

ROS2 **Line** Notify By TESR

IoT(Internet of things)

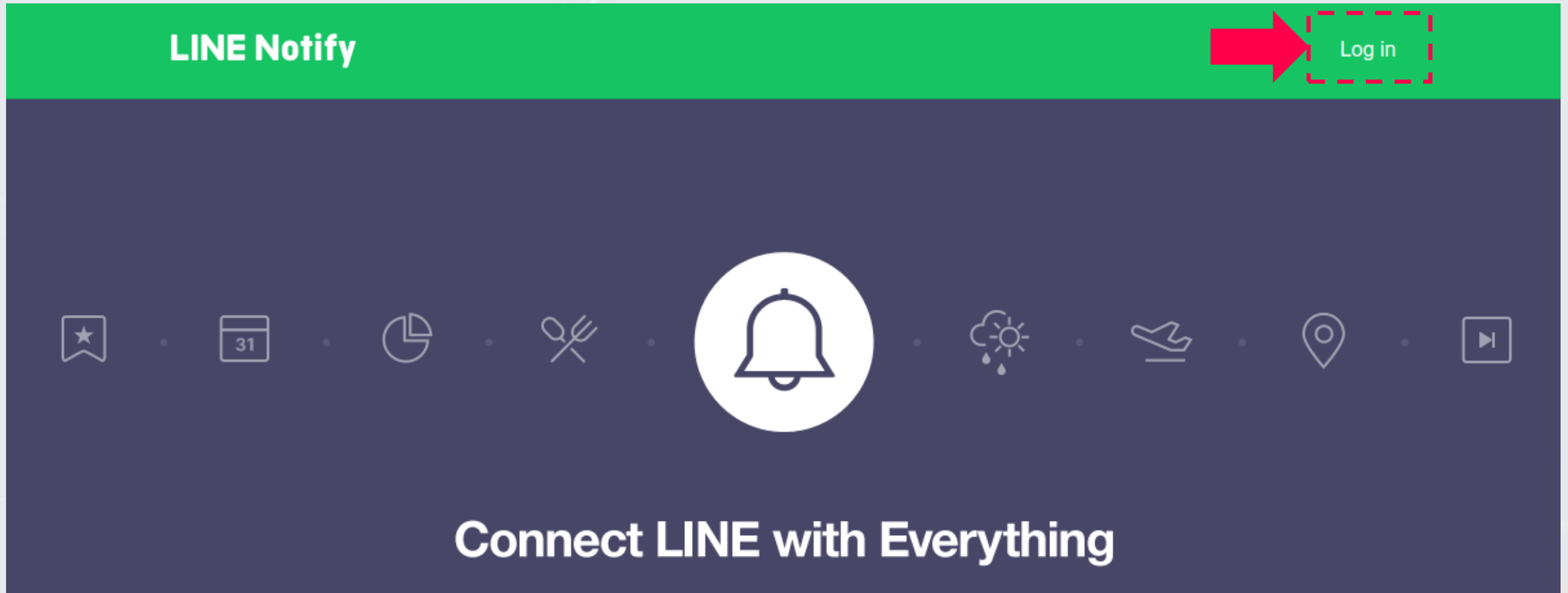
- **Internet of things** – the description of technologies that have the ability to connect and exchange the data through the internet.



Line Notify

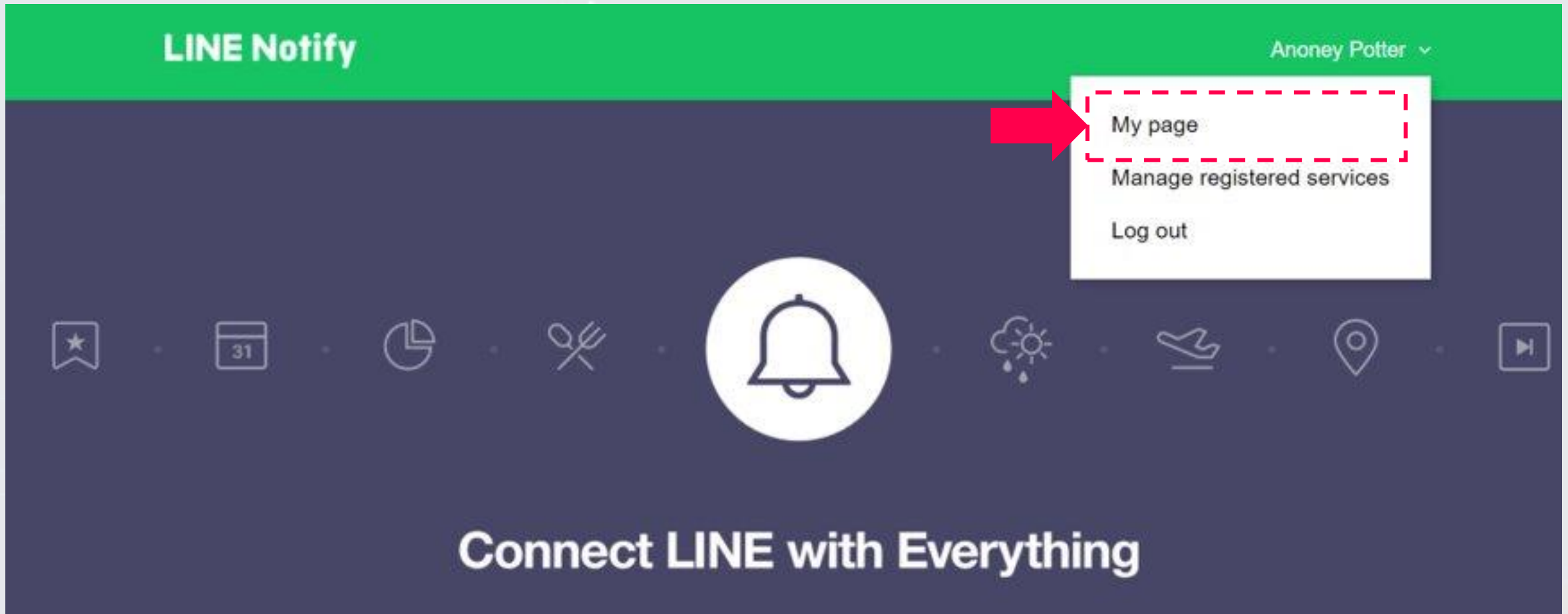


Line Notify



<https://notify-bot.line.me/en/>

Line Notify



Line Notify

Generate access token (For developers)

By using personal access tokens, you can configure notifications without having to add a web service.

Generate token

LINE Notify API Document

Generate token

Please enter a token name to be displayed before each notification.

TESR Notify

Select a chat to send notifications to.

Search by group name

1-on-1 chat with LINE Notify

5Here

AA-IONIC App

Alphabot 1 day

Arduino AlphaBot

Note: Revealing your personal access token can allow a third party to obtain the names of your connected chats as well as your profile name.

Generate token

Line Notify

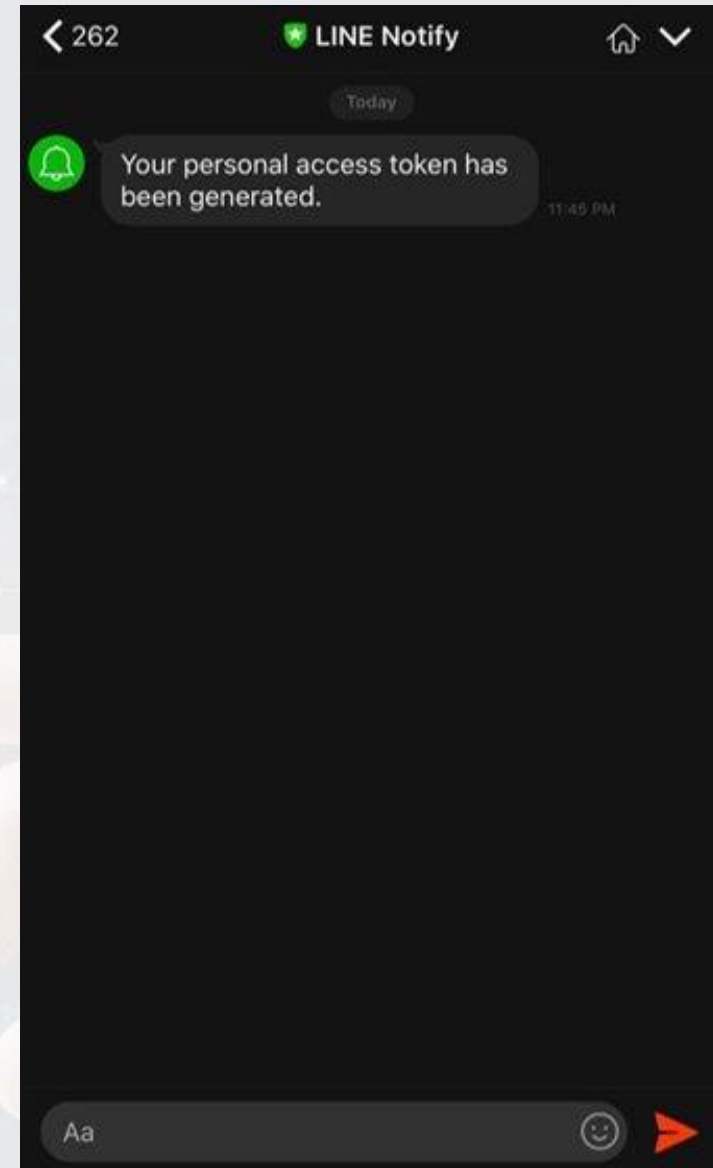
Your token is:

BnnKeef4civQ0yCdkQj06SppDEwPPkOZncONX

If you leave this page, you will not be able to view your newly generated token again. Please copy the token before leaving this page.

Copy

Close



Line Notify: For message

```
#pip3 install requests
import requests

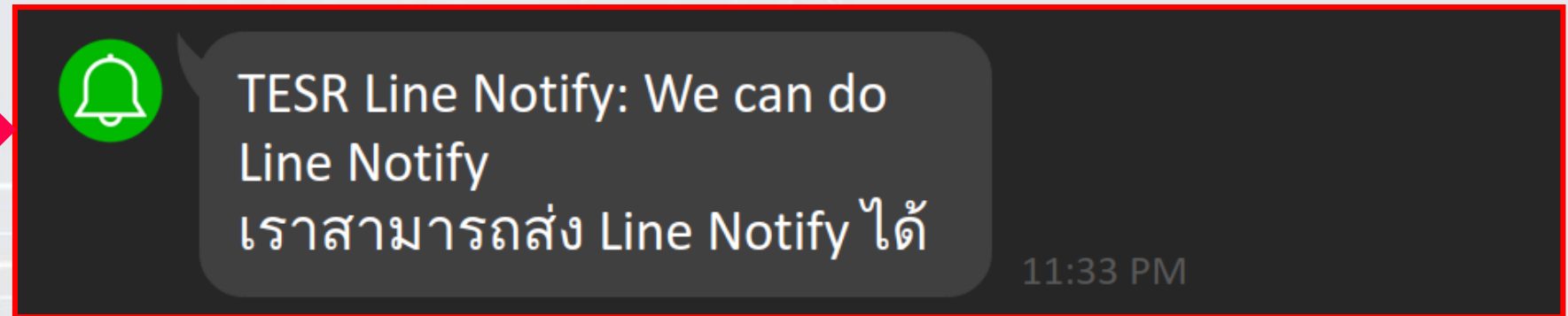
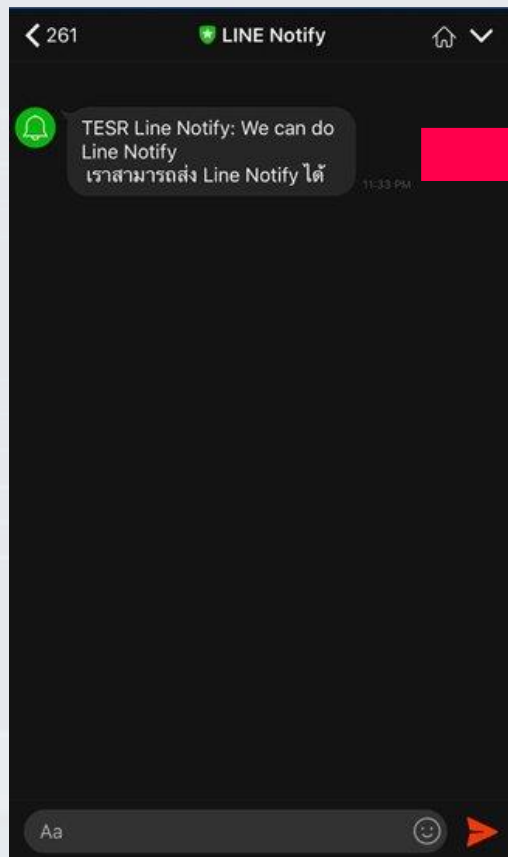
url = "https://notify-api.line.me/api/notify"
token = "WU14vzz83QiGQj04aBS9bC2wmnVGVULDpzL29skobHk" #your Line Notify token
headers = {'Authorization': 'Bearer ' + token}

msg = {
    "message": "We can do Line Notify \n เราสามารถส่ง Line Notify ได้"
}

res = requests.post(url, headers=headers, data = msg)
print(res.text)
```


Line Notify: For message

• Result



In code

```
msg = {  
    "message": "We can do Line Notify \n เราสามารถส่ง Line Notify ได้"  
}
```

Line Notify: For sticker

```
#pip3 install requests
import requests

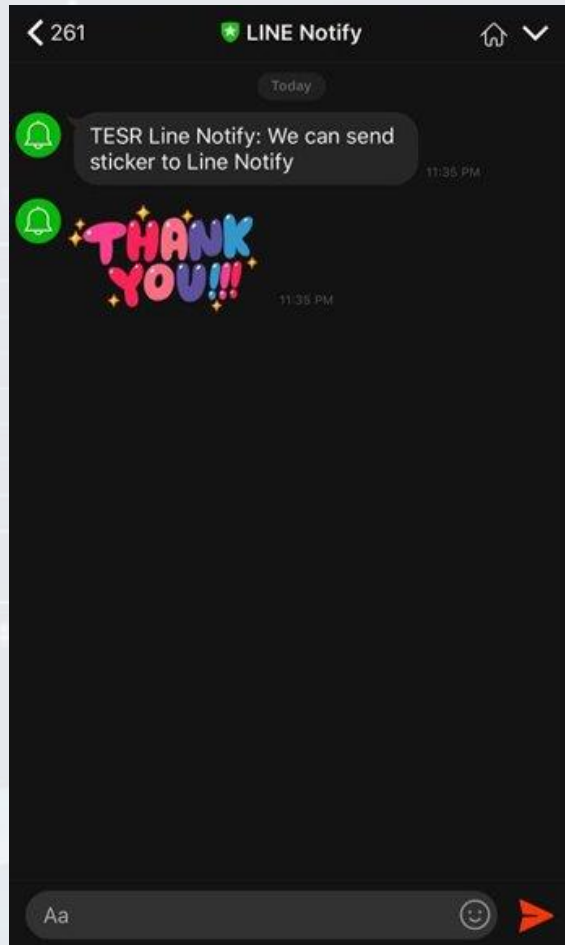
url = "https://notify-api.line.me/api/notify"
token = "WU14vzz83QiGQj04aBS9bC2wmnVGVULDpzL29skobHk" #your Line Notify token
headers = {'Authorization': 'Bearer ' + token}

msg = {
    "message": "We can send sticker to Line Notify",
    "stickerPackageId": "2", "stickerId": "41"
}

res = requests.post(url, headers=headers, data = msg)
print(res.text)
```

Line Notify: For sticker

- Result



Sticker ID list [Link](#)

STKPKGID : 1 (Moon James)			
STKID	STKPKGID	STKVER	IMG_sample
1	1	100	
2	1	100	
3	1	100	
4	1	100	
5	1	100	
6	1	100	
7	1	100	
8	1	100	
9	1	100	
10	1	100	

STKID	STKPKGID	STKVER	IMG_sample
11	1	100	
12	1	100	
13	1	100	
14	1	100	
15	1	100	
16	1	100	
17	1	100	
21	1	100	
100	1	100	
101	1	100	

STKID	STKPKGID	STKVER	IMG_sample
102	1	100	
103	1	100	
104	1	100	
105	1	100	
106	1	100	
107	1	100	
108	1	100	
109	1	100	
110	1	100	
111	1	100	

Line Notify: For URL Photo

```
#pip3 install requests
import requests

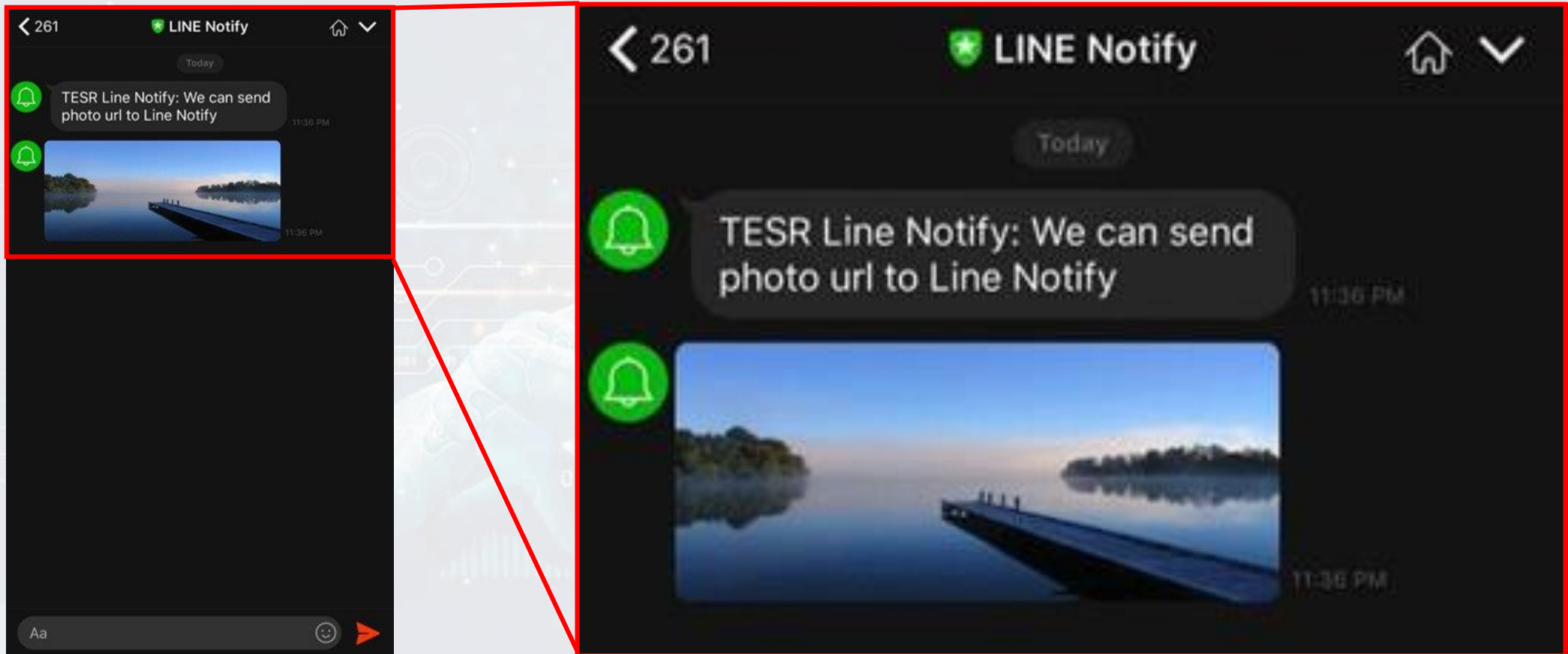
url = "https://notify-api.line.me/api/notify"
token = "WU14vzz83QiGQj04aBS9bC2wmnVGVULDpzL29skobHk" #your Line Notify token
headers = {'Authorization': 'Bearer ' + token}

msg = {
    "message": "We can send photo url to Line Notify",
    "imageThumbnail": "https://jpeg.org/images/jpeg-home.jpg",
    "imageFullsize": "https://jpeg.org/images/jpeg-home.jpg"
}

res = requests.post(url, headers=headers, data = msg)
print(res.text)
```

Line Notify: For URL Photo

- Result



Line Notify: For Photo

```
#pip3 install requests
import requests

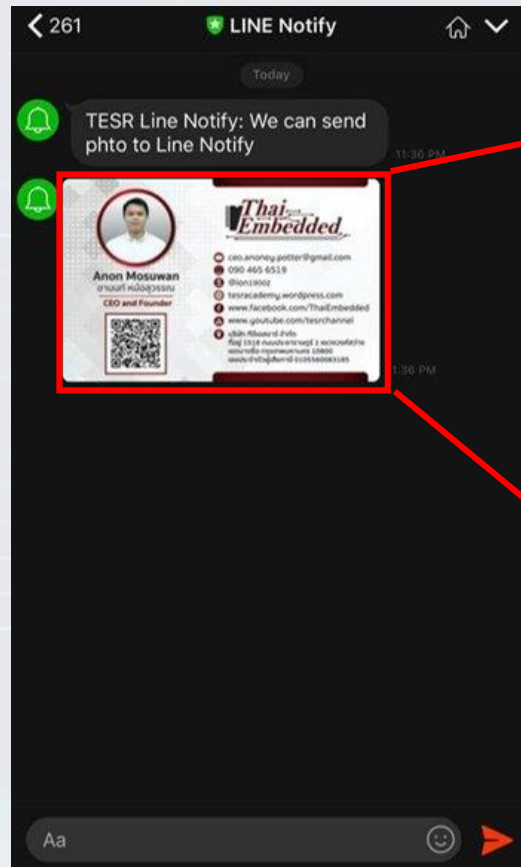
url = "https://notify-api.line.me/api/notify"
token = "WU14vzz83QiGQj04aBS9bC2wmnVGVULDpzL29skobHk" #your Line Notify token
headers = {'Authorization': 'Bearer ' + token}

msg = {
    "message": (None, "We can send photo to Line Notify"),
    "imageFile": open("/home/pi/LineNotify/NameCard.png", "r+b")
}

res = requests.post(url, headers=headers, files=msg)
print(res.text)
```

Line Notify: For Photo

• Result





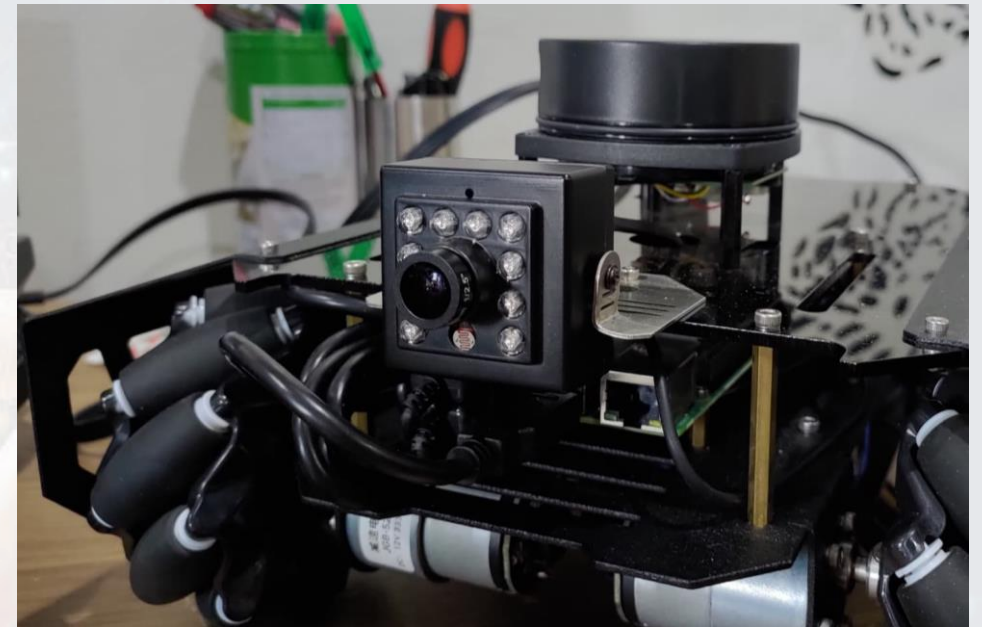
Anon Mosuwan
อานนท์ หม้อสุวรรณ
CEO and Founder





ceo.anoney.potter@gmail.com
090-465-6519
@ion1900z
www.tesrshop.com
www.facebook.com/ThaiEmbedded
www.youtube.com/user/tesrchannel
บริษัท ทีโอเอสอาร์ จำกัด
ที่อยู่ 112/296 หมู่บ้าน เพอร์เฟค มาสเตอร์พีซ หมู่ที่ 2
ตำบลไทรมา อำเภอมืองนนทบุรี จังหวัดนนทบุรี 11000
เลขประจำตัวผู้เสียภาษี 0105560083185

2D Camera on **iron-X**



Line Notify: For 2D Camera

```
#pip3 install requests
import requests
from cv2 as cv
from time import sleep

vid = cv.VideoCapture(0)
ret,frame = vid.read()
img = cv.imwrite("/home/pi/Desktop/image.jpg",frame)
sleep(1.0)

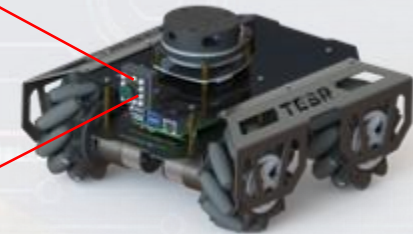
url = "https://notify-api.line.me/api/notify"
token = "WU14vzz83QiGQj04aBS9bC2wmnVGVULDpzL29skobHk" #your Line Notify token
headers = {'Authorization':'Bearer ' + token}

msg = {
    "message":(None,"We can send photo to Line Notify"),
    'imageFile':open("/home/pi/Desktop/image.jpg","r+b")
}

res = requests.post(url, headers=headers ,files=msg)
print(res.text)
```

Line Notify: For 2D Camera

- Result





- 2D Camera will capture the image. it will save as image file and then send to the LINE Notify.

Line Notify: Message from ROS2

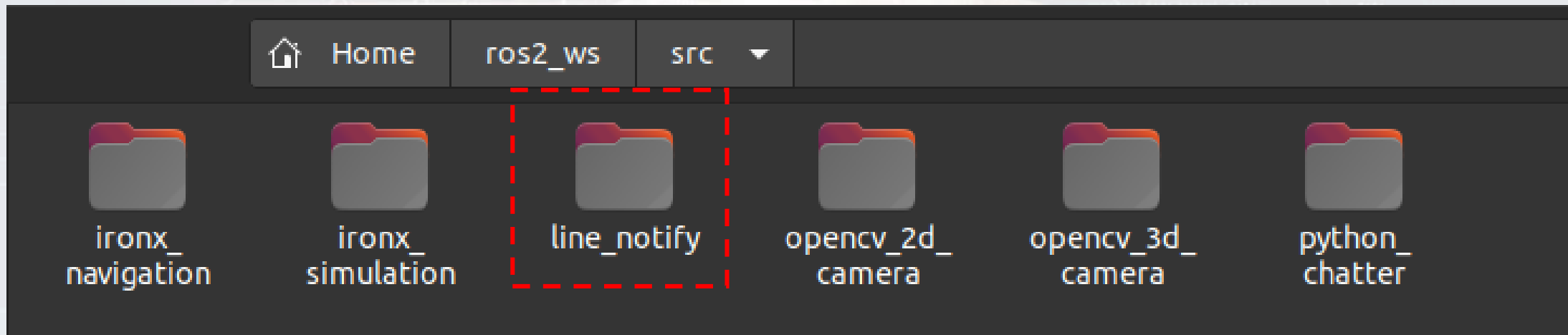


Line Notify: Message from ROS2

- Download ros2 workspace source code: [Link](#)

Name ↓	Owner	Last modified	File size
 line_notify.zip 	me	3:06 PM me	9 KB

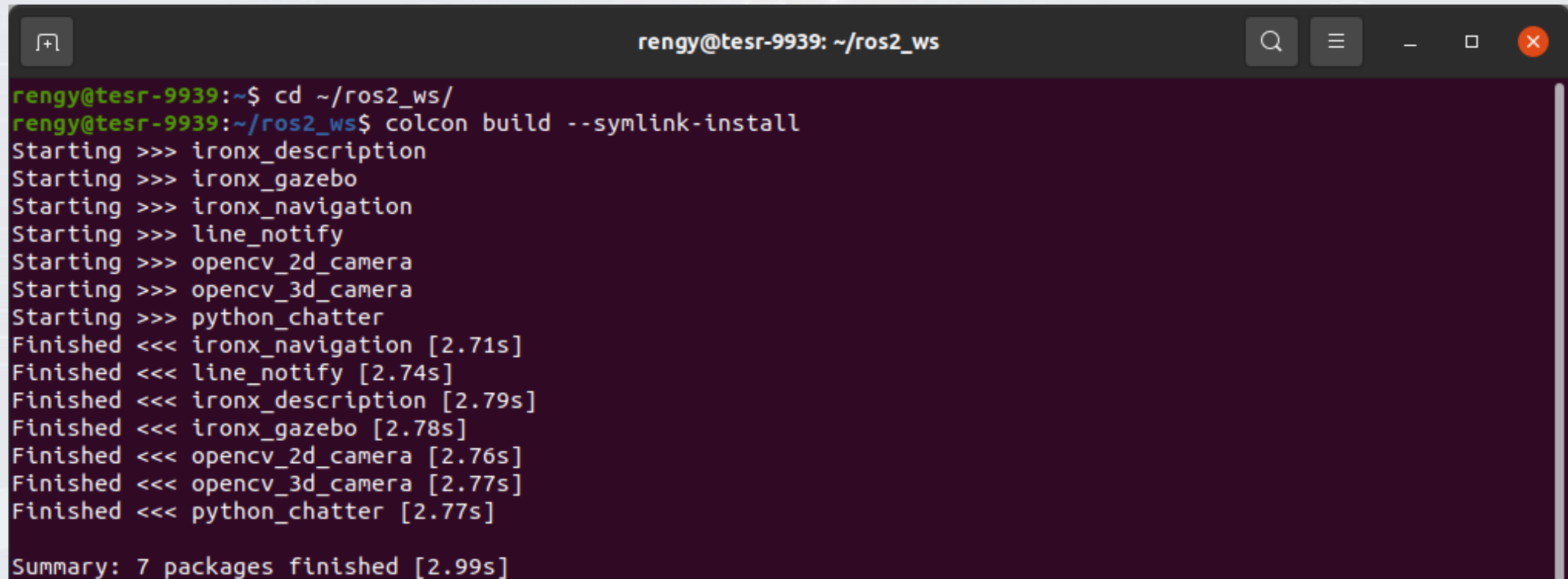
- Extract and move file to ros2 workspace.
(For this case, ros2 workspace is ~/ros2_ws/src)



Line Notify: Message from ROS2

- Change the directory to ros2_ws and build the workspace by using:

```
cd ~/ros2_ws  
colcon build --symlink-install
```

A terminal window titled 'rengy@tesr-9939: ~/ros2_ws' showing the execution of 'colcon build --symlink-install'. The terminal output lists the starting and finishing times for seven packages: ironx_description, ironx_gazebo, ironx_navigation, line_notify, opencv_2d_camera, opencv_3d_camera, and python_chatter. The build process is summarized as '7 packages finished [2.99s]'.

```
rengy@tesr-9939: ~/ros2_ws  
rengy@tesr-9939:~$ cd ~/ros2_ws/  
rengy@tesr-9939:~/ros2_ws$ colcon build --symlink-install  
Starting >>> ironx_description  
Starting >>> ironx_gazebo  
Starting >>> ironx_navigation  
Starting >>> line_notify  
Starting >>> opencv_2d_camera  
Starting >>> opencv_3d_camera  
Starting >>> python_chatter  
Finished <<< ironx_navigation [2.71s]  
Finished <<< line_notify [2.74s]  
Finished <<< ironx_description [2.79s]  
Finished <<< ironx_gazebo [2.78s]  
Finished <<< opencv_2d_camera [2.76s]  
Finished <<< opencv_3d_camera [2.77s]  
Finished <<< python_chatter [2.77s]  
  
Summary: 7 packages finished [2.99s]
```

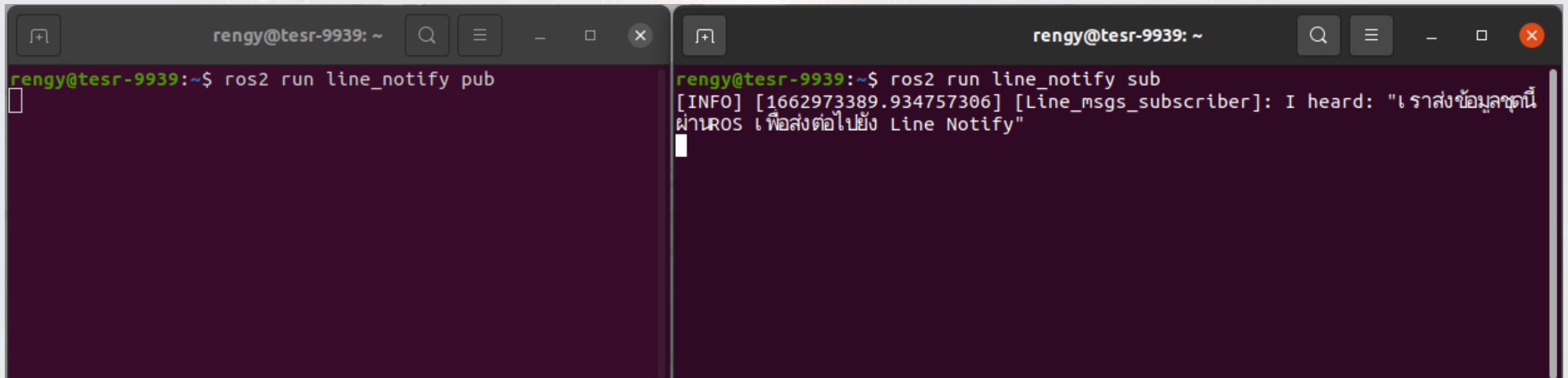
Line Notify: Message from ROS2

- And then, we start with publish the message using:

```
ros2 run line_notify pub
```

- And run subscribe the data from pub on new terminal using:

```
ros2 run line_notify sub
```

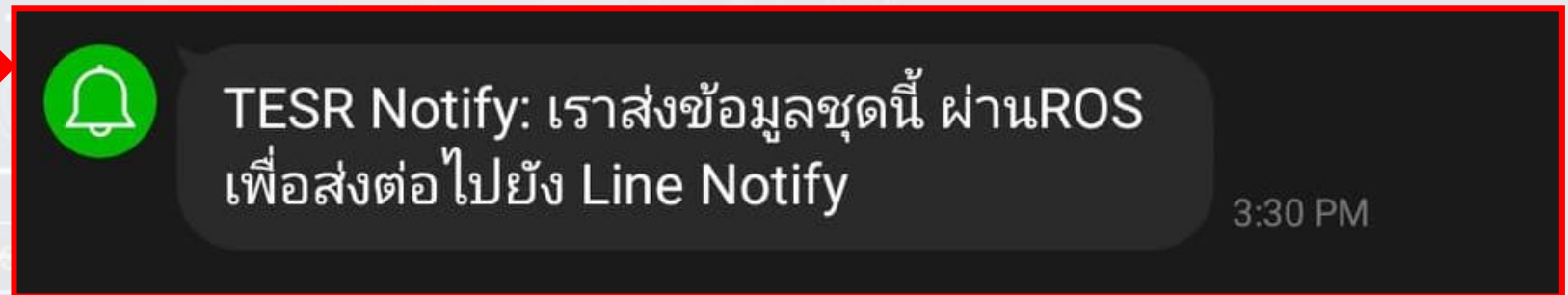
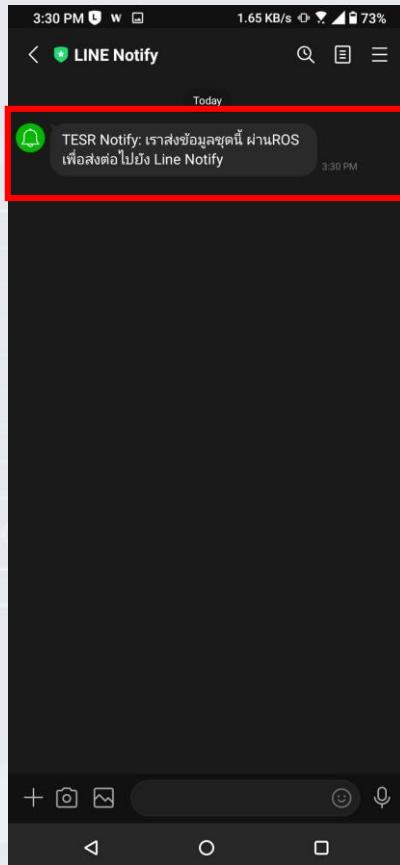


```
rengy@tesr-9939: ~$ ros2 run line_notify pub
```

```
rengy@tesr-9939: ~$ ros2 run line_notify sub
[INFO] [1662973389.934757306] [Line_msgs_subscriber]: I heard: "เราส่งข้อมูลนี้ผ่านROS เพื่อส่งต่อไปยัง Line Notify"
```

Line Notify: Message from ROS2

- Result



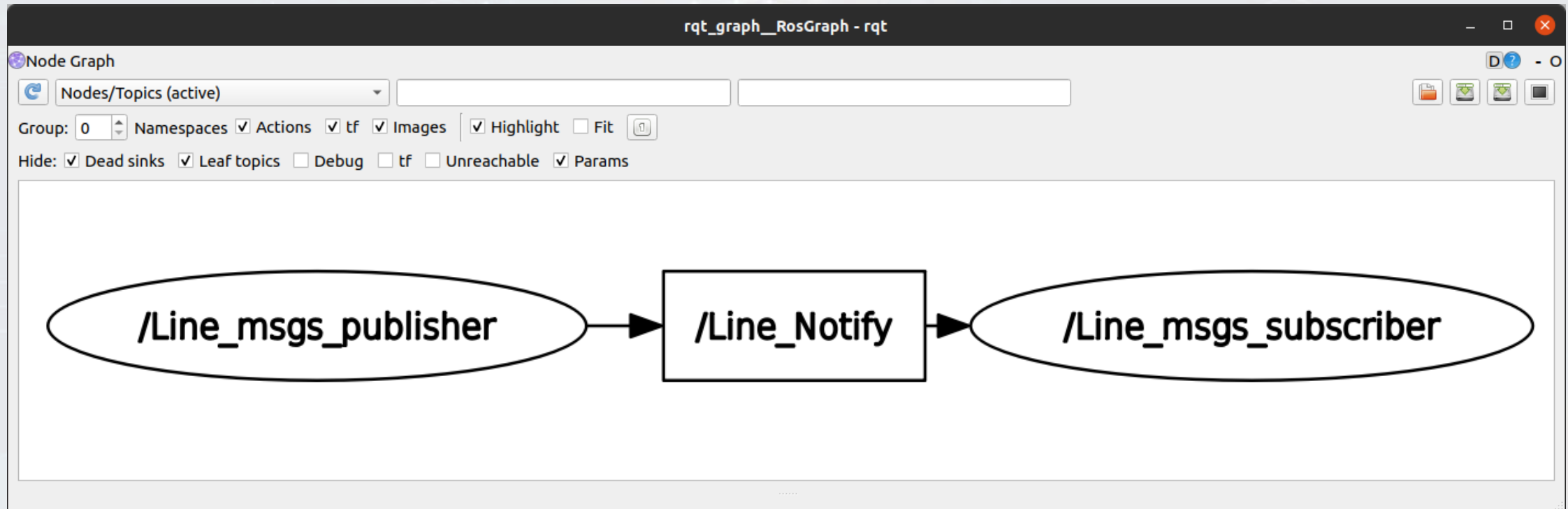
In code

```
def timer_callback(self):  
    msg = String()  
    msg.data = "เราส่งข้อมูลชุดนี้ ผ่านROS เพื่อส่งต่อไปยัง Line Notify"  
    self.publishers_.publish(msg)
```

RosGraph for Line Notify

- You can the **RosGraph** of Line Notify by using:

rqt_graph



Contact Us

Email: tesrshop@gmail.com

Line official Account: @ion1900z

Facebook fanpage: TESR

Tel. 082-983-7768

Scan here



TESR Co., LTD

112/296 หมู่บ้าน เพอร์เฟค มาสเตอร์พีช
หมู่ที่ 2 ตำบลไทรมา อำเภอมะนังนบุรี
จังหวัดนบุรี 11000