

## এলইডির মিটিমিটি:

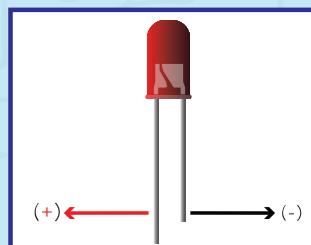
### প্রয়োজনীয় উপকরণ:

- ১) এলইডি লাইট
- ২) JRC বোর্ড
- ৩) ৪৭০০ ওম রেজিস্টর
- ৪) জাম্পার ওয়্যার
- ৫) ২২০ ওহম রেজিস্টর

### বর্ণনা:

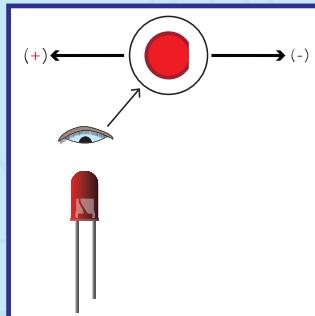
#### এলইডি লাইট:

প্রথমেই বলি একটা এলইডি লাইট কিভাবে কাজ করে। এটায় তুমি যদি ব্যাটারী থেকে শক্তি সরবরাহ করো, তবে এটি জ্বলে উঠবে। সাধারণত এতে দুটি প্রান্তের বিভিন্ন কেবল এক দিক থেকেই দেয়া যায়। মানে এতে ধনাত্মক এবং ঋণাত্মক প্রান্ত থাকে। লাইটের ধনাত্মক প্রান্তের সাথে ব্যাটারির ধনাত্মক প্রান্ত এবং লাইটের ঋণাত্মক প্রান্তের সাথে ব্যাটারির ঋণাত্মক প্রান্তের সংযোগ দিলেই কেবল লাইট টি জ্বলে উঠবে। উল্লেখ সংযোগ দিলে লাইট জ্বলবেনা। সাধারণত আমরা যে ছোট এলইডি ব্যবহার করি, তাতে ধনাত্মক প্রান্ত এবং ঋণাত্মক প্রান্ত চিহ্নিত থাকে। নিচের ছবিটি লক্ষ করোঃ



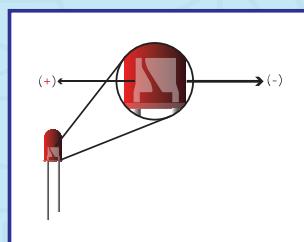
চিত্র ১.১- এলইডি লাইট

এখানে প্রতিটি লাইটে তড়িৎ প্রবাহের জন্য দুইটি পিন থাকে। এক পিনের মাথা লস্বা এবং আরেক পিনের মাথা অপেক্ষাকৃত সামান্য খাটো। যেই প্রান্ত লস্বা সেটি ধনাত্মক প্রান্ত এবং যেই প্রান্তটি খাটো সেখানে ব্যাটারীর ঋণাত্মক প্রান্ত লাগাতে হয়। তোমরা যদি ভুলে এই মাথা কেটে সমান করে ফেলো, তবুও বোঝার আরও দুটি উপায় আছে।



চিত্র ১.২- উপর থেকে দেখে এলইডি লাইট চেনার উপায়

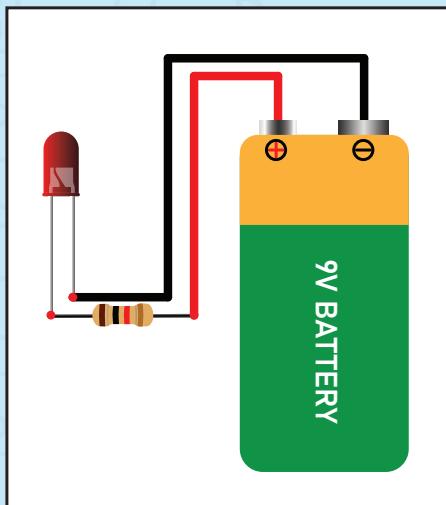
তোমরা যদি এলইডি লাইটের উপরের দিক হতে তাকাও, তবে লক্ষ্য করবে যে লাইট টি আসলে সম্পূর্ণ গোল নয়, বরং মনে হচ্ছে এক দিক থেকে অন্ন একটু গোলাকার অংশ কেটে সমান করে দেয়া হয়েছে। যে প্রান্তের কাছের অংশ সমান সেই প্রান্তটি ঋণাত্মক, এবং অপরটি ধনাত্মক। এছাড়াও আরেকটি উপায় আছে যেটি দেখে নিশ্চিতভাবে বলে দেয়া যায় যে এলইডি লাইটের কোন অংশ ধনাত্মক এবং কোন অংশ ঋণাত্মক।



চিত্র ১.৩ - ভেতরের ধাতব টুকরো দেখে এলইডি লাইট চেনার উপায়

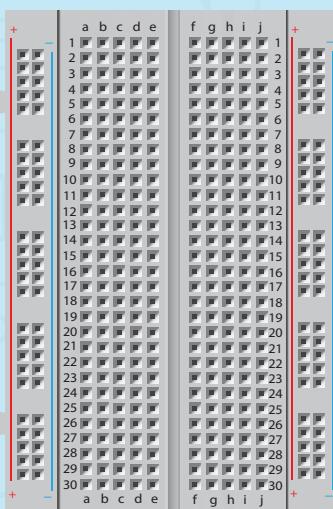
তোমরা যদি লাইটের দিকে ভালো করে লক্ষ্য করো তবে ভেতরে কিছু ধাতব অংশ দেখতে পাবে তাইনা? এখানে দেখলে মনে হবে দুটি ধাতব টুকরার মাঝে একটা চিড়ি ধরা ফাঁকা জায়গা আছে। ধাতব টুকরা দুটি কিন্তু একই আকারের নয়, একটা অপরটির তুলনায় বড় একটু। যে দিকে বড় টুকরা টি রয়েছে সেটি হলো ঋণাত্মক প্রান্ত এবং অপরটি হলো ধনাত্মক প্রান্ত

এবার এলইডি লাইটের সম্পর্কে আরও কিছু কথা বলি। সাধারণত বাজারে যেসকল এলইডি লাইট পাওয়া যায়, সেগুলো ১.৫ ভোল্ট থেকে শুরু করে ৩ ভোল্টে চলে। এর বেশি দিলে অতিরিক্ত তড়িৎপ্রবাহের জন্য পুড়ে যায়। সুতরাং তোমরা যখন একটা লাইটের সংযোগ দিবে, তখন লাইটের এক প্রান্ত ব্যাটারির সাথে সরাসরি সংযোগ থাকলেও অপর প্রান্তে একটা রোধ সিরিজ সংযোগে লাগাবে এবং সেই রোধের অপর প্রান্ত ব্যাটারীতে প্রবেশ করাবে। রোধটি আমাদের সার্কিটে তড়িৎপ্রবাহ কমিয়ে দেয় ফলে লাইট টি নষ্ট হয়ে যায়না। সাধারণত ৫ ভোল্টে থেকে শুরু করে ৯ ভোল্ট এর ব্যাটারীর সাথে সংযোগ দিলে ২২০ ওহম থেকে শুরু করে ১ কিলোওহম পর্যন্ত রোধ ব্যবহার করে যায়।

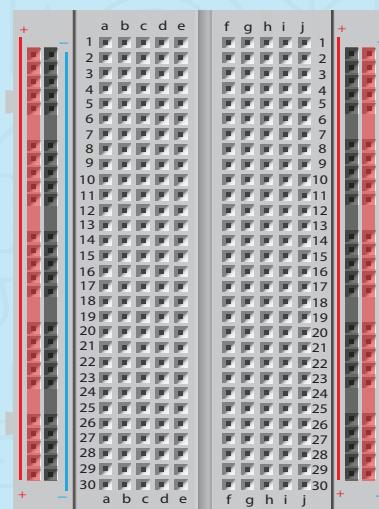


চিত্র ১.৪- ব্যাটারি দিয়ে এলইডি লাইট জ্বালানো উপায়

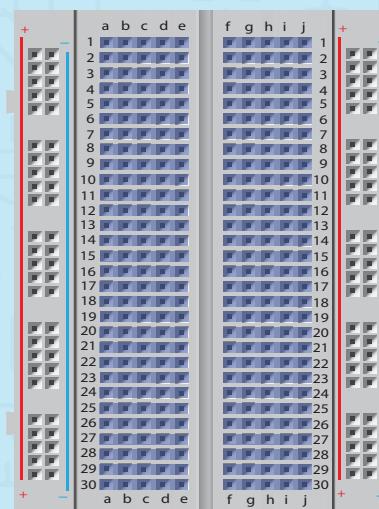
এবার ব্রেডবোর্ড সম্পর্কে একটু বর্ণনা দিয়ে নিই। আমরা যখন অস্থায়ীভাবে কোন কম্পিউনেন্ট এর কার্যকারিতা পরীক্ষা করে দেখতে চাই, তখন এই ধরনের বোর্ডে কম্পিউনেন্ট বসিয়ে পরীক্ষা করে দেখি।



চিত্র ১.৫- ব্রেডবোর্ড



চিত্র ১.৬- ব্রেডবোর্ডে পাওয়ার কানেকশন



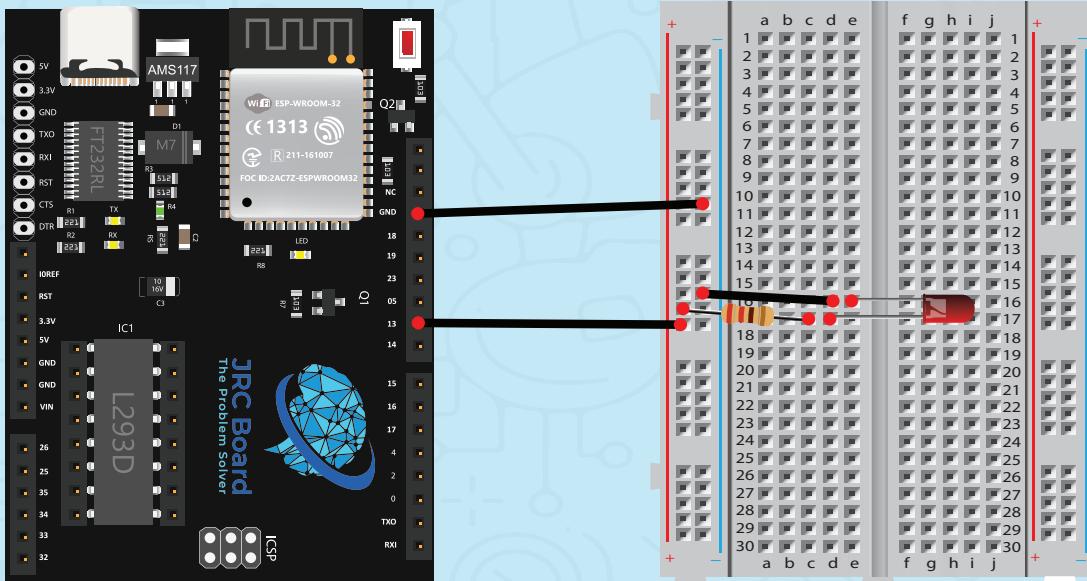
চিত্র ১.৭- ব্রেডবোর্ডে পিন কানেকশন

এই ব্রেডবোর্ডে কতগুলো ছিদ্র থাকে যেখানে কম্পিউনেন্ট বসানো যায়। প্রথমে আমরা দুই কিনারায় দুই সারি করে মোট চারটি সারি দেখতে পাবো যেখানে কেবল + এবং - চিহ্ন দিয়ে লাল লাইন এবং নীল লাইন টেনে রাখা। সাধারণত এগুলো উপর থেকে নিচে সবগুলো সারিই একসাথে সংযোগ দেয়া অবস্থায় থাকে। এর মানে তুমি যদি সর্ববামের একদম উপরের ছিদ্রে বসানো যে কথা, সর্ববামের একদম নিচের ছিদ্রে বসানো একই কথা। চিত্র-১.৬ এ দেখতে পাবে একই রঙ দিয়ে কোন কোন লাইনগুলো সংযুক্ত আছে তা লাল এবং কালো রঙ দিয়ে দেখানো হয়েছে। সাধারণত এই লাইনগুলোতে ব্যাটারির কিংবা একটা বিভিন্ন উৎসের সংযোগ দেয়া হয়। এরপর এখন থেকে যতগুলো কম্পিউনেন্ট এ বিভিন্ন দেয়া দরকার তা আমরা তারের সংযোগ এর মাধ্যমে দিয়ে থাকি। এটা একটা ভালো প্র্যাক্টিস কেননা অনেক সময় ব্যাটারি থেকে সরাসরি তার ভেতরে তার বসিয়ে দিলে আমরা বিভিন্ন নিয়ে দ্বিধায় পড়ে কম্পিউনেন্ট টি ভুলবশত নষ্ট করে ফেলতে পারি।

এবার আসি মাঝের ছিদ্রগুলো নিয়ে। এগুলো নির্দিষ্ট সারি এবং কলামে ভাগ করা আছে এবং কিনারায় এগুলো অক্ষর এবং সংখ্যা দিয়ে চিহ্নিত করাও আছে। সারিগুলো সংখ্যা দিয়ে চিহ্নিত করা এবং কলামগুলো ইংরেজি অক্ষর দিয়ে চিহ্নিত করা এখনে প্রতিটি সারির ছিদ্রগুলো নিজেদের মধ্যে সংযোগ দেয়া। যেমন a1 থেকে e1 এর মধ্যেকার ছিদ্রগুলোতে সংযোগ স্থাপন করা আছে। উদাহরণস্বরূপ আমরা যদি a1 এ একটা তারের সংযোগ দিই এবং d1 এ আরেকটা তারের সংযোগ দিই, তবে এই দুটি তার নিজেরা সংযুক্ত হয়ে যাবে। নিচে ছবিতে কোন কোন ছিদ্র নিজেরা সংযুক্ত তা চিত্র ১.৭ এ দেখতে পাবে।

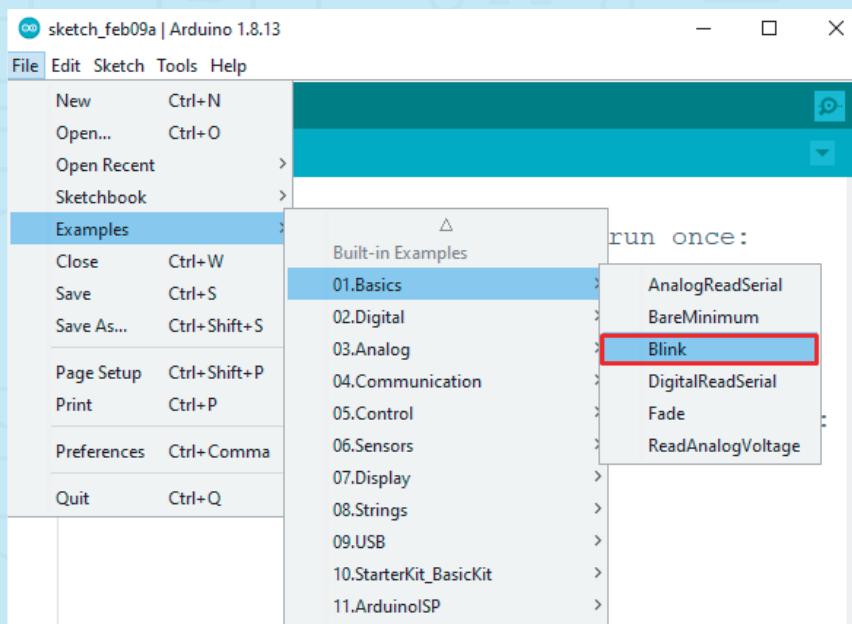
## JRC বোর্ড:

আশা করি সবাই কম্পোনেন্ট সম্পর্কে মোটামুটি আইডিয়া পেয়ে গেছো। এখন তোমাদের JRC বোর্ড নিয়ে প্রথম প্রোগ্রামিং করার পালা। লিঙ্কিং এলইডি এর জন্য আমরা এখানে বোর্ডের ডিজিটাল আউটপুট এর সহায়তা নিবো। আমরা জানি JRC বোর্ডে সবগুলো পিনেই ডিজিটাল আউটপুট পাওয়া সম্ভব যেখানে থেকে ধ্বনাত্মক ৩.৩ ভোল্ট বের হবে প্রোগ্রাম করা সাপেক্ষে। সুতরাং আমরা যদি JRC বোর্ডের যেকোন GPIO পিনের সাথে এলইডি লাইটের ধনাত্মক প্রান্ত এবং ঋণাত্মক প্রান্তের সাথে বোর্ডের GND পিন এর সংযোগ স্থাপন করি, সেক্ষেত্রে প্রোগ্রাম দিয়ে সেই পিনটি HIGH করে দিলে লাইটটি জ্বলে উঠবে এবং LOW করে দিলে লাইটটি নিন্তে যাবে। যেহেতু ৩.৩ ভোল্ট আউটপুট দিচ্ছে, তাই সাধারণত কোনপ্রকার রোধ স্থাপন না করলেও হয়। তবে চাইলে ১০০ ওহম থেকে ২২০ ওহম মানের রোধ সিরিজ সংযোগ দিয়ে রাখতে পারো। নিচের ছবি দেখে সার্কিট বানিয়ে নিতে পারো:



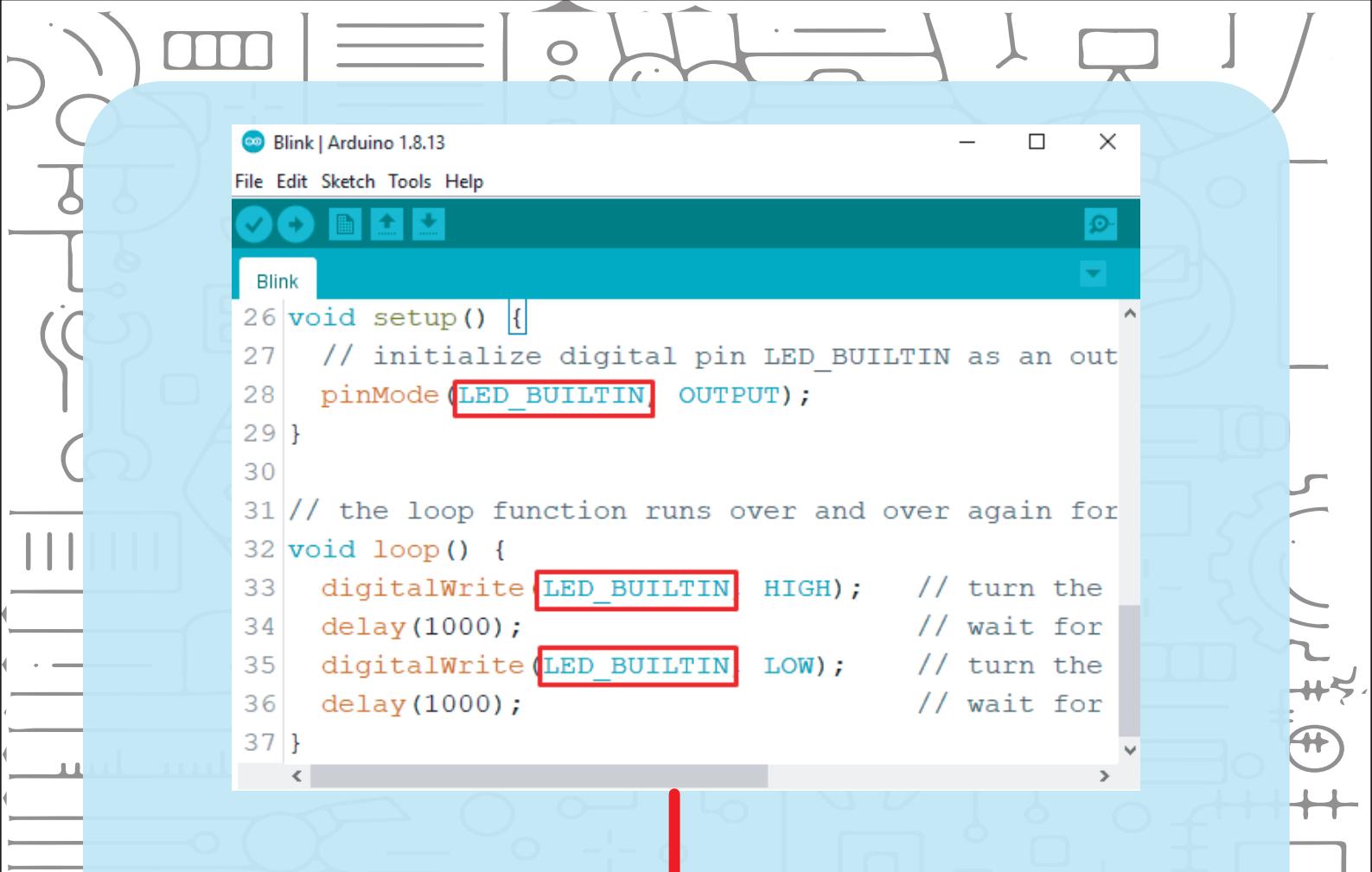
চিত্র ১.৮- JRC Board এর সাথে এলইডির কানেকশন

এবার কোডের পালা। তোমরা আরডুইনো IDE টি ওপেন করলে চাইলে এক্সাম্পল থেকে লিঙ্কিং কোড টি পেতে পারো। প্রথমে file এ ক্লিক করে এরপর examples এ ক্লিক করতে হবে। সেখানে তুমি basic অপশন এ ক্লিক করলে সেটার ভেতরে blink নামে অপশন খুজে পাবে।

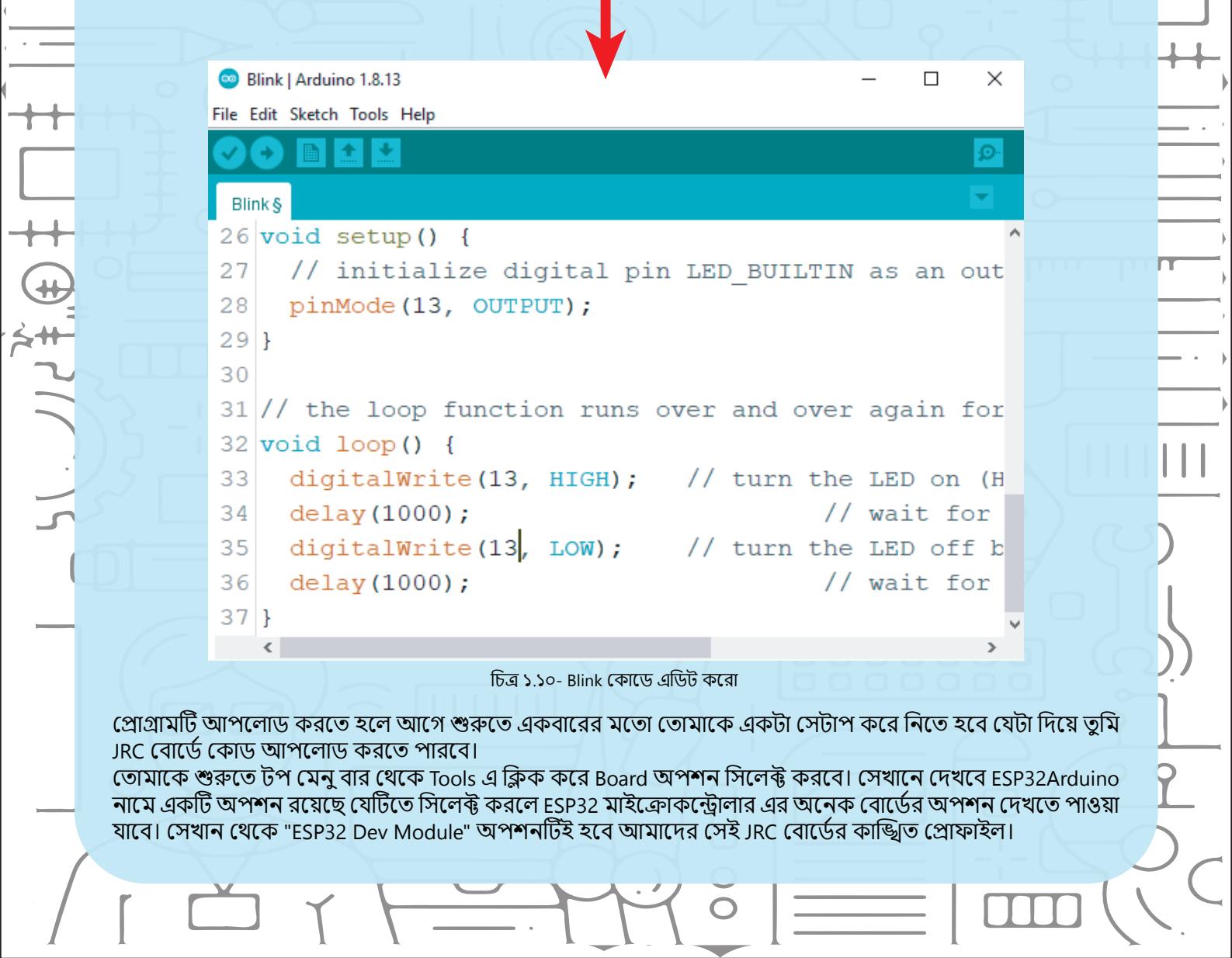


চিত্র ১.৯- Blink কোডের এক্সাম্পল খুজে বের করা

সেটিতে ক্লিক করলে নতুন একটি উইন্ডো ওপেন হবে যেখানে আরডুইনো তার বোর্ডে বিল্টইন প্রোগ্রামেবল এলইডি লাইট টি জ্বালানোর জন্য কোড করে রাখা। আমরা চাইলে এই কোডটিই ব্যবহার করতে পারি, কিন্তু এখানে আমাদের বোর্ডের মাইক্রোকন্ট্রোলার টি আলাদা হবার জন্য "LED\_BUILTIN" নামক কোন পিনের অস্তিত্ব পাওয়া যাবেনা। এর বদলে আমরা আমাদের বোর্ডের বিল্ট ইন প্রোগ্রামেবল লাইটটি জ্বালানোর জন্য সেই জাইয়গায় "13" লিখে দিলেই হয়ে যাবে।



```
Blink | Arduino 1.8.13
File Edit Sketch Tools Help
Blink
26 void setup() {
27 // initialize digital pin LED_BUILTIN as an output
28 pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again for
32 void loop() {
33 digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH)
34 delay(1000);                      // wait for a second
35 digitalWrite(LED_BUILTIN, LOW);     // turn the LED off (LOW)
36 delay(1000);                      // wait for a second
37 }
```

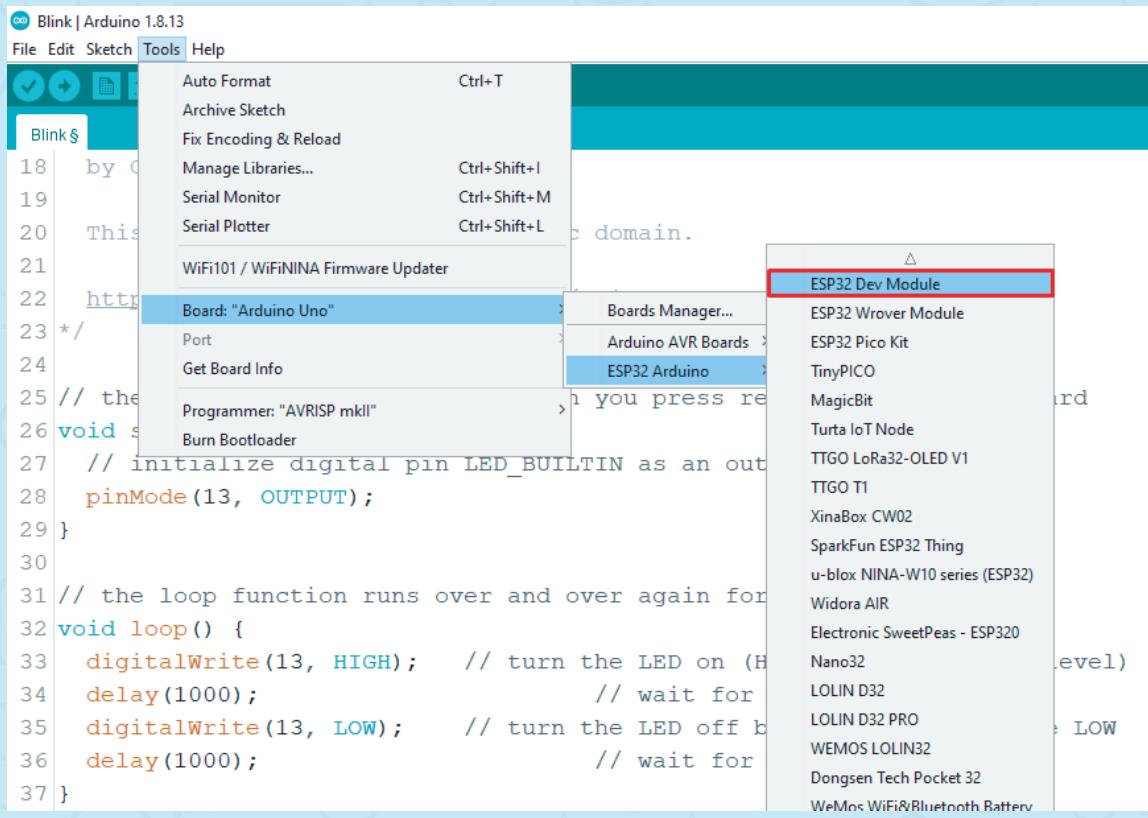


```
Blink | Arduino 1.8.13
File Edit Sketch Tools Help
Blink§
26 void setup() {
27 // initialize digital pin LED_BUILTIN as an output
28 pinMode(13, OUTPUT);
29 }
30
31 // the loop function runs over and over again for
32 void loop() {
33 digitalWrite(13, HIGH);      // turn the LED on (HIGH)
34 delay(1000);                      // wait for a second
35 digitalWrite(13, LOW);     // turn the LED off (LOW)
36 delay(1000);                      // wait for a second
37 }
```

চিত্র ১.১০- Blink কোডে এডিট করো

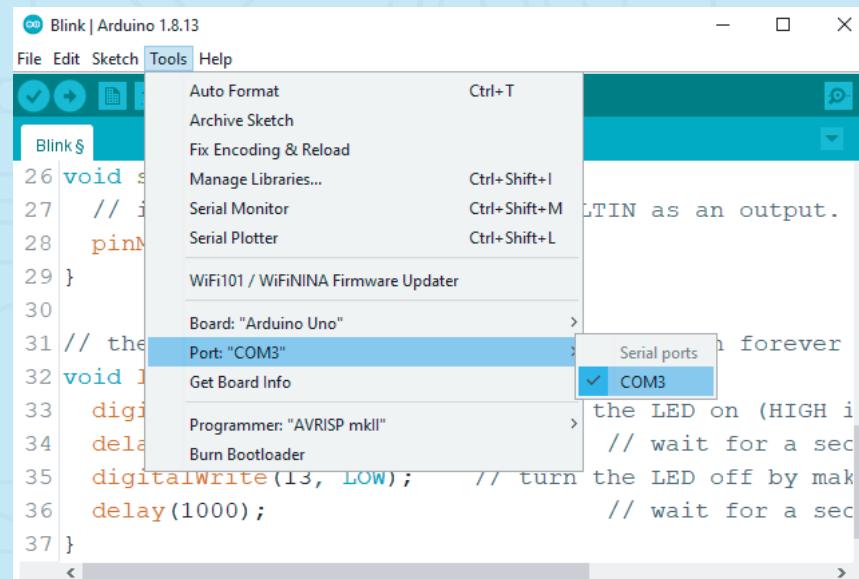
প্রোগ্রামটি আপলোড করতে হলে আগে শুরুতে একবারের মতো তোমাকে একটা স্টোপ করে নিতে হবে যেটা দিয়ে তুমি JRC বোর্ডে কোড আপলোড করতে পারবে।

তোমাকে শুরুতে টপ মেনু বার থেকে Tools এ ক্লিক করে Board অপশন সিলেক্ট করবে। সেখানে দেখবে ESP32Arduino নামে একটি অপশন রয়েছে যেটিতে সিলেক্ট করলে ESP32 মাইক্রোকন্ট্রোলার এর অনেক বোর্ডের অপশন দেখতে পাওয়া যাবে। সেখান থেকে "ESP32 Dev Module" অপশনটিই হবে আমাদের সেই JRC বোর্ডের কান্তিমুক্ত প্রোফাইল।



চিত্র ১.১১- সঠিক বোর্ড নির্বাচন করা

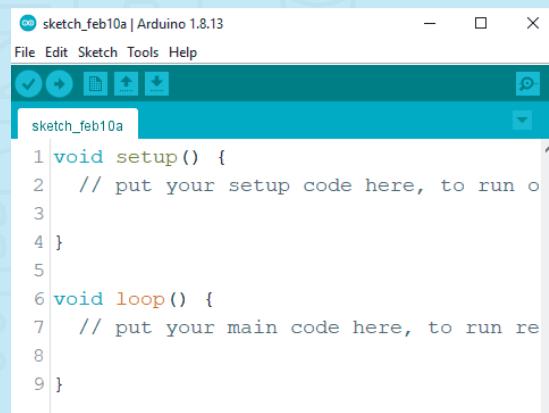
এরপর প্রোগ্রাম আপলোড করার আগে Tools অপশনের ভেতরে Board অপশনের ঠিক নিচেই Port নামের একটি অপশন থাকবে যেখানে ক্লিক করে তুমি কোন বোর্ডে আপলোড করবে সেটা সিলেক্ট করে নিতে হবে। যখনই তুমি JRC বোর্ড টি ইউএসবি টাইপ সি ক্যাবলের মাধ্যমের কম্পিউটারে কানেক্ট করলে Port অপশনের ভেতরে নিচের ছবির মতো অপশন দেখতে পাবে।



চিত্র ১.১২- সঠিক পোর্ট নির্বাচন করা

এবার আপলোড বাটনে চাপ দিলে কিংবা **ctrl + u** প্রেস করলেই প্রোগ্রামটি আপলোড হয়ে যাবে। তখন দেখতে পাবে তোমার বানানো সার্কিটে সেই লাইটটি ১ সেকেন্ড পর পর অন-অফ করছে বা ব্লিংক করছে। এভাবেই কিন্তু তোমরা এই বোর্ডে প্রথম কোডটি করে ফেললে।

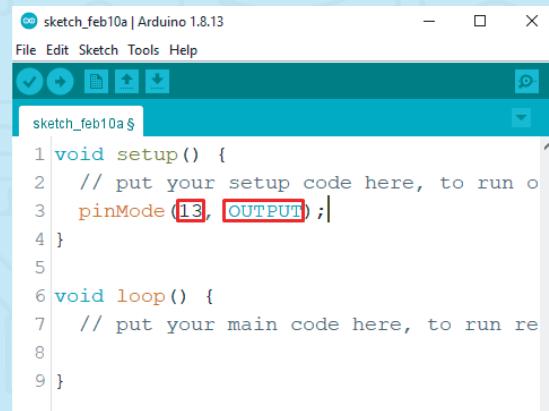
এবার চলো আমরা কোডের ব্যাখ্যায় আসি। তোমরা যারা Arduino IDE এবং পিন-আউট সিটেমে এর সাথে নতুন শুধুমাত্র তাদের জন্যেই এই লেখা। আমাদের JRC বোর্ডে মোট ২০ টি GPIO পিন রয়েছে যার মধ্যে ৩৪, ৩৫ ব্যাতিত সকল পিনে ডিজিটাল আউটপুট সম্ভব। এখন তুমি যদি কোন পিন থেকে ডিজিটাল আউটপুট পেতে চাও, তবে তোমাকে আগে সেই পিনটির ভূমিকা নির্ধারণ করে দিতে হবে। অর্থাৎ এই পিনটি কি কোন বিভব ইনপুট নিবে নাকি কোন বিভব আউটপুট দিবে সেটা নির্ধারণ করে দিতে হবে।



```
sketch_feb10a | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_feb10a
1 void setup() {
2 // put your setup code here, to run o
3 }
4
5
6 void loop() {
7 // put your main code here, to run re
8 }
9 }
```

চিত্র ১.১৩- কোডের বেসিক স্ট্রাকচার

এখনে দুটি ফাংশন দেখতে পাবে। একটি হলো `setup()` নামক ফাংশন এবং আরেকটি `loop()` নামক ফাংশন। `setup()` নামক ফাংশনে যা স্টেটমেন্ট লিখবে তা কেবল একবারই কাজ করবে। আর `loop()` ফাংশনের ভেতরে যা লিখবে তা সারাজীবনই বারবার চলতে থাকবে। এখন চিন্তা করে দেখো, তোমাকে যদি কোন একটি ডিজিটাল পিনের ভূমিকা নির্ধারণ করে দিতে হয়, তবে কি সেটা বার বার করতে হবে? অবশ্যই না, কেবল একবার বলে দিলেই হয়ে যায় যে পিনটি কি হিসেবে কাজ করবে। সেক্ষেত্রে এর জন্য যে কোড বা স্টেটমেন্ট লিখতে হবে সেটা `setup()` ফাংশনের ভেতরে লিখলেই হয়ে যাবে। আমাদের সার্কিট অনুযায়ী এলইডি লাইটের ধনাত্মক প্রান্ত বোর্ডের ডিজিটাল ১৩ নাম্বার পিনের সাথে লাগানো তাইনা? সেক্ষেত্রে ১৩ নাম্বার পিন কে কিন্তু আউটপুট হিসেবে আকে নির্ধারণ করে দিতে হবে। এর জন্য আমরা `pinMode()` নামক একটি ফাংশন ব্যবহার করি।

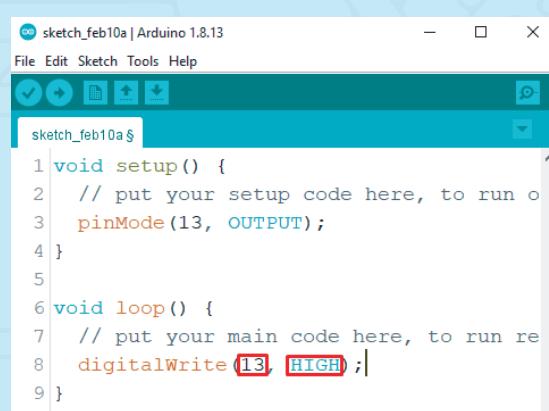


```
sketch_feb10a | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_feb10a §
1 void setup() {
2 // put your setup code here, to run o
3 pinMode(13, OUTPUT);|
4 }
5
6 void loop() {
7 // put your main code here, to run re
8 }
9 }
```

চিত্র ১.১৪- `pinMode` ফাংশনের ব্যবহার

এখনে ফাংশনটির ব্র্যাকেটের ভেতরে প্রথমে পিন নাম্বার লিখতে হয়, পরে একটি কমা দিয়ে সেটির ভূমিকা লিখতে হয়। যদি পিনটি আউটপুট হিসেবে কাজ করাতে চাও তবে "OUTPUT" লিখবে। যদি ইনপুট হিসেবে কাজ করাতে চাও তবে "INPUT" লিখবে। লেখার সময় লক্ষ্য রাখবে যে সবগুলো অক্ষর ঘাতে বড় হাতের হয়। তবেই এই ফাংশনটি ঠিকমতো কাজ করবে।

এবার সেই পিনে আউটপুট কি হবে তা কমান্ত দেবার পালা। এই কাজ টা আমরা `loop()` ফাংশনের ভেতরে করে ফেলতে পারি। কোন পিন থেকে ডিজিটাল আআউটপুট দিতে গেলে `digitalWrite()` নামক একটি ফাংশন লিখতে হয়।



```
sketch_feb10a | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_feb10a §
1 void setup() {
2 // put your setup code here, to run o
3 pinMode(13, OUTPUT);
4 }
5
6 void loop() {
7 // put your main code here, to run re
8 digitalWrite(13, HIGH);|
9 }
```

চিত্র ১.১৫- `digitalWrite` ফাংশনের ব্যবহার

এখানে digitalWrite() ফাংশনের ব্র্যাকেটের ভেতরে শুরুতে কোন পিনে কমান্ড দিতে চাও সেটার নাম্বার লিখতে হয়, এরপর সেই পিনে কি ধরনের কমান্ড দিতে চাও সেটা লিখতে হয়। যদি পিন থেকে 3.3 ভোল্ট বের করতে চাও, তবে HIGH অথবা 1 লিখে দিবে, যদি 0 ভোল্ট বের করতে চাও, তবে সেটা LOW অথবা 0 লিখে দিবে। আপাতত আমরা digitalWrite(13, HIGH) লিখেছি, যার ফলে সেই 13 নাম্বার পিন থেকে 3.3 ভোল্ট বের হবে এবং সার্কিটে থাকা লাইট টি জ্বালিয়ে দিবে। এছাড়াও তোমরা বোর্ডে থাকা সেই প্রোগ্রামেবল লাইট টিও জ্বলতে দেখবে, কেননা আমরা বাহিরে যে সার্কিট তৈরী করেছি, সে একই সার্কিট বোর্ডের ভেতরেই 13 নাম্বার পিনের সাথে যুক্ত অবস্থায় স্থাপন করা আছে। সুতরাং তোমরা চাইলে বাহিরে এক্সট্রা কোন সার্কিট না তৈরী না করেই বোর্ডের এলাইডি ব্যবহার করে কোড পরীক্ষা করতে পারো। বোর্ড টি ঠিক আছে কিনা সেটাও আমরা এভাবে পরীক্ষা করতে পারি।

এখন যদি তুমি digitalWrite(13, LOW) অথবা digitalWrite(13, 0) লিখে কোড আপলোড দাও, তবে দেখবে যে লাইটটি বন্ধ হয়ে গেছে।

এবার আরও সামনে আগাই। নিচের কোডটি লক্ষ করোঃ

```

sketch_feb10a | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_feb10a §
1 void setup() {
2   // put your setup code here, to run o
3   pinMode(13, OUTPUT);
4 }
5
6 void loop() {
7   // put your main code here, to run re
8   digitalWrite(13, HIGH);
9   delay(1000);
10  digitalWrite(13, LOW);
11  delay(1000);
12 }

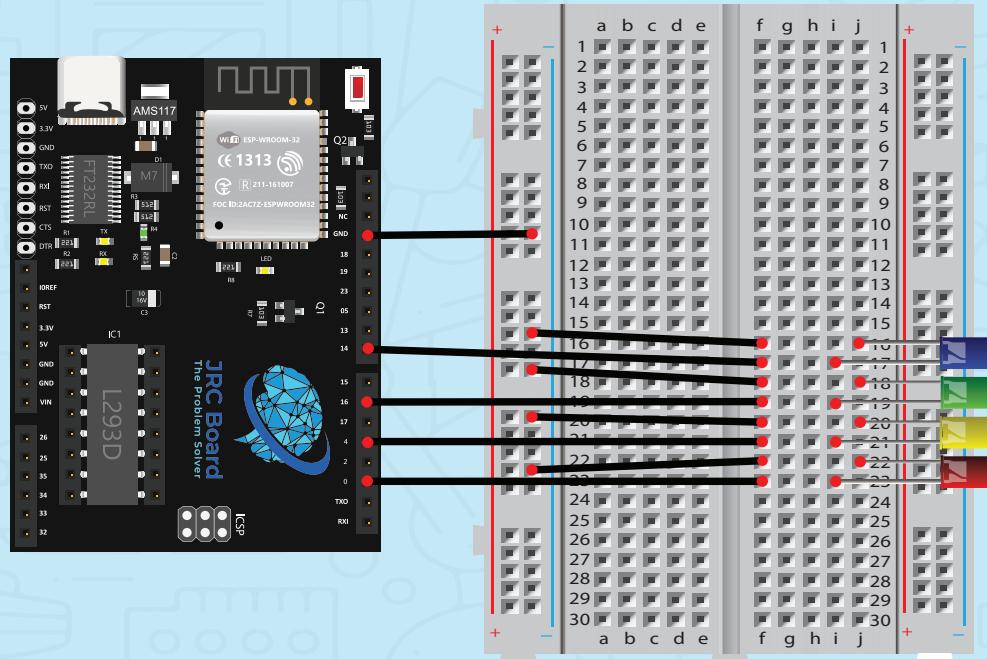
```

চিত্র ১.১৬- পুরো কোডটি পর্যবেক্ষণ করো

এখানে কোডে একটি নতুন ফাংশন দেখতে পাচ্ছো যার নাম delay()। এটির ব্যাপারে ব্যাখ্যার আগে বলে রাখি JRC বোর্ড এখানে কীভাবে কাজ করছে। এটি প্রথমে setup() ফাংশন এর ভেতরে যা আছে তা উপর থেকে নিচে পড়ে নেয়। এরপর loop() ফাংশনে প্রবেশ করে। এখানে সে প্রথমে 8 নাম্বার লাইন থেকে একের পর এক স্টেটমেন্ট কাজ করতে শুরু করে এবং যখন দেখতে পায় যে 11 নাম্বার লাইনের পর আর কোন স্টেটমেন্ট থাকি নেই, তখন আবার সে ঘুরে এসে 8 নাম্বার লাইন থেকে কাজ করা শুরু করে দেয়। এবং এই কাজ টা সে খুব দ্রুত করে ফেলতে পারে। যদি আমরা কোন জ্যায়গায় delay() ফাংশন টা ব্যবহার করি, তবে প্রোগ্রাম টা সেই জ্যায়গায় আটকে থাকে। delay() ফাংশনের ভেতরে নিউমেরিক ভ্যালু লেখা যায় যা আমাদের মিলিসেকেন্ড এককে চিন্তা করে নিতে হয়। মানে ধরো, আমরা যদি লিখি delay(1000), তবে কোডের কাজ সেই জ্যায়গায় 1000 মিলিসেকেন্ড, অর্থাৎ 1 সেকেন্ড আটকে থাকে। এরপর আবার সে পরবর্তী স্টেটমেন্ট পড়া শুরু করে দেয়। এখন লক্ষ্য করে দেখ যে 8 নাম্বার লাইনে আমরা লাইট কে জ্বালানোর কমান্ড দিয়েছিলাম। এর পরের স্টেটমেন্ট delay(1000) হবার কারণে প্রোগ্রাম টা এখানে 1 সেকেন্ড আটকে থাকবে। ততক্ষণ আগে যা কমান্ড দিয়েছিলাম তা বহাল থাকবে। অর্থাৎ লাইট টি জ্বলে থাকবে। এর পরের 10 নাম্বার লাইনে আমরা লাইট বন্ধ করার কমান্ড দিয়েছি। তখন লাইট টি বন্ধ হয়ে যাবে। এরপর আবার delay(1000) থাকার কারণে কন্ট্রোলার এখানে এক সেকেন্ড অপেক্ষা করার পর পুরো লুপ তা আবার নতুন করে কাজ করতে শুরু করে দিবে।

তাহলে পুরো ঘটনা একেবারে চিন্তা করলে বুঝে যাচ্ছ যে এখানে একটা লাইট কেবল 1 সেকেন্ড পর পর অন অফ হতে থাকবে অর্থাৎ বিল্ক করতে থাকবে। এইভাবে তুমি এই প্রথম প্রজেক্ট টি আয়ত্ত করে নিলে!

এই ডিজিটাল আউটপুট নিয়ে চাইলে আরও মজা করতে পারো। চলো চিত্র ১.১৭ এর মতো একটা সার্কিট তৈরী করিঃ



চিত্র ১.১৭- একের অধিক এলইডি দিয়ে সার্কিট তৈরী

এখানে আমরা ব্রেডবোর্ডে পর পর ৪ টি লাইট বসিয়েছি যাদের পিনগুলো অবশ্যই ভিন্ন ভিন্ন সারিতে স্থাপন করা। এদের ধ্বনাত্মক প্রান্ত গুলো একসাথে যুক্ত করে GND পিনে সংযোগ দেই এবং ধ্বনাত্মক প্রান্তগুলো আলাদা আলাদা ডিজিটাল পিনে সংযোগ দেয়া আছে। যেমন লাল রঙের লাইটের ধ্বনাত্মক প্রান্ত ০ নাম্বার পিনে, হলুদ রঙের জন্য ৪ নাম্বার পিন, সবুজ রঙের জন্য ১৬ নাম্বার পিন এবং নীল রঙের জন্য ১৪ নাম্বার পিনের সাথে ধ্বনাত্মক প্রান্তের সংযোগ দেয়া হয়েছে জাম্পার ওয়্যার এর মাধ্যমে। এখন নিচের কোড টি আপলোড করে দেখিঃ

```
#define red 0
#define yellow 4
#define green 16
#define blue 14

void setup(){
    pinMode(red, OUTPUT);
    pinMode(yellow, OUTPUT);
    pinMode(green, OUTPUT);
    pinMode(blue, OUTPUT);
}

void loop(){
    digitalWrite(red, HIGH); delay(500); digitalWrite(red, LOW);
    digitalWrite(yellow, HIGH); delay(500); digitalWrite(yellow, LOW);
    digitalWrite(green, HIGH); delay(500); digitalWrite(green, LOW);
    digitalWrite(blue, HIGH); delay(500); digitalWrite(blue, LOW);
}
```

চিত্র ১.১৮- একের অধিক এলইডি জ্বালানোর কোড

কোডে কি লেখা আছে সে ব্যাপারে অন্ন একটু ব্যাখ্যা করি। শুরুতেই red, green, yellow, blue এই চার নামে প্লেবাল কন্ট্রোল ভ্যারিয়েবল ডিক্রেয়ার করা আছে যেগুলো মান হিসেবে পিন নাম্বার দিয়ে দেয়া আছে যে পিনে প্রতিটি পিনের ধ্বনাত্মক প্রান্ত সংযুক্ত করা আছে। এটার সুবিধা হলো তুমি যদি লাল লাইটের ধ্বনাত্মক প্রান্ত অন্য কোন পিনে লাগাতে চাও, সেক্ষেত্রে পুরো কোডে পরিবর্তন না এনে শুধু শুরুতেই red ভ্যারিয়েবলটির মান পরিবর্তন করে দিলেই হয়। কেবল এই ব্যাপারে লক্ষ্য রাখবে যে এখানে কোডের বাকি অংশে যত জায়গায় পিন নাম্বার লেখার কথা ছিলো, সেখানে নাম্বার এর বদলে red অথবা যে নামে ভ্যারিয়েবল ডিক্রেয়ার করেছো, সেই নাম দিয়ে দিবে।

void loop() এর ভেতরে লক্ষ্য করলে বুঝতে পারবে যে এখানে প্রথমে প্রতিটি লাইট জ্বালানো হচ্ছে, এরপর ৫০০ মিলিসেকেন্ড অপেক্ষা করছে, এরপর সেই লাইট টি বন্ধ করে দিয়ে নতুন আরেকটি লাইট জ্বালাচ্ছে। এভাবে একটি সুন্দর এলইডি প্যাটার্ন দেখা যাবে কোড আপলোড দেবার পর।