Telemetry & Real-Time Analytics :

Overall Flow (Use Case)

Telemetry & Real-Time Analytics		
Use Case:	Edge Data Filtering	Implementing logic to decide what data stays on the car and what goes to the cloud (e.g., sending GPS every second, but transmission temperature only if it exceeds a threshold).

- AWS Cloud requires a paid subscription; therefore, Firebase is used as the cloud backend instead of AWS.
- First, we sign in to Firebase after creating a project and database in the Firebase console.
- The Firebase database is then linked to our PC.
- The Firebase backend setup is completed successfully, and PC-to-Firebase connectivity is validated.
- The temperature sensor measures the current temperature.
- The ESP32 reads this value continuously.
- ESP32 connects to Wi-Fi and sends the temperature to Firebase.
- Firebase stores the value in the Realtime Database.
- A threshold of **30°C** is set in the code.
- If the temperature is **below 30°C**, no alert is generated.
- If the temperature **exceeds 30°C**, an alert flag becomes true.
- This alert is updated instantly in Firebase.
- The system works in real time.
- It shows how sensor data can trigger cloud alerts automatically.
- Next, we will connect the OKT507-C board to the PC and link it to our Firebase project.
- Finally, the OKT507-c board fetches the temperature values from the firebase

realtime database.

Firebase Creation:

Step-by-Step Implementation Using Firebase

Create Firebase Project

1. Go to **Firebase Console**
2. Create a new project
3. Enable **Realtime Database**

Set database rules (for testing):

STEP 1: Create Firebase Project

1□ Click the big box:

“Get started by setting up a Firebase project”

(Highlighted in your screenshot)

□ STEP 2: Project Creation Screen

After clicking, you will see **Create a project** page.

2□ Enter Project name

Example (use this or your own):

edge-data-filtering

3□ Click Continue

STEP 3: Google Analytics Screen

4□ You will see “Enable Google Analytics”

- Turn **OFF** (recommended for simplicity)
- Click **Create project**

Wait 10–20 seconds

5□ Click Continue once project is ready

Firebase Project Created

Now you will be inside your **Firebase project dashboard**.

NEXT IMPORTANT STEP: Enable Realtime Database

STEP 4: Enable Realtime Database

6□ In left sidebar:

Click **Build**

7□ Click Realtime Database

8□ Click Create Database

STEP 5: Database Setup

9□ Choose location:

- Select **asia-southeast1** (best for India)

Click **Next**

STEP 6: Set Rules (Testing Mode)

□ **Select:**

Start in test mode

Click **Enable**

STEP 7: Set Rules Manually

After database opens:

1. Click **Rules** tab
2. Replace everything with:

```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

3. Click **Publish**

STEP 8: Copy Database URL (IMPORTANT)

At the top of Realtime Database page, copy:

https://console.firebase.google.com/project/edge-data-filtering/database/edge-data-filtering-default-rtdb/rules

Save this URL — we will use it in OKT507-C code.

WHAT YOU HAVE DONE SO FAR

✔Firebase project created

✔Realtime Database enabled

✔Database rules configured

✔Database URL ready

STEP 9: Get Database Secret (for OKT507-C)

Then:

- Send GPS data
- Apply edge filtering logic
- Push filtered data to Firebase

Before proceeding:

Tell me when you see “Realtime Database” screen
or
Send the next screenshot

STEP 9: GET DATABASE SECRET (FOR OKT507-C)

This Database Secret is what allows your edge device (OKT507-C) to write data into Firebase using HTTP.

STEP 9.1: Open Project Settings

1□ In Firebase Console, look at the top-left corner
2□ Click the □ (Gear icon)
3□ Click Project settings

STEP 9.2: Go to Service Accounts

4⃣ In Project settings page, click the Service accounts tab

5⃣ Scroll down until you see Database secrets

STEP 9.3: Generate Database Secret

6⃣ Under Database secrets, you will see a row with:

Secret

7⃣ Click Show

8⃣ Copy the long string (example):

AIzaSyBxxxxxxxxxxxxxxxxxxxxxxxx

THIS IS YOUR DATABASE SECRET

Keep it private (do not share publicly)

WHAT YOU HAVE NOW

You should have two things saved:

1⃣ Realtime Database URL

Example:

https://edge-data-filtering-default-rtdb.firebaseio.com/

2⃣ Database Secret

Example:

AIzaSyBxxxxxxxxxxxxxxxxxxxxx

HOW THESE ARE USED TOGETHER

Firebase HTTP format:

```
https://DATABASE_URL/path.json?auth=DATABASE_SECRET
```

Example:

```
https://edge-data-filtering-default-rtdb.firebaseio.com/test.json?auth=AIzaSyBxxxx
```

STEP 10: TEST FIREBASE CONNECTION FROM PC (VERY IMPORTANT)

Before touching OKT507-C, test from PC.

Open Terminal and run:

```
curl -X POST \
https://edge-data-filtering-default-rtdb.firebaseio.com/test.json \
-d '{"status":"firebase connected"}'
```

Now check Realtime Database → Data
You should see:

```
{
  "test": {
    "status": "firebase connected"
  }
```

}

If you see this → Firebase is working perfectly

Ubuntu Terminal Command:

```
curl -X POST \
https://edge-data-filtering-default-rtdb.asia-southeast1.firebasedatabase.app/test.json \
-d '{"status":"firebase connected"}'
```

Secret code: Ngz9KSI3boRFGeRMU4mb5VijUETYNLOnbA5dVqis

To send Temperature data from arduino to firebase

we need
1.APIKEY: apiKey: "AIzaSyCR_FwvqqMGctW9i6MNn4ZUAGcuIjxPAqQ"
2.Database Url :https://edge-data-filtering-default-rtdb.asia-southeast1.firebasedatabase.app/test/-OhEqfdWzZsTghqT2xMh

3.Email:teja123@gmail.com
4,Password:Votary@123

PART A: How to Get Firebase API Key

Step A1: Open Firebase Console

1. **Go to https://console.firebase.google.com**

2. **Open your project**
   edge-data-filtering

Step A2: Go to Project Settings

1. **Click □ Settings (top-left)**

2. **Click Project settings**

Step A3: Copy API Key

1. **You will land on General tab**
2. **Scroll to Your apps**
3. **You will see:**

Web API Key
AIzaSyXXXXXXXXXXXX

4. **Copy this key**

This is your API_KEY

Example:

```
#define API_KEY "AIzaSyXXXXXXXXXXXX"
```

That message is normal
It simply means you have not added any app to your Firebase project yet.
API Key is generated only after you add an app.

Below is the exact step-by-step fix.

WHY THIS IS HAPPENING

Firebase shows "There are no apps in your project" because:

- **Firebase project exists**
- **Realtime Database exists**

- **Authentication exists**
- **No app (Web / Android / iOS) is registered**

□ ESP32 uses the Web API key, so we must add a Web App.

SOLUTION: ADD A WEB APP (REQUIRED FOR ESP32)

STEP 1: Open Your Firebase Project

1. **Go to https://console.firebase.google.com**
2. **Select edge-data-filtering**

STEP 2: Go to Project Settings

1. **Click □ Settings (top-left)**
2. **Click Project settings**

STEP 3: Add a Web App

1. **Scroll to Your apps**
2. **Click the </> Web icon**

STEP 4: Register the App

1. **App nickname**
   Example:

ESP32_Web_App

1. **Do NOT check “Firebase Hosting”**
2. **Click Register app**

STEP 5: Copy the API Key

After registering, Firebase will show:

```
const firebaseConfig = {
  apiKey: "AIzaSyXXXXXXXXXXXX",
  authDomain: "...",
  databaseURL: "...",
}
```

COPY ONLY THIS:

apiKey: "AIzaSyXXXXXXXXXXXX"

This is your Firebase API Key

STEP 6: Finish Setup

1. **Click Continue to console**

Now go back to:

□ Settings → Project settings → General

You will now see:

Web API Key AIzaSyXXXXXXXXXXXX

IN APP you find this code

```
<script type="module">

  // Import the functions you need from the SDKs you need

  import { initializeApp } from "https://www.gstatic.com/firebasejs/12.7.0/firebase-app.js";

  // TODO: Add SDKs for Firebase products that you want to use

  // https://firebase.google.com/docs/web/setup#available-libraries


  // Your web app's Firebase configuration

  const firebaseConfig = {

    apiKey: "AIzaSyCR_FwvqqMGctW9i6MNn4ZUAGcuIjxPAqQ",

    authDomain: "edge-data-filtering.firebaseapp.com",

    databaseURL: "https://edge-data-filtering-default-rtdb.asia-southeast1.firebasedatabase.app",

    projectId: "edge-data-filtering",

    storageBucket: "edge-data-filtering.firebasestorage.app",

    messagingSenderId: "799430050434",

    appId: "1:799430050434:web:41e7c0cf3e23921f15ea9b"

  };


  // Initialize Firebase

  const app = initializeApp(firebaseConfig);

</script>
```

3.Email
4,Password
To get this email and password follow these steps

How to Get Firebase Email & Password (Step-by-Step)

Firebase ESP32 library uses **Email/Password authentication**, so we must **create a user** in Firebase.

✅STEP 1: Open Firebase Console

1. Go to **https://console.firebase.google.com**
2. Open your project
   edge-data-filtering

✅STEP 2: Enable Email/Password Sign-In

1. In left menu → click **Authentication**
2. Click **Get started** (if not enabled already)
3. Go to **Sign-in method** tab
4. Click **Email/Password**
5. **Enable** the toggle
6. Click **Save**

✅Email/Password authentication is now enabled

✅STEP 3: Create a Firebase User (This gives Email & Password)

1. Still inside **Authentication**
2. Click **Users** tab
3. Click **Add user**

Fill details:

Field	Example
Email	esp32device@gmail.com
Password	esp32@123

Use **simple credentials** (this is for device auth)

4. Click **Add user**

✅User created successfully

✅STEP 4: Use These in ESP32 Code

Now you have:

API Key
AIzaSyCR_FwvqqMGctW9i6MNn4ZUAGcuIjxPAqQ

□ **Database URL (IMPORTANT: base URL only)**

DO NOT include /test/-OhEqfdWzZsTghqT2xMh

Correct format:

https://edge-data-filtering-default-rtdb.asia-southeast1.firebasedatabase.app/

Email
esp32device@gmail.com

Password
esp32@123

Final arduino code

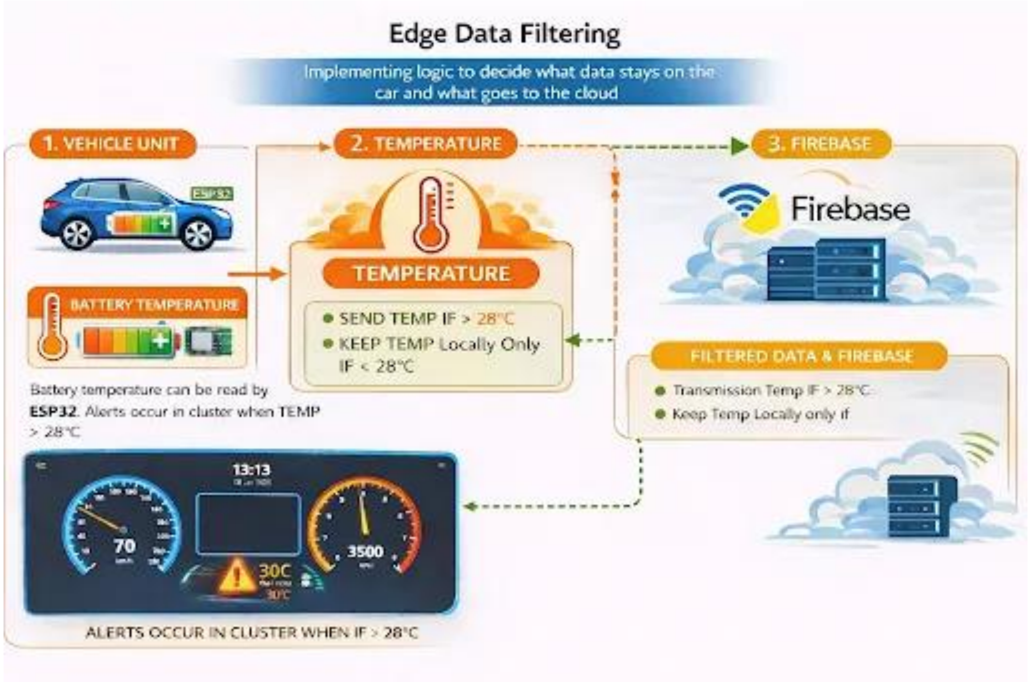
```
#include <Arduino.h>
#include <WiFi.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Firebase_ESP_Client.h>

#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"

/* ================= WiFi Credentials ================= */
#define WIFI_SSID        "iSprout-NRE"
#define WIFI_PASSWORD    "Isprout@n-202$"

/* ================= Firebase Credentials ================= */
#define API_KEY          "AIzaSyCR_FwvqqMGctW9i6MNn4ZUAGcuIjxPAqQ"
#define DATABASE_URL     "https://edge-data-filtering-default-rtdb.asia-
southeast1.firebasedatabase.app/"
#define USER_EMAIL       "tejaswini.kopperla@votarytech.com"
#define USER_PASSWORD    "Votarytech@2025"

/* ================= Temperature Sensor ================= */
#define ONE_WIRE_BUS     15
#define TEMP_THRESHOLD   28.0

/* ================= Indicator Switch Pins ================= */
#define LEFT_INDICATOR_PIN   4
#define RIGHT_INDICATOR_PIN  5

/* ================= Objects ================= */
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

/* ================= Timing ================= */
unsigned long lastSend = 0;
const unsigned long interval = 5000;
```

```
/* ================= Alert Tracking ================= */
int highTempCount = 0;
bool wasAboveThreshold = false;

/* ================= Diagnostics ================= */
#define DTC_BATTERY_TEMP_SENSOR_FAULT "P0A1A"

bool dtc_active = false;
unsigned long fault_start_time = 0;
const unsigned long FAULT_CONFIRM_TIME = 3000; // ms

void diagnostics_check_temperature(float temperatureC)
{
   bool fault_condition =
       (temperatureC == DEVICE_DISCONNECTED_C ||
        temperatureC < -40.0 ||
        temperatureC > 125.0);

   if (fault_condition)
   {
       if (!dtc_active)
       {
           if (fault_start_time == 0)
               fault_start_time = millis();

           if (millis() - fault_start_time >= FAULT_CONFIRM_TIME)
           {
               dtc_active = true;
               Serial.println("DTC SET: P0A1A - Battery Temp Sensor
Fault");

               Firebase.RTDB.setBool(&fbdo, "/dtc/P0A1A/status", true);
               Firebase.RTDB.setString(
                   &fbdo,
                   "/dtc/P0A1A/description",
                   "Battery Temperature Sensor Fault");
               Firebase.RTDB.setInt(
                   &fbdo,
                   "/dtc/P0A1A/timestamp",
                   millis());
           }
       }
   }
   else
   {
       fault_start_time = 0;

       if (dtc_active)
       {
           Serial.println("DTC CLEARED: P0A1A");

           Firebase.RTDB.setBool(&fbdo, "/dtc/P0A1A/status", false);
           Firebase.RTDB.setString(
               &fbdo,
               "/dtc/P0A1A/description",
               "Battery Temperature Sensor OK");
       }

       dtc_active = false;
   }
}

void setup()
{
   Serial.begin(115200);

   pinMode(LEFT_INDICATOR_PIN, INPUT_PULLUP);
   pinMode(RIGHT_INDICATOR_PIN, INPUT_PULLUP);

   /* ---------- WiFi ---------- */
   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
   while (WiFi.status() != WL_CONNECTED)
       delay(500);

   /* ---------- Temperature Sensor ---------- */
   sensors.begin();

   /* ---------- Firebase ---------- */
   config.api_key = API_KEY;
   config.database_url = DATABASE_URL;
   auth.user.email = USER_EMAIL;
   auth.user.password = USER_PASSWORD;

   Firebase.begin(&config, &auth);
   Firebase.reconnectWiFi(true);
}

void loop()
{
   if (millis() - lastSend < interval)
       return;

   lastSend = millis();

   /* ================= TEMPERATURE ================= */
   sensors.requestTemperatures();
   float temperatureC = sensors.getTempCByIndex(0);

   /* Always send the temperature to Firebase, even if disconnected */
   Firebase.RTDB.setFloat(&fbdo, "/sensor/temperature", temperatureC);

   if (temperatureC == DEVICE_DISCONNECTED_C)
   {
       Serial.println("DS18B20 not detected");

       Firebase.RTDB.setBool(
           &fbdo,
           "/diagnostics/battery_temp_sensor/status",
           true);
       Firebase.RTDB.setString(
           &fbdo,
           "/diagnostics/battery_temp_sensor/msg",
           "DS18B20 not detected");

       diagnostics_check_temperature(DEVICE_DISCONNECTED_C);
   }
   else
   {
       Serial.print("Temperature: ");
```

```
    Serial.print(temperatureC);
    Serial.println(" °C");

    diagnostics_check_temperature(temperatureC);

    /* Clear sensor diagnostic when OK */
    Firebase.RTDB.setBool(
        &fbdo,
        "/diagnostics/battery_temp_sensor/status",
        false);
    Firebase.RTDB.setString(
        &fbdo,
        "/diagnostics/battery_temp_sensor/msg",
        "Sensor OK");
}

/* ================= THRESHOLD ================= */
bool isAboveThreshold = (temperatureC > TEMP_THRESHOLD);

if (isAboveThreshold && !wasAboveThreshold)
    highTempCount++;

wasAboveThreshold = isAboveThreshold;

if (isAboveThreshold)
{
    Firebase.RTDB.setBool(
        &fbdo,
        "/alerts/high_temp/status",
        true);
    Firebase.RTDB.setFloat(
        &fbdo,
        "/alerts/high_temp/value",
        temperatureC);

    Firebase.RTDB.setString(
        &fbdo,
        "/alerts/high_temp/msg",
        (highTempCount >= 3)
            ? "Service Required: Battery Temperature High"
```

```
                : "Temperature crossed 28C");
    }
    else
    {
        Firebase.RTDB.setBool(
            &fbdo,
            "/alerts/high_temp/status",
            false);
    }

    /* ================= INDICATORS ================= */
    bool leftOn   = (digitalRead(LEFT_INDICATOR_PIN) == LOW);
    bool rightOn  = (digitalRead(RIGHT_INDICATOR_PIN) == LOW);
    bool hazardOn = (leftOn && rightOn);

    int indicatorStatus = hazardOn ? 3 : (leftOn ? 1 : (rightOn ? 2 : 0));

    Firebase.RTDB.setInt(
        &fbdo,
        "/vehicle/indicator/status",
        indicatorStatus);
    Firebase.RTDB.setBool(
        &fbdo,
        "/vehicle/indicator/left_indicator",
        leftOn);
    Firebase.RTDB.setBool(
        &fbdo,
        "/vehicle/indicator/right_indicator",
        rightOn);
    Firebase.RTDB.setBool(
        &fbdo,
        "/vehicle/indicator/hazard",
        hazardOn);
}
```

}


Edge Data Filtering
Implementing logic to decide what data stays on the car and what goes to the cloud
1. VEHICLE UNIT
ESP32
BATTERY TEMPERATURE
Battery temperature can be read by ESP32. Alerts occur in cluster when TEMP > 28°C
2. TEMPERATURE
TEMPERATURE
SEND TEMP IF > 28°C
KEEP TEMP Locally Only IF < 28°C
3. FIREBASE
Firebase
FILTERED DATA & FIREBASE
Transmission Temp IF > 28°C
Keep Temp Locally only if
13:13
70
3500
30C
30°C
ALERTS OCCUR IN CLUSTER WHEN IF > 28°C

Overall Architecture (Simple View)

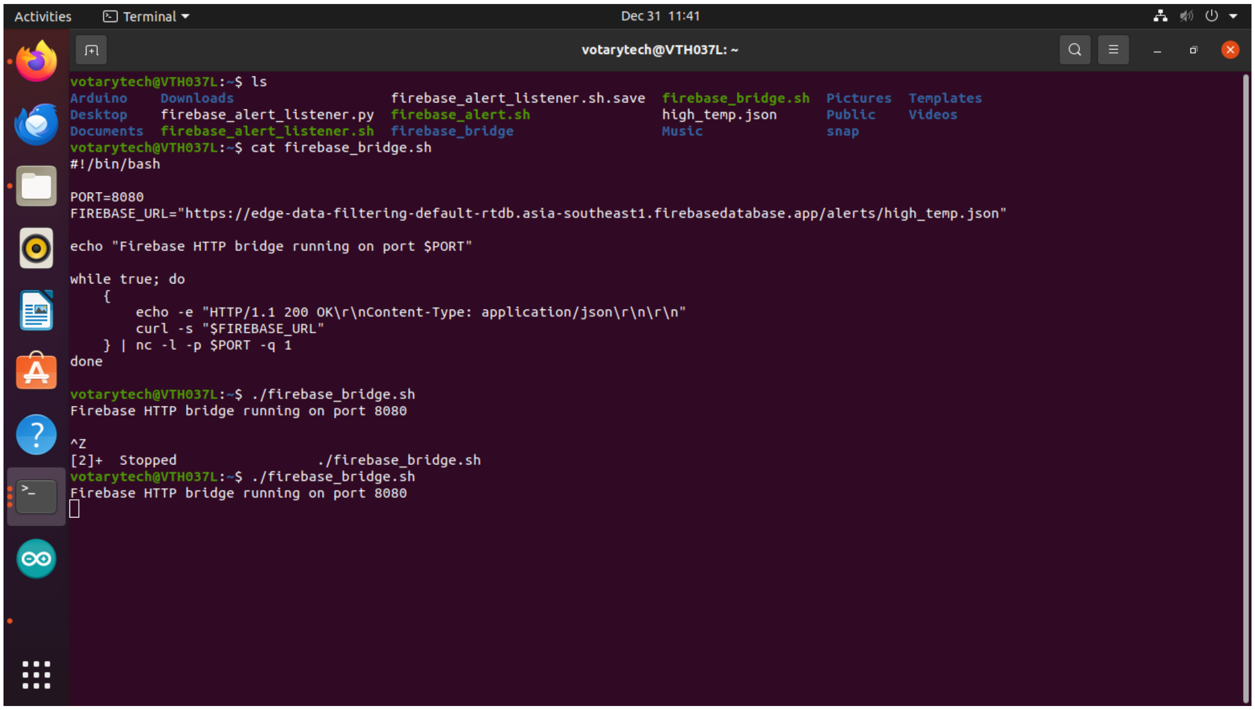
```
DS18B20
   ↓
ESP32 ──(Wi-Fi)──► Firebase Realtime Database
                                ↓
                        OKT507-C (Linux)
                         (reads alert)
                                ↓
                          Display Alert
```

OKT507-C (wget http)
↓
PC / Laptop (curl https)
↓
Firebase (https)

OKT507-C ──wget http──► PC / Laptop ──curl https──► Firebase

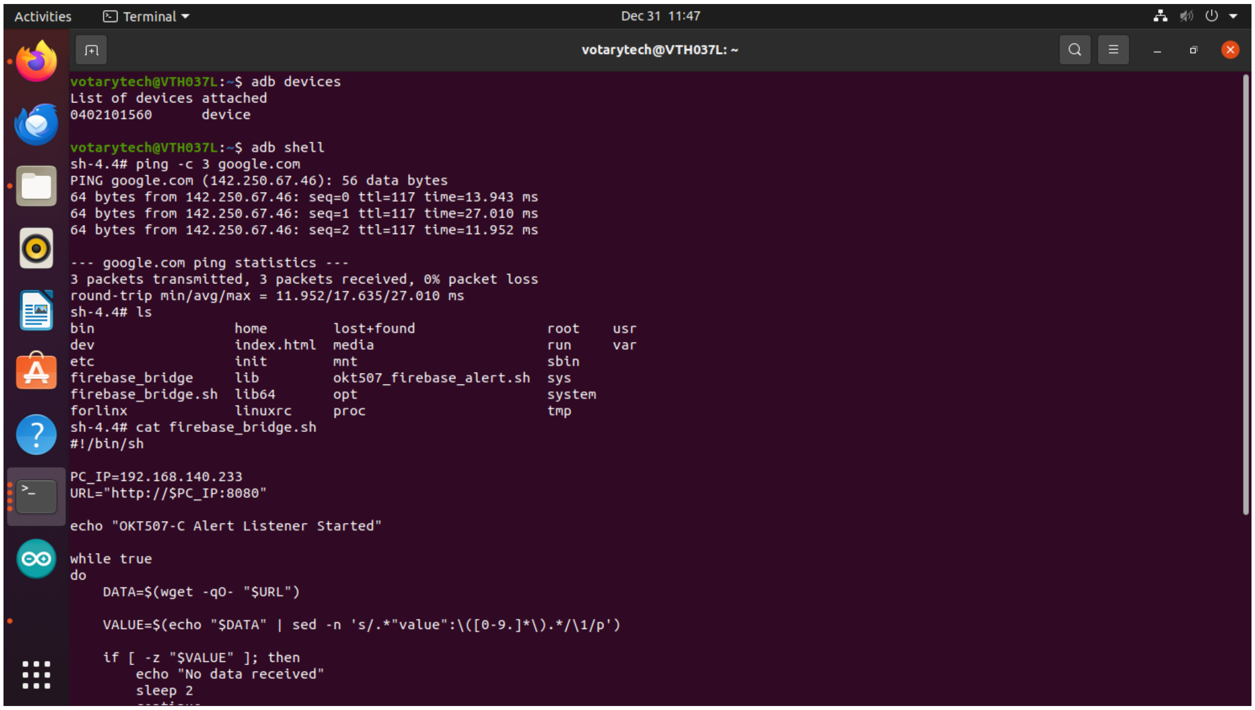
Bridge MUST run on your Ubuntu PC
NOT inside adb shell
NOT on OKT507-C

We can do this steps on PC terminal,finally firebase url connect and run on Firebase HTTP bridge running on port 8080.



```
votarytech@VTH037L:~$ ls
Arduino     Downloads                  firebase_alert_listener.sh.save  firebase_bridge.sh  Pictures  Templates
Desktop     firebase_alert_listener.py  firebase_alert.sh                high_temp.json      Public    Videos
Documents   firebase_alert_listener.sh  firebase_bridge                  Music               snap
votarytech@VTH037L:~$ cat firebase_bridge.sh
#!/bin/bash

PORT=8080
FIREBASE_URL="https://edge-data-filtering-default-rtdb.asia-southeast1.firebasedatabase.app/alerts/high_temp.json"

echo "Firebase HTTP bridge running on port $PORT"

while true; do
    {
        echo -e "HTTP/1.1 200 OK\r\nContent-Type: application/json\r\n\r\n"
        curl -s "$FIREBASE_URL"
    } | nc -l -p $PORT -q 1
done

votarytech@VTH037L:~$ ./firebase_bridge.sh
Firebase HTTP bridge running on port 8080

^Z
[2]+  Stopped                 ./firebase_bridge.sh
votarytech@VTH037L:~$ ./firebase_bridge.sh
Firebase HTTP bridge running on port 8080
```

Follows steps on OKT507-c board side



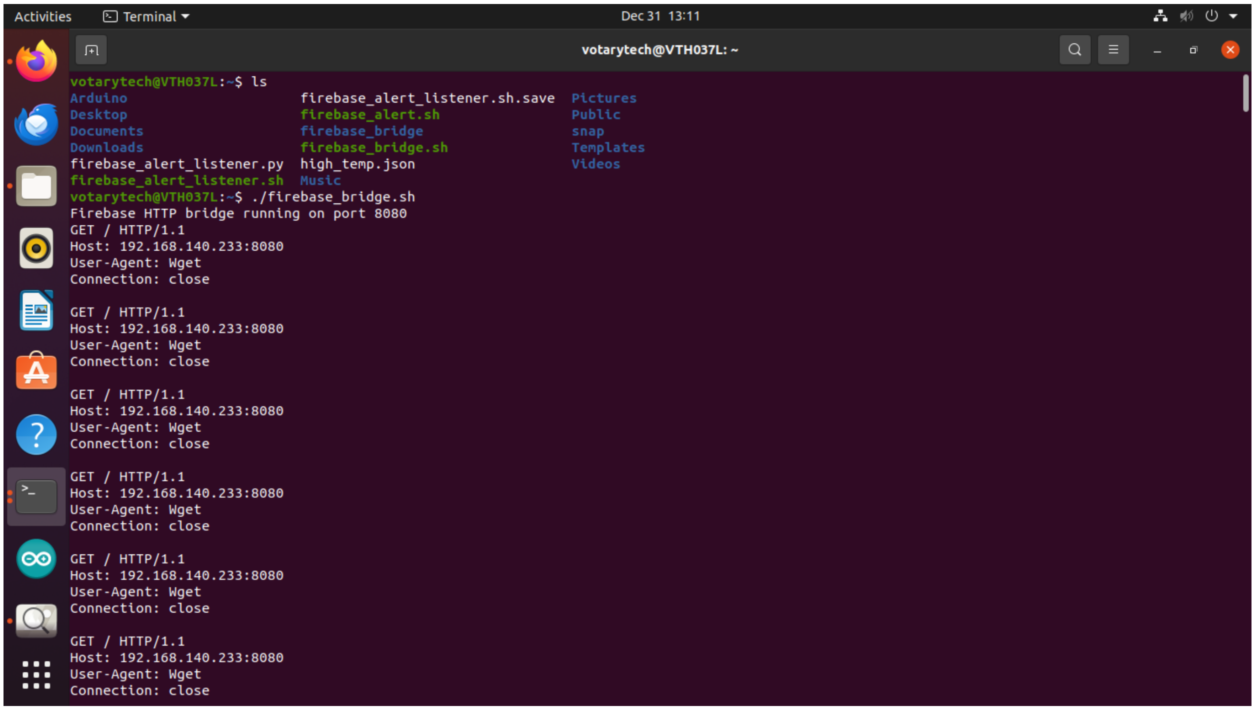
```
votarytech@VTH037L:~$ adb devices
List of devices attached
0402101560      device

votarytech@VTH037L:~$ adb shell
sh-4.4# ping -c 3 google.com
PING google.com (142.250.67.46): 56 data bytes
64 bytes from 142.250.67.46: seq=0 ttl=117 time=13.943 ms
64 bytes from 142.250.67.46: seq=1 ttl=117 time=27.010 ms
64 bytes from 142.250.67.46: seq=2 ttl=117 time=11.952 ms

--- google.com ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 11.952/17.635/27.010 ms
sh-4.4# ls
bin                 home        lost+found                root    usr
dev                 index.html  media                     run     var
etc                 init        mnt                       sbin
firebase_bridge     lib         okt507_firebase_alert.sh  sys
firebase_bridge.sh  lib64       opt                       system
forlinx             linuxrc     proc                      tmp
sh-4.4# cat firebase_bridge.sh
#!/bin/sh

PC_IP=192.168.140.233
URL="http://$PC_IP:8080"

echo "OKT507-C Alert Listener Started"

while true
do
    DATA=$(wget -qO- "$URL")

    VALUE=$(echo "$DATA" | sed -n 's/.*"value":\([0-9.]*\).*/\1/p')

    if [ -z "$VALUE" ]; then
        echo "No data received"
        sleep 2
        continue
```

In firebase.sh file we can write the code

#!/bin/sh

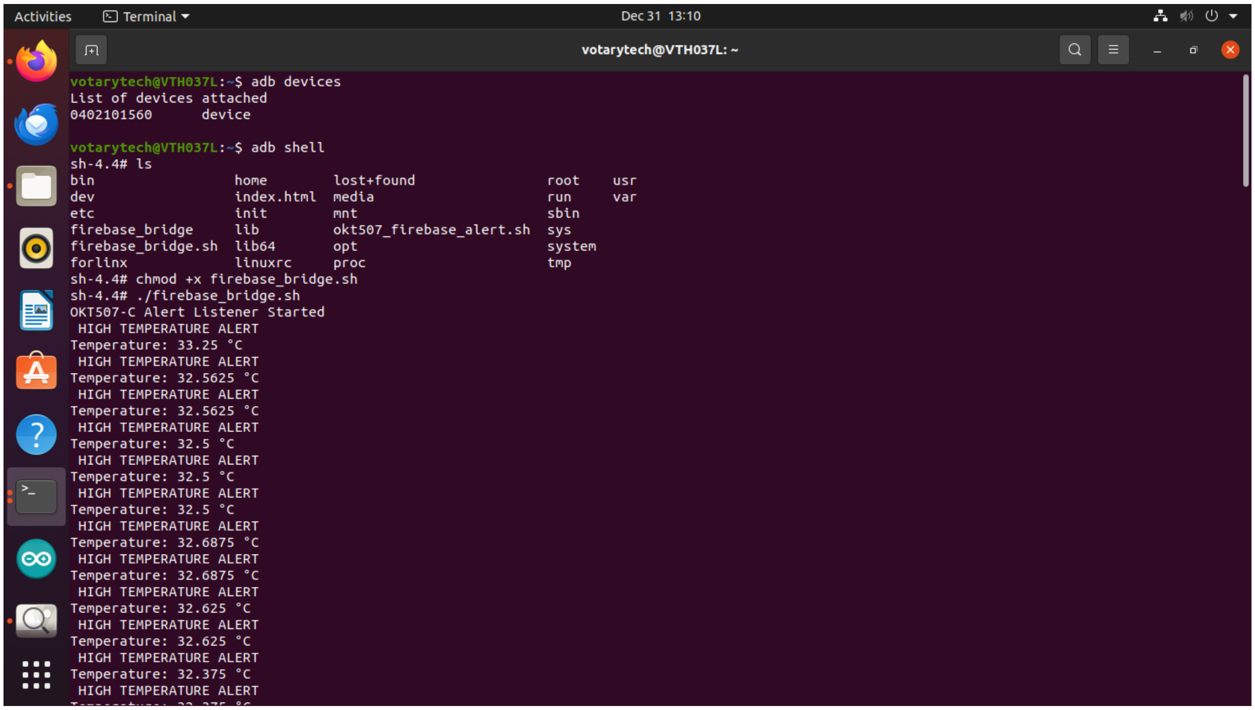
```
PC_IP=192.168.140.233
URL="http://$PC_IP:8080"

echo "OKT507-C Alert Listener Started"

while true
do
        DATA=$(wget -qO- "$URL")

        VALUE=$(echo "$DATA" | sed -n 's/.*"value":\([0-9.]*\).*/\1/p')

        if [ -z "$VALUE" ]; then
        echo "No data received"
        sleep 2
        continue
        fi

        VALUE_INT=${VALUE%.*}

        if [ "$VALUE_INT" -gt 28 ]; then
        echo " HIGH TEMPERATURE ALERT "
        echo "Temperature: $VALUE °C"
        else
        echo "Temperature Normal: $VALUE °C"
        fi

        sleep 2
done
```

Then follow below codes for giving permissions and run the file



```
votarytech@VTH037L:~$ ls
Arduino                      firebase_alert_listener.sh.save  Pictures
Desktop                      firebase_alert.sh                Public
Documents                    firebase_bridge                  snap
Downloads                    firebase_bridge.sh               Templates
firebase_alert_listener.py   high_temp.json                   Videos
firebase_alert_listener.sh   Music
votarytech@VTH037L:~$ ./firebase_bridge.sh
Firebase HTTP bridge running on port 8080
GET / HTTP/1.1
Host: 192.168.140.233:8080
User-Agent: Wget
Connection: close

GET / HTTP/1.1
Host: 192.168.140.233:8080
User-Agent: Wget
Connection: close

GET / HTTP/1.1
Host: 192.168.140.233:8080
User-Agent: Wget
Connection: close

GET / HTTP/1.1
Host: 192.168.140.233:8080
User-Agent: Wget
Connection: close

GET / HTTP/1.1
Host: 192.168.140.233:8080
User-Agent: Wget
Connection: close

GET / HTTP/1.1
Host: 192.168.140.233:8080
User-Agent: Wget
Connection: close
```

After running firebase HTTP port on PC then go to OKTboard terminal



```
votarytech@VTH037L:~$ adb devices
List of devices attached
0402101560      device

votarytech@VTH037L:~$ adb shell
sh-4.4# ls
bin                 home        lost+found                root    usr
dev                 index.html  media                     run     var
etc                 init        mnt                       sbin
firebase_bridge     lib         okt507_firebase_alert.sh  sys
firebase_bridge.sh  lib64       opt                       system
forlinx             linuxrc     proc                      tmp
sh-4.4# chmod +x firebase_bridge.sh
sh-4.4# ./firebase_bridge.sh
OKT507-C Alert Listener Started
 HIGH TEMPERATURE ALERT
Temperature: 33.25 °C
 HIGH TEMPERATURE ALERT
Temperature: 32.5625 °C
 HIGH TEMPERATURE ALERT
Temperature: 32.5625 °C
 HIGH TEMPERATURE ALERT
Temperature: 32.5 °C
 HIGH TEMPERATURE ALERT
Temperature: 32.5 °C
 HIGH TEMPERATURE ALERT
Temperature: 32.5 °C
 HIGH TEMPERATURE ALERT
Temperature: 32.6875 °C
 HIGH TEMPERATURE ALERT
Temperature: 32.6875 °C
 HIGH TEMPERATURE ALERT
Temperature: 32.625 °C
 HIGH TEMPERATURE ALERT
Temperature: 32.625 °C
 HIGH TEMPERATURE ALERT
Temperature: 32.375 °C
 HIGH TEMPERATURE ALERT
Temperature: 32.375 °C
```