

## แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต

### IoT Approaches to Manufacturing System

4/4 - แนะนำ IoTs Platform และ Ubidots IoTs Platform

- Industrial Internet of Things – IIoT
- Internet of Bodies – IoB
- IoT Platform TOP 10 ในปี 2021
- การโปรแกรมใช้งาน Ubidots เพื่อแสดงค่าและควบคุมอุปกรณ์
- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

### 1/5 Industrial Internet of Things – IIoT

<https://www.trendmicro.com/vinfo/us/security/definition/industrial-internet-of-things-iiot>

<https://www.sumipol.com/knowledge/transformation-iiot/>

<https://th.cc-link.org/th/cclink/article/5-things-that-misunderstanding-about-iiot.html>

<https://www.marketingoops.com/digital-life/6-technology-iiot-2017/>



#### 1.1. What is the industrial internet of things (IIoT)?

The industrial internet of things (IIoT) refers to the extension and use of the [internet of things](#) (IoT) in industrial sectors and applications. With a strong focus on machine-to-machine (M2M) communication, [big data](#), and [machine learning](#), the IIoT enables industries and enterprises to have better efficiency and reliability in their operations. The IIoT encompasses industrial applications, including robotics, medical devices, and software-defined production processes.

The IIoT goes beyond the normal consumer devices and internetworking of physical devices usually associated with the IoT. What makes it distinct is the intersection of information technology (IT) and operational technology (OT). OT refers to the networking of operational processes and [industrial control systems](#) (ICSs), including human machine interfaces (HMIs), supervisory control and data acquisition (SCADA) systems, distributed control systems (DCSs), and programmable logic controllers (PLCs).

The convergence of IT and OT provides industries with greater system integration in terms of automation and optimization, as well as better visibility of the supply chain and logistics. The monitoring and control of physical infrastructures in industrial operations, such as in agriculture, healthcare,

manufacturing, transportation, and utilities, are made easier through the use of smart sensors and actuators as well as remote access and control.

In the context of the fourth industrial revolution, dubbed [Industry 4.0](#), the IIoT is integral to how cyber-physical systems and production processes are set to transform with the help of big data and analytics. Real-time data from sensors and other information sources helps industrial devices and infrastructures in their “decision-making,” in coming up with insights and specific actions. Machines are further enabled to take on and automate tasks that previous industrial revolutions could not handle. In a broader context, the IIoT is crucial to use cases related to connected ecosystems or environments, such as how cities become [smart cities](#) and factories become smart factories.

## 1.2. จาก IoT สู่อIIoT เมื่อโลกดิจิทัลเข้าเปลี่ยนแปลงอุตสาหกรรม

แนวคิดของอุตสาหกรรม 4.0 มีอิทธิพลในการสร้างแรงกระตุ้นให้เกิดการเปลี่ยนแปลงด้านนวัตกรรมเพื่อเข้าสู่ยุคแห่งการปฏิวัติอุตสาหกรรมแห่งอนาคต ซึ่งเป็นการบูรณาการการผลิตเข้ากับการเชื่อมต่อเครือข่ายอินเทอร์เน็ตในรูปแบบ “The Internet of Things (IoT)” ทำให้กระบวนการผลิตตลอดทั้งซัพพลายเชนเชื่อมต่อกันบนโลกดิจิทัลได้อย่างสมบูรณ์ยิ่งขึ้น และในปัจจุบันก็มีการปรับใช้ในอุตสาหกรรมทั้งขนาดใหญ่และขนาดย่อมในชื่อของ Industrial Internet of Things

INDUSTRIAL INTERNET OF THINGS คืออะไร

Industrial Internet of Things หรือ IIoT คือ การนำเครื่องจักร ระบบการวิเคราะห์ขั้นสูง และคนมาทำงานร่วมกันผ่านโครงข่ายของอุปกรณ์ที่เชื่อมต่อกันด้วยเทคโนโลยีการสื่อสาร ส่งผลให้เกิดระบบที่สามารถติดตาม เก็บข้อมูล แลกเปลี่ยนและแสดงผลข้อมูลเชิงลึกที่เป็นประโยชน์ อย่างที่ไม่เคยทำได้มาก่อน ข้อมูลดังกล่าวสามารถช่วยให้ภาคอุตสาหกรรมสามารถตัดสินใจทางธุรกิจได้อย่างชาญฉลาดและรวดเร็วยิ่งขึ้น



### 2.1 ประโยชน์ของ IIoT

IIoT ช่วยปรับปรุงการเชื่อมต่อ เพื่อเพิ่มประสิทธิภาพการสื่อสาร ยืดหยุ่น ประหยัดเวลา และประหยัดค่าใช้จ่ายในการวางแผนการทำงานในโรงงานอุตสาหกรรม บริษัทที่มีความพร้อม เริ่มได้รับประโยชน์จาก IIoT ในการลดต้นทุนการผลิตจากการบำรุงรักษาที่คาดการณ์ล่วงหน้า (Predictive maintenance) การตรวจสอบสถานะของเครื่องจักร (Monitor) และหลีกเลี่ยงการ Downtime ของระบบเพิ่มความปลอดภัยและช่วยเพิ่มประสิทธิภาพในการปฏิบัติงานในส่วนอื่นได้ดียิ่งขึ้น

ระบบเครือข่าย IIoT สามารถเชื่อมต่อข้อมูลขนาดใหญ่ของแต่ละหน่วยงาน ตั้งแต่ไลน์การผลิต ไปจนถึงระดับออฟฟิศ และทุกคนในองค์กรอย่างเป็นระบบ เพื่อที่จะทำให้ผู้บริหารสามารถใช้ข้อมูลจาก IIoT ได้อย่างครบถ้วนและมีประสิทธิภาพซึ่งจะ

เป็นตัวแปรสำคัญนำไปสู่การตัดสินใจในอนาคต การขยายตัวของ IoT สำหรับภาคอุตสาหกรรมและโรงงานการผลิต (Industrial Internet of Thing – IIoT) ทั่วโลกมีแนวโน้มการเติบโตสูงขึ้นอย่างต่อเนื่อง

## 2.2 การเติบโตของตลาด IIoT ระดับสากล

ผลจากการศึกษาวิจัยข้อมูลทางการตลาดโดย Market and Market พบว่าในปี 2015 ตลาด IIoT มีมูลค่าตลาดโดยรวมสูงถึง 113 พันล้าน เหรียญสหรัฐ ซึ่งคาดว่าจะภายในปี 2022 จะมีมูลค่ากว่า 195 พันล้านเหรียญสหรัฐ อัตราการเติบโตเฉลี่ยต่อปี 7.9% การเติบโตของ IIoT มีผลมาจากหลักการโรงงานอัจฉริยะ (Smart Factory) และเทคโนโลยีการผลิตแบบอัตโนมัติขั้นสูงโดยการริเริ่มของภาครัฐ

ตัวอย่างเช่น อุตสาหกรรม 4.0 ของประเทศเยอรมันและฝรั่งเศสที่มีการส่งเสริมการใช้โซลูชัน IIoT ในยุโรป เช่นเดียวกันกับประเทศผู้นำอย่างเช่น สหรัฐอเมริกา ในขณะที่ประเทศจีนยังถือครองตลาดทางด้าน IIoT ที่ใหญ่ที่สุดในกลุ่มเอเชียแปซิฟิก ในทิศทางเดียวกันตลาดในอินเดียคาดว่าจะมีการเติบโตสูงขึ้น ทั้งนี้ปัจจัยที่ส่งผลต่อการขยายตลาด IIoT ได้แก่ ความก้าวหน้าด้านเทคโนโลยีในอุตสาหกรรมเมคคานิคัลและอิเล็กทรอนิกส์ และวิวัฒนาการด้านเทคโนโลยีคลาวด์ที่เติบโตอย่างรวดเร็ว

## 2.3 ส่วนประกอบสำคัญของ IIoT

การนำระบบ Internet of Thing (IoT) มาใช้ในอุตสาหกรรมการผลิตนั้นจำเป็นต้องพัฒนาองค์ประกอบหลายประการ อาทิ การปรับปรุงเครื่องจักรกลและอุปกรณ์แบบเดิมให้รองรับเทคโนโลยี IIoT การเพิ่มอุปกรณ์ตรวจวัดและการเชื่อมต่อระบบการทำงานเข้าด้วยกัน รวมถึงระบบรักษาความปลอดภัยของข้อมูล การควบคุมและจัดการข้อมูลที่ได้รับทั้งโครงข่ายข้อมูล

การใช้สมาร์ตเซนเซอร์ในการเชื่อมต่อสื่อสารถึงกันระหว่างเครื่องจักรกล (Machine-to-Machine M2M Communication) ซึ่งวิธีนี้จะทำให้เครื่องจักรกลสามารถประเมินสถานะการทำงานที่ดีที่สุดและจดจำข้อมูล ตลอดจนการลำดับขั้นตอนการผลิตได้ด้วยตัวเองและการรวบรวมข้อมูลขนาดใหญ่ที่ได้รับ (Big Data) ด้วยระบบการประมวลผลการผลิตแบบเรียลไทม์ (Manufacturing Execution Systems – MES) เข้ามาควบคุมติดตามและบันทึกผลการผลิตผ่านอุปกรณ์ เช่น สมาร์ทโฟน แท็บเล็ต คอมพิวเตอร์ ซึ่งในปัจจุบันแพลตฟอร์มการทำงานของระบบ IIoT ประกอบด้วย 3 เทคโนโลยีหลัก ได้แก่



### 2.3.1 Sensors และ Sensor-Driven Computing

เซนเซอร์ถือเป็นคือด่านแรกในการเก็บข้อมูลที่ได้จากการผลิตและส่งไปยังส่วนของ Processor ซึ่งตัวเซนเซอร์ทำให้อุปกรณ์สามารถรับรู้สถานะต่าง ๆ เช่น อุณหภูมิ ความดัน แรงดันไฟฟ้า การเคลื่อนไหว และด้านเคมี Sensor-Driven Computing จะแปลงการรับรู้นี้เป็นข้อมูลเชิงลึก (Insights) โดยใช้ Industrial Analytics ในลำดับถัดไปที่ผู้ปฏิบัติงานและระบบสามารถนำข้อมูลไปวิเคราะห์ต่อได้

### 2.3.2 Processor หรือ Industrial Analytics

โปรเซสเซอร์เป็นตัวประมวลผลข้อมูลที่ได้รับจากเซนเซอร์ ให้อยู่ในรูปที่สามารถนำไปใช้งานได้ตามวัตถุประสงค์เพื่อเก็บข้อมูลหลาย ๆ ส่วนของเครื่องจักรในคราวเดียวกัน นอกจากนี้ยังทำหน้าที่ส่งคำสั่งไปยังเซนเซอร์ เป็นเสมือนเครือข่ายในการเชื่อมโยงข้อมูลทุกจุดให้สามารถทำงานได้ทันที (Real time)

### 2.3.3 Intelligent Machine Application

ในอนาคตอันใกล้ผู้ผลิตเครื่องจักรจะไม่ผลิตเพียงแค่เครื่องจักรที่มีเฉพาะระบบกลไกเท่านั้น แต่จะรวมฟังก์ชันที่มีสมอง (Intelligence) อีกด้วย เพื่อควบคุมการผลิตอัตโนมัติผ่านซอฟต์แวร์ ซึ่งปกติผู้วางระบบจะทำบนระบบคลาวด์ทั้งหมดเพื่อให้ผู้ใช้งานสามารถควบคุม แก้ไข จัดการตรวจสอบการทำงานผ่านแอปพลิเคชันที่ได้รับการพัฒนาขึ้น และสามารถนำข้อมูลไปวิเคราะห์ประมวลผลเพื่อเพิ่มประสิทธิภาพการผลิตต่อไป แอปพลิเคชันที่มาพร้อมกับเครื่องจะเป็นเครื่องมือสำหรับสร้างรายได้ใหม่ในรูปแบบผสมระหว่างผลิตภัณฑ์และบริการและความก้าวหน้าทางเทคโนโลยีทำให้ง่ายต่อการผสมรวมอุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์เข้าด้วยกัน

## **1.3. อนาคตและความท้าทายของ IIoT**

อย่างไรก็ตาม การพัฒนาระบบ Industrial IoT ซึ่งต้องอาศัยการเชื่อมต่ออุปกรณ์และเครื่องจักรต่างๆ ร่วมกันเป็นหนึ่งเดียวอาจทำได้ยากและต้องใช้เวลา เนื่องจากในเครื่องจักรแต่ละเครื่องมีรูปแบบและระบบโครงสร้างที่แตกต่างกัน ดังนั้นจึงถือเป็นความท้าทายของผู้พัฒนาระบบการเชื่อมต่อทั้งหมดให้สามารถทำงานได้อย่างสมบูรณ์ และคำนึงถึงการเก็บรักษาความปลอดภัยของข้อมูล IIoT กำลังกลายเป็นกระแสการพัฒนาที่สำคัญส่งผลต่อความสำเร็จในธุรกิจอุตสาหกรรมทั้งในปัจจุบันและอนาคต ภาคธุรกิจต่างพยายามผลักดันการพัฒนาระบบและอุปกรณ์ที่ทันสมัยเพื่อตอบสนองให้ทันต่อความต้องการตลาดที่รวดเร็วผันผวนและเผชิญกับเทคโนโลยีที่เปลี่ยนแปลงอย่างฉับพลัน (Disruptive Technology) อุตสาหกรรมที่นำ IIoT มาใช้งานสามารถปรับปรุงและพัฒนากระบวนการผลิต ความปลอดภัยเพิ่มประสิทธิภาพ และให้ผลกำไรที่ดีกว่าในระยะยาว

## **1.4. คนมักเข้าใจผิดเกี่ยวกับ Industrial Internet of Things (IIoT) ใน 5 ประเด็น**

IIoT เป็นสิ่งใหม่ที่เกิดจาก Digital Transformation แม้ว่าบางองค์กรไม่ได้ตระหนักถึงความสามารถที่แท้จริงของ IIoT และยังไม่แน่ใจว่าจะสามารถใช้ประโยชน์จากการใช้ IIoT ได้อย่างไร แต่ด้วยกระแสของ Digital Transformation ที่กำลังเกิดขึ้นทำให้เทคโนโลยีเข้ามามีบทบาทสำคัญในทุกอุตสาหกรรมและทุกธุรกิจ โดยเฉพาะด้านการผลิต ทำให้ข้อมูลเกี่ยวกับการผลิตแต่ละขั้นตอนสามารถเข้าถึงเพื่อใช้งานและสามารถแบ่งปันข้อมูลได้อย่างปลอดภัยจากทุกที่และทุกอุปกรณ์ โดยมี IIoT เป็นส่วนสำคัญ อย่างไรก็ตามหลายท่านอาจมีความเข้าใจผิดเกี่ยวกับ IIoT อยู่มากมายที่ทำให้เกิดความวิตกกังวลเกี่ยวกับการนำเอา IIoT มาใช้ในการดำเนิน งานด้านการผลิต ในบทความนี้จะนำเสนอเกี่ยวกับ 5 สิ่งที่คุณส่วนใหญ่เข้าใจผิดเกี่ยวกับ IIoT พร้อมข้อเท็จจริง



### ความเข้าใจผิดที่ 1 : IIoT ไม่เหมาะกับธุรกิจขนาดกลางและขนาดย่อม (SMEs)

ความเป็นจริง : IIoT เหมาะกับองค์กรทุกขนาด

หลายองค์กรคิดว่า IIoT นั้นเหมาะกับองค์กรขนาดใหญ่เท่านั้น แต่ในความเป็นจริงแล้วองค์กรทุกขนาดสามารถนำเอา IIoT มาประยุกต์ใช้ งานได้ เนื่องจากการพัฒนาล่าสุดใน IIoT ทำให้ใช้งานได้ง่ายขึ้นและช่วยลดต้นทุนได้ โดยในปัจจุบันมี Software เกี่ยวกับ IIoT ที่เป็น ลักษณะ Open-source ที่ช่วยให้ผู้ประกอบการ SMEs ใช้ประโยชน์จาก IIoT ในการผลิตได้ง่ายขึ้นโดยไม่ต้องสร้างโครงสร้างพื้นฐานเพื่อ รองรับ IIoT ที่ครอบคลุมหรือใช้ทีมไอทีและวิศวกรจำนวนมาก ดังนั้นประโยชน์ที่จะได้รับจากการใช้งาน IIoT มีมากกว่าความกังวลด้านค่าใช้จ่าย หรือเวลา

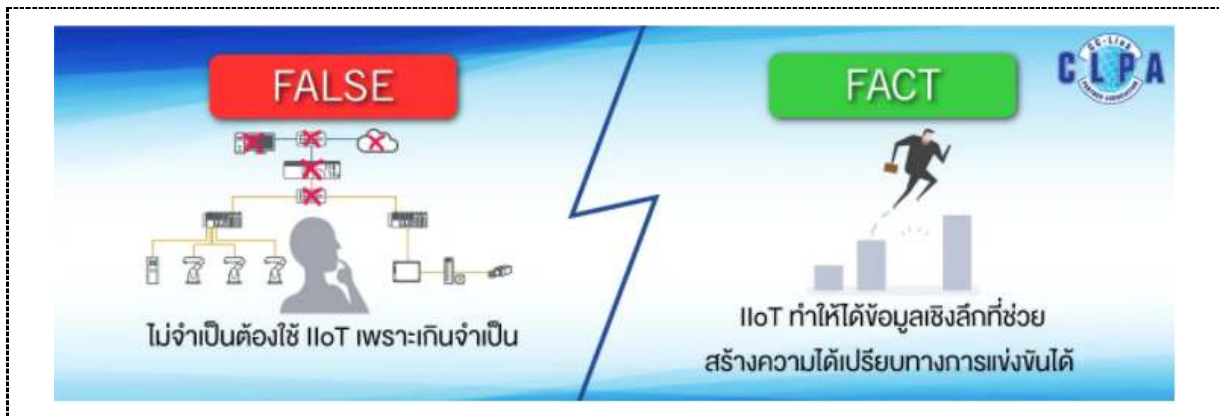


### ความเข้าใจผิดที่ 2 : การใช้งาน IIoT นั้นไม่ปลอดภัย

ความเป็นจริง : การใช้งาน IIoT มีความปลอดภัยและมีมาตรฐานด้านการรักษาความปลอดภัยรองรับ

หนึ่งในประเด็นสำคัญสำหรับผู้ผลิตเกี่ยวกับการใช้งาน IIoT คือ รู้สึกว่ามันไม่ปลอดภัย ในความเป็นจริงการเชื่อมต่ออุปกรณ์ให้สื่อสารผ่าน Cloud ก็เป็นโจทย์ที่ท้าทายในการรักษาความปลอดภัยจริงๆ นั่นแหละ แต่ในปัจจุบันมีมาตรฐานและมีการตระหนักด้านความปลอดภัยมากขึ้น ทำให้ IIoT มีความปลอดภัยเท่ากับโครงสร้างพื้นฐานด้านไอทีอื่นๆ แม้ว่าความเสี่ยงด้านความปลอดภัยอาจเกิดขึ้นได้เสมอเมื่อเชื่อมต่ออุปกรณ์กับอินเทอร์เน็ต แต่มาตรการด้านการรักษาความปลอดภัยที่นำมาใช้งานจะช่วยลดภัยคุกคามและเพิ่มมูลค่าระยะยาวให้กับโครงสร้างพื้นฐานของการทำ IIoT ได้





**ความเข้าใจผิดที่ 3 : ไม่จำเป็นต้องใช้ IIoT เพราะเกินจำเป็น**

**ความเป็นจริง :** การใช้งาน IIoT จะทำให้ธุรกิจของคุณได้ข้อมูลเชิงลึกทางธุรกิจที่ช่วยเพิ่มประสิทธิภาพทางการผลิตและเพิ่มสร้างความได้เปรียบทางการแข่งขันได้

“เราคงไม่ได้ใช้ข้อมูลจำนวนมากจากการใช้ IIoT หรือ” ด้วยความคิดนี้จึงทำให้เกิดความคิดที่ว่าการใช้งาน IIoT นั้นไม่จำเป็น และเกินจำเป็น แต่ในอนาคตมูลค่าทางธุรกิจจะเพิ่มขึ้นได้ด้วยการรวบรวมและการวิเคราะห์ข้อมูล ดังนั้นมันก็เป็นไปได้ที่จะทำให้อุปกรณ์ของคุณมี การเติบโตมากขึ้น ข้อมูลที่ไม่ได้เคยใช้ประโยชน์ใดๆ ก็อาจกลายเป็นสิ่งที่มีค่าได้ และการนำ IIoT มาใช้งานไม่เพียงแต่จะเก็บข้อมูลเท่านั้น แต่ยังช่วย ให้คุณสามารถเข้าถึงข้อมูลเชิงลึกได้ตลอดทั้งห่วงโซ่อุปทาน (Supply Chain), ช่วยสร้างแหล่งรายได้ใหม่, อีกทั้งยังสามารถเพิ่มประสิทธิภาพ และยืดระยะเวลาในการใช้งานของอุปกรณ์ได้อีกด้วย



**ความเข้าใจผิดที่ 4 : การใช้งาน IIoT เป็นแค่การเชื่อมต่อ Sensor เข้ากับอุปกรณ์**

**ความเป็นจริง :** การใช้งาน IIoT เป็นการเข้าถึงข้อมูลเชิงลึกของธุรกิจเพื่อช่วยทำให้ทำการตัดสินใจได้ดีขึ้น

การนำ IIoT มาใช้งานก็เป็นการนำอุปกรณ์และเครื่องจักรมาเชื่อมต่อกับ Sensor จริงๆ นั่นแหละ แต่มันมีอะไรที่มากกว่านั้น นอกเหนือจากการ นำอุปกรณ์ทางกายภาพที่เป็นชิ้นๆ มาเชื่อมต่อกันแล้ว IIoT ยังเกี่ยวข้องกับเครือข่ายเช่น Cloud, Gateway, API, และอื่นๆ ดังนั้นการเชื่อมต่อจึงเป็นพื้นฐานของ IIoT แต่ใจความสำคัญหลักของการนำ IIoT มาใช้งานคือ การสร้างข้อมูลและวิเคราะห์ข้อมูลเชิงลึก แบบเรียลไทม์เพื่อช่วยในการดำเนินการและการตัดสินใจ เมื่ออุปกรณ์ถูกเชื่อมต่อบนเครือข่ายเพื่อรวบรวมข้อมูลและวิเคราะห์ข้อมูลเชิงลึกทางธุรกิจ ผู้ผลิตสามารถคาดการณ์ผลลัพธ์ ป้องกันความล้มเหลวในการทำงาน และเพิ่มเวลาการทำงานของระบบให้ดีขึ้น



#### ความเข้าใจผิดที่ 5 : การใช้งาน IIoT ใช้ต้นทุนสูง

**ความเป็นจริง :** ค่าใช้จ่ายในการติดตั้ง IIoT ลดลงและช่วยในการสร้างช่องทางรายได้ใหม่ในระยะยาว

การทำ IIoT นั้นมีค่าใช้จ่ายสูงเป็นหนึ่งในความเข้าใจผิดที่เกิดขึ้นในวงกว้าง จริงอยู่ที่การลงทุนในการใช้งาน IIoT ต้องใช้เงินทุน หลายองค์กรมองเพียงแต่ด้านต้นทุนที่เกิดขึ้น แต่ไม่ได้มองผลประโยชน์ที่ได้รับจากการทำ IIoT เมื่อมีการนำ IIoT มาใช้งานแล้ว ข้อมูลเชิงลึกใน การดำเนินการที่รวบรวมได้จะช่วยลดต้นทุนในการผลิต ซึ่งจะเป็นการเพิ่มประสิทธิภาพและประสิทธิผล ผลที่ได้คือ ค่าใช้จ่ายน้อยลงและการเติบโตมากขึ้น ซึ่งจะช่วยเพิ่มผลกำไร

สาเหตุสำคัญที่ทำให้องค์กรจำเป็นต้องลดต้นทุนลงโดยที่ยังคงคุณภาพที่ส่งมอบให้ลูกค้าไว้ก็คือ การแข่งขันทางธุรกิจที่สูงขึ้น ประกอบกับการเปลี่ยนแปลงทางเทคโนโลยีเกิดขึ้นมากมาย ทำให้มีการนำระบบอัตโนมัติมาใช้งาน โดยมีการนำข้อมูลเชิงลึกแบบ Real-time มาใช้งาน และมีการติดตามผลอย่างต่อเนื่องเพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้นจากการผลิต ส่งผลให้โรงงานที่เชื่อมต่อกับ IIoT มีศักยภาพเพิ่มขึ้นหลายเท่า ดังนั้นคุณควรตรวจสอบให้แน่ใจว่าองค์กรของคุณไม่ได้ล้าหลังใน Digital Transformation เนื่องจากความเข้าใจผิดเกี่ยวกับ IIoT

## 1.5. ประโยชน์จากการนำ IIoT มาใช้งานด้านการผลิต

IIoT มีส่วนช่วยในการผลิต ตั้งแต่การลดต้นทุนในการผลิต การเพิ่มประสิทธิภาพในการผลิต การช่วยด้านการบำรุงรักษา ตลอดจนการเพิ่มความปลอดภัยในการทำงาน ในความนี้จะมีการยกตัวอย่างคร่าวๆ เพื่อให้เห็นภาพมากขึ้น

### 5.1 ช่วยเพิ่มความปลอดภัย

หนึ่งในวิธีหลักที่เทคโนโลยี IIoT สามารถช่วยเพิ่มความปลอดภัยในการผลิตก็คือ ช่วยให้สามารถรวบรวมข้อมูลได้รวดเร็วขึ้นและรวบรวมได้จำนวนมากขึ้น โดยส่วนมากความผิดปกติของอุปกรณ์เป็นหนึ่งในปัจจัยสำคัญที่ก่อให้เกิดอุบัติเหตุได้ ดังนั้นการนำเอาเทคโนโลยี IIoT มาใช้งานเพื่อเชื่อมต่ออุปกรณ์ต่างๆ จะช่วยให้สามารถตรวจสอบสถานะการทำงานและทราบถึงสภาพการทำงานของอุปกรณ์ได้ ข้อมูลที่เกิดขึ้นจริงสามารถส่งโดยตรงไปยังห้องควบคุมหรือพื้นที่ที่ต้องการได้ แบบ Real-time แทนที่จะต้องเก็บข้อมูลและป้อนข้อมูลด้วยตนเอง การวิเคราะห์ข้อมูลด้วยการใช้ AI สามารถแจ้งเตือนให้พนักงานทราบถึงปัญหาได้ในทันที อีกทั้งการวิเคราะห์ข้อมูลเป็นระยะเวลานานทำให้สามารถตรวจพบสถานะอันตรายที่อาจเกิดขึ้นได้ล่วงหน้า เพื่อป้องกัน

### 5.2 การเกิดอันตรายแก่ผู้ปฏิบัติงานได้

เพิ่มประสิทธิภาพด้านการบำรุงรักษาและปรับปรุงประสิทธิภาพของคลังสินค้า

IIoT สามารถช่วยในการบำรุงรักษาและเพิ่มประสิทธิภาพให้กับคลังสินค้าได้โดยการติดตามสถานะการทำงานตลอดห่วงโซ่อุปทาน ทำให้สามารถมองภาพรวม ในการทำงานได้ตลอดกระบวนการ IIoT ช่วยให้สามารถนับจำนวนในการใช้และจำนวนคงเหลือของวัตถุดิบและอะไหล่ได้อย่างแม่นยำและถูกต้อง ซึ่งจะช่วยป้องกัน การขาดแคลนสต็อกและช่วยลดความผิดพลาดในการบริหารจัดการคลังสินค้า เมื่อวัตถุดิบหรือสินค้าในคลังใกล้หมด sensor ก็จะส่งการแจ้งเตือนเพื่อให้ ผู้ที่รับผิดชอบรีบทราบ โดยสามารถติดตามผลและควบคุมอุปกรณ์จากระยะไกลได้

### 5.3 ช่วยลดต้นทุน

การลดต้นทุนในการผลิตนั้นมีหลายวิธี หนึ่งในวิธีนั้นคือการนำเทคโนโลยี IIoT มาใช้งาน แม้ว่าการนำเอาเทคโนโลยี IIoT มาใช้งานนั้นจะมีต้นทุนสูง แต่วิธีนี้เป็นการลงทุนเพื่อลดต้นทุนในระยะยาว โดยสามารถลดต้นทุนเกี่ยวกับการซ่อมบำรุงได้ ไม่ว่าจะเป็นต้นทุนที่เกิดจากค่าอะไหล่หรือค่าแรง เนื่องจากการเก็บข้อมูลสถานะการทำงานจริงของเครื่องจักรเพื่อคาดการณ์อายุการใช้งานคงเหลือของอุปกรณ์แต่ละชิ้นแบบ Real-time ทำให้ไม่จำเป็นต้องซ่อมบำรุงก่อนล่วงหน้าเพื่อป้องกันปัญหา และไม่จำเป็นต้องใช้แรงงานคนไปเดินจุดเพื่อติดตามสถานะการทำงานของเครื่องจักรและอุปกรณ์ต่างๆ อีกทั้งยังช่วยลดระยะเวลาในการหยุดเครื่องจักรได้อีกด้วยเพราะการบำรุงรักษานั้นอิงจากสภาพอุปกรณ์จริงจึงไม่ต้องหยุดเครื่องจักรเพื่อทำการบำรุงรักษาบ่อยเกินไป จำเป็น หรือต้องเปลี่ยนอะไหล่ก่อนกำหนด ทำให้สามารถใช้งานอุปกรณ์ได้คุ้มค่ามากขึ้นอีกด้วย

### 5.4 ช่วยเพิ่มประสิทธิภาพที่ดีให้กับลูกค้า

การทำ IIoT นั้นคือการบริหารจัดการกับข้อมูล ดังนั้นเมื่อมีการนำข้อมูลลูกค้ามาวิเคราะห์ทำให้องค์กรรู้จักลูกค้าได้ดีขึ้น และสามารถปรับปรุงกระบวนการต่างๆ เพื่อที่จะสามารถให้บริการได้ดียิ่งขึ้น เช่น กระบวนการด้านโลจิสติกส์ อีกทั้งการปรับปรุงและพัฒนาอย่างต่อเนื่องเป็นกุญแจสำคัญในการทำให้ประสิทธิภาพของลูกค้าดีขึ้น



## 1.6. เทคโนโลยีด้าน IIoT จำนวน 6 อย่างที่น่าจับตาในปี 2017

[April 27, 2017 Fayossy](#)



ซันเดอร์ อีเล็คทริก ผู้เชี่ยวชาญระดับโลกด้านการจัดการพลังงาน และระบบอัตโนมัติ เผยอุตสาหกรรมอัตโนมัติในปัจจุบัน พัฒนาไปอย่างรวดเร็วยิ่งกว่าช่วงเวลาใดๆ ในประวัติศาสตร์ที่ผ่านมา หนึ่งในสาเหตุหลักมาจากอินเทอร์เน็ตของสิ่งสำหรับภาคอุตสาหกรรม หรือ IIoT (Industrial Internet of Things) ซึ่ง IIoT และการปฏิรูปสู่ดิจิทัล ช่วยให้เกิดการหลอมรวมเทคโนโลยีส่วนปฏิบัติการ (operational technology หรือ OT) และ เทคโนโลยีสารสนเทศ (information technology หรือ IT) เข้าด้วยกัน ซึ่งนับเป็นหัวใจสำคัญสำหรับผู้ใช้งาน และผู้ประกอบการ OEMs

### 1.การวิเคราะห์ขั้นสูง ปัญญาประดิษฐ์ หรือ AI และเครื่องจักรที่เรียนรู้ได้

สิ่งเหล่านี้เป็นสิ่งที่สร้างศักยภาพให้กับ IIoT

องค์กรธุรกิจนำแพลตฟอร์มธุรกิจอัจฉริยะและระบบอัจฉริยะด้านการผลิต (EMI – Enterprise Manufacturing Intelligence) มาใช้ในการสืบหาสาเหตุและทำความเข้าใจถึงความผิดปกติที่เกิดขึ้น โดยปัจจุบัน IIoT ทำให้ผู้ประกอบการด้านการผลิตสามารถนำการวิเคราะห์ขั้นสูง รวมถึงปัญญาประดิษฐ์ (Artificial Intelligence) และการเรียนรู้ของเครื่องจักรกล (Machine Learning) มาช่วยสนับสนุนโซลูชันด้านการวิเคราะห์เชิงพยากรณ์ (predictive) รวมถึงการวิเคราะห์ในระดับที่สามารถนำเสนอทางเลือกพร้อมคาดการณ์ผลลัพธ์ที่จะตามมา (prescriptive) โดยการเชื่อมต่อข้อมูลจากสมาร์ตเซ็นเซอร์ อุปกรณ์ และสินทรัพย์อื่นๆ ที่มีอยู่เดิมเข้ากับแอปพลิเคชัน และระบบวิเคราะห์เชิงพยากรณ์บนคลาวด์ จะช่วยเพิ่มศักยภาพเชิงกลยุทธ์ให้ IIoT ช่วยปรับปรุงประสิทธิภาพด้านการผลิตได้

## 2.อุปกรณ์สำหรับอุตสาหกรรมมีการใช้งานอยู่บนเอดจ์ (Edge) มากขึ้น

ปัจจัยหลักที่ทำให้ระบบโครงสร้าง IIoT เติบโต คือ อุปกรณ์ปลายทาง หรือ เกตเวย์อัจฉริยะ ซึ่งทำหน้าที่ในการเก็บรวบรวม จัดกลุ่ม คัดกรอง และส่งข้อมูลที่อยู่กับกระบวนการทำงานในอุตสาหกรรมหรือสินทรัพย์ในส่วนของการผลิต นอกจากนี้ยังสามารถนำข้อมูลไปผ่านการวิเคราะห์ ตรวจสอบความผิดปกติได้ในแบบเรียลไทม์ และ ให้การแจ้งเตือน เพื่อให้ผู้ดูแลงานส่วนปฏิบัติการหรือผู้ควบคุมระบบสามารถดำเนินการต่างๆ ได้อย่างเหมาะสม

การย้ายระบบวิเคราะห์ไปอยู่ที่ส่วนปลายของเครือข่าย (Edge) และอยู่ใกล้กับแหล่งข้อมูลมากยิ่งขึ้น นับเป็นการช่วยปรับปรุงคุณภาพด้านการผลิตและให้ผลผลิตที่ดีขึ้น โดยเซ็นเซอร์ที่ราคาย่อมเยาร่วมระบบประมวลผล จะช่วยเพิ่มความสามารถในการรวบรวมข้อมูลในส่วนผลิตและประมวลผลได้จากส่วนปลายเครือข่าย (Edge) ดังนั้น Edge computing ที่มาพร้อมระบบวิเคราะห์แบบฝังตัว จึงนับเป็นทางเลือกที่ดีหากเราไม่สามารถรันระบบวิเคราะห์บนคลาวด์ได้ หรือผู้ประกอบการผลิต (OEM) ไม่ได้เลือกใช้โซลูชันที่ทำงานบนคลาวด์

“ส่วนปลาย (Edge)” ของระบบเครือข่ายในภาคอุตสาหกรรม กำลังกลายเป็นที่รองรับของอุปกรณ์ต่างๆ ทั้ง อีเธอร์เน็ต ไวร์เลสและเซลล์าร์เกตเวย์ อีเธอร์เน็ต สวิตช์ และเราเตอร์ รวมไปถึงคอมพิวเตอร์ขนาดเล็ก ซึ่งอุปกรณ์ที่อยู่ปลายขอบเครือข่ายเหล่านี้ จะช่วยเชื่อมโยงสภาพแวดล้อมการทำงานในส่วนงาน IT กับ OT เข้าด้วยกัน ช่วยนำอุปกรณ์ต่างๆ ไม่ว่าจะเป็นเซ็นเซอร์ และคอนโทรลเลอร์ รวมถึงสินทรัพย์อื่นๆ ไปสู่ระบบอัตโนมัติขั้นหรือสถาปัตยกรรมในระบบเอ็นเตอร์ไพรซ์

## 3.การกำเนิดของ ฝาแฝดดิจิทัล (digital twins)

IIoT และความเป็นดิจิทัล ช่วยให้บริษัทสามารถสร้างสำเนาหรือตัวก๊อปปี้ของสินทรัพย์ให้อยู่ในรูปของดิจิทัล หรือที่เรียกว่า ฝาแฝดดิจิทัล เพื่อใช้ในการจำลองสถานการณ์ และนำมาใช้งานบนสภาพแวดล้อมระบบเสมือนให้เหมาะสม (virtual environment) ก่อนที่จะใช้ทรัพยากรจริง ตัวแทนเสมือนของสินทรัพย์ที่จับต้องได้ ยังครอบคลุมไปถึงการทำสำเนาและจัดเก็บ (Archive) ข้อมูลในประวัติศาสตร์และข้อมูลแบบเรียลไทม์ รูปร่าง โมเดล ใบเสร็จของสิ่งต่างๆ การวิเคราะห์ในเชิงมิติและวิศวกรรม ข้อมูลการผลิตและประวัติศาสตร์การดำเนินงานที่สามารถนำมาใช้เป็นข้อมูลพื้นฐานเพื่อวัดมาตรฐานด้านประสิทธิภาพการทำงาน

ในทำนองเดียวกัน ข้อมูลเรียลไทม์จากการรวบรวมของเซ็นเซอร์ หรือแหล่งข้อมูลจากภายนอกองค์กร สามารถนำมาใช้ดำเนินการด้านการวิเคราะห์ เช่นการตรวจสอบเงื่อนไข การคาดเดาสาเหตุของความล้มเหลว ตลอดจนการวิเคราะห์เชิงพยากรณ์ และการวิเคราะห์เพื่อหาทางเลือกและคาดการณ์ผลลัพธ์ ความรู้ต่างๆ เหล่านี้ สามารถเพิ่มคุณค่าให้กับอายุการใช้งานของสินทรัพย์ในประเด็นต่อไปนี้

- การปรับปรุงประสิทธิภาพ
- ลด downtime
- คาดการณ์ล่วงหน้าเพื่อป้องกันความล้มเหลว
- ช่วยเรื่องการปรับปรุงส่วนงานผลิตและออกแบบให้สามารถดำเนินการได้อย่างต่อเนื่อง

#### **4.IIoT ช่วยให้ผู้ใช้ประโยชน์จาก AR และ VR**

การฝึกอบรมด้วยโปรแกรมจำลองสถานการณ์ให้กับพนักงานใหม่ อาจเป็นแนวทางที่ให้ผลลัพธ์ที่ดีในการถ่ายทอดความรู้เกี่ยวกับโรงงาน ทั้งนี้เทคโนโลยีที่ใช้งานร่วมกับ IIoT เช่น เกมต่างๆ AR/VR และ 3D Immersive ที่ใช้เป็นอุปกรณ์สวมใส่ สามารถจำลองภาพโรงงานและหน้าที่การทำงานได้อย่างสมจริง ตลอดจนระบบควบคุม และสินทรัพย์ เพื่อสร้างประสบการณ์ที่ให้ความถูกต้องแม่นยำสูง

การจำลองสถานการณ์ช่วยให้การเรียนรู้ดีขึ้น และช่วยพัฒนาทักษะในการรับมือกับสถานการณ์ไม่คาดคิดที่เกิดขึ้นในโรงงาน และยังช่วยเพิ่มความมั่นใจให้กับคนทำงานทั้งเรื่องการทำงานและการรับมือกับสถานการณ์ฉุกเฉิน ซึ่งแอปพลิเคชันจำลองสถานการณ์ต่างๆ ยังครอบคลุมถึงเรื่องการทดสอบซอฟต์แวร์ใหม่เพื่อให้ใช้งานได้จริง การสนับสนุนเรื่องการย้ายระบบ รวมไปถึงการทดสอบโปรแกรมและตรวจสอบว่าโปรแกรมนั้นทำงานได้ตรงความต้องการใช้งานได้จริง (validation)

#### **5.MQTT โปรโตคอลส่งข้อมูลสำหรับ IIoT**

Message Queueing Telemetry Transport (MQTT) ซึ่งเป็นโปรโตคอลการส่งผ่านข้อมูล ระหว่างอุปกรณ์ไปยังอุปกรณ์ หรือเครื่องจักรไปยังเครื่องจักร (machine-to-machine) จะเติบโตไปสู่การใช้เป็นโปรโตคอลส่งข้อมูลสำหรับ IIoT เนื่องจาก MQTT ถูกออกแบบมาเพื่อใช้ในการส่งและรับข้อความได้อย่างคล่องตัว ใช้ในการเชื่อมต่อระหว่างสถานที่ที่อยู่ไกลออกไป ซึ่งต้องใช้โค้ดที่มีขนาดเล็กและ/หรือแบนด์วิธเครือข่ายคุณภาพสูง ซึ่งการรับส่งข้อมูลดังกล่าวใช้สำหรับโมบายแอปพลิเคชัน เพราะมีขนาดเล็ก ใช้พลังงานน้อย และช่วยลดแพ็กเก็ตข้อมูลให้เล็กลง สามารถกระจายข้อมูลไปยังผู้รับไม่ว่าจะแค่คนเดียวหรือหลายคนได้อย่างมีประสิทธิภาพ

#### **6.ระบบรักษาความปลอดภัยบนไซเบอร์ที่แข็งแกร่งยิ่งขึ้น จะช่วยลดความกังวลเรื่อง IIoT**

การปรับปรุงเทคโนโลยีและวิธีการรักษาความปลอดภัยที่ดีขึ้น เช่น การรับรองของ Achilles (Achilles certification) จะช่วยลดความกังวล โดยการรักษาความปลอดภัยบนไซเบอร์จะไม่ใช่อุปสรรคอันยิ่งใหญ่อีกต่อไปสำหรับโซลูชันที่ให้ความสามารถด้าน IIoT ในสภาพแวดล้อมด้านอุตสาหกรรมและระบบโครงสร้างสำคัญ ทั้งนี้ Achilles Communications Certification จะให้การรับรองความแข็งแกร่งของเครือข่ายสำหรับอุปกรณ์ที่ใช้ในอุตสาหกรรม ใน 2 ระดับ โดยให้การตรวจสอบสำหรับผู้ประกอบการด้านการผลิตและลูกค้าว่าอุปกรณ์ที่ได้รับการรับรองนั้นได้มาตรฐานเรื่องความแข็งแกร่งในการสื่อสาร ตามมาตรฐานที่ได้รับการยอมรับจากอุตสาหกรรมและมีการบังคับใช้งานสำหรับผู้ดูแลระบบโครงสร้างสำคัญส่วนใหญ่

ข้อเสนอแนะ

ผู้ใช้ปลายทางและผู้ประกอบการด้านการผลิต ควรให้การตอบรับแทนการต่อต้าน เพราะ IIoT นำมาซึ่งการเปลี่ยนแปลงอย่างใหญ่หลวงในทางที่ดี ในประเด็นต่อไปนี้

- การบริหารจัดการสินทรัพย์ และช่วยหลีกเลี่ยงการเกิดดาวนไทม์ซึ่งเป็นสิ่งที่ต้องคำนึงเป็นอันดับแรก
- ซัพพลายเออร์ด้านระบบอัตโนมัติ จะต้องช่วยลูกค้าคำนวณเรื่องการคืนทุนซึ่งเป็นเหตุผลสำคัญสำหรับการลงทุนโซลูชัน IIoT ใหม่ ๆ เหล่านี้
- สินทรัพย์เดิมจะยังคงเป็นส่วนหนึ่ง และจะถูกรวมอยู่ในโซลูชันเทคโนโลยี IIoT เหล่านี้เท่าที่จะเป็นไปได้
- ท้ายที่สุด แนวโน้มและการเปลี่ยนแปลงเหล่านี้ จะทำให้ช่วงเวลานี้เป็นช่วงเวลาที่น่าตื่นเต้นอย่างยิ่งสำหรับระบบอัตโนมัติ พร้อมกับอนาคตที่สว่างไสวยิ่งขึ้นไปอีก

## 2/5 Internet of Bodies – IoB

[https://storagecityplatform.com/English/blog-detail/Internet%20of%20Bodies%20\(IoB\)](https://storagecityplatform.com/English/blog-detail/Internet%20of%20Bodies%20(IoB))

<https://www.forbes.com/sites/bernardmarr/2019/12/06/what-is-the-internet-of-bodies-and-how-is-it-changing-our-world/?sh=231386b68b7a>

<https://www.datadriveninvestor.com/2020/03/02/internet-of-bodies-a-key-notch-in-medical-sustenance-yet-another-wave-of-technological-swindle/>



หลายคนอาจจะเคยชินกับ IoT ที่ทุกสิ่งสามารถควบคุมและเชื่อมต่อผ่านอินเทอร์เน็ตได้ แต่มีอีกอย่างหนึ่งที่คล้ายกันและใกล้ตัวเรามาก ๆ นั่นคือ IoB หรือ Internet of Bodies

IoB ในปัจจุบันที่เราสามารถเห็นกันได้ง่าย ๆ เช่น เครื่องกระตุ้นหัวใจขนาดเล็กที่ฝังไว้ในช่องท้องหรือหน้าอก เพื่อช่วยให้ผู้ป่วยการควบคุมจังหวะการเต้นของหัวใจ โดยใช้แรงกระตุ้นจากกระแสไฟฟ้า หรือ Smart Watch ที่ใช้เก็บข้อมูลต่าง ๆ ในร่างกาย เช่น อุณหภูมิ อัตราเต้นของหัวใจ และอื่น ๆ เพื่อมาเป็นผลวิเคราะห์ทางสุขภาพ

นอกจากนี้ยังมีกรณีที่ IoB ได้ถูกนำไปใช้ในธุรกิจ เช่น การฝังชิปขนาดเล็กเอาไว้ในร่างกาย ทำให้เราเดินเข้าบ้านหรืออาคารต่าง ๆ ได้โดยไม่ต้องใช้บัตรหรือกุญแจ รวมไปถึงการซื้อของต่าง ๆ ก็สามารถซื้อได้เลย โดยไม่ต้องจ่ายเงินหรือใช้บัตรรูดแต่อย่างใด

ในปัจจุบัน แม้ IoB จะดูล้ำสมัยและมีประโยชน์ต่อการดำเนินชีวิต แต่ IoB ยังคงมีประเด็นในเรื่องความเป็นส่วนตัว และด้านจริยธรรมต่าง ๆ ซึ่งเป็นโจทย์ที่ท้าทายสำหรับนักพัฒนาว่าจะแก้ไขปัญหาวงนี้ไม่ให้เกิดผลกระทบได้อย่างไรในอนาคต

?Internet of Bodies (IoB) คือ การเชื่อมต่อระบบจัดเก็บและวิเคราะห์ข้อมูลเข้ากับร่างกายของเราผ่านอุปกรณ์ต่าง ๆ ที่ทำหน้าที่รับส่งข้อมูลของร่างกายเราส่งไปยังระบบผ่านเครือข่ายอินเทอร์เน็ต คล้ายกับแนวคิด Internet of Thing หรือ

อินเทอร์เน็ตของสรรพสิ่ง ข้อมูลจากร่างกายเราจะผ่านการวิเคราะห์โดยระบบ เพื่อดูว่ามีความผิดปกติใด ๆ ที่ต้องได้รับการแก้ไขหรือไม่ ซึ่งโดยส่วนใหญ่แล้ว IoB มักจะใช้เพื่อการติดตามข้อมูลด้านสุขภาพของเรานั้นเอง

การติดตั้งอุปกรณ์ที่ใช้กับร่างกายเรานั้น มีหลายแบบ ตัวอย่างการติดอยู่บนร่างกาย เช่น สมาร์ทวอช การใส่ไว้ในร่างกาย อย่างเครื่องกระตุ้นหัวใจ หรือกระทั่งฝังไว้ในร่างกาย



### What is the Internet of Bodies (IoB)?

When the Internet of Things (IoT) connects with your body, the result is the Internet of Bodies (IoB). The Internet of Bodies (IoB) is an extension of the IoT and basically connects the human body to a network through devices that are ingested, implanted, or connected to the body in some way. Once connected, data can be exchanged, and the body and device can be remotely monitored and controlled.

**There are three generations of Internet of Bodies that include:**

- **Body external:** These are wearable devices such as Apple Watches or Fitbits that can monitor our health.
- **Body internal:** These include pacemakers, cochlear implants, and digital pills that go inside our bodies to monitor or control various aspects of our health.
- **Body embedded:** The third generation of the Internet of Bodies is embedded technology where technology and the human body are melded together and have a real-time connection to a remote machine.



3/5 IoT Platform TOP 10 ในปี 2021

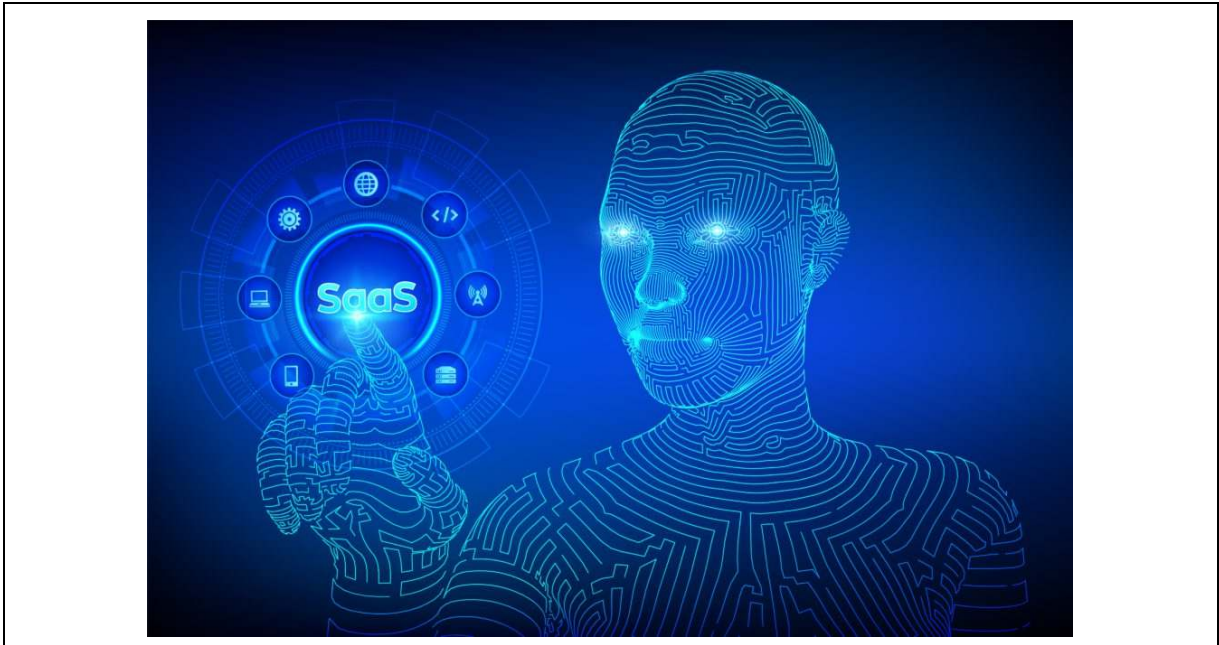
<https://ifra.io/iot-platform-top-10-ในปี-2021/>

ธันวาคม 8, 2020 [SHINJI IoT Platform](#)



ในช่วงไม่กี่ปีที่ผ่านมาเราได้เห็น IoT และศักยภาพที่ไร้ขอบเขตของมัน แม้แต่มือถือที่เราใช้อยู่ถือเป็นอุปกรณ์ IoT ที่ทำให้เราได้เชื่อมต่อกับโลกอินเทอร์เน็ตแบบไร้พรมแดน

ด้วยการเปลี่ยนแปลงอย่างรวดเร็วในปี 2020 ในทุกๆ อุตสาหกรรมบนโลกนี้ นี่ทั้งในโรคของ Covid19 ที่มาเร่งความเร็วของเรื่องเทคโนโลยีขึ้นไปอีก ทำให้ภาคส่วนของ IoT นั้นเติบโตขึ้นอย่างมหาศาล



กลับมาดูพื้นฐานทางด้าน IoT หากเราอยากจะทำ สิ่งๆ หนึ่งในด้าน IoT เราต้องมีพื้นฐานหลายๆ อย่าง ทั้งด้านเครือข่าย Server, การส่งข้อมูล [MQTT](#) และอีกมากมาย

แต่ปัจจุบันเรามีตัวช่วยในการทำโปรเจก IoT หรือ สิ่งๆ หนึ่งขึ้นมาอย่างรวดเร็วและง่ายภายในเวลาไม่ถึง 1 วัน เพื่อสร้างโปรเจก IoT ของเราเอง เช่น ระบบรดน้ำต้นไม้อัตโนมัติ ในหน้าบ้าน สิ่งเหล่านี้ หากมี Platform IoT ที่ช่วยเชื่อมต่อ Board หรืออุปกรณ์ของเราจะทำให้เราสร้างโปรเจกของเราได้อย่างรวดเร็วมากๆ

Platform IoT นั้นมีหลายหลายมากๆ ในตลาดทั้งใช้สำหรับสร้างระบบใหญ่ๆ ที่ใช้ในองค์กร เช่น Google Cloud, Amazon แต่ Platform พวกนี้มักจะใช้งานยากและต้องเป็นผู้เชี่ยวชาญเฉพาะด้าน อีกทั้งยังใช้เวลาศึกษานานมากๆ ด้วย



สำหรับ IoT Platform ในปี 2021 ที่เราเลือกมานี้ จะสามารถช่วยให้เราสามารถทำโปรเจกได้อย่างรวดเร็วและง่ายดาย จากที่เคยเราภายในเวลาสั้นๆ มาดูกันว่ามียอะไรบ้าง โดยการจัดอันดับครั้งนี้เราจะเลือกจากเงื่อนไขดังต่อไปนี้

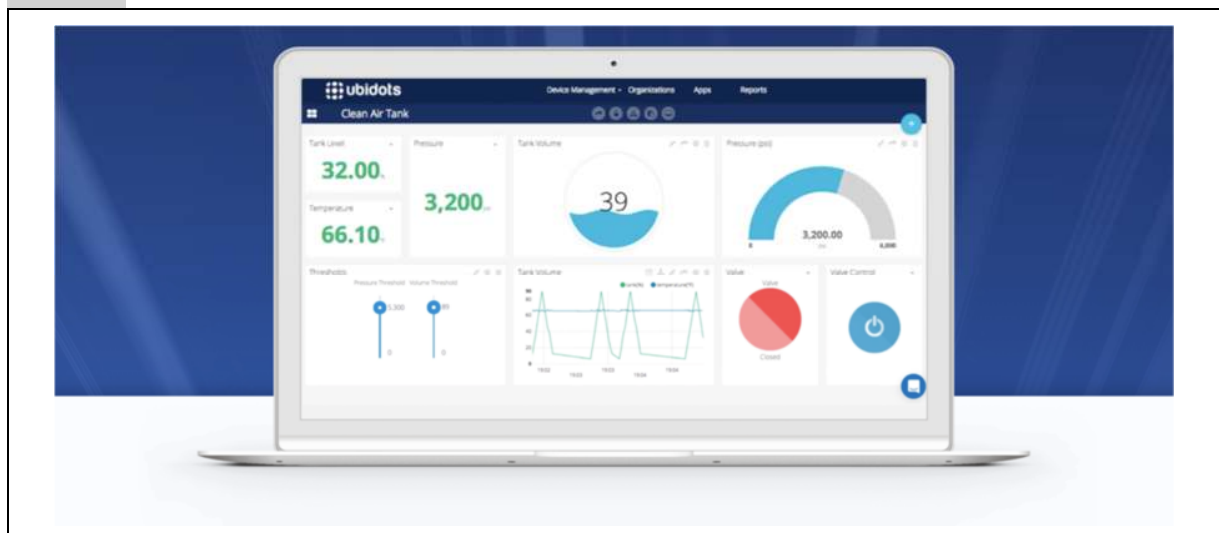
- ใช้งานง่าย สำหรับคนไม่มีพื้นฐานทางด้าน IoT สามารถใช้งานได้
- สามารถสร้างโปรเจกด้าน IoT ได้หลากหลายและรวดเร็ว
- มีคู่มือการใช้งานสามารถอ่านและทำตามได้ง่าย
- ความคุ้มค่าและราคา โดยเทียบแต่ละแพ็คเกจอันไหนราคาถูกและคุ้มค่าที่สุด

## 1. Thingsboard



Thingsboard เป็น Open-source สามารถใช้งานได้ฟรีด้วย Start ใน [Github](https://github.com) ถึง 7,700 ดาว สามารถเชื่อมต่อได้หลากหลายและใช้งานง่าย

## 2. Ubidots



เป็นเว็บแอปพลิเคชัน สามารถสร้าง Internet of Things (IoT) อย่างรวดเร็วโดยไม่ต้องเขียนโค้ดหรือจ้างทีมพัฒนาซอฟต์แวร์ ส่วนราคาเริ่มต้นนั้นแพงกว่าเจ้าอื่นพอสมควรเริ่มต้นที่ \$49 แต่โดยรวม UI ใช้งานได้ง่ายมาก

### 3. Thinger



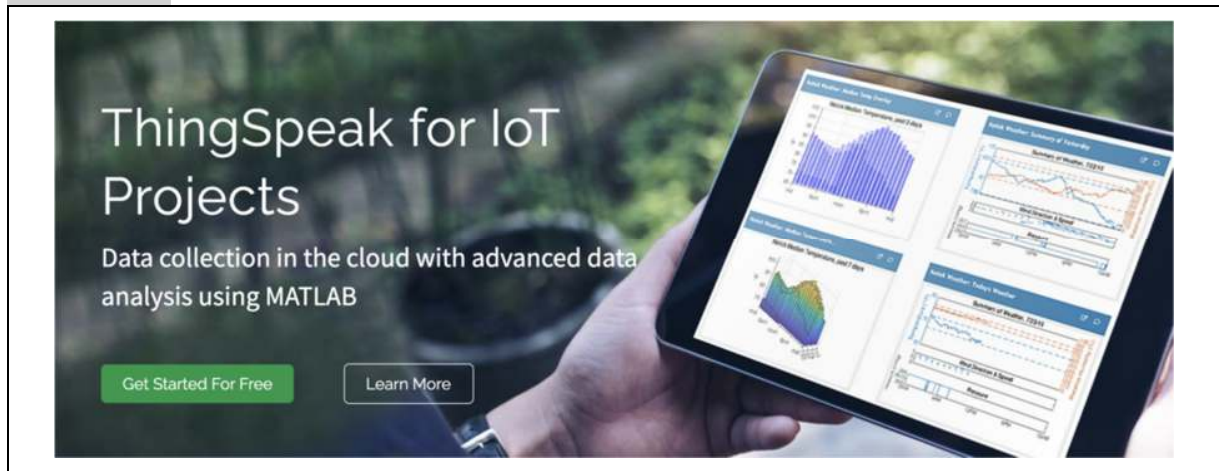
Thinger สามารถสร้างต้นแบบปรับขนาดและจัดการผลิตภัณฑ์ IoT จุดเด่นคือเป็น Open Source สามารถใช้งานได้ฟรี

### 4. Blynk

BlynK นั้นใช้งานได้ง่าย มากกว่า Platform ทั้งหมด โดยข้อดีคือคลิกลากวาง ไม่มี Step เหมาะสำหรับการทำ Prototype เบื้องต้น ข้อเสียคือใช้ได้แค่ใน Mobile ไม่สามารถใช้งานได้บน Browser

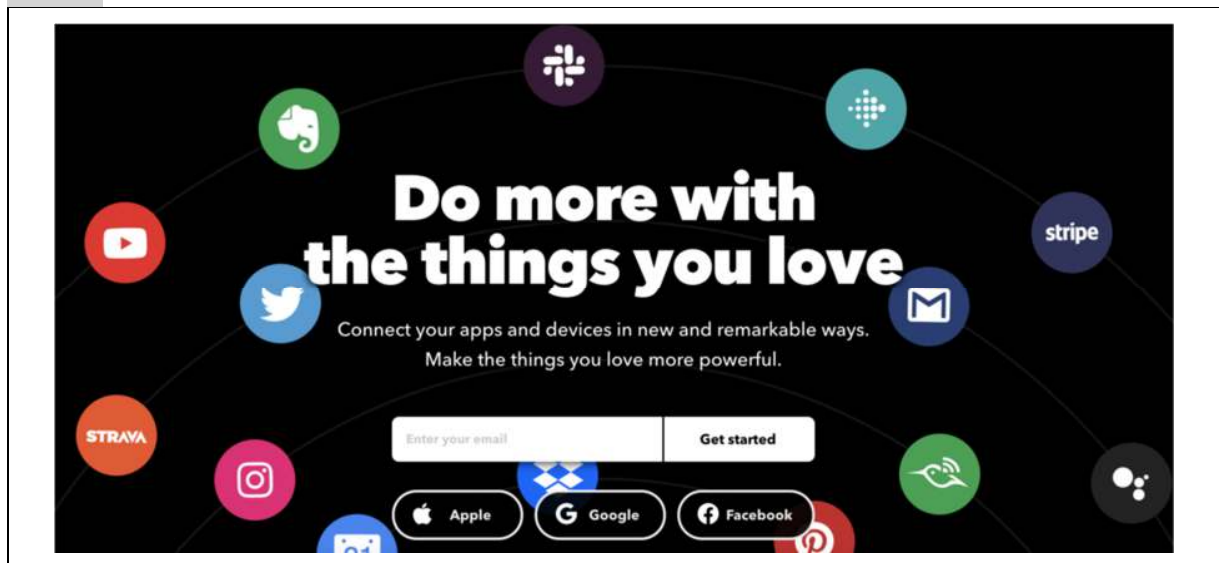


## 5. Thingspeak



Thingspeak เป็น platform รวมข้อมูลในระบบคลาวด์ด้วยการวิเคราะห์ข้อมูลขั้นสูงโดยใช้ MATLAB โดยดำเนินการวิเคราะห์ข้อมูลแบบละเอียดและแม่นยำ จะอยู่เหนือ Platform อื่นๆทั้งหมด

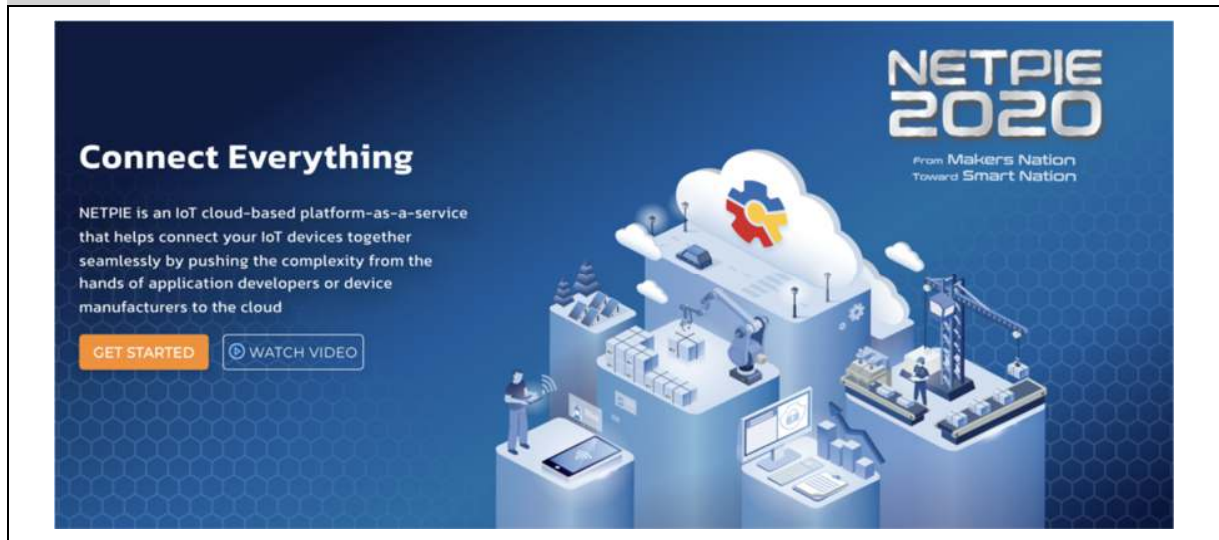
## 6. IFTTT



IFTTT สามารถเชื่อมต่ออุปกรณ์ IoT ไปยัง อุปกรณ์ IoT อีกเครื่องหนึ่งได้อย่างง่ายดายเช่นเชื่อมต่อมือถือหรือแจ้งเตือนเปิด APP Devices ต่าง ๆ ได้อย่างง่ายดาย



## 7. Netpie



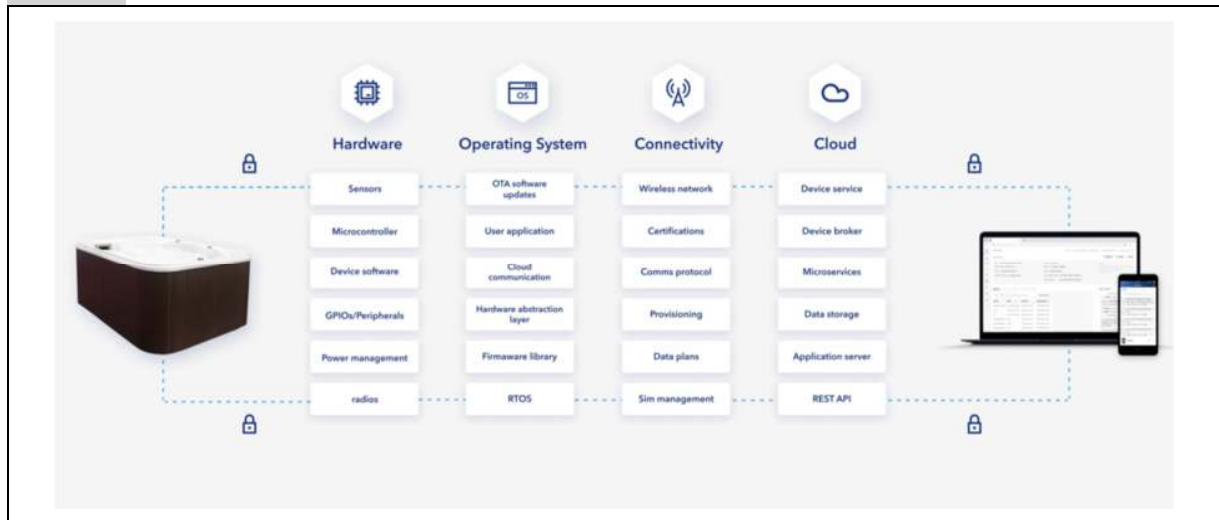
Netpie เป็น Platform ไทย ที่สนับสนุนโดย nectec โดย Netpie สามารถใช้งานได้ฟรี ระบุกับหนึ่ง สามารถเชื่อมต่ออุปกรณ์ได้ง่าย มี Community หลายหมื่นคน ที่เข้มแข็ง ที่ช่วยกันตอบปัญหาและคำถามต่างๆ

## 8. Kaa



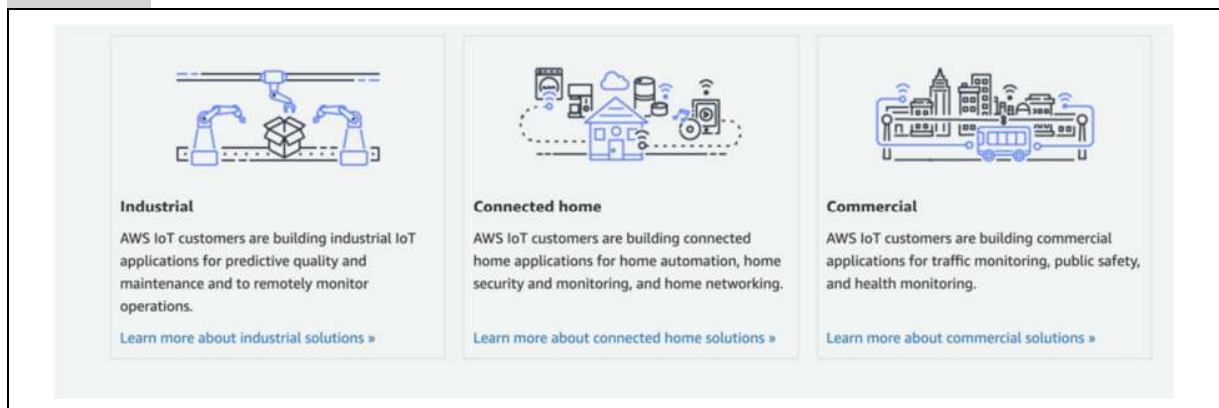
Kaa project มี solution platform ที่เยอะมากๆ สามารถตอบโจทย์การสร้างโปรเจค IoT ที่หลากหลาย ข้อเสียคือใช้งานยากและมีราคาแพงเหมาะสำหรับองค์กร หรือบริษัท ใหญ่ๆ ที่ต้องการตอบโจทย์ solution

## 9. Particle



Particle ถึงแม้จะใช้งานได้ ง่ายๆ หน่อย แต่ราคาไม่แพง เริ่มต้นที่ \$2.99 และทดลองใช้งานได้ฟรี 3 เดือนเลยทีเดียว ข้อเสียคือต้องเชื่อมต่อ Hardware ของ Particle เอง

## 10. AWS IoT



หากต้องการทำ IoT Project ขนาดใหญ่ ขอแนะนำ AWS IoT ที่สามารถตอบโจทย์ได้หลากหลายทุกอุตสาหกรรม และมีฟังก์ชัน Support เยอะมาก ข้อเสียคือ ราคาค่อนข้างแพง และใช้งานยากสำหรับผู้ที่ไม่มีความรู้พื้นฐาน IoT และการเขียนโปรแกรม

#### 4/5 การโปรแกรมใช้งาน Ubidots เพื่อแสดงค่าและควบคุมอุปกรณ์

Read More - Test with MQTTLens

<https://youtu.be/37jVqKN1JJs>

<https://ubidots.com/docs/hw/#mqtt>

#### การทดลองที่ 1:- Start Ubidots with MQTT Protocol

1. Ubidots -- <https://ubidots.com/>



2. Signed Up (or Signed In) -- <https://industrial.ubidots.com/accounts/signin/>

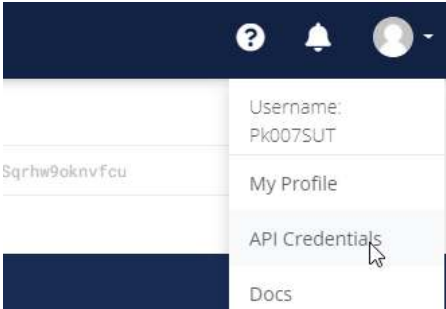


3. Create Device and Variable

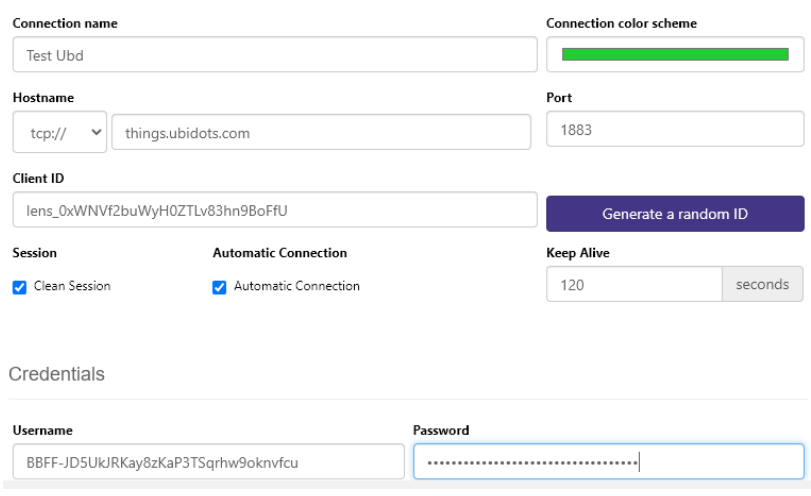
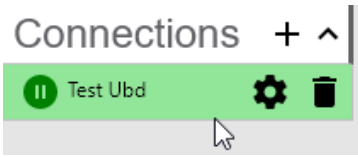
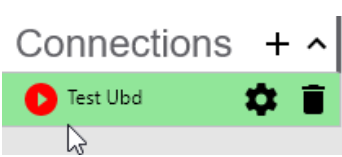
	Device → Device
	Create Device Blank Device  Device Name = pk007test Device Label = pk007test <all lowercase alphabet>
	Click pk007test Device Add Variable  humid temp  <all lowercase alphabet>

<div>humid</div> <div>Description</div> <div>Change description</div> <div>API Label</div> <div>humid</div> <div>ID</div> <div>609167e51d84726807ca16a6</div>	<div>tempp</div> <div>Description</div> <div>Change description</div> <div>API Label</div> <div>tempp</div> <div>ID</div> <div>609167f61d8472692a47034e</div>	Click ... Set API Label = humid Set API Label = tempp <all lowercase alphabet>
---	---	---

#### 4. Get Tokens Key

	API Credentials
<div>Tokens</div> <div>Default token</div> <div>BBFF-JD5UkJRKay8zKaP3TSqrhw9oknvfcu</div> <div>More</div>	Get Tokens Key

#### 5. MQTTLens

	Start MQTTLens  Name = Test Ubd  Host = things.ubidots.com Port = 1883  Username = Tokens Key Password = Tokens Key  Save
 	Left = ready Right = check again

## 6. Subscribe/Publish Test

Subscribe

/v2.0/devices/pk007test/humid

0 - at most once

SUBSCRIBE

Subscribe

/v2.0/devices/pk007test/temp

0 - at most once

SUBSCRIBE

Publish

/v2.0/devices/pk007test

0 - at most once

☐ Retained

PUBLISH

Message

{ "humid" : 12, "temp": 34 }

Subscriptions

Topic: "/v2.0/devices/pk007test/humid"

Showing the last 1 messages

+

-

Messages: 0/1

#

Time

Topic

QoS

0

10:42:11

/v2.0/devices/pk007test/humid

0

JSON

12.0, "timestamp": 1620142914558, "context": {}, "created\_at": 1620142914558

Topic: "/v2.0/devices/pk007test/temp"

Showing the last 1 messages

+

-

Messages: 0/2

#

Time

Topic

QoS

1

10:41:53

/v2.0/devices/pk007test/temp

0

JSON

34.0, "timestamp": 1620142914558, "context": {}, "created\_at": 1620142914558

Subscribe Device / Variable

/v2.0/devices/pk007test/humid

Subscribe Device / Variable

/v2.0/devices/pk007test/temp

Publish

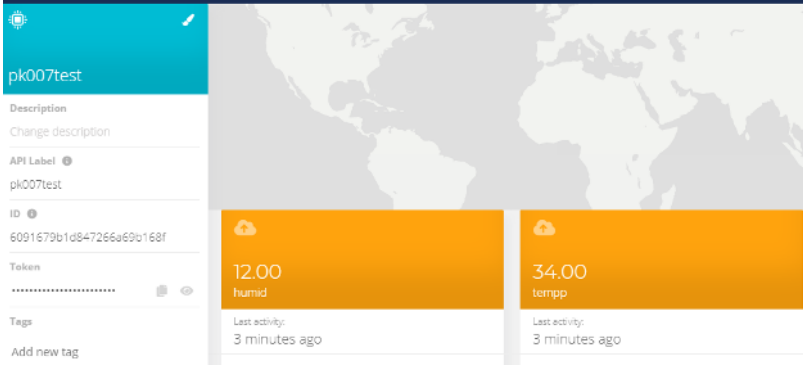
/v2.0/devices/pk007test

{"humid" : 12, "temp" : 34}

Response humid

Response temp

## 7. Result

	<p>Show</p> <p>pk007test</p> <p>humid = 12, temp=34</p>
--	---



## การทดลองที่ 2:- ESP32 to Ubidots via MQTT

8. Prepare - make sure for esp32 library

### 8.1 DHT22 Library

DHT sensor library for ESPx by beegee\_tokyo Ver 1.17.0

#### DHT sensor library for ESPx

by beegee\_tokyo Version 1.17.0 **INSTALLED**

Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized lib  
changes: Reduce CPU usage and add decimal part for DHT11

[More info](#)

### 8.2 PubSubClient

PubSubClient by Nick O'Leary – V2.8.0

#### PubSubClient

by Nick O'Leary Version 2.8.0 **INSTALLED**

A client library for MQTT messaging. MQTT is a lightweight messaging protocol for send and receive MQTT messages. It supports the latest MQTT 3.1.1 protocol if needed. It supports all Arduino Ethernet Client compatible hardware, including CC3000.

9. Coding Exp2-1/3 – Send Fix Data to Ubidots

```
#include <WiFi.h>
#include <PubSubClient.h>
```

```
const char *My_SSID = "Mue.Home";
const char *My_Pass = "pk1212312121";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-JD5UkJRKay8zKaP3TSqrhw9oknvfcu";
const char *MQTT_Pass = "BBFF-JD5UkJRKay8zKaP3TSqrhw9oknvfcu";
const char *PTopic1 = "/v2.0/devices/pk007test";
const char *STopic1 = "/v2.0/devices/pk007test/humid";
const char *STopic2 = "/v2.0/devices/pk007test/tempp";
#define MQTT_Port 1883
```

```
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;
```

```
void Setup_Wifi() {
  delay(10); Serial.println();
  Serial.print("Connecting to ");
  Serial.println(My_SSID);
  WiFi.begin(My_SSID, My_Pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  randomSeed(micros());
  Serial.println(""); Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());
}
```

```

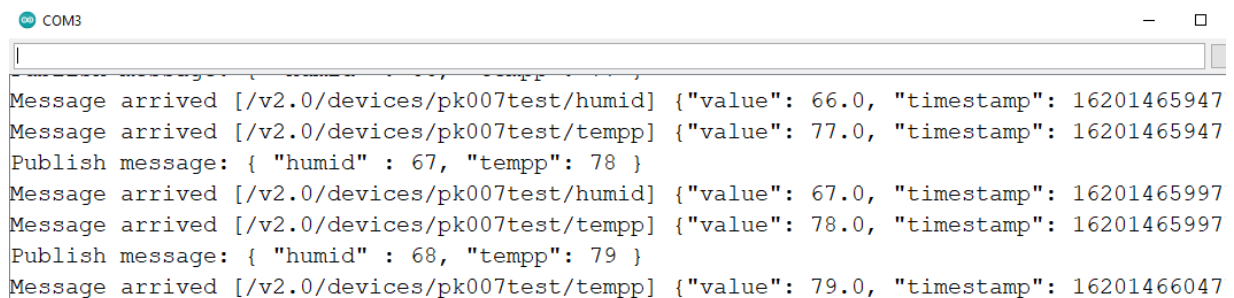
void reconnect()
{ while (!client.connected())          // Loop until we're reconnected
  { Serial.print("Attempting MQTT connection...");
    String clientId = "ESP32 Client-";
    clientId += String(random(0xffff), HEX);    // Create a random client ID
    if (client.connect(clientId.c_str(), MQTT_User, MQTT_Pass)) // Attempt to connect
    { Serial.println("connected");              // Once connected, publish an announcement...
      client.subscribe(STopic1);
      client.subscribe(STopic2);
    } else
    { Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void callback(char* topic, byte* payload, unsigned int length)
{ Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++)
  { Serial.print((char)payload[i]);
  }
  Serial.println();
}

void setup()
{ Serial.begin(115200);
  Setup_Wifi();
  client.setServer(MQTT_Server, MQTT_Port);
  client.setCallback(callback);
}

void loop()
{ if (!client.connected()) reconnect();
  client.loop();
  long now = millis();
  if (now - lastMsg > 5000)
  { lastMsg = now;
    ++value;
    snprintf(msg, 75, "{\"humid\" : %ld, \"tempp\" : %ld }", value, value + 11);
    Serial.print("Publish message: ");
    Serial.println(msg);
    client.publish(PTopic1, msg);
  }
}

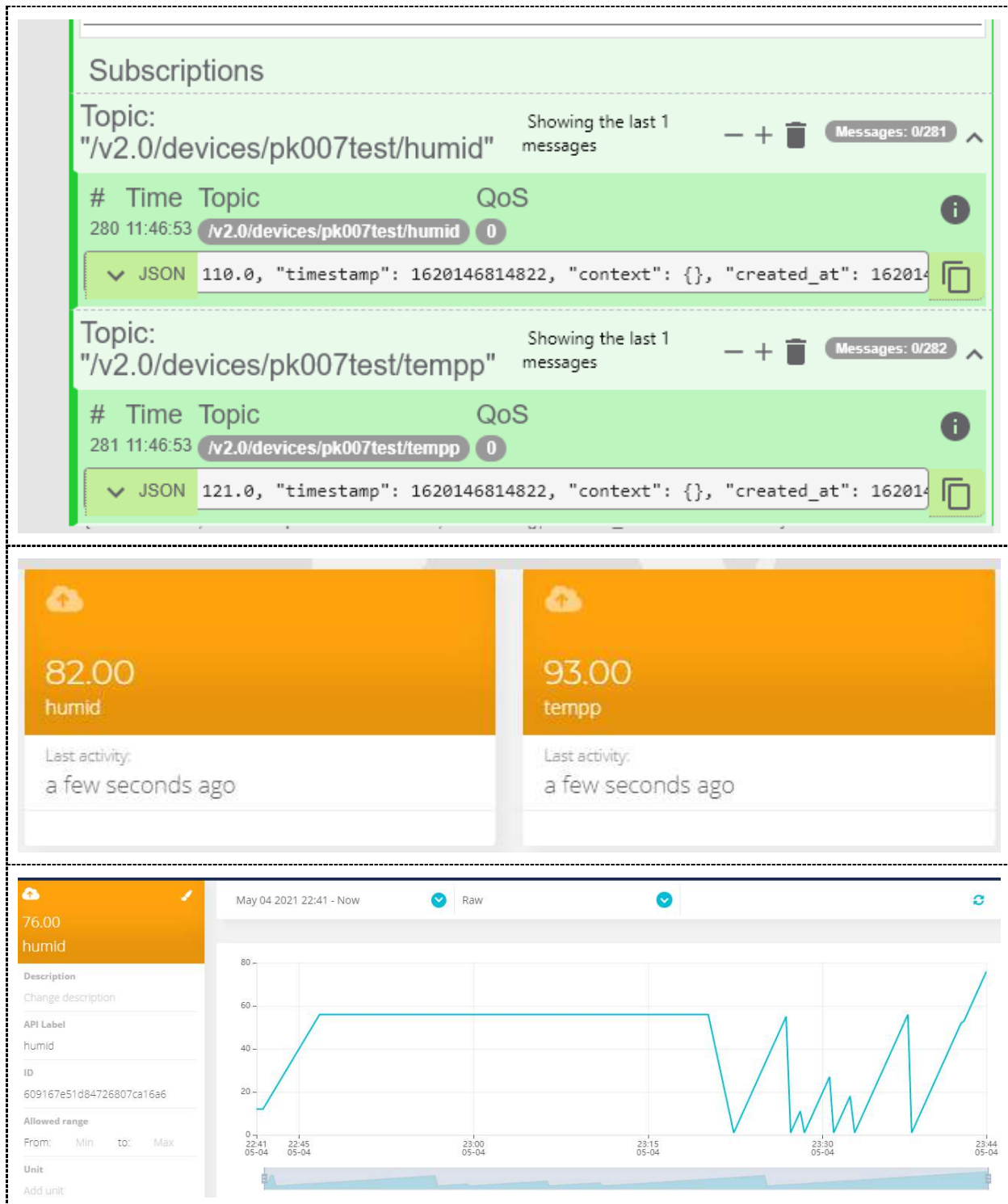
```



```

COM3
Message arrived [/v2.0/devices/pk007test/humid] {"value": 66.0, "timestamp": 16201465947
Message arrived [/v2.0/devices/pk007test/tempp] {"value": 77.0, "timestamp": 16201465947
Publish message: { "humid" : 67, "tempp": 78 }
Message arrived [/v2.0/devices/pk007test/humid] {"value": 67.0, "timestamp": 16201465997
Message arrived [/v2.0/devices/pk007test/tempp] {"value": 78.0, "timestamp": 16201465997
Publish message: { "humid" : 68, "tempp": 79 }
Message arrived [/v2.0/devices/pk007test/tempp] {"value": 79.0, "timestamp": 16201466047

```



## 10. Coding Exp2-2/3 – Send Fix Data Data to Ubidots and LED Control from Ubidots

## 10.1 Subscribe led1 variable and publish data

### Subscribe

0 - at most once

SUBSCRIBE

### Publish

0 - at most once

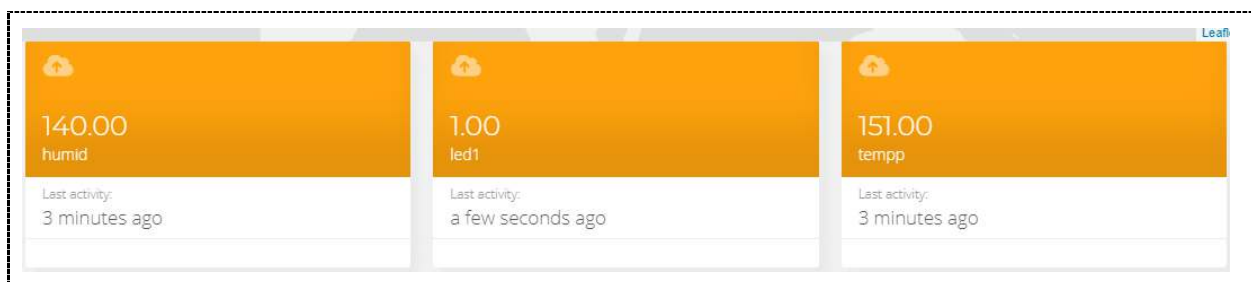
☐ Retained

PUBLISH

Message

```
{ "led1" : 1, "humid" : 56, "tempp": 78 }
```

## 10.2 Variable led1 on Ubidots Device was create



## 10.3 Control On → {"led1":1} for On, {"led1":0} for off

Off → Publish {"led1":0}

### Subscribe

0 - at most once

SUBSCRIBE

### Publish

0 - at most once

☐ Retained

PUBLISH

Message

```
{"led1": 0}
```

### Subscriptions

Topic: "/v2.0/devices/pk007test/led1" Showing the last 1 messages Messages: 0/4

#	Time	Topic	QoS
3	12:14:19	/v2.0/devices/pk007test/led1	0

JSON 0.0, "timestamp": 1620148459778, "context":

On → Publish {"led1":1}

### Subscribe

0 - at most once

SUBSCRIBE

### Publish

0 - at most once

☐ Retained

PUBLISH

Message

```
{"led1": 1}
```

### Subscriptions

Topic: "/v2.0/devices/pk007test/led1" Showing the last 1 messages Messages: 0/6

#	Time	Topic	QoS
4	12:18:22	/v2.0/devices/pk007test/led1	0

JSON 1.0, "timestamp": 1620148703401, "context":

## 10.4 Test Code 3B/4-2/3 Fix Data Data to Ubidots and LED Control from Ubidots

```

#include <WiFi.h>
#include <PubSubClient.h>
const char *My_SSID = "Mue.Home";
const char *My_Pass = "pk1212312121";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-JD5UkJRKay8zKaP3TSqrhw9oknvfcu";
const char *MQTT_Pass = "BBFF-JD5UkJRKay8zKaP3TSqrhw9oknvfcu";
const char *PTopic1 = "/v2.0/devices/pk007test";
const char *STopic1 = "/v2.0/devices/pk007test/humid";
const char *STopic2 = "/v2.0/devices/pk007test/temp";
const char *STopic3 = "/v2.0/devices/pk007test/led1";
#define MQTT_Port 1883
#define Test_LED1 2

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

void Setup_Wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(My_SSID);
  WiFi.begin(My_SSID, My_Pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }

  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void reconnect()
{ while (!client.connected())          // Loop until we're reconnected
  { Serial.print("Attempting MQTT connection...");
    String clientId = "ESP32 Client-";
    clientId += String(random(0xffff), HEX);    // Create a random client ID
    if (client.connect(clientId.c_str(), MQTT_User, MQTT_Pass)) // Attempt to connect
    { Serial.println("connected");              // Once connected, publish an announcement...
      client.subscribe(STopic1);
      client.subscribe(STopic2);
      client.subscribe(STopic3);
    } else
    { Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

```



```

void callback(char *topic, byte *payload, unsigned int length)
{
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++)
  {
    Serial.print((char)payload[i]);
  }

  if (topic[24] == STopic3[24]) {
    Serial.print(" -LED1->> ");
    Serial.print((char)payload[10]);
    if (payload[10] == '1')
      digitalWrite(Test_LED1, HIGH);
    else
      digitalWrite(Test_LED1, LOW);
  }
  Serial.println();
}

void setup()
{
  pinMode(Test_LED1, OUTPUT);
  Serial.begin(115200);
  Setup_Wifi();
  client.setServer(MQTT_Server, MQTT_Port);
  client.setCallback(callback);
}

void loop()
{
  if (!client.connected()) reconnect();
  client.loop();
  long now = millis();
  if (now - lastMsg > 5000)
  {
    lastMsg = now;
    ++value;
    snprintf (msg, 75, "{\"humid\" : %ld, \"tempp\" : %ld }", value, value + 11);
    Serial.print("Publish message: ");
    Serial.println(msg);
    client.publish(PTopic1, msg);
  }
}

```

```

1, "context": {}, "created_at": 1620149156360}
10, "context": {}, "created_at": 1620149156360}
  "context": {}, "created_at": 1620149160372} -LED1->> 1
1, "context": {}, "created_at": 1620149161359}

```

On LED1

```

Message
{"led1": 1}

```

```

187, "context": {}, "created_at": 1620149291387}
187, "context": {}, "created_at": 1620149291387}
1, "context": {}, "created_at": 1620149292632} -LED1->> 0
186, "context": {}, "created_at": 1620149296386}

```

Off LED1

```

Message
{"led1": 0}

```

## 11. Coding Exp2-3/3 – Send DHT-22 Data to Ubidots and LED Control from Ubidots

```

#include <WiFi.h>
#include <PubSubClient.h>
#include "DHTesp.h"

const char *My_SSID = "Mue.Home";
const char *My_Pass = "pk1212312121";
const char *MQTT_Server = "things.ubidots.com";
const char *MQTT_User = "BBFF-JD5UkJRKay8zKaP3TSqrhw9oknvfcu";
const char *MQTT_Pass = "BBFF-JD5UkJRKay8zKaP3TSqrhw9oknvfcu";
const char *STopic1 = "/v2.0/devices/pk007test";
const char *STopic1 = "/v2.0/devices/pk007test/humid";
const char *STopic2 = "/v2.0/devices/pk007test/tempp";
const char *STopic3 = "/v2.0/devices/pk007test/led1";
#define MQTT_Port 1883
#define Test_LED1 2
#define Pin_DHT22 4

DHTesp dht;
WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

void Setup_Wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(My_SSID);
  WiFi.begin(My_SSID, My_Pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void reconnect()
{ while (!client.connected())          // Loop until we're reconnected
{ Serial.print("Attempting MQTT connection...");
  String clientId = "ESP32 Client-";
  clientId += String(random(0xffff), HEX);    // Create a random client ID
  if (client.connect(clientId.c_str(), MQTT_User, MQTT_Pass)) // Attempt to connect
  { Serial.println("connected");              // Once connected, publish an announcement...
    client.subscribe(STopic1);
    client.subscribe(STopic2);
    client.subscribe(STopic3);
  } else
  { Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    delay(5000);
  }
}
}
}

```

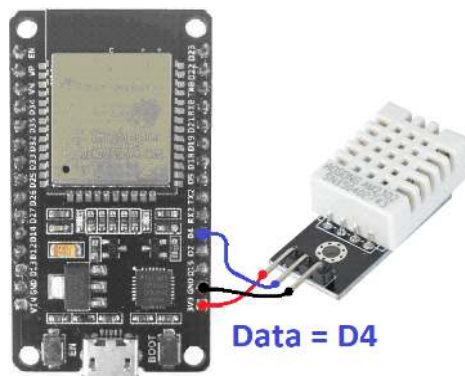
```

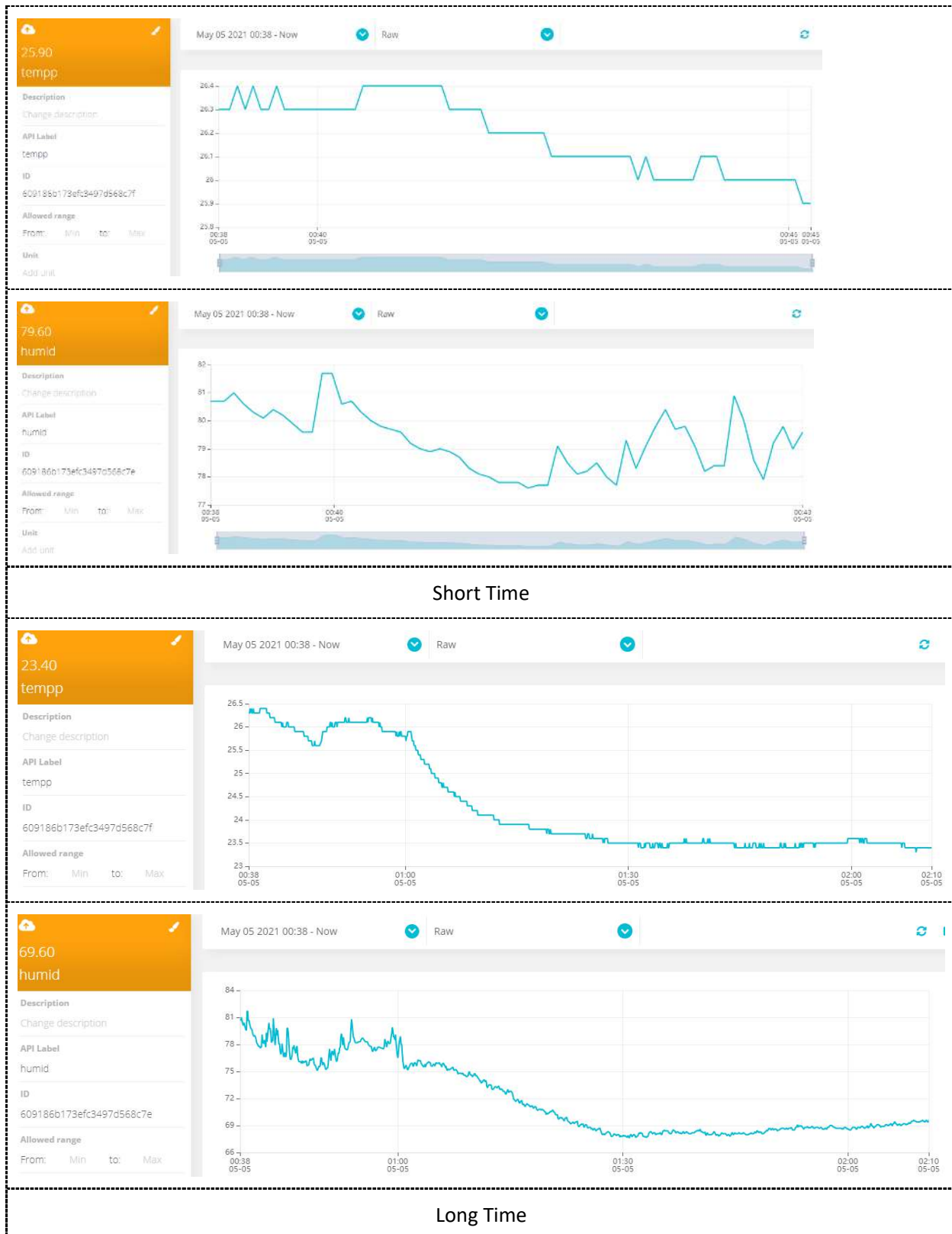
void callback(char *topic, byte *payload, unsigned int length)
{
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++)
  {
    Serial.print((char)payload[i]);
  }
  if (topic[24] == STopic3[24]) {
    Serial.print(" -LED1->> ");
    Serial.print((char)payload[10]);
    if (payload[10] == '1')
      digitalWrite(Test_LED1, HIGH);
    else
      digitalWrite(Test_LED1, LOW);
  }
  Serial.println();
}

void setup()
{
  pinMode(Test_LED1, OUTPUT);
  dht.setup(Pin_DHT22, DHTesp::DHT22);
  Serial.begin(115200);
  Setup_Wifi();
  client.setServer(MQTT_Server, MQTT_Port);
  client.setCallback(callback);
}

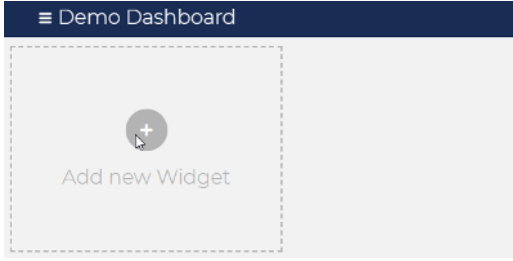
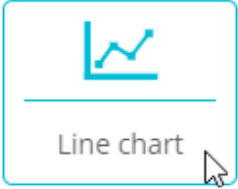
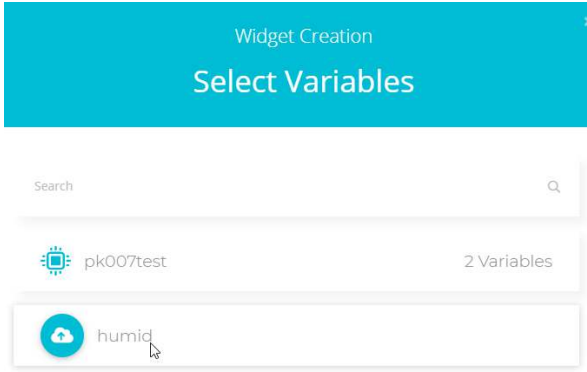
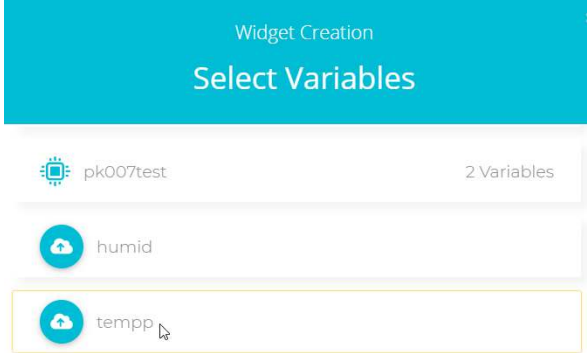
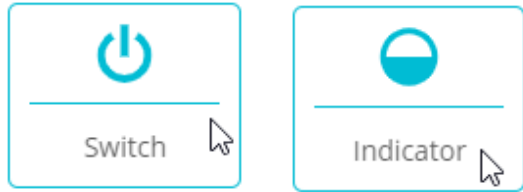
void loop()
{
  if (!client.connected()) reconnect();
  client.loop();
  long now = millis();
  if (now - lastMsg > 5000)
  {
    lastMsg = now;
    float humidity = dht.getHumidity();
    float temperature = dht.getTemperature();
    snprintf(msg, 75, "{\"humid\" : %.2f, \"temp\" : %.2f}", humidity, temperature);
    Serial.print("Publish message: ");
    Serial.println(msg);
    client.publish(PTopic1, msg);
  }
}

```



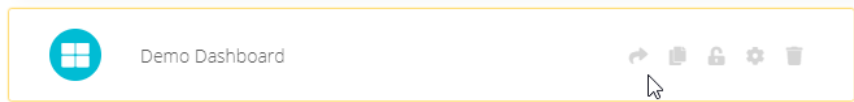




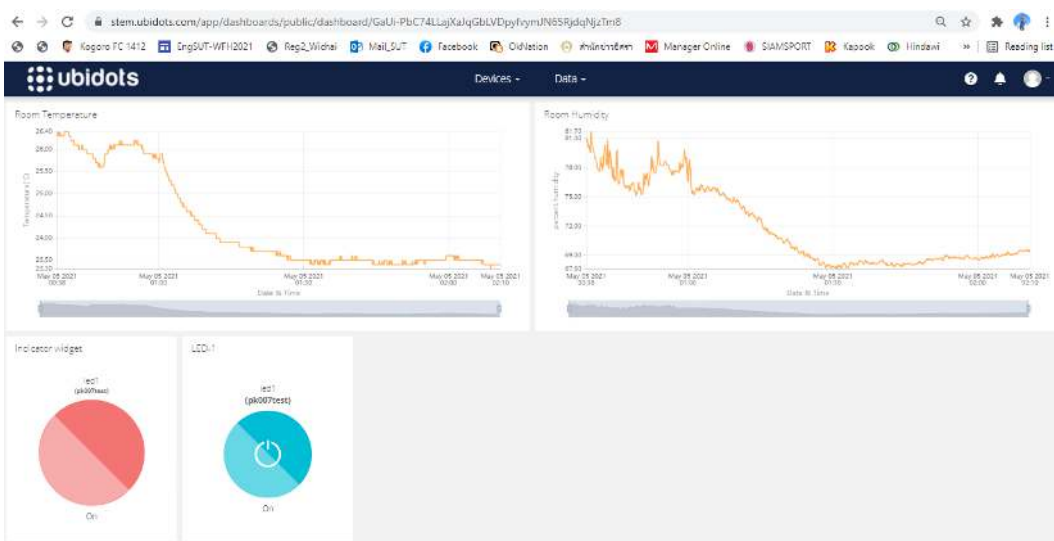
## 12. Create Dashboard

	Add new Widget
	Line chart
	Add Value → humid
	Line chart Add Value → tempp
	Switch Add Value → led1  Indicator Add Value → led1

### 13. Test Dashboard and Create Share Dashboard Link

	
	Click Demo Dashboard
	Click Share Create Link

### 14. Test with Browser



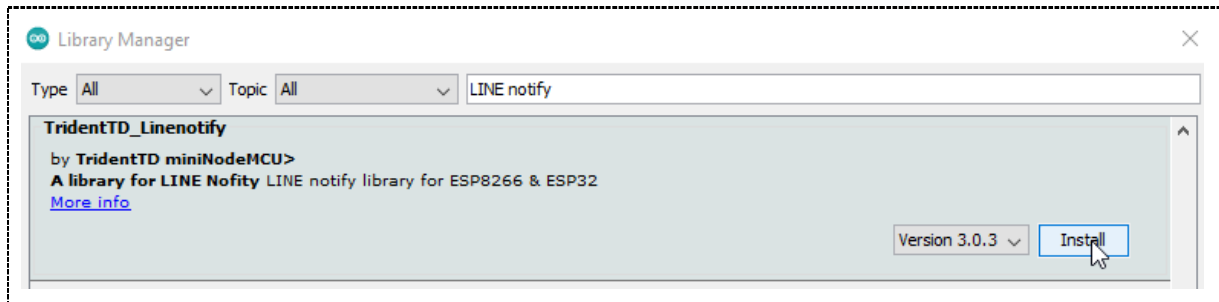


### การทดลองที่ 3:- ESP32 to LINE Notify

[https://github.com/TridentTD/TridentTD\\_LineNotify](https://github.com/TridentTD/TridentTD_LineNotify)  
<https://www.praphas.com/forum/index.php?topic=356.0>

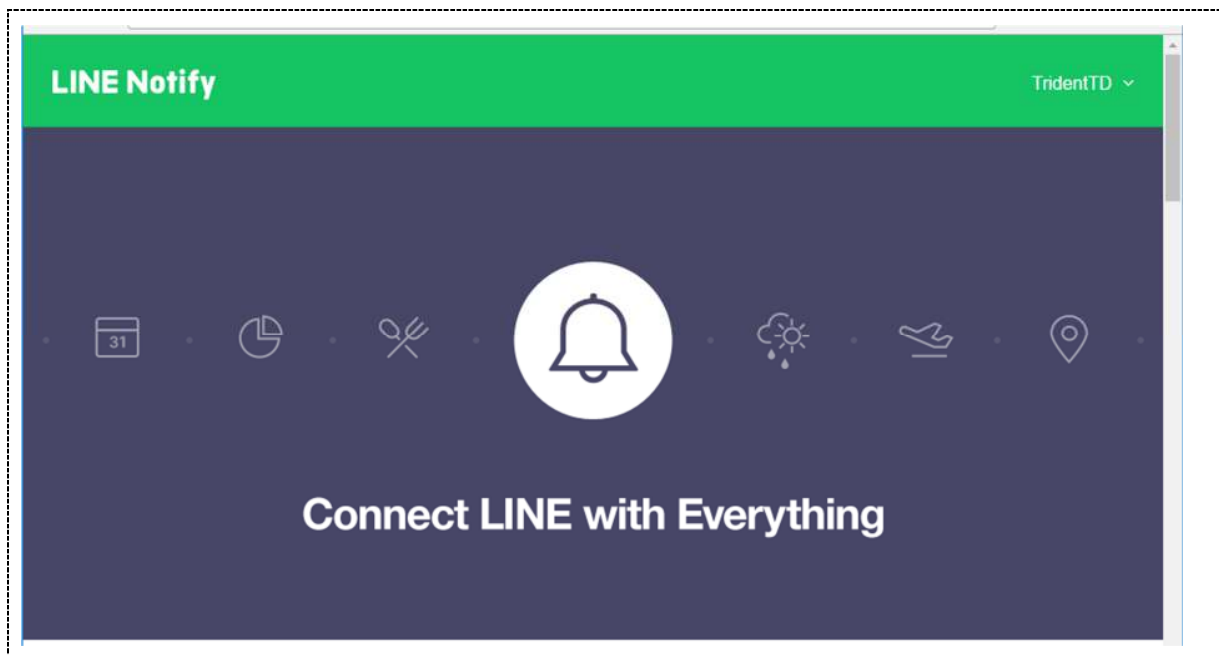
#### 1. Add Library

- Sketck → Inc Lib → Manage
- Filter ด้วยข้อความ LINE notify เพิ่มไลบรารี [TridentTD\\_LineNotify Ver 3.0.3](#)

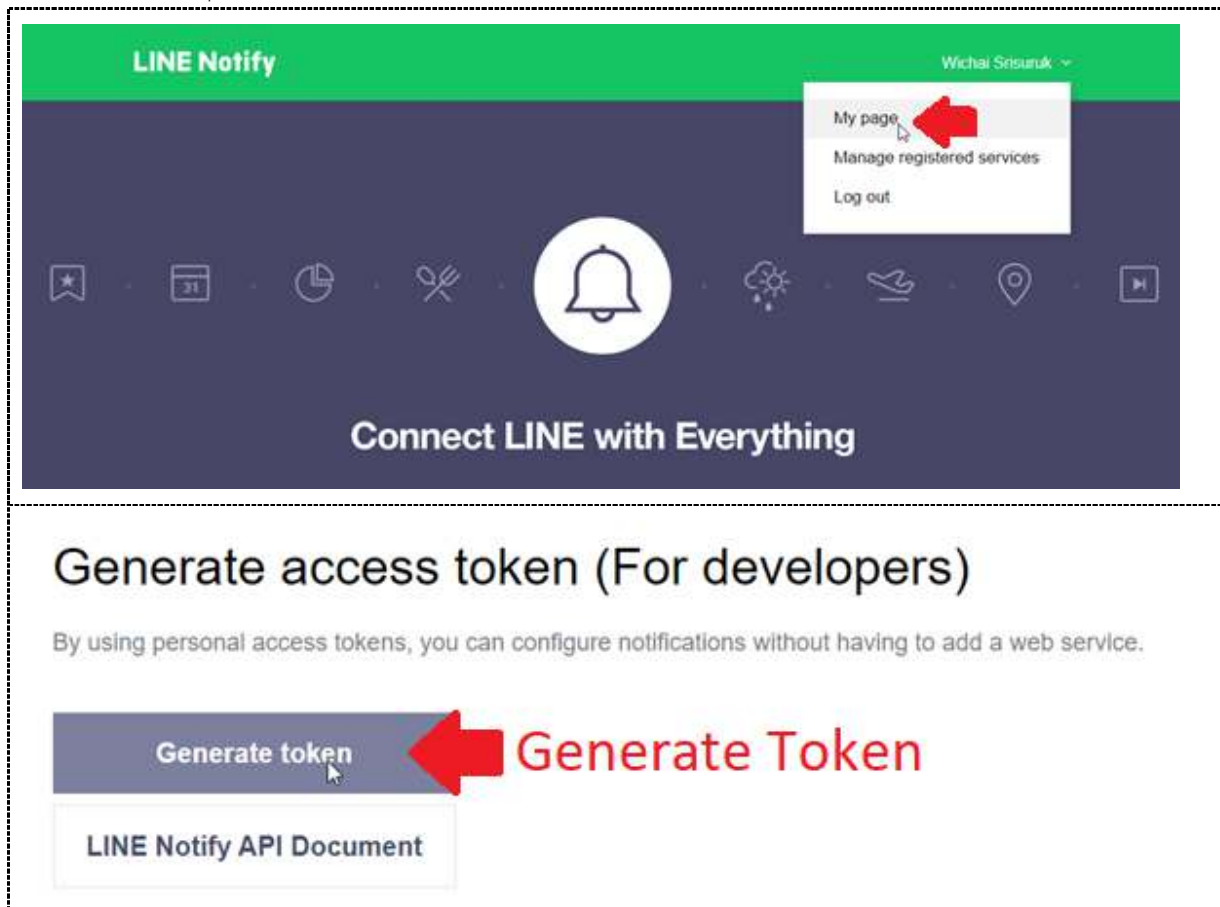


#### 2. Create LINE Token

- ทำการ Log in ที่ <https://notify-bot.line.me/en/>



- จากนั้น เข้าไปที่ <https://notify-bot.line.me/my/> เพื่อทำการ สร้าง Line Token ขึ้นมา ให้กดปุ่ม Generate token








- ตั้งชื่อ Token อะไรก็ได้ พร้อมทั้งเลือก กลุ่มใน Line ที่ต้องการให้มีการแจ้งเตือนไปยังกลุ่ม

### Generate token

Please enter a token name to be displayed before each notification.

**Name**

Select a chat to send notifications to.

-  **1-on-1 chat with LINE Notify** **LINE to**
-  Bluechara
-  Opt.Chem.
-  The gangster
-  ม.6/3 พันธนาวิศวรรรณศาสตร์

Note: Revealing your personal access token can allow a third party to obtain the names of your connected chats as well as your profile name.

**Create**  
**Generate token**

**Your token is:**

**94H0Lqvi4ybwhTlb74hIR8v0bZycZyv81DvDyMg**

If you leave this page, you will not be able to view your newly generated token again. Please copy the token before leaving this page.

**Copy** **Close**

### 3. Code for Test

```
#include <TridentTD_LineNotify.h>

#define SSID          "Test1234"
#define PASSWORD      "MaiMeeJingJing"
#define LINE_TOKEN     "xxxxxxxxxxxxxxxxxxxxxxxx"

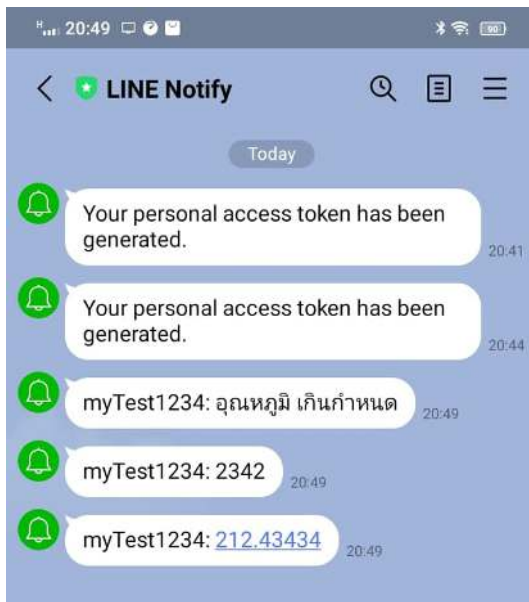
void setup() {
  Serial.begin(115200); Serial.println();
  Serial.println(LINE.getVersion());

  WiFi.begin(SSID, PASSWORD);
  Serial.printf("WiFi connecting to %s\n", SSID);
  while(WiFi.status() != WL_CONNECTED) { Serial.print("."); delay(400); }
  Serial.printf("\nWiFi connected\nIP : ");
  Serial.println(WiFi.localIP());

  LINE.setToken(LINE_TOKEN);

  LINE.notify("อุณห์ภูมิ เกินกำหนด");
  LINE.notify(2342); // จำนวนเต็ม
  LINE.notify(212.43434,5); // จำนวนจริง แสดง 5 หลัก

  void loop() {
    delay(1);
  }
}
```



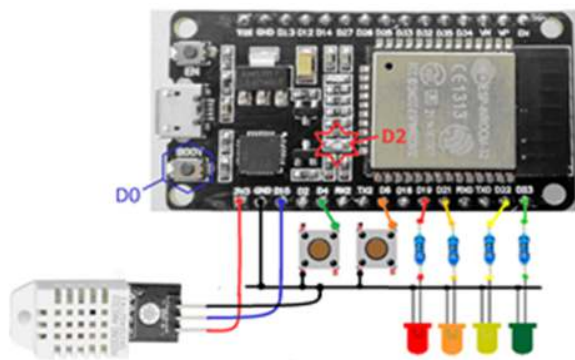
แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต

IoT Approaches to Manufacturing System

ชื่อ-สกุล :

5/5. คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz\_401 – Ubidots: Monitor DHT22, Monitor Digital Switch and Control 4 LED



< Test Code >

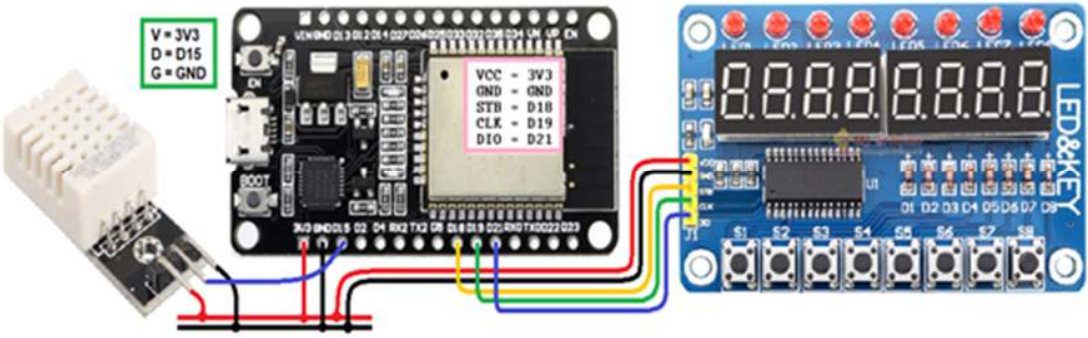
รูปการต่อวงจร – 1

รูปการต่อวงจร – 2

รูปหน้าจอ Ubidot Dashboard

### Quiz\_402 – Ubidots: Monitor DHT22 with TM1638 Display and LINE Alert

- ส่งข้อมูลอุณหภูมิไปยัง Ubidots
- หากอุณหภูมิที่อ่านได้เกิน 28°C ให้แจ้งเตือนผ่าน LINE และบอกด้วยว่าอุณหภูมิเท่าใด
- แสดงอุณหภูมิที่ 7\_Segment Display TM1638 Board


<p>&lt; Test Code &gt;</p>
<p>รูปการต่อวงจร – 1</p>
<p>รูปการต่อวงจร – 2</p>
<p>รูปหน้าจอ Ubidot Dashboard</p>
<p>รูปหน้าจอ LINE ผลการทดสอบ</p>