি শীতাতপ নিয়ন্ত্রণ যন্ত্রটি চালু করতে পারেন। তাপমাত্রা খুব কম হলে, আপনি একটি হিটার(heater) চালু করতে পারেন।

পরীক্ষার এই সেটটিতে আমি আপনাকে দেখিয়েছি কীভাবে আমরা কোনও বৈদ্যুতিক সরঞ্জাম চালু বা বন্ধ করতে একটি মাইক্রোকন্ট্রোলার(microcontroller) দ্বারা নিয়ন্ত্রিত রিলে(relay) ব্যবহার করতে পারি। এই ধারণাটি আপনি যে কোনও ধরণের ডিভাইসের জন্য ব্যবহার করতে পারেন, কোনও বাস্তৱের প্রয়োজন নেই। আমি যেমন বলেছি সেইরকম আপনি কোনও এসি(AC) মেশিন বা হিটার চালু এবং বন্ধ করতে পারেন, আপনি একটি ফ্রিজ চালু এবং বন্ধ করতে পারেন, আমি এখানে যেগুলি দেখিয়েছি শুধু সেগুলিই নয়, আপনার এরকম একাধিক যেকোনোও সরঞ্জাম থাকতে পারে যেগুলোকে আপনি চালু এবং বন্ধ করতে পারেন।

আমরা পরে আরও পরীক্ষা-নিরীক্ষার দিকে নজর দেব, যেখানে আপনি এক ধরণের হোম অটোমেশন সিস্টেম(home automation system) দেখতে পাবেন, যেখানে আরও কিছু পরিশীলিত নিয়ন্ত্রণ এবং যোগাযোগ ব্যবস্থা দেখানো হবে। এর সাথে আমরা এই বক্তৃতাটি শেষ করছি।

ধন্যবাদ।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture – 32
Experiments on Speed Control of DC Motor

এই বক্তৃতাটিতে, আমরা আপনাকে আরও কিছু ইন্টারফেসিং(interfacing) পরীক্ষা দেখাব; এখানে আমরা দেখাব যে কীভাবে STM32 মাইক্রোকন্ট্রোলার(microcontroller) বোর্ডের সাথে ডিসি(DC) মোটরটিকে ইন্টারফেস(interface) করব।এই ধরণের ইন্টারফেসিং কীভাবে করা যায় তা আপনাকে দেখানোর জন্য আমরা উদাহরণ হিসাবে একটি ছোট ডিসি(DC) মোটর ব্যবহার করেছি। আমরা আপনাকে মোটর নিয়ন্ত্রণে এবং গতি কীভাবে বৈচিত্র্যপূর্ণ হতে পারে তার উপর কয়েকটি পরীক্ষা দেখাব।

Refer Slide Time: 00:51)

CONCEPTS COVERED

- Motor interface and speed sensing
- Experiments with motor interface
- Demonstration

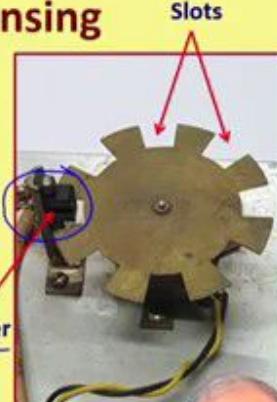
FREE ONLINE EDUCATION
swayam
TEACH ME, LEARN ME

মোটরটিকে কীভাবে ইন্টারফেস করা যায় এবং কীভাবে গতি সংবেদন করা যায় তা নিয়ে আমরা আলোচনা করব।

(Refer Slide Time: 01:03)

About the Motor Interface and Speed Sensing

- A DC motor is used, where the rotating shaft is connected to a metal wheel cut with 8 slots.
- The wheel is attached to an optocoupler circuit, which generates optical interruptions whenever the wheel rotates.
 - 8 optical interrupts for every single rotation of the wheel.
 - An optocoupler circuit generates a pulse for every interruption.
- How is the motor driven?
 - Directly from the PWM port of the microcontroller.
 - By changing the PWM duty cycle, the speed of the motor can be varied.



Optocoupler

Slots

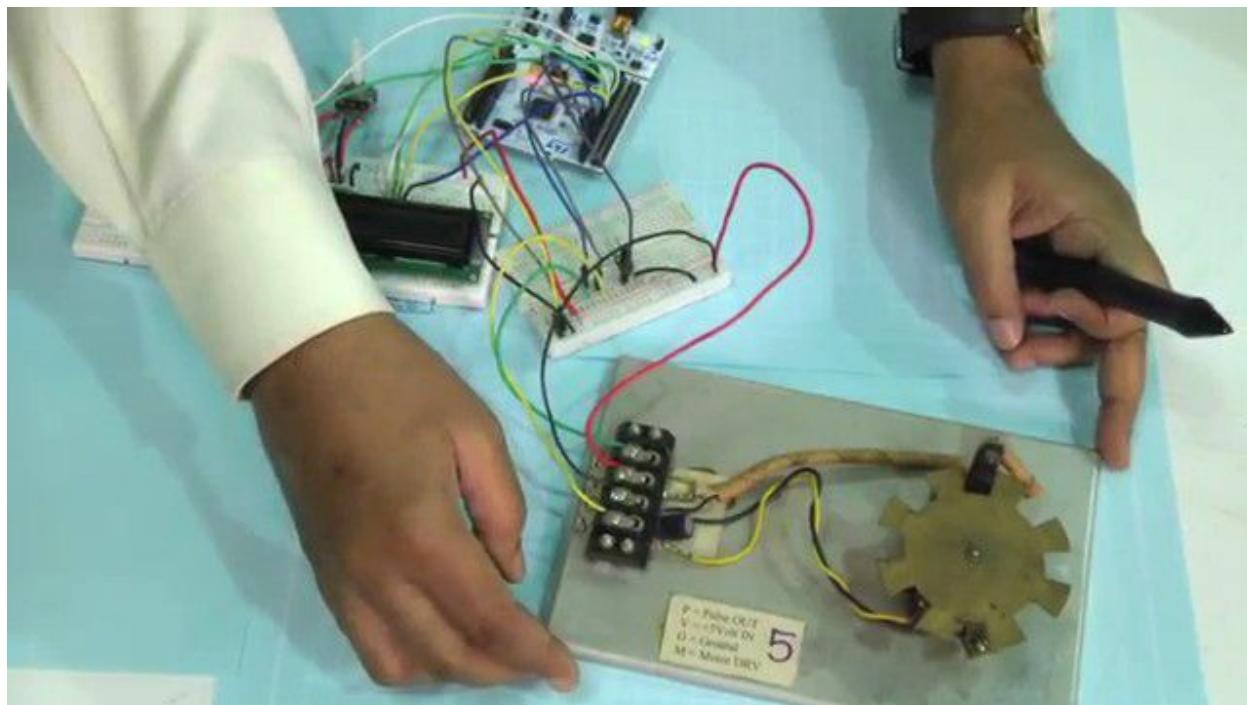
FREE ONLINE EDUCATION

swayam



আসুন আমরা মোটর ইন্টারফেস সম্পর্কে কথা বলি যা আমি আপনাকে এই পরীক্ষায় দেখাব।

(Refer Slide Time: 01:17)



আপনি যদি এটি একবার ভালো করে দেখেন তবে দেখবেন এই মোটরটি আমি ইন্টারফেস করেছি, ঠিক আছে। আপনি নীচে একটি মোটর দেখতে পাচ্ছেন এবং সেখানে একটি চাকা রয়েছে যা শ্যাফট(shaft)এর সাথে সংযুক্ত, যেটা ঘূরতে পারে।

এই স্লাইডে ফিরে আসছি, ডানদিকে ছবিতে আমি একই শ্যাফট(shaft) টি দেখিয়েছি। আপনি যা দেখতে পাচ্ছেন তাতে এখানে 8 টি স্লট(slot) রয়েছে, যেগুলিকে ... 1, 2, 3, 4, 5, 6, 7, 8 এ কাটা হয়েছে কে। মোরানোর শ্যাফট(shaft) টি ধাতব চক্রের সাথে সংযুক্ত যেখানে 8 টি স্লট(slot) রয়েছে। এবং এই দিকে আপনি একটি কালো ধরণের জিনিস দেখতে পাচ্ছেন; একটি অপ্টো-কাপলার সার্কিটের(opto-coupler circuit) একটি ব্যবস্থা রয়েছে, যার বিষয়ে আমি আপনাকে বলব। এর আগে আপনি দেখেছিলেন যে এলডিআর(LDR) সার্কিট(circuit) টি কীভাবে অপটিক্যাল(optical) বাধাগুলি বোঝার জন্য ব্যবহার করা যেতে পারে এবং একটি নির্দিষ্ট পয়েন্টটি অতিক্রম করে এমন বস্তুর সংখ্যা গণনা করতে পারে, এটি কোনও ঘরে প্রবেশ করা এবং ঘর ছেড়ে চলে যাওয়া ব্যক্তির সংখ্যাও হতে পারে।

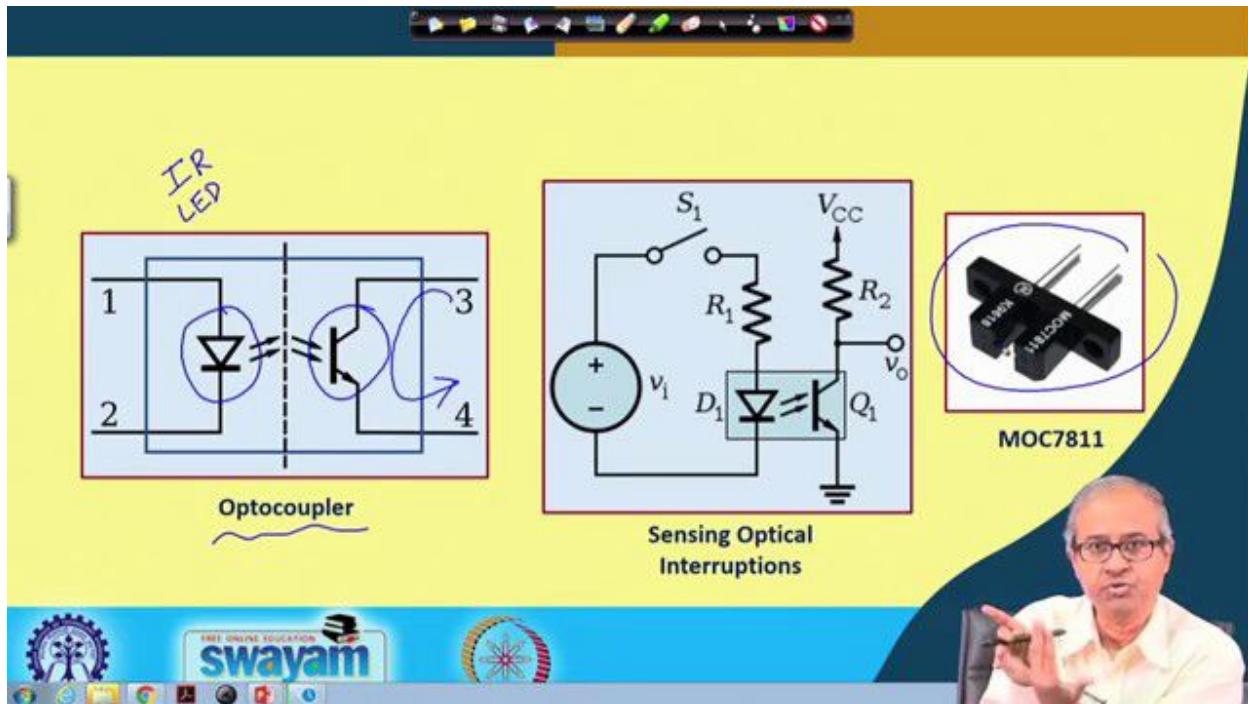
এখানে আমরা এলডিআর(LDR) এবং এক ধরণের এলইডি(LED) আলোর উত্সও ব্যবহার করতে পারতাম, তবে আমরা ভেবেছিলাম যে আমরা অন্য কোনও ধরণের অপটিক্যাল সেন্সিং সার্কিট ব্যবহার করি যা ব্যবহার করে আপনি বাধা পরিমাপ করতে পারেন এবং এটি এমন কিছু যাকে অপ্টো-কাপলার(opto-coupler) বলে।

আমরা আপনাকে বলব যে অপ্টো-কাপলার(opto-coupler) কীভাবে কাজ করে, আপনি দেখতে পাচ্ছেন যে অপ্টো-কাপলার(opto-coupler) টি কৌশলগতভাবে যে স্থানে চাকাটি ঘূরছে ঠিক সেখানেই রাখা হয়, কারণ সেখানে চক্রের মধ্যে 8 টি স্লট রয়েছে এবং একটি সম্পূর্ণ ঘূর্ণনের জন্য 8 টি বাধা থাকবে।

সুতরাং, চক্রের প্রতিটি ঘূর্ণনের জন্য এমন 8 টি অপটিক্যাল(optical) বাধা থাকবে। এবং এই সার্কিটটি ব্যবহার করে আমরা প্রতি বাধার জন্য একটি স্পন্দন তৈরি করব; একটি ঘূর্ণনের জন্য 8 স্পন্দন হবে। এবং আমরা একটি ছোট মোটর ব্যবহার করছি, তাই আমরা এটি মাইক্রোকন্ট্রোলার আউটপুট(output) পোর্ট(port) থেকে সরাসরি চালনা করতে পারি, আমরা একটি PWM পোর্ট ব্যবহার করেছি। তবে, যদি এটি বৃহত্তর মোটর হয় তবে অবশ্যই আপনাকে একটি ড্রাইভার সার্কিট(driver circuit) ব্যবহার করতে হবে যা বৃহত্তর মোটর চালানোর জন্য প্রয়োজনীয় ভোল্টেজ এবং কারেন্ট সরবরাহ করতে পারে, তবে এটি একটি ছোট মোটর তাই আমরা এটিকে সরাসরি চালাতে পারি।

এছাড়াও আমরা PWM পোর্ট ব্যবহার করেছি বলে ডিউটি সাইকেল(duty cycle) পরিবর্তন করে আপনি মোটরটিতে যে নিয়ন্ত্রণ(control) টি পাঠাচ্ছেন তার গড় মান পরিবর্তিত হচ্ছে। এটি করে মোটরের গড় গতি নিয়ন্ত্রণ করা যায়।

(Refer Slide Time: 04:19)



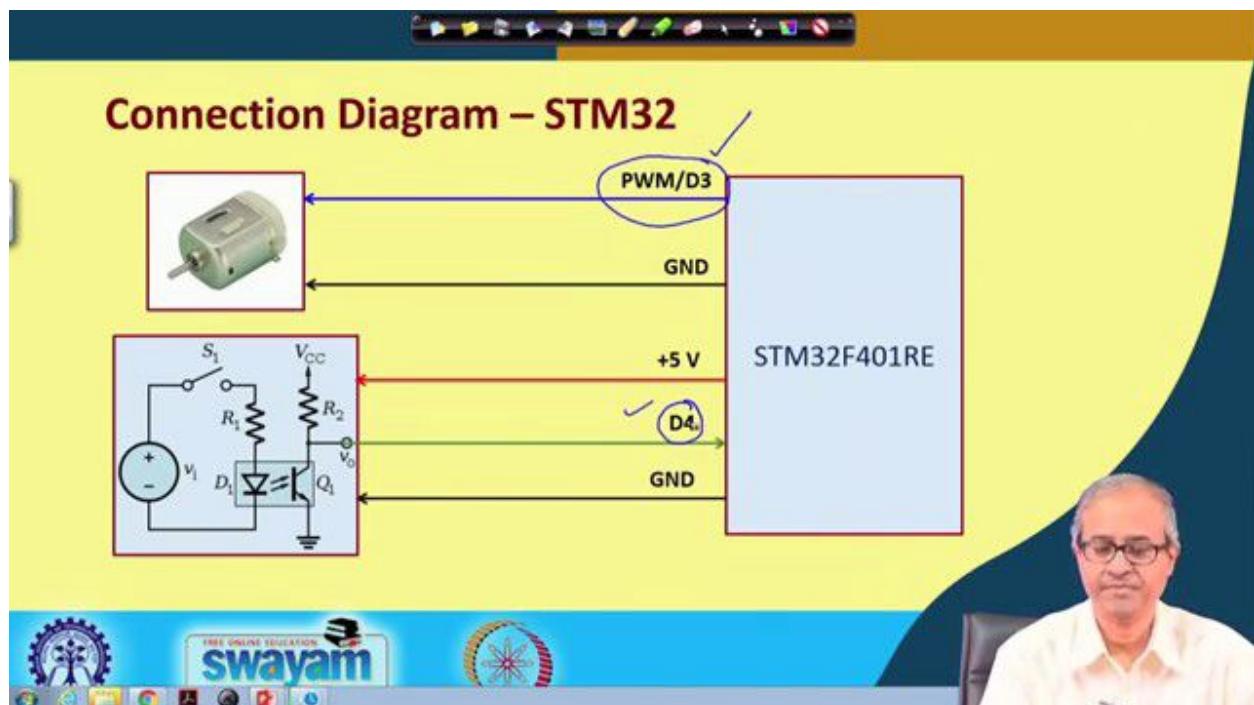
আসুন দেখি কীভাবে এই অপ্টো-কাপলার সার্কিটটি ব্যবহার করে এই অপটিক্যাল(optical) বাধা কাজ করে। মূলতঃ আপনি দেখতে পাচ্ছেন একটি অপ্টো-কাপলার হ'ল দুটি জিনিস এর ব্যবহার: একটি হল কোনও প্রকারের আলোক উৎস, তবে এই সার্কিটে আমরা ইনফ্রারেড(infrared) লাইট(light) এমিটিং(emitting) ডায়োড(diode)(IR LED) কে আলোক উৎস হিসেবে ব্যবহার করছি। এটি থেকে আলো নির্গত হয় তবে তা ইনফ্রারেড(infrared) ফ্রিকোয়েন্সি(frequency) ব্যান্ডে(band)। এবং অন্যদিকে একটি ফটো(photo) ট্রানজিস্টর(transistor) রয়েছে। ফটো(photo) ট্রানজিস্টর(transistor) হ'ল এক ধরণের ট্রানজিস্টর, যেখানে সাধারণ ট্রানজিস্টরের মতো কোনও বেস(base) সংযোগ নেই, কেবল দুটি টার্মিনাল রয়েছে। তবে বেস(base)টি একটি অপটিক্যাল(optical) প্রক্রিয়া দ্বারা প্রতিস্থাপিত হয়, যখনই কোনও আলো থাকে ট্রানজিস্টর চালু থাকে এবং যখন কোনও আলো থাকে না, ট্রানজিস্টর বন্ধ থাকে।

সুতরাং, যখনই এই IR LED এই ট্রানজিস্টরের উপর আলো ফেলে, ট্রানজিস্টর সঞ্চালিত হয় এবং 3 থেকে 4 পর্যন্ত একটি বিদ্যুৎ প্রবাহিত হয়। সুতরাং, আপনি যে সার্কিটটি ব্যবহার করতে পারবেন তা এরকম

কিছু। LED অংশে এক ধরণের LED ড্রাইভিং(driving) সার্কিট থাকবে। যেমনটি আপনি জানেন, এটি একটি বিদ্যুৎ সরবরাহ(power supply) সহ সিরিজের একটি প্রতিরোধের সমন্বয়ে গঠিত এবং এটি আপনি প্রয়োজন হিসাবে স্লিচ(switch) অন(ON) এবং অফ(OFF) করতে পারেন। এবং অন্যদিকে, একটি প্রতিরোধ এবং এই ট্রানজিস্টর নিয়ে অন্য একটি সার্কিট থাকবে, যখনই ট্রানজিস্টর এই আউটপুটটি পরিচালনা করবে তখন ভোল্টেজ কম হবে, এবং যখন এই আউটপুটটি বন্ধ হবে তখন Vcc এর কাছাকাছি থাকবে।

সুতরাং, এই অপ্টো-কাপলার(opto-coupler) এর প্রক্রিয়াটি দেখতে এরকম দেখাচ্ছে। আপনি দেখতে পাচ্ছেন যে এর মাঝখানে একটি ছোট গর্ত রয়েছে এবং চাকাটি এখানে দুটি প্লেটের মাঝে রাখা হয়েছে। এটা এভাবেই সাজানো হয়েছে।

(Refer Slide Time: 06:22)

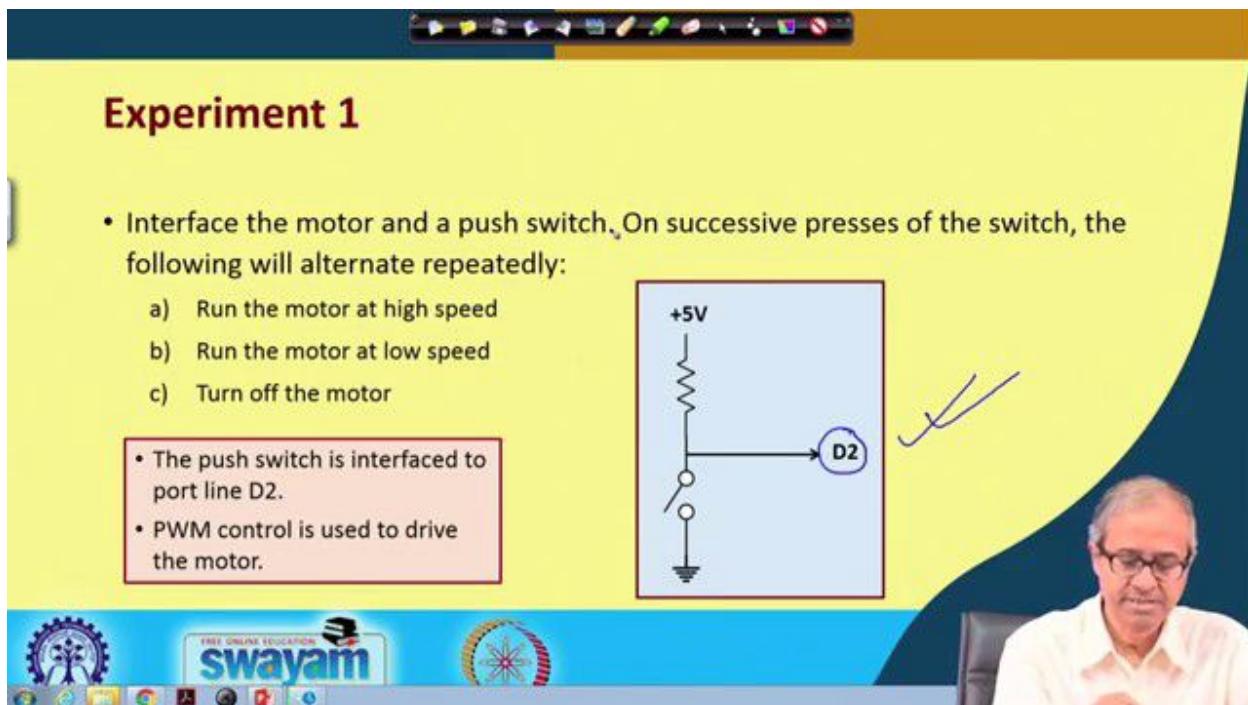


এখন, সংযোগ ডায়াগ্রাম সম্পর্কে কথা বলছি এটি মোটরটি সহজ। এই মোটরটি একটি পিডিএলিউএম(PWM) আউটপুট পোর্ট(port) থেকে সরাসরি চালিত হবে। এখানে আমরা D3 পোর্ট ব্যবহার করেছি, গ্রাউন্ড(ground) টিতে সংযোগ করার জন্য আরও একটি টার্মিনাল(terminal) রয়েছে। এবং এই অপ্টো-কাপলার সার্কিটটি ব্যবহার করে এই মোটরটি চালানোর জন্য বা স্পিড সেন্সিংয়ের জন্য, আমি এই 5 ভোল্ট এবং প্রয়োজনীয় গ্রাউন্ডটি ব্যবহার করছি। এবং অপ্টো-কাপলারের এই আউটপুটটি গতানুগতিক

ডিজিটাল(gigital) ইনপুট(input) পিলে ফিড করানো হয়েছে, কারণ এটি একটি ডিজিটাল আউটপুট হবে, যে 0 বা 1, যা কোনও বাধা নেই বা বাধা আছে তা নির্দেশ করে।

সুতরাং, আমি মোটর এর ঘূর্ণন নিয়ন্ত্রণের জন্য এটি D3 এর সাথে এবং অপ্টো-কাপলারের কাছ থেকে অপটিক্যাল বাধা সংবেদনের জন্য D4 এর সাথে সংযুক্ত করছি। আমাদের প্রথম পরীক্ষায় আমরা কেবল D3 ব্যবহার করব, দ্বিতীয় পরীক্ষায় আমরা D3 এবং D4 উভয়ই ব্যবহার করব।

(Refer Slide Time: 07:31)



Experiment 1

- Interface the motor and a push switch. On successive presses of the switch, the following will alternate repeatedly:
 - Run the motor at high speed
 - Run the motor at low speed
 - Turn off the motor

- The push switch is interfaced to port line D2.
- PWM control is used to drive the motor.

Diagram:

```
graph LR; V[+5V] --- R(( )); R --- S(( )); S --- G[Ground]
```

আসুন প্রথম পরীক্ষাটি সম্পর্কে বলি। প্রথম পরীক্ষায়, আপনি এখনই যে সার্কিটটি দেখলেন তা ব্যবহার করে আমরা মোটর টিকে ইন্টারফেস(interface) করব; তদত্তিরিক্ত, আমরা এটির মতো একটি পুশ বোতাম(push button) সুইচটিও ইন্টারফেস করছি। যদি সুইচ টিপে না রাখা হয় তবে আউটপুট পয়েন্টটি বেশি হবে, যদি সুইচ টিপে রাখা হয় তবে এটি কম হবে; এটি গ্রাউন্ডের সাথে সংযুক্ত হবে। সুতরাং, যখনই সুইচটি টেপা হয় তবে পোর্ট লাইন D2 এর সাথে সংযুক্ত সুইচ আউটপুটটি 0 হয়ে যাবে, এবং যদি সুইচটি টেপা না হয় তবে এটি 1 হবে।

পরীক্ষাটি এভাবে চলে। আমদেরকে ক্রমাগত সুইচটি টিপতে হবে; প্রাথমিকভাবে মোটরটি কিছু গতিতে ঘূরতে থাকবে। যখনই আমরা সুইচটি টিপবো, গতিটি 3 টি বিভিন্ন স্তর পাবে, একটি সর্বাধিক গতি, একটি

মাঝারি গতি এবং অন্যটি বন্ধ; মোটর বন্ধ হয়ে যাবে। সুতরাং, যখন আমি সুইচ টিপতে যাই, সর্বোচ্চ, মাঝারি, বন্ধ, এই চক্রটি পুনরাবৃত্তি করবে।

(Refer Slide Time: 08:49)

```

#include "mbed.h"

PwmOut motor (D3)           // D3 is set as a PWM controlled digital output
DigitalIn push_switch (D2)   // Switch output is connected to pin D2
int main()
{
    int state = 1;
    motor.period (0.10);
    motor = 1.0;           // Duty cycle = 1.0
    while(1) {
        if (push_switch == 0) { // Check for switch press
            state++;
            if (state == 4) state = 1;
        }
        while (push_switch == 0); // Wait for key release
        motor.write(1.0);
    }
}

```

আসুন প্রোগ্রাম কোডটি দেখুন যা কিছুটা দীর্ঘ এবং 2 টি স্লাইডে ছড়িয়ে রয়েছে।

যেমনটি আমি বলেছি, মোটর নিয়ন্ত্রণের জন্য আমরা PWM নিয়ন্ত্রণ আউটপুট লাইন D3 ব্যবহার করছি, ডিউটি সাইকেল(duty cycle) এর উপর নির্ভর করে মোটরের গতি পরিবর্তিত হবে। এবং পুশ(push) সুইচের আউটপুট D2 এর সাথে যুক্ত। এটি একটি DigitalIn টাইপ ইনপুট। মূল প্রোগ্রামে(main program), আমরা "state" নামক একটি পরিবর্তনশীল(variable) সংজ্ঞায়িত(define) করেছি, যার মানের সূচনা 1 এ হয়েছিল। এটি আসলে ইঙ্গিত করে যে মোটরটি কোন অবস্থায় রয়েছে, কারণ সেখানে 3 টি অবস্থা থাকতে পারে, সর্বোচ্চ গতি, মাঝারি গতি এবং বন্ধ।

আমি state = 1 শরু করছি, এটি আবার 1, 2, 3 এরকম চলবে। এখন, motor.period 0.1 এ সেট করা হয়েছে যার অর্থ 100 মিলিসেকেন্ড। সুতরাং, 100 মিলিসেকেন্ড সময়কালে, আমি প্রক্রিয়াটি পুনরাবৃত্তি করি। এবং প্রাথমিকভাবে আমি ডিউটি সাইকেলটি সর্বাধিক 1.0 তে সেট করেছি। এটি motor.write(1.0) এর সমান। সুতরাং, প্রাথমিকভাবে এটি সর্বোচ্চ গতিতে শুরু হচ্ছে।

এই while লুপে, আমরা একটি সুইচ টেপা হচ্ছে কিনা যাচাই করছি; যদি একটি সুইচ টেপা হয় তবে আপনি কীভাবে যাচাই করবেন? push_switch সিগন্যাল(signal) যদি 0 হয় তবে এর অর্থ সুইচ টেপা হয়েছে। যদি এটি 0 হয় তবে state 1 দ্বারা বাড়ানো হবে; তবে আমি এটি 4 হয়ে গেছে কিনা তা যাচাই করে দেখছি কারণ এটি কেবল 1, 2, 3 হতে পারে। সুতরাং, যদি এটি 4 হওয়ার চেষ্টা করে তবে আপনি আবার এটি 1 এ নিয়ে যান। সুতরাং এই লুপটিতে এটি 1, 2, 3 হবে আবার 1, 2, 3 এরকম লাগাতার চলবে।

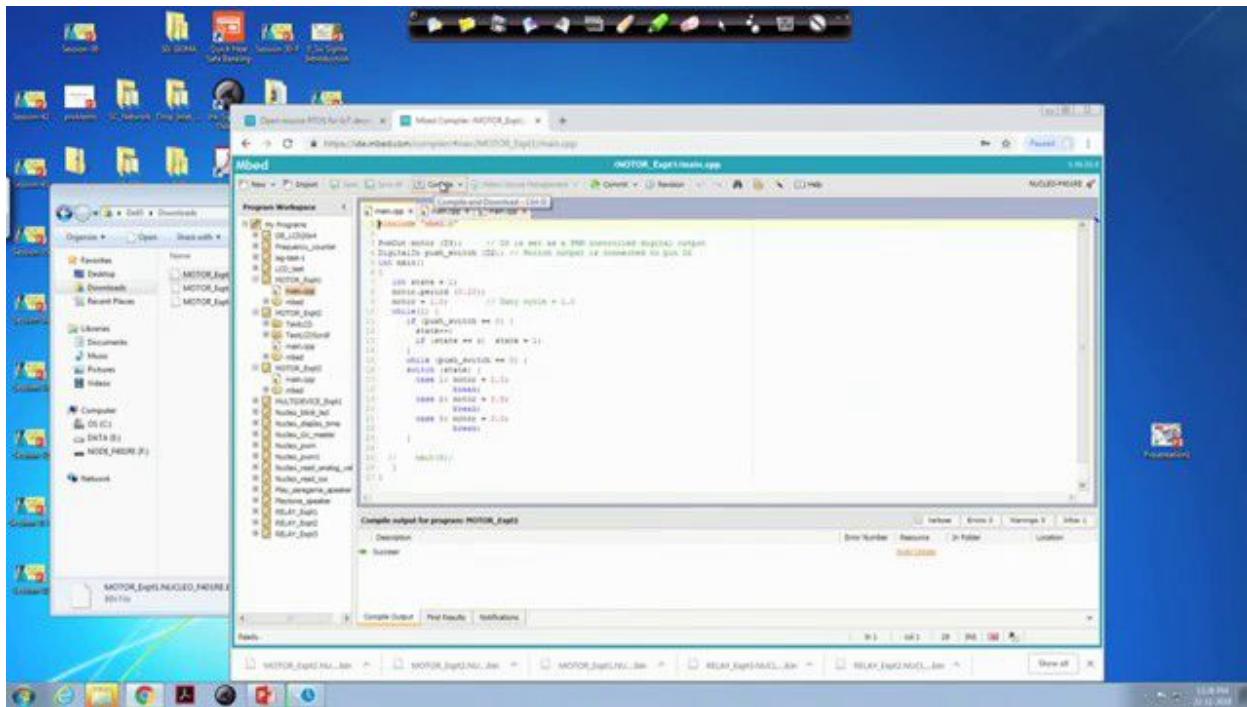
এবং এই সুইচ টেপাটি সন্তুষ্ট হওয়ার পরে আপনি state কে বর্ধিত করেছেন, আপনি সুইচটি না ছাড়া পর্যন্ত অপেক্ষা করুন। সুতরাং, যতক্ষণ সুইচ 0 তে থাকে, আমরা একটি ডামি লুপে(dummy loop) অপেক্ষা করি। সুতরাং, যদি আমরা সুইচটি দীর্ঘ সময়ের জন্য চেপে রাখি তবে এটি এই লুপটিতে অপেক্ষা করবে। আপনি এটি ছাড়ার পরেই কেবল পরবর্তী মোটর গতি কার্যকর হবে।

(Refer Slide Time: 11:48)

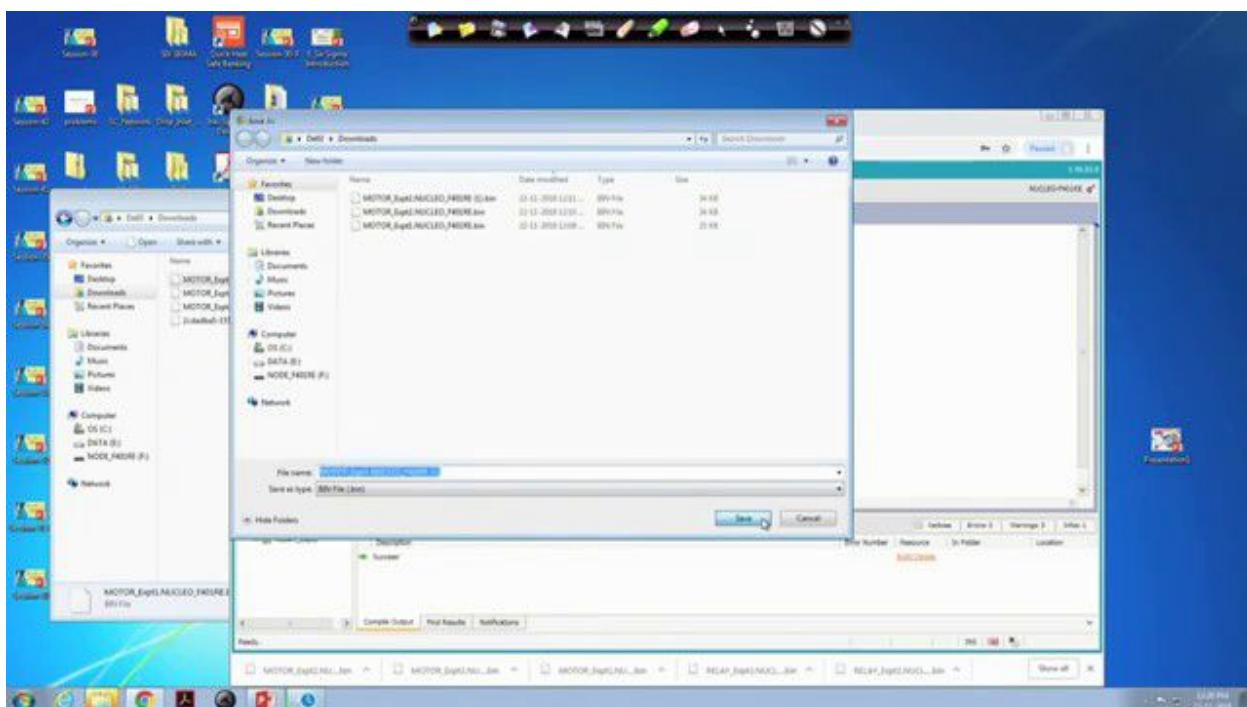
পরবর্তী স্লাইডটি আপনাকে দেখায় কীভাবে গতি নিয়ন্ত্রণ করা হয়। একটি switch case স্টেটমেন্ট(statement) আছে। একটি switch স্টেটমেন্ট(statement) রয়েছে, যা আসলে পরিবর্তনশীল(variavle) "state" এর মান যাচাই করে, এটি 1, 2, বা 3 হতে পারে এটি যদি 1 হয়, তবে আপনি সর্বাধিক ডিউটি সাইকেল(duty cylle) 1.0 ব্যবহার করছেন, যদি এটি হয় 2 তবে 0.8 বা মাঝারি গতি হয়, এবং যদি এটি 3 হয় তবে এটি 0.0 বা বন্ধ। এই জিনিসটি একটি চক্র চলে।

এখন আমরা প্রদর্শনীটি দেখব।

(Refer Slide Time: 12:46)

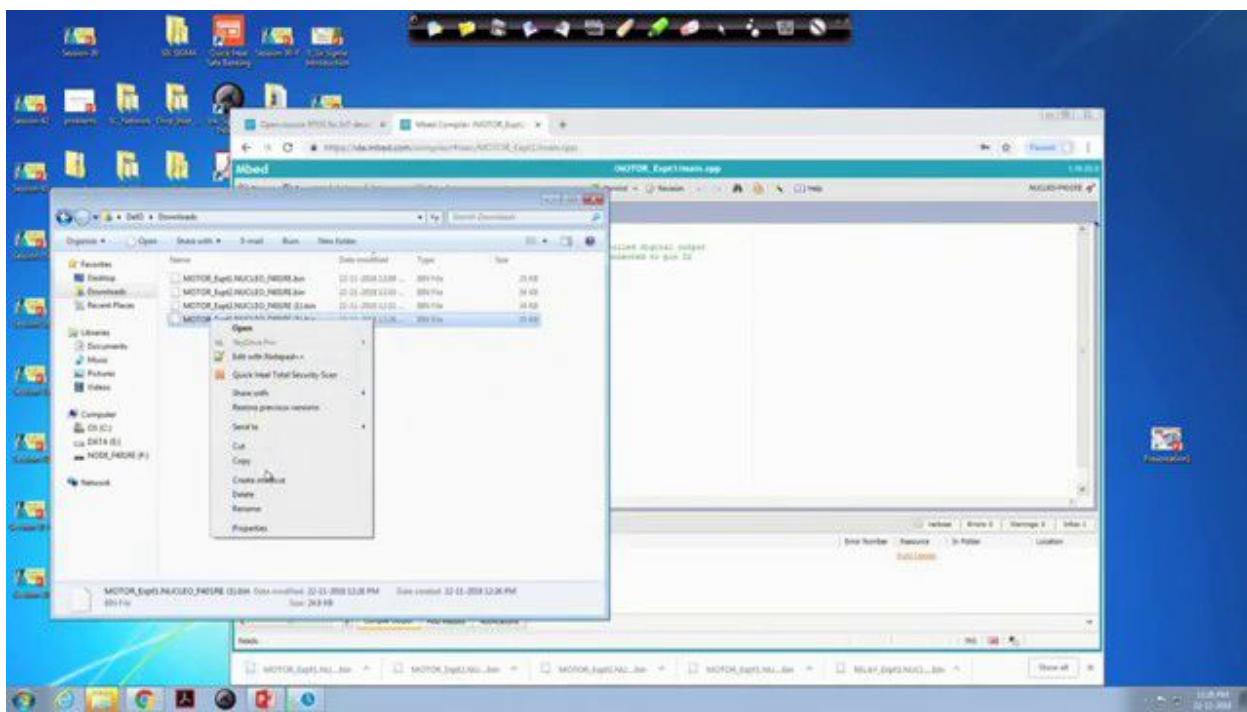


এটি সেই পরীক্ষা যেটি নিয়ে এখন আমরা আলোচনা করেছি, এটি সংকলন(compile) করি।
 (Refer Slide Time: 13:07)



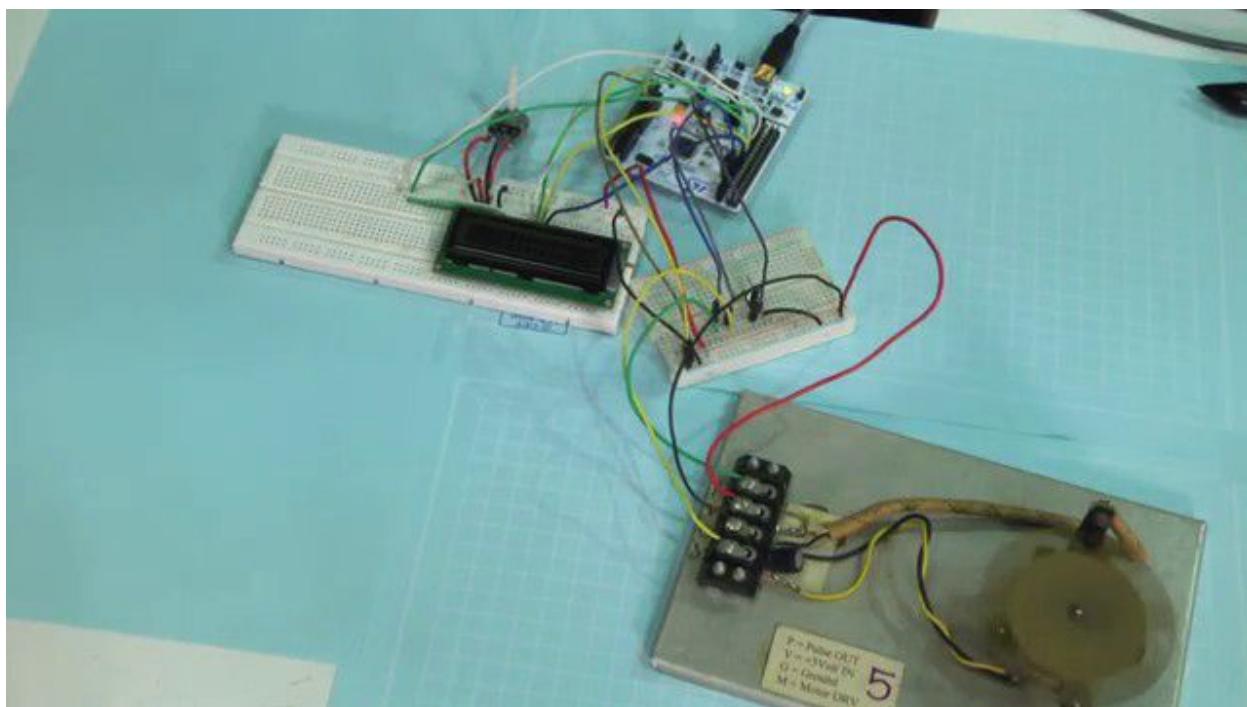
এটি সেভ(save) করুন।

(Refer Slide Time: 13:10)



কপি করুন এবং পেস্ট করুন।

(Refer Slide Time: 13:18)



এখন দেখুন, এই মোটরটি ঘূরতে শুরু করেছে। আপনি এখানে দেখতে পাবেন যে মোটরটি ঘূরছে এবং এটি সর্বাধিক গতি এবং এখানে আমি একটি ছোট পুশ সুইচ(push switch) ইন্টারফেস করেছি। আপনি পুশ সুইচ দেখতে পাচ্ছেন এবং পুশ সুইচটির একটি টার্মিনাল প্রতিরোধের মাধ্যমে Vcc এর সাথে সংযুক্ত এবং অন্য টার্মিনালটি গ্রাউন্ড এর সাথে। এবং সুইচটির মাঝের পয়েন্টটি ইনপুট লাইন D2 এর সাথে সংযুক্ত থাকে।

মোটরের ইন্টারফেস সম্পর্কে, আপনি দেখতে পাবেন যে কয়েকটি টার্মিনাল রয়েছে। এটি 5 ভোল্ট, যা 5 ভোল্ট পয়েন্টের সাথে যুক্ত, এই কালো তারে GND, এবং এই হলুদ তারটি PWM নিয়ন্ত্রণ(control), যা D3 এর সাথে সংযুক্ত রয়েছে। এবং এই সবুজটি একটি অন্য লাইন, এটি আমরা পরবর্তী পরীক্ষায় ব্যবহার করব, এটি অপটো-কাপলারের আউটপুটটি ব্যবহার করবে। তবে এই পরীক্ষায় আমরা অপটো-কাপলার ব্যবহার করছি না।

আপনি দেখছেন মোটর পুরো গতিতে ঘূরছে। যদি আমি এই সুইচটি একবার টিপি(push করি) তবে গতি হ্রাস পাই, এখন এটি মাঝারি। আমি আরও একবার টিপলে, এটি আবার সর্বাধিক গতিতে ঘূরতে শুরু করবে, তারপরে আমি আরও একবার মাঝারি গতিতে টিপব। এটি সর্বোচ্চ গতি, মাঝারি গতি এবং অফ(OFF)। এই চক্রটি পুনরাবৃত্তি হবে।

আমরা চালু রাখি।

(Refer Slide Time: 15:29)

Experiment 2

- Interface the motor and read the pulses generated by the rotary sensor. Count the pulses and display the RPM on a LCD display unit.

- The pulses generated by the optocoupler are fed to interrupt input D4.
- PWM control is used to drive the motor.
- LCD display unit is also interfaced.

এখন দ্বিতীয় পরীক্ষায় আমরা কিছু কাজ করছি। এখন আপনি মোটরটির কথা ভাবেন, আপনার এখানে মোটর আছে; এটি একটি মোটরের প্রতীক। এখন সাধারণত আমরা যা করেছি, আমরা গতি নিয়ন্ত্রণ করতে মাইক্রোকন্ট্রোলার থেকে একটি নিয়ন্ত্রণ লাইন(control line) ব্যবহার করছি। সুতরাং, আমরা PWM আউটপুট লাইন D3 এর সাথে সংযুক্ত করেছি।

এখন এই পরীক্ষায়, আমরা অপ্টো-কাপলারের আউটপুটও ব্যবহার করছি, কারণ আমরা ঘূর্ণনের গতি পরিমাপ করতে চাই, প্রতি সেকেন্ডে বা প্রতি মিনিটে কত স্পন্দন আসছে। অপটিক্যাল আউটপুটটি আমরা ডিজিটাল ইনপুট(input) লাইন D4 এর সাথে সংযোগ করেছি। এখন পয়েন্টটি গণনা করার পরে আমি কীভাবে জানব গতিটি কী এবং কীভাবে দেখাব?

এই কারণে, আমরা মাইক্রোকন্ট্রোলার বোর্ডের সাথে একটি LCD ইন্টারফেসও করেছি। আমরা LCD ইন্টারফেসের বর্ণনা দেখাচ্ছি না, কারণ আপনি ইতিমধ্যে এটি অধ্যয়ন করেছেন এবং পূর্ববর্তী কিছু বক্তৃতায় দেখেছেন। আপনি জানেন কীভাবে একটি LCD ইন্টারফেস করা হয়।

এখন আমি আপনাকে আরেকটি বিষয় ব্যাখ্যা করব যখন আমি আপনাকে প্রোগ্রামটি দেখাব।

(Refer Slide Time: 17:09)

Embed C Code – STM32

```

#include "mbed.h"
#include "TextLCD.h"
#include "TextLCDScroll.h"

PwmOut motor (D3); // D3 is set as a PWM controlled digital output
InterruptIn wheel (D4); // Optocoupler output is connected to pin D4

TextLCDScroll lcd(D13, D12, D11, D10, D9, D8, TextLCD::LCD16x2);

int pulses = 0;

void count()
{
    pulses = pulses + 1;
}

```

Diagram illustrating the interrupt setup: An interrupt (Interrupt) is triggered on pin D4 of the STM32 microcontroller, which then increments the 'pulses' variable in the 'count()' function. A waveform shows the digital signal on D4.

আমি ইতিমধ্যে বলেছি, এই mbed.h কে অন্তর্ভুক্ত করা হয় এবং এলসিডি(LCD) ইন্টারফেসের জন্য, আমাকে TextLCD.h ইন্টারফেস করতে হবে। অবশ্যই, আমাদের স্ক্রোলের(scroll) দরকার নেই, সুতরাং তৃতীয়টির প্রয়োজন নেই, তবে আমি TextLCDScroll.h ও ব্যবহার করেছি। এবং মোটর চালানোর জন্য, আমি D3 এর সাথে সংযুক্ত করেছি এবং D4 হ'ল ইনপুট(input) যা অপটো-কাপলারের আউটপুট(output) থেকে আসছে।

এখন, আসুন আমরা বোঝার চেষ্টা করি। আমি যখন কোনও প্রোগ্রাম লিখি, আমি কীভাবে গণনা করব? ধরা যাক, আমি গণনা করতে চাই, প্রতি সেকেন্ডে এই জাতীয় কত স্পন্দন আসছে। আমরা আমাদের প্রোগ্রামে এটি যেভাবে প্রয়োগ করেছি তা হ'ল আমরা বাধা দেওয়ার ধারণাটি ব্যবহার করেছি। বাধা কী? বাধা হ'ল এই জাতীয় প্রক্রিয়া; মনে করুন, এটি আমাদের প্রসেসর(processor) বোর্ড এবং বাইরে থেকে কিছু বাধা আসছে। এখানে, আমরা এটি ডিজিটাল ইনপুট লাইন D4 এর সাথে সংযুক্ত করেছি।

আমরা যখন আগে বাধা নিয়ে আলোচনা করেছি, আমি আপনাকে স্মরণ করিয়ে দিচ্ছি যে, আমি আপনাকে বলেছিলাম যে কোনও ডিজিটাল IO লাইন একটি বাধা পিন হিসাবে ব্যবহার করা যেতে পারে। সুতরাং, এখানে আমরা এই D4 টি একটি বাধা(interrupt) ইনপুট হিসাবে ব্যবহার করছি। আমি কীভাবে তা ঘোষণা করব? আমি এটিকে DigitalIn হিসাবে বলার পরিবর্তে, আমি এটি InterruptIn হিসাবে উল্লেখ করেছি

এবং এই বক্তুর নামটি আমি "while" হিসাবে দিয়েছি এবং পিডেলিউএম(PWM) এর জন্য আমি এটিকে "motor" বলছি।

এখন, মূল বক্তব্যটি হ'ল বাধার জন্য আসলে কী ঘটে? এর অর্থ, আপনি যদি এটি আগে কোথাও অধ্যয়ন করে থাকেন তবে আপনি জানেন যে কোনও প্রসেসরের কাছে যখন কোনও বাধা সংকেত আসে তখন প্রসেসর যা কিছু চালাচ্ছিল তা সাময়িকভাবে স্থগিত করা হয় এবং নিয়ন্ত্রণটি অন্য প্রোগ্রামের রুটিনে স্থানান্তরিত হয় যাকে বাধা পরিষেবা রুটিন(interrupt service routine) বলে। বাধা পরিষেবা রুটিন কার্যকর করার পরে, নিয়ন্ত্রণ স্থগিত হওয়া প্রোগ্রামে ফিরে আসে। এটা এভাবে কাজ করে।

এখন এক্ষেত্রে প্রতিবারই একটি করে অপটিক্যাল বাধা থাকবে ... একটি ঘূর্ণনের জন্য, এই জাতীয় 8 টি বাধা থাকবে। সুতরাং, এখানে এরকম কিছু স্পন্দন আসবে। প্রতিবার একটি অপটিক্যাল বাধা থাকবে, একটি স্পন্দন হবে এবং এই স্পন্দনগুলির প্রত্যেকটি একটি বাধা সংকেত তৈরি করবে(interrupt signal)। সুতরাং, বাধা পরিষেবা রুটিনটি অনেকবার কল(call) হবে।

(Refer Slide Time: 20:05)

Embed C Code – STM32

```

#include "mbed.h"
#include "TextLCD.h"
#include "TextLCDScroll.h"

PwmOut motor (D3);          // D3 is set as a PWM controlled digital output
InterruptIn wheel (D4);    // Optocoupler output is connected to pin D4

TextLCDScroll lcd(D13, D12, D11, D10, D9, D8, TextLCD::LCD16x2);
int pulses = 0;
void count()
{
    pulses = pulses + 1;
}

```

A man with glasses is visible on the right side of the screen, likely the speaker.

আমরা এখানে যা করেছি তা দেখুন। দেখুন, এটি আসলে বাধা পরিষেবার রুটিন, এটি গণনা ফাংশন, আমি আপনাকে দেখাব যে আমরা কীভাবে এটিকে একটি বাধা পরিষেবার রুটিন হিসাবে ঘোষণা করব। তবে, আপাতত ধরুন এটি বাধা পরিষেবা রুটিন, আমি যা করেছি: আমরা একটি গ্লোবাল(global) ইন্টিজার(integer) ভেরিয়েবল(variable) "pulses" নামে ঘোষণা করেছি, যা 0 তে সূচনা করা হয়েছে। এবং বাধা পরিষেবার রুটিনে আমরা কেবল "pulses" এর মান 1 বাড়িয়ে দিচ্ছি; যাতে যথনই বাধা আসে ভেরিয়েবলটি 1 দ্বারা বৃদ্ধি পায়।

অন্য লক্ষণীয় বিষয়টি হ'ল আমরা একটি এলসিডি(LCD) ইন্টারফেসও করেছি এবং এগুলি স্ট্যান্ডার্ড ইন্টারফেস(standard interface), আপনি স্মরণ করতে পারেন এর মধ্যে প্রথমটি রেজিস্ট্র সিলেক্ট(register select), দ্বিতীয়টি enable এবং শেষ 4 টি 4-বিট ইন্টারফেসের জন্য সর্বাধিক উল্লেখযোগ্য ডেটা(data) লাইন।

সুতরাং, আমরা পিনগুলি এলসিডি(LCD) থেকে D8, D9, D10, D11 এর সাথে এবং এই enable কে D12 এবং RS কে D13 এর সাথে সংযুক্ত করেছি। এবং আপনি স্মরণ করতে পারেন যে আমরা এখানে এমন একটি পোটেনিওমিটার(potentiometer) ব্যবহার করেছি যা আমরা এলসিডি(LCD) কনট্রাস্ট(contrast) বিন্যসের জন্য VEE নামে অন্য একটি পিনের সাথে সংযুক্ত করি। এই পোটেনিওমিটার সার্কিটটিও রয়েছে।

এখন আমি আপনাকে বলেছিলাম এটি হ'ল বাধা পরিষেবার রুটিন(service routine)
(Refer Slide Time: 21:53)

```

int main()
{
    int RPM; char msg[20];
    wheel.rise (&count);
    motor.period (0.10);
    motor = 0.9;
    lcd.cls();

    while (1) {
        wait (1.0);
        RPM = pulses * 60 / 8; // In a minute; 8 slots in the wheel
        sprintf (msg, "%6d", RPM);
        lcd.setLine (0, "The RPM is:");
        lcd.setLine (1, msg);
        pulses = 0;
    }
}

```

আসুন আমরা দেখি কীভাবে এই বাধা পরিষেবার রুটিনটি উল্লেখ করা হয়েছে। এটি আমাদের মূল ফাংশন(main function)। প্রথম জিনিসটি হ'ল আপনি এই জায়গাটি নোট করুন যেখানে আমরা বাধা পরিষেবার রুটিনটি নির্দিষ্ট করছি। আমরা বলছি যে চাকাটি সেই বস্তু, যা বাধাগ্রন্থ রেখার সাথে সংযুক্ত এবং সেটাকে আমরা "while" বলেছিলাম, rise একটি ফাংশন যা প্রতিবার সিগন্যাল 0 থেকে 1 পর্যন্ত বেড়ে গেলে তা যাচাই করে। যখনই কোনও বৃদ্ধি হয় আপনি এই ফাংশনটিকে কল করেন ; &count এর অর্থ এটি কোনও ফাংশনের পয়েন্টার। স্মরণ করুন সেই ফাংশনের নাম ছিল count, যতবারই চাকাটি উর্ঠতি প্রাণ্তে উপস্থিত হয় প্রত্যেকবার আপনি count কল করেন।

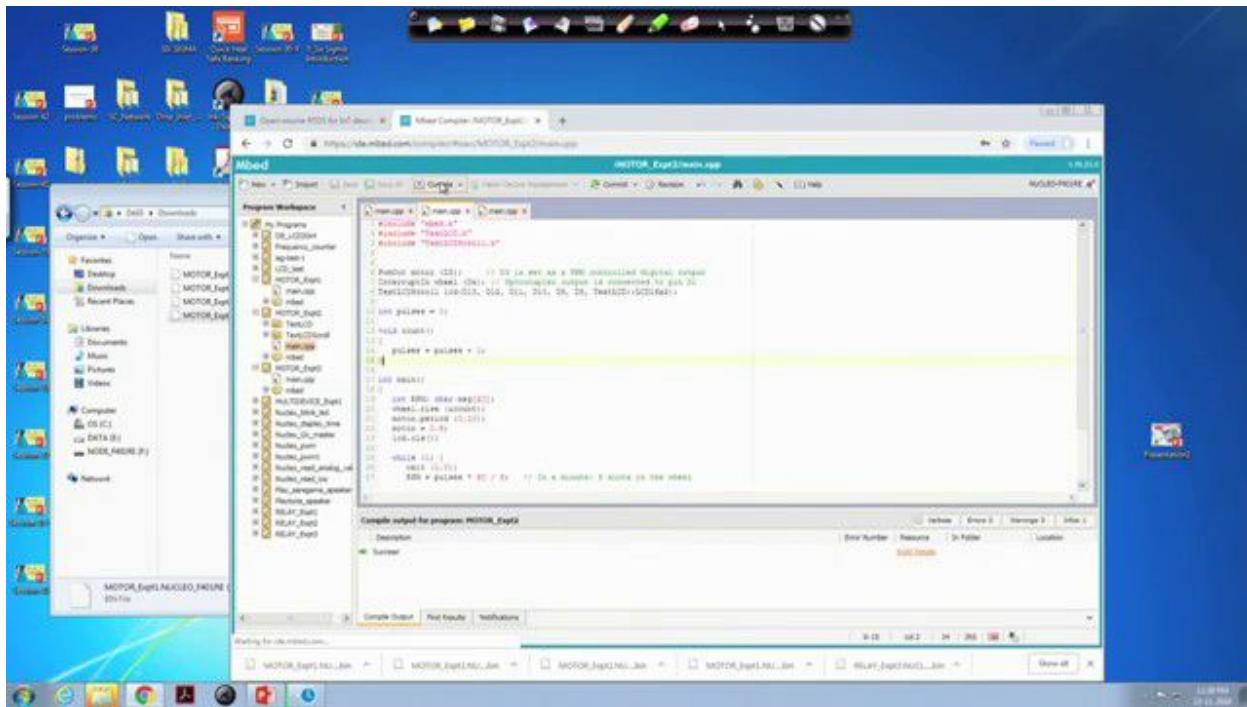
এভাবেই বাধা পরিষেবা রুটিনকে কল(call) করা হয় এবং আমরা RPM নামে একটি ভেরিয়েবল(variable) ঘোষণা করেছি, যেখানে আপনি প্রতি মিনিটে ঘূর্ণন গণনা করছেন। এখানে msg নামক একটি ক্যারেক্টার অ্যারে(character array) রয়েছে। মোটর ড্রাইভিংয়ের জন্য, আমরা একটি 100 মিলিসেকেন্ড সময়কাল ধরে নিয়েছি। পূর্ববর্তী পরীক্ষার মতোই ডিউটি সাইকেলটি 0.9 এ সেট করা হয়েছে এবং cls ফাংশন প্রাথমিকভাবে এলসিডি(LCD) স্ক্রিনটি সাফ করে।

while লুপের মধ্যে, প্রাথমিকভাবে pulses এর মান 0 থেকে শুরু হয়, এখন আমরা 1 সেকেন্ডের জন্য অপেক্ষা করি; এর অর্থ, এই 1 সেকেন্ডে মোটরটি ঘূরছে। সুতরাং, এই এক সেকেন্ডে কতগুলি স্পন্দন আসছে তা গণনা করা হবে; pulses এর মান ততবার বাড়বে। এর শেষে আমরা প্রতি মিনিটে ঘূর্ণন গণনা করি কারণ, আমরা 1 সেকেন্ডের জন্য গণনা করেছি, 1 মিনিটের জন্য, এটি 60 দ্বারা গুণিত হবে, এবং এই চক্রটিতে 8 টি গর্ত রয়েছে, তাই আমরা এটি 8 দ্বারা বিভাজন করব। এটি আপনাকে প্রতি মিনিটে ঘূর্ণন বা আরপিএম(RPM) মান দেবে।

এবং আমরা sprintf এর কী মানে করি, এর অর্থ আপনি এই মানটি একটি স্ট্রিংয়ে(string) মুদ্রণ করতে পারেন, আপনি এই RPM এর মানটি এই ক্যারেক্টার স্ট্রিংয়ের(character string) মধ্যে একটি পূর্ণসংখ্যা হিসাবে মুদ্রণ করছেন। তারপরে, আমরা এলসিডি(LCD) তে প্রদর্শন করছি।

এবং এটি করার পরে আমরা pulses এর মানটিকে পুনরায় 0 তে সূচনা করি যাতে পরবর্তী চক্রটিতে আমরা আবার গণনাটি সম্পাদন করতে পারি এবং আমরা এই প্রক্রিয়াটি পুনরাবৃত্তি করতে পারি। আবার 1 সেকেন্ড অপেক্ষা করুন, আবার স্পন্দনগুলি 1 সেকেন্ডের জন্য গণনা করা হবে, আবার আমরা RMP গণনা করব এবং এটি মুদ্রণ(print) করব। এটি প্রোগ্রাম(program), এবং এখন আমরা ডেমো(demo) দেখব।

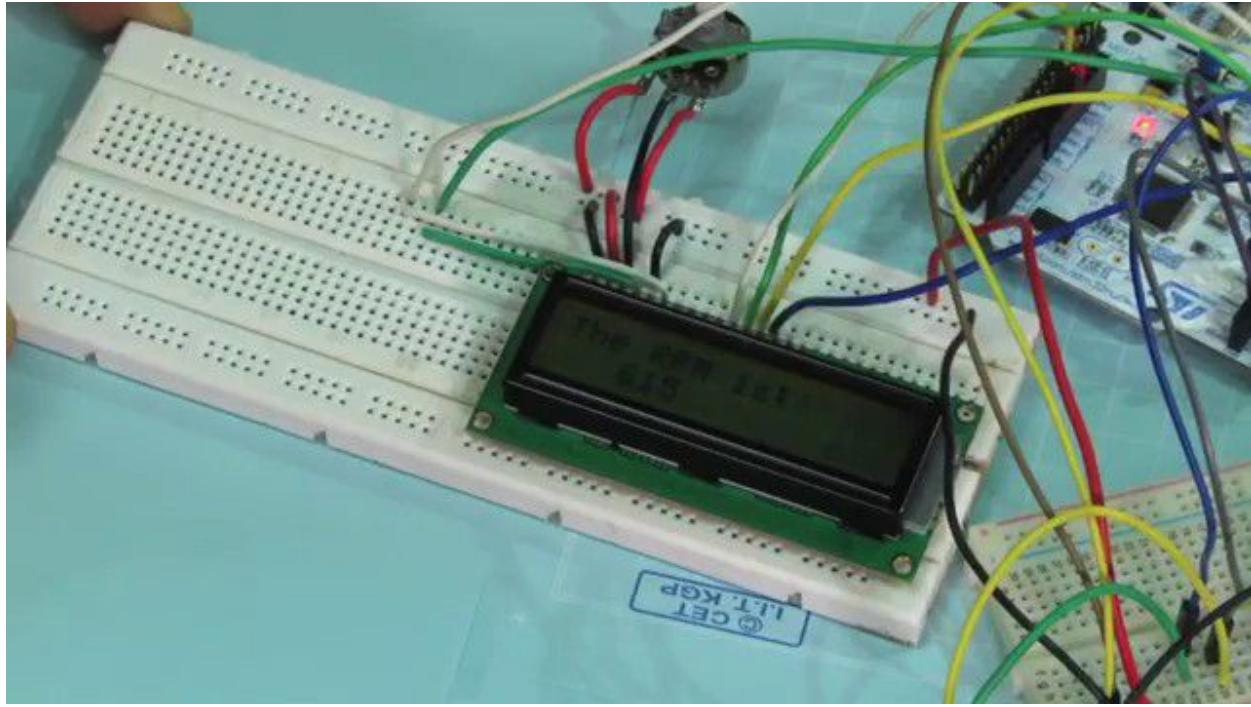
(Refer Slide Time: 25:14)



এটি আমি একই প্রোগ্রামটি দেখিয়েছি, এটি কম্পাইল(compile) করার চেষ্টা করছি।

সুতরাং আমি এটি কম্পাইল(compile) করব, এটি সেভ(save) করব, এটি কপি(copy) করব এবং এটি পেস্ট(paste) করব। এখন, আসুন দেখুন কি ঘটছে।

(Refer Slide Time: 25:50)



দেখুন, প্রোগ্রামটিতে আমরা ডিউটি চক্রটি(duty cycle) 0.9 এ সেট করেছি। সুতরাং, দেখুন মোটরটি খুব উচ্চ গতিতে, উচ্চ গতিতে ঘূরছে। এখন এই পরীক্ষায় সুইচ সার্কিট ব্যবহার করা হয়নি। সুতরাং, এই সুইচটি পড়া হয় না, তবে আপনি এই এলসিডি(LCD) ডিসপ্লে(display)টি এখানে দেখুন।

এটি হল কনট্রাস্ট(contrast) নিয়ন্ত্রণের জন্য পটেনশিওমিটার(potentiometer), তাই আপনি এটি উপযুক্ত কনট্রাস্ট(contrast) এর জন্য সামঞ্জস্য করতে পারেন। এটির RPM যদি 1095 দেখায় তবে মোটরটি খুব সম্ভা এবং ছোট মোটর, সুতরাং এর গতি খুব স্থিতিশীল নয়। সুতরাং, এটি কিছুটা পরিবর্তিত হচ্ছে, যেমন 1027, 1100, 1095। সুতরাং, RPM এভাবে চলেছে। আপনি এই RPM মানটি এখানে দেখতে পারেন।

এখন আসুন দেখি আমি ডিউটি চক্রটি(duty cycle) পরিবর্তন করলে RPM এর মান পরিবর্তন করে কিনা, আমাকে এটি দেখতে দিন। আপনি এখানে দেখুন, মোটর এর ডিউটি চক্রটি(duty cycle) 0.9 এ সেট করা হয়েছিল। আমি এটি 0.9 থেকে 0.7 এ পরিবর্তন করি, বলা যাক আমি এটি 0.7-এ পরিবর্তন করেছি। আমরা এটি সেভ(save) করি, আবার কম্পাইল(compile) করি এবং আমরা এটি সেভ করি, কপি করি, পেস্ট করি। সুতরাং এখন দেখুন একই জিনিসটি ঘটছে তবে এখন মোটরটি অনেক ধীর গতিতে ঘূরছে। আপনি যদি মোটরটি লক্ষ্য করেন তবে এটি একটি ধীর গতিতে ঘূরছে। এবং এখন, আপনি দেখতে পাচ্ছেন যে RPM এর প্রদর্শিত মান 600 দেখাচ্ছে। সুতরাং, আরপিএম মানটি উল্লেখযোগ্যভাবে হ্রাস পেয়েছে।

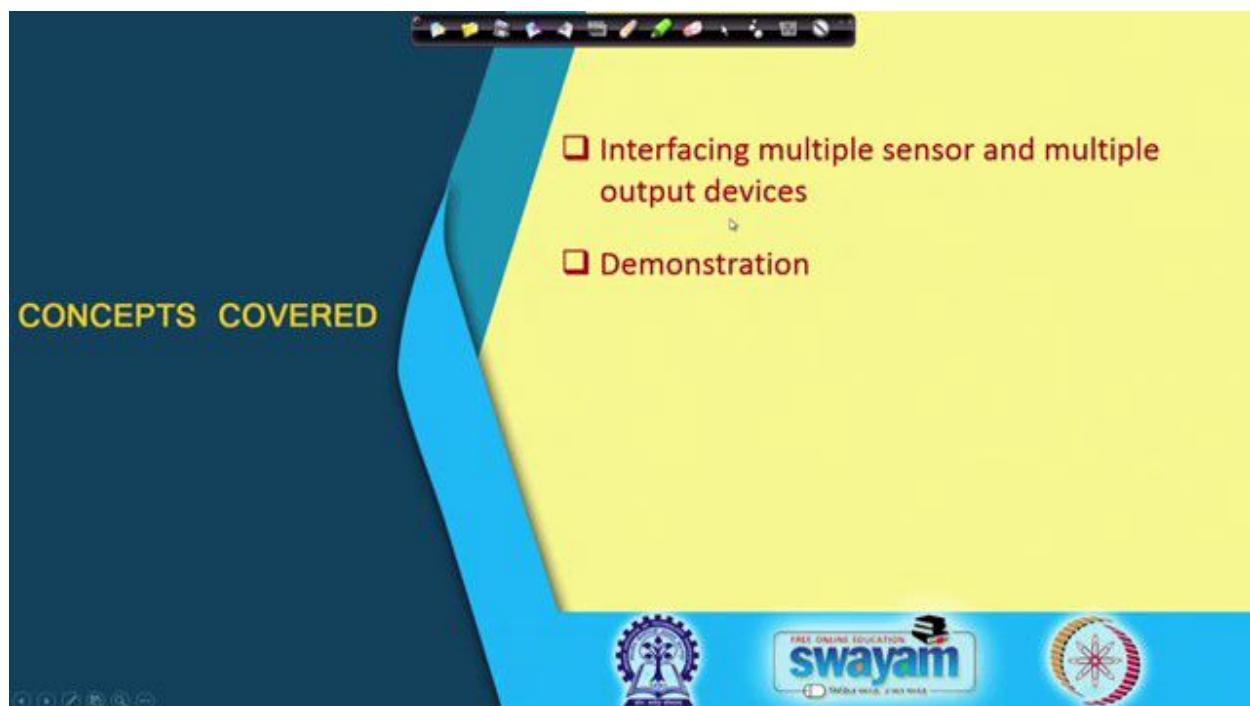
সুতরাং, এইভাবে আপনি গণনা করুন, মানটি পড়ুন এবং আপনি যদি দেখতে পান যে আরপিএম মানটি কম, আপনি ডিউটি সাইকেলটি কিছুটা বাড়িয়ে আবার পড়ুন; যদি এটি কম হয় তবে আপনি আবার পড়ুন। আগেই আমরা এই integral control, ON OFF control, PID control এই সমস্ত সম্পর্কে বলেছিলাম, তাই এই সমস্ত জিনিস এখানে ছবিতে আসতে পারে। তবে, যেহেতু প্রোগ্রামটি জটিল হয়ে উঠবে, তাই আমরা ওগুলিকে প্রদর্শনীতে রাখছি না, তবে আপনি ওগুলিও রাখতে পারেন; আপনি একটি প্রিসেট আরপিএম(preset RPM) মান সেট করতে পারেন এবং মোটর ড্রাইভকে সে অনুযায়ী কাজ করতে আপনি আরপিএম(RPM) সামঞ্জস্য করতে পারেন।

এর সাথেই আমরা বক্তৃতাটি শেষ করছি, ধন্যবাদ।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture – 33
Experiments with Multiple Sensors and Relay

আমরা যদি গত দুটি বক্তৃতার কথা স্মরণ করি তাতে আমরা দেখেছি যে আমরা কীভাবে রিলে(relay), ডিসি(DC) মোটর ইত্যাদি ডিভাইস(device) গুলিকে মাইক্রোকন্ট্রোলার(microcontroller) এর সাথে ইন্টারফেস(interface) করতে পারি, এবং আমরা কিছু প্রকার নিয়ন্ত্রণ করতে পারি এবং মানগুলিও বুঝতে পারি ইত্যাদি ইত্যাদি। এখন এই বক্তৃতার এই পরীক্ষায় আমরা আপনাকে দেখাবো যে, মাইক্রোকন্ট্রোলার কেবলমাত্র একটি ডিভাইস(device) বা একটি সেন্সর(sensor) নয়, এটি একই সাথে একাধিক ডিভাইস নিয়ন্ত্রণ করতে যথেষ্ট শক্তিশালী।

(Refer Slide Time: 01:05)



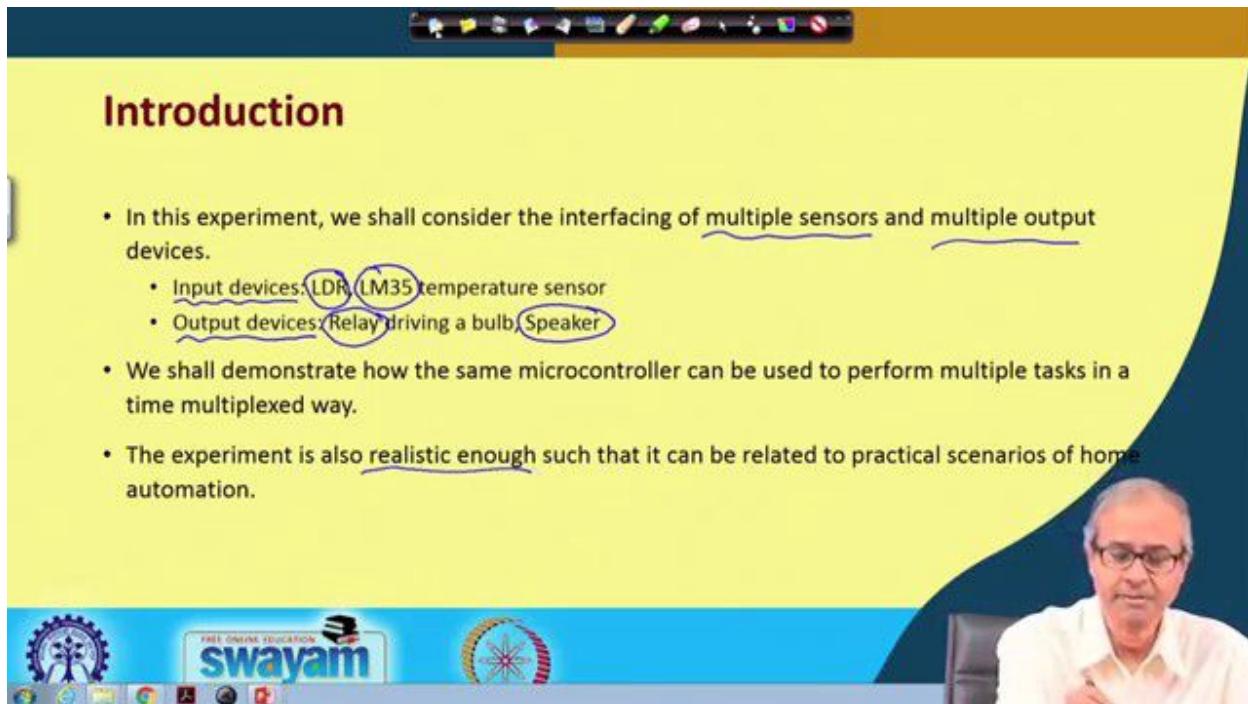
CONCEPTS COVERED

- Interfacing multiple sensor and multiple output devices
- Demonstration

SWAYAM

এখানে আমরা মূলতঃ একাধিক ডিভাইস, একাধিক সেন্সর এবং একাধিক আউটপুট ডিভাইসগুলি কীভাবে ইন্টারফেস করতে হবে সেদিকে লক্ষ্য করব।

(Refer Slide Time: 01:21)



Introduction

- In this experiment, we shall consider the interfacing of multiple sensors and multiple output devices.
 - Input devices: LDR, LM35 temperature sensor
 - Output devices: Relay, driving a bulb, Speaker
- We shall demonstrate how the same microcontroller can be used to perform multiple tasks in a time multiplexed way.
- The experiment is also realistic enough such that it can be related to practical scenarios of home automation.

আসুন প্রথমে পরীক্ষার বিষয়ে বলি। এই পরীক্ষায় যেমনটি বলা হয়েছে যে আমরা একাধিক সেন্সর এবং একাধিক আউটপুট ডিভাইসকে ইন্টারফেস করব; বিশেষত আমরা যে ইনপুট(input) ডিভাইসগুলিকে উল্লেখ করব সেগুলি হ'ল পারিপার্শ্বিক আলোতে সংবেদন করার জন্য একটি LDR এবং পরিবেশের তাপমাত্রায় সংবেদন করার জন্য একটি LM35 তাপমাত্রা সেন্সর(sensor)।

এখন, যে যে আউটপুট ডিভাইসগুলি আমরা ইন্টারফেসিং(interfacing) করব সেগুলি হ'ল একটি রিলে সার্কিট যা আবার একটি এলাইডি(LED) বাল্ব(bulb) কে চালনা করবে যা আমরা আগের বক্তৃতায় দেখেছি, এবং একটি স্পিকার(speaker)। এখন আমরা যে ধরণের পরিবেশ আপনাকে দেখানোর চেষ্টা করছি, প্রথমত মনে করুন এটি আমরা ইতিমধ্যে আপনাকে দেখিয়েছি যে পরিবেশের আলোর উপর নির্ভর করে আপনি একটি রিলে(relay) এর মাধ্যমে একটি বাল্বকে সুইচ অন-অফ(ON-OFF) করতে পারবেন।

এবং দ্বিতীয়ত, এখানে ফায়ার(fire) অ্যালার্ম(alarm) সিস্টেমের মতো কিছু থাকতে পারে, এমন একটি সিস্টেম থাকবে যা তাপমাত্রাটি সংবেদনশীল করবে এবং যখনই তাপমাত্রা একটি নির্দিষ্ট পূর্ব নির্ধারিত সীমাকে অতিক্রম করবে তখন অ্যালার্ম বাজবে এবং সেই অ্যালার্মের জন্য আমরা স্পিকারের সার্কিটটিকে ইন্টারফেস করেছি।

(Refer Slide Time: 03:03)

The Experiment

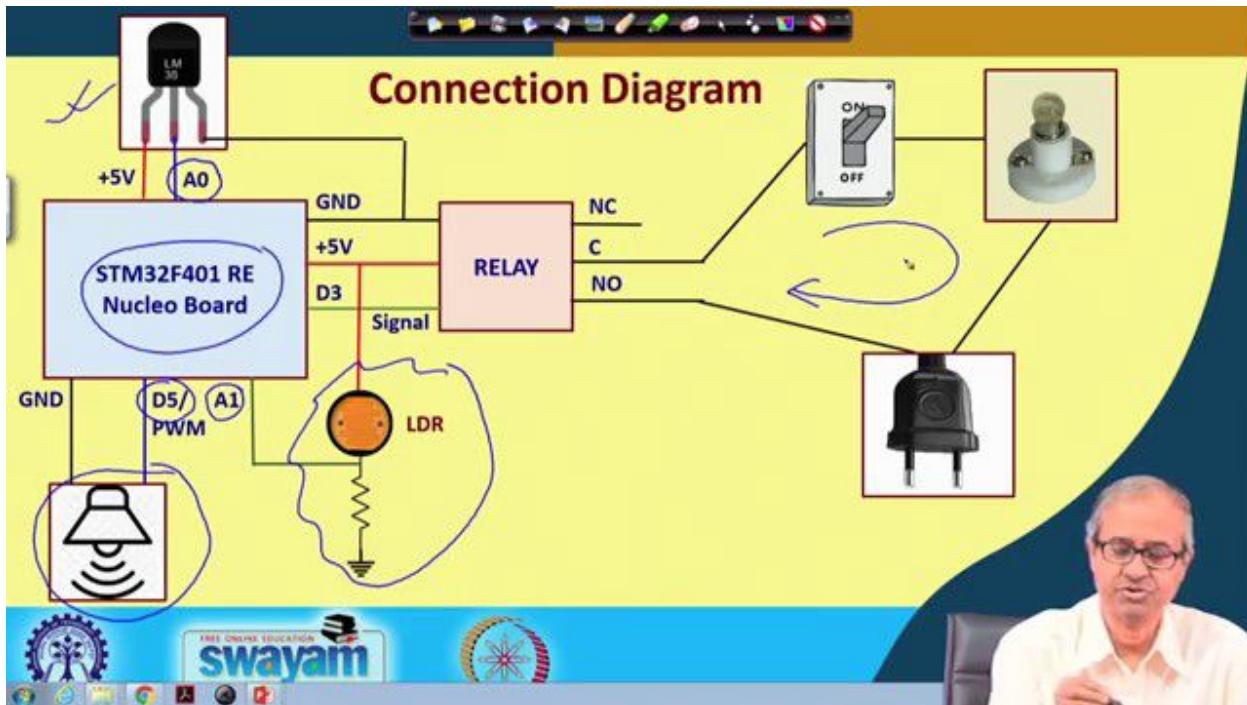
Interface relay, speaker, LDR, and LM35 to the STM32F401 Nucleo board:

- The relay circuit is used to drive a bulb (driven by D3) ✓
- The speaker is interfaced to generate an audible output (driven by D5 using PWM). ✓
- The LDR circuit is used to sense the level of ambient light (connected to A1). ✓
- The LM35 sensor is used to measure the temperature (connected to A0). ✓
- If the ambient light falls below a threshold, the bulb will turn on.
- Whenever the temperature crosses a threshold, an alarm will sound.

বিশেষত এই পরীক্ষায় আমরা একটি রিলে(relay), একটি স্পিকার(speaker), একটি LDR এবং LM35 আন্তঃসম্পর্কিত(interfacing) করব। রিলে ডিজিটাল পোর্ট লাইন D3 দ্বারা চালিত হবে। এই স্পিকারটি পোর্ট(port) লাইন D5 থেকে ইন্টারফেস(interface) করা হবে, LDR সার্কিট(circuit) আউটপুট(output) টি অ্যানালগ ইনপুট(analog input) A1 এর সাথে সংযুক্ত হবে এবং LM35 সেন্সরটি অ্যানালগ ইনপুট A0 এর সাথে সংযুক্ত হবে।

এখন, পরিবেষ্টিত আলোক ইনপুট যদি একটি সীমার নীচে নেমে যায়, বাস্তি চালু হবে। এছাড়াও, যখনই তাপমাত্রা একটি সীমাকে অতিক্রম করবে তখন অ্যালার্ম বাজবে।

(Refer Slide Time: 04:05)



আমরা সংযোগ ডায়াগ্রামটা দেখব। প্রথমতঃ এটি STM বোর্ড, এবং এটি রিলে(relay) সার্কিট যা আমরা আগে দেখেছি। একদিকে আমাদের কাছে LDR সার্কিট রয়েছে যা পরিবেষ্টনের আলোকে সংবেদনশীল করবে; একটি প্রতিরোধ ডিভাইডার(resistance divider) রয়েছে যার আউটপুটটি অ্যানালগ পোর্ট লাইন A1 এর সাথে সংযুক্ত। অন্যদিকে আপনার কাছে LM35 তাপমাত্রা সেন্সর রয়েছে যা অ্যানালগ ইনপুট লাইন A0 তে আউটপুট তৈরি করবে এবং অ্যালার্মের জন্য আপনার কাছে একটি স্পিকার রয়েছে যা D5 এর সাথে সংযুক্ত, যা একটি PWM নিয়ন্ত্রণ(control) আউটপুট(output) পোর্ট।

যখনই তাপমাত্রা একটি সীমাকে অতিক্রম করবে তখন স্পিকারটি বাজবে এবং যখনই আলো কোনও ঘরের গীচে নেমে আসে রিলে সক্রিয় হবে।

(Refer Slide Time: 05:07)

Basic Program Logic

- The steps as shown must run in a repetitive loop.
- There can be a delay before the loop repeats.
- First, the LDR generated sensor voltage is read through A0.
- Then, LM35 generated temperature data is read through A1.

```
begin
  if (value(A0) > thres_light)
    D3 = 0;                                // Turn OFF relay (bulb)
  else
    D3 = 1;                                // Turn ON relay (bulb)
  if (value(A1) > thres_temp)
    D5 = PWM tone;                         // Turn ON alarm
  else
    D5 = No PWM tone;                      // Turn OFF alarm
end
```

আসুন প্রথমে প্রোগ্রামটির লজিকটি দেখুন, তারপরে আমরা আপনাকে কোডটি প্রদর্শন করব। এখানে প্রদর্শিত পদক্ষেপগুলি পুনরাবৃত্ত লুপে(repetitive loop) চলমান হবে। প্রথমে আমাদের অ্যানালগ ইনপুট লাইন A0 এর মানটি পরীক্ষা করতে হবে, যদি এটি আলোকের প্রাণ্তিক স্তরটি অতিক্রম করে যাব অর্থ আমরা বোঝাতে চাইছি আপনি রিলে বন্ধ করে দিজ্জেন; এর অর্থ, আপনার পর্যাপ্ত আলো আছে অন্যথায় রিলে চালু করুন; তার মানে, বাষ্পটি প্রস্তুতি প্রস্তুতি হবে।

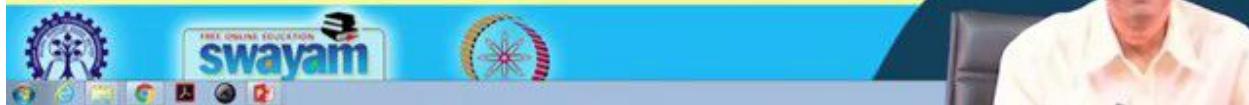
এর পরে আপনি তাপমাত্রার জন্য একই জিনিস পরীক্ষা করে দেখুন; যদি পোর্ট লাইন A1 এর মান প্রাণ্তিক মানের চেয়ে কিছু বেশি হয়, তবে আপনি স্পিকারে একটি টোন(tone) বাজান যা কিছু অ্যালার্ম(alarm) কে নির্দেশ করে, তবে এটি না হলে কোনও PWM টোন থাকবে না; এর অর্থ, আপনি কোনও টোন(tone) বাজান নি। এখন আসুন প্রোগ্রামটি দেখুন।

(Refer Slide Time: 06:13)

Mbed C Code for STM32F401

```
#include "mbed.h"
PwmOut bulb (D3); // Relay connected to PWM output D3
PwmOut speaker (D6); // Speaker connected to PWM output D5
AnalogIn ldr (A1); // LDR circuit output is connected to pin A1
AnalogIn lm35 (A0); // LM35 output is connected to pin A0

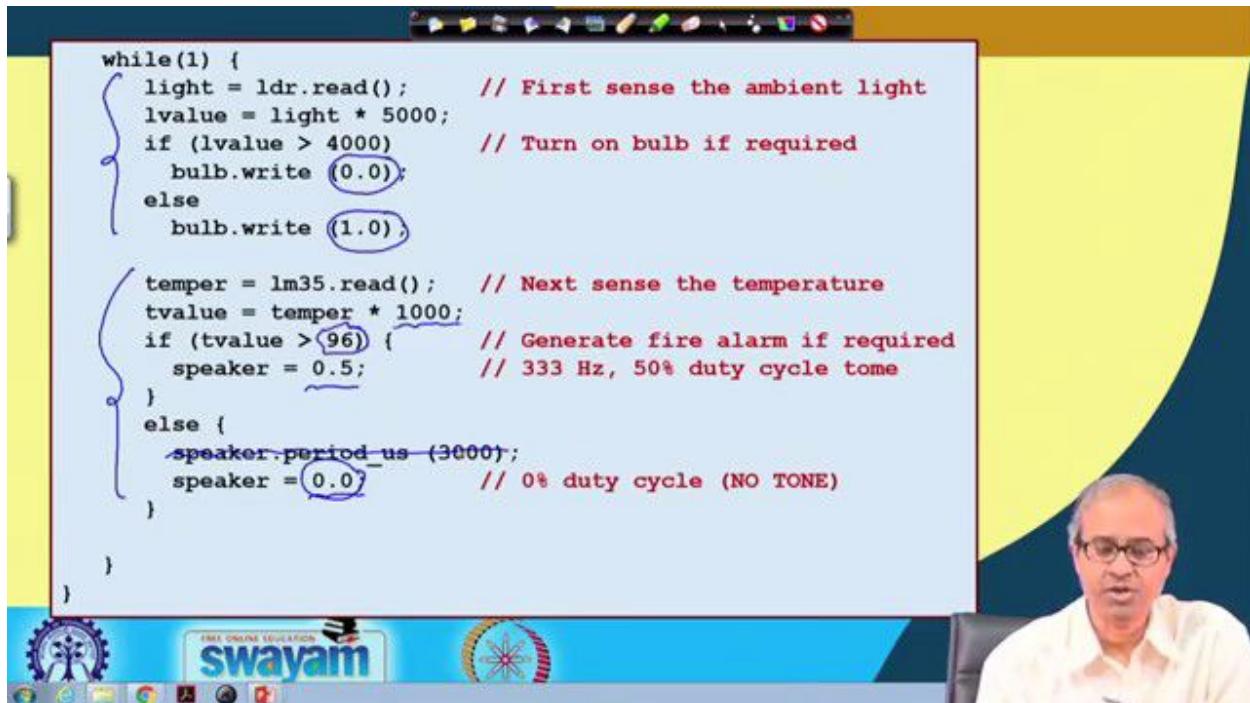
int main()
{
    float light, temper;
    int lvalue, tvalue;
    bulb.period (0.02); // 20ms
    speaker.period_ms (3); // 333Hz
```



সুতরাং, আপনি D3-তে একটি PwmOut লাইন সংজ্ঞায়িত(defined) করছেন, যাকে আপনি "bulb" বলছেন, তারপরে D6 এ অন্য একটি PwmOut রয়েছে, যাকে আপনি "speaker" বলছেন। এবং LDR এবং LM35 এর সাথে সম্পর্কিত দুটি অ্যানালগ ইনপুট A1 এবং A0 রয়েছে।

এবং আমরা কিছু ভেরিয়েবল(variable) সংজ্ঞায়িত করেছি, অ্যানালগ ইনপুট(analog input) গণনা করার জন্য light,temper ভেরিয়েবল(variable) এবং অস্থায়ী তাপমাত্রার মান সংরক্ষণ করার জন্য lvalue এবং tvalue ভেরিয়েবল দুটি। এবং আমরা ধরে নিয়েছি যে, বাস্তু সক্রিয় করার জন্য আমরা 20 মিলিসেকেন্ড সময়কাল এর 50 হার্টজ(Hertz) PWM ব্যবহার করি। এবং স্পিকারের জন্য আমরা 3 মিলিসেকেন্ডের সময়সীমা নির্ধারণ করেছি যার অর্থ স্পিকারে 333 হার্টজ ফ্রিকোয়েন্সি(frequency) বাজানো হবে।

(Refer Slide Time: 07:41)

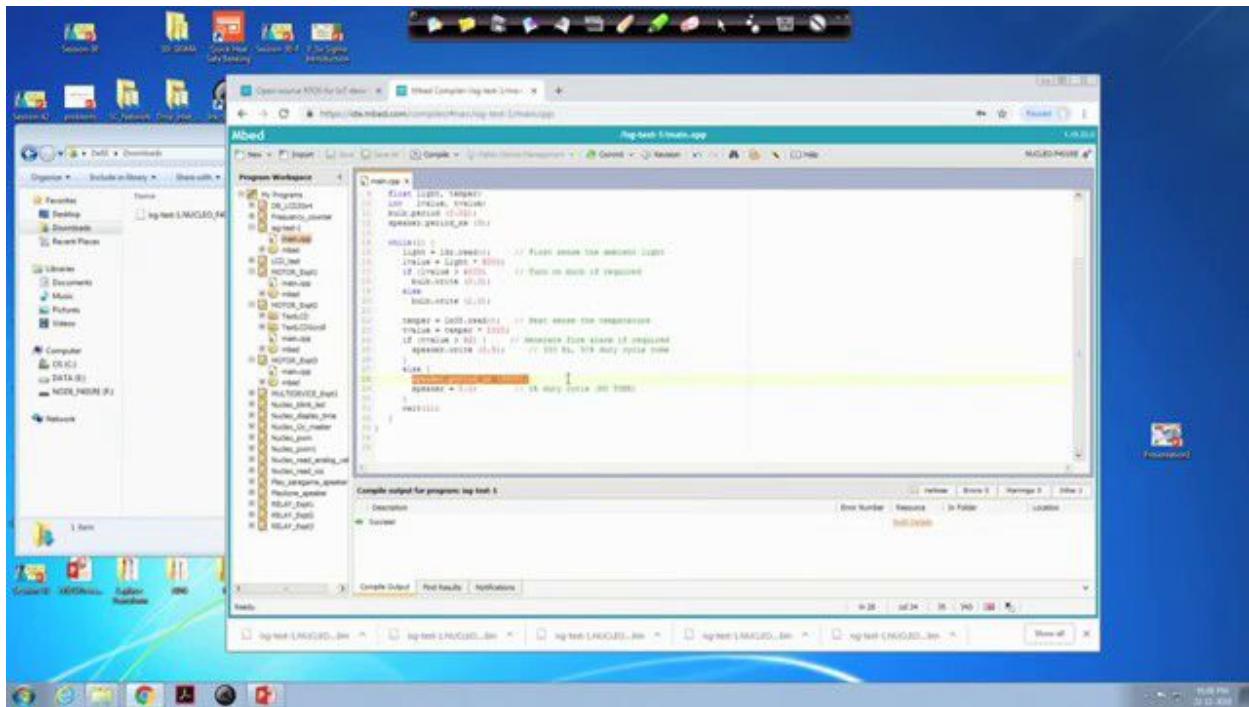


তারপরে আসি মূল ফাংশনের বাকি অংশে। পুনরাবৃত্তি while লুপে আমরা এখানে আলো পরীক্ষা করছি, আমরা এখানে তাপমাত্রা পরীক্ষা করছি। আলোতে আমরা অ্যানালগ ইনপুট ফাংশন ldr.read () ব্যবহার করছি, তারপরে আমরা পূর্বের প্রোগ্রামটি যেমন দেখেছিলাম ঠিক তেমন একটি স্কেল ফ্যাক্টর(scale factor) দ্বারা আপনি এটি গুণ করছেন। এটি যদি প্রাণ্তিক মানটি কিছুটা অতিক্রম করে তবে আমরা আলোটি বন্ধ করে দেব; যদি তা না হয়, আমরা আলোটি চালু করে দেব।

ডিউটি সাইকেল(duty cycle) টি 0.0 বা 1.0 এ সেট করে এটি করা হয়। একইভাবে আপনি যে তাপমাত্রাটি পড়ছেন সেটির জন্য আপনি এখানে কিছু স্কেল ফ্যাক্টর দ্বারা এটি আবার গুণ করছেন। আমি 1000 ব্যবহার করছি এবং তাপমাত্রার উপর নির্ভর করে আপনি যে অ্যালার্মটি শুরু করতে চান সেখানে আবার পরীক্ষার মাধ্যমে কিছু প্রাণ্তিক মান বেছে নেওয়া যেতে পারে। তদনুসারে আমরা স্পিকারে একটি অবিচ্ছিন্ন টোন(tone) বাজাইছি, তবে অন্যথায় আমরা ডিউটি সাইকেল(duty cycle) টি 0 তে সেট করি।

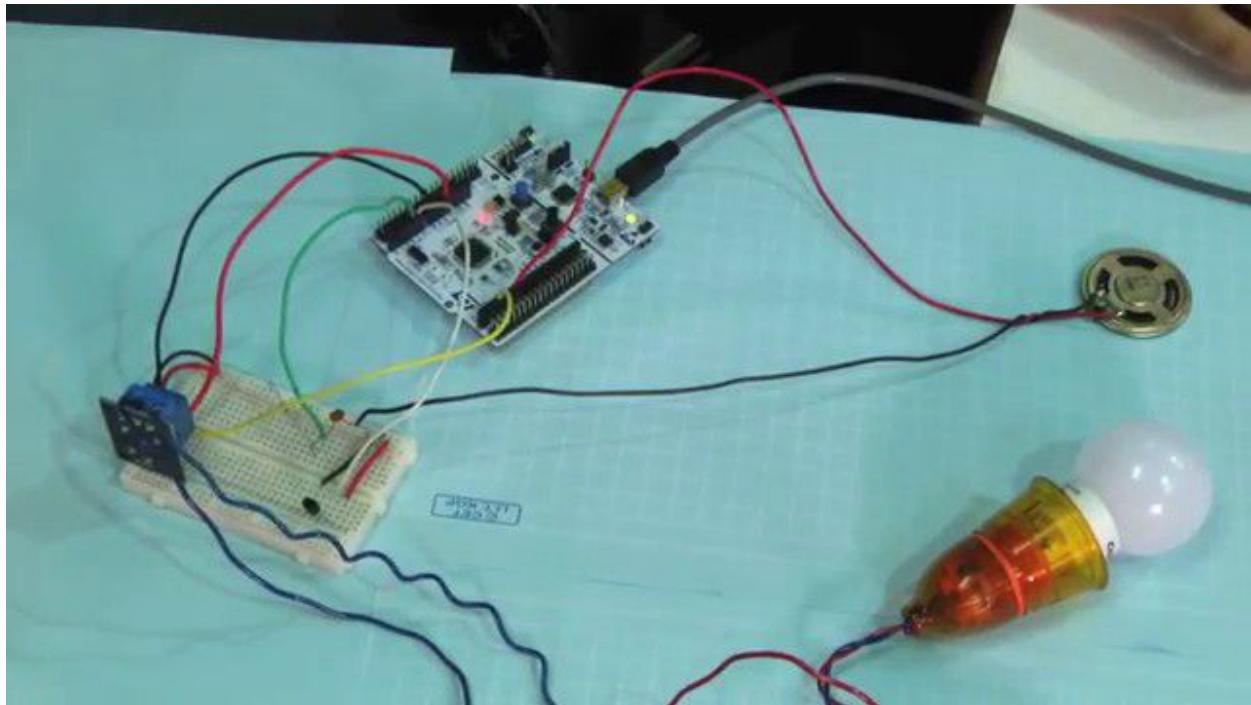
এখন, এই লাইনটির আসলে কোনও প্রয়োজন নেই আপনি এটি বাদ দিতে পারেন, তবে এটিও করার প্রয়োজন হয় না, কারণ একবার আপনি ডিউটি সাইকেল(duty cycle) টি 0 তে নির্ধারণ করে দিলে পিরিয়ড(period) এর জন্য কিছু যায় আসে না ঠিক আছে। আসুন এখন প্রদর্শনীটি দেখুন।

(Refer Slide Time: 09:47)



এই লাইনটি আপনি বাদ দিতে পারেন আমাদের এটির দরকার নেই। সুতরাং, আসুন এই কোডটি সংকলন(compile) করুন। আমরা এটি কম্পাইল(compile) করি, আমরা এটি সেভ(save) করি, কপি(copy) করি এবং পেস্ট(paste) করি।

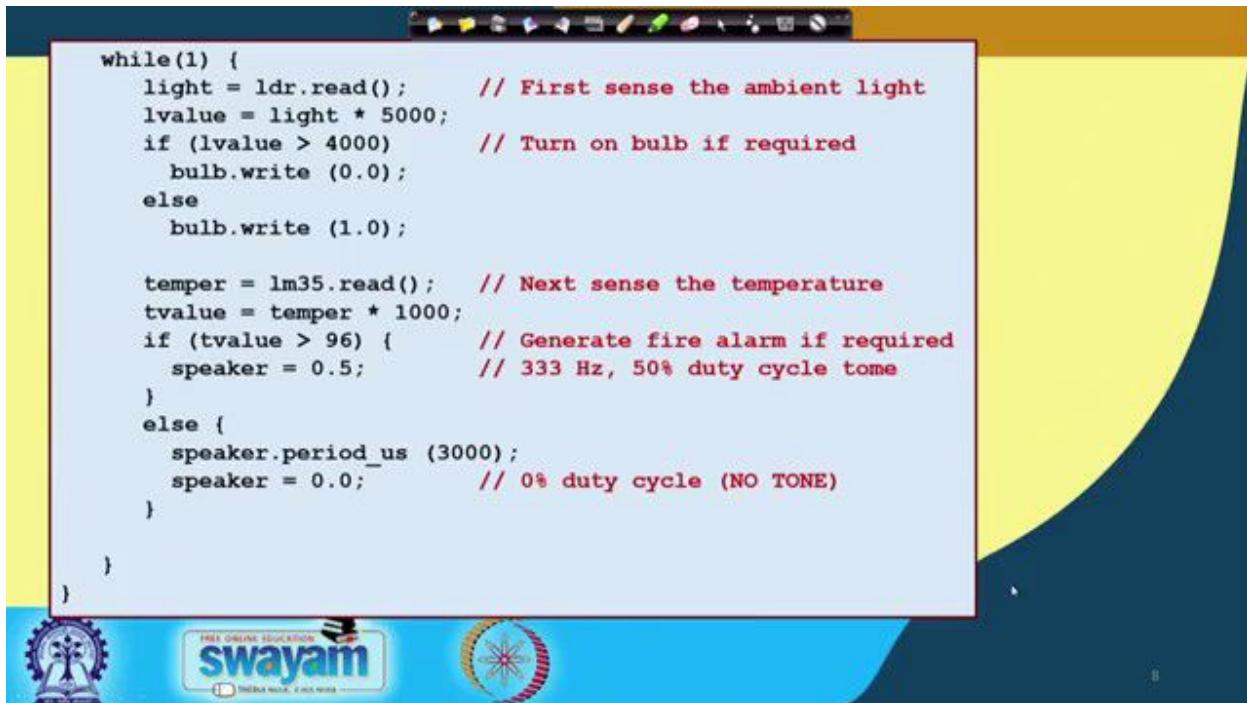
(Refer Slide Time: 10:21)



এখন আসুন সার্কিটে আসা যাক। এখন দেখুন আগে যেমন দেখানো হয়েছিল আমরা সেই একই আলোক ইন্টারফেসিং(interfacing) বর্তনী(circuitry) টি ব্যবহার করেছি; একটি রিলে, একটি বাল্ব একটি স্বিচ(switch) এর মাধ্যমে মেইনের(main) সাথে সংযুক্ত এবং আমরা একই LDR সার্কিট ব্যবহার করেছি। আপনি দেখতে পাচ্ছেন যে LDR যদি বাধাপ্রাপ্ত হয়, আলো অ্বলবে। আপনি দেখতে পাবেন আবার এটি ছেড়ে দিলে আলো বন্ধ হয়ে যাবে। এখানে আপনার কাছে LM35 সেটিং রয়েছে; আপনি দেখবেন তাপমাত্রা প্রাপ্তিকের চেয়ে বেশি হওয়া মাত্রই এখানে যে অ্যালার্মটি আছে সেটি বাজতে শুরু করবে।

আমি আমার আঙুল দিয়ে টিপতেই(just pressing) তাপমাত্রাটি থানিকটা বেড়ে যায়। আপনার কাছে যদি একটি হিটার থাকে তবে আপনি এটি খুব পরিষ্কার ভাবে দেখাতে পাবেন। দেখুন, তাপমাত্রার সীমাটি অতিক্রম করা মাত্রই অ্যালার্মটি শোনা যাচ্ছে, তবে আমরা যদি এটি আরও কিছুটা গরম করতে পারি তবে খুব স্পষ্ট শব্দ আসবে। সুতরাং, এই পরীক্ষায় আমরা যা দেখেছি তা হ'ল আমরা একাধিক ডিভাইস(device) ইন্টারফেস(interface) করতে পারি।

(Refer Slide Time: 12:15)



```
while(1) {
    light = ldr.read();      // First sense the ambient light
    lvalue = light * 5000;
    if (lvalue > 4000)      // Turn on bulb if required
        bulb.write (0.0);
    else
        bulb.write (1.0);

    temper = lm35.read();   // Next sense the temperature
    tvalue = temper * 1000;
    if (tvalue > 96) {      // Generate fire alarm if required
        speaker = 0.5;      // 333 Hz, 50% duty cycle tone
    }
    else {
        speaker.period_us (3000);
        speaker = 0.0;      // 0% duty cycle (NO TONE)
    }
}
```

আপনি যখন আপনার বাড়ির হোম অটোমেশন(home automation) সম্পর্কে বলেন, সেখানে অনেক ধরণের ডিভাইস সংযুক্ত থাকতে পারে। এখন এই মাইক্রোকন্ট্রোলারগুলি যথেষ্ট শক্তিশালী, যেমন একটি একক ডিভাইস পুরো জিনিসটি নিয়ন্ত্রণ করতে সক্ষম হবে। কেবলমাত্র আপনার প্রোগ্রামটি এমনভাবে লিখতে হবে যে কোনও কোনও ডিভাইস ইনপুট(input) পাঠাতে পারবে এবং বাধা চালিত মোডে কিছু ডিভাইস সেগুলিকে একে একে পড়তে পারবে।

এটি ডিভাইসের ধরণের এবং তারা আপনাকে কীভাবে ইনপুট প্রেরণ করছে তার উপর নির্ভর করে। আমরা প্রবর্তী বক্তৃতাগুলিতে এগুলির আরও প্রদর্শনী দেখব, যেখানে আপনি কীভাবে ডিভাইসগুলি বাইরের বিশ্বের সাথে যোগাযোগ করতে পারবে তাও দেখবেন।

ধন্যবাদ।

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya
Lecture - 34
Introduction to Internet of Things

এই সপ্তাহে, ইন্টারনেট অফ থিংস(Internet of Things) নামক একটি বিষয়ের সাথে আমি তোমাদের পরিচয় করবো | অবশ্যই, কিছু উদাহরণের সাহায্যে ইন্টারনেট অফ থিংস (Internet of Things) জিনিসটি কি তা আমি তোমাদের সাথে আলোচনা করব | তারপর দুটি উদাহরণ নেব আমরা | তার মধ্যে একটি হলো এক স্বয়ংক্রিয় গৃহ পরিচালনার ব্যবস্থা, যেখানে আমরা এসএমএস(SMS)- এর মাধ্যমে একটি বাতিকে জ্বালাবো ও বন্ধ করব |

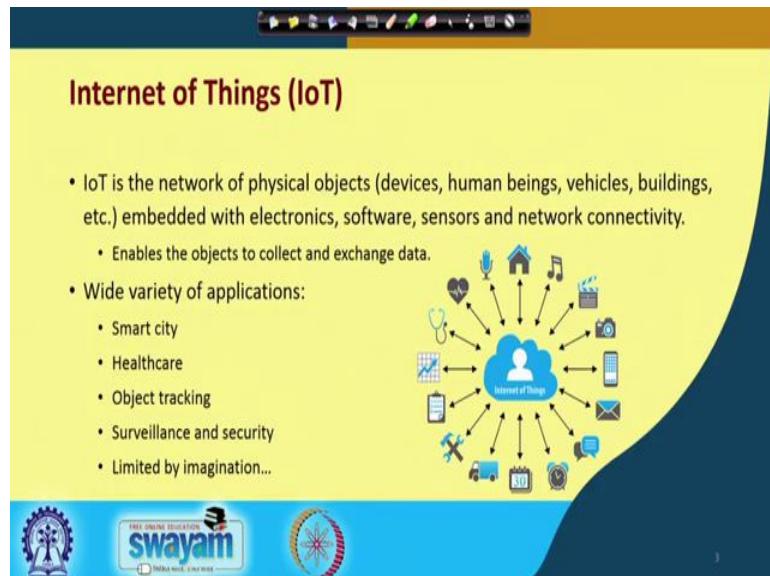
এবং এই সপ্তাহে আমরা তোমাদেরকে একটি টাচ সেন্সর(touch Sensor) -ও দেখাবো, যে কিভাবে কোন অযাচিত ব্যক্তি তোমার স্ক্রিন(Screen)- এ স্পর্শ করলে, সেটির সাহায্যে আমরা তা নির্ণয় করতে পারব |সুতরাং, এই দুটি উদাহরণই আমরা দেখব। এবং বিশেষ করে আজকের আলোচনায়, আমি ইন্টারনেট অফ থিংস (Internet of Things)- এর সাথে তোমাদের পরিচয় করাব।

(Refer Slide Time: 01:29)



এই সপ্তাহে যে যে বিষয়গুলি আমরা দেখব সেগুলি হল ইন্টারনেট অফ থিংস (Internet of Things) জিনিসটি কি, আইওটি(IoT) -র সাথে এমবেডেড সিস্টেমস(embedded systems) কিভাবে সম্পর্কিত এবং আমরা অবজেক্ট ট্র্যাকিং(object tracking)- এর একটি ঘটনা পর্যবেক্ষণ করব।

(Refer Slide Time: 01:49)



ইন্টারনেট অফ থিংস (Internet of Things) কি সেই আলোচনায় ফিরলে, আমরা দেখি এটি হচ্ছে বিভিন্ন বস্তু বিশেষের মধ্যে একপ্রকার নেটওয়ার্ক(network), যেমন কিনা এটি কোন ডিভাইস(device) হতে পারে, এটি কোন মানুষ কিংবা গাড়ি হতে পারে, বা কোন বাতি হতে পারে, যার মধ্যে বৈদ্যুতিন, সফটওয়্যার(software), সেন্সর(sensor) এবং নেটওয়ার্ক (network)সংযোগ ব্যবস্থা গ্রথিত আছে এবং এর সাহায্যে বস্তুটি কি করতে পারে, নাওর সাহায্যে বস্তুটি তথ্য সংগ্রহ ও তথ্য আদান-প্রদানে সক্ষম হয়। আমাকে একটা উদাহরণ দিতে দাও।

পঞ্চম সপ্তাহে, আমরা তাপমাত্রা মাপার মডিউল(Module) নিয়ে আলোচনা করেছিলাম। তো, সেই উদাহরণটিতে আমরা করেছিলাম কি যে, এলএম35(LM35) নামক একটি সেন্সর(sensor) আছে যেখান থেকে আমরা ভৌত স্থিতিমাপ তথা তাপমাত্রা মাপেছিলাম এবং তা এলসিডি(LCD)- র মাধ্যমে দেখাচ্ছিলাম। এই কাজটাই আমরা করেছিলাম, অর্থাৎ একটি এমবেডেড সিস্টেম(embedded system) প্রযুক্তি বস্তু, যা কিনা একটি একক নির্ভরশীল সিস্টেম(System)। এটির যোগাযোগ স্থাপনের ক্ষমতা আছে, কিন্তু আমরা সেন্সর(sensor) থেকে যে ডেটা(Data) পাচ্ছিলাম তা কোথাও কোন সার্ভার(server)- এ বা অন্য কোন

তথ্যভান্ডারে পাঠাচ্ছিলাম নাতো, আমরা সেই তাপমাত্রাটাকে কোথাও সংরক্ষণ করছিলাম না আমরা শুধুমাত্রই এলসিডি(LCD) -র মাধ্যমে দেখাচ্ছিলাম।

কিন্তু, ধরা যাক এমন একটি পরিস্থিতি, যেখানে আমি মাইক্রোকন্ট্রোলার(microcontroller) এর মাধ্যমে সেন্সর(Sensor) থেকে তাপমাত্রা সংগ্রহ করলাম এবং সেই তথ্যটা আমি পাঠিয়ে দিলাম অথবা কোন তথ্যভান্ডার-এ সংরক্ষণ করে রাখলাম |যদি আমরা এটি কোন অনলাইন(online) তথ্যভান্ডারে সংযোগ করে রাখতে চাই, তাহলে এর প্রক্রিয়াকরণ ক্ষমতার পাশাপাশি যোগাযোগ সাধনের ক্ষমতাও থাকা দরকার। ইতিমধ্যেই প্রক্রিয়াকরণ ক্ষমতা এর আছে কিন্তু এখন কোথাও ডেটা(Data) পাঠাবার জন্য অথবা ডেটা(Data) কোন সার্ভার(Server) কিংবা তথ্যভান্ডার-এ সংরক্ষণ করে রাখার জন্য কিছু যোগাযোগ স্থাপনের ক্ষমতাও আমাদের প্রয়োজন।

যখনই একটি এমবেডেড সিস্টেম(embedded system) - এ প্রক্রিয়াকরণ ক্ষমতার পাশাপাশি যোগাযোগ সাধনের ক্ষমতাও থাকে, সেটি একটি আইওটি(IoT) ডিভাইস(device)- এ পরিণত হয়। তার মানে এই যে, তাপমাত্রা মাপার যন্ত্রটি আর স্বতন্ত্র রাইল না এবং এই নির্দিষ্ট জায়গাতে সীমাবদ্ধ রাইল না, যেখানে আমরা শুধুমাত্র তাপমাত্রা দেখাচ্ছিলাম। এখন যদি কেউ এই ঘরের তাপমাত্রা জানতে চায়, তাহলে আমার দেখানো পদ্ধতির মাধ্যমে তোমরা তা পেতে পারো।এটি যে কোন বস্তুকে ডেটা(Data) সংগ্রহ ও ডেটা(Data) আদান-প্রদানে সক্ষম করে তোলে।

আমরা বিভিন্ন জায়গায় এটিকে প্রয়োগ করতে পারি, যার মধ্যে একটি হচ্ছে স্মার্ট শহর(Smart City), অপর একটি হচ্ছে স্বাস্থ্যব্যবস্থা।এর একটি বড় প্রয়োগ হল অবজেক্ট ট্র্যাকিং(object tracking) যাতে আজকাল লোকজন ছোট মডিউল(Module) ব্যবহার করে যে কোন বস্তুর গতিবিধিতে লক্ষ্য রাখছে, আমরা এটার উপর পরে আলাদা ভাবে আলোচনা করবানজরদারি ও নিরাপত্তা ব্যবস্থার ক্ষেত্রে আইওটি(IoT) খুবই প্রযোজ্য। নিরাপত্তা ব্যবস্থায় এর প্রয়োগ দেখানোর জন্য, এই সেন্সর(Sensor)-টিকে ভিত্তি করেই আমি একটা উদাহরণ নেব, কিন্তু এর আরো অনেক রকম প্রয়োগস্কেত্র আছে। এবং অবশ্যই, তা শুধুমাত্রই আমাদের কল্পনাশক্তি দ্বারা সীমিত, তা না হলে এটির ব্যবহারের বিপুল পরিধি আছে।

যখন একটা এমবেডেড সিস্টেম(embedded system)-এর যোগাযোগ স্থাপনের ক্ষমতাও থাকে, আমরা তাকে আইওটি(IoT) বলি।

(Refer Slide Time: 06:13)

IoT and Embedded Systems

- IoT constitutes the larger picture.
 - Billions of “things” generate data and communicate among each other.
- Each IoT node is essentially an embedded system.
 - IoT revolution has given embedded systems a new lease of life.
- Some typical applications leading to embedded system design:
 - Smart home
 - Object tracking
 - Smart transportation
 - Leading to smart cities → smart world

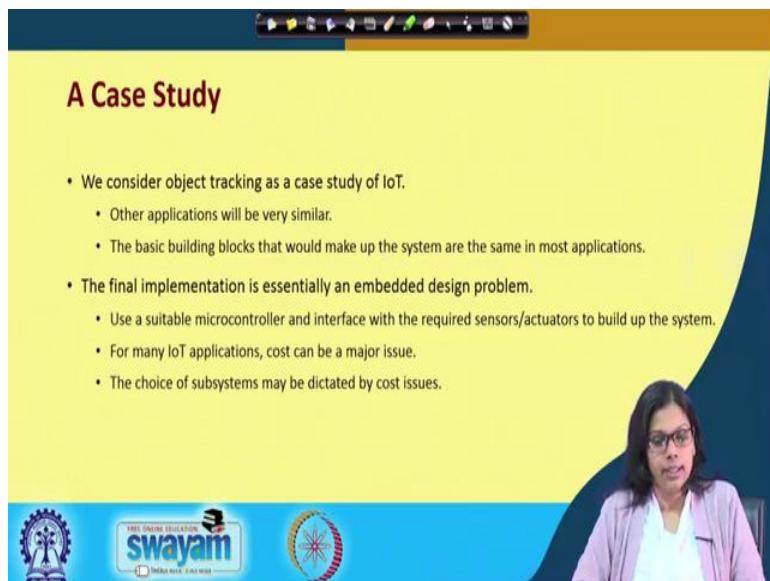
এবার আইওটি(IoT) এবং এমবেডেড সিস্টেম(embedded system) সম্পর্কে বলা যাক, ইতিমধ্যেই এদের দুজনের মধ্যে কি সম্পর্ক তা আমি বলেছি। বৃহত্তর চিত্রটি আইওটি(IoT)- কে ঘিরেই আবর্তিত, যদিও এটি কোন একটা এমবেডেড সিস্টেম (embedded system)নয়, এটি হচ্ছে আরো বহু এমবেডেড সিস্টেমের (embedded system) সমন্বয় যারা একে অপরের সাথে বাত্তালাপ করতে পারে।

এই বিশ্বে এমন কোটি কোটি বস্তু আছে যারা ডেটা(Data) সৃষ্টি করতে পারে ও একে অপরের সাথে যোগাযোগ করতে পারে। এই বস্তুগুলি সেন্সর(Sensor) থাকে বিপুল পরিমাণ ডেটা(Data)-র সৃষ্টি করে। এবং সেই ডেটা(Data)পরবর্তী প্রক্রিয়াকরণের জন্য কোথাও একটা সংরক্ষন করা থাকে। এত বিপুল পরিমাণ ডেটা(Data)কে সংরক্ষণ করা খুব একটা সহজ ব্যাপার নয়। অবশ্যই তোমার কাছে পরিকাঠামো উপস্থিত থাকতে হবে এই ডেটা(Data)কে সঠিকভাবে সংরক্ষণ করার জন্য। কিন্তু কিছু কিছু ক্ষেত্রে যদি সেন্সর গুলি অযথা ডেটা(Data) সৃষ্টি করতে থাকে, এবং তুমি যদি জানো সেই ডেটা(Data) সর্বসময়ের জন্য সংরক্ষণ করতে হবে না তুমি সহজেই সেই ডেটা(Data)মুছে দিতে পারো।

আমি ইতিমধ্যেই বলেছি যে প্রত্যেকটি আইওটি(IoT) নোড(Node) বস্তুত একটি এমবেডেড সিস্টেম(embedded system)। এবং আইওটি(IoT)- র এই জনপ্রিয়তা এমবেডেড সিস্টেম(embedded system) -এর দুনিয়ায় যেন নতুন প্রাণের সঞ্চার করেছে। এমবেডেড সিস্টেম(embedded system) কিন্তু নতুন কিছু নয়, কিন্তু এই চলতি শব্দ আইওটি (IoT)বেশ কয়েক বছর হলো আছে বলা যেতে পারে বিগত পাঁচ বছর ধরে। বলা যেতে পারে যে আইওটি (IoT)বিশ্বে পুরনো এমবেডেড সিস্টেম(embedded system) গুলিকে আবার সজাগ করে তুলেছে। এই ধাঁচের কিছু প্রয়োগ ব্যবস্থা যা কিনা পরে এমবেডেড

সিস্টেম(embedded system) এর নকশায় পরিণত হতে পারে তা হল স্মার্ট গৃহ(Smart Home), অবজেক্ট ট্র্যাকিং(object tracking),স্মার্ট পরিবহন ব্যবস্থা(Smart Transportation System), এবং অবশ্যই আইওটি(IoT) -র বিভিন্ন দিক গুলি কে মিলিত করে যে বৃহত্তর ছবিটি আমাদের চোখের সামনে ভেসে ওঠে তা হল একটি স্মার্ট শহর(Smart City), এবং অবশেষে এটি পরিণত হবে স্মার্ট দুনিয়ায়(Smart World)|

(Refer Slide Time: 08:27)



A Case Study

- We consider object tracking as a case study of IoT.
- Other applications will be very similar.
- The basic building blocks that would make up the system are the same in most applications.
- The final implementation is essentially an embedded design problem.
 - Use a suitable microcontroller and interface with the required sensors/actuators to build up the system.
 - For many IoT applications, cost can be a major issue.
 - The choice of subsystems may be dictated by cost issues.

এবার আমরা ঘটনা পর্যবেক্ষণ করবো যাতে একটা অবজেক্ট ট্র্যাকিং(object tracking) মডিউল(Module) কে ভিত্তি করে চলবো। প্রথমত আমি বলব যে অবজেক্ট ট্র্যাকিং(object tracking) বলতে আসলে আমরা কি বুঝি। আমি একটা বস্তু হতে পারি, আমার ব্যাগ একটা বস্তু হতে পারে ধরা যাক আমি এক জায়গা থেকে আরেক জায়গায় কিছু জিনিস যেমন ওষুধ পাঠাঞ্চি সেটাও একটা বস্তু হতে পারে।

আমার বাচ্চারা বাইরে থেলতে যাচ্ছে তাদেরকেও বস্তু হিসেবে ধরা যেতে পারে। আরো এরকম ধরনের প্রয়োগ আছে। আমরা এই সিস্টেম(System) গঠনকালের প্রাথমিক ধাপগুলি দেখব এবং বেশিরভাগ ক্ষেত্রে এই ধাপগুলি একই রকম থাকে।

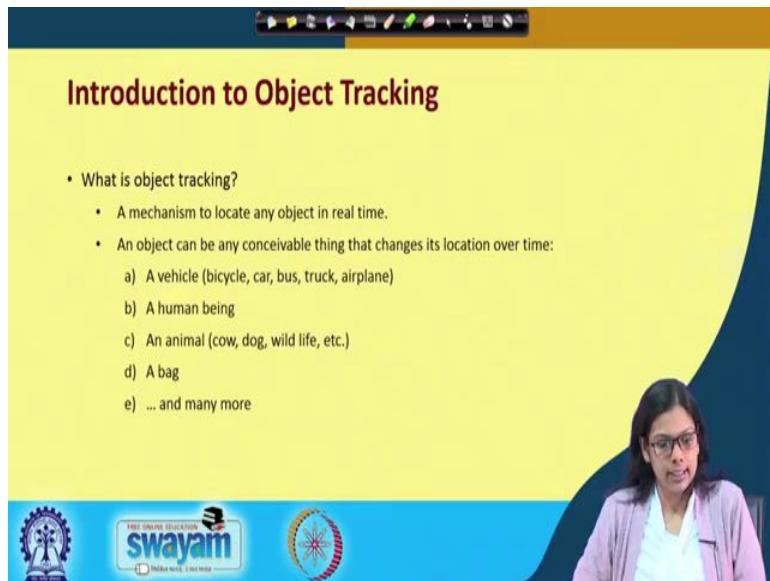
অন্তিম রূপায়নটি আসলে নির্ভর করছে একটি এমবেডেড সিস্টেম(embedded system) -এর উপরাতোমরা যদি অবজেক্ট ট্র্যাকিং(object tracking)- এর সমগ্র ব্যবস্থাটার দিকে দেখো, তাহলে দেখবে যে সবার প্রথমে তোমাকে একটা এমবেডেড সিস্টেম(embedded system) বানাতে হবো এবং তারপর অবশ্যই সেটির যোগাযোগ সাধনের ক্ষমতা থাকতে হবে।

প্রথম ধাপে আমরা একটি উপযুক্ত মাইক্রোকন্ট্রোলার(Microcontroller)- এর সাথে প্রয়োজনমতো বিভিন্ন সেন্সর(Sensor) অথবা একচুয়াটর(Actuator) ব্যবহার করে সিস্টেম-টাকে তৈরি করবাতো আমাদের একটা উপযুক্ত মাইক্রোকন্ট্রোলার(microcontroller) দরকার প্রক্রিয়াকরণের জন্য এবং সমগ্র সিস্টেম(System)-টাকে তৈরি করতে আরো কিছু সেন্সর(Sensor) অথবা একচুয়াটর (Actuatorসত্ত্ব) এর প্রয়োজন।

আইওটি(IoT)- র ক্ষেত্রে বহু সময় খরচ একটা বড় বাধা হয়ে দাঁড়াতে পারে। এখন এই সমস্ত ব্যাপারগুলি আমরা আগের সপ্তাহ গুলিতেই আলোচনা করেছি যে শেষ পর্যন্ত কতগুলি একক তুমি তৈরি করছ তার ওপর খরচ নির্ভর করছে। যদি তুমি সত্তিই খুব অল্প সংখ্যক তৈরি করছ তাহলে খরচ বেশী হয়ে যেতে পারে। কিন্তু ওই একই জিনিস তুমি বেশি সংখ্যক এককের জন্য তৈরি করলে তুমি সেটা বৃহৎ পরিমাণে করছো সুতরাং খরচ সব সময়ই কম হবে।

যেকোনো সাব- সিস্টেম(sub-systems) খরচ দ্বারা নির্ধারিত হতে পারে। এটা নির্ভর করে যে কতগুলি এককের জন্য তুমি বানাচ্ছ , কোন সম্প্রদায়ের মধ্যে ডিভাইস(Device)- টির চাহিদা আছে, তুমি কিভাবে সেটাকে বাজারে নিয়ে আসবে যাতে তার গুরুত্ব বিপুল সংখ্যক খদ্দেরের কাছে পৌঁছে যায়। যদি খদ্দের কিনতে চায় তাহলে তোমাকেও নানাভাবে বিপণন করতে হবো সুতরাং কি কি জিনিস তোমার করা উচিত তোমার উৎপন্ন দ্রব্যটিকে বিক্রি করার জন্য, তাতে আরো অনেক জিনিস জড়িয়ে আছে কিন্তু খরচ একটা গুরুত্বপূর্ণ দিক।

(Refer Slide Time: 12:07)



Introduction to Object Tracking

- What is object tracking?
 - A mechanism to locate any object in real time.
 - An object can be any conceivable thing that changes its location over time:
 - a) A vehicle (bicycle, car, bus, truck, airplane)
 - b) A human being
 - c) An animal (cow, dog, wild life, etc.)
 - d) A bag
 - e) ... and many more

এবাব অবজেক্ট ট্র্যাকিং(object tracking)- এ ফেরত আসা যাক। প্রথমত অবজেক্ট ট্র্যাকিং(object tracking)কি? এটা একটা কৌশল যার সাহায্যে প্রকৃত সময়ে কোন বস্তুর অবস্থান নির্ণয় করতে পারি মানে ধরা যাক আমার বাচ্চা বাইরে খেলতে গেছে কোন এক সময় আমি জানতে চাই আমার বাচ্চাটি কোথায় খেলছে, সে কি খেলার মাঠে আছে, না বাইরে অন্য কোথাও আছে তা প্রকৃত সময়ে আমি জানতে চাই। এই প্রক্রিয়ায় আমাদের কাছে এক ধরনের অ্যাপ্লিকেশন থাকতে পারে যেমন কোন ওয়েব অ্যাপ্লিকেশন(Web Application) অথবা মোবাইল ফোনের(Mobile Phone) কোন অ্যাপ(App) যে কিনা আমার বাচ্চার বর্তমান অবস্থান বলে দেবে। কি কি করার প্রয়োজন আছে, সেসব আমি এক এক করে বলে দেবো, কিন্তু এই একটা ক্ষেত্র যেখানে এই অবজেক্ট ট্র্যাকার(Object tracker) টির প্রয়োজনীয়তা আছে আমার কাছে।

সময়ের সাথে সাথে নিজের অবস্থান পরিবর্তন করতে পারে এমন বৈশিষ্ট্যযুক্ত যেকোনো অনুমেয় জিনিস বস্তু হতে পারে হতে পারে সেটা কোন গাড়ি যেমন আমার বাইসাইকেল, আমার গাড়ি অথবা একটা বাস(Bus) কিংবা ট্রাক(Truck) কিংবা উড়োজাহাজ কিংবা কোন মানুষ হতে পারে যেরকম আমি আগেই উদাহরণ দিলাম এটা কোন বাচ্চা হতে পারে অথবা তোমার বয়স্ক বাবা মা হতে পারেন এটা যে কোন পশ্চ তথা গরু তোমার পোষা কুকুর বন্য জীবজন্তু ইত্যাদি হতে পারে।

(Refer Slide Time: 14:19)

Why do we need object tracking?

Example 1 :: Delivery of medicines

- It is required to deliver some medicines under some required environmental conditions.
- Hand over the medicines to some delivery agency.
- Along with the medicines, attach some device that will sense the temperature, humidity and other relevant environmental conditions, along with its location.
- The information from the various sensors shall be uploaded to a server periodically through some suitable communication mechanism (e.g. GPRS, SMS).
- The concerned persons can track the consignments in real time, and may also have access to historical data.

তো অবজেক্ট ট্র্যাকিং(object tracking) কি জন্য দরকার? আমার মনে হয় অবজেক্ট ট্র্যাকিং(object tracking)-এর মূল উদ্দেশ্য সম্পর্কে তোমাদেরকে জানিয়েছি কিন্তু আমি আরো কিছু উদাহরণ দেব যেখানে দেখবে যে শুধুমাত্র অবজেক্ট ট্র্যাক(Object Track) করার কোনো অর্থই হয় না। আমাদের আরও বেশকিছু সেন্সর এর প্রয়োজন যা বস্তুটির সঙ্গে লেগে থাকে আরো বেশি বিভিন্ন ধরনের ডেটা(Data) সংগ্রহ করবে।

ওষুধ পাঠ্নোর উদাহরণ দেওয়া যাক। ধরা যাক কিছু প্রাকৃতিক বিধিনিষেধ মেনে আমাদেরকে ওষুধ পাঠাতে হবে। তোমরা জানো যে ওষুধ রাখার জন্য নির্দিষ্ট একটা তাপমাত্রা প্রয়োজন আছে। তুমি যদি এটা কে প্রচল্ন উষ্ণ তাপমাত্রায় রেখে দাও তাহলে সেটি খারাপ হয়ে যেতে পারে।

সুতরাং তোমাকে তাপমাত্রা নিয়ন্ত্রণ করে রাখতে হবে।

তো সাধারণত কি করা হয় যে যদি স্থান A থেকে স্থান B তে আমি ওষুধ পাঠাতে চাই আমরা এই ওষুধটা কোন ডেলিভারি(Delivery) সংস্থার হাতে তুলে দিই। ওষুধের সাথে আমরা একটা ডিভাইস ও জুড়ে দি যা তাপমাত্রা, আর্দ্রতা, এবং আরো প্রাসঙ্গিক বিভিন্ন তথ্য মাপার পাশাপাশি অবস্থানও নির্ণয় করতে পারবো। সুতরাং এটা শুধুমাত্র এই তথ্যই পাঠাচ্ছে না যে ওষুধ ও সতিয়ই কোন কোন স্থান দিয়ে নিয়ে যাওয়া হচ্ছে এটা ছাড়াও আমরা প্রিসময়ের তাপমাত্রা মাপতে পারছি।

কিছু সময় অন্তর বিভিন্ন সেন্সর থেকে তথ্য কোন সার্ভার(Server) এ আপলোড(Upload) করা হবে উপর্যুক্ত যোগাযোগ ব্যবস্থার মাধ্যমে। এই কোশলটি হতে পারে জিপিআরএস(GPRS) অথবা এসএমএস(SMS) আমরা পরে দেখব এই জিনিসগুলো।

এবং এটি অবস্থানের পাশাপাশি আরো কিছু স্থিতিমাপ কোন তথ্য ভান্ডারে আপলোড করবে। আমরা অবশ্যই এসএমএসের (SMS)সাহায্যও এটা করতে পারি যার ফলে এই তথ্য অন্য মোবাইল নাম্বারে(Mobile Number) পাঠানো যাবে। কিন্তু তুমি যদি অন্তর্জালের মাধ্যমে পাঠাতে চাও তাহলে তোমাকে জিপিআরএস(GPRS) এর সাহায্য নিতে হবো সুতরাং এই ভাবে এটি প্রাথমিক ভাবে কাজ করবে।

উদ্ভৃত ব্যক্তি পণ্টির গতিবিধি প্রকৃত সময় নির্ণয় করতে পারছে অবশ্যই যদি সময়ের সাথে সাথে ডেটা(Data) আপলোড করা হয়, ধরা যাক প্রতি 5 মিনিট অন্তর আমরা ,যে সমস্ত ডিভাইস (Device)গুলি সার্ভার(Server) এ ডাটা পাঠাচ্ছে তাদের কাছে যেটা পাঠাচ্ছি তাহলে উদ্ভৃত ব্যক্তি খুব সহজেই পণ্টির গতিবিধি প্রকৃত সময় নির্ধারণ করতে পারবে; এবং তথ্যভাওরে সংরক্ষণ করার দরুন পুরনো ডেটা(Data) ও সে দেখতে পাবো সুতরাং এক ঘন্টা আগে কি হয়েছিল তাপমাত্রা কত ছিল সেটা যদি আমি বের করার চেষ্টা করি তাও আমি পারবো কারণ তথ্যভাওরে সমস্ত কিছু সংরক্ষিত আছে।

(Refer Slide Time: 18:19)

Example 2 :: Tracking of living beings

The mechanism is very similar; only the sensors that are required shall vary from one situation to the next.

- Tracking children when they are playing outside or playing.
- Tracking wild life to gain information about their lifestyle.
- Tracking of pets and cattle.

এবার জীবিত বস্তুর গতিবিধি নির্ণয় এর উদাহরণ দেওয়া যাক | ওশুধের ক্ষেত্রে যে কৌশলের কথা আলোচনা করেছিলাম সেই একই কৌশল এখানেও ব্যবহৃত হবে শুধুমাত্র প্রয়োগ বিশেষে সেন্সর(Sensor) -এর ধরন গুলি বদলে যাবে।

বন্য জীবজগতের জীবনধারা নির্ধারণ করতে গিয়ে আরো অনেক তথ্যই আমাদের হাতে আসতে পারে। আমাদের কাছে অ্যাক্সেলেরোমিটার(Accelerometer) নামক কোন যন্ত্র থাকতে পারে সময়ের সাথে সাথে এই জিনিসটি নিয়েও আমরা আলোচনা করব। তো অ্যাক্সেলেরোমিটার(Accelerometer) কি আমরা দেখব এই ধরনের ডিভাইস(Device) আমরা ব্যবহার করতে পারি যাতে কোনো নির্দিষ্ট একটি পশ্চর সারাদিনে কার্যকলাপ পর্যবেক্ষণ করতে পারি। আমরা সবসময়ই আমাদের পোষ্য অথবা গবাদি পশ্চর গতিবিধির ওপর এইভাবে নজরদারি করতে পারি।

(Refer Slide Time: 19:13)

Requirements for building an object tracker

There are four basic requirements for developing an object tracking system:

- Processing** :: A microcontroller is required for carrying out some local processing, and interface with the various sensors and the communication subsystems.
- Communication Capability** :: The system must have some mechanism to communicate with some central server (e.g. GSM/GPRS module).
- Location Sensing** :: The most important criterion of an object tracker is to accurately locate its position. A GPS module can be used for this purpose.
- Parameter Sensing** :: Depending on the application, various physical parameters like temperature, humidity, pressure, etc. may need to be sensed. Such sensors must be interfaced with the microcontroller.

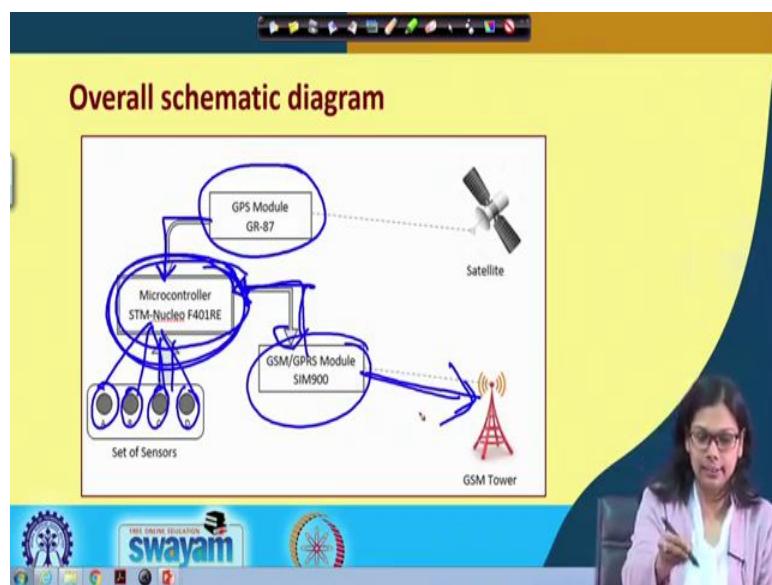
তো অবজেক্ট ট্র্যাকার(Object Tracker) তৈরির প্রয়োজনীয়তা কি? আমি ইতিমধ্যে এর উদ্দেশ্য সম্পর্কে তোমাদেরকে জানিয়েছি। তোমরা খুব সহজেই এটা তৈরি করতে পার, যদিও এর কিছু জিনিস আবশ্যিক আছে। তোমার এরকম কোনো ডিভাইস(Device) থাকতে হবে যা কিনা অবস্থান নির্ণয় করবে এবং ডেটা(Data) সংগ্রহ করবে যদি প্রয়োজন পড়ে ডেটা আপলোড করবে।

দেখা যাক প্রক্রিয়াকরণের জন্য কি কি ডিভাইস(Device) প্রয়োজন স্থানীয় প্রক্রিয়াকরণের জন্য এবং বিভিন্ন সেন্সর(Sensor) ও যোগাযোগ ব্যবস্থার সাথে তথ্য আদান প্রদানের জন্য মাইক্রোকন্ট্রোলার(Microcontroller) দরকার। অবস্থানটিকে নির্ধারণ করতে হবে, সুতরাং মাইক্রোকন্ট্রোলার(Microcontroller) এর জন্য কোন কোড(Code) লিখতে হবে যা কিনা অবস্থান নির্ণয় সাহায্য করবে।

ধরা যাক প্রতি 5 মিনিট অন্তর আমি অবস্থান নির্ণয় করতে চাইছি। তো আমাকে সেইমতো কোড(Code) লিখতে হবে। প্রতি 5 মিনিট পর এটি সেন্সর থেকে ডেটা(Data) সংগ্রহ করবে এবং সার্ভার(Server) এ পাঠিয়ে দেবে। সুতরাং প্রথম ধাপে আমাদের একটা মাইক্রোকন্ট্রোলার(Microcontroller) দরকার।

এরপর যোগাযোগ ক্ষমতার জন্য তোমাদের দরকার একটা জিএসএম মডিউল(GSM Module)। সময়ের সাথে সাথে আমরা এটা আলোচনা করব। এবং অবস্থান নির্ণয়ের জন্য জিপিএস মডিউল(GPS Module) লাগবে জিপিএস অর্থাৎ গ্লোবাল পজিশনিং সিস্টেম(Global Positioning System)। এবং স্থিতিমাপ মাপার জন্য আমাদের বিভিন্ন সেন্সর লাগবে তথা তাপমাত্রা মাপার সেন্সর(Sensor) অথবা আর্দ্রতা মাপার(Sensor) সেন্সর অথবা বাতাসের চাপ মাপার সেন্সর(Sensor)। তো আসলে এই চারটি জিনিসের প্রয়োজন অবজেক্ট ট্রাকিং(Object Tracking) সিস্টেম(System) তৈরীর জন্য।

(Refer Slide Time: 21:53)



এইটাই হচ্ছে সম্পূর্ণ নকশা।

যদি লক্ষ্য করো দেখবে এটা হচ্ছে মাইক্রোকন্ট্রোলার(Microcontroller) এটা হচ্ছে জিপিএম মডিউল(GSM Module) যেটা কিনা মাইক্রোকন্ট্রোলার(Microcontroller) এর সাথে যুক্ত থাকে স্থানাঙ্কগুলি (coordinates) নির্ধারণ করবে। এছাড়াও আরো সেন্সর(Sensor) যুক্ত থাকতে পারে যা থেকে মাইক্রোকন্ট্রোলার(Microcontroller) তথ্য সংগ্রহ করবে।

এবং যেরকম আমি বললাম যে প্রতি 5 মিনিট অন্তর মাইক্রোকন্ট্রোলার(Microcontroller) এর সাহায্যে জিপিএম মডিউল(GSM Module) ব্যবহার করে এই ডেটা(Data) কোন সার্ভার(Server) এ পাঠানো হবে। এইটাই হচ্ছে সামগ্রিক সিস্টেম(System) টা যেখানে প্রক্রিয়াকরণ ক্ষমতার পাশাপাশি ডেটা(Data) সংগ্রহ করার জন্য বিভিন্ন সেন্সর(Sensor) আছে।

জিপিএম মডিউল থাকায় আমাদের কাছে যোগাযোগ সাধনের কিছু ক্ষমতা আছে সেই কারণেই আমরা সার্ভারে(Server) ডেটা(Data) পাঠাচ্ছি। এই মূহূর্তে সার্ভারে(Server) যেটা আছে তা দিয়ে আমরা বর্তমান অবস্থান জানতে পারবো। এবং পূর্বের অবস্থান যদি জানতে চাই তাও তোমরা দেখতে পাবে।

(Refer Slide Time: 23:19)



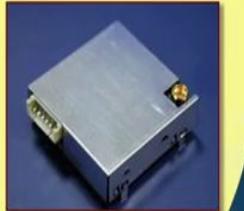
তো এটাই হচ্ছে সেই মাইক্রোকন্ট্রোলার(Microcontroller) যা আমি ইতিমধ্যেই আলোচনা করেছি সুতরাং আর বিষয়ে আমি বলছি না কিছু।

(Refer Slide Time: 23:27)

(b) The GPS module

Basically used for location sensing.

- The Global Positioning System (GPS) is a satellite-based navigation system made up of a network of satellites.
- It can be used anywhere in the world, 24 hours a day.
- Example: Holux GR-87 GPS module, which supports the standard NMEA-0183 format.



জিপিএস মডিউল(GPS Module) প্রসঙ্গে যদি আমি তাহলে বিভিন্ন জিপিএস মডিউল পাওয়া যায়। এটা যেটা দেখছো সেইটা এক ধরনের জিপিএস মডিউল যেটাকে বলা হয় Holax GR-87 জিপিএস মডিউল(GPS Module) কিন্তু তোমরা অন্য ধরনের জিপিএস মডিউল(GPS Module) ও ব্যবহার করতে পারো।

(Refer Slide Time: 23:39)

The diagram illustrates the SIM900A GSM module, a green printed circuit board (PCB) with various components and connectors. A blue circle highlights the central SIM900A chip. Labels on the board include: SIM900A, Network LED, ON/OFF, DC IN, PC Output, ON/OFF Switch, DC Input, AC Input, Power LED, SIM Socket, and a pinout for the SIM card labeled V_{DD}, GND, MOSI, MISO, SCK, CS, and V_{DD}, GND, UNL, L1. A woman in a pink jacket is visible in the bottom right corner, and the Swayam logo is in the bottom left.

(c) The GSM module

Basically used for communication with a remote location.

- Uses the same technology as used by the mobile phone networks for communication.
- It can be used provided there is a mobile tower in the range.
- Example: SIM900A GSM module, which supports the standard AT command set. It is a tri-band GSM/GPRS engine that works on frequencies 900 MHz, 1800 MHz, and 1900 MHz.

এবং অবশ্যে হচ্ছে জিএসএম মডিউল(GSM Module) এই কোর্স(Course) আমরা ব্যবহার করব সিম900A(SIM900A) জিএসএম মডিউল(GSM Module), যা পরে আমি আরো বিশদে আলোচনা করব। এটা আসলে দূরবর্তী কোনো স্থানের সাথে যোগাযোগের জন্য ব্যবহৃত হয় এবং মোবাইল ফোন (Mobile Phone)যে প্রযুক্তিতে কাজ করে সেই একই প্রযুক্তি এতেও ব্যবহৃত হয়।

যখনই আমরা মোবাইল ফোন ব্যবহার করি সেটিতে একটা সিম রাখার জায়গা আছে, যেখানে আমরা সিমকার্ড((SIM Card)) এবং এটাই হচ্ছে সিম900A- (SIM900A)র চিপ(Chip)। এইটা হচ্ছে ডেভেলপমেন্ট বোর্ড(Development Board) তোমরা এই নির্দিষ্ট চিপটি (Chip) অন্যান্য ডেভেলপমেন্ট বোর্ড- দেখতে পাবো এইখানে হচ্ছে সিম(SIM) ভরার জায়গাটি যেখানে তুমি সিম কার্ডটা(SIM Card) রাখবো এবং অবশ্যই এই Vcc, ভূমি(ground) এবং অন্যান্য পিন(Pins) গুলিও যোগ করতে হবে ট্রান্সমিটার(Transmitter) ও রিসিভারের(Receiver) সাথে।

এই একই মডিউল আমাদের পরিচিত পরীক্ষা-নিরীক্ষার জন্য ব্যবহৃত হয়েছে যা কিনা স্ট্যান্ডার্ড এটি কমান্ড(Standard AT Command) সেট সাপোর্ট করে। আমাদেরকে স্ট্যান্ডার্ড এটি কমান্ড(Standard AT Command) ব্যবহার করতে হয় এই জিএসএম মডিউলের(GSM Module) সাথে যোগাযোগ স্থাপনের জন্য। এইটা হচ্ছে ট্রাই-ব্যান্ড(Tri-Band) জিএসএম(GSM) জিপিআরএস(GPRS) ইঞ্জিন(Engine) যা কিনা তিনি ধরনের কম্পাক্সে কাজ করতে পারে যথা 900 মেগাহার্জ(900 MHz), 1800 মেগাহার্জ(MHz), 1900 মেগাহার্জ(MHz)।

(Refer Slide Time: 25:29)



এর সঙ্গে একীভূত করার জন্য তোমার কাছে বিভিন্ন ধরনের সেন্সর(Sensor) থাকতে পারে। তো, যোগাযোগের জন্য আমরা ব্লুটুথ(Bluetooth), বেতার যোগাযোগ ব্যবস্থা অথবা নিয়ার ফিল্ড(Near Field) যোগাযোগ ব্যবস্থা ব্যবহার করতে পারি।

তো আমরা এই আলোচনার অন্তিম পর্যায়ে চলে এসেছি, যেখানে আমি তোমাদেরকে একটি ধারণা দিলাম কিভাবে একটি সামান্য আইওটি(IoT) সিস্টেম(System) বানানো যেতে পারে যাতে আছে প্রক্রিয়াকরণ ক্ষমতা, তার সঙ্গে মিলিত হচ্ছে সেন্সর(Sensor) ডাটা(Data), এবং আমি এর একটি প্রয়োগ ও আলোচনা করলাম যথা অবজেক্ট ট্র্যাকিং(Object Tracking)। আমরা দেখলাম যে পুরো সিস্টেম(System) টাকে তৈরি করা খুবই সহজ, কিন্তু এর এত বিবিধ প্রয়োগ আছে, এটা আরও কত ক্ষেত্রে ব্যবহৃত হতে পারে।

GLOSSARY

English	Transliteration	Translation
Database	ডেটাবেস	তথ্যভান্ডার
Frequency	ফ্রিকোয়েন্সি	কম্পাক্ষ
Megahertz	মেগাহার্জ	কম্পাক্ষ মাপার একক
Object tracking	অবজেক্ট ট্র্যাকিং	বস্তুর গতিবিধি নির্ণয় করা
Embedded systems	এমবেডেড সিস্টেম	একপ্রকার বৈদ্যুতিন যন্ত্র
Data	ডেটা	তথ্য
System	সিস্টেম	ব্যবস্থা
Sensor	সেন্সর	এক প্রকার বৈদ্যুতিন যন্ত্র যা কোন ভৌত স্থিতিমাপকে মাপতে পারে

Bluetooth	ব্লুটুথ	বিভিন্ন বৈদ্যুতিক যন্ত্রপাতির মধ্যে তারবিহীন যোগাযোগের একপ্রকার আদর্শ।
-----------	---------	--

Actuator	একচুয়াটর	যে যন্ত্রের দ্বারা শক্তিকে চলনে রূপান্তরিত করা যায়
Mobile Phone	মোবাইল ফোন	চলভাষ

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Lecture – 35
GSM and Bluetooth

(Refer Slide Time: 00:31)



আজকের আলোচনায় সবাইকে স্বাগত। এই আলোচনায়, আমি যোগাযোগ ব্যবস্থার দুটি প্রোটোকল নিয়ে আলোচনা করব, একটি হচ্ছে জিএসএম(GSM), অন্যটি হচ্ছে ব্লুটুথ(Bluetooth)

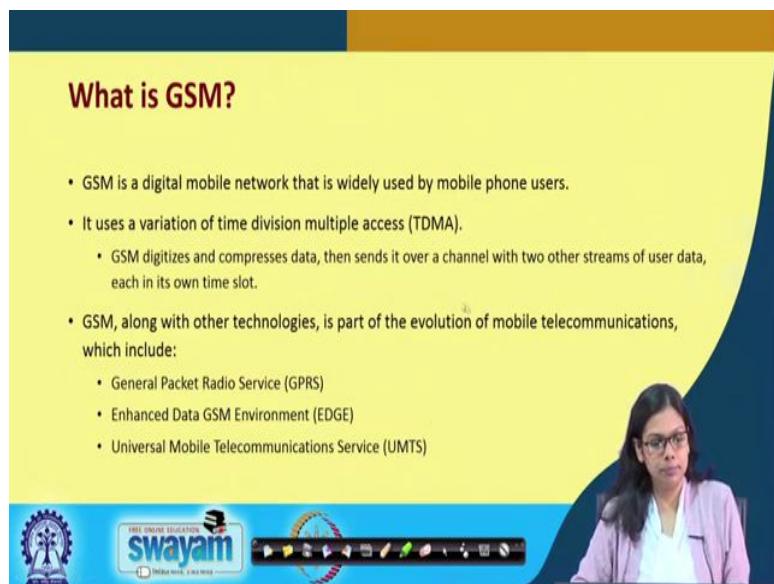
|

(Refer Slide Time: 00:45)



জিএসএম(GSM) অর্থাৎ গ্লোবাল সিস্টেম ফর মোবাইল কমিউনিকেশন(Global System for Mobile Communication)।

(Refer Slide Time: 00:53)



জিএসএম(GSM) হচ্ছে একটি ডিজিটাল(Digital) মোবাইল নেটওয়ার্ক(Mobile Network) যা মোবাইল ফোন(Mobile Phone) ব্যবহারকারীদের মধ্যে বহুল ব্যবহৃত। এটি টাইম ডিভিশন

মাল্টিপল অ্যাক্সেস(Time Division Multiple Access) অথবা টিডিএমএ(TDMA)- এর এক প্রকার বৈচিত্র্য ব্যবহার করে।

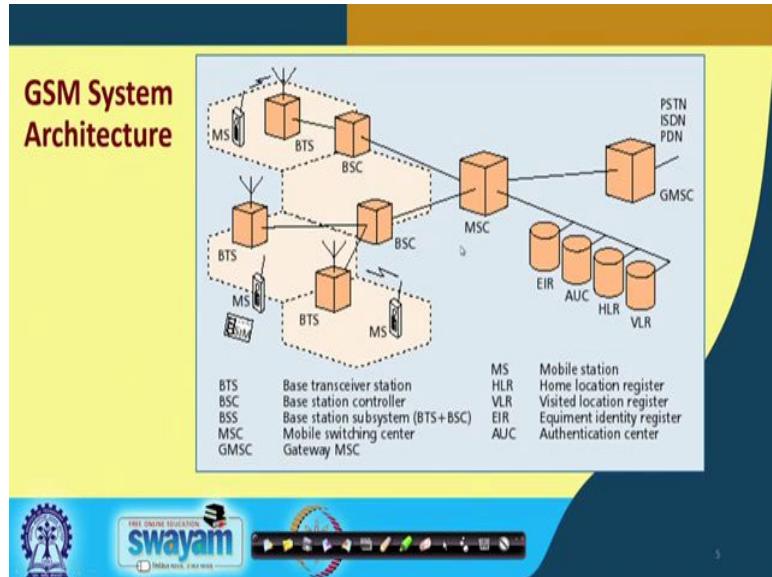
জিএসএম(GSM) ডেটাকে(Data) ডিজিটাইজ(Digitize) করে এবং সংকুচিত করে, তারপর ব্যবহারকারী ডেটা অন্য দুটি প্রবাহের সাথে একটি চ্যানেল(Channel)-এর মাধ্যমে পাঠিয়ে দেয়, এবং প্রত্যেকে নিজের নিজের সময় ভাগে চলতে থাকে। তো, এর অর্থ কি? এর মানে এটি ডেটা(Data) কে ডিজিটাইজ(digitize) করে, সংকুচিত করে, এবং একটি চ্যানেল- এর মাধ্যমে পাঠিয়ে দেয় যার মধ্য দিয়ে অন্যান্য ডেটা(data)-ও চলাফেরা করছে। তো, তারা টাইম ডিভিশন মাল্টিপল অ্যাক্সেস (time division multiple access) ব্যবহার করে চ্যানেল(channel)-টিকে নিজেদের মধ্যে ভাগ করে নিচ্ছে।

অন্যান্য প্রযুক্তির সাথে, জিএসএম(GSM)- ও মোবাইল যোগাযোগ ব্যবস্থার ক্রমোন্নতির একটি অংশ, যার সঙ্গে আরো অনেক প্রোটোকল(Protocol) জড়িয়ে আছে যেমন জেনারেল প্যাকেট রেডিও সার্ভিস (General Packet Radio Service) তথা জিপিআরএস (GPRS) যা কিনা ডেটা (Data) প্যাকেট পাঠাতে ব্যবহৃত হয়, এনহান্সড ডেটা জিএসএম এনডারমেন্ট(Enhanced Data GSM Environment) যথা এজ (EDGE), এবং ইউনিভার্সাল মোবাইল টেলিকমিউনিকেশন সার্ভিস (Universal Mobile Telecommunication Service) যথা ইউএমটিএস (UMTS)।

আমাদের ক্ষেত্রে আমরা জিএসএম (GSM) ব্যবহার করেছি মাইক্রোকন্ট্রোলার(microcontroller) থেকে অন্য যেকোনো মোবাইল ডিভাইস(mobile device)-এ এসএমএস(SMS) পাঠানোর জন্য এবং অন্য যেকোনো মোবাইল ডিভাইস(mobile device) থেকে জিএসএম(GSM)- এ এসএমএস (SMS) পাওয়ার জন্য। তো এই দুটি জিনিসই আমরা আমাদের পরীক্ষা-নিরীক্ষায় দেখিয়েছি কিন্তু অন্যরকম কিছুও করা যেতে পারে। তোমরা জিপিআরএস (GPRS)- এর মাধ্যমে প্যাকেট (packet) পাঠাতে পারো, সেইমতো তোমাকে তোমার জিএসএম (GSM)-টিকে প্রস্তুত করতে হবে। তোমাকে

জিপিআরএস (GPRS) এর মাধ্যমে ডেটা (Data) পাঠাতে হলে নির্দিষ্ট কমান্ড (Command) ব্যবহার করতে হবে।

(Refer Slide Time: 03:11)



এইটাই হচ্ছে বন্ধুত জিএসএম(GSM) সিস্টেম(System) আর্কিটেকচার(Architecture)। এই এমএস(MS)-টা হচ্ছে মোবাইল স্টেশন(Mobile Station), এবং এর দুটি অঙ্গ আছে। একটি হচ্ছে মোবাইল সরঞ্জামটি, এবং অন্যটি হচ্ছে সাবস্ক্রাইবার আইডেন্টিটি মডিউল(Subscriber Identity Module) অথবা সিম(SIM)। বিটিএস(BTS) অর্থাৎ বেস ট্রান্সিভার স্টেশন(Base Transceiver Station), তুমি দেখতে পাচ্ছ এইটা হচ্ছে একটা লোক। তো, এই হচ্ছে একটি সেল(cell)। এই সেলটির মধ্যে একটি বিটিএস(BTS) রয়েছে, তোমার আর একটি সেল(cell) আছে যার মধ্যে আরো একটি বিটিএস(BTS) রয়েছে। এবং এটা ব্যবহৃত হয় এবং এর মধ্যে আছে উচ্চগতিসম্পন্ন ট্রান্সমিটার(transmitter) এবং রিসিভার(receiver)। এবং এটি দুটি চ্যানেল সরবরাহ করে, একটি হচ্ছে সিগনালিং ডেটা(Data) আদান প্রদানের জন্য, এবং অন্যটি হচ্ছে ডেটা চ্যানেল(data channel)।

এরপর আমাদের আছে বেস স্টেশন কন্ট্রোলার(Base Station Controller) অথবা বিএসসি(BSC)। এবং তুমি দেখছো যে এই বেস(Base) যেটি আসছে সেটি দুটি বিটিএস(BTS)- এর সাথে যুক্ত। তো, এই বেস স্টেশন কন্ট্রোলার(Base Station Controller) একাধিক বিটিএস(BTS)- কে নিয়ন্ত্রণ করে, একটি বিটিএস(BTS) এখানে আছে, একটি বিটিএস(BTS) এখানে, যেটা কিনা বিএসসি(BSC) দ্বারা নিয়ন্ত্রিত তোমরা দেখতে পাচ্ছ। এবং এটি রেডিও রিসোর্স ম্যানেজমেন্ট(radio resource management), বিটিএস(BTS)-গুলির মধ্যে সমলয় স্থাপন সময় এবং কম্পাক্ষের সিগন্যাল(Signal)গুলির মাধ্যমে ইত্যাদি কাজও করে। তো, আমাদের কাছে মোবাইল স্টেশন(mobile stations) আছে, এদের মধ্যে প্রত্যেকটির একটি করে বিটিএস(BTS) আছে এই সেল(Cell)-টায়। এবং কতকগুলি বিটিএস(BTS), যেগুলি বিভিন্ন সেল(cell)- এ আছে সেগুলি বিএসসি(BSC) অথবা বেস স্টেশন কন্ট্রোলার(Base Station Controller)- এর সাথে যুক্ত।

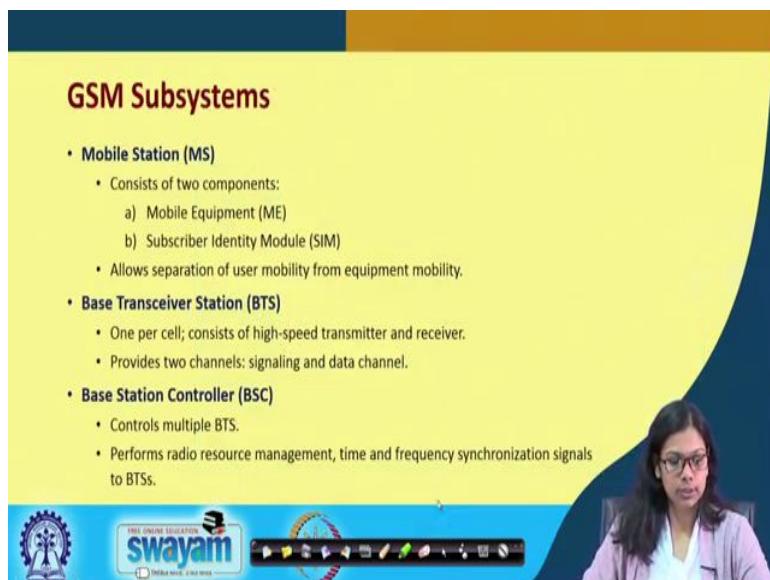
এরপর আমরা দেখব এমএসসি(MSC) অথবা মোবাইল সুইচিং সেন্টার(Mobile Switching Center)। পাবলিক ল্যান্ড মোবাইল নেটওয়ার্ক(Public Land Mobile Network)- এর একটি সুইচিং নোড হ্যান্ডঅফ(handoff)- এর মতন বিভিন্ন বেতার সংস্থান এবং গ্রাহকদের সচলতা যথা অবস্থান, নথিভুক্তকরণ ইত্যাদি নিয়ন্ত্রণ করে। এবং এই পিএলএমএন(PLMN)-এ একাধিক এমএস(MS) থাকতে পারে। এখানে এই পাবলিক ল্যান্ড মোবাইল নেটওয়ার্ক(public land mobile network)- এ একাধিক এমএসসি(MSC) থাকতে পারে।

এরপর তোমার থাকবে একটা জিএমএসসি(GMSC), যেটা হচ্ছে গেটওয়ে(Gateway) এমএসসি(MSC)। তোমরা দেখতে পাচ্ছ যে এই এমএসসি(MSC)-টি গেটওয়ে(Gateway)- এর সাথে যুক্ত। তো, তোমার আছে এই গেটওয়ে(gateway) এমএসসি(MSC), যেটি একটি চলমান নেটওয়ার্কের(Network) সাথে একটি স্থির নেটওয়ার্কের(network) যোগাযোগ করায়। এবং আমরা মাঝেমধ্যেই বলি যে আমরা বাড়িতে আছি অথবা বাইরে আছি, এক্ষেত্রে এইচএলআর(HLR) অথবা হোম লোকেশন রেজিস্টার(Home Location Register), এবং

ভিএলআর(VLR) অথবা ভিজিটর লোকেশন রেজিস্টার(Visitor Location Register) ব্যবহৃত হয়। এই দুটি ব্যাপার জিএসএম(GSM)- এর ক্ষেত্রে গুরুত্বপূর্ণ।

এবং এই এমএসসি(MSC)- টি এইচ এল আর(HLR)- এর সাথে তথ্য আদান প্রদান করে। ধরা যাক যখন তুমি তোমার বাড়ির অবস্থান তথ্য A থেকে অন্য স্থানে তথ্য B- তে যাও, তখনকার তাৎক্ষণিক তথ্য ভিএলআর(VLR)- এ সংরক্ষিত থাকে। এখানেই এই ভিজিটর(visitor) রেজিস্টার(register)-টি থাকে। যে যে গ্রাহক বর্তমানে ওই এমএসসি(MSC)-র পরিষেবা অঞ্চলে আছে কিন্তু যাদের এইচএলআর(HLR) অন্য জায়গায় আছে, অর্থাৎ তারা এখন এই জায়গাটিতে আগন্তুক হিসেবে রয়েছেন, তাদের ব্যাপারে তাৎক্ষণিক তথ্য এর কাছে আছে।

(Refer Slide Time: 08:03)



GSM Subsystems

- **Mobile Station (MS)**
 - Consists of two components:
 - a) Mobile Equipment (ME)
 - b) Subscriber Identity Module (SIM)
 - Allows separation of user mobility from equipment mobility.
- **Base Transceiver Station (BTS)**
 - One per cell; consists of high-speed transmitter and receiver.
 - Provides two channels: signaling and data channel.
- **Base Station Controller (BSC)**
 - Controls multiple BTS.
 - Performs radio resource management, time and frequency synchronization signals to BTSs.

তো একদম এইগুলিই আমি ইতিমধ্যে আলোচনা করেছি এই মোবাইল স্টেশন(Mobile Station), বেস ট্রান্সিভার স্টেশন(Base Transceiver Station) অর্থাৎ বিটিএস(BTS), বেস স্টেশন কন্ট্রোলার(Base Station Controller)।

(Refer Slide Time: 08:13)

- **Mobile Switching Center (MSC)**
 - Switching node of a Public Land Mobile Network (PLMN).
 - Manages allocation of radio resources (handoff), and mobility of subscribers (location registration).
 - There can be several MSCs in a PLMN.
- **Gateway MSC (GMSC)**
 - Connects mobile network to a fixed network.
- **Home Location Register (HLR)**
 - For all registered users, it keeps user profile.
 - MSCs exchange information with HLR.
- **Visitor Location Register (VLR)**
 - Contains temporary information about mobile subscribers that are currently located in the area but whose HLR are elsewhere.

মোবাইল সুইচিং সেন্টার(Mobile Switching Center), গেটওয়ে মোবাইল সুইচিং সেন্টার(Gateway Mobile Switching Center), হোম লোকেশন রেজিস্টার(Home Location Register) এবং ভিজিটর লোকেশন রেজিস্টার(Visitor Location Register), যেগুলি আমি ইতিপূর্বেই আলোচনা করেছি।

(Refer Slide Time: 08:27)

GSM Identifiers

- International mobile subscriber identity (IMSI)
 - Unique 15 digits assigned by service provider, including home country code.
- International mobile station equipment identity (IMEI)
 - Unique 15 digits assigned by equipment manufacturer.
- Temporary mobile subscriber identity (TMSI)
 - 32-bit number assigned by VLR to uniquely identify a mobile station within a VLR's area.

এখন এটা একটা গুরুত্বপূর্ণ জিনিস যে জিএসএম(GSM) আইডেন্টিফায়ার(identifier)কি? কিভাবে একটি জিএসএম(GSM)- কে সনাক্ত করা হয় ? এটির একটি বিশিষ্ট পরিচিতি আছে । প্রথমটা হচ্ছে ইন্টারন্যাশনাল মোবাইল সাবস্ক্রাইবার আইডেন্টিটি(International Mobile Subscriber Identity) যেটাকে বলা হয় আইএমএসআই(IMSI) । এটি একটি 15 টি একক সংখ্যা বিশিষ্ট নম্বর যা পরিষেবা প্রদানকারীর তরফ থেকে সরবরাহ করা হয় । তো, তুমি আসলে পরিষেবা প্রদানকারী সংস্থার থেকে পরিষেবা নিচ্ছ, তারাই তোমাকে তোমার গৃহের জাতীয় কোড(Code) সমেত এটি তোমাকে সরবরাহ করছে । তুমি যদি অন্য কোন দেশে যাও এবং এটাকে ব্যবহার করো, তাহলে সেই দেশের বিশিষ্ট কোড(Code) এর সাথে জুড়ে যাবে ।

এর পরেরটা হচ্ছে ইন্টারন্যাশনাল মোবাইল স্টেশন ইকুইপমেন্ট আইডেন্টিটি(International Mobile station Equipment Identity) অথবা আইএমইআই(IMEI) নম্বর। এটি একটি 15 সংখ্যার বিশিষ্ট নম্বর যেটি সরঞ্জাম প্রস্তুতকারকের তরফ থেকে দেওয়া হয় । যখন প্রস্তুতকারক এই নির্দিষ্ট ডিভাইস(Device) তৈরি করে, তখন এটিকে এই বিশিষ্ট নাম্বারটি(Number) দেয় যেটাকে বলে আইএমইআই(IMEI) নাম্বার । এরপর থাকতে পারে একটা টেম্পোরারি মোবাইল সাবস্ক্রাইবার আইডেন্টিটি(Temporary Mobile Subscriber Identity), যেটি কিনা 32 টি বিট(Bit)- এর সম্মিলিত একটি নাম্বার যেটি ভিএলআর(VLR) আমাদেরকে সরবরাহ করে যখনই আমরা অন্য কোন জায়গায় যাই, এবং এটি দ্বারা সেটি নিজের অঞ্চলের বিশিষ্ট মোবাইল স্টেশনটিকে(mobile station) সনাক্ত করার চেষ্টা করে । তো ,এটি এও হিসেবে রাখবে যে কতজন লোক বাইরে থেকে এই অঞ্চলে ঘোরাফেরা করছে । তারা এই 32- বিটের(Bit) নম্বরটি দেয় যাতে ওই অঞ্চলের সক্রিয় মোবাইল(mobile)- গুলির কাছে তারা পৌঁছাতে পারে ।

(Refer Slide Time: 10:17)

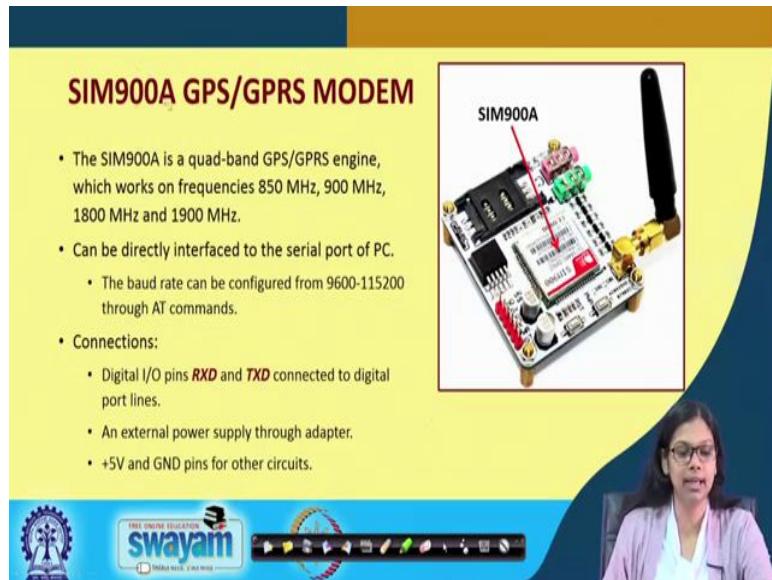
AT Command Set for GSM Modem

- AT commands are used to control MODEMs.
- AT commands with a GSM/GPRS MODEM can be used to access the following information and services:
 - a) Information and configuration pertaining to MODEM and SIM card.
 - b) SMS service.
 - c) MMS service.
 - d) Fax service.
 - e) Data and voice link over mobile network.

The slide features a video of a woman with glasses and a pink jacket speaking. The video is part of the Swayam educational platform, as indicated by the logo and text at the bottom of the slide.

এখন, এই AT কমান্ডগুলো(commands) ব্যবহৃত হয় জিএসএম(GSM) মডিউল(module)-টিকে প্রস্তুত করার জন্য। এইজিএসএম(GSM) মডেম(modem)টি যেটা আমরা ব্যবহার করব সেটি এটি কমান্ড(AT command) ব্যবহার করে, যা আমরা আমাদের কোড(Code)- এ ব্যবহার করেছি। এটি কমান্ড(Command) -গুলি ব্যবহৃত হয় মডেম(Modem)-টির ক্রিয়া-কলাপ নিয়ন্ত্রণের জন্য। এবং এই কমান্ড(Command)-গুলি ও তার সঙ্গে জিএসএম(GSM) অথবা জিপিআরএস(GPRS) -মডেম গুলি নিম্নলিখিত তথ্য এবং পরিষেবাগুলি পেতে ব্যবহার করা যেতে পারে। সেই নিম্নলিখিত তথ্যগুলি কি কি? মডেম(Modem) ও সিম(SIM) সম্পর্কিত তথ্য ও রূপরেখা, তথ্য বলতে আমি বলতে চাইছি যে কি ধরনের সিম কার্ড(SIM Card) আমরা ব্যবহার করছি, মডেম(Modem) টির অন্যান্য রূপরেখা ইত্যাদি এসএমএস(SMS) পরিষেবা, এমএমএস(MMS) পরিষেবা, ফ্যাক্স(FAX) পরিষেবা এবং অতি অবশ্যই মোবাইল নেটওয়ার্কের(mobile network) মাধ্যমে ব্যবহৃত ডেটা(Data) ও ভয়েস(Voice) লিংক(Link)। তো, এটি এই সমস্ত পরিষেবা দান করে। আমাদের ক্ষেত্রে, আমরা এসএমএস(SMS) পরিষেবা ব্যবহার করেছি; তোমরা জিপিআরএস(GPRS) পরিষেবা ব্যবহার করতে পারো প্যাকেট(Packet) পাঠানোর জন্য, কিন্তু আমাদের ক্ষেত্রে আমরা তোমাদের দেখাবো যে কিভাবে আমরা এসএমএস(SMS) পরিষেবা ব্যবহার করব।

(Refer Slide Time: 11:29)



SIM900A GPS/GPRS MODEM

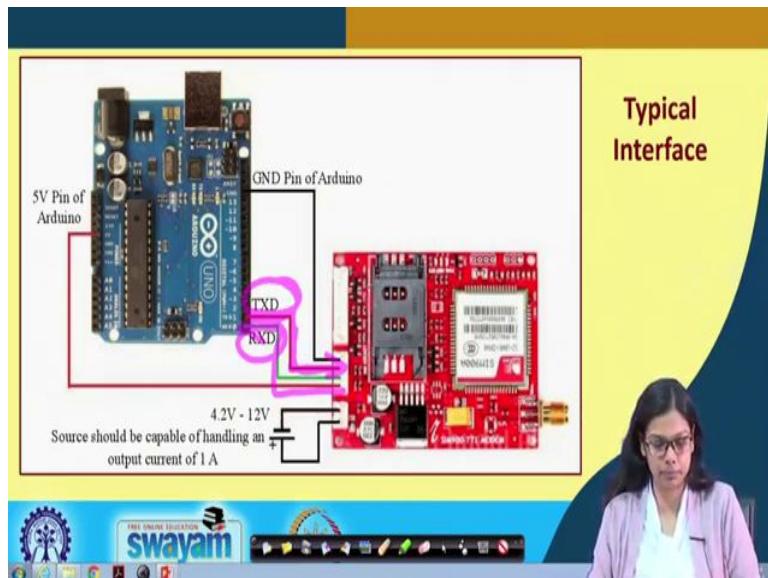
- The SIM900A is a quad-band GPS/GPRS engine, which works on frequencies 850 MHz, 900 MHz, 1800 MHz and 1900 MHz.
- Can be directly interfaced to the serial port of PC.
 - The baud rate can be configured from 9600-115200 through AT commands.
- Connections:
 - Digital I/O pins **RXD** and **TXD** connected to digital port lines.
 - An external power supply through adapter.
 - +5V and GND pins for other circuits.

এইটা হচ্ছে সিম900A(SIM900A), জিপিএস(GPS), জিপিআরএস(GPRS) মডেম(Modem)। তো, তোমরা দেখছো যে চিপ(Chip) যাই হোক না কেন এটা হচ্ছে চিপ(Chip) অঞ্চল, এবং এটা হচ্ছে সিম900A(SIM900A)। আমি যেরকম আগের আলোচনায় বলেছি, যে সিম900A(SIM900A) চারটে ব্যান্ড(quad-band) বিশিষ্ট একটি জিপিএস(GPS), জিপিআরএস(GPRS) ইঞ্জিন(engine) যেটি চারটি কম্পাক্ষ(frequency)- এ কাজ করে যথা 850 মেগাহার্জ, 900 মেগাহার্জ, 1800 মেগাহার্জ এবং 1900 মেগাহার্জ।

এবং এটিই পিসি(PC)-র সিরিয়াল পোর্ট(Serial Port)- এর সাথে সরাসরি জুড়ে দেওয়া যেতে পারে। বড রেট(Baud Rate)-টিকে এটি(AT) কমান্ড(Command) দ্বারা 9600 থেকে 115200 অবধি পরিবর্তন করা যেতে পারে। সংযোগ-টা একেবারেই সোজা। আরএক্সডি(RXD) এবং টিএক্সডি(TXD) ডিজিটাল(Digital) পিনগুলি, যেগুলি ট্রান্সমিটার(Transmitter) এবং রিসিভার(Receiver) হিসেবে কাজ করে, সেগুলিকে ডিজিটাল(Digital) পোর্ট(Port) লাইন(Line)-গুলোর সাথে যুক্ত করা হবে, আমরা দেখব কিভাবে সেটা আমরা করব

|এডাপ্টার(Adapter)- এর মাধ্যমে বাইরের তড়িৎ উৎস প্রয়োজন। এবং অবশ্যই, অন্যান্য বর্তনীর জন্য +5 ভোল্ট(Volt) এবং গ্রাউন্ড পিন(ground pin) গুলি দরকার।

(Refer Slide Time: 13:03)

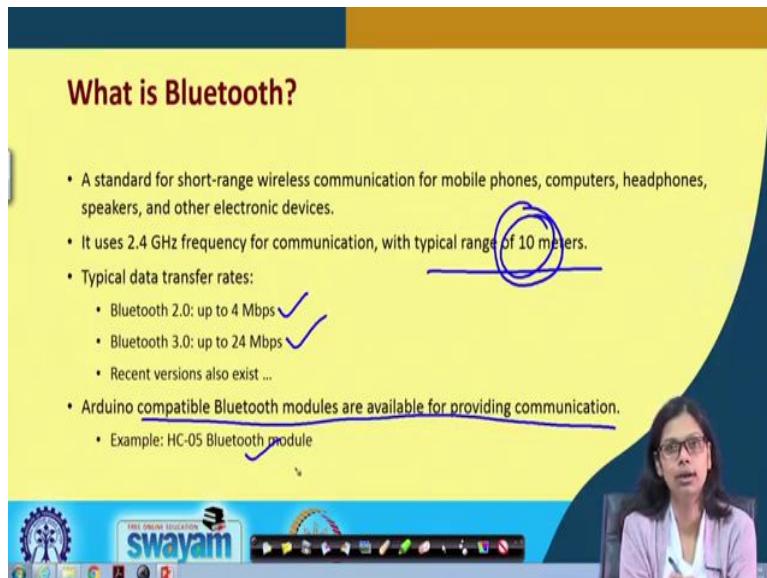


সংযোগ- টা এইভাবে যাচ্ছে। এইটা হচ্ছে আরডুইনো উনো(Arduino Uno) বোর্ড(Board), এটা হচ্ছে 5 ভোল্ট(Volt) এর পিনটি, যেটি জিএসএম(GSM)-এর 5 ভোল্ট(Volt)- এর পিনের সাথে যুক্ত। এবং এইটা হচ্ছে গ্রাউন্ড(ground)- এর জন্য, যেটি জিএসএম(GSM)- এর গ্রাউন্ড(ground)- এর সাথে যুক্ত। এবং জিএসএম(GSM)-এর রিসিভার(Receiver)-টি এই পিনটির ট্রান্সমিটার(Transmitter)- এর সাথে যুক্ত হবে এবং পিনের ট্রান্সমিটার(Transmitter)-টি আরডুইনো(Arduino) বোর্ডের রিসিভার(Receiver) পিন- এর সাথে যুক্ত হবে।

এই ব্যাপারটা আমাদেরকে বুঝতে হবে। যখনই আমি বলছি যে এটা কোন ডেটা(data) পাঠাবে, এটি কিছু ডেটা(data) পাবেও; এটি কিছু ডেটা(data) পাঠাবে এবং এটি কিছু ডেটা(data) গ্রহণ করবে। তো, একমুখী যোগাযোগটি এই দিক থেকে এইদিকে হবে; একমুখী যোগাযোগটি এই দিক থেকে এইদিকে হবে। তো, সেইমতো এই আরডুইনো বোর্ড(Arduino Board)- এর টিএক্স(TX) পিনটি এই সিম(SIM) মডিউল(Module)- টির আরএক্স(RX) পিনটির সাথে যুক্ত হবে। এবং

আরডুইনো(Arduino)- র আরএক্স(Rx) পিনটি যুক্ত হবে এই নির্দিষ্ট মডেলটির টিএক্স(TX) পিনটির সাথে। তো, আমাদের নিশ্চিত করতে হবে যে সংযোগ-টা ঠিকমতো হয়েছে।

(Refer Slide Time: 15:07)



What is Bluetooth?

- A standard for short-range wireless communication for mobile phones, computers, headphones, speakers, and other electronic devices.
- It uses 2.4 GHz frequency for communication, with typical range of 10 meters.
- Typical data transfer rates:
 - Bluetooth 2.0: up to 4 Mbps
 - Bluetooth 3.0: up to 24 Mbps
 - Recent versions also exist ...
- Arduino compatible Bluetooth modules are available for providing communication.
 - Example: HC-05 Bluetooth module

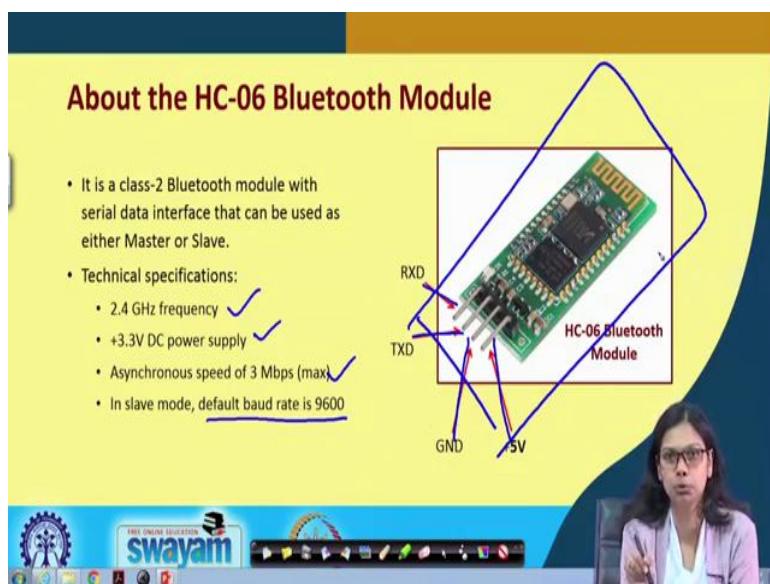
এরপর আমি ব্লুটুথ(Bluetooth) এর ব্যাপারে কথা বলব; আমরা সবাই নিশ্চয়ই মোবাইল ফোনে ব্লুটুথ ব্যবহার করেছি ছবি কিংবা অন্যান্য তথ্য আদান প্রদানের সময়। খুব সংক্ষিপ্ত আকারে ব্লুটুথ(Bluetooth)-এর ব্যাপারে বলব। মোবাইল ফোন(Mobile Phone), কম্পিউটার(Computer), হেডফোন(headphone), স্পিকার(Speaker) এবং অন্যান্য বৈদ্যুতিন যন্ত্রপাতির মধ্যে কম দূরত্বের তারবিহীন যোগাযোগের আদর্শ এটি। আজকালকার দিনে তুমি তারবিহীন মাউস(Mouse) এবং তারবিহীন কিবোর্ড(keyboard) দেখে থাকবে। পিসি(PC) এবং ডিভাইস(Device)-টির মধ্যে যোগাযোগ ব্লুটুথ(Bluetooth) দ্বারা সম্ভব হয় এই ভাবেই তারা একে অপরের সাথে যোগাযোগ করে।

এটি 2.4 GHz- এর কম্পাঙ্ক(frequency) ব্যবহার করে যোগাযোগের জন্য। এবং সাধারণত এর পরিধি 10 মিটার(metre) অবধি হয়। তো, যোগাযোগের পরিধিটা খুব একটা বড় নয়। আগের

জিএসএম(GSM) মডিউল(module) টিতে, আমরা একটা সিম(SIM) মডিউল(Module) ব্যবহার করছিলাম যেটা এসএমএস(SMS)- এর মাধ্যমে ডেটা(Data) পাঠাচ্ছিল। যদি ইন্টারনেট(Internet) সংযোগ থাকে, তুমি বার্তাটি যেকোনো জায়গায় পাঠাতে পারো, যে কোন মোবাইল নাম্বার(Mobile Number) এবং এর কোনো সীমিত পরিধি নেই। কিন্তু, এখানে 10 মিটার(metre) অবধি পরিধি সীমিত।

সাধারণত, ব্লুটুথ(Bluetooth) এর ক্ষেত্রে ডেটা(Data) আদান প্রদানের হার 2 থেকে 4 Mbps - এর আশেপাশে ঘোরাফেরা করে। এবং ব্লুটুথ(Bluetooth) 3.0 - র ক্ষেত্রে এটা 24 Mbps, আরো কিছু নতুন সংস্করণও এসেছে। আরডুইনো(Arduino)- র উপর্যুক্ত ব্লুটুথ(Bluetooth) মডিউল আছে, এবং অতি অবশ্যই এসটিএম(STM) উপর্যুক্ত। ব্লুটুথ(Bluetooth) মডিউল(module) গুলি যে কোন বোর্ডের সাথে খাপ খেয়ে যায়, তোমাকে কেবল সঠিক পদ্ধতিতে লাগাতে হবে এবং সঠিক কমান্ড(Command) ব্যবহার করতে হবে।

(Refer Slide Time: 17:35)

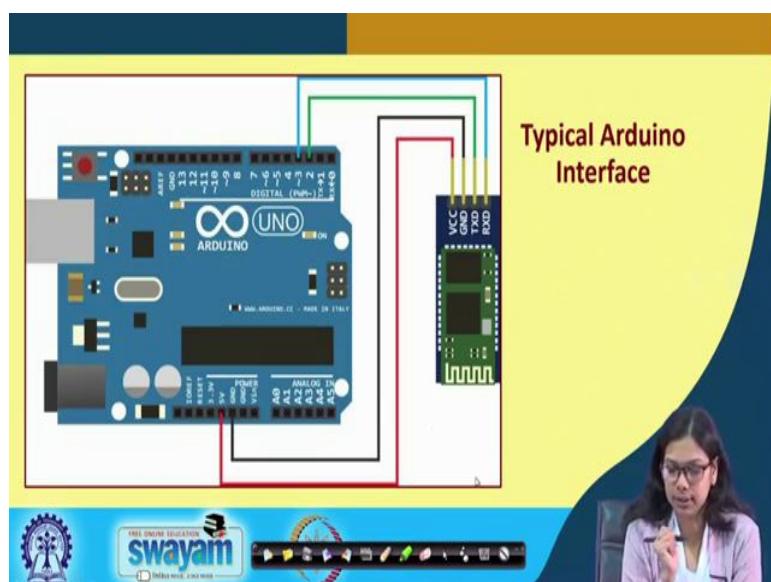


এই HC-06 ব্লুটুথ(Bluetooth) মডিউল(module)টি দেখা যাক। এইটা হচ্ছে ব্লুটুথ মডিউল(Bluetooth module) যেটা আমরা আমাদের পরীক্ষায় ব্যবহার করেছি ঠিক আছে।

এইটা হচ্ছে 5 ভোল্ট(Volt), গ্রাউন্ড(Ground), এবং এইটা হচ্ছে টিএক্স(TX) এবং এটা হচ্ছে আরএক্স(RX)। এইটা হচ্ছে class-2 ব্লুটুথ মডিউল(Bluetooth module) যেটাতে আছে সিরিয়াল(Serial) ডেটা(Data) ইন্টারফেস(interface) যেটি কিনা মাস্টার(master) অথবা স্লেব(slave) হিসেবে ব্যবহৃত হতে পারে।

বস্তুত, এগুলো কিছু প্রযুক্তিগত বিবরণী, যা আমি ইতিপূর্বে আলোচনা করেছি, যেমন 2.4 Ghz কম্পাঙ্ক(frequency), +3.3 ডিসি(DC) পাওয়ার সাপ্লাই(Power Supply), সর্বোচ্চ হার 3 Mbps, এবং স্লিপ মোড(Sleep Mode) বড রেট(Baud Rate)- এর মান হচ্ছে 9600।

(Refer Slide Time: 18:49)



তো এই হচ্ছে, আরডুইনো(Arduino) সংযোগ, যেখানে তুমি দেখছো Vcc 5 ভোল্ট(Volt) এর সাথে যুক্ত, গ্রাউন্ড(Ground) গুলিও একে অপরের সাথে যুক্ত, দুই নম্বর পিন তথা আরএক্স(RX) এর সাথে টিএক্স(TX)সংযুক্ত। এবং তিন নম্বর পিন তথা টিএক্স(TX) এর সাথে আরএক্স(RX) সংযুক্ত।

ধন্যবাদ।

Glossary

English Word	Transliteration	meaning
channel	চ্যানেল	একটি মাধ্যম যার দ্বারা গ্রহীতার কাছে বার্তা পাঠানো হয়।
microcontroller	মাইক্রোকন্ট্রোলার	এক প্রকার বৈদ্যুতিন যন্ত্র
digitize	ডিজিটাইজ	একটি সিগন্যালকে ডিজিটাল রূপে রূপান্তরিত করা
data	ডেটা	তথ্য
signal	সিগন্যাল	সময়ের সাথে সাথে পরিবর্তন হয় এমন তড়িৎ চুম্বকীয় তরঙ্গ ,যা তথ্য বহন করে
gateway	গেটওয়ে	এক প্রকার হার্ডওয়ার যন্ত্রপাতি যা একটি বিচ্ছিন্ন নেটওয়ার্ক থেকে অপর আরেকটিতে তথ্য আদান-প্রদানে সাহায্য করে
code	কোড	একপ্রকার নাম্বার যা কোনো কিছু চিহ্নিতকরণে সাহায্য করে
bit	বিট	তথ্যের একক
band	ব্যান্ড	বর্ণালীতে কম্পাক্ষের প্রসার
engine	ইঞ্জিন	সফটওয়ারের বিভিন্ন অংশের সম্মিলিত এক প্রকার মানসিক রূপরেখা
transmitter	ট্রান্সমিটার	তড়িৎ চুম্বকীয় তরঙ্গ প্রেরক যন্ত্র।
receiver	রিসিভার	গ্রহীতা
port	পোর্ট	একজোড়া টার্মিনাল যা

		একটি ইলেক্ট্রিক্যাল নেটওয়ার্ককে কোন বাহ্যিক বর্তনীর সঙ্গে যুক্ত করে।
adaptor	এডাপ্টার	এক প্রকার বৈদ্যুতিক যন্ত্রপাতি যা পরিবর্তী তড়িৎ প্রবাহকে একমুখী তড়িৎ প্রবাহে রূপান্তরিত করে।
headphone	হেডফোন	যে যন্ত্রটি কানে পড়লে শব্দ শোনা যায়।
Volt	ভোল্ট	তড়িৎচালক বল মাপবার একক
master	মাস্টার	যে ডিভাইস বা বর্তনী নিয়ন্ত্রক হিসেবে কাজ করে।
slave	স্লেব	যে ডিভাইস বা বর্তনী নিয়ন্ত্রক দ্বারা নিয়ন্ত্রিত হয়।
Baud Rate	বড রেট	একটি কমিউনিকেশন চ্যানেল দ্বারা যে হারে ডাটা যাতায়াত করে।

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science & Engineering
National Institute of Technology, Meghalaya

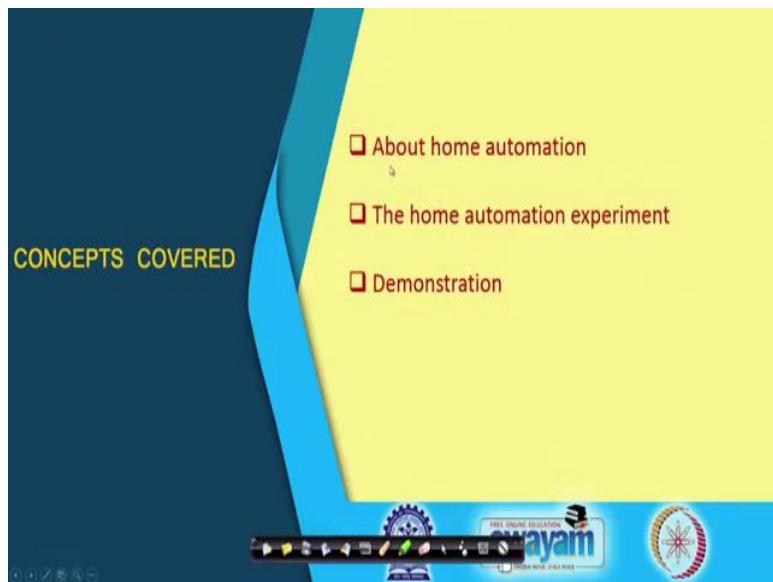
Lecture – 36
Design of Home Automation System

আজকের আলোচনায় সবাইকে স্বাগত। এই আলোচনায় আমি একটি স্বয়ংক্রিয় গৃহ পরিচালন ব্যবস্থা নিয়ে বক্তব্য রাখবো। আমি এখানে একটা বাতির উদাহরণ নেব এবং এসএমএস(SMS)-র মাধ্যমে বাতিটিকে জ্বালাবো।

অবশ্যই আমরা এটি একটি প্রোগ্রাম(Program) দ্বারাও নিয়ন্ত্রণ করতে পারতাম, কিন্তু এখানে আমরা এসএমএসের (SMS) মাধ্যমে করছি যাতে দুনিয়ার যেকোন জায়গা থেকে আইওটি(IoT)-র আস্তানাটা অনুভব করতে পারো। তুমি যদি “অন”(ON) বার্তা লিখে এসএমএস পাঠাও এই ডিভাইসটিতে(Device), তাহলে ডিভাইস(Device)-টি চালু হয়ে যাবে। এবং তুমি যদি “অফ”(OFF) বার্তা লিখে পাঠাও এসএমএস এর মাধ্যমে, ডিভাইস(Device)- টি বন্ধ হয়ে যাবে।

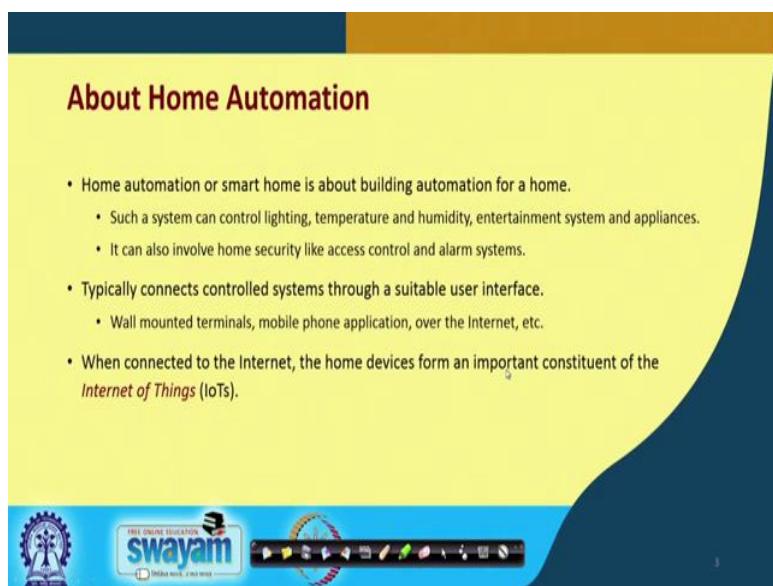
সবার প্রথমে আমি দুধরনের কোড(Code) আলোচনা করব। তার মধ্যে একটিতে, আমি যে কোন মোবাইল(Mobile) থেকে এসএমএস (SMS) পাঠালেই এটি কাজ করবে। এবং অপরটিতে একটি নির্দিষ্ট মোবাইল নাম্বার(Mobile Number) থেকে পাঠানো এসএমএস(SMS)- এই এটি সীমাবদ্ধ থাকবে, কারণ তুমি চাওনা যেকেউ তোমার ঘরের বাতি জ্বালিয়ে দিক। তো তুমি চাও যে শুধুমাত্র তোমার পরিবারের লোকজনই এই কাজটি করতে পারবে। তাহলে তুমি একটা সুরক্ষিত কোড(Code)- এর মধ্যে এই নাম্বার শুলোকে একত্রিত করতে পারো যেখানে শুধুমাত্র নির্দিষ্ট মোবাইল নাম্বার থেকে এসএমএস এলে তবে বাতি জালানো- নেভানো হবে। তো, এই দুটো জিনিস আলোচনায় আমি বলব।

(Refer Slide Time: 02:33)



এবার আমি স্বয়ংক্রিয় গৃহ পরিচালন ব্যবস্থা পরীক্ষা-নিরীক্ষাটি নিয়ে কথা বলব এবং তারপর
আমি প্রদর্শন করব।

(Refer Slide Time: 02:41)

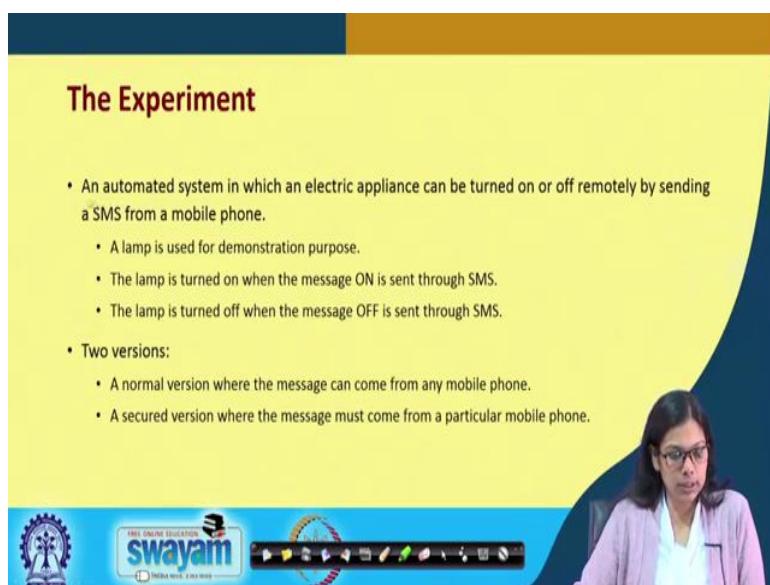


গৃহ স্বয়ংক্রিয় ব্যবস্থা অথবা স্মার্ট গৃহ(Smart home) হল গৃহটির জন্য স্বয়ংক্রিয় ব্যবস্থা গড়ে
তোলা। অবশ্যই, এটা শুধুমাত্র বাতি জালানো-নেডানোতেই সীমাবদ্ধ নয়, আরো অনেক জিনিস
আছে যেগুলি করা যেতে পারে। এইরকম একটা ব্যবস্থা আলো, তাপমাত্রা, আর্দ্রতা ও সরঞ্জাম সবই

নিয়ন্ত্রণ করতে পারো। তোমার বাড়িতেও অনেক যন্ত্রপাতি আছে যা তুমি নিয়ন্ত্রণ করতে পারো। এটির সঙ্গে যোগ হতে পারে গৃহ নিরাপত্তা যথা প্রবেশাধিকার নিয়ন্ত্রণ এমনকি আপৎকালীন ব্যবস্থা পর্যন্ত।

একটি উপযুক্ত ব্যবহারিক ইন্টারফেস(Interface)-এর মাধ্যমে এটি একটি নিয়ন্ত্রিত সিস্টেম(System)- কে সংযুক্ত করে। তুমি ইন্টারনেট(Internet) এর মাধ্যমে ব্যবহার করা যায় এমন কোন মোবাইল ফোন অ্যাপ্লিকেশন(Mobile Phone Application) অথবা টার্মিনাল(Terminal) ব্যবহার করতে পারো। ইন্টারনেট(Internet)- এর সাথে সংযুক্ত হলে গৃহের ডিভাইস(Device) গুলি ইন্টারনেট অফ থিংস(Internet of Things)- এর একটি খুবই গুরুত্বপূর্ণ অঙ্গ হয়ে উঠবে। আমি তো তোমাদের বলেছি, যে গৃহ পরিচালন ব্যবস্থার কথা আমি এখানে আলোচনা করব, তা হল যে আমি এসএমএস(SMS) দ্বারা একটি বাতিকে জ্বালাবো ও নেতৃত্ব দিবো।

(Refer Slide Time: 04:09)



The Experiment

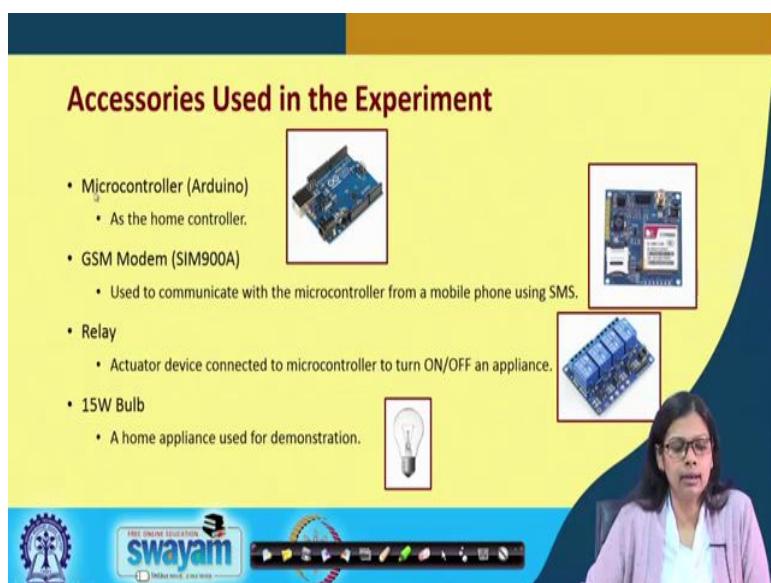
- An automated system in which an electric appliance can be turned on or off remotely by sending a SMS from a mobile phone.
- A lamp is used for demonstration purpose.
- The lamp is turned on when the message ON is sent through SMS.
- The lamp is turned off when the message OFF is sent through SMS.
- Two versions:
 - A normal version where the message can come from any mobile phone.
 - A secured version where the message must come from a particular mobile phone.

SWAYAM

পরীক্ষাটি এইরকম ভাবে হবে; একটি স্বয়ংক্রিয় ব্যবস্থা যাতে কিনা দূর থেকে এসএমএস(SMS)- এর মাধ্যমে একটি বৈদ্যুতিন সরঞ্জাম জ্বালানো ও বন্ধ করা যাবে। এক্ষেত্রে প্রদর্শনের জন্য একটি বাতি ব্যবহৃত হয়েছে। বাতিটি জ্বলে উঠবে যদি এসএমএস(SMS)- এর মাধ্যমে “অন”(ON) বার্তাটি

পাঠানো হয়। এবং যদি আমি এসএমএস(SMS)- এর মাধ্যমে অফ(OFF) বার্তাটি পাঠাই তাহলে বাতিটি বন্ধ হয়ে যাবে। এবং আমি যেরকম বলেছি যে এর দুটি সংস্করণ আছে। একটি হচ্ছে সাধারণ সংস্করণ, যেখানে যেকোনো মোবাইল ফোন(Mobile Phone) থেকে বার্তা আসতে পারে। এবং অবশ্যই একটি সুরক্ষিত সংস্করণ, যেখানে একটি নির্দিষ্ট মোবাইল ফোন(Mobile Phone) থেকেই বার্তা আসতে পারে।

(Refer Slide Time: 05:01)

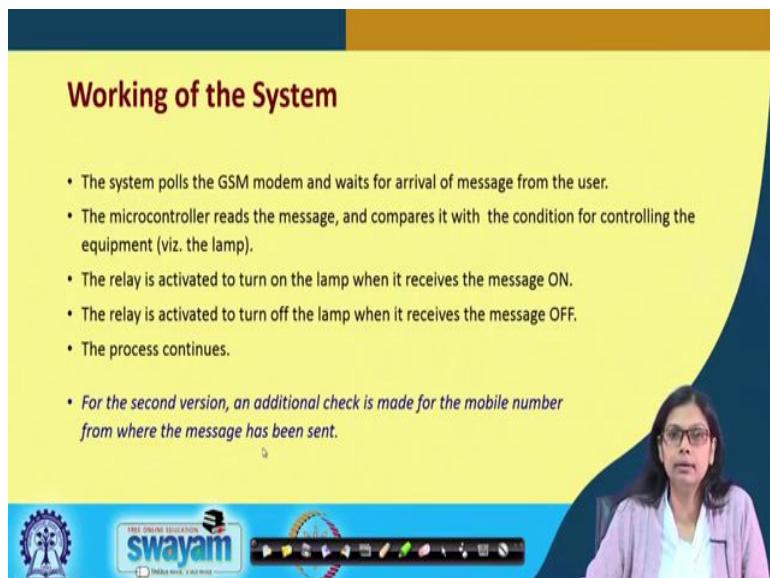


কি কি হার্ডওয়ার(Hardware) যন্ত্রাংশ লাগবে এই পরীক্ষার জন্য? আমাদের অবশ্যই একটি মাইক্রোকন্ট্রোলার(Microcontroller) লাগবে; এই ক্ষেত্রে আমরা ব্যবহার করেছি আরডুইনো উনো(Arduino Uno)। একটি জিএসএম(GSM) মডেম(Modem)-ও প্রয়োজন; আমরা ব্যবহার করেছি SIM900A জিএসএম(GSM) মডিউল(Module), যেটা কিনা এসএমএস(SMS)-এর মাধ্যমে মোবাইল ফোন (Mobile Phone)থেকে মাইক্রোকন্ট্রোলার(Microcontroller) এর সাথে যোগাযোগ করতে সাহায্য করবে।

আমাদের একটি ইলেক্ট্রোমেকানিক্যাল(Electromechanical) ডিভাইস(Device) তথা রিলে(Relay), এখানে প্রয়োজন যা কিনা পূর্বে আলোচিত। এই পরীক্ষা-নিরীক্ষায় এটি একটি

একচুয়াটর(Actuator) হিসেবে কাজ করছে, যেটি মাইক্রোকন্ট্রোলার(microcontroller) এর সাথে সংযুক্ত হয়ে সরঞ্জামটিকে চালু করছে ও বন্ধ করছে। এবং আমরা 15 ওয়াটের(Watt) একটি বাতি ব্যবহার করেছি প্রদর্শনের জন্য তোমরা অন্য যেকোন ডিভাইস(Device)-ও ব্যবহার করতে পারো।

(Refer Slide Time: 06:03)



Working of the System

- The system polls the GSM modem and waits for arrival of message from the user.
- The microcontroller reads the message, and compares it with the condition for controlling the equipment (viz. the lamp).
- The relay is activated to turn on the lamp when it receives the message ON.
- The relay is activated to turn off the lamp when it receives the message OFF.
- The process continues.
- For the second version, an additional check is made for the mobile number from where the message has been sent.

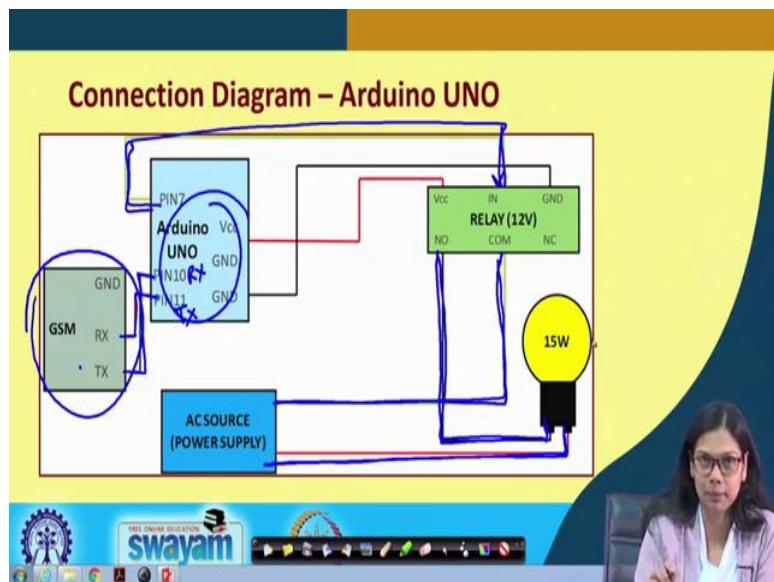
At the bottom of the slide, there is a video player interface with a woman speaking, the 'SWAYAM' logo, and playback controls.

সিস্টেম(System)-টির কাজকর্ম নিম্নলিখিত পদ্ধতিতে চলবে। সিস্টেম(System)-টি জিএসএম(GSM) মডেম(Modem) থেকে তথ্য সংগ্রহ করবে এবং ব্যবহারকারীর তরফ থেকে একটি বার্তার আগমনের অপেক্ষা করবে। এরপর মাইক্রোকন্ট্রোলার(Microcontroller) বার্তাটি সংগ্রহ করবে, সরঞ্জাম নিয়ন্ত্রণের যে শর্ত রয়েছে তার সাথে তুলনা হবে।

সেই মতো, সেটি যা পাবে অন(ON) অথবা অফ(OFF)। রিলে(Relay)-টি অন(ON) বার্তা পাবে তখন সেটি সক্রিয় হয়ে উঠবে বাতিটিকে জ্বালানোর জন্য। এবং যখন রিলে(Relay)টি অফ(OFF) বার্তা পাবে তখন আবার সেটি সক্রিয় হয়ে উঠবে বাতিটিকে বন্ধ করার জন্য। এবং এইভাবে গোটা পদ্ধতিটা চলতে থাকবে। তো এইটাই সমগ্র সিস্টেম(System)-টির কর্মপদ্ধতি। এবং আমি ঘেরকম

বললাম যে দ্বিতীয় সংস্করণটির জন্য, যে মোবাইল নাম্বার(Mobile Number) থেকে বার্তাটি পাঠানো হয়েছে সেটিকে বাড়তি যাচাই করা হবে।

(Refer Slide Time: 07:01)



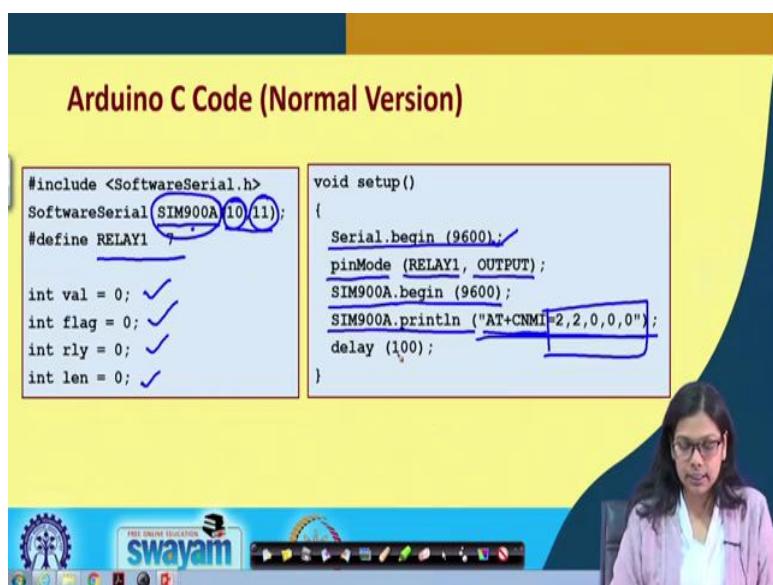
এবার বরঞ্চ সংযোগটি দেখা যাক। এইটা হচ্ছে রিলে(Relay) এবং আমরা জানি যে রিলে(Relay)-তে আছে একটি Vcc, একটি গ্রাউন্ড এবং একটি কন্ট্রোল ইনপুট (Control Input) যেটি আরডুইনো(Arduino)-র সাত নম্বর পিন(Pin) -এর সাথে সংযুক্ত।

এবং এটি হচ্ছে বাতিটি এবং এটি হচ্ছে আমার এসি(AC) উৎস। বাতিটির একটি প্রান্ত রিলে(Relay)-টির নো আউটপুট(No Output) পিন(Pin) এর সাথে সরাসরি যুক্ত। বাতির অপরপ্রান্তি এসি(AC) উৎসের সাথে যুক্ত। এবং এসি(AC) উৎস টির অপরপ্রান্ত রিলে- র কম আউটপুট(COM output) পিন এর সাথে যুক্ত। সাত(7) নম্বর পিন(Pin) ইনপুট (Input)-এর সাথে যুক্ত।

এবং জিএসএম(GSM) বোর্ড(Board)- টির সাথে, যেটি কিনা অন(ON) অথবা অফ(OFF) লিখিত বার্তা গ্রহণ করবে তার সাথে এইভাবে যুক্ত হবে; এইটির আরএক্স(RX) পিন(Pin) 11 নম্বর পিন তথা টি এক্স(Tx)- এর সাথে যুক্ত হবে। এবং এইটির টি এক্স(Tx) পিন যুক্ত হবে 10 নম্বর পিন তথা

আর এক্স(RX) এর সাথে এই -ই হল সংযোগের ব্যাপারে সমস্ত কিছু, এবং অবশ্যই গ্রাউন্ড(ground) ও Vcc সেই মতো যুক্ত হবে। জিএসএম(GSM) একটি এসএমএস(SMS) পাবে এবং তার উপর নির্ভর করে সেটি কিছু প্রক্রিয়াকরণ করবে এবং সাত নম্বর পিন(Pin)- এর মাধ্যমে রিলে (Relay)তে একটি সংখ্যা পাঠাবে। এবং সেইমতো রিলে(Relay)-টি বাতিটিকে জ্বালাবে অথবা বাতিটিকে নিভিয়ে দেবে, আমরা যে শর্ত দিয়েছি, তার উপর নির্ভর করে।

(Refer Slide Time: 09:43)



এবার দেখা যাক কোড(Code)- টি কিভাবে কাজ করছে। তো তোমাদেরকে এই সফটওয়্যার সিরিয়াল(Software Serial)- টা যোগ করতে হবে, তোমাদেরকে সিম900A(SIM900A)- র সাথে আরএক্স(RX) টিএক্স(TX) পিন এর জন্য 10 ও 11 রাখতে হবে। এবং এই RELAY1- র মান হবে সাত। তারপর আমরা আরো কিছু ভ্যারিয়েবল(Variabile) তথা val, flag, relay, len, ইত্যাদি সমস্ত কিছুর মান শূন্য দিয়ে শুরু করবো। আমরা একে একে দেখব এই ভ্যারিয়েবল(variable) গুলি কেন প্রয়োজন।

প্রস্তুতি পর্যায়ে, মনে করো যে, সিরিয়াল(Serial) যোগাযোগ ব্যবস্থার জন্য আমাদের আলাদাভাবে আরডুইনো(Arduino)-র জন্য কোন হাইপার টার্মিনাল(hyper Terminal) লাগবেনা, কিন্তু

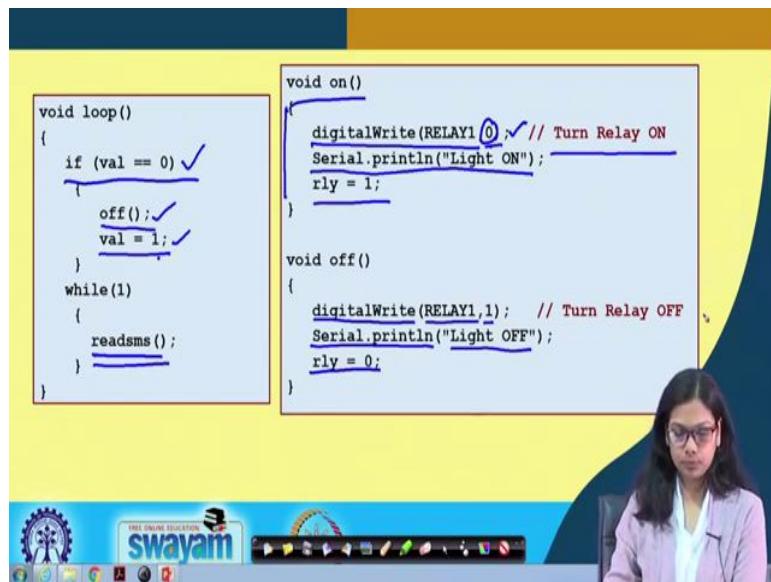
এসটিএম(STM) এর জন্য আমাদের তা প্রয়োজন। আরডুইনো(Arduino)- র ক্ষেত্রে আমরা সিরিয়াল(Serial) মনিটর(Monitor)- এ সরাসরি দেখাতে পারি।

তো প্রস্তুতিপর্বে আমরা প্রথমে সিরিয়াল(Serial) প্রিন্টিং(Printing) এর জন্য বড রেট(Baud rate) ব্যাখ্যা করি যথা `Serial.begin(9600)| PinMode RELAY1-কে আউটপুট(Output)` পোর্ট(Port) হিসেবে নিযুক্ত করো।আমরা আবারও বড রেট(Baud Rate) টাকে 9600 রাখছি এবং একটা গুরুত্বপূর্ণ জিনিস ঘেটা আমরা করছি, এখন যখন একবার এই নির্দিষ্ট অবজেক্ট(Object) `SIM900A.begin(9600)` টিকে বানিয়ে ফেলেছি।

আমাদেরকে এটাও নির্দিষ্ট করে দিতে হবে যে, যে জিএসএম(GSM) মডিউল(Module) আমরা ব্যবহার করছি, সেটা বার্তা আদান প্রদানের জন্য আমরা ব্যবহার করব তো সেই নির্দিষ্ট জিনিসটি আমাদেরকে ব্যাখ্যা করতে হবে। সেই নির্দিষ্ট কারণের জন্য, আমাদেরকে একটা এটি(AT) কমান্ড(Command) ব্যবহার করতে হবে।

সিএন এমআই(CNMI) অর্থাৎ কমান্ড নেম মেসেজ ইন্ডিকেশন (Command Name Message Indication)। এর মানে দাঁড়ায় আমরা এখানে বার্তা গ্রহণ করতে পারব। এবং তারপর 100 মিলি সেকেন্ড(millisecond) একটা বিরতি দেব।

(Refer Slide Time: 12:57)

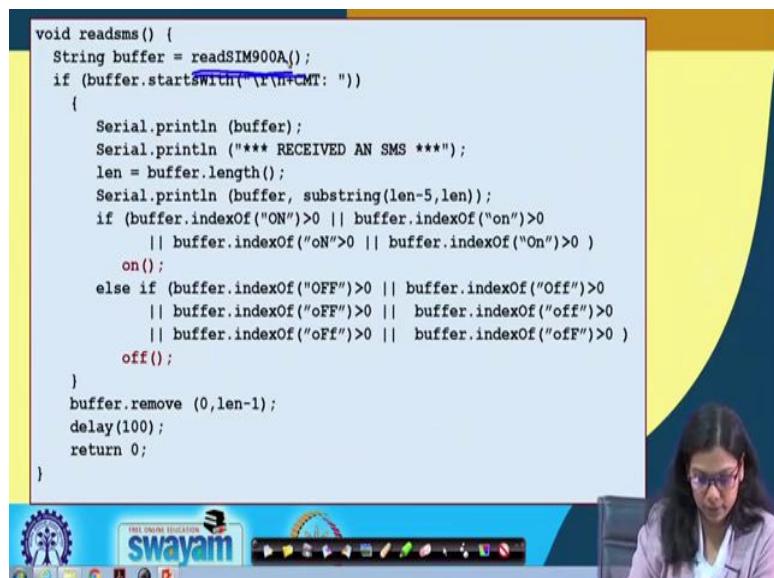


এইটা গেল প্রস্তুতিপর্ব। এবার লুপ (Loop) এর মধ্যে কি করছি আমরা শুরুতে আমরা ড্যারিয়েবেল (variable) `val` টাকে যাচাই করছি। `val`- এর মান এখন শূন্য। আমি এটাকে বন্ধ করে দিচ্ছি এবং তারপর আমি আবার `val`- এর মান এক(1) রাখবো। তো, এটা লুপ(Loop)- এর মধ্যে চুকবে না কারণ `val` এর মান এখন 1 এর সমান। তারপর `while(1)` লুপের(Loop) মধ্যে, এসএমএস(SMS) টাকে পড়ছি।

তো, আমি দেখব এই `readsms()` ফাংশন(Function)-টি কি কাজ করে। আমি বরঞ্চ বলি তোমাদের এই অন(ON) এবং অফ(OFF) ফাংশন(Function)-টি কিভাবে কাজ করে; আসলে অন(ON) এবং অফ(OFF) ফাংশন(Function)-টি রিলে(Relay)- টিকে চালু ও বন্ধ করবে। তোমরা যদি রিলে (Relay)- টিকে চালু অথবা বন্ধ করো, বাতিটিও তাহলে জ্বলবে ও বন্ধ হবে। অন(ON) ফাংশনের (Function) মধ্যে, আমরা করি একটা `digitalWrite (RELAY1,0)`, রিলে (Relay)-টিকে চালু করার জন্য। এবং এই `Serial.println`, এটি সিরিয়াল(Serial) মনিটরে(Monitor) “লাইট অন”(Light On) দেখাবে।

এবং তারপর আমরা আমরা "rly" এর মান এক(1) রাখছি। এবং অফ(OFF) ফাংশনের(Function) মধ্যে, আমরা করছি একটা digitalWrite (RELAY1, 1), যেটি কিনা রিলে(Relay)- টিকে বন্ধ করে দেবে। এবং এটি সিরিয়াল(Serial) মনিটরে(Monitor) দেখাবে " লাইট অফ"(Light Off)। এবং এটি "rly"- কেও শূন্য করে দেবে।

(Refer Slide Time: 15:45)



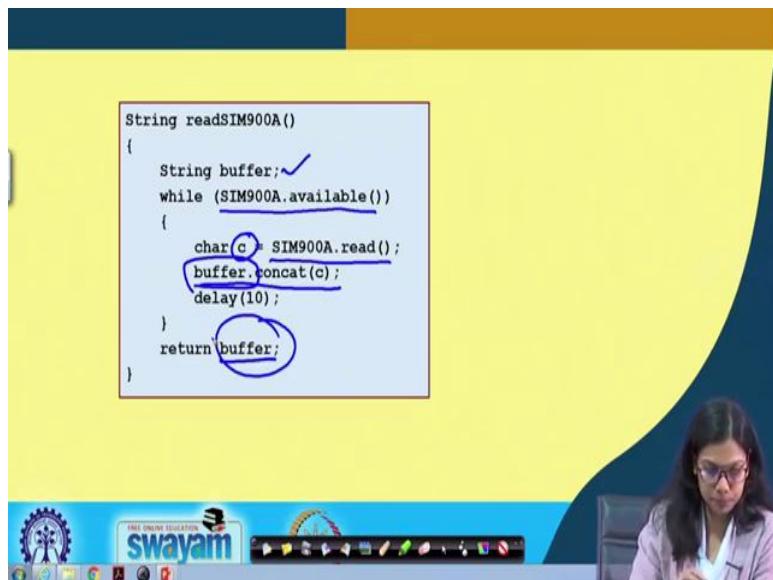
```

void readsms() {
    String buffer = readSIM900A();
    if (buffer.startsWith("rly(MCMT: "))
    {
        Serial.println (buffer);
        Serial.println ("*** RECEIVED AN SMS ***");
        len = buffer.length();
        Serial.println (buffer, substring(len-5,len));
        if (buffer.indexOf("ON")>0 || buffer.indexOf("on")>0
            || buffer.indexOf("oN")>0 || buffer.indexOf("On")>0
            || buffer.indexOf("on")>0 || buffer.indexOf("On")>0 )
            on();
        else if (buffer.indexOf("OFF")>0 || buffer.indexOf("Off")>0
            || buffer.indexOf("oFF")>0 || buffer.indexOf("off")>0
            || buffer.indexOf("oFf")>0 || buffer.indexOf("oFf")>0 )
            off();
    }
    buffer.remove (0,len-1);
    delay(100);
    return 0;
}

```

এরপর হচ্ছে **readsms** ফাংশন(Function)। আরো একটি ফাংশন(Function) আছে যেটা আমরা লিখেছি, **readsim900A**, এটাকে মনোযোগ সহকারে দেখা যাক। **readsms**-এ আমরা কি করছি, না আমরা অনবরত এইটা কে ওই লুপ(Loop)- টার মধ্যে চালাচ্ছি।

(Refer Slide Time: 16:31)



`readsim900A()` তে যাওয়া যাক, এটা কি করছে, এটা `buffer`- নামক একটি স্ট্রিং(String) ব্যাখ্যা করছে। এবং তারপর এই `while condition`- এ যাচাই করছে যে যোগাযোগ ব্যবস্থা আছে কি নেই।

যদি থাকে তাহলে এটি `sim900A.read` কে পড়ছে এই ফাংশন(Function) টির সাহায্যে এবং `C`-তে সঞ্চয় করছে। এবং এই `buffer.concat`, আমরা এক এক করে অক্ষরগুলোকে পড়ছি এবং আমরা এই `buffer.concat(c)`- এর সাথে যোগ করছি। এবং আমরা 10 মিলি সেকেন্ডের(millisecond) একটা বিরতি নিচ্ছি। এবং অবশেষে, আমরা এই `buffer`-টিকে ফেরত পাঠিয়ে দিচ্ছি।

এবার আমরা পূর্ববর্তী স্লাইড(Slide) টিতে ফেরত যাবো যেখানে আছে `readsms`। আমরা সেই ফাংশন(Function) তাকে ডাকছি যেটা আমি এক্সুনি দেখালাম, এবং আমরা সেটাকে `buffer`- এ সঞ্চয় করছি। এবং তারপর আমরা যাচাই করছি `buffer` এই স্ট্রিং(String)-টা দিয়ে শুরু হচ্ছে কিনা।

এবার আবার আমরা প্রদর্শন করছি আমরা যেটা পেয়েছি। এবং আমরা দেখছি যে একবার এটা দেখানো হয়ে গেল, এসএমএস(SMS) গ্রহণ করা হয়ে গেল আমরা সিরিয়ালে(Serial) এটা

দেখাচ্ছি । এবং এই জিনিসগুলো শুধুমাত্র আমাদের যাচাইয়ের জন্য আছে, তুমি সবকিছুর শেষে এগুলোকে মুছে দিতে পারো । এরপর আমরা buffer- এর দৈর্ঘ্য হিসাব করছি । দৈর্ঘ্য হিসেব করার পর, আমরা সাবস্ট্রিং(SubString)- টি নি এবং দৈর্ঘ্য থেকে পাঁচ বাদ দিয়ে যা পড়ে থাকে সেই সম দৈর্ঘ্যের একটি স্ট্রিং(String) এর থেকে বের করে নি।

এবং এখন আমরা কি করছি, যেই একবার আমার কাছে এসএমএস(SMS) চলে এলো, এর সঙ্গে অন্যান্য বর্ণ চলে আসছে । আমরা সেই অংশটা কে বাদ দিয়ে দিচ্ছি, এবং এখন নজর দেবো এসএমএস(SMS)- এর আসল অংশটির দিকে । এবাবে যদি buffer.indexOf অন হয়, indexOf এই buffer থেকে এই স্ট্রিং(String)- টি খোঁজার চেষ্টা করবে, এবং সেই নির্দিষ্ট স্থানটি বলে দেবে যেখানে এটি পাওয়া গেছে । তো সেই ক্ষেত্রে, এটি সবসময়ই শূন্যর থেকে বড় মান ফিরিয়ে দেবে । এবং সেটি যদি তা খুঁজে না পায় তাহলে সেটি শূন্যের চেয়ে কিছু কম তথা নেগেটিভ(Negative) এক (-1)ফেরত পাঠাবে।

তো আমাদের ক্ষেত্রে আমরা 0- র চেয়ে বড় একটা মান পাব, তো সেই জন্যই আমরা এখানে যাচাই করছি buffer.indexOf অন (ON)আছে কিনা । এটি ছোট ও বড়- র মিশ্রন হতে পারে এবং তারপর বড় ও ছোট- র । সুতরাং সমস্ত ধরনের সম্ভাবনাই আমরা খতিয়ে দেখেছি।

বার্তাটির মধ্যে যদি এই জিনিসগুলো থেকে কোন একটি থাকে, তাহলে এটি অন ফাংশন(Function)-টিকে ডাকবে, এবং এটি বাতিটি কে জ্বালবে। সেই রকম ভাবেই, আমরা অফ(OFF) এর শর্তও যাচাই করি, এখানে বিভিন্ন রকমের সম্ভাবনা আছে । তো, আমরা একটা এসএমএস(SMS) পাচ্ছি, আমরা সেটার কিছু অংশ বাদ দিচ্ছি, এবং তারপর আমরা আসল এসএমএস(SMS)-টা পাচ্ছি ।

এবং সেই নির্দিষ্ট এসএমএস(SMS) টি এ তে আমি দেখছি যে সেটিতে অন(ON) বার্তা আছে কি নেই । এবং সবশেষে এই buffer- থেকে আমরা পুরো স্ট্রিং(String)-টিকেই মুছে ফেলেছি, আমরা 100

মিলি সেকেন্ডের(millisecond) একটা বিরতি নিচ্ছি এবং তারপর ০ ফিরিয়ে দিচ্ছি। এই একই জিনিস অনবরত চলছে। এটা একটা সাধারণ কোড(Code); যেকোনো একটা মোবাইল নাম্বার(Mobile Number) থেকে যদি এসএমএস(SMS) আসে তখন এটি এইভাবে কাজ করবে।

(Refer Slide Time: 21:59)

Arduino C Code (Secure Version)

```
#include <SoftwareSerial.h>
SoftwareSerial SIM900A(10,11);
#define RELAY1 7

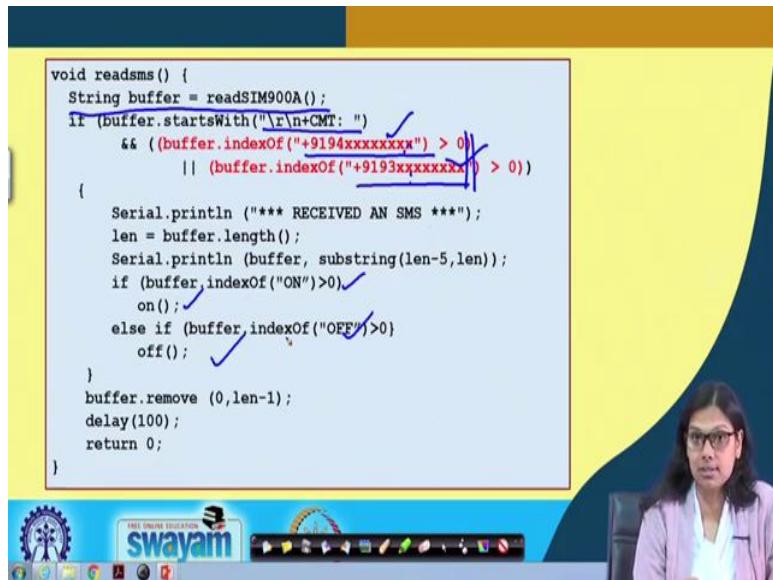
int val = 0;
int flag = 0;
int rly = 0;
int len = 0;
```

```
void setup()
{
    Serial.begin (9600);
    pinMode (RELAY1, OUTPUT);
    SIM900A.begin (9600);
    SIM900A.println ("AT+CNMI=2,2,0,0,0");
    delay (100);
}
```

The slide also features the Swayam logo and navigation icons at the bottom.

এবার আমরা পরের কোড(Code) টা দেখব, যেটা কিনা এর সুরক্ষিত সংস্করণ। সুরক্ষিত সংস্করণ অর্থাৎ, আমি আগেই বলেছি, এটি কোন নির্দিষ্ট নাম্বার(Number) থেকেই গ্রহণ করবে। এবং একমাত্র তখনই, আমি এটাকে ব্যবহার করব, আমি এটাকে জ্বালাবো অথবা বন্ধ করব। আমি যদি এটা অন্য কোন মোবাইল নাম্বার(Mobile Number) থেকে পাই আমি এটাকে জ্বালাবো না বা বন্ধ করব না। তো, কোড(Code) এর এই অংশটা একেবারেই সহজ সরল, এবং এই অংশটাও একই রকম হবে, আসল পার্থক্য শুরু হচ্ছে এখান থেকে।

(Refer Slide Time: 22:23)



আমরা দেখছি স্ট্রিং(String) টা এইটা দিয়েই শুরু হচ্ছে কিনা। এবং আমরা বাফার ডট ইন্ডেক্স টিকে. ও যাচাই করছি। এখানে আমি এই দুটো নাম্বার(Number) দ্বারা যাচাই করছি, তুমি আরো অনেক নাম্বারের(Number) জন্য করতে পারো।

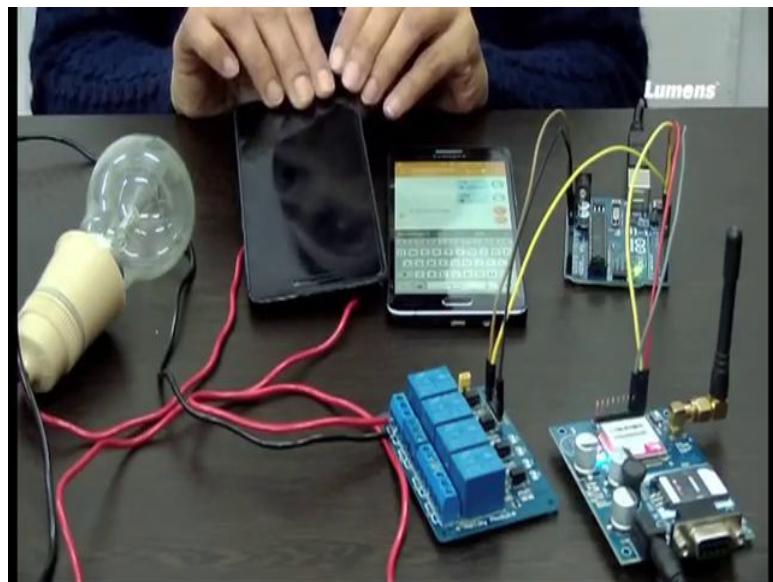
এইটা যদি এই নাম্বার(Number) অথবা এই নাম্বার(Number) থেকে পাঠায় তাহলে এটি কাজ করবে। কিন্তু এখানে যে নাম্বার(Number) গুলির উল্লেখ নেই, তাদের জন্য এটি কাজ করবে না। এবং অবশ্যে, buffer থেকে সরিয়ে দেবো। সুতরাং, এই হচ্ছে কোড(Code) টির দুটি সংস্করণ, যেগুলি কিনা বাতিকে এসএমএসের (SMS)মাধ্যমে জ্বালতে ও বন্ধ করতে সাহায্য করছে।

এবাবে যে পরীক্ষাটা আমরা এতক্ষণ আলোচনা করলাম সেইটা হাতে-কলমে করে দেখব আমরা।

তো এখন আমি তোমাদেরকে পরীক্ষাগুলো দেখাবো যেখানে আমি এসএমএস(SMS) দ্বারা একটি বাতিকে জ্বালাবো ও বন্ধ করব। আমি ইতিমধ্যেই তোমাদের সাথে কোড(Code)-গুলো আলোচনা করেছি। আমি ইতিমধ্যেই বলেছি যে বর্তনীটি কিভাবে বানাবে।

এখানে কোড(Code) টির দুটি সংস্করণ আছে যা আমরা করেছি। একটিতে যেকোনো কেউ পাঠাতে পারে, অপরটিতে তুমি কয়েকটা ফোন নাম্বার(Phone Number) দিতে পারো যাদের থেকে তুমি চাও যে এই জ্বালানো ও বন্ধ করার ঘটনাটি ঘটতে পারে। তো, এই দুটো জিনিস, আমি তোমাদের দেখাবো। তোমরা ইতিমধ্যেই জানো যে কিভাবে প্রোগ্রামের(Program) নিয়ন্ত্রণ দ্বারা একটি রিলে(Relay)- র সাহায্যে একটি বাতি জ্বালানো ঘটে পারে। সেখানে আমরা এসএমএস(SMS) ব্যবহার করছি না, বরঞ্চ শুধুমাত্র রিলে(Relay)- র নিয়ন্ত্রণ দ্বারা একটি একটি বাতিকে জ্বালাচ্ছি ও নেজাচ্ছি। তোমরা ইতিমধ্যেই জানো যে এটা কিভাবে কাজ করে। আমি শুধুমাত্র এটাকে জিএসএম (GSM) মডিউল(Module) টার সাথে একীভূত করব।

(Refer Slide Time: 25:59)



তো, এটা হচ্ছে বাতিটি। এই বাতিটি এই প্লাগ(Plug)-টির সঙ্গে যুক্ত হবে যেখানে কিনা প্লাগ(Plug)-টি একটি এসি (AC) উৎস, তোমাদের কাছে একটা রিলে(Relay) মডিউল(Module) থাকতেই হবে, এটা চারটে মডিউল(Module) বিশিষ্ট রিলে(Relay), তোমার কাছে একটি মডিউল(Module) বিশিষ্ট রিলে(Relay)-ও থাকতে পারে। তোমার কাছে অবশ্যই একটা

জিএসএম(GSM) মডিউল(Module) থাকতে হবে। এটা হচ্ছে SIM900 মডিউল(Module), আমি ইতিমধ্যেই এই মডিউল(Module) টির ব্যাপারে বিশদে আলোচনা করেছি। তোমার এটার জন্য একটা সিম কার্ড(Sim Card) লাগবে, যে সিম কার্ডটা(Sim Card) এখানে ডরা হয়েছে সেটা তোমরা দেখতে পাচ্ছ। এবং তোমার চাই একটা মাইক্রোকন্ট্রোলার(microcontroller) বোর্ড(Board) যেটা এই ক্ষেত্রে হচ্ছে আরডুইনো(Arduino) উনো(UNO)।

এবার দেখো সংযোগটা কিভাবে হচ্ছে। এসি(AC) উৎসের একটা দিক সরাসরি বাতিটির সঙ্গে যুক্ত হচ্ছে। এবং এসি(AC) উৎসের অপর বিন্দুটি, যেটা কিনা হচ্ছে এইটা, এইটা ইনপুট(Input) হয়ে রিলে(Relay)- র কম(COM) বিন্দুতে যাচ্ছে। এবং বাতিটির অপরপ্রান্তি রিলে(Relay)-র খোলা বিন্দুতে গিয়ে যুক্ত হচ্ছে।

এবার আমি দেখাবো, জিএসএম(GSM) বোর্ড(Board) ও মাইক্রোকন্ট্রোলার(microcontroller) বোর্ড(Board)- এর সাথে কি কি জুড়বে। জিএসএম(GSM) বোর্ড(Board)- এ ডুমি আছে। এই ডুমিটি আরডুইনো(Arduino)- র ডুমির সাথে যুক্ত। এবং এই জিএসএম(GSM) বোর্ড(Board) এ আর এক্স এবং টি এক্স পিন রয়েছে।

জিএসএম(GSM) এর টি এক্স(TX) পিন(Pin)টি আরডুইনো(Arduino) র আর এক্স(RX) পিন(Pin)-টির সাথে যোগ হবে। তো আমরা আরডুইনো(Arduino)- র জন্য আর এক্স(RX)ও টি এক্স(TX) দুটি পিন(Pin) বানিয়েছি, যেই গুলি হচ্ছে 10 নম্বর ও 11 নম্বর পিন(Pin)। তো, টি এক্স(TX) যুক্ত হবে আর এক্স(RX) - এর সাথে, এখানে এটা হচ্ছে 10 নম্বর পিন(Pin)। এবং আর এক্স(RX) পিন(Pin) টি যুক্ত হবে 11 নম্বর পিনের(Pin) সাথে যেটি হচ্ছে আরডুইনো(Arduino)- র টি এক্স(TX) পিন(Pin)।

এখন, অবশ্যে আমাকে রিলে(Relay)- র সাথে সংযোগটা করতে হবে। এখন যদি তুমি দেখো আসলে কি ঘটছে, দেখবে যে এসএমএস(SMS) পাঠালে বাতিটি জ্বলে উঠছে। তো, তোমার

জিএসএম(GSM) মডিউল(Module) এবং আরডুইনো(Arduino) বোর্ড(Board)- এর মধ্যে একটা সংযোগ থাকতে হবে | আরডুইনো(Arduino) বোর্ড(Board) টিকে এসএমএস(SMS) গ্রহণ করতে হবে, যার ওপর নির্ভর করে এটি রিলে(Relay)-টিকে চালু করবে অথবা বন্ধ করবে।

আমরা রিলে(Relay)- র ইনপুট(Input) আরডুইনো(Arduino)- র সাত নম্বর পিনটির(Pin) সাথে যুক্ত করেছি। এবং সেখান একটি আছে গ্রাউন্ড(Ground) এবং একটি আছে VCC। তো আমরা রিলে- র চতুর্থ ইনপুট(Input)-টি ব্যবহার করছি।

অবশ্যে, জিএসএম(GSM)- এর মাধ্যমে এসএমএস(SMS) আসবে। এবং মাইক্রোকন্ট্রোলার(microcontroller) নির্দিষ্ট কিছু কাজ করতে বলবে, হয় অফ(OFF) নয়তো অন(ON)। এবার আমরা কোড(Code) গুলি এক এক করে দেখব যাতে দুধরনের কাজ করা- ই আমরা দেখতে পাই।

সুরক্ষিত সংস্করণটি কিভাবে দেখাবো? আমি এই দুটি মোবাইল ফোন (Mobile Phone) ব্যবহার করব। প্রথমে আমি এই নির্দিষ্ট মোবাইল ফোন(Mobile Phone)-টি থেকে এসএমএস(SMS) পাঠাবো এবং তুমি দেখবে যে এটি কাজ করছে।

দ্বিতীয়তঃ, আমি একটা সুরক্ষিত যোগাযোগ দেখাবো যেখানে আমি প্রথমে এই মোবাইল(Mobile)-টি থেকে এসএমএস(SMS) পাঠাবো এবং তুমি দেখবে যে এটি কাজ করবে না। কিন্তু আমি যখন এই মোবাইল ফোন (Mobile Phone)থেকে পাঠাবো তখন এটি কাজ করবে। আমাকে দেখাতে দাও। আমাকে কোড(Code) টা এরমধ্যে ভরতে দাও।

এখন দেখো আমি একটা অন(ON) বার্তা পাঠাবো, মনে হচ্ছে এটার কিছু সমস্যা আছে। তো, আমি খালি অন(ON) করব এবং আমি পাঠিয়ে দেবো। তো তুমি দেখতে পাচ্ছা যে আমি এটি এই মোবাইল(Mobile) থেকে পাঠিয়েছি সুতরাং অবশ্যই এটা সুরক্ষিত কোড(Code) নয়। আমি

আবার অফ(OFF) পাঠাবো আমাৰ অন্য মোবাইল(Mobile) থেকে, যেটা হচ্ছে এটা সেটা কৰা যাক। তো আমি এটাকে অফ(OFF) কৰব ঠিক আছো। তো বাতিটি বন্ধ হয়ে গেল।

আমি আৱেকবাৰ দেখাবো। আমি অন(ON) পাঠাচ্ছি, এটি কাজ কৰছে। আমি আবার অফ(OFF) পাঠাবো, তুমি দেখছো এটি কাজ কৰছে, কিন্তু এটি সুৱাসিত যেটি সেটি নয়। তোমৰা যে রকম বললে যে দুটো মোবাইল(Mobile) থেকেই এটি কাজ কৰছে। এবাৰ আমি নিশ্চিত কৰবো এটা যে যদি আমি এই মোবাইল(Mobile) থেকে পাঠাই তাহলে এটি কাজ কৰবে না। এবং যখন আমি এই নির্দিষ্ট মোবাইল(Mobile)-টি থেকে পাঠাবো, তখনই এটা কাজ কৰবে।

আমি এবাৰ এই মোবাইল(Mobile)-টি চেষ্টা কৰি, আমি ইতিমধ্যেই কোড(Code) ভৱে দিয়েছি। তো আমি এবাৰ অন(ON) বাৰ্তা পাঠাচ্ছি, কিছুক্ষণ অপেক্ষা কৰা যাক কিছু হলো না তাইতো? এবাৰ আমি এই মোবাইল(Mobile)-টা দিয়ে চেষ্টা কৰি এবং এটি কাজ কৰলো। এটি হচ্ছে একটি সুৱাসিত যোগাযোগ ব্যবস্থা, যেখানে তুমি এই স্বয়ংক্রিয়ভাৱে অন(ON) অফ(OFF) কৰাৰ কাজটি কিছু নির্দিষ্ট মোবাইল(Mobile) নাম্বাৰ(Number)- এৰ মধ্যেই সীমাবন্ধ রাখছো এবং অন্যান্য মোবাইল নাম্বাৰ(Mobile Number) এৰ জন্য নয়।

তো, আজকে আমৰা একটা খুবই সাধাৰণ স্বয়ংক্রিয় ব্যবস্থা নিয়ে আলোচনা কৰলাম, সেটা আমৰা একটা **রিলে(Relay)**, একটা **জিএসএম(GSM)** বোর্ড, একটা মাইক্ৰোকন্ট্ৰোলার(microcontroller), এবং অবশ্যই, এটা যে কোন বৈদ্যুতিন ঘন্টপাতি ও হতে পাৱে, কিন্তু এখানে আমৰা একটা সাধাৰণ বাতি ব্যবহাৰ কৰেছি সেটাকে জালানো ও নেজানোৰ জন্য।

ধন্যবাদ।

English	Transliteration	Translation
---------	-----------------	-------------

Program	প্রোগ্রাম	এক প্রকার বিশেষ ভাষায় লেখা কম্পিউটারের জন্য নির্দেশবলি।
Actuator	একচুয়াটর	যে যন্ত্রের দ্বারা শক্তিকে চলনে রূপান্তরিত করা যায়।
Device	ডিভাইস	বিশেষ উদ্দেশ্যসাধনের যন্ত্র
Code	কোড	এক প্রকার বিশেষ ভাষায় লেখা কম্পিউটারের জন্য নির্দেশবলি।
mobile	মোবাইল	চলভাষ
interface	ইন্টারফেস	আধার
terminal	টার্মিনাল	এক প্রকার সফটওয়্যার ঘার মধ্যে কমান্ড লিখে লিখে কাজ করতে হয়
mobile phone application	মোবাইল ফোন অ্যাপ্লিকেশন	মোবাইল ফোন- এ যেসকল সফটওয়্যার থাকে।
internet	ইন্টারনেট	অন্তর্জাল
hardware	হার্ডওয়ার	
microcontroller	মাইক্রোকন্ট্রোলার	এক প্রকার ছোট কম্পিউটার।
modem	মডেম	একটি হার্ডওয়ার ডিভাইস যা কম্পিউটার থেকে আসা ডিজিটাল সিগন্যালকে এনালগ সিগনালে রূপান্তরিত করে।
Serial Monitor	সিরিয়াল মনিটর	আরডুইনো আইডি সফটওয়্যারটির একটি অংশ

		যাতে ঘেকোনো ধরনের আউটপুট দেখা যায়।
Sim Card	সিম কার্ড	এক ধরনের 1 ইন্টিগ্রেটেড সার্কিট যা কল করার সময় একটি মোবাইল ফোন কে বিশিষ্ট ভাবে সনাক্ত করতে সাহায্য করে
Variable	ভেরিয়েবল	একটি সত্তা যার মান পরিবর্তন হতে পারে।
Loop	লুপ	প্রোগ্রামিং ল্যাঙ্গুয়েজ এর একটি সত্তা যার ফলে কিছু নির্দেশ বারবার চলতে থাকে
String	স্ট্রিং	ইংরেজি বর্ণ গুচ্ছ
Port	পোর্ট	একজোড়া টার্মিনাল যা একটি ইলেক্ট্রিক্যাল নেটওয়ার্ককে কোন বাহ্যিক বর্তনীর সঙ্গে যুক্ত করে।

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science & Engineering
National Institute of Technology, Meghalaya

Lecture – 37
Design of a Simple Alarm System Using Touch Sensor

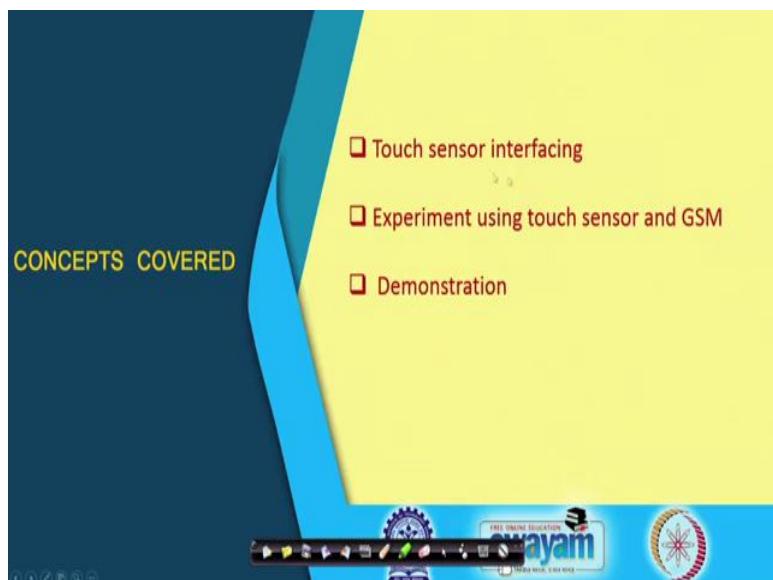
সকলকে এই লেকচারে(Lecture) স্বাগত জানাই। এই আলোচনায়, আমি টাচ(Touch) সেন্সর(Sensor) ব্যবহার করে একটি অতি সাধারণ এলার্ম(Alarm) সিস্টেম(System) তৈরীর পদ্ধতি দেখাব। এমন একটি পরিস্থিতি-র কথা চিন্তা করা যাক, যেখানে তুমি গরমের ছুটিতে অনেক দিনের জন্য বাড়ির বাইরে আছো, অবশ্যই ঘথাঘথ সুরক্ষার আয়োজন করে গেছো, কিন্তু তা সত্ত্বেও চুরির ঘটনা ঘটে। যাই হোক, যদি তুমি জানো তোমার বাড়িতে নির্দিষ্ট কিছু জায়গা আছে যেখান দিয়ে একটা লোক প্রবেশ করতে পারে; সেই দরজাটি যেখানে একটা তালা লাগানো আছে এবং কাউকে ঢুকতে হলে তালা ডেঙে ঢুকতে হবে, অথবা কোন জানলা থেকেও হতে পারে যে কোনো উপায়ে কাউকে জানলাটা ভাঙ্গতে হবে চুরি করার জন্য।

তো, তোমার বাড়িতে যে অসূরক্ষিত জায়গাগুলি আছে, সেখানে তুমি এরকম একটা ডিভাইস(Device) ব্যবহার করার কথা বিবেচনা করতে পারো যেটা একটি সামান্য সেন্সর(Sensor) ব্যতীত আর কিছুই নয়। তুমি যদি ওই সমস্ত জায়গাগুলিতে একটা টাচ(Touch) সেন্সর(Sensor) রেখে দিতে পারো এবং কেউ যদি সেটা স্পর্শ করে, এলার্ম(Alarm) বেজে উঠবে।

তো, সেইমতো তোমার একটা সিস্টেম(System) থাকতে পারে যেখানে তুমি অ্যাপ্লিকেশন(Application)-টার সাথে টাচ(Touch) সেন্সর(Sensor)-টা রাখতে পারো যেখানে ঘথনই কেউ সেন্সর(Sensor)-টাকে স্পর্শ করবে, এসএমএস(SMS) রূপে সতর্কবাণী তোমার মোবাইল(Mobile)-এ চলে যাবে। এবং তারপর তুমি একটা অ্যাপ(App) বানাতে পারো, যেখানে যদি এই নির্দিষ্ট নাম্বার(Number) থেকে এসএমএস(SMS) আসে, তাহলে এলার্মটি(Alarm) সক্রিয় হয়ে যাবে।

আমি এই নির্দিষ্ট জিনিসটি এখানে আলোচনা করব, যখনই কোন অঘাতিত ব্যক্তি তোমার দরজা স্পর্শ করবে তোমার অনুপস্থিতিতে, একটা এসএমএস(SMS) সতর্কবার্তা চলে যাবে তোমার কাছে।
পূর্ববর্তী উদাহরণে আমরা একটা মোবাইল(Mobile) থেকে তোমার সিস্টেম(System) এসএমএস(SMS) পাঠাচ্ছিলাম যার ফলে বাতিটি জ্বলছিল এবং বন্ধ হচ্ছিল। কিন্তু এখানে আমরা কি করছি? এখানে যদি সিস্টেম(System) ধরতে পারে যে চুরির চেষ্টা হয়েছে অথবা সেই রকম কিছু তখন সিস্টেম(System)-টি নির্দিষ্ট একটি মোবাইল নাম্বার(Mobile Number) এসএমএস (SMS)পাঠিয়ে দেবে।

(Refer Slide Time: 03:55)



এবার আমরা টাচ সেন্সর (touch sensor)-এর ব্যাপারে কথা বলব।
সেন্সর(Sensor)ইন্টারফেস(Interface)-টা খুবই সোজা সাপটা। অবশ্যে, আমি প্রদর্শন করব।

(Refer Slide Time: 04:17)

Introduction

- This experiment demonstrates another application of home automation system.
- Related to home security.
 - An automated alarm generation system that warns the owner of the house whenever an unauthorized access takes place.
- How is unauthorized access detected?
 - Through a touch interface circuit, similar to car intrusion alarm systems.
- What happens after the detection?
 - A suitable alarm system can be activated.
 - The owner can be alarmed by sending a SMS on his/her mobile phone.

যেরকম আমি বললাম, যে এটি গৃহ স্বয়ংক্রিয় ব্যবস্থার আরেকটি নির্দশন। এটার সঙ্গে সুরক্ষার ব্যাপারটাও একীভূত করা যেতে পারে। সুতরাং, এটি গৃহ সুরক্ষার সাথে সম্পর্কিত যেখানে একটা স্বয়ংক্রিয় এলার্ম সিস্টেম(Alarm System) গৃহ মালিক-কে সতর্ক করে দেয় যখনই অনভিপ্রেত প্রবেশাধিকার ঘটে।

যখনই অযাচিত প্রবেশ ঘটে, তখনই এটি এসএমএস(SMS) পাঠাবে।

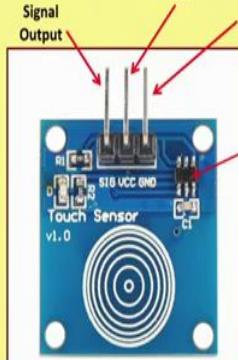
কিন্তু অনভিপ্রেত প্রবেশাধিকার কিভাবে নির্ণয় করবে? এখানে অযাচিত প্রবেশাধিকার নির্ণয় করা যাবে একটি টাচ ইন্টারফেস বর্তনীর(Touch interface circuit) মাধ্যমে যেটা অনেকটা Car Intrusion Alarm System -এর মত, যেখানে কেউ তোমার গাড়িকে স্পর্শ করলেই অ্যালার্ম(Alarm) বেজে উঠবে।

নির্ণয় -এর পরে কি হবে? বেশিরভাগ ক্ষেত্রে এলার্ম সিস্টেম(System) টি সক্রিয় হয়ে ওঠে এবং মালিককে তার মোবাইল (mobile)ফোন(phone)- এ এসএমএস(SMS) পাঠিয়ে সাবধান করে দেয়।

(Refer Slide Time: 05:41)

Interfacing the Touch Sensor Module

- The touch sensor module contains TTP223B, which is a 1-key touch pad detector IC, and it is suitable to detect capacitive element variations.
- The module has a +5V power supply input, a GND input, and a digital signal output that indicates the presence or absence of touch.



The slide also features a watermark for 'swayam' and a video player interface at the bottom.

এইটা হচ্ছে টাচ সেন্সর(Touch Sensor) -টা যেটা আমরা ব্যবহার করেছি। এখানে হচ্ছে Vcc, GND এবং এইটা হচ্ছে সিগন্যাল(Signal) আউটপুট(Output) যেটা এখানে আসছে। এবং এইটা হচ্ছে TTP223B; এইটা হচ্ছে সেই অঞ্চলটা যদি তুমি এই অংশটা স্পর্শ করো অথবা নিচ থেকে যদি স্পর্শ করো তাহলেও এটি কাজ করবে। টাচ সেন্সর (Touch Sensor) মডিউলটিতে(Module) আছে TTP223B যেটি হচ্ছে একটি বিশিষ্ট টাচপ্যাড(Touchpad) ডিটেক্টর(Detector) আইসি(IC) এবং ক্যাপাসিটিভ(Capacitive) পদার্থের তারতম্য নির্ণয়ে উপযুক্ত।

মডিউলটিতে(Module) আছে একটি 5 ভোল্ট(Volt) এর তড়িৎ উৎস এবং একটি গ্রাউন্ড(Ground) এবং একটি ডিজিটাল(Digital) সিগন্যাল(Signal) আউটপুট(Output) যা স্পর্শের দ্বারা উপস্থিতি বা অনুপস্থিতি ইঙ্গিত করবে। ডিজিটাল সিগন্যাল আউটপুট(Digital Signal Output)-টি এই স্পর্শের উপস্থিতি অথবা অনুপস্থিতি ইঙ্গিত করবে।

(Refer Slide Time: 06:45)

The Experiment

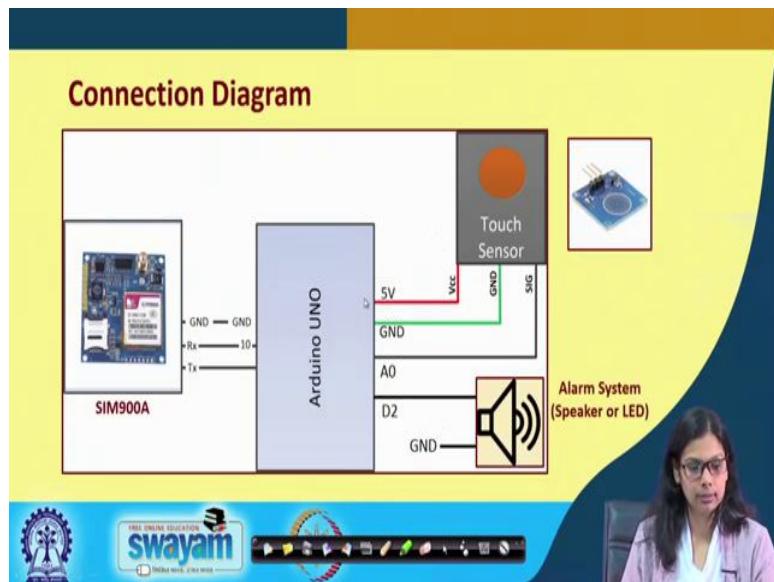
- An Arduino UNO microcontroller system connects to the touch sensor, and a GSM module (SIM900A).
 - The microcontroller continuously polls the touch sensor to find out whether an intrusion (i.e. a touch) has been detected.
 - If so, a suitable message is sent to the person concerned over SMS.
 - An alarm system is also activated.

FREE ONLINE EDUCATION
swayam
Digitize India, Educate India

A video player interface showing a person speaking, with a progress bar and a play button.

যে পরীক্ষাটি আমরা করেছি, তাতে আরডুইনো বোর্ড(Arduino Board) ব্যবহার করেছি যেখানে এই টাচ সেন্সর(Touch Sensor), জিএসএম(GSM) মডিউল(Module), ও মাইক্রোকন্ট্রোলার(Microcontroller) এক সঙ্গে যুক্ত করেছি। অনধিকার প্রবেশ ঘটেছে কি ঘটেনি, তা বোঝার জন্য মাইক্রোকন্ট্রোলারটি(Microcontroller) অনবরত টাচ সেন্সর(Touch Sensor)-টিকে পোল(Poll) করতে থাকে। এবং যদি হয়ে থাকে, এসএমএস(SMS) এর মাধ্যমে একটি উপযুক্ত বার্তা সংশ্লিষ্ট ব্যক্তিকে পাঠানো হবে, অ্যালার্ম সিস্টেম(Alarm System) টিও সক্রিয় হয়ে উঠবে। আমরা একটি স্পিকার(Speaker) ব্যবহার করে অ্যালার্ম সিস্টেম(Alarm System)-টিকে অনুকরণ করেছি, আপৎকালীন ঘন্টার মত একটি শব্দ বের হবে এবং আমরা একটা এসএমএস(SMS) ও পাঠিয়ে দিয়েছি।

(Refer Slide Time: 07:37)



এইটা হচ্ছে বেখ চিত্রটি। তো, এইটা হচ্ছে গ্রাউন্ড(Ground), এইটা হচ্ছে আরএক্স(RX) , এইটা হচ্ছে টিএক্স(TX) যেগুলি কিনা আরডুইনো(Arduino)- র 10 ও 11 নম্বর পিনের সাথে যুক্ত এবং আমরা সিগন্যালটিকে A0 পিন, VCC, এবং গ্রাউন্ড(Ground)- র সাথে একসাথে সংযুক্ত করেছি। এইটা হচ্ছে টাচ সেন্সর (Touch Sensor) এবং আমরা একটি অ্যালার্ম সিস্টেম(Alarm System)-ও এর সাথে যোগ করেছি; অ্যালার্ম সিস্টেম(Alarm System).-টিকে অনুকরণ করতে স্পিকার(Speaker) ব্যবহার করেছি, স্পিকার(Speaker)-টি D2 পিনের সাথে যুক্ত এবং এটি গ্রাউন্ড-এর সাথে যুক্ত।

(Refer Slide Time: 08:15)

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11);
int analogPin = A0;
int touchValue = 0;

void setup()
{
    mySerial.begin (9600);
    Serial.begin (9600);
    delay (100);
    pinMode (2, OUTPUT);
    digitalWrite (2, HIGH);
}

void loop()
{
    touchValue = analogRead(analogPin);
    Serial.print (touchValue);
    Serial.print ("\n");
    if (touchValue > 1000)
    {
        SendMessage();
        digitalWrite (2, LOW);
        delay (2000);
        digitalWrite (2, HIGH);
    }
}
```

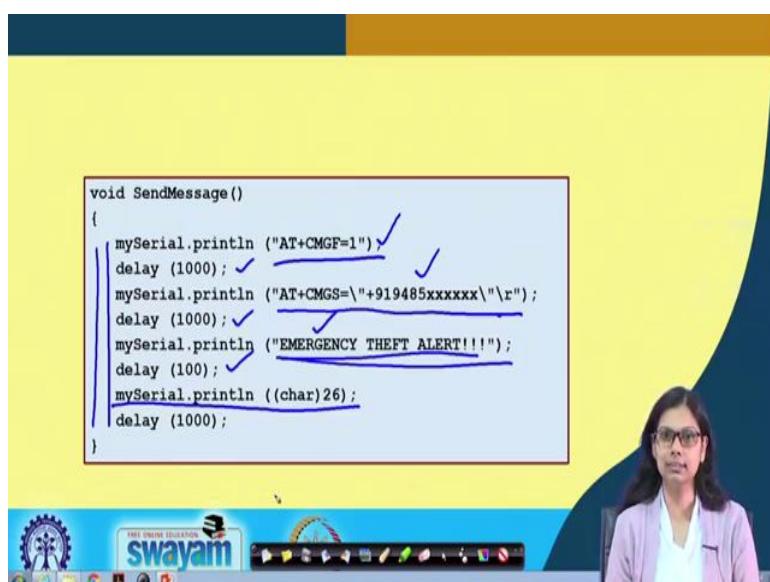
এখন এইটা হচ্ছে আরডুইনো উনো(Arduino Uno)- র কোড, আমরা ইতিমধ্যেই জানি কিভাবে সফটওয়্যার সিরিয়াল(SoftwareSerial) যোগাযোগ বানাতে হয়। আমরা 10 ও 11 নম্বর পিনের সাথে সিরিয়াল(Serial) যোগাযোগ স্থাপন করব এবং A0-টিকে এনালগ পোর্ট(Analog Port) হিসেবে ব্যবহার করব। শুরুর দিকে, স্পর্শের মান আমরা 0 রাখছি। প্রস্তুতিপর্বে আমরা জিএসএম(GSM) এর সাথে এই নির্দিষ্ট সিরিয়াল(Serial) যোগাযোগের জন্য mySerial.begin ব্যবহার করছি, এবং এইটা স্ক্রিনের(Screen) সাথে সিরিয়াল(Serial) যোগাযোগের জন্য। এরপর একটা বিরতি আছে, এবং আমরা pinMode- এর সাহায্যে দুই নম্বর পিনটিকে আউটপুট পোর্ট(OUTPUT Port) হিসেবে নিযুক্ত করছি এবং পোর্ট(port) দুইতে আমরা ডিজিটাল (Digital)মান High রাখছি।

এবার লুপ(Loop) পর্বে আমাদের কি করতে হবে? যে এনালগ পিনটির মাধ্যমে ওই সিগন্যালটি(Signal) টাচ সেন্সর(touch Sensor)- এর সাথে যুক্ত আছে, তার মানটি পড়বো, এবং

তারপর আমরা স্পর্শের যে মান পাঞ্চি সেটাকে দেখাবো, তোমরা সেটা সিরিয়াল মনিটরে(Serial Monitor) দেখতে পাবে। এবার যদি স্পর্শের মান অন্য আরেকটি কোন মানের অধিক হয়, যেটা কিনা এই ক্ষেত্রে 1000,, তাহলে আমরা বার্তাটি পাঠাবো এবং তারপর আমরা পিন 2 এর মান নিম্নমুখী করে দেব এবং আমরা একটা 2000 মিলিসেকেন্ডের(millisecond) অথবা দু সেকেন্ডের(Second) বিরতি দেব এবং আবারও পিন দুইয়ের মান নিম্নমুখী করব।

এসএমএস(SMS) ফাংশন(Function)-টি কি করে সেটা আমরা পরবর্তী অংশে দেখব।

(Refer Slide Time: 10:29)



পূর্ববর্তী উদাহরণে কি করলাম আমরা, না আমরা এসএমএস(SMS) টা পেলাম। আমরা এসএমএস(SMS)-টা পেয়েছিলাম, তারপর তার থেকে কিছু অংশ বাদ দিয়েছিলাম এবং তারপর যা প্রয়োজন ছিল তা করেছিলাম, এখানে আমাদের এসএমএস(SMS)-টি নির্দিষ্ট নাম্বার(number) পাঠাতে হবে। সেক্ষেত্রে তোমাদেরকে এই নির্দিষ্ট এটি কমান্ড(AT command)-টি ব্যবহার করতে হবে যেটি এখানে প্রয়োজন “AT+CMGF=1”।

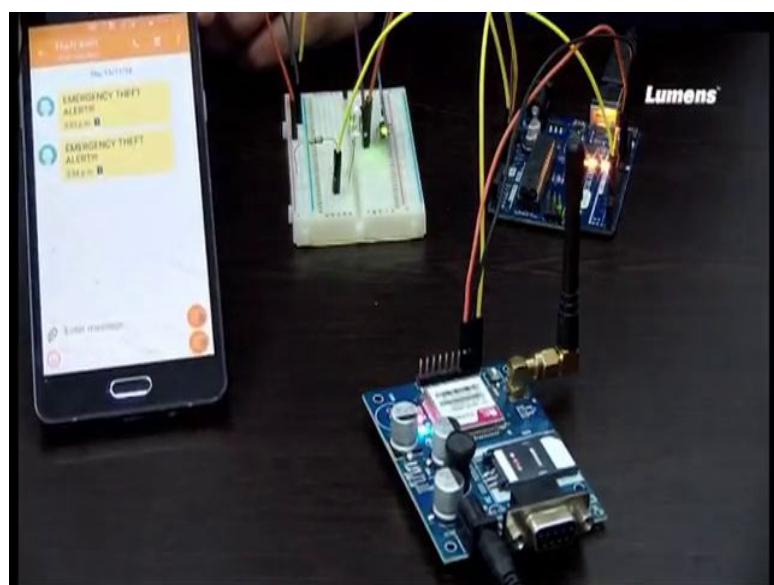
তারপর আমরা এক সেকেন্ডের(Second) একটা বিরতি দেব এবং তারপর আবার একটি এটি কমান্ড(AT Command) পাঠাচ্ছি যার মধ্যে আমরা সেই নির্দিষ্ট মোবাইল নাম্বার(Mobile

Number)- টা দিয়ে দিচ্ছি যেটিতে এসএমএস (SMS)পাঠাতে হবে । আমরা আবার 100 মিলি সেকেন্ড(millisecond) একটা বিরতি নিচ্ছি এবং এসএমএস(SMS)- টাকে পাঠাচ্ছি । এসএমএস(SMS) সতর্কবার্তা তোমার মোবাইল(mobile)- এ পৌঁছে যাবে এবং আবারো একটা বিরতি নিচ্ছি আমরা। যতগুলো অক্ষর আছে গুনলে দেখবে 26 আসছে।

আজকের পরীক্ষায় আমরা একটা টাচ সেন্সরের(Touch Sensor) সাথে যোগাযোগ স্থাপন করব, আমি ইতিমধ্যেই এই টাচ সেন্সরের(Touch Sensor) বৈশিষ্ট্য আলোচনা করেছি।

দেখা যাক ,এই টাচ সেন্সরের(Touch Sensor) সাথে কিভাবে যোগাযোগ স্থাপন করা যায়।

(Refer Slide Time: 14:41)



এইটা হচ্ছে টাচ সেন্সর(Sensor)- টি । অন্যান্য সেন্সর(Sensor)-দের মতই, এরও তিনটি পিন আছে, একটি হচ্ছে GND, পরেরটি হচ্ছে VCC, এবং তারপর এটি হচ্ছে আউটপুট সিগন্যাল(ou~~t~~put Signal) তথা এনালগ সিগনাল(Analog Signal) ।

টাচ সেন্সর(touch Sensor)- টি দু পাশ দিয়েই কাজ করে, তুমি যদি এই পাশ থেকে স্পর্শ করো কিংবা যদি এই পাশ থেকে স্পর্শ করো। একটা আমি Vcc- র সাথে যুক্ত করবো, একটা GND- র সাথে, এবং আরেকটি এনালগ পোর্ট(Analog Port)- এর সাথে যেটা কিনা AO।

এখন, যেরকম আমি বলেছিলাম আমি একটা এলাইডি(LED)- ও যোগ করেছি। এইটা হচ্ছে অ্যানোড(anode) এবং এখান থেকে আমি ডিজিটাল(digital) দু'নম্বর পিন যোগ করব। এটা শুধুমাত্র যাচাইয়ের জন্য যে বার্তাটি পাঠানো হয়েছে কি হয়নি।

জিএসএম(GSM) মডিউল(Module)- টির ক্ষেত্রে, এটি GND- র সাথে যুক্ত এবং অন্য দুটি পিন যার মধ্যে প্রথমটি হচ্ছে টিএক্স(TX) যেটি 10 নম্বর পিন এর সাথে যুক্ত এবং অপরটি হচ্ছে আরএক্স(RX) যেটি 11 নম্বর পিন এর সাথে যুক্ত। তোমরা মনে করে দেখতে পারো যে এসএমএস(SMS) দ্বারা বাতিটা জালানো নেভানোর ক্ষেত্রে, আমরা আমাদের মোবাইল(mobile) থেকে এই জিএসএম(GSM)-এ এসএমএস(SMS) পাঠাচ্ছিলাম এবং তারপর যা পাচ্ছিলাম তার উপর ভিত্তি করে এই মাইক্রোকন্ট্রোলার(microcontroller)- টি কাজ করছিল

কিন্তু এখন এটা সম্পূর্ণ বিপরীত, আমি যখন এটাকে স্পর্শ করব, এই জিএসএম(GSM) থেকে এসএমএস(SMS) সতর্কবার্তা আমার মোবাইল(mobile)- এ চলে যাবে। আমি প্রথমে কোড(Code)-টাকে ডরে দেবো এবং এই হচ্ছে সেই মোবাইল(mobile), যেখানে আমি এসএমএস(SMS) পাব টাচ সেন্সর(Touch Sensor) -টিকে স্পর্শ করলো।

কোড(Code)-টা ডরা হয়ে গেছে। এবার দেখা যাক... আমি এই সেন্সর(Sensor) স্পর্শ করলাম। সবুজ বাতিটি জ্বলছে তার মানে এই নির্দিষ্ট জিএসএম(GSM) মডিউল(Module) থেকে এই

সিম(SIM)-টির মাধ্যমে আমার মোবাইল(Mobile) এসএমএস(SMS) পাঠানো হয়ে গেছে এবং দেখা যাক আমি কি পাই; আমি এসএমএস(SMS) পেয়েছি। এরমধ্যে যে জিএসএম(GSM) মোবাইল নাম্বার(Mobile Number)- টি দেখা যাচ্ছে সেটা আমি সঞ্চয় করে রেখে দিচ্ছি। আমি আরো একবার করছি। তো, এলইডি(LED) টি জ্বলছে এবং আমি একটা এসএমএস(SMS) পাবো। তোমরা দেখতে পাচ্ছ যে আমি আর একটা এসএমএস(SMS) পেয়েছি।

এই সমগ্র পদ্ধতিটা খুবই সোজা কিন্তু আসল কথা হচ্ছে যে এই টাচ সেন্সর(Touch Sensor)- টির বিচিত্র রকমের ব্যবহার আছে।

ধন্যবাদ।

GLOSSARY

English	Transliteration	Translation
Touch Interface	টাচ ইন্টারফেস	একটি কম্পিউটার নির্দেশক প্রযুক্তি যা স্পর্শের অনুভূতির উপর নির্ভর করে।
Sensor	সেন্সর	এক প্রকার ডিভাইস যা কোন ভৌত স্থিতিমাপ কে অনুভব করতে পারে এবং তা মাপতে পারে।
Mobile Phone	মোবাইল ফোন	চলভাষ
Detector	ডিটেক্টর	সনাক্তকরী যন্ত্র
Capacitive	ক্যাপাসিটিভ	আধান ধারণ করতে পারে এমন বৈশিষ্ট্যযুক্ত বস্তু
Lecture Touchpad	লেকচার টাচপেড	আলোচনা

	টাচপ্যাড	একটি তল ঘাতে স্পর্শ করলে কোন কিছু ঘটে
Poll	পোল	যে পদ্ধতি দ্বারা একটি প্রোগ্রাম অথবা ডিভাইস অনবরত অপর আরেকটি ডিভাইসের অবস্থা জানার চেষ্টা করে যায়।
Microcontroller	মাইক্রোকন্ট্রোলার	একটি ইন্টিগ্রেটেড সার্কিট যা একটি ছোট কম্পিউটার হিসেবে কাজ করে।
Speaker	স্পিকার	একটি ঘন্টা যা তড়িৎ শক্তিকে শব্দ তরঙ্গে রূপান্তর করে।
Function	ফাংশন	যে কোনো প্রোগ্রামের একটি অংশ যা কোন নির্দিষ্ট কাজ করে
Digital	ডিজিটাল	সংখ্যা বা অংক সংক্রান্ত
Module	মডিউল	ঘন্টাংশ
Signal	সিগন্যাল	বৈদ্যুতিক তরঙ্গসংকেত
Output	আউটপুট	ফল

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Lecture – 38
Accelerometer

অষ্টম সপ্তাহে সকলকে স্বাগত। আমরা অবশ্যে এমবেডেড সিস্টেম ডিজাইন উইথ এআরএম(Embedded System Design with ARM) কোর্স(Course)-টির অন্তিম সপ্তাহে চলে এসেছি। এই সপ্তাহে আমরা, অ্যাক্সেলোরোমিটার(Accelerometer) এবং তার কার্যকারিতা নিয়ে আলোচনা করব। তো প্রথম আলোচনাটিতে, আমরা দেখব অ্যাক্সেলোরোমিটার(Accelerometer) কি জিনিস এবং এর চালন পদ্ধতিটি কি রকম।

তারপরের দুটি আলোচনায়, আমি প্রথমে বলব যে কিভাবে একটি অ্যাক্সেলোরোমিটার(Accelerometer) কে এসটিএম(STM) বোর্ড(Board)- এর সাথে যুক্ত করা যায়। এবং কিভাবে আমরা ডেটা(Data) গ্রহণ করব, প্রথমে আমি তা দেখাবো। এবং তারপর আমরা কিছু পরীক্ষা-নিরীক্ষার মাধ্যমে ডিভাইস(Device)-টির অবস্থান বের করব, অর্থাৎ ডিভাইসটি চিৎ হয়ে আছে নাকি সোজা দাঢ়িয়ে আছে নাকি উল্টো ভাবে দাঁড়িয়ে আছে ইত্যাদি। তো, এইরকমই কিছু জিনিস আমরা এই সপ্তাহে দেখব অ্যাক্সেলোরোমিটার(Accelerometer) কে কেন্দ্র করে। এবং আরো একটা জিনিস যেটা আমরা দেখব এই সপ্তাহে সেটা হল গ্যাস সেন্সর(Gas Sensor) -কে নিয়ে পরীক্ষা-নিরীক্ষা।

(Refer Slide Time: 01:50)

CONCEPTS COVERED

- ❑ Accelerometer: principle of operation
- ❑ ADXL 335 accelerometer

swayam

অ্যাক্সেলেরোমিটার(Accelerometer)- এর ব্যাপারে কথা বলা যাক। এই নির্দিষ্ট আলোচনায়,
আমি অ্যাক্সিলারোমিটার(Accelerometer)- এর চালনপদ্ধতি নিয়ে কথা বলব। এবং যে
অ্যাক্সেলেরোমিটার(Accelerometer)- টি নিয়ে কথা বলব সেটি হল **ADXL 335**।

(Refer Slide Time: 02:11)

What is an accelerometer?

- A dynamic sensor that can measure acceleration in one, two, or three orthogonal axes.
- Typically used in one of three modes:
 - As an inertial measurement of velocity and position.
 - As a sensor of inclination, tilt or orientation in 2/3 dimensions.
 - As a vibration or impact (shock) sensor.
- Most accelerometers are based on Micro Electro-Mechanical Sensors (MEMS).
 - Based on the displacement of a small mass etched into the silicon surface of the IC, and suspended by small beams.
 - As an acceleration is applied, a force develops ($F = ma$) that displaces the mass.

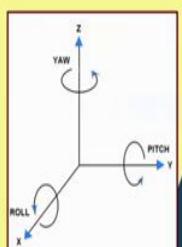
swayam

তো, অ্যাক্সেলেরোমিটার(Accelerometer) কি? এটি একটি গতীয় সেন্সর(Sensor) যেটি কিনা একটি, দুটি, অথবা তিনটি লম্ব অক্ষ যথা X,Y,Z বরাবর ঘূরণ মাপতে পারে। এবং সাধারণত এটির তিনটি ক্রিয়াপদ্ধতির মধ্যে যেকোনো একটি ব্যবহৃত হয়। এটি গতি এবং অবস্থানের জাড়া মাপক হিসেবে ব্যবহৃত হয়, এটি দ্রিমাত্রিক এবং ত্রিমাত্রিক ইনক্লিনেশন(Inclination), টিল্ট(Tilt), কিংবা অবস্থান সেন্সর(Sensor) হিসেবে ব্যবহৃত হতে পারে, অথবা অভিঘাত সেন্সর(Sensor) অথবা কম্পন সেন্সর(Sensor) হিসেবেও ব্যবহৃত হতে পারে।

বেশিরভাগ অ্যাক্সেলেরোমিটার(Accelerometer) যেগুলি আছে সেগুলি তৈরি হচ্ছে মাইক্রো ইলেক্ট্রো মেকানিকাল সেন্সর (Micro Electro Mechanical Sensors (MEMS))-এর ওপর ভিত্তি করে। এবং এটি নির্ভর করে একটি ছোট ভরের অবস্থানগত পরিবর্তনের উপর, যেটি সিলিকন উপরিস্তরের উপর খোদাই করা আছে, এবং ছোট কড়িকাঠ দ্বারা ঝোলানো আছে। আমরা এটা আরো বিশদে দেখব যখন এর চালনপদ্ধতি আলোচনা করব। তো, যেই অ্যাক্সেলেরেশন(Acceleration) এর প্রয়োগ হবে, একটি বলের সৃষ্টি হবে। আমরা জানি যে $P=m \times F$;

(Refer Slide Time: 04:01)

- The displacement of the mass can be measured using capacitive sensing or piezoelectric effect sensing.
- Change in capacitance, or generation of a voltage.
- The basic structure of an accelerometer consists of fixed plates and moving plates (called **mass**).
- Acceleration deflects the moving mass and unbalances the differential capacitor that results in a sensor output voltage amplitude which is proportional to the acceleration.
- By measuring the acceleration along the x, y and z directions, it is possible to calculate the inclination or tilt.
- It is also possible to calculate the angles of rotation along x, y and z axes (called *roll*, *pitch* and *yaw* respectively).





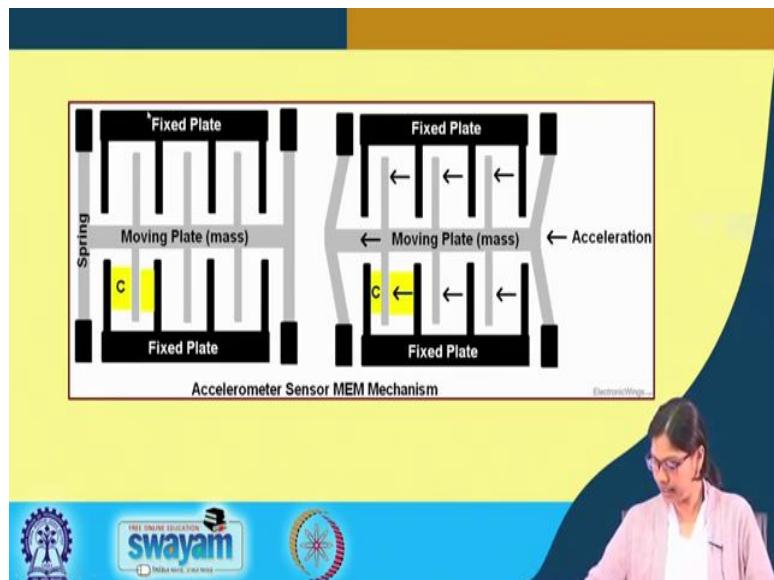

SWAYAM
 FREE ONLINE EDUCATION
 SWAYAM
 TANTRIK VIDYA, KALI VIDYA

ডরটির অবস্থানগত পরিবর্তন, ক্যাপাসিটিভ সেন্সিং (capacitive sensing) অথবা পিয়েজোইলেক্ট্রিক এফেক্ট সেন্সিং(piezoelectric effect sensing) দ্বারা মাপা যেতে পারে। যখন ডরটি এক জায়গা থেকে আরেক জায়গায় স্থানান্তরিত হয়, তো এটির অবস্থানগত পরিবর্তন, ক্যাপাসিটিভ সেন্সিং (capacitive sensing) অথবা পিয়েজোইলেক্ট্রিক এফেক্ট সেন্সিং(piezoelectric effect sensing) দ্বারা মাপা যেতে পারে। আসলে যা ঘটে তা হল ক্যাপাসিট্যান্স(capacitance) পরিবর্তন হয় অথবা ভোল্টেজের(Voltage) সৃষ্টি হয় যার দ্বারা এটিকে মাপা হয়। অ্যাক্সেলেরোমিটার(accelerometer)- এর মূল কাঠামোটিতে একটি স্থির পাত্র এবং একটি চলমান পাত্র আছে। এবং ভুরণ চলমান ডরটিকে বিচ্যুত করে দেয় এবং ডিফারেন্সিয়াল ক্যাপাসিটর(differential capacitor)- টির ভারসাম্যে ব্যাঘাত ঘটায় যার ফলে সেন্সর(Sensor)- এর আউটপুট(output)- এ ভোল্টেজের(Voltage) সৃষ্টি হয় যা কিনা ভুরণের সাথে আনুপাতিক।

এর মানে আসলে দাঁড়ায় যে একটি চলমান ডর আছে। যখনই ভুরণের সৃষ্টি হয় সেটি এই চলমান ডরটিকে বিচ্যুত করে দেয় এবং ডিফারেন্সিয়াল ক্যাপাসিটর(differential capacitor)-টির ভারসাম্যে তারতম্য ঘটায়। তো, ডর- এও এক প্রকার ভারসাম্যহীনতার সৃষ্টি হচ্ছে। x , y , ও z অক্ষ বরাবর ভুরণ মাপলে, ইনক্লিনেশন(Inclination) অথবা টিল্ট(Tilt) হিসেব করা সম্ভব।

এটি কতটা কাত হয়ে আছে, তা অ্যাক্সেলেরোমিটার(Accelerometer)-এর সাহায্যে নির্ধারিতভাবে নির্ণয় করা যাবে। এইটা করা যেতে পারে x , y , ও z অক্ষ বরাবর ভুরণ মাপলে। x , y ও z অক্ষ গুলিকে কেন্দ্র করে ঘূর্ণির কোন- ও হিসাব করা যেতে পারে যেটাকে x অক্ষ বরাবর হলে বলে “রোল”(Roll), y অক্ষ বরাবর হলে বলে “পিচ”(Pitch) এবং z অক্ষ বরাবর হলে বলে “ইয়ে”(Yaw)।

(Refer Slide Time: 06:21)



এখানে, এই স্থির পাত্রগুলি আছে এবং এখানে আছে চলমান ভরটি। এবং তুমি দেখছো যে যখনই ভ্রমণের সৃষ্টি হচ্ছে, এই যে হলুদ জিনিসটি এখানে আছে এটি পরিবর্তন হচ্ছে। এর উপর নির্ভর করে আমরা নিখুঁতভাবে বের করতে পারবো এটা কোন অক্ষ বরাবর বেশি কাত হয়ে আছে। এই হচ্ছে MEMS পদ্ধতি বিশিষ্ট অ্যাক্সেলেরোমিটার(Accelerator) সেন্সর (Sensor)।

(Refer Slide Time: 07:16)

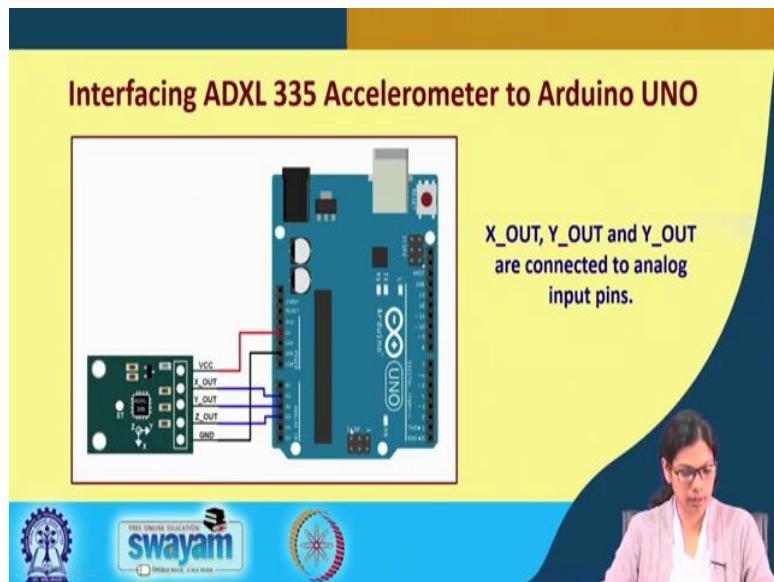


যে অ্যাক্সেলেরোমিটার(Accelerometer)- টা আমরা ব্যবহার করব সেটি হল ADXL-335 , তোমরা দেখতে পাচ্ছ এইগুলি হচ্ছে অ্যাক্সেলেরোমিটার(Accelerometer)- এর পিন। এইটা তোমার Vcc, এইটা হচ্ছে GND, এবং এই হচ্ছে X,Y ও Z স্থানান্তরণগুলি। তো, এই হচ্ছে অন্যদিকটি থেকে এবং এটি হচ্ছে অন্য আরেকটি থেকে।

ADXL-335 হচ্ছে একটি ছোট, পাতলা, কম শক্তির তিনটে পিন বিশিষ্ট অ্যাক্সেলেরোমিটার(Accelerometer) যার আছে সিগন্যাল কন্ডিশন ভোল্টেজ আউটপুট(Signal Condition Voltage output)। এটা আসলে কি করতে পারে, এটি টিল্ট(Tilt) সেন্সিং(Sensing) সরঞ্জামে মাধ্যাকর্ষণ- এর সাহায্যে স্থির ভূরণ মাপতে পারে, এবং চলন, সংঘাত অথবা কম্পন দ্বারা সৃষ্টি গতিশীল ভূরণও মাপতে পারে। এটি x,y, এবং z অক্ষ বরাবর ভূরণ $+- 3g$ -এর মধ্যে পরিমাপ করতে পারে।

আমরা যখন এটি এসটিএম(STM)-এর সাথে যুক্ত করবো, আমরা দেখব যে এই x_OUT,y_OUT,z_OUT, কে নিয়েই বেশি সচেতন। তো আমরা এই সিগনাল(Signal)- গুলিকে এনালগ পোর্টের(Analog Port) সাথে যুক্ত করবো। এইগুলির আউটপুট সিগন্যাল(Output)-গুলি আর কিছুই নয় বস্তুত এনালগ সিগনাল(Analog Signal) যা ভূরণের সাথে আনুপাতিক।

(Refer Slide Time: 09:40)



এখন, সংযোগটা খুবই সহজ ব্যাপার। আমরা এটি আরডুইনো(Arduino) ব্যবহার করে দেখিয়েছি।
আমরা পরীক্ষাটি করেছি এসটিএম(STM) ব্যবহার করে। এইখানেই এটি Vcc-র সাথে যুক্ত, এইটি GND-র সাথে যুক্ত। এবং এই x,y ও z ঘন্থাক্রমে A1,A2,A3 পিনগুলির সাথে যুক্ত।

তো, এই হচ্ছে অ্যাক্সেলেরোমিটার(Accelerometer)- এর ব্যাপারে ঘাবতীয় তথ্য। এরপর আমরা দেখব কিভাবে অ্যাক্সেলেরোমিটার(Accelerometer)- টি এসটিএম(STM) বোর্ডের(Board) সাথে যুক্ত করা যায়। আমরা ইতিমধ্যেই তোমাদেরকে সংযোগটা দেখিয়েছি। পরবর্তী আলোচনায় আমরা সেটা বিশদে দেখব।

ধন্যবাদ।

GLOSSARY

English	Transliteration	Translation
capacitive sensing	ক্যাপাসিটিভ সেন্সিং	যে পদ্ধতিতে ক্যাপাসিটরের বৈশিষ্ট্য কে কাজে লাগিয়ে যে কোন বস্তুর উপস্থিতি বা

		অনুপস্থিতি আন্দাজ করা যেতে পারে।
piezoelectric effect	পিয়েজোইলেক্ট্রিক এফেক্ট	যে বৈশিষ্ট্যের দ্বারা যান্ত্রিক চাপ প্রয়োগ করলে কিছু কিছু বস্তুর ক্ষেত্রে আধানের সৃষ্টি হয়।
capacitance	ক্যাপাসিটেন্স	যে বৈশিষ্ট্যের দ্বারা একটি বস্তু আধান ধরে রাখতে সক্ষম হয়।
inclination	ইনক্লিনেশন	বুঁকে পড়া
tilt	টিল্ট	কাত হওয়া।
micro electro mechanical system	মাইক্রো ইলেক্ট্রো মেকানিকাল সিস্টেম	যে সেন্সর মাইক্রোইলেক্ট্রনিক ফেরিকেশন পদ্ধতি দ্বারা তৈরি
signal condition voltage output	সিগন্যাল কন্ডিশন ভোল্টেজ আউটপুট	সিগন্যালটিকে প্রযোজন মতো পরিবর্তন করার পর আউটপুট তড়িৎচালক বল রূপে পাওয়া যাবে এমন বৈশিষ্ট্য
Course	কোর্স	বিষয়
differential capacitor	ডিফারেন্সিয়াল ক্যাপাসিটর	যে ক্যাপাসিটর টির ক্যাপাসিট্যান্স তড়িৎচালক বলের উপর নির্ভর করে।

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Lecture – 39
Experiment Using Accelerometer

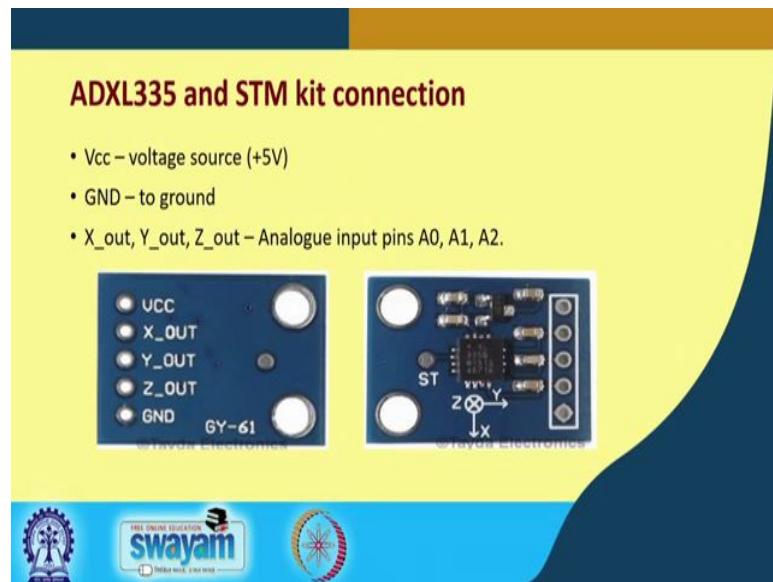
আজকের আলোচনায় সকলকে স্বাগত। এই পরীক্ষাটিতে, আমি তোমাদেরকে দেখাবো যে কিভাবে
অ্যাক্সেলেরোমিটার(Accelerometer)-টাকে এসটিএম বোর্ড(STM Board)- এর সাথে যুক্ত করা
যায় এবং কুলটার্ম(CoolTerm)- এ x,y এবং z এই তিনটি অক্ষের কি মান পাচ্ছি। আমরা
ইতিমধ্যেই জানি কিভাবে এসটিএম বোর্ড(STM Board) সাথে কুলটার্ম(CoolTerm) ব্যবহার
করতে হয়। আরডুইনো(Arduino)-র ক্ষেত্রে এটা সোজা, এটা সিরিয়াল মনিটর(Serial Monitor)-
এ দেখানো যাবে।

(Refer Slide Time: 00:59)



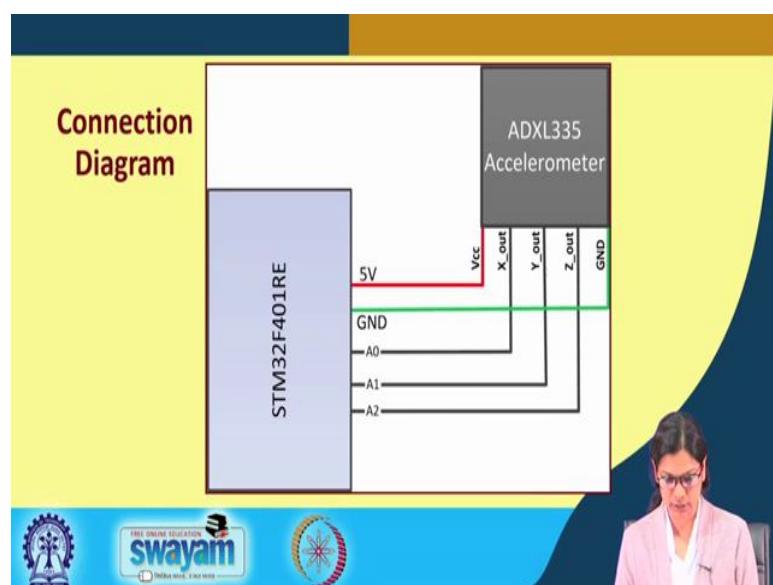
প্রথমে, আমি তোমাদেরকে অ্যাক্সেলেরোমিটার(Accelerometer)-এর উপর পরীক্ষাটি দেখাবো
এবং তারপর আমি প্রদর্শনের কাজটি করব।

(Refer Slide Time: 00:07)

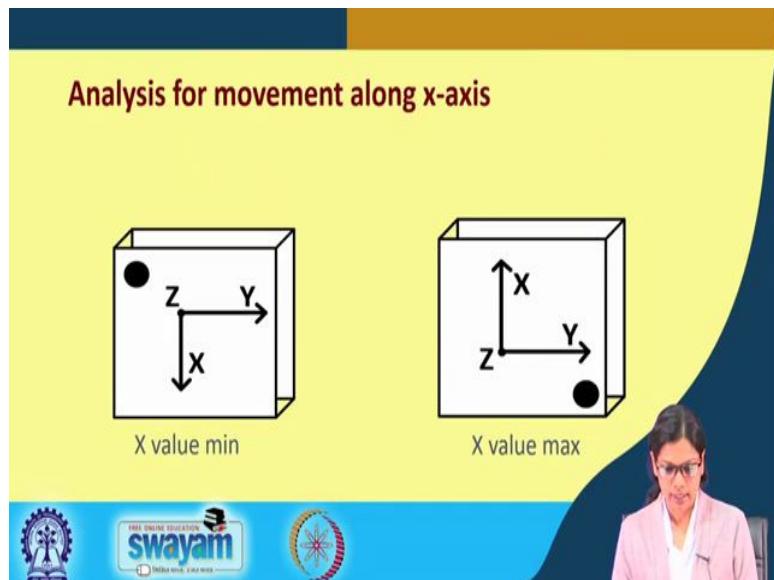


তো, এটি হচ্ছে ADXL-335 অ্যাক্সেলোরোমিটার(Accelerometer), এবং আমি এসটিএম কিট(STM Kit)- এর সাথে সংযোগটা করব। তো, ডোল্টেজ(voltage) উৎস Vcc +5 ডোল্টের (volt)সাথে যুক্ত হবে। এসটিএম বোর্ড(STM Board)- এর গ্রাউন্ড(Ground) পিনটির সাথে GND যুক্ত হবে। এবং X_OUT, Y_OUT এবং Z_OUT এনালগ(Analog) পিনটির আউটপুট গুলি A0, A1 and A2 এনালগ ইনপুট (Analog Input) পিনগুলির সাথে যুক্ত হবে।

(Refer Slide Time: 01:46)

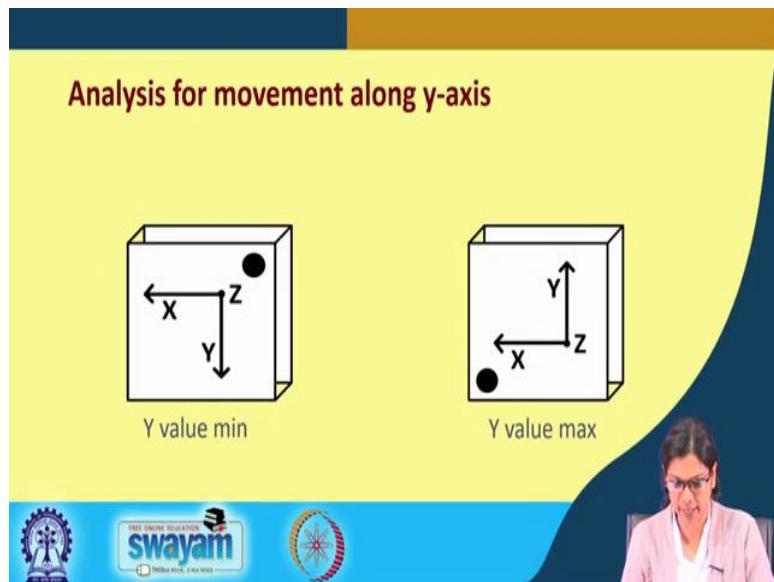


(Refer Slide Time: 02:09)



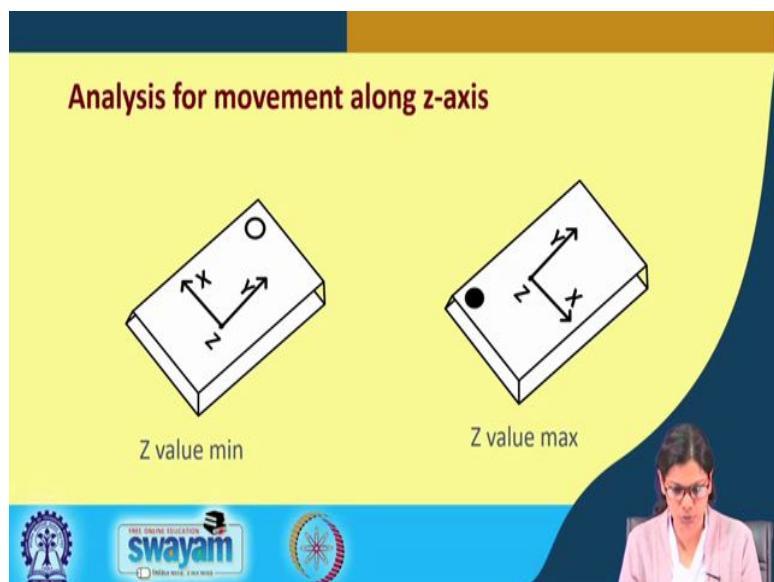
এবার x অক্ষ বরাবর এই গতিবিধি বিশ্লেষণ করা যাক। যখন তুমি অ্যাক্সেলেরোমিটার(Accelerometer)-টিকে ধরে থাকবে, তুমি দেখবে যে অ্যাক্সেলেরোমিটার(Accelerometer)-টির নির্দিষ্ট কিছু অক্ষ আছে, একটি হচ্ছে x বরাবর, অন্যটি হচ্ছে y বরাবর এবং এইটা হচ্ছে z বরাবর। যখন তুমি x অক্ষ বরাবর অ্যাক্সেলেরোমিটার(Accelerometer)-টাকে ধরবে, তখন তুমি এই x - এর জন্য সর্বোচ্চ মান পাবে। সেইরকমই তোমরা যদি এটাকে এইভাবে ধরো, x -এর মান সর্বনিম্ন হবে।

(Refer Slide Time: 02:54)



সেইরকমই Y-এর জন্যেও যদি দেখো, যদি তুমি এইভাবে ধরো, Y-এর মান সর্বোচ্চ হবে; এবং এইভাবে ধরলে Y-এর মান সর্বনিম্ন হবে। এর উপর ভিত্তি করে তুমি আসলে বুঝতে পারবে ডিভাইস(Device)টি তুমি কিভাবে ধরে আছো, ডিভাইস(Device) টি সোজা হয়ে আছে নাকি কাত হয়ে আছে নাকি উল্লম্বভাবে বাঁ দিকে কাত হয়ে আছে নাকি উল্লম্বভাবে ভানদিকে কাত হয়ে আছে। তো, ডিভাইস(Device)-টির অবস্থানগত দিকটি এই মান থেকে বের করা যেতে পারে।

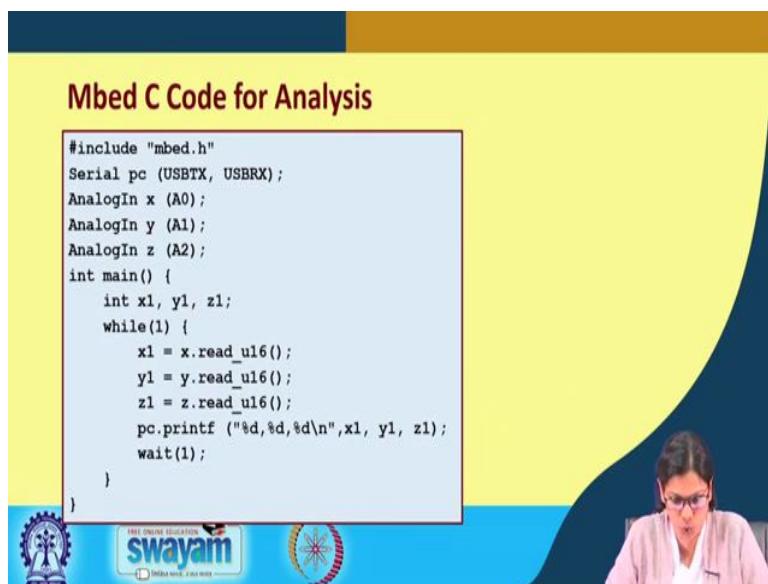
(Refer Slide Time: 03:27)



সেইরকমই, তুমি যদি অ্যাক্সেলেরোমিটার(Accelerometer)-টাকে এইদিকে ধরো, তখন এখানে Z-এর মান সর্বনিম্ন হবে এবং এখানে Z- এর মান সর্বোচ্চ হবে।

আমরা এটা এসটিএম কিট(STM Kit)- এর দ্বারা পরীক্ষা করে দেখবো।

(Refer Slide Time: 03:56)



তো চলো প্রথমে, mbed C কোড(Code) -টা বোঝা যাক। এটা বলছে যে mbed library অর্থাৎ mbed.h-টিকে আমাদেরকে উল্লেখ করতে হবে। প্রথমে, USBTX এবং USBRX - এর সাথে আমাদের সিরিয়াল(Serial) যোগাযোগ স্থাপন করতে হবে যে যোগাযোগটির নাম হবে "pc"। আমরা তিনটি এনালগ(Analog) ইনপুট(Input) সিগন্যাল(Signal) x, y, z ব্যাখ্যা করেছি এবং তাদের সঙ্গে যুক্ত করেছি এনালগ(Analog) পোর্ট(Port) নাম্বার A0, A1 ও A2 যথাক্রমে। X OUT, Y OUT and Z OUT যথাক্রমে A0, A1 ও A2 এসটিএম(STM) পিনগুলির সাথে যুক্ত হবে।

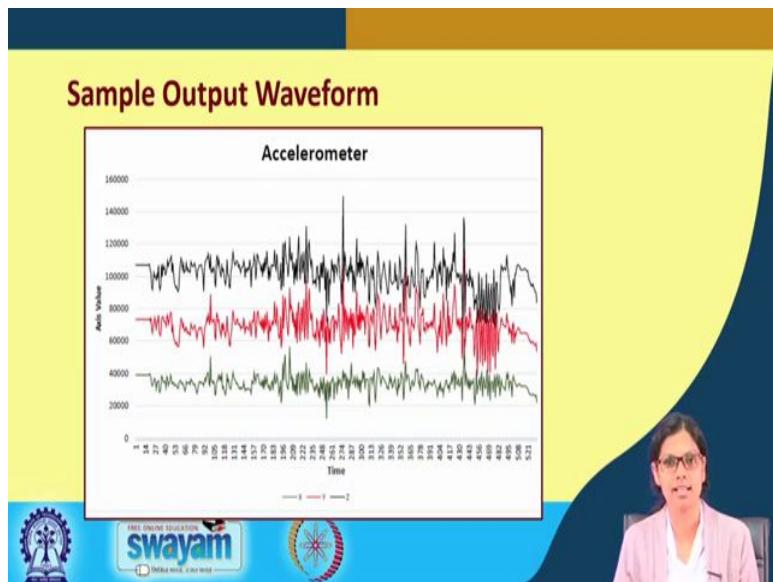
এরপর মেন(Main) ফাংশন(function) - এ আমরা এলাম। মেন(Main) ফাংশনে(function) আমরা তিনটি ভ্যারিয়েবল(variable) ব্যাখ্যা করলাম x1, y1 ও z1, যেখানে আমরা

`x.read_u16` ফাংশন(function)- টির সাহায্যে `x1`- এ এনালগ(Analog) মান সংরক্ষণ করছি, `y1`-এর ক্ষেত্রে এটি হচ্ছে `y.read_u16`, এবং `z1`- এর ক্ষেত্রে `z.read_u16` | এবং “pc” নামে যে সিরিয়াল(Serial) যোগাযোগ- টি আমরা বানিয়েছি, তার সাথে আমরা `x1`, `y1` ও `z1`- এ সংরক্ষিত `x`, `y` এবং `z` স্থানাঙ্কের মানগুলি দেখাবো ।

এবং এই সমস্ত কিছু `while` লুপে(Loop) চলছে। এটা দেখানোর পর এটি এক সেকেন্ডের(Second) জন্য অপেক্ষা করবে, এবং আবার একই পদ্ধতি পুনরায় শুরু করবে।

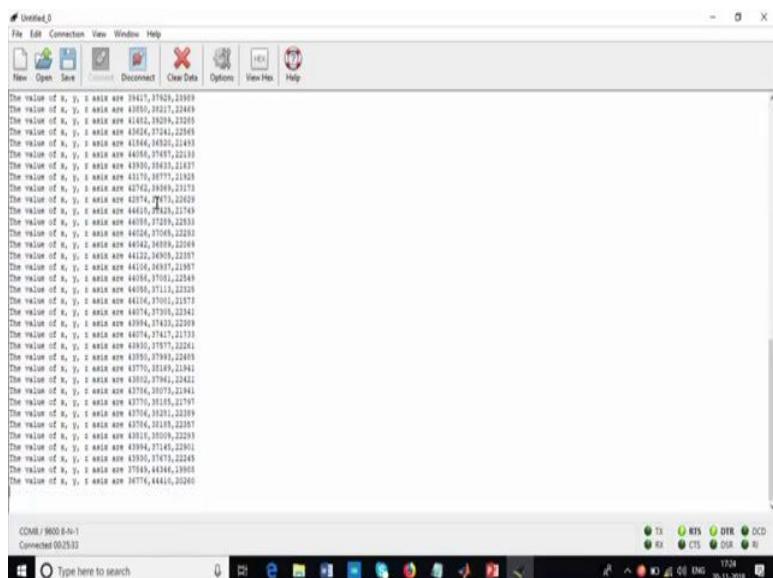
তুমি যদি দেখতে চাও যে কিভাবে `x`, `y`, `z` স্থানাঙ্কের মানগুলি পরিবর্তন হচ্ছে তুমি তাহলে কোন ম্যাটল্যাব(Matlab) কোড ব্যবহার করতে পারো সেটাকে দেখানোর জন্য, যেটা আমরা এখানে দেখাবো নাকিন্তু তোমাদের যদি আগ্রহ থাকে, তাহলে আমরা তোমাদেরকে ওই কোড(Code)-গুলি দিতে পারি, তোমরা তোমাদের দিক থেকে চেষ্টা করতে পারো, `x`, `y` ও `z` স্থানাঙ্কের মানগুলি গ্রাফ(Graph)- এ চিহ্নিত করতে । কিংবা, তোমরা আর একটা কাজ করতে পারো, তোমরা মানটাকে পড়ে নিলে, সেটাকে কোন অনলাইন(Online) তথ্যভাণ্ডারে সংযোগ করে রাখলে যেভাবে এর আগে এসএমএস(SMS)- এর মাধ্যমে করা আমরা দেখিয়েছি। এবং তথ্যভাণ্ডারে যে মান- ই সংরক্ষিত থাকুক না কেন, তুমি বন্স্টেড সেটিকে বিশ্লেষণ করতে পারো এমনকি তার উপর ভিত্তি করে গ্রাফ(Graph)- এ বিন্দুগুলি চিহ্নিত করতে পারো । তো, এটাকে করার বিভিন্ন রকম উপায় আছে।

(Refer Slide Time: 07:38)



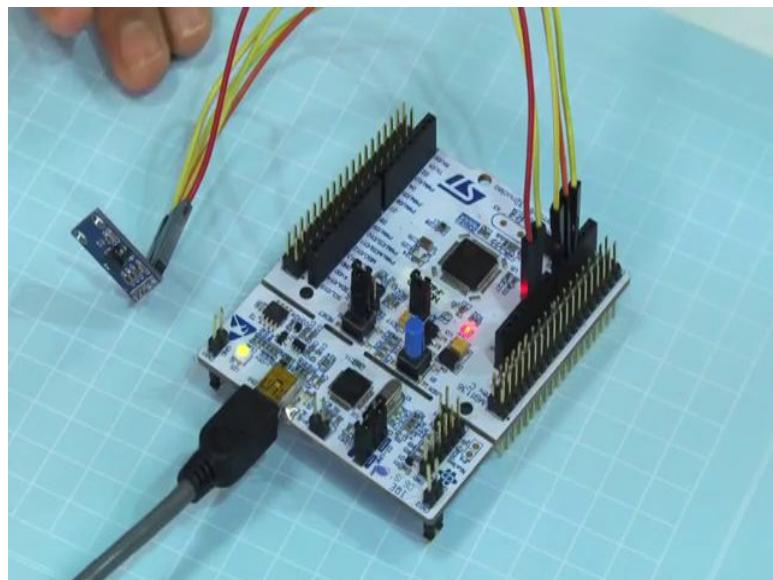
অবশেষে, গ্রাফ- এ চিহ্নিত বিন্দুসমূহটি কিছুটা এরকম দেখতে হবে। এটি হচ্ছে অ্যাক্সেলেরোমিটার(Accelerometer) থেকে পাওয়া নমুনাস্বরূপ আউটপুট (Output)তরঙ্গরূপ। এটি হচ্ছে x অক্ষ,y অক্ষ এবং z অক্ষের মানগুলি,এইভাবে এরা পরিবর্তন হচ্ছে।আমরা দেখতে পাচ্ছি এখানে এবং এখানে ছুঁচোলো ফলার আকার ধারণ করেছে।এবার আমি এটার প্রদর্শনী কুলটার্ম(CoolTerm)- এর সাহায্যে দেখাবো।

(Refer Slide Time: 08:27)



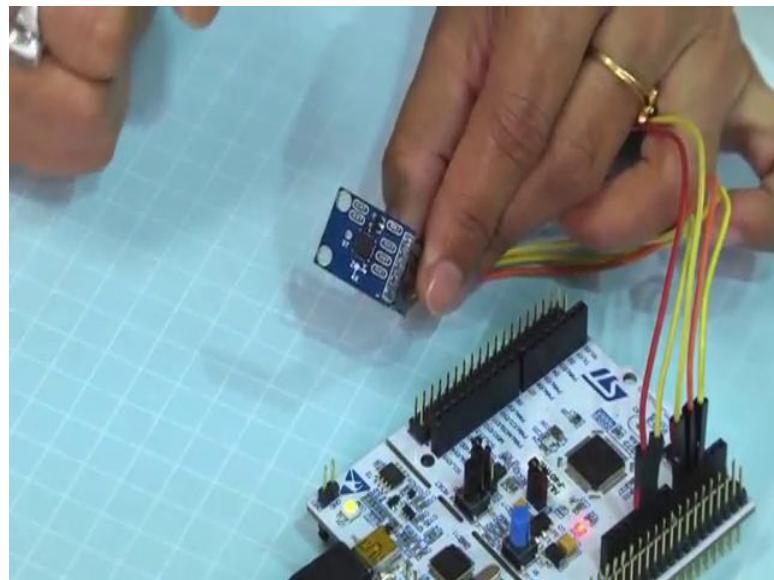
তোমাকে এই কুলটার্ম(CoolTerm)-টা ব্যবহার করতে হবে; তো, কিন্তু ডেটা(Data) আসছে। প্রথমে
আমরা সংযোগটা দেখব।

(Refer Slide Time: 08:42)



এইটা হচ্ছে Vcc, যেটি 5 ভোল্ট(Volt) এর সাথে যুক্ত। এইটা হচ্ছে x অক্ষ যেটা A0- র সাথে যুক্ত।
এইটা হচ্ছে y- অক্ষ যেটা A1- এর সাথে যুক্ত। এবং এইটা হচ্ছে z যেটা A2- এর সাথে যুক্ত। এবং
অবশ্যে, এইটি গ্রাউন্ড(Ground) পিনটির সাথে যুক্ত। তো, সংযোগটা খুবই সোজাসাপ্তা; এই যা
দরকার ছিল সংযোগের জন্য।

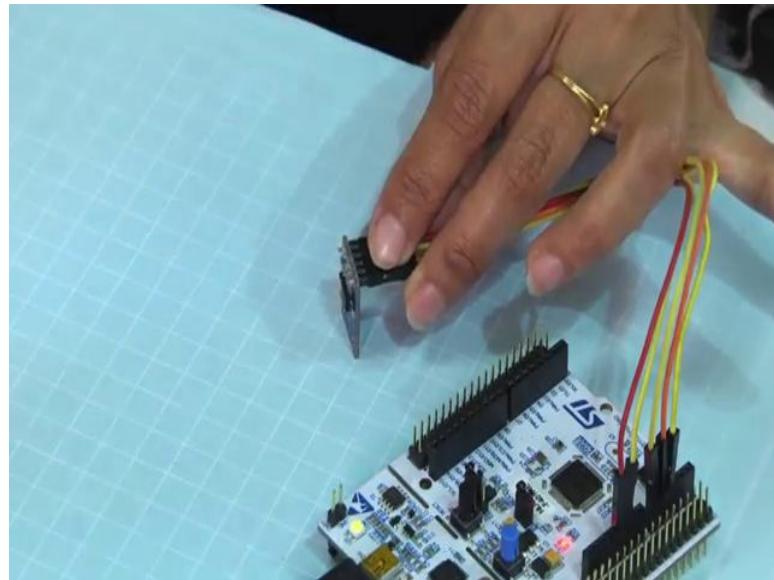
(Refer Slide Time: 09:40)



তোমরা যদি এই অ্যাক্সেলোরোমিটার(Accelerometer)-টিকে ঠিক করে দেখো, তোমরা দেখবে যে
এখানে x- অক্ষ আছে, এখানে হচ্ছে y- অক্ষ, এবং এইটা হচ্ছে z- অক্ষ। যখনই তোমরা
অ্যাক্সেলোরোমিটার(Accelerometer)- কে এমনভাবে ধরবে যে তীরচিহ্নটি x- অক্ষের দিকে মুখ
করে থাকবে, তখন x- অক্ষের মান সর্বোচ্চ হবে।

ধরা যাক, আমি অ্যাক্সেলোরোমিটার(Accelerometer)-টিকে এইভাবে ধরেছি এবং আমি আসবো
এবং হাইপার টার্মিনাল(Hyper Terminal)- এ x অক্ষের মান টি দেখব। এখন দেখো, x, y ও z
অক্ষের মান গুলি, তোমরা দেখতে পাচ্ছা যে x- অক্ষের মান হচ্ছে 44058, y অক্ষের মান হচ্ছে
36873 এবং অবশ্যই এটা পরিবর্তন হচ্ছে। যদিও আমি এটাকে খুব শক্তভাবে ধরে আছি, তা
সত্ত্বেও সামান্য কিছু পরিবর্তন লক্ষ্য করা যাচ্ছে।

(Refer Slide Time: 11:16)



এখন, আমি নিশ্চিত করবো যে y-অক্ষের মান সর্বোচ্চ হবে। আমি অ্যাক্সেলেরোমিটার(Accelerometer)-কে এমন ভাবে ধরেছি যাতে y-অক্ষটি উপর দিকে থাকে। তো, এখন তোমরা হাইপার টার্মিনাল(Hyper Terminal) দেখতে পাচ্ছ কি ধরনের মান পাচ্ছ। আমরা x অক্ষের মান পাচ্ছি 36728 | y-অক্ষের মান হচ্ছে 44426, এবং z অক্ষের মান হচ্ছে 18628।

এবার ধরা যাক, তোমাদের কাছে একটা বাক্স আছে যেখানে তোমরা জানতে চাও যে সব সময় এই জিনিসটির অবস্থানগত দিকটি যেরকম হবে..... ধরা যাক ওই নির্দিষ্ট জিনিসটির মাথাটি সবসময় উপরের দিকে থাকবে। তো, তোমরা এটা সব সময়ই নিশ্চিত করতে পারো যে x- অক্ষের মান সর্বোচ্চ হবে যাতে তোমরা অ্যাক্সেলেরোমিটার(Accelerometer)-টি ওই সরঞ্জাম টিতে এমনভাবে রাখ যে x- অক্ষটি এরকমভাবে থাকবে। তো, এটা যদি এরকম ভাবে থাকে, তাহলে x- অক্ষের মানটি আমরা যা পাবো সেটি সর্বোচ্চ হবে। এবং সেটা এটা যদি সর্বোচ্চ না হয়, তাহলে অপর দুটি মান সর্বোচ্চ হবে, তাহলে তোমরা বুঝবে যে তোমাদের ডিভাইস (Device)টি সঠিকভাবে

বসানো নেই। কোন ধরনের সতর্কবার্তার ব্যবস্থাও করা যেতে পারে যা ইঙ্গিত করবে যে ডিভাইস(Device)-টি ঠিক করে বসানো নেই।

এই পরীক্ষাটি থেকে, আমরা যেটা বুঝতে পেরেছি সেটা হচ্ছে কিভাবে অ্যাক্সেলেরোমিটার(Accelerometer) - টি এসটিএম(STM) কিট- এর সাথে যুক্ত হবে। এবং তুমি যদি ডিভাইস(Device)-টির অবস্থানগত দিকটি জানতে চাও, তুমি এটা ব্যবহার করে সেটা বের করতে পারবে। এবং তুমি যদি এই নির্দিষ্ট ডিভাইস(Device) -টি নির্দিষ্টভাবে বসাতে চাও, তুমি এটা ব্যবহার করে সেটা নিশ্চিত করতে পারো। বেশিরভাগ মোবাইল ডিভাইস(Device)- এ এই অ্যাক্সেলেরোমিটার(Accelerometer)-টি থাকে। তুমি যদি স্বাস্থ্য-নজরদারি সংক্রান্ত কোন অ্যাপ(App) ব্যবহার করে থাকো তাহলে সেটি দেখায়, যে দিনে তুমি কত পা হেঁটেছ।

এইগুলি এই ডিভাইস(Device)-টির সাহায্যে মাপা হয়। আমাদের মোবাইল ফোনের(Mobile Phone) মধ্যে একটা অ্যাক্সেলেরোমিটার(Accelerometer) অথবা গাইরোস্কোপ(gyroscope) আছে যেটা আমাদের সারাদিনের ভৱণের গতিবিধির তথ্য আমাদের দেয়। অবশ্যই, এই ডিভাইস(Device)-গুলিরও কিছু সমস্যা আছে, কারণ তুমি যদি শুধু ঝাঁকুনি খেয়ে থাকো, তাহলেও এটি ধরে নেবে যে তুমি হাঁটছো। তো, আমরা যখন অ্যাপ্লিকেশন(Application) বানাবো তখন এই ব্যাপারগুলি লক্ষ্য রাখতে হবে।

ধন্যবাদ।

GLOSSARY

English	Transliteration	Translation
CoolTerm	কুলটার্ম	এক ধরনের বিশেষ সফটওয়্যার যা সিরিয়াল পোর্ট এর সাথে যুক্ত হার্ডওয়ার- এর সাথে তথ্য আদান-প্রদান

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institutes of Technology, Meghalaya

Lecture – 40
Experiment Using Bluetooth

এই আলোচনায় সকলকে স্বাগত। আজকের আলোচনায় STM বোর্ড(Board)- এর সাথে একটি ব্লুটুথ মডিউল(Bluetooth Module)- কে যুক্ত করবো আমরা। এর সঙ্গে আমরা অ্যাক্সেলেরোমিটার(Accelerometer) নিয়েও কিছু পরীক্ষা-নিরীক্ষা করব।

আমরা দেখাবো যে কিভাবে আমরা একটি ডিভাইসের অবস্থানগত দিকটি বের করব, ডিভাইসটি কি চিং হয়ে আছে, নাকি উল্লম্বভাবে উপরের দিকে মুখ করে দাঁড়িয়ে আছে নাকি উল্লম্বভাবে নিচের দিকে মুখ করে দাঁড়িয়ে আছে নাকি সমান্তরালভাবে বামদিকে আছে নাকি সমান্তরালভাবে ডানদিকে আছে। এইরকমই কিছু অবস্থানগত দিক আমরা বের করার চেষ্টা করব।

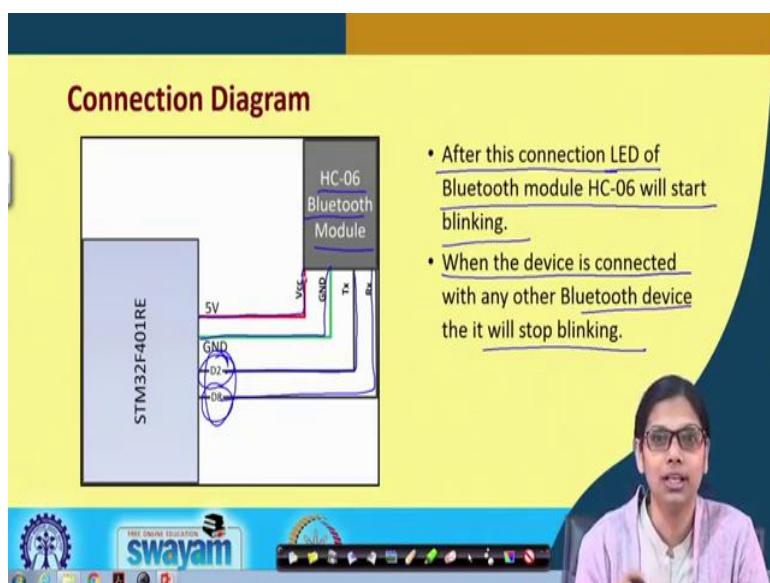
(Refer Slide Time: 01:31)



আমরা প্রথমে দেখাবো কিভাবে মাইক্রোকন্ট্রোলার বোর্ড(microcontroller board)- এর সাথে একটি ব্লুটুথ মডিউল(Bluetooth Module)-কে যোগ করব। এবং আমরা ইতিমধ্যেই জানি

অ্যাক্সেলোমিটাৰ(accelerometer)- টিকে কিভাবে যুক্ত কৰতে হয়, তাৰপৰ আমৰা যেকোনো নিৰ্দিষ্ট ডিভাইসেৰ(Device) অবস্থানগত দিকটি খুঁজে বেৱ কৰব এবং সেটা নিৰ্ণয় কৰব। এবং তা একটা মোবাইল ফোন(mobile phone)- এ দেখা যাবে, যেটিকে আমি STM- এৰ ব্লুটুথ মডিউল(Bluetooth Module) - এৰ মাধ্যমে যুক্ত কৰবো। এবং অবশ্যে, গোটা ব্যাপারটাৰ কাজ কৰাৰ পদ্ধতিটা তোমাদেৱকে আমৰা দেখাবো।

(Refer Slide Time: 02:16)



এখন, আমৰা যে ব্লুটুথ মডিউল(Bluetooth Module)-টি ব্যবহাৰ কৰছি এখানে সেটা হল HC-06। সংযোগ কৰা খুবই সহজ। এইটা হচ্ছে Vcc, যেটা 5 ভোল্ট এৰ সাথে যুক্ত। এইটা হচ্ছে GND, যেটা STM- এৰ গ্রাউন্ড পিনেৰ সাথে যুক্ত। টিএক্স(TX) D2- র সাথে যুক্ত, এবং আরএক্স(RX) যুক্ত D8- এৰ সাথে। সেইৱকমই, আমাদেৱকে এই দুটি পিনেৰ মধ্যে একটা সিৱিয়াল(serial) যোগাযোগ স্থাপন কৰতে হবে। একটি পিনকে বানাতে হবে আরএক্স(RX), অপৰটিকে টিএক্স(TX)।

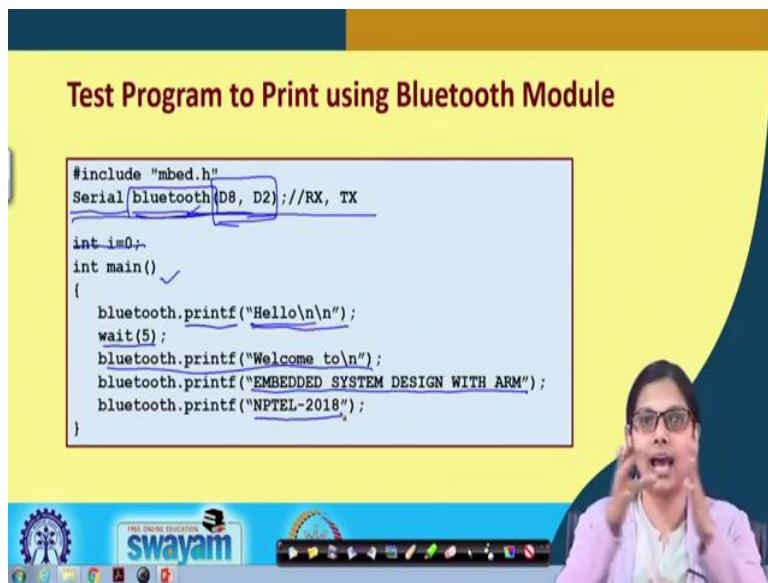
এখন, প্ৰদৰ্শনেৰ সময় আমি তোমাদেৱ দেখাবো যে ব্লুটুথ মডিউল(Bluetooth Module)-টা লাগালে ঠিক কিভাবে এটা কাজ কৰে। প্ৰথমত, ব্লুটুথ মডিউল(Bluetooth Module)-টা STM- এৰ সাথে জুড়তে হবে। একবাৰ যোগাযোগ স্থাপন হয়ে গৈলে, ব্লুটুথ মডিউল(Bluetooth

Module)-এর এলইডি(LED)- টা মিটমিট করে জ্বলতে শুরু করবে । এটি ইস্পিত করতে যে যোগাযোগ স্থাপন হয়ে গেছে ।

এরপর যখন ডিভাইস(Device)-টিকে যুক্ত করা হবে, তখন দুটি ডিভাইসের মধ্যে তথ্য আদান প্রদানের জন্য ব্লুটুথ মডিউল(Bluetooth Module) টি ব্যবহৃত হবে । যখন তোমরা,STM বোর্ড(board)- এর সাথে যুক্ত ব্লুটুথ মডিউল(Bluetooth Module) -টি তোমার মোবাইল ফোন-এর ব্লুটুথ ডিভাইস(Bluetooth Device)-টির সাথে কানেক্ট করবে, এই দুটি নিজেদের মধ্যে কথোপকথন চালাবে । তো, সেটা যখন তোমরা করবে, ব্লুটুথ ডিভাইস(Bluetooth Device)-টির এলইডি(LED) টা মিটমিট করে জ্বলা বন্ধ করে দেবে, অর্থাৎ দুটি ডিভাইসের মধ্যে যোগাযোগ স্থাপন সফল হয়েছে ।

এই দুটো জিনিস থেকেই তুমি বুঝতে পারছো যে প্রথম ধাপে যখন খালি ব্লুটুথ(Bluetooth)- টি মাইক্রোকন্ট্রোলার(microcontroller)-এর সাথে যুক্ত হচ্ছে, তখন যোগাযোগ সফল হলে ব্লুটুথ(Bluetooth) এলইডি(LED)- টি মিটমিট করে জ্বলতে থাকবে । কিন্তু অন্য ডিভাইস(Device)-এর সঙ্গে যোগাযোগ স্থাপন হলে, তখন এটি স্থির হয়ে যাবে ।

(Refer Slide Time: 05:12)



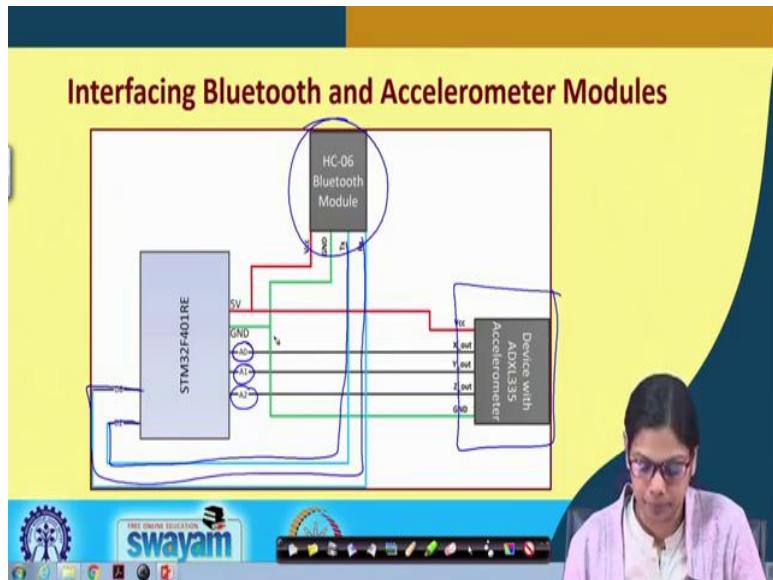
প্রথমে আমি তোমাদেরকে একটা পরীক্ষামূলক প্রোগ্রাম(Program) দেখাবো, যার সাহায্যে আমি STM বোর্ড- এর ব্লুটুথ মডিউল(Bluetooth Module)-টির সাথে আমার মোবাইল ফোন(Mobile Phone)-এর ব্লুটুথ মডিউল(Bluetooth Module)-টির যোগাযোগ করব, এবং তারপর সেটি কিছু বার্তা দেখাবে। মাইক্রোকন্ট্রোলার(microcontroller)বোর্ড থেকে কিছু বার্তা আমার মোবাইল ফোন(mobile phone)- এ চলে যাবে ব্লুটুথ(Bluetooth)- এর মাধ্যমে।

প্রথমে, আমাদেরকে সিরিয়াল(Serial) যোগাযোগ স্থাপন করতে হবে; আমি “Bluetooth” নামটি ব্যবহার করেছি D8 এবং D2 পিন দুটির মধ্যে যোগাযোগটিকে বোঝানোর জন্য। যখন আমরা কোড(code) লিখব, তখন এইটিই প্রথম জিনিস যার মান আমাদেরকে দিয়ে দিতে হবে। এবং তারপর মেন(main)- এর ডেতর আমরা কি করছি, আসলে এটা আমাদের দরকার নেই। মেন(main) এর মধ্যে আমরা কি করছি, না, ব্লুটুথ(Bluetooth) নামক যোগাযোগ টি রয়েছে D8 এবং D2 পিনের মধ্যে, তার সাথে printf ব্যবহার করে “Hello” লেখাটি প্রিন্ট(Print) করছি।

তো একবার, আমার ফোনের ব্লুটুথ মডিউল(Bluetooth Module)-এর সাথে STM- এর ব্লুটুথ মডিউল(Bluetooth Module)-এর যোগাযোগ স্থাপন হয়ে গেলে, আমরা ডেটা (Data) আদান-প্রদান করব। ডেটা (Data) আদান প্রদান এইভাবে হবে, যে অবজেক্ট (object)-টি আমরা তৈরি করেছি সেই অবজেক্ট (object)টি printf- এর মাধ্যমে Hello প্রিন্ট(Print) করবে। তারপর 5 সেকেন্ড(Second) অপেক্ষা করবো আমরা। এবং তারপর এই কয়েক লাইন বার্তা প্রিন্ট(Print) করব।

ব্লুটুথ(Bluetooth) সহ মাইক্রোকন্ট্রোলার(microcontroller)- টি একটিই ডিভাইস (device) হিসেবে কাজ করবে, এবং আরেকটি ডিভাইস(device) হবে তোমার মোবাইল ফোন (mobile phone) যার মাধ্যমে তুমি যোগাযোগটি করবে। এই কোড-টা খুবই সহজ, যেখানে বস্তুত আমরা একটাকে টিএক্স(TX) এবং আরেকটাকে আরএক্স(RX) ধরে যোগাযোগটা বানিয়েছি।

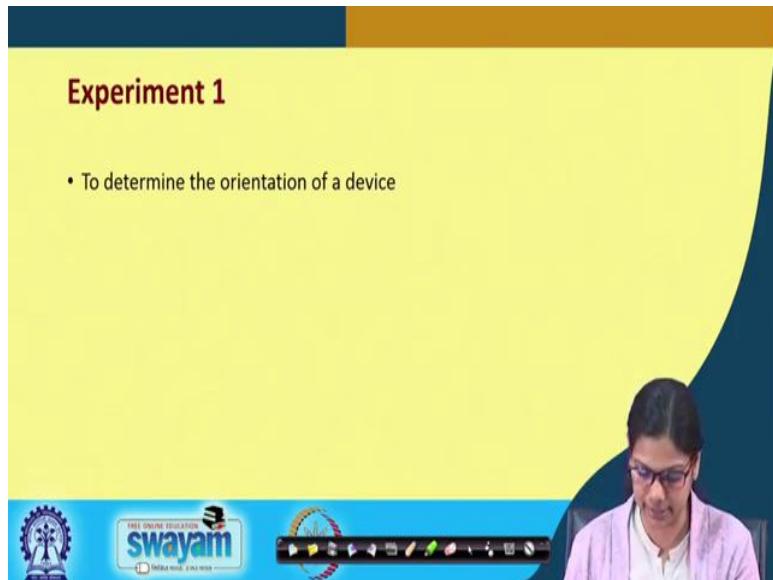
(Refer Slide Time: 08:14)



এখন, ইন্টারফেসিং(Interfacing)-টি নিয়ে আমরা এগিয়ে যাব। এটা খুবই সহজ যেহেতু ব্লুটুথ(Bluetooth)-এর ব্যাপারে আমি আগেই আলোচনা করেছি।

অ্যাক্সেলেরোমিটার(Accelerometer)- টিকে. যুক্ত করার পদ্ধতিটা আমি আগেই আলোচনা করেছি। আমরা এই যোগাযোগটাকে বানাতে পারি, এবং এনালগ পোর্ট A0, A1, এবং A2 - র সাথে যুক্ত করতে পারি। ব্লুটুথ মডিউল(Bluetooth Module)-এর সাথে একসাথে, এটি D2 তার সঙ্গে টিএক্স(TX)- এর সাথে ও D8 তারই সঙ্গে আরএক্স(RX)- এর সাথে যুক্ত হবে। এবং ব্লুটুথ(Bluetooth)-এর Vcc, GND অ্যাক্সেলেরোমিটার(Accelerometer)- এর Vcc, GND- র সাথে যুক্ত হবে।

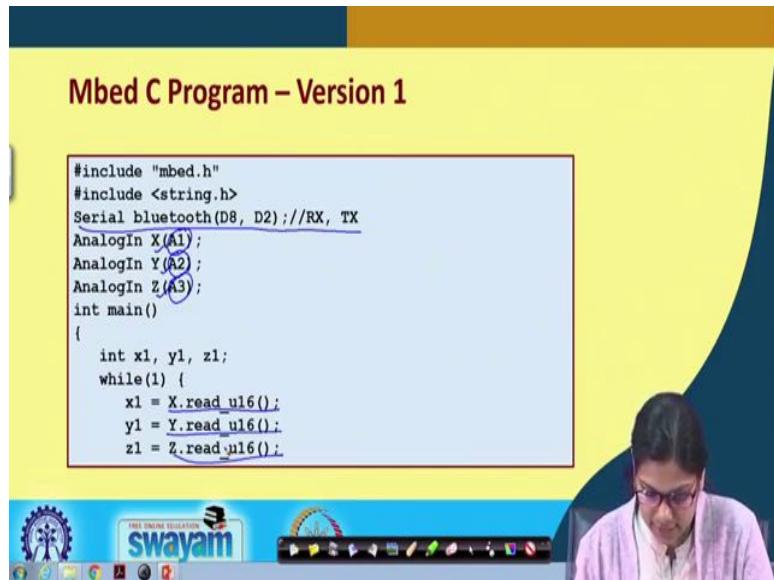
(Refer Slide Time: 09:13)



প্রথম পরীক্ষাটিতে আসা যাক। আমি প্রথমে কোড- টা আলোচনা করব, এবং তারপর এটি যে কাজ করছে তা দেখবো। প্রথম পরীক্ষায় আমরা ডিভাইসটির অবস্থানগত দিকটি নির্ণয় করব। আগের আলোচনায়, আমরা তোমাদেরকে দেখিয়েছিলাম যে x, y, z স্থানাঙ্কের সংগত কিছু মান পাচ্ছিলাম আমরা। সেই মান- এর উপর ভিত্তি করে, আমাদেরকে পরীক্ষাটা করতে হবে। যখন অ্যাক্সেলেরোমিটার(Accelerometer)-টিকে, ডিভাইসটির(Device) সাথে রাখবো, এবং সেটিকে উল্লম্বভাবে সোজা করে দেবো, তখন কি হবে ইত্যাদি।

এবং সেইমতো আমার প্রোগ্রাম(Program)-টা লিখব। যদি x স্থানাঙ্কের মানটি এটা এবং এটার মধ্যে থাকে, অথবা y স্থানাঙ্কের মানটি এটা এবং এটার মধ্যে থাকে অথবা z স্থানাঙ্কের মান টি এটা এবং এটার মধ্যে থাকে তাহলে হয় এটা উল্লম্বভাবে উপরের দিকে মুখ করে দাঁড়িয়ে আছে, অথবা উল্লম্বভাবে নিচের দিকে মুখ করে দাঁড়িয়ে আছে। পরীক্ষাটিতে, আমরা প্রথমে ডিভাইস(Device)-টির অবস্থানগত দিকটির প্রতি নজর রাখব। ব্লুটুথ মডিউল(Bluetooth Module) এবং অ্যাক্সেলেরোমিটার(Accelerometer) সহ বর্তনীর চিত্রটি আগের মতই থাকবে।

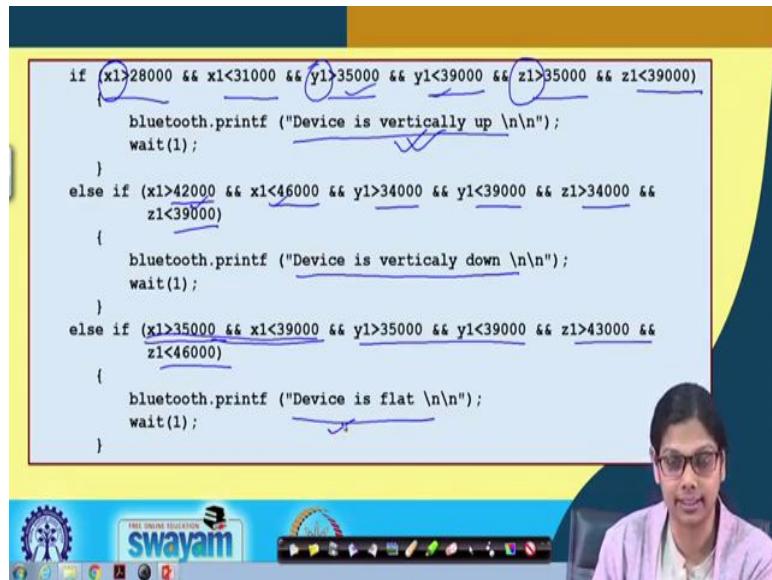
(Refer Slide Time: 10:44)



এইটা হচ্ছে Mbed C কোড। ব্লুটুথ মডিউল(Bluetooth Module)-এ, দেখা যাবে যে এটি উল্লম্বভাবে উপরের দিক করে আছে নাকি উল্লম্বভাবে ডান দিকে আছে। এরপর আমরা X, Y এবং Z আউটপুটগুলি A1, A2, এবং A3-র সাথে যুক্ত করব, তুমি যে কোন এনালগ পোর্ট(Analog Port)-এর সাথে যুক্ত করতে পারো।

তারপর আমরা মানটাকে পড়বো। আমরা তিনটে মান পড়ছি যথা X.read_u16, Y.read_u16, এবং Z.read_u16। মানগুলি পড়ার পর, আমাদেরকে যাচাই করতে হবে।

(Refer Slide Time: 11:38)



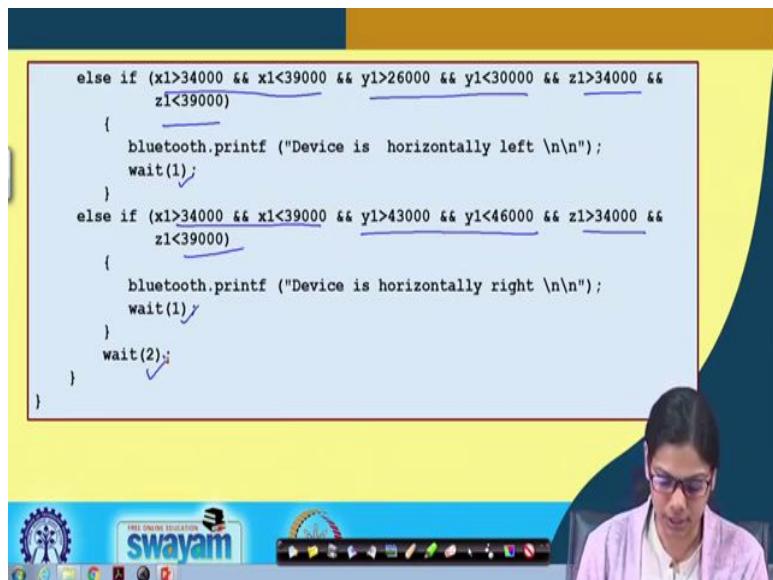
এখানে x_1 -এর মধ্যে রয়েছে x_OUT - এর মান, y_1 - এর মধ্যে রয়েছে y_OUT - এর মান, এবং z_1 - এর মধ্যে রয়েছে z_OUT - এর মান। আমরা দেখেছি যে যখন x_1 - এর মান এই দুটির মধ্যে রয়েছে, y_1 -এর মান এই দুটির মধ্যে রয়েছে, z_1 -এর মান এই দুটির মধ্যে রয়েছে, তখন ধরে নিচ্ছি আমরা যে ডিভাইস(Device)-টি উল্লম্বভাবে উপরের দিকে মুখ করে রয়েছে।

এই কাজটি করার পর, `bluetooth.printf` ফাংশন(Function)-টি উপরুক্ত বার্তা প্রিন্ট(Print) করবে। অন্যান্য অবস্থানগত দিকগুলির জন্য একই জিনিস ঘটবে।

আমরা যা করেছি এটি শুধুমাত্র তার একটি নমুনা। যখন তোমরা এটা করবে, আমি ইতিমধ্যেই দেখিয়েছি যে কিভাবে মান গুলিকে পর্যবেক্ষণ করতে পারবে, যদি আরডুইনো(Arduino) ব্যবহার করো তাহলে তো সিরিয়াল মনিটর- এই দেখতে পাবে, নচেৎ কুলটার্ম(CoolTerm) ব্যবহার করতে হবে। তুমি ডিভাইস(Device)-টিকে উল্লম্বভাবে উপরের দিকে মুখ করে ধরবে এবং তারপর দেখবে x, y, z স্থানাঙ্কের মানগুলি কিরকম আসছে।

সেইমতো তুমি কোড(Code) লিখবে । সেইরকম ভাবে, আরো কিছু শর্ত আছে যেমন ডিভাইস(Device)-টি চিৎ হয়ে শুয়ে আছে, তো যখন এটি চিৎ হয়ে শুয়ে আছে তখন তুমি দেখবে x এবং y স্থানাঙ্কের মানগুলি প্রায় একই রকম কিন্তু z-এর স্থানাঙ্কের মানটি এদের মধ্যে সর্বোচ্চ ।

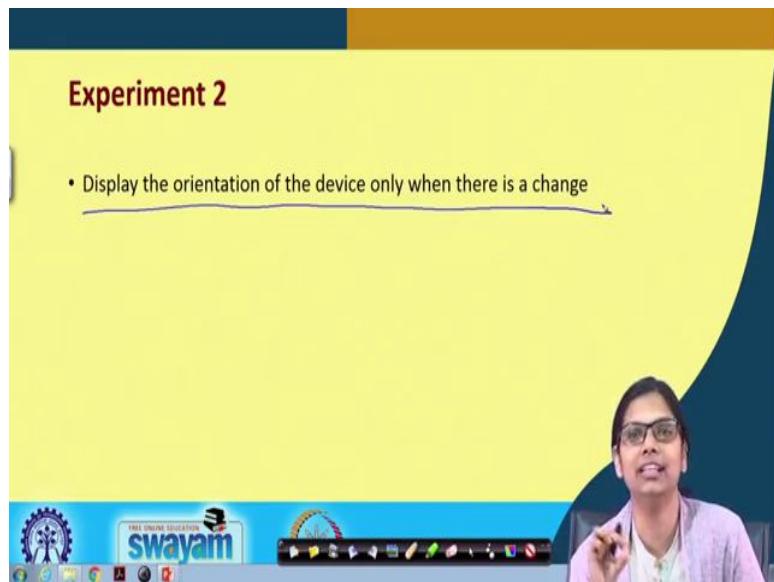
(Refer Slide Time: 14:14)



যদি ডিভাইস(Device)-টি সমান্তরালভাবে বাম দিকে থাকে তাহলে আমাদের আরও কিছু শর্ত আছে । যদি ডিভাইস(Device)-টি সমান্তরালভাবে বাম দিকে থাকে, তখন x1-এর মান এবং z1-এর মানও থাকে 34000 থেকে 39000-এর মধ্যে, এবং y1-এর মান এক্ষেত্রে কিছুটা কম হবে । সেই রকম ভাবেই, এখানে আবার উল্লেখ ঘটবে; এই মানটি এই পরিধির মধ্যে আছে, কিন্তু y1-এর মান সর্বোচ্চ, তা ইঙ্গিত করছে যে এটি সমান্তরালভাবে ডানদিকে রয়েছে । তো, এটা নির্ভর করে যে অ্যাক্সেলেরোমিটার(accelerometer)-টিকে ডিভাইস(Device)-টির মধ্যে কিভাবে বসিয়েছো । তোমাকে সেটা দেখতে হবে এবং সেই মতো তোমার প্রোগ্রাম(Program) লিখতে হবে ।

এবং তারপর প্রতিবারে একবার করে দেখানোর পর 1 সেকেন্ড(Second) অপেক্ষা করবে, এবং তারপর আরো দুই সেকেন্ড(Second) অপেক্ষা করবে এবং তারপর সবকিছু আবার পুনরায় শুরু থেকে চলতে থাকবে ।

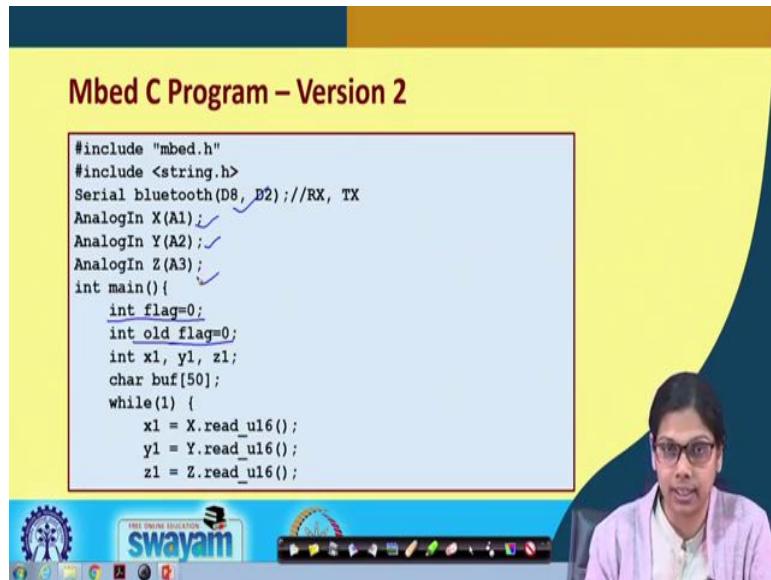
(Refer Slide Time: 15:25)



অন্য একটা পরীক্ষায় যাওয়া যাক। পূর্বের পরীক্ষাটিতে, আমরা এটাকে এমন ভাবে করেছিলাম
যে যে তখনই ডিভাইস(Device)-টির অবস্থানগত দিকের পরিবর্তন ঘটুক কি না ঘটুক, প্রতি দুই
সেকেন্ড(Second) অন্তর এটি ডিভাইস(Device)-টির অবস্থাটিকে দেখাবে।

এখানে, আমরা খালি একটাই পরিবর্তন করেছি, এটা তখনই ডিভাইসটির অবস্থানগত দিকটি
দেখাবে শুধুমাত্র যদি সেটির কোন পরিবর্তন হয়ে থাকে। তার মানে, যদি ডিভাইসটি চিং হয়ে থাকে,
তা হলে চিং হয়েই থাকবে। যদি সমান্তরাল অবস্থা থেকে সেটি উল্লম্বভাবে উপরের দিকে মুখ করে
থাকা অবস্থায় অথবা উল্লম্বভাবে নিচের দিকে মুখ করে থাকা অবস্থায় পরিবর্তিত হয়, তখনই
একমাত্র ব্লুটুথ মডিউল(Bluetooth Module) - এ এটির অবস্থানগত দিকটি দেখানো হবে।
আমাকে দেখাতে দাও, আমি এখানে কি পরিবর্তন করেছি।

(Refer Slide Time: 16:27)

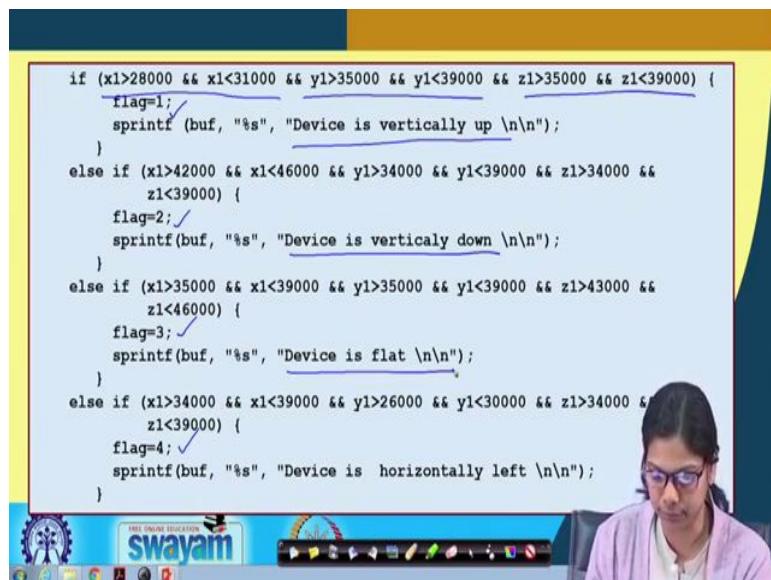


Mbed C Program – Version 2

```
#include "mbed.h"
#include <string.h>
Serial bluetooth(D8, D2); //RX, TX
AnalogIn X(A1);
AnalogIn Y(A2);
AnalogIn Z(A3);
int main() {
    int flag=0;
    int old_flag=0;
    int x1, y1, z1;
    char buf[50];
    while(1) {
        x1 = X.read_u16();
        y1 = Y.read_u16();
        z1 = Z.read_u16();
```

কোড- এর এই অংশটা একই থাকছে, খালি দুটো ভ্যারিয়েবেল(Variable) নিয়েছি আমরা | একটি হচ্ছ `flag`, শুরুতে যেটির মান রাখবো 0 | আরেকটি হচ্ছ `old_flag`, এটিরও মান শুরুতে 0-ই রাখবো | তো, এই দুটো জিনিস আমরা করেছি | অন্য জিনিসগুলো, সব একই রকম রয়েছে | আমরা সিরিয়াল যোগাযোগ তৈরি করেছি, A1, A2, A3 এনালগ পোর্ট(Analog Port) গুলির সাথে অ্যাক্সেলেরোমিটার(Accelerometer)- টির x, y, z যুক্ত করেছি।

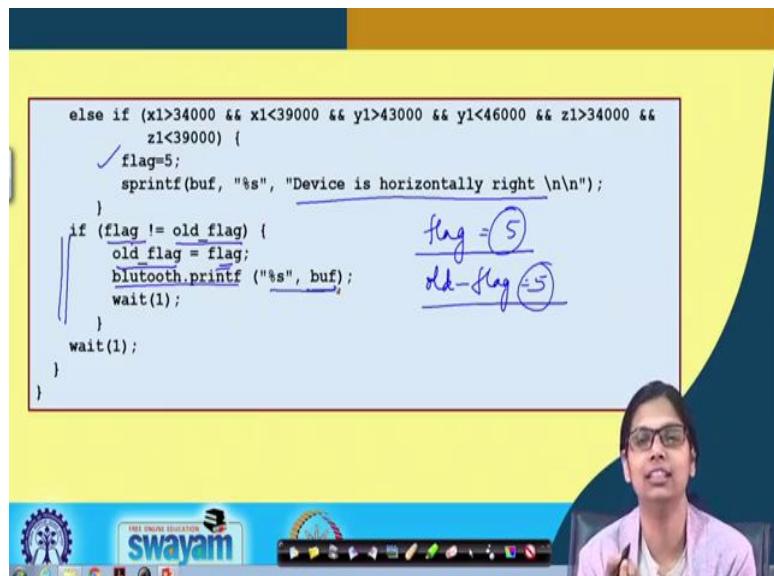
(Refer Slide Time: 17:01)



```
if (x1>28000 && x1<31000 && y1>35000 && y1<39000 && z1>35000 && z1<39000) {
    flag=1;
    sprintf(buf, "%s", "Device is vertically up \n\n");
}
else if (x1>42000 && x1<46000 && y1>34000 && y1<39000 && z1>34000 &&
z1<39000) {
    flag=2;
    sprintf(buf, "%s", "Device is vertically down \n\n");
}
else if (x1>35000 && x1<39000 && y1>35000 && y1<39000 && z1>43000 &&
z1<46000) {
    flag=3;
    sprintf(buf, "%s", "Device is flat \n\n");
}
else if (x1>34000 && x1<39000 && y1>26000 && y1<30000 && z1>34000 &&
z1<39000) {
    flag=4;
    sprintf(buf, "%s", "Device is horizontally left \n\n");
}
```

এবাব এখানে প্রোগ্রাম(Program)-টা দেখা যাক, তুমি যদি একটা শর্তকে দেখো দেখবে যে শর্তগুলি একই আছে। যখনই এটা হয় উল্লম্বভাবে উপরের দিকে মুখ করে আছে অথবা উল্লম্বভাবে নিচের দিকে মুখ করে আছে, তখন আমরা flag ভ্যারিয়েবেল(variable)-টিতে প্রথমবারের মতো মান রাখছি। প্রথম ক্ষেত্রে, flag- এর মান রাখছি 1। এর পরেরটিতে, flag- এর মান রাখছি 2। তৃতীয় ক্ষেত্রে, flag- এর মান হচ্ছে 3। এরপর, অবস্থানগত দিকটির উপর নির্ভর করে flag- এর মান হতে পারে 4।

(Refer Slide Time: 17:44)



এবং তারপর, আমরা আবাব দেখছি যে যখন এটি সমান্তরালভাবে ডান দিকে থাকবে তখন flag- এর মান 5 হবে। এবং এই যাচাই টিতে, আমরা flag- এর মান পরীক্ষা করছি। (শুরুতে flag- এর মান শূন্য ছিল এবং উপরিউক্ত শর্তগুলির মধ্যে প্রবেশ করলে flag- এর মান এক, দুই, 3, 4, 5- এর মধ্যে যেকোনো একটি হবে)

যদি flag- এর মান old_flag- এর সমান না হয়, তার মানে কিছু পরিবর্তন ঘটেছে। প্রথমবারের জন্য শর্তটি সত্য হবে, কারণ flag-এর মান old_flag- এর সমান হবে না। old_flag- এর মান টি flag- এর সমান করে দেওয়া হবে।

ধরা যাক, ডিভাইস(Device)-টি সমান্তরালভাবে ডান দিকে ছিল। তাহলে flag- এর মান ৫ হবে। প্রথমবারের জন্য, old_flag- এর মানটি এই মান টির সমান ছিল না। তো, old_flag- এর মান হয়ে যাবে ৫ এখানে, এবং তারপর আমরা প্রিন্ট(Print) করব “Device is horizontally right”। আমরা এক মিনিট অপেক্ষা করবো, এবং আবার পরীক্ষা করব যে flag এবং old_flag- এর মান দুটি অসম কিনা, কিন্তু না, এবার এরা দুটি সমান। তো, যা দেখা যাচ্ছিল তাই থাকবে।

ধরা যাক এটি সমান্তরালভাবে ডান দিকে ছিল, আমরা এটিকে সমান্তরালভাবে বাঁদিকে পরিবর্তন করে দিলাম। সেক্ষেত্রে, কি হবে, না, এই flag- এর মান টি পরিবর্তন হয়ে যাবে। তো, ব্লুটুথ(Bluetooth) এইরকম একটা বার্তা দেখাবে যে “device is flat” ইত্যাদি।

(Refer Slide Time: 19:37)

Other Application

- Health monitoring
 - Running
 - Walking
 - Sitting
 - Detect for abnormal change in posture of a person

এই দুটি, পরীক্ষায় আমরা তোমাদেরকে হাতে-কলমে করে দেখাবো। অ্যাক্সেলেরোমিটার- এর অন্যান্য কার্যকারিতাও আছে। তার একটি জলজ্যান্ত উদাহরণ হচ্ছে আমাদের মোবাইল ফোন(Mobile Phone) - এ স্বাস্থ্য নজরদারি সংক্রান্ত অ্যাপ(App)গুলি।

বস্তুত, এটা তোমাকে বলবে যে সারাদিনে তুমি কতক্ষণ দৌড়েছ কিংবা কতক্ষণ হেঁটেছ, কতক্ষণ বসেছিলে ইত্যাদি। তোমার সারাদিনের কার্যকলাপের হিসাব রাখতে পারবে তুমি। এর আরো একটি গুরুত্বপূর্ণ প্রয়োগ হলো, যে বয়স্ক মানুষ পড়ে গেলে তা নির্ণয় করা।

দেখো, বেশিরভাগ সময়ই বয়স্ক লোকেদের বাড়িতে একা থাকতে হয়। যদি তাদেরকে দেখার লোক থাকেও, কিন্তু এরকম পরিস্থিতি হতেই পারে, যে তারা একদম একলা রয়েছেন। এবং সেই রকম পরিস্থিতিতে, যখন চারিদিকে কেউ নেই এবং যদি কিছু গুরুতর ঘটনা ঘটে যেমন তারা বাথরুমে পড়ে গেলেন অথবা শুধু বাড়ির মধ্যে পড়ে গেলেন, তখন আত্মীয়-পরিজন পাড়াপড়শিরা জানবেন কিভাবে।

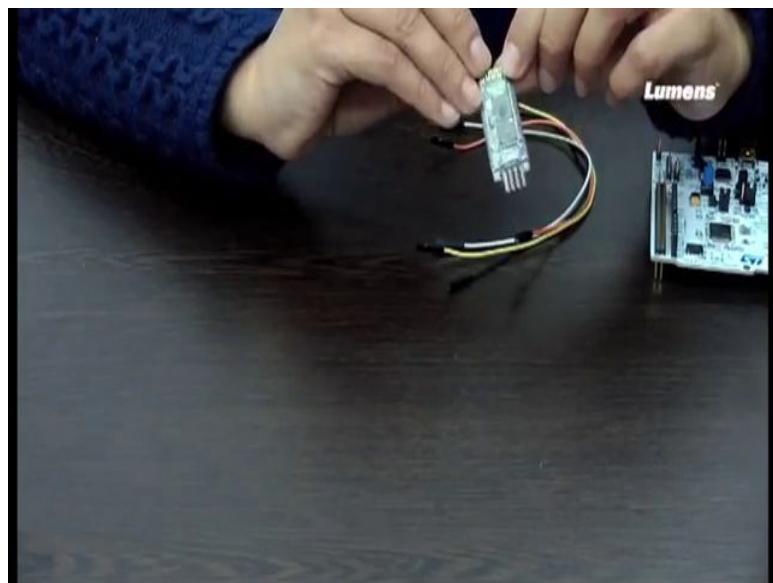
এই পড়ে যাওয়া নির্ণয় করার ব্যাপারটি অ্যাক্সেলেরোমিটার(Accelerometer)- এর সাহায্যে খুব সুন্দর ভাবে প্রয়োগ করা যেতে পারে, কিন্তু খেয়াল রাখতে হবে বিভিন্ন ভুল-ভান্তির দিকে, যেমন এটি যেন অহেতুক এলার্ম(Alarm)- এর সৃষ্টি না করে। তো, আমরা এই আলোচনার শেষে চলে এসেছি। এবার, আমরা পুরো ব্যাপারটা কিভাবে কাজ করছে সেটা দেখাবো।

শেষ সপ্তাহে, আমরা অ্যাক্সেলেরোমিটার(Accelerometer) নিয়ে কাজ করছি। এবং এটা কি করে, তা আমি আগেই ব্যাখ্যা করেছি। আমি কি করতে চাই যে অ্যাক্সেলেরোমিটার(Accelerometer)- এর সাহায্যে যে স্থানাঙ্ক গুলি আমরা পাচ্ছি, সেগুলিকে আমার মোবাইল ফোন- এ পাঠিয়ে দেব। তাহলে আমাদেরকে কিছু যোগাযোগ পদ্ধতির ব্যবস্থা করতে হবে যার মাধ্যমে অ্যাক্সেলেরোমিটার(Accelerometer) যুক্ত

মাইক্রোকন্ট্রোলার(microcontroller) এবং আমার মোবাইল ফোন(mobile phone) একে অপরের সাথে বার্তালাপ করতে পারে।

তো, এই পদ্ধতিতে আমাদের একটা ব্লুটুথ মডিউল(Bluetooth Module) লাগবে। আমরা HC-06 ব্লুটুথ মডিউল (Bluetooth Module) ব্যবহার করব, যেখানে মাইক্রোকন্ট্রোলার(microcontroller) থেকে আমার মোবাইলে ডেটা(data) পাঠাবে। তো, এইটা আমাদের কি জন্য প্রয়োজন? কারণ পরে আমরা অ্যাক্সেলেরোমিটার(accelerometer)- টিকে যুক্ত করব এবং x, y, z স্থানাঙ্কগুলি ব্লুটুথের(Bluetooth) মাধ্যমে আমার মোবাইল ডিভাইস(mobile device)- এ পাঠাবো।

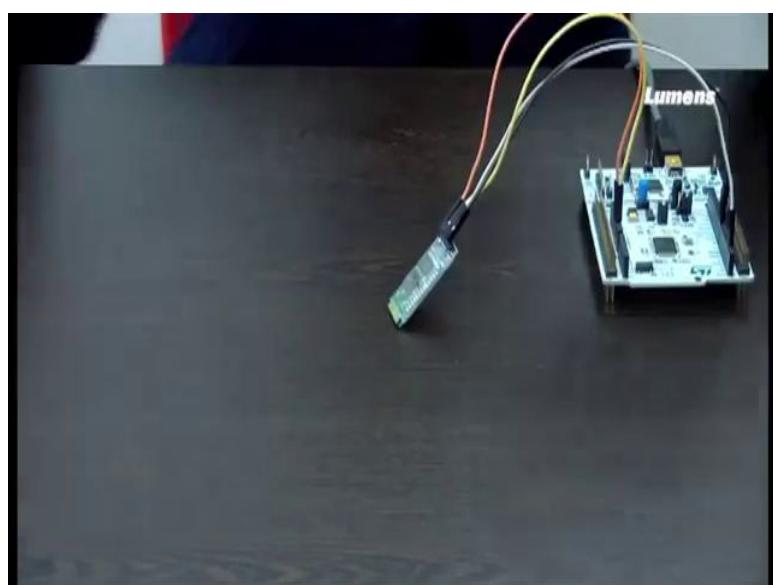
(Refer Slide Time: 23:50)



এই ব্লুটুথ(Bluetooth)-টাকে STM বোর্ডের সাথে কিভাবে যুক্ত করছি সেটা দেখা যাক। এইটা হচ্ছে HC-06 ব্লুটুথ মডিউল(Bluetooth Module), তোমরা দেখতে পাচ্ছ এটিতে চারটে পিন আছে, তো এই চারটি পিন এইভাবে যাবে। একটি হচ্ছে আরএক্স(RX,), একটি হচ্ছে টিএক্স(TX), একটি হচ্ছে GND, এবং আরেকটি হচ্ছে Vcc।

STM মাইক্রোকন্ট্রোলার(microcontroller)- টির GND, Vcc, TX, এবং RX পিন গুলির সাথে যুক্ত করব। প্রথমে এই দুটো যুক্ত করব, এবং মাইক্রোকন্ট্রোলার(microcontroller) থেকে আমার ডিভাইস(device)- এ একটা ডাটা পাঠাবো। এটা করার জন্য তোমাদের আরো একটা জিনিস লাগবে তোমাদেরকে একটা ব্লুটুথ(Bluetooth) টার্মিনাল(Terminal) ইনস্টল(Install) করতে হবে, এই ব্যাপারটায় আমি একটু পরে আসছি।

(Refer Slide Time: 25:05)



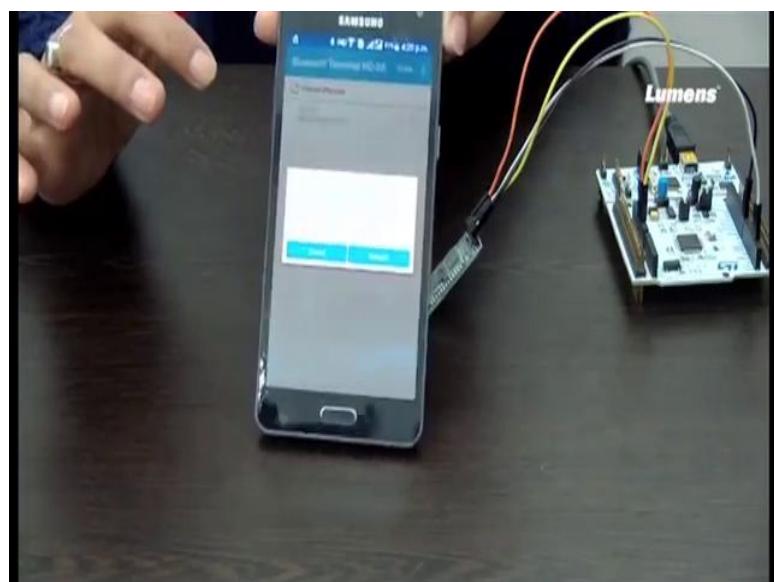
প্রথমে আমাকে দেখাতে দাও যে সংযোগ টা আমাদের করতে হবে। আমি যেরকম বললাম এইটা হচ্ছে Vcc, যেটি এইটার সঙ্গে যুক্ত হবে, এইটা হচ্ছে GND, যেইটা STM বোর্ড- এর গ্রাউন্ড(Ground) পোর্ট- এর সঙ্গে যুক্ত হবে। এবং তারপর আছে আমাদের টিএক্স(TX) এবং আরএক্স(RX)। টিএক্স(TX) যুক্ত হবে D2 পোর্ট- এর সাথে। এবং আরএক্স(RX) যুক্ত হবে D8- এর সাথে।

(Refer Slide Time: 26:40)



একবার ব্লুটুথ মডিউল(Bluetooth Module) -টা কানেক্ট(Connect) হয়ে গেলে, তোমার একটা ব্লুটুথ টার্মিনাল(Bluetooth Terminal) লাগবে। আমি ইতিমধ্যেই একটা ব্লুটুথ টার্মিনাল(Bluetooth Terminal) ইনস্টল(install) করে নিয়েছি, এইটা হচ্ছে সেটি।

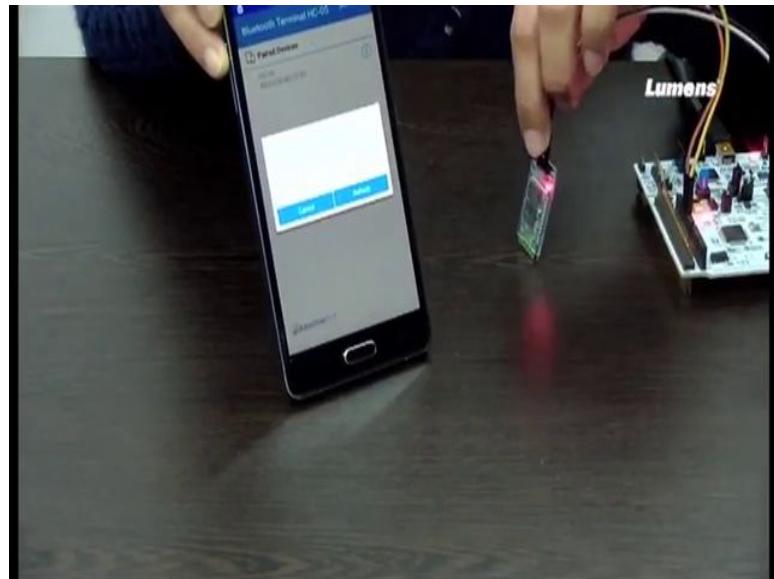
(Refer Slide Time: 27:03)



তো এখানে তুমি স্ক্যান(Scan) করতে পারবে, তখন এটি স্ক্যান করবে এবং কাছাকাছি যে যে ডিভাইস(Device)গুলো থাকবে সেগুলির নাম দেখাবে। তোমরা দেখতে পাচ্ছ আমি ঘন্থন

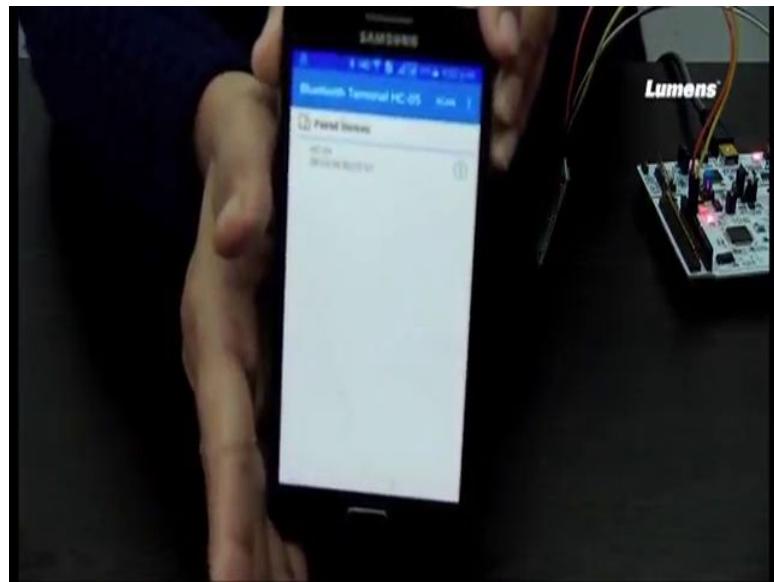
কানেক্ট(Connect) করছি তখনই এলইডি(LED)-টি মিটমিট করে জ্বলছে, তার মানে এটি যোগাযোগ স্থাপনে প্রস্তুত, এটি যেকোন কাছাকাছি ডিভাইস(Device) এর সাথে যোগাযোগ স্থাপন করতে পারবে।

(Refer Slide Time: 28:07)



এখন, এইটা হচ্ছে আমার মোবাইল(mobile) টা, যেটা দিয়ে আমি স্ক্যান করব প্রথমে। আমরা স্ক্যান করছি, এবং তোমরা একটা জিনিস লক্ষ্য করো যে ব্লুটুথ মডিউল(Bluetooth Module) টিতে এলইডি(LED)-টি মিটমিট করে জ্বলছে। দেখাচ্ছে “No device found”।

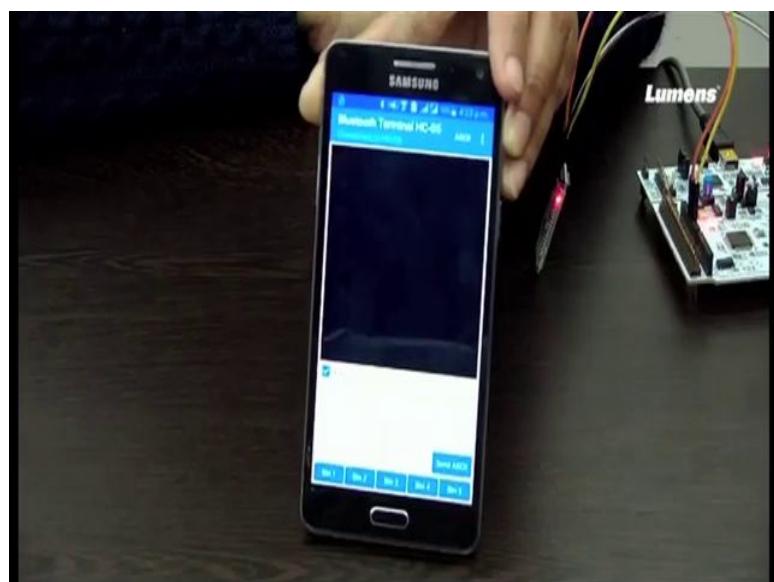
(Refer Slide Time: 29:09)



এখন, এটা কিছু ডিভাইস- এর নাম দেখাচ্ছে , যেগুলোর সাথে আমি আগে কানেক্ট করেছিলাম ।

এই যে “pair device” দেখাচ্ছে এখানে। এই হচ্ছে সেই ডিভাইসটা, তো আমি HC-06 চয়ন করলাম।

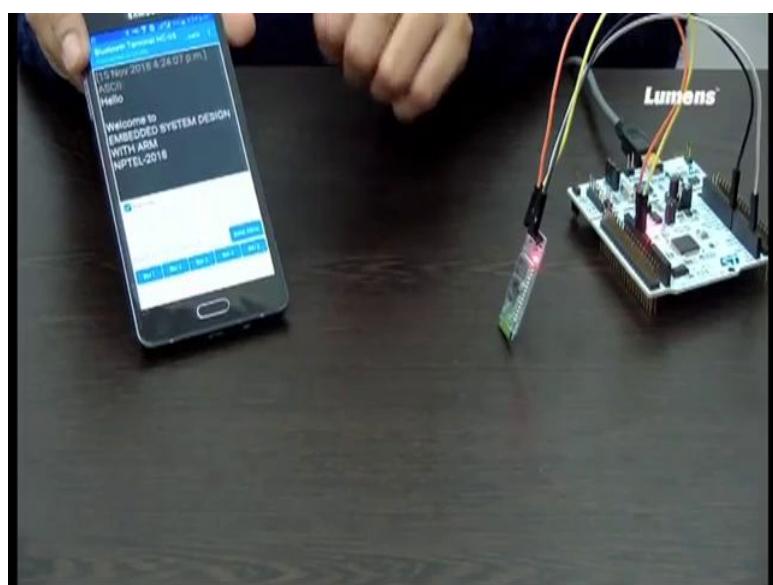
(Refer Slide Time: 29:29)



আমি এটার উপর ক্লিক করব, এবং তোমরা দেখতে পাচ্ছা যে একটা টার্মিনাল আসছে। এবার দেখো কি ঘটেছে । যেই একবার যোগাযোগ স্থাপন হয়ে গেছে মোবাইল টির সাথে,

এলইডি(LED)-টি মিটমিট করা বন্ধ করে দিয়েছে। এবার আমি যোগাযোগটা বিচ্ছিন্ন করে আর একবার দেখাবো। এবং তোমরা দেখছো জে এলইডি(LED) টি আবার মিটমিট করে জ্বলতে শুরু করে দিয়েছে, তো এটি যোগাযোগ স্থাপনের জন্য প্রস্তুত। আবার যেই কানেক্ট(Connect) করছি, এটি স্থির হয়ে যাচ্ছে তার মানে যোগাযোগ স্থাপন হয়ে গেছে। এবার, এই ব্লুটুথ(Bluetooth)-এর মাধ্যমে আমি মাইক্রোকন্ট্রোলার(microcontroller) থেকে মোবাইলে দুটি বার্তা পাঠাবো।

(Refer Slide Time: 30:45)

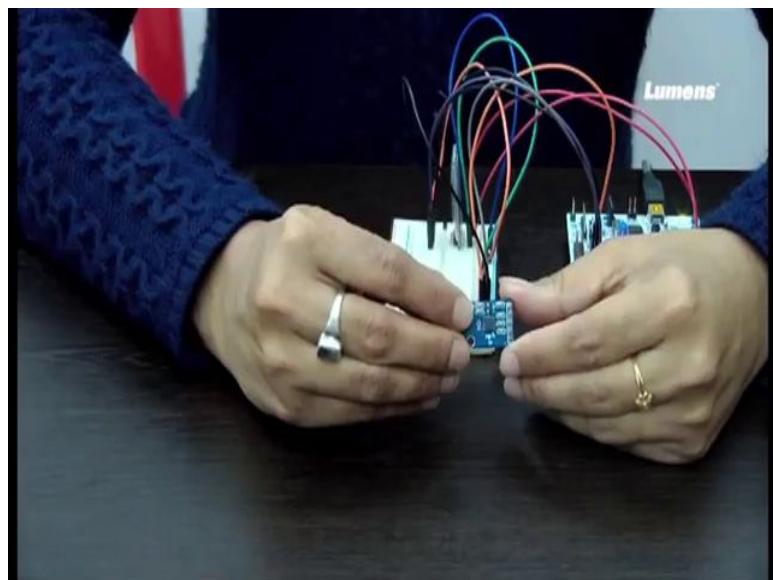


তো, তোমরা দেখতে পাচ্ছ যে বার্তা দুটো চলে এসেছে। এই ভাবেই তুমি একটা একটা ব্লুটুথ মডিউল(Bluetooth Module)-কে যেকোনো ডিভাইসের সাথে কানেক্ট(Connect) করতে পারো এবং তারপর তোমার মাইক্রোকন্ট্রোলার(microcontroller) এবং এই যোগাযোগ স্থাপনকারী ডিভাইস(Device)-টির মাধ্যমে তোমার মোবাইল এ ডেটা(Data) পাঠাতে পারো।

আজকে আমি তোমাদের আরো একটা পরীক্ষা দেখাবো, যেখানে অ্যাক্সেলেরোমিটার(Accelerometer) এবং ব্লুটুথ(Bluetooth) একসঙ্গে ব্যবহার করব। এবং তারপর আমরা যে কোন একটি বন্ধুর অবস্থানগত দিকটি বিশ্লেষণ করবো। তো, অ্যাক্সেলেরোমিটার- টি x,y,z স্থানাঙ্ক ওলি পাবো।

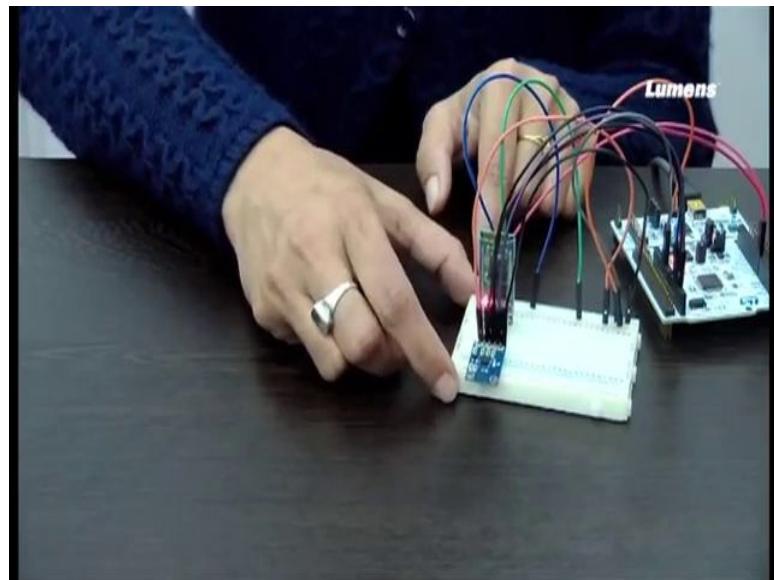
এরপর মাইক্রোকন্ট্রোলার নিজের মত বিশ্লেষণ করে সিদ্ধান্তে আসবে যে ডিভাইসটি চিৎ হয়ে গেছে নাকি উল্লম্বভাবে উপরের দিকে মুখ করে আছে নাকি উল্লম্বভাবে নিচের দিকে মুখ করে আছে নাকি সমান্তরালভাবে বামদিকে আছে নাকি সমান্তরালভাবে ডান দিকে আছে।

(Refer Slide Time: 33:56)



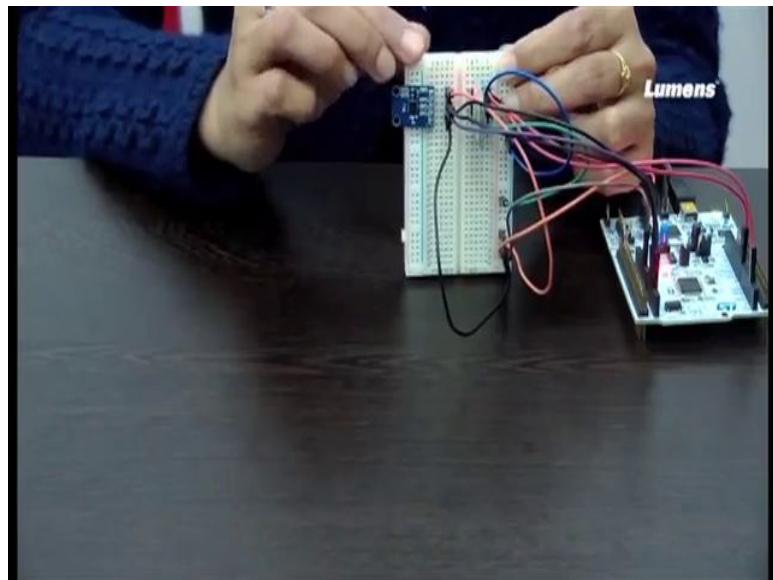
তো তোমরাই অ্যাক্সেলেরোমিটার(Accelerometer)-টা দেখতে পাচ্ছ, এতে পাঁচটি পিন রয়েছে।
এই পিন গুলি হল Vcc, x, y, z এবং GND।

(Refer Slide Time: 34:30)



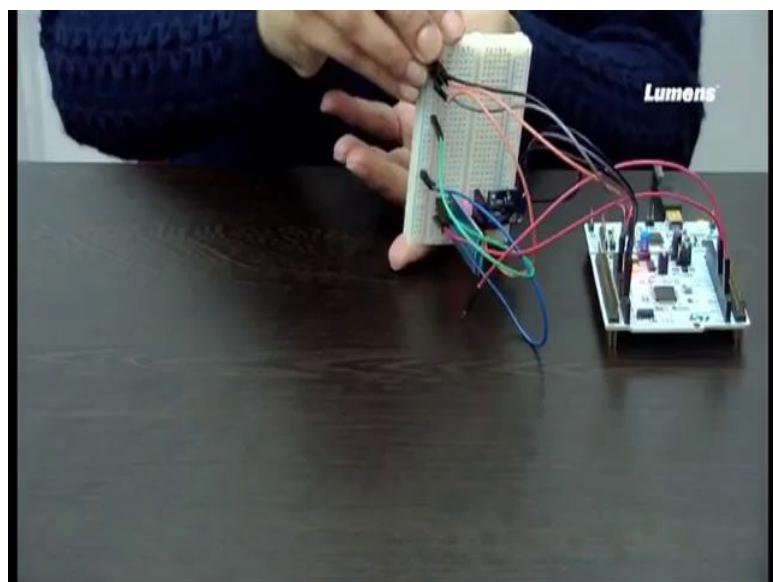
আমি এটি ব্রেডবোর্ড(BreadBoard) এর সাথে আটকাবো যেখানে আমি ইতিমধ্যেই প্রয়োজনীয় তারসমত যোগাযোগগুলি করে রেখেছি। তোমরা এটা দেখতে পাচ্ছ আমি Vcc - র সাথে যুক্ত করেছি, এটাকে GND- র সাথে যুক্ত করেছি, এবং তারপর হচ্ছে x,y এবং z। এবং ব্লুটুথ(Bluetooth) থেকে কিভাবে যোগাযোগ করতে হবে সেটা তো আমরা আগেই দেখেছি। এবং তোমরা দেখতে পাচ্ছ যে ব্লুটুথ(Bluetooth)-এর লাল এলাইডি(LED) টি মিটমিট করে জ্বলছে অর্থাৎ এটি যোগাযোগের জন্য তৈরি, কিন্তু এখনো কোন ডিভাইস এর সাথে যোগাযোগ করেনি।

(Refer Slide Time: 35:19)



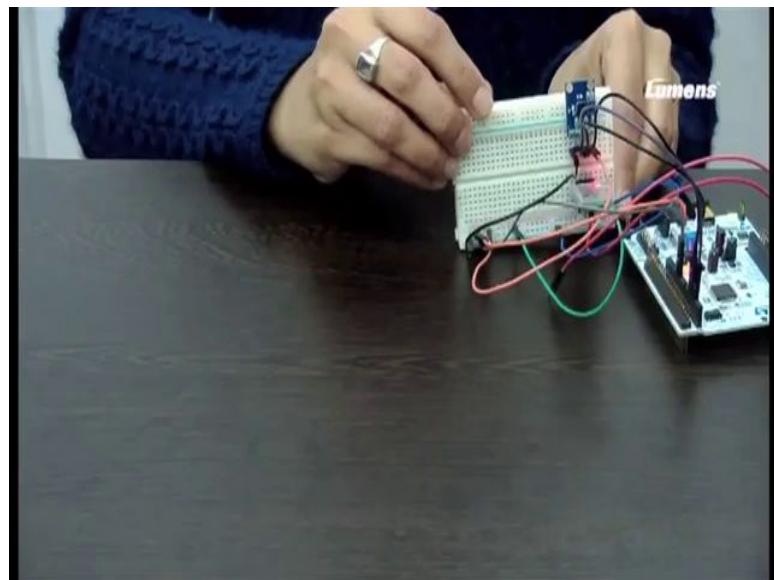
এখন, আমরা প্রোগ্রামটা যেভাবে বানিয়েছি, ডিভাইস(Device)টিকে এইভাবে বসালে ইঙ্গিত করবে
যে এটি উল্লম্বভাবে উপরের দিকে মুখ করে আছে।

(Refer Slide Time: 35:27)



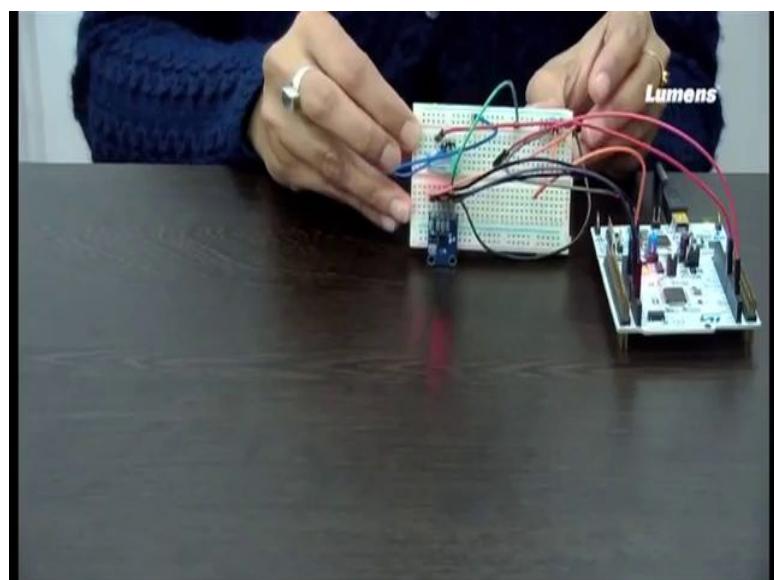
এবং তুমি যদি এটাকে এরকম করো, তার মানে দাঁড়াবে এটি উলমুভাবে নিচের দিকে মুখ করে
দাঁড়িয়ে আছে।

(Refer Slide Time: 35:36)



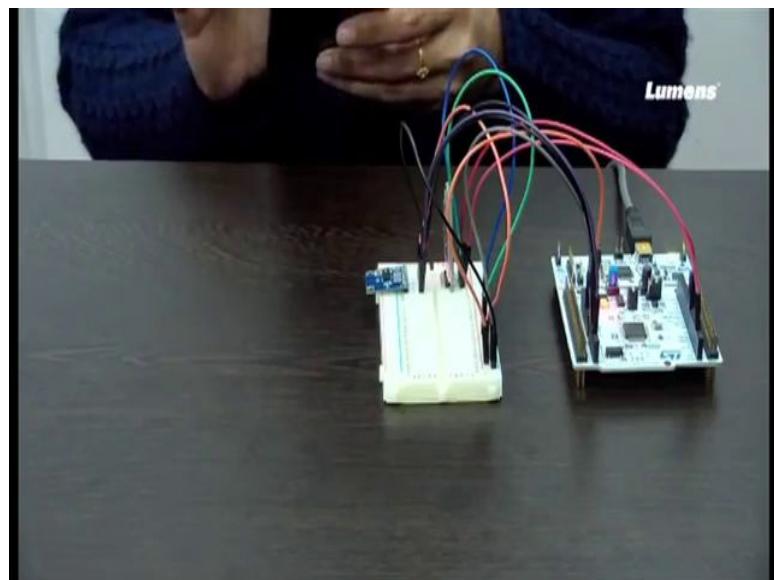
এবং তুমি যদি এটাকে বাঁ দিকে ঘোরাও, তাহলে বুঝতে হবে যে এটি সমান্তরালভাবে বামদিকে
আছে।

(Refer Slide Time: 35:46)



এবং তুমি যখন এরকম করবে তখন এটি সমান্তরালভাবে ডান দিকে আছে।

(Refer Slide Time: 36:08)

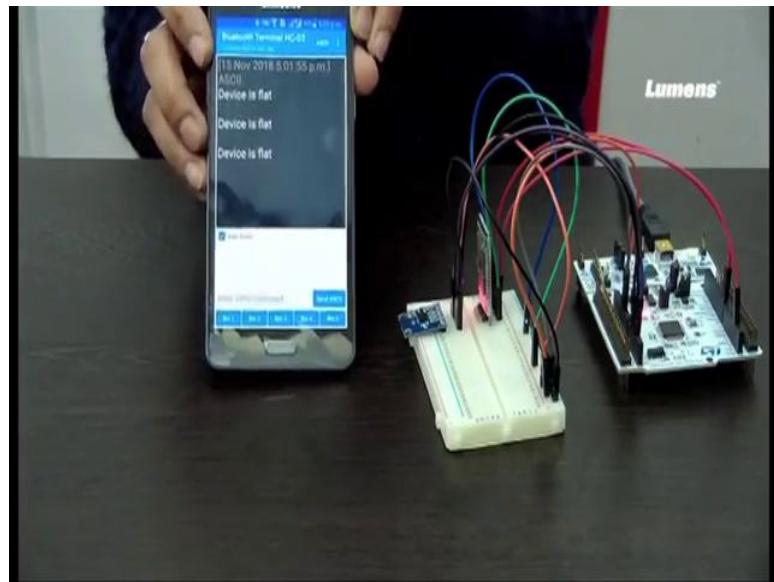


তো আমাকে প্রথমে প্রয়োজনীয় যোগাযোগটা করতে দাও। এইটা হচ্ছে ব্লুটুথ(Bluetooth)-এর সাথে যোগাযোগ, আমরা আগেই দেখেছি। এবং এই যোগাযোগটা আমি অ্যাক্সেলেরোমিটার(Accelerometer)-এর সাথে করেছি।

এবং আমি বলেছিলাম যে কোড(code)-টির দুটি সংস্করণ আছে; একটি কোডের ক্ষেত্রে অনবরত এটি পাঠিয়ে যাবে যে ডিভাইস(Device)-টির অবস্থানগত দিকটি কি।

এর পরের প্রোগ্রাম(Program)টিতে যখন অবস্থানগত দিকটি পরিবর্তন হবে, তখনই একমাত্র এটি ব্লুটুথ মডিউল(Bluetooth Module)-টির মাধ্যমে ডাটা(Data) পাঠাবে। এটি এখনো মিটমিট করে জ্বলছে, এবং এবার আমি যোগাযোগ স্থাপনের চেষ্টা করব। এবং তোমরা দেখতে পাচ্ছ যে এটা স্থির হয়ে গেছে।

(Refer Slide Time: 37:24)

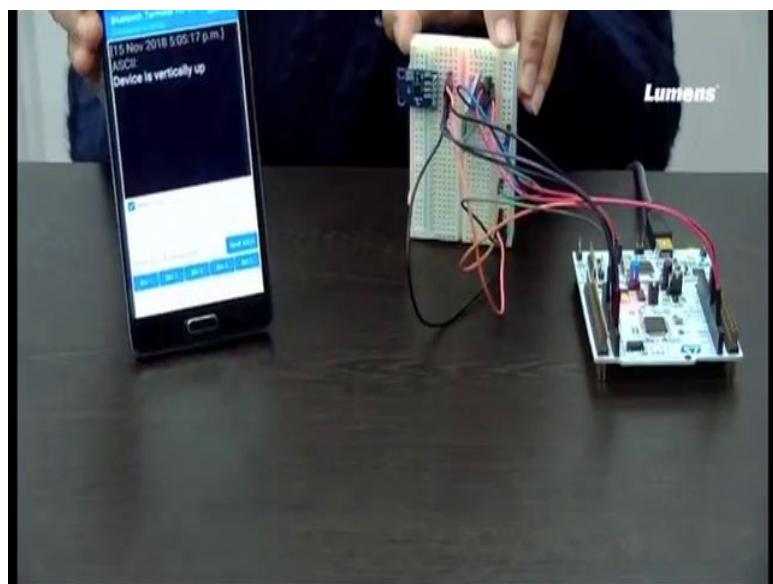


এবং দেখো আমি কি বার্তা পাচ্ছি, "the device is flat", তার মানে এটি চিৎ হয়ে আছে। এবার এই ডিভাইস(Device)-টিকে আমি উল্লম্বভাবে উপরের দিকে মুখ করিয়ে দেব, তো এটা এখন উল্লম্বভাবে উপরের দিকে মুখ করে আছে। এখন, আমি এই ডিভাইসটিকে উল্লম্বভাবে নিচের দিকে মুখ করে রাখবো, তো এখন এটা দেখাচ্ছে "device is vertically down"। এবং এখন ডিভাইসটি চিৎ হয়ে আছে, প্রতি 2 সেকেন্ড অন্তর এটিই সেই তথ্য পাঠাচ্ছে। এখন, আবার আমি এটিকে উল্লম্বভাবে উপরের দিকে মুখ করে দাঁড় করিয়ে দিয়েছি, সে পরিবর্তনটি টার্মিনালে দেখা যাচ্ছে। এবং এরপর আমি এটাকে সমান্তরালভাবে বামদিকে রাখবো। এবং এখন আমি এটাকে সমান্তরালভাবে ডানদিকে রাখবো, তোমরা দেখতে পাচ্ছ যেটা এটা এখন সমান্তরালভাবে ডান দিকে রয়েছে।

তো, x,y,z অক্ষগুলিকে আদর্শ হিসেবে ধরে এই কয়েকটি অবস্থানগত দিক পরিবর্তন করেছি। আমরা ইতিমধ্যেই দেখেছি, যে x,y,z অক্ষের মান গুলি কিরকম আসছে। এবং এখানে অ্যাক্সেলেরোমিটার- টিকে কিভাবে বসিয়েছি তার ওপর নির্ভর করে আমরা কোড লিখেছি।

তো, এই হচ্ছে কিছু অবস্থানগত দিক, যেগুলি আমাদেরকে হিসেবের মধ্যে রাখতে হবে যখন আমরা এই কাজটি করব। এখন, আমি আবার যোগাযোগ বিচ্ছিন্ন করে দেবো। এবং তোমরা দেখতে পাচ্ছ যে এটি আবার মিটমিট করে জ্বলতে শুরু করেছে। এবার, আমি আর একটা কোড ব্যবহার করব, যেখানে খালি অবস্থানগত দিকটি পরিবর্তন হলেই তবে এটি কিছু পাঠাবে এবং বার্তাটি পরিবর্তিত হবে, না হলে এটি কিছু পাঠাবে না।

(Refer Slide Time: 40:42)



এখন, আমি ডিভাইসটিকে কিছু করিনি। এখন ধরা যাক আমি ডিভাইসটিকে উল্লম্বভাবে দাঁড়ান্ত করিয়ে দিলাম। যদি আগের পরীক্ষাটির কথা মনে করো তাহলে দেখবে যে তখন এটি দুই সেকেন্ড ধরে অনবরত দেখিয়ে যাচ্ছিল যে ডিভাইসটি উল্লম্বভাবে রয়েছে, কিন্তু এখন আমি সেটা করছি না, আমি তখনই দেখাবো কিছু যখন কোনো পরিবর্তন ঘটবে। এবার দেখো, ডিভাইসটি যে চিএ হয়ে গেছে, সেই বার্তাটি পাঠানো হয়ে গেছে।

তো, যখনই ডিভাইসটির অবস্থানগত দিক-এর পরিবর্তন হচ্ছে, তখনই একমাত্র লেখাটি পরিবর্তন হচ্ছে, নচেৎ নয়। তো, আমরা অনবরত ভাটা পাঠিয়ে যাচ্ছি না, আমরা কিছু পাঠাচ্ছি তখনই যখন এই নির্দিষ্ট ডিভাইসটির অবস্থানগত দিকে কিছু পরিবর্তন ঘটছে। এখন,

এটি সমান্তরালভাবে ডান দিকে রয়েছে, এখন এটি উপরের দিকে মুখ করে রয়েছে। এবং এটি কোনো বার্তা পাঠাবে না যতক্ষণ না ডিভাইসটির অবস্থানগত দিকে কিছু পরিবর্তন আসছে।

এর জন্য কোড আমরা ইতিপূর্বেই আলোচনা করে ফেলেছি, আমি এর কাজ করাটাও এই এক্সুনি দেখালাম। এই অ্যাক্সেলেরোমিটার(Accelerometer)- টির বিভিন্ন প্রয়োগ রয়েছে, আমি আগে তা আলোচনা করেছি। যেমন তুমি হিসাব করতে পারো যে একটি মানুষ দিনে কত পা হেঁটেছেন। তো আমরা, ইতিমধ্যেই দেখেছি যে যখন আমরা এক্সেলেরোমিটার টিকে বিভিন্নভাবে বিভিন্ন দিকে ঘোরাচ্ছিলাম , x,y,z- এর মান গুলি পরিবর্তন হচ্ছিল, এবং যখন আমরা বসে থাকছি বা খুব আস্তে আস্তে চলাফেরা করছি, তখন মান গুলি কিভাবে পরিবর্তন হচ্ছে।

তোমরা আসলে এই সমস্ত কিছুকে হিসেবের মধ্যে রাখতে পারো মানুষের দৈনিক কার্যকলাপ পর্যবেক্ষণ করার সময়। এবং এছাড়াও আরো অনেক কার্যকারিতাও আছে।

ধন্যবাদ।

GLOSSARY

English	Transliteration	Translation
Data	ডেটা	তথ্য
Print	প্রিন্ট	কোন তথ্যকে কম্পিউটারের স্ক্রিনে দেখানো
Terminal	টার্মিনাল	একটি বিশেষ ধরনের সফটওয়্যার যেখানে কমান্ড লিখে লিখে কাজ করতে হয়
Program	প্রোগ্রাম	নির্দেশ সমূহ যা একটি কম্পিউটার ব্যবহার করে

		নির্দিষ্ট কাজ করার জন্য
Bluetooth	ব্লুটুথ	বিডিএল বৈদ্যুতিক যন্ত্রপাতির মধ্যে তারবিহীন যোগাযোগের একপ্রকার আদর্শ।
LED	এলইডি	একটি অর্ধপরিবাহী যার মধ্য দিয়ে তড়িৎ প্রবাহিত হলে তা রশ্মি উৎপন্ন করে।
Scan	স্ক্যান	খোঁজা
Bluetooth Terminal	ব্লুটুথ টার্মিনাল	এক প্রকার সফটওয়্যার যা দুটি ব্লুটুথ উপযুক্ত ডিভাইসের মধ্যে যোগাযোগ স্থাপনে ব্যবহৃত হয়
App	অ্যাপ	মোবাইলে যে সফটওয়্যার ব্যবহৃত হয়
Variable	জ্যারিয়েবেল	চল
Analog Port	এনালগ পোর্ট	একটি ইনপুট অথবা আউটপুট পোর্ট যা এনালগ সিগনাল- এর সাথে সম্পর্কিত
module	মডিউল	যন্ত্রাংশ
interfacing	ইন্টারফেসিং	যে পদ্ধতিতে মাইক্রোকন্ট্রোলার- কে বিডিএল পেরিফেরাল ডিভাইস এর সাথে যুক্ত করা হয়।
device	ডিভাইস	বৈদ্যুতিন সরঞ্জাম
Code	কোড	নির্দেশ সমূহ যা একটি

		কম্পিউটার ব্যবহার করে নির্দিষ্ট কাজ করার জন্য

Embedded System Design with ARM
Prof. Kamalika Datta
Department of Computer Science and Engineering
National Institute of Technology, Meghalaya

Lecture – 41
Experiment with Gas Sensor

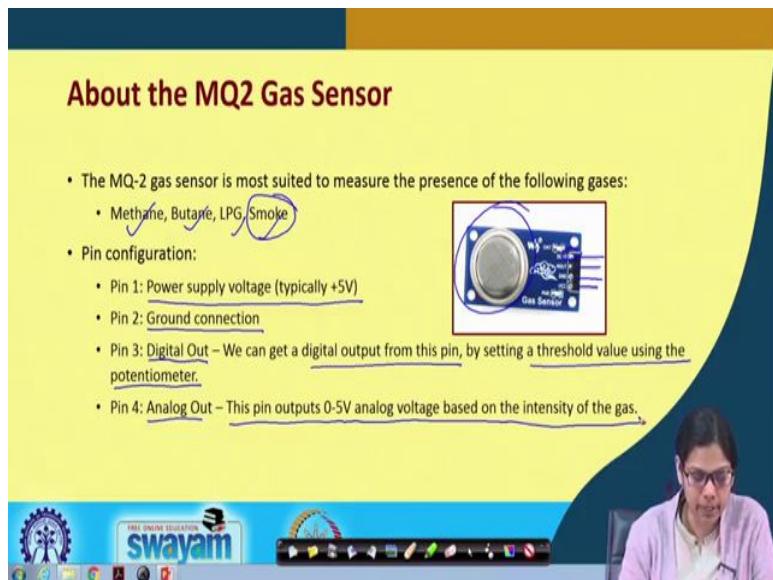
আজকের আলোচনায় স্বাগত। এই আলোচনায়, আমরা তোমাদেরকে গ্যাস সেন্সর(gas sensor)-এর ওপর একটি পরীক্ষা করে দেখাবো। এই পরীক্ষায়, আমরা সেন্সরের (Sensor) মানগুলি পড়বো এবং তার ওপর নির্ভর করে এলার্ম (Alarm)ব্যবস্থাটি সক্রিয় হয়ে উঠবে।

(Refer Slide Time: 01:05)



প্রথমত, আমি দেখাবে কিভাবে আমি MQ2 গ্যাস সেন্সরের(Gas Sensor) সাথে STM32 বোর্ড(Board)- টাকে যুক্ত করছি এবং তারপর পরীক্ষাটি প্রদর্শন করব।

(Refer Slide Time: 01:22)



About the MQ2 Gas Sensor

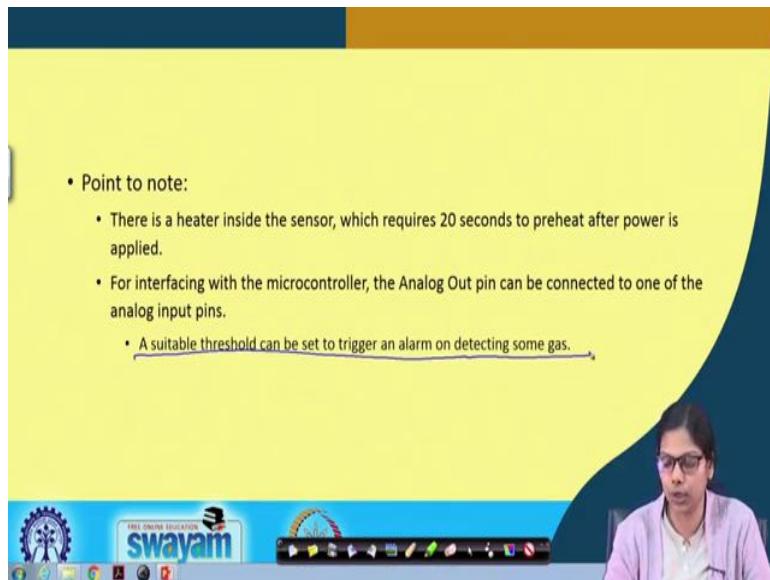
- The MQ-2 gas sensor is most suited to measure the presence of the following gases:
 - Methane, Butane, LPG, Smoke
- Pin configuration:
 - Pin 1: Power supply voltage (typically +5V)
 - Pin 2: Ground connection
 - Pin 3: Digital Out – We can get a digital output from this pin, by setting a threshold value using the potentiometer.
 - Pin 4: Analog Out – This pin outputs 0-5V analog voltage based on the intensity of the gas.

প্রথমত, MQ2 গ্যাস সেন্সর(Gas Sensor)- টি সম্পর্কে বলি, এটা কিছুটা এই রকম দেখতে। তুমি দেখছো যে এখানে আছে একটা Vcc, GND, এনালগ আউট(Analog Out) এবং একটি ডিজিটাল আউট(Digital out)। MQ2 গ্যাস সেন্সর(Gas Sensor) নিম্নলিখিত গ্যাসগুলির নির্ণয়ে উপযুক্ত। এটি মিথেন(Methane), বিউটেন(Butane), এলপিজি(LPG) অথবা ধোঁয়া নির্ণয় করতে পারে।

পিনগুলির রূপরেখাটি এইরকম। প্রথম পিনটি হচ্ছে তড়িৎ উৎস ভোল্টেজ Vcc- র জন্য যেটিতে সাধারণত 2 ভোল্ট(Volt) আছে, তারপর হচ্ছে GND সংযোগ। তারপর আছে ডিজিটাল আউট(Digital out) পিন। এই পিনে, পোটেনশিওমিটার(Potentiometer)- এর সাহায্যে প্রান্তিক মানটি ঠিক করে দিলে তার সাহায্যে আমরা একটি ডিজিটাল(Digital) মান পেতে পারি। আমরা ইতিমধ্যেই এলসিডি (LCD) ডিসপ্লে(Display)-তে পোটেনশিওমিটার(Potentiometer)- এর ব্যবহার দেখেছি। এখানেও যদি তুমি এই ডিজিটাল আউট (Digital Out) মানটিকে ব্যবহার করতে চাও, তাহলে আমাদেরকে পোটেনশিওমিটার(Potentiometer) ব্যবহার করে এই প্রান্তিক মানটিকে ঠিকঠাক করতে হবে। এনালগ পোর্ট(Analog Port)- এর সাথে যুক্ত যে এনালগ

আউট(Analog Out)- পিনটি আছে, তার আউটপুট(Output) নির্ভর করে গ্যাসের তীব্রতার উপর এবং তা ০ থেকে ৫ ভোল্ট এনালগ ভোল্টেজ(Analog Voltage)- এর মধ্যে ঘোরাফেরা করে।

(Refer Slide Time: 03:35)

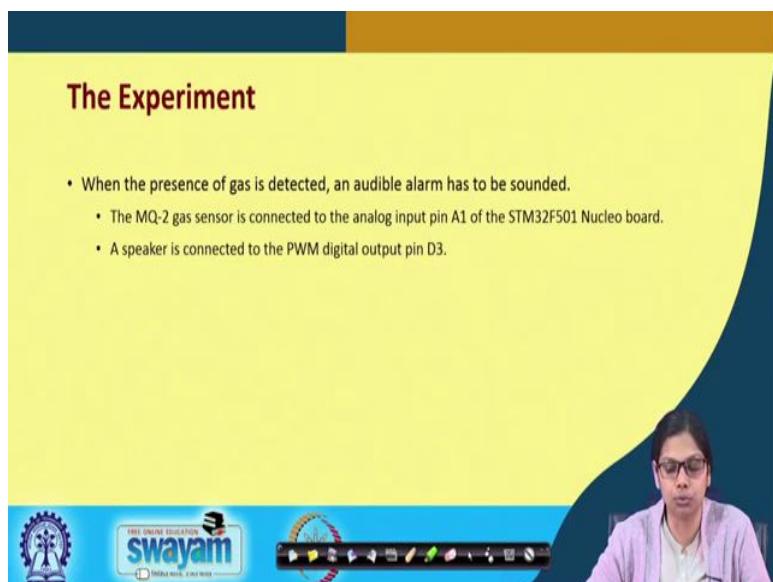


কিছু জিনিস জানা দরকার যে MQ2 গ্যাস সেন্সর(Sensor)- টির মধ্যে একটি হিটার(heater) আছে। এবং তড়িৎ সংযোগ হওয়ার পর এটির কিছুটা সময় লাগে গরম হতে। তো, যখন আমরা সিস্টেম(System)-টিকে চালাই, সেটির চালু হতে বা সেন্সর(Sensor)-টিকে গরম করতে কিছুটা সময় লাগতে পারে। মাইক্রোকন্ট্রোলার(microcontroller)- এর সাথে যুক্ত হওয়ার জন্য এনালগ আউট(Analog out) পিন- টি যেকোনো একটি এনালগ ইনপুট(Analog Input) পিনের সাথে যুক্ত হয়, তো সেটা আমরা ইতিমধ্যেই বেশিরভাগ পরীক্ষায় করেছি। সেই একইভাবে আমরা এটাও করব।

এবং একটা উপযুক্ত প্রাণ্তিক মান ঠিক করা যেতে পারে যাতে কোন গ্যাস নির্ণয় করা মাত্রাই এলার্ম(Alarm)-টি সক্রিয় হয়ে যায়। এখন এটা এমন একটা জিনিস যেটা তোমাকে ভাবতে হবে, যে কোন প্রাণ্তিক মান অতিক্রম করলে তুমি চাও তোমার সিস্টেম(System)-টি সক্রিয় হয়ে উঠবে, হতে পারে সেটা কোন এলার্ম সিস্টেম(Alarm System) অথবা তুমি একটা জিএসএম(GSM)

ব্যবহার করতে পারো যেটা দমকল বাহিনীকে সরাসরি বার্তা পাঠিয়ে দেবে। কিন্তু সাধারণত ধোঁয়ার মাত্রাটি বুঝতেই পারছ তোমাকেই ঠিক করতে হবে, তোমাকে এমন একটা কিছু করতে হবে যে তার ফলে ধোঁয়ার সৃষ্টি হবে এবং তারপর তোমায় কুলটার্ম(CoolTerm)- এর সাহায্যে দেখতে হবে কি মান পাঞ্চ এবং সেইমতো প্রোগ্রাম(Program) লিখতে হবে।

(Refer Slide Time: 05:38)

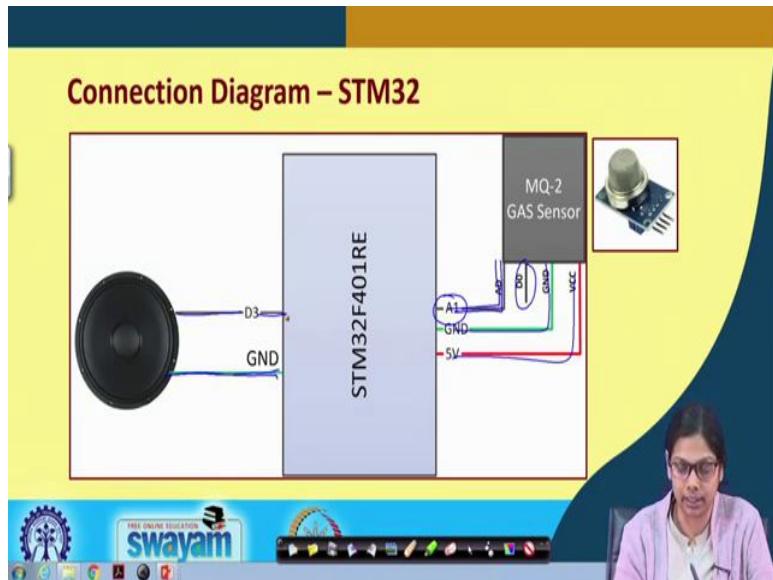


The Experiment

- When the presence of gas is detected, an audible alarm has to be sounded.
- The MQ-2 gas sensor is connected to the analog input pin A1 of the STM32F501 Nucleo board.
- A speaker is connected to the PWM digital output pin D3.

এই পরীক্ষায় আমরা কি করব, না যখনই কোন গ্যাসের উপস্থিতি টের পাওয়া যাবে তখনই এলার্ম(Alarm)-টি সক্রিয় হয়ে যাবে। MQ2 গ্যাস সেন্সর(Gas Sensor)- টি এসটিএম বোর্ড(STM Board)- এর এনালগ ইনপুট(Analog Input) পিন A1- এর সাথে যুক্ত হবে এবং PWM আউটপুট(Output) পিন D3-র সাথে স্পিকার(Speaker)টি যুক্ত হবে।

(Refer Slide Time: 06:13)



এইটা হচ্ছে সংযোগ চিত্রটি যেটা কিনা খুবই সহজ সরল। ডিজিটাল(Digital) পিনটি নিয়ে
আমাদের মাথা ব্যাথা নেই, আমরা এনালগ আউট(Analog out)-টাকে A1- এর সাথে যুক্ত করছি
। এবং আমরা D3 পিনটি যুক্ত করবো স্পিকার(Speaker)-টির সাথে, যেটি এখানে এলার্ম
সিস্টেম(Alarm System) হিসেবে কাজ করছে এবং সেটি কখন সক্রিয় হয়ে উঠবে তা নির্ভর করছে
ঘরের মধ্যে কতটা পরিমাণে গ্যাস(Gas) আছে। তো, এইটাই হচ্ছে প্রয়োজনীয় বর্তনী চিত্রটি।

(Refer Slide Time: 07:00)

The code is written in Mbed C and defines a main loop and a fire() function. The main loop reads the analog input from pin A1, prints the value, and checks if it is greater than 0.30. If it is, it calls the fire() function, waits for 1 second, and then sets the PWM period to 10 microseconds. The fire() function generates a square wave on pin D3 with a period of 3000 microseconds and a pulse width of 1550 microseconds, followed by a period of 3000 microseconds and a pulse width of 1550 microseconds, repeating this sequence 5 times.

```
#include "mbed.h"
PwmOut mypwm(D3);
AnalogIn G(A1);
int i=0;

void fire(){
    i=0;
    while (i<5) {
        mypwm.period_us (3000);
        mypwm.pulsewidth_us (1550);
        mypwm.period_us (3000);
        mypwm.pulsewidth_us (1550);
        wait (0.5);
        mypwm.period_us (2200);
        mypwm.pulsewidth_us (100);
        wait (0.5);
        i++;
    }
}

int main()
{
    float x;
    while(1) {
        x = G.read();
        printf ("%4.2f\n",x);
        if (x > 0.30)
        {
            fire();
            wait(1);
            mypwm.period_us(10);
            mypwm.pulsewidth_us(5);
        }
    }
}
```

এবার Mbed C কোড(Code)-টা দেখা যাক | mypwm.period_us হচ্ছে 3000 মাইক্রোসেকেন্ড(microsecond) , এবং তরঙ্গ প্রস্তু হচ্ছে 1550 , এবং কিছুক্ষণের জন্য অপেক্ষা করছি এবং তারপর আবার আমরা এটাকে 2200 করছি এবং তরঙ্গ প্রস্তুকে 100 রাখছি। তো, বল্তত, তুমি যদি দেখো এই void fire-টাকে, fire হচ্ছে একটা ফাংশন(Function) যেটা আমরা নিজেরা লিখেছি এবং স্পিকার(Speaker) থেকে সাইরেনের(Siren) মত শব্দ বের করার জন্য আমরা আগে যা করছিলাম তার সঙ্গে এই ফাংশন(Function)- টির খুবই মিল আছে।

এটার আগে ,এই জিনিসগুলো আবার খুবই সোজাসাপটা। এটা হচ্ছে PWMOut mypwm যেটা স্পিকার(Speaker) -এর সাথে যুক্ত হওয়ার জন্য রয়েছে এবং AnalogIn G (A1) রয়েছে গ্যাস সেন্সর-টিকে এনালগ পোর্ট(Analog Port) A1- এর সাথে যুক্ত করার জন্য। এই হচ্ছে সংযোগের ব্যাপারে যাবতীয় কিছু এবং কোড(Code)- এর এই অংশটি হচ্ছে ফায়ার এলার্ম(Fire Alarm)- এর জন্য। মেন ফাংশন- এর মধ্যে while(1) রয়েছে মানে এটা পুনরায় চলবে।

`x = G.read()` গ্যাস সেন্সর(Gas Sensor)-এর মানটি পড়ছে। যখনই এই মানটি 0.3 - এর থেকে বেশি হচ্ছে, তখনই আমরা এলার্ম সিস্টেম(Alarm System)- টাকে সক্রিয় করছি। আমরা এক সেকেন্ডের(Second) জন্য অপেক্ষা করছি এবং আবার একটা সময়কাল বেঁধে দিচ্ছি এবং এটা চলতেই থাকবে। এটা যদি আবারও 0.3- এর থেকে বেশি হয়, তাহলে এলার্ম সিস্টেম(Alarm System)- টি সক্রিয় হয়ে যাবে। তো, এটা হচ্ছে উদাহরণস্বরূপ একটি কোড(Code), যেটা আমরা লিখেছি যাতে গ্যাস সেন্সর(Gas Sensor)- টিকে এলার্ম সিস্টেম(Alarm System) তথা স্পিকার(Speaker)- এর সাথে ব্যবহার করতে পারি।

তো আমরা এই আলোচনার শেষে চলে এসেছি। এবং যে সমস্ত পরীক্ষা নিরীক্ষা গুলি তোমাদেরকে আমরা এই কোর্সে(Course) দেখিয়েছি সেগুলি নিয়মমত এখানেই শেষ হয়ে যাচ্ছে। তো, এখন আমি তোমাদের দেখাবো যে গ্যাস সেন্সর(Gas Sensor)-টিকে কিভাবে বাকি জিনিসগুলোর সাথে যুক্ত করবো।

এই পরীক্ষাটিতে, আমি এসটিএম বোর্ড(STM Board)- এর সাথে একটা স্মোক সেন্সর (Smoke Sensor) যুক্ত করবো । এবং স্মোক সেন্সর(Smoke Sensor)-টি কি করবে,না যখনই নিজের পারিপার্শ্বিক ধোঁয়া টের পাবে , তখনই এটি এলার্ম(Alarm) -টিকে সক্রিয় করে দেবে। কিন্তু তুমি এরকমও করতে পারো; যখনই ধোঁয়ার টের পাওয়া গেল তখনই দমকল বাহিনীর কাছে একটা এসএমএস(SMS) চলে যাবে । এক্ষেত্রে, আমাদের ব্যবস্থাটিতে গ্যাস সেন্সর(Gas Sensor) ছাড়াও একটা জিএসএম বোর্ড(GSM Board) লাগবে ।

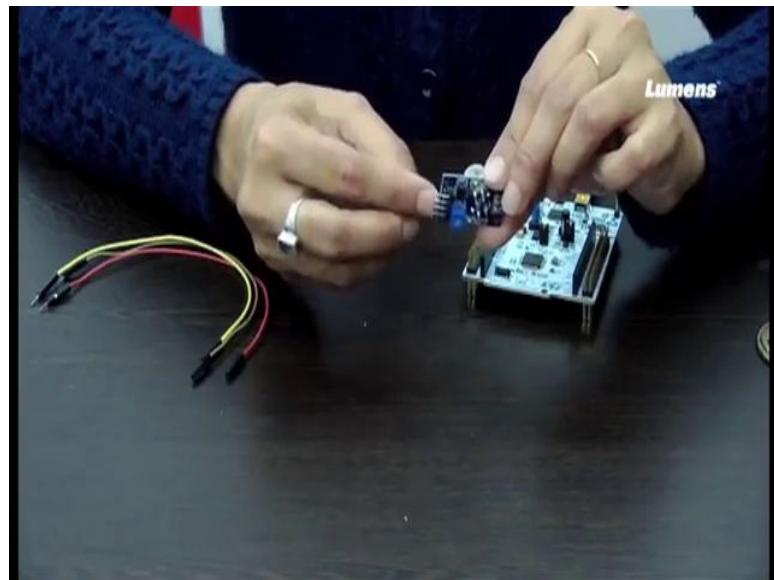
তো, অনেক রকম জিনিসই সন্তুষ্ট, কিন্তু এখানে আমরা যেটা করছি সেটা হচ্ছে এই স্মোক সেন্সর(Smoke Sensor)- টিকে বাকি জিনিসগুলির সাথে কাজ করানো । এবং যখনই ঘরের মধ্যে ধোঁয়া থাকবে, তখনই স্পিকার(Speaker)-টি একটি সাইরেন(Siren) মত শব্দ বের করবে । দেখা যাক কিভাবে আমি এই গ্যাস সেন্সর(Gas Sensor)-টিকে স্পিকার(Speaker) ও এসটিএম বোর্ড(STM Board)- এর সাথে যুক্ত করতে পারি ।

(Refer Slide Time: 12:25)



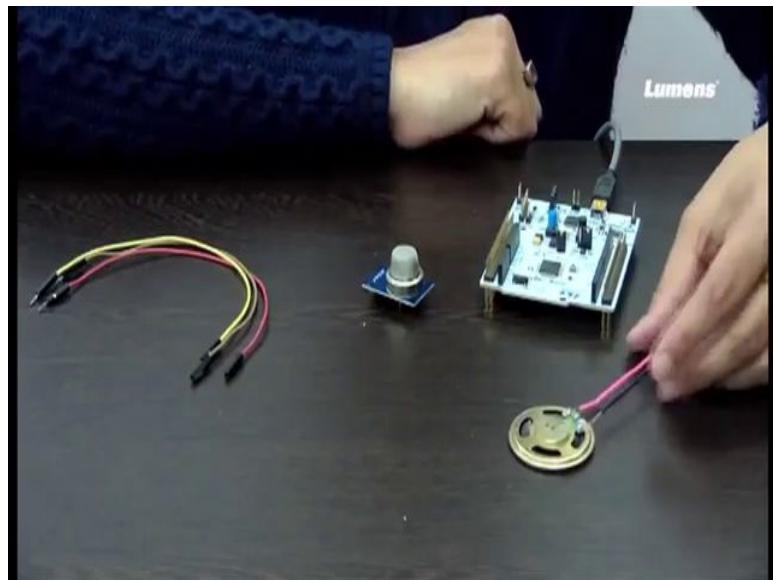
তো, এইটা হচ্ছে গ্যাস সেন্সর(Gas Sensor)- টা ।

(Refer Slide Time: 12:34)



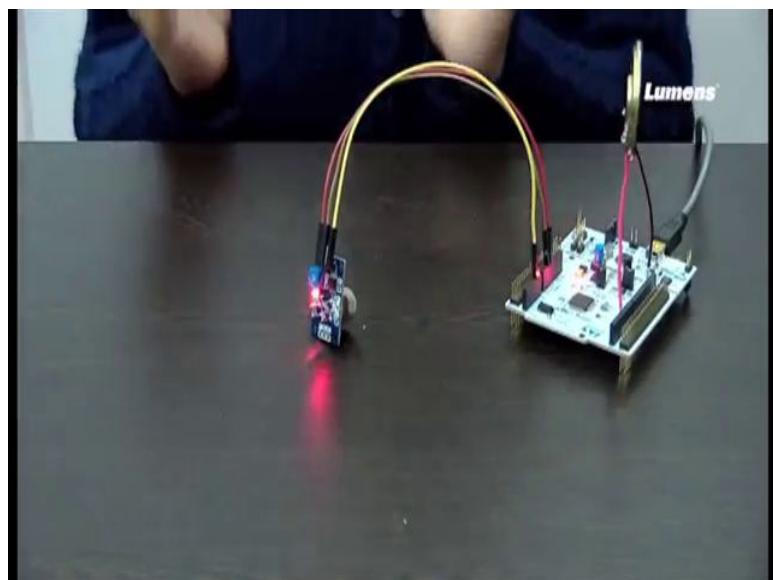
এই গ্যাস সেন্সর(Gas Sensor)-টাই আমরা ব্যবহার করব। তোমরা দেখতে পাচ্ছ যে এখানে চারটে পিন রয়েছে যারা কিনা সংযুক্ত। তো, এই পিনগুলির কি কাজ? একটা হচ্ছে AD, অন্যটি হচ্ছে DD, অপর আরেকটি হচ্ছে GND, এবং অন্যটি হচ্ছে Vcc। আমি GND এবং Vcc যুক্ত করবো, এবং আমি এনালগ আউটপুট(Analog output) টা ব্যবহার করব। তো, ডিজিটাল আউটপুট(Digital Output) নিয়ে আমার মাথাব্যথা নেই। আমি এনালগ আউটপুট(Analog output)-টা নেব এবং এটাকে এসটি-এম বোর্ড(STM Board) A1 পিনের সাথে যুক্ত করবো।

(Refer Slide Time: 13:23)



স্পিকার(Speaker)- টির ক্ষেত্রে তার একটি প্রান্ত আমি গ্রাউন্ড- এর সাথে যোগ করব এবং
আরেকটি প্রান্ত PWM পোর্ট(Port)-এর সাথে যুক্ত করবো |এই ক্ষেত্রে আমরা D3
পোর্ট(Port)-টিকে. PWM পোর্ট(Port) হিসেবে ব্যবহার করেছি।

(Refer Slide Time: 13:41)



তো, প্রথমটা হচ্ছে Vcc | তার পরেরটা হচ্ছে GND , এবং শেষেরটা হচ্ছে এনালগ
আউটপুট(Analog Output) যেটা এসটিএম(STM)- এর A1 পিনের সাথে যুক্ত করবো |স্মোক

সেন্সর(sensor)-টিকে যুক্ত করা নিতান্তই সহজ । এবার এই স্পিকার(Speaker)-টিকে D3- এর সাথে যুক্ত করবো এবং এই দিক থেকে চতুর্থপিনটি GND- র সাথে যুক্ত করবো । তো, যে সংযোগটা আমরা বানিয়েছি এই হচ্ছে তার সমস্ত কিছু ঠিক আছে । কোড(Code)- টা তো তোমাদের সাথে ইতিমধ্যেই আলোচনা করেছি । এবার কোড(Code)- টাকে ভরে দেব ।

এবার, একটা দেশলাই কাঠি ব্যবহার করব । আমি এই দেশলাই কাঠিটাকে জ্বালিয়ে নিভিয়ে দেব এবং তার ধোঁয়াটা ব্যবহার করব । যখনই এটি ধোঁয়া টের পাবে, সাইরেন(Siren)- টি কিছুক্ষণের জন্য সক্রিয় থাকবে । আমি আবার এটা একবার পুনরায় করব । ভালোমতো ধোঁয়া থাকলে, তবেই এটি টের পাবে ।

সুরক্ষার কারণে, তুমি সবসময়ই গৃহ স্বয়ংক্রিয় ব্যবস্থায়, এই স্মোক সেন্সর(Smoke Sensor)-টিকে একসাথে ব্যবহার করতে পারো । তো, এই হল স্মোক সেন্সর(Smoke Sensor)- এর উপর যাবতীয় পরীক্ষা-নিরীক্ষা । আমরা এগিয়ে যাব এবং দেখব এরপর আর কি কি পরীক্ষা-নিরীক্ষা করা যেতে পারে ।

ধন্যবাদ ।

GLOSSARY

English	Transliteration	Translation
Gas Sensor	গ্যাস সেন্সর	বিশিষ্ট সেন্সর যা নির্দিষ্ট কিছু গ্যাসের উপস্থিতি নির্ণয় করতে পারে
Smoke Sensor	স্মোক সেন্সর	বিশিষ্ট সেন্সর যা ধোঁয়ার উপস্থিতি নির্ণয় করতে পারে
Alarm	এলার্ম	বিপদ সংকেত

Potentiometer	পোটেনশিওমিটার	এক প্রকার বৈদ্যুতিক মাপ মাপকারি যন্ত্র যা দুটি সেলের তড়িৎচালক বল তুলনা করতে পারে।
Heater	হিটার	একটি যন্ত্র যা তড়িৎ শক্তি থেকে উষ্ণতা সৃষ্টি করতে পারে।
Program	প্রোগ্রাম	কম্পিউটারকে কাজ করানোর জন্য নির্দেশ সমূহ।
Speaker	স্পিকার	একটি যন্ত্র যা তড়িৎ শক্তি থেকে শব্দ-তরঙ্গের সৃষ্টি করতে পারে
PWM	পি ডিস্ট্রিউ এম	যে পদ্ধতি দাঁড়া একটি বৈদ্যুতিক সিগনালের গড় বৈদ্যুতিক ক্ষমতা কমিয়ে দেওয়া যায় সেটিকে আলাদা আলাদা অংশে ভাগ করে
microsecond	মাইক্রোসেকেন্ড	এক মিলি সেকেন্ডের 1000 তম ভগ্নাংশ
function	ফাংশন	প্রোগ্রামের একটি অংশ যা একটি নির্দিষ্ট কাজ করে।
siren	সাইরেন	একটি যন্ত্র যা প্রচল জোরে শব্দ বের করতে পারে।
analog output	এনালগ আউটপুট	যে আউটপুট টি এনালগ প্রকারের হয়
digital output	ডিজিটাল আউটপুট	যে আউটপুট টি ডিজিটাল প্রকারের হয়

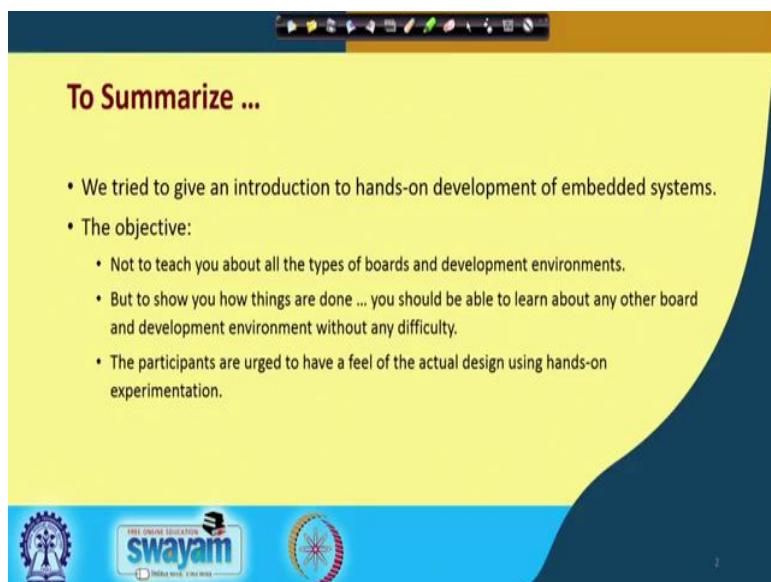
port	পোর্ট	একজোড়া টার্মিনাল যা একটি ইলেক্ট্রিক্যাল নেটওয়ার্ককে কোন বাহ্যিক বর্তনীর সঙ্গে যুক্ত করে।

Embedded System Design with ARM
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 42
Summarization of the Course

তো ,আমরা এমবেডেড সিস্টেম ডিজাইন উইথ এআরএম(**Embedded System Design with ARM**) নামক কোর্স(Course)-টির শেষে চলে এসেছি । আমরা আশা করছি যে তোমরা এই কোর্স(Course)-টা উপভোগ করেছো । তবে, তোমাদেরকে বিদ্যায় জ্ঞানান্তর আগে এই কোর্সটির ব্যাপারে কিছু কথা বলতে চাই ।

(Refer Slide Time: 00:41)



এই কোর্স(Course), আমরা চেষ্টা করেছি যাতে তোমাদেরকে এমবেডেড সিস্টেম ডিজাইনের (**embedded system design**)ওপর হাতে -কলমে কাজ করা শেখানো যায় । আমরা দেখেছি যে বিষয়টির ব্যাপারে তাত্ত্বিক আলোচনা আমরা খুব কমই করেছি । অন্যান্য অনেক কোর্স আছে , ভালো বই আছে এই বিষয়টির ওপর, যেগুলিতে এমবেডেড সিস্টেম ডিজাইনের(**embedded system design**) অন্তর্নির্দিত তত্ত্বের উপর আলোচনা আছে। কিভাবে বিভিন্ন বিষয়ের মধ্যে

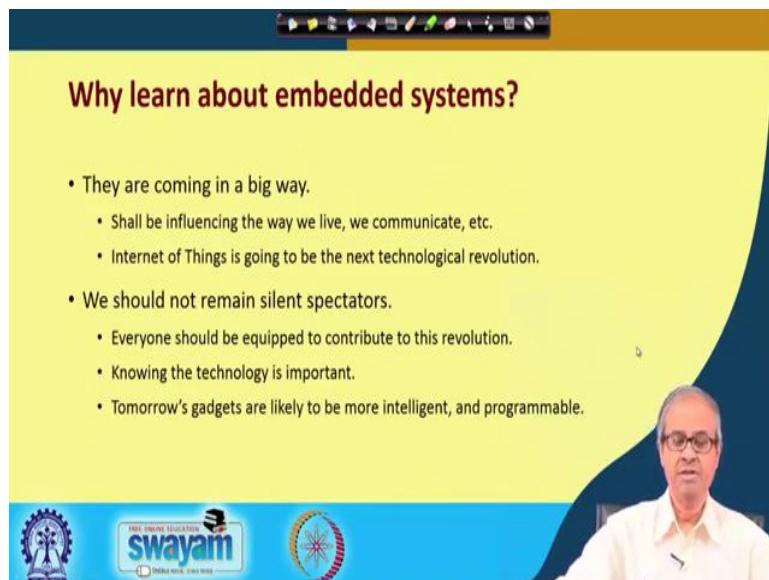
ভারসাম্য বজায় রাখতে হবে, কিভাবে সফটওয়্যার ডিজাইন(Software Design) করতে হবে, কিভাবে হার্ডওয়ার ডিজাইন(Hardware Design) করতে হবে, কিভাবে হার্ডওয়ার- সফটওয়্যার কোড ডিজাইন(Hardware-Software Code Design) নামক জিনিসটি তোমরা পালন করবে ইত্যাদি কিন্তু আমরা চেষ্টা করেছি যে যাতে গঠনমূলক ব্যাপারে হাতেনাতে কাজ করবার ব্যাপারে তোমাদের কিছু ধারণা দেওয়া যায়।

এই কোর্স(Course)-টির মূল লক্ষ্য ছিল না যে তোমাদেরকে বাজারে যত ধরনের বোর্ড পাওয়া যায় সেই সমস্ত কিছু শেখাবে, না আমরা তোমাদেরকে শেখাতে চেয়েছিলাম যত ধরনের সফটওয়্যার ডেভেলপমেন্ট সিস্টেম (Software Development System) বাজারে পাওয়া যায় তার ব্যাপারে শেখাতে, আমরা শুধুই কিছু উদাহরণ নিয়েছি। আমরা STM32 বোর্ড(Board) ব্যাপারে কথা বলেছি, আরডুইনো উনো(Arduino Uno)- র ব্যাপারেও কথা বলেছি যেগুলি খুবই জনপ্রিয় ডেভেলপমেন্ট বোর্ড(Development Board)। এবং এইগুলির সাহায্যে আমরা এটাই দেখাতে চেয়েছি যে প্রয়োজনমাফিক বিভিন্ন ডিভাইস(Device)-কে একসঙ্গে কাজে লাগিয়ে একটি কার্যকরী সিস্টেম(System) বানানো খুবই সহজ।

এমবেডেড সিস্টেম ডিজাইন(embedded system design) করা মোটেই কোন শক্ত ব্যাপার নয়, খুবই সহজ। আমরা আশা করছি যে অংশগ্রহণকারীরা আমাদের সাথে ছিল, তাদের কাছে যে সমস্ত বোর্ড আছে তার ওপর হাতে কলমে কাজ করতে মগ্ন ছিল তারা, কারণ হাতে কলমে কাজ না করলে, তুমি সত্যিই এই পদ্ধতিটির সৌন্দর্য এবং সারল্যের গুরুত্ব অনুভব করতে পারবে না। একটা সময় ছিল যখন হার্ডওয়ারের(Hardware) নকশা তৈরি করা খুবই জটিল ছিল, কিন্তু আজকের দিনে লোকজন উন্মুক্ত হার্ডওয়ার(Hardware) নকশার কথা আলোচনা করছে। তোমরা এই মডিউল(Module)-গুলি বাজার থেকে কিনতে পারো, এইগুলিকে বিভিন্ন বোর্ডের (Board)সাথে যুক্ত করা খুবই সহজ ব্যাপার এবং আমি তোমাদের ইতিমধ্যেই তা দেখিয়েছি। এবং আজকের দিনে মেশিন ল্যাঙ্গুয়েজ(Machine Language) অথবা অ্যাসেম্বলি ল্যাঙ্গুয়েজে (Assembly

Language)কোড লেখা আর প্রযোজনীয় নয়, তুমি C- এর মত হাই লেভেল ল্যাঙ্গুয়েজে(High Level Language) লিখতে পারো যা আমরা ইতিপূর্বে দেখেছি।

(Refer Slide Time: 03:17)



এবার প্রশ্ন হচ্ছে যে এমবেডেড সিস্টেম(embedded System)- এর ব্যাপারে শিখব কেন? প্রথম এবং সর্বপ্রধান কারণ হলো যে আমাদের জীবনে এমবেডেড সিস্টেম(embedded system) খুব বড় ভাবে আসতে চলেছে। আমরা ইতিমধ্যেই ইন্টারনেট অফ থিংস(Internet of Things)- এর ব্যাপারে কথা বলেছি, জীবনের প্রতিটি ক্ষেত্রেই এরা দাবানলের মতো ছড়িয়ে পড়ছে, কিছু কিছু ক্ষেত্রে এরা খুব দ্রুত ছড়িয়ে পড়ছে। এই উন্নিসাধন প্রভাবিত করবে কিভাবে আমরা বাঁচবো, কিভাবে দৈনিক জীবনে যোগাযোগ করব।

মূল বক্তব্যটি এই যে এই বিপ্লবের নীরব দর্শক হয়ে আমরা থাকবো না, সেই কারণেই কোর্স(Course)টির মূল লক্ষ্য ছিল যে চেষ্টা করা যাতে এমবেডেড সিস্টেমের(embedded system) উপর চারিদিকে যে কাজ হচ্ছে তার আস্বাদ দেওয়া। যেমন, আজকের দিনে প্রযুক্তি কোন অবস্থায় আছে যার সাহায্যে তুমি এরকম একটা সিস্টেম(System) তৈরি করতে পারো, কারণ আগামীকালের ডিভাইস(device)গুলি আজকের দিনের থেকে অনেকটাই আলাদা হবে। সেগুলি

বেশিরভাগই প্রোগ্রামেবল(programmable) হবে এবং সেই জন্য প্রযুক্তির ব্যাপারে কিছু জানা আবশ্যিক।

(Refer Slide Time: 04:45)

Course Coverage – Just to Recall

- WEEK 1
 - Introduction to embedded systems
 - Introduction to ARM architecture
- WEEK 2
 - Introduction to ARM instruction set
 - A case study with the STM32F401 Nucleo board
- WEEK 3
 - Data conversion techniques (digital and analog)
 - Output devices, sensors and actuators

খুব দ্রুত, আমরা একবার চোখ বুলিয়ে নেব যা যা আমরা পূর্বের সপ্তাহগুলোতে আলোচনা করেছি।

প্রথম সপ্তাহে, আমরা কথা বলেছিলাম এমবেডেড সিস্টেম ডিজাইনের(embedded system design) কিছু ভূমিকাস্বরূপ বিষয়ের উপর, বিভিন্ন বিষয়ের মধ্যে ভারসাম্য বজায় রাখা এবং ARM মাইক্রোকন্ট্রোলার(microcontroller)- এর আর্কিটেকচার(architecture) বিষয়েও কথা বলেছিলাম। দ্বিতীয় সপ্তাহ, আমরা ARM ইনস্ট্রাকশন সেট(Instruction Set)- এর কিছু চিত্তাকর্ষক ব্যাপার নিয়ে আলোচনা করেছিলাম, কিছু উদ্ধাবনা যেগুলি পরবর্তীকালে ব্যবহৃত হয়েছে, এবং ঘটনা পর্যবেক্ষণ হিসেবে আমরা একটা STM32 বোর্ডকে(Board) নিয়েছিলাম। এবং তৃতীয় সপ্তাহে, আমরা কথা বলেছিলাম A/D কনভার্টার(Converter) এবং D/A কনভার্টার(Converter)- এর নকশার বিষয়ে, তাছাড়াও বিভিন্ন আউটপুট ডিভাইস(OUTPUT

Device), সেন্সর(Sensor), একচুয়াটর(Actuator) ইত্যাদি ব্যাপারে কথা বলেছিলাম। তাদের মধ্যে বেশিরভাগ গুলোই আমরা হাতে-কলমে করে দেখিয়েছি।

(Refer Slide Time: 05:49)



চতুর্থ সপ্তাহে, আমরা কথা বলেছি বিভিন্ন মাইক্রোকন্ট্রোলার(microcontroller) ডেভলপমেন্ট বোর্ড(development board) এবং সফটওয়্যার ডেভলপমেন্ট(Software Development) অথবা প্রোগ্রামিং এনভারমেন্ট(programming environment) এর ব্যাপারে। বিশেষভাবে, আমরা দুটো বোর্ডের মধ্যে সংযোগ পদ্ধতি দেখলাম, যথা STM32 এবং আরডুইনো উনো(Arduino uno)-র মধ্যে। এবং বিশেষভাবে আমরা বিশেষ আলোচনা করেছি 7-সেগমেন্ট এলাইডি ডিসপ্লে(7-segment LED display) এবং এলসিডি ডিসপ্লে(LCD Display)-র ব্যাপারে। পঞ্চম সপ্তাহে আমরা তাপমাত্রার সেন্সর(temperature sensor) LM35, লাইট ডিপেন্ডেন্ট রেজিস্টার(Light Dependent resistor), স্পিকার(Speaker) এবং মাইক্রোফোন(Microphone) এই সমস্ত কিছুকে নিয়ে বিভিন্ন পরীক্ষা-নিরীক্ষার কথা বলেছিলাম।

(Refer Slide Time: 06:31)

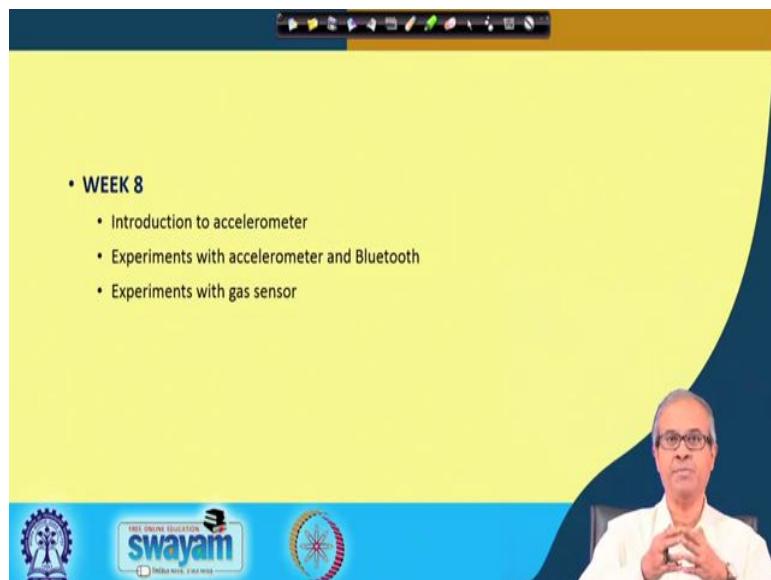
- WEEK 6
 - Introduction to control system design
 - Experiments with relay control
 - Experiments with motor interfacing
- WEEK 7
 - Introduction to IoT systems
 - Experiments on home automation using GSM communication
 - Experiments on simple alarm system based on touch sensing

ষষ্ঠ সপ্তাহে, আমরা খুব সংক্ষিপ্তভাবে দেখলাম যে কন্ট্রোল সিস্টেম(Control System) কিভাবে তৈরি করা হয়, বিভিন্ন ধরনের কন্ট্রোল অ্যালগরিদম(Control Algorithms) যথা আনুপাতিক, সম্পূরক, অন- অফ কন্ট্রোল(On-Off Control) ইত্যাদি। তারপর আমরা দেখলাম রিলে(Relay) ব্যবহার করে কিছু পরীক্ষা-নিরীক্ষা এবং এছাড়াও কিভাবে একটি ডিসি মোটর(DC motor)-কে মাইক্রোকন্ট্রোলার(microcontroller) এর সাথে যুক্ত করা যায়, মোটরের(motor) গতি নির্ণয়ের ক্ষমতা এবং আরো অনেক কিছু দেখলাম। এরইমধ্যে, আমরা এটাও শিখে ফেললাম যে STM32 বোর্ড- এর ইন্টারাপ্ট সিস্টেম(Interrupt System)- টি কিভাবে ব্যবহার করতে হয়। আমরা দেখলাম যে এই ধরনের পরিবেশে ইন্টারাপ্ট(Interrupt) নির্ভর সফটওয়্যার(Software) তৈরি করা খুবই সহজ ব্যাপার।

এবং সপ্তম সপ্তাহে, ইন্টারনেট অফ থিংস(Internet Of Things) এর ব্যাপারে একটা সংক্ষিপ্ত আকারে আলোচনা হয়েছিল, তারপর গৃহ স্বয়ংক্রিয়তার উপর কিছু পরীক্ষা-নিরীক্ষা ছিল। এখানে আমরা বিভিন্ন যোগাযোগ পদ্ধতিও জানলাম, যেমন ধরে জিএসএম(GSM)। এবং আমরা টাচ

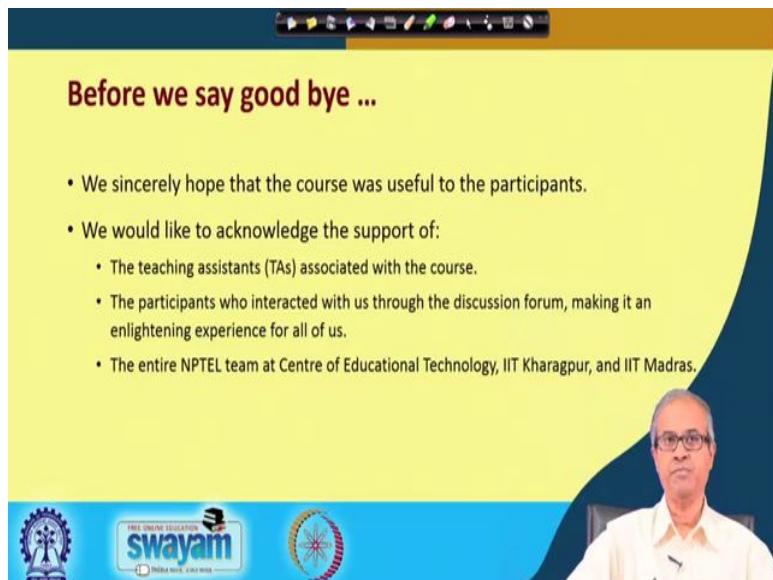
সেন্সর(Touch Sensor) দিয়ে তৈরি একটা সাধারণ এলার্ম সিস্টেম(Alarm System) এবং আরও অনেক কিছু নিয়ে কথা বলেছিলাম।

(Refer Slide Time: 07:41)



এবং শেষ সপ্তাহে, আমরা দেখলাম যে একটা অ্যাক্সেলোরোমিটার(Accelerometer) কিভাবে কাজ করে, এর বিভিন্ন ব্যবহারিক প্রয়োগ কি কি। বিশেষভাবে আমরা কিছু পরীক্ষা-নিরীক্ষা করলাম যেখানে আমরা অ্যাক্সেলোরোমিটার(Accelerometer)- এর সাথে ড্রাইভ প্রযুক্তিকে কাজে লাগালাম, যেখানে আমরা একটা মোবাইল ফোন(Mobile Phone) ব্যবহার করেছিলাম এবং কিছু বার্তা সেই মোবাইল ফোনের(Mobile Phone) স্ক্রিনে দেখা যাচ্ছিল। এবং সবশেষে গ্যাস সেন্সর(Gas Sensor)- এর উপর কিছু পরীক্ষা-নিরীক্ষা ছিল, এবং এই সেন্সরটি(Sensor) বিভিন্ন ক্ষেত্রে খুবই গুরুত্বপূর্ণ একটি সেন্সর(Sensor)।

(Refer Slide Time: 08:19)



কোর্স(Course) শেষ করার আগে এবং তোমাদের সকলকে বিদায় জানাবার আগে, আমি এবং এই কোর্সে আমার সহ নির্দেশক ডাঃ কমলিকা দত্ত, আশা করছি যে এই কোর্সটি তোমাদের কাজে এসেছে। এবং এটা অবশ্যই উচিত সেই সমস্ত মানুষের অবদান স্বীকার করা যাদের বিনা এই কোর্সটিকে(Course) ঘন্টস্থ করা, দিনের পর দিন এই কোর্সটিকে চালিয়ে নিয়ে যাওয়া সম্ভব ছিল না।

প্রথমেই, এই কোর্সের(Course) সঙ্গে যুক্ত সকল টিচিং অ্যাসিস্ট্যান্ট(Teaching Assistant) -দের অবদান স্বীকার করে নিতে চাই। তোমরা যেভাবে আলোচনায় অংশগ্রহণ করেছিলেন, সত্যিই এটা খুবই আনন্দদায়ক এবং উৎসাহব্যঞ্জক অভিজ্ঞতা ছিল আমাদের সকলের জন্য। এবং সর্বশেষে আইআইটি খড়গপুর(IIT Kharagpur) এবং আইআইটি মাদ্রাজে(IIT Madras) অবস্থিত সেন্টার অফ এডুকেশন টেকনোলজির(Center of Education Technology) সমগ্র এনপিটেল(NPTEL) টিম(Team) কে ধন্যবাদ জানাতে চাই যারা পুরো কর্মকাণ্ডটি চালাচ্ছিলেন। তারাই আসলে চেষ্টা করছেন এই প্রযুক্তি যাতে তোমাদের কাছে নিয়ে যাওয়া যায়, চেষ্টা করছেন জ্ঞানের প্রসারে, এবং একটা সুন্দর ও উৎসাহব্যঞ্জক অভিজ্ঞতা সৃষ্টিতে সাহায্য করছেন। তো এই

সামান্য কিছু কথা বলে, আমরা আরেকবার তোমাদের সকলকে ধন্যবাদ জানাতে চাই, এবং আমরা আশা করছি যে ভবিষ্যতে আবার আমাদের দেখা হবে।

ধন্যবাদ।

GLOSSARY

English	Transliteration	Translation
Development Board	ডেভলপমেন্ট বোর্ড	একটি মাইক্রোপ্রসেসর সমেত প্রিন্টেড সার্কিট বোর্ড যাতে কোনো ইউজার ইন্টারফেস থাকে না
hardware	হার্ডওয়ার	কম্পিউটার সম্পর্কিত যন্ত্রাংশ
Sensor	সেন্সর	এক প্রকার বৈদ্যুতিন যন্ত্র যা কোন ভৌত স্থিতিমাপকে মাপতে পারে
Actuator	একচুয়াটর	যে যন্ত্রের দ্বারা শক্তিকে চলনে রূপান্তরিত করা যায়।
development board	ডেভলপমেন্ট বোর্ড	মাইক্রোপ্রসেসর সহ প্রিন্টেড সার্কিট বোর্ড যাতে কোন ইউজার ইন্টারফেস থাকে না।
instruction set	ইনস্ট্রাকশন সেট	একটি সিপিইউ(CPU) কে কাজ করানোর জন্য নির্দিষ্ট নির্দেশসমূহ
machine language	মেশিন ল্যাঙ্গুয়েজ	যে ভাষা কেবলমাত্র ০ ও ১ দ্বারা গঠিত

assembly language	অ্যাসেম্বলি ল্যাঙ্গুয়েজ	মেশিন ল্যাঙ্গুয়েজ এবং হাই লেভেল ল্যাংগুয়েজ এর মধ্যে মধ্যস্থ স্তরে যে ল্যাঙ্গুয়েজটি থাকে
high level language	হাই লেভেল ল্যাংগুয়েজ	প্রোগ্রাম যে ভাষায় লেখা হয়
programmable	প্রোগ্রামেবল	যে বস্তুকে প্রোগ্রাম দ্বারা নিয়ন্ত্রণ করা যায়।



EMBEDDED SYSTEM DESIGN WITH ARM

Bengali

Translated by

Name	Institute
SARASIJ MAJUMDAR	DG 1 BY 1 NEW TOWN MEGA CITY,RAJARHAT KOLKATA
DEBASHIS GHOSH	AMRAPALI GROUP OF INSTITUTES
SAYAN CHAUDHURI	INSTITUTE OF ENGINEERING AND MANAGEMENT,KOLKATA
PRADYUMNA ROY	GOVERNMENT POLYTECHNIC, DAMAN

Verified by

Name	Institute
SAUVIK BAL	UNIVERSITY OF ENGINEERING AND MANAGEMENT



swayam.gov.in/NPTEL



nptel.ac.in