const int led = 13;  
  
void setup() {  
  Serial.begin(9600);  
  pinMode(led, OUTPUT);  
}  
  
void loop() {  
  char c = Serial.read();  
  if(c == 'a'){  
    digitalWrite(led, HIGH);  
  }  
  else if(c == 'b'){  
    digitalWrite(led, LOW);  
  }  
  else{  
        
  }  
  delay(25);  
}

maps - html code:

<html>

<head>

<title>Navigation Functions (Heading)</title>

<meta name="viewport" content="initial-scale=1.0, user-scalable=no">

<meta charset="utf-8">

<style>

/\* Always set the map height explicitly to define the size of the div

\* element that contains the map. \*/

#map {

height: 100%;

}

/\* Optional: Makes the sample page fill the window. \*/

html, body {

height: 100%;

margin: 0;

padding: 0;

}

#floating-panel {

position: absolute;

top: 10px;

left: 25%;

z-index: 5;

background-color: #fff;

padding: 5px;

border: 1px solid #999;

text-align: center;

font-family: 'Roboto','sans-serif';

line-height: 30px;

padding-left: 10px;

}

</style>

</head>

<body>

<div id="map"></div>

<div id="floating-panel">

Origin: <input type="text" readonly id="origin">

Destination: <input type="text" readonly id="destination"><br>

Heading: <input type="text" readonly id="heading"> degrees

</div>

<script>

// This example requires the Geometry library. Include the libraries=geometry

// parameter when you first load the API. For example:

// <script src="https://maps.googleapis.com/maps/api/js?key=YOUR\_API\_KEY&libraries=geometry">

var marker1, marker2;

var poly, geodesicPoly;

function initMap() {

var map = new google.maps.Map(document.getElementById('map'), {

zoom: 4,

center: {lat: 34, lng: -40.605}

});

map.controls[google.maps.ControlPosition.TOP\_CENTER].push(

document.getElementById('info'));

marker1 = new google.maps.Marker({

map: map,

draggable: true,

position: {lat: 12.8239, lng: 80.0439}

});

marker2 = new google.maps.Marker({

map: map,

draggable: true,

position: {lat: 48.857, lng: 2.352}

});

var bounds = new google.maps.LatLngBounds(

marker1.getPosition(), marker2.getPosition());

map.fitBounds(bounds);

google.maps.event.addListener(marker1, 'position\_changed', update);

google.maps.event.addListener(marker2, 'position\_changed', update);

poly = new google.maps.Polyline({

strokeColor: '#FF0000',

strokeOpacity: 1.0,

strokeWeight: 3,

map: map,

});

geodesicPoly = new google.maps.Polyline({

strokeColor: '#CC0099',

strokeOpacity: 1.0,

strokeWeight: 3,

geodesic: true,

map: map

});

update();

}

function update() {

var path = [marker1.getPosition(), marker2.getPosition()];

poly.setPath(path);

geodesicPoly.setPath(path);

var heading = google.maps.geometry.spherical.computeHeading(path[0], path[1]);

document.getElementById('heading').value = heading;

document.getElementById('origin').value = path[0].toString();

document.getElementById('destination').value = path[1].toString();

}

</script>

<script src="https://api key/maps/api/js?key=YOUR\_API\_KEY&libraries=geometry&callback=initMap"

async defer></script>

</body>

</html>

Arduino code:

#include <Arduino.h>

#include <stdio.h>

#define ON 1

#define OFF 0

/\*

 \* Pin Description

 \*/

int vibration\_Sensor = A5;

int LED = 13;

/\*

 \* Programme flow Description

 \*/

int present\_condition = 0;

int previous\_condition = 0;

/\*

 \* Pin mode setup

 \*/

void setup() {

pinMode(vibration\_Sensor, INPUT);

pinMode(LED, OUTPUT);

}

/\*

 \* Led blink

 \*/

void led\_blink(void);

/\*

 \* main\_loop

 \*/

void loop() {

previous\_condition = present\_condition;

present\_condition = digitalRead(A5); // Reading digital data from the A5 Pin of the Arduino.

if (previous\_condition != present\_condition) {

led\_blink();

} else {

digitalWrite(LED, OFF);

}

}

void led\_blink(void) {

digitalWrite(LED, ON);

delay(250);

digitalWrite(LED, OFF);

delay(250);

digitalWrite(LED, ON);

delay(250);

digitalWrite(LED, OFF);

delay(250);

}