ระบบควบคุมการจับเวลาและบันทึกเวลาการแข่งขันเรือพาย

โดย

1. นายชวลิต	จินดาประทุม	รหัสประจำตัว B5047161
2. นายวชิระ	คมกล้า	รหัสประจำตัว B5140435
3. นายวิชิต	บุญมาก	รหัสประจำตัว B5144068

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 427494 โครงงานศึกษาวิศวกรรมโทรคมนาคม

และวิชา 427499 โครงงานวิศวกรรมโทรคมนาคม

หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม หลักสูตรปรับปรุง พ.ศ. 2546

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ประจำภาคการศึกษาที่ 3 ปีการศึกษา 2555

ระบบควบคุมการจับเวลาและบันทึกเวลาการแข่งขันเรือพาย

คณะกรรมการสอบโครงงาน

(รองศาสตราจารย์ ดร.รังสรรค์ วงศ์สรรค์)
กรรมการ/อาจารย์ที่ปรึกษาโครงงาน

(ผู้ช่วยศาสตราจารย์ ดร.ชุติมา พรหมมาก)
กรรมการ

(อ.เศรษฐวิทย์ ภูฉายา)
กรรมการ

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับรายงานโครงงานฉบับนี้ เป็นส่วนหนึ่งของการศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมโทรคมนาคม รายวิชา 427499 โครงงานวิศวกรรมโทรคมนาคม และรายวิชา 427494 โครงงานศึกษาวิศวกรรมโทรคมนาคม ประจำปีการศึกษา 2555

โครงงานเรื่อง ระบบควบคุมการจับเวลาและบันทึกเวลาการแข่งขันเรือพาย

โดย	1. นายชวลิต	จินดาประทุม	รหัส B5047161
	2. นายวชิระ	คมกล้า	รหัส B5140435
	3. นายวิชิต	บุญมาก	รหัส B5144068
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร. รังสรรค์ วงศ์สรรค์		
สาขาวิชา	วิศวกรรมโทรคมนาคม		
ภาคเรียนที่	1, 2 และ 3/2554 และ1/2555		

บทคัดย่อ

การส่งข้อมูลผ่านวิทยุสื่อสารเป็นรูปแบบหนึ่งในหลายๆรูปแบบการติดต่อสื่อสาร การสื่อสารในรูปแบบนี้คือการรับส่งข้อมูลผ่านตัวกลางที่เรียกว่าโมเด็ม (Modem) ซึ่งใช้ทฤษฎีการทำงานแบบ FSK โดยแปลงข้อมูลที่รับมาเป็นสัญญาณเสียงสองความถี่ ตามข้อมูลดิจิตอลที่รับมาเพื่อส่งออกไปทางวิทยุสื่อสาร ในขณะเดียวกันก็ทำการแปลงสัญญาณที่ได้รับจากวิทยุสื่อสารกลับมาเป็นข้อมูลอีกครั้งหนึ่ง

จากหลักการดังกล่าวข้างต้น คณะผู้จัดทำโครงงานได้นำมาประยุกต์ใช้ในการจับเวลาผ่านระบบควบคุมการจับเวลาและบันทึกเวลา สำหรับการแข่งขันกีฬาเรือพาย โดยการทำงานเริ่มจากส่งข้อมูลเวลาที่ถูกแปลงเป็นสัญญาณสียงสองความถี่ ส่งผ่านวิทยุสื่อสารไปยังภาครับซึ่งเป็นจุดแสดงผล เพื่อให้จุดแสดงผลได้ทราบว่าเรือถูกปล่อยออกจากจุดสตาร์ทแล้ว เมื่อเรือเคลื่อนที่ถึงจุดเส้นชัยระบบจะส่งค่าเวลาอีก 1 ชุดมาที่จุดแสดงผล จากนั้นระบบแสดงผลจะนำเวลาจากจุดสตาร์ทมาลบกับเวลาของจุดเส้นชัยทำให้ได้เวลาที่แท้จริงที่ใช้ในการเคลื่อนที่จากจุดสตาร์ทจนถึงจุดเส้นชัย พร้อมทั้งพิมพ์ไฟล์ออกมาเป็นไฟล์ PDF ที่แสดงข้อมูลของประเภทการแข่งขัน ชื่อของเรือ ระยะทางและเวลาของเรือแต่ละลำที่ใช้ไป เพื่อนำมาพิจารณาหาผู้ชนะในการแข่งขันต่อไป

กิตติกรรมประกาศ

การทำโครงงานฉบับนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์ในการให้คำปรึกษาในด้านต่างๆ ในระหว่างการดำเนินการจากบุคคลหลายท่านที่ได้ให้ความช่วยเหลืออย่างดีเสมอมาอันได้แก่

-รองศาสตราจารย์ ดร. รังสรรค์ วงศ์สรรค์ อาจารย์ที่ปรึกษาที่ให้คำปรึกษาในด้านต่างๆทั้งทางด้านวิชาการและการปฏิบัติงาน ซึ่งท่านได้คอยดูแลควบคุมการทำงานอย่างใกล้ชิดและคอยให้กำลังใจเสมอมา

-คุณอำนวย ทีจันทึก วิศวกร มหาวิทยาลัยเทคโนโลยีสุรนารีที่ให้คำปรึกษาด้านการเขียนโปรแกรม PROPELLER TOOL โดยใช้ภาษา SPIN และแนะนำการใช้งานอุปกรณ์อิเล็กทรอนิกส์

-อาจารย์วิชัย ศรีสุรักษ์ อาจารย์ที่ให้คำแนะนำในการออกแบบฮาร์ดแวร์

-คุณชาตรี แก้วอุดร นักเทคโนโลยีการศึกษา มหาวิทยาลัยเทคโนโลยีสุรนารีที่ให้คำปรึกษาด้านการเขียนโปรแกรม VISUAL C# 2010 EXPRESS

สุดท้ายนี้คณะผู้จัดทำขอกราบขอบพระคุณบิดา มารดา ที่อบรมเลี้ยงดู และให้โอกาสทางการศึกษา และคอยให้กำลังใจสนับสนุนด้วยดีตลอดมาอย่างหาที่เปรียบมิได้

จึงเห็นสมควรที่จะมอบคุณงามความดี และเกียรติคุณเหล่านี้แด่ท่านที่ได้กล่าวมานี้ รวมถึงบุคคลที่มิได้กล่าวมา ณ ที่นี้ด้วย

นายชวลิต จินดาประทุม
นายวชิระ คมกล้า
นายวิชิต บุญมาก

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญรูป	จ
บทที่1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตการทำงาน	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่2 หลักการและทฤษฎีที่เกี่ยวข้อง	
2.1 กล่าวนำ	4
2.2 กติกาการแข่งขันเรือพาย และการจับเวลา	4
2.3 มัลติคอร์ไมโครคอนโทรลเลอร์ 32 บิต โพรเพลเลอร์ Parallax P8x32A	5
2.4 ภาษาของโปรแกรมที่ใช้พัฒนาโพรเพลเลอร์ Parallax P8x32A	10
2.5 ตัวนับสัญญาณเวลานาฬิกาจริง (Real Time Clock PCF8583)	13
2.6 การ Modulation แบบ FSK	15
2.7 การเขียนโปรแกรมโดยใช้ VISUAL C# 2010 EXPRESS	18
2.8 กล่าวสรุป	19
บทที่3 การออกแบบและพัฒนาระบบ	
3.1 กล่าวนำ	20
3.2 การออกแบบระบบ Hard ware	
3.2.1 การออกแบบภาคส่ง	20
3.2.2 การออกแบบภาครับ	26
3.3 การพัฒนาซอฟแวร์ (Soft ware)	
3.3.1 การพัฒนาซอฟแวร์ของไมโครคอนโทรลเลอร์ Parallax P8x32A ด้วยภาษาสปิน	30

สารบัญ (ต่อ)

เรื่อง หน้า

3.3.2 การติดตั้งโปรแกรม Propeller Tools เพื่อเขียนโปรแกรมภาษาสปิน 34

3.3.3 การพัฒนาโปรแกรมบนเครื่องคอมพิวเตอร์ 38

3.3.4 การติดตั้งโปรแกรม VISUAL C# 2010 EXPRESS 40

บทที่4 ผลการทดสอบ

4.1 กล่าวนำ 44

4.2 ผลการเปรียบเทียบรูปคลื่นของการรับและส่งข้อมูลแบบ FSK ผ่านวิทยุสื่อสาร 44

4.3 ผลการทดสอบการแสดงผลที่จอ LCD (ภาคส่ง) กับ Hyper Terminal (ภาครับ) 48

4.4 การทดสอบไมโครคอนโทรลเลอร์เชื่อมต่อกับคอมพิวเตอร์ เพื่อแสดงผลด้วยโปรแกรม VISUAL C# 2010 EXPRESS 49

4.5 ผลการเปรียบเทียบความถูกต้องของการส่งและรับที่ระยะต่างๆ 52

4.6 การทดสอบการใช้งานจริงที่กีฬาแห่งชาติเชียงใหม่เกมส์ 54

4.7 กล่าวสรุป 56

บทที่5 บทสรุป

5.1 บทสรุปของโครงงาน 57

5.2 ปัญหาที่มีระหว่างทำการทดสอบ 57

5.3 ข้อเสนอแนะ 58

ภาคผนวก

ภาคผนวก ก Code 71

ภาคผนวก ข คู่มือการใช้งาน 88

ประวัติผู้เขียน 96

บรรณานุกรม 97

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูปภาพ

รายการ	หน้า
รูปที่ 2.1แสดงตำแหน่งขาและรูปร่างของไมโครคอนโทรลเลอร์โพรเพลเลอร์ Parallax P8x32A แบบตัวถัง DIP40 ขา	5
รูปที่ 2.2 Block diagram P8X32A	9
รูปที่ 2.3 Propeller Tool	10
รูปที่ 2.4 Block diagram PCF8583	14
รูปที่ 2.5 แสดงตำแหน่งขาของตัวนับสัญญาณเวลานาฬิกาจริง (Real Time Clock)	14
รูปที่ 2.6 การสร้างสัญญาณ FSK	15
รูปที่ 2.7 การรวมกันของสัญญาณ ASK สองสัญญาณ	16
รูปที่ 2.8 แสดงการ มอดูเลชันแบบ FSK	17
รูปที่ 2.9 รูปแสดงตัวอย่างโปรแกรม VISUAL C# 2010 EXPRESS	19
รูปที่ 3.1 การออกแบบระบบภาคส่ง	21
รูปที่ 3.2 สวิทซ์ควบคุมการจับเวลา	22
รูปที่ 3.3 อินเตอร์เฟสภาคส่ง	23
รูปที่ 3.4 การเชื่อมต่อตัวนับสัญญาณเวลานาฬิกาจริง (Real Time Clock)	24
รูปที่ 3.5 จอแสดงผล (LCD)	25
รูปที่ 3.6 การออกแบบระบบภาครับ	26
รูปที่ 3.7 อินเตอร์เฟสภาครับ	27
รูปที่ 3.8 วงจรอิเล็กทรอนิกส์ (schematic diagram) ของระบบ	29
รูปที่ 3.9 แผนผังการทำงานของโปรแกรมส่วนส่งโทนข้อมูล	30
รูปที่ 3.10 แผนผังการทำงานของโปรแกรมส่วนอ่านค่าเวลา	31
รูปที่ 3.11 แผนผังการทำงานของโปรแกรมส่วนแสดงผลผ่านจอ LCD	33
รูปที่ 3.12 แผนผังการทำงานของโปรแกรมบนคอมพิวเตอร์	38
รูปที่ 4.1 แสดงการจัดอุปกรณ์ทดสอบการส่งผ่านความถี่ผ่านวิทยุสื่อสาร	44
รูปที่ 4.2 แสดงการรับส่งความถี่ 500 Hz	45
รูปที่ 4.3 แสดงการรับส่งความถี่ 1200 Hz	46
รูปที่ 4.4 แสดงการรับส่งความถี่ 2200 Hz	46
รูปที่ 4.5 แสดงการรับส่งความถี่ 3000 Hz	47
รูปที่ 4.6 แสดงผลที่จอ LCD (ภาคส่ง)	48

สารบัญรูปภาพ (ต่อ)

รายการ	หน้า
รูปที่ 4.7 แสดงผลที่ Hyper Terminal (ภาครับ)	48
รูปที่ 4.8 แสดงการทดสอบการส่งข้อมูลเพื่อแสดงผลผ่าน VISUAL C# 2010 EXPRESS	49
รูปที่ 4.9 การ Install โปรแกรม	50
รูปที่ 4.10 หน้าต่างโปรแกรมส่วนที่ 1	50
รูปที่ 4.11 หน้าต่างโปรแกรมส่วนที่ 2	51
รูปที่ 4.12 ผลการทดสอบการส่งข้อมูลเพื่อแสดงผลที่คอมพิวเตอร์	51
รูปที่ 4.13 ตารางแสดงผลการเปรียบเทียบความถูกต้องของข้อมูล	52
รูปที่ 4.14 กราฟแสดงเปอร์เซ็นต์การเปรียบเทียบความถูกต้องของข้อมูล ในการส่งที่ระยะต่างๆ	53
รูปที่ 4.15 การทดสอบการใช้งานจริงที่กีฬาแห่งชาติเชียงใหม่เกมส์	54

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่1
บทนำ

1.1ความเป็นมาและความสำคัญของปัญหา

เรือพายเป็นกีฬาที่ใช้ระยะทางและเวลาเป็นปัจจัยสำคัญในการแข่งขัน ผลแพ้-ชนะตัดสินกันด้วยเวลา ปัญหาสำคัญคือวิธีการจับเวลามักมีความคลาดเคลื่อน เพราะระยะทางจากจุดปล่อยเรือถึงจุดสิ้นสุดที่เรือถึงเส้นชัย มีระยะทางที่ไกลและไม่สามารถตรวจสอบความถูกต้องของการตัดสินย้อนหลังได้

ดังนั้นโครงการนี้จึงได้ทำการศึกษาเกี่ยวกับระบบการจับเวลาและส่งเวลาด้วยเครื่องส่งวิทยุในย่านความถี่ VHF จากจุดปล่อยเรือไปยังจุดแสดงผล ซึ่งที่จุดแสดงผลระบบจะบันทึกเวลาที่รับได้และสามารถพิมพ์เวลาของการแข่งขันได้ทันที สำหรับเพื่อใช้ในการตัดสินหาผู้ชนะต่อไป

1.2 วัตถุประสงค์ของโครงการ

1. ศึกษาการทำงานของการติดต่อสื่อสารแบบ FSK
2. ศึกษาการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล
3. ศึกษาและออกแบบวงจรเชื่อมต่อไมโครคอนโทรลเลอร์ (Parallax P8X32A)
4. ศึกษาการเขียนโปรแกรมควบคุมไมโครคอนโทรเลอร์ ที่ใช้ภาษาสปีน(Spin)
5. ศึกษาและออกแบบโปรแกรมแสดงผล (Visual C# 2010 Express)
6. ศึกษาการเขียนโปรแกรมแสดงผล (Visual C# 2010 Express)
7. ออกแบบและสร้างระบบจับเวลาและบันทึกเวลาในการแข่งขันเรือพาย

1.3 ขอบเขตการทำงาน

1. สร้างวงจรชุดส่งเวลาและชุดรับเวลา
2. เขียนโปรแกรมเพื่อควบคุมชุดส่งเวลาและชุดรับเวลา
3. เขียนโปรแกรมเพื่อแสดงผลของเวลา
4. ติดตั้งชุดส่งเวลาและชุดรับเวลาเข้ากับวิทยุ (VHF)
5. ทดลองใช้งานจริงของระบบส่งเวลา รับเวลาและระบบแสดงผล

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาการทำงานของระบบการมอดูเลชันแบบ FSK
2. ศึกษาการทำงานของไมโครคอนโทรลเลอร์ (Parallax P8x32A)
3. ศึกษาการเขียนโปรแกรม ภาษา Spin
4. จัดหาอุปกรณ์
5. ศึกษาและทดลองอุปกรณ์
6. เขียนโปรแกรมเพื่อควบคุมอุปกรณ์
7. ศึกษาการเขียนโปรแกรม ภาษา Visual C# 2010 Express
8. เขียนโปรแกรมเพื่อแสดงผล
9. ทดลองใช้อุปกรณ์
10. แก้ไขจุดบกพร่องของอุปกรณ์
11. สรุปและทำรายงาน
12. นำเสนอโครงงาน

มหาวิทยาลัยเทคโนโลยีสุรนารี

ตาราง 1.1 ขั้นตอนการดำเนินงาน

กิจกรรม	พ.ศ. 2555 - พ.ศ. 2556						
	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. ศึกษาการทำงานของระบบ Packet radio	↔						
2. ศึกษาการทำงานของไมโครคอนโทรลเลอร์ (Parallax P8x32A)		↔					
3. ศึกษาการเขียนโปรแกรม ภาษา Spin			↔				
4. จัดหาอุปกรณ์			↔				
5. ศึกษาและทดลองอุปกรณ์			↔	↔			
6. เขียนโปรแกรมเพื่อควบคุมอุปกรณ์				↔			
7. ศึกษาการเขียนโปรแกรมภาษา Visual C# 2010				↔	↔		
8. เขียนโปรแกรมเพื่อแสดงผล					↔		
9. ทดลองใช้อุปกรณ์					↔	↔	
10. แก้ไขจุดบกพร่องของอุปกรณ์					↔	↔	
11. สรุปและทำรายงาน						↔	↔
12.นำเสนอโครงงาน							↔

มหาวิทยาลัยเทคโนโลยีสุรนารี

1.5 ผลที่คาดว่าจะได้รับ

1. ได้เรียนรู้หลักการทำงานของการมอดูเลชันแบบ FSK
2. มีความรู้เกี่ยวกับการแปลงสัญญาณอนาล็อกเป็นดิจิตอล และดิจิตอลเป็นอนาล็อก
3. มีความรู้เกี่ยวกับการออกแบบวงจรเชื่อมต่อไมโครคอนโทรลเลอร์
4. มีความเข้าใจในการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์
5. มีทักษะในการทำงานเชิงวิศวกรรมมากขึ้นและการทำงานเป็นทีมได้
6. สามารถนำอุปกรณ์ที่เสนอมาใช้งานได้อย่างมีประสิทธิภาพ

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 กล่าวนำ

ปัจจุบันการแข่งขันเรือพายนั้นมีการใช้งานอุปกรณ์จับเวลาอยู่ 2 ระบบด้วยกันคือ นาฬิกาจับเวลาและระบบกล้องโฟโตฟินิช (photo finished)

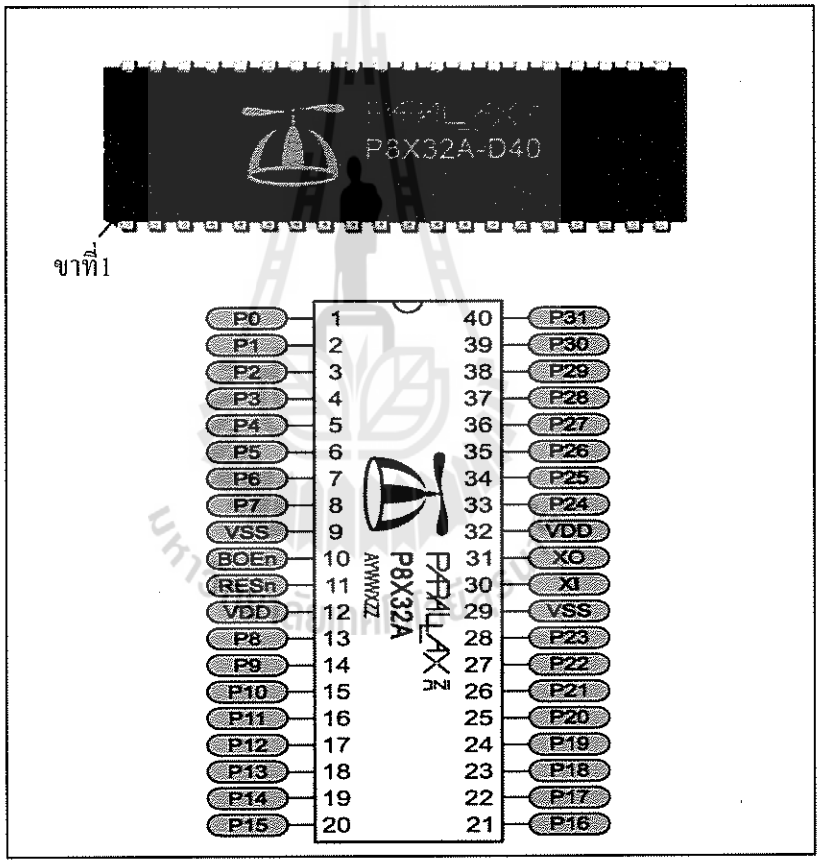
2.2 กติกาการแข่งขันเรือพาย และการจับเวลา

1.นาฬิกาจับเวลา จะใช้กรรมการสองคนในการจับเวลา โดยกรรมการจะอยู่ที่จุดสตาร์ทหนึ่งคนและจุดเส้นชัยหนึ่งคนเมื่อมีการปล่อยเรือออกจากจุดสตาร์ท กรรมการทั้งสองคนจะกดสตาร์ทนาฬิกาจับเวลาพร้อมกันเมื่อเรือเข้าเส้นชัยกรรมการทั้งสองจะกดนาฬิกาเพื่อหยุดเวลา หลังจากนั้นกรรมการทั้งสองจะนำเวลาของแต่ละคนมาเปรียบเทียบกัน เพื่อดูเวลาที่เรือใช้ทั้งหมด ตั้งแต่ออกจากจุดสตาร์ทจนถึงจุดเส้นชัย ซึ่งวิธีการจับเวลาแบบนี้ต้องใช้กรรมการจับเวลาอย่างน้อยหนึ่งคนต่อเรือหนึ่งลำทำให้ต้องใช้กรรมการจำนวนมากในการจับเวลาต่อการแข่งขันหนึ่งครั้ง และเวลาที่ใช้ในแต่ละจุดอาจมีความคลาดเคลื่อนได้เพราะทำการกดไม่พร้อมกัน

2.ระบบกล้องโฟโตฟินิช จะใช้กรรมการสองคน โดยเมื่อมีการแข่งขันกรรมการคนที่หนึ่งจะทำการกดปุ่มสตาร์ทกล้องเพื่อเริ่มบันทึกภาพและจับเวลาเมื่อเรือออกจากจุดสตาร์ท และเมื่อเรือเข้าเส้นชัยกรรมการคนที่สองจะกดหยุดกล้องและหยุดเวลาที่จุดเส้นชัย ทำให้ได้เวลาจากกล้องเป็นตัวอ้างอิงและได้เวลาที่ใช้ในการแข่งขันออกมา โดยระบบนี้มีความแม่นยำกว่าวิธีแรกแต่จะต้องใช้ค่าใช้จ่ายมากและการติดตั้งระบบทำได้ยุ่งยากมากและระบบนี้จะใช้กับการแข่งขันรายการใหญ่ๆเท่านั้นเช่นการแข่งขันกีฬาแห่งชาติ ซึ่งทั้งสองระบบไม่สามารถเช็คสถิติ แต่ละระยะได้ เช่นระยะแข่งขันจริง 1000 เมตร โดยระบบทั้งสองจะอยู่ที่จุดสตาร์ท และจุดเส้นชัยเท่านั้น ซึ่งไม่สามารถเช็คระยะที่จุด 500 เมตร และไม่สามารถเช็คระยะที่จุดต่างๆได้ทำให้ไม่สามารถทราบสถิติของเรือแต่ละลำได้ ว่าแต่ละจุดที่ผ่านมาใช้ระยะเวลาไปเท่าใด ซึ่งปัญหาเหล่านี้จึงเป็นที่มาของโครงงานนี้เพื่อช่วยแก้ปัญหาดังกล่าว

2.3 มัลติคอร์ไมโครคอนโทรลเลอร์ 32 บิต โพรเพลเลอร์ Parallax P8X32A

มัลติคอร์ไมโครคอนโทรลเลอร์ เป็นไมโครคอนโทรลเลอร์ 32 บิตโดยโพรเพลเลอร์มีซีพียูภายใน 8 ตัวหรือ 8 ค็อก ที่สามารถทำงานแยกจากกันอย่างอิสระหรือร่วมกันทำงานก็ได้ นับเป็นแนวคิดใหม่ที่ร่วมสมัยและนับเป็นการปฏิวัติวงการไมโครคอนโทรลเลอร์ 32 บิตครั้งสำคัญ การพัฒนาโปรแกรมสำหรับโพรเพลเลอร์ทำได้ด้วยโปรแกรมภาษาใหม่ที่เรียกว่า สปิน (spin) และภาษาแอสแซมบลี โดยภาษาสปินนั้นเป็นภาษาสูงที่มีการทำงานแบบออบเจ็กต์



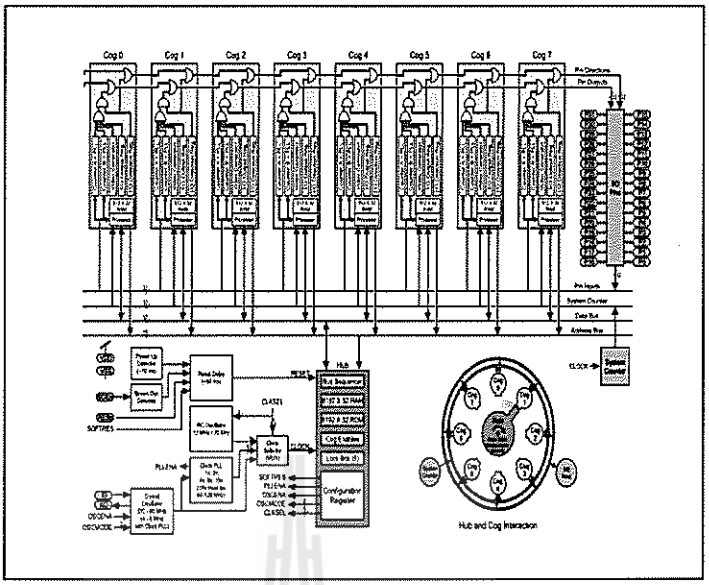
รูปที่ 2.1 แสดงตำแหน่งขาไมโครคอนโทรลเลอร์โพรเพลเลอร์ Parallax P8X32A แบบตัวถัง DIP40 ขา

คุณสมบัติประกอบด้วย 8 ซีพียู หรือเรียกว่า 8 ค็อก ที่สามารถทำงานได้พร้อมกันอย่างเป็นอิสระ โดยมีการควบคุมการใช้ทรัพยากรร่วมกันผ่านทางตัวเชื่อมโยงกลางหรือ central hub

- มีความเร็วในการทำงานสูงและด้วยการทำงานที่เป็นอิสระของแต่ละค็อก ทำให้สามารถรองรับการตอบสนองต่อเหตุการณ์ต่างๆ ที่เกิดขึ้นในระบบได้อย่างเร็วเพียงพอ จึงไม่ต้องใช้ระบวนการอินเตอร์รัปต์ช่วย ทำให้การเขียนโปรแกรมเพื่อรองรับการทำงานในแต่ละเหตุการณ์ลดความซับซ้อนลงได้อย่างมาก
- มีการใช้สัญญาณนาฬิกาของระบบร่วมกัน ทำให้สามารถอ้างอิงเรื่องเวลาหลักเดียวกันได้ ทำให้การทำงานของแต่ละค็อกมีจังหวะที่สอดคล้องกัน
- ภาษาสปินซึ่งมีลักษณะเป็นโปรแกรมภาษาสูงแบบออบเจ็กต์ได้รับการออกแบบให้ง่ายต่อการเรียนรู้ และสามารถรองรับการทำงานของโพรเพลเลอร์ได้อย่างมีประสิทธิภาพสูงสุด
- ภาษาแอสแซมบลีของโพรเพลเลอร์ได้จัดเตรียมคำสั่งที่ใช้ในการตรวจสอบเงื่อนไข และมีตัวแปรที่ใช้ในการตรวจสอบการทำงานอย่างสมบูรณ์ ทั้งยังสามารถรองรับการทำงานในลักษณะที่ต้องมีการตัดสินใจพร้อมๆกันหลายเงื่อนไขด้วย พร้อมกันนั้นยังมีการคำนึงถึงการลดสัญญาณรบกวนและความเพี้ยนของสัญญาณที่เกิดขึ้นจากการประมวลผลคำสั่ง และตัวคำสั่งเองมีรูปแบบการทำงานที่ตรงไปตรงมา ชัดเจน ส่งผลให้ผู้พัฒนาโปรแกรมสามารถลดเวลาในการเขียนโปรแกรมลงอย่างมาก
- แต่ละค็อกจะประกอบด้วยตัวประมวลผลหรือโปรเซสเซอร์ที่มีการทำงานอย่างเป็นอิสระมีแรม 2 กิโลไบต์ ที่กำหนดให้ทำงานเป็นรีจิสเตอร์ 32 บิต จะได้ทั้งสิ้นถึง 512 ตัว มีโมดูลตัวนับความสามารถสูงที่ทำงานร่วมกับเฟสล็อกลูป ทำให้แต่ละค็อกทำงานได้เร็วถึง 800 MHz มีวงจรกำเนิดสัญญาณภาพและส่วนควบคุมพอร์ตอินพุตเอาต์พุตที่เป็นอิสระ
- สัญญาณนาฬิกาของระบบมาได้จาก 3 แหล่งคือ วงจรออสซิลเลเตอร์ RC ภายในเลือกได้ระหว่าง 12 ถึง 20 MHz จากแหล่งกำเนิดสัญญาณนาฬิกาภายนอก และจากการทวีคูณความถี่ของคริสตอลด้วยวงจรเฟสล็อกลูป โดยปกติแล้วจะเลือกใช้คริสตอล 5 MHz แล้วเลือก PLLx16 ทำให้ได้สัญญาณนาฬิกาของระบบที่มีความถี่ 80 MHz ในขณะที่ส่วนเชื่อมโยงกลางจะทำงานด้วยความถี่ที่ลดลงครึ่งหนึ่งของสัญญาณนาฬิกาหลัก

- มีขาพอร์ตอินพุตเอาต์พุตรวม 32 ขา โดยกำหนดให้ใช้งาน และอีก 2 ขาสำหรับการดาวน์โหลดโปรแกรม สามารถขับกระแสซิงก์และซอร์สสูงสุด 40 mA ต่อขา
- โพรเพลเลอร์ใช้หน่วยความจำอีพรอมภายนอกในการเก็บโปรแกรมใหม่ในหน่อยความจำโปรแกรม
- การดาวน์โหลดโปรแกรมทำได้ง่ายมากเพียงต่อเข้ากับวงจรเชื่อมต่อพอร์ตอนุกรม อาทิวงจรของไอซี MAX3232 หรือต่อผ่านชิปแปลงสัญญาณพอร์ต USB เป็นพอร์ตอนุกรมอย่าง FT232RL ไม่ต้องการเครื่องโปรแกรมใดเพิ่มเติม
- ด้วยความเร็วในการทำงานที่สูง และมีส่วนกำเนิดสัญญาณภาพที่มากถึง8ชุด ทำให้เหมาะสมอย่างมากในการนำโพรเพลเลอร์ไปใช้ในการกำเนิดสัญญาณภาพไม่ว่าจะแสดงผลด้วยจอโทรทัศน์ด้วยสัญญาณวีดีโอ หรือแสดงผลด้วยจอ VGA ด้วยสัญญาณแม่สีแสง นั้นคือพื้นฐานหลักในการสร้างเครื่องเล่นวีดีโอเกม และการสร้างระบบนำเสนอ (presentation) ภาพกราฟฟิกด้วยไมโครคอนโทรลเลอร์เพียงตัวเดียว
- สามารถเชื่อมต่อกับคีย์บอร์ดและเม้าได้ และเมื่อรวมกับความสามารถการสร้างสัญญาณภาพได้ จึงสามารถนำโพรเพลเลอร์ไปสร้างเครื่องคอมพิวเตอร์ขนาดเล็กแบบใช้จอโทรทัศน์เป็นตัวแสดงผลได้
- ใช้ไฟเลี้ยงในย่าน 2.7 ถึง 3.6 V กระแสไฟฟ้าสูงสุดเมื่อขับโหลดเต็มที่คือ 300 mA
- มีตัวถังให้เลือกใช้ 3 แบบคือ DIP 40ขา,LQFP 44 ขา และ QFN 44 ขา
- มีซอฟต์แวร์ Propeller IDE ที่มีประสิทธิภาพสูง สามารถสร้างระบบไฟล์ที่มีรูปการต่อวงจรได้ ทำให้นำโค้ดไปถ่ายทอดต่อได้สะดวก และช่วยให้การเรียนรู้ทำได้อย่างสมบูรณ์ครบวงจรและที่สำคัญซอฟต์แวร์แจกฟรี สามารถดาวน์โหลดได้ที่ www.parallax.com

ตำแหน่งขา	หน้าที่
P0-P31	ขาพอร์ตอินพุตเอาต์พุตเอนกประสงค์ พอร์ต A สามารถจ่ายกระแสซอร์ส/ซิงก์ ได้ 40 mA ที่แรงดัน 3.3 VDC ระดับลอจิกมีจุดตัดที่ 1/2 VDC หรือ 16 VDC ที่แรงดันไฟเลี้ยง 3.3 V ขาพอร์ตบางขายังมีหน้าที่พิเศษ เมื่อจ่ายไฟครั้งแรกหรือรีเซต(แต่สามารถสั่งใช้ขาอินพุตเอาต์พุตผ่านทางซอฟต์แวร์)ได้แก่ P28 เป็นขา SLC ของ I2C สำหรับการเชื่อมต่ออีพรอมภายนอก P29 เป็นขา SDA ของ I2C สำหรับการเชื่อมต่ออีพรอมภายนอก P30 เป็นขา Tx ส่งข้อมูล สำหรับการสื่อสารอนุกรมกับคอมพิวเตอร์และการดาวน์โหลดโปรแกรม P31 เป็นขา Rx รับข้อมูล สำหรับการสื่อสารอนุกรมกับคอมพิวเตอร์
VDD	ขาไฟบวก 3.3 V (2.7-3.6 VDC)
VSS	ขากราวด์
BOE	ขาเอ็นเอเบิล Brown out (รีเซตเมื่อมีแรงดันต่ำกว่าที่กำหนด)ทำงานที่ลอจิก “0” ถ้าขานี้เป็น “0” ขารีเซตจะทำหน้าที่เป็นขาเอาต์พุต เพื่อแสดงสถานะ แต่ยังสามารถส่งลอจิก “0” ให้ขารีเซตเพื่อรีเซตไมโครคอนโทรลเลอร์ได้ ถ้าให้ขานี้เป็น “1” ขานี้จะทำหน้าที่เป็นขาอินพุตแบบ มิตทริกเกอร์
RES	ขารีเซต ทำงานที่ลอจิก “0” เมื่อขานี้เป็น “0” Propeller จะถูกรีเซตค็อกทั้งหมดจะถูกดิสเอเบิ้ล ขาพอร์ตอินพุตเอาต์พุตจะอยู่ในสถานะลอย Propeller จะรีเซตระยะเวลา 50 ms เมื่อขารีเซตเปลี่ยนสถานะจาก “0” เป็น “1”
XI	ขาอินพุตของคริสตอล ใช้ต่อกับแหล่งกำเนิดสัญญาณนาฬิกาจากภายนอก (ขา XO ไม่ใช้งาน) หรือต่อกับขาด้านหนึ่งของคริสตอลหรือเรโซเนเตอร์ (ขาอีกข้างต่อกับ XO) โดยไม่จำเป็นต้องต่อตัวต้านทานหรือตัวเก็บประจุภายนอก
XO	ขาเอาต์พุตของคริสตอล การต่อคริสตอลเข้ากับขา XI และ XO จะต้องสัมพันธ์กับค่าที่กำหนดให้กับรีจิสเตอร์ CLK ด้วย



รูปที่ 2.2 Block diagram P8X32A

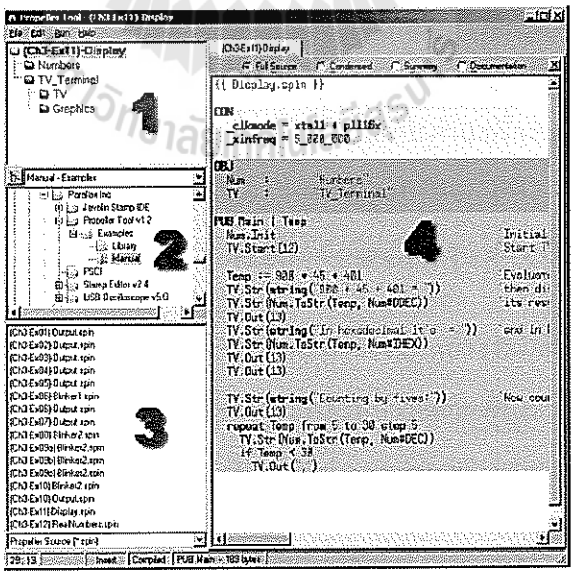
หลักการทำงานของโพรเพลเลอร์ รูปที่ 2.2 แสดงให้เห็นบล็อกไดอะแกรมการทำงานภายในของโพรเพลเลอร์ ซึ่งประกอบด้วยโปรเซสเซอร์ที่ทำงานแยกกันอิสระถึง 8 ชุดโดยจะเรียกโปรเซสเซอร์เหล่านี้ว่าค็อก หมายเลข 0 ถึง 7

- ค็อก ในแต่ละค็อกจะประกอบไปด้วยหน่วยความจำแรม 2 KB โดยกำหนดเป็นหน่วยความจำแบบ 32 บิต จำนวน 512 ตัว นอกจากนี้ภายในโปรเซสเซอร์แต่ละตัวยังมีโมดูลเคาร์เตอร์แบบพิเศษพร้อมเฟสล็อกลูปสองตัว โมดูลสร้างสัญญาณวีดีโอ รีจิสเตอร์พอร์ตอินพุตเอาต์พุต รีจีสเตอร์กำหนดทิศทางของพอร์ตอินพุตเอาต์พุต และรีจีสเตอร์ตัวอื่นๆ ซึ่งไม่ได้แสดงให้เห็นในบล็อกไดอะแกรมค็อกทั้ง 8 ตัว ทำงานด้วยวงจรกำเนิดสัญญาณนาฬิกาหลัก ซึ่งค็อกแต่ละตัวจะอ้างอิงการทำงานกันได้ด้วยสัญญาณนาฬิกา และจะเริ่มต้นทำงานพร้อมกันและใช้ทรัพยากรด้วยกัน ค็อกแต่ละตัวสามารถสั่งให้ทำงานหรือหยุดทำงานได้ในขั้นตอนการทำงานของโปรแกรม และสามารถควบคุมให้แต่ละค็อกทำงานไปพร้อมๆกันได้ โดยจะทำงานเป็นอิสระหรือ เชื่อมโยงถึงกันได้ผ่านหน่วยความจำแรมหลัก ซึ่งแยกไปต่างหากหน่วยความจำภายในค็อกแต่ละตัว เรียกว่าค็อกแรม โดยค็อกแรมจะแบ่งหน่วยความจำเป็นรีจีสเตอร์ขนาด 32 บิต จำนวน 512 ตัว สามารถใช้งานได้อย่างอิสระ ยกเว้นรีจีสเตอร์ 16 ตำแหน่งสุดท้ายซึ่งสงวนไว้สำหรับรีจีสเตอร์ฟังก์ชันพิเศษ เช่น รีจีสเตอร์เคาน์เตอร์ รีจีสเตอร์พอร์ต อินพุตเอาต์พุต เป็นต้น

- ฮับทำหน้าที่จัดระเบียบการทำงานของระบบทั้งหมด โดยจะยอมให้ค็อกทีละตัวเท่านั้นที่จะติดต่อกับทรัพยากรหลักของระบบ โดยจะหมุนเวียนติดต่อกับค็อกตั้งแต่หมายเลข 0 ถึง 7 แล้วกลับไปที่หมายเลย 0 ใหม่เป็นลักษณะวนรอบ ส่วนของฮับและระบบบัสของมั้นทำงานด้วยความเร็วครึ่งหนึ่งของสัญญาณนาฬิกาของทั้งระบบ ทำให้ค็อก1ตัวจะถูกติดต่อทุกๆ 16 ไซเคิลของสัญญาณนาฬิกา และใช้เวลา 7 ไซเคิลเพื่อเอ็กซีคิวซ์ ดังนั้น ฮับจะติดต่อกับค็อกตัวใดตัวหนึ่งได้ อาจใช้คาบเวลาเพียง 7 ไซเคิล หรือนานถึง 22 ไซเคิล เนื่องจากจะต้องรอให้ฮับวนมาจนครบรอบ

2.4 ภาษาของโปรแกรมที่ใช้พัฒนาโพรเพลเลอร์ Parallax P8X32A

ภาษาที่ใช้เขียน แบ่งออกเป็น 2 ภาษาได้แก่ ภาษาระดับล่าง คือ ภาษาแอสเซมบลี และ ภาษาระดับสูงคือ ภาษาสปิน (ปัจจุบันกำลังมีภาษาซี อีกภาษาหนึ่ง แต่ยังอยู่ในช่วงทดลองอยู่) โดย IDE ที่ใช้ในการพัฒนาภาษาสปิน และภาษาแอสเซมบลีซึ่งสามารถเขียนได้ง่าย ทาง Parallax ได้ให้ใช้ฟรีและหน้าต่างของโปรแกรมมีเอกลักษณ์ไม่เหมือนใคร มีการแบ่งโค๊ดออกเป็นช่วงๆ และมีสีแบ่งแต่ละช่วงเพื่อบอกให้ผู้พัฒนาได้เห็นอย่างเป็นลำดับ ทำให้ง่ายต่อการดูการแก้ไขโค๊ด และเขียนโค๊ดอย่างมีระเบียบสวยงาม



รูปที่ 2.3 Propeller Tool

1. OBJECT VIEW เป็นหน้าต่างแสดงรายเอียดของออบเจ็กต์ที่กำหนดในโปรแกรม

2. Folder List เป็นหน้าต่างเช่นเดียวกับ Explorer ของวินโดวส์ โดยสามารถเลือกให้แสดงเฉพาะไลบรารี่ของซอร์สโค้ด Spin ได้โดยกดที่ปุ่ม ที่มุมซ้ายบนของหน้าต่าง สำหรับที่แถบคอมโบบอกซ์ จะมีลิงก์เพื่อเข้าสู่ไลบรารี่ที่เก็บซอร์สโค้ดตัวอย่างอยู่ 3 ตำแหน่งคือ Propeller Library , Propeller Library-DEMO และ Manual-Example

3. File List เป็นหน้าต่างสำหรับแสดงไฟล์ที่อยู่ในโฟลเดอร์ ซึ่งชี้มาจาก Folder List โดยสามารถกำหนดได้ว่าต้องการให้แสดงไฟล์ทั้งหมด แสดงเฉพาะไฟล์ที่มีนามสกุล .Spin ไฟล์ที่มีนามสกุล .birary หรือ .eeprom หรือไฟล์ที่มีนามสกุล .txt ก็ได้

4. Editor เป็นหน้าต่างสำหรับการเขียนหรือแก้ไขซอร์สโค้ด โดยสามารถเปิดไฟล์ได้พร้อมกันหลายๆ ไฟล์ ไฟล์ที่แอกทีฟอยู่ขณะนั้นจะถูกคอมไพล์ หน้าต่าง editor จะมีความสามารถแยกแยะชุดคำสั่ง ชนิดของบล็อกคำสั่งได้ โดยจะใช้สีเพื่อแยกบล็อกส่วนต่างๆของโปรแกรมออกเป็น 5 ส่วนคือ CON,VAR,OBJ,PUB และ DAT สำหรับชุดคำสั่งหรือชื่อของบล็อกจะแสดงด้วยตัวอักษรเข้ม

นอกจากนี้บริเวณแถบบนของโปรแกรมยังใช้เพื่อกำหนดว่า จะให้แสดงรูปแบบของซอร์ดโค้ดในรูปแบบใด มีรายละเอียดดังนี้

1. Full Source เป็นการแสดงซอร์สโค้ดและคำอธิบาย (comment) รวมถึงส่วนที่เป็นรายละเอียดเอกสาร (document) ทั้งหมดบนหน้าต่าง Editor
2. Condensed เป็นการแสดงเฉพาะส่วนของซอร์สโค้ด และคำอธิบายที่อยู่ด้านหลังเท่านั้น
3. Summary เป็นการรวมเฉพาะชื่อของ CON VAR OBJ PUB และ DAT ที่มีใช้ในโปรแกรมเท่านั้น
4. Documentation แสดงรายระเอียดเอกสารที่ระบุในโปรแกรม ผ่านเครื่องหมาย “{{“,”}}” และ” ’ ‘ ” พร้อมระบุชื่อไฟล์และ PUB ที่ใช้ในโปรแกรม จำนวนหน่วยความจำโปรแกรมและตัวแปรที่ใช้การใช้งานปกติ นิยมกำหนดค่าไว้ที่ Full Source

การเขียนโปรแกรมเพื่อพัฒนาโพรเพลเลอร์นั้น จะใช้โปรแกรมภาษาสปิน ซึ่งมีลักษณะการเขียนโปรแกรมเป็นแบบ OBJECT ประกอบด้วยบล็อกต่างๆ คือ CON, VAR, OBJ, PUB, PRI และ DAT โดยแต่ละบล็อกมีรายละเอียดดังนี้

CON Block นี้จะใช้สำหรับกำหนดค่าคงที่ให้กับตัวแปรที่จะใช้งาน ซึ่งตัวแปรที่ประกาศในบล็อกนี้สามารถเรียกใช้ได้ภายในไฟล์เดียวกันจะแสดงสีของบล็อกด้วยสีเหลือง

VAR เป็นการกำหนดค่าตัวแปรแบบครอบคลุมทั้งโปรแกรม โดยชนิดของตัวแปรที่ใช้กับสปินประกอบด้วย

Byte: กำหนดตัวแปรเป็นตัวเลขจำนวน 8 บิต

Word: กำหนดตัวแปรเป็นตัวเลขจำนวน 16 บิต

Long: กำหนดตัวแปรเป็นตัวเลขจำนวน 32 บิต

Float: กำหนดตัวแปรเป็นตัวเลขทศนิยม

OBJ เป็นบล็อกในส่วนที่ใช้ประกาศไลบรารี่ที่ต้องการติดต่อด้วย โดยกำหนดชื่อที่คอลัมน์แรกตามด้วยเครื่องหมาย : แล้วตามด้วยชื่อไลบรารี่ที่ต้องการเรียกใช้ภายในเครื่องหมายคำพูดจะแสดงสีของบล็อกด้วยสีส้ม

PUB เป็นบล็อกสำหรับการสร้างส่วนของโปรแกรมย่อย หรือฟังก์ชัน แบบเรียกใช้งานจากภายในโปรเจกต์หรือเรียกผ่านทางโปรเจกต์อื่นๆ ก็ได้ PUB ตัวแรกที่พบถือว่าเป็นส่วนที่เรียกใช้เป็นอันดับแรก ถ้าเทียบกับภาษา C ซึ่งหมายถึงส่วนของ MAIN นั่นเองจะแสดงสีของบล็อกด้วยสีน้ำเงิน

PRI เป็นบล็อกสำหรับสร้างส่วนของโปรแกรมย่อย หรือฟังก์ชันแบบเรียกใช้งานเฉพาะภายในตัวโปรเจกต์ที่กำลังทำงานอยู่เท่านั้น การทำงานส่วนอื่นๆจะเหมือนกับการทำงานส่วนของบล็อก PUB จะแสดงสีของบล็อกด้วยสีฟ้า

DAT เป็นบล็อกสำหรับกำหนดเก็บตารางของข้อมูลหรือเขียนโค้ดโปรแกรมภาษาแอสเซมบลีจะแสดงสีของบล็อกด้วยสีเขียว

CON แสดงด้วยสีเหลือง

VAR แสดงด้วยสีส้ม

OBJ แสดงด้วยสีชมพู

PUB แสดงด้วยสีน้ำเงิน

PRI แสดงด้วยสีฟ้า

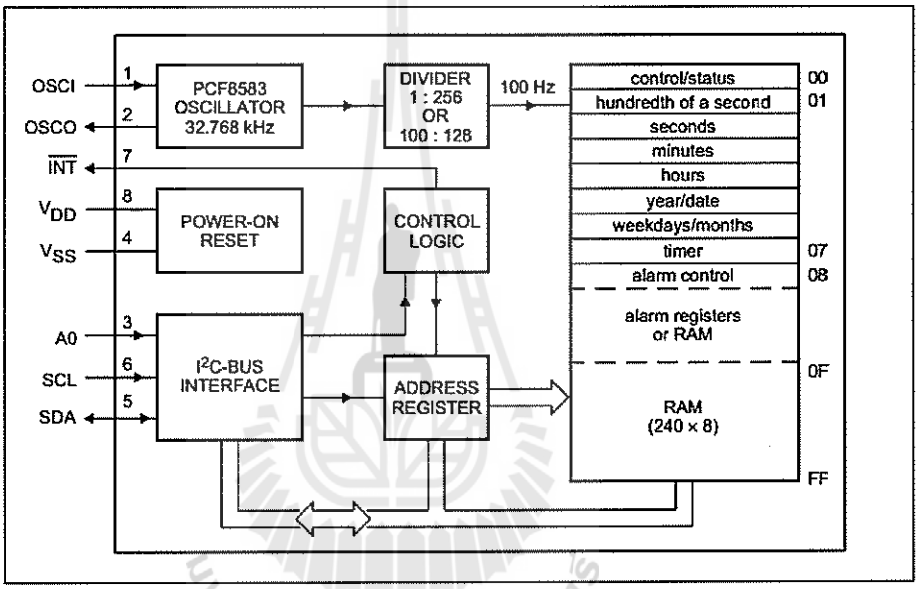
DAT แสดงด้วยสีเขียว

2.5 ตัวนับสัญญาณเวลานาฬิกาจริง (Real Time Clock PCF8583)

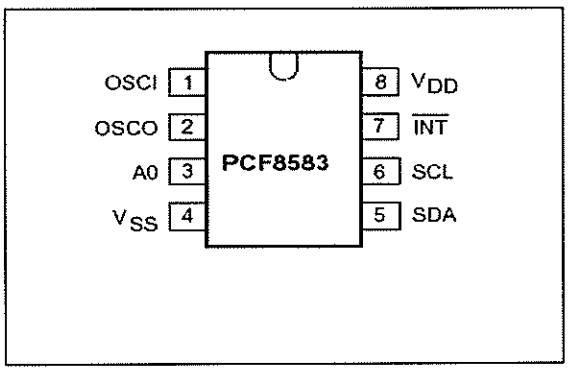
PCF8583 เป็นไอซีตัวนับสัญญาณเวลานาฬิกาจริง Real Time Clock (RTC) ทำหน้าที่เกี่ยวกับเวลา และปฏิทิน ให้กับ ไมโครคอนโทรลเลอร์ โดยต่อใช้งานกับไมโครคอนโทรลเลอร์ แบบอนุกรมตามมาตรฐานของ I2C BUS ระบบฐานเวลา เป็นสิ่งสำคัญที่สามารถนำไปใช้ในอุปกรณ์อีเล็กโทรนิกส์ได้หลากหลาย ภายในไมโครคอนโทรลเลอร์เองก็มีไทเมอร์ เพื่อใช้ในการจับเวลา หรือนำไปใช้เป็นฐานเวลาจริงได้เช่นกัน แต่เนื่องจากไมโครคอนโทรลเลอร์สามารถทำงานได้ต่อเมื่อมีไฟเลี้ยงเท่านั้น ดังนั้นการใช้ ไทเมอร์ของไมโครคอนโทรลเลอร์สร้างฐานเวลาจริงจึงไม่เหมาะสมในบางการใช้งาน PCF8583 เป็น IC ฐานเวลาของ บริษัทฟิลลิปส์ มีบัสรับส่งข้อมูลแบบ I2C ซึ่งเป็นแบบ 2 wire สามารถสื่อสารได้ 2 ทิศทาง (bi-direction bus) ฐานเวลาของ PCF8583 นั้นสามารถเก็บข้อมูล มิลลิวินาที,วินาที, นาที, ชั่วโมง,ได้ระบบเวลาสามารถทำงานโหมดรูปแบบ 24 ชั่วโมง หรือ 12 ชั่วโมง AM/PM ก็ได้ ภายมีระบบตรวจจับแหล่งจ่ายไฟ โดยถ้าแหล่งจ่ายไฟหลักถูกตัดไป PCF8583 สามารถสวิตซ์ไปใช้ไฟจากแบตเตอรี่ และทำงานต่อไป โดยที่ยังสามารถรักษาข้อมูลไว้ได้การต่อใช้งานร่วมกับไมโครคอนโทรลเลอร์ด้วยระบบบัส I2C นั้นสามารถทำได้โดยต่อตัวต้านทาน Pull up ดังแสดงในรูป ในกรณีที่ต้องการต่อร่วมกับอุปกรณ์ Slave หลายตัว ก็สามารถทำได้โดยต่ออุปกรณ์ Slave ขนานกันไป การติดต่อสื่อสารระหว่างอุปกรณ์ Master กับ Slave แต่ละตัวนั้น จะถูกแยกโดย Address ของอุปกรณ์ Slave ซึ่งจะถูกส่งจากอุปกรณ์ Master ไปยังอุปกรณ์ Slave ก่อนเริ่มการรับส่งข้อมูล

- VCC: ใช้ต่อไฟเลี้ยง +5 V
- VSS: ใช้ต่อกราวด์
- A0: ใช้ต่อกับแบตเตอรี่ 3 V เพื่อรักษาการทำงาน ในกรณีที่ไม่มีไฟเลี้ยงจ่าย
- SDA: ขารับส่งข้อมูลด้วยระบบบัส I2C

- SCL: ขาสัญญาณนาฬิกาสำหรับการรับส่งข้อมูลด้วยระบบบัส I2C
- INT: อินเตอร์รับ เอาต์พุต
- OSCI: ออสซิเลเตอร์อินพุต ใช้ต่อกับคริสตอลความถี่มาตรฐาน 32.678 MHz เพื่อสร้างฐานเวลาจริง
- OSCO :ออสซิเลเตอร์ เอาต์พุต



รูปที่ 2.4 Block diagram PCF8583

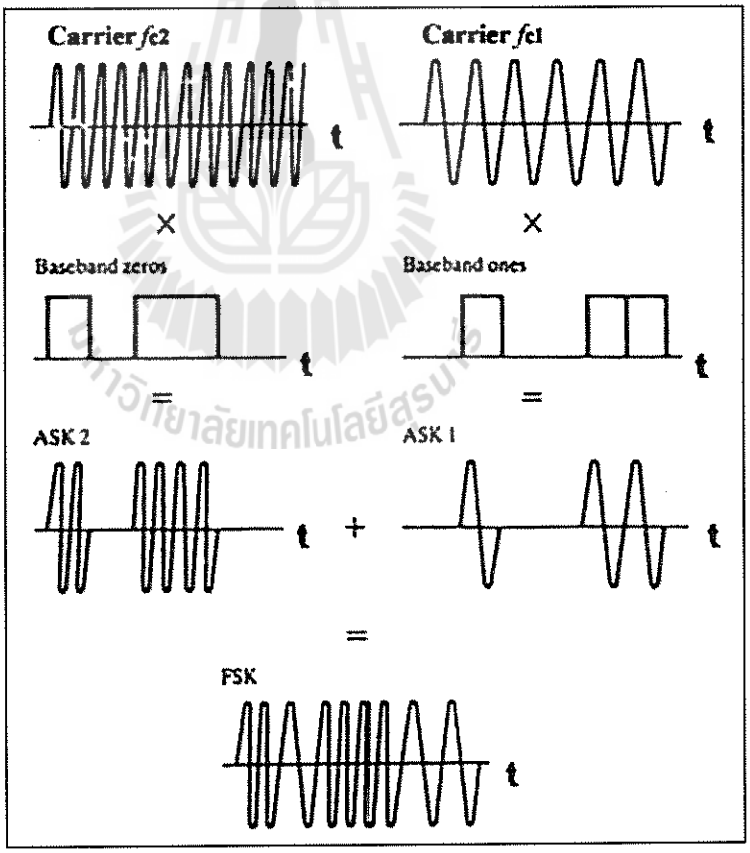


รูปที่ 2.5 แสดงตำแหน่งขาของตัวนับสัญญาณเวลานาฬิกาจริง (Real Time Clock)

2.6 การ Modulation แบบ FSK

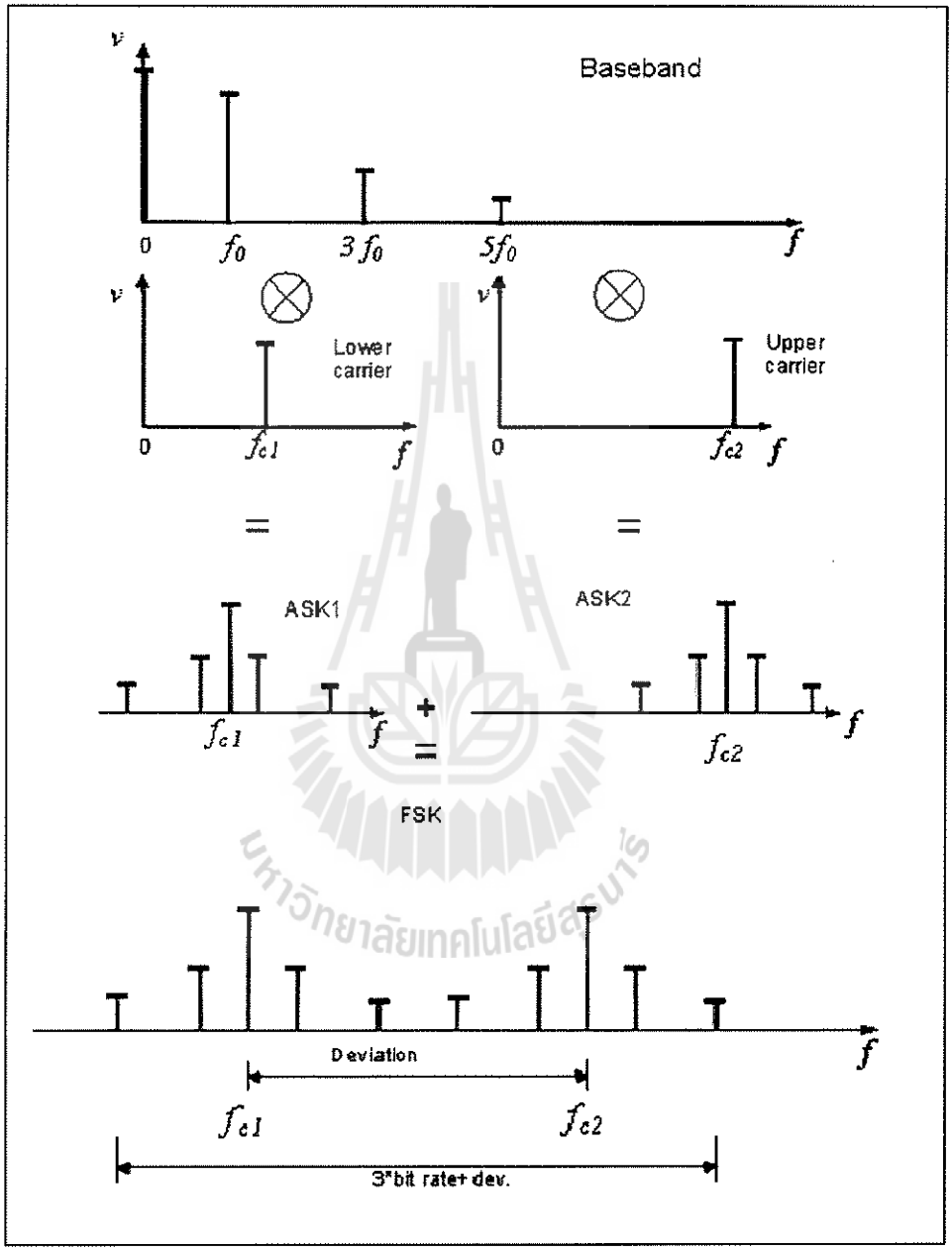
Frequency Shift Keying (FSK) จะให้ความถี่เปลี่ยนแปลงไปตามบิตข้อมูลโดยให้แอมพลิจูดและเฟสคงที่ เช่น ถ้าบิตมีค่า "1" จะให้ความถี่มีค่าสูงกว่าปกติ และบิตมีค่า "0" จะให้ความถี่มีค่าน้อยกว่าปกติ FSK สามารถทนต่อสัญญาณรบกวนได้ดีกว่า ASK เนื่องจากอุปกรณ์ที่รับข้อมูลไม่ต้องสนใจว่า แอมพลิจูดหรือแรงดันไฟฟ้าจะมีค่าเปลี่ยนแปลงไปอย่างไรจะพิจารณาเฉพาะความถี่เท่านั้น แต่ FSK จะมีข้อเสียที่สื่อที่ใช้ในการส่งสัญญาณเท่านั้น จะต้องมีแบนด์วิธที่เพียงพอที่จะสามารถส่งผ่านความถี่ของสัญญาณที่ต้องการได้

FSK สร้างมาจากสัญญาณ ASK สองตัวนำมารวมแบบ interleave โดยที่สัญญาณ ASK ทั้งสองใช้ความถี่คลื่นพาห์คนละความถี่ และสัญญาณเบสแบนด์ของสัญญาณ ASK_1 เป็น complement ของสัญญาณ ASK_2 ดังแสดงในรูป 2.6



รูปที่ 2.6 การสร้างสัญญาณ FSK

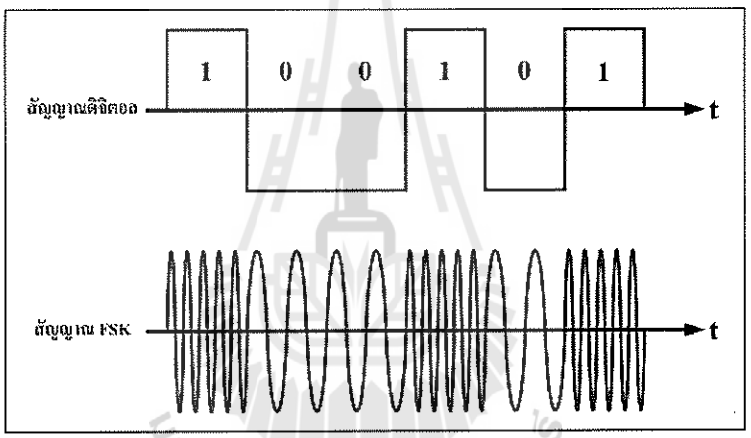
การวิเคราะห์สมการของสัญญาณ FSK ทำได้โดยอาศัยสมการของ ASK สองตัวมาบวกกัน



รูปที่ 2.7 การรวมกันของสัญญาณ ASK สองสัญญาณ

สมการการModulation แบบ FSK สามารถเขียนได้ดังนี้

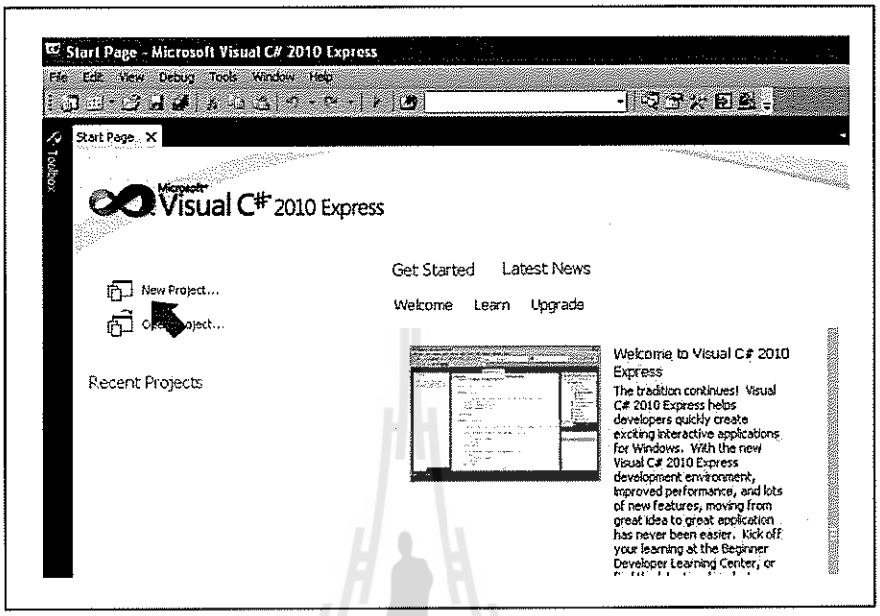
$$\text{FSK (t)} = \begin{cases} (Asin(2\pi fc0t + \phi), \text{data(t)= 0} \\ (Asin(2\pi fc1t + \phi), \text{data(t)= 1} \end{cases}$$



รูปที่ 2.8 แสดงการ มอดูเลชันแบบ FSK

2.7 การเขียนโปรแกรมโดยใช้ VISUAL C# 2010 EXPRESS

ภาษาซีชาร์ป (C# Programming Language) เป็นภาษาโปรแกรมออบเจ็กต์ทำงานบนดอตเน็ทเฟรมเวิร์ก พัฒนาโดยบริษัทไมโครซอฟท์และมี Anders Hejlsberg เป็นหัวหน้าโครงการ โดยมีรากฐานมาจากภาษาซีพลัสพลัสและภาษาอื่นๆ (โดยเฉพาะภาษาเดลไฟและจาวา) โดยปัจจุบันภาษาซีชาร์ปเป็นภาษามาตรฐานรองรับโดย ECMA และ ISOC# คือ ภาษาคอมพิวเตอร์ประเภท object-oriented programming พัฒนาโดย Microsoft โดยมีจุดมุ่งหมายในการรวมความสามารถการคำนวณของ C++ ด้วยการโปรแกรมง่ายกว่าของ Visual Basic โดย C# มีพื้นฐานจาก C++ และเก็บส่วนการทำงานคล้ายกับ Java C# ได้รับการออกแบบให้ทำงานกับ .NET platform ของ Microsoft จุดมุ่งหมายคือ อำนวยความสะดวกในการแลกเปลี่ยนสารสนเทศและบริการผ่านเว็บ และทำให้ผู้พัฒนาสร้างโปรแกรมประยุกต์ในขนาดกะทัดรัด C# ทำให้โปรแกรมง่ายขึ้นผ่านการใช้ Extensible Markup Language (XML) และ Simple Object Access Protocol (SOAP) ซึ่งยอมให้เข้าถึงออบเจ็กต์ของโปรแกรมหรือเมธอด โดยปราศจากความต้องการให้ผู้เขียนโปรแกรมเขียนคำสั่งเพิ่มในแต่ละขั้นตอน เนื่องจากผู้เขียนโปรแกรมสามารถสร้างบนคำสั่งที่มีอยู่ แทนที่การคัดลอกซ้ำ C# ภาษา C# ถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework เป็นการการนำข้อดีของภาษาต่างๆ (เช่นภาษา Delphi , ภาษา C++) มาปรับปรุงเพื่อให้มีความเป็น OOP (โปรแกรมเชิงวัตถุ) มากขึ้น ขณะเดียวกันก็ลดความซับซ้อนในโครงสร้างของภาษาลง (เรียบง่ายกว่าภาษา C++) และมีสิ่งที่เกินความจำเป็นน้อยลง (เมื่อเทียบกับ Java) จุดเด่นหลักๆ ของภาษา C# มีดังนี้ Component oriented เป็นภาษาที่เน้นชิ้นส่วนโดยถูกออกแบบมาเป็นอย่างดีทำให้สามารถนำมาใช้ต่อกันเป็นอะไรก็ได้สิ่งต่าง ๆ ใน C# เป็นออบเจ็กต์ทั้งหมดเป็นภาษา ที่ทนทาน (robust) ทนต่อความผิดพลาด ไม่ทำให้ระบบแฮงค์ หรือระบบทำงานช้า ภาษา C# จัดเตรียมกลไกไว้หลายอย่างที่ช่วยให้ผู้เขียนโปรแกรมสามารถนำโค้ดที่เขียนไว้ในโปรเจคหนึ่งไปใช้กับอีกโปรเจคหนึ่งได้ง่าย นอกจากนั้นภาษา C# ยังสามารถเรียกใช้คลาสหลายพันคลาสใน .NET Framework ได้โดยตรง ทำให้ลดเวลาการพัฒนาซอฟท์แวร์ได้มาก



รูปที่ 2.9 รูปแสดงตัวอย่างโปรแกรม Visual C# 2010 Express

2.8 กล่าวสรุป

ในบทนี้ได้กล่าวถึงทฤษฎีต่างๆ ที่มีความเกี่ยวข้องกับโครงงาน ซึ่งเป็นทฤษฎีที่นำมาใช้ เพื่อการออกแบบและประยุกต์ใช้งานในโครงงานนี้ โดยได้อธิบายถึงที่มาของทฤษฎี และข้อดีของโปรแกรมและอุปกรณ์ที่นำมาใช้กับโครงงานนี้ โดยบอกถึงการใช้งานฟังชั่นต่างๆทฤษฎีที่เกี่ยวข้องเช่น ทฤษฎีไมโครคอนโทรลเลอร์ (Parallax P8X32A) ทฤษฎีภาษาที่ใช้กับไมโครคอนโทรลเลอร์ (Parallax P8X32A) ทฤษฎีตัวนับสัญญาณเวลานาฬิกาจริง (Real Time Clock) ทฤษฎีการมอดูเลตเชิงเลขทางความถี่ (Frequency – Shift Keying: FSK) ทฤษฎีการเขียนโปรแกรมแสดงผล (Visual C# 2010 Express) ซึ่งต้องใช้ข้อมูลเหล่านี้ในการออกแบบโครงงาน ดังได้กล่าวไว้ในบทที่ 3 และหวังว่าจะเป็นประโยชน์สำหรับผู้ที่ต้องการศึกษาต่อไป

บทที่ 3

การออกแบบและพัฒนาระบบ

3.1 กล่าวนำ

ในการออกแบบระบบเครื่องจับเวลาการแข่งขันเรือพายส่งเวลาผ่านวิทยุสื่อสารนั้น โครงงานจะประกอบด้วยตัวนับสัญญาณเวลานาฬิกาจริง บอร์ดไมโครคอนโทรลเลอร์ Parallax P8X32A ปุ่มสวิทซ์สำหรับกด โปรแกรมแสดงผลการจับเวลาและจอ LCD แสดงผล โดยเมื่อกดสวิทซ์ไมโครคอนโทรลเลอร์จะนำเวลาจากตัวนับสัญญาณเวลานาฬิกาจริง มาแปลงเป็นความถี่ตามค่าของเลขฐานสอง โดยกำหนดให้บิตข้อมูลที่เป็นศูนย์อยู่ที่ความถี่ 1200 Hz และบิตข้อมูลที่เป็นหนึ่งอยู่ที่ความถี่ 2200 Hz ซึ่งอยู่ในช่วงความถี่ของเสียง เพื่อส่งสัญญาณความถี่ที่ได้ไปยังภาคส่งของเครื่องส่งวิทยุย่าน VHF และข้อมูลจะแสดงที่จอ LCD ด้วย จากนั้นเครื่องส่งวิทยุทำการส่งสัญญาณไปยังส่วนของภาครับ ซึ่งสัญญาณที่ส่งไปจะอยู่ในรูปของสัญญาณเสียง จากนั้นภาครับซึ่งประกอบไปด้วยเครื่องรับวิทยุย่าน VHF จะทำการรับสัญญาณที่ได้มาส่งต่อไปยังบอร์ดไมโครคอนโทรลเลอร์ แล้วทำการถอดสัญญาณเสียงซึ่งมี 2 ความถี่แปลงออกมาเป็นข้อมูลดิจิตอล โดยให้ความถี่ 1200 Hz มีค่าเป็นบิต 0 และให้ความถี่ 2200 Hz มีค่าเป็นบิต 1 เมื่อข้อมูลดิจิตอลรวมกันจนครบ 8 บิตแล้วจะแปลงเป็นเลขแอสกี้ หลังจากนั้นเอาเลขแอสกี้มาแปลงเป็นเลขฐานสิบทำให้ได้ค่าเวลาออกมา แล้วนำเวลาที่ได้ส่งไปยังชุดแสดงผลซึ่งใช้คอมพิวเตอร์โดยใช้การเขียนโปรแกรม Visual C# 2010 Express เป็นตัวแสดงผล

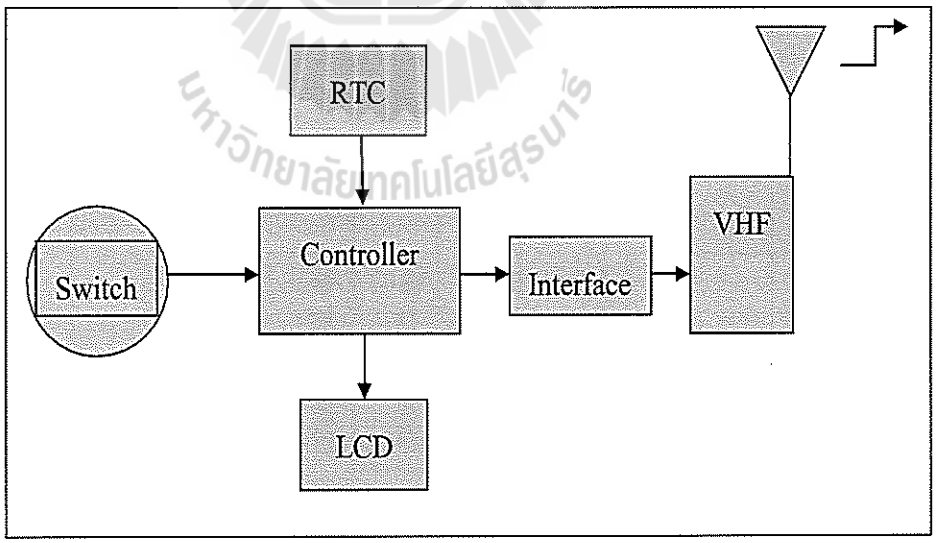
มหาวิทยาลัยเทคโนโลยีสุรนารี

3.2 การออกแบบระบบ Hard Ware

3.2.1 การออกแบบภาคส่ง

ภาคส่งที่จะใช้นั้นต้องทำการส่งข้อมูลซึ่งเป็นค่าเวลา เลข ID และลำดับของสวิทซ์ ที่จะส่งออกจากไมโครคอนโทรลเลอร์ผ่านวิทยุสื่อสารได้ โดยตัวไมโครคอนโทรลเลอร์ Parallax P8x32A จะทำหน้าที่แปลงสัญญาณข้อมูลดิจิตอลเป็นความถี่เสียงออกมาส่งในลักษณะแบบ FSK โดยข้อมูลที่ได้นั้นจะนำมาจากตัวนับสัญญาณเวลานาฬิกาจริง ที่ทำหน้าที่ส่งค่าเวลามาให้ สำหรับในการส่งแต่ล่ะครั้งนั้นต้องมีการกดสวิทซ์ เมื่อทำการกดสวิทซ์แล้วตัวไมโครคอนโทรลเลอร์จะทำการนำเวลา ณ ขณะนั้นที่ตัวนับสัญญาณเวลานาฬิกาจริงส่งมาให้ ส่งออกไปยังจอ LCD เพื่อแสดงค่าเวลาที่กดและทำการส่งออกทางวิทยุสื่อสาร ซึ่งค่าที่ได้จากตัวนับสัญญาณเวลานาฬิกาจริง นั้นเป็นบิตข้อมูลทางเลขดิจิตอลที่เป็นเลขแอสกี้ ข้อดีที่ใช้ระบบการส่งแบบ FSK เนื่องจากการส่ง

สัญญาณนั้นต้องส่งผ่านวิทยุสื่อสารซึ่งต้องใช้ความถี่ที่อยู่ในช่วงความถี่เสียง และตัวไมโครคอนโทรลเลอร์นี้ มีความสามารถในการแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อกในลักษณะแบบ FSK โดยกำหนดความถี่ 1200 Hz เป็นบิต 0 และ ความถี่ 2200 Hz เป็นบิต 1 ซึ่งข้อมูลดิจิตอลที่นำมาแปลงนี้จะเอามาจากเลขแอสกี้ของข้อมูลแล้วแปลงเลขแอสกี้เป็นเลขฐานสองด้วยไมโครคอนโทรลเลอร์ โดยสามารถทำได้ด้วยการเขียนโปรแกรมภาษาสปิน (spin) เพื่อควบคุมการทำงานของคอนโทรลเลอร์ ซึ่งจะทำให้ได้บิตข้อมูลที่เป็น 0 และ 1 ออกมาการส่งแต่ล่ะครั้งจะมีเลข ID และลำดับของสวิทซ์กำกับไปด้วย สำหรับการกำหนดเลข ID นั้นทำได้ด้วยการตั้งค่าที่ดิฟสวิทซ์ (dip switch) เพื่อตั้งค่า ID เป็นเลขในระบบเลขฐานสอง สำหรับเลข ID นี้จะใช้เป็นตัวบอกถึงตำแหน่งต่างๆ ที่ใช้ในการแข่งขันเช่นที่ชุดสตาร์ทกำหนด ID เป็นเลข 00 เพื่อง่ายต่อการระบุตำแหน่งและระยะทาง เมื่อข้อมูลถูกแปลงเป็นความถี่ออกมาแล้วจะถูกส่งไปยังอินเตอร์เฟส เพื่อกรองสัญญาณให้เรียบแล้วส่งไปยังวิทยุสื่อสารเพื่อส่งออกไปยังภาครับ ในการส่งข้อมูลแต่ล่ะครั้งจะส่งเป็นบิตข้อมูลครั้งล่ะ 8 บิต และมีเฮดเดอร์ควบคุมการส่งเพื่อบอกภาครับให้ทราบถึง จุดเริ่มต้นของข้อมูลและจุดสุดท้ายของข้อมูล เพื่อให้ภาครับถอดข้อมูลได้ถูกต้อง



รูปที่ 3.1 การออกแบบระบบภาคส่ง

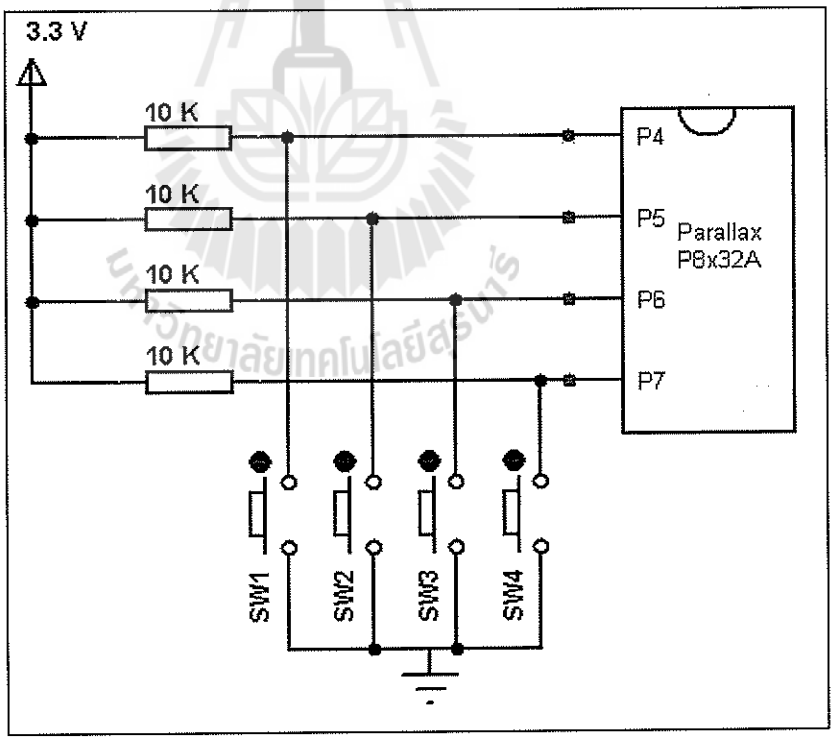
การออกแบบและหลักการทำงานของสวิทซ์ควบคุมการจับเวลาของระบบนั้นประกอบด้วยชุดสวิทซ์ประจำแต่ละบอร์ดบอร์ดละ 4 ตัว เพื่อทำหน้าที่ส่งค่าเวลามายังส่วนของภาครับ โดยสวิทซ์ในสภาวะปกติ (ยังไม่ถูกกด) สวิทซ์จะมีค่าแรงดันตกคร่อมอยู่ที่ 1 V และในสภาวะที่ถูกกดจะมีแรงดันตกคร่อมอยู่ที่ 0 V สวิทซ์จะทำหน้าที่ควบคุมให้ไมคอนโทรลเลอร์ส่งค่าเวลาออกไป เมื่อมีการกดสวิทซ์ในแต่ละครั้ง ซึ่งสวิทซ์แต่ละตัวมีการเชื่อมต่อกับไมโครคอนโทรลเลอร์ดังนี้

Switch 1 เชื่อมต่อกับ P4

Switch 2 เชื่อมต่อกับ P5

Switch 3 เชื่อมต่อกับ P6

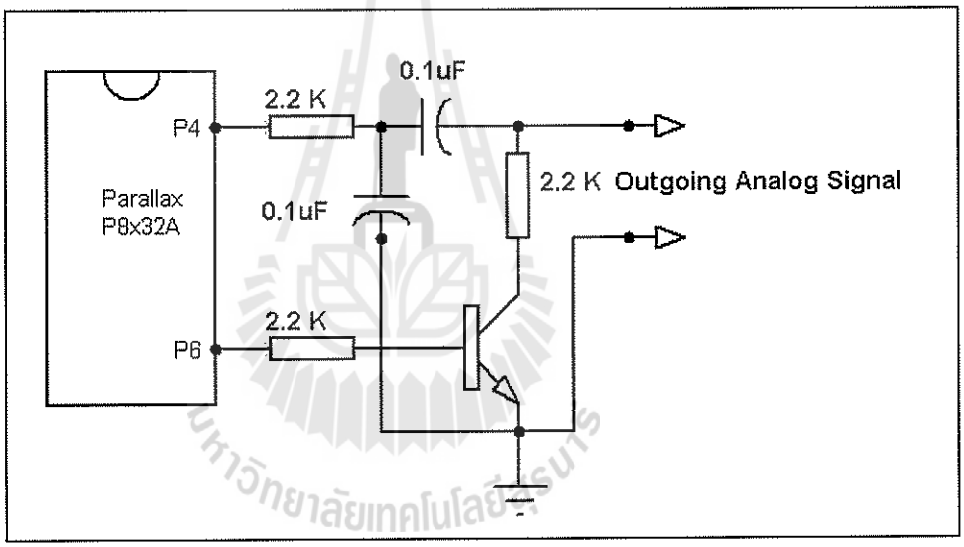
Switch 4 เชื่อมต่อกับ P7



รูปที่ 3.2 สวิทซ์ควบคุมการจับเวลา

การออกแบบอินเตอร์เฟส (interface) ของภาคส่งเพื่อต่อกับวิทยุสื่อสารชุดอินเตอร์เฟสภาคส่ง ทำหน้าที่คล้ายกับฟิลเตอร์ที่จะช่วยกรองสัญญาณที่ออกจากไมโครคอนโทรลเลอร์ก่อนเข้าสู่เครื่องส่งวิทยุ VHF เพื่อให้สัญญาณที่ส่งออกไปมีแรงดันไม่เกิน 3.3 Vp-p

Propeller Duty Mode Output	เชื่อมต่อกับ P22
Propeller PTT Pin	เชื่อมต่อกับ P23
Outgoing Analog Signal	เชื่อมต่อกับ สายไมค์วิทยุและกราวด์

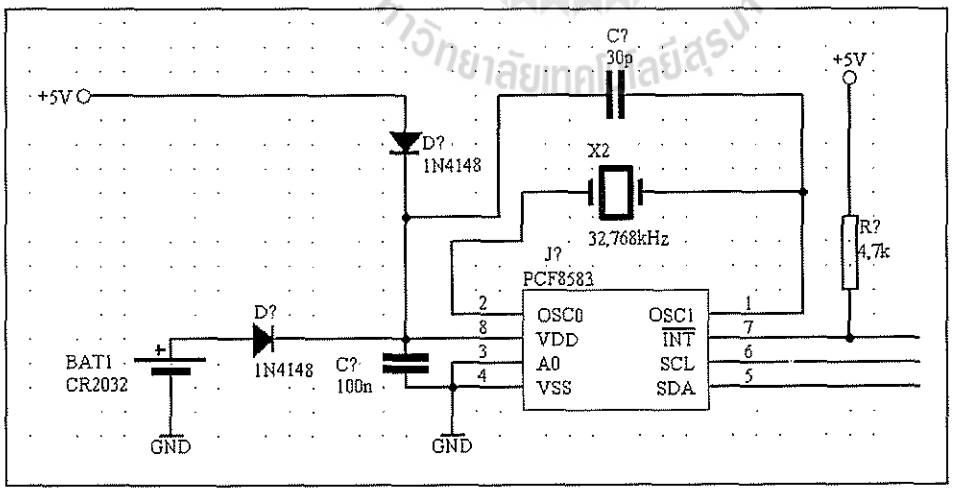


รูปที่ 3.3 อินเตอร์เฟสภาคส่ง

การออกแบบและหลักการทำงานของตัวนับสัญญาณเวลานาฬิกาจริง (real time clock PCF8583) เชื่อมเข้ากับไมโครคอนโทรลเลอร์ Parallax P8X32A ตัวนับสัญญาณเวลานาฬิกาจริงของบริษัทฟิลลิปส์ มีบัสรับส่งข้อมูลแบบ I2C ซึ่งเป็นแบบ 2 wire สามารถสื่อสารได้ 2 ทิศทาง (bi-direction bus) ฐานเวลาของตัวนับสัญญาณเวลานาฬิกาจริง นั้นสามารถเก็บข้อมูล มิลลิวินาที, วินาที, นาที, ชั่วโมง, ได้ ระบบเวลาสามารถทำงานโหมดรูปแบบ 24 ชั่วโมง หรือ 12 ชั่วโมง

AM/PM ก็ได้ ภายในมีระบบตรวจจับแหล่งจ่ายไฟ โดยถ้าแหล่งจ่ายไฟหลักถูกตัดไปตัวนับสัญญาณเวลานาฬิกาจริง สามารถสวิตซ์ไปใช้ไฟจากแบตเตอรี่ และทำงานต่อไปโดยที่ยังสามารถรักษาข้อมูลไว้ได้ โครงสร้างมีขาทั้งหมด 8 ขาดังแสดงในรูปที่ 3.4 และมีรายละเอียดการทำงานของขาต่าง ๆ ดังนี้ ตัวนับสัญญาณเวลานาฬิกาจริง จะทำหน้าที่ส่งค่าเวลามาให้คอนโทรลเลอร์ตลอดเวลา ซึ่งค่าเวลาที่ได้จะเป็นเลขแอสกี้หลังจากนั้นไมโครคอนโทรลเลอร์จะทำการแปลงเป็นความถี่จำนวน 8 บิต แล้วส่งออกไปยังภาครับเมื่อมีการกดสวิทซ์ โดยมีการเชื่อมต่อกับไมโครคอนโทรลเลอร์ดังนี้

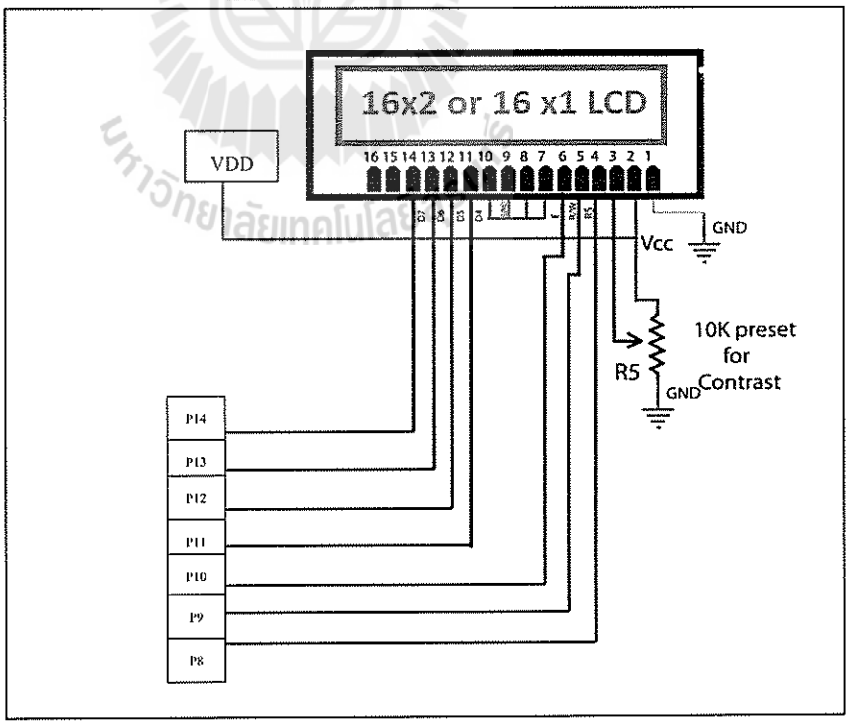
VCC	เชื่อมต่อกับ	ไฟ 5 V
OSCI, OSCO	เชื่อมต่อกับ	คริสตัล
A0	เชื่อมต่อกับ	แบตเตอรี่ 3 V
SDA	เชื่อมต่อกับ	P25
SCL	เชื่อมต่อกับ	P24
VSS	เชื่อมต่อกับ	GND



รูปที่ 3.4 การเชื่อมต่อตัวนับสัญญาณเวลานาฬิกาจริง (real time clock)

การเชื่อมต่อจอ LCD เพื่อจะทำหน้าที่แสดงผลข้อมูลเวลาที่กำลังวิ่งอยู่จากตัวนับสัญญาณเวลานาฬิกาจริง และแสดงข้อมูลเวลาที่ได้จากการกดสวิทซ์ โดยจะแสดงผลในรูปของเลขแอสกี้ที่หน้าจอ โดย LCD จะมีรูปแบบการแสดงผลที่บอกถึง เลน เวลาที่ใช้ในการแข่งขันของเรือ และแสดงเวลาจากตัวนับสัญญาณเวลานาฬิกาจริงซึ่งจะวิ่งไปเรื่อยๆ เพื่อบอกถึงเวลาจริงที่ใช้นับอยู่ โดยข้อมูลจะถูกส่งมาจากคอนโทรลเลอร์แบบ 4 บิต เพื่อนำมาแสดงผล ซึ่งมีการเชื่อมต่อ LCD 16x2 กับ ไมโครคอนโทรลเลอร์ดังนี้

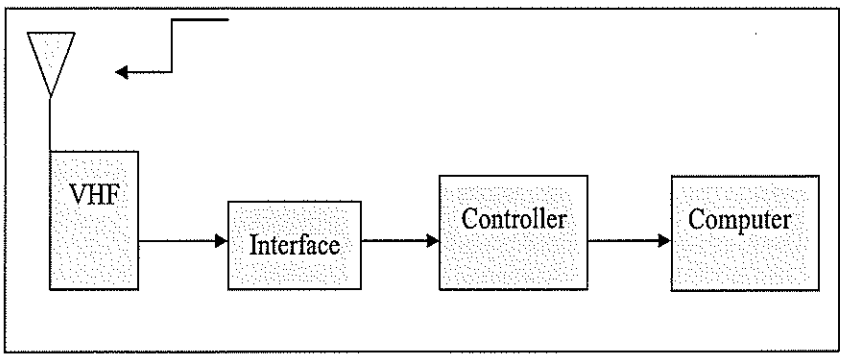
P8	เชื่อมต่อกับ	RS
P9	เชื่อมต่อกับ	R/W
P10	เชื่อมต่อกับ	E
P11	เชื่อมต่อกับ	D4
P12	เชื่อมต่อกับ	D5
P13	เชื่อมต่อกับ	D6
P14	เชื่อมต่อกับ	D7
G	เชื่อมต่อกับ	GND
VDD	เชื่อมต่อกับ	DC+5 V
VEE	เชื่อมต่อกับ	GND



รูปที่ 3.5 จอแสดงผล (LCD)

3.2.2 การออกแบบภาครับ

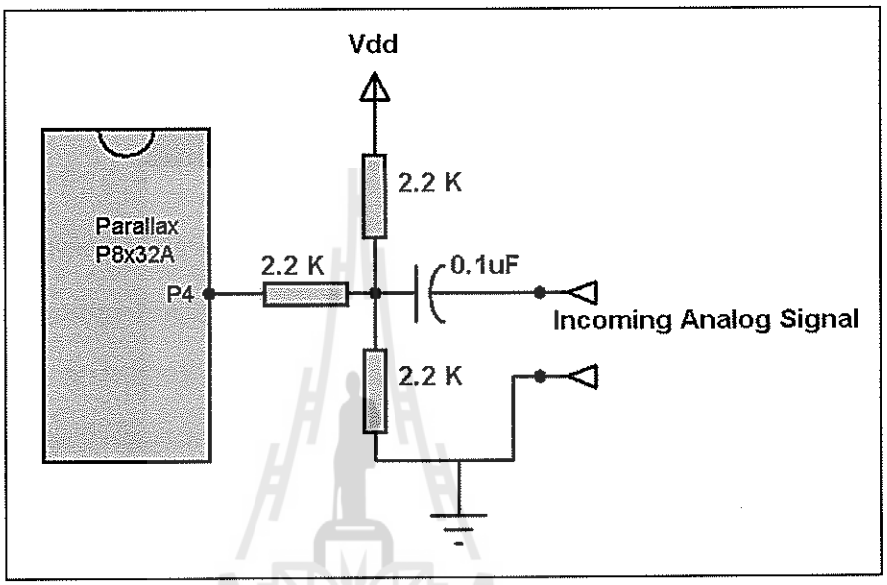
ในการออกแบบภาครับต้องมีความสามารถในการรับข้อมูลที่เป็นความถี่ 1200 Hz และ 2200 Hz แล้วทำการแปลงความถี่นั้นออกมาเป็นข้อมูลดิจิตอลเพื่อแสดงผลได้ โดยมีการทำงานของภาครับดังนี้ เมื่อวิทยุสื่อสารภาครับรับสัญญาณจากภาคส่งมา วิทยุสื่อสารจะส่งสัญญาณออกมาผ่านสายไมค์ของวิทยุสื่อสารที่ต่อกับอินเตอร์เฟสภาครับ ซึ่งเชื่อมกับไมโครคอนโทรลเลอร์ สัญญาณจะอยู่ในรูปของความถี่เสียงโดยมีบิตข้อมูลที่เป็นความถี่ 1200 Hz และ 2200 Hz อยู่และเมื่อส่งมาครบ 8 บิต ตัวคอนโทรลเลอร์จะทำการแปลงความถี่ 1200 Hz เป็นบิต 0 และแปลงความถี่ 2200 Hz เป็นบิต 1 เมื่อทำการแปลงเสร็จครบ 8 บิต จะทำให้ได้ข้อมูลเป็นเลขฐานสองออกมา แล้วแปลงเลขฐานสอง 8 บิต เป็นเลขแอสกี้ โดยการแปลงข้อมูลนี้จะทำได้ด้วยการเขียนโปรแกรมภาษาสปิน เพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์ที่ภาครับ ซึ่งจะทำให้ได้ข้อมูลออกมาเป็นเวลาจริงที่ภาคส่งส่งมา และการแสดงผลข้อมูลนั้นสามารถทำได้ด้วย การเขียนโปรแกรมแสดงผลที่คอมพิวเตอร์ด้วย โปรแกรม Visual C# 2010 สำหรับการทำงานของโปรแกรมเมื่อข้อมูลถูกส่งมาจากไมโครคอนโทรลเลอร์ ตัวโปรแกรมจะนำข้อมูลที่ได้นั้นมาแสดงและทำการเปรียบเทียบข้อมูลว่าอยู่ในเลข ID อะไร เป็นข้อมูลของสวิทซ์ลำดับที่เท่าไหร่ เวลากี่นาฬิกา แล้วทำการเปรียบเทียบโดยจะนำค่าเวลาที่จุดเริ่มต้นส่งมาลบกับเวลาที่จุดต่างๆที่เรือผ่านมา เช่น การแข่งขันที่ระยะทาง 1000 เมตร จะเป็นการนำเวลาที่จุดสตาร์สส่งมาลบกับระยะเวลาที่เรือผ่านจุด 500 เมตร และเวลาที่เรือเข้าเส้นชัย ซึ่งจะทำให้ได้ค่าเวลาจริงที่เรือใช้ในการแข่งขันรอบนั้นออกมา โดยตัวนับสัญญาณเวลานาฬิกาจริง ของแต่ละอุปกรณ์นั้นจะมีฐานเวลาที่เท่ากันทำให้สามารถนำเวลามาลบกันได้ เพื่อให้ได้เวลาที่ใช้ในการแข่งขันออกมา เมื่อได้เวลาแล้วนำค่าเวลาที่ได้ไปเก็บเป็นฐานข้อมูลในรูปของไฟล์ PDF เพื่อเป็นหลักฐานในการแข่งขัน



รูปที่ 3.6 การออกแบระบบภาครับ

การออกแบบอินเตอร์เฟสภาครับเพื่อต่อกับวิทยุสื่อสาร โดยมีการเชื่อต่อกับไมโครคอนโทรลเลอร์ ดังนี้

Propeller Input Pin	เชื่อมต่อกับ	P17
Incoming Analog Signal	เชื่อมต่อกับ	สายไมค์วิทยุและกราวด์



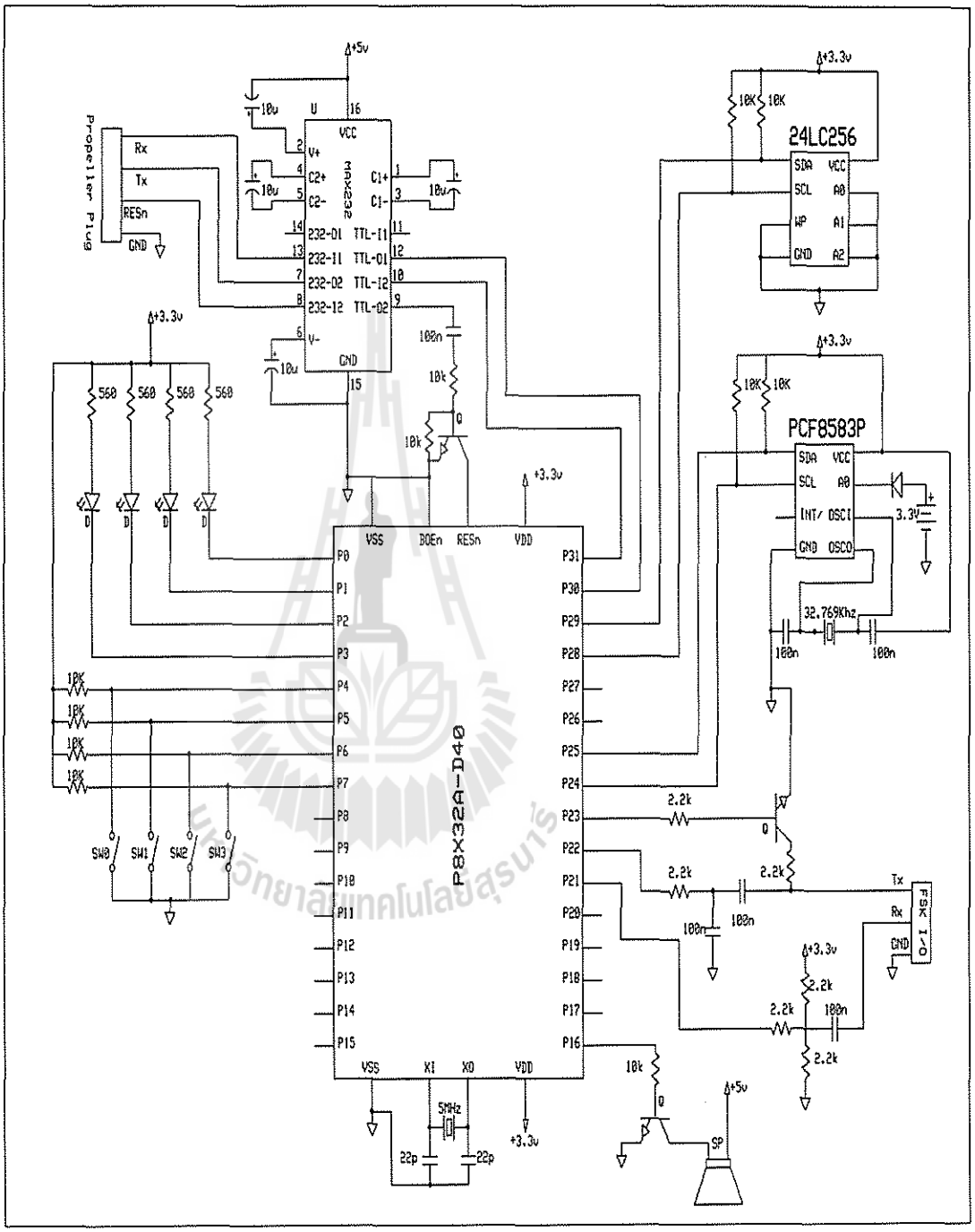
รูปที่ 3.7 อินเตอร์เฟสภาครับ

มหาวิทยาลัยเทคโนโลยีสุรนารี

การออกแบบวงจรอิเล็กทรอนิกส์ของระบบจับเวลา

ในส่วนของวงจรอิเล็กทรอนิกส์นี้ เป็นการรวมวงจรดังที่ได้กล่าวมาแล้วข้างต้น ซึ่งเมื่อนำมารวมกันแล้วจะทำให้ได้วงจรที่สามารถใช้งานได้ตามที่ต้องการ โดยมีการทำงานดังนี้ สำหรับตัวนับสัญญาณเวลานาฬิกาจริงเบอร์ PCF8583 จะทำหน้าที่ส่งค่าเวลามาให้ไมโครคอนโทรลเลอร์ Parallax P8X32A แบบเรียลไทม์ และในไมโครคอนโทรลเลอร์ Parallax P8X32A นี้ได้มีการเขียนโปรแกรมให้ทำงานแยกกันในแต่ละค็อก โดยค็อก แรกให้มีหน้าที่อ่านค่าจากสวิทซ์เมื่อสวิทซ์ถูกกด ซึ่งค็อกนี้จะมีหน้าที่ไปรับค่าเวลาที่ถูกส่งออกมาจากขา SCL และขา SDA ของตัวนับสัญญาณเวลานาฬิกาจริงเบอร์ PCF8583 โดยขา SCL จะต่อกับ P25 และขา SDA ต่อกับ P24 ของไมโครคอนโทรลเลอร์ Parallax P8X32A ข้อมูลที่ถูกส่งมาจะเป็นเลขแอสกี้ หลังจากนั้นส่งข้อมูลต่อมายังค็อกที่ 2 ซึ่งในค็อกที่ 2 จะทำหน้าที่นำข้อมูลแอสกี้ที่ได้มาแปลงเป็นข้อมูลในเลขฐานสองจำนวน 8 บิต และนำข้อมูลเลขฐานสองที่ได้มาทำการแปลงให้เป็นความถี่ในช่วงของความถี่เสียง โดยข้อมูลที่เป็นบิต 0 จะถูกแปลงให้เป็นความถี่ 1200 Hz และข้อมูลที่เป็นบิต 1 จะถูกแปลงเป็นความถี่ 2200 Hz หลังจากนั้นไมโครคอนโทรลเลอร์ Parallax P8X32A จะส่งความถี่ที่ได้ออกไปยัง P22 ที่เชื่อมกับขา Propeller duty mode output ของชุดอินเตอร์เฟสเพื่อกรองความถี่และจะถูกส่งออกไป เมื่อต่อกับสาย Audio in ของเครื่องส่งวิทยุ Icom-IC V80 เครื่องส่งวิทยุถูกตั้งช่องสัญญาณและติดตั้งสายอากาศตามระยะทางการใช้งาน โดยฝั่งภาครับเครื่องรับวิทยุจะถูกต่อการใช้งานเหมือนกับภาคส่ง จากนั้นต่อสัญญาณจากช่อง Audio out ของเครื่องรับวิทยุมาต่อกับ Incoming analog signal ของชุดอินเตอร์เฟส ทำให้ได้สัญญาณออกที่ขา Propeller Input Pin ต่อกับ P17 ของไมโครคอนโทรลเลอร์ Parallax P8X32A จากนั้นไมโครคอนโทรลเลอร์ Parallax P8X32A นำความถี่ที่ได้มาแปลงกลับ เป็นข้อมูลและส่งมายังขา P30 ของไมโครคอนโทรลเลอร์ Parallax P8X32A เพื่อส่งข้อมูลออกไปยัง Serial Port จากนั้นเชื่อมต่อสาย Serial Port เข้าคอมพิวเตอร์ เพื่อนำข้อมูลมาแสดงที่โปรแกรมที่ถูกเขียนขึ้นจาก โปรแกรม Visual C# 2010 Express

วงจรอิเล็กทรอนิกส์ (Schematic Diagram)

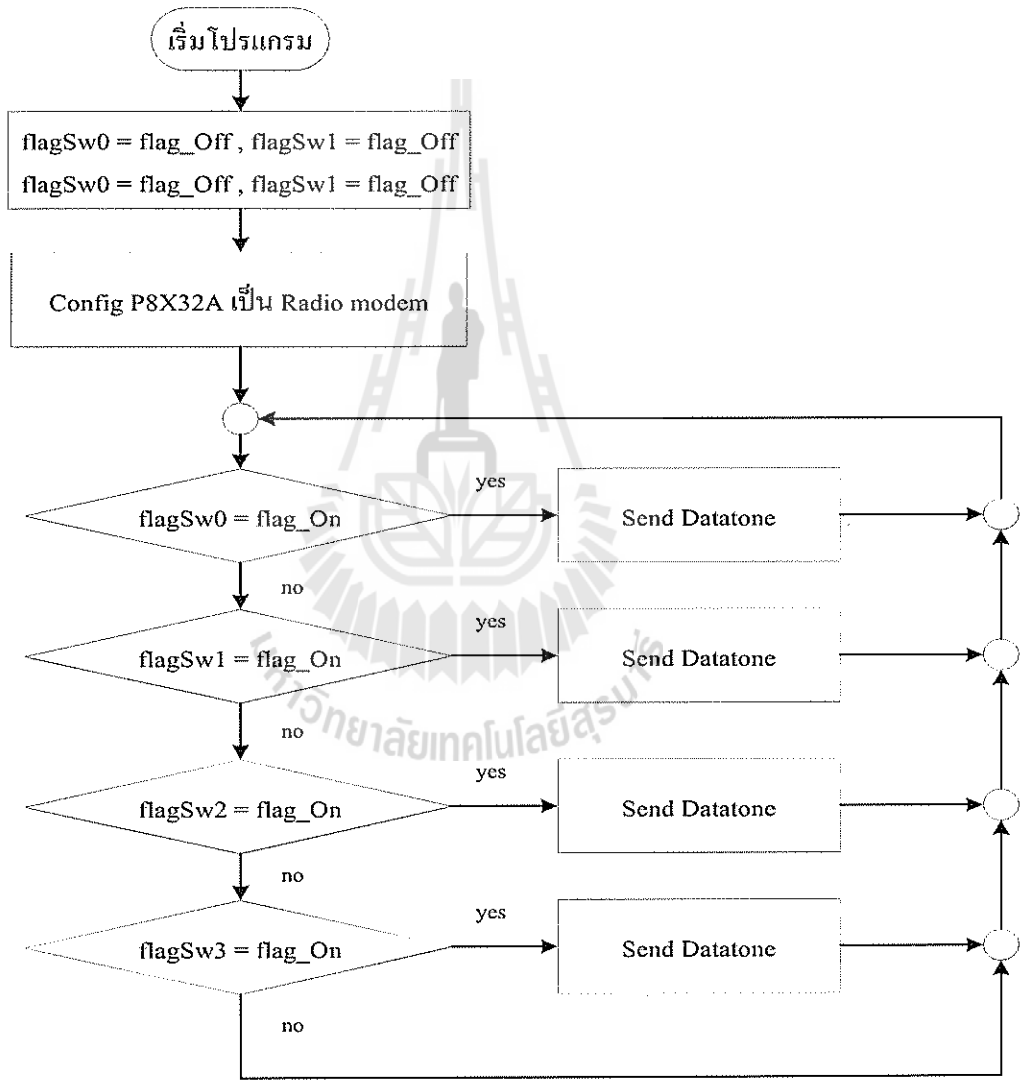


รูปที่ 3.8 วงจรอิเล็กทรอนิกส์ (schematic diagram) ของระบบ

3.3 การพัฒนา Soft Ware

3.3.1 การพัฒนาซอฟแวร์ของไมโครคอนโทรลเลอร์ Parallax P8x32A ด้วยภาษาสปิน

แผนผังการทำงานของโปรแกรมส่วนส่งโทนข้อมูล

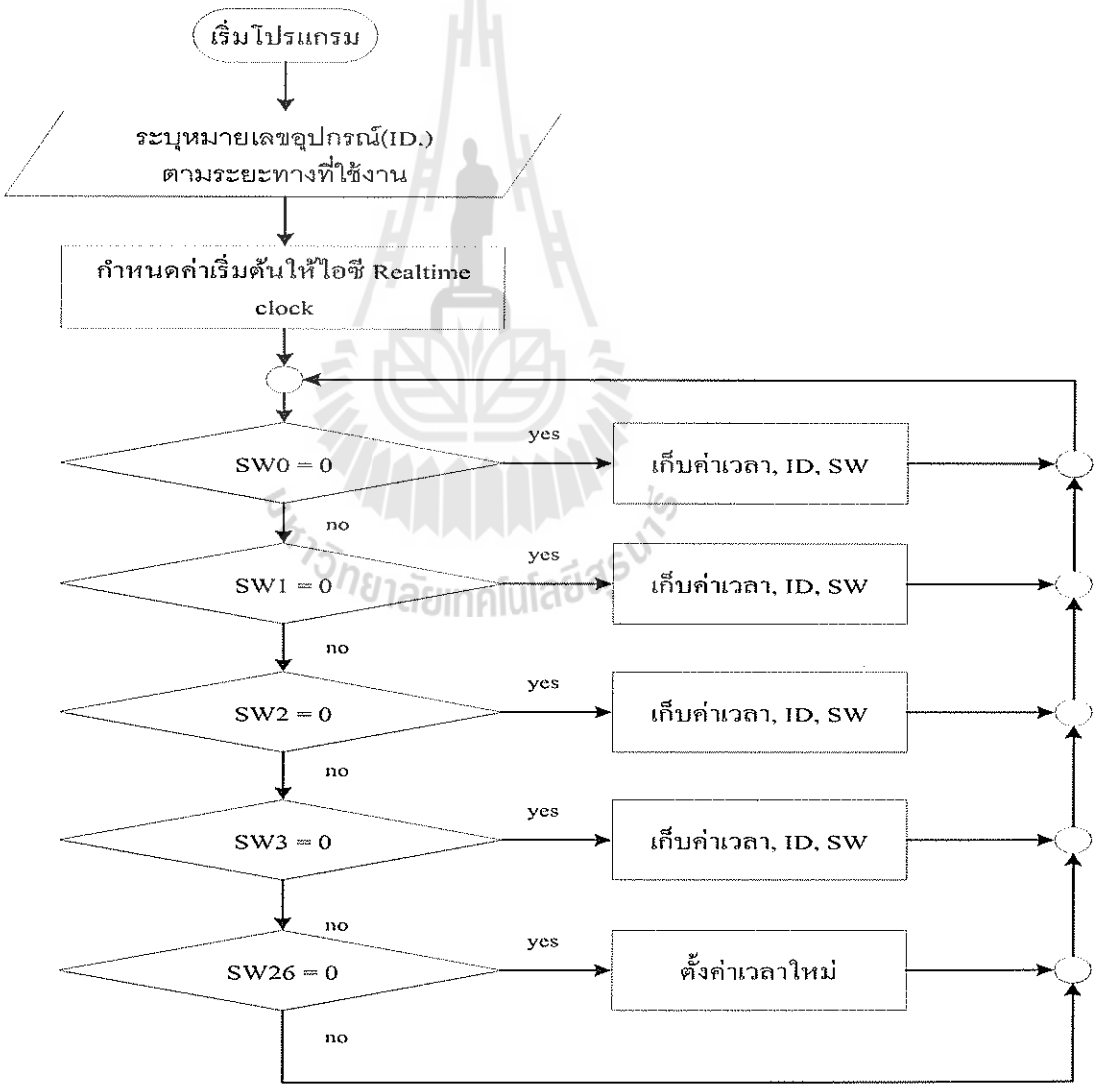


รูปที่ 3.9 แผนผังการทำงานของโปรแกรมส่วนส่งโทนข้อมูล

เริ่มต้นการทำงานด้วยการกำหนดให้ ตัวแปร flagSW0, flagSW1, flagSW2, flagSW3 มีสถานะเท่ากับ flag Off ซึ่งเป็นตัวแปรที่ใช้แทนสถานะ Low (0) เสร็จแล้วทำการกำหนดค่าเริ่มต้น

ให้ไมโครคอนโทรลเลอร์ทำงานเป็น Radio Modem จากนั้นโปรแกรมจะทำการวนลูปเพื่อทำการเช็คสถานะของตัวแปร flagSW0, flagSW1, flagSW2, flagSW3 ว่าตัวแปรใดมีค่าเท่ากับ flag_On ซึ่งเป็นตัวแปรที่แทนสถานะ High (1) หลังจากตรวจพบสอบว่าตัวแปรใดมีสถานะเท่ากับ flag_On แล้วจะทำการส่งโทนข้อมูลที่ได้มา จากการเก็บค่าไว้ที่ส่วนของการอ่านค่าสวิทช์ แล้วทำการส่งโทนข้อมูลออกไปทั้งหมดสามครั้งเพื่อป้องกันการส่งโทนข้อมูลผิดพลาด เมื่อส่งโทนข้อมูลครบแล้วจะทำการเปลี่ยนสถานะจาก flag_On กลับมาเป็น flag_Off อีกครั้งหนึ่ง

แผนผังการทำงานของโปรแกรมส่วนอ่านค่าเวลา

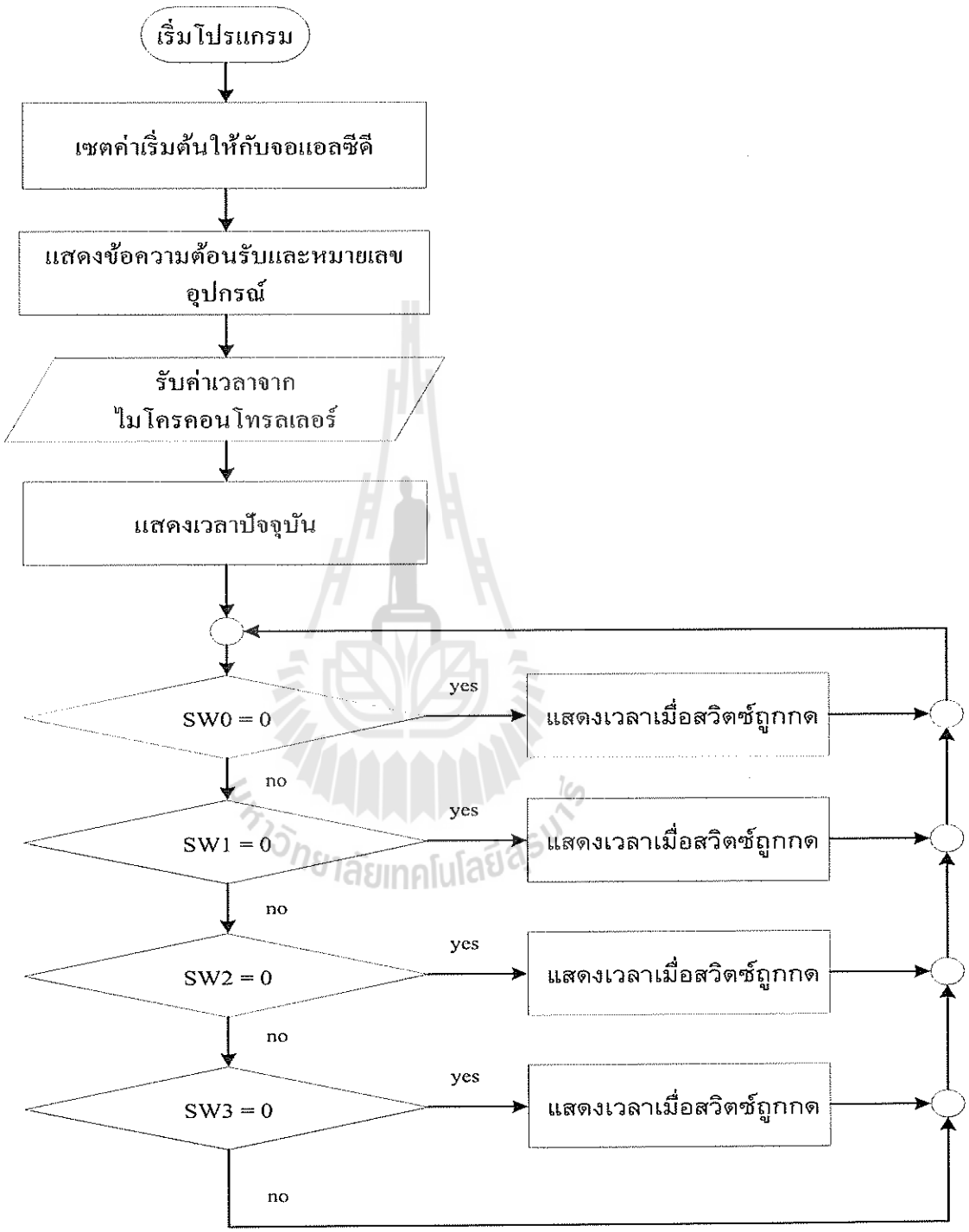


รูปที่ 3.10 แผนผังการทำงานของโปรแกรมส่วนอ่านค่าเวลา

เริ่มต้นการทำงานจะรับค่าจากดิฟสวิทซ์ ที่ใช้กำหนดจุดที่จะทำการจับเวลา ซึ่งจะถูกกำหนดเป็นค่าตัวแปรหมายเลขประจำตัวอุปกรณ์ (ID) เสร็จแล้วจะทำการกำหนดค่าเริ่มต้นให้กับตัวนับสัญญาณเวลานาฬิกาจริง จากนั้นโปรแกรมจะทำการวนลูปเพื่อทำการตรวจสอบการกดสวิทซ์ ในสภาวะปกติ SW0, SW1, SW2, SW3 จะมีสถานะ High (1) หากสวิทซ์ใดถูกกดสถานะจะเปลี่ยนเป็น Low (0) เมื่อตรวจสอบพบว่าสวิทซ์ใดถูกกดไมโครคอนโทรลเลอร์จะทำการเก็บค่าตัวแปรจำนวน 3 ตัว คือ เวลาที่อ่านได้จากตัวนับสัญญาณเวลานาฬิกาจริงขณะที่สวิทซ์ถูกกด หมายเลขประจำตัวอุปกรณ์ (ID) และสวิทซ์ตัวที่ถูกกด และจะทำการกำหนดให้ตัวแปร flag SW ของสวิทซ์ตัวที่กดมีสถานะเป็น flag on เพื่อให้ส่วนของการส่งโทนใช้เป็นเงื่อนไขในการส่งโทนข้อมูลต่อไป

มหาวิทยาลัยเทคโนโลยีสุรนารี

แผนผังการทำงานของโปรแกรมส่วนแสดงผลผ่านจอ LCD



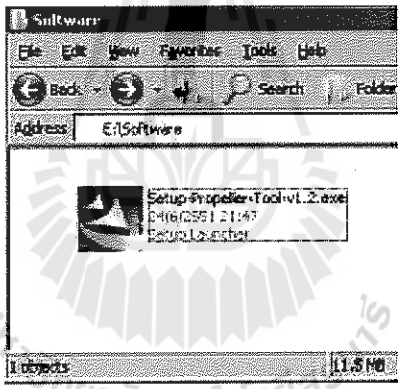
รูปที่ 3.11 แผนผังการทำงานของโปรแกรมส่วนแสดงผลผ่านจอ LCD

เมื่อเริ่มต้นการทำงานจะทำการกำหนดค่าเริ่มต้นให้กับจอ LCD โดยจะแสดงข้อความต้อนรับและหมายเลขประจำอุปกรณ์ จากนั้นรับค่าเวลาจากไมโครคอนโทรลเลอร์มาแสดงเวลาปัจจุบัน โดยโปรแกรมจะทำการวนลูปเพื่อทำการตรวจสอบการกดสวิทซ์ หากสวิทซ์ใดมีสถานะเท่ากับ Low (0) จอ LCD จะทำการแสดงเวลา ณ เวลาที่สวิทซ์ถูกกด

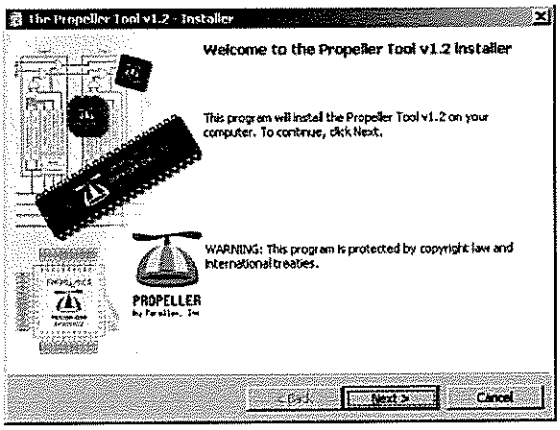
3.3.2 การติดตั้งโปรแกรม Propeller Tools เพื่อเขียนโปรแกรมภาษาสปิน

ซอฟต์แวร์ Propeller Tools สามารถโหลดได้ฟรีที่ www.parallax.com โดยคอมพิวเตอร์ที่จะใช้งานได้ต้องติดตั้งระบบปฏิบัติการวินโดวส์ xpsp2 เป็นอย่างน้อยและมีพอร์ต USB ว่างอย่างน้อย 1 พอร์ต โดยมีการติดตั้งดังนี้

1. ดับเบิลคลิกที่ไฟล์ชื่อ Setup-Propeller-Tool-V1.2.exe

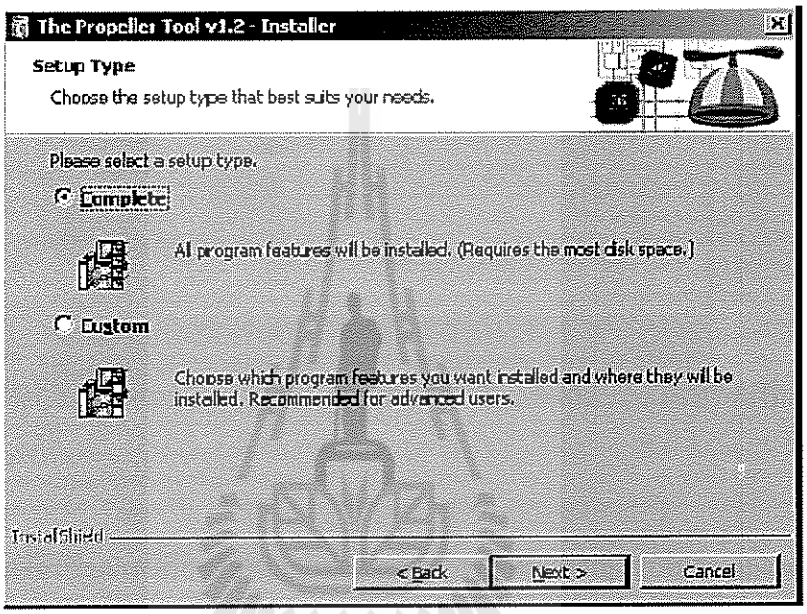


2.เข้าสู่หน้าต่างการติดตั้ง ให้กดปุ่ม Next เพื่อผ่านขั้นตอนนี้

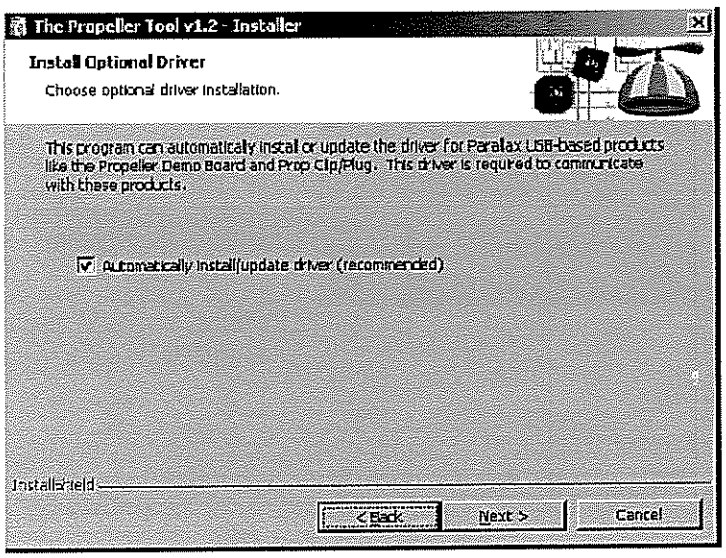


3.โปรแกรมจะแจ้งให้ป้อนชื่อผู้ใช้ชื่อองค์กรหรือบริษัท หลังจากป้อนชื่อเสร็จเรียบร้อยกดปุ่ม Next สู่ขั้นตอนต่อไป

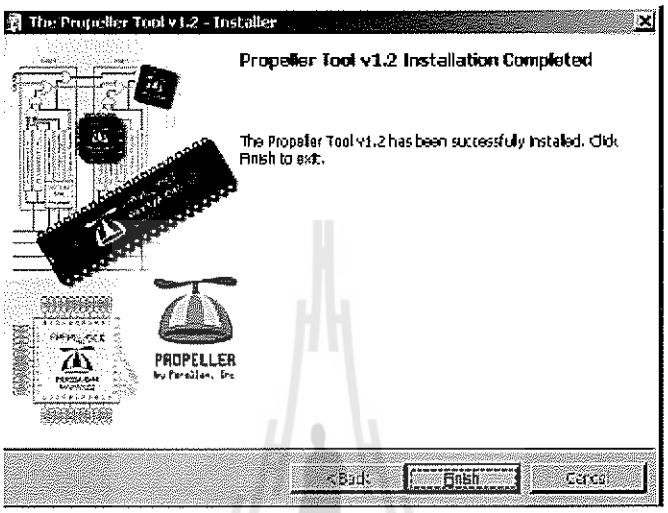
4.หน้าต่างต่อไปเป็นการเลือกรูปแบบการติดตั้งซอฟต์แวร์ ให้ติดตั้งทั้งหมด Complete หรือเลือกติดตั้ง Custom ให้เลือกการติดตั้งแบบสมบูรณ์ จากนั้นกดปุ่ม Next สู่ขั้นตอนต่อไป



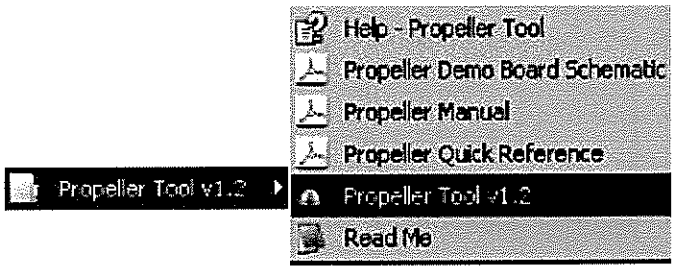
5.ลำดับต่อมาเป็นการเลือกติดตั้งไดรเวอร์ ตัวแปลง USB เป็นอนุกรม ซึ่งแนะนำให้ติดตั้งจากนั้นกดปุ่ม Next สู่ขั้นตอนต่อไป



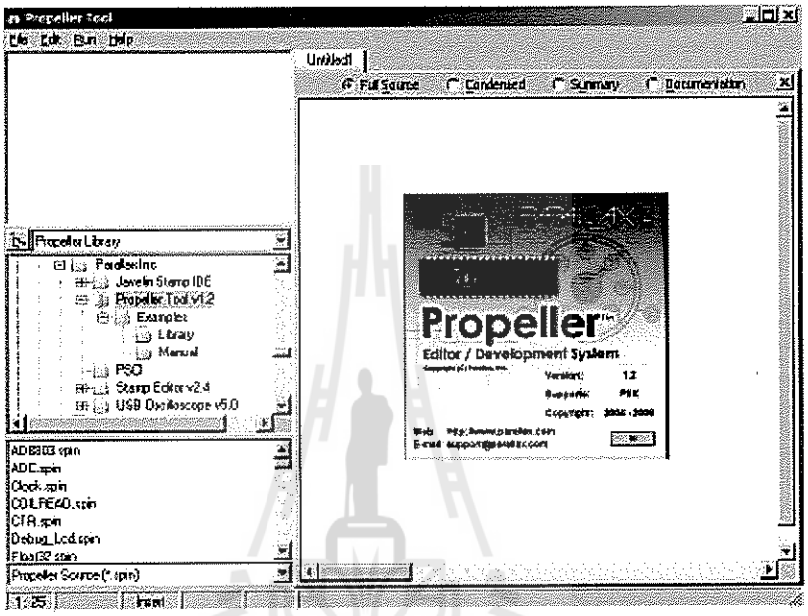
6.โปรแกรมจะใช้เวลาสักครู่ในการติดตั้ง จากนั้นจะแสดงข้อความสิ้นสุดการติดตั้งโปรแกรมให้กด Finish เพื่อจบขั้นตอนการติดตั้ง



7.หลังจากติดตั้งโปรแกรมเรียบร้อย สามารถเปิดโปรแกรมได้จากเมนู START>All Program> Parallax Inc>Propeller Tool V1.2 หรือเรียกที่ ชอร์ตคัต Propeller Tool V1.2 ที่หน้าต่าง Desktop ก็ได้



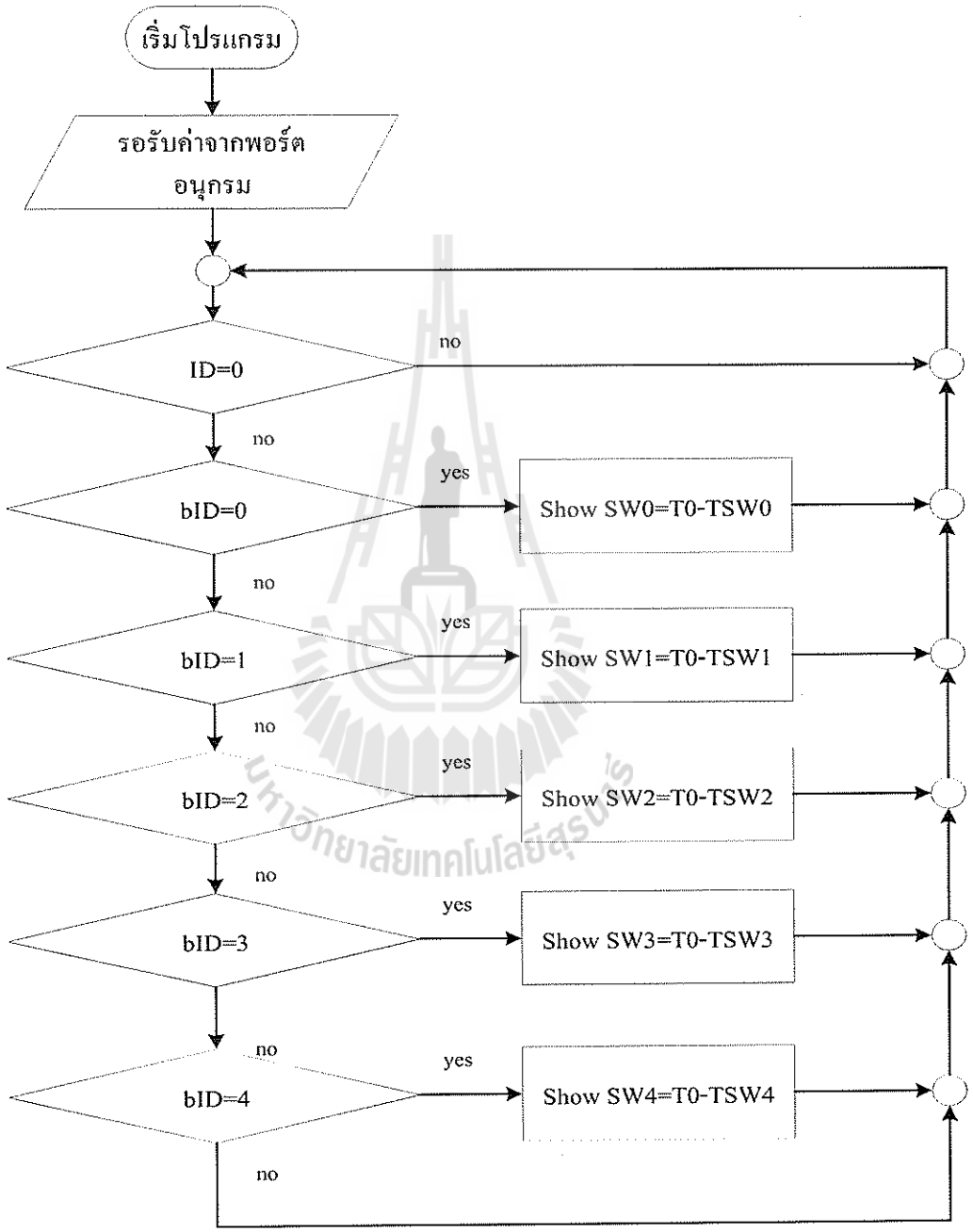
8.จากนั้นหน้าต่างของโปรแกรม Propeller Tool จะปรากฏขึ้นมาให้ใช้งาน



มหาวิทยาลัยเทคโนโลยีสุรนารี

3.3.3 การพัฒนาโปรแกรมบนเครื่องคอมพิวเตอร์

การออกแบบโปรแกรมและหลักการทำงานVisual C# 2010 Express ในการแสดงผลของข้อมูล



รูปที่ 3.12 แผนผังการทำงานของโปรแกรมบนคอมพิวเตอร์

จากรูปที่ 3.12 เมื่อรับข้อมูลมาจาก Serial Port โปรแกรม Visual C# 2010 ที่เขียนขึ้นจะนำข้อมูลที่ได้มาทำการตรวจสอบ ว่าข้อมูลถูกต้องตามการกำหนดไว้หรือไม่โดยรูปแบบข้อมูลที่ถูกต้องคือ

(X1 X2 : Hour :Minute :Second :Milliseconds)

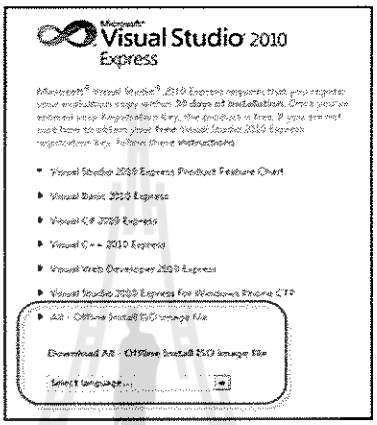
ตำแหน่ง X1	คือ ข้อมูลที่บอกถึงระยะทางของบอร์ดตัวส่ง
ตำแหน่ง X2	คือ ข้อมูลที่บอกปุ่มสวิทซ์ ว่าเป็นสวิทซ์ลำดับใดของบอร์ดตัวส่ง
ตำแหน่ง Hour	คือ ข้อมูลที่บอกเวลาเป็นชั่วโมงของบอร์ดตัวส่ง
ตำแหน่ง Minute	คือ ข้อมูลที่บอกบอกเวลาเป็นนาทีของบอร์ดตัวส่ง
ตำแหน่ง Second	คือ ข้อมูลที่บอกเวลาเป็นวินาทีของบอร์ดตัวส่ง
ตำแหน่ง Milliseconds	คือ ข้อมูลที่บอกเวลาเป็นมิลลิวินาทีของบอร์ดตัวส่ง

เมื่อโปรแกรมตรวจสอบข้อมูลทั้งหมดเสร็จแล้วก็จะนำข้อมูลชุดแรกที่ถูกส่งมาคือข้อมูลจากการกดปุ่มสตาร์ทเข้ามาเก็บไว้ที่ตัวแปร ID จากนั้นเมื่อมีการกดปุ่มสวิทซ์ จุดที่ทำการเช็คระยะ ซึ่งแต่ละระยะนั้นจะส่งข้อมูลมาที่ภาครับ แล้วโปรแกรมจะนำข้อมูลที่เป็นค่าเวลาของการกดสวิทซ์นั้น มาลบกับเวลาจากการกดปุ่มที่จุดสตาร์ท จะทำให้ได้เวลาที่เรือลำนั้นๆผ่านจุดที่ทำการเช็ค โดยจุดแต่ละจุดจะถูกกำหนดด้วยเลข ID เพื่อบอกถึงจุดที่ใช้ในการแข่งขันและลำดับของสวิทซ์ที่ถูกกดมาโดยสวิทซ์แต่ละอันจะบอกถึงลู่ของเรือที่ใช้ในการแข่งขัน เช่น สวิทซ์ที่ 1 บอกถึงลู่ที่ 1 เป็นต้น โดยจะทำอย่างนี้ไปเรื่อยๆจนถึงจุดสุดท้ายของการเช็คระยะ ทำให้ได้เวลาในแต่ละจุดออกมาตลอดจนเวลารวมตั้งแต่เริ่มต้นจนถึงจุดเส้นชัยของการแข่งขัน โดยตัวนับสัญญาณเวลานาฬิกาจริงแต่ล่ะจุดจะมีฐานเวลาที่นับตรงกัน

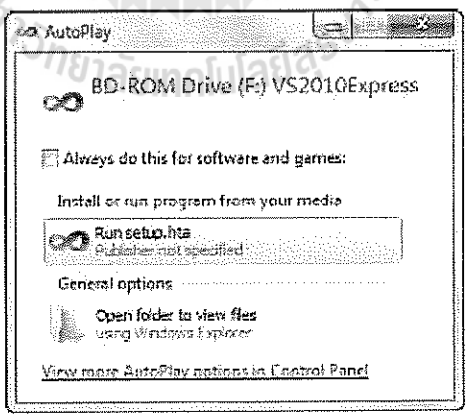
มหาวิทยาลัยเทคโนโลยีสุรนารี

3.3.4 การติดตั้งโปรแกรม VISUAL C# 2010 EXPRESS

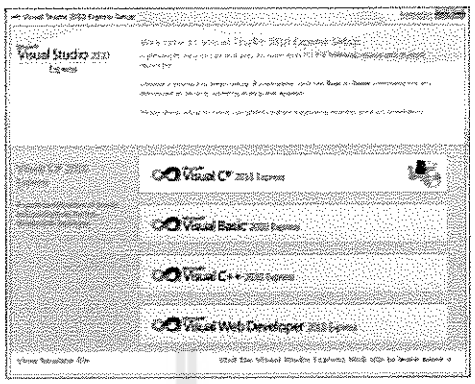
ดาวน์โหลดโปรแกรมติดตั้งได้ที่ URL: http://www.microsoft.com/Express/Downloads ที่รายการดาวน์โหลด Visual Studio 2010 Express กดที่ link: ALL – Offline Install ISO image file (เป็นตัวติดตั้งแบบที่ 1 Offline) จากนั้นเลือกโปรแกรมติดตั้งแบบภาษาอังกฤษจากรายการ Select Language... ดังรูปที่แสดงด้านล่าง



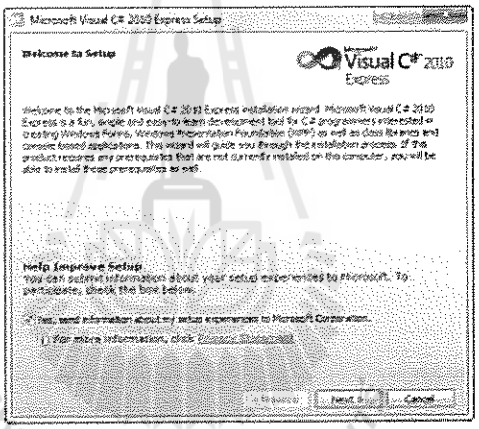
2. เมื่อดาวน์โหลดโปรแกรมติดตั้งเสร็จแล้ว ให้เปิดไฟล์ VS2010Express.iso ด้วยโปรแกรมจำลองไดร์ฟ เช่น DAEMON Tools หรือจะ Burn ลงแผ่น DVD เลยก็ได้ครับ จากนั้นเรียกโปรแกรม Setup.hta โดยเลือกติดตั้งโปรแกรม Visual C# 2010 Express ดังรูป



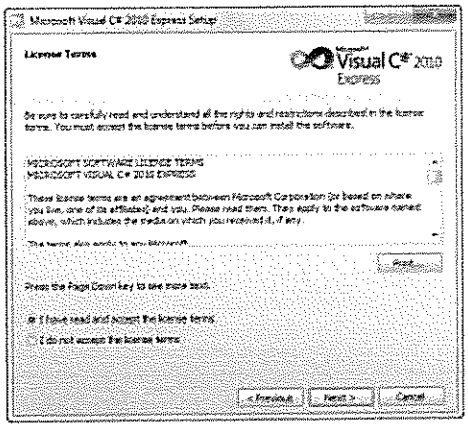
3. เลือกเมนูติดตั้งโปรแกรม ในตัวอย่างนี้เลือก Visual C# 2010 Express



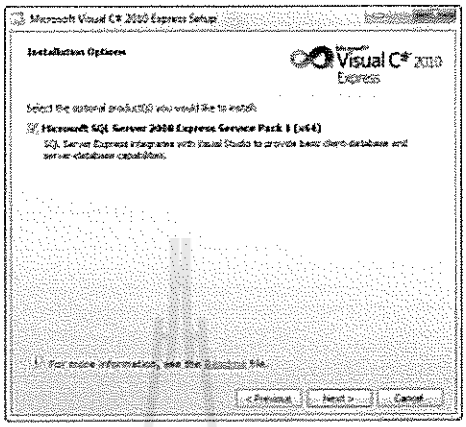
4. โปรแกรมติดตั้งจะแสดงหน้าต่าง Welcome to Setup ขั้นนี้กดปุ่ม Next >



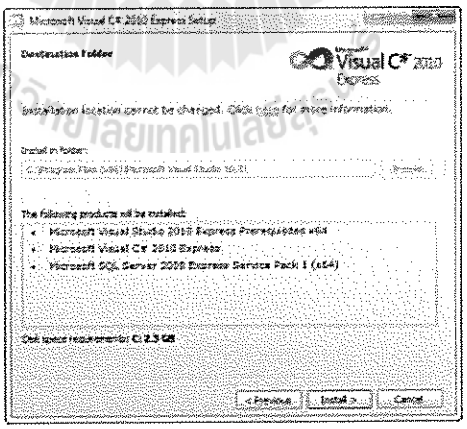
5. อ่านข้อความแสดงลิขสิทธิ์ และการอนุญาตให้ใช้งานโปรแกรม โดยผู้ติดตั้งต้องแสดงการยอมรับ License Terms โดยเลือกที่ I have read and accept the license terms จากนั้นกดปุ่ม Next >



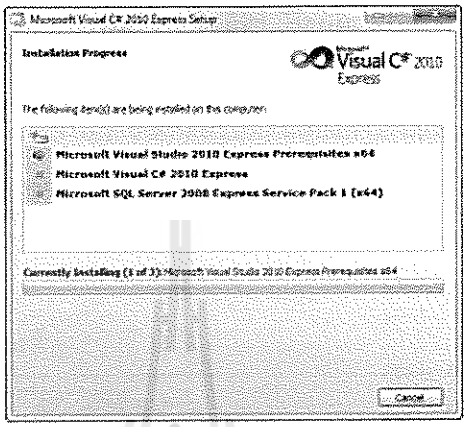
6. เลือกติดตั้งโปรแกรมเสริม โดยเลือกติดตั้ง Microsoft SQL Server 2008 Express Service Pack 1 (x86) ดังรูป จากนั้นกดปุ่ม Next >



7. กำหนดตำแหน่งติดตั้งโปรแกรม โดยระบุที่ช่อง Install in folder หรือกดปุ่ม Browse... เพื่อเลือกตำแหน่งติดตั้งโปรแกรมด้วยตนเอง จากนั้นกดปุ่ม Install >



8. หน้าต่างแสดงความก้าวหน้าในการติดตั้งโปรแกรม ในขณะติดตั้งหรือเมื่อติดตั้งโปรแกรมเสร็จแล้วอาจจะต้องมีการรีสตาร์ทคอมพิวเตอร์



9. เมื่อติดตั้งโปรแกรม Visual C# 2010 Express เสร็จแล้ว ให้ทำขั้นตอนที่ 4 อีกครั้ง โดยเลือกโปรแกรมที่จะติดตั้งเป็น Visual Web Developer 2010 Express และทำตามคำแนะนำของโปรแกรมจนกระทั่งติดตั้งโปรแกรมเสร็จ

มหาวิทยาลัยเทคโนโลยีสุรนารี

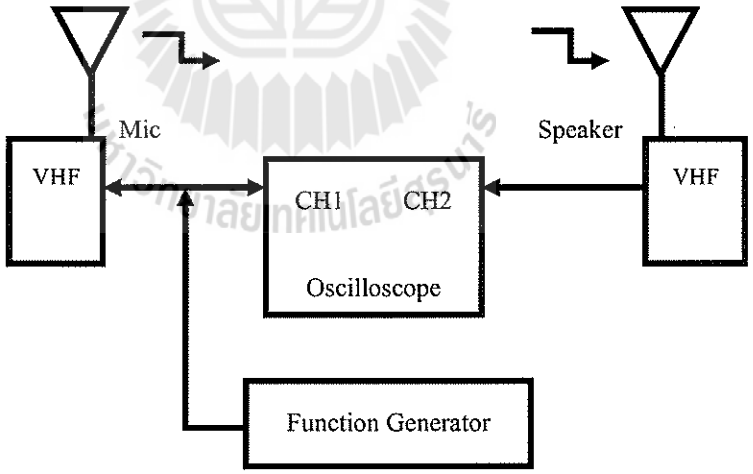
บทที่4 การทดลองและการทดสอบอุปกรณ์

4.1 กล่าวนำ

จากแนวคิดที่ต้องการทดลองการส่งความถี่ 1200 Hz และ 2200 Hz ผ่านวิทยุสื่อสารย่านความถี่ VHF ของวิทยุสมัครเล่นเพื่อเป็นการใช้ข้อมูลยืนยันถึงการออกแบบระบบเพื่อใช้ทดลอง และทดสอบระบบซึ่งการทดสอบมีดังนี้คือ ผลการเปรียบเทียบรูปคลื่นของการรับและส่งข้อมูลแบบ FSK ผ่านวิทยุสื่อสาร การทดสอบการแสดงผลที่จอ LCD (ภาคส่ง) กับ Hyper Terminal (ภาครับ) การทดสอบภาคไมโครคอนโทรลเลอร์เชื่อมต่อกับคอมพิวเตอร์ เพื่อแสดงผลด้วยโปรแกรม VISUAL C# 2010 EXPRESS ผลการเปรียบเทียบความถูกต้องของการส่งและรับที่ระยะต่างๆ และการทดสอบการใช้งานจริงที่กีฬาแห่งชาติเชียงใหม่เกมส์

4.2 ผลการเปรียบเทียบรูปคลื่นของการรับและส่งข้อมูลแบบ FSK ผ่านวิทยุสื่อสาร

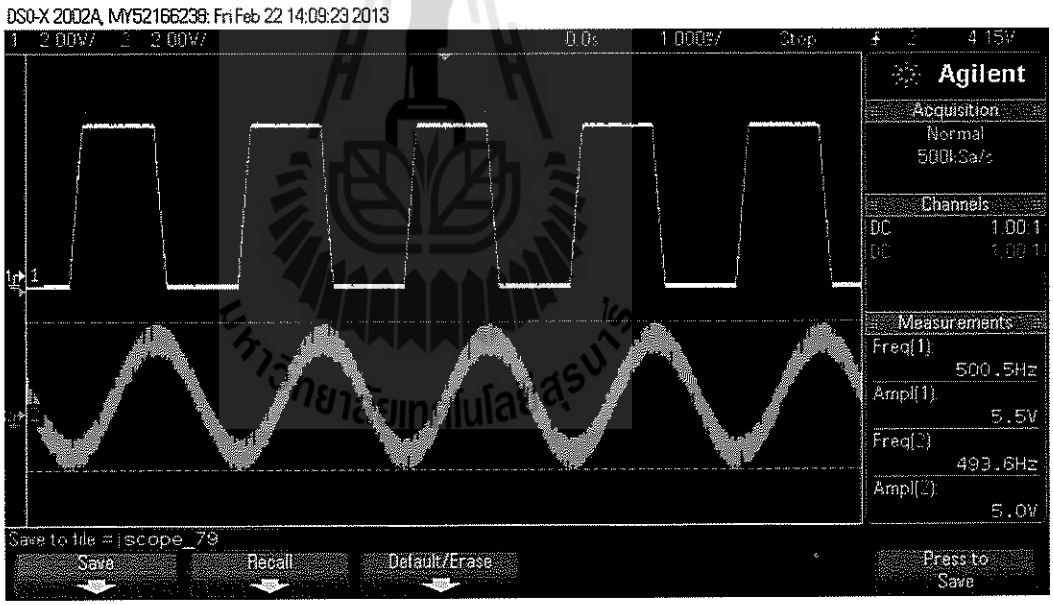
วัตถุประสงค์ของการทดสอบนี้เพื่อใช้ยืนยันเกี่ยวกับแนวคิดที่ต้องการส่งคลื่นแบบ FSK ผ่านอุปกรณ์วิทยุสื่อสาร โดยมีการจัดอุปกรณ์การทดสอบดังรูปที่ 4.1



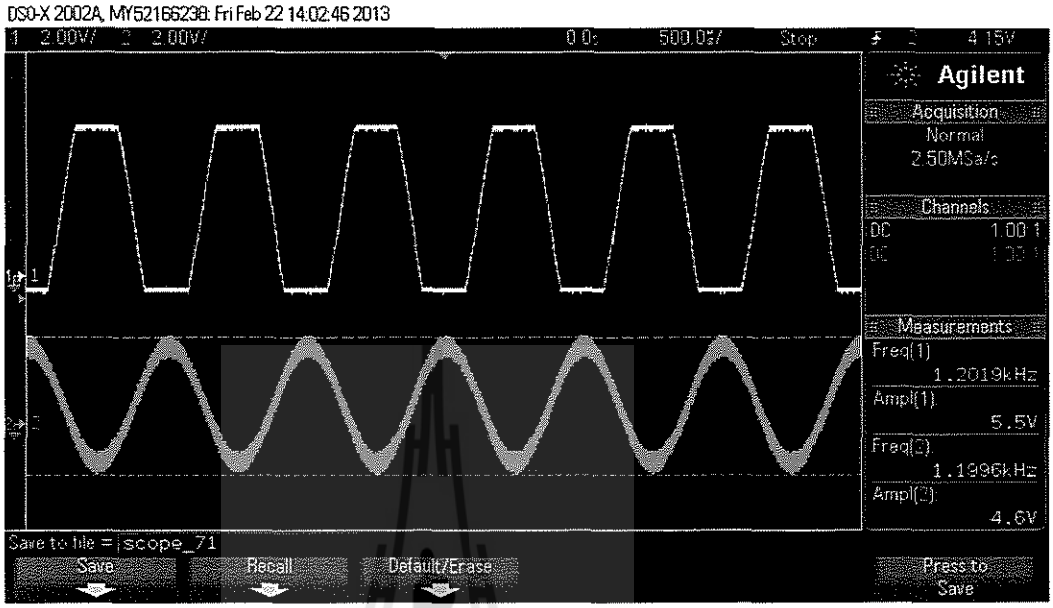
รูปที่ 4.1 แสดงการจัดอุปกรณ์ทดสอบการส่งผ่านความถี่ผ่านวิทยุสื่อสาร

การทดลองนี้จะใช้วิทยุสื่อสาร ICOM IC-V80 ทั้งสองด้าน โดยการติดตั้งอุปกรณ์ประกอบด้วยออสซิลโลสโคปใช้เพื่อวัดสัญญาณที่เข้าและออกจากวิทยุสื่อสารซึ่งได้รับการจ่ายสัญญาณจากฟังก์ชัน

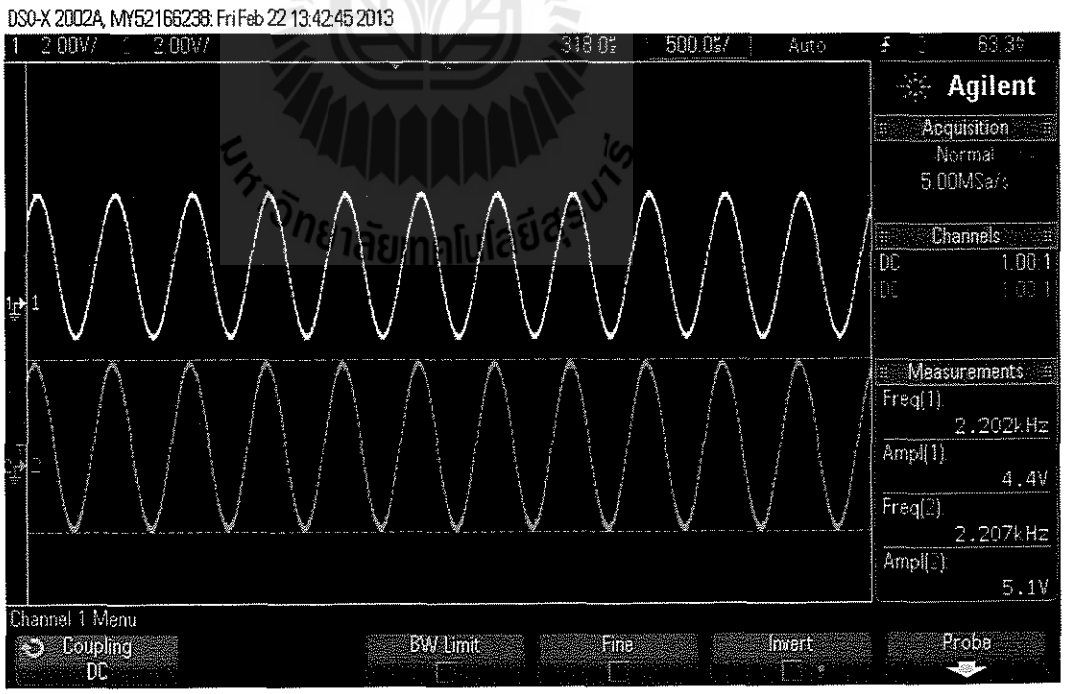
เจนเนอร์เรเตอร์ที่จ่ายความถี่เข้ากับช่องไมค์ของวิทยุสื่อสาร โดยใช้ออสซิลโลสโคปช่อง1 (CH1) วัดความถี่ที่จ่ายให้กับวิทยุสื่อสารภาคส่ง และช่อง 2 (CH2) ใช้วัดความถี่ที่ได้จากวิทยุสื่อสารภาครับโดยต่อจากช่องไมค์ แล้วจ่ายความถี่ 1.5 KHz (ย่านกึ่งกลางของความถี่เสียงมนุษย์ได้ยิน) จากนั้นตั้งช่องสัญญาณการรับ-ส่งของวิทยุให้ตรงกันแล้วปรับขนาดของสัญญาณที่ส่งจนกว่าวิทยุภาครับจะรับสัญญาณได้ต่อไปจึงปรับความถี่ที่ต้องการทดสอบที่ค่าต่างๆและผลที่ได้จากการทดลองสามารถรับและส่งความถี่ที่ 1200 Hz กับ 2200 Hz ได้ดังรูป ขณะที่ความถี่ต่ำกว่า 500 Hz เมื่อส่งสัญญาณออกอากาศผ่านวิทยุสื่อสารภาครับที่รับได้จะเกิดความผิดเพี้ยนของสัญญาณขึ้นส่วนความถี่สูงที่ทดสอบไว้คือ 3000 Hz นั้นจากรูปกราฟที่เกิดขึ้นจะเห็นได้ว่ากราฟมีลักษณะที่กลับเฟสกันระหว่างภาคส่งและภาครับ ดังแสดงในรูปที่ 4.2 - 4.5



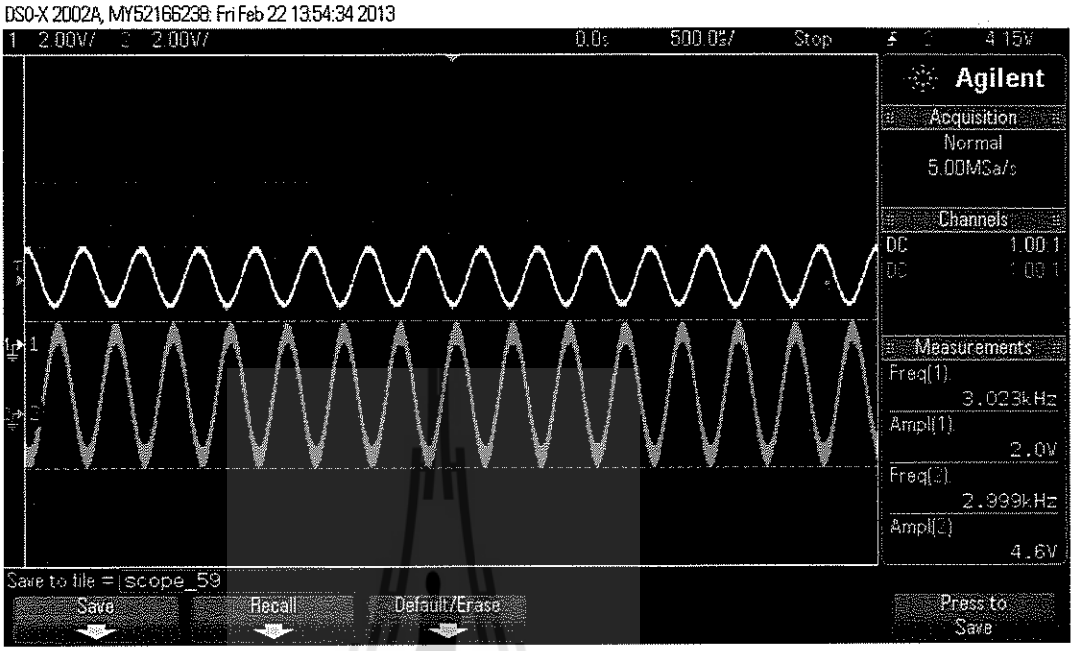
รูปที่ 4.2 แสดงการรับส่งความถี่ 500 Hz



รูปที่ 4.3 แสดงการรับส่งความถี่ 1200 Hz



รูปที่ 4.4 แสดงการรับส่งความถี่ 2200 Hz

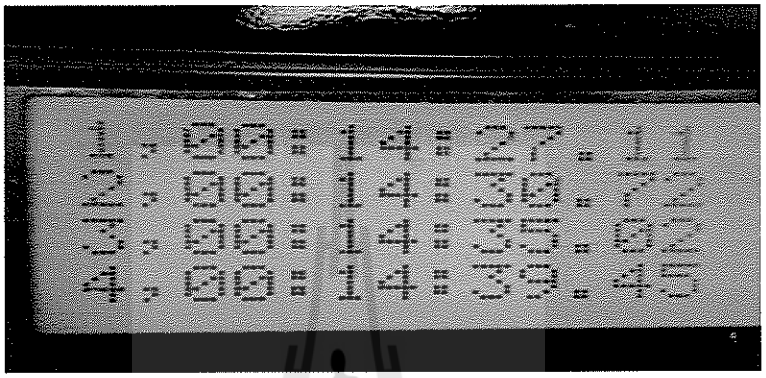


รูปที่ 4.5 แสดงการรับส่งความถี่ 3000 Hz

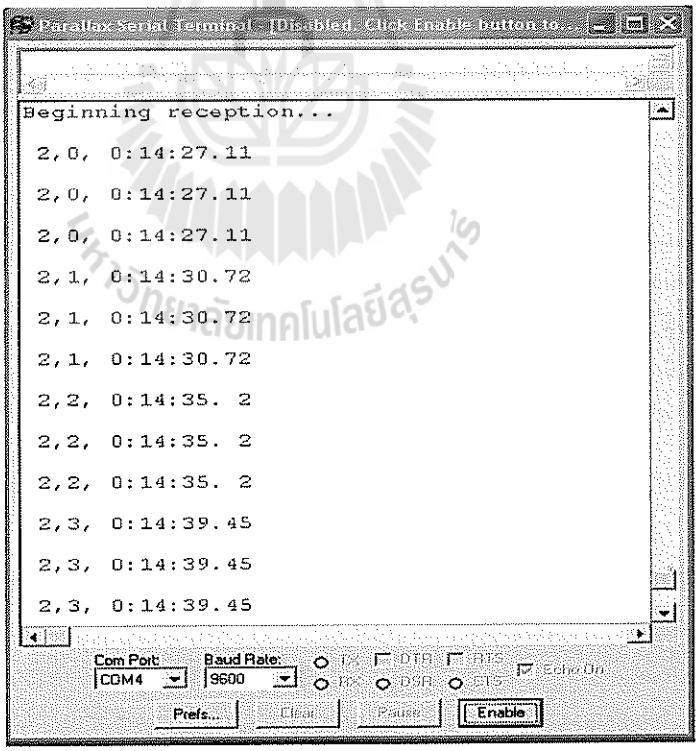
ผลจากการทดลองสามารถสรุปได้ว่าวิทยุสื่อสารสามารถส่งผ่านความถี่ที่สูงกว่า 500 Hz และความถี่ที่ต้องการในการทดสอบคือ 1200 Hz กับ 2200 Hz สามารถส่งผ่านได้ดีและที่ความถี่สูงที่ทดสอบคือ 3000 Hz ยังสามารถรับส่งได้แต่ข้อมูลมีความผิดเพี้ยน เนื่องจากสัญญาณที่ได้มีการกลับเฟสกันของข้อมูลภาคส่งและภาครับ ดังนั้นบอกได้ว่าถ้าต้องการส่งข้อมูลในลักษณะของความถี่แบบ FSK แล้ว ต้องเลือกช่วงความถี่ที่ 1200 Hz และ 2200 Hz เพราะจากลักษณะของสัญญาณที่แสดงให้เห็นว่า ภาครับสามารถรับสัญญาณที่มีลักษณะใกล้เคียงกันกับสัญญาณที่ภาคส่งส่งมาที่ความถี่ทั้งสองนี้

4.3 ผลการทดสอบการแสดงผลที่จอ LCD (ภาคส่ง) กับ Hyper Terminal (ภาครับ)

การทดสอบนี้เป็นการทดสอบเพื่อศึกษาความถูกต้องของภาครับและภาคส่งว่าข้อมูลที่ภาคส่งส่งมาตรงกับข้อมูลที่ภาครับรับได้หรือไม่โดยมีการทดสอบดังนี้



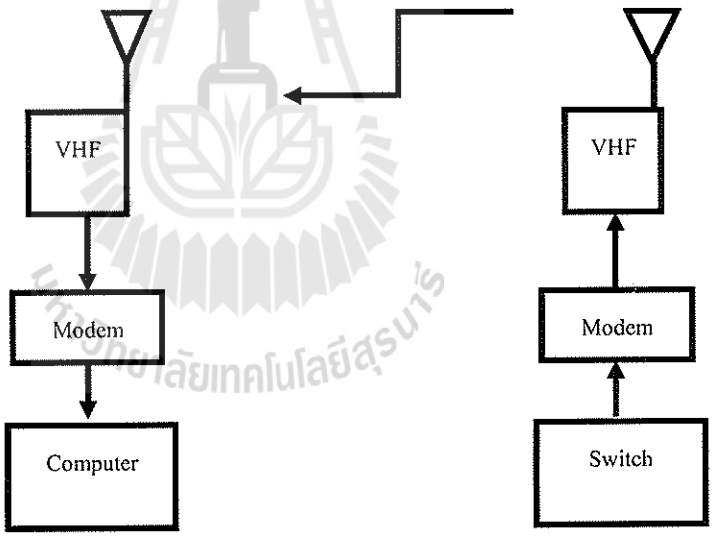
รูปที่ 4.6 แสดงผลที่จอ LCD (ภาคส่ง)



รูปที่ 4.7 แสดงผลที่ Hyper Terminal (ภาครับ)

จากข้อมูลที่รับเข้ามาเป็นการเปรียบเทียบความถูกต้องระหว่างข้อมูลภาคส่งกับภาครับ โดยรูปที่ 4.6 เป็นข้อมูลของภาคส่งที่แสดงบนจอ LCD และรูปที่ 4.7 เป็นข้อมูลของภาครับที่แสดงผ่าน Hyper -Terminal จะเห็นว่าข้อมูลทั้งสอง มีความถูกต้องและสามารถแสดงผลข้อมูลได้อย่างครบถ้วน โดยข้อมูลสองตัวแรกระบุเลข ID Switch เพื่อบอกถึงตำแหน่ง และเป็นลำดับของสวิทช์ ส่วนข้อมูลชุดหลังบอกถึงเวลาที่ถูกกดมาจากภาคส่ง

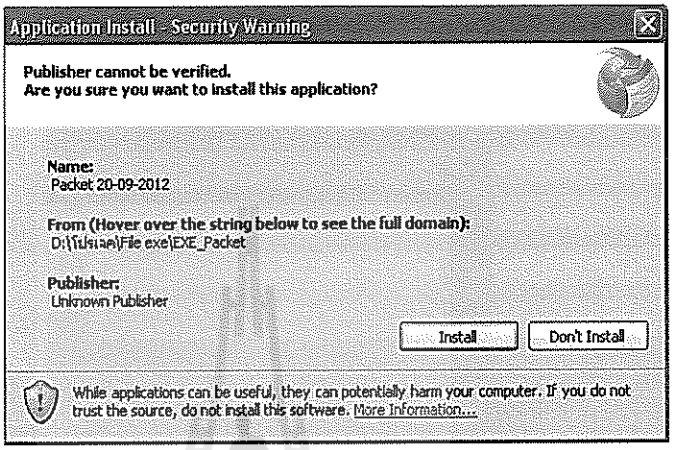
4.4 การทดสอบไมโครคอนโทรลเลอร์เชื่อมต่อกับคอมพิวเตอร์ เพื่อแสดงผลด้วยโปรแกรม VISUAL C# 2010 EXPRESS



รูปที่ 4.8 แสดงการทดสอบการส่งข้อมูลเพื่อแสดงผลผ่าน VISUAL C# 2010 EXPRESS

โปรแกรมจับเวลาการแข่งขันเรือพาย วิธีติดตั้งโปรแกรม จับเวลาเรือพาย

1.รัน Setup.exe แล้วกด Install ดังรูป

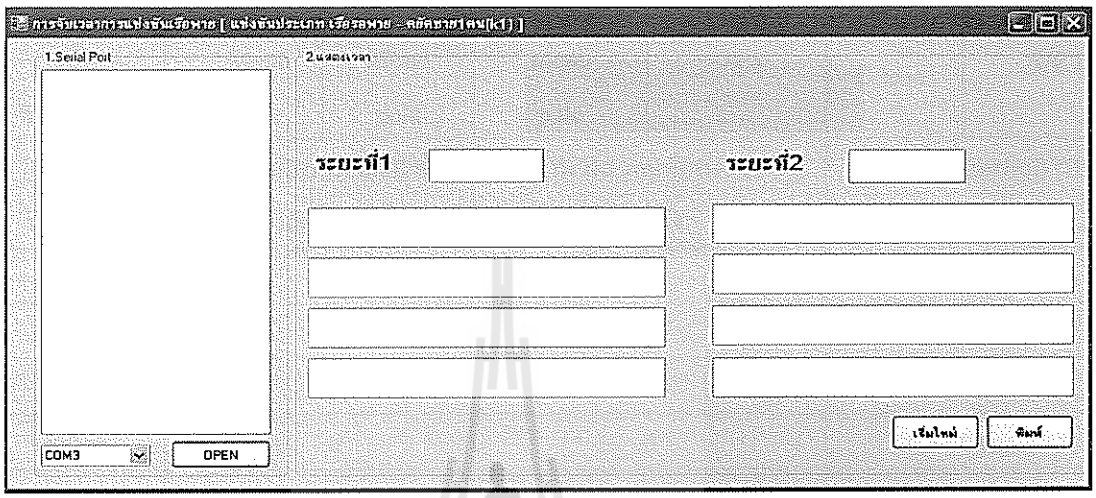


รูปที่ 4.9 การ Install โปรแกรม

2.จะปรากฏหน้าต่างโปรแกรมดังรูป

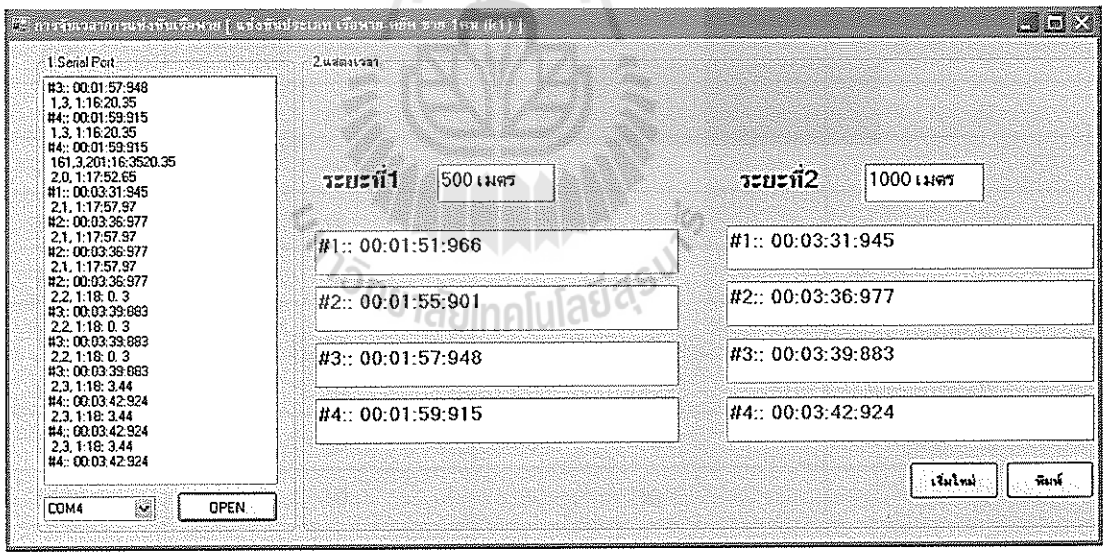
รูปที่ 4.10 หน้าต่างโปรแกรมส่วนที่ 1

โปรแกรมส่วนที่ 2 นี้ จะเปิดขึ้นมาเมื่อทำการกรอกข้อมูลแล้วคลิกคำว่าเริ่มการแข่งขัน ที่โปรแกรมส่วนที่ 1



รูปที่ 4.11 หน้าต่างโปรแกรมส่วนที่ 2

ผลที่ได้จากการทดสอบระบบ



รูปที่ 4.12 ผลการทดสอบการส่งข้อมูลเพื่อแสดงผลที่คอมพิวเตอร์

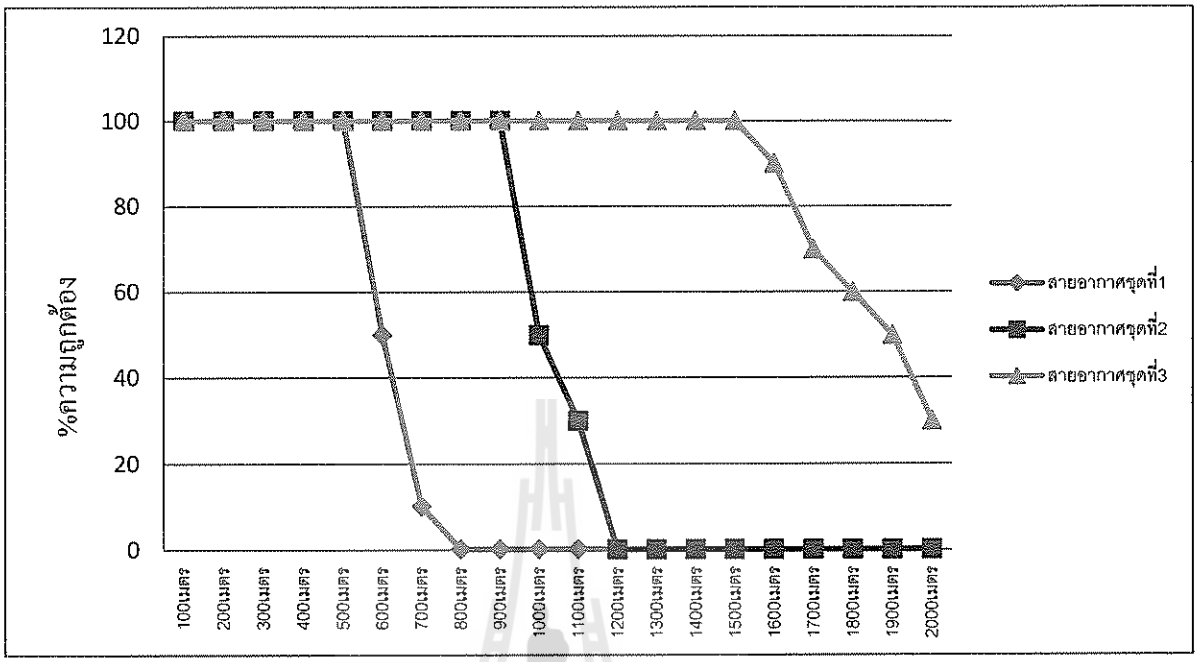
จะเห็นได้ว่าข้อมูลที่ส่งมาสามารถแสดงผลผ่านโปรแกรม VISUAL C# 2010 EXPRESS ทางหน้าจอคอมพิวเตอร์ได้เป็นอย่างดีและข้อมูลไม่มีความผิดเพี้ยนจากข้อมูลต้นทางที่ส่งมา และเวลาที่แสดงมีความถูกต้องชัดเจนน่าเชื่อถือ ใช้เป็นหลักฐานอ้างอิงเพื่อการตัดสินแพ้ชนะได้

4.5 ผลการเปรียบเทียบความถูกต้องของการส่งและรับที่ระยะต่างๆ

ชนิดสายอากาศ	ภาคส่ง Mono pole	ภาครับ Mono pole	ภาคส่ง Mono pole	ภาครับ 5/8 λ	ภาคส่ง C-load	ภาครับ 5/8 λ
ระยะทาง	จำนวนErrorต่อการส่ง 10ครั้ง		จำนวนErrorต่อการส่ง 10ครั้ง		จำนวนErrorต่อการส่ง 10ครั้ง	
100 m	0		0		0	
200 m	0		0		0	
300 m	0		0		0	
400 m	0		0		0	
500 m	0		0		0	
600 m	5		0		0	
700 m	9		0		0	
800 m	10		0		0	
900 m	10		0		0	
1000 m	10		5		0	
1100 m	10		7		0	
1200 m	10		10		0	
1300 m	10		10		0	
1400 m	10		10		0	
1500 m	10		10		0	
1600 m	10		10		1	
1700 m	10		10		3	
1800 m	10		10		4	
1900 m	10		10		5	
2000 m	10		10		7	

รูปที่ 4.13 ตารางแสดงผลการเปรียบเทียบความถูกต้องของข้อมูล

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 4.14 กราฟแสดงเปอร์เซ็นต์การเปรียบเทียบความถูกต้องของข้อมูล ในการส่งที่ระยะต่างๆ

วัตถุประสงค์ของการทดสอบนี้ เพื่อทดสอบหาระยะทางที่ข้อมูลสามารถส่งไปได้ จากผลการทดสอบจะทำให้สามารถนำข้อมูลมาวิเคราะห์ได้ดังนี้ เมื่อสายอากาศที่ใช้มีอัตราขยายต่ำ ระยะทางที่สามารถส่งข้อมูลได้ จะมีค่าลดลงเรื่อยๆเมื่อระยะทางไกลขึ้น และเมื่อเพิ่มอัตราขยายของสายอากาศข้อมูลจะสามารถส่งได้ไกลขึ้น โดยสามารถพิจารณาจากกราฟ ซึ่งเป็นเปอร์เซ็นต์ของความสามารถในการส่งข้อมูลของสายอากาศแบบต่างๆ เมื่อนำมาใช้งานทดสอบในการส่งที่ระยะต่างๆ โดยสายอากาศชุดแรกมีอัตราขยายต่ำสุด จะสามารถส่งข้อมูลได้ที่ 100 % เมื่อระยะทาง 500 เมตร และจะลดลงเหลือ 50 % ระยะทางประมาณ 600 เมตร หลังจากนั้นรับข้อมูลผิดเพี้ยนและรับข้อมูลไม่ได้ เมื่อพิจารณาสายอากาศชนิดที่สามซึ่งมีอัตราขยายที่สูงสุดในการทดสอบ โดยสายอากาศชุดนี้มีความสามารถการส่งข้อมูลได้ 100 % อยู่ที่ระยะทางประมาณ 1650 เมตร และลดลงเหลือ 80 % ที่ระยะทาง 1700 เมตร และข้อมูลจะลดลงเหลือ 50 % ที่ระยะทาง 1900 เมตร หลังจากนั้นข้อมูลจะผิดเพี้ยนและรับไม่ได้ ซึ่งจากการทดสอบนี้จะบอกถึงความสามารถในการส่งข้อมูล เพื่อนำไปใช้ในการใช้งานที่ระยะทางต่างๆได้อย่างเหมาะสมและถูกต้อง

4.6 การทดสอบการใช้งานจริงที่กีฬาแห่งชาติเชียงใหม่เกมส์

รูปที่ 4.15 การทดสอบการใช้งานจริงที่กีฬาแห่งชาติเชียงใหม่เกมส์

ผลที่ได้จากการจับเวลาของกล้องโฟโตฟินิช

รายการ เรือพาย - คยัค ชาย 1 คน (k1) 200 เมตร สาย ไม่ระบุ รอบ รอบรองชนะเลิศ ฮีท Heat 2
วันที่ 14 ธันวาคม 2555 เวลา 08:30 น.
สนาม อ่างเก็บน้ำห้วยตึงเฒ่า อ.แม่ริม

สถานะ	จบการแข่งขัน
ผล	เป็นทางการ

ลำดับ	นักกีฬา		สถิติ (hh:mm:ss:s)	หมายเหตุ (Remark)	ลำดับ (Rank)
6	ชาญฤทธิ์ ฉากเขียน	สิงห์บุรี	00:00:38.41		1
8	วรรธนะ จิตต์พันธุลาภ	เชียงใหม่	00:00:39.35		2
5	เกษมสิทธิ์ บริบูรณ์วศิน	กรุงเทพมหานคร	00:00:40.24		3
7	เทิดศักดิ์ สนธิ	นครสวรรค์	00:00:44.45		4

ผลที่ได้จากเครื่องจับเวลา

ชื่อนักกีฬา	สังกัด	สถิติ	ลำดับ
ชาญฤทธิ์ ฉากเขียน	สิงห์บุรี	00:00:38.59	1
วรรธนะ จิตต์พันธุลาภ	เชียงใหม่	00:00:39.45	2
เกษมสิทธิ์ บริบูรณ์วศิน	กรุงเทพมหานคร	00:00:40.33	3
เทิดศักดิ์ สนธิ	นครสวรรค์	00:00:44.50	4

ผลการจับเวลาจากกล้องโฟโตฟินิช

รายการ เรือพาย - คยัด หญิง 1 คน (k1) 200 เมตร สาย ไม่ระบุ รอบ รอบรองชนะเลิศ ฮีท Heat 2
วันที่ 14 ธันวาคม 2555 เวลา 09:10 น.
สนาม อ่างเก็บน้ำห้วยตึงเฒ่า อ.แม่ริม

สถานะ	จบการแข่งขัน
ผล	เป็นทางการ

ลำดับ	นักกีฬา		สถิติ (hh:mm:ss)	หมายเหตุ (Remark)	ลำดับ (Rank)
5	กนกพรรณ สวนสันต์	สุพรรณบุรี	00:00:45.64		1
6	ศุภัทรา เดหา	เชียงใหม่	00:00:48.75		2
7	ชญานิน ศรีผดุง	สมุทรปราการ	00:00:49.67		3

ผลการจับเวลาจากเครื่องจับเวลาเรือพาย

ชื่อนักกีฬา	สังกัด	สถิติ	ลำดับ
กนกพรรณ สวนสันต์	สุพรรณบุรี	00:00:45.70	1
ศุภัทรา เดหา	เชียงใหม่	00:00:48.80	2
ชญานิน ศรีผดุง	สมุทรปราการ	00:00:49 72	3

จากผลการทดสอบจะเห็นได้ว่าข้อมูลเวลาที่ได้นั้นมีความคลาดเคลื่อน ระหว่างเวลาที่ได้จากกล้องโฟโตฟินิช และจากเครื่องจับเวลา ซึ่งมีความคลาดเคลื่อนเป็นมิลลิวินาที และมีความคลาดเคลื่อนโดยเฉลี่ยแล้วประมาณ 10 มิลลิวินาที โดยสาเหตุที่ทำให้เวลาเกิดความคลาดเคลื่อนนั้น เกิดจากการกดจับเวลาของนาฬิกาที่จุดสตาร์ท และจุดเส้นชัยไม่พร้อมกัน ระหว่างกล้องโฟโตฟินิชกับเครื่องจับเวลาซึ่งใช้กรรมการจับเวลาคนล่ะคนกันในการจับเวลา และมีแนวทางในการแก้ไขโดยการกดปุ่มจับเวลาที่จุดสตาร์ทและจุดเส้นชัยให้พร้อมกัน ถึงจะได้เวลาออกมาได้ใกล้เคียงกัน และมีความคลาดเคลื่อนน้อยที่สุด โดยข้อมูลที่ได้นำมาพิจารณานี้เป็นเพียงส่วนหนึ่งของการทดสอบการจับเวลา และข้อมูลส่วนที่เหลือนั้นมีความคลาดเคลื่อนเช่นเดียวกันกับข้อมูลที่ได้กล่าวมาแล้ว

4.7 กล่าวสรุป

จากผลการทดลองและทดสอบในส่วนต่างๆของระบบ ทำให้ทราบผลการทดลองและทดสอบมาว่าอุปกรณ์และระบบที่ใช้นั้นมีความสามารถในการใช้งานและมีความเที่ยงตรงถูกต้องหน้าเชื่อถือได้ ซึ่งเป็นไปตามที่ได้ออกแบบไว้ โดยอุปกรณ์และระบบนี้สามารถนำไปใช้งานได้จริง ซึ่งมีผลการทดลองและทดสอบระบบเฉพาะส่วนต่างๆดังนี้

-ผลการเปรียบเทียบรูปคลื่นของการรับและส่งข้อมูลแบบ FSK ผ่านวิทยุสื่อสาร

-ผลการทดสอบการแสดงผลที่จอ LCD (ภาคส่ง) กับ Hyper Terminal (ภาครับ)

-การทดสอบไมโครคอนโทรลเลอร์เชื่อมต่อกับคอมพิวเตอร์ เพื่อแสดงผลด้วยโปรแกรม VISUAL C# 2010 EXPRESS

-ผลการเปรียบเทียบความถูกต้องของการส่งและรับที่ระยะต่างๆ

-การทดสอบการใช้งานจริงที่กีฬาแห่งชาติเชียงใหม่เกมส์

ผลการทดสอบทำให้ทราบถึงความแตกต่างของระบบ โดยส่วนสำคัญคือการเปลี่ยนแปลงสายอากาศและโปรแกรม รวมไปถึงการแก้ไขในส่วนต่างของระบบเพื่อให้สามารถรับ-ส่งข้อมูลได้อย่างเหมาะสมกับการทำงานผ่านวิทยุสื่อสารและฟังก์ชั่นการใช้งานในลักษณะต่างๆ ที่มีเพื่อช่วยให้การทำงานของระบบมีความเที่ยงตรงและแม่นยำมากยิ่งขึ้น จากผลการทดสอบของระบบทั้งหมดสามารถทำงานได้ดี โดยภาคไมโครคอนโทรลเลอร์กับตัวนับสัญญาณเวลานาฬิกาจริง สามารถรับค่าและส่งค่าระหว่างกันได้อย่างถูกต้อง และภาคสวิทซ์กับตัวนับสัญญาณเวลานาฬิกาจริงมีความแม่นยำสามารถส่งข้อมูลระหว่างอุปกรณ์ได้อย่างถูกต้อง ในส่วนของโปรแกรมที่ใช้ในการควบคุมนั้นมีความเสถียรและใช้งานจริงได้เป็นอย่างดี สำหรับภาครับนั้นไมโครคอนโทรลเลอร์สามารถถอดข้อมูลที่ส่งไปได้ถูกต้อง และสามารถนำมาแสดงผลร่วมกับโปรแกรม Visual C# 2010 Express ได้ดีโดยสามารถแสดงเวลาผ่านโปรแกรม ได้อย่างแม่นยำตามที่ภาคส่งได้ส่งข้อมูลเวลามาให้ โดยสามารถพิจารณาได้จากจำนวนข้อมูลที่ผิดพลาด (error)

บทที่ 5
บทสรุป

5.1 สรุป

โครงงานนี้ได้ทำขึ้นเพื่อแก้ปัญหาเรื่องการตัดสินการแข่งขันกีฬาเรือพาย โดยปัจจัยสำคัญในการแข่งขันนั้นผลแพ้-ชนะตัดสินกันด้วยเวลา ปัญหาสำคัญคือวิธีการจับเวลามักมีความคลาดเคลื่อน เพราะใช้การจับเวลาจากนาฬิกาจับเวลาที่ควบคุมโดยคน ความคลาดเคลื่อนมักเกิดขึ้นในขณะกดหยุดเวลาและเกิดจากแต่ละจุดมีการใช้เวลไม่เท่ากัน ดังนั้นการแก้ปัญหาคือ ใช้ระบบส่งเวลาจากจุดสตาร์ทและจุดเส้นชัยโดยส่งการกดเวลาผ่านวิทยุสื่อสาร ซึ่งใช้ฐานเวลาที่เท่ากัน โดยใช้ตัว IC Real Time Clock (pcf8583) เป็นตัวส่งค่าเวลาให้คอนโทรเลอร์ จากนั้นตัวคอนโทรลเลอร์จะแปลงข้อมูลเป็นความถี่ในลักษณะแบบ FSK เพื่อส่งออกไปยังวิทยุสื่อสาร ซึ่งผลการทดสอบและใช้งานจริงระบบสามารถส่งเวลาได้ถูกต้องและมีความเที่ยงตรง ไม่มีการผิดเพี้ยนของข้อมูล และในส่วนของโปรแกรมที่ใช้แสดงผล สามารถแสดงผลข้อมูลได้ถูกต้องโดยส่วนแสดงผลสามารถนำเวลาจากจุดสตาร์ทมาลบกับจุดต่างๆที่ต้องการทราบเวลาได้ และทำการแสดงเวลาที่ใช้งานจริงได้อย่างถูกต้องตามข้อมูลที่ได้รับมา

5.2 ปัญหาและอุปสรรคในการพัฒนา

1. เนื่องจากอุปกรณ์ที่ใช้เป็นอุปกรณ์ขนาดเล็กจึงต้องใช้ความระมัดระวังในการสร้างวงจรและต้องระมัดระวังเรื่องการจ่ายไฟเลี้ยงให้แก่อุปกรณ์ เพราะถ้าจ่ายเกินที่กำหนดจะทำให้อุปกรณ์เกินการเสียหายในทันที

2. เนื่องจากสัญญาณนาฬิกาจริงนั้นส่งสัญญาณออกมา ในรูปของรหัสแอสกี้ซึ่งไม่สามารถที่จะใช้ออสซิลโลสโคป ในการวิเคราะห์สัญญาณได้

3. ในการเขียนโปรแกรมภาษาสปินที่ใช้รับค่าจากสัญญาณนาฬิกาจริงและนำมาแสดงผลนั้น ค่อนข้างลำบากเนื่องจากเป็นโปแกรมภาษาใหม่ทำให้หาข้อมูลและผู้เชี่ยวชาญให้คำปรึกษายาก

4. ในการรับข้อมูลจากสัญญาณนาฬิกาจริงนั้น ค่าที่ได้มีการเปลี่ยนแปลงตลอดเวลาทุกๆ การรับข้อมูลใหม่เนื่องจากสัญญาณนาฬิกาจะวิ่งเรื่อยๆ และจะถูกส่งมาตลอด ทำให้การตรวจจับข้อมูลมีความคลาดเคลื่อนได้

5. เนื่องจากการรับส่งสัญญาณเป็นแบบ Half-duplex ซึ่งอาจมีการแทรกทรอดของวิทยุที่ใช้คลื่นเดียวกันได้ ทำให้เกิดการผิดเพี้ยนของข้อมูลได้

6. อุปกรณ์ รับ-ส่ง สัญญาณเสียงเมื่ออยู่ในระยะไกลขึ้นสัญญาณจะไม่ชัดเจน ทำให้การรับข้อมูลมีความผิดพลาดได้

5.3 ข้อเสนอแนะ

1. ขนาดของตัวอุปกรณ์สามารถทำให้มีขนาดเล็กลง และมีคุณสมบัติการใช้งานมากขึ้น

2. สามารถนำระบบไปใช้โดยเพิ่มอุปกรณ์รับส่งข้อมูลที่มากขึ้นเป็นระบบโครงข่ายที่มีขนาดเล็ก สามารถนำไปใช้ในระบบการจัดการระบบควบคุมระยะไกลต่างๆได้

3. ถ้าต้องการให้ข้อมูลส่งได้ในระยะทางไกลขึ้นควรใช้สายอากาศที่มีเกณในการส่งสูง และวิทยุสื่อสารที่มีกำลังส่งมากๆ

4. สามารถนำไปประยุกย์ใช้งานในการจับเวลาการแข่งขันกีฬาชนิดอื่นที่ต้องใช้การจับเวลาในลักษณะเดียวกันนี้

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ก Code

Code โปรแกรมควบคุมไมโครคอนโทรลเลอร์

------ Main program ------

```
CON                                     // กำหนดค่าคงที่
XMT         = true                      // true = 1
USING_BP    = false                     // false = 0
RCVPIN      = 17                        // ขาอินพุต
XMTPIN      = 22                        //ขาเอาท์พุต
PTTPIN      = 23                        //PTTวิทยุ
SW_00       = 4                         //สวิทซ์1
SW_01       = 5                         //สวิทซ์2
SW_02       = 6                         //สวิทซ์3
SW_03       = 7                         //สวิทซ์4
SwResetTime = 26                        //สวิทซ์รีเซตเวลา
Beep        = 16                        //บัซเซอร์
LED_Blue    = 27
LED_Green   = 0
LED_Yellow  = 1
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
LED_Red         = 2

LED_Orange      = 3

flag_On         = 1                                  //กำหนดสถานะ flag_On =high(1)

flag_Off        = 0                                  //กำหนดสถานะ flag_Off = low(0)

RS              = 8                                  //ขาRSของLCD

RW              = 9                                  //ขาRWของLCD

E               = 10                                 //ขาEของLCD

DBLow           = 11                                 //Dlowส่งให้LCD

DBHigh          = 14                                 //Dhighส่งให้LCD

CR              = 13

LF              = 10

PCF8583_Addr    = %1010_0000                         //กำหนดAddressให้PCF8583

i2cSCL          = 24                                 //ขาสัญญาณนาฬิกา

i2cSDA          = 25                                 //ขาสัญญาณข้อมูล

_clkmode        = (xtal1 + pll8x) & USING_BP | (xtal1 + pll16x) & !USING_BP  //กำหนด
โหมด

_xinfreq        = 10_000_000 & USING_BP | 5_000_000 & !USING_BP       //กำหนดความถี่

OBJ                                          // เพิ่มObjectที่ใช้งานในโปรแกรม

mdm   :   "bell202_modem"                    //เพิ่ม"bell202_modem"Objectโมเด็ม
```

```
num   :   "simple_numbers"          //เพิ่ม"simple_numbers"Objectแปลงเลข

PCF8583Obj: "Obj_PCF8583"           //เพิ่ม"Obj_PCF8583"Objectสัญญาณเวลานาฬิกาจริง

LCD :     "LCD_Nx2"                 //เพิ่ม"LCD_Nx2"Objectแสดงผลจอแอลซีดี

VAR                                                    //กำหนดตัวแปล

byte  Hours[5],Minutes[5],Seconds[5],Seconds100[5],Switch[4]

byte  strbuf[64]

byte Id_No, Digit0, Digit1, Digit2, Digit3

byte flagSw0,flagSw1,flagSw2,flagSw3

long stack[400]

byte H,M,S,S100,ode,new


PUB main                                               //โปรแกรมเมน

cognew(SendDataTone, @stack[100])                      //เพิ่มcogใหม่ให้ทำหน้าที่ส่งโทน
ข้อมูล

cognew(GetSwith, @stack[200])                  //เพิ่มcogใหม่ให้ทำหน้าที่เก็บค่าเวลา

cognew(LCD_Disp, @stack[300])                          //เพิ่มcogใหม่ให้ทำหน้าที่แสดงเวลา


PUB SendDataTone                                       //ส่วนส่งโทนข้อมูล

flagSw0 := flag_Off                                    //กำหนด lagSw0=low
```

```
flagSw1 := flag_Off                                    //กำหนด lagSw1=low

flagSw2 := flag_Off                                    //กำหนด lagSw2=low

flagSw3 := flag_Off                                    //กำหนด lagSw3=low


mdm.start_simple(RCVPIN, XMTPIN, PTTPIN, 0) //เรียกใช้ฟังก์ชัน start_simpleใน
bell202_modem

repeat                                                 //วนลูปไม่รู้จบ

if  flagSw0 == flag_On                                 //ถ้าตัวแปร flagSw0 = high

transmit_cmd(0)              //เรียกใช้งานโปรแกรมย่อย transmit_cmd

transmit_cmd(0)              //เรียกใช้งานโปรแกรมย่อย transmit_cmd

transmit_cmd(0)              //เรียกใช้งานโปรแกรมย่อย transmit_cmd

flagSw0 := flag_Off                    //ให้ตัวแปร flagSw0 = low

if  flagSw1 == flag_On                                 //ถ้าตัวแปร flagSw1 = high

transmit_cmd(1)              //เรียกใช้งานโปรแกรมย่อย transmit_cmd

transmit_cmd(1)              //เรียกใช้งานโปรแกรมย่อย transmit_cmd

transmit_cmd(1)              //เรียกใช้งานโปรแกรมย่อย transmit_cmd

flagSw1 := flag_Off                    //ให้ตัวแปร flagSw1 = low

if  flagSw2 == flag_On                                 //ถ้าตัวแปร flagSw2 = high

transmit_cmd(2)              //เรียกใช้งานโปรแกรมย่อย transmit_cmd
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
transmit_cmd(2)                 //เรียกใช้งานโปรแกรมย่อย transmit_cmd
transmit_cmd(2)                 //เรียกใช้งานโปรแกรมย่อย transmit_cmd
flagSw2 := flag_Off                     //ให้ตัวแปร flagSw2 = low
if  flagSw3 == flag_On                          //ถ้าตัวแปร flagSw3 = high
transmit_cmd(3)                 //เรียกใช้งานโปรแกรมย่อย transmit_cmd
transmit_cmd(3)                 //เรียกใช้งานโปรแกรมย่อย transmit_cmd
transmit_cmd(3)                 //เรียกใช้งานโปรแกรมย่อย transmit_cmd
flagSw3 := flag_Off                     //ให้ตัวแปร flagSw3 = low
PUB GetSwith                            //ส่วนอ่าค่าสวิทซ์
PCF8583Obj.init(PCF8583_Addr, i2cSDA, i2cSCL)  //กำหนดค่าเริ่มต้นให้ไอซีสัญญาณเวลา
นาฬิกาจริง
dira[Beep] := 1                 //กำหนดP16เป็นเอาท์พุต
repeat                                  //วนลูปไม่รู้จบ
outa[Beep] := 0                 //กำหนดให้P16มีสถานะlow
GetTime(4)                              //อ่านค่าเวลา
if  ina[SW_00] == 0                     //ถ้าสวิทซ์1มีค่าเท่ากับlow
outa[Beep] := 1                 //กำหนดให้P16มีสถานะhigh
GetTime(0)                      //อ่านค่าเวลา0
GetID                           //อ่านค่าจากดิพสวิทซ์
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
flagSw0 := flag_On                    //ให้ตัวแปร flagSw0 = high
Switch[0] := 0                        //ให้สวิทซ์1 = 0
waitcnt(cnt + 10_000_000)             //หน่วงเวลา
REPEAT WHILE ina[SW_00] == 0                  //วนลูปถ้า SW_00 = low
GetTime(4)                            //อ่านค่าเวลา
waitcnt(cnt + 10_000_000)             //หน่วงเวลา
if  ina[SW_01] == 0                           //ถ้าสวิทซ์2มีค่าเท่ากับlow
outa[Beep] := 1                       //กำหนดให้P16มีสถานะhigh
GetTime(1)                            //อ่านค่าเวลา1
GetID                                 //อ่านค่าจากดิพสวิทซ์
flagSw1 := flag_On                    //ให้ตัวแปร flagSw1 = high
Switch[1] := 1                        //ให้สวิทซ์2 = 1
waitcnt(cnt + 10_000_000)             //หน่วงเวลา
REPEAT WHILE ina[SW_01] == 0                  //วนลูปถ้า SW_01 = low
GetTime(4)                            //อ่านค่าเวลา
waitcnt(cnt + 10_000_000)             //หน่วงเวลา
if  ina[SW_02] == 0                           //ถ้าสวิทซ์3มีค่าเท่ากับlow
outa[Beep] := 1                       //กำหนดให้P16มีสถานะhigh
GetTime(2)                            //อ่านค่าเวลา2
```

```
GetID                                   //อ่านค่าจากดิพสวิทซ์
flagSw2 := flag_On                      //ให้ตัวแปร flagSw2 = high
Switch[2] := 2                          //ให้สวิทซ์3 = 2
waitcnt(cnt + 10_000_000)               //หน่วงเวลา
REPEAT WHILE ina[SW_02] == 0                    //วนลูปถ้า SW_02 = low
GetTime(4)                              //อ่านค่าเวลา
waitcnt(cnt + 10_000_000)               //หน่วงเวลา
if  ina[SW_03] == 0                             //ถ้าสวิทซ์4มีค่าเท่ากับlow
outa[Beep] := 1                         //กำหนดให้P16มีสถานะhigh
GetTime(3)                              //อ่านค่าเวลา3
GetID                                   //อ่านค่าจากดิพสวิทซ์
flagSw3 := flag_On                      //ให้ตัวแปร flagSw3 = high
Switch[3] := 3                          //ให้สวิทซ์4 = 3
waitcnt(cnt + 10_000_000)               //หน่วงเวลา
REPEAT WHILE ina[SW_03] == 0                    //วนลูปถ้า SW_02 = low
GetTime(4)                              //อ่านค่าเวลา
waitcnt(cnt + 10_000_000)               //หน่วงเวลา
if  ina[SwResetTime] == 0                               //ถ้าสวิทซ์รเซตเวลามีค่าเท่ากับ low
outa[Beep] := 1                                 //กำหนดให้P16มีสถานะhigh
```

```
PCF8583Obj.setTime($00,$00,$00,$00)          //Set Time 00:00:00.00
waitcnt(cnt + 10_000_000)                    //หน่วงเวลา
PUB transmit_cmd(i)                          //โปรแกรมย่อยส่งโทนข้อมูล
dira[LED_Red ] := 1                          // ให้P2มีเป็นเอาท์พุต
mdm.transmit                 //เรียกใช้ฟังก์ชั่น transmit ในbell202_modem
outa[LED_Red]:= 0                            // ให้P2มีค่าเท่ากับ low
waitcnt(cnt + 500_000)                       //หน่วงเวลา
mdm.outstr(num.decf(Id_No, 2))               //เรียกใช้ฟังก์ชั่นoutstrและdecf
mdm.outstr(string(","))                      //เรียกใช้ฟังก์ชั่นoutstr
mdm.outstr(num.decf(Switch[i], 1))           //เรียกใช้ฟังก์ชั่นoutstrและdecf
mdm.outstr(string(","))                      //เรียกใช้ฟังก์ชั่นoutstr
mdm.outstr(num.decf(Hours[i], 2))            //เรียกใช้ฟังก์ชั่นoutstrและdecf
mdm.outstr(string(":"))                      //เรียกใช้ฟังก์ชั่นoutstr
mdm.outstr(num.decf(Minutes[i], 2))          //เรียกใช้ฟังก์ชั่นoutstrและdecf
mdm.outstr(string(":"))                      //เรียกใช้ฟังก์ชั่นoutstr
mdm.outstr(num.decf(Seconds[i], 2))          //เรียกใช้ฟังก์ชั่นoutstrและdecf
mdm.outstr(string("."))                      //เรียกใช้ฟังก์ชั่นoutstr
mdm.outstr(num.decf(Seconds100[i], 2))       //เรียกใช้ฟังก์ชั่นoutstrและdecf
mdm.outstr(string(10,13))                    //เรียกใช้ฟังก์ชั่นoutstr
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
mdm.standby                                  //เรียกใช้ฟังก์ชั่นstandby
outa[LED_Red]:= 1                                  // ให้P2มีค่าเท่ากับ high
PUB GetTime(i)                                     //โปรแกรมย่อยอ่านค่าเวลา
Hours[i]     := PCF8583Obj.getHours                //อ่านค่าชั่วโมง
Minutes[i]   := PCF8583Obj.getMinutes          //อ่านค่านาที
Seconds[i]   := PCF8583Obj.getSeconds          //อ่านค่าวินาที
Seconds100[i] := PCF8583Obj.getSeconds100      //อ่านค่ามิลลิเซก
PUB GetID                                          //โปรแกรมย่อยอ่านค่าID
Digit0 := 0                                        //กำหนดให้ Digit0 = 0
Digit1 := 0                                        //กำหนดให้ Digit1 = 0
Digit2 := 0                                        //กำหนดให้ Digit2 = 0
Digit3 := 0                                        //กำหนดให้ Digit3 = 0
if ina[18] == 0                                    // ถ้า P18 = low
Digit0 := 1                                    // ให้ Digit0 = 1
if ina[19] == 0                                    // ถ้า P19 = low
Digit1 := 2                                    // ให้ Digit1 = 2
if ina[20] == 0                                    // ถ้า P20 = low
Digit2 := 4                                    // ให้ Digit2 = 4
if ina[21] == 0                                    // ถ้า P21 = low
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
Digit3 := 8                                    // ให้ Digit3 = 8
Id_No :=  Digit0+Digit1+Digit2+Digit3          //คำนวณค่า Id_No
PUB LCD_Disp                                   //โปรแกรมย่อยแอลซีดีดิสเพลย์
LCD.usDelay( 2_000_000 )                       // time enough to switch to PST
LCD.Init( E, RS, RW, DBHigh, DBLow)            //กำหนดค่าเริ่มต้นให้แอลซีดี
LCD.Clear                                      // เคลียร์หน้าจอแอลซีดี
LCD.Clear                                      // เคลียร์หน้าจอแอลซีดี
LCD.RawSetPos( $00 )                           // กำหนดตำแหน่งที่จะแสดงผล
LCD.PrintStr( string("Rowing Time Ctrl") )     //แสดงข้อความ
LCD.usDelay( 2_000_000 )                       //หน่วงเวลา
GetID                                          //เรียกโปรแกรมย่อยGetID
LCD.RawSetPos( $41 )                           // กำหนดตำแหน่งที่จะแสดงผล
LCD.PrintStr( string("ID.Device : ") )         //แสดงข้อความ
LCD.PrintStr(num.decx(Id_No,2))        //แสดงข้อความค่าIDที่รับจากโปรแกรมย่อยGetID
LCD.usDelay( 4_000_000 )                       //หน่วงเวลา
LCD.Clear                                      // เคลียร์หน้าจอแอลซีดี
Repeat                                 //วนลูปไม่รู้จบ
ShowRealTime                           //เรียกโปรแกรมย่อย ShowRealTime
PUB  Show_LCD(i)                               //โปรแกรมย่อย Show_LCD
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
LCD.PrintStr(num.decx(i+1,1))                    //แสดงข้อความสวิทซ์ตัวที่
LCD.PrintChr(",")                                //แสดงข้อความ  ","
LCD.PrintChr(" ")                                //เว้นช่องไฟ
LCD.PrintStr(num.decx(Hours[i],2))               //แสดงข้อความหลักชั่วโมงที่อ่านได้
LCD.PrintStr(string(":"))                        //แสดงข้อความ  ":"
LCD.PrintStr(num.decx(Minutes[i],2))             //แสดงข้อความหลักนาทีที่อ่านได้
LCD.PrintStr(string(":"))                        //แสดงข้อความ  ":"
LCD.PrintStr(num.decx(Seconds[i],2))             //แสดงข้อความหลักวินาทีที่อ่านได้
LCD.PrintStr(string("."))                        //แสดงข้อความ  "."
LCD.PrintStr(num.decx(Seconds100[i],2))          //แสดงข้อความหลักมิลลิเซกที่อ่านได้
LCD.PrintChr(" ")                                //เว้นช่องไฟ
LCD.PrintChr(" ")                                //เว้นช่องไฟ
LCD.PrintChr(" ")                                //เว้นช่องไฟ


PUB ShowRealTime                                 //โปรแกรมย่อย
ShowRealTime
new := Seconds[4]                                //กำหนดค่าตัวแปร  new
if ode <> new                                    //เปรียบเทียบตัวแปร
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
dira[LED_Orange] := 1                          //กำหนดP3เป็นขาเอาท์พุต
!outa[LED_Orange]                              //สลับสถานะ P3
ode := new                                     //กำหนดค่าให้ตัวแปร
LCD.RawSetPos($41)                  // กำหนดตำแหน่งที่จะแสดงผล
Show_LCD(0)                         //เรียกใช้งานโปรแกรมย่อย Show_LCD
LCD.RawSetPos($04)                  // กำหนดตำแหน่งที่จะแสดงผล
LCD.PrintStr(num.decx(Hours[4],2))  //แสดงข้อความหลักชั่วโมงที่อ่านได้
LCD.RawSetPos($06)                  // กำหนดตำแหน่งที่จะแสดงผล
LCD.PrintStr(string(":"))                      //แสดงข้อความ  ":"
LCD.RawSetPos($07)                             // กำหนดตำแหน่งที่จะแสดงผล
LCD.PrintStr(num.decx(Minutes[4],2))           //แสดงข้อความหลักนาทีที่อ่านได้
LCD.RawSetPos($09)                             // กำหนดตำแหน่งที่จะแสดงผล
LCD.PrintStr(string(":"))                      //แสดงข้อความ  ":"
LCD.RawSetPos($0A)                             // กำหนดตำแหน่งที่จะแสดงผล
LCD.PrintStr(num.decx(Seconds[4],2))           //แสดงข้อความหลักวินาทีที่อ่านได้
LCD.PrintChr(" ")                              //เว้นช่องไฟ
LCD.PrintChr(" ")                              //เว้นช่องไฟ
LCD.PrintChr(" ")                              //เว้นช่องไฟ
LCD.PrintChr(" ")                              //เว้นช่องไฟ
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

Code โปรแกรมแสดงผลด้วย VISUAL C# 2010 EXPRESS

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.Threading;
using System.Drawing.Imaging;
using AForge.Video;
using AForge.Video.DirectShow;
using System.IO;
namespace FSK01
{
public partial class FormMain : Form
{
public string length01;                    ////////////////////
public string length02;                    ////////////////////

private TimeSpan startTime;
//1
private TimeSpan b1_point500Time;
private TimeSpan b1_point1000Time;
//2
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
private TimeSpan b2_point500Time;
private TimeSpan b2_point1000Time;

//3
private TimeSpan b3_point500Time;
private TimeSpan b3_point1000Time;
//4
private TimeSpan b4_point500Time;
private TimeSpan b4_point1000Time;
private string FileName = "";
private string thisTitle;
private string Title;
private string boat1;
private string boat2;
private string boat3;
private string boat4;
private DateTime StartDateTime;
private bool endCount1 = false;
private bool endCount2 = false;
private bool endCount3 = false;
private bool endCount4 = false;
private System.Drawing.Font printFont;
public FormMain()
{
InitializeComponent();
CheckForIllegalCrossThreadCalls = false;
thisTitle = this.Text;
length01 = "";                                    ////////////////////
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
length02 = "";                                    ////////////////////

}
private void Form1_Load(object sender, EventArgs e)
{
String[] strPortNames = SerialPort.GetPortNames();
foreach (String n in strPortNames)
{
comboBox1.Items.Add(n);
}
{
comboBox1.SelectedIndex = 0;
}
Form1 f1 = new Form1();
f1.ShowDialog();
Title = f1.Title;
boat1 = f1.boat1;
boat2 = f1.boat2;
boat3 = f1.boat3;
boat4 = f1.boat4;
this.Text = thisTitle + " [ แข่งขันประเภท " + Title + " ]";
StartDateTime = DateTime.Now;
}
private void button1_Click(object sender, EventArgs e)
{
if (serialPort1.IsOpen)
{
serialPort1.Close();
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
}
serialPort1.PortName=comboBox1.SelectedItem+"";
serialPort1.Open();
}
private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
{
//อ่านค่าจาก serilaport
string readFromSerial = "";
try
{
readFromSerial = serialPort1.ReadLine();
}
catch (Exception)
{
textBox1.AppendText("\n");
}
if (readFromSerial.Equals(string.Empty))
{
return;
}
//แสดงที่ textbox
textBox1.AppendText(readFromSerial);
textBox1.AppendText("\n\n");
//แยกข้อมูล
string[] data = readFromSerial.Split(',');
//แยก ID
string ID = "";
try
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
{
ID = data[0].ToString().Trim();
}
catch (Exception)
{
return;
}
//แยกปุ่ม
string bID = "";
try
{
bID = data[1].ToString().Trim();
}
catch (Exception)
{
return;
}
//แยกเวลา
int h,m,s,ms;
try
{
if (data[2] != null)
{
string[] timedatata = data[2].ToString().Split(':');
h = int.Parse(timedatata[0].ToString().Trim());
m = int.Parse(timedatata[1].ToString().Trim());
s = int.Parse(timedatata[2].ToString().Trim().Split(
ms = int.Parse(timedatata[2].ToString().Trim().Split('.')[1]);
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
if (ID == "0")
{
// เริ่มการทำงาน
StartDateTime = DateTime.Now;
this.Invoke((MethodInvoker)delegate
{
timer1.Enabled = true;
});
endCount1 = false;
endCount2 = false;
endCount3 = false;
endCount4 = false;
this.FileName = @"d:\b-" + DateTime.Now.Hour + DateTime.Now.Minute +
DateTime.Now.Second + ".txt";
if (!File.Exists(this.FileName))
{
// Create a file to write to.
using (StreamWriter sw = File.CreateText(this.FileName))
{
sw.WriteLine("START " +DateTime.Now.ToLongTimeString());
}
}
startTime = new TimeSpan(0, h, m, s, ms*10);
textBox1.AppendText("START\n\n");
}
else if(ID == "1")
{
////////////////////////////////////////////////
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
if (bID == "0")
{
b1_point500Time = new TimeSpan(0, h, m, s, ms);
TimeSpan showtime = b1_point500Time.Subtract(startTime);
ShowTimeUse("#1", showtime, bID, ID);
}
else if (bID == "1")
{
b2_point500Time = new TimeSpan(0, h, m, s, ms);
TimeSpan showtime = b2_point500Time.Subtract(startTime);
ShowTimeUse("#2", showtime, bID, ID);
}
else if (bID == "2")
{
b3_point500Time = new TimeSpan(0, h, m, s, ms);
TimeSpan showtime = b3_point500Time.Subtract(startTime);
ShowTimeUse("#3", showtime, bID, ID);
}
else if (bID == "3")
{
b4_point500Time = new TimeSpan(0, h, m, s, ms);
TimeSpan showtime = b4_point500Time.Subtract(startTime);
ShowTimeUse("#4", showtime, bID, ID);
}
////////////////////////////////////////////////
}
else if (ID == "2")
{
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
//////////////////////////////////////////////
if (bID == "0")
{
b1_point1000Time = new TimeSpan(0, h, m, s, ms);
TimeSpan showtime = b1_point1000Time.Subtract(startTime);
ShowTimeUse("#1", showtime, bID, ID);
endCount1 = true;
}
else if (bID == "1")
{
b2_point1000Time = new TimeSpan(0, h, m, s, ms);
TimeSpan showtime = b2_point1000Time.Subtract(startTime);
ShowTimeUse("#2", showtime, bID, ID);
endCount2 = true;
}
else if (bID == "2")
{
b3_point1000Time = new TimeSpan(0, h, m, s, ms);
TimeSpan showtime = b3_point1000Time.Subtract(startTime);
ShowTimeUse("#3", showtime, bID, ID);
endCount3 = true;
}
else if (bID == "3")
{
b4_point1000Time = new TimeSpan(0, h, m, s, ms);
TimeSpan showtime = b4_point1000Time.Subtract(startTime);
ShowTimeUse("#4", showtime, bID, ID);
endCount4 = true;
```

```
}
////////////////////////////////////////////
}
}
if (endCount1 || endCount2 || endCount3 || endCount4)
{
this.Invoke((MethodInvoker)delegate
{
timer1.Enabled = false;
});
}
}catch(Exception){
textBox1.AppendText("\n");
}
}
private void ShowTimeUse(string msg, TimeSpan showtime, string bID, string ID)
{
string txt = msg + ":: " + showtime.Hours.ToString("00") + ":" + showtime.Minutes.ToString("00") +
":" + showtime.Seconds.ToString("00") + ":" + showtime.Milliseconds.ToString("000");
using (StreamWriter sw = File.AppendText(this.FileName))
{
sw.WriteLine(txt);
}
textBox1.AppendText(txt+"\n\n");
////////////////
if (bID == "0")
{
if (ID == "1")
```

```
{
textBox_b11.Text = txt;
}
else if (ID == "2")
{
textBox_b21.Text = txt;
}
else
{
}
}
if (bID == "1")
{
if (ID == "1")
{
textBox_b12.Text = txt;
}
else if (ID == "2")
{
textBox_b22.Text = txt;
}
else
{
}
}
if (bID == "2")
{
if (ID == "1")
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
{
textBox_b13.Text = txt;
}
else if (ID == "2")
{
textBox_b23.Text = txt;
}
else
{
}
}
if (bID == "3")
{
if (ID == "1")
{
textBox_b14.Text = txt;
}
else if (ID == "2")
{
textBox_b24.Text = txt;
}
else
{
}
}
}
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
}

private void textBox_b21_TextChanged(object sender, EventArgs e)
{
}

private void groupBox2_Enter(object sender, EventArgs e)
{
}
private void button5_Click(object sender, EventArgs e)
{
//
textBox_b11.Text = "";
textBox_b21.Text = "";
textBox_b12.Text = "";
textBox_b22.Text = "";
textBox_b13.Text = "";
textBox_b23.Text = "";
textBox_b14.Text = "";
textBox_b24.Text = "";
textBox2.Text = "";                                    ////////////////////////////////
textBox2.Text = "";                                    ////////////////////////////////
FormMain f2 = new FormMain();                                  ////////////////////////////////
f2.ShowDialog();                                       ////////////////////////////////
length01 = f2.length01;                                 ////////////////////////////////
length02 = f2.length02;                                 ////////////////////////////////
//open detail form
Form1 f1 = new Form1();
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
f1.ShowDialog();
Title = f1.Title;
boat1 = f1.boat1;
boat2 = f1.boat2;
boat3 = f1.boat3;
boat4 = f1.boat4;
this.Text = thisTitle + " [ แข่งขันประเภท " + Title + " ]";
timer1.Enabled = false;
labelShowDate.Text = "00:00:00";
}
private void button4_Click(object sender, EventArgs e)
{
length01 = textBox2.Text;                    ////////////////////////////
length02 = textBox3.Text;                    ////////////////////////////
string printpath = System.Environment.GetFolderPath(Environment.SpecialFolder.Desktop);
printFont = new System.Drawing.Font("Arial", 12);
//printDocument1.Print();
PrintPreviewDialog ppd = new PrintPreviewDialog();
ppd.Document = printDocument1;
ppd.ShowDialog();
}
private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs
e)
{
int fontSize = 16;
float yPos = 0f;
int count = 0;
float leftMargin = e.MarginBounds.Left;
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
float topMargin = e.MarginBounds.Top;
string line = null;
float linesPerPage = e.MarginBounds.Height / printFont.GetHeight(e.Graphics);
//
printFont = new System.Drawing.Font("Arial", 20, FontStyle.Bold);
line = "ประเภท "+ Title;
yPos = topMargin + count * printFont.GetHeight(e.Graphics);
e.Graphics.DrawString(line, printFont, Brushes.Black, leftMargin, yPos, new StringFormat());
count++; count++; count++;
printFont = new System.Drawing.Font("Arial", fontSize, FontStyle.Bold);
//////////////////////
line = "Lane     Name                " + textBox2.Text +"            "+ textBox3.Text;
//////////////////////
yPos = topMargin + count * printFont.GetHeight(e.Graphics);                    //////////////////////
e.Graphics.DrawString(line, printFont, Brushes.Black, leftMargin, yPos, new StringFormat());
//////////////////////
count++;                                                                       //////////////////////
//printFont = new System.Drawing.Font("Arial", fontSize, FontStyle.Bold);
//line = "lane    Name                          500m                1000m";
//yPos = topMargin + count * printFont.GetHeight(e.Graphics);
//e.Graphics.DrawString(line, printFont, Brushes.Black, leftMargin, yPos, new StringFormat());
//count++;
printFont = new System.Drawing.Font("Arial", fontSize, FontStyle.Regular);
line = "  1         " + boat1 + " " + textBox_b11.Text + "" + textBox_b21.Text;
yPos = topMargin + count * printFont.GetHeight(e.Graphics);
e.Graphics.DrawString(line, printFont, Brushes.Black, leftMargin, yPos, new StringFormat());
count++;
line = "  2         " + boat2 + " " + textBox_b12.Text + "" + textBox_b22.Text;
```

```
yPos = topMargin + count * printFont.GetHeight(e.Graphics);
e.Graphics.DrawString(line, printFont, Brushes.Black, leftMargin, yPos, new StringFormat());
count++;
line = "  3.        " + boat3 + " " + textBox_b13.Text + "" + textBox_b23.Text;
yPos = topMargin + count * printFont.GetHeight(e.Graphics);
e.Graphics.DrawString(line, printFont, Brushes.Black, leftMargin, yPos, new StringFormat());
count++;
line = "  4         " + boat4 + " " + textBox_b14.Text + "" + textBox_b24.Text;
yPos = topMargin + count * printFont.GetHeight(e.Graphics);
e.Graphics.DrawString(line, printFont, Brushes.Black, leftMargin, yPos, new StringFormat());
count++;
}
private void timer1_Tick(object sender, EventArgs e)
{
TimeSpan dt = DateTime.Now.Subtract(StartDateTime);
labelShowDate.Text = dt.Minutes.ToString("00") + ":" + dt.Seconds.ToString("00") + ":" +
String.Format( "{0:00}", dt.Milliseconds/10);
}
private void button2_Click(object sender, EventArgs e)
{
StartDateTime = DateTime.Now;
timer1.Enabled = true;
}
private void button3_Click(object sender, EventArgs e)
{
timer1.Enabled = false;
}
}
```

```
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace FSK01
{
public partial class Form1 : Form
{
public string Title;
public string boat1;
public string boat2;
public string boat3;
public string boat4;
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

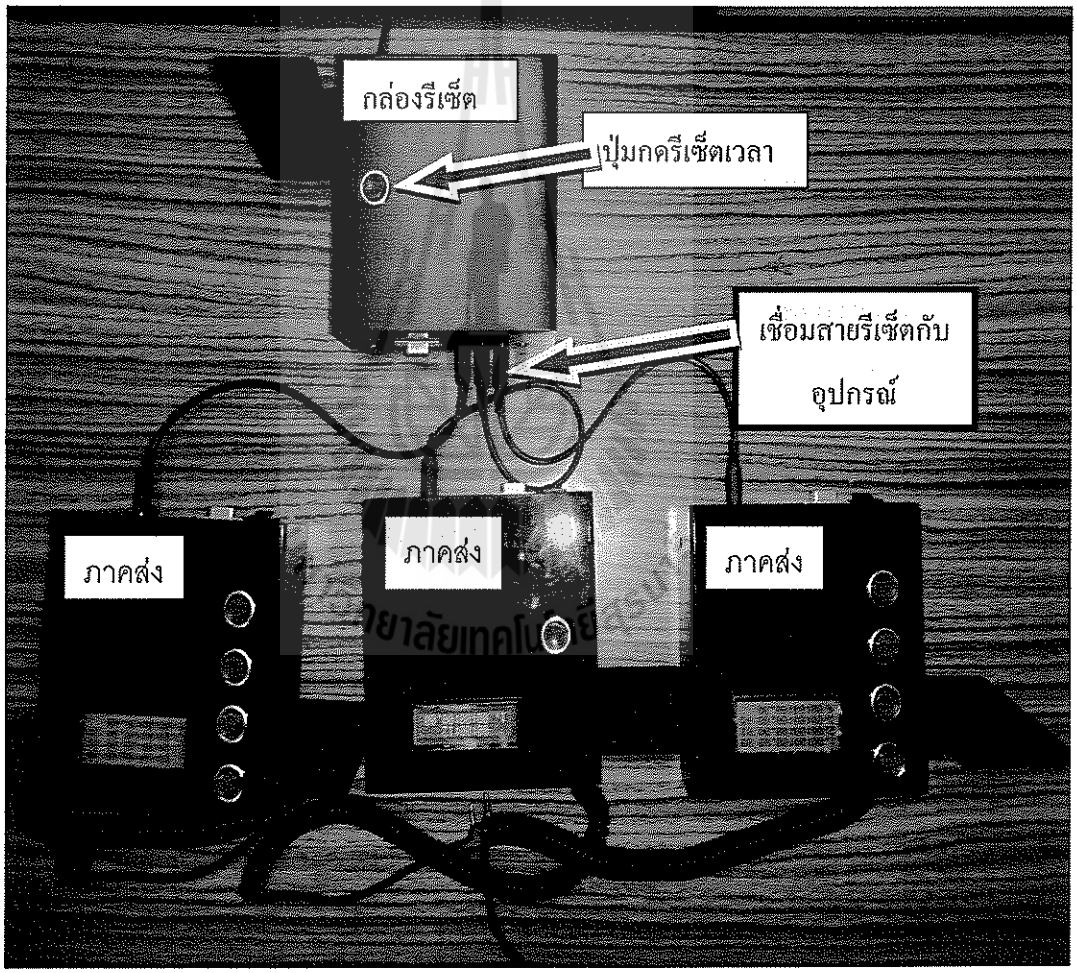
```
public Form1()
{
InitializeComponent();
Title = "";
boat1 = "";
boat2 = "";
boat3 = "";
boat4 = "";
}

private void button1_Click(object sender, EventArgs e)
{
Title = textBox1.Text;
boat1 = textBox2.Text;
boat2 = textBox3.Text;
boat3 = textBox4.Text;
boat4 = textBox5.Text;
this.Close();
}
private void Form1_Load(object sender, EventArgs e)
{
}
}
}
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ข

คู่มือการใช้งาน

1.นำชุดบอร์ดตัวส่งสามชุดมารีเซ็ตเวลาของสัญญาณนาฬิกาจริงเพื่อให้เวลาของบอร์ดทั้งสามเริ่มนับพร้อมกัน โดย ต่อสายรีเซ็ตของกล่องรีเซ็ตเข้ากับบอร์ดทั้งสามตัวและเปิดสวิทช์จ่ายไฟเลี้ยงให้บอร์ดทั้งสาม แล้วกดปุ่มรีเซ็ตหนึ่งครั้ง



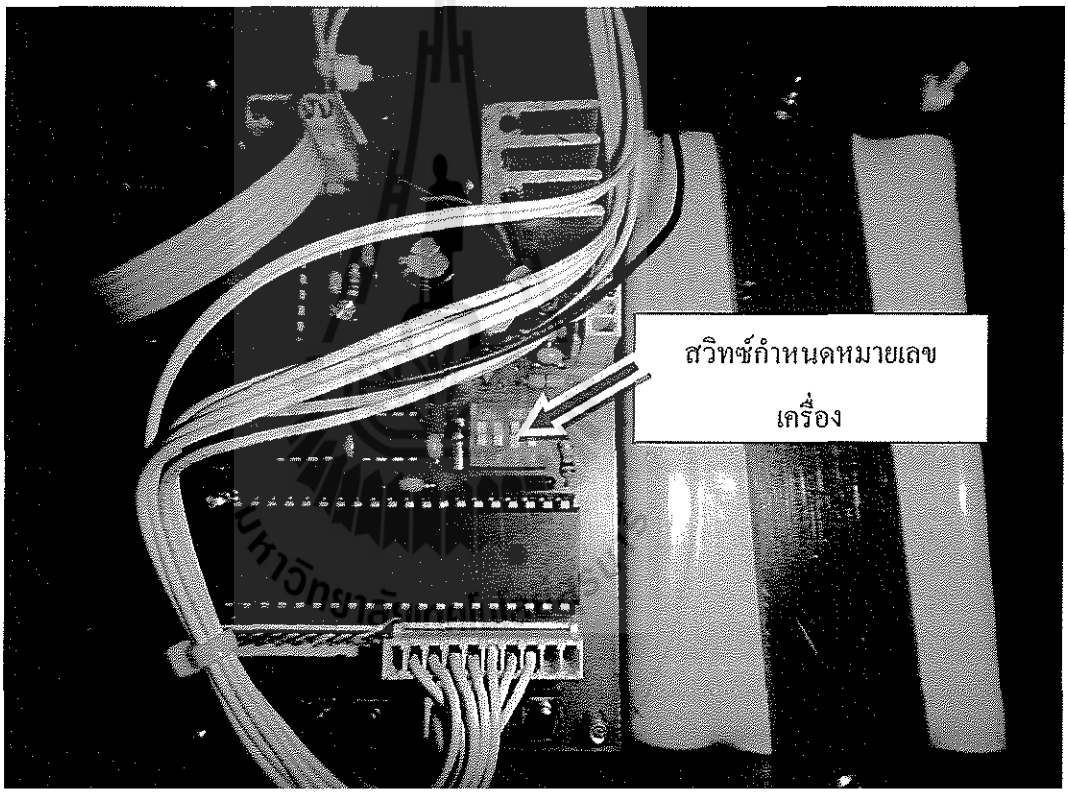
2.นำบอร์ดทั้งสามมาเซ็ตตำแหน่งของระยะทาง ว่าจะให้แต่ละบอร์ดอยู่ที่ตำแหน่งไหน (จุดสตาร์ท จุด 500 เมตร จุดเส้นชัย) โดยการกำหนดเป็นเลขฐานสอง ซึ่งทำได้โดยการเลื่อนสวิทซ์

กำหนดบิตเป็น 00 คือจุดสตาร์ท

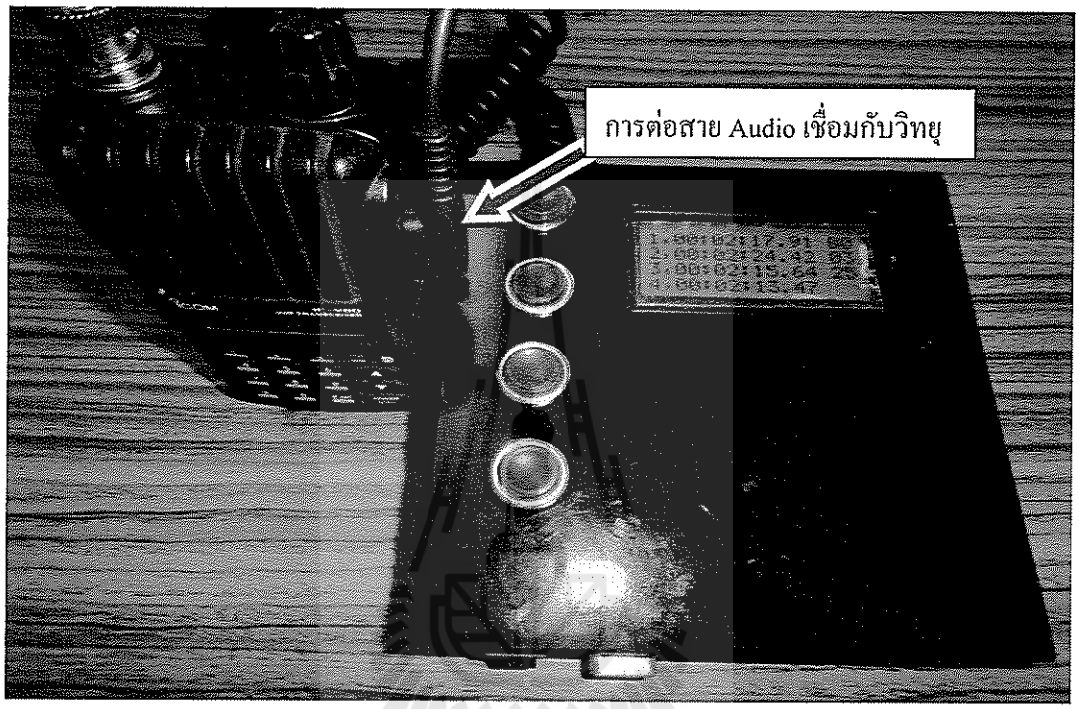
กำหนดบิตเป็น 01 คือจุด 500 เมตร

กำหนดบิตเป็น 02 คือจุดเส้นชัย

ซึ่งกำหนดได้จากการตั้งค่าที่ดิฟสวิทช์ตามหลักการทางตัวเลขฐานสองที่เป็นข้อมูลดิจิตอล

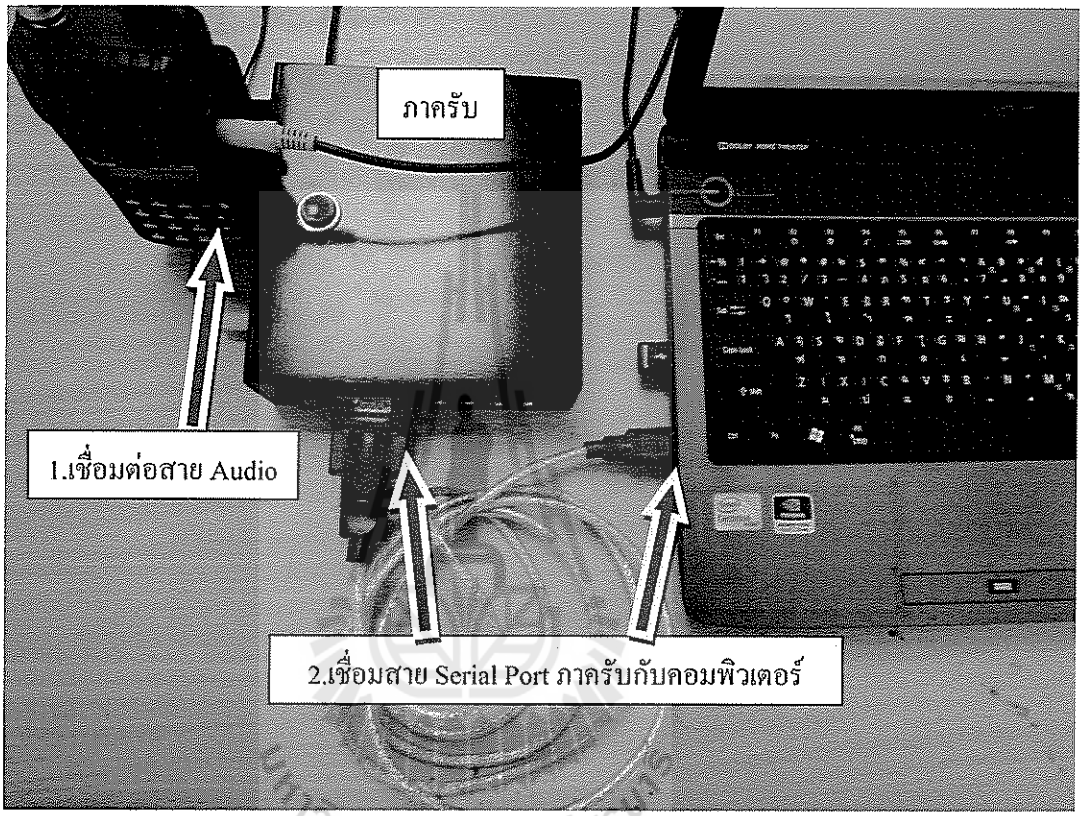


3.นำสาย Audio ของวิทยุต่อเข้ากับบอร์ดภาคส่งทั้งสามชุดดังภาพการต่อแต่ล่ะครั้งต้องปิดวิทยุก่อนเสียบสาย Audio ทุกครั้ง เมื่อต่อเสร็จจึงเปิดเครื่องวิทยุสื่อสาร



มหาวิทยาลัยเทคโนโลยีสุรนารี

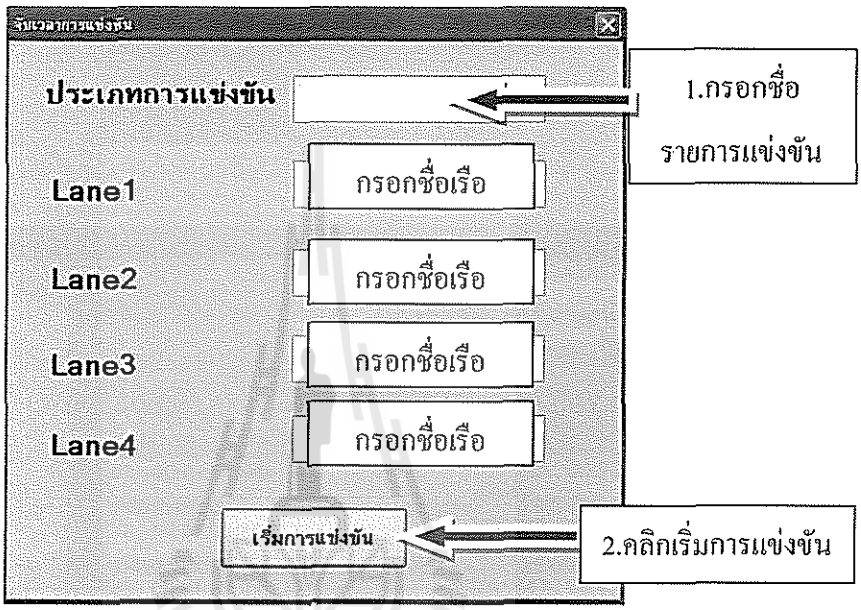
4. นำบอร์ดภาครับต่อกับวิทยุ แล้วปรับระดับความดังของเครื่องวิทยุให้อยู่ในระดับสูงสุดและนำสาย Serial Port ต่อเข้ากับคอมพิวเตอร์ดังภาพ



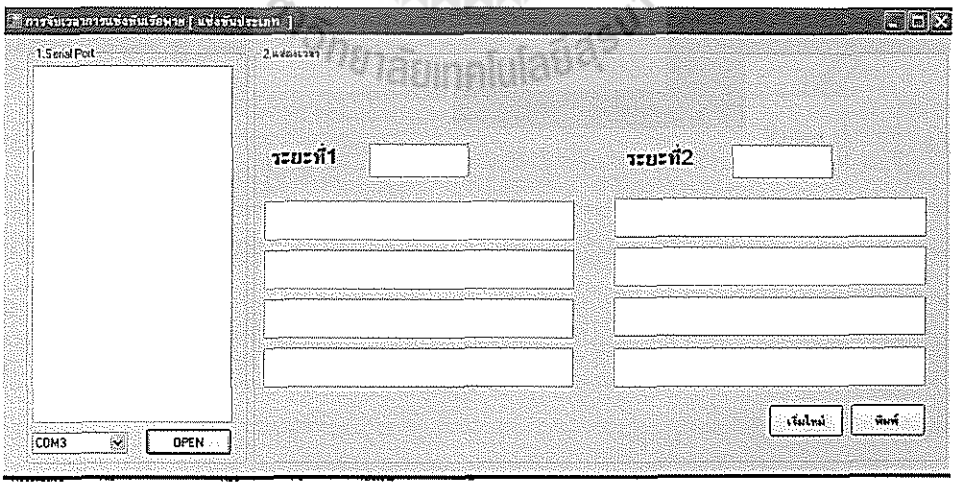
ภาครับสามารถใช้ไฟจากเครื่องชาร์ตแบตเตอร์รี่โดยตรงและจากแบตเตอร์รี่ที่มีในกล่อง ซึ่งการต่อใช้งานแต่ละครั้ง อุปกรณ์ต้องปิดสวิทซ์ทุกครั้ง เมื่อต่อเสร็จจึงเปิดสวิทซ์ใช้งาน ในที่นี้ภาครับได้ใช้เป็นตัวอุปกรณ์ไว้รีเซ็ตเวลาภาคส่งด้วย เมื่อนำไปใช้รีเซ็ตอุปกรณ์ภาคส่งแล้วจึงนำมาใช้เป็นอุปกรณ์ภาครับ โดยมีการต่อใช้งานดังรูป

5. ตัวอย่างการใช้งานโปรแกรมจับเวลาการแข่งขันเรือพาย

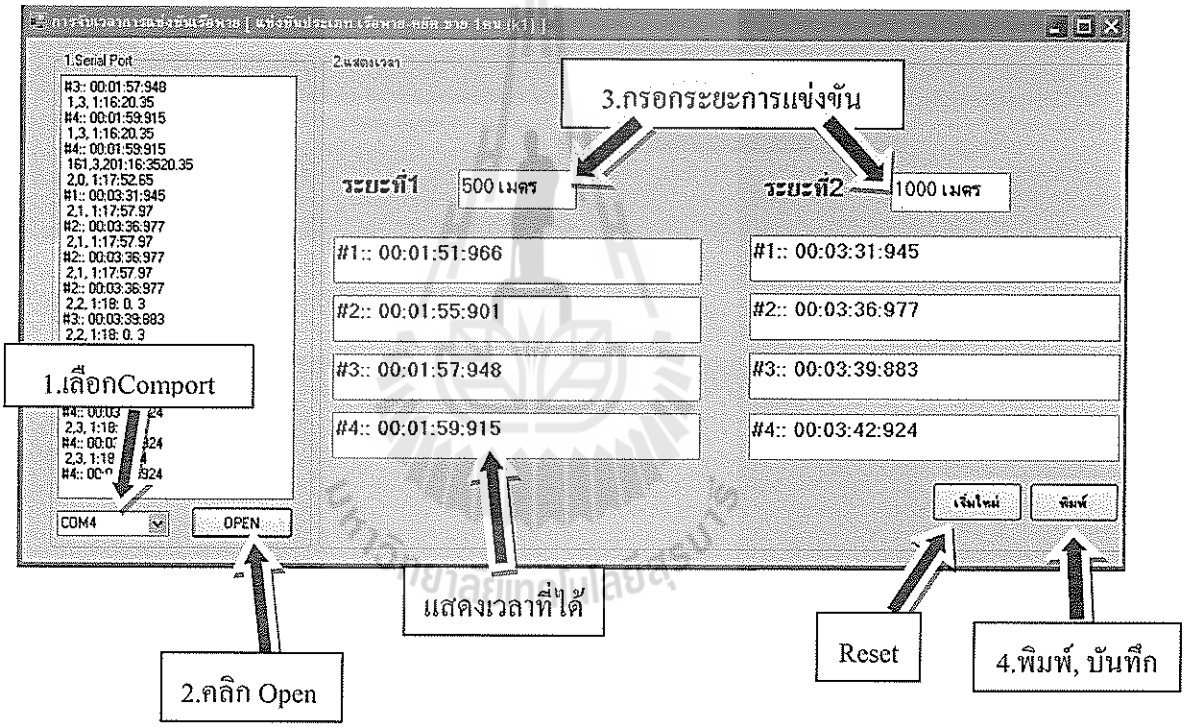
เมื่อลงโปรแกรมแล้ว เปิดโปรแกรมจับเวลา แล้วกรอกชื่อรายการการแข่งขัน กรอกชื่อเรือ และคลิกเริ่มต้นการแข่งขัน



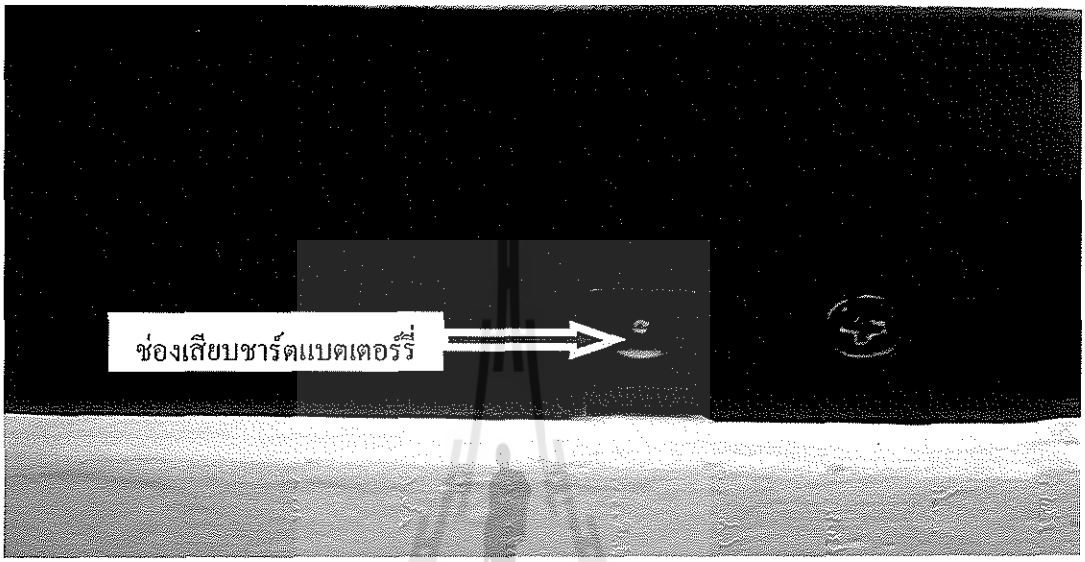
เมื่อคลิกที่เริ่มการแข่งขันแล้ว โปรแกรมจะปรากฏดังรูปเพื่อเริ่มจับเวลาการแข่งขัน



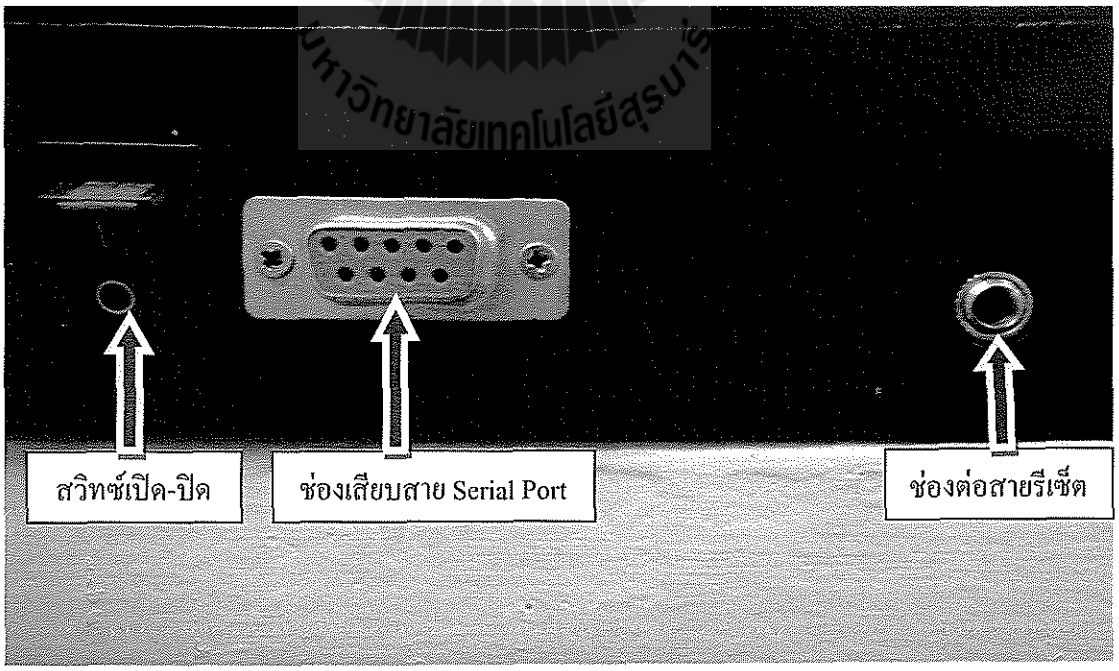
6. เมื่อมีการแข่งขันเกิดขึ้นโปแกรมจะแสดงข้อมูลการแข่งขันดังภาพ และในภาพนี้จะเป็นการบอกถึงการแข่งขันในระยะ1000 เมตร โดยจะบอกถึงตำแหน่งของเรือระยะทางและเวลา โดยมีการแสดงเวลาชั่วโมง นาที วินาที มิลลิวินาที เมื่อข้อมูลถูกส่งมาแล้วข้อมูลเวลาที่ได้จะถูกนำไปเก็บไว้ที่ Drive C ของเครื่องคอมพิวเตอร์ที่ใช้แสดงผล ซึ่งสามารถนำข้อมูลนั้นออกมาพิมพ์เป็น PDF ไฟล์ เพื่อใช้เป็นหลักฐานในการแข่งขันได้



7.การชาร์ตแบตเตอร์รี่



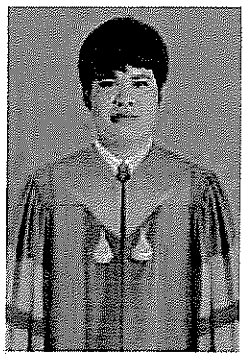
9.ช่องที่ใช้เชื่อมต่อกับอุปกรณ์ภายนอกและสวิทซ์เปิด-ปิด



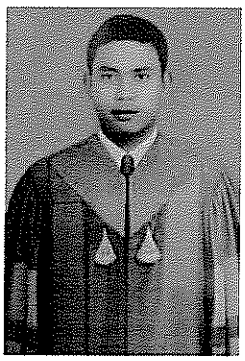
10. การใช้งานอุปกรณ์จับเวลาในการแข่งขัน สำหรับการใช้งานอุปกรณ์จะมีภาคส่งที่จุดสตาร์ท ทำหน้าที่ส่งเวลามาที่ภาครับเมื่อทำการกดปุ่มสตาร์ทขึ้นโดยเวลาจะแสดงที่โปรแกรม และตำแหน่งต่างๆ นั้นจะส่งเวลามาเมื่อเรือวิ่งผ่านโดยกรรมการที่จุดนั้นจะกดปุ่มตามลู่ที่เรือแต่ล่ะลำวิ่งเข้าตามลำดับ เมื่อกรรมการกดปุ่มแล้วจะถูกส่งมาที่ภาครับซึ่งโปรแกรมจะทำการนำเวลาที่จุดต่างๆมาลบจากเวลาที่จุดสตาร์ท จะทำให้ได้เวลาที่ใช้ไปในจุดนั้นของเรือแต่ละลำออกมา และเวลาที่กดนั้นจะแสดงที่จอ LCD ในตัวอุปกรณ์เพื่อป้องกันในกรณีที่เวลาส่งไปไม่ถึงภาครับ ซึ่งสามารถเช็คเวลาได้ด้วยจอ LCD ว่าส่งเวลามาช่วงไหนบ้าง

11. ก่อนการแข่งขันทุกครั้งต้องทำการตรวจสอบอุปกรณ์และระยะการแข่งขันก่อนการแข่งขันทุกครั้ง การที่จะรับส่งข้อมูลได้ใกล้หรือไกล นั้นขึ้นอยู่กับระยะทางและชนิดของสายอากาศ และควรทำการทดสอบระบบก่อนการแข่งขันจริงทุกครั้ง

มหาวิทยาลัยเทคโนโลยีสุรนารี



นายชวลิต จินดาประทุม
เกิดเมื่อวันที่ 31 มีนาคม พ.ศ. 2530
ภูมิลำเนาอยู่ที่ ตำบลบ้านใหม่ไชยมงคล อำเภอทุ่งเสลี่ยม จังหวัดสุโขทัย
สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนชัยมงคลพิทยา
อำเภอทุ่งเสลี่ยม จังหวัดสุโขทัย เมื่อปี พ.ศ. 2549
ปัจจุบันเป็นนักศึกษาชั้นปีที่ 6 สาขาวิชาวิศวกรรมโทรคมนาคม
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี



นายวชิระ คมกล้า
เกิดเมื่อวันที่ 17 สิงหาคม พ.ศ. 2530
ภูมิลำเนาอยู่ที่ ตำบลลำพญากลาง อำเภอมวกเหล็ก จังหวัดสระบุรี
สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนวังม่วงวิทยาคม
อำเภอวังม่วง จังหวัดสระบุรี เมื่อปี พ.ศ. 2549
ปัจจุบันเป็นนักศึกษาชั้นปีที่ 5 สาขาวิชาวิศวกรรมโทรคมนาคม
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

นายวิชิต บุญมาก
เกิดเมื่อวันที่ 11 กุมภาพันธ์ พ.ศ. 2530
ภูมิลำเนาอยู่ที่ ตำบลกาเกาะ อำเภอเมือง จังหวัดสุรินทร์
สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนวีรวัฒน์โยธิน
อำเภอเมือง จังหวัดสุรินทร์ เมื่อปี พ.ศ. 2549
ปัจจุบันเป็นนักศึกษาชั้นปีที่ 5 สาขาวิชาวิศวกรรมโทรคมนาคม
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

บรรณานุกรม

[1] กฤษดา ใจเย็นและชัยวัฒน์ ลิ้มพรจิตรวิไล (ม.ป.ป.).
ก้าวแรกกับ Propeller มัลติคอร์ไมโครคอนโทรเลอร์,บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด.

[2] สมาคมเรือพายแห่งประเทศไทย
ข้อมูลจาก URL : http://www.rcat.or.th

[3] สมศักดิ์ วาณิชอนันต์ชัย, ผศ.ดร. เอกสารการสอนวิชา 427314 Digital Communication
นครราชสีมา : มหาวิทยาลัยเทคโนโลยีสุรนารี,2553

[4] บริษัทอีทีที คู่มือการใช้งานบอร์ด ET-BASE PX32 V1.0 [On line] จาก:
http://www.etteam.com/product/bs/man_ET-BASE_P8X32A_V1.pdf

[5] Harprit Singh Sandhu.
Programming the Propeller™ with Spin™: A Beginner's Guide to Parallel Processing.
The McGraw Hill Companies, 2010

มหาวิทยาลัยเทคโนโลยีสุรนารี