

บทที่ 4

การสร้างและทดสอบระบบ OMD

เนื้อหาในบทนี้ประกอบด้วยการสร้างและทดสอบระบบ OMD เริ่มจากการสร้างระบบ OMD Client และ OMD Server ด้วยการออกแบบที่นำเสนอในบทที่ 3 จากนั้นทำการทดสอบระบบที่สร้างขึ้น ทั้งนี้แบ่งการทดสอบออกเป็นทีละ โหนดโดยให้ทำงานควบคู่กับ โหนด Master ที่ถูกควบคุมการทำงานด้วยซอฟต์แวร์คอมพิวเตอร์สำหรับควบคุมและทดสอบระบบ OMD ท้ายที่สุดจึงทำการต่อทุกโหนดให้ทำงานร่วมกันเพื่อทดสอบการมอนิเตอร์และวินิจัยรถยนต์แบบออนไลน์ผ่านโครงข่ายจีพีอาร์เอส

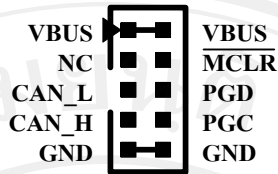
4.1 การสร้างระบบ OMD Client

4.1.1 การสร้างฮาร์ดแวร์

เริ่มจากการนำผังโครงสร้างที่ออกแบบไว้มาเขียนเป็นผังวงจร (Schematic) แล้วออกแบบแผ่นวงจรพิมพ์ (Print Circuit Board) และดำเนินการวางอุปกรณ์บนแผ่นวงจรพิมพ์ด้วยซอฟต์แวร์สำหรับวาดลายวงจร จากนั้นจึงนำลายวงจรที่ได้ไปสร้างเป็นแผ่นวงจรพิมพ์ที่สมบูรณ์ ขั้นตอนสุดท้ายทำการบัดกรีอุปกรณ์ลงบนแผ่นวงจรพิมพ์ให้เรียบร้อย แล้วทำการทดสอบเบื้องต้นก่อนทำการบรรจุเฟิร์มแวร์ลงในไมโครคอนโทรลเลอร์ของแต่ละโหนด ทั้งนี้สามารถสรุปสมบัติทางเทคนิคของโหนดทั่วไปได้ดังนี้

- เป็นแผ่นวงจรพิมพ์หน้าเดียวหรือสองหน้า มีขนาดกระทัดรัดและสามารถติดตั้งในรถยนต์ได้ง่าย โดยมีขนาดเฉลี่ยกว้าง 3 นิ้ว x ยาว 4 นิ้ว (แสดงผังวงจรของแต่ละโหนดไว้ในภาคผนวก)
- ใช้ไมโครคอนโทรลเลอร์เบอร์ PIC18F2680 เป็นหลัก
- ทำงานด้วยสัญญาณนาฬิกาที่มีความถี่เท่ากับ 11.0592 MHz
- มีวงจรเชื่อมต่อ巴士 CAN
- มีขั้วต่อไอดีซี (IDC) จำนวน 10 ขาสัญญาณ ซึ่งมีตำแหน่งขาสัญญาณดังรูป 4.1 โดยกลุ่มขาสัญญาณ *VBUS*, *CAN_L*, *CAN_H*, *GND* ใช้สำหรับสื่อสารข้อมูล巴士 CAN ส่วนกลุ่มขาสัญญาณ *VBUS*, *MCLR*, *PGD*, *PGC*, *GND* ใช้สำหรับการบรรจุและดีบั๊กเฟิร์มแวร์ของโหนด

- มีวงจรควบคุมไฟเลี้ยงคงที่ที่ 5 โวลต์ โดยมีต้นกำเนิดแรงดันไฟฟ้าจากขาสัญญาณ *VBUS* และ *GND*



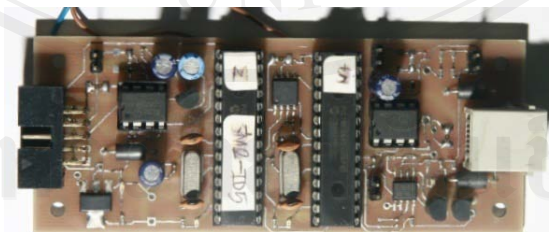
รูป 4.1 ภาพตำแหน่งขาสัญญาณหัวต่อไอซีจำนวน 10 ขาสัญญาณ

โหนด Master มีภาพแผ่นวงจรพิมพ์พร้อมอุปกรณ์ดังรูป 4.2 โดยใช้ผังโครงสร้างของโหนด Master จากรูป 3.14 ในการออกแบบผังวงจร



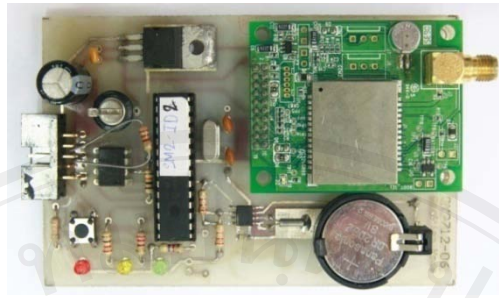
รูป 4.2 ภาพแผ่นวงจรพิมพ์และอุปกรณ์ของโหนด Master

โหนด Gateway มีวงจรเชื่อมต่อกับบัส CAN ของรถยนต์เป็นสมบัติที่แตกต่างจากโหนดอื่น ซึ่งในรูป 4.3 คือภาพแผ่นวงจรพิมพ์พร้อมอุปกรณ์ โดยออกแบบผังวงจรจากผังโครงสร้างของโหนด Gateway จากรูป 3.16



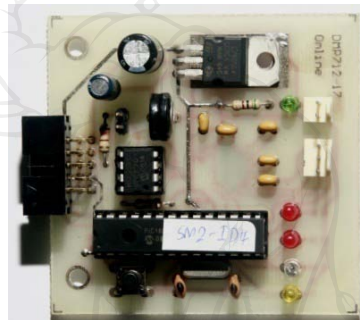
รูป 4.3 ภาพแผ่นวงจรพิมพ์และอุปกรณ์ของโหนด Gateway

โหนด Locator มีวงจรเชื่อมต่อกับไอซีรักษาฐานเวลาจริงและวงจรเชื่อมต่อกับมอดูลจีพีเอส เป็นสมบัติเด่น เนื่องจากมอดูลจีพีเอสมีขนาดเล็กจึงออกแบบให้สามารถวางมอดูลอยู่ในขอบเขตของแผ่นวงจรพิมพ์ได้ดังรูป 4.4 โดยออกแบบผังวงจรจากผังโครงสร้างของโหนด Locator จากรูป 3.18



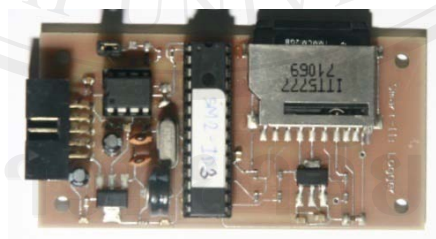
รูป 4.4 ภาพแผ่นวงจรพิมพ์และอุปกรณ์ของ โหนด Locator

โหนด Online มีวงจรเชื่อมต่อกับมอดูลจีพีอาร์เอสเป็นสมบัติเด่น ซึ่งมีภาพแผ่นวงจรพิมพ์พร้อมอุปกรณ์ดังรูป 4.5 (ไม่นับรวมตัวมอดูลจีพีอาร์เอสเนื่องจากมีขนาดใหญ่) และออกแบบผังวงจรจากผังโครงสร้างของโหนด Online จากรูป 3.19



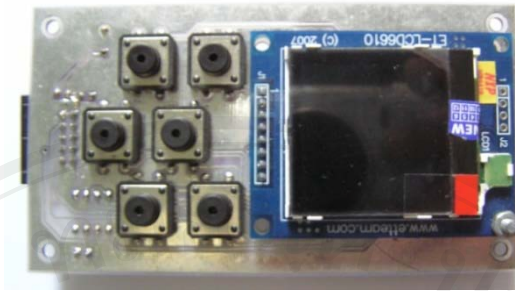
รูป 4.5 ภาพแผ่นวงจรพิมพ์และอุปกรณ์ของ โหนด Online

โหนด Logger มีวงจรเชื่อมต่อกับการ์ด MMC เป็นสมบัติเด่นและมีภาพแผ่นวงจรพิมพ์พร้อมอุปกรณ์ดังรูป 4.6 ซึ่งออกแบบผังวงจรโดยใช้ผังโครงสร้างของโหนด Logger จากรูป 3.21



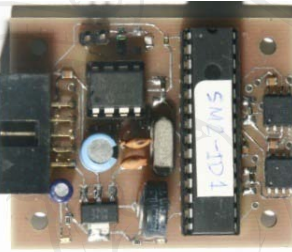
รูป 4.6 ภาพแผ่นวงจรพิมพ์และอุปกรณ์ของ โหนด Logger

โหนด HMI มีวงจรเชื่อมต่อกับมอดูลจอแสดงผลและวงจรเชื่อมต่อกับสวิตช์ปุ่มกดเป็นสมบัติเด่น มีภาพแผ่นวงจรพิมพ์ดังรูป 4.7 ซึ่งออกแบบผังวงจรโดยใช้ผังโครงสร้างของโหนด HMI จากรูป 3.22



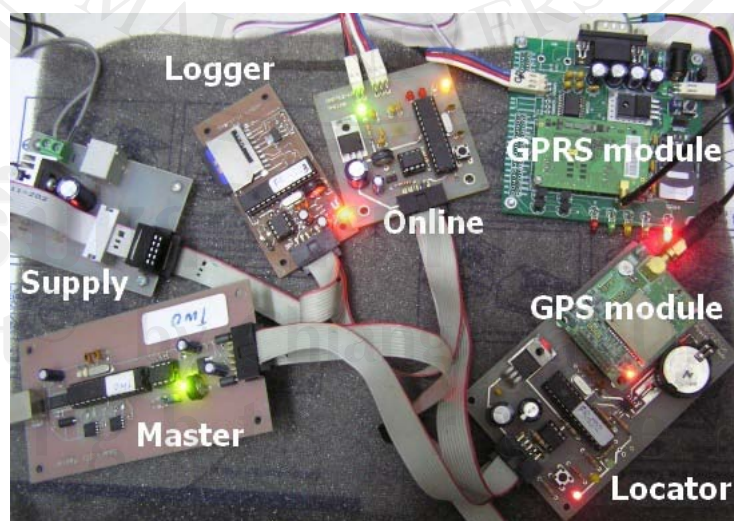
รูป 4.7 ภาพแผ่นวงจรพิมพ์และอุปกรณ์ของโหนด HMI

โหนด Calculator มีวงจรเชื่อมต่อกับอีมูมิเตอร์เป็นสมบัตินี้และมีภาพแผ่นวงจรพิมพ์ดังรูป 4.8 ซึ่งออกแบบผังวงจรโดยใช้ผังโครงสร้างของโหนด Calculator จากรูป 3.23

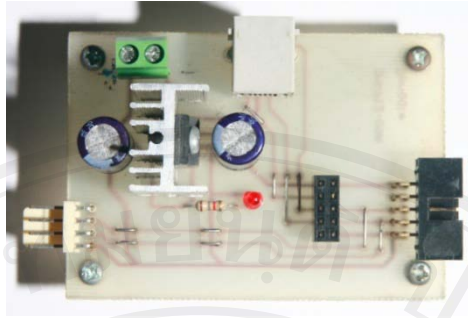


รูป 4.8 ภาพแผ่นวงจรพิมพ์และอุปกรณ์ของโหนด Calculator

ในรูป 4.9 แสดงภาพการต่อใช้ฮาร์ดแวร์ทุกโหนดร่วมกันด้วยขั้วต่อไอดีซีและสายแพ (Ribbon Cable) จำนวน 10 เส้น โดยมีแผ่นวงจรพิมพ์ดังรูป 4.10 ที่ใช้สำหรับจ่ายแรงดันไฟฟ้า 9 โวลต์ให้ขาสัญญาณ *VBUS* และ *GND*



รูป 4.9 ภาพการต่อใช้ฮาร์ดแวร์ทุกโหนดร่วมกัน



รูป 4.10 ภาพแผ่นวงจรพิมพ์ต้นกำเนิดแรงดันไฟฟ้า

4.1.2 การสร้างเฟิร์มแวร์

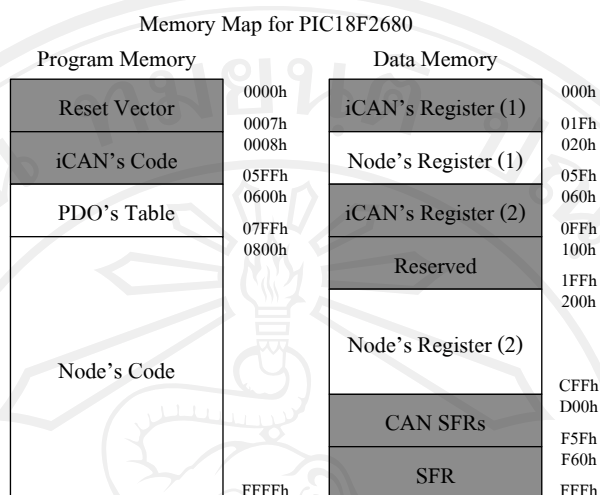
จากการออกแบบที่ได้นำเสนอในบทที่ 3 จึงนำมาสร้างเป็นเฟิร์มแวร์ของทุกโหมด โดยใช้ภาษาแอสเซมบลี (Assembly, ASM) และโปรแกรม MPASMWIN ซึ่งเป็นโปรแกรมสำหรับประมวลผลโค้ดภาษาแอสเซมบลีให้เป็นภาษาเครื่อง (Machine Code) ก่อนถูกบรรจุลงในไมโครคอนโทรลเลอร์ ทั้งนี้เฟิร์มแวร์ของแต่ละโหมดประกอบด้วยโค้ดภาษาแอสเซมบลีไฟล์ย่อยหลายๆ ไฟล์ทำงานร่วมกัน ซึ่งไฟล์ดังกล่าวถูกเรียกใช้จากโค้ดไฟล์หลักอีกที สามารถแสดงโครงสร้างโค้ดแอสเซมบลีของไฟล์หลักได้ดังนี้

```
1.  #INCLUDE <P18F2680.INC>
2.  #INCLUDE "node_name.def"
3.  #INCLUDE "..\iCAN.asm"
4.  #INCLUDE "node_name.asm"
5.  END
```

พิจารณาโค้ดในบรรทัดที่ 1 ถึง 4 คำสั่ง #INCLUDE คือคำสั่งให้โปรแกรม MPASMWIN หรือที่เรียกว่า “แอสเซมเบลอร์ (Assembler)” เรียกใช้ไฟล์ตามพาร์ทที่ระบุต่อท้ายคำสั่ง (File's Path) เพื่อนำไปใช้ประมวลผลร่วมกัน ส่วนในบรรทัดที่ 5 คำสั่ง END คือคำสั่งที่บอกให้แอสเซมเบลอร์ทราบว่าเป็นบรรทัดสุดท้ายของโค้ด (สามารถศึกษาการเขียนโปรแกรมภาษาแอสเซมบลีจากเอกสารของบริษัทไมโครชิพจำกัด) ซึ่งรายละเอียดของแต่ละไฟล์ประกอบด้วยกลุ่มคำสั่งที่ควบคุมให้ไมโครคอนโทรลเลอร์ทำตามฟังก์ชันงานที่ต้องการ

ในรูป 4.11 แสดงแผนภูมิพื้นที่โปรแกรม (Program memory) และพื้นที่หน่วยความจำ (Data Memory) ของไมโครคอนโทรลเลอร์เบอร์ PIC18F2680 โดยพื้นที่โปรแกรมคือพื้นที่สำหรับบรรจุโค้ดภาษาเครื่องซึ่งเป็นผลลัพธ์จากการประมวลผลไฟล์ทั้ง 4 ไฟล์ร่วมกัน (“P18F2680.INC”, “node_name.def”, “iCAN.asm” และ “node_name.asm”) ส่วนพื้นที่หน่วยความจำคือตำแหน่งหน่วยความจำหรือรีจิสเตอร์ที่ถูกประกาศและเรียกใช้จากโค้ดดังกล่าว ทั้งนี้พื้นที่ส่วนที่แรงแงคือตำแหน่งที่เฟิร์มแวร์ของแต่ละโหมดใช้เหมือนกัน นอกจากนี้การประมวลผลของแอสเซมเบลอร์

เป็นการประมวลผลโค้ดจากบรรทัดแรกเรียงลงไปตามลำดับ ดังนั้นตำแหน่งของโค้ดและรีจิสเตอร์จึงสัมพันธ์กับลำดับการเรียกใช้ในแต่ละไฟล์ที่ถูกประมวลผลด้วย



รูป 4.11 แผนภูมิพื้นที่โปรแกรมและพื้นที่หน่วยความจำของ PIC18F2680

จากโค้ดไฟล์แรกที่ถูกเรียกใช้คือไฟล์ “P18F2680.INC” ซึ่งเป็นไฟล์ที่นิยามตำแหน่งและความหมายของกลุ่มรีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register, SFR) สำหรับใช้กับไมโครคอนโทรลเลอร์เบอร์ PIC18F2680 ไว้ (อยู่ในพื้นที่หน่วยความจำส่วนท้าย) นักพัฒนาสามารถควบคุมมอดูลทำงานต่างๆ ผ่านทางรีจิสเตอร์ฟังก์ชันพิเศษดังกล่าวเช่น รีจิสเตอร์ PORTn สำหรับพอร์ตอินพุต-เอาต์พุต รีจิสเตอร์ TXREG และ RCREG สำหรับมอดูลทำงาน UART รีจิสเตอร์ TXBnCON และ RXBnCON สำหรับมอดูล CAN คอนโทรลเลอร์ เป็นต้น

ไฟล์ที่ 2 ที่ถูกเรียกใช้คือไฟล์ “node_name.def” (มีชื่อไฟล์ตามชื่อของโหนด) ซึ่งใช้นิยามตำแหน่งและความหมายของรีจิสเตอร์ที่โหนดนั้นๆ ประกาศใช้ โดยกำหนดช่วงตำแหน่งพื้นที่หน่วยความจำอยู่ตรงส่วน Node's Register (1), (2) ในรูป 4.11 ส่วนไฟล์ที่ 3 คือไฟล์ “iCAN.asm” เป็นไฟล์ที่ประกอบด้วยกลุ่มคำสั่งสำหรับควบคุมให้ไมโครคอนโทรลเลอร์ทำงานตามรูทีนของเฟิร์มแวร์ส่วนคอมมิวนิเคชันและส่วนประยุกต์ดังที่ได้นำเสนอในบทที่ 3 มีช่วงตำแหน่งพื้นที่โปรแกรมและพื้นที่หน่วยความจำอยู่ตรงส่วน iCAN's Code และ iCAN's Register (1), (2) ตามลำดับดังแสดงในรูป 4.11

ส่วนไฟล์ที่ 4 ที่เรียกใช้คือไฟล์ “node_name.asm” (เรียกรวมทุกไฟล์ที่สร้างขึ้นสำหรับแต่ละโหนด) เป็นไฟล์ที่บรรจุโค้ดแอสเซมบลีสำหรับเรียกใช้รูทีนย่อยและฟังก์ชันทำงานของโหนดนั้นๆ ตามที่ได้ออกแบบไว้ในบทที่ 3 ทั้งนี้มีขอบเขตตำแหน่งพื้นที่โปรแกรมอยู่ตรงส่วน Node's Code ในรูป 4.11 นอกจากนี้เมื่อสรุปการใช้ทรัพยากรของส่วนที่ทุกโหนดใช้เหมือนกัน (ส่วนที่แรกๆ)

พบว่าในเบื้องต้นทุกโหนดใช้พื้นที่โปรแกรมขนาด 2 กิโลไบต์จากที่มี 64 กิโลไบต์ (คิดเป็น 3.1 %) และใช้พื้นที่หน่วยความจำขนาด 256 ไบต์จากที่มี 3 กิโลไบต์ (คิดเป็น 7.7%) ทำให้เหลือพื้นที่โปรแกรมและหน่วยความจำเพียงพอที่ใช้พัฒนาไฟล์ “node_name.def” และ “node_name.asm” ของแต่ละโหนดได้

4.2 การทดสอบระบบ OMD Client

การทำงานของแต่ละโหนดถูกทดสอบด้วยซอฟต์แวร์คอมพิวเตอร์ที่พัฒนาขึ้นร่วมกับโหนด Master โดยเน้นทดสอบโปรโตคอลสื่อสารระหว่างโหนดในระบบ OMD Client เช่น การเปลี่ยนสถานะงานของแต่ละโหนด บริการสื่อสาร SDO และ PDO เป็นต้น นอกจากนี้ซอฟต์แวร์คอมพิวเตอร์ที่พัฒนาขึ้น ยังมีจุดประสงค์เพื่อสังเกตการทำงานของโหนด Master ให้สามารถควบคุมการทำงานรวมของระบบ OMD Client ได้เมื่อไม่ได้ต่อกับคอมพิวเตอร์ส่วนบุคคล ทั้งนี้ รายละเอียดเกี่ยวกับซอฟต์แวร์คอมพิวเตอร์และการทดสอบแสดงในหัวข้อถัดไป

4.2.1 ซอฟต์แวร์คอมพิวเตอร์สำหรับทดสอบ OMD Client

ในเบื้องต้นเพื่อทดสอบการทำงานของ OMD Client จึงได้พัฒนาซอฟต์แวร์คอมพิวเตอร์สำหรับควบคุมการสื่อสารข้อมูลระหว่างโหนดและควบคุมแต่ละโหนดให้ทำงานร่วมกัน งานวิจัยนี้เลือกใช้ Visual C# 2008 Express Edition ของบริษัทไมโครซอฟท์จำกัด (Microsoft Inc.) เป็นเครื่องมือในการพัฒนาซอฟต์แวร์ด้วยภาษา C# และเรียกซอฟต์แวร์คอมพิวเตอร์ที่ถูกพัฒนานี้ว่า “Virtual Master” มีภาพซอฟต์แวร์ดังรูป 4.12 ซึ่งแบ่งการทำงานของซอฟต์แวร์ออกเป็น 4 ส่วนหลักๆ ได้แก่

1) ส่วนข้อกำหนดเกี่ยวกับการรับ-ส่งข้อมูลผ่านทางพอร์ตยูเอสบี (Virtual COM Port) ประกอบด้วยรายชื่อของพอร์ต ปุ่ม Connect ปุ่ม Disconnect และแถบแสดงสถานะ ซึ่งมีค่าตั้งต้นสำหรับสื่อสารข้อมูลที่อัตราบอดเท่ากับ 57,600 บิตต่อวินาที ใช้ข้อมูลขนาด 8 บิตไม่มีการตรวจสอบพาริตีบิตและมี 1 บิตหยุด

2) ส่วนหน้าต่างแสดงผล

2.1) หน้าต่างแสดงรายละเอียดของคำสั่งที่ต้องการให้โหนด Master ส่งให้แก่บัส CAN ซึ่งคำสั่งได้จากการพิมพ์หรือได้จากการโหลดแฟ้มข้อความ (Text File) ด้วยปุ่ม Load ในส่วนที่ 3

2.2) หน้าต่างแสดงรายละเอียดของข้อมูลที่โหนด Master ได้รับจากบัส CAN

3) ส่วนข้อกำหนดเกี่ยวกับการส่งคำสั่งให้กับโหนด Master ประกอบด้วยปุ่ม Load ปุ่ม Send ปุ่มควบคุมโหมดทำงานของโหนด Master (P, E และ N) ปุ่ม SYNC (ให้ส่งข้อความ SYNC ตามคาบเวลาที่กำหนด) และคาบเวลาการส่งคำสั่งที่ต่อเนื่องกัน

4) ส่วนบันทึกข้อมูลจากโหนด Master ที่แสดงในหน้าต่างแสดงผล 2.2 ให้อยู่ในรูปแบบของเพิ่มข้อความ

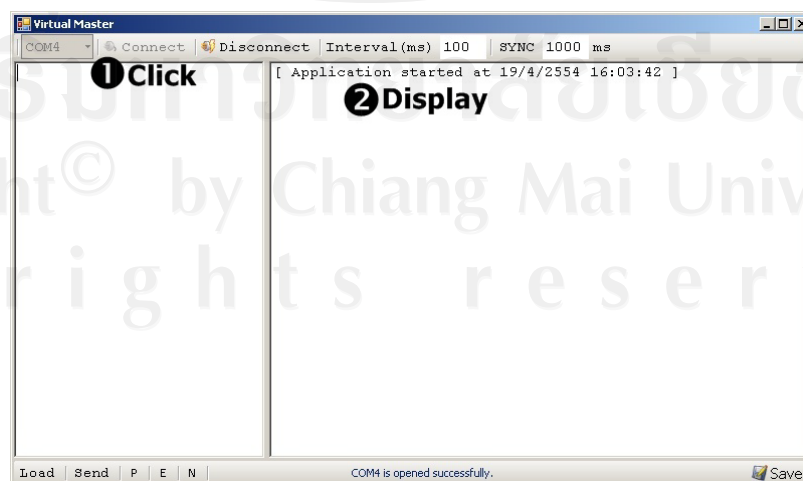


รูป 4.12 ภาพซอฟต์แวร์ Virtual Master ที่พัฒนาด้วยภาษา C#

4.2.2 การทดสอบการทำงานของโหนด Master และซอฟต์แวร์ Virtual Master

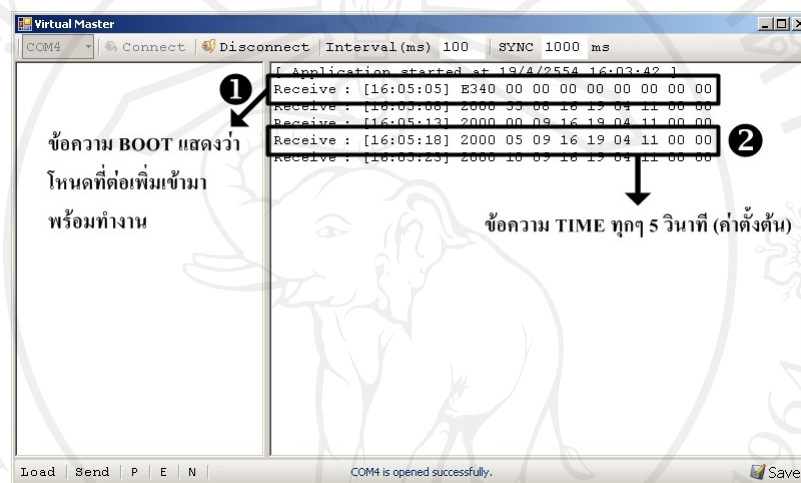
ทำการทดสอบการทำงานของซอฟต์แวร์ Virtual Master และการสื่อสารข้อมูลด้วยแพ็คเกจข้อมูล UART กับโหนด Master ทั้งนี้ได้ต่อโหนด Locator เป็นโหนดทำงานในระบบเพื่อทดสอบบริการสื่อสาร NMT, SDO, PDO, TIME และ SYNC ด้วย ซึ่งมีขั้นตอนการทดสอบดังนี้

- 1) คิดตั้งการทดสอบโดยต่อสายยูเอสบีจากโหนด Master ไปยังคอมพิวเตอร์ส่วนบุคคล แต่ยังไม่ต่อต่อโหนด Locator
- 2) เปิดซอฟต์แวร์ Virtual Master แล้วเลือกพอร์ตยูเอสบีที่ต้องการให้เรียบร้อย จากนั้นกดปุ่ม Connect ตามหมายเลข 1 ดังรูป 4.13 สุดท้ายมีข้อความปรากฏตามหมายเลข 2 เมื่อซอฟต์แวร์ Virtual Master เริ่มทำงาน



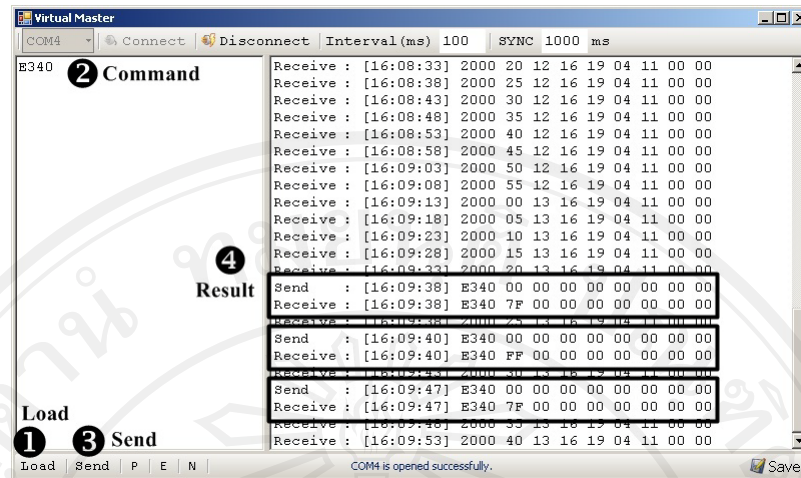
รูป 4.13 ภาพซอฟต์แวร์ Virtual Master หลังจากกดปุ่ม Connect

3) ทำการต่อโหนด Locator ร่วมกับโหนด Master เมื่อโหนด Locator เริ่มทำงานจึงแสดงผลดังรูป 4.14 ปรากฏเป็นข้อความซึ่งข้อความแรกตรงหมายเลข 1 คือ ข้อความ BOOT ที่แสดงว่าโหนด Locator พร้อมทำงานแล้ว โดยมีหมายเลขประจำตัวข้อความเป็น 0xE340 และไม่มีฟิลด์ข้อมูล ส่วนข้อความถัดมาตรงหมายเลข 2 คือ ข้อความ TIME ที่ส่งออกมาจากโหนด Locator ทุกๆ 5 วินาที (ค่าตั้งต้น) มีหมายเลขประจำตัวข้อความเป็น 0x2000 และมีฟิลด์ข้อมูลจำนวน 6 ไบต์ ประกอบด้วยค่าวินาที นาที ชั่วโมง วันที่ เดือน ปี โดยเรียงตามลำดับไบต์ข้อมูลที่ปรากฏ



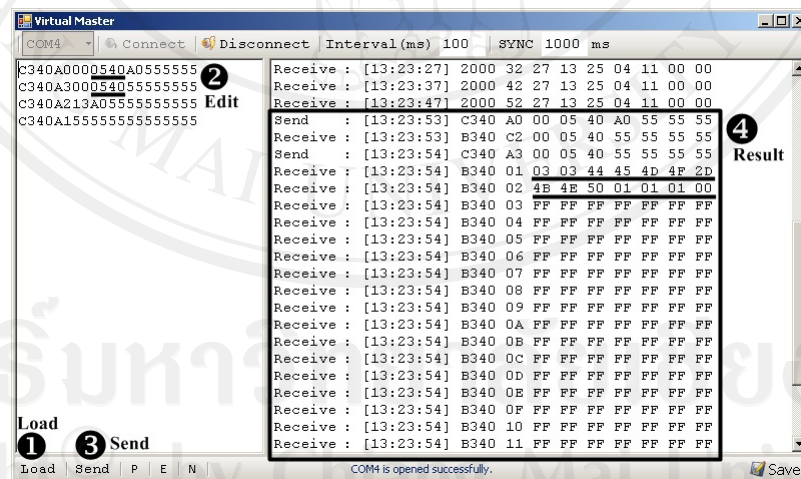
รูป 4.14 ภาพซอฟต์แวร์ Virtual Master เมื่อต่อโหนดทำงานกับบัส CAN

4) ตรวจสอบสถานะงานปัจจุบันของโหนด Locator ด้วยข้อความ GUARD (หรือก็คือข้อความ BOOT จากขั้นตอนข้อ 3) การส่งคำสั่งหรือข้อความที่ต้องการนั้น สามารถโหลดข้อความด้วยปุ่ม Load ดังรูป 4.15 ตรงหมายเลข 1 หรือพิมพ์ข้อความตรงหมายเลข 2 จากนั้นจึงเริ่มส่งด้วยการกดปุ่ม Send ส่วนตรงหมายเลข 4 ในรูป 4.15 แสดงผลตอบกลับจากโหนด Locator เมื่อได้รับข้อความ GUARD 3 ครั้ง (กดปุ่ม Send 3 ครั้ง) โดยผลลัพธ์คือข้อความ GUARD ที่มีฟิลด์ข้อมูลจำนวน 1 ไบต์ ซึ่งใช้อธิบายสถานะงานปัจจุบันของโหนด Locator มีค่าเท่ากับ 0x7F (สถานะ Pre-operational) ทั้งนี้บิต 7 ของข้อมูลสถานะงานที่ได้รับมีค่าสลับกันไปทุกครั้งที่โหนดได้รับข้อความ GUARD (0x7F กับ 0xFF)



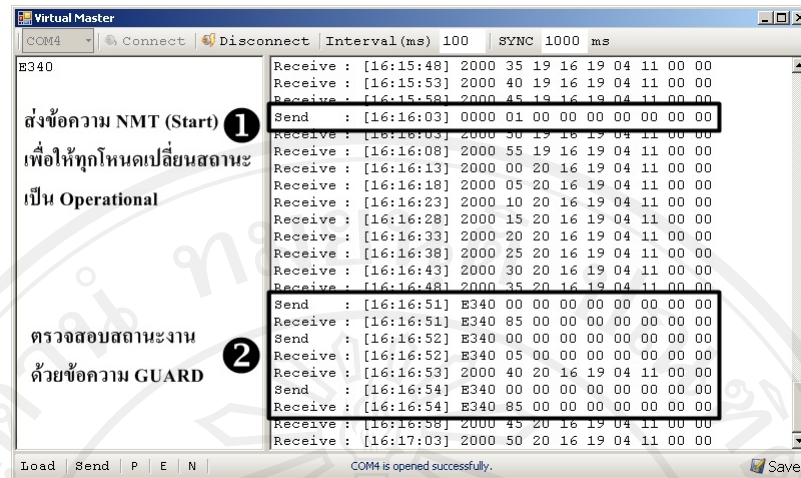
รูป 4.15 ภาพซอฟต์แวร์ Virtual Master เมื่อส่งข้อความ GUARD

5) ทดสอบการสื่อสารข้อมูลด้วยบริการ SDO_RdIntTAB (บริการพื้นฐานสำหรับอ่านข้อมูลพื้นที่โปรแกรมตรงตำแหน่งที่ต้องการ) โดยทำการร้องขออ่านรายละเอียดของโหนด Locator จากพื้นที่โปรแกรมตำแหน่ง 0x0540 เริ่มด้วยการกดปุ่ม Load เพื่อโหลดชุดข้อความ SDO จากแฟ้มข้อความที่เตรียมไว้ ปรากฏชุดข้อความตรงหมายเลข 2 ในรูป 4.16 แก้ไขข้อมูลในข้อความที่ต้องการส่ง (ตำแหน่งพื้นที่โปรแกรมที่ต้องการ) จากนั้นกดปุ่ม Send ตรงหมายเลข 3 เมื่อขั้นตอนการสื่อสารเสร็จสิ้นควรได้ผลลัพธ์ตรงหมายเลข 4 ส่วนที่ขีดเส้นใต้ดังรูป 4.16



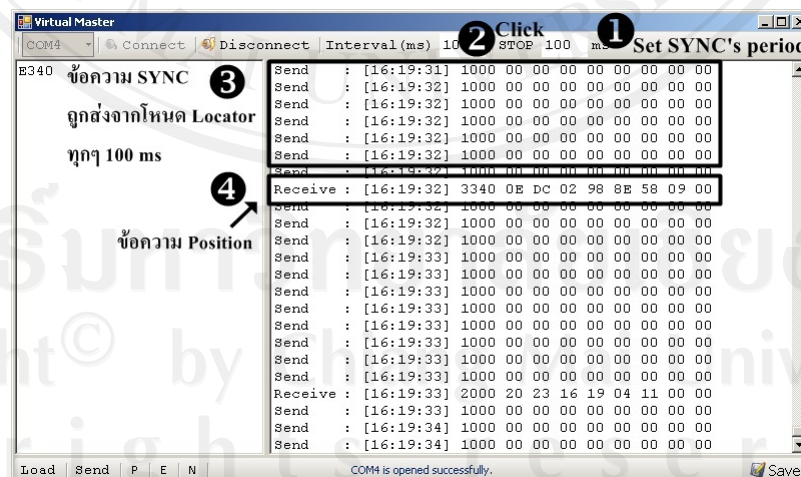
รูป 4.16 ภาพซอฟต์แวร์ Virtual Master เมื่อทดสอบบริการ SDO_RdIntTAB

6) ทดสอบการควบคุมสถานะงานของโหนด Locator ให้อยู่ในสถานะ Operational โดยส่งข้อความ NMT (Start) ดังแสดงในรูป 4.17 ตรงหมายเลข 1 จากนั้นตรวจสอบสถานะงานด้วยข้อความ GUARD ดังรูป 4.17 หมายเลข 2 แสดงผลตอบกลับจากโหนด Locator มีค่าสถานะงานเท่ากับ 0x05 (Operational)



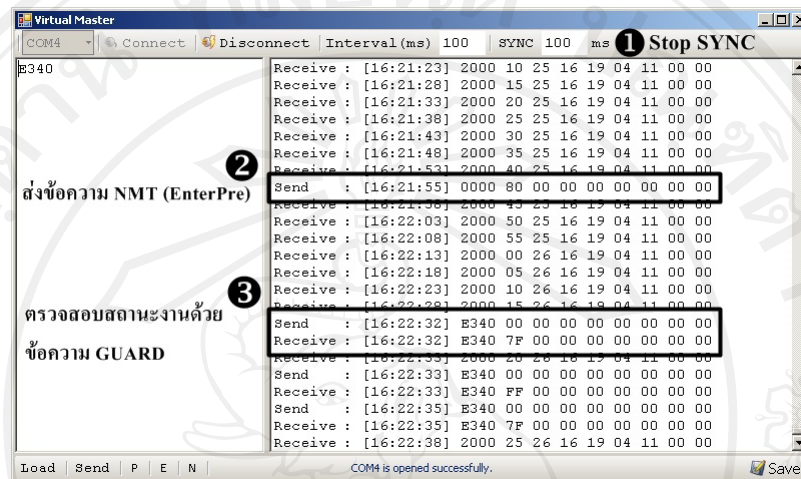
รูป 4.17 ภาพซอฟต์แวร์ Virtual Master เมื่อส่งข้อความ NMT (Start)

7) เมื่อโหนด Locator อยู่ในสถานะงาน Operational ทำการทดสอบการสร้างข้อความตำแหน่ง (Position) ด้วยให้โหนด Master ส่งข้อความ SYNC ให้แก่บัส CAN โดยตั้งคาบเวลาในการส่งตรงหมายเลข 1 ดังรูป 4.18 และกดปุ่ม SYNC ตรงหมายเลข 2 เพื่อเริ่มส่ง (ปรากฏเป็นปุ่ม STOP หลังจากกดปุ่ม SYNC) ซึ่งในรูปตั้งคาบเวลาในการส่งข้อความ SYNC ไว้เท่ากับ 100 มิลลิวินาที จากนั้นเมื่อโหนด Locator ได้รับข้อความ SYNC ครบตามเงื่อนไขที่กำหนดไว้จึงสร้างข้อความตำแหน่งให้แก่บัส CAN (ค่าเริ่มต้นคือทุกๆ จำนวน 50 ข้อความ SYNC) ในรูป 4.18 ตรงหมายเลข 3 แสดงข้อความ SYNC ที่โหนด Master สร้างออกไปและตรงหมายเลข 4 แสดงข้อความตำแหน่งที่ได้รับ



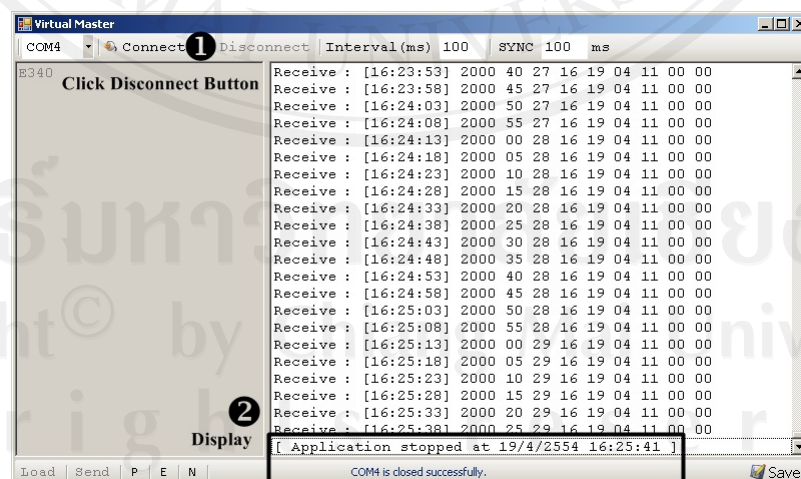
รูป 4.18 ภาพซอฟต์แวร์ Virtual Master ขณะได้รับข้อความ Position

8) ทดสอบการเปลี่ยนสถานะงานกลับเป็น Pre-operational เริ่มจากการกดปุ่ม STOP ตรงหมายเลข 1 ดังรูป 4.19 (ปรากฏเป็นปุ่ม SYNC หลังจากกดปุ่ม STOP) จากนั้นส่งข้อความ NMT (EnterPre) มีรายละเอียดข้อความดังหมายเลข 2 สุดท้ายตรวจสอบสถานะงานด้วยข้อความ GUARD ได้ผลลัพธ์ดังหมายเลข 3 ในรูป



รูป 4.19 ภาพซอฟต์แวร์ Virtual Master เมื่อส่งข้อความ NMT (EnterPre)

9) ในรูป 4.20 แสดงภาพเมื่อต้องการหยุดการทำงานซอฟต์แวร์ Virtual Master เริ่มจากกดปุ่ม Disconnect ตรงหมายเลข 1 จากนั้นปรากฏผลลัพธ์เป็นข้อความดังกรอบหมายเลข 2 ในรูป สำหรับการบันทึกผลลัพธ์นั้น ข้อความทั้งหมดที่แสดงในส่วนแสดงผลถูกบันทึกเป็นแฟ้มข้อความนามสกุล “.txt” ทันทีเมื่อกดปุ่ม Save



รูป 4.20 ภาพการหยุดทำงานซอฟต์แวร์ Virtual Master

4.2.3 การทดสอบการทำงานของโหนด Gateway

โหนด Gateway มีรายชื่อบริการ SDO และ PDO ดังตาราง 4.1

ตาราง 4.1 รายชื่อบริการสื่อสารของโหนด Gateway

ชื่อบริการ	ชนิด	คำอธิบาย
SDO_WrIntTAB	D/L	สำหรับเขียนข้อมูลลงในพื้นที่โปรแกรมของไมโครคอนโทรลเลอร์ตรงตำแหน่งที่กำหนด
SDO_RdIntTAB	U/L	สำหรับอ่านข้อมูลจากพื้นที่โปรแกรมของไมโครคอนโทรลเลอร์ตรงตำแหน่งที่กำหนด
SDO_WrExtTAB	D/L	สำหรับเขียนข้อมูลลงในอีพროมตรงตำแหน่งที่กำหนด
SDO_RdExtTAB	U/L	สำหรับอ่านข้อมูลจากอีพโรมตรงตำแหน่งที่กำหนด
SDO_WrD0	D/L	สำหรับเขียนข้อมูลไปที่อุปกรณ์สแกนทูล
SDO_RdD0	U/L	สำหรับอ่านข้อมูลจากอุปกรณ์สแกนทูล
SDO_WrD0TAB	D/L	สำหรับเขียนข้อมูลลงในพื้นที่โปรแกรมของสแกนทูล
SDO_RdD0TAB	U/L	สำหรับอ่านข้อมูลจากพื้นที่โปรแกรมของสแกนทูล
PDO_CarStatus#(1-6)	OUT	ข้อความสถานะของรถยนต์
PDO_CarDTCs	OUT	ข้อความรหัส DTCs ที่ตรวจพบ

ทำการทดสอบความถูกต้องของอุปกรณ์สแกนทูลที่พัฒนาขึ้นด้วยการวินิจฉัยรถยนต์ โดยใช้บริการ SDO_RdD0 ทำฟังก์ชันตามข้อกำหนด SAE J1979 ดังต่อไปนี้

- อ่านค่าสถานะของรถยนต์ (โหมด 01) โดยเลือกค่าสถานะที่สามารถเปรียบเทียบค่าได้จากแผงหน้าปัดรถยนต์เช่น ความเร็วรอบเครื่องยนต์ ความเร็วรถยนต์และอุณหภูมิหม้อน้ำ เป็นต้น
- อ่านค่ารหัส DTCs ของรถยนต์ (โหมด 03) โดยจำลองสถานการณ์ที่ทำให้ตรวจพบรหัสปัญหาของรถยนต์ เช่น ถอดสายเซ็นเซอร์วงจรตรวจสอบ MAF (Mass Air Flow Circuit) เป็นต้น
- สืบค่ารหัสปัญหาของรถยนต์ที่ตรวจพบหลังจากทำการซ่อมบำรุงรถยนต์ (โหมด 04) โดยได้ทำการทดสอบกับรถยนต์ยี่ห้อ Honda รุ่น City 3 และมีขั้นตอนดังนี้

1) ติดตั้งโหนด Master และโหนด Gateway ให้ทำงานร่วมกัน เชียบตัวเชื่อมต่อเพื่อวินิจฉัยรถยนต์เข้ากับโหนด Gateway ดังรูป 4.21 และต่อสายยูเอสบีพอร์ตจากโหนด Master เข้ากับคอมพิวเตอร์ส่วนบุคคล จากนั้นเริ่มการทำงานซอฟต์แวร์ Virtual Master



รูป 4.21 แสดงการเชื่อมต่อกับตัวเชื่อมต่อเพื่อวินิจฉัยรถยนต์

2) บิดกุญแจไปในตำแหน่ง “ON” แต่ยังไม่สตาร์ทเครื่องยนต์ มีภาพแผงหน้าปัดรถยนต์ดังรูป 4.22 ประกอบด้วยเกจวัดและไฟสถานะสามส่วน โดยส่วนขวามือของรูปคือ เกจวัดความเร็วรอบเครื่องยนต์ ซึ่งมีไฟสถานะเกี่ยวกับ เครื่องยนต์ อุณหภูมิหม้อน้ำรถยนต์ และแรงดันแบตเตอรี่ เป็นต้น เกจวัดส่วนตรงกลางคือ เกจวัดความเร็วรถยนต์ สุดท้ายส่วนซ้ายมือของรูปคือเกจวัดปริมาณน้ำมัน



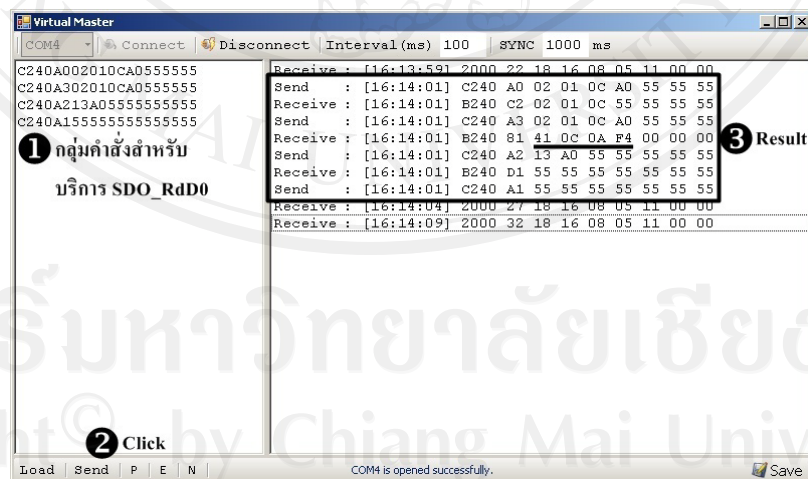
รูป 4.22 แผงหน้าปัดรถยนต์เมื่อบิดกุญแจไปในตำแหน่ง “ON”

3) ทำการสตาร์ทเครื่องยนต์ มีภาพแผงหน้าปัดรถยนต์ดังรูป 4.23 ทำให้เครื่องยนต์อยู่ในสถานะเดินเบาซึ่งมีความเร็วรอบเครื่องยนต์ประมาณ 700 รอบต่อนาที



รูป 4.23 แผงหน้าปัดรถยนต์เมื่อทำการสตาร์ทเครื่องยนต์

4) ทำการโหลดกลุ่มคำสั่งสำหรับทดสอบบริการ SDO_RdD0 มีคำสั่งตามหมายเลข 1 ดังรูป 4.24 แล้วกดปุ่ม Send โหนด Master ทำการส่งคำสั่งดังกล่าวให้โหนด Gateway จากนั้นโหนด Gateway จึงทำการอ่านค่าสถานะของรถยนต์แล้วส่งผลลัพธ์กลับคืนให้โหนด Master ตามหมายเลข 3 ในรูป ในตัวอย่างนี้คือการขออ่านค่าความเร็วรอบเครื่องยนต์ (โหมด 01 พารามิเตอร์ 0C) ผลลัพธ์ที่ได้ประกอบด้วยข้อมูล 4 ไบต์เรียงลำดับจากไบต์ต่ำไปยังไบต์สูง ค่า 0x41 และ 0x0C คือส่วนหัว ส่วนถัดไปคือข้อมูลความเร็วรอบเครื่องยนต์ 2 ไบต์ มีค่าเท่ากับ 0x0AF4 เมื่อคำนวณเป็นเลขจำนวนเต็มได้เท่ากับ 2804 นำไปหารด้วย 4 ได้เป็นค่าความเร็วรอบเครื่องยนต์ มีค่าเท่ากับ 701 รอบต่อนาที



รูป 4.24 ภาพซอฟต์แวร์ Virtual Master เมื่อส่งกลุ่มคำสั่งบริการ SDO_RdD0

นอกจากนี้ได้ทำการทดสอบการทำงานของโหนด Gateway ในสถานะงาน Operational ซึ่งโหนด Gateway ต้องสร้างข้อความ PDO_CarStatus ให้แก่บัส CAN เมื่อโหนดได้รับข้อความ SYNC ทุกๆ จำนวน 4 ข้อความ (ค่าตั้งต้น)

4.2.4 การทดสอบการทำงานของโหนด Locator

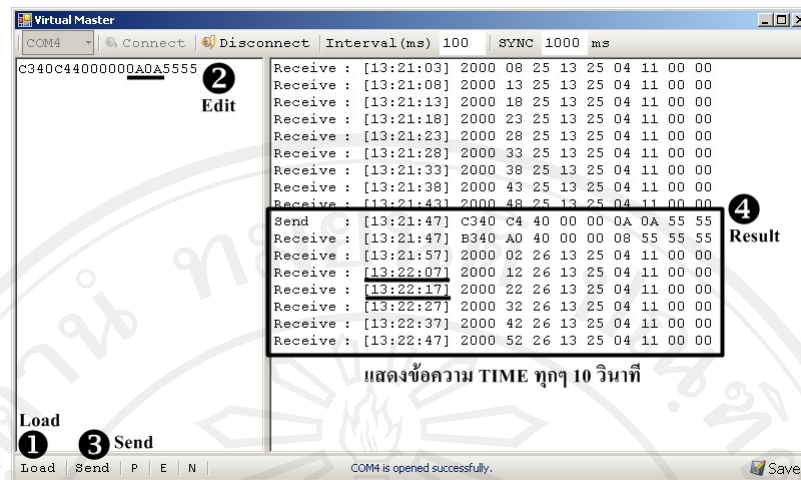
โหนด Locator มีรายชื่อบริการ SDO และ PDO ดังตาราง 4.2

ตาราง 4.2 รายชื่อบริการสื่อสารของโหนด Locator

ชื่อบริการ	ชนิด	คำอธิบาย
SDO_WrIntTAB	D/L	สำหรับเขียนข้อมูลลงในพื้นที่โปรแกรมของไมโครคอนโทรลเลอร์ตรงตำแหน่งที่กำหนด
SDO_RdIntTAB	U/L	สำหรับอ่านข้อมูลจากพื้นที่โปรแกรมของไมโครคอนโทรลเลอร์ตรงตำแหน่งที่กำหนด
SDO_WrParam	D/L	สำหรับเขียนข้อมูลลงในหน่วยความจำของไอซีรักษาฐานเวลาจริง (DS1307) ตรงตำแหน่งที่กำหนดเช่น การตั้งค่าวันและเวลา เป็นต้น
SDO_RdParam	U/L	สำหรับอ่านข้อมูลจากหน่วยความจำของไอซีรักษาฐานเวลาจริง (DS1307) ตรงตำแหน่งที่กำหนดเช่น การอ่านค่าวันและเวลา เป็นต้น
SDO_Setting	D/L	สำหรับแก้ไขคาบเวลาสร้างข้อความตำแหน่งและข้อความ TIME
PDO_AlarmCLKSet	IN	สำหรับกำหนดเวลาในการแจ้งเตือน
PDO_AlarmCLK	OUT	ข้อความแจ้งเตือนเมื่อครบกำหนดเวลา
PDO_LatLong	OUT	ข้อความตำแหน่ง
TIME	OUT	ข้อความฐานเวลา

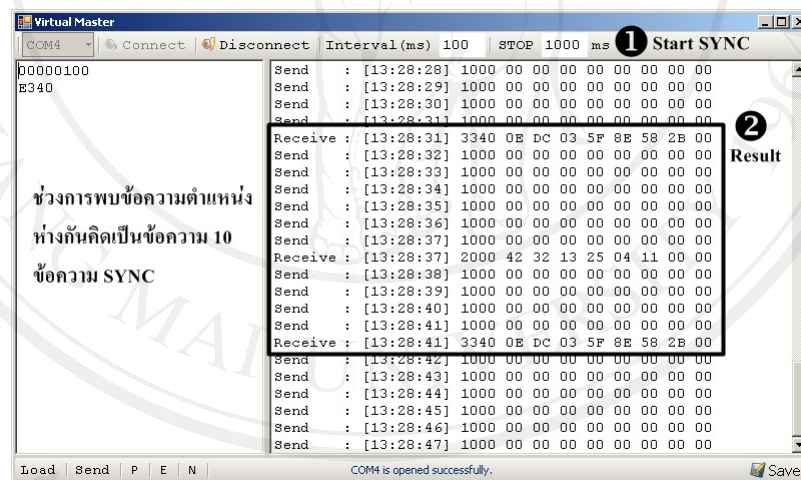
ผลการทดสอบโหนด Locator บางส่วนได้ถูกกล่าวถึงในหัวข้อ 4.2.2 แล้ว เนื้อหาส่วนนี้จึงขอก้าวเพิ่มเติมถึงการใช้บริการ SDO_Setting ทดลองแก้ไขคาบเวลาในการสร้างข้อความตำแหน่งและข้อความ TIME โดยทดลองเปลี่ยนคาบเวลาให้โหนด Locator สร้างข้อความ TIME เป็นทุกๆ 10 วินาทีและสร้างข้อความตำแหน่งทุกๆ 10 ข้อความ SYNC มีขั้นตอนการทดสอบดังนี้

- 1) ทำตามขั้นตอน 1 ถึง 3 ในหัวข้อ 4.2.2 ให้ปรากฏผลลัพธ์ดังรูป 4.13
- 2) โหลดเพิ่มข้อความสำหรับบริการ SDO_Setting ที่เตรียมไว้แล้วแก้ไขรายละเอียดให้ได้ดังรูป 4.25 ตรงหมายเลข 2 กดปุ่ม Send จากนั้นเมื่อการทำงานของบริการ SDO_Setting เสร็จสิ้นควรปรากฏข้อความ TIME ตามคาบเวลาที่ตรงหมายเลข 4



รูป 4.25 ภาพซอฟต์แวร์ Virtual Master เมื่อทดสอบบริการ SDO_Setting

3) เปลี่ยนสถานะงานของโหนด Locator เป็น Operational จากนั้นเริ่มส่งข้อความ SYNC ดังรูป 4.22 ตรงหมายเลข 1 เพื่อดูผลการสร้างข้อความตำแหน่ง แสดงผลลัพธ์ดังรูปตรงตำแหน่ง หมายเลข 2



รูป 4.26 ภาพซอฟต์แวร์ Virtual Master หลังจากเปลี่ยนคาบเวลาของข้อความตำแหน่ง

4.2.5 การทดสอบการทำงานของโหนด Online

โหนด Online มีรายชื่อบริการ SDO และ PDO ดังตาราง 4.3

ตาราง 4.3 รายชื่อบริการสื่อสารของโหนด Online

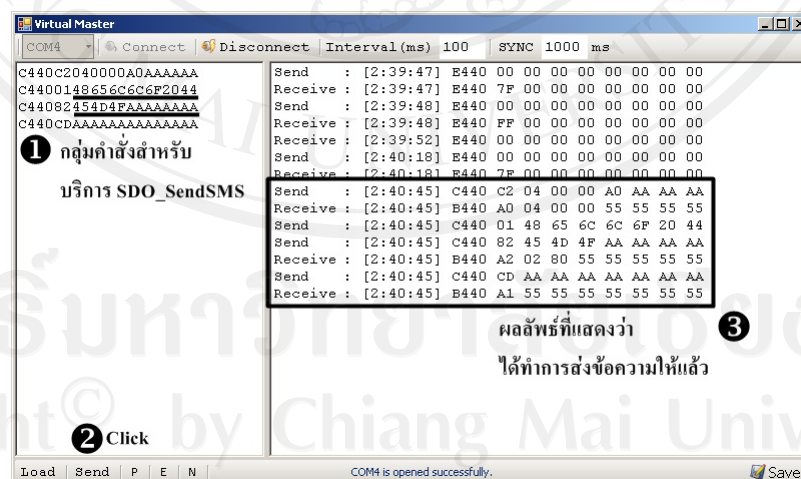
ชื่อบริการ	ชนิด	คำอธิบาย
SDO_WrIntTAB	D/L	สำหรับเขียนข้อมูลลงในพื้นที่โปรแกรมของไมโครคอนโทรลเลอร์ตรงตำแหน่งที่กำหนด

ตาราง 4.3 รายชื่อบริการสื่อสารของโหนด Online (ต่อ)

ชื่อบริการ	ชนิด	คำอธิบาย
SDO_RdIntTAB	U/L	สำหรับอ่านข้อมูลจากพื้นที่โปรแกรมของไมโครคอนโทรลเลอร์ตรงตำแหน่งที่กำหนด
SDO_TestLink	D/L	สำหรับใช้ทดสอบการเชื่อมต่อกับ URL ที่ระบุไว้ (ค่าเริ่มต้น)
SDO_Report	D/L	สำหรับใช้รายงานข้อมูลที่ต้องการไปยัง URL ที่เชื่อมต่อ
SDO_SendSMS	D/L	สำหรับใช้ส่งข้อมูลในรูปแบบข่าวสารสั้น (SMS) ไปยังหมายเลขโทรศัพท์ที่ระบุไว้
SDO_GetResult	U/L	สำหรับอ่านค่าผลตอบกลับจาก URL หรือหมายเลขโทรศัพท์ที่เชื่อมต่อ

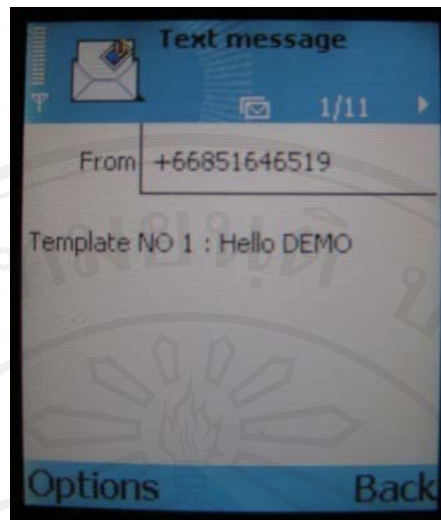
ได้ทำการทดสอบบริการ SDO_SendSMS โดยทดลองส่งข้อความ “Hello DEMO” ที่มีส่วนหัวเป็นข้อความ “Template NO 1 : “ ไปยังหมายเลขโทรศัพท์ปลายทางซึ่งได้บันทึกไว้ในโหนด Online ตั้งแต่ต้น มีขั้นตอนการทดสอบดังนี้

- 1) ติดตั้งโหนด Master และโหนด Online ให้ทำงานร่วมกันและต่อสายยูเอสบีพอร์ตจากโหนด Master เข้ากับคอมพิวเตอร์ส่วนบุคคล จากนั้นเริ่มการทำงานซอฟต์แวร์ Virtual Master
- 2) โหลดแฟ้มกลุ่มคำสั่งสำหรับบริการ SDO_SendSMS ปรากฏคำสั่งดังรูป 4.27 ตรงหมายเลข 1 แล้วกดปุ่ม Send จากนั้นเมื่อการทำงานของบริการ SDO_SendSMS เสร็จสิ้นควรปรากฏข้อความดังรูปตรงหมายเลข 3



รูป 4.27 ภาพซอฟต์แวร์ Virtual Master เมื่อทดสอบบริการ SDO_SendSMS

- 3) ในรูป 4.28 แสดงผลลัพธ์ข้อความที่ทดลองส่งจากบริการ SDO_SendSMS ไปยังโทรศัพท์หมายเลขปลายทาง



รูป 4.28 ภาพผลลัพธ์ SMS จากโทรศัพท์หมายเลขปลายทาง

4.2.6 การทดสอบการทำงานของโหนด Logger

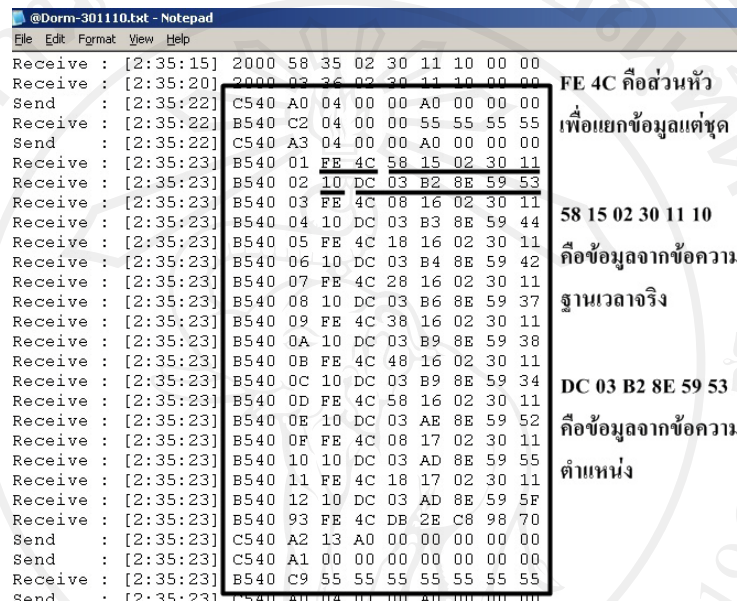
โหนด Logger มีรายชื่อบริการ SDO และ PDO ดังตาราง 4.4

ตาราง 4.4 รายชื่อบริการสื่อสารของโหนด Logger

ชื่อบริการ	ชนิด	คำอธิบาย
SDO_WrIntTAB	D/L	สำหรับเขียนข้อมูลลงในพื้นที่โปรแกรมของไมโครคอนโทรลเลอร์ตรงตำแหน่งที่กำหนด
SDO_RdIntTAB	U/L	สำหรับอ่านข้อมูลจากพื้นที่โปรแกรมของไมโครคอนโทรลเลอร์ตรงตำแหน่งที่กำหนด
SDO_WrIntRAM	D/L	สำหรับเขียนข้อมูลลงในพื้นที่หน่วยความจำที่เตรียมไว้ ก่อนการใช้ข้อความ SDO_Store เพื่อบันทึกข้อมูลดังกล่าวลงในการ์ด MMC
SDO_RdIntRAM	U/L	สำหรับอ่านข้อมูลจากพื้นที่หน่วยความจำที่เตรียมไว้ หลังจากใช้ข้อความ SDO_Load อ่านข้อมูลจากการ์ด MMC มาเก็บไว้ในพื้นที่หน่วยความจำที่เตรียมไว้
SDO_SetLastMMCAddr	D/L	สำหรับใช้กำหนดตำแหน่งล่าสุดที่บันทึกข้อมูลในการ์ด MMC
SDO_GetLastMMCAddr	U/L	สำหรับใช้อ่านตำแหน่งล่าสุดที่บันทึกข้อมูลในการ์ด MMC
SDO_Store	D/L	ใช้บันทึกข้อมูลจากพื้นที่หน่วยความจำที่เตรียมไว้ไปยังการ์ด MMC
SDO_Load	D/L	ใช้สำหรับอ่านข้อมูลจากการ์ด MMC ไปยังพื้นที่หน่วยความจำที่เตรียมไว้

การทดสอบการบันทึกข้อมูลได้กำหนดให้ข้อความตำแหน่งเป็นข้อความหลัก เมื่อโหนด Logger ได้รับข้อความตำแหน่ง กระบวนการสร้างแฟ้มเก็บข้อมูลสำหรับบันทึกจึงเริ่มขึ้น โดยหนึ่ง

แพ็กเก็ตข้อมูลประกอบด้วยส่วนหัว ข้อความตำแหน่งและข้อความฐานเวลาจริง รวมถึงข้อความอื่นๆ ที่ต้องการ จากนั้นแพ็กเก็ตดังกล่าวถูกพักไว้ที่พื้นที่หน่วยความจำที่เตรียมไว้เพื่อรอให้ปริมาณข้อมูลครบตามจำนวนที่ใช้ในการบันทึกข้อมูลลงการ์ด MMC แต่ละครั้ง และเมื่อได้ครบตามจำนวนข้อมูลดังกล่าวถูกบันทึกอัตโนมัติ ในรูป 4.29 แสดงตัวอย่างแพ็กเก็ตข้อมูลที่บันทึกลงการ์ด MMC



รูป 4.29 ตัวอย่างแฟ้มข้อความที่บันทึกด้วยซอฟต์แวร์ Virtual Master เมื่อมีการอ่านข้อมูลจากโหนด Logger

4.2.7 การทดสอบการทำงานของโหนด HMI

โหนด HMI มีรายชื่อบริการ SDO และ PDO ดังตาราง 4.5

ตาราง 4.5 รายชื่อบริการสื่อสารของโหนด HMI

ชื่อบริการ	ชนิด	คำอธิบาย
SDO_WrIntTAB	D/L	สำหรับเขียนข้อมูลลงในพื้นที่โปรแกรมของไมโครคอนโทรลเลอร์ตรงตำแหน่งที่กำหนด
SDO_RdIntTAB	U/L	สำหรับอ่านข้อมูลจากพื้นที่โปรแกรมของไมโครคอนโทรลเลอร์ตรงตำแหน่งที่กำหนด

ในรูป 4.30 คือการทดสอบการแสดงผลซึ่งข้อมูลที่นำมาแสดงได้จากข้อความตำแหน่งและข้อความฐานเวลาจริง ส่วนในรูป 4.31 คือการแสดงผลข้อความตำแหน่ง ข้อความฐานเวลาจริง และข้อมูลสถานะของรถยนต์บางส่วน



รูป 4.30 ภาพการทดสอบโหนด HMI โดยให้แสดงข้อมูลจากข้อความตำแหน่ง
และข้อความฐานเวลาจริง



รูป 4.31 ภาพโหนด HMI ขณะแสดงผลข้อความตำแหน่ง ข้อความฐานเวลาจริงและข้อมูลสถานะ
ของรถยนต์บางส่วน

4.3 การสร้างและทดสอบระบบ OMD Server

เริ่มจากการสร้างเว็บเพจทั้งส่วนที่ใช้แสดงผลและส่วนที่ใช้ประมวลผลด้วยภาษา PHP ร่วมกับโปรแกรมฐานข้อมูล MySQL โดยใช้คอมพิวเตอร์เซิร์ฟเวอร์ของภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเชียงใหม่ เป็นที่ติดตั้งเว็บเพจดังกล่าว มี URL ดังตารางที่ 4.6

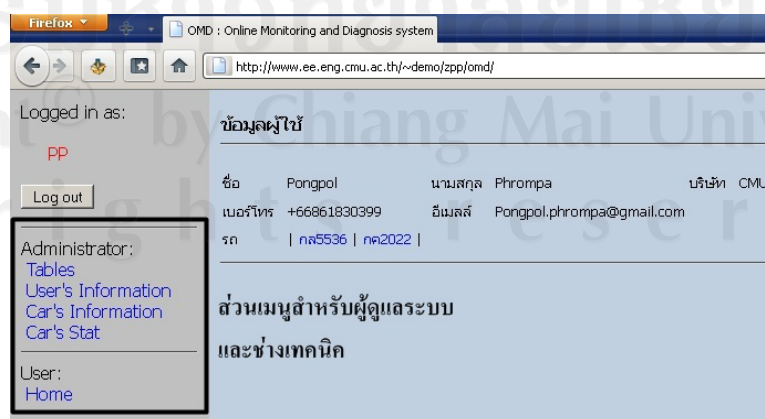
ตาราง 4.6 URL สำหรับเรียกใช้แต่ละเว็บเพจ

ประเภท URL	URL
เว็บเพจหน้าแรก	http://www.ee.eng.cmu.ac.th/~demo/zpp/omd
เว็บเพจสำหรับประมวลผลข้อมูลจาก OMD Client	http://www.ee.eng.cmu.ac.th/~demo/zpp/omd/uc/cstat.php
ทดสอบการประมวลผลข้อมูล	http://www.ee.eng.cmu.ac.th/~demo/zpp/omd/uc/cstat.php?id=1&pt=DBE7268E455C&st=05-00,0C-0000,0D-00,21-0000,31-0000,42-0000,&dtc=02,0102,0107,&cts=004214290311

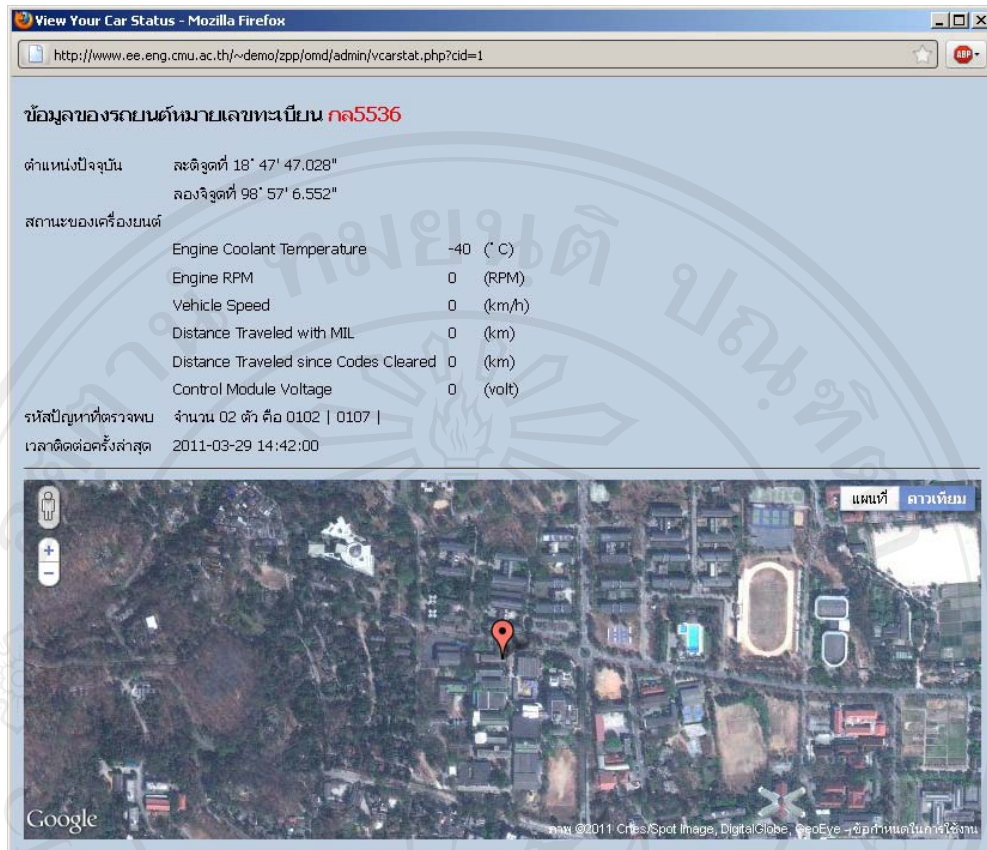
การทดสอบใช้งานระบบ OMD Server สามารถทำได้ทันทีโดยใช้เว็บเบราว์เซอร์ที่ติดตั้งมากับคอมพิวเตอร์ส่วนบุคคลเรียกใช้งาน URL สำหรับทดสอบการประมวลผลข้อมูลดังแสดงในตาราง 4.7 รูป 4.32 แสดงภาพเว็บเพจหน้าแรกของระบบ OMD Server และรูป 4.33 แสดงภาพเว็บเพจเมื่อมีผู้ล็อกอินเข้าใช้งานจากเว็บเพจหน้าแรก โดยมีการแสดงผลข้อมูลที่ส่งเข้าไปทดสอบดังรูป 4.34



รูป 4.32 ภาพเว็บเพจหน้าแรกของระบบ OMD Server



รูป 4.33 ภาพเว็บเพจหน้าข้อมูลผู้ล็อกอินใช้งาน



รูป 4.34 ภาพเว็บเพจแสดงข้อมูลสถานะของรถยนต์ตามข้อมูลที่ใช้ทดสอบ

4.4 การทดสอบการทำงานของระบบ OMD

รูป 4.35 ถึงรูป 4.38 แสดงภาพการติดตั้งระบบ OMD Client ในรถยนต์ทดสอบยี่ห้อ Honda รุ่น City 3 ซึ่งระบบ OMD Client มีขั้นตอนการทำงานภาพรวมดังนี้

1) เริ่มจากผู้ใช้บริการดูแลไปที่สถานะ “ON” หรือผู้ใช้สตาร์ทรถยนต์ ระบบ OMD Client ได้รับการจ่ายแรงดันไฟฟ้าจึงเริ่มต้นทำงาน แต่ละโหนดตั้งค่าเริ่มต้นทำงานของโหนดและอยู่ในสถานะงาน preoperational

2) โหนด Master หน่วงเวลาทำงานของระบบไว้ประมาณ 20 วินาที เมื่อครบกำหนดเวลา โหนด Master เปลี่ยนสถานะงานของทุกโหนดเป็น operational ทำให้โหนดสเลฟทุกโหนดเริ่มทำงานตามที่ได้ตั้งค่าการทำงานเอาไว้

3) โหนดสเลฟวนทำงานตามคาบกำหนดเวลาที่ได้โปรแกรมเอาไว้ โดยใช้จังหวะการทำงานโหนด Master ให้ข้อความ SYNC เพื่อกำหนดจังหวะการทำงานของโหนดสเลฟ

สามารถสรุปค่าตั้งต้นทำงานของแต่ละโหนดที่ใช้ดังนี้

- โหนด Master มีคาบเวลาในการสร้างข้อความ SYNC ทุกๆ 100 มิลลิวินาที และสร้างข้อความร้องขอ PDO_CarDTCs ทุกๆ 24 วินาที

- โหนด Gateway มีกำหนดเวลาสร้างข้อความสถานะของรถยนต์ PDO_CarStatus ทุกๆ 2 ข้อความ SYNC หรือคิดเป็น 200 มิลลิวินาที โดยเบื้องต้นได้กำหนดจำนวน PID ที่อ่านค่าสถานะจากรยนต์ไว้ 7 สถานะดังนี้

Engine Coolant Temperature

Engine RPM

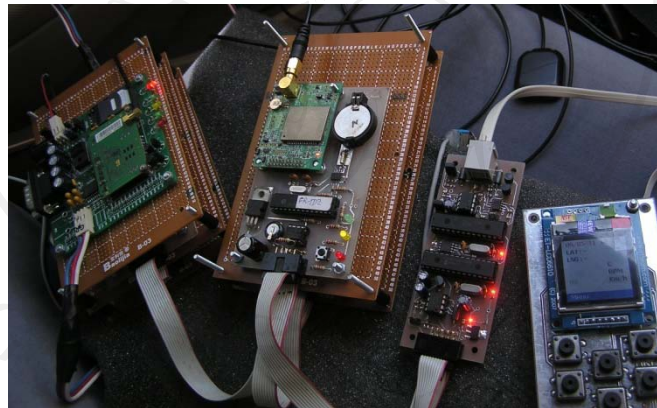
Vehicle Speed

Distance Traveled with MIL

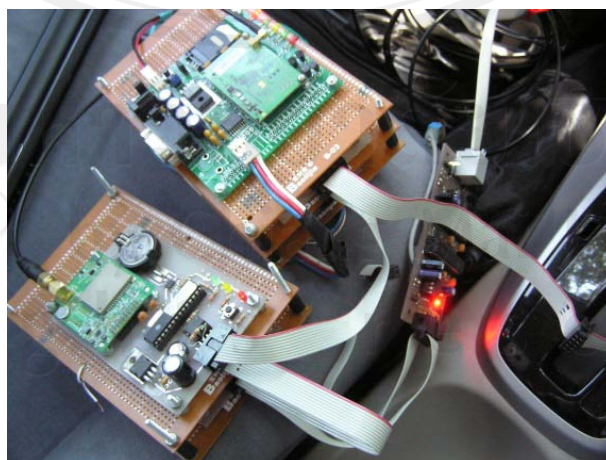
Distance Traveled since Codes Cleared

Control Module Voltage

- โหนด Locator สร้างข้อความฐานเวลาและข้อความตำแหน่งทุกๆ 5 วินาที
- โหนด Online รายงานแพ็กเก็ตข้อมูลบางส่วนไปยัง OMD Server ทุกๆ 24 วินาที



รูป 4.35 ภาพทดสอบการทำงานของระบบ OMD Client ในรถยนต์#1



รูป 4.36 ภาพทดสอบการทำงานของระบบ OMD Client ในรถยนต์#2

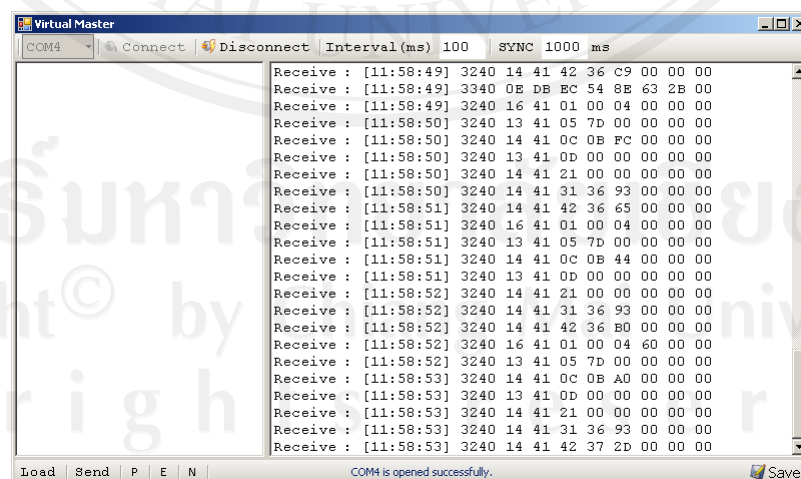


รูป 4.37 ภาพทดสอบการทำงานของระบบ OMD Client ในรถยนต์#3

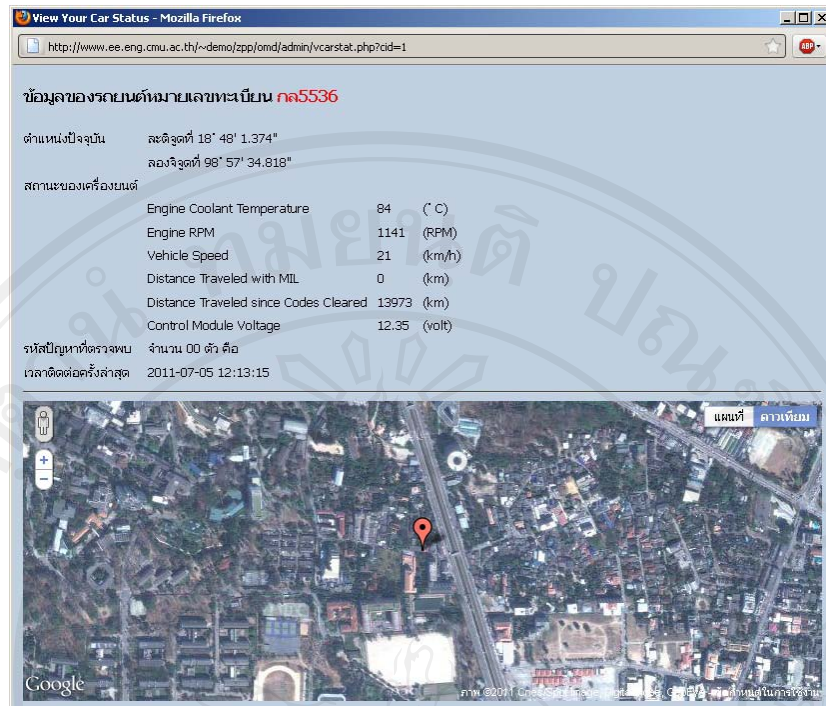


รูป 4.38 ภาพทดสอบการทำงานของระบบ OMD Client ในรถยนต์#4

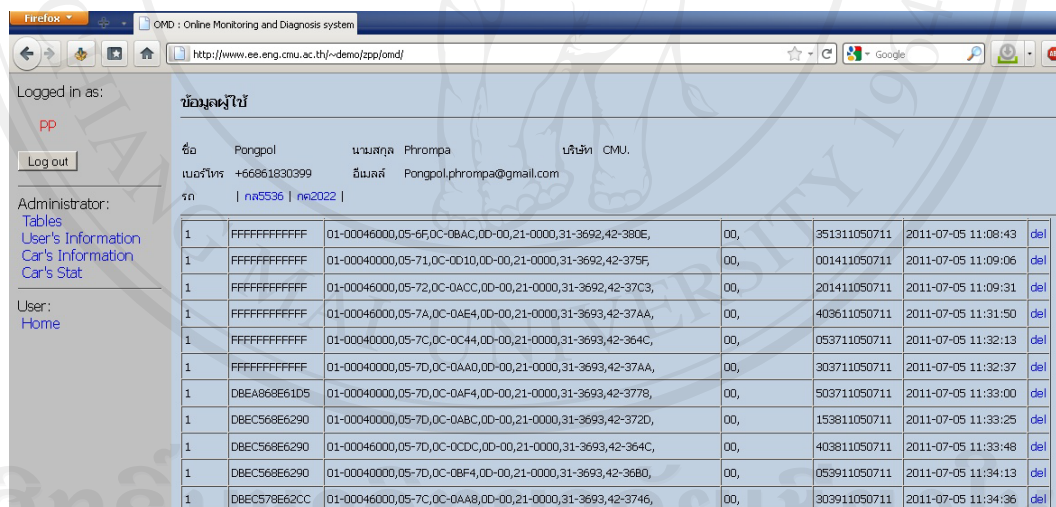
รูป 4.39 แสดงผลภาพซอฟต์แวร์ Virtual Master ที่ได้จากการเชื่อมต่อโหนด Master เข้ากับคอมพิวเตอร์ส่วนบุคคล ขณะที่ระบบ OMD กำลังทำงานในรถยนต์ ส่วนรูป 4.40 คือหน้าเว็บเพจแสดงผลข้อมูลล่าสุดที่ OMD Server ได้รับจาก OMD Client และรูป 4.41 แสดงข้อมูลที่บันทึกในฐานข้อมูลที่ OMD Server สามารถนำข้อมูลเหล่านี้ไปประยุกต์ใช้งานได้ต่อไป



รูป 4.39 ภาพซอฟต์แวร์ Virtual Master ขณะระบบ OMD ทำงาน



รูป 4.40 เว็บเพจแสดงผลข้อมูลล่าสุดของรถยนต์ที่ OMD Server ได้รับ



รูป 4.41 เว็บเพจแสดงฐานข้อมูลที่ OMD Server บันทึกไว้