

SENDING A DATA FROM TELEGRAM-BOT TO ADAFRUIT.IO FOR CONTROLLING LIGHT

SWASTHIKA.B¹, AFROSE BANU.R², HARIVIGNESH.R³, HARIMUKESH.R⁴

^{1,2,3}Student, Department of Electronics and Instrumentation Engineering, Kumaraguru College of Technology, Coimbatore, Tamil Nadu, India.

⁴Student, Department of Information Technology, SASTRA Deemed to be University, Thanjavur, Tamil Nadu, India

Abstract -With the huge progress in the field of information and communication technology, there is a huge increases in the use of electronic products have observed especially in computers and mobile devices are piling up in our homes and offices. With constant up-gradation of electronic devices, users are discarding the obsolete product, which is leading to huge waste in electronic devices and electricity. The electricity waste is considered as one of the important origins of our day to day life. This paper presents a design of Telegram Bot to send the data by controlling light. Such bots serves the information via specific commands.

Key Words: Internet of Things (IoT), Google Colaboratory, Telegram, Adafruit.IO, Heroku

SOFTWARE REQUIREMENTS:

- Google Colaboratory
- Adafruit.io
- Telegram app
- GitHub
- Heroku.

1. INTRODUCTION:

"Technology is important because it creates the future. We're able to be a part of the "next" and create things that don't exist". Smart home, robots, drones, artificial intelligence.. isn't everything sounds fictional? Well, now this our new common, new routine.

Here is a Article about sending a data from telegram bot to Adafruit.IO which can turn on and off the light from anywhere in the world. With a single command /On, the light turns on and with a single command /Off, the light turns off and returns the image of bulb on/off. It's human tendency to forget things. We generally forget to switch off the lights when we are leaving from the room. Electrical energy is precious for our day-to-day life. But, it will be a huge mistake that we are wasting it. **"Enthusiasm is the electricity of life. How do you get it? You act enthusiastic until you make it a habit"- Gordon Parks.** Using the concepts of IOT and

robotics, we can turn on and off the light in a smarter way. Who doesn't love smart things? Smart phones, Smart home, Smart vehicle and so on. We crave for smart technology.

The rise of commission free trading APIs along with cloud computing has made it possible for the average person to run their own algorithmic trading strategies. All you need is a little python and more than a little luck.

1.1 Internet of things(IoT):

Internet of Things (IoT) is a modern concept and technology which uses the electronic equipment which are connected all together through wired or wireless network in a specific area or a town without the user interacting the electronic device. Internet of things (IOT) refers to connect things to Internet, things are an entity that has intelligence to sense data around its environment, and send it to the Internet, where it processes the data and make it useful for engineering purpose. The devices connected together inside Internet of Things (IoT) can be categorized into three main group consumer, enterprise and industrial. Smart TV, wearable and smart devices are connected devices in consumer device. Smart traffic light system, smart city camera – used to monitor traffic and weather conditions are classified under industrial and enterprise devices. The present work experimentally investigates on sending a data from telegram bot to adafruit.io through a wireless network. With Internet of things, Deploy it on Heroku by connecting GitHub. Heroku, it's a cloud platform and the Heroku network runs the customer's apps in virtual containers which execute on a reliable runtime environment.

1.2 API

The term API is an acronym, and it stands for "Application Programming Interface." APIs have been an important part of the computer industry since the early days. API, it is a software intermediary that allows two applications to talk to each other. An API isn't the same as the remote server — rather **it is the part of the server that receives requests and sends responses.** Both the telegram and adafruit.io has a API keys which are unique for the users.

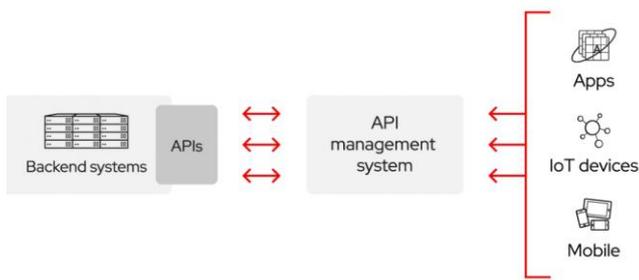


Fig 2: API-Block Diagram

2. SOFTWARE EXPLANATION:

2.1 ADAFRUIT-IO:

It's a cloud service. Display your data in real-time, online. Adafruit.io can handle and visualize multiple feeds of data. Feeds are the core of the Adafruit IO system. This includes settings for whether the data is public or private, what license the stored sensor data falls under, and a general description of the data. The feed also contains the sensor data values that get pushed to Adafruit IO from your device. Dashboards are a feature integrated into Adafruit IO which allow you to chart, graph, gauge, log, status indicator and display your data.

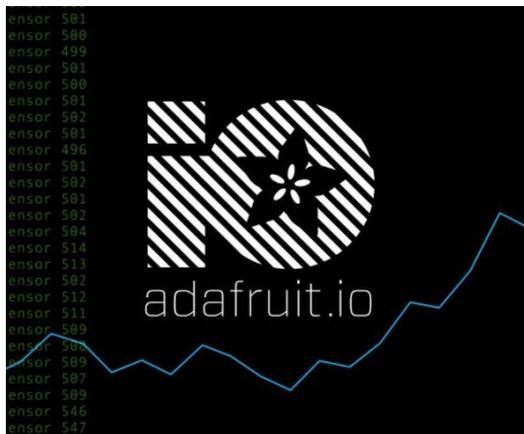


Fig 1: Adafruit.IO Logo

2.2 BOT:

BOT, A Name derived from ROBOT. Bots are everywhere. It seems that only yesterday we did not even know about their existence; now we can barely imagine our life without them. It is a software application that is programmed to do certain tasks and they do repetitive tasks too, and much faster than human users could. Bots are automated, which means they run according to their instructions without a human user needing them to start up. It interacts with existing/potential customers via chat apps. The bot can be used with Messenger, Telegram and other supported chat apps making it easily accessible and globally available. Bots usually operate over a network; more

than half of Internet traffic is bots scanning content, interacting with webpages, chatting with users, or looking for attack targets.

2.2.1 TELEGRAM-BOT:

Bots started to emerge after Telegram announced a new Telegram bot API in 2015. It allows third parties to create bots that will use this messaging application as their main interface. The Telegram bots list is long: many developers have experimented with the provided API to generate unique bots for various purposes.

In short, Telegram bots are applications set to perform specific functions, follow instructions, and interact with users. Bots run inside the Telegram environment and do not require an additional installation procedure. By using bots, people can enhance the functionality of Telegram. For instance, you can send daily notifications to your friends, set reminders for yourself, translate messages, receive weather alerts, play games, and so much more.



Fig 3: Telegram-Bot Father

2.3 HEROKU:

Heroku is a cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. It takes away all the pain of installing software, maintaining it, monitoring the software for required updates, setting up your deployments, etc. Sounds like magic? Feels a bit like it, this platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market.

2.3.1 HOW DOES FREE HEROKU WORKS:

- i) 1 account = N apps (sites)
- ii) Each app goes sleep (idle) after 30 mins if no usage
- iii) 1 account has 1000 dyno hours/mon
- iv) 1 dyno hour = 1 hour site is online. Just anyone who will open your site makes it run again
- v) If you open site for 1 sec with 30 mins interval 10 times you will use 5 dyno hours

vi) To not make site sleep you can use site monitor like New Relic. It will check site availability by pinging it and will make it online each 15 mins (before Heroku will make it sleep).

vii) No visitors limits

So I'm made one more free Heroku account and create only one site there + setup a monitor to avoid sleeping. In this case it will be online all the time and will use 744 dyno hours from 1000 available per month maximum.



Fig 4: Heroku Logo

3. STEPS TO BE FOLLOWED:

3.1 Before starting, You will have to sign up for the following services:

- An adafruit.io account
- A telegram account(on which we will build our bot)
- Heroku (cloud platform).

3.2 Create a feed in adafruit.io, Visit the adafruit.io account. Create a New dashboard and Feed. Click on the new dashboard there you create a Indicator. Connect the indicator with your feed .status indicator is created. Click the adafruit.io key on the top corner. Copy the user name and IO key to the google colaboratory.

3.3 Create a telegram bot, Visit the Bot father in telegram to create a new Telegram bot,

Send a command **/new bot** in Bot Father .Give the Telegram bot a friendly name .Choose the username for your bot, it must be end in 'bot'. Done! On your new bot. Bot Father will provide us with a unique token to access **HTTP API**. Enter the Telegram bot's access token to the google colaboratory.

3.4 Coding the bot using python,

The bot is coded with Python. Apart from this, a Procfile and a requirements.txt file should be created on GitHub. The Procfile is required to create a dyno for your app on Heroku. The requirements file is required for Heroku to install the

necessary libraries to run Python and Adafruit. Snapchat of Python code can be seen below:

```

From Adafruit_IO import Client,Data
import os
def turnoffthelight(update, context):
    context.bot.send_message(chat_id=update.effective_chat.id, text="Bulb turned off")
    context.bot.send_photo(chat_id=update.effective_chat.id, photo="https://pingimg.com/uploads/bulb/bulb_PWG1241.png")
    send_value(0)
def turnonthelight(update, context):
    context.bot.send_message(chat_id=update.effective_chat.id, text="Bulb turned on")
    context.bot.send_photo(chat_id=update.effective_chat.id, photo="https://img.icons8.com/plasticine/2x/light-on.png")
    send_value(1)
def send_value(value):
    feed = aio_feeds('bot001')
    aio_send_data(feed.key, value)
def input_message(update, context):
    text=update.message.text
    if text == "turnonthelight":
        send_value(1)
        context.bot.send_message(chat_id=update.effective_chat.id, text="Bulb turned on")
        context.bot.send_photo(chat_id=update.effective_chat.id, photo="https://img.icons8.com/plasticine/2x/light-on.png")
    elif text == "turnoffthelight":
        send_value(0)
        context.bot.send_message(chat_id=update.effective_chat.id, text="Bulb turned off")
        context.bot.send_photo(chat_id=update.effective_chat.id, photo="https://pingimg.com/uploads/bulb/bulb_PWG1241.png")
def start(update, context):
    start_message=""
    /turnoff the light or 'turn off':to turn off the bulb ,sends value=0 in feed
    /turnon the light or 'turn on' :to turn on the bulb ,sends value=1 in feed
    ...
    context.bot.send_message(chat_id=update.effective_chat.id, text=start_message)
ADAFRUIT_IO_USERNAME = os.getenv('SARJAI171')
ADAFRUIT_IO_KEY = os.getenv('')
TOKEN = os.getenv('')
bot = Client(ADAFRUIT_IO_USERNAME,ADAFRUIT_IO_KEY)
updater=Updater(TOKEN,use_context=True)
dispatcher = updater.dispatcher
dispatcher.add_handler(CommandHandler('turnoffthelight',turnoffthelight))
dispatcher.add_handler(CommandHandler('turnonthelight',turnonthelight))
dispatcher.add_handler(CommandHandler('start',start))
dispatcher.add_handler(MessageHandler(filters.text & (~filters.command),input_message))
updater.start_polling()
updater.idle()
    
```

Fig 5: Snapshot of python code

3.5 Deploying the app on Heroku,

Create a new account or use an existing one to build an app in Heroku. Name your app suitably and click next. And connect GitHub account to Heroku as the 'Deployment Method' instead of using Heroku CLI. Search your repository and connect it. Also, enable 'Automatic Deploys'.

Go to the 'Settings' page and set your 'Config Vars' to values containing your Adafruit username, password, and Telegram API token. Create a 'Procfile' on your connected GitHub repository and refresh the 'Overview' tab page on Heroku. On doing this, you will see that it has recognized your 'Procfile' under 'Dyno formation'. Click on 'Configure Dynos' and set it to ON. With these steps, your app should start building and get deployed successfully!

4. OUTPUT:

Commands

- Command **/start** will return a message instructing the user to give the appropriate inputs.



Fig 6: Snapshot of /start command

- /turnontheight will display the message “Bulb turned ON”. It will also return an image of a glowing bulb and will send a value to Adafruit indicating that its ON.



Fig 7: Snapshot of /turnontheight command

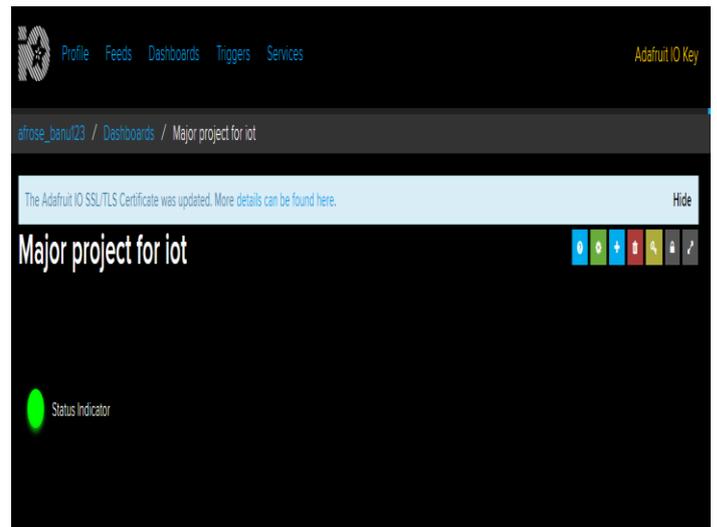


Fig 8: Snapshot of Adafruit Status Indicator as ON condition

- /turnofftheight will display the message “Bulb turned OFF”. It will also return an image of an without glowing bulb and will send a value to Adafruit indicating that its OFF.



Fig 9: Snapshot of /turnofftheight command

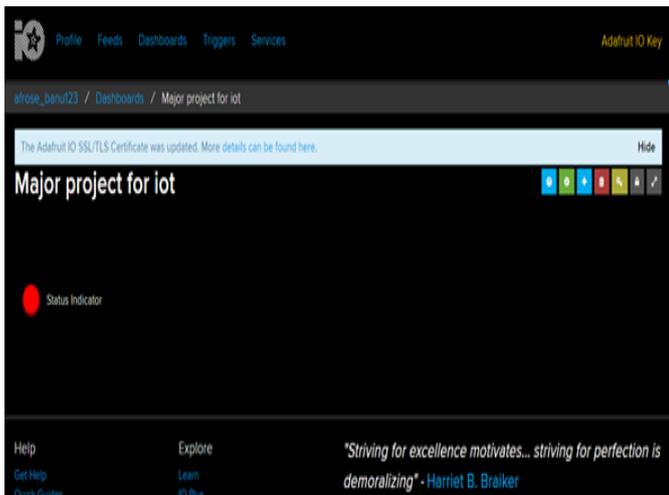


Fig 10: Snapshot of adafruit.io Status Indicator as OFF condition

5. CONCLUSIONS

This project aims at designing and executing the advanced development for saving energy. Nowadays, human has become too busy and unable to find time even to switch the light wherever not necessary. This can be seen more effectively in all houses. The present system is like, the light can be controlled by social medium of communication, not only in Telegram, we can now control light using text messages via social media applications such as Instagram, Twitter, WhatsApp, Facebook, etc.. This project gives the best solution for electrical power wastage. We believe that our idea provides better than the existing system.

REFERENCES

- [1] <https://io.adafruit.com/api/docs/#adafruit-io-http-api>
- [2] <https://core.telegram.org/bots/api>
- [3] <https://devcenter.heroku.com/categories/reference>
- [4] <https://github.com/afrose-banu/Telegrambot-with-adafruit>
- [5] <https://www.python.org/doc/>
- [6] <https://docs.github.com/en>