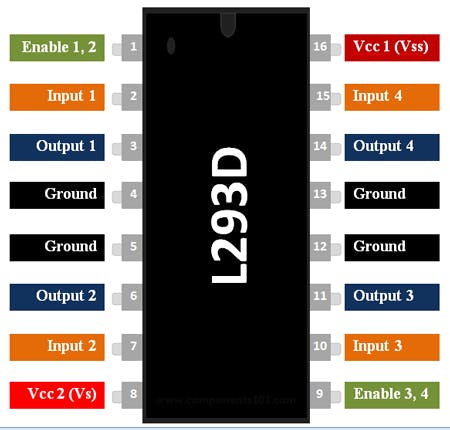
**Destination notifier using GPS**

INTRODUCTION

For that, first you need to find latitude and longitude to define the location. Once you find your location, you can use the latitude and longitude values to find distance to the location and by keeping a range you can turn on the notifier. **GPS** stands for global positioning system and can be used to determine position, time and speed if you are travelling.

NEO - 6M GPS ModuleThis module has an external antenna and built-in EEPROM.

Components

Arduino uno

Neo -6m gps module

Dual h bridge motor driver L293D

Motor

Breadboard

Connecting wired

Applications

* Location — Determining a position.
* Navigation — Getting from one location to another.
* Tracking — Monitoring object or personal movement.
* Mapping — Creating maps of the world.
* Timing — Making it possible to take precise time measurements

Objective

**This activity ,you will help students to achieve following objectives**

**1.** Understanding the principle and operation of GPS system

2. Design algorithm and flowchart for destination notifiction using GSM

3. Programming GPS Module using Arduino uno

4. Interfacing GPS Module with Arduino uno

Programming steps

1. Include tiny GPS library
2. Include software serial library
3. Initialize static constant for Tx and Rx pin and for gps
4. Define LED and motor as output port
5. Read gps signal latttue and longitude pin
6. Continusly check data between updatd position and defined position
7. If defined position is near ,then notify by blinking LED and motor start vibrating

Program

#include <TinyGPS++.h>  
#include <SoftwareSerial.h>  
static const int RXPin = 4, TXPin = 3;  
static const uint32\_t GPSBaud = 9600;  
// The TinyGPS++ object  
TinyGPSPlus gps;  
// The serial connection to the GPS device  
SoftwareSerial ss(RXPin, TXPin);  
// For stats that happen every 5 seconds  
unsigned long last = 0UL;  
int motorpin1=6;  
int motorpin2=7;  
void setup()  
{  
 Serial.begin(115200);  
 ss.begin(GPSBaud);  
pinMode(motorpin1,OUTPUT);  
pinMode(motorpin2,OUTPUT);  
}  
void loop()  
{  
 // Dispatch incoming characters  
 while (ss.available() > 0)  
 gps.encode(ss.read());  
 if (gps.location.isUpdated())  
 {  
 Serial.print(F(" Lat="));  
 Serial.print(gps.location.lat(), 6);  
 Serial.print(F(" Long="));  
 Serial.println(gps.location.lng(), 6);  
 }  
 else if (millis() - last > 5000)  
 {  
 Serial.println();  
 if (gps.location.isValid())  
 {  
 // replace 'Dest\_LAT' and 'Dest\_LON' values based on your location   
 // you can find Latitude and Longitude from Read\_Lat\_Lng.ino   
 static const double Dest\_LAT = 18.786254, Dest\_LON = 73.880798;  
 double distanceToDest =  
 TinyGPSPlus::distanceBetween(  
 gps.location.lat(),  
 gps.location.lng(),  
 Dest\_LAT,   
 Dest\_LON);  
 Serial.print(F("Distance to Destination ="));  
 Serial.print(distanceToDest/1000, 6); // \*Prints distance to destination   
 if(distanceToDest/1000 < 0.050000) //Here when distanceToDest/1000 is less than 0.050000 LED turns on . So change \*distance to destination as per your requirement.   
 {  
 digitalWrite(motorpin1,LOW);  
 digitalWrite(motorpin2,HIGH);  
 }  
 else  
 {  
 digitalWrite(motorpin1,HIGH);  
 digitalWrite(motorpin2,HIGH);   
 }  
 }  
 if (gps.charsProcessed() < 10)  
 Serial.println(F("WARNING: No GPS data. Check wiring."));  
 last = millis();  
 Serial.println();  
 }  
}

Hardware

* Connect 5V to Enable 1, Vs and Vss on the L293D
* Connect digital output pins (we are using 6 and 7 ) to input 1 and input 2 on the L293D.
* Connect your Arduino's GND to both GND pins on the same side of the L293D
* Finally connect output 1 and output 2 of the L293D to your motor pins
* Connect transmit pin (tx )of module to digital pin 2 and receive pin to digital pin 3
* Connect VCC and GNDpin of module to aduino board

