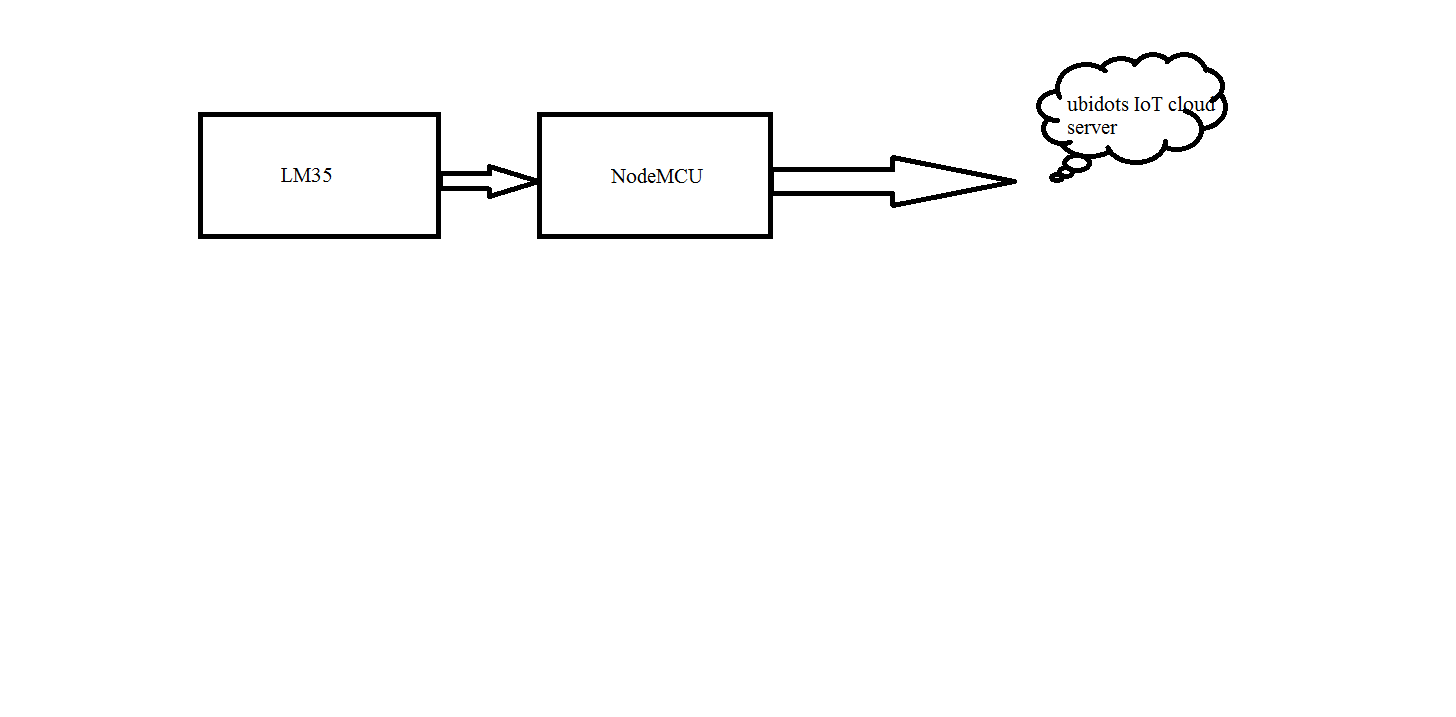
**GROUP 9**

**EXPERIMENT 8A**

**OBJECTIVE:** LM35 SENSOR DATA PUBLISH IN UBIDOTS USING HTTP

**BLOCK DIAGRAM:**

****

**FIG 1A**

**EXPLANATION:**

LM35 sensor is connected to the ubidots IoT cloud server via ESP8266 wi-fi module in NodeMCU. The temperature data from the sensor is published in the ubidots cloud server via the wi-fi module. A new device is created in the server where the temperature value can be seen.

**APPARATUS:**

* LM35 temperature sensor
* NodeMCU
* Jumper wires
* ubidots IoT cloud server
* PC

**PROGRAM:**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Include Libraries

Send LM35 value

\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ESP8266WiFi.h>;

#include <WiFiClient.h>;

#include "Ubidots.h"

const char\* UBIDOTS\_TOKEN = "BBFF-XJNKbcuknuXnr7vWDdkmVY5EPdJ8Po";

const char\* WIFI\_SSID = "Codermaker"; // Put here your Wi-Fi SSID

const char\* WIFI\_PASS = "babi1pal"; // Put here your Wi-Fi password

Ubidots ubidots(UBIDOTS\_TOKEN, UBI\_HTTP);

int val;

int tempPin = A0;

WiFiClient client;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Auxiliar Functions

\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Put here your auxiliar functions

void setup() {

Serial.begin(115200);

ubidots.wifiConnect(WIFI\_SSID, WIFI\_PASS);

// ubidots.setDebug(true); // Uncomment this line for printing debug messages

}

void loop() {

val = analogRead(tempPin);

float mv = ( val / 1024.0) \* 3300;

float cel = mv / 10;

Serial.println(cel);

ubidots.add("abcd", cel);// Change for your variable name

bool bufferSent = false;

bufferSent = ubidots.send(); // Will send data to a device label that matches the device Id

if (bufferSent) {

// Do something if values were sent properly

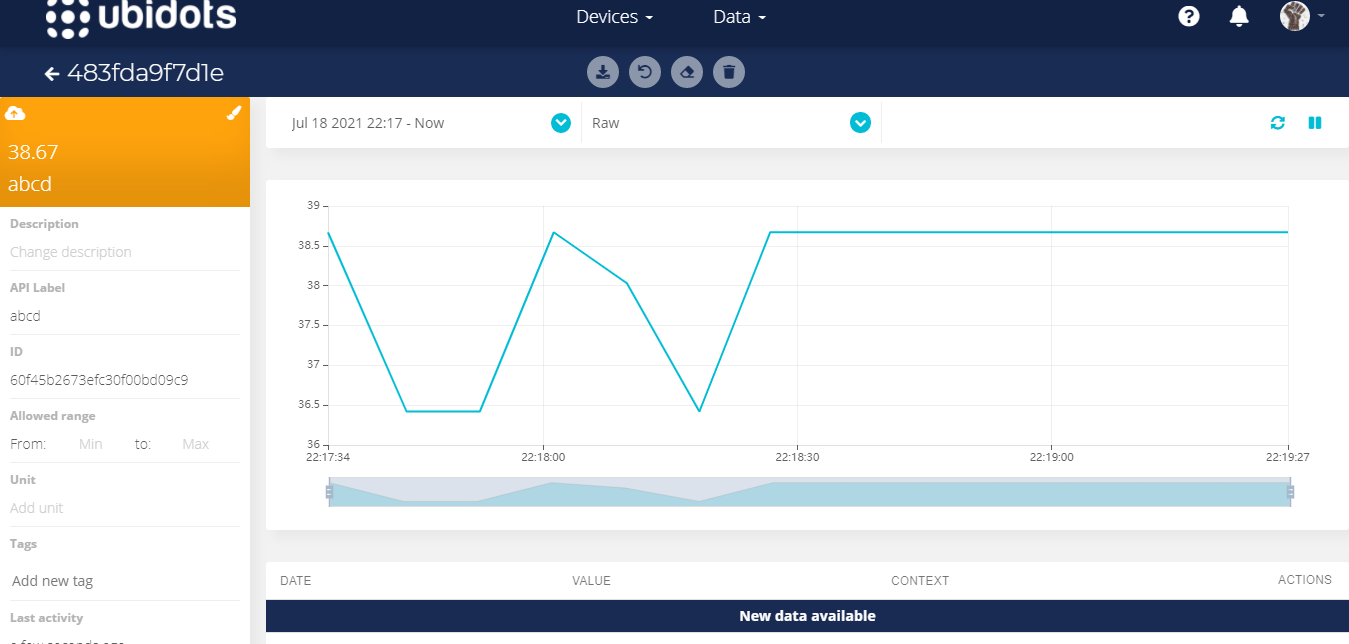
Serial.println("Values sent by the device");

}

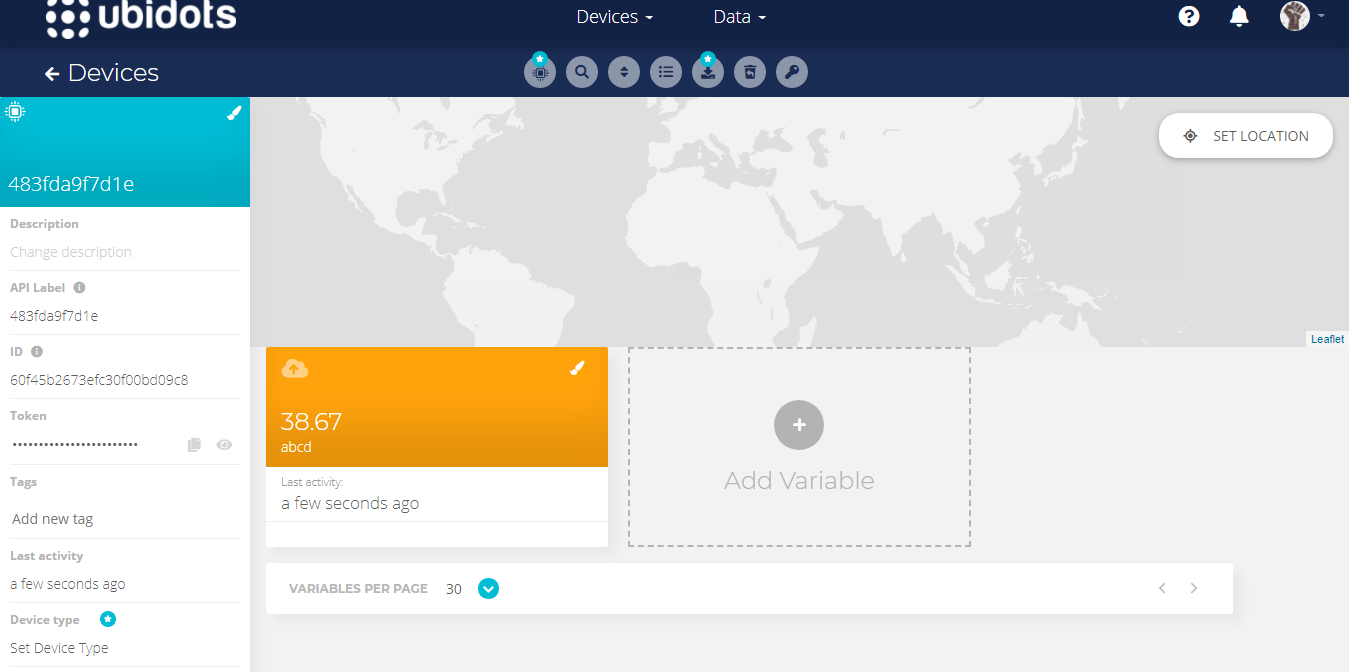
delay(5000);

}

**RESULT:**



**Fig 1B**

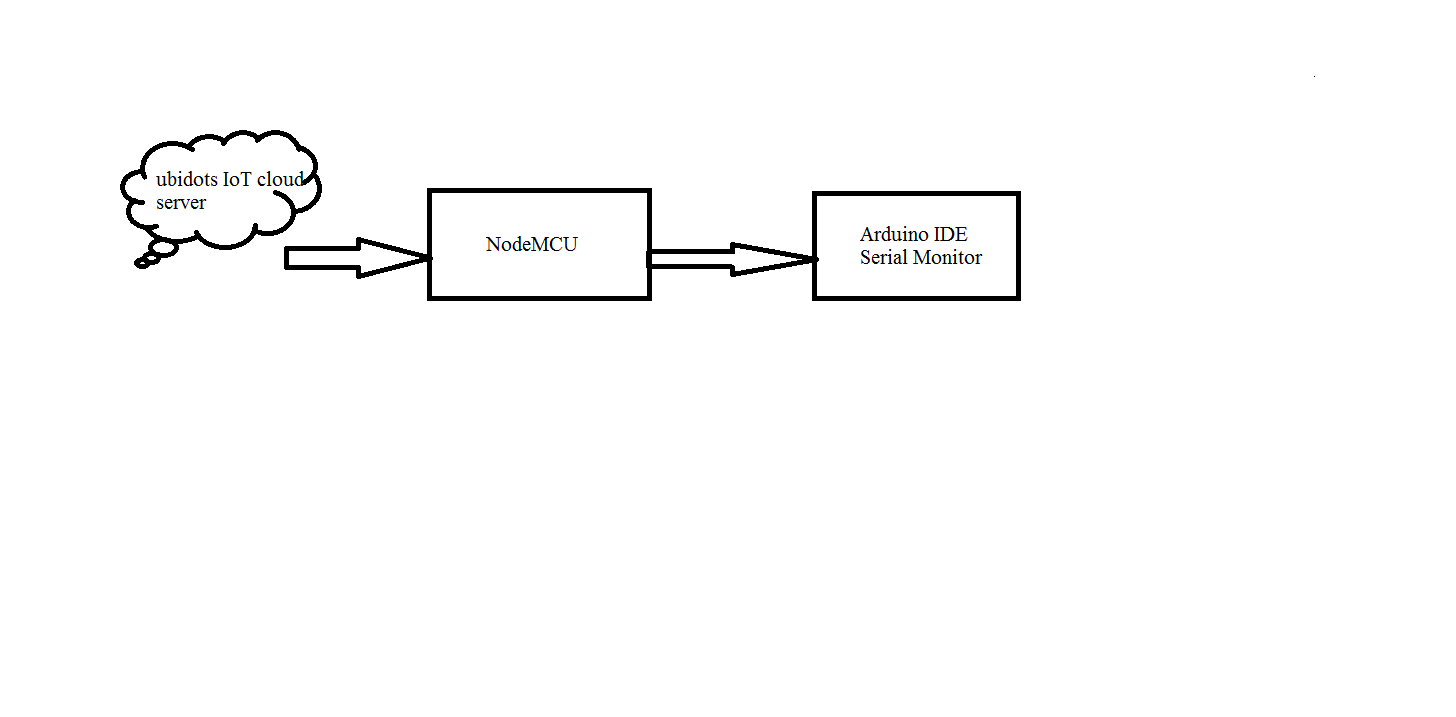


**Fig 1C**

**EXPERIMENT 8B**

**OBJECTIVE:** LM35 DATA SUBSCRIBE FROM UBIDOTS AND DISPLAY IN ARDUINO IDE SERIAL MONITOR

**BLOCK DIAGRAM:**

****

**Fig: 2A**

**EXPLANATION:**

The LM35 temperature value stored in the ubidots cloud server is accessed, read and displayed in Arduino IDE Serial Monitor via the ESP8266 module in NodeMCU.

**APPARATUS:**

* LM35 sensor
* Arduino IDE
* ubidots cloud server
* Jumper wires
* NodeMCU

**PROGRAM:**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Include Libraries

get value from ubidots

\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ESP8266WiFi.h>;

#include <WiFiClient.h>;

#include "Ubidots.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Define Instances and Constants

\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

const char\* UBIDOTS\_TOKEN = "BBFF-XJNKbcuknuXnr7vWDdkmVY5EPdJ8Po";

const char\* WIFI\_SSID = "Codermaker"; // Put here your Wi-Fi SSID

const char\* WIFI\_PASS = "babi1pal"; // Put here your Wi-Fi password

Ubidots ubidots(UBIDOTS\_TOKEN, UBI\_HTTP);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Auxiliar Functions

\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Put here your auxiliar functions

void setup() {

Serial.begin(115200);

ubidots.wifiConnect(WIFI\_SSID, WIFI\_PASS);

}

void loop() {

/\* Obtain last value from a variable as float using HTTP \*/

float value = ubidots.get("dravida", "abcd");

// Evaluates the results obtained

if (value != ERROR\_VALUE) {

Serial.print("Value:");

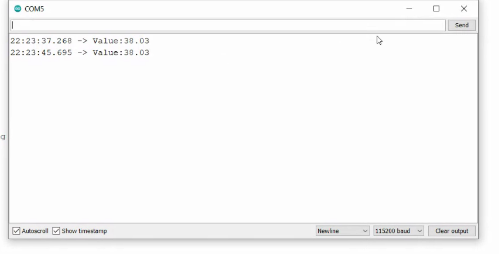
Serial.println(value);

}

delay(5000);

}

**RESULT:**



**Fig 2B**