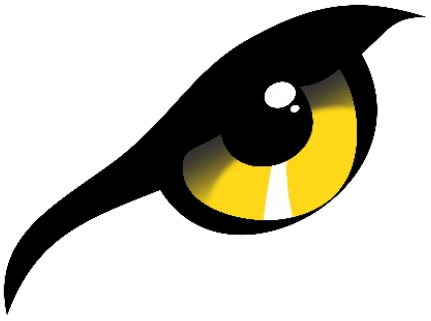
คู่มือการใช้งาน
ชุดพัฒนาซอฟต์แวร์อ่านบัตรประชาชน
ระบบแอนดรอยด์
HawkEye NALib SDK

บริษัท อาร์แอนด์ดี คอมพิวเตอร์ ซิสเท็ม จำกัด



NALib_Manual_Thai_R260520.odt

สารบัญ

สัญญาอนุญาตให้ใช้โปรแกรมคอมพิวเตอร์ i
ส่วนประกอบของชุดพัฒนาซอฟต์แวร์ NALib SDK 1
คุณสมบัติของ HawkEye NALib SDK 1
ขั้นตอนการพัฒนาแอปพลิเคชัน 2
การนำไปใช้งานหลังจากพัฒนาแอปพลิเคชันเสร็จ 3
ผังการทำงานของแอปพลิเคชัน 4
รายละเอียดการใช้งาน NALib API 5
- setListenerNA 5
- openLibNA 6
- closeLibNA 6
- getReaderListNA 6
- selectReaderNA 8
- deselectReaderNA 8
- connectCardNA 9
- disconnectCardNA 9
- getNIDNumberNA 9
- getNIDTextNA 9
- getNIDPhotoNA 12
- getCardStatusNA 12
- getRidNA 12
- getSoftwareInfoNA 13
- getLicenseInfoNA 13
- updateLicenseFileNA 13
- setPermissionsNA 14
- getReaderInfoNA 14
- getEmvTextCA 15
- รหัสส่งกลับและรหัสข้อผิดพลาด (Return Code) 16

การทดสอบการใช้คำสั่งด้วยแอปพลิเคชัน NADemo 17
NASample และตัวอย่างขั้นตอนการใช้คำสั่งในการพัฒนาแอปพลิเคชัน 22
การจัดการแฟ้มใบอนุญาต (DLS) สำหรับเครื่องอ่านแบบ File Licensing 31
การพัฒนาโปรแกรมสำหรับเครื่องอ่านแบบ Virtual Licensing 33
การขอ Permissions 34
ความสามารถในการเชื่อมต่อกับเครื่องอ่านของ NALib 36
หน้าต่าง Pop-up สำหรับการขออนุญาตใช้เครื่องอ่าน USB ภายในและภายนอกแอปพลิเคชัน 37
คำแนะนำการใช้งานเครื่องอ่านแบบต่าง ๆ 39
รายการปรับปรุงของ NALib SDK 45
คำถามที่ถามบ่อยของชุดพัฒนาซอฟต์แวร์อ่านบัตรประชาชนระบบแอนดรอยด์ HawkEye NALib 47

สัญญาอนุญาตให้ใช้โปรแกรมคอมพิวเตอร์

โปรดอ่านข้อความในสัญญาฉบับนี้โดยตลอดก่อนที่ท่านจะเข้าผูกพันตามสัญญาโดยการเปิดใช้งานโปรแกรมคอมพิวเตอร์นี้

หลังจากท่านได้อ่านข้อความในสัญญาฉบับนี้แล้ว หากไม่ตกลงตามสัญญา ขอให้ส่งคืนโปรแกรมคอมพิวเตอร์ที่อยู่ในแผ่นดิสก์หรือสื่อบันทึกชนิดอื่น ๆ รวมทั้งเอกสารและสินค้าอื่น ๆ ที่แนบมาด้วยทั้งหมด (หากมี) คืนไปยังสถานที่ที่ท่านได้รับสิ่งเหล่านี้มาภายใน 7 วัน (เจ็ดวัน) นับแต่วันได้รับมอบโปรแกรมคอมพิวเตอร์ หากท่านได้รับโปรแกรมคอมพิวเตอร์มาด้วยวิธีการดาวน์โหลด หรือ การทำสำเนา ท่านจะต้องทำลายหรือลบโปรแกรมคอมพิวเตอร์ และสำเนาทั้งหมดออกจากฮาร์ดดิสก์, หน่วยบันทึกข้อมูลที่อยู่ในระบบเครือข่าย, หน่วยบันทึกข้อมูลที่อยู่ในระบบเครือข่ายอินเทอร์เน็ต และสื่อเก็บข้อมูลอื่น ๆ โดยทันที

เมื่อท่านได้ติดตั้ง หรือใช้งานโปรแกรมคอมพิวเตอร์นี้ แม้เพียงส่วนใดส่วนหนึ่ง ถือว่าท่านยอมผูกพันในฐานะผู้รับอนุญาตให้ใช้โปรแกรมคอมพิวเตอร์ตามสัญญาฉบับนี้ ซึ่งต่อไปนี้ในสัญญานี้จะเรียกว่า “ผู้รับอนุญาต”

บริษัท อาร์แอนด์ดี คอมพิวเตอร์ ซิสเท็ม จำกัด ตั้งอยู่ที่ 48/20 ซอยรัชดาภิเษก 20 ถนนรัชดาภิเษก แขวงสามเสนนอก เขตห้วยขวาง กรุงเทพมหานคร ซึ่งต่อไปนี้ในสัญญานี้จะเรียกว่า “ผู้อนุญาต”

ผู้อนุญาตและผู้รับอนุญาตตกลงทำสัญญากันมีข้อความดังต่อไปนี้

1. การอนุญาตให้ใช้โปรแกรมคอมพิวเตอร์

1.1 ผู้อนุญาตเป็นเจ้าของลิขสิทธิ์ โปรแกรมคอมพิวเตอร์ ชื่อ Thai ID Card Development Kit for Android (NALib SDK) ซึ่งมีส่วนประกอบย่อย คือ HawkEye TDA, NALib, NADemo, NASample, TDAService SDK, TDAService, TDASample, TDADemo และโปรแกรมคอมพิวเตอร์อื่น ๆ ที่แนบมาพร้อมกัน และเอกสารต่าง ๆ ที่ระบุในสัญญานี้ (ซึ่งในสัญญานี้เรียกว่า “โปรแกรมคอมพิวเตอร์”)

1.2 ผู้อนุญาตตกลงอนุญาตให้ผู้รับอนุญาตใช้โปรแกรมคอมพิวเตอร์ตามที่ระบุในสัญญานี้ในลักษณะดังต่อไปนี้

- ผู้รับอนุญาต สามารถใช้งานโปรแกรมคอมพิวเตอร์นี้เพื่อทดสอบ และเพื่อพัฒนาโปรแกรมใช้งาน (Application Program) ด้วยตนเอง หรือหน่วยงานภายในนิติบุคคล (หากผู้รับอนุญาตเป็นนิติบุคคล) เท่านั้น ไม่สามารถนำไปใช้งานกับบุคคลอื่นหรือนิติบุคคลอื่นได้
- โปรแกรมใช้งาน ที่พัฒนาขึ้นมา หากมีการนำส่วนใดส่วนหนึ่งของโปรแกรมคอมพิวเตอร์นี้ไปควบรวม ผสาน เชื่อมโยง หรือ เรียกใช้งาน ให้ถือว่าโปรแกรมใช้งานนั้นเป็นส่วนหนึ่งของโปรแกรมคอมพิวเตอร์นี้ และผู้รับอนุญาตจะต้องปฏิบัติต่อโปรแกรมใช้งานนั้นเช่นเดียวกับโปรแกรมคอมพิวเตอร์นี้
- ผู้รับอนุญาต จะใช้งานโปรแกรมคอมพิวเตอร์นี้ร่วมกับเครื่องอ่านบัตรสมาร์ทการ์ดที่กำหนดโดยผู้อนุญาตเท่านั้น
- ผู้ใช้โปรแกรมคอมพิวเตอร์ในขณะเดียวกัน จะมีได้ไม่เกิน 1 คน (หนึ่งคน) ต่อ 1 สิทธิ์ (หนึ่งสิทธิ์)
- สัญญานี้ไม่ให้สิทธิ์ผู้รับอนุญาตในการได้รับรหัสต้นฉบับ (Source Code) ของโปรแกรมคอมพิวเตอร์ ยกเว้นส่วนของโปรแกรมตัวอย่าง

2. ระยะเวลาการอนุญาต

2.1 ผู้รับอนุญาต จะใช้งานโปรแกรมคอมพิวเตอร์นี้ได้ตั้งแต่วันที่เริ่มผูกพันในฐานะผู้รับอนุญาตจนกว่าจะบอกเลิกสัญญาหรือจนกว่าผู้อนุญาตจะยกเลิกการให้อนุญาต

3. สิทธิ์และหน้าที่ของผู้รับอนุญาต

3.1 ผู้รับอนุญาตมีสิทธิ์ที่จะใช้โปรแกรมคอมพิวเตอร์และเอกสารต่าง ๆ ตามเงื่อนไขที่ระบุในสัญญานี้

3.2 ผู้รับอนุญาตไม่มีสิทธิ์ที่จะให้ ให้เช่า ให้ยืม หรือขายต่อโปรแกรมคอมพิวเตอร์ หรือเอกสาร หรือสิทธิ์ใด ๆ ที่ให้ไว้ตามสัญญานี้แก่บุคคลอื่น

3.3 ในกรณีผู้รับอนุญาตได้มีการพัฒนาโปรแกรมใช้งานขึ้นมา ผู้รับอนุญาตสามารถนำแฟ้มโปรแกรมคอมพิวเตอร์เฉพาะส่วนที่เป็นแฟ้มแบบ DLS, แฟ้มไลบรารี JAR, AAR และโปรแกรมบริการ (TDAService) ไปเผยแพร่ต่อแก่บุคคลอื่นได้ในลักษณะของการอนุญาตให้ใช้งานเท่านั้น และผู้รับอนุญาตจะต้องส่งมอบสัญญาฉบับนี้ในรูปแบบของเอกสารกระดาษหรือแฟ้มคอมพิวเตอร์แนบไปพร้อมกับแฟ้มแบบ DLS, แฟ้มไลบรารีและโปรแกรมบริการ ที่ส่งมอบแก่บุคคลอื่นด้วย และบุคคลอื่นที่ได้รับแฟ้มแบบ DLS, แฟ้มไลบรารีและโปรแกรมบริการจะต้องยอมผูกพันในฐานะผู้รับอนุญาตให้ใช้โปรแกรมคอมพิวเตอร์ตามสัญญาฉบับนี้ด้วย ผู้รับอนุญาตจะต้องไม่มอบแฟ้มโปรแกรมคอมพิวเตอร์อื่นของโปรแกรมคอมพิวเตอร์นี้ ให้กับบุคคลอื่น

3.4 ผู้รับอนุญาตมีสิทธิ์ทำสำเนาโปรแกรมคอมพิวเตอร์ได้ 1 ชุด (หนึ่งชุด) เพื่อป้องกันการสูญหายในระหว่างการใช้งานโปรแกรมคอมพิวเตอร์ดังกล่าวเท่านั้น

3.5 ผู้รับอนุญาตขอรับรองว่าจะไม่เปลี่ยนแปลง ดัดแปลง ลบ ทำลาย ทำให้เสียหาย หรือทำให้ไม่ชัดเจน ซึ่ง ชื่อของโปรแกรมคอมพิวเตอร์ ชื่อของเจ้าของลิขสิทธิ์ เครื่องหมายหรือสัญลักษณ์แสดงความเป็นเจ้าของลิขสิทธิ์หรือเครื่องหมายการค้าของเจ้าของลิขสิทธิ์ ไม่ว่าจะโดยจงใจหรือประมาทเลินเล่อก็ตาม

3.6 ผู้รับอนุญาตจะดูแลรักษาโปรแกรมคอมพิวเตอร์ เอกสารต่าง ๆ และสำเนาทั้งหลาย เป็นอย่างดีที่สุด จะไม่ยอมให้ผู้อื่นเข้าถึง นำไปใช้ นำไปศึกษา ทำสำเนา หรือกระทำประการหนึ่งประการใดต่อโปรแกรมคอมพิวเตอร์ เอกสารต่าง ๆ และสำเนาทั้งหลาย

4. การไม่ทำวิศวกรรมย้อนกลับ และอื่น ๆ

ผู้รับอนุญาตตกลงว่า ผู้รับอนุญาตจะไม่กระทำการต่อไปนี้แก่โปรแกรมคอมพิวเตอร์ และส่วนประกอบที่ใช้ในการทำงานร่วมกัน ไม่ว่าส่วนประกอบนั้นจะเป็นโปรแกรมคอมพิวเตอร์, เฟิร์มแวร์, โปรโตคอล, รูปแบบการติดต่อกับผู้ใช้ (User Interface), รูปแบบการเชื่อมต่อกับโปรแกรมใช้งาน (Application Programming Interface) ฮาร์ดแวร์ และส่วนประกอบอื่น และจะไม่พยายามกระทำการด้วยตนเองหรืออนุญาต เพิกเฉย มอบหมายหรือสนับสนุนให้บุคคลอื่นกระทำการดังต่อไปนี้ แก้ไข ปรับปรุง เปลี่ยนแปลง ดัดแปลง แก้ไขงานต้นฉบับ ขีดฆ่า ทำให้เสียหาย ทำวิศวกรรมย้อนกลับ (Reverse Engineer) แปล (Translate) แปลย้อนกลับ (Decompile) แปลย้อนกลับภาษาเครื่อง (Disassembly) วิเคราะห์การทำงานภายใน (System Analyze) ทดสอบผ่านกล่องดำ (Black Box Test) ถอดรหัส (Decrypt) ไล่การทำงานผ่านเครื่องมือค้นหาจุดบกพร่อง (Debug) ทั้งในระดับรหัสไบต์ (Bytecode Debugging) และระดับไบนารี (Binary Code Debugging) รื้อและสร้างใหม่ (Deconstruct) ดักจับข้อมูล (Sniff) จำลองการทำงาน (Simulate) เลียนแบบการทำงาน (Emulate) และการกระทำอื่นใดก็ตามที่ให้ผลเช่นเดียวกันกับการกระทำข้างต้น ไม่ว่าจะมีชื่อเรียกว่าอย่างไรก็ตาม

5. การส่งคืนโปรแกรมคอมพิวเตอร์

ในกรณีที่มีการเลิกสัญญา ผู้รับอนุญาตจะต้องปฏิบัติดังนี้

- ส่งคืนโปรแกรมคอมพิวเตอร์ เอกสาร และสำเนาทั้งหมดที่ได้ทำขึ้นตามสัญญานี้ คืนให้กับผู้อนุญาตโดยทันที
- ทำลายหรือลบโปรแกรมคอมพิวเตอร์ และสำเนาทั้งหมดออกจากฮาร์ดดิสก์, หน่วยบันทึกข้อมูลที่อยู่ในระบบเครือข่าย, หน่วยบันทึกข้อมูลที่อยู่ในระบบเครือข่ายอินเทอร์เน็ต และสื่อเก็บข้อมูลอื่น ๆ โดยทันที
- เรียกคืนและทำลายโปรแกรมคอมพิวเตอร์ และสำเนาทั้งหมดจากบุคคลอื่นที่ผู้รับอนุญาตได้ส่งมอบไป

6. การรักษาความลับทางการค้า

ผู้รับอนุญาตทราบว่า บรรดาข้อมูล เทคนิค วิธีการ กระบวนการ ที่เกี่ยวข้องกับโปรแกรมคอมพิวเตอร์นี้เป็นความลับทางการค้าของผู้อนุญาตและเจ้าของลิขสิทธิ์

ผู้รับอนุญาตจะไม่เปิดเผยข้อมูลความลับทางการค้าทั้งหลายเหล่านี้ให้บุคคลอื่นทราบ และจะพยายามจนถึงที่สุดในการรักษาข้อมูลความลับทางการค้าทั้งหลายเหล่านี้ไม่ให้ถูกเปิดเผย

7. ข้อกำหนดการใช้งาน

- ห้ามนำไปใช้ในทางที่ผิดกฎหมาย
- ใช้เพื่อการทำงาน, ให้บริการหรือทำธุรกรรมเพื่อเจ้าของบัตรประจำตัวประชาชนหรือบัตรธนาคารเท่านั้น
- ในการนำไปอ่านข้อมูลจากบัตรประจำตัวประชาชนหรือบัตรธนาคาร จะต้องได้รับความยินยอมจากเจ้าของบัตรเท่านั้น
- ห้ามนำข้อมูลภาพและข้อความที่อ่านได้จากบัตรประจำตัวประชาชนหรือบัตรธนาคารไปเปิดเผยหรือเผยแพร่ให้ผู้อื่นโดยไม่ได้รับอนุญาตจากเจ้าของบัตร
- ต้องปฏิบัติตามพระราชบัญญัติคุ้มครองข้อมูลส่วนบุคคลและกฎหมายที่เกี่ยวข้องโดยเคร่งครัด
- กรณีมีความเสียหายใด ๆ เกิดขึ้น อันเนื่องมาจากการนำโปรแกรมคอมพิวเตอร์นี้ หรือข้อมูลที่ได้จากโปรแกรมคอมพิวเตอร์ไปใช้งาน ผู้รับอนุญาตต้องรับผิดชอบในผลของความเสียหายที่เกิดขึ้นทั้งทางแพ่งและอาญาและอื่น ๆ โดยทางบริษัทอาร์แอนด์ดี คอมพิวเตอร์ ซิสเท็ม จำกัด ไม่ต้องร่วมรับผิดชอบด้วย

คู่มือการใช้งานชุดพัฒนาซอฟต์แวร์อ่านบัตรประชาชนระบบแอนดรอยด์ HawkEye NALib SDK

บริษัท อาร์แอนด์ดี คอมพิวเตอร์ ซิสเท็ม จำกัด

National ID Card Software Development Kit for Android (NALib SDK) คือ ชุดพัฒนาซอฟต์แวร์การติดต่อกับเครื่องอ่านบัตรสมาร์ทคาร์ด เพื่อใช้ในการพัฒนาแอปพลิเคชันสำหรับอุปกรณ์เคลื่อนที่ระบบแอนดรอยด์ให้สามารถอ่านข้อมูลจากบัตรประจำตัวประชาชนไทยได้

1. ส่วนประกอบของชุดพัฒนาซอฟต์แวร์ NALib SDK

1.1 เครื่องอ่านบัตรสมาร์ทคาร์ด 1 เครื่อง (แบบบลูทูธ หรือแบบ USB แล้วแต่รุ่น)

1.2 ซอฟต์แวร์สำหรับการพัฒนาในชื่อ HawkEye NALib SDK หรือเรียกสั้น ๆ ว่า NALib SDK (ดาวน์โหลดได้จากอินเทอร์เน็ต) ซึ่งภายในประกอบด้วย

- ตัวอย่างแอปพลิเคชัน NASample (เป็นไฟล์ APK) พร้อม Source Code สำหรับใช้ศึกษาเรียนรู้วิธีการเขียนโปรแกรมแบบง่าย ๆ
- ตัวอย่างแอปพลิเคชัน NADemo (เป็นไฟล์ APK) พร้อม Source Code สำหรับใช้ศึกษาเรียนรู้คำสั่งต่าง ๆ และ API ของ NALib SDK และเป็นตัวอย่างในการศึกษาวิธีการเขียนโปรแกรมที่มีความซับซ้อนสูง
- แฟ้ม NALib.aar สำหรับนำไปรวมกับโปรแกรมภาษา Java และอื่น ๆ เพื่อใช้ในการสร้างแอปพลิเคชัน
- คู่มือการใช้งาน

2. คุณสมบัติของ HawkEye NALib SDK

2.1 ใช้งานกับเครื่องอ่านบัตรประชาชน ที่มีคุณสมบัติรองรับ NALib ได้แก่

- เครื่องอ่านแบบ USB เช่นรุ่น TDA3310M2, TDA39VC
- เครื่องอ่านแบบบลูทูธ 3.0 เช่นรุ่น TDA301BT
- เครื่องอ่านแบบบลูทูธ 4.0 แบบพลังงานต่ำ (BLE) ในกลุ่ม BLE0 เช่นรุ่น TDA301BL, TDA301BLM, TDAi301BL และ TDA301BLM
- เครื่องอ่านแบบบลูทูธ 4.0 แบบพลังงานต่ำ (BLE) ในกลุ่ม BLE1 เช่นรุ่น TDA3901BE

2.2 ใช้งานกับใบอนุญาตของเครื่องอ่านได้ 2 แบบ คือ

- เครื่องอ่านแบบที่ใช้แฟ้มใบอนุญาต (File Licensing-FL) เช่น TDA3310C2, TDA3310M2, TDA301BT และ TDA3901BE เป็นต้น
- เครื่องอ่านแบบที่ฝังใบอนุญาตอยู่ในตัวเครื่อง (Virtual Licensing-VL) ซึ่งไม่จำเป็นต้องใช้แฟ้มใบอนุญาต เช่นเครื่องอ่านรุ่น TDA301VC, TDAi301VC, TDA39VC, TDA39VM, TDA301BL, TDA301BLM

2.3 ใช้กับโปรแกรมภาษา Java ภายใต้ Android Studio

2.4 มีชุดพัฒนาเสริม รองรับ Kotlin, React Native และ Flutter

2.5 ใช้งานกับระบบปฏิบัติการ Android 7.0 ขึ้นไป

2.6 สามารถอ่านบัตรประจำตัวประชาชนแบบสมาร์ทคาร์ดของกรมการปกครองได้เป็นอย่างดี

2.7 สามารถอ่านข้อมูลจากบัตรธนาคาร เช่นบัตรเครดิตได้ (เฉพาะเครื่องอ่านรุ่นที่รองรับเท่านั้น)

3. ขั้นตอนการพัฒนาแอปพลิเคชัน

3.1 ซื้อชุดพัฒนาเครื่องอ่านบัตรประชาชนสำหรับโทรศัพท์มือถือและแท็บเล็ตระบบแอนดรอยด์ HawkEye NALib แนะนำว่าควรเป็นเครื่องอ่านแบบ Bluetooth 3.0 หรือ BLE เพื่อจะได้สามารถพัฒนาให้ใช้ได้กับเครื่องอ่านทั้งแบบบลูทูธ และแบบ USB

- เครื่องอ่านแบบบลูทูธ โดยปกติจะใช้งานแบบไร้สายผ่านสัญญาณบลูทูธ แต่ผู้พัฒนาสามารถซื้อสาย OTG มาต่อพ่วงกับสาย USB ที่มีอยู่แล้วในกล่องสินค้า แล้วนำมาเสียบต่อกับอุปกรณ์แอนดรอยด์เพื่อใช้พัฒนาและทดสอบ NALib SDK ในรูปแบบของเครื่องอ่าน USB ได้

3.2 ดาวน์โหลดชุดซอฟต์แวร์ NALib SDK จากเว็บไซต์ของบริษัท ฯ ในหน้าสินค้า HawkEye TDA

3.3 ติดตั้งแอปพลิเคชัน NADemo จากแฟ้ม NADemo.apk ลงในเครื่องแอนดรอยด์ แล้วลองเรียกใช้งานคำสั่งต่าง ๆ ของ NALib API จากแอปพลิเคชัน NADemo

3.4 ศึกษา Source Code ตัวอย่างของ NASample และอาจดูตัวอย่างเพิ่มเติมใน Source Code ของ NADemo ด้วยก็ได้

3.5 สร้างโปรเจ็กต์ใหม่ภายใต้ Android Studio

3.6 ให้นำไลบรารี NALib.aar ไปรวมไว้ในโปรเจ็กต์ (Import Library)

3.7 เริ่มพัฒนาแอปพลิเคชันเป้าหมาย และเรียกใช้งานคำสั่งต่าง ๆ ของ NALib.aar ตามต้องการ

3.8 ทดสอบแอปพลิเคชันเป้าหมายที่พัฒนาได้

3.9 สร้างไฟล์ APK เพื่อเผยแพร่ หรือนำแอปพลิเคชันอัปโหลดขึ้นไปยัง Google Play

หมายเหตุ

ผู้พัฒนาควรมีความรู้ในการพัฒนาแอปพลิเคชันสำหรับระบบแอนดรอยด์มาแล้วเป็นอย่างดี ก่อนที่จะเริ่มใช้ชุดพัฒนา HawkEye NALib ได้

4. การนำไปใช้งานหลังจากพัฒนาแอปพลิเคชันเสร็จ

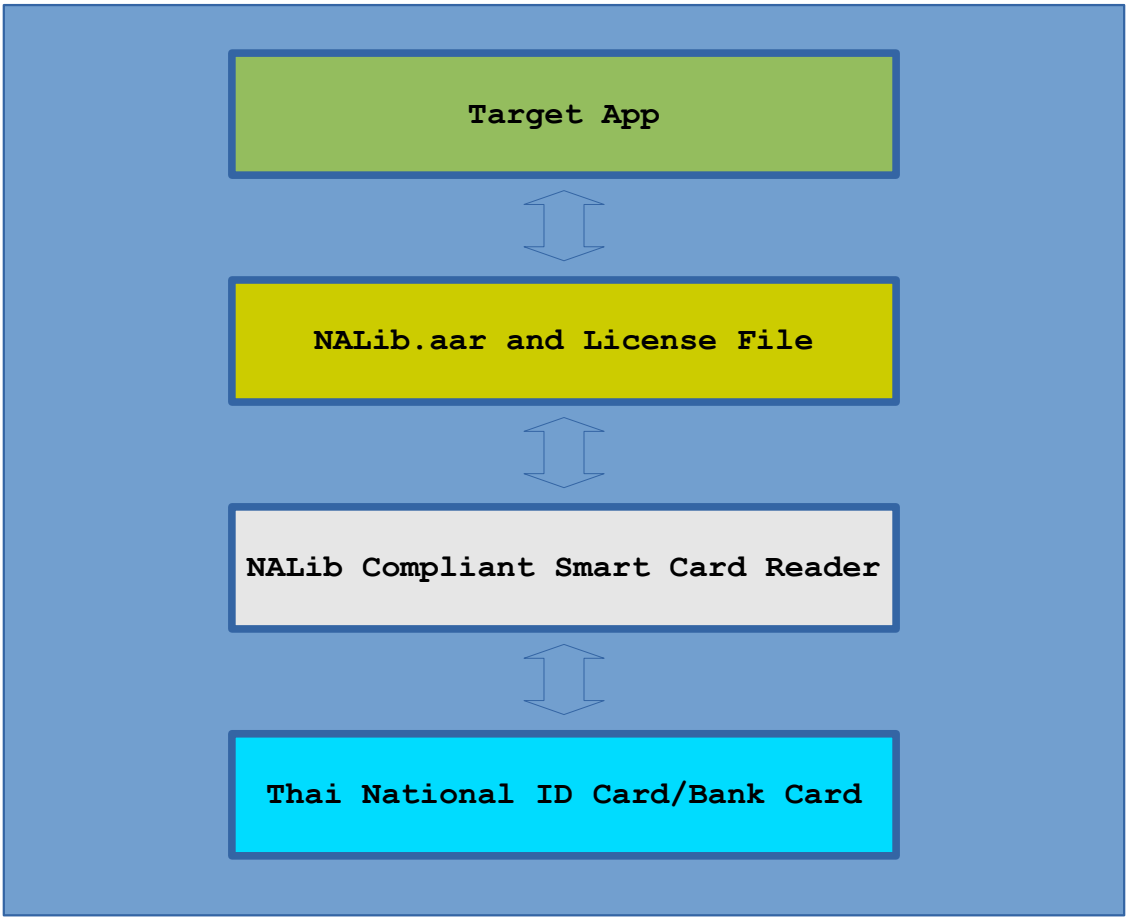
4.1 ให้ผู้ใช้ (Users) จัดซื้อ-จัดหาชุดเครื่องอ่านบัตรประชาชนสำหรับโทรศัพท์มือถือและแท็บเล็ตระบบแอนดรอยด์ HawkEye TDA หรือ HawkEye TDAi ตามรุ่นที่ต้องการ เช่นเลือกรุ่น TDA301BL หากต้องการเครื่องอ่านแบบบลูทูธ หรือรุ่น TDA39VC, TDA3310C2 หากต้องการเครื่องอ่านแบบ USB

4.2 หากเลือกเครื่องอ่านแบบบลูทูธ 3 (TDA301BT, TDAi301BT) ขอให้ผู้ใช้สั่งจับคู่เครื่องอ่านบลูทูธกับอุปกรณ์มือถือก่อนใช้งาน โดยเข้าที่เมนู ติดตั้ง ของระบบแอนดรอยด์ เลือกที่ **Bluetooth** แล้วสั่งจับคู่ (**Pair**) (แต่สำหรับเครื่องอ่านแบบ BLE ที่เป็นบลูทูธ 4 ไม่ต้องสั่งจับคู่ สามารถใช้งานได้ทันที)

4.3 ให้ผู้ใช้ติดตั้งแอปพลิเคชันเป้าหมาย (Target App) จาก Google Play หรือจากไฟล์ APK ของผู้พัฒนา

4.4 ให้ผู้ใช้เปิดใช้งานแอปพลิเคชันเป้าหมาย

5. ผังการทำงานของแอปพลิเคชัน



Target App
คือแอปพลิเคชันที่จะพัฒนาให้สามารถอ่านบัตรประจำตัวประชาชนได้

NALib.aar and License File
คือชุดคำสั่ง NALib API ที่มีไว้ให้เรียกใช้งาน โดยจะให้มาในรูปของไฟล์ NALib.aar ผู้พัฒนาแอปพลิเคชัน (Target App) จะต้องใช้คำสั่งต่าง ๆ เช่นการอ่านข้อมูลจากเครื่องอ่านผ่านไฟล์ AAR นี้
สำหรับไฟล์ไลเซนส์ หรือแฟ้มใบอนุญาต มีไว้เพื่อใช้ในการตรวจสอบเครื่องอ่านที่ใช้ว่าเป็นเครื่องอ่านที่ผ่านการลงทะเบียนและมีสิทธิ์ในการใช้งานหรือไม่
หมายเหตุ กรณีเครื่องอ่านเป็นแบบ Virtual Licensing จะไม่มีการใช้แฟ้มใบอนุญาต จึงไม่จำเป็นต้องมีไฟล์นี้

NALib Compliant Smart Card Reader
คือเครื่องอ่านบัตรสมาร์ทการ์ดที่มาพร้อมกับชุดสินค้า ซึ่งต้องเป็นเครื่องอ่านที่ผ่านการลงทะเบียนไว้แล้วเท่านั้น จึงจะใช้งานได้ ส่วนเครื่องอ่านเปล่าที่ยังไม่ได้ลงทะเบียน จะไม่สามารถใช้งานได้

Thai National ID Card/Bank Card
คือบัตรประจำตัวประชาชนไทยแบบสมาร์ทการ์ด หรือบัตรธนาคาร เช่นบัตรเครดิต และบัตรเดบิต

6. รายละเอียดการใช้งาน NALib API

คำสั่งต่าง ๆ ที่เรียกใช้งานผ่าน API ของ NALib จะมีดังต่อไปนี้

คำสั่ง	หน้าที่
setListenerNA	ผูก ResponseListener สำหรับรอรับผลที่มีการส่งกลับแบบ Callback
openLibNA	เปิดใช้งานไลบรารี
closeLibNA	ปิดการใช้งานไลบรารี
getReaderListNA	ขอรายชื่อเครื่องอ่านบัตรสมาร์ทการ์ดที่ต่ออยู่
selectReaderNA	เลือกหรือเปลี่ยนเครื่องอ่านบัตรสมาร์ทการ์ด
deselectReaderNA	ยกเลิกการเลือกเครื่องอ่านบัตรสมาร์ทการ์ด
connectCardNA	เชื่อมต่อกับบัตรประจำตัวประชาชนหรือบัตรธนาคาร
disconnectCardNA	ยกเลิกการเชื่อมต่อกับบัตรประจำตัวประชาชนหรือบัตรธนาคาร
getNIDNumberNA	อ่านเลขประจำตัวประชาชนจากบัตร
getNIDTextNA	อ่านข้อมูลตัวอักษรทั้งหมดของบัตรประจำตัวประชาชน
getNIDPhotoNA	อ่านรูปภาพถ่ายใบหน้าจากบัตรประจำตัวประชาชน
getCardStatusNA	ขอสถานะการเสียบบัตรที่เครื่องอ่าน
getRidNA	ขอข้อมูลเลขประจำตัวของเครื่องอ่านบัตร (RID)
getSoftwareInfoNA	ขอข้อมูลของไลบารี
getLicenseInfoNA	ขอข้อมูลของใบอนุญาต
updateLicenseFileNA	ปรับเปลี่ยนแฟ้มใบอนุญาต (License File) ให้เป็นปัจจุบัน
setPermissionsNA	ตั้งค่าการอนุญาตที่ใช้กับไลบารี
getReaderInfoNA	ขอข้อมูลของเครื่องอ่านที่กำลังเชื่อมต่ออยู่
getEmvTextCA	อ่านข้อมูลของบัตรธนาคาร (เฉพาะเครื่องอ่านรุ่นที่รองรับ)

6.1 ฟังก์ชัน `void setListenerNA(ResponseListener listener)`

Description	เป็นคำสั่งสำหรับการผูกอินเตอร์เฟซ **`ResponseListener`** เพื่อใช้ในการรอรับผลของฟังก์ชันต่าง ๆ ของ NALib ที่มีการส่งกลับมาแบบ Callback
Input parameter	**`listener`** คือ **`ResponseListener`** ที่สร้างไว้เพื่อรอรับผลลัพธ์จากฟังก์ชันต่าง ๆ
Note	ต้องเรียกฟังก์ชันนี้ก่อนการใช้งานฟังก์ชันอื่น ๆ ทั้งหมดของ NALib และให้เรียกใช้งานครั้งแรกเพียงครั้งเดียวพอ

6.2 ฟังก์ชัน `int openLibNA(String licenseFile)`

Description	เปิดใช้งานไลบรารี
Input parameter	**`licenseFile`** ชื่อแฟ้มและ Path ของใบอนุญาต (License File) ควรใช้ชื่อแฟ้มเป็น rdnidlib.dls เสมอเพื่อป้องกันการสับสน (FL+VL Mode) กรณีจะใช้เครื่องอ่านแบบ Virtual Licensing ที่ไม่มีการใช้แฟ้มใบอนุญาตเพียงอย่างเดียว (VL Mode) ให้ส่งค่า **`licenseFile`** เป็น String ที่ความยาวเป็นศูนย์ (**`""`**)
Return value	0: ฟังก์ชันทำงานปกติ, -1: ฟังก์ชันทำงานผิดพลาด
Callback Listener	**`onOpenLibNA(int result)`**
Output to Listener	**`result`** คือ Return Code ของ **`openLibNA`** มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

6.3 ฟังก์ชัน `int closeLibNA()`

Description	ปิดการใช้งานไลบรารี ให้เรียกฟังก์ชันนี้เมื่อจะหยุดการใช้งาน NALib
Input Parameter	ไม่มี
Return value	เป็น Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

6.4 ฟังก์ชัน `int getReaderListNA(int listOption)`

Description	ขอรายชื่อเครื่องอ่านบัตรสมาร์ทคาร์ดทั้งหมด ทั้งแบบ USB ที่เสียบอยู่ เครื่องอ่านแบบบลูทูธ 3 ที่จับคู่ (Paired) กับอุปกรณ์แอนดรอยด์ไว้แล้ว และเครื่องอ่านบลูทูธแบบ BLE
Input parameter	**`listOption`** กำหนดตัวเลือกของการทำรายชื่อเครื่องอ่าน แบ่งการกำหนดตามแต่ละบิตของตัวแปร **`listOption`** บิต D0 ถึง บิต D7 ดังนี้

D7	D6	D5	D4	D3	D2	D1	D0
NA_POPUP	NA_FIRST	NA_RSVD5	NA_SCAN	NA_BLE1	NA_BLE0	NA_BT	NA_USB
0x80	0x40	ไม่ใช้งาน	0x10	0x08	0x04	0x02	0x01
128	64	ไม่ใช้งาน	16	8	4	2	1

NA_USB (บิต D0, ค่า 0x01) การรวมเครื่องอ่านแบบ USB ไว้ในรายชื่อ

0 = ไม่ต้องรวมเครื่องอ่านแบบ USB ไว้ในรายชื่อ

1 = รวมเครื่องอ่านแบบ USB ไว้ในรายชื่อด้วย

NA_BT (บิต D1, ค่า 0x02) การรวมเครื่องอ่านแบบบลูทูธธรรมดา (Bluetooth 3) (TDA301BT, TDAi301BT) ไว้ในรายชื่อ

0 = ไม่ต้องรวมเครื่องอ่านแบบบลูทูธธรรมดา (TDA301BT, TDAi301BT) ไว้ในรายชื่อ

1 = รวมเครื่องอ่านแบบบลูทูธธรรมดา ไว้ในรายชื่อด้วย

NA_BLE0 (บิต D2, ค่า 0x04) การรวมเครื่องอ่านแบบ BLE ในกลุ่ม BLE0 ได้แก่เครื่องอ่านรุ่น TDA301BL, TDAi301BL, TDA301BLM และ TDAi301BLM ไว้ในรายชื่อ

0 = ไม่ต้องรวมเครื่องอ่านกลุ่ม BLE0 ไว้ในรายชื่อ

1 = รวมเครื่องอ่านกลุ่ม BLE0 ไว้ในรายชื่อด้วย

NA_BLE1 (บิต D3, ค่า 0x08) การรวมเครื่องอ่านแบบ BLE ในกลุ่ม BLE1 ได้แก่เครื่องอ่านรุ่น TDA3901BE ไว้ในรายชื่อ

0 = ไม่ต้องรวมเครื่องอ่านกลุ่ม BLE1 ไว้ในรายชื่อ

1 = รวมเครื่องอ่านกลุ่ม BLE1 ไว้ในรายชื่อด้วย

NA_SCAN (บิต D4, ค่า 0x10) การค้นหาเครื่องอ่านแบบบลูทูธและ BLE ด้วยคลื่นสัญญาณบลูทูธ

0 = ไม่ต้องค้นหา ให้ใช้รายชื่อจาก Paired List ของระบบแอนดรอยด์ (ตัวเลือกนี้ใช้ไม่ได้กับเครื่องอ่านแบบ BLE)

1 = ให้ค้นหาเครื่องอ่านบลูทูธ 3 และ BLE ที่กำลังเปิดใช้งานอยู่จริง

NA_RSVD5 (บิต D5, ค่า 0x20) สงวนไว้ใช้ในอนาคต ให้ใช้ค่าเป็น 0 เสมอ

NA_FIRST (บิต D6, ค่า 0x40) คืนชื่อเครื่องอ่านเครื่องแรกที่พบทันที

0 = ค้นหาเครื่องอ่านทั้งหมดจนกว่าจะครบกำหนดเวลาค้นหา

1 = เมื่อพบเครื่องอ่านเครื่องแรก ให้หยุดการค้นหาแล้วกลับทันที (ไม่เหมาะสำหรับเครื่องอ่านแบบ BLE)

NA_POPUP (บิต D7 ค่า 0x80) การแสดงหน้าต่างโต้ตอบขณะค้นหา

0 = ไม่ต้องแสดงหน้าต่างโต้ตอบขณะค้นหา

1 = ให้แสดงหน้าต่างโต้ตอบขณะค้นหา

Return value: 0: ฟังก์ชันทำงานปกติ, -1: ฟังก์ชันทำงานผิดพลาด

Callback Listener: **`onGetReaderListNA(ArrayList<String> readerList, int result)`**

Output to Listener: **`readerList`** รายชื่อเครื่องอ่านบัตรสมาร์ดคาร์ดแต่ละเครื่องที่พบในระบบจะคืนค่าไว้ใน Array นี้ โดยเครื่องอ่านที่พบเครื่องแรกจะอยู่ใน **`readerList[0]`** หากมีเครื่องที่สองก็จะอยู่ใน **`readerList[1]`** และต่อ ๆ ไปจนครบทุกเครื่องอ่าน ชื่อของเครื่องอ่านเป็นข้อมูลชนิด String

`result` กรณีเป็นเลขมากกว่า 0 จะหมายถึงจำนวนเครื่องอ่านที่พบ กรณีเป็นเลขลบ จะเป็น Return Code ของ **`getReaderListNA`** ซึ่งมีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

Note: หากตั้ง NA_BT หรือ NA_BLE0 หรือ NA_BLE1 เป็น 1 จะต้องมีการอนุญาตให้ใช้ Location Permission ด้วยจึงจะใช้งานได้

ตัวอย่างค่า **`listOption`** ที่แนะนำให้ใช้

0x01 (NA_USB) สำหรับกรณีที่ใช้งานเครื่องอ่านแบบ USB เพียงแบบเดียว (ไม่ใช้เครื่องอ่านแบบบลูทูธ)

0x03 (NA_USB + NA_BT) เหมาะสำหรับงานทั่วไป จะดึงรายชื่อเครื่องอ่าน USB ที่กำลังเสียบอยู่ มาแสดงร่วมกับรายชื่อเครื่องอ่านบลูทูธ 3 จาก Paired List ของระบบแอนดรอยด์ (จะได้รายชื่อรวดเร็วที่สุด)

0xD3 (NA_USB + NA_BT + NA_SCAN + NA_FIRST + NA_POPUP) จะ Scan หาเครื่องอ่าน USB ที่เสียบอยู่ และเครื่องอ่านแบบบลูทูธ 3 ที่เปิดอยู่ และกลับทันทีเมื่อพบเครื่องอ่านเครื่องแรก (หากเสียบเครื่องอ่าน USB จะเลือกเครื่องอ่าน USB ทันที หากไม่ได้เสียบเครื่องอ่าน USB ก็จะรอหาเครื่องอ่านบลูทูธเครื่องแรก) โดยขณะค้นหาจะมีหน้าต่างโต้ตอบแสดงให้ผู้ใช้เห็น

0x93 (NA_USB + NA_BT + NA_SCAN + NA_POPUP) จะ Scan หาเครื่องอ่าน USB ที่เสียบอยู่ และเครื่องอ่านแบบบลูทูธที่เปิดอยู่ทุกเครื่อง จนกว่าจะหมดเวลาโดยขณะค้นหาจะมีหน้าต่างโต้ตอบแสดงให้ผู้ใช้เห็น ผู้ใช้สามารถเลือกเครื่องอ่านที่ต้องการได้เอง

0x95 (NA_USB + NA_BLE0 + NA_SCAN + NA_POPUP) จะ Scan หาเครื่องอ่าน USB ที่เสียบอยู่ และเครื่องอ่านบลูทูธเฉพาะกลุ่ม BLE0 ที่เปิดอยู่ทุกเครื่อง จนกว่าจะหมดเวลาโดยขณะค้นหาจะมีหน้าต่างโต้ตอบแสดงให้ผู้ใช้เห็น ผู้ใช้สามารถเลือกเครื่องอ่านที่ต้องการได้เอง

0x97 (NA_USB + NA_BT + NA_BLE0+NA_SCAN + NA_POPUP) จะ Scan หาเครื่องอ่าน USB ที่เสียบอยู่ และเครื่องอ่านแบบบลูทูธทั้งแบบปกติ และกลุ่ม BLE0 ที่เปิดอยู่ทุกเครื่อง จนกว่าจะหมดเวลาโดยขณะค้นหาจะมีหน้าต่างโต้ตอบแสดงให้ผู้ใช้เห็น ผู้ใช้สามารถเลือกเครื่องอ่านที่ต้องการได้เอง

0x9F (NA_USB + NA_BT + NA_BLE0+ NA_BLE1+NA_SCAN + NA_POPUP) จะ Scan หาเครื่องอ่านทุกชนิด ได้แก่เครื่องอ่าน USB ที่เสียบอยู่ เครื่องอ่านแบบบลูทูธแบบปกติและแบบ BLE (กลุ่ม BLE0 และกลุ่ม BLE1) ที่เปิดอยู่ทุกเครื่อง จนกว่าจะหมดเวลาโดยขณะค้นหาจะมีหน้าต่างโต้ตอบแสดงให้ผู้ใช้เห็น ผู้ใช้สามารถเลือกเครื่องอ่านที่ต้องการได้เอง

6.5 ฟังก์ชัน `int selectReaderNA(String reader)`

Description	เลือกหรือเปลี่ยนเครื่องอ่านบัตรสมาร์ทคาร์ดที่จะใช้ในการทำงานต่อไป
Input parameter	**`reader`** ชื่อของเครื่องอ่าน
Return value	0: ฟังก์ชันทำงานปกติ, -1: ฟังก์ชันทำงานผิดพลาด
Callback Listener	**`onSelectReaderNA(int result)`**
Output to Listener	**`result`** คือ Return Code ของ **`selectReaderNA`** มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด
Note	สำหรับเครื่องอ่านแบบบลูทูธ 3 และ BLE0 เมื่อถูกเลือกด้วยฟังก์ชันนี้แล้ว ต่อมามีการใช้ฟังก์ชันนี้สั่งเปลี่ยนไปเลือกเครื่องอ่านอื่น เครื่องอ่านบลูทูธและ BLE0 ที่ถูกยกเลิกจะปิดตัวเองทันที (Power Off)

6.6 ฟังก์ชัน `int deselectReaderNA()`

Description	ยกเลิกการเลือกเครื่องอ่านบัตรสมาร์ทคาร์ดที่เกิดจากฟังก์ชัน **selectReaderNA**
Input parameter	ไม่มี
Return value	เป็น Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด
Note	สำหรับเครื่องอ่านแบบบลูทูธและแบบ BLE0 เมื่อถูกยกเลิกด้วยฟังก์ชันนี้แล้ว ตัวเครื่องอ่านจะปิดตัวเองเสมอ (Power Off) เนื่องจากคำสั่ง **selectReaderNA** จะมีการสั่งยกเลิกการเลือกเครื่องอ่านอยู่ภายในดังนั้นอาจใช้คำสั่ง **selectReaderNA** ซ้ำเพื่อเปลี่ยนเครื่องอ่านได้โดยไม่ต้องใช้คำสั่ง **deselectReaderNA** ก่อน

6.7 ฟังก์ชัน `int connectCardNA()`

Description	สั่งให้เชื่อมต่อกับบัตรประจำตัวประชาชนหรือบัตรธนาคาร จะต้องเรียกฟังก์ชันนี้อย่างน้อย 1 ครั้งก่อนใช้ฟังก์ชันอื่นในการอ่านบัตร
Input parameter	ไม่มี
Return value	0 คือเชื่อมต่ออยู่กับบัตรประจำตัวประชาชนไทย 1 คือเชื่อมต่ออยู่กับบัตรชนิดอื่นที่ไม่ใช่บัตรประจำตัวประชาชนไทย ค่าอื่น ๆ เป็น Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

6.8 ฟังก์ชัน `int disconnectCardNA()`

Description	ยกเลิกการเชื่อมต่อกับบัตรประจำตัวประชาชนหรือบัตรธนาคารที่เกิดจากฟังก์ชัน **connectCardNA** เพื่อคืนค่าหน่วยความจำให้ระบบ
Input parameter	ไม่มี
Return value	เป็น Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

6.9 ฟังก์ชัน `int getNIDNumberNA()`

Description	อ่านเลขประจำตัวประชาชนจากบัตร
Input parameter	ไม่มี
Return value	0: ฟังก์ชันทำงานปกติ, -1: ฟังก์ชันทำงานผิดพลาด
Callback Listener	**`onGetNIDNumberNA(String cardData, int result)`**
Output to Listener	**cardData** คือเลขประจำตัวประชาชนที่อ่านได้ **result** คือ Return Code ของ **getNIDNumberNA** มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

6.10 ฟังก์ชัน `int getNIDTextNA()`

ฟังก์ชัน `int getNIDTextNA(int getTextOption)`

Description	อ่านข้อมูลตัวอักษรของบัตรประจำตัวประชาชน
Input parameter	ไม่ต้องมี Input parameter หากต้องการอ่านเฉพาะข้อมูลหน้าบัตร หรือมี **getTextOption** เพื่อกำหนดให้อ่านข้อมูลเพิ่ม 0 = ไม่ต้องอ่านข้อมูลเพิ่ม

1 = อ่านข้อมูล AText ต่อท้ายข้อมูลหน้าบัตร

Return value 0: ฟังก์ชันทำงานปกติ, -1: ฟังก์ชันทำงานผิดพลาด

Callback Listener **`onGetNIDTextNA(String textData, int result)`**

Output to Listener **`textData`** คือข้อมูลตัวอักษรของบัตรประจำตัวประชาชนที่อ่านได้ ใช้รหัสแบบ UTF-8 ข้อมูลแต่ละรายการจะถูกคั่นด้วยเครื่องหมาย # ในลักษณะนี้
“เลขประจำตัวประชาชน#คำนำหน้านามไทย#ชื่อไทย#ชื่อรองไทย#นามสกุลไทย#คำนำหน้านามอังกฤษ#ชื่ออังกฤษ#ชื่อกลางอังกฤษ#นามสกุลอังกฤษ#เลขที่#หมู่ที่#ตรอก#ซอย#ถนน#ตำบล-แขวง#อำเภอ-เขต#จังหวัด#เพศ#วันเกิด#หน่วยงานที่ออกบัตร#วันที่ออกบัตร#วันบัตรหมดอายุ#เลขหมายคำขอ”
ในกรณีที่กำหนดให้ **`getTextOption = 1`** จะมีการเพิ่มข้อมูล AText อีก 5 รายการ ต่อท้าย เลขหมายคำขอ ดังนี้
#หมายเลขบัตร/คำร้อง#รหัสผู้ออกบัตร#รุ่นของโครงสร้าง#รหัสประเภทบัตร#รหัสคำนำหน้านาม”
`result` คือ Return Code ของ **`getNIDTextNA`** มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

ตารางรายละเอียดข้อมูลที่อ่านได้

	ข้อมูล	ความยาวข้อมูล	หมายเหตุ
1	เลขประจำตัวประชาชน	13ตัวอักษร	“1234567890123”
2	คำนำหน้านาม (ไทย)	รวม 4 ข้อมูลความยาวไม่เกิน 103 ตัวอักษร	
3	ชื่อ (ไทย)		
4	ชื่อรอง (ไทย)		
5	นามสกุล (ไทย)		
6	คำนำหน้านาม (อังกฤษ)	รวม 4 ข้อมูลความยาวไม่เกิน 103 ตัวอักษร	
7	ชื่อ (อังกฤษ)		
8	ชื่อกลาง (อังกฤษ)		
9	นามสกุล (อังกฤษ)		
10	ที่อยู่ปัจจุบัน (เลขที่)	ข้อมูลที่อยู่ทั้งหมดความยาวรวมไม่เกิน 167 ตัวอักษร	
11	หมู่ที่		
12	ตรอก		
13	ซอย		
14	ถนน		
15	ตำบล/แขวง		
16	อำเภอ/เขต		

	ข้อมูล	ความยาวข้อมูล	หมายเหตุ
17	จังหวัด		
18	เพศ	1 ตัวอักษร	“1” = ชาย, “2” = หญิง
19	วันเกิด	8 ตัวอักษร	ตัวเลข 8 หลักติดกันในรูปแบบ YYYYMMDD YYYY คือปี พ.ศ. MM คือเดือน กรณีไม่รู้เดือนเกิด MM = “00” DD คือวันที่ กรณีไม่รู้วันเกิด DD = “00”
20	หน่วยงานที่ออกบัตร	ไม่เกิน 100 ตัวอักษร	
21	วันออกบัตร	8 ตัวอักษร	ตัวเลข 8 หลักติดกันในรูปแบบ YYYYMMDD YYYY คือปี พ.ศ. MM คือเดือน DD คือวันที่
22	วันบัตรหมดอายุ	8 ตัวอักษร	ตัวเลข 8 หลักติดกันในรูปแบบ YYYYMMDD YYYY คือปี พ.ศ. MM คือเดือน DD คือวันที่ กรณีเป็นบัตรตลอดชีพ จะมีข้อมูลเป็น “99999999”
23	เลขหมายคำขอ	14 ตัวอักษร	เป็นอักขระใต้รูปถ่าย อาจเป็นตัวเลขหรือตัวอักษรก็ได้ บัตรประชาชนบางรุ่นจะไม่มีข้อมูลนี้
ข้อมูล AText ที่จะมีเพิ่มขึ้น เมื่อกำหนดให้ **`getTextOption = 1`**			
24	หมายเลขบัตร/คำร้อง	20 ตัวอักษร	“12345678901/12345678” 11 ตัวอักษรแรกคือ หมายเลขบัตร (BP1 Number) 8 ตัวอักษรท้ายคือ หมายเลขคำร้อง (Request Number) มีเครื่องหมาย ทับ (/) เป็นตัวแบ่ง
25	รหัสผู้ออกบัตร	13 ตัวอักษร	“1234567890123”
26	รุ่นของโครงสร้าง	4 ตัวอักษร	หมายเลขรุ่นของโครงสร้างข้อมูลบัตร เช่น “0003” หรือ “0004”
27	รหัสประเภทบัตร	2 ตัวอักษร	“01” 01 คือ บัตรประจำตัวประชาชน

	ข้อมูล	ความยาวข้อมูล	หมายเหตุ
28	รหัสคำนำหน้านาม (Title Flag)	1 ตัวอักษร	Space = ไม่มีข้อมูล “0” = ปกติ “1” = ยศทหารเรือ (ร.น.) “2” = สมณศักดิ์

Example กรณีไม่มี **getTextOption**

“3650800011234#นาย#สมชาย##สามรักสกุล#Mr.#Somchai##Samruksakul#123/45#หมู่ที่ 12####ตำบลบึงพระ#อำเภอเมืองพิษณุโลก#จังหวัดพิษณุโลก#1#25240119#เทศบาลนครพิษณุโลก#25521026#25590118#12341212345678”

Example กรณีกำหนดให้ **getTextOption = 1**

“3650800011234#นาย#สมชาย##สามรักสกุล#Mr.#Somchai##Samruksakul#123/45#หมู่ที่ 12####ตำบลบึงพระ#อำเภอเมืองพิษณุโลก#จังหวัดพิษณุโลก#1#25240119#เทศบาลนครพิษณุโลก#25521026#25590118#12341212345678**#12345678901/12345678#1234567890123#0004#01#0**”

Note กรณีกำหนดให้ **getTextOption = 1** จะมีข้อมูลต่อท้ายให้อีก 5 รายการ รวมจำนวนรายการขณะนี้อยู่ที่ 28 รายการ แต่ในอนาคตอาจปรับเพิ่มมากขึ้นได้อีก ในการเขียนโปรแกรมควรเผื่อกรณีนี้ไว้ด้วย

6.11 ฟังก์ชัน `int getNIDPhotoNA()`

Description อ่านรูปภาพถ่ายใบหน้าจากบัตรประจำตัวประชาชน

Input parameter ไม่มี

Return value 0: ฟังก์ชันทำงานปกติ, -1: ฟังก์ชันทำงานผิดพลาด

Callback Listener **`onGetNIDPhotoNA(byte[] cardData, int result)`**

Output to Listener **`cardData`** คือข้อมูลรูปภาพใบหน้าผู้ถือบัตรประชาชน รูปแบบไฟล์ JPEG ความยาวไม่เกิน 5120 ไบต์ ขนาดภาพ 297 x 355 จุด หรือ 148 x 178 จุด แล้วแต่รุ่นของบัตร

`result` คือ Return Code ของ **`getNIDPhotoNA`** มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

6.12 ฟังก์ชัน `int getCardStatusNA()`

Description ขอสถานะการเสียบบัตรที่เครื่องอ่าน

Input parameter ไม่มี

Return value 1 คือมีบัตรเสียบอยู่ที่เครื่องอ่าน

-16 คือไม่มีบัตรเสียบอยู่

ค่าอื่น ๆ เป็น Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

6.13 ฟังก์ชัน `int getRidNA(byte[] rid)`

Description ขอข้อมูลเลขประจำตัวของเครื่องอ่านบัตร (Reader ID หรือ RID)

Input parameter	**rid** พื้นที่ว่างอย่างน้อย 256 ไบต์
Output	**rid** ข้อมูล Reader ID ที่อ่านได้ เป็นข้อมูลแบบ Binary
Return value	กรณีเป็นเลขบวก จะเป็นความยาว (จำนวนไบต์) ของ Reader ID กรณีเป็นเลขลบ จะเป็น Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด
Note	คำสั่งนี้จะเรียกใช้ได้ในขณะที่เครื่องอ่านถูกเลือกแล้วเท่านั้น (จากคำสั่ง **selectReaderNA**)

6.14 ฟังก์ชัน `int getSoftwareInfoNA(String[] softwareInfo)`

Description	ขอข้อมูลของไลบารี
Input parameter	ไม่มี
Output	**softwareInfo[0]** ข้อมูลของไลบรารี ประกอบด้วย ชื่อ, รุ่นของไลบรารี และข้อมูลลิขสิทธิ์ โดยมีตัวอักษร # คั่น ในลักษณะนี้ “ชื่อและรุ่นของไลบรารี#ข้อมูลลิขสิทธิ์” เช่น “NALib 0.0.1#Copyright R&D Computer System Co., Ltd.”
Return value	เป็น Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

6.15 ฟังก์ชัน `int getLicenseInfoNA(String[] licenseInfo)`

Description	ขอข้อมูลของใบอนุญาต
Input parameter	ไม่มี
Output	**licenseInfo[0]** ข้อมูลใบอนุญาต ประกอบด้วย ชื่อแฟ้มใบอนุญาต (License File) และวันที่ของแฟ้ม (รูปแบบ YYYY-MM-DD) และข้อมูลใบอนุญาตแบบ Virtual Licensing โดยมีตัวอักษร # คั่น ในลักษณะนี้ “ชื่อแฟ้มใบอนุญาต#วันที่ของแฟ้มใบอนุญาต#ข้อมูลใบอนุญาตแบบ VL” เช่น “/storage/NASample/rdnidlib.dls#2019-12-10#V|R1.02|AR02”
Return value	เป็น Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

6.16 ฟังก์ชัน `int updateLicenseFileNA()`

Description	ปรับเปลี่ยนแฟ้มใบอนุญาต (License File) ให้เป็นปัจจุบัน โดยโหลดจากอินเทอร์เน็ตมาเก็บไว้ใน Path และชื่อแฟ้มตามที่ระบุไว้ใน ฟังก์ชัน **openLibNA**
Input parameter	ไม่มี
Return value	0: ฟังก์ชันทำงานปกติ, -1: ฟังก์ชันทำงานผิดพลาด
Callback Listener	**onUpdateLicenseFileNA(int result)**
Output to Listener	**result** คือ Return Code ของ **updateLicenseFileNA** กรณีเป็นเลขตั้งแต่ 0 ขึ้นไป แสดงว่าทำงานสำเร็จ โดยมีรายละเอียดดังนี้ 0, 1, 2 หรือ 3 คือ ได้อัปเดตแฟ้มใบอนุญาตเป็นแฟ้มใหม่แล้ว จากเครื่องแม่ข่าย หมายเลข 0, 1, 2 หรือ 3 ตามลำดับ

100, 101, 102 หรือ 103 คือ ได้มีการดาวน์โหลดแฟ้มใบอนุญาตจากเครื่องแม่ข่าย หมายเลข 0, 1, 2 หรือ 3 ตามลำดับแล้ว แต่ไม่ได้มีการอัปเดตเพราะแฟ้มที่มีอยู่ในเครื่องโทรศัพท์เป็นรุ่นล่าสุดอยู่แล้ว

กรณีเป็นเลขลบ แสดงว่าทำงานไม่สำเร็จ Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด เช่น

-15: ไม่สามารถติดต่ออินเทอร์เน็ตได้

-18: อัปเดตแฟ้มใบอนุญาตไม่สำเร็จ

Note

ให้เรียกฟังก์ชันนี้เมื่อพบว่าเครื่องอ่านที่นำมาใช้งานยังไม่ได้ลงทะเบียน หรือพบการแจ้งปัญหาเกี่ยวกับแฟ้มใบอนุญาต เช่น ไม่พบแฟ้มใบอนุญาต (Return Code = -12) หรือ ใบอนุญาตใช้ไม่ได้ (Return Code = -2) เพื่อให้ระบบทำการดาวน์โหลดแฟ้มใบอนุญาตรุ่นล่าสุดจากอินเทอร์เน็ตมาใช้งานแทน

ก่อนจะเรียกใช้งานฟังก์ชันนี้ ควรเปิดให้อุปกรณ์แอนดรอยด์เชื่อมต่อกับเครือข่ายอินเทอร์เน็ตให้ได้ก่อน มิฉะนั้น ฟังก์ชันนี้จะแจ้งคืนค่า -15 (ไม่สามารถติดต่ออินเตอร์เน็ตได้) หรือ -18 (อัปเดตแฟ้มใบอนุญาตไม่สำเร็จ)

สำหรับการใช้งานกับเครื่องอ่านแบบ Virtual Licensing ไม่จำเป็นต้องเรียกใช้งานฟังก์ชันนี้ เพราะเครื่องอ่านแบบ Virtual Licensing ไม่มีการใช้แฟ้มใบอนุญาต

6.17 ฟังก์ชัน `int setPermissionsNA(int pms)`

Description

เป็นคำสั่งกำหนดค่าการอนุญาต (Permission) ที่จะมีการใช้งานภายในแอปพลิเคชัน

Input parameter

`pms` คือ Permission ที่จะกำหนด

0: ปิดการขออนุญาตใช้เครื่องอ่าน USB ภายในแอปพลิเคชัน

1: ให้มีการขออนุญาตใช้เครื่องอ่าน USB ภายในแอปพลิเคชัน (USB Reader **In-App** Permission) ซึ่งจะมีหน้าต่าง Pop-up ขึ้นมาถาม

-1: สั่งอ่านค่าปัจจุบันของการตั้งค่าการขออนุญาต

Return value

0: การขออนุญาตใช้เครื่องอ่าน USB ภายในแอปพลิเคชัน ปิดอยู่

1: การขออนุญาตใช้เครื่องอ่าน USB ภายในแอปพลิเคชัน เปิดใช้งานอยู่

กรณีเป็นเลขลบ จะเป็น Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

Note

โดยปกติค่าการขออนุญาตใช้เครื่องอ่าน USB ภายในแอปพลิเคชันจะเป็น 0 คือปิดอยู่ หากตั้งเป็น 1 จะมีหน้าต่าง Pop-up เพื่อขออนุญาตใช้เครื่องอ่าน USB จากภายในแอปพลิเคชัน

6.18 ฟังก์ชัน `int getReaderInfoNA(String[] readerInfo)`

Description

ขอข้อมูลของเครื่องอ่านที่กำลังเชื่อมต่ออยู่

Input parameter

ไม่มี

Output

`readerInfo[0]` คือ ข้อมูลของเครื่องอ่านที่ขอได้ ประกอบด้วย ชนิดของการเชื่อมต่อ, ชนิดของ License และชื่อรุ่นของเครื่องอ่าน โดยมีตัวอักษร # คั่น ในลักษณะนี้

“Interface#License Type#Model#Firmware Version”

	Interface	USB: เป็นเครื่องอ่านแบบ USB BLT: เป็นเครื่องอ่านแบบบลูทูธ BLE: เป็นเครื่องอ่านบลูทูธแบบ BLE
	License Type	VL: เป็นเครื่องอ่านแบบ Virtual Licensing FL: เป็นเครื่องอ่านแบบ File Licensing NL: เป็นเครื่องอ่านที่ไม่มีใบอนุญาต
	Model	ชื่อรุ่นของเครื่องอ่าน
	Firmware Version	รุ่นเฟิร์มแวร์ของเครื่องอ่าน
	ตัวอย่างเช่น “USB#VL#TDAi301VC#8.00”	
Return value	เป็น Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด	
Note	คำสั่งนี้จะเรียกใช้ได้ในขณะที่เครื่องอ่านถูกเลือกแล้วเท่านั้น (จากคำสั่ง **`selectReaderNA`**)	

6.19 ฟังก์ชัน `int getEmvTextCA()`

Description	อ่านข้อมูลตัวอักษรของบัตรธนาคาร
Input parameter	ไม่มี
Return value	0: ฟังก์ชันทำงานปกติ, -1: ฟังก์ชันทำงานผิดพลาด
Callback Listener	**`onGetEmvTextCA(String textData, int result)`**
Output to Listener	**`textData`** คือข้อมูลตัวอักษรของบัตรธนาคารที่อ่านได้ ข้อมูลแต่ละรายการจะถูกคั่นด้วยเครื่องหมาย # ในลักษณะนี้ “เลขบัตรธนาคาร#ชื่อเจ้าของบัตร#เดือนปีหมดอายุ#ชื่อเครือข่าย” **`result`** คือ Return Code ของ **`getEmvTextCA`** มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

ตารางรายละเอียดข้อมูลที่อ่านได้

	ข้อมูล	ความยาวข้อมูล	หมายเหตุ
1	เลขบัตรธนาคาร (PAN)	13-19 ตัวอักษร	เช่น “1234567890123456”
2	ชื่อเจ้าของบัตร (Holder Name)	0 ถึง 26 ตัวอักษร	กรณีมีการแยกชื่อ (First Name) กับนามสกุล (Last Name) จะใช้รูปแบบ “Last Name/First name” เช่น “RAKSAKUL/SOMCHAI”
3	เดือนปีหมดอายุ (Expiry Date)	4 ตัวอักษร	รูปแบบ YYMM YY คือเลข 2 หลักหลังของปี ค.ศ. MM คือเลขเดือน 2 หลัก เช่น “2912” หมายถึงเดือนธันวาคม ปี 2029
4	ชื่อเครือข่าย (Network)	ไม่กำหนด	เช่น “Visa”

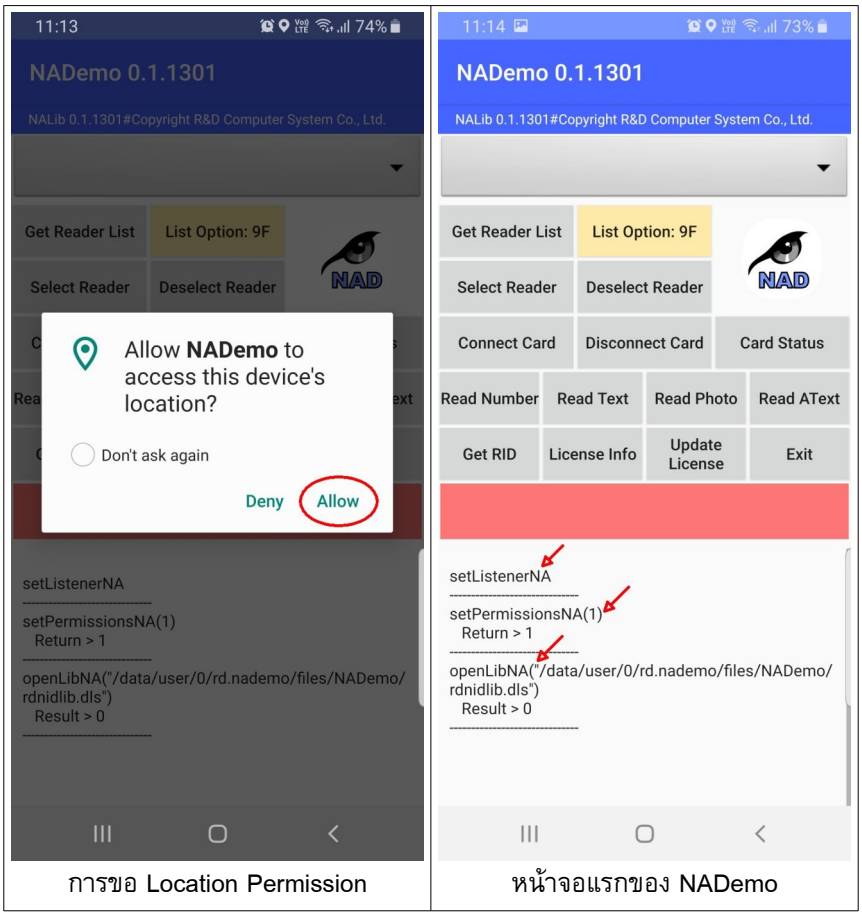
Example	“1234567890123456#RAKSAKUL/SOMCHAI#2912#Visa”
Note	คำสั่งนี้ใช้ได้เฉพาะกับสินค้ารุ่นที่รองรับการอ่านบัตรธนาคารเท่านั้น เช่นสินค้าในตระกูล TDC

6.20 รหัสส่งกลับและรหัสข้อผิดพลาด (Return Code) ที่ได้จากฟังก์ชันต่าง ๆ

Code	Name	ความหมาย
0	NA_SUCCESS	ทำงานได้สำเร็จเสร็จสิ้น
-1	NA_INTERNAL_ERROR	เกิดข้อผิดพลาดภายในระบบ
-2	NA_INVALID_LICENSE	เครื่องอ่านนี้ใช้ไม่ได้เพราะใบอนุญาตไม่ถูกต้อง
-3	NA_READER_NOT_FOUND	ไม่พบเครื่องอ่านบัตร
-4	NA_CONNECTION_ERROR	ไม่สามารถติดต่อกับบัตรได้
-5	NA_GET_PHOTO_ERROR	ไม่สามารถอ่านรูปภาพถ่ายใบหน้าได้
-6	NA_GET_TEXT_ERROR	ไม่สามารถอ่านข้อมูลตัวอักษรได้
-7	NA_INVALID_CARD	บัตรที่อ่านไม่ใช่บัตรประชาชน
-8	NA_UNKNOWN_CARD_VERSION	ไม่รองรับการใช้งานกับบัตรประชาชนรุ่นนี้
-9	NA_DISCONNECTION_ERROR	ไม่สามารถยกเลิกการเชื่อมต่อกับเครื่องอ่านบัตรได้
-10	NA_INIT_ERROR	กระบวนการตั้งค่าเริ่มต้นทำงานผิดพลาด หรือยังไม่ได้เรียกใช้งาน **openLibNA**
-11	NA_READER_NOT_SUPPORTED	ไม่รองรับเครื่องอ่านนี้ หรือไม่พบเครื่องอ่านบัตร
-12	NA_LICENSE_FILE_ERROR	ไม่พบแฟ้มใบอนุญาตหรือแฟ้มใบอนุญาตเสียหาย
-13	NA_PARAMETERS_ERROR	พารามิเตอร์หรือตัวแปรผิดพลาด
-15	NA_INTERNET_ERROR	ไม่สามารถติดต่ออินเทอร์เน็ตได้
-16	NA_CARD_NOT_FOUND	ไม่พบบัตรในเครื่องอ่าน
-17	NA_BLUETOOTH_DISABLED	ไม่ได้เปิดบลูทูธ
-18	NA_LICENSE_UPDATE_ERROR	อัปเดตแฟ้มใบอนุญาตไม่สำเร็จ
-19	NA_READER_SELECTION_ERROR	พบปัญหาการเลือกเครื่องอ่าน
-20	NA_FUNCTION_NOT_SUPPORTED	ไม่รองรับการใช้งานฟังก์ชันนี้
-31	NA_STORAGE_PERMISSION_ERROR	มีข้อผิดพลาดหรือไม่ได้รับอนุญาตให้ใช้พื้นที่จัดเก็บ (Storage)
-32	NA_LOCATION_PERMISSION_ERROR	มีข้อผิดพลาดหรือไม่ได้รับอนุญาตให้รับรู้ตำแหน่งของบลูทูธ (Bluetooth Location)
-33	NA_BLUETOOTH_PERMISSION_ERROR	มีข้อผิดพลาดหรือไม่ได้รับอนุญาตให้ใช้งานบลูทูธ
-41	NA_LOCATION_SERVICE_ERROR	บริการตำแหน่ง (Location Service) ปิดอยู่ หรือมีข้อผิดพลาด

7. การทดสอบการใช้คำสั่งด้วยแอปพลิเคชัน NADemo

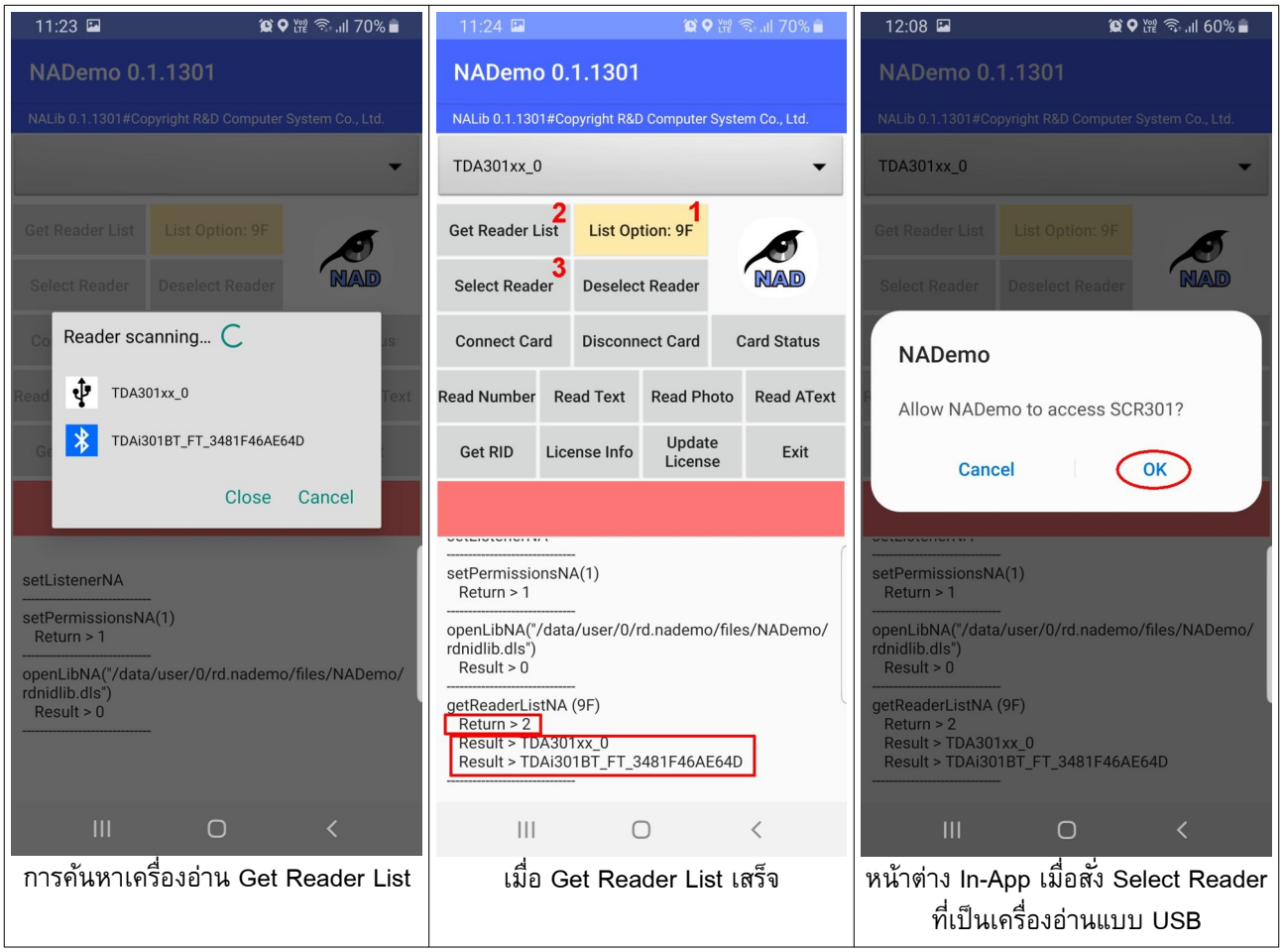
- NADemo เป็นแอปพลิเคชันตัวอย่าง และเป็นเครื่องมือสำหรับการเรียนรู้ และทดสอบชุดคำสั่งของ NALib API
- ในชุดซอฟต์แวร์ SDK ได้เตรียมแฟ้ม NADemo.apk ไว้ให้แล้ว สามารถติดตั้งและเรียกใช้งานได้ทันที
- และในชุดซอฟต์แวร์ SDK ก็ได้เตรียมแฟ้มต้นฉบับ (Source Code) ของ NADemo ไว้ให้แล้วด้วย เผื่อผู้พัฒนาที่ต้องการสามารถไล่ศึกษาดูการทำงานของ NADemo ได้โดยสะดวก
- การใช้งาน NADemo
  - การใช้งานครั้งแรก จะมีการขออนุญาต Access location ควรตอบ **ALLOW** (หรือ **ALLOW ONLY WHILE USING THE APP**) เพื่อใช้งานกับเครื่องอ่านแบบบลูทูธได้ หากตอบ **DENY** จะทำให้ NADemo ไม่สามารถทดสอบการทำงานกับเครื่องอ่านแบบบลูทูธได้ (จะได้ Error Code เป็น -32)



การขอ Location Permission | หน้าจอแรกของ NADemo

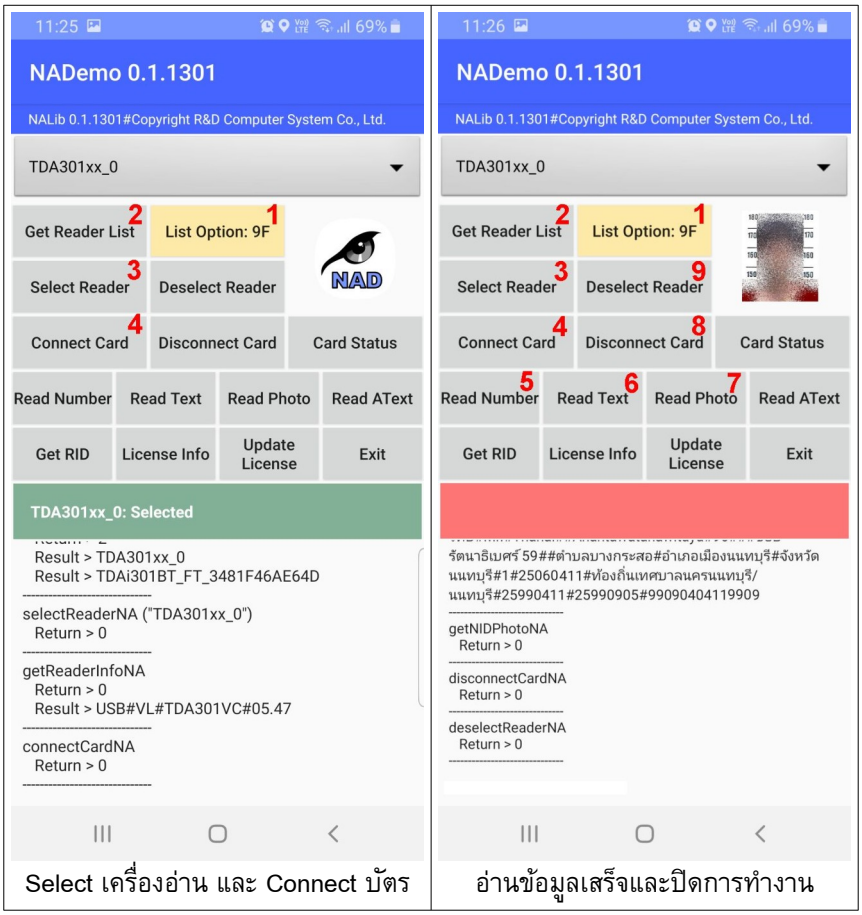
  - เมื่อแอปพลิเคชันทำงาน ให้สังเกตหน้าจอแรก ช่องด้านล่าง จะมีการแสดงข้อความว่ามีการเรียกใช้คำสั่ง **`setListenerNA`**, **`setPermissionsNA(1)`** และ **`openLibNA`** พร้อมแสดงผล Result ของการใช้คำสั่ง ค่า Result ที่แสดงนี้ เช่น 0 หรือ -12 สามารถดูความหมายได้จากคำอธิบายในแต่ละคำสั่ง และใน**ตารางรหัสส่งกลับและรหัสข้อผิดพลาด**ในหัวข้อ NALib API

- คำสั่ง **`setPermissionsNA(1)`** คือการกำหนดให้ NADemo มี Pop-up ขอใช้งานเครื่องอ่าน USB ภายในแอปพลิเคชัน (USB Reader In-App Permission) (NADemo เป็นแอปพลิเคชันตัวอย่างที่ใช้งาน In-App อย่างเดียว ไม่มี Out-App)
- หากจะทดสอบกับเครื่องอ่านแบบ USB ให้เสียบเครื่องอ่าน USB เข้ากับโทรศัพท์มือถือที่รองรับ
  - หากเสียบเครื่องอ่านแล้ว มีหน้าจอ Pop-up ขึ้นมาให้ “เลือกแอปพลิเคชันสำหรับอุปกรณ์ USB” หรือ “Choose an app for the USB device” ให้กด Back (ไม่ต้องเลือก)
- หากจะทดสอบกับเครื่องอ่านแบบ BLE ให้กดสวิตช์เปิดที่เครื่องอ่านรอจนไฟสีน้ำเงินกระพริบแล้วเปิดระบบสัญญาณบลูทูธของโทรศัพท์มือถือ **แต่ไม่ต้องสั่งจับคู่ (Pair)**
- หากจะทดสอบกับเครื่องอ่านแบบบลูทูธธรรมดา ให้กดสวิตช์เปิดที่เครื่องอ่านรอจนไฟสีน้ำเงินกระพริบ แล้วเปิดระบบสัญญาณบลูทูธของโทรศัพท์มือถือ พร้อมกับจับคู่ (Pair) กับเครื่องอ่านโดยเข้าไปที่ **Settings (ตั้งค่า)** ของระบบแอนดรอยด์ > แตะที่ B**luetooth (บลูทูธ)** > **Scan (ค้นหา)** แล้วดูที่บรรทัดล่างสุดของหน้าจอ เลือกเครื่องอ่านที่ขึ้นต้นด้วย “FT_” แล้วตอบตกลง
- ให้แตะที่ช่อง List Option (ช่องสีเหลือง) (1) เพื่อกำหนดค่า List Option เป็น 93 (สำหรับเครื่องอ่าน USB+Bluetooth) หรือ 9F (สำหรับเครื่องอ่าน USB+Bluetooth+BLE)
- แตะปุ่ม Get Reader List (2) จะมีหน้าจอการค้นหาเครื่องอ่านแสดงขึ้นมา ให้แตะเลือกเครื่องอ่านที่ต้องการใช้ หรือแตะที่ปุ่ม OK แต่หากยังไม่พบเครื่องอ่านให้แตะที่ Re-scan เพื่อค้นหาอีกครั้งหนึ่ง
- ให้ดูผลการค้นหาเครื่องอ่านของคำสั่ง **`getReaderListNA`** ที่ช่องด้านล่าง ค่า Return คือจำนวนเครื่องอ่านที่พบ (ว่าพบกี่เครื่อง) ค่า Result คือชื่อเครื่องอ่านที่พบ



การค้นหาเครื่องอ่าน Get Reader List | เมื่อ Get Reader List เสร็จ | หน้าต่าง In-App เมื่อสั่ง Select Reader ที่เป็นเครื่องอ่านแบบ USB

- ที่ช่อง Drop-down ด้านบนของหน้าจอ ให้ลองแตะที่ลูกศรสามเหลี่ยม แอปพลิเคชันจะแสดงรายชื่อเครื่องอ่านที่รับมาได้จากคำสั่ง **`getReaderListNA`** ให้เลือกเครื่องอ่านที่ต้องการจะใช้งาน (ปกติจะเลือกเครื่องอ่านเครื่องแรกให้เป็นค่าปริยาย)
- แตะที่ปุ่ม Select Reader (3) เพื่อส่งคำสั่ง **`selectReaderNA`** ไปพร้อมกับชื่อเครื่องอ่านที่เลือกในช่อง Drop-down แล้วดูผล
- หากเลือกเครื่องอ่านแบบ USB แอปจะมีการถามขออนุญาตที่จะติดต่อเครื่องอ่าน เช่น **“อนุญาตให้ NADemo เข้าถึง XXXX ไหม”** หรือ **“Allow NADemo to access XXXX?”** เมื่อ XXXX คือชื่อเครื่องอ่านของระบบ (ซึ่งอาจจะไม่ตรงกับชื่อรุ่นของสินค้าจริง ไม่ต้องสนใจ) ให้ตอบ **OK**
- ให้เสียบบัตรประชาชนเข้ากับเครื่องอ่าน แล้วแตะที่ปุ่ม Connect Card (4) แล้วดูผล ควรได้ค่า Return เป็น 0
- แตะที่ปุ่ม Read Number (5) แล้วดูผล
- แตะที่ปุ่ม Read Text (6) แล้วดูผล
- แตะที่ปุ่ม Read Photo (7) แล้วดูผล



Select เครื่องอ่าน และ Connect บัตร | อ่านข้อมูลเสร็จและปิดการทำงาน

- เมื่ออ่านข้อมูลจากบัตรเสร็จแล้ว หากต้องการอ่านบัตรใบใหม่ ให้หยุดการใช้งานบัตรเดิมโดยแตะที่ปุ่ม Disconnect Card จึงค่อยดึงบัตรเก่าออก และเสียบบัตรใบใหม่ แตะที่ปุ่ม Connect Card จากนั้นค่อยอ่านด้วยปุ่ม Read Number, Read Text หรือ Read Photo ตามต้องการ
- หากต้องการเปลี่ยนไปใช้งานเครื่องอ่านเครื่องอื่น ให้แตะที่ปุ่ม Disconnect Card (8) และ Deselect Reader (9) ตามลำดับ แล้วค่อยเลือกเครื่องอ่านใหม่จากช่อง Drop-down หรือ สั่ง Get Reader List ใหม่อีกครั้ง

- คำแนะนำในการทดสอบเพิ่มเติม
  - ค่า List Option ที่แนะนำให้ลองทดสอบคือ
    - ค่า 93 จะ Scan หาเครื่องอ่าน USB ที่เสียบอยู่ และเครื่องอ่านแบบบลูทูธที่เปิดอยู่ทุกเครื่อง จนกว่าจะหมดเวลาโดยขณะค้นหาจะมีหน้าต่างโต้ตอบแสดงให้ผู้ใช้เห็น ผู้ใช้สามารถเลือกเครื่องอ่านที่ต้องการได้เอง
    - ค่า D3 จะ Scan หาเครื่องอ่าน USB ที่เสียบอยู่ และเครื่องอ่านแบบบลูทูธที่เปิดอยู่ และกลับทันทีเมื่อพบเครื่องอ่านเครื่องแรก (หากเสียบเครื่องอ่าน USB จะเลือกเครื่องอ่าน USB ทันที หากไม่ได้เสียบเครื่องอ่าน USB ก็จะรอหาเครื่องอ่านบลูทูธเครื่องแรก)
    - ค่า 01 เหมาะสำหรับ Project ที่ใช้งานเครื่องอ่านแบบ USB เพียงชนิดเดียว ไม่ต้องการใช้เครื่องอ่านแบบบลูทูธเลย วิธีนี้จะทำงานเร็วที่สุด
    - ค่า 94 เหมาะสำหรับ Project ที่ใช้งานเครื่องอ่านบลูทูธแบบ BLE0 เพียงชนิดเดียว

- ค่า 98 เหมาะสำหรับ Project ที่ใช้งานเครื่องอ่านบลูทูธแบบ BLE1 (TDA3901BE) เพียงชนิดเดียว
- ค่า 9F เหมาะสำหรับ Project ที่ต้องการเผื่อให้ใช้งานเครื่องอ่านได้ทุกแบบ (USB, บลูทูธธรรมดา และบลูทูธแบบ BLE0 และ BLE1)

- การทดสอบด้วยเครื่องอ่านแบบ USB กับแอปพลิเคชันอ่านบัตรหลาย ๆ ตัว
  - หากทุกแอปเลือกการทำงานแบบ USB Reader In-App Permission จะสามารถสลับการใช้งานระหว่างแอปได้ทันทีโดยไม่ต้องถอดเสียบเครื่องอ่าน USB
  - หากทุกแอปเลือกการทำงานแบบ USB Reader Out-App Permission เพียงอย่างเดียว เวลาจะเปลี่ยนไปใช้งานแอปอื่น จะต้องดึงเครื่องอ่าน USB ออกแล้วเสียบใหม่ แล้วเลือกชื่อแอปพลิเคชันใหม่ให้ถูกต้องก่อนจึงจะใช้งานได้

8. NASample และตัวอย่างขั้นตอนการใช้คำสั่งในการพัฒนาแอปพลิเคชัน

NASample เป็นแอปพลิเคชันตัวอย่างแบบง่าย ๆ พร้อมโปรแกรมต้นฉบับ (Source Code) สำหรับให้ผู้พัฒนาใช้เป็นแม่แบบในการพัฒนา ในชุดซอฟต์แวร์ได้เตรียมแฟ้มต้นฉบับ และแฟ้มที่สร้างเป็น APK ไว้ให้แล้ว ผู้พัฒนาสามารถนำเข้าไปทดสอบใน Android Studio ได้ทันที และสามารถพัฒนาต่อยอดจาก Source Code นี้ได้

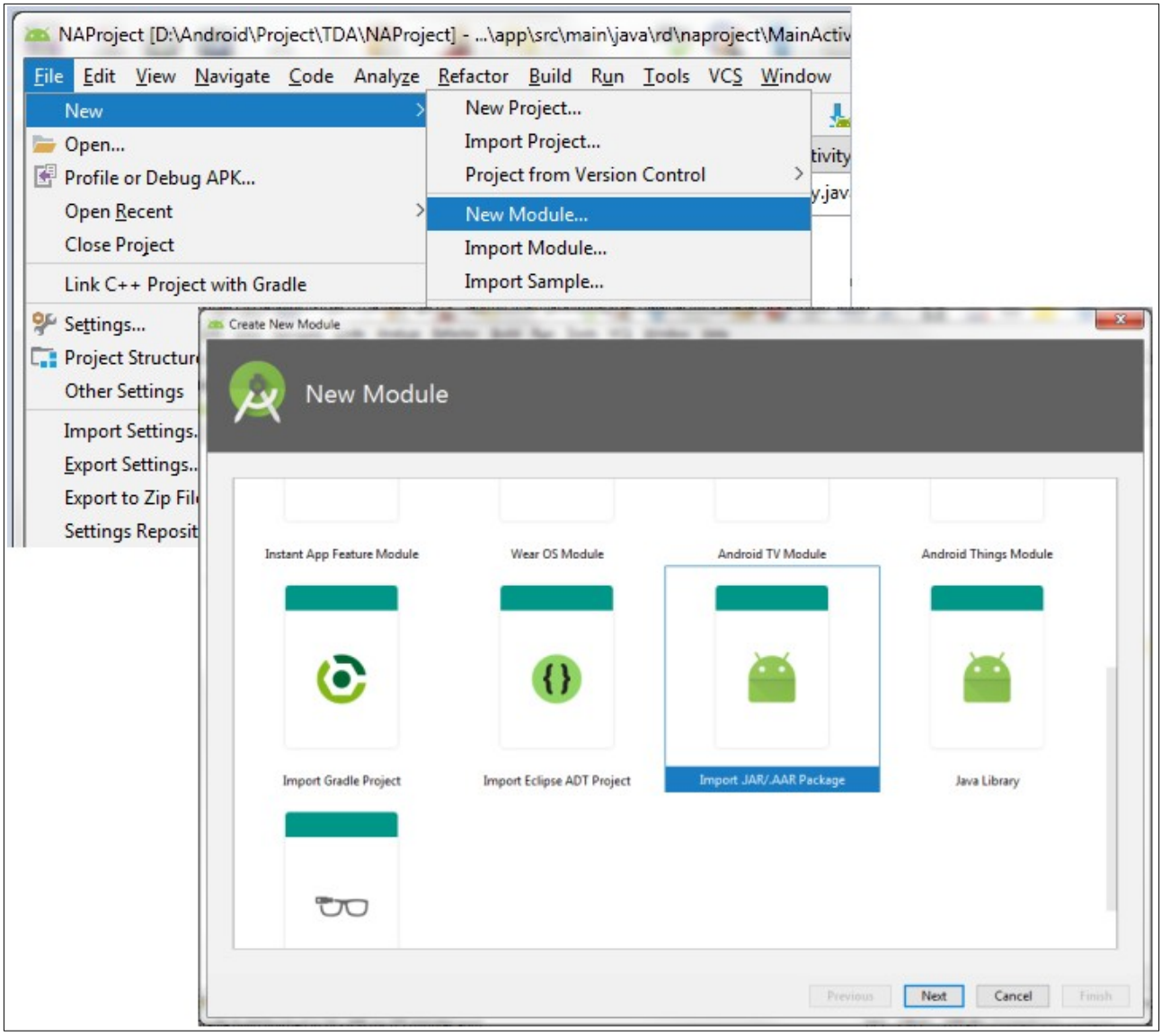
ตัวอย่างขั้นตอนการพัฒนาและการใช้คำสั่งในการเขียนโปรแกรมสามารถดูได้จาก Source Code ของ NASample ซึ่งมีขั้นตอนดังต่อไปนี้

8.1 การImport NALib.aar

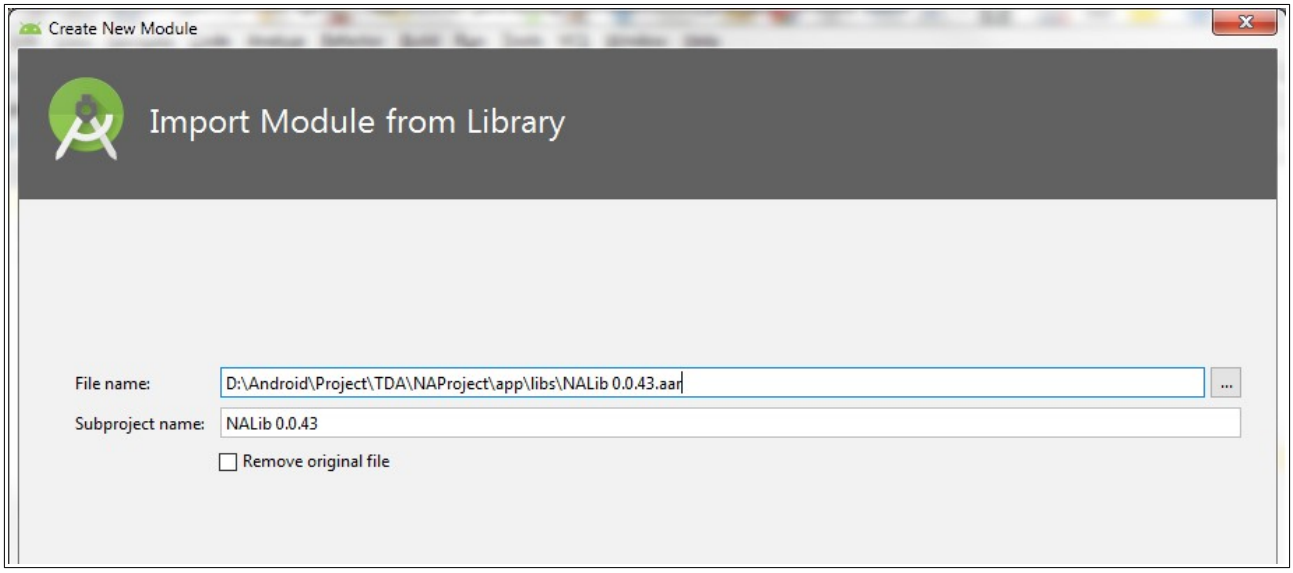
8.1.1 สร้างโปรเจ็กต์ใหม่ภายใต้ Android Studio (ควรเป็นรุ่น 3.2 ขึ้นไป)

8.1.2 เลือก File > New Module…

8.1.3 เลือก Import .JAR/.AAR Package แล้วกด Next



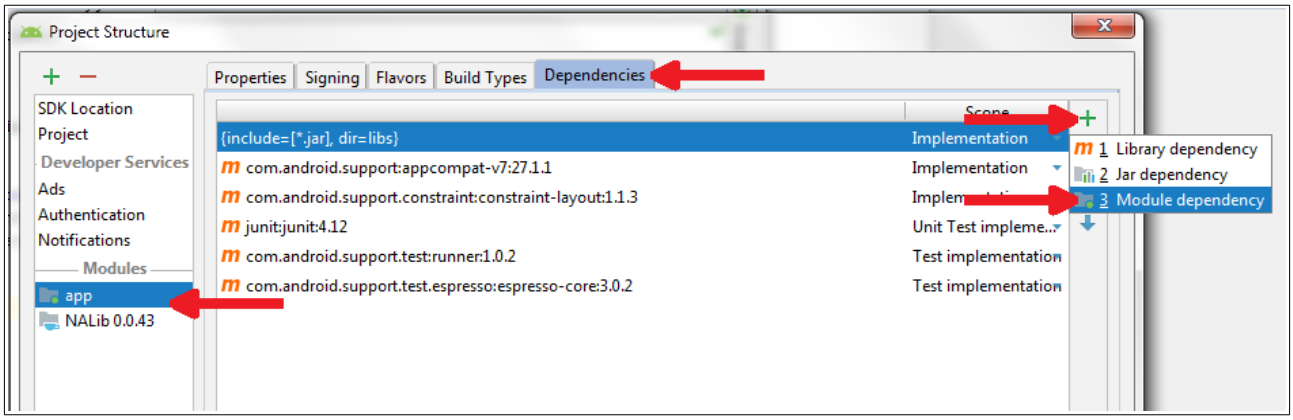
8.1.4 ระบุ Path ของไลบรารี NALib.aar ในช่อง File Name: แล้ว คลิก Finish



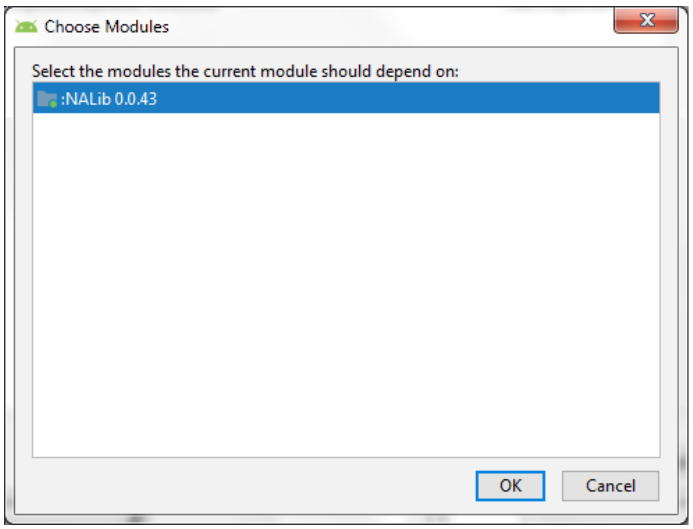
8.1.5 คลิกเมนู File > Project Structure…

8.1.6 คลิกที่ Module ที่เป็น Main Module (ในที่นี้ Main Module ชื่อ app) แล้วเลือกแท็บ Dependencies

8.1.7 คลิกที่รูปเครื่องหมายบวก (+) ทางด้านขวา แล้วเลือก 3 Module dependency



8.1.8 เลือก Module NALib ที่เราทำการ Import เข้ามา แล้วกด OK

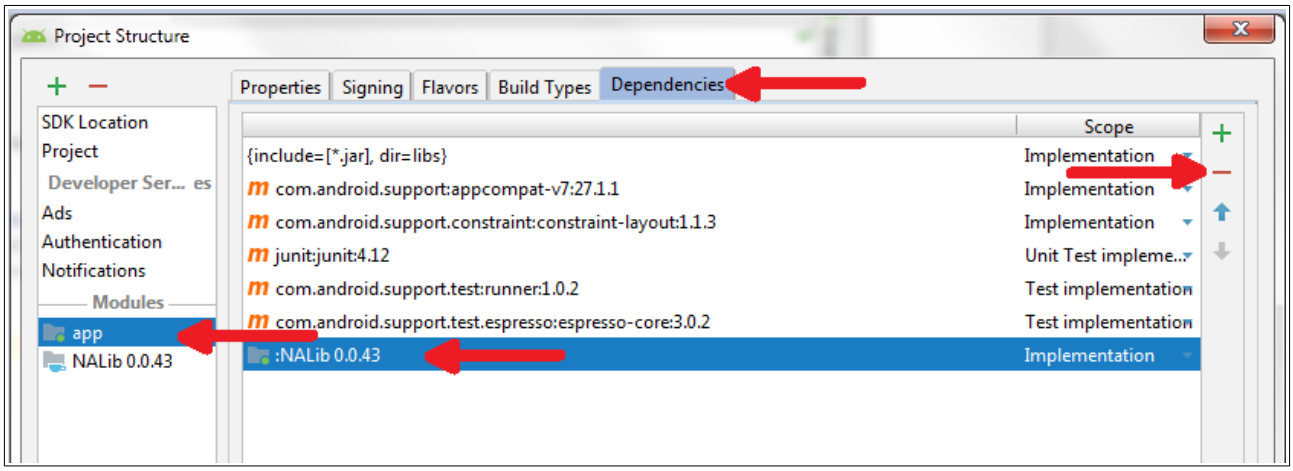


8.2 การลบ NALib.aar เมื่อมีการเปลี่ยนรุ่นของ NALib.aar

8.2.1 คลิกเมนู File > Project Structure…

8.2.2 คลิกที่ Module ที่เป็น Main Module (ในที่นี้ Main Module ชื่อ app) แล้วเลือกแท็บ Dependencies

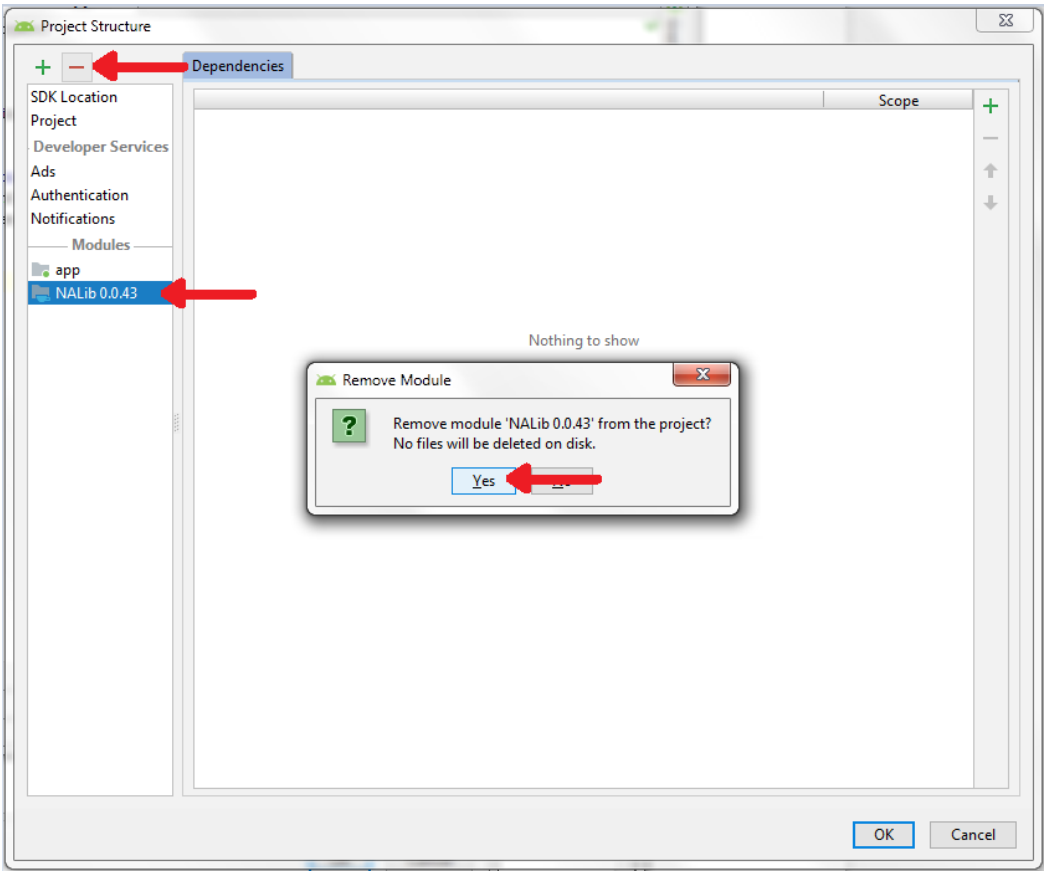
8.2.3 เลือกModule NALib แล้วคลิกที่เครื่องหมายลบ (-) ทางด้านขวา แล้วกด OK



8.2.4 คลิกเมนู File > Project Structure…

8.2.5 คลิกที่ Module NALibแล้วคลิกที่เครื่องหมายลบ (-) ทางด้านบน

8.2.6 กด Yes และ กด OK



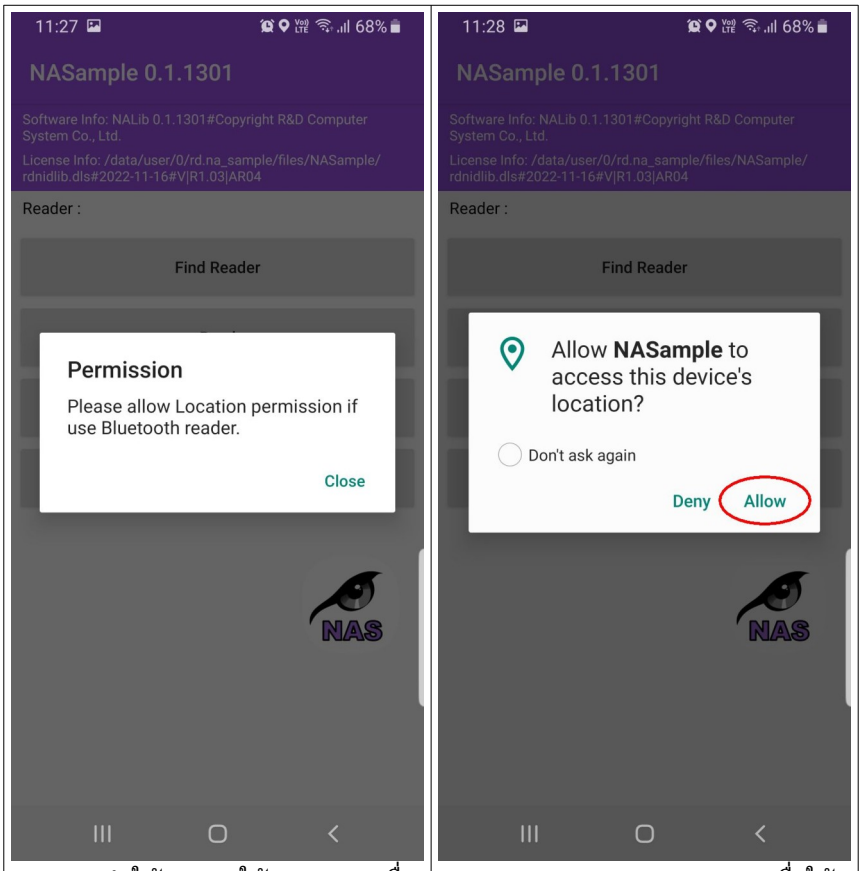
8.3 ขั้นตอนในการอ่านบัตรประชาชนและบัตรธนาคาร ให้เรียกใช้คำสั่งตามลำดับดังนี้

- **`setListenerNA`**: ผูกอินเตอร์เฟซให้กับ ResponseListener
- **`setPermissionsNA(1)`** : หากต้องการให้มี Pop-up ขอใช้งานเครื่องอ่าน USB ภายในแอปพลิเคชัน (USB Reader In-App Permission)
- **`openLibNA`**: เปิดใช้งาน NALib โดยต้องเลือกว่าจะให้ทำงานแบบ VL อย่างเดียว หรือจะให้ทำงานแบบผสม FL+VL
  - ตรวจสอบผลการเปิดใช้งานจาก **`onOpenLibNA`**
- **`getReaderListNA`**: ขอรายชื่อเครื่องอ่าน
  - รับผลและรายชื่อเครื่องอ่านจาก **`onGetReaderListNA`**
- **`selectReaderNA`**: เลือกใช้งานเครื่องอ่านที่ต้องการ
  - ตรวจสอบผลจาก **`onSelectReaderNA`**
- **`connectCardNA`**: เชื่อมต่อกับบัตรประจำตัวประชาชนหรือบัตรธนาคาร
- การอ่านข้อมูลจากบัตรประชาชน
  - **`getNIDTextNA`**: อ่านข้อมูลตัวอักษรจากบัตร
    - รับข้อมูลตัวอักษรผ่าน **`onGetNIDTextNA`**
  - **`getNIDPhotoNA`**: อ่านข้อมูลรูปภาพจากบัตร
    - รับข้อมูลรูปภาพผ่าน **`onGetNIDPhotoNA`**
- การอ่านข้อมูลจากบัตรธนาคาร
  - **`getEmvTextCA`**: อ่านข้อมูลตัวอักษรจากบัตรธนาคาร
    - รับข้อมูลตัวอักษรผ่าน **`onGetEmvTextCA`**
- **`disconnectCardNA`**: ยกเลิกการเชื่อมต่อกับบัตร
- **`deselectReaderNA`**: ยกเลิกการใช้เครื่องอ่าน
- **`closeLibNA`**: ยกเลิกการใช้ NALib

8.4 อธิบายการทำงานของแอปพลิเคชันตัวอย่าง NASample

8.4.1 เมื่อแอปพลิเคชันเริ่มทำงาน

- จะมีการขออนุญาตการใช้งาน Location แต่ก่อนจะขอ NASample จะมีการแสดงหน้าต่างแนะนำให้ผู้ใช้ว่าต้องอนุญาตให้ใช้ Location ได้หากต้องการใช้งานเครื่องอ่านแบบบลูทูธ ดังนั้นจึงให้ตอบ **ALLOW** (หรือ **ALLOW ONLY WHILE USING THE APP**) เมื่อถูกขอ
  - หากตอบ DENY จะทำให้แอปปิดการทำงานที่เกี่ยวกับ Bluetooth ทั้งหมด เหลือเฉพาะการติดต่อกับเครื่องอ่านแบบ USB เท่านั้น



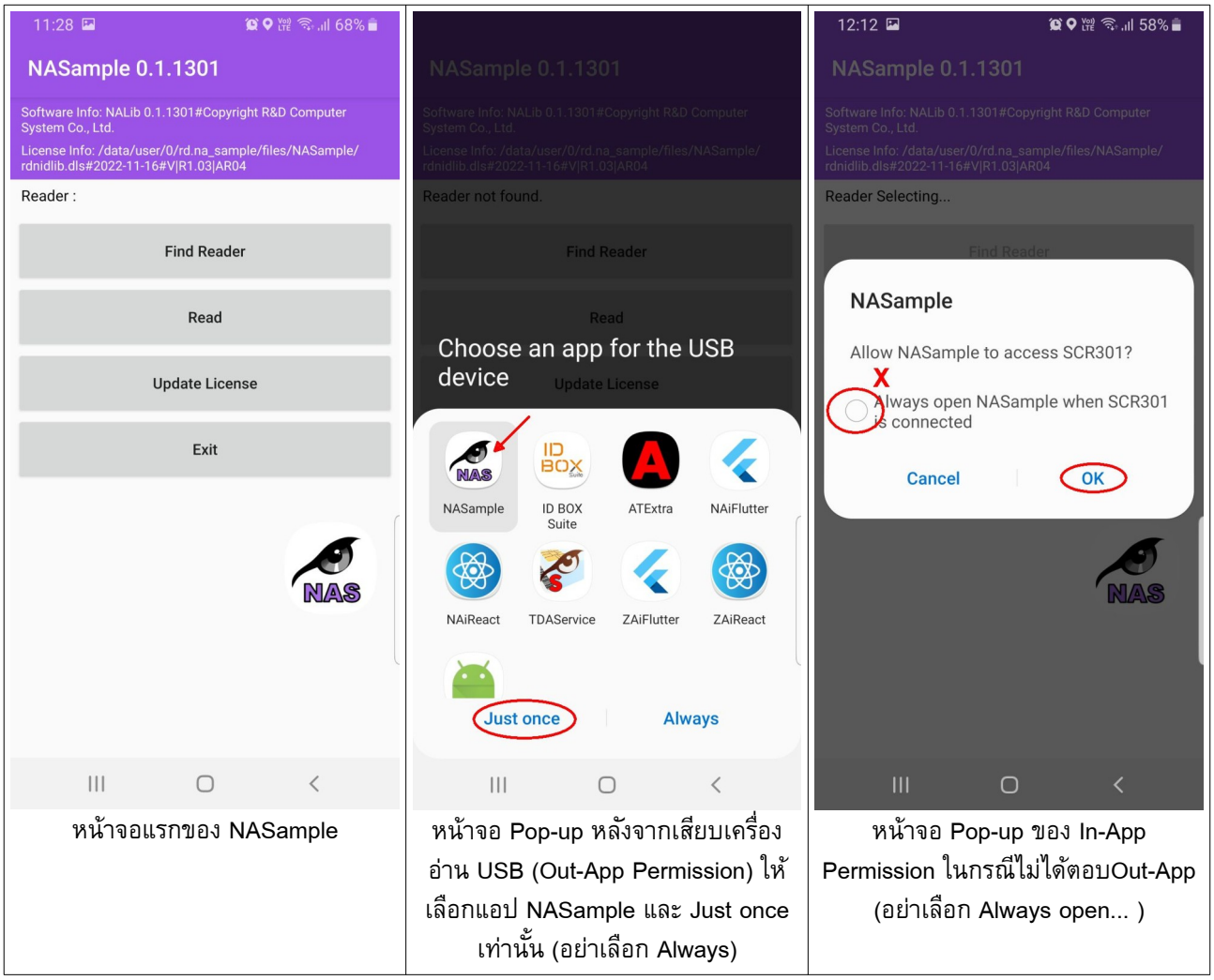
การแนะนำให้อนุญาตใช้ Location เพื่อใช้งานเครื่องอ่านแบบบลูทูธ

การขอ Location Permission เพื่อใช้งานเครื่องอ่านแบบบลูทูธ

- จะเรียก **`setListenerNA(listener)`** เพื่อแจ้งให้ NALib รู้ว่าจะให้ Callback เพื่อส่งข้อมูลกลับมาในชื่อ Listener ใด โดยต้องเรียกคำสั่งนี้ก่อนการใช้งานคำสั่งอื่น ๆ
- จะเรียกใช้งานคำสั่ง **`setPermissionsNA(1)`** คือการกำหนดให้ NASample มี Pop-up ขอใช้งานเครื่องอ่าน USB ภายในแอปพลิเคชันด้วย (USB Reader In-App Permission) (NASample เป็นแอปพลิเคชันตัวอย่างที่ใช้งานทั้ง Out-App และ In-App)
- เรียก **`openLibNA(licenseFile)`** เพื่อเปิดการใช้งาน NALib โดยต้องมีการกำหนดชื่อแฟ้มและ Path ที่อยู่ของแฟ้มใบอนุญาต สำหรับ NASample จะมีการใช้แฟ้มใบอนุญาตเสมอ เพื่อให้รองรับการทดสอบกับเครื่องอ่านได้ทั้งแบบ FL และ VL

8.4.2 การเตรียมเครื่องอ่านเพื่อการทดสอบ NASample

- หากจะทดสอบกับเครื่องอ่านแบบ USB ให้เสียบเครื่องอ่าน USB เข้ากับโทรศัพท์มือถือที่รองรับระบบ OTG รอสักครู่จะมีหน้าจอ Pop-up ของ Out-App Permission แสดงขึ้นมาให้เลือกว่าจะใช้งานอุปกรณ์ USB นี้กับแอปพลิเคชันใด ให้เลือกที่ NASample (หรือจะไม่เลือกก็ได้)
  - หากเสียบแล้ว ไม่มีหน้าจอ Pop-Up ขึ้นมาถาม เป็นไปได้ว่าโทรศัพท์มือถือเครื่องนั้นไม่รองรับระบบ OTG ให้เปลี่ยนไปใช้โทรศัพท์มือถือรุ่นอื่นที่รองรับแทน
  - หากเลือกแอปอื่น หรือไม่ได้เลือกแอปใดเลย NASample จะมีการถามอีกครั้งภายในแอปในตอนค้นหาเครื่องอ่าน เพราะ NASample มีการทำงานทั้งแบบ Out-App และ In-App Permission



หน้าจอแรกของ NASample

หน้าจอ Pop-up หลังจากเสียบเครื่องอ่าน USB (Out-App Permission) ให้เลือกแอป NASample และ Just once เท่านั้น (อย่าเลือก Always)

หน้าจอ Pop-up ของ In-App Permission ในกรณีไม่ได้ตอบOut-App (อย่าเลือก Always open...)

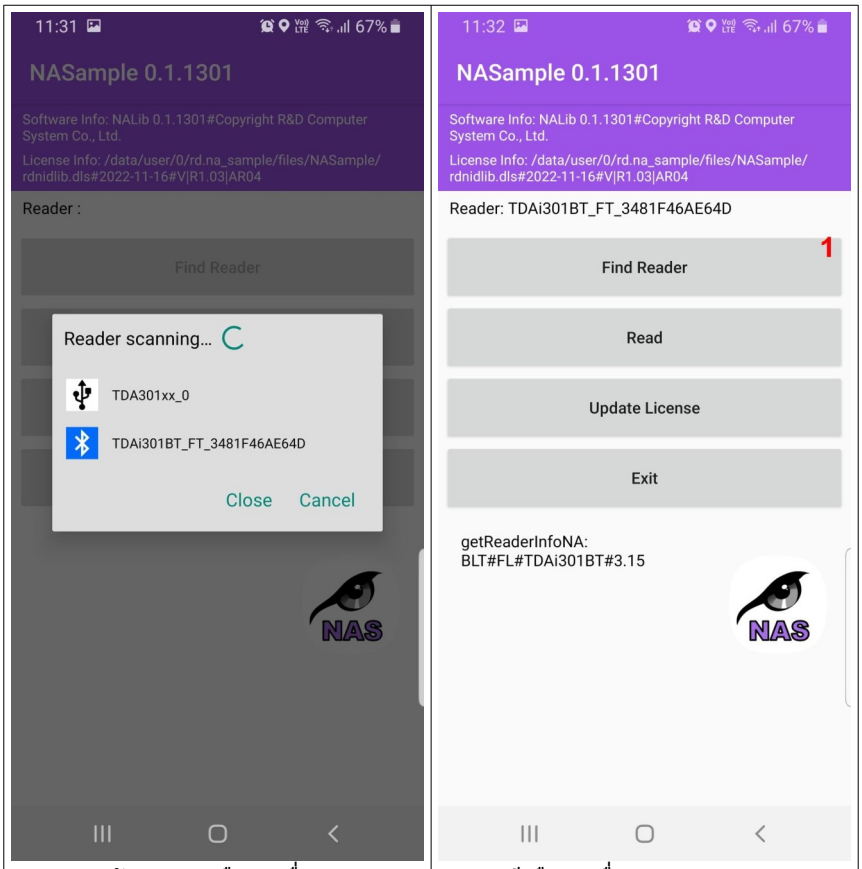
- หากจะทดสอบกับเครื่องอ่านบลูทูธแบบ BLE (เช่น TDA301BL, TDA301BLM, TDA3901BE) ให้กดปุ่มเปิดที่เครื่องอ่านรอจนไฟสีน้ำเงินกระพริบ แล้วเปิดระบบสัญญาณบลูทูธของโทรศัพท์มือถือ **แต่ไม่ต้องจับคู่** (ไม่ต้อง Pair)
- หากจะทดสอบกับเครื่องอ่านแบบบลูทูธ 3 (TDA301BT) ให้กดปุ่มเปิดที่เครื่องอ่านรอจนไฟสีน้ำเงินหรือสีเขียวกระพริบ แล้วเปิดระบบสัญญาณบลูทูธของโทรศัพท์มือถือ พร้อมกับจับคู่ (Pair) กับเครื่องอ่าน โดยการเข้าไปที่ **Settings (ตั้งค่า)** ของระบบแอนดรอยด์ แตะที่ **Bluetooth (บลูทูธ)** > **Scan (ค้นหา)** แล้วดูที่บรรทัดล่างสุดของหน้าจอ เลือกเครื่องอ่านที่ขึ้นต้นด้วย “FT_” แล้วตอบตกลง

8.4.3 การค้นหาเครื่องอ่าน

- NASample จะมีปุ่มสำหรับให้ผู้ใช้ค้นหา หรือสั่งเปลี่ยนเครื่องอ่าน คือปุ่ม Find Reader
- เมื่อผู้ใช้แตะที่ปุ่ม Find Reader จะเป็นการเรียกใช้คำสั่ง **`getReaderListNA(listOption)`** เพื่อขอรายชื่อเครื่องอ่านบัตรสมาร์ทคาร์ดแบบ USB ที่เสียบอยู่ และเครื่องอ่านแบบบลูทูธที่จับคู่ (Paired) กับอุปกรณ์แอนดรอยด์ไว้แล้วมาแสดง โดยสามารถกำหนดวิธีในการค้นหาชื่อเครื่องอ่านได้จาก **`listOption`** ปกติแนะนำให้ใช้ค่าเป็น 0x93, 0x9F หรือ 0x01 แต่ใน NASample จะใช้ค่า 0x9F ซึ่งหมายความว่าโปรแกรมจะ Scan หาเครื่องอ่านทุกชนิด

- ในกรณีที่ผู้ใช้ไม่ต้องการใช้เครื่องอ่านแบบบลูทูธ และได้ปฏิเสธการขอ Location Permission ไว้ โปรแกรม NASample จะใช้ค่า 0x81 ในการทำงานแทน เพื่อตัดการรอเวลาค้นหาเครื่องอ่านแบบบลูทูธออกไป

- รับรายชื่อเครื่องอ่านผ่าน **`onGetReaderListNA`**
- เรียก **`selectReaderNA(reader)`** โดยใช้ชื่อเครื่องอ่านเครื่องแรกที่ได้จากคำสั่ง **`getReaderListNA`** (หรือจะทำ Drop-down ให้ผู้ใช้เลือกเครื่องอ่านอีกทีก็ได้ แต่ NASample ไม่ได้ใช้วิธีนี้)



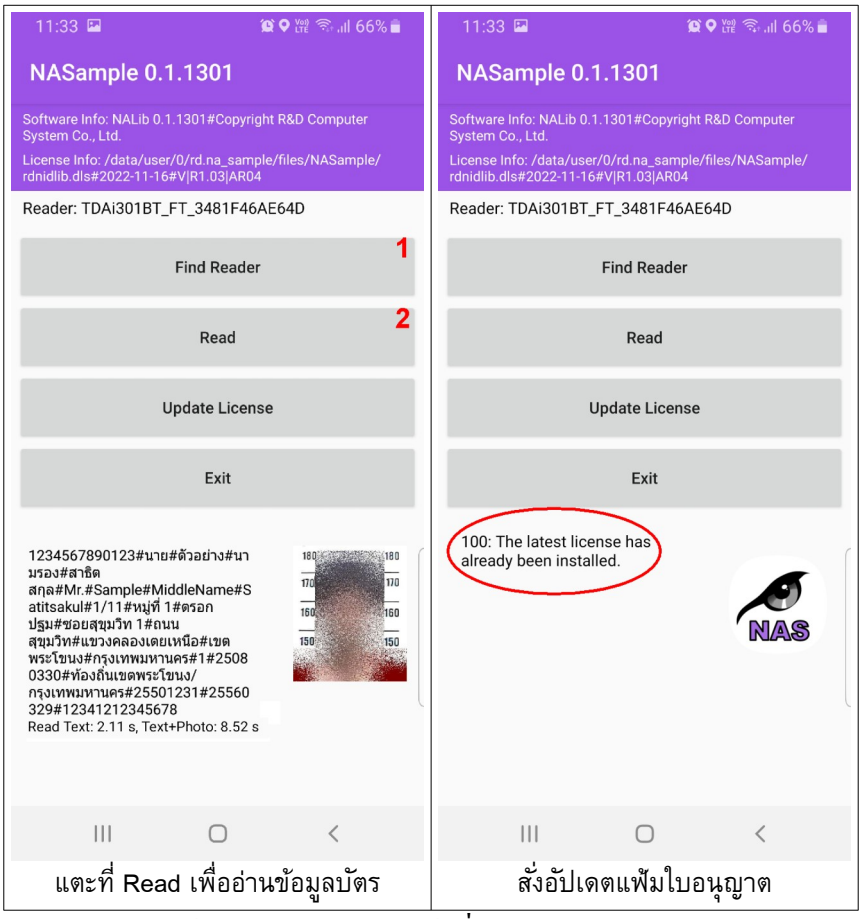
การค้นหาและเลือกเครื่องอ่าน | กรณีเลือกเครื่องอ่าน TDAi310BT

- ตรวจสอบค่า Return Code ผ่าน **`onSelectReaderNA`** หากเป็น 0 ให้ทำงานคำสั่งต่อไปได้ หากเป็นค่า -2 หรือ -12 ให้แจ้งปัญหาให้ผู้ใช้ทราบ หรืออาจทำกระบวนการอัปเดตแฟ้มใบอนุญาตก่อนแล้วค่อยกลับมาเรียก **`selectReaderNA`** อีกครั้ง
- หากต้องการ สามารถเรียกใช้คำสั่ง **`getReaderInfoNA`** เพื่อดูข้อมูลของเครื่องอ่านที่ Select ได้สำเร็จ

8.4.4 เมื่อจะอ่านข้อมูลจากบัตรประชาชน

- เมื่อผู้ใช้สั่งอ่านบัตรประชาชน ให้เรียก **`connectCardNA`** เพื่อจ่ายไฟให้กับบัตรประชาชน และเริ่มเชื่อมต่อกับบัตร โดยหากเชื่อมต่อได้ จะได้ Return Code เป็น 0 หากได้ค่าอื่น เช่น -16 (ยังไม่ได้เสียบบัตรเข้ากับเครื่องอ่าน) ก็ควรแจ้งให้ผู้ใช้ทราบถึงปัญหาต่าง ๆ เหล่านั้น

- เมื่อเชื่อมต่อกับบัตรได้เรียบร้อยแล้ว ให้สั่งอ่านข้อมูลจากบัตรได้เลย โดยใช้คำสั่ง **`getNIDNumberNA`** เพื่ออ่านเฉพาะเลขประจำตัวประชาชน หรือสั่ง **`getNIDTextNA`** เพื่ออ่านข้อมูลตัวอักษรทั้งหมด หรือสั่ง **`getNIDPhotoNA`** เพื่ออ่านรูปภาพใบหน้าของเจ้าของบัตร
- เมื่ออ่านข้อมูลได้ครบแล้วให้หยุดการเชื่อมต่อกับบัตรโดยใช้คำสั่ง **`disconnectCardNA`**



แตะที่ Read เพื่ออ่านข้อมูลบัตร | สั่งอัปเดตแฟ้มใบอนุญาต

8.4.5 การอัปเดตแฟ้มใบอนุญาต (กรณีใช้เครื่องอ่านแบบ FL: File Licensing)

- ให้สั่ง **`updateLicenseFileNA`** เพื่อให้ NALib ดาวน์โหลดแฟ้มใบอนุญาตรุ่นล่าสุดจากอินเทอร์เน็ตมาเก็บไว้เป็นแฟ้มตามชื่อที่ระบุไว้ในฟังก์ชัน **`openLibNA`** หลังทำงานเสร็จให้ตรวจสอบ Return Code จาก **`onUpdateLicenseFileNA`** หากได้ค่าเป็น 0, 1, 2 หรือ 3 แสดงว่า NALib ได้อัปเดตแฟ้มใบอนุญาตเป็นแฟ้มใหม่แล้ว หากได้ค่าเป็น 100, 101, 102 หรือ 103 แสดงว่าแฟ้มใบอนุญาตที่มีอยู่ในเครื่องเป็นรุ่นล่าสุดอยู่แล้ว หากมีค่าเป็น -15 แสดงว่าไม่สามารถดาวน์โหลดได้เนื่องจากไม่สามารถเข้าใช้งานอินเทอร์เน็ตได้ หรือหากเป็น -18 แสดงว่าอัปเดตไม่ได้เนื่องจากสาเหตุอื่น ทั้งนี้ในกรณีของ -15 และ -18 อาจต้องร้องขอให้ผู้ใช้เปิดการใช้งานอินเทอร์เน็ตของโทรศัพท์มือถือแล้วลองสั่งอัปเดตอีกครั้ง

8.4.6 การเปลี่ยนเครื่องอ่าน

- ผู้ใช้สามารถเปลี่ยนเครื่องอ่านได้ 2 วิธี คือ เปลี่ยนโดยสั่ง getReaderList ใหม่ หรือใช้รายชื่อเดิมที่เคย getReaderList ไว้แล้วก็ได้ เช่นทำเป็น Drop-down ให้เลือก

- สำหรับ NASample จะใช้วิธี getReaderList ใหม่ ผ่านปุ่ม Find Reader เพื่อ Scan และหาเครื่องอ่านใหม่

8.4.7 เมื่อจะปิดแอปพลิเคชัน

- ให้สั่ง **`deselectReaderNA`** เพื่อหยุดการใช้งานเครื่องอ่านก่อน
- สั่ง **`closeLibNA`** เพื่อหยุดการใช้งาน NALib และคืนหน่วยความจำให้กับระบบ
- ปิดแอปพลิเคชัน

9. การจัดการแฟ้มใบอนุญาต (DLS) สำหรับเครื่องอ่านแบบ File Licensing

แฟ้มใบอนุญาต หรือ License File เป็นแฟ้มที่มีความสำคัญมากในการใช้งาน NALib SDK กับเครื่องอ่านแบบ File Licensing (FL)

สำหรับเครื่องอ่านแบบ Virtual Licensing (VL) จะไม่มีการใช้งานแฟ้มใบอนุญาต ดังนั้นจึงไม่ต้องสนใจและสามารถข้ามเนื้อหาในบทนี้ไปได้ทั้งหมด

แฟ้มใบอนุญาตจะเป็นที่รวบรวมรายการของเครื่องอ่านที่ได้ลงทะเบียนให้ใช้งานกับ NALib SDK ไว้ ดังนั้นแม้เครื่องอ่านที่มีรูปร่างเหมือนกัน ซื้อจากบริษัท อาร์แอนด์ดี คอมพิวเตอร์ ซิสเท็ม มาเหมือนกัน แต่หากเป็นเครื่องเปล่าที่ยังไม่ได้ลงทะเบียน ก็จะใช้งานกับ NALib SDK ไม่ได้ เครื่องอ่านที่ลงทะเบียนแล้ว จะขายอยู่ในชุดสินค้าที่มีชื่อรุ่นขึ้นต้นด้วย TDA, TDAi หรือ TDC เท่านั้น

- แฟ้มใบอนุญาตของ NALib SDK จะมีชื่อเป็น rdnidlib.dls เสมอ
- ในแอปพลิเคชันตัวอย่าง NADemo เราสามารถดูข้อมูลของแฟ้มใบอนุญาตได้จากคำสั่ง **`getLicenseInfoNA`** ผ่านปุ่ม **License Info** ซึ่งจะแสดงชื่อโฟลเดอร์ ชื่อแฟ้ม และวันที่ของแฟ้ม
- เราสามารถดูข้อมูลของแฟ้มใบอนุญาตในแอปพลิเคชัน NASample ได้ด้วยเช่นเดียวกัน
- ภายในแฟ้มใบอนุญาตจะมีรายการของหมายเลขเครื่อง (S/N) ของเครื่องอ่านที่ลงทะเบียนไว้แล้ว
- ในการทำงานของ NALib จะมีการตรวจสอบ S/N ของเครื่องอ่านว่ามีลงทะเบียนอยู่ในแฟ้มใบอนุญาตหรือไม่ หากมีก็จะสามารถใช้งานเครื่องอ่านนั้นได้ หากไม่มีก็จะใช้ไม่ได้
- ต้นฉบับแฟ้มใบอนุญาตจะมีอยู่บนเซิร์ฟเวอร์ที่สามารถติดต่อได้ผ่านทางอินเทอร์เน็ต ตัวแฟ้มจะมีการปรับปรุงเพิ่มเติม S/N ของเครื่องอ่านโดยการสะสมหมายเลข S/N เพิ่มขึ้นเรื่อย ๆ ตามจำนวนการผลิต แฟ้มรุ่นใหม่กว่าจะมีจำนวนเครื่องอ่านที่มากกว่า และใช้งานแทนแฟ้มรุ่นเก่าได้เสมอ
- ในครั้งแรกที่แอปพลิเคชันเป้าหมาย (Target App) ทำงาน NALib จะมีการสร้างแฟ้ม rdnidlib.dls นี้ไว้ในโฟลเดอร์ของแอปพลิเคชันเป้าหมายให้ก่อน
- เมื่อแอปพลิเคชันพบการเชื่อมต่อกับเครื่องอ่านที่ยังไม่ได้ลงทะเบียนจากคำสั่ง **`selectReaderNA`** (Return Code -2) ตัว NALib จะมีการอัปเดตแฟ้มใบอนุญาตให้แบบอัตโนมัติ แต่จะยกเว้นกรณีที่อัปเดตไม่ได้ เช่น โทรศัพท์มือถือไม่ได้เปิดใช้งานอินเทอร์เน็ต ผู้พัฒนาจะต้องแจ้งให้ผู้ใช้เปิดการใช้งานอินเทอร์เน็ตและสั่งอัปเดตอีกครั้งด้วยคำสั่ง **`updateLicenseFileNA`** เอง
- หากสั่งอัปเดตแล้วยังใช้ไม่ได้ แสดงว่าเครื่องอ่านนั้นเป็นเครื่องอ่านที่ยังไม่ได้ลงทะเบียนจริง ๆ ให้แจ้งผู้ใช้ถอดเปลี่ยนเครื่องอ่านเป็นเครื่องใหม่ที่ลงทะเบียนแล้วแทน
- ในการพัฒนาแอปพลิเคชันสำหรับเครื่องอ่านแบบ FL ผู้พัฒนาควรเขียนโปรแกรมให้มีการโหลดและอัปเดตแฟ้มใบอนุญาตได้หลาย ๆ วิธี เพื่อความสะดวกในการใช้งานของผู้ใช้งาน ได้แก่
  - ให้มีการสั่งโหลดหรืออัปเดตแฟ้มใบอนุญาตทันทีหลังจากติดตั้งแอปพลิเคชันในครั้งแรก
  - ให้มีการสั่งอัปเดตแฟ้มใบอนุญาตทุกครั้งที่พบ Return Code -2 (เครื่องอ่านนี้ใช้ไม่ได้เพราะใบอนุญาตไม่ถูกต้อง) หรือ Return Code -12 (ไม่พบแฟ้มใบอนุญาตหรือแฟ้มใบอนุญาตเสียหาย)
  - ให้ทำปุ่ม “Update License” ไว้ที่ในหน้าจอใดหน้าจอหนึ่งของแอปพลิเคชัน เพื่อให้ผู้ใช้สามารถสั่งอัปเดตได้ด้วยตัวเอง เผื่อให้ End Users ใช้แก้ไขปัญหาแฟ้มใบอนุญาตที่ไม่ยอมอัปเดตหรือแฟ้มสูญหายได้เอง
- ในการสั่งอัปเดตแฟ้มใบอนุญาต จะต้องมีการใช้งานอินเทอร์เน็ตด้วย หากสั่งอัปเดตแล้ว พบข้อผิดพลาด Return Code -15 (ไม่สามารถติดต่ออินเตอร์เน็ตได้) หรือ Return Code -18 (อัปเดตแฟ้มใบ

อนุญาตไม่สำเร็จ) อาจต้องร้องขอให้ผู้ใช้เปิดการใช้งานอินเทอร์เน็ตของโทรศัพท์มือถือแล้วลองสั่งอัปเดตอีกครั้ง

10. การพัฒนาโปรแกรมสำหรับเครื่องอ่านแบบ Virtual Licensing

เครื่องอ่านแบบที่มีใบอนุญาตฝังอยู่ในตัวเครื่อง (Virtual Licensing Readers) หรือ VL เป็นเครื่องอ่านรุ่นใหม่ที่ไม่มีการใช้แฟ้มใบอนุญาต หรือ License File อีกต่อไป ซึ่งมีข้อดีคือ

- ไม่ต้องเขียนโปรแกรมเพื่อจัดการแฟ้มใบอนุญาต เช่นไม่ต้องสร้างปุ่ม Update และไม่ต้องเตรียมการดาวน์โหลด
- ไม่มีการใช้งานอินเทอร์เน็ต
- ผู้ใช้ไม่ต้องคอยสั่งอัปเดตแฟ้มใบอนุญาตอีกต่อไป
- สามารถซื้อเครื่องอ่านมาเพิ่มในระบบได้ทันที ไม่ต้องรอการอัปเดตแฟ้มใบอนุญาต
- ไม่ต้องมีการขอ Permissions ในเรื่องของ Internet และ External Storage
- สามารถใช้งานร่วมกับเครื่องอ่านแบบธรรมดา (FL - File Licensing Readers) ในแอปพลิเคชันเดียวกันได้

เครื่องอ่านรุ่นที่เป็น VL มีดังนี้

- TDA301VC, TDAi301VC
- TDA39VC, TDA39VC2, TDA39VM
- TDA301BL, TDAi301BL
- TDA301BLM, TDAi301BLM
- TDC39VC, TDC301VC

แนวทางในการพัฒนาแอปพลิเคชันเพื่อใช้เครื่องอ่านแบบ VL มีดังนี้

- ควรใช้ไลบรารี NALib.aar รุ่น 0.1.1600 ขึ้นไปเท่านั้น
- ในขั้นตอนการเริ่ม **`openLibNA(licenseFile)`** ให้ส่งค่า **`licenseFile`** เป็นค่าสตริงที่ความยาวเป็น 0 (**`""`**) หรือหากต้องการใช้งานร่วมกับเครื่องอ่านแบบปกติ (File Licensing Readers) ก็สามารถส่งเป็นชื่อของแฟ้มใบอนุญาตตามปกติก็ได้
- ไม่ต้องสร้างปุ่ม Update License และไม่ต้องใช้คำสั่ง **`updateLicenseFileNA`** อีกต่อไป ยกเว้นจะใช้งานร่วมกับเครื่องอ่านแบบ FL
- ไม่ต้องขอ Permissions ในเรื่องของ Internet และ Storage ยกเว้นจะใช้งานร่วมกับเครื่องอ่านแบบ FL

11. การขอ Permissions

NALib อาจต้องขอสิทธิ์ใช้งาน Permissions ต่อไปนี้ในการทำงาน

- **กลุ่มที่ 1 การขออนุญาตใช้งาน Bluetooth**
  - หมายเหตุ หากไม่ได้ใช้งานเครื่องอ่านแบบบลูทูธในแอปพลิเคชันเลย คือจะใช้งานเครื่องอ่านแบบ USB เพียงอย่างเดียวเท่านั้น ก็ไม่จำเป็นต้องขอ Permissions ในกลุ่มนี้
  - Permissions ในกลุ่มนี้ประกอบด้วย
    - Permission ทั่วไป
      - (1) **`android.permission.BLUETOOTH`**
      - (2) **`android.permission.BLUETOOTH_ADMIN`**
    - Android 11 และต่ำกว่า
      - (3) **`android.permission.ACCESS_FINE_LOCATION`** (เป็นแบบ Runtime)
    - Android 12 หรือสูงกว่า
      - (4) **`android.permission.BLUETOOTH_SCAN`** (เป็นแบบ Runtime)
      - (5) **`android.permission.BLUETOOTH_CONNECT`** (เป็นแบบ Runtime)
- **กลุ่มที่ 2 การขออนุญาตใช้งาน Internet**
  - หมายเหตุ หากไม่ได้เขียนโปรแกรมให้มีการใช้งานคำสั่ง **`updateLicenseFileNA`** เพื่ออัปเดตแฟ้มใบอนุญาต ก็ไม่จำเป็นต้องขอ Permissions ในกลุ่มนี้
  - Permissions ในกลุ่มนี้ประกอบด้วย
    - (6) **`android.permission.INTERNET`**
    - (7) **`android.permission.ACCESS_NETWORK_STATE`**

- **กลุ่มที่ 3 การขออนุญาตใช้งาน External Storage เพื่อใช้อ่านหรือเขียนข้อมูลในแฟ้มใบอนุญาต**
  - แบ่งเป็น 3 กรณี
    - กรณีที่ 1 ใช้งานเครื่องอ่านแบบ VL อย่างเดียว (ไม่มีการใช้เครื่องอ่านแบบ FL ในแอปพลิเคชันเลย ซึ่งแปลว่าจะไม่มีการใช้งานแฟ้มใบอนุญาตและไม่ต้องการที่เก็บแฟ้มใด ๆ) ไม่ต้องขอ Permission ในกลุ่มนี้
    - กรณีที่ 2 มีการใช้งานเครื่องอ่านแบบ FL ด้วย ทำให้ต้องเตรียมพื้นที่เก็บแฟ้มใบอนุญาต แต่เลือกใช้พื้นที่ Private Storage ของแอปพลิเคชันเอง ก็ไม่ต้องขอ Permission ในกลุ่มนี้ (แอปพลิเคชันตัวอย่าง NASample และ NADemo เลือกใช้วิธีนี้)
    - กรณีที่3 กรณีต้องการเก็บแฟ้มใบอนุญาตในพื้นที่สาธารณะ (External Storage) จะต้องมีการขอ Permission ในกลุ่มนี้
  - Permissions ในกลุ่มนี้ประกอบด้วย
    - สำหรับ Android 10 และต่ำกว่า
      - (8) **`android.permission.READ_EXTERNAL_STORAGE`** (เป็นแบบ Runtime)

(9) **`android.permission.WRITE_EXTERNAL_STORAGE`** (เป็นแบบ Runtime)

- สำหรับ Android 11 และสูงกว่า

(10) **`android.permission.MANAGE_EXTERNAL_STORAGE`** (เป็นแบบ Runtime)

วิธีการขอ Permissions

- ให้ขอ Permissions ทั้งหมดที่ใช้ โดยการระบุไว้ในไฟล์ AndroidManifest.xml
- สำหรับ (3), (4), (5), (8) และ (9) ซึ่งเป็น Permissions แบบ Runtime ผู้พัฒนาจะต้องเขียนโปรแกรมเพิ่ม ให้มีการตรวจสอบสิทธิ์และร้องขอความยินยอมจากผู้ใช้ในจังหวะที่จะใช้งาน (Runtime) ด้วย
  - เนื่องจาก Permission (8) และ (9) อยู่ใน Group เดียวกัน ดังนั้นในการขอ Permission ขณะ Runtime ให้ขอเพียง (8) หรือ (9) ตัวใดตัวหนึ่งก็พอ ระบบจะอนุญาตให้พร้อมกันทั้ง 2 ตัวเอง
  - ดังนั้นในขณะ Runtime ให้ขอเฉพาะ (3) , (4), (5) และ (8) ก็พอ

ขอให้ดูตัวอย่างไฟล์ AndroidManifest.xml และตัวอย่างการเขียนโปรแกรมได้จากตัวอย่างใน Source Code ของ NASample หรือ NADemo

สำหรับเครื่องอ่านแบบ Virtual Licensing จะไม่มีการใช้งานแฟ้มใบอนุญาต ดังนั้นไม่จำเป็นต้องขออนุญาต Permission กลุ่มที่ 2 (Internet) และกลุ่มที่ 3 (External Storage) และหากเป็นเครื่องอ่านแบบ USB เพียงแบบเดียว ก็ไม่ต้องมีการขอ Permission ใด ๆ เลย

สำหรับการขอ Permission ในกลุ่มที่ 3 นี้ จะขอก็ต่อเมื่อ License File ที่ระบุในตอน **Open** อยู่ใน External Storage หากตำแหน่งไฟล์ที่ระบุไม่ได้อยู่ใน External Storage และไม่มีการเขียนไฟล์อื่นใดใน External Storage อีกก็ไม่จำเป็นต้องขอ Permission ในกลุ่มนี้

- หากต้องการระบุตำแหน่งของ License File ใน External Storage
  - ใน Android ที่ต่ำกว่าหรือเท่ากับ 10 จะต้องขอ Permission (8) และ (9) และต้องใส่ **`android:requestLegacyExternalStorage="true"`** ในแท็ก **Application** ด้วย
  - ใน Android 11 หรือสูงกว่า จะใช้ Permission (10) และต้องเขียนวิธีขอ Permission Runtime เอง เนื่องจาก Android API ณ ปัจจุบัน ยังไม่มี Permission Runtime ให้ใช้
- แนะนำให้บันทึก License File ไว้ใน Internal Storage จะดีที่สุด เพื่อลดความยุ่งยากในส่วนนี้ (จะได้ไม่ต้องขอ Permission ในกลุ่มที่ 3)

12. ความสามารถในการเชื่อมต่อกับเครื่องอ่านของ NALib

- NALib รองรับการใช้งานได้กับเครื่องอ่านทั้งแบบ USB และบลูทูธ แต่สามารถใช้งานกับเครื่องอ่านในขณะใดขณะหนึ่งได้เพียงเครื่องเดียวเท่านั้น (ผ่านคำสั่ง **`selectReaderNA`**)
- ไม่ควรเสียบเครื่องอ่านบัตรแบบ USB ที่โทรศัพท์มือถือมากกว่า 1 เครื่อง (เช่นต่อผ่าน USB Hub) เพราะ NALib อาจทำงานผิดพลาดได้
- เครื่องอ่านแบบบลูทูธธรรมดา (Bluetooth 3.0) ที่จะใช้งานกับ NALib จะต้องผ่านการสั่งจับคู่ (Pair) กับโทรศัพท์มือถือแล้วเท่านั้น หากยังไม่ได้จับคู่ NALib จะ Scan ไม่พบ
- เครื่องอ่านบลูทูธแบบ BLE ที่จะใช้งานกับ NALib **ไม่ต้องสั่งจับคู่** (Pair) สามารถใช้งานได้ทันที

13. หน้าต่าง Pop-up สำหรับการขออนุญาตใช้เครื่องอ่าน USB ภายในและภายนอกแอปพลิเคชัน

- ปกติการใช้งานเครื่องอ่านแบบ USB กับระบบแอนดรอยด์ ผู้ใช้จะต้องยินยอมหรือให้อนุญาตก่อนแอปพลิเคชันจึงจะใช้งานเครื่องอ่านนั้นได้ ซึ่งการขออนุญาตจะแสดงเป็นหน้าต่าง Pop-up ขึ้นมาถาม และสามารถทำให้แสดงได้ใน 2 จุด คือ
  - ให้ Pop-up ถามทันทีที่เสียบเครื่องอ่าน USB ไม่ว่าแอปพลิเคชันกำลังทำงานอยู่หรือไม่ก็ตาม เรียกว่า USB Reader Out-App Permission
  - ให้ Pop-up ถามเมื่อกำลังใช้งานแอปพลิเคชันอยู่ เรียกว่า USB Reader In-App Permission
- NALib รองรับการใช้งานได้ทั้ง 2 วิธี จะเลือกวิธีใดวิธีหนึ่ง หรือจะใช้ทั้ง 2 วิธีคู่กันก็ได้ (แต่ต้องมีอย่างน้อย 1 วิธี)
- การใช้ Out-App อย่างเดียว
  - ข้อดี
    - ผู้ใช้จะรู้ทันทีที่เสียบเครื่องอ่าน USB ว่าเครื่องอ่านยังใช้งานได้ และเสียบได้ถูกต้อง
  - ข้อเสีย
    - หากมีแอปพลิเคชันหลายตัวที่ใช้งานเครื่องอ่าน USB ได้ด้วย หากเลือกแอปพลิเคชันผิด และมีการให้จำแอปพลิเคชันไว้ เช่นเผลอตอบว่า **Always (ทุกครั้ง)** จะทำให้ไม่มีการแสดงหน้าต่าง Pop-up อีกเลยจนกว่าผู้ใช้จะไปล้างค่า Default ของแอปพลิเคชันที่เลือกผิด
    - หากต้องการเปลี่ยนหรือสลับการใช้งานหลายแอปพลิเคชันกับเครื่องอ่านเดียวกัน จะต้องถอดเครื่องอ่านออกแล้วเสียบใหม่เพื่อเลือกแอปใหม่ในหน้าต่าง Pop-up
- การใช้ In-App อย่างเดียว
  - ข้อดี
    - จะมีหน้าต่าง Pop-up ภายในแอปพลิเคชันของเราเสมอ แม้จะเผลอตอบว่า **Always (ทุกครั้ง)** ให้กับแอปพลิเคชันอื่นไปแล้ว
    - สามารถสลับการใช้งานแอปพลิเคชันหลาย ๆ แอปกับเครื่องอ่านเครื่องเดียวกันได้ง่าย ไม่ต้องถอดเสียบเครื่องอ่านใหม่ (กรณีนี้ทุกแอปต้องใช้ In-App เหมือนกันหมด)
  - ข้อเสีย
    - ไม่เหมาะใช้งานกับตู้ Kiosk ที่ไม่มีเจ้าหน้าที่ประจำ เพราะจะไม่มีคนมาตอบโต้กับหน้าต่าง Pop-up
    - หากมีแอปอื่นนที่เลือกใช้วิธี Out-App กับเครื่องอ่านเดียวกัน หลังจากเสียบเครื่องอ่านและมีตอน Pop-up ขึ้นมา จะไม่มีชื่อแอปที่ใช้วิธี In-App ขึ้นมาแสดงด้วย ผู้ใช้จะหาแอปไม่เจอ
- การใช้ทั้ง 2 วิธี Out-App และ In-App คู่กัน
  - ข้อดี
    - เมื่อเสียบเครื่องอ่าน USB จะมีหน้าต่าง Pop-up ขึ้นมาทันที และมีชื่อแอปของเราเสมอ หากเลือกได้ถูกต้อง ก็สามารถใช้งานต่อได้ หากเลือกผิด ก็จะยังมี Pop-up ภายในแอปให้เลือกอีกครั้ง และสามารถสั่งให้จดจำแอปของเราไว้ได้ด้วย
    - สามารถสลับการใช้งานแอปพลิเคชันหลาย ๆ แอปกับเครื่องอ่านเครื่องเดียวกันได้ง่าย ไม่ต้องถอดเสียบเครื่องอ่านใหม่

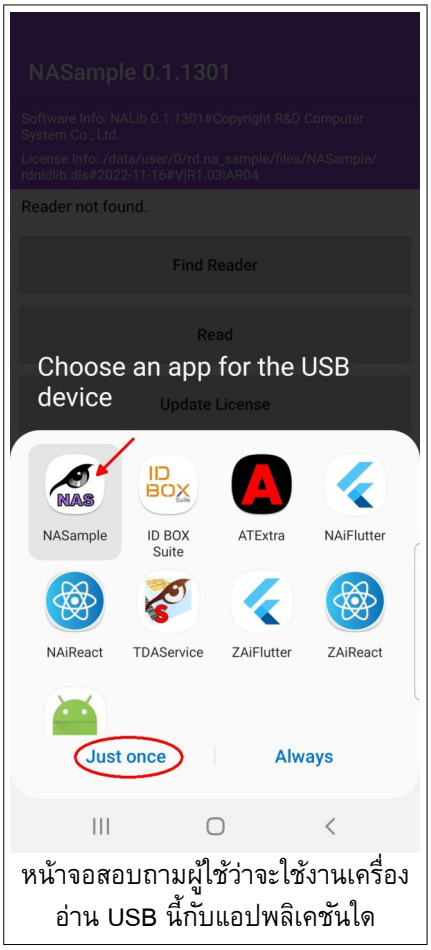
- วิธีการกำหนด
  - การใช้ USB Reader Out-App Permission ผู้พัฒนาจะต้องกำหนด Activity ชื่อ **`rd.nalib.UsbEventReceiverActivity`** ไว้ในไฟล์ AndroidManifest.xml แต่หากไม่ต้องการใช้ Out-App ก็ให้ลบออก ให้ดูตัวอย่างจากไฟล์ในโปรเจ็กต์ NADemo และ NASample
  - การใช้ USB Reader In-App Permission ผู้พัฒนาจะต้องส่งคำสั่ง **`setPermissionsNA(1)`** ไว้ในช่วงแรกของการทำงานของแอปพลิเคชัน ให้ดูตัวอย่างจาก Source Code ของ NASample
- ควรลองใช้ทั้ง Out-App และ In-App เพื่อจับความรู้สึกของผู้ใช้ ว่าควรจะเลือกแบบใดกับตัวอย่างแอปพลิเคชันที่เตรียมไว้ให้
  - NADemo เป็นแอปพลิเคชันตัวอย่างที่ใช้งาน In-App อย่างเดียว
  - NASample เป็นแอปพลิเคชันตัวอย่างที่ใช้ทั้ง Out-App และ In-App
- ปกติแนะนำให้พัฒนาโดยการใช้ทั้ง Out-App และ In-App คู่กันเสมอ

14. คำแนะนำการใช้งานเครื่องอ่านแบบต่าง ๆ (ควรแจ้งให้ผู้ใช้งาน End User ทราบด้วย)

14.1 การใช้งานกับเครื่องอ่านแบบ USB กรณีเลือกใช้งานแบบ Out-App Permission อย่างเดียว

14.1.1 การเลือกแอปพลิเคชันในการใช้งานกับเครื่องอ่านแบบ USB

เมื่อนำเครื่องอ่าน USB มาเสียบเข้ากับพอร์ตของโทรศัพท์ทุกครั้ง ระบบแอนดรอยด์จะมีการตรวจจับอุปกรณ์ USB และจะมีหน้าจอ Pop-up สอบถามผู้ใช้ว่าจะใช้งานอุปกรณ์ USB นี้กับแอปพลิเคชันใด



หน้าจอสอบถามผู้ใช้ว่าจะใช้งานเครื่องอ่าน USB นี้กับแอปพลิเคชันใด

- ให้ผู้ใช้เลือกเป็นชื่อแอปพลิเคชันที่จะใช้งาน เช่นในรูปตัวอย่างให้เลือก NASample แล้วแตะที่ **Just Once (เพียงครั้งเดียว)**
  - โดยปกติแล้วไม่ควรเลือก **Always (ทุกครั้ง)** หรือ **Remember this choice (จำตัวเลือกนี้ไว้)** หากเลือก จะทำให้การเสียบเครื่องอ่านครั้งต่อไป จะไม่มีการถามชื่อแอปพลิเคชันอีก และจะใช้แอปพลิเคชันที่เคยเลือกแล้วไว้ในการทำงานตลอดไป
- หากเลือกแอปพลิเคชันผิดตัวแล้วเผลอแตะที่ **Just Once (เพียงครั้งเดียว)** ให้ถอดเสียบเครื่องอ่านใหม่อีกครั้ง
- หากเลือกแอปพลิเคชันผิดแล้วแตะที่ **Always (ทุกครั้ง)** หรือ **Remember this choice (จำตัวเลือกนี้ไว้)** จะทำให้ไม่สามารถใช้งานเครื่องอ่านนั้นได้ ต้องไปรีเซ็ตการเลือก โดยให้เข้าไปที่ **Setting (ตั้งค่า)** ของระบบแอนดรอยด์ > **Application Manager (จัดการแอปพลิเคชัน)** หรือ **App Info (ข้อมูลแอป)** > เลือกชื่อแอปพลิเคชันที่เลือกผิด > เลื่อนไปที่ **Open by default (เปิดโดยค่าเริ่มต้น)** หรือ **Launch by default (เริ่มโดยค่าพื้นฐาน)**

แตะที่ **CLEAR DEFAULTS (ล้างค่าเริ่มต้น** หรือ **ล้างค่าพื้นฐาน**) แล้วลองถอดเสียบเครื่องอ่านใหม่อีกครั้ง

14.1.2 หากนำเครื่องอ่าน USB มาเสียบเข้ากับพอร์ตของโทรศัพท์เสร็จแล้ว แต่ไม่มีหน้าต่าง Pop-up โผล่ขึ้นมาถามให้เลือกแอปพลิเคชัน และลองใช้งานแอปพลิเคชันแล้วใช้ไม่ได้ อาจเป็นไปได้ 3 ทาง

- ระบบ OTG ของโทรศัพท์อาจถูกปิดอยู่ ให้เข้าหน้า **Setting (ตั้งค่า)** ของแอนดรอยด์ แล้วลองค้นหาเมนูที่เกี่ยวข้องกับการเปิดการใช้งาน OTG (พบปัญหานี้ในเครื่องของ Oppo, realme และ Vivo)
- โทรศัพท์เครื่องนั้นอาจไม่รองรับระบบ OTG ให้เปลี่ยนโทรศัพท์เป็นเครื่องใหม่ที่รองรับระบบ OTG
- เครื่องอ่านอาจถูกล็อคให้ใช้งานกับแอปพลิเคชันอื่นไปแล้ว (เพราะเผลอแตะ **Always** หรือ **Remember this choice** ไปก่อนแล้ว) ให้แก้โดยการ **CLEAR DEFAULTS** ของแอปพลิเคชันตัวที่ล็อคอยู่ตามข้อ 14.1.1

14.1.3 หากใช้งานเครื่องอ่าน USB ไม่ได้ ให้ลองถอดเสียบเครื่องอ่านใหม่ แล้วเลือกชื่อแอปพลิเคชันให้ถูกต้องอีกครั้งหนึ่ง

14.2 การใช้งานกับเครื่องอ่านแบบบลูทูธ 3 (สินค้ารุ่นที่ลงท้ายว่า BT เช่น TDA301BT และ TDAi301BT)

14.2.1 การใช้งานเครื่องอ่านแบบบลูทูธ 3 ที่โทรศัพท์มือถือจะต้องทำดังนี้

- เปิดการใช้งานบลูทูธของเครื่องโทรศัพท์
- เปิดการใช้งาน Location Service ของเครื่องโทรศัพท์ โดยดึงแถบ Quick Settings จากด้านบนลงมา แล้วแตะที่ไอคอน “ตำแหน่ง” หรือ “Location” รูปปักหมุด แบบเดียวกับการเปิดใช้งาน GPS
- จับคู่อุปกรณ์บลูทูธกับเครื่องอ่าน

14.2.2 วิธีการจับคู่เครื่องอ่านแบบบลูทูธ 3

การจะนำเครื่องอ่านแบบบลูทูธ 3 (สินค้า TDA ที่ชื่อรุ่นลงท้ายว่า BT) มาใช้งานกับโทรศัพท์มือถือจะต้องมีการ “จับคู่” (Pair) ก่อนเสมอ การจับคู่ระหว่างเครื่องอ่านบัตรแบบบลูทูธกับโทรศัพท์มือถือให้ทำดังนี้

- หากเครื่องอ่านไม่ได้ใส่แบตเตอรี่ ให้นำแบตเตอรี่มาใส่ให้เรียบร้อยก่อน
- กดปุ่มที่ด้านหน้าเครื่องอ่านบัตร สังเกตจะมีไฟสีน้ำเงินกระพริบ
- ที่โทรศัพท์หรือแท็บเล็ต เข้าเมนู Settings (ตั้งค่า) ของระบบแอนดรอยด์
- แตะที่ **Bluetooth (บลูทูธ)** > **Scan (ค้นหา)** แล้วดูที่บรรทัดล่างสุดของหน้าจอ (ลองเลื่อนหน้าจอไปยังด้านล่างสุด)
- รอสักครู่ จะมีชื่อเครื่องอ่านบัตรขึ้นต้นด้วย “FT_” ปรากฏขึ้น เช่น “FT__xxxxxxxxxxxx” ให้ดูตัวอักษรและตัวเลขที่อยู่ท้ายชื่อ (xxxxxxxx) จะตรงกับตัวอักษรหรือ S/N ที่อยู่ด้านหลังของเครื่องอ่าน ให้แตะที่ชื่อเครื่องอ่านนั้น แล้วตอบตกลง
- หากไม่พบเครื่องอ่าน ให้สั่ง **Scan (ค้นหา)** อีกครั้ง

14.2.3 ควรสั่ง Pair เครื่องอ่านบลูทูธก่อนเรียกใช้งานแอปพลิเคชันในครั้งแรก หากเรียกแอปพลิเคชันไปก่อน แล้วมีการ Pair ภายหลัง จะทำให้แอปพลิเคชันไม่เห็นเครื่องอ่าน และจะไม่ยอมเชื่อมต่อกับเครื่องอ่านนั้น แก้ไขโดยให้ออกจากแอปพลิเคชัน ปิดสัญญาณบลูทูธของอุปกรณ์แอนดรอยด์ แล้วเปิดสัญญาณบลูทูธอีกครั้ง สั่ง Pair ให้เรียบร้อย แล้วค่อยเรียกแอปพลิเคชัน

14.2.4 การใช้งานเครื่องอ่านบลูทูธรุ่น TDA301BT และ TDAi301BT

- การเปิดปิดเครื่องอ่าน
  - การเปิดใช้งาน ให้กดสวิตช์ด้านหน้าเครื่อง 1 ครั้ง สังเกตจะเห็นไฟสีน้ำเงินกระพริบ
  - การปิด ให้กดสวิตช์ด้านหน้าค้างประมาณ 3 วินาที สังเกตไฟจะดับหมด
  - เครื่องจะปิดเองอัตโนมัติเพื่อประหยัดพลังงาน เมื่อ
    - ถูกยกเลิกการเชื่อมต่อ หรือหลุดจากการเชื่อมต่อ (เช่นปิดสัญญาณบลูทูธของโทรศัพท์มือถือ หรือนำเครื่องอ่านออกห่างจากโทรศัพท์มือถือเกินระยะทำงานของสัญญาณบลูทูธ)
  - รอแล้วไม่มีการเชื่อมต่อนานกว่า 3 นาที
- ความหมายของดวงไฟของเครื่องอ่านบลูทูธรุ่น TDA301BT และ TDAi301BT
  - ไฟสีน้ำเงินกระพริบ 1 ครั้ง: รอการเชื่อมต่อ
  - ไฟสีน้ำเงินกระพริบ 2 ครั้ง: เชื่อมต่อสำเร็จ สามารถสั่งอ่านข้อมูลได้
  - ไฟสีน้ำเงินกระพริบ 3 ครั้ง: กำลังตรวจสอบตัวเองหลังจากเปิดเครื่อง และรอการเชื่อมต่อ
  - ไฟสีน้ำเงินติดสว่างค้าง: เชื่อมต่อสำเร็จ สามารถสั่งอ่านข้อมูลได้
  - ไฟสีขาว ติดสว่างหรือกระพริบเมื่อมีการติดต่อกับบัตร
  - ไฟสีเหลือง ติดสว่างเมื่อแบตเตอรี่อ่อน ควรนำเครื่องอ่านไปประจุไฟฟ้าให้เต็ม
  - ไฟสีส้มแดง ติดสว่างเมื่อมีการประจุ (ชาร์จ) ไฟให้กับแบตเตอรี่ และจะดับเมื่อแบตเตอรี่เต็ม
    - หากเสียบสายชาร์จแล้วไฟสีส้มแดง ไม่ติดสว่าง แสดงว่าแบตเตอรี่เต็ม หรือยังเกือบเต็มอยู่ ไม่จำเป็นต้องประจุกระแสไฟอีก (นั่นคือ ถ้าเสียบสายแล้วไฟสีส้มแดงไม่ติด ไม่ได้แปลว่าเครื่องอ่านเสียหรือชาร์จไฟไม่เข้า แต่แปลว่าไม่จำเป็นต้องชาร์จ และเครื่องอ่านยังทำงานดีอยู่ตามปกติ)

14.2.5 ไม่ควรจับคู่ (Pair) เครื่องอ่านบลูทูธเครื่องเดียวกับอุปกรณ์แอนดรอยด์หลาย ๆ เครื่อง เพราะจะทำให้สับสนได้ง่าย และจะมีการแย่งจับคู่จนไม่รู้ว่ากำลังจับคู่กับเครื่องใด

14.2.6 อุปกรณ์แอนดรอยด์หนึ่งเครื่องควรจับคู่กับเครื่องอ่านแบบบลูทูธเพียงเครื่องเดียว เพื่อลดปัญหาการสับสน หากเป็นไปได้ ควรสั่งให้อุปกรณ์แอนดรอยด์ยกเลิกการจับคู่ (Unpair-เลิกจับคู่ หรือ Forget-ไม่จำ) กับเครื่องอ่านบลูทูธที่ไม่ได้ใช้งานแล้ว

14.2.7 ปัญหาเครื่องอ่านบลูทูธปิดเครื่องเอง หรือการเชื่อมต่อบลูทูธหลุด

- อาจเกิดจากแบตเตอรี่อ่อน (สังเกตไฟจะดับทันทีที่สั่งอ่านบัตร) ทดสอบโดยการเสียบสายชาร์จกับอะแดปเตอร์สำหรับชาร์จจะใช้งานได้ทันที (อย่าชาร์จโดยการเสียบสายกับคอมพิวเตอร์หรือโทรศัพท์มือถือ ให้เสียบกับอะแดปเตอร์เท่านั้น)
- อาจเกิดจากอุปกรณ์แอนดรอยด์มีการสั่งปิดสัญญาณบลูทูธเองเมื่อเข้าโหมด Standby หรือเมื่อเข้าโหมดประหยัดพลังงาน หรือเมื่อเข้าโหมดรักษาความปลอดภัย หรือเมื่อมีการกดปุ่มปิดหน้าจอ (ขึ้นอยู่กับอุปกรณ์แอนดรอยด์ แต่ละรุ่นจะไม่เหมือนกัน) การแก้ไข ให้หาเมนูในการตั้งให้ไม่ต้องมีการปิดสัญญาณบลูทูธเมื่อเข้าโหมดต่าง ๆ ข้างต้น หรือยกเลิกการใช้งานโหมดต่าง ๆ ข้างต้น
- เมื่อมีการเชื่อมต่อกับเครื่องอ่านบลูทูธแล้ว แต่ไม่ได้มีการสั่งอ่านเป็นเวลานาน ๆ บางครั้งจะพบว่าการเชื่อมต่อหลุด เนื่องจากมีสัญญาณรบกวน หรือ เครื่องอ่านอยู่ไกลเกินรัศมีทำการของสัญญาณบลูทูธ
- เมื่อการเชื่อมต่อสัญญาณบลูทูธหลุด ไม่ว่าจากกรณีใดก็ตาม เครื่องอ่านบลูทูธจะปิดตัวเอง ดังนั้นผู้ใช้จะต้องกดปุ่มเปิดเครื่องอ่านบลูทูธใหม่เองทุกครั้งที่จะใช้งานต่อ

14.3 การใช้งานกับเครื่องอ่านแบบบลูทูธพลังงานต่ำ (BLE0) ได้แก่สินค้ารุ่นที่ลงท้ายว่า BL หรือ BLM เช่น TDA301BL, TDA301BLM

14.3.1 การใช้งานเครื่องอ่านแบบ BLE ที่โทรศัพท์มือถือจะต้องทำดังนี้

- เปิดการใช้งานบลูทูธของเครื่องโทรศัพท์
- เปิดการใช้งาน Location Service ของเครื่องโทรศัพท์ โดยดึงแถบ Quick Settings จากด้านบนลงมา แล้วแตะที่ไอคอน “ตำแหน่ง” หรือ “Location” รูปปักหมุด แบบเดียวกับการเปิดใช้งาน GPS
- **ไม่ต้อง**จับคู่เครื่องอ่าน BLE กับโทรศัพท์มือถือ แอปพลิเคชันจะแสดงชื่อเครื่องอ่านให้เลือกเองตอนใช้งาน

14.3.2 ความหมายของดวงไฟของเครื่องอ่านบลูทูธ (BLE0) รุ่น TDA301BL, TDA301BLM, TDAi301BL และ TDAi301BLM

- ไฟสีน้ำเงินกระพริบ: รอการเชื่อมต่อ, ไฟสีน้ำเงินติดสว่างค้าง: เชื่อมต่อสำเร็จ สามารถสั่งอ่านข้อมูลได้
- ไฟสีเหลือง ติดสว่างหรือกระพริบเมื่อมีการติดต่อกับบัตร
- ไฟสีแดง(ขวา) ติดสว่างเมื่อมีการประจุ (ชาร์จ) ไฟให้กับแบตเตอรี่ และจะดับเมื่อแบตเตอรี่เต็ม
    - หากเสียบสายชาร์จแล้วไฟสีแดง ไม่ติดสว่าง แสดงว่าแบตเตอรี่เต็ม หรือยังเกือบเต็มอยู่ ไม่จำเป็นต้องประจุกระแสไฟอีก (นั่นคือ ถ้าเสียบสายแล้วไฟสีแดงไม่ติด ไม่ได้แปลว่าเครื่องอ่านเสียหรือชาร์จไฟไม่เข้า แต่แปลว่าไม่จำเป็นต้องชาร์จ และเครื่องอ่านยังทำงานดีอยู่ตามปกติ)

14.3.3 การเปิดปิดเครื่องอ่าน BLE0

- การเปิดใช้งาน ให้กดสวิตช์ด้านล่าง 1 ครั้ง สังเกตจะเห็นไฟสีน้ำเงินกระพริบ

- การปิด ให้กดสวิตช์ด้านล่าง ค้างประมาณ 3 วินาที สังเกตไฟจะดับหมด
- เครื่องจะปิดเองอัตโนมัติเพื่อประหยัดพลังงาน เมื่อ
  - ถูกยกเลิกการเชื่อมต่อ หรือหลุดจากการเชื่อมต่อ (เช่นปิดสัญญาณบลูทูธของโทรศัพท์มือถือ หรือนำเครื่องอ่านออกห่างจากโทรศัพท์มือถือเกินระยะทำงานของสัญญาณบลูทูธ)
  - รอแล้วไม่มีการเชื่อมต่อนานกว่า 3 นาที

14.3.4 ปัญหาเครื่องอ่าน BLE0 ปิดเครื่องเอง หรือการเชื่อมต่อบลูทูธหลุด

- อาจเกิดจากแบตเตอรี่อ่อน (สังเกตไฟจะดับทันทีที่สั่งอ่านบัตร) ทดสอบโดยการเสียบสายชาร์จกับอะแดปเตอร์สำหรับชาร์จจะใช้งานได้ทันที (อย่าชาร์จโดยการเสียบสายกับคอมพิวเตอร์หรือโทรศัพท์มือถือ ให้เสียบกับอะแดปเตอร์เท่านั้น)
- อาจเกิดจากอุปกรณ์แอนดรอยด์มีการสั่งปิดสัญญาณบลูทูธเองเมื่อเข้าโหมด Standby หรือเมื่อเข้าโหมดประหยัดพลังงาน หรือเมื่อเข้าโหมดรักษาความปลอดภัย หรือเมื่อมีการกดปุ่มปิดหน้าจอ (ขึ้นอยู่กับอุปกรณ์แอนดรอยด์ แต่ละรุ่นจะไม่เหมือนกัน) การแก้ไข ให้หาเมนูในการตั้งให้ไม่ต้องมีการปิดสัญญาณบลูทูธเมื่อเข้าโหมดต่าง ๆ ข้างต้น หรือยกเลิกการใช้งานโหมดต่าง ๆ ข้างต้น
- เมื่อมีการเชื่อมต่อกับเครื่องอ่านบลูทูธแล้ว แต่ไม่ได้มีการสั่งอ่านเป็นเวลานาน ๆ บางครั้งจะพบว่าการเชื่อมต่อหลุด เนื่องจากมีสัญญาณรบกวน หรือ เครื่องอ่านอยู่ไกลเกินรัศมีทำการของสัญญาณบลูทูธ
- เมื่อการเชื่อมต่อสัญญาณบลูทูธหลุด ไม่ว่าจากกรณีใดก็ตาม เครื่องอ่านบลูทูธจะปิดตัวเอง ดังนั้นผู้ใช้จะต้องกดปุ่มเปิดเครื่องอ่านบลูทูธใหม่เองทุกครั้งที่จะใช้งานต่อ

14.4 การใช้งานกับเครื่องอ่านแบบบลูทูธพลังงานต่ำ TDA3901BE (BLE1)

14.4.1 การใช้งานเครื่องอ่านแบบ BLE1 ที่โทรศัพท์มือถือจะต้องทำดังนี้

- เปิดการใช้งานบลูทูธของเครื่องโทรศัพท์
- เปิดการใช้งาน Location Service ของเครื่องโทรศัพท์ โดยดึงแถบ Quick Settings จากด้านบนลงมา แล้วแตะที่ไอคอน “ตำแหน่ง” หรือ “Location” รูปปักหมุด แบบเดียวกับการเปิดใช้งาน GPS
- **ไม่ต้อง**จับคู่เครื่องอ่าน BLE กับโทรศัพท์มือถือ แอปพลิเคชันจะแสดงชื่อเครื่องอ่านให้เลือกเองตอนใช้งาน

14.4.2 ความหมายของดวงไฟของเครื่องอ่านบลูทูธ (BLE1) รุ่น TDA3901BE

- ไฟสีน้ำเงิน (ดวงซ้าย) เป็นไฟของโหมดบลูทูธ
  - กระพริบ 2 ครั้ง: รอการเชื่อมต่อ
  - กระพริบช้า ๆ : เชื่อมต่ออยู่ รอเสียบบัตร
  - ติดสว่างค้าง: เชื่อมต่ออยู่, เสียบบัตรและเชื่อมต่อกับบัตรแล้ว
  - กระพริบถี่: กำลังอ่านข้อมูล
  - ดับ: พักเครื่อง (Sleep) เพื่อประหยัดพลังงาน

- ไฟสีแดง (ดวงกลาง)
  - ติดสว่าง: มีการอัดประจุ (ชาร์จ) ไฟให้กับแบตเตอรี่
  - ดับ: แบตเตอรี่เต็ม
  - กระพริบช้า ๆ : แบตเตอรี่อ่อน
- ไฟสีเขียว (ดวงขวา) เป็นไฟของโหมด USB
  - กระพริบช้า ๆ : เชื่อมต่อกับเครื่องคอมพิวเตอร์ทางพอร์ต USB
  - ติดสว่างค้าง: เชื่อมต่ออยู่, เสียบบัตรและเชื่อมต่อกับบัตรแล้ว
  - กระพริบถี่: กำลังอ่านข้อมูล

14.4.3 การเปิดปิดเครื่องอ่าน (BLE1) รุ่น TDA3901BE

- การเปิดใช้งาน ให้เลื่อนสวิตช์ขึ้นด้านบน เป็นสัญลักษณ์เครื่องหมายบลูทูธ สังเกตจะเห็นไฟสีน้ำเงินกระพริบ
- การปิด ให้เลื่อนสวิตช์ไว้ตรงกลาง เป็นสัญลักษณ์ O สังเกตไฟจะดับหมด

14.4.4 โหมดหยุดพัก (Sleep Mode)

- เครื่องจะพักตัวเองอัตโนมัติเพื่อประหยัดพลังงาน เมื่อไม่ได้มีการใช้งานเครื่องนานกว่าเวลาที่ตั้งไว้ (ปกติคือ 1 นาที) สังเกตได้จากไฟสีน้ำเงินจะดับ
- สามารถออกจากโหมดหยุดพักได้หลายวิธี เช่น
  - ถอดหรือเสียบบัตร
  - สั่งให้แอปพลิเคชันอ่านข้อมูลจากบัตร (กรณีได้มีการเชื่อมต่อกับเครื่องอ่านไว้แล้ว)
  - โยกสวิตช์ปิดเปิดเครื่องอ่านใหม่

14.4.5 ปัญหาเครื่องอ่าน BLE ปิดเครื่องเอง หรือการเชื่อมต่อบลูทูธหลุด

- อาจเกิดจากแบตเตอรี่อ่อน (สังเกตไฟจะดับทันทีที่สั่งอ่านบัตร) ทดสอบโดยการเสียบสายชาร์จกับอะแดปเตอร์สำหรับชาร์จจะใช้งานได้ทันที (อย่าชาร์จโดยการเสียบสายกับคอมพิวเตอร์หรือโทรศัพท์มือถือ ให้เสียบกับอะแดปเตอร์เท่านั้น)
- อาจเกิดจากอุปกรณ์แอนดรอยด์มีการสั่งปิดสัญญาณบลูทูธเองเมื่อเข้าโหมด Standby หรือเมื่อเข้าโหมดประหยัดพลังงาน หรือเมื่อเข้าโหมดรักษาความปลอดภัย หรือเมื่อมีการกดปุ่มปิดหน้าจอ (ขึ้นอยู่กับอุปกรณ์แอนดรอยด์ แต่ละรุ่นจะไม่เหมือนกัน) การแก้ไข ให้หาเมนูในการตั้งให้ไม่ต้องมีการปิดสัญญาณบลูทูธเมื่อเข้าโหมดต่าง ๆ ข้างต้น หรือยกเลิกการใช้งานโหมดต่าง ๆ ข้างต้น
- เมื่อมีการเชื่อมต่อกับเครื่องอ่านบลูทูธแล้ว แต่ไม่ได้มีการสั่งอ่านเป็นเวลานาน ๆ บางครั้งจะพบว่าการเชื่อมต่อหลุด เนื่องจากมีสัญญาณรบกวน หรือ เครื่องอ่านอยู่ไกลเกินรัศมีทำการของสัญญาณบลูทูธ
- เมื่อไม่ได้มีการใช้งานเครื่องอ่านนาน ๆ เครื่องจะหยุดพักตัวเอง (เข้า Sleep Mode) ให้ลองเสียบบัตรและสั่งอ่านข้อมูล เครื่องจะตื่นมาทำงานต่อได้เอง

15. รายการปรับปรุงของ NALib SDK

R260520 (NALib 0.1.1614)

1. เพิ่มการรองรับเครื่องอ่านรุ่น TDC39VC และ TDC301VC
2. เพิ่มความสามารถในการอ่านข้อมูลจากบัตรธนาคาร (ฟังก์ชัน getEmvTextCA)

R250924 (NALib 0.1.1600)

1. ปรับปรุงไลบรารี และตัวอย่างแอปพลิเคชันให้สามารถกำหนด Target API Level เป็น 36 (Android 16) ได้
2. เปลี่ยนไลบรารีภายในเป็นแบบ 16 K Page เพื่อเพิ่มประสิทธิภาพ และเป็นไปตามข้อกำหนด
3. แก้ไขจุดบกพร่องต่าง ๆ ของไลบรารี

R250807 (NALib 0.1.1504)

1. เพิ่มการรองรับเครื่องอ่านรุ่น TDA39VC2
2. เพิ่มความเร็วในการอ่านสำหรับเครื่องอ่านรุ่น TDA301VC (Firmware 2.25 ขึ้นไป)
3. ปรับปรุงไลบรารี และตัวอย่างแอปพลิเคชันให้สามารถกำหนด Target API Level เป็น 35 (Android 15) ได้
4. ปรับปรุงการอ่านบัตรประชาชนรุ่น PM0

R2401108 (NALib 0.1.1414)

1. แก้ไขจุดบกพร่องต่าง ๆ ของไลบรารี
2. ยกเลิกการรองรับเครื่องอ่านบลูทูธ รุ่น TDA01BT

R240404 (NALib 0.1.1401)

1. ปรับปรุงไลบรารี และตัวอย่างแอปพลิเคชันให้สามารถกำหนด Target API Level เป็น 34 (Android 14) ได้
2. แก้ไขจุดบกพร่องต่าง ๆ ของไลบรารี

R231011 (NALib 0.1.1306)

1. เพิ่มการรองรับเครื่องอ่าน รุ่น PDA121RS และ PDA3500RS

R230620 (NALib 0.1.1304)

1. เพิ่มการรองรับเครื่องอ่านบลูทูธ รุ่น TDA01BT

R221129 (NALib 0.1.1302)

1. ปรับปรุง NALib ให้รองรับการใช้งานกับเฟิร์มแวร์ 8.00 ของ TDAi301VC ให้ดียิ่งขึ้น

R221123 (NALib 0.1.1301)

1. ปรับปรุง NALib ให้รองรับการใช้งานกับ React Native และ Flutter ใน Android 13 ได้ดียิ่งขึ้น

R221024 (NALib 0.1.13)

1. ปรับปรุงการทำงานของฟังก์ชัน **`getNIDTextNA`** ให้สามารถอ่านข้อมูลตัวอักษรจากบัตรประจำตัวประชาชนได้มากขึ้นอีก 5 รายการ
2. เพิ่มการรองรับเครื่องอ่านบลูทูธแบบ BLE รุ่น TDA3901BE โดยการกำหนดค่า BLE1 ในตัวแปร **`listOption`** ของฟังก์ชัน **`getReaderListNA`**
3. ปรับปรุงตัวอย่างแอปพลิเคชันให้สามารถกำหนด Target API Level เป็น 33 (Android 13) ได้

R220804 (NALib 0.1.12)

1. ปรับปรุงการทำงานของฟังก์ชัน **`updateLicenseFileNA`** ให้สามารถดาวน์โหลดแฟ้มใบอนุญาตได้จากหลายแหล่งแบบอัตโนมัติ
2. ฟังก์ชัน **`getReaderInfoNA`** เพิ่มการคืนค่า Firmware Version ของเครื่องอ่านได้
3. รองรับเครื่องอ่าน TDAi301U8A ที่ใช้เฟิร์มแวร์ รุ่น 6.xx และ TDAi301VC ที่ใช้เฟิร์มแวร์ รุ่น 8.xx

R220520 (NALib 0.1.10)

1. เพิ่มการรองรับเครื่องอ่านบลูทูธพลังงานต่ำ (BLE) 4 รุ่น คือ TDA301BL, TDAi301BL, TDA301BLM และ TDAi301BLM
2. ปรับปรุงตัวอย่างแอปพลิเคชันให้สามารถกำหนด Target API Level เป็น 31 (Android 12) ได้ (เปลี่ยนมาใช้ Private Storage แทน External Storage)
3. ใช้งานกับ Android 12 ได้สะดวกขึ้น ในเรื่องไม่ต้องเปิดใช้งาน Location Service ขณะใช้งานเครื่องอ่านแบบบลูทูธและ BLE
4. เพิ่มปุ่ม Cancel ในหน้าจอ Pop-up ขณะค้นหาเครื่องอ่านของคำสั่ง **`getReaderListNA`**
5. ปรับเพิ่มความเร็วในการอ่านบัตรประชาชนของเครื่องอ่านรุ่น TDA301VC
6. รองรับการอ่านบัตรประชาชนรุ่น 0004 เช่นบัตร JC3
7. ปรับปรุงคู่มือการใช้งาน SDK
8. แก้ไขจุดบกพร่องต่าง ๆ ของไลบรารี และโปรแกรมตัวอย่าง

คำถามที่ถามบ่อยของชุดพัฒนาซอฟต์แวร์อ่านบัตรประชาชนระบบแอนดรอยด์ HawkEye NALib

คำถาม

อะไรคือชุดพัฒนาซอฟต์แวร์อ่านบัตรประชาชนระบบแอนดรอยด์ HawkEye NALib SDK

คำตอบ

ชุดพัฒนาซอฟต์แวร์อ่านบัตรประชาชนระบบแอนดรอยด์ HawkEye NALib หรือThai ID Card Development Kit for Android คือผลิตภัณฑ์ที่จัดเป็นชุดสำหรับการพัฒนาแอปพลิเคชันโปรแกรมบนระบบแอนดรอยด์ ให้สามารถอ่านข้อมูลจากบัตรประจำตัวประชาชนได้ โดยภายในชุดจะประกอบด้วยเครื่องอ่านบัตรสมาร์ทคาร์ด (แบบ USB หรือ บลูทูธ) และชุดซอฟต์แวร์สำหรับการพัฒนา (SDK) ที่เมื่อพัฒนาแอปพลิเคชันเสร็จแล้วจะสามารถใช้งานอ่านบัตรประจำตัวประชาชนไทยได้ทันที

คำถาม

ชุดเครื่องอ่านบัตรประจำตัวประชาชน TDA (เช่น TDA3310M2, TDA3310C2 และ TDA301BT) ต่างจากสินค้าเครื่องอ่านบัตรสมาร์ทคาร์ด และเครื่องอ่านบัตรประชาชนอื่น ๆ อย่างไร

คำตอบ

ผลิตภัณฑ์ TDA คือชุดสำหรับการพัฒนาแอปพลิเคชัน จะใช้งานอ่านบัตรประชาชนได้เมื่อพัฒนาแอปพลิเคชันเสร็จแล้วเท่านั้น (เหมาะสำหรับโปรแกรมเมอร์)

ผลิตภัณฑ์ TRA คือชุดเครื่องอ่านบัตรประจำตัวประชาชนสำเร็จรูป ภายในชุดจะมีทั้งตัวเครื่องอ่านและซอฟต์แวร์ (แอปพลิเคชัน) สำหรับอ่านบัตรประจำตัวประชาชน ซื้อไปแล้วใช้งานได้ทันที (เหมาะสำหรับผู้ใช้ทั่วไปที่ไม่สามารถพัฒนาโปรแกรมได้ด้วยตัวเอง)

ส่วนผลิตภัณฑ์เครื่องอ่านบัตรเปล่าเช่น bR301, SCR3500B และ SCR3310M2 เป็นเพียงเครื่องอ่านบัตรสมาร์ทคาร์ด (เครื่องเปล่า) ไม่มีซอฟต์แวร์ให้ใช้งาน ผู้ใช้ต้องหาซอฟต์แวร์มาเอง (เหมาะสำหรับหน่วยงานที่มีแอปพลิเคชันของตัวเองแล้วเท่านั้น)

คำถาม

สามารถซื้อเฉพาะเครื่องอ่านบัตรสมาร์ทคาร์ด bR301 หรือ SCR3500B หรือ SCR3310M2 (เครื่องเปล่า) มาใช้งานกับ NALib SDK ได้หรือไม่ และสามารถใช้งานร่วมกับเครื่องอ่านของ TRA ได้หรือไม่

คำตอบ

ชุดพัฒนาซอฟต์แวร์ HawkEye NALib จะใช้งานได้กับเครื่องอ่านที่อยู่ในชุดสินค้าตระกูล TDA, TDAi และ TDC เท่านั้น เพราะเครื่องอ่านทุกเครื่องในชุดสินค้าเหล่านี้ จะผ่านการลงทะเบียนให้ใช้งานกับ NALib มาก่อนแล้ว

ส่วนเครื่องอ่านอื่น ที่ยังไม่ได้ลงทะเบียน หรือเป็นเครื่องเปล่า หรือเป็นเครื่องอ่านในกลุ่มสินค้า TRA จะใช้งาน NALib SDK ไม่ได้

คำถาม

ขั้นตอนการพัฒนาแอปพลิเคชันให้สามารถอ่านบัตรประชาชน จนถึงการนำไปใช้งานจริง หรือจำหน่ายได้ต้องทำอย่างไร

คำตอบ

ทำตามขั้นตอนดังนี้

- ซื้อชุดพัฒนาซอฟต์แวร์ HawkEye NALib มาอย่างน้อย 1 ชุด (แนะนำว่าควรเป็น TDA301BL เพื่อจะได้สามารถพัฒนาให้ใช้ได้กับเครื่องอ่านทั้งแบบบลูทูธ และแบบ USB)
- ศึกษาการใช้งาน NALib SDK จากคู่มือการใช้งาน ซึ่งจะอธิบายถึงวิธีการเรียกใช้ (API) และศึกษาตัวอย่างโปรแกรมจากในชุด SDK
- พัฒนาแอปพลิเคชัน และทดสอบการใช้งานกับเครื่องอ่านบัตรประชาชนที่ได้มาพร้อมกับชุดสินค้า
- เมื่อพัฒนาแอปพลิเคชันจนเสร็จ ให้สั่งซื้อสินค้า TDA, TDAi หรือ TDC เพิ่ม ตามจำนวนและชนิดของเครื่องอ่านที่จะใช้งาน
- ติดตั้งแอปพลิเคชันลงในอุปกรณ์แอนดรอยด์ที่จะใช้งาน และมอบเครื่องอ่านที่ได้มากับชุดสินค้า ให้กับผู้ใช้งาน

คำถาม

ซื้อชุดพัฒนาซอฟต์แวร์ HawkEye NALib มา 1 ชุด แล้วนำแอปพลิเคชันที่พัฒนาได้ไปใช้งานกับเครื่องอ่านรุ่นอื่นที่ไม่ใช่ TDA, TDAi หรือ TDC ได้หรือไม่

คำตอบ

ไม่ได้ เพราะชุดพัฒนา NALib SDK นี้ใช้งานได้กับเครื่องอ่านบัตรที่มาพร้อมกับชุดสินค้า TDA, TDAi หรือ TDC เท่านั้น เครื่องอ่านบัตรรุ่นอื่นใช้ไม่ได้ และบริษัท ฯ ไม่อนุญาตให้นำสิทธิ์นี้ไปใช้งานกับเครื่องอ่านบัตรรุ่นอื่นด้วย

คำถาม

เครื่องอ่านบัตรประชาชนแบบ Virtual Licensing (VL) เป็นอย่างไร ต่างจากเครื่องอ่านแบบธรรมดา (File Licensing) อย่างไร

คำตอบ

เครื่องอ่านแบบที่มีใบอนุญาตฝังอยู่ในตัวเครื่อง (Virtual Licensing Readers) หรือ VL เป็นเครื่องอ่านรุ่นใหม่ที่ไม่มีการใช้แฟ้มใบอนุญาต หรือ License File อีกต่อไป ซึ่งมีข้อดีเช่นพัฒนาแอปพลิเคชันได้ง่ายกว่า, ผู้ใช้ไม่ต้องคอยสั่งอัปเดตแฟ้มใบอนุญาต และไม่ต้องการอินเทอร์เน็ตในการใช้งานเลย
นอกจากนี้ เครื่องอ่านแบบ VL ยังสามารถใช้งานร่วมกับเครื่องอ่านรุ่นปกติที่ใช้ License File ได้ด้วย จึงสามารถนำมาทดแทนเรื่องอ่านรุ่นเก่าได้โดยง่าย

คำถาม

หากเครื่องอ่านบัตรประชาชนในชุดเสีย สามารถนำสิทธิ์ (License) ไปใช้งานต่อได้หรือไม่

คำตอบ

หากเครื่องอ่านบัตรเสียหายจนไม่สามารถใช้งานได้ จะถือว่าสิทธิ์ในการใช้งาน (License) นั้นหมดไปด้วย อย่างไรก็ตามเครื่องอ่านบัตรทุกเครื่องจะมีการรับประกันการใช้งาน หากเครื่องเสียในระยะเวลาประกัน ทางบริษัท ฯ มีบริการซ่อมหรือแก้ไขให้ใช้งานเครื่องอ่านได้จนเป็นปกติ ดังนั้นจะยังคงมีสิทธิ์ในการใช้งานได้อยู่เหมือนเดิม

คำถาม

ชุด NALib SDK นี้สามารถขอข้อมูลแบบ Online จากเซิร์ฟเวอร์ของกรมการปกครองได้หรือไม่

คำตอบ

NALib SDK จะอ่านข้อมูลจากชิปภายในบัตรประชาชนเท่านั้น และทำงานแบบ Offline ไม่มีการขอข้อมูลจากเซิร์ฟเวอร์ของกรมการปกครอง

คำถาม

มีข้อกำหนดในการใช้งานอย่างไรบ้าง

คำตอบ

ชุดพัฒนานี้สามารถใช้งานได้ทั่วไป แต่ทั้งนี้มีข้อกำหนดในการใช้งานบางประการดังนี้

- ห้ามนำไปใช้ในทางที่ผิดกฎหมาย
- ใช้เพื่อการทำงาน, ให้บริการหรือทำธุรกรรมเพื่อเจ้าของบัตรประจำตัวประชาชนหรือบัตรธนาคารเท่านั้น
- ในการนำไปอ่านข้อมูลจากบัตรประจำตัวประชาชนหรือบัตรธนาคาร จะต้องได้รับความยินยอมจากเจ้าของบัตรเท่านั้น
- ห้ามนำข้อมูลภาพและข้อความที่อ่านได้จากบัตรประจำตัวประชาชนหรือบัตรธนาคารไปเปิดเผยหรือเผยแพร่ให้ผู้อื่นโดยไม่ได้รับอนุญาตจากเจ้าของบัตร
- ต้องปฏิบัติตามพระราชบัญญัติคุ้มครองข้อมูลส่วนบุคคลและกฎหมายที่เกี่ยวข้องโดยเคร่งครัด
- กรณีมีความเสียหายใด ๆ เกิดขึ้น อันเนื่องมาจากการนำโปรแกรมคอมพิวเตอร์นี้ หรือข้อมูลที่ได้จากโปรแกรมคอมพิวเตอร์ไปใช้งาน ผู้รับอนุญาตต้องรับผิดชอบในผลของความเสียหายที่เกิดขึ้นทั้งทางแพ่งและอาญาและอื่น ๆ โดยทางบริษัทอาร์แอนด์ดี คอมพิวเตอร์ ซิสเท็ม จำกัด ไม่ต้องร่วมรับผิดชอบด้วย

คำถาม

สามารถดาวน์โหลดซอฟต์แวร์ชุดพัฒนา NALib SDK ได้จากที่ใด

คำตอบ

ผู้พัฒนาสามารถดาวน์โหลดซอฟต์แวร์ชุด SDK ได้จากเว็บไซต์ของบริษัท (www.rd-comp.com) โดยการเข้าไปที่หน้าสินค้า HawkEye TDA รุ่นใดรุ่นหนึ่งก็ได้ (เช่น TDA301BL) แล้วคลิกดาวน์โหลดจาก Link ในหน้านั้น

คำถาม

เสียบเครื่องอ่าน USB กับพอร์ตของโทรศัพท์เสร็จแล้ว แต่ไม่มีหน้าต่าง Pop-up โผล่ขึ้นมาถามให้เลือกแอปพลิเคชัน และเมื่อเรียกแอปพลิเคชันให้อ่านข้อมูล ก็ยังไม่มีหน้าต่าง Pop-up

คำตอบ

อาจเป็นไปได้ 3 กรณี ให้ลองแก้ไขทีละจุด

- ระบบ OTG ของโทรศัพท์อาจถูกปิดอยู่ ให้เข้าหน้า Setting (ตั้งค่า) ของแอนดรอยด์ แล้วลองค้นหาเมนูที่เกี่ยวข้องกับการเปิดการใช้งาน OTG ให้สั่งเปิดการใช้งาน OTG (พบปัญหานี้ในเครื่องของ Oppo, realme และ Vivo ซึ่งอาจจะมีการปิด OTG เองเป็นระยะ)

- โทรศัพท์เครื่องนั้นอาจไม่รองรับระบบ OTG ให้เปลี่ยนโทรศัพท์เป็นเครื่องใหม่ที่รองรับระบบ OTG
- กรณีที่ไม่ได้ตั้งให้ใช้งานเครื่องอ่าน USB แบบ In-App เครื่องอ่านอาจถูกล็อคให้ใช้งานกับแอปพลิเคชันอื่นไปแล้ว (เพราะเผลอแตะ Always หรือ Remember this choice ไปก่อนแล้ว) ให้แก้โดยเข้าไปที่ Setting (ตั้งค่า) ของระบบแอนดรอยด์ > Application Manager (จัดการแอปพลิเคชัน) หรือ App Info (ข้อมูลแอป) > เลือกชื่อแอปพลิเคชันที่เลือกผิด > เลื่อนไปที่ Open by default (เปิดโดยค่าเริ่มต้น) หรือ Launch by default (เริ่มโดยค่าพื้นฐาน) แตะที่ CLEAR DEFAULTS (ล้างค่าเริ่มต้น หรือ ล้างค่าพื้นฐาน) แล้วลองถอดเสียบเครื่องอ่านใหม่อีกครั้ง

หากไม่ทราบชื่อแอปพลิเคชันที่มีปัญหา ให้ไล่แก้ไข CLEAR DEFAULTS ทุกแอปพลิเคชัน
เพื่อหลีกเลี่ยงปัญหานี้ ควรปรับแก้แอปพลิเคชันให้มีหน้าต่าง Pop-up ขออนุญาตใช้เครื่องอ่าน USB จากภายในแอปพลิเคชัน (USB Reader In-App Permission) ด้วยคำสั่ง **`setPermissionsNA(1)`**

คำถาม

เครื่องอ่านบลูทูธแบบ BLE ต่างจากเครื่องอ่านบลูทูธธรรมดาอย่างไร

คำตอบ

เครื่องอ่านบลูทูธแบบพลังงานต่ำ Bluetooth Low Energy-BLE (รุ่นที่ลงท้ายด้วย BL หรือ BLM หรือ BE เช่น TDA301BL, TDA301BLM และ TDA3901BE เป็นต้น) เป็นเครื่องอ่านรุ่นใหม่กว่ารุ่นเดิม (รุ่นเดิมจะลงท้ายด้วย BT เช่น TDA301BT) และมีความแตกต่างดังนี้

- BLE เป็นเทคโนโลยีของ Bluetooth 4.0 ขึ้นไป ขณะที่รุ่น BTใช้งานเทคโนโลยีของ Bluetooth 3.0
- BLE ใช้พลังงานน้อยกว่า
- BLE ไม่ต้องสั่งจับคู่ไว้ก่อน สามารถใช้งานได้ทันที แต่ BT ต้องสั่งจับคู่ก่อน
- BLE อ่านข้อมูลได้เร็วกว่า
- BLE มีระบบรักษาความปลอดภัยภายในสูงกว่า BT

คำถาม

เครื่องอ่านบลูทูธแบบ BLE รุ่น TDA3901BE ต่างจากเครื่องอ่านบลูทูธแบบ BLEรุ่นอื่นอย่างไร

คำตอบ

เครื่องอ่านบลูทูธแบบพลังงานต่ำ BLE รุ่น TDA3901BE เป็นเครื่องอ่านของบริษัท ACS เรียกว่า BLE1 ซึ่งมีความแตกต่างจาก เครื่องอ่านBLEรุ่นก่อนหน้า ได้แก่ TDA301BL, TDAi301BL, TDA301BLM และ TDAi301BLM (เรียกว่า BLE0) ดังนี้

- TDA3901BE มี Sleep Mode เครื่องจะไม่ดับเมื่อถูก Deselect และจะพักเครื่อง (Sleep) เมื่อไม่ได้ใช้งานนานเกิน 1 นาที แต่สามารถปลุกได้ทั้งจากซอฟต์แวร์ และจากการถอดเสียบบัตร ในขณะที่เครื่องอ่านในกลุ่ม BLE0 จะดับเครื่องทันทีที่ Deselect หรือเมื่อไม่ได้ใช้งานนานเกิน 3 นาที
- TDA3901BE มีกระบวนการรักษาความปลอดภัยสูงกว่าเครื่องอ่าน BLE รุ่นอื่น เพราะมีการเข้ารหัสข้อมูลแบบ AES-128 ฝังอยู่ในตัวเครื่องด้วย
- TDA3901BE ทำงานแบบ File Licensing (FL) ต้องใช้แฟ้มใบอนุญาตในการทำงาน ในขณะที่ เครื่องอ่านในกลุ่ม BLE0 ทำงานแบบ Virtual Licensing (VL) ไม่ต้องใช้แฟ้มใบอนุญาต



บริษัท อาร์แอนด์ดี คอมพิวเตอร์ ซิสเท็ม จำกัด
R&D Computer System Co., Ltd.

02-693-1745, 02-694-3133
sales@rd-comp.com
https://www.rd-comp.com