แว่นตาสำหรับผู้ป่วยอัมพาต

Eye Glasses for Paralytic Person

นพมงคล เชิดชัย

Nopmonkol Choetchai

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2555

แว่นตาสั่งงานสำหรับผู้เป็นอัมพาต

Eye Glasses for Paralytic Person

นพมงคล เฉิดฉาย
Nopmonkol Choetchai

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แว่นตาสั่งงานสำหรับผู้เป็นอัมพาต

Eye Glasses for Paralytic Person

โดย
นพมงคล เฉิดฉาย

อาจารย์ที่ปรึกษา
ผศ.ดร. กิติพล ชิตสกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2555
สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์
คณะ วิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง แว่นตาสั่งงานสำหรับผู้เป็นอัมพาต
Eye Glasses for Paralytic Person
ผู้จัดทำ นาย นพมงคล เฉิดฉาย รหัส 52010560

รายงานนี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ.....
(ผศ.ดร. กิติพล ชิตสกุล)
อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ การพัฒนาแว่นตาสั่งงานสำหรับผู้เป็นอัมพาต
นักศึกษา นาย นพมงคล เฉิดฉาย รหัสประจำตัว 52010560
ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา 2555
อาจารย์ที่ปรึกษาปริญญานิพนธ์ ผศ.ดร. กิติพล ชิตสกุล

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการพัฒนาแว่นตาสั่งงานสำหรับผู้ป่วยอัมพาต โดยปรับปรุงส่วนแว่นตาที่ใช้ส่งงานและส่วนฉากรับแสงเลเซอร์ ระบบที่ใช้ในการประมวลผลเป็นโทรศัพท์มือถือ Android โดยใช้กล้องของโทรศัพท์ทำการรับข้อมูลจากฉากและนำไปสั่งงานที่ ฮาร์ดแวร์โดยผ่านการสื่อสารไร้สายรูปแบบ บลูทูธ เพื่อนำไปส่งการเปิด-ปิดเครื่องใช้ไฟฟ้าต่างๆ และสั่งงานให้โทรศัพท์ในการเปิดเพลงและส่งข้อความสั้น เพื่อต้องการให้ผู้ป่วยสามารถช่วยเหลือตัวเองได้บางส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Development of Eye Glasses for Paralytic Person
Student	Mr. Nopmonkol Choetchai Student ID 52010560
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2012
Thesis Advisor	Asst. Prof. Dr. Kitiphol Chitsakul

Abstract

This thesis describes design of the eye glasses for paralytic person. By attaching a laser pointer on an eyeglass's fram, the user just commands to turn on the laser by exhaling to a microphone attached at the nasal path and points to a screen. An android phone is used as reciver the images, by built in camera, of the screen and defined the position of laser point (on the screen). The laser pointer just turns on for 7 second for economizing the power supply but sufficient for processing by the phone. The commands generated by the phone depending on the positions of laser point on the screen. Generating music, sending a short message are the current examples. The commands for turning on and of independently four electrical appliances through wireless Bluetooth which work with a specially designed control box have been included.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ ผศ.ดร.กิตติพล ชิตสกุล ที่ให้คำแนะนำเริ่มต้นเกี่ยวกับการทำแว่นตาสั่งงานเพื่อผู้ป่วยอัมพาต รวมถึงการแก้ปัญหาต่างๆ

ผู้จัดทำจึงขอกราบขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

นพมงคล เฉิดฉาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I

Abstract II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาของโครงงาน 1

1.2 ลักษณะของโครงงาน 2

1.3 วัตถุประสงค์ของโครงงาน 2

1.4 ขอบเขตของงานที่ทำ 3

1.5 โครงสร้างของรายงาน 3

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง 4

2.1 อาการอัมพาต 4

2.1.1 อัมพาตครึ่งซีก (Hemiplegia) 4

2.1.2 อัมพาตครึ่งท่อน (Paraplegia) 5

2.1.3 อัมพาตที่ใบหน้า 5

2.2 Android 6

2.2.1 วิวัฒนาการของแอนดรอยด์ 6

2.2.2 โครงสร้างของระบบปฏิบัติการแอนดรอยด์ 7

2.2.3 สถาปัตยกรรมของแอนดรอยด์ 8

2.3 การเขียนโปรแกรมเพื่อพัฒนาแอพพลิเคชั่นบนแอนดรอยด์ 9

2.4 OpenCV 12

2.5 Image Processing 12

2.5.1 แบบจำลองสี (Color Model) 12

2.5.1.1 สเปซ RGB (RGB Space) 13

2.5.1.2 สเปซ NTSC 13

2.5.1.3 สเปซ YCbCr 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.5.1.4 สเปซ HSI (Hue Saturation Intensity) 14
2.5.1.5 สเปซ HSV (Hue, Saturation, Value Space) 14
2.5.2 ฮิสโตแกรม (Histrogram) 15
2.5.3 การแปลงทางเรขาคณิตของภาพ (Geometrical Transformation of Image) 16
2.5.3.1 การเคลื่อนย้ายภาพ (Translation) 16
2.5.3.2 การสเกลภาพ (Scaling) 16
2.5.3.3 การหมุนภาพ (Rotation) 17
2.5.3.4 การเฉือนภาพ (Shearing) 17
2.5.4 การปรับปรุงภาพ (Image Enhancement) 18
2.5.4.1 การขลิบและการทำเทรสโชว์ (Clipping and Thresholding) 18
2.5.4.2 การฟิลเตอร์ภาพ (Image Filtering) 18
2.5.4.2.1 ฟิลเตอร์แบบค่าเฉลี่ย (Averaging) 19
2.5.4.2.2 ฟิลเตอร์แบบค่ามีเดียน (Median Filtering) 19
2.6 การสื่อสารแบบอนุกรม 20
2.6.1 การรับส่งแบบซิงโครนัส (Synchronous) 20
2.6.2 การรับส่งข้อมูลแบบอะซิงโครนัส (Asynchronous) 20
2.6.3 มาตรฐานRS232 21
2.7 เลเซอร์ไดโอด (Laser Diode) 22
2.7.1 กลไกการเปล่งแสงของเลเซอร์ไดโอด 23
2.7.2 ลักษณะสมบัติกำลังเอาท์พุทและกระแสไฟฟ้า 24
2.8 ไมโครคอนโทเลอร์ 26
2.9 Bluetooth 27
2.9.1 ระยะเชื่อมต่อของบลูทูธ 27
2.9.2 มาตรฐานของบลูทูธ 28
2.10 งานวิจัยที่เกี่ยวข้อง 29
บทที่ 3 การออกแบบและการทำงานของโปรแกรม 30
3.1 ฉากรับภาพ 30
3.2 การติดตั้งกล้อง 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.3 การทำงานของโปรแกรม 32
3.4 การเปลี่ยนสเปซสีจาก RGB เป็น Gray scale 35
3.5 การทำเทรสโชว์ภาพ 35
3.6 การเลือกย่านที่ใช้ในการคำนวณ 37
3.7 การออกแบบส่วนควบคุมเครื่องใช้ไฟฟ้า 37
3.8 การออกแบบส่วนแว่นตา 38
บทที่ 4 การทดลองและผลการทดลอง 40
4.1 อุปกรณ์การทดลอง 40
4.2 ขั้นตอนการทดลอง 40
4.2.1 การทดลองหาขนาดตารางในการรับค่า 40
4.3 ผลการทดลอง 41
บทที่ 5 บทสรุป 44
บรรณานุกรม 45
ภาคผนวก 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตาราง 2.1 รุ่นการพัฒนาของแอนดรอยด์ 7
ตาราง 2.2 D Type 9 Pin and D Type 25 Pin Connectors หน้าที่ของแต่ละขา 21
ตาราง 4.1 ผลทำการทดลอง 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 1.1 แสดงส่วนแว่นตาสั่งงาน 1
รูปที่ 1.2 แสดงโครงสร้างโดยรวมของระบบ 2
รูปที่ 1.3 แสดงแนวคิดของอุปกรณ์แว่นตาสั่งงานสำหรับผู้เป็นอัมพาต 3
รูปที่ 2.1 ประเภทของอาการอัมพาต 5
รูปที่ 2.2 สถาปัตยกรรมของแอนดรอยด์ 8
รูปที่ 2.3 การคอมไพล์ 10
รูปที่ 2.4 วงจรการทำงาน (Lifecycle) 11
รูปที่ 2.5 ลูกบาศ์ก RGB 13
รูปที่ 2.6 ลูกบาศ์ก YCbCr 13
รูปที่ 2.7 แบบจำลอง HSV 14
รูปที่ 2.8 กราฟฮิสโทแกรมที่แสดงถึงแสงที่สมดุล 15
รูปที่ 2.9 แสดงฮิสโตแกรมแบบต่างๆ 15
รูปที่ 2.10 (a) ภาพที่ยังไม่ได้ทำเทรสโชว์ (b) ภาพที่ทำการเทรสโชว์แล้วที่ค่าระดับกั้นคือ 128 18
รูปที่ 2.11 เฟรมการส่งข้อมูลแบบอะซิงโครนัส 20
รูปที่ 2.12 DB25 connector และ DB9 connector 21
รูปที่ 2.13 ตัวอย่างเลเซอร์ไดโอดที่มีขายในท้องตลาด 23
รูปที่ 2.14 รอยต่อพีเอ็นที่กำเนิดแสงเลเซอร์ในเลเซอร์ไดโอด 23
รูปที่ 2.15 ลักษณะความสัมพันธ์ของความเข้มแสงและกระแสไฟฟ้าในเลเซอร์ไดโอด 24
รูปที่ 2.16 ลักษณะสมบัติกระแสไฟฟ้าและแรงดันของเลเซอร์ไดโอด 25
รูปที่ 2.17 ลักษณะสมบัติกำลังของเลเซอร์เอาท์พุทและกระแสที่ฉีดเข้าเลเซอร์ไดโอด 25
รูปที่ 2.18 การเปลี่ยนแปลงของกระแสไฟฟ้าขีดเริ่มเปลี่ยนอันเนื่องจากอุณหภูมิของเลเซอร์ไดโอด 26
รูปที่ 2.19 รูปแบบการสื่อสาร 27
รูปที่ 2.20 โครงสร้างชุดข้อมูลของบลูทูธ 28
รูปที่ 3.1 ฉากรับภาพ 30
รูปที่ 3.2 การติดตั้งกล้องให้สามารถรับภาพได้พอดีกับฉากรับภาพ 31
รูปที่ 3.3 แสดง Flow Chart การทำงานของโปรแกรมการประมวลผลภาพ 32
รูปที่ 3.4 แสดง Flow Chart การทำงานของโปรแกรมการประมวลผลภาพ(ต่อ) 33
รูปที่ 3.5 แสดงภาพที่ได้จากกล้องในสเปซ RGB 35
รูปที่ 3.6 แสดงภาพที่อยู่ในสเปซสีของระดับเทา 35
รูปที่ 3.7 ภาพที่ยังไม่ได้ทำเทรสโชว์ 36
รูปที่ 3.8 ภาพหลังจากทำเทรสโชว์ 36
รูปที่ 3.9 ฮิสโตแกรมของภาพที่ยังไม่ได้ทำเทรสโชว์ 36
รูปที่ 3.10 แสดงขอบเขต ที่ใช้ในการคำนวณ 37
รูปที่ 3.11 วงจรส่วนรับข้อมูลและควบคุมการเปิดปิดเครื่องใช้ไฟฟ้า 38
รูปที่ 3.12 เป็นการติดตั้งเลเซอร์ไดโอดบริเวณขาของแว่นตา 38
รูปที่ 3.13 เป็นการติดตั้งเลเซอร์ไดโอดบริเวณขาของแว่นตา 39

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่ 3.14 วงจรขยายวงจรขยาย กรองสัญญาณจากไมโครโฟน วงจรตั้งแวลาและขับเลเซอร์ไดโอด 39
รูปที่ 3.15 ต้นแบบแว่นตาและ กล่องวงจรขับเลเซอร์ 39
รูปที่ 4.1 แสดงการจัดวางอุปกรณ์ 40
รูปที่ 4.2 เลเซอร์ไดโอด 41
รูปที่ 4.3 ฉากรับ 41
รูปที่ 4.4 ขณะทำการทดลอง 41
รูปที่ 4.5 กราฟแสดงผลการทดลอง 43

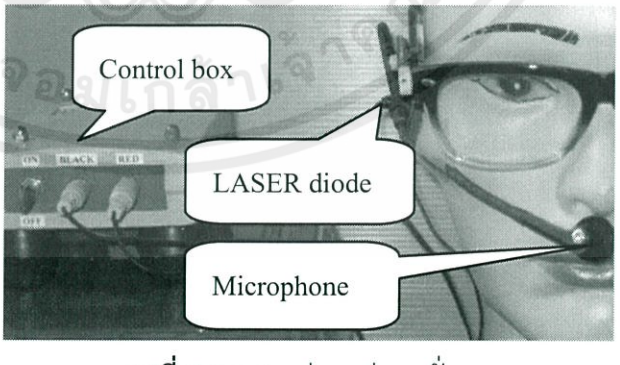
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงาน

แนวโน้มการเพิ่มขึ้นของจำนวนผู้สูงอายุที่มีมาอย่างต่อเนื่อง สำหรับประเทศไทย จากข้อมูลของรายงานสถานการณ์ผู้สูงอายุเมื่อปีพ.ศ. 2551 นั้น อีกห้าปีข้างหน้า ผู้สูงอายุในประเทศไทยจะมีมากถึงเก้าล้านคน นอกจากนี้จำนวนผู้พิการในประเทศไทยมีแนวโน้มเพิ่มมากขึ้น จากผลสำรวจคนพิการ ของสำนักงานสถิติแห่งชาติเมื่อ ปี พ.ศ. 2550 นั้น ประเทศไทยมีผู้พิการประมาณ 1.9 ล้านคน หรือคิดเป็นร้อยละ 2.9 ของประชาการทั้งหมด สำหรับผู้พิการด้านร่างกาย ได้แก่ แขนขาอ่อนแรง อัมพาต มีประมาณ 3.5 แสนคน โดยปัญหาแขนขาอ่อนแรง ไม่สามารถควบคุมการทำงานของแขนขาได้จะพบมากในผู้สูงอายุและผู้พิการด้านแขนขา

อัมพาตเป็นอาการผู้ป่วยจะไม่สามารถขยับส่วนต่างๆของร่างกายได้ตามปกติไม่ว่าจะเป็นอัมพาตครึ่งซีก อัมพาตครึ่งท่อน หรืออัมพาตทั้งตัว ในกรณีอัมพาตทั้งตัว ผู้ป่วยไม่สามารถสั่งงานควบคุมอวัยวะตั้งแต่ไหล่ลงมาได้แต่ยังสามารถหายใจ กรอกตา และขยับหมุนคอได้บ้าง จึงทำให้ไม่สามารถช่วยเหลือตัวเองได้ ต้องนั่งหรือนอนอยู่บนเตียงผู้ป่วยเท่านั้นและต้องมีผู้คอยดูแลช่วยเหลืออยู่ตลอด เวลาดังนั้น เพื่อให้ผู้ป่วยกลุ่มนี้ สามารถพึ่งพาตนเองได้ระดับหนึ่ง จึงได้คิดค้นสร้างเครื่องมือที่ใช้หลักการทางอิเล็กทรอนิกส์ เพื่อช่วยเหลือให้ผู้ที่เป็นอัมพาตสามารถสั่งงานได้ด้วยตนเอง แว่นตาสั่งงานสำหรับผู้เป็นอัมพาต ที่นำเสนอนี้ ใช้หลักการยิงแสงเลเซอร์ไปที่ฉาก ซึ่งกำหนดโซนที่จุดที่ลำแสงตกกระทบ เพื่อกำหนดเป็นรหัสสั่งงานที่แตกต่างกัน เลเซอร์ไดโอดจะถูกติดตั้งบนแว่นตาที่ผู้ใช้งานสวมใส่ ใช้ลมหายใจแรงๆ ทางจมูกสั่งให้เลเซอร์เปล่งแสงชั่วขณะผ่านทางไมโครโฟน เมื่อเลเซอร์เปล่งแสง ผู้ใช้งานเพียงควบคุมให้แสงเลเซอร์ ตกกระทบฉากในโซนที่ต้องการ ในรูปที่ 1.1 แสดงอุปกรณ์ในส่วนนี้ โครงงานนี้เป็นส่วนประมวลผลภาพที่รับมาจากเว็บแคม ซึ่งรับภาพจากฉากนำมาคำนวณตำแหน่งจากการประมวลผลภาพ และแปลงเป็นรหัสควบคุมการเคลื่อนที่รถเข็น ในรูปที่ 1.2 แสดงโครงสร้างของระบบ



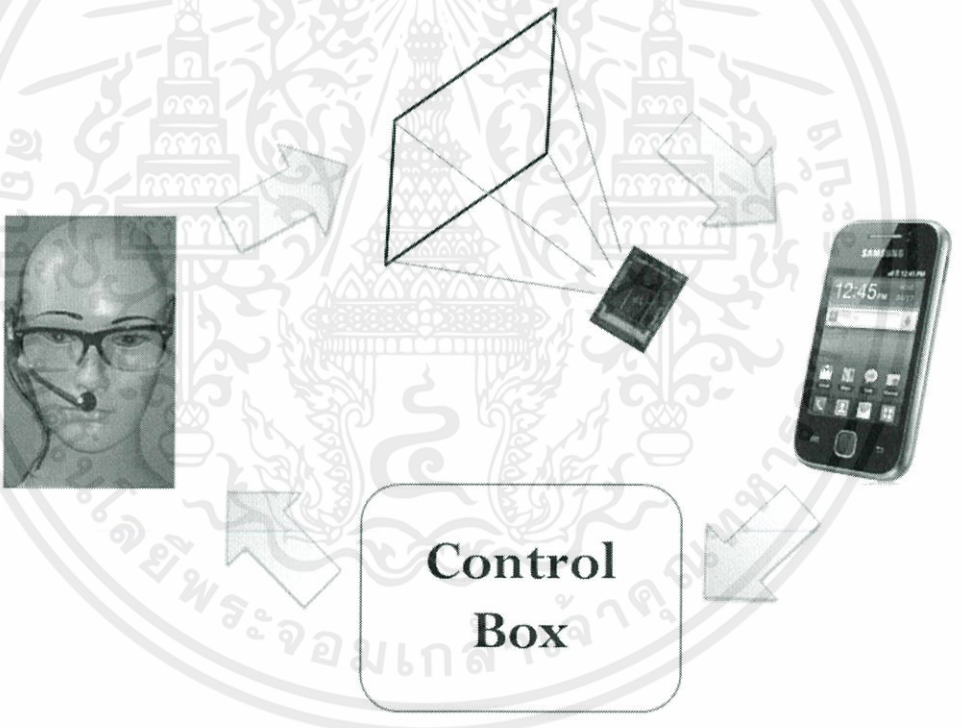
รูปที่ 1.1 แสดงส่วนแว่นตาสั่งงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ลักษณะของโครงงาน

แว่นตาสั่งงานสำหรับผู้เป็นอัมพาตทั้งระบบ จะต้องเปิดให้ทำงานตลอดเวลา เพื่อรองรับคำสั่งที่ผู้ป่วยจะสั่งเข้ามาแว่นตาสั่งงานสำหรับผู้เป็นอัมพาตนี้จะแบ่งการทำงานออกเป็น 3 ส่วน ได้แก่

1. ภาคส่ง จะเป็นแว่นตาที่ติดแหล่งกำเนิดแสงเลเซอร์และไมโครโฟน โดยจะให้ผู้ป่วยสวมใส่แว่นตานี้ไว้ตลอดเวลา เมื่อจะทำคำสั่งที่ต้องการ จะให้ผู้ป่วยเป่าลมเข้าไปในไมโครโฟน โดยสัญญาณเสียงที่ได้จะผ่านเข้าไปในกล่องควบคุมเลเซอร์ ซึ่งเป็นผลทำให้แหล่งกำเนิดแสงเลเซอร์ที่ติดอยู่บนแว่นตานั้นเปล่งแสงออกมา เพื่อส่งแสงเลเซอร์ไปยังภาครับต่อไป
2. ภาครับจะมีฉากรับแบ่งคำสั่งต่างๆ จากสัญลักษณ์ ออกเป็นช่องๆ ผู้ป่วยสามารถเลือกคำสั่งที่ต้องการได้ด้วยการยิงแสงเลเซอร์ไปยังช่องคำสั่งที่ต้องการ เพื่อให้ภาครับทำการประมวลผลและส่งสัญญาณข้อมูลไปยังภาคควบคุม
3. ภาคควบคุม มีหน้าที่ทำคำสั่งตามที่ผู้ป่วยได้เลือกโดยรับสัญญาณจากภาครับผ่านทางพอร์ต บลูทูธ (Bluetooth) ไปยังไมโครคอนโทรลเลอร์ซึ่งทำหน้าที่ควบคุมการทำงานของอุปกรณ์ภายนอก

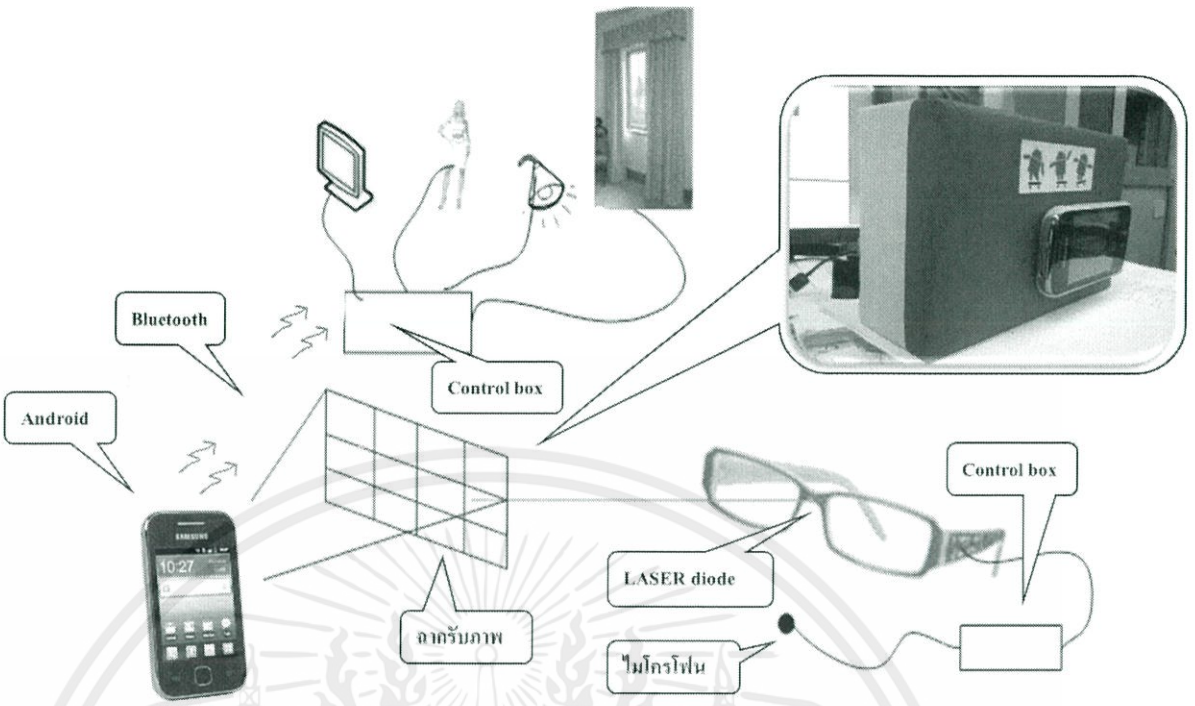


รูปที่ 1.2 แสดงโครงสร้างโดยรวมของระบบ

1.3 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาการประมวลผลภาพบนอุปกรณ์พกพา (mobile phone)
2. สามารถควบคุมการทำงานทาง Hardware ที่สอดคล้องกับการประมวลผลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.3 แสดงแนวคิดของอุปกรณ์แว่นตาสั่งงานสำหรับผู้เป็นอัมพาต

1.4 ขอบเขตของงานที่ทำ

เขียนโปรแกรมประมวลผลภาพบนอุปกรณ์พกพา (Mobile phone) เพื่อระบุตำแหน่งของจุดแสงเลเซอร์บนฉากเพื่อสร้างและส่งรหัสคำสั่งไปยังภาคควบคุมต่อไปทางพอร์ตบลูทูธ (Bluetooth)

1.5 โครงสร้างของรายงาน

รายงานนี้เป็นการรายงานผลจากการศึกษาและทดลองเพื่อออกแบบสร้างต้นแบบแว่นตาสั่งงานสำหรับผู้เป็นอัมพาต ซึ่งเนื้อหานั้นจะประกอบด้วยส่วนต่าง ๆ แยกเป็นบท ๆ ไปดังนี้

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและแนวคิดในการสร้างโครงงาน
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้องของโครงงาน
บทที่ 3 รายละเอียดโปรแกรมที่ใช้ในโครงงาน
บทที่ 4 การทดลองและผลการทดลอง
บทที่ 5 บทสรุป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 อาการอัมพาต

อัมพาต เป็นอาการที่เกิดจากการที่สมองขาดออกซิเจนอันเนื่องมาจากเส้นเลือดไปเลี้ยงสมองเกิดการอุดตันหรือแตก ทำให้การทำงานของสมองส่วนต่างๆ ผิดปกติ

สมองเป็นอวัยวะที่ควบคุมหน้าที่ต่างๆของร่างกายหากสมองตายไปเนื่องจากขาดเลือดไปเลี้ยง ผู้ป่วยอาจจะมีปัญหาเกี่ยวกับการพูดพฤติกรรม และความจำ นอกจากนี้ยังทำให้อวัยวะต่างๆของร่างกายในส่วนที่สมองส่วนนั้นๆควบคุมอ่อนแรง ความผิดปกติของร่างกายที่เกิดขึ้นจะเป็นมากหรือน้อยขึ้นกับว่าสมองส่วนใดขาดเลือดไปเลี้ยง และขาดเลือดไปเลี้ยงมากน้อยขนาดใดตัวอย่างเช่น หากสมองด้านหลังขาดเลือดไปเลี้ยงผู้ป่วยจะมีปัญหาเกี่ยวกับการมองหรือเห็นภาพและสมองมีหลายส่วนแต่ละส่วนมีหน้าที่แตกต่างกันไป ดังนั้น อาการในผู้ป่วยแต่ละคนขึ้นกับความผิดปกติของหลอดเลือดในสมองที่มีเส้นเลือดตีบหรือตันอาการที่พบได้แก่

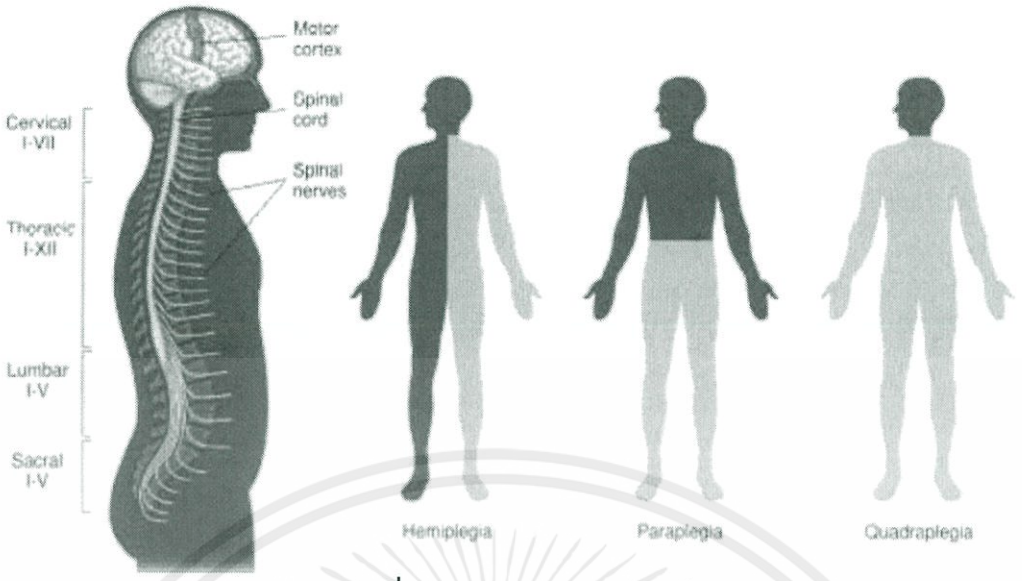
1. แขนขาอ่อนแรงหรือชาซีกใดซีกหนึ่ง (บางกรณีอาจเป็น ทั้ง2ซีก) ขยับตัวไม่ได้ซีกใดซีกหนึ่ง
2. ลิ้นแข็ง พูดไม่ชัด หรือสำลัก
3. พูดไม่ได้ หรือฟังไม่รู้เรื่อง (มีปัญหาด้านความเข้าใจภาษา)
4. เวียนศีรษะมาก เดินเซแบบที่ไม่เคยเป็นมาก่อน
5. มองไม่เห็นซีกใดซีกหนึ่ง

โดยทั่วไปอาการอัมพาตสามารถแบ่งออกเป็นอัมพาตครึ่งซีก (Hemiplegia) อัมพาตครึ่งท่อน (paraplegia) อัมพาตทั้งตัว (quadraplegia) และยังมีอัมพาตใบหน้าซึ่งเป็นอีกหนึ่งในอาการอ่อนแรงของร่างกาย

2.1.1 อัมพาตครึ่งซีก (Hemiplegia)

อาการอัมพาตครึ่งซีกหมายถึงการที่แขนด้านใดด้านหนึ่งของร่างกายเกิดหย่อนสมรรถภาพไม่สามารถทำงานได้สาเหตุเนื่องมาจากสมองส่วนที่ควบคุมด้านนั้นเกิดการผิดปกติหรือพิการโดยมีสาเหตุมาจากโรคหลอดโลหิตในสมองอุดตันกระโหลกศีรษะถูกกระทบกระเทือนอย่างแรงโรคติดเชื้อมะเร็งหรือเนื้องอกในสมอง โรคความดันโลหิตสูงโรคเบาหวานซึ่งโรคเหล่านี้เป็นต้นเหตุให้สมองถูกทำลายโดยตรงหรืออาจเกิดแบบชั่วคราวผู้มีอาการอัมพาตครึ่งซีก จะเริ่มมีอาการปวดศีรษะมากเวียนหัว บางครั้งปวดศีรษะข้างเดียวปวดเสียวแขนขา อ่อนเพลียทั้งแขนขา บางรายมีอาการลิ้นแข็งพูดไม่ชัดหรือพูดไม่ได้เลยระบบขับถ่ายควบคุมไม่ได้ เมื่อมีอาการต่าง ๆ เหล่านี้กล้ามเนื้อจะเกิดการเกร็งตัวทำให้ขยับแขนขาได้แต่มีลักษณะเกร็งไม่ใช้เกิดจากกลไกการทำงานของกล้ามเนื้อไหล่จะหุบไปด้านหลัง แขนจะหนีบชิดลำตัว ข้อศอกงอขึ้นกระดูกเกร็งผิดปกติบางรายมีอาการเกร็งเฉพาะที่แล้วแต่อาการที่เป็นกระดูกข้อมือ นิ้วมือจะเกร็งเข้าหากัน ตะโพกเหยียดไปข้างหลัง ขาแบะออกข้อเข่าจะเหยียดตึงซึ่งเรียกว่าความพิการของอัมพาตครึ่งซีกในระยะหดเกร็งการบำบัดรักษาการบริหารข้อต่อ จะช่วยให้มีการเคลื่อนไหวและช่วยให้โลหิตดีขึ้น รวมทั้งอาจใช้การนวดร่วมด้วยเพื่อให้กล้ามเนื้อได้ถูกกระตุ้นและทำให้กล้ามเนื้อที่อ่อนแอนั้นเพิ่มความแข็งแรงขึ้นมาได้บ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ประเภทของอาการอัมพาต

2.1.2 อัมพาตครึ่งท่อน (Paraplegia)

อัมพาตครึ่งท่อน หมายถึงการที่กล้ามเนื้อขาทั้งสองไม่ทำงานและความรู้สึกของขาก็จะลดลงด้วย แต่ในขณะเดียวกันร่างกายท่อนบนก็ยังคงแข็งแรงและสามารถใช้แขนทั้งสองเข็นรถไปได้ (ดังจะเห็นได้ชัดเจนในส่วนของการเล่นกีฬาคนพิการที่มีการแข่งขันกีฬาประเภทวอลเลย์บอล, บาสเกตบอล ฯลฯ) อัมพาตครึ่งท่อนนี้เป็นการอ่อนแรงหรือไม่มีแรงเพียงร่างกายครึ่งท่อนล่างเท่านั้น (อัมพาตครึ่งท่อนบนไม่มี)ฉะนั้นขาทั้งสองข้างหรืออาจจะรวมถึงบริเวณลำตัวด้วยก็ได้ที่จะอ่อนแรงไปสาเหตุที่เป็นเนื่องมาจากอุบัติเหตุที่เกิดกับกระดูกสันหลัง, โรคติดเชื้อ, เนื้องอก เป็นต้นสาเหตุเหล่านี้จะไปทำให้ไขสันหลังถูกรบกวนหรือถูกทำลายส่วนอาการจะมากหรือน้อยนั้น ก็ขึ้นอยู่กับตำแหน่งที่เกิดโรคถ้าเกิดที่กระดูกสันหลังส่วนบนมากเท่าไหร่อาการก็จะมากตามไปด้วยนอกจากกล้ามเนื้อขาทั้งสองจะไม่ทำงานแล้วความรู้สึกที่ขาก็จะลดลงด้วย ระบบขับถ่ายก็จะเสียไปอาการในระยะแรกที่เห็นเด่นชัดคือกล้ามเนื้อจะอ่อนปวกเปียกไม่มีความตึงตัวเลยระยะต่อมาก็จะเป็นระยะเกร็งตัวซึ่งถ้าไม่ระมัดระวังอย่างถูกวิธีก็จะทำให้เกิดความพิการขึ้นได้การบำบัดรักษาการรักษาทางกายภาพบำบัดจะช่วยฝึกกล้ามเนื้อส่วนบนคือ แขน ขาและลำตัวให้แข็งแรงเพื่อทดแทนส่วนล่างที่เสียไปส่วนท่อนล่างที่เสียไปก็ต้องเคลื่อนไหวข้อต่อไว้และพยายามกระตุ้นกล้ามเนื้อให้คืนมาเท่าที่จะทำได้กายภาพบำบัดจะช่วยให้มีการไหลเวียนของโลหิตดีขึ้นและช่วยให้กล้ามเนื้อมีการฟื้นตัวขึ้นได้

2.1.3 อัมพาตที่ใบหน้า

อัมพาตที่ใบหน้า จะทำให้กล้ามเนื้อใบหน้าซีกหนึ่งไม่ทำงานและมีอาการปากเบี้ยว, ตาหลับไม่ลง, เลิกหน้าผากไม่ขึ้น ฯลฯสาเหตุของการเกิดอัมพาตนี้ยังไม่เป็นที่แน่ชัดอาจเกิดจาก เชื้อโรค, อุบัติเหตุ หรือเส้นประสาทคู่ที่ 7 ที่มาเลี้ยงใบหน้าด้านนั้น ถูก รบกวนหรือเกิดการอักเสบขึ้น ทำให้เกิดอาการได้นอกจากกล้ามเนื้อจะไม่ทำงานแล้วการไหลเวียนของโลหิตบริเวณใบหน้าก็จะลดน้อยลงไปด้วยซึ่งในส่วนนี้การทำกายภาพบำบัดจะช่วยเพิ่มการไหลเวียนของโลหิตได้โดยวิธีการนวดหน้าและอาจจะกระตุ้นกล้ามเนื้อที่เสียไปด้วยกระแสไฟฟ้าเพื่อให้กล้ามเนื้อนั้นมีการหดตัวและกลับมาทำงานได้เหมือนเดิมการบำบัดรักษา การบริหารกล้ามเนื้อใบหน้า เช่นเลิกหน้าผากขึ้น, หลับตาปี๋, ยิ้มกว้างๆ,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห่อปากจู๋ ฯลฯเพื่อให้กล้ามเนื้อทุกมัดบนใบหน้าได้ทำงานการบริหารให้บ่อยครั้งที่นึกได้ ก็จะช่วยได้มากเพราะจะทำให้กล้ามเนื้อได้รับการกระตุ้นบ่อยๆ อัมพาตของหน้านี้มีโอกาสที่จะฟื้นคืนสภาพได้มากถ้าทำการรักษาโดยการบริหารซึ่งหากปล่อยทิ้งไว้นานจนกล้ามเนื้อเกิดการตึงติดหรือเสียสภาพไปแล้วค่อยมารักษาก็จะไม่สามารถช่วยให้ดีขึ้นมาได้กายภาพบำบัดจะเข้ามาช่วยในเรื่องการฝึกกล้ามเนื้อที่เสียไปและกล้ามเนื้อที่ยังอยู่ดีเพื่อให้ผู้ป่วยช่วยตัวเองได้มากที่สุดแม้ผู้ป่วยบางรายอาจไม่ดีขึ้นมากนักก็ตามแต่ก็ช่วยให้เขาใช้ส่วนที่ยังดีมาช่วยส่วนที่เสียไปชีวิตของเขาจะมีค่าขึ้นและไม่เป็นภาระให้กับ ผู้อื่นมากนัก

2.2 Android

2.2.1 วิวัฒนาการของแอนดรอยด์

แอนดรอยด์ (Android) เป็นระบบปฏิบัติการสำหรับอุปกรณ์พกพาเช่น โทรศัพท์มือถือ แท็บเล็ตคอมพิวเตอร์ เน็ตบุ๊ก ซึ่งพัฒนามาจากระบบปฏิบัติการลินุกซ์ (Linux)

แรกเริ่มแอนดรอยด์ถูกพัฒนาโดยบริษัท Android Inc. ซึ่งก่อตั้งในปี ค.ศ.2003 โดย Andy Rubin และ Rich Miner ต่อมาในปี ค.ศ.2005 กูเกิล (Google Inc.) ได้เข้าซื้อบริษัทดังกล่าว หลังจากนั้นได้ร่วมมือกับกลุ่มบริษัททางด้านฮาร์ดแวร์ ซอฟแวร์ และการสื่อสาร เช่น Intel, HTC, LG, Texas Instruments เพื่อจัดตั้งองค์การความร่วมมือที่มีชื่อว่า Open Handset Alliance ขึ้นในปี ค.ศ.2007 โดยมีจุดประสงค์ในการสร้างแพลตฟอร์ม (Platform) สำหรับอุปกรณ์พกพาที่มีพื้นฐานอยู่บนมาตรฐานเปิด (Open Standard) ซึ่งโปรเจ็คแรกที่ได้เปิดตัวออกมานั้นคือ แอนดรอยด์ ซึ่งมีชื่อโปรเจ็คว่า The Android Open Source Project

นักพัฒนาทั่วไปสามารถพัฒนาแอพพลิเคชั่นที่รันบนแอนดรอยด์ได้โดยใช้ภาษาจาวา โดยการเข้าถึงความสามารถต่างๆของแอนดรอยด์ผ่านทาง Java Library ที่ทางกูเกิลได้จัดเตรียมไว้ให้ใน Android SDK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.1 รุ่นการพัฒนาของแอนดรอยด์

รุ่น	ชื่อเล่น	API level	ลินุกซ์ เคอร์เนล	เปิดตัว
1.0	Astro (อัสโทร)	1		23 พฤศจิกายน 2550
1.1	Bender (เบนเดอร์)	2		9 กุมภาพันธ์ 2552
1.5	Cupcake (คัพเค้ก)	3	2.6.27	30 เมษายน 2552
1.6	Donut (โดนัท)	4	2.6.29	15 สิงหาคม 2552
2.0/2.1	Éclair (เอแคลร์)	7	2.6.29	26 ตุลาคม 2552 (2.0) 12 มกราคม 2553 (2.1)
2.2	Froyo (โฟรซเซนโยเกิร์ต)	8	2.6.32	20 พฤษภาคม 2553
2.3.x	Gingerbread (ขนมปังขิง)	9-10	2.6.35	6 ธันวาคม 2553
3.x.x	Honeycomb (รังผึ้ง)	11-13	2.6.36	22 กุมภาพันธ์ 2554
4.0.x	Ice Cream Sandwich (แซนด์วิชไอศกรีม)	14-15		19 ตุลาคม 2554
4.1.x	Jelly Bean (เจลลีบีน)	16		28 มิถุนายน 2555

2.2.2 โครงสร้างของระบบปฏิบัติการแอนดรอยด์

โครงสร้างและระบบปฏิบัติการแอนดรอยด์ที่ติดตั้งอยู่ในอุปกรณ์ต่างๆ มีโครงสร้าง 6 ส่วน ดังนี้

-Bootloader เป็นส่วนที่ถูกเรียกใช้งานเมื่ออุปกรณ์เริ่มทำงาน

-Boot Image เป็นส่วนของเคอร์เนลและแรมดิสก์

-System Image เป็นส่วนที่จัดเก็บระบบปฏิบัติการแอนดรอยด์และแอพพลิเคชั่น

-Data Image เป็นส่วนที่จัดเก็บข้อมูลของระบบปฏิบัติการแอนดรอยด์และแอพพลิเคชั่น

-Recovery Image เป็นส่วนที่จะถูกเรียกใช้งานเมื่อมีการกู้หรืออัพเกรดระบบปฏิบัติการ

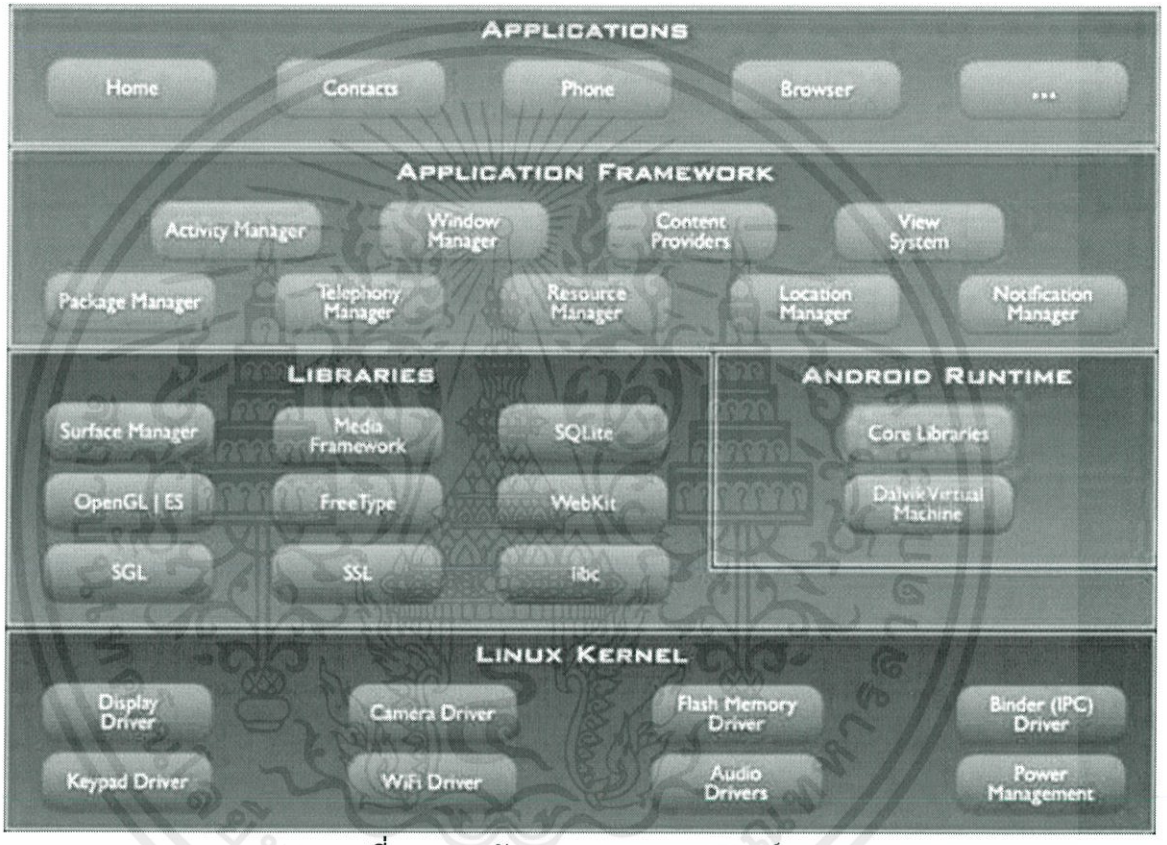
-Radio Image เป็นส่วนที่เก็บข้อมูลของเรดิโอสแต็ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบทั้ง 6 ส่วนนี้จะถูกจัดเก็บลงในหน่วยความจำแฟลช ซึ่งข้อมูลเหล่านี้จะไม่สูญหายแม้จะไม่มีกระแสไฟเลี้ยงระบบ ส่วนของหน่วยความจำแฟลชนี้บางครั้งจะทำงานเป็นหน่วยความจำที่ใช้อ่านเท่านั้น (ROM) แต่บางครั้งเราก็สามารถเขียนข้อมูลลงไปในหน่วยความจำนี้ได้

เมื่ออุปกรณ์เริ่มทำงาน หน่วยประมวลผลจะสั่งให้ Bootloader ทำการโหลดเคอร์เนลและแรมดิสก์เข้าสู่หน่วยความจำหลักเพื่อให้ทำงานได้รวดเร็ว หลังจากนั้นหน่วยประมวลผลจะทำงานจามชุดคำสั่งในเคอร์เนลตามลำดับ และสั่งงานให้ส่วนของ Radio Image เชื่อมต่อกับฮาร์ดแวร์

2.2.3 สถาปัตยกรรมของแอนดรอยด์



รูปที่ 2.2 สถาปัตยกรรมของแอนดรอยด์

-Applications เป็นส่วนบนสุดของสถาปัตยกรรมแอนดรอยด์ เป็นส่วนที่ติดตั้งแอพพลิเคชั่นทั้งในส่วนที่ติดตั้งมากับเครื่องและที่ติดตั้งเพิ่มเติมทีหลัง ซึ่งแอพพลิเคชั่นในส่วนนี้จะถูกพัฒนาด้วยภาษาจาวา

- Application Framework เป็นส่วนที่ใช้พัฒนาแอพพลิเคชั่นซึ่งประกอบด้วยคอมโพเนนต์พื้นฐานต่างๆที่ใช้ในการพัฒนาโดยคอมโพเนนต์เหล่านี้จะถูกติดตั้งมากับแอนดรอยด์อยู่แล้วและสามารถแทนที่คอมโพเนนต์เหล่านั้นได้

ส่วนสำคัญใน Application Framework มีดังนี้

-Activity Manager คือ คอมโพเนนต์ที่ควบคุม Lifecycle ของแอพพลิเคชั่น

-Content Providers คือ คอมโพเนนต์ที่ทำให้แอพพลิเคชั่นต่างๆสามารถแชร์ข้อมูลกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-View System คือ คอมโพเนนต์ที่ใช้สร้างส่วนติดต่อกับผู้ใช้

-Resource Manager คือ คอมโพเนนต์ที่ใช้จัดการข้อมูลต่างๆในแอพพลิเคชั่นที่ไม่ใช่โค้ด เช่น รูปภาพ ข้อความ เป็นต้น

-Notification Manager คือ คอมโพเนนต์ที่ทำให้แอพพลิเคชั่นสามารถแสดงข้อความแจ้งเตือนผู้ใช้ออกมาที่แถบสถานะได้

-Android Runtime เป็นส่วนที่ทำงานอยู่บนเคอร์เนลของลีนุกซ์ ซึ่งประกอบไปด้วย Core Library สำหรับภาษาจาวาและ Dalvik Virtual Machine ซึ่งถ้าอธิบายง่ายๆก็คือ Java Virtual Machine ในแบบของแอนดรอยด์ ที่ถูกออกแบบมาสำหรับอุปกรณ์ที่ใช้ไฟฟ้าจากแบตเตอรี และมีหน่วยความจำจำกัด

ในแอนดรอยด์แต่ละแอพพลิเคชั่นจะรันอยู่ในโพรเซลเซอร์ของตัวเอง และมี Dalvik Virtual Machine ของตัวเองซึ่งแยกจากกัน

สำหรับ Core Library ซึ่งเป็น Java Library นั้นส่วนใหญ่จะเหมือนกันใน Java Standard Edition (Java SE) ที่ใช้พัฒนาบนคอมพิวเตอร์ แต่บางไลบรารีที่มีใน Java SE จะไม่มีในแอนดรอยด์ และบางไลบรารีที่มีในแอนดรอยด์แต่ก็มีรูปแบบการใช้งานที่แตกต่างออกไป

-Libraries เป็นส่วนที่เก็บไลบรารีของแอนดรอยด์ ซึ่งทั้งหมดเขียนด้วยภาษา C หรือ C++ และถูกคอมไพล์มาสำหรับฮาร์ดแวร์ของอุปกรณ์แต่ละรุ่น

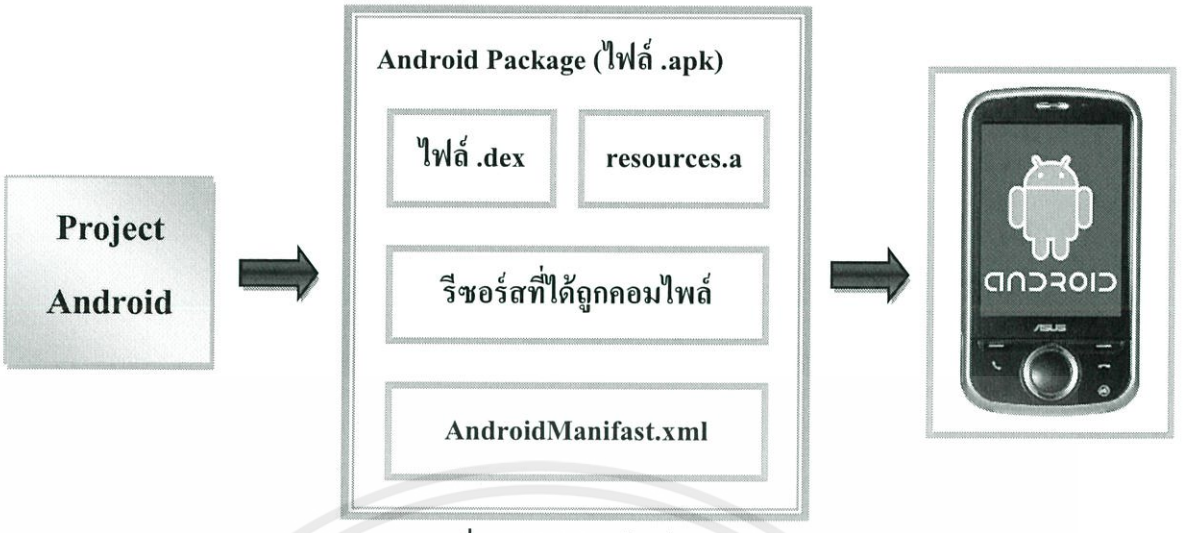
-Linux Kernal เป็นส่วนแกนหลักหรือ เคอร์เนล (Kernal) ของแอนดรอยด์ เคอร์เนลของลินุกซ์ ซึ่งเป็นระบบปฏิบัติการที่สร้างโดย Linus Torvalds ในปี ค.ศ.1991 ปัจจุบันเราสามารถพบได้ในทุกสิ่งทุกอย่าง

ในส่วนของ Linux Kernal ของแอนดรอยด์ จะทำหน้าที่เป็น Hardware Abstraction Layer กล่าวคือเป็นตัวกลางระหว่างฮาร์ดแวร์กับซอฟต์แวร์ และทำหน้าที่บริหารทรัพยากรต่างๆของเครื่อง ซึ่งผู้ผลิตอุปกรณ์สามารถ “พอร์ต” (Port) แอนดรอยด์ให้ไปรันบนฮาร์ดแวร์ชนิดใดก็ได้ โดยทำการเปลี่ยนแปลงในส่วนของ Linux Kernal นี้

2.3 การเขียนโปรแกรมเพื่อพัฒนาแอพพลิเคชั่นบนแอนดรอยด์

การพัฒนา Application บน Android จะใช้ภาษา JAVA ซึ่งในไฟล์ต่างๆในโปรเจ็คจะถูกคอมไพล์และบีบรวมไว้เป็นไฟล์เดียวที่เรียกว่า แพ็คเกจไฟล์ (Package File) ซึ่งมีนามสกุลไฟล์เป็น .apk (Android Package) ไฟล์นี้จะรวบรวมตัวแอพและข้อมูลทุกอย่างที่จำเป็นสำหรับการรันแอพเอาไว้เช่น ไฟล์ .dex ที่คอมไพล์แล้ว (แปลงมาจากไฟล์ .class เพื่อให้เป็น Bytecode สำหรับ Dalvik VM), ไฟล์ AndroidManifast.xml ในรูปแบบไบนารี, รีซอร์สที่ถูกคอมไพล์ (ไฟล์ resources.arsc) และรีซอร์สที่ได้ถูกคอมไพล์ เป็นต้น

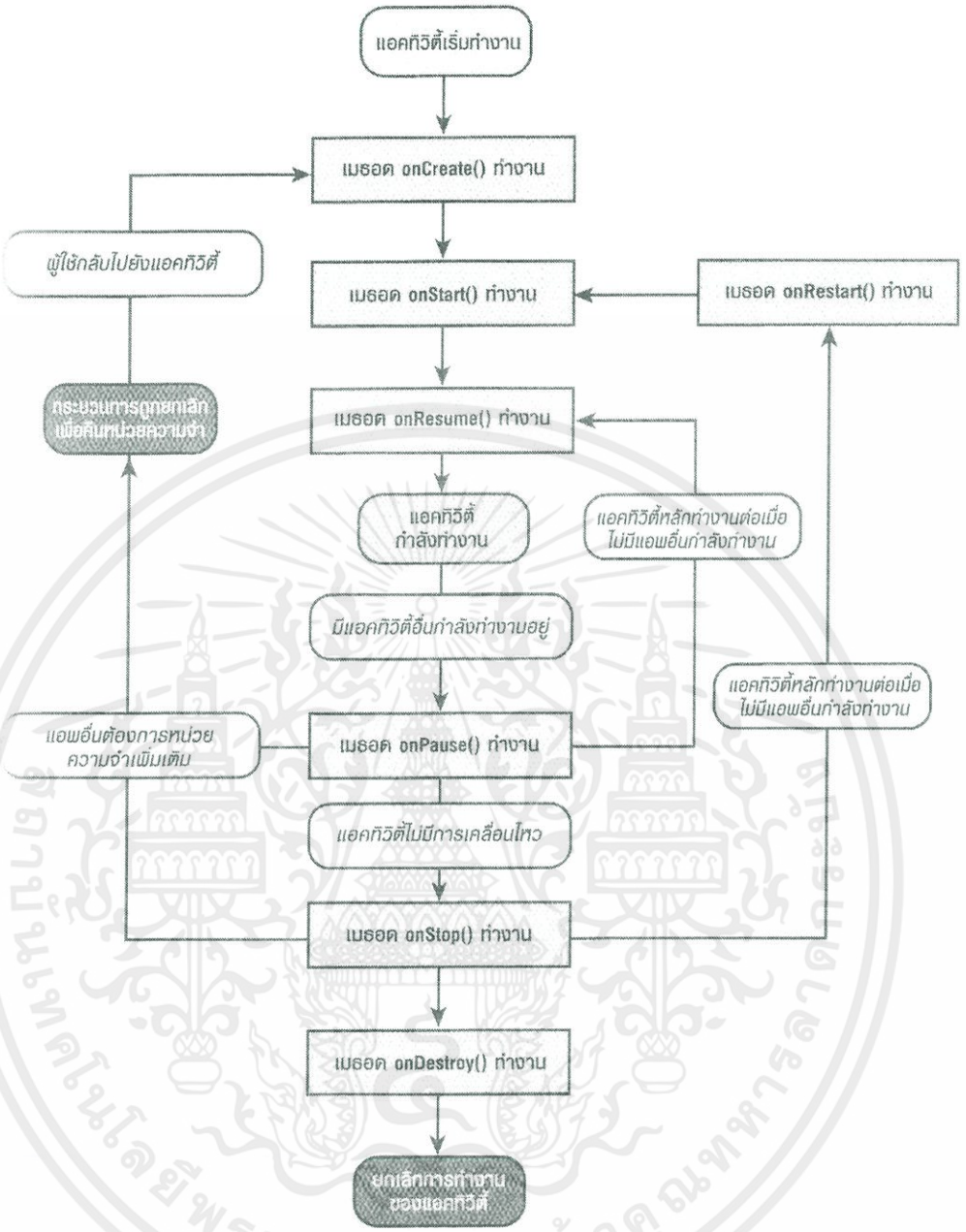
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 การคอมไพล์

ในแอนดรอยด์นั้นแอคทิวิตี(Activity) ต่างๆจะถูกจัดการในลักษณะของสแตค (Stack) กล่าวคือ แอคทิวิตีใหม่ที่เริ่มต้นขึ้นมาจะถูกวางไว้ที่ด้านบนสุดของสแตค และกลายเป็นแอคทิวิตีที่กำลังทำงาน (Running Activity) ส่วนแอคทิวิตีที่ทำงานมาก่อนจะอยู่ถัดลงไป และจะกลับมาอยู่ด้านบนตามเดิมเมื่อแอคทิวิตีใหม่จบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 วงจรการทำงาน (Lifecycle)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 OpenCV

OpenCV คือ การพัฒนาโปรแกรมทางด้าน การมองเห็นของคอมพิวเตอร์ (Computer Vision) คือสามารถประมวลผลภาพดิจิตอลได้ทั้งภาพนิ่ง และภาพเคลื่อนไหวเช่น ภาพจากกล้องVDO หรือ VDO File เป็นไปได้อย่างสะดวก มีฟังก์ชันสำเร็จรูปสำหรับจัดการข้อมูลภาพ และการประมวลผลภาพพื้นฐานเช่น การหาขอบภาพ การกรองข้อมูลภาพ

2.5 Image Processing

Image processing คือ การประมวลผลภาพ เป็นการประยุกต์ใช้งานการประมวลผลสัญญาณบนสัญญาณ 2 มิติ เช่น ภาพนิ่ง และยังรวมถึงสัญญาณวีดิโอ, ภาพวีดิทัศน์ หรือ ภาพ 2 มิติอื่น ๆ ที่ไม่ใช่ภาพด้วยแนวความคิดและเทคนิค ในการประมวลผลสัญญาณ สำหรับสัญญาณ 1 มิตินั้นสามารถปรับมาใช้กับภาพได้ไม่ยาก แต่นอกเหนือจาก เทคนิคจากการประมวลผลสัญญาณแล้ว การประมวลผลภาพก็มีเทคนิคและแนวความคิดที่เฉพาะ (เช่น connectivity และ rotation invariance) ซึ่งจะมีความหมายกับสัญญาณ 2 มิติเท่านั้น แต่อย่างไรก็ตามเทคนิคบางอย่าง จากการประมวลผลสัญญาณใน 1 มิติ จะค่อนข้างซับซ้อนเมื่อนำมาใช้กับ 2 มิติ เมื่อหลายสิบปีมาแล้ว การประมวลผลภาพนั้น จะอยู่ในรูปของการประมวลผลสัญญาณอนาลอก (analog) โดยใช้อุปกรณ์ปรับแต่งแสง (optics) ซึ่งวิธีเหล่านั้นก็ไม่ได้หายสาบสูญ หรือเลิกใช้ไป ยังมีใช้เป็นส่วนสำคัญ สำหรับการประยุกต์ใช้งานบางอย่าง เช่น ฮอโลกราฟี (holography) แต่เนื่องจากอุปกรณ์คอมพิวเตอร์ในปัจจุบัน ราคาถูกลง และเร็วขึ้นมาก การประมวลผลภาพดิจิตอล (digital image processing) จึงได้รับความนิยมมากกว่า เพราะการประมวลผลที่ทำได้ซับซ้อนขึ้น แม่นยำ และง่ายในการลงมือปฏิบัติ กระบวนการต่าง ๆ มีอยู่ด้วยกันหลายอย่าง ยกตัวอย่างเช่น

- การแปลงข้อมูลรูปภาพ (Image Transformation)
- การนิยามภาพ (Image Description)
- การกรองภาพ (Image Filters)
- การคืนภาพ (Image Restorlation)
- การปรับปรุงคุณภาพของภาพ (Image Enhancement)
- การแบ่งภาพและการหาขอบภาพในวัตถุ (Image Segmentation and EdgDeTection)
- การบีบอัดข้อมูลภาพ (Image Complession)

2.5.1 แบบจำลองสี (Color Model)

การศึกษาเกี่ยวกับสีเป็นสิ่งสำคัญในการออกแบบพัฒนาระบบการเห็นสี การใช้สีในภาคแสดงผลไม่เพียงแต่ทำให้เกิดความพอใจ ยังทำให้เราสามารถรับรู้ข้อมูลการมองเห็นอื่น ๆ อีก ตาของมนุษย์สามารถแยกความแตกต่างของระดับเทาได้ไม่กี่สิบระดับในขณะที่ตาสามารถแยกความแตกต่างของสีได้เป็นพันสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.1 สเปซ RGB (RGB Space)

การแทนสี (Color Representation) มักใช้ทฤษฎีของ Thomas Young (1802) ซึ่งกล่าวว่าสีใดก็ตามเกิดจากการผสมสีแม่สี 3 สี ซึ่งสอดคล้องกับเซลล์รับแสงชนิดโคนในตาของมนุษย์แบ่งออกเป็น 3 ชนิด ซึ่งแต่ละชนิดไวต่อแสงในย่านแดง เขียว และน้ำเงิน ทฤษฎีการแทนสีของ Young สามารถแทนด้วยแบบจำลองลูกบาศ์ก RGB ดังแสดงในรูปที่ 1 ค่าของสีแทนด้วยค่าพิกัดในสเปซ RGB โดยที่จุดยอดของลูกบาศ์กคือแม่สี (Red Green และ Blue) และสีของแสงขั้นที่สอง (Secondary) (Cyan, Margenta และ Yellow)

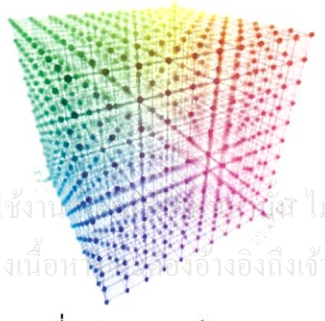
รูปที่ 2.5 ลูกบาศ์ก RGB

2.5.1.2 สเปซ NTSC

ระบบสเปซ NTSC ใช้ในโทรทัศน์ในประเทศสหรัฐอเมริกา ข้อได้เปรียบอันหนึ่งคือ ข้อมูลระดับเทา (Gray Level) ถูกแยกออกมาจากข้อมูลสี ดังนั้นสัญญาณหนึ่งสามารถใช้ได้ทั้งโทรทัศน์สีและขาวดำ ในรูปแบบ NTSC ข้อมูลภาพประกอบด้วยองค์ประกอบ 3 องค์ประกอบได้แก่ ลูมิแนนซ์ (Luminance) แทนด้วย Y , โทนสี (Hue) แทนด้วย I และแซ็ททูเรชัน (Saturation) แทนด้วย Q ดังนั้นระบบนี้อาจเรียกว่า YIQ องค์ประกอบลูมิแนนซ์แทนด้วยข้อมูลระดับเทา ส่วนอีก 2 องค์ประกอบแทนข้อมูลสีของสัญญาณโทรทัศน์

2.5.1.3 สเปซ YCbCr

สเปซ YCbCr ถูกใช้อย่างกว้างขวางในสัญญาณวีดีโอดิจิตอล ในรูปแบบนี้ข้อมูลลูมิแนนซ์ถูกแทนด้วยองค์ประกอบเดียว Y ในขณะที่ข้อมูลสีถูกเก็บในองค์ประกอบผลต่างของสีนั่นคือ Cb คือผลต่างระหว่างองค์ประกอบน้ำเงินกับค่าอ้างอิง และ Cr คือ ผลต่างระหว่างองค์ประกอบสีแดงกับค่าอ้างอิง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

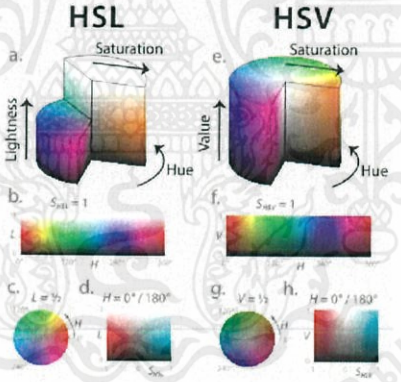
รูปที่ 2.6 ลูกบาศ์ก YCbCr

2.5.1.4 สเปซ HSI (Hue Saturation Intensity)

สเปซ HSI มีความใกล้เคียงกับระบบ RGB ในด้านการรับรู้สี (Color Sensation) คุณสมบัติเกี่ยวกับการรับรู้สีแบ่งเป็นค่าความสว่าง (Brightness), ค่าโทนสี (Hue) และค่าแซ็ททูเรชัน (Saturation) ค่าความสว่าง (Brightness) เป็นการรับรู้ค่าความส่องสว่าง (Luminance) ค่าโทนสีของสีหมายถึงความแดง (เช่นแดงอ่อน แดงแก่ แดงเลือดหมูเป็นต้น) ความเขียว (เขียวอ่อน เขียวแก่ เขียวขี้ม้า เป็นต้น) และอื่น ส่วนค่าแซ็ททูเรชันเกี่ยวข้องกับความแรงของแสงสีขาว (White)

2.5.1.5 สเปซ HSV (Hue, Saturation, Value Space)

HSV เป็นหนึ่งในระบบสีที่เปรียบเสมือนถูกใช้โดยคนซึ่งเลือกสีจากจานสี สเปซ HSV ถูกสร้างโดยการมองจากลูกบาศ์กสี RGB ตามแนวแกนเทา (Gray Axis) ซึ่งคือแกนที่เชื่อมระหว่างจุดมุมสีขาวและดำซึ่งทำให้ได้ลักษณะรูปหกเหลี่ยมดังแสดงในรูปที่ 2 เมื่อทำการเคลื่อนที่ตามแนวดิ่ง (แกนเทา) ตามไดอะแกรมในรูปที่ 2 ทำให้ขนาดของรูปหกเหลี่ยมมีขนาดลดลง องค์ประกอบโทนสีคือมุมในรูปหกเหลี่ยมโดยถือว่าแกนสีแดงมีมุมเป็นศูนย์องค์ประกอบแวร์ลูย์ (Value) ถูกวัดตามแนวแกนของโคน (แกนดิ่ง) โดยที่ V = 0 คือยอดของโคนสีดำ ส่วน V = 1 คือส่วนฐานตรงกับสีขาว ค่าแซ็ททูเรชันถูกวัดจากระยะทางจากแกน V โดยลักษณะนี้เราถือว่าระบบ HSV คือระบบในพิกัดทรงกะบอก (Cylindrical Coordinate System) ส่วน RGB อยู่ในระบบพิกัดคาร์ทิเซียน (Cartesian Coordinate System)

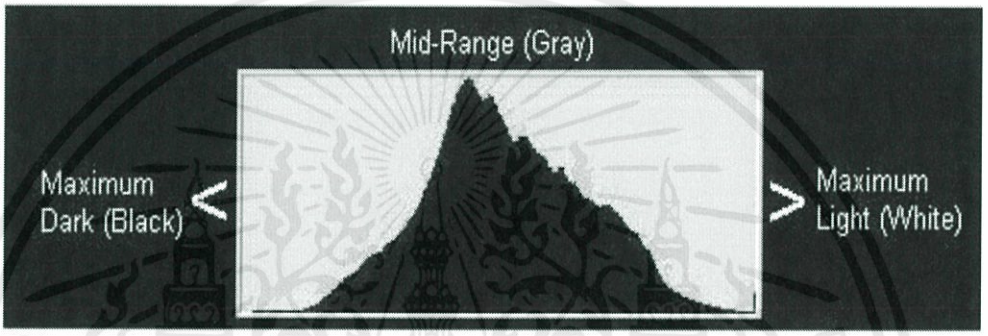


รูปที่ 2.7 แบบจำลอง HSV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 ฮิสโตแกรม (Histrogram)

ฮิสโทแกรม (Histogram) เป็นกราฟแสดงจำนวนพิกเซลที่ความสว่างต่างๆ ของภาพ สังเกตได้จากภาพด้านล่าง แกนนอนเป็นระดับความสว่างที่แบ่งระดับเป็น 256 ระดับ (มักเรียกว่าระดับสีเทา หรือ gray level) โดยมีค่าตั้งแต่ 0-255 เมื่อระดับสีเทามีค่าต่ำ (ด้านซ้ายมือ) หมายถึง มีความสว่างน้อย จะมองเห็นเป็นสีดำ ค่าระดับสีเทามาก (ด้านขวามือ) หมายถึงมีความสว่างมากจะมองเห็นเป็นสีขาว แกนตั้งของกราฟแสดงจำนวนพิกเซลในแต่ละความระดับสีเทาซึ่งเป็นค่าสัมพัทธ์ สรุปง่ายๆก็คือ ถ้ากราฟฮิสโทแกรมเอนไปทางซ้าย แปลว่าภาพนั้นจะมีความมืดเกินไปแต่ถ้าเอนไปทางขวาแปลว่าภาพนั้นมีความสว่างเกินไปฮิสโทแกรมนั้นใช้เพื่อดูความแปรปรวนของกระบวนการโดยการสังเกตรูปร่างของฮิสโตแกรมที่สร้างขึ้นจากข้อมูลที่ได้มาโดยการสุ่มตัวอย่าง



รูปที่ 2.8 กราฟฮิสโทแกรมที่แสดงถึงแสงที่สมดุล

(a) แสงน้อยเกินไป (b) แสงที่พอเหมาะ (c) แสงมากเกินไป

รูปที่ 2.9 แสดงฮิสโตแกรมแบบต่างๆ

(a) เป็นตัวอย่างของภาพที่ขาดความสว่าง ทำให้ขาดรายละเอียดของภาพโดยเฉพาะด้านหลังของภาพ โดยที่กราฟฮิสโทแกรมแสดงผลเอนไปทางด้านซ้าย ซึ่งแสดงว่าภาพนี้มีแสงน้อยเกินไป

(b) แสงในภาพนี้เกือบมีความสมบรูณ์ของภาพมากโดยมีแค่ความมืดของฉากหลังอยู่บ้าง โดยที่กราฟฮิสโทแกรมแสดงว่าภาพภาพมีการกระจายของแสงที่ดี แต่กราฟเอนไปทางด้านซ้ายมากกว่าทางขวา เนื่องจากมีเงาอยู่ที่ฉากหลังของภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(c) แสดงถึงแสงที่มีมากเกินไปในภาพ จะสังเกตได้ว่าวัตถุที่อยู่ด้านหน้าจะมีความสว่างเกินไป ซึ่งทำให้ไม่เห็นถึงรายละเอียดของภาพ แต่ภาพนี้จะแสดงให้เห็นถึงรายละเอียดฉากหลังของภาพได้ดี

ที่สุด ซึ่งจุดหลักของภาพนี้คือด้านหน้า ดังนั้นควรลดแสงของภาพนี้ลงเพื่อให้เห็นถึงรายละเอียดด้านหน้า โดยกราฟฮิสโทแกรมเอนไปทางขวาสุด และมีแสดงในช่วงของความมืดน้อยมาก

2.5.3 การแปลงทางเรขาคณิตของภาพ (Geometrical Transformation of Image)

การแปลงทางเรขาคณิตของภาพ เป็นขบวนการขั้นพื้นฐานในการประมวลผลภาพแบบดิจิตอล เป็นขบวนการที่เกี่ยวข้องกับการสเกล (Scaling) การเคลื่อนย้ายภาพ (Translation) การหมุนภาพ (Rotation) และการเฉือนภาพ (Shear) ภาพทางดิจิตอลโดยคำจำกัดความแล้วคือฟังก์ชัน 2 มิติ f(x,y) โดยที่ x และ y คือพิกัดหรือตำแหน่งของพิกเซล ค่าของฟังก์ชันที่ตำแหน่ง (x,y) หรือ f(x,y) โดยปกติแล้วเราหมายถึงค่าความเข้มแสง (Intensity) หรือเรียกว่าค่าระดับเทา (Gay Level) การแปลงทางเรขาคณิตของภาพเกี่ยวข้องกับ 2 ขบวนการ (1) การแปลงเรขาคณิตของพิกัด (Geometric Spatial Transformation) ซึ่งกระทำกับตัวแปร x และy และ (2) การกำหนดค่าระดับเทาใหม่ให้กับภาพ (Gray – level Interpolation) ซึ่งกระทำกับฟังก์ชัน f เราขอกล่าวถึงการแปลงเรขาคณิตของพิกัดก่อน โดยในที่นี้จะกล่าวถึง การแปลงเรขาคณิตของพิกัด และการใช้ OpenCV ในการทำการแปลงเรขาคณิตของภาพ

2.5.3.1 การเคลื่อนย้ายภาพ (Translation)

การเคลื่อนย้ายภาพ F(x,y) เทียบกับจุดกำเนิด Cartesian ไปเป็นภาพ g(u,v) เกี่ยวข้องกับการคำนวณตำแหน่งออฟเซ็ทสัมพัทธ์ (Relative Offset Address) ของภาพทั้งสอง ความสัมพันธ์ระหว่างตำแหน่งที่เคลื่อนย้ายคือ

$$u = x + t_x$$

$$v = y + t_y$$

โดยที่ t_x และ t_y แทนระยะของการย้ายตำแหน่ง

สมการการแปลงแบบเคลื่อนย้ายภาพสามารถเขียนในรูปของเมทริกซ์ได้เป็น

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2.5.3.2 การสเกลภาพ (Scaling)

การสกลภาพสามารถทำได้โดยการเปลี่ยนพิกัด Cartesian ของภาพอินพุทตามความสัมพันธ์

$$u = S_x x$$

$$v = S_y y$$

โดยที่ S_x และ S_y คือพารามิเตอร์ของสเกล (Scaling Parameter) ถ้าค่าของ S_x และS_y มากกว่าหนึ่งผลคือทำให้เกิดการขยายในทิศทาง x และ y ตามลำดับ ถ้ามีค่าอยู่ในช่วง (0,1)ผลคือทำให้เกิดการย่อส่วนในทิศทางที่สอดคล้องกัน ในกรณีที่ค่าของ S_x และ S_y น้อยกว่าศูนย์ จะเกิดลักษณะที่เรียกว่า Reflection หรือภาพกลับขวาเป็นซ้าย ซ้ายเป็นขวา สมการเรขาคณิตของการสเกลสามารถเขียนในรูปของเมทริกซ์ได้เป็น

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.3 การหมุนภาพ (Rotation)

การหมุนภาพรอบจุดกำเนิดสามารถทำได้โดยการคำนวณดังนี้

$$u = x\cos\theta - y\sin\theta$$
$$v = x\sin\theta + y\cos\theta$$

โดยที่ θ แทนมุมของการหมุนในทิศทางทวนเข็มนาฬิกาเมื่อเทียบกับแกนนอน (Horizontal Axis) สมการการหมุน 5 และ 6 แทนการหมุนภาพรอบจุดกำเนิดบนระบบพิกัด Cartesian การหมุนภาพรอบจุดใดๆ สามารถทำได้โดย (1) เลื่อนจุดหมุนไปที่จุดกำเนิด (2) ทำการหมุนภาพ และ (3) เลื่อนจุดหมุนกลับไปที่เดิม สมการการแปลงเรขาคณิตของการหมุนรอบจุด (x_c,y_c) ในรูปเมทริกซ์แทนด้วย

$$\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}1 & 0 & xc\\0 & 1 & yc\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}cos\theta & -sin\theta & 0\\sin\theta & cos\theta & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & -xc\\0 & 1 & -xy\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix}$$

2.5.3.4 การเฉือนภาพ (Shearing)

การแปลงภาพในลักษณะที่ทำให้ภาพเปลี่ยนจากรูปร่างสี่เหลี่ยมผืนผ้า (หรือจัตุรัส) ไปเป็นสี่เหลี่ยมขนมเปียกปูนเรียกว่าการเฉือนภาพ ซึ่งแบ่งเป็น 3 แบบ

สมการของการเฉือนไปข้างหน้าในทิศทาง x คือ

$$u = x + sh_y$$
$$v = y$$

สมการของการเฉือนไปข้างหน้าในทิศทาง y คือ

$$u = x$$
$$v = y + sh_x$$

สมการของการเฉือนไปข้างหน้าในทิศทาง x และ y คือ

$$u = x + sh_y$$
$$v = y + sh_x$$

สมการเรขาคณิตของการเฉือนภาพสามารถเขียนในรูปเมทริกซ์ได้เป็น

$$\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}1 & shx & 0\\shy & 1 & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix}$$

การแปลงเรขาคณิตเชิงเส้นสามารถรวมกันได้ สมการการแปลงเรขาคณิตรวมสามารถเขียนได้เป็น

$$T = S \cdot Sh \cdot R \cdot Tr$$

โดยที่

$$S = \begin{bmatrix}sx & 0 & 0\\0 & sy & 0\\0 & 0 & 1\end{bmatrix} \quad Sh = \begin{bmatrix}1 & shx & 0\\shy & 1 & 0\\0 & 0 & 1\end{bmatrix} \quad Tr = \begin{bmatrix}1 & 0 & tx\\0 & 1 & ty\\0 & 0 & 1\end{bmatrix}$$

$$R = \begin{bmatrix}1 & 0 & xc\\0 & 1 & yc\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}cos\theta & -sin\theta & 0\\sin\theta & cos\theta & 0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & -xc\\0 & 1 & -xy\\0 & 0 & 1\end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

ซึ่งการแปลง T ของสมการลดรูปอยู่ในรูปเมทริกซ์ขนาด 3×3 ดังนั้นเราได้

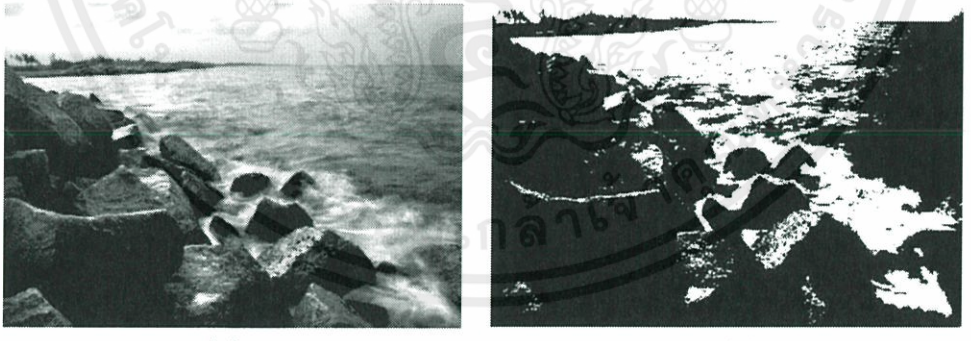
$$\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}a0 & a1 & a2\\b0 & b1 & b2\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix}$$

2.5.4 การปรับปรุงภาพ (Image Enhancement)

การปรับปรุงภาพ (Image Enhancement) หมายถึงการเน้น (หรือการทำให้คม) ลักษณะของภาพ (Features) เช่น ขอบ (Edge) เขต (Boundaries) หรือความแตกต่างระหว่างบริเวณขาวกับดำหรือคอนทราส (Contrast) เพื่อทำให้การแสดงกราฟฟิกมีความชัดเจนมากขึ้นสำหรับการแสดงและการวิเคราะห์ ขบวนการปรับปรุงภาพไม่ได้เป็นการเพิ่มองค์ประกอบข้อมูลในภาพ แต่จะไปเพิ่มช่วงทำงาน (Dynamic Range) ของลักษณะของภาพที่เลือกเพื่อให้ง่ายต่อการตรวจจับ ข้อที่ยากที่สุดสำหรับการปรับปรุงภาพคือการตั้งเกณฑ์ (Criterion) ที่เหมาะสมสำหรับการปรับปรุงภาพ ดังนั้นเทคนิคการปรับปรุงภาพส่วนใหญ่จะมีลักษณะเป็นการทดลองหรือต้องการการตอบโต้ของผู้ใช้ (User Interface) เพื่อให้ได้ผลที่น่าพอใจ อย่างไรก็ตามการปรับปรุงภาพยังคงเป็นหนึ่งในหัวข้อที่สำคัญที่สุดเพราะพบการประยุกต์ใช้งานอย่างมากในการประมวลผลภาพทางดิจิตอล

2.5.4.1 การขลิบและการทำเทรสโชว์ (Clipping and Thresholding)

Thresholding คือ การแยกส่วนของภาพ ซึ่งเป็นการพิจารณาความเข้มของจุดต่าง ๆ ภายในภาพ (Pixel) ซึ่งผลของการแยกส่วนของภาพ จะขึ้นอยู่กับวิธีการ Threshold ของส่วนประกอบที่เป็นความเข้มหรือสีของภาพ สำหรับภาพบางชนิดจะมีลักษณะวัตถุที่เราสนใจซึ่งมีความเข้มที่คงที่เมื่อเทียบกับพื้นหลังตัวอย่างได้แก่ ภาพของตัวอักษร (Text) เป็นต้น ซึ่งภาพเหล่านี้จะมีความเข้มของวัตถุที่เราสามารถแยกออกพื้นหลังได้อย่างชัดเจน(มีความเข้มข้นสองระดับได้แก่ ความเข้มของวัตถุและความเข้มของพื้นหลัง) การทำการแยกส่วนของภาพ สามารถทำได้โดยการกำหนดค่า Thresholdซึ่งเป็นค่าความเข้มให้มีค่าที่สามารถแยกความแตกต่างของวัตถุและพื้นหลังได้ ตัวอย่างอย่างเช่น ภาพของตัวอักษรที่มีความความเข้มของตัวอักษรเป็น 0 (สีดำและมีความเข้มของพื้นหลังเป็น 255 (สีขาว) ดังนั้นค่า Threshold จึงควรจะมีค่าเท่ากับ 128 เพื่อที่จะให้สามารถแยกวัตถุออกจากพื้นหลังได้ โดยปกติแล้วการเลือกค่า Threshold จะขึ้นอยู่กับ Histogram ของภาพ



(a) (b)

รูปที่ 2.10 (a) ภาพที่ยังไม่ได้ทำเทรสโชว์ (b) ภาพที่ทำการเทรสโชว์แล้วที่ค่าระดับกั้นคือ 128

2.5.4.2 การฟิลเตอร์ภาพ (Image Filtering)

การฟิลเตอร์ภาพเป็นการประมวลผลภาพที่สำคัญในการกำจัดองค์ประกอบความถี่ที่ไม่ต้องการออกจากภาพ การกำจัดความถี่สูงออกจากภาพซึ่งเรียกว่าฟิลเตอร์ความถี่ต่ำผ่าน (Low Pass Filter) ผลลัพธ์คือทำให้ภาพดูนุ่มนวลขึ้น (Soft) ซึ่งอาจเรียกฟิลเตอร์นี้ว่าฟิลเตอร์การเฉลี่ย (Averaging) ในทางตรงข้าม การกำจัดความถี่ต่ำออกจากภาพซึ่งเรียกว่าฟิลเตอร์ความถี่สูงผ่าน (High

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

Pass Filter) ผลลัพธ์คือทำให้เหลือแต่องค์ประกอบที่เป็นขอบภาพ หรืออาจเรียกว่าการตรวจหาขอบภาพ (Edge Detection) โดยเรากล่าวถึงฟิลเตอร์ความถี่ต่ำผ่านที่ใช้กำจัดสัญญาณรบกวนที่มีคุณสมบัติแบบเกาส์เซียน (Guassian Noise) และฟิลเตอร์กรองแบบมีเดียน (Median Filter) ที่ใช้กำจัดสัญญาณรบกวนแบบเม็ดขาวและดำหรือที่รู้จักกันในชื่อ Salt and Pepper Noise

2.5.4.2.1 ฟิลเตอร์แบบค่าเฉลี่ย (Averaging)

ฟิลเตอร์แบบนี้แต่ละพิกเซลจะถูกแทนด้วยค่าเฉลี่ยถ่วงน้ำหนักของพิกเซลข้างเคียง ซึ่งหมายถึงการคอนโวลูชัน นั่นคือ

$$y(m,n) = \sum_{(k,l)\in W} \sum h(k,l) \times (m-k, n-l)$$

โดยที่ y(m,n) และ x(m,n) คือภาพเอาท์พุทและภาพอินพุทตามลำดับ W คือหน้าต่าง และ h(k,l) คือค่าน้ำหนักการฟิลเตอร์ ชนิดของฟิลเตอร์แบบค่าเฉลี่ยที่รู้จักกันดีคือชนิดที่มีค่าน้ำหนักเท่ากันนั่นคือ

$$y(m,n) = \frac{1}{N} \sum_{(k,l)\in W} \sum h(k,l) \times (m-k, n-l)$$

โดยที่ $h(k,l) = 1/N_w$ และ N_w คือจำนวนพิกเซลในหน้าต่าง W ฟิลเตอร์การเฉลี่ย นั่นคือ แต่ละพิกเซลจะถูกแทนด้วยค่าเฉลี่ยของสี่พิกเซลที่ใกล้ที่สุด โดยการใช้ฟิลเตอร์แบบค่าเฉลี่ยกำจัดสัญญาณรบกวนที่มีคุณสมบัติแบบเกาส์เซียน (Gaussian Noise) พบว่าขนาดหน้ากากที่เพิ่มขึ้นกำจัดสัญญาณรบกวนได้ดี แต่ผลเสียที่เกิดคือภาพจะเบลอ

2.5.4.2.2 ฟิลเตอร์แบบค่ามีเดียน (Median Filtering)

ฟิลเตอร์แบบค่ามีเดียน (Median Filtering) ที่ใช้กำจัดสัญญาณรบกวนแบบเม็ดขาวและดำหรือที่รู้จักกันในชื่อ Salt and Pepper Noise ในกรณีนี้พิกเซลเอาท์พุทจะถูกแทนที่ด้วยค่ามีเดียนของพิกเซลที่อยู่ในหน้าต่างรอบๆ พิกเซลของภาพอินพุท นั่นคือ

$$y(m,n) = median\{\times (m-k, n-l), (k,l) \in W\}$$

โดยที่ W คือหน้าต่าง อัลกิริทึมสำหรับการฟิลเตอร์แบบมีเดียนจะทำการเรียงค่าพิกเซลในลักษณะที่เพิ่มขึ้นหรือลดลงแล้วทำการเลือกค่าที่อยู่ตรงกลางโดยทั่วไปแล้วขนาดของหน้าต่างถูกเลือกให้ N_w มีค่าเป็นเลขคี่ เช่นขนาด 3×3 เป็นต้น ถ้า N_w เป็นเลขคู่ค่ามีเดียนจะเป็นค่าเฉลี่ยระหว่างสองค่าตรงกลาง เนื่องจากสัญญาณรบกวน Salt & Peppers มีค่าความเข้ม น้อยสุด หรือมากสุดในหน้าต่าง จึงไม่มีผลต่อการหาค่ามีเดียน กล่าวคือค่ามีเดียนเป็นค่าของความเข้มแสงของพิกเซลที่มีค่าตรงกลางในหน้าต่างซึ่งเป็นของพิกเซลปกติที่ไม่ปนเปื้อนสัญญาณรบกวน ฟิลเตอร์กรองแบบมีเดียนจะได้ผลดีกรณีที่เม็ด Salt & Pepper มีขนาดไม่ใหญ่มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรม นับว่ามีความสำคัญ ต่อการใช้งาน ไมโครคอนโทรลเลอร์มากเพราะสามารถใช้แป้นพิมพ์ และจอภาพของ PC เป็น อินพุต และ เอาต์พุต ในการติดต่อ หรือควบคุมไมโครคอนโทรลเลอร์ ด้วยสัญญาณอย่างน้อย เพียง 3 เส้นเท่านั้น คือ

- สายส่งสัญญาณ TX
- สายรับสัญญาณ RX
- สาย GND

2.6.1 การรับส่งแบบซิงโครนัส (Synchronous)

การรับส่งข้อมูล จะมีสัญญาณนาฬิกา ซึ่งเป็นตัวกำหนด จังหวะเวลา การส่งข้อมูลร่วมอยู่ด้วยอีกเส้นหนึ่ง ใช้คู่กับสัญญาณข้อมูล

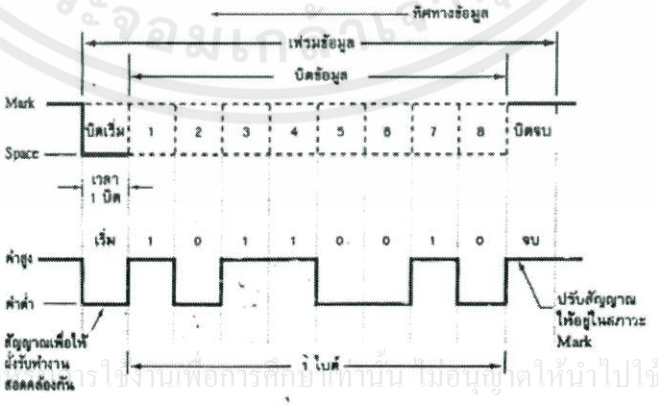
2.6.2 การรับส่งข้อมูลแบบอะซิงโครนัส (Asynchronous)

การรับส่งข้อมูล โดยที่ไม่จำเป็นต้อง มีสัญญาณนาฬิกา ร่วมด้วย แต่จะใช้ให้ตัวส่งและตัวรับมีอัตราส่งข้อมูล ที่เท่ากัน

รูปแบบข้อมูลแบบอะซิงโครนัส ประกอบด้วย 4 ส่วนคือ

- บิตเริ่มต้น (Start bit) มีขนาด1 บิต
- บิตข้อมูล (Data) มีขนาด 5, 6, 7 หรือ 8 บิต
- บิตตรวจสอบพาริตี้ (Parity bit) มีขนาด 1 บิต หรือไม่มี
- บิตหยุด (Stop bit) มีขนาด 1, 1.5, 2 บิต

เพื่อเข้าใจการทำงานของการสื่อสารแบบอะซิงโครนัส เริ่มจากการส่งข้อมูลครั้งละ 1 บิต ผ่านพอร์ทอนุกรม สถานะของสัญญาณเริ่มโดยบิตเริ่มต้นแสดงถึงการเริ่มต้นของการส่งของข้อมูลไบต์ เหมือนในรูปที่ 2.11 บิตเริ่มต้นใช้สำหรับบอกจุดเริ่มต้นของไบต์ข้อมูล โดยการทำให้สถานะของสายส่งมีค่าเป็น low โดยมีค่าเป็น 1 บิต จากนั้นจะเป็นบิตของข้อมูลตามด้วยพาริตี้บิต ซึ่งจะมีหรือไม่มีก็ได้ สุดท้ายคือบิตสิ้นสุดซึ่งจะมี 1 หรือ 2 บิตก็ได้



รูปที่ 2.11 เฟรมการส่งข้อมูลแบบอะซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

สถานะของสายส่งในขณะที่ไม่มีข้อมูลจะมสถานะเป็น High และเมื่อข้อมูลบิตใดมีค่าเป็น 0 จะทำให้สายส่งมีค่าเป็น low ข้อมูลบิตใดมีค่า 1 ก็จะมีสถานะ high

2.6.3 มาตรฐานRS232

โดยปกติโมโครคอมพิวเตอร์จะมีพอร์ทอนุกรมที่เรียกว่า RS 232 อยู่ในตัว โดยพอร์ทนี้ทำหน้าที่รับและส่งข้อมูลในแบบอนุกรมเรียกว่า Universal Asynchronous Adapter มาตรฐาน RS 232 ได้จัดพิมพ์ขึ้นเมื่อปี ค.ศ.1969 โดยที่ RS ย่อมาจาก Recommend Standard ส่วน 232 เป็นหมายเลขบ่งบอกของมาตรฐานตัวนี้ส่วน C เป็นหมายเลขท้ายสุดของมาตรฐานฉบับนี้ จุดประสงค์ของมาตรฐานนี้คือเพื่อบรรยายคุณลักษณะของการเชื่อมต่ออุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment: DTE) กับอุปกรณ์สื่อสารข้อมูล (Data Communication Equipment : DCE) ซึ่งจะขึ้นอยู่กับผู้ผลิต ความเร็วและระยะทางของการเชื่อมตอ่ RS232 นั้นสามารถเชื่อมต่อการถ่ายโอนข้อมูลได้ประมาณ 0-20000 บิตต่อวินาที ส่วนความยาวของสายเชื่อมต่อสัญญาณตามมาตรฐานของ RS232 จำกัดอยู่ที่ 50 ฟุต พอร์ทอนุกรมส่วนใหญ่จะมีรูปร่างขึ้นอยู่กับมาตรฐานของ RS-232 คือมีขาคอนเนคเตอร์ทั้งแบบ 25 ขาและแบบ 9 ขาแสดงดังรูปที่ 2.12

รูปที่ 2.12 DB25 connector และ DB9 connector

สัญญาณพื้นฐานของ RS-232 ที่กำาหนดให้แต่ละขาของคอนเนคเตอร์แสดงดังตารางที่ 2.2

ตาราง 2.2 D Type 9 Pin and D Type 25 Pin Connectors หน้าที่ของแต่ละขา

Common name	Description	Pin numbers 25-pin connector	Pin numbers 9-pin connector
TxD	Transmit Data	2	3
RxD	Receive Data	3	2
RTS	Request To Send	4	7
CTS	Clear To Send	5	8
DSR	Data Set Ready	6	6
SG	Signal Ground	7	5
CD	Carrier Detect	8	1
DTR	Data terminal ready	20	4
RI	Ring Indicator	22	9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง

Transmit Data (TxD) เป็นสัญญาณที่ส่งออกจาก DTE (หรือโมโครคอมพิวเตอร์) ไปยังโมเดมหรือต่อเข้าโดยตรงกับไมโครคอมพิวเตอร์ ข้อมูลแบบอนุกรมจะถูกส่งออกจากคอมพิวเตอร์ด้วยขานี้ สถานะของขานี้จะมีค่าเท่ากับ “1” หรือเทียบเท่ากับบิตหยุด

Receive Data (RxD) เป็นทางของสัญญาณเข้าไปยัง DTE หรือ โมโครคอมพิวเตอร์ข้อมูลแบบอนุกรมจะรับเข้าเครื่องคอมพิวเตอร์ด้วยขานี้ เมื่อไม่มีสัญญาณเข้ามาขานี้จะมีสถานะทางลอจิกเป็น “1”

Request To Send (RTS) สัญญาณที่ขานี้จะเป็นส่วนที่บอกโมเดมว่า UART พร้อมที่จะส่งข้อมูล ซึ่งขานี้ใช้สำหรับส่งสัญญาณไปยังโมเดมหรือเครื่องพิมพ์ เป็นการร้องขอที่จะส่งสัญญาณมาทางขา 2 สัญญาณนี้ใช้คู่กับ CTS (Clear To Send) อุปกรณ์รับหากได้สัญญาณ RTS จะตรวจสอบตัวเองว่าพร้อมจะรับสัญญาณได้หรือยัง หากพร้อมที่จะรับก็จะส่งสัญญาณออกไปที่ขา CTS

Clear To Send (CTS) สัญญาณที่ขานี้จะเป็นส่วนที่แสดงว่าโมเดมพร้อมที่จะส่งข้อมูล เมื่อสัญญาณนี้อยู่ในสภาวะออฟ (แรงดันมีค่าเป็นลบ หรือ ลอจิก 1) ซึ่งหมายความว่าอุปกรณ์พร้อมที่จะรับข้อมูล

Data Set Ready (DSR) สัญญาณที่ขานี้จะเป็นตัวบอก UART ว่าโมเดมพร้อมที่จะทำการเชื่อมต่อ เมื่อสัญญาณนี้อยู่ในสภาวะออน (ลอจิก 0) เป็นการบอกโมโครคอมพิวเตอร์หรือฝ่ายส่งว่าโมเดมต่อเข้ากับสายโทรศัพท์เรียบร้อยแล้วและพร้อมที่จะส่งได้แล้ว โมเดมที่มีการหมุนหมายเลขอัตโนมัติจะส่งสัญญาณนี้ออกไปบอกให้คอมพิวเตอร์รู้ว่าต่อโทรศัพท์ได้สำเร็จแล้ว

Signal Ground (SG) ขากราวนด์ทำหน้าที่เป็นระดับแรงดันอ้างอิงสำหรับทุกๆสัญญาณ

Carrier Detect (CD) โมเดมจะทำการส่งสัญญาณนี้ให้กับเครื่องคอมพิวเตอร์เมื่อได้รับสัญญาณ carrier จากโมเดมปลายทางอีกฝั่งหนึ่ง

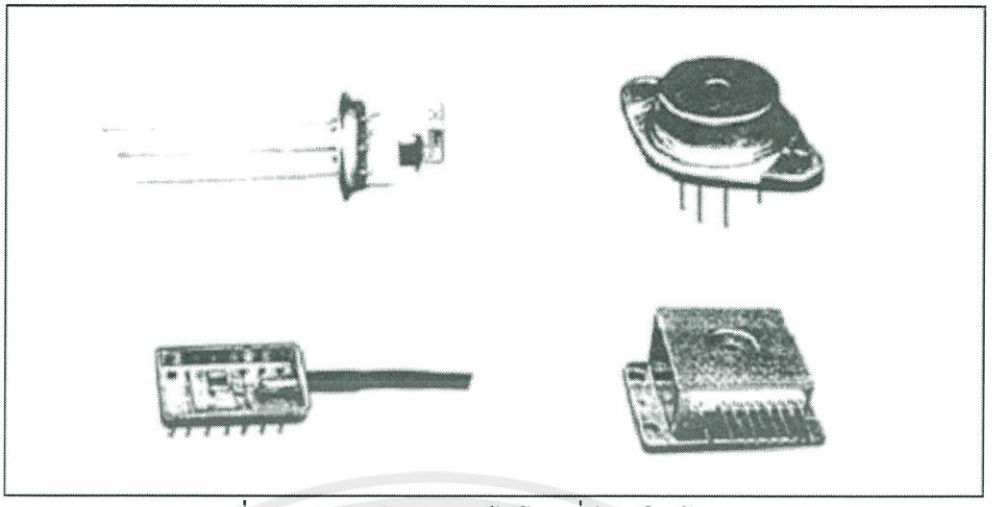
Data Terminal Ready (DTR) สัญญาณดังกล่าวจะตรงกันข้ามกับสัญญาณ DSR นั้นสัญญาณที่ขานี้จะเป็นตัวบอกโมเดมว่า UARTพร้อมที่จะทำาการเชื่อมต่อ

Ring Indicator (RI) จะทำงานเมื่อโมเดมได้รับสัญญาณ Ringing จากโครงข่าย PSTN สัญญาณทั้งหมดนี้

2.7 เลเซอร์ไดโอด (Laser Diode)

เลเซอร์ไดโอดมีโครงสร้างรอยต่อแบบ p-n ของสารกึ่งตัวนำทำจากสารประกอบกึ่งตัวนำ เช่น GaAs, InP มีขนาดเล็กมาก สิ่งประดิษฐ์มีปริมาตรเพียงไม่ถึง 1 ลูกบาศก์มิลลิเมตร บรรจุอยู่ในกล่อง โลหะ (metal package) ที่มีขาติดอยู่เพื่อใช้ป้อนกระแสไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 ตัวอย่างเลเซอร์ไดโอดที่มีขายในท้องตลาด

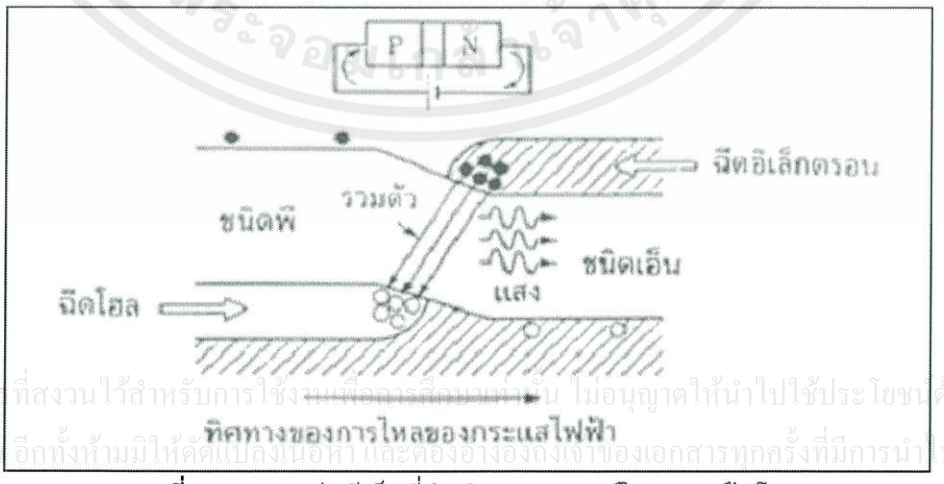
คุณสมบัติเด่นของเลเซอร์ไดโอดซึ่งเลเซอร์ชนิดอื่นไม่มี ได้แก่

1. มีขนาดเล็กและเบามาก
2. มีประสิทธิภาพสูง
3. สามารถมอดูเลตความเข้มของการเปล่งแสงด้วยกระแสไฟฟ้าได้โดยตรง
4. มีอายุการใช้นานยาวนาน
5. ผลิตจำนวนมากได้ง่ายและราคาถูก

ปัจจุบันเลเซอร์ไดโอดถูกนำไปใช้งานอย่างกว้างขวาง เช่น การสื่อสารด้วยแสง การวัด สารสนเทศ คอมแพกต์ดิสก์ เลเซอร์ดิสก์ ซีดีรอม การแพทย์ การบันเทิง ฯลฯ

2.7.1 กลไกการเปล่งแสงของเลเซอร์ไดโอด

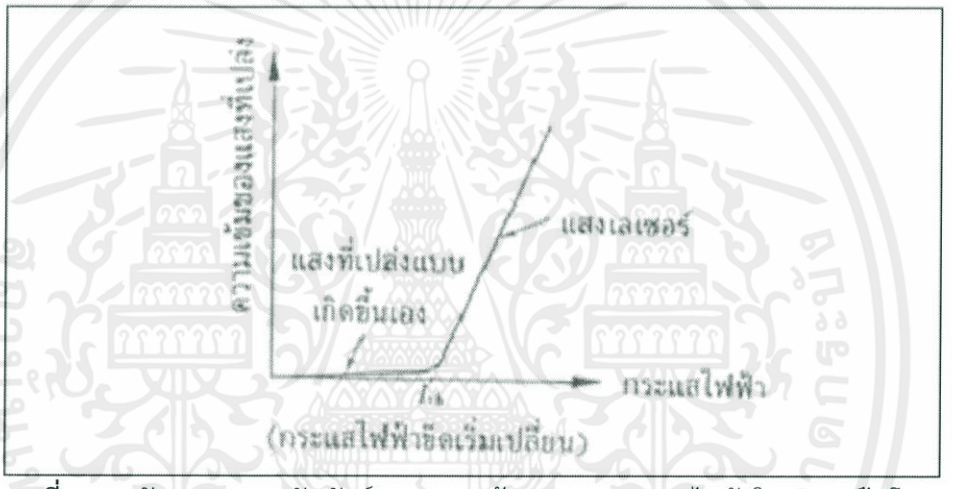
ในการกระตุ้นพาหะเพื่อให้เกิดแสงเลเซอร์นั้น จะใช้วิธีฉีดกระแสไฟฟ้าจากภายนอกให้ ไหลเข้าสู่รอยต่อ p-n โดยป้อนแรงดันไฟฟ้าแบบไบแอสตาม นั่นคือ ป้อนแรงดันบวกเข้าด้าน p และป้อนแรงดันลบเข้าด้าน n จะทำให้แถบพลังงานเปลี่ยนรูปเป็นรูปที่ 2.14 อิเล็กตรอนจะไหลจากด้าน n เข้าสู่รอยต่อ และโฮลจะไหลจากด้าน p เข้าสู่รอยต่อ



รูปที่ 2.14 รอยต่อพีเอ็นที่กำเนิดแสงเลเซอร์ในเลเซอร์ไดโอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่บริเวณรอยต่อจะมีคู่ของอิเล็กตรอนและโฮลเกิดขึ้นจำนวนมาก เมื่ออิเล็กตรอนและโฮลรวมตัวกัน (recombine) จะเกิดการเปล่งแสงออกมาด้วยพลังงานโฟตอนเท่ากับช่องว่างพลังงานของรอยต่อ (E = hv) หลักการเปล่งแสงเช่นนี้คล้ายกับกรณีของ LED แต่การทำให้เกิดแสงเลเซอร์นั้นจะต้องฉีดกระแสไฟฟ้าให้มากเป็นพิเศษดังแสดงในรูปที่ 2.15 ถ้ากระแสไฟฟ้ามีค่าน้อย การเปล่งแสงจะเป็นแบบเกิดขึ้นเอง (spontaneous) และเมื่อเพิ่มกระแสไฟฟ้าให้สูงกว่ากระแสไฟขีดเริ่มเปลี่ยน (threshold current : I_{th}) จะเกิดปรากฏการณ์ประชากรผกผัน (population inversion) กล่าวคือ ถ้าจำนวนคู่ของอิเล็กตรอนและโฮลมีมากกว่าค่าๆหนึ่ง ก็จะทำให้เกิดการรวมตัวของอิเล็กตรอนและโฮลและเกิดการเปล่งแสงแบบเร่งเร้า (stimulated emission) ที่ด้านข้างภายนอกของรอยต่อ p-n จะถูกออกแบบให้เป็นผิวนอกแบบกระจก ให้ทำหน้าที่เป็นออปติคัลเรโซเนเตอร์ (optical resonator) แสงที่สะท้อนกลับไปมาภายในเรโซเนเตอร์นี้ จะกลายเป็นแสงเลเซอร์วิ่งออกสู่ภายนอกในที่สุด



รูปที่ 2.15 ลักษณะความสัมพันธ์ของความเข้มแสงและกระแสไฟฟ้าในเลเซอร์ไดโอด

2.7.2 ลักษณะสมบัติกำลังเอาท์พุทและกระแสไฟฟ้า

เราเรียกความสัมพันธ์ระหว่างกระแสไฟฟ้า (Injection current) ที่ฉีดเข้าสู่เลเซอร์ไดโอดและพลังงานเอาท์พุท (energy output) ว่า ลักษณะสมบัติความสัมพันธ์ของอินพุทและเอาท์พุท (input-output characteristics) ในรูปที่ 2.15 เมื่อเพิ่มแรงดันไฟฟ้าให้ใกล้เคียงค่าช่องว่างพลังงานจะทำให้กระแสไฟฟ้าเริ่มไหลและจะเริ่มมีแสง เปล่งแบบเกิดขึ้นเองออกมาดังรูปที่ 2.16 และเมื่อเพิ่มกระแสอีกมากๆ จนสูงกว่าค่าขีดเริ่มเปลี่ยนค่าหนึ่งจะเกิดการ (lasing) และกำลังของแสงของเอาท์พุทจะสูงขึ้นอย่างรวดเร็ว ค่ากระแสไฟฟ้าขีดเริ่มเปลี่ยนนี้จะขึ้นกับอุณหภูมิขณะทำงาน ถ้าอุณหภูมิสูงขึ้น จะทำให้อัตราขยาย (gain) ลดลงและจะทำให้กระแสไฟฟ้าขีดเริ่มเปลี่ยนมีค่าสูงขึ้น ความสัมพันธ์ระหว่างกระแสไฟฟ้าขีดเริ่มเปลี่ยน(I_{th}) และอุณหภูมิของชั้นแอกทีฟ (T_j) คือ

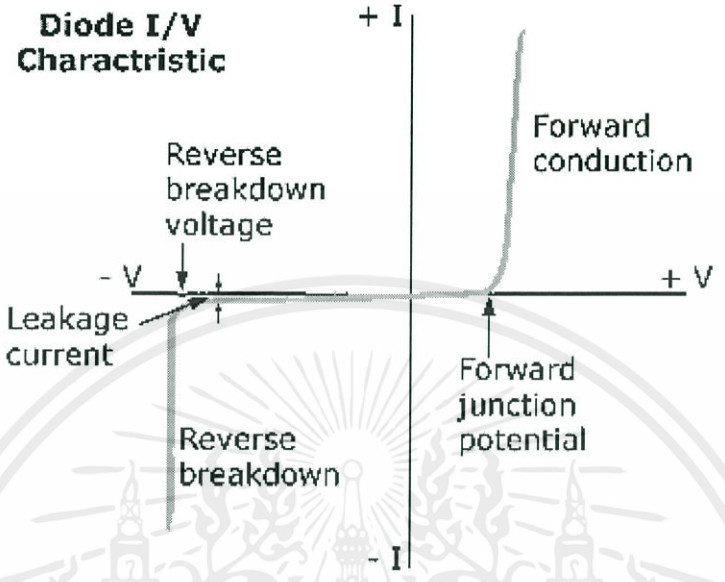
$$I_{th} = I_{tho}\exp(T_j/T_o)$$

โดยที่ Ith : กระแสไฟฟ้าขีดเริ่มเปลี่ยนที่อุณหภูมิห้อง

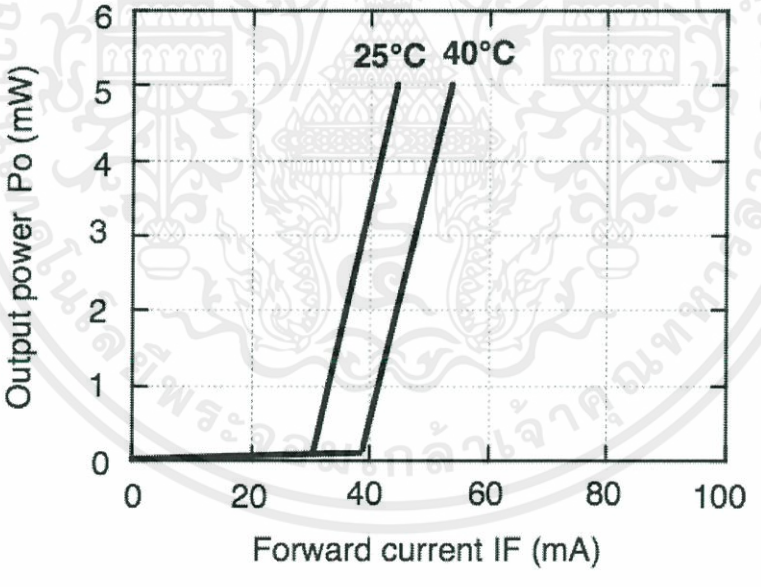
To : ค่าคงตัวที่ขึ้นกับวัสดุ มีชื่อเรียกว่า “อุณหภูมิลักษณะ” (characteristic temperature)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.18 แสดงตัวอย่างการเปลี่ยนแปลงของกระแสไฟฟ้าขีดเริ่มเปลี่ยนเมื่ออุณหภูมิของเลเซอร์ไดโอดเปลี่ยนแปลง

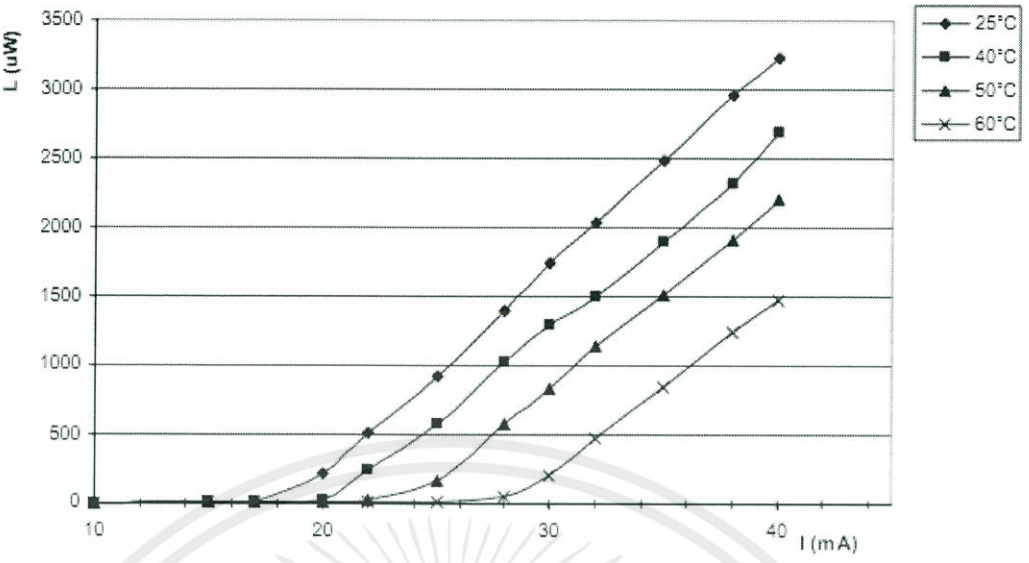


รูปที่ 2.16 ลักษณะสมบัติกระแสไฟฟ้าและแรงดันของเลเซอร์ไดโอด



รูปที่ 2.17 ลักษณะสมบัติกำลังของเลเซอร์เอาท์พุทและกระแสที่ฉีดเข้าเลเซอร์ไดโอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 การเปลี่ยนแปลงของกระแสไฟฟ้าขีดเริ่มเปลี่ยนอันเนื่องจากอุณหภูมิของเลเซอร์ไดโอด

2.8 ไมโครคอนโทเลอร์

อุปกรณ์อิเล็กทรอนิกส์ ที่สามารถนำมาทดแทน ระบบดิจิตอลแบบที่ใช้อุปกรณ์ตัวเดี่ยว ๆ ได้ จึงสามารถลดความซับซ้อน ในการออกแบบและสร้าง วงจรที่ให้ฟังก์ชันทางดิจิตอล ได้หลากหลาย จากการเขียนโปรแกรม ทำให้เป็นอุปกรณ์ที่มีอยู่ในระบบทางดิจิตอลสมัยใหม่ที่ซับซ้อนเกือบ ทั้งหมด ในงานวิจัยนี้ ใช้ไมโครคอนตระกูล PIC® ซึ่งหาซื้อได้ได้ง่าย และราคาในการพัฒนาซอฟต์แวร์ไม่แพง มาก เนื่องจากไมโครชิปยึดถือการออกแบบให้รวมทุกอย่างไว้ใน ชิป ตัวเดียวโดยไม่ต้องต่ออุปกรณ์ ใดๆ เพิ่มเติม ทำให้ระบบมีขนาดเล็ก และอุปกรณ์ที่ใช้จะไม่มาก บางงานอาจจะใช้แค่ PIC®เพียงตัว เดียวโดยไม่ต้องใช้ ชิป อื่นมาเพิ่มเติมเลย แต่ข้อเสีย เนื่องจาก แนวคิด ที่จะรวมทุกอย่างไว้ในชิปเดียว ทำให้ program memory และ data memory ไม่สามารถขยายโดยใช้กับ memory ภายนอกได้ ยาก PIC จึงเหมาะสำหรับงานเล็กๆ ไม่ใช่งานใหญ่ๆ ที่ต้องใช้การคำนวณ และ memory เยอะๆ PIC®ที่นำมาใช้ในงานวิจัยนี้ จะเป็นตระกูล 16F8XXX ซึ่งมีสมบัติคร่าว ๆดังนี้

PIC16F8XXX (FLASH MCUs)

- มีคำสั่งในภาษา assembly 35 คำสั่ง
- มี I/O, มี Timer มากกว่า 1 ตัว, Watch dog, I2C, USART, SPI, PWM
- มี A/D ขนาด 10 bits
- มี Program memroy เป็นแบบ Flash ทำให้สามารถโปรแกรมใหม่ได้หลายครั้ง
- มี EEPROM ภายใน
- ในตระกูล 16F87X สนับสนุน In Circuit Debugger (ICD) เป็นผลทำให้ไม่จำเป็นต้องซื้อ Emulator ราคาแพง

จุดด้อยของ PIC ก็คือ program memory มีลักษณะเป็น page ทาง Microchip เองก็คง เห็นจุดด้อยตรงนี้จึงได้พัฒนาออกมาเป็น PIC18CXXX ซึ่ง PROGRAM MEMORY ไม่ถูกแบ่งเป็น page อีกต่อไป และเพิ่มคำสั่ง ASSEMBLY เป็น 77 คำสั่ง รวมทั้งออกแบบให้ PROGRAM MEMORY มีขนาดใหญ่ขึ้นเพื่อรองรับ

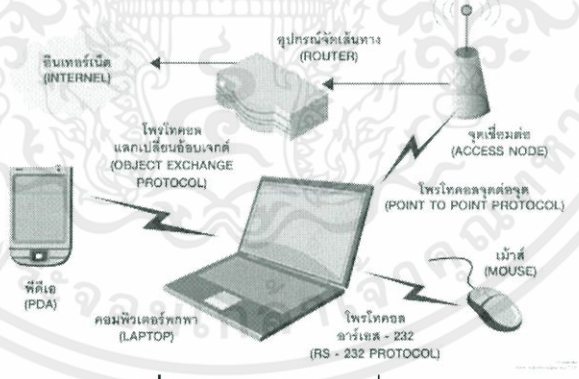
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

2.9 Bluetooth

บลูทูท คือ เทคโนโลยีการสื่อสารไร้สายระยะใกล้แบบเครือข่ายไร้สายส่วนบุคคล(wireless personal area networks: WPAN) เป็นมาตรฐานที่ถูกออกแบบมาเพื่อใช้ในการเชื่อมต่ออุปกรณ์ไร้สายขนาดเล็ก เช่น เครื่องพีดีเอ (personal digital assistant:PDA) อุปกรณ์สื่อสารแบบพกพาหรือเคลื่อนที่รวมไปถึงการเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตผ่านอุปกรณ์ปลายทางที่ให้บริการ

เทคโนโลยีการสื่อสารไร้สายบลูทูท ถูกนำมาประยุกต์ใช้งานในการเชื่อมต่อกับอุปกรณ์ต่างๆ ทำให้สะดวกต่อการใช้งาน เนื่องจากไม่จำกัดพื้นที่ มิต้องใช้อุปกรณ์ที่เป็นสายสัญญาณ สามารถเชื่อมต่อได้ไกล เช่น การส่งข้อมูลจากโทรศัพท์เคลื่อนที่เครื่องหนึ่งไปยังโทรศัพท์เคลื่อนที่อีกเครื่องหนึ่ง หากส่งผ่านสายสัญญาณ จำเป็นต้องใช้อุปกรณ์เสริมเพื่อทำให้อุปกรณ์ทั้งสองเชื่อมต่อกันได้ แต่เทคโนโลยีบลูทูธ ช่วยให้การส่งข้อมูลของอุปกรณ์ทั้งสองสะดวกขึ้นโดยการส่งผ่านคลื่นวิทยุ

ระบบเครือข่ายไร้สายส่วนบุคคลที่ใช้เชื่อมต่อ โดยตรงระหว่างอุปกรณ์ที่อยู่ใกล้กันชนิดนี้ ในแต่ละเครือข่าย จะมีอุปกรณ์ตัวหนึ่ง เรียกว่า มาสเตอร์(Master) หรือตัวแม่ข่าย ทำหน้าที่ควบคุมการทำงานและประสานงานให้กับอุปกรณ์ตัวอื่นๆในเครือข่ายเดียวกัน ส่วนอุปกรณ์ที่เชื่อมต่อตัวอื่นๆ เรียกว่า สลาฟ(Slave) หรือตัวลูกข่าย ซึ่งโครงสร้างการทำงานของบลูทูธนี้คล้ายกับระบบบัสอนุกรมแบบใช้ร่วมร่วม (universal serial bus : USB) ที่ใช้กับเครื่องคอมพิวเตอร์หรืออุปกรณ์อิเล็กทรอนิกส์ทั่วๆไป แต่ต่างกันในส่วนของการเชื่อมต่อ โดยอุปกรณ์บลูทูธส่วนใหญ่สามารถทำหน้าที่ได้ทั้งมาสเตอร์หรือสลาฟ ตามความเหมาะสม ซึ่งภายในเครือข่ายจะมีการจัดการกันเองโดยอัตโนมัติด้วย โพรโทคอลมาตรฐานอุปกรณ์บลูทูธแต่ละตัวจะมีแอดเดรส (Address) หรือการระบุตำแหน่ง ซึ่งเป็นรหัสประจำตัวที่ไม่ซ้ำกับอุปกรณ์ตัวอื่น มีความยาวขนาด 48 บิต เรียกว่า บีดี แอดเดอ (BD_ADDR) ใช้ในการจำแนกอุปกรณ์แต่ละตัวและใช้ในการระบุความถี่ที่ใช้สำหรับอุปกรณ์ตัวนั้นๆ



รูปที่ 2.19 รูปแบบการสื่อสาร

2.9.1 ระยะเชื่อมต่อของบลูทูธ

ความสามารถในการส่งข้อมูลของบลูทูธนั้นขึ้นกับแต่ละ class ที่ใช้ ซึ่งมี 3 class ดังนี้

- Class 1 กำลังส่ง 100 มิลลิวัตต์ ระยะประมาณ 100 เมตร
- Class 2 กำลังส่ง 2.5 มิลลิวัตต์ ระยะประมาณ 10 เมตร
- Class 3 กำลังส่ง 1 มิลลิวัตต์ ระยะประมาณ 1 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 มาตรฐานของบลูทูธ

ปัจจุบันข้อกำหนด Bluetooth ออกมาแล้ว 8 รุ่น

- Bluetooth 1.0
- Bluetooth 1.1
- Bluetooth 1.2 z
- Bluetooth 2.0
- Bluetooth 2.0 EDR
- Bluetooth 2.1 EDR
- Bluetooth 3.0
- Bluetooth 4.0
    - ระบบ EDR : Enhanced Data Rate เพิ่มความเร็วในการส่งข้อมูลสูงสุดเป็น 3 Mbps.

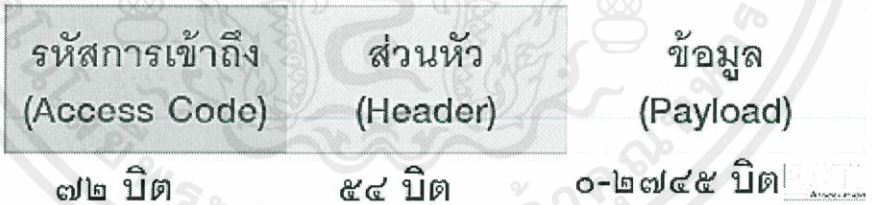
2.9.3.ส่วนประกอบของชุดข้อมูล

ข้อมูลที่รับส่งอยู่ในเครือข่ายบลูทูทถูกแบ่งออกเป็นหน่วยย่อยๆ เรียกว่า พีดียู (packet data unit: PDU) ซึ่งประกอบไปด้วย

ก) รหัสการเข้าถึง (Access Code) เป็นส่วนที่เก็บข้อมูลหมายเลขเครือข่ายและแอดเดรส หรือตำแหน่งของอุปกรณ์ต้นและปลายทาง มีขนาดยาว 72 บิต

ข) ส่วนหัว (Header) เป็นส่วนที่เก็บข้อมูลเส้นทางที่เหมาะสมในการส่งข้อมูล มีขนาดยาว 54 บิต

ค) ข้อมูล (Payload) คือข้อมูลที่ต้องการส่งไปยังปลายทาง มีขนาดระหว่าง 0 – 1745 บิต ขึ้นอยู่กับการใช้งาน ดังรูปที่ 2.20



รูปที่ 2.20 โครงสร้างชุดข้อมูลของบลูทูธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 งานวิจัยที่เกี่ยวข้อง

ผู้ป่วยอัมพาตตั่งแต่ไหล่ลงมายังสามารถควบคุมลมหายใจ หมุนคอ เอียงหน้าได้ แต่ส่วนใหญ่จะไม่สามารถสื่อสารด้วยวาจาได้ การสื่อสารและสั่งงานไปยังผู้ช่วยเหลือ ดูแล ทำได้โดยการใช้สายตาหรือภาษากายซึ่งมีขีดจำกัดมากมาย ทำให้เกิดความเครียดส่งผลกับสภาพจิตใจและร่างกายตามมา ได้มีการพัฒนาอุปกรณ์ช่วยเหลือผู้ป่วยอัมพาตให้สามารถช่วยเหลือตัวเอง หรือสื่อสารได้บ้าง เช่นใช้คลื่นสมองในการสั่งการ ใช้อุปกรณ์ตรวจจับการกระพริบตา หรือเคลื่อนไหวดวงตา ซึ่งต้อการเครื่องมือที่ซับซ้อนต้นทุน สูงในการพัฒนาเป็นอุปกรณ์ช่วยเหลือผู้ป่วยเฉพาะราย ได้มีการนำเลเซอร์ไดโอดในรูปของเลเซอร์พอยน์เตอร์มาใช้ในการประดิษฐ์เครื่องช่วยสื่อสารสำหรับผู้ป่วยกล้ามเนื้ออ่อนแรง เนื่องจากราคาถูก ลำแสงเป็นเส้นตรงเหมาะกับการชี้เป้าหมาย อันตรายน้อยและน้ำหนักเบา เป้าหมายเป็นกระดานตัวอักษรหรือประโยคง่าย ๆ เพื่อการสื่อสารกับผู้ดูแล หรือชี้เป้าเป็นอุปกรณ์รับแสงอิเล็กทรอนิกส์ควบคุมการเปิดปิดไฟฟ้าแบบง่าย ๆ งานวิจัยนี้ใช้หลักการของเลเซอร์ พัฒนาเป็นอุปกรณ์สั่งงาน ที่มีฟังก์ชั่นมากขึ้น เน้นการออกแบบให้ใช้งานสะดวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการทำงานของโปรแกรม

3.1 ฉากรับภาพ

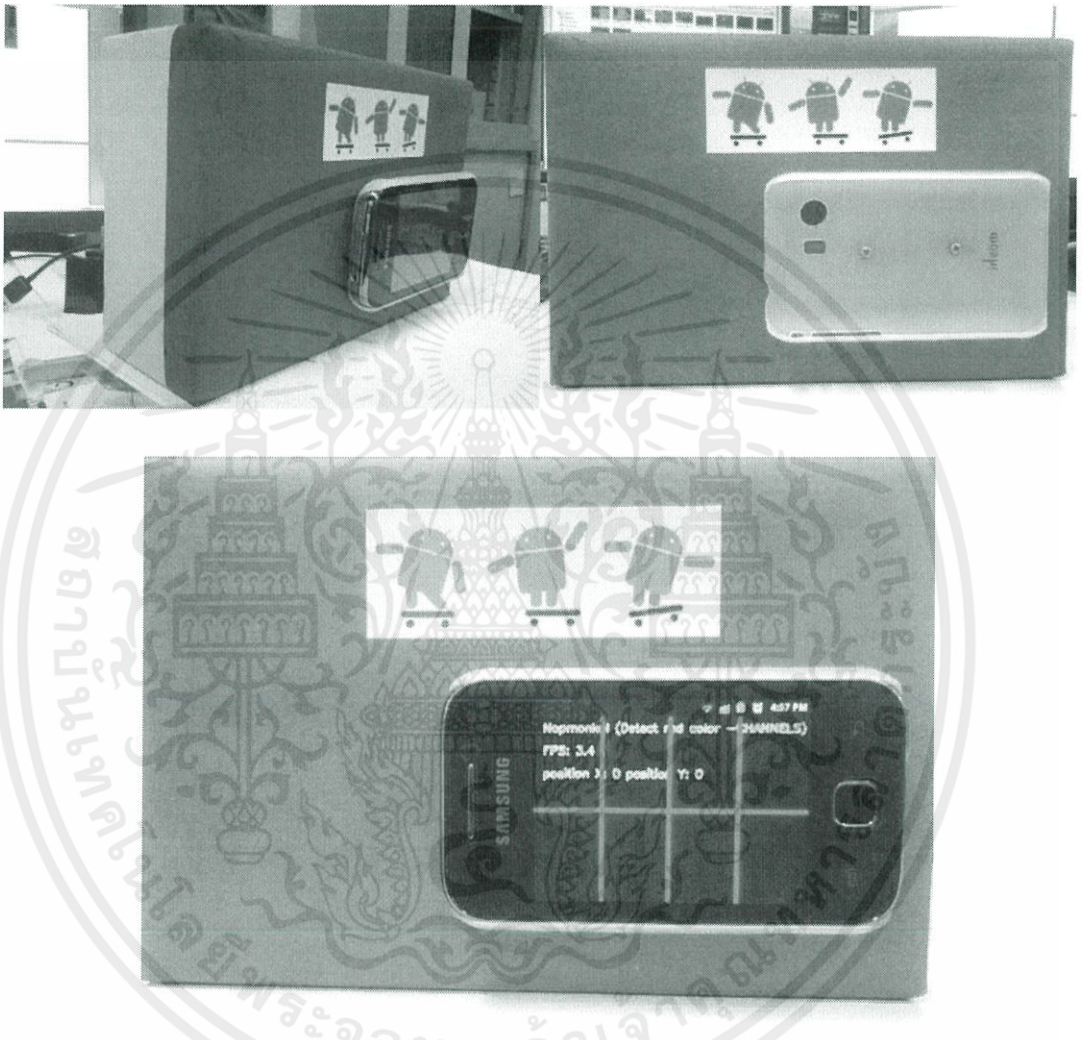
ฉากรับภาพที่เป็นพื้นสีขาวซึ่งจะมีสีที่ให้ความแตกต่างกับเลเซอร์สีแดงดีและสวยงาม วัสดุที่นำมาเป็นฉากรับภาพจำเป็นต้องไม่ทำให้แสงหักเหจนจุดเกินบริเวณกว้างเกินไปและต้องควรเป็นวัสดุโปรงแสงที่มีสีพื้นที่ให้ความแตกต่างกับเลเซอร์สีแดงได้ดีและต้องมีความสวยงามและสะอาดเพื่อความสะดวกกับผู้ใช้ ดังรูปที่ 3.1

รูปที่ 3.1 ฉากรับภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การติดตั้งกล้อง

การติดตั้งกล้องนั้นจะต้องติดตั้งกล้องให้สามารถรับภาพที่มีแต่ฉากรับภาพเท่านั้นและสามารถเห็นส่วนของฉากรับภาพได้อย่างชัดเจนและมีขนาดพอเหมาะให้ผู้ใช้สามารถใช้ได้สะดวก จึงต้องหาระยะที่เหมาะสม โดยจะต้องตั้งให้ขนานกับระนาบเดียวกันกับฉากรับภาพ และห่างจากฉากรับภาพพอสมควรซึ่งขึ้นกับความพอใจในขนาดของฉากรับภาพ ซึ่งจะได้การตั้งกล้องดังแสดงในรูปที่ 3.2

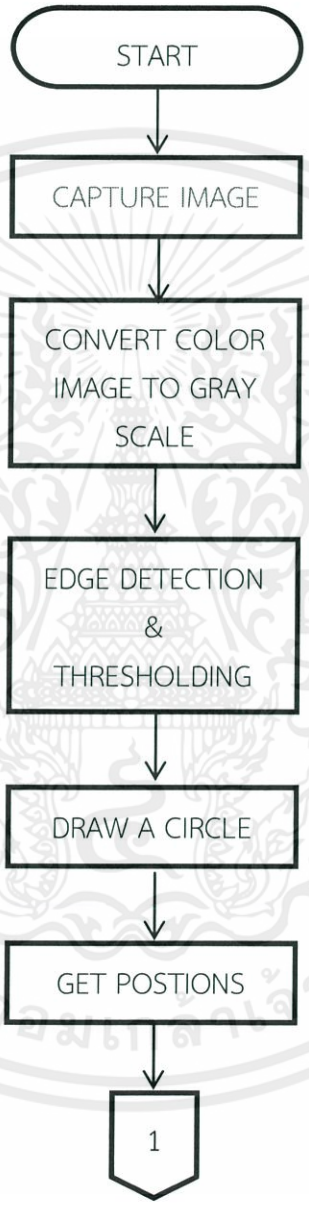


รูปที่ 3.2 การติดตั้งกล้องให้สามารถรับภาพได้พอดีกับฉากรับภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

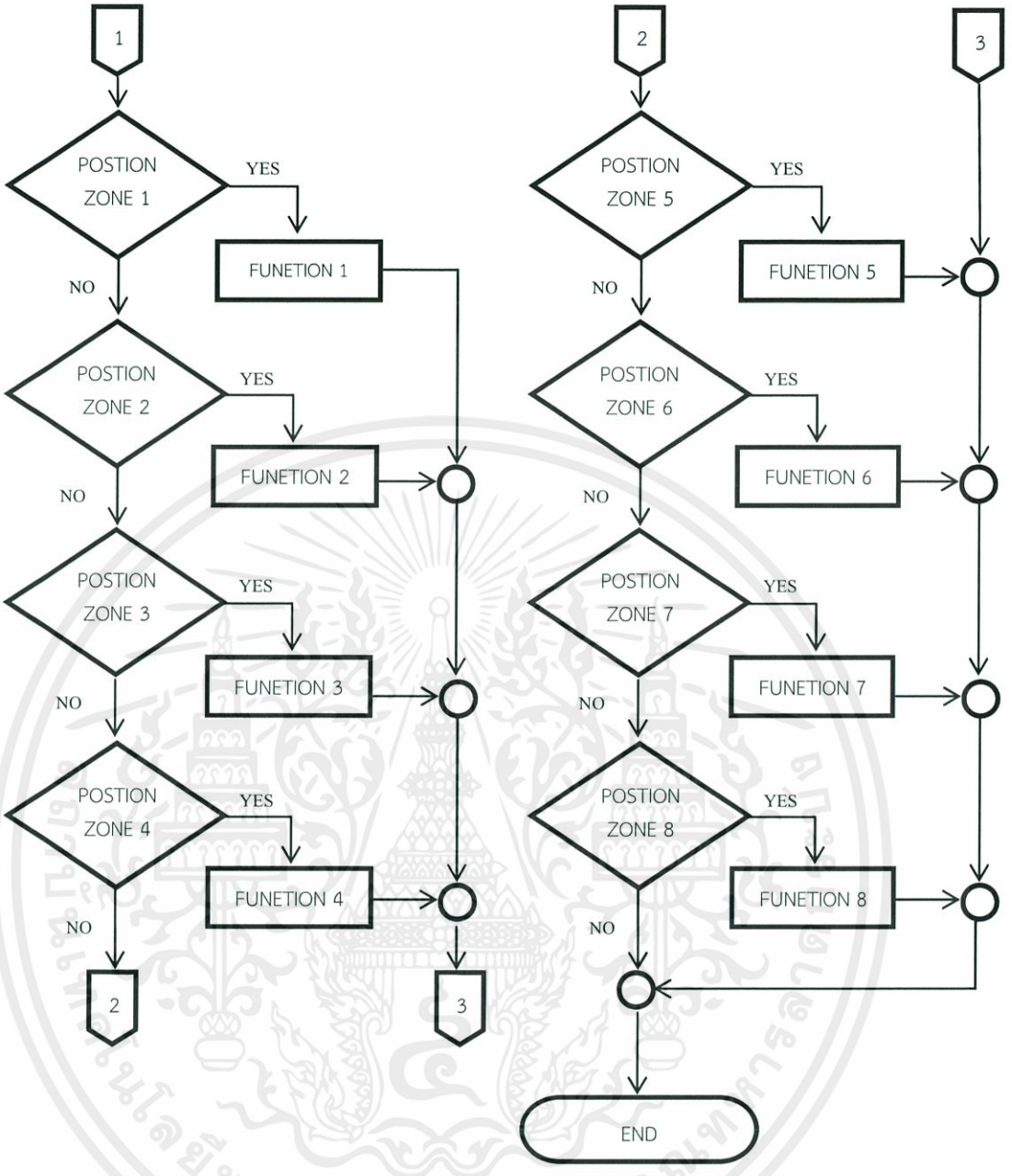
3.3 การทำงานของโปรแกรม

เมื่อเริ่มโปรแกรม โปรแกรมจะมีการทำงานคือ เมื่อมีเลเซอร์มาตกกระทบกับฉากรับภาพในบริเวณที่กำหนด กล้องจะจับภาพนั้นโดยการจับแบบจับตลอดเวลา (Real Time) และนำภาพไปประมวลผลภาพเมื่อประมวลผลเสร็จจะคืนค่าตำแหน่งบนหน้าจอออกมา ตามรูปที่ 3.3 และ รูปที่ 3.4 ซึ่งแสดง Folw Chart ของโปรแกรมหลักที่ใช้ในการประมวลผลภาพ



รูปที่ 3.3 แสดง Flow Chart การทำงานของโปรแกรมการประมวลผลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดง Flow Chart การทำงานของโปรแกรมการประมวลผลภาพ(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function 1-8 เป็นการทำงาน ต่างๆที่ต้องการใช้งาน โดยได้ทำการตั้งค่าไว้เป็น

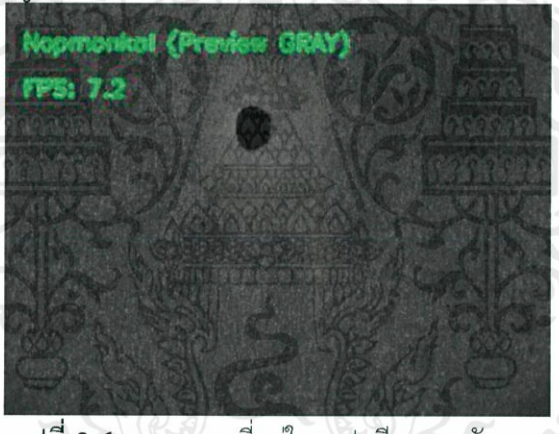
1. เปิดเสียงเพลง
2. ปิดเสียงเพลง
3. ส่งข้อความสั้น (SMS) "**** Call Me****"
4. ส่งข้อความสั้น (SMS) "**** Help Me****"
5. เปิด-ปิด เครื่องใช้ไฟฟ้า ช่อง 1
6. เปิด-ปิด เครื่องใช้ไฟฟ้า ช่อง 2
7. เปิด-ปิด เครื่องใช้ไฟฟ้า ช่อง 3
8. เปิด-ปิด เครื่องใช้ไฟฟ้า ช่อง 4
9. เปิด-ปิด เครื่องใช้ไฟฟ้า ช่อง 5
10. เปิด-ปิด เครื่องใช้ไฟฟ้า ช่อง 6
11. เปิด-ปิด เครื่องใช้ไฟฟ้า ช่อง 7
12. เปิด-ปิด เครื่องใช้ไฟฟ้า ช่อง 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การเปลี่ยนสเปซสีจาก RGB เป็น Gray scale

จากรูปที่ 3.5 จะพบว่าภาพที่ได้นั้นยังคงเป็นภาพสีซึ่งถ้าจะใช้ในการคำนวณภาพที่หาจุดสีแดงนั้นเป็นไปได้ยาก จึงต้องการทำให้ภาพไปอยู่ในระดับเทาเพื่อให้เห็นถึงความแตกต่างของความสว่างที่ชัดเจนมากขึ้นจึงต้องมีการเปลี่ยนสเปซของสี และได้ดังรูปที่ 3.6 แสดงภาพที่อยู่ในสเปซสีของระดับเทา

รูปที่ 3.5 แสดงภาพที่ได้จากกล้องในสเปซ RGB

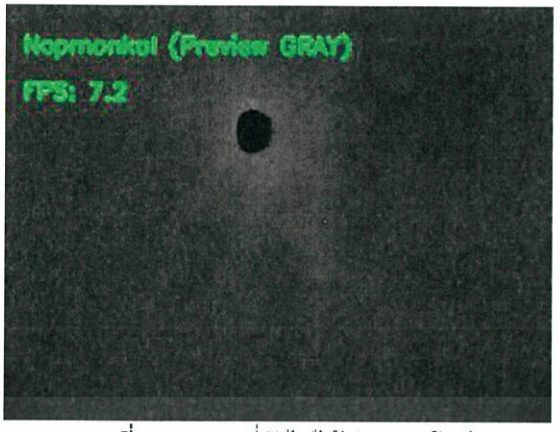


รูปที่ 3.6 แสดงภาพที่อยู่ในสเปซสีของระดับเทา

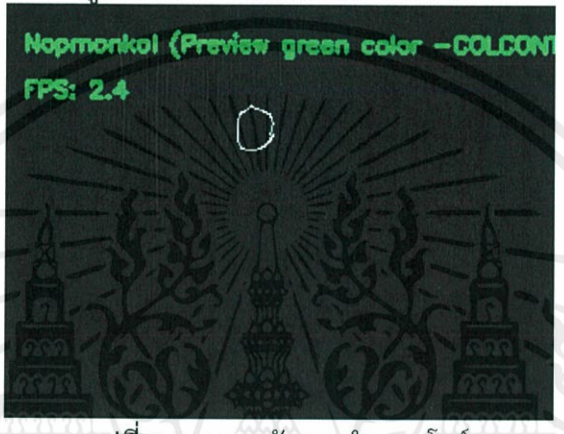
3.5 การทำเทรสโชว์ภาพ

จากภาพระดับเทาทำให้เราสามารถมองเห็นจุดของแสงเลเซอร์แล้ว แต่ยังคงเห็นแสงเลเซอร์ไม่ชัดเจนดังรูปที่ 3.7 จึงต้องมีการปรับปรุงภาพอีกครั้ง โดยการทำเทรสโชว์ (Thresholding) ซึ่งจะเป็นการแบ่งภาพออกเป็น 2 ระดับ คือ ขาวกับดำ ซึ่งการที่จะรู้ได้นั้นว่าจุดที่แตกต่างนั้นมีค่าระดับเทาของจุดอยู่ที่เท่าไรเราสามารถหาได้จาก กราฟฮิสโตแกรม ดังรูปที่ 3.9 และเมื่อทำการการหาขอบภาพโดยค่าระดับกั้นอยู่ที่ 35-100 ค่าของพิกเซลที่มีค่าอยู่นอกช่วงนี้จะเป็นสีดำ(มีค่า 0) และถ้ามีค่าอยู่ในช่วงนี้ จะกลายเป็นสีขาว (มีค่า 255) แล้วจะได้รูปที่มีจุดสีขาวที่เด่นชัดขึ้น ดังรูปที่ 3.7

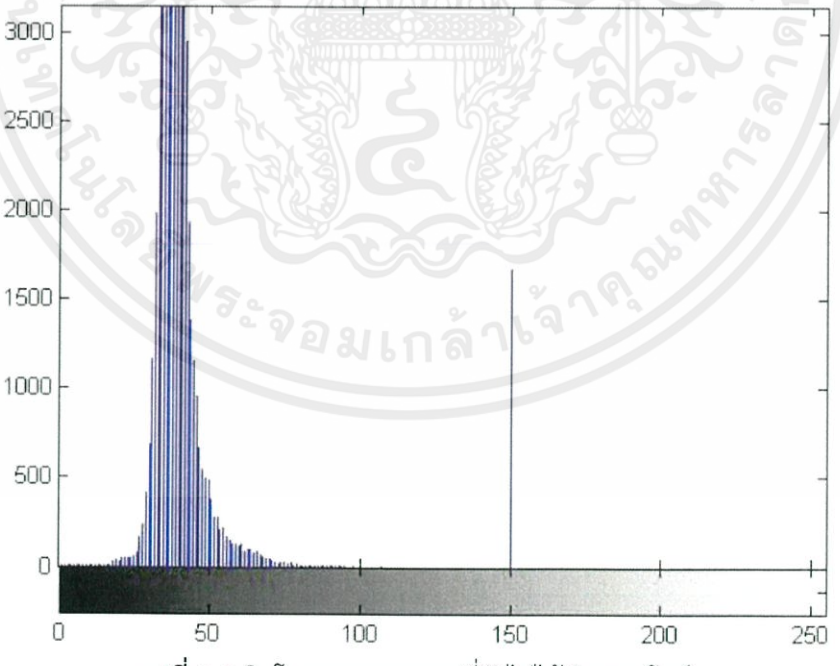
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 ภาพที่ยังไม่ได้ทำเทรสโชว์



รูปที่ 3.8 ภาพหลังจากทำเทรสโชว์

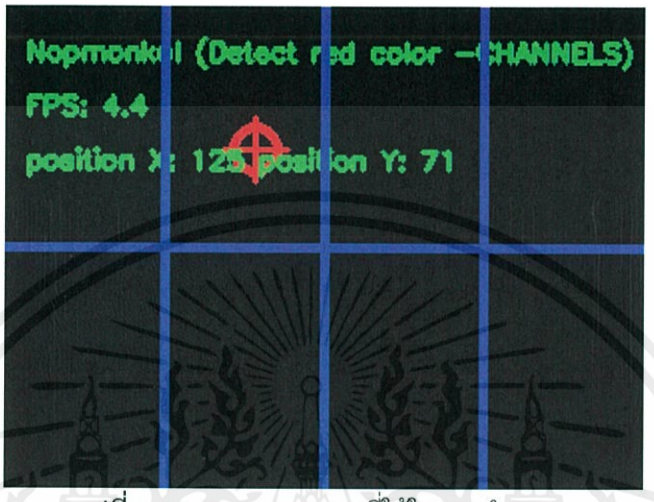


รูปที่ 3.9 ฮิสโตแกรมของภาพที่ยังไม่ได้ทำเทรสโชว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การเลือกย่านที่ใช้ในการคำนวณ

ภาพที่ได้จากการประมวลผลแล้วจะทำให้คอมพิวเตอร์มองภาพเห็นในแบบเดียวกับที่ผู้ใช้เห็น นั้นจะต้องกำหนดขอบเขตให้โดยใช้ตารางในการกำหนดระยะซึ่งจำนวนช่องของตารางผู้ออกแบบสามารถใช้งานได้เท่าไหร่ก็ได้จนมากที่สุดตามผลการทดลองในบทถัดไป ในที่นี้ได้สมมติใช้งานเป็น 8 ช่อง ดังที่ได้แสดงในรูปที่ 3.10

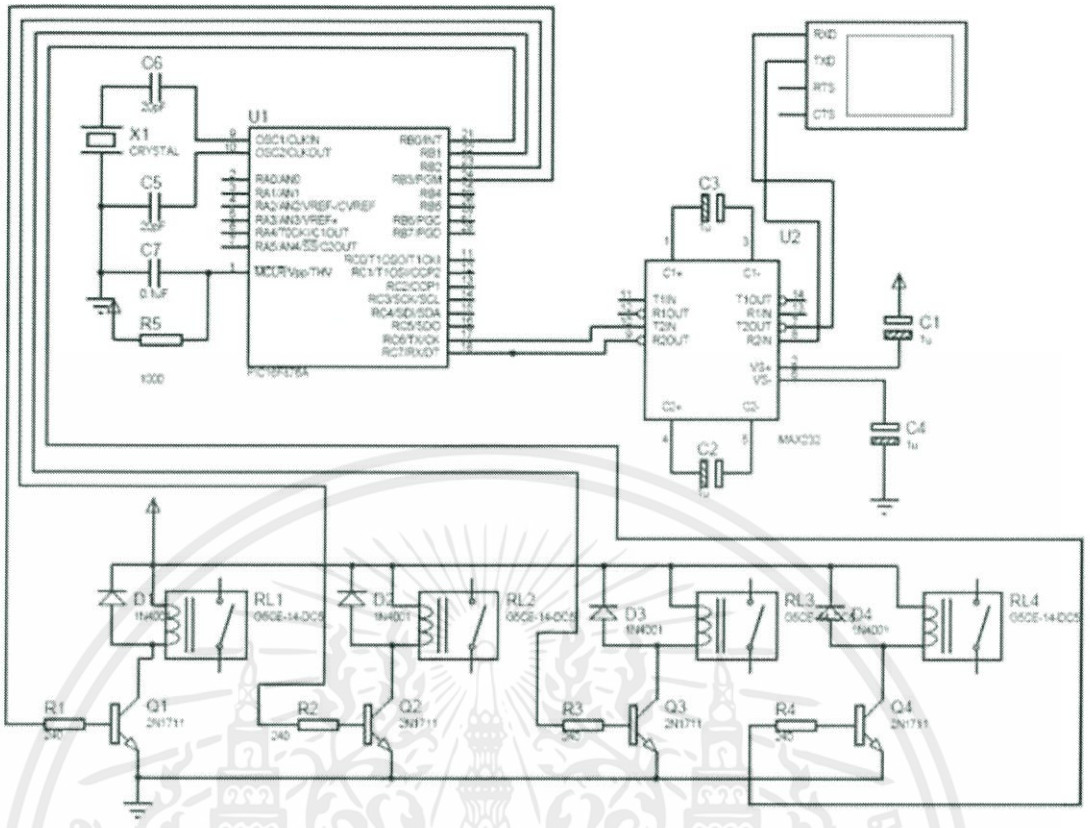


รูปที่ 3.10 แสดงขอบเขต ที่ใช้ในการคำนวณ

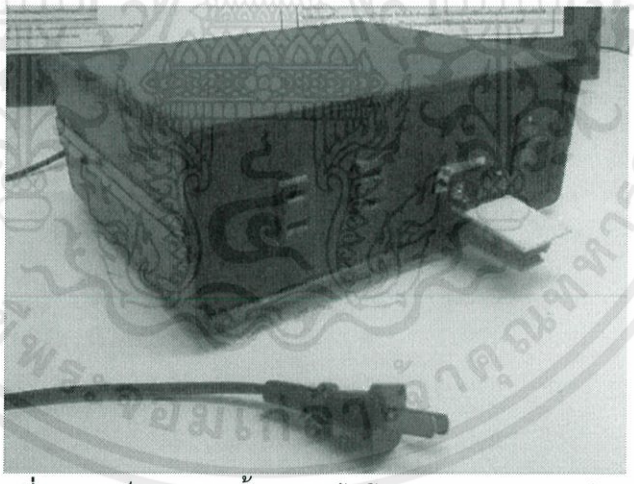
3.7 การออกแบบส่วนควบคุมเครื่องใช้ไฟฟ้า

ส่วนนี้เป็นส่วนที่รับรหัสจากส่วนประมวลผลภาพ ที่คำนวณตำแหน่งของจุดแสงเลเซอร์บนฉาก ผ่าน พอร์ตอนุกรม เพื่อควบคุมการทำงานของ รีเลย์เปิด/ปิดการทำงานของเตรื่องใช้ไฟฟ้า ระบบที่ได้ออกแบบแสดงในรูปที่ 3.11 รหัสคำสั่งจะถูก ส่งผ่านทางพอร์ตอนุกรม RS-232 โดยส่งที่ขาส่งสัญญาณ TX และผ่านเข้าสู่ Max232 ซึ่งเป็นวงจรรวมปรับระดับสัญญาณตามมาตรฐาน RS-232ที่ขา R IN แล้วออกที่ขา R OUT ของ Max232 เพื่อเข้าสู่ PIC16F876A ที่ขา18 (Rx) และทำหน้าที่ถอดรหัสควบคุมการทำงานของ Relay ผ่านทางขา RB0-RB7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 วงจรส่วนรับข้อมูลและควบคุมการเปิดปิดเครื่องใช้ไฟฟ้า

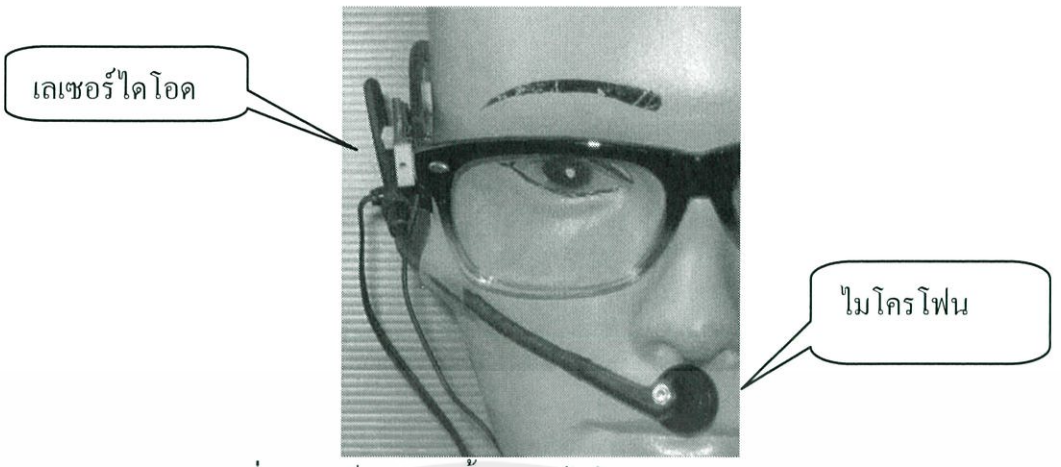


รูปที่ 3.12 เป็นการติดตั้งเลเซอร์ไดโอดบริเวณขาของแว่นตา

3.8 การออกแบบส่วนแว่นตา

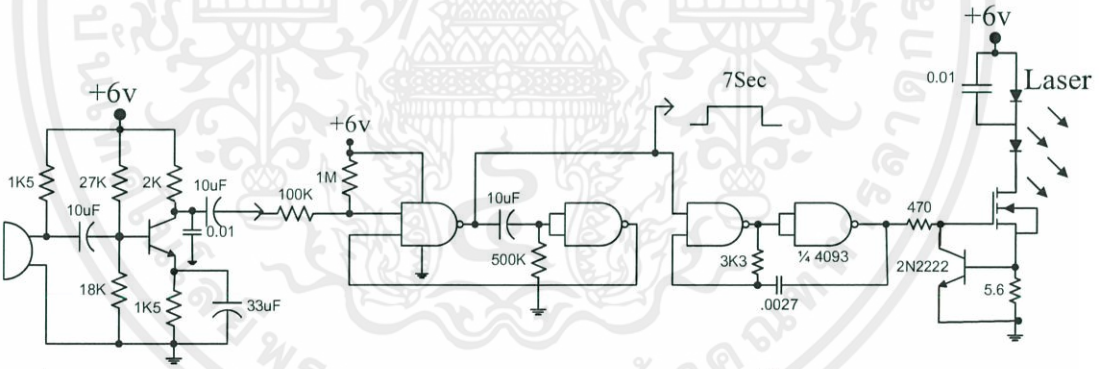
ในผู้ป่วยที่อาการอัมพาตทั้งตัว ที่มีอาการไม่หนักมากจะหมายถึงผู้ป่วยที่เป็นอัมพาตตั้งแต่ไหล่ลงมา ซึ่งเป็นกลุ่มเป้าหมาย ในการใช้งานของแว่นตาสั่งงานด้วยแสงเลเซอร์นี้ ผู้ป่วยกลุ่มนี้ยังสามารถ ควบคุมการหมุนหันศีรษะได้บ้าง แว่นตาจึงเป็นอุปกรณ์ ที่เหมาะกับการใช้ควบคุมตำแหน่งของลำแสงเลเซอร์ให้ตกกระทบบนฉากได้ดี ในรูปที่ 3.13 เป็นการติดตั้งเลเซอร์ไดโอดบริเวณขาของแว่นตา ที่มีกระจกแบบปกติไม่มีกำลังขยาย อย่างไรก็ดี การใช้เลนส์เป็นกระจกแว่นตา ก็สามารถช่วยลดปัญหาการใช้สายตาได้ในผู้ป่วยที่มีความบกพร่องทางสายตาด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

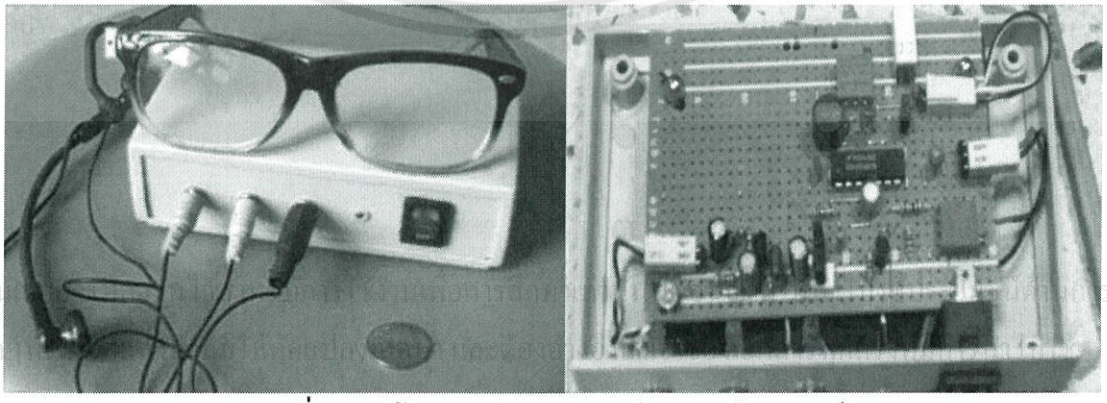


รูปที่ 3.13 เป็นการติดตั้งเลเซอร์ไดโอดบริเวณขาของแว่นตา

เปิดเลเซอร์ไดโอดในช่วงเวลาสั้น ๆ การออกแบบเบื้องต้นจึงใช้ คอนเดนเซอร์ไมโครโฟน ในการตรวจจับ ลมหายใจที่ผู้ป่วยเป่าออกมาแรง ๆ กว่าปกติเล็กน้อย (เพื่อให้แตกต่างจากการหายใจตามปกติเล็กน้อย) วงจรขยายสัญญาณเสียงจากไมโครโฟนและวงจรกรองสัญญาณ จะกระตุ้นให้วงจรขับเลเซอร์ไดโอดทำงาน ในเวลาประมาณ 5-10 วินาทีโดยวงจรตั้งเวลาอิเล็กทรอนิกส์ ในรูปที่ 3.14 แสดงวงจรขยายและกรองสัญญาณจากไมโครโฟนในการตรวจจับเสียงจากการหายใจออกแรง ๆ หรือเสียงเกิดจากการเป่า และและกรองเป็นวงจรตั้งแวลาและขับเลเซอร์ไดโอด กำหนดกระแสเฉลี่ยของวงจรให้ต่ำที่สุดไม่เกิน 5 มิลลิแอมป์ เพื่อสามารถใช้งานต่อเนื่องได้นานที่สุดแหล่งจ่ายไฟเป็นแบตเตอรี่ขนาด AA 4 ก้อน



รูปที่ 3.14 วงจรขยายวงจรขยาย กรองสัญญาณจากไมโครโฟน วงจรตั้งแวลาและขับเลเซอร์ไดโอด



รูปที่ 3.15 ต้นแบบแว่นตาและ กล่องวงจรขับเลเซอร์

เอกสารนี้เป็น
ไม่ว่ากรณีใด

บทที่ 4
การทดลองและผลการทดลอง

4.1 อุปกรณ์การทดลอง

1. โทรศัพท์มือถือระบบปฏิบัติการแอนดรอยด์ 1 อัน
2. เลเซอร์ 1 อัน
4. ฉากรับเลเซอร์ 1 อัน

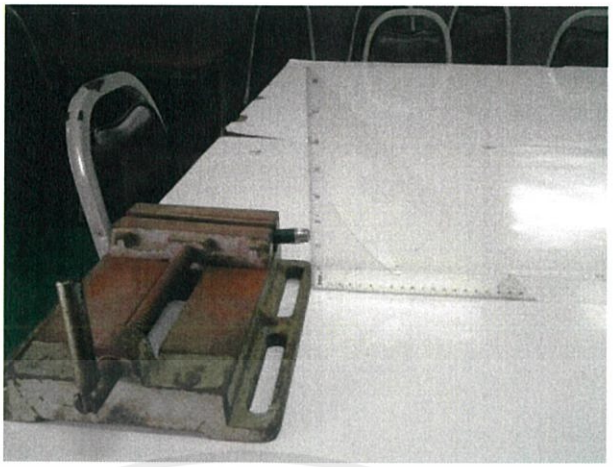
4.2 ขั้นตอนการทดลอง

4.2.1 การทดลองหาขนาดตารางในการรับค่า

รูปที่ 4.1 แสดงการจัดวางอุปกรณ์

1. จัดวางอุปกรณ์ตามรูปที่ 4.1
2. เปิดโปรแกรมที่ได้พัฒนา
3. เลื่อนเลเซอร์ให้อยู่ที่บริเวณฉากรับ
4. เลื่อนเลเซอร์ไปที่ตำแหน่งต่างๆเพื่อบันทึกผล

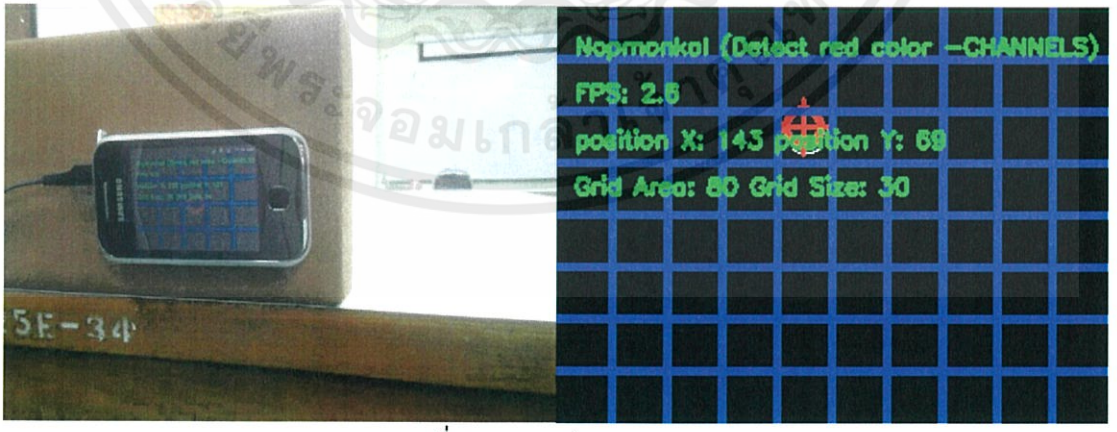
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 เลเซอร์ไดโอด

รูปที่ 4.3 ฉากรับ

4.3 ผลการทดลอง



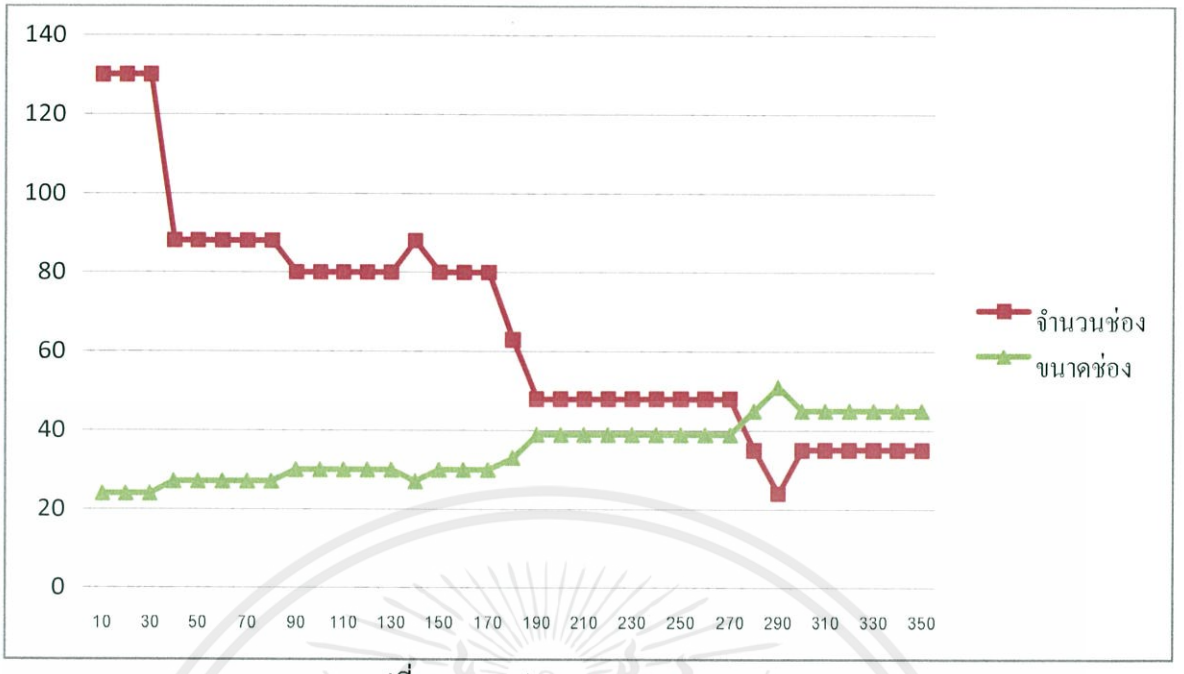
รูปที่ 4.4 ขณะทำการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 4.1 ผลทำการทดลอง

ระยะทาง	จำนวนช่อง	ขนาดช่อง
10	130	24
20	130	24
30	130	24
40	88	27
50	88	27
60	88	27
70	88	27
80	88	27
90	80	30
100	80	30
110	80	30
120	80	30
130	80	30
140	88	27
150	80	30
160	80	30
170	80	30
180	63	33
190	48	39
200	48	39
210	48	39
220	48	39
230	48	39
240	48	39
250	48	39
260	48	39
270	48	39
280	35	45
290	24	51
300	35	45
310	35	45
320	35	45
330	35	45
340	35	45
350	35	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 กราฟแสดงผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

แว่นตาสั่งงานเพื่อผู้ป่วยอัมพาตเป็นอุปกรณ์ออกแบบสำหรับผู้ป่วยพิการตั้งแต่ไหล่ ลงมาและไม่สามารถใช้มือและแขนได้สะดวก โดยการส่วนแว่นตาใช้การสั่งงานด้วยลมหายใจแรง ๆ ไปยังไมโครโฟน ซึ่งจะทำให้เลเซอร์ไดโอดติดสว่างในช่วงสั้น ๆ ผู้ใช้งานเพียงควบคุมให้ลำแสงชี้ไปยังฉากจน แสงเลเซอร์ดับ

ส่วนสร้างรหัสควบคุมใช้กล้องใน โทรศัพท์มือถือจับภาพของจุดเลเซอร์และส่งภาพไปประมวลผล ซึ่งทำงานตามโปรแกรมที่เขียนขึ้นมาตามหลักการของ Digital Image Processing เพื่อคำนวณตำแหน่งของจุดเลเซอร์เทียบกับจุดอ้างอิงเพื่อแปลงเป็นรหัสส่งออก โปรแกรมที่เขียนใช้ภาษา JAVA และไลบรารี ของ Open CV บน Android

รหัสข้อมูลที่ส่งออกมาจะนำไปใช้สั่งการ การทำงานต่างๆ ไม่ว่าจะเป็นการส่งเสียง ส่งข้อความสั้น(SMS) บนตัวเครื่องของโทรศัพท์เอง หรือสั่งสัญญาณออกมาทาง Bluetooth เพื่อสั่งการเปิด-ปิดเครื่องใช้ไฟฟ้า ที่มีกล่องที่ออกแบบขึ้นมาเพื่อเชื่อมต่อกับเครื่องใช้ไฟฟ้าที่ต้องการควบคุม

การควบคุมสั่งงานที่ผ่านทางแว่นตานั้น จะเหมาะสำหรับผู้ใช้ที่สามารถหายใจได้เองและควบคุมการหมุนศีรษะได้เท่านั้นและอาจจะต้องจำเป็นในการฝึกการควบคุมบ้างในระยะแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ผศ.ดร.ชูชาติ ปิณฑวิรุจณ์, Digital Image Processing with c++, ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพ พ.ศ.25xx

[2]http://msdn.microsoft.com/enus/library/aa365747%28VS.85%29.aspx.

[3]http://www.shadowwares.com/forum/index.php?topic=375.0;prev_next=next#new.

[4]http://www.webcam2home.com/webcam-knowledge-th.htm.

[5]http://guru.google.co.th/guru/thread?tid=301f4a7ef195bee2.

[6]http://www.vcharkarn.com/vblog/18204/1.

[7]
http://itd.htc.ac.th/st_it50/it5012/P_2/Object%20Oriented%20Programming%201/B2.htm

[8]http://th.wikipedia.org/wiki/%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%9B%E0%B8%A3%E0%B8%B0%E0%B8%A1%E0%B8%A7%E0%B8%A5%E0%B8%9C%E0%B8%A5%E0%B8%A0%E0%B8%B2%E0%B8%9E

[9]http://imageprocessingblog.blogspot.com/2007/09/image-processing.html

[10]http://guru.google.co.th/guru/thread?tid=06bdd4176c180bb3

[11]http://vblogza.blogspot.com/2009/06/open-cv-1.html

[12]http://javawork.exteen.com/20090110/start-opencv-c

[13]http://th.answers.yahoo.com/question/index?qid=20090523084409AAgq3Rw

[14] http://my.sony.co.th/tips/097_CyberShot/tipn097_th.html

[15]http://www.ce.kmitl.ac.th/subject.php?action=view&SUBJECT_ID=7

[16]http://pornpisit.blogspot.com/2009/06/blog-post_11.html

[17] http://www.cs.unc.edu/Research/stc/FAQs/OpenCV/OpenCVReferenceManual.pdf

[18]http://www.barrythomas.co.uk/code.html

[19]http://docs.opencv.org/

[20]http://www.ibookengineering.com/index.php?option=com_content&view=article&id=63&Itemid=11

[21]cdn.learners.in.th/assets/media/files/000/061/269/

[22]http://www.thaicreate.com/mobile/basic-android.html

[23]http://www.datasprings.com/resources/articles-information/android-sdk-example-application-sample-code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง Mode การทำงาน

1	เปิดเล่นเสียงเพลง
2	หยุดเล่นเสียงเพลง
3	ส่งข้อความ SMS "**** Call Me****"
4	ส่งข้อความ SMS "**** Help Me ****"
5	เปิด-ปิด เครื่องใช้ไฟฟ้า ช่อง 1
6	เปิด-ปิด เครื่องใช้ไฟฟ้า ช่อง 2
7	เปิด-ปิด เครื่องใช้ไฟฟ้า ช่อง 3
8	เปิด-ปิด เครื่องใช้ไฟฟ้า ช่อง 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sample2NativeCamera.java

```
1  package org.opencv.samples.tutorial2;
2
3  import java.io.IOException;
4  import java.io.OutputStream;
5  import java.util.UUID;
6
7  import org.opencv.android.BaseLoaderCallback;
8  import org.opencv.android.LoaderCallbackInterface;
9  import org.opencv.android.OpenCVLoader;
10 import android.app.Activity;
11 import android.app.AlertDialog;
12 import android.app.ProgressDialog;
13 import android.bluetooth.BluetoothAdapter;
14 import android.bluetooth.BluetoothDevice;
15 import android.bluetooth.BluetoothSocket;
16 import android.content.Context;
17 import android.content.DialogInterface;
18 import android.content.Intent;
19 import android.media.MediaPlayer;
20 import android.os.Bundle;
21 import android.os.Handler;
22 import android.os.Message;
23 import android.os.Vibrator;
24 import android.telephony.gsm.SmsManager;
25 import android.util.Log;
26 import android.view.Menu;
27 import android.view.MenuItem;
28 import android.view.MotionEvent;
29 import android.view.Window;
30 import android.view.WindowManager;
31 import android.widget.Toast;
32
33
34 public class Sample2NativeCamera extends Activity {
35
36     private static final String TAG              =
        "Sample::Activity";
37
38     // Intent request codes
39     private static final int REQUEST_CONNECT_DEVICE = 1;
40     private static final int REQUEST_ENABLE_BT = 2;
41
42     // Bluetooth Stuff
43     private BluetoothAdapter mBluetoothAdapter = null;
44     private BluetoothSocket btSocket = null;
45     private OutputStream mOutStream = null;
46     private ConnectThread mConnectThread = null;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

```
47      private String deviceAddress = null;
48      private boolean connectStat = false;
49      // Well known SPP UUID (will *probably* map to
        RFCOMM channel 1 (default) if not in use);
50      private static final UUID SPP_UUID = UUID.
        fromString("00001101-0000-1000-8000-00805F9B34FB");
51
52      private byte DATAOut;
53
54      // Mode from the Sample2View
55      public static final int     VIEW_MODE_RGBA  = 0;
56      public static final int     VIEW_MODE_GRAY  = 1;
57      public static final int     VIEW_MODE_CHANNELS = 2;
58      public static final int
        VIEW_MODE_CHANNELS_CIRCLE_TEST = 3;
59      public static final int     VIEW_MODE_LAST = 4;
60      public static final int     VIEW_MODE_FULL = 5;
61
62      public static int           viewMode        =
        VIEW_MODE_RGBA;
63
64      //Setting to Send SMS
65      private String phoneNo1 = "0895189251";
66  //    private String message = "Hello world";
67      private String message1 = "**** Call Me ****";
68      private String message2 = "**** Help Me ****";
69
70
71      private MenuItem            mItemPreviewRGBA;
72      private MenuItem            mItemPreviewGray;
73      private MenuItem            mItemPreviewChannels;
74      private MenuItem            mItemPreviewscircletest;
75      private MenuItem            mItemPreviewlast;
76      private MenuItem            mItemPreviewfull;
77      private MenuItem
        mItemPreviewConnectBluetooth;
78
79      private Sample2View         mView;
80      private MediaPlayer         mPlayer;
81      private Vibrator            mVibrator;
82      private ProgressDialog      myProgressDialog;
83
84      private BaseLoaderCallback  mOpenCVCallBack = new
        BaseLoaderCallback(this) {
85          @Override
86          public void onManagerConnected(int status) {
87              switch (status) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sample2NativeCamera.java

```
88                  case LoaderCallbackInterface.SUCCESS:
89                  {
90                      Log.i(TAG, "OpenCV loaded
                        successfully");
91                      // Create and set View
92                      mView = new Sample2View(mAppContext
                        , mHandler);
93                      setContentView(mView);
94                      // Check native OpenCV camera
95                      if( !mView.openCamera() ) {
96                          AlertDialog ad = new
                            AlertDialog.Builder(mAppContext
                            ).create();
97                          ad.setCancelable(false); //
                            This blocks the 'BACK' button
98                          ad.setMessage("Fatal error:
                            can't open camera!");
99                          ad.setButton("OK", new
                            DialogInterface.OnClickListener
                            () {
100                             public void onClick(
                                DialogInterface dialog, int
                                 which) {
101                             dialog.dismiss();
102                             finish();
103                             }
104                         });
105                         ad.show();
106                     }
107                 } break;
108                 default:
109                 {
110                     super.onManagerConnected(status);
111                 } break;
112             }
113         }
114     };
115
116     public Sample2NativeCamera() {
117         Log.i(TAG, "Instantiated new " + this.getClass
            ());
118     }
119
120     /** Called when the activity is first created. */
121     @Override
122     public void onCreate(Bundle savedInstanceState) {
123         Log.d(TAG, "onCreate");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
124             super.onCreate(savedInstanceState);
125
126             requestWindowFeature(Window.FEATURE_NO_TITLE);
127             getWindow().setFlags(WindowManager.LayoutParams
                .FLAG_KEEP_SCREEN_ON, WindowManager.
                LayoutParams.FLAG_KEEP_SCREEN_ON);
128             myProgressDialog = new ProgressDialog(this);
129
130             Log.i(TAG, "Trying to load OpenCV library");
131             if (!OpenCVLoader.initAsync(OpenCVLoader.
                OPENCV_VERSION_2_4_2, this, mOpenCVCallBack))
132             {
133                 Log.e(TAG, "Cannot connect to OpenCV
                    Manager");
134             }
135
136             // Get local Bluetooth adapter
137             mBluetoothAdapter = BluetoothAdapter.
                getDefaultAdapter();
138
139             // If the adapter is null, then Bluetooth is
                not supported
140             if (mBluetoothAdapter == null) {
141                 Toast.makeText(this, "Bluetooth is not
                    available", Toast.LENGTH_LONG).show();
142                 finish();
143                 return;
144             }
145         }
146
147         @Override
148         public void onStart() {
149             Log.d(TAG, "onStart");
150             super.onStart();
151
152             Log.i(TAG, "Vibrator");
153             mVibrator = (Vibrator)getSystemService(Context.
                VIBRATOR_SERVICE);
154             Log.i(TAG, "MediaPlayer");
155             mPlayer = new MediaPlayer();
156
157             // If BT is not on, request that it be enabled.
158             // setupChat() will then be called during
                onActivityResult
159             if (!mBluetoothAdapter.isEnabled()) {
160                 Intent enableIntent = new Intent(
                    BluetoothAdapter.ACTION_REQUEST_ENABLE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
161                 startActivityForResult(enableIntent,
                    REQUEST_ENABLE_BT);
162             // Otherwise, setup the chat session
163             }
164         }
165
166         @Override
167         protected void onResume() {
168             Log.d(TAG, "onResume");
169             super.onResume();
170
171             if((null != mView) && !mView.openCamera() ) {
172                 AlertDialog ad = new AlertDialog.Builder(
                    this).create();
173                 ad.setCancelable(false); // This blocks
                    the 'BACK' button
174                 ad.setMessage("Fatal error: can't open
                    camera!");
175                 ad.setButton("OK", new DialogInterface.
                    OnClickListener() {
176                     public void onClick(DialogInterface
                        dialog, int which) {
177                         dialog.dismiss();
178                         finish();
179                     }
180                 });
181                 ad.show();
182             }
183         }
184
185         @Override
186         protected void onPause() {
187             Log.d(TAG, "onPause");
188             super.onPause();
189
190             if (mPlayer != null){
191                 mPlayer.stop();
192                 mPlayer.release();
193                 mPlayer = null;
194             }
195         }
196
197         @Override
198         public void onStop() {
199             Log.d(TAG, "onStop");
200             super.onStop();
201
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
202             if (null != mView)
203                 mView.releaseCamera();
204         }
205
206         @Override
207         public void onDestroy() {
208             Log.d(TAG, "onDestroy");
209             super.onDestroy();
210
211             emptyOutStream();
212             disconnect();
213         }
214
215         private final Handler mHandler = new Handler() {
216
217             private boolean Status_ch1 = true;
218             private boolean Status_ch2 = true;
219             private boolean Status_ch3 = true;
220             private boolean Status_ch4 = true;
221             @Override
222             public void handleMessage(Message msg) {
223                 Log.d(TAG, "handleMessage case " + msg.what
                    );
224                 if (myProgressDialog.isShowing()) {
225                     myProgressDialog.dismiss();
226                  }
227                 switch (msg.what) {
228                 case 1:
229                     Toast.makeText(getBaseContext(), "Play
                        Song", Toast.LENGTH_SHORT).show();
230                     if(mPlayer!=null){
231 //                      mPlayer.stop();
232                         mPlayer.release();
233                     }
234                     mPlayer = MediaPlayer.create(
                        getBaseContext(), R.raw.main);
235                     mPlayer.setLooping(true);
236                     mPlayer.start();
237                     Sample2View.return_handler = false;
238                     break;
239                 case 2:
240                     if(mPlayer!=null){
241                         Toast.makeText(getBaseContext(),
                            "Stop Song", Toast.LENGTH_SHORT).
                            show();
242 //                      mPlayer.stop();
243                         mPlayer.release();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
244                     }
245                     Sample2View.return_handler = false;
246                     break;
247                 case 3:
248                     Toast.makeText(getBaseContext(),
                        "sendSMS " +message1, Toast.
                        LENGTH_SHORT).show();
249
250                     sendSMS(phoneNo1, message1);
251                     Sample2View.return_handler = false;
252                     break;
253                 case 4:
254                     Toast.makeText(getBaseContext(),
                        "sendSMS " +message2, Toast.
                        LENGTH_SHORT).show();
255
256                     sendSMS(phoneNo1, message2);
257                     Sample2View.return_handler = false;
258                     break;
259                 case 5:
260                     if(Status_ch1){
261                         Toast.makeText(getBaseContext(),
                            "CH1 NO", Toast.LENGTH_SHORT).show
                            ();
262                         DATAOut = '1';
263                         Status_ch1 = false;
264                     }
265                     else{
266                         Toast.makeText(getBaseContext(),
                            "CH1 OFF", Toast.LENGTH_SHORT).show
                            ();
267                         DATAOut = '4';
268                         Status_ch1 = true;
269                     }
270                     write(DATAOut);
271                     Sample2View.return_handler = false;

272                     break;
273                 case 6:
274                     if(Status_ch2){
275                         Toast.makeText(getBaseContext(),
                            "CH2 NO", Toast.LENGTH_SHORT).show
                            ();
276                         DATAOut = '2';
277                         Status_ch2 = false;
278                     }
279                     else{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
280                         Toast.makeText(getBaseContext(),
                            "CH2 OFF", Toast.LENGTH_SHORT).show
                            ();
281                         DATAOut = '5';
282                         Status_ch2 = true;
283                     }
284                     write(DATAOut);
285                     Sample2View.return_handler = false;
286                     break;
287                 case 7:
288                     if(Status_ch3){
289                         Toast.makeText(getBaseContext(),
                            "CH3 NO", Toast.LENGTH_SHORT).show
                            ();
290                         DATAOut = '3';
291                         Status_ch3 = false;
292                     }
293                     else{
294                         Toast.makeText(getBaseContext(),
                            "CH3 OFF", Toast.LENGTH_SHORT).show
                            ();
295                         DATAOut = '6';
296                         Status_ch3 = true;
297                     }
298                     write(DATAOut);
299                     Sample2View.return_handler = false;
300                     break;
301                 case 8:
302                     if(Status_ch4){
303                         Toast.makeText(getBaseContext(),
                            "CH4 NO", Toast.LENGTH_SHORT).show
                            ();
304                         DATAOut = '7';
305                         Status_ch4 = false;
306                     }
307                     else{
308                         Toast.makeText(getBaseContext(),
                            "CH4 OFF", Toast.LENGTH_SHORT).show
                            ();
309                         DATAOut = '8';
310                         Status_ch4 = true;
311                     }
312                     write(DATAOut);
313                     Sample2View.return_handler = false;
314                     break;
315                 case 'a':
316                     Toast.makeText(getBaseContext(),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    "Connected to device.", Toast.
                    LENGTH_SHORT).show();
317
318                 // Set button to display current status
319                 connectStat = true;
320
321                 // Reset the DATAOut
322                 DATAOut = 0;
323                 write(DATAOut);
324                 break;
325             case 'b':
326                 Toast.makeText(getBaseContext(),
                    "Failed to connect.", Toast.
                    LENGTH_SHORT).show();
327                 break;
328             }
329         }
330     };
331
332     @Override
333     public boolean onCreateOptionsMenu(Menu menu) {
334         Log.d(TAG, "onCreateOptionsMenu");
335         mItemPreviewRGBA = menu.add("Preview -RGBA");
336         mItemPreviewGray = menu.add("Preview -GRAY");
337         mItemPreviewChannels = menu.add("Preview
            -BINARY");
338         mItemPreviewscircletest= menu.add("Test
            -CIRCLE,GRID");
339         mItemPreviewlast = menu.add("Preview -LAST");
340         mItemPreviewfull = menu.add("FULL Ver.");
341         mItemPreviewConnectBluetooth = menu.add(
            "CONNECT BLUETOOTH");
342         return true;
343     }
344
345     @Override
346     public boolean onOptionsItemSelected(MenuItem item)
         {
347         Log.d(TAG, "Menu Item selected " + item);
348         if (item == mItemPreviewRGBA){
349             Log.i(TAG, "VIEW_MODE_RGBA");
350             viewMode = VIEW_MODE_RGBA;
351             mVibrator.vibrate(50);
352         }
353         else if (item == mItemPreviewGray){
354             Log.i(TAG, "VIEW_MODE_GRAY");
355             viewMode = VIEW_MODE_GRAY;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
356                 mVibrator.vibrate(50);
357             }
358             else if (item == mItemPreviewChannels){
359                 Log.i(TAG, "VIEW_MODE_CHANNELS");
360                 viewMode = VIEW_MODE_CHANNELS;
361                 mVibrator.vibrate(50);
362             }
363             else if (item == mItemPreviewscircletest){
364                 Log.i(TAG, "VIEW_MODE_CHANNELS_CIRCLE_TEST"
                    );
365                 viewMode = VIEW_MODE_CHANNELS_CIRCLE_TEST;
366                 mVibrator.vibrate(50);
367             }
368             else if (item == mItemPreviewlast){
369                 Log.i(TAG, "VIEW_MODE_LAST");
370                 viewMode = VIEW_MODE_LAST;
371                 mVibrator.vibrate(50);
372             }
373             else if (item == mItemPreviewfull){
374                 Log.i(TAG, "VIEW_MODE_FULL");
375                 viewMode = VIEW_MODE_FULL;
376                 mVibrator.vibrate(50);
377             }
378             else if (item == mItemPreviewConnectBluetooth){
379                 Log.i(TAG, "CONNECT BLUETOOTH");
380                 if (connectStat) {
381                     // Attempt to disconnect from the device
382                     disconnect();
383                 }else{
384                     // Attempt to connect to the device
385                     connect();
386                 }
387                 mVibrator.vibrate(50);
388             }
389             return true;
390         }
391
392         public boolean onTouchEvent(final MotionEvent event
            ) {
393             Log.v(TAG, "vibrate");
394             mVibrator.vibrate(500);
395             Toast.makeText(this, "Save Photo", Toast.
                LENGTH_SHORT).show();
396             Sample2View.bShootNow = true;
397             return true;
398         }
399
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
400     @SuppressWarnings("deprecation")
401     private void sendSMS(String phoneNumber, String
        message)
402     {
403         Log.v(TAG, "phoneNumber " + phoneNumber);
404         Log.v(TAG, "Message " + message);

405         SmsManager sms = SmsManager.getDefault();
406         sms.sendTextMessage(phoneNumber, null, message,
             null, null);
407     }
408
409     public void connect() {
410      // Launch the DeviceListActivity to see devices
        and do scan
411         Intent serverIntent = new Intent(this,
            DeviceListActivity.class);
412         startActivityForResult(serverIntent,
            REQUEST_CONNECT_DEVICE);
413     }
414
415     public void disconnect() {
416      if (mOutStream != null) {
417          try {
418                  mOutStream.close();
419                  Toast.makeText(getBaseContext(),
                     "Disconnected to device.", Toast.
                     LENGTH_SHORT).show();
420                  connectStat = false;
421              } catch (IOException e) {
422              }
423      }
424     }
425
426     /** Thread used to connect to a specified
        Bluetooth Device */
427     public class ConnectThread extends Thread {
428     private String address;
429     private boolean connectionStatus;
430
431         ConnectThread(String MACaddress) {
432             address = MACaddress;
433             connectionStatus = true;
434         }
435
436         public void run() {
437             // When this returns, it will 'know' about the
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            server,
438             // via it's MAC address.
439              try {
440                  BluetoothDevice device =
                     mBluetoothAdapter.getRemoteDevice(
                     address);
441
442                  // We need two things before we can
                     successfully connect
443                  // (authentication issues aside): a
                     MAC address, which we
444                  // already have, and an RFCOMM channel.
445                  // Because RFCOMM channels (aka ports)
                     are limited in
446                  // number, Android doesn't allow you
                     to use them directly;
447                  // instead you request a RFCOMM
                     mapping based on a service
448                  // ID. In our case, we will use the
                     well-known SPP Service
449                  // ID. This ID is in UUID (GUID to you
                     Microsofties)
450                  // format. Given the UUID, Android
                     will handle the
451                  // mapping for you. Generally, this
                     will return RFCOMM 1,
452                  // but not always; it depends what
                     other BlueTooth services
453                  // are in use on your Android device.
454                  try {
455                      btSocket = device.
                         createRfcommSocketToServiceRecord(
                         SPP_UUID);
456                  } catch (IOException e) {
457                      connectionStatus = false;
458                  }
459              }catch (IllegalArgumentException e) {
460                  connectionStatus = false;
461              }
462
463             // Discovery may be going on, e.g., if
                you're running a
464             // 'scan for devices' search from your
                handset's Bluetooth
465             // settings, so we call cancelDiscovery().
                It doesn't hurt
466             // to call it, but it might hurt not to...
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                discovery is a
467             // heavyweight process; you don't want it
                in progress when
468             // a connection attempt is made.
469             mBluetoothAdapter.cancelDiscovery();
470
471             // Blocking connect, for a simple client
                nothing else can
472             // happen until a successful connection is
                made, so we
473             // don't care if it blocks.
474             try {
475                 btSocket.connect();
476             } catch (IOException e1) {
477                 try {
478                     btSocket.close();
479                 } catch (IOException e2) {
480                 }
481             }
482
483             // Create a data stream so we can talk to
                server.
484             try {
485              mOutStream = btSocket.getOutputStream();
486             } catch (IOException e2) {
487              connectionStatus = false;
488             }
489
490             // Send final result
491             if (connectionStatus) {
492              mHandler.sendEmptyMessage('a'); //true(1)
493             }else {
494              mHandler.sendEmptyMessage('b'); //false(0)
495             }
496         }
497     }
498     public void onActivityResult(int requestCode, int
        resultCode, Intent data) {
499      switch (requestCode) {
500         case REQUEST_CONNECT_DEVICE:
501          // When DeviceListActivity returns with a
             device to connect
502             if (resultCode == Activity.RESULT_OK) {
503                 // Show please wait dialog
504                 myProgressDialog = ProgressDialog.show(
                    this, "Please Wait", "Connecting to
                    bluetooth device...", true);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
505
506                     // Get the device MAC address
507                     deviceAddress = data.getExtras().
                        getString(DeviceListActivity.
                        EXTRA_DEVICE_ADDRESS);
508                     // Connect to device with specified
                        MAC address
509                     mConnectThread = new ConnectThread(
                        deviceAddress);
510                     mConnectThread.start();
511
512                 }else {
513                     // Failure retrieving MAC address
514                     Toast.makeText(this,"Error retrieving
                        MAC address.", Toast.LENGTH_SHORT).show
                        ();
515                 }
516                 break;
517             case REQUEST_ENABLE_BT:
518                 // When the request to enable Bluetooth
                    returns
519                 if (resultCode == Activity.RESULT_OK) {
520                     // Bluetooth is now enabled
521                 } else {
522                     // User did not enable Bluetooth or an
                        error occured
523                     Toast.makeText(this,"Bluetooth was not
                        enabled.", Toast.LENGTH_SHORT).show();
524                     finish();
525                 }
526             }
527         }
528
529         public void write(byte data) {
530          if (mOutStream != null) {
531                 try {
532                  mOutStream.write(data);
533                 } catch (IOException e) {
534                 }
535             }
536         }
537
538         public void emptyOutStream() {
539          if (mOutStream != null) {
540                 try {
541                 mOutStream.flush();
542                 } catch (IOException e) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
543                 }
544             }
545         }
546     }
547
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sample2View.java

```
1  package org.opencv.samples.tutorial2;
2
3  import java.util.List;
4  import java.util.ArrayList;
5  import java.text.SimpleDateFormat;
6
7  import java.sql.Date;
8
9  import org.opencv.android.Utils;
10 import org.opencv.core.Core;
11 import org.opencv.core.Mat;
12 import org.opencv.core.MatOfPoint;
13 import org.opencv.core.MatOfPoint2f;
14 import org.opencv.core.Point;
15 import org.opencv.core.Scalar;
16 import org.opencv.core.Size;
17 import org.opencv.highgui.Highgui;
18 import org.opencv.highgui.VideoCapture;
19 import org.opencv.imgproc.Imgproc;
20
21 import android.content.Context;
22 import android.graphics.Bitmap;
23
24 import android.os.Environment;
25 import android.os.Handler;
26 import android.util.Log;
27 import android.util.DisplayMetrics;
28 import android.view.SurfaceHolder;
29
30 class Sample2View extends SampleCvViewBase {
31     private static final String TAG = "Sample2View";
32
33     private Mat mRgba;
34     private Mat mGray;
35     private Mat mIntermediateMat;
36     private Mat mMatRed;
37     private Mat mMatGreen;
38     private Mat mMatBlue;
39     private Mat mMatRedInv;
40     private Mat mMatGreenInv;
41     private Mat mMatBlueInv;
42
43     private Scalar colorRed, colorGreen, colorBlue;
44     private String string;
45     private Point pt;
46     private final Handler mHandler;
47
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
48      private double dTextScaleFactor;
49
50      public static byte[] byteColourTrackCentreHue;
51      public static int iContourAreaMin = 1000;
52      public static boolean bShootNow = false,
        bDisplayTitle = true, return_handler = false;
53      public static long lFrameCount = 0, lMilliStart = 0
        , lMilliNow = 0;
54
55      public static Size sMatSize;
56      public static List<MatOfPoint> contours;
57      public static MatOfPoint2f mMOP2f;
58      public static Point ptRedTL, ptGreenTL, ptBlueTL,
        ptRedBR, ptGreenBR, ptBlueBR;
59
60      int x, grid_cols, grid_rows, radius, iLineThickness
         = 3, iCannyUpperThreshold, iCannyLowerThreshold,
        iMinRadius, iMaxRadius, iAccumulator, choice;
61
62      public Sample2View(Context context, Handler handler
        ) {
63          super(context);
64          mHandler = handler;
65
66      }
67
68      @Override
69      public void surfaceCreated(SurfaceHolder holder) {
70          Log.i(TAG, "surfaceCreated");
71          synchronized (this) {
72              // initialize Mats before usage
73              mGray = new Mat();
74              mRgba = new Mat();
75              mMOP2f = new MatOfPoint2f();
76              sMatSize = new Size();
77              mMatRed = new Mat();
78              mMatGreen = new Mat();
79              mMatBlue = new Mat();
80              mMatRedInv = new Mat();
81              mMatGreenInv = new Mat();
82              mMatBlueInv = new Mat();
83              mIntermediateMat = new Mat();
84
85              pt = new Point (0, 0);
86              ptRedTL = new Point();
87              ptGreenTL = new Point();
88              ptBlueTL = new Point();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
89                  ptRedBR = new Point();
90                  ptGreenBR = new Point();
91                  ptBlueBR = new Point();
92
93                  contours = new ArrayList<MatOfPoint>();
94
95                  colorRed = new Scalar(255, 0, 0, 255);
96                  colorGreen = new Scalar(0, 255, 0, 255);
97                  colorBlue = new Scalar(0, 0, 255, 255);
98
99                  byteColourTrackCentreHue = new byte[3];
100                 byteColourTrackCentreHue[0] = 27; // mid
                    yellow
101                 byteColourTrackCentreHue[1] = 0; // red
102                 byteColourTrackCentreHue[2] = 60; // green
103
104                 string = "";
105
106                 DisplayMetrics dm = this.getResources().
                    getDisplayMetrics();
107                 int densityDpi = dm.densityDpi;
108                 dTextScaleFactor = ((double)densityDpi /
                    240.0) * 0.3;
109
110             }
111
112             super.surfaceCreated(holder);
113         }
114
115         @Override
116         protected Bitmap processFrame(VideoCapture capture)
             {
117
118 //          Log.i(TAG, "processFrame");
119
120             if (lMilliStart == 0)
121                 lMilliStart = System.currentTimeMillis();
122
123             if ((lMilliNow - lMilliStart) > 10000) {
124                 lMilliStart = System.currentTimeMillis();
125                 lFrameCount = 0;
126                 }
127
128             switch (Sample2NativeCamera.viewMode) {
129             case Sample2NativeCamera.VIEW_MODE_GRAY:
130                 capture.retrieve(mRgba, Highgui.
                    CV_CAP_ANDROID_COLOR_FRAME_RGBA);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
131                 sMatSize.width = mRgba.width();
132                 sMatSize.height = mRgba.height();
133
134                 // extract just one channel
135                 Core.extractChannel(mRgba, mMatRed, 0);
136                 Core.extractChannel(mRgba, mMatGreen, 1);
137                 Core.extractChannel(mRgba, mMatBlue, 2);
138
139                 Core.bitwise_not(mMatRed, mMatRedInv);
140                 Core.bitwise_not(mMatGreen, mMatGreenInv);
141                 Core.bitwise_not(mMatBlue, mMatBlueInv);
142
143                 Core.multiply(mMatRed, mMatGreenInv,
                    mIntermediateMat, (double)1/255);
144                 Core.multiply(mIntermediateMat, mMatBlueInv
                    , mMatRed, (double)1/255);
145                 Imgproc.cvtColor(mMatRed, mRgba, Imgproc.
                    COLOR_GRAY2RGBA, 4);
146
147                 if (bDisplayTitle) {
148                     ShowTitle ("Nopmonkol (Preview GRAY)",
                        1);
149                 }
150                 break;
151             case Sample2NativeCamera.VIEW_MODE_RGBA:
152                 capture.retrieve(mRgba, Highgui.
                    CV_CAP_ANDROID_COLOR_FRAME_RGBA);
153
154                 if (bDisplayTitle) {
155                     ShowTitle ("Nopmonkol (Preview RGBA)",
                        1);
156                 }
157                 break;
158             case Sample2NativeCamera.VIEW_MODE_CHANNELS:
159                 capture.retrieve(mRgba, Highgui.
                    CV_CAP_ANDROID_COLOR_FRAME_RGBA);
160                 sMatSize.width = mRgba.width();
161                 sMatSize.height = mRgba.height();
162
163                 // extract just one channel
164                 Core.extractChannel(mRgba, mMatRed, 0);
165                 Core.extractChannel(mRgba, mMatGreen, 1);
166                 Core.extractChannel(mRgba, mMatBlue, 2);
167
168                 Core.bitwise_not(mMatRed, mMatRedInv);
169                 Core.bitwise_not(mMatGreen, mMatGreenInv);
170                 Core.bitwise_not(mMatBlue, mMatBlueInv);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
171
172             Core.multiply(mMatRed, mMatGreenInv,
                mIntermediateMat, (double)1/255);
173             Core.multiply(mIntermediateMat, mMatBlueInv
                , mMatRed, (double)1/255);
174
175             Imgproc.GaussianBlur(mMatRed, mMatRed, new
                Size(5, 5), 2, 2);
176
177             iCannyLowerThreshold = 35;
178             iCannyUpperThreshold = 100;
179
180             Imgproc.Canny(mMatRed, mIntermediateMat,
                iCannyLowerThreshold, iCannyUpperThreshold);
181             Imgproc.cvtColor(mIntermediateMat, mRgba,
                Imgproc.COLOR_GRAY2BGRA, 4);
182
183             if (bDisplayTitle) {
184                 ShowTitle ("Nopmonkol (Preview green
                    color -COLCONTOUR)", 1);
185             }
186             break;
187         case Sample2NativeCamera.
            VIEW_MODE_CHANNELS_CIRCLE_TEST:
188             capture.retrieve(mRgba, Highgui.
                CV_CAP_ANDROID_COLOR_FRAME_RGBA);
189             sMatSize.width = mRgba.width();
190             sMatSize.height = mRgba.height();
191
192             // extract just one channel
193             Core.extractChannel(mRgba, mMatRed, 0);
194             Core.extractChannel(mRgba, mMatGreen, 1);
195             Core.extractChannel(mRgba, mMatBlue, 2);
196
197             Core.bitwise_not(mMatRed, mMatRedInv);
198             Core.bitwise_not(mMatGreen, mMatGreenInv);
199             Core.bitwise_not(mMatBlue, mMatBlueInv);
200
201             Core.multiply(mMatRed, mMatGreenInv,
                mIntermediateMat, (double)1/255);
202             Core.multiply(mIntermediateMat, mMatBlueInv
                , mMatRed, (double)1/255);
203
204             Imgproc.GaussianBlur(mMatRed, mMatRed, new
                Size(5, 5), 2, 2);
205
206             iCannyLowerThreshold = 35;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
207                 iCannyUpperThreshold = 100;
208                 iAccumulator = 10;
209                 iMinRadius = 1;
210                 iMaxRadius = 20;
211
212                 Imgproc.Canny(mMatRed, mGray,
                    iCannyLowerThreshold, iCannyUpperThreshold);
213
214                 Imgproc.cvtColor(mGray, mRgba, Imgproc.
                    COLOR_GRAY2BGRA, 4);
215
216                 Imgproc.HoughCircles(mGray,
                    mIntermediateMat, Imgproc.CV_HOUGH_GRADIENT
                    , 2, mGray.rows() / 8,
217                         iCannyUpperThreshold, iAccumulator,
                             iMinRadius, iMaxRadius);
218
219                 if (mIntermediateMat.cols() > 0)
220                     for (int x = 0; x < 1; x++) {
221                         double vCircle[] = mIntermediateMat
                        .get(0,x);
222
223                         if (vCircle == null)
224 //                          Log.i(TAG, "vCircle == null");
225                             break;
226
227                         pt = new Point(Math.round(vCircle[0
                        ]), Math.round(vCircle[1]));
228                         radius = (int)Math.round(vCircle[2
                        ]);
229                         // draw the found circle
230                         Core.circle(mRgba, pt, radius,
                        colorRed, iLineThickness);
231
232                         // draw a cross on the center of
                        the circle
233                         DrawCross (mRgba, pt);
234
235                         //draw Grid
236                         grid_cols = (int)sMatSize.width / (
                        radius*3);
237                         grid_rows = (int)sMatSize.height /
                        (radius*3);
238
239                         drawGrid((int)sMatSize.width, (int)
                        sMatSize.height, grid_cols,
                        grid_rows);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
240                 }
241
242             if (bDisplayTitle) {
243                 ShowTitle ("Nopmonkol (Detect red
                    color -CHANNELS)", 1);
244                 string = String.format("position X: %d
                    position Y: %d", (int)pt.x, (int)pt.y);
245                 ShowTitle (string, 3);
246                 string = String.format("Grid Area: %d
                    Grid Size: %d", grid_cols*grid_rows,
                    radius*3);
247                 ShowTitle (string, 4);
248             }
249
250             break;
251         case Sample2NativeCamera.VIEW_MODE_LAST:
252
253             capture.retrieve(mRgba, Highgui.
                CV_CAP_ANDROID_COLOR_FRAME_RGBA);
254             sMatSize.width = mRgba.width();
255             sMatSize.height = mRgba.height();
256
257             // extract just one channel
258             Core.extractChannel(mRgba, mMatRed, 0);
259             Core.extractChannel(mRgba, mMatGreen, 1);
260             Core.extractChannel(mRgba, mMatBlue, 2);
261
262             Core.bitwise_not(mMatRed, mMatRedInv);
263             Core.bitwise_not(mMatGreen, mMatGreenInv);
264             Core.bitwise_not(mMatBlue, mMatBlueInv);
265
266             Core.multiply(mMatRed, mMatGreenInv,
                mIntermediateMat, (double)1/255);
267             Core.multiply(mIntermediateMat, mMatBlueInv
                , mMatRed, (double)1/255);
268
269             Imgproc.GaussianBlur(mMatRed, mMatRed, new
                Size(5, 5), 2, 2);
270
271             iCannyLowerThreshold = 35;
272             iCannyUpperThreshold = 100;
273             iAccumulator = 10;
274             iMinRadius = 1;
275             iMaxRadius = 20;
276
277             //draw Grid
278             grid_cols = 4;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
279                 grid_rows = 2;
280
281                 Imgproc.Canny(mMatRed, mGray,
                    iCannyLowerThreshold, iCannyUpperThreshold);
282
283                 Imgproc.cvtColor(mGray, mRgba, Imgproc.
                    COLOR_GRAY2BGRA, 4);
284
285                 Imgproc.HoughCircles(mGray,
                    mIntermediateMat, Imgproc.CV_HOUGH_GRADIENT
                    , 2, mGray.rows() / 8,
286                         iCannyUpperThreshold, iAccumulator,
                             iMinRadius, iMaxRadius);
287
288                 if (mIntermediateMat.cols() > 0)
289                     for (int x = 0; x < 1; x++) {
290                         double vCircle[] = mIntermediateMat
                        .get(0,x);
291
292                         if (vCircle == null){
293 //                          Log.i(TAG, "vCircle == null");
294                             pt = new Point(0,0);
295                             if(return_handler == true){
296                                 Log.i(TAG, "return_handler
                                =" +choice);
297                                 mHandler.sendEmptyMessage(
                                    choice);
298                             }
299                             break;
300                         }
301
302                         pt = new Point(Math.round(vCircle[0
                        ]), Math.round(vCircle[1]));
303                         radius = (int)Math.round(vCircle[2
                        ]);
304                         // draw the found circle
305                         Core.circle(mRgba, pt, radius,
                        colorRed, iLineThickness);
306
307                         // draw a cross on the center of
                        the circle
308                         DrawCross (mRgba, pt);
309
310                         ////////////////////////////////////
                            ////////////////////////////////////
                            ////////////////////////////////////
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

```
311
312     //                    (int)(sMatSize.width/grid_cols)=80
        (X)
313     //
        (int)(sMatSize.height/grid_rows)=120 (Y)
314
315                           if( ((int)pt.x>(0*(int)(sMatSize.
                              width/grid_cols)))&&((int)pt.x<(1*(
                              int)(sMatSize.width/grid_cols))) &&
                               ((int)pt.y>(0*(int)(sMatSize.
                              height/grid_rows)))&&(pt.y<(1*(int
                              )(sMatSize.height/grid_rows))) ){
316                               choice = 4;
317                               return_handler = true;
318                           }
319                           else if( ((int)pt.x>(1*(int)(
                              sMatSize.width/grid_cols)))&&((int)
                              pt.x<(2*(int)(sMatSize.width/
                              grid_cols))) && ((int)pt.y>(0*(int
                              )(sMatSize.height/grid_rows)))&&(pt
                              .y<(1*(int)(sMatSize.height/
                              grid_rows))) ){
320                               choice = 3;
321                               return_handler = true;
322                           }
323                           else if( ((int)pt.x>(2*(int)(
                              sMatSize.width/grid_cols)))&&((int)
                              pt.x<(3*(int)(sMatSize.width/
                              grid_cols))) && ((int)pt.y>(0*(int
                              )(sMatSize.height/grid_rows)))&&(pt
                              .y<(1*(int)(sMatSize.height/
                              grid_rows))) ){
324                               choice = 2;
325                               return_handler = true;
326                           }
327                           else if( ((int)pt.x>(3*(int)(
                              sMatSize.width/grid_cols)))&&((int)
                              pt.x<(4*(int)(sMatSize.width/
                              grid_cols))) && ((int)pt.y>(0*(int
                              )(sMatSize.height/grid_rows)))&&(pt
                              .y<(1*(int)(sMatSize.height/
                              grid_rows))) ){
328                               choice = 1;
329                               return_handler = true;
330                           }
331
332                           else if( ((int)pt.x>(0*(int)(
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    sMatSize.width/grid_cols)))&&((int)
                    pt.x<(1*(int)(sMatSize.width/
                    grid_cols))) && ((int)pt.y>(1*(int
                    )(sMatSize.height/grid_rows)))&&(pt
                    .y<(2*(int)(sMatSize.height/
                    grid_rows))) ){
333                     choice = 8;
334                     return_handler = true;
335                 }
336                 else if( ((int)pt.x>(1*(int)(
                    sMatSize.width/grid_cols)))&&((int)
                    pt.x<(2*(int)(sMatSize.width/
                    grid_cols))) && ((int)pt.y>(1*(int
                    )(sMatSize.height/grid_rows)))&&(pt
                    .y<(2*(int)(sMatSize.height/
                    grid_rows))) ){
337                     choice = 7;
338                     return_handler = true;
339                 }
340                 else if( ((int)pt.x>(2*(int)(
                    sMatSize.width/grid_cols)))&&((int)
                    pt.x<(3*(int)(sMatSize.width/
                    grid_cols))) && ((int)pt.y>(1*(int
                    )(sMatSize.height/grid_rows)))&&(pt
                    .y<(2*(int)(sMatSize.height/
                    grid_rows))) ){
341                     choice = 6;
342                     return_handler = true;
343                 }
344                 else if( ((int)pt.x>(3*(int)(
                    sMatSize.width/grid_cols)))&&((int)
                    pt.x<(4*(int)(sMatSize.width/
                    grid_cols))) && ((int)pt.y>(1*(int
                    )(sMatSize.height/grid_rows)))&&(pt
                    .y<(2*(int)(sMatSize.height/
                    grid_rows))) ){
345                     choice = 5;
346                     return_handler = true;
347                 }
348                 else choice = 0;
349             }
350
                ////////////////////////////////////////////
                ////////////////////////////////////////////
                ////////////////////
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
351
352             if (bDisplayTitle) {
```

Sample2View.java

```
353                     ShowTitle ("Nopmonkol (Detect red
                        color -CHANNELS)", 1);
354                     string = String.format("position X: %d
                        position Y: %d", (int)pt.x, (int)pt.y);
355                     ShowTitle (string, 3);
356
357                     string = String.format("Grid Area: %d "
                        , choice);
358                     ShowTitle (string, 4);
359
360                     drawGrid((int)sMatSize.width, (int)
                        sMatSize.height, grid_cols, grid_rows);
361                 }
362
363                 break;
364             case Sample2NativeCamera.VIEW_MODE_FULL:
365                 capture.retrieve(mRgba, Highgui.
                    CV_CAP_ANDROID_COLOR_FRAME_RGBA);
366                 sMatSize.width = mRgba.width();
367                 sMatSize.height = mRgba.height();
368
369                 // extract just one channel
370                 Core.extractChannel(mRgba, mMatRed, 0);
371                 Core.extractChannel(mRgba, mMatGreen, 1);
372                 Core.extractChannel(mRgba, mMatBlue, 2);
373
374                 Core.bitwise_not(mMatRed, mMatRedInv);
375                 Core.bitwise_not(mMatGreen, mMatGreenInv);
376                 Core.bitwise_not(mMatBlue, mMatBlueInv);
377
378                 Core.multiply(mMatRed, mMatGreenInv,
                    mIntermediateMat, (double)1/255);
379                 Core.multiply(mIntermediateMat, mMatBlueInv
                    , mMatRed, (double)1/255);
380
381                 Imgproc.GaussianBlur(mMatRed, mMatRed, new
                    Size(5, 5), 2, 2);
382
383                 iCannyLowerThreshold = 35;
384                 iCannyUpperThreshold = 100;
385                 iAccumulator = 10;
386                 iMinRadius = 1;
387                 iMaxRadius = 20;
388
389                 //draw Grid
390                 grid_cols = 4;
391                 grid_rows = 2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
392
393             Imgproc.Canny(mMatRed, mGray,
                iCannyLowerThreshold, iCannyUpperThreshold);
394
395             Imgproc.cvtColor(mGray, mRgba, Imgproc.
                COLOR_GRAY2BGRA, 4);
396
397             Imgproc.HoughCircles(mGray,
                mIntermediateMat, Imgproc.CV_HOUGH_GRADIENT
                , 2, mGray.rows() / 8,
398                     iCannyUpperThreshold, iAccumulator,
                         iMinRadius, iMaxRadius);
399
400             if (mIntermediateMat.cols() > 0)
401                 for (int x = 0; x < 1; x++) {
402                     double vCircle[] = mIntermediateMat
                        .get(0,x);
403
404                     if (vCircle == null){
405 //                      Log.i(TAG, "vCircle == null");
406                         pt = new Point(0,0);
407                         if(return_handler == true){
408                             Log.i(TAG, "return_handler
                                =" +choice);
409                             mHandler.sendEmptyMessage(
                                choice);
410                         }
411                         break;
412                     }
413
414                     pt = new Point(Math.round(vCircle[0
                        ]), Math.round(vCircle[1]));
415                     radius = (int)Math.round(vCircle[2
                        ]);
416                     // draw the found circle
417                     Core.circle(mRgba, pt, radius,
                        colorRed, iLineThickness);
418
419                     // draw a cross on the center of
                        the circle
420                     DrawCross (mRgba, pt);
421
422
                        ////////////////////////////////////
                        ////////////////////////////////////
                        ////////////////////////////////////
423
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

```
424    //                    (int)(sMatSize.width/grid_cols)=80
       (X)
425    //
       (int)(sMatSize.height/grid_rows)=120 (Y)
426
427                          if( ((int)pt.x>(0*(int)(sMatSize.
                             width/grid_cols)))&&((int)pt.x<(1*(
                             int)(sMatSize.width/grid_cols))) &&
                              ((int)pt.y>(0*(int)(sMatSize.
                             height/grid_rows)))&&(pt.y<(1*(int
                             )(sMatSize.height/grid_rows))) ){
428                              choice = 4;
429                              return_handler = true;
430                          }
431                          else if( ((int)pt.x>(1*(int)(
                             sMatSize.width/grid_cols)))&&((int)
                             pt.x<(2*(int)(sMatSize.width/
                             grid_cols))) && ((int)pt.y>(0*(int
                             )(sMatSize.height/grid_rows)))&&(pt
                             .y<(1*(int)(sMatSize.height/
                             grid_rows))) ){
432                              choice = 3;
433                              return_handler = true;
434                          }
435                          else if( ((int)pt.x>(2*(int)(
                             sMatSize.width/grid_cols)))&&((int)
                             pt.x<(3*(int)(sMatSize.width/
                             grid_cols))) && ((int)pt.y>(0*(int
                             )(sMatSize.height/grid_rows)))&&(pt
                             .y<(1*(int)(sMatSize.height/
                             grid_rows))) ){
436                              choice = 2;
437                              return_handler = true;
438                          }
439                          else if( ((int)pt.x>(3*(int)(
                             sMatSize.width/grid_cols)))&&((int)
                             pt.x<(4*(int)(sMatSize.width/
                             grid_cols))) && ((int)pt.y>(0*(int
                             )(sMatSize.height/grid_rows)))&&(pt
                             .y<(1*(int)(sMatSize.height/
                             grid_rows))) ){
440                              choice = 1;
441                              return_handler = true;
442                          }
443
444                          else if( ((int)pt.x>(0*(int)(
                             sMatSize.width/grid_cols)))&&((int)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    pt.x<(1*(int)(sMatSize.width/
                    grid_cols))) && ((int)pt.y>(1*(int
                    )(sMatSize.height/grid_rows)))&&(pt
                    .y<(2*(int)(sMatSize.height/
                    grid_rows))) ){
445                     choice = 3;
446                     return_handler = true;
447                 }
448                 else if( ((int)pt.x>(1*(int)(
                    sMatSize.width/grid_cols)))&&((int)
                    pt.x<(2*(int)(sMatSize.width/
                    grid_cols))) && ((int)pt.y>(1*(int
                    )(sMatSize.height/grid_rows)))&&(pt
                    .y<(2*(int)(sMatSize.height/
                    grid_rows))) ){
449                     choice = 2;
450                     return_handler = true;
451                 }
452                 else if( ((int)pt.x>(2*(int)(
                    sMatSize.width/grid_cols)))&&((int)
                    pt.x<(3*(int)(sMatSize.width/
                    grid_cols))) && ((int)pt.y>(1*(int
                    )(sMatSize.height/grid_rows)))&&(pt
                    .y<(2*(int)(sMatSize.height/
                    grid_rows))) ){
453                     choice = 6;
454                     return_handler = true;
455                 }
456                 else if( ((int)pt.x>(3*(int)(
                    sMatSize.width/grid_cols)))&&((int)
                    pt.x<(4*(int)(sMatSize.width/
                    grid_cols))) && ((int)pt.y>(1*(int
                    )(sMatSize.height/grid_rows)))&&(pt
                    .y<(2*(int)(sMatSize.height/
                    grid_rows))) ){
457                     choice = 5;
458                     return_handler = true;
459                 }
460                 else choice = 0;
461             }
462
            ////////////////////////////////////////////
            ////////////////////////////////////////////
            ////////////////////
463
464         if (bDisplayTitle) {
465             ShowTitle ("Nopmonkol (Detect red
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    color -CHANNELS)", 1);
466                 string = String.format("position X: %d
                    position Y: %d", (int)pt.x, (int)pt.y);
467                 ShowTitle (string, 3);
468
469                 string = String.format("Grid Area: %d "
                    , choice);
470                 ShowTitle (string, 4);
471
472                 drawGrid((int)sMatSize.width, (int)
                    sMatSize.height, grid_cols, grid_rows);
473             }
474
475             break;
476         }
477
478         // get the time now in every frame
479         lMilliNow = System.currentTimeMillis();
480
481         // update the frame counter
482         lFrameCount++;
483
484         if (bDisplayTitle) {
485             string = String.format("FPS: %2.1f", (float
                )(lFrameCount * 1000) / (float)(lMilliNow -
                 lMilliStart));
486             ShowTitle (string, 2);
487             }
488
489         if (bShootNow) {
490             bShootNow = false;
491             SaveImage (mRgba);
492             }
493
494         Bitmap bmp = Bitmap.createBitmap(mRgba.cols(),
            mRgba.rows(), Bitmap.Config.ARGB_8888);
495
496         try {
497             Utils.matToBitmap(mRgba, bmp);
498             return bmp;
499         } catch(Exception e) {
500             Log.e("org.opencv.samples.tutorial2",
                    "Utils.matToBitmap() throws an exception: "
                     + e.getMessage());
501             bmp.recycle();
502             return null;
503         }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
504         }
505
506         private void drawGrid(int width, int height, int
            grid_cols, int grid_rows) {
507             for (int i = 1; i < grid_rows; i++) {
508                 Core.line(mRgba, new Point(0, i * height /
                    grid_rows), new Point(width, i * height /
                    grid_rows), colorBlue, 3);
509             }
510             for (int i = 1; i < grid_cols; i++) {
511                 Core.line(mRgba, new Point(i * width /
                    grid_cols, 0), new Point(i * width /
                    grid_cols, height), colorBlue, 3);
512             }
513         }
514
515         private void ShowTitle (String s, int iLineNum) {
516             Core.putText(mRgba, s, new Point(10, (int)(
                dTextScaleFactor * 60 * iLineNum)),
517                     Core.FONT_HERSHEY_SIMPLEX,
                        dTextScaleFactor, colorGreen, 2);
518         }
519
520         public void DrawCross (Mat mat, Point pt) {
521             int iCentreCrossWidth = 22;
522
523             Point pt1 = new Point (pt.x - (
                iCentreCrossWidth >> 1), pt.y);
524             Point pt2 = new Point (pt.x + (
                iCentreCrossWidth >> 1), pt.y);
525
526             Core.line(mRgba, pt1, pt2, colorRed,
                iLineThickness - 1);
527
528             pt1.x = pt.x;
529             pt1.y = pt.y + (iCentreCrossWidth >> 1);
530             pt2.x = pt.x;
531             pt2.y = pt.y  - (iCentreCrossWidth >> 1);
532
533             Core.line(mRgba, pt1, pt2, colorRed,
                iLineThickness - 1);
534
535         }
536
537         public void SaveImage (Mat mat) {
538             Log.i(TAG, "SaveImage");
539
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
540             Mat mIntermediateMat = new Mat();
541
542             Imgproc.cvtColor(mat, mIntermediateMat, Imgproc
                .COLOR_RGBA2BGR, 3);
543
544             String filename = Environment.
                getExternalStorageDirectory().getAbsolutePath();
545             filename += "/OpenCV_";
546             SimpleDateFormat fmt = new SimpleDateFormat(
                "yyyy-mm-dd_HH-mm-ss");
547             Date date = new Date(System.currentTimeMillis
                ());
548             String dateString = fmt.format(date);
549             filename += dateString;
550             filename += ".png";
551
552             Highgui.imwrite(filename, mIntermediateMat);
553
554             }
555
556         @Override
557         public void run() {
558             Log.i(TAG, "Starting processing thread");
559             super.run();
560
561             synchronized (this) {
562                 // Explicitly deallocate Mats
563                 if (mRgba != null)
564                     mRgba.release();
565                 if (mGray != null)
566                     mGray.release();
567                 if (mIntermediateMat != null)
568                     mIntermediateMat.release();
569
570                 mRgba = null;
571                 mGray = null;
572                 mIntermediateMat = null;
573             }
574         }
575     }
576
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SampleCvViewBase.java

```
1  package org.opencv.samples.tutorial2;
2
3  import java.util.List;
4
5  import org.opencv.core.Size;
6  import org.opencv.highgui.VideoCapture;
7  import org.opencv.highgui.Highgui;
8
9  import android.content.Context;
10 import android.graphics.Bitmap;
11 import android.graphics.Canvas;
12 import android.util.Log;
13 import android.view.SurfaceHolder;
14 import android.view.SurfaceView;
15
16 public abstract class SampleCvViewBase extends
   SurfaceView implements SurfaceHolder.Callback, Runnable
    {
17      private static final String TAG =
        "Sample::SurfaceView";
18
19      private SurfaceHolder       mHolder;
20      private VideoCapture        mCamera;
21
22      public SampleCvViewBase(Context context) {
23          super(context);
24          mHolder = getHolder();
25          mHolder.addCallback(this);
26          Log.i(TAG, "Instantiated new " + this.getClass
            ());
27      }
28
29      public boolean openCamera() {
30          Log.i(TAG, "openCamera");
31          synchronized (this) {
32              releaseCamera();
33              mCamera = new VideoCapture(Highgui.
                CV_CAP_ANDROID);
34              if (!mCamera.isOpened()) {
35                  mCamera.release();
36                  mCamera = null;
37                  Log.e(TAG, "Failed to open native
                    camera");
38                  return false;
39              }
40          }
41          return true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
42      }
43
44      public void releaseCamera() {
45          Log.i(TAG, "releaseCamera");
46          synchronized (this) {
47              if (mCamera != null) {
48                      mCamera.release();
49                      mCamera = null;
50              }
51          }
52      }
53
54      public void setupCamera(int width, int height) {
55          Log.i(TAG, "setupCamera("+width+", "+height+")"
            );
56          synchronized (this) {
57              if (mCamera != null && mCamera.isOpened()) {
58                  List<Size> sizes = mCamera.
                    getSupportedPreviewSizes();
59                  int mFrameWidth = width;
60                  int mFrameHeight = height;
61
62                  // selecting optimal camera preview size
63                  {
64                      double minDiff = Double.MAX_VALUE;
65                      for (Size size : sizes) {
66                          if (Math.abs(size.height -
                            height) < minDiff) {
67                              mFrameWidth = (int) size.
                                width;
68                              mFrameHeight = (int) size.
                                height;
69                              minDiff = Math.abs(size.
                                height - height);
70                          }
71                      }
72                  }
73
74                  mCamera.set(Highgui.
                    CV_CAP_PROP_FRAME_WIDTH, mFrameWidth);
75                  mCamera.set(Highgui.
                    CV_CAP_PROP_FRAME_HEIGHT, mFrameHeight);
76              }
77          }
78
79      }
80
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SampleCvViewBase.java

```
81      public void surfaceChanged(SurfaceHolder _holder,
        int format, int width, int height) {
82          Log.i(TAG, "surfaceChanged");
83          setupCamera(width, height);
84      }
85
86      public void surfaceCreated(SurfaceHolder holder) {
87          Log.i(TAG, "surfaceCreated");
88          (new Thread(this)).start();
89      }
90
91      public void surfaceDestroyed(SurfaceHolder holder) {
92          Log.i(TAG, "surfaceDestroyed");
93          releaseCamera();
94      }
95
96      protected abstract Bitmap processFrame(VideoCapture
         capture);
97
98      public void run() {
99          Log.i(TAG, "Starting processing thread");
100         while (true) {
101             Bitmap bmp = null;
102
103             synchronized (this) {
104                 if (mCamera == null)
105                     break;
106
107                 if (!mCamera.grab()) {
108                     Log.e(TAG, "mCamera.grab() failed");
109                     break;
110                 }
111
112                 bmp = processFrame(mCamera);
113             }
114
115             if (bmp != null) {
116                 Canvas canvas = mHolder.lockCanvas();
117                 if (canvas != null) {
118                     canvas.drawBitmap(bmp, (canvas.
                        getWidth() - bmp.getWidth()) / 2, (
                        canvas.getHeight() - bmp.getHeight
                        ()) / 2, null);
119                     mHolder.unlockCanvasAndPost(canvas);
120                 }
121                 bmp.recycle();
122             }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
123             }
124
125             Log.i(TAG, "Finishing processing thread");
126         }
127     }
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DeviceListActivity.java

```
1  package org.opencv.samples.tutorial2;
2
3  import java.util.Set;
4
5  import android.app.Activity;
6  import android.bluetooth.BluetoothAdapter;
7  import android.bluetooth.BluetoothDevice;
8  import android.content.BroadcastReceiver;
9  import android.content.Context;
10 import android.content.Intent;
11 import android.content.IntentFilter;
12 import android.os.Bundle;
13 import android.view.View;
14 import android.view.Window;
15 import android.view.View.OnClickListener;
16 import android.widget.AdapterView;
17 import android.widget.ArrayAdapter;
18 import android.widget.Button;
19 import android.widget.ListView;
20 import android.widget.TextView;
21 import android.widget.AdapterView.OnItemClickListener;
22
23 public class DeviceListActivity extends Activity{
24
25     // Return Intent extra
26     public static String EXTRA_DEVICE_ADDRESS =
       "device_address";
27
28     // Member fields
29     private BluetoothAdapter mBtAdapter;
30     private ArrayAdapter<String>
       mPairedDevicesArrayAdapter;
31     private ArrayAdapter<String>
       mNewDevicesArrayAdapter;
32
33     @Override
34     protected void onCreate(Bundle savedInstanceState) {
35         super.onCreate(savedInstanceState);
36
37         // Setup the window
38         requestWindowFeature(Window.
           FEATURE_INDETERMINATE_PROGRESS);
39         setContentView(R.layout.device_list);
40
41         // Set result CANCELED incase the user backs out
42         setResult(Activity.RESULT_CANCELED);
43
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DeviceListActivity.java

```
44          // Initialize the button to perform device
            discovery
45          Button scanButton = (Button) findViewById(R.id.
            button_scan);
46          scanButton.setOnClickListener(new
            OnClickListener() {
47              public void onClick(View v) {
48                  doDiscovery();
49                  v.setVisibility(View.GONE);
50              }
51          });
52
53          // Initialize array adapters. One for already
            paired devices and
54          // one for newly discovered devices
55          mPairedDevicesArrayAdapter = new ArrayAdapter<
            String>(this, R.layout.device_name);
56          mNewDevicesArrayAdapter = new ArrayAdapter<
            String>(this, R.layout.device_name);
57
58          // Find and set up the ListView for paired
            devices
59          ListView pairedListView = (ListView)
            findViewById(R.id.paired_devices);
60          pairedListView.setAdapter(
            mPairedDevicesArrayAdapter);
61          pairedListView.setOnItemClickListener(
            mDeviceClickListener);
62
63          // Find and set up the ListView for newly
            discovered devices
64          ListView newDevicesListView = (ListView)
            findViewById(R.id.new_devices);
65          newDevicesListView.setAdapter(
            mNewDevicesArrayAdapter);
66          newDevicesListView.setOnItemClickListener(
            mDeviceClickListener);
67
68          // Register for broadcasts when a device is
            discovered
69          IntentFilter filter = new IntentFilter(
            BluetoothDevice.ACTION_FOUND);
70          this.registerReceiver(mReceiver, filter);
71
72          // Register for broadcasts when discovery has
            finished
73          filter = new IntentFilter(BluetoothAdapter.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ACTION_DISCOVERY_FINISHED);
74          this.registerReceiver(mReceiver, filter);
75
76          // Get the local Bluetooth adapter
77          mBtAdapter = BluetoothAdapter.getDefaultAdapter
            ();
78
79          // Get a set of currently paired devices
80          Set<BluetoothDevice> pairedDevices = mBtAdapter
            .getBondedDevices();
81
82          // If there are paired devices, add each one
            to the ArrayAdapter
83          if (pairedDevices.size() > 0) {
84              findViewById(R.id.title_paired_devices).
                setVisibility(View.VISIBLE);
85              for (BluetoothDevice device : pairedDevices
                ) {
86                  mPairedDevicesArrayAdapter.add(device.
                    getName() + "\n" + device.getAddress());
87              }
88          } else {
89              String noDevices = getResources().getText(R
                .string.none_paired).toString();
90              mPairedDevicesArrayAdapter.add(noDevices);
91          }
92      }
93
94      @Override
95      protected void onDestroy() {
96          super.onDestroy();
97
98          // Make sure we're not doing discovery anymore
99          if (mBtAdapter != null) {
100             mBtAdapter.cancelDiscovery();
101         }
102
103         // Unregister broadcast listeners
104         this.unregisterReceiver(mReceiver);
105     }
106
107     /**
108      * Start device discover with the BluetoothAdapter
109      */
110     private void doDiscovery() {
111
112         // Indicate scanning in the title
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
113             setProgressBarIndeterminateVisibility(true);
114             setTitle(R.string.scanning);
115
116             // Turn on sub-title for new devices
117             findViewById(R.id.title_new_devices).
                setVisibility(View.VISIBLE);
118
119             // If we're already discovering, stop it
120             if (mBtAdapter.isDiscovering()) {
121                 mBtAdapter.cancelDiscovery();
122             }
123
124             // Request discover from BluetoothAdapter
125             mBtAdapter.startDiscovery();
126         }
127
128         // The on-click listener for all devices in the
            ListViews
129         private OnItemClickListener mDeviceClickListener =
            new OnItemClickListener() {
130             public void onItemClick(AdapterView<?> av, View
                  v, int arg2, long arg3) {
131                 // Cancel discovery because it's costly
                    and we're about to connect
132                 mBtAdapter.cancelDiscovery();
133
134                 // Get the device MAC address, which is
                    the last 17 chars in the View
135                 String info = ((TextView) v).getText().
                    toString();
136                 try {
137                     // Attempt to extract a MAC address
138                     String address = info.substring(info.
                        length() - 17);
139
140                     // Create the result Intent and
                        include the MAC address
141                     Intent intent = new Intent();
142                     intent.putExtra(EXTRA_DEVICE_ADDRESS,
                        address);
143
144                     // Set result and finish this Activity
145                     setResult(Activity.RESULT_OK, intent);
146                     finish();
147                 }catch (IndexOutOfBoundsException e) {
148                     // Extraction failed, set result and
                        finish this Activity
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DeviceListActivity.java

```
149                 setResult(Activity.RESULT_CANCELED);
150                 finish();
151             }
152         }
153     };
154
155     // The BroadcastReceiver that listens for
        discovered devices and
156     // changes the title when discovery is finished
157     private final BroadcastReceiver mReceiver = new
        BroadcastReceiver() {
158         @Override
159         public void onReceive(Context context, Intent
            intent) {
160             String action = intent.getAction();
161
162             // When discovery finds a device
163             if (BluetoothDevice.ACTION_FOUND.equals(
                action)) {
164                 // Get the BluetoothDevice object from
                    the Intent
165                 BluetoothDevice device = intent.
                    getParcelableExtra(BluetoothDevice.
                    EXTRA_DEVICE);
166                 // If it's already paired, skip it,
                    because it's been listed already
167                 if (device.getBondState() !=
                    BluetoothDevice.BOND_BONDED) {
168                     mNewDevicesArrayAdapter.add(device.
                        getName() + "\n" + device.
                        getAddress());
169                 }
170             // When discovery is finished, change the
                Activity title
171             } else if (BluetoothAdapter.
                ACTION_DISCOVERY_FINISHED.equals(action)) {
172                 setProgressBarIndeterminateVisibility(
                    false);
173                 setTitle(R.string.select_device);
174                 if (mNewDevicesArrayAdapter.getCount()
                    == 0) {
175                     String noDevices = getResources().
                        getText(R.string.none_found).
                        toString();
176                     mNewDevicesArrayAdapter.add(
                        noDevices);
177                 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
178                 }
179             }
180         };
181     }
182
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเผยแพร่ข้อมูลทางวิชาการ (วิทยุ)

สถานีวิทยุเนชั่น FM 90.5 MHz

สถานีวิทยุแห่งจุฬาลงกรณ์มหาวิทยาลัย FM 101.5 MHz

การเผยแพร่ข้อมูลทางวิชาการ (โทรทัศน์)