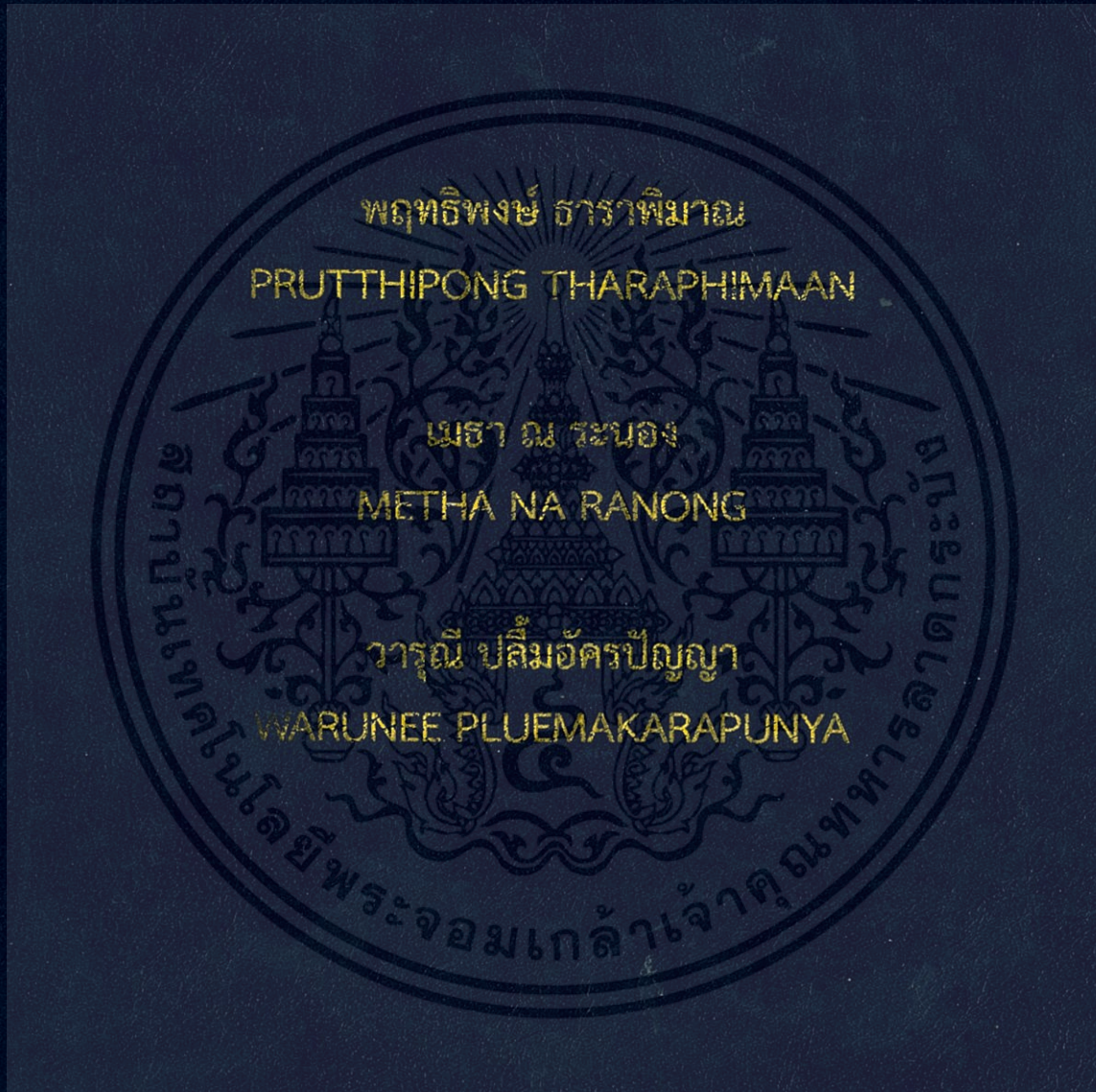


หุ่นยนต์อลวนบนอินเทอร์เน็ต
Chaotic Robot On The Internet



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2555

หุ่นยนต์อลวนบนอินเทอร์เน็ต
Chaotic Robot On The Internet



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Chaotic Robot On The Internet



THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2012

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญาบัตร
รายชื่อนักศึกษา

หุ่นยนต์อลวนบนอินเทอร์เน็ต
นายพฤทธิพงษ์ ธาราพิมาณ
นายเมธา ณ ระนอง
นางสาววารุณี ปลื้มอักษรปัญญา

รหัสประจำตัว 52010796
รหัสประจำตัว 52010956
รหัสประจำตัว 52011106

ปริญญา
สาขาวิชา
พ.ศ.

วิศวกรรมศาสตร์บัณฑิต
วิศวกรรมสารสนเทศ
2555

อาจารย์ที่ปรึกษาปริญญาบัตร

ผศ.ดร. กฤตากร กล่อมการ
ผศ.ไพศาล สิทธิโยภาสกุล

ปริญญาบัตรฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง




.....
(ผศ.ดร. กฤตากร กล่อมการ)


.....
(ผศ.ดร. ไพศาล สิทธิโยภาสกุล)

อาจารย์ผู้ควบคุมปริญญาบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

หุ่นยนต์อลวนบนอินเทอร์เน็ต

รายชื่อนักศึกษา

นายพฤทธิพงษ์ ธาราพิมาณ

รหัสประจำตัว 52010796

นายเมธา ณ ระนอง

รหัสประจำตัว 52010956

นางสาววารุณี ปลื้มอัครปัญญา

รหัสประจำตัว 52011106

ปริญญา

วิศวกรรมศาสตรบัณฑิต

สาขาวิชา

วิศวกรรมสารสนเทศ

พ.ศ.

2555

อาจารย์ที่ปรึกษาปริญญานิพนธ์

ผศ.ดร. กฤตากร กล่อมการ

ผศ.ไพศาล สิทธิโยภาสกุล

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอ การขับเคลื่อนหุ่นยนต์โดยนำเอาสัญญาณอลวน (Chaotic) มาประยุกต์ใช้เพื่อควบคุมวงโคจรการเคลื่อนที่ของหุ่นยนต์โดยมีการบังคับหุ่นยนต์ผ่าน Web base เพื่อใช้ในการศึกษาสัญญาณอลวน ของนักศึกษาชั้นปีที่ 1 ในวิชา Fundamentals electronics Laboratory เพราะในปัจจุบันนี้นักศึกษาหลายคนยังคงไม่เข้าใจถึงสิ่งที่เรียนไปว่า วงจรอลวนนั้นใช้งานด้านสารสนเทศประยุกต์ได้อย่างไรบ้าง

ด้วยเหตุนี้ทางเราจึงคิดว่าเราจะสามารถช่วยให้นักศึกษาชั้นปี 1 เข้าใจถึงประโยชน์ในการเรียนวิชานี้และเข้าใจว่านำไปใช้ในทางสารสนเทศได้อย่างไร นอกจากนี้แล้วระบบที่จะจัดทำขึ้นนั้น ยังเป็นไปในทิศทางเดียวกับการศึกษาในโลกยุคปัจจุบัน คือสามารถเพิ่มการเรียนรู้นอกห้องเรียนให้นักศึกษาได้ศึกษาหาความรู้ได้ตลอดเวลา และได้ด้วยตนเองอีกด้วย

Thesis Title	Chaostic Robot On The Internet	
Student	Mr. Pruetthipong tharaphimaan	Student ID. 52010796
	Mr. Metha na ranong	Student ID. 52010956
	Mrs. Warunee Pluemakarapunya	Student ID. 52011106
Degree	Bachelor of Engineering	
Program	Information Engineering	
Year	2012	
Thesis Advisor	Assoc.Prof.Dr.KitdakornKlomkarn	
	Assoc.Prof.Paisan Sithiyopasakul	

Abstract

The aim of this Bachelor's thesis is to adapt the chaotic signal into the robot in order to control the movement of the robot by using the web-based. This thesis will be used to examine the chaotic signal in a subject Fundamentals Electronics Laboratory to show how the chaotic signal works and how could we adapt the information technology into it. The result of this thesis would help in understanding the chaotic signal laboratory for the student and the example of adapting the information technology in laboratory. Besides, this thesis support the view that the students could study anywhere they wanted to and improve the self-studying habits for the students.

กิตติกรรมประกาศ

การจัดทำปริญญาบัตรฉบับนี้จะไม่สำเร็จลุล่วงไปด้วยดีได้เลย ถ้าไม่ได้รับการช่วยเหลือจาก ผศ.ดร.กฤตากร กล่อมการ ซึ่งคอยแนะนำทุกอย่างตั้งแต่เริ่มทำงานอย่างเป็นระบบ การระบวนการศึกษาการทำโครงการเบื้องต้น หาความรู้ใหม่ๆ ไม่ใช่เพียงแต่การหาความรู้ในอินเทอร์เน็ต รวมไปถึงหนังสือจากต่างประเทศ เอกสารและตำราต่างๆ รวมถึงวิธีการทำงานเป็นกลุ่มถือเป็นเรื่องที่สำคัญมาก ซึ่งท่านทุ่มเทให้กับนักศึกษาอย่างเต็มที่ คณะผู้จัดทำจึงขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณทางภาควิชาวิศวกรรมสารสนเทศที่ให้ความรู้และสถานที่ที่ใช้ในการทำงานต่างๆ ทางผู้จัดทำได้รับความรู้และมีความสามารถทำให้เกิดความชัดเจนในการทำปริญญาบัตรฉบับนี้มากขึ้น โดยนำความรู้ที่เหล่าคณาจารย์สอนมาประยุกต์ใช้เต็มความสามารถ ขอขอบพระคุณอาจารย์ที่ทุ่มเทการสอนมาตลอดเมื่อพบปัญหาใดใด ถึงแม้ว่าอาจารย์จะไม่สามารถให้คำแนะนำได้แต่อาจารย์ก็สามารถแนะแนวทางอื่นที่เป็นประโยชน์อย่างยิ่ง ทำให้มีแนวทางการแก้ปัญหาเพิ่มมากขึ้น

ขอขอบพระคุณ นายอนุรักษ์ จันศรี นักศึกษาปริญญาเอกที่ได้ให้ความช่วยเหลือในด้านการประดิษฐ์ตัวหุ่นยนต์ว่าควรทำไปในทิศทางใดจึงจะออกมาเหมาะสม ถูกต้อง และใช้งานได้จริง ทำให้ทางคณะผู้จัดทำได้ทำหุ่นยนต์ออกมามีความสวยงามและนำไปใช้งานได้จริงตามจุดประสงค์ที่ต้องการ

ขอขอบพระคุณ นายธัชพงษ์ พงศ์สุทธิยากร นักศึกษาปริญญาโทที่ได้ให้ความช่วยเหลือเกี่ยวกับทฤษฎีอลวน ทำให้ทางคณะผู้จัดทำได้เข้าใจอย่างถ่องแท้ และนำไปปฏิบัติได้อย่างถูกต้อง

ขอขอบพระคุณ นายสถาพร จันทมงคล นักศึกษาปริญญาโทที่ได้ให้ความช่วยเหลือเกี่ยวกับเรื่องการเชื่อมต่อแบบไร้สาย ว่าควรใช้อุปกรณ์อะไร เพราะอะไร หากเกิดปัญหาควรแก้ไขอย่างไร ทำให้ทางคณะผู้จัดทำได้เลือกใช้อุปกรณ์ได้อย่างเหมาะสมในการนำมาใช้งานจริง

ขอขอบคุณเพื่อนๆทุกคนที่อยู่ในภาควิชาเดียวกัน กับความมีน้ำใจที่หยิบยื่นมิตรให้ตลอดมา และคอยช่วยเหลือ แนะนำหรือเสนอแนะเมื่อพบเจอกับปัญหาเสมอมา รวมถึงเพื่อนที่อยู่ภาควิชาอื่นๆที่คอยอธิบายความรู้อื่นๆ ที่ยังไม่เข้าใจให้ได้รับฟัง หรือเนื้อหาอื่นๆ ที่นอกเหนือจากเนื้อหาที่เคยได้เรียนมา ให้เกิดความถ่องแท้ในเรื่องนั้นๆ

และที่สำคัญขอขอบคุณกำลังใจจากครอบครัวของคณะผู้จัดทำทุกคนที่มีมาเสมอไม่เคยขาด ขอขอบคุณ บิดา มารดา และญาติๆในหลายๆเรื่อง รวมไปถึงการสนับสนุนด้านการเงินจวบจนปีสุดท้ายของการเรียนการศึกษาในวิศวกรรมศาสตร์ในสถาบันแห่งนี้

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
Abstract.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VII
สารบัญตาราง.....	XI
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 จุดประสงค์.....	1
1.3 ขอบเขตของโครงการ.....	1
1.4 ผลที่คาดว่าจะได้รับ.....	2
1.5 อุปกรณ์ที่ต้องใช้.....	2
1.6 ขั้นตอนการดำเนินงาน.....	3
บทที่ 2 ทฤษฎีพื้นฐานที่ใช้.....	4
2.1 Chaotic.....	4
2.1.1 นิยามของ Chaotic.....	4
2.1.2 ประวัติและความเป็นมา.....	4
2.1.3 คุณลักษณะของ Chaotic.....	5
2.1.4 ประโยชน์ของ Chaotic.....	6
2.1.5 ทฤษฎีลอเรนซ์.....	7
2.1.6 Lorenz's Equation.....	9
2.1.7 Chen's Equation.....	10
2.1.8 Rossler's Equation.....	10
2.2 ระบบที่เชื่อมต่อกับระบบอินเตอร์เน็ต.....	11
2.2.1 การมอดูเลตสัญญาณ RF.....	12
2.3 โปรแกรม Artisteer.....	12
2.4 การใช้งานโมดูล UART กับพ็อตอนุกรม.....	14
บทที่ 3 การออกแบบและโครงสร้าง.....	15

สารบัญ (ต่อ)

	หน้า
3.1 ภาพรวมของการทำงานทั้งหมด.....	15
3.2 วงจรรวมของหุ่นยนต์.....	16
3.2.1 LM7805.....	17
3.2.2 Microcontroller Dspic30F 2010	18
3.2.3 L298.....	19
3.3 วงจรโดยรวมของหุ่นยนต์เป็นไฟล์ PCB.....	20
3.4 วงจรโดยรวมของหุ่นยนต์หลังจากทำการกัดปรินท์และบัดกรีเสร็จ.....	20
3.5 การออกแบบเว็บไซต์โดยใช้โปรแกรม Artisteer	21
3.5.1 หน้า HOME	21
3.5.2 หน้า CONTACT	22
3.5.3 หน้า OVERVIEW.....	23
3.5.4 หน้า DEFINITION	23
3.5.5 หน้า THEORY.....	24
3.5.6 หน้า DESIGN	28
3.5.7 หน้า EXPERIMENT.....	29
3.6 การทำงานในโปรแกรม Microcontroller	29
บทที่ 4 ผลการทดลอง.....	31
4.1 อุปกรณ์ที่ใช้ในการทดลอง.....	31
4.2 อุปกรณ์ที่ใช้ในวงจรขับเคลื่อน.....	32
4.3 การประกอบระหว่างตัวหุ่นยนต์และวงจรขับเคลื่อน.....	33
4.4 ทดลองเขียนโปรแกรมตามทฤษฎี Chaotic โดยผ่านโปรแกรม MATHLAB.....	33
4.4.1 เขียนโปรแกรมตามทฤษฎีของ Lorenz	33
4.4.2 เขียนโปรแกรมตามทฤษฎีของ Chen	35
4.4.3 เขียนโปรแกรมตามทฤษฎีของ Rossler	37
4.5 การทดลองส่งข้อมูลแบบไร้สาย	39
4.6 การทดลองส่งข้อมูลแบบไร้สาย.....	39
4.7 วงจรการแปลงไฟจาก 5V เป็น 3.3 V.....	41
บทที่ 5 สรุปผลการทดลอง	42
5.1 สรุปผลการทดลอง.....	42
5.2 ปัญหาที่เกิดขึ้นในการทดลอง	42
5.3 แผนปฏิบัติในอนาคต.....	43
เอกสารอ้างอิง	44
ภาคผนวก.....	45

สารบัญ (ต่อ)

	หน้า
ภาคผนวก ก. LM7805	46
ภาคผนวก ข. Dspic30F 2010	50
ภาคผนวก ค. L298.....	53
ภาคผนวก ง. K-USB to serail	58
ภาคผนวก จ. Code.....	60
ภาคผนวก ฉ. การติดต่อสื่อสารของ Xbee เพื่อนำไปใช้งาน.....	78



สารบัญรูป

รูปที่	หน้า
1.1 แสดงไดอะแกรมโดยรวมของโครงการทั้งหมด.....	2
1.2 ภาพแสดงการวางแผนงานของโครงการ.....	3
2.1 สัญญาณเคออสที่สร้างจากสมการ Lorenz.....	8
2.2 สัญญาณเคออสจากสมการ Lorenz ในลักษณะ 3 มิติ.....	9
2.3 สัญญาณเคออสจากสมการ Lorenz ในลักษณะ 2 มิติ ในระนาบต่าง.....	10
2.4 สัญญาณเคออสจากสมการ Rossler ในลักษณะ 3 มิติ.....	11
2.5 สัญญาณเคออสจากสมการ Rossler ในลักษณะ 2 มิติ ในระนาบต่าง.....	11
2.5 หน้าแรกของโปรแกรม Artisteer.....	14
3.1 ภาพรวมของการทำงาน.....	15
3.2 ภาพการออกแบบวงจรรวมของหุ่นยนต์ผ่านโปรแกรม Microsoft visio.....	16
3.3 ภาพวงจรรวมของหุ่นยนต์ผ่านโปรแกรม Protel.....	16
3.4 LM7805.....	17
3.5 LM7805.....	17
3.6 DC Parameters.....	17
3.7 Microcontroller Dspic30F 2010.....	18
3.8 Pin Diagrams ของ Dspic30F 2010.....	18
3.9 L298.....	18
3.10 L298.....	19
3.11 Block Diagram.....	19
3.12 ภาพวงจรรวมของหุ่นยนต์ผ่านโปรแกรม Portel (ไฟล์ PCB).....	20
3.13 ภาพวงจรรวมของหุ่นยนต์ที่ผ่านการกัดปรินต์และบัดกรีเสร็จ.....	20
3.14 หน้า HOME.....	22
3.15 หน้า CONTACT.....	22
3.16 หน้า OVERVIEW.....	23
3.17 หน้า DEFINITION.....	24
3.18 หน้า THEORY.....	25
3.19 หน้า Lorenz.....	26
3.20 หน้า Rossler.....	27
3.21 หน้า Chen.....	27
3.22 หน้า DESIGN.....	28
3.23 หน้า EXPERIMENT.....	29
3.24 Flowchart แสดงการทำงานของโปรแกรมใน Microcontroller.....	30
4.1 ขนาดของหุ่นยนต์.....	31

สารบัญรูป (ต่อ)

หน้า

รูปที่		
4.2	ช่องสำหรับวางแบตเตอรี่.....	31
4.3	รูปวงจรขับมอเตอร์ที่ทำการต่อเสร็จแล้ว.....	32
4.4	รูปวงจรขับมอเตอร์ที่ทำการบัดกรีและต่อเสร็จแล้ว.....	32
4.5	รูปของตัวหุ่นยนต์ที่ต่อกับวงจรขับมอเตอร์.....	33
4.6	รูปของตัวหุ่นยนต์ที่ต่อกับวงจรขับมอเตอร์ที่ทำการบัดกรีแล้ว.....	33
4.7	สัญญาณ Chaos จากสมการ Lorenz ในลักษณะ 3 มิติ.....	34
4.8	สัญญาณเคออสจากสมการ Lorenz ในลักษณะ 2 มิติ ในระนาบต่าง.....	34
4.9	สโคปที่ได้จากการวัด output ของสมการ Lorenz.....	34
4.10	ภาพการเดินของหุ่นยนต์ตามทฤษฎีของ Lorenz ในระนาบ X,Y.....	35
4.11	สัญญาณ Chaos จากสมการ Chen ในลักษณะ 3 มิติ.....	36
4.12	สัญญาณเคออสจากสมการ Chen ในลักษณะ 2 มิติ ในระนาบต่าง.....	36
4.13	สโคปที่ได้จากการวัด output ของสมการ Chen.....	36
4.14	ภาพการเดินของหุ่นยนต์ตามทฤษฎีของ Chen ในระนาบ X,Y.....	37
4.15	สัญญาณ Chaos จากสมการ Rossler ในลักษณะ 3 มิติ.....	38
4.16	สัญญาณเคออสจากสมการ Rossler ในลักษณะ 2 มิติ ในระนาบต่าง.....	38
4.17	สโคปที่ได้จากการวัด output ของสมการ Rossler.....	38
4.18	ภาพการเดินของหุ่นยนต์ตามทฤษฎีของ Rossler ในระนาบ X,Y.....	39
4.19	ภาพของ Module XY-MK-5V ที่ใช้ในการทดลอง.....	40
4.20	ภาพของ Bluetooth ที่ใช้ในการทดลอง.....	40
4.21	ภาพของ XBEE ที่ใช้ในการทดลอง.....	40
4.22	วงจรแปลงไฟจาก 5 V เป็น 3.3 V.....	41
5.1	ตัวอย่างปัญหาที่เกิดขึ้นในการทดลอง.....	43

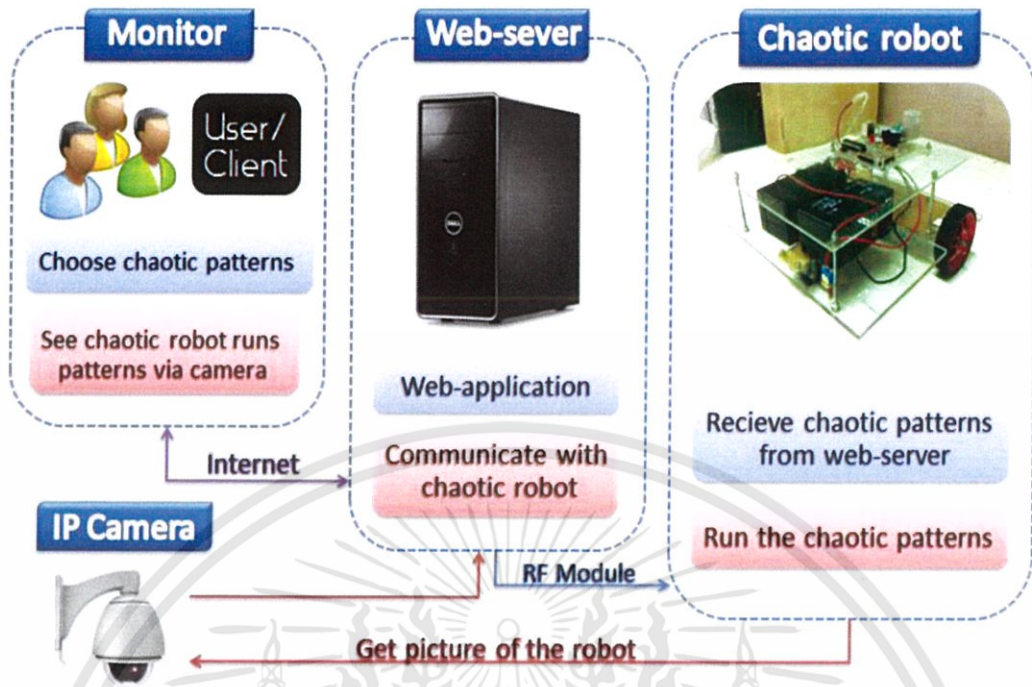
สารบัญตาราง

หน้า

รูปที่

- 2.1 พฤติกรรมต่างๆของระบบเมื่อ $a = 10$, $c = 8/3$ และค่า b มีการเปลี่ยนค่า
ในช่วงต่าง ๆ กันมีผลลัพธ์ในแต่ละช่วงดังต่อไปนี้ 8
- 2.2 ข้อแตกต่างของ 2 เวอร์ชันมีดังนี้ 13





รูปที่ 1.1 แสดงไดอะแกรมโดยรวมของโครงการทั้งหมด

- 1.3.1 มีการรับคำสั่งทาง GUI เพื่อเป็นการเลือกสมการ
- 1.3.2 สมการที่ได้เป็นการกำหนดการเคลื่อนที่ของหุ่นยนต์
- 1.3.3 หุ่นยนต์ตอบสนองโดยการเคลื่อนที่ตามรูปแบบที่ได้รับคำสั่ง
- 1.3.4 มีการแสดงผลเป็นรูปภาพจาก IP Camera ซึ่งเป็นแบบ Streaming ผ่านทาง

Web base

1.4 ผลที่คาดว่าจะได้รับ

- 1.4.1 นักศึกษาชั้นปี 1 ภาควิชาสารสนเทศ เกิดการใฝ่รู้ใฝ่เรียนในวิชา Fundamentals electronics Laboratory มากขึ้น
- 1.4.2 ก่อให้เกิดความรู้ในการศึกษาวงจรไฟฟ้าที่สามารถสร้างสัญญาณอลวน มาใช้แทน Microcontroller เปรียบเสมือนเป็นองค์ความรู้ใหม่ที่นำมาใช้ควบคุมหุ่นยนต์ได้
- 1.4.3 ก่อให้เกิดการเรียนรู้ในห้องเรียนโดยสามารถทำได้ด้วยตนเองมากขึ้น
- 1.4.4 ก่อให้เกิดการเรียนรู้ในห้องเรียนที่สามารถทำได้ทุกเวลามากขึ้น เพียงผ่านระบบ Internet

1.5 อุปกรณ์ที่ต้องใช้

1.5.1 ฮาร์ดแวร์

- แผงวงจรไฟฟ้าที่สร้างสัญญาณอลวน

จำนวน 2 ชิ้น

- IP Camera ที่ใช้ในการสั่งการและแสดงผล จำนวน 2 ตัว
- หุ่นยนต์ จำนวน 1 ตัว
- Computer ที่ใช้ในการสั่งการโดยสั่งการผ่าน Web base จำนวน 1 เครื่อง
- PX-700 จำนวน 1 ตัว
- dspic30F จำนวน 1 ตัว
- K-USB 2 Serail จำนวน 1 ตัว
- RF Module (Xbee) จำนวน 2 ตัว

1.5.2 ซอฟต์แวร์

- MATHLAB
- PICKit2 V2.61
- Protel 99SE
- MPLab IDE
- Hyperterminal
- Artisteer

1.6 ขั้นตอนการดำเนินงาน

ID	Task Name	2012						2013			
		Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar
1	สัมภาษณ์น้องปี 1	█									
2	รวบรวมเอกสาร	█	█								
3	ที่หุ่นยนต์		█	█							
4	ติดตั้ง Server			█	█						
5	เปิด Module Remote			█	█						
6	Test & Debug				█	█					
7	Analysis & Design Web-base						█	█	█		
8	Documentation	█	█	█	█	█	█	█	█	█	█

รูปที่ 1.2 ภาพแสดงการวางแผนงานของโครงการ

บทที่ 2

ทฤษฎีพื้นฐานที่ใช้

2.1 Chaotic

2.1.1 นิยามของ Chaotic

Chaos ตามพจนานุกรมนั้นหมายถึง ความสับสนวุ่นวาย ไร้ระเบียบ แต่ Chaos ที่เรากำลังศึกษานั้นคือการศึกษาเกี่ยวกับเคออสในทางคณิตศาสตร์ หรือ Deterministic Chaos ซึ่งก็ยังคงเค้าของความหมายเดิมและมีความคล้ายคลึงกับในพจนานุกรมอยู่ ส่วนในความหมายทางวิทยาศาสตร์คือ สภาพหรือกระบวนการที่ไม่เสถียรภาพ (Unstable) หากมีการกระทบเพียงเล็กน้อย อาจทำให้เกิดสัญญาณที่ไม่เป็นเส้นตรง แต่เป็นเส้นทางคดเคี้ยววุ่นวายอลวนหรือในบางครั้งอาจเกิดการกระโดดข้ามคั้งนั้นผลลัพธ์ที่เกิดขึ้นจึงไม่สามารถคาดเดาหรือทำนายให้ถูกต้องแม่นยำได้

พฤติกรรมแบบอลวน (Chaotic behavior) เป็นพฤติกรรมที่ดูไร้ระเบียบ ที่ดูเหมือนว่าเป็นปรากฏการณ์ที่เกิดขึ้นอย่างสะเปะสะปะหรือเชิงสุ่มเอา (random) แต่เป็น deterministic ที่สามารถอธิบายได้ด้วยสมการทางคณิตศาสตร์ซึ่งที่จริงแล้วแฝงไปด้วยความเป็นระเบียบ (Order)

ระบบที่ประพุดิตัวแบบ Chaos จะไวต่อการเปลี่ยนแปลงของเงื่อนไขเริ่มต้นเป็นอย่างมาก (sensitivity dependent on initial condition) ทำให้ไม่สามารถทำนายได้ในระยะยาว (long-term unpredictable) โดยความคลาดเคลื่อนเพียงเล็กน้อยจากการวัดเงื่อนไขเริ่มต้นจะเพิ่มขึ้นแบบเอ็กโปเนนเชียล (Exponential) เมื่อเวลาผ่านไปเคออสได้มีการค้นพบมานานแล้วตั้งแต่สมัยของ J. H. Poincare (1854-1912) ซึ่งตอนนั้น Poincare ได้พิสูจน์ว่าระบบสุริยะจักรวาล แบบ Chaos แต่สมัยนั้นยังไม่บัญญัติศัพท์อลวน Chaos ปรากฏการณ์นี้ถูกเรียกว่า Pathological case

และจากหลัก uncertainty principle ของ Heisenberg ทำให้เราไม่สามารถวัดค่าพารามิเตอร์ (parameter) ทุกตัวได้อย่างถูกต้องแม่นยำ เพราะการให้ความแม่นยำอย่างมาก จะส่งผลให้เกิดความไม่แน่นอนในอีกอย่างหนึ่ง และถึงแม้ตัวแปรที่อธิบายระบบของเรามีเพียงตัวแปรเดียว ความแม่นยำของค่าที่วัดได้ก็จะจำกัดอยู่ค่าหนึ่ง ความไม่แน่นอนที่เหลืออยู่จะเพิ่มขึ้นเรื่อยๆ เมื่อเวลาผ่านไป ซึ่งทำให้เลขนัยสำคัญที่เราได้ลดลงเรื่อยๆจนเราทำนายอะไรไม่ได้เลย ดังนั้น chaos จึงเกิดขึ้นได้ทั้งในทฤษฎีและในทางปฏิบัติ

สำหรับระบบที่เป็นเชิงเส้น จะเกิดความคลาดเคลื่อนเมื่อเวลาผ่านไป เป็นสัดส่วนกับความคลาดเคลื่อนเริ่มแรก เนื่องจากความสัมพันธ์เป็นแบบเชิงเส้นพฤติกรรมแบบอลวน (chaotic behavior) จึงไม่มีในระบบที่เป็นเชิงเส้นเคออส เกิดขึ้นได้เฉพาะกับระบบที่ไม่เป็นเชิงเส้น (non-linear) แต่ไม่ใช่ว่าระบบที่ไม่เป็นเชิงเส้นจะอลวนเสมอไป จะมีช่วงพารามิเตอร์ที่ทำให้ระบบประพุดิตัวแบบอลวน แต่ระบบที่ไม่เป็นเชิงเส้นบางระบบก็เป็นเคออสตลอดเวลา

จุดที่เป็นประเด็นสำคัญอีกอย่างในทฤษฎีเคออส คือความอ่อนไหวอย่างสูงของเงื่อนไขเริ่มต้น กับ principle of weak ที่ถือว่าสาเหตุเบื้องต้นเพียงนิดเดียว สามารถก่อให้เกิดผลกระทบอย่างมากได้

2.1.2 ประวัติและความเป็นมา

จุดเริ่มต้นของทฤษฎีความอลวนนี้ สามารถสืบย้อนกลับไปได้ถึงในช่วงปี พ.ศ. 2443 (ค.ศ.

1900) จากการศึกษาปัญหาวงโคจรของวัตถุสามชิ้นในสนามแรงดึงดูดระหว่างกัน ซึ่งมีชื่อเรียกเป็นทางการว่า ปัญหาสามวัตถุ (three-body problem) โดย อองรี ปวงกาเร ซึ่งได้ค้นพบว่า วงโคจรที่ศึกษานั้นอาจจะมีลักษณะที่ไม่ได้เป็นวงรอบ (periodic) คือไม่ได้มีทางวิ่งซ้ำเป็นวงรอบ ยิ่งไปกว่านั้นวงโคจรนั้นก็ไม่ได้ขยายวงออกไปเรื่อย ๆ หรือมีลักษณะที่ลู่ออกเข้าหาจุดใด ๆ ต่อมาได้มีการศึกษาถึงปัญหาสมการเชิงอนุพันธ์ไม่เป็นเชิงเส้นที่เกี่ยวข้อง โดยที่

เบอร์คอฟ (G.D. Birkhoff) นั้นศึกษาปัญหาสามวัตถุ คอลโมโกรอฟ ศึกษาปัญหาความปั่นป่วน (หรือ เทอร์บิวเลนซ์) และปัญหาเกี่ยวกับดาราศาสตร์.

ส่วน คาร์ทไรท์ (M.L. Cartwright) และ ลิตเติลวูด (J.E. Littlewood) นั้นศึกษาปัญหาทางวิศวกรรมการสื่อสารด้วยคลื่นวิทยุ.

สเมล (Stephen Smale) นั้นอาจเป็นนักคณิตศาสตร์คนแรก ที่ทำการศึกษถึงปัญหาทางด้านพลศาสตร์ของระบบไม่เป็นเชิงเส้น. ถึงแม้ว่าความอลวนของเส้นทางโคจรของดาว นั้นยังไม่ได้มีการทำการสังเกตบันทึกแต่อย่างใด แต่ก็ได้มีการสังเกตพบ พฤติกรรมความอลวนในความปั่นป่วนของการเคลื่อนที่ของของไหล

เอ็ดเวิร์ด ลอเรนซ์ (Edward Lorenz) เป็นผู้ริเริ่มบุกเบิกทฤษฎีความอลวน เขาได้สังเกต พฤติกรรมความอลวน ในขณะที่ทำการทดลองทางด้านการพยากรณ์อากาศ ในปี ค.ศ. 1961 ลอเรนซ์ใช้คอมพิวเตอร์ซิมูเลชันแบบจำลองสภาพอากาศ ซึ่งในการคำนวณครั้งถัดมาเขาไม่ต้องการเริ่มซิมูเลชันจากจุดเริ่มต้นใหม่ เพื่อประหยัดเวลาในการคำนวณ เขาจึงใช้ข้อมูลในการคำนวณก่อนหน้านี้เพื่อเป็นค่าเริ่มต้น ปรากฏว่าค่าที่คำนวณได้มีความแตกต่างไปจากเดิมอย่างสิ้นเชิง เขาพบว่าสาเหตุเกิดจากการปัดเศษ ของค่าที่พิมพ์ออกมา จากค่าที่ใช้ในคอมพิวเตอร์ ซึ่งมีค่าน้อยมาก แต่สามารถนำไปสู่ความแตกต่างอย่างมากมาย เรียกว่า ไวต่อสภาวะเริ่มต้น

คำ "butterfly effect" ซึ่งเป็นคำที่นิยมใช้เมื่อกล่าวถึงทฤษฎีความอลวน นั้นมีที่มาไม่ชัดเจน เริ่มปรากฏแพร่หลายหลังจากการบรรยายของ ลอเรนซ์ ในปี ค.ศ. 1972 ภายใต้อีโก้หัวข้อ "Does the Flap of a Butterfly's Wings in Brazil Set Off a Tornado in Texas?" นอกจากนี้แล้วยังอาจมีส่วนมาจาก รูปแนวโคจรของตัวดึงดูดลอเรนซ์ ที่มีรูปร่างคล้ายผีเสื้อ ซึ่งเขาได้ตีพิมพ์ในบทความวิชาการก่อนหน้านี้

2.1.3 คุณสมบัติของ Chaos

ระบบที่เป็น Chaos จะต้องประกอบไปด้วยลักษณะดังต่อไปนี้

1) มีคุณสมบัติแบบไม่เป็นเชิงเส้น โดยที่ฟังก์ชัน f จะมีคุณสมบัติเชิงเส้นก็ต่อเมื่อ

$f(x + y) = f(x) + f(y)$ นั่นก็คือในระบบแบบไม่เป็นเชิงเส้น ผลลัพธ์ของระบบทั้งหมดไม่เท่ากับผลรวมของผลลัพธ์ที่เกิดจากส่วนย่อยๆ รวมกัน (โดยอาจจะมากหรือน้อยก็ได้) ความไม่เป็นเชิงเส้นนี้เองที่ทำให้ระบบที่แสดงความเป็นอลวนไม่ได้รับการศึกษามากในอดีต เพราะค่อนข้างยากในการที่จะศึกษาและเข้าใจ ข้อพึงระวังคือ การที่กล่าววาระบบเคออส จำต้องเป็นระบบที่ไม่เป็นเชิงเส้นนั้น ไม่ได้หมายความว่าระบบที่ไม่เป็นเชิงเส้นทุกระบบจะเป็น Chaos เสมอไป

2) ไม่ใช่เกิดแบบสุ่ม (คือเป็น deterministic ไม่ใช่ probabilistic) หรือกล่าวอีกแบบหนึ่งก็คือ ในระบบ Chaos พฤติกรรมทั้งหลายเกิดขึ้นภายใต้กฎเกณฑ์ที่แน่นอน เหตุการณ์ที่ไม่สามารถทำนายล่วงหน้าแบบการทอดลูกเต๋า จึงไม่ใช่ความเป็น Chaos แต่เป็นการสุ่ม การแปลคำนี้เป็น

ภาษาไทยโยใช้คำว่า “ทฤษฎีอลวน” จึงเป็นการแปลที่เหมาะสมเพื่อป้องกันความเข้าใจผิดว่าการเป็นเคออสไม่ใช่การสุ่มจึงมีคนเรียกว่า deterministic chaos

3) ไวต่อสภาวะเริ่มต้น (sensitive to initial conditions) การเริ่มต้นที่เงื่อนไขต่างกันนิดเดียวอาจทำให้ผลในตอนสุดท้ายต่างกันมาก สาเหตุที่ระบบ Chaos ไวต่อสภาวะเริ่มต้นก็เพราะว่ามันจะขยายความแตกต่างให้เพิ่มมากขึ้นอย่างรวดเร็วเช่น ที่พบบ่อยก็คือการขยายความแตกต่างให้เร็วขึ้นในระดับเลขยกกำลัง (exponential) ของเวลา

4) ไม่สามารถทำนายล่วงหน้าในระยะยาวได้ (long-term prediction is impossible) ซึ่งเป็นผลสืบเนื่องจากข้อ 3 เพราะการที่ไวต่อสภาวะเริ่มต้น จะทำให้เราไม่รู้ว่า ระบบที่น่าสนใจอยู่จะเป็นอย่างไรในระยะยาว อย่างไรก็ตามคุณสมบัติข้อนี้ไม่ได้แปลว่า การทำนายระยะสั้น (short-term prediction) จะเป็นสิ่งที่เป็นไปไม่ได้

นอกจากที่กล่าวมาข้างต้นแล้ว ระบบ Chaos ยังมีสมบัติอีกประการหนึ่ง คือ การแสดงลักษณะ “คล้ายกับตัวเอง” (self-similarity) หรือที่เรียกว่า “แฟร็กทัล” (fractal) นั่นเอง ลักษณะนี้จะปรากฏขึ้นเมื่อเราพล็อตเส้นทางการเคลื่อนที่ของระบบในระบบพิกัดที่บ่งถึงสภาวะ (phase space) ลักษณะคล้ายกับตัวเองนี้ หมายความว่า ไม่ว่าเราจะมองเส้นทางการเคลื่อนที่นี้จากสเกลเล็กหรือใหญ่แค่ไหน มันก็ยิ่งเหมือนเดิมอย่างไรก็ตามลักษณะคล้ายกับตัวเองแบบแฟร็กทัลนี้ ไม่ได้มีส่วนเกี่ยวข้องเป็นเงื่อนไขในการเกิด Chaos แต่อย่างไรก็ตามก็พบร่วมกันบ่อยครั้งเท่านั้น

2.1.4 ประโยชน์ของ Chaos

ในการศึกษาทฤษฎี Chaos มีประโยชน์อย่างน้อยใน 3 ทางด้วยกัน คือ ใช้การวิเคราะห์ระบบทำนายอนาคต ใช้ในการสร้างระบบ Chaos และใช้ในการควบคุม – สร้างความเสถียรให้กับระบบ ดังสามารถอธิบายได้ดังต่อไปนี้

1) ใช้ในการวิเคราะห์ระบบและทำนายอนาคตอย่างที่กล่าวมาข้างต้นว่า แม้ว่าเราจะไม่สามารถทำนายอนาคตของระบบ Chaos ในระยะยาวได้ แต่เราก็ยังสามารถทำนายอนาคตของมันในระยะสั้นได้ หากเราสามารถโมเดลที่อธิบายพฤติกรรมของระบบนั้น และทราบสภาวะเริ่มต้นอย่างแม่นยำพอสมควร เช่น ในปัจจุบันได้ใช้ Chaos เพื่อการวิจัยและทำนายอนุกรมลำดับเวลา (time – series data)

2) ใช้ในการสร้างระบบ Chaosทำไมเราต้องสร้างระบบ Chaos ขึ้นมาด้วยก็เพราะ มีผู้เชื่อว่า “ในธรรมชาติ ความอลวนเป็นสิ่งสากลมากกว่าและดีกว่าระเบียบแบบง่าย” อย่างเช่น บริษัท มัทซึตะยังใช้ทฤษฎี Chaos ควบคุมหัวฉีดของเครื่องล้างจานซึ่งพบว่าสามารถล้างจานได้สะอาดโดยประหยัดน้ำได้กว่าเครื่องล้างจานแบบอื่นๆ ทั้งนี้เพราะเส้นทางการเคลื่อนที่ของหัวฉีดที่ดูเหมือนไร้ระเบียบทำให้ครอบคลุมพื้นที่ได้ดีกว่าการเคลื่อนที่ตามแบบแผนปกติ

3) ใช้ในการควบคุม – สร้างความเสถียรให้กับระบบการที่ระบบ Chaos นั้นไวต่อสภาวะเริ่มต้นมาก การรบกวนเพียงเล็กน้อยจึงอาจก่อให้เกิดผลขยายได้มากในงานด้านวิศวกรรมควบคุม (control engineering) การเติมสัญญาณรบกวนเพียงเล็กน้อยที่เหมาะสมสู่ระบบที่เป็น Chaos จึงสามารถใช้ควบคุมให้ระบบนั้นอยู่ในสภาวะเสถียรหรือขับเคลื่อนให้ระบบนั้นไปสู่สภาวะที่ต้องการได้อย่างมีประสิทธิภาพ

นอกจากการประยุกต์ใช้หลักๆ ดังกล่าวข้างต้นแล้ว ทฤษฎี Chaos ยังสามารถประยุกต์ใช้ได้อีกในหลายสาขา เช่น ในด้านการสื่อสาร เราสามารถใช้สัญญาณแบบ Chaos ในการเข้ารหัส

ข้อมูล (encryption) เพื่อป้องกันคนแอบดูข้อมูล หรือใช้หลักการของทฤษฎีนี้ช่วยให้การหาค่าที่ดีที่สุด (Optimization) ของฟังก์ชันหนึ่ง ได้ค่าที่ดีที่สุดที่แท้จริง (global optimum) ได้ง่ายขึ้น เพราะ Chaos สามารถช่วยให้หาค่าที่ดีที่สุดเฉพาะบริเวณ (local optimum) ได้

2.1.5 ทฤษฎีของลอเรนซ์ (Lorenz's Theory)

จากความก้าวหน้าของการประดิษฐ์เครื่องคอมพิวเตอร์ทำให้เกิดเหตุการณ์ที่สำคัญคือ Edward Lorenz แห่งสถาบัน MIT เมื่อกลางทศวรรษที่ 60 อาจารย์ด้านอุตุนิยมวิทยา (meteorology) ผู้ซึ่งพยายามสร้างโมเดลการคำนวณในการพยากรณ์อากาศโดยใช้สมการง่ายๆ แสดงการปฏิสัมพันธ์ระหว่างอุณหภูมิกับกระแสลม เขาป้อนข้อมูลที่จุดทศนิยม 6 หลัก คือ 0.506127 เข้าเครื่องคอมพิวเตอร์ซึ่งพิมพ์ออกมาทุกๆ นาที ด้วยความเบื่อหน่ายที่ต้องนั่งคอยผลลัพธ์นานๆ เพราะคอมพิวเตอร์ยุคนั้นทำงานช้า Lorenz จึงตัดตัวเลขหลังจุดทศนิยมออกไปเสีย 3 หลัก ให้เหลือ 0.506 Lorenz เอาผลลัพธ์จากระยะหนึ่งมาเป็นจุดเริ่มต้นของการคำนวณ แล้วเริ่มต้นโปรแกรมใหม่ ผลของการคำนวณระยะแรกเหมือนกับการทดลองเก่าๆ ที่เคยทำมาครั้งแล้วครั้งเล่า หลังจากลูกโป่งคาแฟแล้วกลับมาดูตัวเลขใหม่ ปรากฏว่าผลลัพธ์ที่ได้นั้นมีความแตกต่างกันสิ้นเชิง โมเดลของดินฟ้าอากาศไปกันคนละทาง

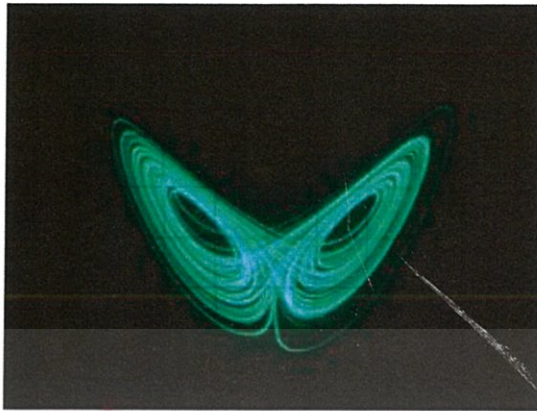
Lorenz คิดว่าเครื่องคอมพิวเตอร์คงจะมีอะไรผิดพลาด จึงทดลองซ้ำอีก ผลคำนวณยังคงยืนยันค่าความแตกต่างเหมือนครั้งแรก เมื่อทดลองต่อไปอีกสองสามครั้ง ซึ่งผลลัพธ์ที่ได้ก็ยิ่งถึงความถูกต้องของคอมพิวเตอร์ จนในที่สุดเขาก็ได้รู้ว่าการแตกต่างของโมเดลนั้นมาจากการลดตัวเลข 3 หลักหลังจุดทศนิยมซึ่งเป็น “เงื่อนไขเบื้องต้น” (initial condition) ในการคำนวณ เพราะความต่างของตัวเลขหลังจุดทศนิยมเพียงน้อยนิด ในเหตุเบื้องต้นคือ 1 : 1,000 เกิดเป็นผลลัพธ์ที่ต่างออกไปจากจุดเริ่มต้นจำนวนมหาศาลอย่างไม่น่าเชื่อถือ เปรียบเสมือนการไหวตัวของกระแสมหาสมุทรที่มาจากการกระพือปีกผีเสื้อแต่กลับกลายเป็นสร้างผลกระทบใหญ่หลวงทางดินฟ้าอากาศได้ Lorenz ซึ่งขนานนามว่าเป็น “ผลกระทบผีเสื้อ” (Butterfly effect) ซึ่งกลายเป็นถ้อยคำที่โด่งดังไปทั่วโลก

ในที่สุดเขาก็สรุปว่ากาลดตัวเลขขนาด 3 หลักหลังจุดทศนิยมซึ่งเป็นเงื่อนไขเบื้องต้นในการคำนวณทำให้เกิดผลลัพธ์ที่แตกต่างกันไปจากจุดเริ่มต้นจำนวนมหาศาลอย่างไม่น่าเชื่อถือ โดยสมการของลอเรนซ์ เป็นดังนี้

$$\frac{dX}{dt} = \sigma(y - x) \quad (2.1)$$

$$\frac{dY}{dt} = rx - y - xz \quad (2.2)$$

$$\frac{dZ}{dt} = xy - bz \quad (2.3)$$



รูปที่ 2.1 สัญญาณเคออสที่สร้างจากสมการ Lorenz

เมื่อ a, b และ c เป็นค่าคงที่ และสำหรับ $a = 10, c = 8/3$ เมื่อมีการเปลี่ยนค่า b จะทำให้เกิดพฤติกรรมต่างๆ ตามตารางที่ 2-1 โดยที่ลอเรนซ์ พบว่าระบบจะเกิดพฤติกรรมไร้ระเบียบเมื่อ b มีค่ามากกว่า 24.74 และผลที่ได้แตกต่างกันไปขึ้นอยู่กับเงื่อนไขเริ่มต้นและค่าพารามิเตอร์ที่ใช้จากเหตุการณ์ดังกล่าวกระตุ้นให้นักวิจัยหลายท่านหันมาวิจัยทฤษฎีไร้ระเบียบมากขึ้น ทั้งที่นำสมการของลอเรนซ์มาศึกษาต่อและปรับเข้ากับงานของตนหรือคิดค้นระบบไร้ระเบียบใหม่ ในงานที่ตนเองเกี่ยวข้องอยู่

ตารางที่ 2.1 พฤติกรรมต่างๆของระบบเมื่อ $a = 10, c = 8/3$ และค่า b มีการเปลี่ยนค่าในช่วงต่างๆกันมีผลลัพธ์ในแต่ละช่วงดังต่อไปนี้

b	Attractor
$B < 1.00$	-มีจุดสมดุลเสถียรที่จุดกำเนิด $[0,0,0]$
$1.00 \leq b \leq 13.93$	-มีจุดสมดุลเสถียรเกิดขึ้นใหม่ 2 จุด ในขณะที่จุดกำเนิด $[0,0,0]$ กลายเป็นจุดสมดุลไม่เสถียรเกิดพฤติกรรมไร้ระเบียบชั่วคราว
$13.93 \leq b \leq 24.74$	-เกิดพฤติกรรมไร้ระเบียบ
$B > 24.74$	-เกิดพฤติกรรมไร้ระเบียบ

ศาสตราจารย์ Lorenz กล่าวว่า “ในทางทฤษฎีด้านอุตุนิยมวิทยา ผีเสื้อใหญ่ตัวหนึ่งกระพือปีกที่ย่องงก สามารถทำให้ดินฟ้าอากาศที่แคลิฟอร์เนียเปลี่ยนแปลงเป็นพายุได้เมื่อหนึ่งเดือนให้หลัง” ทฤษฎีไร้ระเบียบบอกจากพูดถึงความสำคัญของการเปลี่ยนแปลงของสาเหตุเบื้องต้นแล้วยังพูดถึงเรขาคณิตเศษส่วน (fractal geometry)

คุณสมบัติสำคัญของ fractal หรือเศษส่วนไม่รู้จบ คือ “ความคล้ายตนเอง” (self-similarity) นั่นคือโครงสร้างของเศษส่วนที่ขยายตัวออกไปอย่างไม่สิ้นสุด ไม่ว่าในมิติทางกาล(time) หรือมิติทางเทศะ (space) ล้วนแล้วแต่มีรูปร่างคล้ายคลึงซ้ำกับรูปแบบอันแรก อันเป็นแม่แบบของมัน

Benoit Mandelbrot ผู้คนคว่าทฤษฎีไร้ระเบียบทางคณิตศาสตร์ สรุปลความเชื่อมโยงแบบแน่นระหว่างคณิตศาสตร์เศษส่วนกับทฤษฎีไร้ระเบียบไว้ดังนี้

“ความสัมพันธ์ระหว่างรูปร่างที่เป็นเศษส่วน (fractal) กับระบบที่ไร้ระเบียบ (Chaotic system) เกิดจากคุณสมบัติที่ว่า การเปลี่ยนแปลงที่ขึ้นค่าเริ่มต้น ประกอบกับการพัฒนาลักษณะพลวัตอันซับซ้อนและกลับไปกลับมา ทำให้เศษส่วนแต่ละอันจึงไม่เหมือนกันร้อยเปอร์เซ็นต์ แต่ละภาพจะมีความคลาดเคลื่อนไปบ้างย่อมต่างกันไป เป็นสิ่งที่ทฤษฎีไร้ระเบียบพยายามตอบคำถามว่าทำไมเกล็ดหิมะแต่ละเกล็ดที่โปรยปรายลงมา จึงมีความแตกต่างไปเล็กน้อย ก็เนื่องจากว่าผลรวมของตัวแปรที่มีอิทธิพลต่อการเกิดเกล็ดแต่ละเกล็ดต่างกัน”

2.1.6 สมการของลอเรนซ์ (Lorenz Equation)

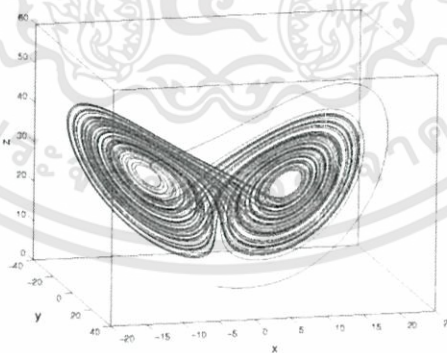
สมการลอเรนซ์เป็นสมการเชิงอนุพันธ์ (Different Equation) เป็นระบบพลวัตไม่เป็นเชิงเส้น 3 มิติ ทฤษฎีดังกล่าวใช้อธิบายการเคลื่อนตัวของก๊าซและของเหลว สมการที่ใช้อธิบายปรากฏการณ์ Lorenz Attractor เขียนอยู่ในรูปสมการทางคณิตศาสตร์เชิงอนุพันธ์ได้ดังต่อไปนี้

$$\frac{dx}{dt} = \sigma(y - x) \quad (2.1)$$

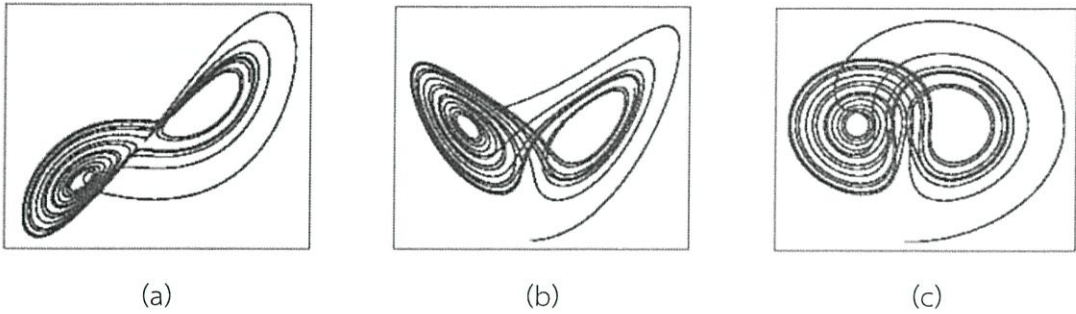
$$\frac{dy}{dt} = rx - y - xz \quad (2.2)$$

$$\frac{dz}{dt} = xy - bz \quad (2.3)$$

σ คือค่าอัตราส่วนระหว่างแรงต้านของของเหลวและแรงดันของก๊าซหรือของเหลว, ρ คือค่าความต่างของอุณหภูมิระหว่างช่วงบนสุดกับล่างสุดของตัวกลาง และ β คืออัตราส่วนระหว่างความกว้างกับความยาวของภาชนะที่ใช้บรรจุ เราเรียกปรากฏการณ์นี้ว่า The Butterfly effect เนื่องด้วยลักษณะการกระจายตัวของค่าที่ได้มีลักษณะคล้ายกับปีกผีเสื้อ ถ้าเราเปลี่ยนแปลงค่า σ, ρ หรือ β เพียงตัวเดียว ถึงแม้ว่าจะเป็นเพียงเล็กน้อยก็ตาม ผลลัพธ์ที่ได้จะเปลี่ยนแปลงไปจากเดิมโดยสิ้นเชิง จากสมการของ Lorenz เมื่อพล็อตความสัมพันธ์ลงบนแกน X,Y,Z จะได้ดังรูปที่ 2.2



รูปที่ 2.2 สัญญาณเคออสจากสมการ Lorenz ในลักษณะ 3 มิติ



รูปที่ 2.3 สัญญาณเคออสจากสมการ Lorenz ในลักษณะ 2 มิติ ระหว่างระนาบต่างๆ
 (a) ระนาบ x,y (b) ระนาบ x,z (c) ระนาบ y,z

2.1.7 สมการของเซน (Chen's Equation)

สมการนี้เกิดขึ้นจากการที่มีผู้ต้องการจะควบคุมการเกิดสัญญาณเคออสติก (Chaotic Signal) ที่เกิดจากสมการลอเรนซ์ ดังนั้นจึงได้มีการคิดค้นโดยใช้สมการลอเรนซ์เป็นพื้นฐาน ซึ่งเรียกว่า สมการของเซน และผู้ค้นพบคือ Tetsushi Ueta และ Guanrong Chen [1] ซึ่งมีรูปแบบสมการดังนี้

$$\frac{dx}{dt} = a(y - x) \quad (2.4)$$

$$\frac{dy}{dt} = (c - a)x - xz - cy \quad (2.5)$$

$$\frac{dz}{dt} = xy - bz \quad (2.6)$$

และหลังจากนั้นได้มีการพัฒนาสมการของเซน เพื่อให้ปราศจากพจน์ที่เป็นการคูณกันของ XZ และ XY โดยใช้ Signum Function หรือเขียนย่อว่า $\text{sgn}(\cdot)$ ซึ่งถูกคิดค้นโดย M.A. Aziz - Alaout และ Guanrong Chen [2] ซึ่งมีรูปแบบสมการดังนี้

$$\frac{dx}{dt} = a(y - x) \quad (2.7)$$

$$\frac{dy}{dt} = \text{sgn}(x)(c - az) + cxy \quad (2.8)$$

$$\frac{dz}{dt} = \text{sgn}(y)x - bz \quad (2.9)$$

2.1.8 สมการของ Rossler's (Rossler Equation)

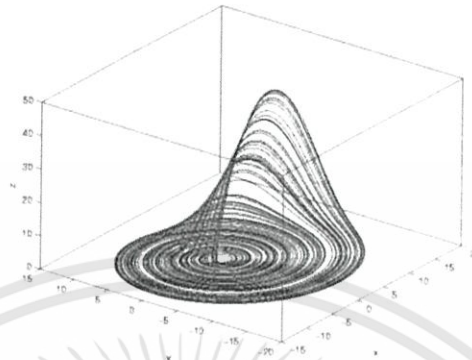
สมการ Rossler นี้คิดโดย Otto Rossler ซึ่งสร้างจากชุดของสมการอนุพันธ์ปกติเขียนในรูปสมการทางคณิตศาสตร์ได้ดังต่อไปนี้ :

$$\frac{dX}{dt} = -y - z \quad (2.10)$$

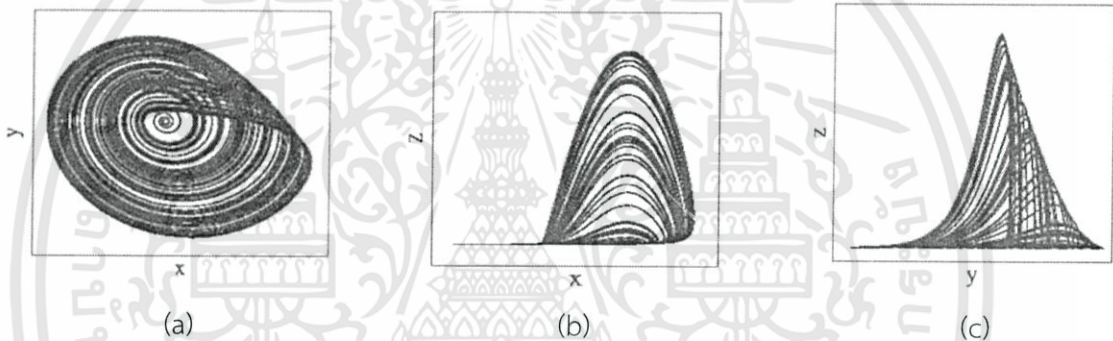
$$\frac{dY}{dt} = x - ay \quad (2.11)$$

$$\frac{dZ}{dt} = a + z(x - c) \quad (2.12)$$

จากสมการของ Rossler จะเห็นได้ว่าสมการของ Rossler จะมีเทอมผลคูณ 2 ตัวแปร เพียงเทอมเดียวซึ่งน้อยกว่ารูปแบบสมการของ Lorenz ทำให้การใช้วงจร multiplex ลดลง และเมื่อพล็อตความสัมพันธ์ลงบนแกน x,y,z จะได้ดังรูปที่ 2.4



รูปที่ 2.4 สัญญาณเคออสจากสมการ Rossler ในลักษณะ 3 มิติ



รูปที่ 2.5 สัญญาณเคออสจากสมการ Rossler ในลักษณะ 2 มิติ ระหว่างระนาบต่างๆ
(a) ระนาบ x,y (b) ระนาบ x,z (c) ระนาบ y,z

2.2 ระบบที่เชื่อมต่อกับระบบอินเตอร์เน็ต

RF module (Radio Frequency Module) เป็นวงจรอิเล็กทรอนิกส์เล็กๆ ที่ใช้สำหรับการส่งหรือรับสัญญาณวิทยุบนคลื่นความถี่ที่กำหนดไว้ซึ่งนิยมใช้อย่างแพร่หลายเนื่องมาจากความยากลำบากในการสร้างวงจรวิทยุ การออกแบบวงจรอิเล็กทรอนิกส์วิทยุที่ดีนั้นมีความซับซ้อนมาก เพราะการที่จะทำวงจรที่คลื่นความถี่ใดความถี่หนึ่งต้องการความแม่นยำของส่วนประกอบของวงจร การจัดวางที่ดีและต้องคำนึงถึงอุปสรรคในเรื่องความอ่อนไหวของคลื่นวิทยุซึ่งวิศวกรผู้ที่ทำหน้าที่ออกแบบส่วนใหญ่จะนิยมการใช้ไมโครมากกว่าเนื่องจากสามารถประหยัดเวลาและค่าใช้จ่ายในการพัฒนามากกว่า

RF module มักจะนิยมใช้ในเครื่องใช้ไฟฟ้าประจำบ้านเช่น นำไปติดตั้งไว้ที่ประตูโรงรถ สัญญาณเตือนภัยแบบไร้สาย รีโมทคอนโทรลต่างๆ ระบบเซนเซอร์อัจฉริยะและอุปกรณ์ภายในบ้านที่ทำงานแบบอัตโนมัติ ซึ่งจะเห็นได้ว่าอุปกรณ์ที่กล่าวมา RF module จะเข้าไปแทนที่ระบบสื่อสารผ่านคลื่นอินฟราเรดเนื่องจากสามารถเดินทางผ่านสิ่งกีดขวางได้

คลื่นความถี่ที่ RF module ใช้นั้นโดยทั่วไปจะใช้คลื่นความถี่เชิงพาณิชย์อยู่แล้ว เนื่องมาจากข้อตกลงในประเทศและระหว่างประเทศในการใช้คลื่นวิทยุสื่อสารโดยมีคลื่นความถี่ที่ 433.92 MHz, 315 MHz, 868 MHz และ 915 MHz

2.2.1 การมอดูเลตสัญญาณ RF

การมอดูเลตสัญญาณ RF ที่นิยมใช้กันทั่วไปในการรับส่งสัญญาณ RF นั้นจะมี 3 แบบ

- 1). ASK
- 2). FSK
- 3). OOK

2.3 โปรแกรม Artisteer

ในปัจจุบันนี้ CMS ได้เข้ามามีบทบาทช่วยให้ใครหลายคน ทำเว็บไซต์ได้ง่ายขึ้นโดยไม่ต้องมีพื้นฐานทางภาษา html มากนัก ก็สามารถมีเว็บไซต์ได้ง่ายๆ เพียงแค่คลิกเลือกทำงานกับโมดูลสำเร็จรูปที่พัฒนา CMS จัดเตรียมไว้ให้ป้อนข้อมูลแทรกรูปภาพ แล้วบันทึกผล สิ่งที่ต้องการก็จะปรากฏบนโลกออนไลน์แล้ว CMS ที่คนไทยเรารู้จักและคุ้นเคยเช่น Wordpress , Joomla , Drupal , จึงเป็นที่นิยมกันมากในเมืองไทยและทั่วโลก แต่จะมีเว็บไซต์สักกี่แห่งที่สวยงามโดดเด่น ส่วนใหญ่ก็จะมีลักษณะคล้ายๆกันหมด เปลี่ยนแค่ชื่อโลโก้และสีสันทันเท่านั้นเพราะส่วนใหญ่ล้วนใช้ของฟรีกันเป็นหลัก ยิ่งถ้าไม่เก่งจากเรื่องการออกแบบดีไซน์ และมีความรู้ภาษา PHP , CSS ก็มักจะหนีจากรูปแบบเดิมๆได้ยาก

Artisteer จึงเป็นโปรแกรมหนึ่งที่จะช่วยในการออกแบบเว็บไซต์ โดยผู้ใช้ไม่จำเป็นต้องมีความรู้เกี่ยวกับภาษา HTML หรือโปรแกรมที่ใช้ในการสร้างเว็บไซต์ อย่างhtml , css , Photoshop แม้แต่น้อยหรืออาจจะไม่มีหัวศิลปะไม่มาก ก็สามารถทำเว็บออกมาได้เลย โดยใช้เวลานานอีกด้วย ที่สำคัญยังสามารถแก้ไขเว็บที่สร้างขึ้นมานี้ได้ตลอดเท่าที่ต้องการ

Artisteer สามารถนำไปประยุกต์ใช้งานได้อย่างหลากหลาย ทั้งสร้างเว็บไซต์ขึ้นมาเอง หรือใช้สร้างเทมเพลตเพื่อนำไปใช้กับเว็บที่สร้างจาก CMS ต่างๆ เช่น Joomla ,Wordpress, Drupal , Blogger ฯลฯ เป็นต้น

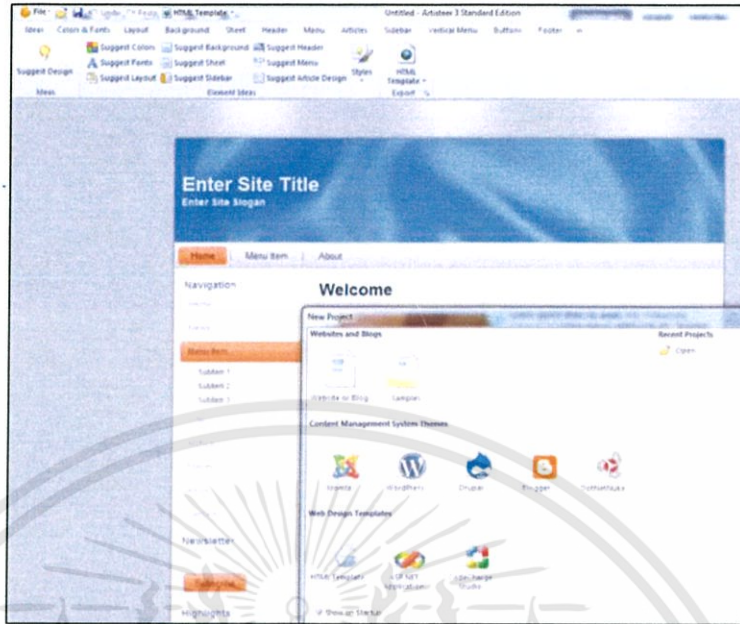
Artisteer ปัจจุบันได้พัฒนาจนถึงเวอร์ชัน 4 มีให้ใช้งานทั้งบน windows และ Mac สำหรับราคามี 2 แบบให้เลือกคือ Home&Academic Edition ราคา \$49.95 ตกประมาณ 1,600 กว่าบาท กับแบบ Standard Edition ราคา \$129.95 ตกประมาณ 4,xxx กว่าบาท

ตารางที่ 2.2 ข้อแตกต่างของ 2 เวอร์ชัน มีดังนี้ครับ

เวอร์ชัน	Home & Academic Edition	StandardEdition
ราคา	ประมาณ 1600 บาท	ประมาณ 4,XXX บาท
Free upgrades for 1 year	✔	✔
Design Suggestions	✔	✔

Design Features	✓	✓
Export as XHTML+CSS	✓	✓
Export as Wordpress Theme	✓	✓
Export as Blogger Template	✓	✓
Export as Joomla 1.5 - 2.5 Template		✓
Export as Drupal 5/6/7 Theme		✓
Export as DotNetNuke 4.9+ Skin		✓
Export as ASP.NET Application		✓
Export as ASP.NET Application		✓
Export to CodeCharge Studio		✓
Custom Value Dialogs (screenshot)		✓
Library of Textures, Glares and Gradients	partial (75%) (มีน้อยกว่ารุ่นมาตรฐาน)	✓
Number of Color and Font Schemes	50+ (มีน้อยกว่ารุ่นมาตรฐาน)	70+ (มี 70 กว่าแบบ)

ตัวอย่างหน้าต่างของโปรแกรม Artisteer เมื่อเราทำการติดตั้งเสร็จแล้วทำการเปิดขึ้นมา เพื่อจะทำการสร้าง เทมเพลตที่เราต้องการ



รูปที่ 2.5 หน้าแรกของโปรแกรม Artisteer

2.4 การใช้งานโมดูล UART กับพอร์ตอนุกรม

โมดูล Universal Asynchronous Receiver Transmitter (UART) หรือการสื่อสารข้อมูลในรูปแบบอะซิงโครนัส เป็นหนึ่งในโมดูลที่เกี่ยวข้องกับการอินพุต/เอาต์พุตแบบอนุกรมที่มีอยู่ใน dsPIC30F โดยโมดูล UART จะเป็นอะซิงโครนัสแบบฟูลดูเพล็กซ์ (full duplex) รับส่งข้อมูลได้พร้อมกัน (สองทิศทาง) สามารถนำมาประยุกต์ใช้งานกับการเชื่อมต่อกับอุปกรณ์ภายนอก เช่น คอมพิวเตอร์, RS-232 และ RS-485 เป็นต้น

ไมโครคอนโทรลเลอร์ dsPIC30F2010 มีโมดูล UART ให้ใช้งานได้ 1 UART โดยกำหนดขาพอร์ตสำหรับโมดูล UART ไว้ที่ขาพอร์ต U1RX/RF2 สำหรับข้อมูลอนุกรม และ U1TX/RF3 สำหรับส่งข้อมูลอนุกรม นอกจากนี้ยังได้เพิ่มเติมพอร์ตอนุกรมเสริม (UART Alternate I/O) ที่ตำแหน่งขาพอร์ต U1ATX/RC13 สำหรับการส่งข้อมูลและ U1ARX/RC14 สำหรับการรับข้อมูลอนุกรม การใช้งานต้องเลือกอย่างใดอย่างหนึ่งเท่านั้น โดยสามารถเลือกการใช้งานผ่านการเซตบิต ALTO (บิต 10) ในรีจิสเตอร์ UxMODE (UARTx Mode Register)

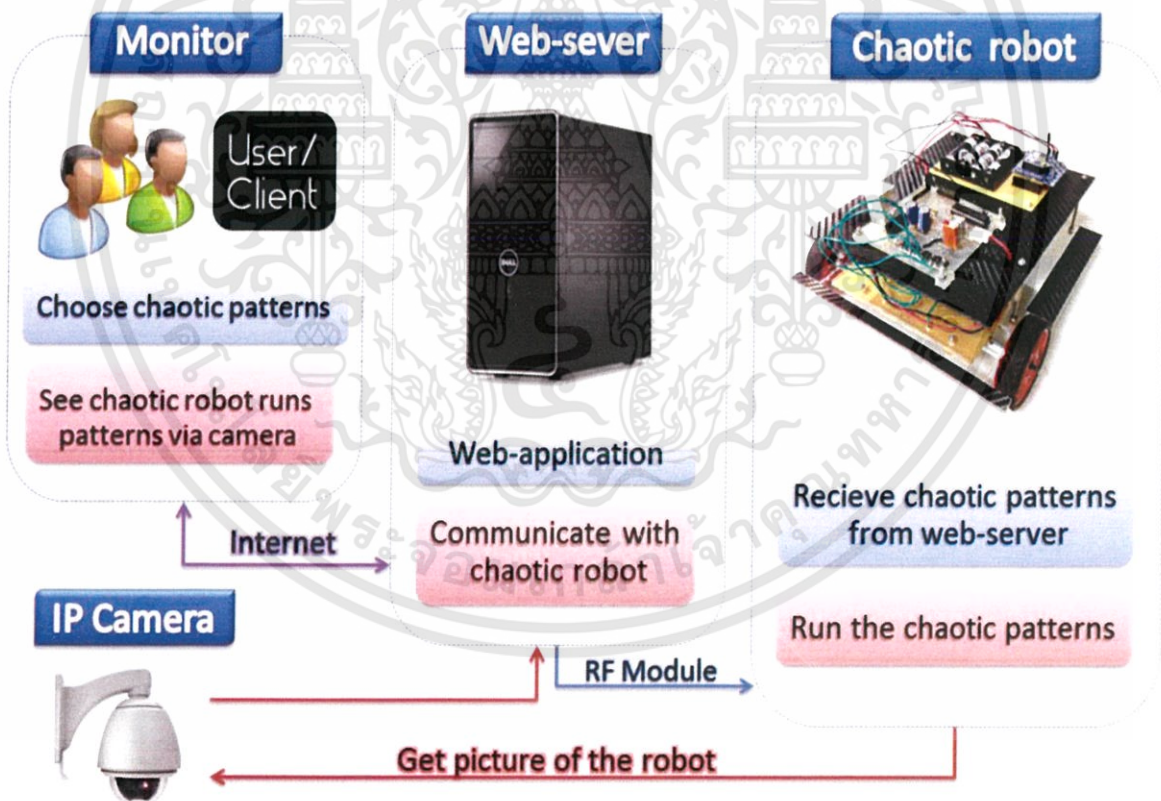
บทที่ 3

การออกแบบและโครงสร้าง

3.1 ภาพรวมของการทำงานทั้งหมด

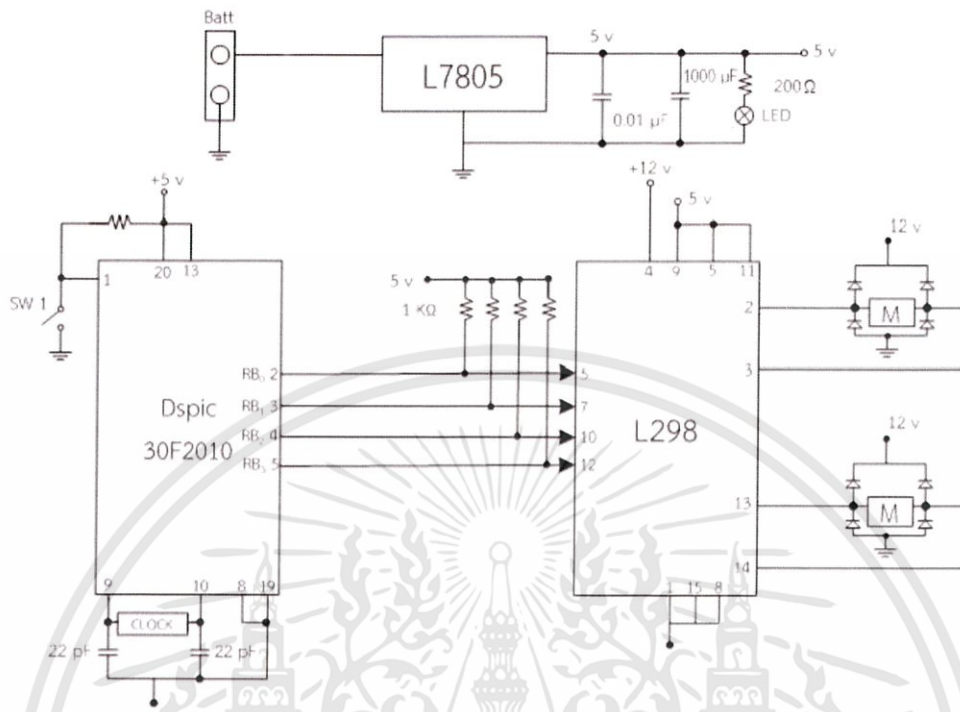
การทำงานของระบบนั้น จะแบ่งเป็น 3 โมดูล ดังนี้

1. ในส่วนของ User เมื่อเข้าใช้งานหน้าเว็บไซต์ User สามารถศึกษาทฤษฎีคลื่นในแบบต่างๆ ทั้ง 3 ทฤษฎีได้ และเมื่อ User เกิดความสนใจในทฤษฎีใด User สามารถทำการคัดเลือกทฤษฎีที่สนใจในหน้าเพจของการทดลอง เพื่อดูการเดินของหุ่นยนต์จริงว่าเป็นอย่างไร ผ่าน IP Camera ได้
2. ในส่วนของ Web-server เมื่อ User ทำการเลือกในหน้าเพจของการทดลองว่า User ต้องการดูการทดลองของทฤษฎีใดแล้ว Web-server ทำการรับคำสั่งมา แล้วส่งคำสั่งนั้นๆ ไปยังตัวหุ่นยนต์โดยผ่าน RF Module และเมื่อหุ่นยนต์ได้รับคำสั่งและเดินตามแบบแพทเทินแล้ว IP Camera ทำการแสดงผลเป็นแบบ live Streaming แล้วส่งกลับมายัง Web-server
3. ในส่วนของตัวหุ่นยนต์ เมื่อได้รับคำสั่งจาก Web-server ว่าหุ่นยนต์จะเดินตามแพทเทินใด หุ่นยนต์จะเดินตามแบบแพทเทินนั้นที่มีการเขียนโปรแกรมไว้

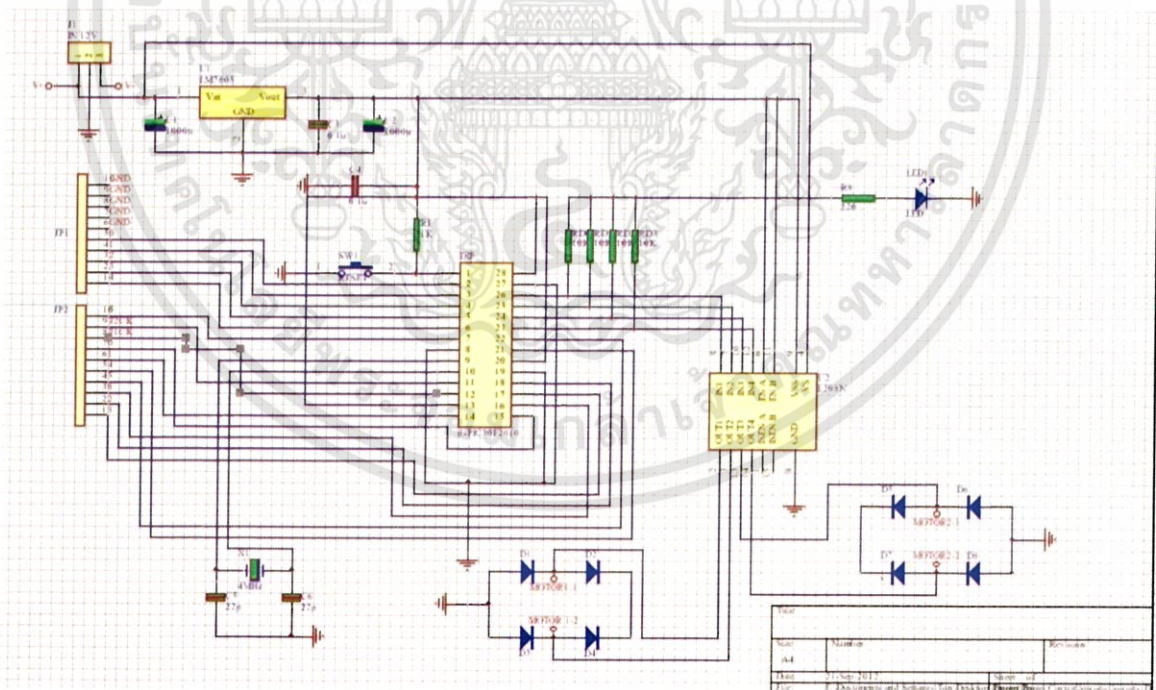


รูปที่ 3.1 ภาพรวมของการทำงาน

3.2 วงจรรวมของหุ่นยนต์



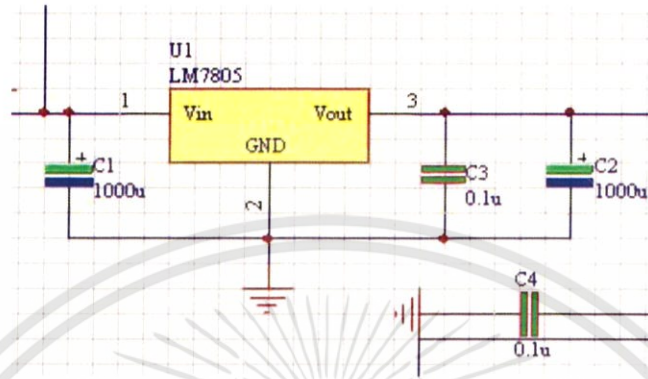
รูปที่ 3.2 ภาพการออกแบบวงจรรวมของหุ่นยนต์ผ่านโปรแกรม Microsoft visio



รูปที่ 3.3 ภาพวงจรรวมของหุ่นยนต์ผ่านโปรแกรม Protel

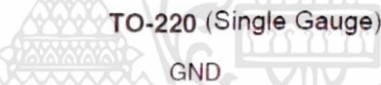
วงจรรวมของหุ่นยนต์นี้ เราจะมีชิ้นส่วนที่สำคัญ 3 ชิ้นส่วนด้วยกันคือ LM7805 , L298 , Microcontroller Dspic30F 2010 ซึ่งจะอธิบายถึงการทำงานและรายละเอียดของมันดังต่อไปนี้

3.2.1 LM7805

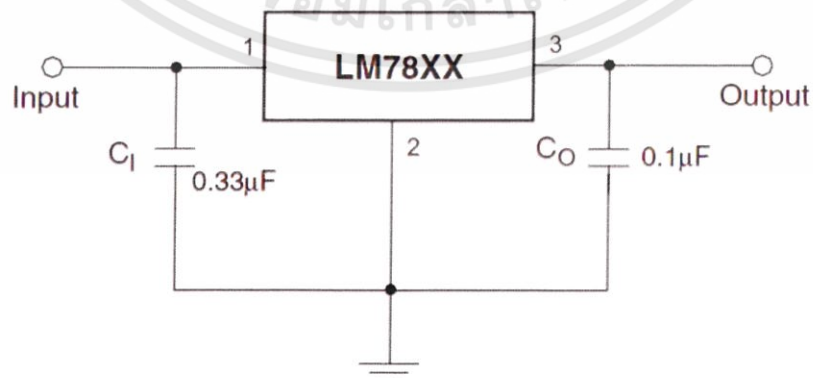


รูปที่ 3.4 LM7805

LM7805 ทำหน้าที่แปลงกระแสไฟฟ้าจาก 12 V ไปเป็น 5 V เพื่อจ่ายไฟต่อให้กับ L298 และ Microcontroller Dspic30F 2010 เปรียบเสมือนกับเป็นแหล่งจ่ายไฟ



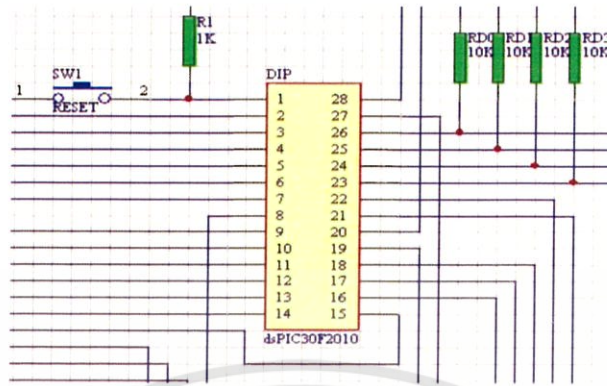
รูปที่ 3.5 LM7805



รูปที่ 3.6 DC Parameters.

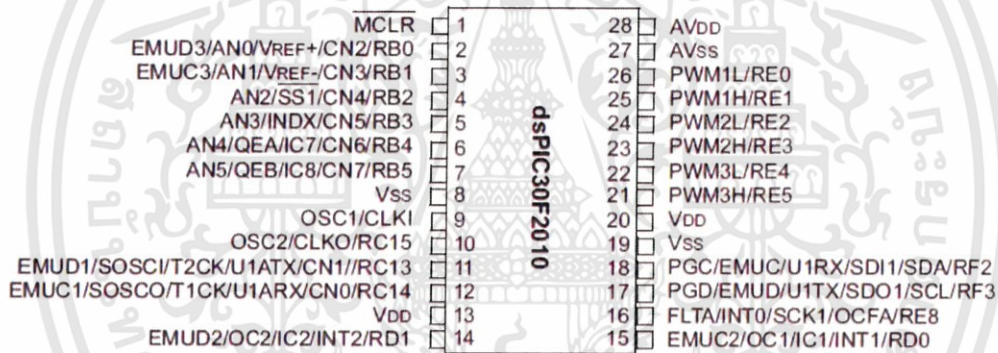
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Micorcontroller Dspic30F 2010



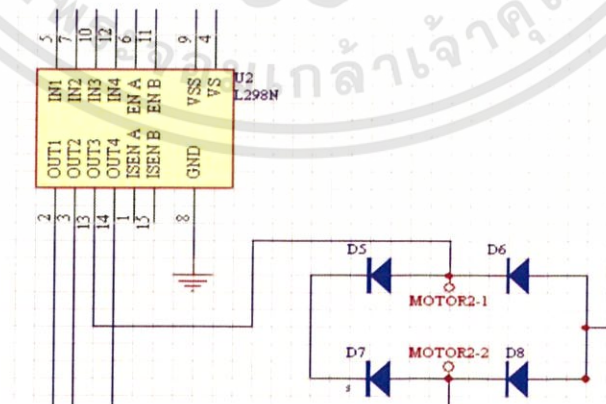
รูปที่ 3.7 Microonroller Dspic30F 2010

Microcontroller Dspic30F 2010 ทำหน้าที่รับข้อมูลจากผู้ใช้ (pattern) เพื่อนำมาประมวลผลโดยการคำนวณ แล้วส่งผลเป็น Output ออกมาในรูปแบบของเลขฐานสอง เพื่อนำไปใช้กับ L298 ต่อไป



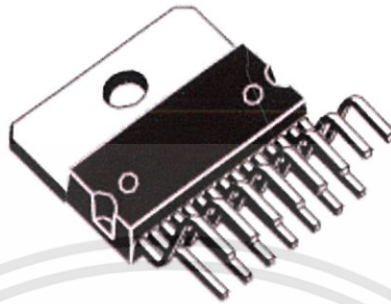
รูปที่ 3.8 Pin Diagrams ของ Dspic30F 2010

3.2.3 L298

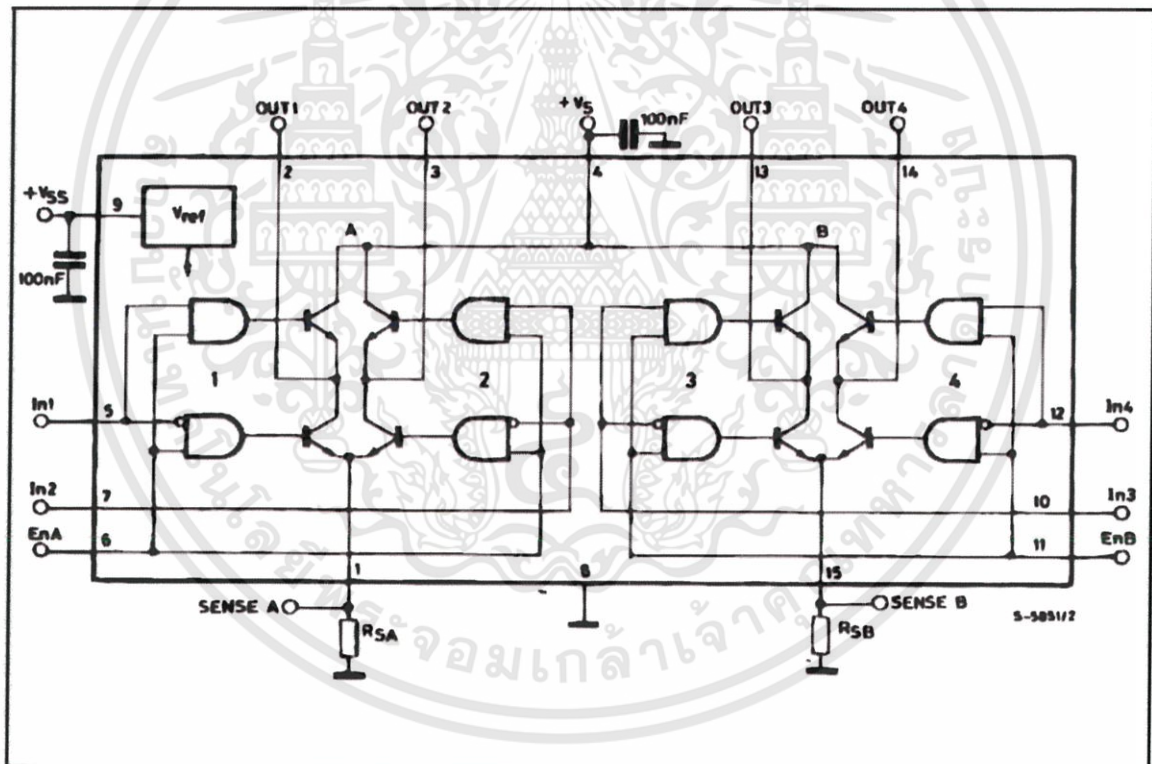


รูปที่ 3.9 L298

L298 ทำหน้าที่รับข้อมูลจาก MicrocontrollerDspic30F 2010 เพื่อนำไปใช้กับการขับเคลื่อนมอเตอร์ 2 ตัวและ 2 ทิศทางในแบบอิสระ L298



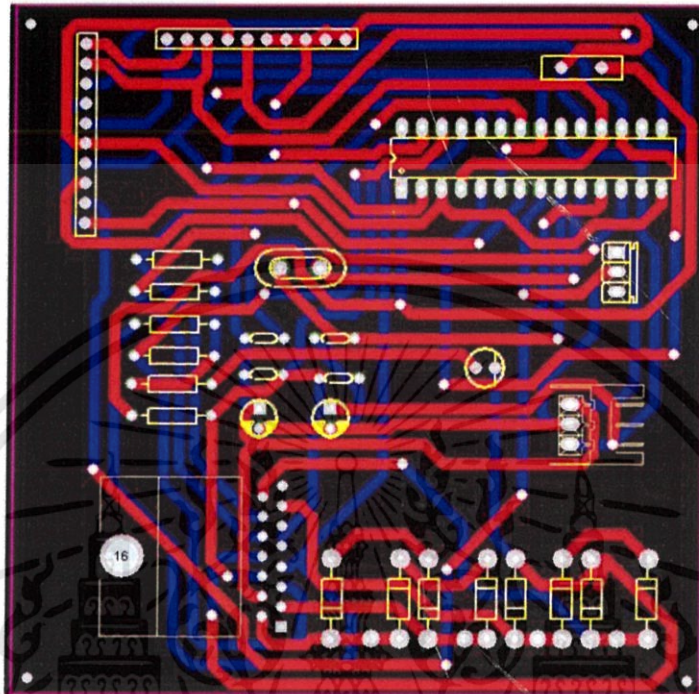
รูปที่ 3.10 L298



รูปที่ 3.11 Block Diagram

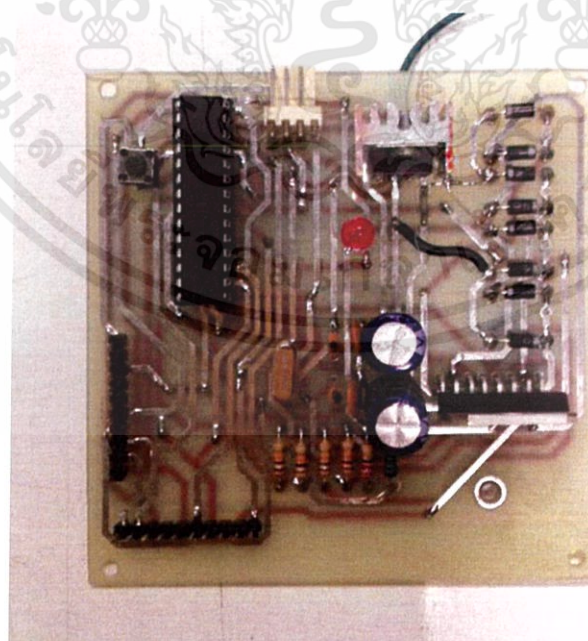
3.3 วงจรโดยรวมของหุ่นยนต์ เป็นไฟล์ PCB

เป็นวงจรเดียวกับกับ 3.2 ผ่านโปรแกรม Protel เช่นกันเพียงแต่บันทึกเป็นไฟล์ PCB ใช้สำหรับการกัดปรินต์



รูปที่ 3.12 ภาพวงจรรวมของหุ่นยนต์ผ่านโปรแกรม Portel (ไฟล์ PCB)

3.4 วงจรโดยรวมของหุ่นยนต์หลังจากทำการกัดปรินต์และบัดกรีเสร็จ



รูปที่ 3.13 ภาพวงจรรวมของหุ่นยนต์ที่ผ่านการกัดปรินต์และบัดกรีเสร็จ

3.5 การออกแบบเว็บไซต์โดยใช้โปรแกรม Artisteer

ใช้โปรแกรม Artisteer ในการสร้างเว็บไซต์ โดยการแบ่งเป็นหัวข้อใหญ่ๆได้ 8 หัวข้อดังต่อไปนี้

3.5.1 หน้า HOME

ในหน้า HOME จะเป็นหน้าหลักมีข้อความบอกว่าเว็บไซต์นี้เป็นผลงานของใคร มหาลัยอะไร และมีปุ่มต่างๆดังนี้ ใน Menu bar ประกอบไปด้วย

- 1) ปุ่ม HOME เมื่อเรากดคลิกจะทำการกลับมาหน้าแรกคือหน้า HOME
- 2) ปุ่ม CONTRACT เมื่อเรากดคลิกจะทำการเปลี่ยนหน้าใหม่ไปหน้า CONTRACT

ส่วนต่อไปคือส่วนเนื้อหาของเว็บไซต์ที่อยู่ตรงกลางจะประกอบไปด้วย layout ที่มีลักษณะ 5 Columns ในแต่ละ Columns จะแสดงรูปภาพพร้อมปุ่ม “Click!” เพื่อให้ User ที่สนใจในหัวข้อดังกล่าวได้เข้าไปอ่านต่อไป โดยทั้ง 5 Columns นี้ แบ่งออกเป็น

- 1) Columns OVERVIEW จะเป็นเนื้อหาโดยรวมของการทำงานว่าทำงานอย่างไร
- 2) Columns DEFINITION จะเป็นเนื้อหาเกี่ยวกับความหมายและคำนิยาม
- 3) Columns THEORY จะเป็นเนื้อหาเกี่ยวกับทฤษฎีที่เราใช้ในการทดลอง
- 4) Columns DESIGN จะเป็นเนื้อหาเกี่ยวกับการออกแบบที่เกิดขึ้นในการทดลอง
- 5) Columns EXPERIMENT จะเป็นเนื้อหาที่เกี่ยวกับการทดลองเดินของหุ่นยนต์

สำหรับส่วนสุดท้ายที่มีอยู่ด้านล่างของเว็บไซต์ในหน้า HOME ประกอบไปด้วย รูปภาพ 2 รูปทางด้านขวา 1 รูปและทางด้านซ้ายอีก 1 รูป ซึ่งเปรียบเสมือนปุ่มกดที่สามารถนำพา User ไปยังหน้าใหม่ที่ต้องการได้ โดยทั้ง 2 รูปแบ่งออกเป็นดังนี้

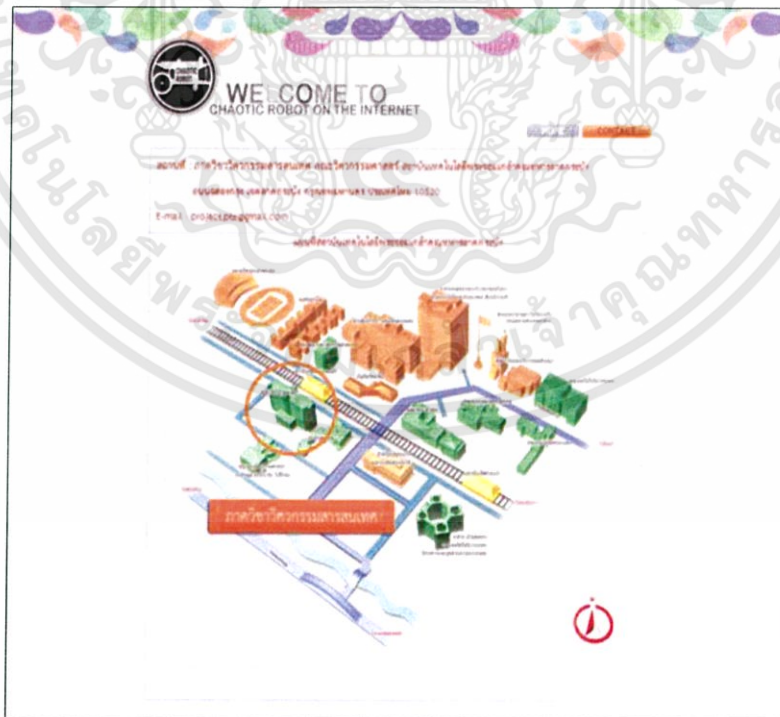
- 1) ทางด้านขวาหากเรากดที่รูป จะทำการเปิดเพจใหม่ โดยเป็นการเปิดเว็บไซต์ www.ite.kmitl.ac.th ขึ้นมา
- 2) ทางด้านซ้ายหากเรากดที่รูป จะทำการเปิดเพจใหม่ โดยเป็นการเปิดเว็บไซต์ www.kmitl.ac.th ขึ้นมา



รูปที่ 3.14 หน้า HOME

3.5.2 หน้า CONTACT

ในหน้า CONTACT นี้เป็นข้อมูลของการติดต่อว่าอยู่ที่ไหนติดต่อได้ทางใด อย่งไรบ้าง



รูปที่ 3.15 หน้า CONTACT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 หน้า OVERVIEW

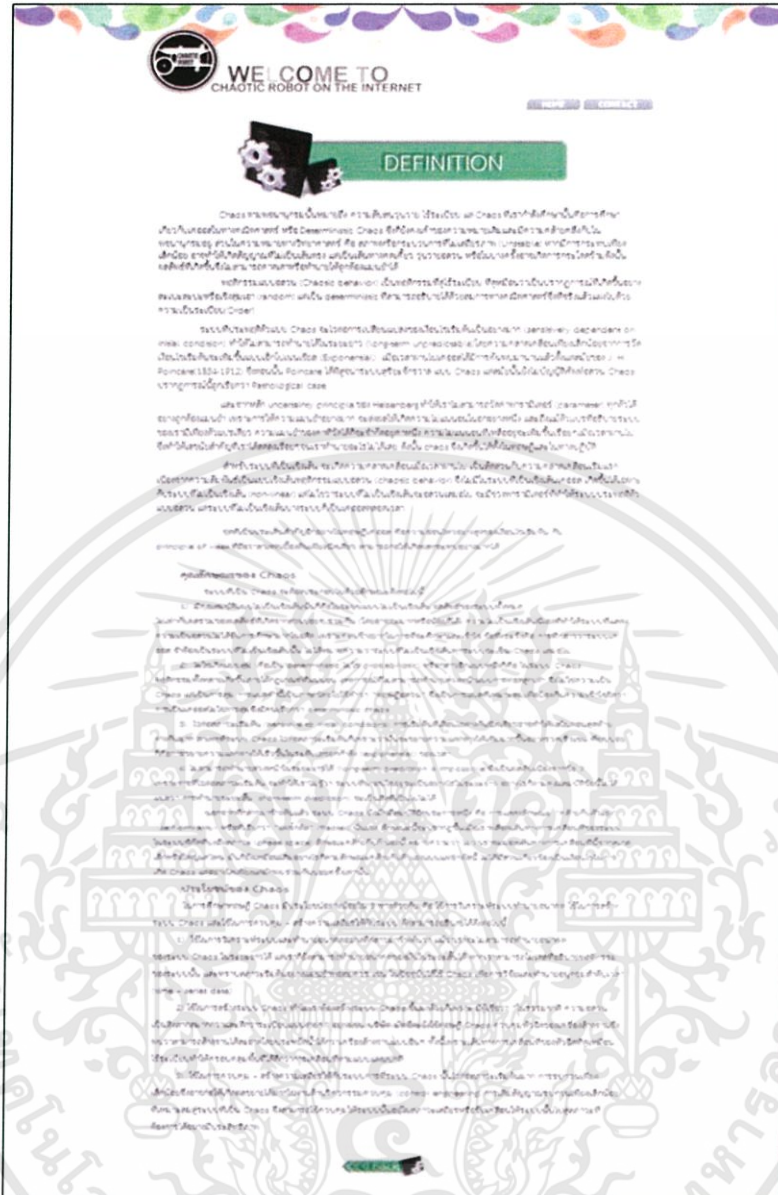
ในหน้า OVERVIEW เป็นการแสดงเนื้อหาของการทำงานโดยรวมของระบบว่าทำงานอย่างไร และมีปุ่ม back เพื่อสำหรับกดกลับมายังหน้า HOME



รูปที่ 3.16 หน้า OVERVIEW

3.5.4 หน้า DEFINITION

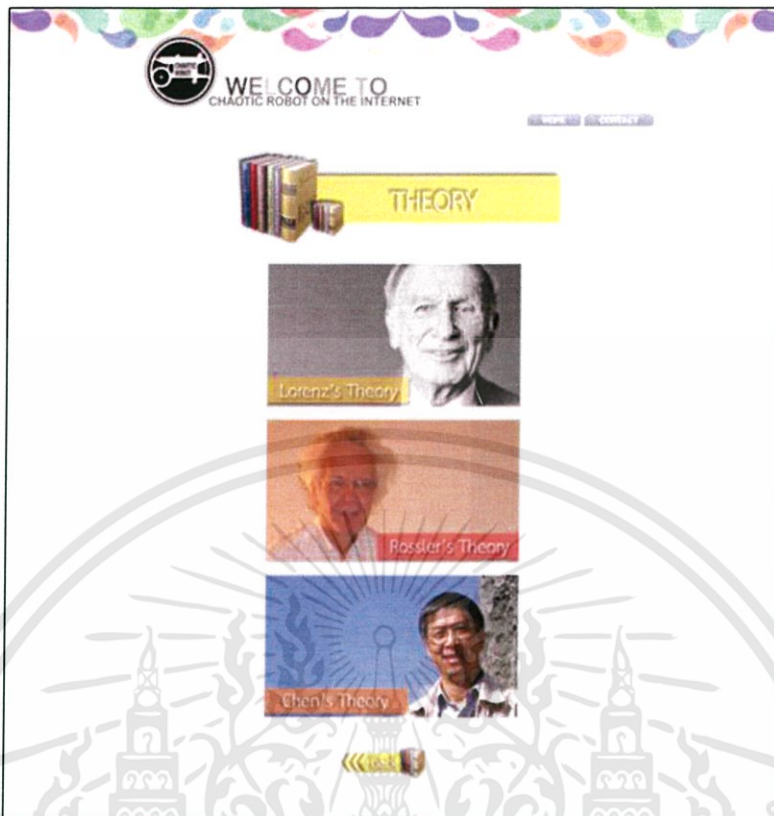
ในหน้า DEFINITION จะกล่าวถึงคำนิยามของคำว่า Chaos ว่าหมายความว่าอย่างไร และมีปุ่ม back เพื่อกดกลับมายังหน้า HOME



รูปที่ 3.17 หน้า DEFINITION

3.5.5 หน้า THEORY

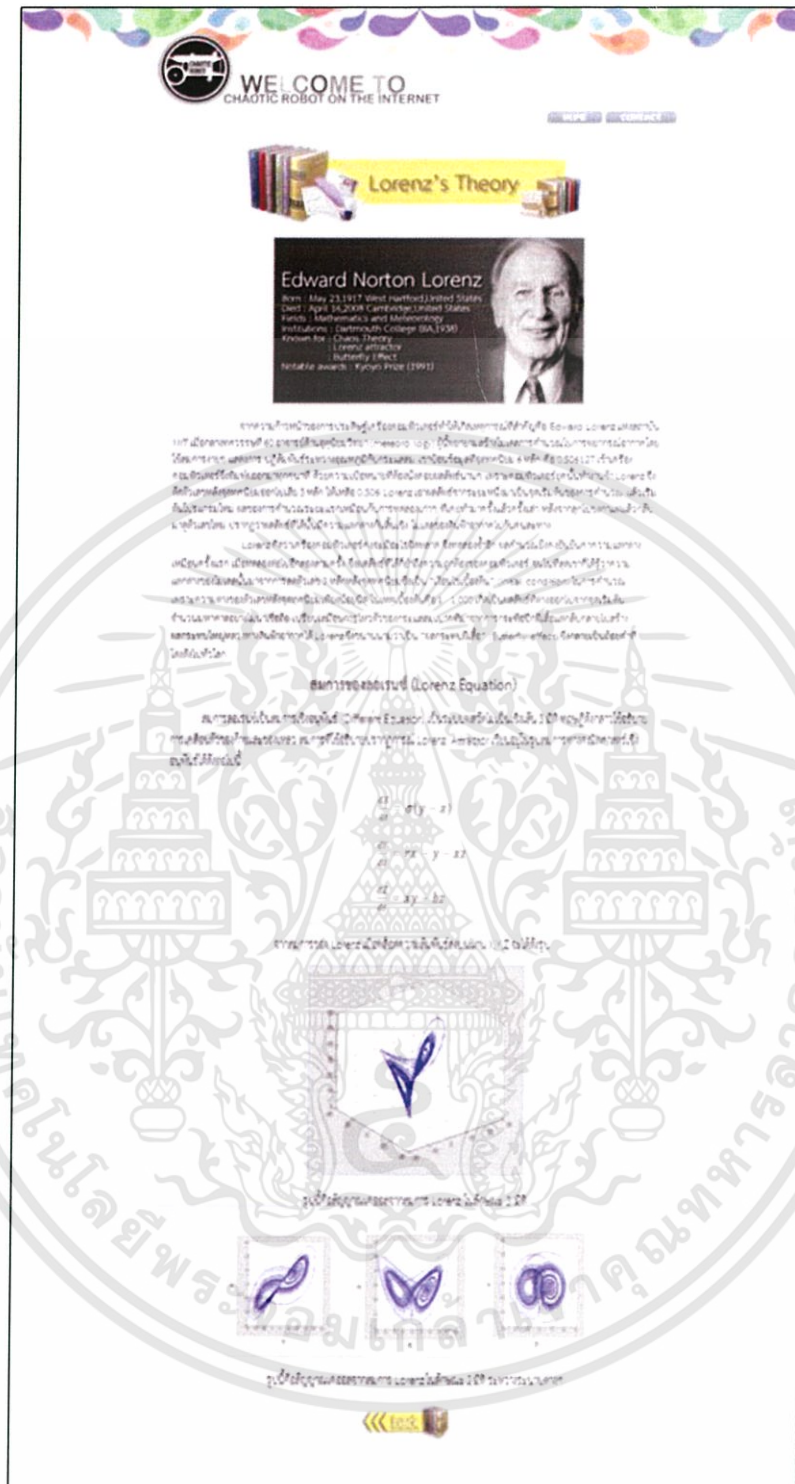
ในหน้า THEORY จะมีรูปของ 3 ศาสตราจารย์เจ้าของทฤษฎีทั้ง 3 ทฤษฎีที่เราใช้ในการทำการทดลอง เป็นลิงเพื่อให้เกิดแล้วเปลี่ยนไปยังหน้าของทฤษฎีของศาสตราจารย์คนนั้นๆ ที่เราให้ความสนใจที่จะศึกษา และมีปุ่ม back เพื่อกดกลับมายังหน้า HOME



รูปที่ 3.18 หน้า THEORY

3.5.6.1 หน้า Lorenz

เมื่อเราทำการคลิกที่รูปที่หนึ่ง จะปรากฏ Lorenz's Theory ซึ่งจะบอกรายละเอียดของสมการที่ใช้ในการทดลอง และมีปุ่ม back เพื่อกดกลับมายังหน้า THEORY



รูปที่ 3.19 หน้า Lorenz

3.5.6.2 หน้า Rossler

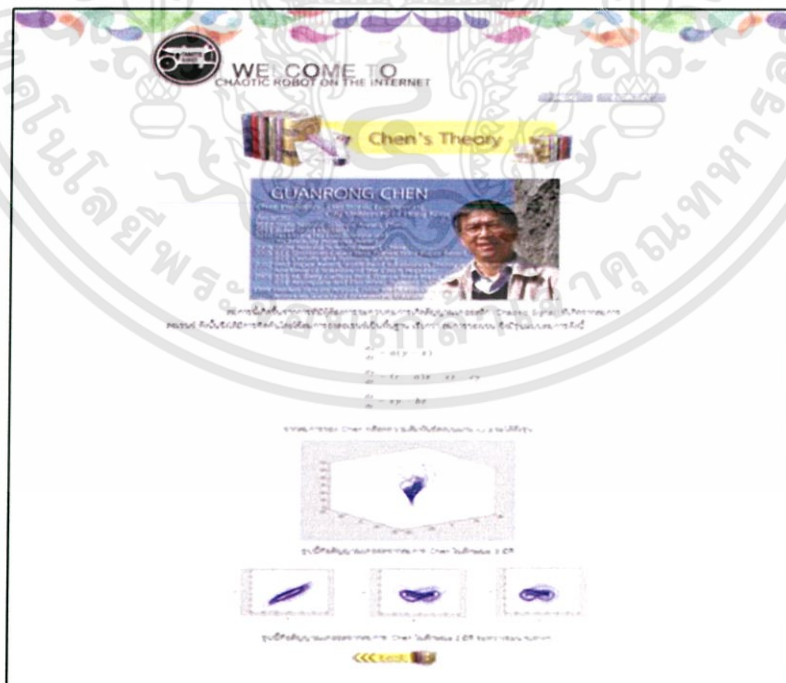
เมื่อเราทำการคลิกที่รูปที่สอง จะปรากฏ Rossler's Theory ซึ่งจะบอกรายละเอียดของสมการที่ใช้ในการทดลอง และมีปุ่ม back เพื่อกดกลับมายังหน้า THEORY



รูปที่ 3.20 หน้า Rossler

3.5.6.2 หน้า Chen

เมื่อเราทำการคลิกที่รูปที่สาม จะปรากฏ Chen's Theory ซึ่งจะบอกรายละเอียดของสมการที่ใช้ในการทดลอง และมีปุ่ม back เพื่อกดกลับมายังหน้า THEORY



รูปที่ 3.21 หน้า Chen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.6 หน้า DESIGN

ในหน้า DESIGN จะเป็นการแสดงเนื้อหาในส่วนของการออกแบบตัวหุ่นยนต์กับวงจรที่ใช้ในการทดลองและมีปุ่ม back เพื่อกดกลับมายังหน้า HOME



รูปที่ 3.22 หน้า DESIGN

3.5.7 หน้า EXPERIMENT

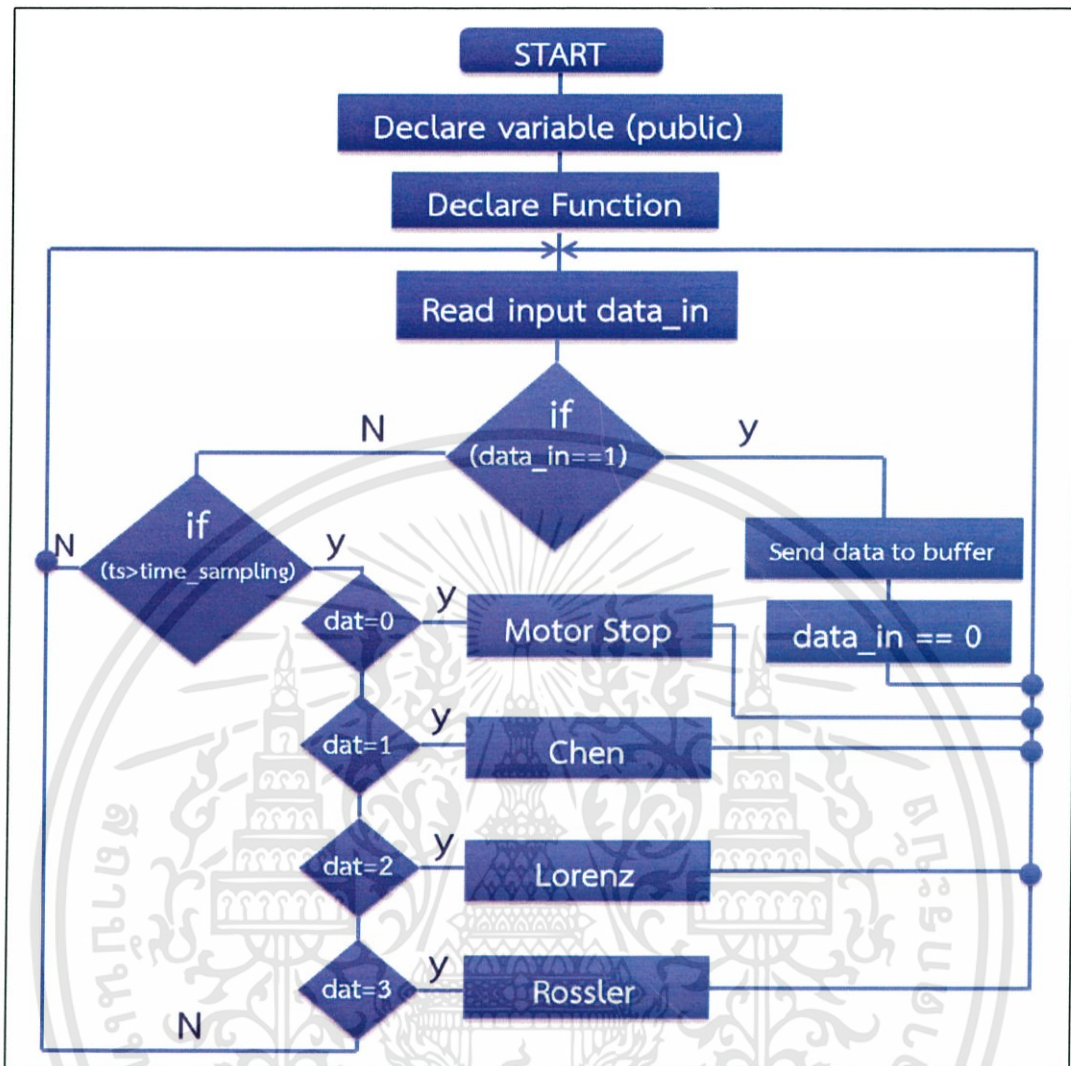
ในหน้า EXPERIMENT จะหน้าของการทดลองโดยจะให้ User เป็นคนเลือกว่าจะทดลองตามทฤษฎีของใคร เมื่อเรากดปุ่มเลือกทฤษฎีจะมีการส่งภาพ จาก IP Camera มาแสดงผลให้ได้ชม และมีปุ่ม back เพื่อกดกลับมายังหน้า HOME



รูปที่ 3.23 หน้า EXPERIMENT

3.6 การทำงานในโปรแกรม Microcontroller

เป็น Flowchart นำเสนอการขั้นตอนการทำงานใน Microcontroller ว่าหุ่นยนต์จะเดินตามแพทเทิร์นไหน เมื่อ User ทำการป้อนข้อมูลผ่านทางอินเทอร์เน็ต หรือสั่งการให้หุ่นยนต์หยุดเดิน ซึ่งสามารถศึกษา Code ที่ใช้ในการควบคุมได้อย่างละเอียดใน ภาคผนวก จ.



รูปที่ 3.24 Flowchart แสดงการทำงานของโปรแกรมใน Microcontroller

บทที่ 4

ผลการทดลอง

4.1 อุปกรณ์ที่ใช้ในการสร้างหุ่นยนต์

1. แผ่นอะคริลิก ขนาด 23 X 23	1	ชิ้น
2. แผ่นอะคริลิก ขนาด 18 X 22	1	ชิ้น
4. ล้อรถ	2	ล้อ
5. ล้อหมุนอิสระ	1	ล้อ
6. มอเตอร์	2	ตัว
7. แบตเตอรี่	2	ชิ้น
8. สวิตช์เปิดปิดมอเตอร์	1	ตัว



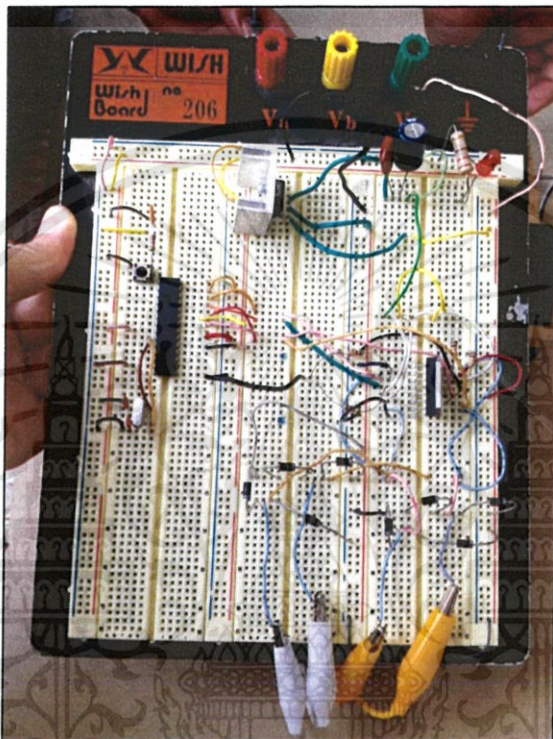
รูปที่ 4.1 ขนาดของหุ่นยนต์



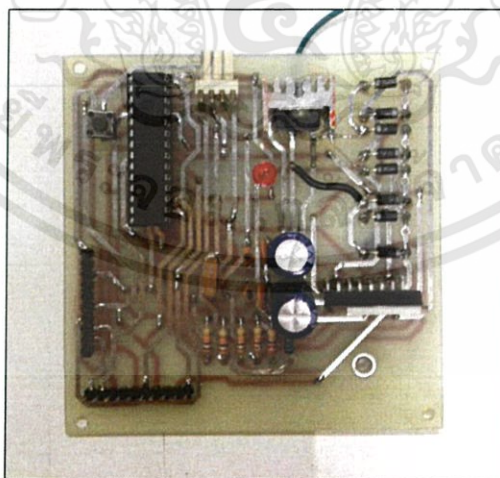
รูปที่ 4.2 ช่องสำหรับวางแบตเตอรี่

4.2 อุปกรณ์ที่ใช้ในวงจรขับมอเตอร์

1. LM7805
2. Micorcontroller Dspic30F 2010
3. L298
4. Diode
5. Resistor
6. Capacitor
7. อุปกรณ์สร้างสัญญาณ Clock

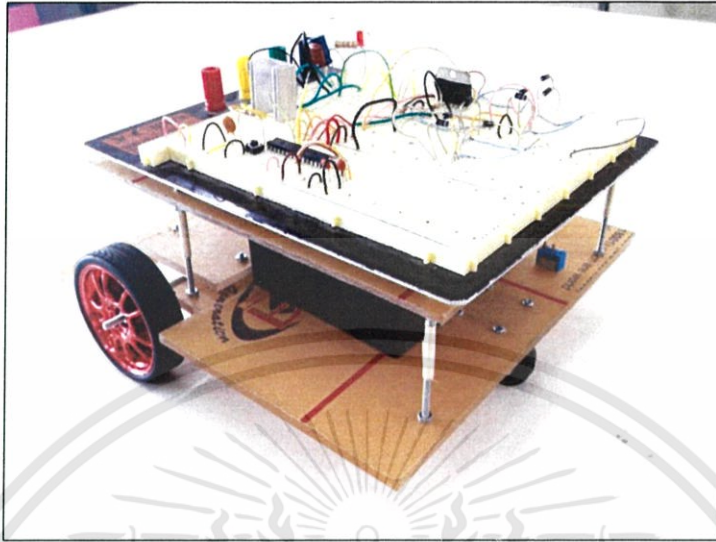


รูปที่ 4.3 รูปวงจรถับมอเตอร์ที่ทำการต่อเสร็จแล้ว

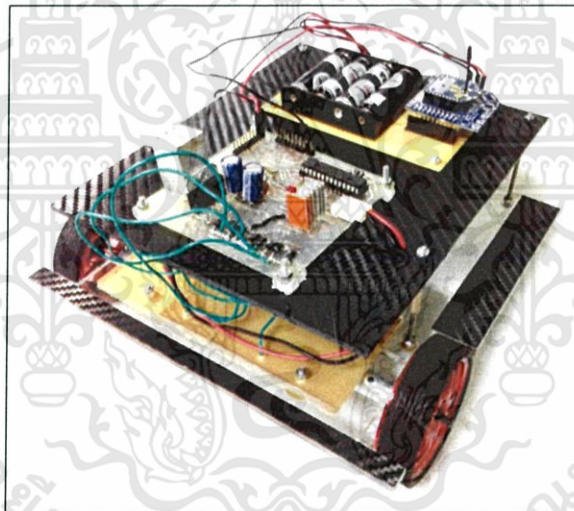


รูปที่ 4.4 รูปวงจรถับมอเตอร์ที่ทำการบัดกรีและต่อเสร็จแล้ว

4.3 การประกอบระหว่างตัวหุ่นยนต์และวงจรขับเคลื่อนมอเตอร์



รูปที่ 4.5 รูปของตัวหุ่นยนต์ที่ต่อกับวงจรขับเคลื่อนมอเตอร์



รูปที่ 4.6 รูปของตัวหุ่นยนต์ที่ต่อกับวงจรขับเคลื่อนมอเตอร์ที่ทำการบัดกรีแล้ว

4.4 ทดลองเขียนโปรแกรมตามทฤษฎี Chaotic โดยผ่านโปรแกรม MathLab

4.4.1 เขียนโปรแกรมตามทฤษฎีของ Lorenz

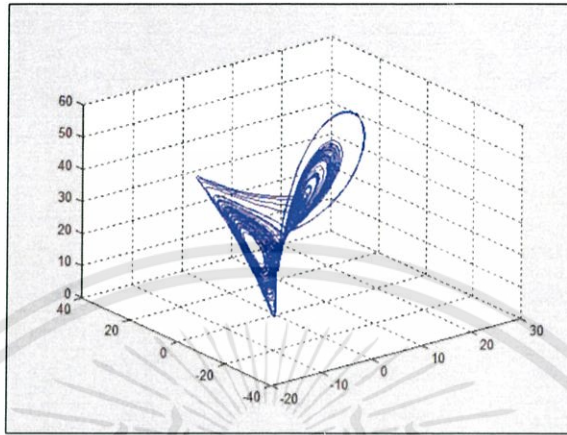
เมื่อทดลองเขียนโปรแกรมผ่านโปรแกรม MATHLAB จากสมการดังต่อไปนี้ ตามทฤษฎีของ Lorenz

$$\frac{dX}{dt} = \sigma(y - x) \quad (4.1)$$

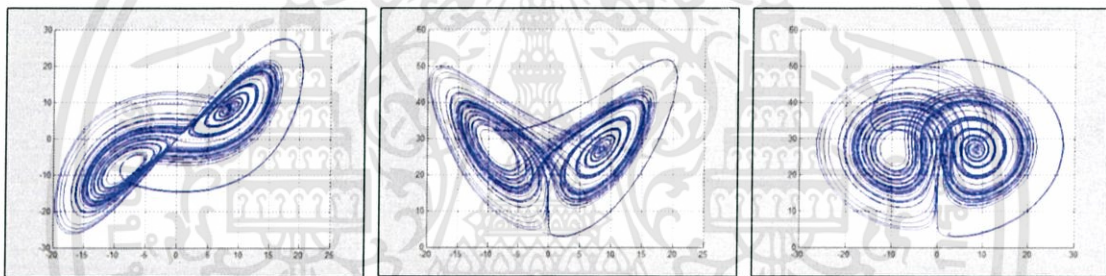
$$\frac{dY}{dt} = rx - y - xz \quad (4.2)$$

$$\frac{dZ}{dt} = xy - bz \quad (4.3)$$

จะทำให้ได้ค่า ของ $x = -1$, $y = 3$, $z = 4$
 $r = 28$, $\sigma = 10$, $b = 8/3$, $h = 0.01$
 เมื่อทำการประมวลผลจะได้รูปสัญญาณ Chaos ดังต่อไปนี้



รูปที่ 4.7 สัญญาณ Chaos จากสมการ Lorenz ในลักษณะ 3 มิติ

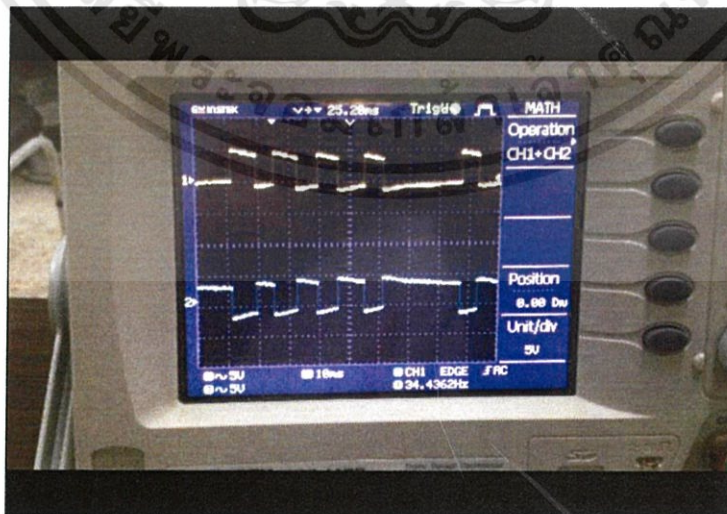


(a) (b) (c)

รูปที่ 4.8 สัญญาณเคออสจากสมการ Lorenz ในลักษณะ 2 มิติ ระหว่างระนาบต่างๆ

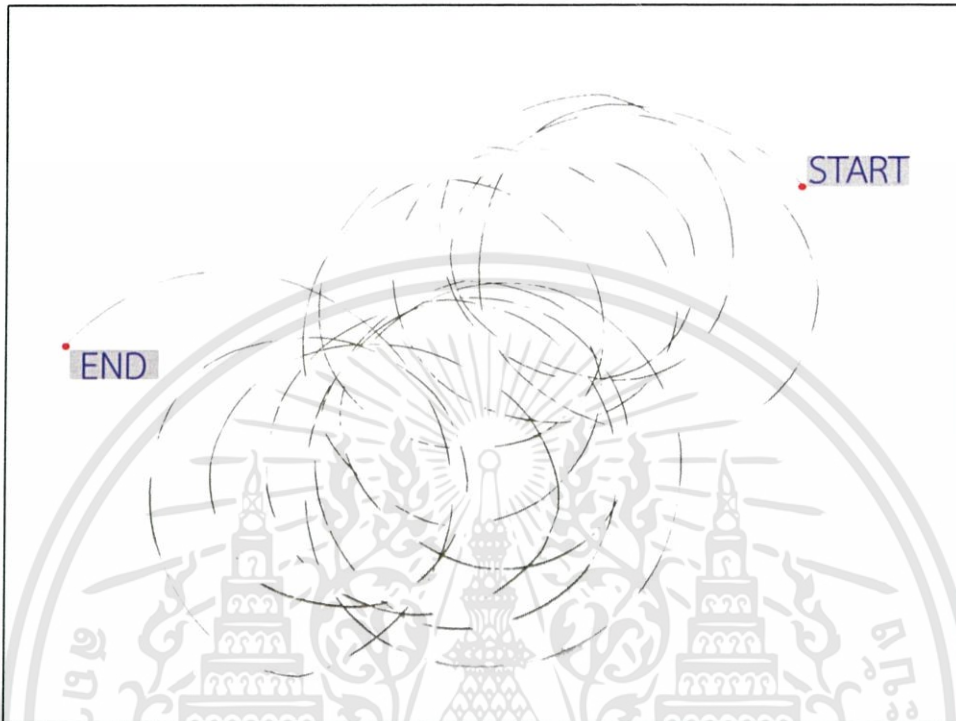
(a) ระนาบ x,y (b) ระนาบ x,z (c) ระนาบ y,z

จะเห็นว่าเป็นไปตามทฤษฎีที่กล่าวไว้ในบทที่ 2 หน้าที่ 9



รูปที่ 4.9 สโคปที่ได้จากการวัด output ของสมการ Lorenz

และเมื่อทดลองใส่โปรแกรมดังกล่าวลงไป ใน Microcontroller Dspic30F 2010 แล้ว ทดสอบโดยการปล่อยให้หุ่นยนต์เดินบนโต๊ะที่มีพื้นสีขาวแล้วนำปากกามาเสียบที่ตัวหุ่นยนต์เพื่อบันทึกเส้นทางของหุ่นยนต์แล้วพบว่า หุ่นยนต์เดินในลักษณะดังนี้



รูปที่ 4.10 ภาพการเดินของหุ่นยนต์ตามทฤษฎีของ Lorenz ในระนาบ X,Y

4.4.2 เขียนโปรแกรมตามทฤษฎีของ Chen

เมื่อทดลองเขียนโปรแกรมผ่านโปรแกรม MATLAB จากสมการดังต่อไปนี้ ตามทฤษฎีของ

Chen

$$\frac{dx}{dt} = a(y - x) \quad (4.4)$$

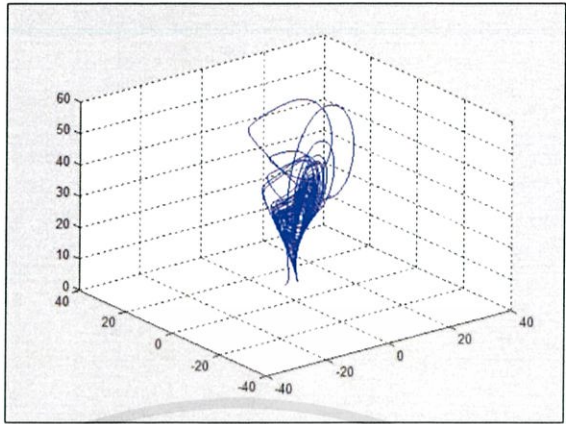
$$\frac{dy}{dt} = (c - a)x - xz - cy \quad (4.5)$$

$$\frac{dz}{dt} = xy - bz \quad (4.6)$$

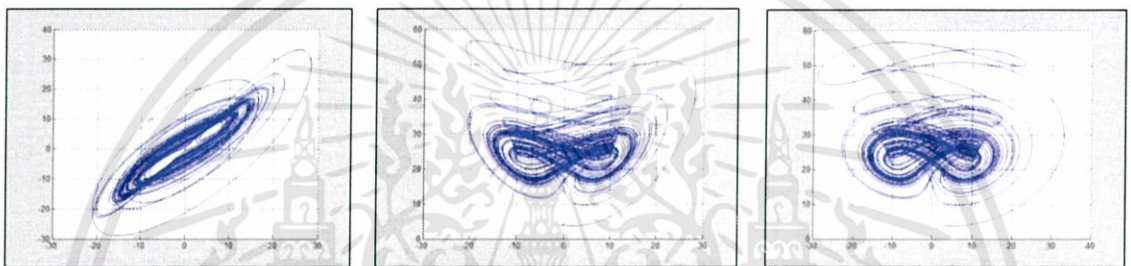
จะทำให้ได้ค่า ของ $x = -1$, $y = 3$, $z = 4$

$a = 35$, $b = 8/3$, $c = 28$, $h = 0.01$

เมื่อทำการประมวลผลจะได้รูปสัญญาณ Chaos ดังต่อไปนี้



รูปที่ 4.11 สัญญาณ Chaos จากสมการ Chen ในลักษณะ 3 มิติ



(a) (b) (c)

รูปที่ 4.12 สัญญาณเคออสจากสมการ Chen ในลักษณะ 2 มิติ ระหว่างระนาบต่างๆ

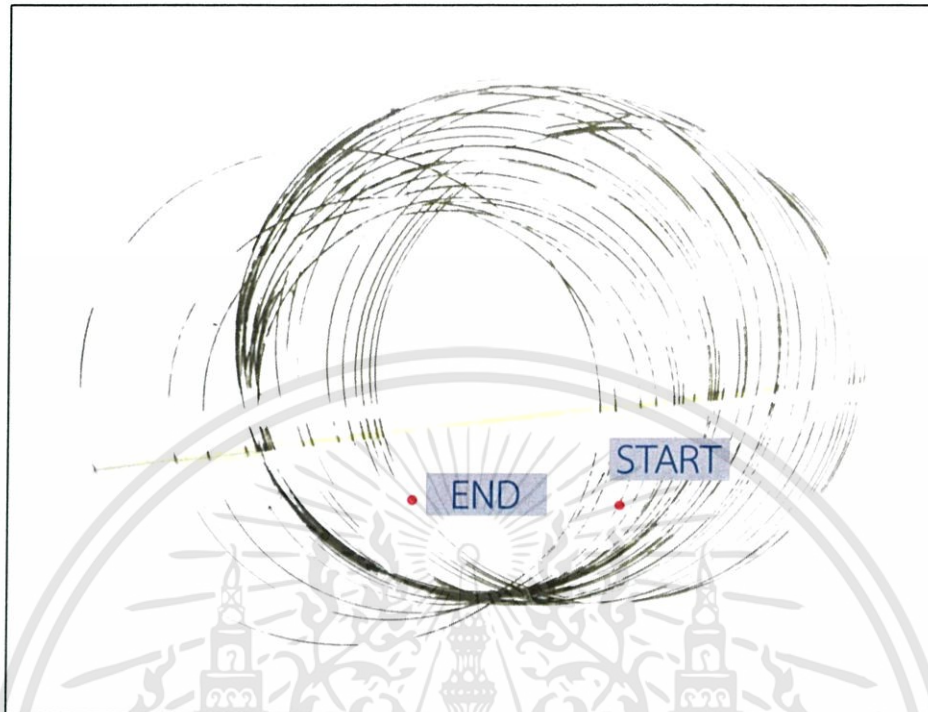
(a) ระนาบ x,y (b) ระนาบ x,z (c) ระนาบ y,z

จะเห็นว่าเป็นไปตามทฤษฎีที่กล่าวไว้ในบทที่ 2 หน้าที่ 10



รูปที่ 4.13 สโคปที่ได้จากการวัด output ของสมการ Chen

และเมื่อทดลองใส่โปรแกรมดังกล่าวลงไป ใน Microcontroller Dspic30F 2010 แล้ว ทดสอบโดยการปล่อยให้หุ่นยนต์เดินบนโต๊ะที่มีพื้นสีขาวแล้วนำปากกามาเสียบที่ตัวหุ่นยนต์เพื่อบันทึกเส้นทางของหุ่นยนต์แล้วพบว่า หุ่นยนต์เดินในลักษณะดังนี้



รูปที่ 4.14 ภาพการเดินของหุ่นยนต์ตามทฤษฎีของ Chen ในระนาบ X,Y

4.4.3 เขียนโปรแกรมตามทฤษฎีของ Rossler

เมื่อทดลองเขียนโปรแกรมผ่านโปรแกรม MATLAB จากสมการดังต่อไปนี้ ตามทฤษฎีของ

Rossler

$$\frac{dx}{dt} = -y - z \quad (4.7)$$

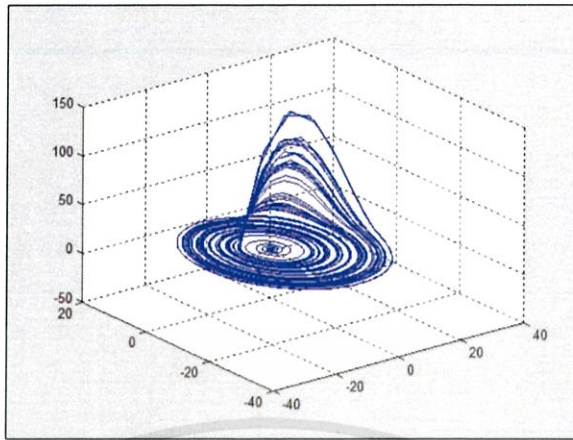
$$\frac{dy}{dt} = x - ay \quad (4.8)$$

$$\frac{dz}{dt} = a + z(x - c) \quad (4.9)$$

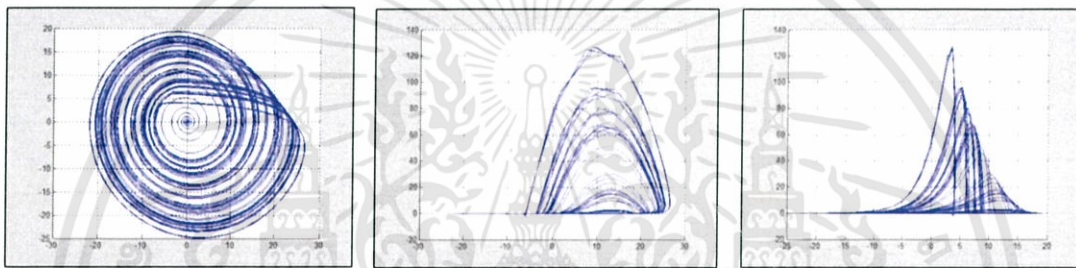
จะทำให้ได้ค่า ของ $x = 0$, $y = 0$, $z = 0$

$a = 0.1$, $b = 0.1$, $c = 14$, $h = 0.06$

เมื่อทำการประมวลผลจะได้รูปสัญญาณ Chaos ดังต่อไปนี้



รูปที่ 4.15 สัญญาณ Chaos จากสมการ Rossler ในลักษณะ 3 มิติ



(a)

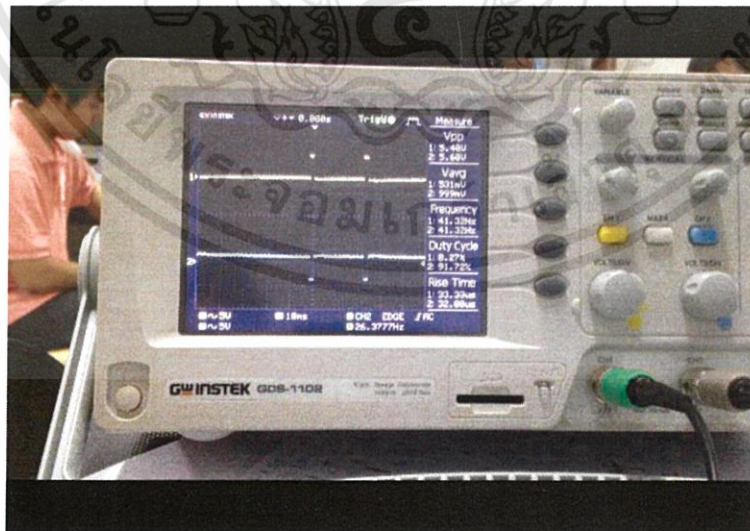
(b)

(c)

รูปที่ 4.16 สัญญาณเคออสจากสมการ Rossler ในลักษณะ 2 มิติ ระหว่างระนาบต่างๆ

(a) ระนาบ x,y (b) ระนาบ x,z (c) ระนาบ y,z

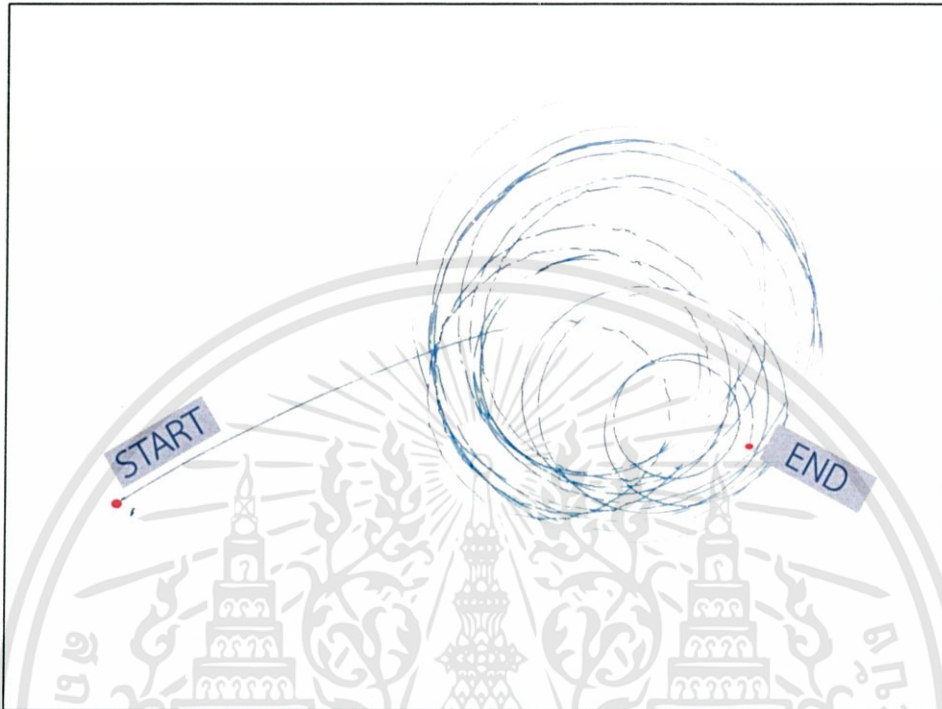
จะเห็นว่าเป็นไปตามทฤษฎีที่กล่าวไว้ในบทที่ 2 หน้าที่ 10



รูปที่ 4.17 สโคปที่ได้จากการวัด output ของสมการ Rossler

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ 38 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมื่อทดลองใส่โปรแกรมดังกล่าวลงใน Microcontroller Dspic30F 2010 แล้ว ทดสอบโดยการปล่อยให้หุ่นยนต์เดินบนโต๊ะที่มีพื้นสีขาวแล้วนำปากกามาเสียบที่ตัวหุ่นยนต์เพื่อบันทึกเส้นทางของหุ่นยนต์แล้วพบว่า หุ่นยนต์เดินในลักษณะดังนี้



รูปที่ 4.18 ภาพการเดินของหุ่นยนต์ตามทฤษฎีของ Chen ในระนาบ X,Y

4.5 การทดลองส่งข้อมูลแบบไร้สาย

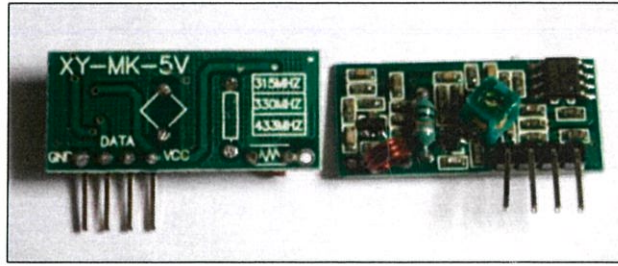
เป็นการใช้งานโมดูล UART เพื่อใช้ในการรับส่งข้อมูล ซึ่งในที่นี้ input ก็คือ แพทเทิร์น ที่ User ทำการเลือกแล้ว output ก็คือ สมการอลวน โดยการส่งข้อมูลนั้นจะทำการส่งผ่าน USB to Serial ด้วยโปรแกรม Hyperterminal ซึ่งผลการทดลองที่ได้รับคือ สามารถสั่งการให้หุ่นยนต์เคลื่อนที่ได้จริง

หลังจากนั้นจึงนำไปพัฒนาต่อ โดยการส่งข้อมูลผ่าน PHP ต่อไป ซึ่งผลหลังจากทดลองส่งผ่าน Web-app ก็ยังสามารถสั่งการให้หุ่นยนต์เคลื่อนที่ได้จริง เช่นกัน

4.6 การทดลองส่งข้อมูลแบบไร้สาย

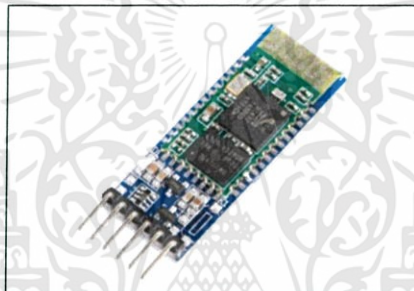
หลังจากทำการทดลองแบบไร้สายแล้ว จึงทำการพัฒนาในรูปแบบของการทำงานแบบไร้สาย โดยการใช้ RF Module ในการรับส่งข้อมูล

ในตอนแรกทำการทดลองใช้ Module XY-MK-5V ในการรับส่งข้อมูล เมื่อทำการทดลองพบว่าเกิดปัญหา สัญญาณรบกวนมาก ทำให้หุ่นยนต์เคลื่อนที่แบบติดขัด จึงสรุปได้ว่าไม่สามารถใช้งานได้อย่างมีประสิทธิภาพ



รูปที่ 4.19 ภาพของ Module XY-MK-5V ที่ใช้ในการทดลอง

ต่อมาทำการทดลองโดยใช้การสื่อสารผ่าน Bluetooth ซึ่งพบว่า หากสั่งการผ่าน Hyperterminal จะสามารถทำงานได้ปกติ โดยไม่มีสัญญาณรบกวน หุ่นยนต์รับคำสั่งและเคลื่อนที่ได้ อย่างไม่ติดขัด หลังจากนั้นจึงทำการทดลองส่งค่าผ่าน Web-app พบว่าเกิดปัญหาสัญญาณไม่สามารถติดต่อผ่าน Com port ได้จึงสรุปได้ว่า Bluetooth นั้นไม่เหมาะสมจะนำมาใช้งานไร้สายกับโครงการนี้



รูปที่ 4.20 ภาพของ Bluetooth ที่ใช้ในการทดลอง

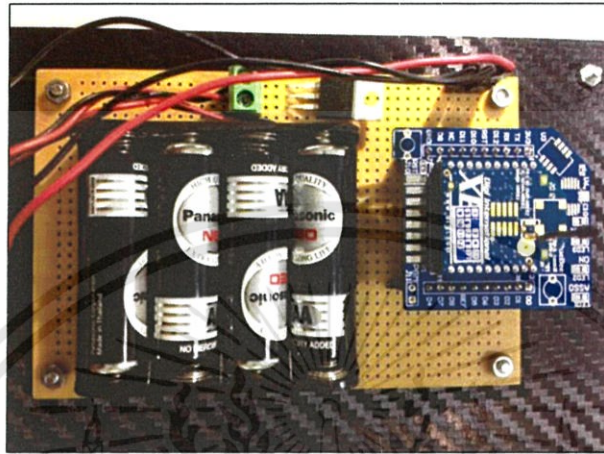
ต่อมาจึงทำการทดลองอีกครั้งโดยใช้อุปกรณ์คือ XBee ซึ่งประกอบด้วยอุปกรณ์ทั้งในภาครับ และภาคส่ง ในภาคส่งนั้นจะนำไปเชื่อมต่อกับ USB to Serial เพื่อรับข้อมูล (แพทเทิร์น) จาก User และในภาครับจะนำไปติดเข้ากับตัวหุ่นยนต์เพื่อใช้ในการรับข้อมูล (แพทเทิร์น) แล้วสั่งการให้หุ่นยนต์เคลื่อนที่ เมื่อทำการทดลองเสร็จจึงพบว่า สามารถสั่งการผ่านได้ทั้ง Hyperterminal และ Web-app หุ่นยนต์รับคำสั่งได้อย่างถูกต้องและเคลื่อนที่ได้อย่างไม่ติดขัด สัญญาณก็ไม่เกิดการรบกวน จึงสรุปได้ว่าโครงการนี้ใช้อุปกรณ์ในการเชื่อมต่อแบบไร้สายคือ XBee



รูปที่ 4.21 ภาพของ XBEE ที่ใช้ในการทดลอง

4.7 วงจรการแปลงไฟจาก 5 V เป็น 3.3 V

ในการใช้ XBEE นั้นไม่สามารถจะใช้ไฟ 5 V ได้เพราะจะทำให้ XBEE พัง จึงต้องทำการแปลงไฟจาก 5 V เป็น 3.3 V ก่อนการใช้งาน จึงจำเป็นต้องทำการสร้างวงจรเพื่อแปลงไฟติดกับตัวหุ่นยนต์ เพื่อให้ใช้งาน XBEE ได้



รูปที่ 4.21 วงจรแปลงไฟจาก 5V เป็น 3.3 V

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

จุดมุ่งหมายของการพัฒนาโครงการนี้คือ การนำเอาสัญญาณอน (Chaos) มาประยุกต์ใช้ในการขับเคลื่อนหุ่นยนต์ โดยบังคับผ่าน web base เพื่อใช้ในการศึกษาสัญญาณอนของนักศึกษาชั้นปีที่ 1 ในวิชา Fundamental electronics Laboratory ซึ่งการใช้งานจะสามารถเข้าใช้งานเพื่อเรียนรู้เกี่ยวกับสัญญาณอนนั้นผ่าน Web base ได้โดยผ่านระบบอินเทอร์เน็ต โดยทำให้สามารถเกิดการเรียนรู้ได้ด้วยตนเองตลอดเวลาและทำให้เข้าใจถึงการนำไปประยุกต์ใช้งานด้านสารสนเทศได้มากขึ้น

5.2 ปัญหาที่เกิดขึ้นในการทดลอง

ในการสร้างวงจรกำเนิดสัญญาณอนแบบมีความซับซ้อนมากแบบต่างๆ ปัญหาที่พบโดยทั่วไปก็คือ ถ้าเราเอาค่าพารามิเตอร์ที่ได้จากโปรแกรมและทฤษฎีมาทำการปรับแต่งให้กับอุปกรณ์แล้ว สัญญาณที่ได้ไม่ตรงตามที่ได้จากโปรแกรม เป็นเพราะค่าอุปกรณ์ต่างๆมีความผิดพลาดเกิดขึ้น และสัญญาณอนเป็นสัญญาณที่มีความละเอียดอ่อนมากในการหาค่าพารามิเตอร์ที่เหมาะสมกับวงจร เพราะถ้าเราทำการปรับค่าพารามิเตอร์ไปเพียงเล็กน้อย ก็จะทำให้รูปแบบของสัญญาณเปลี่ยนแปลงหายไปเลย ต้องทำการปรับค่าพารามิเตอร์ใหม่ ในบางครั้งการทำวงจรมันก็ยังมีขาดประสิทธิภาพจึงทำให้เกิดไฟช็อตที่วงจรบ่อยครั้ง

ในส่วนการสร้างวงจรถ่ายสัญญาณอนที่พบคือ อุปกรณ์บางประเภทไม่ได้คุณภาพจึงทำให้ผลที่ได้ออกมาเกิดความคลาดเคลื่อน อีกทั้งเครื่องมือยังไม่มี ความเที่ยงตรงจึงเป็นปัญหาในการเก็บผลการทดลอง

ในส่วนของโปรแกรมที่ใช้ในการทดลอง เรายังคงขาดความชำนาญในการใช้โปรแกรมไม่ว่าจะเป็น MATHLAB , MPlab IDE , Protel 99SE จึงเสียเวลาในการทำเป็นอย่างมาก

ในส่วนของ RF Module ที่ใช้ในการทดลองแบบไร้สาย เราได้ทำการเปลี่ยนถึง 3 ครั้งอันเนื่องมาจาก อุปกรณ์ที่ใช้ในตอนแรกคือ Module XY-MK-5V พบว่าเกิดสัญญาณรบกวนเป็นอย่างมาก ทำให้หุ่นยนต์ที่เราต้องการให้เคลื่อนที่ไม่สามารถเคลื่อนที่ได้โดยไม่ติดขัด และอุปกรณ์ตัวที่สองคือ Bluetooth เราพบว่าเกิดปัญหาสัญญาณไม่สามารถติดต่อผ่าน Com port ได้จึงไม่เกิดการสั่งการให้หุ่นยนต์เคลื่อนที่ และสุดท้ายทางเราจึงทำการเปลี่ยนมาเป็น XBee

ในส่วนของการทดลองเก็บผลลัพธ์ของการเดินของหุ่นยนต์เราพบว่าในตอนแรกเราใช้กระดาษมาต่อกันขนาดใหญ่แล้วนำหุ่นยนต์มาเสียบปากกาไว้เพื่อบันทึกร่องรอยการเดินแล้วปล่อยให้หุ่นยนต์เดิน ปรากฏว่าเกิดปัญหาขึ้นหลายอย่างไม่ว่าจะเป็น กระดาษมีความกว้างไม่พอ หุ่นยนต์เดินล้นกระดาษไปเขียนเส้นทางเดินบนพื้นทำให้เกิดความสปรก พื้นที่ใช้ในการวางกระดาษไม่เรียบ เป็นเหตุให้การเก็บผลไม่เป็นที่น่าพอใจ เราจึงต้องปรึกษาปัญหานี้กันใหม่จึงได้ข้อสรุปว่า เปลี่ยนจากกระดาษเป็นมาทำบนโต๊ะที่มีพื้นผิวหน้าเป็นไวท์บอร์ดจะสะดวกกว่า เนื่องจากจะลบออกได้ง่ายแล้วยังไม่สิ้นเปลืองการใช้กระดาษ และเนื่องจากโต๊ะเป็นสีขาวจึงทำให้ผลที่เราต้องการค่อนข้างชัดเจนมาก แต่แน่นอนว่ากว่าเราจะได้ผลเป็นที่น่าพอใจนั้น ก็เสียเวลาและทรัพยากรไปมากด้วย



รูปที่ 5.1 ตัวอย่างปัญหาที่เกิดขึ้นในการทดลอง

5.3 แผนปฏิบัติในอนาคต

1. จัดทำระบบการสมัครสมาชิกในการเข้าใช้เว็บไซต์ เพื่อเก็บข้อมูลของผู้สนใจศึกษาทฤษฎีอลวน
2. ทำการเพิ่มรูปแบบของสมการอลวน
3. จัดทำการทดลองให้สามารถปรับค่าพารามิเตอร์ภายในสมการอลวน ผ่านระบบอินเทอร์เน็ตได้

เอกสารอ้างอิง

[1] ISCAS 2000 – IEEE International Symposium On Circuit And System , May 28-31, 2000 , Geneva , Switzerland

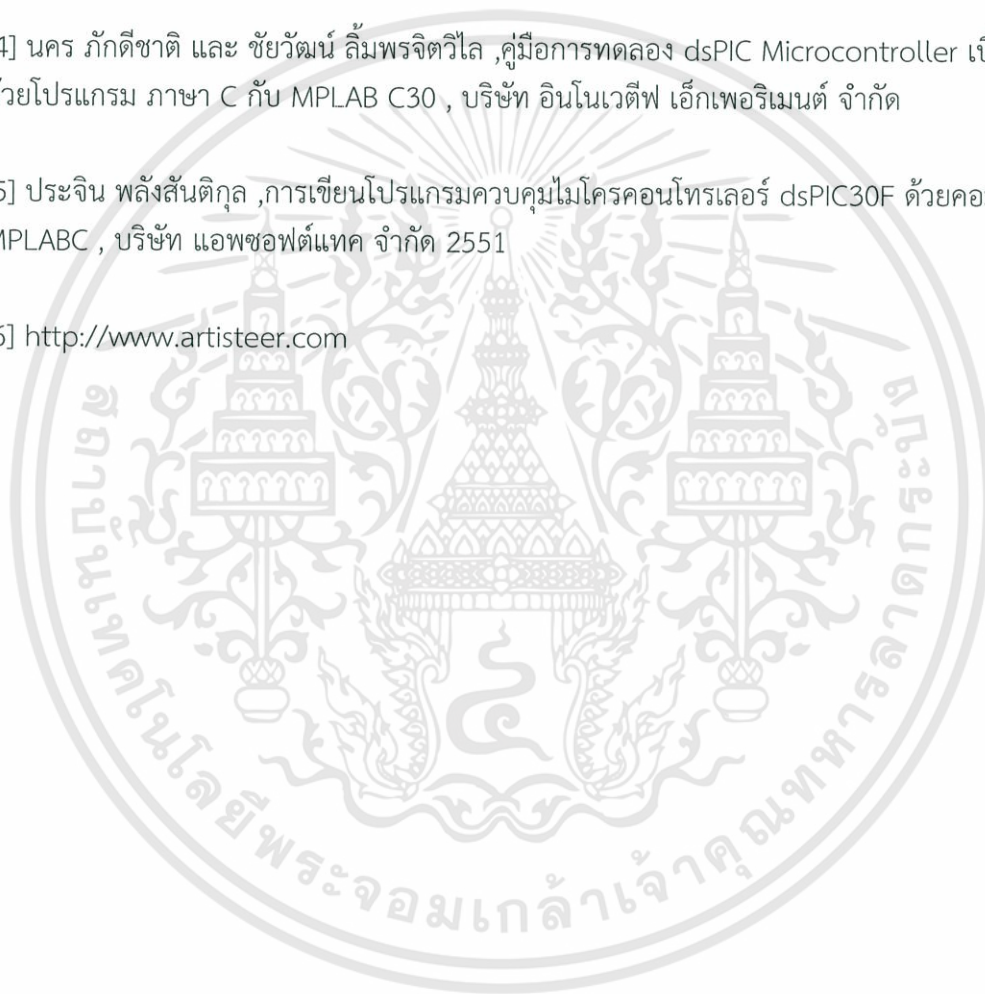
[2] International Journal of Bifurcation And Chaos , Vol. 12 ,No.1 (2000) 147-157

[3] <http://complex.upf.es/~josep/Chaos.html>

[4] นคร ภัคดีชาติ และ ชัยวัฒน์ ลิ้มพรจิตวิไล ,คู่มือการทดลอง dsPIC Microcontroller เบื้องต้น ด้วยโปรแกรม ภาษา C กับ MPLAB C30 , บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด

[5] ประจัน พลังสันติกุล ,การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ dsPIC30F ด้วยคอมพิวเตอร์ MPLABC , บริษัท แอปซอพท์เทคโนโลยี จำกัด 2551

[6] <http://www.artisteer.com>





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



LM78XX/LM78XXA 3-Terminal 1A Positive Voltage Regulator

Features

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

General Description

The LM78XX series of three terminal positive regulators are available in the TO-220 package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

Ordering Information

Product Number	Output Voltage Tolerance	Package	Operating Temperature
LM7805CT	±4%	TO-220 (Single Gauge)	-40°C to +125°C
LM7808CT			
LM7808CT			
LM7809CT			
LM7810CT			
LM7812CT			
LM7815CT			
LM7818CT			
LM7824CT			
LM7805ACT			
LM7806ACT			
LM7808ACT			
LM7809ACT			
LM7810ACT			
LM7812ACT			
LM7815ACT			
LM7818ACT			
LM7824ACT			

Block Diagram

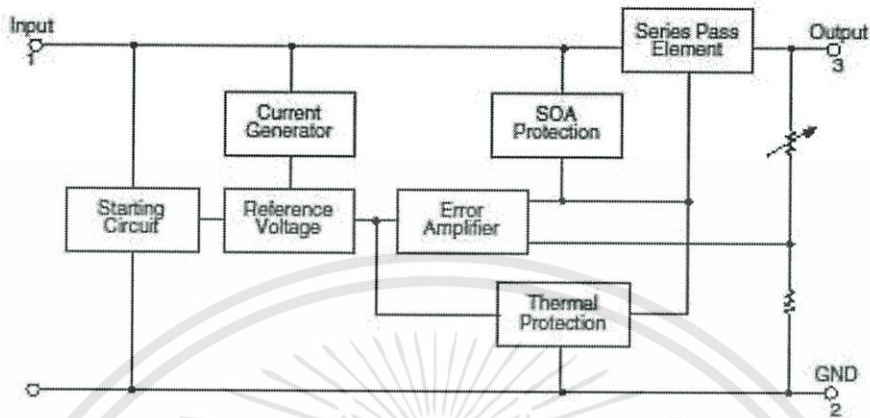


Figure 1.

Pin Assignment

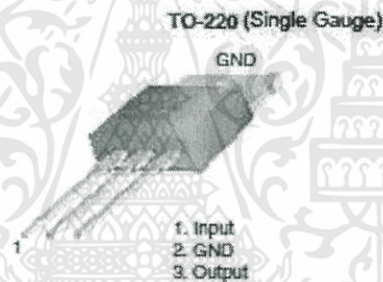


Figure 2.

Absolute Maximum Ratings

Absolute maximum ratings are those values beyond which damage to the device may occur. The datasheet specifications should be met, without exception, to ensure that the system design is reliable over its power supply, temperature, and output/input loading variables. Fairchild does not recommend operation outside datasheet specifications.

Symbol	Parameter	Value	Unit
V_I	Input Voltage	$V_O = 5V$ to $18V$	35
		$V_O = 24V$	40
$R_{\theta JC}$	Thermal Resistance Junction-Cases (TO-220)	5	$^{\circ}C/W$
$R_{\theta JA}$	Thermal Resistance Junction-Air (TO-220)	65	$^{\circ}C/W$
T_{OPR}	Operating Temperature Range	LM78xx	-40 to +125
		LM78xxA	0 to +125
T_{STG}	Storage Temperature Range	-65 to +150	$^{\circ}C$

Electrical Characteristics (LM7805)Refer to the test circuits. $-40^{\circ}\text{C} < T_J < 125^{\circ}\text{C}$, $I_O = 500\text{mA}$, $V_I = 10\text{V}$, $C_I = 0.1\mu\text{F}$, unless otherwise specified.

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit	
V_O	Output Voltage	$T_J = +25^{\circ}\text{C}$	4.8	5.0	5.2	V	
		$5\text{mA} \leq I_O \leq 1\text{A}$, $P_O \leq 15\text{W}$, $V_I = 7\text{V to } 20\text{V}$	4.75	5.0	5.25		
Regline	Line Regulation ⁽¹⁾	$T_J = +25^{\circ}\text{C}$	$V_O = 7\text{V to } 25\text{V}$	–	4.0	100	mV
			$V_I = 8\text{V to } 12\text{V}$	–	1.6	50.0	
Regload	Load Regulation ⁽¹⁾	$T_J = +25^{\circ}\text{C}$	$I_O = 5\text{mA to } 1.5\text{A}$	–	9.0	100	mV
			$I_O = 250\text{mA to } 750\text{mA}$	–	4.0	50.0	
I_Q	Quiescent Current	$T_J = +25^{\circ}\text{C}$	–	5.0	8.0	mA	
ΔI_Q	Quiescent Current Change	$I_O = 5\text{mA to } 1\text{A}$ $V_I = 7\text{V to } 25\text{V}$	–	0.03	0.5	mA	
			–	0.3	1.3		
$\Delta V_O/\Delta T$	Output Voltage Drift ⁽²⁾	$I_O = 5\text{mA}$	–	–0.8	–	mV/ $^{\circ}\text{C}$	
V_N	Output Noise Voltage	$f = 10\text{Hz to } 100\text{kHz}$, $T_A = +25^{\circ}\text{C}$	–	42.0	–	$\mu\text{V}/V_O$	
RR	Ripple Rejection ⁽²⁾	$f = 120\text{Hz}$, $V_O = 8\text{V to } 18\text{V}$	62.0	73.0	–	dB	
V_{DROP}	Dropout Voltage	$I_O = 1\text{A}$, $T_J = +25^{\circ}\text{C}$	–	2.0	–	V	
r_O	Output Resistance ⁽²⁾	$f = 1\text{kHz}$	–	15.0	–	m Ω	
I_{SC}	Short Circuit Current	$V_I = 35\text{V}$, $T_A = +25^{\circ}\text{C}$	–	230	–	mA	
I_{PK}	Peak Current ⁽²⁾	$T_J = +25^{\circ}\text{C}$	–	2.2	–	A	

Notes:

1. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.
2. These parameters, although guaranteed, are not 100% tested in production.



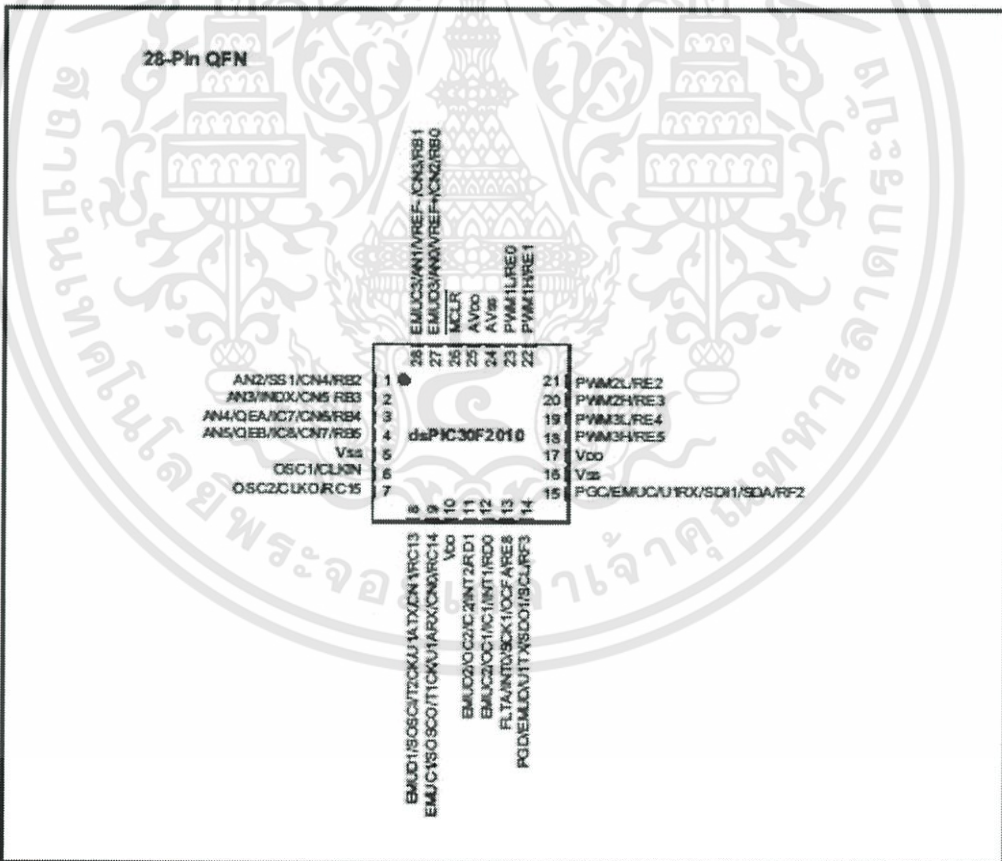
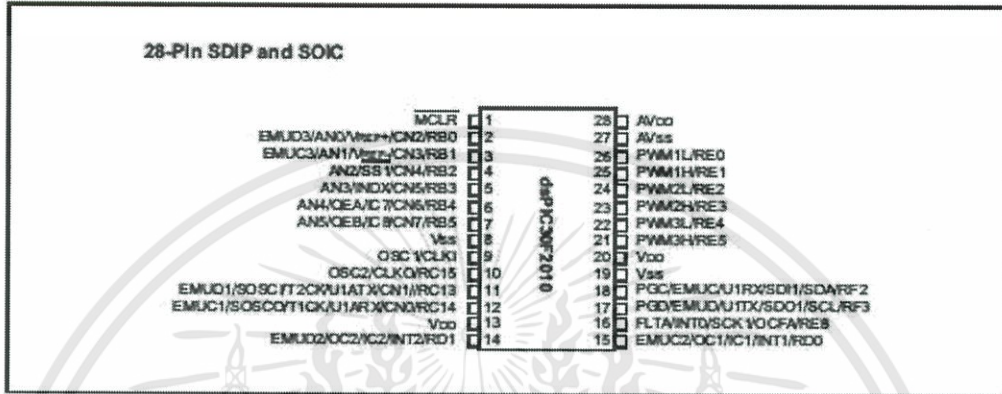


MICROCHIP

dsPIC30F2010
Data Sheet

28-pin High-Performance
Digital Signal Controllers

Pin Diagrams





ภาคผนวก ค.

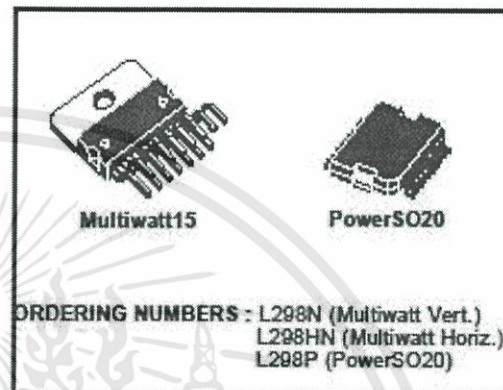
L298

DUAL FULL-BRIDGE DRIVER

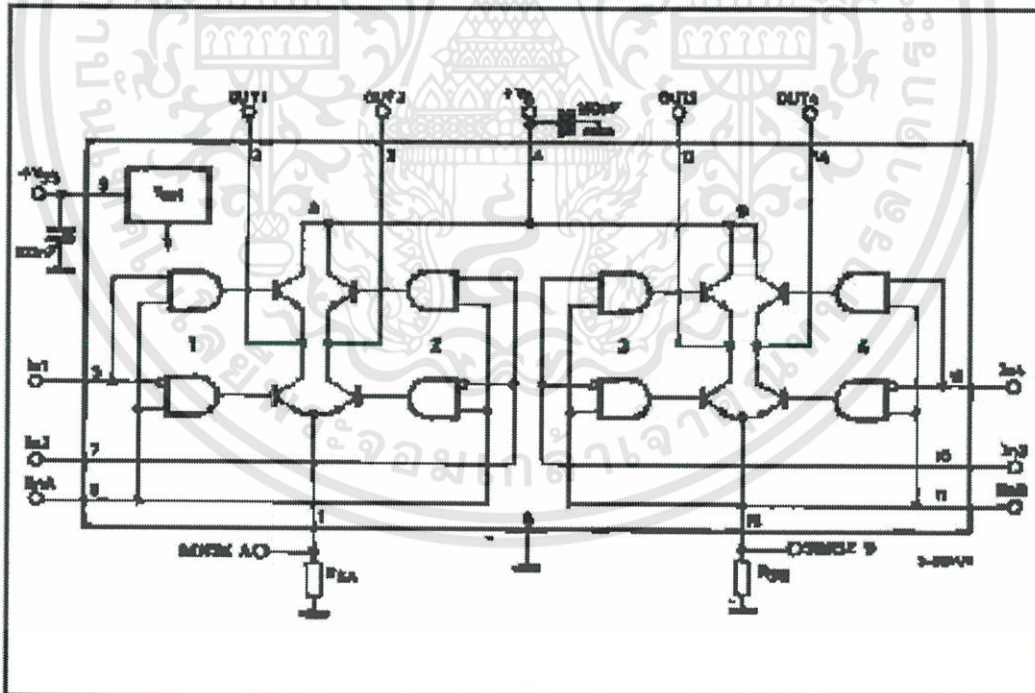
- OPERATING SUPPLY VOLTAGE UP TO 48 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



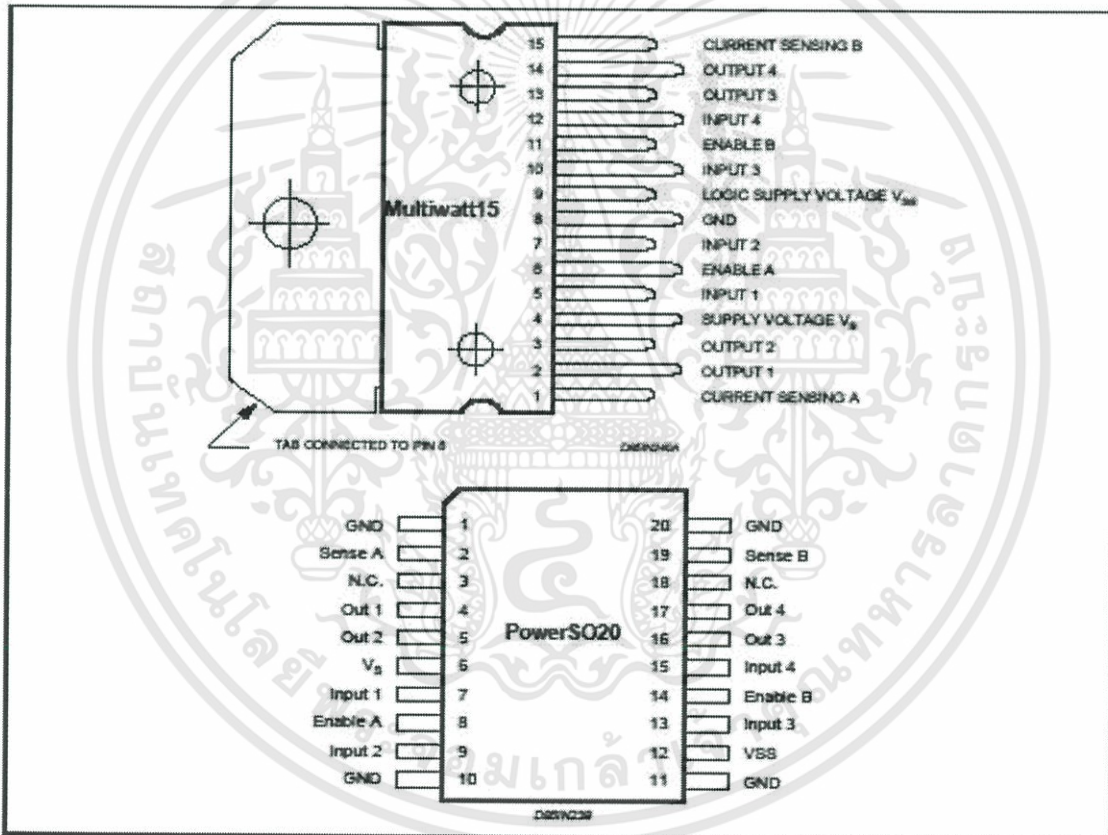
nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM


ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{GS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_o	Peak Output Current (each Channel)		
	- Non Repetitive ($t = 100\mu s$)	3	A
	- Repetitive (80% on -20% off, $t_{on} = 10ms$)	2.5	A
	- DC Operation	2	A
V_{sense}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th(j-case)}$	Thermal Resistance Junction-case	Max.	-	3	$^\circ C/W$
$R_{th(j-amb)}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(*) Mounted on aluminum substrate

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A 100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_J = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{EH} +2.5		46	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0 V _I = L V _I = H		13 50	22 70	mA mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = L V _{en} = H; I _L = 0 V _I = L V _I = H V _{en} = L V _I = X		24 7	36 12 6	mA mA mA
V _L	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V _H	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _L	Low Voltage Input Current (pins 5, 7, 10, 12)	V _I = L			-10	μA
I _H	High Voltage Input Current (pins 5, 7, 10, 12)	V _I = H ≤ V _{SS} - 0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			-10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} - 0.6V		30	100	μA
V _{CEsat (H)}	Source Saturation Voltage	I _L = 1A I _L = 2A	0.95	1.35 2	1.7 2.7	V V
V _{CEsat (L)}	Sink Saturation Voltage	I _L = 1A (5) I _L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V _{CEsat}	Total Drop	I _L = 1A (5) I _L = 2A (5)	1.80		3.2 4.9	V V
V _{sense}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V



Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

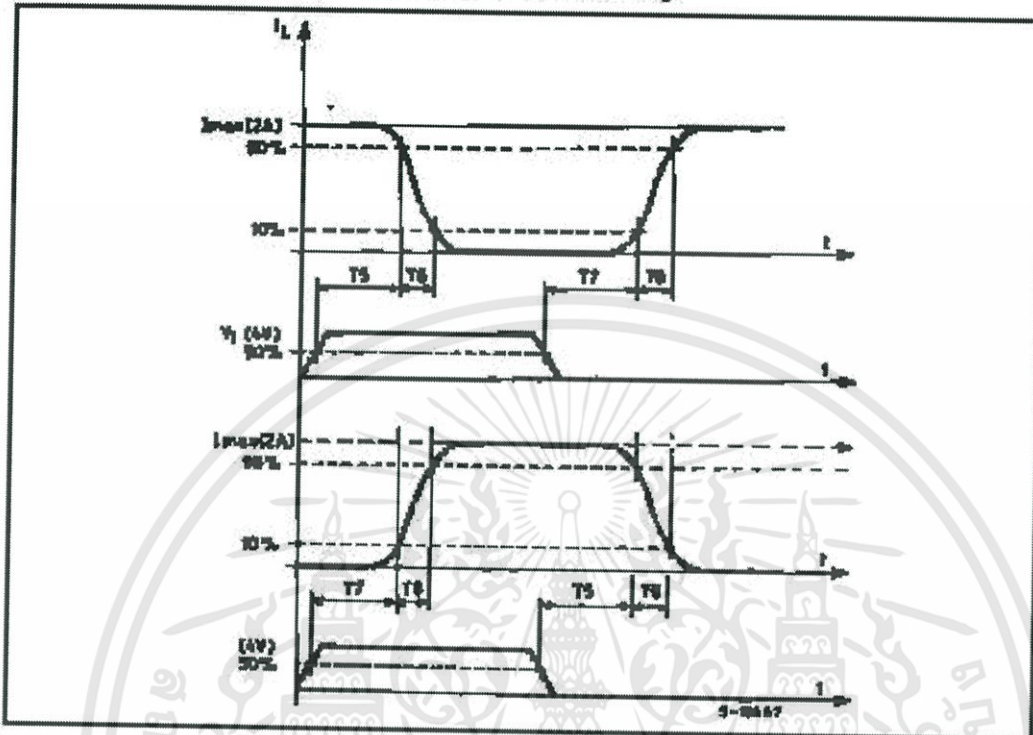
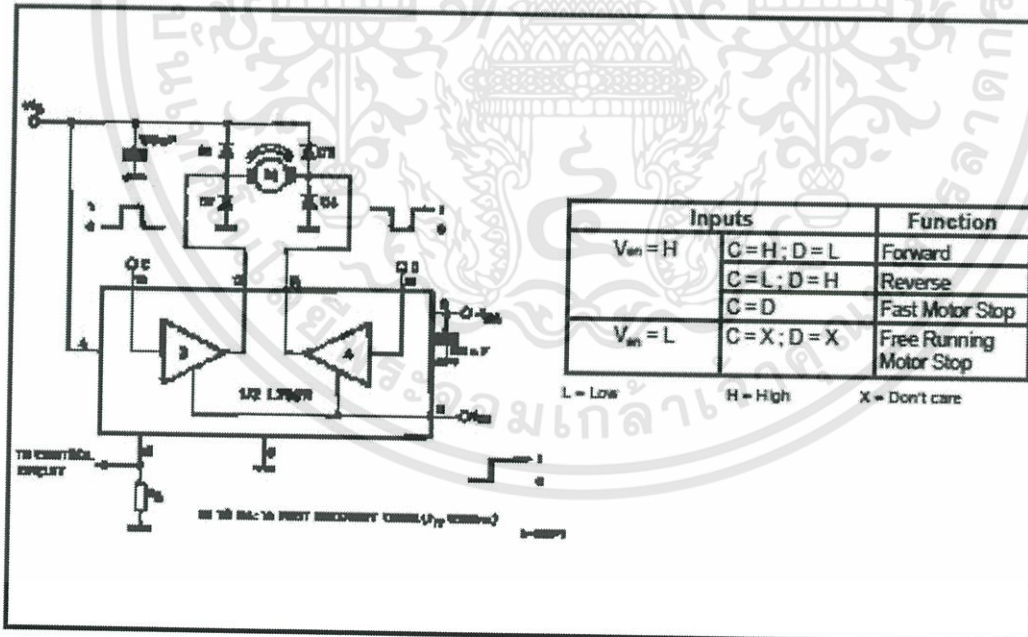


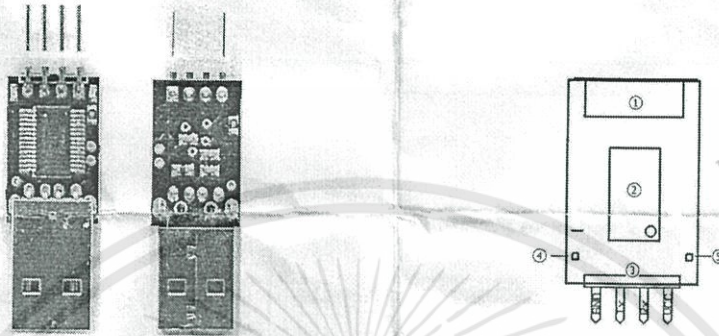
Figure 6 : Bidirectional DC Motor Control.







คู่มือการใช้งาน K-USB 2 Serial



คุณสมบัติบอร์ด

1. เป็นบอร์ดที่ไว้จำลอง พอร์ตเชื่อมต่ออนุกรม (Serial Port) ในรูป VCP (Virtual COM Port)
2. ง่ายสำหรับติดตั้งระหว่างไมโครคอนโทรลเลอร์ กับ คอมพิวเตอร์ โดยใช้โมดูล USART ของ ไมโครคอนโทรลเลอร์ได้
3. สัญญาณ Output เป็นแบบ TTL ปรับแรงดันตาม แรงดันของไมโครคอนโทรลเลอร์ปลายทาง โดยจ่าย เข้าขา VTG
4. สามารถติดต่อกับไมโครคอนโทรลเลอร์ได้โดยตรง ต่าง จากพอร์ต RS232 ที่ต้องมีตัวปรับแรงดันของสัญญาณ
5. ใช้แรงดันจาก USB 5V ในการเลี้ยง บอร์ด
6. ใช้ชิพ USB Bridge ของ FTDI รุ่น FT232R
7. สามารถใช้ Baudrate ได้สูงถึง 921600 bps

โครงสร้างบอร์ด K-USB 2 Serial

1. หัว USB
2. ตัวชิพ FT232RL
3. พอร์ต Serial ในรูปแบบ คอนเน็คเตอร์ 4 พิน
4. LED แสดงสถานะ รับค่า Rx
5. LED แสดงสถานะ ส่งค่า Tx

หมายเหตุ

1. ขา VTG จำเป็นต้องจ่ายแรงดันจากบอร์ดปลายทางเข้า มา เนื่องจากส่วนให้ FT232R ปรับแรงดัน I/O ให้ เท่ากับ แรงดันปลายทาง
2. หากการส่งข้อมูลด้วยความเร็วต่ำ จำนวนมาก ส่งกด ไฟแสดงสถานะจะยังกะพริบต่อ หลังจากหยุดส่ง ข้อมูลไปแล้วได้ ไม่ถือว่ามีผิดปกติ



โค้ดที่ใช้ในการเดินของหุ่นยนต์ตามทฤษฎี Lorenz

```
#include<p30f2010.h>
#include<timer.h>
#define LED    LATDbits.LATD0
#define L298_M2_2  LATEbits.LATE3
#define L298_M2_1  LATEbits.LATE4
#define L298_M1_2  LATEbits.LATE1
#define L298_M1_1  LATEbits.LATE0
#define time_sampling 1
char ts=0;
float x=1.0,y=1.0,z=1.0;
void _ISR_T1Interrupt(void)
{
    IFS0bits.T1IF = 0;
    ts++;
}
void open_timer_service(void)
{
    ConfigntTimer1(T1_INT_PRIOR_2 & T1_INT_ON);
    WriteTimer1(0);
    unsigned int TimeInterval = 1000;
    OpenTimer1(T1_ON & T1_GATE_OFF &
        T1_IDLE_STOP & T1_PS_1_1 &
        T1_SYNC_EXT_OFF & T1_SOURCE_INT,
        TimeInterval);
}
void init(void)
{
    TRISEbits.TRISE0 = 0;
    TRISEbits.TRISE1 = 0;
    TRISEbits.TRISE4 = 0;
    TRISEbits.TRISE3 = 0;
    open_timer_service();
}
float fx(float x,float y)
{
    return(10.0*(y-x));
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float fy(float x,float y,float z)
{
    return((28.0*x)-y-(x*z));
}
float fz(float x,float y,float z)
{
    return((x*y)-(2.66666667*z));
}
void Euler()
{ float h=0.01;
  x=x+(fx(x,y))*h;
  y=y+(fy(x,y,z))*h;
  z=z+(fz(x,y,z))*h;
}
int main(void)
{
    init();
    unsigned int x_buf1=0,y_buf1=0,z_buf1=0;
    int x_buf2=0,y_buf2=0,z_buf2=0;
    float x_buf3=0,y_buf3=0,z_buf3=0;
    while(1)
    {
        if(ts>time_sampling)
        {
            ts=0;
            Euler();
            x_buf3=(x+30)*40;
            y_buf3=(y+50)*40;
            z_buf3=(z+0)*40;
            x_buf2=x_buf3;
            y_buf2=y_buf3;
            z_buf2=z_buf3;
            x_buf1=x_buf2;
            y_buf1=y_buf2;
            z_buf1=z_buf2;
            if(x_buf1>1440)
            {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และตั้งชื่ออ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

L298_M1_1 =1;
L298_M1_2 =0;
}
else
{
L298_M1_1 =0;
L298_M1_2 =1;
}
if(y_buf1>2400)
{
L298_M2_1 =1;
L298_M2_2 =0;
}
else
{
L298_M2_1 =0;
L298_M2_2 =1;
}
}
}

```



โค้ดที่ใช้ในการเดินของหุ่นยนต์ตามทฤษฎี Rossler

```
#include<p30f2010.h>
#include<timer.h>
#define LED    LATDbits.LATD0
#define L298_M2_2  LATEbits.LATE3
#define L298_M2_1  LATEbits.LATE4
#define L298_M1_2  LATEbits.LATE1
#define L298_M1_1  LATEbits.LATE0
#define time_sampling 1
char ts=0;
float x=1.0,y=1.0,z=1.0;
void _ISR_T1Interrupt(void)
{
    IFS0bits.T1IF = 0;
    ts++;
}
void open_timer_service(void)
{
    ConfigIntTimer1(T1_INT_PRIOR_2 & T1_INT_ON);
    WriteTimer1(0);
    unsigned int TimeInterval = 1000;
    OpenTimer1(T1_ON & T1_GATE_OFF &
        T1_IDLE_STOP & T1_PS_1_1 &
        T1_SYNC_EXT_OFF & T1_SOURCE_INT,
        TimeInterval);
}
void init(void)
{
    TRISEbits.TRISE0 = 0;
    TRISEbits.TRISE1 = 0;
    TRISEbits.TRISE4 = 0;
    TRISEbits.TRISE3 = 0;
    open_timer_service();
}
float fx(float x,float y,float z)
{
    return((-y)-z);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float fy(float x,float y,float z)
{
    return(x+(0.1*y));
}
float fz(float x,float y,float z)
{
    return(0.1+((x-14.0)*z));
}
void Euler()
{ float h=0.006;
  x=x+(fx(x,y,z))*h;
  y=y+(fy(x,y,z))*h;
  z=z+(fz(x,y,z))*h;
}
int main(void)
{
    init();
    unsigned int x_buf1=0,y_buf1=0,z_buf1=0;
    int x_buf2=0,y_buf2=0,z_buf2=0;
    float x_buf3=0,y_buf3=0,z_buf3=0;
    while(1)
    {
        if(ts>time_sampling)
        {
            ts=0;
            Euler();
            x_buf3=(x+40)*40;
            y_buf3=(y+30)*40;
            z_buf3=(z+20)*40;
            x_buf2=x_buf3;
            y_buf2=y_buf3;
            z_buf2=z_buf3;
            x_buf1=x_buf2;
            y_buf1=y_buf2;
            z_buf1=z_buf2;

            if(x_buf1>960)

```

```

    {
    L298_M1_1 =1;
    L298_M1_2 =0;
    }
    else
    {
    L298_M1_1 =0;
    L298_M1_2 =1;
    }
    if(y_buf1>600)
    {
    L298_M2_1 =1;
    L298_M2_2 =0;
    }
    else
    {
    L298_M2_1 =0;
    L298_M2_2 =1;
    }
    }
}

```



โค้ดที่ใช้ในการเดินของหุ่นยนต์ตามทฤษฎี Chen

```
#include<p30f2010.h>
#include<timer.h>
#define LED    LATDbits.LATD0
#define L298_M2_2  LATEbits.LATE3
#define L298_M2_1  LATEbits.LATE4
#define L298_M1_2  LATEbits.LATE1
#define L298_M1_1  LATEbits.LATE0
#define time_sampling 1
char ts=0;
float x=1.0,y=1.0,z=1.0;
void _ISR_T1Interrupt(void)
{
    IFS0bits.T1IF = 0;
    ts++;
}
void open_timer_service(void)
{
    ConfigTimer1(T1_INT_PRIOR_2 & T1_INT_ON);
    WriteTimer1(0);
    unsigned int TimeInterval = 1000;
    OpenTimer1(T1_ON & T1_GATE_OFF &
        T1_IDLE_STOP & T1_PS_1_1 &
        T1_SYNC_EXT_OFF & T1_SOURCE_INT,
        TimeInterval);
}
void init(void)
{
    TRISEbits.TRISE0 = 0;
    TRISEbits.TRISE1 = 0;
    TRISEbits.TRISE4 = 0;
    TRISEbits.TRISE3 = 0;
    open_timer_service();
}
float fx(float x,float y)
{
    return(35.0*(y-x));
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
float fy(float x,float y,float z)
{
    return(((28.0-35.0)*x)-(x*z)+(28.0*y));
}
```

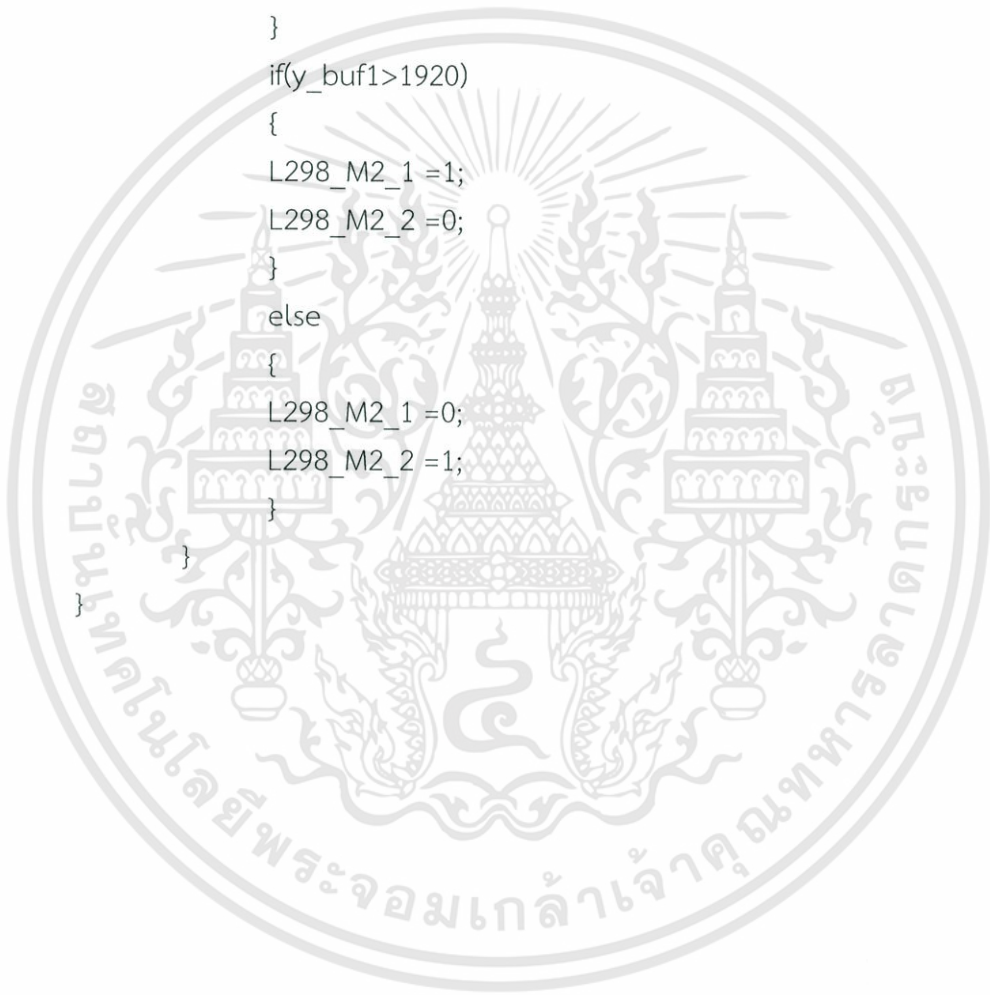
```
float fz(float x,float y,float z)
{
    return((x*y)-(2.66666667*z));
}
```

```
void Euler()
{ float h=0.01;
  x=x+(fx(x,y))*h;
  y=y+(fy(x,y,z))*h;
  z=z+(fz(x,y,z))*h;
}
int main(void)
{
    init();
    unsigned int x_buf1=0,y_buf1=0,z_buf1=0;
    int x_buf2=0,y_buf2=0,z_buf2=0;
    float x_buf3=0,y_buf3=0,z_buf3=0;
    while(1)
    {
        if(ts>time_sampling)
        {
            ts=0;
            Euler();
            x_buf3=(x+30)*40;
            y_buf3=(y+35)*40;
            z_buf3=(z+0)*40;
            x_buf2=x_buf3;
            y_buf2=y_buf3;
            z_buf2=z_buf3;
            x_buf1=x_buf2;
            y_buf1=y_buf2;
            z_buf1=z_buf2;
        }
    }
}
```

```

if(x_buf1>1680)
{
L298_M1_1 =1;
L298_M1_2 =0;
}
else
{
L298_M1_1 =0;
L298_M1_2 =1;
}
if(y_buf1>1920)
{
L298_M2_1 =1;
L298_M2_2 =0;
}
else
{
L298_M2_1 =0;
L298_M2_2 =1;
}
}
}
}

```



โค้ดคำสั่งที่ใช้ในการเคลื่อนที่ของหุ่นยนต์

```
#include<p30f2010.h>
#include<uart.h>          #include<timer.h>
#include<ports.h>
#define LED    LATDbits.LATD0
#define L298_M2_2 LATEbits.LATE3
#define L298_M2_1 LATEbits.LATE2
#define L298_M1_2 LATEbits.LATE1
#define L298_M1_1 LATEbits.LATE0
#define time_sampling 10
char Txdata[] = "\r\nChoose attractor by enter the number 0-3\r\n 0.Stop\r\n 1.Chen\r\n 2.Lorenz\r\n 3.Rossler\r\n\r";
char dat=0,data_in=0;
char ts=0;
float x=1.0,y=1.0,z=1.0, h=0.0;
float x_buf=0.0,y_buf=0.0,z_buf=0.0;
void _ISR_U1TXInterrupt(void)
{
    IFS0bits.U1TXIF = 0;
}
void _ISR_U1RXInterrupt(void)
{
    IFS0bits.U1RXIF = 0;
    data_in=1;
}
void _ISR_T1Interrupt(void)
{
    IFS0bits.T1IF = 0;
    ts++;
}
void uart1_init()
{
    unsigned int baudvalue;
    unsigned int U1MODEvalue;
    unsigned int U1STAValue;
    CloseUART1();
    ConfigIntUART1(UART_RX_INT_EN &
UART1
```

```

UART_RX_INT_PR6&
UART_TX_INT_EN &
UART_TX_INT_PR4);

baudvalue = 25;

U1MODEvalue =          UART_EN &
                        UART_IDLE_CON &
                        UART_RX_TX&
                        UART_DIS_WAKE &
                        UART_DIS_LOOPBACK &
                        UART_DIS_ABAUD &
                        UART_NO_PAR_8BIT &
                        UART_1STOPBIT;

U1STAvaleue =          UART_INT_TX_BUF_EMPTY &
                        UART_TX_PIN_NORMAL &
                        UART_TX_ENABLE &
                        UART_INT_RX_3_4_FUL&
                        UART_ADR_DETECT_DIS &
                        UART_RX_OVERRUN_CLEAR;

OpenUART1(U1MODEvalue, U1STAvaleue, baudvalue);
}
void open_timer_service(void)
{
    ConfigIntTimer1(T1_INT_PRIOR_2&T1_INT_ON);
    WriteTimer1(0);
    unsigned int TimeInterval = 1000;
    OpenTimer1(T1_ON & T1_GATE_OFF &
               T1_IDLE_STOP & T1_PS_1_1&
               T1_SYNC_EXT_OFF & T1_SOURCE_INT,
               TimeInterval);
}
void init(void)
{
    TRISEbits.TRISE0 = 0;
    TRISEbits.TRISE1 = 0;
    TRISEbits.TRISE2 = 0;

```

```

TRISEbits.TRISE3 = 0;
open_timer_service();
uart1_init();
}

float chen_fx(float x,float y)
{
    return(35.0*(y-x));
}

float chen_fy(float x,float y,float z)
{
    return(((28.0-35.0)*x)-(x*z)+(28.0*y));
}

float chen_fz(float x,float y,float z)
{
    return((x*y)-(2.6666667*z));
}

float lorenz_fx(float x,float y)
{
    return(10.0*(y-x));
}

float lorenz_fy(float x,float y,float z)
{
    return((28.0*x)-y-(x*z));
}

float lorenz_fz(float x,float y,float z)
{
    return((x*y)-(2.6666667*z));
}

float rossler_fx(float x,float y,float z)
{
    return((-y)-z);
}

float rossler_fy(float x,float y,float z)
{
    return(x+(0.1*y));
}

```

```

float rossler_fz(float x,float y,float z)
{
    return(0.1+((x-14.0)*z));
}
void motor_stop()
{

```

```

    L298_M1_1 =0;
    L298_M1_2 =0;

    L298_M2_1 =0;
    L298_M2_2 =0;
}

```

```

void motor_drive()
{
    if(x_buf>0.0)
    {
        L298_M1_1 =1;
        L298_M1_2 =0;
    }
    else
    {
        L298_M1_1 =0;
        L298_M1_2 =1;
    }

    if(y_buf>0.0)
    {
        L298_M2_1 =1;
        L298_M2_2 =0;
    }
    else
    {
        L298_M2_1 =0;
        L298_M2_2 =1;
    }
}

```

```

void Chen_Function()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    h=0.01;
    ts=0;
    x=x+(chen_fx(x,y))*h;
    y=y+(chen_fy(x,y,z))*h;
    z=z+(chen_fz(x,y,z))*h;

    x_buf=x;
    y_buf=y;
    z_buf=z;
    motor_drive();
}
void Lorenz_Function()
{
    h=0.01;
    ts=0;
    x=x+(lorenz_fx(x,y))*h;
    y=y+(lorenz_fy(x,y,z))*h;
    z=z+(lorenz_fz(x,y,z))*h;

    x_buf=x;
    y_buf=y;
    z_buf=z;
    motor_drive();
}
void Rossler_Function()
{
    h=0.006;
    ts=0;
    x=x+(rossler_fx(x,y,z))*h;
    y=y+(rossler_fy(x,y,z))*h;
    z=z+(rossler_fz(x,y,z))*h;

    x_buf=x;
    y_buf=y;
    z_buf=z;
    motor_drive();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int main(void)
{
    init();
    putsUART1 ((unsigned int *)Txdata);
    while(1)
    {
        if(data_in==1)
        {
            dat=ReadUART1();
            while(BusyUART1());
            data_in=0;
        }

        if(ts>time_sampling)
        {
            switch (dat)
            {
                case '0':{motor_stop(); break;}
                case '1':{Chen_Function(); break;}
                case '2':{Lorenz_Function(); break;}
                case '3':{Rossler_Function(); break;}
            }
        }
    }
    return 0;
}

```

โค้ดคำสั่งที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ผ่าน PHP

```
<?php
$val = $_GET['cmd'];
//$val = $_POST['Stop'];
if($val == "Stop"){
exec("mode com3: BAUD=9600 PARITY=n DATA=8 STOP=1 to=off dtr=off rts=off");
$fp =fopen("com3", "w");
//$fp = fopen('/dev/ttyUSB0','r+'); //use this for Linux
fwrite($fp, $val);
fclose($fp);
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Chaotic Robot</title>
</head>
<body>
<input type="button" name="Stop" value="Stop"
onClick="parent.location='chaoticrobot.php?cmd=00000000'"/><br />
<input type="button" name="Chen" value="Chen"
onClick="parent.location='chaoticrobot.php?cmd=11111111'"/><br />
<input type="button" name="Lorenz" value="Lorenz"
onClick="parent.location='chaoticrobot.php?cmd=22222222'"/><br />
<input type="button" name="Rossler" value="Rossler"
onClick="parent.location='chaoticrobot.php?cmd=33333333'"/><br />
</body>
</html>
```

โค้ดคำสั่งที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ผ่าน PHPตามทฤษฎีของ Chen

```
<?php
exec("mode com3: BAUD=9600 PARITY=n DATA=8 STOP=1 to=off dtr=off rts=off");
$fp =fopen("com3", "w");
//$fp = fopen('/dev/ttyUSB0','r+'); //use this for Linux

fwrite($fp, "11111111"); //write string to serial
fclose($fp);
?>
```

โค้ดคำสั่งที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ผ่าน PHPตามทฤษฎีของ Lorenz

```
<?php
exec("mode com3: BAUD=9600 PARITY=n DATA=8 STOP=1 to=off dtr=off rts=off");
$fp =fopen("com3", "w");
//$fp = fopen('/dev/ttyUSB0','r+'); //use this for Linux

fwrite($fp, "22222222"); //write string to serial
fclose($fp);
?>
```

โค้ดคำสั่งที่ใช้ในการเคลื่อนที่ของหุ่นยนต์ผ่าน PHPตามทฤษฎีของ Rossler

```
<?php
exec("mode com3: BAUD=9600 PARITY=n DATA=8 STOP=1 to=off dtr=off rts=off");
$fp =fopen("com3", "w");
//$fp = fopen('/dev/ttyUSB0','r+'); //use this for Linux

fwrite($fp, "33333333"); //write string to serial
fclose($fp);
?>
```

โค้ดคำสั่งที่ใช้ในการหยุดการเคลื่อนที่ของหุ่นยนต์ผ่าน PHP

```
<?php
exec("mode com3: BAUD=9600 PARITY=n DATA=8 STOP=1 to=off dtr=off rts=off");
$fp =fopen("com3", "w");
//$fp = fopen('/dev/ttyUSB0','r+'); //use this for Linux

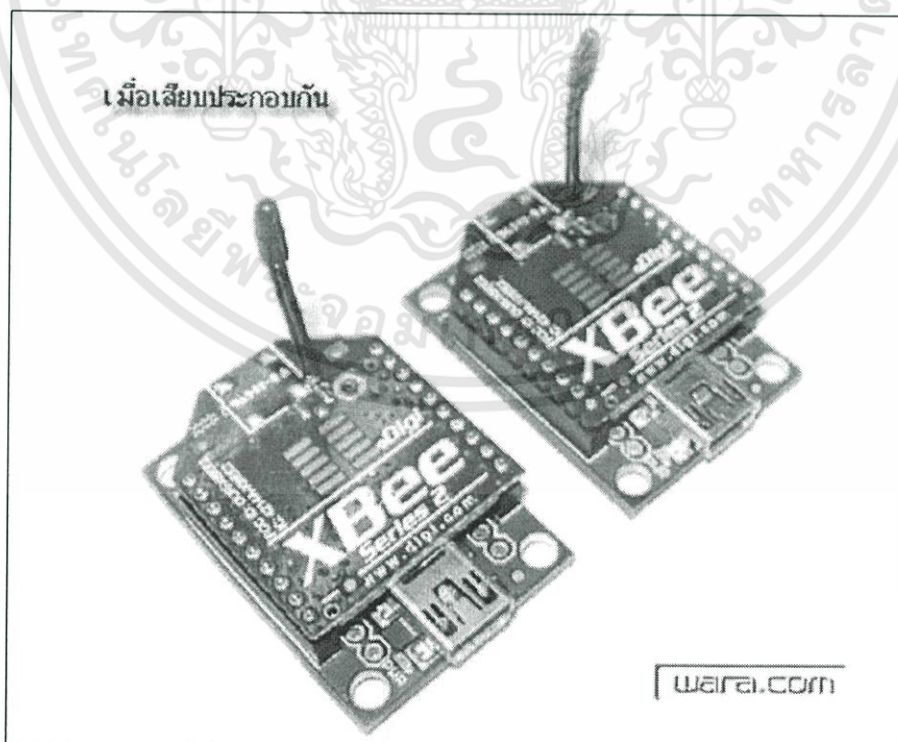
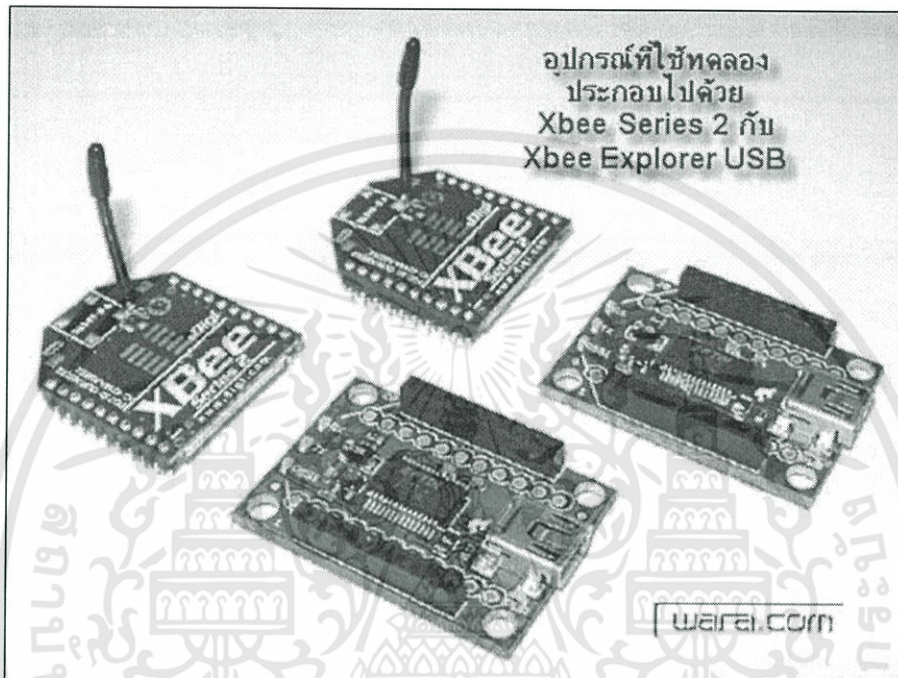
fwrite($fp, "00000000"); //write string to serial
fclose($fp);?>
```



การติดต่อสื่อสารของ Xbee เวลानำไปใช้งาน จะต้องมี

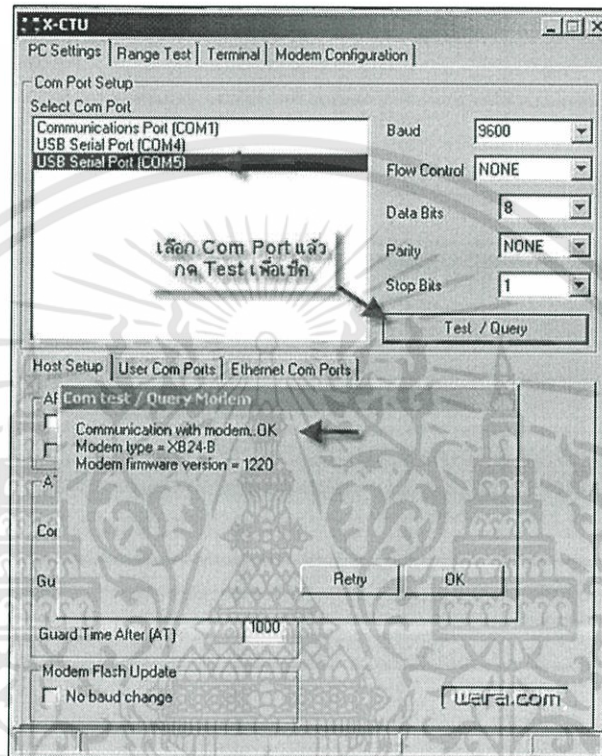
- โมดูล 1 ตัวที่เป็น Coordinator ซึ่งทำหน้าที่เชื่อมโยงเครือข่าย
- และโมดูลตัวอื่นๆ จะต้องเป็น Router / End Device

สรุป คือ ถ้ามีโมดูลสองตัว ต้องมีตัวหนึ่งตั้งค่าให้มัน เป็น Coordinator ตัว อื่นๆให้ เป็น Router/end device ครับ กฎเกณฑ์การใช้งานเจ้า Xbee ก็มีอยู่เท่านี้
เรามาทดลองเบื้องต้นกันครับสำหรับอุปกรณ์ที่ใช้ทดลองนี้จะมี

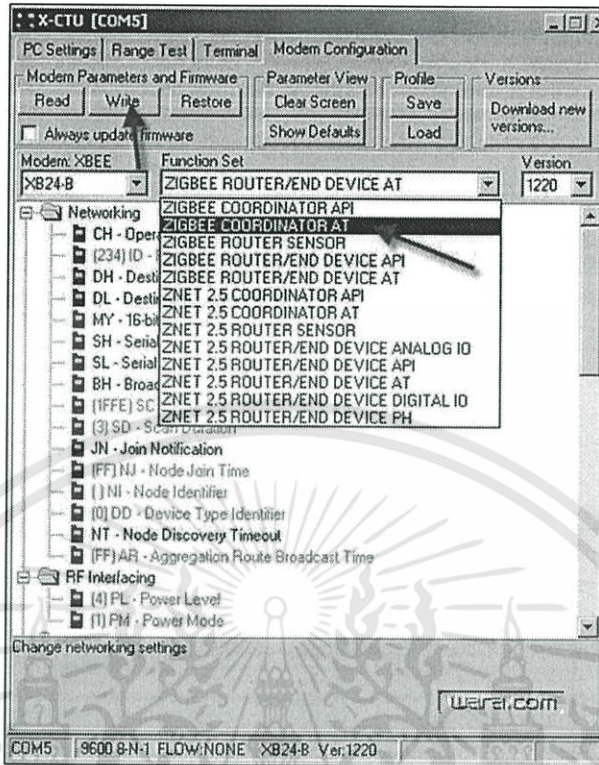


หลังจากที่เราเทียบประกอบกันเรียบร้อยแล้วต่อมาเราจะต้องโหลดโปรแกรมที่เอาไว้ช่วย Set ค่าของ Xbee โปรแกรมนี้คือ X-CTU

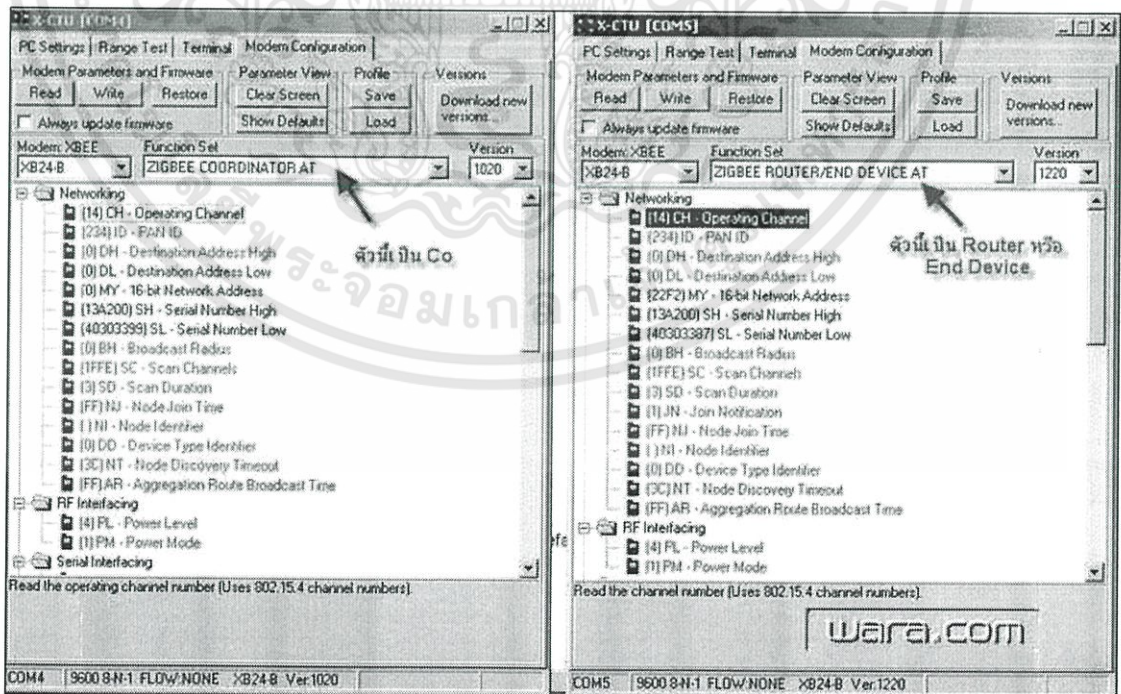
ต่อมาให้ทำการเสียบ Xbee ของเราเข้า ที่พอร์ต USB จากนั้นคอมพิวเตอร์จะถามหา Driver ของ FT232 RL ซึ่งทำหน้าที่เป็น USB to Serial เมื่อเครื่องเราพอร์ต Serial ที่เพิ่มเข้าไปเรียบร้อยแล้ว ต่อมา ให้เปิดโปรแกรม X-CTU ขึ้นมา



จากนั้นเลือก Comport แล้วกดที่ Test เพื่อตรวจเช็ค ถ้าขึ้นมาแบบดังภาพแสดงว่า OK จากนั้นคลิกที่ TAP Modem Configuration แล้วคลิกที่ Read หลังจากนั้น ทางโปรแกรมจะทำ อัปเดตข้อมูล Firmware ของ Xbee ซึ่งจะต้องต่อเน็ตด้วยเมื่ออัปเดตเสร็จแล้วจะขึ้นมาแบบนี้ครับ ซึ่งจะบอกว่า ขณะนี้ Xbee ของเรานั้นอยู่ใน โหมดอะไรอย่างตัวนี้อยู่ในโหมดของ Router and Device AT จากนั้นให้เราทำการเซตให้มันเป็น Coordinator ซึ่งในระบบจะต้องมี 1 ตัวที่เป็นครีป ให้เลือกดังนี้



ให้คลิกที่ช่องในแถบ Function Set แล้วเลือก Xbee Coordinator AT แล้วกด Write ต่อมาให้เราเสียบ Xbee อีกตัว แล้วทำการเซตให้เป็น Router / End Device AT ซึ่งส่วนใหญ่ Default จะตั้งมาแล้ว (ตรงนี้ลองทำเองดูครับไม่ยาก ทำเหมือนเดิม) เมื่อ set เรียบร้อยแล้วจะเป็นแบบนี้

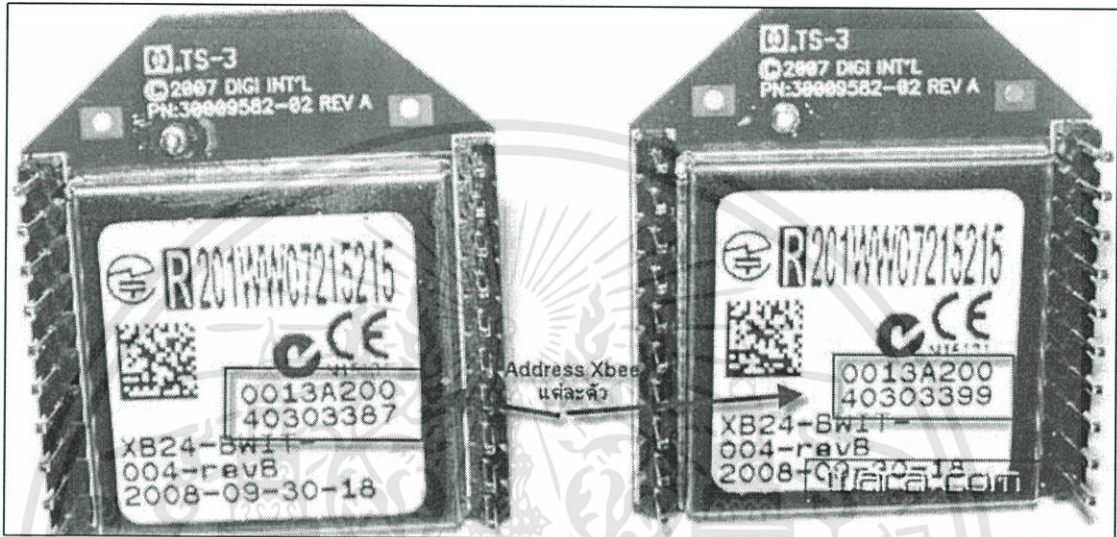


เมื่อทำการ Set เรียบร้อยแล้วต่อมาระยะต่อมาระยะต้องมา set ให้มัน รู้จักจุดหมายปลายทางที่มันจะส่งข้อมูลหรือ เจ้า Destination Address ซึ่งจะมี

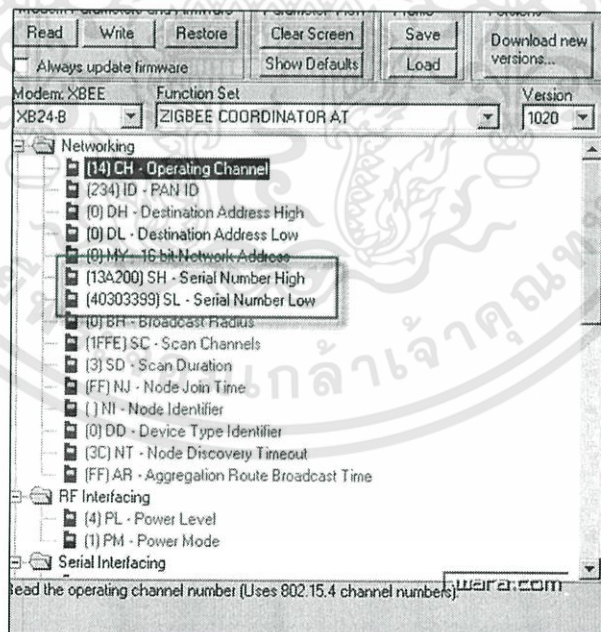
DH = Address บิตสูง

DL = address บิตต่ำ

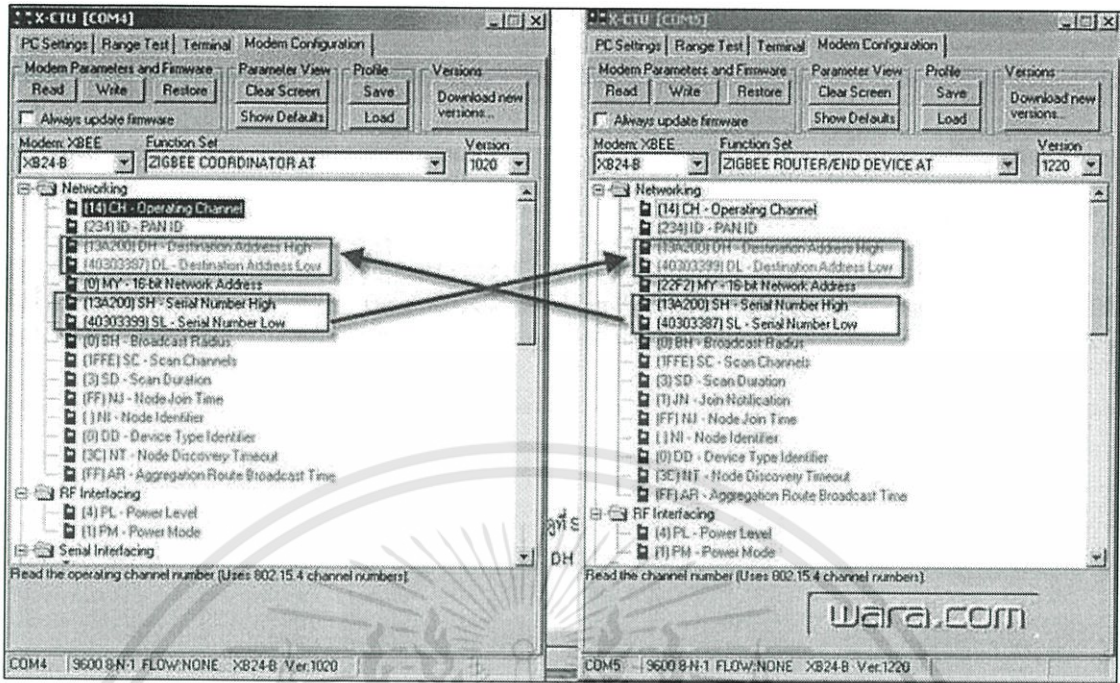
ซึ่ง Xbee แต่ละตัวจะมี Address Serial ที่โรงงานกำหนดมาให้แล้วแต่แต่ละตัวสามารถดูได้จาก 1. ที่ตัวโมดูล Xbee



2. ที่โปรแกรม X-ctu read ขึ้นมา



เราจะต้องทำการ Set ให้ Xbee รู้ว่าจะส่งข้อมูลไปให้ใคร ไปที่ตัวไหน Serial Address ไหน โดยดูที่ SH และ SL ในที่นี้เราต้องการให้อีกตัวส่งข้อมูลไปให้อีกตัว เลยให้กรอก Address ของตัวที่เราจะส่งที่ DH และ DL ได้เลย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้