LAPORAN PRAKTIK
System iot

Nama : Ludang Prasetyo Nugroho
Nim 225510017
Matkul : Prak system IOT

Teknik Komputer (S1)

PRAKTEK

>>>> Code Program 1

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>

// Wifi network station credentials
#define WIFI_SSID "RPLA_2.4"
#define WIFI_PASSWORD "utdijogja"

// Telegram BOT Token (Get from Botfather)
#define BOT_TOKEN "7650684774:AAGob6Ye1lWLl0ZtvmRbhDjqZAoXHN9a0ik"

#define LED1 19
#define LED2 18
#define LED3 5
#define LED4 17

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

int led1Status = 0;
int led2Status = 0;
int led3Status = 0;
int led4Status = 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";

    // Commands for LED 1
    if (text == "/led1on")
```

```
  {
   digitalWrite(LED1, HIGH);
   led1Status = 1;
   bot.sendMessage(chat_id, "Led1 ON", "");
  }
  if (text == "/led1off")
  {
   digitalWrite(LED1, LOW);
   led1Status = 0;
   bot.sendMessage(chat_id, "Led1 OFF", "");
  }

  // Commands for LED 2
  if (text == "/led2on")
  {
   digitalWrite(LED2, HIGH);
   led2Status = 1;
   bot.sendMessage(chat_id, "Led2 ON", "");
  }
  if (text == "/led2off")
  {
   digitalWrite(LED2, LOW);
   led2Status = 0;
   bot.sendMessage(chat_id, "Led2 OFF", "");
  }

  // Commands for LED 3
  if (text == "/led3on")
  {
   digitalWrite(LED3, HIGH);
   led3Status = 1;
   bot.sendMessage(chat_id, "Led3 ON", "");
  }
  if (text == "/led3off")
  {
   digitalWrite(LED3, LOW);
   led3Status = 0;
   bot.sendMessage(chat_id, "Led3 OFF", "");
  }

  // Commands for LED 4
  if (text == "/led4on")
  {
   digitalWrite(LED4, HIGH);
   led4Status = 1;
   bot.sendMessage(chat_id, "Led4 ON", "");
  }
  if (text == "/led4off")
  {
   digitalWrite(LED4, LOW);
   led4Status = 0;
   bot.sendMessage(chat_id, "Led4 OFF", "");
  }

  if (text == "/status")
```

```
    {
     String statusMessage = "Status:\n";
     statusMessage += "Led1: " + String(led1Status ? "ON" : "OFF") + "\n";
     statusMessage += "Led2: " + String(led2Status ? "ON" : "OFF") + "\n";
     statusMessage += "Led3: " + String(led3Status ? "ON" : "OFF") + "\n";
     statusMessage += "Led4: " + String(led4Status ? "ON" : "OFF") + "\n";
     bot.sendMessage(chat_id, statusMessage, "");
    }

    if (text == "/start")
    {
     String welcome = "Welcome to Universal Arduino Telegram Bot library, " + from_name + ".\n";
     welcome += "This is Flash Led Bot example.\n\n";
     welcome += "/led1on : to switch Led1 ON\n";
     welcome += "/led1off : to switch Led1 OFF\n";
     welcome += "/led2on : to switch Led2 ON\n";
     welcome += "/led2off : to switch Led2 OFF\n";
     welcome += "/led3on : to switch Led3 ON\n";
     welcome += "/led3off : to switch Led3 OFF\n";
     welcome += "/led4on : to switch Led4 ON\n";
     welcome += "/led4off : to switch Led4 OFF\n";
     welcome += "/status : Returns current status of all LEDs\n";
     bot.sendMessage(chat_id, welcome, "");
    }
  }
}

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

  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  delay(10);

  digitalWrite(LED1, LOW);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  digitalWrite(LED4, LOW);

  Serial.print("Connecting to Wifi SSID ");
  Serial.print(WIFI_SSID);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  secured_client.setCACert(TELEGRAM_CERTIFICATE_ROOT);

  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");
  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();
  }
}
```

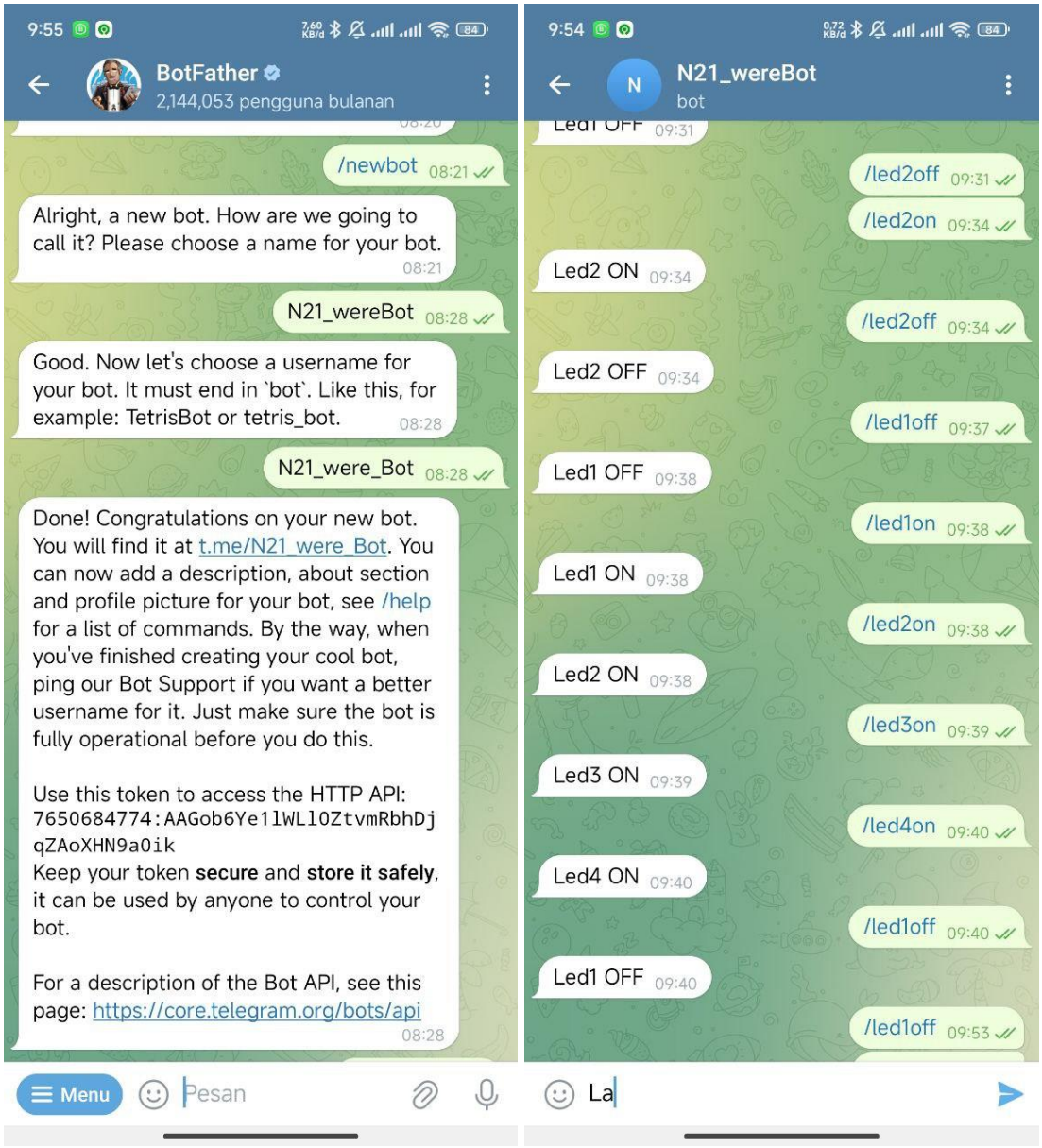
Serial monitor

```
E (2777) wifi:Association refused temporarily, comeback time 0 mSec
E (2784) wifi:Association refused temporarily, comeback time 0 mSec
E (2791) wifi:Association refused temporarily, comeback time 0 mSec
E (2799) wifi:Association refused temporarily, comeback time 0 mSec
E (2806) wifi:Association refused temporarily, comeback time 0 mSec
E (2813) wifi:Association refused temporarily, comeback time 0 mSec
E (2820) wifi:Association refused temporarily, comeback time 0 mSec
.........
WiFi connected. IP address: 172.18.104.149
Retrieving time: ........................................1731638291
got response
handleNewMessages 1
got response
handleNewMessages 1
got response
handleNewMessages 1
got response
handleNewMessages 1
```

Code berjalan sesuwai arahan saat menerima pesan dari telegram akan di munculkan di serial monitor

LAPORAN PRAKTIK
System iot

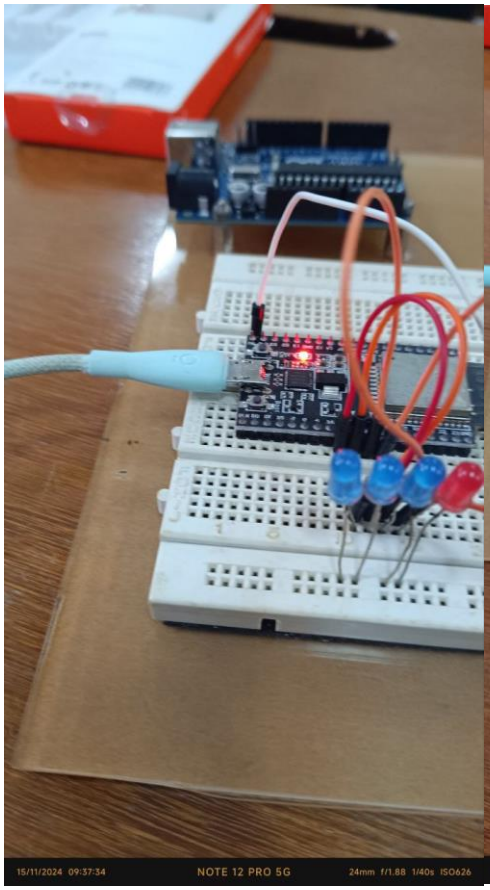
Membuat bod dan mengambil toket Bod

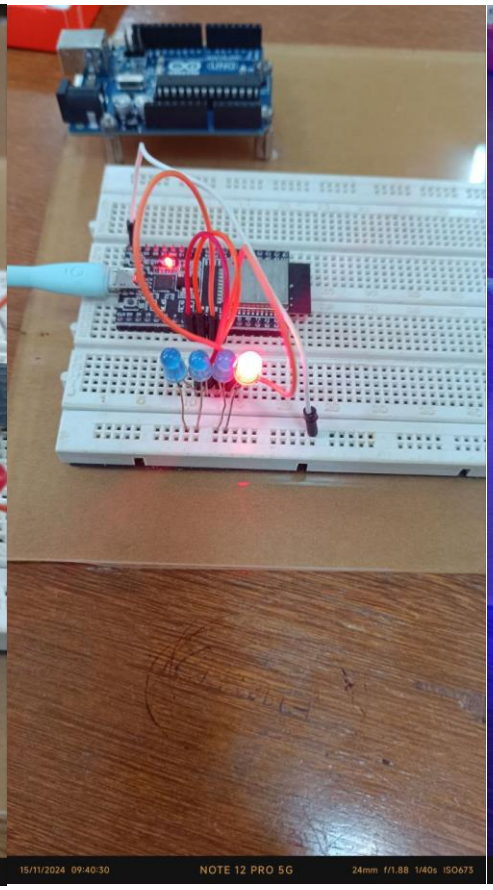


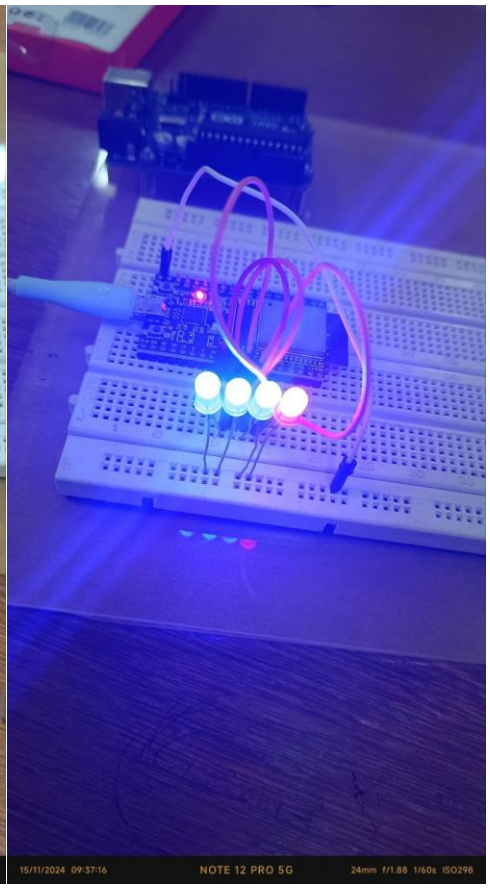
Membuat sebuah token bod dan memberikan Input untuk mengontrol led

LAPORAN PRAKTIK
System iot

Kontrol led







Ini saat led tidak di berikan masukan atau lednya masih mati semua dan di kananya saat led di berikan masukan untuk di nyalakan akan menyalakan led 1 dan led seterusnya akan menyala juga

>>>> Code Program 2

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>

// Wifi network station credentials
#define WIFI_SSID "ssid"
#define WIFI_PASSWORD "passwd"
// Telegram BOT Token (Get from Botfather)
#define BOT_TOKEN "token"

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

void handleNewMessages(int numNewMessages)
{
```

```
  for (int i = 0; i < numNewMessages; i++)
  {
   // Inline buttons with callbacks when pressed will raise a callback_query message
   if (bot.messages[i].type == "callback_query")
   {
    Serial.print("Call back button pressed by: ");
    Serial.println(bot.messages[i].from_id);
    Serial.print("Data on the button: ");
    Serial.println(bot.messages[i].text);
    bot.sendMessage(bot.messages[i].from_id, bot.messages[i].text, "");
   }
   else
   {
    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 == "/options")
    {
     String keyboardJson = "[[{ \"text\" : \"Go to Google\", \"url\" : \"https://www.google.com\" }],[{ \"text\" :
\"Send\", \"callback_data\" : \"This was sent by inline\" }]]";
     bot.sendMessageWithInlineKeyboard(chat_id, "Choose from one of the following options", "",
keyboardJson);
    }

    if (text == "/start")
    {
     String welcome = "Welcome to Universal Arduino Telegram Bot library, " + from_name + ".\n";
     welcome += "This is Inline Keyboard Markup example.\n\n";
     welcome += "/options : returns the inline keyboard\n";

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

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

  // 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();
  }
}
```

>>>> Code Program 3

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>

const unsigned long BOT_MTBS = 1000; // mean time between scan messages

// Wifi network station credentials
#define WIFI_SSID "ssid"
#define WIFI_PASSWORD "paswd"
// Telegram BOT Token (Get from Botfather)
#define BOT_TOKEN "token"

unsigned long bot_lasttime;          // last time messages' scan has been done
WiFiClientSecure secured_client;
UniversalTelegramBot bot(BOT_TOKEN, secured_client);

void handleNewMessages(int 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 (bot.messages[i].longitude != 0 || bot.messages[i].latitude != 0)
   {
    Serial.print("Long: ");
    Serial.println(String(bot.messages[i].longitude, 6));
    Serial.print("Lat: ");
    Serial.println(String(bot.messages[i].latitude, 6));

    String message = "Long: " + String(bot.messages[i].longitude, 6) + "\n";
    message += "Lat: " + String(bot.messages[i].latitude, 6) + "\n";
    bot.sendMessage(chat_id, message, "Markdown");
   }
   else if (text == "/start")
   {
    String welcome = "Welcome to Universal Arduino Telegram Bot library, " + from_name + ".\n";
    welcome += "Share a location or a live location and the bot will respond with the co-ords\n";

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

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

 // 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();
  }
}
```

>>>> Code Program 4

```
/*******************************************************************
    A telegram bot for your ESP32 that demonstrates sending an image
    from URL.

    Parts:
    ESP32 D1 Mini stlye Dev board* - http://s.click.aliexpress.com/e/C6ds4my
    (or any ESP32 board)

      = Affilate

    If you find what I do useful and would like to support me,
    please consider becoming a sponsor on Github
    https://github.com/sponsors/witnessmenow/

    Example originally written by Vadim Sinitski

    Library written by Brian Lough
    YouTube: https://www.youtube.com/brianlough
    Tindie: https://www.tindie.com/stores/brianlough/
    Twitter: https://twitter.com/witnessmenow
 *******************************************************************/
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>

// Wifi network station credentials
#define WIFI_SSID "harjolukito"
#define WIFI_PASSWORD "ami1971da"
// Telegram BOT Token (Get from Botfather)
```

```
#define BOT_TOKEN "6392538293:AAEwF1QytFs2JKM1fqoWZWIwI0spCVtnDjU"

const unsigned long BOT_MTBS = 1000; // mean time between scan messages

unsigned long bot_lasttime;          // last time messages' scan has been done
WiFiClientSecure secured_client;
UniversalTelegramBot bot(BOT_TOKEN, secured_client);

String test_photo_url = "https://www.arduino.cc/en/uploads/Trademark/ArduinoCommunityLogo.png";

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 == "/get_test_photo") {
      bot.sendPhoto(chat_id, test_photo_url, "Caption is optional, you may not use photo caption");
    }

    if (text == "/start") {
      String welcome = "Welcome to Universal Arduino Telegram Bot library, " + from_name + ".\n";
      welcome += "This is Send Image From URL example.\n\n";
      welcome += "/get_test_photo : getting test photo\n";

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

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

  // 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();
  }
}
```