

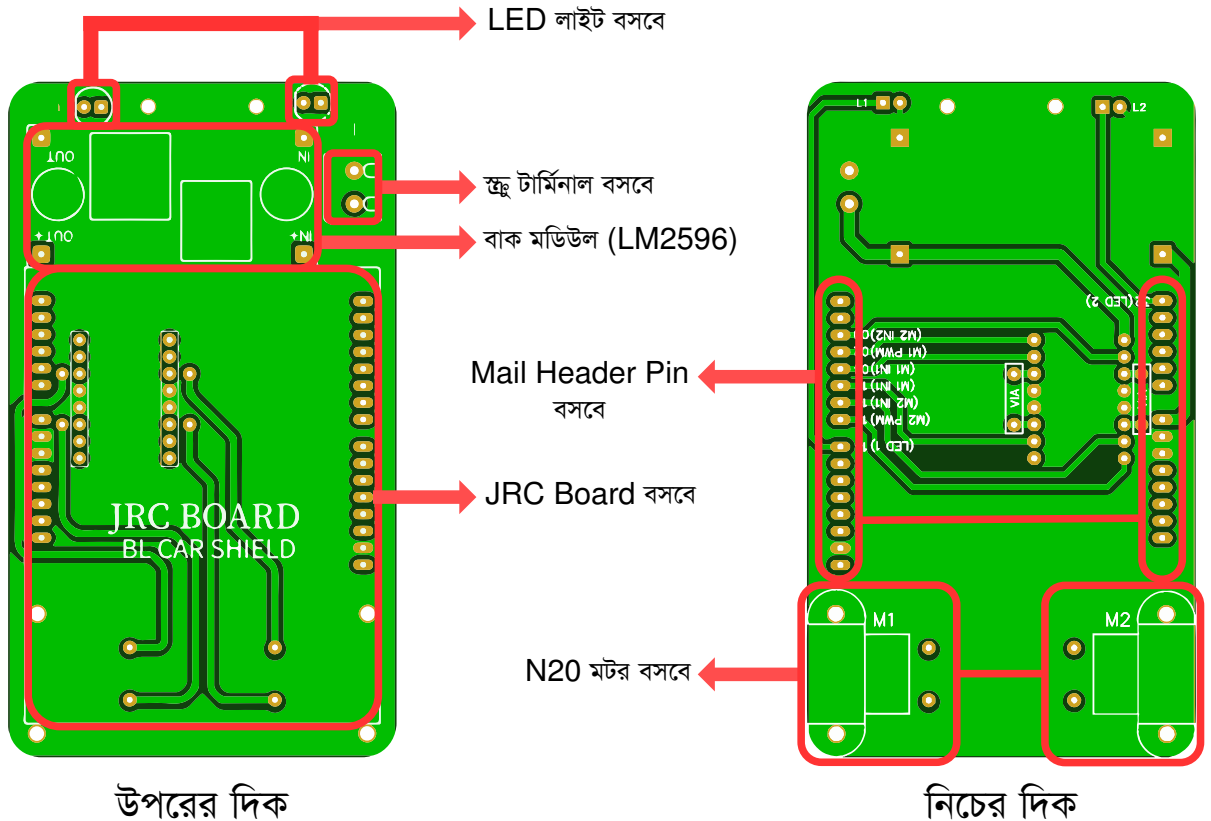
প্রজেক্ট ১৪ঃ ব্লুটুথ কন্ট্রোল কার

উপকরনঃ

- ১/ ব্লুটুথ কার শিল্ড
- ২/ JRC Board
- ৩/ বাক মডিউল
- ৪/ স্ক্রু টার্মিনাল
- ৫/ লিপো ব্যাটারি
- ৬/ লিপো ব্যাটারি কানেক্টর
- ৭/ এলইডি
- ৮/ মেইল হেডার পিন
- ৯/ N20 মটর
- ১০/ মটর মাউন্ট

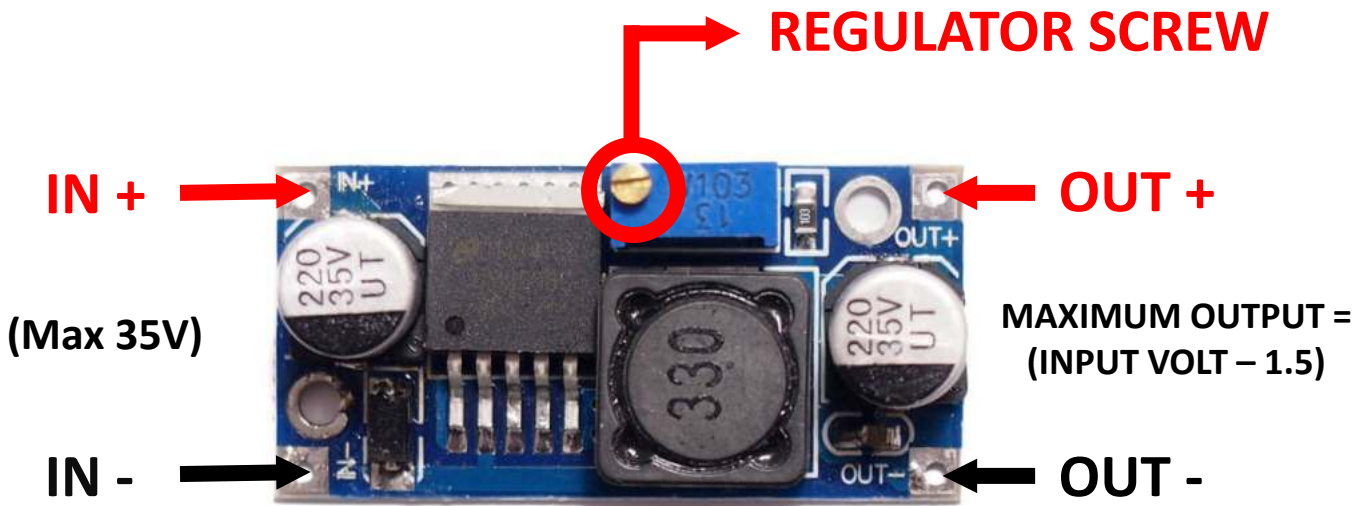
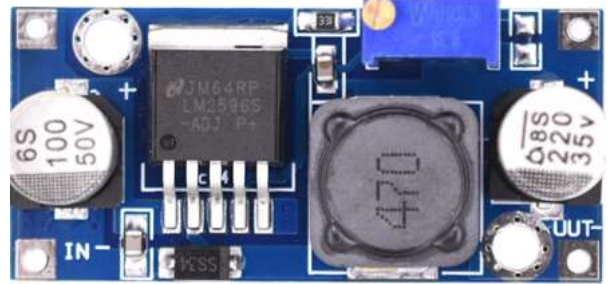
বর্ণনাঃ

আমরা ইতিমধ্যেই কিভাবে ব্লুটুথের মাধ্যমে যোগাযোগ করতে হয় সেটা শিখে গেছি তাইনা? এখন সেই প্রযুক্তি ব্যবহার করেই একটা রেডিমেড পিসিবির উপরে বিভিন্ন কম্পোনেন্ট বসিয়ে একটি ব্লুটুথ কন্ট্রোল কার তৈরী করবো যেটি মোবাইল থেকে যেকোন কমান্ড নিয়ে চলতে পারবে। প্রথমেই আমরা পিসিবিটি দেখে নিইঃ

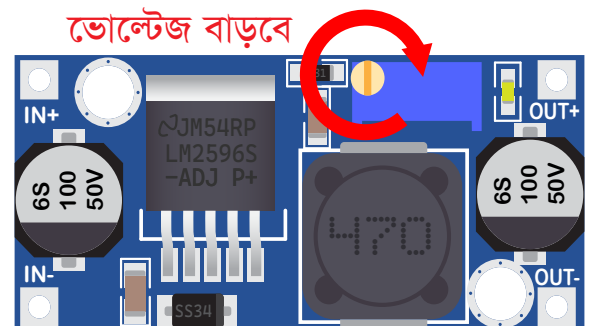
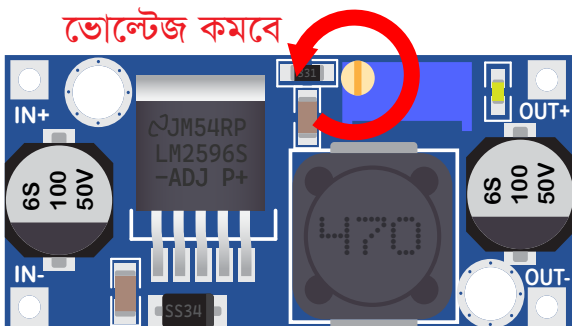


এখানে পিসিবি বোর্ডের বিভিন্ন অংশ দেখানো হয়েছে। যদি বেসিক মেকানিজম কিংবা সার্কিট সম্পর্কে বলতে হয়, তাইলে সেটা আসলে খুবই সিম্পল। আমরা ইতিমধ্যেই জেনেছি যে JRC Board এর VIN পিনে ৫ থেকে ১৬ ভোল্ট পর্যন্ত দেয়া যায় তাইনা? সেখানে সত্যিকার অর্থে ৫-৬ ভোল্ট দেয়া সবথেকে নিরাপদ এবং আমরা যখন ২-৩ সেল এর লিপো ব্যাটারি ব্যবহার করবো, তখন সেই ৮-১২ ভোল্ট কে ৫ ভোল্টে নামিয়ে আনতে বাক কনভার্টার সবথেকে ভালো কাজ করে থাকে।

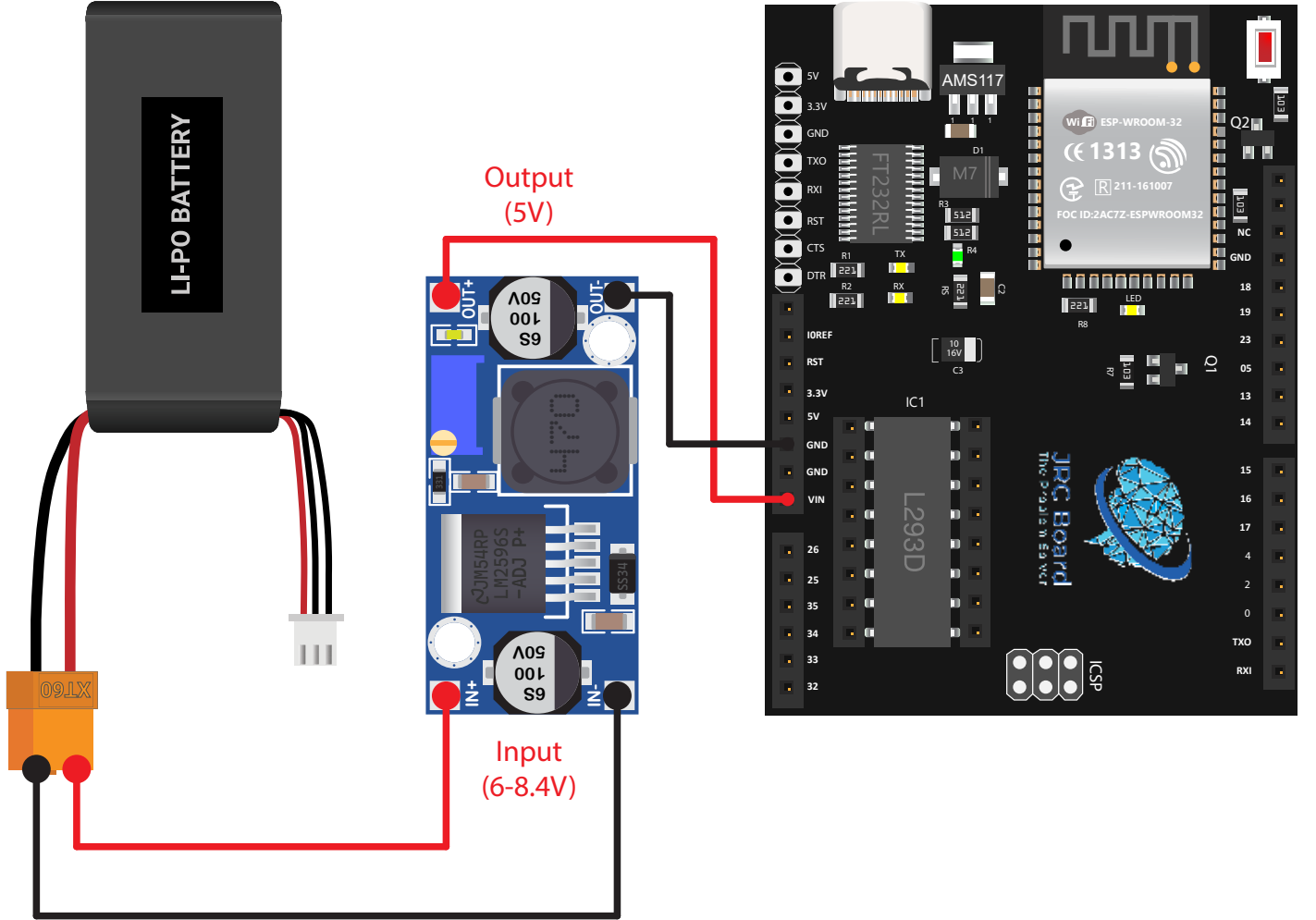
LM2596 হচ্ছে একটি DC-DC ভোল্টেজ কনভার্টার। সোজা কথায় বলতে এটা ব্যাটারির উচ্চ ভোল্টেজ কে ব্যবহারযোগ্য নিম্ন ভোল্টেজে নামিয়ে আনতে পারে। এটা ব্যবহার খুবই সহজ এবং দামেও সস্তা। চলো এর পিন-আউট দেখে নিইঃ



এখানে আমরা দেখতে পাচ্ছি দুটি ইনপুট টার্মিনাল (বামে) এবং দুটি আউটপুট টার্মিনাল (ডানে) রয়েছে। এদের নামগুলো মডিউলের গায়েই চিহ্নিত করে দেয়া থাকে। বলা বাহুল্য IN+ এবং IN- টার্মিনাল এ ব্যাটারির সংযোগ দেয়া হয় এবং OUT+ এবং OUT- টার্মিনাল থেকে আমরা দরকারি ভোল্টেজ টা পাই। এখন আমাদের যেই ভোল্টেজ টা দরকার সেই ভোল্টেজ টা পেতে উপরে একটি নীল রঙের রেগুলেটর থাকে যেটাকে ঘড়ির কাটার উলটো দিকে ঘুরালে ভোল্টেজ কমে এবং ঘড়ির কাটার দিকে ঘুরালে ভোল্টেজ বাড়তে থাকে।

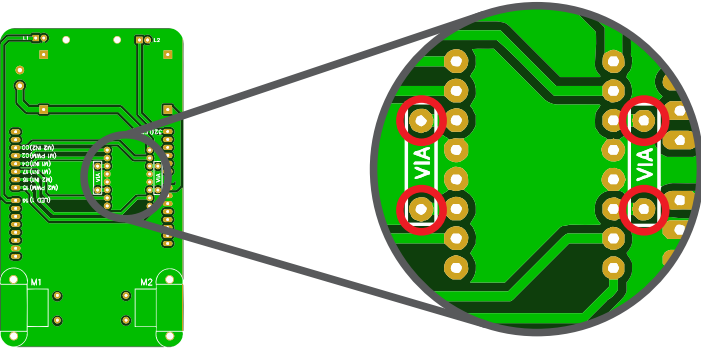


আশা করছি সহজেই এই বাক কনভার্টার এর মেকানিজম টা বুঝে ফেলেছো। সোজা কথায় বললে প্রথমে ব্যাটারির সাথে বাক কনভার্টার এর ইনপুট টার্মিনাল (IN+ এবং IN-) এর সংযোগ দিবে। এরপর রেগুলেটর স্ক্রু টা ঘুরিয়ে তোমার দরকারমতো ভোল্টেজে নামিয়ে আনবে (আপাতত ৫ ভোল্ট), এরপর আউটপুট প্রান্তের (OUT+ এবং OUT-) সাথে যেকোন ডিভাইস (আপাতত JRC Board) এর সংযোগ দিবে। নিচের চিত্রটি দেখলে পাওয়ার সার্কিট সম্পর্কে ধারণা হয়ে যাবেঃ

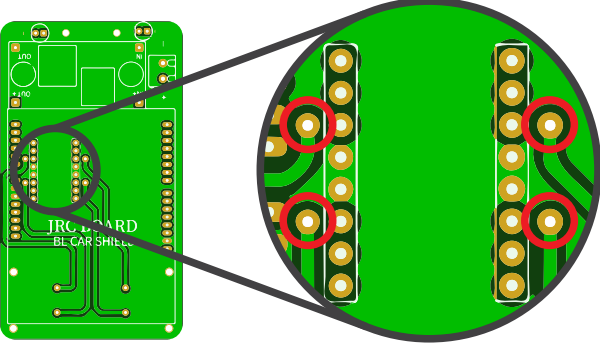


ঠিক এই সার্কিটটাই পিসিবিতে তৈরী করে রাখা আছে যেখানে কেবল তোমাদের বাক মডিউল এর সাথে Mail Header Pin ঝালাই করে জায়গামতো বসিয়ে দিতে হবে। উল্লেখ্য যে বাক মডিউল টা কোন দিকে বসাতে হবে বা কোথায় কোন পিন থাকা উচিত সেটি পিসিবির গায়েই লিখে দেয়া আছে। এবং বাক মডিউলের ভোল্টেজ এডজাস্ট করার আগে কোনভাবেই এর আউটপুটে কোন ডিভাইস যুক্ত করবেনা! কেননা ভোল্টেজ এডজাস্ট না করে কোন ৫ ভোল্টের ডিভাইস যুক্ত করলে সেক্ষেত্রে ব্যাটারী যুক্ত করার সাথে সাথেই বাক মডিউল আউটপুটে ব্যাটারি ভোল্টেজ পাঠিয়ে দেয় (যেহেতু রেগুলেটর স্ক্রু টা ঘুরিয়ে ভোল্টেজ নামানো হয়নি)। এক্ষেত্রে ৫ ভোল্টের জায়গায় সেই ডিভাইসটি ৭-৮ ভোল্ট বা তারও বেশি ভোল্টেজ পেতে পারে এবং সেটি ডিভাইসটিকে নষ্ট করে দিতে পারে!

এভাবে আমরা বাক মডিউলের কাজ শিখে গেলাম এবং পিসিবিতে কেন এটি ব্যবহার হচ্ছে সেটিও বুঝে গেলাম। এবার তাহলে ছবির মত প্রসেস অবলম্বন করিঃ

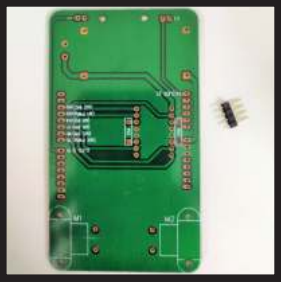


চিহ্নিত পিনগুলোতে একটি করে
mail header pin ঝালাই করি

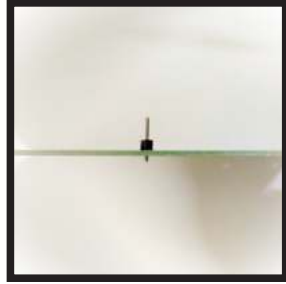


নিচ থেকে যে প্যাডগুলোতে মেইল হেডার পিন ঝালাই
করা হয়েছিলো, উপরের দিক হতেও একই দিকে মেইল
হেডার পিন এর প্লাস্টিক অংশ খুলে ফেলে দিয়ে এরপর
পিনের সাথে সোল্ডার প্যাডের ঝালাই দিই।

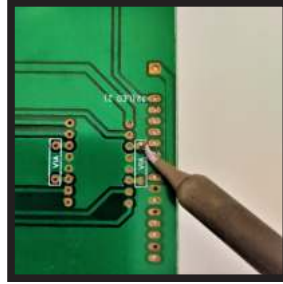
এই স্টেপগুলো সম্পন্ন করলে এটি নিচের ছবির মতো অবস্থা দেখাবেঃ



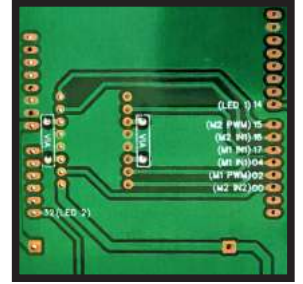
১) বোর্ড এবং হেডার পিন



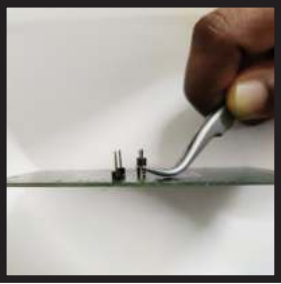
২) হেডার পিন বসানোর ধরণ



৩) নিচ থেকে পিন
সোল্ডার করা



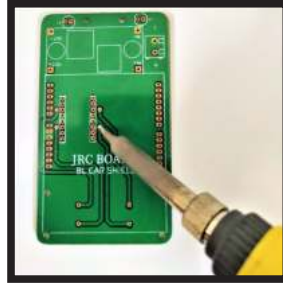
৪) সবগুলো হেডার পিন
এভাবে সোল্ডার করা



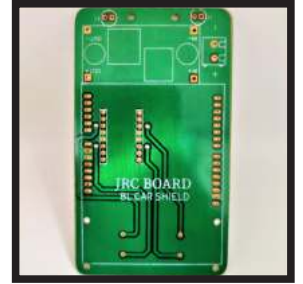
৫) উপর থেকে প্লাস্টিক
হোল্ডার টি খুলে ফেলা



৬) পিনের বাড়তি অংশ
কেটে ফেলা

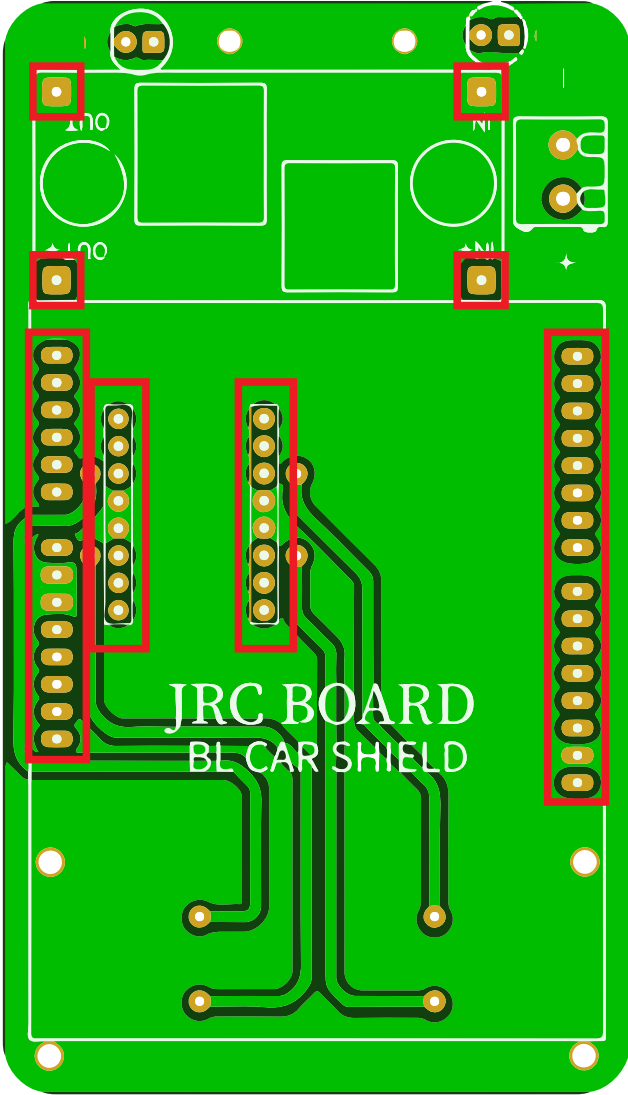


৭) উপর থেকেও একে একে
প্যাডের সাথে পিন ঝালাই করা

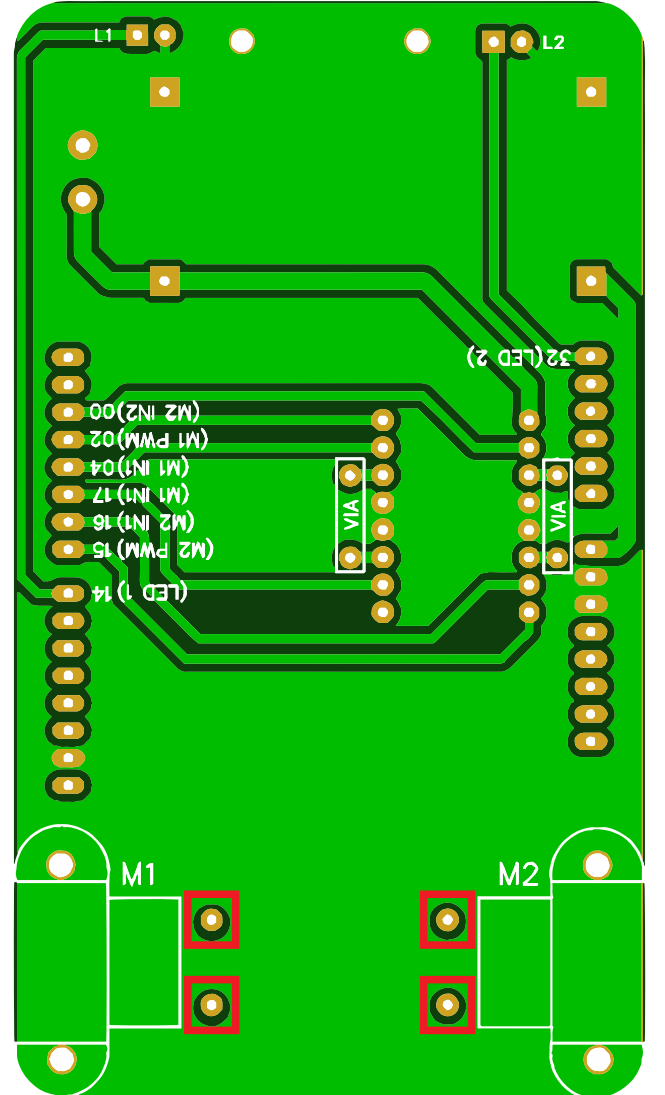


৮) ৪টি প্যাড উপর এবং নিচ থেকে
সোল্ডারের পরে প্রথম কাজ শেষ

এভাবে চাইলে বাকি প্যাডগুলোতেও হেডার পিন বসিয়ে দিতে পারি, তবে পার্থক্য হলো এখানে আর পিনের প্লাস্টিক অংশ খুলতে হবেনা এবং কোন কিছু কাটতে হবেনা, কেবলমাত্র বসিয়ে দিলেই হবে। কোন দিক থেকে পিন বসাতে হবে তার একটি ম্যাপ নিচে দেখানো হচ্ছেঃ

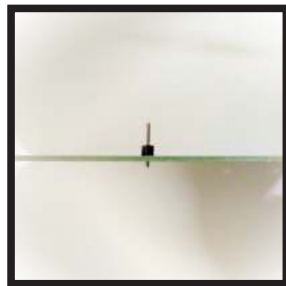


উপর থেকে যেসকল জায়গায় মেইল হেডার পিন বসিয়ে নিচ দিয়ে সোল্ডার করতে হবে (লাল চিহ্নিত অংশ)

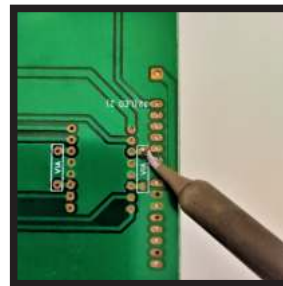


নিচ থেকে যেসকল জায়গায় মেইল হেডার পিন বসিয়ে উপর দিয়ে সোল্ডার করতে হবে (লাল চিহ্নিত অংশ)

এক্ষেত্রে স্টেপ হবে কেবল দুটিঃ

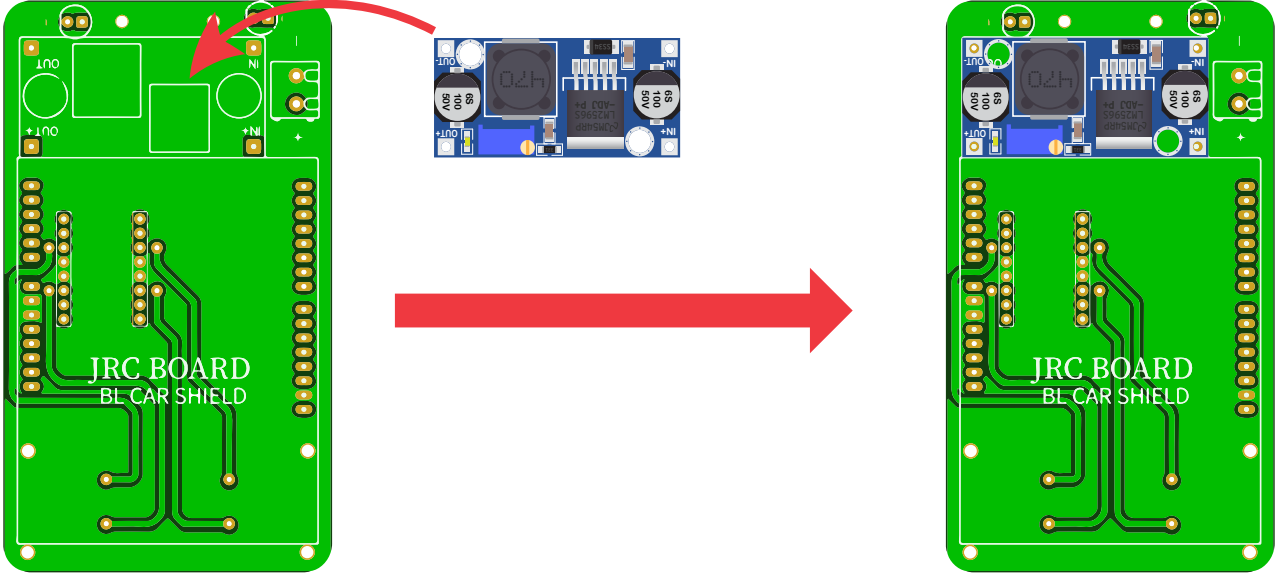


১) হেডার পিন বসানোর ধরণ

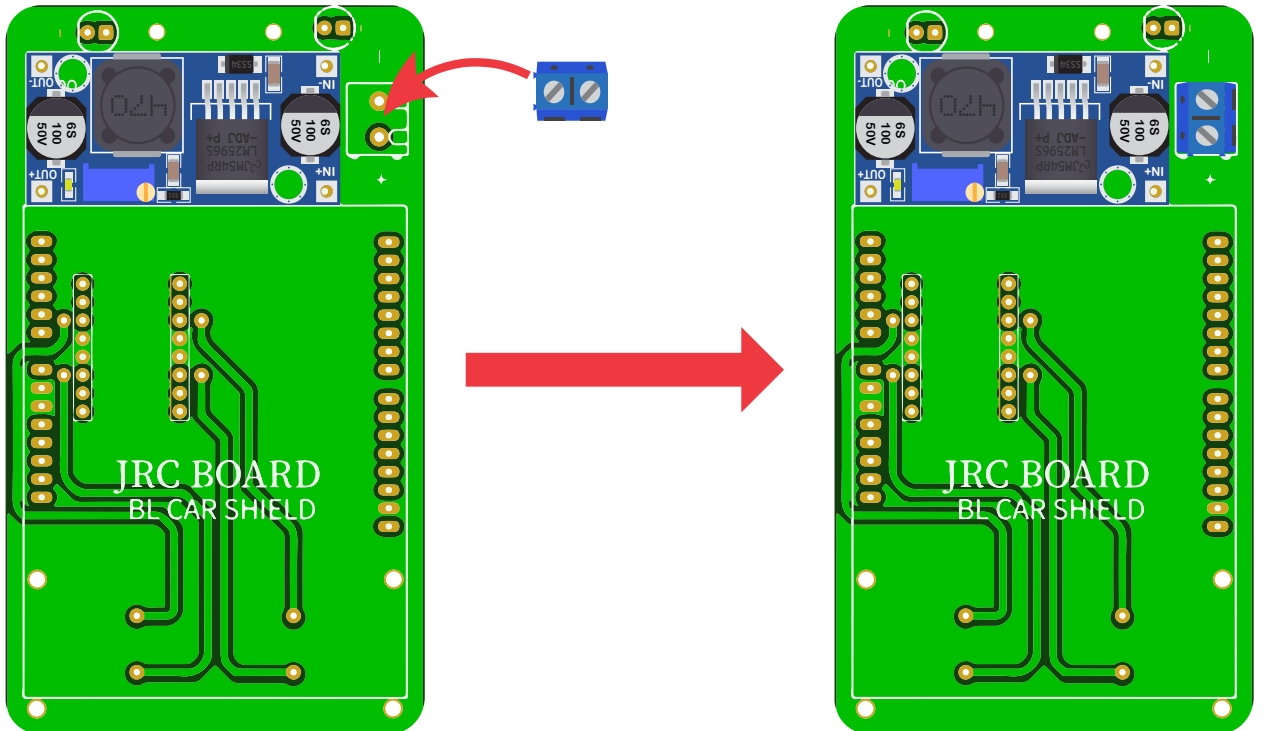
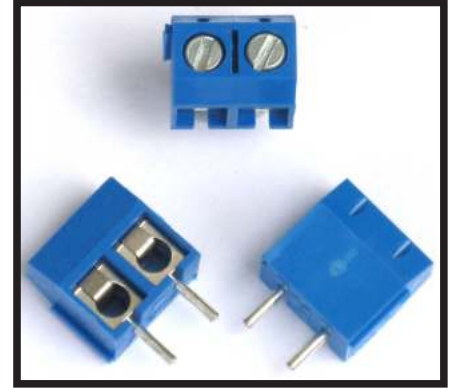


২) পিন সোল্ডার করা

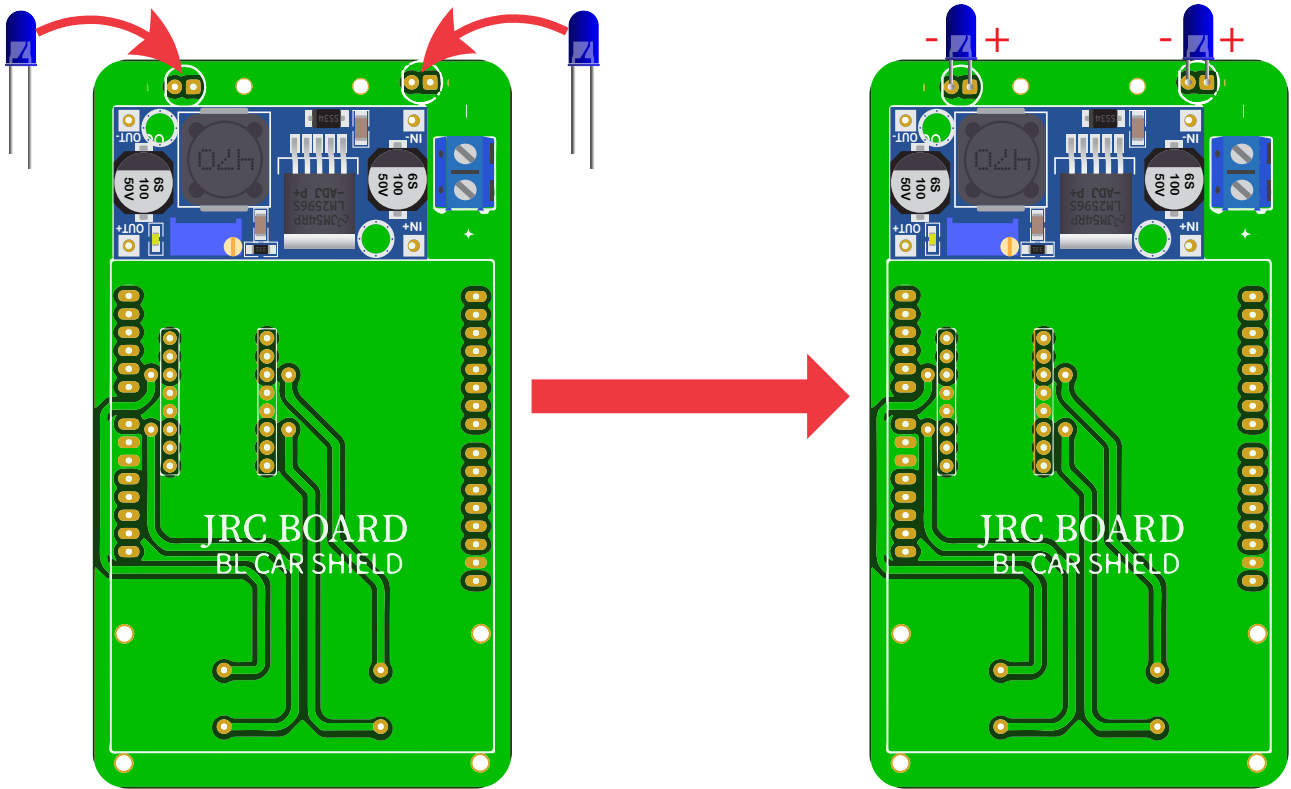
এভাবে সবগুলো মেইল হেডার পিনগুলো বসিয়ে সোল্ডার করার পরে এরপর বাক মডিউল টি পিসিবির উপরের দিক থেকে জায়গা বরাবর মেইল হেডার পিনের ভেতরে ঢুকিয়ে এরপর সোল্ডার করে স্থাপন করা যেতে পারে।



এবার বাক মডিউলের ঠিক পাশেই স্ক্রু টার্মিনাল বসিয়ে ঝালাই করতে হবে। স্ক্রু টার্মিনাল হলো এমন এক ধরনের বিশেষ পিন যা বোর্ডের সাথে সোল্ডার করার মাধ্যমে যুক্ত হয় এবং উপরে একটি ছিদ্র এবং তার উপরেই স্ক্রু থাকে। ছিদ্রে তার ঢুকিয়ে এরপর উপর থেকে স্ক্রু আটকে দিলেই বোর্ডের নির্দিষ্ট প্যাডের সাথে ঐ তারের সংযোগ স্থাপন হয়ে যায়। এক্ষেত্রে সুবিধা হলো চাইলে সহজেই তার পাল্টানো যায় এবং তারের সাথে বোর্ডের কোনরূপ ঝালাই করত হয়না।



এরপর আমরা চাইলে বোর্ডের সামনের অংশে দুইটি এলইডি লাইট স্থাপন করে এরপর সেগুলো সোল্ডার করে ফেলি। লক্ষ্য রাখতে হবে যেন ধনাত্মক প্রান্ত এবং ঋণাত্মক প্রান্ত উলটো না হয়।



এবার স্ট্রু টার্মিনাল এর দিকে লক্ষ্য করা যাক। এটা দেয়া হয়েছে এই কারণে যেন এখানে লিপো ব্যাটারির পাওয়ার ক্যাবলের সংযোগ দেয়া যায়। তোমাদের লিপো ব্যাটারিতে লক্ষ্য করে দেখবে যে মোটা দুটি লাল এবং কালো রঙের তার বের হয়ে আছে যার শেষ মাথায় একটা বিশেষ কানেক্টর বসানো থাকে পাওয়ার সংযোগ এর জন্য। বাজারে তো অনেক ধরনেরই লিপো ব্যাটারি আছে, এবং তাদের মাথাও ভিন্ন ভিন্ন হতে পারে। তবে সবথেকে কমন দুটি কানেক্টর হলো Dean T Connector এবং XT60 Connector। চলো নিচে দেখে নিই কোন কোন ধরনের কানেক্টরঃ



Male Female



XT60
Connector



Female Male



Dean T
Connector

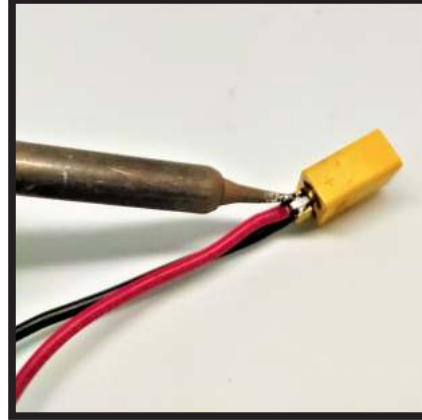
এখানে XT60 এবং Dean T কানেক্টর দেখা যাচ্ছে যেখানে প্রত্যেকটিতে দুই টাইপের কানেক্টর দেখা যায়; একটি হলো মেইল টাইপ আরেকটি হলো ফিমেইল টাইপ। এই মেইল এবং ফিমেইল টাইপ একসাথে লাগিয়েই সংযোগ স্থাপন করা হয় যা কিনা টান দিয়ে খুলে ফেলা যায়। প্রত্যেক টাইপেই দুটি করে টার্মিনাল থাকে যার মধ্যে একটি সবসময় পজেটিভ টার্মিনাল এবং আরেকটি নেগেটিভ টার্মিনাল হিসেবে ব্যবহার হয়ে থাকে। কোনটি পজেটিভ এবং কোনটি নেগেটিভ সেটা XT60 কানেক্টর এর গায়েই লিখে দেয়া থাকে। তারপরেও নিচে চিত্রের মাধ্যমে বুঝিয়ে দেয়া হচ্ছে যেটা কোনভাবেই উল্টো লাগানো উচিত হবেনা। এতে করে পুরো রোবটটিই নষ্ট হয়ে যেতে পারে এমনকি ব্যাটারিতে আগুনও ধরে যেতে পারে।



আরেকটু বিষয় খেয়াল করে দেখবে যে ব্যাটারির সাথে সবসময় ফিমেইল টাইপ টি লাগানো থাকে। বাজারে আলাদা করে এই ধরনের কানেক্টর জোড়ায় কিনতে পাওয়া যায় যেখান থেকে আমরা কেবল মেইল টাইপটি ব্যবহার করতে পারি ব্যাটারির সাথে পিসিবির সংযোগ এর জন্য। আপাতত এখানে আমরা XT60 কানেক্টরের ব্যবহার করছি। এক্ষেত্রে নিচের ধাপগুলো সম্পন্ন করলেই হয়ে যাবেঃ



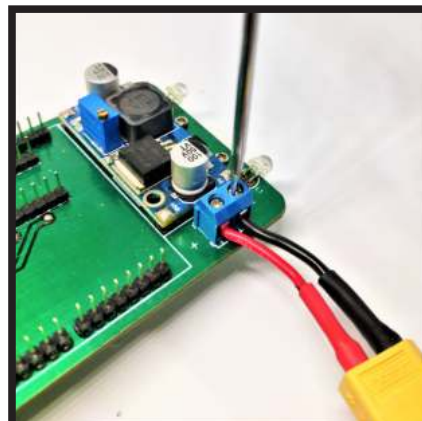
১) মেইল টাইপ XT60 কানেক্টর নাও



২) দুই মাথায় দুটি তার সোল্ডার করা

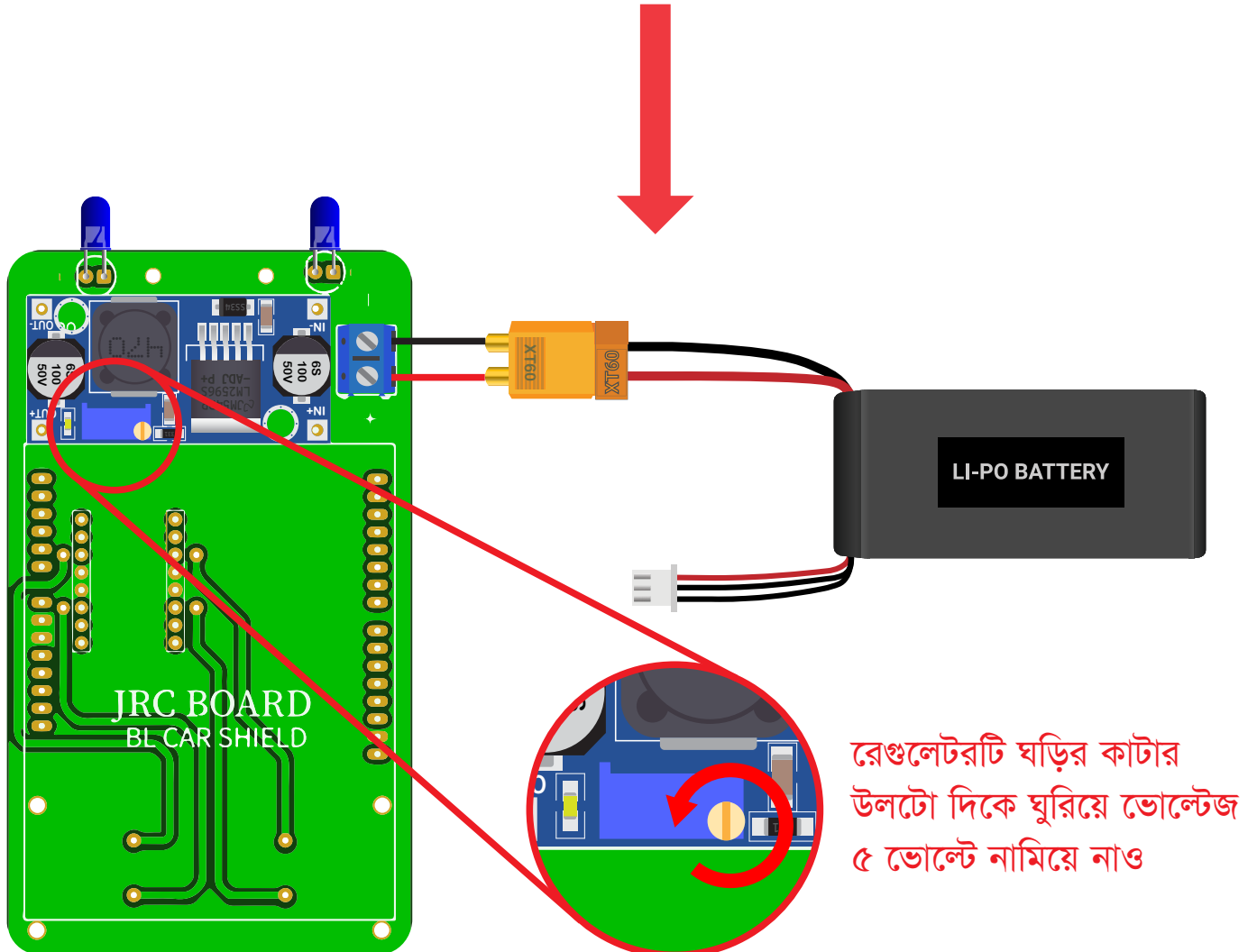
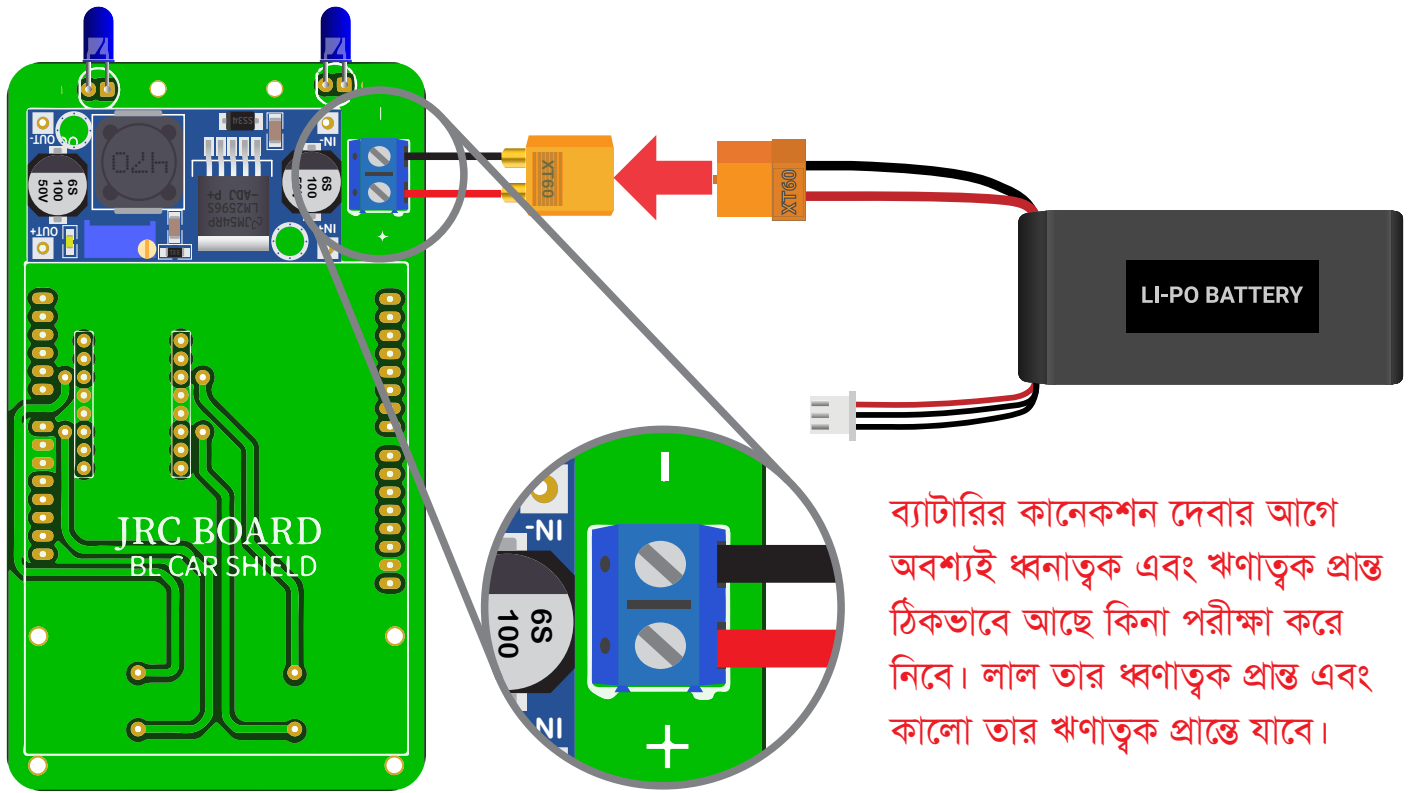


৩) জয়েন্টের অংশ গু গান বা Heat Shrinker দিয়ে সুরক্ষা করা

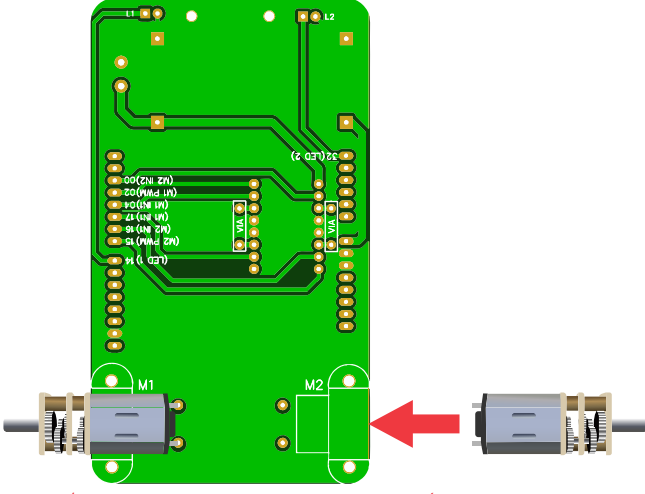


৪) তারের অপর প্রান্ত স্ক্রু টার্মিনাল এ লাগিয়ে স্ক্রু টাইট করে দাও

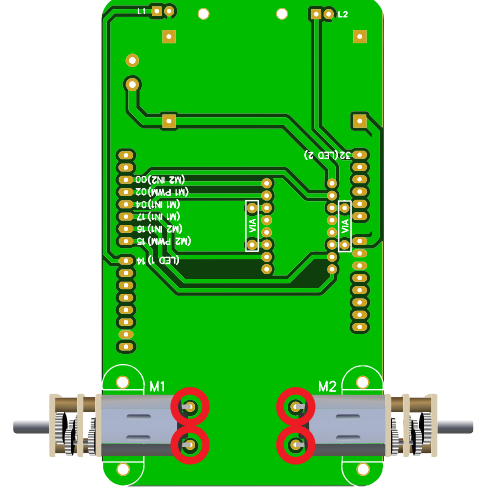
এর পরেই আমরা লিপো ব্যাটারি সংযোগ করে বাক কনভার্টার এর রেগুলেটরটি ঘুরিয়ে ভোল্টেজ টি নামিয়ে নিতে পারি যাতে করে পরবর্তীতে কোনপ্রকার সমস্যা না হয়।



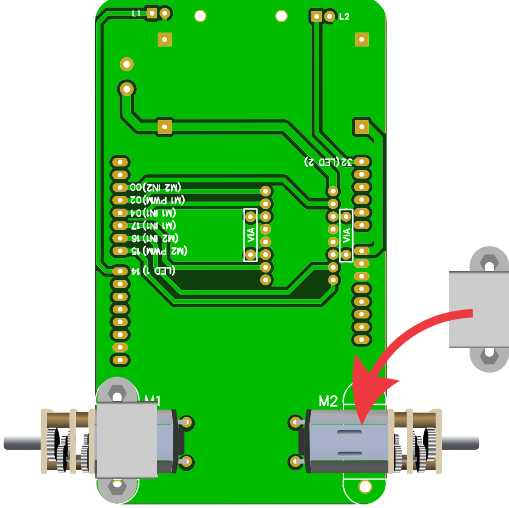
এভাবে বোর্ডের উপরের অংশ মোটামুটি প্রস্তুত হয়ে গেলো। এবার JRC Board বসানোর আগে নিচের অংশে যা যা লাগাবার সেগুলোও লাগাতে শুরু করি। চিত্রের ধাপগুলো অনুসরণ করলেই হয়ে যাবেঃ



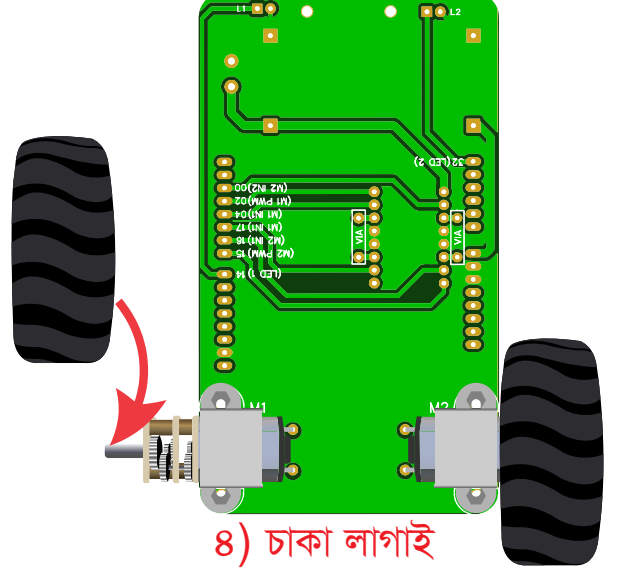
১) মটরগুলো জায়গামতো বসাই



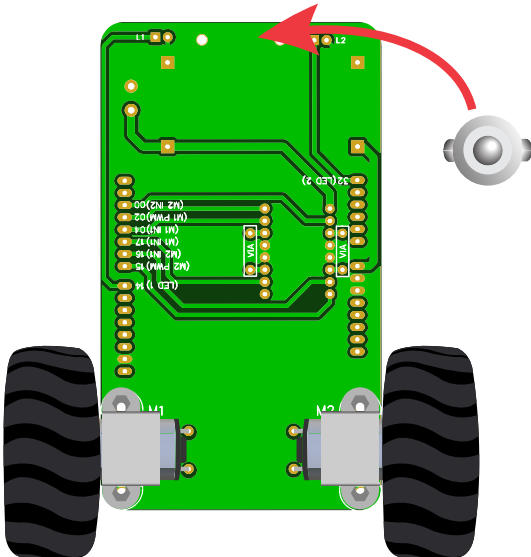
২) চিহ্নিত অংশ সোল্ডার করি



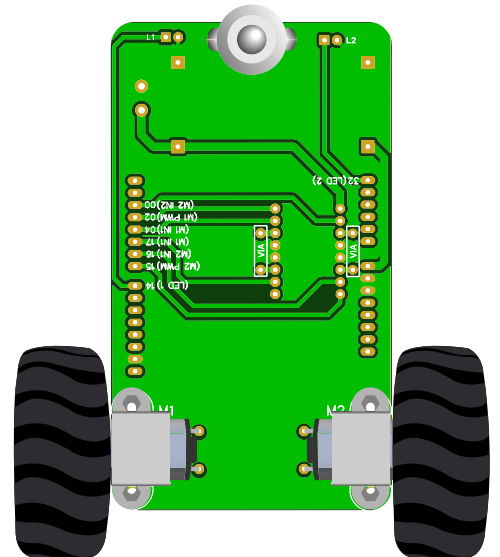
৩) মটর মাউন্ট বসাই



৪) চাকা লাগাই

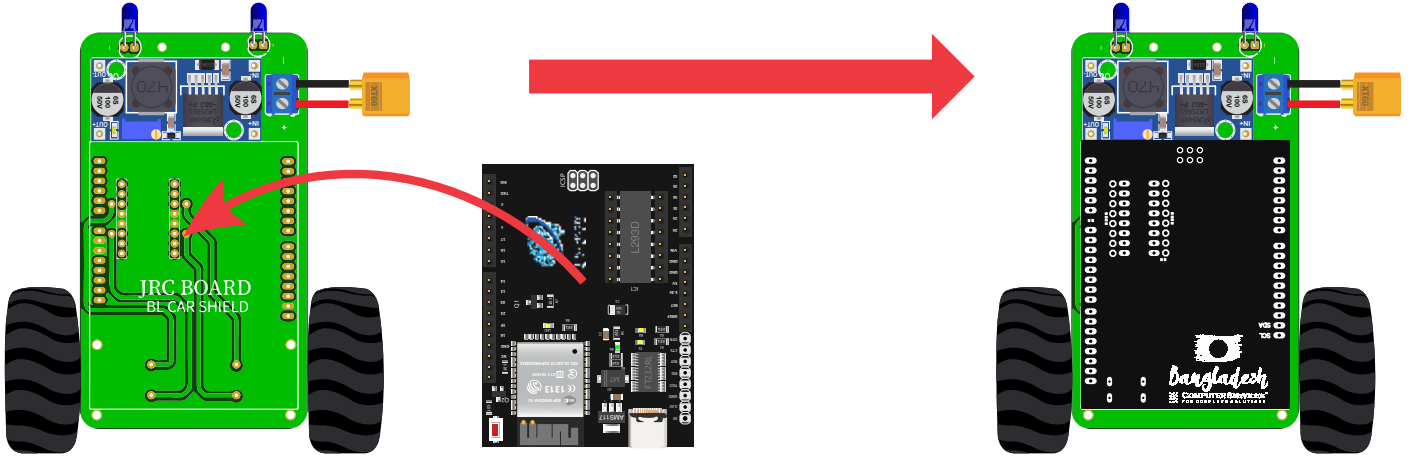


৫) বল কাস্টার লাগাই

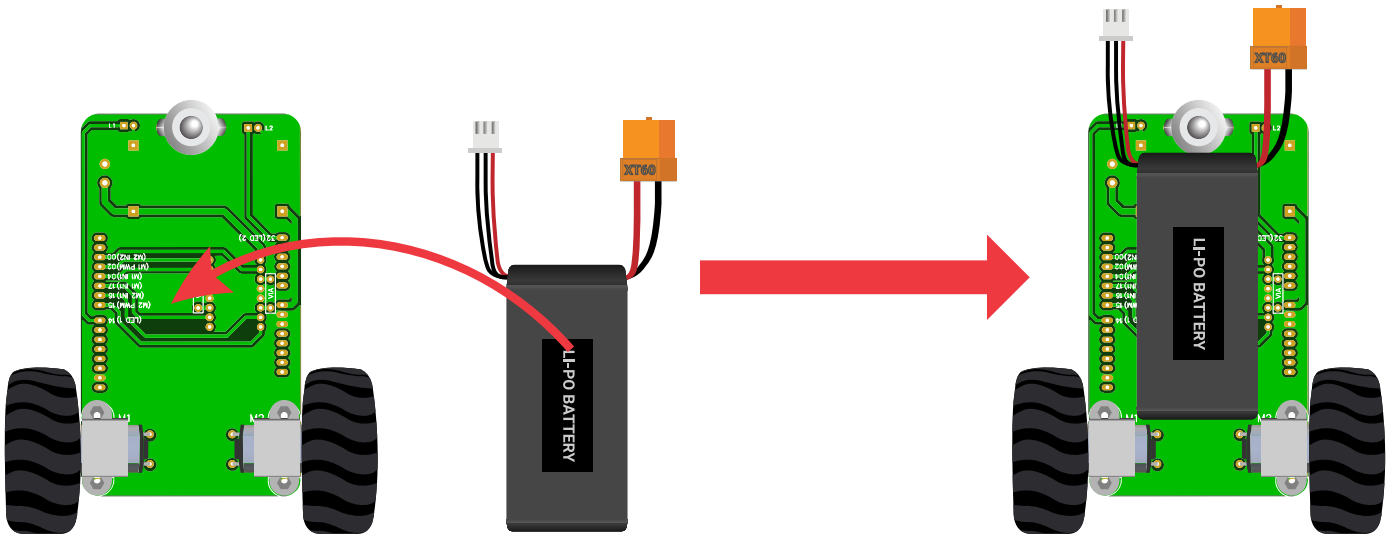


৬) রোবট তৈরী!

এবার সর্বশেষে উপরের দিকে JRC Board টি স্থাপন করতে পারিঃ

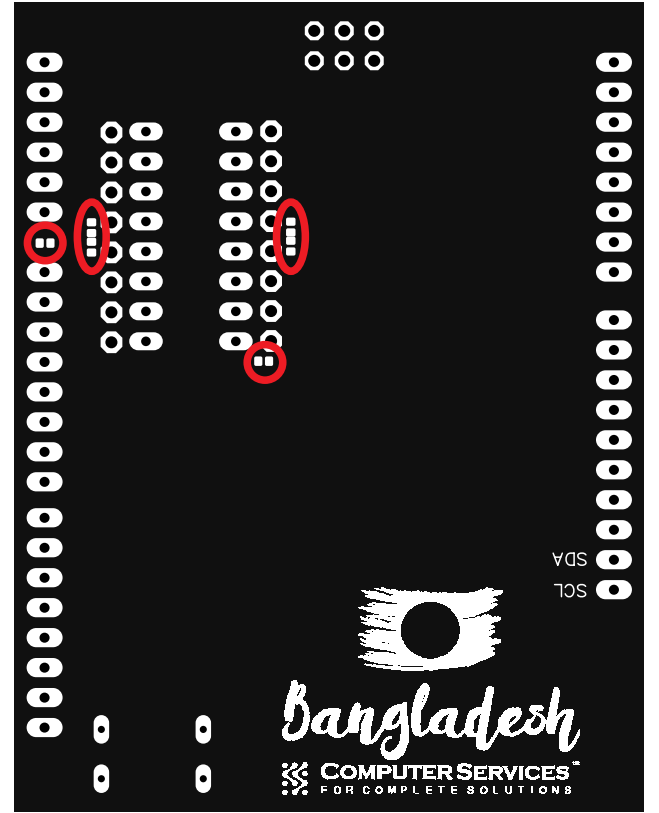


যেহেতু আপাতত ২ সেল লিপো ব্যাটারি ব্যবহার হচ্ছে, সেটা চাইলে রোবট এর নিচ থেকে বসিয়ে দেয়া যায়। তবে বেশি মোটা হলে উপর থেকেও বসাতে কোনপ্রকার অসুবিধা নেই। তবে উল্লেখ্য যে এখানে সর্বোচ্চ ৩ সেল অর্থাৎ ১২.৬ ভোল্ট ব্যবহার করা নিরাপদ এবং ইনপুট ভোল্টেজ কোনভানবেই ১৬ ভোল্টের বেশি হওয়া যাবেনা।

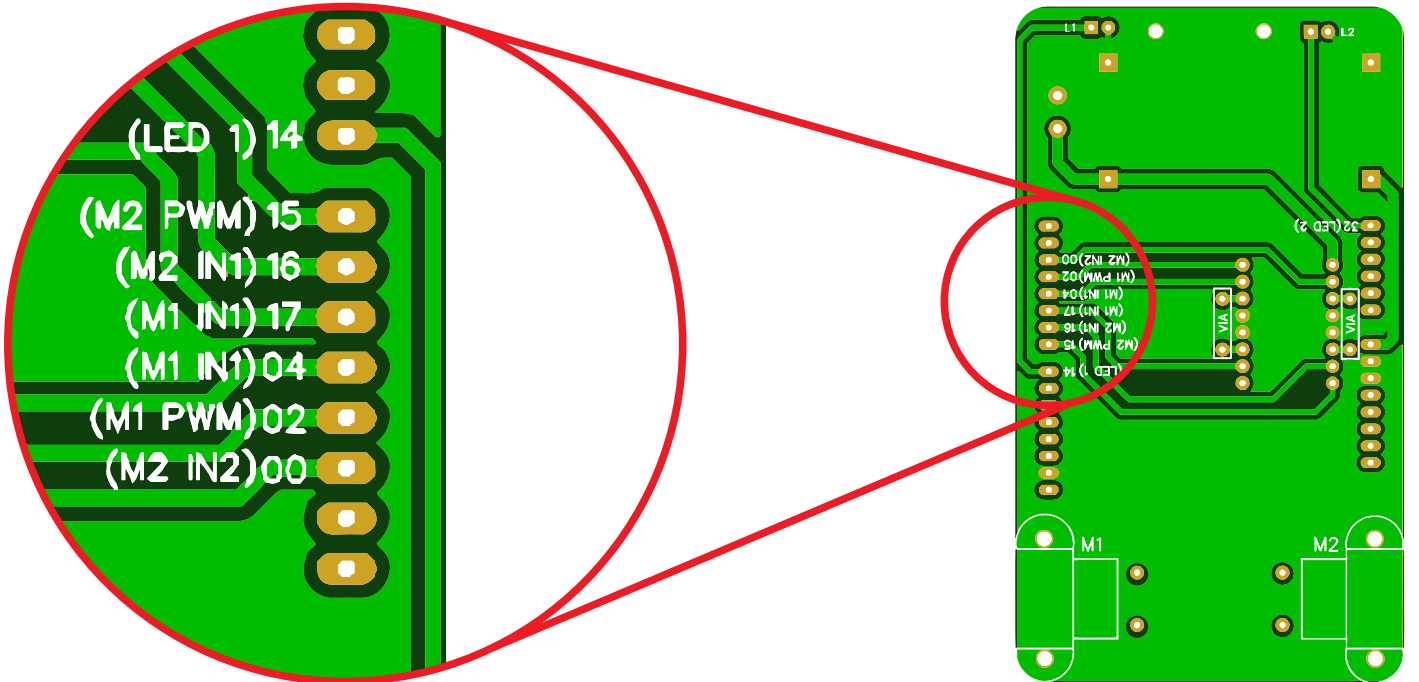


এবার প্রোগ্রামিং এর পালা। যেহেতু এটি একটি রিমোট কন্ট্রোল গাড়ি, এর মানে এখানে কেবল দুটি ফিচারই ব্যবহার হচ্ছে, এক হচ্ছে মটর ড্রাইভার কন্ট্রোল, আরেক হচ্ছে ব্লুটুথ রিসিভিং যে সম্পর্কে আমরা ইতিপূর্বেই আলোচনা করে এসেছি। এখানে পিসিবিতে এমনভাবে সার্কিট ডিজাইন করা আছে যেন মটর ড্রাইভার এর মটর কন্ট্রোল পিনগুলো JRC Board এর নির্দিষ্ট পিনের সাথে সংযুক্ত হয়। উল্লেখ্য যে JRC Board এ একটি L293D মটর ড্রাইভার আইসি বসানো যায় যা দিয়ে দুটি মটর আলাদা আলাদাভাবে নিয়ন্ত্রন করা সম্ভব। এবং এই মটর ড্রাইভার এন্টিভেট করতে কিছু প্যাড ঝালাই করে নিতে হয় যা নিচের ছবিতে আবারও দেখিয়ে দেয়া হচ্ছে। এর মানে আমাদের আর কোনপ্রকার তারের ঝামেলাই নেই এবং অতি সহজেই পিন বুঝে প্রোগ্রামিং করে এরপর এই রোবটটি কর্মক্ষম করে তুলতে পারি।

চিহ্নিত প্যাডগুলো সোল্ডার করে যুক্ত করে দিতে হবে।



এই পিসিবির মাধ্যমে JRC Board এর কোন পিনের সাথে মটর ড্রাইভার এর কোণ পিন লাগানো আছে সেটা খুব সুন্দর করে পিসিবির নিচের অংশে চিহ্নিত করে দেয়া আছে। লক্ষ্য করলেই দেখা যায়ঃ



LED 1	LED 2	Motor 1 Input 1	Motor 1 Input 2	Motor 1 PWM pin	Motor 2 Input 1	Motor 2 Input 2	Motor 1 PWM pin
14	32	17	04	02	16	00	15

উপরের তালিকা থেকে আমরা পিন নাম্বার গুলো জেনে নিতে পেরেছি। এখন আমাদের প্রোগ্রামিং এর সময় পিন নাম্বারগুলো মাথায় রাখলেই বাকি কোডিং করা অনেকটুকু সহজ হয়ে যাবে। প্রথমেই আমরা বেসিক ব্লুটুথ রিসিভিং সেটাপ করি এবং পিন গুলো কতগুলো ভ্যারিয়েবলের নামে ডিক্লেয়ার করিঃ

```
#include "BluetoothSerial.h"
BluetoothSerial SerialBT;
int LED1 = 14, LED2 = 32;
int M1_front = 17, M1_back = 4, M1_pwm = 2;
int M2_front = 16, M2_back = 0, M2_pwm = 15;
void setup(){
    Serial.begin(9600);
    SerialBT.begin("Bluetooth RC Car");
    Serial.print("BT Started!");
    ledcSetup(0,1000,8);
    ledcSetup(1,1000,8);
    ledcAttachPin(M1_pwm,0);
    ledcAttachPin(M2_pwm,1);
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(M1_front, OUTPUT);
    pinMode(M1_back, OUTPUT);
    pinMode(M2_front, OUTPUT);
    pinMode(M2_back, OUTPUT);
}
void loop(){
    if(SerialBT.available()) {
        char x = SerialBT.read();
        Serial.write(x);
    }
}
```

মটর এবং এলইডি কন্ট্রোল করার জন্য দরকারি পিনসমূহের নাম্বার ভ্যারিয়েবল হিসেবে নিই

ব্লুটুথ কমিউনিকেশন শুরু করছি যেখানে ডিভাইসটির নাম হবে Bluetooth RC Car

মটর এর স্পিড কন্ট্রোল এর জন্য PWM পিনসমূহ সেটাপ করি

বাকি এলইডি এবং মটর এর ঘূর্ণন দিক নিয়ন্ত্রন এর জন্য দরকারি ডিজিটাল পিনসমূহ কে আউটপুট হিসেবে নির্বাচন করি

আপাতত ব্লুটুথ এর মাধ্যমে কোন কমান্ড আসলে সেটি একটি ক্যারেক্টার টাইপ ভ্যারিয়েবলে জমা রাখছি এবং এখানে কোন ভ্যালু আসলো সেটি সিরিয়াল মনিটরে দেখানোর ব্যবস্থা করছি

আমাদের বেসিক সেটাপ সম্পন্ন হয়েছে। একটু চিন্তা করে দেখো তো কোডে এরপর কিরকম কাজ হওয়া উচিত? আমরা স্মার্টফোন থেকে ব্লুটুথ এর মাধ্যমে যেকোন অক্ষর বা তথ্য পাঠাবো এবং সেটা রোবট পড়ে নিয়ে এরপর সে অনুযায়ী মটরগুলোকে পরিচালনা করবে তাইনা? সেক্ষেত্রে প্রথমে আমাদের এমন একটা ফাংশন তৈরী করে রাখা দরকার যেখানে কিনা দুটি মটর এর জন্য আলাদা আলাদা মান ইনপুট নিয়ে এরপর সে মান অনুযায়ী মটর এর গতি এবং দিক নিয়ন্ত্রন করবে। প্রথমে দেখে নিই যে একটি ফাংশনে একটি ইনপুট নিয়ে সেটি দিয়ে কিভাবে মটর পরিচালনা করা যায়ঃ

```
void motor1(int a) {
    if (a > 0) {
        digitalWrite(M1_front, 1);
        digitalWrite(M1_back, 0);
    }
    else{
        a = -a;
        digitalWrite(M1_front, 0);
        digitalWrite(M1_back, 1);
    }
    ledcWrite(0, a);
}
```

পাশের কোডে লক্ষ্য করে দেখো যে একটা ম্যানুয়াল ফাংশন "motor1" নামে বানানো হয়েছে যেখানে একটা ভ্যালু ইনপুট নিতে যা কিনা a নামক ভ্যারিয়েবল এ জমা হয়। এখন a এর মান যদি ধনাত্মক হয় তবে মটরটি সামনে ঘুরতে শুরু করার কথা (বিস্তারিত দেখো ৭ নং প্রজেক্ট এ)। আর যদি a এর মান ঋণাত্মক হয় তাইলে মটরটি উলটো দিকে, অর্থাৎ পেছনের দিকে ঘুরতে শুরু করার কথা। তবে লক্ষ্য করলে দেখতে পাবে a এর ধনাত্মক এবং ঋণাত্মক মান দিয়ে যে শুধুমাত্র মটর এর ঘূর্ণন দিক নিয়ন্ত্রন করা হচ্ছে

তাইই নয়, a এর মানের উপরে নির্ভর করে মটর এর ঘূর্ণন গতি কিরকম হবে সেটাও ঠিক করে দেয়া যাচ্ছে। কারণ আমরা `ledcWrite(0, a);` এর মাধ্যমে Motor 1 এর জন্য গতি নিয়ন্ত্রক Enable পিনে PWM সিগনাল পাঠাচ্ছি। পূর্বের মটর ড্রাইভিং সিস্টেম এর কথা যদি মনে থাকে, তবে এটা বুঝে ফেলার কথা যে `ledcSetup(0,1000,8);` এর মাধ্যমে এখানে PWM সিগনাল এর সর্বোচ্চ মান ৮-বিট অর্থাৎ ২৫৫ পর্যন্ত নিয়ে রেখেছি এবং সর্বনিম্ন মান শূন্য। এর মানে যদি এখানে `ledcWrite(0, a);` ফাংশনে a এর মান ঋণাত্মক হয়, তবে এই ফাংশনটি কাজই করবেনা। যে কারণে যদি আমরা a এর মান ঋণাত্মক পাই, সেক্ষেত্রে $a = -a$; লেখার মাধ্যমে a এর মান ধনাত্মক করে ফেলতে পারছি। এর মানে ফাংশন অনুযায়ী মটর এর ঘূর্ণন দিক ঠিকই উলটে যাচ্ছে, কিন্তু গতির বেলাতে কোন সমস্যা হচ্ছেনা। এই ফাংশন টেস্ট করতে মূল কোডে কিছু জিনিস যোগ করে নিইঃ

```
#include "BluetoothSerial.h"
BluetoothSerial SerialBT;
int LED1 = 14, LED2 = 32;
int M1_front = 17, M1_back = 4, M1_pwm = 2;
int M2_front = 16, M2_back = 0, M2_pwm = 15;
void setup(){
    Serial.begin(9600);
    SerialBT.begin("Bluetooth RC Car"); Serial.print("BT Started!");
    ledcSetup(0,1000,8); ledcAttachPin(M1_pwm,0);
    ledcSetup(1,1000,8); ledcAttachPin(M2_pwm,1);
    pinMode(LED1, OUTPUT); pinMode(LED2, OUTPUT);
    pinMode(M1_front, OUTPUT); pinMode(M1_back, OUTPUT);
    pinMode(M2_front, OUTPUT); pinMode(M2_back, OUTPUT);
}
void loop(){
    motor1(255); delay(5000); —————> মটর পূর্ণ গতিতে সামনের দিকে ঘুরবে ৫ সেকেন্ডের জন্য
    motor1(0); delay(2000); —————> মটর থেমে থাকবে ২ সেকেন্ডের জন্য
    motor1(-100); delay(5000); —————> মটর অল্প গতিতে পেছনের দিকে ঘুরবে ৫ সেকেন্ডের জন্য
    motor1(-255); delay(5000); —————> মটর পূর্ণ গতিতে পেছনের দিকে ঘুরবে ৫ সেকেন্ডের জন্য
}
void motor1(int a) {
    if (a > 0) {
        digitalWrite(M1_front, 1);
        digitalWrite(M1_back, 0);
    }
    else{
        a = -a;
        digitalWrite(M1_front, 0);
        digitalWrite(M1_back, 1);
    }
    ledcWrite(0, a);
}
```

এভাবে চাইলে motor 1 সঠিকভাবে ঘুরছে কিনা সেটা পরীক্ষা করে দেখতে পারি। যদি দেখতে পাও যে মটর সামনেও দিকে ঘুরতে বলা হলেও সে পেছনের দিকে ঘুরছে সেক্ষেত্রে কারণ এটা যে মটর এর কানেকশন উলটো হয়েছে। তবে অযথা মটর খুলে আবার ঘুরিয়ে বসানোর ঝামেলা বাদ দিয়ে চাইলে কেবল কোডেই এটি ঠিক করে নেয়া যায়। সেক্ষেত্রে কেবল শুরুতে M1_front এবং M1_back এর মান অদল বদল করে দিতে হবে।

```
#include "BluetoothSerial.h"
BluetoothSerial SerialBT;
int LED1 = 14, LED2 = 32;
int M1_front = 17, M1_back = 4, M1_pwm = 2;
int M2_front = 16, M2_back = 0, M2_pwm = 15;
```



```
#include "BluetoothSerial.h"
BluetoothSerial SerialBT;
int LED1 = 14, LED2 = 32;
int M1_front = 4, M1_back = 17, M1_pwm = 2;
int M2_front = 16, M2_back = 0, M2_pwm = 15;
```

এভাবে আমরা motor 1 (ডানদিকের মটর) এর সেটআপ সম্পন্ন করে ফেললাম। ঠিক একই উপায়ে motor 2 এর সবকিছু সেটআপ করে ফেলা যায়।

```
#include "BluetoothSerial.h"
BluetoothSerial SerialBT;
int LED1 = 14, LED2 = 32;
int M1_front = 17, M1_back = 4, M1_pwm = 2;
int M2_front = 16, M2_back = 0, M2_pwm = 15;
void setup(){
    Serial.begin(9600);
    SerialBT.begin("Bluetooth RC Car"); Serial.print("BT Started!");
    ledcSetup(0,1000,8); ledcAttachPin(M1_pwm,0);
    ledcSetup(1,1000,8); ledcAttachPin(M2_pwm,1);
    pinMode(LED1, OUTPUT); pinMode(LED2, OUTPUT);
    pinMode(M1_front, OUTPUT); pinMode(M1_back, OUTPUT);
    pinMode(M2_front, OUTPUT); pinMode(M2_back, OUTPUT);
}
void loop(){
    motor2(255); delay(5000); —————> মটর পূর্ণ গতিতে সামনের দিকে ঘুরবে ৫ সেকেন্ডের জন্য
    motor2(0); delay(2000); —————> মটর থেমে থাকবে ২ সেকেন্ডের জন্য
    motor2(-100); delay(5000); —————> মটর অল্প গতিতে পেছনের দিকে ঘুরবে ৫ সেকেন্ডের জন্য
    motor2(-255); delay(5000); —————> মটর পূর্ণ গতিতে পেছনের দিকে ঘুরবে ৫ সেকেন্ডের জন্য
}
void motor2(int a) {
    if (a > 0) {
        digitalWrite(M2_front, 1);
        digitalWrite(M2_back, 0);
    }
    else{
        a = -a;
        digitalWrite(M2_front, 0);
        digitalWrite(M2_back, 1);
    }
    ledcWrite(1, a);
}
```

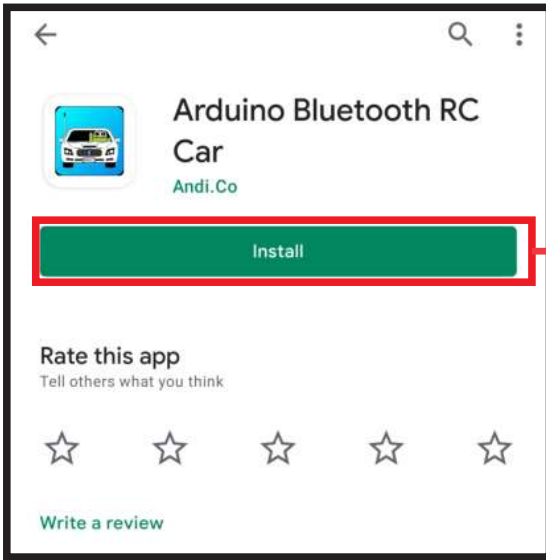
এভাবে দুই মটরই টেস্ট করা সম্পন্ন হয়ে গেলে এরপর চাইলে দুই মটর এর ফাংশন একত্রে একটি ফাংশন আকারেই লিখে ফেলা যায়।

```

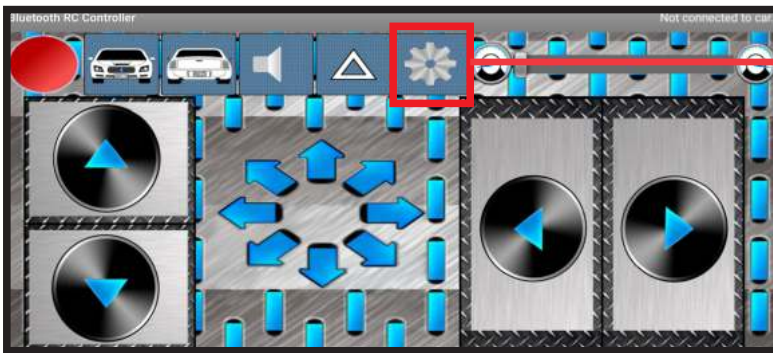
#include "BluetoothSerial.h"
BluetoothSerial SerialBT;
int LED1 = 14, LED2 = 32;
int M1_front = 17, M1_back = 4, M1_pwm = 2;
int M2_front = 16, M2_back = 0, M2_pwm = 15;
void setup(){
    Serial.begin(9600);
    SerialBT.begin("Bluetooth RC Car"); Serial.print("BT Started!");
    ledcSetup(0,1000,8); ledcAttachPin(M1_pwm,0);
    ledcSetup(1,1000,8); ledcAttachPin(M2_pwm,1);
    pinMode(LED1, OUTPUT); pinMode(LED2, OUTPUT);
    pinMode(M1_front, OUTPUT); pinMode(M1_back, OUTPUT);
    pinMode(M2_front, OUTPUT); pinMode(M2_back, OUTPUT);
}
void loop(){
    motor(255,255); delay(5000); —————> দুই মটরই পূর্ণ গতিতে সামনে ঘুরবে
    motor(0,255); delay(5000); —————> বামের মটর বন্ধ কিন্তু ডানের মটর পূর্ণ গতিতে সামনে ঘুরছে
    motor(0,0); delay(2000); —————> দুই মটরই বন্ধ
    motor(100,100); delay(5000); —————> দুই মটরই অল্প গতিতে সামনে ঘুরবে
    motor(-100,100); delay(5000); —————> বামের মটর পিছনে এবং ডানের মটর সামনে ঘুরছে অল্প গতিতে
    motor(-100,-100); delay(5000); —————> দুই মটরই অল্প গতিতে পেছনে ঘুরবে
    motor(-255,-255); delay(5000); —————> দুই মটরই পূর্ণ গতিতে পেছনে ঘুরবে
}
void motor(int a, int b) {
    if (a > 0) {
        digitalWrite(M2_front, 1); digitalWrite(M2_back, 0);
    }
    else{
        a = -a;
        digitalWrite(M2_front, 0); digitalWrite(M2_back, 1);
    }
    if (b > 0) {
        digitalWrite(M1_front, 1); digitalWrite(M1_back, 0);
    }
    else{
        b = -b;
        digitalWrite(M1_front, 0); digitalWrite(M1_back, 1);
    }
    ledcWrite(1, a); ledcWrite(0, b);
}

```

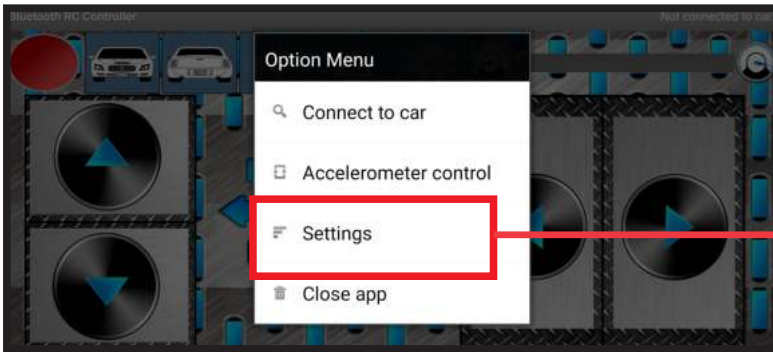
এভাবে চাইলে দুইটি মটরেই বিভিন্ন রকম মুভমেন্ট পরীক্ষা করে নেয়া যায় এবং এখানে সর্বশেষ একবার নিশ্চিত হয়ে যেয়া যায় যে রোবটের মটর এবং চাকা আসলেই ঠিক আছে কিনা এবং আমাদের কথামতো কাজ করছে কিনা। এরপর এখন কাজ হচ্ছে বুটুথ এ সিগনাল কিংবা একটি অক্ষর রিসিভ করা এবং সেই অনুযায়ী কাজ সম্পাদন করা। এর জন্য আমাদের ফোনে একটি এন্ড্রয়েড অ্যাপ লাগবে যেটির নাম হচ্ছে **Bluetooth RC Car**। এই অ্যাপটি বুটুথ দিয়ে **JRC Board** এর সাথে যুক্ত হবে এবং ভেতরে বিভিন্ন বাটন প্রেস করলে বিভিন্ন রকম অক্ষর পাঠায়। যে অক্ষরটি **JRC Board** রিসিভ করে, সে অনুযায়ী আমরা যদি বোর্ডের ভেতরে প্রোগ্রাম করে মটর এর মুভমেন্ট নির্ধারণ করে দিই, তাহলেই কিন্তু আমাদের প্রজেক্ট দাড়িয়ে যায়। প্রথমেই তাহলে এপটি নামিয়ে ইন্সটল করে নিই এবং এরপর ওপেন করে সেটিংস চেক করি। **JRC Board** এর বুটুথ এর সাথে কিভাবে ফোনের বুটুথ এর পেয়ার করতে হবে সেটি আমরা ইতিমধ্যেই ৮ নং প্রজেক্টে দেখে এসেছি।



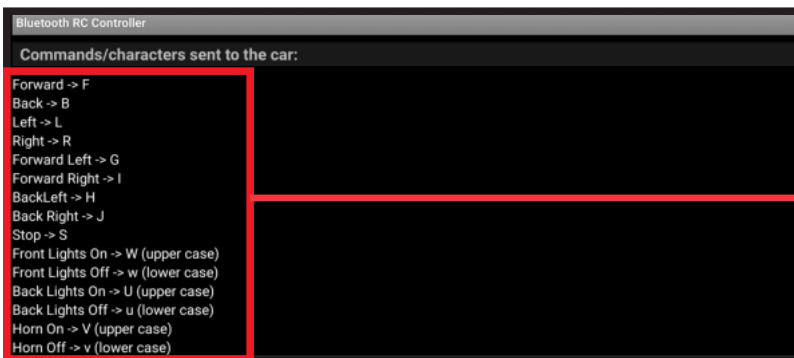
প্রথমে এই এপটি ইন্সটল করে নিই এবং ওপেন করি



সেটিংস চিহ্নে ক্লিক করি



সেটিংস অপশন নির্বাচন করি



এখানে স্ক্রল করে একটু নিচে নামলেই কোন কমান্ডে কোন অক্ষর পাঠাচ্ছে তার একটি তালিকা পাওয়া যাবে

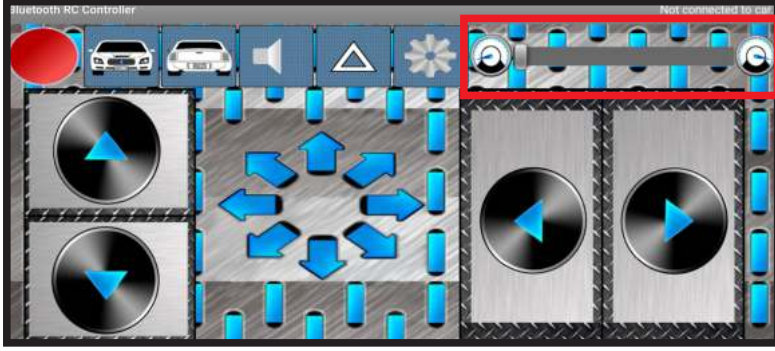
এভাবে আমরা দেখে নিতে পারলাম যে এই ব্লুটুথ রিমোট কন্ট্রোল এপ এর ভেতরে কোন বাটন প্রেস করলে কোন অক্ষর কমান্ড হিসেবে যায়। সে অক্ষরগুলো আমরা যদি কোডে ইনপুট নিয়ে সেই অনুযায়ী লজিক সাজিয়ে ফেলি, তাহলে সেটি এমন দেখায়ঃ

```

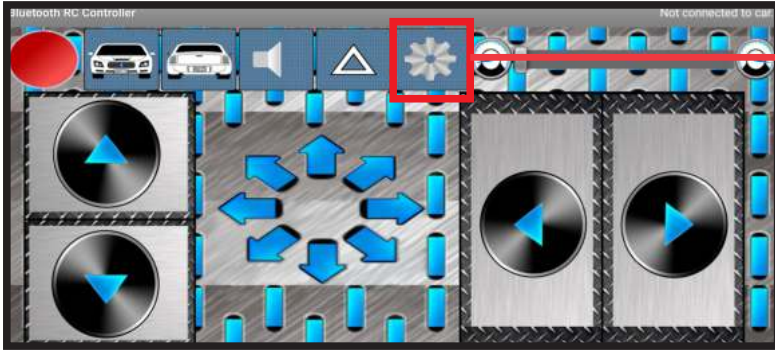
#include "BluetoothSerial.h"
BluetoothSerial SerialBT;
int LED1 = 14, LED2 = 32, speed = 10;
int M1_front = 17, M1_back = 4, M1_pwm = 2;
int M2_front = 16, M2_back = 0, M2_pwm = 15;
void setup(){
    Serial.begin(9600);
    SerialBT.begin("Bluetooth RC Car"); Serial.print("BT Started!");
    ledcSetup(0,1000,8); ledcAttachPin(M1_pwm,0);
    ledcSetup(1,1000,8); ledcAttachPin(M2_pwm,1);
    pinMode(LED1, OUTPUT); pinMode(LED2, OUTPUT);
    pinMode(M1_front, OUTPUT); pinMode(M1_back, OUTPUT);
    pinMode(M2_front, OUTPUT); pinMode(M2_back, OUTPUT);
}
void loop(){
    if(SerialBT.available()) {
        char x = SerialBT.read();
        if(x == 'F') motor(25*speed , 25*speed);
        if(x == 'B') motor(-25*speed , -25*speed);
        if(x == 'L') motor(-25*speed , 25*speed);
        if(x == 'R') motor(25*speed , -25*speed);
        if(x == 'G') motor(0 , 25*speed);
        if(x == 'I') motor(25*speed , 0);
        if(x == 'H') motor(0 , -25*speed);
        if(x == 'J') motor(-25*speed , 0);
        if(x == 'S') motor(0 , 0);
        if(x == 'W') {
            digitalWrite(LED1, 1); digitalWrite(LED2, 1);
        }
        if(x == 'w') {
            digitalWrite(LED1, 0); digitalWrite(LED2, 0);
        }
    }
}
void motor(int a, int b) {
    if (a > 0) {
        digitalWrite(M2_front, 1); digitalWrite(M2_back, 0);
    }
    else{
        a = -a;
        digitalWrite(M2_front, 0); digitalWrite(M2_back, 1);
    }
    if (b > 0) {
        digitalWrite(M1_front, 1); digitalWrite(M1_back, 0);
    }
    else{
        b = -b;
        digitalWrite(M1_front, 0); digitalWrite(M1_back, 1);
    }
    ledcWrite(1, a); ledcWrite(0, b);
}

```

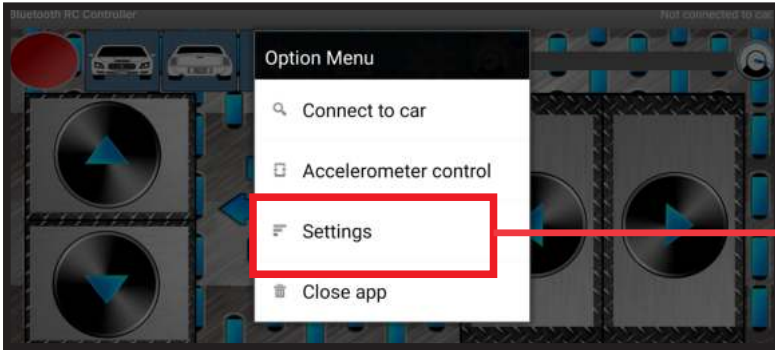

এখানে লক্ষ্য করে দেখবে যে `motor()` ফাংশনে সরাসরি স্পিডের মান না বসিয়ে `speed` নামের একটি ভ্যারিয়েবল ব্যবহার করা হয়েছে যার সর্বোচ্চ মান ১০ এবং সর্বনিম্ন মান ০ হতে পারে। এর মান ১০ হলে $25 \times \text{speed}$ এর মান হয় ২৫০ যা কিনা মটর কে পূর্ণ গতিতে ঘোরানোর কথা। এর মানে তুমি চাইলে `speed` ভ্যারিয়েবলের মান ০ থেকে ১০ এর মধ্যে ইচ্ছামতো যেকোন মান লিখে দিলে রোবটটি সেই স্পিড অনুযায়ী চলবে। ০ মানে ০% স্পিড, ১০ মানে ১০০% স্পিড। আবার চাইলে এপ থেকেও রোবটের স্পিড কন্ট্রোল করা যায়। নিচের ছবির দিকে লক্ষ্য করঃ



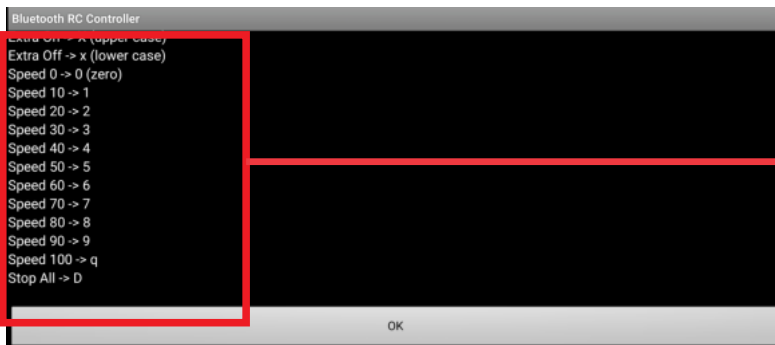
এই স্লাইডার দিয়ে স্পিড নিয়ন্ত্রন করে



সেটিংস চিহ্নে ক্লিক করি



সেটিংস অপশন নির্বাচন করি



কোন স্পিডে কেমন অক্ষর কমান্ড পাঠাবে সেই তালিকা স্ক্রল করে সবার নিচে পাওয়া যাবে

এভাবে আমরা দেখে নিতে পারলাম যে এই বুটুথ রিমোট কন্ট্রোল এপ এর ভেতরে স্পিড নিয়ন্ত্রন ক্রয়ারও অপশন আছে এবং সেটি দিয়ে চাইলে রোবটের গতি নির্দেশ করে দেয়া যায়। সেক্ষেত্রে কোডের কিছু বাড়তি অংশ জুড়ে দিলেই হয়ে যাবে।

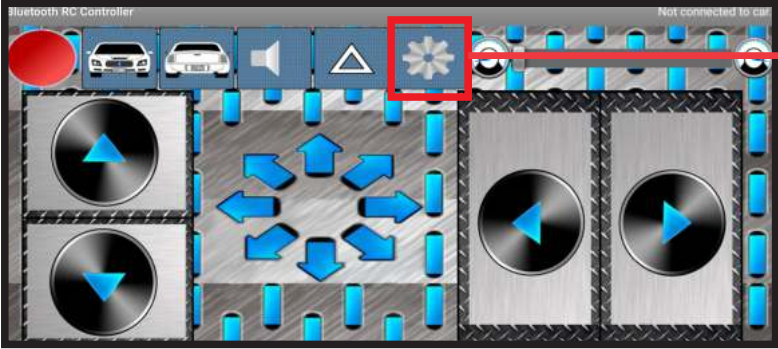
```

#include "BluetoothSerial.h"
BluetoothSerial SerialBT;
int LED1 = 14, LED2 = 32, speed = 10;
int M1_front = 17, M1_back = 4, M1_pwm = 2;
int M2_front = 16, M2_back = 0, M2_pwm = 15;
void setup(){
    Serial.begin(9600);
    SerialBT.begin("Bluetooth RC Car"); Serial.print("BT Started!");
    ledcSetup(0,1000,8); ledcAttachPin(M1_pwm,0);
    ledcSetup(1,1000,8); ledcAttachPin(M2_pwm,1);
    pinMode(LED1, OUTPUT); pinMode(LED2, OUTPUT);
    pinMode(M1_front, OUTPUT); pinMode(M1_back, OUTPUT);
    pinMode(M2_front, OUTPUT); pinMode(M2_back, OUTPUT);
}
void loop(){
    if(SerialBT.available()) {
        char x = SerialBT.read();
        if(x == 'F') motor(25*speed , 25*speed);
        else if(x == 'B') motor(-25*speed , -25*speed);
        else if(x == 'L') motor(-25*speed , 25*speed);
        else if(x == 'R') motor(25*speed , -25*speed);
        else if(x == 'G') motor(0 , 25*speed);
        else if(x == 'I') motor(25*speed , 0);
        else if(x == 'H') motor(0 , -25*speed);
        else if(x == 'J') motor(-25*speed , 0);
        else if(x == 'S') motor(0 , 0);
        else if(x == 'W') {
            digitalWrite(LED1, 1); digitalWrite(LED2, 1);
        }
        else if(x == 'w') {
            digitalWrite(LED1, 0); digitalWrite(LED2, 0);
        }
        else if(x >= '0' && x <= '9') speed = x-'0';
        else if(x == 'q') speed = 10;
    }
}

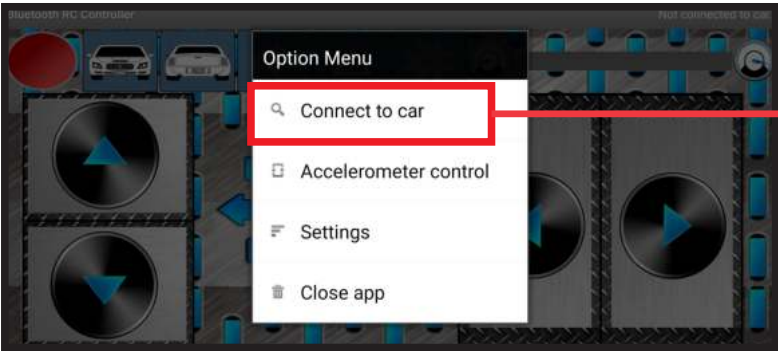
void motor(int a, int b) {
    if (a > 0) {
        digitalWrite(M2_front, 1); digitalWrite(M2_back, 0);
    }
    else{
        a = -a;
        digitalWrite(M2_front, 0); digitalWrite(M2_back, 1);
    }
    if (b > 0) {
        digitalWrite(M1_front, 1); digitalWrite(M1_back, 0);
    }
    else{
        b = -b;
        digitalWrite(M1_front, 0); digitalWrite(M1_back, 1);
    }
    ledcWrite(1, a); ledcWrite(0, b);
}

```

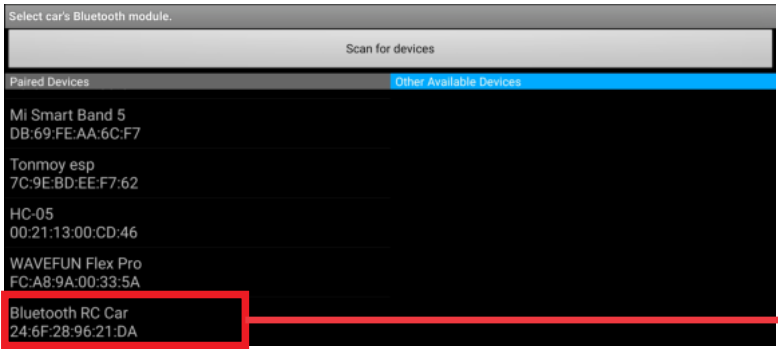
এভাবে আমাদের পুরো কোডটি দাঁড়িয়ে গেলো। এখন চাইলে আমরা রোবটে এই কোড টি আপলোড করে এরপর রোবটে ব্যাটারির সংযোগ দিয়ে এরপর মোবাইল এপ থেকে রোবটের বুটুথ সংযোগ স্থাপন করে মনের মতো রোবটটিকে চালিয়ে দেখতে পারি।



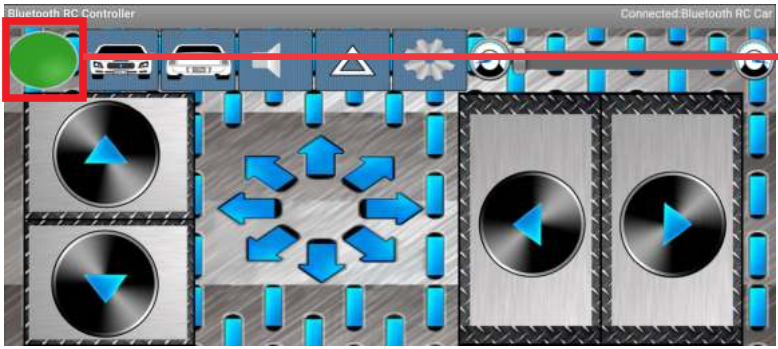
সেটিংস চিহ্নে ক্লিক করি



Connect to Car নির্বাচন করি



কাজিত বুটুথ ডিভাইসের নামটি খুঁজে বের করে তাতে ক্লিক করে কানেক্ট করি



কানেক্ট হয়ে গেলে এখানে লাল এর বদলে সবুজ লাইট জ্বলে উঠবে।

এভাবে একদম শুরু থেকে শেষ পর্যন্ত ধাপগুলো ফলো করে একটি আকর্ষণীয় বুটুথ কার বানিয়ে ফেলতে পারো। এটা কেবলই একটা বেজ মডেল, যাতে চাইলে আরও অনেক অনেক ফিচার যোগ করা যায় বা অন্যান্য আরও অনেক কাজ করে দেখানো সম্ভব। নিজেই একবার চেষ্টা করে দেখো যে এটায় আর কি কি যোগ করা যায়! তোমার কৌতূহল এই রোবটকে আরও বেশি পূর্ণতা প্রদান করবে।