

บทคัดย่อ

โครงการวิจัยนี้นำเสนอการออกแบบอุปกรณ์สั่งอาหารแบบไร้สาย ซึ่งประกอบไปด้วยชุดทำงานส่วนของลูกค้า และส่วนประมวลผลกลาง โดยใช้การป้อนรหัสของรายการอาหารเข้าสู่เครื่องสั่งอาหาร ประกอบด้วยไมโครคอนโทรเลอร์และแสดงผลทางจอLCD ชุดทำงานจะทำการส่งรายการอาหาร เข้าสู่เครื่องรับของหน่วยประมวลผลกลางแบบไร้สาย และแสดงผลบนจอคอมพิวเตอร์ พร้อมทั้งคิดเงินได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Abstract

This project describes a study and implementation of wireless menu ordering equipment. It consists of user command and server units. The user command unit is including a LCD display, keypad, microcontroller and RF unit. The order command is sent from user command unit to server unit based on radio frequency. The server is able to make a billing reported.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
สารบัญ	III
สารบัญตาราง	IV
สารบัญรูป	V
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	2
บทที่ 3 ผลการทดลอง	4
บทที่ 4 สรุปและข้อเสนอแนะ	11
บรรณานุกรม	12



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 3.1 การกำเนิตรหัสที่ผิดจากตัวนับแบบสุ่ม	5
ตารางที่ 3.2 การสลับตำแหน่งบิตในภาคส่ง	6
ตารางที่ 3.3 การสลับตำแหน่งบิตในภาครับ	10



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 แสดงผังวงจรของระบบป้อนข้อมูล	2
รูปที่ 3.1 แสดงวงจรซีพียูรีจิสเตอร์ป้อนกลับของลำดับ $p(x) = x^8 + x^4 + x^3 + x^2 + 1$	4
รูปที่ 3.2 วงจรจัดเฟรมข้อมูลภาคส่ง	7
รูปที่ 3.3 วงจรจัดเฟรมข้อมูลภาครับ	7
รูปที่ 3.4 แสดงการเข้ารหัสเสียงสัญญาณ	8



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

จากที่เห็นว่าปัจจุบันนี้ ร้านอาหารได้เปิดกิจการขึ้นเป็นจำนวนมากมาย ทำให้เกิดการ แข่งขันกันสูงระหว่างร้านอาหารต่างๆ ยิ่งถ้าร้านไหนมีการบริการที่ดี อาหารอร่อยถูกใจลูกค้า ก็ จะเป็นสิ่งที่ทำให้มีลูกค้าเข้ามารับประทานอาหารมากขึ้น และเมื่อมีลูกค้ามากขึ้น ก็จะทำให้เกิด ปัญหาในด้านการบริการลูกค้า เช่น บริการรับรายการอาหารจากลูกค้าได้ช้า ลูกค้าต้องรอนาน จน ลูกค้าเกิดความไม่พอใจ อาจทำให้ร้านอาหารร้านนั้นเสียลูกค้าได้ จึงมีความจำเป็นอย่างยิ่งที่ จะต้องมีเครื่องส่งอาหารแบบ ไร้สายในกรณีที่ให้ลูกค้าส่งอาหารได้อย่างสะดวกและรวดเร็วขึ้น การทำงานของเครื่องส่งอาหารแบบ ไร้สายนั้น ประกอบด้วย 2 ส่วน ได้แก่ ส่วนที่หนึ่ง เป็นส่วน ของชุดส่งงานของลูกค้า ซึ่งจะใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มาควบคุมคีย์แพด ให้ทำ หน้าที่รับคำสั่งที่ลูกค้ากด แล้วนำค่านั้นเก็บไว้ในหน่วยความจำพร้อมทั้งแสดงผลที่หน้าจอ LCD ส่วนค่าที่เก็บไว้ในหน่วยความจำนั้น จะถูกส่งออกไปผ่าน วงจร RF โมดูลของตัวส่ง ในส่วนที่ สองคือ ส่วนของหน่วยประมวลผลกลาง ในส่วนนี้จะมี RF โมดูลตัวรับทำหน้าที่รับข้อมูลจาก ส่วนของชุดส่งงานของลูกค้า แล้วส่งผ่านพอร์ตอนุกรมเข้าไปยังเครื่อง Server ซึ่งเครื่อง Server จะ ทำการประมวลผลของรหัสต่างๆ นอกจากนี้ ยังสามารถตอบกลับไปยังชุดส่งงานของลูกค้าเพื่อ ยืนยันการส่งอาหาร ว่าได้รับแล้ว หรือต้องการยกเลิกรายการอาหารที่ได้สั่งไป และยังสามารถเรียก เก็บเงินพร้อมสั่งให้พิมพ์เพื่อใช้สำหรับการคิดเงินกับลูกค้าได้

บทที่ 2

ทฤษฎี

2.1 ทฤษฎีของไมโครคอนโทรลเลอร์ตระกูล MCS-51

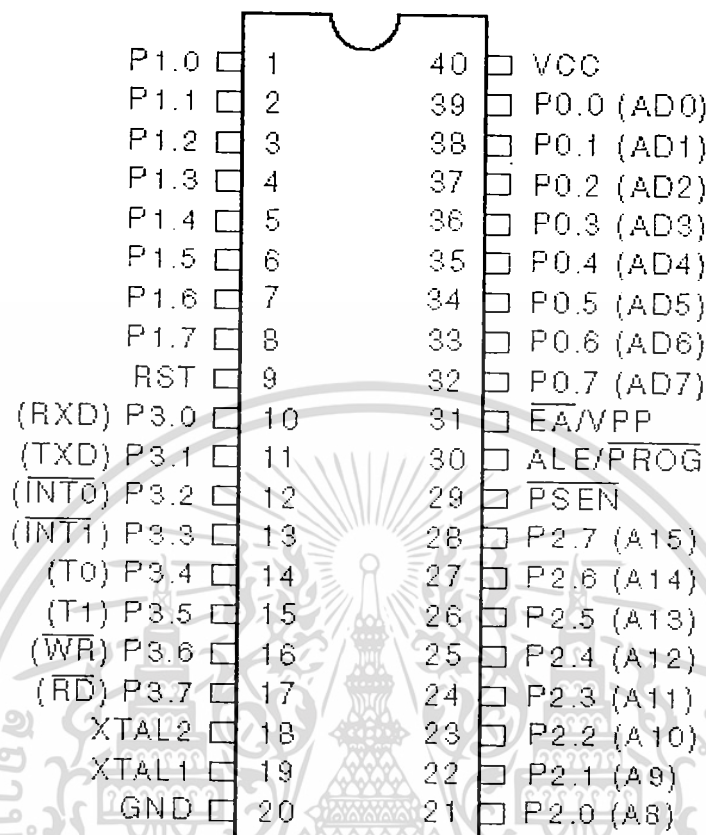
ในเครื่องสั่งอาหารแบบไร้สายนั้นจะมีการนำไมโครคอนโทรลเลอร์มาใช้งานด้วย ดังนั้นจึงจำเป็นต้องทราบถึงทฤษฎีต่างๆของไมโครคอนโทรลเลอร์ซึ่งมีดังนี้

2.1.1 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล MCS-51

คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีดังต่อไปนี้

- เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต
- มีวงจรถ่ายสัญญาณและวงจรมัลติเพล็กซ์สัญญาณภายในไอซี
- มีขาสัญญาณอินพุตและเอาต์พุตจำนวน 32 บิต
- สามารถเชื่อมหน่วยความจำข้อมูลภายนอก (External Data Memory) โดยการอ้างตำแหน่งแอดเดรสได้ถึง 64 K
- สามารถเชื่อมต่อหน่วยความจำโปรแกรมภายนอก (External Program Memory) โดยการอ้างตำแหน่งแอดเดรสได้ถึง 64 K
- มีหน่วยความจำข้อมูลภายในตัว (On-Chip Data Memory) ขนาด 128 ไบต์ โดยเฉพาะหมายเลข 8032 และ M8052 จะมีหน่วยความจำในส่วนนี้ถึง 256 ไบต์
- มีหน่วยความจำโปรแกรมภายในตัว (On-Chip Program Memory) ขนาด 4 K โดยเฉพาะหมายเลข 8052 จะมีหน่วยความจำในส่วนนี้ถึง 8 K สำหรับหมายเลข 8031 และ M8032 จะไม่มีหน่วยความจำในส่วนนี้
- หน่วยความจำข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลระดับบิตได้ด้วย ทำให้สามารถควบคุมหรือตรวจสอบสถานะบิตได้ง่าย ส่งผลให้สามารถเขียนโปรแกรมได้ง่ายขึ้น
- มีไทมเมอร์/เคาน์เตอร์ (Timer/Counter) ขนาด 16 บิตจำนวน 2 ตัว โดยเฉพาะหมายเลข 8032 หรือ 8052 จะมีไทมเมอร์/เคาน์เตอร์จำนวน 3 ตัว
- สามารถทำการอินเตอร์รัพท์ได้จากแหล่งกำเนิด 5 แหล่ง โดยเฉพาะหมายเลข 8032 และ 8052 จะทำการอินเตอร์รัพท์ได้จากแหล่งกำเนิด 6 แหล่ง พร้อมทั้งยังสามารถจัดลำดับความสำคัญของการอินเตอร์รัพท์ได้ 2 ระดับ
- มีพอร์ตสื่อสารอนุกรมภายในตัวเอง ซึ่งการทำงานจะเป็นแบบฟูลดูเพล็กซ์ (Full Duplex)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C5x

2.2 การสื่อสารข้อมูลอนุกรม

การสื่อสารข้อมูลอนุกรมเป็นการรับหรือส่งข้อมูลในลักษณะกลุ่มของบิตคราวละหนึ่งบิต เรียงลำดับเรื่อยไปจนสิ้นสุดการสื่อสารแบบนี้จะมีข้อแตกต่างจากการสื่อสารแบบขนานเป็นอย่างมากเนื่องจากข้อมูลมีการโอนย้ายมาพร้อมกันจึงมีความจำเป็นต้องใช้จำนวนเส้นสัญญาณมากขึ้นตามจำนวนบิตของข้อมูลด้วย ในขณะที่การสื่อสารแบบอนุกรมนี้ต้องการเส้นสัญญาณเพียงสองหรือสามเส้นเท่านั้นดังนั้นในการสื่อสารแบบขนานจึงไม่เหมาะสมในการสื่อสารกับอุปกรณ์ภายนอกเป็นระยะไกลๆ เพราะจะทำให้สิ้นเปลืองค่าใช้จ่ายมาก

ความเร็วของการสื่อสารแบบอนุกรม

เนื่องจากการสื่อสารข้อมูลแบบอนุกรมเป็นการรับส่งข้อมูลในลักษณะกลุ่มของบิตข้อมูล (Bit Stream) ดังนั้นจึงต้องให้ความสนใจในการพิจารณาถึงเรื่องอัตราความเร็วในการรับส่งบิต เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหล่านี้เป็นลำดับแรก โดยทั่วไปมักจะระบุกันในหน่วยของจำนวนบิตข้อมูลภายในเวลาหนึ่งวินาที เรียกว่า อัตราบอด ตามค่ามาตรฐานเหล่านี้ ได้แก่ 110,150,300,1200,2400,4800,9600,19200 บอด ข้อมูลทั้ง 8 บิตนี้ หากถูกส่งออกมาด้วยอัตรา 2400 บอด จะใช้เวลาในการส่งข้อมูลหนึ่งบิตมีค่าเท่ากับ $1/2400$ หรือ $416 \mu\text{S}$ และ เวลาในการส่งข้อมูลทั้ง 8 บิตมีค่าเท่ากับ (8×416) หรือ $3328 \mu\text{S}$

รูปแบบของการส่งข้อมูลอนุกรม

การสื่อสารอนุกรมแบบอะซิงโครนัสจะใช้การแปลงข้อมูลขนานให้เป็นอนุกรมแล้วเพิ่มบิตบางอย่างร่วมไปกับการส่งข้อมูลจริงซึ่งได้แก่

บิตเริ่มต้น (Start Bit)

บิตเริ่มต้นมีหน้าที่สำหรับการบ่งบอกให้ทราบถึงตำแหน่งจุดเริ่มต้นก่อนบิตข้อมูลตามปกติแล้วค่าของบิตเริ่มต้นจะเป็นระดับลอจิกต่ำ

บิตแสดงภาวะความเป็นเลขคู่หรือเลขคี่ (Parity Bit)

บิตนี้มีหน้าที่เพื่อการตรวจสอบความถูกต้องของข้อมูล โดยทั่วไปมักเรียกว่า บิตพาริตีและจะนำไปต่อท้ายบิตข้อมูล ค่าของบิตนี้จะขึ้นอยู่กับจำนวนค่าของบิตที่เป็น 1 ซึ่งจะเป็นได้สองลักษณะคือ พาริตีคู่ (Even Parity) หรือพาริตีคี่ (Odd Parity) ตัวอย่างเช่นระบบที่ติดต่อกัน โดยระบุว่า จะใช้พาริตีคี่ ทางด้านส่งจะนำข้อมูลที่จะส่งมาพิจารณา จำนวนของบิตที่มีค่า 1 เป็นเลขจำนวนคู่ อยู่แล้ว ค่าของบิตพาริตีจะมีค่าเป็น 0 แต่หากว่าจำนวนของบิตที่มีค่าเป็น 1 เป็นจำนวนเลขคี่ ค่าของบิตพาริตีจะมีค่าเป็น 1 การพิจารณาทางด้านรับเป็นการตรวจสอบจำนวนบิตที่มีค่า 1 ของข้อมูลที่ได้รับมาทั้งหมดรวมทั้งบิตพาริตี ถ้ามีค่าเป็นเลขจำนวนคู่ แสดงว่าข้อมูลที่รับเข้ามานี้ถูกต้อง หากไม่เป็นเลขจำนวนคู่แสดงว่าเกิดการผิดพลาดของข้อมูลขึ้น เป็นต้น

บิตสุดท้าย (Stop Bit) เป็นบิตที่เพิ่มขึ้นเพื่อระบุถึงขอบเขตการสิ้นสุดของกลุ่มบิตสุดท้ายนี้สามารถโปรแกรมได้คือ 1 บิต $1 \frac{1}{2}$ บิต และ 2 บิต ดังนั้นกรณีของการส่งข้อมูล 8 บิตหากข้อมูลถูกส่งออกไปด้วยอัตราเร็ว 2400 บอด เวลาโดยรวมในการส่งข้อมูลหนึ่งไบต์จะมีค่าเป็น $(12 \times 416) \mu\text{S}$ หรือ 4.99 mS

การส่งข้อมูลอนุกรมของ 8051

พอร์ตอนุกรมของ 8051 มีโครงสร้างการทำงานในแบบที่เรียกว่า ฟูลดูเพล็กซ์ (Full Duplex) ในการรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน โดยทางด้านวงจรของตัวส่ง (Transmitter) ประกอบด้วยข้อมูลออกไปยังพอร์ตอนุกรมทางขาสัญญาณ TxD (พอร์ต 3.1) ส่วนวงจรรด้านตัวรับ (Receiver) ประกอบด้วย SUBF เช่นเดียวกับสัญญาณข้อมูลอนุกรมที่รับเข้ามาทางขาสัญญาณ RxD (พอร์ต 3.0)

พอร์ตอนุกรมของ 8051

สามารถโปรแกรมการทำงานได้หลายโหมดด้วยกันโดยเลือกที่บิต SM 0 และ SM 1 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานทั้ง 4 โหมดของพอร์ตอนุกรม มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 0 : ใช้รับส่งข้อมูล 8 บิต โดยการส่งจะเลื่อนออกทีละบิต โดยส่งบิต 0 ออกไปก่อนทางขา RxD และไม่มีการส่ง start bit แต่จะส่ง shift clock ทางขา TxD ความเร็ว 1/12 เท่าของ CPU CLOCK

โหมด 1: ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART (Universal Asynchronous Receiver/Transmitter) โดยส่งแบบ 10 บิต ข้อมูล 8 บิต 1 start bit และ 1 stop bit และสามารถเปลี่ยนแปลงอัตราเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ Timer 1

โหมด 2 : ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART โดยการใช้กลุ่มข้อมูลแบบ 11 บิต และกำหนดอัตราความเร็วในการส่งข้อมูลเท่ากับ 1/32 และ 1/64 ของ CPU CLOCK โดยโปรแกรมที่บิต SMOD ใน PCON

โหมด 3 : ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART โดยการใช้กลุ่มข้อมูลแบบ 11 บิต และสามารถเปลี่ยนแปลงอัตราความเร็วในการส่งข้อมูลได้ โดยควบคุมที่บิต SMOD และอัตราโอเวอร์โพล์ของ Timer 1 นอกจากนี้ โหมด 2 และ 3 ยังมีการดำเนินการอีกแบบหนึ่ง โดยสามารถนำมาใช้ประโยชน์ในการสื่อสารข้อมูลแบบที่มีไมโครโปรเซสเซอร์หลายตัวทำงานร่วมกันได้ซึ่งมีชื่อเรียกว่า Multiprocessor Mode

2.3 Single chip 2.4 GHz Transceiver nRF2401

Mode of operation (โหมดดำเนินการ)

2.3.1 Overview

ระบบย่อยของ nRF2401 สามารถจัดอยู่ในโหมดหลักได้ดังต่อไปนี้ โดยจะขึ้นกับ 3 pins ควบคุม

ตารางที่ 2.7 nRF2401 subsystem main modes

Mode	PWR_UP	CE	CS
Active	1	1	0
Configuration	1	0	1
Stand by	1	0	0
Power down	0	X	X

2.3.2 Active modes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบย่อย nRF2401 มี 2 Active โหมด (TX / Rx) :

- Shock Burst
- Direct Mode (not supported by nRF24E1)

ฟังก์ชันการทำงานของอุปกรณ์ในโหมดนี้เลือกโดย เนื้อหาของ Configuration word โดย Configuration word จะแสดงในส่วนของ Configuration

2.3.3 Shock Burst

เทคโนโลยี Shock Burst ใช้บัฟเฟอร์ FIFO เพื่อจับเวลาข้อมูลที่อัตราข้อมูลต่ำและส่งต่อที่อัตราข้อมูลสูงมาก ด้วยเหตุนี้จะสามารถทำให้เกิดการลดทอนของกำลังสูงสุด เมื่อดำเนินการระบบย่อย nRF2401 ใน Shock Burst เราจะได้อัตราข้อมูลสูงสุด (1 Mbps) โดย band 2.4 GHz โดยไม่ต้องใช้ไมโครคอนโทรลเลอร์ความเร็วสูงและราคาสูง (MCU) สำหรับการดำเนินการทางข้อมูล

โดยการใส่การประมวลผลข้อมูลความเร็วสูง สัมพันธ์กับ RF Protocol บนชิป nRF24E1 มีประโยชน์ ดังนี้

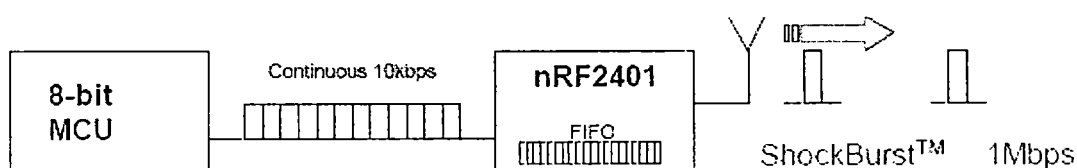
- ลดการใช้กระแส
- ระบบที่มีราคาถูก (ใช้งานได้ง่ายสำหรับ microcontroller ราคาไม่แพง)
- ลดอัตราการเสี่ยงของการชนกันในอากาศ เนื่องจากการส่งสัญญาณในช่วงเวลาสั้นๆ ได้ดี

ระบบย่อย nRF 2401 สามารถโปรแกรมโดยใช้ สาย interface พื้นฐาน 3 เส้น ซึ่งอัตราของข้อมูลจะเลือกโดยความเร็ว CPU โดยยอมรับส่วนดิจิทัลของ application เพื่อจะ run ที่ความเร็วต่ำ ขณะที่อัตราข้อมูลบน RF Link มีค่าเพิ่มสูงสุด โหมด Shock Burst จะลดการใช้กระแสเฉลี่ยใน application

2.3.3.1 หลักการของ Shock Burst

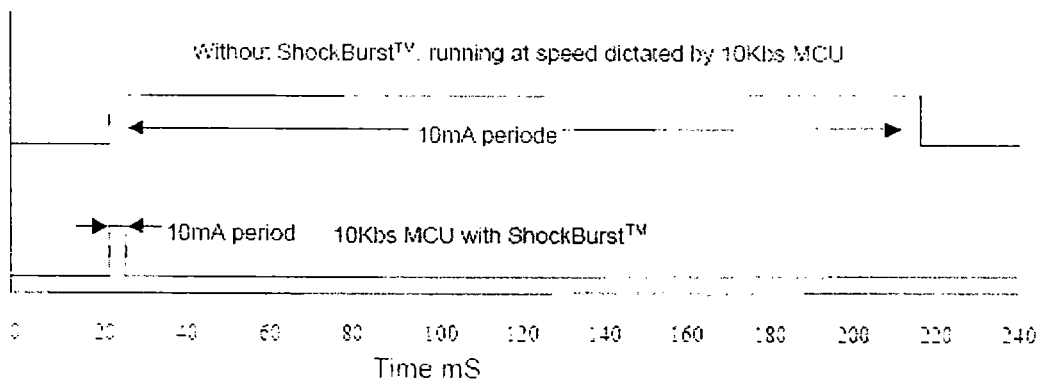
เมื่อระบบย่อย nRF 2401 มีโครงร่างใน Shock Burst กระบวนการ Tx จะดำเนินการดังต่อไปนี้

ตัวอย่าง 10 kbps



รูปที่ 2.13 จับเวลาข้อมูลโดย CPU ส่งโดยเทคโนโลยี Shock Burst

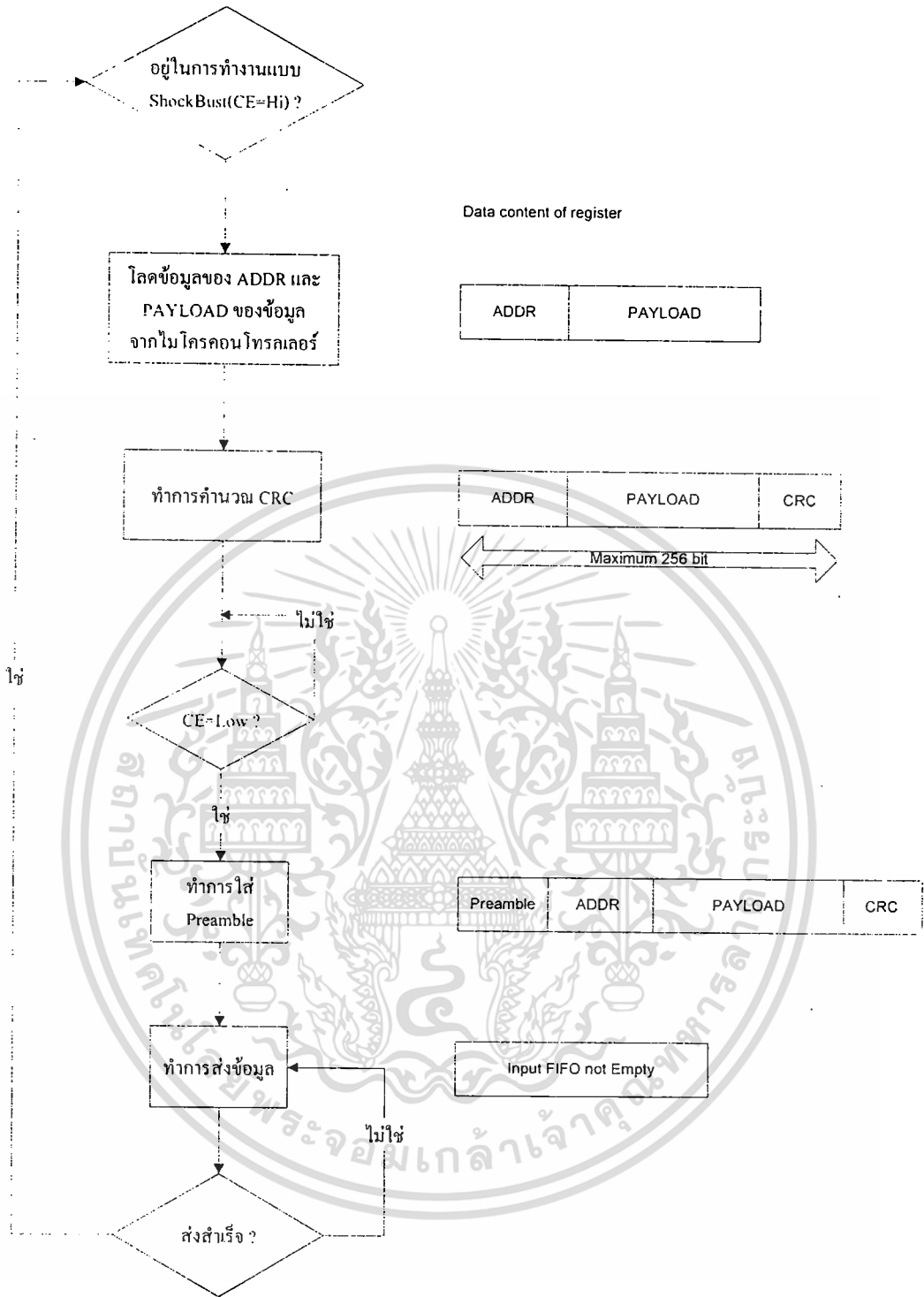
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 การใช้กระแส RF โดยใช้และไม่ใช้เทคโนโลยี Shock Burst



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงแผนผังงาน(Flow Chart) ShockBurst การส่งของระบบย่อย nRF2401

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

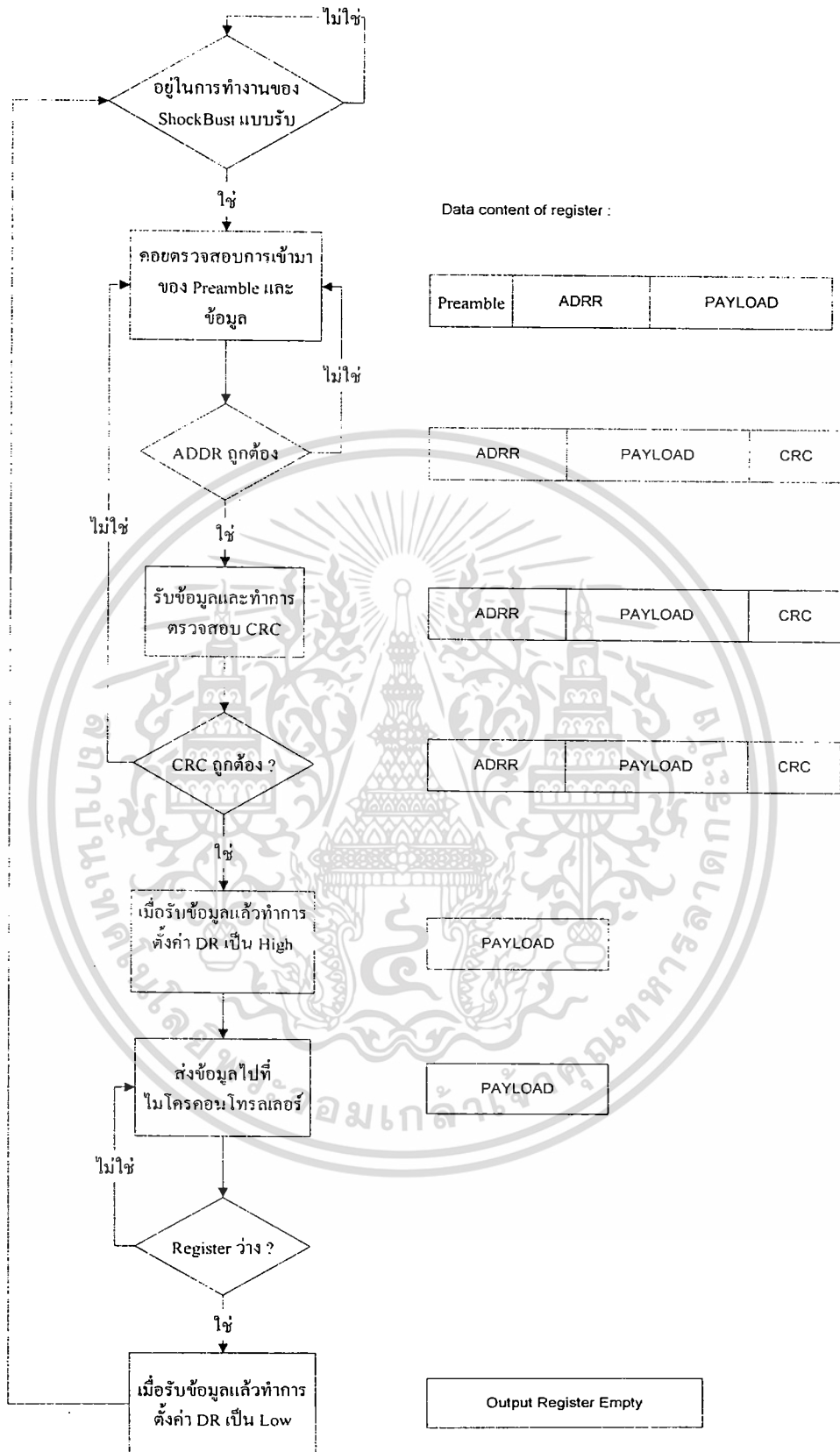
2.3.3.2 การส่ง Shock Burst

CPU interface pins : CE, CLK1, DATA

1. เมื่อ CPU มีข้อมูลที่จะส่ง set CE high คือ กระตุ้น nRF2401 ให้ประมวลผลบนบอร์ด
2. แอดเดรสของโหนดรับ (Rx) และ payload ข้อมูลจะจับเวลาเข้าสู่ช่วงเวลาระบบย่อย nRF2401 protocol หรือ CPU ตั้งความเร็วต่ำกว่า 1 Mbps
3. CPU ตั้ง CE low คือ กระตุ้นการส่ง Shock Burst
4. Shock Burst
 - RF front end จะมีกำลังเพิ่มขึ้น
 - RF Package จะสมบูรณ์
 - ข้อมูลจะถูกส่งที่ความเร็วสูง (250 bps หรือ 1Mbps แล้วแต่ผู้ใช้)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แสดงแผนผังงาน (Flow Chart) ShockBurst การรับระบบย่อย nRF2401

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3.3 การรับ Shock Burst

CPU interface pins : CE, DRT, CLK1, DATA (1 ช่องรับ Rx)

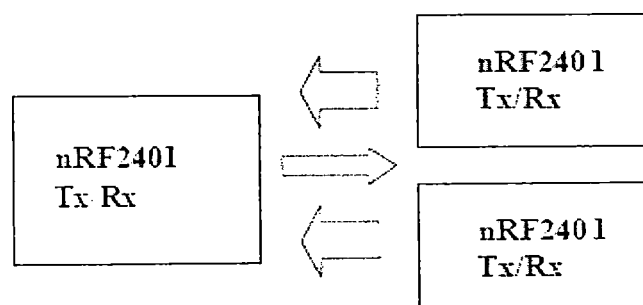
1. แอดเดรสถูกตั้งและขนาดของ payload ของ RF package ที่เข้ามาจะ set เมื่อระบบย่อย nRF
2. 2401 เป็นโครงสร้างของ Shock Burst Rx
3. เมื่อกระตุ้น Rx ; set CE high
4. หลังจากตั้ง 200us nRF 2401 จะตรวจสอบอากาศสำหรับการสื่อสารที่จะเข้ามา เมื่อ package ที่ถูกต้องถูกเรียกมา (แอดเดรสถูกตั้งและพบ CRC) nRF 2401 จะทำการย้ายบิตนำแอดเดรสและบิต CRC
5. จากนั้นระบบย่อย nRF 2401 จะ interrupt CPU โดยการตั้ง DR1 high
6. CPU จะ set CE low เพื่อป้องกัน RF front end (โหมดที่กระแสต่ำ)
7. CPU จะจับเวลาที่หยุด payload data ที่อัตราที่เหมาะสม
8. เมื่อ payload data ทั้งหมดถูกเรียกมา nRF 2401 set DR1 low อีกครั้ง และพร้อมสำหรับข้อมูลที่จะเข้ามา ถ้า CE ยัง high ระหว่างที่ download ข้อมูล ถ้า CE set low การลำดับ start up ใหม่จะสามารถเริ่มได้

2.3.4 DUOCiever Simultaneous two channel Receive mode

ในโหมด Shock Burst ของ nRF 24E1 สามารถรองรับความถี่ 2channel ที่ขนานกันและเป็นอิสระต่อกัน ที่อัตราข้อมูลสูงสุด ได้สะดวกในเวลาเดียวกัน ซึ่งหมายความว่า

- RF 2401 สามารถรับข้อมูลจากเครื่องส่ง 2 เครื่อง ที่ 1 Mbps และ 8 MHz ผ่าน 1 เสออากาศ
- Output จาก 2 ช่องสัญญาณข้อมูล จะป้อนให้กับ 2 interface pins ที่แยกจากกัน
- Data Channel 1: CLK1, DATA, DR1
- Data Channel 2: CLK2, DATA2, DR2

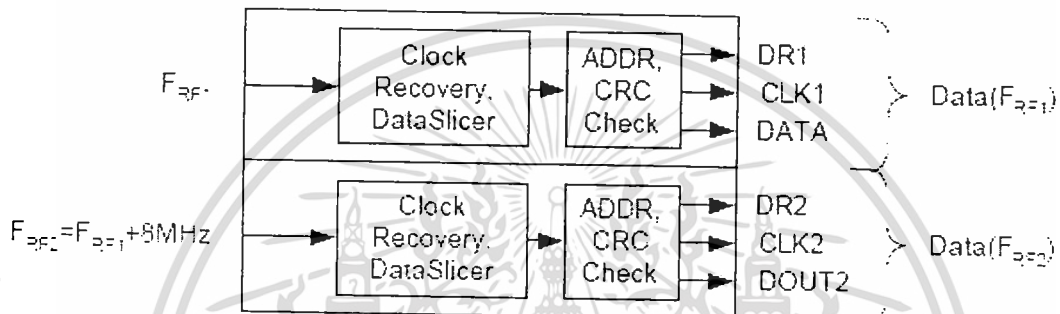
เทคโนโลยี DUOCiever ให้ 2 ช่องสัญญาณข้อมูลที่แยกกันอย่างละเอียดสำหรับ Rx และตอบสนองความต้องการต่อ 2 channel โดยมีระบบรับเพียง 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.17 Simultaneous 2 channel receiver on nRF24E1

สำหรับ nRF24E1 จะสามารถรับที่ช่องสัญญาณข้อมูลช่องที่ 2 ได้ นั่น ความถี่ช่องสัญญาณ ต้องอยู่สูงกว่าอยู่ 8 MHz ความถี่ของช่องสัญญาณข้อมูล nRF2401 ต้อง program เพื่อรับข้อมูลที่มีความถี่ของช่องสัญญาณที่ 1 ไม่มีการใช้เวลา multiplexing เพื่อทำฟังก์ชันนี้ให้สำเร็จ ใน Direct Mode ถ้ามันไม่รับส่งได้พร้อมกันระหว่าง 2 ช่องสัญญาณข้อมูล CPU ต้องสามารถจัดการข้อมูลที่เข้ามาพร้อมกันได้ ส่วนใน Shock Burst สามารถทำให้ CPU หยุดจับเวลาข้อมูลช่วงนี้ ขณะที่ไม่มีเกิดการสูญเสีย package ข้อมูลและไม่ลดประสิทธิภาพของ CPU



รูปที่ 2.18 DUOCiever โดยมี 2 ช่องสัญญาณรับข้อมูลที่เป็นอิสระต่อกัน

2.3.5 Device configuration

Configuration ทั้งหมดของระบบย่อย nRF2401 จะสามารถทำผ่านทาง 3-wire interface ของการติดต่อสื่อสาร (CS, CLK1, DATA) กับ register เดียว โดย configuration word สามารถมีได้ถึง 18 bytes ส่วน configuration bit (DATA) จะต้องจับเวลาโดย clocked (CLK1) โดย msb มาก่อน ขณะที่ CS = 1 ไม่เกิน 18 bytes อาจถูกควาน์โหลด

2.3.5.1 Configuration for ShockBurst operation (Configuration สำหรับ การดำเนินการ ShockBurst)

Configuration word ใน Shock Burst ทำให้ระบบย่อย nRF2401 สามารถจัดการกับ RF protocol ได้ หาก protocol สมบูรณ์ และโหลดไประบบย่อย nRF2401 เพียง 1 byte :bit[7:0] จำเป็นที่จะต้อง update ตลอดช่วงที่ดำเนินการ

Configuration block โดยละเอียด สำหรับ Shock Burst ดังนี้

- ความกว้างของ payload: ระบุจำนวนของบิต payload ใน RF package ซึ่งมันทำให้ระบบย่อย nRF2401 สามารถแบ่งแยกระหว่างข้อมูล payload และ CRC bytes ในตัวรับ package ได้

- ความกว้าง address: set จำนวนบิตที่แอดเดรสใช้ใน RF package ซึ่งมันทำให้ระบบย่อย nRF2401 สามารถแบ่งแยะระหว่างแอดเดรสและข้อมูล package ได้

- แอดเดรส (Rx ช่อง 1 และ 2) : กำหนดแอดเดรสสำหรับข้อมูล
- CRC : สามารถทำให้ on-chip CRC generate และถอดรหัสได้

Configuration Block นี้ โดยการยกเว้น CRC จะละเอียดสำหรับ package ซึ่งระบบ nRF2401 ที่จะรับในโหมด Tx CPU ต้อง generate แอดเดรส และส่วน payload ให้เหมาะกับระบบย่อย nRF2401 เพื่อจะรับข้อมูลเมื่อใช้ระบบย่อย nRF2401 on-chip รูปแบบ CRC ต้องแน่ใจว่า CRC นั้นใช้ได้และใช้ที่ความยาวเดียวกันทั้งอุปกรณ์ Tx และ Rx

PRE-AMBLE	ADDRESS	PAYLOAD	CRC
-----------	---------	---------	-----

รูปที่ 2.19 DATA package set-up

2.3.5.2 Configuration for Direct Mode operation (Configuration สำหรับการดำเนินการใน Direct Mode)

สำหรับใน Direct mode จะดำเนินการเพียงแคใน 2 ไบต์แรกของ Configuration word ที่เกี่ยวข้อง

ตารางที่ 2.8 ตารางของ Configuration word Configuration word จะ shift MSB ก่อนที่ clock ขอบขาขึ้น configuration ใหม่จะเกิดที่ขอบขาลงของ CS

	Bit position	Number of bits	Name	Function
Shock Burst™ configuration	143:120	24	TEST	Reserved for testing
	119:112	8	DATA2_W	Length of data payload section RX channel 2
	111:104	8	DATA1_W	Length of data payload section RX channel 1
	103:64	40	ADDR2	Up to 5 byte address for RX channel 2
	63:24	40	ADDR1	Up to 5 byte address for RX channel 1
	23:18	6	ADDR_W	Number of address bits (both RX channels).
	17	1	CRC_L	8 or 16 bit CRC
	16	1	CRC_EN	Enable on-chip CRC generation/checking.
General device configuration	15	1	RX2_EN	Enable two channel receive mode
	14	1	CM	Communication mode (Direct or ShockBurst™)
	13	1	RFDR_SB	RF data rate (1Mbps requires 16MHz crystal)
	12:10	3	XO_F	Crystal frequency
	9:8	2	RF_PWR	RF output power
	7:4	4	RF_CH#	Frequency channel
	0	1	RXEN	RX or TX operation

2.3.6 Configuration Word Detailed Description (อธิบายรายละเอียดของ configuration word)

อธิบายฟังก์ชันของ 144 บิต (บิต 143 = MSB) ซึ่งใช้ configure ระบุ nRF2401

Configuration อุปกรณ์ทั่วไป: bit [15:0]

Configuration Shock Burst: bit [119:0]

Test Configuration: bit [143:120]

ตารางที่ 2.9 Configuration data word

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MSB	TEST							
D143	D142	D141	D140	D139	D138	D137	D136	
Reserved for testing								
0	0	0	0	1	1	1	0	Default

MSE	TEST																
D135	D134	D133	D132	D131	D130	D129	D128	D127	D126	D125	D124	D123	D122	D121	D120		
Reserved for testing															Clear PLL m TX		
0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	0	Default

DATA2 W								
D119	D118	D117	D116	D115	D114	D113	D112	
Data width channel=2 m = of bit: excluding addr crc								
0	0	1	0	0	0	0	0	Default

DATA1 W								
D111	D110	D109	D108	D107	D106	D105	D104	
Data width channel=1 m = of bit: excluding addr crc								
0	0	1	0	0	0	0	0	Default

ADDR2													
D103	D102	D101	...	D71	D70	D69	D68	D67	D66	D65	D64		
Channel=2 Address RX (up to 40bit)													
0	0	0	...	1	1	1	0	0	1	1	1	1	Default

ADDR1													
D63	D62	D61	...	D31	D30	D29	D28	D27	D26	D25	D24		
Channel=1 Address RX (up to 40bit)													
0	0	0	...	1	1	1	0	0	1	1	1	1	Default

ADDR W						
D23	D22	D21	D20	D19	D18	
Address width m = of bit (both channel)						
0	0	1	0	0	0	Default

CRC				
D17		D16		
CRC Mode 1 = 16bit, 0 = 3bit		CRC 1 = enable, 0 = disable		
0	0	1	1	Default

RF-Programming															MSB	
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	
Two Ch	BUF	OD	NO Frequency			RF Power			Channel selection					RXEN		
0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	Default

โดย MSB bit ควรจะถูกโหลดสู่ configuration register เป็นอันดับแรก

Default Configuration word: h8E08.1C20.2000.0000.00E7.0000.0000.E721.0F04.

2.3.6.1 Shock Burst configuration:

ช่วงบิต [119:16] ประกอบด้วย segments อย่างละเอียดของ configuration register ไปยัง protocol การทำงานของ Shock Burst บิตรวมของ 120 บิตต้อง shift เข้าเพื่อจะ switch ระหว่าง Rx และ Tx

2.3.6.1.1 PLL_CTRL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.10 PLL setting

PLL_CTRL		
D121	D120	PLL
0	0	Open TX Closed RX
0	1	Open TX Open RX
1	0	Closed TX Closed RX
1	1	Closed TX Open RX

Bit 121-120:

PLL_CTRL ควบคุมการ set ของ PLL ใน Tx จะไม่เกิดการเบี่ยงเบน สำหรับ โหมดดำเนินการปกติ 2 บิตนี้ต้องเป็น bit low

2.3.6.1.2 DATAx_W

ตารางที่ 2.11 จำนวนของบิตใน payload

DATA2_W							
119	118	117	116	115	114	113	112

DATA1_W							
111	110	109	108	107	106	105	104

Bit 119-112:

DATA2_W ความยาว package ของ RF payload สำหรับสัญญาณช่อง 2

Bit 111-104:

DATA1_W ความยาว package ของ RF payload สำหรับสัญญาณช่อง 1

NOTE: จำนวนบิตทั้งหมดใน Shock Burst RF package อาจไม่เกิน 256! ความยาวสูงสุดของส่วน payload หาได้จาก $DATAx_W (bits) = 256 - ADDR_W - CRC$ โดย

ADD_W: ความยาวแอดเดรสของ Rx ที่ set ใน configuration word Bit[23:18]

CRC: Check sum 8 หรือ 16 บิต ที่เซตใน configuration word Bit[17]

PRE: Preamble 8 บิตซึ่งรวมอัดโนมัติ

แอดเดรส และ CRC ที่สั้นขึ้น จะทำให้มีที่สำหรับ payload data มากขึ้น

2.3.6.1.3 ADDR_x

ตารางที่ 2.12 แอดเดรสของตัวรับ 1 และ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADDR2											
105	102	101	.	71	70	69	68	67	66	65	64

ADDR1											
63	62	61	.	31	30	29	28	27	26	25	24

Bit 103-64:

ตัวรับ ADDR2 แอดเดรส channel2 มีได้ถึง 40 บิต

Bit 63-24:

ตัวรับ ADDR1 แอดเดรส channel1 มีได้ถึง 40 บิต

NOTE: บิตใน ADDR_x มากกว่าความกว้างที่ set ใน ADDR_W ซึ่งยืดยาวเกินไปและสามารถ set เป็น logic 0 ได้

2.3.6.1.4 ADDR_W & CRC

ตารางที่ 2.13 จำนวนบิตที่ต้องคงไว้สำหรับ RX address + CRC setting

ADDR_W						CRC_L	CRC_EN
23	22	21	20	19	18	17	16

Bit 23-18:

ADDR_W : จำนวนบิตที่ต้องสงวนไว้สำหรับแอดเดรส Rx ใน Shock Burst

NOTE: จำนวนบิตแอดเดรสสูงสุดคือ 40 บิต (5 ไบต์) ค่าที่มากกว่า 40 บิต ADDR_W จะไม่สามารถใช้ได้

Bit 17:

CRC_L: ความยาว CRC ที่จะคำนวณใน Shock Burst

Logic 0: 8 bit CRC

Logic 1: 16 bit CRC

Bit 16:

CRC_EN: ทำให้ CRC on-chip Tx และ Rx ใช้ได้

Logic 0: on-chip CRC generation / checking disable

Logic 1: on-chip CRC generation / checking enable

NOTE:

บิต 8 CRC อาจเพิ่มจำนวน payload ใน Shock Burst ได้แต่จะลดความสมบูรณ์

ของระบบลง

2.3.6.2 General RF configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

General RF configuration ของ configuration word จะควบคุม RF และ พารามิเตอร์ที่เกี่ยวข้องกับอุปกรณ์

2.3.6.2.1 Modes

ตารางที่ 2.14 RF operational setting

RX2_EN	CM	RFDR_SB	XO_F			RF_PWR	
15	14	13	12	11	10	9	8

Bit 15: Rx2_EN

Logic 0: Channel 1 รับ

Logic 1: Channel 2 รับ

NOTE: ในตัวรับทั้ง 2 channel nRF 24E1 รับทั้ง 2 channel แยกความถี่ของช่องสัญญาณทันที ความถี่ของตัวรับ channel 1 จะ set ในบิต [7-1] ของ configuration word ตัวรับ channel 2 จะเหนือตัวรับ channel 1 อยู่ 8 channel (8 MHz)

Bit 14:

Communication Mode:

Logic 0: ระบบย่อย nRF 2401 ดำเนินการใน Direct Mode

Logic 1: ระบบย่อย nRF 2401 ดำเนินการใน Shock Burst Mode

Bit 13:

RF Data Rate:

Logic 0: 250 kbps

Logic 1: 1 Mbps

NOTE: ใช้ 250 kbps แทน 1 Mbps จะช่วยปรับปรุง sensitivity ของตัวรับ 10 db 1Mbps ต้องใช้ crystal 16 MHz

Bit 12-10:

XO_F: เลือกความถี่ crystal ที่จะใช้

ตารางที่ 2.15 crystal frequency setting

XO FREQUENCY SELECTION			
D12	D11	D10	Crystal Frequency [MHz]
0	0	0	4
0	0	1	8
0	1	0	12
0	1	1	16
1	0	0	20

Bit 9-8:

RF_PWR set nRF 24E1 RF กำลังของเอาต์พุตในโหมดการส่ง

ตารางที่ 2.16 RF output power setting

RF OUTPUT POWER		
D9	D8	P [dBm]
0	0	-20
0	1	-10
1	0	-5
1	1	0

2.3.6.2.2 RF channel & direction

ตารางที่ 2.17 ช่องความถี่ และการ set Rx / Tx

RF_CH#							RXEN
7	6	5	4	3	2	1	0

Bit 7-1:

RF_CH# ตั้งความถี่ช่องสัญญาณของ nRF 24E1 ให้ดำเนินการความถี่ของช่องสัญญาณในการส่งได้โดย

$$\text{Channel RF} = 2400 \text{ MHz} + \text{RF_CH\#} * 1.0 \text{ MHz}$$

RF_CH#: อาจตั้งไว้ระหว่าง 2400 MHz และ 2527 MHz

NOTE: ช่วงที่มากกว่า 33 สามารถใช้ได้เพียงแคในอาณาเขตที่แน่ใจได้

ความถี่ช่วงสัญญาณในช่องข้อมูล 2 หาได้โดย

$$\text{Channel RF} = 2400 \text{ MHz} + \text{RF_CH\#} * 1.0 \text{ MHz} + 8 \text{ MHz} \text{ (รับที่ PIN\#4)}$$

RF_CH#: อาจตั้งไว้ระหว่าง 2400 MHz และ 2527 MHz

Bit 0:

Active Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Logic 0: transmit mode โหมดส่ง

Logic 1: receive mode โหมดรับ

2.3.7 Data package Description

PRE-AMBLE	ADDRESS	PAYLOAD	CRC
-----------	---------	---------	-----

รูปที่ 2.20 Data package Diagram

ชุดข้อมูลสำหรับการสื่อสารใน Shock Burst mode และ Direct mode จะแบ่งเป็น 4 ส่วน ดังนี้

ตารางที่ 2.18 รายละเอียดของ Data package

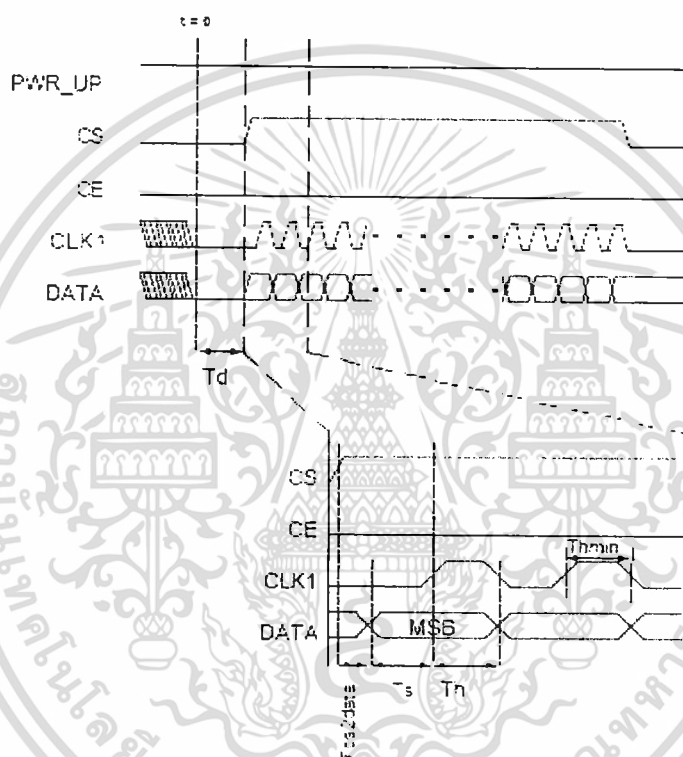
1. PREMBLE	<ul style="list-style-type: none"> - ส่วน preamble ต้องมีใน Shock Burst mode และ Direct mode - Preamble มีความยาว 8 บิตและ ขึ้นกับบิตแรกของแอดเดรส <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 20px;">PREMBLE</td> <td>1st ADDR_BIT</td> </tr> <tr> <td>01010101</td> <td>0</td> </tr> <tr> <td>10101010</td> <td>1</td> </tr> </table> <ul style="list-style-type: none"> - Preamble จะใส่ไปในชุดข้อมูลโดยอัตโนมัติและดังนั้นจึงมีที่ว่างพิเศษสำหรับ payload ใน Direct mode MCU ต้องควบคุม preamble <p>ใน ShockBurst mode Rx preamble จะถูกย้ายจากรับข้อมูลเอาท์พุท ใน Direct mode preamble จะเห็นได้ชัดเจนสำหรับข้อมูลเอาท์พุท</p>	PREMBLE	1 st ADDR_BIT	01010101	0	10101010	1
PREMBLE	1 st ADDR_BIT						
01010101	0						
10101010	1						
2. ADDRESS	<ul style="list-style-type: none"> - ส่วนของแอดเดรสต้องมีใน ShockBurst โหมด และ Direct โหมด - ความยาว 80 – 40 บิต - แอดเดรสจะถูกย้ายโดยอัตโนมัติจากชุดรับใน ShockBurst mode <p>ใน Direct mode MCU ต้องควบคุมแอดเดรส</p>						
3. PAYLOAD	<ul style="list-style-type: none"> - ข้อมูลถูกส่ง - ใน ShockBurst โหมด payload มีขนาดน้อยที่สุด 256 บิต (แอดเดรส 8-40 บิต +CRC 8หรือ16 บิต - ใน Direct mode ความยาวมากที่สุดสำหรับ 1 Mbps คือ 4000 บิต 						

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. CRC	<ul style="list-style-type: none"> - CRC จะเลือกได้ใน ShockBurst โหมด และไม่ใช่ใน Direct โหมด - ความยาว 8 หรือ 16 บิต - ShockBurst Rx CRC จะถูกย้ายจากรับข้อมูลเอาที่พูด
--------	---

2.3.8 Configuration mode timing

เมื่อ 1 บิต หรือ มากกว่า 1 บิต ใน configuration word จำเป็นที่จะต้องเปลี่ยนแปลง เราจะไปดูได้จาก timing

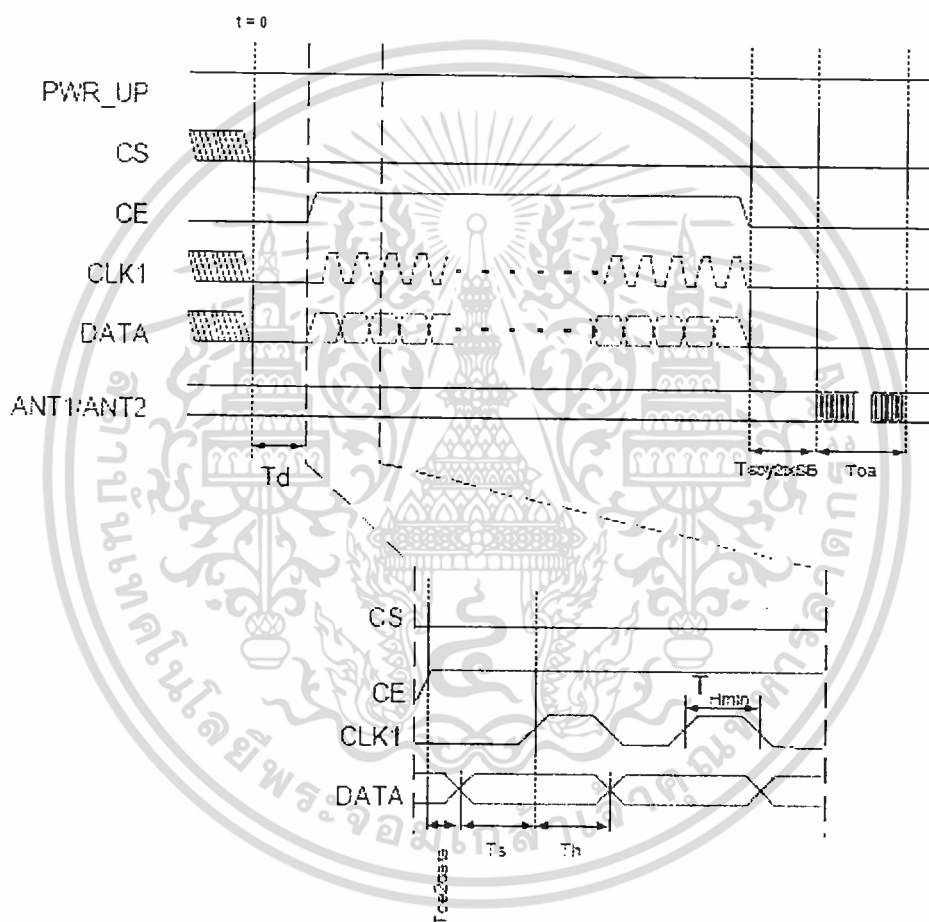


รูปที่ 2.21 Timing Diagram สำหรับ configuration ระบบย่อย nRF2401

ถ้า configuration mode เข้ามาจาก power down CS สามารถ set high หลังจาก T_{pd2sby} โดยแสดงให้เห็นในรูปที่ 2.20

2.3.9 ShockBurst Mode Timing

ShockBurst TX:



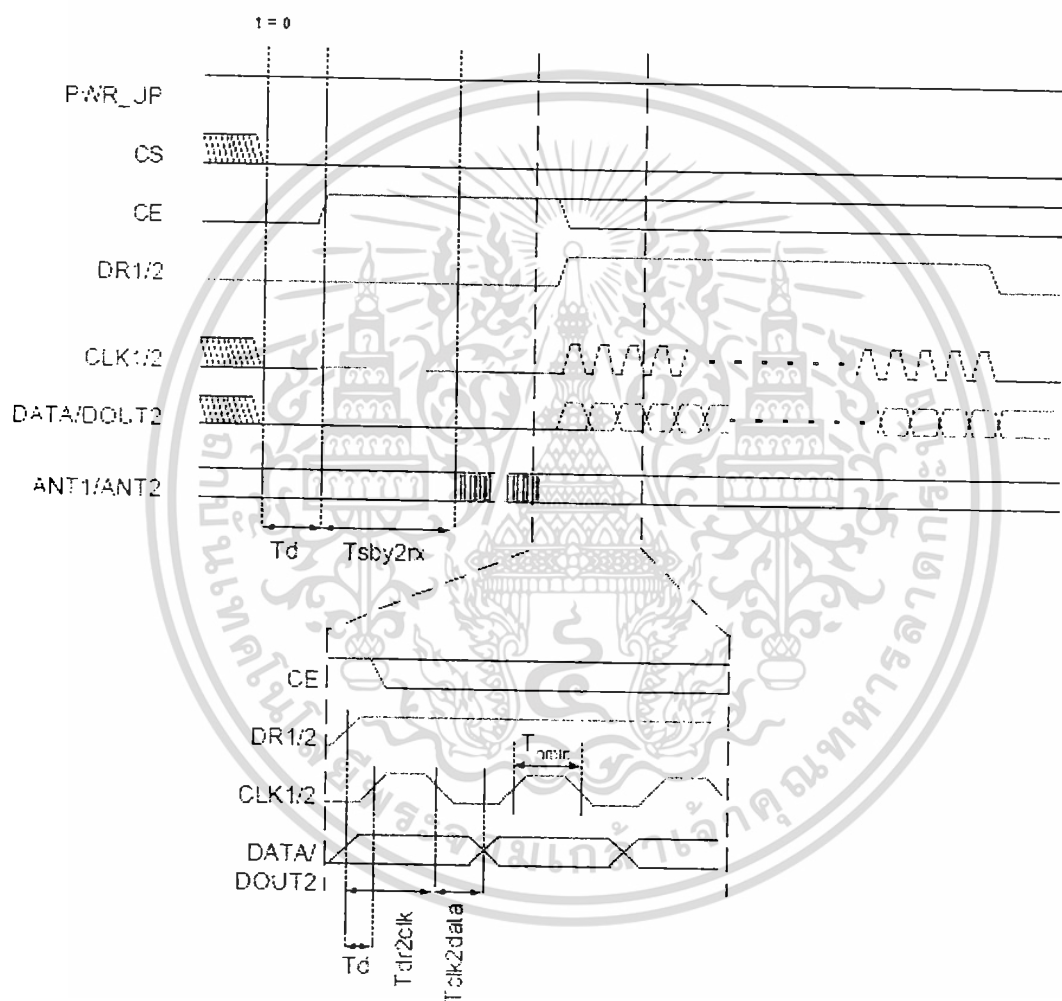
รูปที่ 2.22 Timing ของ ShockBurst ใน TX

ความยาวชุดข้อมูลและอัตราข้อมูลทำให้ delay T_{oa} (Time on air : เวลาในอากาศ) ซึ่งแสดงในสมการ บิตข้อมูลจะเป็นผลรวมของบิตที่รวมทุกๆบิต CRC และบิต preamble ซึ่งอาจเพิ่มขึ้น

$$T_{oa} = 1/\text{data rate} * (\#\text{databits} + 1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ShockBurst Rx:



รูปที่ 2.23 Timing ของ ShockBurst ใน Rx

CE อาจยังคงสถานะ high ช่วงที่ดาวน์โหลดข้อมูล แต่จะทำให้ค่าการใช้กระแสสูงกว่า (19mA) และผลประโยชน์คือ เวลา start-up น้อย (200us) เมื่อ DR1 ขาลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51

จากโครงการจะทำการต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ที่พอร์ต 1 โดยใช้สายทั้งหมด 7 เส้น เป็นสายของคอลัมน์ 3 เส้นและสายของโรว 4 เส้น โดยไมโครคอนโทรลเลอร์จะทำการส่งข้อมูล "0" ไปยัง P1.4 – P1.6 ตามลำดับ ในทุกครั้งที่มีการส่งข้อมูลไปยังสายคอลัมน์ของคีย์แพด ไมโครคอนโทรลเลอร์จะทำการอ่านค่าที่ P1.0 – P1.3 เข้ามาด้วย หากไม่มีการกด ค่าของ P1.0 – P1.3 ก็จะเป็น "1" ทั้งหมด ถ้าหากมีการกดคีย์ ค่าของ P1.0 – P1.3 ก็จะไม่เป็น 1111 อีกต่อไป เป็นการแจ้งให้ทราบว่า มีการกดคีย์แพดขึ้นแล้ว จากนั้นไมโครคอนโทรลเลอร์ก็จะทำการค้นหาตำแหน่งต่อไป โดยการค้นหาตำแหน่งนั้น สิ่งที่จะได้มาอย่างแรกคือ ค่าตำแหน่งของคีย์นั้น จากนั้นก็จะนำค่าตำแหน่งนั้นไปเปิดตารางข้อมูล เพื่อที่จะได้ค่าที่ต้องการนำไปแสดงผลที่แท้จริง

2.5 การเชื่อมต่อกับโมดูลแอลซีดี(LCD Module)

ในโมดูลแอลซีดีนั้นจะมีส่วนประกอบ 3 ส่วนหลักดังนี้

- ตัวแสดงผล(Display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นได้โดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงบนจอแอลซีดี
- ตัวควบคุม(Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอก มาควบคุมการทำงานของโมดูลแอลซีดี เช่น ลบจอภาพ แสดงตัวอักษรหรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะชิป ที่นิยมใช้คือเบอร์ HD44780 และ HD61830 โดย HD44780 จะใช้ควบคุม LCD แบบอักขระ และ HD61830 ใช้ควบคุม LCD แบบกราฟฟิก
- ตัวขับ(Driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด ชิปที่ใช้ทำงานเป็นตัวขับนี้ได้แก่ เบอร์ HD44100H และ MSMS5259 เป็นต้น

2.5.1 โครงสร้างภายในของตัวควบคุมโมดูล LCD

ในการใช้งาน โมดูล LCD จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งในการควบคุมให้ดีเสียก่อน โดยในที่นี้จะทำการแสดงตัวอย่าง โมดูล LCD แบบอักขระ เพราะสามารถเข้าใจง่าย โดยบล็อกไดอะแกรมภายในของชิปควบคุม LCD เบอร์ HD44780H ซึ่งใช้ในโมดูล LCD แบบอักขระ ประกอบด้วย

บัฟเฟอร์อินพุตเอาต์พุต เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

รีจิสเตอร์คำสั่ง (Instruction Register:IR) เป็นรีจิสเตอร์ที่ใช้รับข้อมูลจากอุปกรณ์ภายนอก เพื่อถ่ายทอดไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

แรมเก็บข้อมูลแสดงผล(Display Data RAM:DDRAM) เป็นหน่วยความจำแบบแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (Look up table) ของตัวอักษรที่เก็บไว้ในหน่วยความจำรวมและแรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

รวมเก็บตัวอักษร (Character Generator ROM:CGROM)เป็นหน่วยความจำแบบรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงที่ตัวแสดงผลได้ มีขนาด 7200 บิต โดยจะถูกอ่านด้วยของข้อมูลใน DRAM

แรมเก็บตัวอักษร(Character Generator RAM:CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิต การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือ เขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGROM เอง

แฟล็กบัสซี(Busy Flag) เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่าตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนที่จะส่งข้อมูลหรือคำสั่งมายังโมดูล LCD ขนาด 20 ตัวอักษร 4 บรรทัด

สำหรับโมดูล LCD ที่ใช้ในโครงการ เป็นขนาด 20 ตัวอักษร 4 บรรทัด เนื่องจากเป็นขนาดที่เหมาะสมที่สุดในการใช้การเป็นจอแสดงผลสำหรับการสั่งอาหารซึ่งพิจารณาจากประโยชน์ใช้สอยและความประหยัด ซึ่งประกอบด้วยขาทั้งหมด 14 ขา มีการจัดวางตามรูป

รูปที่ แสดงการจัดขาของโมดูลซีดีแบบอักษร

Vss(ขา1) : ต่อกราวด์

Vdd(ขา2) : ต่อไฟเลี้ยง +5 v

Vo(ขา4) : เป็นอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS(ขา4) : เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยขานี้เป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่งแต่ถ้าขาเป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

R/W(ขา5) : เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลจากโมดูล LCD ถ้าเป็น “0” เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

E(ขา6) : เป็นขาสำหรับรับสัญญาณพัลส์เอ็นเอเบิลโมดูล LCD ให้ทำงาน

D0-D7 : เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต

(ขา 7-14) : หนึ่งขา RS,R/W และ E จะทำงานร่วมกัน โดยมีความสัมพันธ์แสดงดังตาราง

2.5.2 คำสั่งควบคุมโมดูล LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเขียนคำสั่งลงในตัวควบคุม แน่ใจว่าต้องกำหนดให้ขา RS และ R/W เป็น “0” แล้วเขียนคำสั่งตามไป คำสั่งควบคุมโมดูล LCD ของชิปควบคุม HD44780 ที่สำคัญมี 10 คำสั่งดังนี้

1. คำสั่งเคลียร์ตัวแสดงผล(Clear display) มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่าง หรือ space เข้าไปใน DDRAM ทั้งหมด เมื่อตัวควบคุมเอ็กซิวต์คำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซ็ทบิต I/D ให้เป็น “1”
2. คำสั่ง(Return home) ต้องกำหนดให้บิต 1 ของข้อมูลเป็น “1” เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลง นั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้
3. คำสั่งเลือกโหมดการป้อนข้อมูล (Entry mode set) มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต
0							
0	0	0	0	0	1	I/D	S

- บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูล ถ้าหากบิต S เป็น “1” เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต่ถ้าหากบิตนี้เป็น “0” เมื่อเกิดข้อมูลใหม่ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ
- บิต I/D เป็นบิตที่ใช้กำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว แอดเดรสของ DDRAM เพิ่มขึ้น หรือลดลงหนึ่งแอดเดรส โดยถ้าบิตนี้เป็น “1” แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น “0” แอดเดรสจะลดลง ดังนั้นข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H-07H (4 ข้อมูลคำสั่งและที่ใช้บ่อยคือ 06H หมายถึง กำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือ และแอดเดรสของ DDRAM เพิ่มขึ้น

2.5.3 คำสั่งควบคุมการแสดงผล

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต
0							
0	0	0	0	0	D	C	B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าบิตนี้เป็น "1" จะเป็นการเปิดจอแสดงผล ถ้าเป็น "0" จะเป็นการปิดจอแสดงผล
- บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงผลบนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น "1" ถ้ากำหนดให้เป็น "0" จะเป็นการปิดเคอร์เซอร์ หรือไม่แสดงเคอร์เซอร์
- บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น "1" เคอร์เซอร์จะกระพริบดังนั้น จะมีข้อมูลคำสั่งได้ตั้งแต่ 08H-0FH ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0FH เป็นการสั่งให้เปิดจอแสดงผล แสดงเคอร์เซอร์ และสั่งให้เคอร์เซอร์กระพริบ

2.5.4 คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต
0	0	0	1	S/C	R/L	*	*

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผล ขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H-17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H-1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1CH-1FH

2.5.5 คำสั่งกำหนดฟังก์ชันการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต
0	0	1	DL	N	F	*	*

- บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น "0" จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าบิตนี้เป็น "1" จะเป็นแบบ 8 บิต
- บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าเป็น "0" จะแสดงผล 1 บรรทัด ถ้าเป็น "1" จะแสดงผล 2 บรรทัด ในกรณี ในกรณีที่จอแสดงผลสามารถแสดงได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า 2 บรรทัด ก็กำหนดบิต N นี้ให้เป็น "1" จุดที่น่าสังเกตคือ โมดูล LCD แบบ 16 ตัวอักษร 1 บรรทัด แม้จะมีบรรทัดการแสดงผลเพียง 1 บรรทัด แต่จะต้องกำหนด N ให้เป็น "1" เนื่องจากแอดเดรสของ DDRAM แบ่งเป็น 2 ช่อง คือ 00H และ 40H
- บิต F ใช้เลือกความละเอียดของตัวอักษรในการแสดงผล ถ้าบิตนี้เป็น "0" จะเป็นการแสดงผลแบบ 5x7 และถ้าเป็น "1" จะแสดงเป็นแบบ 5x10 จุด ข้อมูลคำสั่งที่ใช้บ่อยคือ 38H เป็นการกำหนดให้โมดูล LCD ทำงานในแบบ 8 บิต แสดงผล 4 บรรทัด และเลือกความละเอียดเป็น 5x7 จุด

คำสั่งเลือกแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น "0" บิต 6 เป็น "1" ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ ก่อนจะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00H-3FH

2.5.6 คำสั่งเลือกแอดเดรสของ DDRAM

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิต 7 ต้องเป็น "1" และข้อมูลอีก 7 บิตที่เหลือจะเป็นคู่แอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH-0FFH ทั้งนี้จำนวนแอดเดรสวิ่งขึ้นกับการกำหนดสถานะบิตที่ N ด้วย เป็น "0" แอดเดรสของ DDRAM จะอยู่ระหว่าง 80H-0CFH และถ้าบิต N เป็น "1" แอดเดรสของ DDRAM จะมีสองช่วงคือ 8CH-87H และ 0C0H-0C7H

2.5.7 คำสั่งอ่านบิตซีแฟลคและแอดเดรส

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต
0							
BF	A	A	A	A	A	A	A

เป็นคำสั่งที่ใช้อ่านบิตซีเฟลก(BF) โดยเฟลกนี้จะเป็นตัวบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น “0” แสดงว่าพร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น “1” แสดงว่า ขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง เมื่อต้องการอ่านค่าเฟลกต้องกำหนดให้ขา R/W เป็น “1” ด้วย แต่สัญญาณที่ RS ยังต้องเป็น “0” อยู่เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง นอกจากนี้ ยังใช้เป็นคำสั่งอ่านข้อมูล แอดเดรสของ CGRAM และ DDRAM ด้วย

2.6 Delphi 7

Delphi 7 คือ ซอฟต์แวร์ที่นำมาใช้ในการเขียนโปรแกรมเพื่อสร้างแอปพลิเคชัน หรือซอฟต์แวร์อีกที โดยมันจะประกอบไปด้วยเครื่องมือชนิดต่างๆที่ใช้ให้การเขียนโปรแกรมทำได้สะดวก

Delphi 7 จัดเป็นเครื่องมือเขียนโปรแกรมชนิด Visual Programming เช่นเดียวกับ Visual Basic หรือ Visual C++ โดยมีข้อดีคือ สามารถเขียนโปรแกรมได้ง่าย และให้ผลงานออกมาอย่างรวดเร็ว ซึ่งจะแตกต่างจากเครื่องมือเขียนโปรแกรมรุ่นเดิมๆ เช่น Turbo Pascal หรือ Borland C ที่มีความยุ่งยากในการใช้งานและการเรียนรู้ในการเขียนโปรแกรม ดังนั้นจึงจัดให้ Delphi 7 เป็นซอฟต์แวร์ประเภท RAD หรือ Rapid Application Development ซึ่งแปลว่าสามารถสร้างแอปพลิเคชันได้อย่างรวดเร็ว

Delphi 7 นั้นผ่านการพัฒนามาเกือบ 10 ปี ตั้งแต่เวอร์ชัน 1.0 ที่ทำงานบน Windows 3.1X โดยมีจุดเด่นหลายๆ ตั้งแต่สมัยนั้นคือ โปรแกรมที่ได้จากการเขียนโปรแกรมมีขนาดเล็ก ทำงานได้รวดเร็ว ซึ่งมักจะถูกนำไปเปรียบเทียบกับ Visual Basic 3.0 ในสมัยนั้น อีกประการหนึ่ง Delphi ใช้ภาษาออบเจกต์ปาสคาล จึงเคยถูกเปรียบว่าเป็น Visual Pascal มาแล้ว

เวอร์ชันปัจจุบันของ Delphi นั้นได้รับการพัฒนาให้สามารถสร้างแอปพลิเคชันที่ทำงานบน Windows ได้ดีเหมือนเดิม โดยมีการปรับปรุงให้สามารถพัฒนาแอปพลิเคชันตามแนวความคิดของ .NET ซึ่งจะช่วยให้สามารถเขียนโปรแกรมครั้งเดียว แล้วนำไปใช้งานบนอุปกรณ์ต่างๆไม่ว่าจะเป็น PDA , โทรศัพท์มือถือ และ บนเว็บได้

2.6.1 ความสามารถของ Delphi 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Delphi 7 นั้นมีความสามารถมากมาย ได้รับการต้อนรับเป็นอย่างดีจากนักพัฒนาแอปพลิเคชันทั่วโลก รวมทั้งเมืองไทยด้วย ซึ่งจะเห็นได้จากการนำ Delphi ไปประกอบการเรียนการสอน การฝึกอบรม ตลอดจนการนำไปสร้างเป็นซอฟต์แวร์เชิงพาณิชย์จำนวนมาก

2.6.1.1 สร้างแอปพลิเคชันสำหรับ Windows

Delphi ได้ชื่อว่าเป็นเครื่องมือสำหรับพัฒนาแอปพลิเคชันที่ทำงานบน Windows ที่ใช้งานกันแพร่หลายมาก สามารถสร้างงานได้อย่างรวดเร็ว นิยมนำไปสร้างแอปพลิเคชันทั้งเพื่อการศึกษา และแอปพลิเคชันที่ใช้ในระบบงานจริงๆ ในโลกธุรกิจ

Delphi สามารถสร้างแอปพลิเคชันบน Windows ได้หลายเวอร์ชัน ซึ่งแอปพลิเคชันที่สร้างจาก Delphi ได้รับการยกย่องเป็นอย่างมากในด้านขนาด โปรแกรมที่เล็กกะทัดรัด และทำงานได้อย่างรวดเร็ว เมื่อเทียบกับแอปพลิเคชันที่สร้างจากเครื่องมือตัวอื่นๆ ทำให้ Delphi แต่ละเวอร์ชันได้รับรางวัลเกียรติยศจากสถาบันต่างๆ ทั่วโลกมาโดยตลอด

2.6.1.2 สร้างระบบงานด้านฐานข้อมูล

ถ้าจะถามว่า Delphi ถูกนำไปสร้างผลงานอะไรมากที่สุดสำหรับเมืองไทยแล้ว หนีไม่พ้นเรื่องของการพัฒนาระบบงานด้านฐานข้อมูล ซึ่งถูกนำไปใช้งานอย่างแพร่หลายตั้งแต่องค์กรขนาดเล็กไปจนถึงองค์กรระดับประเทศ โดย Delphi มีจุดเด่นในการสร้างแอปพลิเคชันที่ติดต่อกับฐานข้อมูลได้หลายรูปแบบ มีวิธีการและเครื่องมือในการสร้างระบบงาน ระบบบริหารงานบุคคล ระบบคลังสินค้า หรือแม้แต่ระบบจองตั๋วในโรงพยาบาลชั้นนำ

แอปพลิเคชันที่เกี่ยวกับฐานข้อมูลที่สร้างจาก Delphi สามารถนำไปใช้งานกับระบบฐานข้อมูลชั้นนำแทบทุกชนิดทั่วโลก นับตั้งแต่ระบบฐานข้อมูลส่วนบุคคลทั้ง Access , Paradox , Foxpro ไปจนถึงระบบฐานข้อมูลขนาดใหญ่ทั้ง Oracle , Sybase , SQL Server รวมถึงระบบไฟล์ชนิดต่างๆได้ด้วย

2.6.1.3 สร้างแอปพลิเคชันรองรับ .NET Web Service

แนวโน้มของการพัฒนาแอปพลิเคชันในยุคหน้าที่ต้องเตรียมตัวให้พร้อมก็คือ การพัฒนาแอปพลิเคชันด้วยเทคโนโลยี .NET ซึ่งจะช่วยให้แอปพลิเคชันชนิดต่างๆ ไม่ว่าจะทำงานบนพีซี โน้ตบุ๊ก PDA หรือแม้แต่โทรศัพท์มือถือ สามารถเชื่อมโยงข้อมูล และคุยกันได้

ใน Delphi 7 นั้นรองรับการพัฒนาแอปพลิเคชันที่เรียกว่า .NET Web Service แล้ว โดยสามารถใช้เครื่องมือชนิดต่างๆ พัฒนาแอปพลิเคชันตามแนวความคิดของ .NET เป็นอย่างดี

2.6.1.4 ขั้นตอนการเขียนโปรแกรมเพื่อสร้างแอปพลิเคชันกับ Delphi7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนในการสร้างแอปพลิเคชันด้วย Delphi 7 มีขั้นตอนดังต่อไปนี้

ขั้นที่ 1 : ออกแบบโปรแกรม

นับเป็นขั้นตอนแรกของการเขียนโปรแกรมที่ควรต้องทำ โดยผู้เขียนโปรแกรมจะต้องออกแบบเสียก่อนว่าจะให้โปรแกรมมีหน้าตาอย่างไร มีขั้นตอนการทำงานเป็นอย่างไร ซึ่งบ่อยครั้งที่ผู้เริ่มต้นเขียนโปรแกรมมือใหม่หลายรายละเลย ซึ่งจะส่งผลเสียในอนาคตคือ ถ้าออกแบบโปรแกรมไม่ดี (ขาดการทำงานที่ดี , ขาดความยืดหยุ่นในการปรับแต่ง , ขาดความสวยงามนำไปใช้งาน) ก็จะทำให้เกิดความยุ่งยากในการปรับปรุงโปรแกรมนี้อีกในอนาคต

ในการพัฒนาระบบงานขนาดใหญ่ขึ้น งานในการออกแบบโปรแกรมกับการเขียนโปรแกรมจะแยกจากกัน อาจจะใช้ทีมงานคนละทีมงานกันทำให้ผู้ออกแบบโปรแกรมต้องมีเครื่องมือที่จะใช้สื่อสาร และบอกรายละเอียดของการออกแบบให้กับผู้เขียนโปรแกรมได้ทราบ ซึ่งจะใช้สิ่งที่เรียกว่า โฟลว์ชาร์ต (Flow Chart) เป็นเครื่องมือในการสื่อสารนั้น

โฟลว์ชาร์ตนั้นจะบอกถึงขั้นตอนการทำงาน การตัดสินใจ และข้อมูลที่จะใช้ให้ผู้เขียนโปรแกรมทราบอย่างครบถ้วน เพราะฉะนั้นนอกจากจะใช้สื่อสารระหว่างกัน ยังใช้เป็นเครื่องมือสำคัญในการควบคุมการทำงานของผู้เขียนโปรแกรมได้ด้วย

ขั้นที่ 2 : วางคอม โปเนนต์ต่างๆลงใน Form Designer

ผู้เขียนโปรแกรมจะนำแบบที่คิดไว้จากขั้นที่ 1 มาสร้างเป็นแอปพลิเคชันไว้ใช้งาน

ขั้นที่ 3 : กำหนดค่าพรีอเพอร์ตีให้กับคอม โปเนนต์ต่างๆ

ในขั้นตอนนี้ผู้เขียนโปรแกรมจะกำหนดค่าพรีอเพอร์ตี หรือคุณสมบัติให้กับคอม โปเนนต์ต่างๆที่วางอยู่ใน Form Designer

ขั้นที่ 4 : เขียนคำสั่งกำกับการทำงานให้คอม โปเนนต์

หลังจากกำหนดค่าพรีอเพอร์ตี ซึ่งก็เป็นการกำหนดหน้าตา ลักษณะการทำงานต่างๆ แล้ว ต่อไปจะต้องเขียนคำสั่งเพื่อกำกับการทำงานของคอม โปเนนต์โดยเขียนคำสั่งผ่าน Code Editor

ขั้นที่ 5 : คอมไพล์โปรแกรมที่เขียนขึ้น

เมื่อเขียนคำสั่งเพื่อกำกับการทำงานในรูปแบบต่างๆจนครบแล้ว ต่อไปผู้เขียนโปรแกรมก็จะคอมไพล์โปรแกรมที่ได้เขียนขึ้น ซึ่งการคอมไพล์ (Compile) ก็คือ การแปลจากคำสั่งที่ได้เขียนทั้งภาษาไทย ภาษาอังกฤษที่ผู้เขียนโปรแกรมอ่านรู้เรื่องนำมาแปลเป็นภาษาที่คอมพิวเตอร์เข้าใจ และทำงานตามที่ผู้เขียนโปรแกรมตั้งใจไว้ได้

ในการคอมไพล์โปรแกรมที่เขียนขึ้นโดย Delphi 7 นั้นให้คลิกเมนู Run > Run หรือง่ายกว่านั้นให้กดปุ่ม<F9> แทนที่

Delphi 7 จะทำการคอมไพล์คำสั่งที่ได้เขียนขึ้นทั้งหมด แล้วเปลี่ยนสถานะ (Mode) การทำงานจากสถานะการออกแบบเขียนโปรแกรม (Design Mode) ไปสู่สถานะการรันโปรแกรม (Run Mode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 6 : ทดสอบการทำงานของแอปพลิเคชัน

เมื่อคอมไพล์ผ่านแล้ว ผู้เขียนโปรแกรมจะทดสอบการทำงานว่าสิ่งที่สร้างนั้นทำงานได้ถูกต้องหรือไม่ เพราะบางครั้งอาจจะเขียนคำสั่งได้ถูกต้องตามหลักภาษาโปรแกรมทุกอย่าง แต่อาจจะทำงานผิดพลาดก็ได้ จึงต้องตรวจสอบการทำงานให้แน่ใจเสียก่อน

ขั้นที่ 7 : บันทึกผลการทำงาน

เมื่อทดสอบจนเป็นที่น่าพอใจแล้ว ผู้เขียนโปรแกรมก็สามารถบันทึกสิ่งที่ได้สร้างขึ้นทั้งหมดเอาไว้ ซึ่งจะสามารถเรียกขึ้นมาแก้ไขปรับปรุงภายหลังได้ โดยจะบันทึกไว้ 2 ส่วนคือ ไฟล์ยูนิท (.pas) และ โปรเจกต์ (.dpr)

2.6.1.5 ภาษาออบเจกต์ปาสคาล

ภาษาออบเจกต์ปาสคาลในการเขียนโปรแกรมกับ Delphi 7 นั้นจะเก็บซอร์สโค้ดที่ได้เขียนขึ้นไว้ในไฟล์ที่เรียกว่า ยูนิท (Unit) ซึ่งจะถูกบันทึกไว้ในไฟล์ .pas และเมื่อไฟล์ยูนิทถูกคอมไพล์ก็จะสร้างไฟล์ที่เป็นผลจากการคอมไพล์มีนามสกุลเป็น .dcu

ปกติเมื่อเริ่มสร้างแอปพลิเคชันใน Delphi 7 (ก็คือการสร้างโปรเจกต์ใหม่ขึ้นมา) จะมีการสร้างไฟล์ชื่อ Unit1.pas ขึ้นมาให้เขียนโค้ดเข้าไป เมื่อตรวจสอบจาก Code Editor จะพบว่า Delphi ได้เขียนโค้ดบางส่วนไว้ให้ผู้เขียนโปรแกรมอยู่แล้ว

ไฟล์ยูนิทหนึ่งๆ จะใช้กับการเขียนโค้ดสำหรับฟอร์ม 1 ฟอร์ม ดังนั้นถ้าโปรเจกต์ที่ประกอบด้วย ฟอร์มมากกว่าหนึ่งฟอร์มสามารถจะมีไฟล์ยูนิทที่มีนามสกุล .pas ได้มากกว่า 1 ไฟล์ได้

เมื่อผู้เขียนโปรแกรมมีการนำคอมโพเนนต์เข้ายังฟอร์มก็จะพบว่า Delphi ได้เขียนโค้ดให้แล้วในตอนต้นของไฟล์ยูนิท และเมื่อเขียนโปรแกรมเพื่อเลือกจัดการกับอีเวนต์ที่สนใจ Delphi ก็เตรียมโค้ดที่เป็นโครงเอาไว้ให้ผู้เขียนโปรแกรมเขียนเพิ่มเติม

2.6.1.6 โครงสร้างของยูนิท

เมื่อกลับมาดูไฟล์ Unit1.pas ที่ Delphi สร้างไว้ให้ นั้น สามารถแบ่งโค้ดออกเป็นส่วนต่างๆ ดังนี้

- ส่วน interface ส่วนนี้เป็นส่วนที่ใช้ประกาศชนิดข้อมูล , ตัวแปร , ค่าคงที่ , ออบเจกต์ , โพรซีเจอร์ และฟังก์ชัน ซึ่งทุกสิ่งที่ยกมาไว้ในส่วนนี้สามารถเข้าถึง และใช้งานได้จากยูนิทอื่นๆ ต้องไม่ลืมว่าโปรเจกต์หนึ่งๆสามารถมีไฟล์ยูนิทได้หลายตัว ซึ่งพื้นที่ส่วน interface นี้จะเริ่มจากคำว่า interface ไปจนถึง implementation
- ส่วน implementation ส่วนนี้จะทำหน้าที่เหมือนกับส่วน Interface ต่างกันตรงที่ขอบเขตการเข้าถึงข้อมูลคือ จะเข้าถึงข้อมูล , ตัวแปร , ค่าคงที่ , ออบเจกต์ , โพรซีเจอร์ และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน ได้จากสิ่งที่อยู่ขอบเขตเฉพาะภายในยูนิตนี้เท่านั้น ยูนิตอื่นๆ ไม่มีสิทธิเข้าถึง ซึ่งพื้นที่ส่วน implementation ไปจนถึง initialization

- ส่วน initialization ส่วนนี้จะใช้เก็บคำสั่งที่ถูกเรียกใช้งานก่อนการทำงานของแอปพลิเคชัน โดยปกติจะทำงานก่อนที่จะมีการสร้างออบเจกต์ หรือฟอร์มขึ้นมา ดังนั้นผู้เขียนโปรแกรมจึงมักใช้พื้นที่ส่วนนี้กำหนดค่าให้กับตัวแปรบางตัว ซึ่งพื้นที่ของส่วน initialization นี้จะเริ่มต้นจากคำว่า initialization ไปจนถึง finalization (ถ้าไม่มีคำว่า finalization ก็จะถึงคำว่า end.) สำหรับส่วนนี้จะมีก็ได้ ไม่มีก็ได้แล้วแต่โปรแกรม
- ส่วน finalization ส่วนนี้จะทำหน้าที่ตรงกันข้ามกับ initialization ซึ่งพื้นที่ส่วน finalization นี้จะเริ่มตั้งแต่คำว่า finalization ไปจนถึง end.

2.6.1.7 ไฟล์โปรเจกต์

ปกติการเขียน โปรแกรมจะเขียนที่ไฟล์ยูนิตเท่านั้น แต่ในการทำงานจริงๆของแอปพลิเคชันนั้นจะต้องเริ่มจากการทำงานของไฟล์โปรเจกต์ก่อน ซึ่งรายละเอียดการทำงานของไฟล์โปรเจกต์นี้ผู้เขียน โปรแกรมไม่ต้องจัดการเอง แต่สามารถดูการทำงานภายในได้โดยเลือกเมนู Project > View Source ซึ่งใน Code Editor จะเพิ่มแท็บที่เป็นรายละเอียดของไฟล์โปรเจกต์ดังนี้

สำหรับโค้ดของไฟล์โปรเจกต์นั้นประกอบไปด้วยส่วนสำคัญ 3 ส่วนคือ

- Program heading
- Uses clause
- การทำงานของโปรแกรม

สำหรับส่วนที่เป็น program heading นั้นจะบอกถึงชื่อโปรแกรม ในที่นี้ก็คือชื่อที่ผู้เขียน โปรแกรมตั้งให้ในตอนบันทึกชื่อไฟล์โปรเจกต์นั่นเอง และชื่อนี้จะถูกสร้างเป็นไฟล์ .EXE ด้วย

ส่วน uses clause จะเป็นส่วนที่ใช้ประกาศยูนิตที่ใช้งาน ถ้ามีมากกว่าหนึ่งยูนิตในนี้ก็จะเก็บรายการของยูนิตทั้งหมดที่ใช้เอาไว้

ส่วนสุดท้ายจะเก็บรายละเอียดของคำสั่งในการทำงานของแอปพลิเคชัน ซึ่งทุกๆแอปพลิเคชันที่เป็นชนิดเดียวกันจะมีส่วนนี้เหมือนกันหมดคือ มีส่วนที่กำหนดค่าเริ่มต้น สร้างฟอร์มขึ้นมา แล้วเริ่มการทำงานให้กับแอปพลิเคชัน (ซึ่งถ้าเป็นแอปพลิเคชันทั่วไป ฟอร์มที่ถูกสร้างขึ้นมาก็จะรอการโต้ตอบจากผู้ใช้โปรแกรมงานนั่นเอง)

2.6.1.8 การเขียนโปรแกรมแบบ OOP

การเขียนโปรแกรม OOP เป็นการเขียนแบบแนวคิดของสิ่งที่ผู้เขียน โปรแกรมพบเห็น

ในสิ่งต่างๆที่อยู่รอบตัวทั้งนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คลาสกับออบเจกต์

แนวคิดของ OOP นั้นจะกำหนดให้สิ่งต่างๆเป็นวัตถุ หรือออบเจกต์ (Object) ซึ่งออบเจกต์แต่ละตัว จะถูกสร้างขึ้นมาจากแม่พิมพ์ที่เรียกว่า คลาส (Class) ดังนั้นออบเจกต์ที่สร้างจากคลาสจะถือว่าเป็นคนละตัวกัน แต่มีชนิดเดียวกัน เพราะสร้างจากคลาสเดียวกัน

เมื่อเรามองถึงออบเจกต์ตัวหนึ่งๆ จะเห็นว่ามันจะต้องมีข้อมูลที่ให้กำหนดลักษณะเฉพาะของออบเจกต์ ซึ่งจะถูกเรียกว่า พร็อพเพอร์ตี้ (Property)

เมื่อมีพร็อพเพอร์ตี้แล้วสิ่งหนึ่งที่มีมักจะขาดไม่ได้ในออบเจกต์ก็คือ ความสามารถในการทำงานของออบเจกต์ ซึ่งจะถูกเรียกว่า เมธอด (Method)

เมื่อผู้เขียน โปรแกรมสร้างออบเจกต์หนึ่งๆ ขึ้นมาจากคลาส ในภาษาของการเขียนโปรแกรมจะกล่าวได้ว่าเป็น “ออบเจกต์ ก็คือ อินสแตนซ์ (Instance) ของคลาส”

2.6.1.9 การเข้าถึงข้อมูลที่เก็บภายในคลาส

ในคลาสจะประกอบด้วยข้อมูลต่างๆนั่นคือ พร็อพเพอร์ตี้ และเมธอด ซึ่งศัพท์ของนักเขียนโปรแกรม จะเรียกทั้งสองอย่างรวมๆว่า เมมเบอร์ (Member) ของคลาส

จุดคืออย่างหนึ่งของคลาสคือ การที่ไม่อนุญาตให้เข้าถึงข้อมูลภายในของคลาสได้โดยตรง จะต้องกระทำผ่านเมมเบอร์ของคลาสซึ่งก็สอดคล้องกับชีวิต ที่ผู้เขียนโปรแกรมใช้งานรถยนต์ได้จากสิ่งที่เตรียมไว้ให้ สำหรับการเข้าถึงข้อมูลภายในของคลาส Delphi นั้นแบ่งได้เป็น 3 ระดับได้แก่

- Private เป็นระดับการเข้าถึงข้อมูลที่เข้าถึงได้เฉพาะข้อมูลที่อยู่ภายในตัวคลาส หรือ อินสแตนซ์ของคลาสนั้นๆ ไม่อนุญาตให้ผู้ใช้โปรแกรม หรือคลาสอื่นใดเข้าถึงได้
- Public เป็นระดับการเข้าถึงข้อมูลที่เข้าถึงได้จากผู้ใช้โปรแกรมคลาสทุกคน หรือคลาสอื่นๆ โดยไม่จำกัด
- Protected เป็นระดับการเข้าถึงข้อมูลเฉพาะคลาสที่ถูกสืบทอดมาจากคลาสนี้เท่านั้น

2.6.1.10 Constructor และ Destructor

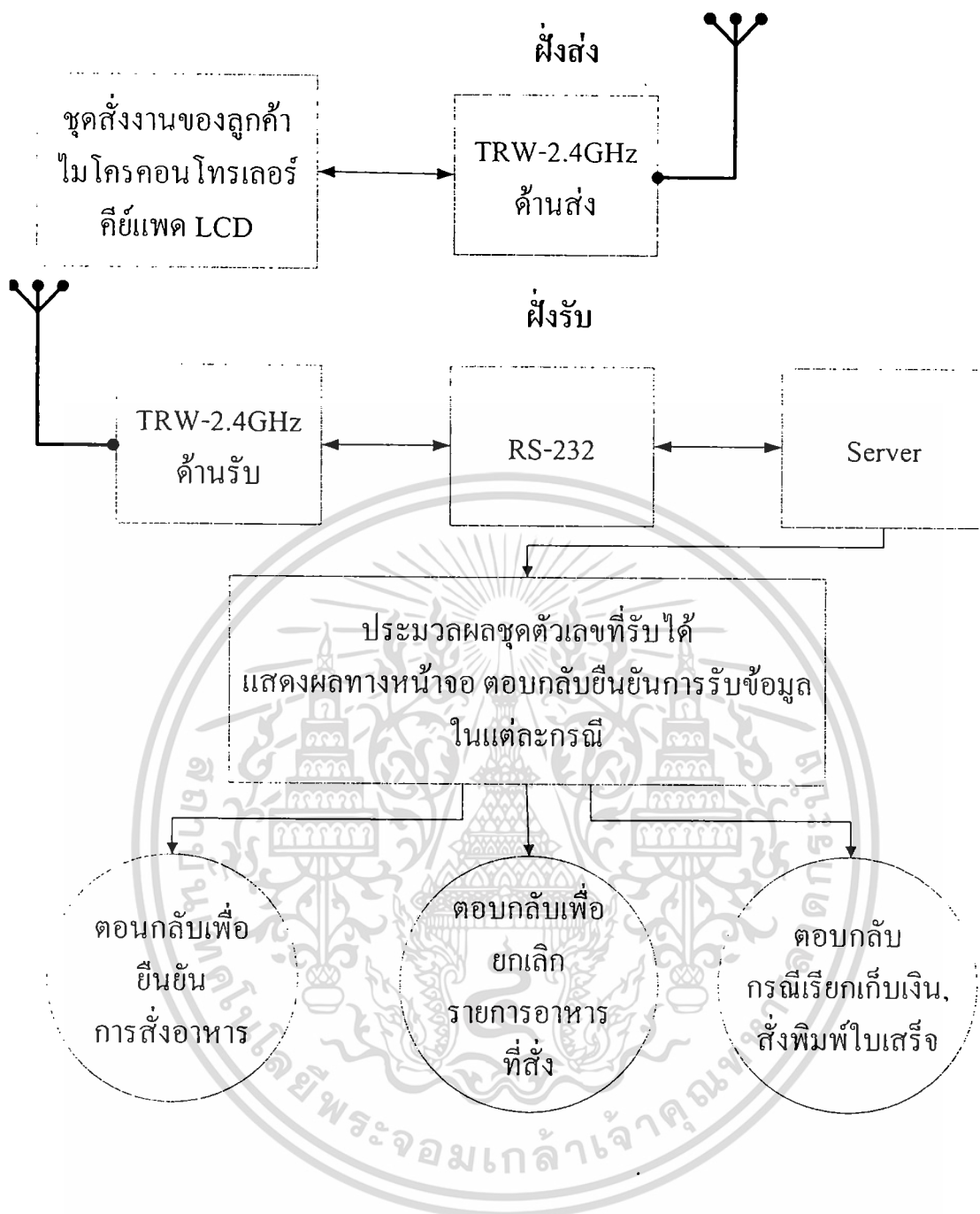
นอกเหนือจากเมธอดที่ได้กำหนดให้คลาสแล้ว ในการเขียนโปรแกรมแบบ OOP นั้นยังต้องมีเมธอดที่จำเป็นอยู่อีก 2 ตัว ซึ่งจะต้องทำหน้าที่ทุกครั้งเมื่อออบเจกต์นั้นเริ่มสร้างขึ้นมา หรือกำลังจะเลิกใช้งานไป

Constructor เป็นเมธอดที่ทำงานตอนที่ผู้เขียน โปรแกรมเริ่มสร้างออบเจกต์ขึ้นมาจาก คลาส โดยจะทำหน้าที่กำหนดค่าเริ่มต้นที่จำเป็นให้กับพรีอเพอร์ตีต่างๆ จับจองทรัพยากรของ ระบบที่จำเป็นต่อการทำงานของออบเจกต์

ส่วน Destructor เป็นเมธอดที่ทำงานในลักษณะตรงกันข้ามคือ ทำหน้าที่ ตอนที่ ออบเจกต์นั้นเลิกใช้งาน โดยจะคืนทรัพยากรให้แก่ระบบปฏิบัติการ



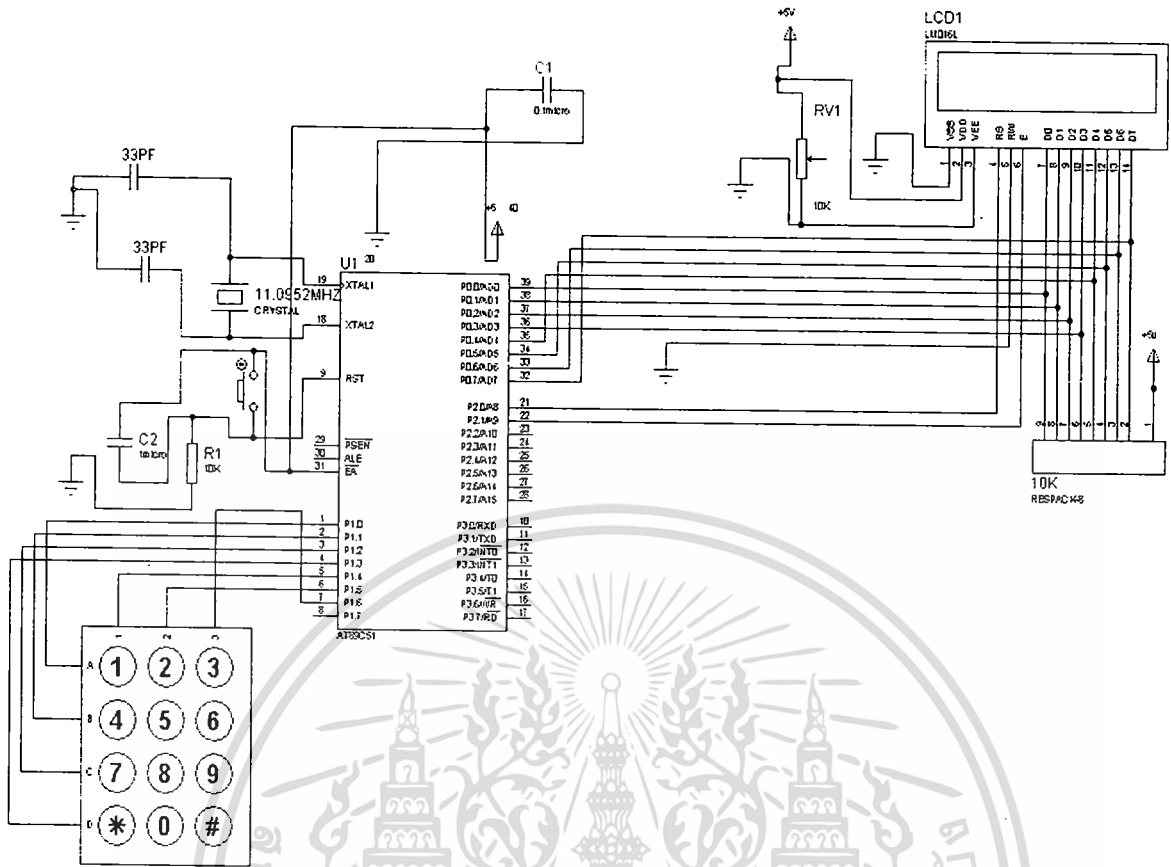
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงบล็อกไดอะแกรมของเครื่องสั่งอาหารแบบไร้สาย

3.1 ชุดสั่งงานของลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



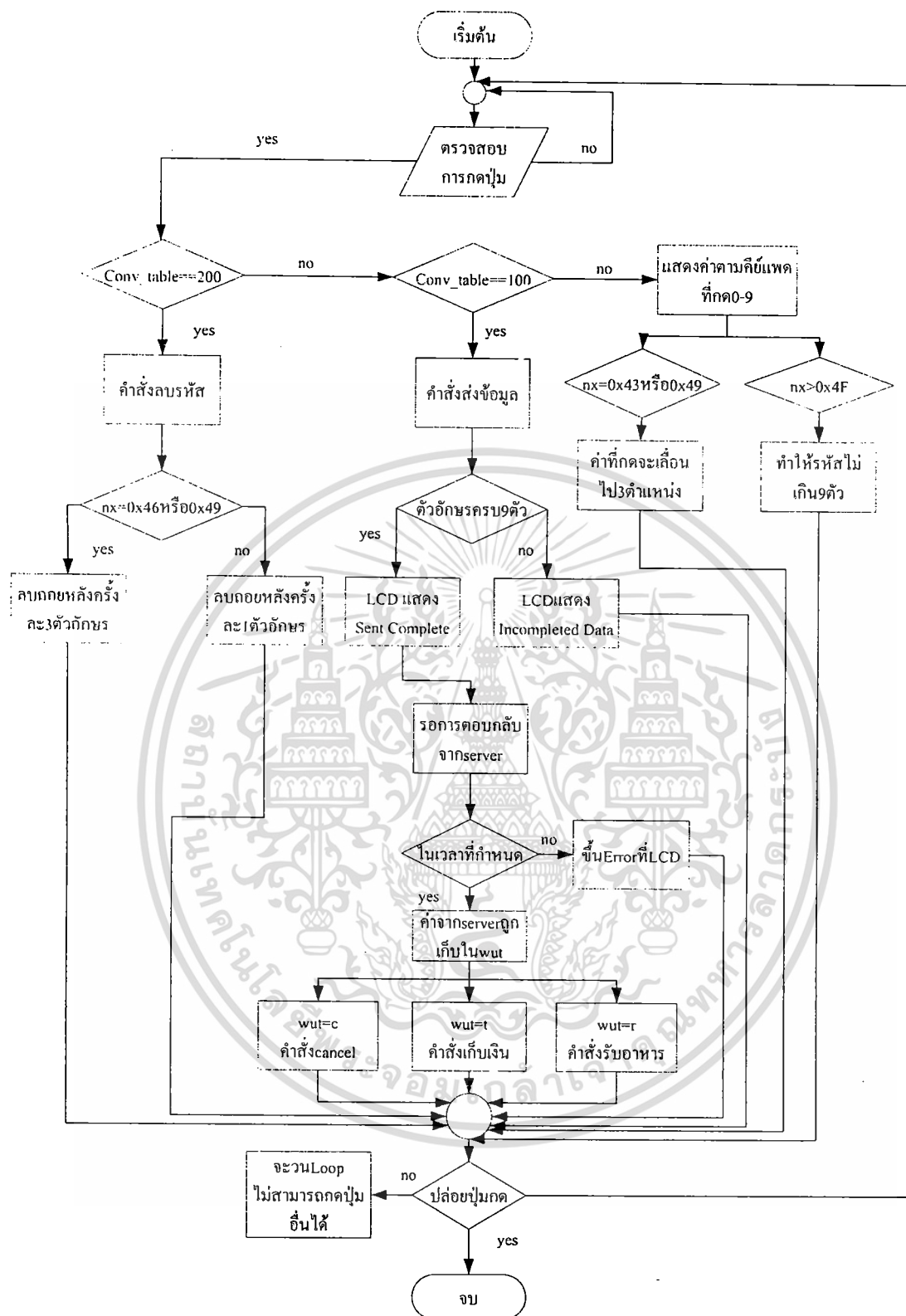
รูปที่ 3.2 แสดงวงจรชุดสั่งงานของลูกค้า

ประกอบไปด้วยจอ LCD คีย์แพดและ ไมโครคอนโทรเลอร์ เมื่อลูกค้ามาใช้บริการก็จะทำการ กดรหัสโดยดูจากเมนูอาหาร แล้วทำการกดรหัสผ่านคีย์แพด ซึ่งตัวคีย์แพดจะทำการตรวจสอบที่แถวและหลักของคีย์แพดที่เชื่อมกับไมโครคอนโทรเลอร์ เพื่อนำค่าหมายเลขรหัสที่กดนั้นไปแสดงผลที่หน้าจอ LCD และเก็บค่าไว้ที่หน่วยความจำ เพื่อทำการส่งต่อไปยัง วงจรTRW-2.4GHz

ในหน้าปัดของคีย์แพดนั้นนอกจากจะมีปุ่มกดหมายเลข 0-9 แล้วยังมีปุ่ม * ที่ทำหน้าที่ในการลบที่ละตัวอักษร และปุ่ม # สำหรับใช้ส่งข้อมูล ส่วนในหน้าจอ LCD การแสดงผลมีรหัส 9 ตัว ซึ่งมี3ช่วงๆละ3ตัว โดยที่รหัส 3 ตัวแรกจะเป็นรหัสของอาหาร อีก 3 ตัวต่อมาจะเป็นรหัสของจำนวนอาหารที่ลูกค้าสั่ง ส่วนรหัส 3 ตัวสุดท้ายจะเป็นหมายเลขโต๊ะ ซึ่งการทำงานทั้งหมดนั้น จะถูกควบคุมโดยไมโครคอนโทรเลอร์

ในการที่จะส่งข้อมูลรหัสอาหารออกไปนั้นจะต้องทำการกดปุ่ม# แล้วข้อมูลจะถูกส่งออกจากขา Tx ไปยังขาRx ของวงTRW-2.4GHz นอกจากนี้วงจรชุดสั่งงานของลูกค้ายังสามารถแสดงผลการตอบกลับที่มาจาก Server แล้วแสดงผลที่จอ LCD ได้ด้วยโดยข้อมูลจะเข้ามาทางขา Rx ของไมโครคอนโทรเลอร์ในชุดสั่งงานของลูกค้า

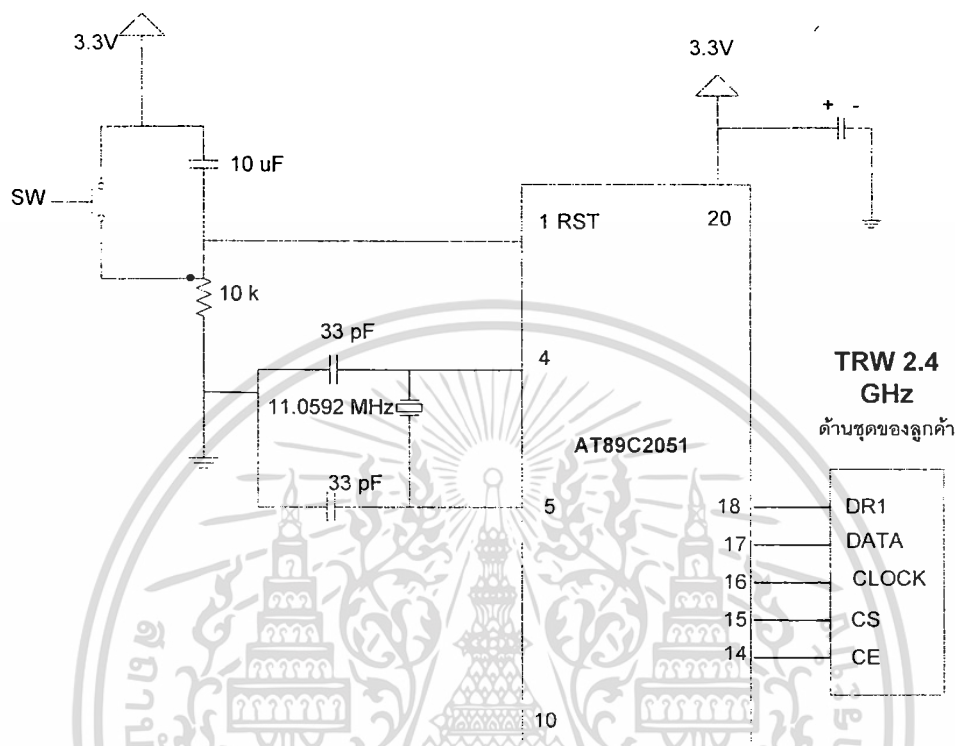
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงแผนผังงาน(Flow Chart) การทำงานในไมโครคอนโทรเลอร์ของชุดตั้งงานของลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วงจร TRW-2.4GHz ฟังก์ชัน

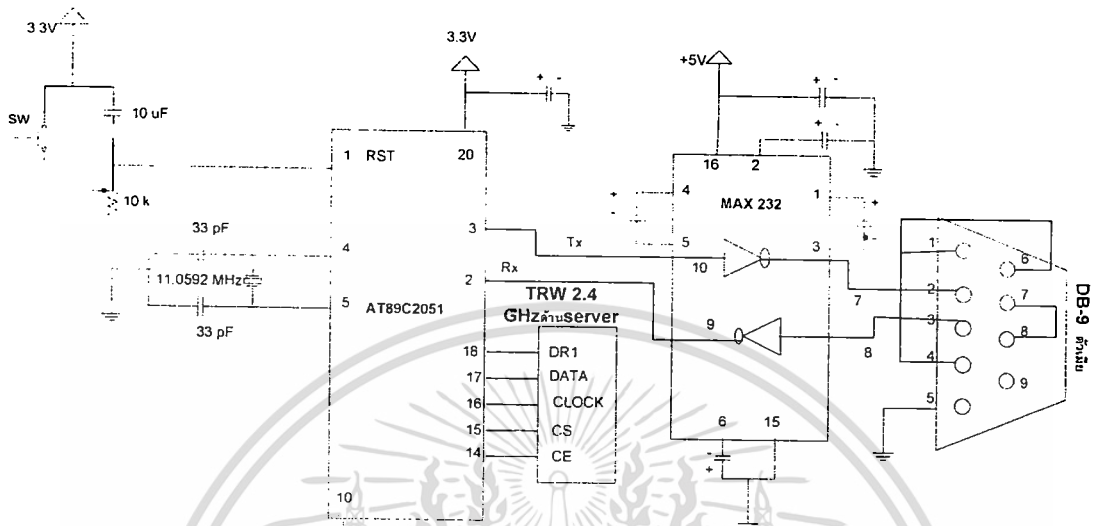


รูปที่ 3.4 แสดงวงจร TRW-2.4GHz ด้านส่ง

ในวงจรนี้จะใช้ไมโครคอนโทรลเลอร์ชนิด 20 ขา และ TRW-2.4GHz เนื่องจาก TRW-2.4GHz จะใช้ไฟได้ไม่เกิน 3.3 โวลต์ ดังนั้นจึงต้องมี regulator แปลงไฟให้ไม่เกิน 3.3 โวลต์ ถ้าไม่เป็นเช่นนั้น TRW-2.4GHz อาจเสียหายได้

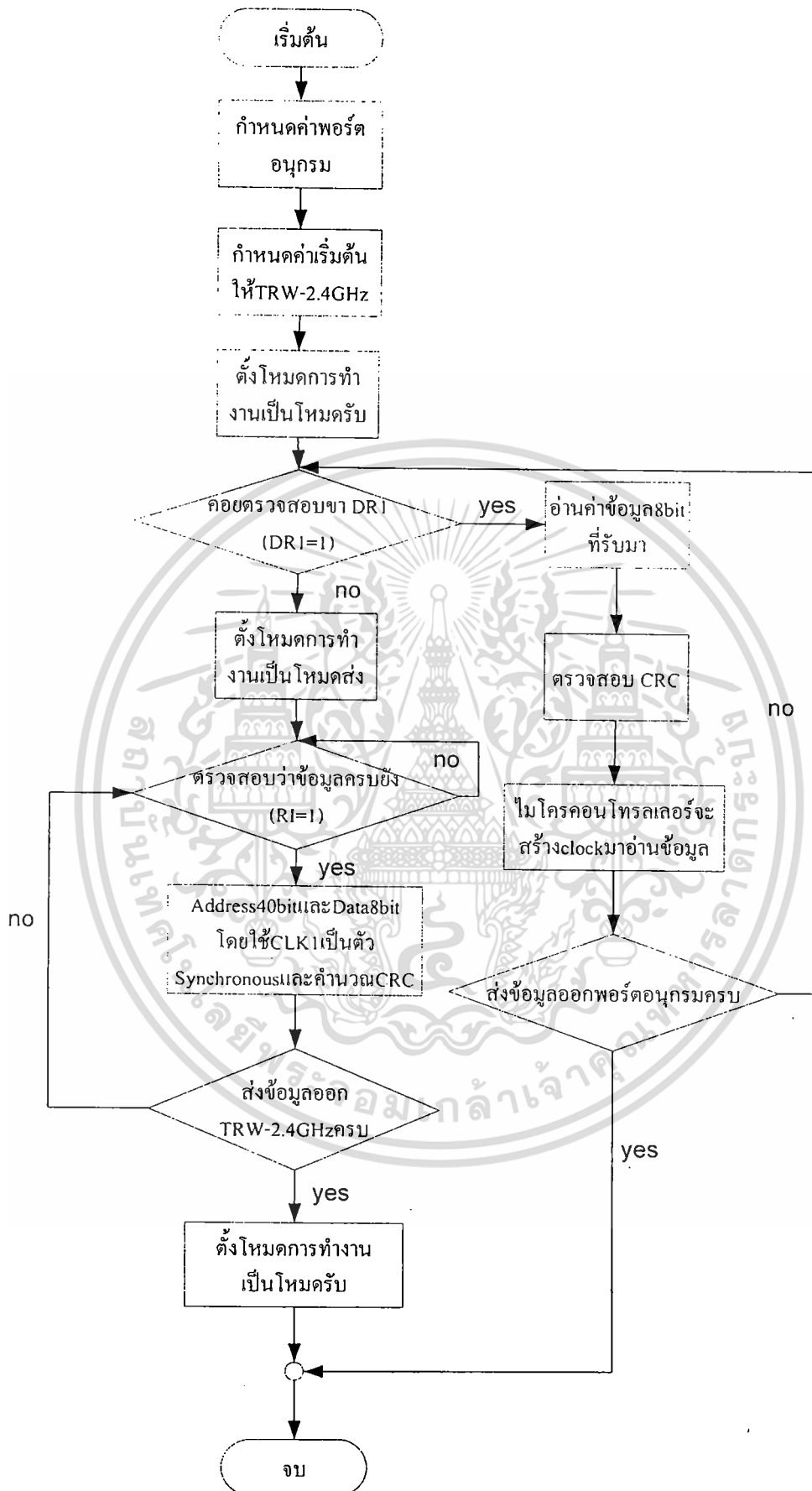
ในการทำงานนั้นไมโครคอนโทรลเลอร์จะรับข้อมูลที่ส่งจากชุดส่งงานของลูกค้า เข้าทางขา Rx แล้วทำการส่งต่อไปแบบไร้สาย ไปยัง TRW-2.4GHz ด้านรับต่อไป จากนั้นจะคอยการตอบกลับจากฝั่ง Server แล้วส่งไปยังวงจรชุดส่งงานของลูกค้าเพื่อแสดงผล

3.3 วงจร TRW-2.4GHz ฟังเครื่องServer



รูปที่ 3.5 แสดงวงจร TRW-2.4GHz ด้านรับ

ในส่วนวงจรนี้จะทำการรับข้อมูลที่ส่งมาจาก TRW-2.4GHz ด้านส่ง ซึ่งจะทำการส่งออกด้วยขา Tx ของไมโครคอนโทรลเลอร์ไปยังวงจรส่งผ่านพอร์ตอนุกรม(Max-232) ซึ่งไอซีMax-232 ทำหน้าที่แปลงระดับสัญญาณไฟฟ้า แล้วส่งข้อมูลผ่านDB-9 เข้าสู่คอมพิวเตอร์เพื่อทำการตรวจสอบรหัสที่ถูกต้องตามต่อไป จากนั้นเมื่อ Server ทำการประมวลผลแล้วจะส่งข้อมูลที่แสดงการยืนยันสิ่งที่ถูกต้องการกลับมาแล้วส่งต่อไปยังฝั่งลูกค้าต่อไป

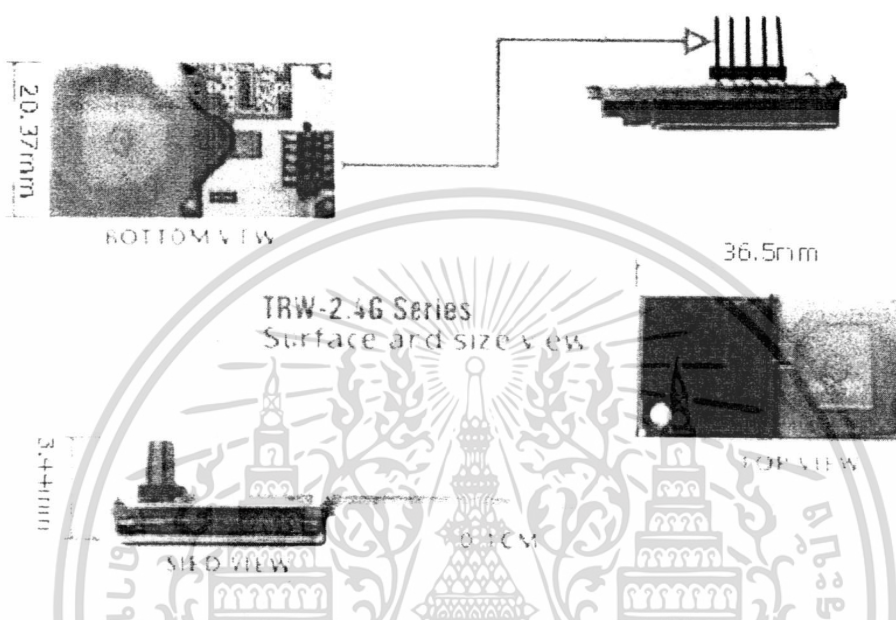


รูปที่ 3.6 แสดงแผนผังงาน(Flow chart) การรับ-ส่งของTRW-2.4GHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การใช้งาน RF Module (TRW-2.4GHz)

TRW-2.4GHz เป็น RF Module ที่ใช้ความถี่ 2.4-2.524 GHz สื่อสาร RF ด้วยเทคนิค GFSK มีความเร็วในการส่งข้อมูล 250 kbps ในช่วง 280 m แต่มีความเร็วสูงขึ้นเป็น 1 Mbps ในระยะทำงาน 150 m ใช้ไฟเลี้ยง Module $\approx 3V$ สามารถตั้งค่าให้เป็น transmitter หรือ Receiver ก็ได้และมี mode การทำงานแบบ ShockBurst แบบ Duo Ceiver มี Antenna ภายใน Module



รูปที่ 3.7 แสดง RF Module (TRW-2.4GHz)

3.4.1 การทำงานของ Mode ShockBurst

เมื่อต้องการส่งข้อมูล (ใช้ Pin CE, CLK1, DATA)

- ตั้ง Pin CE = 1 (เปิด ShockBurst Mode)
- ส่งข้อมูล (ADDRESS และ PAYLOAD) ทาง pin DATA โดยใช้ CLK1 เป็นตัว Synchronous Data ที่ส่งให้ TRW-2.4G
- TRW-2.4GHz จะคำนวณ CRC ให้ (ขนาดตามที่เรา config ไว้)
- ตั้ง pin CE = 0 (ปิด ShockBurst Mode) TRW-2.4GHz ใส่ Preamble และเริ่มส่ง RF จนกว่าจะเสร็จแล้วกลับมา Standby

เมื่อข้อมูลเข้ามาที่ตัวรับ (ใช้ Pin CE, DR1, CLK1, DATA)

- เมื่อมีข้อมูลเข้ามาที่ตัวรับ TRW-2.4GHz จะทำการตรวจสอบ ADDR ที่ส่งมาค่าตรงกับของตัวมันที่เรา Config ไว้หรือไม่ หากตรงจะทำงานต่อ หากไม่ตรงก็จะเพิกเฉยรอ package ใหม่
- ทำการ Check CRC ที่ส่งมาว่าถูกต้องหรือไม่ ถ้าถูกทำงานต่อหากผิดจะไม่มีการทำงานตอบกลับว่า Error ใดๆ ทั้งสิ้น
- TRW-2.4GHz จะ set ขา DR1 เป็น 1 ดังนั้น microcontroller จึงคอยวน check ขา DR1 ว่าเป็น 1 เมื่อใดแสดงว่า Dataพร้อมแล้ว
- Microcontroller ต้องสร้าง clock มาที่ CLK1 เพื่ออ่านข้อมูลจากทาง DATA ออกไป ถ้าครบแล้ว DR1 จะกลายเป็น 0 อีกครั้ง

ถ้าหากใช้ Duo Ceiver (คือ สามารถกำหนดให้ TRW-2.4GHz ตัวนี้มีช่องรับ 2 ทาง) การแบ่งการทำงานออกเป็น 2 channel แต่ละ channel มี ADDR ของมันเอง โดยที่ใช้ขาสัญญาณไม่ตรงกัน

Rx1 ใช้ pin (DR1, CLK1, DATA)

Rx2 ใช้ pin (DR2, CLK2, DOUT2)

3.4.2 การ Config TRW-2.4GHz

- ตั้งให้ขา CS = 1 (เริ่มการ config) แต่ขา CE ต้องเป็น 0 ด้วยเพราะจะไม่อนุญาตให้ config และรับส่งข้อมูลพร้อมกันได้
- ตั้งขา DATA ที่ใช้ config 144bit เข้าไปทาง DATA pin เริ่มจาก MSB และใช้ CLK1 ในการ Synchronous
- เมื่อเสร็จสิ้นให้ set CS = 0 เหมือนเดิม

ตารางที่ 3.1 วิธีการ Config TRW-2.4GHz

ตำแหน่ง bit	จำนวน bit	ชื่อ	คำอธิบาย
143:120	24	Test	สงวนใช้Test
119:112	8	DATA2_1	ความยาวของ payload จำนวนจาก Payload = 256-ADDR_W-CRC
111:104	8	DATA1_W	
103:64	40	ADDR2	ค่าของADDR ของตัวรับ ตั้งค่าได้ ทั้ง channel 1 และ 2 มีขนาดสูงสุดได้ 5 byte
63:24	40	ADDR1	
23:18	6	ADDR_W	กำหนดความยาวของ ADDR ได้
17	1	CRC_L	ความยาวของ CRC '0' = 8 bit '1' = 16 bit
16	1	CRC_EN	กำหนดใช้หรือไม่ใช้ CRC
15	1	RX2_EN	กำหนดใช้กี่ channel(รับ)
14	1	CM	เปิดปิด Shockburst Mode
13	1	RFDR_SB	กำหนด Data Rate '0' = 250 kbps '1' = 1 Mbps
12:10	3	XO_F	กำหนดเกี่ยวกับ CRYSTAL แต่ กำหนดตายตัวเป็น 011
9:8	2	RF_PWR	กำหนด Power ในการส่ง '00' (-20dBm) '01' (-1dBm) '10' (-5dBm) '11' (0dBm)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7:1	7	RF_CH	กำหนดความถี่ที่ใช้ในการส่งคำนวณ ดังนี้ $RF_{ส่ง} = 2400 + RF_CH \text{ MHz}$
0	1	RE_EN	กำหนดให้ TRW เป็น Tx-Rx '0'(Tx) '1'(RX)

ดังนั้นในวงจร TRW-2.4GHz ที่ใช้นั้นจึงได้กำหนดค่าต่างๆไว้ดังนี้

1. ความยาวข้อมูลใน channel 1 และ channel 2 คือ 8 bit
2. Address Channel 2 ทั้ง 5 bytes คือ (0xD1),(0xAA),(0x55),(0xAA),(0x55)
Address Channel 1 ทั้ง 5 bytes คือ (0xB5),(0x55),(0xAA),(0x55),(0xAA)
3. กำหนดให้ใช้ CRC ขนาด 16 bit
4. ใช้ 1 channel ในการรับ
5. เปิด ShockBurst mode
6. Data Rate คือ 250 kbps
7. RF Power คือ 0 dbm
8. ความถี่ในการส่งคือ 2,410 MHz

3.5 ส่วนของเครื่อง Server

หน้าที่ของเครื่อง Server

1. รับข้อมูลจากลูกค้าแต่ละโต๊ะ โดยรับค่าผ่านทาง Serial Port
2. แสดงรายละเอียดของข้อมูลที่ลูกค้าแต่ละโต๊ะสั่ง โดยแสดงผ่านหน้าจอแสดงผลที่สร้างขึ้นจากโปรแกรมเคลไฟล์
3. ส่งค่ากลับเพื่อยืนยันการส่งข้อมูลของลูกค้าแต่ละโต๊ะ ว่าข้อมูลที่ลูกค้าส่งมานั้นเป็นข้อมูลแบบใด
4. สามารถพิมพ์ใบเสร็จเพื่อใช้สำหรับการเรียกเก็บเงินจากลูกค้าได้

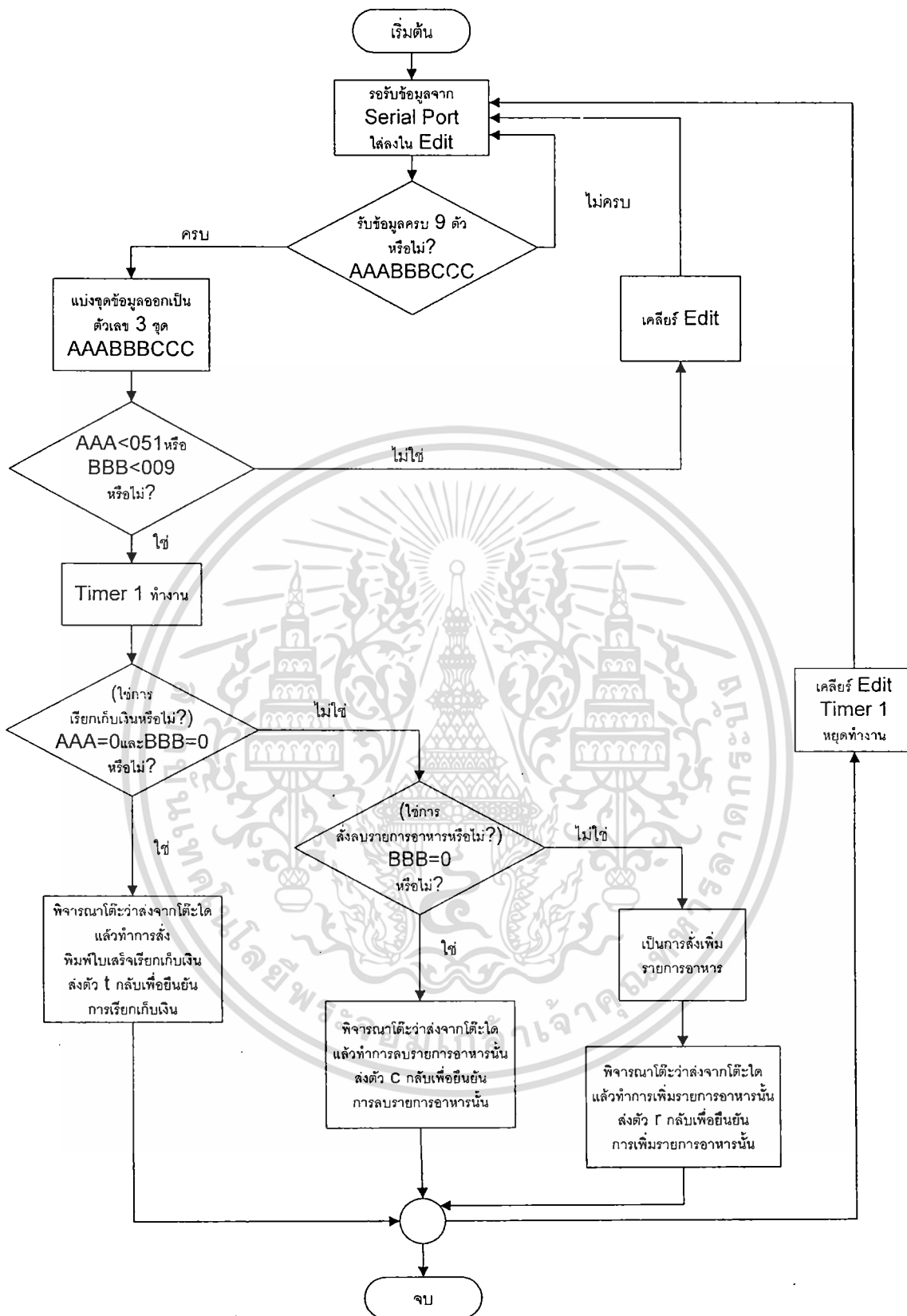
โปรแกรมของเครื่อง Server มีลักษณะการทำงาน ดังนี้

เมื่อโปรแกรมของเครื่อง Server อยู่ในสถานะพร้อมทำงาน โปรแกรมจะทำการรอรับข้อมูลที่ถูกส่งเข้ามาทาง Serial Port โดยลักษณะของข้อมูลที่ต้องการของ โปรแกรมจะเป็นชุดตัวเลข 9 ตัวเท่านั้น โปรแกรมจะรอค่าจนกว่ารับค่าตัวเลขครบ 9 ตัว แล้วจึงนำชุดตัวเลข 9 ตัวนั้น ไปเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมวลผล แล้วแสดงผลของข้อมูลที่ได้รับมาได้ออกทางหน้าจอแสดงผลที่สร้างขึ้นจากโปรแกรม Delphi และมีการตอบกลับเพื่อใช้สำหรับการยืนยันว่าเครื่อง Server นั้น ได้รับข้อมูลจากลูกค้าแล้ว โดยการส่งการยืนยันไปยัง โตะของลูกค้าที่ส่งข้อมูลมา การทำงานของโปรแกรมเครื่อง Server นั้น จะเป็นการทำงานแบบอัตโนมัติ โดยใช้ Timer ควบคุมในการรับค่าและประมวลผลทั้งหมด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



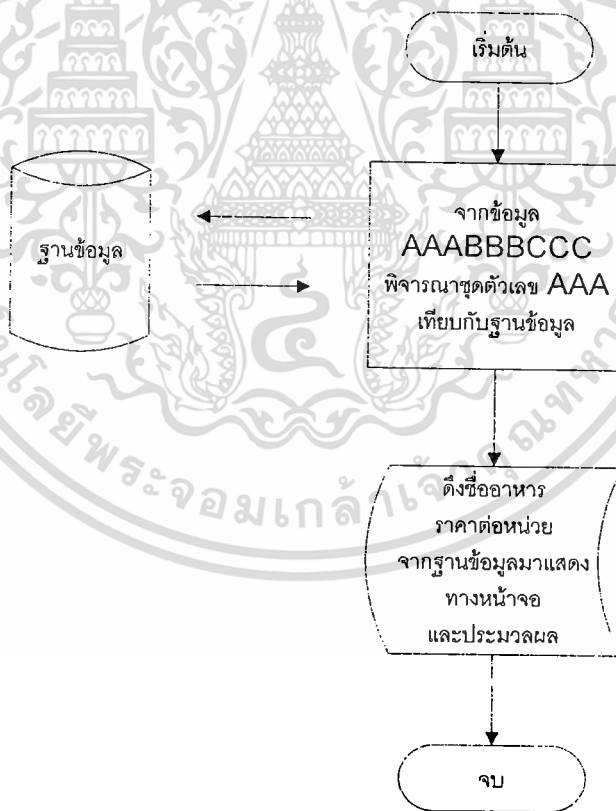
รูปที่ 3.8 แสดงแผนผังงาน(Flow Chart) การทำงานในส่วน of เครื่อง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายได้ดังนี้

1. เมื่อโปรแกรมอยู่ในสถานะพร้อมทำงาน โปรแกรมจะทำการรอรับข้อมูลจาก Serial Port โดยรับข้อมูลมาใส่ในช่อง Edit ข้อมูลที่เข้ามานั้น จะต้องเป็นข้อมูลตัวเลข 9 ตัว โดยโปรแกรมจะต้องรอรับข้อมูลให้ครบ 9 ตัวก่อน โปรแกรมถึงจะดำเนินการทำในขั้นตอนต่อไป เพื่อกันไม่ให้เกิดการผิดพลาดในการนำข้อมูลไปประมวลผลโดยที่ข้อมูลยังไม่ครบ 9 ตัว
2. เมื่อข้อมูลตัวเลขใน Edit ครบ 9 ตัวแล้ว โปรแกรมจะทำการแบ่ง ชุดตัวเลข 9 ตัวที่รับมานั้น ออกเป็นชุดตัวเลข 3 ชุด ชุดละ 3 ตัวเลข โดยหลักการแบ่งชุดตัวเลขมีดังนี้

- ชุดที่หนึ่ง ใช้สำหรับแสดงรหัสอาหารที่ถูกคำสั่ง ชุดตัวเลขนี้จะถูกนำไปเปรียบเทียบกับฐานข้อมูล แล้วดึงค่า ชื่อรายการอาหาร และ ราคาต่อหน่วย จากฐานข้อมูลมาแสดงทางจอ และนำไปใช้สำหรับการคิดราคาอาหารของโต๊ะนั้น ตัวอย่างเช่น รหัสอาหาร 002 เมื่อเทียบกับฐานข้อมูลแล้ว จะได้ชื่ออาหารคือ ข้าวผัดรวมมิตร ราคาต่อหน่วยเท่ากับ 30 บาท



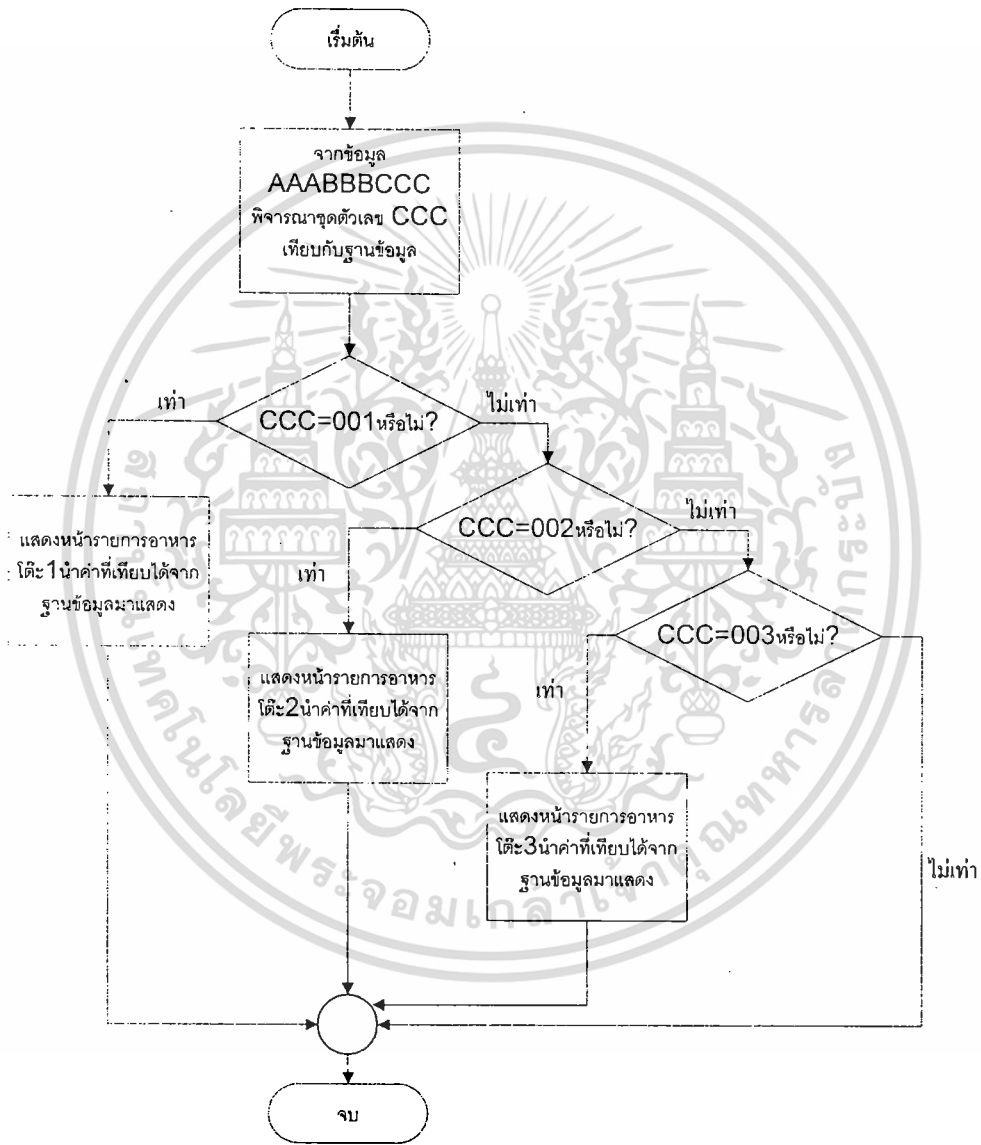
รูปที่ 3.9 แสดงแผนผังงาน(Flow Chart) การทำงานในส่วนของเครื่อง Server ในการพิจารณาตัวเลขชุดที่หนึ่ง(รหัสอาหาร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชุดที่สอง ใช้สำหรับแสดงจำนวนอาหารต่อหนึ่งรายการอาหารที่ถูกคำสั่ง โดยชุดตัวเลขที่สองนี้ จะถูกนำไปคูณกับราคาต่อหน่วยของรายการอาหารที่ถูกคำสั่งมา เพื่อนำไปใช้สำหรับการคิดราคาอาหารของโต๊ะนั้น

- ชุดที่สาม ใช้สำหรับแสดงหมายเลขของโต๊ะที่ถูกคำสั่งส่งข้อมูลตัวเลขเข้ามา เพื่อนำไปพิจารณาว่า ข้อมูลถูกส่งมาจากลูกค้าโต๊ะใด เช่น 002001003 นั่นคือ ข้อมูลถูกส่งมาจากลูกค้าโต๊ะที่

3



รูปที่ 3.10 แสดงแผนผังงาน (Flow Chart) การทำงานในส่วนเครื่อง Server ในการพิจารณาตัวเลขชุดที่สาม(โต๊ะที่)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น สมมติให้ข้อมูลที่ได้รับได้มีค่าเท่ากับ 002001003 (AAA=002, BBB=001, CCC=003)

ชุดที่หนึ่ง	ชุดที่สอง	ชุดที่สาม
002	001	003
รหัสอาหาร	จำนวน	โต๊ะที่

รูปที่ 3.11 แสดงข้อมูลที่ได้รับได้มีค่าเท่ากับ 002001003

นั่นคือ เป็นข้อมูลที่ลูกค้าจาก โต๊ะที่ 3 ส่งมาเพื่อ สั่งอาหารที่มีรหัสเท่ากับ 002 = ข้าวผัดรวมมิตร ราคาต่อหน่วย 30 บาท สั่งเป็นจำนวน 1 หน่วย(จาน)

3. โปรแกรมจะทำการพิจารณาว่า รหัสอาหารนั้นเกินในฐานข้อมูลหรือไม่ และหมายเลขโต๊ะมีอยู่ใน โปรแกรมหรือไม่ ถ้ารหัสอาหารเกิน หรือหมายเลขโต๊ะไม่มีอยู่ใน โปรแกรมก็จะไม่มีการประมวลผลในขั้นตอนนี้ต่อไป

4. เมื่อพิจารณาแล้วว่าชุดตัวเลขแต่ละชุดเข้ากับ โปรแกรม Timer 1 จะเริ่มทำงาน โปรแกรมจะทำการพิจารณาชุดตัวเลขทีละส่วน ว่าลูกค้าส่งข้อมูลตัวเลขมาเพื่อจุดประสงค์ใด

การพิจารณาจุดประสงค์ของชุดตัวเลขแบ่งออกได้เป็น 3 กรณี ดังนี้
(สมมติให้ รหัสอาหาร=002 ,จำนวน=001 และ โต๊ะที่=003)

กรณีที่ 1 เป็นการส่งข้อมูลจากลูกค้า เพื่อที่จะต้องการเรียกเก็บเงิน โดยชุดตัวเลขจะมีเลขชุดที่หนึ่ง และชุด ที่สอง มีค่าเป็น 000

ชุดที่หนึ่ง	ชุดที่สอง	ชุดที่สาม
000	000	003
รหัสอาหาร	จำนวน	โต๊ะที่

รูปที่ 3.12 แสดงข้อมูลจากลูกค้าเพื่อเรียกเก็บเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่าง หลังจากทีโปรแกรมพิจารณาแล้วว่า เป็นข้อมูลตัวเลขที่ส่งมาเพื่อที่ต้องการเรียกเก็บเงิน โปรแกรมจะทำการส่งพิมพ์ใบเสร็จของโต๊ะ 3 พร้อมทั้งส่งตัว e ออกไปเพื่อยืนยันการเรียกเก็บเงินของโต๊ะ 3

กรณีที่ 2 เป็นการส่งข้อมูลจากลูกค้า เพื่อที่จะต้องการลบบรายการอาหาร ที่ได้ทำการส่งมาแล้ว โดยชุด

ตัวเลขจะมีเลขชุดที่สองมีค่าเป็น 000

ชุดที่หนึ่ง	ชุดที่สอง	ชุดที่สาม
002	000	003
รหัสอาหาร	จำนวน	โต๊ะที่

รูปที่ 3.13 แสดงข้อมูลจากลูกค้าเพื่อลบบรายการอาหาร

จากตัวอย่าง หลังจากทีโปรแกรมพิจารณาแล้วว่า เป็นข้อมูลตัวเลขที่ส่งมาเพื่อที่ต้องการลบบรายการอาหาร โปรแกรมจะทำการลบบรายการอาหารที่มีรหัสเท่ากับ 002 ของโต๊ะ 3 และทำการคิดราคารวมของค่าอาหารของโต๊ะนั้นใหม่ แล้วทำการส่งตัว c ออกไป เพื่อยืนยันการลบบรายการอาหารของโต๊ะ 3

กรณีที่ 3 เป็นการส่งข้อมูลจากลูกค้า เพื่อที่จะต้องการสั่งเพิ่มรายการอาหารตามรายละเอียดของชุดตัวเลขที่ส่งมา

ชุดที่หนึ่ง	ชุดที่สอง	ชุดที่สาม
002	001	003
รหัสอาหาร	จำนวน	โต๊ะที่

รูปที่ 3.14 แสดงข้อมูลจากลูกค้าเพื่อสั่งเพิ่มรายการอาหาร

จากตัวอย่าง หลังจากทีโปรแกรมพิจารณาแล้วว่า เป็นข้อมูลตัวเลขที่ต้องการสั่งเพิ่มรายการอาหาร โปรแกรมจะทำการเพิ่มรายการอาหารที่มีรหัสเท่ากับ 002 ของโต๊ะ 3 และทำการคิดเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ราคารวมของค่าอาหารของโต๊ะนั้นใหม่ แล้วทำการส่งตัว r ออกไป เพื่อยืนยันการเพิ่มรายการอาหารของโต๊ะ 3

4. เมื่อทำการประมวลผลและแสดงผลทางหน้าจอแสดงผลในแต่ละกรณีเรียบร้อยแล้ว โปรแกรมจะทำการเคลียร์ค่าข้อมูลตัวเลข 9 ตัว ออกจาก Edit และ Timer 1 หยุดทำงาน โปรแกรมจะทำการรอรับค่าที่จะเข้ามาใหม่อีกครั้ง

บทที่ 4

สรุปและข้อเสนอแนะ

การทดลอง

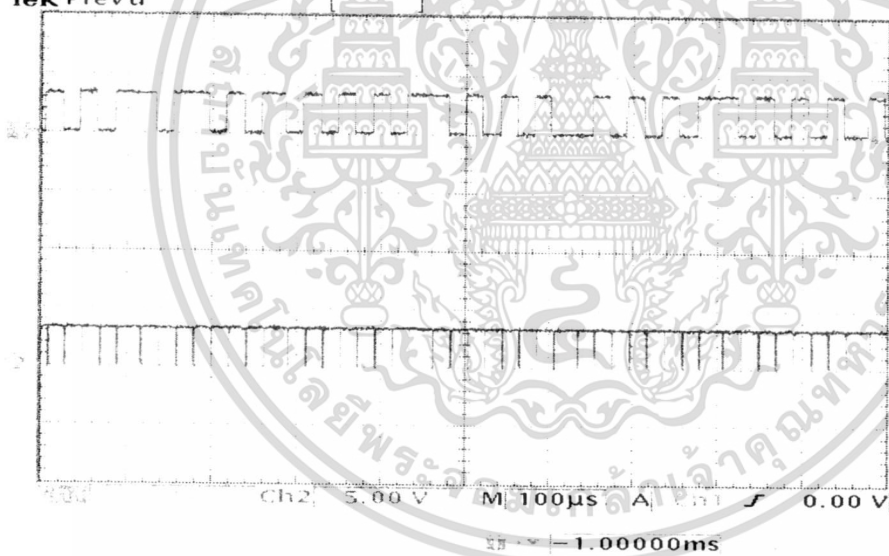
4.1 การทดลองเรื่อง การวัดสัญญาณการรับ – ส่งระหว่าง TRW-2.4GHz ด้านส่งและด้านรับ ขั้นตอนการทดลอง

1. ทำการวัดสัญญาณโดยออสซิลอสโคป ที่ขา Data เทียบกับ Clock ของ TRW-2.4GHz ฟังชุดส่งงานของลูกค้า(ด้านส่ง)
2. ทำการวัดสัญญาณโดยออสซิลอสโคป ที่ขา Data เทียบกับ CS ของ TRW-2.4GHz ฟังชุดส่งงานของลูกค้า(ด้านส่ง)
3. ทำการวัดสัญญาณโดยออสซิลอสโคป ที่ขา Data เทียบกับ CE ของ TRW-2.4GHz ฟังชุดส่งงานของลูกค้า(ด้านส่ง)
4. ทำการวัดสัญญาณโดยออสซิลอสโคป ที่ขา Data เทียบกับ Clock ของ TRW-2.4GHz ฟัง Server (ด้านรับ)
5. ทำการวัดสัญญาณโดยออสซิลอสโคป ที่ขา DR1 เทียบกับ Data ของ TRW-2.4GHz ฟัง Server (ด้านรับ)
6. ทำการวัดสัญญาณโดยออสซิลอสโคป ที่ขา Data เทียบกับ CS ของ TRW-2.4GHz ฟัง Server (ด้านรับ)
7. ทำการวัดสัญญาณโดยออสซิลอสโคป ที่ขา Data เทียบกับ CE ของ TRW-2.4GHz ฟัง Server (ด้านรับ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

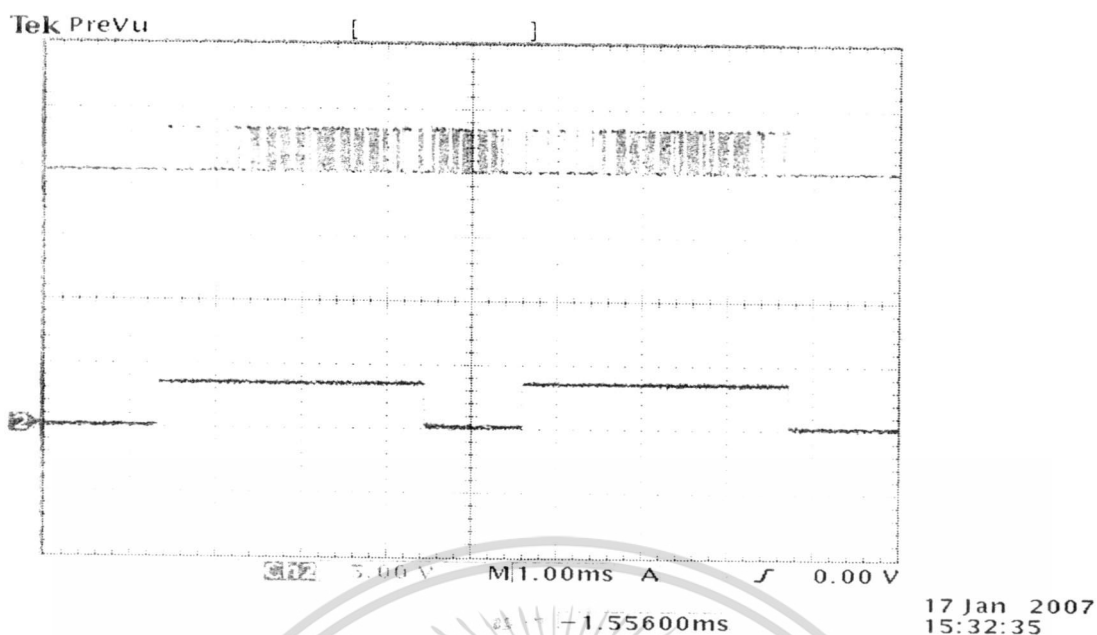
ผลการทดลอง

Tek PreVu

17 Jan 2007
16:09:42

รูปที่ 4.1 การวัดสัญญาณที่ขาData (บน) เทียบกับสัญญาณนาฬิกาที่ขาCLK1 (ล่าง) ฟังก์ชันของลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



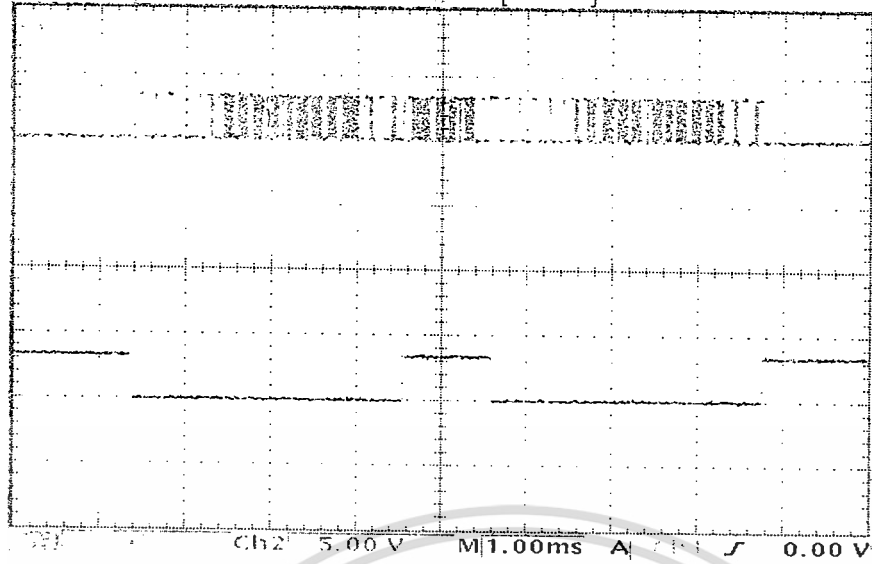
รูปที่ 4.2 การวัดสัญญาณที่ขาData (บน) กับสัญญาณที่ขาCS (ล่าง) ฟังชุดของลูกค้า



รูปที่ 4.3 ภาพขยายการวัดสัญญาณที่ขาData (บน) กับสัญญาณที่ขาCS (ล่าง) ฟังชุดของลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

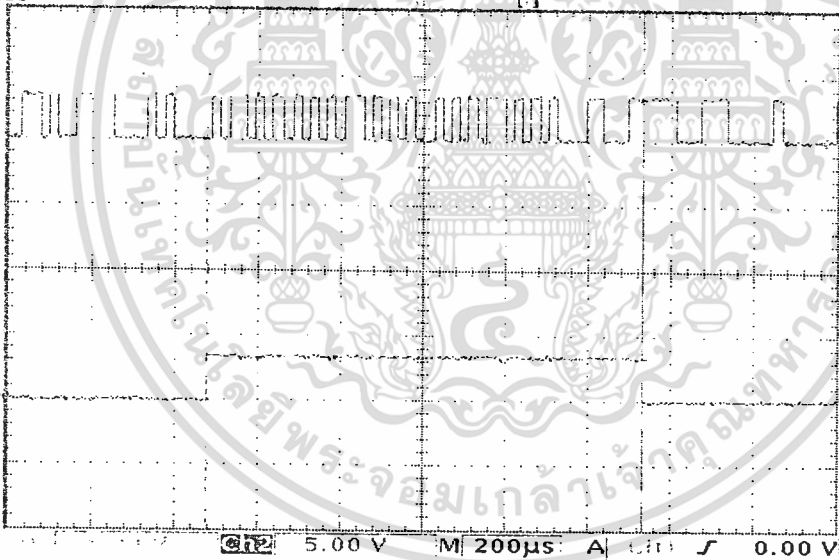
Tek PreVu



17 Jan 2007
15:47:20

รูปที่ 4.4 การวัดสัญญาณที่ขาData (บน) กับสัญญาณที่ขาCE (ล่าง) ฟังชุดของลูกค้ำ

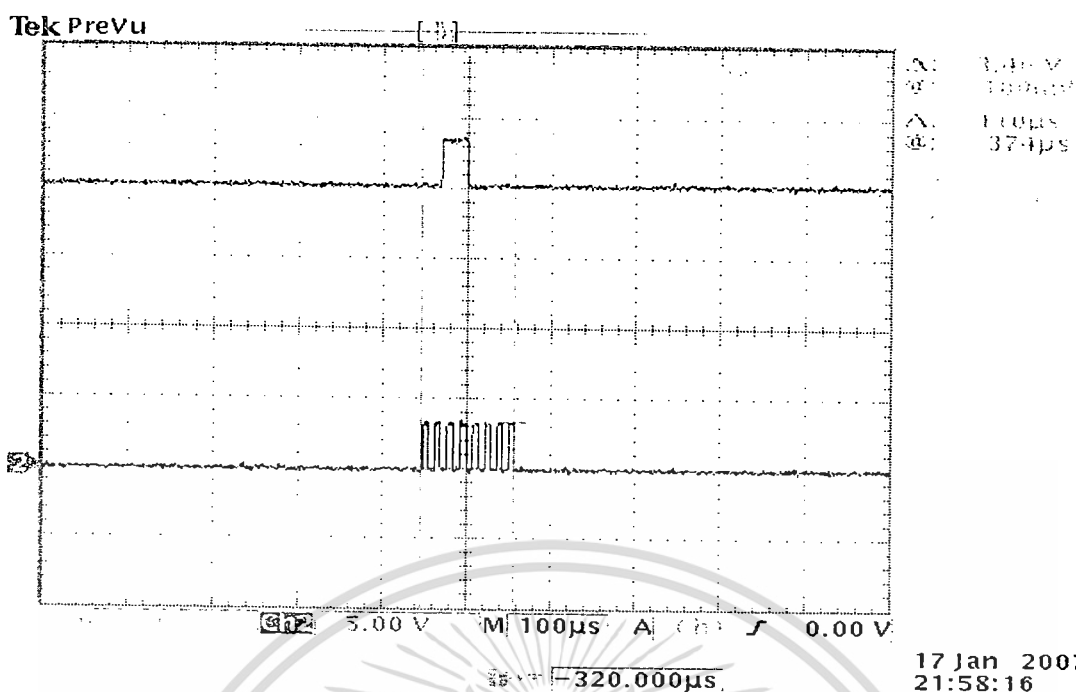
Tek PreVu



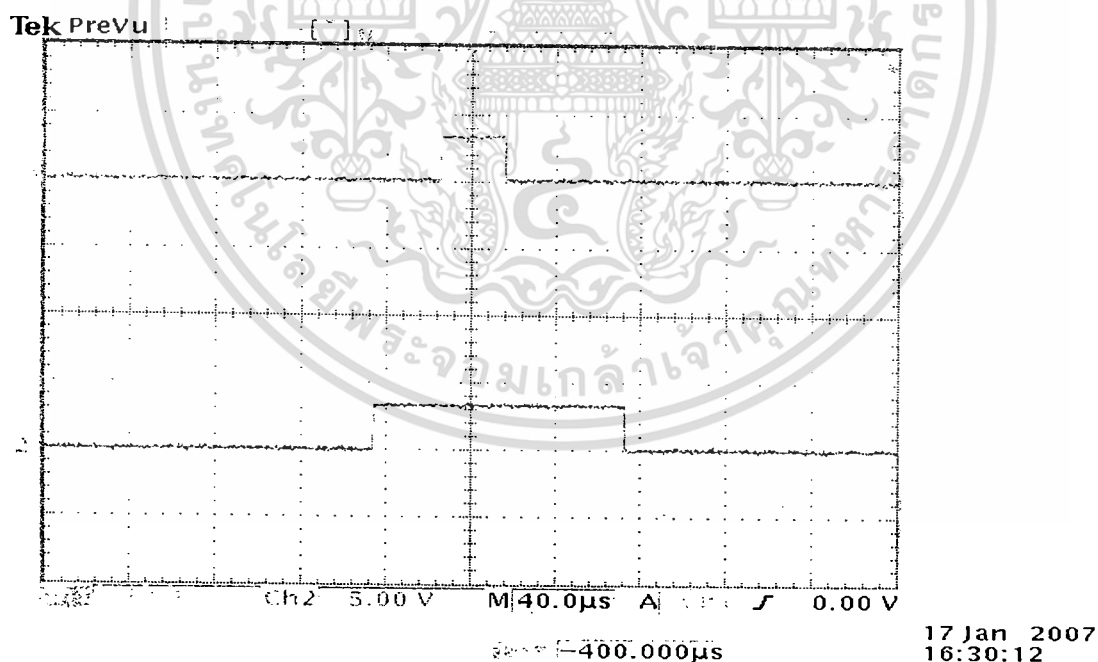
17 Jan 2007
15:52:01

รูปที่ 4.5 ภาพขยายการวัดสัญญาณที่ขาData (บน) กับสัญญาณที่ขาCE (ล่าง) ฟังชุดของลูกค้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

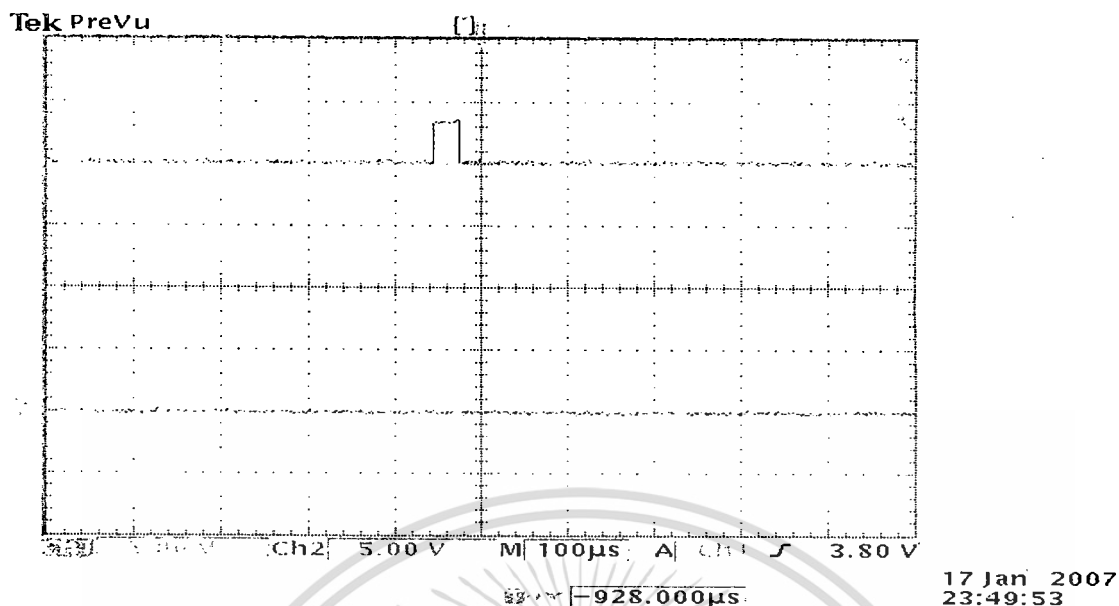


รูปที่ 4.6 การวัดสัญญาณที่ขา Data (บน) กับสัญญาณนาฬิกาที่ขา CLK1 (ล่าง) ฟังเครื่อง Server

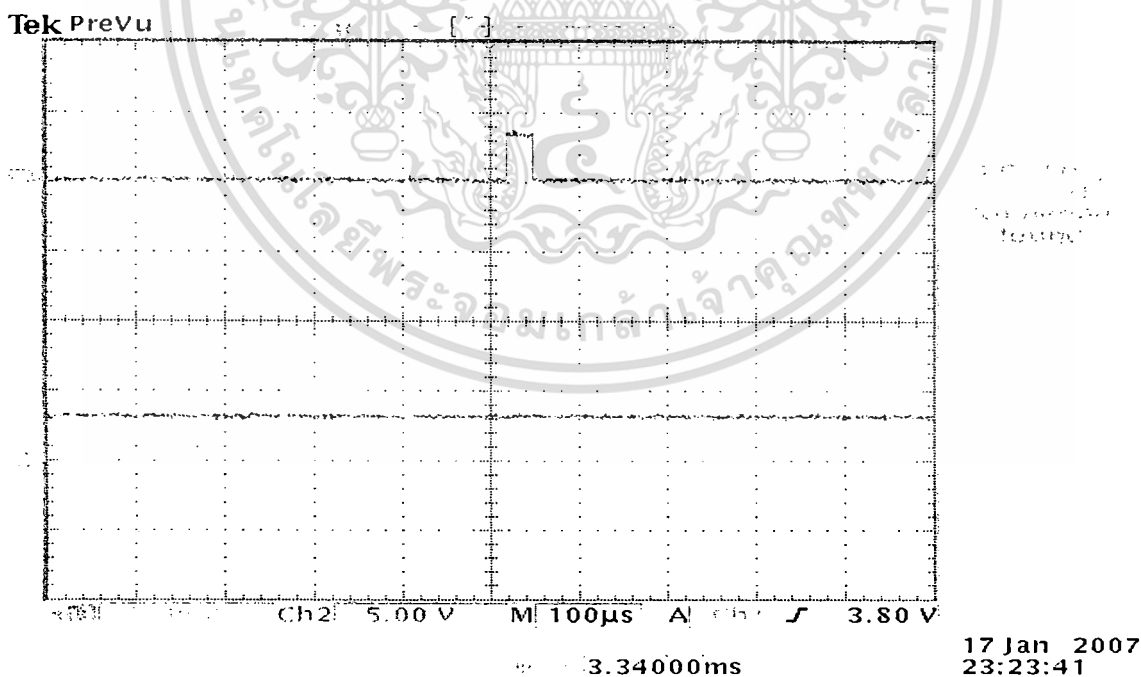


รูปที่ 4.7 การวัดสัญญาณที่ขา Data (บน) กับสัญญาณที่ขา DR1 (ล่าง) ฟังเครื่อง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 การวัดสัญญาณที่ขาData (บน) กับสัญญาณที่ขาCS (ล่าง) ฝั่งเครื่อง Server



รูปที่ 4.9 การวัดสัญญาณที่ขาData (บน) กับสัญญาณที่ขาCE (ล่าง) ฝั่งเครื่อง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

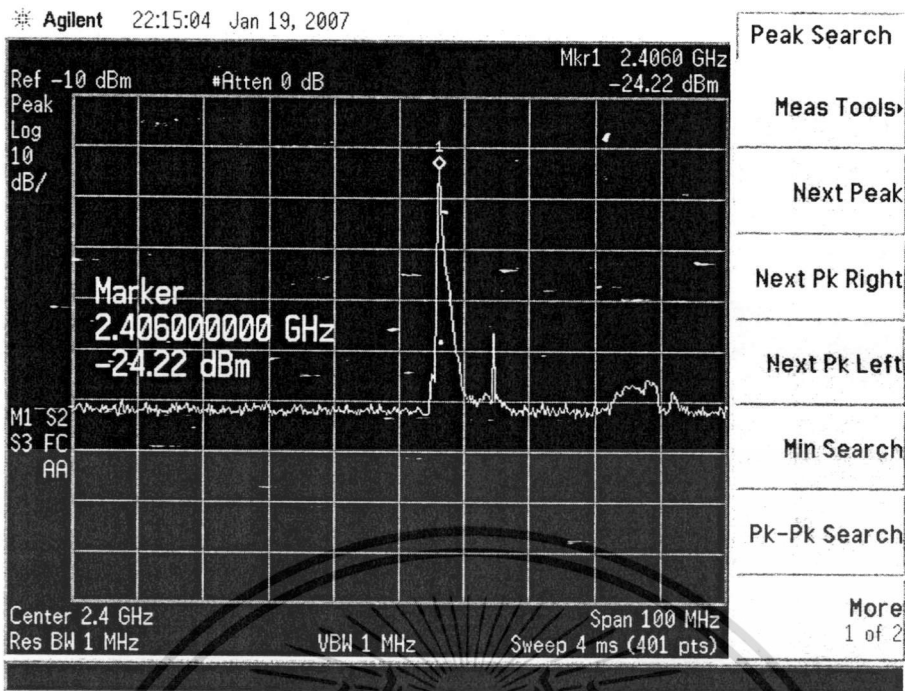


4.2 การทดลองเรื่อง การวัดความถี่ที่ใช้ส่งและกำลังของสัญญาณของTRW-2.4GHz ขั้นตอนการทดลอง

1. ทำการวัดความถี่และกำลังส่งของสัญญาณ โดยใช้เครื่อง Spectrum Analyzer
2. ทำการสังเกตและบันทึกผลการทดลอง

ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดงผลการวัดความถี่และกำลังของสัญญาณที่ใช้ส่งของ TRW-2.4GHz

จากรูปที่ 4.10 จะเห็นได้ว่าความถี่ที่วัดได้คือ 2.4060 GHz หรือ 2.406 MHz และวัดค่ากำลังสัญญาณได้ -24.22 dBm

4.3 การทดลองเรื่อง การแสดงผลของเครื่อง Server และการแสดงผลตอบรับที่จอ LCD กรณีที่ ข้อมูลที่ส่งมาจากชุดส่งงานของลูกค้าถูกต้อง (รหัสอาหาร < 050 , หมายเลขโต๊ะ < 009)
ขั้นตอนการทดลอง

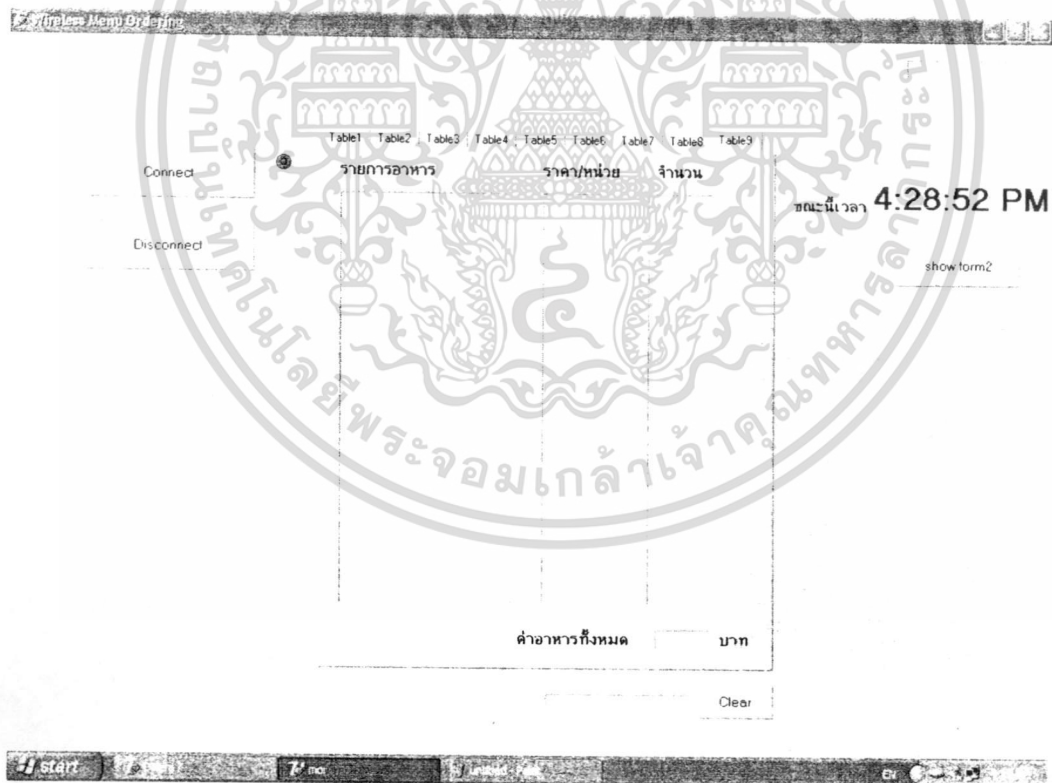
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เปิดเครื่องทางด้านชุดสั่งงานของลูกค้าและเครื่องทางด้าน Server ให้อยู่ในสถานะพร้อมทำงาน และเชื่อมต่อเครื่องทางด้าน Server เข้ากับเครื่อง Server โดยเชื่อมต่อผ่านทาง Serial Port



รูปที่ 4.11 แสดงหน้าจอ LCD เมื่อชุดสั่งงานของลูกค้าอยู่ในสถานะพร้อมทำงาน

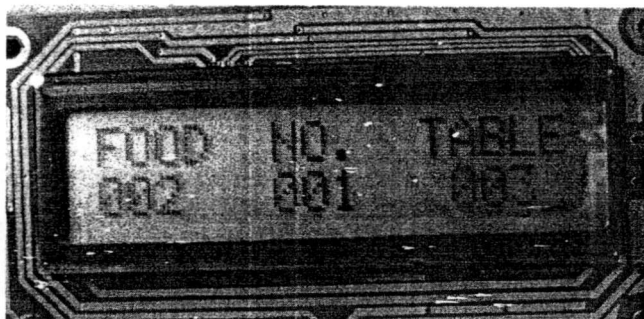
2. Run โปรแกรมเดสทอป หน้าฟอร์ม Wireless Food Ordering จะปรากฏขึ้น ทำการคลิกปุ่ม Connect เพื่อทำการเชื่อมต่อโปรแกรมเข้ากับ Serial Port เครื่อง Server จะอยู่ในสถานะพร้อมรอรับข้อมูล



รูปที่ 4.12 แสดงหน้าจอแสดงผลของเครื่อง Server ในสถานะพร้อมทำงาน

3. ทำการกดคีย์แพดทางด้านชุดสั่งงานของลูกค้า ในการทดลองที่ 4.3 ให้ส่งข้อมูลที่มีค่าเท่ากับ 002001003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



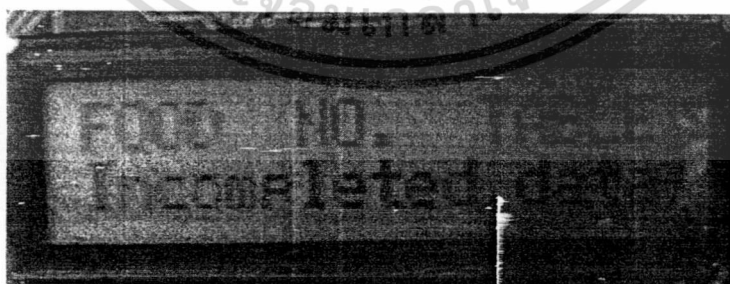
รูปที่ 4.13 แสดงหน้าจอ LCD พร้อมทั้งจะส่งข้อมูลที่มีค่าเท่ากับ 002001003

กดปุ่ม # เพื่อทำการส่งข้อมูล หน้าจอ LCD จะปรากฏคำว่า Sent complete ดังรูปที่ 4.14



รูปที่ 4.14 แสดงหน้าจอ LCD หลังจากการกดปุ่ม #

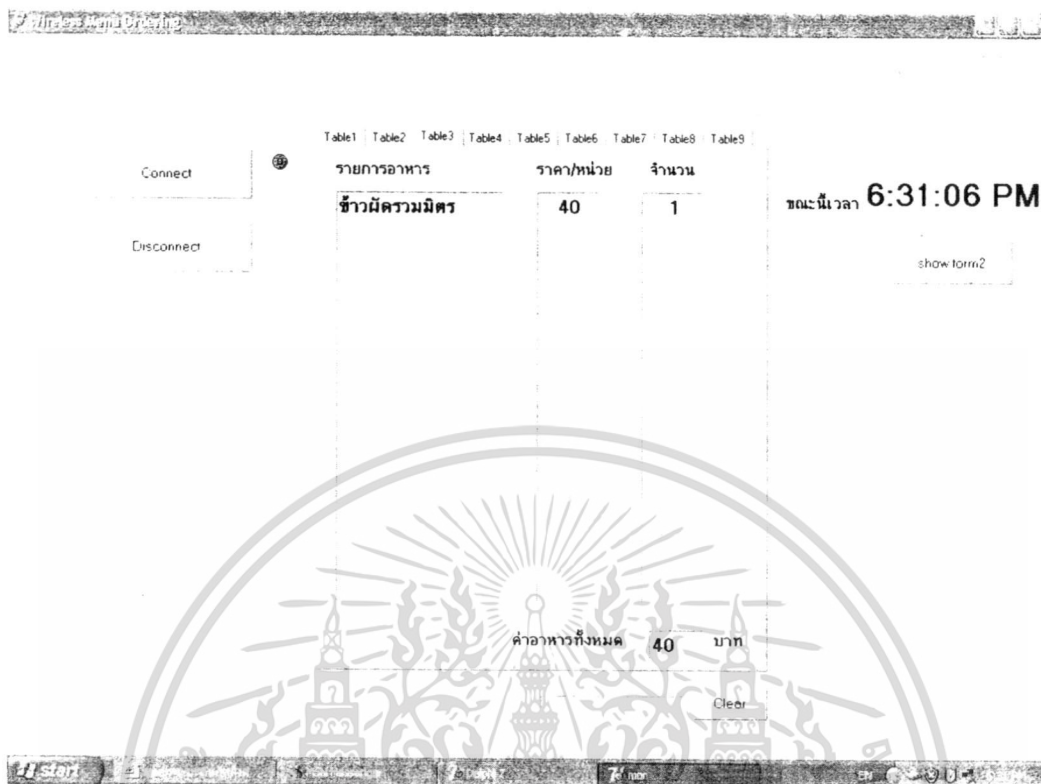
ถ้ากดตัวเลขไม่ครบ 9 หลัก แล้วกดปุ่ม # หน้าจอ LCD จะปรากฏคำว่า Incompleted data ดังรูปที่ 4.15



รูปที่ 4.15 แสดงหน้าจอ LCD เมื่อมีกดตัวเลขไม่ครบ 9 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง



รูปที่ 4.16 แสดงหน้าจอแสดงผลของเครื่อง Server หลังจากรับข้อมูลที่มีค่าเท่ากับ 002001003

หน้าจอแสดงผลของเครื่อง Server จะแสดงหน้ารายการอาหารของโต๊ะที่ 3 และทำการเพิ่มรายการอาหารที่มีรหัสเท่ากับ 002 = ข้าวผัดรวมมิตร และราคาต่อหน่วยเท่ากับ 30 บาท และตั้งเป็นจำนวน 1 หน่วย ค่าอาหารทั้งหมด 30 บาท

ส่วนหน้าจอ LCD ของเครื่องชุดสั่งงานฝั่งลูกค้า จะปรากฏคำว่า Receive 002 เป็นการยืนยันว่า เครื่อง Server นั้น ได้รับข้อมูลการสั่งอาหารรหัส 002 เรียบร้อยแล้ว ดังรูปที่ 4.17

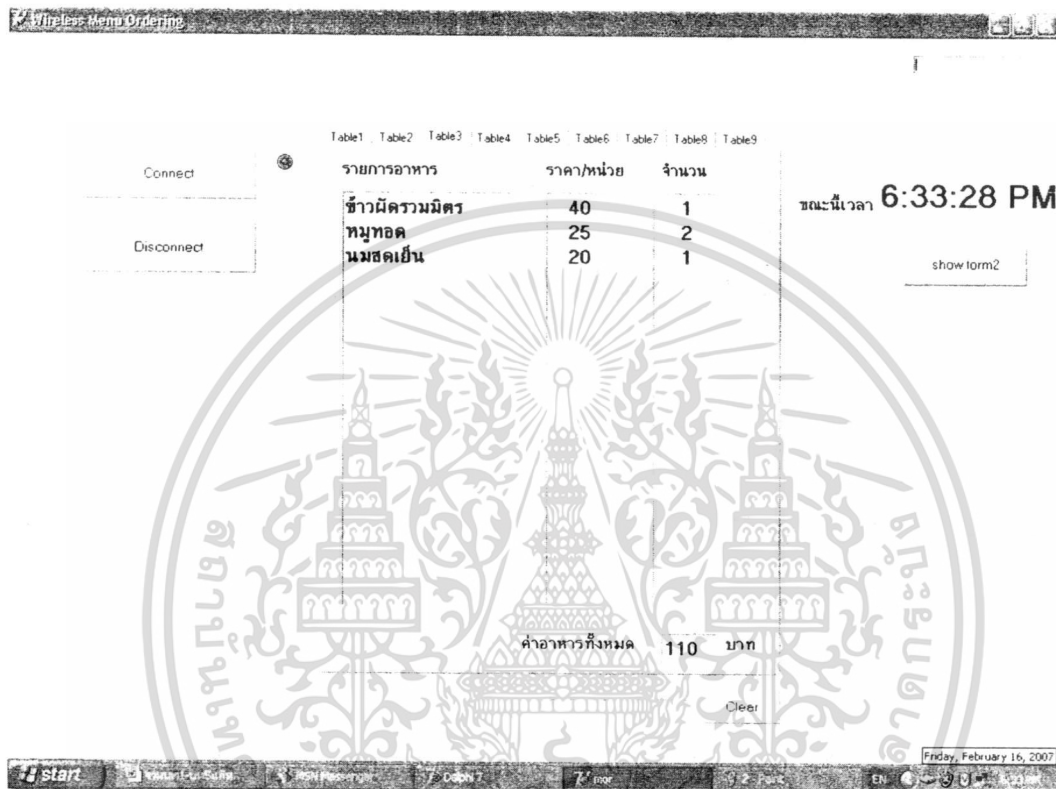


รูปที่ 4.17 แสดงหน้าจอ LCD เมื่อได้รับการยืนยันการสั่งเพิ่มรายการอาหารที่ 002 จากเครื่อง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการส่งข้อมูลจากชุดสั่งงานฝั่งลูกค้าอีก 2 ครั้ง โดยให้ข้อมูลมีค่าเท่ากับ 021002003 และ 045001003 ทำการส่งตามลำดับ

ผลการทดลอง



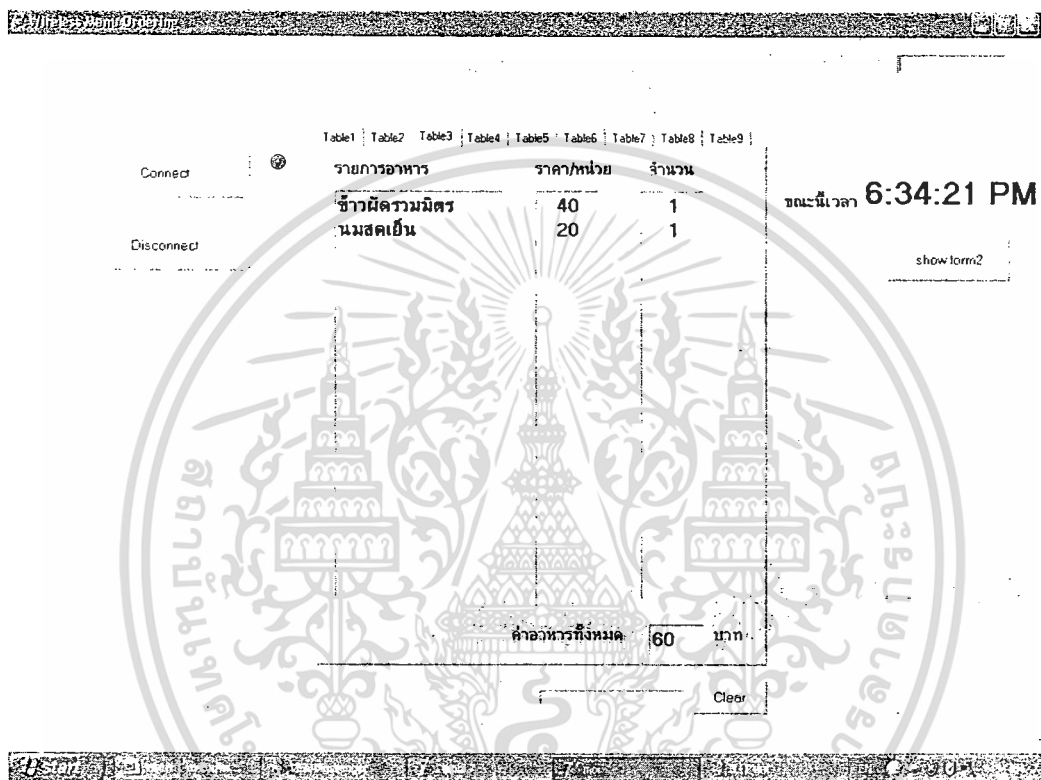
รูปที่ 4.18 แสดงหน้าจอแสดงผลของเครื่อง Server หลังจากรับข้อมูลที่มีค่าเท่ากับ 021002003 , 045001003 ตามลำดับ

จากรูปที่ 4.18 หน้ารายการอาหารของโต๊ะที่ 3 จะถูกเพิ่มรายการอาหารเข้าไปอีก 2 รายการตามลำดับที่ชุดสั่งงานฝั่งลูกค้าส่งมา พร้อมทั้งคิดราคาค่าอาหารรวมของโต๊ะที่ 3 ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ทำการส่งข้อมูลจากชุดสั่งงานของลูกค้ายีกครั้ง โดยให้ข้อมูลมีค่าเท่ากับ 021000003

ผลการทดลอง

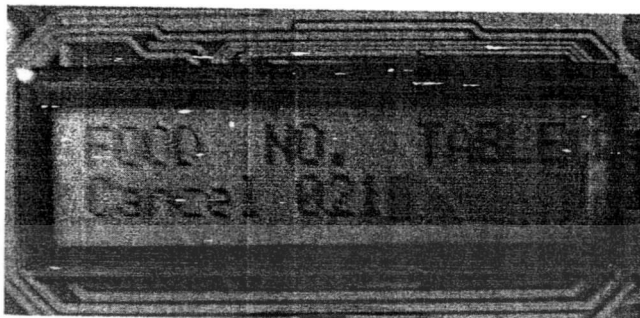


รูปที่ 4.19 แสดงหน้าจอแสดงผลของเครื่อง Server หลังจากรับข้อมูลที่มีค่าเท่ากับ 021000003

จากรูปที่ 4.19 หน้ารายการอาหารของโต๊ะที่ 3 รายการอาหารรหัส 021= หมูทอด จะถูกลบไป โปรแกรมจะทำการคิดราคาค่าอาหารรวมใหม่

ส่วนหน้าจอ LCD ของเครื่องชุดสั่งงานฝั่งลูกค้า จะปรากฏคำว่า Cancel 021 เป็นการยืนยันว่าเครื่อง Server นั้น ได้รับข้อมูลการลบอาหารรหัส 012 เรียบร้อยแล้ว ดังรูปที่ 4.20

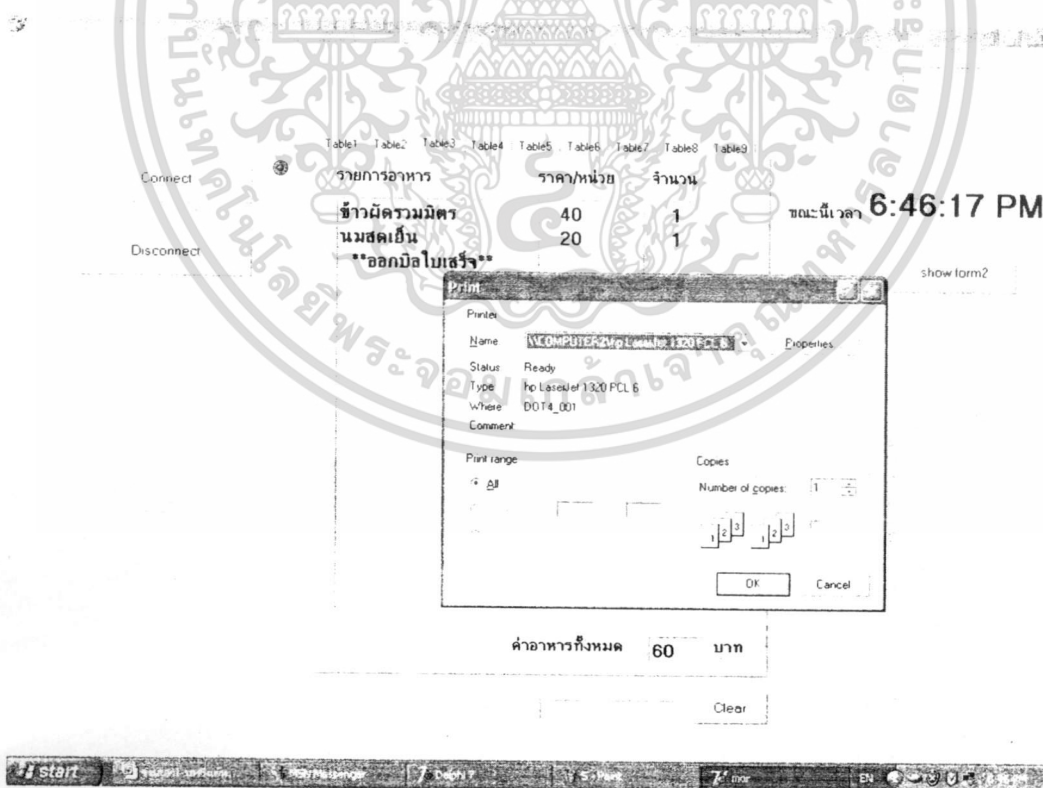
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 แสดงหน้าจอ LCD เมื่อได้รับการยืนยันการสั่งรายการอาหารที่ 021 จากเครื่อง Server

6. ทำการส่งข้อมูลจากชุดตั้งงานของลูกค้ายีกครั้งโดยให้ข้อมูลมีค่าเท่ากับ 000000003

ผลการทดลอง



รูปที่ 4.21 แสดงหน้าจอแสดงผลของเครื่อง Server หลังจากรับข้อมูลที่มีค่าเท่ากับ 000000003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.21 หน้ารายการอาหารของโต๊ะที่ 3 จะปรากฏ ***** ออกบิลใบเสร็จ ***** แล้วหน้าต่าง Print จะปรากฏขึ้น แล้วทำการคลิกปุ่ม OK เพื่อทำการพิมพ์ใบเสร็จ สำหรับใช้ในการคิดเงินกับลูกค้า

ใบเสร็จค่าอาหารโต๊ะ 3

ราคา/หน่วย จำนวนที่สั่ง รายการอาหาร

40 บาท 1 ข้าวผัดรวมมิตร

25 บาท 2 หมูทอด

20 บาท 1 นมสดเย็น

ยกเลิกรายการ หมูทอด

***** ค่าอาหารทั้งหมด 60 บาท *****

6:46:17 PM

รูปที่ 4.22 แสดงใบเสร็จที่ใช้สำหรับการเรียกเก็บเงินกับลูกค้าโต๊ะ 3

ส่วนหน้าจอ LCD ของเครื่องชุดสั่งงานฝั่งลูกค้า จะปรากฏคำว่า Thank you 003 เป็นการยืนยันว่า เครื่อง Server นั้น ได้รับข้อมูลการเรียกเก็บเงินของโต๊ะที่ 3 เรียบร้อยแล้ว ดังรูปที่ 4.23



รูปที่ 4.23 แสดงหน้าจอ LCD เมื่อได้รับการยืนยันการเรียกเก็บเงินของลูกค้าโต๊ะที่ 3 จากเครื่อง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดลองเรื่อง การแสดงผลของเครื่อง Server และการแสดงผลตอบรับที่จอLCD กรณีที่ ข้อมูลที่ส่งมาจากชุดสั่งงานของลูกค้าไม่ถูกต้อง (รหัสอาหาร > 050 , หมายเลขโต๊ะ > 009)

ขั้นตอนการทดลอง

1. ทำการทดลองต่อจากการทดลองที่ 4.4 โดยทำการส่งข้อมูลจากชุดสั่งงานของลูกค้าอีกครั้ง โดยให้ข้อมูลมีค่าเท่ากับ 085001003

ผลการทดลอง



รูปที่ 4.24 แสดงหน้าจอ LCD เมื่อไม่ได้รับการยืนยันการรับข้อมูลจากเครื่อง Server เนื่องจากรหัสอาหารไม่มีในฐานข้อมูล

หน้าจอ LCD ของเครื่องชุดสั่งงานฝั่งลูกค้า ก็จะปรากฏคำว่า Error เนื่องจากไม่มีการตอบยืนยันจากเครื่อง Server เพราะข้อมูลที่เครื่อง Server รับได้นั้น เป็นข้อมูลที่มีรหัสอาหารเกิน 050 ส่วนของหน้าจอแสดงผลของเครื่อง Server นั้น จะไม่มีการเปลี่ยนแปลงใดๆ

2. ทำการส่งข้อมูลจากชุดสั่งงานของลูกค้าอีกครั้ง โดยให้ข้อมูลมีค่าเท่ากับ 002001090

ผลการทดลอง

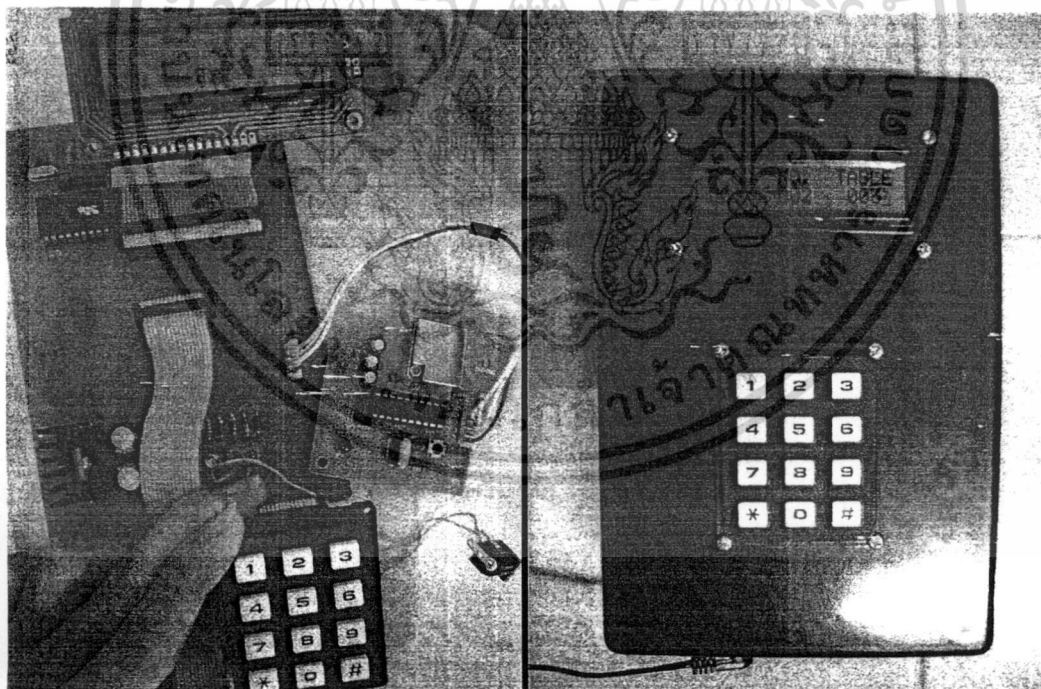
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 แสดงหน้าจอ LCD เมื่อไม่ได้รับการยืนยันการรับข้อมูลจากเครื่อง Server
เนื่องจากหมายเลขโต๊ะที่ไม่มีในโปรแกรม

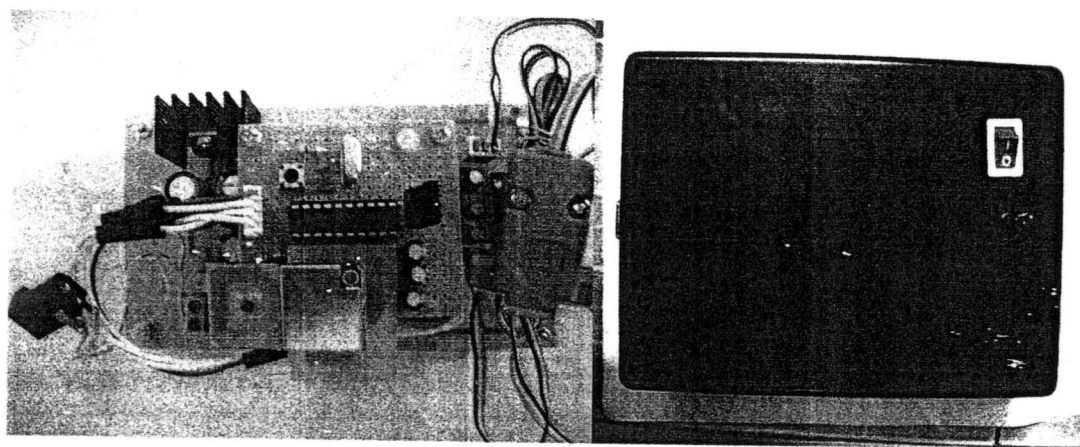
จากรูปจะเห็นได้ว่า ผลการทดลองนั้น จะเหมือนผลการทดลองในข้อที่ 1 คือ หน้าจอ LCD ของเครื่องชุดสั่งงานฝั่งลูกค้า จะปรากฏคำว่า Error เนื่องจากไม่มีการตอบยืนยันจากเครื่อง Server เพราะข้อมูลที่เครื่อง Server รับได้นั้น เป็นข้อมูลที่มีรหัสโต๊ะที่เกิน 009

ในส่วนของหน้าจอแสดงผลของเครื่อง Server ให้ผลเหมือนการทดลองในข้อที่ 1 คือไม่มีการเปลี่ยนแปลงใดๆ



รูปที่ 4.26 แสดงชุดสั่งงานของลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.27 แสดงภาครับทางฝั่ง Server



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

- ชุดสั่งงานของลูกค้า

จากการทำงานของชุดสั่งงานของลูกค้า ซึ่งมีการทำงานผ่านคีย์แพดและแสดงผลที่ LCD นั้น ทำให้ลูกค้าสามารถระบุชนิดอาหาร จำนวนที่สั่ง และโต๊ะที่สั่งได้โดยผ่านรหัส 9 ตัวที่จะส่งออกไป นอกจากนี้ยังรอกการตอบกลับจาก Server โดยการตอบกลับนั้นจะบ่งบอกว่า Server ได้รับความต้องการของลูกค้าโดยจะแสดงผลที่ LCD เช่นกัน ซึ่งมีดังนี้ คือ

1. Server รับทราบรายการอาหารที่ลูกค้าสั่ง ซึ่งจะตอบกลับว่า Received ตามด้วยรหัสอาหาร
2. กรณีลูกค้าต้องการลบรายการอาหารที่ส่งไปแล้ว จะส่ง รหัสอาหารที่จะลบ ตามด้วยจำนวนอาหาร 000 และหมายเลขโต๊ะ แล้ว Server จะตอบกลับว่า Cancel ตามด้วยรหัสอาหาร
3. กรณีลูกค้าจะเรียกเก็บเงิน จะส่งรหัสอาหาร000 จำนวนอาหาร000 และตามด้วยหมายเลขโต๊ะ แล้ว Server จะตอบกลับว่า Thank you ตามด้วยหมายเลขโต๊ะ
4. กรณีมีความผิดพลาดในการส่งข้อมูล หรือ ข้อมูลที่ลูกค้าส่งนั้น ไม่มีในฐานข้อมูลจะแสดงคำว่า Error ที่ LCD แล้วให้ลูกค้าทำการส่งข้อมูลใหม่

ปัญหาของชุดสั่งงานของลูกค้า

1. ความสว่างของจอ LCD บางครั้งจออาจมืดหรือสว่างไป ซึ่งในการใช้งานจริงลูกค้าอาจไม่ทราบว่าต้องทำอะไร
2. การที่ใช้ LCD ชนิด 2 บรรทัดแสดงผลนั้น ทำให้การที่จะเขียน โปรแกรมเพิ่มรายละเอียดนั้นเช่น อาหารไม่ใส่ผัก หรือไม่เผ็ดมาก มีข้อจำกัด ซึ่งการที่ใช้ LCD ชนิดนี้ก็เพราะราคาถูกกว่า

วิธีแก้ไขปัญหา

1. ก่อนที่จะให้บริการทุกครั้งควรตรวจสอบความสว่างของหน้าจอโดยการปรับที่ตัวด้านทานปรับค่าได้
2. ถ้าหากมีงบประมาณสูงขึ้น การใช้ LCD ที่แสดงผลได้มากขึ้นจะทำให้ สามารถระบุความละเอียดของข้อมูลอาหารได้มากขึ้น

บรรณานุกรม

1. ชีรบูลย์ หล่อวิเชียรรุ่ง , นคร ภัคศิชาติ , ชัยวัฒน์ ลัมพรจิตรวิไล “ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษาซี” , บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด , กรุงเทพฯ, 2521 .
2. รศ.สมยศ จุณณปิยะ , “การประยุกต์ใช้งาน ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ” คณะวิศวกรรมศาสตร์ , สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2546.
3. อุดม รานอก , “ภาษาซีสำหรับงานควบคุมไมโครคอนโทรลเลอร์ MCS-51 ” , นนทบุรี , ไอดีซีฯ, 2548.
4. สัจจะ จรัสรุ่งวิীর , จักรพงษ์ สุขประเสริฐ “เริ่มต้นอย่างมืออาชีพด้วย Delphi 7 ฉบับสมบูรณ์” . บริษัท เอช เอ็น กรุป จำกัด , กรุงเทพฯ ,2546.

1.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้