



## CPE 314: Computer Networks

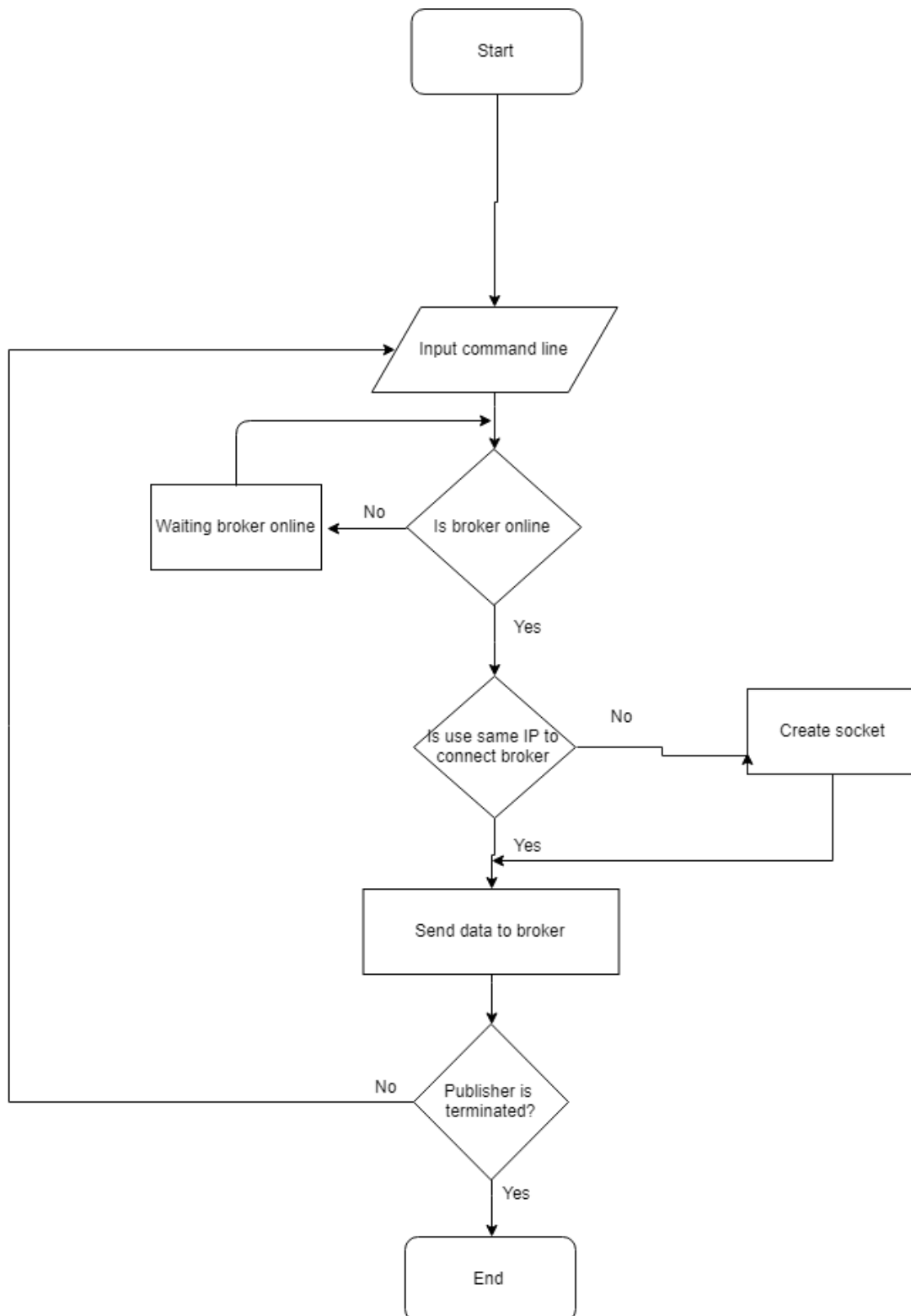
### Project I

#### Group member

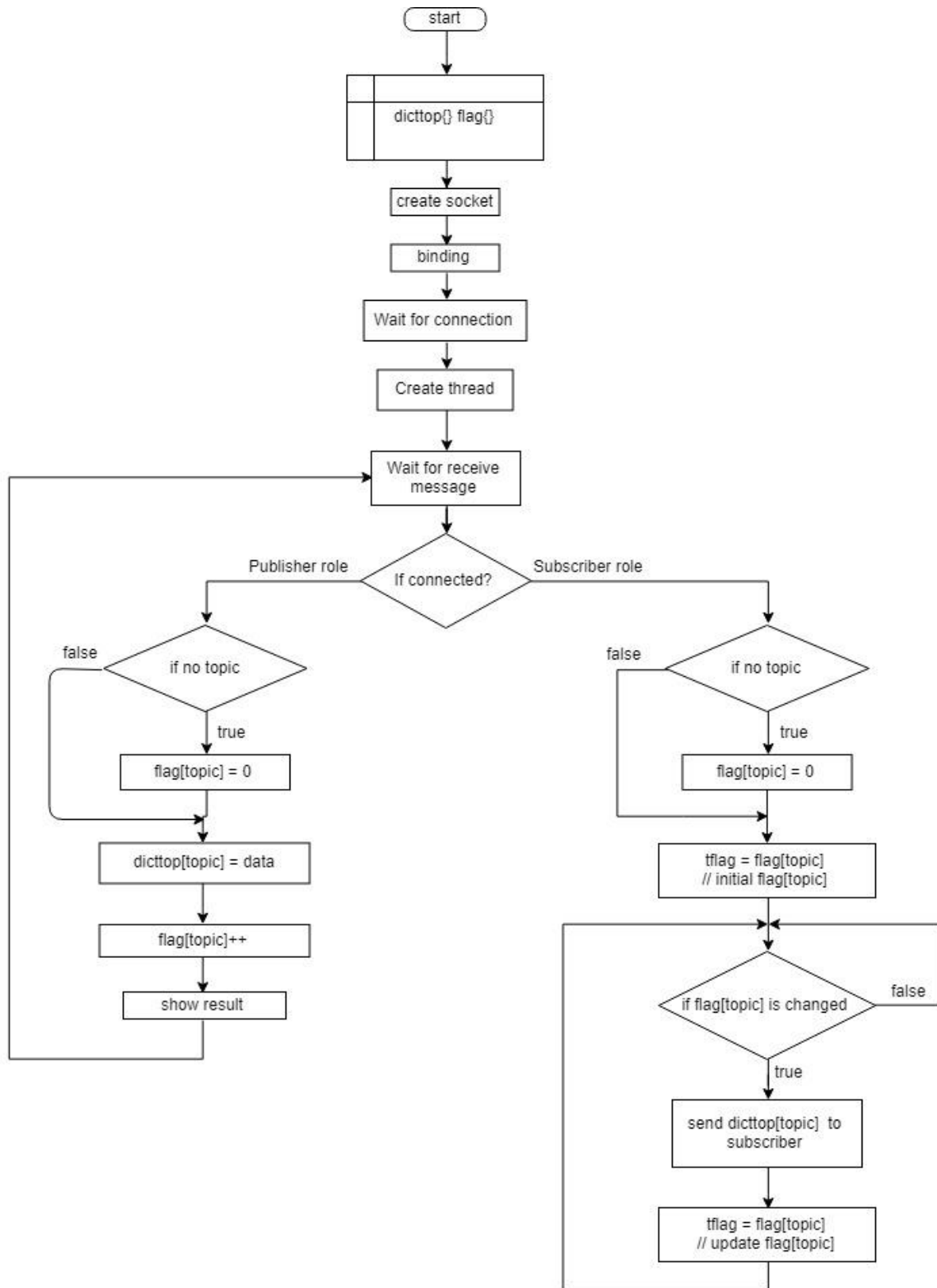
1.	Kanin	Ngamsanlerd	59070501004
2.	Kunuch	Terditinwitid	59070501005
3.	Chanothai	Aeksitthichok	59070501015
4.	Sahawat	Kwankaew	59070501072
5.	Sarin	Anuttranon	59070501073

## Program flowchart

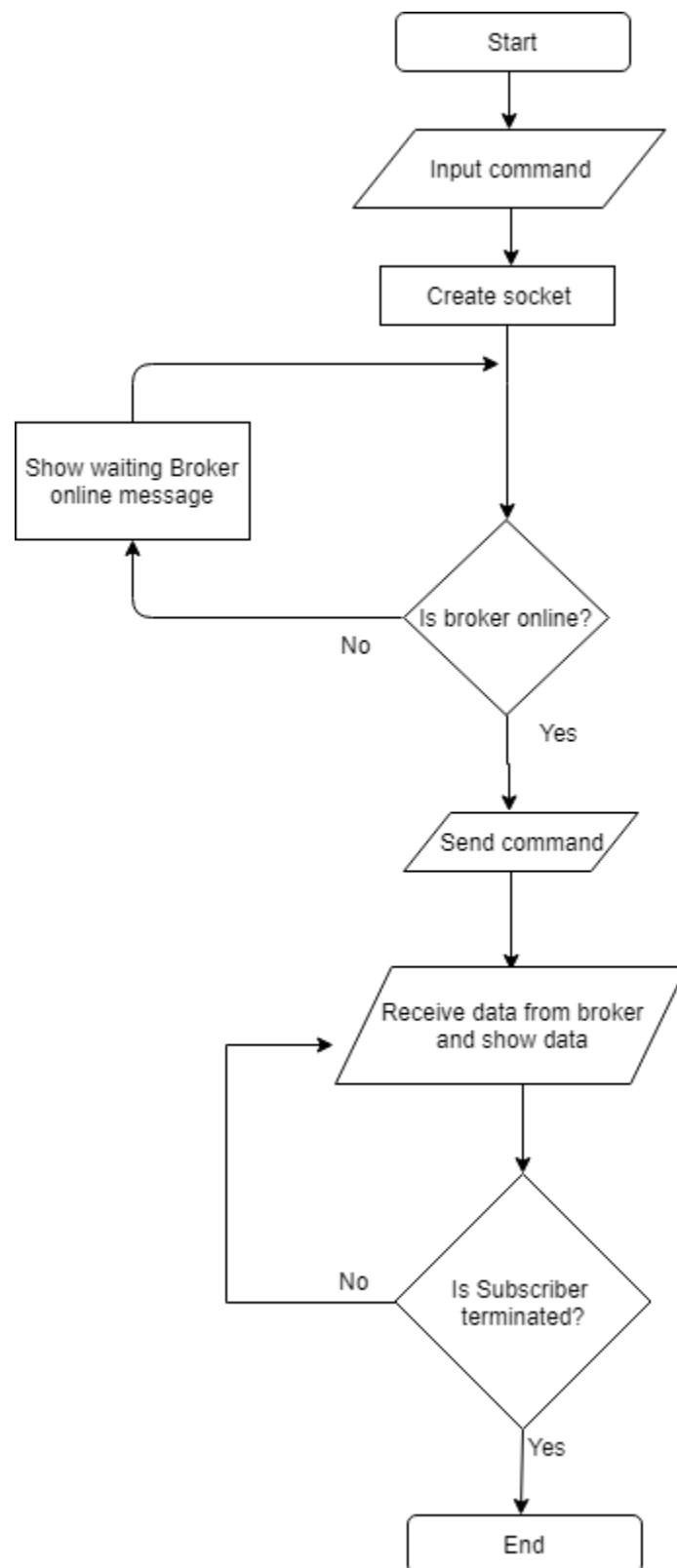
Publisher



## Broker



## Subscriber



## Protocol design

### Message Syntax:

Message ที่ใช้ใน MQTT นี้มีอยู่ 2 รูปแบบ ได้แก่

1. publish                broker\_ip\_address                topic                data

Ex. publish 127.0.0.1 Fruit Lemon

message publish นั้นสามารถใช้ได้แค่ใน Publisher เท่านั้น

2. subscribe                broker\_ip\_address                topic

Ex. subscribe 127.0.0.1 Fruit

message subscribe นั้นสามารถใช้ได้แค่ใน Subscriber เท่านั้น

### Message Sematic:

ความหมายของแต่ละส่วนใน message มีดังนี้

1. publish                broker\_ip\_address                topic                data

publish: command ที่ใช้ในการประกาศ data ใน Topic ที่กำหนด

broker ip address: IP address ของตัวกลางที่ใช้รับ, บันทึกและส่งข้อมูลต่อ

topic: หัวข้อของข้อมูลที่ต้องการจะส่ง

data: ข้อมูลที่ต้องการให้แสดงผลบนหน้าจอของผู้รับ

2. subscribe                broker\_ip\_address                topic

subscribe: command ที่ใช้ในการแสดงตนว่าต้องการรับ data จาก Topic ที่กำหนด

broker ip address: IP address ของตัวกลางที่ใช้รับ, บันทึกและส่งข้อมูลต่อ

topic: หัวข้อของข้อมูลที่ต้องการจะรับจากผู้ส่ง

### Timing:

การส่งข้อมูลนั้นจะแบ่งเป็น 2 ช่วงคือ

1. ช่วงที่ Publisher ส่งข้อมูลให้กับ Broker

โดยจะส่งข้อมูลเมื่อพิมพ์คำสั่ง “ publish broker\_ip\_address topic data ” ได้สำเร็จ

2. ช่วงที่ Broker ส่งข้อมูลให้กับ Subscriber

โดยจะส่งข้อมูลเมื่อ Broker ได้รับข้อมูลจาก Publisher และบันทึกลง dictionary ได้สำเร็จ Broker ก็  
จะทำการส่งข้อมูลให้กับ Subscriber ทันที

ส่วน Subscriber จะรับข้อมูลได้จะต้องทำการพิมพ์คำสั่ง “ subscribe broker\_ip\_address topic ”

ให้สำเร็จ จากนั้น Subscriber ก็จะได้รับข้อมูลทุกครั้งเมื่อ Broker ส่งข้อมูลที่ตรงกับ topic ที่  
subscribe ไว้

ส่วนความเร็วของการส่งข้อมูลนั้นก็ขึ้นอยู่กับความเร็วของการส่งของ local network

### Action:

1. การใช้ MQTT นั้นจะต้องเริ่มจากการเปิด Broker เป็นอันดับแรกบน command line
2. เปิด Publisher ขึ้นมาในอีก command line หนึ่ง จากนั้นก็พิมพ์ command “ publish broker\_ip\_address topic data ” เมื่อทำการพิมพ์เสร็จ Publisher จะส่งข้อมูลไปยัง Broker จาก  
broker\_ip\_address ที่ต้องการ ซึ่งใน MQTT ของเราได้ตั้งค่า broker\_ip\_address ให้เป็น 127.0.0.1  
ตลอด
3. เมื่อ Broker ได้รับข้อมูลแล้ว Broker จะทำการบันทึก Topic และ Data ลงบนตัวแปร Dictionary  
จากนั้นจึงส่งข้อมูลให้กับ Subscriber ที่ทำการ subscribe topic นี้ไว้
4. เปิด Subscriber และพิมพ์ command “ subscribe broker\_ip\_address topic ” เมื่อ command  
รันสำเร็จ, Subscriber จะได้รับข้อมูลเมื่อ Broker ทำการส่งข้อมูลใน topic ที่ทำการ subscribe ไว้

Extension action:

5. หากเปิด Publisher หรือ Subscriber และพิมพ์ command จนสำเร็จ แต่ยังไม่ทำการเปิด Broker ตัวของ program จะยังไม่ทำการส่งข้อมูลให้กับ Broker และจะบอกกับ User ว่ารอการตอบกลับจาก Broker
6. หาก Subscriber subscriber ไปยัง topic ที่ยังไม่ถูกสร้างขึ้น, Subscriber จะไม่แสดงข้อมูลจนกว่า topic จะถูกสร้างขึ้นและส่งข้อมูลมา

## Implementation

การสร้าง MQTT ในครั้งนี้ เราได้ใช้ภาษา Python ในการ Implement เพราะว่าภาษา Python สามารถ import socket programming มาใช้ได้และมีตัวอย่างได้จาก Slide ที่ใช้เรียน ทำให้ง่ายต่อการ Implement

ส่วน Transport protocol ที่ใช้ ทางเราเลือกใช้ Transport Control Protocol หรือ TCP เพราะว่าเราต้องการให้ Broker ส่งข้อมูลไปยัง Subscriber ที่ถูกต้อง

และเนื่องจากทางเราไม่มีอุปกรณ์จำพวก router จึงใช้ PC ของตัวเองเป็นทั้ง Publisher, Subscriber และ Broker จึงใช้ IP Address ของ Broker เป็น 127.0.0.1 ซึ่งเป็น Loopback address สำหรับให้ข้อมูลเข้าและออกใน PC ตัวเอง

ส่วน port ที่ใช้นั้นจะเป็น Ephemeral ports ซึ่งอยู่ในช่วง ID ports ที่ 49,152 - 65,535 ซึ่งสาเหตุที่ใช้เพราะเป็น Private port ส่วนการเลือกใช้จะกำหนดไว้ที่ port ID ที่ 50,000 ทุก Publisher, Subscriber และ Broker

และเพื่อรองรับการเชื่อมต่อจากหลายๆ Client ( Command line ) จึงได้ทำการใช้ import thread เพื่อทำเป็น Thread server

จากหลักการของ MQTT นั้นจะประกอบไปด้วย 3 ส่วน ได้แก่

1. Publisher: เป็นส่วนที่ทำหน้าที่ในการให้ข้อมูล โดยจะส่งให้กับตัวกลางเพื่อให้ตัวกลางส่งให้กับตัวรับอีกที ผ่านคำสั่ง “ publish broker\_ip\_address topic data ” โดยพิมพ์คำสั่งบน command line

เมื่อได้รับคำสั่ง ตัว Program จะทำการเช็คความถูกต้องของคำสั่ง ถ้าไม่แสดงบนหน้าจอว่าผิดปกติอย่างไร หากสำเร็จก็จะทำการแยก String และรวม publish, topic และ data เพื่อส่งให้กับ Broker อีกที ส่วน broker\_ip\_address จะนำไปใช้ในการเชื่อม socket กับ Broker ถ้า Broker ยังไม่ถูกเปิด, Publisher จะแสดงหน้าจอว่ารอการเชื่อมต่อ ถ้าเชื่อมต่อได้สำเร็จจะเพิ่ม String ที่แสดงที่มาของคำสั่งและทำการส่ง topic และ data ให้กับ Broker และวนรอบกลับไปรับ Topic และ Data อีกครั้งจนกว่าจะปิดโปรแกรมหรือทำการปิด Broker

2. Broker: เป็นส่วนตัวกลางทำหน้าที่รับข้อมูลจากผู้ส่ง, บันทึกข้อมูล และส่งให้กับผู้รับทุกตัวที่ต้องการข้อมูลนี้ โดยเมื่อเปิด Broker ก็ทำการเชื่อม socket ด้วย IP 127.0.0.1 โดยทันที เมื่อ Publisher หรือ Subscriber ส่งข้อมูลมายัง Broker, ตัว Broker ก็ทำการสร้าง Thread มารองรับการเชื่อมต่อของแต่ละ Client

จากนั้น Broker ก็จะนำข้อมูลที่ส่งมาไปทำการเช็คว่ามีมาจาก Client ไດระหว่าง Publisher หรือ Subscriber และทำการเช็คความถูกต้องของคำสั่งซึ่งถ้าไม่ถูกต้องก็จะแสดงหน้าจอแสดงว่าผิดปกติอย่างไร ส่วนถ้าไม่ผิดพลาดอะไร Program ก็ทำงานต่อ

หากเป็น Publisher, Broker ก็ทำการเก็บ data ลง dictionary ที่ตรงกับ topic ที่ Publisher ส่งมา และส่งข้อความบอกกับ Publisher ว่า ส่งข้อมูลสำเร็จ



ส่วนถ้าเป็น Subscriber, Broker ก็จะทำการส่งข้อมูลไปให้ Subscriber จาก dictionary และ topic ที่ตัว Subscriber ได้ทำการ subscribe ไว้ และเนื่องจากเราใช้การวน while loop ในการส่งข้อมูลจึงมีการเช็คไม่ให้ส่งข้อมูลซ้ำ แต่ถ้า Publisher ส่งข้อมูลซ้ำใน topic เดิมจะอนุญาตให้ส่งข้อมูลได้

3. Subscriber: เป็นตัวรับข้อมูล โดยจะรับจากตัวกลาง แต่ก่อนจะรับข้อมูลได้นั้น จะต้องทำการส่ง command เพื่อบอกว่าต้องการรับข้อมูลก่อน ผ่านคำสั่ง “ subscribe broker\_ip\_address topic ” โดยพิมพ์คำสั่งบน command line

เมื่อได้รับคำสั่ง ตัว Program จะทำการเช็คความถูกต้องของคำสั่ง ถ้าไม่แสดงบนหน้าจอว่าผิดอย่างไร หากสำเร็จก็จะทำการแยก String และรวม subscribe และ topic เพื่อส่งให้กับ Broker อีกที ส่วน broker\_ip\_address จะนำไปใช้ในการเชื่อม socket กับ Broker ถ้า Broker ยังไม่ถูกเปิด, Subscriber จะแสดงหน้าจอว่ารอการเชื่อมต่อ ถ้าเชื่อมต่อได้สำเร็จจะเพิ่ม String ที่แสดงที่มาของคำสั่งและทำการส่ง topic ให้กับ Broker จากนั้นตัวของ Subscriber ก็จะทำการวนรอรับข้อมูลจาก Broker จนกว่าจะปิดโปรแกรมหรือทำการปิด Broker