রোবটিকসের মাস্টারপ্ল্যান

রোবটিকসের মাস্টারপ্ল্যান

আবরার শাহরিয়ার স্বপ্নীল
সুবহা নাওয়ার পুষ্পিতা

আদর্শ

প্রকাশক : আদর্শ
কনকর্ড অ্যাম্পোরিয়াম, কাঁটাবন, ঢাকা ১২০৫
৩৮ পি. কে. রায় রোড, বাংলাবাজার (২য় তলা), ঢাকা ১১০০
☎+০২-৯৬১২৮৭৭, ০১৭৯৩২৯৬২০২, ০১৭১০৭৭৯০৫০
info@adarshapublications.com
www.adarshapublications.com

রোবটিকসের মাস্টারপ্ল্যান

১ম প্রকাশ : ০০ পৌষ ০০০০; ০০ জানুয়ারি ০০০০

© আবরার শাহরিয়ার স্বপ্নীল, সুবহা নাওয়ার পুষ্পিতা
সর্বস্বত্ব সংরক্ষিত; লেখক, প্রকাশক ও অনুবাদকের লিখিত অনুমতি ব্যতীত
বইটি বা বইটির অংশবিশেষ যেকোনো মাধ্যমে প্রকাশ সম্পূর্ণ নিষিদ্ধ

প্রচ্ছদ : মোস্তাফিজ কারিগর
মুদ্রণ ব্যবস্থাপনা : আদর্শ প্রিন্টার্স
আদর্শর বই অনলাইনে কেনার লিংক : www.rokomari.com/adarsha

মূল্য : বাংলাদেশে ৩০০ টাকা

Robotics-er Masterplan (Published in Bengali)
by *Abrar Shahriar Swopnil, Subha Nawer Pushpita*
Published by Adarsha
Concord Emporium, Kataban, Dhaka 1205
38 P. K. Ray Road, Banglabazar (1st floor), Dhaka 1100

ISBN: 978-984-8040-42-3

উৎসর্গ

আমাদের শ্রদ্ধেয় পিতা মাতা

সূচি

ভূমিকা ১১

রোবটিকসের প্রাথমিক ধারণা ১৩

স্বনিয়ন্ত্রিত ব্যবস্থা বা মাইক্রোকন্ট্রোলার ১৩

মেশিন ল্যাঙ্গুয়েজ ও প্রোগ্রামিং ভাষা ১৪

সিগন্যাল ১৪

সেন্সর ১৬

রোধ ১৭

রোধের সমবায় ১৯

নিষ্ক্রিয়কারী উপাদান ২০

ফটো রেসিস্টর ২০

ক্যাপাসিটর বা ধারক ২০

ডায়োড ২২

স্কটকি ডায়োড ২২

লাইট-এমিটিং ডায়োড ২৩

ট্রানজিস্টর ২৪

ট্রান্সফরমার ২৫

বিভিন্ন ধরনের সেন্সর ২৬

মোটর ২৮

মোটর স্পেসিফিকেশন ৩০

পাওয়ার সাপ্লাই ৩২
সাধারণ ডিসি ব্যাটারি ৩৩
ব্রেড বোর্ড ৩৬
এলসিডি ডিসপ্লে (LCD) ৩৭
সোল্ডারিং কী? ৪০
মাল্টিমিটার ৪১

আরডুইনো ও মাইক্রোকন্ট্রোলার প্রোগ্রামিংয়ের বেসিক ৪৩

কিছু সফটওয়্যার বেসিক ৪৫
সি প্রোগ্রামিং দিয়ে শুরু ৬১
অ্যানালগ ও ডিজিটাল ইনপুট ও আউটপুট ৮২

প্রজেক্ট (প্রথম অংশ) ৮৬

প্রজেক্ট ১ : ডিজিটাল আউটপুটের সাহায্যে LED বাল্ব জ্বালানো ৮৬
প্রজেক্ট ২ : অ্যানালগ আউটপুটের সাহায্যে LED বাল্ব জ্বালানো ৯৪
প্রজেক্ট ৩ : ডিজিটাল ও অ্যানালগ ইনপুট ও এর কাজ ৯৬

মোটর ১০৭

ডিসি মোটর ১০৭
মোটর চালানো ১০৯
মোটর ড্রাইভার ১১১
প্রজেক্ট ১ : মোটরের গতি পর্যবেক্ষণ ১১৫
প্রজেক্ট ২ : সার্ভো মোটর নির্দিষ্ট কোণে রাখা ১২৫

সেন্সর ১২৮

আই আর সেন্সর ১২৮
সোনার সেন্সর ১৩১
টেম্পারেচার সেন্সর ১৩৩
পিএজো ইলেক্ট্রিক সেন্সর ১৩৪
স্ট্রেইন গজ ১৩৬

বেন্ডিং সেন্সর ১৩৭

রোবটিকসের প্রোজেক্ট ১৪০

লাইট ফলোয়ার রোবট ১৪০

অটোমেটিক পাম্প কন্ট্রোল ১৪৩

সার্ভো কন্ট্রোল্ড রোবটিক আরম ১৫২

লাইন ফলোয়ার রোবট ১৫৮

পরিশিষ্ট ১৬৫

প্রয়োজনীয় উপকরণ ১৬৫

শেষের শুরু ১৬৭

ভূমিকা

সবাইকে নতুন এক রঙিন জগতে স্বাগত!!

রোবটিকস ও অটোমেশন পুরো পৃথিবীতে একসময় বিপ্লবের কারণ হবে। তবে আমরা সব কাজের ক্ষেত্রে প্রাধান্য দিই আনন্দ ও ভালোবাসাকে। রোবটিকসের সবচেয়ে আনন্দদায়ক ব্যাপার হচ্ছে এতে যেমন সমস্যা সমাধানের আনন্দ আছে, তেমনি আছে সৃষ্টিশীল হয়ে নতুন কিছু নিজ হাতে তৈরি করার উচ্ছ্বাস। কারণ সবার কাছেই তার সৃষ্টি অনেক প্রিয় হয়।

পৃথিবী শাসনে অংশ নিচ্ছে এবং নেবে এমন একটি বিষয়ের ওপর ভালো বাংলা বইয়ের অভাব সব সময় ছিল। কিন্তু নতুন প্রজন্মের সামর্থ্য বিস্ময়-জাগানিয়া। তাই এই বইটিতে রোবটিকস ও ইলেকট্রনিকসের ওপর প্রাথমিক ধারণা থেকে শুরু করে একেবারে বিস্তারিত অনেকগুলো বড় প্রজেক্ট দেওয়া হয়েছে। এটি এমনভাবে লেখা হয়েছে যে প্রোগ্রামিং ও ইলেকট্রনিকসে আগ্রহ এবং সমস্যা সমাধানের পেছনে লেগে থাকার মানসিকতা আছে এমন যে কেউ এই বইয়ের পুরোটা বুঝতে পারবে। বইটিতে সবকিছু এমনভাবে উপস্থাপন করা হয়েছে যে তা রোবটিকসের প্রতি পাঠকের আগ্রহ বৃদ্ধির পাশাপাশি এ বিষয়ের ওপর তার দক্ষতার ভিত আরও শক্তিশালী করবে, যা তাকে ভবিষ্যতে বড় স্বপ্ন দেখতে সহায়তা করবে।

এই বইয়ের কাজী শাফি সামি রোবটিকসের প্রাথমিক ধারণা অংশ এবং মো. আশিকুর রহমান আশিক সেন্সর অংশটি লিখেছে, তাদের প্রতি রইল আমাদের বিশেষ কৃতজ্ঞতা।

এই বইয়ে নাম না-জানা অনেক ব্লগারের থেকে আমরা উপকৃত হয়েছি, তাদের জন্য রইল অনেক কৃতজ্ঞতা। বাংলায় ভালো রোবটিকস বইয়ের যে অভাব ছিল এই বই তা মেটাতে পারবে বলে আমাদের বিশ্বাস।

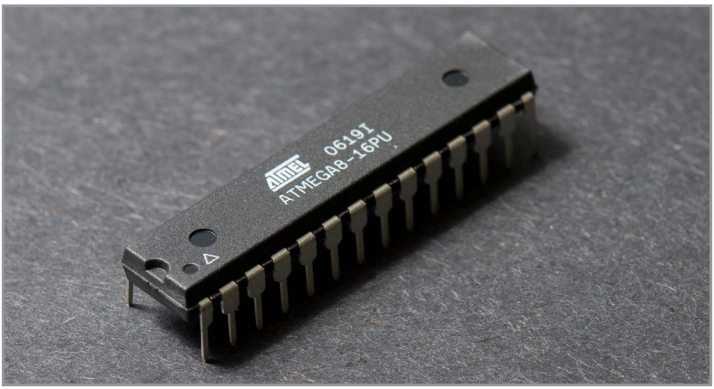
যারা এই অসাধারণ জগতে যাত্রা শুরু করতে যাচ্ছ, তাদের জন্য রইল শুভকামনা।

লেখকবৃন্দ

প্রথম অধ্যায়

রোবটিকসের প্রাথমিক ধারণা

স্বনিয়ন্ত্রিত ব্যবস্থা বা মাইক্রোকন্ট্রোলার



চিত্র ১.১ : ATMEGA-8 মাইক্রোকন্ট্রোলার

মাইক্রোকন্ট্রোলার একটি ইংরেজি শব্দ। মাইক্রো শব্দের অর্থ 'অতিক্ষুদ্র', কন্ট্রোলার শব্দের অর্থ 'নিয়ন্ত্রণকারী'। মাইক্রোকন্ট্রোলার বস্তুত একটা ছোট কম্পিউটার, এখানে কম্পিউটারের বিভিন্ন অংশ (রম, প্রসেসর, ইনপুট, আউটপুট, র‍্যাম, সিরিয়াল কমিউনিকেশন ইত্যাদি) একটা চিপের মধ্যে থাকে। সহজ ভাষায় বলতে গেলে মাইক্রোকন্ট্রোলার কম্পিউটারের মতো কিছু নির্দিষ্ট পদ্ধতি মেনে কিছু নির্দিষ্ট কাজ সম্পাদন করে। এখানে এই নিয়মটাই হল 'প্রোগ্রাম'। এখন প্রশ্ন হলো প্রোগ্রাম কী? প্রোগ্রাম হলো একটা কম্পিউটারের জন্য তৈরি কাজের ধারা, আর কম্পিউটার এই কাজগুলো সম্পন্ন করে এক নির্দিষ্ট ধারা বজায় রেখে। আর প্রোগ্রাম করাকে বলা হয় প্রোগ্রামিং। মাইক্রোকন্ট্রোলারে প্রোগ্রামিংয়ের মাধ্যমে

যেভাবে কাজ করতে নির্দেশ দেওয়া হবে এটা ঠিক সেই অনুযায়ী কাজ করে দেবে। তাই মাইক্রোকন্ট্রোলারকে কোনো ব্যবস্থার মস্তিষ্ক বলা হয়ে থাকে।

মেশিন ল্যাঙ্গুয়েজ ও প্রোগ্রামিং ভাষা

সহজ ভাষায় বলতে গেলে, কম্পিউটার বস্তুত শুধু বাইনারি নম্বরই পড়তে পারে। বাইনারি সংখ্যায় দুটি অঙ্ক আছে—শূন্য (০) আর এক (১)। কম্পিউটারকে যা-ই ইনপুট দেওয়া হোক না কেন যন্ত্রটি শূন্য আর এক অর্থাৎ বাইনারিতে সেই উপাত্ত সংরক্ষণ করে, হিসাব করে এবং সেই অনুসারেই কাজ করে। কম্পিউটার বা মাইক্রোকন্ট্রোলার শূন্য ও একের মাধ্যমে প্রদান করা ইনপুটগুলো বুঝতে পারে। তাই বলা যায় কম্পিউটারের এই ভাষার নাম মেশিন ল্যাঙ্গুয়েজ। কিন্তু আমাদের অনেক বড় বড় ডেটা নিয়ে কাজ করতে হয়, তখন তখন আমাদের এই বড় ডেটাগুলোকে বাইনারিতে রূপান্তর করা প্রায় দুঃসাধ্য। তাই এমন একটা পদ্ধতি প্রয়োজন, যার মাধ্যমে আমরা আমাদের মতো করে কাজের প্রয়োজন অনুযায়ী সহজ একটা ভাষায় লিখে দেব কম্পিটারকে। আর একটি বিশেষ পদ্ধতি বা সফটওয়্যারটি আমাদের ভাষাকে কনভার্ট করে বাইনারিতে রূপান্তর করে দেবে যেন কম্পিউটার বা মেশিন বুঝতে পারে, যাকে বলা হয় ইন্টারপ্রেটার। আর সেই বিশেষ আমাদের ইনপুট দেওয়া ভাষাটাই হলো প্রোগ্রামিং ভাষা। পৃথিবীতে অজস্র প্রোগ্রামিং ভাষা আছে। যেমন সি, জাভা, ওরাকল, ফরট্রান, পাইথন, রুবি ইত্যাদি।

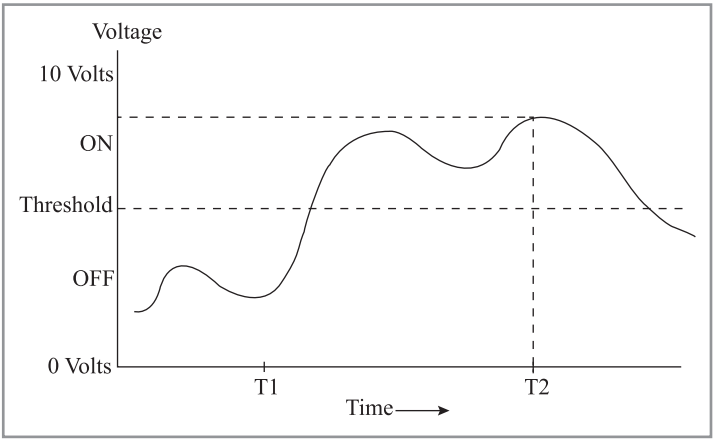
সিগন্যাল

ইলেকট্রনিকসে সিগন্যাল হলো একধরনের পদ্ধতি যার মাধ্যমে কিছু গুরুত্বপূর্ণ তথ্য প্রেরণ করা যায়। একটা রেকর্ডিং যন্ত্র তৈরি করতে গেলে সেই যন্ত্র নিশ্চইয় আমরা যেভাবে গান গেয়েছি, সেভাবে গানটিকে সংরক্ষণ করবে না। সে যা করবে তা হলো, আমাদের গলার শব্দকে কিছু তথ্যের প্যাকেট (অডিও সিগন্যাল) আকারে গ্রহণ করবে এবং একে প্রসেস বা প্রক্রিয়াকরণ করার পর তা সংরক্ষণ করবে। সহজ ভাষায় এই তথ্য উপাত্তের প্যাকেটটাই হলো সিগন্যাল।

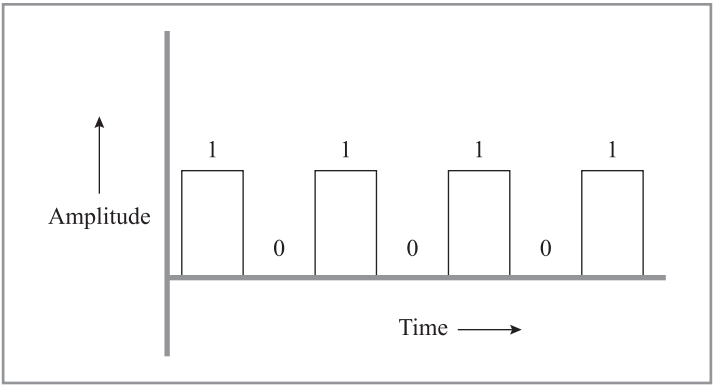
সিগন্যাল অ্যানালগ বা ডিজিটাল হতে পারে। সময়ের সাথে পরিবর্তনশীল বৈদ্যুতিক সিগন্যালকে অ্যানালগ সিগন্যাল বলা হয়। যেমন অডিও সিগন্যাল, ছবির উজ্জ্বলতা, আসলে দুনিয়ায় যা কিছু গণনা ও কোন কিছু

সাপেক্ষে বিশ্লেষণ করা যায়, তার সবকিছুই অ্যানালগ সিগন্যাল আকারে প্রকাশ করা যায়। অ্যানালগ সিগন্যালের বিস্তার বা অ্যামপ্লিচিউড সর্বোচ্চ মান এবং সর্বনিম্ন মানের মধ্যে ইনপুট অনুযায়ী যেকোনো মান গ্রহণ করতে পারে। এই নির্দিষ্ট মান হঠাৎ করেই যেকোনো মানে পরিবর্তিত হতে পারে না। অ্যানালগ সিগন্যালের যেকোনো মানে পরিবর্তিত হতে কিছুটা সময়ের দরকার হয়। অ্যানালগ সিগন্যালের সমস্যা হলো, এর ক্ষমতা ধীরে ধীরে কমতে থাকে যাকে বাঁচিয়ে রাখতে হলে বিবর্ধন করতে হয়। কিন্তু এতে নয়েজ বা বিশৃঙ্খলা বেড়ে যায়। তাই সিগন্যালের মানের পরিবর্তন, অর্থাৎ সিগন্যাল বিকৃত হয়ে যায় এবং এমনকি হারিয়েও যেতে পারে। তাই অ্যানালগ সিগন্যাল প্রক্রিয়াকরণেরর জন্য স্যাম্পল এবং হোল্ড সার্কিট দরকার হয়।

ডিজিটাল সিগন্যাল এমন একটি সিগন্যাল যা সাধারণত দুটি ভোল্টেজ লেভেল নিয়ে গঠিত। সচরাচর ডিজিটাল সিগন্যালের ভোল্টেজ লেভেল দুটিকে লজিক 0 এবং লজিক 1 দ্বারা প্রকাশ করা হয়। ডিজিটাল সিগনাল খুব সহজে যেকোন মানে পরিবর্তিত হতে পারে। ডিজিটাল সিগনাল ধাপে ধাপে সম্মুখে বিবর্ধিত হয় এবং এর সর্বশেষ সিগন্যালটি আগের সিগন্যালগুলোর মতো গুণগত মান বজায় রাখে। এর সবচেয়ে বড় সুবিধা হলো প্রতি সেকেন্ডে অনেক বেশি সিগন্যাল বা তথ্য প্রেরণ করা যায়। অ্যানালগ সিগন্যালে ক্রস কানেকশন হলেও ডিজিটাল সিগন্যালে ক্রস কানেকশন হওয়ার কোনো সুযোগ নেই।



চিত্র ১.২ : অ্যানালগ সিগন্যাল গ্রাফ



চিত্র ১.৩ : ডিজিটাল সিগন্যাল গ্রাফ

সেন্সর

এবার আসি সেন্সর প্রসঙ্গে। সেন্সরকে বলা যায় রোবটের ইন্দ্রিয়। ধরা যাক, আমরা যখন আগুনের কাছে যাই, চারপাশের তাপমাত্রার পরিবর্তন হয়। আমাদের স্পর্শ ইন্দ্রিয় আপনার মস্তিষ্কে এই তাপমাত্রার পরিবর্তনকে তথ্যের প্যাকেট আকারে পাঠায় এবং আমরা বুঝতে পারব যে আমাদের গরম লাগছে।

সহজ কথায় বলতে, সেন্সর হলো একধরনের পরিবর্তনকারী ব্যবস্থা যা পরিবেশগত কোনো পরিবর্তনকে (যেমন চাপ, তাপ, উষ্মতা ইত্যাদিকে) একটি তড়িৎ সিগন্যালে (যেমন-ভোল্টেজ বা বিভব, ধারকত্ব, চুম্বক কাপলিং) পরিণত করে। সেই সাথে প্রাপ্ত সিগন্যালকে বিশেষ প্রক্রিয়ার মাধ্যমে পরবর্তীতে একটি মাধ্যমে আউটপুট বা ফলাফল হিসেবে প্রদর্শন করা যায়। ধরা যাক, আমরা একটি ভারী জিনিসের ওজন মাপতে চাই। তাই একটি বিদ্যুৎ পরিবাহী তামার তার নিলাম। প্রথমে তামার তার দুই প্রান্তে ব্যাটারি লাগিয়ে ভোল্টমিটারের সাহায্যে ভোল্টেজ মাপলাম। এরপর তামার তারে জানা ভরে ভারী বস্তু ঝুলিয়ে আগের পদ্ধতি অনুসরণ করলে দেখা যাবে ভোল্টেজের মানে সামান্য পরিবর্তন এসেছে। কারণ বস্তু ঝোলানোর পর তারের রোধ বেড়ে যায়। আর রোধ বাড়লে বা কমলে বিভব বা ভোল্টেজের মানের পরিবর্তন হয়। এভাবে আমরা যদি কিছু জানা ভরের সাহায্যে এই প্রক্রিয়া চলমান রাখি, একসময় আমরা এই ভোল্টেজ আর বস্তুর ভরের মধ্যে একটা সম্পর্ক পাব। এভাবে সম্পর্ক তৈরির প্রক্রিয়াকে বলা হয় ক্যালিব্রেশন। উপরিউক্ত সম্পর্ক

থেকে যেকোনো একটা প্রাপ্ত ভোল্টেজের মান থেকে আমরা যেকোনো বস্তুর ভর বের করতে পারব। এখন এই পুরো ব্যবস্থাকে যদি তোমরা ডিজিটাল করতে পার, তবে তোমাদের অভিনন্দন। তোমরা সেন্সর বানিয়ে ফেলেছ।

ক্যালিব্রেশন দ্বারা কোনো সেন্সরকে আরও নিখুত করে তোলা যায়। কিছু সেন্সরের উদাহরণ দিতে হলে যা বলতে হয়—

কন্টাক্ট সেন্সর : এই সেন্সর বলে দেয় কোনো কিছু খোলা বা বন্ধ হয়ে ছিল কি না। নিরাপত্তার কাজ বহুল ব্যবহৃত।

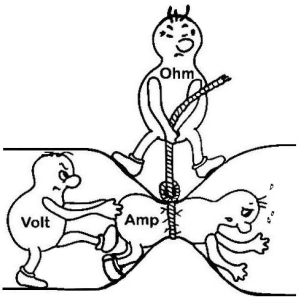
টেম্পারেচার সেন্সর : তাপমাত্রা মাপার কাজে ব্যবহৃত।

ডিসটেন্স বা দূরত্ব সেন্সর : অবস্থান ও চলনের গতি নির্দেশ করতে পারে।

প্রক্সিমিটি সেন্সর : আশপাশে কিছু আছে কি না, তা বের করতে পারে।

রোধ

রোধের ব্যাপারটাকে সহজভাবে বোঝানোর জন্য নিচের চিত্রটি আনন্দদায়ক হতে পারে—



চিত্র ১.৪ : রোধের কার্যপদ্ধতি

রোধের একক ও'ম (Ω), বিভবের একক ভোল্টেজ (V), প্রবাহের একক অ্যাম্পিয়ার (A)।

কোনো পরিবাহকের রোধ মূলত চারটি বিষয়ের ওপর নির্ভরশীল, যেমন 'সেন্সর' শীর্ষক অংশে একটা উদাহরণ দেওয়া হয়েছিল— যে পরিবাহকের দৈর্ঘ্য বাড়লে রোধ বেড়ে যায়। শুধু তা-ই না পরিবাহকের প্রস্থচ্ছেদের ক্ষেত্রফল বাড়ালেও তড়িৎ পরিবহনের সুযোগ বেড়ে যায়, রোধ কমে

যায়। এ ছাড়া রোধ পরিবাহকের উপাদান ও তাপমাত্রার ওপর নির্ভর করে। তাপমাত্রার যেহেতু হরহামেশাই পরিবর্তন হয় তাই রোধকের রোধের মান সুনির্দিষ্টভাবে বলা খুব কঠিন। এ জন্য নির্মাতারা রোধকের মান কতটুকু পরিবর্তন হতে পারে, তাকে রোধকের শুদ্ধতা বা টলারেন্স আকারে প্রকাশ করেন। এই কার্বন রোধকের রোধ এবং টলারেন্সকে প্রকাশ করার জন্য নির্মাতারা কিছু রঙিন পট্টি বা দাগ ব্যবহার করেন। এই পদ্ধতিতে রোধকের মান প্রকাশ করাকে বলা হয়ে থাকে রোধের বর্ণ কোড বা কালার কোড। কালার কোডের মাধ্যমে পট্টির রং এবং অবস্থান অনুসারে রোধের মান বের করা যায়। বর্তনী তৈরি করার সময় বিভিন্ন বর্ণের কোড লাগতে পারে, তাই নিচে দিয়ে দেওয়া হলো—

কোনো resistor-এ যদি ৪টি রঙের দাগ দেয়ো থাকে, তাহলে তাকে বলা হয় 4-band resistor. একইভাবে ৫টি ব্যান্ড থাকলে হবে 5-band resisitor.

4-band code-এর ক্ষেত্রে রোধের মান বের করার সূত্রটি হলো—

R = FS × 10^T ± Tolerance

এখানে,

F = 1st band বা 1st digit

S = 2nd band

T = 3rd band (10^T-কে Multiplier বা গুণক বলে)

5-band code এর ক্ষেত্রে,

R = FSV × 10^T ± Tolerance

এখানে,

F = 1st band

S = 2nd band

V = 3rd band

T = 4th band

যেমন ধরা যাক—

1st digit, F = Yellow = 4

2nd digit, S = Violet = 7

3rd digit, T = Red = 2; অর্থাৎ, Multiplier = 10^2

Tolerance = Silver = ±10%

সুতরাং,

R = FS × 10^T ± Tolerance

= 47 × 10^2 ± 10

= 4700 ohms ± 10%

Color	Digit	Multiplier	Tolerance (%)
Black	0	10^0 (1)	
Brown	1	10^1	1
Red	2	10^2	2
Orange	3	10^3	
Yellow	4	10^4	
Green	5	10^5	0.5
Blue	6	10^6	0.25
Violet	7	10^7	0.1
Grey	8	10^8	
White	9	10^9	
Gold		10^{-1}	5
Silver		10^{-2}	10
(none)			20

ছক ১.১ : রোধের কালার কোডের সমবায়

রোধের সমবায়

রোধের সমবায় সম্পর্কে বলতে হলে প্রথমে ওমের সূত্র সম্পর্কে বলতে হয়। ১৮১৮ সালে জার্মান বিজ্ঞানী জর্জ সাইমন ওম এই সূত্রটি আবিষ্কার করেন। তিনি বলেন যে তাপমাত্রার পরিবর্তন না করলে বিদ্যুৎ প্রবাহ পরিবাহকের দুই প্রান্তের বিভব পার্থক্যের সমানুপাতিক। সেই সাথে তিনি প্রমাণ করেন যে, অর্থাৎ বিভব রোধ এবং বিদ্যুৎ প্রবাহের গুণফলের সমান। এই সূত্রের সাহায্যে হয়তো প্রথম তুল্য সিস্টেম সম্পর্কে ধারণা হয়েছিল। তুল্য রোধ বলতে এমন একটি রোধ বোঝায়, যেখানে

অনেকগুলো রোধের পরিবর্তে সেই রোধটা ব্যবহার করলে বিভবের মানের এবং তড়িৎ প্রবাহের পরিবর্তন হয় না। এবার আসি সমবায় প্রসঙ্গে। রোধের সমবায় বা রোধ সাজানো সাধারণত দুই ধরনের হয়ে থাকে। সিরিজ ও সমান্তরাল সমবায়। শ্রেণি সমবায়ে রোধগুলো একটির এক প্রান্তের সাথে অপরটির এক প্রান্ত, এভাবে পরপর সাজানো থাকে এবং রোধগুলোর মাঝে তড়িৎ প্রবাহের পরিবর্তন হয়না।

সমান্তরাল সমবায়ে রোধগুলোর মধ্যে সাধারণত দুটি সাধারণ প্রান্ত থাকে। সমান্তরাল সমবায়ে দুই প্রান্তে বিভবের মানের পরিবর্তন হয় না। সাধারনত বর্তনীগুলো দুটির সমন্বয়ে গঠিত হয়।

নিষ্ক্রিয়কারী উপাদান

যেসব ইলেকট্রিক্যাল কম্পোনেন্ট তড়িৎ শক্তি উৎপাদন করতে পারে না; তবে শক্তি সঞ্চয় ও রূপান্তর করতে পারে, তাদের নিষ্ক্রিয়কারী উপাদান বলে।

ফটো রেসিস্টর

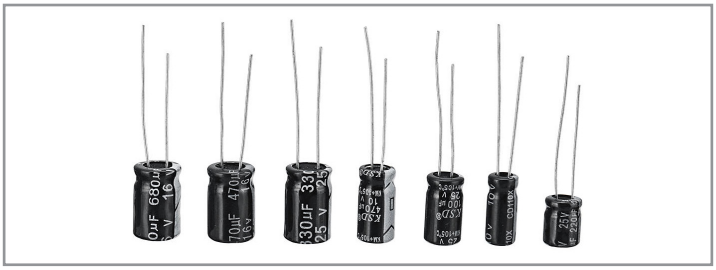
এ ধরনের রোধকগুলো আলোর তীব্রতার ওপর নির্ভরশীল। আলোর তীব্রতা বাড়লে রোধকের মান কমে যায় এবং আলোর তীব্রতা কমলে রোধকের মান বেড়ে যায়। এই রোধকগুলো সাধারণত উচ্চ রোধসম্পন্ন অর্ধ পরিবাহী দ্বারা তৈরি করা হয়। আলোক সংবেদি বর্তনীতে ফটো রেসিস্টরের বহুল ব্যবহার রয়েছে।

ক্যাপাসিটর বা ধারক

ক্যাপসিটার একধরনের নিষ্ক্রিয়কারী উপাদান, যা আধানরূপে তড়িৎ শক্তি সঞ্চয় করতে পারে। রোধকের মতো এর দুটি প্রান্ত থাকে। তড়িৎ কোষ বা ব্যাটারির সাথে সংযোগ করলে এক প্রান্তে ধনাত্মক আধান সঞ্চিত হয় এবং অপর প্রান্তে ঋণাত্মক আধান সঞ্চিত হয়। ধারকের চার্জ সঞ্চয় করার ধর্মকে ধারকত্ব বলা হয় থাকে।

ধারকে দুটি পরিবাহী পাতের মধ্যে ডাই-ইলেকট্রিক পদার্থ (একধরনের অন্তরক) দ্বারা পূর্ণ থাকে। ধারকেদী ধারকত্ব এই পাতদ্বয়ের ক্ষেত্রফল ও পাতদ্বয়ের মধ্যবর্তী দূরত্বের ওপর নির্ভর করে। কোনো ধারক সম্পূর্ণ আহিত হলে, সমপ্রবাহ বা ডিসি কারেন্ট প্রবাহিত হতে পারে না। কিন্তু

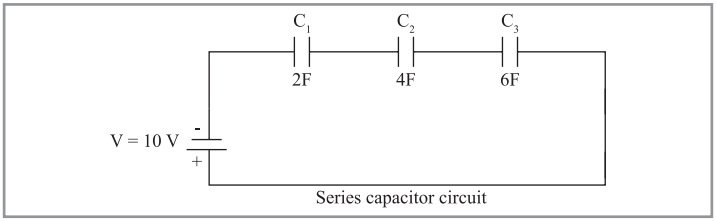
দিক পরিবর্তিত প্রবাহ বা এসি কারেন্ট ব্যবহার করা হলে, একটি নির্দিষ্ট পর্যায় পরপর ধারক আহিত ও ক্ষরিত হয়। সেই সাথে তড়িৎ আধান সঞ্চয় করতে পারে। ধারকত্বের একক ফ্যারাড। ধারক বা ক্যাপাসিটর রোধের মতো ছোট না হওয়ায় এর গায়ে ধারকত্বের মান দেওয়া থাকে। যেমন নিচের ক্যাপাসিটরে ধারকত্ব ৩৩০০ মাইক্রো ফ্যারাড ও কার্যকারী ভোল্টেজ ২৫ ভোল্ট। অর্থাৎ সর্বাধিক ২৫ ভোল্ট ক্যাপাসিটরে দেওয়া যাবে।



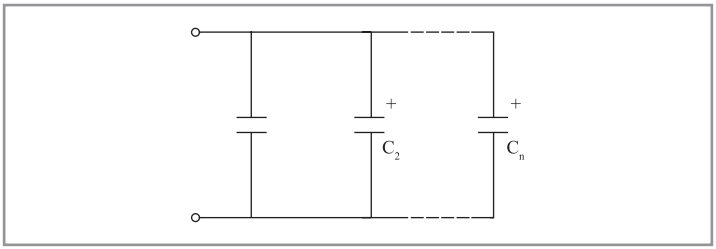
চিত্র ১.৫ : ক্যাপাসিটর

ধারকের সমবায়

রোধকের মতো ধারকের তিন ধরনের সমবায় তৈরি করা যায়।



চিত্র ১.৬ : ধারকের শ্রেণি সমবায়



চিত্র ১.৭ : ধারকের সমান্তরাল সমবায়

ধারকের ব্যবহার : ধারক শক্তি সঞ্চয়, ফিল্টার, মোটর স্টারটার, বিভিন্ন সেন্সর, রেডিও টিউনিং সার্কিটসহ আরও অসংখ্য কাজে ব্যবহার করা হয়।

ডায়োড

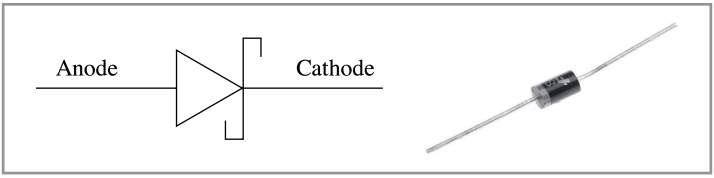
ডায়োড মূলত দুই প্রান্তবিশিষ্ট একধরনের ইলেকট্রনিক যন্ত্রাংশ, যার মাধ্যমে নির্দিষ্ট দিকে বিদ্যুৎ প্রবাহ বা রেক্টিফিকেশন করা যায়। সহজভাবে বলা যায়, ডায়োড শুধু একটি নির্দিষ্ট দিকেই বিদ্যুৎ পরিবহন করতে পারে। স্বাভাবিকভাবেই বুঝতে পারছো, পরিবর্তী প্রবাহকে সমপ্রবাহে পরিণত করতে আমাদের ডায়োড প্রয়োজন।

ডায়োড অর্ধপরিবাহী পদার্থ যেমন সিলিকন, জারমেনিয়াম দিয়ে তৈরি করা হয়ে থাকে। সাধারণত বিশুদ্ধ অর্ধপরিবাহী ক্রিস্টালের মধ্যে গ্যালিয়াম বা অ্যালুমিনিয়াম (পর্যায় সারণির ১৩ নং গ্রুপ) মিশিয়ে পি-টাইপ অর্ধপরিবাহী এবং আর্সেনিক বা অ্যান্টিমনি (পর্যায় সারণির ১৫ নং গ্রুপ) মিশিয়ে এন-টাইপ অর্ধপরিবাহী তৈরি করা হয় থাকে। এই দুই অর্ধপরিবাহীর সংযোগের ফলে তৈরি হয় পি-এন জাংশন। এই একটি পি-এন জাংশন দিয়ে তৈরি হয় ডায়োড। ডায়োডের এক প্রান্তে আনোড এবং অপর প্রান্তে ক্যাথোড থাকে। সাধারণত ডায়োডে বিদ্যুৎ প্রবাহের সময় খানিকটা ভোল্টেজ ড্রপ হয়।

আমার এই বইয়ে পাওয়ার সাপ্লাই ও অন্যান্য কাজে স্কটকি ডায়োড ও লাইট এমিটিং ডায়োড সংক্ষেপে এলইডি নিয়ে কাজ করব।

স্কটকি ডায়োড

ডিজিটাল সার্কিটে সাধারণত কম ভোল্টেজ ড্রপ করে এ রকম ডায়োড দরকার হয়। এ কারণেই মূলত স্কটকি ডায়োড ব্যবহার করা হয়। এই ডায়োডের ভোল্টেজ ড্রপ অনেক কম যেমন ০.১ বা ০.২ ভোল্টের মতো। একটি এন-টাইপ অর্ধপরিবাহীর সাথে ধাতুর সংযোগে এই ডায়োড তৈরি করা হয়। পাওয়ার সাপ্লাই তৈরির সময় আমরা এই ডায়োড ব্যবহার করব।



চিত্র ১.৮ : ডায়োড

লাইট-এমিটিং ডায়োড

লাইট-এমিটিং ডায়োড (Light-Emitting Diode) অথবা এলইডি (LED) ইলেকট্রনিক ডিভাইসে প্রচুর পরিমাণে ব্যবহার করা হয়। আমরা অনেকেই বিজ্ঞানী প্ল্যাঙ্কের E=hv সূত্র সম্পর্কে জানি, যেটা বিজ্ঞানী বোর ব্যাখ্যা করেছিলেন যে, উচ্চ শক্তিস্তর থেকে নিম্ন শক্তিস্তরে ইলেকট্রন স্থানান্তর হলে, শক্তির বিকিরণ ঘটে। দৃশ্যমান আলোরূপে এটা ঘটে থাকে। এলইডি ঠিক এভাবেই আলো বিকিরণ করতে পারে। মজার বিষয়, এটা এমন একটি ডায়োড যা কি না একই সাথে একটি সেমিকন্ডাক্টর এবং একই সাথে আলোও বিকিরণ করে থাকে। এতে খুবই কম তড়িৎ প্রবাহ প্রয়োজন হয়। সাধারণত ১০-২০ মিলি অ্যাম্পিয়ার কারেন্ট ও ৩ ভোল্ট একটি এলইডি জ্বালানোর জন্য ব্যবহার হয় থাকে। কিছু ডায়োড সাতটি পর্যন্ত রং প্রদর্শন করতে পারে।

এলইডি প্রকৃতপক্ষে একটি সম্মুখ ঝোঁকবিশিষ্ট P-N জাংশন ডায়োড। এটি GaAs (গ্যালিয়াম-আর্সেনাইড) GaP (গ্যালিয়াম ফসফাইড) ইত্যাদি অর্ধপরিবাহী যৌগ দ্বারা তৈরি করা হয়, যাতে তাদের বেশির ভাগ শক্তি আলো হিসেবে বিকিরিত হয়।

এই বইয়ে আমরা এলইডি নিয়ে একটি প্রজেক্ট পরবর্তী সময়ে দেখাব।

ডায়োডের ব্যবহার

১. রেক্টিফিকেশন বা বিদ্যুৎ প্রবাহ একমুখীকরণ।

২. অতিরিক্ত ভোল্টেজ বৈদ্যুতিক যন্ত্রের সুরক্ষা।

৩. আইসি তৈরিতে।

ট্রানজিস্টর

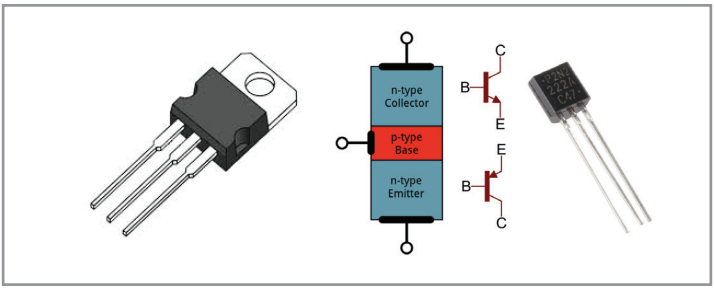
ট্রানজিস্টর (Transistor) একটি তিন প্রান্তবিশিষ্ট অর্ধপরিবাহী যন্ত্র যা সাধারণত অ্যামপ্লিফায়ার এবং বৈদ্যুতিকভাবে নিয়ন্ত্রিত সুইচ হিসেবে ব্যবহার করা হয়। আধুনিক ইলেকট্রনিক্সের মূল গাঠনিক উপাদান হিসেবে ট্রানজিস্টর ব্যবহার করা হয়। ট্রানজিস্টর সত্যিকার অর্থেই ইলেকট্রনিকসের জগতে বিপ্লব নিয়ে এসেছে। দ্রুত সাড়া প্রদানের ক্ষমতা এবং সঠিক সম্পূর্ণ সঠিকভাবে কাজ সম্পদনের ক্ষমতার কারণে এটি আধুনিক ডিজাটাল বা অ্যানালগ যন্ত্রপাতি তৈরিতে প্রচুর ব্যবহার করা হয়। নির্দিষ্ট ব্যবহারগুলোর মধ্যে রয়েছে ইলেকট্রনিক অ্যামপ্লিফায়ার, সুইচ, ভোল্টেজ নিয়ন্ত্রক, সিগন্যাল প্রসেসিং এবং ওসিলেটর। আবার সমন্বিত বর্তনী বা আইসির অভ্যন্তরে একটি অতি ক্ষুদ্র স্থানে কয়েক মিলিয়ন পর্যন্ত ট্রানজিস্টর সংযুক্ত করা যায়।

দুই ধরনের ট্রানজিস্টর সবচেয়ে বেশি দেখা যায়। যেমন—

১. বাইপোলার।

২. ফিল্ড ইফেক্ট ট্রানজিস্টর।

বাইপোলার ট্রানজিস্টর দুটি ভিন্ন ধরনের অর্ধপরিবাহীর (পি-টাইপ ও এন-টাইপ) তিনটি দিয়ে ট্রানজিস্টর তৈরি করা হয়ে থাকে। ট্রানজিস্টরে সজ্জিত অর্ধপরিবাহী কেলাসগুলোর প্রথমটিকে বলা হয় নিঃসরক বা এমিটার, যা চার্জ সরবরাহ করে থাকে। আরেক পাশের অংশে সংগ্রাহক বা কালেক্টর থাকে, এখানে চার্জ সংগ্রহ করা হয়ে থাকে। নিঃসারক ও সংগ্রাহকের মাঝে যে অংশ থাকে, তাকে ভূমি বলা হয়ে থাকে।



চিত্র ১.৯ : ট্রানজিস্টর

ট্রান্সফরমার

ট্রান্সফরমার শব্দটির সাথে সবাই কমবেশি পরিচিত। না, এখানে ট্রান্সফরমার রোবটের কথা বলা হবে না। বরং আমাদের দৈনন্দিন ব্যবহার করা অত্যন্ত জরুরি একটা যন্ত্র সম্পর্কে আলোচনা করা হবে। ট্রান্সফরমার এমন একটি যন্ত্র, যার সাহায্যে দিক পরিবর্তিত বা এসি উচ্চ বিভবকে নিম্ন বিভবে অথবা নিম্ন বিভবকে উচ্চ বিভবে পরিণত করা যায়। ট্রান্সফরমার সাধারণত দুই অথবা তিন ফেজের হয়ে থাকে।

কাঁচা লোহার আয়তাকার মজ্জার দুই বিপরীত বাহুতে তামার তার পেঁচিয়ে ট্রান্সফরমার তৈরি করা হয়। যে বাহুতে পাওয়ার সোর্স সংযোগ করা হয় অর্থাৎ বিভব প্রয়োগ করা হয় তাকে বলা হয় মুখ্য কুণ্ডলী এবং যেখানে চুম্বক ফ্ল্যাস্কের পরিবর্তনের জন্য বিভব আবিষ্ট হয় থাকে, তাকে গৌণ কুণ্ডলী বলা হয়ে থাকে।

সাধারণত দুই ধরনের ট্রান্সফরমার ব্যবহার করা হয়ে থাকে—

১. **স্টেপ আপ ট্রান্সফরমার** (Step Up Transformer): স্টেপআপ ট্রান্সফরমার অল্প বিভবের অধিক বিদ্যুৎ প্রবাহকে অধিক বিভবের অল্প বিদ্যুৎ প্রবাহে পরিণত করে। এ ধরনের ট্রান্সফরমারে গৌণ কুণ্ডলীতে তামার তার বেশি পেঁচিয়ে প্যাক সংখ্যা বাড়ানো হয়।

২. **স্টেপ ডাউন ট্রান্সফরমার**: এটি পাওয়ার সাপ্লাই তৈরি করার সময় আমাদের কাজে লাগবে। এই ট্রান্সফরমার অধিক বিভবের অল্প বিদ্যুৎ প্রবাহকে অল্প বিভবের অধিক বিদ্যুৎ প্রবাহে পরিণত করে।

ট্রান্সফরমার রেটিং

আদর্শ ট্রান্সফরমারে কোনো পাওয়ার লস থাকে না। কিন্তু বাস্তবিক ক্ষেত্রে চার ধরনের পাওয়ার লস দেখা যায়—

১. **কপার লস**: যা বিদ্যুৎ প্রবাহের ওপর নির্ভর করে।

২. **এডি কারেন্ট লস**: যা ট্রান্সফরমার মজ্জার রোধের কারণে হয়।

৩. **হিস্টোরেসিস লস**: ট্রান্সফরমার যেহেতু এসি কারেন্টে চলে তাই বিদ্যুৎ প্রবাহের দিক পরিবর্তনের জন্য যে ধরনের লস হয়।

৪. **লিকেজ ফ্লাক্স**: এর মান যত বাড়ে, কোর ফ্লাক্সের মান তত কমে বলে এর মান সবসময় কম রাখতে হয়।

কপার লস ছাড়া বাকিগুলোকে **কোর লস** বলা হয়ে থাকে।

যেহেতু **কোর লস** নির্ভর করে বিভবের ওপর এবং কপার লস নির্ভর করে বিদ্যুৎ প্রবাহের ওপর, তাই ট্রান্সফরমারের মোট পাওয়ার লস নির্ভর করে প্রদত্ত ভোল্টেজ () এবং বিদ্যুৎ প্রবাহের () মানের ওপর।

তাই বলে ট্রান্সফরমার VA বা KVA তে রেটিং করা হয়।

কাজের সুবিধার জন্য, এখন বাজারে প্রাপ্ত ট্রান্সফরমারগুলোর গায়ে ইনপুট ও আউটপুট ভোল্টেজ সম্পর্কে বলা হয়ে থাকে। যেমন—

I/P-220v-50 Hz (এর অর্থ হলো ইনপুট ভোল্টেজ হবে ২২০ আর তার কম্পাঙ্ক হবে ৫০ হার্জ)

O/P- 12v 3000mA (আমরা আউটপুট পাব ১২ ভোল্ট এবং কারেন্ট পাব ৩ অ্যাম্পিয়ার)।

ট্রান্সফরমার সাধারণত দুই ধরনের হয়। একটা বিষয় খেয়াল রাখা প্রয়োজন, ট্রান্সফরমার কখনো ডিসি লাইনে ব্যবহার করা যায় না।

বিভিন্ন ধরনের সেন্সর

এবার আসি সেন্সর প্রসঙ্গে। এখানে মূলত আমরা প্রজেক্টে যেসব সেন্সর ব্যবহার করা হবে, সেগুলো নিয়েই সংক্ষেপে আলোচনা করব—

ইনফ্রারেড সেন্সর

ইনফ্রারেড বা অবলোহিত রশ্মি বলতে দৃশ্যমান আলো থেকে অধিক তরঙ্গ দৈর্ঘ্যবিশিষ্ট আলোকে বোঝায়, তবে আমরা খালি চোখে তা দেখতে পাই না। এই সেন্সরে মূলত এই অবলোহিত রশ্মিটাকেই ব্যবহার করা হয়। এই সেন্সরে দুটি অংশ থাকে। সহজ বাংলায় বলা যায়, গ্রাহক আর প্রেরক থাকে। এই সেন্সরে থাকে অবলোহিত এলইডি এবং সিগন্যাল নির্দেশক বা ডিটেক্টর একধরনের ফটো ট্রান্সিস্টর। প্রথমে এলইডি থেকে যেকোনো ধরনের পৃষ্ঠের ওপর অবলোহিত আলো ফেলা হয়। খুব স্বাভাবিকভাবেই পৃষ্ঠ থেকে আলো প্রতিফলিত হয়। এই প্রতিফলিত আলোটাকেই সিগন্যাল হিসেবে গ্রহণ করে বিশেষ প্রক্রিয়ায় নির্দিষ্ট রঙের আলোর জন্য একটা আলাদা আলাদা মান পাওয়া যায়। এই মান দেখেই বোঝা যায়, একটা রোবট কোনো রঙের পৃষ্ঠের ওপর চলছে।

এই সেন্সরটিকে আমরা লাইন ফলোয়িং রোবট বানানোর সময় কাজে লাগাব। এই সেন্সরটিকে আইআর সেন্সর বলা হয় থাকে। প্রাথমিকভাবে বাজারে আইআর সেন্সরগুলোর মধ্যে বাংলাদেশে জনপ্রিয় হলো টিসিআরটি-৫০০০।

আল্ট্রাসনিক সেন্সর

আল্ট্রাসনিক শব্দের ভালো বাংলা হয় শ্রবণাতীত, যা শোনা যায় না। আল্ট্রাসনিক সেন্সর মূলত এই শ্রবণাতীত শব্দটাকেই ব্যবহার করে। একটি রোবট বানিয়ে তাকে গতিশীল রাখতে হলে তাকে বাধা এড়িয়ে সামনের দিকে এগিয়ে যেতে হবে। এই বাধা এড়িয়ে যাওয়ার জন্য বাধা থেকে রোবটের দূরত্বটা জানা আবশ্যক এবং সেই অনুযায়ী রোবটের সিস্টেমটা সিদ্ধান্ত নেবে। খুব সূক্ষ্ম কাজ না হলে, সেন্সরটি এ ক্ষেত্রে ব্যবহার করা যায়। সেন্সরটি দূরত্ব নির্ণয় করে নির্দিষ্ট কম্পাঙ্কের শব্দ তরঙ্গের সাহায্যে।

আল্ট্রাসনিক সেন্সরে অধিকাংশ সেন্সরের মতোই দুটি অংশ থাকে। প্রেরক অংশে একটা নির্দিষ্ট কম্পাঙ্কের শব্দ প্রেরণ করে থাকে। আমরা প্রতিধ্বনি সম্পর্কে সবাই কমবেশি জানি। আলোর মতো শব্দেরও প্রতিফলন ঘটে। এই নির্দিষ্ট কম্পাঙ্কের শব্দ সেই প্রতিবন্ধক তলে বাধা পায় এবং প্রতিফলিত হয়। সেন্সরের গ্রাহক অংশে এই প্রতিফলিত শব্দ তরঙ্গ গ্রহণ করা হয়। এই সেন্সর মূলত শব্দ বাধা পাওয়া ও ফিরে আসার সময়টা বলে দিতে পারে। এখন প্রশ্ন হলো, সেন্সরটি কীভাবে এই দূরত্বটা মাপে? এ কাজটা কিন্তু মাইক্রোকন্ট্রলারকে করতে হবে।

ধ্রুব বেগের ক্ষেত্রে, দূরত্ব= বেগ*সময়

এ ক্ষেত্রে আমরা শব্দের বেগটা মোটামুটি জানি। ২৫ ডিগ্রি সেলসিয়াস তাপমাত্রায় শব্দের বেগ ৩৪৬ মিটার বা সেকেন্ড। এখানে সেন্সর আমাদেরকে সেন্সর থেকে শব্দের যাওয়া ও আসার মধ্যকার সময়টা বলে দিচ্ছে। তো অবশ্যই খুব সহজে মাইক্রোকন্ট্রলার তার সিস্টেমে দূরত্বটা বের করতে হবে। এখানে শব্দ যেহেতু যায় আর আসে তাই বাধা থেকে রোবটের দূরত্বের দ্বিগুণ পথ শব্দ অতিক্রম করে।

আমাদের দেশে মূলত প্রাথমিক পর্যায়ে এই সেন্সরটি ব্যবহার করা হয়ে থাকে। কিন্তু সেন্সরটি একদম নির্ভুল আউটপুট দিতে পারে না। কারণ শব্দের বেগ পরিবর্তনশীল, তাপমাত্রার ওপর নির্ভর করে থাকে। আর

আমাদের পরিবেশে বিভিন্ন ধরনের আল্ট্রাসনিক শব্দ থাকে, তাই ক্রস সিগন্যাল হওয়ার সম্ভাবনা অনেক বেশি। সেই সাথে, শব্দের প্রতিফলন প্রতিবন্ধকের পৃষ্ঠের ওপর অনেকাংশে নির্ভরশীল। তাই প্রতিফলিত আল্ট্রাসনিক শব্দের ওপর পুরোপুরি নির্ভর করা অনেকটা কঠিন। তবু সহজলভ্য ও দামের কথা চিন্তা করে এই সেন্সর লাইন ফলোয়িং রোবটে ব্যবহার করা হয়েছে।

টেম্পারেচার সেন্সর

আগেই বলেছি, টেম্পারেচার বা তাপমাত্রার পরিবর্তনের সাথে সাথে বৈদ্যুতিক ধর্ম যেমন রোধের পরিবর্তন হয়। আগে এই পরিবর্তন ব্যবহার করে থার্মোকাপল, থারমিস্টর ব্যবহার করা হতো। কিন্তু বর্তমানে তাপমাত্রা নির্ণয় করার জন্য বিশেষ ধরনের ইন্টিগ্রেটেড সার্কিট ব্যবহার করা হয়ে থাকে। যেমন টিও-৯২।

এই সেন্সর সাধারণত ভোল্টেজ আউটপুট দিয়ে থাকে। তাপমাত্রা বৃদ্ধির সাথে সাথে আউটপুট ভোল্টেজের একটা সরলরৈখিক সম্পর্ক রয়েছে। তাপমাত্রা বৃদ্ধির সাথে আউটপুট ভোল্টেজের মানটা বাড়তে থাকে।

হল ইফেক্ট সেন্সর

উচ্চমাধ্যমিক শ্রেণির সবাই কমবেশি হল প্রভাব সম্পর্কে জানে। ১৮৭৯ সালে এডউইন হল (Edwin Hall) দেখান যে, পরিবাহীর মধ্যে বিদ্যুৎ প্রবাহ থাকলে চুম্বক ক্ষেত্রের প্রভাবে বিচ্যুতি ঘটে। এই ধর্মের ওপর ভিত্তি করে হল ইফেক্ট সেন্সর তৈরি করা হয়ে থাকে। কোনো চৌম্বক ক্ষেত্রের অবস্থান নির্ণয় করার জন্য এই সেন্সর কার্যকরী। এ ছাড়া ইঞ্জিনের ফুয়েল ইন্ডিকেটর, ডিসি ট্রান্সফরমার বিভিন্ন জায়গায় এই সেন্সর বহুল ব্যবহৃত।

আমরা রোবটিক হ্যান্ড তৈরির সময় এই সেন্সর ব্যবহার করব। বাংলাদেশে সবচেয়ে ব্যবহৃত হল ইফেক্ট সেন্সরটি হলো Em-১৮৮১।

মোটর

ছোটবেলায় আমরা সবাই কমবেশি ভাঙা খেলনার ভেতর মোটর দেখেছি। রোবটিকস নিয়ে কাজ করেছে কিন্তু মোটর নিয়ে কাজ করেনি, এমন কাউকে পাওয়া কঠিন। মোটরকে সহজ ভাষায় বলা যায়, এমন একটা যন্ত্র যেখানে ইনপুট হিসেবে বিদ্যুৎ শক্তি দেওয়া হয় এবং তা যান্ত্রিক

শক্তিতে পরিণত করে। সাধারণত মোটর আউটপুট হিসেবে ঘূর্ণনশীল গতি দিয়ে থাকে। বৈদ্যুতিক মোটর মূলত ফ্যারাডের আবেশ সূত্রের ওপর ভিত্তি করে তৈরি করা হয়ে থাকে। ফ্যারাডের আবেশ সূত্র হলো, বিদ্যুৎ ক্ষেত্রের প্রভাবে চৌম্বক ক্ষেত্রে পরিবর্তন হয়।

মোটর প্রধানত দুই প্রকারের হয়—

১. এসি মোটর : পরিবর্তী প্রবাহ দ্বারা চালিত।

২. ডিসি মোটর : ডিসি প্রবাহে চলে, এমনকি ক্ষেত্র বিশেষে গঠনের জন্য এসি প্রবাহেও চলতে পারে। যেমন আরমেচার উইন্ডিং ফিল্ড উইন্ডিংয়ের সাথে সিরিজে যুক্ত থাকলে এই মোটর মোটামুটি সাধারণভাবেই চলবে কিন্তু গঠনে যদি আরমেচার উইন্ডিং ফিল্ড উইন্ডিংয়ের সাথে সমান্তরালে যুক্ত থাকে, তাহলে মোটর গরম হয়ে যাবে এবং নয়েজ তৈরি হবে, কিন্তু চলবে না।

এ ছাড়া সিঙ্ক্রোনাস মোটর, ইন্ডাকশন মোটর প্রভৃতি ধরনের মোটর আছে। রোবটিকস নিয়ে আলোচনার জন্য আমরা এখানে মোটামুটি তিন ধরনের মোটর নিয়ে আলোচনা করব।

ডিসি মোটর

এই মোটর সমপ্রবাহ দ্বারা চালিত। এই মোটরে তিনটি অংশ থাকে—

১. একটি স্থায়ী U shape স্থায়ী চুম্বক থাকে।

২. তামার তার জড়ানো আরমেচার থাকে।

৩. মোটরের সাথে ঘূর্ণমান একটি যন্ত্রাংশ থাকে যাতে বিদ্যুৎ প্রবাহের দিক পরিবর্তন করলে ঘূর্ণনের দিকও পরিবর্তিত হয়। এটি মূলত একটি তামার তার কুণ্ডলী।

ব্যাটারি বা পাওয়ার সাপ্লাই থেকে মোটরের ধনাত্মক আর ঋণাত্মক প্রান্তের সাথে সংযোগ দিলে আরমেচার কুণ্ডলীর মধ্য দিয়ে বিদ্যুৎ প্রবাহ শুরু হয়। এই বিদ্যুৎ প্রবাহের জন্য কুণ্ডলীর যে অংশে প্যাঁচানো, তার ওপর দুটি সমান, সমান্তরাল ও বিপরীতমুখী বল কাজ করতে শুরু করে। এই বলের কারণে টর্ক বা বলের ভ্রামক প্রাপ্ত হয় এবং কুণ্ডলীটি ঘুরতে শুরু করে। মূলত এই তত্ত্বের ওপর ভিত্তি করে মোটর তৈরি করা হয়ে

থাকে। ডিসি মোটর লাইন ফলোয়িং রোবট থেকে শুরু করে অসংখ্য কাজে ব্যবহার করা হবে।

ডিসি মোটরে ব্রাশ থাকতে পারে আবার না-ও থাকতে পারে। ব্রাশ ছাড়া ডিসি মোটরকে ব্রাশলেস ডিসি মোটর বলে। ব্রাশলেস ডিসি মোটরকে সিঙ্ক্রোনাস ডিসি মোটরও বলা হয়ে থাকে। এই মোটরে একধরনের ইনভারটার থাকে যা এসি কারন্টকে ডিসি কারেন্টে পরিণত করে থাকে। দুটি প্যারামিটার দ্বারা এই মোটরের পারফরম্যান্স বোঝা যায়— ধ্রুব টর্ক আর ব্যাক ইএমএফ (পরে আলোচিত)। এই মোটরের একটা উল্লেখযোগ্য ব্যবহার হলো—বিভিন্ন ধরনের ড্রোনে এই মোটর ব্যবহার করা হয়।

ব্রাশড মোটরের চেয়ে ব্রাশলেস মোটরে জটিলতা কম। কারণ এটি ঘোরানোর জন্য বাইরে থেকে ইনপুট কারেন্ট দেওয়ার ঝামেলা কম। এই মোটরগুলো দিয়ে অনেক দিন কাজ করা যায়, স্বল্প রক্ষণাবেক্ষণ এবং উচ্চকর্মদক্ষতা থাকে। ঝামেলা হলো দাম বেশি আর স্পিড কন্ট্রোল করা ঝামেলার। এর মজার দিক হলো, সাধারণ ডিসি মোটরে একটা নির্দিষ্ট ভোল্টেজের পর যত ভোল্টেজই বাড়ানো হোক না টর্কের মান আর বাড়ানো যায় না। কিন্তু ব্রাশলেস মোটরে এ রকম হয় না।

মোটর স্পেসিফিকেশন

মোটরের টর্ক

এবার একটু বলবিদ্যা ঘুরে আসি। পদার্থবিজ্ঞানে বলের ভ্রামককে টর্ক বলা হয়ে থাকে। যে বল ঘূর্ণন সৃষ্টি করে এবং ঘূর্ণনকেন্দ্র থেকে সেই বলের ক্রিয়ারেখার মধ্যবর্তী যে লম্ব দূরত্ব, এদের গুণফলকে গাণিতিকভাবে টর্ক বলে। টর্কের মান তিনটি বিষয়ের ওপর নির্ভর করে : প্রযুক্ত বল (F), ব্যাসার্ধ ভেক্টর (r) এবং বলের দিক ও ব্যাসার্ধ ভেক্টরের মধ্যবর্তী কোণ (θ) গাণিতিকভাবে—

$T = r \times F$; $|T| = r * f * \sin\theta$

T হলো টর্ক বা বলের ভ্রামক, r ঘূর্ণন অক্ষ থেকে বলের প্রয়োগবিন্দুর দূরত্ব বা ব্যাসার্ধ ভেক্টর,

F বস্তুর ওপর ক্রিয়ারত বল, θ দ্বারা F এবং τ এর মধ্যবর্তী কোণ

এখানে যেহতু দুটি বিপরীতমুখী বলের ক্রিয়ারেখার কারণে একধরনের দ্বন্দ্বের সৃষ্টি হয়, এই দ্বন্দ্বের ভ্রামক টর্ক আরমেচারকে ঘোরাতে চেষ্টা করে। আর আমরা জেনেছি সহজ ভাষায় এভাবেই মোটর চালিত হয়ে থাকে।

মোটর নিয়ে কাজ করতে হলে মোটরে টর্ক মান সম্পর্কে জানা প্রয়োজন। ডিজাইনকৃত রোবটটি ওজনে বেশ ভারী হলে রোবটটিকে চালাতে হাই টর্ক মোটরের প্রয়োজন, কারণ অধিক টর্ক মোটরটিকে সামনের দিকে নিয়ে যাওয়ার জন্য অধিক শক্তি প্রদান করে থাকে।

মোটর স্পিড

মোটরে যে গতি, সেটি মূলত কৌণিক গতি। তাই মোটরের গতি সম্পর্কে জানতে হলে কৌণিক বেগ জানা প্রয়োজন। সাধারণত এই কৌণিক বেগকে আরপিএম দ্বারা প্রকাশ করা হয়। সহজ ভাষায়, এক মিনিটে মোটর যতগুলো পূর্ণ ঘূর্ণন সম্পন্ন করে, সেই সংখাকে ঘূর্ণন প্রতি মিনিট বা আরপিএম দ্বারা প্রকাশ করা হয়।

রোবটকে দ্রুতগামী করতে হলে আরপিএমের মান বেশি থাকা প্রয়োজন। তবে সেই সাথে পাওয়ার সাপ্লাই এবং ভারসাম্যের সাথে সামঞ্জস্যা রাখা জরুরি।

এখানে একটি নির্দিষ্ট ভোল্টেজর জন্য, মোটরের স্পিড এবং মোটরের টর্ক ব্যস্তানুপাতিক। আরেকটু পরিষ্কার করি, গাণিতিকভাবে, $P=T*\omega$। একটা নির্দিষ্ট ভোল্টেজের জন্য ইনপুট কৃত পাওয়ার ধ্রুব থাকে। তাই বলা যায়, অর্থাৎ টর্ক বাড়লে স্পিড কমে যাবে।

স্টেপার মোটর

স্টেপার মোটর বিশেষ ধরনের মোটর। এই মোটরের বিশেষত্ব হলো- মোটরটি একটি পূর্ণ ঘূর্ণনকে অনেকগুলো অংশে ভাগ করতে পারে নির্দিষ্ট ধাপ অনুসারে। মোটরটির অন্যতম বিশেষত্ব হলো, মোটরের অবস্থান নির্ণয়ের জন্য কোনো হল ইফেক্ট বা অন্য সেন্সরের প্রয়োজন হয় না। স্টেপার মোটরের গঠনে ডিসি মোটরের থেকে খুব বেশি পার্থক্য নেই। তবে আমরা অনেকেই বিভিন্ন খেলনা যন্ত্র বা গাড়িতে গিয়ার দেখেছি। গিয়ার খাঁজকাটা থাকে, যাকে বলা হয় গিয়ার দাঁত। স্টেপার মোটরে এ

রকম খাঁজকাটা তড়িৎ চুম্বক থাকে। এই অংশে বিশেষ ব্যবস্থা দ্বারা স্টেপ বা ধাপ নির্ণয় করা হয়।

স্টেপার মোটরে পুরো সিস্টেমকে সহজভাবে বোঝাতে তিন ভাগে ভাগ করা যেতে পারে। যেমন

১. নিয়ন্ত্রণকারী ইউনিট : মাইক্রো কন্ট্রোলার থেকে নির্দেশনা অনুসারে এখানে সিগন্যাল অথবা পালস তৈরি করা হয়।

২. অ্যাম্পিলিফাইয়ার : এই অংশকে মোটর ড্রাইভারও বলা যায়। সিগন্যাল অনুযায়ী এই অংশ মোটরে পাওয়ার সাপ্লাই করে।

৩. মোটর : এই অংশে সিগন্যাল অনুসারে মোটরের রোটর ঘূর্ণন সম্পন্ন করে।

স্টেপার মোটর বিভিন্ন ধরনের হয় থাকে। স্টেপার মোটরে দুই বা ততোধিক ফেজ থাকতে পারে। এ ধরনের মোটর মূলত আমরা কোনো সিস্টেমের সাথে সংযুক্ত বস্তুর অবস্থান নিয়ন্ত্রণের জন্য ব্যবহার করব।

পাওয়ার সাপ্লাই

কোনো বর্তনীকে চালনার জন্য একটা শক্তির উৎস লাগে। আর পাওয়ার সাপ্লাই এমন একটা যন্ত্রাংশ যা বর্তনীতে বিদ্যুৎ শক্তি সরবারহ করে থাকে। পাওয়ার সাপ্লাইকে বর্তনীর প্রাণ বলা হয়ে থাকে। কারণ এর দ্বারাই কোনো বর্তনী সচল হয়ে থাকে। পাওয়ার সাপ্লাই অনেক রকমের হতে পারে। আমরা এখানে মূলত ডিসি পাওয়ার সাপ্লাই নিয়েই কথা বলব। ডিসি পাওয়ার সাপ্লাই একধরনের বিভবের উৎস বা ভোল্টেজ সোর্স, যার মাধ্যমে একটা নির্দিষ্ট ভোল্টেজ বা বিভব আমরা বর্তনীতে সরবরাহ করতে পারব। ডিসি পাওয়ার সরবরাহ আমরা দুভাবে করতে পারি—

১. সরাসরি উৎস থেকে। যেমন বিভিন্ন ধরনের ব্যাটারি।

২. কনভারটারের মাধ্যমে।

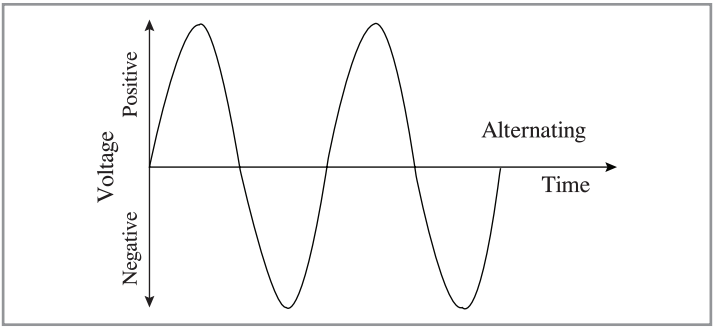
আমরা রোবটিকস নিয়ে কাজ করার সময় বিভিন্ন ব্যাটারি নিয়ে কাজ করব। সেই ব্যাটারিগুলো নিয়ে একটু বলি।

সাধারণ ডিসি ব্যাটারি

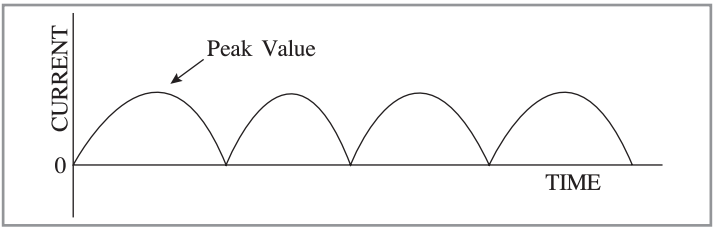
আমাদের হাতের কাছেই এ ব্যাটারিগুলো পাওয়া যায়। আমরা দৈনন্দিন ব্যবহার করা ইলেকট্রনিক যন্ত্রাংশের জন্য এই ব্যাটারিগুলো ব্যবহার করে থাকি। ডিসি ব্যাটারিগুলো বিভিন্ন বিভবের পাওয়ার সাপ্লাই করে থাকে। এটা সুবিধামতো বাচাই করে নেওয়া যায়।

যেসব প্রজেক্টের কন্ট্রোল করার অংশ স্থির, যেমন রোবটিক হ্যান্ড, সেখানে ব্যাটারি বা সেল ব্যবহার করা বুদ্ধিমানের কাজ নয়। কারণ ব্যাটারি ব্যয়বহুল। এসব ক্ষেত্রে আমরা এসি টু ডিসি অ্যাডাপ্টার ব্যবহার করতে পারি। তবে যদি ইলেকট্রনিক্যাল কম্পোনেন্ট নিয়ে কাজ করার আসল অনুভূতিটা পাওয়ার জন্যই আমরা এবার আলোচনা করব কীভাবে ঘরে বসেই এটা বানানো যায়।

শুরু করার আগে একটা ব্যাপার একটু মনে করিয়ে দিই। আমরা কমবেশি রেক্টিফিকেশন সম্পর্কে শুনেছি। আমরা বাসাবাড়িতে যে কারেন্ট ব্যবহার করি, এগুলো দিক পরিবর্তী প্রবাহ। বাংলাদেশের প্রবাহ বা কারেন্ট এক সেকেন্ডে ৫০ বার দিক পরিবর্তন করে। এই কারেন্ট সরাসরি যন্ত্রে ব্যবহার করলে কী হতে পারে ভাবতেই শিউরে উঠতে হয়। তাই এই দিক পরিবর্তী প্রবাহ যন্ত্রাংশগুলোর উপযোগী করে একমুখী করতে হবে। এই একমুখী করার প্রক্রিয়াকে বলা হয় রেক্টিফিকেশন, আর আমরা যে বর্তনীটি তৈরি করব, সেটাকে রেক্টিফায়ার বলে। রেক্টিফায়ার দুই ধরনের হয়। তবে আমরা এখানে যে রেক্টিফায়ার বানাব তা পূর্ণ তরঙ্গ রেক্টিফায়ার। যাহোক প্রথমে একটু তত্ত্ব কথা বলে নিই। আমরা ত্রিকোণমিতি নিয়ে যারা ঘাঁটাঘাঁটি করেছি, তারা সাইন ফাংশনের লেখচিত্র সম্পর্কে জানি। আমরা এখানে রেক্টিফায়ার বানাতে যে দিক পরিবর্তী প্রবাহ ইনপুট হিসেবে দেব, তাকে একধরনের সাইন ফাংশনরূপে প্রকাশ করা যায়। আর আমরা রেক্টিফিকেশনের পর যেটা আউটপুট হিসেবে পাব, সেটা একধরনের পালসেটিং (ক্রমশ পরিবর্তনশীল) ডিসি বা সমপ্রবাহ। এখানে ইনপুট হিসেবে এসি ভোল্টেজ এবং আউটপুট হিসেবে ডিসি ভোল্টেজ দেখানো হয়েছে।

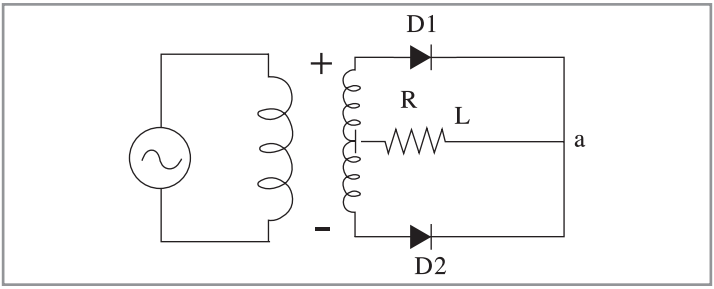


চিত্র ১.১০ : ইনপুট— এসি



চিত্র ১.১১ : আউটপুট— পালসেটিং ডিসি প্রবাহ

এবার আমরা একটু পূর্ণ তরঙ্গ রেক্টিফায়ার বর্তনীর দিকে তাকাই—

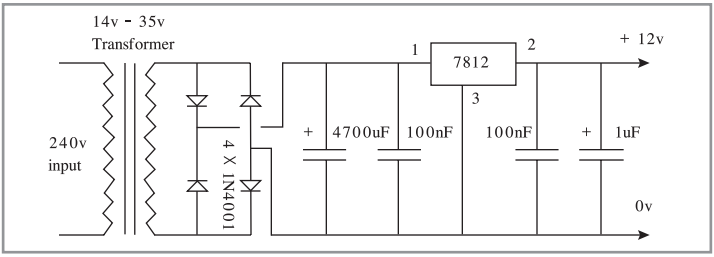


চিত্র ১.১২ : পূর্ণ তরঙ্গ রেক্টিফায়ার

এসি ইনপুটের প্রথম ধনাত্মক অর্ধচক্রের জন্য ডি১ ডায়োডের মধ্য দিয়ে বিদ্যুৎ পরিবাহিত হয় কিন্তু ডি২ ডায়োড এই বিদ্যুৎ প্রবাহে বাধাদান করে। আবার ঋণাত্মক অর্ধচক্রের জন্য ডি২ ডায়োড বিদ্যুৎ প্রবাহিত করে কিন্তু ডি১ বাধা দেয়। এভাবেই পালসেটিং ডিসি পাওয়া যায়। এখানে পূর্ণ চক্র নিয়ে কাজ করা হয় বলে একে পূর্ণ তরঙ্গ রেক্টিফায়ার বলে।

এবার আমরা সরাসরি চলে যাব কীভাবে নির্দিষ্ট ১২ ভোল্টের একটা পাওয়ার সাপ্লাই তৈরি করব।

প্রথমে ধারণা দেওয়ার জন্য ব্লক ডায়াগ্রামটা দেখানো হলো—



চিত্র ১.১৩ : ১২ ভোল্টের পাওয়ার সাপ্লাইয়ের ব্লক ডায়াগ্রাম (www.instructables.com)

পাওয়ার সাপ্লাই তৈরি করতে আমাদের যেসব কম্পোনেন্টের প্রয়োজন হবে—

১. স্টেপ ডাউন ট্রান্সফরমার—

Rating :

Input 220 A/C volt

Output- 12 Volt/3 amp

২. রেক্টিফায়ার ডায়োড—

1N4007/1N4001

৩. ক্যাপাসিটর—

১. 1000 μF থেকে 4700 μF পর্যন্ত যেকোনো মানের রেটেড ক্যাপাসিটর

২. 100 μF

৩. 1 μF

৪. রেগুলেটর (ভোল্টেজ রেগুলেট করবে) : LM7812 regulator

৫. জাম্পার তার

৬. ব্রেডবোর্ড

আমরা কানেকশন চেকের জন্য মাল্টিমিটার অবশ্যই ব্যবহার করব এবং কয়েকটি এলইডি ব্যবহার করতে পারি সার্কিট কানেকশন অন/ অফ দেখানোর জন্য।

স্পেশাল নোট : ১২ ভোল্ট আউটপুটের জায়গায় ৫ ভোল্ট আউটপুট চাইলে LM7805

ব্রেড বোর্ড

নামটা শুনে অনেকে হয়তো অবাক হবে। অনেকের কাছে সম্পূর্ণ নতুন হলেও কিছুক্ষণ পরেই জানা যাবে এই কম্পোনেন্ট কত কাজের। ব্রেড বোর্ড আমাদের সোল্ডারিং করার যন্ত্রণা থেকে রেহাই দেবে এবং সার্কিটের বিভিন্ন অংশে পাওয়ার সাপ্লাইয়ের সুযোগ করে দেবে।

ব্রেড বোর্ড একধরনের প্লাস্টিক বক্স, যেখানে অনেকগুলো ক্লিপ থাকে।

ব্রেড বোর্ডে দুটি অংশ থাকে—

১. বাস স্ট্রিপ

২. সকেট/টারমিনাল স্ট্রিপ

ব্রেড বোর্ডের মাঝখানে ডিআইপি সাপোর্ট বলে একটি অংশ থাকে।

বাজারে যেসব ব্রেড বোর্ড পাওয়া যায়, পাশে দেখা যাবে দুটি কলাম আছে, ওপরে পজেটিভ ও নেগেটিভ চিহ্ন দেওয়া আছে—ওই অংশটি বাস স্ট্রিপ। জাম্পার তারের সাহায্যে আমরা যদি ব্যাটারির ধনাত্মক ও ঋণাত্মক প্রান্তের সাথে সংযোগ দিই, বাস স্ট্রিপ পাওয়ার সাপ্লাইয়ের কাজ করবে।

এবার আসি টারমিনাল স্ট্রিপ প্রসঙ্গে। আমরা আগেই জেনেছি ব্রেড বোর্ড অনেকগুলো ক্লিপের সমন্বয়ে গঠিত। খুব সহজেই বুঝতে পারছি যে এই ক্লিপের প্রতিটি অংশ শর্ট করা। অর্থাৎ প্রতিটি অংশের ভোল্টেজ একই। এই ক্লিপগুলো অনুভূমিক বরাবর বসানো থাকে, তাই ব্রেড বোর্ডের অনুভূমিক বরাবর প্রতিটি পয়েন্ট শর্ট করা থাকে। শুধু ক্লিপের এক মাথায় তার যোগ করে, আরও অনেকগুলো তার ওই সারির ক্লিপের মাথায় যুক্ত করা যাবে।

আমরা এই বইয়ের বিভিন্ন অংশে রোবটের প্রটোটাইপ তৈরি করার জন্য ব্রেড বোর্ড ব্যবহার করব।

পাসওয়ার্ড দিয়ে দরজা খোলার দৃশ্য নিশ্চয়ই আমরা অনেক মুভিতে দেখেছি।

এ রকম কিছু নিজে তৈরি করলে কী রকম লাগতে পারে? নিশ্চয়ই ভালো লাগবে। শুধু তা-ই না। আমরা এখন যে কম্পোনেন্ট দুটি নিয়ে আলোচনা করব, সেটা আমাদের প্রটোটাইপ রোবটে তৈরির কাজে সাহায্য করবে।

এলসিডি ডিসপ্লে (LCD)

আমরা অনেকেই হয়তো ক্যালকুলেটরের এলইডি ডিসপ্লের সাথে পরিচিত। তাত্ত্বিকভাবে একই সাধারণ এলইডি ডিসপ্লে নিয়ে আলোচনা করব। তাত্ত্বিক আলোচনা কিছুটা জটিল হতে পারে অনেকের জন্য। তাই কীভাবে কাজ করব এ বিষয়ে আলোচনা করছি।

আমরা এখানে মূলত একটা এলসি ডিসপ্লে আরডুইনো প্ল্যাটফরম দিয়ে কীভাবে চালাতে হয় এ বিষয়টি শিখব।

প্রথমে এলসিডি পিনগুলো নিয়ে আলোচনা করছি: (১ নম্বর পিন থেকে ক্রমানুসারে—

১. Ground Pin (O volt) (GND/VSS pin) : ব্যাটারির ঋণাত্মক প্রান্তের সাথে যুক্ত করতে হবে।

২. Power supply pin (5V) (VCC pin) : ব্যাটারি বা পাওয়ার সাপ্লাইয়ের ধনাত্মক প্রান্ত যুক্ত করতে হবে।

৩. Contrast pin (VE pin) : কন্ট্রাস্ট বা ডার্কনেস কন্ট্রোল করা যাবে।

৪. রিড বা রাইট পিন (Read/Write Pin): প্রোগ্রাম কোডে লো ইন্সট্রাকশন দিলে ডিসপ্লে ডেটা রিড করে এবং প্রোগ্রাম কোডে হাই ইন্সট্রাকশন দিলে ডিসপ্লে ডেটা রাইট শুরু করে।

৫. Enable Pin (En pin) : রেজিস্টারগুলোয় ডেটা রাইট শুরু করে।

৬. 8bit DATA pins (D0-D7) : ডেটা পিনগুলো ডেটা সেন্ড করতে সাহায্য করে।

৭. LED + pin

৮. LED - pin

পিনগুলো নিয়ে তো আলোচনা হলো এবার প্রজেক্টের কাজের জন্য যেসব কম্পোনেন্ট প্রয়োজন তা দেওয়া হলো—

১. আরডুইনো

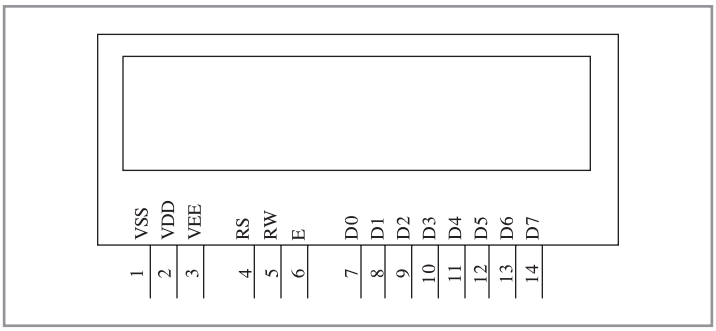
২. breadboard

৩. 16*2 এলসিডি ডিসপ্লে

৪. পিন

৫. জাম্পার তার

এই মডিউলে আলাদা করে পিন সোল্ডারিং করে নিতে হবে। সোল্ডারিং নিয়ে আগে কথা বলা হয়েছে।



চিত্র ১.১৪ : এলসিডি

এবার আসি কীভাবে এলসিডি মডিউলে এবং আরডুইনোর সাথে যে পিনগুলো জাম্পার দিয়ে সংযোগ দিতে হবে। পিন কানেকশন নিচে দেওয়া হলো—

ভিএসএস পিন : ব্যাটারি বা পাওয়ার সাপ্লাইয়ের ঋনাত্মক প্রান্ত বা গ্রাউন্ডের সাথে সংযোগ দেবে।

ভিসিসি পিন : আরডুইনোর ৫ ভোল্ট অথবা ৫ ভোল্ট পাওয়ার সাপ্লাইয়ের সাথে সংযোগ দেবেন

VO pin — Digital PWM pin 6~

RS pin — Digital PWM 11~

R/W — ground(GND)

En	—	Digital PWM Pin 12~
DO	—	D7-
D7	—	Digital pin2
D6	—	Digital PWM pin 3~
D5	—	Digital Pin 4
D4	—	Digital PWM Pin 5
LED+	—	Digital Pin 10
LED-	—	ground (GND)

প্রোগ্রামিং অংশ—

```
#include <LiquidCrystal.h> // এলসিডি লাইব্রেতি থেকে ইম্পোরট করা হয়েছে।
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); /*এলসিডি লাইব্রেরি শুরু করা হলো এবং
যেসব পিন দিয়ে কানেকশন তৈরি করা হয়েছে তা দেওয়া হলো
#define contra 6 //খেয়াল করলে দেখবে যে আমরা কন্ট্রাস্ট কন্ট্রোল করার জন্য পিন
৬ ব্যবহার করেছি
#define bri 10 //এলইডি লাইটের ব্রাইটনেস কমানো-বাড়ানো নিয়ন্ত্রণের জন্য এটি
ব্যবহার হয়।

void setup()

 lcd.begin(16, 2); //যেহেতু 16x2 LCD, তাই আরডুইনোকে সেভাবে ইন্সট্রাকশন
দেওয়া হচ্ছে
  pinMode(contra, OUTPUT); // আরডুইনো ৬ নম্বর পিনে ডেটা আউটপুট দেবে
  pinMode(bri, OUTPUT); // আরডুইনো ১০ নম্বর পিনে ডেটা আউটপুট দেবে

  digitalWrite(contra, LOW); //আপাতত কন্ট্রাস্ট পিনে পাওয়ার অফ
  analogWrite(bri, 255); // পাওয়ার কন্ট্রোল করা যাবে
}
//আসল কোড এখান থেকে শুরু
void loop() {
lcd.print("!!"); //আমরা আউটপুট হিসেবে এই বাক্যটা দেখতে চাই
  lcd.setCursor(0, 1); // একটা অদৃশ্য কার্সর রাখা হল
  lcd.print("");// দ্বিতীয় লাইন এ বাক্যটি আউটপুট হিসেবে দেখা যাবে
  delay(1000); // এক সেকেন্ড অপেক্ষা করতে হবে
//এখন আমরা এই অংশ নিয়ে আরও কিছু করতে চাই। আরডুইনো প্রোগ্রামিং সম্পর্কে
আশা করি কম বেশি ধারণা আপনাদের নিশ্চয়ই হয়েছে। এবার আমার এই অংশে
লুপের ব্যবহার করে আরও মজার একটি কাজ করব
```

```
for (int  l = 0; l < 16; l++) {
    lcd.scrollDisplayRight(); //স্ক্রিনটা এবার ডান দিকে সরে যাবে, এ জন্য ফাংশন ব্যাখ্যা বা ডিফাইন করা হলো
    delay(200); //কিছুটা আস্তে করা হলো
  }
  delay(1000);// আরেকটু সময় দেওয়া হলো
  for (int l = 0; l < 16; l++)
    lcd.scrollDisplayLeft();// ঠিক আগের মতোই একটা ফাংশন
  for (int l = 51; l > -1; l--) { //এ কাজটা ১৫ বার পুনরায় করতে চাই।
    analogWrite(bri, l * 5);// অ্যানালগ রাইট করে ইচ্ছেমতো ব্রাইটনেস কম করা হচ্ছে
    delay(35);
  }
  delay(1000);
  digitalWrite(bri, HIGH);
  for (int l = 0; l < 51; l++) { //Repeating 51 times
    analogWrite(contra, l * 5);// অ্যানালগ রাইট করে ইচ্ছেমতো ব্রাইটনেস কম করা হচ্ছে
    delay(35);
  }
  delay(1000);
  digitalWrite(contra, LOW);
  lcd.setCursor(0, 0);
}
```

সোল্ডারিং কী?

সোল্ডারিং একধরনের বিশেষ পদ্ধতি যার সাহায্যে দুই বা ততোধিক ধাতুকে যুক্ত করে সংযোগ সৃষ্টি করা হয় এবং এ কাজের জন্য সাহায্যকারী হিসেবে একটি ফিলার ধাতব (পরিচিত নাম) উপাদান ব্যবহৃত হয় যাকে গলে যাওয়ার পর অবস্থায় যেখানে সংযোগ দেওয়া হবে, সেখানে প্রবাহের মাধ্যমে সংযোগ সৃষ্টি হয়।

সোল্ডারিং করতে যেসব যন্ত্রপাতি ও কাঁচামাল দরকার তা নিচে দেওয়া হলো—

১. সোল্ডারিং আয়রন

২. ফিলার ম্যাটেরিয়াল বা সোল্ডার

৩. সোল্ডারিং ফ্লাক্স (পেস্ট অথবা গাম রজন)

যেভাবে সোল্ডারিং করতে হবে

যেখানে সোল্ডারিং করতে হবে সেই জায়গা আগে ঠিক করে ফেলতে হবে। তারপর সোল্ডারিং আয়রনকে প্রয়োজন মতো গরম হতে দিতে হবে। এবার গরম হওয়ার পর ফিলার মেটালটি যে জায়গার কাছে সোল্ডারিং করব, সেখানে নিয়ে যেতে হবে। আয়রন স্পর্শ করার পর দেখা যাবে, ফিলার মেটালটা গলে যাবে আর এবং যেখানে সংযোগ দিতে চাই, সেখানে সংযোগ পেয়ে যাব।

মাল্টিমিটার

মাল্টিমিটার দুই ধরনের হয়ে থাকে— ডিজিটাল মাল্টিমিটার আর অ্যানালগ মাল্টিমিটার। আমরা এখানে ডিজিটাল মিটার নিয়ে কথা বলব, যেখানে সরাসরি ভ্যালু স্ক্রিনে রিডিং পাওয়া যায়।

মাল্টিমিটারের মাঝখানে আছে নব সিলেক্টর। যেটা দিয়ে আমরা নির্দেশনা দেব কীভাবে রিডিং পেতে হয়। । যেমন ভোল্টেজ (V), কারেন্ট (A), রেসিস্টেন্স (Ω) রিডিং ইত্যাদি। সাথে আরও আছে শর্ট টেস্ট আর ট্রানজিস্টর টেস্টের জন্য আলাদা আলাদা ব্যবস্থা।

চিত্র ১.১৫ : ডিজিটাল মাল্টিমিটার

মিটারের রেঞ্জ

রেঞ্জ বলতে এখানে মূলত কতটুকু পর্যন্ত মাপতে পারব সেটা বোঝায়। ধরা যাক রেসিস্টরের ভ্যালু মাপতে নব সিলেক্টরকে 10KΩ এ সিলেক্ট করা হলো। তাহলে 0 থেকে 10KΩ-এর ভেতর যেকোনো ভ্যালুর জন্য মিটারে রিডিং দেখাবে। এর চেয়ে বড় ভ্যালুর রেসিস্টর মাপতে গেলে

আমরা মাপতে পারব না এবং Out of range দেখাবে। তখন রেঞ্জটাকে আরেকটু বাড়িয়ে নিতে হবে। ভোল্টেজ মাপার আগে অবশ্যই প্রথমে লাল প্রোবটিকে VΩ আর কালোটিকে COM চিহ্নিত স্থানে বসাতে হবে। এরপর পাওয়ার অন করে নিতে হবে।

ডিসি ভোল্টেজ টেস্ট

প্রথমে ভোল্টেজের রেঞ্জ ঠিক করে নিতে হবে। +5 Volt পাওয়ার সাপ্লাই দেওয়া সার্কিটের জন্য 20V রেঞ্জ রাখতে হবে। এরপর যে দুটো পয়েন্টের মধ্যে ভোল্টেজ মাপতে হবে সেখানে প্রোব ধরা লাগবে।

এসি ভোল্টেজ টেস্ট

আগের মতোই কাজ করতে হবে, শুধু এবার রেঞ্জ সিলেক্ট করে নেবেন। (V~) এই জায়গার প্রয়োজনীয় রেঞ্জ।

রেসিস্টর টেস্ট

প্রথমে Ω লেখা জায়গায় নিয়ে মাঝামাঝি একটা রেঞ্জ সিলেক্ট করতে হবে। এরপর রেসিস্টরের দুই মাথা দুই প্রোবে ধরে ভ্যালু দেখতে হবে। ভ্যালু খুব ছোট হলে রেঞ্জ কমাতে হবে, Out of range এমন কিছু দেখবে। তারপর রেঞ্জ বাড়াতে হবে।

ডিসি কারেন্ট টেস্ট

রেঞ্জ অবশ্যই বেশি নিতে হবে। 2mA রেঞ্জে রেখে 200mA মাপলে মিটারের ক্ষতি হতে পারে, এমনকি মিটার নষ্ট হয়ে যেতে পারে। পদার্থবিজ্ঞান পড়ার দৌলতে আপনার নিশ্চয়ই জানা আছে, রেসিস্টর আর ভোল্টেজ মাপা হয়েছিল প্রোবকে কম্পোনেন্টের প্যারালাল ধরে, কিন্তু কারেন্ট মাপতে গেলে প্রোবকে সিরিজে কানেকশন দিতে হয়।

শর্ট সার্কিট টেস্ট

খুব কাজের একটা টেস্ট! যে নবটাতে ডায়োড টেস্ট করে, সেটাই শর্ট সার্কিট টেস্ট করার জন্য ব্যবহার করা হয়। নবকে এই অবস্থানে রেখে প্রোব দুটো একসাথে লাগিয়ে মিটারের LED জ্বলে ওঠার সাথে সাথে শব্দ হতে থাকবে।

দ্বিতীয় অধ্যায়

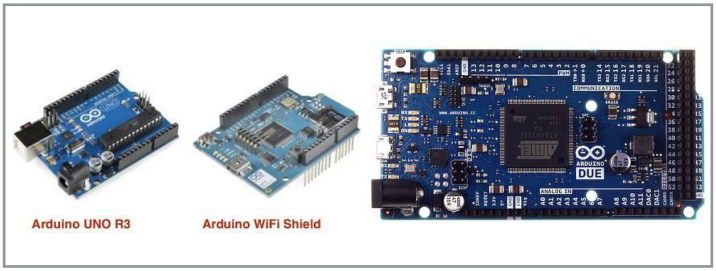
আরডুইনো ও মাইক্রোকন্ট্রোলার প্রোগ্রামিংয়ের বেসিক

আরডুইনো নিয়ে হয়তোবা কমবেশি সবাই জানি। বিশ্ববিদ্যালয় পর্যায়ে মোটামুটিভাবে এটা নিয়ে বেশি কাজ হয়। আরডুইনো কি তার আগে শুধু এক লাইনে এর উৎপত্তি সম্পর্কে বলা যাক।

আরডুইনো একটি মাইক্রোকন্ট্রোলার ডিভাইস। উৎপত্তি ইতালি শহরে। মূলত মাইক্রোকন্ট্রোলারের সাথে কম্পিউটারের সংযোগের জন্য আরও অনেক ডিভাইস যোগ করতে হয়। এটা অনেকটাই সমস্যা। এটা দূর করতেই আরডুইনোর উৎপত্তি।

আগে মাইক্রোকন্ট্রোলার সিস্টেমে যেটা করতে হতো যে atmega8p, atmega328p এক্সট্রা ব্যবহার করতে হতো। মূলত কম্পিউটার থেকে বাইনারি প্রোগ্রাম আপলোড করতেই এটার ব্যবহার। আর সাথে সংশ্লিষ্ট বর্তনী তৈরি করতে হতো। পরবর্তী সময়ে আরডুইনোর উৎপত্তি হওয়ায় এ সমস্যা আর রইল না।

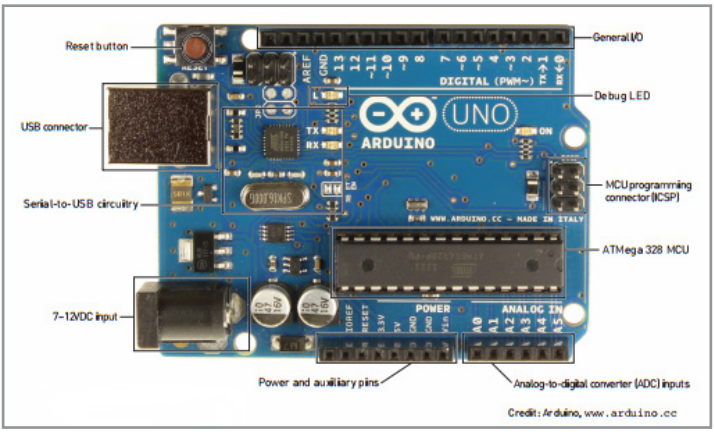
ছোট একটা ডিভাইসে একইসাথে সব রকম প্যাকেজ এই আরডুইনোতে।



চিত্র ২.১ : আরডুইনোর কয়েকটি মডেল

আরডুইনোতে মূলত কয়েকটি প্রধান অংশ।

এগুলো দেখানো হলো সাথে ছোট একটু বর্ণনা।



চিত্র ২.২ : আরডুইনো ইউএনও

১. USB-Connector : এই অংশের কাজ প্রধানত কম্পিউটারের সাথে সংযোগ করা। মূলত কম্পিউটার থেকে কোড আপলোড করা। আর এরপরে সে অনুযায়ী কাজ করা।

২. General-Io : এই অংশে মূলত ডিজিটাল ইনপুট আর আউটপুট সিস্টেম। সে সম্পর্কে পরে আলোচনা করা হবে। এই অংশে আউটপুট মূলত ১ আর ০। এখানে ০ এর জন্য ০ ভোল্ট আর ১-এর জন্য ৫ ভোল্ট।

৩. Analog to Digital converter : মূলত সেন্সর থেকে ইনপুট নিতেই এটি ব্যবহার করা হয়। ০ থেকে ২৫৫ পর্যন্ত এটি নির্ধারিত হয়। ০ এর জন্য ০ ভোল্ট আর ২৫৫, ৫ ভোল্টের জন্য। আসলে এখানে ২৫৫ ডেসিমাল সংখ্যা একে বাইনারিতে পরিণত করে কাজ করা হয়। ২৫৫-এর বাইনারি মূলত ৮টা ১। যেটা একটা ৮ বিট মাইক্রোকন্ট্রোলারের রেজিস্টরকে পূর্ণ করে।

৪. Reset switch : একটা প্রজেক্টে কাজ করাতে গিয়ে কোনো কারণে যদি সমস্যা দেখা যায়, সে ক্ষেত্রে এই সুইচটা চাপ দিলে তখন নতুন করে আবার কাজ করা শুরু করে।

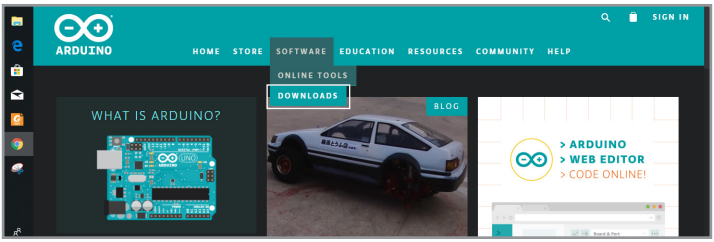
৫. Battery Port : এই পোর্টে মূলত ব্যাটারি সংযোগ দেওয়া হয়। ২০ ভোল্ট পর্যন্ত ইনপুট দেওয়া যায়।

মূলত একটা মাইক্রোকন্ট্রোলার অপারেশন করতে হলে আমাদের এই কয়টা জিনিস ভালোমতো জানতে হবে।

কিছু সফটওয়্যার বেসিক

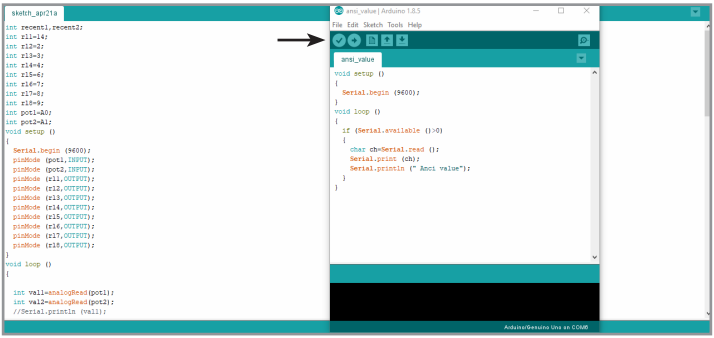
আমরা আরডুইনো নিয়ে কাজ করব। তার আগে আমাদের প্রধানত কয়েকটি বিষয় সম্পর্কে খুব ভালোমতো জানতে হবে।

১. সফটওয়্যার : আমরা আরডুইনো নিয়ে প্রোগ্রাম লিখে আপলোড করব। আর এ জন্য একটা সফটওয়্যার অবশ্যই দরকার। আরডুইনোর এই সফটওয়্যার মূলত তাদের ওয়েবসাইটেই পাওয়া যায়। www.arduino.cc এই ওয়েবসাইটে আরডুইনোর সফটওয়্যার পাওয়া যাবে। এখানে উল্লেখ্য যে এই প্রোগ্রামিং কোডটা সি প্রোগ্রামিং ভাষায় লিখতে হয়।



চিত্র ২.৩ : আরডুইনো সফটওয়্যার ডাউনলোড

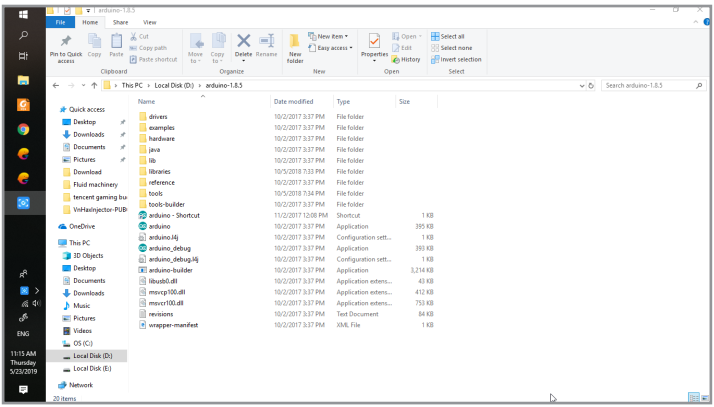
সফটওয়্যার ইন্টারফেস



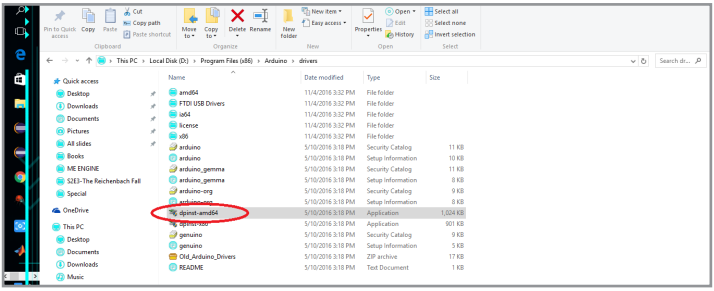
চিত্র ২.৪ : আরডুইনো সফটওয়্যার ইন্টারফেস

আরডুইনো সফটওয়্যারের ইন্টারফেস এ রকম। এখানে ওপরে টিক চিহ্ন দেওয়া GUI বাটনটি দিয়ে নির্ধারণ করা হয় যে লেখা কোডটি ঠিক আছে কিনা। এরপরে ডান পাশের তীর চিহ্ন দেওয়া বাটনটি দিয়ে প্রোগ্রাম আপলোড দেওয়া হয়। বাটনটির ওপরে মাউসের কার্সর রাখলেই বোঝা যাবে সেটি কোন কাজের জন্য।

২. ডিভাইস ড্রাইভার নিয়ে একটি কথা বলব, এটি একটি প্রয়োজনীয় ব্যাপার। প্রথম যারা কাজ করে তাদের ক্ষেত্রে দেখা যায় প্রোগ্রাম আপলোড দেওয়া হলেও আপলোড হয় না। বরং ভুল মেসেজ দেখায়। এটার কারণ একটাই। ডিভাইস ড্রাইভার। ডিভাইস ড্রাইভার একটা সফটওয়্যার যার মাধ্যমে একটা কম্পিউটার একটা ডিভাইস চিনতে পারে। তো আরডুইনো সফটওয়্যারের জিপ ফাইল অথবা ইএক্সই ফাইল ইনস্টল দিলে কোথায় আমরা ডিভাইস ড্রাইভার পাব, সেটাই প্রশ্ন। তো যে ফোল্ডারে ইনস্টল দাওয়া হয়েছে, সেই ফোল্ডারে যেতে হবে।



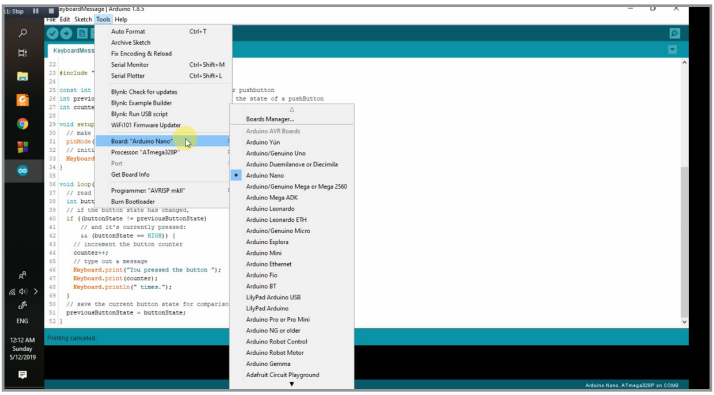
চিত্র ২.৫ : কম্পিউটারের ভিতর ফোল্ডার শনাক্তকরণ



চিত্র ২.৬ : dpints-amd64 শনাক্তকরণ

এখানে ছবিতে যে ফোল্ডারে ইনস্টল করা হয়েছে, সেখানে ড্রাইভারস ফোল্ডারে গেলেই বৃত্তাকার দাগ দেওয়া ২টি ইএক্সই ফাইলে ক্লিক করলেই হয়ে যাবে। dpinst-amd64 হচ্ছে ৬৪ বিট কম্পিউটারের জন্য আর dpinst-x86 ক্লিক করতে হবে ৩২ বিট কম্পিউটারের জন্য। ফোল্ডার লোকেশন Arduino > drivers। Arduino এখানে যে ফোল্ডারে ইনস্টল করা হয়েছে সে ফোল্ডারটি। এই ফোল্ডারে drivers অপশনে প্রবেশ করলেই পাওয়া যাবে।

৩. বোর্ড সিলেকশন করতে হয় যদি আমার লেখা কোড কোনো ডিভাইসে আপলোড দিতে চাই তখন কী টাইপ বোর্ডে আপলোড দেব তার ওপরে। বর্তমানে পাওয়া যায় সবচেয়ে বেশি বোর্ডগুলোর মধ্যে উল্লেখ্য হচ্ছে Arduino UNO, Arduino Mega, Arduino Nano, Arduino Mini, Arduino Leonardo। মেইন সফটওয়্যার ওপেন করে তাতে Tools অপশন ক্লিক করলেই তাতে Board অপশনে ক্লিক করতে হবে। এরপরে বোর্ড অপশনে গিয়ে আমরা কোন বোর্ডটা ব্যবহার করছি, সেটা ক্লিক করতে হবে। আর পোর্ট নির্বাচনে খেয়াল রাখতে হবে আমরা যে বোর্ডটা ব্যবহার করছি, সেটার পোর্ট কোনটা। পাশে একটা বন্ধনীতে লেখা থাকবে। সেটাতে ক্লিক করতে হবে।



চিত্র ২.৭ : বোর্ড সিলেকশন

সবশেষে বলব আপলোড দিতে গেলে তীর চিহ্ন দেওয়া অপশনে ক্লিক করতে হবে। তাহলেই আমার বোর্ডটা কাজ করা শুরু করবে।

কিছু ইলেকট্রনিকস বেসিক

আরডুইনো নিয়ে কাজ করার আগে আমাদের কিছু ইলেকট্রনিকস বেসিক সম্পর্কে জানতে হবে। রোবটিকস নিয়ে কাজ করার আগে এ ব্যাপারগুলো সম্পর্কে জানা উচিত। রোবটিকস নিয়ে কাজ করা মানে একটা রোবট বানাতে হলে আমাদের প্রধানত কয়েকটি কাজ করতে হবে।

১. সেন্সরের পাঠ নেওয়া। এই পাঠ থেকে প্রয়োজনীয় সিদ্ধান্ত গ্রহণ করা।

২. সিদ্ধান্ত থেকে আউটপুট বের করা। এই আউটপুট হতে পারে মোটরের গতি অথবা নিউম্যাটিক সিস্টেম অথবা হাইড্রোলিক সিস্টেমের মাধ্যমে।

সেন্সর থেকে পাঠ নেওয়া এটি হচ্ছে রোবটের সবচেয়ে গুরুত্বপূর্ণ ধাপ। কারণ মূলত এই ধাপের ওপরেই রোবটের কাজের নীতি নির্ভর করে। একটা রোবট আমাদের প্রয়োজনীয় কাজের জন্য কীভাবে কাজ করবে সেটা নির্ভর করে এই সেন্সরের ওপরেই। তাই সেন্সরের পাঠ যতটা ভালো হবে, ততটাই ভালো আউটপুট পাব আমাদের রোবট থেকে।

তো শুরু করা যাক।

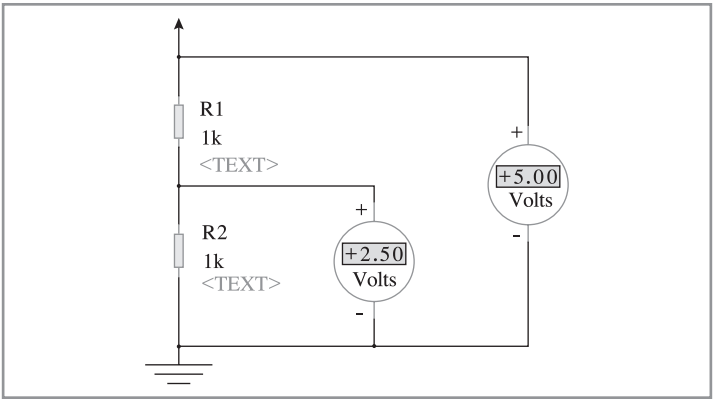
সেন্সরের পাঠ নেওয়া

অধিকাংশ সেন্সর পরিবর্তনশীল রোধক হিসেবে কাজ করে। সব সেন্সর রোধক না। কারণ কিছু রোধক তাদের ধারকত্ব অথবা ম্যাগনেটিক ফ্লাক্সের পরিবর্তনের ওপর নির্ভর করে। তবে অধিকাংশ সেন্সর, যেগুলো বাজারে পাওয়া যায়, সেগুলো পরিবর্তনশীল রোধক।

একটা বিষয় মাথায় রাখতে হবে। ভোল্টেজ ডিভাইডার সিস্টেম ব্যবহার সবচেয়ে প্রচলিত সিস্টেম।

তো ভোল্টেজ ডিভাইডার সিস্টেমটা কী রকম? সেটা প্রশ্ন

আমরা সবাই ছোট থাকতে এ ব্যাপারটা সম্পর্কে জেনেছি।



চিত্র ২.৮ : ভোল্টেজ ডিভাইডার সিস্টেম

এখানে ছবিতে আমরা খুবই সিম্পল একটা সার্কিট দেখতে পাচ্ছি। এটি ভোল্টেজ ডিভাইডার সিস্টেম। এখানে প্রতিটা রেসিস্টর পার করার সময় ভোল্টেজ ড্রপ করতে থাকে। এই সিম্পল টাইপ সার্কিট ব্যবহার করে আমরা একটা সেন্সর থেকে পাঠ নিতে পারি।

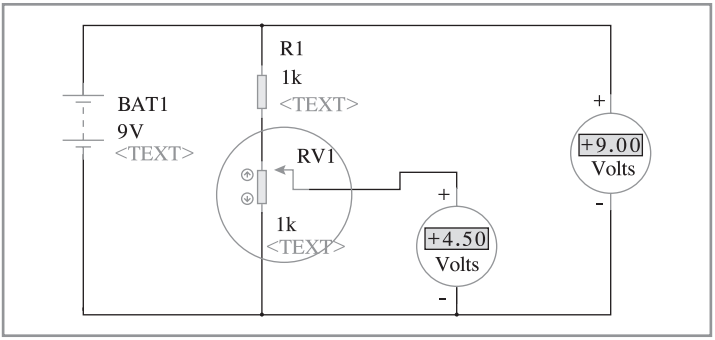
২টি রোধকের মান ১ ওহম করে হলে ইনপুট ভোল্টেজ ৫ ভোল্ট হলে আউটপুট ভোল্টেজ ২.৫ ভোল্ট। সহজ হিসাব।

উল্লেখ্য, একটা বিষয়, আমরা দেখি যে রোধকগুলোর মধ্যে একটি যদি পরিবর্তনশীল হয়, তবে সে ক্ষেত্রে দেখা যাবে যে ভোল্টেজ ড্রপটাও পরিবর্তনশীল হবে।

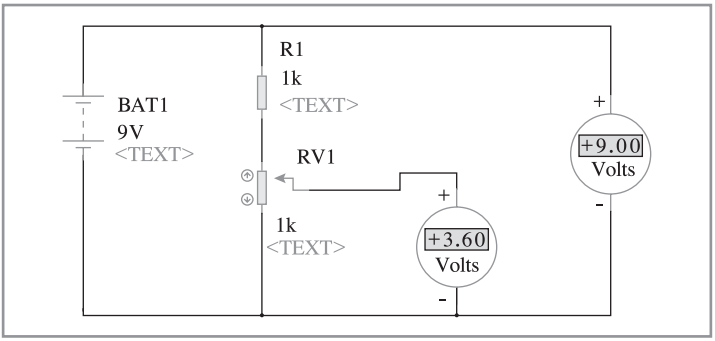
আর এটাই আমাদের সবচেয়ে বড় দরকারি বিষয় একটা সেন্সরের পাঠ নেওয়ার ক্ষেত্রে।

আমি আগেই বলেছিলাম যে অধিকাংশ সেন্সর এক একটা পরিবর্তনশীল রোধক। পরিবেশের তাপমাত্রা, চাপ, আর্দ্রতা মূলত আরও কয়েকটা পরিবর্তনশীল ব্যাপার এই সেন্সরের রোধ পরিবর্তন করে। কাজেই খুব সহজেই আমরা আন্দাজ করতে পারি বর্তনীটা কেমন হবে।

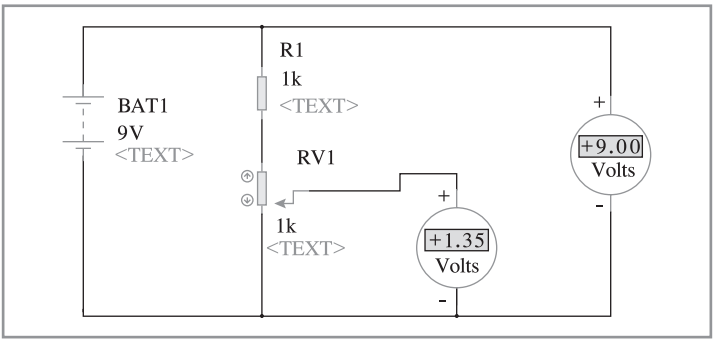
বর্তনীটা আগের ভোল্টেজ ড্রপের মতোই বর্তনীকাঠামো।



চিত্র ২.৯ : পরিবর্তনশীল রোধকের ডায়াগ্রাম ১



চিত্র ২.১০ : পরিবর্তনশীল রোধকের ডায়াগ্রাম ২



চিত্র ২.১১ : তীর চিহ্ন দেয়া অংশটি সরালে ভোল্টেজ পরিবর্তন হচ্ছে

এখানে বৃত্তাকার মার্ক করা অংশুটুকু হচ্ছে একটা পরিবর্তনশীল রোধক বা পটেনশিওমিটার (চিত্র ২.৯ দ্রষ্টব্য)। বলা চলে সেন্সর। এখানে তীর চিহ্ন দেওয়া অংশটুকু তার জায়গা পরিবর্তন করলে রোধকও পরিবর্তন

হয়। অনেকটা পটেনশিওমিটারের মতো। নব ঘোরালে রোধক পরিবর্তন হয়। ফলে ভোল্টেজ ড্রপও পরিবর্তন হতে থাকে। এখানে আমরা দেখতে পাচ্ছি তীর চিহ্নের অংশটি নিচে নামলে ভোল্টেজও কমতে থাকে। এই বর্তনীর ডিজাইনে অধিকাংশ রোধকের পাঠ নেওয়া সম্ভব। আরডুইনো দিয়ে কীভাবে পাঠ নেওয়া যায়, সেটা কোডিং অংশে দেখানো হয়েছে।

ট্রানজিস্টর

ট্রানজিস্টর সম্পর্কে এই বইয়ে আগেই বলা হয়েছে। এখন আমরা দেখব এটা কীভাবে ব্যবহার করতে হয়। ট্রানজিস্টর মূলত ২টি কাজে ব্যবহার করা যায়।

১. কোনো সিগনাল অ্যামপ্লিফিকেশন করতে বা বিবর্ধন করতে।

২. সুইচ হিসেবে।

সিগনাল অ্যামপ্লিফিকেশন করতে বর্তমানে ট্রানজিস্টরের চেয়ে অপারেশনাল অ্যামপ্লিফায়ার বেশি ব্যবহার হয়। সেটা কীভাবে ব্যবহার করতে হয় এটা ট্রানজিস্টরের পরেই বলছি। বর্তমানে এর আসল ব্যবহার সুইচ হিসেবে। অনেকে হাই ভোল্টেজের জন্য মসফেটও ব্যবহার করে।

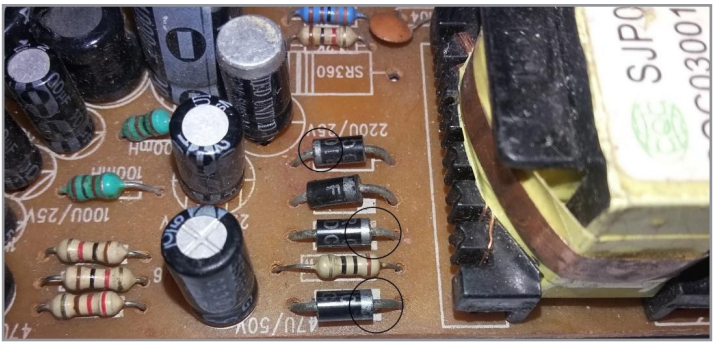
সুইচ হিসেবে: সুইচ হিসেবে ট্রানজিস্টরের ব্যবহার বর্তমানে অনেক বেশি। অপারেশনাল অ্যামপ্লিফায়ার সিগনাল বা সংকেত বিবর্ধনের কাজে বেশি ব্যবহার হয়। তো সুইচ হিসেবে কীভাবে এই ট্রানজিস্টর ব্যবহার করা যায়, সেটাই দেখানো হলো।

থিওরিটা খুব সহজ। ট্রানজিস্টরের এমিটার আর বেসের মধ্যে আমরা যদি বিদ্যুৎ প্রবাহ করি তাহলে এমিটার আর কালেকটরের মধ্যে বিদ্যুৎ প্রবাহ হবে।

আগে ডায়োড নিয়ে কিছু বলি। ডায়োডের ব্যাপারটা আলোচনা করা দরকার। কারণ একটা ট্রানজিস্টর ২টা ডায়োড দিয়ে গঠিত। আমরা জানি যে একটা ডায়োড p টাইপ আর n টাইপ এই ২ টাইপের সমন্বয়ে গঠিত। আমরা যদি p টাইপে ব্যাটারির ধনাত্মক আর n টাইপে ব্যাটারির ঋণাত্মক প্রান্ত সংযোগ করি, তবে বিদ্যুৎ প্রবাহ হবে। অর্থাৎ এটি একটি শর্টসার্কিট হিসেবে কাজ করবে। এর উল্টা সংযোগে বিদ্যুৎ প্রবাহ হবে না।

একটা ডায়োডের কোনটি ধনাত্মক আর কোনটি ঋণাত্মক, সেটি বোঝা যায় খুব সহজেই। একটা ডায়োড নিয়ে খুব ভালোভাবে লক্ষ করতে

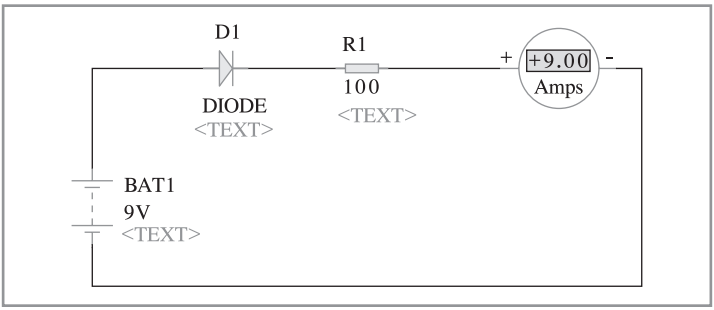
হবে। এর সাদা প্রান্তটি n টাইপ, মানে negative টাইপ অর্থাৎ এতে ব্যাটারির ঋণাত্মক প্রান্ত যুক্ত হবে। আর কালো প্রান্তটি p টাইপ, মানে positive টাইপ অর্থাৎ এতে ব্যাটারির ধনাত্মক প্রান্ত যোগ হবে।



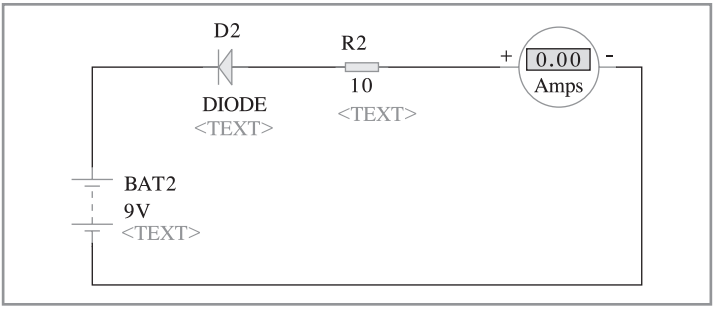
চিত্র ২.১২ : ডায়োডের নেগেটিভ প্রান্ত

এখানে কালো দাগ দেওয়া অংশুটুকুতে একটু সাদা দাগ দেখা যাচ্ছে। এটাই ডায়োডের ঋণাত্মক প্রান্ত।

ডায়োডের সংযোগে সংযোগ যদি ফরওয়ার্ড বায়াসে থাকে অর্থাৎ ব্যাটারির ধনাত্মক ডায়োডের ধনাত্মক আর ব্যাটারির ঋণাত্মক যদি ডায়োডের ঋণাত্মক হয়, তবে বিদ্যুৎ প্রবাহ হবে। না হলে হবে না। একটা মালটিমিটার দিয়ে পরীক্ষা করলেই এটা বোঝা যাবে। একটা বিষয় মনে রাখতে হবে ডায়োডের ফরওয়ার্ড বায়াস মানে তড়িৎ প্রবাহ আর এর উল্টা রিভার্স বায়াস, এতে কোনো তড়িৎ প্রবাহিত হবে না।



চিত্র ২.১৩ : ফরোয়ার্ড বায়াস

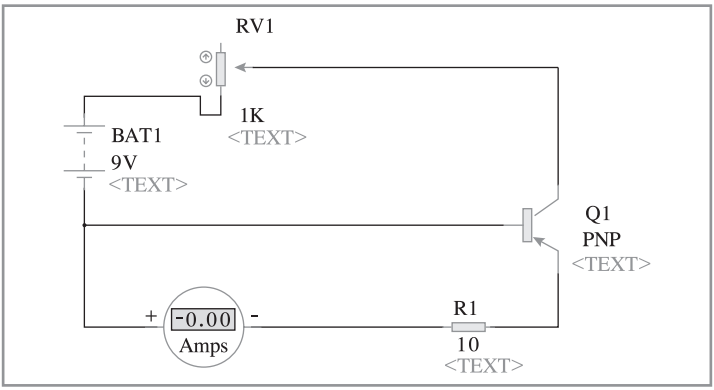


চিত্র ২.১৪ : রিভার্স বায়াস

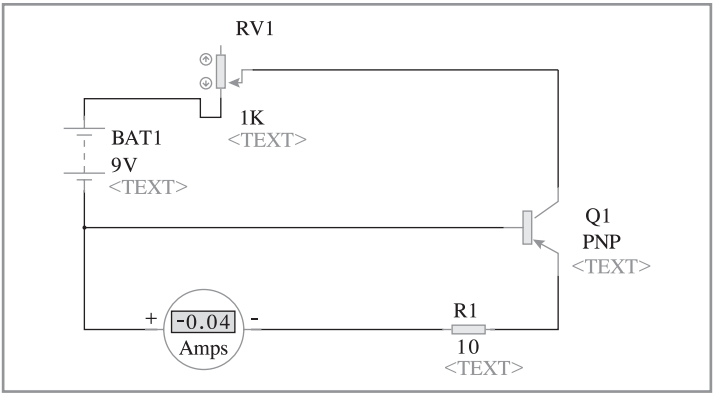
এখানে দেখা যাচ্ছে ফরওয়ার্ড বায়াসে একটা নির্দিষ্ট মাত্রার তড়িৎ প্রবাহিত হচ্ছে। আর উল্টাতে হচ্ছে না।

এবার আসি আমাদের ট্রানজিস্টরে

আমাদের যে কাজটা মূলত করতে হবে তা হচ্ছে ট্রানজিস্টরের এমিটার বেস ফরওয়ার্ড বায়াসে রাখতে হবে। এই বইয়ে এর আগেই ট্রানজিস্টরের এই ৩টি অংশ সম্পর্কে আলোচনা করা হয়েছে। এখন এমিটার বেসে আমরা যত বেশি পরিমাণে তড়িৎ প্রবাহিত করব, এমিটার কালেকটর অংশে তত বেশি তড়িৎ প্রবাহিত হবে। এভাবেই ট্রানজিস্টরের সুইচিং করা হয়। চিত্র ২.১৫ থেকে ব্যাপারটা পরিষ্কার হবে।



চিত্র ২.১৫ : সার্কিটে তড়িৎ প্রবাহ শূন্য



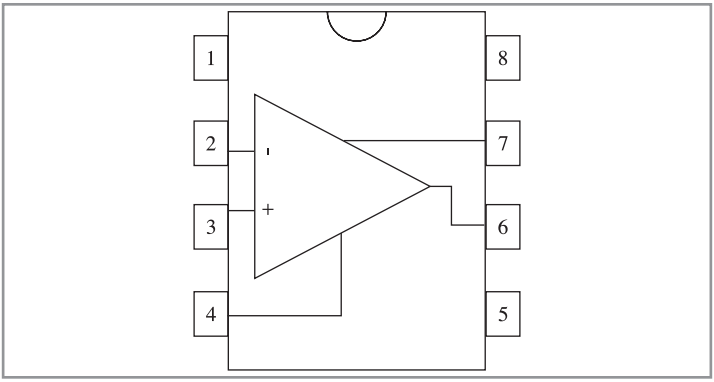
চিত্র ২.১৬ : পরিবর্তনশীল রোধকের উপর তীর চিহ্নের জায়গা পরিবর্তনে তড়িৎ প্রবাহর বৃদ্ধি

এখানে একটা পরিবর্তনশীল রোধক দেখা যাচ্ছে। তীর চিহ্নযুক্ত তারের জায়গা পরিবর্তন করলে রোধক বাড়ে বা কমে। এখানে শুরুতে তড়িৎ প্রবাহ এমিটার আর কালেকটরের মধ্যে ছিল ০। পরে জায়গা পরিবর্তন করায় রোধ কমে যায়। তড়িৎ প্রবাহ বাড়ে। ফলে এমিটার আর কালেকটরের মধ্যেও তড়িৎ প্রবাহ বাড়ে।

এভাবেই একটা ট্রানজিস্টরকে সুইচ হিসেবে ব্যবহার করা যায়।

আমরা এই বইয়ে একটা ডিসি মোটরের ঘূর্ণন দিক কীভাবে নিয়ন্ত্রণ করা যায় সেটি দেখব। সাথে দেখব কীভাবে একটা ডিসি মোটরের ঘূর্ণন গতি কমানো কিংবা বাড়ানো যায়। এই সবটুকুই করা সম্ভব ট্রানজিস্টরের সাহায্যে।

অপারেশনাল অ্যামপ্লিফায়ার : একটা সিগনাল বিবর্ধিত করতে এটি মূলত ব্যবহার করা হয়। বলা চলে মাইক্রো লেভেলের ভোল্টেজকেও চাইলে ভোল্টেজ লেভেলে আনা যায়। একই সিগনালকে চাইলে অনেকগুণে বিবর্ধিত করে ফেলা যায়। সবচেয়ে বড় ব্যাপার যে সিগনালটা মোটামুটি প্রায় একই রকমের থাকে। বর্তমানে বাজারে বিভিন্ন মডেলের অপারেশনাল অ্যামপ্লিফায়ার পাওয়া যায়। LM741 এ রকম একটি মডেল।



চিত্র ২.১৭ : LM741-এর অভ্যন্তরীণ কাঠামো

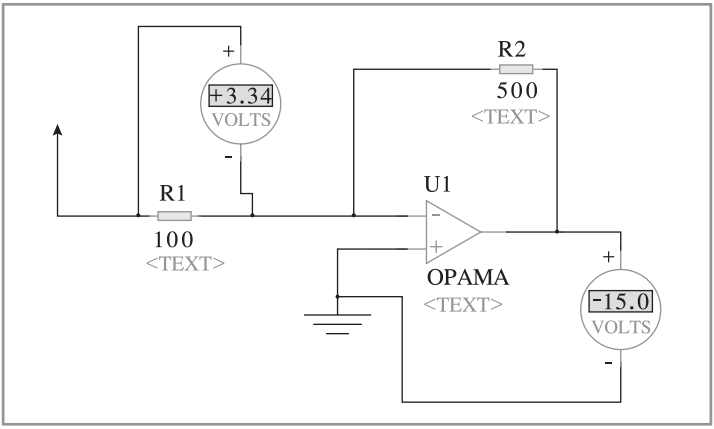
চিত্র ২.১৭-এ LM741 অপারেশনাল অ্যামপ্লিফায়ারের অভ্যন্তরীণ কাঠামো দেখানো হলো। এখানে পিন ২ আর পিন ৩ এই ২টো পিন ইনপুট হিসেবে কাজ করে, ৪ ও ৭ নম্বর পিন ব্যাটারির সাথে যুক্ত থাকে। অর্থাৎ এই ২টি পিন আইসিটিকে পাওয়ারিং করতে ব্যবহার করা হয়। আর ৬ নম্বর পিন আউটপুট হিসেবে কাজ করে।

তো অপারেশনাল অ্যামপ্লিফায়ার বিবর্ধন হিসেবে অনেকভাবে ব্যবহার করা যায়।

১. ইনভারটিং ইনপুট

২. নন-ইনভারটিং ইনপুট

ইনভারটিং ইনপুট : এই ইনপুট সিস্টেম কীভাবে সংযোগ দিতে হয় সেটি নিচের চিত্রে দেখানো হলো। LM741 পিন ডায়াগ্রামের সাথে মিলিয়ে বসালেই হবে। শুধু এখানে ব্যাটারি দেখানো হয়নি।



চিত্র ২.১৮ : LM741-এর সংযোগ ব্যবস্থা

এখানে বিবর্ধন কতটুকু হবে সেটা নির্ভর করে R2 এবং R1 এই ২টি রেসিস্টরের মানের ওপরে। এখানে দেখা যাচ্ছে ইনপুটের ধনাত্মক প্রান্ততে বিভব শূন্য। আর একটা জিনিস মাথায় রাখতে হবে যে এই ধনাত্মক আর ঋণাত্মক প্রান্ত শর্ট সার্কিট করা। ফলে এই প্রান্তে ভোল্টেজ ড্রপ শূন্য হবে। অর্থাৎ ২টি রোধকের মাঝে যেখানে একটা নির্দিষ্ট ভোল্টেজ ড্রপ থাকার কথা, সেখানে দেখা যাচ্ছে ভোল্টেজ ড্রপ শূন্য হয়ে যাচ্ছে। আর এ কাজটাই করে থাকে এই অপারেশনাল অ্যামপ্লিফায়ার।

তো কার্শফের তড়িৎ প্রবাহের সূত্র থেকে আমরা জানি যে 'একটি বর্তনীর কোনো একটি নির্দিষ্ট বিন্দুতে একত্র হওয়া তড়িৎ প্রবাহের মিলিত সমষ্টি শূন্য'। এখানে তীর চিহ্ন দেওয়া অংশটি দিয়ে ধনাত্মক বিভব বোঝানো হয়েছে যার মান ৫ ভোল্ট। এখন এটিকে V1 আর অপারেশনাল অ্যামপ্লিফায়ার থেকে বের হওয়া বিভবকে যদি v2 দিয়ে প্রকাশ করা হয় তাহলে কার্শফের তড়িৎ প্রবাহের সূত্র অনুযায়ী—

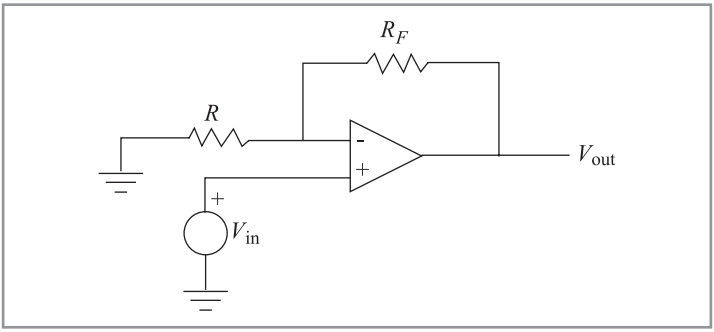
$$\frac{V_1}{R_1} + \frac{V_2}{R_2} = 0$$

$$\rightarrow -\frac{V_1}{R_1} = \frac{V_2}{R_2}$$

$$V_2 = -\frac{V_1 * R_2}{R_1}$$

কাজেই আমরা দেখলাম যে অপারেশনাল অ্যামপ্লিফায়ার দিয়ে বিবর্ধন নির্ভর করে রেসিসটেন্সের। এখানে R2-এর মান যত বাড়বে R1-এর মান যত ছোট হবে বিবর্ধন ততগুণ হবে। এখানে আউটপুট ভোল্টেজ দেখা যাচ্ছে ১৫ আর ইনপুট ৩-এর একটু বেশি। R1-এর মান ১০০ R2-এর মান ৫০০। কাজেই R2/R1-এর মান ৫। অর্থাৎ প্রায় ৫ গুণ বিবর্ধন আমরা পাব।

অপারেশনাল অ্যামপ্লিফায়ার এ রকম আরও অনেকভাবে ব্যবহার করা যায়।



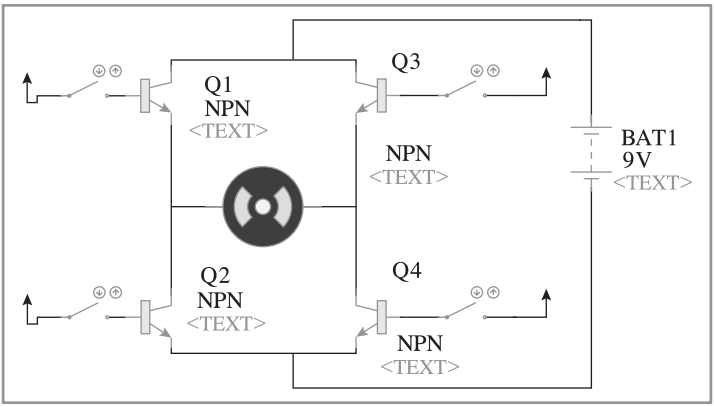
চিত্র ২.১৯ : নন ইনভারটিং সিস্টেম

চিত্র ২.১৯-এ যে অপারেশনটি দেখানো হয়েছে, সেটি নন-ইনভারটিং সিস্টেম। এখন এটির হিসাব তোমরাই বের করো।

মোটর ড্রাইভার

মোটর ড্রাইভারের ব্যবহার রোবটিকসে অনেক বেশি। মোটর ড্রাইভার ব্যবহার করা হয় মোটরের ঘূর্ণন গতির দিক নির্ধারণ করার জন্য। মোটরের ঘূর্ণনগতির পরিমাণ অর্থাৎ মোটরটি কতটুকু গতিতে ঘুরবে, সেটি নির্ধারণ করা যায় মোটর ড্রাইভারের সাহায্যে। আরডুইনো কিংবা এ.ভি.আর দিয়ে করা সম্ভব। তবে একটি মোটর ঘুরতে ০ থেকে ৩০/৪০ ভোল্ট প্রয়োজন। এত অধিক পরিমাণ ভোল্টেজ সরবরাহ করা সম্ভব না এসব ডেভেলপমেন্ট বোর্ডের পক্ষে। এ জন্যই মোটর ড্রাইভার ব্যবহার করা হয়।

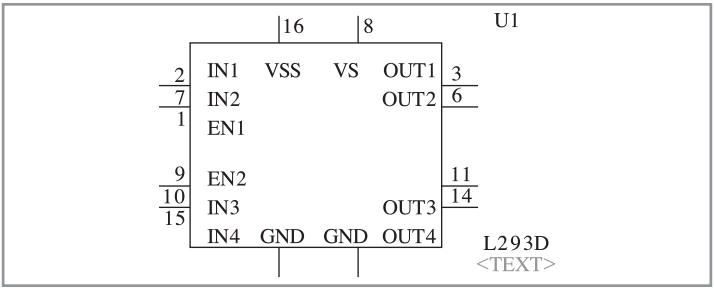
বাজারে L298N, L283D এই মডেলগুলো বেশি প্রচলিত। ট্রানজিস্টরের থিওরি অনুযায়ী এগুলো বানানো হয়। ভেতরের অভ্যন্তরীণ কাঠামোর একটা সার্কিট ডায়াগ্রাম নিচে দেখানো হলো।



চিত্র ২.২০ : মোটর ড্রাইভারের অভ্যন্তরীন কাঠামো

এখানে আমরা ৪টি সার্কিটসহ একটি কাঠামো দেখতে পাচ্ছি। আগেই বলা হয়েছে যে এমিটার বেসে তড়িৎ প্রবাহিত হলে এমিটার কালেকটরে তড়িৎ প্রবাহিত হয়। এখানেও সেই পদ্ধতি। সিগনালকে সহজভাবে বোঝানোর জন্য সুইচ অন অব পদ্ধতি হিসাবে দেখানো হলো। এখানে Q1, Q2, Q3, Q4 এই ট্রানজিস্টর ৪টি দিয়ে মোটরটির জন্য সংকেত প্রবেশ করে। যখন ট্রানজিস্টরে এমিটার বেস দিয়ে বিদ্যুৎ প্রবাহিত হয় তখন একটা ব্যপার দেখা যাচ্ছে যে যখন Q1 ও Q4 অন করা মানে Q1 ও Q4-এর এমিটার আর কালেকটরের মধ্যে বিদ্যুৎ প্রবাহিত হচ্ছে তখন ব্যাটারি থেকে তড়িৎ এসে Q1 দিয়ে ঢুকে তারপর Q4-এ যায়। এরপর ব্যাটারির ঋণাত্মক প্রান্তে প্রবেশ করে। অনুরূপভাবে Q3 ও Q2-এর জন্য একই কথা প্রযোজ্য। এখানে দেখা যাচ্ছে প্রতিবার সিগনালের পরিবর্তনের কারণে তড়িৎ প্রবাহের দিক পরিবর্তিত হচ্ছে। তাই মোটরের ঘূর্ণন গতির দিক পরিবর্তিত হচ্ছে। এখন মোটরের গতি নির্ভর করছে অনেকটাই ব্যাটারি থেকে কতটুকু ভোল্টেজ সাপলাই দেওয়া হচ্ছে তার ওপরে। এখানেও ট্রানজিস্টর ব্যবহার করে ভোল্টেজের মাত্রা নির্ধারণ করা যায়। ফলে মোটরের ঘূর্ণন গতিও পরিবর্তন করা সম্ভব।

এবার আমরা দেখব কীভাবে আইসি ব্যবহার করে মোটর কন্ট্রোল করা সম্ভব।



চিত্র ২.২১ : আইসি বা ইন্টাগ্রেটেড সার্কিট

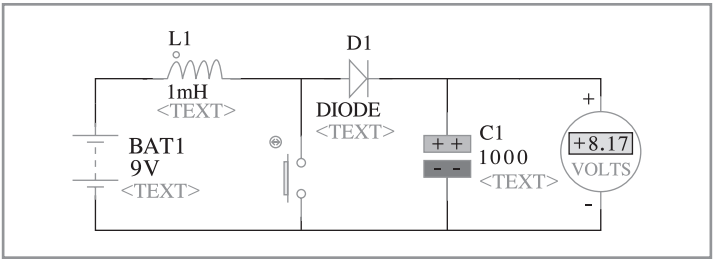
এখানে বামপাশে ৬টি আর ডানপাশে ৪টি পিন দেখা যাচ্ছে। বামপাশের ৬টি পিন ইনপুট আর ডানপাশের ৪টি পিন আউটপুট। প্রথম ২ নম্বর পিনটা In1 অথবা Input 1 আর ৭ নম্বর In2 অথবা Input 2। আর ১ নম্বর পিন হচ্ছে En1 অথবা enable 1। En1 একটা মোটরের গতি কেমন হবে সেটি নির্ধারণ করে সেটি যে দিকেই ঘুরুক না কেন, পরিবর্তনশীল ভোল্টেজ প্রয়োগের মাধ্যমে। Enable 1 পিনে ০ থেকে ৫ ভোল্টের মধ্যে যেকোনো মানের ভোল্টেজ প্রয়োগ করা যায়, কিন্তু ইনপুট পিনে শুধু ০ ভোল্ট অথবা ৫ ভোল্ট যেকোনো একটি সংকেত প্রয়োগ করা হয়, শুধু ডিজিটাল সংকেত প্রয়োগ করা হয়। প্রোগ্রামিংয়ের মাধ্যমে কীভাবে এই আইসি দিয়ে মোটর কন্ট্রোল করা যায় সেটি আমরা পরবর্তী সময়ে দেখাব।

ক্যাপাসিটর

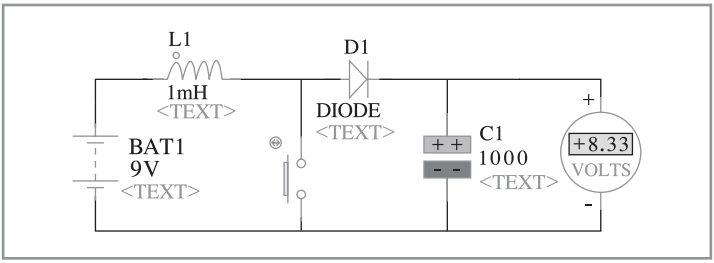
ক্যাপাসিটরের ব্যবহার খুব একটা বেশি নেই। যেহেতু রোবটে সাধারণত ডিসি ভোল্টেজ প্রয়োগ করা হয় তাই ক্যাপাসিটর খুব একটা ব্যবহার হয় না। তবে প্রধানত ২টি কাজে এর ব্যবহার রয়েছে।

পাওয়ার সাপ্লাই ডিজাইন আর ভোল্টেজ বুস্টার হিসেবে।

পাওয়ার সাপ্লাই ডিজাইন আগেই আলোচনা করা হয়েছে। এখন আমরা ভোল্টেজ বুস্টার কীভাবে তৈরি করতে হয় আর কীভাবে কাজ করে সেটি বোঝাব। বাজারে অনেক রকমের ভোল্টেজ বুস্টার পাওয়া যায়। এখানে ভোল্টেজ বুস্টারের থিওরি আলোচনা করলে সহজেই বোঝা যাবে কোন কাজের জন্য কার কেমন ভোল্টেজ বুস্টার লাগবে। প্রথমে একটি খুব সাধারণ ভোল্টেজ বুস্টার সম্পর্কে আলোচনা করব। পরে আরও উন্নতমানের ভোল্টেজ বুস্টার সম্পর্কে আলোচনা করা হবে।



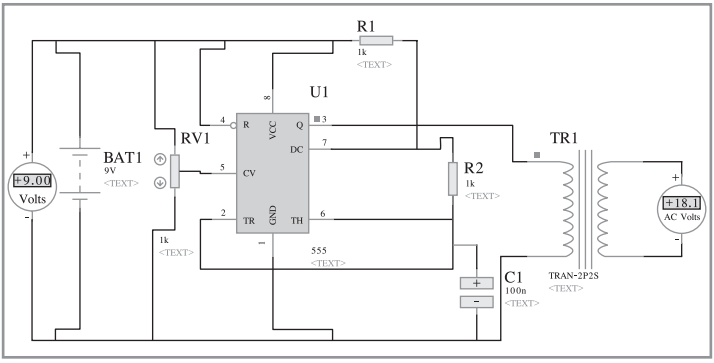
চিত্র ২.২২ : ভোল্টেজ বুস্টার



চিত্র ২.২৩ : ভোল্টেজ বুস্টার (ভোল্টেজ বৃদ্ধি)

এখানে দেখা যাচ্ছে একটি ভোল্টেজ বুস্টারের খুব সাধারণ একটি সার্কিট ডায়াগ্রাম। কপার কয়েল প্যাঁচানো একটি ইনডাকটর, ডায়োড আর ক্যাপাসিটরের সমন্বয়ে এটি গঠিত। এখানে ক্যাপাসিটরের স্বাভাবিক ভোল্টেজ ২৪ ভোল্ট। ইনডাকটরের সম্পর্কে কিছু বলি। ইনডাকটরের ভেতরে তড়িৎ প্রবাহিত হলে এটি চুম্বকায়িত হয়। ফলে কিছু শক্তি সঞ্চিত হয়। পরবর্তী সময়ে এই শক্তি ব্যবহার করা যায়। এই সার্কিটে যখন পুশ বাটন অফ থাকে, তখন ক্যাপাসিটরে আধান সঞ্চিত হতে থাকলে ভোল্টেজ বাড়তে থাকে। ২৪ ভোল্টে পৌঁছানোর পরে তড়িৎ প্রবাহ বন্ধ হয়ে যায়। এই ব্যাপারটা মনে রাখতে হবে ডিসি ভোল্টেজে ক্যাপাসিটর একটা নির্দিষ্ট সময় পরে পুরোপুরি চার্জিত হওয়ার পরে এরপর আর চার্জ নেয় না। যার ফলে ওই পথে আর বিদ্যুৎ প্রবাহ হয় না। এখানে ক্যাপাসিটর পুরোপুরি চার্জ হয়ে গেলে আমরা পুশ বাটন অন করলে অতি অল্প সময়ের মধ্যে ইনডাকটরে একটি নির্দিষ্ট পরিমাণ শক্তি সঞ্চয় হয়। এরপর পুশ বাটন অফ করা হলে এই শক্তি সরাসরি ক্যাপাসিটরে চলে যায়। এভাবে ভোল্টেজ বাড়তে থাকে। এখানে পুশ বাটন দেখানো হলো। কিন্তু স্বাভাবিকভাবে PWM সিগনাল দেওয়া হয়।

এখন 555 টাইমার আইসি ব্যবহার করে একটি বুস্টার দেখানো হলো।



চিত্র ২.২৪ : 555 টাইমার ব্যবহার করে বুস্টার

যেহেতু মাইক্রোকন্ট্রোলার নিয়ে কাজ করব, সেহেতু টাইমার আইসির খুব একটা প্রয়োজন হবে না। কারণ মাইক্রোকন্ট্রোলার দিয়ে খুব সহজেই টাইমার আইসির কাজ করা যায়।

ইলেকট্রনিক বেসিক নিয়ে এতটুকুই বলার ছিল। আশা করি অনেক সহজবভাবে বোঝাতে পেরেছি। পরবর্তী অধ্যায়ে প্রোগ্রামিং নিয়ে আলোচনা করব।

সি প্রোগ্রামিং দিয়ে শুরু

সি প্রোগ্রামিং মাইক্রোকন্ট্রোলার সিস্টেমে সবচেয়ে বেশি ব্যবহার করা প্রোগ্রামিং ভাষা। বলা চলে আসলে সি দিয়ে অধিকাংশ মাইক্রোকন্ট্রোলার সিস্টেম প্রোগ্রাম করা হয়। এ জন্যই সি সম্পর্কে একটু ধারণা রাখা উচিত। এ অধ্যায়টিতে সি নিয়ে আমরা কিছু প্রাথমিক আলোচনা করব। আরডুইনো নিয়ে কাজ করতে হলে সি নিয়ে আমাদের অবশ্যই ধারণা রাখতে হবে। তাহলে শুরু করা যাক। এখানে ব্যাপকভাবে আলোচনা করা হবে না। শুধু বেসিক নিয়ে আলোচনা করা হবে। যেহেতু উচ্চমাধ্যমিক এবং এর ওপরের পর্যায়ের পাঠকদের জন্য লেখা আশা করা যায় যে সি-এর ধারণা সবার রয়েছে। তাই আরডুইনোতে যেগুলো প্রয়োজন তার ওপরেই আলোচনা করা হবে।

সি প্রোগ্রামিংয়ের গঠন বা স্ট্রাকচার

প্রতিটি প্রোগ্রামিংয়ের কোডে একটি স্ট্রাকচার থাকে। সে স্ট্রাকচার অনুযায়ী প্রোগ্রাম লিখতে হয়। সি-এরও সে রকম একটি স্ট্রাকচার আছে। আর

সেটা মানাটাও অনেক গুরুত্বপূর্ণ। না হলে প্রোগ্রাম কখনো রান করবে না।

হেডার ফাইল

সি শুরু হয় একটি হেডার ফাইল দিয়ে। হেডার ফাইল সম্পর্কে সংক্ষেপে বলি। প্রতিটি কোডে ফাংশন থাকে অনেকগুলো। কিছু হয়তো প্রোগ্রামারের বানানো। আর কিছু সি-এর নিজস্ব। আর সি-এর নিজস্ব ফাংশন ব্যবহার করার আগে মাথায় রাখতে হবে যে ওই ফাংশন কোন হেডার ফাইলের ভেতরে আছে। সেটাই প্রথমে লিখতে হয়। লেখার পদ্ধতিটি আগে #include <something.h> দিয়ে শুরু করতে হবে। এখানে something-এর জায়গায় stdio, string, conio, math এসব বিভিন্ন হেডার ফাইল বসবে। কোনটি বসবে সেটি নির্ভর করে ফাংশনের ওপরে। কোন হেডার ফাইলের ভেতরে কোন ফাংশন থাকে সে অনুযায়ী।

যেমন কতগুলো ফাংশনসহ কাজ নিচে দেওয়া হলো—

ফাংশন	কাজ	হেডার ফাইল	লেখার পদ্ধতি
printf ()	কোনো কিছু প্রিন্ট করা	stdio.h	#include <stdio.h>
scanf ()	কোন কিছু ইনপুট নেওয়া	stdio.h	#include <stdio.h>
getch ()	কিবোর্ডে কিছু ক্লিক করা পর্যন্ত অপেক্ষা করা	conio.h	#include <conio.h>
sleep ()	একটি নির্দিষ্ট সময় পর্যন্ত অপেক্ষা করা	time.h	#include <time.h>
sine ()	সাইনের কোনো একটি নির্দিষ্ট কোণের মান প্রকাশ	math.h	#include <math.h>
cosine ()	কোসাইনের কোনো একটি নির্দিষ্ট কোণের মান প্রকাশ	math.h	#include <math.h>
fclose ()	ফাইল বন্ধ করা	stdlib.h	#include <stdlib.h>
fprintf ()	ফাইলে কোনো কিছু প্রিন্ট করা	stdlib.h	#include <stdlib.h>

ছক ২.১ : সি প্রোগ্রামে বিভিন্ন ফাংশনের কাজ

হেডার ফাইল অনেকগুলো আছে। কোনটা কোন কাজে ব্যবহার করতে হবে, সেটা নির্ভর করে কোন ফাংশন ব্যবহার করা হবে তার ওপরে।

ফাংশন : ফাংশন সম্পর্কে আমরা কিছু না কিছু জানি। অঙ্কে এ সম্পর্কে জানা হয়েছে। কাজেই খুব পরিষ্কার ধারণা থাকার কথা। অঙ্কে আমরা ফাংশনের যে রকম ব্যবহার জানি অনেকটা সে রকম ব্যবহার প্রোগ্রামিং কোডে। ফাংশনকে প্রধানত ৩ ভাগে ভাগ করা যায়।

১. প্রধান ফাংশন বা মেইন ফাংশন

২. বিভিন্ন কাজের জন্য হেডার ফাইলের অধীনে থাকা ফাংশন

৩. একটি নির্দিষ্ট কাজ বারবার সম্পাদন করার জন্য নিজ থেকে ডিক্লেয়ার করা ফাংশন

প্রধান ফাংশন বা মেইন ফাংশন : একটি প্রোগ্রামে একটি মাত্র প্রধান ফাংশন থাকে। শুধু সি না সব প্রোগ্রামিং ল্যাঙ্গুয়েজে একটি মাত্র প্রধান ফাংশন থাকে। প্রধান ফাংশন অন্য সব ফাংশনকে ধারণ করে। সি প্রোগ্রামে প্রধান ফাংশনকে কয়েকভাবে প্রকাশ করা যায়। যেমন— int main (), void main () ইত্যাদি।

ফাংশনে নামের অংশটা লেখার পরে ফার্স্ট ব্র্যাকেট দিতে হয়। এটা সি প্রোগ্রামের একটা নিয়ম। ব্র্যাকেটের ভেতরে কিছু থাকবে নাকি থাকবে না, সেটা নির্ভর করে ফাংশনের ওপরে। মেইন ফাংশনের ব্র্যাকেটের ভেতরে সাধারণত কিছু থাকে না।

একটি উপযুক্ত সি কোডের উদাহরণ—

```
#include <something.h> // Depended on the functions that we will use
int main ()
  {
   // Working principle or what the code will do
  }
```

ওপরে মেইন ফাংশন কীভাবে ব্যবহার করতে হয় সেটি দেখানো হলো। শুধু মেইন ফাংশন না ব্যবহারের জন্য আমরা যদি ফাংশন তৈরি করি তখন সেগুলো একইভাবে লিখতে হবে। সেটা পরে দেখানো হবে।

```
#include <stdio.h>
int main ()
```

```
{
        printf ("Hello world");
}
```

এখানে এই প্রোগ্রামটি রান করলে সেটি Hello world প্রিন্ট করবে। এখানে printf একটি ফাংশন যেটা stdio.h-এর অধীনে রয়েছে।

একাধিক কাজ বারবার করার জন্য ফাংশন তৈরি করতে হয়। আগে আমরা লজিক, লুপ এগুলো দেখাব। তারপর সবশেষে ফাংশন তৈরি নিয়ে আলোচনা করা হবে।

** একটা ব্যাপার বিশেষভাবে লক্ষ করতে হবে যে সি প্রোগ্রামের প্রতিটি লাইনে সেমিকোলন দিতে হয়। না হলে ভুল দেখাবে কম্পাইলার।

গাণিতিক চিহ্ন

লক্ষণীয় যে সি প্রোগ্রামে গাণিতিক অপারেশনের জন্য যোগ, বিয়োগ, গুণ, ভাগ এ ছাড়া ভাগশেষ ব্যবহার করার জন্য ৫টি প্রধান চিহ্ন ব্যবহার করা হয়—

চিহ্ন	কাজ
+	যোগ করা
-	বিয়োগ করা
*	গুণ করা
/	ভাগ করা
%	ভাগশেষ বের করা

ছক ২.২ : গানিতিক অপারেটর

পরবর্তী সময়ে বিভিন্ন উদাহরণ প্রোগ্রামে আমরা বিষয়টা দেখব।

চলক

প্রোগ্রামিংয়ের অনেক গুরুত্বপূর্ণ অংশ। বলা চলে এটি ছাড়া প্রোগ্রাম প্রায় অচল। অঙ্কের হিসাব-নিকাশ করার জন্য, একটি মান ধারণ করা, একটি অক্ষর ধারণ করা অথবা একটি পুরো বাক্যকে ধারণ করা—এসব কাজের জন্য চলক বা ভ্যারিয়েবল অনেক গুরুত্বপূর্ণ।

এখানে ছক ২.৩-এ কোন কাজের জন্য কোন ভ্যারিয়েবল সেটি দেখানো হলো।

Variable type	Working principle
Int	সংখ্যা ধারণ করা
Char	অক্ষর ধারণ করা
Array	একসাথে অনেকগুলো সংখ্যা ধারণ করা
String	একটি সুনির্দিষ্ট শব্দ অথবা বাক্য ধারণ করা
Float	অঙ্ক ধারণ করা, তবে এর ধারণক্ষমতা অনেক বেশি। দশমিক সংখ্যাও ধারণ করতে পারে
Double	একই রকম float-এর মতো
Bool	এটি একটু ভিন্ন টাইপের। এটি মূলত সত্য-মিথ্যার তথ্য ধারণ করে

ছক ২.৩ : ভ্যারিয়েবলের ধরন

এখানে কয়েকটি উদাহরণ দেখলেই বোঝা যাবে ব্যাপারটা। তবে মনে রাখতে হবে যে সি প্রোগ্রামসহ অধিকাংশ প্রোগ্রামে কাজ চলে প্রতি লাইন পরপর। একটি লাইনে কী আদেশ দেওয়া আছে, সেটি করে তারপর পরবর্তী লাইনে প্রবেশ করবে।

১.

```
#include <stdio.h>
int main ()
{
    int a;  // প্রথমে a নামে একটি চলক ডিক্লেয়ার করা হলো
    scanf ("%d", &a); // a-এর মান কত হবে, সেটি ইনপুট নেবে
    printf ("%d", a);  // c কে প্রিন্ট করা হলো
}
```

এ প্রোগ্রামটি রান করলে প্রথমে একটি সংখ্যা ইনপুট নেবে। তারপর সেটা প্রিন্ট করবে। লক্ষণীয় যে এখানে আমরা যদি ইনপুটে দশমিকের কোনো সংখ্যা দিই, তবে সে ক্ষেত্রে সেটি আউটপুট দেবে অন্য রকম কিছু অথবা ইনপুট নেবে না। ভুল দেখাবে। বোঝার সুবিধার জন্য কোডের প্রতি লাইনে // চিহ্নের পরে এটি কী কাজ করছে, সেটি লিখে দেখানো হলো। লক্ষণীয় যে সি-তে // চিহ্নের পরে যাই থাকুক না কেন সেটি কমেন্ট হিসেবে ধরা হয় এবং প্রোগ্রাম সেটা সম্পাদন করে না

২.

```
#include <stdio.h>
int main ()
{
    int a ;      // প্রথমে a নামে একটি চলক ডিক্লেয়ার করা হলো
    int b ;      // b নামে আরেকটি চলক ডিক্লেয়ার করা হলো
    int c ;      // c নামে আরেকটি চলক ডিক্লেয়ার করা হলো
    scanf ("%d", &a);  // a-এর মান কত হবে সেটি ইনপুট নেবে
    scanf ("%d", &b);  // b-এর মান কত হবে, সেটি ইনপুট নেবে
    c=a+b;   // a আর b যোগ করল তারপর সেটি প তে সংরক্ষিত রাখা হলো
    printf ("%d", c); // c কে প্রিন্ট করা হলো
}
```

এই প্রোগ্রামে প্রথমে ২টি সংখ্যা ইনপুট নেবে তারপর সেটি যোগ করে সেটাকে প্রিন্ট করবে।

এবার দশমিকের ওপরে ভিত্তি করে একটি কোড দেখাব। দশমিকের জন্য শুধু int-এর বদলে float লিখতে হবে। এটাই পার্থক্য।

```
 #include <stdio.h>
int main ()
{
     float a; // প্রথমে ধ নামে একটি চলক ডিক্লেয়ার করা হলো যা দশমিক সংখ্যা
ধারণ করতে পারে
    scanf ("%f", &a); );  // a-এর মান কত হবে, সেটি ইনপুট নেবে
    printf ("%f", a); // a-কে প্রিন্ট করা হলো
}
```

এখানে কোডটি যে দশমিক সংখ্যা ইনপুট দেওয়া হবে সেই সংখ্যাটি আউটপুট দেখাবে।

আরেকটি কোড দেখলে পরিষ্কার হবে ব্যাপারটা

৩.

```
#include <stdio.h>
int main ()
{
    float a, b, c; // a, b, c এই ৩টি চলক ডিক্লেয়ার করা হলো
    scanf ("%f", &a); // a-এর ইনপুট নেওয়া হলো। সংখ্যাটি দশমিক হবে
    scanf ("%f", &b); // b-এর ইনপুট নেওয়া হলো। এটাও দশমিক
```

```
    c=a+b; // c হলো a আর b-এর যোগফল। a ও b-এর যোগফল c-তে সংরক্ষণ করা হবে
    printf ("%f", c); // cকে প্রিন্ট করা হলো
}
```

সি প্রোগ্রাম লেখার সময় কিছু সংখ্যা অথবা কোনো অক্ষর প্রিন্ট করার সময় একটা বিশেষভাবে চিহ্ন দিয়ে তারপর অক্ষর দিয়ে একটা পদ্ধতিতে লিখতে হয়। এটা বিশেষভাবে মনে রাখতে হবে। সি প্রোগ্রামসহ আরও কয়েকটি ল্যাঙ্গুয়েজে এ পদ্ধতিতে লেখা হয়। যেমন— সাধারণ ডেসিমাল সংখ্যা প্রিন্ট করতে %d দিয়ে আবার দশমিক সংখ্যা %f দিয়ে অন্যদিকে অক্ষর প্রকাশ করতে হয় %c এর মাধ্যমে। কোনটির জন্য কীভাবে লিখতে হবে, সেটির একটি ছক দিয়ে দেওয়া হলো।

প্রতীক	কাজ
%d	সাধারণ সংখ্যা প্রিন্ট করতে
%f	দশমিক সংখ্যা প্রিন্ট করতে
%c	অক্ষর প্রিন্ট করতে
%s	শব্দ প্রিন্ট করার জন্য
%p	পয়েন্টারের জন্য। অনেক কম ব্যবহার হয়

ছক ২.৪ : ফরম্যাট স্পেসিফায়ার

কন্ট্রোল স্টেটমেন্ট : (লজিক ও লুপ)

লজিক রোবটিকসের সবচেয়ে বড় ভিত্তি এটা সবার আগে মনে রাখতে হবে। মেইনলি লজিকের ওপরে ভিত্তি করে একটা রোবট চলে। কারণ একটি রোবটকে চলতে হলে রোবটকে সিদ্ধান্ত নিতে হয়। চলতে গেলে একটা রোবটকে ঠিক করতে হয় সে কী করবে সেটা সেন্সরের পাঠ থেকে। একটা ব্যাপার বললে অনেক পরিষ্কার হবে খুব ভালো মতো যে আর্টিফিশিয়াল ইন্টেলিজেন্স তৈরি হয় অসংখ্য লজিকের সমন্বয়ে। একটা প্রোগ্রাম কতটা ভালো সেটা নির্ভর করে এর লজিক কতটা ভালো এবং কী পরিমাণে লজিক আছে তার ওপর নির্ভর করে। তাহলে আসল কোডে যাওয়া যাক।

শুরু করতে হয় if দিয়ে। ফরম্যাটটি অনেকটাই এরকম।

```
if ( এখানে লজিক বসবে )
{
    ফাংশন, অর্থাৎ যেটা সম্পাদন করবে।
   বি.দ্র. যদি লজিক মেলে অর্থাৎ লজিক অনুযায়ী শর্ত পূরণ হয়, তবেই এটা সম্পাদন হবে।
}
```

কিছু উদাহরণ দিলে ব্যাপারটা পরিষ্কার হবে

উদাহরণ ১

এই কোডটি খুবই সিম্পল। এখানে শুধু দেখানো হবে যে একটি প্রোগ্রামে লজিক অনুযায়ী হয় একটি কাজ নয়তো আরেকটি কাজ সম্পাদন করবে—

```
#include <stdio.h>
int main ()
{
    int j;
   scanf("%d", &j);
    if (j==1)
    {
        printf ("You have given input 1");
    }
    else if (j==0)
    {
        printf ("You have given input 0");
    }
    else
    {
        printf ("You have not given input binary");
    }
}
```

প্রোগ্রামটি চালালে বোঝা যাবে ব্যাপারটা। এখন যে ব্যাপারটি বলব else এবং else if ব্যবহার করা হয়েছে। এই ২টি ব্যবহার করা হয় তখনই, যখন যদি if-এ যে অংশ অথবা শর্ত রয়েছে সেটি যদি শর্ত অনুযায়ী না পড়ে সে ক্ষেত্রে। উল্লেখ্য, if সব শর্তের জন্য ব্যবহার করা যায়। তবে পুরো প্রোগ্রাম চলতে অনেক সময় নেয়।

উদাহরণ দিলে ব্যাপারটি পরিষ্কার হবে।

উদাহরণ ২

```
#include <stdio.h>
int main ()
{
    int i, even_2=0, even_4=0 ;
    for (i=1;i<=1000;i++)
    {
       if (i%2==0)              // % এই চিহ্ন টি দিয়ে ভাগশেষ বের করা বোঝায়
        {
           even_2=even_2+1;
        }
        if (i%4==0)
        {
           even_4=even_4+1;
        }
    }
    if (even_2>=even_4)
    {
       printf ("Numbers divided by 2 is greater");
   }
    if (even_2<even_4)
    {
        printf ("Numbers divided by 4 is greater");
    }
}
```

একই প্রোগ্রাম যদি অন্যভাবে লেখা যেত, যেখানে if-এর বদলে else if ব্যবহার করা হয়েছে, সে ক্ষেত্রে ব্যাপারটি ভিন্ন ছিল না। কিন্তু এই কোড রান করতে যতটুকু সময় লেগেছে তার চেয়ে কম সময় লাগত। ছোটখাটো কোডে এ ব্যাপারটি বোঝা যায় না। কিন্তু প্রোগ্রাম যখন আকারে বড় হতে থাকে, তখন এরও আউটপুট বের হতেও সময় লাগে। এই সময়কে কমানোর জন্যই এই else অথবা else if ব্যবহার করা হয়। এখানে আসলে if ব্যবহার করলেও চলে। কোনো সমস্যা নেই।

সুইচ সিস্টেম

অনেক সময় দেখা যায় যে একটি কোডে অনেকগুলো লজিক থাকে। এখন শুধু if, else, else if ব্যবহার করে বারবার লেখা হয়তো সম্ভব কিন্তু কোডটি আকারে যখন বড় হয়, তখন if, else, else if ব্যবহার

করে লিখলে কোডের আকার বাড়তে থাকে আর লেখার সৌন্দর্য নষ্ট হয়। তাই এ ক্ষেত্রে সুইচ ব্যবহার করে। একাধিক লজিকের ক্ষেত্রে সুইচ ব্যবহারে একদিকে যেমন কোডের সৌন্দর্য বাড়ে, তেমনিভাবে লেখার বাহুল্য থাকে না। একটি উদাহরণ দিলে বোঝা যাবে।

```
#include <stdio.h>
int main ()
{
    int a;
    scanf ("%d", &a);
    switch(a)
    {
    case 1 :
        printf ("You have given input 1\n");
        break;
    case 2 :
        printf ("You have given input 2\n");
        break;
   case 3 :
       printf ("You have given input 3\n");
        break;
    default :
        printf ("You have given other input\n");
    }
}
```

এখন এ জায়গায় যদি if, else, else if ব্যবহার করা হয়, তবে সে ক্ষেত্রে—

```
#include <stdio.h>
int main ()
{
    int a;
    scanf ("%d", &a);
    if (a==1)
    {
        printf ("You have given input 1\n");
    }
    else if (a==2)
    {
        printf ("You have given input 2\n");
```

```
    }
  else if (a==3)
    {
        printf ("You have given input 3\n");
    }
    else
    {
        printf ("You have given input other number\n");
    }
}
```

২টি কোডের তুলনামূলক বিশ্লেষণ করলেই বোঝা যাবে পার্থক্যটি।

লুপ

লুপ কথাটায় সহজেই বোঝা যাচ্ছে কী নিয়ে আলোচনা করব। এই বইয়ে আগেই বলা হয়েছে যে সি প্রোগ্রাম সম্পর্কিত ধারণা নিয়ে আসতে হবে। তাই সে অনুযায়ী বিস্তারিত আলোচনা করা হয়নি। শুধু পূর্ব জ্ঞানের সারমর্মের ওপরে আলোকপাত করা হয়েছে। এই লুপের ওপরে আলোচনাটাও এমন হবে। লুপ সম্পর্কে আমাদের একটি ভালো জ্ঞান থাকা দরকার। একটা রোবট বানাতে হলে অসংখ্যবার এই লুপ ব্যবহার করতে হবে। তাই পূর্ববর্তী জ্ঞান একটু সমৃদ্ধ থাকলে ভালো।

লুপ সম্পর্কিত আলোচনায় প্রথমেই যে কথাটি বলব যে একটি কাজ একটি নির্দিষ্টবার করার জন্য এই লুপ ব্যবহার করা হয়। আমার একটি কাজ করার জন্য ধরা যাক অন্য একটি কাজ নির্দিষ্টবার সম্পাদন করব। একটা উদাহরণ দিলে পরিষ্কার হবে। ফেসিয়াল রিকগনিশন পদ্ধতিতে কিছু মানুষকে একটি রোবট চিনবে। তার ডেটাবেইসে রাখা ছবির সাথে মিলাবে। এখন ধরা যাক মানুষের পরিমাণ প্রায় ১০০-এর পরে। অথবা এমনো হতে পারে যে মানুষের পরিমাণ সংখ্যাটি কমল অথবা বাড়ল। এই যে পরিবর্তিত সংখ্যা এখানেই ব্যাপার। আমি কিন্তু প্রোগ্রামে এক কাজ বারবার করার জন্য বারবার কোড লিখব না। শুধু একবারই লিখব। আর এ কারণেই লুপ। শুরু করা যাক তাহলে।

লুপ সাধারণত ৩ ভাবে ব্যবহার করা যায়। for, while আর do while এই ৩ ভাবে। আমি এখানে শুধু for, while নিয়েই আলোকপাত করব। কারণ এই ২টি সবচেয়ে ব্যবহার করা হয়।

প্রতিটা লুপ-এর প্রধান ব্যাপার মনে রাখতে হবে।

১. কোথায় শুরু

২. কোথায় শেষ

৩. কীভাবে পরিবর্তিত হবে

এখানে লুপটি একটি নির্দিষ্ট অবস্থান থেকে শুরু হবে। ধরা যাক একটি চলক। এটি একটি লুপ পার করে একটি নির্দিষ্ট সময় পরে অন্য একটি মান থাকবে চলকটিতে। কোথায় শুরু এটি আমরা প্রাথমিকভাবে নির্ধারণ করে দেব। যেমন একটি উদাহরণ দেখলে স্পষ্ট হবে ব্যাপারটি।

```
#include <stdio.h>
int main ()
{
    int i;
    for (i=1;i<=10;i=i+1)
    {
       printf ("I love robotics \n");
    }
}
```

এখানে কথাটি I love robotics ১০ বার প্রিন্ট হবে। এখানে প্রথমে i=1 দিয়ে বোঝানো হয়েছে কোথায় শুরু হবে। এখানে i একটি চলক। তাই i-এর মান যে পরিবর্তিত হবে সেটাই বোঝা যাচ্ছে। ধাপটি শেষ হবে i-এর মান যখন ১০ হবে। আর প্রতিটা ধাপ একবার করে আগাবে। এখানে তাহলে for loop-এর ব্যাপারটি দাঁড়াল এ রকম।

for (i= কোথা থেকে শুরু; i<= কোথায় এর মান শেষ অর্থাৎ কোথায় লুপটির কাজ থামবে; i = i-এর মান ধাপে ধাপে কীভাবে বাড়বে অর্থ লুপটি একটি কাজ শেষ করার পরে চলকটির মান পরবতীর্ ধাপে কত হবে)

```
#include <stdio.h>
int main ()
{
    int i;
    for (i=1;i<=10;i++)
    {
        printf ("%d\n", i);
    }
}
```

এই উদাহরণটির মাধ্যমে পরিষ্কার হবে i-এর মান সম্পর্কে।

এখানে কীভাবে কাজ করবে কোডটি সেটি বলি। i প্রথমে কোডটির কম্পাইলার লুপে ঢুকল। সে যখন দেখল এর মান ১ তখন পরবর্তী কমান্ডে কাজটি করল যেটি হচ্ছে ১ প্রিন্ট করা। প্রিন্ট করে তখন সে এর মান ১ বাড়াল যেটি সে দেখতে পারবে for loop-এর ব্র্যাকেটের ভেতরে থাকা কমান্ড i=i+1 অথবা i++ এর মাধ্যমে। এখন শর্তটি দেওয়া আছে যে i এর সর্বোচ্চ মান ১০। যখন মান ১০ হবে তখন এই লুপের কাজটি শেষ হবে। তখন লুপ থেকে বের হয়ে যাবে। আর এভাবে for লুপের কাজ শেষ হবে।

আরও কয়েকটি উদাহরণ দিলে ব্যাপারটি পরিষ্কার হবে। কোডগুলো সফটওয়্যার দিয়ে চালিয়ে দেখার জন্য অনুরোধ করা হলো।

```
#include <stdio.h>
int main ()
{
    int i, j;
    for (i=1;i<=10;i++)
    {
        for (j=i;j<=10;j++)
        {
            printf ("*");
        }
        printf ("\n");
    }
}
```

এখানে প্রোগ্রামটি চালালে একটি নির্দিষ্ট আকৃতি দেখা যাবে। প্রোগ্রামটি সম্পর্কে বিস্তারিত বলি। এখানে প্রথম লুপটি শুরু হয়েছে i দিয়ে। পরবর্তী যে লুপ সেটি শুরু হয়েছে j দিয়ে। এখানে ত্রিভুজ আকৃতির একটি আকৃতি তৈরি হবে। প্রথম যে লুপটি শুরু হয়েছে i দিয়ে সেটি কন্ট্রোল করছে মোট কয়টা লাইন হবে। আর পরের যে লুপটি দেখা যাচ্ছে সেটি কন্ট্রোল করছে প্রতি লাইনে কয়টি তার চিহ্ন থাকবে। এখানে প্রথমে লুপে ঢুকার সময়ে i-এর মান ১। ধাপ বাড়বে ১ করে এবং সবশেষে যখন মান ১০ হবে তখন এটি কাজ করা বন্ধ করে দেবে। প্রথমে লুপটিতে ঢুকল। তখন মান হবে ১ এরপরে i লুপের ভেতরে থাকা j লুপে ঢুকল। এখানে j-এর মান আগের লুপের i-এর মানের ওপরে নির্ভরশীল। তাই প্রথমে i-এর মান যখন ১ তখন j-এর মানও

১। j-এর মান ১ হওয়ায় সে এই লাইনে ১টি স্টার প্রিন্ট করবে। তারপরে এই লুপের কাজ শেষ হলে পরবর্তী সময়ে j-এর লুপ থেকে বের হয়ে যাবে এবং প্রবেশ করবে i লুপে। এখানে উল্লেখ্য যে i লুপের কাজ কিন্তু শেষ হয়নি। শুধু একটি ধাপ শেষ হলো। পরবর্তী i লুপে একটি ধাপ অতিক্রমের পরে যে কমান্ড দেওয়া আছে printf ("\n") এটি সম্পাদন করবে। যার অর্থ এই লাইন শেষ পরবর্তী লাইনে প্রবেশ করতে হবে। এ কমান্ডটি সম্পাদনের মাধ্যমে ২টি লুপের কাজ শেষ হলো। এরপরে প্রতিবার i-এর মান ১ করে বাড়ার কারণে j-এর মানও ১ করে বাড়বে। সমান মান যেহেতু। তাই সে অনুযায়ী প্রথম লাইনে ১টি, পরবর্তীতে ২টি এভাবে ১০টি লাইনে লাইনের ক্রম অনুযায়ী স্টার প্রিন্ট হবে। প্রতি লাইনের পরে স্টারের সংখ্যা বাড়বে লাইনের সংখ্যা বাড়ার কারণে।

```
#include <stdio.h>
int main ()
{
    int i, j;
    for (i=10;i>=1;i--)
    {
        for (j=i;j<=10;j++)
        {
            printf ("*");
        }
        printf ("\n");
    }
}
```

While Loop

for loop-এর মতো while loop ও মোটামুটি ব্যবহার অনেক বেশি। while loop-এর গঠন for loop থেকে একটু ভিন্ন। কিন্তু কাজের সিস্টেম প্রায় একই। পার্থক্য শুধু এখানেই for loop-এ যেখানে কোথায় শুরু, কোথায় শেষ এবং কীভাবে ধাপ পরিবর্তিত হবে এ কয়টি জিনিস একটি ব্র্যাকেট এর ভেতরে থাকে সেখানে while loop-এ কোথায় শুরু হবে এটি থাকে লুপের বাইরে, কোথায় শেষ এটি থাকে লুপের ভেতরে ব্র্যাকেটে এবং কীভাবে ধাপ পরিবর্তিত হবে সেটি থাকে লুপের ভেতরে ফাংশনে। একটি উদাহরণ দিলে ব্যাপারটি বুঝতে সুবিধা হবে।

```
#include <stdio.h>
int main ()
{
    int i;
    printf ("Enter the value of i between 1 and 2  : ");
    scanf ("%d", &i);
    if (i==1 || i==2)
    {
        printf ("You have given correct input \n");
    }
    else
    {
        i=1;
   }   // এখানে i এর প্রাথমিক মান কত হবে এটা আগে থেকেই নির্ধারণ করা হয়েছে
    while (i<=10)   // i-এর মান কোথায় গিয়ে শেষ হবে
    {
        printf ("%d\n", i);
        i=i+1;  // i-এর মান কত ধাপে কীভাবে বাড়বে
    }
}
```

আরেকটি উদাহরণ দিয়ে পরিষ্কার করি। এটি হচ্ছে while loop এর ভেতরে while loop-এর একটি উদাহরণ।

```
#include <stdio.h>
int main ()
{
    int i, j;
    i=1;
    while (i<=10)
    {
        j=i;
        while (j<=10)
        {
            printf ("*");
            j=j+1;
        }
        printf ("\n");
        i=i+1;
    }
}
```

লুপের আলোচনা এতটুকুই। এখানে আগেই বলেছি সি নিয়ে বিস্তারিত কিছু বলব না শুধু আগের আলোচনাটুকুই আবার একটু রিপিট করলাম।

অ্যারে

অ্যারে নিয়ে আলোচনার খুব বেশি কিছু নেই। শুধু কয়েকটি কথা বলেই ফাংশন সম্পর্কে বলা শুরু করা হবে। অ্যারে নিয়ে আলোচনায় আমি বলব অ্যারে কি এবং কীভাবে ব্যবহার করা যায়।

১. অ্যারে কী?

সোজা কথায় অ্যারে অনেকগুলো চলকের সমষ্টি। অন্যভাবে উদাহরণ দিলে এমনভাবে বলি ধরা যাক আমার কাছে একটি কাপড় রাখার আলমারি আছে। এখানে আলমারিতে অনেকগুলো ড্রয়ার। প্রতিটি ড্রয়ার বাসার বিভিন্ন মানুষের জন্য বরাদ্দ রাখা। একদম সবার ওপরেরটা, ১ নম্বরটা আব্বুর, পরেরটা মানে ২ নম্বরটা আম্মুর, ৩ নম্বর, ৪ নম্বর এবং ৫ নম্বর যথাক্রমে ছেলে, মেয়ে এবং বাসার অন্যদের। এখানে আলমারিটা বিভিন্ন ড্রয়ার বিভিন্ন চলক সংগ্রহে রাখছে। আর পুরো আলমারিটা একটি অ্যারে।

অ্যারের বেশ কিছু নিয়ম আছে। এগুলো বলা হলো।

১. অ্যারের চলকগুলো সংখ্যা দিয়ে নির্ধারণ হয়। এটাকে অনেকটা index-এর সাথে তুলনা করা যায়।

২. index শুরু হয় শূন্য দিয়ে এটা মনে রাখতে হবে। শেষ কত দিয়ে হবে এটা নির্ধারণ করে দিতে হবে।

৩. index-এর যে মান দিয়ে শেষ হবে সেটি ছাড়া এর পরবর্তী সময়ে যেকোনো মানের index-এ গারবেজ মান থাকবে বলা চলে আসলে এরপরে আর কোনো index ব্যবহার করা যাবে না।

উদাহরণ ১

```
#include <stdio.h>
int main ()
{
    int ara[10]; // মোট ১০টি ইনটিজার ডিক্লেয়ার করা হলো
    int i;
    for (i=0;i<=9;i++) // লক্ষণীয় যে i-এর মান ০ থেকে শুরু হয়ে ৯ পর্যন্ত
```

```
মোটমাট ১০টি
    {
        ara[i]=i;
    }
    for (i=0;i<=9;i++)
    {
        printf ("%d \n", ara[i]);
    }
   printf ("%d", ara[10]);
}
```

এই কোডটি রান করলে ০ থেকে ৯ পর্যন্তও দেখা যাবে। প্রথমে ১০টি ইনটিজার ডিক্লেয়ার করা হয়েছে। এখানে index-এর যাত্রা শুরু হয়েছে ০ থেকে। যেকোনো অ্যারের index শুরু হয় ০ থেকে এটা মনে রাখতে হবে। আর আরেকটি ব্যাপার যেটি সেটি হচ্ছে ১০টি ইনটিজার ডিক্লেয়ার করার কারণে index ৯-এ গিয়ে শেষ হয়েছে। এ কারণে ১০ নম্বর index-এ গারবেজ মান দেখাচ্ছে।

উদাহরণ ২

```
// sensor reading system of a line following robot
#include <stdio.h>
int main ()
{

    int i, ara[8], min, max, thresh=0, thresh_prev=0;
    for (i=0;i<=7;i++)
    {
        printf ("Enter the reading of sensor %d  : ", i+1);
        scanf ("%d", &ara[i]); // এখান থেকে সেন্সরের পাঠ নেওয়া হবে
    }
    for (i=0;i<=7;i++)
    {
        thresh=ara[i];
        if (thresh>thresh_prev)
        {
            max=thresh;
         }
        thresh_prev=thresh;
    }
```

```
    thresh=0;
    thresh_prev=max;
      for (i=0;i<=7;i++)
    {
        thresh=ara[i];
        if (thresh<thresh_prev)
        {
            min=thresh;
       }
        thresh_prev=thresh;
    }
    printf ("\nSensor minimum value is %d", min);
    printf ("\nSensor maximum value is %d", max);
}
```

রোবটিকসে অ্যারের মূলত কাজ সেন্সরের মান সংরক্ষণে। ধরা যাক একটি রোবটে একটি নির্দিষ্ট কাজ করার জন্য কয়েকটি সেন্সর রয়েছে। এখন সবগুলো থেকে আমার মান সংরক্ষণ করে সেখান থেকে একটি সিদ্ধান্তে আসতে হবে। এই যে মান নিলাম সেখান থেকে একটি সিদ্ধান্তে আসা এ জন্য অ্যারে অনেক কার্যকরী। ওপরে একটি উদাহরণ দিলাম। পরবর্তী সময়ে Arduino প্রোগ্রামিংয়ে এটি কীভাবে কাজে লাগানো হবে সেটি দেখাব। **** উল্লেখ্য কোনো অ্যারেতে সদস্যগুলো যদি ক্যারেকটার, মানে অক্ষর হয়, তবে সে ক্ষেত্রে সেই অ্যারেকে স্ট্রিং (string) বলে। কোনো ভিন্নতা নেই, পদ্ধতি একই।

অ্যারে-সংক্রান্ত আলোচনা এতটুকুই, রোবটিকসের জন্য যতটুকু প্রয়োজন তা-ই আলোচনা করা হয়েছে।

ফাংশন

বলা চলে এটি হচ্ছে সবচেয়ে গুরুত্বপূর্ণ অংশ। একটি জটিল প্রোগ্রামের মধ্যে অনেকগুলো করে ফাংশন থাকে এবং একটি মেইন ফাংশন থাকে। প্রোগ্রামগুলো কতটুকু কার্যকরী হবে সেটা নির্ভর করছে ফাংশনের ওপরে। একটি প্রোগ্রামে অনেকগুলো ফাংশন থাকতে পারে। তবে মেইন ফাংশন একটাই। ফাংশনের নিয়মগুলো দেওয়া হলো।

১. মেইন ফাংশন ছাড়া অন্য ফাংশনগুলো তৈরি করতে হয় মেইন ফাংশনের আগে, তবে পরে তৈরি করলেও ওই ফাংশনের প্রটোটাইপ মেইন ফাংশনের আগে বলে দিতে হয়।

২. অন্য ফাংশনগুলো তৈরি করতে হবে মেইন ফাংশনের বাইরে। এর পর মেইন ফাংশনে এই ফাংশন যতবার ইচ্ছা ততবার ব্যবহার করা হয়।

৩. অন্য ফাংশনগুলোর ভেতরে যে চলকগুলো ব্যবহার করা হয়, সেটি মেইন ফাংশনে আর ব্যবহার হয় না। যদি একই syntax ব্যবহার করা হয়, তবে সে ক্ষেত্রে চলকের মান নতুনভাবে ইনপুট দিতে হবে।

৪. ফাংশনের আউটপুট কী রকম হবে, সেটি আগে থেকেই নির্ধারণ করে দিতে হবে।

একটি ফাংশনের গঠন কী রকম হবে, সেটি নিচে দেখানো হলো।

```
রিটার্ন টাইপ ফাংশনের নাম (চলক ১, চলক ২, চলক ৩, ....., যতগুলো চলক
প্রয়োজন ফাংশনের কাজে)
{
   ফাংশনের কাজগুলো অর্থ যা যা করবে, সেগুলো
   ফাংশনটি কী কী রিটার্ন করবে
}
```

একটি উদাহরণ দেওয়া হলো—

```
#include <stdio.h>
void print_name ()
{
    printf ("I am Asimo . I am a robot");
}
int main ()
{
    print_name();
}
```

এখানে ফাংশনের টাইপে void লেখা হলো। কারণ এখানে আসলে কোনো সংখ্যা রিটার্ন করছে না। তাই এ ক্ষেত্রে void হিসেবে বোঝানো হলো।

রিটার্ন টাইপ কী রকম হবে তার ওপরে ফাংশনের আউটপুট নির্ভর করে। এখানে আমি আরও ২টি ভিন্ন ভিন্ন রিটার্ন টাইপের উদাহরণ দিচ্ছি।

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
int math (int a, int b, int c)//এখানে int দিয়ে বোঝানো হয়েছে যে ডেসিমাল সংখ্যা
আউটপুট হিসেবে আসবে
```

```
{
    switch (c)
    {
    case 1 :
        return a+b;  //c-এর মান ১ এই শর্ত পূরণে যে আউটপুট আসবে
        break;
    case 2 :
       return a-b;  // c-এর মান ২ এই শর্ত পূরণে যে আউটপুট আসবে
        break;
    case 3 :
        return a*b; // c-এর মান ৩ এই শর্ত পূরণে যে আউটপুট আসবে
        break;
    case 4 :
        return a/b; // c-এর মান ৪ এই শর্ত পূরণে যে আউটপুট আসবে
        break;
    }
}
int main ()
{
   int a, b, c;
    printf ("Enter 2 numbers  : \n");
    scanf ("%d", &a);
    scanf ("%d", &b);
    printf ("For add enter 1, for subtraction enter 2, for multiplication enter 3,
for division enter 4  : ");
    scanf ("%d", &c);
    printf ("Your result is %d", math(a, b, c));
}
```

এখানে পাশে // এই লাইনগুলোর পাশে বুঝিয়ে দেওয়া হয়েছে ঘটনাগুলো। কোডটি চালিয়ে দেখলেই বোঝা যাবে।

কিন্তু এখানে একটি জিনিস লক্ষ করার মতো। ফাংশনে রিটার্ন টাইপ যদি দশমিক আউটপুট হিসেবে পাওয়ার ইচ্ছা থাকত, তবে সে ক্ষেত্রে int-এর বদলে float ব্যবহার করলেই চলবে। একই কোড শুধু এভাবে পরিবর্তন করতে হবে।

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
float math (float a, float b, float c)
```

```
{
    switch (c)
    {
    case 1 :
        return a+b;
        break;
    case 2 :
       return a-b;
        break;
    case 3 :
        return a*b;
        break;
    case 4 :
        return a/b;
        break;
    }
}
int main ()
{
    float a, b, a1;
    int c;
    printf ("Enter 2 numbers : \n");
    scanf ("%f", &a);
    scanf ("%f", &b);
    printf ("For add enter 1, for subtraction enter 2, for multiplication enter 3,
for division enter 4 : ");
    scanf ("%d", &c);
    a1=math(a, b, c);
    printf ("Your result is %f", a1);
}
```

লক্ষণীয় যে দশমিকের জন্য '%f' এটি ব্যবহার করতে হয়। যেখানে সাধারণ ডেসিমাল সংখ্যার জন্য '%d' ব্যবহার করতে হবে।

ফাংশন সম্পর্কে যাবতীয় আলোচনা এতটুকুই। আশা করি বোঝাতে পেরেছি। এতটুকু ধারণা থাকলেই মোটামুটি অপারেশন করা যাবে।

সি প্রোগ্রাম সম্পর্কে যতটুকু জানা প্রয়োজন ছিল, ততটুকুই করলাম। আরও জানার ইচ্ছা থাকলে সি প্রোগ্রামের ওপরে বই থেকে জানা যেতে পারে। মাইক্রোকন্ট্রোলার প্রোগ্রামিংয়ের জন্য সি সম্পর্কে এতটুকু

ধারণা থাকলেই চলবে। পরবর্তী অধ্যায় থেকে Arduino and Avr নিয়ে প্রোগ্রামিং বিষয়ক আলোচনা শুরু করা হবে।

Arduino ও Microcontroller প্রোগ্রামিং

অবশেষে আমাদের বহুল প্রতীক্ষিত প্রোগ্রামিংয়ের অংশটি এল। Arduino মাইক্রোকন্ট্রোলার প্রোগ্রামিং সবচেয়ে বেশি ব্যবহার করা সিস্টেম। এ কারণে এটি সবচেয়ে ব্যবহার করা হয়। আর ব্যবহার করা সার্কিট বানানো থেকে শুরু করে প্রোগ্রামিং করাটাও অনেক সহজ। যে কারণে এটি অনেক বহুল প্রচলিত সারা পৃথিবীতে।

এই অধ্যায়য়ে Arduino নিয়ে যেসব বিষয় আলোচনা করা হবে সেগুলো হলো—

১. অ্যানালগ ও ডিজিটাল ইনপুট-আউটপুট সম্পর্কে।
২. বিভিন্ন রকমের মোটর প্রোগ্রামিং করে চালানো ও এ-সম্পর্কে জানা।
৩. LCD, Keypad এই ২টি সম্পর্কে বিশেষভাবে বলা হবে।
৪. PWM Signal সম্পর্কে জানা।
৫. সেন্সর থেকে ইনপুট নেওয়া।
৬. সেন্সর থেকে ইনপুটের ডেটার তথ্যের ভিত্তিতে উপযুক্ত আউটপুট তৈরি করা বলা চলে অনেকটাই রোবটিক সিস্টেম তৈরি করা।
৭. সবশেষে থাকবে কয়েকটি প্রজেক্ট।

এ ছাড়া এখানে আরও এক ধরনের মাইক্রোকন্ট্রোলার প্রোগ্রামিং সম্পর্কে ধারণা দেওয়া হবে। বাইনারি সিস্টেমে কীভাবে করা যায় Arduino প্রোগ্রামিং।

পরবর্তী অধ্যায়ে আমরা Arduino প্রোগ্রামিংয়ে প্রথমে জানব কীভাবে ডিজিটাল এবং অ্যানালগ ইনপুট ও আউটপুট তৈরি করা যায়।

অ্যানালগ ও ডিজিটাল ইনপুট ও আউটপুট

অ্যানালগ ও ডিজিটাল ইনপুট ও আউটপুট সম্পর্কে জানার আগে আমাদের জানতে হবে অ্যানালগ ও ডিজিটাল সিগন্যাল কী এবং এটি কীভাবে তৈরি করা যায়।

অ্যানালগ ও ডিজিটাল এই ২টি শব্দ শুনলে মনে হয় অ্যানালগ বলতে সেই পুরানো কালের যন্ত্রের কৌশল আর ডিজিটাল মানে আধুনিককালের। বাস্তবিক পক্ষে সংজ্ঞাটি একটু ভিন্ন। প্রাচীনকালের ডায়ালযুক্ত কাঁটাওয়ালা মেশিন, যেগুলো নিজ হাতে অপারেট করতে হতো, সেগুলোই আমরা অ্যানালগ মেশিন হিসেবে চিনি।

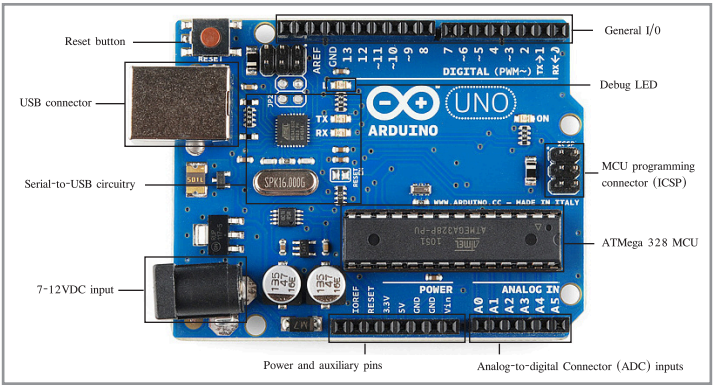
কিন্তু এখানে আমরা যে অ্যানালগ সম্পর্কে কথা বলছি, সেটা ওরকম অ্যানালগ মেশিন না। ধরা যাক একটি সেন্সর যেটি এক অর্থে একপ্রকার রোধক হিসেবে কাজ করে এবং যেই রোধকের মান সময়ের এবং পরিবেশের সাথে সাথে পরিবর্তন হয়। এখন পরিবেশের পরিবর্তনের সাথে সাথে আমি যদি পাঠ নিই, তবে বিভিন্ন সময় বিভিন্ন মানের রোধকের জন্য ভিন্ন ভিন্ন পাঠ পাওয়া যাবে। এই যে ভিন্ন ভিন্ন পাঠ পেলাম এই পাঠগুলো যদি সময়ের সাথে সাথে একটি গ্রাফে প্লট করি, তবে একটি কার্ভ পাওয়া যাবে। এটি অ্যানালগ সিগন্যাল। এ-সম্পর্কে আগেই আলোচনা করা হয়েছে।

ডিজিটাল সিগন্যাল অনেকটা এ রকম— ভিন্ন ভিন্ন সময় শুধু ১ অথবা ০ এ রকম ভোল্টেজ আউটপুট দেয়। ৫ ভোল্টের জন্য ১ আর ০ ভোল্টের জন্য যদি ০ ধরি, তবে সে ক্ষেত্রে ১০১১০১০১০১০১০১০১০১০১১১১০ এ রকম একটি আউটপুট পাওয়া গেল যেখানে পাঠের ভিন্নতা পাওয়া যাবে সময়ের সাথে সাথে। এ সিগন্যালটি ডিজিটাল সিগন্যাল।

এ নিয়ে বইয়ে এর আগেই আলোচনা করা হয়েছে।

এখন আমি এটাকে Arduino প্রোগ্রামিংয়ে কীভাবে ব্যবহার করা যাবে সেটি দেখাব।

কাজের জন্য অনেকের মনে প্রশ্ন থাকতে পারে কোন মডেলের Arduino ব্যবহার করতে হবে। এখানে আমি যে Arduino মডেল নিয়ে কাজ করব, সেটি হচ্ছে Arduino Uno। এটি দামেও কম আর সহজলভ্য।



চিত্র ২.২৫ : আরডুইনো ইউএনও বোর্ড

চিত্র ২.২৫-এ একটি বোর্ড দেখা যাচ্ছে। এখানে অ্যানালগ পোর্ট A চিহ্নিত করা। A0, A1, A2, A3, A4 ও A5। এখানে অ্যানালগ পোর্টগুলোর অর্থ হচ্ছে যে এখান থেকে একই সাথে অ্যানালগ সিগন্যাল তৈরি করা যাবে আউটপুট হিসেবে আবার তার সাথে সাথে সিগন্যালকে ইনপুট হিসেবেও নেওয়া যাবে।

বোর্ডে মোট ১৪টি ডিজিটাল পিন দেওয়া। এখানে ডিজিটাল পিনগুলো শুধু সংখ্যা দিয়ে চিহ্নিত করা। ০, ১, ২, ৩, ৪ থেকে ১৩ পর্যন্ত। কতগুলো পিনের পাশে টাইল্ড (~) চিহ্ন দেওয়া। এগুলো বিশেষ ধরনের ডিজিটাল পোর্ট। এর কাজ অনেক। সার্ভো মোটর চালানো আবার অ্যানালগ সিগন্যাল প্রস্তুত করতেও এটি ব্যবহার করা হয়।

প্রয়োজনীয় ফাংশন ও এর ব্যবহার—

Function Name	Working Pins	Working Principle
digitalWrite (pin number, High/Low)	সব ডিজিটাল পিন এমনকি PWM ও এর অন্তর্গত	এর কাজ হচ্ছে ডিজিটাল সিগন্যাল আউটপুট হিসাবে দেওয়া। বলা চলে এটি শুধু ০ ভোল্ট অথবা ৫ ভোল্ট আউটপুট দেয়।
digitalRead (pin number)	সব ডিজিটাল পিন এমনকি PWM ও এর অন্তর্গত	এর কাজ হচ্ছে ডিজিটাল সিগন্যাল ইনপুট হিসেবে নেওয়া। একটি সার্কিট থেকে ভোল্টেজ ইনপুট নেবে।

analogWrite (pin number, 0-255)	সব অ্যানালগ পিন A0, A1, A2 ..... ইত্যাদি	অ্যানালগ সিগন্যাল আউটপুট দেওয়া। এ মান ০ ভোল্ট থেকে ৫-এর মধ্যে ২৫৫টি ভাগে ভাগ করা থাকে।
analogRead (pin number)	সব অ্যানালগ পিন A0, A1, A2 ..... ইত্যাদি	সেন্সর থেকে ইনপুট নেওয়া। সেন্সর থেকে যে ভোল্টেজ আউটপুট হিসেবে পাওয়া যায় সেটি ইনপুট নেয়। এর মান অবশ্যই ০ থেকে ৫ ভোল্ট হওয়া উচিত।

ছক ২.৫ : আরডুইনো প্রোগ্রামিংয়ে ব্যবহৃত কিছু ফাংশন

এখানে চার্টটিতে দেখালাম অ্যানালগ ও ডিজিটাল ইনপুট আউটপুট হিসেবে কী রকম এবং কোন কোন পিনে ব্যবহার করা হয়।

এখন আমরা দেখাব কীভাবে সার্কিট প্রস্তুত করে Arduino-তে প্রোগ্রাম আপলোড দিয়ে সেটি চালানো যায়।

প্রথমে দেখতে হবে digitalWrite ফাংশন কীভাবে ব্যবহার করতে হয়। এ অনুযায়ী সার্কিট তৈরি করা এবং সার্কিট থেকে প্রয়োজনীয় আউটপুট বের করা। আউটপুট হিসেবে কিছু LED বাল্ব ব্যবহার করা হবে। এ ছাড়া প্রয়োজনীয় সার্কিট ডায়াগ্রাম দেওয়া হবে।

তৃতীয় অধ্যায়

প্রজেক্ট (প্রথম অংশ)

প্রজেক্ট ১ : ডিজিটাল আউটপুটের সাহায্যে LED বাল্ব জ্বালানো

প্রয়োজনীয় উপকরণ—

১. Arduino Uno

২. কিছু LED বাল্ব

৩. জাম্পার তার এই প্রজেক্টের জন্য আমরা Male to male জাম্পার ব্যবহার করব।

৪. ব্যাটারি ৯ ভোল্টের হলে ভালো হয়। এর বেশি না হওয়াটাই ভালো। Lipo ব্যাটারি নিয়ে পরে কথা হবে

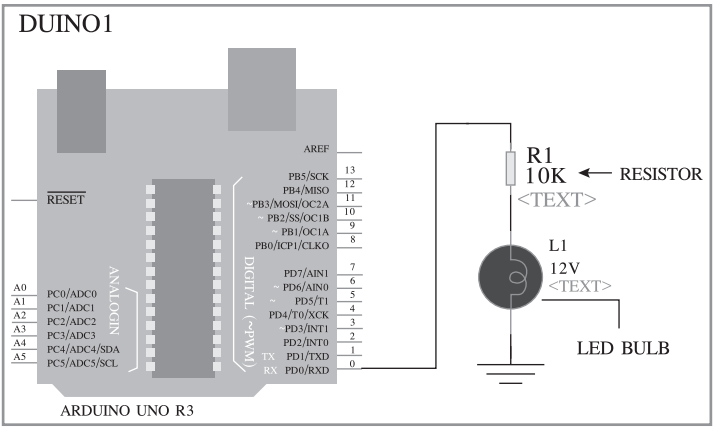
৫. ব্যবহারের জন্য একটি কম্পিউটার ও প্রোগ্রাম আপলোডের জন্য একটি উপযুক্ত তার যেটি Arduino Uno কেনার সময় দেওয়া হয়। সাধারণত নীল রঙের হয় এটি।

৬. বিভিন্ন মানের কিছু রেসিস্টর।

৭. ব্রেডবোর্ড।

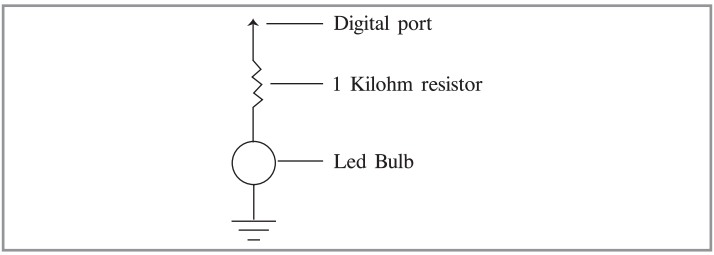
এখানে বইয়ের শেষে সব উপকরণের একটি বড় লিস্ট দেওয়া হবে। যাতে বইয়ের যেকোনো প্রজেক্ট তৈরিতে সমস্যা তৈরি না হয়।

সার্কিট ডায়াগ্রাম



চিত্র ৩.১ : প্রধান সার্কিট ডায়াগ্রাম

এখানে পুরো সার্কিট ডায়াগ্রাম দেওয়া হলো। এখন একটি Arduino Uno হাতে নেই। এর ডিজিটাল পোর্ট যেগুলো আছে ০ থেকে শুরু করে ১৩ পর্যন্ত এর মধ্যে যেকোনো একটি পোর্ট বেছে নিই। যে পোর্ট দিয়ে প্রোগ্রামিং করব।



চিত্র ৩.২ : সিম্পল সার্কিট ডায়াগ্রাম

এখানে আমরা সার্কিটের একটি সিম্পল ডায়াগ্রাম দেখলাম। সবচেয়ে শেষে যেটি ৩টি সমান্তরাল দাগ দেওয়া বড় থেকে ছোট সেটি হচ্ছে গ্রাউন্ড। মাঝে একটি রেসিস্টর ১ কিলো ওহম এবং একটি LED বাল্ব সিরিজে লাগানো। এখানে মনে রাখতে হবে ১ কিলো ওহম রেসিস্টর না লাগালেও চলত। লাগালাম এই কারণে যাতে Arduino Uno থেকে আসা অতিরিক্ত কারেন্ট LED বাল্বের ক্ষতি না করতে পারে। এখানে তীর চিহ্ন অংশটি দিয়ে ডিজিটাল পোর্ট থেকে আউটপুট বোঝাচ্ছে।

এখন ধরা যাক যে ডিজিটাল পোর্ট আউটপুট দেবে সেটি হচ্ছে ০ নম্বর পোর্ট ডিজিটাল। এখন একটি Male to male জাম্পার নিয়ে আমরা Arduino Uno এর ০ নম্বর ডিজিটাল পোর্টে লাগাই। এরপরে একটি রেসিস্টর ১ কিলোওহম এবং একটি LED বাল্ব সিরিজে লাগাই। তারপরে সর্বশেষে LED বাল্বের ঋণাত্মক প্রান্ত Arduino Uno-এর GND অথবা Ground লেখা অংশে সংযুক্ত করি। LED বাল্বের ধনাত্মক এবং ঋণাত্মক প্রান্ত কোনটি হবে সেটি দেখে নির্ণয় করা যায়। লম্বা মাথা যেটি, সেটি হচ্ছে ধনাত্মক প্রান্ত আর ছোট করে মাথাটি ঋণাত্মক। তো সংযোগ হয়ে গেলে আমরা প্রোগ্রাম আপলোড দেব।

প্রথমেই বলে রাখি। এখানে সি-এর সব প্রোগ্রামের মতো হেডার ফাইল লাগে না। আর সি-এর মতো int main () নেই। এখানে স্ট্রাকচারটি এ রকম—

```
void setup() {
কোডের এই অংশে উল্লেখ করতে হবে কোন পোর্টের সাথে কোন কোন সার্কিটের কোন
কোন অংশ যুক্ত হবে। এটা পরে বিভিন্ন কোড দেখতে দেখতে বোঝা যাবে সহজেই।
}
void loop() {
  এটি ইনফাইনাইট লুপ। মানে কখনো শেষ হবে না এমন। এটি অনেকটাই সি-এর
মেইন ফাংশনের মতো।
এখানে আসল কাজের ধারা উল্লেখ করতে হবে।
}
```

মেইন প্রোগ্রাম

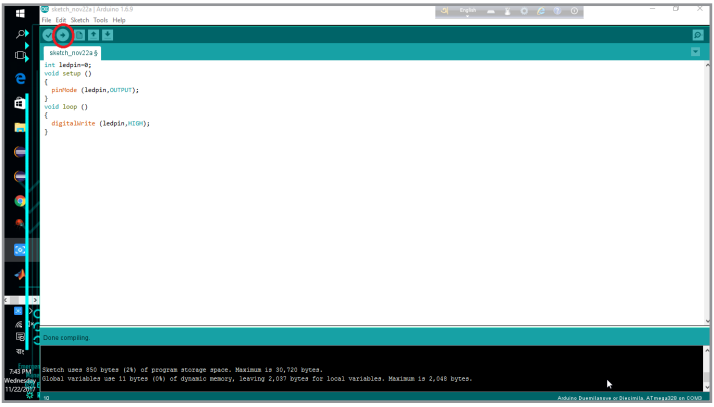
মেইন প্রোগ্রামে void setup-এ আমরা একটি ফাংশন ব্যবহার করব। pinMode (pin number, Input/Output)

এখানে pin number বসবে, আসলে আমরা কত নম্বর পিন দিয়ে ইনপুট অথবা আউটপুট নেব। আর Input/Output ব্যবহার করতে হবে সব ক্ষেত্রেই। যদি পিন থেকে সিগন্যাল আউটপুট হিসেবে বের হয়, তবে সে ক্ষেত্রে Output লিখতে হবে। আর যদি সেন্সর থেকে ভোল্টেজ ইনপুট হিসেবে প্রবেশ করে, তবে সে ক্ষেত্রে Input লিখতে হবে। এখন তাহলে মেইন প্রোগ্রামটি দেখি।

```
int ledpin=0; // এখানে আমরা ০ নম্বর পিনকে ব্যবহার করছি
void setup ()
```

```
{
  pinMode (ledpin, OUTPUT); // এখানে ০ নম্বর পিন থেকে আউটপুট সিগন্যাল বের হচ্ছে
}
void loop ()
{
  digitalWrite (ledpin, HIGH); // HIGH ব্যবহার করা হয়েছে। তার কারণ LED কে জ্বালানো হবে
}
```

২টি জিনিস মনে রাখতে হবে। OUTPUT, INPUT, HIGH, LOW এই কয়টি শব্দের সবগুলো অক্ষর বড় হাতের হতে হবে। প্রোগ্রাম লেখা হয়ে গেলে আপলোড দিতে হবে। প্রথমে ওপরে tools অপশনে ক্লিক করতে হবে। এরপরে এখানে Board অপশনে সিলেক্ট করতে হবে Arduino/Genuino Uno। এরপরে আবার tools অপশনে ক্লিক করে port সিলেক্ট করতে হবে। port অপশনের ওপরে কার্সর রাখলে দেখা যাবে com (1-90) ১ থেকে ৯০-এর মধ্যে একটি সংখ্যা। ধরা যাক com 11। com 11-এর পাশে একটি ব্র্যাকেটে Arduino/Genuino Uno লেখা থাকতে পারে আবার না-ও থাকতে পারে। যে com পোর্ট নম্বর দিয়ে লেখা সেটি ক্লিক করতে হবে। যদি একের অধিক Arduino থাকে, তবে সে ক্ষেত্রে ব্যাপারটি ভিন্ন।



চিত্র ৩.৩ : আপলোড করার জন্য তীর চিহ্নে ক্লিক করতে হবে

বোর্ড সেটআপ ঠিক করা হয়ে গেলে তারপরে ওপরে তীর চিহ্ন দেওয়া বাটনে ক্লিক করলেই প্রোগ্রাম আপলোড হবে এবং এখানে আউটপুট হিসেবে দেখা যাবে LED বাল্ব শুধু জ্বলে থাকবে।

প্রজেক্ট ১-২

একটি নির্দিষ্ট সময় পরপর LED বাল্ব জ্বলা এবং নেভা।

এখানে আমরা একটি নতুন ফাংশন সম্পর্কে জানব। এই নতুন ফাংশনটিকে বলা চলে টাইমের একটি ফাংশন। এর কাজ আসলে একটি নির্দিষ্ট সময়ের জন্য এটি বন্ধ রাখা। ফাংশনটি হচ্ছে delay ()। এর লেখার পদ্ধতি হচ্ছে এমন delay () এর ব্র্যাকেটের ভেতরে একটি সংখ্যা বসবে। এই সংখ্যাটি বোঝায় আসলে মিলিসেকেন্ড। অর্থ কত মিলিসেকেন্ড থামবে। ধরা যাক যদি ১০০০ লিখি, তবে এই ফাংশনের আগের কাজ এবং পরের কাজের মধ্যবর্তী সময়ের পার্থক্য ১০০০ মিলিসেকেন্ড অথবা ১ সেকেন্ড।

এবার প্রোগ্রামটি চালাই।

```
int ledpin=0;
void setup ()
{
  pinMode (ledpin, OUTPUT);
}
void loop ()
{
  digitalWrite (ledpin, HIGH);          // LED বাল্ব জ্বলা
  delay (1000);                   // ১০০০ মিলিসেকেন্ড বা ১ সেকেন্ড থামা
  digitalWrite (ledpin, LOW);         // LED বাল্ব নেভা
  delay (1000);                   // ১০০০ মিলিসেকেন্ড বা ১ সেকেন্ড থামা
}
```

আগের সার্কিটের সেটআপে এই কোডটি আপলোড দিলে LED বাল্বটি একবার জ্বলবে এবং আরেকবার নিভবে। জ্বলা ও নেভার মধ্যবর্তী সময় ১ সেকেন্ড।

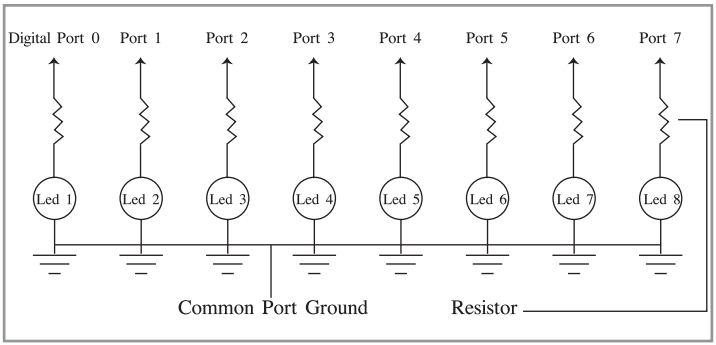
তাহলে আমরা খুবই সহজ ২টি কোডের মাধ্যমে জানলাম ২টি ঘটনা। প্রথম প্রজেক্টটি হয়তো অনেকের কাছে হাস্যকর মনে হতে পারে। কারণ শুধু একটি লাইট জ্বালিয়েছি আমরা। কিন্তু ২ নম্বর কোডে আমরা জানলাম কীভাবে কোনো সুইচের সাহায্য ছাড়া একটি LED বাল্বের জ্বলা ও নেভা কন্ট্রোল করা যাবে।

আমাদের এর পরের প্রজেক্টটি হবে আরও কতগুলো LED বাল্বের সমন্বয়ে। এই কাজের জন্য আমাদের লাগবে মোট ৮টি LED বাল্ব।

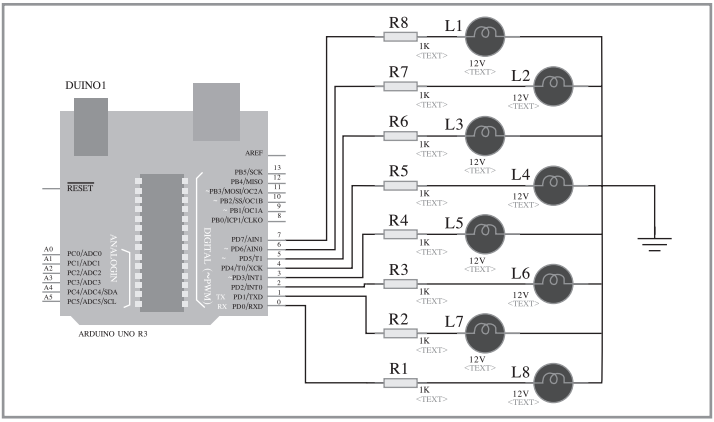
এই প্রজেক্টটি তৈরি করলে আমাদের ডিজিটাল আউটপুটের ব্যাপারে সবকিছু পরিষ্কার হবে। এখানে মোট ৮টি LED বাল্ব ব্যবহার করা হবে। আমাদের প্রজেক্টটি হচ্ছে এই ৮টির মধ্যে একবার ৪টি জ্বলা তখন অন্য ৪টি নেভা অবস্থায় থাকা এরপরে ১ সেকেন্ড পরে, যে ৪টি নিভে ছিল সে ৪টি জ্বলা এবং আগের ৪টি নেভা অর্থাৎ অনেকটাই ফ্লিপফ্লপ।

এখানে সার্কিট সাজাতে হবে এভাবে।

যেহেতু Arduino Uno এর ৮টি ডিজিটাল পোর্ট ব্যবহার করা হবে। ধরা যাক ৮টি পোর্ট হচ্ছে ডিজিটাল পোর্ট ০ থেকে শুরু হয়ে ৭ পর্যন্ত এই কয়টি ডিজিটাল পোর্ট। প্রথমে সব কয়টি ডিজিটাল পোর্ট থেকে জাম্পার তার সংযোগ করতে হবে রেসিস্টরের সাথে। তারপর এই রেসিস্টরের সাথে ৮টি LED বাল্বের ধনাত্মক প্রান্ত সংযোগ দিতে হবে। এরপরে ৮টি LED বাল্বের ঋণাত্মক প্রান্ত সবগুলো একটি ব্রেডবোর্ডের কমন পোর্টের সাথে সংযোগ দিতে হবে। যেখান থেকে একটি জাম্পার তার যাবে Arduino Uno-এর Ground এ। মোট কথা ৮টি LED বাল্বের ঋণাত্মক প্রান্ত Arduino Uno-এর Ground-এ সংযোগ হবে।



চিত্র ৩.৪ : চারটি বাল্ব জ্বলা এবং চারটি বাল্ব নেভার জন্য প্রয়োজনীয় সার্কিট



চিত্র ৩.৫ : সিমুলেটেড সার্কিট ডায়াগ্রাম

এখানে চিত্র ৩.৪ ও ৩.৫-এর মাধ্যমে সার্কিট ডায়াগ্রাম বোঝা যাবে সঠিকভাবে। চিত্র ৩.৫-এ Arduino Uno-এর সাথে সংযোগ দেখানো হলো। এখন প্রোগ্রামিং করার পালা। Ground-এর অংশটি Arduino Uno-এর Gnd-এর সাথে সংযোগ দিতে হবে।

```
int ledpin1=0;
int ledpin2=1;
int ledpin3=2;
int ledpin4=3;
int ledpin5=4;
int ledpin6=5;
int ledpin7=6;
int ledpin8=7;

void setup ()
{
  pinMode (ledpin1, OUTPUT);
  pinMode (ledpin2, OUTPUT);
  pinMode (ledpin3, OUTPUT);
  pinMode (ledpin4, OUTPUT);
  pinMode (ledpin5, OUTPUT);
  pinMode (ledpin6, OUTPUT);
  pinMode (ledpin7, OUTPUT);
  pinMode (ledpin8, OUTPUT);
}
```

```
void loop ()
{
  digitalWrite (ledpin1, HIGH);
  digitalWrite (ledpin3, HIGH);
  digitalWrite (ledpin5, HIGH);
  digitalWrite (ledpin7, HIGH);
  digitalWrite (ledpin2, LOW);
  digitalWrite (ledpin4, LOW);
  digitalWrite (ledpin6, LOW);
  digitalWrite (ledpin8, LOW);
  delay (1000);
  digitalWrite (ledpin1, LOW);
  digitalWrite (ledpin3, LOW);
  digitalWrite (ledpin5, LOW);
  digitalWrite (ledpin7, LOW);
  digitalWrite (ledpin2, HIGH);
  digitalWrite (ledpin4, HIGH);
  digitalWrite (ledpin6, HIGH);
  digitalWrite (ledpin8, HIGH);
  delay (1000);
}
```

আগের মতোই। এখানে প্রোগ্রাম আপলোডের আগে tools অপশনে গিয়ে বোর্ড চেক করতে হবে। আর সাথে পোর্ট কত এবং ঠিকমতো পোর্ট সিলেক্ট করা আছে নাকি, সেটাও দেখতে হবে।

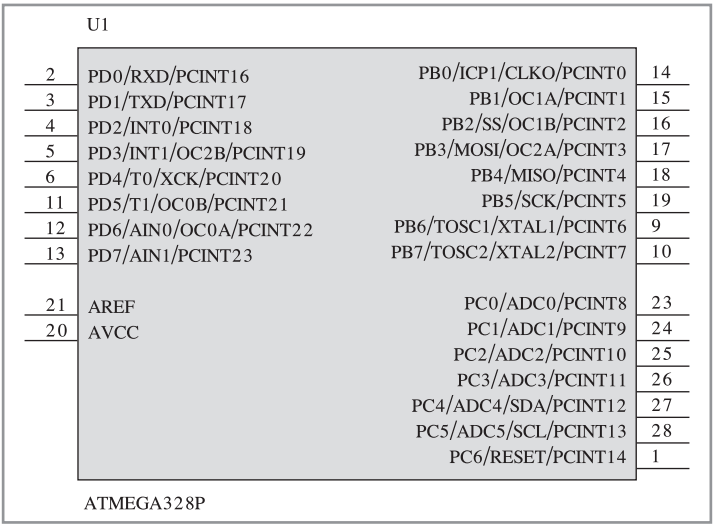
সব দেখা হলে প্রোগ্রাম আপলোড দিতে হবে। প্রোগ্রাম আপলোড দিলেই সার্কিটটি চালু হবে। সব ঠিকঠাক থাকলে অনেক সুন্দর আউটপুট দেখা যাবে।

এখানে Led বাল্বগুলো ফ্লিপফ্লপ করতে থাকবে। অর্থাৎ প্রথমে ১, ৩, ৫, ৭ এই ৪টি জ্বলবে তখন ২, ৪, ৬ এবং ৮ এই ৪টি বন্ধ থাকবে। তার ঠিক ১ সেকেন্ড পরে ২, ৪, ৬ আর ৮ জ্বললে ১, ৩, ৫, ৭ এই ৪টি Led বাল্ব বন্ধ থাকবে। এভাবে এটি সব সময় চলতে থাকবে। শুধু ব্যাটারি সংযোগ বিচ্ছিন্ন করলেই এটির সংযোগ বিচ্ছিন্ন হবে। সময়ের পরিমাণ বাড়িয়ে কমিয়ে প্রোগ্রাম আপলোড দিয়ে আবার পরীক্ষা করা যেতে পারে। একটা জিনিস মনে রাখতে হবে যে প্রতিবার কোড পরিবর্তনের জন্য নতুন করে আবার আপলোড দিতে হবে। না হলে কাজ হবে না।

এরপরের প্রজেক্টটিতে আমরা দেখব অ্যানালগ আউটপুট সম্পর্কে। এ-সম্পর্কে বিস্তারিত আলোচনা করা হবে।

প্রজেক্ট ২ : অ্যানালগ আউটপুটের সাহায্যে LED বাল্ব জ্বালানো

অ্যানালগ সিগন্যাল সম্পর্কে আমরা অনেক আগেই জেনেছি। ডিজিটাল সিগন্যাল থেকে আউটপুট যেমন বাইনারিতে বের হয় কিন্তু অ্যানালগ সিগন্যাল থেকে আউটপুট একটি নির্দিষ্ট মান থেকে অন্য একটি নির্দিষ্ট মানের মধ্যে থাকে। মানের শুরু হয় ০ থেকে এবং শেষ হবে ২-এর কোনো একটি ঘাতের মান থেকে ১ কম। ২-এর ঘাত কত হবে সেটি নির্ভর করে মাইক্রোকন্ট্রোলার কত বিটের তার ওপরে। ধরা যাক এটি ৮ বিটের মাইক্রোকন্ট্রোলার। তাহলে অ্যানালগ আউটপুটের মান বের হবে ০ থেকে ২৫৫-এর মধ্যে। ০ থেকে শুরু করলে ২৫৫ তে শেষ হবে। আমরা Arduino Uno ব্যবহার করছি। এখানে যে মাইক্রোকন্ট্রোলার ব্যবহার করা হচ্ছে সেটি হচ্ছে ৮ বিটের Atmega 328p।



চিত্র ৩.৬ : ৮ বিটের Atmega 328p

এখানে দেখা যাচ্ছে প্রতিটি পোর্টে মোট ৮টি করে রেজিস্টর। এই ৮টি করে রেজিস্টর প্রত্যেকে অর্থাৎ প্রত্যেক পোর্ট থেকে সর্বোচ্চ ২৫৫ আউটপুট বের করা সম্ভব।

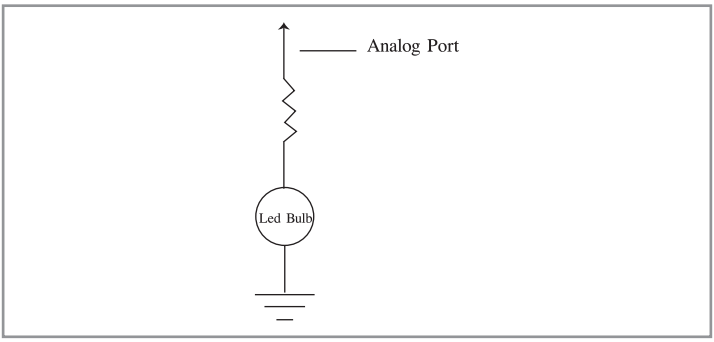
এখন অ্যানালগ আউটপুটের সিস্টেম সম্পর্কে আগেই বলেছি একটি ছকে। আগের যে সার্কিট আছে সেটিতেই প্রোগ্রাম করা যাবে। সার্কিট ডায়াগ্রাম একই। কোনো পার্থক্য নেই। এখানে একটি সার্কিট ডায়াগ্রাম দেওয়া হয়েছে এ থেকেই পরিষ্কার হবে এটি।

Arduino Uno তে অ্যানালগ পোর্ট আছে মোট ৬টি। একটি Arduino Uno নিলে কতগুলো পোর্ট দেখা যাবে A0, A1, A2, A3, A4 এবং A5 এই ৬টি হচ্ছে অ্যানালগ পোর্ট।

এখন যেকোনো একটি পোর্ট বেছে নিই।

ধরা যাক পোর্টটি হচ্ছে ০ নম্বর পোর্ট।

এখন আমরা যে প্রোগ্রামটি চালাব সেটি হচ্ছে একটি Led বাল্বের উজ্জ্বলতা নিয়ন্ত্রণ করা। প্রোগ্রামের আউটপুট এমন হবে যে Led বাল্বটি একবার নেভা থেকে জ্বলে একদম সবচেয়ে বেশি উজ্জ্বলতায় পৌঁছাবে। এরপরেই আবার ওইখানে থেকে আসতে আসতে উজ্জ্বলতা কমতে কমতে একদম নিভে যাবে।



চিত্র ৩.৭ : সিম্পল সার্কিট ডায়াগ্রাম

প্রোগ্রামে আমরা সময়ের ব্যবধান রাখব ৫০ মিলিসেকেন্ড। এত কম কেন সেটি বলছি। প্রোগ্রামটি এ রকম।

```
int ledpin = A0;
int i;

void setup ()
{
  pinMode (ledpin, OUTPUT);
```

```
}
void loop ()
{
  for (i=0;i<=255;i++)
  {
    analogWrite (ledpin, i);
    delay (50);
  }
  for (i=255;i>=0;i--)
  {
    analogWrite (ledpin, i);
    delay (50);
  }
}
```

এখানে চলক i দিয়ে আসলে বোঝা যাচ্ছে কী পরিমাণে ভোল্টেজ আউটপুট হবে। ০ তে আউটপুট হবে ০ ভোল্ট আর ২৫৫ তে আউটপুট হবে ৫ ভোল্ট। মাঝে ০ থেকে ৫ ভোল্টকে ২৫৫টি সমান ভাগে ভাগ করা হয়েছে।

সময়ের পার্থক্য ৫০ মিলিসেকেন্ড। এখানে লুপের ভেতরে একবার পার হলে ৫০ মিলিসেকেন্ড সময় নেয়। তাহলে ২৫৬ বারে সময় নেবে ২৫৬*৫০ = ১২৮০০ মিলিসেকেন্ড অথবা ১২.৮ সেকেন্ড। অর্থাৎ একবার পুরো উজ্জ্বলতায় যেতে সময় নেবে প্রায় ১২.৮ সেকেন্ডের মতো। তাহলে ২ বারে মোট সময় লাগবে ২৫.৬ সেকেন্ড। একটু লম্বা সময়। এ জন্যই সময়ের পার্থক্য অনেক কম নেওয়া হয়েছে।

প্রজেক্ট ৩ : ডিজিটাল ও অ্যানালগ ইনপুট ও এর কাজ

এর আগে আমরা ডিজিটাল আউটপুট সম্পর্কে জেনেছি। এবার আমরা জানব ডিজিটাল ইনপুট এবং একে কীভাবে ব্যবহার করা যায়।

এর আগে কয়েকটি কথা বলে নেব। এ অংশটিতে আমরা ইনপুট এবং এর লজিকের মাধ্যমে আউটপুট বের করার ব্যাপারটি শিখব। রোবটিকস যে আমরা শুরু করব তার মূলভিত্তি বলা চলে আসলে এটি এবং এর মাধ্যমেই এ সম্পর্কে বিস্তারিত আলোচনা শেষে আমরা একটি ছোটখাটো রোবট তৈরি করব।

ইনপুট ও আউটপুটের বেসিক

ইনপুট ও আউটপুটের বেসিক সম্পর্কে বলার আগে একটি কথা মনে করিয়ে দিই। এর আগে আমরা সেন্সর সম্পর্কে পড়েছি যে সেন্সর কী কীভাবে কাজ করে। এখানে আমাদের সেন্সরের বেসিক কাজে লাগবে।

সেন্সর থেকে আমরা কী করতাম? সেন্সর একটি ভোল্টেজ ডিভাইডার হিসেবে কাজ করে। এখানে পরিবেশের পরিবর্তনের সাথে সাথে আউটপুট ভোল্টেজের মানও পরিবর্তন হয়। এভাবে ওই আউটপুট থেকে আমরা পাঠ নিই। এই যে পরিবর্তনশীল রোধ এটাই হচ্ছে অ্যানালগ মান বলা চলে।

ডিজিটাল ইনপুটটি কী রকম?

অ্যানালগ সম্পর্কে আমরা যেমন জানি যে আরডুইনোতে অ্যানালগ পাঠ কাজ করে ০ ভোল্ট থেকে ৫ ভোল্টের মধ্যে ২৫৫টি ভাগে ভাগ করা হয়, তেমনিভাবে ডিজিটালের ক্ষেত্রে সেটি শুধু ০ ভোল্ট আর ৫ ভোল্টের মধ্যেই সীমাবদ্ধ।

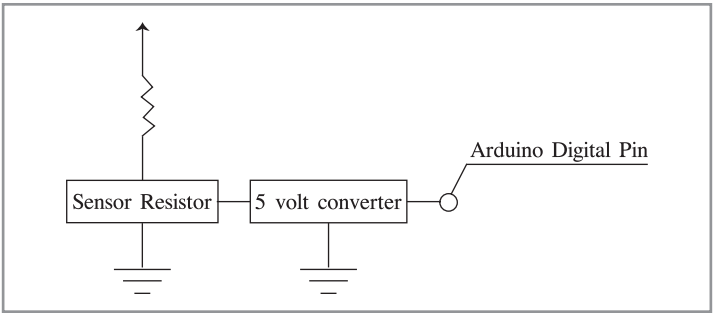
প্রথমে বলি ডিজিটাল ইনপুট সম্পর্কে।

ডিজিটালের ক্ষেত্রে যেহেতু ০ ভোল্ট এবং ৫ ভোল্টের মধ্যে সীমাবদ্ধ সে জন্য ব্যাপারটি একটু কঠিন। অনেক ক্ষেত্রে এনকোডার ডিকোডার অথবা বাক বা বুস্ট কনভার্টার ব্যবহার করা হয়।

আবার অপারেশনাল অ্যামপ্লিফায়ারও ব্যবহার করা হয়।

এখানে ব্যাপারটি হচ্ছে এ রকম। যেহেতু শুধু ২টি সংকেত আমার ধরতে হবে, সেহেতু আমি ধরব বিদ্যুৎ প্রবাহ হচ্ছে নাকি হচ্ছে না। যদি বিদ্যুৎ প্রবাহ হয়, তবে সেটাকে যদি ১ ধরি এ জন্য ৫ ভোল্ট লাগে, তবে বিদ্যুৎ প্রবাহ না করলে সে ক্ষেত্রে সেটিকে আমরা ০ ধরব। এখন যেকোনো ভোল্টেজকে ৫ ভোল্ট করে পেতে হলে আমাদের একটি বাক বা বুস্ট কনভার্টার ব্যবহার করতে হবে।

পরীক্ষার জন্য আমরা একটি সেন্সর নেব। বাজারে অনেকগুলো সেন্সর পাওয়া যায়। তার মধ্যে আমরা যেকোনো একটি বেছে নিই। চিত্র ৩.৮-এ সার্কিট ডায়াগ্রামটি দেওয়া হলো।



চিত্র ৩.৮ : সিম্পল সার্কিট ডায়াগ্রাম

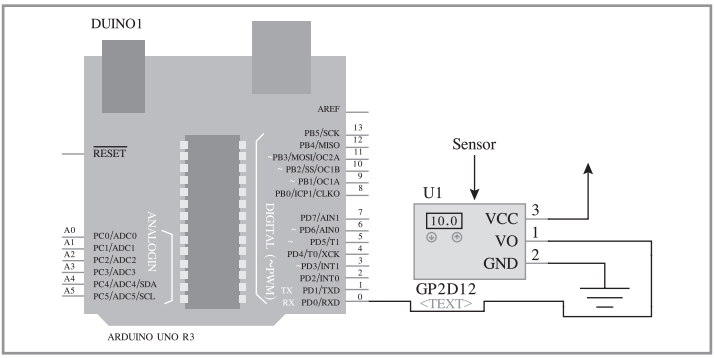
এখানে আমরা সার্কিট ডায়াগ্রাম থেকে সহজেই দেখতে পাচ্ছি ব্যাপারটি। সেন্সর থেকে শুধু পাঠ আসলেই ৫ ভোল্ট কনভার্টার যেকোনো মানের ভোল্টেজকে ৫ ভোল্টে আউটপুট হিসেবে বের করবে। আর যদি কোনো ভোল্টেজ আউটপুট না আসে, তবে ০ ভোল্ট হিসেবেই বের হবে।

কাজের জন্য এই সার্কিট ডায়াগ্রাম খুব সুবিধার না। এ জন্য ফিল্টার ব্যবহার করতে হবে। নইলে যেকোনো প্রকার নয়েস তৈরি হবে; যার কারণে সঠিক পাঠ পাওয়া যাবে না। এখানে আমরা যদি ফিল্টার হিসেবে একটি ক্যাপাসিটর ব্যবহার করি, তবে সে ক্ষেত্রে আউটপুট অনেক ভালো পাওয়া যাবে। বাজারে বর্তমানে যে ভোল্টেজ বুস্টার পাওয়া যায়, সেগুলোতে ফিল্টার থাকে। অথবা সবচেয়ে ভালো হয় যদি আমরা একটি রিলে ব্যবহার করি।

নিচে কোডটি দেওয়া হলো।

```
int sensor_pin = 0;
void setup ()
{
  pinMode (sensor_pin, INPUT);
  Serial.begin (9600);
}
void loop ()
{
  int val=digitalRead (sensor_pin);
  Serial.println (val);
  delay (100);
}
```

এই কোডটি চালিয়ে দেখতে হবে। এখানে চিত্র ৩.৯-এ সার্কিট ডায়াগ্রামটির মডেল দেওয়া হলো।



চিত্র ৩.৯ : প্রধান সার্কিট ডায়াগ্রাম

এখানে সেন্সরটি একটি ডিজিটাল সেন্সর। বর্তমানে বাজারে যেসব ডিজিটাল সেন্সর পাওয়া যায় তার মধ্যে সাধারণত লেজার সেন্সর বেশি উল্লেখযোগ্য। এটি এমন একটি সেন্সর, যেটিতে লেজারের মাঝপথে কোনো বাধা যদি পড়ে, তবে সেটি আউটপুট ১ দেখায় অথবা ০ দেখায়। ক্ষেত্রবিশেষে পরিবর্তন হতে পারে। যদি কোনো বাধা না থাকা অবস্থায় আউটপুট ০ দেখায়, তবে লেজার রশ্মির মাঝপথে কোনো বাধা পড়লে ১ দেখাবে।

কোডটি আপলোড করে রান করলে সিরিয়াল পোর্টে শুধু ১ অথবা ০ দেখা যাবে।

ডিজিটাল ইনপুট কেবল তখনই ব্যবহার করা যায় যদি শুধু ২টি লজিক যে ক্ষেত্রে রয়েছে। বাস্তবে ২টি লজিক দিয়ে চলে এ ধরনের মেশিন খুব কম রয়েছে। সে জন্যই অ্যানালগ ইনপুট সিস্টেম প্রায় সব ক্ষেত্রে ব্যবহার করা হয়। অ্যানালগ সিস্টেমে একই সাথে অনেকগুলো মানের ভোল্টেজের জন্য ভিন্ন ভিন্ন মানের ইনপুট রয়েছে। আরডুইনোতে এই ইনপুট দিলে বিভিন্ন মানের ভোল্টেজের জন্য ভিন্ন ভিন্ন আউটপুট বের হবে প্রয়োজনীয় লজিক অনুযায়ী।

কয়েকটি উদাহরণ দেখলেই ব্যাপারটি পরিষ্কার হবে।

আমরা জানি যে সেন্সর হচ্ছে একটি পরিবর্তনশীল চলক। মানে যার রোধকের মান পরিবর্তন হয় পরিবেশের বিভিন্ন উপাদান পরিবর্তনের

সাথে সাথে। এখন এখানে রোধকের মান যেহেতু পরিবর্তিত হচ্ছে, সেহেতু আউটপুট ভোল্টেজও পরিবর্তন হবে। এর আগে এ-সম্পর্কে বিস্তারিত আলোচনা করা হয়েছে।

এখন আমরা একটি সার্কিট ডায়াগ্রাম দেখব। এ অনুযায়ী সার্কিট তৈরি করতে হবে। সেন্সর সম্পর্কে ভালো ধারণা পাওয়ার জন্য প্রথমে আমরা একটি পটেনশিওমিটার ব্যবহার করব।

এ জন্য যা যা লাগবে—

১. একটি রেসিস্টর (১ কিলোওহম হলে ভালো হয়)

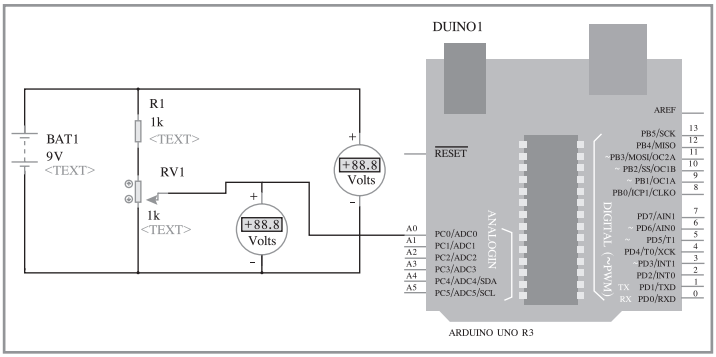
২. একটি পরিবর্তনশীল রোধক বা পটেনশিওমিটার (০ থেকে ১০ অথবা ১০০ কিলোওহম)

৩. একটি ব্যাটারি (৯ ভোল্ট হলে ভালো হয়)

৪. কিছু জাম্পার তার

প্রথমে আমরা আউটপুটের সিস্টেম মনিটর করব। এরপরে লজিক অনুযায়ী প্রয়োজনীয় কোড তৈরি করব।

প্রথমে সার্কিট ডায়াগ্রামটি দেখা যাক—



চিত্র ৩.১০ : সার্কিট ডায়াগ্রাম

সার্কিট ডায়াগ্রামে আমরা দেখছি একটি পরিবর্তনশীল রোধ এবং একটি সাধারণ রোধ সিরিজে সাজানো আছে। ২টির মধ্যে থেকে একটি সংযোগ নিয়ে আরডুইনোতে অ্যানালগ পিনে সংযোগ দিতে হবে। এখানে আমরা অ্যানালগ পিন ০ তে সংযোগ দিলাম। আরেকটি ব্যাপার মনে রাখতে

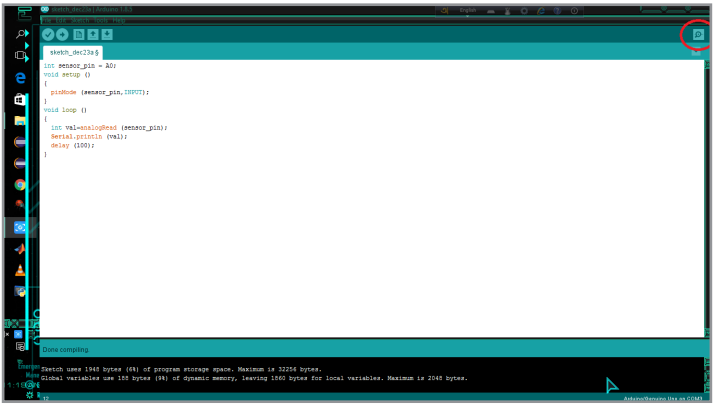
হবে যে ব্যাটারির ঋণাত্মক প্রান্ত আরডুইনোর গ্রাউন্ডে সংযোগ দিতে হবে। এটা না দিলে সার্কিট চালু হবে না।

এখন কোডটি নিম্নরূপ—

```
int sensor_pin = A0;
void setup ()
{
  pinMode (sensor_pin, INPUT);
  Serial.begin (9600);
}
void loop ()
{
  int val=analogRead (sensor_pin);
  Serial.println (val);
  delay (100);
}
```

সার্কিট পূর্ণ করে এরপরে কোডটি আপলোড দিতে হবে।

এখন কোডটি আপলোড দেওয়া শেষে এরপরে আসল কাজ। সেন্সর মনিটর করতে হবে। পটেনশিওমিটারের নব ঘুরিয়ে দেখতে হবে।



চিত্র ৩.১১ : প্রোগ্রাম রান করা হচ্ছে

এখানে ছবিতে বৃত্তাকার মার্ক করা অংশটিতে ক্লিক করতে হবে। ক্লিক করলেই আরেকটা স্ক্রিন আসবে যেখানে সেন্সরের সামগ্রিক বিভিন্ন দিক পরিবর্তনের সাথে সাথে পরিবর্তনের অংশটি বোঝা যাবে যেটি পর্দায় দেখা যাবে। এখানে বিভিন্ন মানের পাঠ। ০ থেকে ১০২৩ পর্যন্ত এর

মাঝে পাঠ পাওয়া যাবে। নব ঘোরালে এখানে ভোল্টেজ পরিবর্তন হবে। ভোল্টেজ পরিবর্তনের সাথে সাথে সিরিয়াল মনিটরে সংখ্যাগুলো পরিবর্তন হতে থাকবে। লক্ষ রাখতে হবে void setup () ফাংশনে Serial. begin (9600); এই লাইনটা লিখতে হবে। নাইলে আমরা সিরিয়াল মনিটরে প্রয়োজনীয় সংখ্যা আউটপুট পাব না।

এবার আমরা একটি ছোট প্রজেক্ট করব। এখানে আমরা যে কাজটি করব, সেটি হচ্ছে LDR সেন্সরের বেসিকে একটি প্রজেক্ট। LDR সেন্সর এমন একটি সেন্সর যেটি আলোতে উপযুক্ত সাড়া দেয়। এখানে আলোর উপস্থিতিতে এর রোধের মান পরিবর্তন হতে থাকে।

তো আমরা লজিক বেসিসে কাজটি করব। আমাদের প্রজেক্টটি হবে LDR সেন্সরের ওপরে যখন আলো পড়বে, তখন একটি LED বাল্ব নিভে যাবে। আর যখন LDR সেন্সরের ওপরে কোনো আলো পড়বে না, তখন LED বাল্ব জ্বলবে। অর্থাৎ আউটপুটটি হচ্ছে একটি ঘর অন্ধকার থাকলে সেখানে থাকা LED বাল্বটি জ্বলে উঠবে আর যদি ঘর আলোকিত থাকে, তখন LED বাল্বটি নিভে যাবে। এটি প্রথম প্রজেক্ট যেটিতে আমরা লজিক ব্যবহার করব।

যা যা লাগবে—

১. একটি রেসিস্টর (১ কিলোওহম হলে ভালো হয়)

২. একটি LDR সেন্সর

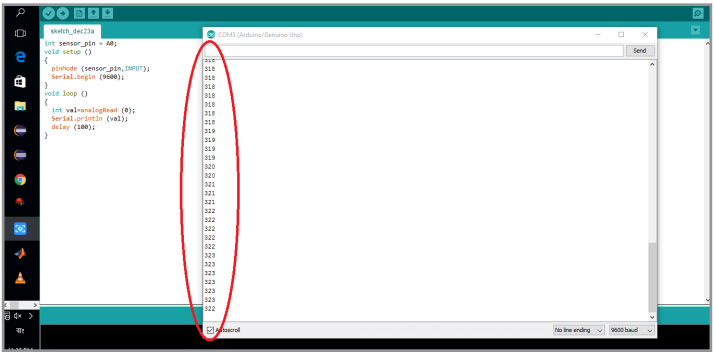
৩. একটি ব্যাটারি (৯ ভোল্ট হলে ভালো হয়)

৪. কিছু জাম্পার তার

৫. LED বাল্ব

এ প্রজেক্টটির মাধ্যমে আমরা আরও জানব একটি সেন্সর কীভাবে ক্যালিব্রেট করতে হয়। এখানে আমরা সিরিয়াল মনিটরে অনেকগুলো মান পাব। মানগুলো হবে ০ থেকে ১০২৩-এর মধ্যে। এই মানগুলো আসলে লজিকের জন্যই ব্যবহার করা হবে। ধরা যাক সেন্সরটি অন্ধকার পরিবেশে থাকার জন্য যদি মানগুলো ০ থেকে ৫০০-এর মধ্যে দেখায় তবে আলোর মধ্যে থাকার জন্য এটি ৫০০ থেকে ১০০০ কিংবা ১০২৩ এ রকম থাকবে। মধ্যবর্তী মান কত থেকে কত-এর মধ্যে হবে সেটি আসলে নির্ভর করে পরিবেশের ওপরে এবং রোধের মান কত তার ওপরেও নির্ভর করে।

চিত্র ৩.১২-তে মানগুলোর ব্যাপারটি দেখা যাচ্ছে। মার্ক করা অংশটিতে মানগুলোকে দেখা যাচ্ছে।



চিত্র ৩.১২ : মান দেখা যাচ্ছে

মানগুলো থেকেই আমরা সেন্সরের জন্য প্রয়োজনীয় পাঠ পাব।

কাজের সুবিধার জন্য আমরা একটি পদ্ধতি প্রয়োগ করব। পরিবেশ এবং কম্পিউটারের ওপরে নির্ভর করে এখানে সিরিয়াল মনিটরে কী রকম পাঠ আসবে। এসব বিবেচনা করে আমরা একটি নির্দিষ্ট পরিবেশের কথা চিন্তা করে মানগুলোর সর্বোচ্চ ও সর্বনিম্ন মানগুলোকে x ও y হিসেবে প্রকাশ করব।

ধরা যাক অন্ধকার পরিবেশে অর্থাৎ LDR সেন্সরে যখন কোনো আলো পড়ে না, তখন সিরিয়াল মনিটরে যদি পাঠগুলোর মান সর্বোচ্চ x ও সর্বনিম্ন y অর্থাৎ ধরি অন্ধকারে সিরিয়াল মনিটরে পাঠ আসছে ২০০ থেকে ৩৫০ এ রকম একটি রেঞ্জের মধ্যে সবগুলো পাঠ আসছে। তবে আমরা নিচের দিক থেকে ১০ অথবা ২০ কমিয়ে ১৮০ আর ওপরের দিকে ৩৫০ থেকে ১০ অথবা ২০ বাড়িয়ে একটি সাধারণ মান ৩৭০ এ রকম একটি রেঞ্জ নিতে হবে। অর্থাৎ অন্ধকারে আমার রেঞ্জ ১৮০ থেকে ৩৫০। ১৮০ থেকে ৩৫০ এখানে অন্ধকারের রেঞ্জের ৩৫০ এবং ১৮০ কে যথাক্রমে x ও y দ্বারা প্রকাশ করব।

কোডটি নিম্নরূপ—

```
int sensor_pin = A0;
int led_pin = 13;
int flag;
```

```
void setup ()
{
  pinMode (sensor_pin, INPUT);
  pinMode (led_pin, OUTPUT);
  Serial.begin (9600);
}
void loop ()
{
  int val=analogRead (sensor_pin);
  Serial.println (val);
  delay (100);
  if (val>180 && val<350) // dark room, no light logic
  {
    digitalWrite (led_pin, HIGH);
    flag=1;
  }
  else if (val>540 && val<700) // environmental light, enlightened room logic
  {
    digitalWrite (led_pin, LOW);
    flag=0;
  }
  else
  {
    flag=1;
  }
}
```

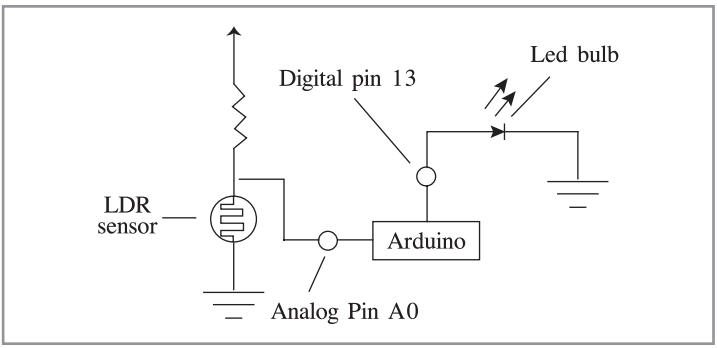
কোডটির সিরিয়াল প্রিন্টের মানগুলোর সর্বোচ্চ ও সর্বনিম্ন মানগুলো যদি x ও y হিসেবে প্রকাশ করি, তবে কোডটি নিম্নরূপ হবে—

```
int sensor_pin = A0;
int led_pin = 13;
int flag;
void setup ()
{
  pinMode (sensor_pin, INPUT);
  pinMode (led_pin, OUTPUT);
  Serial.begin (9600);
}
void loop ()
{
```

```
  int val=analogRead (sensor_pin);
  Serial.println (val);
  delay (100);
  // y is the lowest value found in the serial monitor for dark room logic
  // x is the highest value found in the serial monitor for dark room logic
  if (val>y && val<x) // dark room no light logic
  {
    digitalWrite (led_pin, HIGH);
    flag=1;
  }
  // y1 is the lowest value found in the serial monitor for enlightened room logic
  // x1 is the highest value found in the serial monitor for enlightened room logic
  else if (val>y1 && val<x1) // environmental light, non enlightened room logic
  {
    digitalWrite (led_pin, LOW);
    flag=0;
  }
  else
  {
    flag=1;
  }
}
```

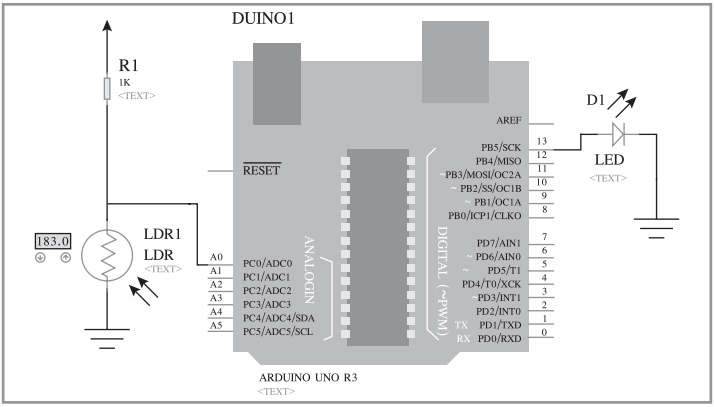
এখানে আমরা x ও y ব্যবহার করেছি শুধু বোঝাতে। কোডে কিন্তু কোনোভাবেই ব্যবহার করা যাবে না। এটি শুধু বোঝানোর পরিপ্রেক্ষিতে।

এখন আমরা চিত্র ৩.১৩-এর সার্কিট ডায়াগ্রামটি দেখব।



চিত্র ৩.১৩ : সিম্পল সার্কিট ডায়াগ্রাম

এখন ব্যবহারিক সার্কিটটি কী রকম হবে, সেটি দেখব আমরা।



চিত্র ৩.১৪ : সিমুলেটেড সার্কিট ডায়াগ্রাম

সেন্সরটি উপযুক্তভাবে ক্যালিব্রেট করলে এখানে আমরা যে আউটপুটটি পাব, সেটি হচ্ছে একটি অন্ধকার ঘরে আলো না থাকলে LED বাল্বটি জ্বলবে, আলো থাকলে LED বাল্বটি জ্বলবে না। এ প্রজেক্টটির ওপরে ভিত্তি করে আমরা পরবর্তী সময়ে আরেকটি প্রজেক্ট তৈরি করব। সেটি হচ্ছে লাইট ফলোয়ার রোবট। তার আগে আমাদের মোটর সম্পর্কে বিস্তারিত আলোচনা হবে।

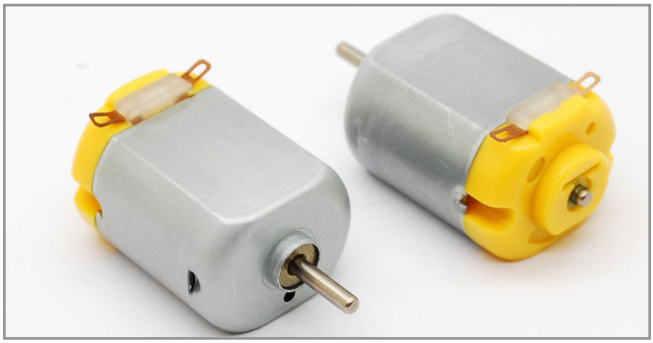
চতুর্থ অধ্যায়

মোটর

মোটর সম্পর্কে আলোচনা মোট ৩ ভাগে বিভক্ত। এই আলোচনায় আমরা মোটর কী রকম হতে পারে এবং কীভাবে প্রোগ্রামিং করে চালানো যায়, সে সম্পর্কে বলব। আর এ আলোচনায় ডিসি মোটর ঘড়ির কাঁটার দিকে ঘড়ির কাঁটার বিপরীতে কীভাবে প্রোগ্রামিং করে ঘোরাতে হয় এবং কীভাবে স্পিড বাড়াতে কিংবা কমাতে হয় প্রয়োজনীয় লজিক অনুযায়ী, সেটি দেখব। সার্ভো মোটরের অংশে আলোচনা করব কীভাবে কত কোণ কোন সময়ে থাকবে, সেটি নির্ধারণ করা হবে প্রোগ্রামিংয়ের মাধ্যমে। সবশেষে আলোচনা করা হবে স্টেপার মোটর সম্পর্কে।

ডিসি মোটর

ডিসি মোটর অনেক রকমের হয়। বর্তমানে বাজারে কাজের ওপরে ভিত্তি করে ডিসি মোটর তৈরি করা হয়। গোলাকার, চতুর্ভুজাকার, সিলিন্ডার বিভিন্ন আকৃতির ডিসি মোটর রয়েছে।



চিত্র ৪.১ : ডিসি মোটর

এখানে আমরা ২টি ডিসি গিয়ার মোটর দেখছি। ডিসি মোটরের সাথে পরিচিত হওয়ার আগে সাধারণ ডিসি মোটর এবং গিয়ার ডিসি মোটরের সাথে পরিচিত হব।

সাধারণ মোটর : সাধারণ মোটরের স্পিড অনেক বেশি গিয়ার মোটরের তুলনায়। সাধারণ মোটরের সাথে কোনো গিয়ারবক্স লাগানো থাকে না। সে জন্য স্পিড অনেক বেশি কিন্তু লোড নেওয়ার ক্ষমতা অনেক কম। ওপরের ছবিতে প্রথম মোটরটি একটি সাধারণ মোটর। গিয়ারবক্স ছাড়া সাধারণ মোটর সাধারণত সিপিইউ কুলার ফ্যান হিসেবে বেশি ব্যবহার হয়।

গিয়ার মোটর : গিয়ার মোটরে সাধারণ মোটরের সাথে একটি গিয়ারবক্স লাগানো থাকে। এটি ডিসি গিয়ার মোটর। সাধারণ মোটর খুব বেশি লোড নিতে পারে না। ব্যবহারিক কাজে এ জন্য একটি গিয়ারবক্স ব্যবহার করা হয়। এটি ডিসি গিয়ার মোটর। এর বিশেষত্ব অনেক ভার বহনে সক্ষম। চাকাওয়ালা রোবটগুলোতে চাকার সাথেই ব্যবহার হয়। সাধারণ মোটরের যে শ্যাফট থাকে সেটিতে একটি গিয়ারবক্স লাগানো হয়। বাজারে যে বিভিন্ন গিয়ার মোটর পাওয়া যায় তা কেনার আগে কয়েকটি জিনিস মনে রাখতে হবে।

১. মোটর সাধারণত R.P.M (Revolution per minute) হিসেবে বিক্রি হয়। একই মোটর কিন্তু ২টি ভিন্ন গিয়ারবক্সের জন্য যে মোটরের R.P.M কম সেটি বেশি ভার বহনে সক্ষম। যে মোটরের R.P.M বেশি সেটি কম ভার বহনে সক্ষম। অর্থাৎ—

$R.P.M \propto 1/Load$

এটি আসলে নির্ভর করছে গিয়ারবক্সের ভেতরে গিয়ারগুলোর দাঁতের অনুপাত অনুযায়ী।

২. কত কেজি নিতে পারবে সে হিসেবে বিক্রি হয়। ৫ কেজি ওজন বহনে সক্ষম একটি মোটর অবশ্যই ৩ কেজির চেয়ে বেশি ওজন বহনে সক্ষম হবে।

৩. একটি মোটর সর্বোচ্চ কত ভোল্টেজ ইনপুট নিতে পারবে, সেটি মনে রাখতে হবে। সর্বোচ্চ ৬ ভোল্ট ইনপুট নেওয়া মোটরকে যদি আমি ১২ ভোল্টের একটি ব্যাটারির সাথে সংযোগ দিই, তবে মোটরটি নষ্ট হওয়ার আশঙ্কা বেশি।

৪. গিয়ার মোটরকে উল্টা দিক দিয়ে ঘোরানো যাবে না। এতে ভেঙে যাওয়ার আশঙ্কা আছে।

মোটর চালানো

মোটর চালাতে আমরা সবাই পারি। একটি মোটরের ২টি পাতকে যদি ব্যাটারির সাথে সংযোগ দিই, তবে মোটরটি ঘোরা শুরু করে। তবে এখানে কয়েকটি বিষয় মনে রাখতে হবে। কাজ শুরুর আগে কয়েকটি বিষয় খেয়াল করতে হবে। আমরা এখানে কয়েকটি পরীক্ষা করব প্রথমে।

১. মোটরের ওপরে ভোল্টেজের প্রয়োগ করে দেখব। ভোল্টেজ বাড়িয়ে-কমিয়ে।

২. রেসিস্টরের রেসিস্টেন্স বাড়িয়ে কমিয়ে এরপর আমরা এর মাধ্যমে মোটরের গতি পরিবর্তন করব।

৩. ট্রানজিস্টরের মাধ্যমে একটি মোটরের গতি কমানো ও বাড়ানো এটিও দেখানো হবে।

ভোল্টেজ বাড়িয়ে-কমিয়ে

একটি সাধারণ ডিসি মোটর নেই। সাথে ৩/৪টি ব্যাটারি নিতে হবে। প্রতিটি ব্যাটারি ১.৫ ভোল্টের পেনসিল ব্যাটারি নিলে ভালো হয়। এখানে একটি ব্যাটারি কেসে নিয়ে আমরা পরীক্ষাটি করব। প্রথমে ১টি, তারপরে ২টি, তারপরে ৩টি এবং সবশেষে ৪টি ব্যাটারি দিয়ে প্রতিবার মোটরটি চালাব।

ফল : প্রতিবার মোটরের সাথে ১টি করে ব্যাটারি যুক্ত করার ফলে এখানে গতি বাড়ে। আপাতদৃষ্টিতে এটি খুব সামান্য ঘটনা। কিন্তু একটি ব্যাপার মনে রাখতে হবে। ভোল্টেজ বাড়ালে মোটরের গতিও বাড়ে। মোটরের গতি ভোল্টেজের ওপরে নির্ভর করে।

রেসিস্টেন্স বাড়িয়ে-কমিয়ে

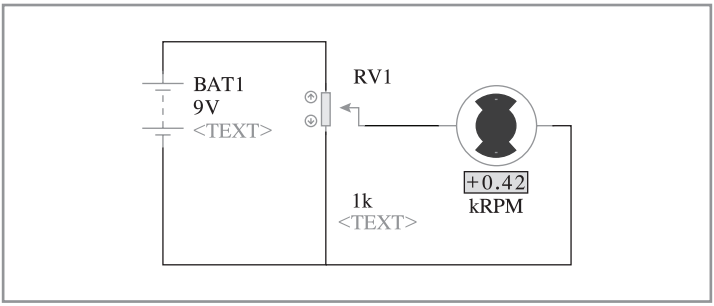
রেসিস্টেন্স বাড়ানো-কমানোর ওপরে এর সাথে আউটপুট ভোল্টেজ বাড়ে ও কমে। যে কারণে মোটরের ঘূর্ণন বেগ বাড়ে ও কমে।

এ পরীক্ষাটি করার জন্য যা যা লাগবে—

১. ব্যাটারি

২. পটেনশিওমিটার

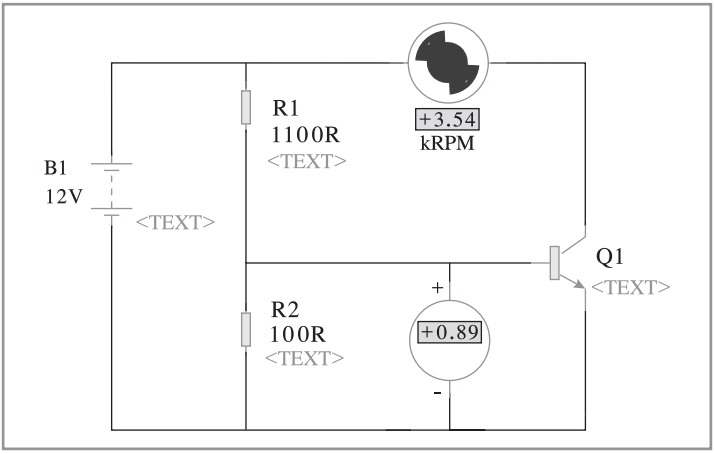
৩. একটি সাধারণ মোটর



চিত্র ৪.২ : একটি মোটরের গতি পরিবর্তনের সার্কিট ডায়াগ্রাম

এখানে ওপরে যে সার্কিটটি দেখা যাচ্ছে সেই অনুযায়ী সার্কিট আমাদের তৈরি করতে হবে। এখানে পটেনশিওমিটার ঘোরালে মোটরের গতি সে অনুযায়ী পরিবর্তন হবে। রোধের পরিমাণ বেশি হলে আউটপুট বিভব অনেক কমে যাবে। ফলে মোটরের গতিও অনেক কমে যাবে।

ট্রানজিস্টরের মাধ্যমে একটি মোটরের গতি কমানো ও বাড়ানো



চিত্র ৪.৩ : ট্রানজিস্টর দিয়ে মোটরের গতি বাড়ানো ও কমানো

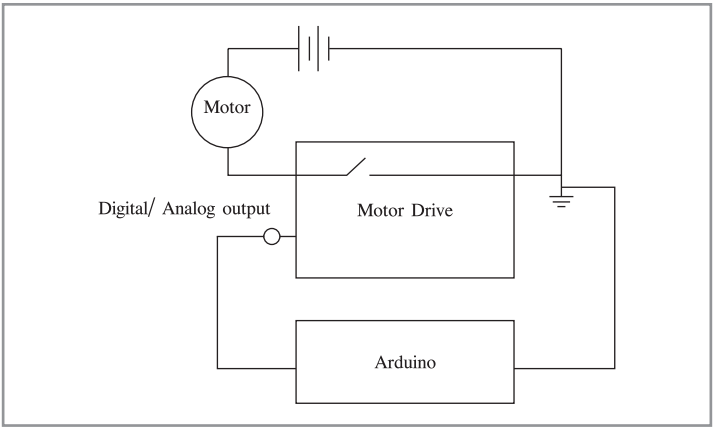
এখানে R2 অংশের রোধ বাড়ানো-কমানোর সাথে সাথে এমিটার ও কালেকটরের মধ্যে প্রবাহ বাড়ে এবং কমতে থাকে অর্থাৎ বিভব বাড়ে ও কমে। যার জন্য মোটরের স্পিড বাড়ে ও কমে। R2 অংশের রোধ বাড়ানো-কমানোর বদলে এখানে যদি একটি কৃত্রিমভাবে আরডুইনো দিয়ে আমরা প্রবাহ বাড়াই কিংবা কমাই, তবে সেভাবেও মাত্র ৫ ভোল্ট আরডুইনোর আউটপুট দিয়ে, অনেক বেশি মানের ভোল্টেজ দিয়েও মোটর চালানো সম্ভব। এখানে আসলে ট্রানজিস্টরটি সুইচ হিসেবে কাজ করে।

মোটরের ঘূর্ণন গতির দিক পরিবর্তন

এ পরীক্ষাটি করা খুব সহজ। একটি মোটরের সাথে ব্যাটারির সংযোগ দিই। ব্যাটারির সংযোগ উল্টে দিলে মোটরের ঘোরার দিকও পরিবর্তন হয়ে যাবে।

মোটর ড্রাইভার

ওপরে আমরা যে কাজগুলো দেখলাম মোটরের এই সবগুলো কাজ ১টি I.C দিয়ে করা সম্ভব প্রোগ্রামিংয়ের মাধ্যমে। এই I.C হচ্ছে মোটর ড্রাইভার। মোটর ড্রাইভার এমন একটি আইসি, যেটি প্রোগ্রামিংয়ের লজিকের মাধ্যমে অনেক কম ভোল্টেজের মাধ্যমে অনেক বেশি ভোল্টেজের মোটরকে চালানো সম্ভব। আরডুইনোর সিগন্যাল সাধারণত ৫ ভোল্টের হয়। এই মাত্র ৫ ভোল্ট দিয়ে মোটর চালানো সম্ভব না খুব বেশি জোরে। ব্যবহারিক কাজের জন্য যে মোটর ব্যবহার করা হয়, সেটি সাধারণত ১২ থেকে ২০ ভোল্ট সর্বোচ্চ ভোল্টেজ ধারণ করতে পারে। এত বড় ভোল্টেজ সাপ্লাইয়ের জন্যই মূলত মোটর ড্রাইভার দরকার। চিত্র ৪.৪-এর ব্লক ডায়াগ্রামটি দেখলেই ব্যাপারটি পরিষ্কার হবে।



চিত্র ৪.৪ : মোটর ড্রাইভারের কার্যবিধি

ব্লক ডায়াগ্রাম থেকে একটি বিষয় পরিষ্কার। কয়েকটি ট্রানজিস্টরের সমষ্টি বলে মোটর ড্রাইভার আসলে একটি সুইচ হিসেবে কাজ করে।

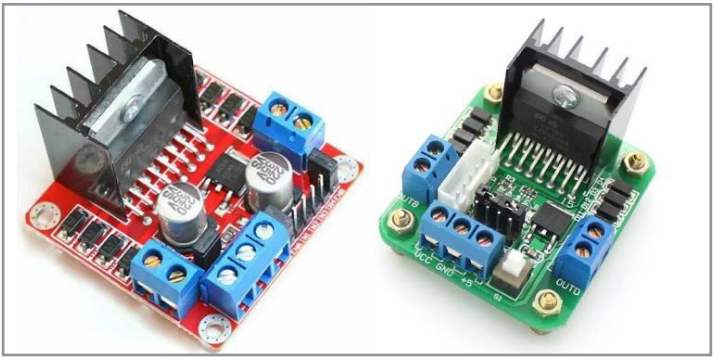
এখানে আরডুইনো থেকে আসা ডিজিটাল বা অ্যানালগ আউটপুট সিগন্যাল থেকে প্রয়োজনীয় ভোল্টেজ অনুযায়ী মোটর ও ব্যাটারির মধ্যকার ভোল্টেজের পরিমাণ নির্ধারণ হয়। এটি আসলে কয়েকটি ট্রানজিস্টরের সমষ্টি, যে কারণে এটি করা যাচ্ছে। আগেই এ-সম্পর্কে বলেছি।

এখন বর্তমানে যে ২টি মোটর ড্রাইভার পাওয়া যায় সেই ২টি হচ্ছে—

১. L298N

২. L293D

এই ২ মডেলের ড্রাইভার আইসি। তবে L298N আইসিসহ একটি প্যাকেজ সার্কিট বোর্ড পাওয়া যায়। এটি সবচেয়ে বেশি ব্যবহার হওয়া আইসি।



চিত্র ৪.৫ : L298N মোটর

এখানে ২টি L298N মোটর ড্রাইভারকে দেখা যাচ্ছে। সাধারণত লাল, সবুজ ও নীল এই ৩ রঙের পিসিবি প্যাকেজ আকারে পাওয়া যায়। তবে অনেকে শুধু আইসি ব্যবহার করেও মোটর চালানোর কাজটা করে থাকে। সে জন্য প্রয়োজনীয় সেটআপ করে নিতে হবে।

এই পিসিবি প্যাকেজটি হাতে নিয়ে দেখতে হবে ভালোমতো। অনেকগুলো পোর্ট রয়েছে। এখানে নিচে প্রতিটি পোর্টের কাজ সম্পর্কে দেওয়া হলো। এখানে পোর্টের নাম অথবা / দিয়ে সাথে থাকা নাম যেকোনো নামেই থাকতে পারে। কিন্তু ২টি একই পোর্ট।

পোর্ট	কাজ	সিগন্যাল
ena / en1 (enable a / 1)	১ নম্বর মোটর চালু করা। এখানে কোনো সিগন্যাল না থাকলে মোটর চালু হবে না	যেকোনো ভোল্টের ব্যাটারির জন্য সর্বোচ্চ যতটুকু গতি পাওয়া যায় তার জন্য ডিজিটাল পোর্টে ১ অথবা অ্যানালগ পোর্টে ২৫৫ ইনপুট হিসেবে দিতে হবে। অ্যানালগ পোর্টে ০ থেকে ২৫৫-এর মধ্যে যেকোনো ইনপুটের জন্য মোটরের ঘূর্ণন গতি নির্ধারণ হয়।
enb / en2 (enable b / 2)	২ নম্বর মোটর চালু করা। এখানে কোনো সিগন্যাল না থাকলে মোটর চালু হবে না	যেকোনো ভোল্টের ব্যাটারির জন্য সর্বোচ্চ যতটুকু গতি পাওয়া যায় তার জন্য ডিজিটাল পোর্টে ১ ও অ্যানালগ পোর্টে ২৫৫ ইনপুট হিসেবে দিতে হবে। অ্যানালগ পোর্টে ০ থেকে ২৫৫-এর মধ্যে যেকোনো ইনপুটের জন্য মোটরের ঘূর্ণন গতি নির্ধারণ হয়।

ina / in1 (input a / 1)	১ নম্বর মোটরের ঘোরার দিক নির্ধারণ করে। মোটরটি ঘড়ির কাঁটার দিকে নাকি বিপরীতে ঘুরবে।	এখানে ডিজিটাল ইনপুট হিসেবে ১ ইনপুট দিতে হবে। যদি পোর্টটির সিগন্যাল হাই বা ১ থাকে তবে inb / in2 লো বা ০ থাকতে হবে।
inb / in2 (input b / 2)	১ নম্বর মোটরের ঘোরার দিক নির্ধারণ করে। মোটরটি ঘড়ির কাঁটার দিকে নাকি বিপরীতে ঘুরবে।	এখানে ডিজিটাল ইনপুট হিসেবে ১ ইনপুট দিতে হবে। যদি পোর্টটির সিগন্যাল হাই বা ১ থাকে তবে ina / in1 লো বা ০ থাকতে হবে।
inc / in3 (input c / 3)	২ নম্বর মোটরের ঘোরার দিক নির্ধারণ করে। মোটরটি ঘড়ির কাঁটার দিকে নাকি বিপরীতে ঘুরবে।	এখানে ডিজিটাল ইনপুট হিসেবে ১ ইনপুট দিতে হবে। যদি পোর্টটির সিগন্যাল হাই বা ১ থাকে তবে ind / in4 লো বা ০ থাকতে হবে।
ind / in4 (input d / 4)	২ নম্বর মোটরের ঘোরার দিক নির্ধারণ করে। মোটরটি ঘড়ির কাঁটার দিকে নাকি বিপরীতে ঘুরবে।	এখানে ডিজিটাল ইনপুট হিসেবে ১ ইনপুট দিতে হবে। যদি পোর্টটির সিগন্যাল হাই বা ১ থাকে তবে inc / in3 লো বা ০ থাকতে হবে।

ছক ৪.১ : পিসিবি প্যাকেজের পোর্টের কাজ

আউটপুট ও পাওয়ার পোর্ট

পোর্ট	কাজ
out 1 (output 1)	মোটরের সাথে সংযোগ দিতে হয়। এখানে সংযোগস্থল ২টি। এই ২টিকে মোটরের সাথে সংযোগ দিতে হবে। এটি ড্রাইভারের ১ নম্বর মোটর।
out 2 (output 2)	মোটরের সাথে সংযোগ দিতে হয়। এখানে সংযোগস্থল ২টি। এই ২টিকে মোটরের সাথে সংযোগ দিতে হবে। এটি ড্রাইভারের ২ নম্বর মোটর।
VCC	ব্যাটারির ধনাত্মক প্রান্ত সংযোগ দিতে হয়।
GND	ব্যাটারির ঋণাত্মক প্রান্ত এবং আরডুইনোর GND সংযোগ দিতে হয়।

ছক ৪.২ : আউটপুট ও পাওয়ার পোর্টের কাজ

একটি কথা মনে রাখতে হবে যখন ina / in1-তে ইনপুট সিগন্যাল হাই থাকবে, তখন inb / in2-তে ইনপুট সিগন্যাল লো থাকবে এবং একইভাবে inc / in3-তে ইনপুট সিগন্যাল যখন হাই থাকবে, তখন ind / in4-তে ইনপুট সিগন্যাল লো থাকবে। ধরা যাক ina / in1-তে ইনপুট সিগন্যাল ১ তখন inb / in2-তে ইনপুট সিগন্যাল লো এই লজিকে মোটর যদি ঘড়ির কাঁটার দিকে ঘোরে, তবে ina / in1-তে ইনপুট সিগন্যাল ০ এবং inb / in2-তে ইনপুট সিগন্যাল ১ হলে মোটর ঘড়ির কাঁটার বিপরীতে ঘুরবে।

এখন আমরা একটি প্রজেক্ট করব। প্রজেক্টটি এমন আরডুইনোতে আমরা মোটরের ena / en1 এবং enb / en2 পোর্টে বিভিন্ন মানের ইনপুটের জন্য মোটরটির গতির দিক পর্যবেক্ষণ করব। মনে রাখতে হবে ena / en1 এবং enb / en2 সব সময় অ্যানালগ অথবা P.W.M ডিজিটাল পোর্টে সংযোগ দিতে হবে। আরডুইনো বোর্ডে যেসব ডিজিটাল পোর্টে ঢেউয়ের মতো একটি চিহ্ন আছে অনেকটা নম্বরের পাশে—একটি চিহ্ন ওগুলোই P.W.M পোর্ট।

প্রজেক্ট ১ : মোটরের গতি পর্যবেক্ষণ

এর জন্য আমাদের প্রয়োজনীয় উপকরণ—

১. L298N মোটর ড্রাইভার

২. জাম্পার তার

৩. আরডুইনো বোর্ড

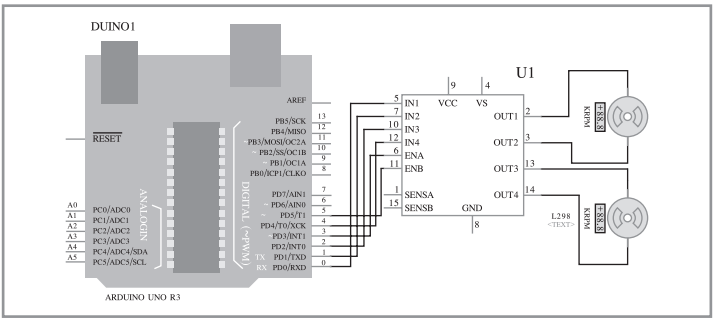
৪. ব্যাটারি

তারের সংযোগগুলো নিচে দেওয়া হলো। কীভাবে দিতে হবে।

পোর্ট	সংযোগ (আরডুইনোর পিন নম্বর)
ena	3
enb	5
ina / in1	0
inb / in2	1
inc / in3	2
ind / in4	4

ছক ৪.৩ : পোর্টের সাথে তারের সংযোগ

এখানে সংযোগগুলো Arduino Uno-এর জন্য দেখানো হলো। তবে মডেল অনুযায়ী সংযোগ পরিবর্তন হতে পারে। তবে অবশ্যই মনে রাখতে হবে ena, enb অবশ্যই P.W.M পোর্টে সংযোগ দিতে হবে। output 1 এবং output 2-তে ২টি মোটর সংযোগ দিতে হবে। VCC ও GND-এর সাথে ব্যাটারির ধনাত্মক ও ঋণাত্মক প্রান্ত সংযোগ দিতে হবে। এখানের GND এবং আরডুইনোর GND এই ২টি সংযোগ দিতে হবে একত্রে। মোটর ড্রাইভারের 5v থেকে আরডুইনোর Vin-এ আরেকটি সংযোগ যাবে।



চিত্র ৪.৬ : সংযোগগুলো দেখানো হয়েছে

এখানে আমরা যে প্রজেক্টগুলো করব, সেগুলো কয়েকটি ভাগে বিভক্ত। প্রথম ভাগে আমরা শুধু একটি মোটরের বিভিন্ন সিগন্যাল ইনপুটে মোটর চালাব ভালোভাবে বোঝার জন্য। শেষ ভাগে আমরা ২টি মোটরের জন্য কাজগুলো করব।

শর্ত—

Port	Signal Input
ena	255
in1	1
in2	0

ছক ৪.৪ : পোর্ট আর ইনপুট

কোডটি নিম্নরূপ—

```
int ena=3;
int enb=5;
```

```
int in1=0;
int in2=1;
int in3=2;
int in4=4;

void setup ()
{
  pinMode (ena, OUTPUT);
  pinMode (in1, OUTPUT);
  pinMode (in2, OUTPUT);
}
void loop ()
{
  analogWrite (ena, 255);
  digitalWrite (in1, 1);
  digitalWrite (in2, 0);
}
```

কোডটি আপলোড দিয়ে চালিয়ে দেখলে আউটপুটের ব্যাপারটি পরিষ্কার হবে। ১ নম্বর মোটরটি ঘোরা শুরু করবে এবং ফুল স্পিডে ঘুরতে থাকবে।

এটি করা শেষ হলে আমরা ২ নম্বর পরীক্ষাটি করব।

২.

শর্ত—

Port	Signal Input
ena	255
in1	0
in2	1

ছক ৪.৫ : পোর্ট আর ইনপুট

কোডটি নিম্নরূপ—

```
int ena=3;
int enb=5;
int in1=0;
int in2=1;
int in3=2;
```

```
int in4=4;

void setup ()
{
  pinMode (ena, OUTPUT);
  pinMode (in1, OUTPUT);
  pinMode (in2, OUTPUT);
}
void loop ()
{
  analogWrite (ena, 255);
  digitalWrite (in1, 0);
  digitalWrite (in2, 1);
}
```

কোডটি আপলোড দিয়ে চালালে দেখা যাবে যে আগে মোটরটি যে দিকে ঘুরছিল এখন তার বিপরীত দিকে ঘুরছে।

এটি শেষ হলে এবার ২য় মোটরের জন্য একই কাজ করতে হবে।

এবারে আমরা যেটি করব, সেটি হচ্ছে স্পিড টেস্ট। এ জন্য আগের সেটআপই থাকবে।

৩.

শর্ত—

Port	Signal Input
ena	150
in1	1
in2	0

ছক ৪.৬ : পোর্ট আর ইনপুট

কোডটি নিম্নরূপ—

```
int ena=3;
int enb=5;
int in1=0;
int in2=1;
int in3=2;
int in4=4;
```

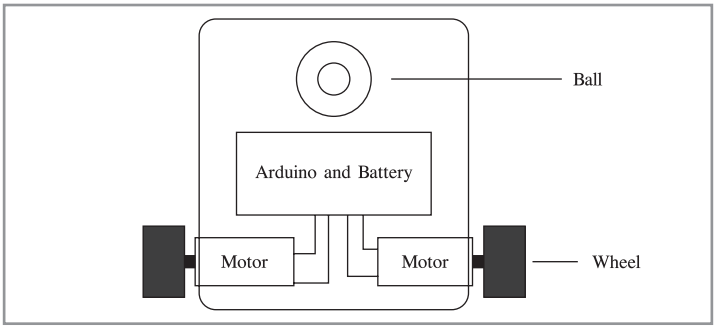
```
void setup ()
{
  pinMode (ena, OUTPUT);
  pinMode (in1, OUTPUT);
  pinMode (in2, OUTPUT);
}
void loop ()
{
  analogWrite (ena, 150);
  digitalWrite (in1, 1);
  digitalWrite (in2, 0);
}
```

কোডটি আপলোড দিয়ে চালিয়ে দেখলে দেখা যাবে ১ নম্বর পরীক্ষাটিতে ১ নম্বর মোটরের স্পিড যে রকম ছিল এখন স্পিড তার চেয়ে কমে যাবে। প্রায় অর্ধেক বলা চলে।

এবারে আমরা খুব গুরুত্বপূর্ণ একটি কাজ করব। এখানে আমরা যেটি করব তার জন্য লাগবে একটি ছোট পিভিসি বোর্ড এবং বল ক্যাসটার। এখানে পিভিসির পেছনে ২ পাশে মোটরসহ চাকা যুক্ত করতে হবে এবং সামনে থাকবে বল। ব্যাটারি থাকবে মাঝখানে।

**** এখানে মোটর হিসেবে আমরা গিয়ার মোটর ব্যবহার করব এটি মনে রাখতে হবে। ****

চিত্র ৪.৭-এর মতো।



চিত্র ৪.৭ : স্পিড টেস্ট ডায়াগ্রাম

এখানে ডিজাইন অনুযায়ী লাগাতে হবে প্রয়োজনীয় উপকরণগুলো। মোটরগুলোর সাথে চাকা থাকতে হবে। ব্যাটারি লাইপো ব্যাটারি হলে

ভালো হয়। জিনিসগুলো একসাথে লাগানোর জন্য স্ক্রু থাকলে ভালো হয়। স্ক্রু পাওয়া যায় পাইকারিতে। পাটুয়াটুলীতে ইলেকট্রনিকসের দোকানগুলোর পাশেই থাকে এসব দোকান। কোথায় কী কী পাব সেগুলো সম্পর্কে বইয়ের শেষে বলা হবে। এখন আমরা এ রকম সেটআপ হয়ে গেলে নিচের কোডটি লিখে আপলোড দেব।

```
int ena=3;
int enb=5;
int in1=0;
int in2=1;
int in3=2;
int in4=4;

void setup ()
{
  pinMode (ena, OUTPUT);
  pinMode (enb, OUTPUT);
  pinMode (in1, OUTPUT);
  pinMode (in2, OUTPUT);
  pinMode (in3, OUTPUT);
  pinMode (in4, OUTPUT);
}
void loop ()
{
  analogWrite (ena, 255);
  analogWrite (enb, 255);
  digitalWrite (in1, 1);
  digitalWrite (in2, 0);
  digitalWrite (in3, 1);
  digitalWrite (in4, 0);
 }
```

মোটরের সংযোগ ঠিকমতো থাকলে এই ছোট রোবট গাড়িটি সামনের দিকে চলে আগাবে। অথবা সংযোগ উল্টো থাকলে একটি নির্দিষ্ট বিন্দুকে কেন্দ্র করে ঘুরতে থাকবে।

আগের মতোই বিভিন্ন কম্বিনেশনে বিভিন্ন ইনপুট সিগন্যালের জন্য পরীক্ষা করে দেখতে হবে আউটপুট কেমন আসে।

এবার আমরা একটি ছোট প্রজেক্ট করব। একটি ছোট রোবট গাড়ি বানাব। গাড়িটি একবার সামনে যাবে ৫ সেকেন্ড ধরে তারপর ঘুরে

পেছনে ফিরবে এবং শেষে আবার আগের জায়গায় ফিরে আসবে। এখানে সেটআপটি এ রকমই থাকবে। শুধু কোডটি পরিবর্তন করতে হবে। কোডটি নিম্নরূপ—

```
int ena=3;
int enb=5;
int in1=0;
int in2=1;
int in3=2;
int in4=4;

void setup ()
{
  pinMode (ena, OUTPUT);
  pinMode (enb, OUTPUT);
  pinMode (in1, OUTPUT);
  pinMode (in2, OUTPUT);
  pinMode (in3, OUTPUT);
  pinMode (in4, OUTPUT);
}
void loop ()
{
  analogWrite (ena, 255);
  analogWrite (enb, 255);
  digitalWrite (in1, 1);
  digitalWrite (in2, 0);
  digitalWrite (in3, 1);
  digitalWrite (in4, 0);
  delay (5000);
  ////robot turning
  analogWrite (ena, 100);
  analogWrite (enb, 100);
  digitalWrite (in1, 1);
  digitalWrite (in2, 0);
  digitalWrite (in3, 0);
  digitalWrite (in4, 1);
  delay (500); //// turning time
  analogWrite (ena, 255); /// reaching destination
  analogWrite (enb, 255);
 digitalWrite (in1, 1);
 digitalWrite (in2, 0);
```

```
  digitalWrite (in3, 1);
  digitalWrite (in4, 0);
  delay (5000);
}
```

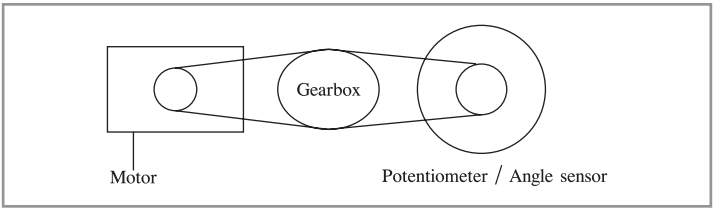
এখানে ঘোরার সময়টা নির্ভর করছে মোটরের ঘূর্ণন গতি, ব্যাটারির ভোল্টেজের ওপরে। কাজেই এখানে সিগন্যাল ও সময় কত হবে সেটি আগে থেকে কয়েকবার চালিয়ে নির্ধারণ করতে হবে। তারপর ওখানে বসাতে হবে।

সার্ভো মোটর

সার্ভো মোটর সাধারণ গিয়ার মোটরের মতোই। পার্থক্য শুধু এখানে সেন্সর এবং পটেনশিওমিটারের একটি বিশেষ সার্কিট থাকে। এর কাজ সাধারণ গিয়ার মোটরের মতো ঘোরা না। যেকোনো রোবটিক অ্যাপলিকেশনে একটি প্রধান কাজ হচ্ছে মোটর দিয়ে একটি নির্দিষ্ট মাপের কোণ কন্ট্রোল করা। ধরা যাক আমি একটি রোবটিক হাত তৈরি করব। এখন রোবটিক হাত কী করে? একটি রোবটিক হাতের কয়েকটি জয়েন্ট রয়েছে। ভিন্ন ভিন্ন জয়েন্টের জন্য কোণ ভিন্ন। এখন এখানে আমরা সেন্সরবিহীন সাধারণ গিয়ার মোটর যদি ব্যবহার করি তবে সে ক্ষেত্রে সঠিকভাবে মাপমতো কোণ রাখা সম্ভব হবে না। এই সমস্যার জন্য সার্ভো মোটর ব্যবহার করতে হয়।

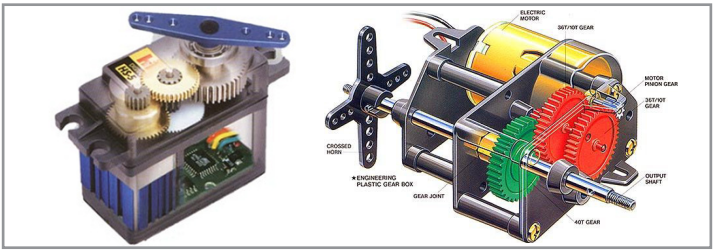
সার্ভো মোটরের থিওরি

সার্ভো মোটরে আসলে একটি মোটরের সাথে একটি পটেনশিওমিটার লাগানো থাকে। মোটরের বিভিন্ন কোণের জন্য পটেনশিওমিটারে বিভিন্ন কোণ পাওয়া যায়। এখানে মনে রাখতে হবে যে মোটরের সাথে একটি ছোট গিয়ারবক্স লাগানো থাকে। এই গিয়ারবক্স মোটরের গতি কমিয়ে লোড নেওয়ার ক্ষমতা বাড়িয়ে দেয়। এখন এই গিয়ারবক্সের আউটপুটের সাথেই একটি পটেনশিওমিটার লাগানো থাকে। এই পটেনশিওমিটার ঘুরে যদি মোটরটি ঘোরে। একটি ছবি দিলে বোঝা যাবে।



চিত্র ৪.৮ : সার্ভো মোটরের কার্যপদ্ধতি

এখানে ছবি থেকে দেখে বলা চলে যে মোটরটি গিয়ারবক্সকে ঘোরাচ্ছে এবং গিয়ারবক্স ঘোরাচ্ছে পটেনশিওমিটারকে। আরও কয়েকটি ছবি থেকে বোঝা যাবে ব্যাপারটি।



চিত্র ৪.৯ : সার্ভো মোটর

ছবিগুলো থেকে এ বিষয়টি পরিষ্কার হবে। এখানে খুব সহজেই বোঝা যাচ্ছে যে মোটরটি ড্রাইভ করছে গিয়ারবক্সকে এবং গিয়ারবক্সটি ড্রাইভ করছে পটেনশিওমিটারকে। পরে পটেনশিওমিটার থেকে আউটপুট ফিডব্যাক যেটি পাওয়া গেল সেই অনুযায়ী মোটরটি তার অবস্থান ঠিক করবে। সেটি কীভাবে?

অনেকগুলো প্রশ্ন থেকে যায়। যখন ভোল্টেজ কম থাকবে তখন কী হবে? কিংবা সিগন্যাল দেওয়া বন্ধ হয়ে গেলে তখন?

এখানে ভেতরে ছোট একটি কন্ট্রোলার, ১টি অ্যামপ্লিফায়ার আর কিছু সার্কিট কম্পোনেন্ট থাকে সাথে। এখানে আমরা একটি ছোট চার্ট দেখব একটি পটেনশিওমিটারের জন্য।

কোণ (ডিগ্রি)	রোধ (ওহম)
০	০
৩০	১৪০০০

৬০	২৮০০০
৯০	৪২০০০
১২০	৫৬০০০
১৫০	৭০০০০
১৮০	৮৪০০০

ছক ৪.৭ : কোণের সাথে রোধের পরিবর্তন

এখানে সম্পর্কটি সরলরৈখিক রাখা হলো কাজের সুবিধার জন্য। প্রথমে মোটর ঘুরে ধরা যাক ঘুরে ৩০ ডিগ্রিতে এল। তখন পটেনশিওমিটারেও ৩০ ডিগ্রি কোণ থাকবে। রোধ দেখা গেল ১৪০০০ ওহম। এখানে মোটরের সাথে সংযুক্ত কন্ট্রোলারটি যখন দেখবে ৩০ ডিগ্রিতে এসেছে তখনই মোটর থামিয়ে দেবে। এখন প্রশ্ন কন্ট্রোলারটি কীভাবে বুঝবে কত কোণে আছে? এখানে রোধের মান যে পরিবর্তন হচ্ছে সেটি থেকেই সহজেই বুঝতে পারবে।

এখানে আমি খুব সহজভাবে বোঝালাম। বাস্তবে সার্ভো মোটরের থিওরি অনেক কঠিন। এতটা সহজ না। এখানে আমি যে সরলরৈখিক সম্পর্ক দেখিয়েছি বাস্তবে এ রকম সরলরৈখিক সম্পর্ক থাকে না। এখানে এনকোডার ডিকোডার, তারপর এরর বা ভুল সংশোধনের জন্য আলাদা সার্কিটের একটি অংশ থাকে।

একটি সার্ভো মোটর হাতে নিয়ে দেখা যাবে এর থেকে ৩টি তার বের হয়েছে। সাধারণ মোটর থেকে ২টি বের হয়। এখানে সার্ভো মোটর থেকে ৩টি। ৩টির মধ্যে লক্ষ করলে দেখা যাবে যে ১টি তার মৃদু লাল অথবা লাল অথবা কমলা একটি তার কালো অথবা মৃদু বাদামি এবং অপরটি হলুদ রঙের। এখানে লাল এবং কালো রঙের তারটি যথাক্রমে ব্যাটারির ধনাত্মক ও ঋণাত্মক প্রান্তে যুক্ত হবে। আর হলুদ রঙের যেটি সেটি হচ্ছে প্রোগ্রামিং পোর্ট। এখান থেকে আরডুইনোতে যাবে। আরডুইনোর PWM পোর্টগুলোর সাথেই যুক্ত হবে। এটি মনে রাখতে হবে।

এখন আরডুইনোতে প্রোগ্রামিং করা অনেক সহজ সার্ভো মোটরের জন্য। কিন্তু আগে ব্যবহার হওয়া AVR কিংবা PIC-এর জন্য সেটি কঠিন ছিল। এখানে নিচে একটি চার্টে সার্ভো মোটর চালানোর জন্য প্রয়োজনীয় লাইব্রেরি এবং আর অন্যান্য ফাংশন সম্পর্কে দেওয়া হলো।

হেডার ফাইল	Servo.h (#include<Servo.h>)	কাজ
সার্ভো ডিক্লেয়ার	Servo s1	এটি অনেকটাই সি প্রোগ্রামিংয়ের ইনটিজার ডিক্লেয়ার এর মত। এখানে সার্ভো মোটর ডিক্লেয়ার দেওয়া হচ্ছে। প্রতিটি নতুন সার্ভো মোটরের জন্য একটি করে নতুন হবে।
ফাংশন	(servo motor's name).attach	সার্ভো মোটর ডিক্লেয়ার দেওয়ার পরে কোণ পোর্টে যুক্ত আছে সেটি বোঝাতে ব্যবহার করা হয়। void setup () অংশে ব্যবহার করা হয়।
	(servo motor's name).write	সার্ভো মোটরটি কত কোণে থাকবে সেটি বোঝাতে ব্যবহার হয়। void loop () অংশে ব্যবহার করা হয়।

ছক ৪.৮ : সার্ভো মোটরের জন্য প্রয়োজনীয় লাইব্রেরি ফাংশন এবং এর ব্যবহার

প্রজেক্ট ২ : সার্ভো মোটর নির্দিষ্ট কোণে রাখা

এই প্রজেক্টটির মাধ্যমে আমরা শিখব কীভাবে সার্ভো মোটরের মাধ্যমে একটি নির্দিষ্ট কোণে রাখা যায়। মানে স্থির রাখা যায়।

উপকরণ—

১. ১টি সার্ভো মোটর

২. আরডুইনো

৩. জাম্পার তার

সংযোগ—

সার্ভো মোটর	আরডুইনো
লাল বা কমলা তার	5 volt
কালো বা বাদামি তার	GND
হলুদ তার	3 (PWM)

ছক ৪.৯ : সার্ভো মোটরের তারের সংযোগ

সংযোগগুলো জাম্পার তার দিয়ে দিতে হবে।

সংযোগ দিয়ে নিচের কোডটি আপলোড দিতে হবে।

```
#include <Servo.h>
Servo s1;
void setup ()
{
  s1.attach(3);
}
void loop ()
{
  s1.write (60);
}
```

কোডটি আপলোড দিয়ে চালালে দেখা যাবে সার্ভো মোটরটি ৬০ ডিগ্রিতে স্থির হয়ে থাকবে। এবারে একই সেটআপে আমরা আরেকটি কোড আপলোড দেব।

```
#include <Servo.h>
Servo s1;
void setup ()
{
  s1.attach(3);
}
void loop ()
{
  s1.write (60);
  delay (500);
 s1.write (70);
  delay (500);
}
```

এবারে এই কোডটি আপলোড দিলে দেখা যাবে ৫০০ মিলিসেকেন্ড অন্তর একবার ৬০ ডিগ্রিতে থাকবে এবং আরেকবার ৭০ ডিগ্রিতে থাকবে। বারবার অবস্থান পরিবর্তন করবে এবং বলা চলে যতক্ষণ পাওয়ার আছে, ততক্ষণ চলতে থাকবে এভাবেই।

এখন আমরা আরেকভাবে চালাব। এখানে আমরা ০ ডিগ্রি থেকে শুরু করে ১৮০ ডিগ্রি পর্যন্ত যাবে। মাঝে একটু সময়ের পার্থক্য থাকবে। এরপর আবার ০ ডিগ্রিতে যাবে। এরপরে এভাবে আবার ১৮০ ডিগ্রিতে যাবে।

```
#include <Servo.h>
Servo s1;
void setup ()
{
  s1.attach(3);
}
void loop ()
{
  for (int i=1;i<=180;i++)
 {
    s1.write (i);
    delay (50);
  }
}
```

নিজেদের একটি কাজ করতে হবে। কাজটি হচ্ছে একটি সেন্সরের মাধ্যমে সার্ভো মোটরের পজিশন নির্ধারণ করতে হবে। যেকোনো একটি সেন্সরের মাধ্যমে সেন্সরের বিভিন্ন রোধের জন্য সার্ভো মোটরের পজিশন নির্ধারণ হবে। সেন্সরটি হচ্ছে পটেনশিওমিটার। এখানে নবটি ঘুরিয়ে সার্ভো মোটরের পজিশন নির্ধারণ হবে।

আমাদের সব বেসিক আপাতত এতটুকুই। এরপরে থেকে আমরা কয়েকটি গুরুত্বপূর্ণ প্রজেক্ট, অর্থাৎ ছোটখাটো রোবট তৈরি করব। প্রজেক্টগুলো কীভাবে তৈরি করতে হবে, সে সম্পর্কে বলা হবে। এখানে বেসিক নিয়ে বলা হবে না খুব বেশি।

পঞ্চম অধ্যায়

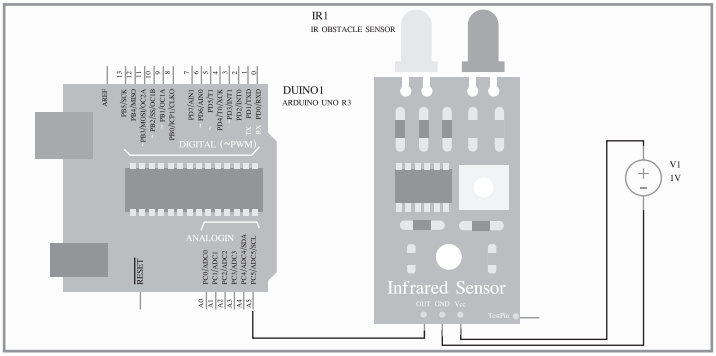
সেন্সর

আই আর সেন্সর

প্রয়োজনীয় উপকরণ—

১. Arduino microcontroller

২. IR sensor



চিত্র ৫.১ : ইনফ্রারেড সেন্সর

সেন্সরের জন্য যে রকম সেটআপ দিতে হবে—

সেন্সর	Arduino সংযোগ
Vcc	5V
GND	GND
OUT	Analog pin A5

ছক ৫.১ : সেন্সরের সংযোগ

সেন্সরের মান বের করার জন্য নিচের কোডটি আপলোড করতে হবে।

```
int sensorPin = A5; //analog pin 0

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

void loop () {
  int val = analogRead(sensorPin);
  Serial.println(val);

  //just to slow down the output - remove if trying to catch an object passing by
  delay (100);

}
```

এই কোডটা আপলোড করে সিরিয়াল মনিটরে ক্লিক করলে সেন্সরের মান দেখা যাবে।

চিত্র ৫.২ : সেন্সরের মান দেখা যাচ্ছে

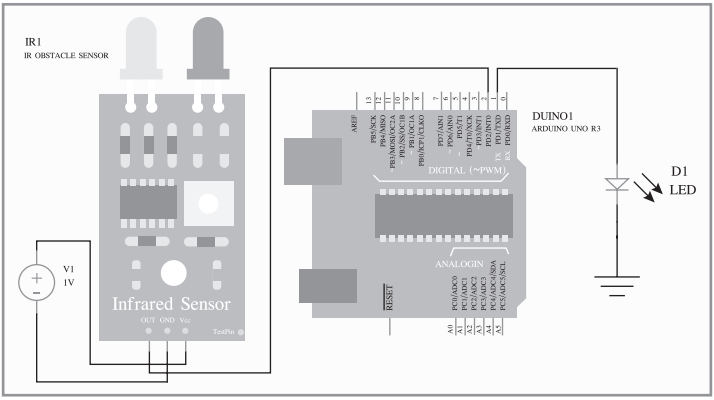
আই আর প্রক্সিমিটি সেন্সর

প্রয়োজনীয় উপকরণ—

১. Arduino microcontroller

২. IR sensor

৩. LED



চিত্র ৫.৩ : আইআর প্রক্সিমিটি সেন্সর

সেন্সরের জন্য যে রকম সেটআপ দিতে হবে—

সেন্সর	Arduino সংযোগ
Vcc	5V
GND	GND
OUT	Analog pin A0

ছক ৫.২ : সেন্সরের সংযোগ সেটআপ

LED-এর জন্য যে সংযোগ দিতে হবে—

LED	Arduino সংযোগ
+ pin	Digital pin 1
GND	GND

ছক ৫.৩ : লেডের সংযোগ সেটআপ

এরপর নিচে কোডটি আপলোড করতে হবে—

```
const int ProxSensor=2;
void setup() {
  pinMode(1, OUTPUT);
 pinMode(ProxSensor, INPUT);
}

void loop() {
 if(digitalRead(ProxSensor)==HIGH)        //Check the sensor output
```

```
  {
    digitalWrite(13, HIGH);   // set the LED on
  }
  else
  {
    digitalWrite(13, LOW);    // set the LED off
  }
  delay(100);                 // wait for a second
}
```

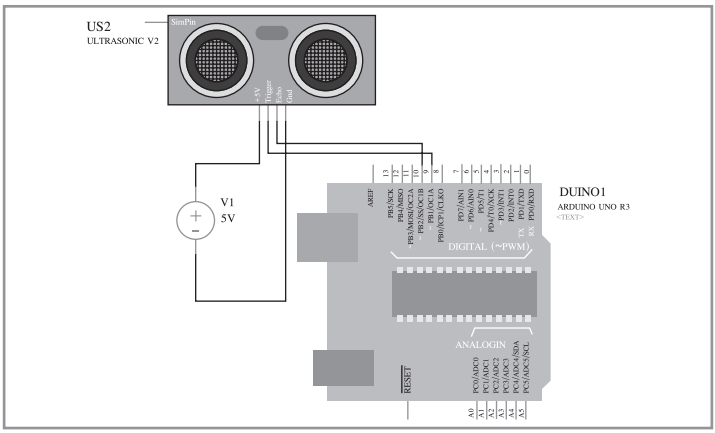
কোডটা আপলোড করার পর দেখা যাবে, একটা নির্দিষ্ট দূরত্বে সেন্সরে বাধা আসলে LED বাতিটি জ্বলবে আবার বাধা সরালে বাতি নিভবে।

সোনার সেন্সর

প্রয়োজনীয় উপকরণ—

১. Arduino microcontroller

২. Sonar sensor



চিত্র ৫.৪ : সোনার সেন্সর

সেন্সরের জন্য যে রকম সেটআপ দিতে হবে—

সেন্সর	Arduino সংযোগ
Vcc/5v	5V

GND	GND
Trigger	Digital pin 9
Echo	Digital pin 10

ছক ৫.৪ : সেন্সর সেটআপ

এরপর নিচের কোডটি আপলোড করতে হবে—

```
const int trigPin = 9;
const int echoPin = 10;
long duration;
int distanceCm, distanceInch;
void setup() {
Serial.begin(9600);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
}
void loop() {
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distanceCm= duration*0.034/2;
distanceInch = duration*0.0133/2;
Serial.print("Distance : "); // Prints string "Distance" on the Serial monitor
Serial.print(distanceCm); // Prints the distance value from the sensor
Serial.print(" cm");
delay(10);
Serial.setCursor(0, 1);
Serial.print("Distance : ");
Serial.print(distanceInch);
Serial.print(" inch");
delay(10);
}
```

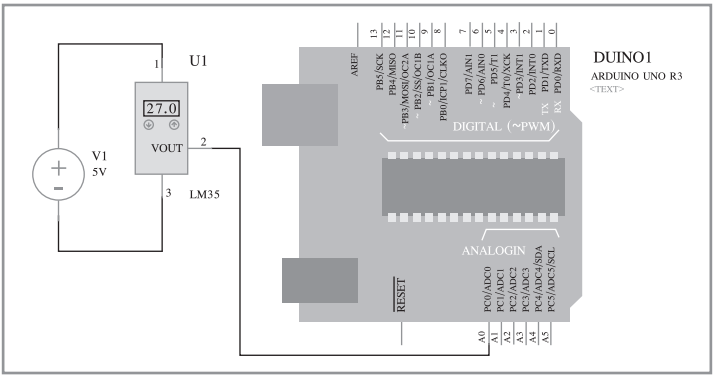
এই কোডটা আপলোড করে সিরিয়াল মনিটরে ক্লিক করলে সেন্সরের মাধ্যমে সামনে কোনো বাধা থাকলে সেটার দূরত্ব দেখাবে।

টেম্পারেচার সেন্সর

প্রয়োজনীয় উপকরণ—

১. Arduino microcontroller

২. LM35 sensor



চিত্র ৫.৫ : টেম্পারেচার সেন্সর

সেন্সরের জন্য যে রকম সেটআপ দিতে হবে—

সেন্সর	Arduino সংযোগ
1	5V
2	GND
3	Analog pin A0

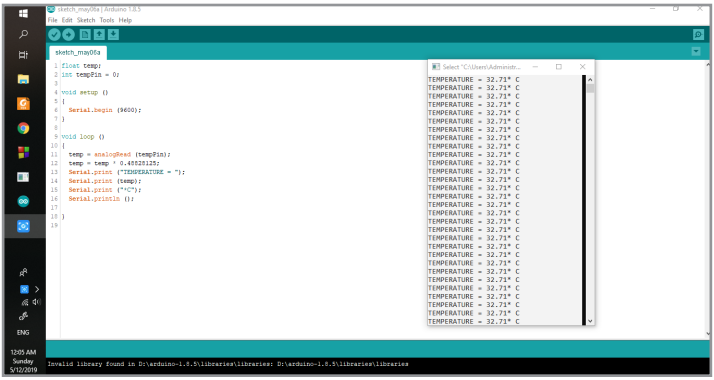
ছক ৫.৫ : সেন্সর সেটআপ

এরপর নিচে কোডটি আপলোড করতে হবে—

```
int val;
int tempPin = 1;

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

```
val = analogRead(tempPin);
float mv = ( val/1024.0)*5000;
float cel = mv/10;
float farh = (cel*9)/5 + 32;

Serial.print("TEMPRATURE = ");
Serial.print(cel);
Serial.print("*C");
Serial.println();
delay(1000);

/* uncomment this to get temperature in farenhite
Serial.print("TEMPRATURE = ");
Serial.print(farh);
Serial.print("*F");
Serial.println();
}
```

আপলোড করলে সিরিয়াল মনিটরে তাপমাত্রা দেখাবে—



চিত্র ৫.৬ : তাপমাত্রা দেখা যাচ্ছে মনিটরে

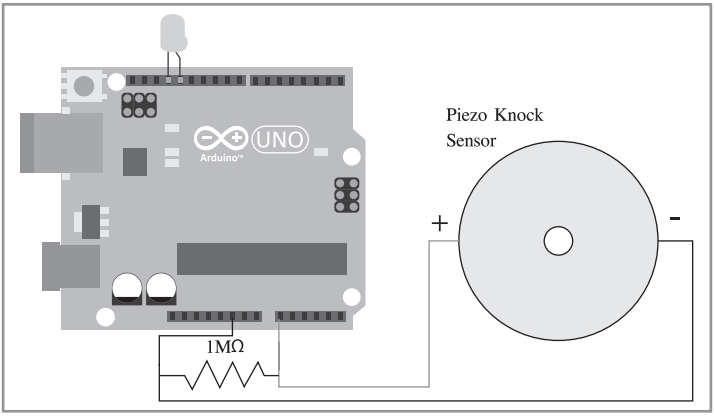
পিএজো ইলেক্ট্রিক সেন্সর

প্রয়োজনীয় উপকরণ—

১. Arduino microcontroller

২. piezo electric sensor

৩. 1M ohm resistance



চিত্র ৫.৭ : পিএজো ইলেক্ট্রিক সেন্সর

সেন্সরের জন্য যে রকম সেটআপ দিতে হবে—

সেন্সর	Arduino সংযোগ
+ ve	Analog pin A0
-ve	GND

ছক ৫.৬ : সেন্সর সেটআপ

এরপর নিচে কোডটি আপলোড করতে হবে—

```
int piezo_Pin= 0;
int LED_Pin= 13;
int threshold= 500; //Set the threshold levels

void setup()
{
Serial.begin(9600);
pinMode(LED_Pin, OUTPUT);
}
void loop()
{
int reading= analogRead(piezo_Pin);
Serial.println(reading);
if (reading > threshold)
{
digitalWrite(LED_Pin, HIGH);
```

```
delay(1000);
digitalWrite(LED_Pin, LOW);
}
}
```

কোডটি আপলোড দিলে সেন্সরটিতে টোকা দিলে আরডুইনোতে থাকা LED জ্বলবে।

স্ট্রেইন গজ

প্রয়োজনীয় উপকরণ—

১. Arduino microcontroller

২. লোড সেল (Load cell)

৩. লোড সেন্সর (Load sensor)

সেন্সরের জন্য যে রকম সেটআপ দিতে হবে—

লোড সেল

সেন্সর	Arduino সংযোগ
+ ve	5v
GND	GND
Signal pin	Load sensor input pin

ছক ৫.৭ : সেন্সর সেটআপ

লোড সেন্সর

সেন্সর	Arduino সংযোগ
+ ve	5v
GND	GND
Output pin	Analog pin A0

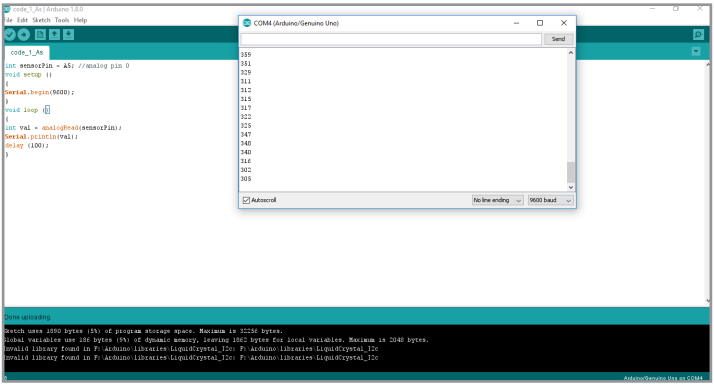
ছক ৫.৮ : সেন্সর সেটআপ

এরপর নিচের কোডটি আপলোড করতে হবে—

```
int sensorPin = A0;     // select the input pin for the Strain Gauge
int sensorValue = 0;  // variable to store the value coming from the sensor
```

```
void setup() {
  // declare the ledPin as an OUTPUT :
   Serial.begin(9600);
}

void loop() {
  // read the value from the sensor :
  sensorValue = analogRead(sensorPin);
  delay(1000);
  Serial.print("sensor = " );
  Serial.println(sensorValue);
}
```

Output হিসেবে সিরিয়াল মনিটরে সেন্সর মান আসবে যা পরে চাপ পরিমাপ করা যাবে।



চিত্র ৫.৮ : আউটপুট দেখা যাচ্ছে

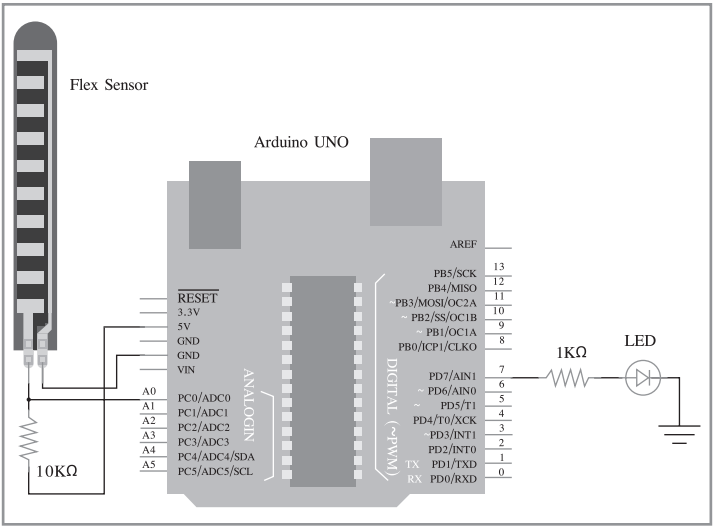
বেন্ডিং সেন্সর

প্রয়োজনীয় উপকরণ—

১. Arduino microcontroller

২. Bending sensor

৩. 22 k ohm



চিত্র ৫.৯ : Arduino Uno-তে বেন্ডিং সেন্সর দেখা যাচ্ছে

সেন্সরের জন্য যে রকম সেটআপ দিতে হবে—

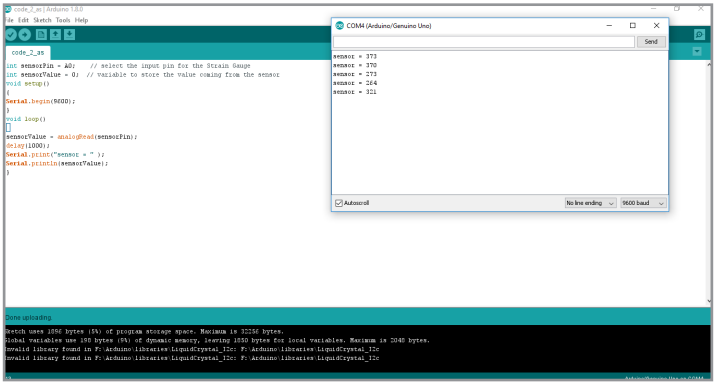
সেন্সর	Arduino সংযোগ
+ ve	5V
-ve	Analog pin A0

ছক ৫.৯ : সেন্সর সেটআপ

এরপর নিচে কোডটি আপলোড করতে হবে—

```
const int flexPin = A0;

void setup(){
  Serial.begin(9600);        //Begin serial communication
}

void loop(){
value = analogRead(flexPin);          //Read and save analog value from
potentiometer
Serial.println(value);                //Print value
delay(100);                           //Small delay
}
```

Code আপলোড দিলে সিরিয়াল মনিটরে সেন্সর মান আসবে যা পরে অন্যান্য ডিভাইসে ইনপুট হিসেবে ব্যবহার হয়।



চিত্র ৫.১০ : মান দেখা যাচ্ছে

ষষ্ঠ অধ্যায়

রোবটিকসের প্রোজেক্ট

লাইট ফলোয়ার রোবট

প্রয়োজনীয় উপকরণ—

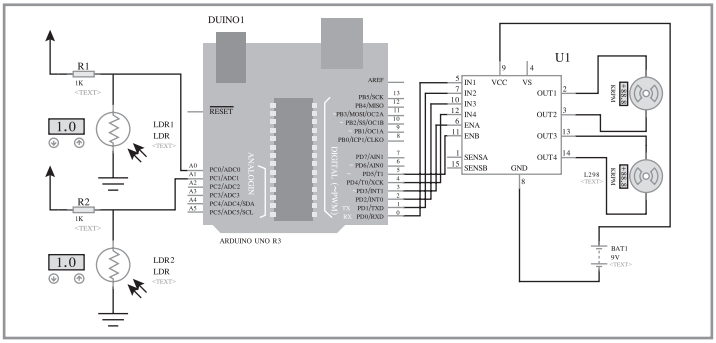
১. ২টি LDR সেন্সর

২. গিয়ার মোটর ২টি

৩. মোটর ড্রাইভার

৪. ব্যাটারি

৫. রেসিস্টর ১ কিলোওহম মানের



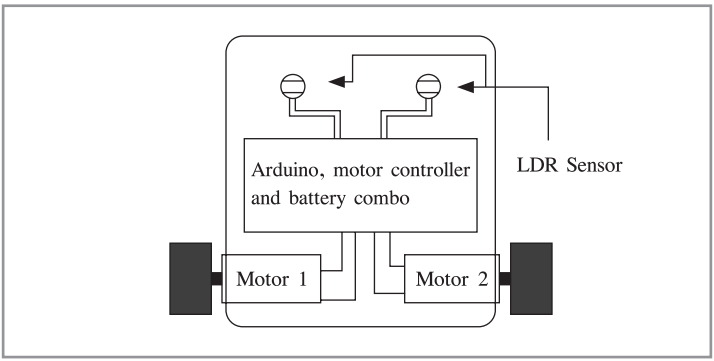
চিত্র ৬.১ : লাইট ফলোয়ার রোবটের সিমুলেটেড সেটআপ

সেন্সরের জন্য সেটআপটি কী রকম হবে সেটি নিচে দেওয়া হলো।

সেন্সর	সংযোগ
সেন্সর ১	অ্যানালগ পোর্ট ০
সেন্সর ২	অ্যানালগ পোর্ট ১

ছক ৬.১ : মোটর সেটআপ

রোবটটির গঠন হবে নিচের ছবির মতো।



চিত্র ৬.২ : লাইট ফলোয়ার রোবটের স্কেলেটাল কাঠামো

এখানে সামনে ১টি বল ক্যাসটার রাখতে হবে। মোটরের জন্য সেটআপের চার্ট ছক ৬.২-এ দ্রষ্টব্য।

মোটর ড্রাইভারের পোর্ট	আরডুইনো পোর্টে সংযোগ
ena	3
enb	5
in1	0
in2	1
in3	2
in4	4
out 1 + out 2	motor 1
out 3 + out 4	motor 2

ছক ৬.২ : মোটর সেটআপ

ব্যাটারি সংযোগ

ব্যাটারি সংযোগ একটু বিশেষভাবে দিতে হবে। এখানে আমরা লাইপো ব্যাটারি ব্যবহার করব। লাইপো ব্যাটারি ৭ থেকে শুরু করে ২০ ভোল্ট পর্যন্ত হয়। আমরা এখানে ১২ ভোল্ট ব্যবহার করব। লাইপো ব্যাটারির ধনাত্মক প্রান্ত সংযোগ দেব মোটর ড্রাইভারের VCC-এর সাথে এবং ব্যাটারির ঋণাত্মক প্রান্ত সংযোগ দেব ড্রাইভারের GND-এর সাথে। ড্রাইভারের ৫ ভোল্ট বা 5v লেখা পোর্ট থেকে একটি সংযোগ দিতে হবে আরডুইনোর Vin এ। আরডুইনোর GND এবং মোটর ড্রাইভারের GND একত্রে সংযোগ দিতে হবে।

এখানে আমাদের রোবটটির আউটপুট হবে অনেকটা এ রকম। যখন টর্চের আলো আমরা সেন্সরের ওপরে ফেলব, তখন রোবটটি চলবে।

সার্কিট তৈরি শেষ হলে এবং রোবটের যাবতীয় সেটআপ শেষ হলে আমরা কোডটি আপলোড দেব। এখানে মনে রাখতে হবে যে গিয়ার মোটর ব্যবহার করতে হবে।

```
int sen1 = A0;
int sen2 = A1;
int ena = 3;
int enb = 5;
int in1 = 0;
int in2 = 1;
int in3 = 2;
int in4 = 4;

void setup ()
{
  pinMode (ena, OUTPUT);
  pinMode (enb, OUTPUT);
  pinMode (in1, OUTPUT);
  pinMode (in2, OUTPUT);
  pinMode (in3, OUTPUT);
  pinMode (in4, OUTPUT);
  Serial.begin (9600);
}
void loop ()
{
 int val = analogRead (sen1);
```

```
  int val1 = analogRead (sen2);
  int val3=val+val1; //adding the value of the sensors
  if (val3>y && val3<x)
  {
    analogWrite (ena, 255);
    analogWrite (enb, 255);
    digitalWrite (in1, 1);
    digitalWrite (in3, 1);
  }
  else
  {
    analogWrite (ena, 0);
    analogWrite (enb, 0);
  }
}
```

সেন্সরের পাঠ পাওয়ার জন্য সাধারণ পরিবেশে আমাদের টর্চের আলো সেন্সরের ওপরে ফেলে দেখতে হবে। এখানে আমরা ২টি সেন্সরের পাঠকে যোগ করেছি এবং সেন্সরের অনেকগুলো মানের জন্য যোগফলের একটি সর্বনিম্ন এবং সর্বোচ্চ পাঠ বের করেছি। এখানে আমরা চাইলে লজিকগুলো পরিবর্তন করে যেকোনো মজার ফাংশন যোগ করা যাবে।

অটোমেটিক পাম্প কন্ট্রোল

বাসাবাড়ির যে পাম্পগুলো ব্যবহার করি আমরা সেগুলোতে চাইলেই অটোমেশন করা যায়। আমরা আরডুইনো ব্যবহার করেই এটি করতে পারি। এখানে আমরা একটি পানির লেভেল ইন্ডিকেটর সেন্সর থেকে পাঠ নেব। এবং সেই পাঠ থেকে লজিকের মাধ্যমে পাম্প বন্ধ হবে অথবা চালু হবে। এই প্রজেক্টটির মাধ্যমে আমরা জানব কীভাবে আরডুইনো ব্যবহার করে বাসাবাড়ির এ.সি সংযোগ চালানো যায়। এ প্রজেক্টটি থেকে পাওয়া জ্ঞান শুধু যে এ প্রজেক্টে ব্যবহার করা যাবে সেটি না; বরং যেকোনো এ.সি সোর্স থেকে চলা মাধ্যমকে আমরা কীভাবে আরডুইনো দিয়ে চালাব সেটি দেখব।

প্রয়োজনীয় উপকরণ—

১. একটি পাম্প

২. রিলে মডিউল

৩. প্রয়োজনীয় লম্বা তার

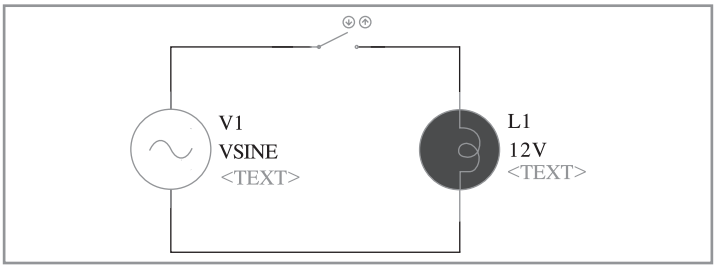
৪. আরডুইনো

৫. পানির উচ্চতা মাপার জন্য প্রয়োজনীয় সার্কিট

৬. ব্যাটারি বা ভোল্টেজ কনভার্টার (এখানে ভোল্টেজ কনভার্টার ব্যবহার করা হবে)

৭. জাম্পার তার

প্রথমে আমরা একটি এ.সি সংযোগ চালাব। ধরা যাক ১টি লাইট। এ.সি সংযোগের জন্য কতগুলো সুইচবোর্ড পাওয়া যায়। আর এ.সি বাল্ব চালানোর জন্য বোর্ড পাওয়া যায়। আমাদের ওগুলো দরকার হবে। এ.সি সংযোগ চালানো খুব সহজ। সুইচ বোর্ডে থেকে একটি তার লাগিয়ে তার এক প্রান্ত বাল্ব অথবা অন্য কোনো রোধের সাথে সংযোগ দিতে হবে।



চিত্র ৬.৩ : এসি সংযোগ দিয়ে বাল্ব জ্বালানো হচ্ছে

এখানে সংযোগটা দেখতে পাওয়া যাচ্ছে। সংযোগটি খুব সহজ। এখানে সাইন ভোল্টেজ সোর্স দিয়ে এসি ভোল্টেজ সোর্স বোঝানো হচ্ছে। সুইচ অন-অফ করলেই সংযোগটি চালু হবে আবার নিভবে।

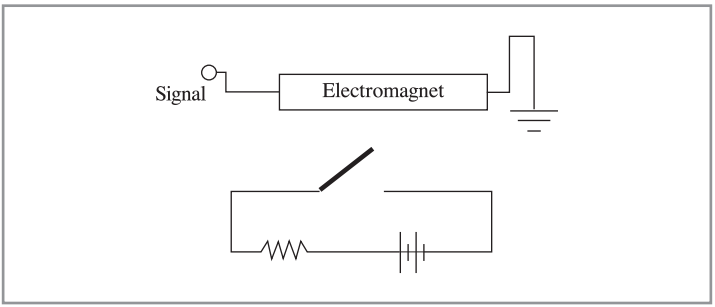
এবার আমরা রিলে মডিউল দিয়ে কাজটি করব। তবে এ কাজটি করার আগে অনেক সতর্কতা অবলম্বন করতে হবে। এসি ভোল্টেজ সোর্স ২২০ ভোল্ট মানের, তাই কারেন্ট শক খাওয়ার সম্ভাবনা অনেক। এ জন্য পরামর্শ হচ্ছে আগে ডিসি ৩০/৪০ ভোল্ট এ রকম ভোল্টের সার্কিট চালিয়ে তারপরে এসি সোর্স দিয়ে কাজ করা।

রিলে মডিউল

রিলে আসলে এক ধরনের সুইচ। ম্যাগনেটিক সুইচ। আমাদের পরিচিত সুইচগুলোর সাথে এটির পার্থক্য হচ্ছে রিলে মডিউল ডিজিটাল কোড অথবা একটি নির্দিষ্ট মানের ভোল্টেজ প্রয়োগে সুইচ অন করতে হয়।

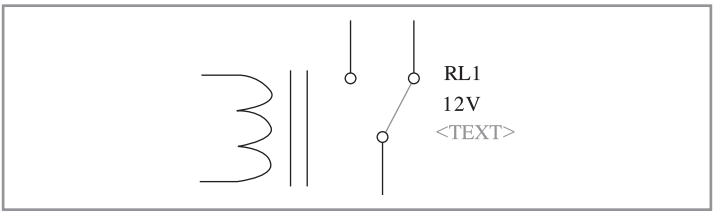
রিলের গঠন

রিলে একটি তড়িৎচৌম্বকওয়ালা একটি সুইচ। এখানে রিলেতে একটি বিশেষ সুইচ থাকে যেটি স্থায়ী চুম্বক। আরেকটি থাকে অস্থায়ী চুম্বক। কয়েলে তড়িৎ প্রবাহিত করলে সেটি তখন চুম্বকে পরিণত হয়।



চিত্র ৬.৪ : রিলের গঠন

এখানে যে তড়িৎ চুম্বকটি দেখা যাচ্ছে, সেটি সিগন্যাল পেলেই সক্রিয় হবে। আর সুইচের যে ঘন কালো অংশটি দেখা যাচ্ছে, সেটি হচ্ছে স্থায়ী চুম্বক। এখানে উল্লেখ্য যে তড়িৎ চুম্বকটি যখন সচল হবে, তখন সেটি স্থায়ী চুম্বককে নিজের দিকে টানবে। ফলে বর্তনীটির সংযোগ বিচ্ছিন্ন হয়ে যাবে। আর যখন তড়িৎ চুম্বকটির চৌম্বকত্ব থাকবে না, তখন সেটি বর্তনীটির সংযোগ দিয়ে দেবে। এখানে একটি দুর্বল স্প্রিং রয়েছে। সেটি তখন স্থায়ী চুম্বককে তার আগের জায়গায় নিয়ে আসবে। এভাবে রিলে মডিউল কাজ করে।



চিত্র ৬.৫ : সার্কিট ডায়াগ্রাম

অনেক রকম রিলে মডিউল পাওয়া যায়। সাধারণত ৫ ভোল্ট থেকে ১০ ভোল্টের হয় এগুলো। অধিকাংশ ৫ ভোল্ট। আর রিলে মডিউল প্রস্তুতকারীরা সাধারণত মডিউলের সাথে প্রয়োজনীয় সার্কিট সংযোগ দিয়ে দেয়। আরডুইনোতে যেহেতু ৫ ভোল্টের আউটপুট সবচেয়ে বেশি এ জন্য কেনার সময় ৫ ভোল্টের রিলে মডিউল কিনতে হবে।

এখানে আমরা প্রথমে যে কাজটি করব ২০/৩০ ভোল্টের একটি সার্কিট চালাব। কয়েকবার করার পরে পাঠকেরা চাইলে এসি সংযোগ নিয়ে কাজ করতে পারেন। তবে সেটি নিজের ঝুঁকি নিয়ে করতে হবে। অভিজ্ঞ মানুষের সাহায্য ছাড়া করা যাবে না।

প্রয়োজনীয় উপকরণ—

১. আরডুইনো

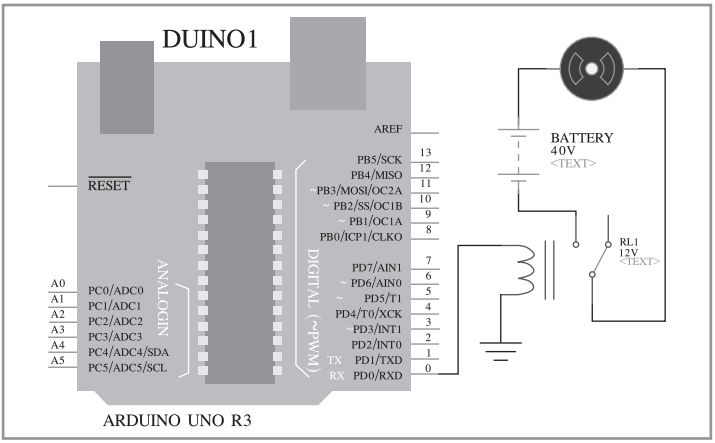
২. ৫ ভোল্টের রিলে মডিউল

৩. ৩০/৪০ ভোল্টের মানের ব্যাটারি। ৩০/৪০ ভোল্টের ব্যাটারি সহজে পাওয়া যায় না, সে জন্য ১২ ভোল্টের কয়েকটি ব্যাটারিকে সিরিজে সংযোগ দিয়ে কাজ চালাতে হবে।

৪. হাই ভোল্টেজের মোটর সর্বোচ্চ ৩০/৪০ ভোল্টের সংযোগে চলতে পারবে এমন।

৩০/৪০ ভোল্টের ব্যাটারি তৈরির জন্য ১২ ভোল্টের পাওয়া যায় এমন ব্যাটারি সিরিজে সংযোগ দিলেই হবে। যেহেতু আমরা ৫ ভোল্টের রিলে ব্যবহার করব, সেহেতু এখানে আরডুইনো থেকে ৫ ভোল্টের সংকেত দিলেই চলবে। এখানে আমরা ডিজিটাল সংকেত ব্যবহার করব। আরডুইনো থেকে ডিজিটাল পোর্ট ০ থেকে সংকেত যাবে রিলের সংকেত দেওয়ার অংশে। আরেক অংশ ব্যবহার করা হবে সাধারণ সুইচের মতোই।

এখানে সার্কিট ডায়াগ্রামটি নিম্নরূপ—



চিত্র ৬.৬ : সার্কিট ডায়াগ্রাম

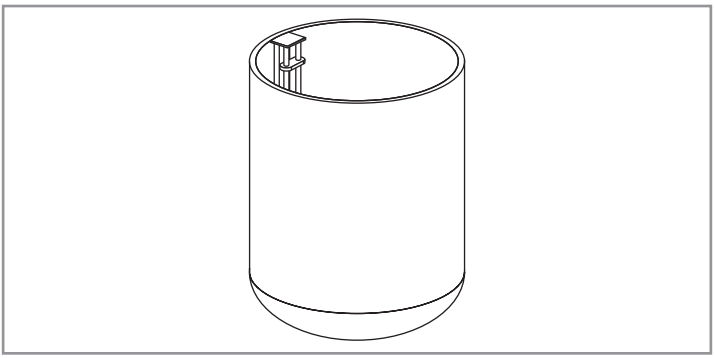
```
int relay = 0 ;
void setup ()
{
  pinMode (relay, OUTPUT);
}
void loop ()
{
  digitalWrite (relay, HIGH);
  delay (2000);
  digitalWrite (relay, LOW);
  delay (2000);
}
```

কোডটি খুব সাধারণ কোড। এটি চালালে শুধু ২ সেকেন্ড পরপর রিলে একবার চলবে আরেকবার বন্ধ হবে। অর্থাৎ হাই ভোল্টেজের মোটরটি একবার চালু হবে আরেকবার বন্ধ হবে।

এবার আমরা সেন্সর বেসিক একটি সিস্টেম তৈরি করব।

পানির লেভেল ইনডিকেটর হিসেবে সেন্সর পাওয়া যায় অনেক। পানির উচ্চতা পরিবর্তিত হলে ক্যাপাসিটেন্স পরিবর্তিত হয়। ওইভাবে পাওয়া যায়। আবার লেসার সেন্সর দিয়েও কাজটা করা যায়। কোন সেন্সর দিয়ে করা হবে, সেটার ওপরে নির্ভর করে প্রোগ্রামিং কোড তৈরি করতে হবে।

এখানে একটি ছোট ইলেকট্রো মেকানিক্যাল সিস্টেম তৈরি করা হবে। যে পদ্ধতিতে করা হবে, সেটি একটু জটিল। এর চেয়েও সহজে করা যায়। তবে জটিলভাবে করা হবে। কারণ বড় বড় প্রজেক্টে শুধু ইলেকট্রনিকস না জটিল মেকানিক্যাল সিস্টেমও সংযুক্ত থাকে।



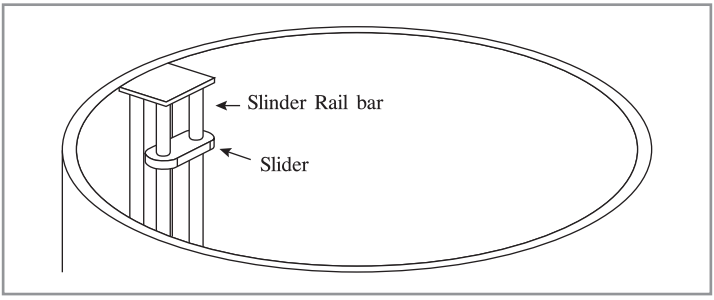
চিত্র ৬.৭ : পানির ট্যাংক

এখানে ছবিতে যে পানির ট্যাংক দেখা যাচ্ছে, ধরা যাক আমাদের পানির ট্যাংক অনেকটা এ রকম। ডিজাইনটা Solidworks সফটওয়্যার দিয়ে করা। তবে এখানে কিছু জিনিস সংযুক্ত করতে হবে।

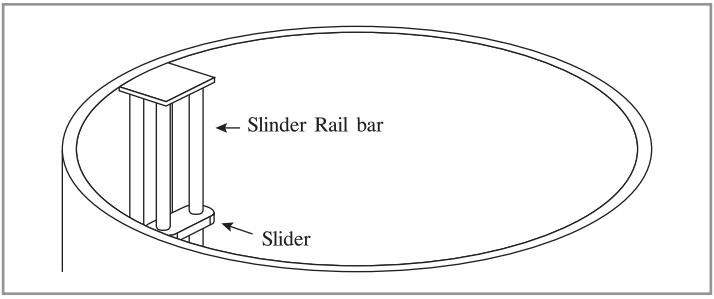
১. স্লাইডার

২. ছিদ্রযুক্ত একটি রেইল

৩. পানিতে ভাসবে এমন একটি বস্তু



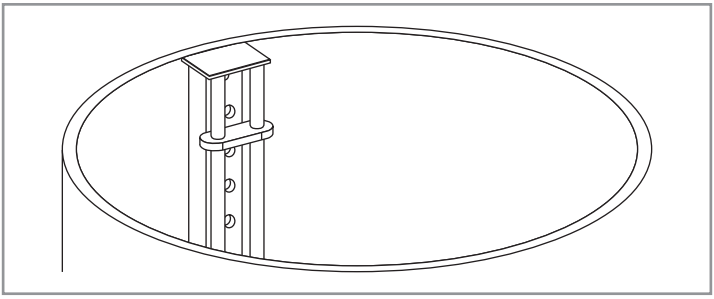
চিত্র ৬.৮ : স্লাইডারের প্রাথমিক অবস্থা



চিত্র ৬.৯ : স্লাইডার নিচে নামছে

৩টি ছবিতে স্লাইডারের ওপরে থেকে নিচে ৩টি অবস্থান দেখা যাচ্ছে। এখানে স্লাইডারটি ওপরে থেকে নিচে ওঠানামা করে।

*** একটা ব্যাপার মাথায় রাখতে হবে যে স্লাইডারটি এমন পদার্থের তৈরি হতে হবে যেন পানিতে ভাসে। পানির উচ্চতা বাড়লে তখন স্লাইডারটিও ওপরে-নিচে নামবে। আর স্লাইডার রেইল বার স্টেইনলেস স্টিলের হতে হবে। স্লাইডারের হোলের ব্যাস আর স্লাইডারের বারের ব্যাসে একটি বড় পার্থক্য থাকতে হবে। তাহলে খুব সহজেই স্লাইডারটি ওপরে-নিচে উঠবে-নামবে।

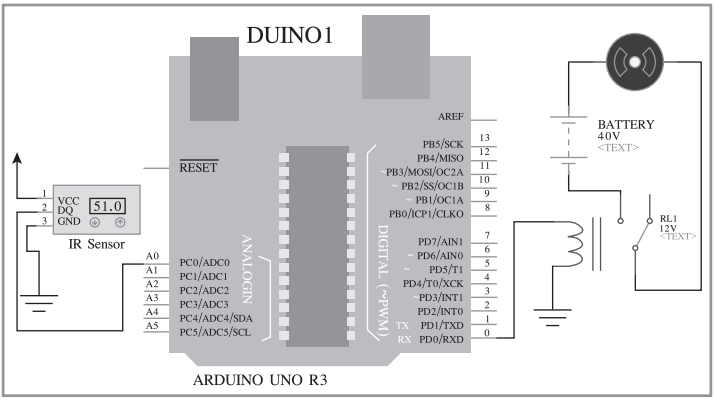


চিত্র ৬.১০ : স্লাইডারের পেছনে ছিদ্র দেখা যাচ্ছে

এখানে স্লাইডারের পেছনে যে অংশ রয়েছে, সেখানে অনেকগুলো ছিদ্র দেখা যাচ্ছে। ছিদ্রগুলো আলাদাভাবে করা একটি বারের মধ্যে ছিদ্র করে ট্যাংকের সাথে লাগানো হয়েছে। *** ট্যাংকে কোনো ছিদ্র করা হয়নি।

এখন স্লাইডারটিতে আমরা সার্কিট বসাব। সার্কিটটি আলট্রাসনিক সোনার অথবা আই.আর সেন্সর দিয়ে করলেও চলবে। তবে আই.আর দিয়ে করলেই ভালো। সার্কিটটি পানিরোধক বাক্সের মধ্যে রাখলে ভালো হয়।

সার্কিট ডায়াগ্রামটি নিম্নরূপ।

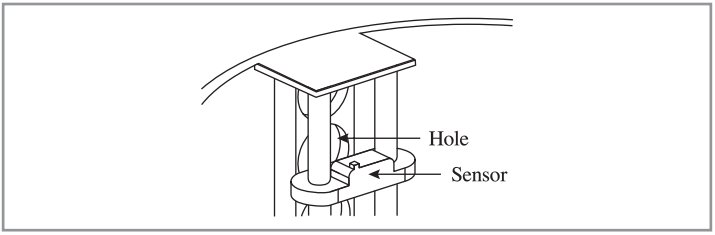


চিত্র ৬.১১ : সার্কিট ডায়াগ্রাম

আই.আর সেন্সরের ৩টি তার থাকে। এর মধ্যে লাল তারটি ধনাত্মক, কালো তারটি ঋণাত্মক। আরেকটি হলুদ রঙের তার রয়েছে সেটি প্রোগ্রামিং পোর্ট মানে এই প্রান্ত দিয়ে সিগন্যাল নেবে আরডুইনো। লাল তারটি সংযোগ দিতে হবে আরডুইনোর ৫ ভোল্টের কালো তারটি গ্রাউন্ডের সাথে। আর হলুদ তারটি দিতে হবে ০ নম্বর অ্যানালগ পোর্টের সাথে।

রিলের সংযোগ : রিলে মডিউলের সংযোগের জন্য ম্যাগনেটিক কয়েলের অংশে একটি প্রান্ত ডিজিটাল পোর্ট ০-তে আরেকটি গ্রাউন্ডে সংযুক্ত হবে। অপর প্রান্তে সাধারণ সুইচের মতোই কাজ করবে।

** এখানে একটি ব্যাপার লক্ষ রাখতে হবে যে স্লাইডারে আই.আর সেন্সর বড় ছিদ্রযুক্ত দণ্ডটি যেদিকে, সেদিকে মুখ করে রাখতে হবে। চিত্র ৬.১২ দ্রষ্টব্য।



চিত্র ৬.১২ : সেন্সর সহ স্লাইডার হোলের সামনে দেখা যাচ্ছে

আই.আর সেন্সরের আই.আর এলইডি থেকে প্রতিফলিত আলো যখন গর্তের ভেতরে পড়বে, তখন আলোর প্রতিফলনের জন্য একবার দূরত্ব কম হবে আরেকবার বেশি হবে। আলোর দূরত্ব পরিবর্তনের সাথে সাথে কম্পাঙ্ক পরিবর্তন হয়।

```
int sensor_pin = A0;
int relay = 0;
int count=0;
int n, x, y;
// say there are n number of holes for a specific height
void setup ()
{
  Serial.begin (9600);
  pinMode (sensor_pin, INPUT);
  pinMode (relay, OUTPUT);
}

void loop ()
{
  int val = analogRead (sensor_pin);
 digitalWrite (relay, HIGH); // motor on
  // suppose when the sensor faces the hole the reading is x or a little daviation
from x
  //else when in plane y or a little deviation from y
  if (val==x) // make this logic according to the calibration of sensor
  {
    count=count+1;
    if (count == n)
    {
      digitalWrite (relay, LOW); // switch off / motor off
    }
  }
  else
  {
    count=count;
  }
}
```

এখানে মোটর ট্যাংকটিকে পানিপূর্ণ করলে যখন বন্ধ হয়ে যাবে, তখন মোটরটি আর পানি ভরবে না। এরপরে আবার যখন পানির উচ্চতা কমে যাবে, তখন নতুন করে শুরু হবে। মোটর বন্ধ হওয়ার পরে তখন সুইচ

বন্ধ করলে আরডুইনো বন্ধ হবে। তখন নতুন করে আবার চালু করে সবগুলো চলক নতুন করে মান পাবে।

সার্ভো কন্ট্রোল্ড রোবটিক আরম

সার্ভো মোটর সম্পর্কে আগেই আলোচনা করেছি। কীভাবে সার্ভো মোটর কাজ করাতে হয়, সেটিও আমরা জেনেছি। এখন আমরা জানব সার্ভো মোটরের সাহায্যে কীভাবে একটি রোবটিক হাত তৈরি করা যায়। আমরা খুব সাধারণ ভাবে তৈরি করব।

উপকরণ

১. সার্ভো মোটর ১টি ১৮০ ডিগ্রি ১টি ৩৬০ ডিগ্রি (১৮০ ডিগ্রিগুলোর টর্ক যাতে ২ কেজির বেশি হয় এটা মনে রাখতে হবে) সার্ভো হর্নসহ।

২. আরডুইনো মেগা ১টি

৩. অ্যালুমিনিয়াম পাত। (আমাদের অ্যালুমিনিয়াম পাত কাটতে হবে সিএনসি মেশিনের সাহায্যে)

৪. ১টি বেজ যেটার ওপরে হাতটি থাকবে আর ভেতরে থাকবে সব যন্ত্রপাতি

৫. পটেনশিওমিটার ৬টি (৬টি সার্ভো মোটর কন্ট্রোল করতে)

৬. ২টি প্লাস্টিকের গিয়ার

৭. পিভিসি অথবা প্লাস্টিক

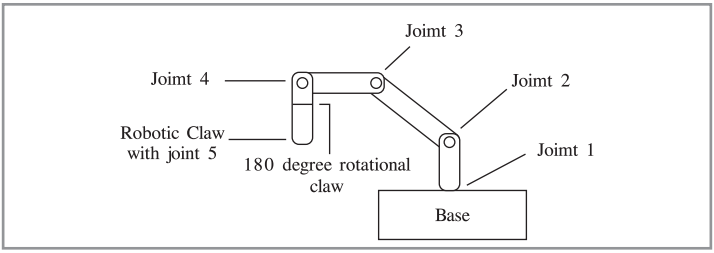
৮. তার ও জাম্পার

৯. ব্যাটারি (লাইপো ১৮০০ থেকে ৩৬০০ মিলি অ্যাম্পিয়ার ঘণ্টা যেকোনো)

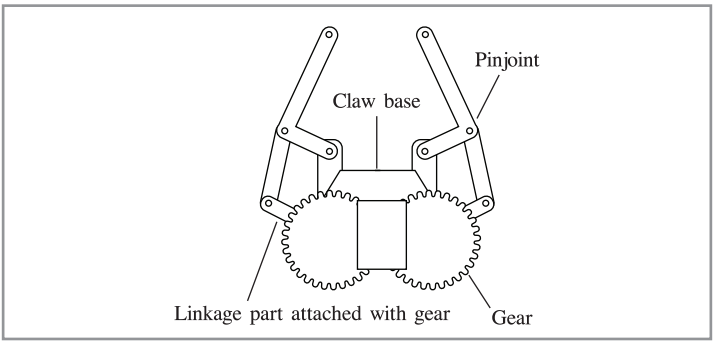
১০. স্ক্রু, নাট আর বোল্ট অনেকগুলো।

প্রথমে আমাদের একটি স্ট্রাকচার তৈরি করতে হবে। প্রাথমিকভাবে নিজেদের ডিজাইন তৈরি করে নিতে হবে। এরপরে ওই ডিজাইন অনুযায়ী অ্যালুমিনিয়াম পাত কাটা হবে। সবশেষে স্ক্রু দিয়ে জোড়া দিতে হবে। নাট-বোল্ট সিস্টেম ব্যবহার করা হবে এখানে।

আমাদের রোবটিক হাতের কাঠামো হবে নিচের ছবির মতোই।



চিত্র ৬.১৩ : রোবটিক হাতের আসল কাঠামো



চিত্র ৬.১৪ : হোল্ডারের কাঠামো

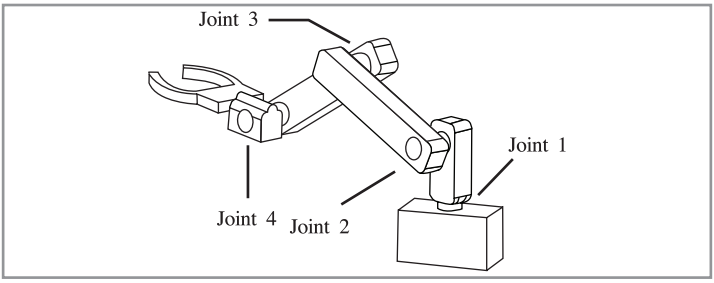
ওপরের ১ নম্বর ছবিতে দেখা যাচ্ছে রোবটিক হাতের আসল কাঠামো। আর ২ নম্বর ছবিতে দেখা যাচ্ছে হোল্ডার বা হাতের অংশটি, যেটি দিয়ে রোবটটি কোনো কিছু ধরবে।

১ নম্বর ছবিতে Joint 1 এটিতে ৩৬০ ডিগ্রি ঘুরতে পারে এমন একটি সার্ভো মোটর রাখলে ভালো হয়। ১৮০ ডিগ্রি হলেও হবে। তবে খেয়াল রাখতে হবে যাতে এটি ৬-৭ কেজি নেওয়ার মতো সামর্থ্য রাখে। কারণ এই মোটরটি আমাদের পুরো হাতের ওজন বহন করবে।

Joint 2, Joint 3, Joint 4 এই ৩টি জয়েন্টে সাধারণ সার্ভো মোটর ব্যবহার করতে হবে ১৮০ ডিগ্রি হলেই হবে। এখানে সর্বোচ্চ ২-৩ কেজি তুলার মতো সামর্থ্য থাকলেই চলবে। Joint 2, Joint 3, Joint 4 এই ৩টি জয়েন্ট একই সমতলে অবস্থান করবে। কারণ এখানে লিঙ্কেজগুলোর ঘূর্ণন একটি তল বরাবর।

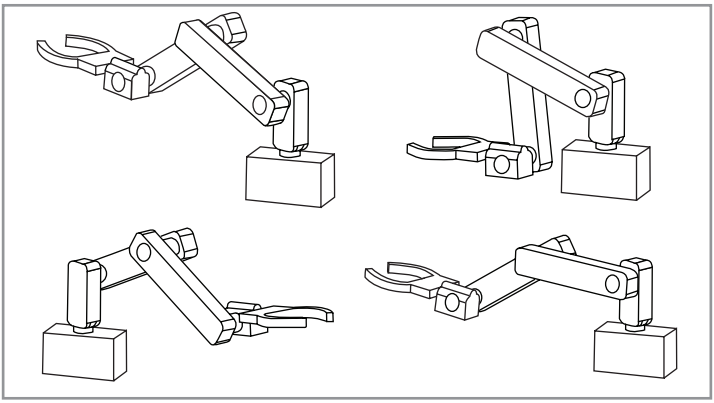
Joint 5 এটি রোবটিক হাতের হোল্ডারকে ধরে রাখে আর হোল্ডারকে ঘোরাবে।

এখানে থ্রিডি ছবি দেখলেই বোঝা যাবে।



চিত্র ৬.১৫ : রোবটিক হাত যেভাবে কাজ করে তার ত্রিমাত্রিক ছবি

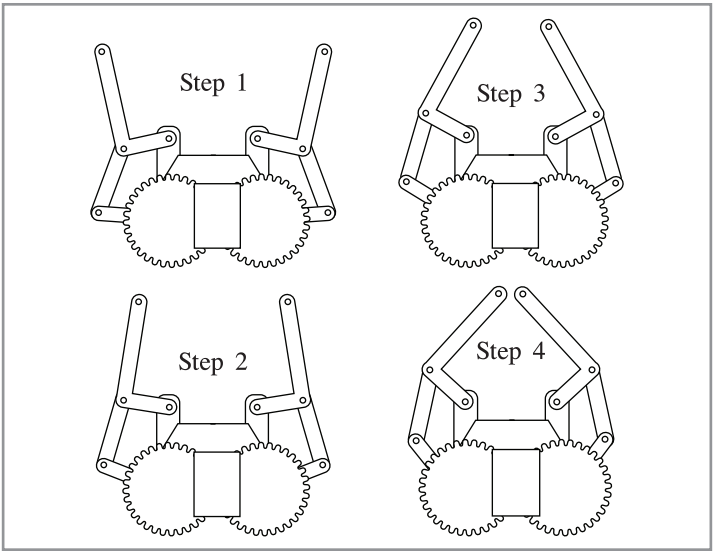
এখানে থ্রিডি ছবিতে বোঝা যাচ্ছে কীভাবে রোবটিক হাতটি কাজ করবে। এবং চিত্র ৬.১৬-এ আছে এর বিভিন্ন অংশের মুভমেন্ট।



চিত্র ৬.১৬ : হাতের বিভিন্ন অংশের মুভমেন্ট

এখানে ছবিগুলো দেখলে মুভমেন্টগুলো সহজে বোঝা যাবে।

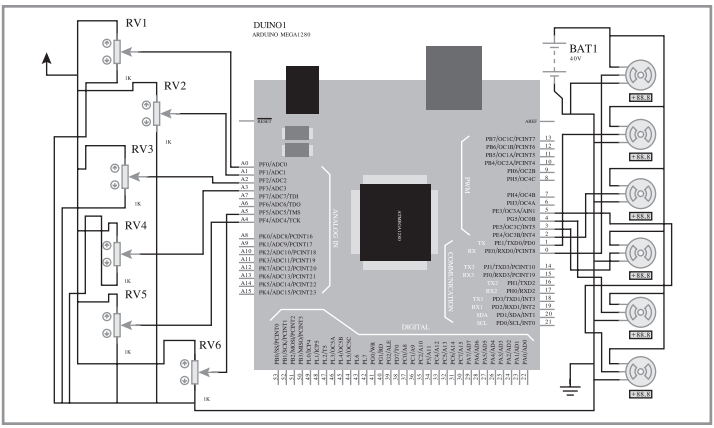
রোবটের হাতের হোল্ডার বা Claw যেটি, সেটির মুভমেন্ট সহজেই বোঝা যাবে। এখানে ছবিতে গিয়ারের সাথে যুক্ত লিঙ্কেজ গিয়ার ঘোরার সাথে সাথে ঘুরবে। আর একটি গিয়ার ঘোরার কারণে অপর গিয়ারটিও ঘুরবে ঠিক তার বিপরীতে। ফলে দুটি আঙুল একত্র হবে কোনো কিছু ধরার জন্য। চিত্র ৬.১৭ দ্রষ্টব্য।



চিত্র ৬.১৭ : রোবটিক হাতের মুভমেন্টের ৪টি ধাপ

এখানে ২টি গিয়ারের জন্য ১টি সার্ভো লাগবে।

এখন আমরা সার্কিট ডায়াগ্রাম দেখব (চিত্র ৬.১৮ দ্রষ্টব্য)।



চিত্র ৬.১৮ : সার্কিট ডায়াগ্রাম

সার্কিট ডায়াগ্রাম থেকে বোঝা যাচ্ছে যে এটি একটি জটিল সার্কিট। সহজভাবে বোঝানোর জন্য আমি একে একে বর্ণনা করছি।

এখানে মোট ৬টি সার্ভো রয়েছে। এর মধ্যে ১টি সবচেয়ে বড় যেটি পুরো হাতের ওজন বহন করবে। বাকিগুলো হাতের অন্যান্য অংশের সাথে সংযুক্ত থাকে।

মোট ৬টি সার্ভোর প্রতিটির ১টি হলুদ তার ১টি PWM পোর্টে যুক্ত হবে। PWM পোর্ট কোনটি সেটি আগেই বলেছি। কতগুলো পোর্টের পাশে টাইল্ড (~) চিহ্ন থাকে। সার্ভো মোটর ওখানেই সংযুক্তদ্ধ করতে হবে।

Hand joint (servo yellow wire)	Arduino Digital port number
Joint 1 servo	0
Joint 2 servo	1
Joint 3 servo	3
Joint 4 servo	5
Joint 5 servo	6
Joint 6 servo	9

ছক ৬.৩ : সার্ভো মোটরের তারের সাথে সংযোগ

সার্ভো মোটরের বাকি অংশের মানে লাল ও কালো তারগুলো যুক্ত করার জন্য ৬টি লাল তার একত্রে সংযুক্ত এবং ৬টি কালো তার একত্রে সংযুক্ত করতে হবে। তারপরে ৬টি লাল তারের সংযোগ প্রান্ত ব্যাটারির ধনাত্মক প্রান্তে এবং ৬টি কালো তার ব্যাটারির ঋণাত্মক প্রান্তে সংযোগ দিতে হবে। তারপর ওই ঋণাত্মক প্রান্ত থেকে আরেকটি সংযোগ আরডুইনোর গ্রাউন্ডে সংযোগ দিতে হবে।

সেন্সরের সংযোগ : যেহেতু এখানে পটেনশিওমিটার ব্যবহার করা হবে, সেহেতু ৬টি পটেনশিওমিটারের এক মাঝের পিনটি বাদে ২টি পিনের একটি ব্যাটারির ধনাত্মক প্রান্তে অপরগুলো ব্যাটারির ঋণাত্মক প্রান্তে সংযুক্ত হবে। মাঝের পিনটি থেকে সেন্সরের পাঠ নেওয়া হবে। ৬টি সেন্সরের সবগুলো পর্যায়ক্রমে অ্যানালগ পোর্ট ০, ১, ২, ৩, ৪, ৫-তে সংযোগ দিতে হবে।

আমরা এখানে ১ নম্বর সেন্সরের জন্য জয়েন্ট নম্বর ১ এভাবে প্রতিটি সেন্সরের জন্য ভিন্ন ভিন্ন জয়েন্টে যেমন ২ নম্বর সেন্সরের জন্য ২ নম্বর জয়েন্ট এ রকম।

এখন আমরা প্রোগ্রামিং কোডটি দেখব।

```
#include <Servo.h>

Servo j1, j2, j3, j4, j5, j6;
int s1, s2, s3, s4, s5, s6;

void setup ()
{
  Serial.begin (9600);
  j1.attach (0);
  j2.attach (1);
  j3.attach (2);
  j4.attach (3);
  j5.attach (4);
  j6.attach (5);
  pinMode (A0, INPUT);
  pinMode (A1, INPUT);
  pinMode (A2, INPUT);
  pinMode (A3, INPUT);
  pinMode (A4, INPUT);
  pinMode (A5, INPUT);
}
void loop ()
{
  s1=analogRead (A0);
  s2=analogRead (A1);
  s3=analogRead (A2);
  s4=analogRead (A3);
  s5=analogRead (A4);
  s6=analogRead (A5);
  //here we will divide sensor reading into some equal parts to match with servo
  s1 = map(s1, 0, 1023, 0, 360);
  j1.write (s1);
  s2 = map(s2, 0, 1023, 0, 180);
  j2.write (s2);
  s3 = map(s3, 0, 1023, 0, 180);
  j3.write (s3);
  s4 = map(s4, 0, 1023, 0, 180);
  j4.write (s4);
  s5 = map(s5, 0, 1023, 0, 180);
  j5.write (s5);
  s6 = map(s6, 0, 1023, 0, 180);
  j6.write (s6);
}
```

প্রোগ্রামটি আপলোড দিয়ে রান করলেই দেখা যাবে। পটেনশিওমিটারের নব ঘোরালে তখন প্রতিটা জয়েন্টের মুভমেন্ট দেখা যাবে ভালোমতো।

রোবটিক হাতের কাজ আপাতত এতটুকুই।

এখানে পরবর্তী সময়ে আলোচনা করা হবে বহুল পরিচিত লাইন ফলোয়ার রোবট নিয়ে

লাইন ফলোয়ার রোবট

লাইন ফলোয়ার রোবট অনেক বহুল প্রচলিত রোবট। একটি পরিপূর্ণ রোবট কীভাবে চলে সেটির অন্যতম সুন্দর উদাহরণ এটি। লাইন ফলোয়ার রোবট অনেকটাই ছোটখাটো কৃত্রিম বুদ্ধিমত্তার সাহায্যে চলে। সেন্সর থেকে পাঠ নিয়ে সে অনুযায়ী পথ তৈরি করে। একটি রোবটের একদম প্রাথমিক এবং সব থেকে সহজ উদাহরণ এটি।

আমরা শুরু করি তাহলে।

এখানে আমরা প্রথমে সেন্সর সম্পর্কে আলোচনা করব। তারপরে পুরা রোবট সম্পর্কে সামগ্রিক আলোচনা করা হবে।

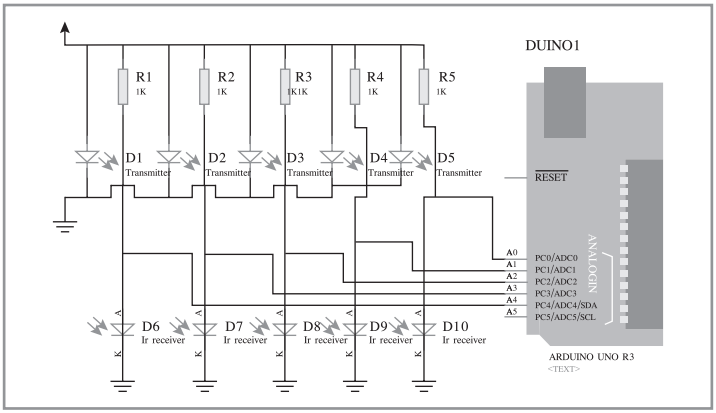
সেন্সর

এখানে আমরা কোনো স্পেসিফিক সেন্সর ব্যবহার করব না। এখানে সেন্সরের একটি অ্যারে ব্যবহার করা হবে। সর্বনিম্ন ২টি থেকে ৬/৭/৮টি অ্যারের সেন্সরও ব্যবহার করা যায়।

এখানে একটি আই.আর ট্রান্সমিটার এবং আই.আর রিসিভার ২টি পাশাপাশি লাগানো থাকে। আমরা আগেই দেখেছি আই.আর সেন্সর সম্পর্কে। এখন লাইন ফলোয়ার রোবটে কীভাবে কাজ করে এটি?

লাইন ফলোয়ার রোবট একটি কালো লাইনকে অনুসরণ করে চলে। এখানে আই.আর সেন্সরের ট্রান্সমিটার থেকে আলোর রশ্মি কালো লাইনে পড়ে। এখন কালোবস্তু যেকোনো রঙের আলো শোষণ করে নেয়। ফলে আই.আর সেন্সরের ট্রান্সমিটার থেকে যে রশ্মি পড়ল কালো লাইনে, সেটি খুব কম পরিমাণে প্রতিফলিত হয়। অন্যদিকে সাদা লাইনের ওপরে পড়লে প্রতিফলনের পরিমাণ বেশি। লাইন ফলোয়ার রোবটের সেন্সর পাওয়া যায়। আবার চাইলে নিজে থেকেও তৈরি করা যায়।

চিত্র ৬.১৯-এ সার্কিট ডায়াগ্রামটি দ্রষ্টব্য।



চিত্র ৬.১৯ : লাইন ফলোয়ার রোবটে ব্যবহার করা সেন্সরের অ্যারে

এখানে আই.আর ট্রান্সমিটার থেকে অবলোহিত রশ্মি বের হবে এবং রিসিভার সেটি গ্রহণ করবে। কালো ও সাদা অংশের জন্য ২টি ক্ষেত্রে ভিন্ন ভিন্ন পাঠ পাওয়া যাবে। এই ২টি ভিন্ন পাঠ থেকে লজিক নির্ধারণ করে রোবটটি তার পথ নির্ণয় করে।

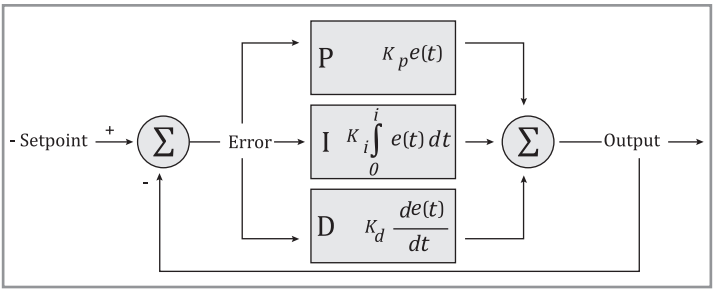
এখানে ওপরে আমি যেভাবে সার্কিট ডায়াগ্রাম দেখালাম, সেভাবেই তৈরি করতে হবে।

এখানে যে রোধগুলো দেখা যাচ্ছে, সেগুলো চাইলে পরিবর্তন করা যায়। ভালো মানের পাঠ পাওয়ার জন্য চাইলেই রোধ পরিবর্তন করে দেখতে হবে। আর অ্যানালগ আউটপুট ব্যবহার করতে হবে। আমরা আগেই দেখেছি অ্যানালগ আউটপুট কী রকম। ওভাবেই করতে হবে।

মোটর

এখানে আমরা ২ চাকার আর ১ বল ক্লাস্টারের সমন্বয়ে রোবটটি তৈরি করব। সে জন্য ২টি মোটর এবং ১টি মোটর ড্রাইভারের সমন্বয়ে রোবটটি তৈরি করব। এখানে ২টি অংশের মধ্যে ১টি সেন্সরের অংশ এবং অপরটি মোটরের অংশ। মাঝে লজিক নিয়ে কাজ করবে আরডুইনো। মোটর ড্রাইভার সম্পর্কে আগেই আলোচনা করেছি। এ জন্য এ সম্পর্কে এখানে আলোচনা করব না।

আগে মোটর ড্রাইভার ও সেন্সরের বিস্তারিত প্রোগ্রাম নিয়ে আলোচনা করা হয়েছে। কাজেই এ নিয়ে এখানে আর আলোচনা করা হলো না। এখানে পিআইডি কন্ট্রোলার সিস্টেম ব্যবহার করা হয়েছে। পিআইডির পূর্ণরূপ হচ্ছে Proportional Integral Derivative। নিচের ছবিতে এটি সম্পর্কে সাধারণ ধারণা দেওয়া হয়েছে। এই প্রোগ্রামটিতে মূলত পিআইডির একটি সাধারণ অ্যালগরিদম ব্যবহার করা হয়েছে।



চিত্র ৬.২০ : পিআইডির ধারণা

নিচে মূল প্রোগ্রামটি লেখা হয়েছে, আশা করা যায় আগের ধারণা থাকায় সবাই এটি সহজে বুঝতে পারবে। আমরা সব সময় চিন্তা করার শক্তিকে উৎসাহিত করি এবং সবাই পুরো প্রোগ্রাম চিন্তা করে এটা থেকে সত্যিকারের আনন্দ পাবে বলে আশা করা যায়।

```
#define pinA 2
#define pinB 3
#define pinC 4
#define pinD 7
#define ena 5
#define enb 6

int threshold[5]={512, 512, 512, 512, 512}; // threshold value
int led[5]={8, 9, 10, 11, 12};

int i, sen[5], s[5], lastSensor, lastError;

int base_L=180;
int base_R=180;
int kp=12;
int kd=10;
```

```
void setup()
{
  mot_init(); // motor function (with this function the motor will work)
  other_init();
}

void loop()
{
  Serial.println(readSensor()); //sensor reading function
  //wheel(200, 200);
  line_follow(); //main function of line following
}

void other_init()
{
  for(i=0;i<5;i++) pinMode(led[i], OUTPUT);
  lastSensor=0;
  lastError=0;
  Serial.begin(9600);
}

void mot_init()
{
  pinMode(pinA, OUTPUT);
  pinMode(pinB, OUTPUT);
  pinMode(pinC, OUTPUT);
  pinMode(pinD, OUTPUT);
  pinMode(ena, OUTPUT);
  pinMode(enb, OUTPUT);

  digitalWrite(pinA, HIGH);
  digitalWrite(pinB, HIGH);
  digitalWrite(pinC, HIGH);
  digitalWrite(pinD, HIGH);
}

///leftmotor leftmotor speed, rightmotor rightmotor speed
void wheel(int leftmotor, int rightmotor)
{
 if(leftmotor==0)
  {
```

```
 digitalWrite(pinC, HIGH);
 digitalWrite(pinD, HIGH);
}

else if(leftmotor>0)
{
  digitalWrite(pinC, HIGH);
 digitalWrite(pinD, LOW);
}

else if(leftmotor<0)
{
  digitalWrite(pinC, LOW);
  digitalWrite(pinD, HIGH);
}

if(rightmotor==0)
{
  digitalWrite(pinA, HIGH);
 digitalWrite(pinB, HIGH);
}

else if(rightmotor>0)
{
  digitalWrite(pinA, HIGH);
  digitalWrite(pinB, LOW);
}

else if(rightmotor<0)
{
  digitalWrite(pinA, LOW);
  digitalWrite(pinB, HIGH);
}

if(leftmotor>254) leftmotor=254;
if(leftmotor<-254) leftmotor=-254;
if(rightmotor>254) rightmotor=254;
if(rightmotor<-254) rightmotor=-254;

analogWrite(ena, abs(rightmotor));
analogWrite(enb, abs(leftmotor));
```

```
}

int readSensor()
{
  for(i=0;i<5;i++)
  {
    sen[i]=analogRead(i);     //taking reading from sensors
    Serial.print(sen[i]);
    Serial.print("  ");
  }
  Serial.print("              ");
  for(i=0;i<5;i++)
  {
    if(sen[i]<threshold[i])
    {
     s[i]=1;
      digitalWrite(led[i], HIGH);
   }
    else
    {
      s[i]=0;
      digitalWrite(led[i], LOW);
    }                    //if black then set 1 else 0
  }
  int error, sum;
  sum=s[0]+s[1]+s[2]+s[3]+s[4];
  if(sum!=0) error=(s[0]*10+s[1]*20+s[2]*30+s[3]*40+s[4]*50)/sum-30;
  else error=420;  // line lost
if(s[0]==1) lastSensor=1;
else if(s[4]==1) lastSensor=2;
 return error;
}
void line_follow()
{
  int error, p, d, corr;
  error=readSensor();
  if(error==420)
 {
   if(lastSensor==1) wheel(-100, 100);
    else if(lastSensor==2) wheel(100, -100);
 }
```

```
else
  {
    p=kp*error;
    d=kd*(error-lastError);
    corr=p+d;
    wheel(base_L+corr, base_R-corr);
    lastError=error;
  }
}
```

পরিশিষ্ট

প্রয়োজনীয় উপকরণ

১. Arduino UNO, Arduino Mega, Arduino Nano

২. রেসিস্টর (১০০টি রিল আকারে পাওয়া যায়, বিশেষত ২২০, ৫০০ ও ১০০০ ওহম লাগবেই)

৩. ট্রানজিস্টর, মসফেট

৪. ক্যাপাসিটর (প্রজেক্টের ওপর নির্ভরশীল)

৫. পটেনশিওমিটার (১-১০০০ ওহম, ১-১০০০০ ওহম) (প্রজেক্টভেদে ৫-৬টা লাগাতে হবে)

৬. অপারেশনাল অ্যামপ্লিফায়ার

৭. জাম্পার (মেইল টু মেইল, মেইল টু ফিমেইল, ফিমেইল টু ফিমেইল)

৮. L298N, L293D Driver IC (মোটর ড্রাইভার, আইসিসহ প্যাকেজ)

৯. ডায়োড

১০. LED (10-15টি)

১১. ব্রেডবোর্ড

১২. সুইচ (পুশ) (SPST, SPST, DPDT)

১৩. বিভিন্ন রকম সেন্সর (প্রজেক্টভেদে বিভিন্ন উল্লেখ করা হয়েছে)

১৪. পাওয়ার সাপ্লাই (প্রজেক্টভেদে উল্লেখ করা হয়েছে)

১৫. ছোট প্রজেক্ট লাইন ফলোয়ার রোবট, মেজ সল্ভার (ছোটখাটো ২ অথবা ৪ চাকার রোবটের জন্য ৩ সেল ১২০০-১৫০০ mAh Lipo battery-এ যথেষ্ট। প্রজেক্টের সাইজ যদি বড় হয় তাহলে লেড অ্যাসিড ব্যাটারি লাগবে। ঢাকার পাটুয়াটুলীতে ৬ ভোল্ট এবং ১২ ভোল্টের লেড অ্যাসিড ব্যাটারি পাওয়া যায়। এগুলো অনেক হাই পাওয়ার এবং হাই mAh-এর হয়ে থাকে

১৬. সল্ডারিংকিট

১৭. ভেড়োবোর্ড

১৮. 3D Printing / শিট মেটালের কাজ / প্লাস্টিক প্রসেস system / PVC / অ্যালুমিনিয়াম / acrylic / CLC মেশিনের যেকোনো ২টি দিয়ে কাজ করা জানতে হবে। তবে PVC আর acrylic দিয়ে কাজ জানলে সবচেয়ে ভালো হয়।

শেষের শুরু

আমরা বইয়ের শেষ অধ্যায়ে চলে এসেছি। আগের অধ্যায়গুলো ঠিকমতো পড়ে থাকলে, নিজে নিজে সার্কিট, ব্রেডবোর্ড নিয়ে অনেক ঘাঁটাঘাঁটি করে বা কম্পিউটারে কোডগুলো চালিয়ে দেখতে গিয়ে ছোট ভুলের জন্যও যদি নির্ঘুম রাত পার করে থাকো, তাহলে তোমাকে অভিনন্দন। তুমি একটা নতুন জগতে ঢুকে পড়েছ। সত্যি বলতে গেলে ঢোকার জন্য প্রস্তুত হয়েছ।

এখানে আমরা চাইলে অনেক লিঙ্ক কিংবা আরও অনেক কিছুর একটা লিস্ট দিতে পারতাম। কিন্তু ইচ্ছা করেই করছি না। কেন? কারণ আমরা চাই তোমরা নিজেরা সেগুলো খুঁজে বের করো। এসব জায়গায়, এমনকি যেকোনো আগ্রহের জায়গায় আসলে খুঁজে বের করা শিখতে পারাটা অনেক জরুরি। কারণ এসব জায়গায় চর্চা আর শ্রমের বিকল্প নেই। এখানে সবচেয়ে বড় যে দক্ষতা অর্জন করতে হয় তা হলো, কোথাও আটকে গেলেই ইন্টারনেট থেকে তথ্য খুঁজে বের করা। আজকাল মানুষের মাঝে অস্থিরতা অনেক বেশি দেখা যায়। খুঁজে না ঘেঁটে অন্যকে জিগ্যেস করে সহজে কিছু পেয়ে যাওয়ার প্রবণতা দেখা যায়। কিন্তু আমাদের অভিজ্ঞতা বলে ওভাবে তুমি হয়ত যা চাচ্ছো তা পেয়ে যাবে সত্যি, কিন্তু অনেক ধাক্কা খেয়ে যেটা শিখবে, সেটা অনেক বেশি দীর্ঘস্থায়ী হবে। তোমার চলার পথ যদি মসৃণ হয় সব সময় তাহলে তুমি সংগ্রামটা বুঝতে পারবে না। শেষ পর্যন্ত লেগে থাকাটাই সবচেয়ে বড় কথা। রোবটিকসে কোনো কিছু সমাধানের জন্য আসলে গাণিতিক বা প্রোগ্রামিংয়ে দক্ষতার বাইরেও অনেক ভালো পর্যবেক্ষণের দরকার হয়। এগুলো আসলে অনেক চর্চার বিষয়।

কাজেই এই বইয়ের শুরু হয়েছিল তোমাদের স্বাগত জানিয়ে, শেষেও থাকবে একই কথা, তোমাদের নতুন এমন রঙিন এক জগতে স্বাগত, যার শুরু আছে কিন্তু শেষ নেই!!