

১৫/ ব্লুটুথ রিমোট তৈরীঃ

উপকরণঃ

- ১/ JRC Board
- ২/ Joystick মডিউল
- ৩/ বাক মডিউল
- ৪/ লিপো ব্যাটারী

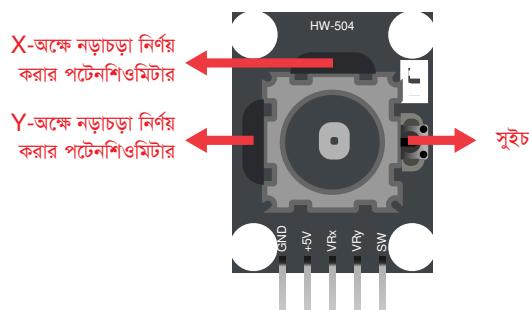
বর্ণনাঃ

ইতিমধ্যেই আমরা শিখে গেছি দুটি JRC Board এর মধ্যে কিভাবে ব্লুটুথ এর মাধ্যমে যোগাযোগ করতে হয় এবং আমরা এটাও শিখে গিয়েছি যে কিভাবে সেই ব্লুটুথ কমিউনিকেশন ব্যবহার করে একটি RC Car ও বানিয়ে ফেলা যায় যা কিনা স্মার্টফোন দিয়ে নিয়ন্ত্রণ সম্ভব। এখানে কিন্তু চাইলে স্মার্টফোন ব্যবহার না করে এই JRC Board দিয়ে একটি রিমোটও বানিয়ে ফেলা সম্ভব। সেক্ষেত্রে আমাদের এখন একটি কম্পোনেন্ট সম্পর্কে জানতে হবে যেটি হচ্ছে joystick module।

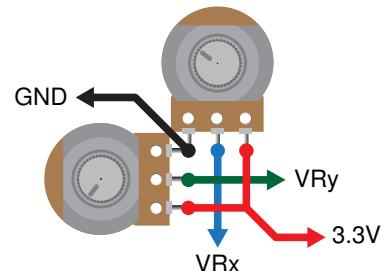
joystick module হলো এমন একটি ডিভাইস যেটি চাইলে উপরে-নিচে এবং বামে-ডানে নাড়ানো যায় এবং এটি কেনে দিকে নাড়ানো হয়েছে সেটি সে সিগ্নাল হিসেবে আউটপুট দিতে পারে। এটির ভেতরে একটি সুইচও থাকে যেটি চাইলে পুশ করেও সিগনাল পাঠানো যায়। এতে কানেকশন এর জন্য ৫ টি পিন থাকে। দুটি হলো পাওয়ার পিন যেখানে GND পিনের সাথে JRC Board এর গ্রাউন্ড পিনের সংযোগ দেয়া উচিত, এবং 5V পিনের সাথে JRC Board এর 3.3V পিনের সংযোগ দেয়া উচিত যেহেতু এই মডিউল থেকে JRC Board এর কাছে সিগনাল যাবে এবং JRC Board নিজে 3.3V এর বেশি ইনপুট সহ্য করতে পারেনা। বাকি তিনটি পিন হলো VRx, VRy এবং SW পিন। SW পিনটি জয়স্টিক টি বরাবর চাপ দিলে সিগনাল পাঠায় যেখানে প্রেস না করলে কিছুই হ্যান্ডে এবং প্রেস করলে লো (0V) সিগনাল পাঠায়। VRx পিনের মানে হলো Variable Resistor (x-axis), একইভাবে VRy পিনের মানে হলো Variable Resistor (y-axis)। এই পিনগুলো কেমন আউটপুট দেয় তা বোঝার আগে এটা জেনে নেয়া জরুরি যে এটার ভেতরে আসলে কি আছে:



জয়স্টিক মডিউলের পিনআউট



ক্যাপ খুলে আভ্যন্তরীন গঠন বর্ণনা



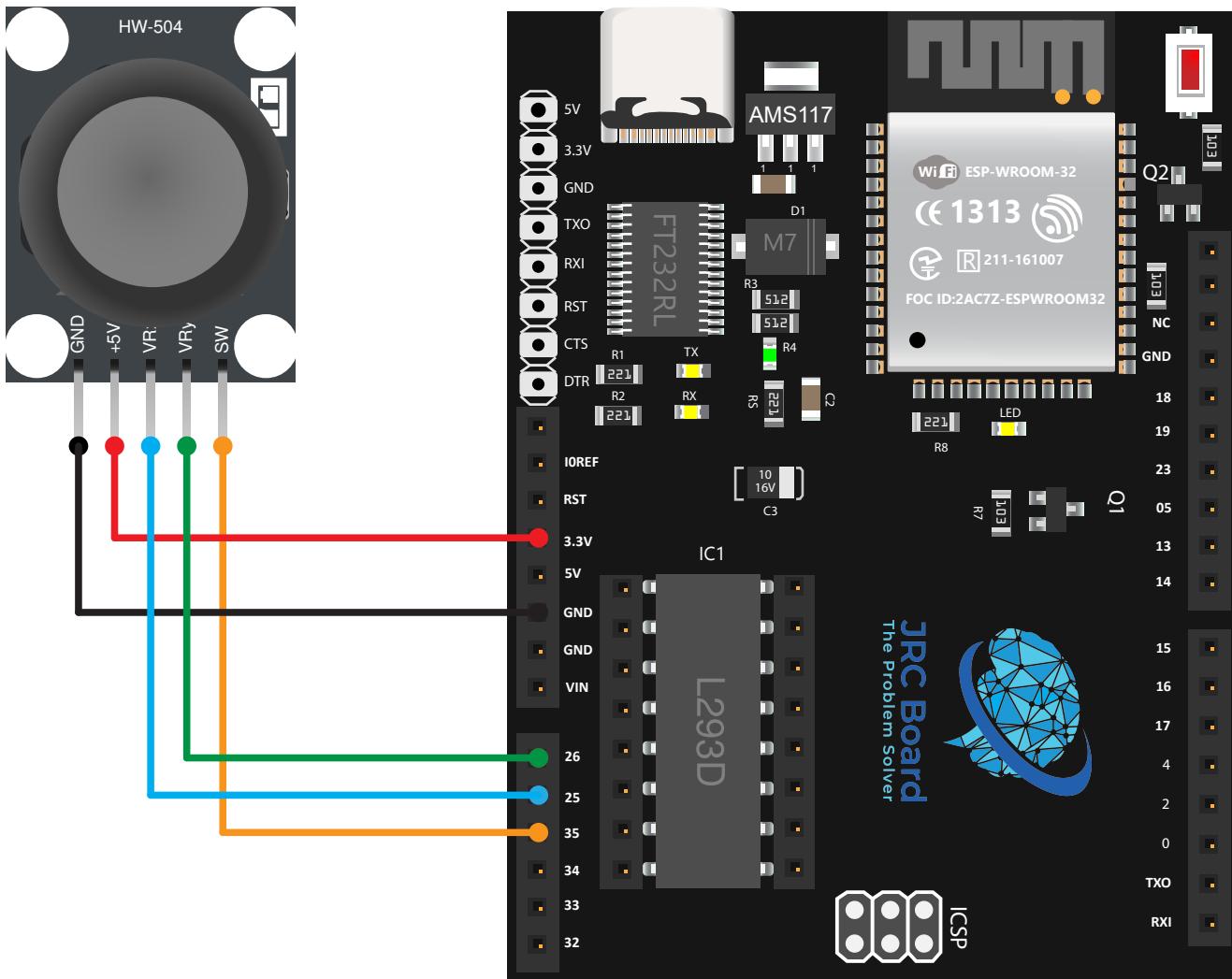
ভেতরে সার্কিট যেভাবে কাজ করে

আমরা ইতিমধ্যেই পটেনশিওমিটার অর্থাৎ ভ্যারিয়েবল রেজিস্টর (পরিবর্তনশীল রেজিস্টর) কিভাবে JRC Board এ ব্যবহার করতে হয় সেটা শিখে গেছি তাইনা? এখানে জয়স্টিক এর X-অক্ষ এবং y-অক্ষ এর দিকে নড়াচড়া হলে সেটি দুটি ভ্যারিয়েবল রেজিস্টরের মাধ্যমে মেপে দেখা হয় যেগুলো কিনা জয়স্টিক নাড়ানোর সাথে সাথে ঘূরতে থাকে। জয়স্টিক টা যদি ছেড়ে দেই, সেক্ষেত্রে স্পিং এর কারণে সেটা মাঝামাঝি অবস্থান করে, সেক্ষেত্রে পটেনশিওমিটারেও মাঝামাঝি রোধ দেখা যায় এবং উপরে-নিচে বা ডানে-বামে সরালে এই রোধের মান বাড়ে বা কমে।

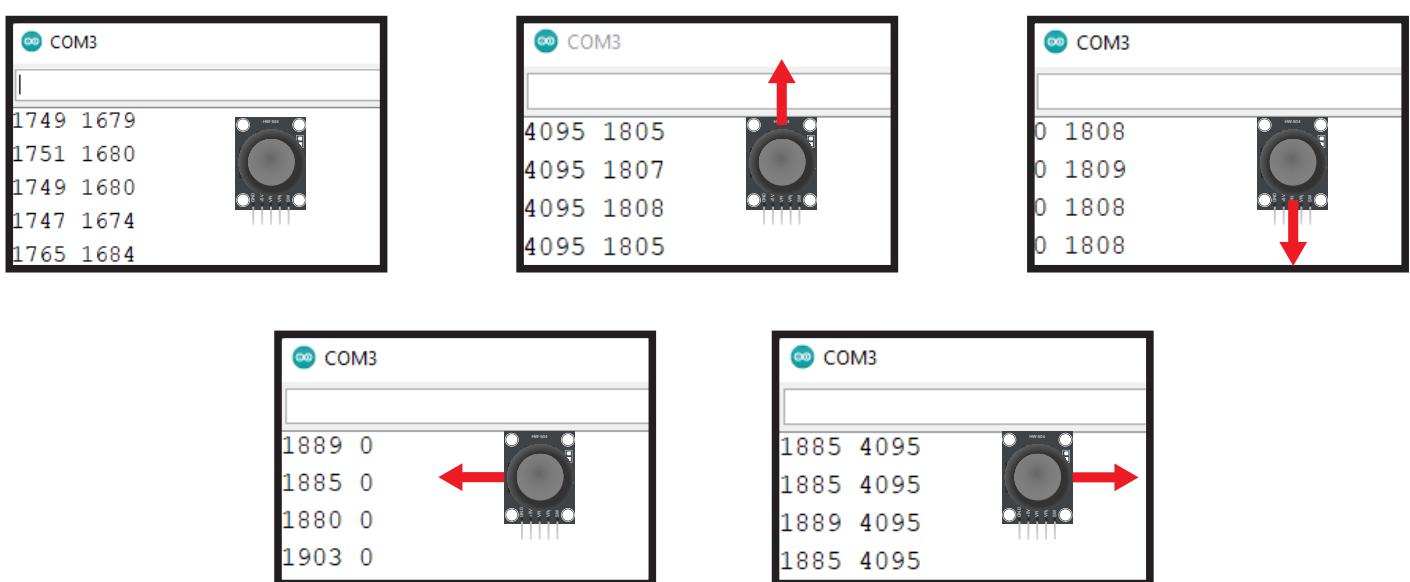
তাহলে চলো দেখে নিই এই জয়স্টিক মডিউলের সাথে JRC Board এর কিভাবে সংযোগ দিতে হয়। এবং কিভাবে এতে ভ্যালু চেক করতে পারি এবং প্রোগ্রাম করতে পারি:



```
byte x_pin=25, y_pin=26;
int x,y;
void setup(){
  Serial.begin(9600);
}
void loop(){
  Serial.println(String(analogRead(y_pin))+ " " + String(analogRead(x_pin)));
}
```



লক্ষ্য করে দেখ, আমরা এখানে কেবল যে দুটি পিনে VR_X এবং VR_Y পিন লাগিয়েছি, সেই পিনগুলোথেকে এনালগ রিডিং নিচ্ছি। আমরা যদি এই সার্কিটে চিত্রের মতো কোড আপলোড দিই, সেক্ষেত্রে সিরিয়াল মনিটরে নিচের মতো জয়স্টিক এর বিভিন্ন অবস্থানে পটেনশিওমিটার এর বিভিন্ন মান দেখতে পাবো:



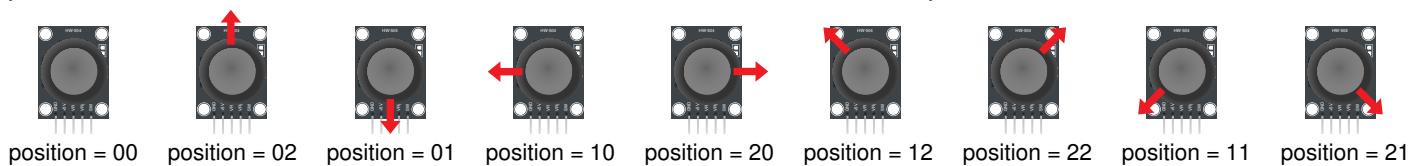
এর মানে আমরা যেদিকে জয়স্টিক ঘুরাচ্ছি, সে অনুযায়ী মান পরিবর্তন হচ্ছে তাইনা? এর মানে আমরা একটা মোটামুটি সীমা ধরতে পারি যে সীমার বাইরে গেলে JRC Board বুবো নিবে যে আমি জয়স্টিক নিয়ে নাড়াচাড়া করছি। আমরা ইতিমধ্যেই সিরিয়াল মনিটর থেকে দেখে নিয়েছি যে আপাতত জয়স্টিক এ হাত না দিলে মোটায়ুটি ১৭০০-১৯০০ এর মধ্যে মান থাকে। সেক্ষেত্রে ধরে রাখা যায় যে ১০০০ এর কম ভ্যালু পেলে একদিকে সরানো হয়েছে আর ৩০০০ এর বেশি ভ্যালু পাওয়া মানে আরেকদিকে সরিয়ে দেয়া। যেমন y-অক্ষের মান যদি ১০০০ এর কম হয় তবে এটি নিচের দিকে নামানো হয়েছে, আবার যদি ৩০০০ এর বেশি হয় তার মানে এটা উপরের দিকে ঠেলে দেয়া হয়েছে। একইভাবে X-ক্ষেত্রের মান যদি ১০০০ এর কম হয় তার মানে জয়স্টিকটিকে বাম দিকে ঠেলে দেয়া হয়েছে এবং ৩০০০ এর বেশি পাওয়া মানে এটিকে ডান দিকে ঠেলে দেয়া হয়েছে। এটা কোডে এভাবে লজিক দেয়া যায়ঃ

```

byte x_pin=25, y_pin=26;
int x,y,position;
void setup(){
    Serial.begin(9600);
}
void loop(){
    x = analogRead(x_pin);
    y = analogRead(y_pin);
    if(x<=1000) x=1;
    else if(x>=3000) x=2;
    else x=0;
    if(y<=1000) y=1;
    else if(y>=3000) y=2;
    else y=0;
    position = x*10 + y;
}

```

কোডের দিকে লক্ষ্য করলে দেখতে পাবো যে প্রথমে X-অক্ষের জন্য পটেনশিওমিটারের এনালগ রিডিং নেবার পরে যদি দেখা যায় এর মান 3000 এর বেশি সেক্ষেত্রে X এর মান 2 রাখা হচ্ছে আর 1000 এর কম হলে তার মান 1 নির্ধারণ করা হচ্ছে। যদি এর কোনটাই না হয় (1000 থেকে 3000 এর মধ্যে), সেক্ষেত্রে তার মান 0 রাখা হচ্ছে। একই কাজ y অক্ষের বেলোতেও করা হচ্ছে। এরপর দুইটি অক্ষের সর্বশেষ মান মিলিয়ে একটি সংখ্যা বানিয়ে সেটা position নামক ভ্যারিয়েবল এ ধারণ করে রাখা হচ্ছে। চলো দেখে নিই যে কখন কোন অবস্থায় এই position এর মান কেমন হওয়া উচিতঃ



তাহলে এখন কাজ হচ্ছে কোন পজিশনে আছে সেই মোতাবেক আমাদের ব্লুটুথ এর মাধ্যমে কমান্ড পাঠানো তাইনা? আমরা ইতিমধ্যেই ৯ নং প্রজেক্টে ব্লুটুথ মাস্টার মোডে কিভাবে ব্যবহার করতে হয় সেটি শিখে গোছি। সেই জন্য কাজে লাগিয়ে আমরা সর্বশেষ ১৪ নং প্রজেক্টে যে Bluetooth RC Car বানিয়েছিলাম তাতে কানেক্ট করে দরকারী কমান্ড পাঠানোর ব্যবস্থা করি। নিচের কোড দেখলেই পরিষ্কার হয়ে যাবেঃ

```

#include "BluetoothSerial.h"
BluetoothSerial SerialBT;
byte x_pin=25, y_pin=26, connect;
int x, y, position;
void setup(){
    Serial.begin(9600);
    pinMode(13, OUTPUT);
    SerialBT.begin("Remote", true); connect = SerialBT.connect("Bluetooth RC Car");
    if(connect) digitalWrite(13,HIGH);
}
void loop(){
    x = analogRead(x_pin); y = analogRead(y_pin);
    if(x<=1000) x=1; else if(x>=3000) x=2; else x=0;
    if(y<=1000) y=1; else if(y>=3000) y=2; else y=0;
    position = x*10 + y;
    if (position == 00) SerialBT.write('S');
    else if (position == 02) SerialBT.write('F');
    else if (position == 01) SerialBT.write('B');
    else if (position == 10) SerialBT.write('L');
    else if (position == 20) SerialBT.write('R');
    else if (position == 12) SerialBT.write('G');
    else if (position == 22) SerialBT.write('I');
    else if (position == 11) SerialBT.write('H');
    else if (position == 21) SerialBT.write('J');
    delay(50);
}

```

এখানে উল্লেখ্য যে `SerialBT.connect("Bluetooth RC Car")` লেখার মাধ্যমে আমাদের পূর্বের বানানো ব্লুটুথ কন্ট্রোল গাড়ির সাথে কানেক্ট হবার চেষ্টা করবে এবং কানেকশন সফল হলে ১৩ নাম্বার পিনে যে বিল্ট-ইন এলাইডি ছিলো সেটা জ্বলে উঠবে। যদি আমরা রিসিভার ডিভাইসের নাম পরিবর্তন করি, সেই নাম হ্রব্রহ্ম এখানে বসাতে হবে। একবার কানেকশন সফল হয়ে গেলে এরপরে তুমি মনে মতো জয়স্টিক ঘুরিয়ে গাড়িটি চালনা করতে পারবে!