

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

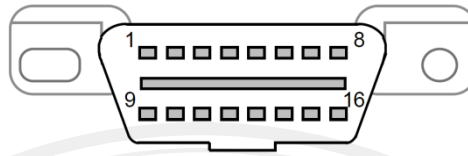
2.1 การวินิจฉัยรถยนต์ (On-Board Diagnostic, OBD)

การวินิจฉัยรถยนต์เริ่มต้นในปี ค.ศ. 1960 จากการที่มลรัฐแคลิฟอร์เนียได้จัดตั้งหน่วยงานพิเศษที่เรียกว่า CARB (California Air Resources Board) ซึ่งมีจุดประสงค์เพื่อเฝ้าสังเกตสถานะมลพิษในเขตตัวเมืองและได้ตรากฎหมายรวมถึงข้อกำหนดต่างๆ เพื่อให้ควบคุมปริมาณมลภาวะจากการเผาไหม้ของเครื่องยนต์ โดยการเริ่มใช้ชุดควบคุมด้วยอิเล็กทรอนิกส์หรืออีซียู (ECU) ในระบบควบคุมการเผาไหม้ ระบบการส่งจ่ายน้ำมันและระบบการส่งจ่ายกำลัง เป็นต้น จากนั้นในปี ค.ศ.1989 ด้วยความร่วมมือระหว่างหน่วยงาน CARB, EPA (Environment Protection Agent) และ SAE (Society of Automotive Engineering) ได้กำหนดเป็นมาตรฐานในการวินิจฉัยรถยนต์ โดยเรียกมาตรฐานนี้ว่า “ระบบ OBD-I” และถูกพัฒนาต่อเนื่องจนเป็นระบบ OBD-II ในปัจจุบัน [1]

ระบบ OBD-II ทำหน้าที่ควบคุมและเฝ้าระวังการทำงานของอีซียูที่ต่อร่วมกันเป็นเครือข่ายในรถยนต์ (In-Vehicle Network) โดยสื่อสารข้อมูลต่างๆ ผ่านบัสสื่อสารข้อมูลที่รองรับ นอกจากนี้ยังเพิ่มความสามารถในการวินิจฉัยรถยนต์ซึ่งมีผลต่อการปรับปรุงประสิทธิภาพของการซ่อมบำรุงรถยนต์ เช่น ทดสอบการทำงานของเครือข่ายเซนเซอร์และมอเตอร์ควบคุม บันทึกลงและรายงานผลการวินิจฉัย รวมถึงรายงานค่าที่วัดได้จากเซนเซอร์และสถานะของมอเตอร์ควบคุม เป็นต้น ทั้งนี้ระบบ OBD-II มีมาตรฐานและข้อกำหนดสากล เพื่อเป็นแนวทางในการวินิจฉัยรถยนต์ [7] ซึ่งได้แก่ ลักษณะและขาสัญญาณของตัวเชื่อมต่อเพื่อวินิจฉัยปัญหา ฟังก์ชันและโหมดการทำงานของสแกนทูล ขั้นตอนการวินิจฉัย บัสสื่อสารข้อมูลที่รองรับและการรหัสอาการเสียของรถยนต์ด้วยรหัส DTC เป็นต้น

2.1.1 ข้อกำหนดของระบบ OBD-II

1) *SAE J1962* คือมาตรฐานสำหรับกำหนดตำแหน่งและขาสัญญาณของตัวเชื่อมต่อเพื่อวินิจฉัยปัญหา (OBD connector) ซึ่งมีขาสัญญาณ 16 ขาและติดตั้งที่ตำแหน่งใดๆ ภายในห้องโดยสารรถยนต์ (ส่วนมากติดตั้งอยู่ใต้ส่วนควบคุมฝั่งผู้ขับขี่) มีโครงสร้างและตำแหน่งขาสัญญาณดังรูป 2.1 และอธิบายชื่อแต่ละขาสัญญาณดังตาราง 2.1 ในการใช้งานจริงมีเพียงสองหรือสามสัญญาณที่ถูกใช้พร้อมกัน ซึ่งขึ้นอยู่กับบัสสื่อสารข้อมูลที่รถยนต์รองรับ



รูป 2.1 ลักษณะและตำแหน่งขาสัญญาณของตัวเชื่อมต่อเพื่อวินิจฉัยปัญหา [7]

ตาราง 2.1 ชื่อสัญญาณแต่ละขาของตัวเชื่อมต่อเพื่อวินิจฉัยปัญหา [7]

ขา	ชื่อสัญญาณ	ขา	ชื่อสัญญาณ
1	-	9	-
2	J1850 Bus+	10	J1850 Bus-
3	-	11	-
4	Chassis Ground	12	-
5	Signal Ground	13	-
6	CAN-H	14	CAN-L
7	ISO 9141-2 K Line	15	ISO 9141-2 L Line
8	-	16	Battery Power

2) **SAE J1978** คือมาตรฐานใช้กำหนดและอธิบายฟังก์ชันพื้นฐานของสแกนทูลประกอบด้วย

2.1) ฟังก์ชันตรวจสอบและควบคุมลำดับการสื่อสาร รวมถึงชนิดของบัสสื่อสาร ข้อมูลที่รถยนต์รองรับโดยอัตโนมัติ

2.2) ฟังก์ชันอ่านและแสดงสถานะ ผลการวินิจฉัยรถยนต์และค่าพารามิเตอร์ดังนี้

ก) รหัสปัญหาหรือรหัส DTC

ข) พารามิเตอร์ที่เกี่ยวข้องกับการปล่อยมลพิษจากการเผาไหม้เชื้อเพลิง
(Emission related parameter)

ค) Emission freeze frame data

ง) พารามิเตอร์ที่ถูกทดสอบด้วยโหมดการทดสอบที่ 06 (เลขฐานสิบหก)

จ) พารามิเตอร์จากการทดสอบอื่นๆ ตามที่ระบุในมาตรฐาน SAE J1979

2.3) ฟังก์ชันลบรหัส DTC Emission freeze frame data และผลการวินิจฉัย

รถยนต์

3) **SAE J1979** คือมาตรฐานสำหรับใช้อธิบายโหมดการทดสอบหรือโหมดทำงานของระบบ OBD-II ซึ่งทุกการร้องขอทำฟังก์ชันจากสแกนทูลต้องระบุโหมดการทำงาน (Mode) และพารามิเตอร์ที่เกี่ยวข้อง (Parameter Identification, PID) ตามข้อกำหนดของโหมดที่เลือกใช้ โดยได้กำหนดมาตรฐานการวินิจฉัยรถยนต์ไว้ 9 โหมดดังนี้ (ชื่อโหมดแสดงด้วยเลขฐานสิบหก)

- 3.1) โหมด 01 สำหรับอ่านข้อมูลปัจจุบันของพารามิเตอร์ที่ระบุ
- 3.2) โหมด 02 สำหรับอ่านข้อมูลของพารามิเตอร์ที่กำหนดให้เฝ้าระวังไว้ (Freeze Frame Data)
- 3.3) โหมด 03 สำหรับอ่านรหัส DTC ที่อียูตรวจพบและบันทึกไว้
- 3.4) โหมด 04 สำหรับลบหรือคืนค่ารหัส DTC สถานะไฟ MIL และข้อมูลอื่นๆ ที่อียูบันทึกไว้
- 3.5) โหมด 05 สำหรับตรวจสอบการทำงานของเซนเซอร์ออกซิเจน (Oxygen sensor)
- 3.6) โหมด 06 สำหรับตรวจสอบการทำงานของส่วนควบคุมที่มีผลลัพธ์แบบไม่ต่อเนื่อง เช่น Catalyst, Exhaust Gas Re-circulation (EGR) และ Evaporative System เป็นต้น
- 3.7) โหมด 07 สำหรับตรวจสอบการทำงานของส่วนควบคุมที่มีผลลัพธ์แบบต่อเนื่อง เช่น Fuel Trim, Misfire และ Comprehensive Components เป็นต้น
- 3.8) โหมด 08 โหมดควบคุมพิเศษ
- 3.9) โหมด 09 สำหรับอ่านข้อมูลของรถยนต์ เช่น หมายเลขตัวถังและรหัสรุ่นรถยนต์ เป็นต้น

รถยนต์แต่ละคันที่มีระบบ OBD-II ไม่จำเป็นต้องรองรับการทำงานทั้ง 9 โหมด นอกจากนี้บริษัทผู้ผลิตรถยนต์สามารถเพิ่มโหมดและพารามิเตอร์ของแต่ละบริษัทเอง แต่ต้องสนับสนุนการทำงานในโหมด 01 ด้วยพารามิเตอร์ 00 (เลขฐานสิบหก) ซึ่งใช้แสดงพารามิเตอร์ที่สามารถอ่านค่าได้ (ในการทำงานแต่ละโหมด พารามิเตอร์ 00 ถูกสำรองไว้เพื่อรหัสว่าโหมดดังกล่าว สามารถเข้าถึงพารามิเตอร์ได้บ้าง)

4) **SAE J2012** คือมาตรฐานที่อธิบายความหมายของรหัส DTC

5) **บัสสื่อสารข้อมูล** คือมาตรฐานที่ใช้กำหนดชนิดของบัสสื่อสารข้อมูลระหว่างระบบ OBD-II กับสแกนทูล ตาราง 2.2 แสดงบัสสื่อสารข้อมูลที่รองรับการทำงานกับระบบ OBD-II ในปัจจุบัน ทั้งนี้บัส ISO 15765-4 ซึ่งเป็นบัสที่ครอบคลุมการทำงานของบัส CAN ถูกกำหนดให้เป็นบัสเดียวที่ใช้ทำงานกับรถยนต์ที่ออกแบบและผลิตใหม่ตั้งแต่ปี ค.ศ. 2008 [1, 7]

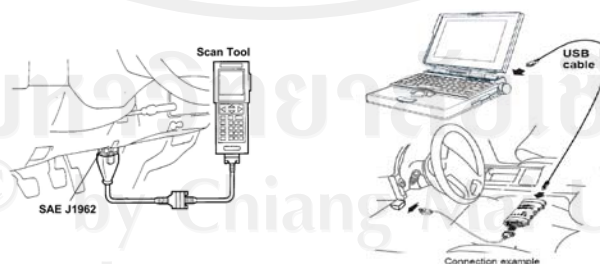
ตาราง 2.2 บัสสื่อสารข้อมูลในระบบ OBD-II รองรับ

ลำดับ	ชื่อบัส	ความเร็วในการรับ-ส่งข้อมูล
1	SAE J1850 (PWM)	41.6 กิโลบิตต่อวินาที
2	SAE J1850 (VPW)	10.4 กิโลบิตต่อวินาที
3	ISO 9141-2	10.4 กิโลบิตต่อวินาที
4	ISO 14230-4 (KWP 2000)	10.4 กิโลบิตต่อวินาที
5	ISO 15765-4 (CAN)	250, 500 กิโลบิตต่อวินาที

2.1.2 ขั้นตอนในการวินิจฉัยปัญหาและซ่อมบำรุงรถยนต์ทั่วไป

การวินิจฉัยปัญหาและซ่อมบำรุงรถยนต์สามารถสรุปเป็นขั้นตอนได้ดังนี้

- 1) ระบบ OBD-II ตรวจพบและบันทึกข้อมูลที่เกี่ยวข้องกับความเสียหายในรูปแบบของรหัส DTC
- 2) ระบบ OBD-II ประเมินความเสียหายและแจ้งเตือนผู้ขับขี่ด้วยไฟสถานะบนแผงหน้าปัดรถยนต์
- 3) ระบบ OBD-II ปรับแต่งการทำงานของรถยนต์ให้เป็นปกติด้วยการคำนวณค่าพารามิเตอร์ชดเชยให้กับส่วนที่เสียหาย จึงทำให้สามารถใช้งานรถยนต์ได้แต่จำกัดความสามารถ เช่น มีความเร็วหรือกำลังเครื่องยนต์ลดลง เป็นต้น
- 4) ช่างเทคนิคเชื่อมต่อสแกนทูลกับตัวเชื่อมต่อเพื่อวินิจฉัยปัญหา (SAE J1962) มีลักษณะการต่อใช้งานดังรูป 2.2



รูป 2.2 ลักษณะการต่อใช้งานสแกนทูล [7]

- 5) ช่างเทคนิคเลือกฟังก์ชันและโหมดการทำงานของสแกนทูลเพื่อวินิจฉัยปัญหารถยนต์ (ค้นหาสาเหตุและตรวจสอบความเสียหายที่ไม่สามารถแจ้งเตือนผ่านแผงหน้าปัดรถยนต์)

6) ช่างเทคนิควิเคราะห์ผลการวินิจฉัยและรหัส DTC ที่ตรวจพบ จากนั้นทำการซ่อมบำรุงรถยนต์และทดสอบการทำงานของรถยนต์หลังจากการซ่อมบำรุง (Readiness Test)

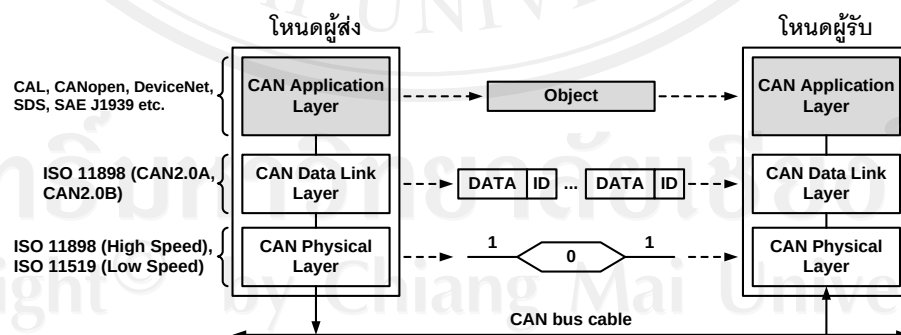
2.2 บัส CAN (Controller Area Network Bus)

2.2.1 โครงสร้างและลักษณะทั่วไปของบัส CAN

บัสCAN [8] เป็นบัสสื่อสารข้อมูลแบบอนุกรมด้วยสัญญาณแบบผลต่างแรงดัน (Differential Voltage Signal) ที่ออกแบบสำหรับระบบอิเล็กทรอนิกส์ในรถยนต์ เพื่อเพิ่มความน่าเชื่อถือในการสื่อสารข้อมูล ความปลอดภัย และประสิทธิภาพในการทำงาน โดยสามารถสรุปจุดเด่นของบัส CAN ดังนี้

- 1) ใช้สายสัญญาณเพียง 2 เส้น ทำให้ลดความซับซ้อนในการเชื่อมต่อระหว่างอุปกรณ์
- 2) สนับสนุนการทำงานแบบเวลาจริงด้วยอัตราเร็วข้อมูลสูงสุดถึง 1 เมกะบิตต่อวินาที
- 3) สื่อสารข้อมูลเป็นเครือข่ายระหว่างอุปกรณ์โดยไม่ต้องมีหมายเลขที่อยู่ของอุปกรณ์
- 4) อุปกรณ์ทุกตัวในเครือข่ายสามารถร้องขอใช้งานบัสได้พร้อมกันหลายตัว (Multi-master)
- 5) มีระบบป้องกันและตรวจสอบความผิดพลาดในการสื่อสารข้อมูล

จากสมบัติเด่นดังกล่าวทำให้บัส CAN ถูกนำไปประยุกต์ใช้งานอย่างแพร่หลายในหลายๆ ด้าน จึงได้มีการกำหนดมาตรฐานให้กับบัส CAN โดยครอบคลุมการทำงานของ CAN ในสองชั้นแรกตามแบบจำลอง ISO/OSI คือชั้นกายภาพ (Physical Layer) และชั้นเชื่อมโยงข้อมูล (Data Link Layer) ดังแสดงในรูป 2.3

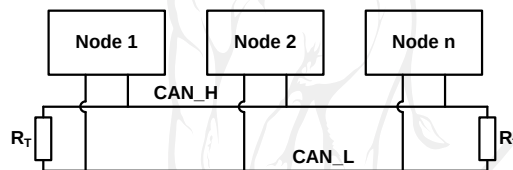


รูป 2.3 โครงสร้างการทำงานของบัส CAN ตามแบบจำลอง ISO/OSI [8]

มาตรฐานชั้นกายภาพของบัส CAN กล่าวถึงส่วนประกอบทางกายภาพของบัสได้แก่ ตัวกลางที่ใช้ส่งผ่านสัญญาณข้อมูล ตัวเชื่อมต่อและวิธีเชื่อมต่อสายสัญญาณ รวมถึงสมบัติทางไฟฟ้าของสัญญาณข้อมูล (แรงดัน / กระแส) ของสัญญาณ โดยส่วนประกอบทางกายภาพดังกล่าวได้ถูก

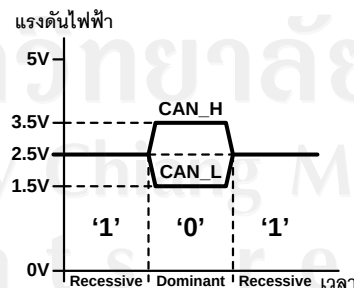
กำหนดเป็น 2 มาตรฐาน คือ ISO 11519 (ความเร็วต่ำ, 5 ถึง 125 กิโลบิตต่อวินาที) และ ISO 11898 (ความเร็วสูง, สูงสุดถึง 1 เมกะบิตต่อวินาที) สำหรับมาตรฐานชั้นเชื่อมโยงข้อมูลกล่าวถึงโพรโทคอลและรูปแบบของข้อมูลที่ใช้สื่อสารในเชิงตรรกะ รวมถึงการป้องกันความผิดพลาดของข้อมูล โดยได้แก้ไขและขยายข้อกำหนดในมาตรฐาน ISO 11898 และจัดทำเป็น 2 เวอร์ชันคือ CAN 2.0A และ CAN 2.0B [8] สำหรับชั้นสื่อสารที่สูงกว่าสองชั้นแรกจนถึงชั้นประยุกต์ใช้งาน (Application Layer) นั้นไม่ได้กำหนดเป็นมาตรฐานร่วมกับสองชั้นแรก แต่มีการสร้างโพรโทคอลสำหรับประยุกต์ใช้งานบนพื้นฐานการทำงานของสองชั้นแรกไว้หลายรูปแบบขึ้นอยู่กับลักษณะงานที่นำไปใช้ เช่น CAL, CANopen, DeviceNet, SDS, CANKingdom และ SAE J1939 เป็นต้น [9]

บัส CAN มีรูปแบบการสื่อสารข้อมูลเป็นแบบอนุกรม ด้วยสัญญาณแบบผลต่างแรงดัน โดยใช้สายสัญญาณ 2 เส้นแบบบิดเกลียวคู่ (Twisted-pair Cable) และมีลักษณะการเชื่อมต่อสายสัญญาณ (Topology) ดังแสดงในรูป 2.4 อุปกรณ์ทุกตัวถูกต่ออยู่บนสายสัญญาณคู่เดียวกันและมีเทอร์มินัลชันอิมพีแดนซ์ (Termination Impedance, R_T) ปิดปลายคู่สายสัญญาณทั้งสองข้าง



รูป 2.4 ลักษณะการเชื่อมต่ออุปกรณ์ของบัส CAN

การสื่อสารข้อมูลด้วยผลต่างแรงดันของสัญญาณ สถานะของสัญญาณขณะนั้นได้จากการเปรียบเทียบระดับแรงดันไฟฟ้าจากสายสัญญาณทั้งสองเส้นคือ CAN_H และ CAN_L ซึ่งมีอยู่ 2 สถานะคือ สถานะรีเซสซีฟ (Recessive) และสถานะ โดมิแนนต์ (Dominant) ดังแสดงในรูป 2.5



รูป 2.5 ลักษณะของสัญญาณข้อมูลแต่ละสถานะ

สถานะของสัญญาณมีสถานะเป็นรีเซสซีฟหรือแทนด้วยข้อมูลที่มีลอจิกเป็น '1' ถ้าผลต่างของแรงดันระหว่าง CAN_H และ CAN_L เป็น 0 โวลต์ (หรือ ≤ 0.5 โวลต์) และมีสถานะเป็นโด-

มีแชนแนล หรือแทนด้วยข้อมูลที่มีลอจิกเป็น '0' ถ้าผลต่างเป็น 2 โวลต์ (หรือ ≥ 0.9 โวลต์) ข้อดีของวิธีการสื่อสารข้อมูลโดยใช้ผลต่างแรงดันของคู่สายสัญญาณคือ ช่วยลดการรบกวนจากผลของ EMI (Electromagnetic Interference) ทำให้สามารถสื่อสารข้อมูลในอัตราเร็วที่สูงและได้ระยะไกลมากขึ้น ทั้งนี้อัตราเร็วในการส่งข้อมูลแปรผกผันกับระยะทางที่ส่งได้ดังแสดงในตาราง 2.3 ตัวอย่างเช่น ที่อัตราการส่งข้อมูลสูงสุด 1 เมกะบิตต่อวินาที ความยาวของสายส่งต้องไม่เกิน 40 เมตร เป็นต้น

ตาราง 2.3 ความสัมพันธ์ของสมบัติสายส่งกับอัตราเร็วข้อมูล

สมบัติของสายส่ง			เทอร์มินเนชัน อิมพีแดนซ์	อัตราเร็วข้อมูลสูงสุด
ความยาว	ขนาด	ความต้านทาน		
0 – 40 m	0.25 – 0.34 mm ²	70 mΩ/m	124 Ω (1%)	1 Mbit/s ที่ 40 m
40 – 300 m	0.34 – 0.6 mm ²	<60 mΩ/m	127 Ω (1%)	500 kbit/s ที่ 100 m
300 – 600 m	0.5 – 0.6 mm ²	<40 mΩ/m	150 – 300 Ω	100 kbit/s ที่ 500 m
600 – 1000 m	0.75 – 0.8 mm ²	<26 mΩ/m	150 – 300 Ω	50 kbit/s ที่ 1 km

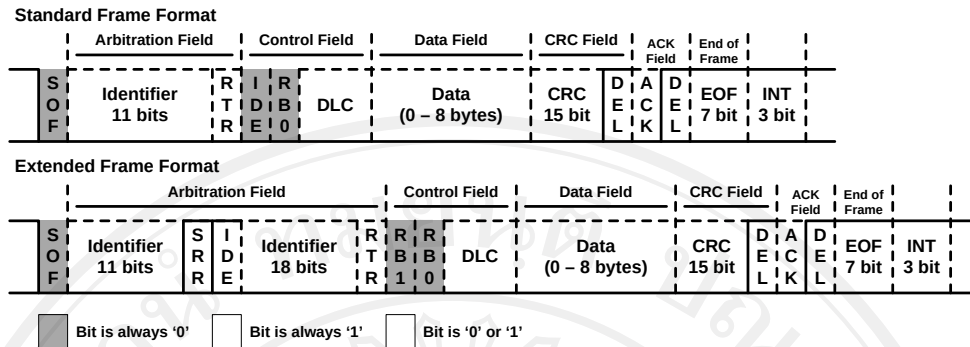
2.2.2 เฟรมสื่อสารข้อมูลของบัส CAN

รูปแบบการสื่อสารข้อมูลในบัส CAN ข้อมูลที่ต้องการส่งถูกแบ่งออกเป็นเฟรม (Frame) หรือข้อความ (Message) สามารถจำแนกออกเป็น 4 ประเภทได้แก่

- 1) เฟรมข้อมูล (Data Frame) คือเฟรมสำหรับส่งข้อมูลตามปกติ
- 2) เฟรมร้องขอข้อมูล (Remote Frame) คือเฟรมสำหรับร้องขอข้อมูลจากโหนดอื่น
- 3) เฟรมแสดงความผิดพลาด (Error Frame) คือเฟรมแสดงการตรวจพบความผิดพลาด
- 4) เฟรมโอเวอร์โหลด (Overload Frame) คือเฟรมแสดงการทับซ้อนของข้อมูลที่ได้รับ

ในเนื้อหาส่วนนี้กล่าวถึงเฟรมข้อมูลและเฟรมร้องขอข้อมูล ซึ่งมีรายละเอียดของเฟรมที่คล้ายกันแต่แตกต่างกันที่ขนาดของฟิลด์ข้อมูล สำหรับเฟรมแสดงความผิดพลาดและเฟรมโอเวอร์โหลดรวมถึงรายละเอียดส่วนย่อยอื่นๆ สามารถศึกษาเพิ่มเติมได้จากเอกสารอ้างอิง [8]

เฟรมข้อมูลแบ่งออกเป็นสองรูปแบบตามขนาดของรหัสประจำตัวของเฟรม (Identifier) คือเฟรมข้อมูลแบบมาตรฐาน (Standard Frame Format) และเฟรมข้อมูลแบบขยาย (Extended Frame Format) ดังแสดงในรูป 2.6



รูป 2.6 เฟรมข้อมูลของบัส CAN

สามารถจำแนกส่วนประกอบสำคัญของเฟรมข้อมูลดังนี้

- 1) *SOF (Start of Frame)* คือบิตเริ่มต้นของเฟรม มีสถานะเป็นโดมิแนนต์หรือลอจิก '0' เสมอ เป็นบิตที่ทุกโหนดใช้ในการเริ่มต้นเข้าจังหวะเพื่อเริ่มการรับหรือส่งข้อมูลในบัส
 - 2) *Identifier* คือรหัสประจำตัวของเฟรมหรือข้อความ ใช้จำแนกรูปแบบของเฟรมข้อมูลแบบมาตรฐาน (รหัสประจำตัวมีขนาด 11 บิต) และเฟรมข้อมูลแบบขยาย (รหัสประจำตัวมีขนาด 29 บิต)
 - 3) *RTR (Remote Transmission Request)* คือบิตที่ใช้ระบุว่าเฟรมข้อมูล (โดมิแนนต์หรือลอจิก '0') หรือเฟรมร้องขอข้อมูล (รีเซตสีฟหรือลอจิก '1')
 - 4) *DLC (Data Length Code)* คือบิตข้อมูลขนาด 4 บิต สำหรับแสดงขนาดข้อมูลที่อยู่ในฟิลด์ข้อมูล
 - 5) *Data Field* คือฟิลด์สำหรับบรรจุข้อมูลที่ต้องการส่งมีขนาด 0 ถึง 8 ไบต์ โดยเฟรมร้องขอข้อมูลมีฟิลด์ข้อมูลเป็น 0 ไบต์
 - 6) *CRC Field (Cyclic Redundancy Check Field)* คือฟิลด์ตรวจสอบความถูกต้องของเฟรมด้วยวิธีคำนวณค่า CRC จากบิตเริ่มต้นของเฟรมจนถึงบิตสุดท้ายของฟิลด์ข้อมูล
 - 7) *ACK Field (Acknowledge Field)* คือฟิลด์ตอบสนองของเฟรม เป็นส่วนที่โหนดซึ่งได้รับเฟรมข้อมูลใช้ตอบกลับไปยังโหนดผู้ส่งเพื่อยืนยันความถูกต้องของข้อมูลที่ได้รับ
 - 8) *EOF (End of Frame)* คือบิตสิ้นสุดเฟรม ประกอบด้วยบิตลอจิก '1' จำนวน 7 บิต
- นอกจากนี้เพื่อให้เวลากับโหนดผู้รับสำหรับใช้ประมวลผลหรือจัดการกับข้อมูลที่ได้รับให้เรียบร้อย ดังนั้นมาตรฐานจึงกำหนดให้ส่งค่าลอจิก '1' ต่อท้ายบิตสิ้นสุดเฟรมอย่างน้อย 3 บิตก่อนที่จะสามารถส่งเฟรมข้อมูลใหม่ได้

ในการสื่อสารข้อมูลผ่านบัส CAN รหัสประจำตัวข้อความถูกใช้เพื่อระบุชนิดและระดับความสำคัญของข้อความ (Message-based communication) ทุกโหนดที่เชื่อมต่อกับบัสสามารถรับ

ข้อความที่กำลังสื่อสารอยู่บนบัสได้ หากรหัสประจำตัวของข้อความบนบัสตรงกับที่โหนดนั้นๆ ต้องการ ซึ่งเรียกลักษณะการส่งข้อมูลแบบนี้ว่า “การส่งข้อมูลแบบbroadcast (Broadcast Communication)” มีผลทำให้ในการส่งข้อความหนึ่งครั้ง โหนดที่เหลือสามารถรับข้อความนั้นได้พร้อมกัน นอกจากนี้บัส CAN ยังรองรับการเข้าใช้บัสจากหลายโหนดได้พร้อมกัน (Multi-master Communication) ในกรณีที่มีโหนดมากกว่าหนึ่งโหนดต้องการส่งข้อความในเวลาเดียวกัน กลไกเปรียบเทียบความสำคัญของข้อความถูกเรียกใช้เพื่อให้สิทธิในการใช้บัสแก่โหนดที่มีความสำคัญสูงสุดขณะนั้น โหนดที่เหลือต้องหยุดทำการส่งข้อความเพื่อเริ่มส่งข้อความหรือเปรียบเทียบความสำคัญอีกครั้งหลังจากบัสว่าง เรียกกลไกดังกล่าวว่า “Arbitration on Message Priority” ซึ่งข้อความที่มีความสำคัญสูงสุดคือข้อความที่มีรหัสประจำตัวค่าน้อยๆ (มีลอจิก ‘0’ มาก) และกลไกนี้เป็นการเปรียบเทียบในระดับบิต

เนื่องจากการสื่อสารข้อมูลผ่านบัส CAN ใช้วิธีการส่งสัญญาณแบบ NRZ (Non-Return-to-Zero) และเป็นการส่งข้อมูลแบบอะซิงโครนัส (Asynchronous) ซึ่งต้องอาศัยการเข้าจังหวะสัญญาณนาฬิกาจากสัญญาณข้อมูลที่ส่งมา ในกรณีที่สื่อสารสัญญาณข้อมูลที่มีค่าเดียวกันติดๆ กันหลายบิต อาจทำให้การเข้าจังหวะกับสัญญาณข้อมูลนั้นเกิดความผิดพลาด ดังนั้นนอกจากการเข้ารหัสข้อความด้วยรหัส CRC จึงได้กำหนดให้มีการเติมสตฟฟ์บิต (Stuff Bit) เพื่อป้องกันปัญหาดังกล่าว โดยวิธีการเติมสตฟฟ์บิตคือ ถ้ามีการส่งบิตข้อมูลที่มีค่าเดียวกันติดกัน 5 บิตต้องคั่นด้วยบิตที่มีค่าตรงข้ามกับ 5 บิตนั้น 1 บิต ซึ่งมีเพียงบิตข้อมูลตั้งแต่บิตเริ่มต้นเฟรมจนถึงบิตสุดท้ายของฟิลด์ CRC เท่านั้นที่ถูกเติมสตฟฟ์บิต

2.2.3 ความสามารถทั่วไปของโปรโตคอลชั้นสูงกว่าบัส CAN

ด้วยความสามารถและสมบัติพื้นฐานของบัส CAN เมื่อถูกนำไปประยุกต์ใช้กับระบบงานที่มีขนาดใหญ่ขึ้นหรือระบบงานที่มีอุปกรณ์ต่างชนิดและต่างผู้ผลิตต่อร่วมกัน เพื่อให้ระบบดังกล่าวสามารถทำงานร่วมกันได้จึงมีการพัฒนาชั้นสื่อสารหรือโปรโตคอลชั้นสูงกว่าเพื่อควบคุมฟังก์ชันพื้นฐานของบัส CAN ขึ้น ซึ่งสรุปความสามารถทั่วไปได้ดังนี้

- 1) การจัดกลุ่มหมายเลขประจำตัวข้อความ รูปแบบและความหมายของเฟรมสื่อสารข้อมูล
- 2) ส่วนควบคุมทิศทางของการสื่อสารของข้อมูล
- 3) ส่วนจัดลำดับการสื่อสารข้อมูลที่มีขนาดมากกว่า 8 ไบต์
- 4) การนิยามหมายเลขประจำตัวของอุปกรณ์
- 5) ส่วนจัดการการสื่อสารข้อมูลระหว่างอุปกรณ์ในเครือข่าย เช่น การเริ่มต้นหรือหยุดการสื่อสารและจัดการกับข้อผิดพลาดของเฟรมสื่อสารข้อมูล เป็นต้น

ในปัจจุบันมีโพรโทคอลชั้นสูงกว่าบัส CAN ที่ถูกกำหนดมาตรฐานไว้หลายรูปแบบ โดยพัฒนาความสามารถให้เหมาะสมกับลักษณะงานที่นำไปใช้ เช่น โพรโทคอลสำหรับระบบควบคุมมอเตอร์และเซนเซอร์ที่เกี่ยวข้องกับการเคลื่อนไหว (Motion Control System) โพรโทคอลสำหรับระบบวินิจฉัยรถยนต์ (Diagnostic System) เป็นต้น

2.3 การวินิจฉัยรถยนต์โดยใช้บัส CAN

ข้อกำหนดในการวินิจฉัยรถยนต์สามารถแบ่งออกเป็นสองประเภทคือ ข้อกำหนดภายใต้ตรากฎหมายเกี่ยวกับการวินิจฉัย (Legislated on-board diagnostics) และข้อกำหนดการวินิจฉัยที่เพิ่มเติมความสามารถของแต่ละผู้ผลิตรถยนต์ (Vehicle manufacturer enhanced diagnostics) ซึ่งข้อกำหนดประเภทหลังเป็นการเพิ่มเติมความสามารถในการวินิจฉัยรถยนต์ของแต่ละผู้ผลิตรถยนต์ โดยครอบคลุมข้อกำหนดประเภทแรก สามารถแสดงมาตรฐานการสื่อสารที่ใช้อธิบายการทำงานของข้อกำหนดทั้งสองประเภทดังตาราง 2.4 เนื้อหาส่วนนี้นำเสนอเฉพาะสาระสำคัญตามข้อกำหนดภายใต้ตรากฎหมายเกี่ยวกับการวินิจฉัยเท่านั้น

ตาราง 2.4 มาตรฐานการสื่อสารภายใต้ข้อกำหนดการวินิจฉัย [10]

OSI layers	Vehicle manufacturer enhanced diagnostics	Legislated on-board diagnostics (OBD)
Diagnostic application	User defined	ISO 15031-5
Application layer	ISO 15765-3	ISO 15031-5
Presentation layer	N/A	N/A
Session layer	N/A	N/A
Transport layer	N/A	N/A
Network layer	ISO 15765-2	ISO 15765-4
Data link layer	ISO 11898-1	ISO 15765-4
Physical layer	User defined	ISO 15765-4

2.3.1 มาตรฐาน ISO 15765-4

มาตรฐาน ISO 15765-4 คือมาตรฐานที่ครอบคลุมการทำงานของมาตรฐาน ISO 15765-2 และ ISO 11898-1 ซึ่งสามารถสรุปสาระสำคัญได้ดังนี้

1) ข้อกำหนดเพิ่มเติมในชั้นกายภาพ มาตรฐาน ISO 15765-4 ได้กำหนดอัตราบอดของ บัส CAN เพื่อสื่อสารข้อมูลระหว่างระบบ OBD กับสแกนทูลไว้ 2 อัตราบอดคือ 250 กิโลบิตต่อวินาทีและ 500 กิโลบิตต่อวินาที

2) ข้อกำหนดเพิ่มเติมในชั้นเชื่อมโยงข้อมูล มาตรฐาน ISO 15765-4 ได้กำหนดให้ฟิลด์ข้อมูลของเฟรมข้อมูลมีขนาด 8 ไบต์ตลอด (DLC มีค่าเท่ากับ 0x08) แม้ว่าจำนวนข้อมูลที่ใช้จริงต่อหนึ่งเฟรมมีไม่ถึง 8 ไบต์ ส่วนไบต์ข้อมูลที่เหลือมีค่าเป็นอะไรก็ได้แต่ส่วนมากนิยมใช้ค่าเท่ากับ 0x55 หรือ 0xAA (มีค่าลอจิก '0' และ '1' สลับกัน) เพื่อหลีกเลี่ยงกระบวนการเติมสต๊าฟฟ์บิตของ โปรโตคอล CAN ที่อาจส่งผลให้มีจำนวนบิตและใช้เวลาในการสื่อสารข้อมูลต่อหนึ่งครั้งเพิ่มขึ้น

3) ข้อกำหนดในชั้นเครือข่าย (Network layer) มาตรฐาน ISO 15765-4 (หรือมาตรฐาน ISO 15765-2) ได้กำหนดรูปแบบเฟรมข้อมูล CAN หมายเลขประจำตัวของอีซียู ทิศทางการเคลื่อนตัวของข้อมูลและขั้นตอนการสื่อสารระหว่างโหนดร้องขอและโหนดให้บริการ ซึ่งโดยปกติสแกนทูลคือโหนดร้องขอและอีซียูคือโหนดให้บริการ

3.1) หมายเลขประจำตัวของอีซียู สำหรับข้อกำหนดภายใต้ตรากฎหมายเกี่ยวกับการวินิจฉัยได้กำหนดให้มีจำนวนอีซียูในระบบ OBD ไม่เกิน 8 ตัวและกำหนดให้สามารถติดต่อกับอีซียูได้โดยตรง (Physical request) หรือติดต่อทางอ้อม (Functional request) นอกจากนี้ได้กำหนดหมายเลขประจำตัวเพื่อใช้สื่อสารระหว่างอีซียูและสแกนทูลไว้แบ่งออกเป็น 2 รูปแบบตามขนาดของหมายเลขประจำตัว CAN แสดงดังตาราง 2.5 และ 2.6 ตามลำดับ

ตาราง 2.5 หมายเลขประจำตัวอีซียูขนาด 11 บิต

หมายเลขประจำตัว CAN (เลขฐานสิบหก)	รายละเอียด
7DF	หมายเลขประจำตัวอีซียูสำหรับติดต่อทางอ้อม
7E0 – 7E7	หมายเลขประจำตัวอีซียูสำหรับติดต่อโดยตรงของอีซียู #1-8
7E8 – 7EF	หมายเลขประจำตัวอีซียูที่ตอบกลับจากอีซียู #1-8

ตาราง 2.6 หมายเลขประจำตัวอีซียูขนาด 29 บิต

หมายเลขประจำตัว CAN (ฐานสิบหก)	รายละเอียด
18DB33F1	หมายเลขประจำตัวอีซียูสำหรับติดต่อทางอ้อม
18DAxxF1	หมายเลขประจำตัวอีซียูสำหรับติดต่อโดยตรงของอีซียู #xx
18DAF1xx	หมายเลขประจำตัวอีซียูที่ตอบกลับจากอีซียู #xx

F1 คือหมายเลขประจำตัวของสแกนทูล

xx คือหมายเลขประจำตัวของอีซียู [10]

3.2) การจัดรูปแบบเฟรมข้อมูล ไบต์แรกส่วนฟิลด์ข้อมูลของเฟรมข้อมูล CAN ถูกใช้เป็น PCI ไบต์ (Protocol Control Information) และถูกแบ่งออกเป็น 2 ส่วนประกอบด้วยรหัส 4 บิตบนของ PCI ไบต์เพื่อระบุชนิดของเฟรมข้อมูล CAN และรหัส 4 บิตล่างเพื่อระบุพารามิเตอร์ของแต่ละเฟรมข้อมูล CAN โดยจำแนกออกเป็น 4 ชนิดและมีรายละเอียดดังตาราง 2.7

ตาราง 2.7 รูปแบบและพารามิเตอร์ของแต่ละชนิดเฟรมข้อมูล CAN [10]

PCI Frame Name	PCI Byte structure (Byte 1)					Data Bytes (2 – 8)		
	Message type encoding bits				Message specific encoding bits			
	7	6	5	4	3 - 0	2	3	4 – 8
SF	0	0	0	0	Data Length	Payload data (2 – 8)		
FF	0	0	0	1	Extended Data Length	Data Length	Payload data (3 – 8)	
CF	0	0	1	0	Sequence Number	Payload data (2 – 8)		
FC	0	0	1	1	Flow Status	BS	STmin	-

ก) *SF* (Single Frame) คือเฟรมสำหรับการรับ-ส่งข้อมูลที่มีขนาด 0 ถึง 7 ไบต์ โดยใช้พารามิเตอร์ Data Length (DL) เพื่อระบุจำนวนข้อมูลในฟิลด์ข้อมูล ทั้งนี้ SF สามารถถูกใช้เป็นเฟรมสำหรับร้องขอข้อมูลหรือใช้ตอบกลับข้อมูล

ข) *FF* (First Frame) คือเฟรมเริ่มต้นในการรับ-ส่งข้อมูลที่มีมากกว่า 7 ไบต์โดยใช้ส่วน 4 บิตล่างของ PCI ไบต์และไบต์ข้อมูลถัดไปเพื่อระบุจำนวนข้อมูลทั้งหมดที่ถูกส่งหรือรับอย่างต่อเนื่อง (Extended Data Length and Data Length, DL 12 บิต) ซึ่ง FF ถูกใช้เป็นเฟรมตอบกลับเมื่อข้อมูลที่ร้องขอมีขนาดมากกว่า 7 ไบต์

ค) *CF* (Consecutive Frame) คือเฟรมต่อเนื่องในการรับ-ส่งข้อมูลที่มีมากกว่า 7 ไบต์ โดยใช้พารามิเตอร์ Sequence Number (SN) เพื่อระบุลำดับของเฟรมและไบต์ที่เหลือคือข้อมูลที่ถูกรับหรือส่งต่อเนื่องจากไบต์ข้อมูลของ FF

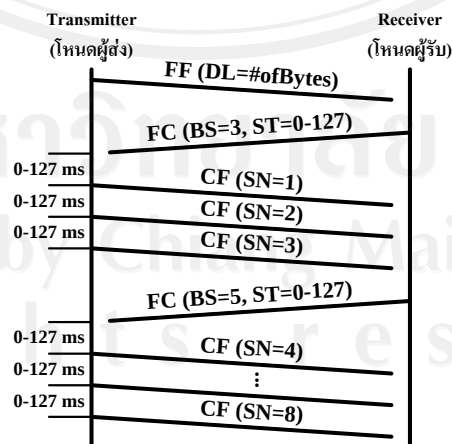
ง) *FC* (Flow Control) คือเฟรมกำหนดขนาดของส่วนข้อมูลและอัตราการตอบสนองในการรับ-ส่งข้อมูลที่มีมากกว่า 7 ไบต์ โดยส่วน 4 บิตล่างของ PCI ไบต์คือพารามิเตอร์ Flow Status สำหรับแสดงสถานะของการเคลื่อนข้อมูลมีรายละเอียดดังตาราง 2.8

ตาราง 2.8 รายละเอียดของ Flow Status

State	Flow Status nibble	Description
Continue to Send (CTS)	0	ร้องขอให้ส่ง CF โดยมีจำนวนของเฟรม (BS) และอัตรา การส่ง (STmin) ตามที่ระบุ
Wait (WT)	1	ร้องขอให้หยุดส่ง CF และรอจนกว่าตรวจพบ FC ถัดไป
Overflow (OVFLW)	2	ร้องขอให้ยกเลิกการส่งข้อมูลเนื่องจากเกิดการทับซ้อน ของข้อมูล

ส่วนไบต์ที่สองคือพารามิเตอร์ Block Size (BS) ใช้กำหนดจำนวนของ CF ต่อช่วงการส่ง FC หนึ่งครั้ง เช่นเมื่อใช้ FC ตอบกลับ FF ด้วยสถานะ CTS และพารามิเตอร์ BS เท่ากับ 10 จากนั้นจะ
ได้รับ CF จำนวน 10 เฟรมตอบกลับมา (ครั้งละ 10 เฟรมจากเฟรมข้อมูลทั้งหมด) และหยุดการ
สื่อสารข้อมูลจนกว่าจะได้รับ FC อีกครั้ง (สำหรับ 10 เฟรมถัดไป) หากพารามิเตอร์ BS มีค่าเท่ากับ
0 คืออนุญาตให้ส่งเฟรมข้อมูลทั้งหมดอย่างต่อเนื่องโดยไม่ต้องรอ FC

ไบต์ที่สามของ FC คือพารามิเตอร์ Minimum Separation Time (STmin, ST) สำหรับใช้
กำหนดระยะห่างระหว่าง CF แต่ละเฟรม โดย 0x00 – 0x7F มีค่าเท่ากับ 0 – 127 มิลลิวินาทีและ
0xF1 – F9 มีค่าเท่ากับ 100 – 900 ไมโครวินาที รูป 2.7 แสดงตัวอย่างลำดับการสื่อสารข้อมูลที่มี
ขนาดมากกว่า 7 ไบต์ด้วยเฟรมข้อมูลชนิดต่างๆ



รูป 2.7 แผนผังลำดับการสื่อสารข้อมูลที่มีขนาดมากกว่า 7 ไบต์

2.3.2 การประยุกต์ใช้ระบบ OBD ด้วยบัส CAN

มาตรฐาน ISO 15031-5 ประกอบด้วยข้อกำหนดในการวินิจฉัยดังที่นำเสนอในหัวข้อที่ 2.1 ซึ่งอธิบายถึงพารามิเตอร์ต่างๆ ขั้นตอนการทดสอบระบบย่อยในรถยนต์ และกระบวนการทดสอบ หลังจากการวินิจฉัยที่บังคับด้วยตรากฎหมาย เป็นต้น ในหัวข้อนี้นำเสนอตัวอย่างการประยุกต์ใช้ข้อกำหนดตาม SAE J1979 (หัวข้อ 2.1.1) กับการจัดรูปแบบเฟรมข้อมูลตามมาตรฐาน ISO 15765-4 (หัวข้อ 2.3.1)

ตาราง 2.9 แสดงตัวอย่างการวินิจฉัยรถยนต์ด้วยโหมดการทำงาน 03 เป็นการร้องขอข้อมูลบันทึก รหัส DTCs ที่อียูตรวจพบ ซึ่งการทำงานด้วยโหมดนี้ไม่จำเป็นต้องระบุพารามิเตอร์ที่เกี่ยวข้อง จากตัวอย่างนี้ได้ผลตอบกลับจากอียูหมายเลข 0x11 ตัวเลขข้อมูลจาก PCI ไบต์คือจำนวนข้อมูลที่ได้รับไม่นับรวม PCI ไบต์ ตัวเลข 0x43 ระบุว่าผลตอบกลับนี้เป็นผลจากการร้องขอด้วยโหมดการทำงาน 03 ซึ่งบิตที่ 6 จะมีค่าลอจิกเป็น '1' ตัวเลข 0x01 ถัดไปแสดงจำนวนของรหัส DTCs ที่ตรวจพบและตัวเลขคู่ถัดไป 0x0102 คือรหัส DTCs เมื่อนำไปตีความตามมาตรฐาน SAE J2012 ได้เป็น หมายเลขรหัส P0102 – Mass or Volume Air Flow Circuit Low Input

ตาราง 2.9 ตัวอย่างการวินิจฉัยรถยนต์ด้วยโหมดการทำงาน 03 (ไม่มีพารามิเตอร์)

และตรวจพบรหัส DTC คือ P0102

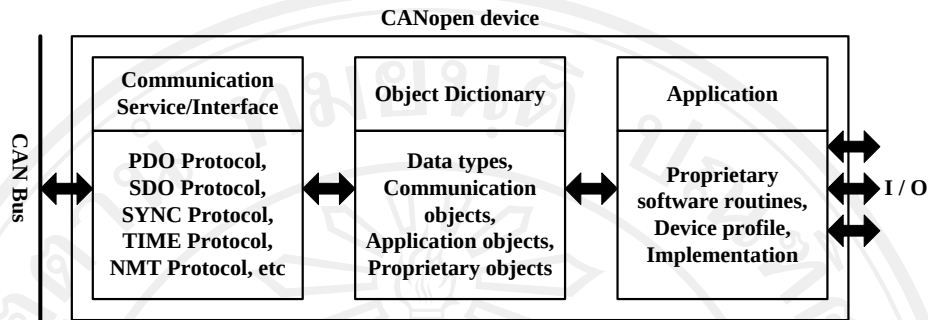
ทิศทาง	หมายเลขประจำตัว CAN	ไบต์ข้อมูล							
		PCI	2	3	4	5	6	7	8
ส่งให้อียู	18DB33F1	01	03	AA	AA	AA	AA	AA	AA
ตอบกลับจากอียู	18DAF111	04	43	01	01	02	AA	AA	AA

**ตัวเลขข้อมูลทุกตัวเป็นฐานสิบหก

2.4 โพรโทคอล CANopen

โพรโทคอล CANopen [9] คือข้อกำหนดการสื่อสารข้อมูลในชั้นประยุกต์ของบัส CAN (CAN Application Layer) ที่ถูกพัฒนาด้วยความร่วมมือของกลุ่มอุตสาหกรรมที่เกี่ยวข้องกับบัส CAN (CAN in Automation association, CiA) โดยการพัฒนาชั้นสื่อสารข้อมูลที่ประยุกต์ใช้ความสามารถของบัส CAN เพื่อให้รองรับการทำงานร่วมกันของอุปกรณ์ที่ต่างชนิดและต่างผู้ผลิต ซึ่งจุดเด่นของโพรโทคอล CANopen คือเป็นระบบเปิดสามารถเพิ่มหรือลดสมบัติจากที่กำหนดเป็นมาตรฐานได้และไม่มีค่าใช้จ่ายในการประยุกต์ใช้งาน หลักการของโพรโทคอล CANopen ที่ทำให้อุปกรณ์ต่างๆ ทำงานร่วมกันได้คือ การพัฒนาชุดบริการสื่อสารข้อมูล (Communication Service) และพจนานุกรมวัตถุ (Object Dictionary) รูป 2.8 แสดงโครงสร้างการทำงานของอุปกรณ์

CANopen ซึ่งมีพจนานุกรมวัตถุเป็นสื่อกลางประสานการทำงานของโปรแกรมประยุกต์และส่วนสื่อสารข้อมูลด้วยบัส CAN



รูป 2.8 โมเดลการทำงานของอุปกรณ์ CANopen

2.4.1 พจนานุกรมวัตถุ (Object Dictionary)

พจนานุกรมวัตถุหรือตารางข้อมูลของแต่ละอุปกรณ์ ประกอบด้วยข้อกำหนด 2 ส่วน ส่วนแรกคือข้อกำหนดเกี่ยวกับรายละเอียดของอุปกรณ์ (Device Profile) เช่น ชนิดและประเภทของอุปกรณ์ (แอนะล็อกหรือดิจิทัล มอเตอร์ เอนโค้ดเดอร์) และฟังก์ชันการทำงาน เป็นต้น ส่วนที่สองคือข้อกำหนดเกี่ยวกับการสื่อสารของตัวอุปกรณ์ (Communication Profile) ซึ่งระบุวิธีและรูปแบบที่ใช้สื่อสารกับอุปกรณ์แต่ละตัว ทั้งนี้กลุ่ม CiA ได้พัฒนาต้นแบบของข้อกำหนดเกี่ยวกับรายละเอียดของอุปกรณ์เรียกว่า “CiA 4xx” และข้อกำหนดเกี่ยวกับการสื่อสารของอุปกรณ์เรียกว่า “CiA/DS 301 และ CiA/DS 302” เพื่อเป็นแนวทางในการพัฒนาระบบงานต่อไป [9]

ข้อกำหนดทั้งสองส่วนถูกเก็บเป็นฐานข้อมูลในพจนานุกรมวัตถุของตัวอุปกรณ์ตามตำแหน่งครรชนิต่างๆ ดังตาราง 2.10 การเข้าถึงข้อมูลทั้งการอ่านหรือเขียนข้อมูลสามารถทำได้โดยใช้ชุดบริการสื่อสารที่โพรโทคอลเตรียมไว้ร่วมกับตัวครรชนิตัวหลักขนาด 16 บิตและตัวครรชนิรองขนาด 8 บิต เพื่อชี้ตำแหน่งข้อมูลที่ต้องการในพจนานุกรมวัตถุ

ตาราง 2.10 ตำแหน่งข้อมูลวัตถุตามครรชนิตัวในพจนานุกรมวัตถุ

ครรชนิตัวหลัก (Index)	วัตถุ (Object)
0000	not used
0001-001F	Static Data Types (standard data types, e.g. Boolean, Integer16)
0020-003F	Complex Data Types (predefined structures composed of standard data types, e.g. PDOCommPar, SDOParameeter)

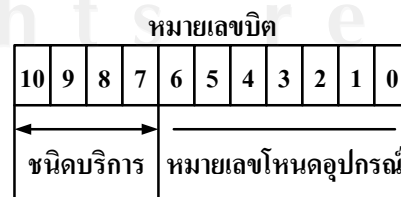
ตาราง 2.10 ตำแหน่งข้อมูลวัตถุตามดัชนีในพจนานุกรมวัตถุ (ต่อ)

ดัชนีหลัก (Index)	วัตถุ (Object)
0040-005F	Manufacturer Specific Complex Data Types
0060-007F	Device Profile Specific Static Data Types
0080-009F	Device Profile Specific Complex Data Types
00A0-0FFF	Reserved
1000-1FFF	Communication Profile Area (e.g. Device Type, Error Register, Number of PDOs supported)
2000-5FFF	Manufacturer Specific Profile Area
6000-9FFF	Standardized Device Profile Area (e.g. “DSP-401 Device Profile for I/O Modules”)
A000-FFFF	Reserved

**ตัวเลขข้อมูลทุกตัวเป็นฐานสิบหก

2.4.2 ขาดบริการสื่อสารข้อมูล (Communication Service)

1) การจัดรูปแบบเฟรมข้อมูล ขาดบริการสื่อสารข้อมูลของโพรโทคอล CANopen ถูกพัฒนาด้วยการใช้หมายเลขข้อความ CAN ขนาด 11 บิต ผสานกับสมบัติสำคัญของ CAN คือข้อความที่มีค่าหมายเลขต่ำสุดจะมีความสำคัญสูงสุดและได้รับอนุญาตให้สื่อสารข้อมูลบนบัสก่อน [8] โพรโทคอล CANopen แบ่งกลุ่มบิตของหมายเลขข้อความดังแสดงในรูป 2.9 และแบ่งชนิดบริการสื่อสารออกเป็น 16 ชนิดดังตาราง 2.11 เรียงตามลำดับความสำคัญ นอกจากนี้กำหนดให้โหนดอุปกรณ์บนเครือข่ายมีหมายเลขโหนดที่ไม่ซ้ำกันได้ตั้งแต่ 1 ถึง 127 หมายเลข (หมายเลข 0 ถูกจองไว้สำหรับโหนดอุปกรณ์ที่มีศักดิ์เป็นโหนดมาสเตอร์) เพื่อสนับสนุนการสื่อสารแบบมาสเตอร์-สลาฟบนบัส CAN



รูป 2.9 การจัดรูปแบบเฟรมข้อมูล CAN โดยโพรโทคอล CANopen

ตาราง 2.11 ชนิดบริการสื่อสารข้อมูลของโปรโตคอล CANopen

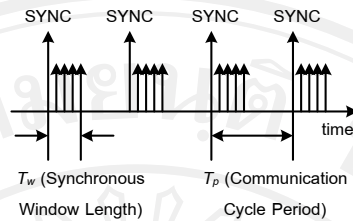
เลข	ชื่อบริการ	รายละเอียด
0	Network management protocol, NMT	สำหรับควบคุมการสื่อสารของเครือข่ายอุปกรณ์
1	Synchronous protocol, SYNC	สำหรับการซิงโครไนส์ให้ทุกโหนดอุปกรณ์ทำงานพร้อมกัน
2	Time of system protocol, TIME	สำหรับเป็นฐานเวลาของเครือข่ายอุปกรณ์
3-10	Process data object protocol, PDO	สำหรับสื่อสารข้อมูลตามรูปแบบเดิมของ CAN
11, 12	Service data object protocol, SDO	สำหรับสื่อสารข้อมูลที่มีปริมาณมากและใช้อ่านหรือเขียนข้อมูลในพจนานุกรมวัตถุ
14	Emergency protocol	สำหรับแสดงความผิดพลาดที่เกิดจากการสื่อสารข้อมูล

**13, 15 สำรองไว้

2) บริการ PDO (Process Data Object) บริการหรือข้อความ PDO ใช้ในการสื่อสารข้อมูลปริมาณสั้นๆ แต่ต้องการสื่อสารบ่อยครั้ง (ขนาด 0 ถึง 8 ไบต์) มีสมบัติเหมือนเฟรมข้อมูล CAN ดั้งเดิม ข้อความ PDO แบ่งเป็นสองประเภทตามทิศทางการสื่อสารข้อมูลอ้างอิงจากตัวโหนดอุปกรณ์เอง ประเภทที่หนึ่งคือ TPDO (Transmit PDO) เป็นข้อความสำหรับส่งข้อมูลออกจากโหนดและประเภทที่สองคือ RPDO (Receive PDO) เป็นข้อความที่โหนดได้รับ บริการ PDO มีโมเดลการสื่อสารเป็นแบบผู้ผลิต-ผู้บริโภค (producer/consumer) ผู้ผลิตคือโหนดที่สร้างข้อความ TPDO และผู้บริโภคคือโหนดที่รับข้อความ RPDO ผู้ผลิตสามารถเป็นผู้บริโภคได้ในเวลาเดียวกัน และหนึ่งผู้ผลิตสามารถสร้างข้อความ TPDO ให้แก่ผู้บริโภคมากกว่า 2 โหนดพร้อมกัน วิธีการดังกล่าวทำให้ข้อมูลถูกกระจายไปยังโหนดต่างๆ ในเวลาเดียวกัน

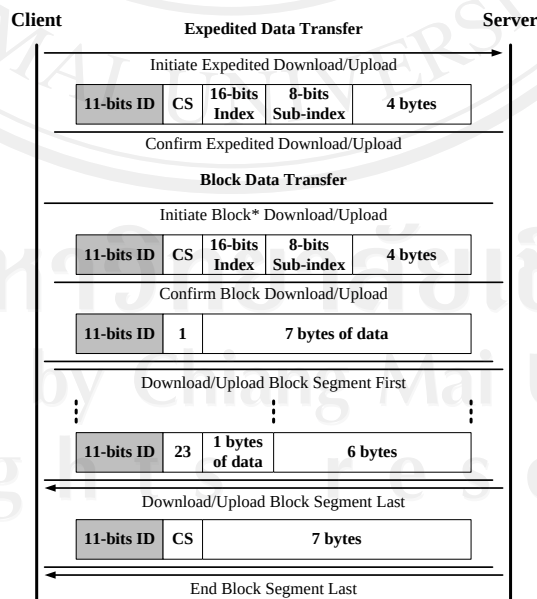
โหนดจะส่งข้อความ TPDO เมื่อถูกกระตุ้นด้วยเหตุการณ์ที่กำหนดไว้หรือได้รับข้อความ SYNC จากโหนดมาสเตอร์ หลังจากที่ได้รับข้อความ SYNC โหนดที่เป็นผู้ผลิตจะแย่งกันส่งข้อความ TPDO ออกมาภายในเวลาที่กำหนด T_w ดังแสดงในรูป 2.10 การใช้ข้อความ SYNC ทำให้ทุกโหนดมีจังหวะทำงานเดียวกันและสามารถกำหนดให้ส่งข้อความ TPDO ทุกๆ n ครั้งของการเกิดข้อความ SYNC ได้ เมื่อ n คืออัตราที่ต้องการข้อความ TPDO ซึ่งพารามิเตอร์ดังกล่าวรวมถึงหมายเลขของข้อความ TPDO และ RPDO ตำแหน่งที่ใช้เก็บข้อมูล คาบของข้อความ SYNC (T_p)

และเวลาในการตอบสนองข้อความ SYNC (T_w) ถูกกำหนดในพจนานุกรมวัตถุไว้ล่วงหน้าด้วย กระบวนการที่เรียกว่า PDO mapping



รูป 2.10 การกำหนดจังหวะทำงานด้วยข้อความ SYNC

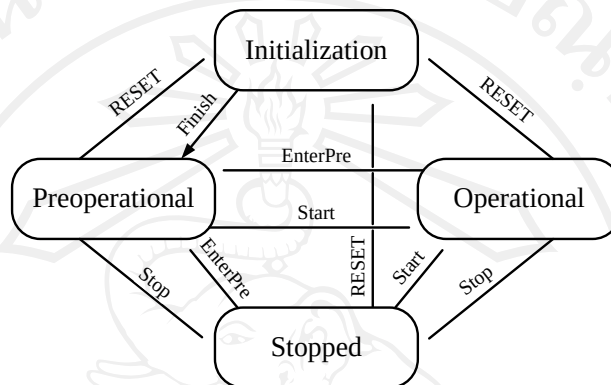
3) บริการ SDO (Service Data Object) บริการหรือข้อความ SDO ใช้สำหรับอ่านและเขียนข้อมูลในพจนานุกรมวัตถุ มีโมเดลการสื่อสารเป็นแบบ client/server โหนด client คือโหนดที่ร้องขอเข้าถึงข้อมูลในพจนานุกรมวัตถุและโหนด server คือโหนดที่พจนานุกรมวัตถุถูกใช้งาน การอ่านหรือเขียนข้อมูลเริ่มเมื่อโหนด server ตอบรับการร้องขอจากโหนด client เท่านั้น ข้อความ SDO มีสองรูปแบบคือ แบบ expedited transfer และแบบ segmented transfer แบบ expedited transfer ใช้สองขั้นตอน เพื่อสื่อสารข้อมูลได้สูงสุด 4 ไบต์ และแบบ segmented transfer ใช้หลายๆ ขั้นตอนการสื่อสารเพื่อสื่อสารข้อมูลที่มากกว่า 4 ไบต์ โดยข้อมูลที่มากกว่า 4 ไบต์ถูกแบ่งเป็นหลายๆ ท่อนเพื่อบรรจุในข้อความ SDO และใช้ไบต์แรกของส่วนข้อมูลซึ่งเรียกว่า command specifier (cs) กำกับลำดับของข้อความ ก่อนถูกสื่อสารด้วยหลายๆ ขั้นตอนสื่อสารดังแสดงในรูป 2.11



*1 Block Data = 160 Bytes of Data , = not used / reserved

รูป 2.11 การสื่อสารด้วยข้อความ SDO

4) บริการ NMT (Network Management) บริการหรือข้อความ NMT ใช้ควบคุมการทำงานของโหนดอุปกรณ์ในเครือข่าย รวมถึงการตั้งค่าพารามิเตอร์เริ่มต้นในการทำงานและการเปลี่ยนสถานะของโหนดอุปกรณ์เป็นต้น รูป 2.12 แสดงผังสถานะโหนดอุปกรณ์สถาฟประกอบด้วย 4 สถานะคือ Initialization, Preoperational, Operational และ Stopped ซึ่งโหนดมาสเตอร์ใช้กลุ่มข้อความ NMT เพื่อควบคุมและตรวจสอบสถานะของโหนดสเลฟ



รูป 2.12 ผังสถานะโหนดอุปกรณ์สถาฟ

เมื่อเริ่มจ่ายไฟให้ระบบ โหนดสเลฟเข้าสู่สถานะ Initialization เพื่อเตรียมความพร้อมและตั้งค่าเริ่มต้นของอุปกรณ์บริหารที่โหนดต่อฟ่วงอยู่ จากนั้นเข้าสู่สถานะ Preoperational โดยอัตโนมัติ ในสถานะนี้โหนดมาสเตอร์ใช้ข้อความ SDO เพื่อกำหนดการทำงานส่วนต่างๆ ของโหนดสเลฟ หลังจากนั้นโหนดมาสเตอร์ส่งข้อความ NMT(Start) เพื่อสั่งให้โหนดสเลฟเข้าสู่สถานะ Operational ในสถานะนี้โหนดมาสเตอร์สามารถใช้ข้อความ PDO และข้อความ SYNC เพื่อสื่อสารข้อมูลระหว่างโหนดตามฟังก์ชันงานที่กำหนดไว้ นอกจากนี้เมื่อต้องการให้โหนดสเลฟหยุดทำงาน โหนดมาสเตอร์สามารถส่งข้อความ NMT(Stop) เพื่อสั่งให้โหนดสเลฟเข้าสู่สถานะ Stopped

2.5 เทคโนโลยีจีพีอาร์เอส (General Packet Radio Service, GPRS)

จีพีอาร์เอส (General Packet Radio Service, GPRS) [2] คือบริการสื่อสารข้อมูลผ่านระบบโทรศัพท์เคลื่อนที่จีเอสเอ็ม (Global System for Mobile communication, GSM) ซึ่งเป็นบริการเสริมที่พัฒนาเพื่อทดแทนบริการสื่อสารข้อความสั้น (Short Message Service, SMS) และบริการสื่อสารข้อมูลผ่านระบบการสวิตซ์วงจร (Circuit Switched Data, CSD) ข้อมูลที่ต้องการส่งด้วยบริการจีพีอาร์เอสถูกแบ่งออกเป็นส่วนย่อยๆ ในรูปแบบของแพ็คเกจข้อมูล (packet) และถูกส่งไปพร้อมๆ กันในเครือข่ายจีพีอาร์เอส โดยมีการรวมกันของแพ็คเกจย่อยๆ ดังกล่าวอีกครั้งที่ทางด้านรับ ลักษณะการส่งข้อมูลดังกล่าวเรียกว่า การสลับกลุ่มข้อมูล (Packet switching) ซึ่งเป็นระบบเดียวกันกับการส่งข้อมูลในระบบอินเทอร์เน็ต

ด้วยรูปแบบการส่งข้อมูลที่เป็นแพ็คเกจดังกล่าว ช่องสัญญาณในบริการจีพีอาร์เอสถูกเรียกใช้งานเมื่อผู้ใช้ทำการรับ-ส่งข้อมูลเท่านั้น ผู้ใช้รายอื่นๆ สามารถแบ่งใช้ช่องสัญญาณนี้ได้ซึ่งต่างจากบริการซีเอสดีทีที่ต้องจองช่องสัญญาณไว้ตลอดเวลาที่มีการสื่อสารและคืนช่องสัญญาณให้เมื่อจบการสื่อสารนั้นๆ จากสมบัติดังกล่าวส่งผลต่อการคิดค่าบริการด้วยปริมาณข้อมูลที่รับ-ส่งทำให้ประหยัดค่าใช้จ่ายและยังสามารถรองรับผู้ใช้งานจำนวนมากขึ้นเมื่อเทียบกับบริการในรูปแบบเดิม นอกจากนี้บริการสื่อสารจีพีอาร์เอสทำให้ผู้ใช้งานสามารถเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตเพื่อใช้งานด้านต่างๆ เช่น การค้นหาข้อมูลในอินเทอร์เน็ตและการรับ-ส่งอีเมลล์ เป็นต้น

2.5.1 สมบัติของบริการจีพีอาร์เอส

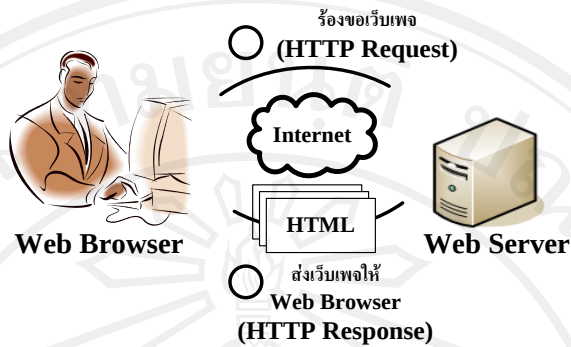
- 1) การสื่อสารข้อมูลที่มีความสามารถในการรับ และส่งผ่านเครือข่ายอินเทอร์เน็ตได้สูงถึง 40 กิโลบิตต่อวินาที (kbps) ซึ่งทำให้สามารถรองรับการสื่อสาร ข้อมูลในรูปแบบมัลติมีเดีย เช่น วิดีโอไฟล์ หรือภาพเคลื่อนไหว เสียงและวีดิโอ เป็นต้น พร้อมทั้ง สามารถ เชื่อมต่อกับเครือข่ายอินเทอร์เน็ตได้รวดเร็วและมีประสิทธิภาพมากกว่าเดิม
- 2) “Always on” คือการเชื่อมต่อเครือข่ายและ การรับ-ส่งข้อมูลสามารถดำเนินต่อไป แม้ในขณะที่มีสายติดต่อเข้ามา จึงทำให้การรับ-ส่งข้อมูลไม่ขาดตอน
- 3) การสื่อสารอินเทอร์เน็ตแบบไร้สายผ่านเครือข่ายโทรศัพท์เคลื่อนที่จีเอสเอ็ม

2.6 การสื่อสารข้อมูลบนเครือข่ายอินเทอร์เน็ตด้วยโปรโตคอล HTTP

เครือข่ายอินเทอร์เน็ตเป็นเครือข่ายสาธารณะที่เชื่อมโยงถึงกันทั่วโลก เครื่องคอมพิวเตอร์ในเครือข่ายได้รับหมายเลข IP (IP Address) เพื่อใช้ระบุตัวตนบนเครือข่ายอินเทอร์เน็ต โดยมีโปรโตคอลที่ควบคุมกลไกการทำงานของ TCP/IP (Transfer Control Protocol/Internet Protocol) อีกทีสำหรับใช้ในการแลกเปลี่ยนข้อมูล โปรโตคอลสำหรับสื่อสารข้อมูลที่นิยมใช้ในปัจจุบัน เช่น HTTP (HyperText Transfer Protocol) FTP (File Transfer Protocol) และ E-mail เป็นต้น

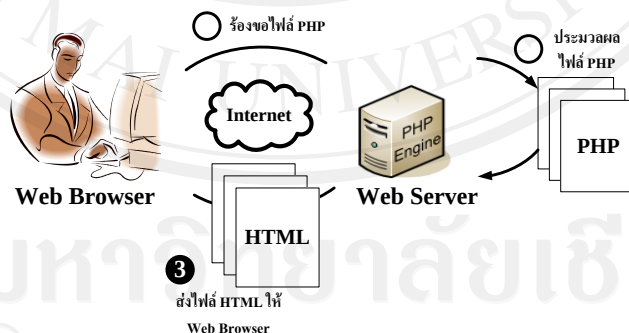
HTTP [11] เป็นโปรโตคอลหนึ่งที่ถูกใช้สื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์บนเครือข่ายอินเทอร์เน็ตของเทคโนโลยีเวิลด์ไวด์เว็บ หลักการทำงานของ HTTP แสดงดังรูป 2.13 แบ่งออกเป็น 2 ด้านคือด้านคอมพิวเตอร์ไคลเอนต์และด้านคอมพิวเตอร์เว็บเซิร์ฟเวอร์ เริ่มจากคอมพิวเตอร์ไคลเอนต์ทำการติดต่อและร้องขอข้อมูลเว็บเพจไปยังเครื่องคอมพิวเตอร์เซิร์ฟเวอร์ด้วยโปรแกรมเว็บเบราว์เซอร์โดยใช้ URL (Uniform Resource Locators) อ้างถึงแอดเดรสของเครื่องคอมพิวเตอร์เซิร์ฟเวอร์ เรียกกระบวนการนี้ว่า *HTTP Request* จากนั้นเครื่องคอมพิวเตอร์เซิร์ฟเวอร์ส่งข้อมูล

ตอบกลับมาในรูปแบบของภาษา HTML (HyperText Markup Language) สำหรับใช้อธิบายเนื้อหาของเว็บเพจ กระบวนการส่วนนี้ถูกเรียกว่า *HTTP Response*



รูป 2.13 กลไกการทำงานของโปรโตคอล HTTP ของเทคโนโลยีเวิร์ลด์ไวด์เว็บ

ผลลัพธ์ที่ได้จาก HTTP Request ด้วย URL เดิมจะได้ HTTP Response หรือเนื้อหาเว็บเพจเดิมตลอด จึงมีแนวคิดในการพัฒนาเว็บเพจด้วยภาษาที่สามารถทำให้เนื้อหาของเว็บเพจเปลี่ยนแปลงตามเงื่อนไขของกระบวนการ HTTP Request ได้ ภาษา PHP [12] เป็นภาษาหนึ่งที่มีความสามารถดังกล่าว ด้วยการแทรกพารามิเตอร์หรือเงื่อนไขไปกับ URL และส่งให้คอมพิวเตอร์เซิร์ฟเวอร์ประมวลผล จากนั้นเนื้อหาเว็บเพจตามผลลัพธ์ที่ได้จะถูกส่งกลับให้คอมพิวเตอร์ไคลเอนท์ในรูปแบบภาษา HTML อีกที ทำให้การแสดงผลเว็บเพจมีความหลากหลายมากขึ้น มีการทำงานแสดงดังรูป 2.14



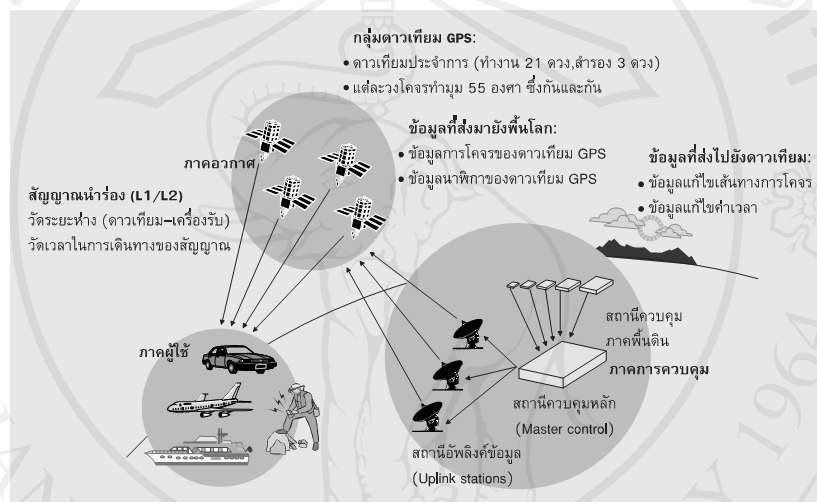
รูป 2.14 กลไกการทำงานของภาษา PHP ร่วมกับโปรโตคอล HTTP

โปรโตคอล HTTP ในปัจจุบันพัฒนาขึ้นมาเป็นเวอร์ชัน 1.1 (จากเดิมคือเวอร์ชัน 1.0) และมีวิธี (Method) ในการสื่อสารข้อมูลหลักๆ 2 แบบคือวิธี GET และวิธี POST จุดเด่นของวิธี GET คือสามารถแทรกข้อมูลร้องขอต่อท้าย URL ได้ทันทีโดยไม่ต้องมีส่วนหัวกำกับข้อมูล (Header) เช่นการแสดงผลเว็บเพจโดยใช้เลขหน้าเป็นพารามิเตอร์ในตัวอย่างดังนี้

www.yourweb.com/viewpage.php?page=2

ในตัวอย่างนี้ URL คือ `www.yourweb.com/viewpage.php` โดย `viewpage.php` คือเว็บเพจที่พัฒนาด้วยภาษา PHP และหลังเครื่องหมาย “?” คือเงื่อนไขที่ร้องขอ `page=2` (หน้าที่ 2) หากเปลี่ยนตัวเลขเว็บเพจก็แสดงตามเนื้อหาหน้านั้นๆ เป็นต้น นอกจากนี้ยังสามารถแทรกได้มากกว่าหนึ่งเงื่อนไขต่อการ HTTP Request หนึ่งครั้ง (สามารถศึกษาเพิ่มเติมได้จากรายละเอียดของโปรโตคอล HTTP และภาษา PHP)

2.7 ระบบจีพีเอส (Global Positioning System, GPS)



รูป 2.15 ภาพรวมการทำงานของระบบจีพีเอส

ในรูป 2.15 แสดงหลักการทำงานของระบบจีพีเอส คือการรับสัญญาณที่ส่งจากกลุ่มดาวเทียมจีพีเอสที่โคจรรอบโลกตลอดเวลา ด้วยเครื่องรับสัญญาณจีพีเอสที่อยู่บนพื้นโลก โดยได้กำหนดเป็นมาตรฐานในการตีความหมายและรูปแบบของสัญญาณซึ่งเรียกว่า มาตรฐาน NMEA ข้อมูลจากกลุ่มดาวเทียมจีพีเอสอย่างน้อย 4 ดวงถูกคำนวณและถอดรหัสเพื่อหาพิกัดตำแหน่งบนพื้นโลก ขณะนั้นของเครื่องรับสัญญาณจีพีเอส จากนั้นรายงานพิกัดดังกล่าวในรูปแบบของข้อความ NMEA ซึ่งประกอบด้วยเรคอร์ดและฟิลด์ย่อย มาตรฐาน NMEA แต่ละเวอร์ชันมีเรคอร์ดข้อมูลที่ต่างกัน และมีเรคอร์ดที่ใช้เป็นหลัก 6 เรคอร์ดดังนี้

- 1) GGA (Global Positioning System Fixed Data)
- 2) GLL (Geographic Position - Latitude/Longitude)
- 3) GSA (GNSS DOP and Active Satellites)
- 4) GSV (GNSS Satellites in View)

5) RMC (Recommended Minimum Specific GNSS)

6) VTG (Course Over Ground and Ground Speed)

ในแต่ละเรคอร์ดประกอบด้วยฟิลด์ข้อมูลย่อยแตกต่างกันไป เช่น ละติจูด ลองจิจูด เวลา ความเร็วและระดับความสูงของระดับน้ำทะเล เป็นต้น ซึ่งสามารถประยุกต์ใช้ฟิลด์ข้อมูลจากแต่ละเรคอร์ดอย่างแพร่หลายทั้งทางบก ทางอากาศหรือทางทะเล เช่น ติดตั้งเครื่องรับสัญญาณจีพีเอสบนรถยนต์เพื่อนำทาง การเดินป่า การเดินเรือและระบบการขนส่ง เป็นต้น ตาราง 2.13 แสดงตัวอย่างข้อมูลในเรคอร์ด GGA เรคอร์ดนี้ประกอบด้วยข้อมูลซึ่งใช้บอกตำแหน่งพิกัด ละติจูด , ลองจิจูด, เวลา, จำนวนดาวเทียมที่ใช้คำนวณพิกัด (satellite used) และความสูงจากระดับน้ำทะเล (MSL Altitude) เป็นต้น

ตาราง 2.122 ฟิลด์ข้อมูลย่อยในเรคอร์ด GGA [13]

NAME	Example	Units	Description
Message ID	\$GPGGA		GGA protocol header
UTC Position	161229.487		hhmmss.sss
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		Dddmm.mmmm
E/W Indicator	W		E=east or W=west
Position Fix Indicator	1		
Satellite Used	07		Range 0 to 12
HDOP	1.0		Horizontal Dilution of Precision
MSL Altitude	9.0	Meters	
Units	M	Meters	
Geoid Separation		meters	
Units	M	Meters	
Age of Diff. Corr.		Second	Null fields when DGPS is not used
Diff. Ref. Station ID	0000		
Checksum	*18		
<CR><LF>			End of message termination