



ภาคผนวก

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่

Copyright© by Chiang Mai University

All rights reserved

ภาคผนวก ก

สรุป AT command ที่ใช้ควบคุมการทำงานของมอดูล GR64 [20]

กลุ่มควบคุมการโทร (Call Control)

ATH	HANG UP
ATO	RETURN TO ONLINE DATA MODE

กลุ่มควบคุมและสถานะของมอดูล (Control and Status)

ATS2	ESCAPE SEQUENCE CHARACTER
AT+CFUN	SET PHONE FUNCTIONALITY
AT+E2RESET	RESTART WIRELESS CPU
AT&F	SET TO FACTORY DEFINED CONFIGURATION PROFILE
AT&W	STORE CONFIGURATION PROFILE

กลุ่มข้อมูล-จีพีอาร์เอส (Data – GPRS)

AT+CGACT	PDP CONTEXT ACTIVATE OR DEACTIVATE
AT+CGATT	GPRS ATTACH OR DETACH
AT+CGDCONT	DEFINE PDP CONTEXT

กลุ่มข้อมูล-ทีซีพี/ไอพี (Data – TCP/IP)

AT+E2IPA	IP ACTIVATE
AT+E2IPC	IP SOCKET CLOSE
AT+E2IPO	IP OPEN/CONNECT
AT+E2IPS	IP SETUP IP PARAMETERS
AT+E2IPATO	SOCKET ATO COMMAND

กลุ่มระบุตัวตนมอดูล (Identification)

AT	ATTENTION COMMAND
----	-------------------

กลุ่มส่วนติดต่ออนุกรม (Serial Interface)

AT+IPR	CABLE INTERFACE PORT BAUD RATE
ATE	COMMAND ECHO

**ตัวอย่างการใช้มอดูล GR64 ร้องขอข้อมูลจากเว็บเซิร์ฟเวอร์
ผ่านโปรโตคอล HTTP ด้วยวิธี GET**

1. ทดสอบการติดต่อกับมอดูล

คำสั่ง AT<CR>

ผลตอบรับ OK<CR><LF>

2. ตั้งค่าอัตราบอดในการสื่อสารกับมอดูลเท่ากับ 9,600 บิตต่อวินาที

คำสั่ง AT+IPR=9600<CR>

ผลตอบรับ OK<CR><LF>

3. ตั้งค่า PDP context หรือ access point name ตามโอเปอเรเตอร์ที่ให้บริการ SIM

คำสั่ง AT+CGDCONT=1,"IP","www.dtac.co.th"<CR>

ผลตอบรับ OK<CR><LF>

4. Activate IP กับ PDP context ที่ตั้งไว้

คำสั่ง AT+E2IPA=1,1<CR>

ผลตอบรับ *E2IPA:000<CR><LF>OK<CR><LF>

5. เปิดใช้งาน Escape sequence (+++AT)

คำสั่ง AT+E2IPS=2,8,2,1020,1,64<CR>

ผลตอบรับ OK<CR><LF>

6. ดูหมายเลข IP ของมอดูล ที่ได้จากขั้นตอน 4

คำสั่ง AT+E2IPI=0<CR>

ผลตอบรับ *E2IPI: "10.122.124.119"<CR><LF>OK<CR><LF>

เลขตัวอย่าง

7. ดูหมายเลข Gateway ของเครือข่ายที่ให้บริการ

คำสั่ง AT+E2IPI=1<CR>

ผลตอบรับ *E2IPI: "192.168.165.1"<CR><LF>OK<CR><LF>

เลขตัวอย่าง

8. ดูหมายเลข DNS ของเครือข่ายที่ให้บริการ

คำสั่ง AT+E2IPI=2<CR>

ผลตอบรับ *E2IPI: "203.155.33.2"<CR><LF>OK<CR><LF>

เลขตัวอย่าง

9. เปิด Socket ที่ใช้เป็นช่องทางการติดต่อ

คำสั่ง AT+E2IPO=1,"161.200.85.47",80<CR><LF>

Socket: 80, การติดต่อด้วย

ผลตอบรับ CONNECT

HTTP

เข้าสู่โหมด online data ทุก
ข้อมูลที่มาดูลได้รับคือข้อมูล
สำหรับโปรโตคอล HTTP

10. เริ่มส่งคำสั่งและข้อมูลตามข้อกำหนดของโปรโตคอล HTTP

คำสั่ง GET /pp.php?g=80<Enter>

ตัวอย่างการใช้วิธี GET ของโปร
โตคอล HTTP

ผลตอบรับ Hello P
NO CARRIER

เมื่อรับ-ส่งข้อมูลเสร็จจัดการ
ติดต่อทันที

11. การ Deactivate IP จาก PDP context

คำสั่ง AT+E2IPA=0,1<CR>

ผลตอบรับ *E2IPA:000<CR><LF>OK<CR><LF>

12. การปิดการเชื่อมต่อกับบริการ GPRS

คำสั่ง AT+CGATT=0<CR>

ผลตอบรับ +CGATT: 0<CR><LF>OK<CR><LF>

13. การรีเซ็ตการทำงานของโมดูล GR64

คำสั่ง AT+E2RESET<CR>

ผลตอบรับ OK<CR><LF>

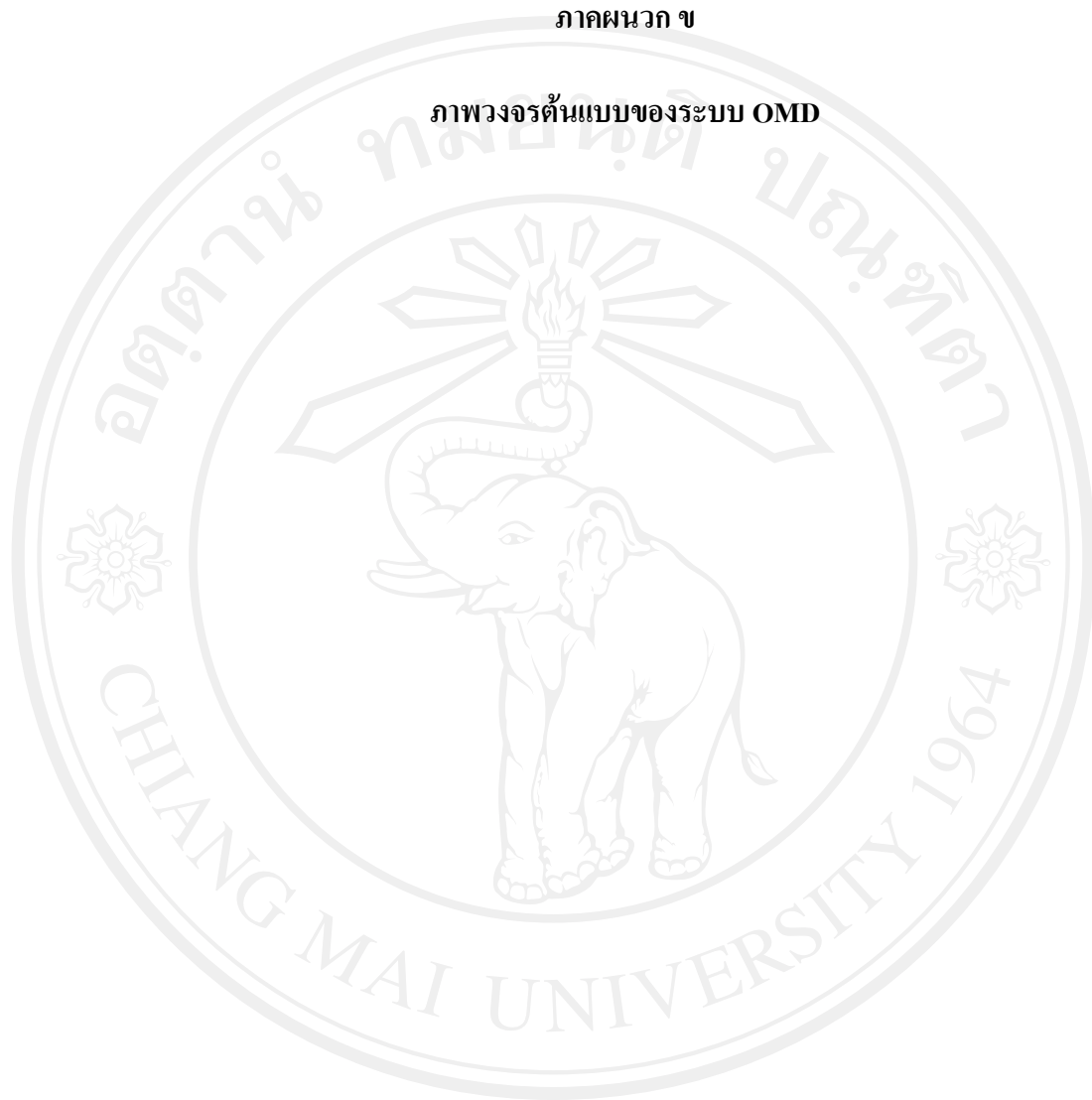
14. การปิดการทำงานของโมดูล GR64

คำสั่ง AT+CFUN=0<CR>

ผลตอบรับ OK<CR><LF>

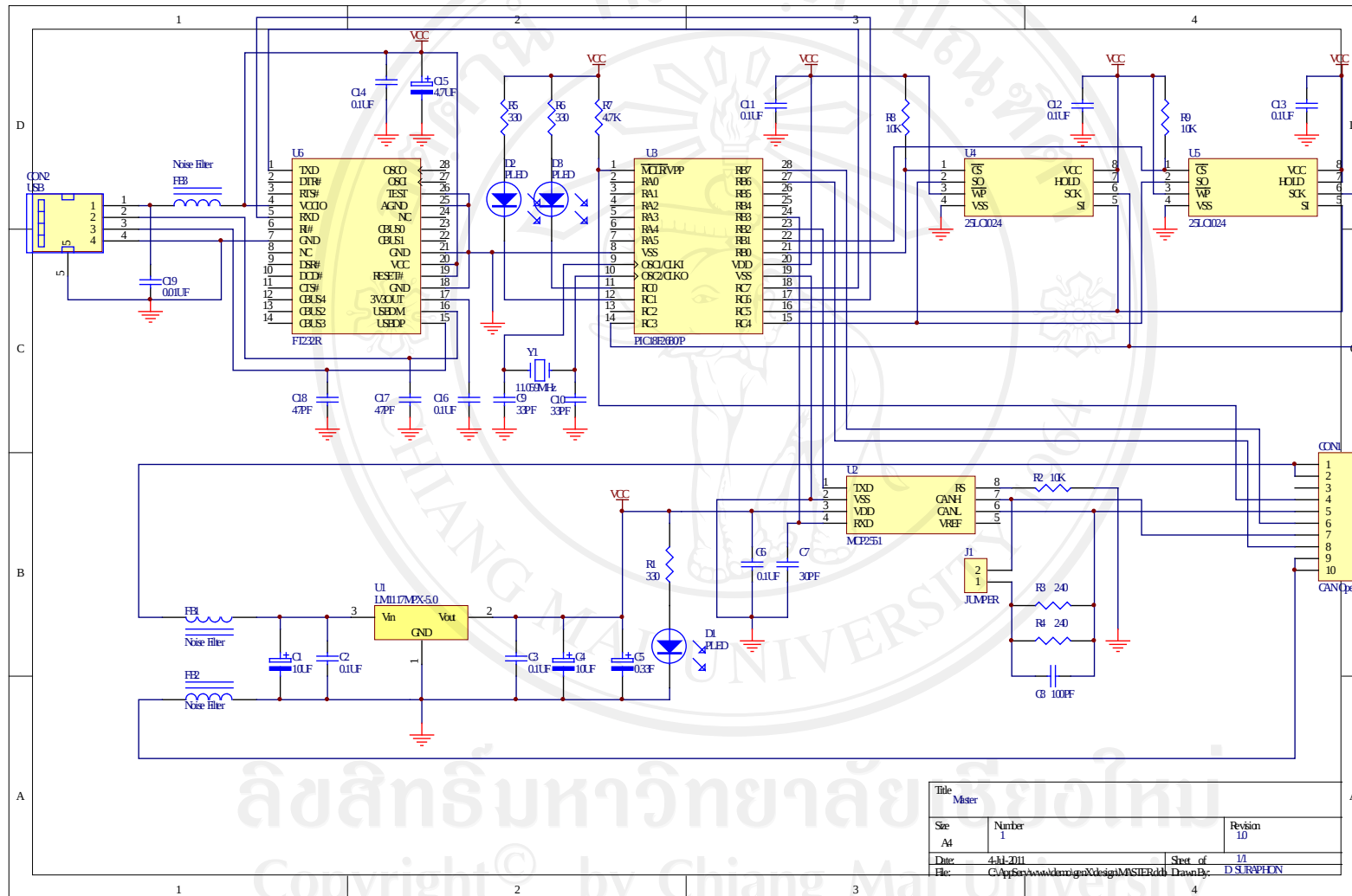
ภาคผนวก ข

ภาพวงจรต้นแบบของระบบ OMD

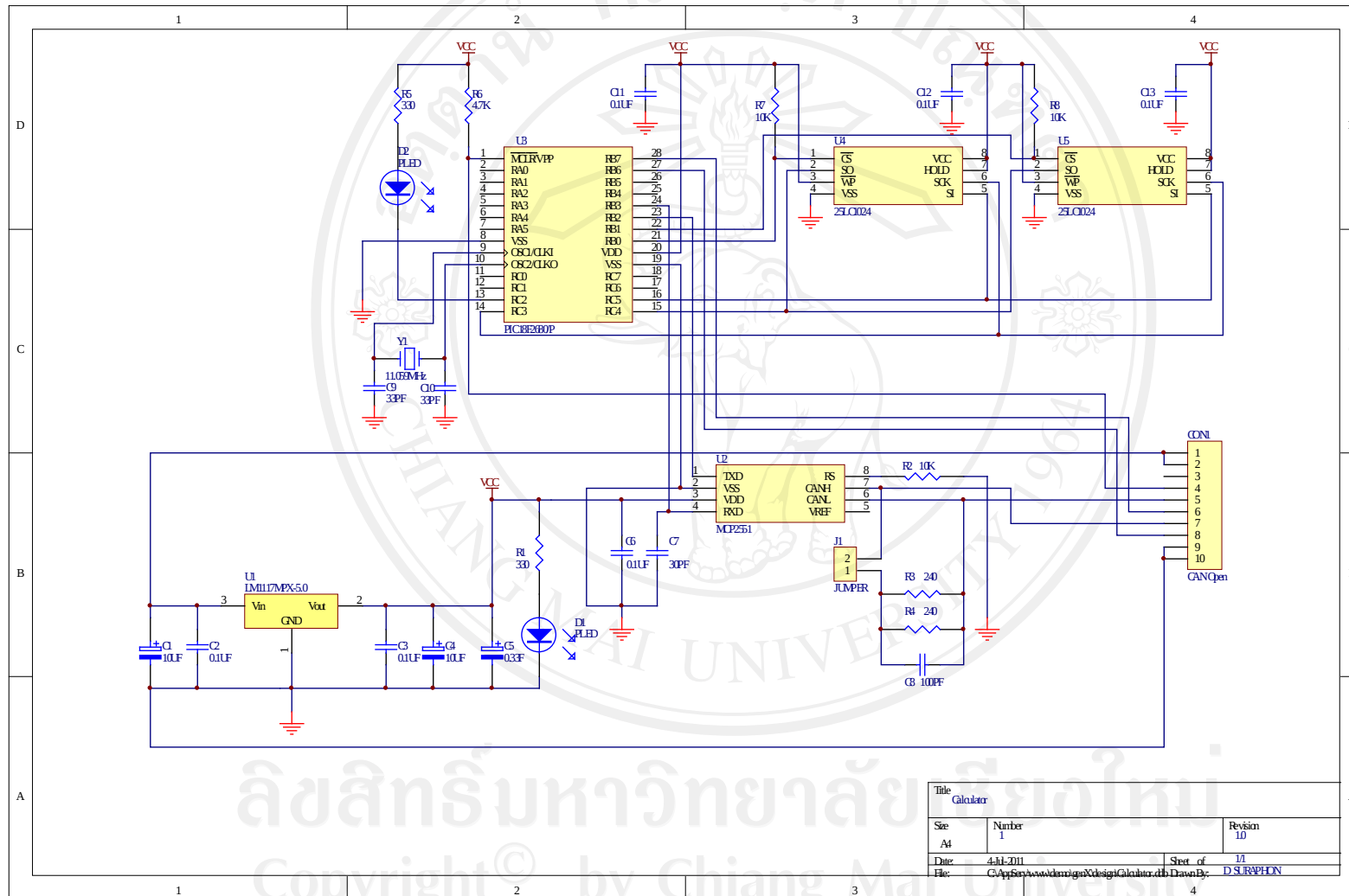


ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่

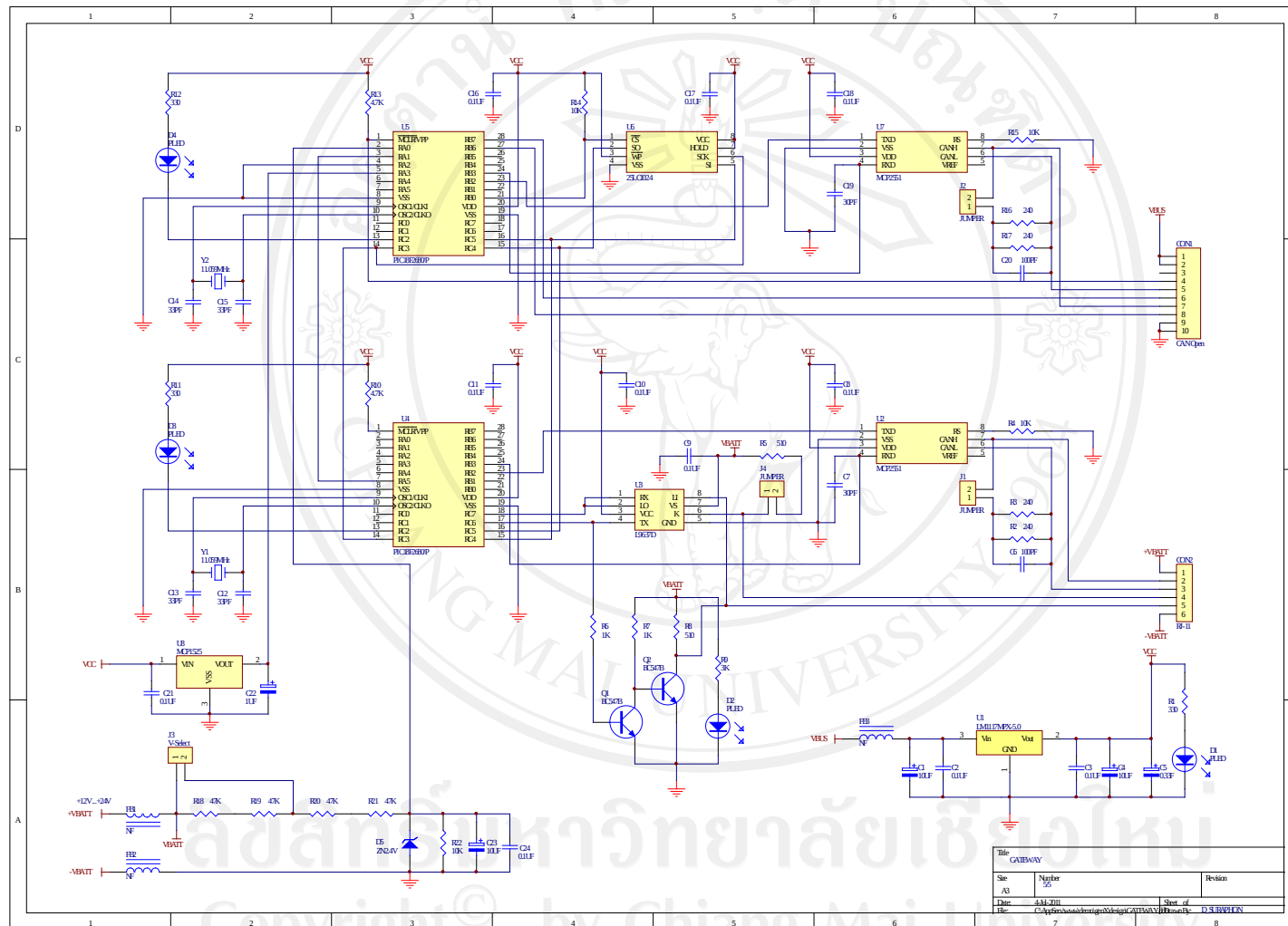
Copyright© by Chiang Mai University
All rights reserved



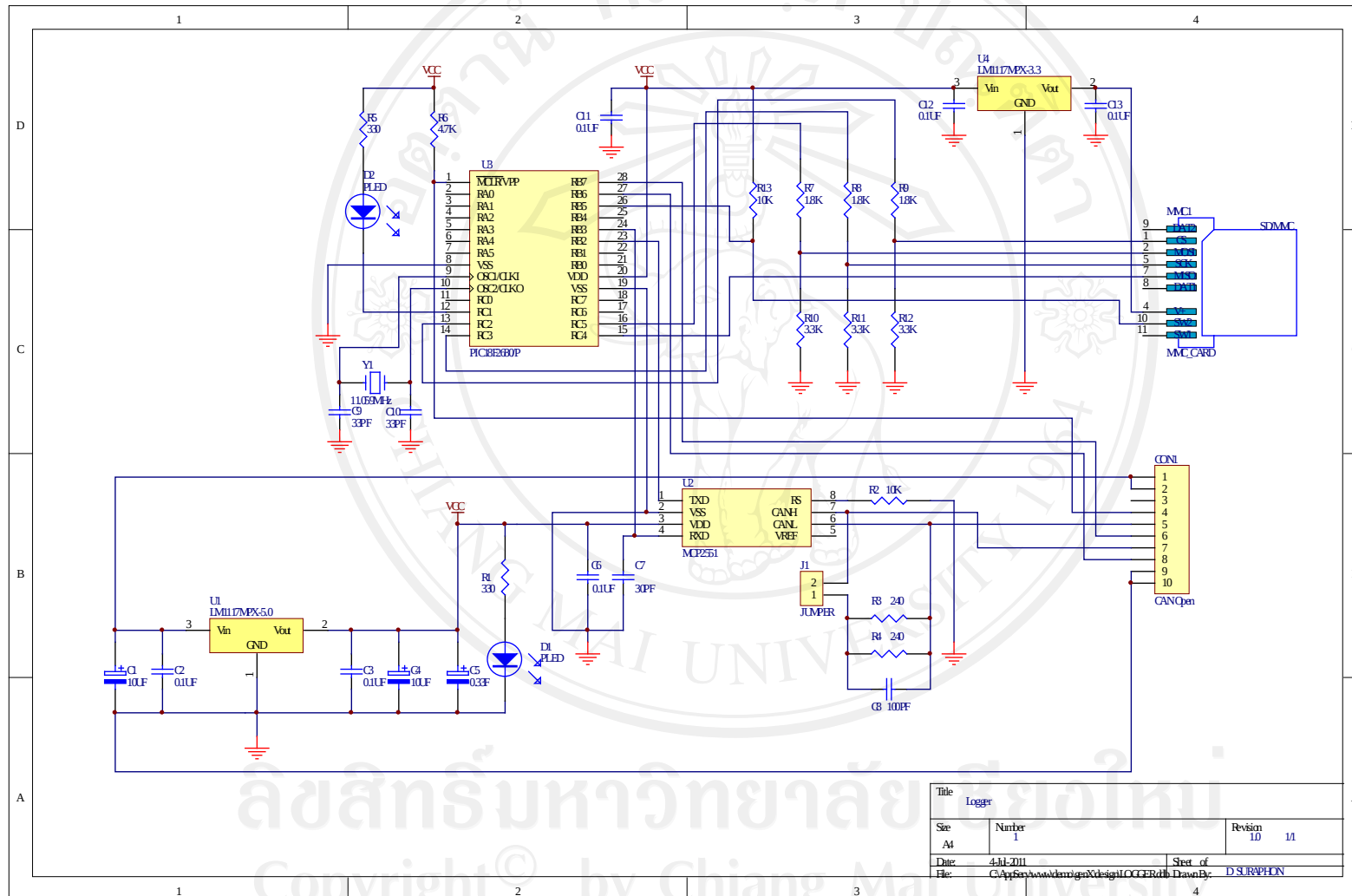
รูป ผ.1 ฟังวงจรโหนด Master



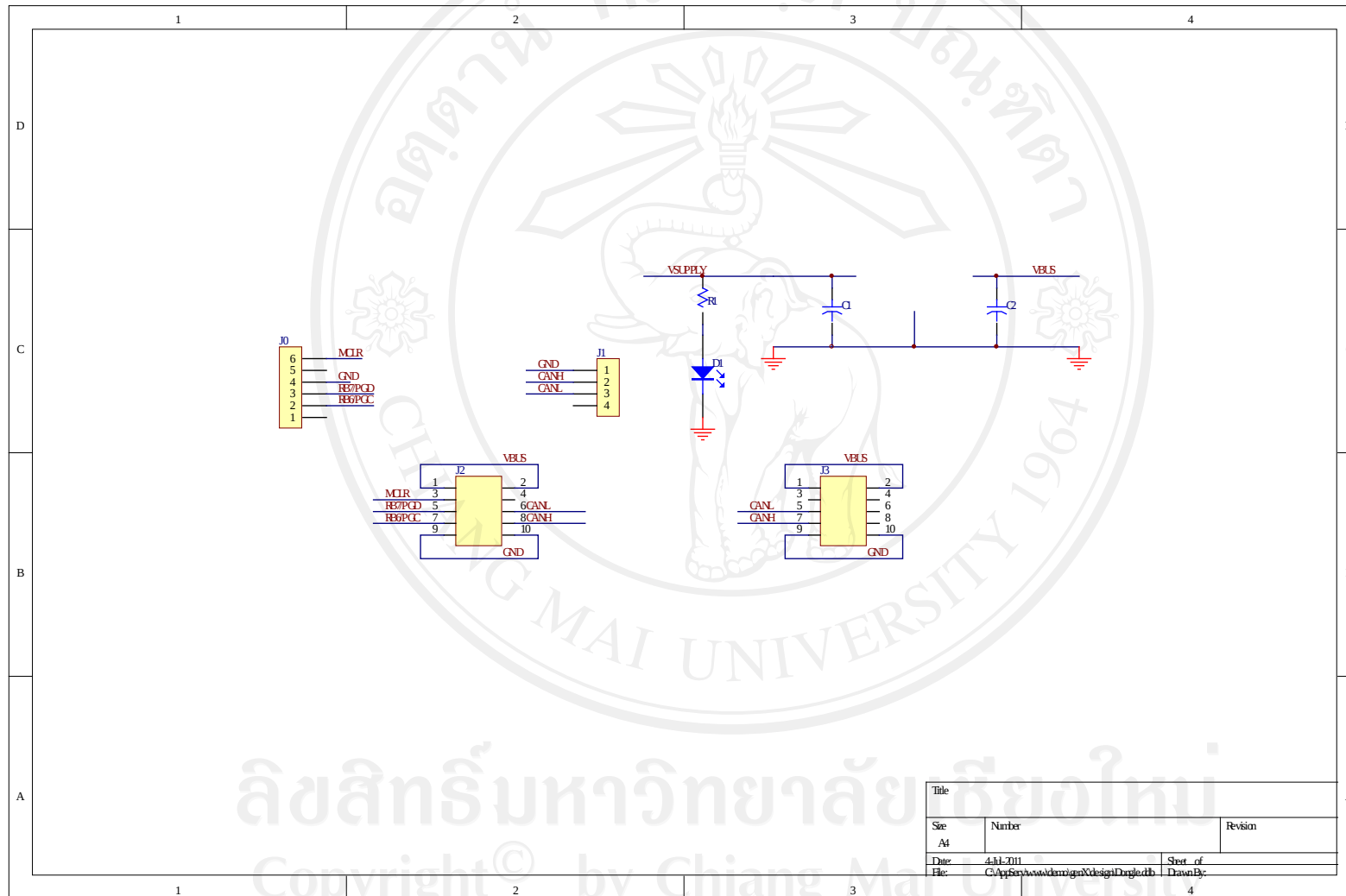
รูป ผ.2 ฟังวงจรไหนด Calculator



รูป ผ.3 แผงจอร์โนด Gateway



รูป ผ.5 ฟังวงจรไหนด Logger



รูป ผ.7 แผงวงจรต้นกำเนิดแรงดันไฟฟ้า

ภาคผนวก ค

ผลงานวิจัยที่ได้รับการตีพิมพ์

การประชุมวิชาการทางวิศวกรรมไฟฟ้าครั้งที่ 33 (EECON-33)

วันที่ 1-3 ธันวาคม พ.ศ. 2553 จังหวัดเชียงใหม่

หมวดงานวิจัยที่เกี่ยวข้องกับวิศวกรรมไฟฟ้า



ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่

Copyright© by Chiang Mai University

All rights reserved

การพัฒนาการติดตามและควบคุมโหนดอุปกรณ์ CAN โดยใช้ระบบฝังตัวแบบกระจาย

Development of Monitoring and Control System Using Distributed Embedded System Via CAN Bus

ปองพล พรหมปา¹ และ นิภาพร ศรีพล²

สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

239 ถนนห้วยแก้ว ตำบลสุเทพ อำเภอเมือง เชียงใหม่ 50200

¹ โทรศัพท์ : 0-5394-4140 E-mail: pongpol.phrompa@gmail.com

² โทรศัพท์ : 0-5394-4140 E-mail: nipapon@ee.eng.cmu.ac.th

บทคัดย่อ

บทความนี้นำเสนอระบบติดตามและควบคุมเครือข่ายโหนดอุปกรณ์ โดยใช้ระบบฝังตัวแบบกระจายเป็นแนวทางในการออกแบบและเชื่อมต่อโหนดด้วยบัส CAN ซึ่งแต่ละโหนดใช้ไมโครคอนโทรลเลอร์ขนาด 8 บิต ที่มีส่วนควบคุม CAN ภายในตัวเพื่อเป็นตัวควบคุมโหนด มีการพัฒนาโพรโทคอลชั้นสูงกว่า CAN เพื่อทำหน้าที่จัดลำดับการสื่อสารข้อมูลระหว่างโหนดด้วยอัตราบิต 500 กิโลบิตต่อวินาที และควบคุมฟังก์ชันงานของแต่ละโหนด หลังจากทดสอบด้วยระบบติดตามตำแหน่งรถยนต์แบบออนไลน์ พบว่าสามารถรายงานตำแหน่งล่าสุดของรถยนต์ทุกๆ คาบเวลา 15-30 วินาทีผ่านทางโครงข่าย GPRS นอกจากนี้สามารถใช้โหนดอุปกรณ์ต่อร่วมกับเซนเซอร์ หรืออุปกรณ์มาตรฐาน เช่น เซนเซอร์อุณหภูมิ ความชื้น และความดัน เพื่อประยุกต์ใช้เป็นโหนดสำหรับวิเคราะห์สภาพอากาศ

คำสำคัญ: CAN, โพรโทคอลชั้นสูงกว่า, ระบบฝังตัวแบบกระจาย

Abstract

This paper presents the node monitoring and control system using distributed embedded system and CAN bus. Each node adopts 8-bit microcontroller with built-in CAN controller to function as a node's controller. The CAN higher layer protocol is developed to communicate the data among nodes with CAN's baud rate of 500 kbps and to handle node's applications. The proposed system was tested by the on-line car's position monitoring system. It is found that the last car's position is reported in every period of 15-30 s via GPRS network. Furthermore, more applications can be created by connecting the devices with the node – such as temperature, humidity and pressure sensor to be the weather analyzer node.

Keywords: CAN, Higher Layer Protocol, Distributed Embedded System.

1. คำนำ

ด้วยคุณสมบัติเด่นของโพรโทคอล CAN [1] เช่น ความทนทานต่อสถานะแทรกสอดของคลื่นแม่เหล็กไฟฟ้า การตอบสนองต่อเหตุการณ์ในบัสอย่างรวดเร็ว และมีส่วนควบคุมความผิดพลาดจากการสื่อสารข้อมูล เป็นต้น ทำให้มีการประยุกต์ใช้บัส CAN กับเครือข่ายอุปกรณ์ในหลายๆ ที่ได้อย่างแพร่หลาย ในการใช้งานจริงเพื่อให้อุปกรณ์จากผู้ผลิตหรือต่างคุณสมบัติสามารถทำงานร่วมกันได้ มีการพัฒนาโพรโทคอลชั้นสูงกว่า CAN เพื่อให้สามารถจัดการกับเครือข่ายอุปกรณ์ จัดลำดับการสื่อสาร และจัดการกับความผิดพลาดในการสื่อสารข้อมูล ทั้งนี้ก็พัฒนาสามารถพัฒนาชั้นโพรโทคอลชั้นสูงกว่า CAN เองหรือเลือกใช้โพรโทคอลจากนักพัฒนาหลายๆ กลุ่ม [2]

โพรโทคอล CANopen เป็นโพรโทคอลที่ไม่มีการใช้ภายในการนำไปใช้งาน ซึ่งนักพัฒนาสามารถเลือกใช้รูปแบบการสื่อสารให้เหมาะสมกับงานที่นำไปประยุกต์ คุณสมบัติเด่นที่ทำให้ CANopen แตกต่างจากโพรโทคอลอื่นคือ อุปกรณ์แต่ละชิ้นมีตารางข้อมูลที่เรียกว่า ดิคชันนารีวัตถุเป็นของตัวเองเพื่อใช้อธิบายชนิดและการทำงานของอุปกรณ์ นอกจากนี้ยังมีชุดข้อความสื่อสารซึ่งถูกใช้เพื่อแก้ไขตารางข้อมูลสื่อสารข้อมูลระหว่างโหนด รวมถึงควบคุมและกำหนดการทำงานของโหนดอุปกรณ์ในเครือข่าย [2] อย่างไรก็ตามโดยทั่วไปแล้วการประยุกต์ใช้งานโพรโทคอล CANopen ตามรูปแบบมาตรฐาน ต้องใช้ทรัพยากรจำนวนมาก โดยใช้โปรแกรม 40-60 กิโลไบต์ และใช้หน่วยความจำ 600 ไบต์ หรือมากกว่า [2-3] ทำให้ไม่เหมาะสมในการนำไปใช้งานร่วมกับตัวควบคุมโหนดอุปกรณ์ที่มีทรัพยากรจำกัด

งานวิจัยนี้นำเสนอการพัฒนาการติดตามและควบคุมโหนดอุปกรณ์ที่เชื่อมต่อโหนดด้วยบัส CAN และการพัฒนาโพรโทคอลชั้นสูงกว่า CAN ซึ่งประยุกต์จากหลักการสื่อสารและรูปแบบตารางข้อมูลของโพรโทคอล CANopen เพื่อสร้างระบบติดตามตำแหน่งรถยนต์แบบออนไลน์ โดยมีฟังก์ชันติดตามและบันทึกข้อมูลตำแหน่งจากมอดูล GPS และมีฟังก์ชันสื่อสารข้อมูลแบบออนไลน์กับเว็บเซิร์ฟเวอร์ผ่านโครงข่าย GPRS จากผลการทดลองพบว่าระบบดังกล่าวสามารถรายงานตำแหน่งล่าสุดของรถยนต์ทุกๆ คาบเวลา 15-30 วินาที นอกจากนี้ด้วยโครงสร้างแบบ



กระจายโหนดสามารถแยกพัฒนาแต่ละโหนดได้อย่างอิสระ และสนับสนุนการเพิ่มหรือลดโหนดอุปกรณ์ โดยที่สร้างผลกระทบต่อระบบหลักน้อยที่สุด

2. หลักการทำงานของระบบติดตามตำแหน่งรถยนต์แบบออนไลน์

ระบบติดตามตำแหน่งรถยนต์แบบออนไลน์ประกอบด้วย 2 ส่วนคือ ส่วนเครือข่ายโหนดอุปกรณ์ที่ติดตั้งในรถยนต์ และส่วนคอมพิวเตอร์แม่ข่ายหรือเซิร์ฟเวอร์ ดังแสดงในรูปที่ 1 ส่วนที่ติดตั้งในรถยนต์ทำหน้าที่รายงานตำแหน่งของรถยนต์เพื่อประมวลผล และบันทึกหลักฐานข้อมูลของส่วนเว็บเซิร์ฟเวอร์ โดยสามารถเรียกดูตำแหน่งล่าสุดของรถยนต์จากเว็บเพจของคอมพิวเตอร์ลูกข่ายที่เชื่อมต่ออินเทอร์เน็ต ในบทความนี้เสนอเฉพาะการพัฒนาส่วนเครือข่ายโหนดอุปกรณ์ที่ติดตั้งในรถยนต์ สำหรับการพัฒนาระบบเซิร์ฟเวอร์นั้น เป็นการพัฒนาซอฟต์แวร์เบื้องต้นเพื่อใช้ทดสอบการทำงานส่วนที่ติดตั้งในรถยนต์

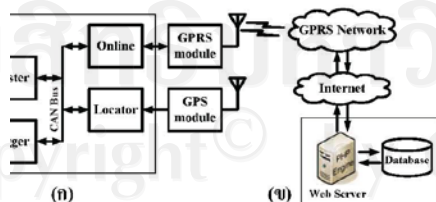
เนื่องจากระบบนี้ใช้คอมพิวเตอร์แม่ข่ายที่ติดตั้งในรถยนต์



โดยที่ระบบนี้ใช้คอมพิวเตอร์แม่ข่ายที่ติดตั้งในรถยนต์ โหนดที่เชื่อมต่อกันด้วยบัสจึงสามารถแบ่งฟังก์ชันให้แตกต่างกันประมวลผล เป็นผลให้สามารถพัฒนาแต่ละโหนดได้ปที่ 1 (ก) แสดงโครงสร้างของโหนดอุปกรณ์ที่ติดตั้งในรถด้วย 4 โหนดคือ โหนด Master ทำหน้าที่ควบคุมการโหนด และจัดการการสื่อสารบนบัส โหนด Logger ทำข้อมูลลงการ์ดหน่วยความจำ โหนด Locator ทำหน้าที่ต้องการจากมอดูล GPS แล้วส่งไปบนบัส สุดท้ายคือทำหน้าที่สื่อสารข้อมูลกับเซิร์ฟเวอร์ผ่านโครงข่าย GPRS

ฟังก์ชันการวัด [2] งานวิจัยนี้ออกแบบให้ใช้ส่วนข้อมูลจำนวนหนึ่งไบต์ของข้อความเป็นคำอ้างอิง โดยเรียกคำอ้างอิงนี้ว่าคำอ้างอิง PDO ซึ่งถูกเข้ารหัส 5 บิตส่วนต้น เพื่อใช้อธิบายความหมายหรือฟังก์ชันงานของข้อความ และ 3 บิตส่วนท้ายถูกเข้ารหัสเพื่อใช้ระบุจำนวนข้อมูลของข้อความ ด้วยกลไกดังกล่าวพบว่าสามารถออกแบบฟังก์ชันงานที่หลากหลายได้ถึง 32 รูปแบบจากหนึ่งหมายเลขข้อความ ตัวอย่างข้อความ เช่น ข้อความตำแหน่ง ซึ่งใช้สื่อสารข้อมูลตำแหน่งจากโหนด Locator ใช้คำอ้างอิง PDO เท่ากับ 14 และมีข้อมูลจำนวน 6 ไบต์ เป็นต้น

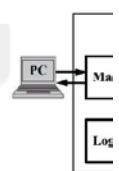
จากโหนดหล
และโหนดแยก
อย่างอิสระ ร
รถยนต์ ประ
ทำงานของทุ
หน้าที่บันทึก
กรองข้อมูลที่
โหนด Online



โครงสร้างระบบติดตามตำแหน่งรถยนต์แบบออนไลน์
ที่ติดตั้งในรถยนต์ (ข) ส่วนคอมพิวเตอร์แม่ข่าย

3.3 สถานะของโหนด

ในงานวิจัยนี้โหนด Master ถูกกำหนดให้เป็นโหนดมาสเตอร์ และโหนดที่เหลือคือโหนดสเลฟ รูปที่ 2 แสดง 4 สถานะของโหนดสเลฟ คือ สถานะ initialization, preoperational, operational และ stopped โหนดมาสเตอร์ใช้กลุ่มข้อความ NMT (RESET, EnterPre, Start และ Stop) เพื่อควบคุมและติดตามสถานะงานของโหนดสเลฟ ทั้งนี้แบ่งการทำงานของแต่ละข้อความในแต่ละสถานะไว้ในวงเล็บ ซึ่งต่างจากโพรโทคอล CANopen [2] เพื่อลดความซับซ้อนและลดจำนวนทรัพยากรที่ใช้ในการพัฒนาซอฟต์แวร์



รูปที่ 1 โ
(ก) ส

ประยุกต์ใช้โพรโทคอล CANopen

SDO

ความ SDO ใช้เพื่อสื่อสารข้อมูลที่มีขนาดไม่ใหญ่และเล็ก



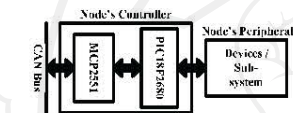
3. หลักการ

3.1 ข้อความ

4. การพัฒนาฮาร์ดแวร์และซอฟต์แวร์

4.1 ฮาร์ดแวร์

ในการพัฒนาระบบเครือข่ายอุปกรณ์โหนด CAN มีส่วนประกอบสำคัญ 3 ส่วนที่ต้องพิจารณาคือ ไมโครคอนโทรลเลอร์ CAN คอนโทรลเลอร์และ CAN ทรานซีฟเวอร์ งานวิจัยนี้เลือกใช้ ไมโครคอนโทรลเลอร์เบอร์ PIC18F2680 (8 บิต) ที่มี CAN คอนโทรลเลอร์ภายในตัว [4] และ CAN ทรานซีฟเวอร์เบอร์ MCP2551 เพื่อสร้างส่วนควบคุมโหนด ในรูปที่ 3 PIC18F2680 มีพื้นที่โปรแกรมขนาด 64 กิโลไบต์และมีหน่วยความจำ 3 กิโลไบต์ ส่วนควบคุมโหนดทุกตัวทำงานด้วยสัญญาณนาฬิกาที่ความถี่เท่ากับ 11.0592 เมกะเฮิร์ตซ์ และมีบัส CAN เป็นส่วนติดต่อหลัก โดยทำหน้าที่สื่อสารกันด้วยอัตราบิตเท่ากับ 500 กิโลบิตต่อวินาที ทั้งนี้แต่ละโหนดติดต่อกับอุปกรณ์บริวารที่แตกต่างกันไปตามฟังก์ชันงานของโหนด



รูปที่ 3 โครงสร้างฮาร์ดแวร์ของแต่ละโหนด

โหนด Master มีส่วนติดต่อวงจรแปลงสัญญาณผลจากจาก ไมโครคอนโทรลเลอร์เป็นสัญญาณยูเอสบี เพื่อสื่อสารข้อมูลระหว่างโหนดกับคอมพิวเตอร์ ด้วยอัตราบิตเท่ากับ 56,700 บิตต่อวินาที ส่วนโหนด Logger ติดต่อกับการ์ดหน่วยความจำแบบ MMC ขนาด 2 กิโลไบต์ สำหรับโหนด Locator มีส่วนติดต่อกับวงจรฐานเวลาจริง และส่วนติดต่อกับมอดูล GR82 ซึ่งให้ข้อมูล GPS ตามมาตรฐาน NMEA สามารถระบุตำแหน่ง โดยมีความคลาดเคลื่อนไม่เกิน 20 เมตร สุดท้ายคือโหนด Online มีส่วนติดต่อกับมอดูล GR64 ซึ่งสามารถติดต่ออินเทอร์เน็ตผ่านโครงข่าย GPRS โดยใช้โพรโทคอล HTTP เพื่อสื่อสารข้อมูลระหว่างเครือข่ายโหนดอุปกรณ์ที่ติดตั้งในรถยนต์ และเซิร์ฟเวอร์

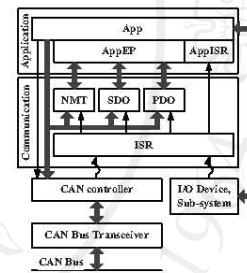
4.2 ซอฟต์แวร์

รูปที่ 4 แสดงภาพรวมโครงสร้างซอฟต์แวร์ ประกอบด้วย 2 มอดูลหลักคือ มอดูลสื่อสารและมอดูลโปรแกรมประยุกต์ และมีมอดูลย่อย 7 มอดูลคือ มอดูล ISR, AppISR, NMT, SDO, PDO, AppEP และมอดูล App ซึ่งทิศทางของข้อมูล สัญญาณขัดจังหวะ และการเรียกใช้ฟังก์ชันงานถูกแทนด้วย ลูกศรเงา ลูกศรคลื่น และลูกศรตรง ตามลำดับ โดยโหนดแต่ละฟังก์ชันมอดูลสื่อสารเหมือนกันทุกโหนด และมีฟังก์ชันมอดูลโปรแกรมประยุกต์ที่แตกต่างกันไปตามฟังก์ชันงานของแต่ละโหนด

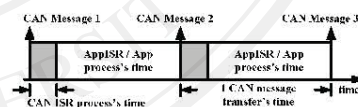
ในการสื่อสารข้อมูลระหว่างโหนดด้วยบัส CAN ที่อัตราบิต 500 กิโลบิตต่อวินาที หนึ่งข้อความ CAN ใช้เวลาสื่อสารประมาณ 94-

260 ไมโครวินาที [5] ในรูปที่ 5 แสดงช่วงเวลาประมวลผลฟังก์ชันงานเมื่อโหนดได้รับข้อความ CAN มากกว่า 2 ข้อความต่อเนื่องกัน ส่วนที่แรเงาคือช่วงเวลาที่มีมอดูล ISR ใช้ประมวลผลข้อความ CAN ที่ได้รับ เพื่อตัดสินใจว่าฟังก์ชันมอดูลใดจะประมวลผลข้อมูลที่รับ และส่วนที่เหลือคือช่วงเวลาที่มีมอดูลย่อยอื่นใช้ประมวลผล ดังนั้นถ้าฟังก์ชันมอดูล ISR ใช้เวลาประมวลผลสั้นๆ จะทำให้เวลาสำหรับประมวลผลฟังก์ชันงานอื่นมีเพิ่มขึ้น

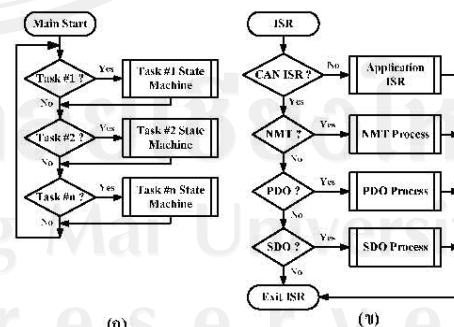
สำหรับงานวิจัยนี้มีการพัฒนาซอฟต์แวร์ออกเป็นฟังก์ชันงานย่อยดังแสดงผังการทำงานในรูปที่ 6 เพื่อให้สามารถประมวลผลฟังก์ชันงานดังกล่าวให้เสร็จทันก่อนที่โหนดจะได้รับข้อความ CAN ถัดไป ทั้งนี้พบว่าเวลาที่มอดูล ISR ใช้ประมวลผลข้อความ CAN มีค่าน้อยที่สุดประมาณ 49 รอบทำงานของ PIC18F2680 หรือ 18 ไมโครวินาที โดยใช้วิธีการคำนวณเวลาการทำงานตาม [4] นอกจากนี้ซอฟต์แวร์ส่วนมอดูลสื่อสารที่พัฒนาขึ้นใช้พื้นที่โปรแกรมเพียง 2 กิโลไบต์และใช้พื้นที่หน่วยความจำเพียง 256 ไบต์ ทำให้เหลือทรัพยากรเพื่อทำฟังก์ชันมอดูลโปรแกรมประยุกต์ของแต่ละโหนดได้มากขึ้น



รูปที่ 4 ภาพรวมโครงสร้างซอฟต์แวร์



รูปที่ 5 ช่วงเวลาประมวลผลฟังก์ชันงาน



(ก)

(ข)

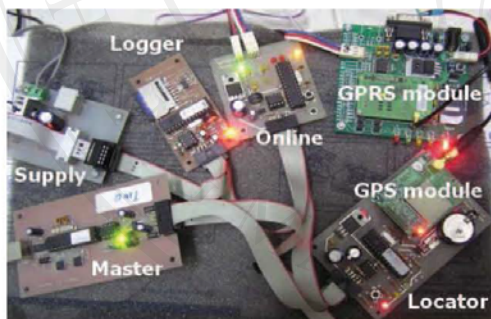
รูปที่ 6 ผังการทำงานของซอฟต์แวร์

(ก) ผังทำงานหลัก (ข) ผังทำงานมอดูล ISR

5. ผลการทดสอบ

ส่วนเครือข่ายโหนดอุปกรณ์ดังแสดงในรูปที่ 7 ถูกติดตั้งในรถยนต์ และถูกกำหนดพารามิเตอร์ของแต่ละโหนดผ่านบัส CAN ด้วยการเชื่อมต่อคอมพิวเตอร์กับโหนด Master โดยที่กำหนดให้โหนด Locator สร้างข้อความตำแหน่งข้อมูลทุกๆ 20 (1-240) ของจำนวนข้อความ SYNC และสร้างข้อความ TIME ทุกๆ 5 (0-60) วินาทีให้แก่ระบบ ส่วนโหนด Logger ถูกกำหนดให้บันทึกข้อความตำแหน่งและข้อความ TIME ทั้งหมด สุดท้ายคือโหนด Online ถูกกำหนดให้รายงานข้อความตำแหน่งและข้อความ TIME ให้แก่เว็บเซิร์ฟเวอร์ทุกๆ 20 (15-30) วินาที ทั้งนี้ควบคุมจังหวะทำงานทั้งคันงานด้วยข้อความ SYNC ทุกๆ 10 มิลลิวินาทีจากโหนด Master

จากผลการทดลองพบว่าสามารถตรวจสอบตำแหน่งล่าสุดของรถยนต์ด้วยข้อมูลและแผนที่ในหน้าเว็บเพจทุกๆ 20 วินาทีแสดงในรูปที่ 8 ทั้งนี้พบว่าโหนดอุปกรณ์สามารถใช้สร้างงานที่มีความซับซ้อนและมีหลายขั้นตอน เช่น การติดต่ออินเทอร์เน็ต และการบันทึกข้อมูลในการมัลติมีเดียได้ เมื่อเปรียบเทียบกับ [3], [6] พบว่าจำนวนทรัพยากรที่ใช้มีขนาดเล็กใกล้เคียงกัน และโหนดอุปกรณ์นี้ไม่มีค่าใช้จ่ายในการนำไปใช้งาน นอกจากนี้ได้ทดลองพัฒนาโหนดวัดสภาพอากาศ โดยเชื่อมต่อโหนดกับเซนเซอร์วัดอุณหภูมิ ความชื้น และความดัน พบว่าสามารถทำงานร่วมกับโหนดอื่นๆ ได้ทันทีด้วยการปรับเปลี่ยนพารามิเตอร์ทางซอฟต์แวร์



รูปที่ 7 เครือข่ายโหนดอุปกรณ์ที่ติดตั้งในรถยนต์



6. สรุป

บทความนี้ได้นำเสนอการพัฒนาแบบติดตามตำแหน่งรถยนต์แบบออนไลน์ผ่านโครงข่าย GPRS โดยใช้ระบบฝังตัวแบบกระจายที่เชื่อมต่อโหนดด้วยบัส CAN และจัดลำดับการสื่อสารด้วยโพรโทคอลที่ประยุกต์จากหลักการสื่อสารของโพรโทคอล CANopen จากการผลิตผลร่วมกันของโหนดอุปกรณ์ทั้ง 4 โหนด ที่สื่อสารกันด้วยอัตราบิต CAN เท่ากับ 500 กิโลบิตต่อวินาที ทำให้ระบบสามารถติดตามบันทึกและรายงานตำแหน่งของรถยนต์แบบออนไลน์ผ่านโครงข่าย GPRS ทุกๆ 20 วินาที นอกจากนี้ด้วยโครงสร้างแบบกระจายโหนดและการแบ่งพัฒนาซอฟต์แวร์ออกเป็นมอดูลย่อย ทำให้สามารถเพิ่มหรือลดโหนด โดยสร้างผลกระทบต่อโหนดอื่นน้อยที่สุด ทั้งนี้ทุกโหนดใช้ทรัพยากรเบื้องต้นสำหรับสื่อสารข้อมูลระหว่างโหนด โดยใช้พื้นที่โปรแกรม 2 กิโลไบต์ (3.1%) และใช้พื้นที่หน่วยความจำ 256 (7.7%) ทำให้สามารถประยุกต์ใช้โหนดเพื่อพัฒนาฟังก์ชันงานได้มากขึ้น ด้วยประสิทธิภาพและจำนวนทรัพยากรที่เพียงพอของโหนด

เอกสารอ้างอิง

- [1] M. Farsi, K. Ratcliff, M. Barbosa, "An overview of controller area network," Computing & Control Engineering Journal, vol.10, no.3, pp.113-120, Aug. 1999
- [2] M. Farsi, K. Ratcliff, M. Barbosa, "An introduction to CANopen," Computing & Control Engineering Journal, vol.10, no.4, pp. 161-168, Aug. 1999
- [3] P. Krist, "MicroCANopen distributed control node Ethernet HTTP monitoring," Applied Electronics, 2009. AE 2009, pp.165-168, 9-10 Sept 2009
- [4] Microchip Technologies Inc, "PIC18F2X8X Datasheet," 2004
- [5] K. Tindell, A. Burns, "Guaranteeing Message Latencies on Controller Area Network (CAN)," Proceedings 1st International CAN Conference, Mainz (Germany), Sep 1994
- [6] J. Portillo, E. Estevez, I. Cabanes, M. Marcos, "CANopen Network for microcontroller-based Real Time Distributed Control Systems," IEEE Industrial Electronics, IECON 2006 – 32nd Annual Conference, pp. 4644-4649, 6-10 Nov 2006

Timestamp	Position	Connected Time
2010-05-12 15:20:00	18° 47' 46.866", 96° 57' 6.21"	2010-05-12 15:21:15

ประวัติผู้เขียน

ชื่อ-สกุล

นายปองพล พรหมป่า

วัน เดือน ปี เกิด

18 มีนาคม พ.ศ. 2527

ประวัติการศึกษา

สำเร็จการศึกษาระดับปริญญาตรี วิทยาศาสตร์บัณฑิต
สาขาวิชาฟิสิกส์ คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่ ปี
การศึกษา 2548

ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่
Copyright© by Chiang Mai University
All rights reserved