Hardware Programming

STM32 IoT Node

1. อ่านค่า Temperature และ Humidity จากเซ็นเซอร์ที่มีบนบอร์ด

2. คำนวณจำนวนคนเข้า และ จำนวนคนออก จากเซ็นเซอร์ที่มีบนบอร์ด

3. ส่งค่า Temperature, Humidity, จำนวนคนเข้า และ จำนวนคนออก ตาม payload format ของ Cayenne

4. ส่งข้อมูลผ่านโครงข่าย LoRa ผ่านทางบอร์ด STM32 LoRa โดยใช้การสื่อสารแบบอนุกรม (Serial Communication)

5. กำหนดให้ใช้ Baud Rate 9600 bps และใช้ UART4 ของบอร์ด STM32 IoT Node

STM32 LoRa

1. กำหนดให้ใช้ Setting ตามที่ใช้ในการแข่งขัน Hardware Day#2

Project: STM32CubeExpansion_LRWAN_V1.1.5\Projects\Multi\Applications\LoRa

Operation: AT_Slave

Activation Mode: OTAA

LoRa : Class A

Account: TGR13_XX

2. ส่งค่าที่ได้รับจากบอร์ด IoT Node ผ่านไปยัง Server ด้วยโครงข่าย LoRa

3. ทำการส่งไปที่ Server ที่ทีม Server ของแต่ละทีมจัดเตรียมไว้เอง จากห้องเรียน Server

4. กำหนดให้ใช้ Baudrate 9600 bps และใช้ UART ตาม Default ของ Project AT_Slave

5. ส่งข้อมูลขึ้นทุก 30 วินาที

หมายเหตุ ให้ต่อไฟเลี้ยงตาม Slide Hardware Day#1

Server Programming

1. ทำการออกแบบ และสร้าง Database เพื่อบันทึกข้อมูลต่างๆที่ได้รับมาจากบอร์ด ST32 LoRa และบันทึกค่าของคนเข้า และออกในบริเวณของ Beacon

ตัวอย่างของฐานข้อมูล

SensorData	
Temperature	Double
Humidity	Double
P-IN	Integer
P-OUT	Interger
Timestamp	Timestamp

BeaconData	
P-IN	Integer
P-OUT	Interger
Timestamp	Timestamp

2. ทำการออกแบบ และสร้าง RESTful API เพื่อใช้ในการส่งค่า ประมวลผล และบันทึกค่าจากช่องทางต่างๆ
3. นำ Dataset ที่แจกให้รวมกับค่าที่ได้จาก Beacon เพื่อทำการส่งค่าให้ ML ไปประมวลผล โดยให้ยึด Format ของ Dataset ที่แจกให้ ตอนส่งค่าให้ ML

Intelligent Monitoring System

LINE

1. ให้สามารถนำค่า Temperature, ค่า Humidity และจำนวนคนเข้าและออกที่ได้จากการคำนวณจากบอร์ด STM32 LoRa แสดงผ่าน Line Messaging เมื่อมีการพิมพ์ “**Admin_Mon**”
2. LINE Beacon ต้องส่งค่าจำนวนคนเข้า และออก ให้กับ Server เพื่อทำการบันทึก หากจำนวนคนเข้ามีมากกว่าจำนวนคนออก 2 คน ให้ BOT ทำการแจ้งว่า “**จำนวนคนเกิน กรุณาเชิญคนออกจากบริเวณ**” โดยอัตโนมัติ

ML

2018-12-29	0	0	0	0	0	77	90	57	88	333	341	445	446	213	773	686	237	58	67	65	0	0	0	0
2018-12-30	0	0	0	0	0	89	78	85	85	250	262	477	604	898	883	956	615	77	57	65	0	0	0	0
2018-12-31	0	0	0	0	0	78	59	64	61	51	340	599	564	1104	963	744	544	55	66	58	0	0	0	0

พระราชวังสนามจันทร์ เปิดให้บริการนักท่องเที่ยวเข้าชมทุกวัน ตั้งแต่เวลา ตี 5 ถึง 1 ทุ่ม โดยมีการบันทึกสรุปจำนวนของผู้เข้าชมทุก 1 ชม ดังตัวอย่างในภาพด้านบน

เมื่อมีสัญญาณ Line Simple Beacon จะมีการบันทึกการเข้าออกแต่ละครั้ง ผ่าน Resful API แบบ POST เช่น https://aaa.foo.com/putSanam โดยส่ง JSON

```
{"beacon":{"datetime":"2012-11-04 14:55:45", "status":"enter"}}
```

และมีการสรุปรวม 1 ชม (เช่น เมื่อพบว่าช่วงตี 5-6 โมงเช้า ของวันที่ 1 ม.ค. 62 มีนักท่องเที่ยวทั้งหมด 75 คน จะมีข้อมูลเพิ่มดังภาพด้านล่าง)

2018-12-29	0	0	0	0	0	77	90	57	88	333	341	445	446	213	773	686	237	58	67	65	0	0	0	0
2018-12-30	0	0	0	0	0	89	78	85	85	250	262	477	604	898	883	956	615	77	57	65	0	0	0	0
2018-12-31	0	0	0	0	0	78	59	64	61	51	340	599	564	1104	963	744	544	55	66	58	0	0	0	0
2019-01-01	0	0	0	0	0	75																		

โดยผู้เข้าแข่งขันจะสามารถดึงข้อมูลจำนวนผู้เข้าชมแต่ละชั่วโมงที่จัดเก็บล่าสุดใน Database ผ่าน Restful API แบบ GET โดยระบุ Parameter เป็น X ชั่วโมงย้อนหลัง เช่น https://aaa.foo.com/getSanam?hours=X ซึ่งจะได้ข้อมูลตอบกลับในรูปแบบ JSON คือ

```
{"number_of_tourist":
["0","0","0","0","78","59","64","61","51","340","599","564","1104","963","744","544","55","66","58","0","0","0","0","0","0","0","0","75"]}
```

***ถ้าข้อมูลที่ request มามากกว่าข้อมูลที่มีในฐานข้อมูลให้ return Error**

ให้ผู้เข้าแข่งขันดึงข้อมูลล่าสุดจาก API จำนวน 8000 ชั่วโมงล่าสุด และนำข้อมูลมา Scaling Data (Normalization) ให้มีค่าตั้งแต่ 0 – 1 และนำไป Train Model เพื่อใช้ทำนายจำนวนนักท่องเที่ยว 3 ชั่วโมงถัดไปจากเวลา ณ ปัจจุบัน ให้ได้ประสิทธิภาพดี โดยมีค่า val_loss ไม่เกิน 0.008

ในการทำนายจำนวนนักท่องเที่ยวให้ผู้เข้าแข่งขัน Load Model จากที่ได้ Train มาใช้ทำนาย และสร้าง Restful API แบบ GET คือ https://aaa.foo.com/predict ซึ่ง API จะส่งค่าจำนวนนักท่องเที่ยงที่ทำนายกลับ เป็น JSON format คือ

```
{"number_of_tourist":["70","100","60"]}
```

โดยสรุป

1.สามารถบันทึกการเข้าออกจากสัญญาณ Beacon แต่ละครั้งผ่าน API
2.สามารถจัดเตรียมข้อมูลเพื่อฝึกสอนได้ (ทำ Scaling Data ให้มีค่าตั้งแต่ 0 – 1)
3.สามารถ Train Model แบบเข้าแบบ Many (หลายชั่วโมง) ออกแบบ 3 ชั่วโมง ได้
4.ได้ค่า val_loss (MSE) ไม่เกิน 0.008
5.สามารถบันทึก Model และนำนาย 3 ชั่วโมงถัดผ่าน API ได้