MiniProject 4 姓名:郭紘安 學號:109062578

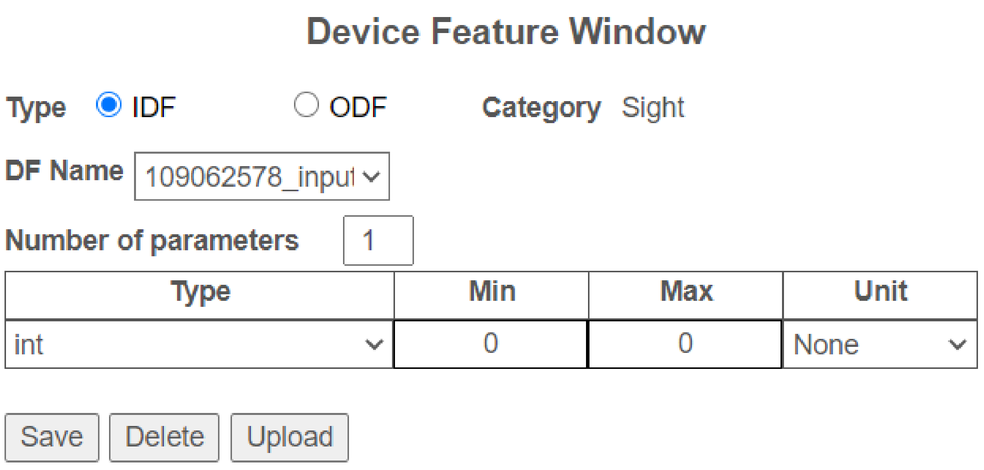
1. Scenario Description

使用 IoTtalk 建立一個自己的 model 後,再使用 LineBot 傳資料 Push 到 IoTtalk 的 IDF,經過一個 Join 計算後將結果傳給 ODF,ODF 有結果之後再使用 LineBot 通知去 Pull 結果出來並透過 LineBot 傳給使用者。

2. How to Design

Step 1:建立兩個 IDF 的 Device Features

109062578_input_1:



Device Feature Window

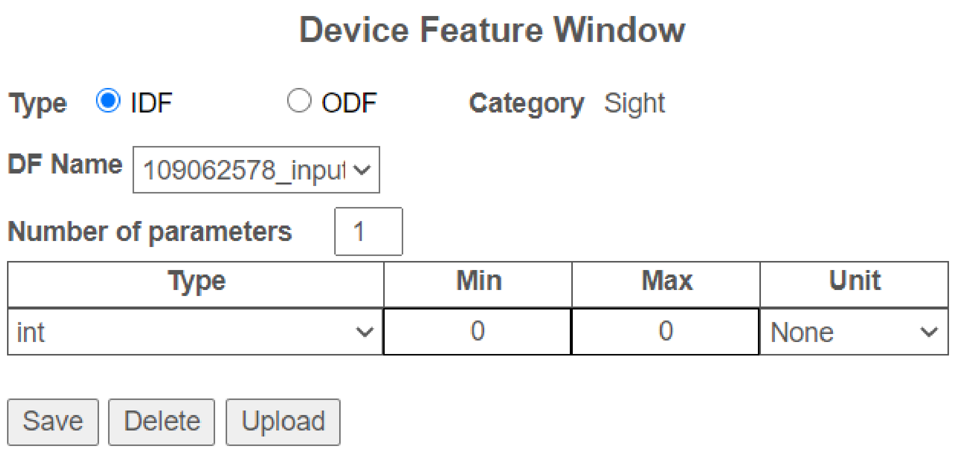
Type [✓ IDF] ODF Category Sight

DF Name 109062578_input

Number of parameters 1

Type	Min	Max	Unit
int	0	0	None

109062578_input_2:



Device Feature Window

Type [✓ IDF] ODF Category Sight

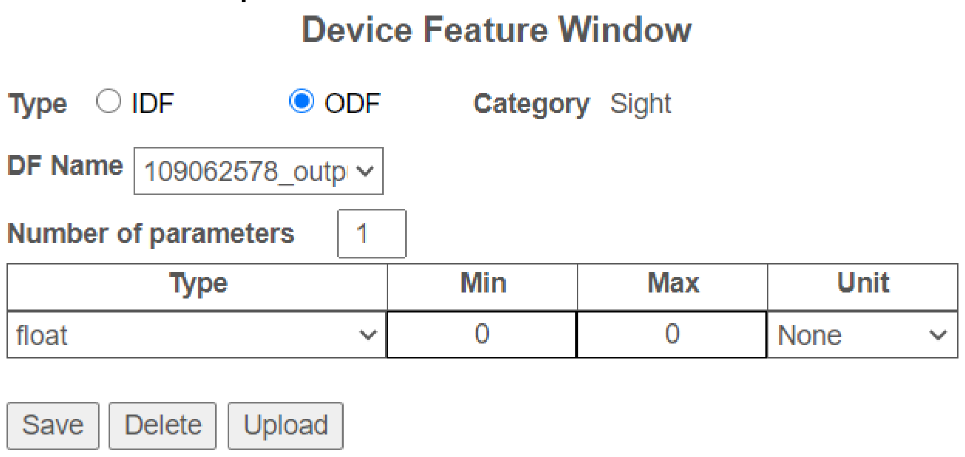
DF Name 109062578_input

Number of parameters 1

Type	Min	Max	Unit
int	0	0	None

Step 2:建立一個 ODF 的 Device Feature

109062578_output:



Device Feature Window

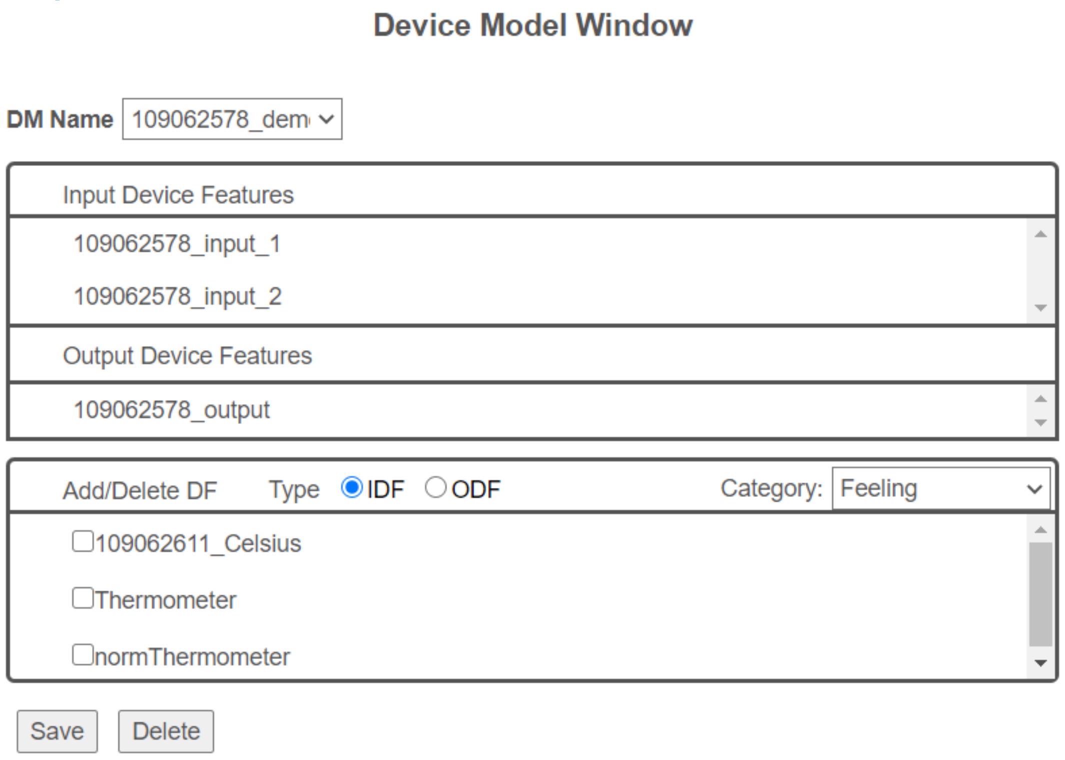
Type ○ IDF ● ODF Category Sight

DF Name 109062578_outp

Number of parameters 1

Type	Min	Max	Unit
float	0	0	None

Step 3:Device Model



Device Model Window

DM Name 109062578_dem

Input Device Features

109062578_input_1

109062578_input_2

Output Device Features

109062578_output

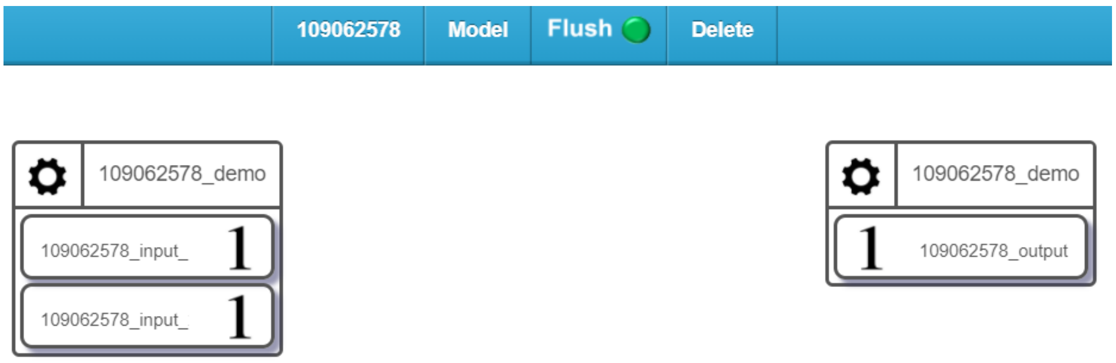
Add/Delete DF Type ● IDF ○ ODF Category: Feeling

☐109062611_Celsius

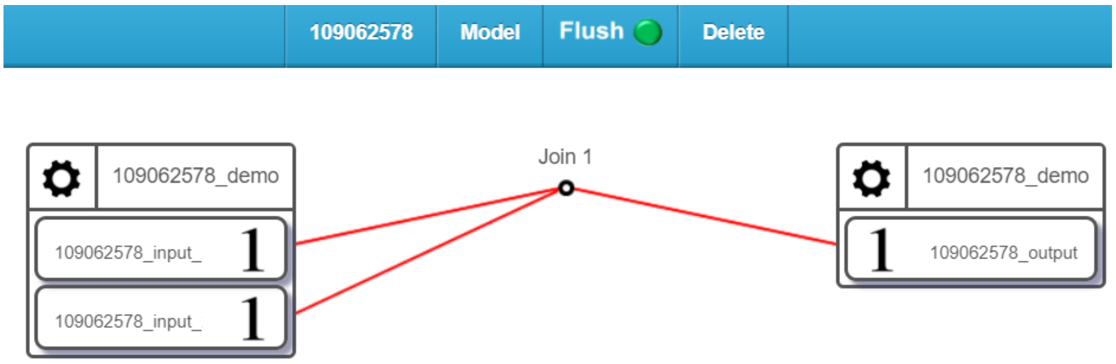
☐Thermometer

☐normThermometer

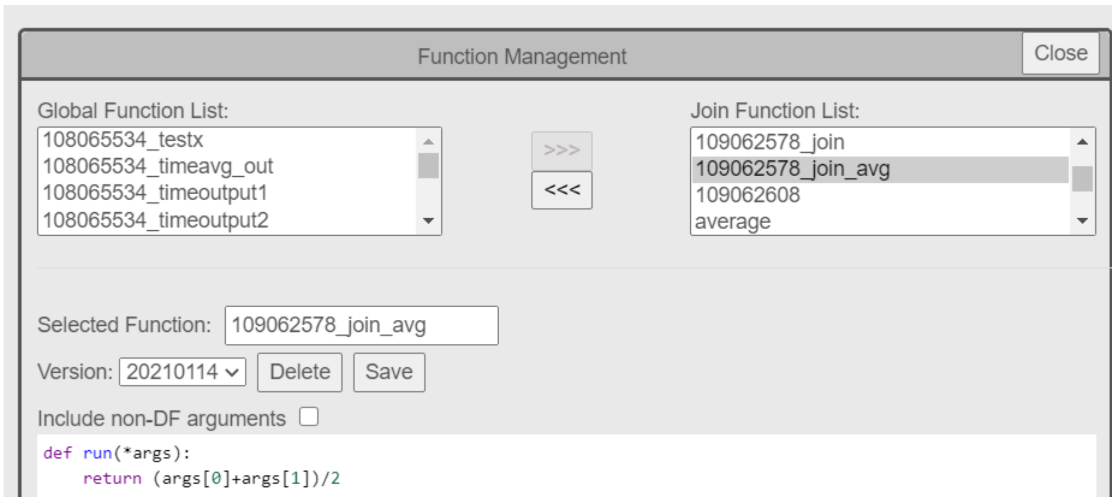
Step 4:新增 Device Model 到 IoTtalk 之中



Step 5:建立 Join 1



Step 6:建立 Function 109062578_join_avg



Global Function List:
108065534_testx
108065534_timeavg_out
108065534_timeoutput1
108065534_timeoutput2

Join Function List:
109062578_join
109062578_join_avg
109062608
average

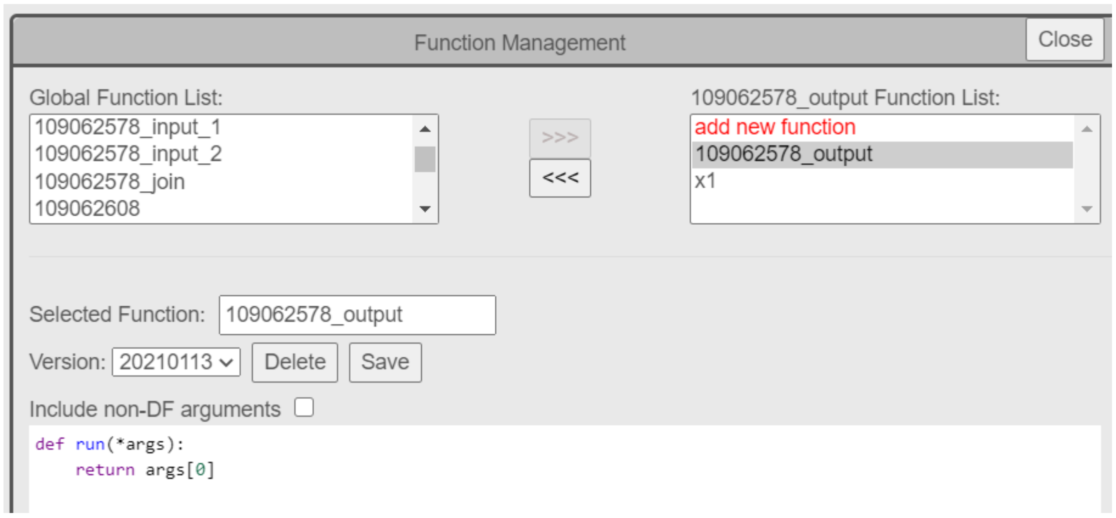
Selected Function: 109062578_join_avg

Version: 20210114

Include non-DF arguments ☐

```
def run(*args):
    return (args[0]+args[1])/2
```

Step 7:建立 Function 109062578_output



Global Function List:
109062578_input_1
109062578_input_2
109062578_join
109062608

109062578_output Function List:
add new function
109062578_output
x1

Selected Function: 109062578_output

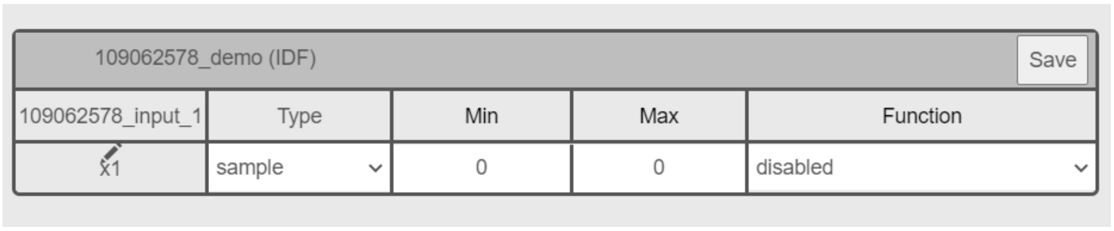
Version: 20210113

Include non-DF arguments ☐

```
def run(*args):
    return args[0]
```

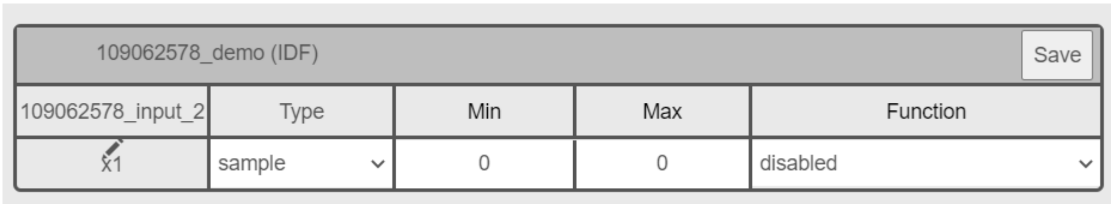
Step 8:設定 IDF,Join,ODF 的 Function

109062578_input_1:



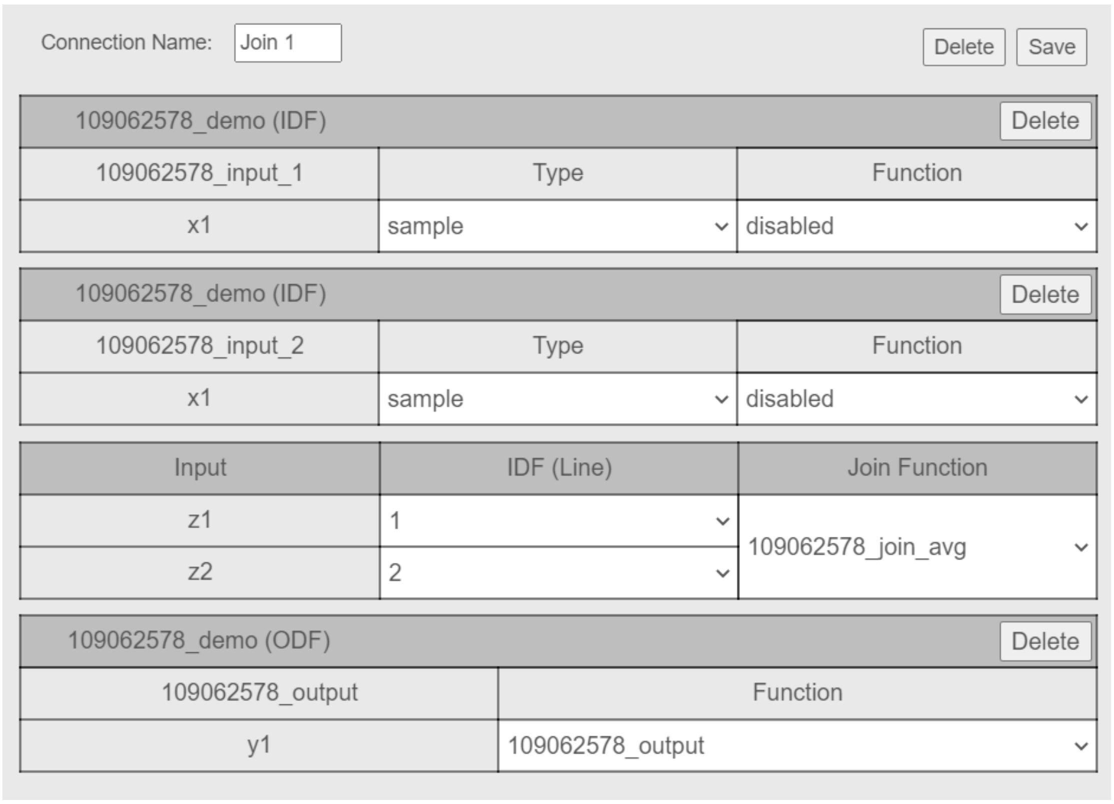
109062578_demo (IDF)				Save
109062578_input_1	Type	Min	Max	Function
x1	sample	0	0	disabled

109062578_input_2:



109062578_demo (IDF)				Save
109062578_input_2	Type	Min	Max	Function
x1	sample	0	0	disabled

Join 1:



Connection Name: Join 1 Delete Save

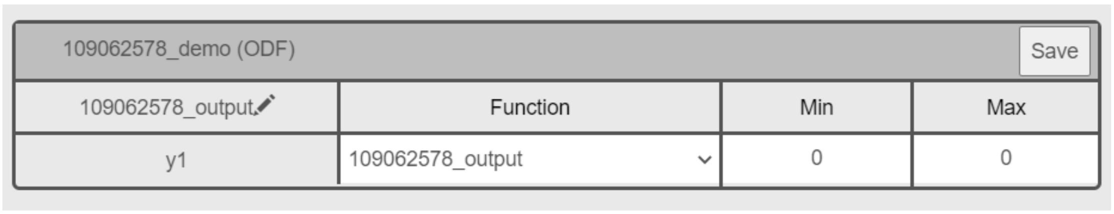
109062578_demo (IDF)		Delete
109062578_input_1	Type	Function
x1	sample	disabled

109062578_demo (IDF)		Delete
109062578_input_2	Type	Function
x1	sample	disabled

Input	IDF (Line)	Join Function
z1	1	109062578_join_avg
z2	2	

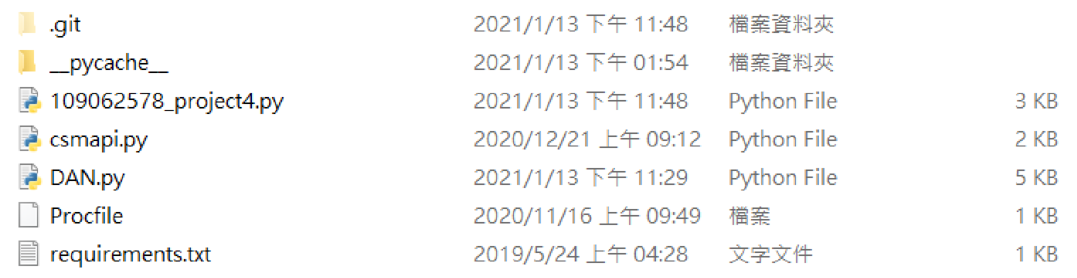
109062578_demo (ODF)	Delete
109062578_output	Function
y1	109062578_output

109062578_output:

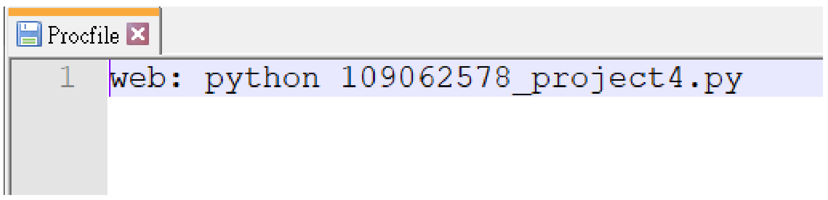


109062578_demo (ODF)			Save
109062578_output	Function	Min	Max
y1	109062578_output	0	0

Step 9:將 app.py 改成 109062578_project4.py 並更改 Procfile 中的內容

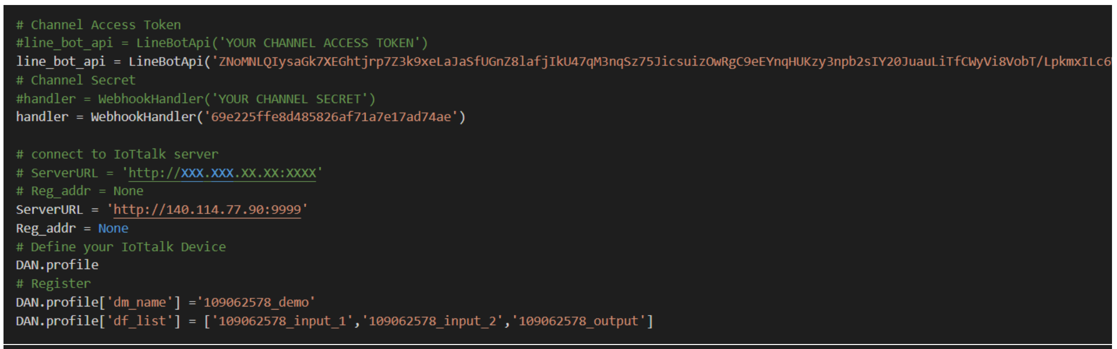


.git	2021/1/13 下午 11:48	檔案資料夾	
__pycache__	2021/1/13 下午 01:54	檔案資料夾	
109062578_project4.py	2021/1/13 下午 11:48	Python File	3 KB
csmapi.py	2020/12/21 上午 09:12	Python File	2 KB
DAN.py	2021/1/13 下午 11:29	Python File	5 KB
Procfile	2020/11/16 上午 09:49	檔案	1 KB
requirements.txt	2019/5/24 上午 04:28	文字文件	1 KB



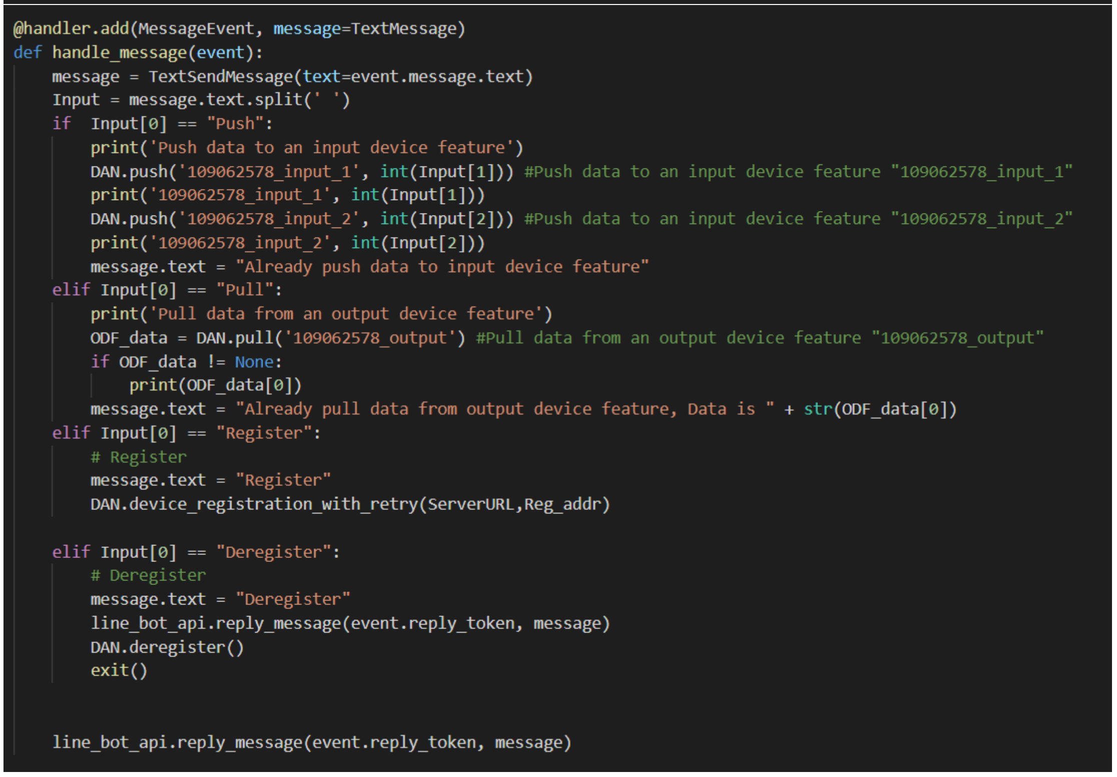
```
web: python 109062578_project4.py
```

Step 10:更改 app.py 的內容



```
# Channel Access Token
#line_bot_api = LineBotApi('YOUR CHANNEL ACCESS TOKEN')
line_bot_api = LineBotApi('ZNoMNLQIysaGk7XEGhtjrp7Z3k9xeLaJaSfUGnZ8lafjIkU47qM3nqSz75JicsuizOwRgC9eEYnqHUKzy3npb2sIY20JuauLiTfCWyVi8VobT/LpkmxILc6
# Channel Secret
#handler = WebhookHandler('YOUR CHANNEL SECRET')
handler = WebhookHandler('69e225ffe8d485826af71a7e17ad74ae')

# connect to IoTtalk server
# ServerURL = 'http://XXX.XXX.XX.XX:XXXX'
# Reg_addr = None
ServerURL = 'http://140.114.77.90:9999'
Reg_addr = None
# Define your IoTtalk Device
DAN.profile
# Register
DAN.profile['dm_name'] ='109062578_demo'
DAN.profile['df_list'] = ['109062578_input_1','109062578_input_2','109062578_output']
```

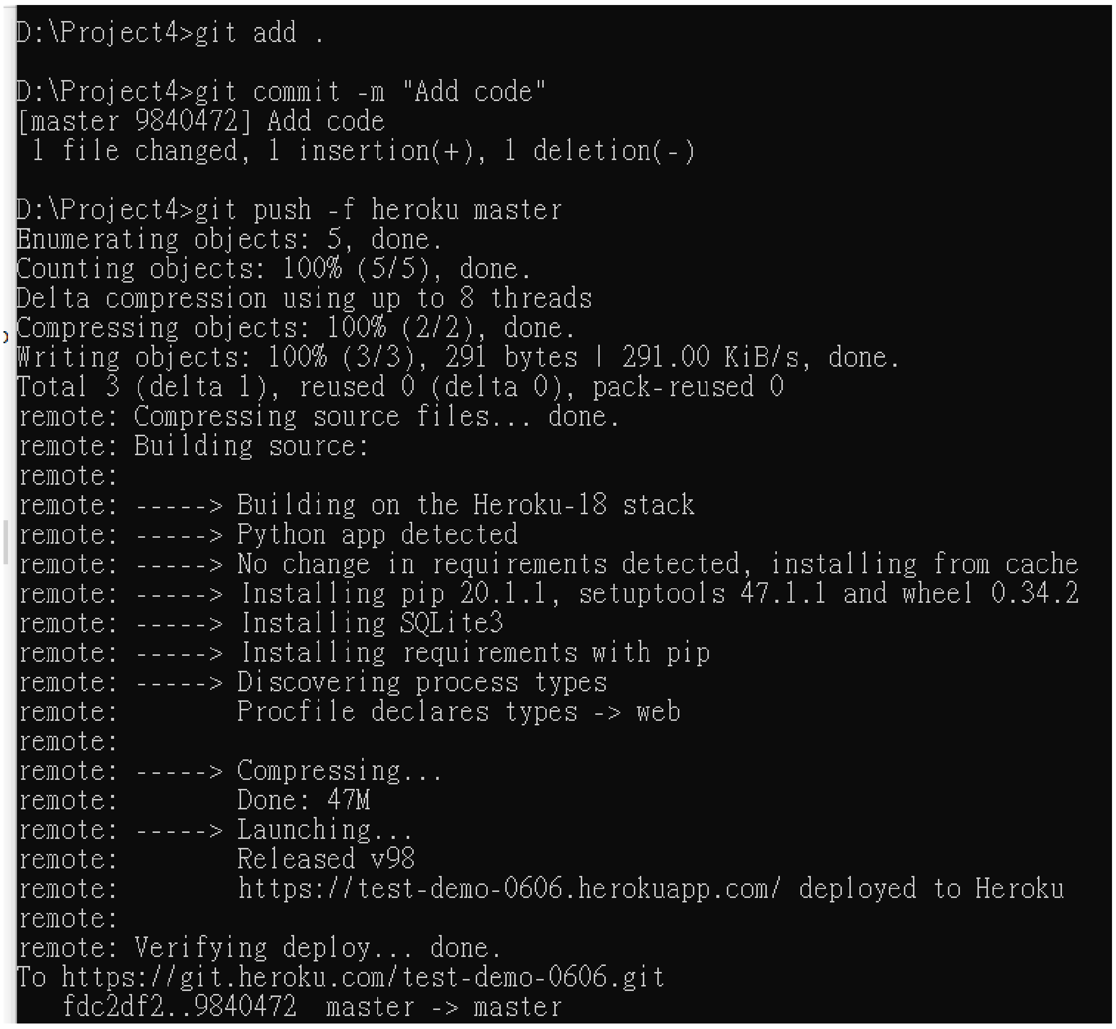


```
@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    message = TextSendMessage(text=event.message.text)
    Input = message.text.split(' ')
    if  Input[0] == "Push":
        print('Push data to an input device feature')
        DAN.push('109062578_input_1', int(Input[1])) #Push data to an input device feature "109062578_input_1"
        print('109062578_input_1', int(Input[1]))
        DAN.push('109062578_input_2', int(Input[2])) #Push data to an input device feature "109062578_input_2"
        print('109062578_input_2', int(Input[2]))
        message.text = "Already push data to input device feature"
    elif Input[0] == "Pull":
        print('Pull data from an output device feature')
        ODF_data = DAN.pull('109062578_output') #Pull data from an output device feature "109062578_output"
        if ODF_data != None:
            print(ODF_data[0])
        message.text = "Already pull data from output device feature, Data is " + str(ODF_data[0])
    elif Input[0] == "Register":
        # Register
        message.text = "Register"
        DAN.device_registration_with_retry(ServerURL,Reg_addr)

    elif Input[0] == "Deregister":
        # Deregister
        message.text = "Deregister"
        line_bot_api.reply_message(event.reply_token, message)
        DAN.deregister()
        exit()


    line_bot_api.reply_message(event.reply_token, message)
```

Step 11:上傳到 heroku



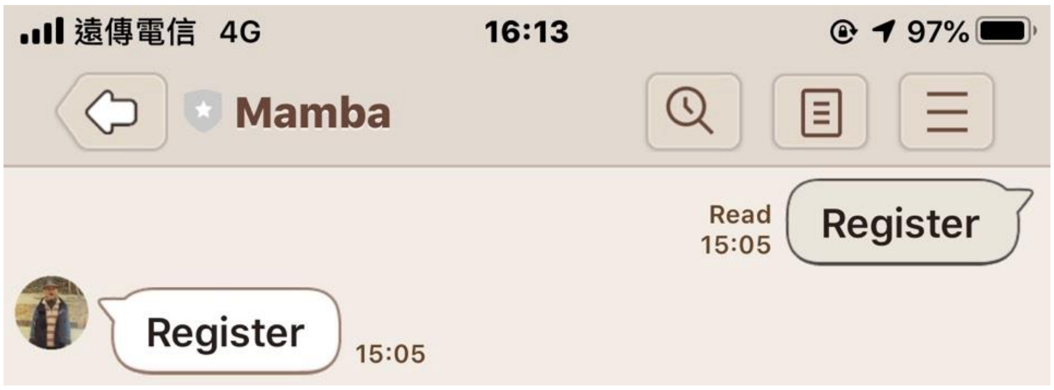
```
D:\Project4>git add .

D:\Project4>git commit -m "Add code"
[master 9840472] Add code
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\Project4>git push -f heroku master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 291 bytes | 291.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-18 stack
remote: -----> Python app detected
remote: -----> No change in requirements detected, installing from cache
remote: -----> Installing pip 20.1.1, setuptools 47.1.1 and wheel 0.34.2
remote: -----> Installing SQLite3
remote: -----> Installing requirements with pip
remote: -----> Discovering process types
remote:        Procfile declares types -> web
remote:
remote: -----> Compressing...
remote:        Done: 47M
remote: -----> Launching...
remote:        Released v98
remote:        https://test-demo-0606.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/test-demo-0606.git
   fdc2df2..9840472  master -> master
```

Step 12:使用 linebot 輸入 Register 後,當 linebot 回傳 Register 代表已經產生一個已註冊的 Device Model

Linebot:



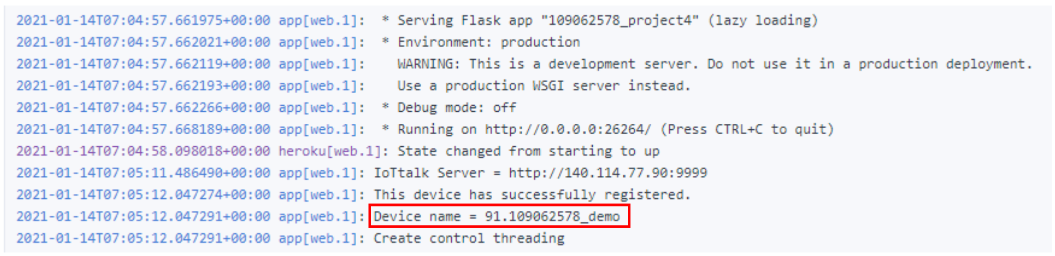
遠傳電信 4G 16:13 97%

Mamba

Read 15:05 Register

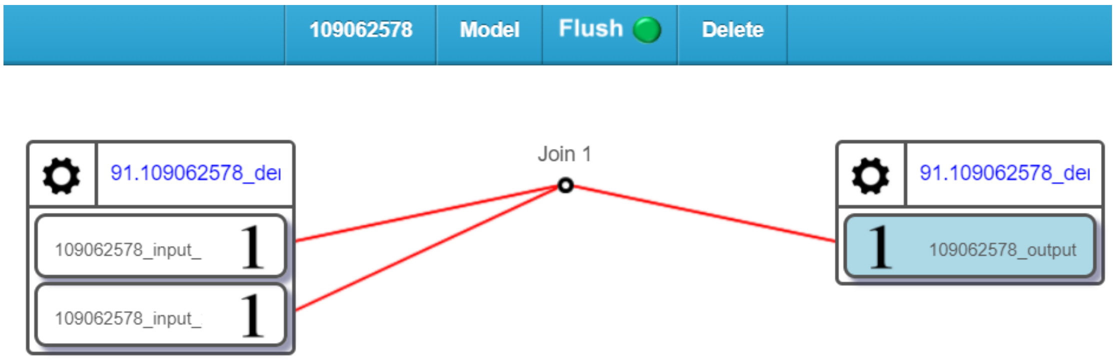
Register 15:05

Log:



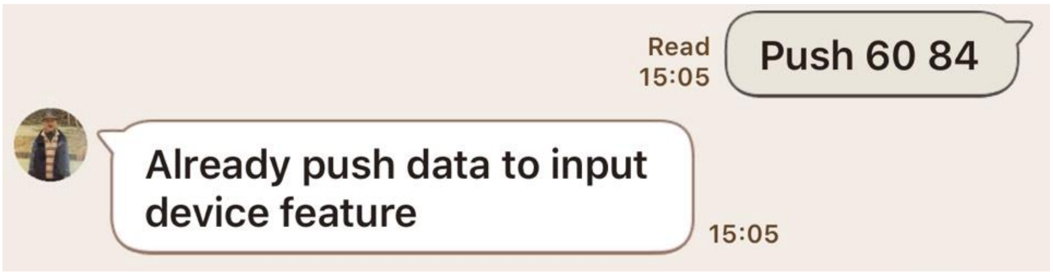
```
2021-01-14T07:04:57.661975+00:00 app[web.1]:  * Serving Flask app "109062578_project4" (lazy loading)
2021-01-14T07:04:57.662021+00:00 app[web.1]:  * Environment: production
2021-01-14T07:04:57.662119+00:00 app[web.1]:    WARNING: This is a development server. Do not use it in a production deployment.
2021-01-14T07:04:57.662193+00:00 app[web.1]:    Use a production WSGI server instead.
2021-01-14T07:04:57.662266+00:00 app[web.1]:  * Debug mode: off
2021-01-14T07:04:57.668189+00:00 app[web.1]:  * Running on http://0.0.0.0:26264/ (Press CTRL+C to quit)
2021-01-14T07:04:58.098018+00:00 heroku[web.1]: State changed from starting to up
2021-01-14T07:05:11.486490+00:00 app[web.1]: IoTtalk Server = http://140.114.77.90:9999
2021-01-14T07:05:12.047274+00:00 app[web.1]: This device has successfully registered.
2021-01-14T07:05:12.047291+00:00 app[web.1]: Device name = 91.109062578_demo
2021-01-14T07:05:12.047291+00:00 app[web.1]: Create control threading
```

Step 13:到 IoTtalk 中選擇成剛剛註冊的 91.109062578_demo



Step 14:輸入 Pull 數字 1 數字 2 給 linebot 後,linebot 回傳 Already push data to input device feature 代表已經將兩個數字

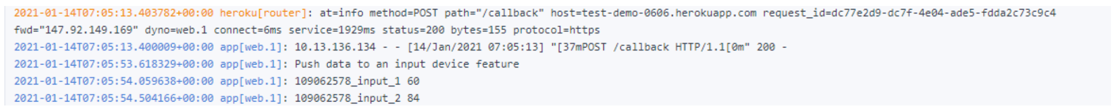
Linebot:



Read
15:05
Push 60 84

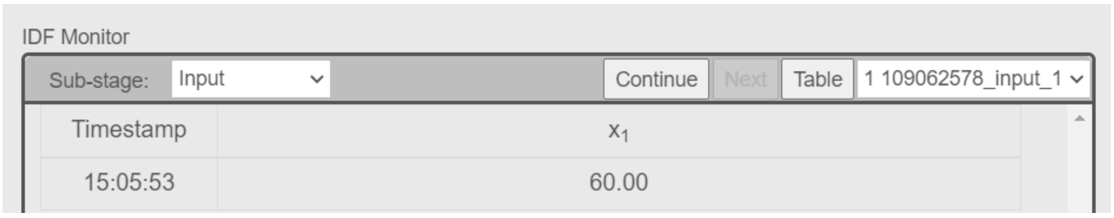
Already push data to input device feature
15:05

Log:



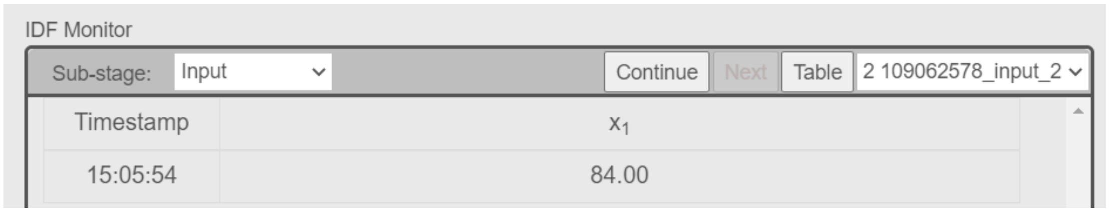
```
2021-01-14T07:05:13.403782+00:00 heroku[router]: at=info method=POST path="/callback" host=test-demo-0606.herokuapp.com request_id=dc77e2d9-dc7f-4e04-ade5-fdda2c73c9c4
fwd="147.92.149.169" dyno=web.1 connect=6ms service=1929ms status=200 bytes=155 protocol=https
2021-01-14T07:05:13.400009+00:00 app[web.1]: 10.13.136.134 - - [14/Jan/2021 07:05:13] "[37mPOST /callback HTTP/1.1[0m" 200 -
2021-01-14T07:05:53.618329+00:00 app[web.1]: Push data to an input device feature
2021-01-14T07:05:54.059638+00:00 app[web.1]: 109062578_input_1 60
2021-01-14T07:05:54.504166+00:00 app[web.1]: 109062578_input_2 84
```

109062578_input_1:



Timestamp	x_1
15:05:53	60.00

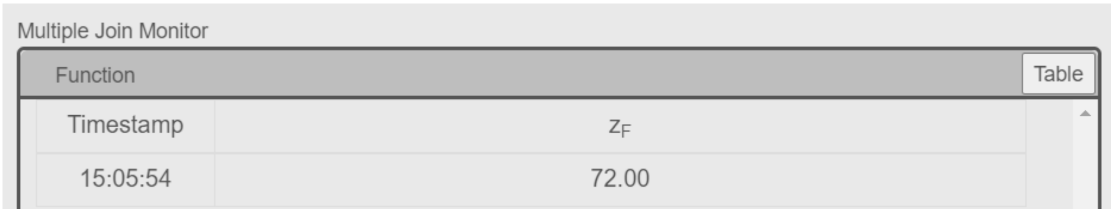
109062578_input_2:



Timestamp	x_1
15:05:54	84.00

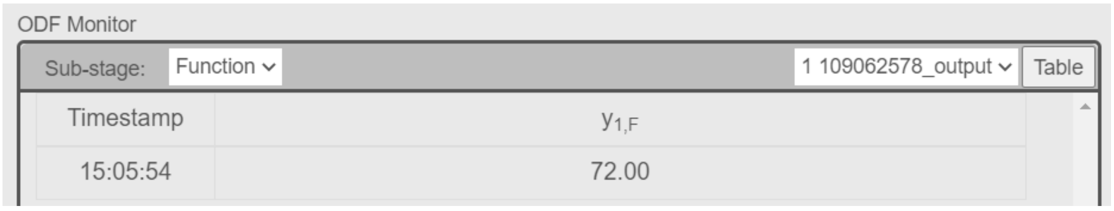
Step 15:經過 Join 1 算完兩數的平均後 Input 到 ODF

Join:



Timestamp	z_F
15:05:54	72.00

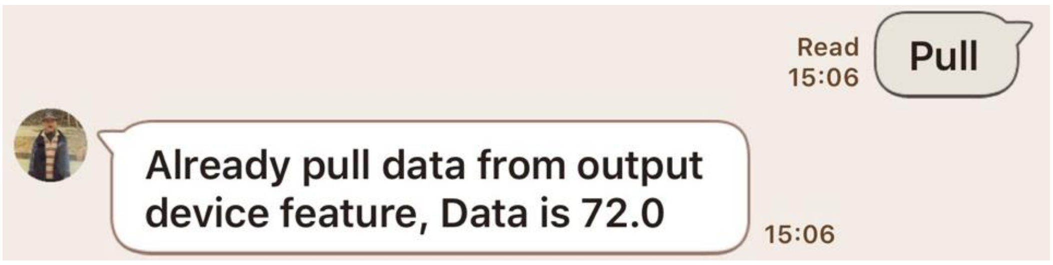
109062578_output:



Timestamp	$y_{1,F}$
15:05:54	72.00

Step 16:輸入 Pull 將 ODF 中的值取出來

Linebot:



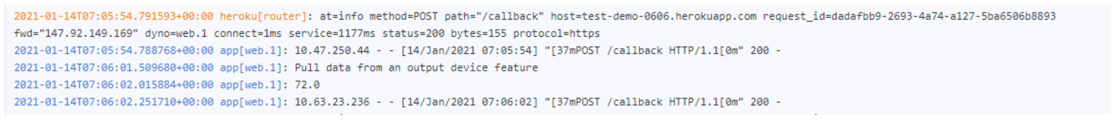
Pull

Read
15:06

Already pull data from output device feature, Data is 72.0

15:06

Log:

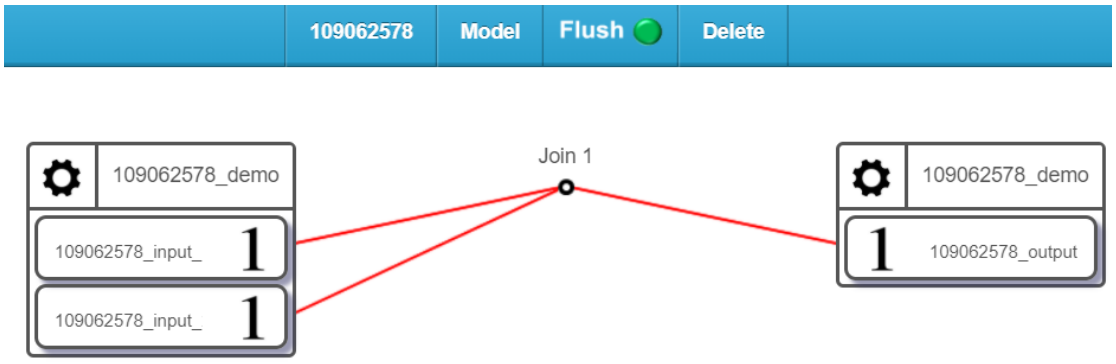


```
2021-01-14T07:05:54.791593+00:00 heroku[router]: at=info method=POST path="/callback" host=test-demo-0606.herokuapp.com request_id=dadafbb9-2693-4a74-a127-5ba6506b8893
fwd="147.92.149.169" dyno=web.1 connect=1ms service=1177ms status=200 bytes=155 protocol=https
2021-01-14T07:05:54.788768+00:00 app[web.1]: 10.47.250.44 - - [14/Jan/2021 07:05:54] "[37mPOST /callback HTTP/1.1[0m" 200 -
2021-01-14T07:06:01.509680+00:00 app[web.1]: Pull data from an output device feature
2021-01-14T07:06:02.015884+00:00 app[web.1]: 72.0
2021-01-14T07:06:02.251710+00:00 app[web.1]: 10.63.23.236 - - [14/Jan/2021 07:06:02] "[37mPOST /callback HTTP/1.1[0m" 200 -
```

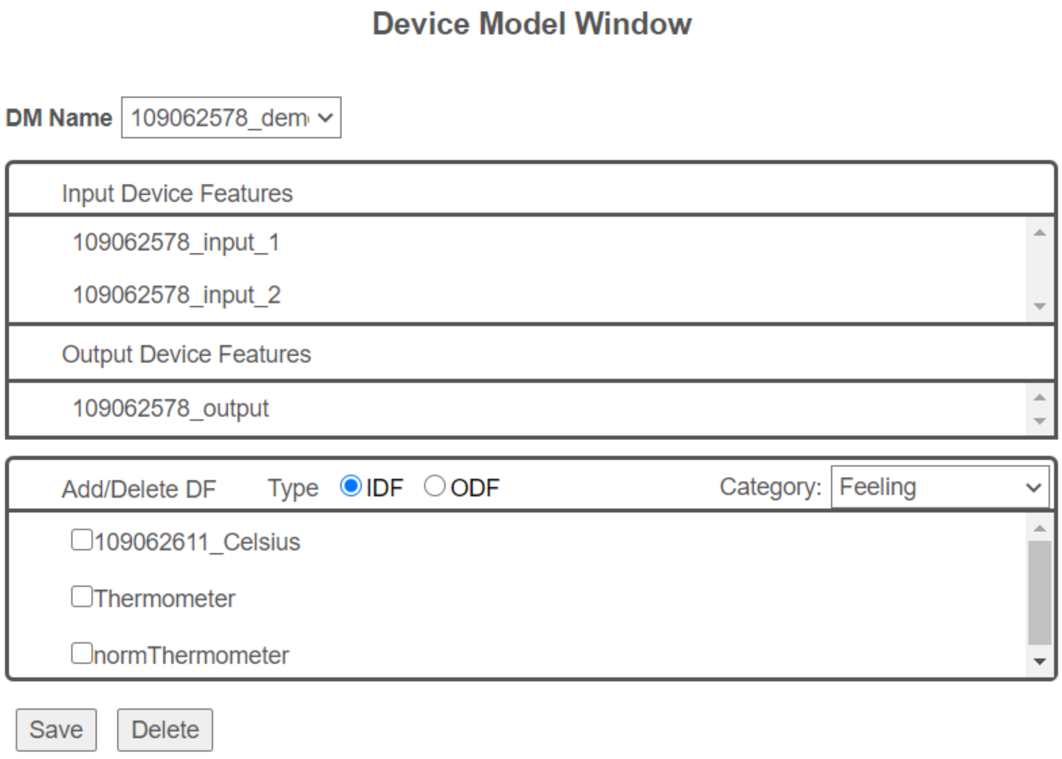
3. Screenshots

(1)IoTtalk GUI connection, DM/DF creation, join connection:

IoTtalk GUI connection:



DM/DF creation:


Device Model Window

DM Name 109062578_dem

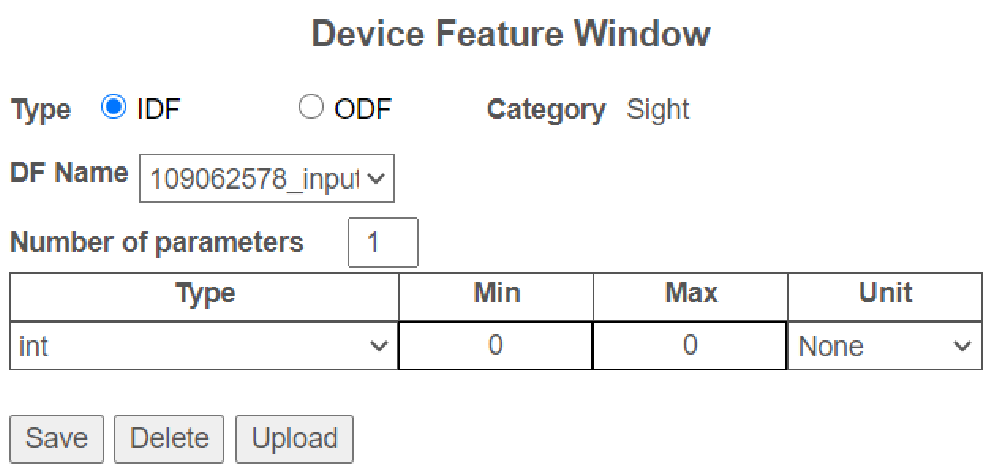
Input Device Features
- 109062578_input_1
- 109062578_input_2

Output Device Features
- 109062578_output

Add/Delete DF Type [✓ IDF] ODF Category: Feeling

- ☐ 109062611_Celsius
- ☐ Thermometer
- ☐ normThermometer

Save Delete



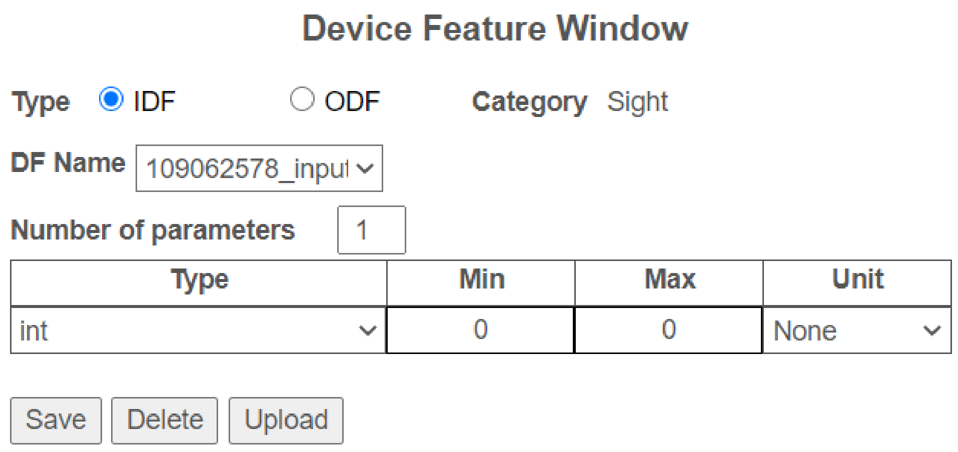
Device Feature Window

Type [✓ IDF] ODF Category Sight

DF Name 109062578_input

Number of parameters 1

Type	Min	Max	Unit
int	0	0	None



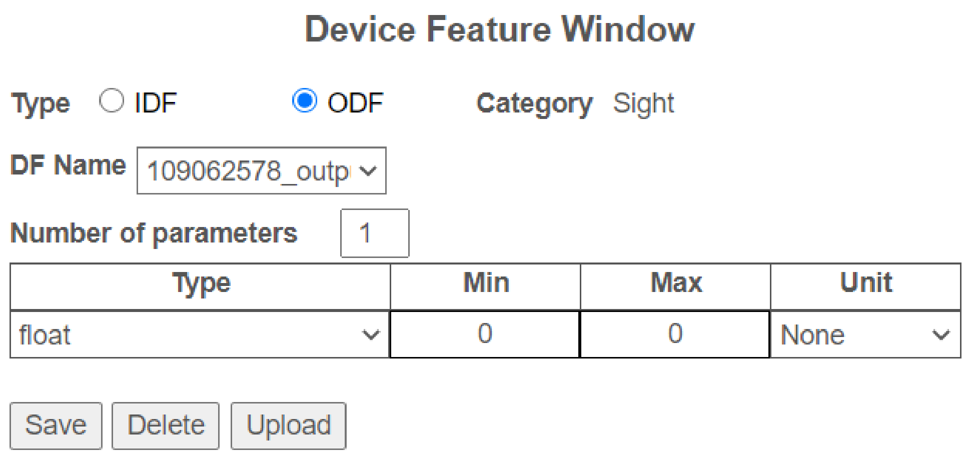
Device Feature Window

Type [✓ IDF] ODF Category Sight

DF Name 109062578_input

Number of parameters 1

Type	Min	Max	Unit
int	0	0	None



Device Feature Window

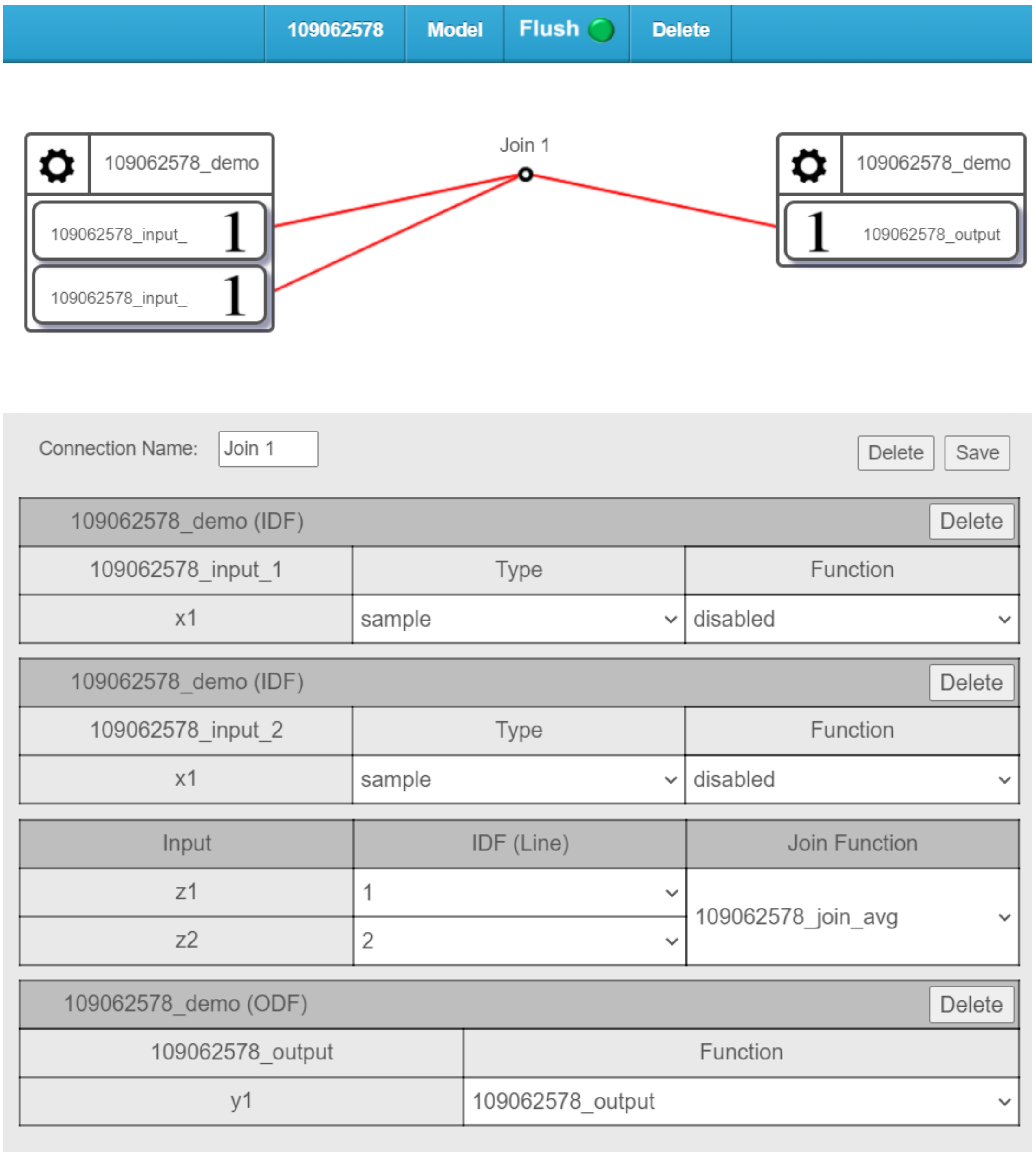
Type IDF [✓ ODF] Category Sight

DF Name 109062578_outp

Number of parameters 1

Type	Min	Max	Unit
float	0	0	None

Join connection:



Connection Name: Join 1 Delete Save

109062578_demo (IDF)		Delete
109062578_input_1	Type	Function
x1	sample	disabled

109062578_demo (IDF)		Delete
109062578_input_2	Type	Function
x1	sample	disabled

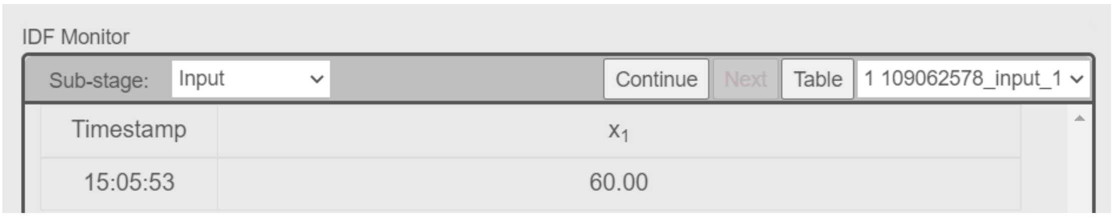
Input	IDF (Line)	Join Function
z1	1	109062578_join_avg
z2	2	

109062578_demo (ODF)	Delete
109062578_output	Function
y1	109062578_output

(2)IDF/ODF Monitor:

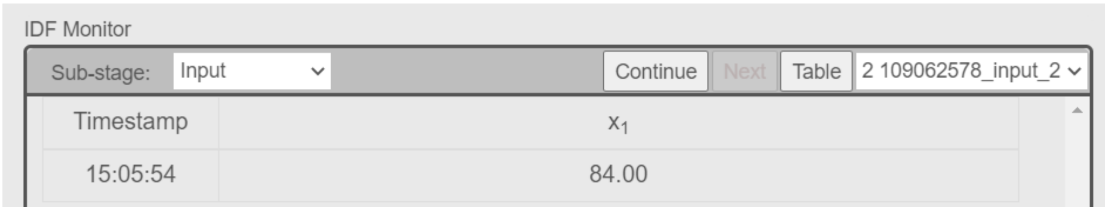
IDF:

109062578_input_1:



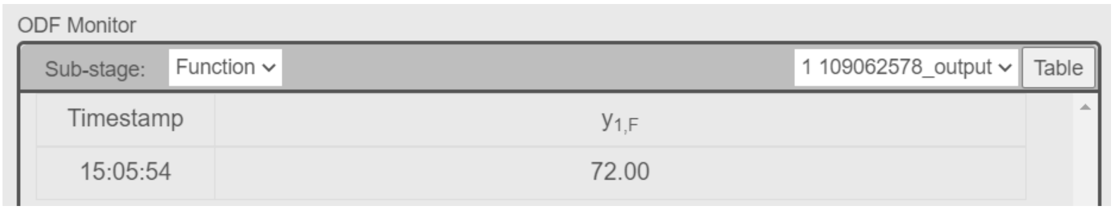
Timestamp	x_1
15:05:53	60.00

109062578_input_2:



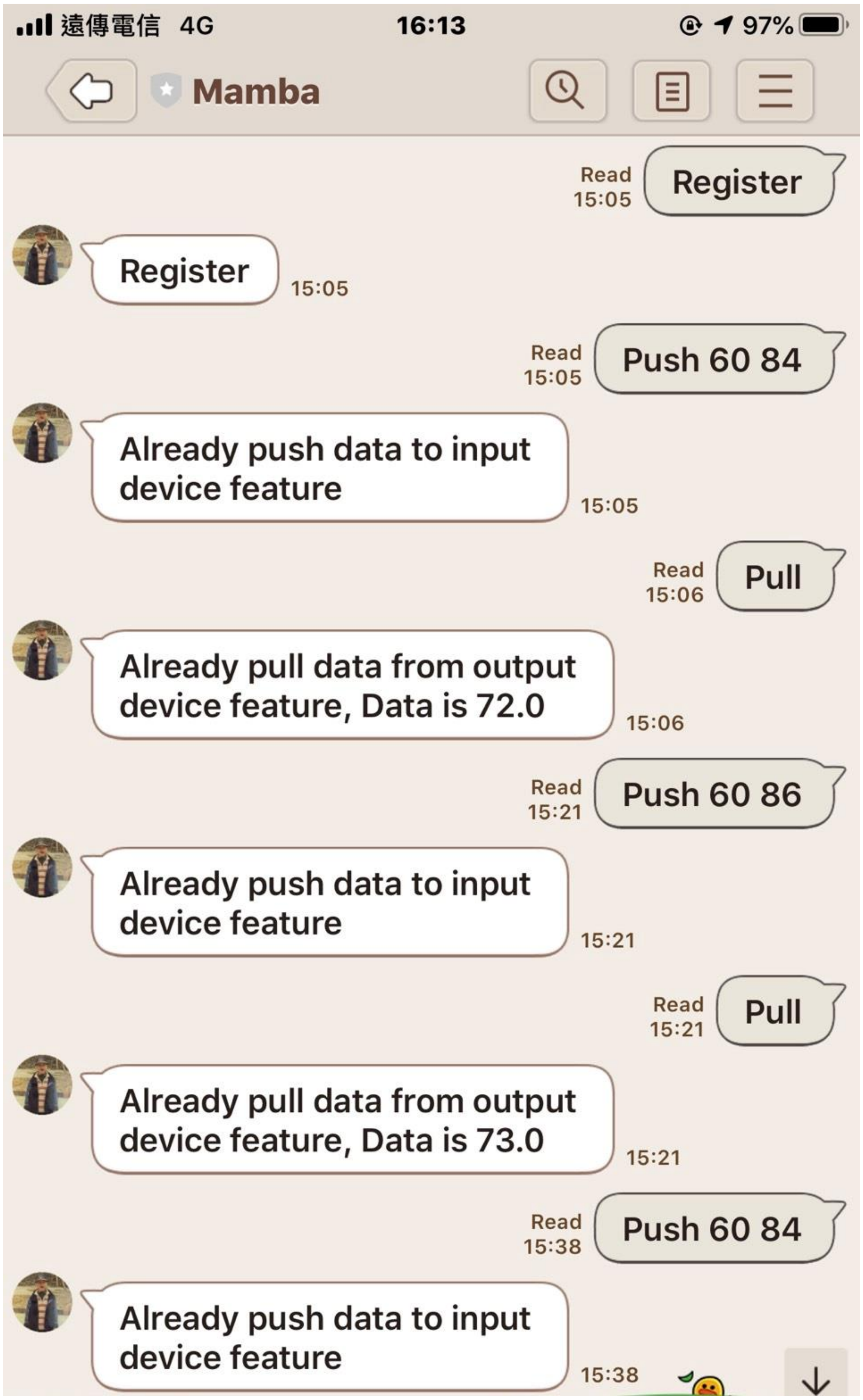
Timestamp	x_1
15:05:54	84.00

ODF:



Timestamp	$y_{1,F}$
15:05:54	72.00

(3)LineBotresult:



遠傳電信 4G 16:13 97%

Mamba

Read 15:05 Register

Register 15:05

Read 15:05 Push 60 84

Already push data to input device feature 15:05

Read 15:06 Pull

Already pull data from output device feature, Data is 72.0 15:06

Read 15:21 Push 60 86

Already push data to input device feature 15:21

Read 15:21 Pull

Already pull data from output device feature, Data is 73.0 15:21

Read 15:38 Push 60 84

Already push data to input device feature 15:38

4. What you learn

學到如何在 IoTtalk 上建立自己的 IDF,ODF 和 DM,並且透過 linebot 來控制,透過 linebot 下指令來註冊並且透過,指令輸入兩個 input 到 IoTtalk 中計算,IoTtalk 透過 join 算完後輸入到 ODF 之中後,則可以在使用指令來從 IoTtalk 中取出計算完的兩數平均值。