MINI PROJECT

18ITT62 – IOT

TEAM MEMBERS

SANDHIYA G(19ITR079)

VARSHINI P(19ITR103)

VINOTHINI P(19ITR106)

SUBMITTED ON:19.06.2022

DISPLAYING TEMPERATURE AND HUMIDITY IN TELEGRAM

AIM:

To monitor the temperature and humidity using DHT11 sensor and display it in the telegram.

SCOPE:

The scope of this project is to monitor the temperature and humidity without any manual effort. Whenever the temperature needs to check it can be easily done with a text in telegram using /temperature or /humidity.

BRIEF HISTORY:

Whenever we need to check the temperature and humidity, we can check using temperature and humidity by simple text in telegram. This Temperature and Humidity Sensor is extremely sensitive. This Temperature and Humidity sensor is ideal for short-range temperature and humidity detection and can be used to monitor projects.

PROPOSED METHODOLOGY:

First the Arduino IDE needs to be set up. Select tools -> choose board -> then board name -> then choose ESP8266-> after that telegram library should be installed in IDE for that choose Sketch -> choose include library then manage library from that installed universal telegram. Now the IDE is perfectly fit for the project. The next step is to connecting the sensor and node MCU. The sensor is connected to the input of the nodemcu with the help of connecting cables or jumper wires. Connect vcc pin to 3v pin in nodemcu then data pin to D1 and negative pin to ground. Now the node mcu and temperature sensor are connected. Next open telegram app search Bot father, start the chat. Type new bot then choose a name for bot, then create username that should end with a bot, a new chat will be created. Now a token will be generated by the bot father to access HTTP API node mcu.

COMPONENTS REQUIRED:

S.NO	**COMPONENTS**	**NO'S**
1	Node MCU	1
2	DHT11 Temperature and Humidity sensor	1
3	Jumper Wires	As required
4	USB Cable	1

DESCRIPTION:

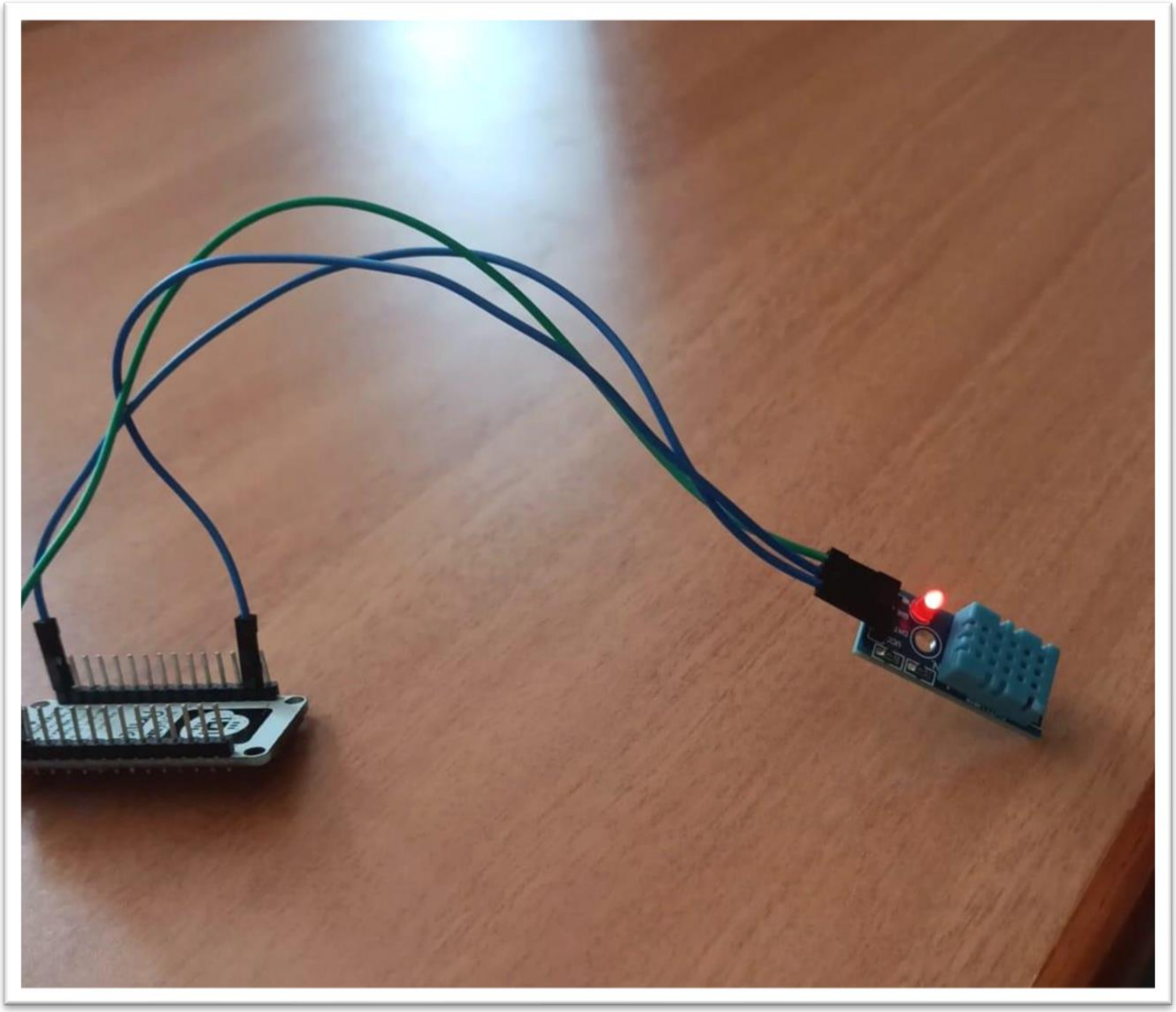
The **NodeMCU** ESP8266 development board comes with the ESP-12E module containing the ESP8266 chip having Ten silica Xtensa 32-bit LX106 RISC microprocessor. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects. NodeMCU can be powered using a Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface**.**

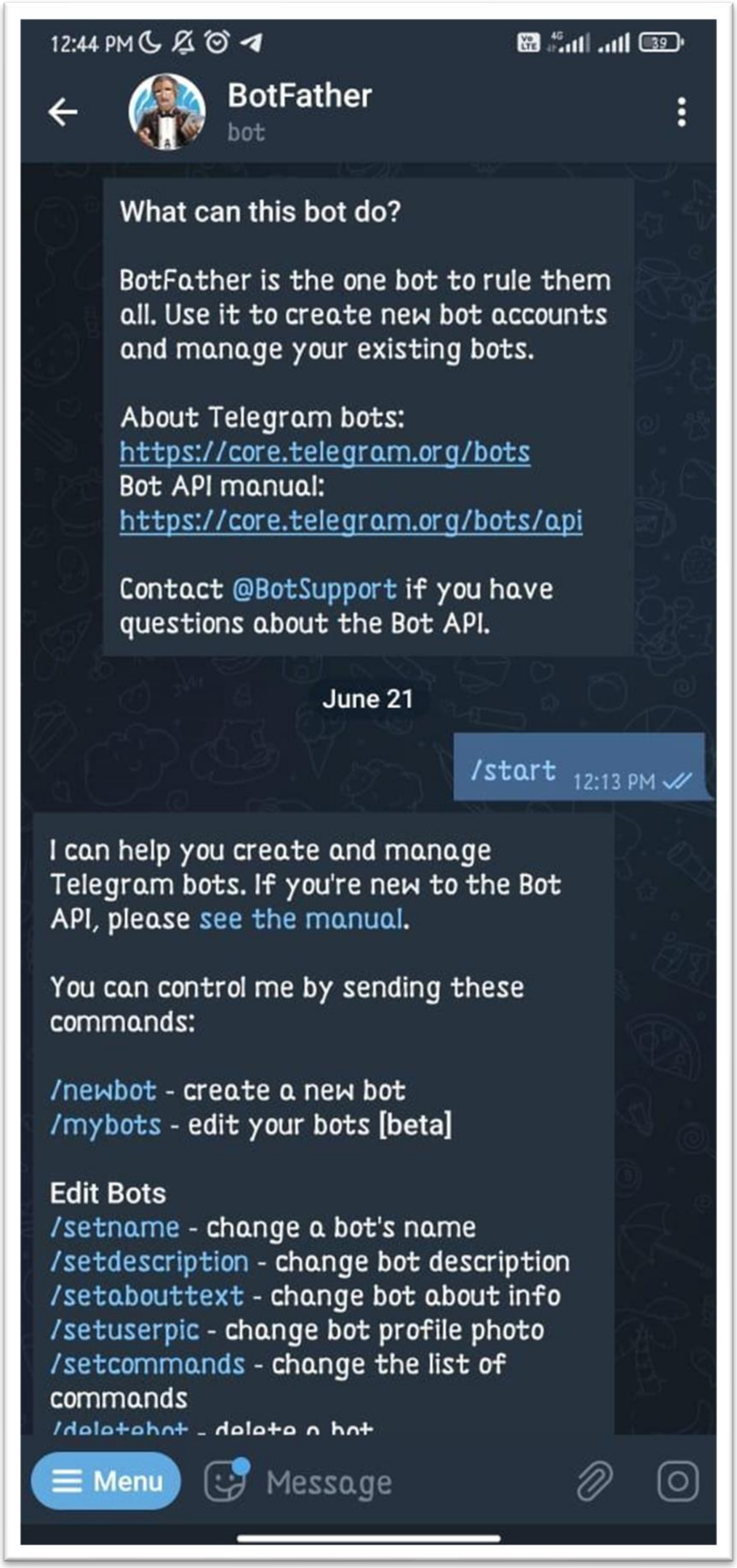
DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz. i.e., it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA. DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controller.

A **JUMPER WIRE** is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

SCREENSHOTS:





What can this bot do?

BotFather is the one bot to rule them all. Use it to create new bot accounts and manage your existing bots.

About Telegram bots:
https://core.telegram.org/bots
Bot API manual:
https://core.telegram.org/bots/api

Contact @BotSupport if you have questions about the Bot API.

June 21

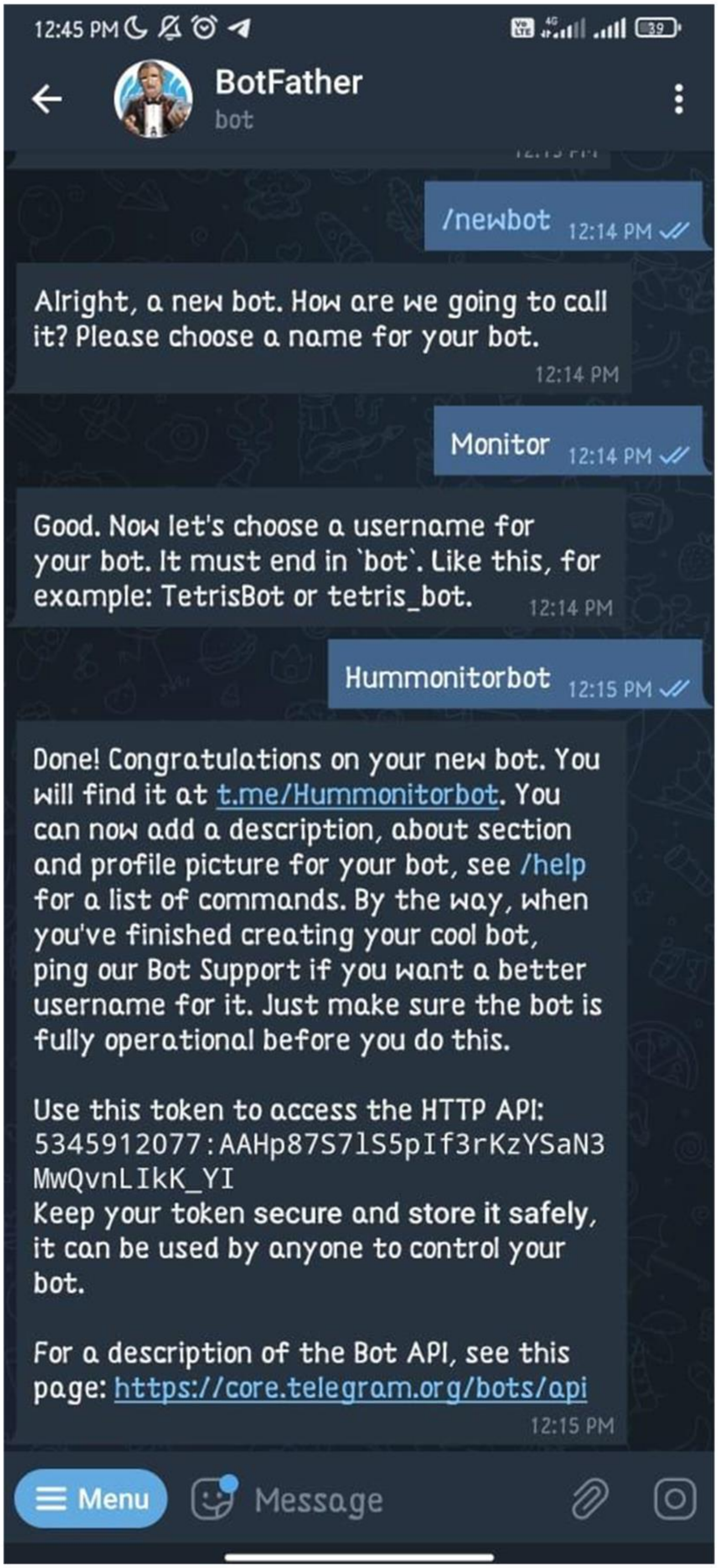
/start 12:13 PM

I can help you create and manage Telegram bots. If you're new to the Bot API, please see the manual.

You can control me by sending these commands:

/newbot - create a new bot
/mybots - edit your bots [beta]

Edit Bots
/setname - change a bot's name
/setdescription - change bot description
/setabouttext - change bot about info
/setuserpic - change bot profile photo
/setcommands - change the list of commands
/deletebot - delete a bot



/newbot 12:14 PM

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 12:14 PM

Monitor 12:14 PM

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot. 12:14 PM

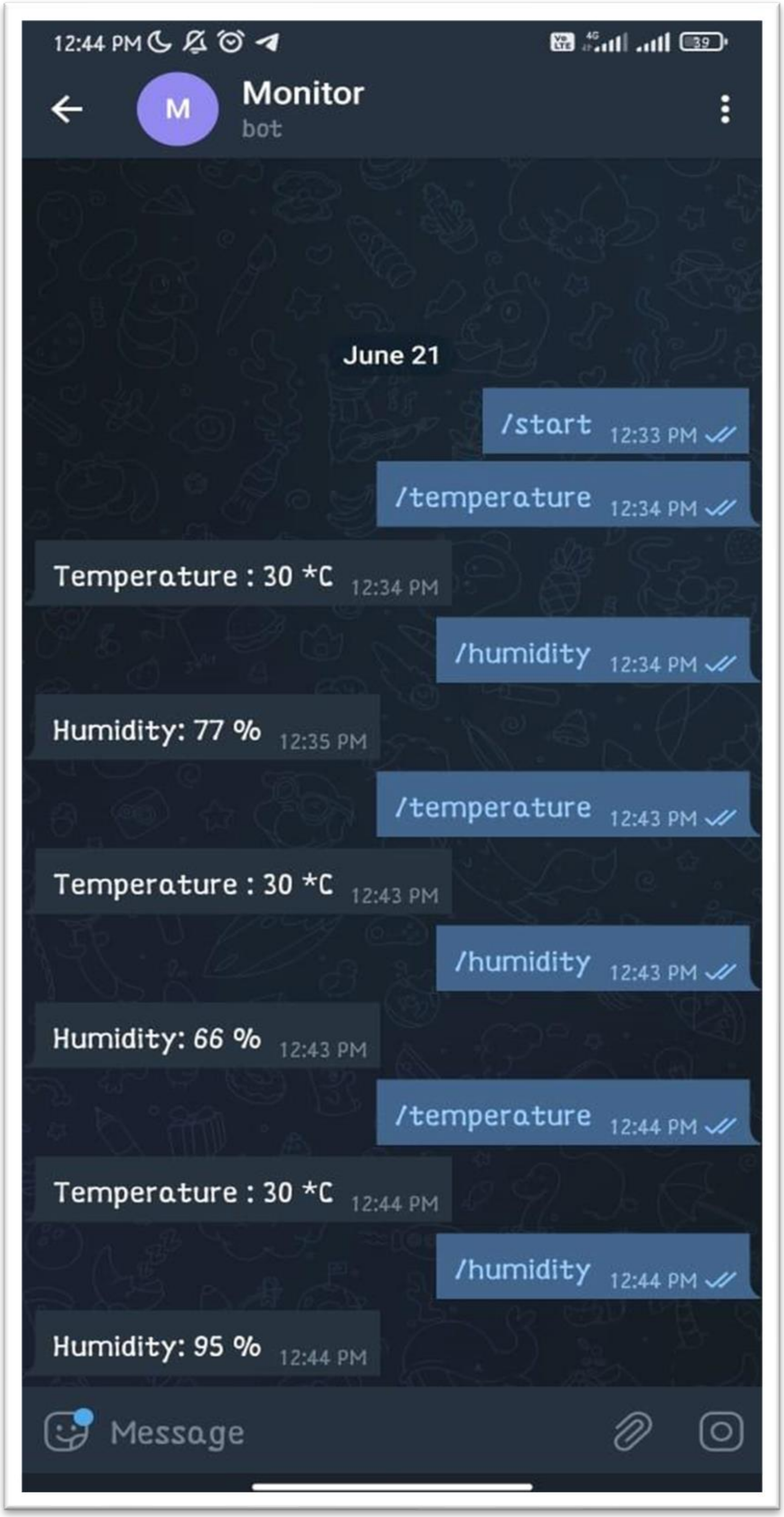
Hummonitorbot 12:15 PM

Done! Congratulations on your new bot. You will find it at t.me/Hummonitorbot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
5345912077:AAHp87S71S5pIf3rKzYSaN3MwQvnLIkK_YI
Keep your token secure and store it safely, it can be used by anyone to control your bot.

For a description of the Bot API, see this page: https://core.telegram.org/bots/api
12:15 PM

OUTPUT:



/start 12:33 PM

/temperature 12:34 PM

Temperature : 30 *C 12:34 PM

/humidity 12:34 PM

Humidity: 77 % 12:35 PM

/temperature 12:43 PM

Temperature : 30 *C 12:43 PM

/humidity 12:43 PM

Humidity: 66 % 12:43 PM

/temperature 12:44 PM

Temperature : 30 *C 12:44 PM

/humidity 12:44 PM

Humidity: 95 % 12:44 PM

CODING:

```
#include <ESP8266WiFi.h>

#include <WiFiClientSecure.h>

#include <UniversalTelegramBot.h>

#include "DHT.h"

#define DHTPIN D1

#define DHTTYPE DHT11 // DHT 11

DHT dht(DHTPIN, DHTTYPE);

// Initialize Wifi connection to the router

char ssid[] = "vinosiva"; // Wifi Name

char password[] = "Vino@2001"; // Wifi Password

// Initialize Telegram BOT

#define BOTtoken
"5345912077:AAHp87S7lS5pIf3rKzYSaN3MwQvnLIkK_YI"
// Bot token from telegram app

WiFiClientSecure client;

UniversalTelegramBot bot(BOTtoken, client);

//Checks for new messages every 1 second.
```

```
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;
void handleNewMessages(int numNewMessages) {
  Serial.println("handleNewMessages");
  Serial.println(String(numNewMessages));
  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    String text = bot.messages[i].text;
    String from_name = bot.messages[i].from_name;
    if (from_name == "") from_name = "Guest";
    if (text == "/temperature") {
      int t = dht.readTemperature();
      String temp = "Temperature : ";
      temp += int(t);
      temp += " *C\n";
      bot.sendMessage(chat_id, temp, "");
    }
    if (text == "/humidity") {
      int h = dht.readHumidity();

      String temp = "Humidity: ";

      temp += int(h);

      temp += " %";

      bot.sendMessage(chat_id, temp, "");

    }

    if (text == "/start") {

      String welcome = "Welcome " + from_name + ".\n";

      welcome += "/temperature : Temperature reading\n";

      welcome += "/humidity : Humiditiy reading\n";

      bot.sendMessage(chat_id, welcome, "Markdown");

    }

  }

}

void setup() {

  Serial.begin(115200);

  dht.begin();
  // This is the simplest way of getting this working
  // if you are passing sensitive information, or controlling
  // something important, please either use certStore or at
  // least client.setFingerPrint
  client.setInsecure();
  // Set WiFi to station mode and disconnect from an AP if it was
Previously
  // connected
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);
  // attempt to connect to Wifi network:
  Serial.print("Connecting Wifi: ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  Serial.println("");

  Serial.println("WiFi connected");

  Serial.print("IP address: ");

  Serial.println(WiFi.localIP());

}

void loop() {

  int t = dht.readTemperature();

  int h = dht.readHumidity();

  if (millis() > lastTimeBotRan + botRequestDelay) {

    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);

    while (numNewMessages) {

      Serial.println("got response");

      handleNewMessages(numNewMessages);

      numNewMessages = bot.getUpdates(bot.last_message_received + 1);

    }

    lastTimeBotRan = millis();
  }
}
```

CONCLUSION:

Thus, the temperature and humidity using DHT11 sensor is monitored and displayed in the telegram successfully.