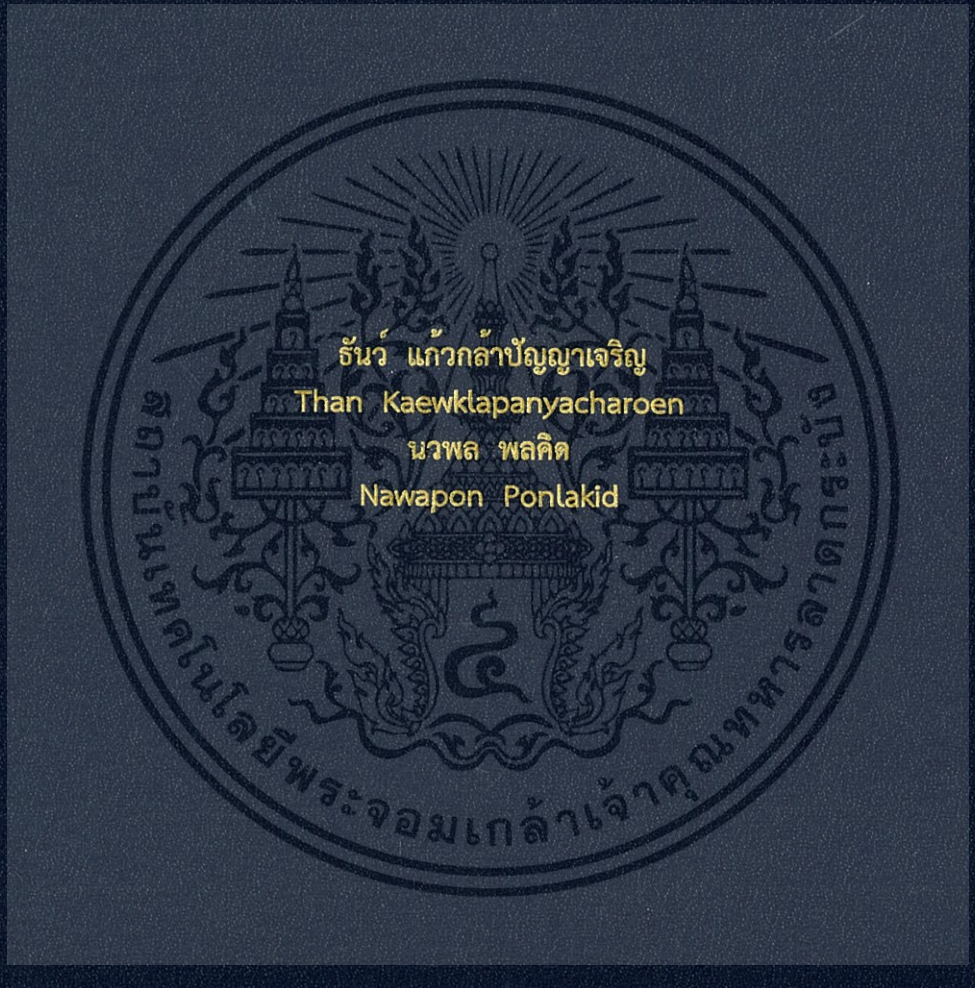
ระบบระบุพิกัดเสียงระเบิด
Explosion Location Detector



ธันว์ แก้วกล้าปัญญาเจริญ
Than Kaewklapanyacharoen
นวพล พลคิด
Nawapon Ponlakid

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2559

ระบบระบุพิกัดเสียงระเบิด

Explosion Location Detector

ธันว์ แก้วกล้าปัญญาเจริญ

นวพล พลคิด

อาจารย์ที่ปรึกษา

อ.ชินภัทร นันทจิวากรชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมบัณฑิต

สาขาวิชาวิสวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2559

ภาควิชา อิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบระบุพิกัดเสียงระเบิด

Explosion Location Detector

ผู้จัดทำ นายธันว์ แก้วกล้าปัญญาเจริญ รหัสประจำตัว 56010604

นายนวพล พลคิด รหัสประจำตัว 56010660

รายงานนี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

.....

(อาจารย์ชินภัทร นันทจิวากรชัย)

อาจารย์ที่ปรึกษาปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบระบุพิกัดเสียงระเบิด
นักศึกษา	นายธันว์ แก้วกล้าปัญญาเจริญ รหัสนักศึกษา 56010604
	นายนวพล พลคิด รหัสนักศึกษา 56010660
ปริญญา	วิศวกรรมศาสตร์บัณฑิต
ภาควิชา	อิเล็กทรอนิกส์
ปีการศึกษา	2559
อาจารย์ที่ปรึกษาปริญญานิพนธ์	อาจารย์ชินภัทร นันทจิวากรชัย

บทคัดย่อ

การระเบิดถือเป็นอุบัติเหตุที่สร้างความเสียหายอย่างมากต่อชีวิตและทรัพย์สิน การเกิดระเบิดแต่ละครั้งจะทำให้เกิดผลกระทบตามมาหลายๆอย่างตามมา เช่น ไฟไหม้ ตึกถล่ม พื้นดินทรุดตัว ซึ่งมีความอันตรายอย่างมากต่อชีวิตของประชาชนบริเวณนั้นๆ รวมถึงการปฏิบัติการของเจ้าหน้าที่ ในการเข้าสู้พื้นที่เพื่อระงับเหตุ โดยปริญญานิพนธ์ฉบับนี้ได้นำเสนอวิธีการ ตรวจจับการระเบิดโดยอาศัยสมบัติทางกายภาพของเสียง ส่งผ่านและประมวลผลระบุพิกัดด้วย อุปกรณ์อิเล็กทรอนิกส์ที่มีระบบปฏิบัติการณ์แอนดรอยด์ ซึ่งเป็นระบบที่แพร่หลายละมีประสิทธิภาพ มากในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis	Explosion Location Detector
Student	Than Kaewklapanyacharoen ID.56010604
	Nawapon Ponlakid ID.56010660
Degree	Bachelor Of Engineering
Program	Electronic Engineering
Year	2016
Thesis Advisor	Advisor Chinnaphat Nantajiwakornchai

ABSTRACT

An explosion is accident that can damage life and assets. Each occasion of explosion has followed effect such as fire, building collapse and ground collapse. This effect is danger for life of population in that areas including operation of staffs approach the areas for suppress the incident. The thesis also presents how to detect explosives based on the physical properties of sound transmission and processing position with android device. This is a very powerful system prevalent today.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การวิจัยระบบระบุพิกัดเสียงระเบิด จะไม่สำเร็จลุล่วงไปได้ด้วยดี ถ้าปราศจากบุคคลเหล่านี้ อ.ชินภัทร นันทจิวากรชัย อาจารย์ที่ปรึกษา ที่คอยให้คำปรึกษาและให้คำแนะนำเป็นอย่างดี ในการแก้ไขปัญญาต่างๆ มาโดยตลอด คุณพ่อ คุณแม่ ที่คอยให้กำลังใจ เป็นที่ปรึกษา และคอยสนับสนุนมาโดยตลอด เพื่อนๆ ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ ที่คอยให้ความช่วยเหลือ และข้อคิดเห็นดีที่นำมาใช้ในการทำงานวิจัยชิ้นนี้ รวมถึงเพื่อนภาควิชาครุศาสตร์วิศวกรรม คณะครุสาสตร์อุตสาหกรรม ที่ช่วยให้คำแนะนำ และคอยให้คำปรึกษา

ขอบพระคุณทุกท่านที่มีส่วนร่วมทำให้งานวิจัยสำเร็จลุล่วงไปได้ด้วยดีไว้ ณ โอกาสนี้ด้วย

ธันว์ แก้วกล้าปัญญาเจริญ

นวพล พลคิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมา 1

1.2 วัตถุประสงค์ของงานวิจัย 1

1.3 ขอบเขตของงานวิจัย 1

1.4 วิธีการดำเนินการ 1

1.5 ประโยชน์ที่ได้รับจากโครงงาน 2

1.6 แนวคิดและหลักการของโครงงาน 2

บทที่2 หลักการและทฤษฎีที่เกี่ยวข้อง 3

2.1 ระบบปฏิบัติการแอนดรอยด์ 3

2.1.1. Android Open SourceProject 3

2.1.2 Open Handset Mobile 3

2.1.3 Cooking หรือ Customize 3

2.2 ภาษาจาวา 4

2.2.1 ข้อดีของภาษาจาวา 4

2.2.2 การเขียนโปรแกรมด้วยภาษาจาวา 6

2.3 โปรแกรม Android Studio 7

2.4 การแปลงฟูเรียร์ 7

2.5 เสียงและคุณสมบัติของสียง 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 อัตราเร็วของเสียง 9

2.6 การใช้อัตราเร็วของเสียงหาระยะทาง 10

บทที่ 3 การออกแบบการทดลอง 11

3.1 อุปกรณ์การทดลอง 11

3.2 วิธีการทดลอง 12

3.2.1 การทดลองเก็บค่าเสียง 12

3.2.2 กรรมวิธีแปลงสัญญาณเสียงให้เป็นสเปคตรัม 13

3.2.3 การทดลองจับเสียงระเบิดและนำมาหาพิกัด 13

3.3 การติดตั้งอุปกรณ์ 13

3.4 อุปกรณ์รับเสียง 15

3.4.1 ส่วนประกอบของอุปกรณ์รับเสียง 15

3.4.3 flow chart 16

บทที่ 4 ผลการทดลองและวิเคราะห์ข้อมูล 17

4.1 ผลการทดลองเก็บเสียงในตำแหน่งการวางไมโครโฟน ที่ระยะห่างต่างกัน 17

4.2 ผลการทดลองจับเสียงและหาพิกัด 18

บทที่ 5 บทสรุป 27

5.1 สรุปผลการทดลอง 27

5.2 ปัญหาและอุปสรรค 27

5.3 ข้อเสนอแนะ 28

เอกสารอ้างอิง 29

ภาพผนวก ก. 30

1. ชุดคำสั่งภาษาจาวาที่ใช้ในอุปกรณ์รับส่งข้อมูล 31

2. โค้ดสมการการหาพิกัดด้วยโปรแกรม Matlab 59

ภาคผนวก ข. 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง 2.1 ตารางแสดงค่าความเร็วของเสียงในตัวกลางต่างๆ ที่อุณหภูมิ 0 ,20 ,25 องศาเซลเซียส 9
ตาราง 4.1 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 10,50 19
ตาราง 4.2 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,50 19
ตาราง 4.3 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 10,50 20
ตาราง 4.4 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,50 20
ตาราง 4.5 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 10,100 21
ตาราง 4.6 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,100 21
ตาราง 4.7 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 10,100 22
ตาราง 4.8 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,100 22
ตาราง 4.9 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 5,80 23
ตาราง 4.10 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 5,80 23
ตาราง 4.11 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 5,80 24
ตาราง 4.12 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 5,80 24
ตาราง 4.13 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 15,80 25
ตาราง 4.14 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 15,80 25
ตาราง 4.15 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 15,80 26
ตาราง 4.16 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 15,80 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่ 1.1 ภาพแสดงผังและขั้นตอนการทำงานของระบบตรวจจับเสียงระเบิด 2
รูปที่ 2.1 สัญญาณเสียงเลข "2" ที่แปลงฟูเรียร์ให้อยู่ในรูปโดเมนความถี่ 7
รูปที่ 2.2 สัญญาณเสียงเลข "2" ที่ผ่านกระบวนการเซปตรัม 8
รูปที่ 3.1 รูปแสดงโทรศัพท์สมาร์ทโฟน True Smart 4G Max 4.0 11
รูปที่ 3.2 ไมโครโฟนแบบไดนามิค / มูฟวิ่งคอล์ย 12
รูปที่ 3.3 รูปแสดงการแปลงสัญญาณด้วย FFT ของแต่ละบิต 13
รูปที่ 3.4 รูปแบบจำลองการติดตั้งอุปกรณ์รับเสียง 14
รูปที่ 3.5 รูปแสดงการติดตั้งอุปกรณ์รับเสียง 15
รูปที่ 3.6 รูปแสดงผังการทำงานของระบบตรวจจับเสียงระเบิด 16
รูปที่ 4.1 กราฟแสดงสเปคตรัมของเสียงประทัดในการทดลอง 17
รูปที่ 4.2 กราฟแสดงค่าเสียงที่เก็บเมื่อไมโครโฟนห่างกันระยะ 2 เมตร 18
รูปที่ 4.3 กราฟแสดงค่าเสียงที่เก็บเมื่อไมโครโฟนห่างกันระยะ 5 เมตร 18
รูปที่ 4.4 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,50 19
รูปที่ 4.5 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,50 20
รูปที่ 4.6 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,100 21
รูปที่ 4.7 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,100 22
รูปที่ 4.8 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 5,80 23
รูปที่ 4.9 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 5,80 24
รูปที่ 4.10 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 15,80 25
รูปที่ 4.11 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 15,80 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

การระเบิดถือเป็นอุบัติเหตุที่สร้างความเสียหายอย่างมากต่อชีวิตและทรัพย์สิน การเกิดระเบิดแต่ละครั้งจะทำให้เกิดผลกระทบตามมาหลายๆอย่างตามมา เช่น ไฟไหม้ ตึกถล่ม พื้นดินทรุดตัว ซึ่งมีความอันตรายอย่างมากต่อชีวิตของประชาชนบริเวณนั้นๆ รวมถึงการปฏิบัติการของเจ้าหน้าที่ ในการเข้าสู้พื้นที่เพื่อระงับเหตุ

เป็นผลทำให้กลุ่มผมคิดว่าจะทำยังไงที่จะช่วยลดการสูญเสียได้หากเกิดเหตุขึ้น ซึ่งกลุ่มผมได้คิดระบบแจ้งเตือนการระเบิดขึ้นเพื่อช่วยอพยพผู้คนออกจากที่เกิดเหตุ ประเมินสถานการณ์ และช่วยให้เจ้าหน้าที่มาถึงจุดเกิดเหตุได้อย่างรวดเร็วละปลอดภัยที่สุด

1.2 วัตถุประสงค์ของงานวิจัย

1.2.1 เพื่อนำเสนอระบบตรวจจับเสียงระเบิดเพื่อช่วยนำผู้คนออกห่างจากที่เกิดเหตุ

1.2.2 นำเสนอระบบแจ้งเตือนผ่านมือถือโดยใช้เครือข่าย 3G

1.2.3 คำนวณหาสถานที่ที่เกิดระเบิด เพื่อประเมินสถานการณ์ก่อนลงพื้นที่

1.3 ขอบเขตของงานวิจัย

1.3.1 ทำการทดลองตรวจจับเสียงที่คล้ายระเบิด (ประทัด) บริเวณสวนผลไม้

1.3.2 ทำให้ระบบสามารถตรวจจับแบบอัตโนมัติได้

1.3.3 ระบบสามารถส่งค่าเวลามาและนำไปคำนวณหาจุดพิกัดการเกิดระเบิดได้

1.4 วิธีการดำเนินการ

1.4.1 ศึกษาการเขียนแอปพลิเคชันใน Android เพื่อให้สามารถตรวจจับเสียงระเบิดได้ รวมถึงส่งข้อมูลไปวิเคราะห์แบบ Realtime

1.4.2 ศึกษาลักษณะของ Power Spectral Density ของเสียงเพื่อแยกออกจากเสียงปกติ

1.4.3 ศึกษาลักษณะการคำนวณพิกัดเพื่อหาตำแหน่งที่เกิดเสียง

1.4.4 นำข้อมูลที่ได้มาออกแบบและสร้างระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่ได้รับจากโครงงาน

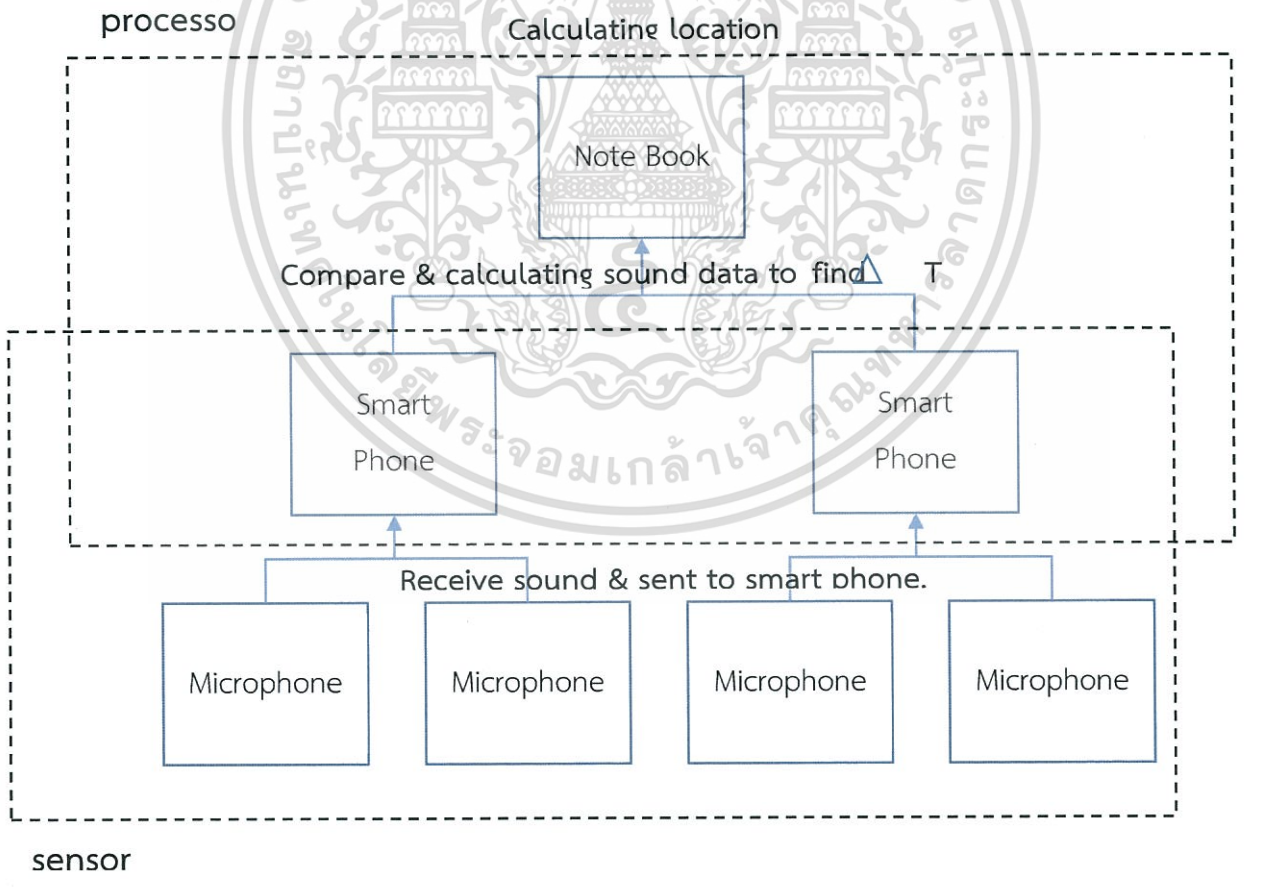
1.5.1 ช่วยอำนวยความสะดวกเวลาเกิดเหตุระเบิด

1.5.2 มีความรู้ในเรื่องของการเขียนแอปพลิเคชันใน Android

1.5.3 ทราบถึงการจำแนกสัญญาณเสียง

1.6 แนวคิดและหลักการของโครงงาน

เสียงที่เดินทางไปถึงเซนเซอร์ตัวรับแต่ละตัวใช้เวลาไม่เท่ากัน จึงสามารถใช้ความต่างของเวลานี้มาหาตำแหน่งย้อนกลับได้ โดยเซนเซอร์ก็คือไมโครโฟน ต่อเข้าโทรศัพท์ และตั้งไว้ห่างกันเพื่อป้องกันลูกคลื่นเสียงซ้อนกัน การต่อไมโครโฟนสองตัวเข้าโทรศัพท์ 1 เครื่องจะทำให้เวลาต่างกันที่ได้ไม่คลาดเคลื่อน การประมวลผลจะใช้สมบัติสเปคตรัมของเสียงในบางความถี่ที่แสดงถึงลักษณะเฉพาะของเสียงระเบิด ถ้าเกินค่าที่ตั้งไว้ก็ส่งเวลาไปคำนวณ



รูปที่ 1.1 ภาพแสดงผังและขั้นตอนการทำงานของระบบตรวจจับเสียงระเบิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 ระบบปฏิบัติการแอนดรอยด์ (Android Operating System : Android OS)

Android คือ ระบบปฏิบัติการสำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือ แท็บเล็ต คอมพิวเตอร์ เน็ตบุ๊ก ทำงานบนลินุกซ์ เคอร์เนล เริ่มพัฒนาโดยบริษัทแอนดรอยด์ (อังกฤษ: Android Inc.) จากนั้นบริษัทแอนดรอยด์ถูกซื้อโดยกูเกิล และนำแอนดรอยด์ไปพัฒนาต่อ ภายหลังถูกพัฒนาในนามของ Open Handset Alliance ทางกูเกิลได้เปิดให้นักพัฒนาสามารถแก้ไขโค้ดต่างๆ ด้วยภาษาจาวา และควบคุมอุปกรณ์ผ่านทางชุด Java libraries ที่กูเกิลพัฒนาขึ้น โดยแอนดรอยด์ (Android) ถูกตั้งชื่อเลียนแบบหุ่นยนต์ในเรื่อง สตาร์วอร์ส ที่ชื่อดรอยด์ ซึ่งเป็นหุ่นยนต์ที่สร้างขึ้นมาเลียนแบบมนุษย์เป็นซอฟท์แวร์ระบบปฏิบัติการที่มีโครงสร้างแบบเรียงทับซ้อนหรือแบบสแต็ก (Stack) โดยใช้ลินุกซ์ เคอร์เนล(Linux Kernel) เป็นพื้นฐานของระบบ และใช้ภาษา Java ในการพัฒนา มี Android SDK เป็นเครื่องมือสำหรับการพัฒนาแอพพลิเคชั่นบนระบบปฏิบัติการแอนดรอยด์ อีกทีหนึ่ง โดยระบบปฏิบัติการ แอนดรอยด์ เริ่มพัฒนาเมื่อปี พ.ศ. 2550 โดยบริษัทแอนดรอยด์ร่วมกับ Google จากนั้นเมื่อปี พ.ศ.2550 ได้มีการร่วมมือกันกว่า 30 บริษัทชั้นนำเพื่อพัฒนาระบบ

ประเภทของชุดซอฟท์แวร์ เนื่องจากแอนดรอยด์ (Android) เปิดให้นักพัฒนาเข้าไปชมรหัสต้นฉบับได้ทำให้มีผู้พัฒนาจากหลายฝ่ายนำเอารหัสต้นฉบับมาปรับแต่ง และสร้างแอนดรอยด์ในแบบฉบับของตนเองขึ้นสามารถแบ่งประเภทของแอนดรอยด์ออกได้เป็น 3 ประเภท ดังต่อไปนี้

2.1.1. Android Open SourceProject (AOSP) เป็นแอนดรอยด์ประเภทแรกที่กูเกิล (Google) เปิดให้สามารถนำต้นฉบับแบบเปิด ไปติดตั้งและใช้งานในอุปกรณ์ต่างๆ ได้โดยไม่ต้องเสียค่าใช้จ่ายใดๆ

2.1.2 Open Handset Mobile (OHM) เป็นแอนดรอยด์ที่ได้รับการพัฒนาร่วมกับกลุ่มบริษัทผู้ผลิตอุปกรณ์พกพา ที่เข้าร่วมกับกูเกิลในนาม Open Handset Alliances (OHA) บริษัทเหล่านี้จะพัฒนาแอนดรอยด์ในแบบฉบับของตนออกมา โดยรูปร่างหน้าตา การแสดงผล และฟังก์ชั่นการใช้งาน จะมีความเป็นเอกลักษณ์ และมีลิขสิทธิ์เป็นของตน พร้อมได้รับสิทธิ์ในการมีบริการเสริมต่างๆ จากกูเกิล ที่เรียกว่า Google Mobile Service (GMS) ซึ่งเป็นบริการเสริมที่ทำให้แอนดรอยด์มีประสิทธิภาพ เป็นไปตามจุดประสงค์ของแอนดรอยด์ แต่การจะได้มาซึ่ง GMS นั้น ผู้ผลิตจะต้องทำการทดสอบระบบ และขออนุญาตกับทางกูเกิลก่อน จึงจะนำเครื่องออกสู่ตลาดได้

2.1.3 Cooking หรือ Customize เป็นแอนดรอยด์ที่นักพัฒนานำเอารหัสต้นฉบับจากแหล่งต่างๆ มาปรับแต่งในแบบฉบับของตนเอง โดยจะต้องทำการปลดล็อคสิทธิ์การใช้งานอุปกรณ์ หรือ Unlock เครื่องก่อนจึงจะสามารถติดตั้งได้ แอนดรอยด์ประเภทนี้ถือเป็นประเภทที่มีความสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากที่สุดเท่าที่อุปกรณ์เครื่องนั้นๆ จะรองรับได้เนื่องจากได้รับการปรับแต่งให้เข้ากับอุปกรณ์นั้นๆ จากผู้ใช้งานจริง

2.2 ภาษาจาวา (Java)

จาวา เป็นภาษาโปรแกรมที่ใช้ในการเขียนคำสั่งในการสั่งงานคอมพิวเตอร์ ซึ่งพัฒนาขึ้นโดยบริษัท ซันไมโครซิสเต็มส์ จำกัด (Sun Microsystems Inc.) ในปี ค.ศ. 1991 เป็นส่วนหนึ่งของโครงการวิจัยเพื่อพัฒนาซอฟต์แวร์ สำหรับอุปกรณ์อิเล็กทรอนิคส์ต่างๆ เช่น โทรทัศน์ โทรศัพท์มือถือ โดยมีเป้าหมายการท างานเชื่อมต่อกับอุปกรณ์ฮาร์ดแวร์ต่างๆได้อย่างกว้างขวาง และมีประสิทธิภาพ ใช้เวลาน้อย รวดเร็วในการพัฒนาโปรแกรม และสามารถเชื่อมต่อไปยังแพล็ตฟอร์ม (Platform) อื่นๆได้ง่าย Java เป็นภาษาสำหรับเขียนโปรแกรมภาษาหนึ่งที่มีลักษณะสนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) ที่ชัดเจน โปรแกรมต่าง ๆ ถูกสร้างภายในคลาส (Class) โปรแกรมเหล่านั้นเรียกว่า Method หรือ Behavior โดยปกติจะเรียกแต่ละ Class ว่า Object โดยแต่ละ Object มีพฤติกรรมมากมาย โปรแกรมที่สมบูรณ์จะเกิดจากหลาย object หรือหลาย Class มารวมกัน โดยแต่ละ Class จะมี Method หรือ Behavior แตกต่างกันไป

2.2.1 ข้อดีของภาษาจาวา

- ภาษา Java เป็นภาษาโปรแกรมที่ง่ายในการเรียนรู้ ภาษา Java มีคุณลักษณะต่างๆ ดังนี้ เช่น เชื่อมต่อข้ามแพล็ตฟอร์ม (Platforms) ต่างๆ ได้ สามารถเขียนโปรแกรมแบบ OOP (Object-Oriented Programming) ได้ง่ายมาก โปรแกรมมีขนาดเล็ก และมีวิธีการ เขียนไม่ยุ่งยากซับซ้อน ดังนั้นโปรแกรมที่เขียนด้วยภาษา Java จึงคอมไพล์ได้ง่าย ตลอดจนตรวจสอบหาข้อผิดพลาดโปรแกรมได้ง่ายด้วย ภาษา java เป็นภาษาที่ทำความ เข้าใจได้ง่ายมาก มีขนาดเล็กและยากที่จะเกิดข้อผิดพลาด เขียนคำสั่งได้ง่าย มี ประสิทธิภาพในการท างานและมีความยืดหยุ่นสูง
- ภาษา Java เป็นการเขียนโปรแกรมเชิงวัตถุ OOP (Object-Oriented Programming) การเขียนโปรแกรมเชิงวัตถุ เป็นเทคนิคการเขียนโปรแกรมให้มีลักษณะ เป็นโมดูล (Module) แบ่งโปรแกรมเป็นส่วนๆ ตามสภาวะแวดล้อมการทำงานของ โปรแกรมซึ่งเรียกว่า Method โดยทุก Method ก็คือ ระเบียบวิธี หรือการทำงานอย่าง ใดอย่างหนึ่ง โดยจะถูกรวบรวมอยู่ในคลาส ซึ่งหลักการเขียนโปรแกรมเชิงวัตถุจะมอง องค์ประกอบของโปรแกรมต่างๆเป็นคลาสหรือวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรียกว่า Object ตัวอย่าง เช่น วัตถุ ที่มองเห็นได้ เช่น รถ สินค้า หรือ วัตถุที่ไม่สามารถมองเห็นได้ เช่น เหตุการณ์ ต่างๆ ข้อมูลต่างๆของ Object จะถูกซ่อนไว้คลาสเรียกว่า Data Encapsulation ซึ่งมี ประโยชน์ในการแก้ไขข้อมูลหรือ Method ใดๆ ที่อยู่ในคลาส โดยไม่ส่งผลกระทบต่อ การทำงานหรือเรียกใช้งานของ Object นั้น นอกจากนั้น Java ยังมีคุณสมบัติการสืบทอด (Inheritance) เพื่อส่งผ่านและถ่ายทอด ลักษณะต่างๆ ของคลาสแม่ไปยังคลาสลูก ทำให้เขียนโปรแกรมได้ง่ายขึ้น และมี โครงสร้างการท างานที่เข้าใจง่ายและมีความสัมพันธ์กัน

- ภาษา Java เป็นอิสระต่อแพล็ตฟอร์ม (Java is Platform-Independent) Java เป็น อิสระต่อแพล็ตฟอร์ม ทั้งระดับซอร์ซโค้ด (Source Code) และไบนารีโค้ด (Binary Code) ช่วยให้สามารถเคลื่อนย้ายโปรแกรมจากระบบคอมพิวเตอร์หนึ่งไปยังระบบ คอมพิวเตอร์อื่นได้อย่างง่ายดาย เพราะว่าโปรแกรมที่เขียนด้วยภาษา Java ได้รวบรวม คำสั่งต่างๆไว้ในไลบรารีคลาสพื้นฐานต่างๆ เป็น Java Packages ช่วยอำนวยความ สะดวกในการเขียนคำสั่ง เมื่อย้ายโปรแกรมไปยังแพล็ตฟอร์มอื่นโดยไม่ต้องเขียนซอร์ซโค้ด (Source Code) ขึ้นใหม่ทำให้ประหยัดเวลามาก เมื่อคอมไพล์ซอร์ซโค้ด จะได้ไฟล์ไบนารีโค้ด ที่เรียกว่า Bytecode การรันโปรแกรมของ Java จะทำงานในลักษณะ อินเทอร์พรีเตอร์ (Interpreter) ของไฟล์ Bytecode ซึ่งสามารถรันบนแพล็ตฟอร์มใดๆ ก็ได้ รวมทั้งระบบปฏิบัติการต่างๆ เช่น ระบบ Windows, Solaris, Linux หรือ MacOS โดยการแปลคำสั่งทีละคำสั่ง แพล็ตฟอร์มที่ Java ทำงานได้จะต้องประกอบด้วย 2 ส่วน คือ Java Virtual Machine (JVM) และ Java Application Programming Interface (Java API) โดย Java Virtual Machine คือเครื่องมือที่รวบรวมคำสั่งคอมไฟล์และรัน โปรแกรม Java ส่วน Java API เป็นกลุ่มของคลาส และอินเตอร์เฟส (Interface) ที่ รวมอยู่ในไลบรารีที่เรียกว่า Java Package เช่น java.awt, java.util หรือ java.io เป็น ต้น ลักษณะการทำงานของ Java ที่เป็นอิสระต่อแพล็ตฟอร์มโดยการเขียนโปรแกรม เพียงครั้งเดียวแต่สามารถนำไปใช้ทำงานยังเครื่องอื่นๆ ได้ นั้นเรียกว่า Write once, Run anywhere นั้นเอง
- ภาษา Java มีระบบการทำงานและมีระบบความปลอดภัยที่ดี Java จะคำสั่งต่างๆที่เป็น ส่วนประกอบของ Java API โดยมีการรวบรวมเป็นคลาสต่างๆไว้มากมาย ช่วยอำนวย ความสะดวกในการเขียนโปรแกรม นอกจากนั้นยังมี Garbage Collector

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยมีระบบ จัดการหน่วยความจำเพื่อเก็บขยะของโปรแกรมและคืนหน่วยความจำให้กับระบบ โปรแกรมที่เขียนด้วยภาษา Java มีระบบจัดการข้อผิดพลาดที่เกิดจากการทำงานของ โปรแกรมที่เรียกว่า Exception Handling ด้วยทำให้สามารถตรวจสอบโปรแกรม (Debug) โปรแกรมได้ง่ายขึ้น Java มีระบบความปลอดภัยที่ดี เช่น โปรแกรม Java ที่ ทำงานบนเว็บบราวเซอร์ (Web Browser) ที่เรียกว่า Java Applet นั้นจะทำงานเฉพาะ บนเครื่องแม่ข่าย (Server) โดยไม่สามารถเข้าถึงเครื่องลูกข่าย (Client) ไปทำลายไฟล์ หรือไฟล์ระบบ (System file) ได้ ทำให้มีระบบความปลอดภัยที่ดี ป้องกันข้อมูลจาก ไวรัส และโปรแกรมที่เขียนด้วย Java ไม่มีพฤติกรรมเป็นไวรัส ได้

2.2.2 การเขียนโปรแกรมด้วยภาษาจาวา การเขียนโปรแกรมด้วยภาษา Java ประกอบด้วยขั้นตอนการท างานทั้งหมด 3 ขั้นตอน ดังนี้

ขั้นตอนที่ 1 สร้างโปรแกรมซอร์สโค้ด โดยการพิมพ์คำสั่งต่างๆ ตามหลักการเขียน โปรแกรมด้วยภาษา Java โดยใช้เอดิเตอร์ (Editor) หรือโปรแกรมที่สามารถพิมพ์ข้อความ (Text Editor) และสามารถบันทึกไฟล์เป็นรหัสแอสกี (ASCII) ได้ เช่น โปรแกรม Notepad หรือ โปรแกรม Editplus เป็นต้น หลังจากเขียนโปรแกรมเสร็จเรียบร้อยแล้ว ต้องทำการ บันทึกข้อมูลเป็นไฟล์ที่มีชื่อเดียวกันกับชื่อคลาสของ Java และใช้นามสกุลไฟล์เป็น java ตัวอย่างเช่น TestJava.java

ขั้นตอนที่ 2 คอมไพล์โปรแกรมซอร์สโค้ด โดยการใช้คำสั่ง javac.exe ที่มากับการ ติดตั้ง JDK แล้ว มีรูปแบบคำสั่ง คือ javac FileName.java เมื่อ FileName.java คือ ชื่อ ไฟล์ใดๆ ที่มีนามสกุล java ถ้าไม่มีข้อผิดพลาดใดๆ ผลลัพธ์ที่ได้จากการคอมไพล์ จะได้ไฟล์ ไบต์โค้ดที่ชื่อเดียวกับชื่อคลาส ตัวอย่างเช่น javac TestJava.java หลังจากการคอมไพล์จะ ได้ไฟล์ TestJava.class ข้อสาคัญในการคอมไพล์ไฟล์ซอร์สโค้ด คือต้องพิมพ์ชื่อไฟล์พร้อม นามสกุลเป็น java เสมอ และต้องพิมพ์ชื่อไฟล์ด้วยตัวอักษรตัวใหญ่หรือตัวเล็กให้ถูกต้องตาม การตั้งชื่อคลาส

ขั้นตอนที่ 3 ทำการรันโปรแกรม เพื่อดูผลลัพธ์ทางจอภาพโดยการรันไฟล์ไบต์โค้ด โดยการใช้คำสั่ง javac.exe ที่มากับการติดตั้ง JDK แล้วซึ่งมีรูปแบบคำสั่งคือ java FileName เมื่อ FileName คือ ชื่อไฟล์ใดๆ ไม่ต้องมีนามสกุล

ดังนั้นการรันโปรแกรมเพียงแค่พิมพ์ชื่อไฟล์ไม่ต้องพิมพ์นามสกุลของไฟล์ และต้อง พิมพ์ชื่อไฟล์ด้วยตัวอักษรตัวใหญ่หรือตัวเล็กให้ถูกต้องตามชื่อ คลาส ตัวอย่างเช่น java TestJava เมื่อ TestJava คือชื่อไฟล์ TeatJava.class

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

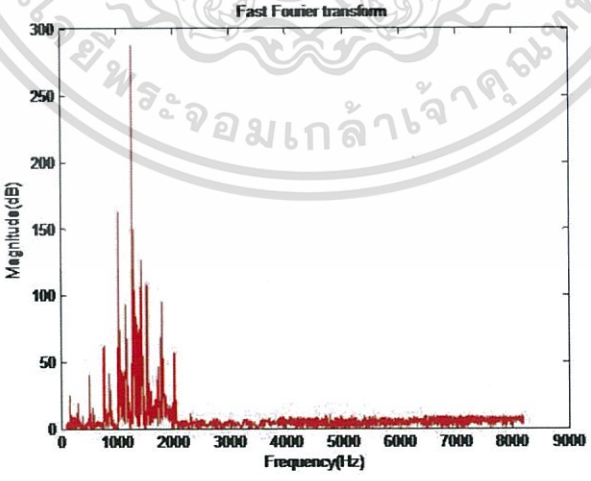
2.3 โปรแกรม Android Studio

เป็นเครื่องมือพัฒนา (IDE : Integrated Development Environment) ที่ถูกสร้างขึ้นมาเพื่อการพัฒนาแอนดรอยด์แอปพลิเคชั่น บนพื้นฐานของแนวคิด IntelliJ โดยมีความสามารถในการทำงานเด่น ๆ ในขณะนี้ ดังต่อไปนี้

- มีความยืดหยุ่นในการใช้งานด้วยใช้ Gradle-based
- การสร้างตัวแปรและการสร้างไฟล์ APK ในหลาย ๆ แม่แบบ
- แม่แบบที่ช่วยอำนวยความสะดวกในการใช้งานคุณสมบัติที่ถูกใช้งานบ่อยๆ
- ตัวช่วยแก้ไขรูปแบบ (Layout) ที่รองรับการลากและวาง
- เครื่องมือที่จะตรวจจับประสิทธิภาพการใช้งาน,การท างานร่วมกันรุ่นและปัญหาอื่นๆ
- การสนับสนุนบิวท์อินสำหรับแพลตฟอร์ม Google Cloud ทำให้มันง่ายต่อการรวม Google Cloud Messaging และ App Engine

2.4 การแปลงฟูเรียร์ (Fast Fourier transform)

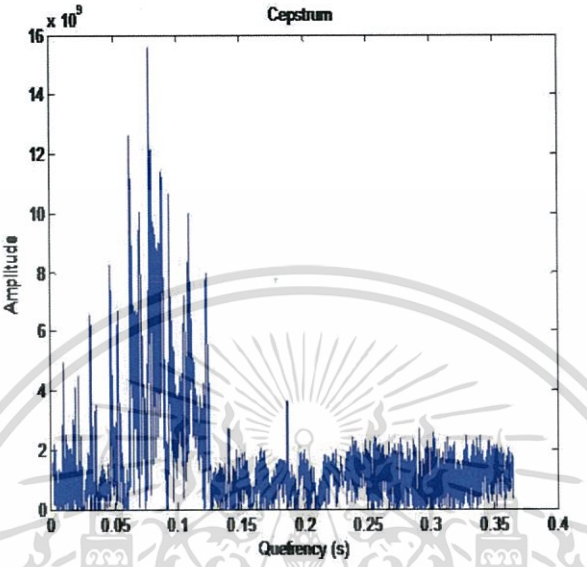
การแปลงฟูเรียร์ (Fast Fourier transform) เป็นการแปลงรูปสัญญาณเสียงที่อยู่ในโดเมนเวลาให้อยู่ในรูปโดเมนความถี่เพื่อจัดการดึงคุณลักษณะเด่นของข้อมูลเสียงออกมา



รูปที่ 2.1 สัญญาณเสียงเลข "2" ที่แปลงฟูเรียร์ให้อยู่ในรูปโดเมนความถี่
(ที่มา http://www.rmcs.buu.ac.th/cecos/project/tsp/images/fft.jpg)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซปสตรัม(Cepstrum) เป็นผลที่ได้มาจากการแปลงฟูเรียร์ (Fourier Transform) ของ ระเบียบวิธี (logarithm) ของสเปกตรัมสัญญาณเสียงในช่วงเวลาสั้น ซึ่งเซปสตรัมนี้จะนำไปใช้ ประโยชน์สำหรับในการหาลักษณะเด่นของเสียง



รูปที่ 2.2 สัญญาณเสียงเลข "2" ที่ผ่านกระบวนการเซปตรัม
(ที่มา http://www.rmcs.buu.ac.th/cecos/project/tsp/images/Ceps1.jpg)

การปรับค่าเซปตรัมให้มีความเรียบ (Smooth Cepstrum)เพื่อให้ง่ายต่อการวิเคราะห์ข้อมูลเสียงต้องทำ การกำจัดข้อมูลเสียงที่ยังมีความถี่ของเสียงรบกวนผสมอยู่ตรงช่วงฐานข้อมูลด้านล่าง และเมื่อกำจัด ความถี่ของเสียงรบกวนที่ผสมอยู่ในข้อมูลเสียงแล้วนั้นให้ทำกราฟข้อมูลเสียงให้เป็นมาตรฐานเดียวกัน อีกครั้ง

2.5 เสียงและคุณสมบัติของสียง

เสียงเกิดจากการสั่นของวัตถุ วัตถุที่มีการสั่นแล้วทำให้เกิดเสียงเรียกว่า แหล่งกำเนิดเสียง สำหรับมนุษย์เสียงพูดเกิดจากการสั่นสะเทือนของสายเสียงซึ่งอยู่ภายในกล่องเสียงบริเวณด้านหน้าของลำคอเรียกว่าลูกกระเดือก มนุษย์สามารถควบคุมเสียงที่พูดพูดขึ้นโดยใช้ฟัน ลิ้น ริมฝีปาก ทำให้เกิดเสียงที่แตกต่างกัน แต่เสียงจะมีประโยชน์อย่างสมบูรณ์ต้องมีการได้ยิน เมื่อเสียงเกิดจากสั่นสะเทือนของวัตถุ แสดงว่าวัตถุได้รับพลังงาน พลังงานนี้ก็จะถูกถ่ายโอนผ่านอากาศมายังหูผู้ฟัง ถ้าไม่มีอากาศเป็นตัวกลางในการถ่ายโอนพลังงาน เราจะไม่ได้ยินเสียงเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 อัตราเร็วของเสียง

ช่วงเวลาที่เสียงเคลื่อนที่จากแหล่งกำเนิดเสียงผ่านอากาศมาถึงหูผู้ฟัง ขึ้นกับระยะทางระหว่างต้นกำเนิดเสียงกับผู้รับฟัง ถ้าระยะห่างมาก เสียงต้องใช้ช่วงเวลานานกว่าจะได้ยินเสียง แต่ถ้าระยะใกล้ เสียงใช้ช่วงเวลาสั้นกว่า

การเคลื่อนที่ของเสียงผ่านตัวกลางหนึ่งไปยังอีกตัวกลางหนึ่งความถี่จะมีค่าคงที่ โดยความเร็วของคลื่นเสียงจะขึ้นอยู่กับชนิดของตัวกลางและอุณหภูมิ เมื่อนักฟิสิกส์ศึกษาอัตราเร็วของเสียงในอากาศ เขาได้พบว่าอัตราเร็วของเสียงในอากาศมีความสัมพันธ์กับอุณหภูมิของอากาศโดยประมาณ ตามสมการ

$$v_t = 331 + 0.6t$$

เมื่อ v_t **เป็นอัตราเร็วของเสียงในอากาศที่อุณหภูมิ** t **ใด ๆ มีหน่วยเป็น เมตร/วินาที**

t **เป็นอุณหภูมิของอากาศมีหน่วยเป็นองศาเซลเซียส**

การเคลื่อนที่ของเสียงในตัวกลางหนึ่งๆ จะคงตัว เมื่ออุณหภูมิของตัวกลางคงตัว ดังในตาราง

ตัวกลาง	อัตราเร็ว (เมตร / วินาที)
แก๊ส	
อากาศ (0° C)	331
อากาศ (20° C)	343
ไฮโดรเจน (0° C)	1286
ออกซิเจน (0° C)	317
ฮีเลียม (0° C)	972
ของเหลว (25° C)	
น้ำ	1493
เมทิลแอลกอฮอล์	1143
น้ำทะเล	1533
ของแข็ง	
อะลูมิเนียม	5100
ทองแดง	3560
เหล็ก	5130
ตะกั่ว	1322

ตารางที่ 2.1 ตารางแสดงค่าความเร็วของเสียงในตัวกลางต่างๆ ที่อุณหภูมิ 0 ,20 ,25 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การใช้อัตราเร็วของเสียงหาระยะทาง

การใช้คลื่นเสียงวัดระยะทาง ส่วนมากจะใช้ในน้ำ เนื่องจากอัตราเร็วของเสียงในน้ำมีค่าสูงกว่ายานพาหนะหรือวัตถุอื่นที่เคลื่อนที่ในน้ำมาก เช่นการวัดความลึกของทะเล หรือการใช้คลื่นโซนาร์เป็นเรดาร์ของชาวประมงในการสำรวจหาฝูงปลาเป็นต้น และคลื่นเสียงก็สามารถหาระยะทางในอากาศได้เช่นกัน โดยอาศัยสมบัติความเร็วของเสียงในอากาศเข้ามาวิเคราะห์ ซึ่งความเร็วของเสียงในอากาศมีคามแปรผันเนื่องจากอุณหภูมิ กล่าวคืออุณหภูมิเปลี่ยนแปลงสามารถมีผลกระทบต่ออัตราเร็วของเสียงได้ถ้าอุณหภูมิของอากาศเพิ่มขึ้น ณ ความดันคงที่ อากาศย่อม ขยายตัวออกตามกฎของชาร์ลและจะมีความหนาแน่นลดลงทำให้อัตราเร็วของเสียงเพิ่มขึ้นตามอุณหภูมิอัตราเร็วของเสียงในอากาศจะแปรผันโดยตรงกับอุณหภูมิ(อุณหภูมิเคลวิน)

อัตราเร็วของเสียงในอากาศโดยประมาณหาได้จาก

$$c_{\text{air}} \approx (331 + (0.606 \cdot \theta)) \quad \text{m/s}$$

โดยที่ θ คือ อุณหภูมิ ในหน่วย องศาเซลเซียส ความแม่นยำในการประมาณในช่วงของอุณหภูมิในช่วง -20°C ถึง 40°C จะมีค่าความผิดพลาดไม่เกิน 0.2% ในช่วงอุณหภูมิสูงกว่าหรือ ต่ำกว่านั้นอัตราเร็วของเสียงจะประมาณโดย

$$c_{\text{air}} \approx 331\sqrt{1 + \frac{\theta}{273}} \quad \text{m/s}$$

จากหลักการข้างต้นทำให้สามารถหาระยะทางได้จากสมการ

ระยะทาง = ความเร็วของเสียง x เวลาที่เสียงใช้เดินทาง

โดยเวลาที่เสียงใช้เดินทางนั้นสามารถหาได้ด้วยอุปกรณ์จับเสียง หรือเซนเซอร์เสียง ดังนั้นลักษณะการวางอุปกรณ์รับเสียงจึงมีผลอย่างมากในการคำนวณหาระยะทางในการเดินทางของเสียง

การตั้งสมการอิงจาก ทฤษฎี พีทาโกรัส ซึ่งอธิบายถึงการหาระยะทางโดยอ้างอิงถึงสามเหลี่ยมมุมฉาก ในทางการคำนวณนั้นจะต้องให้ตัวรับทั้งหมด 4 ตัววางในพิกัดที่ทราบค่า โดยสามารถเขียนสมการได้ดังนี้

$$\sqrt{(X-X1)^2+(Y-Y1)^2} - \sqrt{(X-X2)^2+(Y+Y2)^2} = 345 * \Delta t_1$$
$$\sqrt{(X-X3)^2+(Y-Y3)^2} - \sqrt{(X-X4)^2+(Y+Y4)^2} = 345 * \Delta t_2$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบการทดลอง

3.1 อุปกรณ์การทดลอง

(1) โทรศัพท์สมาร์ทโฟน ระบบปฏิบัติการแอนดรอยด์ True Smart 4G MAX 4.0 ใช้ในการรับข้อมูลเสียงจากไมโครโฟน มาวิเคราะห์ หาค่าเวลาและส่งไปประมวลผล ซึ่งมีสเปคดังนี้

ระบบปฏิบัติการ: Android 6.0 (Marshmallow)

หน่วยประมวลผล : Quad Core

- ความเร็ว : 1.3 GHz

หน่วยความจำ

- ROM 8GB
- RAM 512MB

การเชื่อมต่อ

- Bluetooth
- Wifi

รูปที่ 3.1 รูปแสดงโทรศัพท์สมาร์ทโฟน True Smart 4G Max 4.0 (ที่มา http://www.checkraka.com/uploaded/gallery/42/428ba772277adf896968d7c51f555f7a.jpg)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) ไมโครโฟนแบบไดนามิค / มูฟวิ่งคอล์ย ทำหน้าที่รับเสียงระเบิด

รูปที่ 3.2 ไมโครโฟนแบบไดนามิค / มูฟวิ่งคอล์ย
(ที่มา http://www.sherman.co.th/img/products/microphone/mic-112-1.jpg)

(3) วงจรรวมสัญญาณ ทำหน้าที่รวม และขยายสัญญาณที่ได้จากไมโครโฟน และส่งไปยังโทรศัพท์

(4) คอมพิวเตอร์ ใช้ในการประมวลผล หาตำแหน่งพิกัดและส่งไปยังผู้รับ

3.2 วิธีการทดลอง

3.2.1 การทดลองเก็บค่าเสียง

ทำการทดลองจับเสียงระเบิด โดยการจุดประทัดบริเวณต่างๆของสถานที่ทดสอบ โดยตั้งอุปกรณ์จับเสียงในรูปแบบต่างๆ เพื่อหารูปแบบที่ดีที่สุด และคลาดเคลื่อนน้อยที่สุด โดยการทดลองแบ่งการจัดเรียงของอุปกรณ์รับเสียงดังนี้

(1) จัดเรียงเป็นแนวขนาน โดยไมโครโฟนห่างกัน ตัวละ 2 เมตร

(2) จัดเรียงเป็นแนวขนาน โดยไมโครโฟนห่างกัน ตัวละ 5 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 กรรมวิธีแปลงสัญญาณเสียงให้เป็นสเปคตรัม

ทำการแปลงสัญญาณเสียงที่รับมาด้วยวิธี Fast Fourier Transform

k	$N = 2^k$	จำนวนครั้งของการบวก			จำนวนครั้งของการคูณ		
		ตามนิยาม	FFT	นิยาม/FFT	ตามนิยาม	FFT	นิยาม/FFT
4	16	225	64	3.5	256	32	8
6	64	3969	384	10.3	4096	192	21.3
8	256	65025	2048	31.8	65536	1024	64
10	1024	1.04×10^6	1.02×10^4	102.2	1.05×10^6	5.12×10^3	204.8
12	4096	1.68×10^7	4.92×10^4	341.2	1.68×10^7	2.46×10^4	682.7
14	16384	2.68×10^8	2.29×10^5	1.17×10^3	2.68×10^8	1.15×10^5	2.34×10^3
16	65536	4.29×10^9	1.05×10^6	4.09×10^3	4.29×10^9	5.24×10^5	8.19×10^3
18	262144	6.87×10^{10}	4.72×10^6	1.45×10^4	6.87×10^{10}	2.36×10^6	2.91×10^4
20	1048576	1.10×10^{12}	2.10×10^7	5.24×10^4	1.10×10^{12}	1.05×10^7	1.05×10^5

รูปที่ 3.3 รูปแสดงการแปลงสัญญาณด้วย FFT ของแต่ละบิต

3.2.3 การทดลองจับเสียงระเบิดและนำมาหาพิกัด

ทำการทดลองโดยการจุดประทัดในตำแหน่งต่างๆของสถานที่ทดสอบ และนำค่าเวลาที่ได้มาหาตำแหน่ง โดยใช้อุปกรณ์รับเสียง 2 ประเภท ได้แก่ โทรศัพท์สมาร์ทโฟน และคอมพิวเตอร์ เพื่อนำค่าที่ได้มาเปรียบเทียบ และหาค่าความคลาดเคลื่อนที่เกิดขึ้น

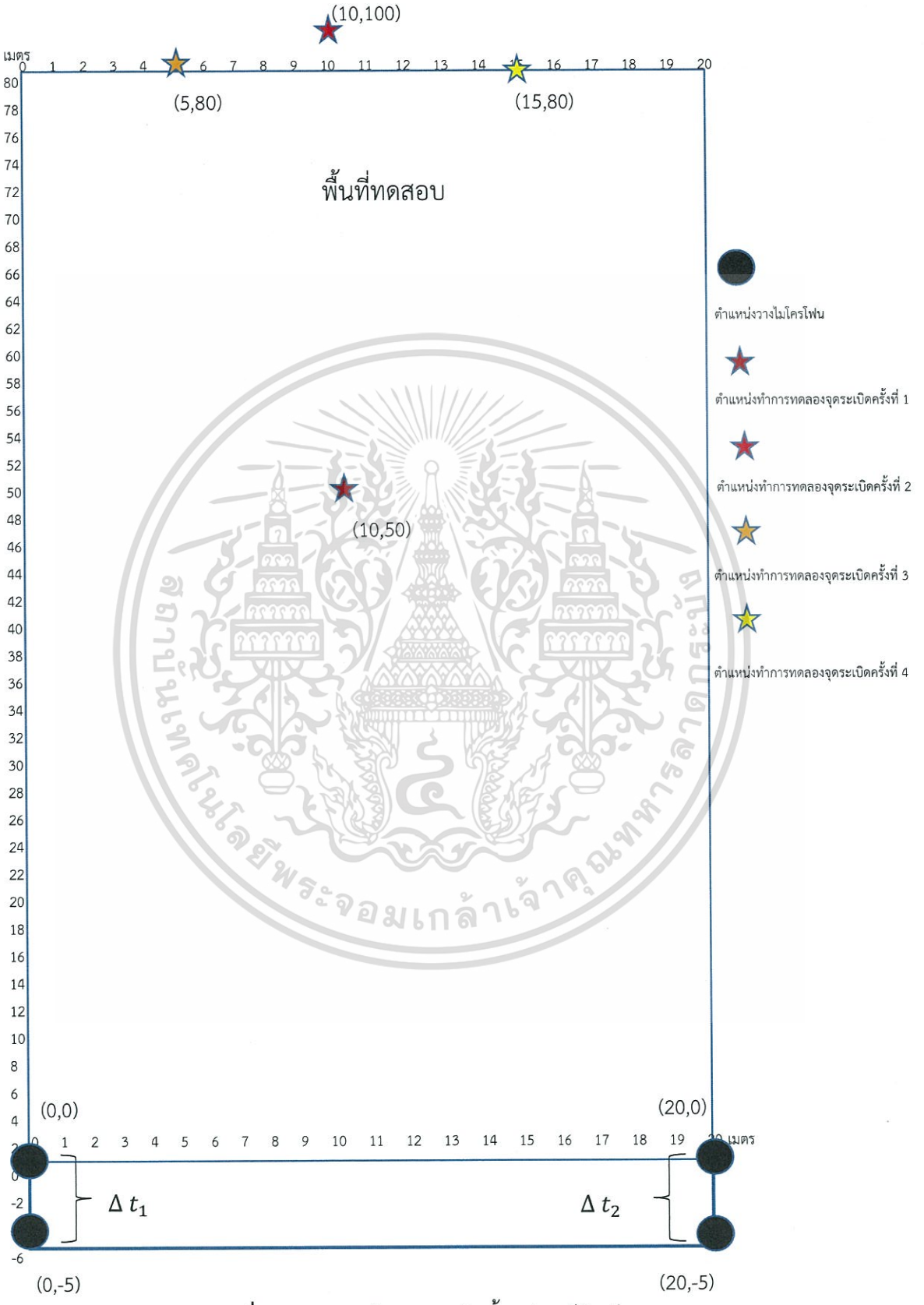
โดยใช้โปรแกรม Matlab ในการคำนวณหาพิกัด และมีสมการ ดังนี้

```
syms x y
t1=;
t2=;
eqn1 =( sqrt((x-0)^2 + (y-5)^2) -sqrt((x-0)^2 + (y-0)^2) )==345*(-t1);
eqn2 = (sqrt((x-20)^2 + (y-5)^2)- sqrt((x-20)^2 + (y-0)^2))==345*(-t2);
sol = solve([eqn1, eqn2], [x, y]);
xSol = sol.x
ySol = sol.y
```

3.3 การติดตั้งอุปกรณ์

การติดตั้งอุปกรณ์ จะต้องติดตั้งให้ห่างจากสถานที่ที่มีคนพลุกพล่าน เพื่อลดสัญญาณรบกวน ทำให้อุปกรณ์มีประสิทธิภาพมากขึ้น โดยตั้งอุปกรณ์แต่ละคู่ห่างกัน 20 ม. และแต่ละตัวห่างกัน 5 ม.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 รูปแบบจำลองการติดตั้งอุปกรณ์รับเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.5 รูปแสดงการติดตั้งอุปกรณ์รับเสียง

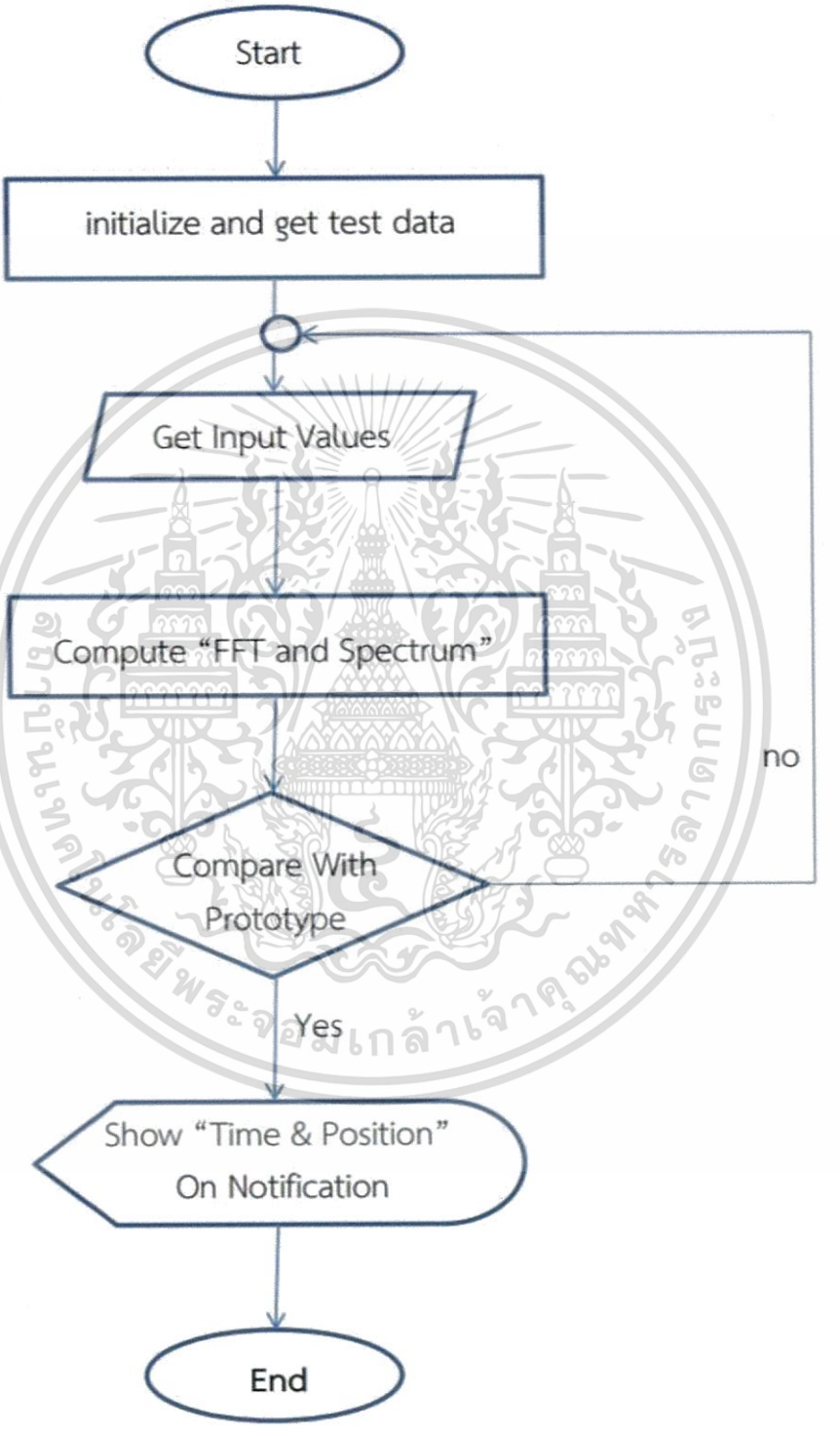
3.4 อุปกรณ์รับเสียง

3.4.1 ส่วนประกอบของอุปกรณ์รับเสียง

ใช้ ไมโครโฟนมูฟวิ่งคอล์ย 2ตัว ต่อเข้ากับโมดูลรับเสียงของโทรศัพสมาร์ทโฟน ระบบปฏิบัติการแอนดรอยด์ True Smart 4G MAX 4.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 flow chart



รูปที่ 3.6 รูปแสดงผังการทำงานของระบบตรวจจับเสียงระเบิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลองและวิเคราะห์ข้อมูล

จากการทดลองสามารถแบ่งได้ 2 ส่วนหลักๆ คือ ส่วนของการหาสเปคตรัมของเสียงและการเก็บค่า และส่วนของการจับเสียงและส่งค่าไปคำนวณ เพื่อหาค่าพิกัดของเสียงระเบิดที่เกิดขึ้น

4.1 ผลการทดลองเก็บเสียงในตำแหน่งการวางไมโครโฟน ที่ระยะห่างต่างกัน

ในการทดลองส่วนนี้เป็นการเก็บค่าเสียง เพื่อนำมาเก็บเป็นค่ามาตรฐาน และวิเคราะห์หาระยะวางไมโครโฟนที่เหมาะสม โดยค่าที่ได้แสดงในรูปแบบของกราฟดังนี้

รูปที่ 4.1 กราฟแสดงสเปคตรัมของเสียงประทัดในการทดลอง

จากรูป 4.1 ทำให้ทราบถึงลักษณะเฉพาะของเสียงที่นำมาทดสอบ โดยใช้การแปลง FFT ในการจำแนกและเก็บลักษณะของเสียงต่างๆในการทดลอง รวมถึงการนำมากำหนดเป็นค่ามาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 กราฟแสดงค่าเสียงที่เก็บเมือไมโครโฟนห่างกันระยะ 2 เมตร

รูปที่ 4.3 กราฟแสดงค่าเสียงที่เก็บเมือไมโครโฟนห่างกันระยะ 5 เมตร

จากรูปที่ 4.2 และ 4.3 แสดงให้เห็นว่าระยะห่างระหว่างไมโครโฟนมีผลกับการรับและ การจับเสียง โดยการจับเสียงของระบบ ต้องมีระยะห่างของข้อมูลในระยะที่พอเหมาะ จึงจะสามารถทำงานได้อย่างมีประสิทธิภาพ ซึ่งจากการทดลองพบว่าระยะที่เหมาะสมคือ ระยะตั้งแต่ 5 เมตรขึ้นไป

4.2 ผลการทดลองจับเสียงและหาพิกัด

ในการทดลองตรงส่วนนี้เป็นการทดลองเพื่อหาค่าผลต่างเวลา ไมโครโฟน 2 ตัว 2 ชุด โดยแบ่งตัวรับเป็น 2 ชนิด คือ โทรศัพท์สมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์ และคอมพิวเตอร์ เพื่อนำค่าที่ได้มาหาจุดพิกัดการเกิดระเบิด โดยพิกัดการจุดระเบิดมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งจุดระเบิด ตามแกน X 10 เมตร, ตามแกน Y 50 เมตร

จับโดยใช้ คอมพิวเตอร์

ความต่างของเวลาจากการอ่านกราฟเสียง (มิลลิวินาที)

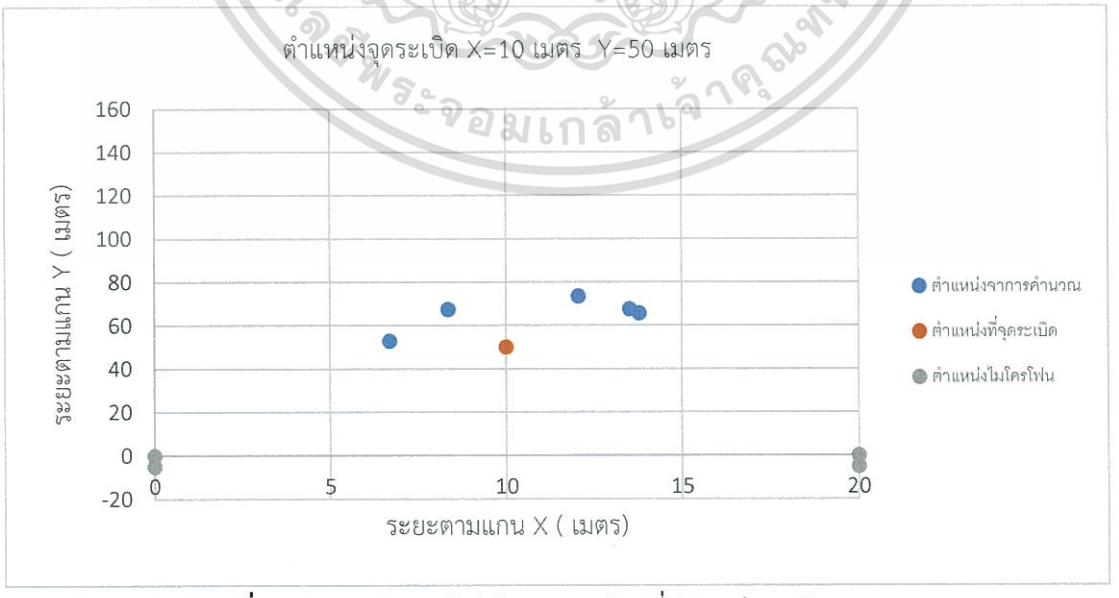
	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
จากไมโครโฟน คู่ที่ 1	13.99	13.81	14.34	14.13	14.18
จากไมโครโฟน คู่ที่ 2	14.10	14.02	13.76	14.04	14.25

ตาราง 4.1 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 10,50

ผลการหาตำแหน่ง (เมตร)

	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
ตำแหน่ง X	10.64	10	6.1354	9.4223	10.651
ตำแหน่ง Y	41.93	51.5561	44.4962	43.89	52.9945

ตาราง 4.2 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,50



รูปที่ 4.4 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งจุดระเบิด ตามแกน X 10 เมตร, ตามแกน Y 50 เมตร
จับโดยใช้ โทรศัพท์สมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์

ความต่างของเวลาจากการอ่านกราฟเสียง (มิลลิวินาที)

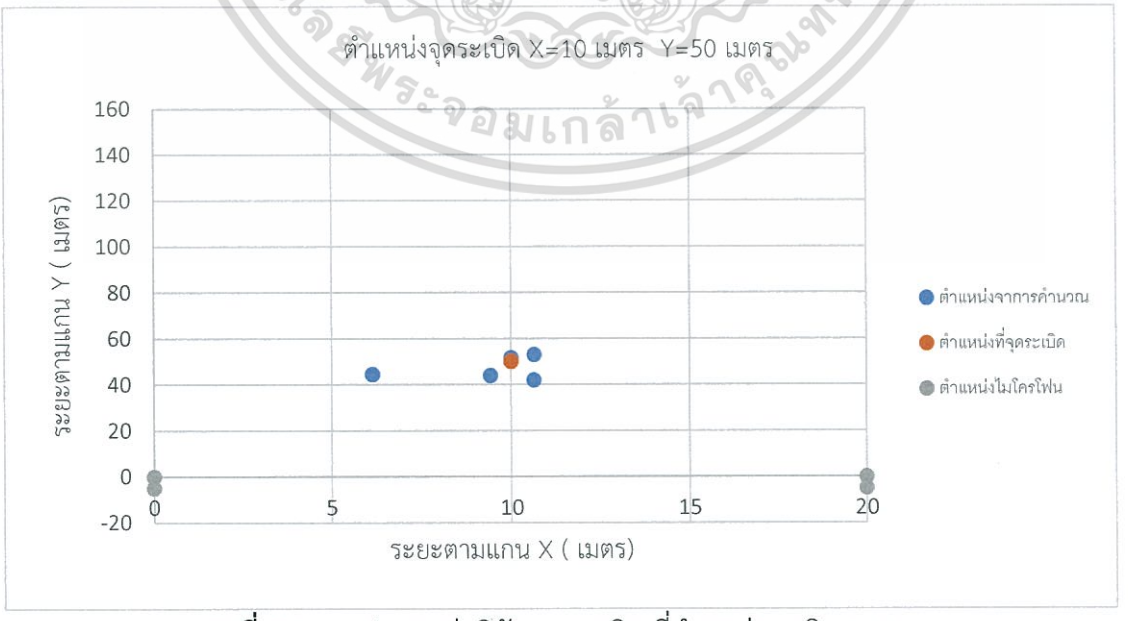
	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
จากไมโครโฟน คู่ที่ 1	14.159	14.374	14.189	14.366	14.288
จากไมโครโฟน คู่ที่ 2	14.423	14.265	14.421	14.011	14.403

ตาราง 4.3 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 10,50

ผลการหาตำแหน่ง (เมตร)

	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
ตำแหน่ง X	13.78507	8.358301	13.51235	6.696306	12.06202
ตำแหน่ง Y	65.66522	67.43668	67.50206	52.85918	73.53723

ตาราง 4.4 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,50



รูปที่ 4.5 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งจุดระเบิด ตามแกน X 10 เมตร, ตามแกน Y 100 เมตร
จับโดยใช้ คอมพิวเตอร์

ความต่างของเวลาจากการอ่านกราฟเสียง (มิลลิวินาที)

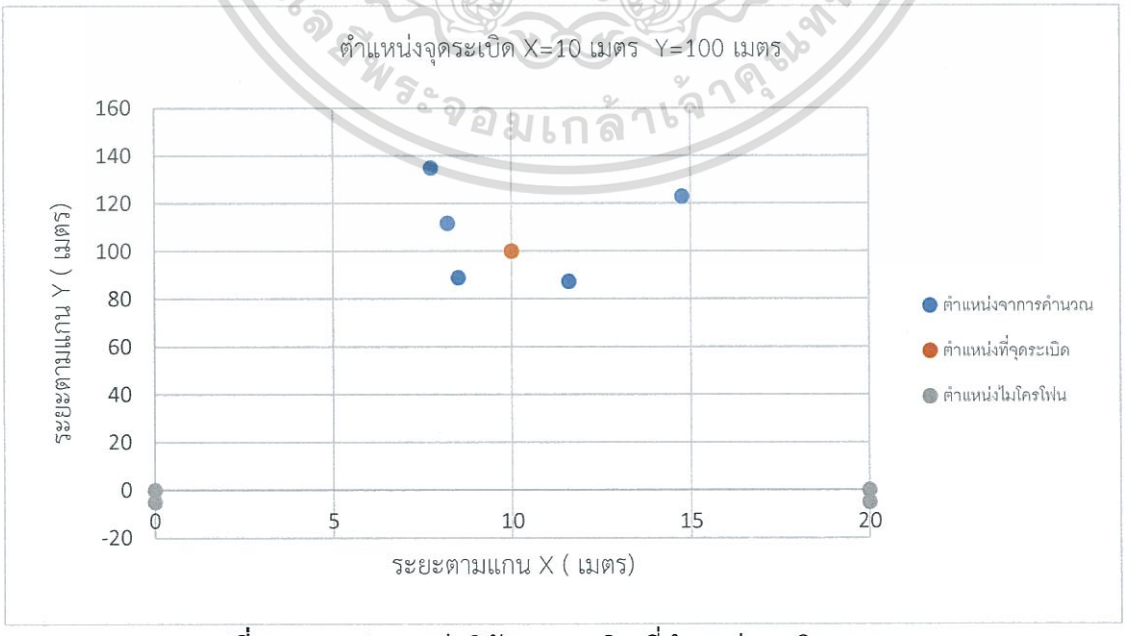
	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
จากไมโครโฟน คู่ที่ 1	14.43	14.36	14.39	14.47	14.38
จากไมโครโฟน คู่ที่ 2	14.38	14.42	14.46	14.43	1.441

ตาราง 4.5 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 10,100

ผลการหาตำแหน่ง (เมตร)

	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
ตำแหน่ง X	8.5326	11.5076	12.7997	12.7997	10.7795
ตำแหน่ง Y	93.9265	86.9715	109.445	109.445	88.4474

ตาราง 4.6 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,100



รูปที่ 4.6 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งจุดระเบิด ตามแกน X 10 เมตร, ตามแกน Y 100 เมตร
จับโดยใช้ โทรศัพท์สมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์

ความต่างของเวลาจากการอ่านกราฟเสียง (มิลลิวินาที)

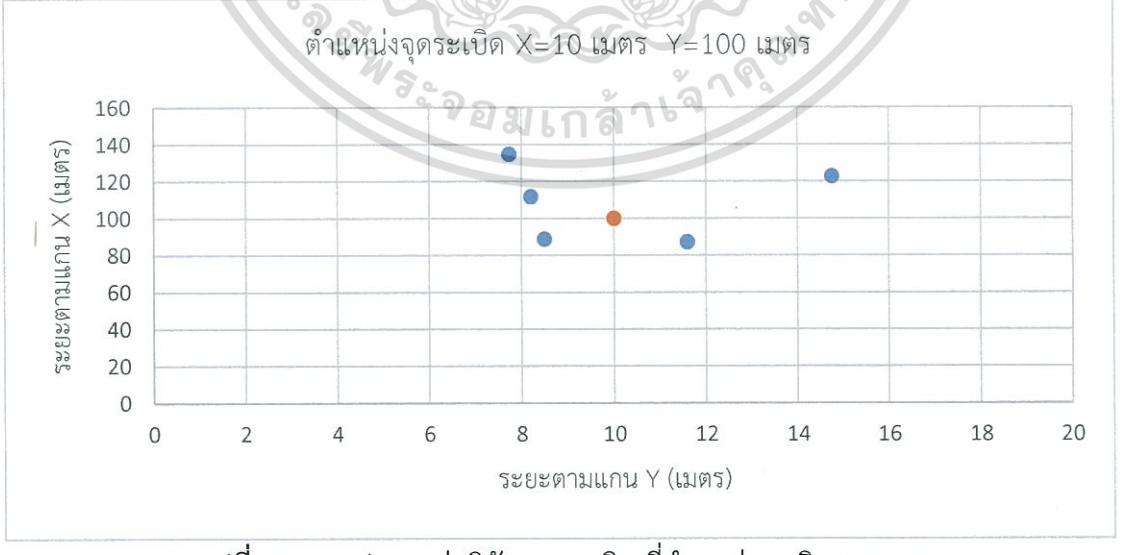
	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
จากไมโครโฟน คู่ที่ 1	14.385	14.452	14.243	14.468	14.359
จากไมโครโฟน คู่ที่ 2	14.479	14.409	14.366	14.431	14.422

ตาราง 4.7 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 10,100

ผลการหาตำแหน่ง (เมตร)

	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
ตำแหน่ง X	14.75445	8.207548	8.503257	7.744401	11.59472
ตำแหน่ง Y	122.8448	111.7412	88.89139	134.8577	87.2873

ตาราง 4.8 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,100



รูปที่ 4.7 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 10,100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งจุดระเบิด ตามแกน X 5 เมตร, ตามแกน Y 80 เมตร

จับโดยใช้ คอมพิวเตอร์

ความต่างของเวลาจากการอ่านกราฟเสียง (มิลลิวินาที)

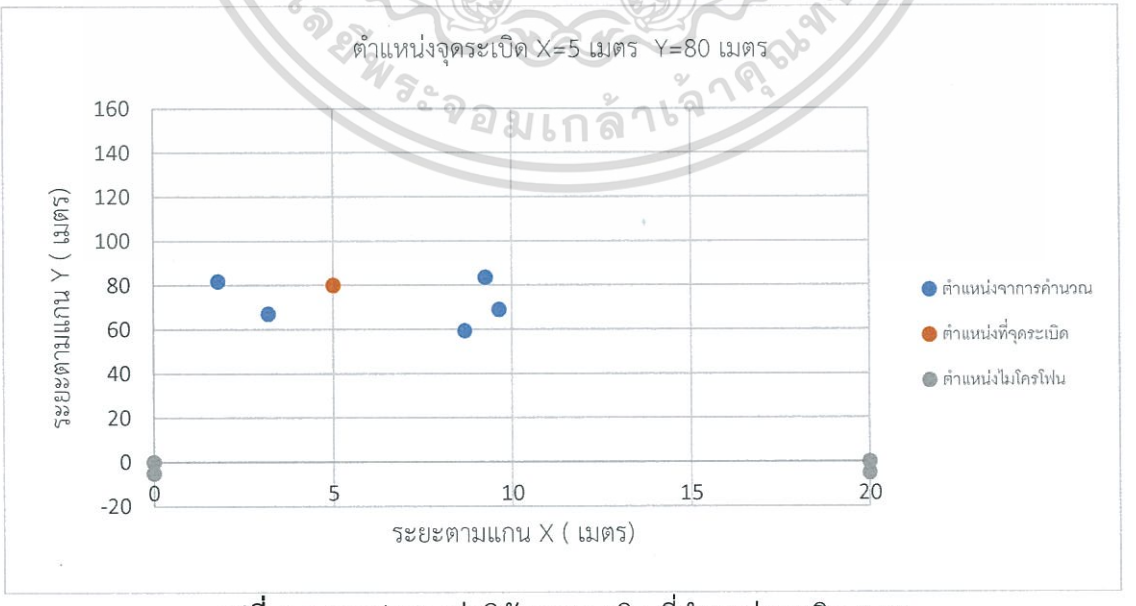
	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
จากไมโครโฟน คู่ที่ 1	14.49	14.48	14.45	14.46	14.43
จากไมโครโฟน คู่ที่ 2	14.24	14.16	14.19	14.26	14.23

ตาราง 4.9 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 5,80

ผลการหาตำแหน่ง (เมตร)

	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
ตำแหน่ง X	1.8679	3.228643	5.408765	5.414456	6.519242
ตำแหน่ง Y	98.34335	79.4495	72.80444	82.93728	72.37149

ตาราง 4.10 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 5,80



รูปที่ 4.8 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 5,80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งจุดระเบิด ตามแกน X 5 เมตร, ตามแกน Y 80 เมตร
จับโดยใช้ โทรศัพท์สมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์

ความต่างของเวลาจากการอ่านกราฟเสียง (มิลลิวินาที)

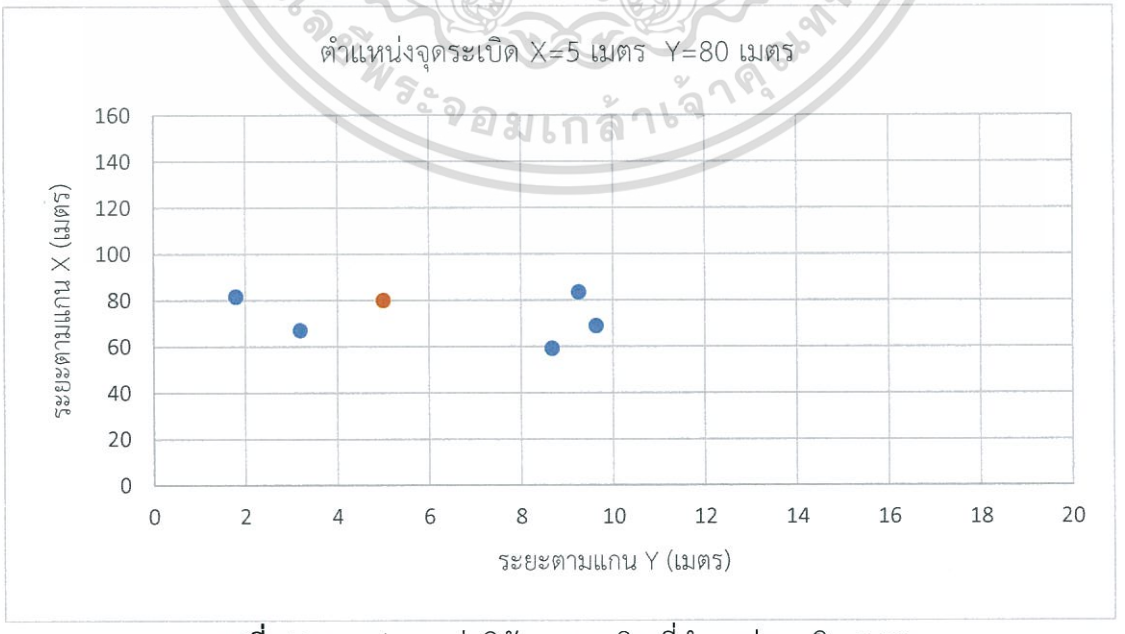
	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
จากไมโครโฟน คู่ที่ 1	14.421	14.489	14.399	14.326	14.475
จากไมโครโฟน คู่ที่ 2	14.319	14.124	14.367	14.212	14.025

ตาราง 4.11 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 5,80

ผลการหาตำแหน่ง (เมตร)

	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
ตำแหน่ง X	9.637454	1.801195	9.258862	8.675318	3.196649
ตำแหน่ง Y	68.86337	81.66405	83.54253	59.24757	67.07101

ตาราง 4.12 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 5,80



รูปที่ 4.9 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 5,80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งจุดระเบิด ตามแกน X 15 เมตร, ตามแกน Y 80 เมตร

จับโดยใช้ คอมพิวเตอร์

ความต่างของเวลาจากการอ่านกราฟเสียง (มิลลิวินาที)

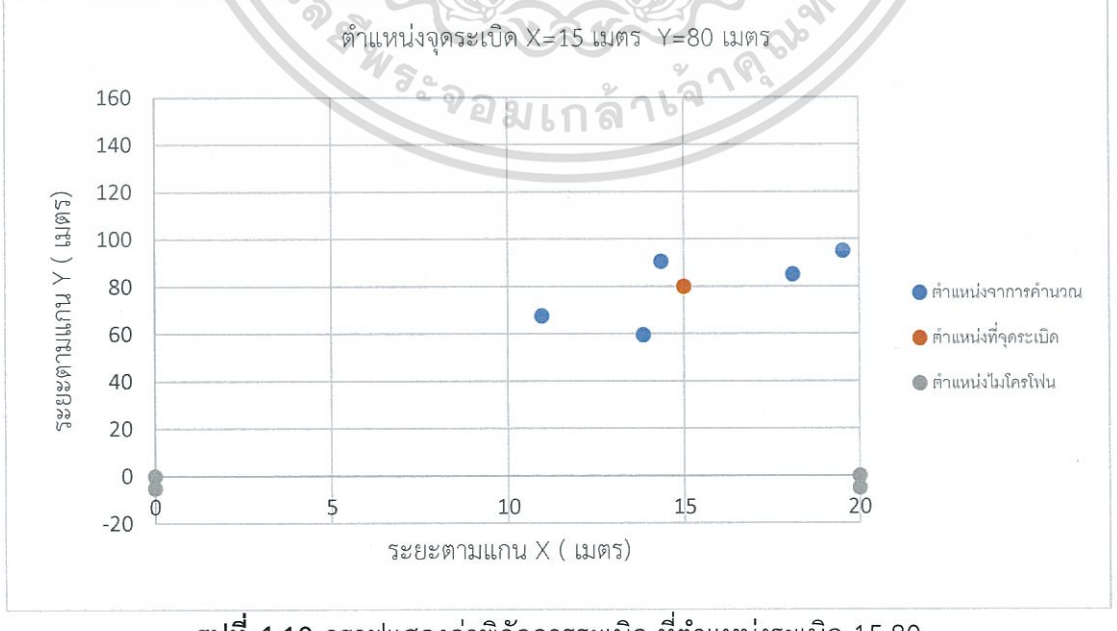
	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
จากมโครโฟน คู่ที่ 1	14.23	14.15	14.19	14.26	14.24
จากมโครโฟน คู่ที่ 2	14.48	14.45	14.47	14.43	14.47

ตาราง 4.13 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 15,80

ผลการหาตำแหน่ง (เมตร)

	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
ตำแหน่ง X	16.42781	14.84072	15.74625	13.20451	15.42681
ตำแหน่ง Y	87.62908	69.56575	78.3631	75.32856	84.05412

ตาราง 4.14 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 15,80



รูปที่ 4.10 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 15,80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งจุดระเบิด ตามแกน X 15 เมตร, ตามแกน Y 80 เมตร
จับโดยใช้ โทรศัพท์สมาร์ทโฟนระบบปฏิบัติการแอนดรอยด์

ความต่างของเวลาจากการอ่านกราฟเสียง (มิลลิวินาที)

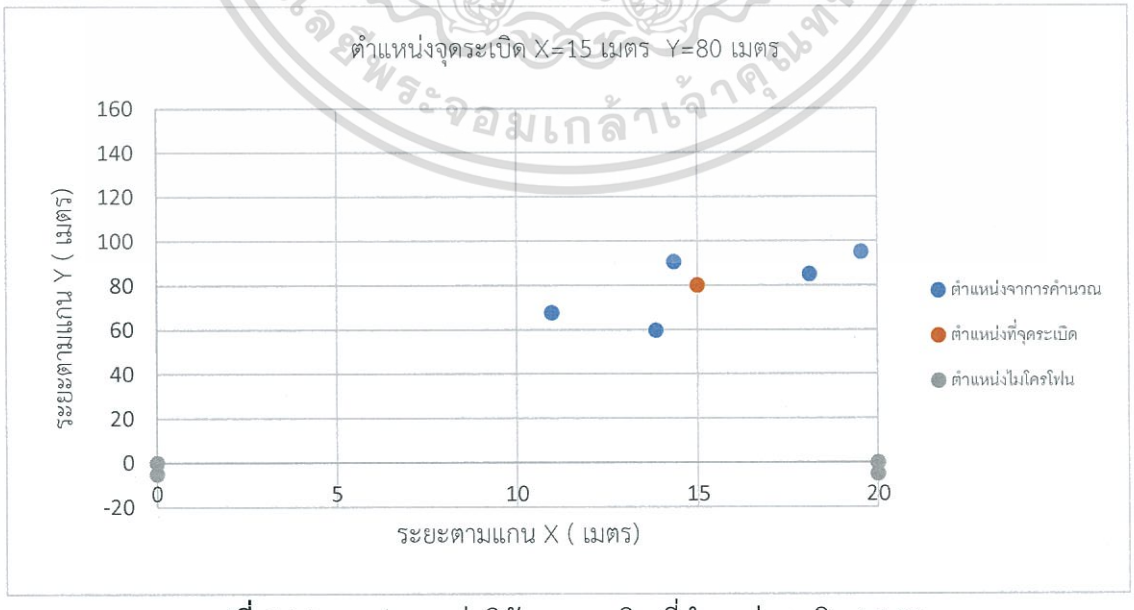
	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
จากไมโครโฟน คู่คู่ที่ 1	14.084	14.304	14.187	14.291	14.156
จากไมโครโฟน คู่ที่ 2	14.409	14.463	14.5	14.355	14.489

ตาราง 4.15 ตารางแสดงค่าผลต่างเวลาของคู่ที่ 1 และ 2 ที่ตำแหน่งระเบิด 15,80

ผลการหาตำแหน่ง (เมตร)

	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5
ตำแหน่ง X	13.84142	14.34984	19.55157	10.96773	18.12011
ตำแหน่ง Y	59.59037	90.57563	95.12159	67.58889	85.11942

ตาราง 4.16 ตารางแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 15,80



รูปที่ 4.11 กราฟแสดงค่าพิกัดการระเบิด ที่ตำแหน่งระเบิด 15,80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

5.1 สรุปผลการทดลอง

ระบบตรวจจับตำแหน่งเสียงระเบิด สามารถทำตามข้อกำหนดดังต่อไปนี้ได้

(1) สามารถใช้สามารถใช้เครื่องโทรศัพท์ประเภทแอนดรอยด์ในการประมวลผลเสียงที่เข้ามาได้ โดยต้องตั้งไมโครโฟนรับเสียงไว้ห่างกันอย่างน้อย 5 เมตร เพื่อให้ลูกคลื่นของเสียงระเบิดที่เข้ามามีความห่างกันมากกว่า 10 mS ระบบจะทำการตรวจวัด Spectrum ของเสียงที่ความถี่ต่ำประมาณ 0-50Hz ซึ่งเป็นองค์ประกอบความถี่ของเสียงระเบิด ซึ่งจะได้เวลาของลูกคลื่นที่1 เป็น t_1 และลูกคลื่นที่ 2 เป็น t_2 ตามลำดับ ความละเอียดของเวลาคือ ไมโครวินาที

(2) ระบบสามารถทำการส่งผ่านข้อมูลเวลาที่ได้ผ่านเครือข่าย Wireless Lan (TCP) เพื่อส่งเวลาที่ได้ไปยังคอมพิวเตอร์เพื่อทำการคำนวณหาความต่างของเวลาและตำแหน่งที่เกิดระเบิด

(3) ระบบตรวจจับสามารถตรวจจับได้เฉพาะเสียงที่มาจากการระเบิดเท่านั้น

(5) ความตลาดเคลื่อนของตำแหน่งที่ตรวจจับได้มากที่สุดคือ 34 เมตร

5.2 ปัญหาและอุปสรรค

(1) อุปกรณ์ตรวจวัด และอุปกรณ์รับสัญญาณไม่ทนต่อสภาพที่ร้อน ฝนตก หรือมีฝุ่นได้ เพราะอาจจะทำให้อุปกรณ์อิเล็กทรอนิกส์ภายในเกิดความเสียหาย ดังนั้นจึงไม่สามารถติดตั้งไว้เป็นเวลานานได้

(2) จากการที่ต้องลดจำนวนอุปกรณ์ประมวลผลเป็น ไมโครโฟน 2 ตัวต่อ 1 อุปกรณ์ประมวลผล ทำให้พื้นที่สามารถตรวจจับได้มีเพียงด้านหน้าของไมโครโฟนเท่านั้น

(3) ในบางครั้งระบบไม่สามารถจับเสียงระเบิดที่เข้ามาได้

(4) ต้องใช้ในที่ที่ไม่มีเสียงรบกวนดังเกินไป

(5) ระบบสามารถตรวจจับไดที่ระยะ 20-100 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

(1) ใช้อุปกรณ์ที่คงทนต่อความร้อนได้ดีหรือมีระบบระบายความร้อน เช่น มีพัดลมระบายอากาศให้อุปกรณ์

(2) พัฒนาระบบวิเคราะห์เสียงให้มีความสามารถมากขึ้นให้สามารถจับเสียงได้ทุกครั้ง

(3) เพิ่มระบบคำนวณที่คอมพิวเตอร์ให้สามารถอ่านค่าอัตโนมัติเมื่ออุปกรณ์ตรวจจับส่งมา

(4) การเพิ่มระยะการตรวจจับสามารถทำได้ เมื่อระเบิดมีความดังมากขึ้น(ใช้ระเบิดจริง)

(5) นำวงจรฟิลเตอร์มากรองความถี่ที่ไม่ต้องการใช้ออกเพื่อให้สามารถใช้ในที่ที่มีเสียงรบกวนได้

(6) หากใช้ในที่ที่ไม่มี Wireless Lan ระบบสามารถส่งผ่านข้องมูลโดยใช้สามจีได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1]. Reto Meier (2009). Professional android application development. Wiley Publishing.

[2]. FFT based simple Spectrum Analyzer with Source Code. สืบค้นเมื่อ 1 ตุลาคม 2559. สืบค้นจาก : http://som-itsolutions.blogspot.com.es/2012/01/fft-based- simple-spectrum-analyzer.html

[3]. Status bar Notification Example in Android. สืบค้นเมื่อ 20 คุลาคม 2559. สืบค้นจาก : http://www.compiletimeerror.com/2013/10/status-bar-notification-example-in.html#.WBnyiS2LTIV

[4]. Android Write Text file. สืบค้นเมื่อ 1 พฤศจิกายน2559. สืบค้นจาก : http://www.thaicreate.com/mobile/android-java-write-text-file.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพผนวก ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ชุดคำสั่งภาษาจาวาที่ใช้ในอุปกรณ์รับส่งข้อมูล

1.1 ส่วนตรวจสอบข้อมูล

```
package com.somitsolutions.android.spectrumanalyzer;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.media.AudioFormat;
import android.media.AudioRecord;
import android.media.MediaRecorder;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.Display;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup.LayoutParams;
import android.view.ViewGroup.MarginLayoutParams;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import ca.uol.aig.fftpack.RealDoubleFFT;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import android.app.Activity;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class SoundRecordAndAnalysisActivity extends Activity implements
OnClickListener{
    int checkpsd[]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
    int frequency = 22000;
    int channelConfiguration = AudioFormat.CHANNEL_CONFIGURATION_MONO;
    int audioEncoding = AudioFormat.ENCODING_PCM_16BIT;
    AudioRecord audioRecord;
    private RealDoubleFFT transformer;
    int blockSize;// = 256;
    Button startStopButton;
    boolean started = false;
    boolean CANCELLED_FLAG = false;
    RecordAudio recordTask;
    ImageView imageViewDisplaySectrum;
    MyImageView imageViewScale;
    Bitmap bitmapDisplaySpectrum;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Canvas canvasDisplaySpectrum;
Paint paintSpectrumDisplay;
Paint paintScaleDisplay;
static SoundRecordAndAnalysisActivity mainActivity;
LinearLayout main;
int width;
int height;
int left_Of_BimapScale;
int left_Of_DisplaySpectrum;
private final static int ID_BITMAPDISPLAYSPECTRUM = 1;
private final static int ID_IMAGEVIEWSCALE = 2;
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    Display display = getWindowManager().getDefaultDisplay();
    //Point size = new Point();
    //display.get(size);
    width = display.getWidth();
    height = display.getHeight();
    blockSize = 256;
}
@SuppressWarnings("deprecation")
private void Notify(String notificationTitle, String notificationMessage) {
    NotificationManager notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
    @SuppressWarnings("deprecation")
    Notification notification = new Notification(R.drawable.ic_launcher, "New
Message", System.currentTimeMillis());
    Intent notificationIntent = new Intent(this,SoundRecordAndAnalysisActivity.class);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, notificationIntent,
0);
        notification.setLatestEventInfo(SoundRecordAndAnalysisActivity.this,
notificationTitle,
                notificationMessage, pendingIntent);
        notificationManager.notify(9999, notification);
    }
    @Override
    public void onWindowFocusChanged (boolean hasFocus) {
        //left_Of_BimapScale = main.getC.getLeft();
        MyImageView  scale = (MyImageView)main.findViewById(ID_IMAGEVIEWSCALE);
        ImageView                    bitmap                    =
(ImageView)main.findViewById(ID_BITMAPDISPLAYSPECTRUM);
        left_Of_BimapScale = scale.getLeft();
        left_Of_DisplaySpectrum = bitmap.getLeft();
    }
    private class RecordAudio extends AsyncTask<Void, double[], Boolean> {
        @Override
        protected Boolean doInBackground(Void... params) {
            int bufferSize = AudioRecord.getMinBufferSize(frequency,
                    channelConfiguration, audioEncoding);
            audioRecord = new AudioRecord(
                    MediaRecorder.AudioSource.DEFAULT, frequency,
                    channelConfiguration, audioEncoding, bufferSize);
            int bufferReadResult;
            short[] buffer = new short[blockSize];
            double[] toTransform = new double[blockSize];
            try {
                audioRecord.startRecording();
            } catch (IllegalStateException e) {
                Log.e("Recording failed", e.toString());
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    while (started) {
        if (isCancelled() || (CANCELLED_FLAG == true)) {
            started = false;
            //publishProgress(cancelledResult);
            Log.d("doInBackground", "Cancelling the RecordTask");
            break;
        } else {
            bufferReadResult = audioRecord.read(buffer, 0, blockSize);
            for (int i = 0; i < blockSize && i < bufferReadResult; i++) {
                toTransform[i] = (double) buffer[i] / 32768.0; // signed 16 bit
            }
            transformer.ft(toTransform);
            publishProgress(toTransform);
        }
    }
    return true;
}
@Override
protected void onProgressUpdate(double[]...progress) {
    Log.e("RecordingProgress", "Displaying in progress");
    Log.d("Test:", Integer.toString(progress[0].length));
    for (int i = 0; i < progress[0].length; i++) {

        int x = 2 * i;
        int downy = (int) (150 - (progress[0][i] * 10));
        int upy = 150;
        canvasDisplaySpectrum.drawLine(x, downy, x, upy, paintSpectrumDisplay);
    }
    int z=0;
    for(int j=0;j < progress[0].length; j++ )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {if(progress[0][j] * 10>=1)
        {z++;}
        }
        if(z>200)
        {   long a= System.currentTimeMillis()% (24 * 60 * 60 * 1000);
            String b = Long.toString(a);
            Notify("baaaaaa",b);
        }
        imageViewDisplaySectrum.invalidate();
    }
    @Override
    protected void onPostExecute(Boolean result) {
        super.onPostExecute(result);
        try{
            audioRecord.stop();
        }
        catch(IllegalStateException e){
            Log.e("Stop failed", e.toString());
        }
        canvasDisplaySpectrum.drawColor(Color.BLACK);
        imageViewDisplaySectrum.invalidate();
    }
}
protected void onCancelled(Boolean result){
    try{
        audioRecord.stop();
    }
    catch(IllegalStateException e){
        Log.e("Stop failed", e.toString());
    }
        /* //recordTask.cancel(true);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Log.d("FFTSpectrumAnalyzer","onCancelled: New Screen");
            Intent intent = new Intent(Intent.ACTION_MAIN);
            intent.addCategory(Intent.CATEGORY_HOME);
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
*/
    }
    public void onClick(View v) {
        if (started == true) {
            //started = false;
            CANCELLED_FLAG = true;
            //recordTask.cancel(true);
            try{
                audioRecord.stop();
            }
            catch(IllegalStateException e){
                Log.e("Stop failed", e.toString());
            }
            startStopButton.setText("Start");
            canvasDisplaySpectrum.drawColor(Color.BLACK);
        }
        else {
            started = true;
            CANCELLED_FLAG = false;
            startStopButton.setText("Stop");
            recordTask = new RecordAudio();
            recordTask.execute();
        }
    }
    static SoundRecordAndAnalysisActivity getMainActivity(){
        return mainActivity;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
public void onStop(){
    super.onStop();
     /* try{
            audioRecord.stop();
        }
        catch(IllegalStateException e){
            Log.e("Stop failed", e.toString());
        }*/
    recordTask.cancel(true);
    Intent intent = new Intent(Intent.ACTION_MAIN);
    intent.addCategory(Intent.CATEGORY_HOME);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent);
}
 public void onStart(){
    super.onStart();
    main = new LinearLayout(this);
    main.setLayoutParams(new
LayoutParams(LayoutParams.MATCH_PARENT,android.view.ViewGroup.LayoutParams.
M ATCH_PARENT));
    main.setOrientation(LinearLayout.VERTICAL);
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
          WindowManager.LayoutParams.FLAG_FULLSCREEN);
    transformer = new RealDoubleFFT(blockSize);
    imageViewDisplaySectrum = new ImageView(this);
      bitmapDisplaySpectrum                                                    =
Bitmap.createBitmap((int)512,(int)300,Bitmap.Config.ARGB_8888);
    LinearLayout.LayoutParams layoutParams_imageViewScale = null;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//Bitmap scaled = Bitmap.createScaledBitmap(bitmapDisplaySpectrum, 320, 480,
true);
canvasDisplaySpectrum = new Canvas(bitmapDisplaySpectrum);
//canvasDisplaySpectrum = new Canvas(scaled);
paintSpectrumDisplay = new Paint();
paintSpectrumDisplay.setColor(Color.GREEN);
imageViewDisplaySectrum.setImageBitmap(bitmapDisplaySpectrum);
   //imageViewDisplaySectrum.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,LinearLayout
.LayoutParams.WRAP_CONTENT));
   LinearLayout.LayoutParams  layoutParams_imageViewDisplaySpectrum=new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,
LinearLayout.LayoutParams.WRAP_CONTENT);
   ((MarginLayoutParams)
layoutParams_imageViewDisplaySpectrum).setMargins(100, 600, 0, 0);

imageViewDisplaySectrum.setLayoutParams(layoutParams_imageViewDisplaySpectru
m);
   layoutParams_imageViewScale=                                    new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,
LinearLayout.LayoutParams.WRAP_CONTENT);
   //layoutParams_imageViewScale.gravity = Gravity.CENTER_HORIZONTAL;
   ((MarginLayoutParams) layoutParams_imageViewScale).setMargins(100, 20, 0,
0);
32
imageViewDisplaySectrum.setId(ID_BITMAPDISPLAYSPECTRUM);
main.addView(imageViewDisplaySectrum);
imageViewScale = new MyImageView(this);
imageViewScale.setLayoutParams(layoutParams_imageViewScale);
imageViewScale.setId(ID_IMAGEVIEWSCALE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        //imageViewScale.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,LinearLayout
.LayoutParams.WRAP_CONTENT));
        main.addView(imageViewScale);
        startStopButton = new Button(this);
        startStopButton.setText("Start");
        startStopButton.setOnClickListener(this);
        startStopButton.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,LinearLayou
t.LayoutParams.WRAP_CONTENT));
        main.addView(startStopButton);
        setContentView(main);
        mainActivity = this;
    }
    @Override
    public void onBackPressed() {
        super.onBackPressed();        try{
            audioRecord.stop();
        }
        catch(IllegalStateException e){
            Log.e("Stop failed", e.toString());
        }
        recordTask.cancel(true);
        Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
    }
    @Override
    protected void onDestroy() {
        // TODO Auto-generated method stub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        super.onDestroy();
        try{
            audioRecord.stop();
        }
        catch(IllegalStateException e){
            Log.e("Stop failed", e.toString());
        }
        recordTask.cancel(true);
        Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);    }
    //Custom Imageview Class
    public class MyImageView extends ImageView {
        Paint paintScaleDisplay;
        Bitmap bitmapScale;
        Canvas canvasScale;
        public MyImageView(Context context) {
            super(context);
            // TODO Auto-generated constructor stub
                bitmapScale                                                    =
Bitmap.createBitmap((int)512,(int)50,Bitmap.Config.ARGB_8888);
            paintScaleDisplay = new Paint();
            paintScaleDisplay.setColor(Color.WHITE);
            paintScaleDisplay.setStyle(Paint.Style.FILL);
            canvasScale = new Canvas(bitmapScale);
            setImageBitmap(bitmapScale);
            invalidate();
        }
        @Override
        protected void onDraw(Canvas canvas)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
            // TODO Auto-generated method stub
            super.onDraw(canvas);
                canvasScale.drawLine(0, 30,  512, 30, paintScaleDisplay);
               for(int i = 0,j = 0; i< 512; i=i+128, j++){
                    for (int k = i; k<(i+128); k=k+16){
                        canvasScale.drawLine(k, 30, k, 25, paintScaleDisplay);
                    }
                    canvasScale.drawLine(i, 40, i, 25, paintScaleDisplay);
                    String text = Integer.toString(j) + " KHz";
                    canvasScale.drawText(text, i, 45, paintScaleDisplay);
                }
                canvas.drawBitmap(bitmapScale, 0, 0, paintScaleDisplay);
        }
    }
}
```

1.2 ส่วนประมวลผลและส่งข้อมูล

```
package com.somitsolutions.android.spectrumanalyzer;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.media.AudioFormat;
import android.media.AudioRecord;
import android.media.MediaRecorder;
import android.os.AsyncTask;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
import android.os.Bundle;
import android.util.Log;
import android.view.Display;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup.LayoutParams;
import android.view.ViewGroup.MarginLayoutParams;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import ca.uol.aig.fftpack.RealDoubleFFT;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import java.util.ArrayList;
import simpletcplibrary.SimpleTCPClient;
import simpletcplibrary.SimpleTCPServer;
import simpletcplibrary.TCPUtils;
import simpletcplibrary.SimpleTCPClient.SendCallback;
import simpletcplibrary.SimpleTCPServer.OnDataReceivedListener;


public class SoundRecordAndAnalysisActivity extends Activity
implementsOnClickListener{
        ouble sum=0;
        int frequency = 20000;
        int channelConfiguration = AudioFormat.CHANNEL_CONFIGURATION_MONO;
        int audioEncoding = AudioFormat.ENCODING_PCM_16BIT;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
AudioRecord audioRecord;
private RealDoubleFFT transformer;
int blockSize=256;// = 256;
Button startStopButton;
boolean started = false;
boolean CANCELLED_FLAG = false;


RecordAudio recordTask;
ImageView imageViewDisplaySectrum;
MyImageView imageViewScale;
Bitmap bitmapDisplaySpectrum;

Canvas canvasDisplaySpectrum;

Paint paintSpectrumDisplay;
Paint paintScaleDisplay;
static SoundRecordAndAnalysisActivity mainActivity;
LinearLayout main;
int width;
int height;
int left_Of_BimapScale;
int left_Of_DisplaySpectrum;
private final static int ID_BITMAPDISPLAYSPECTRUM = 1;
private final static int ID_IMAGEVIEWSCALE = 2;


@Override
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
public void onWindowFocusChanged (boolean hasFocus) {
    //left_Of_BimapScale = main.getC.getLeft();
    MyImageView scale =
(MyImageView)main.findViewById(ID_IMAGEVIEWSCALE);
    ImageView bitmap =
(ImageView)main.findViewById(ID_BITMAPDISPLAYSPECTRUM);
    left_Of_BimapScale = scale.getLeft();
    left_Of_DisplaySpectrum = bitmap.getLeft();
}
@SuppressWarnings("deprecation")
private void Notify(String notificationTitle, String notificationMessage) {
    NotificationManager notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
    @SuppressWarnings("deprecation")
    Notification notification = new Notification(R.drawable.ic_launcher, "New
Message", System.currentTimeMillis());

    Intent notificationIntent = new
Intent(this,SoundRecordAndAnalysisActivity.class);
    PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,
notificationIntent, 0);

    notification.setLatestEventInfo(SoundRecordAndAnalysisActivity.this,
notificationTitle,
            notificationMessage, pendingIntent);
    notificationManager.notify(9999, notification);
}
```

```
private class RecordAudio extends AsyncTask<Void, double[], Boolean> {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
@Override
protected Boolean doInBackground(Void... params) {

    int bufferSize = AudioRecord.getMinBufferSize(frequency,
            channelConfiguration, audioEncoding);
    audioRecord = new AudioRecord(
            MediaRecorder.AudioSource.DEFAULT, frequency,
            channelConfiguration, audioEncoding, bufferSize);
    int bufferReadResult;
    short[] buffer = new short[blockSize];
    double[] toTransform = new double[blockSize];
    try {
        audioRecord.startRecording();
    } catch (IllegalStateException e) {
        Log.e("Recording failed", e.toString());

    }
    while (started) {

        if (isCancelled() || (CANCELLED_FLAG == true)) {

            started = false;
            //publishProgress(cancelledResult);
            Log.d("doInBackground", "Cancelling the RecordTask");
            break;
        } else {
            bufferReadResult = audioRecord.read(buffer, 0, blockSize);

            for (int i = 0; i < blockSize && i < bufferReadResult; i++) {
                toTransform[i] = (double) buffer[i] / 32768.0; // signed 16 bit
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }

            transformer.ft(toTransform);

            publishProgress(toTransform);

        }

    }
    return true;
}
@Override
protected void onProgressUpdate(double[]...progress) {
    Log.e("RecordingProgress", "Displaying in progress");

    Log.d("Test:", Integer.toString(progress[0].length));


   int z=0;
    double t=0;
    for(int j=0;j <2; j++ )
   {
      if((int)(Math.abs(progress[0][j])*1000)>100000)
      sum++;
   }

    if(sum>=1)
    {   long a= System.nanotime();
        String b = Long.toString(a);
        String message = "Device1 : "+ b;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        String ip = "192.168.1.2";
        SimpleTCPClient.send(message, ip, 21111, new SendCallback() {
            public void onSuccess(String tag) {}

            public void onFailed(String tag) {}
        }, "TAG");
        Notify("ALERT",b);
        try { Thread.sleep(0);
        }
        catch (InterruptedException ie) {
            // Handle the exception
        }

    }
//  sum=0;
    imageViewDisplaySectrum.invalidate();


}
@Override
protected void onPostExecute(Boolean result) {
    super.onPostExecute(result);
    try{
        audioRecord.stop();
    }
    catch(IllegalStateException e){
        Log.e("Stop failed", e.toString());

    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            canvasDisplaySpectrum.drawColor(Color.BLACK);
            imageViewDisplaySectrum.invalidate();

        }
    }

    protected void onCancelled(Boolean result){

        try{
            audioRecord.stop();
        }
        catch(IllegalStateException e){
            Log.e("Stop failed", e.toString());

        }
        /* //recordTask.cancel(true);
            Log.d("FFTSpectrumAnalyzer","onCancelled: New Screen");
            Intent intent = new Intent(Intent.ACTION_MAIN);
            intent.addCategory(Intent.CATEGORY_HOME);
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
*/
    }

    public void onClick(View v) {
        if (started == true) {
            //started = false;
            CANCELLED_FLAG = true;
            //recordTask.cancel(true);
            try{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            audioRecord.stop();
        }
        catch(IllegalStateException e){
            Log.e("Stop failed", e.toString());

        }
        startStopButton.setText("Start");

        canvasDisplaySpectrum.drawColor(Color.BLACK);

    }

    else {
        started = true;
        CANCELLED_FLAG = false;
        startStopButton.setText("Stop");
        recordTask = new RecordAudio();
        recordTask.execute();
    }

}
static SoundRecordAndAnalysisActivity getMainActivity(){

    return mainActivity;
}

public void onStop(){
    super.onStop();
     /* try{
          audioRecord.stop();
       }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            catch(IllegalStateException e){
                Log.e("Stop failed", e.toString());

            }*/
        recordTask.cancel(true);
        Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
    }

    public void onStart(){

        super.onStart();
        main = new LinearLayout(this);
        main.setLayoutParams(new
LayoutParams(LayoutParams.MATCH_PARENT,android.view.ViewGroup.LayoutP
arams.MATCH_PARENT));
        main.setOrientation(LinearLayout.VERTICAL);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                WindowManager.LayoutParams.FLAG_FULLSCREEN);

        transformer = new RealDoubleFFT(blockSize);

        imageViewDisplaySectrum = new ImageView(this);
        if(width > 512){
            bitmapDisplaySpectrum                                    =
Bitmap.createBitmap((int)512,(int)300,Bitmap.Config.ARGB_8888);
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else{
    bitmapDisplaySpectrum =
Bitmap.createBitmap((int)256,(int)150,Bitmap.Config.ARGB_8888);
}
LinearLayout.LayoutParams layoutParams_imageViewScale = null;
//Bitmap scaled = Bitmap.createScaledBitmap(bitmapDisplaySpectrum,
320, 480, true);
canvasDisplaySpectrum = new Canvas(bitmapDisplaySpectrum);
//canvasDisplaySpectrum = new Canvas(scaled);
paintSpectrumDisplay = new Paint();
paintSpectrumDisplay.setColor(Color.GREEN);
imageViewDisplaySectrum.setImageBitmap(bitmapDisplaySpectrum);
if(width >512){
    //imageViewDisplaySectrum.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,LinearLa
yout.LayoutParams.WRAP_CONTENT));
    LinearLayout.LayoutParams
layoutParams_imageViewDisplaySpectrum=new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,
LinearLayout.LayoutParams.WRAP_CONTENT);
    ((MarginLayoutParams)
layoutParams_imageViewDisplaySpectrum).setMargins(100, 600, 0, 0);

imageViewDisplaySectrum.setLayoutParams(layoutParams_imageViewDisplayS
pectrum);
    layoutParams_imageViewScale= new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,
LinearLayout.LayoutParams.WRAP_CONTENT);
    //layoutParams_imageViewScale.gravity =
Gravity.CENTER_HORIZONTAL;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ((MarginLayoutParams) layoutParams_imageViewScale).setMargins(100,
20, 0, 0);

    }

    else if ((width >320) && (width<512)){
        LinearLayout.LayoutParams
layoutParams_imageViewDisplaySpectrum=new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,
LinearLayout.LayoutParams.WRAP_CONTENT);
        ((MarginLayoutParams)
layoutParams_imageViewDisplaySpectrum).setMargins(60, 250, 0, 0);
        //layoutParams_imageViewDisplaySpectrum.gravity                =
Gravity.CENTER_HORIZONTAL;

imageViewDisplaySectrum.setLayoutParams(layoutParams_imageViewDisplayS
pectrum);

        //imageViewDisplaySectrum.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,LinearLa
yout.LayoutParams.WRAP_CONTENT));
        layoutParams_imageViewScale=new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,
LinearLayout.LayoutParams.WRAP_CONTENT);
        ((MarginLayoutParams)  layoutParams_imageViewScale).setMargins(60,
20, 0, 100);
        //layoutParams_imageViewScale.gravity                =
Gravity.CENTER_HORIZONTAL;
    }

    else if (width < 320){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                /*LinearLayout.LayoutParams
layoutParams_imageViewDisplaySpectrum=new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
LinearLayout.LayoutParams.WRAP_CONTENT);
            ((MarginLayoutParams)
layoutParams_imageViewDisplaySpectrum).setMargins(30, 100, 0, 100);

imageViewDisplaySectrum.setLayoutParams(layoutParams_imageViewDisplayS
pectrum);*/
            imageViewDisplaySectrum.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,Linea
rLayout.LayoutParams.WRAP_CONTENT));
            layoutParams_imageViewScale=new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,
LinearLayout.LayoutParams.WRAP_CONTENT);
            //layoutParams_imageViewScale.gravity = Gravity.CENTER;
        }
        imageViewDisplaySectrum.setId(ID_BITMAPDISPLAYSPECTRUM);
        main.addView(imageViewDisplaySectrum);

        imageViewScale = new MyImageView(this);
        imageViewScale.setLayoutParams(layoutParams_imageViewScale);
        imageViewScale.setId(ID_IMAGEVIEWSCALE);

        //imageViewScale.setLayoutParams(new
 LinearLayout.LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,LinearLa
yout.LayoutParams.WRAP_CONTENT));
        main.addView(imageViewScale);

        startStopButton = new Button(this);
        startStopButton.setText("Start");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        startStopButton.setOnClickListener(this);
        startStopButton.setLayoutParams(new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,Linea
rLayout.LayoutParams.WRAP_CONTENT));

        main.addView(startStopButton);

        setContentView(main);

        mainActivity = this;

    }
    @Override
    public void onBackPressed() {
        super.onBackPressed();

        try{
            audioRecord.stop();
        }
        catch(IllegalStateException e){
            Log.e("Stop failed", e.toString());

        }
        recordTask.cancel(true);
        Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
    }

    @Override
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
    try{
        audioRecord.stop();
    }
    catch(IllegalStateException e){
        Log.e("Stop failed", e.toString());

    }
    recordTask.cancel(true);
    Intent intent = new Intent(Intent.ACTION_MAIN);
    intent.addCategory(Intent.CATEGORY_HOME);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(intent);
}
//Custom Imageview Class
public class MyImageView extends ImageView {
    Paint paintScaleDisplay;
    Bitmap bitmapScale;
    Canvas canvasScale;
    public MyImageView(Context context) {
        super(context);
        // TODO Auto-generated constructor stub
        if(width >512){
            bitmapScale =
Bitmap.createBitmap((int)512,(int)50,Bitmap.Config.ARGB_8888);
        }
        else{
            bitmapScale =
Bitmap.createBitmap((int)256,(int)50,Bitmap.Config.ARGB_8888);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }

    paintScaleDisplay = new Paint();
    paintScaleDisplay.setColor(Color.WHITE);
    paintScaleDisplay.setStyle(Paint.Style.FILL);

    canvasScale = new Canvas(bitmapScale);

    setImageBitmap(bitmapScale);
    invalidate();
}
@Override
protected void onDraw(Canvas canvas)
{
    // TODO Auto-generated method stub
    super.onDraw(canvas);

    if(width > 512){
        canvasScale.drawLine(0, 30,  512, 30, paintScaleDisplay);
        for(int i = 0,j = 0; i< 512; i=i+128, j++){
            for (int k = i; k<(i+128); k=k+16){
                canvasScale.drawLine(k, 30, k, 25, paintScaleDisplay);
            }
            canvasScale.drawLine(i, 40, i, 25, paintScaleDisplay);
            String text = Integer.toString(j) + " KHz";
            canvasScale.drawText(text, i, 45, paintScaleDisplay);
        }
        canvas.drawBitmap(bitmapScale, 0, 0, paintScaleDisplay);
    }
    else if ((width >320) && (width<512)){
        canvasScale.drawLine(0, 30, 0 + 256, 30, paintScaleDisplay);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                for(int i = 0,j = 0; i<256; i=i+64, j++){
                    for (int k = i; k<(i+64); k=k+8){
                        canvasScale.drawLine(k, 30, k, 25, paintScaleDisplay);
                    }
                    canvasScale.drawLine(i, 40, i, 25, paintScaleDisplay);
                    String text = Integer.toString(j) + " KHz";
                    canvasScale.drawText(text, i, 45, paintScaleDisplay);
                }
                canvas.drawBitmap(bitmapScale, 0, 0, paintScaleDisplay);
            }

            else if (width <320){
                canvasScale.drawLine(0, 30,  256, 30, paintScaleDisplay);
                for(int i = 0,j = 0; i<256; i=i+64, j++){
                    for (int k = i; k<(i+64); k=k+8){
                        canvasScale.drawLine(k, 30, k, 25, paintScaleDisplay);
                    }
                    canvasScale.drawLine(i, 40, i, 25, paintScaleDisplay);
                    String text = Integer.toString(j) + " KHz";
                    canvasScale.drawText(text, i, 45, paintScaleDisplay);
                }
                canvas.drawBitmap(bitmapScale, 0, 0, paintScaleDisplay);
            }
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โค้ดสมการการหาพิกัดด้วยโปรแกรม Matlab

```
syms x y
t1=;
t2=;
eqn1 =( sqrt((x-0)^2 + (y-5)^2) -sqrt((x-0)^2 + (y-0)^2) )==345*(-t1);
eqn2 = (sqrt((x-20)^2 + (y-5)^2)- sqrt((x-20)^2 + (y-0)^2))==345*(-t2);
sol = solve([eqn1, eqn2], [x, y]);
xSol = sol.x
ySol = sol.y
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Android Studio Development

Essentials

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Chapter 3

3. Creating an Example Android App in Android Studio

The preceding chapters of this book have covered the steps necessary to configure an environment suitable for the development of Android applications using the Android Studio IDE. Before moving on to slightly more advanced topics, now is a good time to validate that all of the required development packages are installed and functioning correctly. The best way to achieve this goal is to create an Android application and compile and run it. This chapter will cover the creation of a simple Android application project using Android Studio. Once the project has been created, a later chapter will explore the use of the Android emulator environment to perform a test run of the application.

3.1 Creating a New Android Project

The first step in the application development process is to create a new project within the Android Studio environment. Begin, therefore, by launching Android Studio so that the "Welcome to Android Studio" screen appears as illustrated in Figure 3-1:

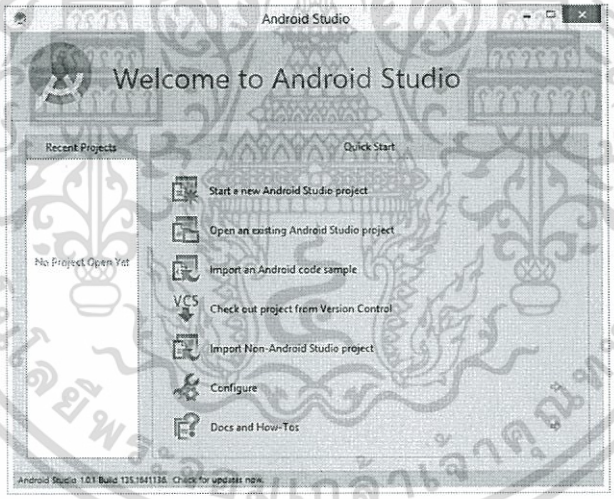


Figure 3-1

Once this window appears, Android Studio is ready for a new project to be created. To create the new project, simply click on the *Start a new Android Studio project* option to display the first screen of the *New Project* wizard as shown in Figure 3-2:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

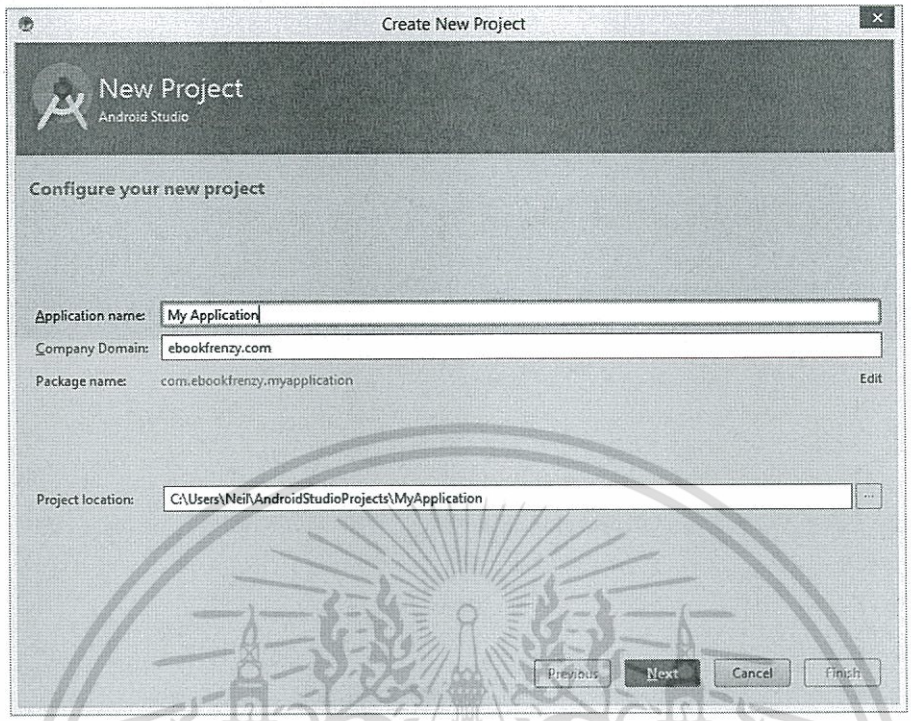


Figure 3-2

3.2 Defining the Project and SDK Settings

In the *New Project* window, set the *Application name* field to *AndroidSample*. The application name is the name by which the application will be referenced and identified within Android Studio and is also the name that will be used when the completed application goes on sale in the Google Play store.

The *Package Name* is used to uniquely identify the application within the Android application ecosystem. It should be based on the reversed URL of your domain name followed by the name of the application. For example, if your domain is *www.mycompany.com*, and the application has been named *AndroidSample*, then the package name might be specified as follows:

```
com.mycompany.androidsample
```

If you do not have a domain name, you may also use *ebookfrenzy.com* for the purposes of testing, though this will need to be changed before an application can be published:

```
com.ebookfrenzy.androidsample
```

The *Project location* setting will default to a location in the folder named *AndroidStudioProjects* located in your home directory and may be changed by clicking on the button to the right of the text field containing the current path setting.

Click Next to proceed. On the form factors screen, enable the *Phone and Tablet* option and set the minimum SDK setting to API 8: Android 2.2 (Froyo). The reason for selecting an older SDK release is that this ensures that the finished application will be able to run on the widest possible range of Android devices. The higher the minimum SDK selection, the more the application will be restricted to newer Android devices. A useful chart (Figure 3-3) can be viewed by clicking on the *Help me choose* link. This outlines the various SDK versions and API levels available for use and the percentage of Android devices in the marketplace on which the application will run if that SDK is used as the minimum level. In general it should only be necessary to select a more

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

recent SDK when that release contains a specific feature that is required for your application. To help in the decision process, selecting an API level from the chart will display the features that are supported at that level.

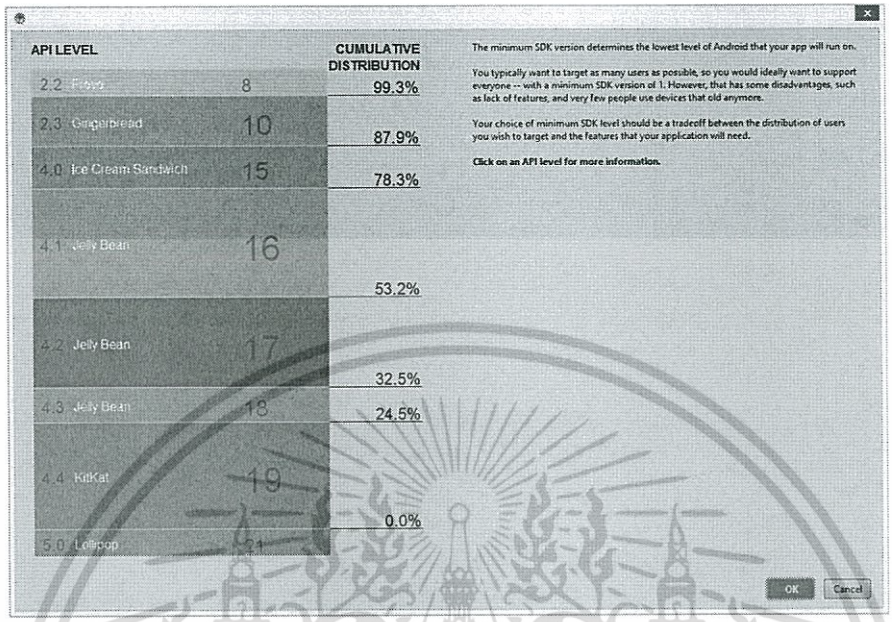


Figure 3-3

Since the project is not intended for Google TV, Google Glass or wearable devices, leave the remaining options disabled before clicking *Next*.

3.3 Creating an Activity

The next step is to define the type of initial activity that is to be created for the application. A range of different activity types is available when developing Android applications. The *Master/Detail Flow* option will be covered in a later chapter. For the purposes of this example, however, simply select the option to create a *Blank Activity*.

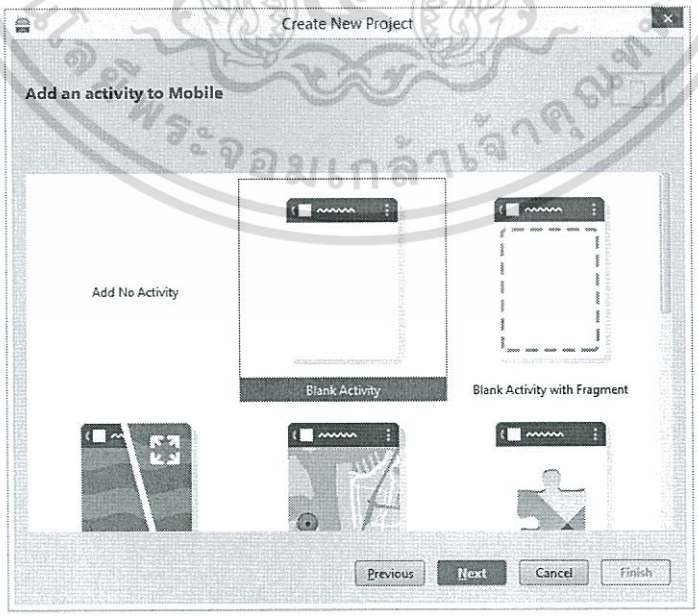


Figure 3-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

With the Blank Activity option selected, click *Next*. On the final screen (Figure 3-5) name the activity and title *AndroidSampleActivity*. The activity will consist of a single user interface screen layout which, for the purposes of this example, should be named *activity_android_sample* as shown in Figure 3-5 and with a menu resource named *menu_android_sample*:

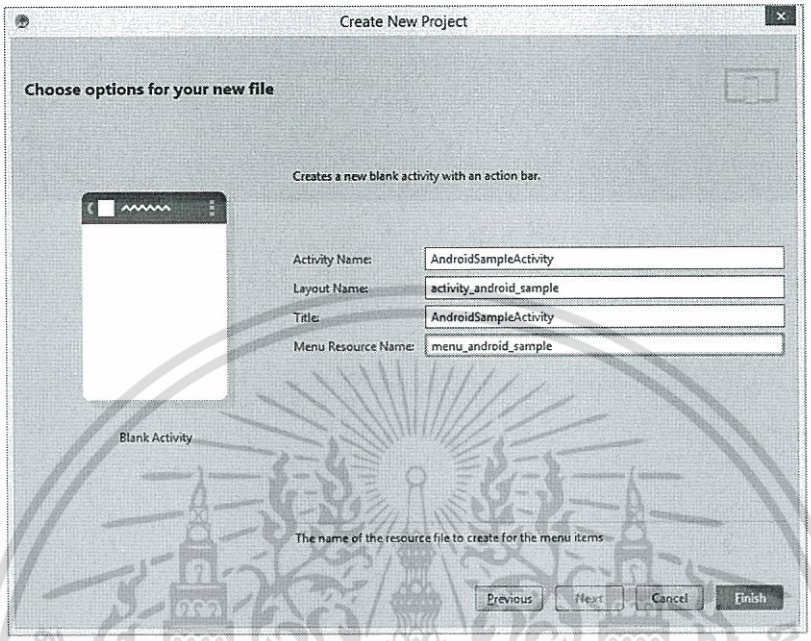


Figure 3-5

Finally, click on *Finish* to initiate the project creation process.

3.4 **Modifying the Example Application**

At this point, Android Studio has created a minimal example application project and opened the main window.

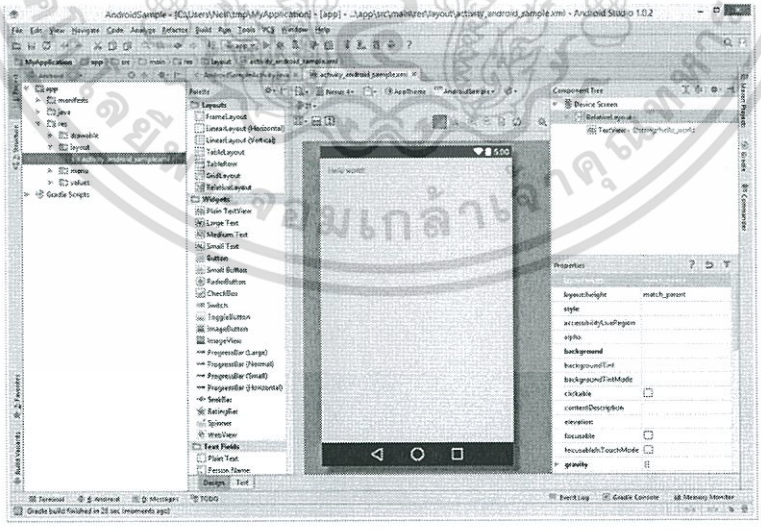


Figure 3-6

The newly created project and references to associated files are listed in the *Project* tool window located on the left hand side of the main project window. The Project tool window has a number of modes in which information can be displayed. By default,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

this panel will be in *Android* mode. This setting is controlled by the drop down menu at the top of the panel as highlighted in Figure 3-6. If the panel is not currently in Android mode, click on this menu and switch to Android mode:

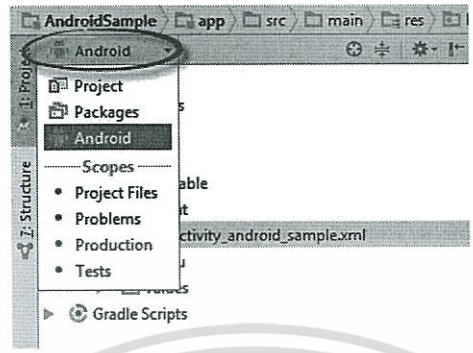


Figure 3-7

The example project created for us when we selected the option to create an activity consists of a user interface containing a label that will read "Hello World" when the application is executed.

The next step in this tutorial is to modify the user interface of our application so that it displays a larger text view object with a different message to the one provided for us by Android Studio.

The user interface design for our activity is stored in a file named *activity_android_sample.xml* which, in turn, is located under *app -> res -> layout* in the project file hierarchy. Using the Project tool window, locate this file as illustrated in Figure 3-8:

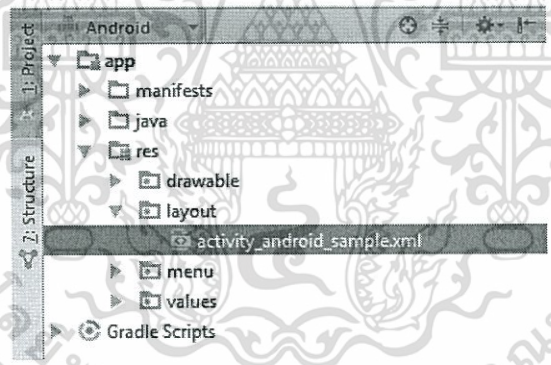


Figure 3-8

Once located, double click on the file to load it into the User Interface Designer tool which will appear in the center panel of the Android Studio main window:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Creating an Example Android App in Android Studio

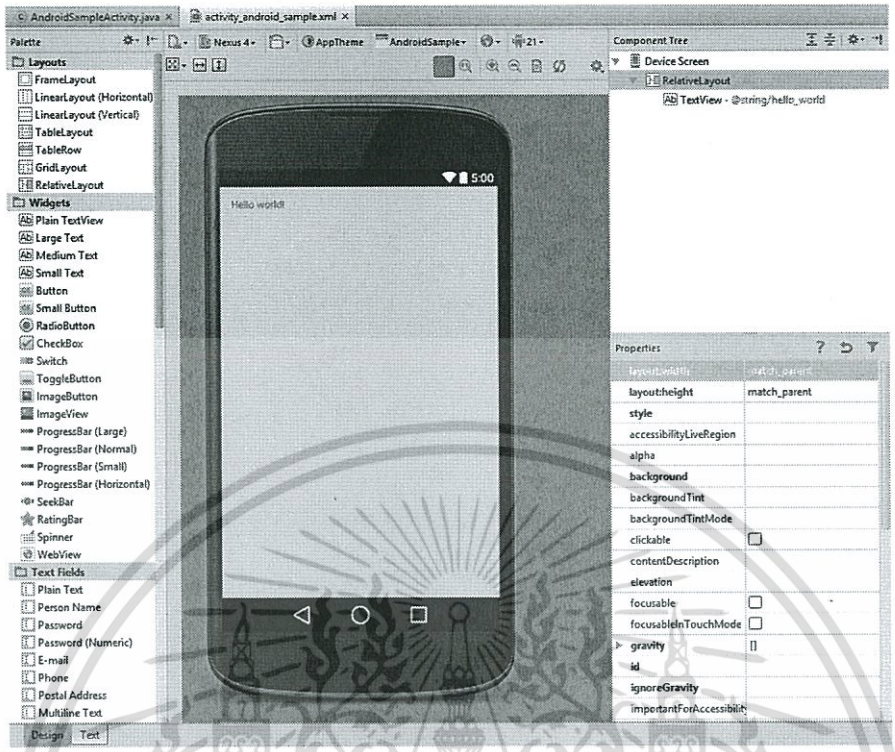


Figure 3-9

In the toolbar across the top of Designer window is a menu currently set to *Nexus 4* which is reflected in the visual representation of the device within the Designer panel. A wide range of other device options are available for selection by clicking on this menu.

To change the orientation of the device representation between landscape and portrait simply use the drop down menu immediately to the right of the device selection menu showing the icon.

As can be seen in the device screen, the layout already includes a label that displays a Hello World! message. Running down the left hand side of the panel is a palette containing different categories of user interface components that may be used to construct a user interface, such as buttons, labels and text fields. It should be noted, however, that not all user interface components are obviously visible to the user. One such category consists of *layouts*. Android supports a variety of different layouts that provide different levels of control over how visual user interface components are positioned and managed on the screen. Though it is difficult to tell from looking at the visual representation of the user interface, the current design has been created using a RelativeLayout. This can be confirmed by reviewing the information in the *Component Tree* panel which, by default, is located in the upper right hand corner of the Designer panel and is shown in Figure 3-10:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

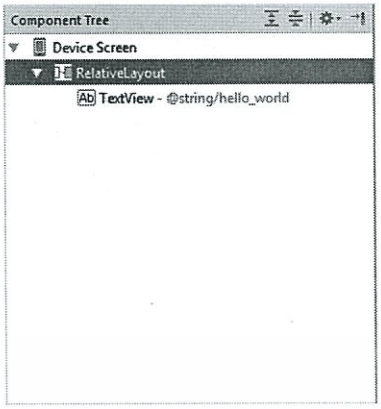


Figure 3-10

As we can see from the component tree hierarchy, the user interface consists of a RelativeLayout parent with a single child in the form of a TextView object.

The first step in modifying the application is to delete the TextView component from the design. Begin by clicking on the TextView object within the user interface view so that it appears with a blue border around it. Once selected, press the Delete key on the keyboard to remove the object from the layout.

In the Palette panel, locate the *Widgets* category. Click and drag the *Large Text* object and drop it in the center of the user interface design when the green marker lines appear to indicate the center of the display:

Figure 3-11

The Android Studio Designer tool also provides an alternative to dragging and dropping components from the palette on to the design layout. Components may also be added by selecting the required object from the palette and then simply clicking on the layout at the location where the component is to be placed.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The next step is to change the text that is currently displayed by the TextView component. Double click on the object in the design layout to display the text and id editing panel as illustrated in Figure 3-12. Within the panel, change the text property from "Large Text" to "Welcome to Android Studio".

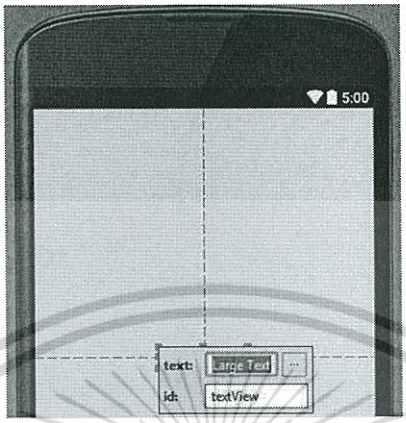


Figure 3-12

At this point it is important to explain the light bulb next to the TextView object in the layout. This indicates a possible problem and provides some recommended solutions. Clicking on the icon in this instance informs us that the problem is as follows:

```
[I18N] Hardcoded string "Welcome to Android Studio", should use @string resource
```

This I18N message is informing us that a potential issue exists with regard to the future internationalization of the project ("I18N" comes from the fact that the word "internationalization" begins with an "I", ends with an "N" and has 18 letters in between). The warning is reminding us that when developing Android applications, attributes and values such as text strings should be stored in the form of *resources* wherever possible. Doing so enables changes to the appearance of the application to be made by modifying resource files instead of changing the application source code. This can be especially valuable when translating a user interface to a different spoken language. If all of the text in a user interface is contained in a single resource file, for example, that file can be given to a translator who will then perform the translation work and return the translated file for inclusion in the application. This enables multiple languages to be targeted without the necessity for any source code changes to be made. In this instance, we are going to create a new resource named *welcomestring* and assign to it the string "Welcome to Android Studio".

Click on the arrow to the right of the warning message to display the menu of possible solutions (Figure 3-13).

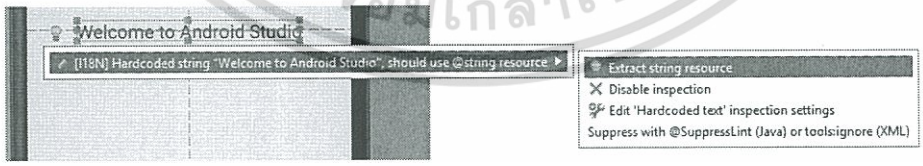


Figure 3-13

From the menu, select the *Extract string resource* option to display the *Extract Resource* dialog. In this dialog, enter *welcomestring* into the *Resource name:* field before clicking on *OK*. The string is now stored as a resource in the *app -> res -> values -> strings.xml* file.

3.5 Reviewing the Layout and Resource Files

Before moving on to the next chapter, we are going to look at some of the internal aspects of user interface design and resource handling. In the previous section, we made some changes to the user interface by modifying the *activity_android_sample.xml*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

file using the UI Designer tool. In fact, all that the Designer was doing was providing a user-friendly way to edit the underlying XML content of the file. In practice, there is no reason why you cannot modify the XML directly in order to make user interface changes and, in some instances, this may actually be quicker than using the Designer tool. At the bottom of the Designer panel are two tabs labeled *Design* and *Text* respectively. To switch to the XML view simply select the *Text* tab as shown in Figure 3-14:

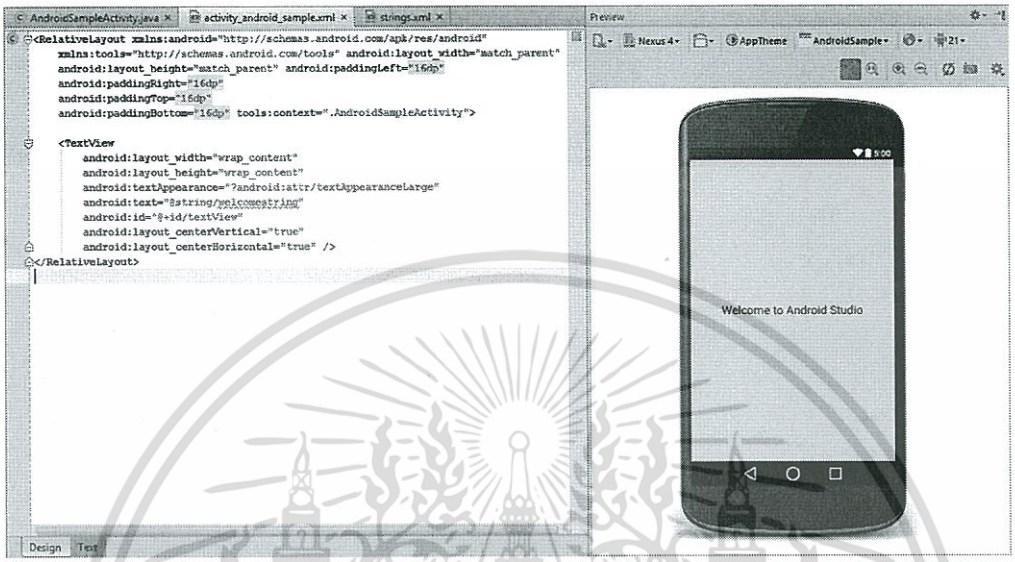


Figure 3-14

As can be seen from the structure of the XML file, the user interface consists of the RelativeLayout component, which in turn, is the parent of the TextView object. We can also see that the *text* property of the TextView is set to our *welcomestring* resource. Although varying in complexity and content, all user interface layouts are structured in this hierarchical, XML based way.

One of the more powerful features of Android Studio can be found to the right hand side of the XML editing panel. This is the Preview panel and shows the current visual state of the layout. As changes are made to the XML layout, these will be reflected in the preview panel. To see this in action, modify the XML layout to change the background color of the RelativeLayout to a shade of red as follows:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".AndroidSampleActivity"
    android:background="#ff2438">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/welcomestring"
        android:id="@+id/textView"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
</RelativeLayout>
```

Note that the color of the preview changes in real-time to match the new setting in the XML file. Note also that a small red square appears in the left hand margin (also referred to as the *gutter*) of the XML editor next to the line containing the color setting. This is a visual cue to the fact that the color red has been set on a property. Change the color value to #a0ff28 and note that both the small square in the margin and the preview change to green.

Finally, use the Project view to locate the *app -> res -> values -> strings.xml* file and double click on it to load it into the editor. Currently the XML should read as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">AndroidSample</string>
    <string name="hello_world">Hello world!</string>
    <string name="action_settings">Settings</string>
    <string name="welcomestring">Welcome to Android Studio</string>

</resources>
```

As a demonstration of resources in action, change the string value currently assigned to the *welcomestring* resource and then return to the Designer by selecting the tab for the layout file in the editor panel. Note that the layout has picked up the new resource value for the welcome string.

There is also a quick way to access the value of a resource referenced in an XML file. With the Designer tool in Text mode, click on the "@string/welcomestring" property setting so that it highlights and then press Ctrl+B on the keyboard. Android Studio will subsequently open the *strings.xml* file and take you to the line in that file where this resource is declared. Use this opportunity to revert the string resource back to the original "Welcome to Android Studio" text.

3.6 Previewing the Layout

So far in this chapter, the layout has only been previewed on a representation of the Nexus 4 device. As previously discussed, the layout can be tested for other devices by making selections from the device menu in the toolbar across the top edge of the Designer panel. Another useful option provided by this menu is *Preview All Screen Sizes* which, when selected, shows the layout in all currently configured device configurations as demonstrated in Figure 3-15:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

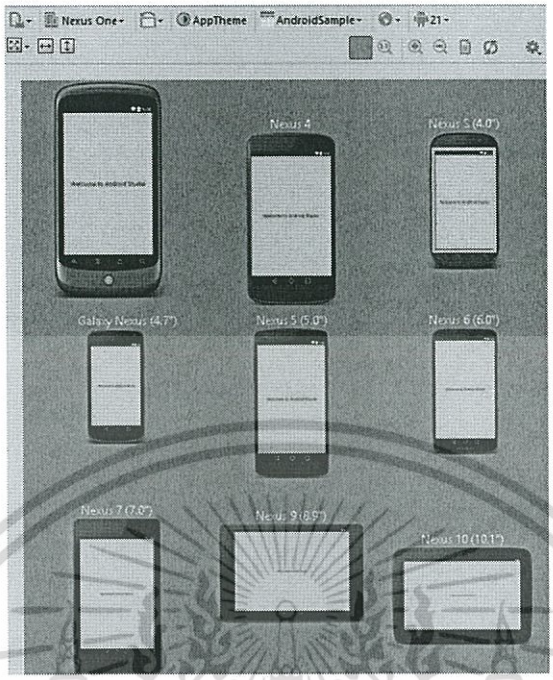


Figure 3-15

To revert to a single preview layout, select the device menu once again, this time choosing the *Remove Previews* option.

3.7 Summary

Whilst not excessively complex, a number of steps are involved in setting up an Android development environment. Having performed those steps, it is worth working through a simple example to make sure the environment is correctly installed and configured. In this chapter, we have created a simple application and then used the Android Studio UI Designer to modify the user interface layout. In doing so, we explored the importance of using resources wherever possible, particularly in the case of string values, and briefly touched on the topic of layouts. Finally, we looked at the underlying XML that is used to store the user interface designs of Android applications.

Whilst it is useful to be able to preview a layout from within the Android Studio Designer tool, there is no substitute for testing an application by compiling and running it. In a later chapter entitled *Creating an Android Virtual Device (AVD) in Android Studio*, the steps necessary to set up an emulator for testing purposes will be covered in detail. Before running the application, however, the next chapter will take a small detour to provide a guided tour of the Android Studio user interface.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTRODUCTION TO MATLAB FOR ENGINEERING STUDENTS

David Houcque
Northwestern University

(version 1.2, August 2005)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

recommend the following books: [2], [3], [4], [5], [6], [7], [8], and [9]. They are excellent in their specific applications.

1.2 Basic features

As we mentioned earlier, the following TUTORIAL lessons are designed to get you started quickly in MATLAB. The lessons are intended to make you familiar with the basics of MATLAB. We urge you to complete the EXERCISES given at the end of each lesson.

1.3 A minimum MATLAB session

The goal of this *minimum* session (also called *starting* and *exiting* sessions) is to learn the first steps:

- How to log on
- Invoke MATLAB
- Do a few simple calculations
- How to quit MATLAB

1.3.1 Starting MATLAB

After logging into your account, you can enter MATLAB by double-clicking on the MATLAB shortcut *icon* (MATLAB 7.0.4) on your Windows desktop. When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains *other* windows. The major tools within or accessible from the desktop are:

- The COMMAND WINDOW
- The COMMAND HISTORY
- The WORKSPACE
- The CURRENT DIRECTORY
- The HELP BROWSER
- The START button

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

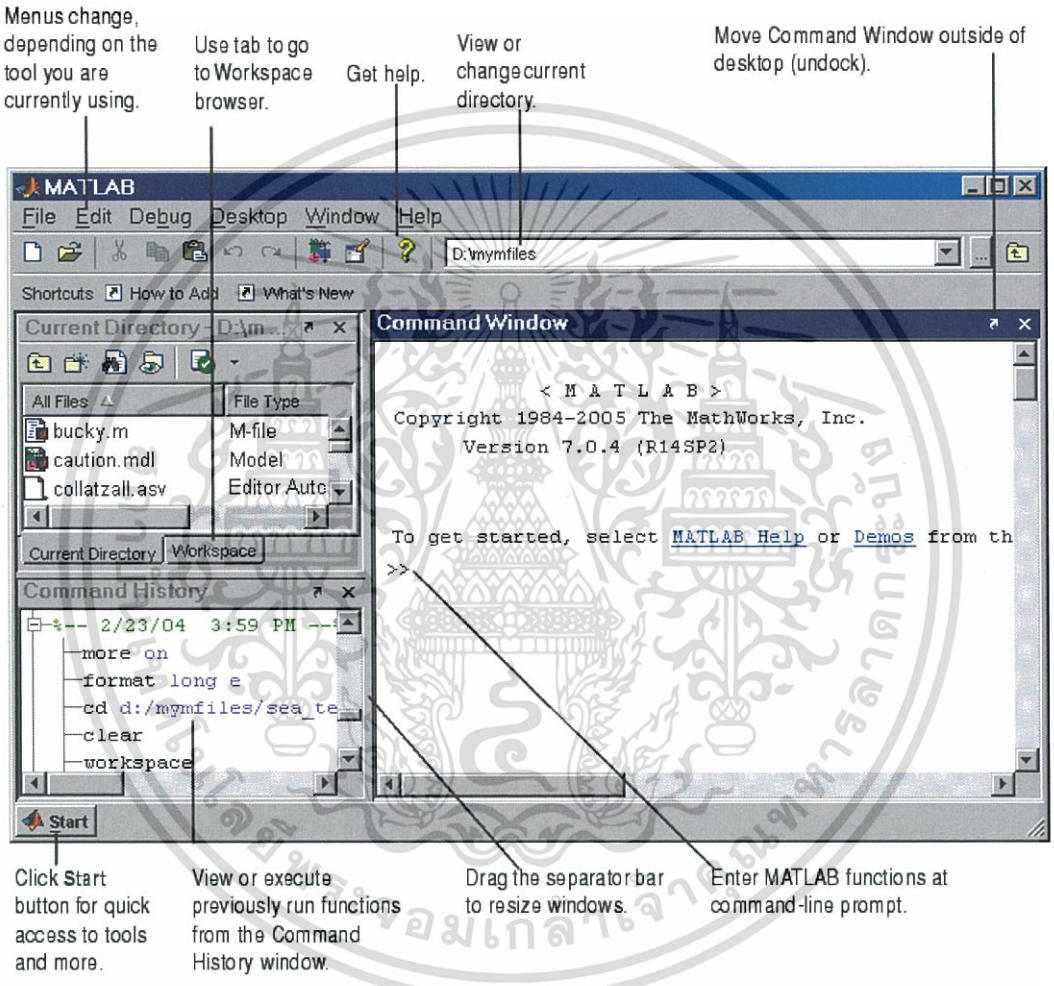


Figure 1.1: The graphical interface to the MATLAB workspace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

When MATLAB is started for the first time, the screen looks like the one that shown in the Figure 1.1. This illustration also shows the default configuration of the MATLAB desktop. You can customize the arrangement of tools and documents to suit your needs.

Now, we are interested in doing some simple calculations. We will assume that you have sufficient understanding of your computer under which MATLAB is being run.

You are now faced with the MATLAB desktop on your computer, which contains the prompt (>>) in the Command Window. Usually, there are 2 types of prompt:

```
>>          for full version
EDU>        for educational version
```

NOTE: To simplify the notation, we will use this prompt, >>, as a standard prompt sign, though our MATLAB version is for educational purpose.

1.3.2 Using MATLAB as a calculator

As an example of a simple interactive calculation, just type the expression you want to evaluate. Let's start at the very beginning. For example, let's suppose you want to calculate the expression, $1 + 2 \times 3$. You type it at the prompt command (>>) as follows,

```
>> 1+2*3
ans =
   7
```

You will have noticed that if you do not specify an output variable, MATLAB uses a default variable `ans`, short for `answer`, to store the results of the current calculation. Note that the variable `ans` is created (or overwritten, if it is already existed). To avoid this, you may assign a value to a variable or output argument name. For example,

```
>> x = 1+2*3
x  =
   7
```

will result in `x` being given the value $1 + 2 \times 3 = 7$. This variable name can always be used to refer to the results of the previous computations. Therefore, computing $4x$ will result in

```
>> 4*x
ans  =
     28.0000
```

Before we conclude this minimum session, Table 1.1 gives the partial list of arithmetic operators.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1.1: Basic arithmetic operators

SYMBOL	OPERATION	EXAMPLE
$+$	Addition	$2+3$
$-$	Subtraction	$2-3$
$*$	Multiplication	$2*3$
$/$	Division	$2/3$

1.3.3 Quitting MATLAB

To end your MATLAB session, type **quit** in the Command Window, or select **File** $\longrightarrow$ **Exit MATLAB** in the desktop main menu.

1.4 Getting started

After learning the minimum MATLAB session, we will now learn to use some additional operations.

1.4.1 Creating MATLAB variables

MATLAB variables are created with an assignment statement. The syntax of variable assignment is

```
variable name = a value (or an expression)
```

For example,

```
>> x = expression
```

where `expression` is a combination of numerical values, mathematical operators, variables, and function calls. On other words, `expression` can involve:

- manual entry
- built-in functions
- user-defined functions

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.2 Overwriting variable

Once a variable has been created, it can be reassigned. In addition, if you do not wish to see the intermediate results, you can suppress the numerical output by putting a semicolon (;) at the end of the line. Then the sequence of commands looks like this:

```
>> t = 5;
>> t = t+1
t   =
    6
```

1.4.3 Error messages

If we enter an expression incorrectly, MATLAB will return an error message. For example, in the following, we left out the multiplication sign, *, in the following expression

```
>> x = 10;
>> 5x
??? 5x
    |
Error: Unexpected MATLAB expression.
```

1.4.4 Making corrections

To make corrections, we can, of course retype the expressions. But if the expression is lengthy, we make more mistakes by typing a second time. A previously typed command can be recalled with the up-arrow key ↑. When the command is displayed at the command prompt, it can be modified if needed and executed.

1.4.5 Controlling the hierarchy of operations or precedence

Let's consider the previous arithmetic operation, but now we will include *parentheses*. For example, $1 + 2 \times 3$ will become $(1 + 2) \times 3$

```
>> (1+2)*3
ans   =
      9
```

and, from previous example

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
>> 1+2*3
ans    =
       7
```

By adding parentheses, these two expressions give different results: `9` and `7`.

The order in which MATLAB performs arithmetic operations is exactly that taught in high school algebra courses. *Exponentiations* are done *first*, followed by *multiplications* and *divisions*, and finally by *additions* and *subtractions*. However, the standard order of precedence of arithmetic operations can be changed by inserting *parentheses*. For example, the result of $1+2\times3$ is quite different than the similar expression with parentheses $(1+2)\times3$. The results are 7 and 9 respectively. Parentheses can always be used to overrule *priority*, and their use is recommended in some complex expressions to avoid ambiguity.

Therefore, to make the evaluation of expressions unambiguous, MATLAB has established a series of rules. The order in which the arithmetic operations are evaluated is given in Table 1.2. MATLAB arithmetic operators obey the same *precedence* rules as those in

Table 1.2: Hierarchy of arithmetic operations

PRECEDENCE	MATHEMATICAL OPERATIONS
First	The contents of all parentheses are evaluated first, starting from the innermost parentheses and working outward.
Second	All exponentials are evaluated, working from left to right
Third	All multiplications and divisions are evaluated, working from left to right
Fourth	All additions and subtractions are evaluated, starting from left to right

most computer programs. For operators of *equal* precedence, evaluation is from *left* to *right*. Now, consider another example:

$$\frac{1}{2+3^2}+\frac{4}{5}\times\frac{6}{7}$$

In MATLAB, it becomes

```
>> 1/(2+3^2)+4/5*6/7
ans  =
     0.7766
```

or, if parentheses are missing,

```
>> 1/2+3^2+4/5*6/7
ans  =
     10.1857
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

So here what we get: two different results. Therefore, we want to emphasize the importance of precedence rule in order to avoid ambiguity.

1.4.6 Controlling the appearance of floating point number

MATLAB by default displays only 4 decimals in the result of the calculations, for example -163.6667, as shown in above examples. However, MATLAB does numerical calculations in *double* precision, which is 15 digits. The command `format` controls how the results of computations are displayed. Here are some examples of the different formats together with the resulting outputs.

```
>> format short
>> x=-163.6667
```

If we want to see all 15 digits, we use the command `format long`

```
>> format long
>> x= -1.636666666666667e+002
```

To return to the standard format, enter `format short`, or simply `format`.

There are several other formats. For more details, see the MATLAB documentation, or type `help format`.

NOTE - Up to now, we have let MATLAB repeat everything that we enter at the prompt (>>). Sometimes this is not quite useful, in particular when the output is pages en length. To prevent MATLAB from echoing what we type, simply enter a semicolon (;) at the end of the command. For example,

```
>> x=-163.6667;
```

and then ask about the value of `x` by typing,

```
>> x
x   =
    -163.6667
```

1.4.7 Managing the workspace

The contents of the workspace persist between the executions of separate commands. Therefore, it is possible for the results of one problem to have an effect on the next one. To avoid this possibility, it is a good idea to issue a `clear` command at the start of each new independent calculation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
>> clear
```

The command `clear` or `clear all` removes all variables from the workspace. This frees up system memory. In order to display a list of the variables currently in the memory, type

```
>> who
```

while, `whos` will give more details which include size, space allocation, and class of the variables.

1.4.8 Keeping track of your work session

It is possible to keep track of everything done during a MATLAB session with the `diary` command.

```
>> diary
```

or give a name to a created file,

```
>> diary FileName
```

where `FileName` could be any arbitrary name you choose.

The function `diary` is useful if you want to save a complete MATLAB session. They save all input and output as they appear in the MATLAB window. When you want to stop the recording, enter `diary off`. If you want to start recording again, enter `diary on`. The file that is created is a simple text file. It can be opened by an editor or a word processing program and edited to remove extraneous material, or to add your comments. You can use the function type to view the diary file or you can edit in a text editor or print. This command is useful, for example in the process of preparing a homework or lab submission.

1.4.9 Entering multiple statements per line

It is possible to enter multiple statements per line. Use commas (,) or semicolons (;) to enter more than one statement at once. Commas (,) allow multiple statements per line without suppressing output.

```
>> a=7; b=cos(a), c=cosh(a)
b   =
    0.6570
c   =
    548.3170
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.10 Miscellaneous commands

Here are few additional useful commands:

- To clear the Command Window, type `clc`
- To abort a MATLAB computation, type `ctrl-c`
- To continue a line, type ...

1.4.11 Getting help

To view the online documentation, select MATLAB Help from Help menu or MATLAB Help directly in the Command Window. The preferred method is to use the *Help Browser*. The Help Browser can be started by selecting the ? icon from the desktop toolbar. On the other hand, information about any command is available by typing

```
>> help Command
```

Another way to get help is to use the `lookfor` command. The `lookfor` command differs from the `help` command. The help command searches for an exact function name match, while the `lookfor` command searches the quick summary information in each function for a match. For example, suppose that we were looking for a function to take *the inverse of a matrix*. Since MATLAB does not have a function named `inverse`, the command `help inverse` will produce nothing. On the other hand, the command `lookfor inverse` will produce detailed information, which includes the function of interest, inv.

```
>> lookfor inverse
```

NOTE - At this particular time of our study, it is important to emphasize one main point. Because MATLAB is a huge program; it is impossible to cover *all the details* of each function one by one. However, we will give you information how to get help. Here are some examples:

- Use on-line `help` to request info on a specific function

  ```
  >> help sqrt
  ```

- In the current version (MATLAB version 7), the `doc` function opens the on-line version of the help manual. This is very helpful for more complex commands.

  ```
  >> doc plot
  ```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Use `lookfor` to find functions by keywords. The general form is

```
>> lookfor FunctionName
```

1.5 Exercises

NOTE: Due to the teaching class during this Fall 2005, the *problems* are *temporarily* removed from this section.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Chapter 2

Tutorial lessons 2

2.1 Mathematical functions

MATLAB offers many predefined mathematical functions for technical computing which contains a large set of mathematical functions.

Typing `help elfun` and `help specfun` calls up full lists of *elementary* and *special* functions respectively.

There is a long list of mathematical functions that are *built* into MATLAB. These functions are called *built-ins*. Many standard mathematical functions, such as $\sin(x)$, $\cos(x)$, $\tan(x)$, e^x, $ln(x)$, are evaluated by the functions `sin`, `cos`, `tan`, `exp`, and `log` respectively in MATLAB.

Table 2.1 lists some commonly used functions, where variables `x` and `y` can be numbers, vectors, or matrices.

Table 2.1: Elementary functions

`cos(x)`	Cosine	`abs(x)`	Absolute value
`sin(x)`	Sine	`sign(x)`	Signum function
`tan(x)`	Tangent	`max(x)`	Maximum value
`acos(x)`	Arc cosine	`min(x)`	Minimum value
`asin(x)`	Arc sine	`ceil(x)`	Round towards $+\infty$
`atan(x)`	Arc tangent	`floor(x)`	Round towards $-\infty$
`exp(x)`	Exponential	`round(x)`	Round to nearest integer
`sqrt(x)`	Square root	`rem(x)`	Remainder after division
`log(x)`	Natural logarithm	`angle(x)`	Phase angle
`log10(x)`	Common logarithm	`conj(x)`	Complex conjugate

In addition to the elementary functions, MATLAB includes a number of predefined

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Chapter 3

Array operations and Linear equations

3.1 Array operations

MATLAB has two different types of arithmetic operations: matrix arithmetic operations and array arithmetic operations. We have seen matrix arithmetic operations in the previous lab. Now, we are interested in array operations.

3.1.1 Matrix arithmetic operations

As we mentioned earlier, MATLAB allows arithmetic operations: $+$, $-$, $*$, and $\hat{}$ to be carried out on matrices. Thus,

`A+B` or `B+A`	is valid if `A` and B are of the same size
`A*B`	is valid if `A`'s number of column equals `B`'s number of rows
`A^2`	is valid if `A` is square and equals `A*A`
α`*A` or `A*`α	multiplies each element of `A` by α

3.1.2 Array arithmetic operations

On the other hand, array arithmetic operations or *array operations* for short, are done *element-by-element*. The period character, ., distinguishes the array operations from the matrix operations. However, since the matrix and array operations are the same for addition $(+)$ and subtraction $(-)$, the character pairs $(.+)$ and $(.-)$ are not used. The list of array operators is shown below in Table 3.2. If `A` and `B` are two matrices of the same size with elements $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$, then the command

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.*	Element-by-element multiplication
./	Element-by-element division
.^	Element-by-element exponentiation

Table 3.1: Array operators

```
>> C = A.*B
```

produces another matrix C of the same size with elements $c_{ij} = a_{ij}b_{ij}$. For example, using the same 3×3 matrices,

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix}$$

we have,

```
>> C = A.*B
C   =
    10    40    90
   160   250   360
   490   640   810
```

To raise a scalar to a power, we use for example the command 10^2. If we want the operation to be applied to each element of a matrix, we use .^2. For example, if we want to produce a new matrix whose elements are the square of the elements of the matrix **A**, we enter

```
>> A.^2
ans   =
     1     4     9
    16    25    36
    49    64    81
```

The relations below summarize the above operations. To simplify, let's consider two vectors U and V with elements $U = [u_i]$ and $V = [v_j]$.

$$\begin{array}{lll} U.*V & \text{produces} & [u_1v_1\ u_2v_2 \ldots\ u_nv_n] \\ U./V & \text{produces} & [u_1/v_1\ u_2/v_2 \ldots\ u_n/v_n] \\ U.\hat{}V & \text{produces} & [u_1^{v_1}\ u_2^{v_2} \ldots\ u_n^{v_n}] \end{array}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operation	Matrix	Array
Addition	+	+
Subtraction	−	−
Multiplication	*	.*
Division	/	./
Left division	\	.\
Exponentiation	^	.^

Table 3.2: Summary of matrix and array operations

3.2 Solving linear equations

One of the problems encountered most frequently in scientific computation is the solution of systems of simultaneous linear equations. With matrix notation, a system of simultaneous linear equations is written

$$Ax = b \tag{3.1}$$

where there are as many equations as unknown. A is a given square matrix of order n, b is a given column vector of n components, and x is an unknown column vector of n components.

In linear algebra we learn that the solution to $Ax = b$ can be written as $x = A^{-1}b$, where A^{-1} is the inverse of A.

For example, consider the following system of linear equations

$$\begin{cases} x + 2y + 3z &= 1 \\ 4x + 5y + 6z &= 1 \\ 7x + 8y &= 1 \end{cases}$$

The coefficient matrix A is

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad \text{and the vector} \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

With matrix notation, a system of simultaneous linear equations is written

$$Ax = b \tag{3.2}$$

This equation can be solved for x using linear algebra. The result is $x = A^{-1}b$.

There are typically two ways to solve for x in MATLAB:

1. The first one is to use the matrix inverse, `inv`.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
>> A = [1 2 3; 4 5 6; 7 8 0];
>> b = [1; 1; 1];
>> x = inv(A)*b
x   =
    -1.0000
     1.0000
    -0.0000
```

2. The second one is to use the *backslash* (\)operator. The numerical algorithm behind this operator is computationally efficient. This is a numerically reliable way of solving system of linear equations by using a well-known process of Gaussian elimination.

```
>> A = [1 2 3; 4 5 6; 7 8 0];
>> b = [1; 1; 1];
>> x = A\b
x   =
    -1.0000
     1.0000
    -0.0000
```

This problem is at the heart of many problems in scientific computation. Hence it is important that we know how to solve this type of problem efficiently.

Now, we know how to solve a system of linear equations. In addition to this, we will see some additional details which relate to this particular topic.

3.2.1 Matrix inverse

Let's consider the same matrix A.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$$

Calculating the inverse of A manually is probably not a pleasant work. Here the hand-calculation of A^{-1} gives as a final result:

$$A^{-1} = \frac{1}{9} \begin{bmatrix} -16 & 8 & -1 \\ 14 & -7 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

In MATLAB, however, it becomes as simple as the following commands:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
>> A = [1 2 3; 4 5 6; 7 8 0];
>> inv(A)
ans =
    -1.7778    0.8889   -0.1111
     1.5556   -0.7778    0.2222
    -0.1111    0.2222   -0.1111
```

which is similar to:

$$A^{-1} = \frac{1}{9}\begin{bmatrix} -16 & 8 & -1 \\ 14 & -7 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

and the determinant of A is

```
>> det(A)
ans  =
      27
```

For further details on applied numerical linear algebra, see [10] and [11].

3.2.2 Matrix functions

MATLAB provides many matrix functions for various matrix/vector manipulations; see Table 3.3 for some of these functions. Use the online help of MATLAB to find how to use these functions.

`det`	Determinant
`diag`	Diagonal matrices and diagonals of a matrix
`eig`	Eigenvalues and eigenvectors
`inv`	Matrix inverse
`norm`	Matrix and vector norms
`rank`	Number of linearly independent rows or columns

Table 3.3: Matrix functions

3.3 Exercises

NOTE: Due to the teaching class during this Fall Quarter 2005, the *problems* are *temporarily* removed from this section.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Chapter 4

Introduction to programming in MATLAB

4.1 Introduction

So far in these lab sessions, all the commands were executed in the Command Window. The problem is that the commands entered in the Command Window cannot be saved and executed again for several times. Therefore, a different way of executing repeatedly commands with MATLAB is:

1. to *create* a file with a list of commands,
2. *save* the file, and
3. *run* the file.

If needed, corrections or changes can be made to the commands in the file. The files that are used for this purpose are called script files or *scripts* for short.

This section covers the following topics:

- M-File Scripts
- M-File Functions

4.2 M-File Scripts

A *script file* is an external file that contains a sequence of MATLAB statements. Script files have a filename extension `.m` and are often called M-files. M-files can be *scripts* that simply execute a series of MATLAB statements, or they can be *functions* that can accept arguments and can produce one or more outputs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 Examples

Here are two simple scripts.

Example 1

Consider the system of equations:

$$\begin{cases} x + 2y + 3z &= 1 \\ 3x + 3y + 4z &= 1 \\ 2x + 3y + 3z &= 2 \end{cases}$$

Find the solution x to the system of equations.

SOLUTION:

- Use the MATLAB *editor* to create a file: **File** → **New** → **M-file**.
- Enter the following statements in the file:

```
A = [1 2 3; 3 3 4; 2 3 3];
b = [1; 1; 2];
x = A\b
```

- Save the file, for example, `example1.m`.
- Run the file, in the command line, by typing:

```
>> example1
x   =
    -0.5000
     1.5000
    -0.5000
```

When execution completes, the variables (`A`, `b`, and `x`) remain in the workspace. To see a listing of them, enter `whos` at the command prompt.

NOTE: The MATLAB editor is both a text editor specialized for creating M-files and a graphical MATLAB debugger. The MATLAB editor has numerous menus for tasks such as *saving*, *viewing*, and *debugging*. Because it performs some simple checks and also uses color to differentiate between various elements of codes, this text editor is recommended as the tool of choice for writing and editing M-files.

There is another way to open the editor:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
>> edit
```

or

```
>> edit filename.m
```

to open `filename.m`.

Example 2

Plot the following cosine functions, $y_1 = 2\cos(x)$, $y_2 = \cos(x)$, and $y_3 = 0.5 * \cos(x)$, in the interval $0 \leq x \leq 2\pi$. This example has been presented in previous Chapter. Here we put the commands in a file.

- Create a file, say `example2.m`, which contains the following commands:

```
x = 0:pi/100:2*pi;
y1 = 2*cos(x);
y2 = cos(x);
y3 = 0.5*cos(x);
plot(x,y1,'--',x,y2,'-',x,y3,':')
xlabel('0 \leq x \leq 2\pi')
ylabel('Cosine functions')
legend('2*cos(x)','cos(x)','0.5*cos(x)')
title('Typical example of multiple plots')
axis([0 2*pi -3 3])
```

- Run the file by typing `example2` in the Command Window.

4.2.2 Script side-effects

All variables created in a script file are added to the workspace. This may have undesirable effects, because:

- Variables already existing in the workspace may be overwritten.
- The execution of the script can be affected by the state variables in the workspace.

As a result, because scripts have some undesirable side-effects, it is better to code any complicated applications using rather function M-file.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 M-File functions

As mentioned earlier, functions are programs (or *routines*) that accept *input* arguments and return *output* arguments. Each M-file function (or *function* or *M-file* for short) has its *own* area of workspace, separated from the MATLAB base workspace.

4.3.1 Anatomy of a M-File function

This simple function shows the basic parts of an M-file.

```
function f = factorial(n)                    (1)
% FACTORIAL(N) returns the factorial of N.   (2)
% Compute a factorial value.                 (3)

f = prod(1:n);                               (4)
```

The first line of a function M-file starts with the keyword `function`. It gives the function *name* and order of *arguments*. In the case of function `factorial`, there are up to one output argument and one input argument. Table 4.1 summarizes the M-file function.

As an example, for $n = 5$, the result is,

```
>> f = factorial(5)
f =
   120
```

Table 4.1: Anatomy of a M-File function

Part no.	M-file element	Description
(1)	Function definition line	Define the function name, and the number and order of input and output arguments
(2)	H1 line	A one line summary description of the program, displayed when you request Help
(3)	Help text	A more detailed description of the program
(4)	Function body	Program code that performs the actual computations

Both *functions* and *scripts* can have all of these parts, except for the *function definition line* which applies to *function* only.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

In addition, it is important to note that *function name* must begin with a letter, and must be no longer than than the maximum of 63 characters. Furthermore, the name of the text file that you save will consist of the function name with the extension `.m`. Thus, the above example file would be `factorial.m`.

Table 4.2 summarizes the differences between *scripts* and *functions*.

Table 4.2: Difference between scripts and functions

SCRIPTS	FUNCTIONS
- Do not accept input arguments or return output arguments.	- Can accept input arguments and return output arguments.
- Store variables in a workspace that is shared with other scripts	- Store variables in a workspace internal to the function.
- Are useful for automating a series of commands	- Are useful for extending the MATLAB language for your application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 Input and output arguments

As mentioned above, the input arguments are listed inside parentheses following the function name. The output arguments are listed inside the brackets on the left side. They are used to transfer the output from the function file. The general form looks like this

```
function [outputs] = function_name(inputs)
```

Function file can have none, one, or several output arguments. Table 4.3 illustrates some possible combinations of input and output arguments.

Table 4.3: Example of input and output arguments

`function C=FtoC(F)`	One input argument and one output argument
`function area=TrapArea(a,b,h)`	Three inputs and one output
`function [h,d]=motion(v,angle)`	Two inputs and two outputs

4.4 Input to a script file

When a script file is executed, the variables that are used in the calculations within the file must have assigned values. The assignment of a value to a variable can be done in three ways.

1. The variable is defined in the script file.
2. The variable is defined in the command prompt.
3. The variable is entered when the script is executed.

We have already seen the two first cases. Here, we will focus our attention on the third one. In this case, the variable is defined in the script file. When the file is executed, the user is *prompted* to assign a value to the variable in the command prompt. This is done by using the `input` command. Here is an example.

```
% This script file calculates the average of points
% scored in three games.
% The point from each game are assigned to a variable
% by using the 'input' command.

game1 = input('Enter the points scored in the first game ');
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
game2 = input('Enter the points scored in the second game ');
game3 = input('Enter the points scored in the third game ');
average = (game1+game2+game3)/3
```

The following shows the command prompt when this script file (saved as `example3`) is executed.

```
>> example3
>> Enter the points scored in the first game   15
>> Enter the points scored in the second game  23
>> Enter the points scored in the third game   10

average =
        16
```

The `input` command can also be used to assign *string* to a variable. For more information, see MATLAB documentation.

A typical example of M-file function programming can be found in a recent paper which related to the solution of the ordinary differential equation (ODE) [12].

4.5 Output commands

As discussed before, MATLAB automatically generates a *display* when commands are executed. In addition to this automatic display, MATLAB has several commands that can be used to generate displays or outputs.

Two commands that are frequently used to generate output are: `disp` and `fprintf`. The main differences between these two commands can be summarized as follows (Table 4.4).

Table 4.4: `disp` and `fprintf` commands

`disp`	. Simple to use. . Provide limited control over the appearance of output
`fprintf`	. Slightly more complicated than `disp`. . Provide total control over the appearance of output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้