**HOW TO CREATE A TELEGRAM BOT TO CONROL LED.**

NOW WE ARE GOING TO CREATE A TELEGRAM BOT TO CONTROL LED

LED ON

LED OFF

STATUS OF LED

FIRST INSTALL THE REQUIRED LIBRARIES AND MODULES.

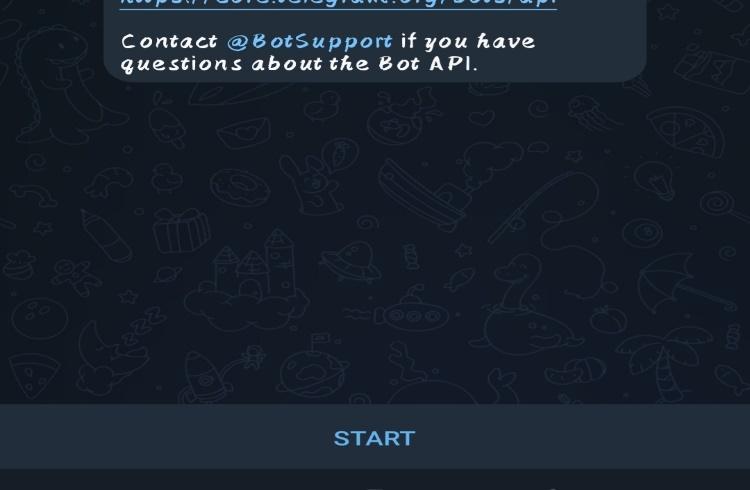
STEP 1:

OPEN TELEGRAM APP > SIGN UP>

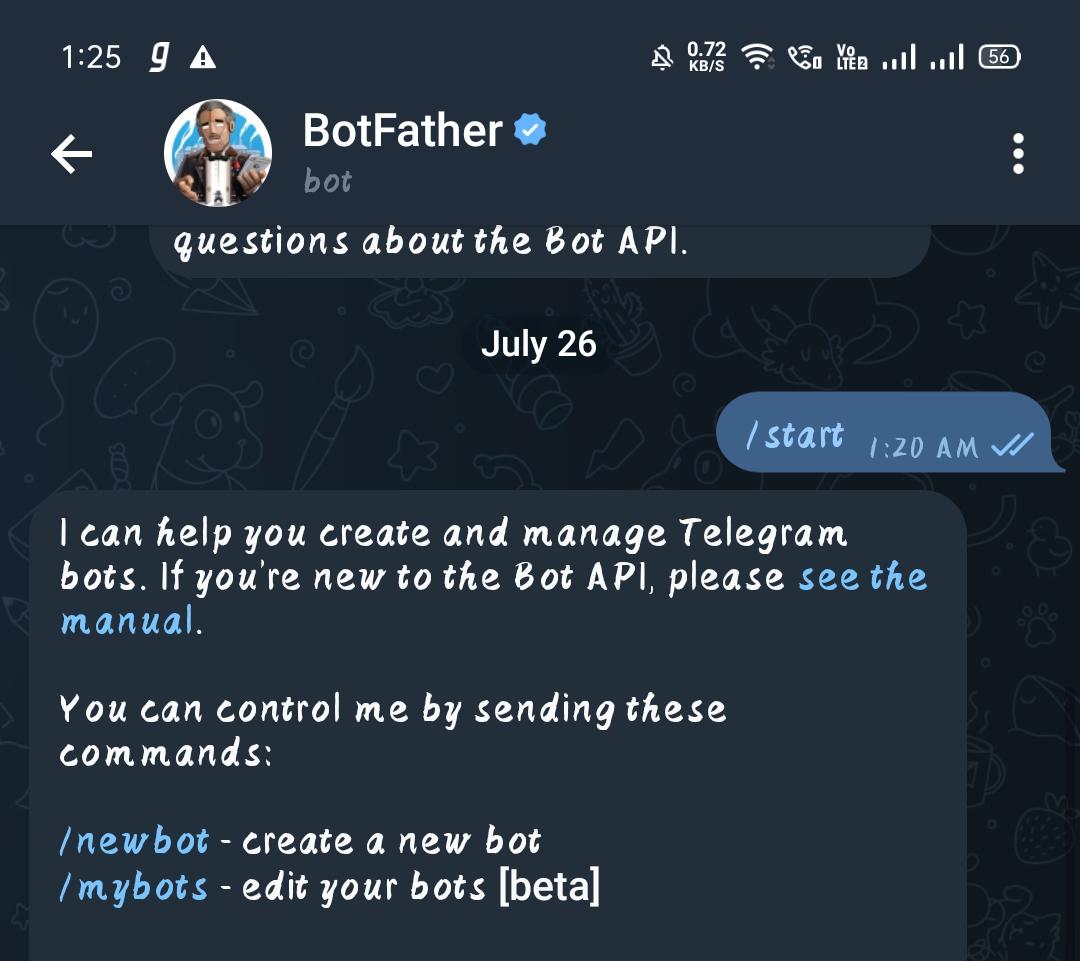


CLICK ON SEARCH BAR ON TOP-RIGHT > SEARCH FOR “BotFather”.

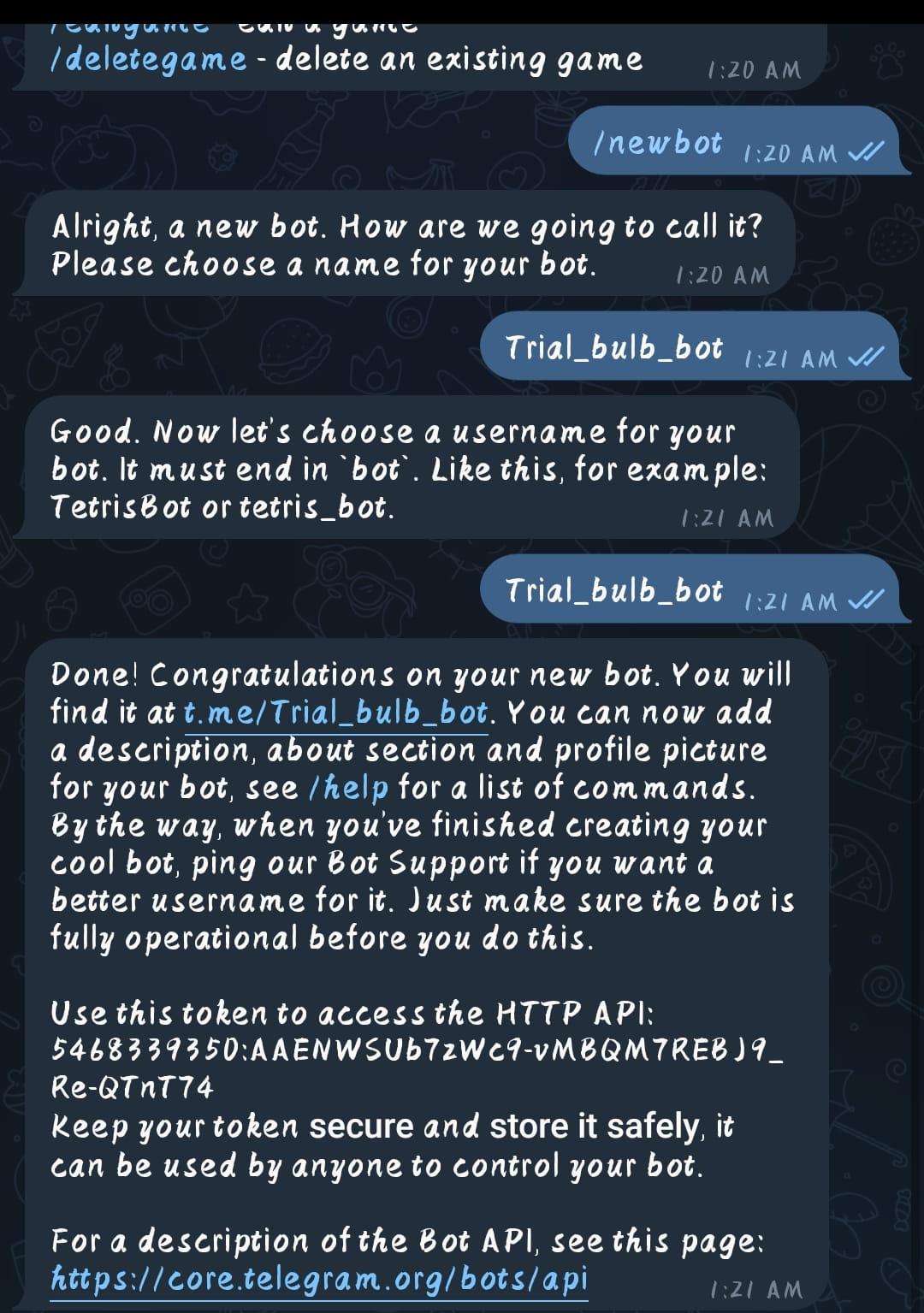
SELECT THE VERIFIED ONE ONLY.

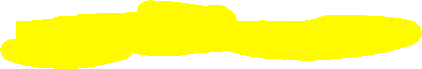


CLICK ON START > A MENU DROPSDOWN > NOW CLICK ON “/newbot” TO CREATE A NEWBOT.



NEXT GIVE THE NAME AND USERNAME TO THE BOT , MAKE SURE IT ENDS WITH “bot” AS FOLLOW.





COPY AND SAVE THE HIGHLIGHTES HTTP API FOR YOUR BOT.



AS IT HAS TO BE PASTED IN THE PROGRAM

NOW LET US PROCEED TO CONNECTIONS PART AND HARDWARE PART.

WE NEED A LED AND A NODEMCU ESP32

CONNECTIONS:

NEGATIVE(LED) 🡪 GND (ESP32)

POSITIVE(LED) 🡪 D15(ESP32)

AND IN ARDUINO IDE .

ENTER THIS PROGRAM..:

#include <WiFi.h>

#include <WiFiClientSecure.h>

#include <UniversalTelegramBot.h>

// Wifi network station credentials

#define WIFI\_SSID "LG"

#define WIFI\_PASSWORD "getjar123"

#define BOT\_TOKEN "5320062699:AAFEzAqmX-KvlRTlTX657XMJ\_izp-UKk9sw"

const unsigned long BOT\_MTBS = 1000; // mean time between scan messages

WiFiClientSecure secured\_client;

UniversalTelegramBot bot(BOT\_TOKEN, secured\_client);

unsigned long bot\_lasttime; // last time messages' scan has been done

const int lightPin = 15;

int lightStatus = 0;

void handleNewMessages(int numNewMessages)

{

Serial.print("handleNewMessages ");

Serial.println(numNewMessages);

for (int i = 0; i < numNewMessages; i++)

{

String chat\_id = 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 == "/lighton")

{

digitalWrite(lightPin, HIGH); // turn the light on (HIGH is the voltage level)

lightStatus = 1;

bot.sendMessage(chat\_id, "Light is ON", "");

}

if (text == "/lightoff")

{

lightStatus = 0;

digitalWrite(lightPin, LOW); // turn the light off (LOW is the voltage level)

bot.sendMessage(chat\_id, "Light is OFF", "");

}

if (text == "/status")

{

if (lightStatus)

{

bot.sendMessage(chat\_id, "Light is ON", "");

}

else

{

bot.sendMessage(chat\_id, "Light is OFF", "");

}

}

if (text == "/start")

{

String welcome = "Telegram Based Home Automation, " + from\_name + ".\n";

welcome += "This is Telegram Bot example.\n\n";

welcome += "/lighton : to switch the light ON\n";

welcome += "/lightoff : to switch the light OFF\n";

welcome += "/status : Returns current status of light\n";

bot.sendMessage(chat\_id, welcome, "Markdown");

}

}

}

void setup()

{

Serial.begin(115200);

Serial.println();

pinMode(lightPin, OUTPUT); // initialize digital lightPin as an output.

delay(10);

digitalWrite(lightPin, LOW); // initialize pin as off (active LOW)

// attempt to connect to Wifi network:

Serial.print("Connecting to Wifi SSID ");

Serial.print(WIFI\_SSID);

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

secured\_client.setCACert(TELEGRAM\_CERTIFICATE\_ROOT); // Add root certificate for api.telegram.org

while (WiFi.status() != WL\_CONNECTED)

{

Serial.print(".");

delay(500);

}

Serial.print("\nWiFi connected. IP address: ");

Serial.println(WiFi.localIP());

Serial.print("Retrieving time: ");

configTime(0, 0, "pool.ntp.org"); // get UTC time via NTP

time\_t now = time(nullptr);

while (now < 24 \* 3600)

{

Serial.print(".");

delay(100);

now = time(nullptr);

}

Serial.println(now);

}

void loop()

{

if (millis() - bot\_lasttime > BOT\_MTBS)

{

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);

}

bot\_lasttime = millis();

}

}

NOW SEE THE OUTPUT AS FOLLOWS:

