

การวัดความสูงของเสาเข็ม
MEASURE THE HEIGHT OF THE PILE



โดย

นายกฤษ

วงศ์วัฒนฤกษ์

นายเจตพล

จงเจริญสมบัติ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

การวัดความสูงของเสาเข็ม
MEASURE THE HEIGHT OF THE PILE



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาโทรคมนาคม
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงแก้ไข หรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวัดความสูงของเสาเข็ม
MEASURE THE HEIGHT OF THE PILE

โดย

นายกฤษ วงศ์วัฒนฤกษ์ 53010034

นายเจตพล จงเจริญสมบัติ 53010251


อาจารย์ที่ปรึกษา
ผศ.สมภพ แก้วมีชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาโทรคมนาคม
คณะวิศวกรรมศาสตร์


สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

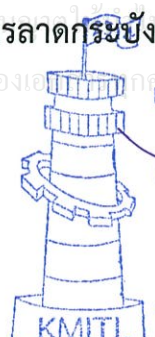
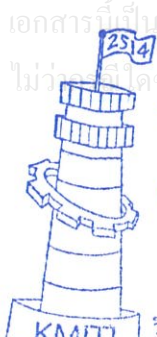
ปีการศึกษา 2556

ผ่านการตรวจรูปเล่มแล้ว


อาจารย์ที่ปรึกษา
7/3/57

ผ่านการตรวจชิ้นงานแล้ว


กรรมการผู้ตรวจชิ้นงาน
24/3/57



ปริญญาโทปีการศึกษา 2556

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การวัดความสูงของเสาเข็ม

MEASURE THE HEIGHT OF THE PILE

ผู้จัดทำ

1. นายกฤษ วงศ์วัฒนฤกษ์

53010034

2. นายเจตพล จงเจริญสมบัติ

53010251



อาจารย์ที่ปรึกษา

(ผศ.สมภาพ แก้วมีชัย)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำโครงการชิ้นนี้สำเร็จลุล่วงไปได้ด้วยดี ซึ่งทางผู้จัดทำได้รับความช่วยเหลือจากท่าน ผศ.สมภพ แก้วมีชัย ซึ่งท่านเป็นอาจารย์ที่ปรึกษาในโครงการนี้ และยังให้คำปรึกษาแนะนำต่างๆ ที่มีประโยชน์อย่างยิ่งต่อการทำโครงการนี้ให้เป็นผลสำเร็จลุล่วงไปได้ด้วยดีตลอดมา

ผู้จัดทำขอขอบคุณทุกๆ ท่านที่ได้ให้ความช่วยเหลือแนะนำสิ่งต่างๆ รวมถึงเพื่อนๆ ที่ช่วยแก้ไข ปัญหาต่างๆ ทำให้โครงการนี้เสร็จสมบูรณ์ ทางผู้จัดทำหวังว่าโครงการนี้จะเป็นประโยชน์แก่ผู้สนใจไม่มากก็น้อย หากในการจัดทำโครงการนี้มีข้อผิดพลาดประการใดผู้จัดทำก็ขออภัยไว้ ณ ที่นี้ด้วย

นายกฤษ วงศ์วัฒน์ฤกษ์

นายเจตพล จงเจริญสมบัติ

ผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวัดความสูงของเสาเข็ม MEASURE THE HEIGHT OF THE PILE

โดย นายกฤษ วงศ์วัฒนฤกษ์ 53010034
นายเจตพล จงเจริญสมบัติ 53010251

อาจารย์ที่ปรึกษา ผศ.สมภพ แก้วมีชัย

บทคัดย่อ

โครงการนี้เป็นการศึกษาเกี่ยวกับระบบตรวจสอบความเรียบร้อยพร้อมทั้งโปรแกรมวิเคราะห์ภาพเพื่อระบุความสูงของเสาเข็ม โดยระบบรับส่งกล้องระยะไกลจะใช้การส่งสัญญาณความถี่ย่าน VHF เป็นความถี่ช่วง 30-300 MHz และในส่วนของการรับสัญญาณจะใช้กล่องรับสัญญาณ Radio AV Receiver ในส่วนของการควบคุมกล้องระยะไกลจะใช้อุปกรณ์รับส่งสัญญาณ XBee ในการรับส่งข้อมูลจากคอมพิวเตอร์เข้าสู่คอนโทรลเลอร์เพื่อควบคุมสเต็ปมอเตอร์ โดยอุปกรณ์จะทำการรับภาพเข้าสู่คอมพิวเตอร์ ซึ่งสามารถเรียกดูภาพจากกล้องแบบ Real Time ได้ทันทีหรือสามารถทำการบันทึกภาพในช่วงเวลาหนึ่งเพื่อวิเคราะห์หาความสูงของเสาเข็มที่กำลังตอกได้โดยโปรแกรมวิเคราะห์ภาพ

ABSTRACT

This project is research on the monitoring and analysis to specify the height of the pile. The TV transmitter is used to send signals in VHF frequency range 30-300 MHz and the receiver is used Radio AV Receiver. The remote camera control is used XBee to send and receive signals from computer to microcontroller to control stepper motors. The device will send a real time videos to computer. You can browse images of the Real Time Camera DVR immediately or can be analyzed over time to determine the height of the piles when hammering by the image processing program.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

ชื่อเรื่อง	หน้า
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
1.4 ขั้นตอนการดำเนินงาน	2
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	3
2.1 งานเสาเข็ม	3
2.1.1 การควบคุมและตรวจสอบงานเสาเข็มตอก	3
2.1.2 การตรวจสอบ Blows Count	4
2.1.3 รายละเอียดของใบตรวจสอบ Blows Count	5
2.2 TV TRANSMITTER	6
2.2.1 ระบบการแพร่ภาพโทรทัศน์	6
2.2.2 สัญญาณภาพรวม (Composite Video Signal)	6
2.2.3 การผสมสัญญาณ (Modulation System)	7
2.2.4 ความถี่ที่ใช้ในการส่งสัญญาณโทรทัศน์	9
2.3 EasyCAP USB 2.0 Audio/Video Capture	12
2.4 กล้องไร้สาย (Wireless Camera)	13
2.4.1 ประเภทของกล้องไร้สาย	14
2.5 หน่วยความละเอียดของภาพ (Pixel)	15
2.6 ET USB/RS232 MINI	16
2.6.1 คุณสมบัติของ ET USB/RS232 MINI	16
2.6.2 การใช้งาน ET-USB/RS232 MINI	16
2.7 XBee	17
2.7.1 คุณสมบัติของ XBee	18
2.7.2 สายอากาศของ XBee	18
2.7.3 Data Throughput ของ XBee	19
2.8 Arduino mega 2560	20
2.9 STEPPER MOTOR	21
2.9.1 ชนิดของสเต็ปเปอร์มอเตอร์	21
2.9.2 ชนิดของสเต็ปเปอร์มอเตอร์แบ่งตามลักษณะสายที่ใช้ต่อกับวงจรขับเคลื่อน	23

2.7.3 วิธีการขับ (Driving) หรือวิธีการกระตุ้นเฟส (Phase Excitation) ของสเต็ปเปอร์มอเตอร์	25
-----------------------------------------------------------------------------------------------	----

บทที่ 3 การออกแบบและการจัดทำโครงงาน	27
--------------------------------------------	-----------

3.1 การออกแบบ	27
3.1.1 ส่วนของโปรแกรมวัดความสูง	27
3.1.2 การหาความสูงจากภาพในหน่วยจำนวนจุดพิกเซล	28
3.1.3 การหาความสูงของเสาเข็มโดยอ้างอิงจากระยะห่างระหว่างกล้อง และเสาเข็ม	29
3.1.4 การหาความสูงของเสาเข็มโดยอ้างอิงจากความสูงของบันจันตอก เสาเข็ม	30
3.1.5 ส่วนของระบบควบคุมกล้องไร้สาย	32
3.1.6 ส่วนของระบบรับ-ส่งภาพไร้สาย	35
3.2 เครื่องมือที่ใช้ในการทดลอง	35
3.2.1 ส่วนของระบบควบคุมกล้องระยะไกล	35
3.2.2 ส่วนของระบบรับ-ส่งภาพไร้สาย	38
3.3 การจัดเก็บผลการทดลอง	40

บทที่ 4 ผลการทดลอง	41
---------------------------	-----------

4.1 โปรแกรมวัดความสูง	41
4.1.1 โปรแกรมวัดความสูงหน้าต่างที่หนึ่ง	41
4.1.2 โปรแกรมวัดความสูงหน้าต่างที่สอง	42
4.1.3 โปรแกรมวัดความสูงหน้าต่างที่สาม	43
4.2 ค่าอัตราส่วนความสูงระหว่างเมตรต่อจำนวนจุดพิกเซลที่ระยะต่างๆ	44
4.3 การวัดความสูงของเสาเข็มโดยอ้างอิงจากระยะห่างระหว่างกล้องและเสาเข็ม	49
4.3.1 วัดความสูงของเสาเข็มที่ระยะ 30 เมตร	50
4.3.2 วัดความสูงของเสาเข็มที่ระยะ 35 เมตร	51
4.3.3 วัดความสูงของเสาเข็มที่ระยะ 45 เมตร	52
4.3.4 วัดความสูงของเสาเข็มที่ระยะ 55 เมตร	53
4.3.5 สรุปผลการวัดความสูงของเสาเข็มโดยอ้างอิงจากระยะห่าง ระหว่างกล้องและเสาเข็ม	54
4.4 การวัดความสูงของเสาเข็มโดยอ้างอิงจากความสูงของบันจัน	55

เอกสารนี้เป็นเอกสารที่	4.4.1 วัดความสูงโดยอ้างอิงจากความสูงของบันจันครั้งที่หนึ่ง	หน้าไปใช้ประโยชน์	55	ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีก	4.4.2 วัดความสูงโดยอ้างอิงจากความสูงของบันจันครั้งที่สอง		56	

4.4.3 วัดความสูงโดยอ้างอิงจากความสูงของบันจันครั้งที่สาม	57
4.4.4 วัดความสูงโดยอ้างอิงจากความสูงของบันจันครั้งที่สี่	58
4.4.5 สรุปผลการวัดความสูงของเสาเข็มโดยอ้างอิงจากความสูงของบันจัน	59
4.5 การรับ-ส่งข้อมูลระหว่าง XBee ผ่านโปรแกรม X-CTU	60
4.6 ทดสอบไมโครคอนโทรลเลอร์ในการติดต่อกับคอมพิวเตอร์ผ่าน Serial Port	61
4.7 ทดสอบไมโครคอนโทรลเลอร์ในการติดต่อกับคอมพิวเตอร์ผ่านอุปกรณ์ XBee	62
4.8 ULN2003 และ Stepper Motor	63
4.9 ทดสอบการรับ-ส่งภาพไร้สายที่ระยะต่างๆ	63
4.10 อุปกรณ์ควบคุมกล้องโดยรวม	65
บทที่ 5 สรุปผลและข้อเสนอแนะ	67
5.1 สรุปผลการทดลอง	67
5.2 ข้อเสนอแนะ	67
บรรณานุกรม	
ภาคผนวก ก หน้าต่างโปรแกรมรับภาพจากกล้อง	
ภาคผนวก ข หน้าต่างโปรแกรมวัดค่าความสูงโดยอ้างอิงจากระยะห่างระหว่างกล้องและเสาเข็ม	
ภาคผนวก ค หน้าต่างโปรแกรมวัดค่าความสูงโดยอ้างอิงจากความสูงของบันจันที่ใช้ตอก	
ภาคผนวก ง โปรแกรมควบคุมสเต็ปมอเตอร์	
ภาคผนวก ฉ Data Sheet	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

ชื่อรูป	หน้า
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	3
รูปที่ 2.1 งานเสาเข็ม	3
รูปที่ 2.2 การนับค่า Blow Count	4
รูปที่ 2.3 การมอดูเลชัน	7
รูปที่ 2.4 วิธีการ Vestigial Sideband	8
รูปที่ 2.5 EasyCAP USB 2.0 Audio/Video Capture	12
รูปที่ 2.6 กล้องไร้สาย (Wireless Camera)	13
รูปที่ 2.7 ภาพการขยายรูปจนเห็นจุดพิกเซล	15
รูปที่ 2.8 แสดงการระบุตำแหน่งของจุดพิกเซลในภาพ	15
รูปที่ 2.9 ET USB/RS232 MINI	16
รูปที่ 2.10 การต่อใช้งาน ET-USB/RS232 MINI	17
รูปที่ 2.11 XBee รุ่นสายอากาศแบบ Wire (Whip)	18
รูปที่ 2.12 Data Throughput ของ Xbee	19
รูปที่ 2.13 Arduino MEGA 2560	20
รูปที่ 2.14 โครงสร้างสเต็ปเปอร์มอเตอร์ชนิดวาริเอเบิลรีล็กแตนต์และสัญลักษณ์	22
รูปที่ 2.15 สเต็ปเปอร์มอเตอร์ชนิดเพอร์มาเนนต์แมกเน็ต	23
รูปที่ 2.16 สัญลักษณ์, โครงสร้างและวงจรขับที่ใช้กับมอเตอร์แบบโพลลา 2 เฟส	24
รูปที่ 2.17 สัญลักษณ์, โครงสร้างและวงจรขับที่ใช้กับมอเตอร์แบบยูนิโพลลา 2 เฟส	24
บทที่ 3 การออกแบบและการจัดทำโครงงาน	27
รูปที่ 3.1 บล็อกไดอะแกรมของโปรแกรมวัดความสูง	27
รูปที่ 3.2 แสดงการหาตำแหน่งจุดพิกเซลจากภาพ	28
รูปที่ 3.3 (ก) ประตูดาวสูง 2 เมตร	30
รูปที่ 3.3 (ข) แสดงการวัดความสูงของประตูดาว (2 เมตร)	30
รูปที่ 3.4 ภาพปั้นจั่นความสูง 18 เมตร ขณะทำการตอกเสาเข็ม	31
รูปที่ 3.5 ระบบควบคุมกล้องไร้สายฝั่งส่ง	32
รูปที่ 3.6 ระบบควบคุมกล้องไร้สายฝั่งรับ	32
รูปที่ 3.7 วงจรรวมของระบบควบคุมกล้องระยะไกล	33
รูปที่ 3.8 โฟลวชาร์ตการทำงานของไมโครคอนโทรลเลอร์	34
รูปที่ 3.9 ระบบรับ-ส่งภาพไร้สาย	35
รูปที่ 3.10 อุปกรณ์ XBee ฝั่งส่ง (Tx) และ ET USB/RS232 MINI	35

รูปที่ 3.11 อุปกรณ์ XBee ฝั่งรับ (Rx)	36
รูปที่ 3.12 รูปวงจร Stepping Motor	36
รูปที่ 3.13 วงจร Stepping Motor	37
รูปที่ 3.14 รูปการเชื่อมต่อ XBee เข้าสู่ไมโครคอนโทรลเลอร์	37
รูปที่ 3.15 วงจร XBee เชื่อมต่อกับไมโครคอนโทรลเลอร์	38
รูปที่ 3.16 กล้องไร้สาย (Wireless Camera)	38
รูปที่ 3.17 รูปแสดงกล่องรับสัญญาณ Radio AV Receiver	39
รูปที่ 3.18 EasyCAP USB Video Capture Adapter	40
บทที่ 4 ผลการทดลอง	41
รูปที่ 4.1 โปรแกรมวัดความสูงหน้าต่างที่หนึ่ง	41
รูปที่ 4.2 โปรแกรมวัดความสูงหน้าต่างที่สอง	42
รูปที่ 4.3 โปรแกรมวัดความสูงหน้าต่างที่สาม	43
รูปที่ 4.4 กราฟแสดงแนวโน้มของอัตราส่วนความสูงในหน่วยเมตรต่อ จำนวนจุดพิกเซลที่ระยะต่าง	48
รูปที่ 4.5 แสดงการวัดเสาต้นที่ 1 ส่วนที่โผล่จากพื้นดิน (50 เซนติเมตร)	49
รูปที่ 4.6 การเชื่อมเสาต้นที่สองและเสาต้นที่หนึ่ง	49
รูปที่ 4.7 ความสูงของเสาเข็มที่ระยะ 30 เมตร	50
รูปที่ 4.8 ความสูงของเสาเข็มที่ระยะ 35 เมตร	51
รูปที่ 4.9 ความสูงของเสาเข็มที่ระยะ 45 เมตร	52
รูปที่ 4.10 ความสูงของเสาเข็มที่ระยะ 55 เมตร	53
รูปที่ 4.11 กราฟแสดงค่าความผิดพลาดและเปอร์เซ็นต์ความผิดพลาด	54
รูปที่ 4.12 วัดความสูงโดยอ้างอิงจากความสูงของบันจันครั้งที่หนึ่ง	55
รูปที่ 4.13 วัดความสูงโดยอ้างอิงจากความสูงของบันจันครั้งที่สอง	56
รูปที่ 4.14 วัดความสูงโดยอ้างอิงจากความสูงของบันจันครั้งที่สาม	57
รูปที่ 4.15 วัดความสูงโดยอ้างอิงจากความสูงของบันจันครั้งที่สี่	58
รูปที่ 4.16 กราฟแสดงค่าความผิดพลาดและเปอร์เซ็นต์ความผิดพลาด	59
รูปที่ 4.17 การรับส่งข้อมูลระยะไกลโดยอุปกรณ์ XBee	60
รูปที่ 4.18 โปรแกรมคอนโทรลเลอร์ Arduino ในการควบคุม Step Motor	61
รูปที่ 4.19 โปรแกรมคอนโทรลเลอร์ X-CTU ในการรับส่งสัญญาณกับ ไมโครคอนโทรลเลอร์	62
รูปที่ 4.20 ตำแหน่งรับ-ส่งสัญญาณระยะ 30 เมตร	64
รูปที่ 4.21 ตำแหน่งรับ-ส่งสัญญาณระยะ 40 เมตร	64
รูปที่ 4.22 ตำแหน่งรับ-ส่งสัญญาณระยะ 45 เมตร	65

รูปที่ 4.23 อุปกรณ์ควบคุมกล้องโดยรวม

65

รูปที่ 4.24 อุปกรณ์ควบคุมกล้องโดยรวม

66



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ชื่อรูป	หน้า
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	3
ตารางที่ 2.1 ความถี่วิทยุในประเทศไทย	9
ตารางที่ 2.2 ตารางแสดงความถี่การส่งที่วีระบบ FCC ย่าน VHF	10
ตารางที่ 2.3 ตารางแสดงความถี่การส่งที่วีระบบ CCIR ย่าน VHF	10
ตารางที่ 2.4 ตารางแสดงความถี่การส่งที่วีระบบ CCIR ย่าน UHF	11
ตารางที่ 2.5 ระยะทางสัญญาณ XBee ที่การใช้งานต่างๆ	19
ตารางที่ 2.6 แสดงการกระตุ้นเฟสแบบ One Phase	25
ตารางที่ 2.7 แสดงการกระตุ้นเฟสแบบ Two Phase	25
ตารางที่ 2.8 แสดงการกระตุ้นแบบ One – Two Phase Excitation	26
บทที่ 4 ผลการทดลอง	41
ตารางที่ 4.1 ค่าอัตราส่วนความสูงโดยกล่องไร้สายที่ใช้ในการทดลอง	44
ตารางที่ 4.1 (ต่อ) ค่าอัตราส่วนความสูงโดยกล่องไร้สายที่ใช้ในการทดลอง	45
ตารางที่ 4.2 ค่าอัตราส่วนความสูงโดยกล่องจากโทรศัพท์มือถือ	46
ตารางที่ 4.2 (ต่อ) ค่าอัตราส่วนความสูงโดยกล่องจากโทรศัพท์มือถือ	47
ตารางที่ 4.3 สรุปผลการวัดความสูงของเสาเข็มโดยอ้างอิงจากระยะห่างระหว่างกล่องและเสาเข็ม	54
ตารางที่ 4.4 สรุปผลการวัดความสูงของเสาเข็มโดยอ้างอิงจากความสูงของบันจัน	69
ตารางที่ 4.5 แสดงการจ่ายไฟของ ULN2003 เข้าสู่ Stepping Motor	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในช่วงเวลาปัจจุบันได้มีการก่อสร้างสิ่งต่างๆ มากขึ้นอย่างรวดเร็วเนื่องจากจำนวนประชากรที่เพิ่มขึ้นทำให้มีความต้องการที่อยู่อาศัยเพิ่มขึ้นตาม ในการก่อสร้าง วิศวกรผู้รับผิดชอบงานนั้นๆ จะต้องทำการตรวจสอบความถูกต้องและคอยเฝ้าระวังปัญหาต่างๆ ที่อาจเกิดขึ้น รวมไปถึงการทำทุจริตในค่าวัสดุที่ใช้ในการก่อสร้าง วิศวกรจึงต้องคอยเฝ้าดูการก่อสร้างอยู่ตลอดเวลาของการทำงาน ทำให้วิศวกรผู้รับผิดชอบต้องเสียเวลาในการเฝ้าระวังโดยเปล่าประโยชน์

ดังนั้น เพื่อเป็นการอำนวยความสะดวกสำหรับวิศวกรที่มีหน้าที่คอยเฝ้าดูขั้นตอนการดำเนินการก่อสร้าง จึงได้สร้างระบบกล้องถ่ายระยะไกลรวมไปถึงการวัดความสูงของเสาเข็มที่กำลังตอก เพื่อป้องกันความมั่งงายของผู้ก่อสร้างและการทุจริตค่าวัสดุที่อาจทำให้การก่อสร้างนั้นๆ ไม่เป็นไปตามมาตรฐานที่ได้วางแผนไว้และก่อให้เกิดความเสียหายติดตามมา

1.2 วัตถุประสงค์

1. เพื่อศึกษาและออกแบบโปรแกรมตรวจจับภาพเพื่อวัดความสูงของเสาเข็มขณะทำการตอก
2. เพื่อศึกษาและออกแบบอุปกรณ์รับส่งภาพไร้สาย
3. เพื่อศึกษาและออกแบบอุปกรณ์ควบคุมไร้สาย
4. เพื่อตรวจสอบ ระบบฐานราก ให้เป็นไปตามแบบแปลน
5. เพิ่มความสะดวกสบายในการตรวจสอบความคืบหน้าของการสร้างฐานราก

1.3 ขอบเขตของปริญญาานิพนธ์

1. โปรแกรมตรวจจับภาพเพื่อวัดความสูงของเสาเข็มขณะทำการตอก
2. อุปกรณ์ส่งและรับสัญญาณโทรทัศน์
3. อุปกรณ์การควบคุมกล้อง
4. นำภาพมาแสดงในจอมอนิเตอร์พร้อมทั้งระบุความสูงของเสาเข็มขณะทำการตอกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาและออกแบบโปรแกรมในการประมวลผลภาพเบื้องต้น
2. ออกแบบอุปกรณ์ที่ใช้ในการถ่ายทอดสัญญาณโทรทัศน์
3. ออกแบบอุปกรณ์ที่ใช้ในการควบคุมกล้องไร้สาย
4. การสร้างระบบควบคุมโดยใช้โปรแกรมไมโครคอนโทรลเลอร์
5. โปรแกรมการวัดความสูงของเสาเข็มขณะทำการตอก
6. บันทึกผลและปรับปรุงระบบโดยรวม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 งานเสาเข็ม

เสาเข็มเป็นส่วนประกอบของฐานรากในการถ่ายน้ำหนักของฐานรากลงสู่ดิน ซึ่งชนิดของเสาเข็มที่ใช้จะเปลี่ยนแปลงไปตามดินที่มีคุณสมบัติการรับน้ำหนักได้มากหรือน้อยเช่น ดินเหนียว ดินทราย ดินแม่น้ำ เป็นต้น เสาเข็มที่นิยมใช้มีอยู่ 2 ประเภทใหญ่ๆ คือ เสาเข็มตอก และเสาเข็มเจาะ การที่จะเลือกใช้ประเภทของเสาเข็มขึ้นอยู่กับ สิ่งแวดล้อมในพื้นที่ก่อสร้าง เช่น เส้นทางคมนาคม สิ่งก่อสร้างใกล้เคียง สภาพชุมชน และขนาดพื้นที่ของงานก่อสร้างเอง

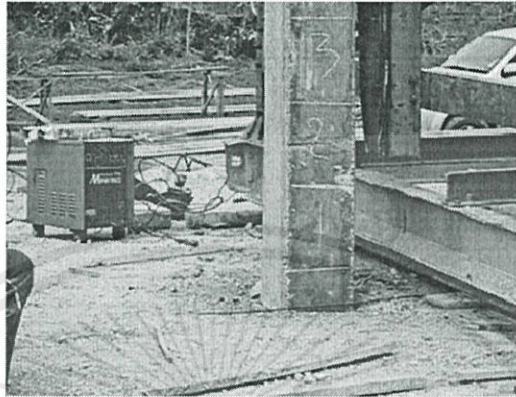


รูปที่ 2.1 งานเสาเข็ม

2.1.1 การควบคุมและตรวจสอบงานเสาเข็มตอก

เสาเข็มตอกในปัจจุบันนี้ทำด้วยคอนกรีตอัดแรงซึ่งมีให้เลือกอยู่ 4 ชนิดคือ เสาเข็มสี่เหลี่ยมตันใช้กับดินแข็ง เสาเข็มหน้าตัดตัวโอซึ่งใช้กับดินเหนียวหรือดินอ่อน เพราะมีพื้นที่ผิวรับแรงเสียดทานได้มาก เสาเข็มแรงเหวี่ยงซึ่งมีรูปหน้าตัดกลมและมีรูอยู่ตรงกลางสามารถรับน้ำหนักได้มากกว่าเสาเข็มสองชนิดข้างต้นเพราะมีความแข็งแรงสูงจึงสามารถตอกลงได้ลึกมากกว่าอีกทั้งต้องใช้เครื่องมือตอกเฉพาะทางอีกด้วย เสาเข็มหกเหลี่ยมเส้นผ่านศูนย์กลาง 15 ซม. เป็นเสาเข็มที่ผลิตขึ้นมาทดแทนเสาเข็มไม้ ซึ่งจะใช้ในฐานรากดินอ่อนที่ต้องตอกเป็นกลุ่ม การควบคุมงานและตรวจสอบงานเสาเข็มให้ถูกต้องนั้นต้องประกอบด้วยเครื่องมือที่สมรรถนะดี มีบุคลากรที่มีประสบการณ์ รวมถึงขั้นตอนในการเอกสาคือ ควบคุมงานและตรวจสอบที่ดี รับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 การตรวจสอบ Blows Count



รูปที่ 2.2 การนับค่า Blow Count

Blow Count เป็นการนับจำนวนครั้งในการตอก โดยจะแบ่งซัดเป็นช่วงๆ ช่วงละ 1 ฟุต โดยมีทั้งหมด 10 ช่วงคือ 10 ฟุต = 3 เมตร จากปลายเข็ม การดูหน้างานก็คือนับจำนวนครั้งว่าในช่วง 1 ฟุต ตอกไปกี่ครั้ง โดยในการตอกในแต่ละฟุตก็จะจดบันทึกค่าว่าตอกไปกี่ครั้งและเมื่อเห็นสมควรว่าควรหยุดการทำการตรวจสอบ Last Ten Blow คือการตอก 10 ครั้งสุดท้ายให้มีค่าที่ได้ไม่เกินระยะการทรุดตัวจากการคำนวณ ส่วนค่า Blow เป็นค่าที่ดูว่าควรหยุดตอกหรือยังโดยใช้ประสบการณ์ดูเอาเองและเพื่อป้องกันเสาเข็มเสียหายค่า Blow ไม่ควรเกิน 100 ครั้ง/ฟุต เสาเข็มทุกต้นจะต้องตอกต่อเนื่องกันโดยไม่มี การหยุดชะงักนานเกินสมควรตั้งแต่เริ่มตอกจนถึงตำแหน่งสุดท้ายของเสาเข็มนั้นๆ และอย่างน้อยต้อง ตอกให้ปลายเสาเข็มถึงความลึกที่กำหนดไว้ในแบบก่อสร้างโดยสามารถแบ่งเป็นหัวข้อได้ดังนี้

1. เมื่อตอกเสาเข็มจนถึงจุดที่ไม่สามารถตอกลงไปได้อีก เมื่อเสาเข็มตอกไม่ลงวิศวกร ผู้ออกแบบพิจารณาแล้วเห็นว่า Blow Count สูงกว่าค่าการคำนวณ การตอกต่อไปจะอันตรายและเกิดผลเสียหายต่อเสาเข็มได้ จะต้องหยุดการตอกเสาเข็มต้นนั้น ในกรณีเช่นนี้วิศวกรอาจสั่งให้เปลี่ยนความยาวของเสาเข็มเพื่อให้เหมาะสมกับงานได้

2. เมื่อตอกถึงระดับที่ต้องการแล้วจำนวน Blow Count ต่ำกว่าที่วิศวกรผู้ออกแบบ กำหนดไว้ผู้รับจ้างจะต้องตอกส่งเสาเข็มให้ลึกกว่าระดับที่กำหนดไว้ในแบบก่อสร้างจนกว่าจะได้ Blow Count ตามกำหนดและเสริมต่อความยาวเสาเข็ม

3. การตอกเสาเข็มจะต้องตอกให้ตรงศูนย์และได้ดิ่ง โดยระยะผิดศูนย์ต้องไม่เกิน 70 มม. สำหรับเสาเข็มกลุ่มและ 40 มม. สำหรับเสาเข็มเดี่ยว, คู่แเบะที่เรียงกันเป็นแถวเดียว ระยะดิ่งไม่เกิน 1% ของความยาวเสาเข็ม หากเสาเข็มต้นใดตอกออกศูนย์และผิดดิ่งเกินกว่าข้อกำหนดนี้โดยวัดขนาดกับ แกน Coordinate ทั้งสองแกนจะต้องให้วิศวกรผู้ออกแบบเป็นผู้วินิจฉัยเสาเข็มต้นนั้นทันที

4. ระยะของเสาเข็มเมื่อตอก 10 ครั้งสามชุดสุดท้าย ของเสาเข็มทุกต้น หากระยะจมของการ ตอก 10 ครั้งของสามชุดสุดท้ายไม่ลดลงตามลำดับ ผู้รับจ้างต้องตอกเสาเข็มลงไปอีกโดยตอก 10 ครั้ง

อีกสามชุดจนกว่าระยะจมของสามครั้งสุดท้ายต้องน้อยลงตามลำดับหรือจำนวนครั้งที่ตอกสำหรับ 100 มม. สุดท้าย จำนวนสามชุดจะต้องเพิ่มขึ้นตามลำดับ ถ้าหากจำนวนครั้งที่ตอกไม่เพิ่มขึ้นผู้รับจ้างจะต้องตอกไปจนกว่าจำนวนครั้งของสามชุดสุดท้ายเพิ่มขึ้นตามลำดับ ในกรณีที่ผู้รับจ้างเสนอให้เสาเข็มหรือดัดมตอกขนาดอื่นนอกเหนือกว่าที่กำหนด วิศวกรผู้ออกแบบจะเป็นผู้กำหนดเกี่ยวกับ Blow Count และระยะจมของเสาเข็มให้ ในภายหลังเป็นกรณีๆไป

2.1.3 รายละเอียดของใบตรวจสอบ Blows Count

ในใบเช็ค Blows Count หัวกระดาษต้องระบุรายละเอียดเกี่ยวกับบริษัท เช่น ชื่อ ที่อยู่ ของบริษัท ในส่วนต่อมาจะเป็นการระบุชื่ออาคารและสถานที่ที่จะทำ การก่อสร้าง น้ำหนักของดัดมยอตตอก และผู้ว่าจ้างโดยส่วนประกอบอื่นที่ประกอบในใบตรวจสอบ Blows Count มีดังนี้

- ระบุถึงขนาดเสาเข็มที่ใช้ตอก เช่น 35 x 8 m เป็นต้น
- ระบุถึงตำแหน่งที่จะตอกเสาเข็มซึ่งอาจจะบอกเป็นตัวเลขมีไว้เพื่อจะได้ระบุตำแหน่งที่จะตอกได้ถูกต้อง ที่ระบุไว้ในแบบของฐานราก
- ระบุถึงเบอร์เสาเข็มที่ออกมาจากโรงงานที่ผลิตมีเพื่อไว้บันทึกจำนวนเสาเข็มว่าครบหรือไม่
- ระบุถึงเวลาที่ตอกเสาเข็ม ณ ตอนที่เริ่มตอกและตอนที่ตอกเสร็จแล้ว
- จำนวน Blows ต่อฟุต หมายความว่า นับจำนวนครั้งที่ตอกให้ได้ตามที่แบบระบุไว้ เช่น ตัวอย่างความยาวมีตั้งแต่ 1 ถึง 12 ฟุต ในแต่ละฟุตต้องนับจำนวนครั้งให้ได้ 60 ครั้งก่อนที่จะใช้งาน
- หัวเสาเข็มที่โผล่ขึ้นมา
- ความเอียงของเสาเข็มว่าเอียงไปทางทิศใดและเอียงไปกี่เซนติเมตร
- ความลึกของเสาเข็มที่จะตอกในแบบว่าตอกลึกลงไปเท่าใด (P.Top) และส่วนที่เหลือของเสาเข็มที่พื้นดินเหลือเท่าใด (P.Tip)
- ความลึกของเสาเข็มในงานจริงว่าตอกลึกลงไปเท่าใด (P.Top) และส่วนที่เหลือของเสาเข็มที่พื้นดินเหลือเท่าใด (P.Tip)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 TV TRANSMITTER

2.2.1 ระบบการแพร่ภาพโทรทัศน์

ในกระบวนการแพร่ภาพโทรทัศน์ภาคส่งจะรับสัญญาณภาพเข้ามามอดูเลตแบบ AM และจะรับสัญญาณเสียงเข้ามามอดูเลตแบบ FM จากนั้นจึงนำสัญญาณทั้งสองมารวมกับสัญญาณพาห้ที่ความถี่ที่ต้องการแล้วส่งออกไปผ่าน Antenna ภาครับคือรับผ่านเสาอากาศรับที่วิศวกรรมดาเข้าทีวีแบบปกติ

2.2.2 สัญญาณภาพรวม (Composite Video Signal)

สัญญาณโทรทัศน์ประกอบด้วยสัญญาณใหญ่ ๆ 2 สัญญาณคือ สัญญาณภาพและสัญญาณเสียง ดังนั้นระบบการผสมความถี่หรือการมอดูเลชัน (Modulation System) จึงค่อนข้างยุ่งยากพอสมควร ประการแรกก็คือสัญญาณภาพเป็นสัญญาณที่มีตัวประกอบหลายต่อหลายตัว ส่วนสัญญาณเสียงเป็นสัญญาณธรรมดาที่ไม่มีความยุ่งยากมากนัก โดยสัญญาณภาพจะถูกมอดูเลชันในระบบเอเอ็ม. (Amplitude Modulation) ทั้งนี้ในระบบเอเอ็ม. การผสมสัญญาณมิได้ทำให้เกิดการแปรความถี่เป็นความถี่ใหม่ตามสัญญาณความถี่ วิทยุแต่อย่างใด สัญญาณภาพเป็นสัญญาณที่มีสัญญาณซิงโครไนซ์ , แบลกกิ่ง, อีควอลไลซิง ฯลฯ รวมอยู่ด้วย จึงทำให้แบนด์วิดธ์ในการมอดูเลชันสูงกว่าที่ควรจะเป็น ในเครื่องรับโทรทัศน์จึงค่อนข้างจะยุ่งยาก เมื่อเข้าสู่วิธีการเกี่ยวกับระบบการแปรสภาพสัญญาณให้เป็นภาพที่หน้าจอ แต่หากเรามอดูเลชันในระบบ เอเอ็ม. เราจะพบว่าความยุ่งยากจะลดน้อยลง

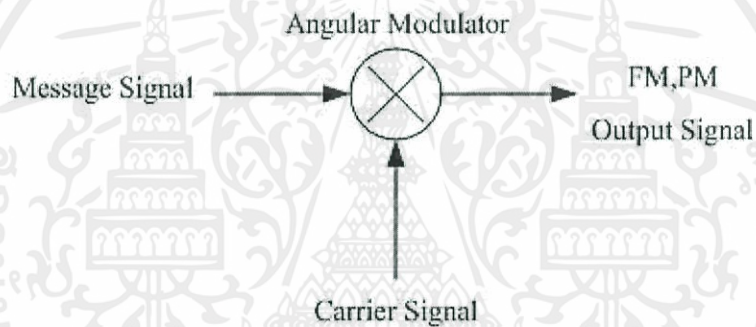
กล้องโทรทัศน์จะรับเอาสัญญาณภาพในรูปของพลังงานแสงเข้าไปยังตัวของมัน เพื่อเปลี่ยนให้เป็นพลังงานไฟฟ้า สัญญาณภาพดังกล่าวจะถูกส่งไปขยายให้แรงขึ้น แล้วนำไปผสมกับซิงโครไนซิงพัลส์ (Synchronizing Pulse) แล้วทำการผสมสัญญาณทางแอมพลิจูดกับคลื่นพาห้ภาพ (Picture Carrier) เพื่อให้ได้สัญญาณภาพเป็น เอเอ็ม. ส่วนสัญญาณเสียงจะถูกส่งเข้าสู่วงจรขยายสัญญาณก่อนที่จะผสมสัญญาณกับคลื่นพาห้เสียง เพื่อให้ได้สัญญาณเสียงเป็นเอฟเอ็ม. เครื่องรับโทรทัศน์จะรับสัญญาณจากสายอากาศเข้ามาทั้งภาพและเสียง นำสัญญาณนี้ไปทำการตีเทคเตอร์แยกภาพและเสียงออกไปใช้งาน โดยสัญญาณภาพนั้นจะถูกส่งไปจอหลอดภาพ (CRT) ซึ่งมีส่วนคล้ายหลอดออสซิลโลสโคปที่แก้วข้างหน้าเคลือบฉาบฟลูออเรสเซนต์ไว้ ภายในหลอดจะมีปืนอิเล็กตรอนที่มีการบังคับการบีบลำให้ยิงไปยังหน้าจอ ทำให้สารฟอสเฟอร์เกิดการเรืองแสงขึ้นที่จอ การที่เราต้องส่งสัญญาณทั้ง 5 ตัวออกอากาศแพร่คลื่นออกไปเพื่อวัตถุประสงค์ดังนี้

2.2.2.1 สัญญาณภาพและสัญญาณเสียง เป็นสัญญาณที่ส่งไปเพื่อให้เกิดภาพและเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2.2.2.2 สัญญาณแบลงกิ้ง เป็นสัญญาณที่ส่งเพื่อให้ลบเส้นสับกลับทั้งในแนวตั้งและแนวนอน
- 2.2.2.3 สัญญาณซิงโครไนซ์ เป็นสัญญาณที่ส่งมาเพื่อช่วยให้วงจรหักเหทางแนวตั้งและวงจรหักเหทางแนวนอน เพื่อให้เครื่องส่งกับเครื่องรับทำงานสอดคล้องตรงกัน
- 2.2.2.4 สัญญาณอ็ควอไลซิง เป็นสัญญาณที่ช่วยให้สัญญาณซิงโครไนซ์ทั้งแนวตั้งและแนวนอนยังคงรูปเดิมอยู่ได้ แม้ว่าจะเป็นการสแกนแบบสลับเส้นก็ตาม

2.2.3 การผสมสัญญาณ (Modulation System)



รูปที่ 2.3 การมอดูเลชัน

สัญญาณภาพและเสียงควรมีความห่างของความถี่ที่ไม่มากเกินไป ถ้าห่างมากจะทำให้ Bandwidth กว้างมาก ทำให้รับสัญญาณยาก มีระบบการผสมสัญญาณโทรทัศน์อยู่ 3 แบบคือ

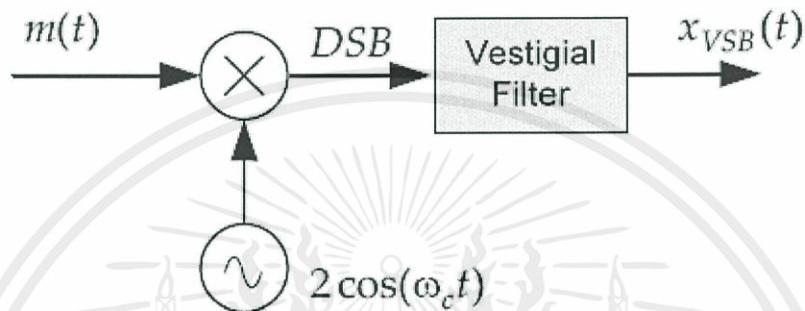
2.2.3.1 Amplitude Modulation ทำให้ Carrier มี Amplitude เปลี่ยนไปตามข้อมูล แต่ความถี่ของสัญญาณคงที่ สัญญาณภาพมีตัวประกอบหลายตัวทำให้ยุ่งยากในการมอดูเลชัน จึงใช้ระบบ Amplitude Modulation ข้อดีคือความถี่ของสัญญาณจะคงที่ การ Modulation นั้นง่าย แต่ข้อเสียคือเกิด Noise ได้ง่ายเพราะความถี่ในการส่งค่อนข้างต่ำ

2.2.3.1.1 ระบบเอเอ็มแบบธรรมดา

การส่งเอเอ็มแบบธรรมดานั้น Bandwidth จะกว้างมาก ทำให้ย่านความถี่ส่งคลื่นได้น้อยช่อง ดังนั้นจึงใช้เอเอ็มแบบ Vestigial Sideband ซึ่งต่างจากเอเอ็มธรรมดา นั่นคือ มอดูเลชันแบบเอเอ็มธรรมดาจะเกิดแถบความถี่ข้างเคียง Lower Sideband และ Upper Sideband ซึ่ง Sideband รวมทั้งหมดหากเป็นสัญญาณภาพจะได้ 10 MHz ถ้าหากเราจะลด Bandwidth ก็ทำได้โดยการเอา Sideband ออกไป 1 ด้าน ซึ่งระบบนี้จะตัด Carrier ที่ทำให้การ Decoder หรือ

Demodulation ซับซ้อนมากขึ้น และทำให้วิธีการ Demodulation สร้างความถี่ Carrier ขึ้นเอง เราจึงเอาระบบ Vestigial Sideband มาใช้

2.2.3.1.2 ระบบเอเอ็มแบบ Vestigial Sideband



รูปที่ 2.4 วิธีการ Vestigial Sideband

การมอดูเลตแบบแบนด์เดียว (Single Sideband) สามารถที่จะใช้แถบความถี่ได้อย่างมีประสิทธิภาพ แต่ข้อจำกัดที่ต้องใช้วงจรกรองความถี่ที่มีคุณสมบัติที่ดีมาก ๆ ซึ่งทำๆ ได้ยาก (หรือมีราคาแพง) ในทางปฏิบัติ ดังนั้นจึงมีการมอดูเลตอีกแบบคือแบบ Vestigial ที่จะอยู่ระหว่าง DSB และ SSB คือสัญญาณซีกแบนด์หนึ่งผ่านไปทั้งหมด ส่วนสัญญาณอีกซีกแบนด์หนึ่งผ่านไปบางส่วนทำให้สามารถที่จะส่งความถี่ต่ำมาก ๆ (เช่นสัญญาณภาพ) ได้ด้วย การมอดูเลตแบบนี้จึงไม่จำเป็นต้องใช้วงจรคัดกรองความถี่ที่ดีมาก และต้องการความกว้างของแบนด์ความถี่เพิ่มขึ้นอีกเล็กน้อย (ประมาณ 1.25 เท่าของสัญญาณ SSB) เพื่อให้การตีมอดูเลตด้วยวิธีการ Synchronous Detector

2.2.3.2 Frequency Modulation ทำให้ Carrier มีความถี่เปลี่ยนแปลงไปตามข้อมูล แต่ Amplitude คงที่ สัญญาณเสียงใช้ระบบ Frequency Modulation โดยมี Carrier เสียงห่างจาก Carrier ภาพ 4.5 MHz ในระบบ 525 เส้น และ 5.5 MHz ในระบบ 625 เส้น โดยทั่วไปจะให้ความถี่เสียงกว้าง 12 kHz

2.2.3.3 Phase Modulation ทำให้ค่าพารามิเตอร์ของข้อมูลเปลี่ยนแปลงไปตามเฟส Carrier

2.2.3.3.1 Pulse Amplitude Modulation (PAM) ใช้หลักการ Sampling

2.2.3.3.2 Pulse Code Modulation (PCM) ผสมความถี่เป็นสัญญาณ Pulse

2.2.3.3.3 Pulse Width Modulation (PWM) กำหนดความถี่ให้คงที่แต่ความกว้างของลูกคลื่น Pulse ไม่คงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้ 2.2.3.3.4 Pulse Position Modulation (PPM) ซึ่งใช้การจัดการข้อมูลโดยใช้ระยะเวลาความกว้างของลูกคลื่น

2.2.4 ความถี่ที่ใช้ในการส่งสัญญาณโทรทัศน์

การส่งสัญญาณสื่อสารด้วยระบบโทรทัศน์นั้นมีมาหลายปีแล้ว และได้เริ่มเข้ามาในประเทศไทยประมาณ พ.ศ. 2498 มาตรฐานการส่งโทรทัศน์ตามมาตรฐานสากลมีหลายระบบ ในแต่ละระบบก็มีใช้กันหลายกลุ่มประเทศ แต่ละกลุ่มก็ใช้กันในย่านความถี่ VHF และย่าน UHF โดยแต่ละระบบก็มีจำนวนช่องและความถี่ที่ใช้แตกต่างกัน โดยในระบบ VHF มีจำนวนช่องอยู่ประมาณ 12 ช่อง ส่วนในย่านความถี่ UHF จะมีจำนวนช่องอยู่ประมาณ 83 ช่อง ระบบใหญ่ๆที่ใช้ในการส่งทั่วโลกขณะนี้แบ่งเป็นสองระบบคือ ในระบบแรกเป็นระบบที่ใช้อักษรย่อว่า FCC เป็นส่วนหนึ่งของการส่งโทรทัศน์ในระบบย่อยตามมาตรฐานสากลของระบบความถี่มาตรฐาน ระบบนี้ใช้กันในอเมริกา ญี่ปุ่น เกาหลีใต้ ฟิลิปปินส์ ใต้หวัน พม่า กัมพูชา อาร์เจนตินา โบลิเวีย อูรุกวัย เป็นต้น ส่วนในอีกระบบหนึ่ง เป็นระบบ CCIR เป็นระบบการส่งทีวีส่วนหนึ่งที่ใช้ในประเทศแถบยุโรป โดยทั้งระบบ FCC และ CCIR แสดงให้เห็นดังตารางต่อไปนี้

ตารางที่ 2.1 ความถี่วิทยุในประเทศไทย

สถานีวิทยุ	ช่องความถี่	ความยาวเฉลี่ย
เอเอ็ม	550-1600 kHz	360 เมตร
เอฟเอ็ม	88-108 MHz	3 เมตร
คลื่นสั้น	3-12 MHz	62.5 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ตารางแสดงความถี่การส่งทีวีระบบ FCC ย่าน VHF

แถบความถี่	ช่อง	ความถี่ (MHz)	คลื่นพาห้(Carrier)		ความยาวคลื่น (นิ้ว)	ช่องกว้างความถี่ % of FM	
			สัญญาณภาพ (Video:MHz)	สัญญาณเสียง (Audio:MHz)			
(VHF Low Band)	2	54-60	52.25	59.75	207.5	10.5	
	3	60-66	61.25	65.75	187.5	9.5	
	วีเอชเอฟแถบล่าง	4	66-72	67.25	71.75	171.5	8.7
		5	76-82	77.25	81.75	149.5	7.6
		6	82-88	38.25	87.75	139	7.1
วีเอชเอฟแถบบน (High Band)	7	174-180	175.25	179.75	66.73	3.39	
	8	180-186	181.25	185.75	64.54	3.28	
	9	186-192	187.25	191.75	62.49	3.18	
	10	192-198	193.25	197.75	60.58	3.08	
	11	198-204	199.5	203.75	58.78	2.99	
	12	204-210	205.25	209.75	57.06	2.9	
	13	210-266	211.25	215.75	55.45	2.82	

ตารางที่ 2.3 ตารางแสดงความถี่การส่งทีวีระบบ CCIR ย่าน VHF

แถบความถี่	ช่อง	ความถี่ (MHz)	คลื่นพาห้ (Carrier)		
			สัญญาณภาพ (Video:MHz)	สัญญาณเสียง (Audio:MHz)	
(VHF Low Band)	2	47-54	48.25	53.75	
	วีเอชเอฟแถบล่าง	3	54-61	55.25	60.75
		4	61-68	62.25	67.75
วีเอชเอฟแถบบน (High Band)	5	174-181	175.25	180.75	
	6	181-188	182.25	187.75	
	7	188-195	188.25	194.75	
	8	195-202	196.25	201.75	
	9	202-209	203.25	208.75	
	10	209-216	210.25	215.75	
	11	216-223	217.25	222.75	
12	223-230	224.25	229.75		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการอ้างอิงเท่านั้น ไม่สามารถนำเอกสารนี้ไปใช้
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่เอกสารนี้โดยไม่ได้รับอนุญาตให้นำไปใช้

ตารางที่ 2.4 ตารางแสดงความถี่การส่งทีวีระบบ CCIR ย่าน UHF

แถบความถี่	ช่อง	ความถี่ (MHz)	คลื่นพาห์ (Carrier)	
			สัญญาณภาพ (Video:MHz)	สัญญาณเสียง (Audio:MHz)
UHF	21	470-477	471.25	476.75
	22	478-485	479.25	484.75
	23	486-493	487.25	492.75
	24	494-501	495.25	500.75
	25	502-509	503.25	516.75
	26	510-517	511.25	524.75
	27	518-525	519.25	532.75
	28	526-533	527.25	540.75
	29	534-541	535.25	548.75
	30	542-549	543.25	556.75
	31	550-557	551.25	564.75
	32	558-565	559.25	572.75
	33	566-573	567.25	580.75
	34	574-581	575.25	588.75
	35	582-589	583.25	596.75
	36	590-597	591.25	604.75
	37	598-605	599.25	612.75
	38	606-613	607.25	620.75
	39	614-621	615.25	628.75
	40	622-629	623.25	636.75
	41	630-637	631.25	644.75
”	”	”	”	
”	”	”	”	
83	884-890	885.25	889.75	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 EasyCAP USB 2.0 Audio/Video Capture

สายสัญญาณ วิดีโอ/ออดิโอ ที่นิยมใช้ในปัจจุบันนี้มีอยู่หลายประเภทดังนี้

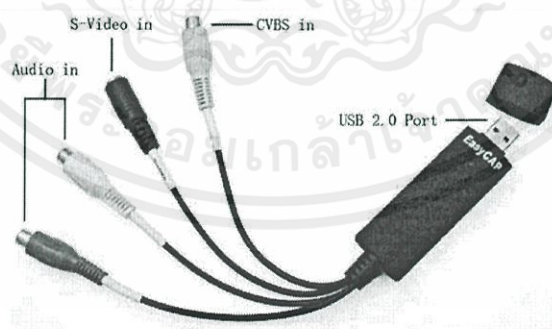
1. สาย S-Video สาย S-Video (ย่อมาจาก Separate Video) หรือบางที่เรียกว่า Y/C เป็นสายสัญญาณวิดีโอแบบอนาล็อก ซึ่งใช้ส่งผ่านสัญญาณ Video Data โดยแยกออกเป็น 2 สัญญาณ คือ ความสว่าง และสี (Brightness and Color) และเหมาะสำหรับใช้กับ Standard Definition Video ที่มี Bandwidth 480i หรือ 576i

2. สาย Component Video สาย Component Video ที่นิยมใช้กันแพร่หลาย เป็นสายที่ส่งผ่านสัญญาณวิดีโอซึ่งแบ่งออกเป็น 3 สัญญาณ เป็นวิธีแยกสัญญาณวิดีโอออกเพื่อลดการรบกวน ซึ่งกันและกัน สาย Component Video สามารถส่งผ่านสัญญาณได้ เช่น 480i, 480p, 576i, 576p, 1080i และ 1080p

3. สาย Composite Audio/Video เป็นสายสัญญาณภาพและเสียงที่เราคุ้นเคยกันมาก คือ สายสีเหลือง สำหรับสัญญาณภาพ และสายสีแดงกับสีขาว สำหรับสัญญาณเสียง ซึ่งใช้ต่อจากเครื่องเสียงทั่วไป เช่น เครื่องเล่นดีวีดี ทีวี เป็นต้น

4. สายสัญญาณ HDMI เป็นสายที่สามารถส่งได้ทั้งภาพและเสียง โดยส่งผ่าน Video Resolutions จาก 480i ขึ้นไปถึง 1080p

ในตัวอุปกรณ์ EasyCAP USB 2.0 Audio/Video Capture จะใช้สาย Composite Audio/Video ในการรับสัญญาณภาพและเสียงและแปลงสัญญาณส่งเข้าสู่ USB 2.0 เพื่อเข้าคอมพิวเตอร์ หรืออุปกรณ์อื่นๆต่อไป

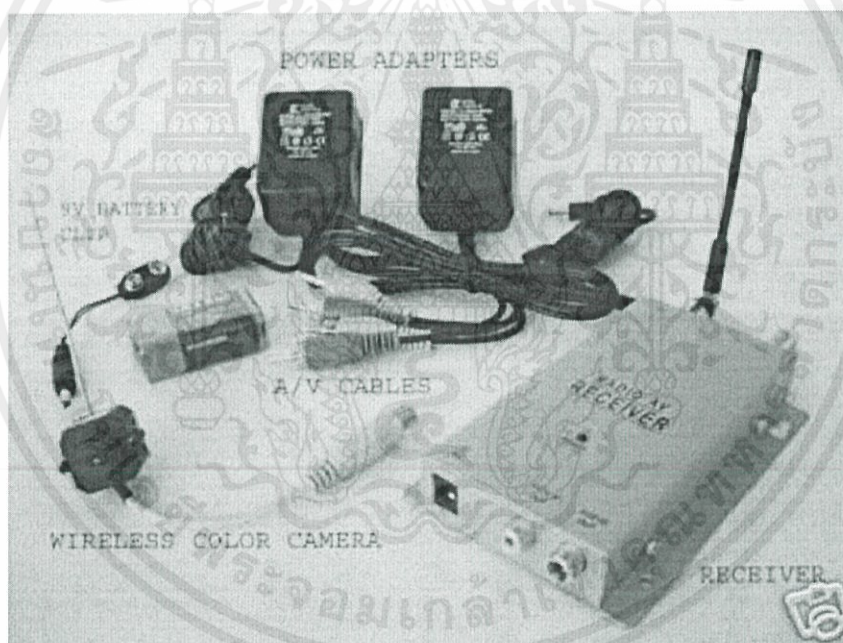


รูปที่ 2.5 EasyCAP USB 2.0 Audio/Video Capture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 กล้องไร้สาย (Wireless Camera)

กล้องไร้สายเป็นกล้องที่มีการส่ง ภาพวิดีโอและสัญญาณเสียงโดยใช้สัญญาณวิทยุแบบไร้สาย กล้องวงจรปิดที่ใช้รักษาความปลอดภัยที่มีการเชื่อมต่อแบบไร้สายนี้ อย่างน้อยก็ต้องมีการใช้สายเคเบิลหรือสายสำหรับส่งสัญญาณวิดีโอ และเสียงอยู่ในตัวกล้องวงจรปิดด้วย แต่กล้องวงจรปิดแบบไร้สายนี้จะใช้แบตเตอรี่เป็นตัวขับเคลื่อน แทนไฟฟ้า ถ้าเรามองโดยรวม ๆ แล้ว จะทำให้กล้องวงจรปิดตัวนี้เป็นกล้องวงจรปิดแบบไร้สายอย่างแท้จริง กล้องวงจรปิดแบบไร้สายได้พิสูจน์แล้วว่า เป็นกล้องวงจรปิดที่ได้รับความนิยมในหมู่ผู้บริโภคที่ชื่นชอบ การรักษาความปลอดภัยที่ทันสมัย เนื่องจากมีค่าใช้จ่ายในการติดตั้งที่ไม่แพง (สำหรับผู้ที่มีความจำเป็น ต้องใช้ แต่มีงบประมาณที่น้อย) และสามารถเลือกการติดตั้งได้หลากหลายรูปแบบ กล้องวงจรปิดแบบไร้สายนี้สามารถติดตั้งหรือไม่ต้องการติดตั้งร่วมกับกล้องวงจรปิดแบบใช้สายได้



รูปที่ 2.6 กล้องไร้สาย (Wireless Camera)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 ประเภทของกล้องไร้สาย

1. กล้องไร้สายแบบอะนาล็อก (Analog Wireless Camera)

การส่งสัญญาณภาพและ เสียงโดยใช้คลื่นความถี่วิทยุ โดยปกติอะนาล็อกไร้สายจะมีช่วงการส่งสัญญาณประมาณ 300 ฟุต (91 เมตร) ในพื้นที่เปิด,ผนัง,ประตู, เพอร์นิเจอร์ จะลดช่วงการส่งสัญญาณ ถ้ามีสิ่งกีดขวางจำนวนมาก กล้องที่พบจะมีความถี่ 900 MHz, 2.4 GHz และ 5.8 GHz ปัจจุบันส่วนใหญ่กล้องวงจรปิดแบบไร้สาย (Wireless Security CCTV) จะทำงานบนความถี่ 2.4 GHz ซึ่งของใช้ในครัวเรือน โทรศัพท์ไร้สาย วีดีโอเกมส์และไมโครเวฟ จะทำงาน บนความถี่ 2.4 GHz และอาจทำให้เกิดการรบกวนกับสัญญาณของกล้องวงจรปิดได้ ส่วนที่เรารู้จักกันในนามของ Wi-Fi ที่ทำงานบนความถี่ 900 MHz เป็นสัญญาณของอินเทอร์เน็ตไร้สายจะเป็นสัญญาณที่เป็นมิตร เพราะมันจะไม่เกี่ยวข้องกับสัญญาณของกล้องวงจรปิด ข้อดี คือ สัญญาณจากกล้องวงจรปิดสามารถเรียก ขึ้นมาดูได้จากสถานที่ต่าง ๆ ด้วยอินเทอร์เน็ตไร้สายของคุณ ข้อเสีย คือ ไวต่อการรบกวนจากอุปกรณ์ที่ ใช้ในครัวเรือน เช่น ไมโครเวฟ โทรศัพท์ไร้สาย วีดีโอเกมส์ และเราเตอร์

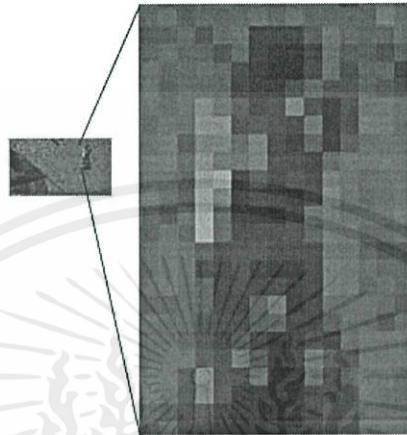
2. กล้องไร้สายแบบดิจิตอล (Digital Wireless Camera)

กล้องแบบนี้จะมีการแปลงสัญญาณเสียงและวีดีโอ แบบอะนาล็อกให้เป็นดิจิตอลโดยการเข้ารหัสโดยมีข้อดี คือ

- ส่งสัญญาณได้ทั้งใกล้และไกลถึง 450 ฟุต
- วีดีโอและเสียงมีคุณภาพสูง
- มีการสื่อสารได้ 2 ทาง ระหว่างกล้องวงจรปิดและเครื่องรับสัญญาณ
- คุณสามารถใช้สัญญาณดิจิตอลส่งคำสั่งและฟังก์ชันได้ ไม่ว่าจะไฟจะติดหรือดับ
- คุณสามารถเชื่อมต่อไปยังอุปกรณ์รับได้มากกว่า 1 ตัว เช่นเดียวกับเครื่องบันทึก DVR

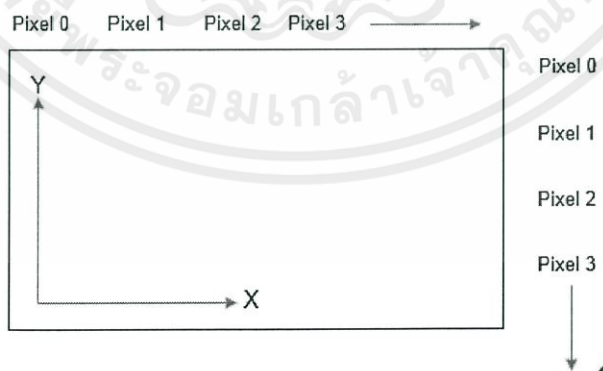
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 หน่วยความละเอียดของภาพ (Pixel)



รูปที่ 2.7 ภาพการขยายรูปจนเห็นจุดพิกเซล

Pixel เป็นคำย่อมาจาก Picture Element (PEL) เป็นหน่วยที่เล็กที่สุดของภาพที่คอมพิวเตอร์สามารถพิมพ์หรือแสดงได้ จากรูปแสดงถึงการขยายภาพ 1600% จนสามารถเห็นเป็นบล็อกสี่เหลี่ยม โดยในแต่ละบล็อกจะประกอบด้วยสีเพียงสีเดียวเท่านั้น ไม่มีการผสมจึงเรียกว่า 1 จุดพิกเซล สำหรับจอที่มีจำนวนพิกเซลมาก จะมีความละเอียดของภาพมาก โดยส่วนมากจะระบุความละเอียดในหน่วยของพิกเซล แนวนอน x แนวตั้งเช่น 1366x768 หมายความว่าในแนวนอนหรือแนวแกน X จะประกอบไปด้วยจำนวนพิกเซล 1366 จุด และแนวตั้งหรือแนวแกน Y จะประกอบไปด้วยจำนวนพิกเซล 768 จุด โดยทั่วไปจะเราสามารถระบุตำแหน่งของจุดพิกเซลได้ตามแกนดังรูป 2.8

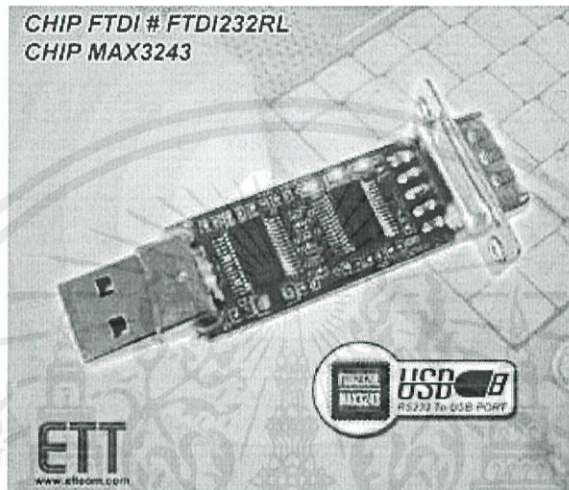


รูปที่ 2.8 แสดงการระบุตำแหน่งของจุดพิกเซลในภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ET USB/RS232 MINI

เป็นตัวเชื่อมต่อระหว่างคอมพิวเตอร์กับอุปกรณ์อื่นที่เป็นอุปกรณ์พอร์ต DB9 ผ่าน Port USB ของคอมพิวเตอร์ คือสามารถเสียบกับ Port USB ของคอมพิวเตอร์ได้เลยไม่ต้องต่อกับพอร์ต DB9



รูปที่ 2.9 ET USB/RS232 MINI

2.6.1 คุณสมบัติของ ET USB/RS232 MINI

1. รองรับมาตรฐาน USB 1.1 และ USB 2.0
2. อัตรารับส่งข้อมูล 300 bps ถึง 128 kbps
3. ใช้ไฟเลี้ยงจาก USB Port โดยตรงไม่ต้องต่อเพิ่มภายนอก
4. USB Connector แบบ TYPE A, RS232 Connector แบบ DP9 ตัวผู้
5. แสดงสถานะการทำงานด้วย LED 3 สี คือ การรับ(Rx) สีเขียว, การส่ง(Tx) เหลือง และ Power (PWR) สีแดง

2.6.2 การต่อใช้งาน ET-USB/RS232 MINI

1. เสียบ ET-USB/RS232 MINI เข้ากับพอร์ต USB ของคอมพิวเตอร์ขณะนี้จะสังเกตเห็นว่าไฟ แสดงสถานะ PWR จะยังไม่ติดสว่าง
2. ติดตั้ง Driver สำหรับ ET-USB/RS232 MINI โดยจากคู่มือการติดตั้ง Driver เมื่อติดตั้งเรียบร้อยแล้ว ไฟแสดงสถานะ PWR จะติดสว่างแสดงว่าพร้อมที่จะใช้งานแล้ว
3. ต่อสายอนุกรม เพื่อใช้งานโดยตำแหน่งขาใช้งาน จะเป็นดังรูป 2.10

RS-232 DB-9 Male Pinout

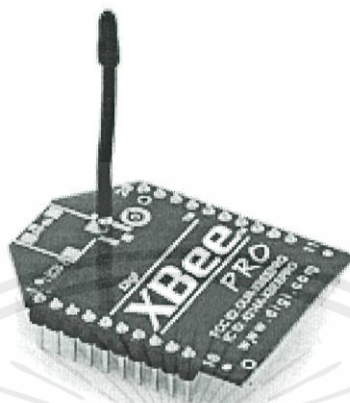


รูปที่ 2.10 การต่อใช้งาน ET-USB/RS232 MINI

2.7 XBee

XBee เป็นอุปกรณ์ที่มี Microcontroller และ RF IC อยู่ภายใน ทำหน้าที่เป็นอุปกรณ์ Transceiver (อุปกรณ์รับ-ส่งสัญญาณ) แบบ แบบ Half Duplex ย่านความถี่ 2.4 GHz มีการจัดการโดยใช้พลังงานต่ำ ใช้งานง่าย มี Interface ที่ใช้รับและส่งข้อมูลกับ XBee เป็น UART (TTL) ซึ่งสำหรับทางด้านไมโครคอนโทรลเลอร์ เราสามารถนำขาที่ใช้ติดต่อสื่อสาร UART ของ XBee ต่อเข้ากับ UART ของ ไมโครคอนโทรลเลอร์ ได้โดยตรง XBee สามารถใช้งานตามมาตรฐาน ZigBee ได้ โดยที่ไม่ต้องเขียนโปรแกรมสร้าง เครือข่าย ZigBee เพราะทางผู้ผลิตได้จัดทำ firmware ที่จะโหลดเข้าไปในตัว XBee ให้สามารถ Set Parameter ผ่าน Software Interface (X-CTU หรือโปรแกรมที่เขียนขึ้น) , ผ่านทาง At Command (เหมือนกับการควบคุม GSM Module) โดยใช้ Hyper Terminal หรือ ผ่านทางการรับส่งข้อมูลด้วยไมโครคอนโทรลเลอร์ ได้อย่างง่ายดาย โดยเมื่อ Set XBee ให้ทำงานเป็นอุปกรณ์ในเครือข่าย ZigBee แล้ว เราจะเรียก XBee แต่ละตัวว่าเป็น Node โดย Firmware ที่ใช้กับ XBee จะใช้โหลดผ่านโปรแกรม ทั้งนี้ XBee แต่ละรุ่น จะสามารถ Setting Function การใช้งานได้มากมาย ทำให้ Firmware ที่จะต้องโหลดเข้าไปนั้น มีมากมายหลายแบบ เราต้องเลือกให้เหมาะสมกับการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 XBee รุ่นสายอากาศแบบ Wire (Whip)

2.7.1 คุณสมบัติของ XBee

1. Operating Frequency ISM Band 2.4 Ghz
2. มีสายอากาศหลายชนิด คือ แบบ Chip Ant , Whip Ant , UFL con , RPSMA con
3. Supply Voltage 2.8-3.4 V
4. Power Down Current < 10uA
5. มี RF data rate อยู่ที่ 250 KBpS
6. มี Serial interface data rate ระหว่าง 1200 – 115200 BpS
7. เป็น Spread Spectrum ชนิด DSSS (Direct Sequence Spread Spectrum)
8. การกำหนด addressing มีลำดับลักษณะคือ กำหนด PAN ID สำหรับเครือข่ายหนึ่ง กำหนด Channel และ กำหนด Address ของแต่ละตัว

2.7.2 สายอากาศของ XBee

XBee ใช้ย่านความถี่ 2.4 Ghz ซึ่งเป็นย่านเดียวกันกับ Bluetooth หรือ Wireless Lan ดังนั้นในการใช้งานเมื่อมีอุปกรณ์อื่นๆที่ใช้สัญญาณ Bluetooth หรือ Wireless อยู่ในระยะของสัญญาณ อาจทำให้มีการรบกวนเกิดขึ้น ในช่วงความถี่ 2.4 Ghz เป็นย่านคลื่นไมโครเวฟซึ่งการวางตำแหน่งของอุปกรณ์ต้องคำนึงถึงหลักการ LOS (Line Of Sight) จึงจะได้กำลังส่งสูงสุด ระยะทางในการส่งสัญญาณของ XBee นั้นจะแสดงได้ตามตารางที่ 2.5

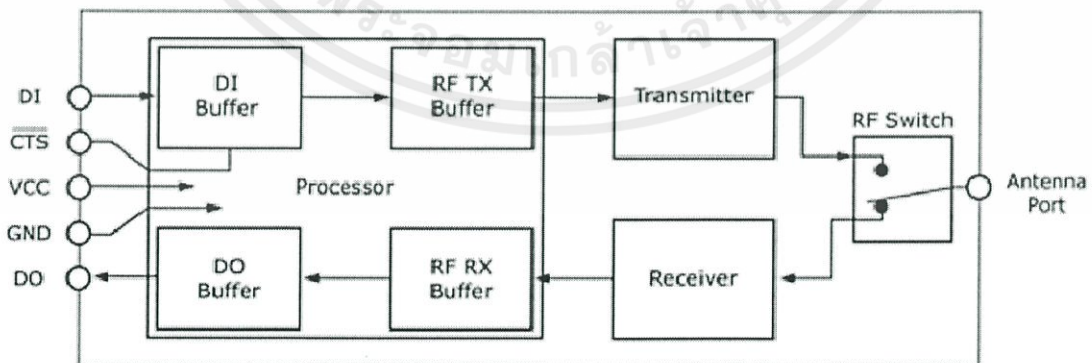
เอกสารนี้เป็นเอกสารลิขสิทธิ์ของ บริษัท การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 ระยะทางสัญญาณ XBee ที่การใช้งานต่างๆ

Module	Antenna Type	Outdoor Distance (Visual Line-of-Sight)	Indoor Distance (Office Building)	Indoor Distance (Warehouse)
XBee	Chip	470 ft. (143 m)	80 ft. (24 m)	-
	Whip	845 ft. (258 m)	80 ft. (24 m)	84 ft. (26 m)
XBee-PRO	Chip	1690 ft. (515 m)	140 ft. (43 m)	-
	Whip	4382 ft. (1335 m)	140 ft. (43 m)	355 ft. (108 m)

2.7.3 Data Throughput ของ XBee

โดยทั่วไปการใช้งาน RF Module จะต้องมี Buffer ด้วย เพื่อเป็นการปรับอัตรารับส่งข้อมูลระหว่างตอนที่รับส่งทางอากาศ กับตอนที่รับส่งไปยัง ไมโครคอนโทรลเลอร์ หรือ อุปกรณ์อื่นๆ ได้อย่างเหมาะสม Data ที่รับส่งระหว่าง MCU กับ XBee จะมีข้อจำกัด เรื่อง Packet อาจถูก Drop ได้ เนื่องจากจาก Data Over Flow โดยสำหรับด้านการส่งข้อมูลไปที่ XBee เพื่อออกอากาศนั้น ที่ขา DI จะมี Buffer อยู่ประมาณ 202 Bytes หากส่งเกิน Buffer จะเกิดการ Drop packet ที่ขา DI ซึ่งทางภาครับข้อมูล ที่ขา DO ก็มี Buffer อยู่เช่นกัน โดยจะมี Parameter ที่เกี่ยวข้องกับ Data Throughput คือ RO และ BD โดย ค่า RO คือค่า Packetization Timeout ซึ่งเป็น delay ของข้อมูลที่อยู่ใน DI Buffer ก่อนที่จะถูก Encapsulate ไปที่ส่วน RF Transmission เพื่อส่งข้อมูลออกอากาศ หากตั้ง RO = 0 Data ที่รับเข้ามาจาก MCU จะถูก XBee Encapsulate Packet ส่ง ออกอากาศทันที ดังนั้นเราจะมี Parameter RO และ BD ที่ช่วยในการปรับ Data รับส่งให้สามารถรับส่งกันได้ทัน ไม่ให้มีการ Drop Packetได้ในกรณีที่ส่งข้อมูลเกิน 200 Bytes



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 2.12 Data Throughput ของ XBee
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 Arduino MEGA 2560

ET-MEGA2560 ADK เป็นบอร์ดรุ่นใหม่ ในตระกูลของ Arduino โดยเพิ่มความสามารถ ในการเชื่อมต่อสื่อสาร สั่งงานบอร์ด Arduino ผ่าน อุปกรณ์ Android OS ต่างๆ เช่น โทรศัพท์มือถือ หรือ แท็บเล็ต นับจากที่ โครงการพัฒนาระบบ Arduino ของ MCU AVR ที่เป็นแบบ OPEN SOURCE ได้รับการพัฒนาสู่สาธารณะก็ได้รับความนิยมเป็นอย่างมาก โดยได้มีการพัฒนาทั้ง SOFTWARE และ HARDWARE จากรุ่นเล็กที่ใช้ MCU 28 ขา เบอร์ ATMEGA88/168/328 (ET-BASE AVR EASY88/168/328) หรือ MCU ที่มีขนาดใหญ่ขึ้น เป็นขนาด 100 ขา เบอร์ ATMEGA1280 (ET-EASYMEGA1280) ทาง อีทีที ได้นำ MCU ตระกูล AVR เบอร์ ATMEGA2560 และ MAX3421 มา พัฒนาเป็นบอร์ด Arduino แบบมี USB HOST เพื่อรองรับการเชื่อมต่ออุปกรณ์ USB DEVICE และ อุปกรณ์ Android OS โดยออกแบบ ให้มีการจัดสรร PIN I/O ต่างๆ รวมทั้งขนาดบอร์ดตามมาตรฐาน ของบอร์ด Arduino MEGA



รูปที่ 2.13 Arduino MEGA 2560

คุณสมบัติของบอร์ด ET-MEGA2560 ADK

1. ใช้ ATMEGA2560 เป็น MCU ประจำบอร์ด, RUN ความถี่ 16 MHz, 100PIN TQFP
2. 256 KBYTE FLASH ในการใช้เขียนโปรแกรม (8 KBYTE สำหรับ BOOTLOADER) / 8 KBYTE SRAM / 4 KBYTE EEPROM
3. รองรับการพัฒนาโปรแกรมด้วยภาษา C++ ของ Arduino ตามแบบ Arduino MEGA ได้ 100%, รองรับ การทำงานบน คอมพิวเตอร์ พีซี ทั้ง WINDOWS, LINUX และ MACINTOSH OSX

4. ขั้วต่อ USB MINI ใช้ USB BRIDGE ของ FTDI เบอร์ FT232RL สำหรับใช้ ติดต่อสื่อสารและDOWNLOAD ข้อมูล CODE ที่เขียนจากคอมพิวเตอร์มายังตัวบอร์ดโดยไม่ต้องซื้อ

อุปกรณ์บอร์ด DOWNLOAD ใดๆ เพิ่มเติม, เพิ่มข้อต่อ 6 PIN สำหรับปรับใช้งานPROGRAM ตรงเข้า MCU โดยไม่ต้องผ่านโปรแกรม BOOTLOADER ของ Arduino

5. ON BOARD USB HOST (ใช้ MAX3421) สำหรับการเชื่อมต่อ USB DEVICE

- รองรับการพัฒนาด้วย ADK (Android Open Accessories Development Kit) เมื่อใช้กับ Android
- รองรับการพัฒนาด้วย ADB (Android Debug Bridge) ของ Microbridge เมื่อใช้กับ Android V1.5 หรือสูงกว่า

2.9 STEPPER MOTOR

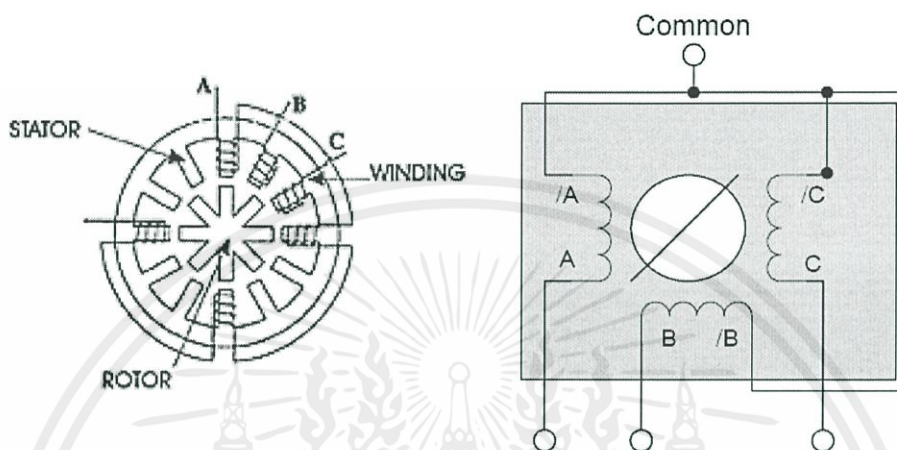
สเต็ปเปอร์มอเตอร์ เป็นมอเตอร์ที่มีการหมุนเป็นสเต็ป (STEP) เมื่อจ่ายกระแสไฟฟ้าไปยังขดลวดที่พันอยู่บนสเตเตอร์(stator) ในลักษณะเป็นพัลส์ (Pulse) ในกรณีที่จ่ายกระแสไฟฟ้าให้กับขดลวดขดนั้นๆ ของสเต็ปเปอร์มอเตอร์ตลอดเวลา ก็จะเกิดการหมุนเพียงหนึ่งสเต็ปเท่านั้นซึ่งต่างจาก DC มอเตอร์ เมื่อจ่ายกระแสไฟฟ้าให้ ก็จะหมุนตามจนกว่าจะหยุดจ่ายกระแสไฟฟ้าให้ โดยข้อดีของสเต็ปเปอร์มอเตอร์คือ

1. สามารถใช้ในงานควบคุมตำแหน่งในลักษณะ วงจรควบคุมแบบเปิด (Open Loop Control) ได้โดยไม่ต้องการสัญญาณป้อนกลับ (Feedback Signal) แต่อาศัยการนับจำนวนของพัลส์ที่ส่งไปควบคุมการหมุนแทน
2. ไม่มีส่วนของแปรงถ่านที่จะสึกหรอและไม่เกิดการสปาร์คที่แปรงถ่านซึ่งอาจก่อให้เกิดสัญญาณรบกวน

2.9.1 ชนิดของสเต็ปเปอร์มอเตอร์

2.9.1.1 วารีเอเบิลรีลักแตนซ์ (Variable Reluctance : VR) โรเตอร์(Rotor) ทำด้วยเหล็กอ่อน รูปทรงกระบอกและทำเป็นลักษณะฟัน (Teeth) สเตเตอร์(Stator) จะมีขดลวดพัน และจะทำเป็นลักษณะของฟันเช่นกัน เมื่อจ่ายกระแสไฟฟ้าให้กับขดลวดที่สเตเตอร์จะเกิดเป็นขั้วแม่เหล็กที่ฟันของสเตเตอร์และเหนี่ยวนำ ทำให้ฟันของโรเตอร์เกิดเป็นขั้วแม่เหล็กที่มีขั้วตรงกันข้ามกับสเตเตอร์ ทำให้ดึงดูดกันเกิดการหมุนของโรเตอร์ขึ้น มอเตอร์ชนิดนี้โดยปกติจะมีขนาด 3 เฟส ในบางครั้งอาจพบถึง 4 เฟส มอเตอร์ชนิดนี้ถ้าไม่จ่ายกระแสไฟฟ้าให้กับขดลวดบนสเตเตอร์ ตัวโรเตอร์จะไม่เกิดแรงดึงดูดกับสเตเตอร์ มอเตอร์ชนิดนี้ไม่นิยมนำไปใช้ในงานอุตสาหกรรมแต่จะถูกนำไปใช้กับงานที่มีขนาดเล็ก เช่น Micro-Positioning Table เป็นต้น เพราะไม่มีส่วนที่เป็นแม่เหล็กถาวรดังนั้นในขณะไม่จ่ายกระแสไฟฟ้าให้กับขดลวดที่สเตเตอร์จึงไม่เกิดแรงดึงดูด วิธีการขับ (Driving) หรือการกระตุ้นเฟส (Phase Excitation) จะทำดังนี้คือ ต่อปลายด้าน Common เข้ากับแหล่งจ่ายไฟขั้วบวก(+) แล้วทำการสวิตซ์ให้ปลายด้าน A , B ,

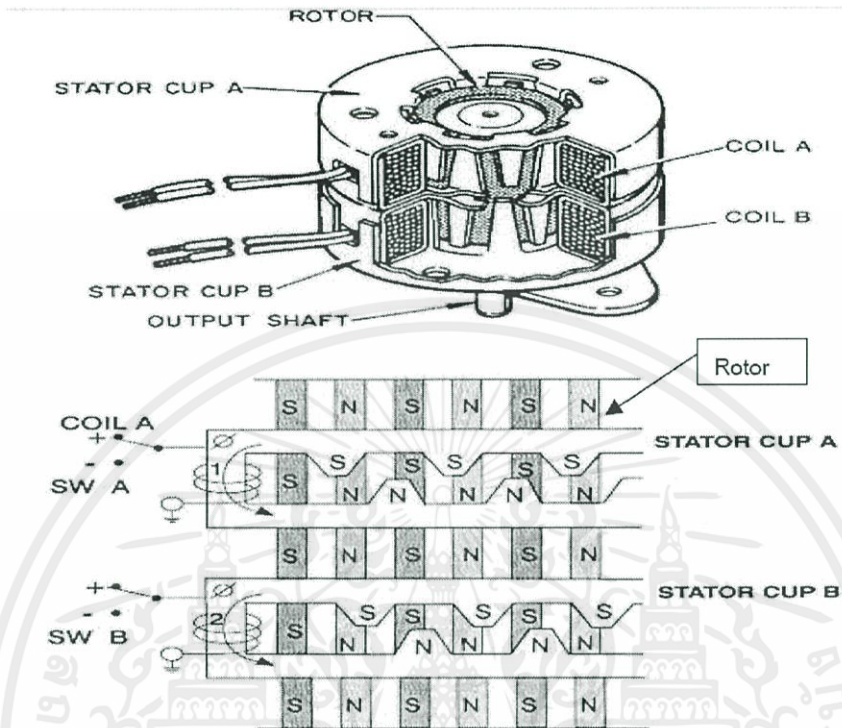
C ต่อลงกราวด์ (Ground) ตามลำดับ ทีละปลายแล้วทำเช่นนี้เรื่อยไป แต่ถ้าต้องการให้หมุนกลับก็สวิตช์ย้อนกลับ



รูปที่ 2.14 โครงสร้างสเต็ปเปอร์มอเตอร์ชนิดวาริเอเบิลรีลักแตนซ์และสัญลักษณ์

2.9.1.2 (Permanent Magnet : PM) โรเตอร์(Rotor) ทำ ด้วยแม่เหล็กถาวรรูปทรงกระบอกเรียบ สเตเตอร์(Stator) จะมีขดลวดพัน และก็จะทำเป็นฟัน เมื่อจ่ายกระแสไฟฟ้าให้กับขดลวดที่สเตเตอร์จะเกิดเป็นขั้วแม่เหล็กที่ฟันของสเตเตอร์และจะดึงดูดกับขั้วของแม่เหล็กถาวรที่โรเตอร์ทำให้เกิดการหมุนของโรเตอร์ขึ้นมอเตอร์ชนิดนี้โดยจะมีตั้งแต่ขนาด 2 เฟสขึ้นไปมอเตอร์ชนิดนี้ไม่นิยมนำไปใช้ในงานอุตสาหกรรมแต่จะถูกนำไปใช้กับอุปกรณ์คอมพิวเตอร์เช่นตัวขับวงล้อที่ใช้หมุนเพื่อเลื่อนกระดาษของเครื่องพิมพ์ เป็นต้น เพราะความเร็วต่ำ แรงบิดต่ำ และนอกจากนี้ด้วยโครงสร้างของมอเตอร์ชนิดนี้ทำให้มุมที่หมุนไปแต่ละสเต็ป (StepAngle) ไม่ละเอียดเช่น สเต็ปละ 3.6 , 7.5 , 15 , 18 องศาเป็นต้น มอเตอร์ชนิดนี้ถึงไม่จ่ายกระแสไฟฟ้าให้กับขดลวดบนสเตเตอร์(Stator) ตัวโรเตอร์จะเกิดแรงดึงดูดกับสเตเตอร์ซึ่งเกิดจากอำนาจของแม่เหล็กถาวรที่โรเตอร์ทำให้หมุนได้ยาก จำนวนขั้วแม่เหล็กที่โรเตอร์สามารถนับได้จากจำนวนขั้วแม่เหล็กที่จะเกิดขึ้นจากกระแสไฟฟ้าไหลผ่านขดลวดที่สเตเตอร์ชุดใดชุดหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



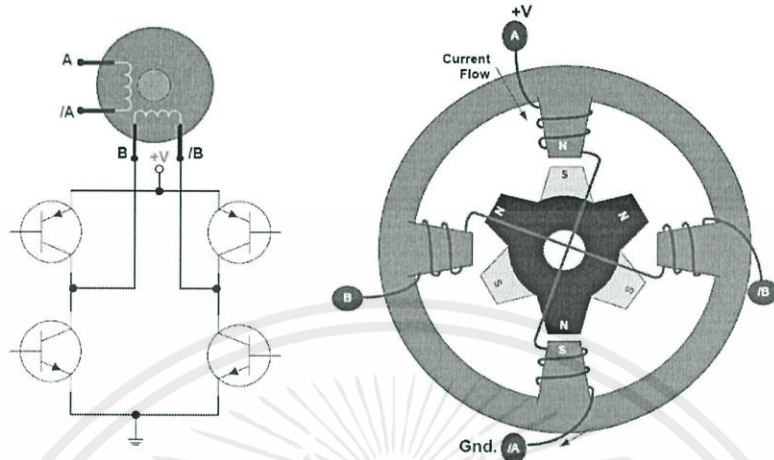
รูปที่ 2.15 สเต็ปเปอร์มอเตอร์ชนิดเพอร์มาเนนต์แมกเน็ต

2.9.1.3 แบบผสม (Hybrid : HB) ใช้หลักการทำงานของทั้งสองแบบมาออกแบบโดยที่ สเตเตอร์จะคล้ายกับแบบ VR ส่วนโรเตอร์จะคล้ายแบบ PM แต่จะให้เป็นฟันมอเตอร์แบบนี้นิยมใช้ในงานอุตสาหกรรมเพราะแรงบิดสูงความละเอียดของสเต็ปในการหมุนสูง, ความเร็วสูงกว่าสองแบบที่กล่าวมาแล้ว มอเตอร์ชนิดนี้โดยปกติจะมีขนาด 2 เฟส ถึง 5 เฟส และมอเตอร์ชนิดนี้ได้มีการพัฒนาให้มีประสิทธิภาพเหนือกว่าเดิมไปอีกโดยให้ชื่อว่า “Enhanced Hybrid”

2.9.2 ชนิดของสเต็ปเปอร์มอเตอร์แบ่งตามลักษณะสายที่ใช้ต่อกับวงจรขับ

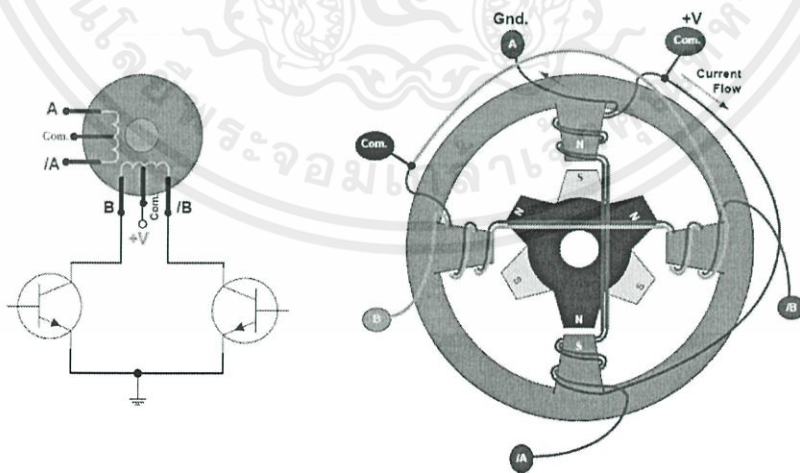
2.9.2.1 แบบไบโพลาร์ (Bipolar) ขดลวดที่สเตเตอร์แต่ละชุดจะไม่มีจุดร่วม การต่อเข้ากับวงจรขับจะใช้ปลายทั้งสองด้านของขดลวดแต่ละชุด การทำให้เกิดขั้วแม่เหล็กที่สเตเตอร์ทำได้โดยการจ่ายกระแสไฟจากปลายด้านหนึ่งไปยังปลายอีกด้านหนึ่งของขดลวดและการเปลี่ยนขั้วแม่เหล็กที่สเตเตอร์ชุดเดียวกันนี้ก็ได้โดยสลับทิศทางกระแสไฟฟ้านั้นเอง ดังนั้นวงจรขับที่ใช้จึงจำเป็นต้องสามารถกลับทิศทางกระแสไฟได้ กรณีเป็นมอเตอร์ 2 เฟสจะมีสายที่ใช้ต่อกับวงจร 4 สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 สัญลักษณ์โครงสร้างและวงจรขับที่ใช้กับมอเตอร์แบบโพล่า 2 เฟส

2.9.2.2 แบบยูนิโพล่า (Unipolar) ขดลวดที่สเตเตอร์แต่ละชุดจะมีจุดร่วม การพันขดลวดจะพันในแบบ Bifilar การต่อเข้ากับวงจรขับจะใช้ปลายของขดลวดแต่ละด้านต่อเข้ากับวงจรขับ และใช้จุดร่วมต่อเข้ากับขั้วบวกของแหล่งจ่ายไฟ การทำให้เกิดขั้วแม่เหล็กที่สเตเตอร์ทำได้โดยการจ่ายกระแสไฟให้ไหลจากจุดร่วมลงกราวด์มาครบวงจรที่ปลายด้านหนึ่งของขดลวด การเปลี่ยนขั้วแม่เหล็กที่สเตเตอร์ชุดเดียวกันนี้ก็ได้โดยเปลี่ยนการจ่ายกระแสไฟฟ้าจากขดหนึ่งไปยังอีกขดหนึ่งของขดลวดที่พันอยู่บนสเตเตอร์ชุดเดียวกัน ดังนั้นวงจรขับจึงเป็นวงจรสวิตช์เพื่อทำให้จ่ายกระแสไฟที่ผ่านขดลวดครบวงจรเท่านั้น กรณีเป็นมอเตอร์ 2 เฟสจะมีสายที่ใช้ต่อเข้ากับวงจร 5 หรือ 6 สาย



เอกสารนี้เป็นเอกสารรูปที่ 2.17 สัญลักษณ์โครงสร้างและวงจรขับที่ใช้กับมอเตอร์แบบยูนิโพล่า 2 เฟส โยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.3 วิธีการขับ (Driving) หรือวิธีการกระตุ้นเฟส (Phase Excitation) ของสเต็ปเปอร์มอเตอร์

การกระตุ้นเฟสของสเต็ปเปอร์มอเตอร์ คือ การจ่ายกระแสไฟฟ้าไปยังขดลวดที่สเตเตอร์ของแต่ละเฟสเพื่อทำให้มอเตอร์หมุนนั่นเอง แบ่งออกเป็น 3 วิธีคือ

2.9.3.1 การกระตุ้นเฟสแบบ One Phase Excitation หรือ Wave หรือ Half Drive การกระตุ้นเฟสแบบนี้ทำได้โดยการจ่ายกระแสไฟฟ้าไปยังขดลวดครั้งละหนึ่งขดบนสเตเตอร์ดังนี้

ตารางที่ 2.6 แสดงการกระตุ้นเฟสแบบ One Phase

Step	Unipolar		Bipolar	
	Supply		Supply	
	+V	Gnd.	+V	Gnd.
1	Com.	A	A	/A
2	Com.	B	B	/B
3	Com.	/A	/A	A
4	Com.	/B	/B	B

2.9.3.2 การกระตุ้นเฟสแบบ Two Phase Excitation หรือ Full Step การกระตุ้นเฟสแบบนี้ทำได้โดยการจ่ายกระแสไฟฟ้าไปยังขดลวดครั้งละสองขดที่อยู่ใกล้กันบนสเตเตอร์ดังนี้

ตารางที่ 2.7 แสดงการกระตุ้นเฟสแบบ Two Phase

Step	Unipolar		Bipolar	
	Supply		Supply	
	+V	Gnd.	+V	Gnd.
1	Com.	A , B	A , B	/A , /B
2	Com.	/A , B	/A , B	A , /B
3	Com.	/A , /B	/A , /B	A , B
4	Com.	A , /B	A , /B	/A , B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่สามารถเผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.3.3 การกระตุ้นเฟสแบบ One – Two Phase Excitation หรือ Half Step การกระตุ้นเฟสแบบนี้ทำได้โดยการจ่ายกระแสไฟฟ้าไปยังขดลวดครึ่งละสองขดที่อยู่ใกล้กันบนสเตเตอร์สลับกับการจ่ายกระแสไฟฟ้าครึ่งละหนึ่งขดบนสเตเตอร์ ดังนี้

ตารางที่ 2.8 แสดงการกระตุ้นแบบ One – Two Phase Excitation

Step	Unipolar		Bipolar	
	Supply		Supply	
	+V	Gnd.	+V	Gnd.
1	Com.	A , B	A , B	/A , /B
2	Com.	B	B	/B
3	Com.	/A , B	/A , B	A , /B
4	Com.	/A	/A	A
5	Com.	/A , /B	/A , /B	A , B
6	Com.	/B	/B	B
7	Com.	A , /B	A , /B	/A , B
8	Com.	A	A	/A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

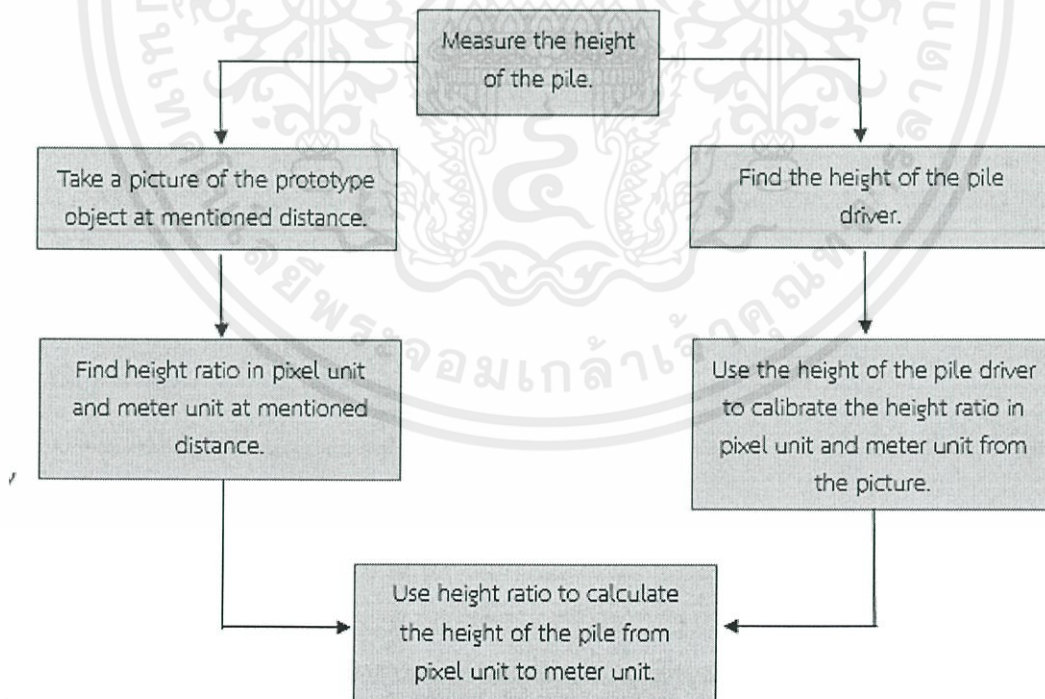
บทที่ 3

การออกแบบและการจัดทำปริยานิพนธ์

3.1 การออกแบบ

3.1.1 ส่วนของโปรแกรมวัดความสูง

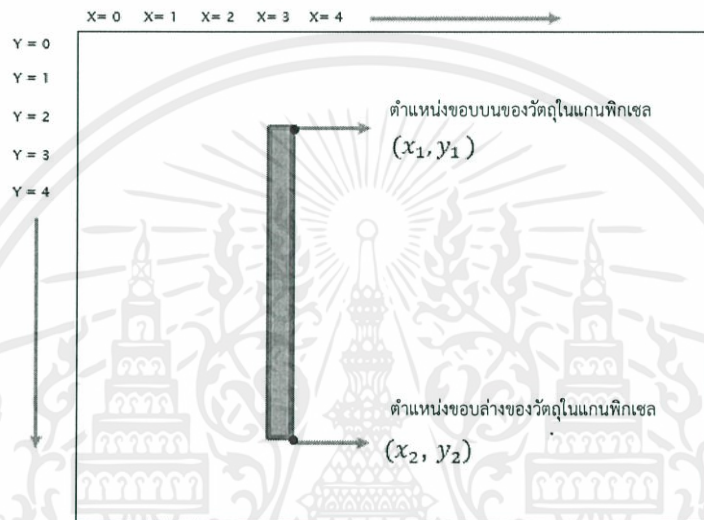
ในส่วนของโปรแกรมวัดความสูงจะใช้หลักการเปรียบเทียบความสูงจริงในหน่วยเมตร เทียบกับจำนวนจุดพิกเซลจากภาพ โดยโปรแกรมจะใช้ระบบการอ้างอิงจากข้อมูล 2 ระบบด้วยกัน ระบบที่หนึ่งคือการใช้ระยะห่างระหว่างกล้องและเสาเข็มเป็นข้อมูลอ้างอิง ในการหาความสูงด้วยระบบนี้ จะทำการหาค่าอัตราส่วนความสูงระหว่างหน่วยเมตรต่อจำนวนจุดพิกเซลจากภาพที่ระยะต่างๆ โดยใช้วัตถุต้นแบบที่ทราบความสูงแน่ชัด จากนั้นจึงนำค่าอัตราส่วนความสูงที่ได้ไปคำนวณหาความสูงของเสาเข็ม ระบบที่สองคือการใช้วัตถุอ้างอิงที่ปรากฏในภาพ เพื่อหาค่าอัตราส่วนความสูงระหว่างหน่วยเมตรต่อจำนวนจุดพิกเซล จากนั้นจึงนำค่าอัตราส่วนความสูงที่ได้ไปคำนวณหาความสูงของเสาเข็มเช่นเดียวกันกับระบบที่หนึ่ง โดยการเขียนโปรแกรมที่ใช้ในการวัดค่าความสูงจะใช้ภาษา C# ในการเขียน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้าม **รูปที่ 3.1** บล็อกไดอะแกรมของโปรแกรมวัดความสูง การทุกครั้งที่มีการนำไปใช้

3.1.2 การหาความสูงจากภาพในหน่วยจำนวนจุดพิกเซล

ในการหาความสูงจากภาพในหน่วยจำนวนจุดพิกเซลจะใช้ตำแหน่งขอบบนและขอบล่างของภาพ ซึ่งแสดงในรูปแบบของตำแหน่งจุดพิกัดของจุดพิกเซลในภาพตามแนวแกน X และแกน Y ดังแสดงในรูป 3.2



รูปที่ 3.2 แสดงการหาตำแหน่งจุดพิกเซลจากภาพ

เมื่อทราบตำแหน่งของจุดพิกเซลจากภาพซึ่งเป็นตำแหน่งขอบบนและขอบล่างของวัตถุ จากนั้นจะใช้ค่าจุดพิกัดสองจุดในการหาระยะห่างระหว่างจุดพิกัดโดยใช้สูตรการหาขนาดของเวกเตอร์เพื่อเป็นการป้องกันการผิดพลาดที่เกิดขึ้นในกรณีที่ภาพมีความเอียง

$$D = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$$

โดย D ระยะห่างระหว่างจุดสองจุดหรือความสูงของวัตถุในภาพ

X_1, Y_1 พิกัดจุดพิกเซลตำแหน่งขอบบนของวัตถุในภาพ

X_2, Y_2 พิกัดจุดพิกเซลตำแหน่งขอบล่างของวัตถุในภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การหาความสูงของเสาเข็มโดยอ้างอิงจากระยะห่างระหว่างกล้องและเสาเข็ม

ในการหาความสูงของเสาเข็มโดยอ้างอิงจากระยะห่างระหว่างกล้องและเสาเข็มนั้น จะทำการหาค่าอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดพิกเซลของวัตถุต้นแบบที่ระยะห่างระหว่างกล้องและวัตถุที่ระยะต่างๆกันตั้งแต่ 10-55 เมตร ซึ่งเป็นระยะห่างที่เหมาะสมและมีความปลอดภัยในการวางกล้อง โดยจะหาค่าทุกๆระยะการเปลี่ยนแปลง 1 เมตร โดยการนำค่าความสูงจริงในหน่วยเมตรหารด้วยความสูงที่ปรากฏในภาพในหน่วยจำนวนจุดพิกเซล

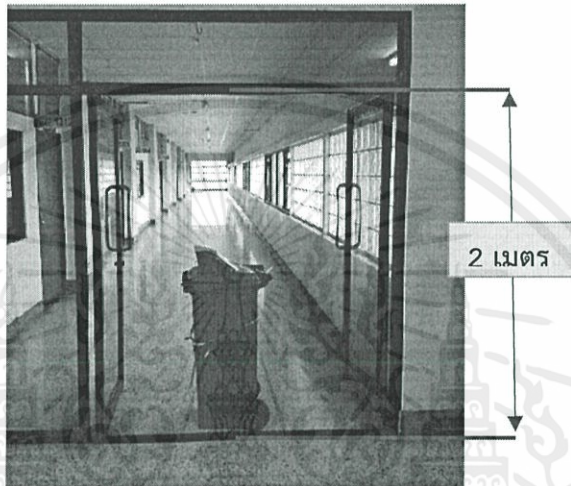
$$\text{อัตราส่วนความสูง (เมตร/พิกเซล)} = \frac{\text{ความสูงจริงของวัตถุต้นแบบ (เมตร)}}{\text{ความสูงที่ปรากฏในภาพของวัตถุต้นแบบ (พิกเซล)}}$$

เมื่อทราบค่าอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดพิกเซลของวัตถุต้นแบบที่ระยะต่างๆแล้ว จึงนำค่าอัตราส่วนความสูงที่ได้ไปคำนวณหาความสูงของเสาเข็ม โดยนำค่าอัตราส่วนความสูงคูณกับความสูงของเสาเข็มในหน่วยจำนวนจุดพิกเซลที่ปรากฏในภาพขณะนั้นๆ ซึ่งผลลัพธ์สุดท้ายที่ได้จะได้ค่าความสูงจริงของเสาเข็มในหน่วยเมตร

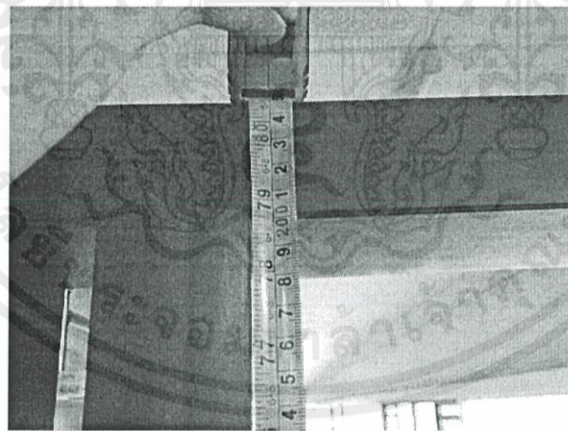
$$\text{ความสูงจริงของเสาเข็ม (เมตร)} = \text{อัตราส่วนความสูง (เมตร/พิกเซล)} \times \text{ความสูงที่ปรากฏในภาพของเสาเข็ม (พิกเซล)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของวัตถุต้นแบบนั้นจะใช้กรอบประตูของอาคาร 12 ชั้น ซึ่งมีความสูง 2 เมตร และสามารถมองเห็นได้จากกล้องอย่างชัดเจนที่ระยะต่างๆตั้งแต่ 10 เมตร ถึง 55 เมตร ในการคำนวณหา ค่าอัตราความสูงในหน่วยเมตรต่อจำนวนจุดพิกเซลที่ระยะตั้งแต่ 10-55 เมตร โดยจะหาค่าทุกๆระยะการ เปลี่ยนแปลง 1 เมตร



รูปที่ 3.3 (ก) กรอบประตูความสูง 2 เมตร



รูปที่ 3.3 (ข) แสดงการวัดความสูงของประตู (2 เมตร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 การหาความสูงของเสาเข็มโดยอ้างอิงจากความสูงของบันจันตอกเสาเข็ม

ในการหาความสูงของเสาเข็มโดยอ้างอิงจากความสูงของบันจันตอกเสาเข็มจะทำการหาค่าอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดพิกเซล โดยใช้ความสูงของบันจันตอกเสาเข็มที่ใช้ในไซต์งานนั้นๆ ซึ่งมีขนาดความสูงมาตรฐานตั้งแต่ 18-24 เมตร โดยใช้หลักการคำนวณเช่นเดียวกันกับการหาความสูงของเสาเข็มโดยอ้างอิงจากระยะห่างระหว่างกล้องและเสาเข็มนั้นคือการนำค่าความสูงจริงในหน่วยเมตรหารด้วยความสูงที่ปรากฏในภาพในหน่วยจำนวนจุดพิกเซล



รูปที่ 3.4 ภาพบันจันความสูง 18 เมตร ขณะทำการตอกเสาเข็ม

$$\text{อัตราส่วนความสูง (เมตร/พิกเซล)} = \frac{\text{ความสูงจริงของบันจัน (เมตร)}}{\text{ความสูงที่ปรากฏในภาพของบันจัน (พิกเซล)}}$$

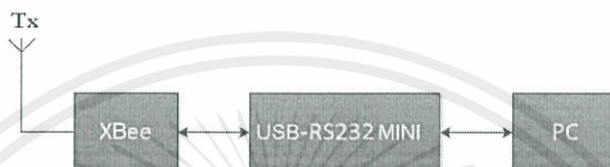
เมื่อทราบค่าอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดพิกเซลของบันจัน จึงนำค่าอัตราส่วนความสูงที่ได้ไปคำนวณหาความสูงของเสาเข็ม โดยนำค่าอัตราส่วนความสูงคูณกับความสูงของเสาเข็มในหน่วยจำนวนจุดพิกเซลที่ปรากฏในภาพขณะนั้นๆ ซึ่งผลลัพธ์สุดท้ายที่ได้จะได้ค่าความสูงจริงของเสาเข็มในหน่วยเมตร

$$\text{ความสูงจริงของเสาเข็ม (เมตร)} = \text{อัตราส่วนความสูง (เมตร/พิกเซล)} \times \text{ความสูงที่ปรากฏในภาพของเสาเข็ม (พิกเซล)}$$

3.1.5 ส่วนของระบบควบคุมกล่องไร้สาย

ในการวัดความสูงด้วยวิธีการวัดระยะระหว่างกล่องและวัตถุจำเป็นต้องปรับมุมกล่อง เพื่อความเที่ยงตรงของตารางวัดความสูง จึงได้ออกแบบระบบควบคุมกล่องไร้สายเพื่ออำนวยความสะดวก

3.1.5.1 บล็อกไดอะแกรมของระบบควบคุมกล่องไร้สาย



รูปที่ 3.5 ระบบควบคุมกล่องไร้สายภาคส่ง

ในการออกแบบระบบควบคุมกล่องไร้สายภาคส่งนั้นจะใช้คอมพิวเตอร์เป็นต้นทางในการส่งตัวอักษร ASCII ผ่าน USB-RS232 MINI เพื่อแปลงพอร์ตจาก USB เป็น RS232 จากนั้นจึงสัญญาณเข้าสู่ XBee ภาคส่งเพื่อส่งสัญญาณต่อไปยัง XBee ภาครับ

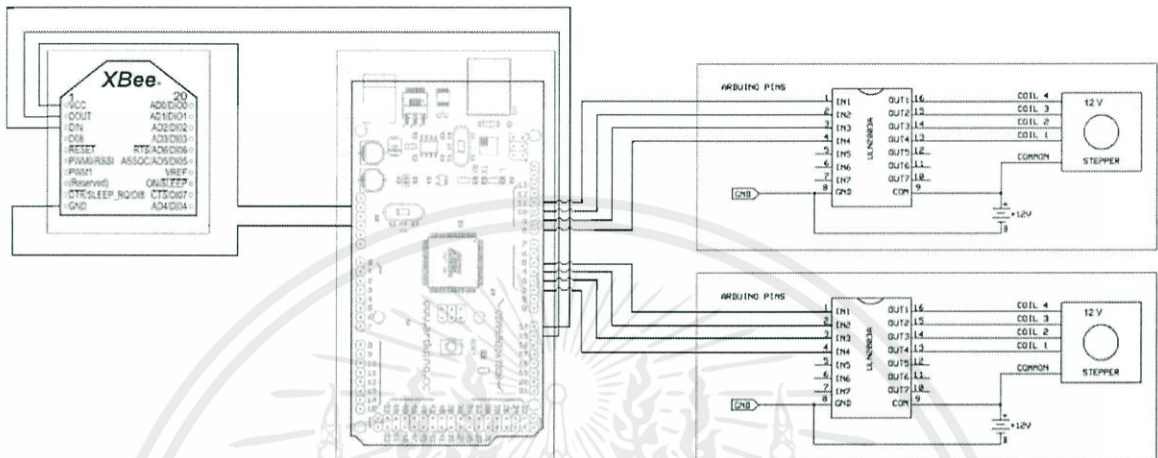


รูปที่ 3.6 ระบบควบคุมกล่องไร้สายภาครับ

ในการออกแบบระบบควบคุมกล่องไร้สายภาครับนั้น XBee ภาครับจะทำการรับข้อมูลรหัส ASCII จาก XBee ภาคส่งจากนั้นจึงส่งสัญญาณเข้าสู่ไมโครคอนโทรลเลอร์ Arduino ผ่านทางพอร์ต Serial เมื่อไมโครคอนโทรลเลอร์ได้รับรหัส ASCII ซึ่งเป็นตัวอักษรที่ตั้งค่าไว้ ไมโครคอนโทรลเลอร์จะทำการส่งสัญญาณเข้าสู่ IC ULN2003 เพื่อขับสเต็ปมอเตอร์ต่อไป ไมโครคอนโทรลเลอร์ที่ใช้คือ Arduino MEGA 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5.2 วงจรรวมของระบบควบคุมกล้องระยะไกล



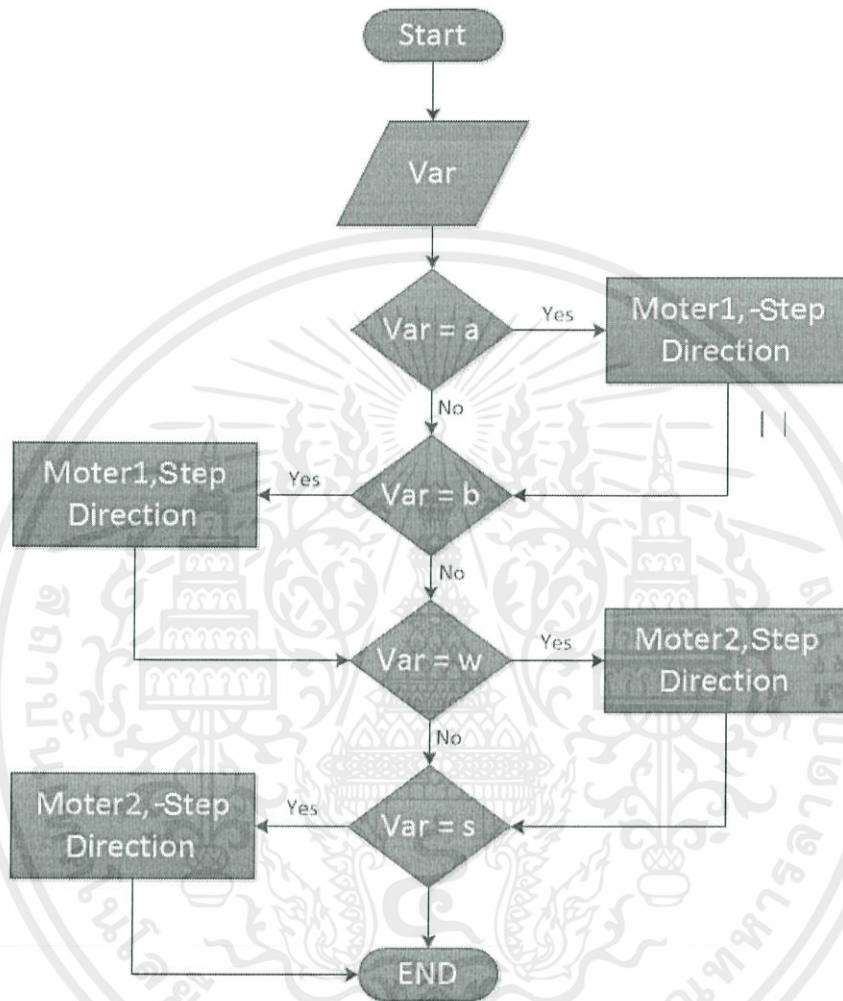
รูปที่ 3.7 วงจรรวมของระบบควบคุมกล้องระยะไกล

จากรูป ในส่วนของ XBee นั้นจะเป็น XBee ปลายทางที่รับข้อมูลเป็นตัวอักษรจาก XBee ต้นทางซึ่งได้ทำการเชื่อมต่ออยู่กับคอมพิวเตอร์ผ่านทาง ET USB/RS232 MINI เพื่อรับข้อมูล โดยในการต่อ XBee ปลายทางนั้นจะทำการต่อเข้ากับพอร์ต Tx3 และ Rx3 และใช้ไฟเลี้ยง 3.3V ของบอร์ด Arduino MEGA 2560 โดยข้อมูลที่ใช้ในการส่งทั้งหมดนั้นจะเป็นการส่งตัวอักษร ASCII

สำหรับวงจรที่ใช้ควบคุม Stepping Motor นั้นใช้พอร์ต 2,3,4,5 สำหรับวงจรที่ 1 และพอร์ต 8, 9, 10, 11 สำหรับวงจรที่ 2 ในการส่งข้อมูล เข้าสู่ IC ULN2003 เพื่อแปลงสัญญาณไฟเข้าสู่ Stepping Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5.3 โฟลวชาร์ตของไมโครคอนโทรลเลอร์

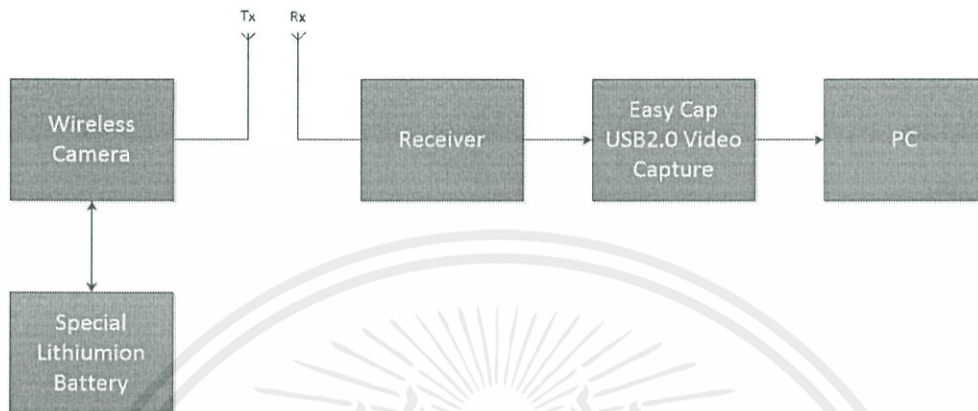


รูปที่ 3.8 โฟลวชาร์ตการทำงานของไมโครคอนโทรลเลอร์

ในการออกแบบการทำงานของไมโครคอนโทรลเลอร์ เมื่อไมโครคอนโทรลเลอร์เริ่มทำงานจะทำการเก็บค่า Var ซึ่งเป็นค่าที่อ่านจาก Serial Port ถ้าค่าที่ได้รับเป็นตัวอักษร 'a' จะทำการสั่งให้สเตปป์มอเตอร์ตัวที่หนึ่งเคลื่อนที่ 1 Step ถ้าได้รับตัวอักษร 'd' จะทำการสั่งให้สเตปป์มอเตอร์ตัวที่หนึ่งเคลื่อนที่ในทิศทางตรงกันข้าม 1 Step ถ้าค่าที่ได้รับเป็นตัวอักษร 's' จะทำการสั่งให้สเตปป์มอเตอร์ตัวที่สองเคลื่อนที่ 1 Step ถ้าได้รับตัวอักษร 'w' จะทำการสั่งให้สเตปป์มอเตอร์ตัวที่สองเคลื่อนที่ในทิศทางตรงกันข้าม 1 Step

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6 ส่วนของระบบรับ-ส่งภาพไร้สาย



รูปที่ 3.9 ระบบรับ-ส่งภาพไร้สาย

ในการออกแบบระบบรับส่งภาพไร้สายจะใช้กล้องไร้สาย(Wireless Camera) ความถี่ 1.2GHz ในการส่งสัญญาณภาพเข้าสู่ตัวรับ (Receiver) จากนั้นจึงนำสัญญาณอนาล็อกที่ได้เข้าสู่อุปกรณ์ EasyCAP USB2.0 Video Capture เพื่อแปลงเป็นสัญญาณดิจิทัลก่อนเข้าสู่คอมพิวเตอร์

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 ส่วนของระบบควบคุมกล้องระยะไกล

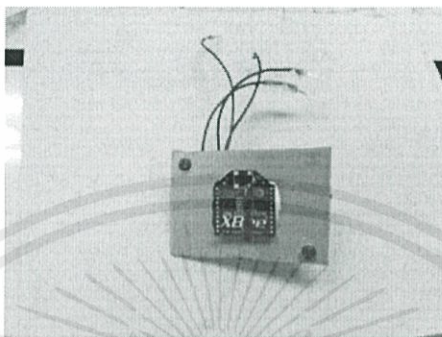
3.2.1.1 อุปกรณ์รับ-ส่งสัญญาณไร้สาย XBee



รูปที่ 3.10 อุปกรณ์ XBee ภาคส่ง (Tx) และ ET USB/RS232 MINI

ในการรับส่งข้อมูลจากคอมพิวเตอร์กับ XBee นั้น จะเป็นการรับส่งข้อมูลผ่านพอร์ต Serial โดยปัจจุบันคอมพิวเตอร์ที่มีพอร์ต Serial นั้นไม่ได้รับความนิยมมากนัก ส่วนใหญ่จะมีเพียงพอร์ต

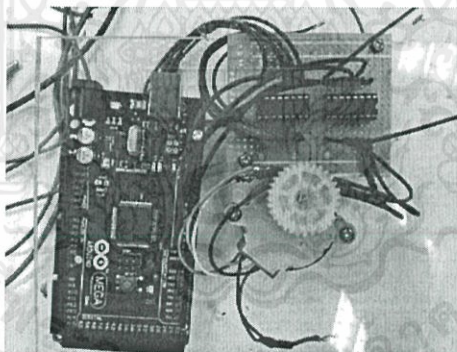
USB เท่านั้น ในการนี้จึงต้องใช้ ET USB/RS232 MINI แปลงพอร์ต USB เป็น Serial ก่อน จึงสามารถเชื่อมต่อคอมพิวเตอร์เข้ากับ XBee ได้



รูปที่ 3.11 อุปกรณ์ XBee ภาครับ (Rx)

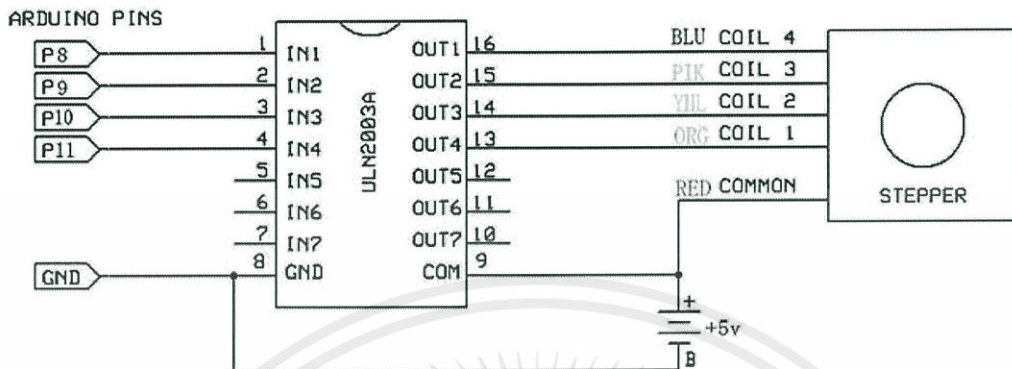
ในอุปกรณ์ XBee ภาคลายทางนั้น ใช้เชื่อมต่อกับไมโครคอนโทรลเลอร์ที่อ่านค่าจากพอร์ต Serial อยู่แล้ว จึงไม่จำเป็นต้องมีการแปลงพอร์ตใดๆทั้งสิ้น

3.2.1.2 ส่วนของวงจรขับ Stepping Motor



รูปที่ 3.12 รูปวงจร Stepping Motor

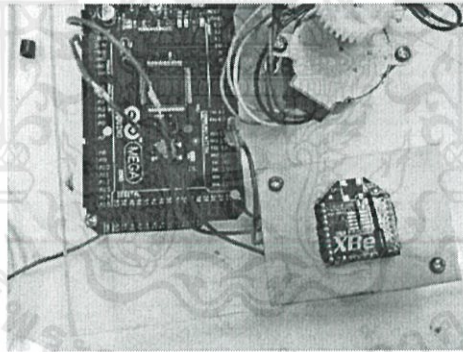
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 วงจร Stepping Motor

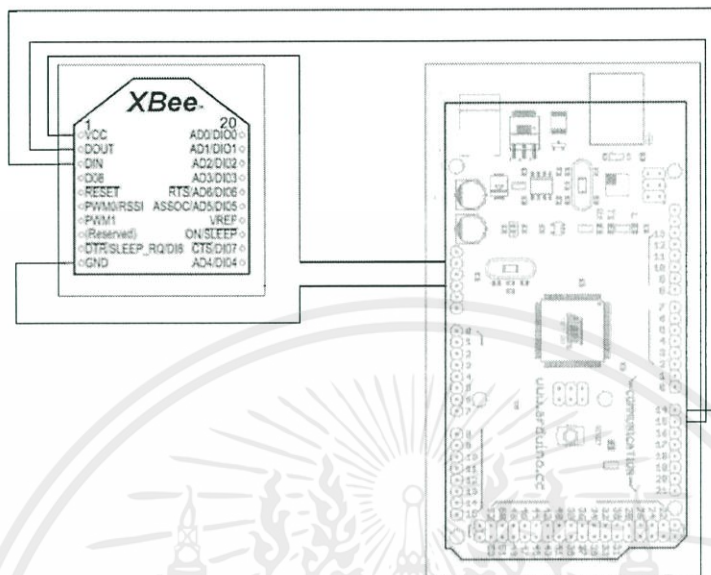
ในการออกแบบวงจรขับสเต็ปมอเตอร์จะใช้ IC เบอร์ ULN2003 ซึ่งเป็น IC เฉพาะที่ใช้ในการขับสเต็ปมอเตอร์ โดยมอเตอร์ที่ใช้มีแรงดันในการทำงาน 12V ในการต่อใช้งานจะต่อขา 16 เข้ากับเฟส 4 ขา 15 เข้ากับเฟส 3 ขา 14 เข้ากับเฟส 2 และขา 13 เข้ากับเฟส 1 ของสเต็ปมอเตอร์

3.2.1.3 ส่วนของการเชื่อมต่อ XBee เข้าสู่ไมโครคอนโทรลเลอร์



รูปที่ 3.14 รูปการเชื่อมต่อ XBee เข้าสู่ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 วงจร XBee เชื่อมต่อกับไมโครคอนโทรลเลอร์

ในการต่อ XBee กับไมโครคอนโทรลเลอร์นั้นใช้การรับส่งข้อมูลผ่านพอร์ต Serial3 (Tx3, Rx3) โดย Arduino MEGA 2560 นั้นมีวงจร Regulator ในตัวทั้ง 5V และ 3.3 V ดังนั้น XBee จึงสามารถใช้ไฟจากไมโครคอนโทรลเลอร์ได้เลย

3.2.2 ส่วนของระบบรับ-ส่งภาพไร้สาย

3.2.2.1 ส่วนของการส่งภาพ (Transmitter)



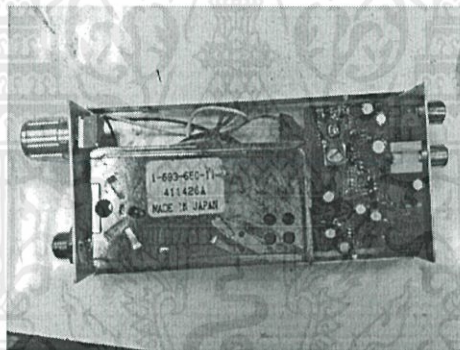
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 3.16 กล้องไร้สาย (Wireless Camera) นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการส่งภาพจะใช้กล้องไร้สาย (Wireless Camera) ซึ่งมีคุณสมบัติดังต่อไปนี้

- กล้องรุ่น 801COLOR CMOS
- ความละเอียด 628x582 พิกเซล
- กำลังขับ 200mW
- ความถี่ 1.2 GHz
- ระยะทางที่สามารถส่งถึงภาครับ 50-100 เมตร
- ใช้แรงดัน 8-12V
- ใช้กระแส 200mA

3.2.2.2 ส่วนของการรับภาพ (Receiver)

ในส่วนของภาครับสัญญาณวิดีโอแบบไร้สายนั้นเราจะใช้กล่องรับสัญญาณ เรียกว่า Radio AV Receiver



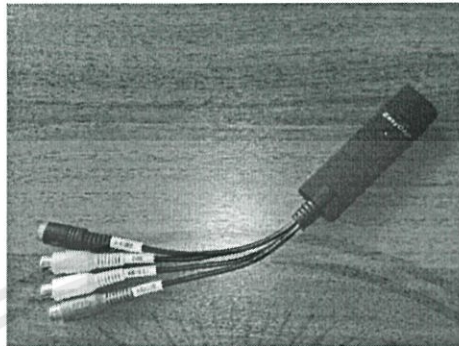
รูปที่ 3.17 รูปแสดงกล่องรับสัญญาณ Radio AV Receiver

ในการรับสัญญาณจากวงจรส่งภาพ ซึ่งในกล่องรับสัญญาณ Radio AV Receiver จะแบ่งการทำงานเป็นหลายภาคส่วนดังนี้

- ภาคส่วนสายอากาศภาครับ (Antenna Receiver) ทำหน้าที่ในการเปลี่ยนพลังงานแม่เหล็กไฟฟ้าเป็นพลังงานไฟฟ้า เพื่อส่งต่อไปยังภาคส่วนวงจรจูนเนอร์ต่อไป
- ภาคส่วนวงจรจูนเนอร์ (Junner) หน้าที่หลักคือการรับสัญญาณจากภาคส่งเปลี่ยนไปเป็นสัญญาณภาพขึ้นที่เครื่องรับโทรทัศน์ หรือที่คอมพิวเตอร์ของผู้ใช้ โดยสรุปหน้าที่การทำงานของตัวจูนเนอร์ได้ดังนี้ เปลี่ยนสัญญาณจากสายอากาศเป็นความถี่ไอเอฟค่าหนึ่ง จากนั้นทำการขยายสัญญาณที่รับมาให้แรงขึ้นอยู่กับความแรงของสัญญาณที่รับเข้ามา และทำการขจัดสัญญาณที่อยู่ภายนอกย่านความถี่สัญญาณโทรทัศน์และสัญญาณรบกวนต่างๆ
- ภาคส่วนขยายกำลัง IF และตรวจจับสัญญาณ

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ อนุญาตให้นำไปใช้โดยไม่ว่าละเมิดลิขสิทธิ์ หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างยิ่งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.3 ส่วนของการรับภาพอนาล็อกเข้าสู่คอมพิวเตอร์



รูปที่ 3.18 EasyCAP USB Video Capture Adapter

ใช้อุปกรณ์ EasyCAP USB Video Capture Adapter ในการรับภาพจากสาย AV เข้าสู่พอร์ต USB เพื่อแปลงเป็นสัญญาณดิจิทัลเพื่อส่งสัญญาณเข้าสู่คอมพิวเตอร์

3.3 การจัดเก็บผลการทดลอง

- 3.3.1 แสดงโปรแกรมวัดความสูงของเสาเข็ม
- 3.3.2 เก็บค่าอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดพิคเซลที่ระยะห่างระหว่างกล้องและวัตถุต้นแบบที่ระยะห่าง 10-55 เมตร โดยวัดทุกระยะการเปลี่ยนแปลง 1 เมตร พร้อมทั้งพล็อตกราฟแสดงการเปลี่ยนแปลง
- 3.3.3 วัดความสูงของเสาเข็มโดยใช้ระบบอ้างอิงจากระยะห่างระหว่างกล้องและเสาเข็มพร้อมทั้งคำนวณหาเปอร์เซ็นต์ความผิดพลาด
- 3.3.4 วัดความสูงของเสาเข็มโดยใช้ระบบอ้างอิงจากความสูงของบันจันพร้อมทั้งคำนวณหาเปอร์เซ็นต์ความผิดพลาด
- 3.3.5 ทดสอบการรับส่งตัวอักษร ASCII ของอุปกรณ์ XBee โดยโปรแกรม X-CTU
- 3.3.6 ทดสอบไมโครคอนโทรลเลอร์ในการเชื่อมต่อกับคอมพิวเตอร์ผ่านพอร์ต Serial
- 3.3.7 ทดสอบไมโครคอนโทรลเลอร์ในการเชื่อมต่อกับคอมพิวเตอร์ผ่าน XBee
- 3.3.8 ทดสอบการทำงานของ IC ULN2003 เพื่อขับ Stepping Motor เมื่อมีการสั่งการทำงานผ่านทางไมโครคอนโทรลเลอร์
- 3.3.9 ทดสอบการรับส่งภาพไร้สายที่ระยะต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

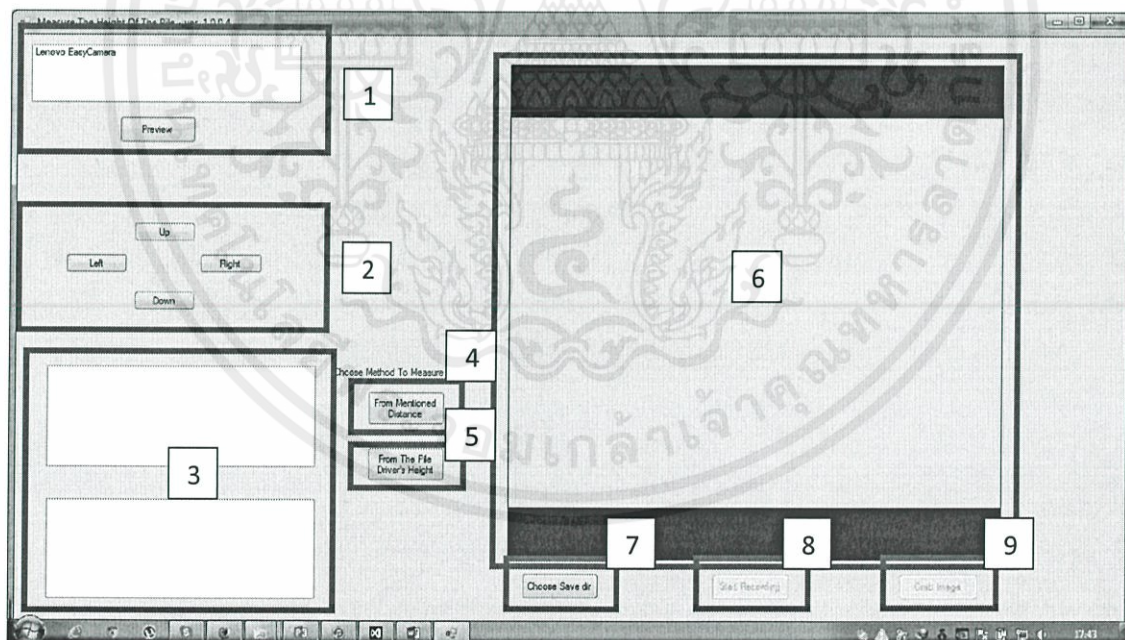
บทที่ 4

ผลการทดลอง

4.1 โปรแกรมวัดความสูง

ในส่วนของโปรแกรมวัดความสูงจะประกอบไปด้วยหน้าต่าง 3 หน้าต่างด้วยกัน หน้าทีหนึ่งประกอบด้วยส่วนของการรับภาพจากกล้อง การส่งสัญญาณควบคุมการหมุนของกล้องผ่าน Serial Port และ การบันทึกภาพวิดีโอ หน้าต่างที่สองจะเป็นส่วนของการวัดความสูงด้วยระบบอ้างอิงจากระยะกล้องและเสาเข็ม รวมไปถึงตารางแสดงค่าอัตราส่วนความสูงระหว่างเมตรต่อจำนวนจุดพิกเซล หน้าต่างที่สามจะเป็นส่วนของการวัดความสูงด้วยระบบอ้างอิงจากความสูงของบันจัน ในส่วนของหน้าต่างวัดความสูงทั้งสองระบบจะใช้การคลิกของพ้อยเตอร์ที่ตำแหน่งขอบบนและขอบล่างของวัตถุที่ต้องการวัดความสูงเพื่อเก็บค่าตำแหน่งพิกัดในแกนพิกเซลเพื่อ Paste เข้าสู่ช่องเก็บค่าเพื่อทำการคำนวณ

4.1.1 โปรแกรมวัดความสูงหน้าต่างที่หนึ่ง



รูปที่ 4.1 โปรแกรมวัดความสูงหน้าต่างที่หนึ่ง

หมายเลข 1 แสดงอุปกรณ์ถ่ายภาพที่อยู่ในสถานะพร้อมใช้งาน

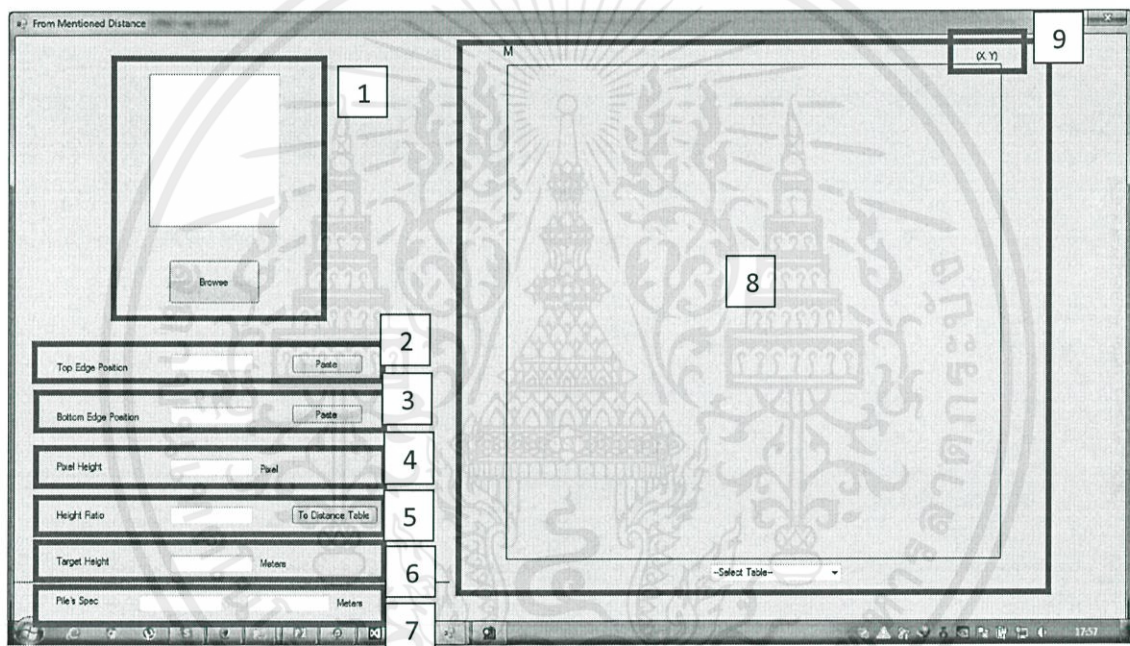
หมายเลข 2 ปุ่มควบคุมกล้องผ่าน Serial Port

หมายเลข 3 แสดงไฟล์ภาพและวิดีโอที่เซฟอยู่ในคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดก็ตาม หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อผู้จัดทำเอกสารทุกครั้งที่มีการนำไปใช้

- หมายเลข 4 ปุ่มเปิดหน้าต่างการวัดความสูงโดยอ้างอิงจากระยะห่างระหว่างกล้องและเสาเข็ม
 หมายเลข 5 ปุ่มเปิดหน้าต่างการวัดความสูงโดยอ้างอิงจากความสูงของป็นจัน
 หมายเลข 6 กรอบแสดงภาพจากกล้องแบบ Real Time
 หมายเลข 7 เลือก Folder ที่ทำการเซฟไฟล์
 หมายเลข 8 ปุ่มเริ่มต้นที่วิดีโอ
 หมายเลข 9 ปุ่มเซฟภาพที่แสดง ณ เวลานั้นๆ

4.1.2 โปรแกรมวัดความสูงหน้าต่างที่สอง

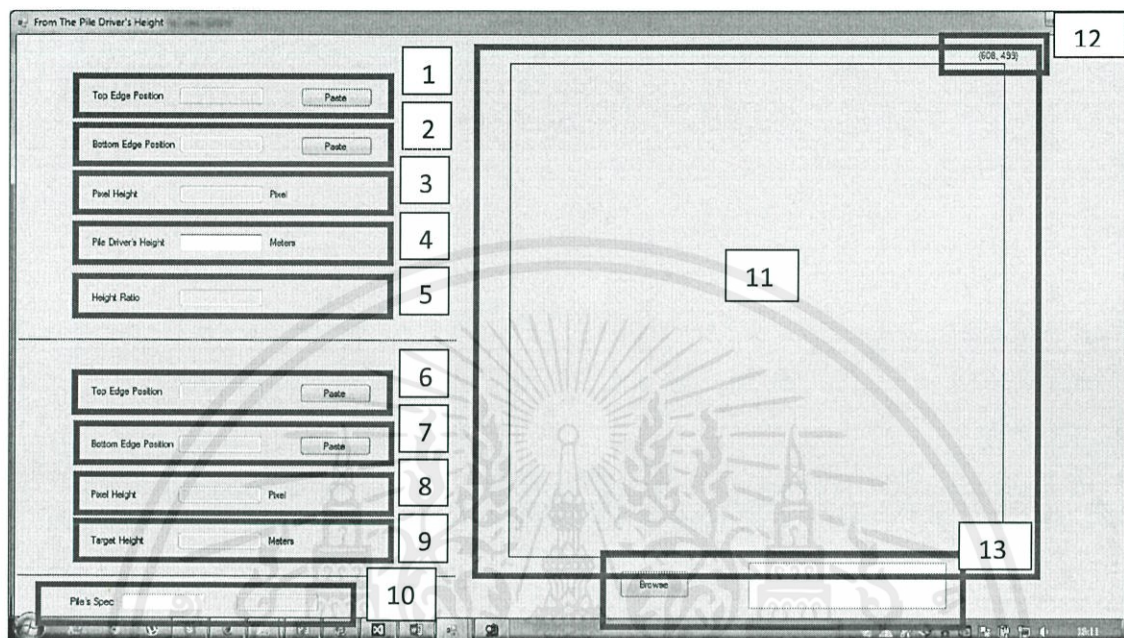


รูปที่ 4.2 โปรแกรมวัดความสูงหน้าต่างที่สอง

- หมายเลข 1 เลือก Folder ที่เก็บไฟล์พร้อมทั้งแสดงไฟล์ที่เซฟในเครื่องคอมพิวเตอร์
 หมายเลข 2 ช่องเก็บค่าพิกัดขอบบนของเสาเข็ม
 หมายเลข 3 ช่องเก็บค่าพิกัดขอบล่างของเสาเข็ม
 หมายเลข 4 ช่องแสดงผลการคำนวณความสูงจากภาพในหน่วยจำนวนจุดพิกเซล
 หมายเลข 5 ช่องใส่ค่าอัตราส่วนความสูงจากตารางและปุ่มเปิดตารางแสดงค่าอัตราส่วนความสูง
 หมายเลข 6 ช่องแสดงผลการคำนวณความสูงจริงในหน่วยเมตร
 หมายเลข 7 ช่องแสดงค่าความลึกของเสาเข็มที่ตอกลงดินในหน่วยเมตร
 หมายเลข 8 ช่องแสดงภาพ
 หมายเลข 9 แสดงตำแหน่งจุดพิกเซลที่พ้อยเตอร์ทำการชี้ ณ เวลานั้น

เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุผลเชิงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 โปรแกรมวัดความสูงหน้าต่างที่สาม



รูปที่ 4.3 โปรแกรมวัดความสูงหน้าต่างที่สาม

หมายเลข 1 ช่องเก็บค่าพิกัดขอบบนของบันจัน

หมายเลข 2 ช่องเก็บค่าพิกัดขอบล่างของบันจัน

หมายเลข 3 ช่องแสดงผลการคำนวณความสูงจากภาพในหน่วยจำนวนจุดพิกเซล

หมายเลข 4 ช่องเก็บค่าความสูงจริงของบันจันที่ใช้ในการตอกในหน่วยเมตร

หมายเลข 5 ช่องแสดงผลการคำนวณอัตราความสูงในหน่วยเมตรต่อจำนวนจุดพิกเซล

หมายเลข 6 ช่องเก็บค่าพิกัดขอบบนของเสาเข็ม

หมายเลข 7 ช่องเก็บค่าพิกัดขอบล่างของเสาเข็ม

หมายเลข 8 ช่องแสดงความสูงของเสาเข็มในหน่วยพิกเซล

หมายเลข 9 ช่องแสดงความสูงจริงของเสาเข็มในหน่วยเมตร

หมายเลข 10 ช่องแสดงค่าความลึกของเสาเข็มที่ตอกลงดินในหน่วยเมตร

หมายเลข 11 ช่องแสดงภาพ

หมายเลข 12 แสดงตำแหน่งจุดพิกเซลที่พ้อยเตอร์ทำการชี้ ณ เวลานั้น

หมายเลข 13 เลือก Folder ที่เก็บไฟล์พร้อมทั้งแสดงไฟล์ที่เซฟในเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ค่าอัตราส่วนความสูงระหว่างเมตรต่อจำนวนจุดพิกเซลที่ระยะต่างๆ

ในการเก็บค่าอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดพิกเซลที่ระยะห่างระหว่างกล้องและวัตถุต้นแบบจะเก็บค่าที่ระยะห่าง 10-55 เมตร ซึ่งเป็นระยะที่มีความปลอดภัยในการเก็บภาพ โดยวัดทุกระยะการเปลี่ยนแปลง 1 เมตร โดยจะทำการวัดโดยกล้องสองชนิด คือ กล้องไร้สายที่ใช้ในการทดลอง และกล้องจากโทรศัพท์มือถือ จะได้ค่าดังนี้

ตารางที่ 4.1 ค่าอัตราส่วนความสูงโดยกล้องไร้สายที่ใช้ในการทดลอง

ระยะห่างระหว่างกล้องและวัตถุต้นแบบ (เมตร)	จำนวนจุดพิกเซลต่อความสูง 1 เมตร	อัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดพิกเซล
10	73	0.0136
11	66.5	0.0150
12	62.5	0.0160
13	56.5	0.0176
14	52	0.0192
15	50	0.0200
16	47	0.0212
17	43.5	0.0229
18	41.5	0.0240
19	38.5	0.0259
20	37	0.0270
21	35.5	0.0281
22	34.5	0.0289
23	33	0.030
24	31.5	0.0317
25	30	0.0333
26	28.5	0.0350
27	27.5	0.0363
28	27	0.0370
29	26.5	0.0377
30	26	0.0384

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 (ต่อ) ค่าอัตราส่วนความสูงโดยกล้องไร้สายที่ใช้ในการทดลอง

ระยะห่างระหว่าง กล้องและวัตถุ ต้นแบบ (เมตร)	จำนวนจุดพิกเซลต่อความ สูง 1 เมตร	อัตราส่วนความสูงในหน่วยเมตร ต่อจำนวนจุดพิกเซล
31	25.5	0.0392
32	24.5	0.0408
33	24	0.0416
34	23.5	0.0425
35	23	0.0434
36	22	0.0454
37	21.5	0.0465
38	21	0.0476
39	19.5	0.0512
40	19	0.0526
41	18.5	0.0540
42	18	0.0555
43	18	0.0555
44	17.5	0.0571
45	17.5	0.0571
46	17	0.0588
47	17	0.0588
48	16.5	0.0606
49	16	0.0625
50	15.5	0.0645
51	15	0.0666
52	15	0.0666
53	15	0.0666
54	14.5	0.0689
55	14	0.0714

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ค่าอัตราส่วนความสูงโดยก่อกองจากโทรศัพท์มือถือ

ระยะห่างระหว่าง ก่อกองและวัตถุ ต้นแบบ (เมตร)	จำนวนจุดพิกเซลต่อความ สูง 1 เมตร	อัตราส่วนความสูงในหน่วยเมตร ต่อจำนวนจุดพิกเซล
10	59	0.0169
11	52	0.0192
12	47	0.0213
13	45.5	0.022
14	41.5	0.0241
15	38	0.0263
16	33.5	0.0299
17	33	0.0303
18	32	0.0313
19	30	0.0333
20	29	0.0345
21	27.5	0.0364
22	26	0.0385
23	25	0.04
24	24.5	0.0408
25	24	0.0417
26	22.5	0.0444
27	21.5	0.0465
28	20	0.05
29	19	0.0526
30	18	0.0556

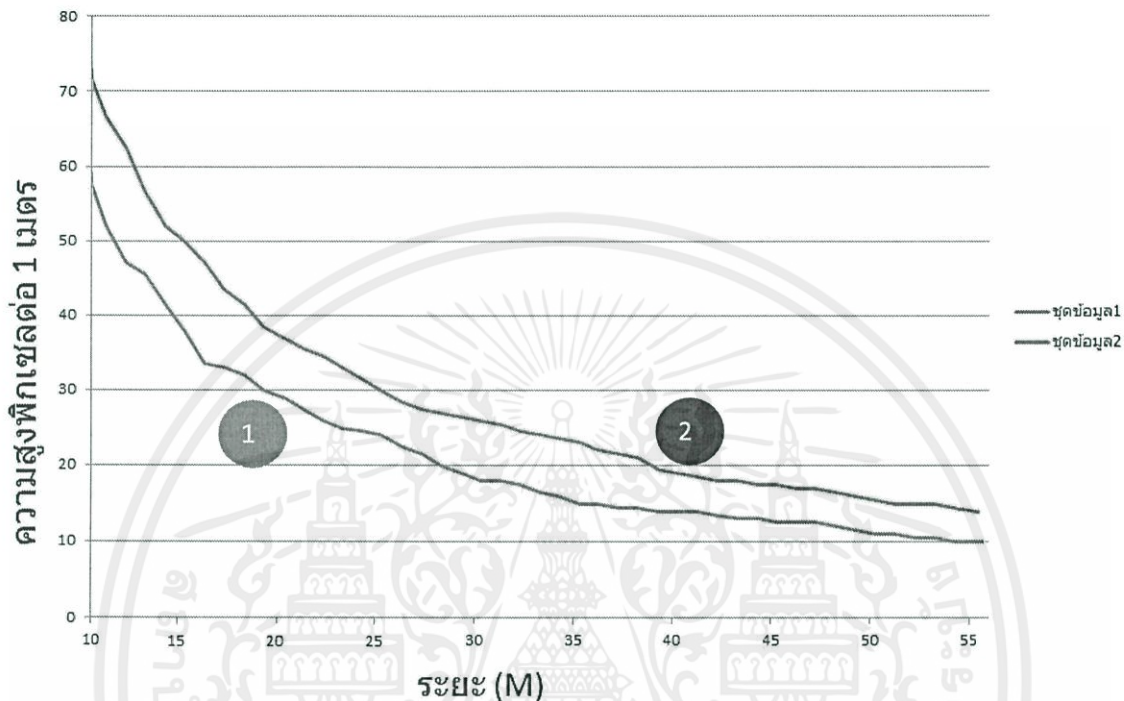
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 (ต่อ) ค่าอัตราส่วนความสูงโดยกล้องจากโทรศัพท์มือถือ

ระยะห่างระหว่าง กล้องและวัตถุ ต้นแบบ (เมตร)	จำนวนจุดพิกเซลต่อความ สูง 1 เมตร	อัตราส่วนความสูงในหน่วยเมตร ต่อจำนวนจุดพิกเซล
31	18	0.0556
32	17.5	0.0571
33	16.5	0.0606
34	16	0.0625
35	15	0.0667
36	15	0.0667
37	14.5	0.069
38	14.5	0.069
39	14	0.0714
40	14	0.0714
41	14	0.0714
42	13.5	0.0741
43	13	0.0769
44	13	0.0769
45	12.5	0.08
46	12.5	0.08
47	12.5	0.08
48	12	0.0833
49	11.5	0.087
50	11	0.0909
51	11	0.0909
52	10.5	0.0952
53	10.5	0.0952
54	10	0.1
55	10	0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำค่าจากตารางมาพล็อตกราฟจะได้กราฟดังรูป



รูปที่ 4.4 กราฟแสดงแนวโน้มของอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดฟีกเซลที่ระยะต่างๆ

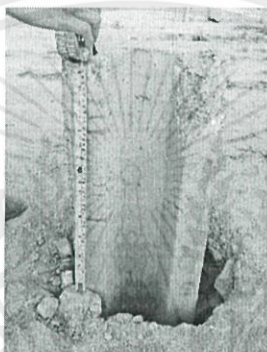
จากกราฟแสดงแนวโน้มของอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดฟีกเซลที่ระยะต่างๆนั้น คือ เมื่อทำการถ่ายภาพที่ระยะใกล้ค่าการเปลี่ยนแปลงความสูงในหน่วยจำนวนจุดฟีกเซลจะมีการเปลี่ยนแปลงมากเมื่อทำการเปลี่ยนตำแหน่งกล้องที่ทุกระยะ 1 เมตร (ความชันของกราฟมีค่ามาก) ในขณะเดียวกัน เมื่อทำการถ่ายภาพที่ระยะไกล การเปลี่ยนแปลงความสูงในหน่วยจำนวนจุดฟีกเซลจะมีการเปลี่ยนแปลงน้อยกว่าเมื่อทำการเปลี่ยนตำแหน่งกล้องที่ทุกระยะ 1 เมตร (ความชันของกราฟมีค่าน้อย)

ในกราฟแสดงการถ่ายภาพจากกล้อง 2 ชนิดนั้นคือกล้องไร้สายที่ใช้ในการทดลองและกล้องจากโทรศัพท์มือถือ แสดงให้เห็นความแตกต่างของภาพที่ถ่ายจากกล้องชนิดจะไม่สามารถใช้ค่าอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดฟีกเซลเดียวกันได้ เนื่องจากการขยายภาพของเลนส์กล้องแต่ละชนิดมีค่าไม่เท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การวัดความสูงของเสาเข็มโดยอ้างอิงจากระยะห่างระหว่างกล่องและเสาเข็ม

ในการวัดความสูงของเสาเข็มจะทำการวัดเสาเข็มเสาเข็มแบบต่อเชื่อมสองต้นคือเสาเข็มความสูง 11 เมตร และ 12 เมตร โดยในการวัดจะทำการวัดเสาเข็มต้นที่สอง(12 เมตร) ขณะทำการเชื่อมกับเสาต้นที่หนึ่ง(11 เมตร) ที่ตกลงไปในพื้นดินและโผล่จากพื้นดิน 50 เซนติเมตร ในการวัดจะวัดที่ระยะ 30, 35, 45 และ 55 เมตร



รูปที่ 4.5 แสดงการวัดเสาต้นที่ 1 ส่วนที่โผล่จากพื้นดิน (50 เซนติเมตร)

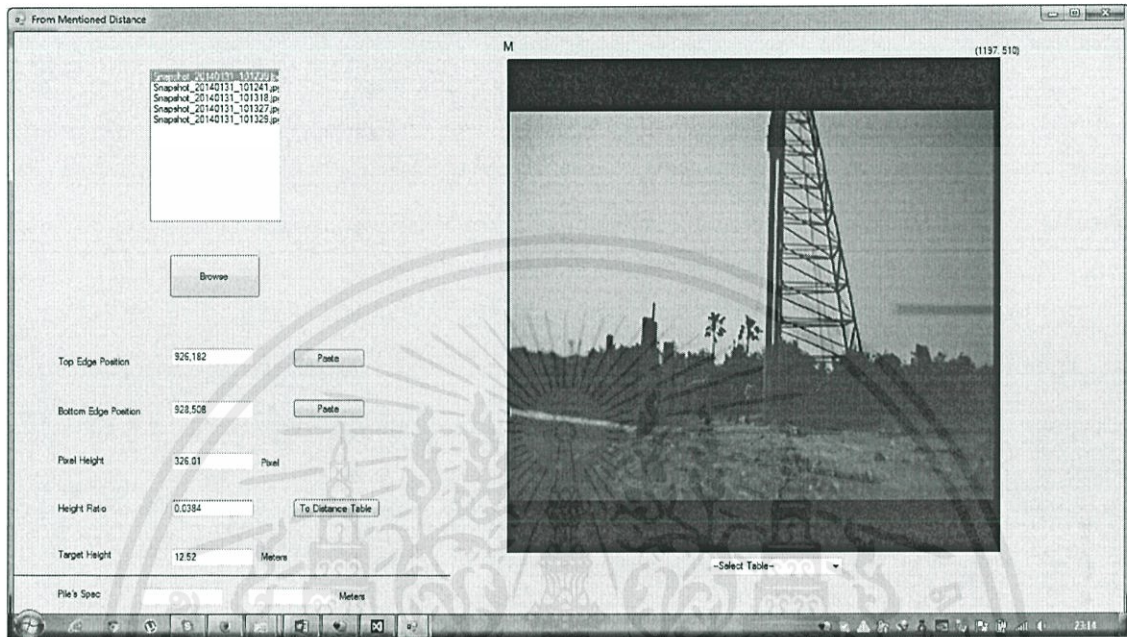


รูปที่ 4.6 การเชื่อมเสาต้นที่สองและเสาต้นที่หนึ่ง

จากรูปที่ 4.5 จะเห็นว่าการวัดความสูงของเสาต้นที่สองจะต้องทำการบวกความสูงของเสาต้นแรกที่โผล่จากพื้นดิน 50 เซนติเมตรด้วย เพื่อความถูกต้องของการวัดค่าความสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1 วัดความสูงของเสาเข็มที่ระยะ 30 เมตร

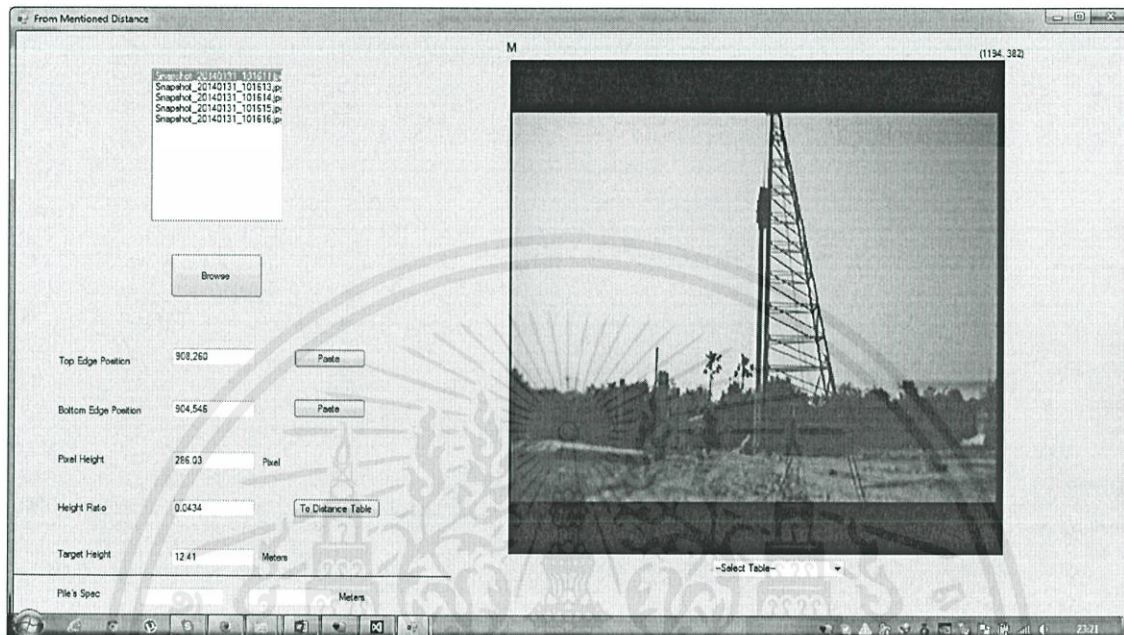


รูปที่ 4.7 ความสูงของเสาเข็มที่ระยะ 30 เมตร

ทำการเลือกไฟล์ภาพถ่ายของเสาเข็มที่ระยะ 30 เมตร จากนั้นคลิกเก็บค่าขอบบนของเสาเข็ม (926,182) และขอบล่างของเสาเข็ม (928,508) โปรแกรมคำนวณความสูงในหน่วยจำนวนจุดพิกเซลได้เท่ากับ 326.01 จุดพิกเซล จากตารางจะได้ค่าอัตราส่วนความสูงระหว่างเมตรต่อจำนวนจุดพิกเซลที่ระยะ 30 เมตร เท่ากับ 0.0384 จากความสูงจริงของเสาเข็ม 12.5 เมตร ความสูงจริงที่โปรแกรมคำนวณได้เท่ากับ 12.52 เมตร ค่าความผิดพลาดเท่ากับ 0.16 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 วัดความสูงของเสาเข็มที่ระยะ 35 เมตร

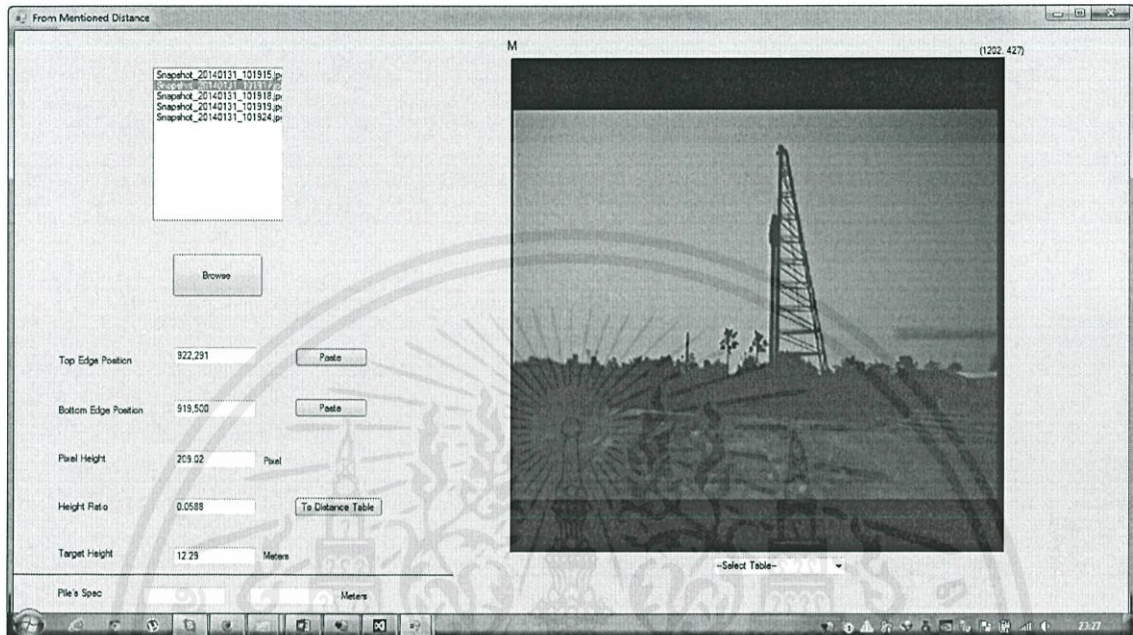


รูปที่ 4.8 ความสูงของเสาเข็มที่ระยะ 35 เมตร

ทำการเลือกไฟล์ภาพถ่ายของเสาเข็มที่ระยะ 35 เมตร จากนั้นคลิกเก็บค่าขอบบนของเสาเข็ม (908,260) และขอบล่างของเสาเข็ม (904,546) โปรแกรมคำนวณความสูงในหน่วยจำนวนจุดพิกเซลได้เท่ากับ 286.03 จุดพิกเซล จากตารางจะได้ค่าอัตราส่วนความสูงระหว่างเมตรต่อจำนวนจุดพิกเซลที่ระยะ 35 เมตร เท่ากับ 0.0434 จากความสูงจริงของเสาเข็ม 12.5 เมตร ความสูงจริงที่โปรแกรมคำนวณได้เท่ากับ 12.41 เมตร ค่าความผิดพลาดเท่ากับ 0.72 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 วัดความสูงของเสาเข็มที่ระยะ 45 เมตร

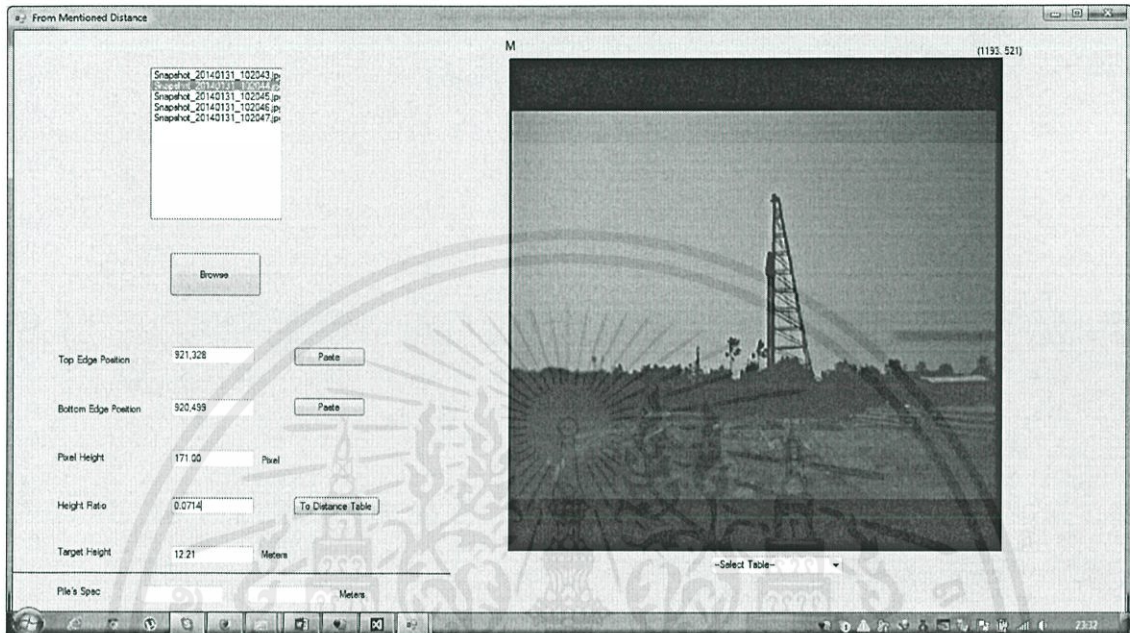


รูปที่ 4.9 ความสูงของเสาเข็มที่ระยะ 45 เมตร

ทำการเลือกไฟล์ภาพถ่ายของเสาเข็มที่ระยะ 45 เมตร จากนั้นคลิกเก็บค่าขอบบนของเสาเข็ม (922,291) และขอบล่างของเสาเข็ม (919,500) โปรแกรมคำนวณความสูงในหน่วยจำนวนจุดพิกเซลได้เท่ากับ 209.02 จุดพิกเซล จากตารางจะได้ค่าอัตราส่วนความสูงระหว่างเมตรต่อจำนวนจุดพิกเซลที่ระยะ 45 เมตร เท่ากับ 0.0588 จากความสูงจริงของเสาเข็ม 12.5 เมตร ความสูงจริงที่โปรแกรมคำนวณได้เท่ากับ 12.29 เมตร ค่าความผิดพลาดเท่ากับ 1.68 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 วัดความสูงของเสาเข็มที่ระยะ 55 เมตร



รูปที่ 4.10 ความสูงของเสาเข็มที่ระยะ 55 เมตร

ทำการเลือกไฟล์ภาพถ่ายของเสาเข็มที่ระยะ 55 เมตร จากนั้นคลิกเก็บค่าขอบบนของเสาเข็ม (921,328) และขอบล่างของเสาเข็ม (929,499) โปรแกรมคำนวณความสูงในหน่วยจำนวนจุดพิกเซลได้เท่ากับ 171 จุดพิกเซล จากตารางจะได้ค่าอัตราส่วนความสูงระหว่างเมตรต่อจำนวนจุดพิกเซลที่ระยะ 55 เมตร เท่ากับ 0.0714 จากความสูงจริงของเสาเข็ม 12.5 เมตร ความสูงจริงที่โปรแกรมคำนวณได้เท่ากับ 12.21 เมตร ค่าความผิดพลาดเท่ากับ 2.32 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

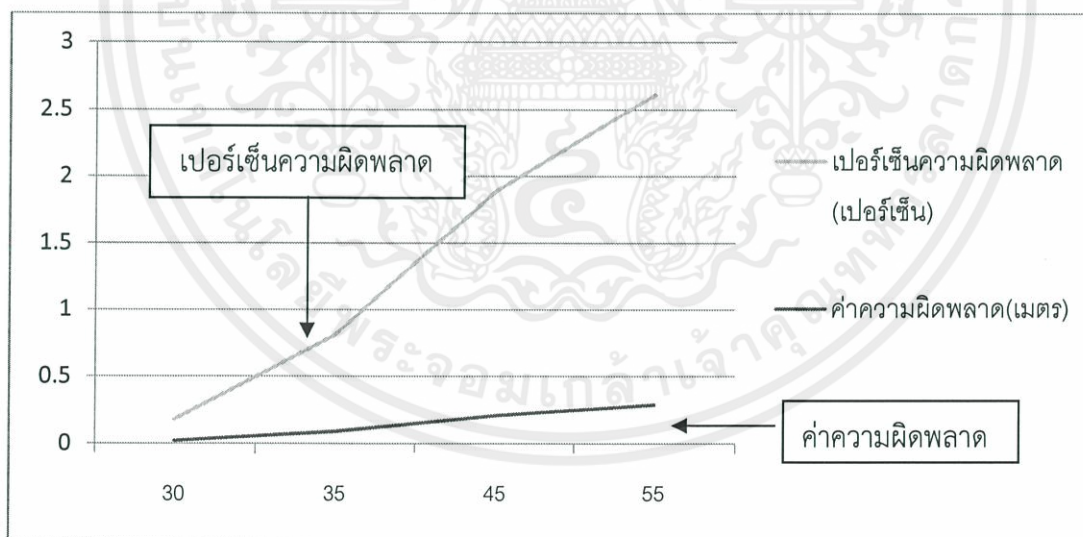
4.3.5 สรุปผลการวัดความสูงของเสาเข็มโดยอ้างอิงจากระยะห่างระหว่างกล้องและเสาเข็ม

จากการทดสอบการวัดความสูงของเสาเข็มโดยอ้างอิงจากระยะห่างระหว่างกล้องและเสาเข็มนั้นจะได้ค่าความสูงจริงและความสูงที่วัดได้ พร้อมทั้งเปอร์เซ็นต์ความผิดพลาดดังนี้

ตารางที่ 4.3 สรุปผลการวัดความสูงของเสาเข็มโดยอ้างอิงจากระยะห่างระหว่างกล้องและเสาเข็ม

ระยะห่าง (เมตร)	ความสูงจริง (เมตร)	ความสูงที่วัดได้ (เมตร)	ค่าความผิดพลาด (เมตร)	เปอร์เซ็นต์ความผิดพลาด (เปอร์เซ็นต์)
30	12.50	12.52	0.02	0.16
35	12.50	12.41	0.09	0.72
45	12.50	12.29	0.21	1.68
55	12.50	12.21	0.29	2.32

เมื่อนำค่าจากตารางมาพล็อตเป็นกราฟจะได้กราฟดังรูป



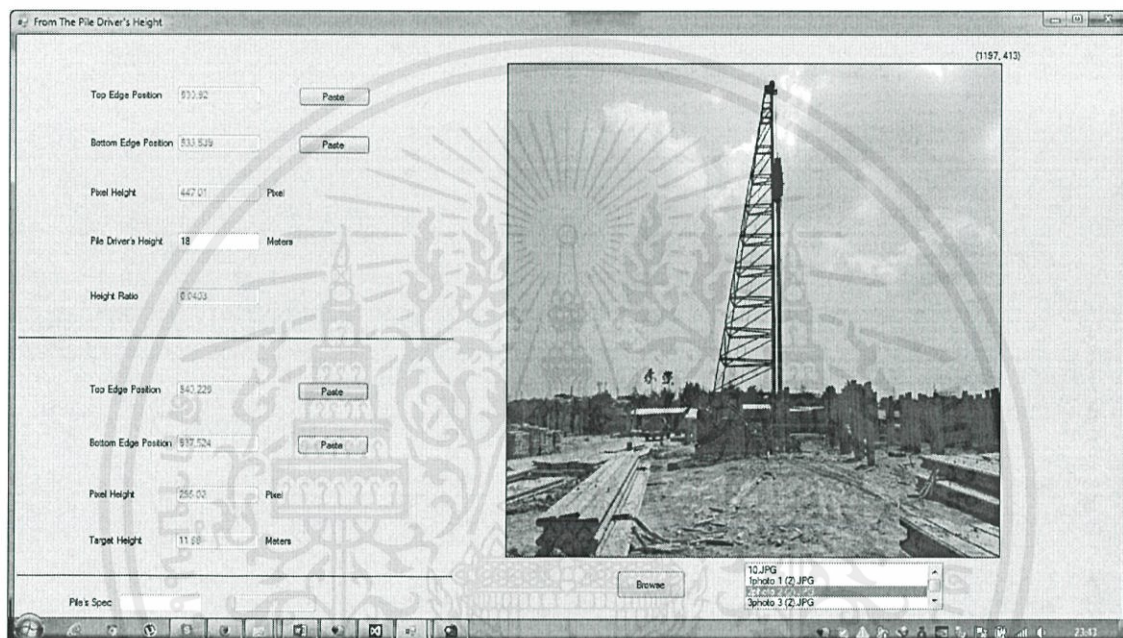
รูปที่ 4.11 กราฟแสดงค่าความผิดพลาดและเปอร์เซ็นต์ความผิดพลาด

ค่าความผิดพลาดที่เกิดขึ้นมีค่า 0.16 ถึง 2.32 เปอร์เซ็นต์ โดยมีค่าเฉลี่ยความผิดพลาดเท่ากับ 1.22 เปอร์เซ็นต์ จากกราฟแสดงให้เห็นค่าความผิดพลาดที่ระยะใกล้จะมีค่าน้อยกว่าระยะไกล เนื่องจากระยะใกล้จะมีความละเอียดของภาพมากกว่า ทำให้การวัดความสูงจากภาพมีความแม่นยำกว่า

4.4 การวัดความสูงของเสาเข็มโดยอ้างอิงจากความสูงของปั้นจั่น

ในการวัดความสูงของปั้นจั่นโดยอ้างอิงจากความสูงของปั้นจั่น จะทำการสุ่มวัดที่ระยะต่างๆ โดยปั้นจั่นที่ใช้ในการตอกมีความสูง 18 เมตร

4.4.1 วัดความสูงโดยอ้างอิงจากความสูงของปั้นจั่นครั้งที่หนึ่ง

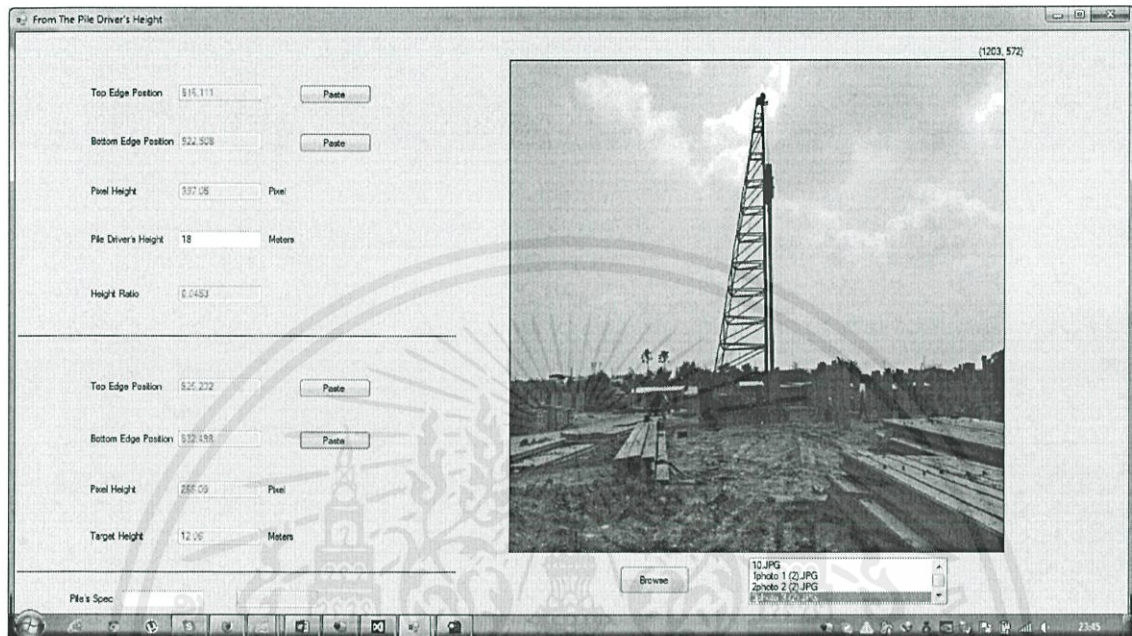


รูปที่ 4.12 วัดความสูงโดยอ้างอิงจากความสูงของปั้นจั่นครั้งที่หนึ่ง

เมื่อทำการคลิกเก็บค่าขอบบนของปั้นจั่น (930,92) และขอบล่างของปั้นจั่น (933,539) โปรแกรมคำนวณได้ความสูงในหน่วยจำนวนจุดพิกเซลเท่ากับ 447.01 จุดพิกเซล และอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดพิกเซลเท่ากับ 0.0403 จากนั้นจึงคลิกเก็บค่าขอบบนของเสาเข็ม (940,229) และขอบล่างของเสาเข็ม (937,524) โปรแกรมคำนวณความสูงในหน่วยพิกเซลได้เท่ากับ 295.02 จุดพิกเซล ความสูงจริงของเสาเข็ม 12 เมตร ความสูงจริงที่โปรแกรมคำนวณได้เท่ากับ 11.88 เมตร ค่าความผิดพลาดเท่ากับ 1.0 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 วัดความสูงโดยอ้างอิงจากความสูงของปั้นจั่นครั้งที่สอง

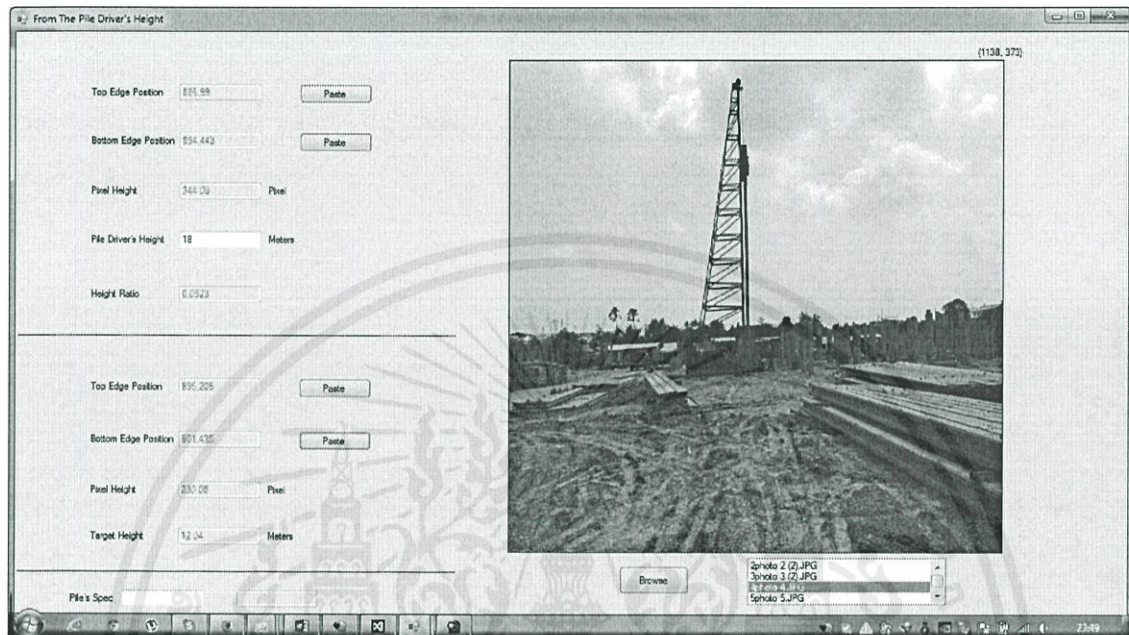


รูปที่ 4.13 วัดความสูงโดยอ้างอิงจากความสูงของปั้นจั่นครั้งที่สอง

เมื่อทำการคลิกเก็บค่าขอบบนของปั้นจั่น (916,111) และขอบล่างของปั้นจั่น (922,508) โปรแกรมคำนวณได้ความสูงในหน่วยจำนวนจุดพิกเซลเท่ากับ 397.05 จุดพิกเซล และอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดพิกเซลเท่ากับ 0.0453 จากนั้นจึงคลิกเก็บค่าขอบบนของเสาเข็ม (925,232) และขอบล่างของเสาเข็ม (932,498) โปรแกรมคำนวณความสูงในหน่วยพิกเซลได้เท่ากับ 266.09 จุดพิกเซล ความสูงจริงของเสาเข็ม 12 เมตร ความสูงจริงที่โปรแกรมคำนวณได้เท่ากับ 12.06 เมตร ค่าความผิดพลาดเท่ากับ 0.5 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.3 วัดความสูงโดยอ้างอิงจากความสูงของปั้นจั่นครั้งที่สาม

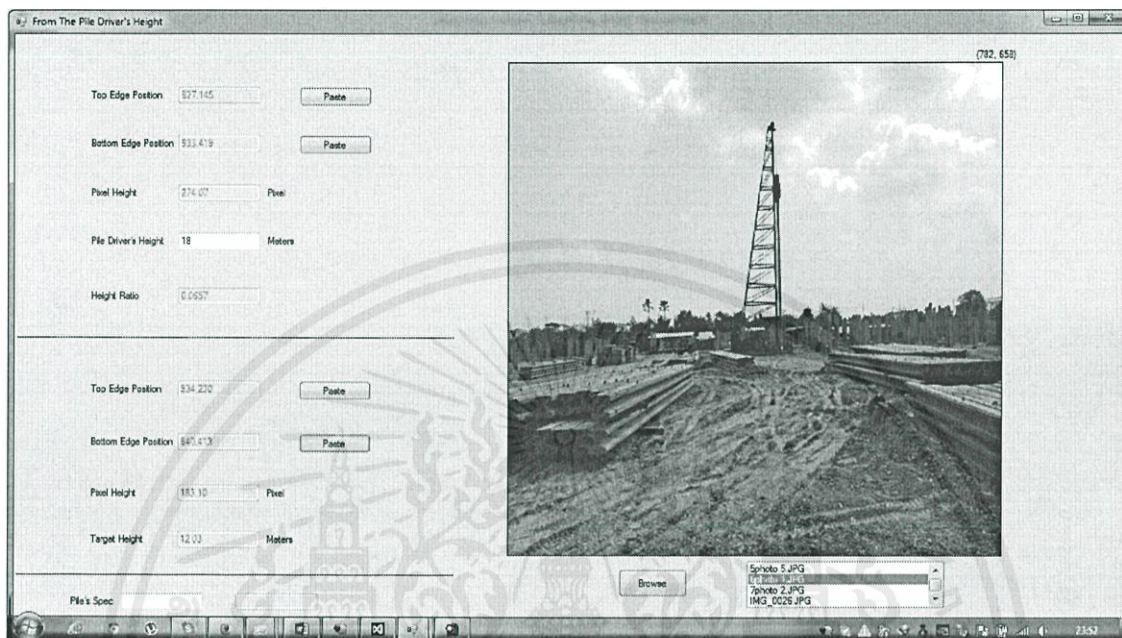


รูปที่ 4.14 วัดความสูงโดยอ้างอิงจากความสูงของปั้นจั่นครั้งที่สาม

เมื่อทำการคลิกเก็บค่าขอบบนของปั้นจั่น (886,99) และขอบล่างของปั้นจั่น (894,443) โปรแกรมคำนวณได้ความสูงในหน่วยจำนวนจุดพิกเซลเท่ากับ 344.09 จุดพิกเซล และอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดพิกเซลเท่ากับ 0.0523 จากนั้นจึงคลิกเก็บค่าขอบบนของเสาเข็ม (895,205) และขอบล่างของเสาเข็ม (895,205) โปรแกรมคำนวณความสูงในหน่วยพิกเซลได้เท่ากับ 230.08 จุดพิกเซล ความสูงจริงของเสาเข็ม 12 เมตร ความสูงจริงที่โปรแกรมคำนวณได้เท่ากับ 12.04 เมตร ค่าความผิดพลาดเท่ากับ 0.33 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.4 วัดความสูงโดยอ้างอิงจากความสูงของปั้นจั่นครั้งที่สี่



รูปที่ 4.15 วัดความสูงโดยอ้างอิงจากความสูงของปั้นจั่นครั้งที่สี่

เมื่อทำการคลิกเก็บค่าขอบบนของปั้นจั่น (927,145) และขอบล่างของปั้นจั่น (933,419) โปรแกรมคำนวณได้ความสูงในหน่วยจำนวนจุดพิกเซลเท่ากับ 274.07 จุดพิกเซล และอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดพิกเซลเท่ากับ 0.0657 จากนั้นจึงคลิกเก็บค่าขอบบนของเสาเข็ม (934,230) และขอบล่างของเสาเข็ม (940,413) โปรแกรมคำนวณความสูงในหน่วยพิกเซลได้เท่ากับ 183.10 จุดพิกเซล ความสูงจริงของเสาเข็ม 12 เมตร ความสูงจริงที่โปรแกรมคำนวณได้เท่ากับ 12.03 เมตร ค่าความผิดพลาดเท่ากับ 0.25 เปอร์เซ็นต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

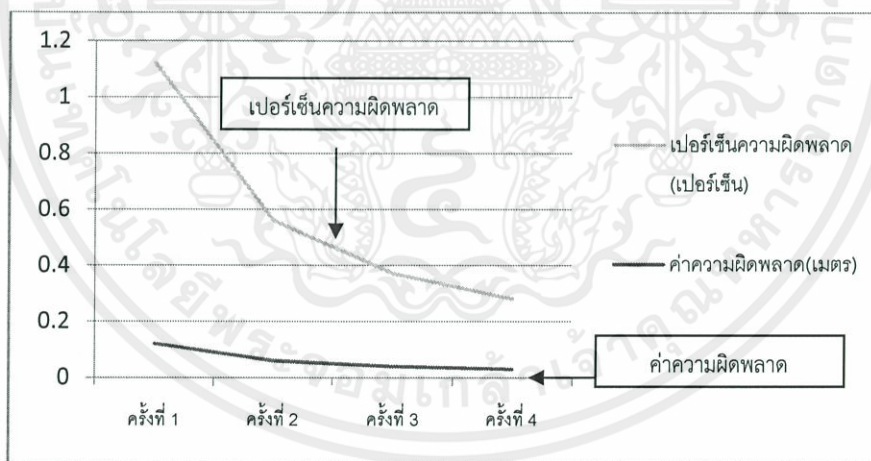
4.4.5 สรุปผลการวัดความสูงของเสาเข็มโดยอ้างอิงจากความสูงของบ้นจั่น

จากการทดสอบการวัดความสูงของเสาเข็มโดยอ้างอิงจากความสูงของบ้นจั่นนั้นจะได้ค่าความสูงจริงและความสูงที่วัดได้ พร้อมทั้งเปอร์เซ็นต์ความผิดพลาดดังนี้

ตารางที่ 4.4 สรุปผลการวัดความสูงของเสาเข็มโดยอ้างอิงจากความสูงของบ้นจั่น

ครั้งที่	ความสูงจริง (เมตร)	ความสูงที่วัดได้ (เมตร)	ค่าความผิดพลาด (เมตร)	เปอร์เซ็นต์ความผิดพลาด (เปอร์เซ็นต์)
1	12.00	11.88	0.12	1.00
2	12.00	12.06	0.06	0.50
3	12.00	12.04	0.04	0.33
4	12.00	12.03	0.03	0.25

เมื่อนำค่าจากตารางมาพล็อตเป็นกราฟจะได้กราฟดังรูป

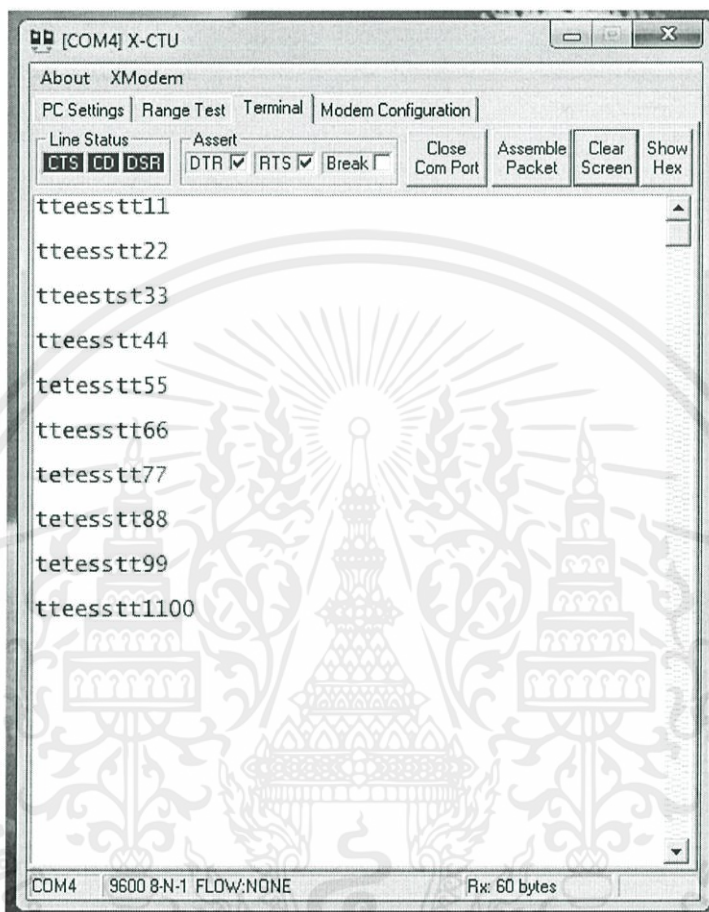


รูปที่ 4.16 กราฟแสดงค่าความผิดพลาดและเปอร์เซ็นต์ความผิดพลาด

ค่าความผิดพลาดที่เกิดขึ้นมีค่า 0.25 ถึง 1.00 เปอร์เซ็นต์ โดยมีค่าเฉลี่ยความผิดพลาดเท่ากับ 0.52 เปอร์เซ็นต์ จากกราฟแสดงให้เห็นค่าความผิดพลาดครั้งที่สี่ซึ่งมีระยะไกลมีค่าน้อยกว่าครั้งที่หนึ่งซึ่งมีระยะใกล้ เนื่องจากค่าความผิดพลาดที่เกิดจากการไม่อยู่ในระนาบเดียวกันของบ้นจั่นและเสาเข็ม จะมีค่ามากที่ระยะใกล้จากการเปลี่ยนแปลงความอัตราส่วนความสูง รูปที่ 4.4

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การรับ-ส่งข้อมูลระหว่าง XBee ผ่านโปรแกรม X-CTU

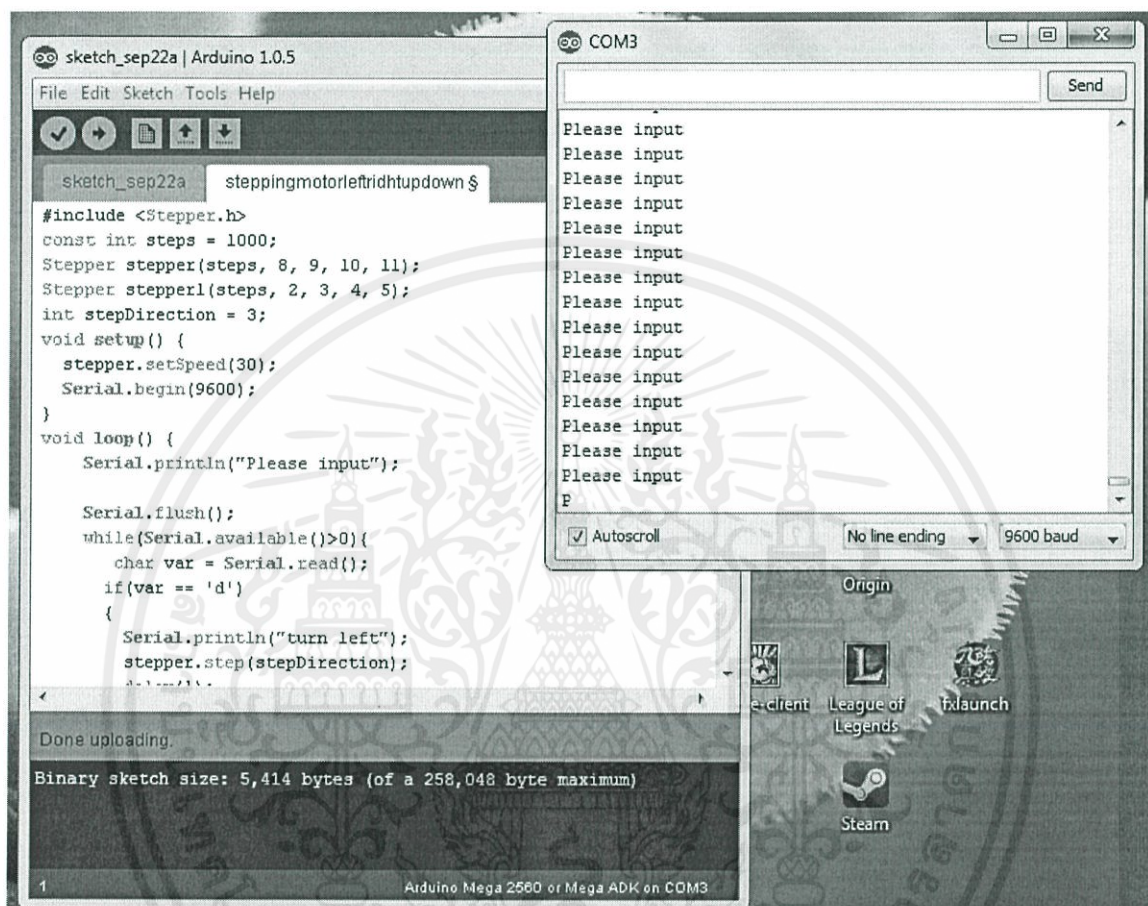


รูปที่ 4.17 การรับส่งข้อมูลระยะไกลโดยอุปกรณ์ XBee

ในการทดสอบการรับส่งข้อมูลของ XBee ทั้งสองภาคนั้น ใช้การเชื่อมต่อ Rx และ Tx ของ XBee ที่เป็นตัวปลายทางเพื่อให้มีการส่งข้อมูลกลับมายัง XBee ตัวต้นทาง โดยข้อมูลที่ส่งนั้นเป็นตัวอักษร test1 - test10 ซึ่งจะเห็นว่าเมื่อส่งอักขรใดๆไปก็ตามจะมีการส่งข้อมูลกลับมาทันที ดังนั้นจึงเห็นเป็นตัวอักษรซ้ำๆกันใน Terminal ของ XBee ภาควงที่เชื่อมต่อกับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ทดสอบไมโครคอนโทรลเลอร์ในการติดต่อกับคอมพิวเตอร์ผ่าน Serial Port

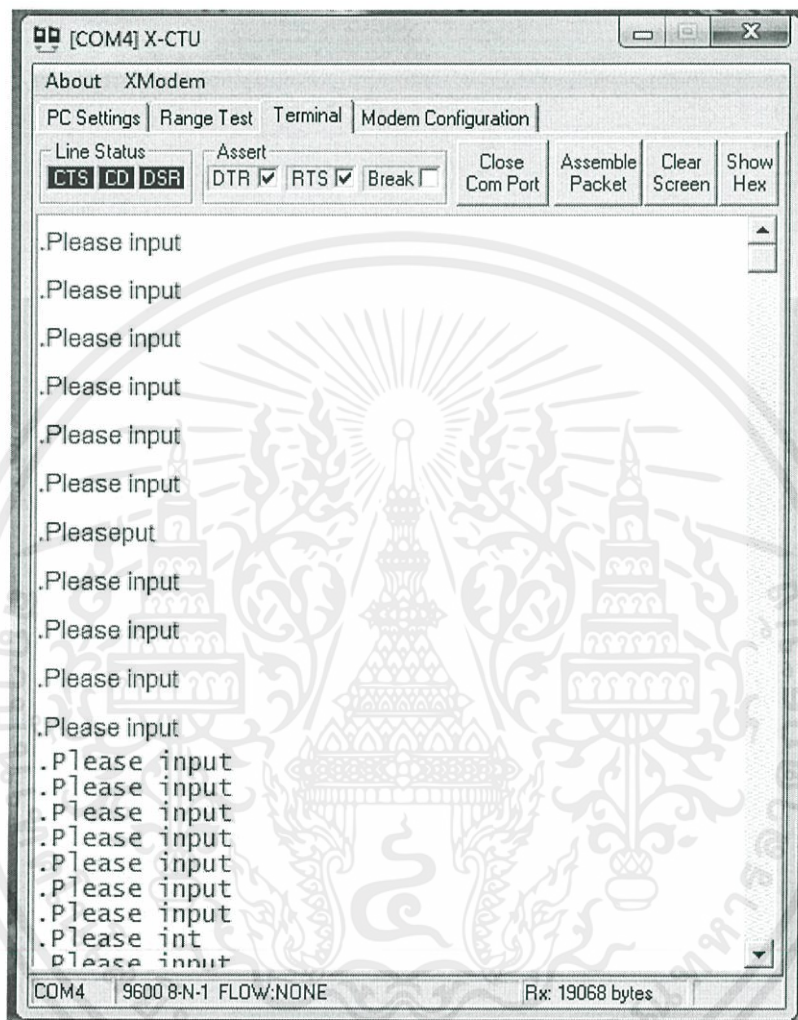


รูปที่ 4.18 โปรแกรมคอนโทรลเลอร์ Arduino ในการควบคุม Step Motor

ในการทดสอบการรับส่งข้อมูลของคอมพิวเตอร์กับไมโครคอนโทรลเลอร์นั้น ได้ทำการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ทำการส่งข้อมูลเข้าสู่คอมพิวเตอร์เป็นตัวอักษร "Please input" เพื่อแสดงให้เห็นว่าได้มีการเชื่อมต่อเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ทดสอบไมโครคอนโทรลเลอร์ในการติดต่อกับคอมพิวเตอร์ผ่านอุปกรณ์ XBee



รูปที่ 4.19 โปรแกรมคอนโทรลเลอร์ X-CTU ในการรับส่งสัญญาณกับไมโครคอนโทรลเลอร์

ในการทดสอบการรับส่งข้อมูลของคอมพิวเตอร์กับไมโครคอนโทรลเลอร์โดยผ่าน XBee ได้ทำการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ทำการส่งข้อมูลออกจากพอร์ต Serial3 เป็นตัวอักษร "Please input" เพื่อแสดงให้เห็นว่าได้มีการเชื่อมต่อ ดังจะเห็นได้จาก Terminal ของ XBee ภาคต้นทางว่ามีการส่งข้อมูลจากไมโครคอนโทรลเลอร์ผ่าน XBee ภาคปลายทางย้อนกลับเข้าสู่ XBee ต้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 ULN2003 และ Stepper Motor

เมื่อทำการส่งคำสั่งจนมาถึงไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์จะทำส่งสัญญาณเข้าสู่ ULN 2003 เพื่อขับมอเตอร์ ในส่วนของไฟเลี้ยงที่ใช้เลี้ยงมอเตอร์นั้นคือไฟ 12V เมื่อทำการวัดที่ขาภาคส่งไฟเข้าสู่เต็ปปิงมอเตอร์ของ ULN2003 จะได้ผลดังนี้

ตารางที่ 4.5 แสดงการจ่ายไฟของ ULN2003 เข้าสู่ Stepping Motor

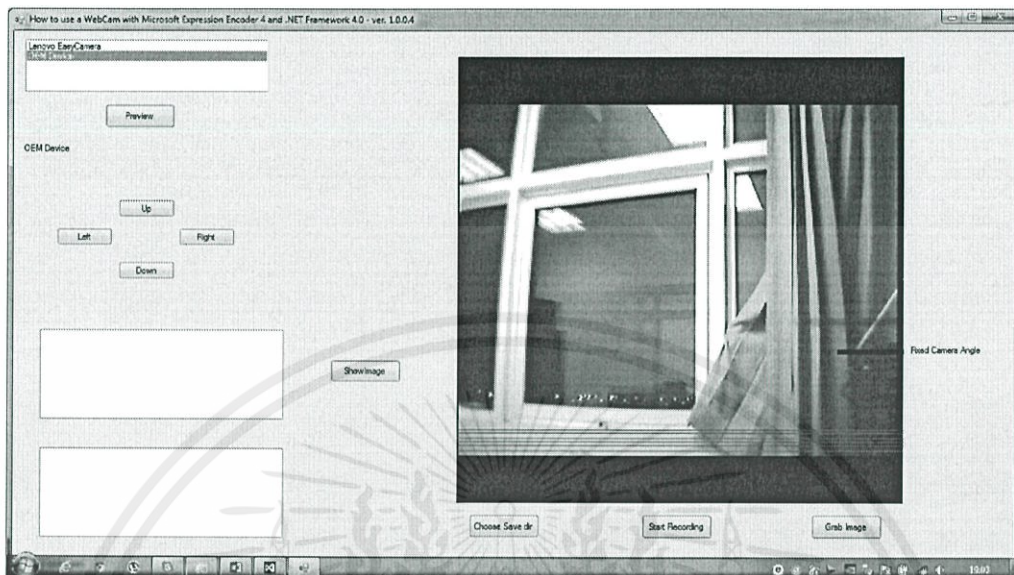
Step	ขา 16 (Phase1)	ขา 15 (Phase2)	ขา 14 (Phase3)	ขา 13 (Phase4)
Step1	0.01 V	11.74 V	11.61 V	11.61 V
Step2	-0.59 V	-0.63 V	11.65 V	11.61 V
Step3	11.74 V	-0.64 V	11.61 V	11.68 V
Step4	11.67 V	-0.65 V	-0.59 V	11.61 V
Step5	11.61 V	11.74 V	-0.63 V	11.67 V
Step6	11.67 V	11.57 V	0.01 V	0.01 V
Step7	11.67 V	11.57 V	11.57 V	-0.58 V
Step8	-0.59 V	11.66 V	11.74 V	-0.65 V

จากรูป จะพบว่าเมื่อทำการส่งตัวอักษร ASCII หนึ่งครั้งเท่ากับสั่งการให้ไมโครคอนโทรลเลอร์ส่งสัญญาณกระตุ้นเข้าสู่ ULN2003 หนึ่งครั้ง หรือทำการเคลื่อนที่ 1 Step (7.5 องศา) โดยมีการส่งสัญญาณไฟแรงดัน 12 โวลต์ออกมาที่ขา 13,14,15 (กระตุ้นเฟส 2,3,4) และมีค่าตามตารางไปเรื่อยๆจนกระทั่ง Step8 จากนั้นจึงมีการวนกลับไปส่งสัญญาณใน Step ที่ 1 อีกครั้ง

4.9 ทดสอบการรับ-ส่งภาพไร้สายที่ระยะต่างๆ

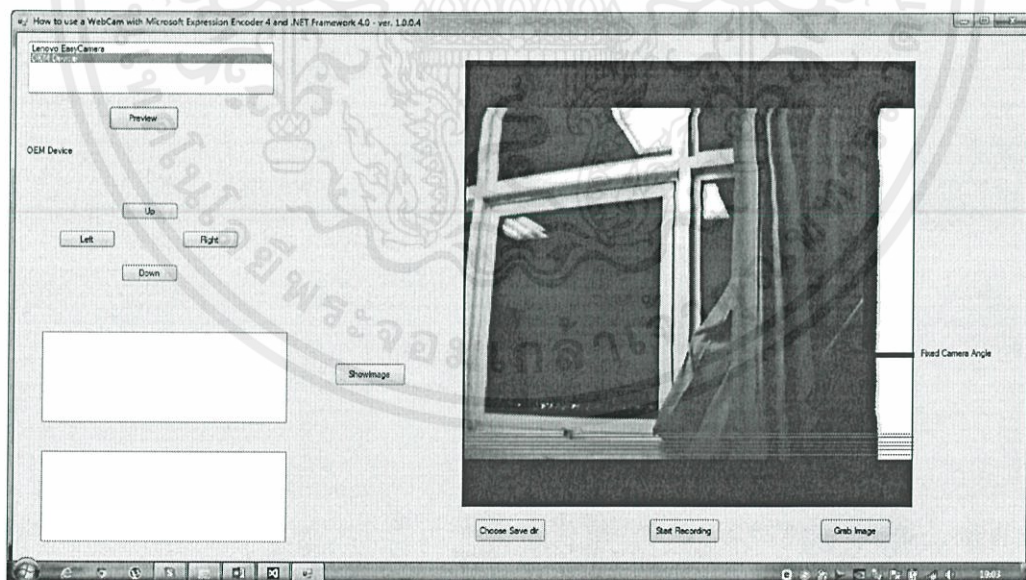
ในการทดสอบประสิทธิภาพในการรับ-ส่งภาพจะทำการวัดระยะห่างตั้งแต่ระยะเริ่มต้นที่สามารถรับภาพได้ชัดเจนไปจนถึงระยะที่เริ่มรับภาพได้ไม่ดีและระยะที่ไม่สามารถรับภาพได้เลยเพื่อหาระยะทางสูงสุดที่ระบบรับ-ส่งภาพไร้สายสามารถทำงานได้ โดยการวัดจะทำการวางกล้องไว้ในห้องทดลองและเครื่องรับอยู่นอกห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



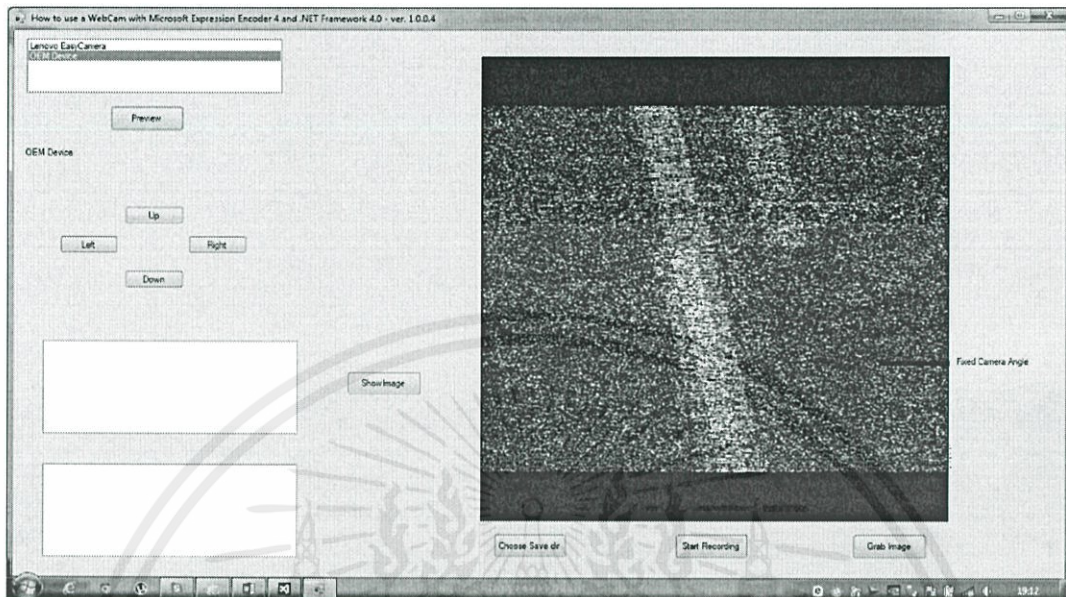
รูปที่ 4.20 ตำแหน่งรับ-ส่งสัญญาณระยะ 30 เมตร

เมื่อวางกล้องไว้ที่ระยะห่างจากเครื่องรับสัญญาณ 30 เมตร จะเห็นได้ว่ายังคงรับภาพได้อย่างชัดเจน



รูปที่ 4.21 ตำแหน่งรับ-ส่งสัญญาณระยะ 40 เมตร

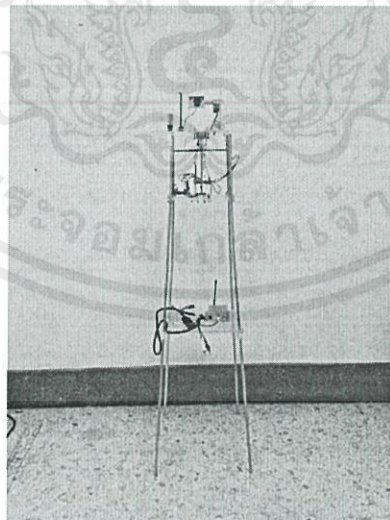
เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยพระจอมเกล้าพระนครเหนือ เมื่อวางกล้องไว้ที่ระยะห่างจากเครื่องรับสัญญาณ 40 เมตร จะเห็นได้ว่าภาพเริ่มมีการริบหรี่แต่การถ่ายภาพยังสามารถใช้งานได้ทั้งที่มีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 ตำแหน่งรับ-ส่งสัญญาณระยะ 45 เมตร

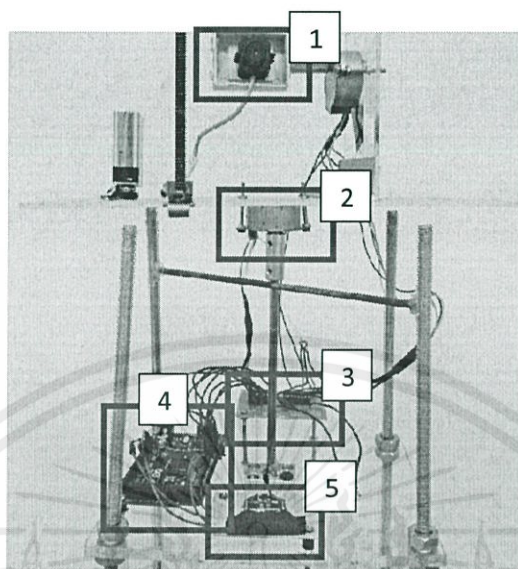
เมื่อวางกล้องไว้ที่ระยะห่างจากเครื่องรับสัญญาณ 45 เมตรในที่โล่ง จะเห็นได้ว่าไม่สามารถรับภาพได้

4.10 อุปกรณ์ควบคุมกล้องโดยรวม



รูปที่ 4.21 อุปกรณ์ควบคุมกล้องโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 อุปกรณ์ควบคุมกล้องโดยรวม

หมายเลข 1 คือ กล้องไร้สาย (Wireless Camera)

หมายเลข 2 คือ สเต็ปปีงมอเตอร์ขนาด 12V ในส่วนควบคุมการหมุนของกล้อง

หมายเลข 3 คือ IC ULN2003 ใช้ในการขับให้สเต็ปปีงมอเตอร์มีการเคลื่อนไหว

หมายเลข 4 คือ ไมโครคอนโทรลเลอร์ Arduino Mega 2560

หมายเลข 5 คือ XBee ฝั่งรับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

สามารถออกแบบโปรแกรมในการวัดความสูงของเสาเข็มโดยการอ้างอิงสองระบบ ระบบที่หนึ่งคือการอ้างอิงจากระยะห่างระหว่างกล้องกับเสาเข็ม ระบบที่สองคือการอ้างอิงจากความสูงของปั้นจั่น โดยใช้อัตราส่วนความสูงระหว่างหน่วยเมตรต่อจำนวนจุดพิกเซล ซึ่งค่าความผิดพลาดจากวิธีอ้างอิงจากระยะห่างระหว่างกล้องกับเสาเข็มมีค่าเฉลี่ยประมาณ 1.22 เปอร์เซ็นต์ และ ค่าความผิดพลาดจากวิธีอ้างอิงจากความสูงของปั้นจั่นมีค่าเฉลี่ยประมาณ 0.52 เปอร์เซ็นต์

สามารถออกแบบระบบควบคุมกล้องไร้สายได้โดยใช้ไมโครคอนโทรลเลอร์ Arduino ร่วมกับ IC ULN2003 เพื่อขับสเต็ปมอเตอร์และส่งสัญญาณไร้สายผ่านอุปกรณ์ XBee ซึ่งใช้ความถี่ในการส่งสัญญาณ 2.4 GHz รวมไปถึง ระบบส่งสัญญาณภาพไร้สายโดยใช้กล้องไร้สาย (Wireless Camera) ความถี่ 1.2GHz และแปลงสัญญาณภาพอนาล็อกเป็นดิจิทัลเพื่อนำภาพเข้าสู่คอมพิวเตอร์โดยอุปกรณ์ EasyCAP USB 2.0

5.2 ข้อเสนอแนะ

1. การใช้ค่าอัตราส่วนความสูงในหน่วยเมตรต่อจำนวนจุดพิกเซลจะสามารถใช้ได้กับกล้องเฉพาะรุ่นเท่านั้น เนื่องจากเลนส์ของกล้องแต่ละชนิดจะมีขนาดของจุดโฟกัสไม่เท่ากัน ทำให้ภาพมีขนาดไม่เท่ากันด้วย
2. การวัดความสูงของเสาเข็มโดยอ้างอิงจากระยะห่างระหว่างกล้องและเสาเข็ม เพื่อความสะดวกในการวัดระยะสามารถใช้อุปกรณ์วัดระยะด้วยแสงเลเซอร์ประกอบเพื่อวัดระยะได้รวดเร็วและสะดวกมากยิ่งขึ้น
3. การวัดความสูงทั้งสองระบบ คุณภาพของกล้องและความละเอียดของภาพที่มากขึ้นจะมีส่วนในการลดค่าความผิดพลาดที่เกิดให้น้อยลง
4. โปรแกรมสามารถเพิ่มเติมการหาความเอียงของเสาเข็มได้จากการหาค่ามุมเทียบกับแกน Y จากค่าพิกัดของจุดขอบบนและขอบล่างของเสาเข็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] ฉลวย ศรีเมืองและชาคริต เสวตจาม. "เครื่องส่งโทรทัศนระบบยูเอชเอฟ ช่อง 22 กำลังส่ง 10 วัตต์" ปรินิพนธ์ปริญาวิศวกรรมบัณฑิต, ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2549
- [2] นายประยูร เอี่ยมพาทล และเอกฉัตร พรหมจำปา. "เครื่องรับส่งวิทยุเอฟเอ็ม." ปรินิพนธ์ปริญาวิศวกรรมบัณฑิต, ภาควิชาอุตสาหกรรม คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2535
- [3] นายโฆสิต ลิ้มจันทร์ประภาพ นายฉัตรชัย บังจิตร และนายพงษ์เทพ สุตมี. "ระบบตรวจจับเมื่อมีวัตถุตัดผ่านแบบไร้สาย" ปรินิพนธ์ปริญา วิศวกรรมบัณฑิต, ภาควิชาอุตสาหกรรม คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2554
- [4] รศ.ดร. ชูชาติ ปิณฑวิรุจน์. "การประมวลผลภาพดิจิทัลด้วย C++" คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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.Runtime.InteropServices;
using System.Diagnostics;

// Reference path for the following assemblies --> C:\Program Files\Microsoft
Expression\Encoder 4\SDK\
using Microsoft.Expression.Encoder.Devices;
using Microsoft.Expression.Encoder.Live;
using Microsoft.Expression.Encoder;

using EE4WebCam;
using System.IO;
using System.Collections;
using System.Drawing.Imaging;
using System.Timers;

namespace EE4Test
{
    public partial class frmEE4WebCam : Form
    {
        Boolean key_w_stat = false;
        Boolean key_a_stat = false;
        Boolean key_s_stat = false;
        Boolean key_d_stat = false;
        string turnup;
        string turnleft;
        string turnright;
        string turndown;
        /// <summary>
        /// Creates job for capture of live source
        /// </summary>
        private LiveJob _job;

        /// <summary>
        /// Device for live source
        /// </summary>
        private LiveDeviceSource _deviceSource;

        private bool _bStartedRecording = false;

        private Size VideoSize = new Size(600, 600);

        System.Timers.Timer BtnUpTimer;
        System.Timers.Timer BtnLeftTimer;
        System.Timers.Timer BtnRightTimer;
        System.Timers.Timer BtnDownTimer;
    }
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ว่ากรณีใดๆก็ตาม ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public frmEE4WebCam()
{
    InitializeComponent();
}

private void frmEE4WebCam_Load(object sender, EventArgs e)
{
    this.Text += " - ver. " + Application.ProductVersion;

    lstVideoDevices.ClearSelected();
    foreach (EncoderDevice edv in
EncoderDevices.FindDevices(EncoderDeviceType.Video))
    {
        lstVideoDevices.Items.Add(edv.Name);
    }
    lblVideoDeviceSelectedForPreview.Text = "";

    serialPort1.Open();
}

private void btnPreview_Click(object sender, EventArgs e)
{
    EncoderDevice video = null;
    EncoderDevice audio = null;

    GetSelectedVideoAndAudioDevices(out video);
    StopJob();

    if (video == null)
    {
        return;
    }

    // Starts new job for preview window
    _job = new LiveJob();

    // Checks for a/v devices
    if (video != null)
    {
        // Create a new device source. We use the first audio and video
devices on the system
        _deviceSource = _job.AddDeviceSource(video, audio);

        // Is it required to show the configuration dialogs ?
        //if (checkBoxShowConfigDialog.Checked)
        //{
        //    // Yes
        //    // VFW video device ?
        //    if (lstVideoDevices.SelectedItem.ToString().EndsWith("(VFW)",
StringComparison.OrdinalIgnoreCase))
        {
            // ...
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดก็ตาม และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//      {
//          // Yes
//          if
(_deviceSource.IsDialogSupported(ConfigurationDialog.VfwFormatDialog))
//          {
//
_deviceSource.ShowConfigurationDialog(ConfigurationDialog.VfwFormatDialog, (new
HandleRef(panelVideoPreview, panelVideoPreview.Handle)));
//          }

//          if
(_deviceSource.IsDialogSupported(ConfigurationDialog.VfwSourceDialog))
//          {
//
_deviceSource.ShowConfigurationDialog(ConfigurationDialog.VfwSourceDialog, (new
HandleRef(panelVideoPreview, panelVideoPreview.Handle)));
//          }

//          if
(_deviceSource.IsDialogSupported(ConfigurationDialog.VfwDisplayDialog))
//          {
//
_deviceSource.ShowConfigurationDialog(ConfigurationDialog.VfwDisplayDialog, (new
HandleRef(panelVideoPreview, panelVideoPreview.Handle)));
//          }
//      }
//      else
//      {
//          // No
//          if
(_deviceSource.IsDialogSupported(ConfigurationDialog.VideoCapturePinDialog))
//          {
//
_deviceSource.ShowConfigurationDialog(ConfigurationDialog.VideoCapturePinDialog,
(new HandleRef(panelVideoPreview, panelVideoPreview.Handle)));
//          }

//          if
(_deviceSource.IsDialogSupported(ConfigurationDialog.VideoCaptureDialog))
//          {
//
_deviceSource.ShowConfigurationDialog(ConfigurationDialog.VideoCaptureDialog, (new
HandleRef(panelVideoPreview, panelVideoPreview.Handle)));
//          }

//          if
(_deviceSource.IsDialogSupported(ConfigurationDialog.VideoCrossbarDialog))
//          {
//
_deviceSource.ShowConfigurationDialog(ConfigurationDialog.VideoCrossbarDialog,
(new HandleRef(panelVideoPreview, panelVideoPreview.Handle)));

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//      if
(_deviceSource.IsDialogSupported(ConfigurationDialog.VideoPreviewPinDialog))
//      {
//
_deviceSource.ShowConfigurationDialog(ConfigurationDialog.VideoPreviewPinDialog,
(new HandleRef(panelVideoPreview, panelVideoPreview.Handle)));
//      }

//      if
(_deviceSource.IsDialogSupported(ConfigurationDialog.VideoSecondCrossbarDialog))
//      {
//
_deviceSource.ShowConfigurationDialog(ConfigurationDialog.VideoSecondCrossbarDialog,
(new HandleRef(panelVideoPreview, panelVideoPreview.Handle)));
//      }
//    }
//}
//else
//{
//  // No
//  // Setup the video resolution and frame rate of the video
device
//  // NOTE: Of course, the resolution and frame rate you
specify must be supported by the device!
//  // NOTE2: May be not all video devices support this call,
and so it just doesn't work, as if you don't call it (no error is raised)
//  // NOTE3: As a workaround, if the .PickBestVideoFormat
method doesn't work, you could force the resolution in the
//  //      following instructions (called few lines below):
'panelVideoPreview.Size=' and '_job.OutputFormat.VideoProfile.Size='
//  //      to be the one you choosed (640, 480).
_deviceSource.PickBestVideoFormat(new Size(VideoSize.Width,
VideoSize.Height), 15);
//}

// Get the properties of the device video
SourceProperties sp = _deviceSource.SourcePropertiesSnapshot();

// Resize the preview panel to match the video device resolution
set
//panelVideoPreview.Size = new Size(sp.Size.Width,
sp.Size.Height);
panelVideoPreview.Size = new Size(VideoSize.Width,
VideoSize.Height);

// Setup the output video resolution file as the preview
//_job.OutputFormat.VideoProfile.Size = new Size(sp.Size.Width,
sp.Size.Height);
_job.OutputFormat.VideoProfile.Size = new Size(VideoSize.Width,
VideoSize.Height);

```

```

// Display the video device properties set
toolStripStatusLabel1.Text = sp.Size.Width.ToString() + "x" +
sp.Size.Height.ToString() + " " + sp.FrameRate.ToString() + " fps";

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของ บริษัท ไซเบอร์เนชั่น จำกัด
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        // Sets preview window to winform panel hosted by xaml window
        _deviceSource.PreviewWindow = new PreviewWindow(new
HandleRef(panelVideoPreview, panelVideoPreview.Handle));

        // Make this source the active one
        _job.ActivateSource(_deviceSource);

        btnStartStopRecording.Enabled = true;
        btnGrabImage.Enabled = true;

        toolStripStatusLabel1.Text = "Preview activated";
    }
    else
    {
        // Gives error message as no audio and/or video devices found
        MessageBox.Show("No Video/Audio capture devices have been found.",
"Warning");
        toolStripStatusLabel1.Text = "No Video/Audio capture devices have
been found.";
    }
}

private void btnStartStopRecording_Click(object sender, EventArgs e)
{
    // Is it Recoring ?
    if (_bStartedRecording)
    {
        // Yes
        // Stops encoding
        _job.StopEncoding();
        btnStartStopRecording.Text = "Start Recording";
        toolStripStatusLabel1.Text = "";
        _bStartedRecording = false;
    }
    else
    {
        // Sets up publishing format for file archival type
        FileArchivePublishFormat fileOut = new FileArchivePublishFormat();

        // Sets file path and name
        fileOut.OutputFileName =
String.Format("C:\\WebCam{0:yyyyMMdd_hhmmss}.wmv", DateTime.Now);

        // Adds the format to the job. You can add additional formats as
well such as
        // Publishing streams or broadcasting from a port
        _job.PublishFormats.Add(fileOut);

        // Start encoding
        _job.StartEncoding();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์โดยมหาวิทยาลัยราชภัฏวชิรเวศน์บุรีรัมย์ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น } ยกเว้นให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    private void cmdGrabImage_Click(object sender, EventArgs e)
    {
        // Create a Bitmap of the same dimension of panelVideoPreview (Width x
        Height)
        using (Bitmap bitmap = new Bitmap(panelVideoPreview.Width,
        panelVideoPreview.Height))
        {
            using (Graphics g = Graphics.FromImage(bitmap))
            {
                // Get the paramters to call g.CopyFromScreen and get the
                image
                Rectangle rectanglePanelVideoPreview =
                panelVideoPreview.Bounds;
                Point sourcePoints = panelVideoPreview.PointToScreen(new
                Point(panelVideoPreview.ClientRectangle.X, panelVideoPreview.ClientRectangle.Y));
                g.CopyFromScreen(sourcePoints, Point.Empty,
                rectanglePanelVideoPreview.Size);
            }
            string pathSaveFile = @"D:\Snapshot";
            if (!System.IO.Directory.Exists(pathSaveFile))
            {
                Directory.CreateDirectory(pathSaveFile);
            }
            string strGrabFileName = String.Format(pathSaveFile +
            @"\Snapshot_{0:yyyyMMdd_hhmmss}.jpg", DateTime.Now);
            toolStripStatusLabel1.Text = strGrabFileName;
            bitmap.Save(strGrabFileName,
            System.Drawing.Imaging.ImageFormat.Jpeg);
        }
    }

    //private void Broadcast_Click(object sender, EventArgs e)
    //{
    //    EncoderDevice video = null;
    //    EncoderDevice audio = null;

    //    GetSelectedVideoAndAudioDevices(out video);
    //    StopJob();

    //    if (video == null)
    //    {
    //        return;
    //    }

    //    _job = new LiveJob();

    //    _deviceSource = _job.AddDeviceSource(video, audio);
    //    _job.ActivateSource(_deviceSource);

    //    // Finds and applys a smooth streaming preset
    //    _job.ApplyPreset(LivePresets.VC1256kDSL16x9);

```

เอกสารนี้เป็นเอกสารที่... ให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// // Creates the publishing format for the job
// PullBroadcastPublishFormat format = new
PullBroadcastPublishFormat();
// format.BroadcastPort = 8080;
// format.MaximumNumberOfConnections = 2;

// // Adds the publishing format to the job
// _job.PublishFormats.Add(format);

// // Starts encoding
// _job.StartEncoding();

// toolStripStatusLabel1.Text = "Broadcast started on localhost at port
8080, run WpfShowBroadcast.exe now to see it";
//}

private void GetSelectedVideoAndAudioDevices(out EncoderDevice video)
{
    video = null;
    lblVideoDeviceSelectedForPreview.Text = "";
    if (lstVideoDevices.SelectedIndex < 0)
    {
        MessageBox.Show("No Video capture devices have been
selected.\nSelect a video devices from the listboxes and try again.", "Warning");
        return;
    }
    // Get the selected video device
    foreach (EncoderDevice edv in
EncoderDevices.FindDevices(EncoderDeviceType.Video))
    {
        if (String.Compare(edv.Name,
lstVideoDevices.SelectedItem.ToString()) == 0)
        {
            video = edv;
            lblVideoDeviceSelectedForPreview.Text = edv.Name;
            break;
        }
    }
}

void StopJob()
{
    // Has the Job already been created ?
    if (_job != null)
    {
        // Yes
        // Is it capturing ?
        //if (_job.IsCapturing)
        if (_bStartedRecording)
        {
            // Yes
            // Stop Capturing
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามทำซ้ำ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        btnStartStopRecording.PerformClick();
    }

    _job.StopEncoding();

    // Remove the Device Source and destroy the job
    _job.RemoveDeviceSource(_deviceSource);

    // Destroy the device source
    _deviceSource.PreviewWindow = null;
    _deviceSource = null;
}

}

private void frmEE4WebCam_FormClosing(object sender, FormClosingEventArgs
e)
{
    StopJob();
}

private void btn_saveFolder_Click(object sender, EventArgs e)
{
    if (folderBrowserDialog1.ShowDialog() == DialogResult.OK)
    {
        lstBx_mediaFile.Items.Clear();
        string path = folderBrowserDialog1.SelectedPath;
        this.label_saveDirectory.Text = path;

        string[] filePicVid = getFiles(path,
        "*.jpg|*.jpeg|*.png|*.avi|*.wma", SearchOption.AllDirectories);

        if (filePicVid.Count() > 0)
        {
            lstBx_mediaFile.Items.AddRange(filePicVid);
        }
    }
}

public string[] getFiles(string SourceFolder, string Filter,
System.IO.SearchOption searchOption)
{
    // ArrayList will hold all file names
    ArrayList alFiles = new ArrayList();

    // Create an array of filter string
    string[] MultipleFilters = Filter.Split('|');
    foreach (string FileFilter in MultipleFilters)
    {
        if (string.IsNullOrEmpty(SourceFolder))
            SourceFolder = this.label_saveDirectory.Text;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

searchOption);
    string[] fileWPath = Directory.GetFiles(SourceFolder, FileFilter,
    foreach (string file in fileWPath)
    {
        alFiles.Add(Path.GetFileName(file));
    }
    return (string[])alFiles.ToArray(typeof(string));
}

public static void OpenShowImages()
{
    ShowImages show = new ShowImages();
    show.ShowDialog();
}

private void ShowImage_Click(object sender, EventArgs e)
{
    System.Threading.Thread t = new System.Threading.Thread(new
System.Threading.ThreadStart(OpenShowImages));
    t.SetApartmentState(System.Threading.ApartmentState.STA);
    t.Start();
}

private void goto_Calibrate_Click(object sender, EventArgs e)
{
    System.Threading.Thread t = new System.Threading.Thread(new
System.Threading.ThreadStart(OpenCalibrate));
    t.SetApartmentState(System.Threading.ApartmentState.STA);
    t.Start();
}

private void OpenCalibrate()
{
    Calibrate cb = new Calibrate();
    cb.ShowDialog();
}

private void timer1_Thick(object sender, EventArgs e)
{
    if (key_w_stat == true)
    {
        turnup = "w";
        serialPort1.Write(turnup);
    }

    if (key_a_stat == true)
    {
        turnleft = "a";
        serialPort1.Write(turnleft);
    }

    if (key_s_stat == true)
    {
        turndown = "s";
        serialPort1.Write(turndown);
    }
}

```

เอกสารนี้เป็นเอกสาร { สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังอาจถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    if (key_d_stat == true)
    {
        turnright = "d";
        serialPort1.Write(turnright);
    }
}

private void btn_Up_MouseDown(object sender, MouseEventArgs e)
{
    key_w_stat = true;
}

private void btn_Up_MouseUp(object sender, MouseEventArgs e)
{
    key_w_stat = false;
}

private void btn_Right_MouseDown(object sender, MouseEventArgs e)
{
    key_d_stat = true;
}

private void btn_Right_MouseUp(object sender, MouseEventArgs e)
{
    key_d_stat = false;
}

private void btn_Down_MouseDown(object sender, MouseEventArgs e)
{
    key_s_stat = true;
}

private void btn_Down_MouseUp(object sender, MouseEventArgs e)
{
    key_s_stat = false;
}

private void btn_Left_MouseDown(object sender, MouseEventArgs e)
{
    key_a_stat = true;
}

private void btn_Left_MouseUp(object sender, MouseEventArgs e)
{
    key_a_stat = false;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 }
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข
หน้าต่างโปรแกรมวัดค่าความสูงโดยอ้างอิงจากระยะห่างระหว่างกล้องและเสาเข็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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 EE4Test;
using System.IO;
using System.Collections;

namespace EE4WebCam
{
    public partial class ShowImages : Form
    {
        private string FilePath;

        private Dictionary<int, string> tableDic;

        public ShowImages()
        {
            InitializeComponent();
        }

        private void ShowImages_Load(object sender, EventArgs e)
        {
            //this.Size = Screen.PrimaryScreen.Bounds.Size;
            //this.Dock = DockStyle.Left;

            Size lstBxFileSize = new Size()
            {
                Height = this.Height - btn_Browse.Size.Height - 10,
                Width = this.Width / 5
            };

            //lstBx_listFile.Size = lstBxFileSize;
            //lstBx_listFile.Dock = DockStyle.Left;

            Size imageBox = new Size()
            {
                Height = this.Height,
                Width = (this.Width * 4) / 5
            };
            //box_Img.Size = imageBox;
            //box_Img.Dock = DockStyle.Right;
            //box_Img.Left = lstBxFileSize.Width + 60;
            //ปรับขนาดภาพ
            box_Img.SizeMode = PictureBoxSizeMode.StretchImage;

            //box_Img.Image = Image.FromFile(@"C:\Users\Public\Pictures\Sample
Pictures\Koala.jpg");
            //btn_Browse.Dock = DockStyle.Bottom;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        foreach (var item in tableDic)
        {
            comboTableList.Items.Add(item.Value);
        }
        comboTableList.SelectedIndex = 0;
    }

    private void btn_Browse_Click(object sender, EventArgs e)
    {
        if (folderBrowserDialog1.ShowDialog() == DialogResult.OK)
        {
            lstBx_listFile.Items.Clear();
            string path = FilePath = folderBrowserDialog1.SelectedPath;
            //this.label_saveDirectory.Text = path;

            string[] filePicVid = getFiles(path, "*.jpg|*.jpeg|*.png",
SearchOption.AllDirectories);

            if (filePicVid.Count() > 0)
            {
                lstBx_listFile.Items.AddRange(filePicVid);
            }
        }
    }

    public string[] getFiles(string SourceFolder, string Filter,
System.IO.SearchOption searchOption)
    {
        // ArrayList will hold all file names
        ArrayList alFiles = new ArrayList();

        // Create an array of filter string
        string[] MultipleFilters = Filter.Split('|');

        // for each filter find mathing file names
        foreach (string FileFilter in MultipleFilters)
        {
            // add found file names to array list

            string[] fileWPath = Directory.GetFiles(SourceFolder, FileFilter,
searchOption);
            foreach (string file in fileWPath)
            {
                alFiles.Add(Path.GetFileName(file));
            }
        }

        // returns string array of relevant file names
        return (string[])alFiles.ToArray(typeof(string));
    }
}

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยราชภัฏวไลยอลงกรณ์
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

private void ChangeImage(object sender, EventArgs e)
{
    this.SetImage(FilePath + "\\\" + lstBx_listFile.SelectedItem);
}

private void SetImage(string filePath)
{
    try
    {
        box_Img.Image = Image.FromFile(filePath);
    }
    catch (FileNotFoundException)
    {
        box_Img.Image =
Image.FromFile(AppDomain.CurrentDomain.BaseDirectory + "\\fnf.jpg");
    }
}

private void GetPositionOnClick(object sender, MouseEventArgs e)
{
    Point P = Control.MousePosition;
    Clipboard.SetText(P.X + "," + P.Y);
}

private void btn_TargetP1_Click(object sender, EventArgs e)
{
    txtb_TargetP1.Text = Clipboard.GetText();
}

private void btn_TargetPoint2_Click(object sender, EventArgs e)
{
    txtb_TargetP2.Text = Clipboard.GetText();
}

private void btn_CalcTargD_Click(object sender, EventArgs e)
{
    string strP1 = txtb_TargetP1.Text;
    string strP2 = txtb_TargetP2.Text;

    if (!string.IsNullOrEmpty(strP1) && !string.IsNullOrEmpty(strP2))
    {
        try
        {
            string[] arrP1 = strP1.Split(',');
            string[] arrP2 = strP2.Split(',');

            double x1 = double.Parse(arrP1[0]);
            double y1 = double.Parse(arrP1[1]);
            double x2 = double.Parse(arrP2[0]);
            double y2 = double.Parse(arrP2[1]);

            double d = Math.Sqrt((Math.Pow((x2 - x1), 2) + Math.Pow((y2 -
ไม่ว่า (y1), 2))););
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่า (y1), 2)));; อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        txtb_TargetD.Text = string.Format("{0:N2}", d);
    }
    catch (Exception ex)
    {
    }
}

private void box_Img_Click(object sender, EventArgs e)
{
    Point P = Control.MousePosition;
    Clipboard.SetText(P.X + "," + P.Y);
}

private void btn_GoToDistTbl_Click(object sender, EventArgs e)
{
    System.Threading.Thread t = new System.Threading.Thread(new
System.Threading.ThreadStart(OpenDistanceTbl));
    t.SetApartmentState(System.Threading.ApartmentState.STA);
    t.Start();
}
private void ShowMousePosition(object sender, MouseEventArgs e)
{
    Point m = Control.MousePosition;

    lbl_MousePos.Text = "(" + m.X + ", " + m.Y + ")";
}
public void OpenDistanceTbl()
{
    DistanceTable dt = new DistanceTable();
    dt.ShowDialog();
}

private void EventSubtractTextBoxesChange(object sender, EventArgs e)
{
    string strP1 = txtb_TargetP1.Text;
    string strP2 = txtb_TargetP2.Text;

    if (!string.IsNullOrEmpty(strP1) && !string.IsNullOrEmpty(strP2))
    {
        try
        {
            string[] arrP1 = strP1.Split(',');
            string[] arrP2 = strP2.Split(',');

            double x1 = double.Parse(arrP1[0]);
            double y1 = double.Parse(arrP1[1]);
            double x2 = double.Parse(arrP2[0]);
            double y2 = double.Parse(arrP2[1]);

            double d = Math.Sqrt((Math.Pow((x2 - x1), 2) + Math.Pow((y2 -
เอกส y1), 2)));
        }
        catch { }
    }
    txtb_TargetD.Text = string.Format("{0:N2}", d);
}

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้า



ภาคผนวก ค

หน้าต่างโปรแกรมวัดค่าความสูงโดยอ้างอิงจากความสูงของบันจันที่ใช้ตอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Windows.Forms;

```

```

namespace EE4WebCam
{

```

```

    public partial class Calibrate : Form
    {
        public const string CancelTxt = "Cancel";
        public string TempText = "";
        public bool IsWaitMouseClicked = false;
        private string FilePath;
        Point P;

        public Calibrate()
        {
            InitializeComponent();
        }
        private void Form_Load(object sender, EventArgs e)
        {
            P = new Point();
            txtb_P1.Enabled = false;
            txtb_P2.Enabled = false;
            txtb_D.Enabled = false;
            txtb_Ratio.Enabled = false;
            txtb_TargetP1.Enabled = false;
            txtb_TargetP2.Enabled = false;
            txtb_TargetD.Enabled = false;
            txtb_TargetHeight.Enabled = false;
            txtb_BaseHeight.Enabled = false;

            picb_MainPic.SizeMode = PictureBoxSizeMode.StretchImage;
        }
        private void CalculateHeight()
        {
            double knownHeight = double.Parse(txtb_D.Text);
        }

        private void ShowMousePosition(object sender, MouseEventArgs e)
        {
            Point m = Control.MousePosition;

            lbl_MousePos.Text = "(" + m.X + ", " + m.Y + ")";
        }

```

```

        private void GetPositionOnClick(object sender, MouseEventArgs e)

```

```

        {
            P = Control.MousePosition;

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
 ไม่ว่ากรณีใดๆทั้งสิ้น ยี่สิบแปด

และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Clipboard.SetText(P.X + "," + P.Y);
    }

    private void btn_P1_Click(object sender, EventArgs e)
    {
        txtb_P1.Text = Clipboard.GetText();
    }

    private void btn_P2_Click(object sender, EventArgs e)
    {
        txtb_P2.Text = Clipboard.GetText();
    }

    private void btn_CalcD_Click(object sender, EventArgs e)
    {
        string strP1 = txtb_P1.Text;
        string strP2 = txtb_P2.Text;

        if (!string.IsNullOrEmpty(strP1) && !string.IsNullOrEmpty(strP2))
        {
            try
            {
                string[] arrP1 = strP1.Split(',');
                string[] arrP2 = strP2.Split(',');

                double x1 = double.Parse(arrP1[0]);
                double y1 = double.Parse(arrP1[1]);
                double x2 = double.Parse(arrP2[0]);
                double y2 = double.Parse(arrP2[1]);

                double d = Math.Sqrt((Math.Pow((x2 - x1), 2) + Math.Pow((y2 -
y1), 2)));

                txtb_D.Text = string.Format("{0:N2}", d);
            }
            catch (Exception ex)
            {
            }
        }
    }

    private void btn_calibrate_Click(object sender, EventArgs e)
    {
        string strD = txtb_D.Text;
        string strKnownHeight = txtb_KnownHeight.Text;
        if (!string.IsNullOrEmpty(strD) &&
!string.IsNullOrEmpty(strKnownHeight))
        {
            double dbD = double.Parse(strD);
            double dbKH = double.Parse(strKnownHeight);

            double ratio = dbKH / dbD;

            txtb_Ratio.Text = string.Format("{0:N6}", ratio);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น } อกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

private void btn_TargetP1_Click(object sender, EventArgs e)
{
    txtb_TargetP1.Text = Clipboard.GetText();
}

private void btn_TargetPoint2_Click(object sender, EventArgs e)
{
    txtb_TargetP2.Text = Clipboard.GetText();
}

private void btn_CalcTargD_Click(object sender, EventArgs e)
{
    string strP1 = txtb_TargetP1.Text;
    string strP2 = txtb_TargetP2.Text;

    if (!string.IsNullOrEmpty(strP1) && !string.IsNullOrEmpty(strP2))
    {
        try
        {
            string[] arrP1 = strP1.Split(',');
            string[] arrP2 = strP2.Split(',');

            double x1 = double.Parse(arrP1[0]);
            double y1 = double.Parse(arrP1[1]);
            double x2 = double.Parse(arrP2[0]);
            double y2 = double.Parse(arrP2[1]);

            double d = Math.Sqrt((Math.Pow((x2 - x1), 2) + Math.Pow((y2 -
y1), 2)));

            txtb_TargetD.Text = string.Format("{0:N2}", d);
        }
        catch (Exception ex)
        {
        }
    }
}

private void btn_CalcTarg_Click(object sender, EventArgs e)
{
    string strRatio = txtb_Ratio.Text;
    string strTGD = txtb_TargetD.Text;
    if (!string.IsNullOrEmpty(strRatio) && !string.IsNullOrEmpty(strTGD))
    {
        try
        {
            double dbRatio = double.Parse(strRatio);
            double dbTGD = double.Parse(strTGD);

            double tgh = dbRatio * dbTGD;
            txtb_TargetHeight.Text = string.Format("{0:N4}", tgh);
        }
        catch (Exception ex)
        {
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        catch (Exception)
        {
        }
    }
}

private void lstBx_listFile_SelectedIndexChanged(object sender, EventArgs
e)
{
    this.SetImage(FilePath + "\\\" + lstBx_listFile.SelectedItem);
}
private void SetImage(string filePath)
{
    try
    {
        picb_MainPic.Image = Image.FromFile(filePath);
    }
    catch (FileNotFoundException)
    {
        picb_MainPic.Image =
Image.FromFile(AppDomain.CurrentDomain.BaseDirectory + "\\fnf.jpg");
    }
}
private void btn_Browse_Click(object sender, EventArgs e)
{
    if (folderBrowserDialog1.ShowDialog() == DialogResult.OK)
    {
        lstBx_listFile.Items.Clear();
        string path = FilePath = folderBrowserDialog1.SelectedPath;
        //this.label_saveDirectory.Text = path;

        string[] filePicVid = getFiles(path, "*.jpg|*.jpeg|*.png",
SearchOption.AllDirectories);

        if (filePicVid.Count() > 0)
        {
            lstBx_listFile.Items.AddRange(filePicVid);
        }
    }
}
public string[] getFiles(string SourceFolder, string Filter,
System.IO.SearchOption searchOption)
{
    // ArrayList will hold all file names
    ArrayList alFiles = new ArrayList();

    // Create an array of filter string
    string[] MultipleFilters = Filter.Split('|');
    // for each filter find mathing file names
    foreach (string FileFilter in MultipleFilters)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น ยกเว้นผู้ที่ได้รับอนุญาตเป็นลายลักษณ์อักษร และขอสงวนสิทธิ์ในเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        // add found file names to array list
        string[] fileWPath = Directory.GetFiles(SourceFolder, FileFilter,
searchOption);
        foreach (string file in fileWPath)
        {
            alFiles.Add(Path.GetFileName(file));
        }
    }

    // returns string array of relevant file names
    return (string[])alFiles.ToArray(typeof(string));
}

private void EventSubtractTextBoxesChange(object sender, EventArgs e)
{
    string strP1 = txtb_P1.Text;
    string strP2 = txtb_P2.Text;

    if (!string.IsNullOrEmpty(strP1) && !string.IsNullOrEmpty(strP2))
    {
        try
        {
            string[] arrP1 = strP1.Split(',');
            string[] arrP2 = strP2.Split(',');

            double x1 = double.Parse(arrP1[0]);
            double y1 = double.Parse(arrP1[1]);
            double x2 = double.Parse(arrP2[0]);
            double y2 = double.Parse(arrP2[1]);

            double d = Math.Sqrt((Math.Pow((x2 - x1), 2) + Math.Pow((y2 -
y1), 2)));

            txtb_D.Text = string.Format("{0:N2}", d);
        }
        catch (Exception ex)
        {
        }
    }
}

private void EventRatioTextBoxChange(object sender, EventArgs e)
{
    string strTGD = txtb_D.Text;
    string strRatio = txtb_KnownHeight.Text;
    if (!string.IsNullOrEmpty(strRatio) && !string.IsNullOrEmpty(strTGD))
    {
        try
        {

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ © 2015 โดย บริษัท เทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        double tgh = dbRatio / dbTGD;

        txtb_Ratio.Text = string.Format("{0:N4}", tgh);
    }
    catch (Exception ex)
    {
    }
}

private void EventTargetSubtractTextBoxesChange(object sender, EventArgs
e)
{
    string strP1 = txtb_TargetP1.Text;
    string strP2 = txtb_TargetP2.Text;

    if (!string.IsNullOrEmpty(strP1) && !string.IsNullOrEmpty(strP2))
    {
        try
        {
            string[] arrP1 = strP1.Split(',');
            string[] arrP2 = strP2.Split(',');

            double x1 = double.Parse(arrP1[0]);
            double y1 = double.Parse(arrP1[1]);
            double x2 = double.Parse(arrP2[0]);
            double y2 = double.Parse(arrP2[1]);

            double d = Math.Sqrt((Math.Pow((x2 - x1), 2) + Math.Pow((y2 -
y1), 2)));

            txtb_TargetD.Text = string.Format("{0:N2}", d);
        }
        catch (Exception ex)
        {
        }
    }
}

private void EventTargetRatioTextBoxChange(object sender, EventArgs e)
{
    string strRatio = txtb_Ratio.Text;
    string strTGD = txtb_TargetD.Text;
    string strD = txtb_D.Text;
    string strKnownHeight = txtb_KnownHeight.Text;
    try
    {
        if (!string.IsNullOrEmpty(strRatio) &&
!string.IsNullOrEmpty(strTGD))
        {
            double dbRatio = double.Parse(strRatio);
            double dbTGD = double.Parse(strTGD);
            double dbD = double.Parse(strD);
            double dbKH = double.Parse(strKnownHeight);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <Stepper.h>

const int steps = 1000;

Stepper stepper(steps, 8, 9, 10, 11);

Stepper stepper1(steps, 2, 3, 4, 5);

int stepDirection = 3;

void setup() {

  stepper.setSpeed(30);

  Serial3.begin(9600);

}

void loop()

{

  int character;

  Serial3.println("Please input");

  Serial.flush();

  while(Serial3.available()>0){

    char character = Serial3.read();

    if(character == 'd')

    {

      stepper.step(stepDirection);

      delay(1);

      Serial3.flush();


```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if(character == 'a')
{
    stepper.step(-stepDirection);
    delay(1);
    Serial3.flush();
}
else if(character == 'w')
{
    stepper1.step(stepDirection);
    delay(1);
    Serial3.flush();
}
else if(character == 's')
{
    stepper1.step(-stepDirection);
    delay(1);
    Serial3.flush();
}
}
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. XBee®/XBee-PRO® RF Modules

The XBee and XBee-PRO RF Modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices.

The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other.



Key Features

Long Range Data Integrity

XBee

- Indoor/Urban: up to 100' (30 m)
- Outdoor line-of-sight: up to 300' (90 m)
- Transmit Power: 1 mW (0 dBm)
- Receiver Sensitivity: -92 dBm

XBee-PRO

- Indoor/Urban: up to 300' (90 m), 200' (60 m) for International variant
- Outdoor line-of-sight: up to 1 mile (1600 m), 2500' (750 m) for International variant
- Transmit Power: 63mW (18dBm), 10mW (10dBm) for International variant
- Receiver Sensitivity: -100 dBm

RF Data Rate: 250,000 bps

Advanced Networking & Security

Retries and Acknowledgements
DSSS (Direct Sequence Spread Spectrum)
Each direct sequence channels has over 65,000 unique network addresses available
Source/Destination Addressing
Unicast & Broadcast Communications
Point-to-point, point-to-multipoint and peer-to-peer topologies supported

Low Power

XBee

- TX Peak Current: 45 mA (@3.3 V)
- RX Current: 50 mA (@3.3 V)
- Power-down Current: < 10 μ A

XBee-PRO

- TX Peak Current: 250mA (150mA for international variant)
- TX Peak Current (RPSMA module only): 340mA (180mA for international variant)
- RX Current: 55 mA (@3.3 V)
- Power-down Current: < 10 μ A

ADC and I/O line support

Analog-to-digital conversion, Digital I/O
I/O Line Passing

Easy-to-Use

No configuration necessary for out-of box RF communications
Free X-CTU Software (Testing and configuration software)
AT and API Command Modes for configuring module parameters
Extensive command set
Small form factor

Worldwide Acceptance

FCC Approval (USA) Refer to Appendix A [p64] for FCC Requirements. Systems that contain XBee®/XBee-PRO® RF Modules inherit Digi Certifications.

ISM (Industrial, Scientific & Medical) 2.4 GHz frequency band

Manufactured under **ISO 9001:2000** registered standards

XBee®/XBee-PRO® RF Modules are optimized for use in the United States, Canada, Australia, Japan, and Europe. Contact Digi for complete list of government agency approvals.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Specifications

Table 1-01. Specifications of the XBee®/XBee-PRO® RF Modules

Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 μ A	< 10 μ A
General		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses
Agency Approvals		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	R201WW07215214	R201WW08215111 (Max. 10 dBm transmit power output)*
Australia	C-Tick	C-Tick

* See Appendix A for region-specific certification requirements.

Antenna Options: The ranges specified are typical when using the integrated Whip (1.5 dBi) and Dipole (2.1 dBi) antennas. The Chip antenna option provides advantages in its form factor; however, it typically yields shorter range than the Whip and Dipole antenna options when transmitting outdoors. For more information, refer to the "XBee Antennas" Knowledgebase Article located on Digi's Support Web site

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ขออนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Signals

Figure 1-03. XBee®/XBee-PRO® RF Module Pin Numbers

(top sides shown - shields on bottom)

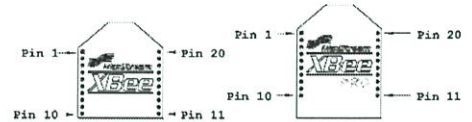


Table 1-02. Pin Assignments for the XBee and XBee-PRO Modules
(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	<u>RESET</u>	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	<u>CTS</u> / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	<u>RTS</u> / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

* Function is not supported at the time of this release

Design Notes:

- Minimum connections: VCC, GND, DOUT & DIN
- Minimum connections for updating firmware: VCC, GND, DIN, DOUT, RTS & DTR
- Signal Direction is specified with respect to the module
- Module includes a 50k Ω pull-up resistor attached to RESET
- Several of the input pull-ups can be configured using the PR command
- Unused pins should be left disconnected

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

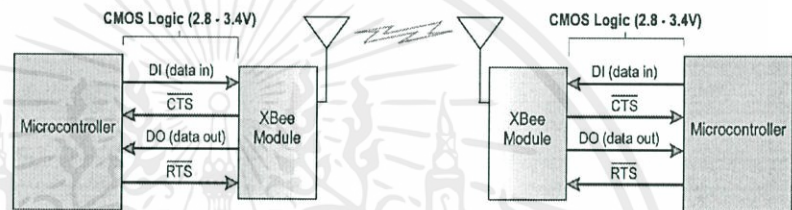
Serial Communications

The XBee®/XBee-PRO® RF Modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device (For example: Through a Digi proprietary RS-232 or USB interface board).

UART Data Flow

Devices that have a UART interface can connect directly to the pins of the RF module as shown in the figure below.

Figure 2-01. System Data Flow Diagram in a UART-interfaced environment
(Low-asserted signals distinguished with horizontal line over signal name.)

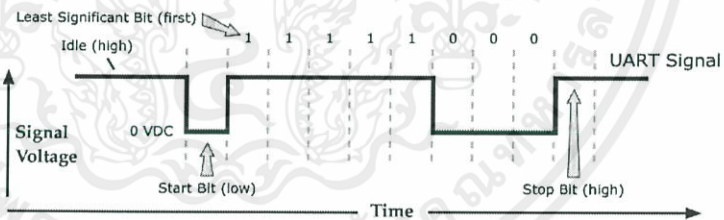


Serial Data

Data enters the module UART through the DI pin (pin 3) as an asynchronous serial signal. The signal should idle high when no data is being transmitted.

Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following figure illustrates the serial bit pattern of data passing through the module.

Figure 2-02. UART data packet 0x1F (decimal number "31") as transmitted through the RF module
Example Data Format is 8-N-1 (bits - parity - # of stop bits)



Serial communications depend on the two UARTs (the microcontroller's and the RF module's) to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).

The UART baud rate and parity settings on the XBee module can be configured with the BD and SB commands, respectively. See the command table in Chapter 3 for details.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Transparent Operation

By default, XBee®/XBee-PRO® RF Modules operate in Transparent Mode. When operating in this mode, the modules act as a serial line replacement - all UART data received through the DI pin is queued up for RF transmission. When RF data is received, the data is sent out the DO pin.

Serial-to-RF Packetization

Data is buffered in the DI buffer until one of the following causes the data to be packetized and transmitted:

1. No serial characters are received for the amount of time determined by the RO (Packetization Timeout) parameter. If RO = 0, packetization begins when a character is received.
2. The maximum number of characters that will fit in an RF packet (100) is received.
3. The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the DI buffer before the sequence is transmitted.

If the module cannot immediately transmit (for instance, if it is already receiving RF data), the serial data is stored in the DI Buffer. The data is packetized and sent at any RO timeout or when 100 bytes (maximum packet size) are received.

If the DI buffer becomes full, hardware or software flow control must be implemented in order to prevent overflow (loss of data between the host and module).

API Operation

API (Application Programming Interface) Operation is an alternative to the default Transparent Operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module.

When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module.

Transmit Data Frames (received through the DI pin (pin 3)) include:

- RF Transmit Data Frame
- Command Frame (equivalent to AT commands)

Receive Data Frames (sent out the DO pin (pin 2)) include:

- RF-received data frame
- Command response
- Event notifications such as reset, associate, disassociate, etc.

The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and payload information instead of using command mode to modify addresses. The module will send data frames to the application containing status packets; as well as source, RSSI and payload information from received data packets.

The API operation option facilitates many operations such as the examples cited below:

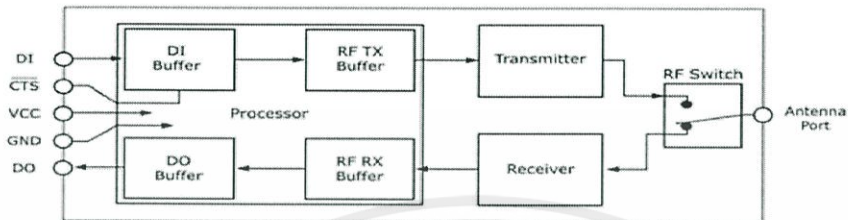
- > Transmitting data to multiple destinations without entering Command Mode
- > Receive success/failure status of each transmitted RF packet
- > Identify the source address of each received packet

To implement API operations, refer to API sections [p57].

เอกสารนี้เป็นเอกสาร
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow Control

Figure 2-03. Internal Data Flow Diagram



DI (Data In) Buffer

When serial data enters the RF module through the DI pin (pin 3), the data is stored in the DI Buffer until it can be processed.

Hardware Flow Control (CTS). When the DI buffer is 17 bytes away from being full; by default, the module de-asserts $\overline{\text{CTS}}$ (high) to signal to the host device to stop sending data [refer to D7 (DIO7 Configuration) parameter]. $\overline{\text{CTS}}$ is re-asserted after the DI Buffer has 34 bytes of memory available.

How to eliminate the need for flow control:

1. Send messages that are smaller than the DI buffer size (202 bytes).
2. Interface at a lower baud rate [BD (Interface Data Rate) parameter] than the throughput data rate.

Case in which the DI Buffer may become full and possibly overflow:

If the module is receiving a continuous stream of RF data, any serial data that arrives on the DI pin is placed in the DI Buffer. The data in the DI buffer will be transmitted over-the-air when the module is no longer receiving RF data in the network.

Refer to the RO (Packetization Timeout), BD (Interface Data Rate) and D7 (DIO7 Configuration) command descriptions for more information.

DO (Data Out) Buffer

When RF data is received, the data enters the DO buffer and is sent out the serial port to a host device. Once the DO Buffer reaches capacity, any additional incoming RF data is lost.

Hardware Flow Control (RTS). If RTS is enabled for flow control (D6 (DIO6 Configuration) Parameter = 1), data will not be sent out the DO Buffer as long as RTS (pin 16) is de-asserted.

Two cases in which the DO Buffer may become full and possibly overflow:

1. If the RF data rate is set higher than the interface data rate of the module, the module will receive data from the transmitting module faster than it can send the data to the host.
2. If the host does not allow the module to transmit data out from the DO buffer because of being held off by hardware or software flow control.

Refer to the D6 (DIO6 Configuration) command description for more information.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ULN2001A-ULN2002A ULN2003A-ULN2004A

SEVEN DARLINGTON ARRAYS

- SEVEN DARLINGTONS PER PACKAGE
- OUTPUT CURRENT 500mA PER DRIVER (600mA PEAK)
- OUTPUT VOLTAGE 50V
- INTEGRATED SUPPRESSION DIODES FOR INDUCTIVE LOADS
- OUTPUTS CAN BE PARALLELED FOR HIGHER CURRENT
- TTL/CMOS/PMOS/DTL COMPATIBLE INPUTS
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY LAYOUT

DESCRIPTION

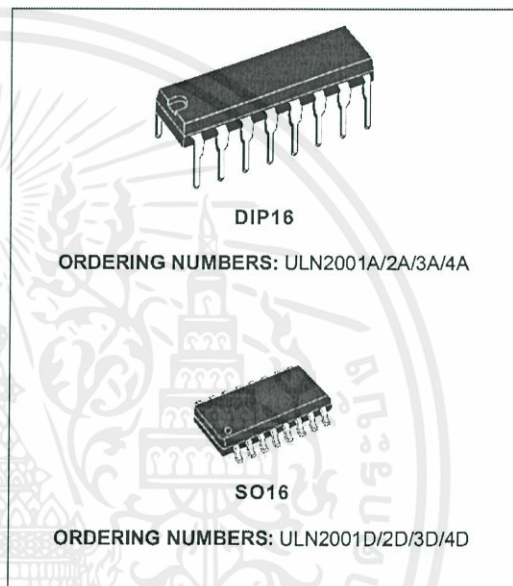
The ULN2001A, ULN2002A, ULN2003 and ULN2004A are high voltage, high current darlington arrays each containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout.

The four versions interface to all common logic families :

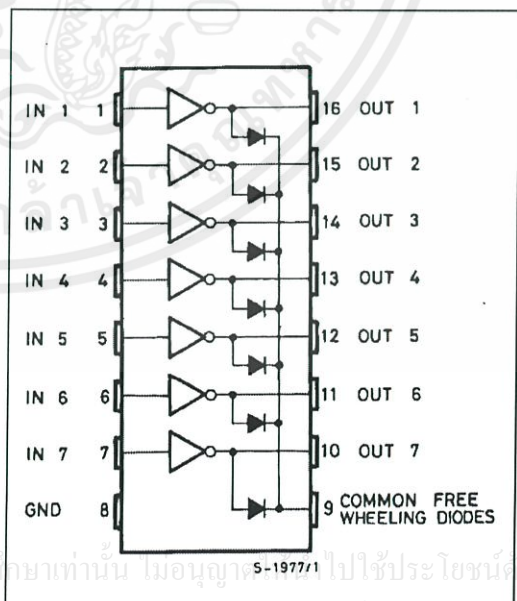
ULN2001A	General Purpose, DTL, TTL, PMOS, CMOS
ULN2002A	14-25V PMOS
ULN2003A	5V TTL, CMOS
ULN2004A	6-15V CMOS, PMOS

These versatile devices are useful for driving a wide range of loads including solenoids, relays DC motors, LED displays filament lamps, thermal print-heads and high power buffers.

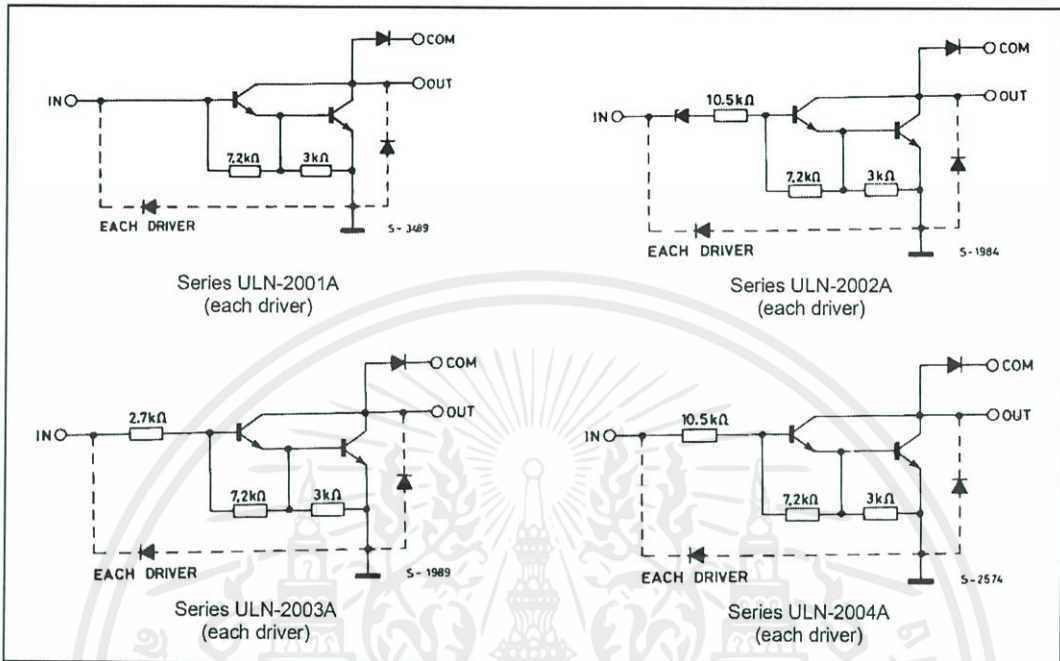
The ULN2001A/2002A/2003A and 2004A are supplied in 16 pin plastic DIP packages with a copper leadframe to reduce thermal resistance. They are available also in small outline package (SO-16) as ULN2001D/2002D/2003D/2004D.



PIN CONNECTION



SCHEMATIC DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_o	Output Voltage	50	V
V_{in}	Input Voltage (for ULN2002A/D - 2003A/D - 2004A/D)	30	V
I_c	Continuous Collector Current	500	mA
I_b	Continuous Base Current	25	mA
T_{amb}	Operating Ambient Temperature Range	- 20 to 85	°C
T_{stg}	Storage Temperature Range	- 55 to 150	°C
T_j	Junction Temperature	150	°C

THERMAL DATA

Symbol	Parameter	Max.	DIP16	SO16	Unit
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient		70	120	°C/W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS ($T_{amb} = 25^{\circ}\text{C}$ unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
I_{CEX}	Output Leakage Current	$V_{CE} = 50\text{V}$ $T_{amb} = 70^{\circ}\text{C}, V_{CE} = 50\text{V}$			50 100	μA μA	1a 1a
		$T_{amb} = 70^{\circ}\text{C}$ for ULN2002A $V_{CE} = 50\text{V}, V_i = 6\text{V}$ for ULN2004A $V_{CE} = 50\text{V}, V_i = 1\text{V}$			500 500	μA μA	1b 1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	$I_C = 100\text{mA}, I_B = 250\mu\text{A}$		0.9	1.1	V	2
		$I_C = 200\text{mA}, I_B = 350\mu\text{A}$		1.1	1.3	V	2
		$I_C = 350\text{mA}, I_B = 500\mu\text{A}$		1.3	1.6	V	2
$I_{i(on)}$	Input Current	for ULN2002A, $V_i = 17\text{V}$		0.82	1.25	mA	3
		for ULN2003A, $V_i = 3.85\text{V}$		0.93	1.35	mA	3
		for ULN2004A, $V_i = 5\text{V}$		0.35	0.5	mA	3
		$V_i = 12\text{V}$		1	1.45	mA	3
$I_{i(off)}$	Input Current	$T_{amb} = 70^{\circ}\text{C}, I_C = 500\mu\text{A}$	50	65		μA	4
$V_{i(on)}$	Input Voltage	$V_{CE} = 2\text{V}$ for ULN2002A $I_C = 300\text{mA}$			13	V	5
		for ULN2003A $I_C = 200\text{mA}$			2.4		
		$I_C = 250\text{mA}$			2.7		
		$I_C = 300\text{mA}$			3		
		for ULN2004A $I_C = 125\text{mA}$			5		
		$I_C = 200\text{mA}$			6		
		$I_C = 275\text{mA}$ $I_C = 350\text{mA}$			7 8		
h_{FE}	DC Forward Current Gain	for ULN2001A $V_{CE} = 2\text{V}, I_C = 350\text{mA}$	1000				2
C_i	Input Capacitance			15	25	pF	
t_{PLH}	Turn-on Delay Time	$0.5 V_i$ to $0.5 V_o$		0.25	1	μs	
t_{PHL}	Turn-off Delay Time	$0.5 V_i$ to $0.5 V_o$		0.25	1	μs	
I_R	Clamp Diode Leakage Current	$V_R = 50\text{V}$ $T_{amb} = 70^{\circ}\text{C}, V_R = 50\text{V}$			50 100	μA μA	6 6
		$I_F = 350\text{mA}$		1.7	2	V	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEST CIRCUITS

Figure 1a.

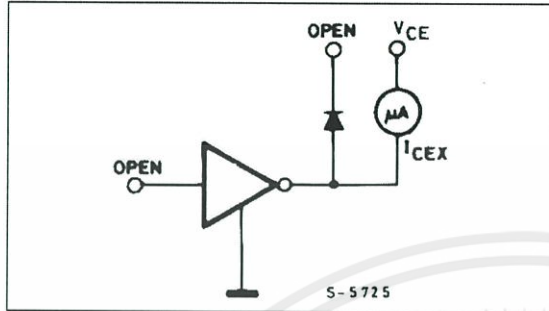


Figure 1b.

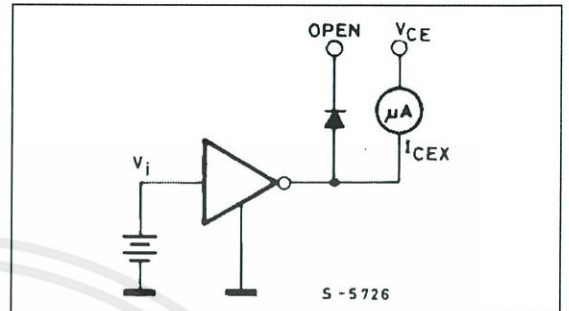


Figure 2.

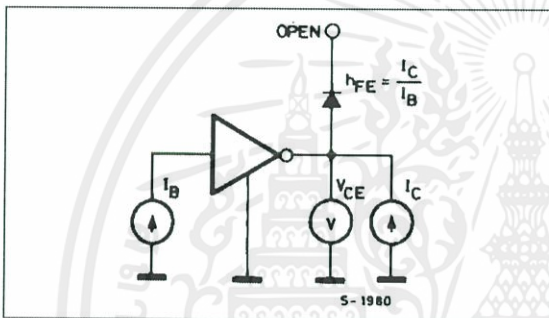


Figure 3.

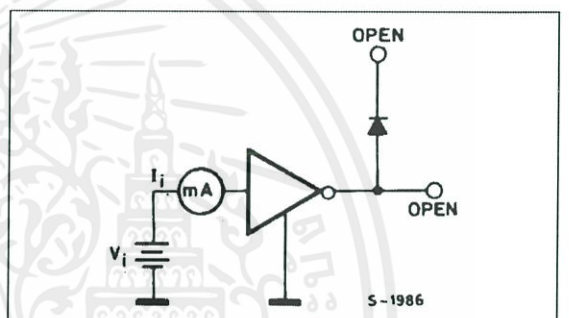


Figure 4.

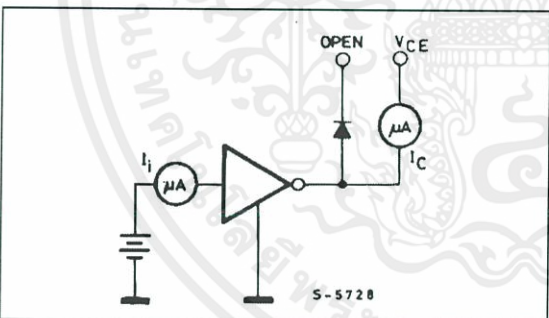


Figure 5.

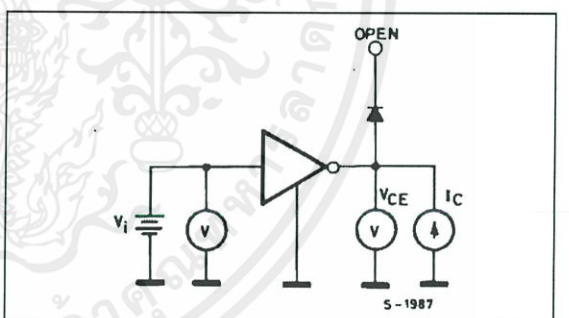


Figure 6.

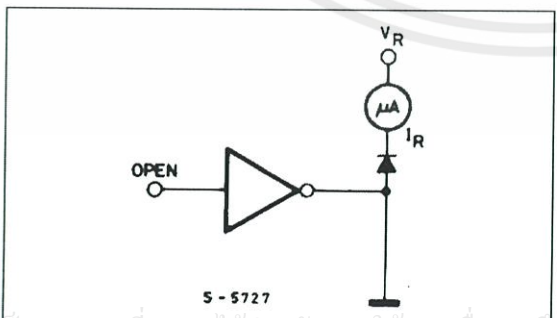
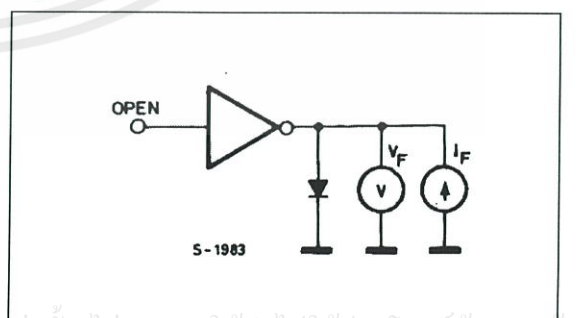


Figure 7.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 8: Collector Current versus Input Current

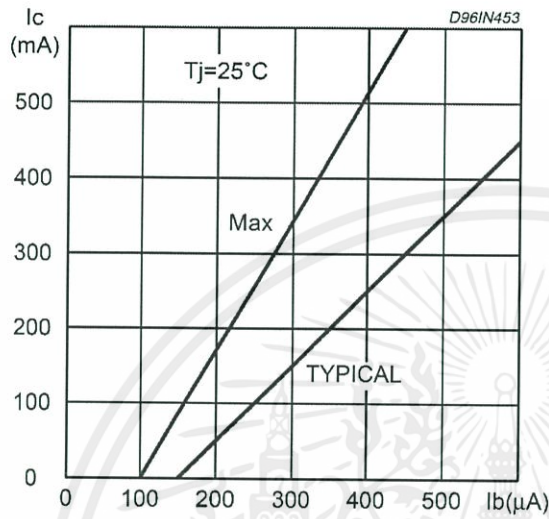


Figure 9: Collector Current versus Saturation Voltage

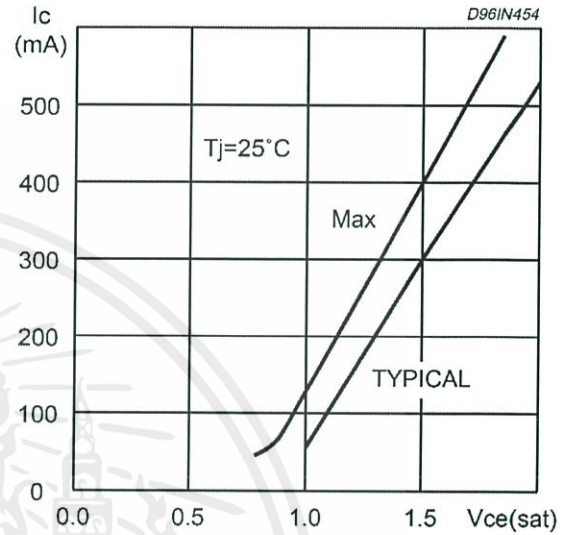


Figure 10: Peak Collector Current versus Duty Cycle

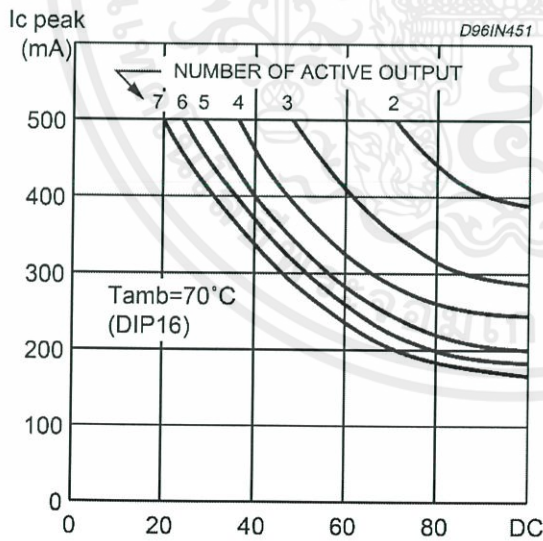
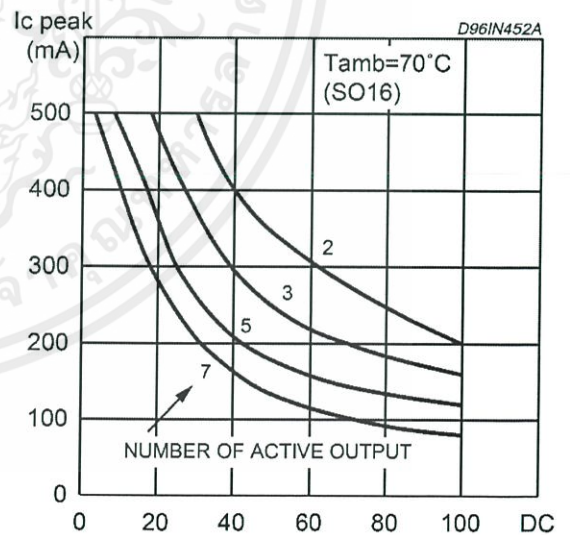


Figure 11: Peak Collector Current versus Duty Cycle



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้