

Unit 5 IOT core and system architecture



7-Layer IOT world forum reference model

Levels

- 7 Collaboration & Processes**
(Involving People & Business Processes)
- 6 Application**
(Reporting, Analytics, Control)
- 5 Data Abstraction**
(Aggregation & Access)
- 4 Data Accumulation**
(Storage)
- 3 Edge Computing**
(Data Element Analysis & Transformation)
- 2 Connectivity**
(Communication & Processing Units)
- 1 Physical Devices & Controllers**
(The "Things" in IoT)



Data at
Rest



Data in
Motion

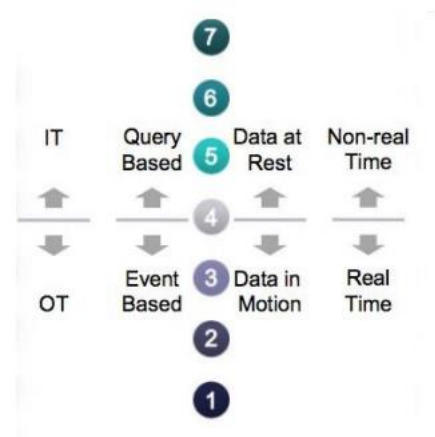


7-Layer IOT world forum reference model



Reference Model คือ แบบจำลองที่อ้างอิงถึงระบบ IOT จะบอกถึงศัพท์ที่ใช้เรียกส่วนประกอบต่างๆ เพื่อให้เห็นการทำงานของระบบไปถึงการไหลของข้อมูลในระบบ ประมวลผล ตักนำเนิดข้อมูล ฮาร์ดแวร์ ซอฟต์แวร์ ส่วนที่เกี่ยวข้องกับผู้ใช้

-สรุป- IOT Reference Model คือ แผนภาพแสดงลักษณะทางกายวิภาค (anatomy) ของระบบ IOT

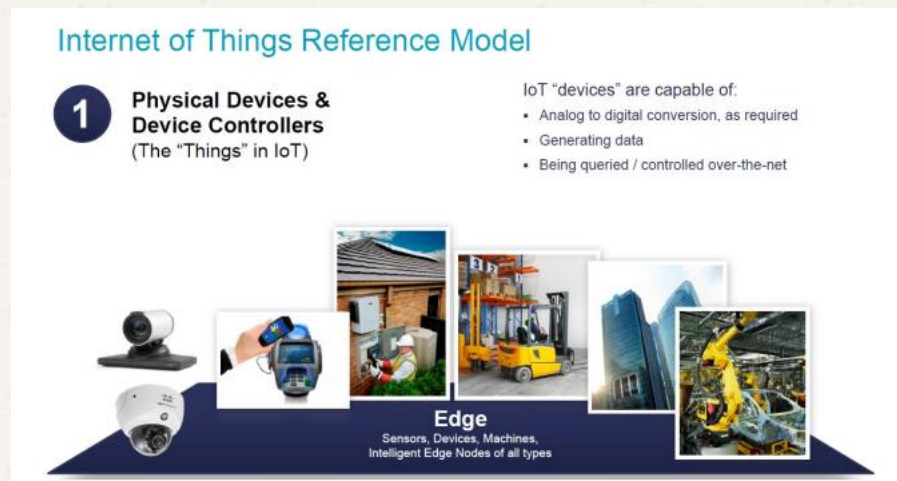


7-Layer IOT world forum reference model

- จากภาพจะเห็นว่า layer ล่างจะเป็นการทำงานของฮาร์ดแวร์นั้นจะอยู่ในลักษณะของการทำงานอยู่ตลอดเวลาด้วยเซ็นเซอร์ต่างๆ เป็นการเก็บค่าตลอดเวลาและจะส่งต่อไปส่วนที่รับข้อมูล level 4 และ 5-7 เป็นส่วนของ application ที่จะถูกร้องขอข้อมูลตามการ request และ query โดยปกติแล้วจะไม่ได้มีการส่งข้อมูลตลอดเวลา
- Layer 1-3 เรียกว่า edge side layer เป็นการทำงานในส่วนฮาร์ดแวร์
- Layer 4-6 เรียกว่า server / cloud side layer เป็นการทำงานในส่วนที่เก็บข้อมูล layer 1-3 จะอยู่บนเน็ตอาจะมี DB หรือไม่ก็ได้รวมไปถึงการวิเคราะห์ข้อมูลก่อนถึงมือผู้ใช้
- Layer 7 เรียกว่า user side layer คือ ส่วนที่ข้อมูลจะถูกนำเสนอต่อผู้ใช้รวมไปถึงการนำข้อมูลไปวิเคราะห์โดยมนุษย์

layer 1 Physical Devices and Controllers

- อุปกรณ์ IOT มีความสามารถแปลง and เป็น digital, สร้างข้อมูล, เรียกค้นข้อมูล, ควบคุมข้ามเครือข่าย
- ไม่มีกฎตายตัวว่าอะไรถือว่าเป็นอุปกรณ์ IOT ไม่ได้ดูที่ขนาด สถานที่ รูปทรง จุดกำเนิด จำนวนชิ้นของอุปกรณ์



layer 2 Connectivity

- การสื่อสารระหว่างอุปกรณ์ layer 1
- การส่งข้อมูลถึงปลายทางผ่านข้ามหลายเครือข่ายอย่างน่าเชื่อถือ
- ดำเนินการ (implement) โพรโตคอลหลายรูปแบบ
- ทำหน้าที่สลับและค้นหาเส้นทาง แปลงระหว่างโพรโตคอล ควบคุมความปลอดภัยระดับเครือข่าย วิเคราะห์การทำงานของเครือข่าย (โดยระบบจะเรียนรู้ด้วยตนเอง)
- การสื่อสารและเชื่อมโยงของ things ถูกดูแลใน layer นี้สำหรับการส่งข้อมูลที่น่าเชื่อถือ ตรงเวลา
- สารสนเทศสามารถ flow ระหว่างอุปกรณ์เครือข่าย level 1 กับเครือข่าย, ข้ามเครือข่าย (traffic), ต่वनออกไปตก, ระหว่างเครือข่ายใน layer 2
- การประมวลผลสารสนเทศระดับต่ำที่เกิดที่ level 3 communication gateway สามารถถูกแนะนำในอุปกรณ์ที่เชื่อมต่ออยู่แล้วจากที่ใช้ IP ไม่ได้บางครั้ง กิจกรรมการคำนวณคล้ายกับการประยุกต์นโยบายความปลอดภัยของเครือข่ายสามารถเกิดขึ้นได้ที่ layer 2

layer 3 Edge Computing



- กรองข้อมูล ทำความสะอาดข้อมูลเป็นกลุ่ม
- คอบตรวจสอบเนื้อหาของแพ็คเกจข้อมูล
- วิเคราะห์ระดับข้อมูลและเครือข่ายผสมผสาน
- การประเมินข้อมูล (ว่าผ่านหรือแจ้งเตือน)
- สร้างเหตุการณ์แจ้งเตือน
- มีความต้องการสำหรับการแปลงข้อมูลแบบจำกัด (localized conversation) ของข้อมูลเครือข่ายที่ไหลภายในสถานการณ์สามารถเพิ่ม optional overhead as well as being as well as being time-consuming ในการส่ง context data จากอุปกรณ์เซนเซอร์ไปยังคลาวด์ที่เป็นศูนย์กลางสำหรับการประมวลผลการทำงานอย่างอื่น
- ในส่วนนี้นำไปสู่แนวความคิดของ edge (fog computing) สำหรับประมวลผลสารสนเทศ as close to edge ของเครือข่ายที่เป็นไปได้ด้วย latency ที่ต่ำสุดจากการจับข้อมูล
- เป็นการประมวลผลแบบกระจายศูนย์ด้วยจุดรวบรวมข้อมูลที่กำหนด
- layer 3 เน้นไปที่การกระทำ computation tasks คล้ายกับการตรวจสอบแพ็คเกจสามารถกระทำที่ระดับนี้
- การประมวลผลสารสนเทศถูกจำกัดและทำพื้นฐานที่ละแพ็คเกจ การประมวลผลระดับสูงกว่าของสารสนเทศคือ layer 4
- Ex. ที่รวมกับ data evaluation ของเกณฑ์ที่กำหนด การจัดรูปแบบใหม่ของข้อมูลสำหรับการประมวลผลข้อมูลในระดับที่สูงขึ้น ขยายแ เข้ารหัส ข้อมูล การกลั่น ลดขนาดของข้อมูล
- การประเมินข้อมูลสำหรับเกณฑ์การผ่านของการสำเร็จหรือการสร้างการเตือนและอื่นๆ

layer 4 Data Accumulation

- แปลงข้อมูลที่มีการรับส่ง (data-in-motion) ไปยังข้อมูลที่ถูกจัดเก็บไว้ในฐานข้อมูล (data-at-rest)
- แปลงข้อมูลจากแฟ้มเกจเครือข่ายไปยังฐานข้อมูลเชิงสัมพันธ์
- แปลงข้อมูลจากข้อมูล event-based data ให้เป็นข้อมูล query-based data
- ลดจำนวนข้อมูลโดยผ่านการกรองและการเลือกข้อมูลเก็บ

layer 5 Data Abstraction

- สร้างเค้าโครงสร้างและมุมมองของข้อมูลในลักษณะที่โปรแกรมประยุกต์ต้องการ
- ผสานข้อมูลจากแหล่งต่างๆ และลดความซ้ำซ้อน
- กรอง เลือก แสดงผล เปลี่ยนรูปแบบของข้อมูลสำหรับโปรแกรมของลูกค้า
- ปรับความแตกต่างในข้อมูล รูปร่าง รูปแบบ โครงร่าง โปรโตคอลเข้าถึงและความปลอดภัย



layer 6 Application

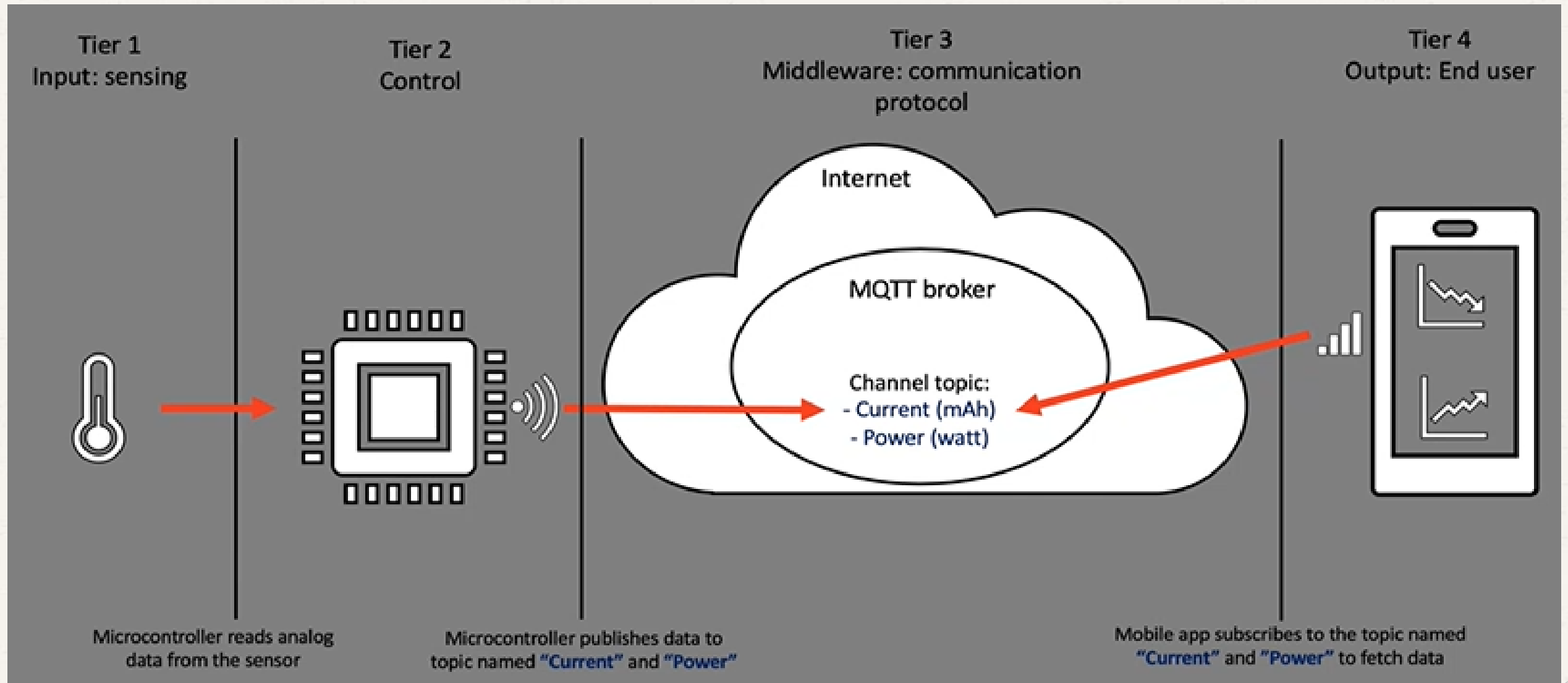
- จะเป็นส่วนของ business intelligence ทำหน้าที่วิเคราะห์ข้อมูลที่ได้เพื่อประโยชน์ทางการดำเนินด้านธุรกิจไม่ว่าจะเป็นในส่วนขององค์กรเอกชน รัฐบาลโดยปกติจะทำการวิเคราะห์และรายงานผลข้อมูล



layer 7 Collaboration and Process

- เกี่ยวข้องกับคนและกระบวนการทางธุรกิจ
- ทำให้คนสามารถสื่อสารและทำงานร่วมกันเพื่อทำ IOT ให้เป็นประโยชน์

4-tier IOT system architecture



Sensing (input) > Control > Cloud > User (Output)

- จากภาพการทำงานจะมี 4 ขั้นตอน คือ เริ่มต้นจากการที่เซ็นเซอร์ตรวจวัดค่าที่ได้จากนั้นส่งต่อไปให้กับหน่วย control หนึ่งระบบอาจมีเซ็นเซอร์หลายตัว/ประเภทแต่มีตัว control แค่ 1 ตัวทำหน้าที่รับค่าจากเซ็นเซอร์ทุกตัวในระบบและตัว control ยังทำหน้าที่ในการส่งค่าไปยังผู้ใช้ผ่านระบบอินเทอร์เน็ต (cloud) โดยอาจจะมีการเก็บค่าลงในระบบฐานข้อมูลหรือจะแสดงค่าอย่างเดี๋ยวกว่าข้อมูลจะถูกส่งต่อไปยังแอปพลิเคชันบนอุปกรณ์พกพา เช่น โทรศัพท์ แท็บเล็ต เสร็จสิ้นกระบวนการส่งข้อมูลจากเซ็นเซอร์ไปยังผู้ใช้

Tier 1 : Sensing

- คือ การรับค่าสภาพแวดล้อมต่างๆ ตามวัตถุประสงค์ของระบบ เช่น ตรวจวัดอุณหภูมิในอากาศ อัตราการเต้นของหัวใจและอื่นๆ รวมไปถึงสวิทช์เปิดปิดไฟอุปกรณ์พวกนี้จะเรียกว่า เซ็นเซอร์ sensor
- โดยปกติแล้วระบบ IOT เป็นการนำอาระบบคอมพิวเตอร์เข้ามาทำงานแทนมนุษย์ เช่น ระบบจับฝนตกซึ่งเซ็นเซอร์บางประเภทสามารถเปรียบเทียบกับกับประสาทสัมผัสของมนุษย์จะเห็นได้ว่าการพัฒนาเซ็นเซอร์ที่มีการทำงานในลักษณะเดียวกับประสาทสัมผัสของมนุษย์
 1. รูป : เซ็นเซอร์รับภาพ (กล้อง), เซ็นเซอร์ตรวจจับความร้อน เคลื่อนไหว
 2. รส : เซ็นเซอร์ตรวจจับสารเคมี
 3. กลิ่น : เซ็นเซอร์ตรวจจับแก๊ส ควัน
 4. เสียง : เซ็นเซอร์รับเสียง (ไมโครโฟน)
 5. สัมผัส : เซ็นเซอร์ตรวจวัดแรงสั่นสะเทือน
- เซ็นเซอร์บางประเภทมีการพัฒนาให้สามารถตรวจวัดค่าได้เกินขอบเขตที่มนุษย์จะสัมผัสได้ เช่น เสียงที่มนุษย์ไม่ได้ยิน แก๊สที่ไม่มีกลิ่น

Tier 2 : Control

- มีหน้าที่ในการประมวลผลทำหน้าที่คล้าย CPU ในคอมพิวเตอร์
- ระบบ IOT 1 ระบบอาจมีการใช้งานเซ็นเซอร์มากกว่า 1 ตัว โดยเซ็นเซอร์จะส่งค่าที่ได้มายังตัวควบคุม หลังจากนั้นตัวควบคุมจะทำการส่งค่าไปยังผู้ใช้ผ่านอินเทอร์เน็ตโดยอุปกรณ์ที่ใช้ในตัวควบคุมมักจะใช้อุปกรณ์ที่เรียกว่า Microcontroller
- Microcontroller จะมีการใช้งานโมดูลสื่อสารไร้สาย เช่น Wi-Fi, Bluetooth, 3G, 4G



โมดูล 3G สำหรับเชื่อมต่อกับ microcontroller เพื่อเชื่อมต่ออินเทอร์เน็ตผ่านเครือข่ายโทรศัพท์ด้วยการใส่ซิมการ์ด

microcontroller ที่มีชิพ Wi-Fi ในตัวสามารถส่งข้อมูลผ่านการเชื่อมต่อ Wi-Fi



เสาอากาศ Wi-Fi



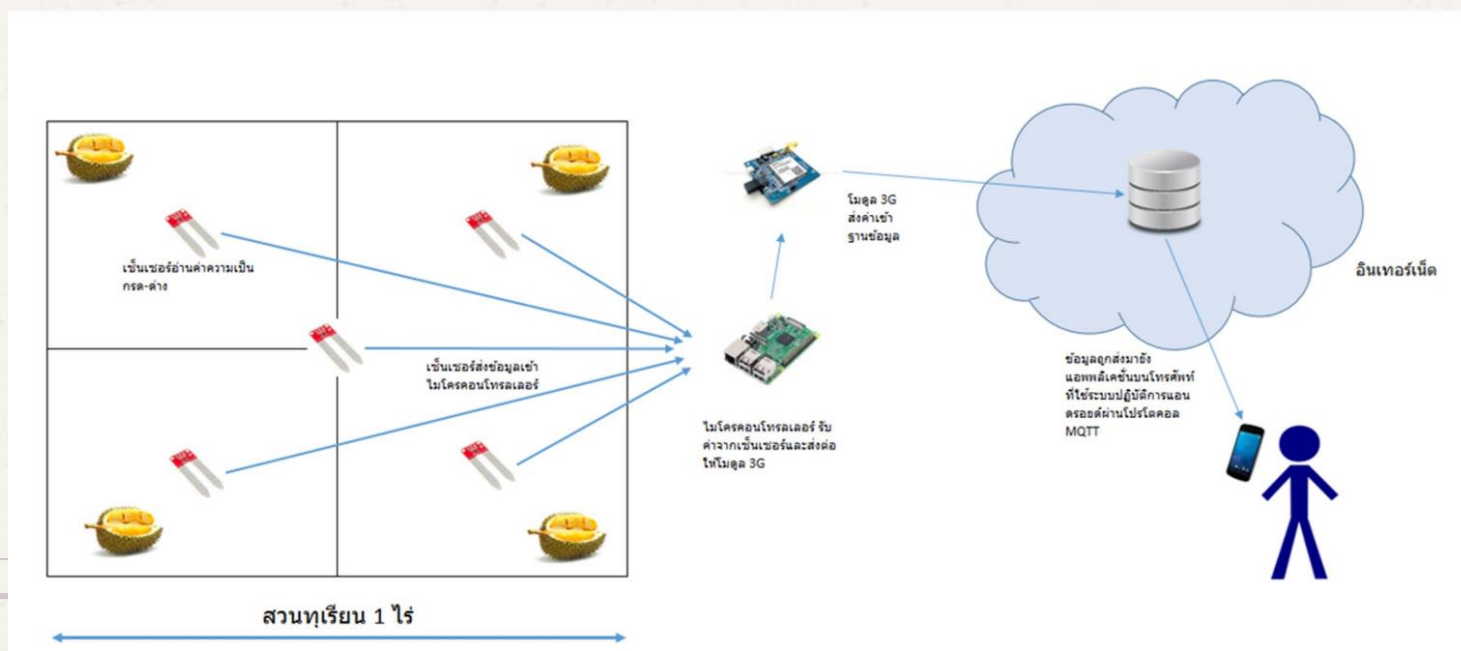
Tier 3 : Cloud

- หรือระบบอินเทอร์เน็ตนี้คือส่วนสำคัญที่ทำให้ระบบเป็น IOT เนื่องจากเป็นสื่อกลางในการส่งข้อมูลระหว่างระบบกับผู้ใช้จะมีการพูดถึงวิธีการส่งข้อมูลต่างๆ ใน layer 4

Tier 4 : User

- คือ การแสดงค่าที่รับได้จาเซ็นเซอร์ไปยังผู้ใช้งานซึ่งเป็นมนุษย์ ปัจจุบันมีค่าแสดงค่าผ่านแอปพลิเคชันบน โทรศัพท์ ผ่านเว็บ หรือโปรแกรม โดยสามารถเข้าถึงได้จาก PC, Laptop โดยจะมีการแสดงค่าหลากหลาย เช่น ตัวเลข กราฟ แผนภูมิหรือการตีความค่าที่ได้จากตัวเลขเป็นข้อความ เช่น หากอุณหภูมิมีค่ามากกว่า 36C ให้เป็นสีแดงหรือ”อากาศร้อนมาก” และอุณหภูมิต่ำกว่า 25C ให้เป็นสีฟ้าหรือ “อากาศเริ่มหนาว”
- มุมมองและสามารถพัฒนาระบบ IOT ให้มีประสิทธิภาพ
 1. โปรโตคอลสื่อสาร IOT
 2. Edge computing
 3. Cloud computing
 4. UX/UI design
 5. Network Security

- จากภาพจะเห็นว่ามีการพัฒนาในส่วนของฮาร์ดแวร์ที่เป็นเซ็นเซอร์ตรวจวัดสภาพดิน ได้แก่ ค่าความชื้น (moisture) ค่าความเป็นกรด/ด่างของดิน (pH) ค่าอุณหภูมิในอากาศ (temperature) โดยเซ็นเซอร์จะส่งค่าไปยังอุปกรณ์ควบคุมเรียกว่า microcontroller ทำหน้าที่รับค่าจากเซ็นเซอร์และทำการประมวลผลข้อมูลก่อนจะส่งข้อมูลไปยังระบบ cloud ซึ่ง microcontroller อีกหนึ่งหน้าที่คือการเชื่อมต่อระบบเครือข่ายหรือ internet เพื่อส่งข้อมูล
- ระบบ cloud จะเป็นการพัฒนาในส่วนซอฟต์แวร์ในการเก็บข้อมูลและเตรียมข้อมูลเพื่อที่จะนำไปแสดงยังหน้าจอผู้ใช้ผ่านแอปพลิเคชันบนโทรศัพท์มือถือหรือเว็บไซต์
- ดังนั้นการพัฒนาระบบ IOT ระบบหนึ่งประกอบไปด้วยการพัฒนาทั้งในส่วนของฮาร์ดแวร์ ซอฟต์แวร์

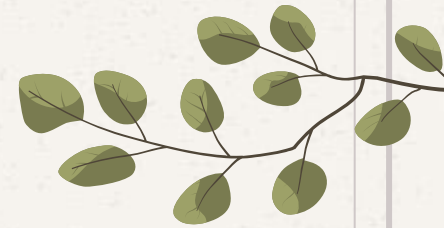


สถาปัตยกรรมของระบบ IOT

- คือ โครงสร้างของระบบ IOT ที่มีการพัฒนาและใช้งานกันในปัจจุบัน โดยเป็นการนำเอาอุปกรณ์อาร์ดแวร์ ได้แก่ สิ่งของ เครื่องใช้ต่างๆ และเซ็นเซอร์เข้ามาทำงานกับระบบคอมพิวเตอร์และอินเทอร์เน็ตและมีการติดต่อกับผู้ใช้ผ่านทางโทรศัพท์มือถือซึ่ง การที่จะสร้างระบบที่มีความสามารถดังกล่าวนั้นจะต้องใช้เทคโนโลยีที่มีอยู่ในปัจจุบันซึ่งมีตัวเลือกจำกัด ดังนั้นนักพัฒนาทั่วโลกจึง มีการสร้างระบบที่มีโครงสร้างที่มีความคล้ายกัน เนื่องจากใช้เทคโนโลยีและพื้นฐานความรู้เดียวกัน

RTOS : Real-Time Operating System

- การใช้งาน RTOS จะทำให้ระบบสามารถประมวลผล process ได้มากกว่า 1 process ได้ในเวลาอีกทั้งยังมีการการันตีการประมวลผลที่จะเกิดขึ้น หากเกิดการ interrupt หรือการร่นตีการประมวลผลเมื่อถึงกำหนดเวลา
- RTOS = เมื่อมีคนเคาะประตู (interrupt) เรามักที่จะหยุดทำสิ่งที่กำลังทำอยู่และไปเปิดประตูทันที (ตอบสนองทันทีและการร่นตีการประมวลผลภายในระยะเวลา เช่น 5 s)
- Non-RTOS = เมื่อมีคนเคาะประตู (interrupt) เราไม่จำเป็นต้องตอบสนองทันทีปล่อยให้เชารอไปก่อนและปล่อยให้เคาะไปสักระยะเวลาหนึ่ง (delay) หรือตลอดไป (crashed)
- ในระบบขับเคลื่อนอัตโนมัติ (autonomous car) ถ้ารถไม่ตอบสนองเมื่อมีคนวิ่งตัดหน้ากระแทกนั้นอาจทำให้เกิดอุบัติเหตุได้ รถยนต์ต้องสามารถที่จะตอบสนอง (เบรก) ทันทีเมื่อตรวจพบว่ามีคนตัดหน้ารถ
- ยิ่งระบบปฏิบัติการจัดการงานมากเท่าไร ระบบก็จะมีประสิทธิภาพและประสิทธิภาพน้อยลงเท่านั้นเนื่องจากแต่ละงานต้องใช้ทรัพยากรดำเนินการ
- Non-RTOS ในบางครั้งถือว่าเป็น RTOS แบบซอฟต์แวร์เพราะไม่ได้รับประกันระยะเวลาที่จะตอบสนองต่อการขัดจังหวะ แต่จะพยายามตอบสนองให้ได้มากที่สุดเท่าที่จะเป็นไปได้



RTOS เหมาะสำหรับระบบ เช่น ระบบควบคุมไฟฟ้า ระบบความแม่นยำหรือเครื่องจักร ระบบการรักษาพยาบาล เครื่องบินและระบบควบคุมการจราจรทางอากาศ

GPOS : General-Purpose Operating System	RTOS
ส่วนใหญ่ใช้ในพีซีและแล็ปท็อป	เหมาะสมกับระบบสมองกลฝังตัว
การตั้งเวลาตามกระบวนการ	การตั้งเวลาตามเวลา
เวลาที่ถูกขัดจังหวะไม่ถือว่ามีความสำคัญ	ไม่สามารถยอมรับเวลาในการขัดจังหวะได้
ต้องการหน่วยความจำมากขึ้นเพื่อรองรับการทำงานหลายอย่างพร้อมกัน	งานน้อยและต้องการหน่วยความจำน้อย (ทรัพยากรน้อยลง)
เวลาตอบกลับไม่สามารถคาดเดาได้และไม่รับประกัน	เวลาตอบสนองสามารถคาดเดาได้และรับประกันได้

ซ้าย : Non-Multitasking

```
1 void setup() {
2
3   pinMode(12, OUTPUT);
4   pinMode(13, OUTPUT);
5 }
6
7
8 void loop() {
9
10  digitalWrite(12, HIGH);
11  delay(200);
12  digitalWrite(12, LOW);
13  delay(200);
14
15  digitalWrite(13, HIGH);
16  delay(500);
17  digitalWrite(13, LOW);
18  delay(500);
19 }
```

ขวา : Multitasking

```
1 #include <Arduino_FreeRTOS.h>
2
3 void TaskBlink1( void *pvParameters );
4 void TaskBlink2( void *pvParameters );
5 void Taskprint( void *pvParameters );
6
7 void setup() {
8   // initialize serial communication at 9600 bits per second:
9   Serial.begin(115200);
10  xTaskCreate(led_1_task, "led1", 128, NULL, 1, NULL);
11  xTaskCreate(led_2_task, "led2", 128, NULL, 1, NULL);
12
13  vTaskStartScheduler();
14 }
15
16 void loop()
17 {
18 }
19
20 void led_1_task(void *pvParameters) {
21   pinMode(8, OUTPUT);
22   while(1)
23   {
24     Serial.println("Task1");
25     digitalWrite(8, HIGH);
26     vTaskDelay( 200 / portTICK_PERIOD_MS );
27     digitalWrite(8, LOW);
28     vTaskDelay( 200 / portTICK_PERIOD_MS );
29   }
30 }
31 void led_2_task(void *pvParameters)
32 {
33   pinMode(7, OUTPUT);
34   while(1)
35   {
36     Serial.println("Task2");
37     digitalWrite(7, HIGH);
38     vTaskDelay( 300 / portTICK_PERIOD_MS );
39     digitalWrite(7, LOW);
40     vTaskDelay( 300 / portTICK_PERIOD_MS );
41   }
42 }
```

Arduino programming commands

Basic	Basic & Core
Structure: setup(), loop()	Condition: IF-ELSE
pinMode, INPUT, OUTPUT, LED_BUILTIN	Loop: for, while
Data types: int, float, char, boolean	Library management
Digital: digitalWrite(), digitalRead()	Time: delay(), millis()
Analog: analogRead(), analogWrite(), analogReference	Real-time OS, interrupts()

Connect to the Internet

- แนวคิดของระบบอัจฉริยะ เช่น บ้านอัจฉริยะ สำนักงานอัจฉริยะ เป็นที่นิยมมากขึ้น
- มีระบบฝังตัวเชิงพาณิชย์ที่สามารถเชื่อมต่อกับอุปกรณ์พกพาได้มากขึ้น เราเรียกมันว่า Internet of Things เพื่อให้สามารถสร้างระบบประเภทนี้ได้ จำเป็นต้องได้รับความรู้บางอย่างเกี่ยวกับระบบเครือข่าย, DB, cloud computing