

## ১৯/ ভূমিকম্প শনাক্তকরণঃ

উপকরণঃ

১/ JRC Board

২/ Tilt সেন্সর

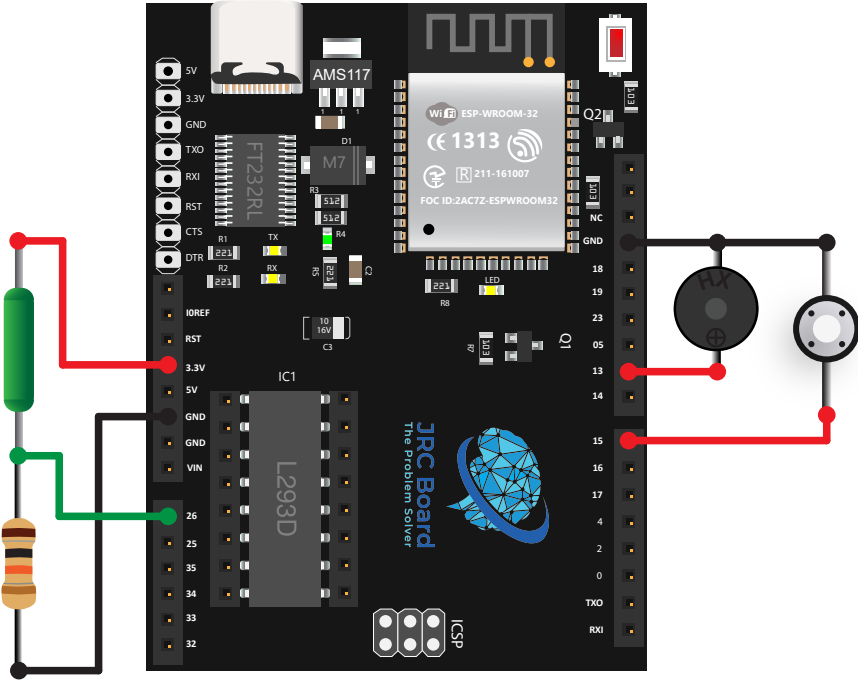
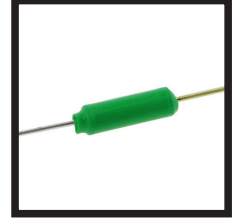
৩/ buzzer

৪/ পুশ বাটন

বর্ণনাঃ

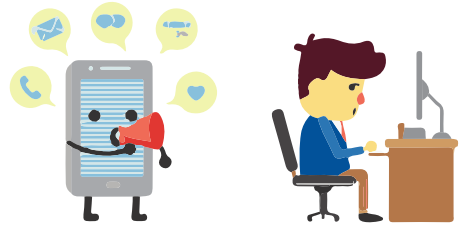
এই প্রজেক্টে একটি gas সেন্সর মডিউল দিয়ে রান্নাঘরে মিথেন গ্যাস অথবা সিলিন্ডারে থাকা বিউটেন গ্যাসের উপস্থিতি শনাক্ত করা হবে এবং গ্যাস বা ধোঁয়া শনাক্ত হলে সেটা সাথে সাথে এলার্ম বাজিয়ে সতর্ক করে দিবে। এই প্রজেক্টে gas সেন্সর মডিউল সম্পর্কে ধারণা নেয়া দরকার। চলো আগে দেখে নিই এর গঠন কিরূপঃ

Tilt সেন্সর হলো এক বিশেষ ধরনের সেন্সর যেখানে নড়াচড়া হলে বা একদিকে কাত হলে সেন্সর এর দুটি মাথা শর্ট হতে পারে কিংবা ওপেনও হতে পারে। JRC Board এ এটিকে সুইচ হিসেবে ব্যবহার করলে সেক্ষেত্রে এর অবস্থার পরিবর্তন কে নড়াচড়া ধরে নিয়ে ভূমিকম্পের উপস্থিতি নির্ধারণ করা যায়। এটি নিজে কোন ভোল্টেজ দেয়া, কেবল দুটি প্রান্ত কানেক্ট করে দিতে বা খুলে দিতে পারে। সেক্ষেত্রে আমরা যদি এটিকে LDR এর ন্যায় ব্যবহার করি (প্রজেক্ট নং ৪) তাহলেই আরামসেই এটাকে ডিজিটাল সুইচ হিসেবে চালিয়ে দেয়া যায়। চলো দেখে নিই এর সার্কিট কিভাবে বানানো যেতে পারেঃ



এবার কোডিং এর পালা। এখানে আমরা কেবল ডিজিটাল রিডিং নিয়েই হয়তোবা কাজ চালিয়ে দিতে পারতাম, কিন্তু সেক্ষেত্রে সবসময় সঠিক মান পেতে পারিনা কেননা আমরা জানিনা যে ঠিক কোন মুহূর্তে সুইচ অন থাকবে নাকি অফ থাকবে। সেক্ষেত্রে আমরা একটা বিশেষ সিস্টেম ব্যবহার করতে পারি যেটার নাম হচ্ছে Interrupt।

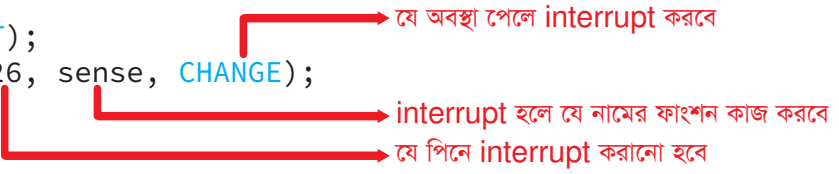
JRC Board এ কোন প্রোগ্রাম করে দিলে সে নিজের মতো করে সেই প্রোগ্রাম রান করতে থাকে তাইনা? এখানে কিন্তু একটা কাজ চলাকালীন তাকে আরেকটা কাজ দিতে পারার কথা না। কিন্তু এক বিশেষ পদ্ধতিতে তাকে "interrupt" বা বিরক্ত করলে সেক্ষেত্রে সে তার হাতের কাজ থামিয়ে রেখে যে কাজ করতে দেয়া হয়েছে সেই কাজ সম্পাদন করে আবার পূর্বের কাজে ফেরত আসে। দুইভাবে JRC Board কে ইন্টারাপ্ট করা যায়, এক হচ্ছে হার্ডওয়্যার আরেকটি হলো সফটওয়্যার। আমরা আপাতত কাজের স্বার্থে এখানে হার্ডওয়্যার দিয়ে কিভাবে ইন্টারাপ্ট করতে হয় সেটা দেখবো। হার্ডওয়্যার ইন্টারাপ্ট এর ক্ষেত্রে আমরা JRC Board এর নির্দিষ্ট পিনে কোন নির্দিষ্ট সিগনাল পাঠিয়ে তাকে ইন্টারাপ্ট করতে পারি। মজার কথা হলো JRC Board এর যেকোন GPIO পিনেই চাইলে Interrupt সিস্টেম প্রয়োগ কর যায়।



এখন চলো আমরা দেখে নিই কিভাবে কোডে এই interrupt ফাংশনের প্রয়োগ করতে হয়। কোন একটি পিন কে interrupt হিসেবে ব্যবহার করতে void setup() ফাংশনের ভেতরে নিচের মতো কোড লিখতে হয়ঃ

```
void setup(){
  pinMode(26, INPUT);
  attachInterrupt(26, sense, CHANGE);
}

void loop(){
}
```



এখানে আমরা দেখতে পাচ্ছি একটি attachInterrupt() নামের ফাংশন কল করা হয়েছে যার ভেতরে প্রথমে পিন নাম্বার, এরপর যে ঐ পিনে ট্রিগার হলে যে ফাংশন কাজ করাতে চাই, এরপর ঐ পিনের কি অবস্থা হলে এটা ট্রিগার করবে সেটি লিখে দিতে হয়। এখানে কিছু বিশেষ বিশেষ অবস্থায় ট্রিগার করানো যায় যা নিচে বর্ণনা করা হচ্ছেঃ

**CHANGE** : যদি পিনের ডিজিটাল রিডিং হাই থেকে লো হয় অথবা লো থেকে হাই হয় তখন একবার কাজ করবে

**RISING** : যদি পিনের ডিজিটাল রিডিং লো থেকে হাই হয় তখন একবার কাজ করবে

**FALLING** : যদি পিনের ডিজিটাল রিডিং হাই থেকে লো হয় তখন একবার কাজ করবে

**HIGH** : যতক্ষণ ধরে পিনের ডিজিটাল রিডিং হাই থাকবে ততক্ষণ ধরে কাজ করবে

**LOW** : যতক্ষণ ধরে পিনের ডিজিটাল রিডিং লো থাকবে ততক্ষণ ধরে কাজ করবে

এখানে লক্ষ্য করলে দেখতে পাচ্ছি যে আমাদের কিন্তু একটি ম্যানুয়াল ফাংশনও বানাতে হচ্ছে তাইনা? আমরা পূর্বেই শিখে এসেছি যে ম্যানুয়াল ফাংশন কিভাবে বানাতে হয়। এখানেও পদ্ধতি প্রায় হুবহু একই, কেবল ছোট্ট একটি বিষয় যুক্ত হয়। নিচের কোডটি দেখে নিইঃ

```
void IRAM_ATTR sense(){
}

void setup(){
  pinMode(26, INPUT);
  attachInterrupt(26, sense, CHANGE);
}

void loop(){
}
```

এখানে লক্ষ্য করলে দেখতে পাবে যে আমরা সাধারণত void sense() লেখার বদলে void IRAM\_ATTR sense() লিখেছি, এর কারণ হচ্ছে আমরা চাইলেই JRC Board কে বিরক্ত করতে পারবোনা, বেশি বিরক্ত করলে সে রেগে যাবে তাইনা? যদি বিরক্ত করতেই হয়, এবং অল্প সময়ের জন্য বিরক্ত করা যায় এবং ছোট্ট কোন কাজ করিয়ে নেয়া যায়। এই কাজ টা যত দ্রুত করা যায় ততোই ভালো কেননা তার তো মেইন প্রোগ্রাম পড়েই রয়েছে তাইনা? এক্ষেত্রে আমরা যদি IRAM\_ATTR লিখি সেক্ষেত্রে এই ফাংশনটি JRC Board এর ফ্ল্যাশ মেমোরি তে জমা না রেখে এর internal RAM এ জমা রাখে। তোমরা তো জানোই ফ্ল্যাশ মেমোরী বা ROM এর তুলনায় RAM অধিকতর দ্রুত এবং এখানে কিছু রাখলে মুহূর্তেই কাজ সমাধা হয়ে যায়। IRAM এর মানে হচ্ছে Internal RAM।

এবার এই ফাংশনের ভেতরে নিজের দরকারমতো স্টেটমেন্ট লিখে দিলেই কাজ হয়ে যায়। কেবল খেয়াল রাখতে হবে যে এখানে যাতে কাজের পরিমাণ বা স্টেটমেন্ট এর পরিমাণ খুব বেশি না হয়, সেক্ষেত্রে মেইন প্রোগ্রাম কাজ করতে চাইবেনা।

আমাদের এই প্রজেক্টে কি করার কথা? যদি সেন্সর এর অবস্থার পরিবর্তন হয়, সেক্ষেত্রে এলার্ম বাজানো তাইনা? সেক্ষেত্রে আমরা alarm নামক একটি ভ্যারিয়েবল ধরে নিতে পারি যেটার মান 0 হলে এলার্ম অফ থাকবে আর 1 থাকা মানে অন থাকবে। এখন তাইলে সেন্সর পিনে interrupt হলে কি করতে হবে? interrupt এর জন্য করা ফাংশনের ভেতরে কেবল alarm এর মান 1 বানিয়ে দিলেই কিন্তু হয়ে যায়, বাকি কাজ আমরা মেইন প্রোগ্রামে সেরে ফেলতে পারি। নিচের কোডটি লক্ষ্য করোঃ

```

bool alarm = 0;
int sensor = 26, buzzer = 13;
void IRAM_ATTR sense(){
    alarm = 1;
}
void setup(){
    pinMode(sensor, INPUT); pinMode(buzzer, OUTPUT);
    attachInterrupt(sensor, sense, CHANGE);
}
void loop(){
    if(alarm == 1){
        digitalWrite(buzzer,1); delay(250);
        digitalWrite(buzzer,0); delay(250);
    }
    else digitalWrite(buzzer,0);
}

```

এখানে দেখতে পাচ্ছি যে alarm নামক একটা ভ্যারিয়েবল রাখা হয়েছে যেটার মান 1 হলে দেখা যাচ্ছে যে void loop() এর ভেতরে 250 মিলিসেকেন্ড অন্তর buzzer টি বেজে চলছে, আর alarm এর মান 0 হলে buzzer অফ থাকছে। এখন আমরা void IRAM\_ATTR sense() ফাংশনের ভেতরে দেখতে পাচ্ছি যে alarm এর মান 1 বানিয়ে দেয়া হচ্ছে। এর মানে কি দাঁড়ায়? যদি ২৬ নম্বার পিনে সেন্সর এর জন্য interrupt হয়, সেক্ষেত্রে alarm এর মান 1 হয় যাবে, এবং সেক্ষেত্রে এলার্ম বেজে উঠবে। এর ফলে প্রজেক্টের একটি অংশ মটামুটি তৈরী হয়ে গেছে।

এখন আমাদের এই এলার্ম অফ করতে হবে তাইনা? সেক্ষেত্রে ছবিতে দেয়া সার্কিট অনুযায়ী একটা বাটনও যোগ করে ফেলতে পারো এবং সেখানেও চাইলে আরেকটি interrupt এর প্রয়োগ করতে পারো। এই বাটন ইন্টারপ্ট করলে কাজ হবে alarm এর মান 0 করে দেয়া যার ফলে এলার্ম অফ হয়ে যাবে। নিচের কোড দেখলেই বুঝে যাবে।

```

bool alarm = 0;
int sensor = 26, buzzer = 13, button = 15;
void IRAM_ATTR sense(){
    alarm = 1;
}
void IRAM_ATTR push(){
    alarm = 0;
}
void setup(){
    pinMode(sensor, INPUT); pinMode(button, INPUT_PULLUP);
    pinMode(buzzer, OUTPUT);
    attachInterrupt(sensor, sense, CHANGE);
    attachInterrupt(button, push, FALLING);
}
void loop(){
    if(alarm == 1){
        digitalWrite(buzzer,1); delay(250);
        digitalWrite(buzzer,0); delay(250);
    }
    else digitalWrite(buzzer,0);
}

```

এখানে লক্ষ্য করলে দেখতে পাবে যে আমরা পুশ বাটনে ইন্টারপ্ট এর ক্ষেত্রে FALLING ব্যবহার করেছি, কারণ আমরা পূর্বে ব্যবহার থেকে জেনেছি যে সাধারণত পুশ বাটন চাপ না দিলে হাই স্টেটে থাকে, এবং প্রেস করলে লো হয়ে যায়। এর মানে যেহেতু প্রেস করলে ঐ পিনের ডিজিটাল রিডিং হাই থেলে লো হচ্ছে, ঠিক সেই মুহূর্তে ট্রিগার করার জন্য FALLING লেখা হয়েছে।