

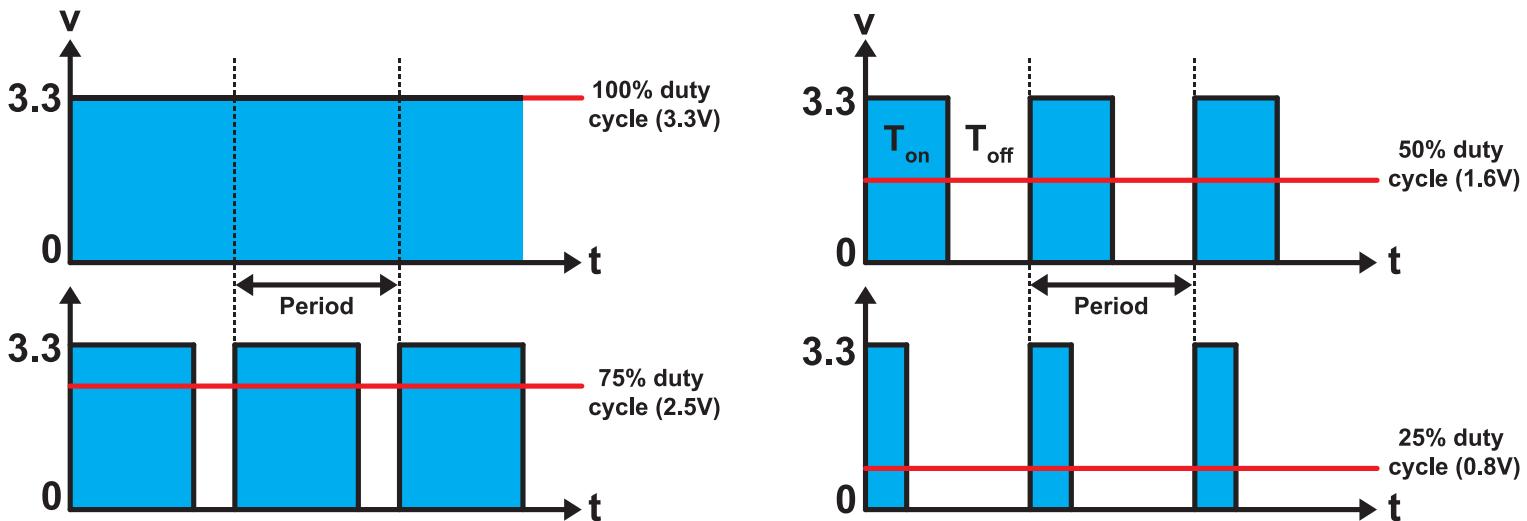
৫/ এনালগ আউটপুট এবং ডিজিটাল লাইট:

প্রয়োজনীয় উপকরণ:

- ১/ ব্রেডবোর্ড
- ২/ জাম্পার ওয়্যার
- ৩/ JRC বোর্ড
- ৪/ এলইডি লাইট
- ৫/ একটি ২২০ ওহম রেজিস্ট্র

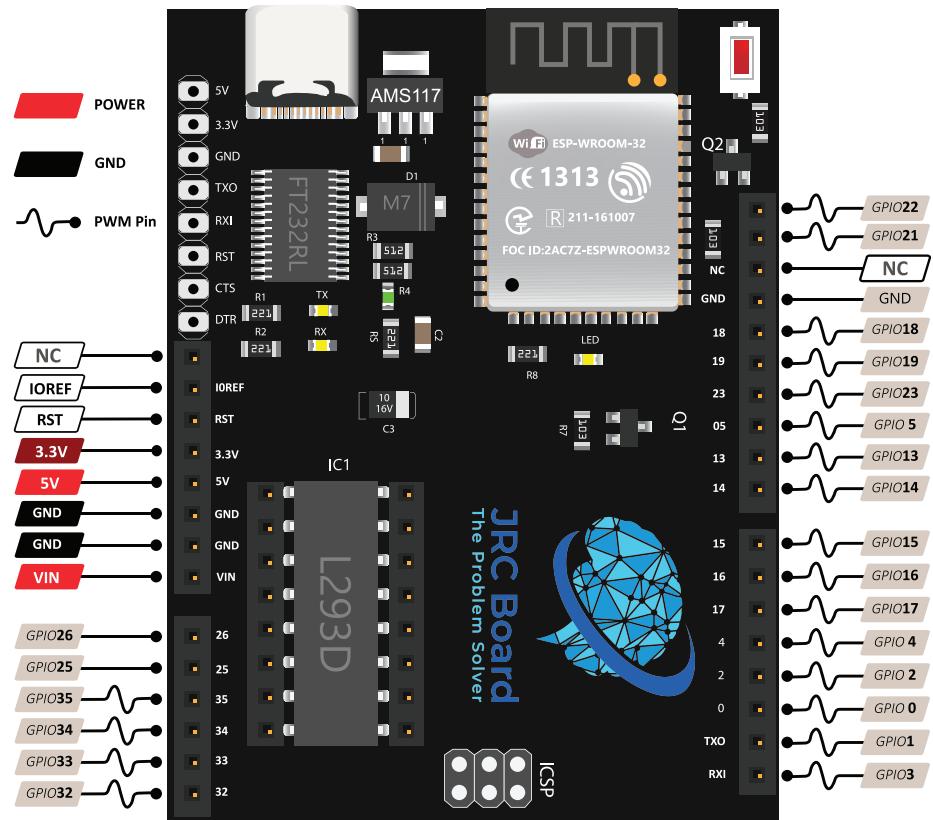
বর্ণনা: আমরা ডিজিটাল আউটপুট সম্পর্কে শিখে এসেছি তাইনা? সেখানে JRC Board থেকে হয় ৩.৩ভোল্ট আউটপুট পেতাম নয়তোৰা ০ ভোল্ট আউটপুট পেতাম। কিন্তু চাইলেই আমরা এর মাঝামাঝি যেকোন ভোল্টেজ পেতে পারি। সেটা পাবার উপায় হলো পিন এ পালস উইডথ মড্যুলেশন (সংক্ষেপে PWM) সিস্টেম প্রয়োগ করা। ব্যাপার টা কিরকম? তার একটা ব্যাখ্যা দিই:

ধরো তুমি রুমে ফ্যানের সুইচ টা অন করলে। সেক্ষেত্রে কি দেখতে পাবে? ফ্যান টা আন্তে আন্তে ঘুরতে শুরু করে একসময় ফুল স্পিডে উঠে গেছে তাইনা? এখন সুইচ টা বন্ধ করে দাও। সেক্ষেত্রে দেখতে পাবে ফ্যান টা আন্তে আন্তে গতি কমতে কমতে একটা সময় থেমে যাবে। এখন যদি এমন করো যে সুইচ টা এক সেকেন্ড পর পর অন-অফ করছো, সেক্ষেত্রে কি দেখতে পাবে? সেক্ষেত্রে দেখা যাবে ফ্যানটি না পূর্ণ গতিতে চলতে পারছে, আবার না পুরোপুরি থেমে যেতে পারছে...মোটামুটি অর্ধেক গতিতে চলছে। এই যে বার বার অন-অফ করে ফ্যান কে তুমি অর্ধেক গতিতে চালাচ্ছো, এটিই হচ্ছে PWM সিস্টেম এর মূলনীতি। এখানে একটা পিন কে উচ্চ কম্পাঙ্কে অন-অফ করে চাইলে তার বিভব ০ থেকে ৩.৩ ভোল্ট এর মাঝামাঝি যেকোন ভোল্টেজে রূপান্তর করা যায়। অন থাকার সময় এবং অফ থাকার সময়ের অনুপাতের উপর নির্ভর করে এটি কত উচ্চ বিভব পাবে। যেমন তুমি অন-অফ এর অনুপাত ৫০-৫০ রাখলে সেক্ষেত্রে পিনের বিভব মোট বিভবের অর্ধেক দেখতে পাবে। যদি অন-অফ সময়ের অনুপাত ৭৫:২৫ হয়, তবে দেখতে পাবে যে পিনে মোট বিভবের ৭৫% বিভব পাচ্ছে। এই ব্যাপার টিকে ডিউটি সাইকেলে সর্বোচ্চ বিভব পাবে, ০% ডিউটি সাইকেলে সর্বনিম্ন বিভব পাবে।



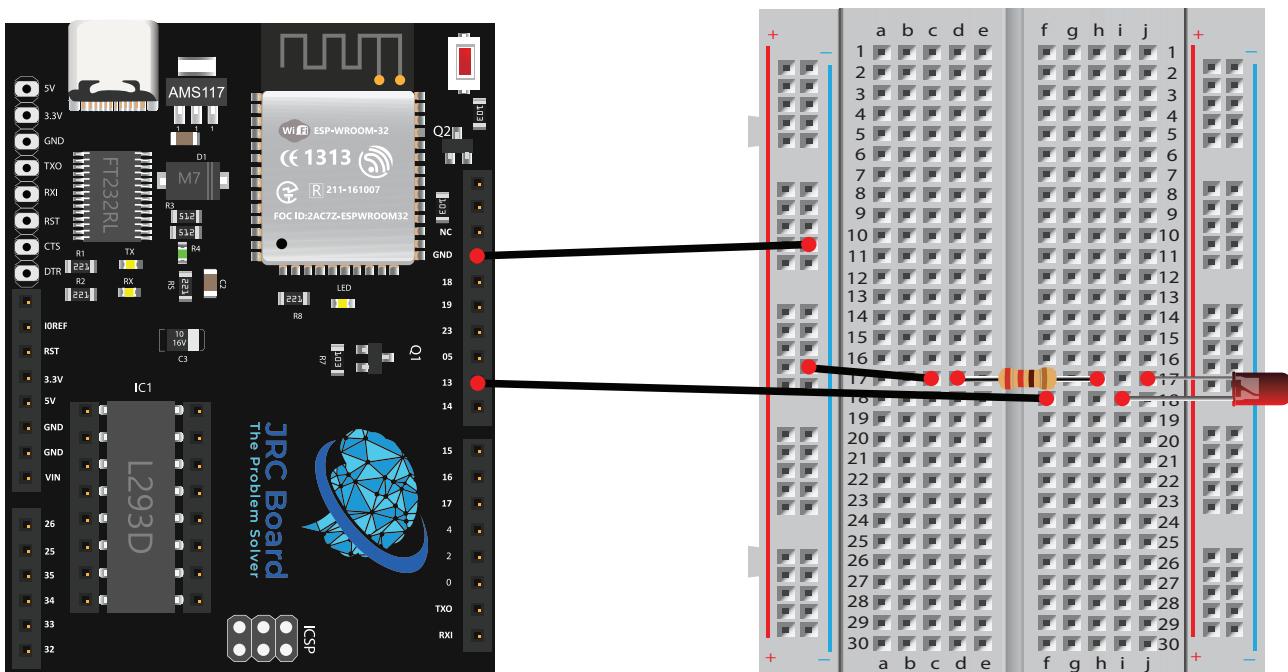
সুতরাং বুঝতেই পারছো কিভাবে ডিজিটাল পিন থেকেও টাইমার এর খুব সুন্দর সিস্টেম ব্যবহার করে আমরা চাইলে ০ থেকে ৩.৩ এর মধ্যে যেকোন ভোল্টেজ পেতে পারি। কিন্তু এখানে কিছু কথা আছে। প্রথমত এখানে কেবল পিন টাকে একটানা লো কিংবা একটানা হাই স্টেটে রাখলে কিন্তু ভ্যারিয়েবল ভোল্টেজ বা এনালগ আউটপুট পাওয়া সম্ভব নয়। একে সবসময় একটা নির্দিষ্ট কম্পাঙ্কে পরিচালনা করে যেতে হবে যেখানে অন এবং অফ থাকার সময়কালের অনুপাতের উপর নির্ভর করে কত ভোল্টেজ পাবে সেটা নির্ধারণ করে দেয়া যায়। তো এই যে একটা নির্দিষ্ট কম্পাঙ্কে পরিচালনা করছে, তার এই পালস কিন্তু কাউকে না কাউকে দেখতাল করতেই হবে তাইনা? যদি ESP32 এর কোর প্রসেসর এই কাজে ব্যাস্ত থাকে, তাহলে তো সাধারণ কোন কাজ করতে পারবেনা। তার জন্য রয়েছে আলাদা টাইমার চ্যানেল। এই চ্যানেল এর সাথে পিনের সংযোগ ঘটিয়ে এরপর কমান্ড দিলেই ব্যাস! বাকি কাজ তারাই করে নিবে!

আপাতত জনপ্রিয় যে মাইক্রোকন্ট্রোলার আরডুইনো, তাতেও কিন্তু একইভাবে PWM আউটপুট দেয়া হয়। কিন্তু এখানে ঘটনা হলো আরডুইনো উনো তে এরকম PWM আউটপুট দেবার মতো সক্ষম পিন রয়েছে কেবল ৬টি এবং তাদের কে নিয়ন্ত্রণ করার জন্য টাইমার চ্যানেল রয়েছে মাত্র ৩টি। এর মানে প্রতি জোড়ায় মাত্র একটা টাইমার চ্যানেল থাকে। এটার অসুবিধা কি সে ব্যাপারে পরে আলোচনা করা হবে। আপাতত আমরা বেসিক এর দিকে লক্ষ্য করি। নিচের ছবিতে JRC বোর্ডের PWM আউটপুট দিতে সক্ষম পিনের ডায়াগ্রাম দেখানো হচ্ছে:



এখানে পরিষ্কারভাবে দেখতে পাচ্ছা যে JRC বোর্ড এর প্রায় সব GPIO পিনই PWM হিসেবে আউটপুট প্রদান করা যায়। এই আউটপুট প্রদানের জন্য টাইমার চ্যানেল রয়েছে মোট ১৬ টি। অর্থাৎ তুমি ১৬ টি আলাদা আলাদা ডিভাইসে আলাদা PWM সিগ্নাল দিতে পারবে। আবার চাইলে তুমি সবগুলো পিন থেকে একই সিগ্নাল নিয়েও আসতে পারবে! এটা JRC বোর্ডের ডুয়াল কোর প্রসেসর থাকার একটা বড় সুবিধা! ইচ্ছামতো পিনে তুমি PWM সিগ্নাল পেতে পারো এবং কোনপ্রকার চিন্তা ছাড়াই সার্কিট ডিজাইন করতে পারবে!

চলো আমরা একটা সার্কিট বানিয়ে এই ব্যাপার টি পরীক্ষা করি। আমরা পুর্বে ব্লিংকিং প্রজেক্টের জন্য যে সার্কিট টি ব্যবহার করেছিলাম, তবহু সেই সার্কিট টিই চাইলে এখানে ব্যবহার করা যায়।



এবার কোড লিখবার পালা। PWM আউটপুট এর জন্য প্রথমেই আমাকে টাইমার চ্যানেল অবশ্যই সেট করতে হবে। JRC বোর্ডে মোট ১৬ টি টাইমার চ্যানেল রয়েছে যার প্রত্যেকটিকে আলাদা আলাদাভাবে সেট করা যায়। এবং এই টাইমার চ্যানেল এর নাম্বার ০ থেকে শুরু হয়ে ১৫ পর্যন্ত যায়। নিচের কোডে কিভাবে টাইমার চ্যানেল সেট করা হচ্ছে দেখোঃ

```
void setup(){  
    ledcSetup(0,1000,8);  
}  
  
    ↗  
    ↗  
    ↗
```

যেহেতু টাইমার চ্যানেল সেট করতে হবে তাই তার জন্য দরকারী কোড `setup()` ফাংশনের ভেতরে লিখে নেয়া ভালো। টাইমার চ্যানেল সেট করার জন্য আমরা `ledcSetup()` ফাংশনটি ব্যবহার করি। এর ভেতরে ৩ টি প্যারামিটার দিতে হয়। প্রথমে যে চ্যানেল নিয়ে আমরা কাজ করতে চাই সেটি দিতে হয়। এরপর কমা দিয়ে সে চ্যানেল কত কম্পাক্ষে আমরা **PWM** আউটপুট চাচ্ছি সেটি দিতে হয়। এটা কিন্তু JRC বোর্ডের একটি বড় সুবিধা যা কিনা অন্য মাইক্রোকন্ট্রোলারে পাওয়া যায়না! সাধারণত এখানে ১২কিলোহার্টজ পর্যন্ত কম্পাক্ষ আউটপুট দিতে সক্ষম। তবে রেজিলুশন কমিয়ে দিলে আরও বেশি কম্পাক্ষে আউটপুট সন্তুষ্ট। এরপর কমা দিয়ে সবশেষে লিখতে হয় কত বিট রেজিলুশনে এই চ্যানেল নিয়ন্ত্রন করতে চাচ্ছি সেটা লিখতে হয়। উদাহরণস্বরূপ আমি এখানে ৮ লিখেছি এর মানে এটি ৮বিট রেজিলুশন প্রদান করবে। অর্থাৎ এটি ০ থেকে ৩.৩ ভোল্ট কে ২^৮ ভাগে বা ২৫৬ ভাগে ভাগ করে ফেলবে। এবং আমরা ০ থেকে ২৫৫ পর্যন্ত যে ভ্যালু পরবর্তীতে দিবো সে অনুযায়ী সেই চ্যানেলটি ভোল্টেজ প্রদান করবে। ০ দিলে ০ ভোল্ট দিবে, ২৫৫ দিলে ৩.৩ ভোল্ট দিবে, ১২৮ দিলে অর্ধেক ভোল্টেজ দিবে - এরকম। JRC বোর্ড সর্বোচ্চ ১৬ বিট পর্যন্ত রেজিলুশন সাপোর্ট করে। যত বেশি রেজিলুশন দিবে, তত বেশি সূক্ষ্ম ভোল্টেজ পাওয়া সন্তুষ্ট হবে, তবে এক্ষেত্রে বেশি কম্পাক্ষ ব্যবহার করতে পারবেনা আর। যদি তোমার প্রজেক্টে খুব বেশি সূক্ষ্ম হিসেব নিকেশ বা ভোল্টেজ দরকার না হয়, তবে এই রেজিলুশন কমিয়ে রাখাই ভালো।

এই তো গেলো চ্যানেল সেটাপ এর কাজ। এখন এই চ্যানেল টা আমরা কোন পিনে ব্যবহার করবো সেটাও নির্ধারণ করে দিতে হবে। নিচের কোডের দিকে লক্ষ্য করোঃ

```
void setup(){
    ledcSetup(0,1000,8);
    ledcAttachPin(13,0);
}
```

এখানে `ledcAttachPin()` ফাংশনের মাধ্যমে আমরা সেই পিন নির্ধারণের কাজ টি করে থাকি। এখানে ভেতরে দুইটি প্যারামিটার এর প্রথমটায় পিন নামার এবং পরের টায় চ্যানেল নামার বসাতে হয়। ব্যাস, এতেই পিন সেটাপ হয়ে যায়! চাইলে আমরা একই চ্যানেল এর সাথে একাধিক পিন যুক্ত করতে পারি এবং সেক্ষেত্রে ঐসকল পিনে হ্রবহু একই সিগ্নাল পাওয়া সন্তুষ্ট হবে!

এবার চলো দেখে নিই কিভাবে ভোল্টেজ নিয়ন্ত্রন করতে হবে। তুমি দেখতে পাচ্ছো যে আমরা চ্যানেল সেটাপ এর সময়ে ৮ বিট রেজিলুশন নির্ধারণ করেছিলাম সুতরাং আমরা ২৫৫ ভ্যালুর জন্য ৩.৩ ভোল্ট পাবো সেটা মনে রাখতে হবে। নিচের কোডের দিকে লক্ষ্য করোঃ

```
void setup(){
    ledcSetup(0,1000,8);
    ledcAttachPin(13,0);
    ledcWrite(0,255);
}
```

এখানে `ledcWrite()` ফাংশনের মাধ্যমে তুমি সেই ভোল্টেজ প্রদানের কাজ টি করে থাকবে। এর ভেতরে শুরুতে কোন চ্যানেল নিয়ন্ত্রন করছো সেটির নামার লিখে দিবে, এরপর সেটায় কত মান দিতে চাও সেই মান টি লিখে দিবে। যেমন আমরা যেহেতু ৮ বিট রেজিলুশন ব্যবহার করছি, সেক্ষেত্রে এখানে ২৫৫ লিখে দিলেই ৩.৩ ভোল্ট পেয়ে যাবে, এবং তখন লাইট টি পূর্ণ উজ্জ্঳লতায় জ্বলে উঠবে। যদি ২৫৫ এর জায়গায় ১২৮ লিখে দিই, সেক্ষেত্রে লাইট টি অর্ধেক উজ্জ্঳লতায় জ্বলে উঠবে। এভাবে চাইলে আমরা **PWM** পিনের মাধ্যমে লাইটের উজ্জ্঳লতা নিয়ন্ত্রন করতে পারি। শুধু এই ক্ষেত্রেই নয়, যেসকল মডিউলে এনালগ সিগ্নাল দিয়ে নিয়ন্ত্রন করতে হয় (যেমন মটর ড্রাইভার), সেখানেও কিন্তু আমরা এই **PWM** সিস্টেম ব্যবহার করে কাজ চালাতে পারি!

চলো এবার এই লাইটটি নিয়ে আরেকটু মজা করা যাক! নিচের কোডের দিকে লক্ষ্য করোঃ

```
void setup(){
    ledcSetup(0,1000,8);
    ledcAttachPin(13,0);
}
void loop(){
    for(int i=0;i<=255;i++){
        ledcWrite(0,i); delay(10);
    }
    for(int i=255;i>=0;i--){
        ledcWrite(0,i); delay(10);
    }
}
```

এই কোডে ভোল্টেজ নিয়ন্ত্রণ এর কাজ টি `loop()` ফাংশনের ভেতরে করা হয়েছে। এখানে ভেতরে আরেকটি `for()` লুপের ভেতরে ০ নামার চ্যানেলের ক্রমাগত ১০ মিলিসেকেন্ড অন্তর ০ থেকে শুরু করে ২৫৫ পর্যন্ত ভ্যালু দিচ্ছি এবং একইরকম আরেকটা লুপে উলটো কাজ টা করছি। এর ফলে পুরো ব্যাপার টা এরকম দাঁড়ায় যে ০ নামার চ্যানেলে আস্তে আস্তে ভোল্টেজ বাড়ানো হচ্ছে এবং আবার আস্তে আস্তে ভোল্টেজ কমানো হচ্ছে এবং এই কাজ টা `loop()` ফাংশনের ভেতরে চলতেই থাকবে। যেহেতু ০ চ্যানেলের সাথে JRC বোর্ডের ১৩ নামার পিন যুক্ত আছে, সেক্ষেত্রে তুমি দেখতে পাবে তোমার প্রজেক্টে এলাইটি লাইট টি আস্তে আস্তে উজ্জ্঳লতা বাড়ছে আবার আস্তে আস্তে উজ্জ্঳লতা কমছে যা দেখতে খুবই সুন্দর লাগে! এভাবে চাইলে তুমি নিজের বুদ্ধি খাটিয়ে বিভিন্ন পরিস্থিতিতে নানা ভাবে **PWM** পিন কন্ট্রোল করতে পারো।