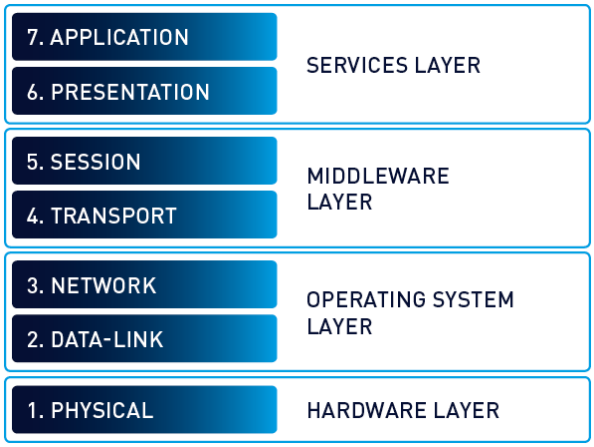
OSI Model and Industrial Protocol

1 OSI Model -- Open System Interconnection Model

https://www.ablenet.co.th/2020/08/28/what-is-osi-model/

OSI Model (Open Systems Interconnection Model) คือรูปแบบการรับส่งข้อมูลระหว่างอุปกรณ์อิเล็กทรอนิกส์ผ่านระบบเครือข่าย เป็นตัวกำหนดรูปแบบของผู้ส่งข้อมูล (Sender) และ ผู้รับข้อมูล (Receiver) จะแบ่งการทำงานออกเป็น 7 Layers โดย Layer 4-7 จะเน้นไปที่การติดต่อกับ User ผ่าน Software เป็นหลัก ส่วน Layer 1-3 จะเน้นที่การสื่อสารในระดับ Hardware เป็นหลัก โดยแต่ละ Layer จะมีบทบาท, หน้าที่และหลักการทำงานที่แตกต่างกันแต่จะทำงานร่วมกับ Layer ที่อยู่ติดกันดังนี้

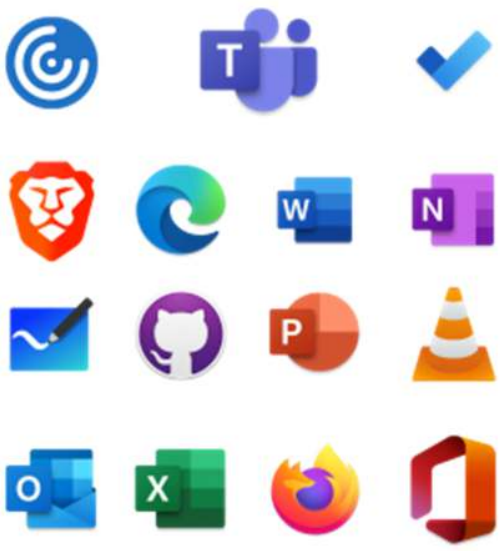


(ที่มา: https://blog.paessler.com)

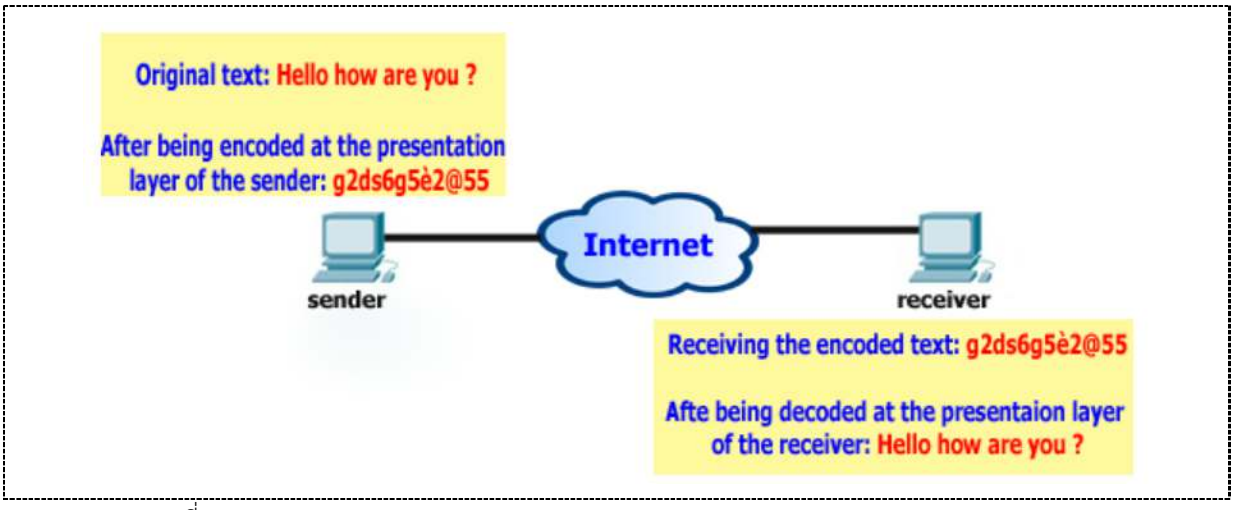
โดยทั้ง 7 Layers จะถูกแบ่งตามลักษณะการทำงานได้เป็น 2 กลุ่มใหญ่ๆ คือ

1. Application-oriented layers (Layer 4-7) คือ กลุ่มของ Layers ที่ใช้สื่อสารการเชื่อมต่อข้อมูลระหว่าง Sender และ Receiver เข้ากับ Application ต่างๆ โดยจะเกี่ยวข้องกับ Software เป็นหลัก

Layer 7: Application Layer เป็น Layer ที่อยู่ใกล้กับ Users มากที่สุด โดยจะใช้ Software ในการ Interact กับ Users โดยตรง เช่น แอพพลิเคชั่นจำพวก Browser, Line, etc



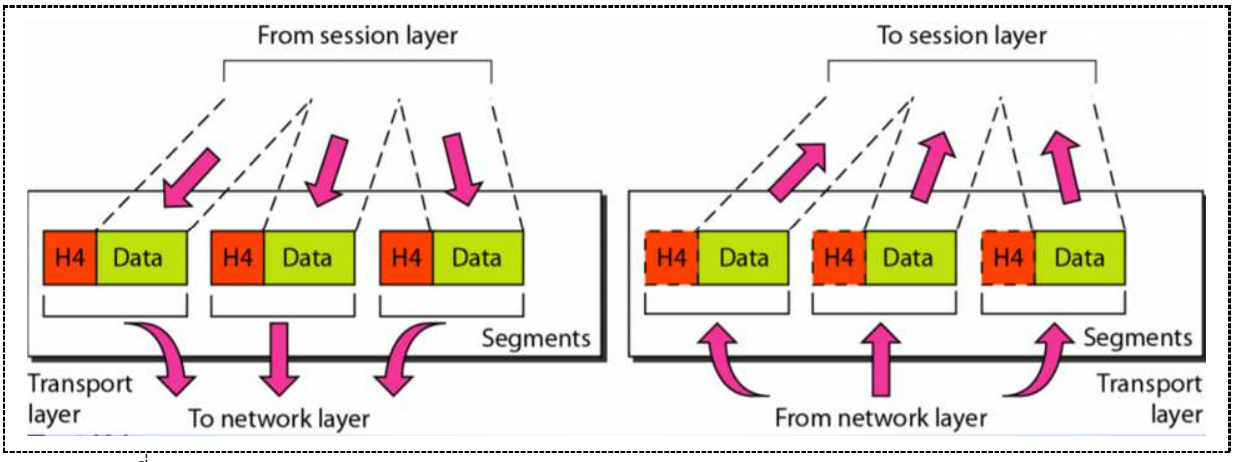
Layer 6: Presentation Layer เป็น Layer ที่ใช้ในการ Translate ข้อมูลจาก/ไปยัง Application layer เช่น Sender พิมพ์ข้อความว่า “Hello, how are you?” layer นี้จะทำการแปลงข้อความเหล่านั้นเป็นรหัส และให้ Presentation layer จากฝั่ง Receiver เป็นตัวแปลงรหัสเหล่านั้นให้กลับมาเป็นข้อความ “Hello, how are you?” ให้ Receiver ได้รับ



(ที่มา: http://www.cnt4all.com/2016/07/05-presentation-layer-layer-6-of-osi.htm)

Layer 5: Session Layer เป็น Layer ที่มีการ Sync เงื่อนไขการใช้งานระหว่างเครื่องต้นทางกับเครื่องปลายทาง เช่น User ต้องการขอใช้บริการบางอย่างจาก Server เป็นเวลา 20 นาที ผ่าน port 99, Server ก็จะส่งข้อความอนุญาตให้ User ดังกล่าวใช้บริการผ่าน port 99 ได้ เป็นเวลา 20 นาที หาก Session ที่ขอใช้งานเกิดหมดเวลา ก็จะไม่สามารถใช้บริการนั้นได้

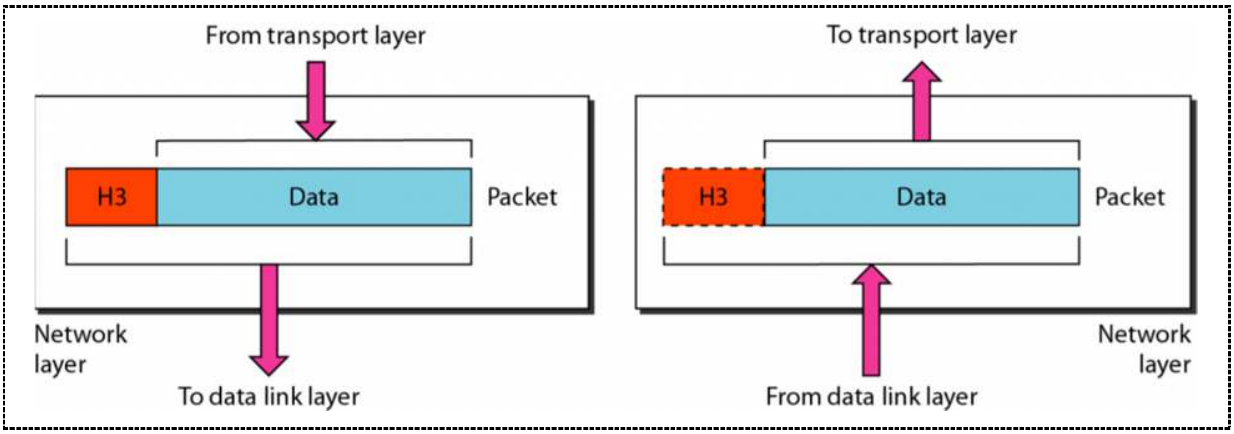
Layer 4: Transport Layer เป็น Layer ที่จะควบคุมการขนส่งข้อมูลจาก Sender ไปยัง Receiver หรือจาก Receiver ไปยัง Sender เมื่อเกิดการรับส่งข้อมูล ตัว Transport layer จะทำการแบ่งชิ้นส่วนข้อความดังกล่าวเป็นชิ้นเล็กๆหลายๆชิ้นเรียกว่า “Segment” และทำการ Add L4 Header (ประกอบด้วย Protocol ที่ใช้, Source Port และ Destination Port) เข้าไปบน Segments แต่ละชิ้น เพื่อให้ง่ายต่อการส่งและตรวจสอบความถูกต้อง โดยวิธีการนี้เรียกว่า Segmentation



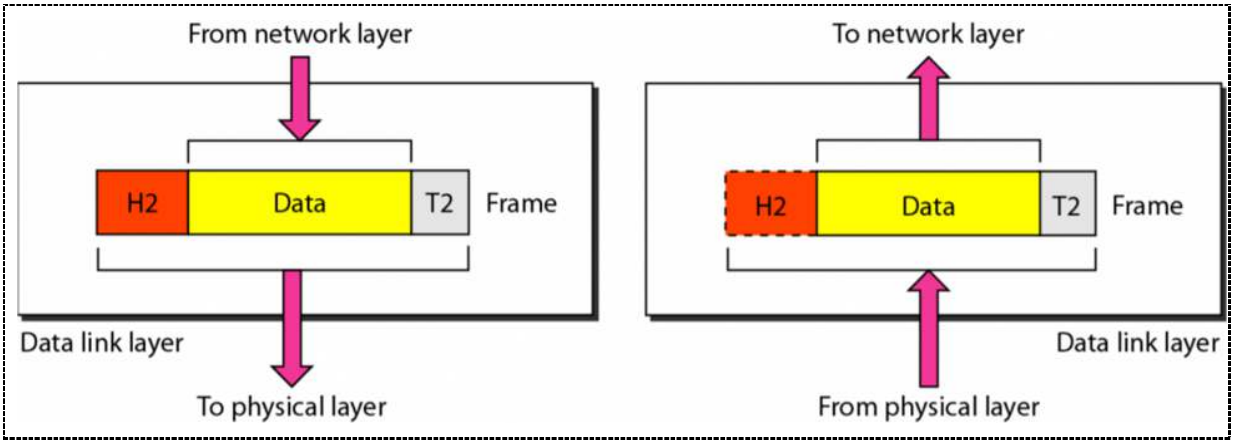
(ที่มา: https://apipong.weebly.com/3623363635943634360736373656362636293609.html)

2. Network-dependent Layers (Layer 1-3) คือ กลุ่มของ Layers ที่ทำหน้าที่เชื่อมต่อคอมพิวเตอร์ของทั้ง Senders และ Receivers ผ่านระบบเครือข่ายทั้งแบบมีสายและไร้สาย โดยจะเกี่ยวข้องกับ Hardware เป็นหลัก ซึ่งสำหรับบุคลากรที่ทำงานสาย Network จะเน้นศึกษาที่ Layers เหล่านี้

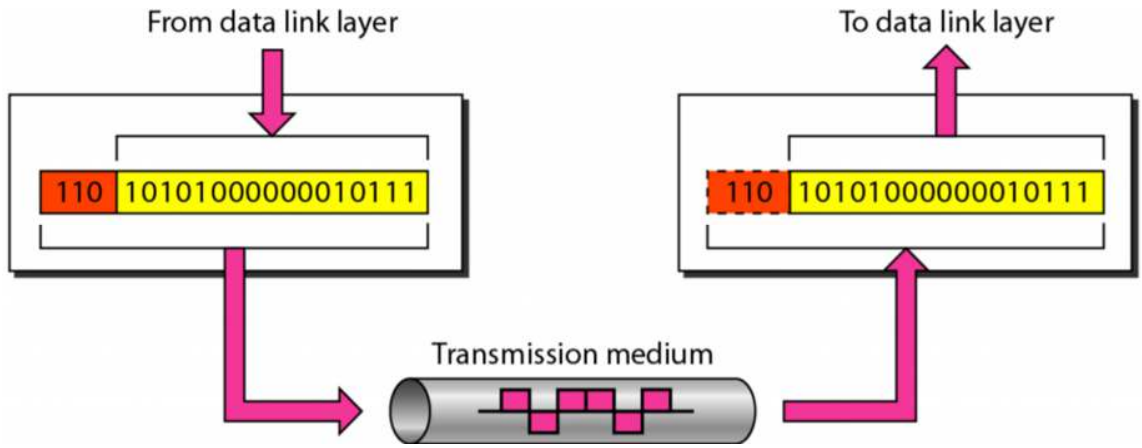
Layer 3: Network Layer เป็น Layer ที่ทำการสร้างช่องทางการเชื่อมต่อระหว่าง Network ของ Sender และ Receiver เข้าด้วยกันผ่าน IP Address รวมถึง โดย Layer นี้จะรับ Segments จาก Transport Layer มา Add L3 Header (ประกอบด้วย Source IP และ Destination IP) และตั้งชื่อให้ใหม่ว่า “Packet” โดยอุปกรณ์ที่ทำหน้าที่บน Layer3 ได้แก่ Router, L3 Switch(Multilayer Swith), Wireless Router เป็นต้น



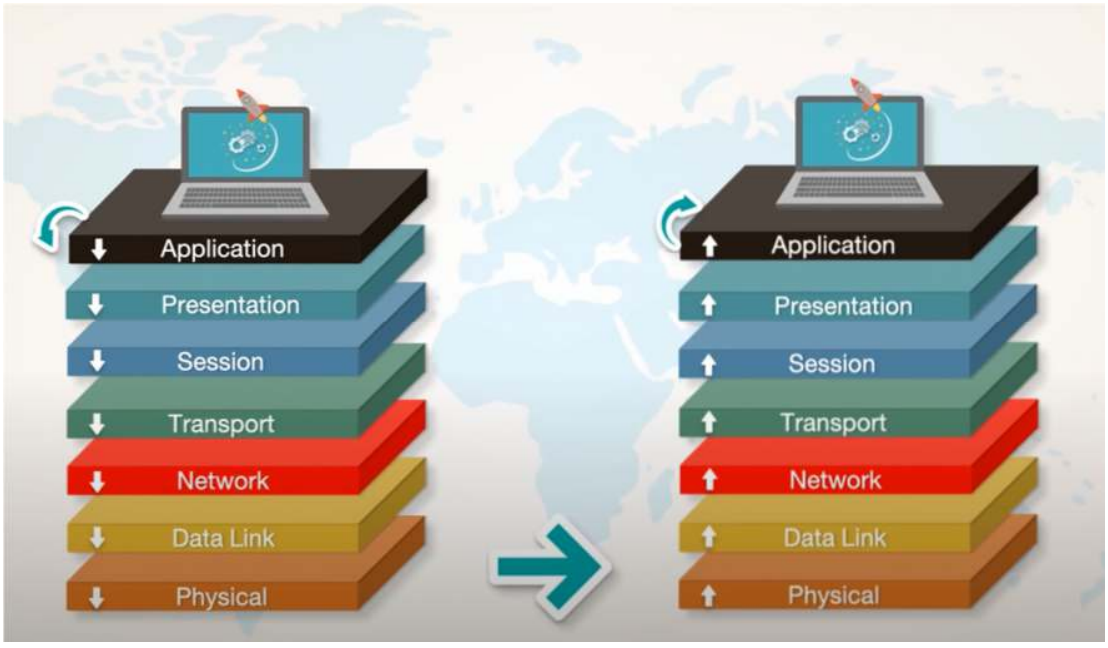
Layer 2: Data link Layer เป็น Layer ที่ทำการเชื่อมต่อข้อมูลแบบ node to node เช่น PC-Switch, Swith-Switch หรือ Switch-Router เป็นต้น โดยจะใช้ MAC Address ส่วนมากจะใช้สาย UTP เป็นตัวเชื่อมต่ออุปกรณ์เหล่านี้เข้าด้วยกัน โดย Layer นี้จะรับ Packet จาก Network Layer มาทำการ Add L2 Header และ L2 Trailer (ประกอบด้วย Source MAC, Destination MAC, Tag VLAN, etc) และเรียกชื่อใหม่ว่า “Frame” โดยอุปกรณ์ที่ทำหน้าที่บน Layer2 ได้แก่ Switch, Bridge



Layer 1: Physical Layer เป็น Layer ที่ทำการนำ Frame ข้อมูลจาก Data Link Layer ส่งระหว่างอุปกรณ์ Network ผ่านตัวกลาง เช่น สาย UTP, สาย Fiber optic โดยเราเรียกสิ่งที่ส่งผ่านตัวกลางเหล่านี้ว่า “Bits”หรือ “Bytes” (8 Bits = 1 Byte)

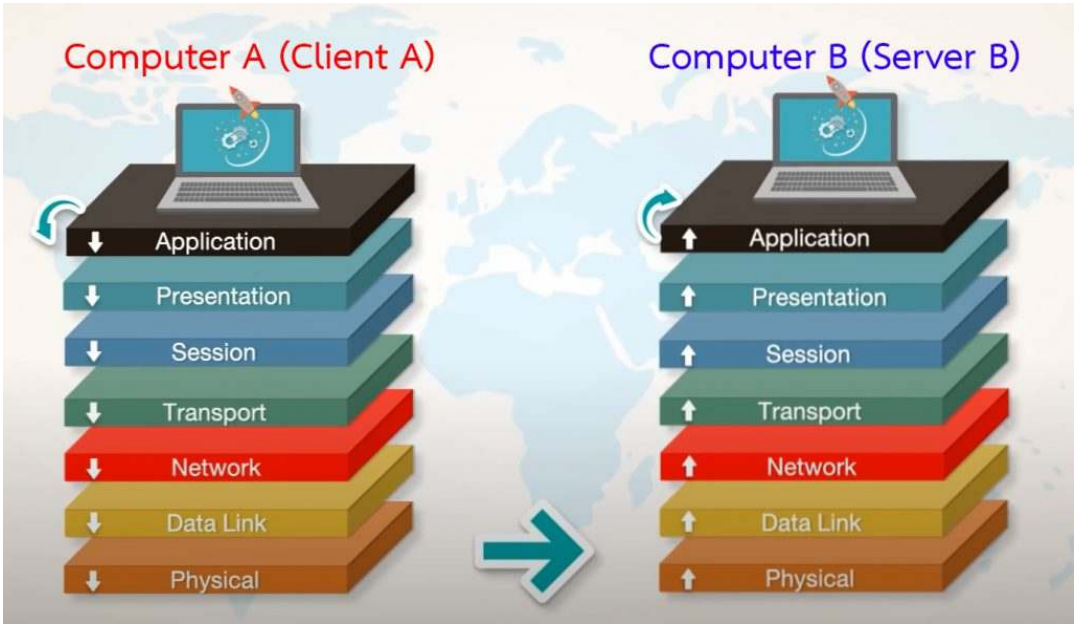


โดยทั้ง 7 Layers มีหลักการทำงานที่สัมพันธ์กัน คล้ายกับการขึ้นลงบันได โดยจะเริ่มจากชั้น Application Layer จากฝั่ง Sender และจบที่ Application Layer จากฝั่ง Receiver



ที่มา (https://www.youtube.com/watch?v=gvXKtgLn-28)

ตัวอย่างเช่น Computer A (Client A) ต้องการติดต่อขอใช้บริการกับ Computer B (Server B)



———————————— Sender A ————————————

<Sender A> Application Layer: ผู้ใช้กรอกข้อมูล Username/Password จากนั้นกด Enter

<Sender A> Presentation Layer: Encode Username/Password ไปเป็น Data

<Sender A> Session Layer: รอรับการ Synchronize จากปลายทาง

<Sender A> Transport Layer: นำ Data มาแบ่งเป็นชิ้นเล็กๆจากนั้น แนบ L4 Header (มี Protocol, Source port, Destination port เป็นส่วนประกอบ) ลงไปเรียกแต่ละชิ้นว่า “Segment”

<Sender A> Network Layer: นำ Segments มาแนบ L3 Header (มี Source IP, Destination IP เป็นส่วนประกอบ) เรียกว่า “Packet”

<Sender A> Data Link Layer: นำ Packet มาแนบ L2 Header และ L2 Trailer (มี Source MAC, Destination MAC, ฯลฯ เป็นส่วนประกอบ) เรียกว่า “Frame”

<Sender A> Physical Layer: นำ Frame ส่งผ่านสายนำข้อมูล เรียกว่า bits, Bytes (8 bits = 1 Bytes)

———————————— Receiver B ————————————

<Receiver B> Physical Layer: รับ bits, Bytes ผ่านสายนำข้อมูล

<Receiver B> Data Link Layer: รับ Frame มาทำการ แกะ L2 Header, L2 Trailer ออก เพื่อตรวจสอบ Source MAC, Destination MAC, ฯลฯ

<Receiver B> Network Layer: รับ Packet มาทำการ แกะ L3 Header ออก เพื่อตรวจสอบ Source IP, Destination IP

<Receiver B> Transport Layer: รับ Segment มาทำการ แกะ L4 Header ออก เพื่อตรวจสอบ Protocol, Source Port, Destination Port

<Receiver B> Session Layer: รอรับการ Synchronize จากต้นทาง

<Receiver B> Presentation Layer: ทำการ Decode Data ไปเป็น Username/Password เพื่อตรวจสอบในฐานข้อมูลว่า มีข้อมูล User ดังกล่าวหรือไม่

<Receiver B> Application Layer: รายงานผลการข้อใช้บริการจาก Client A ให้ Server B ทราบ

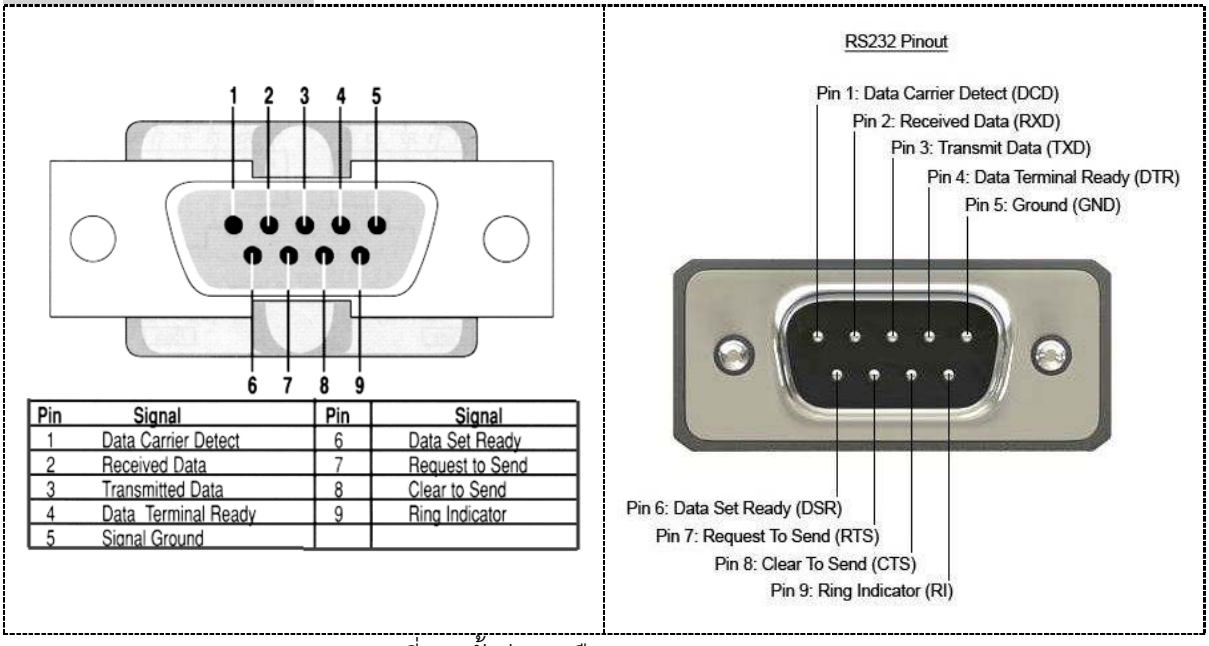
————————————————————————————-

** จากนั้น Server B จะเปลี่ยนสถานะตัวเองจาก Receiver B ไปเป็น Sender B เพื่อส่ง รายละเอียดการให้บริการกลับไปให้ Client A (Sender A ไปเป็น Receiver A) ผ่านทาง OSI Model ทั้ง 7 Layers (ครั้งนี้ Session Layer จะทำการ ส่งเงื่อนไขการให้บริการเพื่อ Sync ระหว่าง Server B และ Client A)

2 การสื่อสารข้อมูลแบบ RS232/RS485/RS422

https://www.omi.co.th/th/article/rs232
https://www.omi.co.th/th/article/rs485

2.1 การสื่อสารแบบ RS232



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

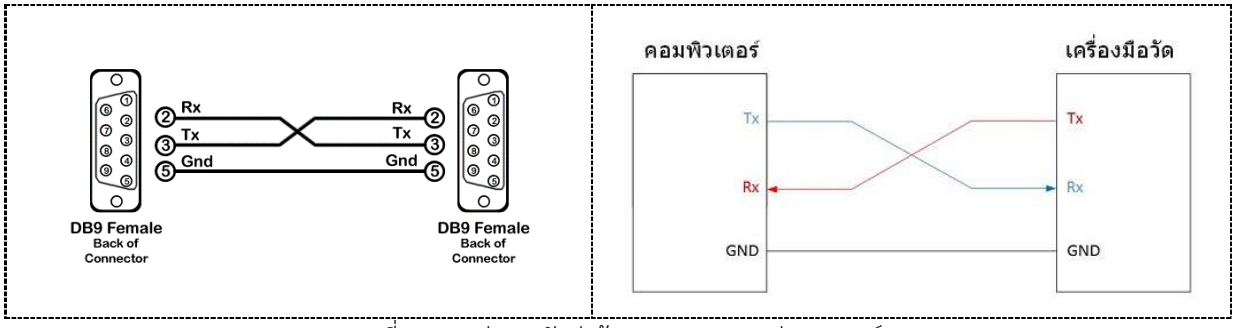
รูปที่ 10 ขั้วต่อสารสื่อสารอนุกรมแบบ RS232

RS232 (ย่อมาจาก: Recommended Standard no. 232) คือมาตรฐานการสื่อสารข้อมูลดิจิตอลแบบอนุกรม (serial communication) ซึ่งถูกกำหนดขึ้นครั้งแรกในปี ค.ศ. 1960 โดย EIA (Electronic Industries Association) หรือ สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ของอเมริกา ซึ่งในยุคแรก RS232 เป็นที่นิยมมากขนาดที่คอมพิวเตอร์ทุกเครื่องจะต้องมี Serial port สำหรับการสื่อสารมาตรฐานนี้และเชื่อว่าคอมพิวเตอร์ของผู้ใช้หลายๆท่านก็ยังมี Port เชื่อมต่อนี้อยู่ แต่ในปัจจุบันได้มี USB ซึ่งเป็นมาตรฐานสื่อสารที่รับ/ส่งข้อมูลได้เร็วกว่าเข้ามาแทนที่ ทำให้มาตรฐานการสื่อสารอย่าง RS232 ก็ค่อยๆมีอุปกรณ์ที่รองรับน้อยลงเรื่อยๆ ตามกาลเวลา

หลักการทำงานของ RS232

มาตรฐาน RS232 เป็นมาตรฐานที่รับ/ส่งข้อมูลแบบ Full duplex หรือจะให้พูดง่ายๆคือสามารถรับและส่งข้อมูลได้พร้อมกันทั้งคู่ในเวลาเดียวกัน โดยการรับ/ส่งข้อมูลนั้นจะใช้สายไฟทั้งหมด 3 เส้น ได้แก่

- Tx (Transmit data) คือ สายส่งข้อมูล ซึ่งสายเส้นนี้จะมีหน้าที่ในการส่งข้อมูลเท่านั้น
- Rx (Receive data) คือ สายรับข้อมูล ซึ่งสายเส้นนี้จะมีหน้าที่ในการรับข้อมูลเท่านั้น
- GND (Signal ground) คือ สายกราวด์ เป็นสายเทียบหรืออ้างอิงแรงดันไฟฟ้า 0V



รูปที่ 11 การต่อสารรับส่งข้อมูล RS232 ระหว่างอุปกรณ์

จากภาพเป็นตัวอย่างการเชื่อมต่อแบบ RS232 ของเครื่องมือวัดอุตสาหกรรมกับคอมพิวเตอร์ เพื่อตั้งค่าเครื่องมือวัดผ่าน Software โดย

- Tx (เครื่องวัด) จะถูกต่อเข้าที่ Rx (คอม) เพื่อส่งข้อมูลจากเครื่องวัดไปยังตัวรับของคอมพิวเตอร์
- Rx (เครื่องวัด) จะถูกต่อเข้าที่ Tx (คอม) เพื่อรับข้อมูลที่ถูกส่งมาจากคอมพิวเตอร์
- GND (เครื่องวัด) จะถูกต่อเข้าที่ GND (คอม) เพื่อเทียบสัญญาณแรงดัน 0V

ตารางเปรียบเทียบคุณสมบัติของ RS232

ตารางนี้เป็นตารางการเปรียบเทียบคุณสมบัติต่างๆ ของ RS232, RS423, RS422 และRS485

ตารางเปรียบเทียบคุณสมบัติ RS232 RS423 RS422 RS485

	RS232	RS423	RS422	RS485
Differential	no	no	yes	yes
Max number of drivers Max number of receivers	1 1	1 10	1 10	32 32
Modes of operation	half duplex full duplex	half duplex	half duplex	half duplex
Network topology	point-to-point	multidrop	multidrop	multipoint
Max distance (acc. standard)	15 m	1200 m	1200 m	1200 m
Max speed at 12 m Max speed at 1200 m	20 kbs (1 kbs)	100 kbs 1 kbs	10 Mbs 100 kbs	35 Mbs 100 kbs
Max slew rate	30 V/µs	adjustable	n/a	n/a
Receiver input resistance	3..7 kΩ	≧ 4 kΩ	≧ 4 kΩ	≧ 12 kΩ
Driver load impedance	3..7 kΩ	≧ 450 Ω	100 Ω	54 Ω
Receiver input sensitivity	±3 V	±200 mV	±200 mV	±200 mV
Receiver input range	±15 V	±12 V	±10 V	-7..12 V
Max driver output voltage	±25 V	±6 V	±6 V	-7..12 V
Min driver output voltage (with load)	±5 V	±3.6 V	±2.0 V	±1.5 V

รูปที่ 12 คุณสมบัติต่างๆ ของ RS232, RS423, RS422 และRS485

ข้อดีของสัญญาณ RS232

จากที่กล่าวมาข้างต้นการสื่อสารแบบ RS232 ถูกคิดค้นมาตั้งแต่ปี 1960 ซึ่งถือว่านานมาก จากการถือกำเนิดมาอย่างยาวนานนั้นก็ทำให้ข้อดีเหลือน้อยลงไปทุกทีเพราะมีการสื่อสารรูปแบบใหม่ที่ถูกพัฒนาให้ดีกว่าเกิดขึ้นอยู่ทุกวัน แต่ถึงกระนั้น RS232 ก็ยังพอมีข้อดีหลงเหลืออยู่ ซึ่งจะขออธิบายเป็นข้อๆดังนี้

- ความคุ้นเคยของผู้ใช้

  ปัจจุบันรูปแบบการสื่อสารได้ถูกพัฒนามาอย่างยาวไกลจนทิ้ง RS232 แบบไม่เห็นฝุ่นและการคงอยู่ของ RS232 จะเป็นไปได้ก็เพราะตัวผู้ใช้ยังคงใช้งานมันอยู่นั่นเองและสาเหตุหลักที่ยังมีการใช้อยู่ก็คงหนีไม่พ้นสิ่งที่เรียกว่า "ความคุ้นเคย" เนื่องจากการใช้งานสัญญาณดิจิตอลต้องมีการเขียนโปรแกรม (ยกเว้นซื้อสำเร็จรูป) และการเขียนโปรแกรมนั้นต้องมีความรู้เรื่องสัญญาณนั้นๆ ด้วยถึงจะเขียนโปรแกรมได้ ซึ่งหากผู้ใช้มีความรู้เกี่ยวกับ RS232 แล้ว จึงไม่ใช่เรื่องแปลกที่จะเลือกใช้สัญญาณนี้

- มีอุปกรณ์รองรับการใช้งาน

  RS232 เป็นระบบที่ถูกคิดค้นมาตั้งแต่ปี 1960 และเป็นที่นิยมในยุคแรกซึ่งมีข้อดีคือ มีอุปกรณ์ที่รองรับเยอะ การสื่อสารแบบ RS232 เป็นการสื่อสารที่มีอยู่ในเมนบอร์ดคอมพิวเตอร์แทบทุกรุ่น ซึ่งคนทั่วไปจะรู้จักกันในชื่อ Serial port ซึ่งทำให้การสื่อสารแบบ RS232 ไม่จำเป็นต้องใช้ Converter (ตัวแปลงสัญญาณ) ในการเชื่อมต่อกับคอมพิวเตอร์ ซึ่งต่างจากมาตรฐานใหม่อย่าง RS422, RS485 ที่ถึงแม้จะมีข้อดีที่มากกว่าแต่ก็ต้องใช้ Converter ในการแปลงสัญญาณอยู่ดี แต่ข้อดีข้อนี้อาจอยู่ได้อีกไม่นาน เพราะปัจจุบันเมนบอร์ดรุ่นใหม่ๆ ได้นำ Serial port ออกจากเมนบอร์ดและเพิ่ม Port การสื่อสารน้องใหม่ที่กำลังเป็นที่นิยมเข้าไปแทนนั้นคือการสื่อสารแบบ USB ซึ่งทำให้การสื่อสารรุ่นเก่าอย่าง RS232 ค่อยๆ เลือนหายไปตามการเวลา

ข้อเสียของสัญญาณ RS232

ปัจจุบันได้มีการพัฒนาการสื่อสารข้อมูลรูปแบบใหม่ขึ้นมามากมาย RS232 ซึ่งเป็นการสื่อสารรุ่นเก่าก็ย่อมมีข้อเสียอยู่เช่นกัน ซึ่งจะขออธิบายเป็นข้อๆดังนี้

- ปัญหาการส่งสัญญาณในระยะไกล

  RS232 สามารถรับรับ/ส่งข้อมูลที่ความเร็วสูงสุด 19.2 kbit/s ได้ที่ระยะ 15 เมตร ซึ่งแตกต่างกันมากเมื่อเทียบกับการสื่อสารมาตรฐานใหม่อย่าง RS485 ซึ่งสามารถรับ/ส่งข้อมูลได้ไกลถึง 1,200 เมตร ที่ความเร็ว 100 kbit/s เนื่องจากการสื่อสารแบบ RS232 นั้นเป็นระบบที่ง่ายต่อการถูกสัญญาณรบกวน (Noise) เข้าแทรกแซง ทำให้ระยะการสื่อสารของ RS232 ไม่สามารถส่งในระยะไกลได้ แต่หากมองดูเผินๆแล้ว 15 เมตร อาจจะถือว่าไกลมากสำหรับการใช้งานทั่วๆไป แต่ในโรงงานอุตสาหกรรมแล้ว การส่งข้อมูลในเครื่องมือวัดอุตสาหกรรมหรือเครื่องมือทางวิศวกรรมมายังห้องควบคุมด้วยระยะ 15 เมตรนั้นถือว่าสั้นมากๆ เมื่อเทียบกับขนาดของโรงงาน

- รับ/ส่งข้อมูลได้เฉพาะแบบ 1 ต่อ 1

  อีกหนึ่งปัญหาของ RS232 คือไม่สามารถส่งข้อมูลจากอุปกรณ์พร้อมกันหลายๆตัวมายังคอมพิวเตอร์ได้ โดยทำได้เพียงแค่ส่งข้อมูลมาที่คอมพิวเตอร์ทีละตัวแบบ 1 ต่อ 1 ซึ่งแตกต่างกันมากเมื่อเทียบกับการสื่อสารมาตรฐานใหม่อย่าง RS485 ซึ่งสามารถส่งข้อมูลจากอุปกรณ์พร้อมกันได้ถึง 32 ตัว

ความเร็วที่ล่าช้าในการรับ/ส่งข้อมูล

อีกจุดเปลี่ยนของการสื่อสารแบบ RS232 คือ ความล่าช้าในการรับ/ส่งข้อมูล นี่คือสาเหตุหลักที่ Microsoft เคยประกาศยกเลิกการสนับสนุน RS232 และถูกแทนที่ด้วยการสื่อสารแบบใหม่นั้นคือการสื่อสารแบบ USB ซึ่งเชื่อมต่อง่ายและรวดเร็วกว่า RS232 ถึงเกือบ 100 เท่าในยุคแรกๆ ซึ่งปัจจุบันอาจเร็วกว่านี้มาก แต่อย่างไรก็ตาม ระบบการรับ/ส่งข้อมูลในเครื่องมือวัดอุตสาหกรรมหรือเครื่องมือทางวิศวกรรมก็ยังใช้การสื่อสารแบบ RS232 อยู่ เพราะผู้ใช้จำนวนมากยังคงคุ้นชินและยังมีอุปกรณ์จำนวนมากในโรงงานอุตสาหกรรมที่ยังรองรับมาตรฐานนี้อยู่ บวกกับงานบางประเภทเป็นการสื่อสารแบบ 1 ต่อ 1 ในระยะสั้นเช่น การตั้งค่าเครื่องมือวัดอุตสาหกรรมโดยใช้ Note book ในการตั้งค่าตามจุดต่างๆที่เป็นปัญหา เป็นต้น

ความยาวสายเคเบิลสูงสุดของ RS232

ความยาวของสายเคเบิล RS232 เป็นอีกหนึ่งสิ่งที่ถูกกล่าวถึงมากที่สุดในโลก ซึ่งตัวมาตรฐานได้พูดไว้อย่างชัดเจนว่าความยาวสูงสุดของสายเคเบิล RS232 คือ 50 ฟุต (15 เมตร) หรือสายเคเบิลต้องมีค่า capacitance สูงสุดเท่ากับ 2,500 pF ซึ่งกฎข้อหลังนี้มักจะถูกลืม นั่นหมายความว่าการใช้สายเคเบิลที่มีค่า capacitance ต่ำๆ จะช่วยขยายระยะสายเคเบิลให้ไกลขึ้นได้

ความยาวสูงสุดของสายเคเบิล RS232 ที่ระบุในมาตรฐานเป็นความยาวที่จะช่วยให้สามารถรับ/ส่งข้อมูลได้ที่ความเร็วสูงสุด ถ้าความเร็วในการรับ/ส่งข้อมูลลดลง นั่นก็หมายความว่าความยาวสูงสุดของสายเคเบิลก็จะเพิ่มขึ้น ซึ่งทาง Texas Instruments ได้ทดลองในทางปฏิบัติเมื่อหลายปีก่อน โดยใช้ความเร็วในการส่งข้อมูลที่แตกต่างกันเพื่อหาความยาวสูงสุดของสายเคเบิล โปรดจำไว้ว่ามาตรฐาน RS232 เดิมได้รับการพัฒนาขึ้นสำหรับความเร็ว 20,000 bit/s ซึ่งหากลดความเร็วลงครึ่งนึงจะทำให้ความยาวสายเคเบิลเพิ่มขึ้นได้อีกถึง 10 เท่า เลยทีเดียว

ความยาวสายเคเบิล RS232 ตาม Texas Instruments

อัตราการถ่ายโอนข้อมูล	ความยาวสายเคเบิลสูงสุด (ฟุต)
19200	50
9600	500
4800	1000
2400	3000

รูปที่ 13 ความยาวสูงสุดของสายเคเบิล RS232

รหัสสัญญาณของ RS232

RS232 เป็นรูปแบบการส่งข้อมูลดิจิตอลรูปแบบหนึ่ง ซึ่งอย่างที่ทุกคนทราบกันดีว่าข้อมูลดิจิตอลจะประกอบด้วยตัวเลขเพียงสองตัว คือ 0 และ 1 เรียงต่อกันเป็นรหัสหรือชุดคำสั่งเพื่อสั่งงานอุปกรณ์ต่างๆ ซึ่ง RS232 จะใช้ ระดับของแรงดันไฟฟ้า เป็นตัวบอกว่าข้อมูลไหนคือ 0 และ 1 ตามตาราง

แรงดันไฟฟ้าของ RS-232

รหัส	ตัวส่งสัญญาณ (V)	ตัวรับสัญญาณ (V)
0	+5 ... +15	+3 ... +25
1	-5 ... -15	-3 ... -25
Undefined	-	-3 ... +3

รูปที่ 14 ข้อกำหนดทางไฟฟ้าของ RS232

- สัญญาณรบกวนที่เกิดขึ้น ในสายนำสัญญาณ มักจะมีแรงดันเป็นบวก เมื่อเทียบกับกราวน์
    - เพื่อป้องกันสัญญาณรบกวนนี้ จึงออกแบบแรงดัน ของโลจิก "1" เป็นลบ คืออยู่ในช่วง -3V ถึง -15V ส่วนแรงดันของโลจิก "0" อยู่ในช่วง +3V ถึง +15V
    - และสาเหตุที่ ระดับสัญญาณ ของ RS232 อยู่ในช่วง +15V ถึง -15V ก็เพื่อให้ต่อสายสัญญาณไปได้ไกลขึ้น
- ดังนั้นจึงจำเป็นจะต้องมีวงจรเปลี่ยนระดับแรงดันของ RS232 มาเป็นระดับแรงดันของ TTL

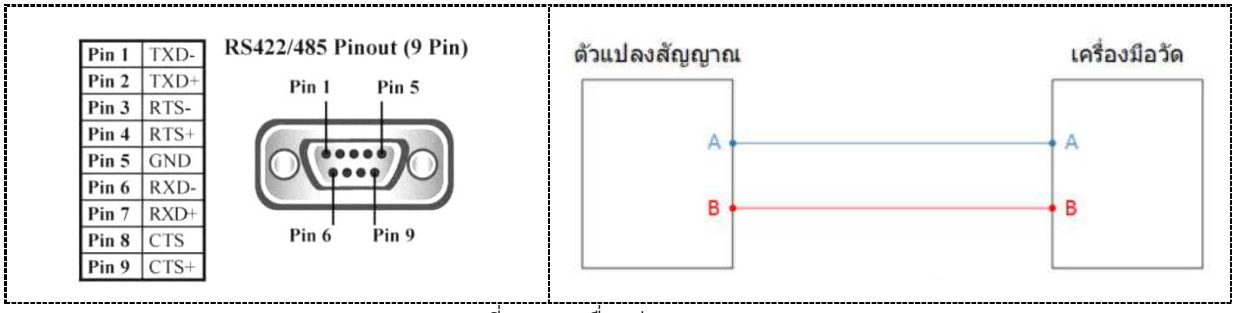
2.2 การสื่อสารแบบ RS485

RS485 (ย่อมาจาก: Recommended Standard no. 485) คือมาตรฐานการสื่อสารข้อมูลดิจิตอลแบบอนุกรม (serial communication) ซึ่งถูกกำหนดขึ้นครั้งแรกในปี ค.ศ. 1998 โดยความร่วมมือของ TIA (Telecommunications Industry Association) และ EIA (Electronic Industries Association) มาตรฐาน RS485 ถูกใช้อย่างแพร่หลายในโรงงานอุตสาหกรรม เนื่องจากสามารถส่งสัญญาณได้ไกลและยังสามารถส่งพร้อมๆ กันได้หลายจุด

ปกติแล้ว EIA จะตั้งชื่อมาตรฐานของตัวเองโดยการใช้คำนำหน้าว่า "RS" (Recommended Standard) แต่เนื่องจากมาตรฐานนี้เป็นความร่วมมือระหว่าง 2 หน่วยงาน คือ TIA และ EIA ทั้งสองหน่วยงานจึงตกลงเปลี่ยนจากคำว่า "RS" เป็น "TIA/EIA" แทนอย่างเป็นทางการ เพื่อระบุถึงแหล่งที่มาของมาตรฐานอย่างชัดเจน โดยต่อมาทาง EIA ก็ได้ยกเลิกมาตรฐานนี้และมาตรฐาน RS485 นี้ก็ได้ถูกพัฒนาอย่างต่อเนื่องจนถึงปัจจุบันโดย TIA ทำให้มาตรฐาน RS485 ถูกเปลี่ยนชื่อเป็น "TIA-485" อย่างเป็นทางการ แต่สุดท้ายเพราะความเคยชินทำให้วิศวกรทั่วโลกยังเรียกมาตรฐานการสื่อสารนี้ว่า RS485 เหมือนเดิม

หลักการทำงานของ RS485

มาตรฐาน RS485 เป็นมาตรฐานที่รับ/ส่งข้อมูลในแบบที่เรียกว่า Half duplex คือสามารถรับและส่งข้อมูลได้ทีละอย่างเท่านั้นไม่สามารถทำทั้งสองอย่างได้ในเวลาเดียวกัน ถ้าจะให้พูดแล้วเห็นภาพก็คงคล้ายๆลักษณะของวิทยุสื่อสารที่ต้องคอยสลับกันพูดทีละครั้ง



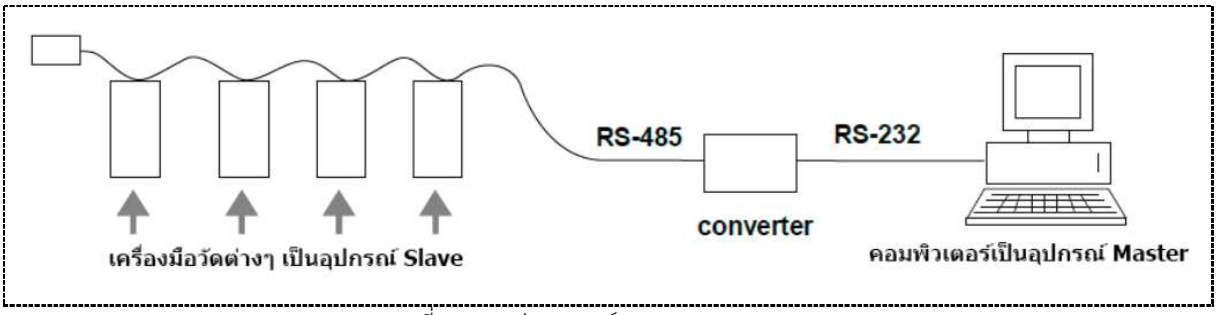
รูปที่ 15 การเชื่อมต่อแบบ RS485

สำหรับการรับ/ส่งข้อมูลดิจิตอลแบบ RS485 นั้น จะส่งข้อมูลโดยใช้สายไฟเพียงแค่ 2 เส้นคือ A และ B เป็นตัวบอกค่ารหัสดิจิตอล(Digital code) โดยใช้ความแตกต่างของแรงดันไฟฟ้าระหว่างขั้ว A และ B เป็นตัวบอกดังนี้

- เมื่อ Va – Vb ได้แรงดันไฟฟ้าน้อยกว่า -200 mV คือสัญญาณดิจิตอลเป็น 1
- เมื่อ Va – Vb ได้แรงดันไฟฟ้ามากกว่า +200 mV คือสัญญาณดิจิตอลเป็น 0

หลักการทำงานของ RS485 แบบ NETWORK

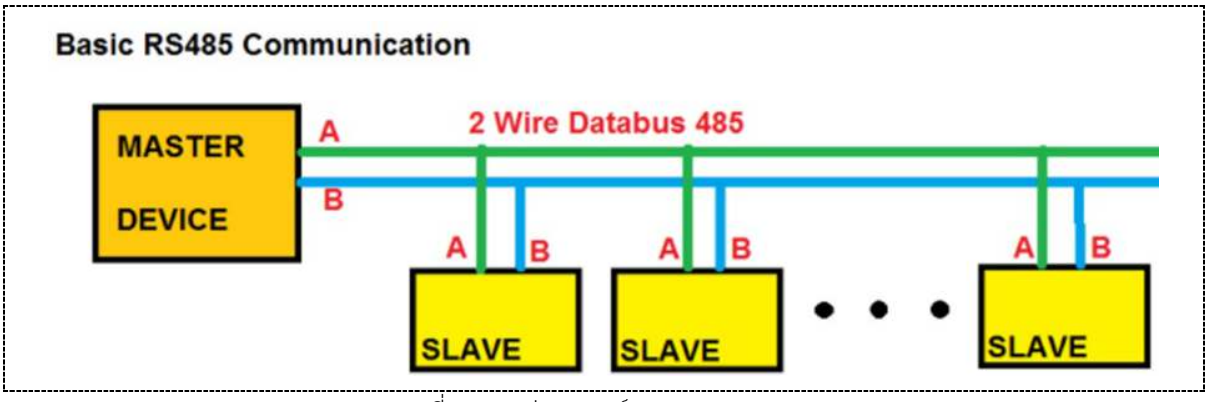
มาตรฐาน RS485 สามารถเชื่อมต่อการรับส่งข้อมูลแบบเครือข่าย (Network) โดยมีอุปกรณ์ในเครือข่ายได้สูงสุดถึง 32 ตัว ซึ่งในเครือข่ายนั้น จะต้องมีอุปกรณ์อยู่ 1 ตัว ทำหน้าที่คอยจัดคิวการสื่อสารในเครือข่าย ซึ่งเราจะเรียกอุปกรณ์ตัวนี้ว่า "Master" และอุปกรณ์ส่วนที่เหลือเราจะเรียกว่า "Slave" โดยที่ Slave แต่ละตัวจะมีหมายเลข Address ของตัวเอง และเมื่อตัว Master ต้องการสั่งการตัว Slave ตัว Master จะส่งชุดคำสั่งพร้อมระบุหมายเลข Address ไปยังอุปกรณ์ Slave ทุกตัว เมื่ออุปกรณ์ Slave ได้รับคำสั่งและคำสั่งนั้นมีหมายเลข Address ตรงกับตัวเอง อุปกรณ์ Slave ถึงจะทำตามคำสั่งของ Master เป็นลำดับไป



รูปที่ 16 การต่ออุปกรณ์ RS485 แบบ Network

จำนวนอุปกรณ์สูงสุดในเครือข่าย RS485

นี่เป็นอีกหนึ่งคำถามที่ผู้ใช้หน้าใหม่สงสัยมากที่สุดในโลก หากตามมาตรฐานแล้ว เครือข่าย RS485 สามารถมีอุปกรณ์ในระบบได้สูงสุด 32 ตัว เมื่ออุปกรณ์เหล่านั้นมีความต้านทานไฟฟ้าภายใน 12 kΩ แต่ปัจจุบันการออกแบบอุปกรณ์อิเล็กทรอนิกส์ได้พัฒนาจนมีความต้านทานไฟฟ้าภายในที่สูงมาก (ในหลัก MΩ) ซึ่งทำให้เครือข่าย RS485 สามารถมีอุปกรณ์ในระบบได้สูงสุดถึง 256 ตัว นอกจากนี้เครือข่าย RS485 ยังสามารถใช้ตัวขยายสัญญาณ (Repeater) สำหรับเพิ่มอุปกรณ์ในเครือข่ายได้ถึงหลายพันตัวและครอบคุมระยะหลายกิโลเมตรกันเลยทีเดียว



รูปที่ 17 การต่ออุปกรณ์ RS485 แบบ 2-Wire

ตารางเปรียบเทียบคุณสมบัติของ RS485

ตารางนี้เป็นตารางการเปรียบเทียบคุณสมบัติต่างๆ ของ RS485, RS232, RS423 และ RS422

ตารางเปรียบเทียบคุณสมบัติ RS232 RS423 RS422 RS485

	RS232	RS423	RS422	RS485
Differential	no	no	yes	yes
Max number of drivers Max number of receivers	1 1	1 10	1 10	32 32
Modes of operation	half duplex full duplex	half duplex	half duplex	half duplex
Network topology	point-to-point	multidrop	multidrop	multipoint
Max distance (acc. standard)	15 m	1200 m	1200 m	1200 m
Max speed at 12 m Max speed at 1200 m	20 kbs (1 kbs)	100 kbs 1 kbs	10 Mbs 100 kbs	35 Mbs 100 kbs
Max slew rate	30 V/µs	adjustable	n/a	n/a
Receiver input resistance	3..7 kΩ	≧ 4 kΩ	≧ 4 kΩ	≧ 12 kΩ
Driver load impedance	3..7 kΩ	≧ 450 Ω	100 Ω	54 Ω
Receiver input sensitivity	±3 V	±200 mV	±200 mV	±200 mV
Receiver input range	±15 V	±12 V	±10 V	-7..12 V
Max driver output voltage	±25 V	±6 V	±6 V	-7..12 V
Min driver output voltage (with load)	±5 V	±3.6 V	±2.0 V	±1.5 V

รูปที่ 18 การเปรียบเทียบคุณสมบัติต่างๆ ของ RS485, RS232, RS423 และ RS422

ข้อดีของสัญญาณ RS485

เป็นที่ทราบกันดีว่า RS485 เป็นมาตรฐานที่ถูกพัฒนาขึ้นมาเพื่อลบจุดด้อยของมาตรฐานรุ่นเก่าๆ อย่าง RS232 RS422 RS423 เป็นต้น ซึ่งข้อดีหลักๆของมาตรฐาน RS485 มีดังนี้

- สามารถส่งสัญญาณได้ไกล

  RS485 สามารถส่งสัญญาณได้ไกลสูงสุดถึง 1,200 เมตร ซึ่งถือว่าเป็นระยะทางที่ไกลมาก เพียงพอต่อการใช้งานในโรงงานอุตสาหกรรมอย่างแน่นอนและจะเห็นได้ชัดว่าระยะการส่งสัญญาณได้ถูกพัฒนาขึ้นมากจนทิ้งห่างมาตรฐานรุ่นเก่าอย่าง RS232 ที่สามารถส่งสัญญาณได้เพียง 15 เมตร เท่านั้น

- สามารถเชื่อมต่อเป็นเครือข่ายได้

  นอกจากจะส่งสัญญาณได้ไกลแล้ว RS485 ยังสามารถเชื่อมต่อเป็นเครือข่าย (Network) แบบ Multipoint ได้ด้วย ซึ่งสามารถเชื่อมต่ออุปกรณ์ในระบบได้สูงสุดถึง 32 ตัว ซึ่งสิ่งนี้ถือว่าเป็นอีกหนึ่งจุดเด่นของสัญญาณ RS485 เลยทีเดียว

- ประหยัดงบประมาณในการเดินสาย

  มาตรฐาน RS485 เป็นมาตรฐานที่ใช้สายไฟเพียง 2 เส้นในการรับส่งข้อมูล เมื่อเปรียบเทียบกับมาตรฐานรุ่นเก่าที่สามารถส่งสัญญาณในระยะเท่ากันอย่าง RS422 ที่ต้องใช้สายไฟถึง 4 เส้นในการรับส่งข้อมูล ซึ่งราคาสายเคเบิลแบบ 2 แกน จะถูกกว่าสายเคเบิลแบบ 4 แกน ถึงเกือบครึ่ง

ข้อเสียของสัญญาณ RS485

ถึงแม้ RS485 จะเป็นมาตรฐานที่ถูกพัฒนาขึ้นจนลบข้อด้อยที่มีอยู่ในมาตรฐานเก่าๆไปมากแล้วก็ตาม แต่ก็ไม่ใช่ว่าจะไม่มีข้อเสียอยู่เลย โดยข้อเสียหลักๆของ RS485 มีดังนี้

- ต้องใช้ตัวแปลงสัญญาณในการเชื่อมต่อกับคอมพิวเตอร์

  เนื่องจากปัจจุบันคอมพิวเตอร์ที่เราใช้กันอยู่นั้นไม่มี port เชื่อมต่อสัญญาณ RS485 โดยตรง จะมีก็แต่ USB หรือ RS232 เท่านั้น ฉนั้นหากเราจะเชื่อมต่ออุปกรณ์ที่ใช้ RS485 กับคอมพิวเตอร์นั้น เราต้องเสียงบประมาณเพิ่มขึ้นในการซื้อตัวแปลงสัญญาณ (Converter) เพื่อแปลงสัญญาณจาก RS485 เป็น USB หรือ RS232 ในการเชื่อมต่อนั้นเอง

- ความเร็วในการรับส่งข้อมูล

  ถึงแม้ RS485 จะถูกพัฒนาด้านความเร็วในการรับส่งข้อมูลขึ้นมากแล้วก็ตามเมื่อเทียบกับมาตรฐานเก่า แต่ก็ยังมีความล่าช้าอยู่เมื่อเชื่อมต่อในลักษณะเครือข่ายจำนวนมากๆ

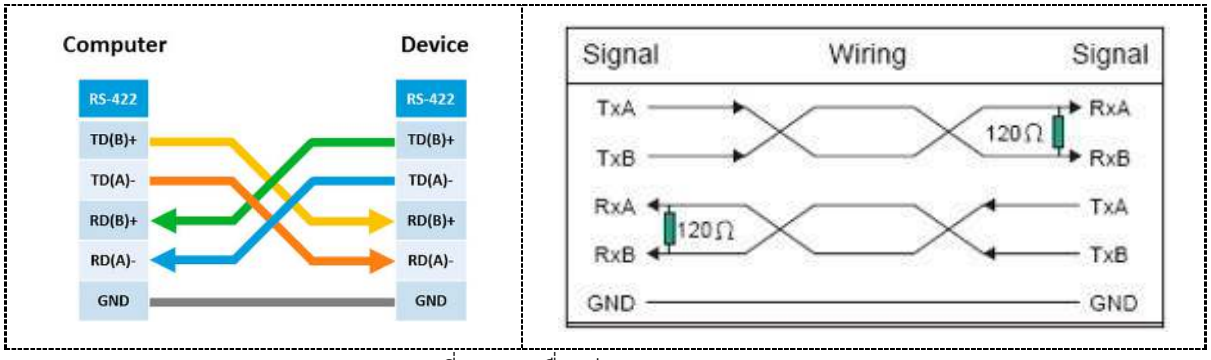
2.3 การสื่อสารแบบ RS422

RS-422A

มาตรฐาน RS-422-A กำหนดไว้ให้ใช้กับ Balanced Digital Circuit ซึ่งจะให้ความเร็วสูงขึ้นถึงประมาณ 10 Mbps ระยะห่างระหว่าง DTE (Data Communication Equipment) และ DCE (Data Terminal Equipment) ก็มากขึ้นด้วยทั้งยังทนทานต่อสภาพแวดล้อมที่มีสัญญาณรบกวนมากกว่า RS-232-C ด้วย

RS-422

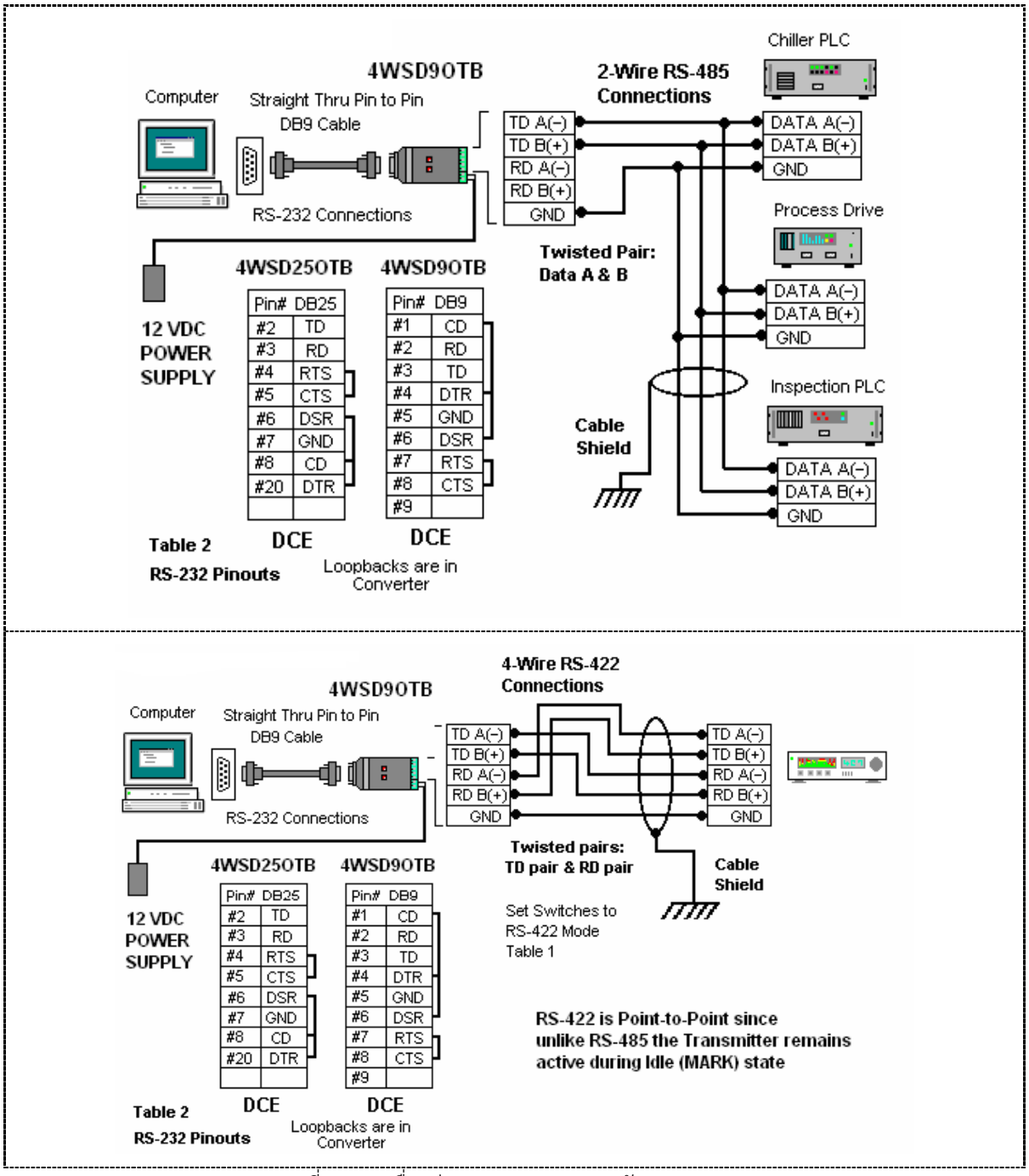
ถือเป็นตัวสำคัญของ RS-232 เป็นโปรโตคอลที่มืออาชีพต้องเลือกใช้ RS-422 ใช้ขั้วต่อแบบ DB 9 ขา หรือ DB 25 เหมือนกับ RS-232 สำหรับส่งคำสั่งและข่าวสารระหว่างเครื่องเล่นเทปและตัวควบคุม กรณีที่ใช้รหัสเวลา การใช้โปรโตคอล RS-422 จะควบคุมความแม่นยำในการเข้าหาภาพในระดับเฟรม (Frame-accurate) อย่างไรก็ตาม RS-422 มีใช้ในอุปกรณ์มืออาชีพเท่านั้น



รูปที่ 19 การเชื่อมต่อแบบ RS422 – 4Wire

RS422 สามารถที่จะรับส่งได้ในระยะทางที่ไกลกว่า RS232 ความเร็วในการส่งก็สูงกว่า RS232 ด้วย RS422 สามารถส่งได้ 10Mbps ที่ความยาว 50 ฟุต และ 100Kbps ที่ความยาว 4000 ฟุต เป็นแบบ full duplex หรือ haft duplex ในการใช้งานจะต้องแปลงสัญญาณ RS232 เป็น RS422 เสียก่อน โดยที่ RS422 จะมีขาที่ใช้ในการส่งข้อมูล Tx สองขา ซึ่งมีเฟสตรงกันข้าม 180 องศา คือขาหนึ่งเป็นลอจิก "1" อีกขาหนึ่งก็จะเป็นลอจิก "0" ทำให้กระแสที่ไหลวนในสายมีค่าคงที่ไม่ว่าจะเป็น 1 หรือ 0 ส่วนขาที่ใช้ในการ

รับข้อมูลของ RS422 Rx ก็มีสองขาด้วยเช่นกัน ดังนั้นจะเห็นว่า RS422 มีสายสัญญาณเพิ่มขึ้นอีกสองเส้นเป็น 4 เส้น แต่ส่งได้ไกลกว่า RS232 แต่ RS422 ก็จะต้องเสียค่าใช้จ่ายของสายสัญญาณเพิ่มขึ้น ก็เลยเกิด การส่งแบบ RS485 ซึ่งเป็นการลดสายสัญญาณออก 2 เส้น ทำให้ไม่สามารถที่จะรับส่งพร้อมๆ กันได้เรียกว่า haft duplex แต่มีหลักการรับและส่งแบบเดียวกับ RS422 ทำให้ระยะทางได้ไกลเท่ากัน



รูปที่ 20 การเชื่อมต่อแบบ 2Wire-RS485 กับ 4Wire-RS422

3 Industrial Communication Protocol

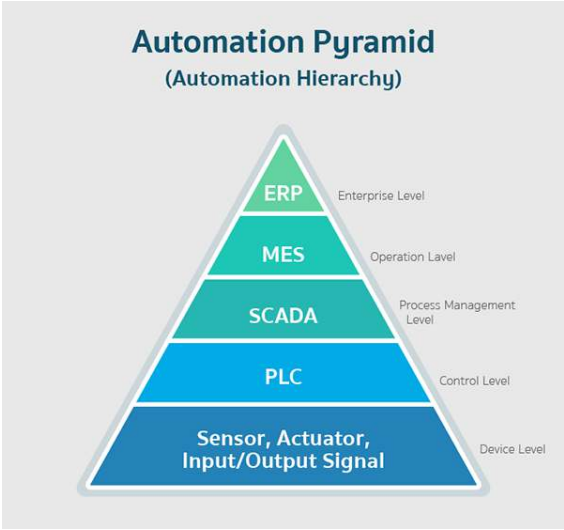
ระบบอัตโนมัติอุตสาหกรรม (Industrial Automation Systems) ทำหน้าที่เชื่อมโยงอุปกรณ์ต่าง ๆ ประกอบด้วย เครื่องกล อิเล็กทรอนิกส์ เซนเซอร์ ตัวขับเร้า (actuator) และคอมพิวเตอร์ที่ถูกออกแบบเพื่อลดหรือหลีกเลี่ยงการสั่งงานจากมนุษย์ และสามารถทำงานได้หลากหลายตามที่ต้องการได้ เพื่อให้เกิดประโยชน์ในด้านการลดค่าใช้จ่าย มีคุณภาพเพิ่มขึ้น และมีประสิทธิภาพโดยรวมมากขึ้น ตลอดจนมีความปลอดภัยในการทำงาน [1,2] การสื่อสารในอุตสาหกรรม (Industrial Communication) เป็นระบบสื่อสารระหว่างอุปกรณ์ต่าง ๆ ในระบบอัตโนมัติอุตสาหกรรม และมีส่วนสำคัญมากในการพัฒนาระบบอัตโนมัติอุตสาหกรรมให้มีความทันสมัย รายงานนี้เป็นการศึกษาการสื่อสารในอุตสาหกรรม Industrial Communications ที่มีโพรโทคอลหลักที่ใช้แพร่หลาย เช่น PROFINET, PROFIBUS, EtherCAT, EtherNet/IP, Modbus RTU, Modbus TCP เป็นต้น

3.1 การพัฒนาการสื่อสารในอุตสาหกรรม

ระบบสื่อสารในอุตสาหกรรมได้ผ่านการพัฒนาอย่างต่อเนื่องตั้งแต่เริ่มแรกในปีคริสต์ทศวรรษ 1980 [3,4] จนถึงปัจจุบัน โดยอาจแบ่งการพัฒนาที่สำคัญออกเป็น 3 ช่วง ได้แก่ 1. Classical Fieldbus system, 2.Industrial Ethernet และ 3.Wireless networks in automation โดยรายงานฉบับนี้จะกล่าวถึงเฉพาะสองลำดับแรกเท่านั้น

1) Classical Fieldbus System

การสื่อสารในอุตสาหกรรมในยุคเริ่มแรกจะใช้คำว่า Fieldbus system โดยคำนี้ถือกำเนิดจากกระบวนการในโรงงานเคมี และได้ถูกใช้มาจนถึงปัจจุบัน และนิยามของคำนี้จาก International Electrotechnical Commission (IEC) 61158 ระบุว่า “A fieldbus is a digital, serial, multidrop, data bus for communication with industrial control and instrumentation devices such as-but not limited to-transducers, actuators and local controllers.” และ Fieldbus ถูกสร้างขึ้นมาเพื่อทดแทนการเชื่อมต่อสายสัญญาณแบบ Star ซึ่งเป็นการเชื่อมสายสัญญาณจุดต่อจุดระหว่างขาเข้าและขาออกของอุปกรณ์ดิจิทัลหรืออนาล็อกกับอุปกรณ์ควบคุมศูนย์กลาง โดยเชื่อมต่อสายสัญญาณเป็นแบบ Line หรือ Bus ด้วยสายสัญญาณเส้นเดียวและมีขั้วต่อปลายทางแบบ DB 9 pins โดยมีอินเตอร์เฟซแบบ RS232 หรือ RS485 ระบบการสื่อสารแบบ Fieldbus ถูกให้ความสำคัญจากแนวคิดเรื่อง Automation Pyramid ซึ่งมีรายละเอียดใน The International Society of Automation (ISA) ในมาตรฐานเลขที่ 95 หรือ ISA-95 [5]



รูปที่ 21 Automation Pyramid (หรือ Automation Hierarchy)

มาตรฐาน ISA-95 พัฒนาต่อยอดจาก Purdue Reference Model for Computer-integrated Manufacturing (CIM) ซึ่งได้สร้างแนวคิดในช่วงคริสต์ทศวรรษ 1970 ในยุคที่อุปกรณ์ทางอิเล็กทรอนิกส์และคอมพิวเตอร์เพิ่งพัฒนาในยุคแรก และใน ISA-95 Part 1: Models and Terminology ได้กล่าวถึง Automation pyramid ดังแสดงในรูปที่ 1 ประกอบด้วย 5 ชั้น ตามลำดับดังนี้

เริ่มจากชั้นบนสุดคือชั้นที่ 4 บทบาทหรือหน้าที่ทางธุรกิจ ได้แก่ Enterprise Resource Planning (ERP), Material Requirement Planning (MRP), และ Supply Chain Management (SCM) เป็นต้น

ชั้นที่ 3 การจัดการควบคุมการผลิต ประกอบด้วยชั้นที่ 3 ลงไปจนถึงชั้นที่ 0 โดยชั้นที่ 3 ประกอบด้วย Manufacturing Execution System (MES), Laboratory Information Management System (LIMS), Quality Management (QM) เป็นต้น ชั้นที่2 ประกอบด้วย Supervisory Control And Data Acquisition (SCADA) ชั้นที่ 1 ประกอบด้วย Distributed Control System (DCS), Open Control System (OCS), และ Programmable Logic Controller (PLC) ที่ใช้เชื่อมต่อกับ ชั้นที่ 0 ประกอบด้วย sensors และ ตัวขับเร้า เป็นต้น ซึ่งเป็นกระบวนการทางฟิสิกส์และเคมีที่ระบบต้องการควบคุม โดยการเชื่อมต่อของการจัดการควบคุมการผลิตจะใช้เทคโนโลยี Fieldbus เชื่อมต่อแบบ serial ชนิด RS232 ที่เป็นการเชื่อมต่อจุดต่อจุดในช่วงแรก และต่อมาเป็นชนิด RS485 ซึ่งเชื่อมต่อแบบ bus network ที่มีความสามารถแบบ multi-dropper และ multi-point [4] และทำให้ความสามารถในการเชื่อมต่ออุปกรณ์ต่าง ๆ มีจำนวนมากสุดเท่ากับ 32 ชิ้น รวมถึงมีความยาวสายสัญญาณสูงสุดเท่ากับ 1,200 เมตร และมีความเร็วส่งผ่านข้อมูลสูงสุดเท่ากับ 10 Mbit/s ในยุคแรก และต่อมาอาศัยเทคโนโลยี Industrial Ethernet ทำให้มีความเร็วส่งผ่านข้อมูลสูงสุดเท่ากับ 10/100 Mbit/s ในยุคต่อมา

ในพัฒนาการของระบบ Fieldbus จะเห็นได้ว่า โพรโทคอลคือข้อกำหนดที่ใช้เพื่อเป็นมาตรฐานสำหรับการสื่อสารระหว่างคอมพิวเตอร์ หรือภาษาที่ใช้สื่อสารระหว่างคอมพิวเตอร์บนเครือข่าย จะมีความหลากหลายเพิ่มขึ้นอย่างมากในช่วงคริสต์ทศวรรษ 1980 อันเกิดจากพัฒนาการของฮาร์ดแวร์ด้าน Microprocessors การพัฒนาของวิศวกรรมด้านการส่งผ่านข้อมูลในสายโทรศัพท์ และที่สำคัญการพัฒนาของวิศวกรรมคอมพิวเตอร์ที่ได้เสนอรูปแบบมาตรฐาน The International Organization for Standardization (ISO) เกี่ยวกับ Open System Interconnection (OSI) ในช่วงต้นปี ค.ศ 1980 เพื่อเป็นมาตรฐานกลางอ้างอิงการสื่อสารและเปรียบเทียบการทำงานบนเครือข่าย หลังจากนั้น โพรโทคอลเกี่ยวกับ Fieldbus ได้เกิดขึ้นมากมายนับจากนั้น เพราะระบบอัตโนมัติได้ถูกใช้แก้ปัญหาในหลากหลายอุตสาหกรรมไม่เพียงแต่ในโรงงานการผลิต ยังใช้ในระบบการควบคุมอาคารและบ้านเรือน ระบบควบคุมในรถยนต์ ระบบควบคุมในอากาศยาน ระบบควบคุมไฟฟ้ากำลัง และ ระบบควบคุมเครื่องจักรในอุตสาหกรรมผลิตเซมิคอนดักเตอร์ เป็นต้น ซึ่งการประยุกต์ใช้แก้ปัญหาในหลากหลายอุตสาหกรรมก็มีข้อกำหนดการใช้งาน รวมถึงตัวแปรต้นและตัวแปรตามที่แตกต่างกัน จึงทำให้โพรโทคอลของ Fieldbus มีจำนวนมากและหลากหลาย แต่โพรโทคอลที่สามารถใช้งานได้ต่อเนื่องยังปัจจุบันจะขึ้นอยู่กับปัจจัยสำคัญอันหนึ่งกล่าวคือ โพรโทคอลนั้นจะต้องมีคุณสมบัติแบบเปิดและมีมาตรฐาน (open and standard) ซึ่งหมายถึงเป็นโพรโทคอลที่เปิดเผยข้อมูลแก่สาธารณะเพื่อให้โอกาศบริษัทผู้ผลิตต่าง ๆ สามารถเข้าถึงและใช้โพรโทคอลในอุปกรณ์ที่บริษัทนั้น ๆ ผลิตได้ ไม่ใช่เป็นสมบัติเอกสิทธิ์เฉพาะสำหรับบริษัทหนึ่งบริษัทใดที่ปกปิดและห้ามใช้ทั่วไป และมีมาตรฐานรองรับที่อุปกรณ์ที่ผลิตจากหลากหลายบริษัทสามารถทำงานร่วมกันได้ อันนำไปสู่ความน่าเชื่อถือของผู้บริโภคที่จะเลือกใช้โพรโทคอลนั้นและลงทุนซื้อเครื่องจักรในกระบวนการผลิตที่รองรับโพรโทคอลที่มีความน่าเชื่อถือต่อไป และเป็นสูตรสำเร็จที่ทำให้บริษัทผู้ผลิตเครื่องจักรเลือกใช้โพรโทคอลที่ได้รับการยอมรับต่อเนื่องและยืนยาว

นักพัฒนาโพรโทคอลของระบบ Fieldbus ได้พัฒนาตามแนวทางรูปแบบ OSI โดยอาศัยโพรโทคอล Medium Access Control (MAC) ในชั้น data link ในการสื่อสารโดยอาศัยเทคนิค time division multiple access (TDMA) ซึ่ง bandwidth จะ

ถูกใช้ร่วมกันใน time domain เป็นหลัก และการสื่อสารด้วยเทคนิคอื่นกล่าวคือ Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA) และ Space Division Multiple Access (SDMA) ไม่ได้ถูกใช้งานในระบบ Fieldbus

ตัวอย่างของโพรโทคอลในระบบ Fieldbus [4] ที่ใช้วิธีการสื่อสารที่ต่างกัน มีบางโพรโทคอลที่ยังได้รับความสนใจจากผู้ใช้งานในปัจจุบัน เช่น PROFIBUS (Process Field Bus) ซึ่งแนะนำเข้าสู่อุตสาหกรรมจาก BMBF (German department of education and research) และต่อมาใช้โดยบริษัท Siemens , Modbus RTU ซึ่งแนะนำเข้าสู่อุตสาหกรรมจากบริษัท Modicon และต่อมาใช้โดยบริษัท Schneider Electric, CAN (Controller Area Network) ซึ่งแนะนำมาจากอุตสาหกรรมรถยนต์ โดยบริษัท Bosch GmbH และมีการใช้โพรโทคอลอื่น ๆ ซึ่งอาศัยการสร้างชั้น application อยู่บน CAN เช่น DeviceNet, CANopen และ SDS เป็นต้น

ข้อเด่นของระบบ Fieldbus ได้แก่ โพรโทคอลที่เป็นแบบ deterministic ซึ่งสามารถคาดการณ์เหตุการณ์ในอนาคตได้จากข้อมูลปัจจุบันและอดีต มีการสื่อสารแบบ real time มีความทันเวลา (timeliness) และ มีความเชื่อมั่นของสัญญาณ (reliability)

2) Industrial Ethernet

ระบบสื่อสารในโครงข่ายคอมพิวเตอร์ส่วนบุคคล อาศัยเทคโนโลยี Ethernet ซึ่งปัจจุบันได้พัฒนาให้ทันสมัยมาก ในแง่การรับส่งข้อมูล มีความสามารถในการรับส่งข้อมูลได้เร็วมากในช่วง Gigabit ต่อวินาที ตลอดจนความสามารถของ Application Specific IC (ASIC) ในการประมวลผลได้เพิ่มขึ้น รวดเร็วขึ้น และ มีขนาดเล็กลง ทำให้ติดตั้งลงในอุปกรณ์เคลื่อนที่ไร้สายได้ นอกจากนี้ราคาของอุปกรณ์ในระบบ Ethernet ก็ถูกลงมากสามารถพบเห็นในอุปกรณ์อิเล็กทรอนิกส์ทั่วไป เมื่อเปรียบเทียบการสื่อสารระหว่าง Ethernet กับ Fieldbus จะเห็นได้ว่ามีความแตกต่างกันในด้านประสิทธิภาพเป็นอย่างมาก จึงมีผลให้มีความต้องการ ใช้การสื่อสารแบบ Ethernet เข้าแทนที่การสื่อสารแบบ Fieldbus ในปัจจุบันมากขึ้นเรื่อย ๆ

Industrial Ethernet Protocols	พัฒนาโดย	องค์กรที่ดูแล
Ethernet/IP	Rockwell	Open DeviceNet Vendors Association (ODVA)
PROFINET	Siemens	PROFIBUS & PROFINET International (PI)
EtherCAT	Beckhoff	The EtherCAT Technology Group (ETG)
Modbus TCP	Modicon (now Schneider Electric)	The Modbus Organization
POWERLINK	B&R	The Ethernet POWERLINK Standardization Group (EPSG)
Sercos III	ABB, AEG, AMK, Robert Bosch, Indramat, Siemens	Sercos International e.V
CC-Link IE	Mitsubishi Electric Corporation	CC-Link Partner Association (CLPA)

รูปที่ 22 รายละเอียด Industrial Ethernet โพรโทคอลต่าง ๆ

ในขณะเดียวกัน ระบบอัตโนมัติมีความต้องการที่ซับซ้อนขึ้น ทำให้จำนวนตัวแปรต้นและตัวแปรตามเพิ่มขึ้นมาก นอกจากนี้ความต้องการใช้งานเกี่ยวกับภาพและเสียงในระบบอัตโนมัติอุตสาหกรรมเพื่อเพิ่มประสิทธิภาพของระบบ ทำให้ระบบ Fieldbus แบบเดิมไม่สามารถรองรับความต้องการในการสื่อสารข้อมูลที่เพิ่มมากขึ้นได้

โดยพื้นฐานแล้วการสื่อสารแบบ Ethernet ไม่สามารถประยุกต์ใช้โดยตรงบนระบบอัตโนมัติอุตสาหกรรม เนื่องจาก Ethernet เป็นการสื่อสารแบบ nondeterministic ทำให้ต้องพัฒนาโพรโทคอลใหม่ที่มีความสามารถนี้ และสามารถควบคุมในเวลาจริง (Real time control) ได้

โพรโทคอลสำหรับ Industrial Ethernet ได้แก่ EtherNet/IP, PROFINET, EtherCAT, Modbus TCP, POWERLINK, SERCOS III และ CC-Link IE โดยมีรายละเอียดในรูปที่ 22

ในส่วนของอุปกรณ์ที่เกี่ยวข้องกับ Industrial Ethernet นั้นประกอบด้วย hardware เช่น Ethernet hub, Ethernet switches, Routers และสายเคเบิล เป็นต้น และ software solution โดยมีบริษัทผู้เล่นรายใหญ่ต่าง ๆ ที่มีสินค้าและบริการ ดังรายนามต่อไปนี้ Siemens, Schneider Electric, Rockwell Automation, Beckhoff Automation, Cisco, Belden, B&R Automation, Eaton, Endress+Hauser, Parker Hannifin, ABB, Bosch Rexroth, GE, Honeywell International, IDEC, Hitachi, OMRON, ACS Motion Control Ltd.

นอกจากนี้ยังมีข้อมูลการวิจัยตลาดเกี่ยวกับ Industrial Ethernet โดยต้องเสียค่าสมาชิกในการเข้าถึงข้อมูลเช่น https://www.marketresearchfuture.com/reports/industrial-ethernet-market-4829 เป็นต้น

3.2 การเปรียบเทียบโพรโทคอล Industrial Ethernet สำหรับ PLC

ในเอกสารซึ่งเผยแพร่ใน www.ethercat.org ได้นำเสนอเอกสารเปรียบเทียบโพรโทคอล Industrial Ethernet ได้แก่ EtherCAT, EtherNet/IP, Powerlink, PROFINET IRT (Isochronous Real-Time) และ SERCOSIII โดยแบ่งลักษณะของโพรโทคอลออกเป็น 3 ประเภทได้แก่

1) ชนิด Standard Software และ Standard Ethernet

เป็นโพรโทคอลที่อาศัยลำดับชั้นของTCP/IP และมีกลไกการทำงานเวลาจริงบนชั้นสูงสุด การใช้เทคนิคนี้จะให้ประสิทธิการสื่อสารอยู่ในช่วงที่จำกัด โพรโทคอลที่ใช้เทคนิคนี้ได้แก่ Ethernet/IP

2) ชนิด Open Software และ Standard Ethernet

เป็นโพรโทคอลที่มีชั้นสร้างใหม่อยู่บนชั้นของ Ethernet เดิม และมีการสร้างการควบคุมโดยซอฟต์แวร์ที่เป็นกรรมสิทธิ์ในชั้น 3 และ 4 ในแบบจำลอง OSI เพื่อทำให้การจัดการลำดับเวลาในการสื่อสารเป็นแบบ determinism โพรโทคอลที่ใช้เทคนิคนี้ได้แก่ Powerlink

3) ชนิด Open Software และ Modified Ethernet

เทคนิคนี้อาศัยการสร้างมาตรฐานใหม่และยังคงใช้อุปกรณ์เดิมของ Ethernet แต่มีการเพิ่มเติมด้วยการสร้างซอฟต์แวร์ใหม่ที่เปิดเผยต่อสาธารณะและมีการเพิ่มเติมอุปกรณ์การตัดต่อพิเศษ (Special switch) หรือชิป ASIC ติดตั้งบนอุปกรณ์สื่อสาร โพรโทคอลที่ใช้เทคนิคนี้ได้แก่ EtherCAT, PROFINET IRT และ SERCOS III

3.3 ในการประยุกต์ใช้ Industrial Ethernet สามารถแยกความต้องการการใช้งานออกเป็น 3 กลุ่ม ได้แก่

ก) กลุ่มต้องการความเร็วต่ำโดยมีความต้องการเวลาในการตอบสนองในช่วง 100 ms ซึ่งกลุ่มนี้มีลักษณะให้มนุษย์มีส่วนร่วมในการสังเกตุการณ์ เช่น การตรวจสอบกระบวนการ โดยส่วนมากระบบอัตโนมัติควบคุมการผลิตและระบบควบคุมอาคารจะอยู่ในกลุ่มนี้

ข) กลุ่มต้องการความเร็วปานกลางโดยมีความต้องการเวลาในการตอบสนองน้อยสุดต่ำกว่า 10 ms ซึ่งเป็นกลุ่มงานการควบคุมเครื่องจักรด้วย PLC

ค) กลุ่มต้องการความเร็วสูงโดยมีความต้องการเวลาในการตอบสนองต่ำกว่า 1 ms ซึ่งเป็นกลุ่มงานเกี่ยวข้องกับการควบคุมการเคลื่อนไหวหรือหุ่นยนต์

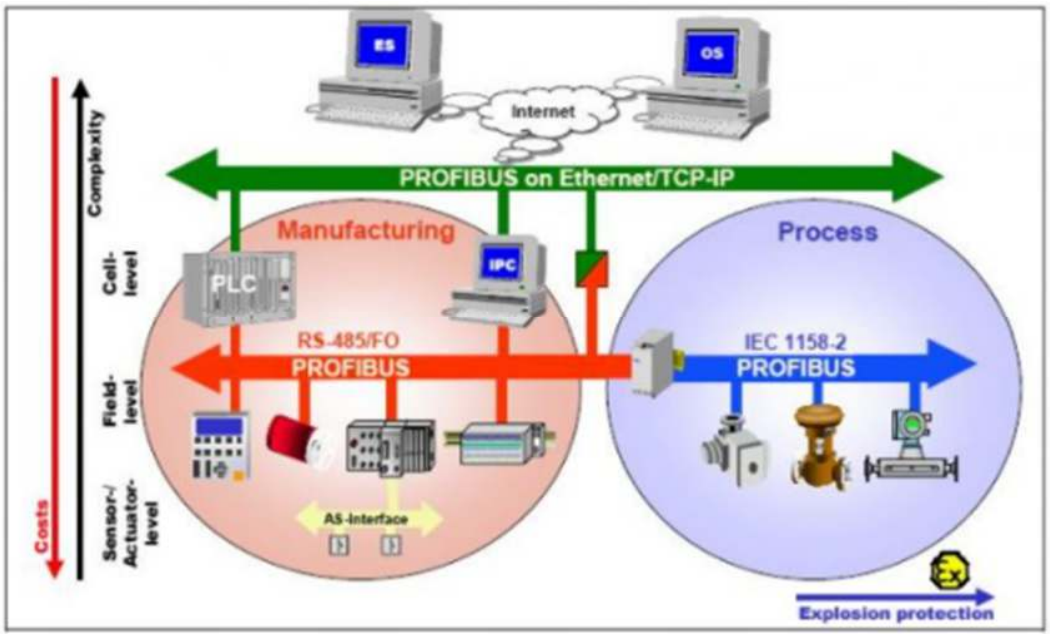
ความสามารถของโพรโทคอลทั้ง 5 พบว่า เวลาในการตอบสนองมีค่า เท่ากับ หรือ น้อยกว่า 1 ms และแต่ละองค์กรที่ควบคุมดูแลโพรโทคอลแต่ละชนิดก็จะประชาสัมพันธ์เกี่ยวกับจุดเด่นของประโตคอลที่ตัวเองดูแล และมักจะมีข้อมูลแสดงผลว่า โพรโทคอลของตัวเองมีเวลาในการตอบสนองมีค่าน้อยที่สุด เมื่อเทียบกับโพรโทคอลอื่น ๆ ดังนั้นการพิจารณาเอกสารจากแหล่งข้อมูลใด ๆ ที่อ้างว่า เป็นโพรโทคอลที่มีเวลาในการตอบสนองมีค่าน้อยที่สุด [6,7] ซึ่งมีนัยยะว่าเป็นโพรโทคอลที่ดีที่สุด จึงต้องทำการวิเคราะห์อย่างระมัดระวังในเรื่องกระบวนการทดสอบที่ใช้เทคนิคแตกต่างกัน เพราะการประชาสัมพันธ์ว่าเป็นโพรโทคอลที่ดีที่สุดจะเล็งเห็นผลให้เกิดผลประโยชน์ทางธุรกิจมหาศาลเกิดขึ้นกับองค์กรที่ควบคุมดูแลโพรโทคอลนั้น ๆ

3.4 การวิเคราะห์และสรุปผล

เนื่องจากประเทศไทยไม่ได้ผลิตเทคโนโลยีเกี่ยวกับ Industrial Communications เอง ดังนั้นองค์กรภาครัฐและบริษัทเอกขนจึงต้องให้ความสำคัญในการประเมินเทคโนโลยีที่จะใช้ในแง่ของประสิทธิภาพต่อราคาเสมอ จากข้อมูลทางเทคนิคจะเห็นได้ว่าเทคโนโลยีการสื่อสาร Industrial Ethernet จะมีความสำคัญเพิ่มมากขึ้นในระบบอัตโนมัติอุตสาหกรรม โดยจะเป็นเทคโนโลยีการสื่อสารที่บริษัทผู้ผลิตเครื่องจักรใหม่เลือกใช้แทนที่ระบบ Fieldbus เดิม แต่การใช้เทคโนโลยี Industrial Ethernet ยังไม่สามารถแทนที่ระบบ Fieldbus ได้ทั้งหมดในเวลาอันใกล้นี้ เนื่องจากสายการผลิตที่ใช้งานเครื่องจักรเก่ายังมีการใช้งานอยู่มาก ดังนั้นการพัฒนาปรับปรุงสายการผลิตเก่าเป็นแบบใหม่จึงเป็นเหตุปัจจัยที่แต่ละองค์กรหรือบริษัทจะต้องทำการตัดสินใจทางธุรกิจเพื่อให้คุ้มค่ากับการลงทุน

4 Profibus / ProfiNet

https://pmc.co.th/profibus-ตอนที่-1/

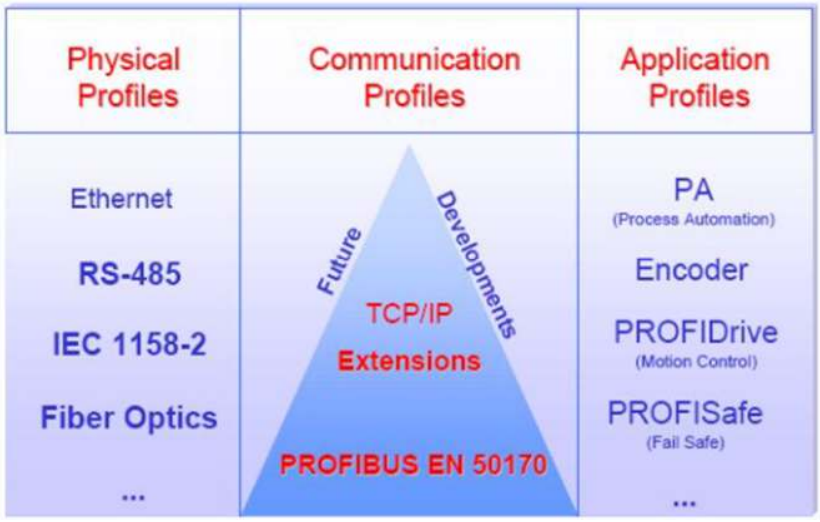


รูปที่ 23 การเชื่อมต่อผ่าน profibus

PROFIBUS เป็นมาตรฐานแบบเปิดประยุกต์ใช้อย่างแพร่หลายในกระบวนการผลิต และระบบอัตโนมัติจากผู้ผลิตจำนวนมาก ภายใต้มาตรฐานสากล EN50170 และ EN50254 แม้ผลิตภัณฑ์จะมาจากหลากหลายผู้ผลิตก็สามารถที่จะสื่อสารถึงกันได้โดยมิต้องใช้อุปกรณ์พิเศษ หรือการปรับแต่งเพิ่มเติมแต่อย่างใด

- **At the sensor-actuator level** อุปกรณ์ในระดับนี้จะผลิตสัญญาณลักษณะ 0 และ 1 จะถูกส่งไปยัง Bus ที่มีการแชร์ใช้ร่วมกัน ผ่าน AS-Interface ซึ่งจะทำหน้าที่ในการรับสัญญาณเหล่านี้อย่างเหมาะสม
- **At field level** อุปกรณ์ปลายทาง เช่น I/O modules , Transducers , Drive Units , Analyzer , Terminals จะสื่อสารกันภายในระบบ เรียกรวมว่า Automation System อย่างเป็นเรียลไทม์ การสื่อสารจะในกระบวนการจะเป็นไซท์เคิลในขณะที่ Additional interrupts , Configuration Data และ diagnosis data จะสามารถสื่อสารได้ทันทีตามกำหนดโดยไม่ต้องรอคอยไซท์เคิล
- **At the Cell level** พีแอลซี หรือ คอมพิวเตอร์ สามารถสื่อสารระหว่างกันหรือโลกภายนอกผ่านระบบ IT โดยการใช้ Ethernet TCP/IP ซึ่งเป็นการสื่อสารที่ใช้ในการส่งข้อมูลขนาดใหญ่ PROFIBUS ที่ถูกพัฒนาให้เชื่อมต่อการคือสารกับ Ethernet TCP/IP คือ PROFinet
- **Fieldbuses** คือระบบการสื่อสารสำหรับอุตสาหกรรมที่ใช้ตัวกลางเช่น ทองแดง ไฟเบอร์ออฟติค หรือ ไวท์ไฟโดยมีรูปแบบของสัญญาณเดินทางเป็นอนุกรมกันไป โดยสัญญาณเหล่านี้จะมาจาก Sensors , Actuators หรือ Transducers สัญญาณจะถูกส่งไปยัง ระบบควบคุมกลาง หรือ ระบบบริหารจัดการกลาง Field Bus Technology เริ่มคิดค้นในช่วงปี 80 วัตถุประสงค์เพื่อทดแทนการส่งสัญญาณแบบขนานที่ต้องใช้สายสัญญาณจำนวนมาก โดยใช้เทคโนโลยีดิจิตอลด้วย

ความต้องการที่หลากหลาย แตกต่างกันในแต่ละผู้ผลิต การประยุกต์ใช้ที่ต่างกันออกไปเพื่อให้การพัฒนามีความเป็นหนึ่งเดียว สามารถทำงานร่วมกันได้จึงต้องมีการกำหนดมาตรฐานการสื่อสารสำหรับกระบวนการผลิตในอุตสาหกรรมให้เหมือนกัน PROFIBUS คือหนึ่งในนั้น



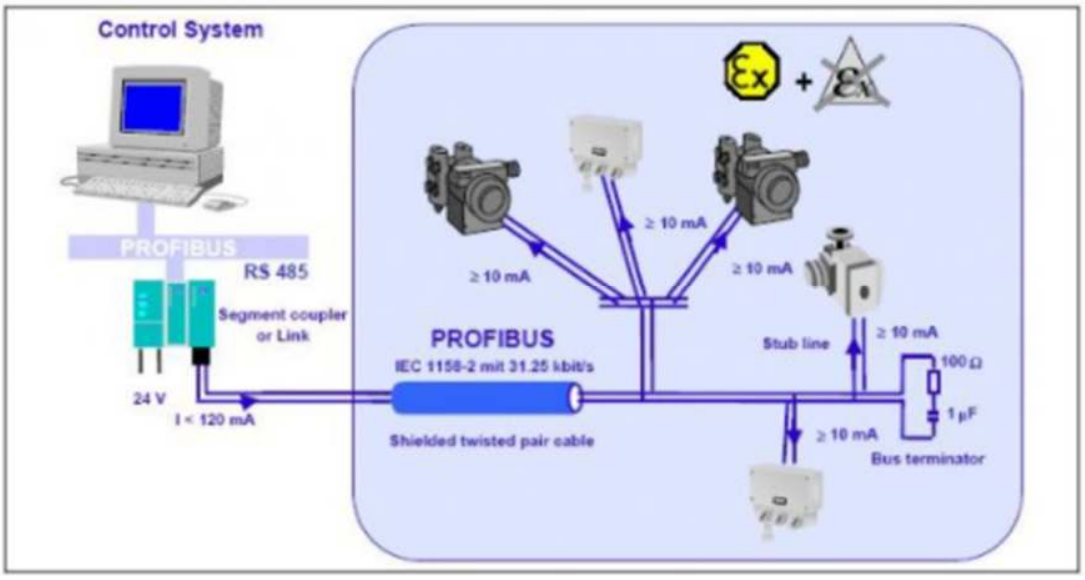
รูปที่.24 Physical, Communication และ Application Layer

- **Communication Profiles** เป็นการกำหนดว่าผู้ใช้งานจะส่งข้อมูลอย่างไร มี 2 แบบคือ DP และ FMS
- **DP** ถูกใช้งานมากที่สุด มีความเร็วประสิทธิภาพ เหมาะสม การต่อใช้งานไม่ยุ่งยากมาก ออกแบบมาเพื่อการสื่อสารระหว่างระบบอัตโนมัติและอุปกรณ์ต่อพวกปลายทาง
- **FMS** จะถูกประยุกต์ใช้ในงานที่มีความซับซ้อนกับอุปกรณ์อัจริยะทั้งหลาย ปัจจุบันมีการใช้ PROFIBUS ร่วมกับ TCP/IP ทำให้ FMS ลดความสำคัญลง
- **Physical Profiles** ปัจจุบันมีกการสื่อสารผ่านสายอยู่ 3 แบบคือ
    - RS485
    - IEC 1158-2
    - Optical Fiber

RS485 เป็นเทคโนโลยีที่พบเห็นได้บ่อยที่สุดของ PROFIBUS ความเร็วในการสื่อสารสูง การต่อใช้งานทำได้ง่ายและต้นทุนไม่สูง สามารถใช้สายทองแดง Twisted pair ที่มีชีลก็เพียงพอต่อการประยุกต์ใช้งานรูปแบบสายส่งสัญญาณ (Bus) ลักษณะนี้จะสามารถเพิ่มอุปกรณ์หรือเอาอุปกรณ์ออก โดยไม่สร้างผลกระทบใดๆ ต่ออุปกรณ์ตัวอื่นๆ ความเร็วในการสื่อสารจะอยู่ระหว่าง 9.6 Kbit/sec และ 12 Mbit/sec

IEC 1158-2 เป็นการสื่อสารแบบ Synchronous ที่ความเร็ว 31.25kbit/sec เหมาะสมสำหรับกระบวนผลิตที่ต้องใช้ระมัดระวังเป็นพิเศษ เช่นกระบวนการทางเคมีกระบวนการทางปิโตเรียม ที่ต้องคำนึงถึงความปลอดภัยสูงหรือบริเวณที่อาจเกิดการระเบิดได้ รูปแบบสายส่งสัญญาณจะใช้สายเพียง 2 เส้น ส่งผ่านกระแสไฟฟ้าจากภายนอก อุปกรณ์แต่ละตัวออกแบบให้กินกระแส

เพียง 10mA การสื่อสารจะเกิดจากการ modulation สัญญาณ +/-9 mA กับกระแสไฟฟ้าปกติ อุปกรณ์ที่ชื่อว่า Segment Coupler จะทำหน้าที่แปลงสัญญาณจาก RS485 PROFIBUS เป็น IEC1158-2 PROFIBUS



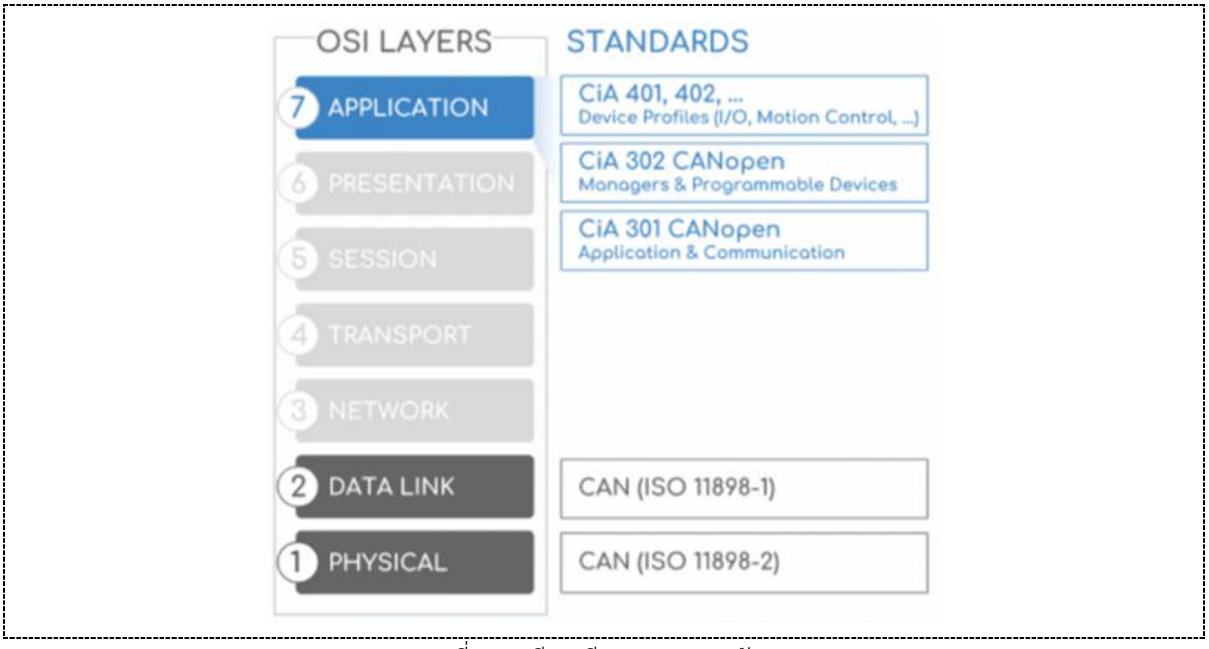
รูปที่ 25 Profibus Connection

Fiber Optic เหมาะสมกับสภาพแวดล้อมที่มีการรบกวนทางแม่เหล็กไฟฟ้าสูง เช่นต้องผ่านหม้อแปลงไฟฟ้า เสาส่งสัญญาณโทรคมนาคม จะทำให้สัญญาณไม่มีความผิดเพี้ยน และสามารถส่งสัญญาณด้วยความเร็วสูงได้ระยะทางไกล PROFIBUS แบบนี้จะนี้ผู้ใช้งานจะต้องเลือกว่าจะต่อสายสื่อสารแบบ Star หรือ แบบ Ring ผู้ผลิตบางรายออกแบบให้มีระบบ Redundant สามารถสลับไปใช้สายสำรอง เมื่อสายหลักได้รับความเสียหาย ผู้ผลิตจำนวนมากมีอุปกรณ์แปลงระหว่าง RS485 กับ Optical Fiber ให้เลือกใช้

5 Can Open

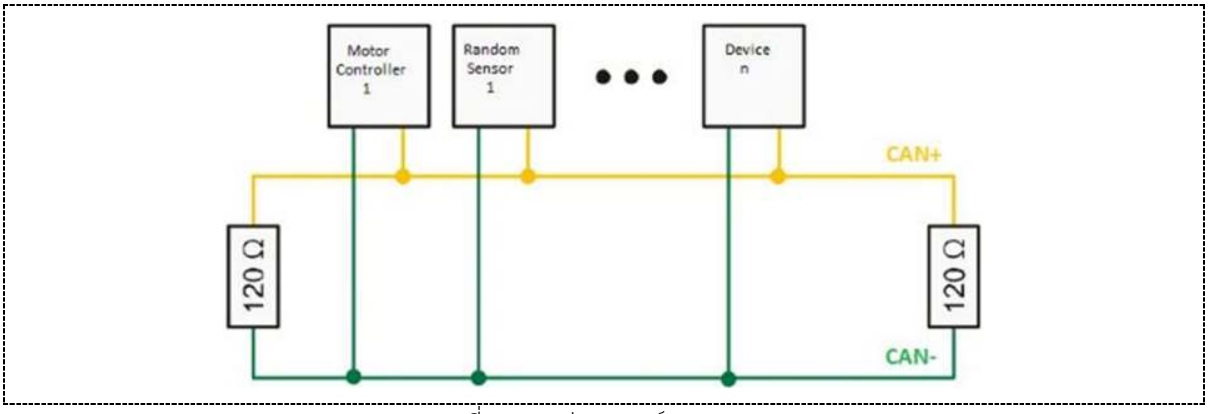
https://pmc.co.th/can-open-ตอนที่-1/

CAN ย่อมากจากคำว่า Controller Area Network พัฒนาขึ้นโดย Bosch Germany สำหรับใช้ในการควบคุมอุปกรณ์ภายในรถยนต์ ต่อมาก็ถูกนำมาใช้ในอุตสาหกรรมอย่างแพร่หลาย CAN เป็นระบบที่สามารถมีได้หลาย Master เพราะอุปกรณ์ทุกตัวสามารถเข้าถึง Bus ได้ตลอดเวลาที่ไม่มีการส่งสัญญาณ CAN มิได้ทำงานในแบบ address แต่ทำงานในแบบการส่ง message การเข้าถึง Bus ของอุปกรณ์ทุกตัวจะใช้ความสามารถของ CSMA/CA Protocol ย่อยในชั้น Data Link layer ของ OSI โมเดล เพื่อบริหารการส่ง message ของทุกอุปกรณ์ลงมาที่ Bus แต่ละอุปกรณ์จะ listen ว่า Bus ว่างอยู่หรือไม่ ถ้าว่างอยู่อุปกรณ์ก็สามารถส่ง message ออกมายัง Bus ได้ ถ้าหากอุปกรณ์ 2 ตัวพยายามที่จะส่ง message ลงมาที่ Bus พร้อม ๆ กัน message ที่มีลำดับความสำคัญสูงกว่า(lowest identifier) จะสามารถส่ง message ได้ก่อน ส่วนอุปกรณ์ที่ส่ง message ที่มีลำดับรองลงมาจะสามารถส่ง message ในจังหวะที่ Bus ว่างในจังหวะถัดไป message สามารถถูกรับโดยอุปกรณ์ทุกตัวที่ต่ออยู่กับ Bus โดยอุปกรณ์แต่ละตัวจะคัดเอาเฉพาะ message ที่เป็นของตนเองเท่านั้น



รูปที่ 26 เปรียบเทียบ CAN Bus กับ OSI

CAN จะทำงานที่ Physical Layer และ Datalink Layer ตาม OSI โมเดล ส่วน CAN open เป็นมาตรฐานที่ถูกกำหนดขึ้นมาทำงานที่ Application Layer ซึ่งมีอยู่หลายมาตรฐานย่อย เช่น CAN in Automation Group (Cia) 301 จะเป็นหลักเกณฑ์ในการสร้าง message เพื่อการสื่อสารพื้นฐาน CiA 302 จะเป็นหลักเกณฑ์ในการสร้าง message ของ PLC sหรือ System Monitoring นอกจากนี้ CiA 406 ยังเป็นหลักเกณฑ์ในการสร้าง message ของอุปกรณ์ Sensor เช่น Encoder



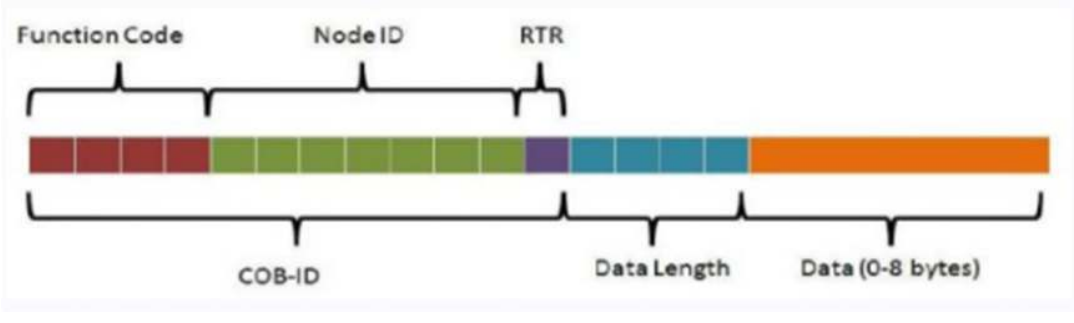
รูปที่ 27 การต่ออุปกรณ์บน CAN Bus

CAN open จะสร้างสายสื่อสารทำหน้าที่เป็น Bus เพียง 2 เส้น มีค่า Baud Rate อยู่ระหว่าง 10kBoud ถึง 1 MBoud ขึ้นอยู่กับความยาวสาย สายยิ่งยาวมากก็จำเป็นจะต้องลดค่า Baud Rate ลงมาเพื่อรักษาคุณภาพของข้อมูล การต่อสายสัญญาณลักษณะนี้สามารถลดผลกระทบของสัญญาณรบกวนได้ดี

Bus length	Bit Rate	Bit time
25 meters	1000 kbit/s	1 us
50 meters	800 kbit/s	1.25 us
100 meters	500 kbit/s	2 us
250 meter	250 kbit/s	4 us
500 meters	125 kbit/s	8 us
1000 meter	50 kbit/s	20 us
2500 meters	20 kbit/s	50 us

รูปที่ 28 Baud Rate ของ CAN Open

Message จะถูกส่งในรูปของ message telegram ประกอบด้วย 3 ส่วนคือ Communication Object IDentifier หรือเรียกย่อ ๆ ว่า COB-ID ขนาด 12 bits ประกอบด้วย 3 ส่วนย่อย คือ Function Code 4 bits , Node ID 7 bits และ RTR 1 bit ส่วนนี้ปกติจะถูกเรียกว่า CAN-ID นอกจากนี้แล้ว Node ID ขนาด 7 bits หมายถึงจำนวนอุปกรณ์ต่ออยู่กับ Bus มีได้สูงสุดไม่เกิน 127 อุปกรณ์ Data length ขนาด 4 bits Data ขนาด 0 – 8 Bytes



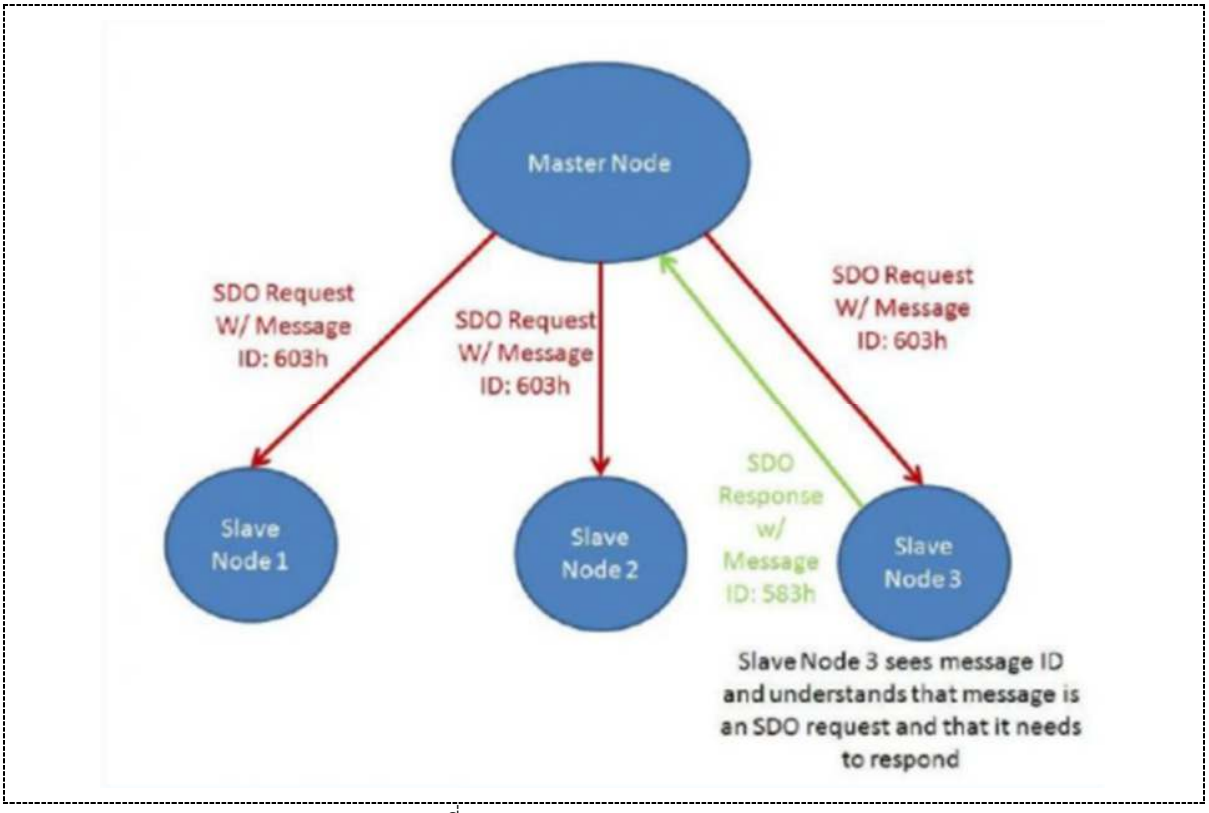
รูปที่ 29 CAN Data Frame

COB-Identifier ในแต่ละ message จะมีลำดับความสำคัญไม่เท่ากัน ซึ่งค่า COB-ID จะประกอบด้วย Function Code และ Node ID ส่วนค่า RTR ไม่นิยมนำมาคิดเป็นค่า COB-ID ค่า Node ID จะต้องกำหนดให้แก่อุปกรณ์โดยไม่ซ้ำกัน ส่วนค่า Function Code จะแปรเปลี่ยนตามชนิดของ message แบ่งออกได้ 4 ชนิด Administrative message (LMT, NMT) สำหรับใช้ในการควบคุม สถานะของ Bus สามารถใช้ในการ start , stop หรือ reset อุปกรณ์ โดย message กลุ่มนี้จะมีลักษณะ Broadcast ไปยังอุปกรณ์ทุกตัวใน network Service data objects (SDOs) สำหรับใช้เพื่อการกำหนดค่าให้กับอุปกรณ์ การส่งสัญญาณไม่เป็นคาบเวลา ปกติพบในช่วงที่มีการเริ่มจ่ายไฟฟ้าให้แก่ระบบ Process data objects (PDOs) สำหรับการแลกเปลี่ยนข้อมูลที่เป็นเรียลไทม์ ซึ่งมักเป็น message ที่มีลำดับความสำคัญสูง Pre-defined message (Synchronization, emergency) เป็น message เพื่อใช้ในการ Sync สัญญาณ หรือ รับสัญญาณ Emergency จากอุปกรณ์ มีลำดับความสำคัญสูงสุด

Object	Function code (binary)	COB-ID result	Hex.	Priority class*
NMT	0000	0		0
SYNC	0001	128	80	0
Emergency	0010	129-255	81-FF	0,1
PDO (tx)	0011	385-511	181-1FF	1,2
PDO (rx)	0100	513-639	201-27F	2
PDO (tx)	0101	641-767	281-2FF	2,3
PDO (rx)	0110	769-895	301-37F	3,4
SDO (tx)	1011	1409-1535	581-5FF	6
SDO (rx)	1100	1537-1663	601-67F	6,7

รูปที่ 30 Function Code Table

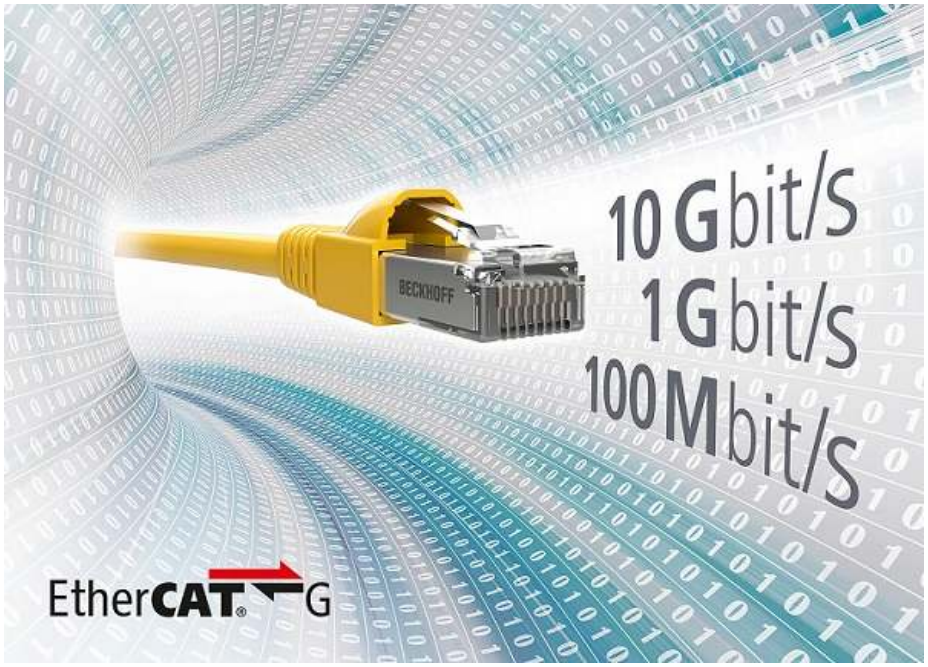
จากตารางจะเห็นว่า message กลุ่ม NMT จะมี Function Code = 0000 (เลขฐาน 2) และเนื่องจาก message กลุ่มนี้มีลักษณะ Broadcast จึงไม่ต้องกำหนด Node ID ค่า COB-ID ทั้ง 11 bits จะเขียนได้เป็น 0000000000 (เลขฐาน 2) จึงมีค่าเท่ากับ 0 (เลขฐาน 10) และ message กลุ่มนี้มี ลำดับความสำคัญสูงสุดคือ 0 SDO(rx) เป็น message ใช้เพื่อกำหนดค่าให้กับอุปกรณ์ จะมีค่า จะมี Function Code = 1100 (เลขฐาน 2) และต้องมีการกำหนด Node ID จำนวน 7 bits ด้วย เมื่อนำค่า Function Code + Node ID = COB-ID ที่จะมีค่าตั้งแต่ 1537 – 1663 (เลขฐาน 10) หรือ 601 – 67F (เลขฐาน 16) มีลำดับความสำคัญต่ำสุดคือ 6 หรือ 7



รูปที่ 31 Communication from Master

จากภาพด้านบน Master Node ส่ง SDO Message ออกมายัง Bus โดยอุปกรณ์ทุกตัวจะได้รับ message ซึ่งกำหนดค่า CAD-ID = 600 + Node ID (เลขฐาน 16) = 603 เมื่ออุปกรณ์ Slave Node 1 และ 2 ได้รับ message ก็จะไม่ตอบสนองใด ๆ เพราะมีใช่ Node ID ของตน เมื่อ Slave Node 3 ได้รับ message และพบว่าเป็นของตนก็จะมีการตอบสนองตามค่า Index และ Sub Index ซึ่งจะได้อธิบายต่อไป Heartbeats เป็นวิธีที่มาตรฐาน CAN open ตรวจสอบว่าอุปกรณ์ที่ต่ออยู่ใน network ยังคง alive หรือทำงานเป็นปกติหรือไม่ โดยกำหนดให้อุปกรณ์ส่ง Heartbeat Message ออกมาเป็นระยะตามช่วงเวลา เพื่อให้ CAN Open Master ทราบว่าอุปกรณ์ยังคงทำงานปกติ หากไม่มีการส่ง Heartbeat Message ออกมาตามเวลาที่กำหนด Master จะดำเนินการอย่างใดอย่างหนึ่ง เช่น ทำการรีเซ็ตอุปกรณ์ตัวนั้น หรืออาจจะมี message alarm เกิดขึ้นเพื่อแจ้งให้ผู้ดูแลระบบทราบ ปกติ Heartbeat Message จะมี CAN-ID = 600 + Node ID (เลขฐาน 16)

6 EtherCAT Field Bus สำหรับงาน ออโตเมชั่น



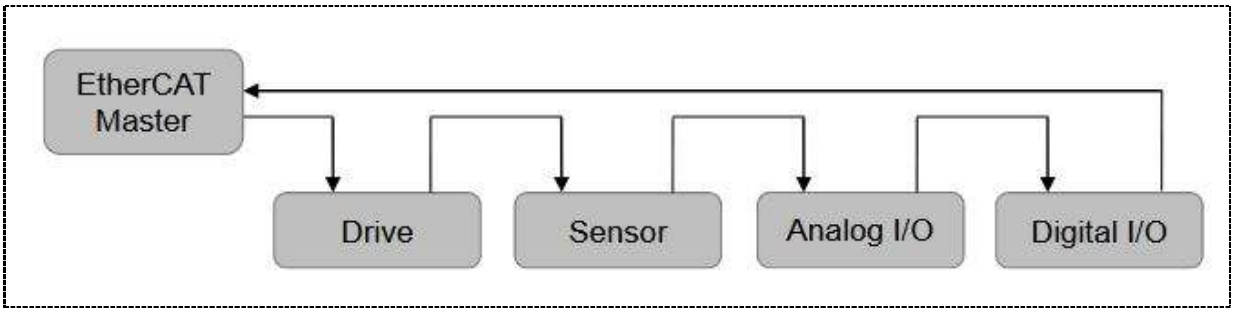
6.1 Introduction

EtherCAT (Ethernet for Control Automation Technology) เป็นโปรโตคอลสำหรับงานด้านออโตเมชั่นมีลักษณะเป็นการสื่อสารแบบเรียลไทม์ ระหว่างอุปกรณ์เช่น อุปกรณ์อินพุต/เอ้าพุต เซนเซอร์ หรือ พีแอลซี ถูกคิดค้นโดย Beckhoff Automation ประเทศเยอรมันนี ปัจจุบันถูกนำมาพัฒนาต่อโดย EtherCAT Technology Group ซึ่งถูกตั้งขึ้นมาเพื่อขยายผลให้เป็นมาตรฐานการสื่อสารสำหรับงานด้านอุตสาหกรรม(Field Bus) อย่างเป็นทางการ ปัจจุบันทางกลุ่มมีสมาชิกกว่า 1,900 องค์กรจาก 52 ประเทศ ที่นำ EtherCAT ไปประยุกต์ใช้กับผลิตภัณฑ์ของตน Ethernet เดิมทีเดียวถูกนำไปประยุกต์ใช้ในหลากหลายผู้ผลิตแต่ละรายนำไปประยุกต์กันตามเฉพาะแบบของตนเอง แต่ก็ยังไม่สามารถที่จะใช้งานได้อย่างมีประสิทธิภาพมากนักภายใต้สภาพแวดล้อมของเทคโนโลยีอุตสาหกรรมที่ต้องการการแลกเปลี่ยนข้อมูลจำนวนมากแต่มีขนาดเล็ก และ Ethernet ก็ไม่ค่อยมีความเป็นเรียลไทม์ ใช้งานในรูปแบบ Star Topoloy (ผ่าน Hub หรือ Switch) โปรโตคอล EtherCAT จึงถูกสร้างขึ้นโดยการปรับแต่ง Ethernet เดิมให้มีความเหมาะสม มีรูปแบบการสื่อสารผ่านโครงข่าย ที่ตอบโจทย์งาน Automation ได้มากกว่า การออกแบบ EtherCAT จึงทำให้อุปกรณ์ PC Based สามารถที่จะสื่อสารกันในงานออโตเมชั่น โดยมีตัวหนึ่งทำหน้าที่เป็น EtherCAT Master และที่เหลือเป็น EtherCAT Slave

6.2 Technology

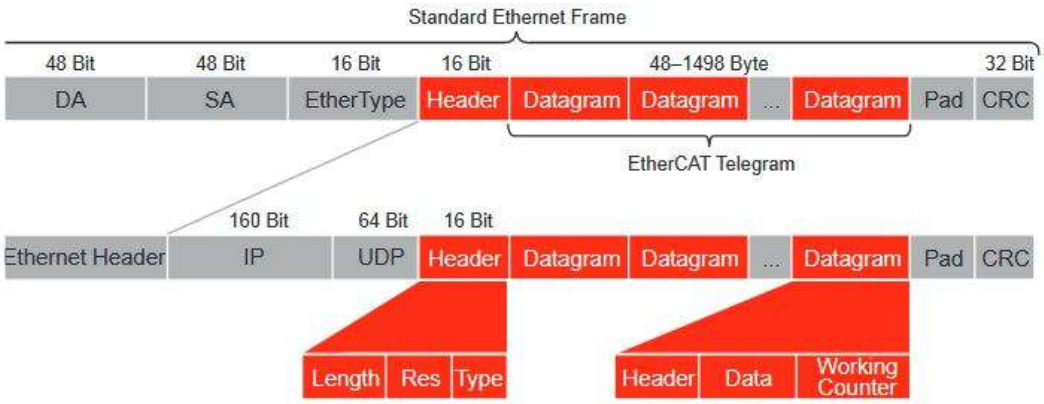
EtherCAT พัฒนามาจาก Ethernet เดิม โดยใช้แนวคิด “on-the-fly” โดยโหนดที่เกาะอยู่ในโครงข่าย EtherCAT Network จะอ่านข้อมูลจาก EtherCAT Frame ที่วิ่งผ่านตัวมัน EtherCAT Frame จะถูกส่งมาจาก EtherCAT Master หลักงานที่ Slave อ่านข้อมูลที่ถูกส่งมาแล้วหากมีข้อมูลใดที่จะส่งต่อไปก็จะใช้จังหวะนั้นในการเขียนข้อมูลลงใน EtherCAT Frame และส่งผ่านตัวเองไป ซึ่งการเขียนข้อมูลลงใน EtherCAT Frame จะเป็นข้อมูลเล็ก ๆ อันเป็นลักษณะของข้อมูลในงานออโตเมชั่น ซึ่งมีลักษณะ

ไม่ใหญ่มากอยู่แล้ว หากเราใช้ Ethernet แบบเดิม ทุก ๆ ข้อมูลขนาดเล็กของแต่ละโหนดจะต้องถูกส่งในรูปแบบ Ethernet Frame เพราะฉะนั้นแม้ข้อมูลจะมีขนาดเล็กมาก แต่ทุก ๆ ข้อมูลเล็ก จะต้องถูกส่งออกมาเป็น Ethernet Frame จำนวนเท่า ๆ กับจำนวนข้อมูล ทำให้ Ethernet แบบเดิมไม่สามารถใช้ประสิทธิภาพของ Ethernet Frame ได้อย่างเต็มที่ แต่ด้วย EtherCAT Frame ทุก ๆ ข้อมูลเล็กๆ ของทุก ๆ อุปกรณ์ที่ EtherCAT Frame วิ่งผ่านจะถูกเขียนข้อมูลของอุปกรณ์แต่ละตัวลงไปใน Frame เดียวกันแล้วจึงส่งผ่านตัวเองไป ทำให้ความเร็ว 100Mbps สามารถถูกใช้ไปในการส่งผ่านข้อมูลขนาดเล็ก ๆ ของแต่ละโหนดได้สูงสุดถึง 90% จึงเป็นการใช้ศักยภาพของ Ethernet Frame ได้อย่างสูงสุดเมื่อมีการปรับเปลี่ยนรูปแบบมาเป็น EtherCAT



6.3 EtherCAT Telegram

EtherCATTelegram จะถูกบรรจุลงใน Ethernet Frame ซึ่งอาจจะมีเพียงหนึ่งหรือหลาย ๆ EtherCAT Datagram ซึ่งมีจุดหมายคือ EtherCAT Slave ซึ่งจริง ๆ ก็คือ Ethernet Frame ตามมาตรฐาน IEEE802.3 ที่ถูกนำมาประยุกต์เป็น EtherCAT โดยกำหนด Header ของ Ethernet Frame เป็น 88A4H ก็จะทำให้อุปกรณ์เดิมที่ใช้เทคโนโลยี Ethernet เช่น LAN Card ก็สามารถทราบได้ทันทีว่า Frame นี้เป็น EtherCAT ทำให้เทคโนโลยีนี้มีราคาถูก EtherCAT Datagram ก็คือคำสั่งที่ประกอบด้วย Header , Data , working counter โดย Header และ Data จะหมายถึงสิ่งที่ EtherCAT Slave จะต้องดำเนินการ และส่วน working counter จะหมายถึง การอัพเดทโดย Slave เพื่อบอกให้ Master ทราบว่า Slave ได้ดำเนินการตามคำสั่งแล้ว

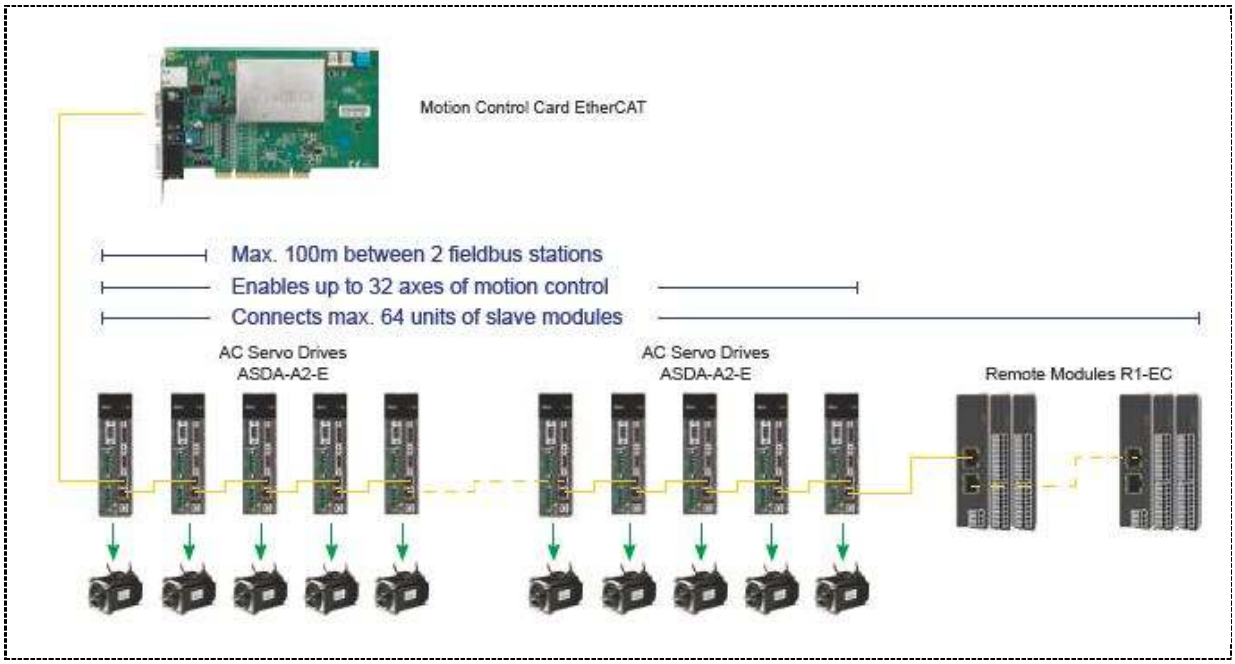


EtherCAT Slave จะอ่าน EtherCAT packets ในแบบ “on-the-fly”คือการอ่านข้อมูลใน EtherCAT Frame และดำเนินการตามคำสั่งถ้าหากว่ามี Datagram ที่ตรงกับ address ของตนเอง และทำการส่งผ่าน Datagram ทั้งหมดออกไปยัง output port ซึ่งเชื่อมต่ออยู่กับ Slave ตัวถัดไปในขณะเดียวกันก็สามารถที่จะอัพเดทค่าในแพคเก็จ ก่อนที่จะส่งออกไป EtherCAT

Master สามารถที่จะสร้าง EtherCAT Frame ที่มีขนาดได้สูงถึง 4GB อันประกอบไปด้วย Datagram ที่อ้างถึง Slave ได้สูงสุดถึง 65,536 Slaves และไม่มีข้อกำหนดในการจัดเรียงลำดับ

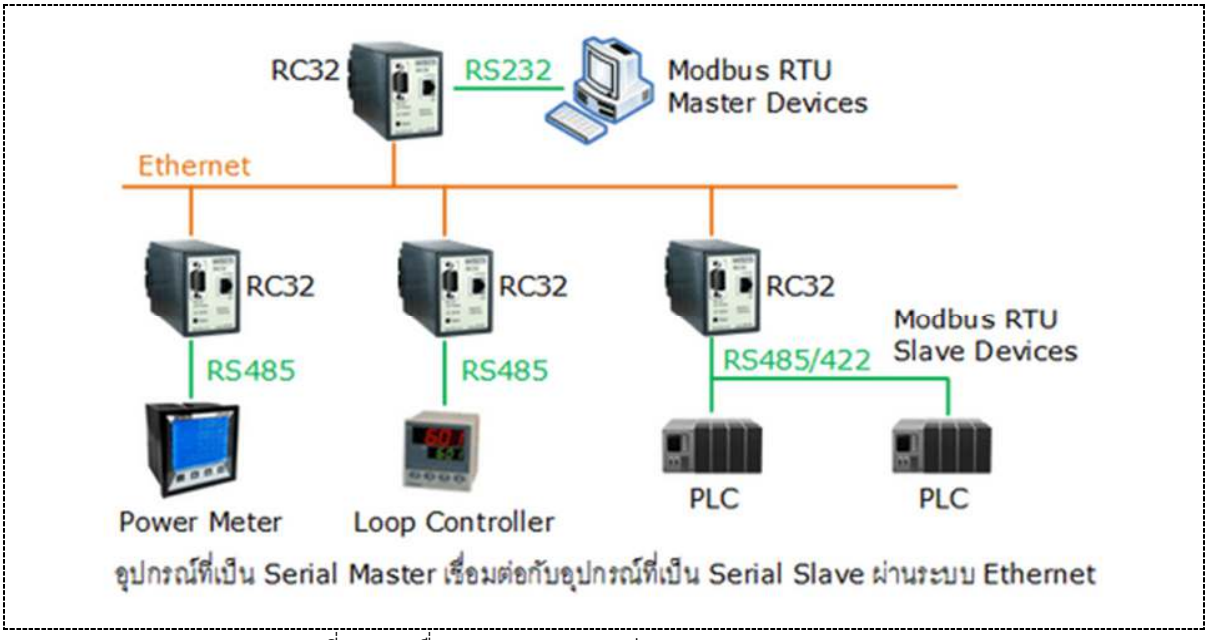
6.4 High-Speed Motion Control System – EtherCAT Product Feature

Delta EtherCAT Motion Control Card PCI-L221-P1 สามารถนำมาประยุกต์ใช้ในการทำเรียลไทม์โมชั่นคอนโทรล และสามารถควบคุมการทำงานได้ถึง 100 Slaves และยังสามารถการันตีการควบคุม 64 Axis ได้ใน 1ms cycle time นอกจากนี้ Delta Electronics Inc ยังผลิต Servo Drive รุ่น ASDA-A2-E ปรับเปลี่ยนให้ Communication Port เป็น EtherCAT ขึ้นมาเฉพาะ



7 Modbus Protocol

คือโปรโตคอลหรือรูปแบบการสื่อสารข้อมูลดิจิตอลแบบอนุกรมรูปแบบหนึ่ง ซึ่งถูกเผยแพร่ครั้งแรกในปี ค.ศ. 1979 โดย Modicon ซึ่งปัจจุบันคือบริษัท Schneider Electric เพื่อใช้มันกับ PLC (Programmable Logic Controllers) ซึ่งทางบริษัทได้เปิดให้ MODBUS เป็น Open Protocol หรือก็คือผู้สนใจสามารถนำโปรโตคอลนี้ไปใช้หรือพัฒนาได้โดยไม่มีค่าใช้จ่ายใดๆ ตั้งแต่นั้นเป็นต้นมา MODBUS จึงกลายเป็นโปรโตคอลที่ได้รับความนิยมและถูกใช้เป็นโปรโตคอลมาตรฐานในระบบอิเล็กทรอนิกส์อุตสาหกรรมจนถึงปัจจุบัน



รูปที่ 36 การสื่อสาร Modbus ระหว่าง Master และ Slave Device

7.1 Modbus ASCII

การรับส่งข้อมูลในโหมด ASCII นั้นมีความแตกต่างจากโหมด RTU ตรงที่ในโหมด RTU ข้อมูลที่จะส่งขนาด 1 ไบต์ นำมารวมกับบิตประกอบต่างๆ ก็สามารถส่งออกไปได้เลย แต่สำหรับโหมด ASCII จะมองข้อมูล 1 ไบต์ นั้นออกมาเป็นตัวอักษร 2 ตัว เช่น ค่า 0x5B ซึ่งเป็นเลขฐานสิบหก ก็จะถูกมองเป็นตัวอักษร ‘5’ และตัวอักษร ‘B’ จากนั้นก็จะทำการค้นหารหัส ASCII ของตัวอักษรทั้ง 2 ตัวนั้น ซึ่งได้แก่ 0x35 สำหรับ ‘5’ และ 0x42 สำหรับ ‘B’ แล้วทำการส่งรหัส ASCII ทั้ง 2 ค่านี้ออกไป ซึ่งจะได้ผลเท่ากับการส่งค่า 0x5B ซึ่งเป็นข้อมูลขนาด 1 ไบต์ ในโหมด RTU

จะเห็นได้ว่าการส่งข้อมูลในโหมด ASCII จะต้องทำงานมากกว่าการส่งข้อมูลในโหมด RTU ซึ่งทำให้อัตราเร็วในการสื่อสารมีค่าต่ำกว่า สาเหตุที่เป็นแบบนี้ก็เพราะว่า โหมด ASCII ได้ถูกออกแบบมาสำหรับอุปกรณ์ที่ไม่มีความสามารถในการกำหนดช่วงระยะห่างของเวลาในการส่งเฟรมข้อมูล อย่างเช่นในโหมด RTU ที่อุปกรณ์สามารถกำหนดได้ว่าจะส่งเฟรมข้อมูลแต่ละเฟรมออกมาด้วยเวลาห่างกันเท่าใด และอุปกรณ์ที่รอรับข้อมูลก็ต้องสามารถตรวจจับและแยกแยะได้ว่าเฟรมข้อมูลแต่ละเฟรมที่รับเข้ามานั้นมีระยะเวลาห่างกันภายในช่วงเวลาที่กำหนดหรือไม่ เพื่อทำให้สามารถตรวจสอบหาจุดเริ่มต้นและจุดสิ้นสุดของเฟรมข้อมูลแต่ละเฟรมได้ แต่ในความเป็นจริงยังมีอุปกรณ์อีกหลายชนิดที่ไม่มีความสามารถพิเศษนี้ จึงต้องใช้วิธีอื่นที่จะช่วยให้สามารถรับรู้จุดเริ่มต้นและจุดสิ้นสุดของเฟรมข้อมูลได้นั่นได้แก่โหมด ASCII ซึ่งในโหมดนี้จะเริ่มต้นเฟรมข้อมูลด้วยการส่งรหัส ASCII ที่กำหนดให้หมายถึง

จุดเริ่มต้น คือ 0x3A ซึ่งตรงกับตัวอักษร ‘:’ ตามด้วยแอดเดรสของ Slave, หมายเลขฟังก์ชัน, ข้อมูล, รหัสตรวจสอง RLC และรหัส ASCII 2 ตัว ที่กำหนดให้หมายถึงจุดสิ้นสุด คือ รหัส 0x0D และ 0x0A คือรหัส CR (Carriage Return) และ LF (Line Feed) ตามลำดับ โดยในขณะที่บัสข้อมูลว่างจากการรับส่งข้อมูล อุปกรณ์ทุกตัวจะคอยตรวจสอบข้อมูลในบัสว่ามีการส่งรหัส ASCII ของ ‘:’ ออกมาหรือไม่ ถ้ามีก็จะรับรู้ว่าขณะนี้ได้มีการเริ่มต้นส่งเฟรมข้อมูลออกมาแล้ว ก็จะเข้ากระบวนการรับข้อมูลต่อไป

Start	Address	Function	Data	LRC	End
1 char :	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2 chars CR,LF

รูปที่ 37 เฟรมข้อมูลการสื่อสารของ Modbus ASCII

- มีการเพิ่มส่วนของ Start โดยต้องขึ้นต้นด้วย “:” ก่อน
- ส่วนของ CRC จะเปลี่ยนเป็น LRC แทน ซึ่งมีวิธีการคำนวณแตกต่างจาก CRC ทำให้ไม่สามารถใช้คำสั่ง Set Features ได้
- เพิ่มส่วน End จำนวน 2 characters คือปิดท้ายด้วย CR+LF
- การส่งค่าทุกอย่างจะส่งเป็นแบบ String หรือ ASCII code นั่นเอง

ASCII Framing

ในโหมด ASCII ข้อความจะเริ่มต้นด้วยอักขระ ‘colon (‘) (ASCII 0x3A) และจบด้วย ‘carriage return - line feed’ (CRLF)’ (ASCII 0x0D และ 0x0A) โดยอักขระที่ยอมให้ส่งทั้งหมดจะอยู่ในรูปแบบเลขฐานสิบหก 0 - 9, A – F

START	ADDRESS	FUNCTION	DATA	LRC CHECK	END
1 CHAR :	2 CHARS	2 CHARS	*n* CHARS	2 CHARS	2 CHARS CRLF

รูปที่ 38 เฟรมข้อมูล

- Address อุปกรณ์ Slave ที่ถูกต้องอยู่ในช่วง 0–247 ทศนิยม อุปกรณ์ Slave แต่ละตัวได้รับการกำหนดอยู่ Address ในช่วง 1–247 เมื่อ Slave มีส่งการตอบสนองของมัน Address ของตัวมันเองจะอยู่ในขอบเขต Address Field ของการตอบสนองเพื่อให้อุปกรณ์ Master ทราบว่าอุปกรณ์ Slave กำลังตอบสนองอยู่และ ใช้ Address 0 ถูกใช้สาหรับการสืบค้นข้อมูล broadcast query (ข้อมูลที่ออกอากาศ)
- Function รหัส Function Field ที่ถูกต้องอยู่ในช่วง 1–255 ทศนิยม เมื่อมีการส่งข้อความจากอุปกรณ์ Master ไปยังอุปกรณ์ Slave โดย function code field จะบอกให้อุปกรณ์ Slave ทราบว่าควรดำเนินการอะไรบ้าง เมื่ออุปกรณ์ Master มีการตอบสนองต่ออุปกรณ์ Slave จะ function code field เพื่อระบุการตอบสนองตามปกติ (ปราศจากข้อผิดพลาด error-free) หรือว่าเกิดข้อผิดพลาดบางประเภท (เรียกว่าการตอบสนองข้อยกเว้น exception)

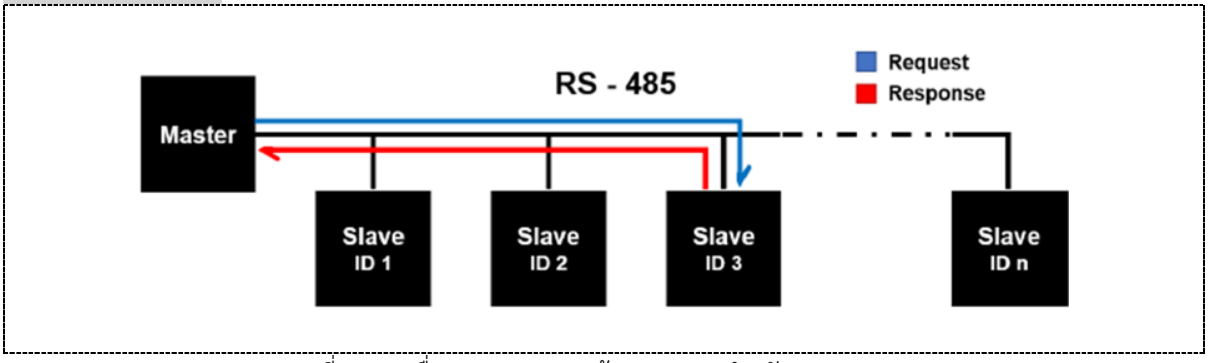
สำหรับการตอบสนองแบบปกติทั่วไปนั้นอุปกรณ์ Slave จะสะท้อน(echoes) โค้ดฟังก์ชันเดิม สำหรับการตอบสนองข้อยกเว้น(exception) อุปกรณ์ Slave จะส่งคืนรหัสที่เทียบเท่ากับโค้ดฟังก์ชันเดิม โดยมีบิตที่สำคัญที่สุดของชุดที่กำหนดให้เป็น Logic 1 โดยอุปกรณ์ Master จะบอกให้ทราบว่ามีข้อผิดพลาดเกิดขึ้นหรือเหตุผลในการยกเว้น ไม่ว่าจะเป็นรหัสฟังก์ชันเฉพาะหรือรหัสฟังก์ชันก์ที่ไม่ขึ้นอยู่กับอุปกรณ์ Slave ก็ตาม โดยตรวจสอบข้อกำหนดสำหรับแต่ละอุปกรณ์ Slave

- Data ของข้อความที่ส่งจากอุปกรณ์ Master ไปยังอุปกรณ์ Slave มีข้อมูลที่ Slave ต้องใช้เพื่อใช้ในการกระทำที่จะกำหนดโดยรหัสฟังก์ชัน (function code) ซึ่ง Data field อาจมีความยาวต่างกันหรืออาจไม่มี (ความยาวเป็นศูนย์) ดูข้อกำหนดสำหรับอุปกรณ์ Slave แต่ละชนิดสำหรับโครงสร้างและความหมายของ Data field
- Error Checking Field

  ASCII เมื่อใช้โหมด ASCII ในส่วน error checking จะประกอบด้วยอักขระ ASCII สองตัว เป็นการตรวจสอบข้อผิดพลาดเกิดขึ้นจากการคำนวณการตรวจสอบความซ้ำซ้อนแบบ Longitudinal Redundancy Check (LRC) ที่ตรวจสอบเนื้อหาของข้อความยกเว้นจุดเริ่มต้น ‘colon’ และจุดสิ้นสุด CRLF

  LRC Checking เมื่อใช้โหมด ASCII ในส่วน error checking เกิดขึ้นจากการคำนวณการตรวจสอบความซ้ำซ้อนแบบ Longitudinal Redundancy Check (LRC) โดยจะตรวจสอบเนื้อหาของข้อความยกเว้นจุดเริ่มต้น ‘colon’ และจุดสิ้นสุด CRLF ใน LRC จะเป็นข้อมูล 8 บิต เป็นการนำบล็อกของบิตข้อมูลมาจัดในตาราง(จัดเป็นแถวและหลัก) โดยการนำบล็อกของข้อมูล 32 บิตมาจัดในตารางให้เป็น 4 แถวแถวละ 8 หลัก แล้วทำการตรวจสอบพาริตี้บิตของหลักทุกหลักแล้วจะได้แถวของข้อมูล 8 บิตขึ้นมาใหม่ 1 แถว การทำพาริตี้บิต บิตที่ 1 ในแถวที่ 5 ได้จากการทำพาริตี้คู่ของบิตแรกในทุก ๆ แถว, พาริตี้บิต บิตที่ 2 ในแถวที่ 5 ได้จากการทำพาริตี้คู่ของบิตที่ 2 ในทุก ๆ แถว และเป็นเช่นนี้จนถึงบิตที่ 8 จากนั้นทำการส่งพาริตี้บิต 8 บิตนี้ต่อจากข้อมูลเดิมส่งไปยังผู้รับ

7.2 Modbus RTU



รูปที่ 39 การสื่อสารแบบอนุกรมด้วย RS-485 สำหรับ Modbus RTU

Modbus RTU คือ โพรโทคอลที่ใช้การสื่อสารแบบอนุกรม (Serial-based Protocol) ด้วยสถาปัตยกรรมการสื่อสารแบบ Master/Slave หรืออาจกล่าวได้ว่าอุปกรณ์ Slave จะไม่ส่งข้อมูล (Response) กลับมาจนกว่าจมีการร้องขอ (Request) จากอุปกรณ์ Master ดังรูปที่ 4.4 Modbus RTU โดยทั่วไปจะใช้การสื่อสารในระดับกายภาพ (Physical Layer) แบบ RS-232 หรือ RS-485 ข้อมูลในโพรโทคอล Modbus จะถูกเก็บ 4 รูปแบบคือ Output coils, Input contacts, Input registers และ Holding registers

โดย 2 แบบแรก Output coils และ Input contacts แต่ละแอดเดรสจะเก็บค่าเพียง 1 บิต หรือมีค่าได้แค่ “0” กับ “1” เปรียบเสมือนค่าการเปิดและปิดของอุปกรณ์รีเลย์และสวิตช์ที่พบได้ในระบบงานอัตโนมัติอุตสาหกรรม

ในขณะที่ 2 แบบหลัง Input registers และ Holding registers สามารถเก็บค่าเป็นตัวเลขได้ถึง 16บิต เปรียบเสมือนค่าที่มาจากอุปกรณ์ตรวจวัดที่ส่งข้อมูลแบบอนาล็อก (Analog)

การสื่อสารของข้อมูลในระบบ Modbus RTU จะรับส่งเป็นชุดข้อมูล โดยที่ใน 1 ชุดข้อมูลนั้นจะประกอบด้วยส่วน 6 ส่วน ดังแสดงในรูปที่ 4.5 ซึ่งเริ่มต้นด้วยชุดบิตเริ่มต้น (Start bits) อ้างอิงถึงการเริ่มต้นชุดข้อมูล ตามด้วยค่าตำแหน่งแอดเดรส (Address) ของอุปกรณ์ที่ต้องการสื่อสารด้วย ตามด้วยชุดสาหรับFunction Code และข้อมูลที่ต้องการ (Data) ต่อด้วยชุดข้อมูลตรวจสอบความผิดพลาด (Cyclic Redundancy Check : CRC) และชุดบิตปิดท้าย (End bits) อ้างอิงถึงการสิ้นสุดข้อมูล

Field Name	Bit length	Function
Start	28	At least 3.5 character times of silence (mark condition)
Address	8	Station address
Function	8	Indicates function code eg. read coils/holding registers
Data	n x 8	Data + length will be filled depending on message type
CRC	16	Cyclic Redundancy Check
End	28	At least 3.5 character times of silence between frames

รูปที่ 40 ชุดข้อมูลสำหรับการสื่อสาร Modbus RTU

1. ฟังก์ชันการทำงานสาหรับ Modbus RTU (Function code)

ชุดฟังก์ชันการทางานสามารถแบ่งหน้าที่ต่างๆ ได้ตามรหัส หรือ Function code รายละเอียดแสดงดังรูปที่ 3โดยหลักๆ แล้วจะมีฟังก์ชันการทางานอยู่ 2 แบบ คือ การอ่าน (Read) และเขียน (Write) โดยสามารถเลือกที่จะอ่านหรือเขียนข้อมูลไปยัง Coils หรือ Contacts สาหรับข้อมูลแบบดิจิตอล (Digital) หรือ “0” กับ “1” และ Registers สาหรับอ่านหรือเขียนข้อมูลแบบแอนะล็อก โดยมีขนาด 16 บิต หรือ ตั้งแต่ 0000 ถึง FFFF

Function Code (DEC)	Action	Data Type	Object Type
01	Read	Single bit	Output Coils
05	Write Single	Single bit	Output Coils
15	Write Multiple	Single bit	Output Coils
02	Read	Single bit	Input Contacts
04	Read	Word (16bit)	Input Registers
03	Read	Word (16bit)	Holding Registers
06	Write Single	Word (16bit)	Holding Registers
16	Write Multiple	Word (16bit)	Holding Registers

รูปที่ 41 รายละเอียดชุดข้อมูล Function Code

2. ตำแหน่งแอดเดรสของ Modbus RTU (Address)

ตำแหน่งแอดเดรสใน Modbus RTU จะมีขนาด 16 บิต หรือ 65535 ตำแหน่ง ในแต่ละรูปแบบการทำงาน โดยที่ Output coils (ตำแหน่งแอดเดรสจะเริ่มต้นที่ 000001) Input contacts (ตำแหน่งแอดเดรสจะเริ่มต้นที่ 100001) Input registers (ตำแหน่งแอดเดรสจะเริ่มต้นที่ 300001) และ Holding registers (ตำแหน่งแอดเดรสจะเริ่มต้นที่ 400001) ดังรูปที่ 18

หมายเหตุ สาหรับอุปกรณ์รุ่นเก่าอาจจะมีได้เพียง 9999 ตำแหน่งในแต่ละช่วง

Register Number (DEC)	Register Address (HEX)	Extended Register Number (DEC)	Extended Register Address (HEX)	Type	Object Type
00001-09999	0000 to 270E	000001-065535	0000 to FFFF	Read-Write	Output Coils
10001-19999	0000 to 270E	100001-165535	0000 to FFFF	Read-Only	Input Contacts
30001-39999	0000 to 270E	300001-365535	0000 to FFFF	Read-Only	Input Registers
40001-49999	0000 to 270E	400001-465535	0000 to FFFF	Read-Write	Holding Registers

รูปที่ 42 ตำแหน่งแอดเดรสใน Modbus RTU โดยแบ่งตามรูปแบบการทำงาน

3. ชุดข้อมูล (Data)

ในส่วนชุดข้อมูล Data Field นั้นจะถูกแบ่งเป็น 2 ชุด คือ ชุดคำสั่งสำหรับการอ่าน (Read Command) ตามรูปที่ 4.8 และชุดคำสั่งสำหรับการเขียน (Write Command) ตามรูปที่ 4.9 โดยชุดคำสั่งทั้ง 2 จะถูกส่งจากอุปกรณ์ที่ทำหน้าที่เป็น Master เท่านั้น เพื่อสั่งการไปยังอุปกรณ์ Slave ที่ต้องการสื่อสาร

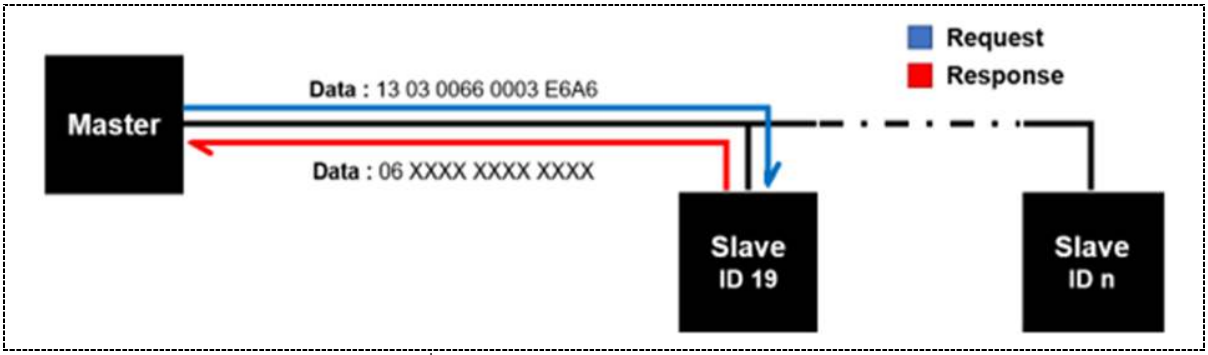
Read Command	
Request Message	start register address (2 bytes) + no. of registers (2 bytes)
Response Message	byte count (1 byte) + data (no. of registers * 2 bytes)

รูปที่ 43 ชุดคำสั่งสำหรับการอ่าน (Read Command)

Write Command	
Request Message	start register address (2 bytes) + no. of registers (2 bytes) + byte count (1 byte) + data (no. of registers * 2 bytes)
Response Message	start register address (2 bytes) + no. of registers (2 bytes)

รูปที่ 44 ชุดคำสั่งสำหรับการเขียน (Write Command)

ตัวอย่าง การอ่านค่าของ Holding register ที่แอดเดรส 40103 ถึง 40105 จากอุปกรณ์ Slave หมายเลข 19 ดังนั้น Frame Message (ไม่รวม Start และ End bits) ที่ถูกส่งไป คือ 13 03 0066 0003 E6A6 โดยที่



รูปที่ 45 การ รับ-ส่ง เฟรมข้อมูล Modbus RTU

ชุดข้อความสำหรับการอ่านค่าจาก Holding register (Request)

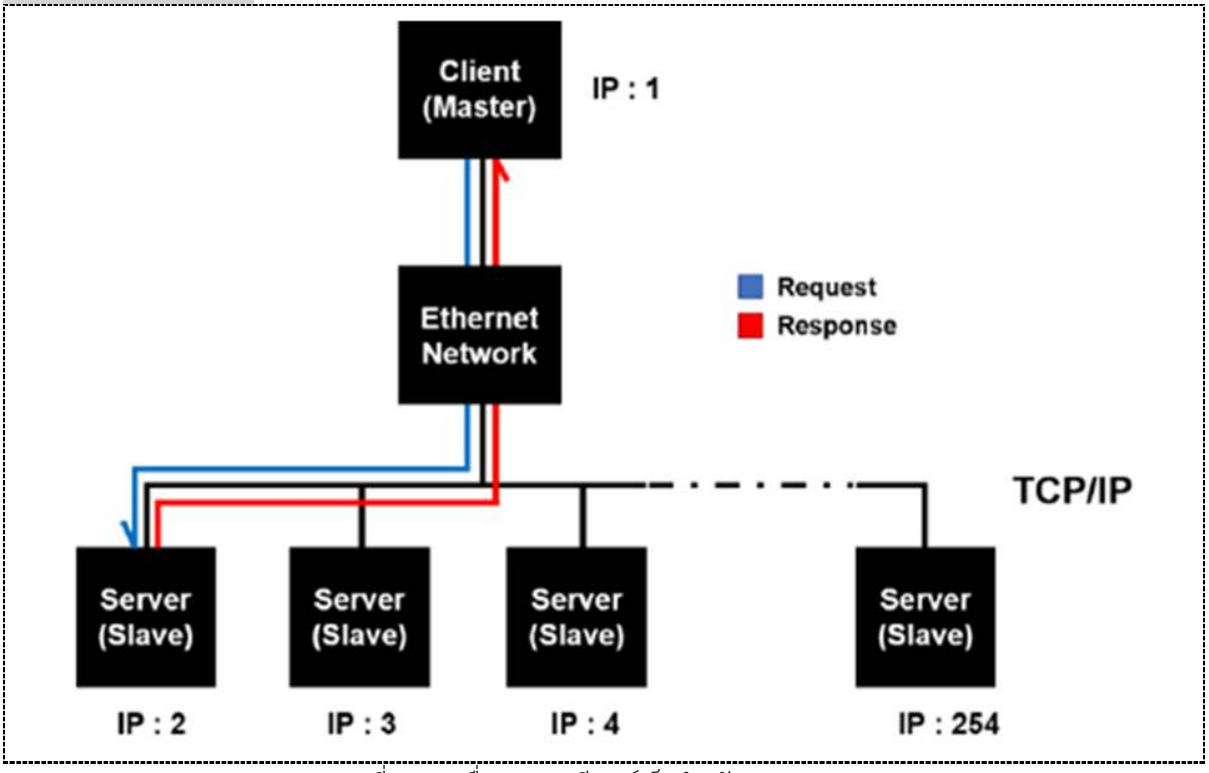
- 13 คือ Station address (19 DEC = 13 HEX)
- 03 คือ Function code (การอ่านค่าที่ Holding registers)
- 0066 คือ Address ของ register ตัวแรก (40103 - 40001 = 102 DEC= 66 HEX)
- 0003 คือ จำนวน Registers ที่ต้องการอ่าน (ทั้งหมด 3 ตัว คือ 40103 ถึง 40105)
- E6A6 คือ ค่า CRC (Cyclic Redundancy Check) สำหรับเช็คความผิดพลาดของชุดข้อมูล

ชุดข้อความที่ตอบกลับมา (Response)

ส่วน Frame message (ไม่รวม Start และ End bits) ที่ตอบกลับมาคือ 06 XXXX XXXX XXXX

- 06 คือ จานวน byte ของข้อมูลที่ตอบกลับมา
- XXXX XXXX XXXX คือ ข้อมูลของทั้ง 3 ตำแหน่ง ที่ตอบกลับมา (ตำแหน่งละ 2 bytes)

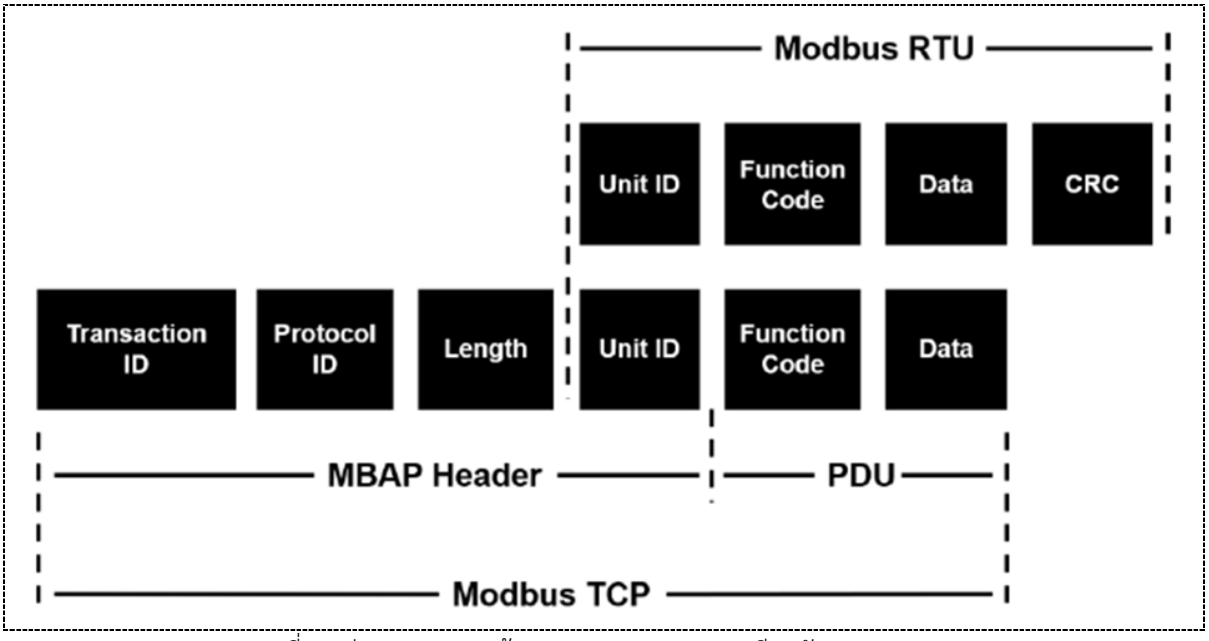
7.3 Modbus TCP/IP



รูปที่ 46 การสื่อสารแบบอีเทอร์เน็ตสำหรับ Modbus TCP

Modbus TCP คือ โพรโทคอลที่ครอบ Modbus RTU เพื่อใช้การสื่อสารแบบอีเทอร์เน็ต (Ethernet-based protocol) ด้วย TCP/IP (Transmission control protocol) ที่พอร์ต (Port) 502 แทนการใช้การสื่อสารแบบอนุกรม ดังรูปที่ 4.12 ทำให้อุปกรณ์สามารถสร้างการสื่อสารผ่านเครือข่ายเฉพาะบริเวณ (Local area network : LAN) หรือ เครือข่ายอินเตอร์เน็ต (Internet network) รวมไปถึงการเชื่อมต่อแบบไร้สาย (Wireless) โดยมีอุปกรณ์กระจายสัญญาณ (Router หรือ Access point) เป็นตัวกลาง

ในการเชื่อมต่อ โดยชุดข้อความใน Modbus TCP มีรายละเอียดดังรูปที่ 4.13 และ 4.14 เริ่มต้นข้อมูลด้วย Modbus application protocol (MBAP) Header ซึ่งประกอบด้วย Transaction ID, Protocol ID, Length, Unit ID ซึ่งเพิ่มเติมขึ้นมาจาก Modbus RTU ส่วนชุดข้อมูล Function code และ Data จะยังคงเหมือนเดิม ยกเว้นชุดข้อมูล CRC สำหรับเช็คความผิดพลาดจะไม่มี แต่เปลี่ยนไปใช้ของ Ethernet ใน Data link layer แทน



รูปที่ 47 ส่วนประกอบชุดข้อมูลของ Modbus TCP เทียบกับ Modbus RTU

Area Name		Area Size	Description
MBAP header (MODBUS® application header)	Transaction ID	2 bytes	Used by the master for matching of the response message from the slave.
	Protocol ID	2 bytes	Indicates the protocol of the PDU (protocol data unit). Stores 0 in the case of MODBUS® /TCP.
	Message length	2 bytes	Stores the message size in byte unit. The message length after this field is stored. (See the above figure.)
	Module ID	1 byte	Used to specify the slave connected to the other line, e.g. MODBUS® serial protocol.
PDU (Protocol data unit)	Function code	1 byte	The master specifies the processing to be performed for the slave.
	Data	1 to 252 bytes	[When master sends request message to slave] Stores the requested processing. [When slave sends response message to master] Stores the result of processing execution.

รูปที่ 48 รายละเอียดของแต่ละ Field ในหนึ่งเฟรมของ Modbus TCP

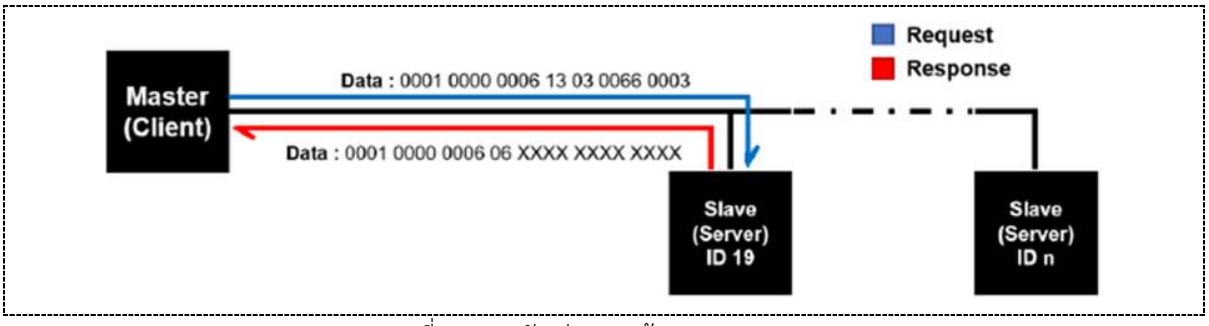
Modbus Function Formats

Primary tables	Object type	Type of	Comments
Discretes Input	Single bit	Read-Only	This type of data can be provided by an I/O system.
Coils	Single bit	Read-Write	This type of data can be alterable by an application program.
Input Registers	16-bit word	Read-Only	This type of data can be provided by an I/O system
Holding Registers	16-bit word	Read-Write	This type of data can be alterable by an application program.

รูปที่ 49 Modbus Function Formats

- Data Address จะถูกใช้ในข้อความค้นหา Modbus เมื่ออ่านหรือแก้ไขข้อมูล มีการใช้ข้อมูลสี่ประเภทคือ Coil, Input Status, Input Register และ Holding Register.
- Coil ใช้เพื่อบังคับให้สถานะ ON / OFF ของเอาต์พุตแบบไม่ต่อเนื่อง (DO) ไปที่ฟิลด์หรือปรับเปลี่ยนโหมดหรือสถานะของอุปกรณ์ Slave โดยข้อมูล Coil เป็น ON หรือ OFF ซึ่งสามารถอ่านและแก้ไขได้ Addresses ที่ใช้ได้จะอยู่ในช่วง 1–9999
- Input Status สถานะการป้อนข้อมูลใช้สำหรับสถานะเปิด / ปิดของอินพุตแบบไม่ต่อเนื่อง (DI) จากฟิลด์หรือสถานะของอุปกรณ์ slave สถานะการป้อนข้อมูลเป็น ON หรือ OFF ซึ่งสามารถอ่านได้เท่านั้น Addresses ที่ใช้ได้จะอยู่ในช่วง 10001–19999
- Input Register ใช้สำหรับค่า Analog inputs (AI) จากฟิลด์หรือข้อมูลของอุปกรณ์ Slave ใส่ข้อมูล Register มีความยาว 16 บิตซึ่งสามารถอ่านได้เท่านั้น Addresses ที่ใช้ได้จะอยู่ในช่วง 30001–39999 ข้อมูลแบบ Floating or double-floating สามารถจัดการเมื่อกำหนดที่อยู่ติดต่อกัน

ตัวอย่าง การอ่านค่าของ holding register # 40103 ถึง 40105 จาก slave หมายเลข 19 เหมือนในตัวอย่างของ Modbus RTU ที่กล่าวมาก่อนหน้านี้ ในกรณีของ Modbus TCP frame message จะเป็น0001 0000 0006 13 03 0066 0003



รูปที่ 50 การ รับ-ส่ง เฟรมข้อมูล Modbus TCP

- 0001 คือ Transaction identifier
- 0000 คือ Protocol identifier
- 0006 คือ จานวน byte ของข้อมูลที่ต่อจากไบต์นี้
- 13 คือ Module identifier หรือ station address (19 ในระบบเลขฐาน 10 จะมีค่าเท่ากับ 13 ในเลขฐาน 16)
- 03 คือ Function code (การอ่านค่าจาก Holding registers)
- 0066 คือ Address ของ register ตัวแรก (40103-40001 = 102 = 66 hex)
- 0003 คือ จานวน Registers ที่ต้องการอ่าน (ทั้งหมด 3 ตัว คือ 40103 40104 และ 40105)