OTA Progress Status

0. Table of Contents

1. Description

This file will read/process the IoT Jobs. If the firmware version is new (if the firmware version from the job document is greater than the current firmware version), start downloading the new firmware from the S3 bucket described in the job document. Meanwhile, it will publish the current downloading progress (by bytes) to IoT with the topic *'CarVi/test/firmware/[IMEI]'.* When the downloading is done, the IoT Jobs will be completed.

2. Module and Function

2-1. Module

class JobsMessageProcessor() : Default class offered by AWS for the OTA (at line 25).
class OTAProgressBar(): Main class for producing OTA progress bar (at line 128).
(This will catch the downloaded chunk and publish the current progress status to IoT.)

2-2. Function

TimeStamp(): outputs OTA start time (at line 107)
GetJobInfo(): outputs Job documents information (at line 113)
listener(): subscribe topic (at line 159)
firmwareCallback(): callback function (at line 249)

3. Implementation

3-1. Requirement

- IoT Job document (*rasp-ota-job-docs.json*)
- JobProgress.py file

Parameters (*changeable*):

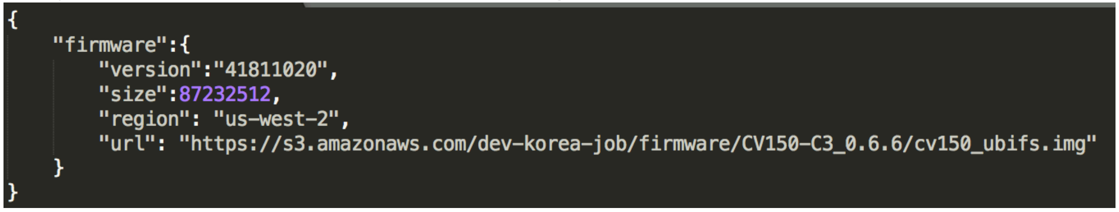
- **Time zone is currently in '*Chicago*'** (at line 108)

- Current version is *41811019*. Always lower than the version from the job document file which is *41811020* (at line 311).

3-2. Create an IoT Job

3-2-1. IoT Job document

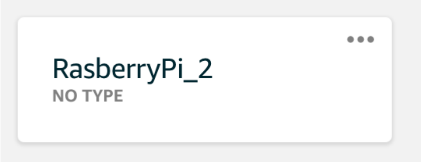
The job document should contain following information:



```
{
    "firmware":{
        "version":"41811020",
        "size":87232512,
        "region": "us-west-2",
        "url": "https://s3.amazonaws.com/dev-korea-job/firmware/CV150-C3_0.6.6/cv150_ubifs.img"
    }
}
```

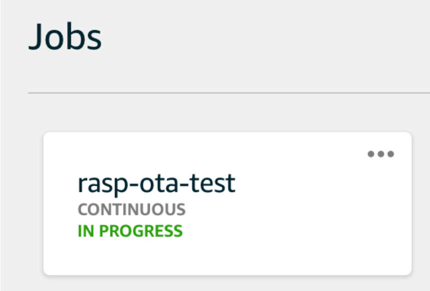
3-2-2. Create an IoT Job

- Thing name: RasberryPi_2 (IMEI)



RasberryPi_2
NO TYPE

- Job name: *rasp-ota-test (continuous)*



Jobs

rasp-ota-test
CONTINUOUS
IN PROGRESS

3-3. Run the Job file

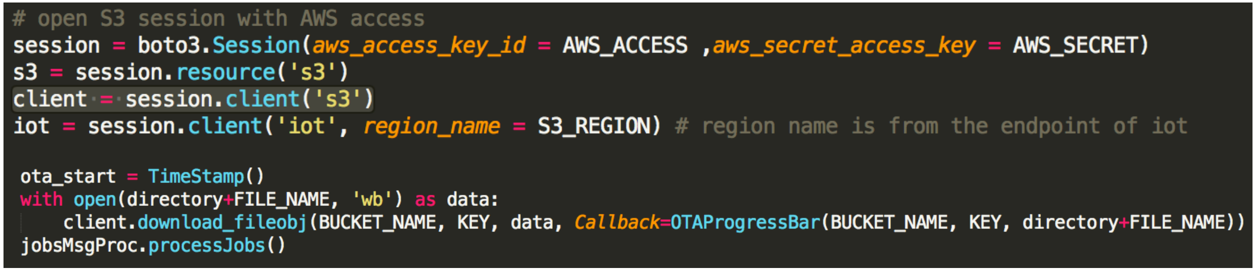
- Job file name: *JobProgress.py*
- How to command

python JobProgress.py -e [endpoint] -r [rootCA] -c [certificate] -k [private key] -n [Thing name]

i.e.
python JobProgress.py -e azozmhv0nd77.iot.us-west-2.amazonaws.com -r root_CA.pem -c 3ddd7abd2c-certificate.pem.crt -k 3ddd7abd2c-private.pem.key -n RasberryPi_2

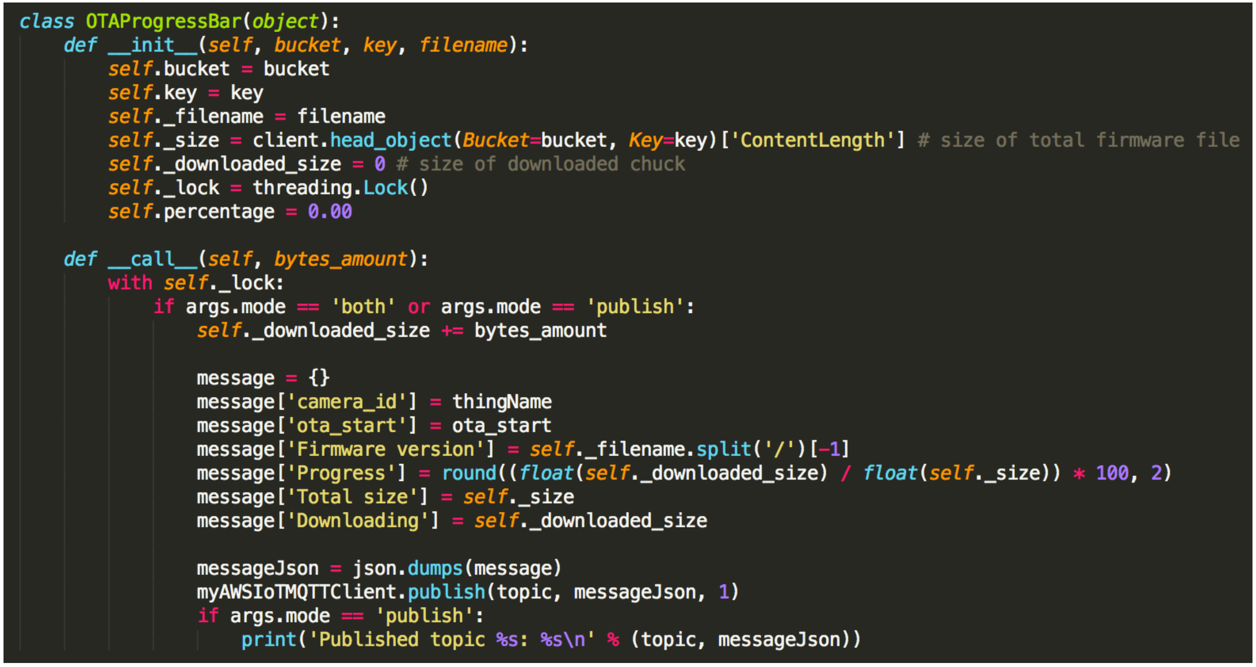
- Main body
While downloading the firmware, it will call '*OTAProgressBar*' class to produce the progress status and publish it to the Topic.

Downloading the firmware part:



```
# open S3 session with AWS access
session = boto3.Session(aws_access_key_id = AWS_ACCESS ,aws_secret_access_key = AWS_SECRET)
s3 = session.resource('s3')
client = session.client('s3')
iot = session.client('iot', region_name = S3_REGION) # region name is from the endpoint of iot

ota_start = TimeStamp()
with open(directory+FILE_NAME, 'wb') as data:
    client.download_fileobj(BUCKET_NAME, KEY, data, Callback=OTAProgressBar(BUCKET_NAME, KEY, directory+FILE_NAME))
jobsMsgProc.processJobs()
```

Producing the progress status:

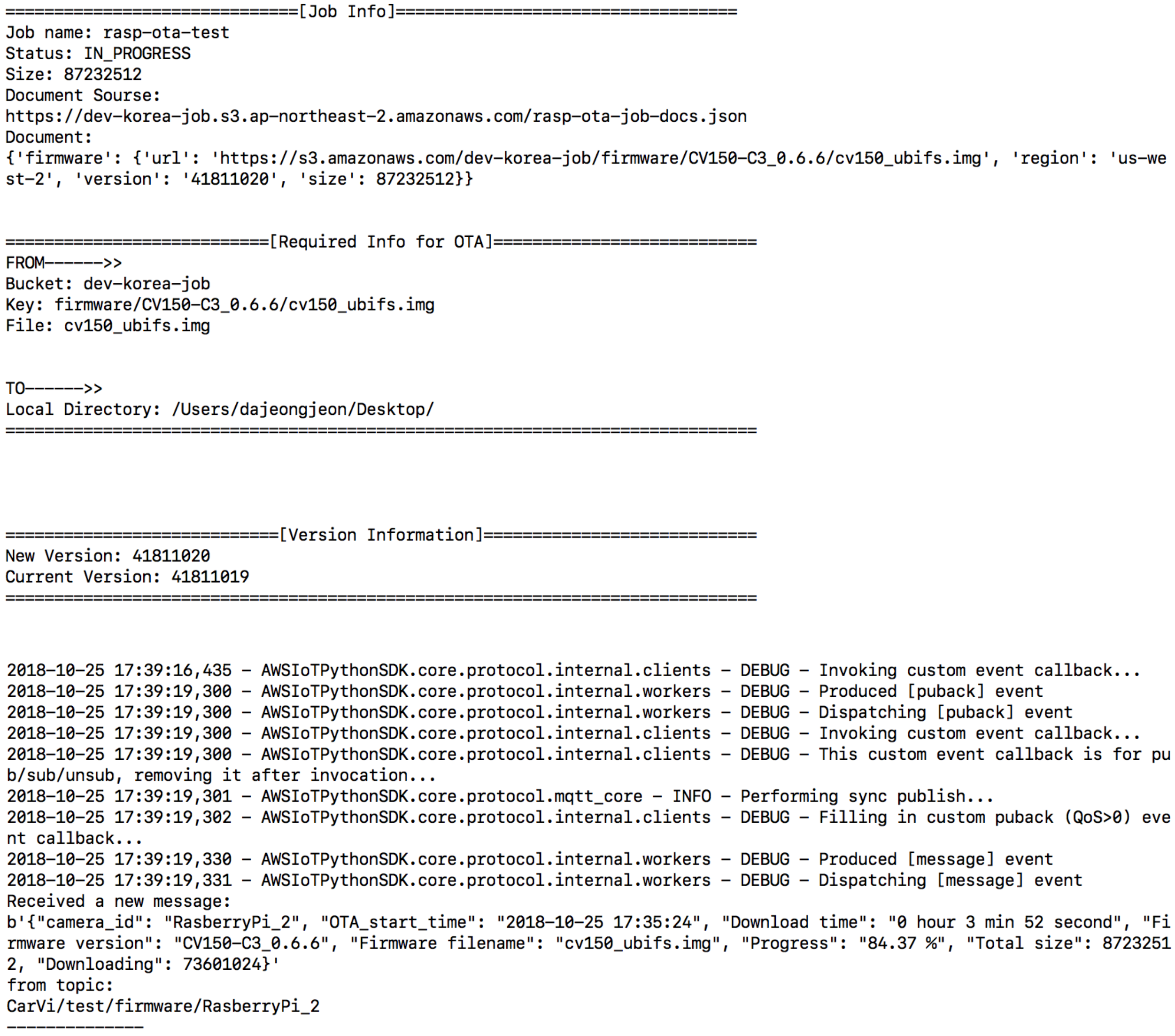


```
class OTAProgressBar(object):
    def __init__(self, bucket, key, filename):
        self.bucket = bucket
        self.key = key
        self._filename = filename
        self._size = client.head_object(Bucket=bucket, Key=key)['ContentLength'] # size of total firmware file
        self._downloaded_size = 0 # size of downloaded chuck
        self._lock = threading.Lock()
        self.percentage = 0.00

    def __call__(self, bytes_amount):
        with self._lock:
            if args.mode == 'both' or args.mode == 'publish':
                self._downloaded_size += bytes_amount

                message = {}
                message['camera_id'] = thingName
                message['ota_start'] = ota_start
                message['Firmware version'] = self._filename.split('/')[-1]
                message['Progress'] = round((float(self._downloaded_size) / float(self._size)) * 100, 2)
                message['Total size'] = self._size
                message['Downloading'] = self._downloaded_size

                messageJson = json.dumps(message)
                myAWSIoTMQTTClient.publish(topic, messageJson, 1)
                if args.mode == 'publish':
                    print('Published topic %s: %s\n' % (topic, messageJson))
```

- Steps

1. Turn on the device (*RaspberryPi*).
2. Start processing the Job.
3. Read the job document and get the information *(firmware version, size, and location in S3).*
4. Compare the current version and the version in the job document.
5. If it has the NEW version, start processing OTA
6. When the OTA is processing (*downloading firmware file from S3 bucket)*, it will catch the chunk of downloaded bytes and publish the current OTA status to IoT Topic.
7. When the OTA is finished, the Job is completed.

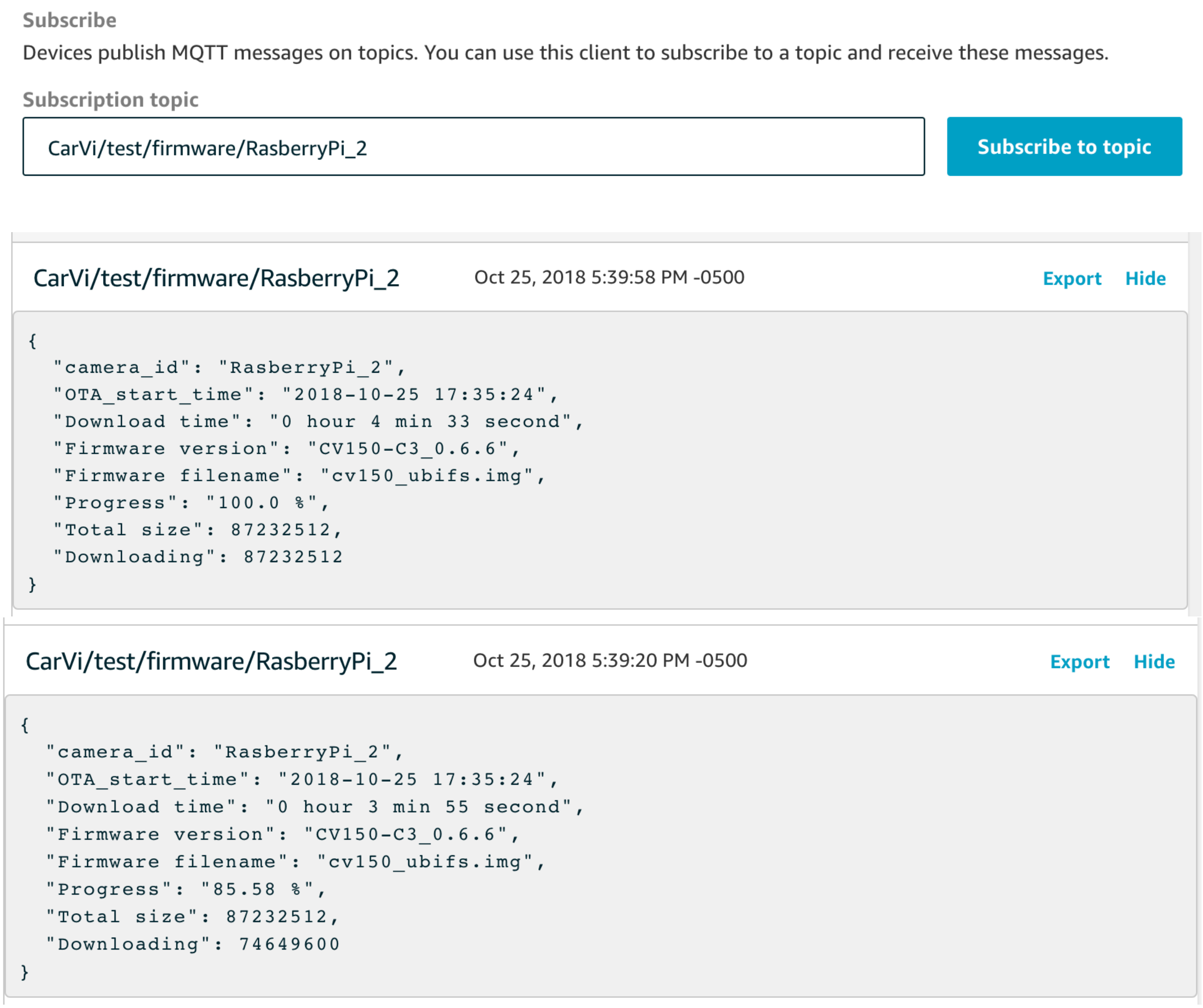
- Output



```
==============================[Job Info]===================================
Job name: rasp-ota-test
Status: IN_PROGRESS
Size: 87232512
Document Sourse:
https://dev-korea-job.s3.ap-northeast-2.amazonaws.com/rasp-ota-job-docs.json
Document:
{'firmware': {'url': 'https://s3.amazonaws.com/dev-korea-job/firmware/CV150-C3_0.6.6/cv150_ubifs.img', 'region': 'us-we
st-2', 'version': '41811020', 'size': 87232512}}


===========================[Required Info for OTA]===========================
FROM------>>
Bucket: dev-korea-job
Key: firmware/CV150-C3_0.6.6/cv150_ubifs.img
File: cv150_ubifs.img


TO------>>
Local Directory: /Users/dajeongjeon/Desktop/
=============================================================================




============================[Version Information]============================
New Version: 41811020
Current Version: 41811019
=============================================================================



2018-10-25 17:39:16,435 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...
2018-10-25 17:39:19,300 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Produced [puback] event
2018-10-25 17:39:19,300 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Dispatching [puback] event
2018-10-25 17:39:19,300 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...
2018-10-25 17:39:19,300 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - This custom event callback is for pu
b/sub/unsub, removing it after invocation...
2018-10-25 17:39:19,301 - AWSIoTPythonSDK.core.protocol.mqtt_core - INFO - Performing sync publish...
2018-10-25 17:39:19,302 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Filling in custom puback (QoS>0) eve
nt callback...
2018-10-25 17:39:19,330 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Produced [message] event
2018-10-25 17:39:19,331 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Dispatching [message] event
Received a new message:
b'{"camera_id": "RasberryPi_2", "OTA_start_time": "2018-10-25 17:35:24", "Download time": "0 hour 3 min 52 second", "Fi
rmware version": "CV150-C3_0.6.6", "Firmware filename": "cv150_ubifs.img", "Progress": "84.37 %", "Total size": 8723251
2, "Downloading": 73601024}'
from topic:
CarVi/test/firmware/RasberryPi_2
--------------
```

- Subscribe topic output

Topic: *'CarVi/test/firmware/RasberryPi_2'*



Subscribe

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic

CarVi/test/firmware/RasberryPi_2

CarVi/test/firmware/RasberryPi_2 Oct 25, 2018 5:39:58 PM -0500

```
{
  "camera_id": "RasberryPi_2",
  "OTA_start_time": "2018-10-25 17:35:24",
  "Download time": "0 hour 4 min 33 second",
  "Firmware version": "CV150-C3_0.6.6",
  "Firmware filename": "cv150_ubifs.img",
  "Progress": "100.0 %",
  "Total size": 87232512,
  "Downloading": 87232512
}
```

CarVi/test/firmware/RasberryPi_2 Oct 25, 2018 5:39:20 PM -0500

```
{
  "camera_id": "RasberryPi_2",
  "OTA_start_time": "2018-10-25 17:35:24",
  "Download time": "0 hour 3 min 55 second",
  "Firmware version": "CV150-C3_0.6.6",
  "Firmware filename": "cv150_ubifs.img",
  "Progress": "85.58 %",
  "Total size": 87232512,
  "Downloading": 74649600
}
```