

**คู่มือการใช้งาน**  
**ชุดพัฒนาซอฟต์แวร์อ่านบัตรประชาชน**  
**ระบบ Android และ iOS**  
**HawkEye TDai NiBLE SDK**  
**(สำหรับเครื่องอ่าน BLE)**

*บริษัท อาร์แอนด์ดี คอมพิวเตอร์ ซิสเต็ม จำกัด*

# สารบัญ

สัญญาอนุญาตให้ใช้โปรแกรมคอมพิวเตอร์ .....	i
ส่วนประกอบของสินค้า HawkEye TDAi .....	1
คุณสมบัติของ HawkEye TDi NiBLE SDK .....	1
ไลบรารี .....	2
ขั้นตอนการพัฒนาแอปพลิเคชันสำหรับ iOS, iPadOS .....	3
การนำไปใช้งานหลังจากพัฒนาแอปพลิเคชันเสร็จ .....	4
ผังการทำงานของแอปพลิเคชัน .....	5
การทดสอบแอปพลิเคชันตัวอย่าง NiSample .....	7
รายละเอียดการใช้งาน NiLib API .....	10
openLibNi .....	11
closeLibNi .....	11
getReaderListNi .....	11
selectReaderNi .....	11
deselectReaderNi .....	12
connectCardNi .....	12
disconnectCardNi .....	12
getNIDNumberNi .....	12
getNIDTextNi .....	12
getNIDPhotoNi .....	14
getCardStatusNi .....	14
getRidNi .....	14
getSoftwareInfoNi .....	15
getLicenseInfoNi .....	15
updateLicenseFileNi .....	15
getContextNi .....	16
getCardHandleNi .....	16
attachContextNi .....	16
detachContextNi .....	17
getReaderInfoNi .....	17
getATextNi .....	18
setReaderTypeNi .....	19
scanReaderListBleNi .....	19
stopReaderListBleNi .....	20
รหัสส่งกลับและรหัสข้อผิดพลาด (Return Code) .....	21
ตัวอย่างการใช้งาน NiLib และการพัฒนาแอปพลิเคชัน .....	22

ขั้นตอนการเรียกใช้ฟังก์ชันต่าง ๆ ของ NiLib .....	22
การพัฒนาแอปพลิเคชันด้วย Objective-C และ Swift ใน Xcode .....	23
การกำหนดให้ใช้เครื่องอ่านเป็นแบบ BLE ในโปรเจกต์ .....	26
การใช้งานระบบ Notification ของ NiLib .....	26
การจัดการแฟ้มไบนารี (.dlt) .....	30
ความสามารถในการเชื่อมต่อกับเครื่องอ่านของ NiLib .....	31
การลงทะเบียนแอปพลิเคชันและขอรหัส MFi PPID .....	32
คำแนะนำการใช้งานกับเครื่องอ่านบลูทูธ .....	34
รายการปรับปรุงของ TDi NiBLE SDK .....	37
คำถามที่ถามบ่อยของชุดพัฒนาซอฟต์แวร์อ่านบัตรประชาชน HawkEye TDAi .....	38

# สัญญาอนุญาตให้ใช้โปรแกรมคอมพิวเตอร์

โปรดอ่านข้อความในสัญญานี้โดยตลอดก่อนที่จะเข้าผูกพันตามสัญญาโดยการเปิดใช้งานโปรแกรมคอมพิวเตอร์นี้

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

เมื่อท่านได้ติดตั้ง หรือ ใช้งานโปรแกรมคอมพิวเตอร์นี้ แม้เพียงส่วนใดส่วนหนึ่ง ถือว่าท่านยอมผูกพันในฐานะผู้รับอนุญาตให้ใช้โปรแกรมคอมพิวเตอร์ตามสัญญานี้ ซึ่งต่อไปในสัญญานี้จะเรียกว่า “ผู้รับอนุญาต”

บริษัท อาร์แอนด์ดี คอมพิวเตอร์ ซิสเต็ม จำกัด ตั้งอยู่ที่ 48/20 ถนนรัชดาภิเษก ซอย 20 แขวงสามเสนนอก เขตห้วยขวาง กรุงเทพมหานคร ซึ่งต่อไปในสัญญานี้จะเรียกว่า “ผู้อนุญาต”

ผู้อนุญาตและผู้รับอนุญาตตกลงทำสัญญากันมีข้อความดังต่อไปนี้

## 1. การอนุญาตให้ใช้โปรแกรมคอมพิวเตอร์

1.1 ผู้อนุญาตเป็นเจ้าของลิขสิทธิ์ โปรแกรมคอมพิวเตอร์ ชื่อ Thai ID Card Development Kit for Android and iOS (HawkEye TDAi) ซึ่งมีส่วนประกอบย่อย คือ HawkEye TDi, TDi SDK, R&D NID Card iOS Library (NiOS Lib, NiLib), NiSample, NIOS Sample - Thai ID Card Viewer, NiSample, NIOS Sample (Objective-C และ Swift) และโปรแกรมคอมพิวเตอร์อื่น ๆ ที่แนบมาพร้อมกัน และเอกสารต่าง ๆ ที่ระบุในสัญญานี้ (ซึ่งในสัญญานี้เรียกว่า “โปรแกรมคอมพิวเตอร์”)

1.2 ผู้อนุญาตตกลงอนุญาตให้ผู้รับอนุญาตใช้โปรแกรมคอมพิวเตอร์ตามที่ระบุในสัญญานี้ในลักษณะดังต่อไปนี้

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

## 2. ระยะเวลาการอนุญาต

2.1 ผู้รับอนุญาต จะใช้งานโปรแกรมคอมพิวเตอร์นี้ได้ตั้งแต่วันที่เริ่มผูกพันในฐานะผู้รับอนุญาตจนกว่าจะบอกเลิกสัญญาหรือจนกว่าผู้อนุญาตจะยกเลิกการให้อุญาต

## 3. สิทธิ์และหน้าที่ของผู้รับอนุญาต

3.1 ผู้รับอนุญาตมีสิทธิ์ที่จะใช้โปรแกรมคอมพิวเตอร์และเอกสารต่าง ๆ ตามเงื่อนไขที่ระบุในสัญญานี้

3.2 ผู้รับอนุญาตไม่มีสิทธิ์ที่จะให้ ให้เช่า ให้ยืม หรือขายต่อโปรแกรมคอมพิวเตอร์ หรือเอกสาร หรือสิทธิ์ใด ๆ ที่ให้ไว้ตามสัญญานี้แก่บุคคลอื่น

3.3 ในกรณีผู้รับอนุญาตได้มีการพัฒนาโปรแกรมใช้งานขึ้นมา ผู้รับอนุญาตสามารถนำแฟ้มโปรแกรมคอมพิวเตอร์เฉพาะส่วนที่เป็นแฟ้มโลบารี ไปเผยแพร่ต่อแก่บุคคลอื่นได้ในลักษณะของการอนุญาตให้ใช้งานเท่านั้น และผู้รับอนุญาตจะต้องส่งมอบสัญญานี้ในรูปแบบของเอกสารกระดาษหรือแฟ้มคอมพิวเตอร์แนบไปพร้อมกับแฟ้มโลบารี ที่ส่งมอบแก่บุคคลอื่นด้วย และบุคคลอื่นที่ได้รับแฟ้มโลบารีจะต้องยอมผูกพันในฐานะผู้รับอนุญาตให้ใช้โปรแกรมคอมพิวเตอร์ตามสัญญานี้ด้วย ผู้รับอนุญาตจะต้องไม่มอบแฟ้มโปรแกรมคอมพิวเตอร์อื่นของโปรแกรมคอมพิวเตอร์นี้ ให้กับบุคคลอื่น

3.4 ผู้รับอนุญาตมีสิทธิ์ทำสำเนาโปรแกรมคอมพิวเตอร์ได้ 1 ชุด (หนึ่งชุด) เพื่อป้องกันการสูญหายในระหว่างการใช้งานโปรแกรมคอมพิวเตอร์ดังกล่าวเท่านั้น

3.5 ผู้รับอนุญาตขอรับรองว่าจะไม่เปลี่ยนแปลง คัดแปลง ลบ ทำลาย ทำให้เสียหาย หรือทำให้ไม่ชัดเจน ซึ่ง ชื่อของโปรแกรม คอมพิวเตอร์ ชื่อของเจ้าของลิขสิทธิ์ เครื่องหมายหรือสัญลักษณ์แสดงความเป็นเจ้าของลิขสิทธิ์หรือเครื่องหมายการค้าของเจ้าของลิขสิทธิ์ ไม่ว่าจะโดยจงใจหรือประมาทเลินเล่อก็ตาม

3.6 ผู้รับอนุญาตจะดูแลรักษาโปรแกรมคอมพิวเตอร์ เอกสารต่าง ๆ และสำเนาทั้งหลาย เป็นอย่างดีที่สุด จะไม่ยอมให้ผู้อื่นเข้าถึง นำไปใช้ นำไปศึกษา ทำสำเนา หรือกระทำการประการหนึ่งประการใดต่อโปรแกรมคอมพิวเตอร์ 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 คัดแปลง แก้ไขงานต้นฉบับ ซัดมา ทำให้เสียหาย ทำวิศวกรรมย้อนกลับ (Reverse Engineer) แปล (Translate) แปลย้อนกลับ (Decompile) แปลย้อนกลับภาษาเครื่อง (Disassembly) วิเคราะห์การทำงานภายใน (System Analyze) ทดสอบผ่านกล่องดำ (Black Box Test) ถอดรหัส (Decrypt) ได้การทำงานผ่านเครื่องมือค้นหาจุดบกพร่อง (Debug) ทั้งในระดับรหัสไบต์ (Bytecode Debugging) และระดับไบนารี (Binary Code Debugging) รื้อและสร้างใหม่ (Deconstruct) ดักจับข้อมูล (Sniff) จำลองการทำงาน (Simulate) เลียนแบบการทำงาน (Emulate) และการกระทำอื่นใดก็ตามที่ให้ผลเช่นเดียวกันกับการกระทำข้างต้น ไม่ว่าจะมิชื่อเรียกว่าอย่างไรก็ตาม

#### 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, ให้บริการหรือทำธุรกรรมเพื่อเจ้าของบัตรหรือผู้ถือบัตรประจำตัวประชาชนเท่านั้น
- ในการนำไปอ่านข้อมูลจากบัตรประจำตัวประชาชน จะต้องได้รับความยินยอมจากเจ้าของบัตรหรือผู้ถือบัตรเท่านั้น
- ห้ามนำข้อมูลภาพและข้อความที่อ่านได้จากบัตรประจำตัวประชาชนไปเปิดเผยหรือเผยแพร่ให้ผู้อื่นโดยไม่ได้รับอนุญาตจากเจ้าของบัตรหรือผู้ถือบัตร
- กรณีมีความเสียหายใด ๆ เกิดขึ้น อันเนื่องมาจากการนำโปรแกรมคอมพิวเตอร์นี้ หรือข้อมูลที่ได้จากโปรแกรมคอมพิวเตอร์ไปใช้งาน ผู้รับอนุญาตต้องรับผิดชอบในผลของความเสียหายที่เกิดขึ้นทั้งทางแพ่งและอาญาและอื่น ๆ โดยทางบริษัทอาร์แอนดีคอมพิวเตอร์ ซิสเต็ม จำกัด ไม่ต้องร่วมรับผิดชอบด้วย

# คู่มือการใช้งานชุดพัฒนาซอฟต์แวร์อ่านบัตรประชาชน ระบบ Android และ iOS HawkEye TDAi NiBLE SDK (สำหรับเครื่องอ่าน BLE)

บริษัท อาร์แอนดี คอมพิวเตอร์ ซิสเต็ม จำกัด

Thai ID Card Development Kit for Android and iOS (HawkEye TDAi) คือชุดโปรแกรมคอมพิวเตอร์สำหรับการพัฒนาแอปพลิเคชันเพื่อการติดต่อกับเครื่องอ่านบัตรสมาร์ทการ์ดและอ่านข้อมูลจากบัตรประจำตัวประชาชนไทยแบบสมาร์ทการ์ด ภายใต้ระบบปฏิบัติการ Android และ iOS

คู่มือนี้จัดทำเพื่อใช้งานกับเครื่องอ่าน Bluetooth แบบ BLE เป็นหลัก ได้แก่เครื่องอ่านรุ่น TDAi301BL และ TDAi301BLM

## 1. ส่วนประกอบของสินค้า HawkEye TDAi จะประกอบด้วย 3 ส่วนหลักคือ

- 1.1 เครื่องอ่านบัตรประจำตัวประชาชนที่ลงทะเบียนให้ใช้งานกับ HawkEye TDi และ HawkEye TDA ไว้แล้ว
- 1.2 ซอฟต์แวร์ชุดพัฒนาสำหรับระบบ iOS (รวมทั้ง iPadOS ด้วย) ในชื่อ TDi SDK ซึ่งจะอธิบายวิธีการใช้งานทั้งหมดในเอกสารนี้ ผู้พัฒนาสามารถดาวน์โหลดชุด SDK ของ TDi ได้จากเว็บไซต์ของบริษัท ฯ ส่วนประกอบของชุดพัฒนาซอฟต์แวร์ TDi SDK ประกอบด้วย
  - ตัวอย่างแอปพลิเคชัน NiBLESample พร้อม Source Code สำหรับใช้ศึกษาเรียนรู้วิธีการเขียนโปรแกรมแบบง่าย ๆ ด้วยภาษา Swift
  - แฟ้มไลบรารี NiLib สำหรับนำไปรวมกับโปรแกรมที่พัฒนาเพื่อใช้ในการสร้างแอปพลิเคชัน
  - คู่มือการใช้งาน
- 1.3 ซอฟต์แวร์ชุดพัฒนาสำหรับระบบแอนดรอยด์ ในชื่อ TDA NALib SDK ซึ่งเป็นชุดพัฒนาชุดที่สอง ใช้ในกรณีที่ต้องการนำเครื่องอ่านนี้ไปใช้งานกับระบบแอนดรอยด์ด้วย

ในคู่มือฉบับนี้จะเน้นให้ข้อมูลของ TDi SDK หรือชุดพัฒนาสำหรับระบบ iOS/iPadOS เป็นหลัก หากต้องการทราบรายละเอียดการพัฒนาของ TDA SDK สำหรับระบบแอนดรอยด์ สามารถดูได้จากคู่มือของ TDA NALib SDK ที่ได้จัดทำแยกไว้ต่างหาก

## 2. คุณสมบัติของ HawkEye TDi NiBLE SDK

- 2.1 ใช้งานได้กับเครื่องอ่านบัตรสมาร์ทการ์ดที่มีคุณสมบัติรองรับ R&D NID Card iOS Library (NiLib) เช่นเครื่องอ่านแบบเสียบ (Lightning, USB Type-C) และเครื่องอ่านแบบบลูทูธธรรมดา (Bluetooth 3.0) และแบบพลังงานต่ำ (BLE) (NiBLE SDK เหมาะสำหรับเครื่องอ่านแบบ BLE เท่านั้น)
- 2.2 ใช้งานกับใบอนุญาตของเครื่องอ่านได้ 2 รูปแบบ คือ
  - 2.2.1 เครื่องอ่านแบบที่ใส่แฟ้มใบอนุญาต (File Licensing-FL) เช่น TDAi301U8A และ TDAi301BT เป็นต้น
  - 2.2.2 เครื่องอ่านแบบที่ฝังใบอนุญาตอยู่ในตัวเครื่อง (Virtual Licensing-VL) ซึ่งไม่จำเป็นต้องใส่แฟ้มใบอนุญาต เช่นเครื่องอ่านรุ่น TDAi301BL และ TDAi301BLM
- 2.3 ใช้งานร่วมกับโปรแกรมภาษา Objective-C และ Swift ได้

2.4 ใช้กับเครื่อง **iDevice** ของ Apple ได้แก่ iPhone และ iPad ที่ใช้ระบบปฏิบัติการ iOS รุ่น 8.0 หรือ iPadOS รุ่น 13.0 ขึ้นไป (รุ่นของระบบปฏิบัติการที่ใช้งานได้จริง ขึ้นกับการกำหนดขั้นต่ำของ Apple App Store และ เครื่องมือที่ใช้ในการพัฒนา)

2.5 สามารถอ่านบัตรประจำตัวประชาชนแบบสมาร์ทการ์ดของกรมการปกครองได้เป็นอย่างดี

### 3. ไบรารี

ในชุดพัฒนา TDi จะมีแฟ้มไลบรารีอยู่ 2 ชุด ชุดแรกเป็นไลบรารีสำหรับการอ่านบัตรประชาชน และชุดที่ 2 เป็นไลบรารี CCID ของตัวเครื่องอ่านสำหรับการติดต่อกับฮาร์ดแวร์

#### 3.1 ข้อมูลของไลบรารี

3.1.1 NiLib เป็นไลบรารีสำหรับการอ่านข้อมูลจากบัตรประจำตัวประชาชนไทย พัฒนาโดยและเป็นลิขสิทธิ์ของบริษัท อาร์แอนดีดี คอมพิวเตอร์ ซิสเต็ม จำกัด (R&D Library)

3.1.2 ในการติดต่อกับเครื่องอ่านบัตรสมาร์ทการ์ด NiLib จะอาศัยการใช้งานผ่านไลบรารีแบบ CCID สำหรับเครื่องอ่านบัตรสมาร์ทการ์ดของ Feitian Technologies Co., Ltd. (FT Library)

3.1.3 ในการพัฒนาแอปพลิเคชันสำหรับการอ่านบัตรประจำตัวประชาชนจึงต้องใช้ไลบรารี 2 ชุดด้วยกันเสมอ คือไลบรารี NiLib สำหรับการอ่านบัตรประชาชนของ R&D และไลบรารี CCID ของ Feitian Technologies Co., Ltd.

#### 3.2 องค์ประกอบของไลบรารี

3.2.1 ไบรารีของ NiLib ที่เตรียมให้ไว้สำหรับการพัฒนา จะเป็นคลาส (Class) ชื่อ NiOS โดยมีแฟ้มที่เกี่ยวข้องดังต่อไปนี้

- NiLib.a แฟ้มไลบรารีสำหรับนำไป Link ใช้งานในขั้นตอนการพัฒนาแอปพลิเคชัน
- NiOS.h แฟ้ม Header สำหรับนำไปรวมกับ Source Code เพื่อใช้ในขั้นตอนการพัฒนาแอปพลิเคชัน
- rdnidlib.dlt แฟ้มใบอนุญาต (License File) สำหรับใช้ในการตรวจสอบสิทธิการใช้งานของเครื่องอ่านแบบ FL แต่ละเครื่อง (ตรวจสอบจาก S/N) ผู้พัฒนาจะต้องนำแฟ้มนี้ไปวางไว้ในจุดที่แอปพลิเคชันสามารถเข้าถึงและอ่านแฟ้มนี้ได้ในขณะที่ใช้งานแอปพลิเคชัน (Runtime) อย่างไรก็ตามหากตั้งใจให้แอปพลิเคชันใช้งานเฉพาะเครื่องอ่านแบบ VL ก็ไม่จำเป็นต้องมีแฟ้มนี้ไว้ในแอปพลิเคชัน
  - **เครื่องอ่านแบบ BLE ไม่จำเป็นต้องใช้ แฟ้มใบอนุญาต**

3.2.2 ไบรารีเครื่องอ่านบัตรสมาร์ทการ์ด CCID ของ Feitan Technologies (CCID Lib) ประกอบด้วยแฟ้มดังต่อไปนี้

- libiRockey301\_ccid\_x.xx.x.a แฟ้มไลบรารีสำหรับนำไป Link ใช้งานในขั้นตอนการพัฒนาแอปพลิเคชัน
- แฟ้ม Header สำหรับนำไปรวมกับ Source Code ได้แก่
  - ft301u.h
  - ReaderInterface.h
  - winscard.h
  - wintypes.h

#### 4. ขั้นตอนการพัฒนาแอปพลิเคชันสำหรับ iOS, iPadOS

- 4.1 ซึ่ชุดพัฒนาซอฟต์แวร์อ่านบัตรประชาชน HawkEye TDAi มาอย่างน้อย 1 ชุด
- 4.2 ลองติดตั้งแอปพลิเคชันตัวอย่าง NiSample (Swift) จาก Apple App Store แล้วลองทดสอบใช้งานว่า  
เครื่องอ่านทำงานได้ถูกต้อง
  - **เครื่องอ่านแบบ BLE ไม่สามารถใช้กับแอปพลิเคชันตัวอย่างนี้ได้**
- 4.3 ดาวน์โหลดชุดซอฟต์แวร์ TDi SDK จากเว็บไซต์ของบริษัท (www.rd-comp.com) โดยการเข้าไปที่หน้า  
สินค้า HawkEye TDAi รุ่นใดรุ่นหนึ่งก็ได้ (เช่น TDAi301BT) แล้วคลิกดาวน์โหลดจาก Link ในหน้า  
นั้น หรือใช้เบราว์เซอร์เข้าไปโดยตรงที่ URL นี้  
**[bit.ly/2fK8sER](https://bit.ly/2fK8sER)**
  - **สำหรับเครื่องอ่านแบบ BLE ให้คลิกดาวน์โหลดจาก Link ในหน้าสินค้า TDAi301BL**
- 4.4 ศึกษา Source Code ตัวอย่างของ NiBLESample (Swift) ที่ดาวน์โหลดมาและดูคำแนะนำในคู่มือนี้เพิ่มเติม
- 4.5 สร้างโปรเจกต์ใหม่ภายใต้ Xcode แล้วพัฒนาแอปพลิเคชันตามต้องการ
- 4.6 ทดสอบแอปพลิเคชันที่พัฒนาได้
- 4.7 เมื่อพัฒนาแอปพลิเคชันเสร็จแล้ว ในการเผยแพร่ขึ้น App Store ให้ติดต่อบริษัทอาร์แอนดีดี  
คอมพิวเตอร์ ซิสเต็ม จำกัด เพื่อขอข้อมูล PPID ที่จะนำไปใช้ในการอัปโหลดขึ้นไปยัง App Store
  - **สำหรับเครื่องอ่านแบบ BLE ไม่ต้องขอ PPID สามารถอัปโหลดขึ้นไปยัง App Store ได้เลย**
- 4.8 นำแอปพลิเคชันอัปโหลดขึ้นไปยัง App Store
- 4.9 ดาวน์โหลดแอปพลิเคชันจาก App Store มาทดสอบใช้งาน

#### หมายเหตุ

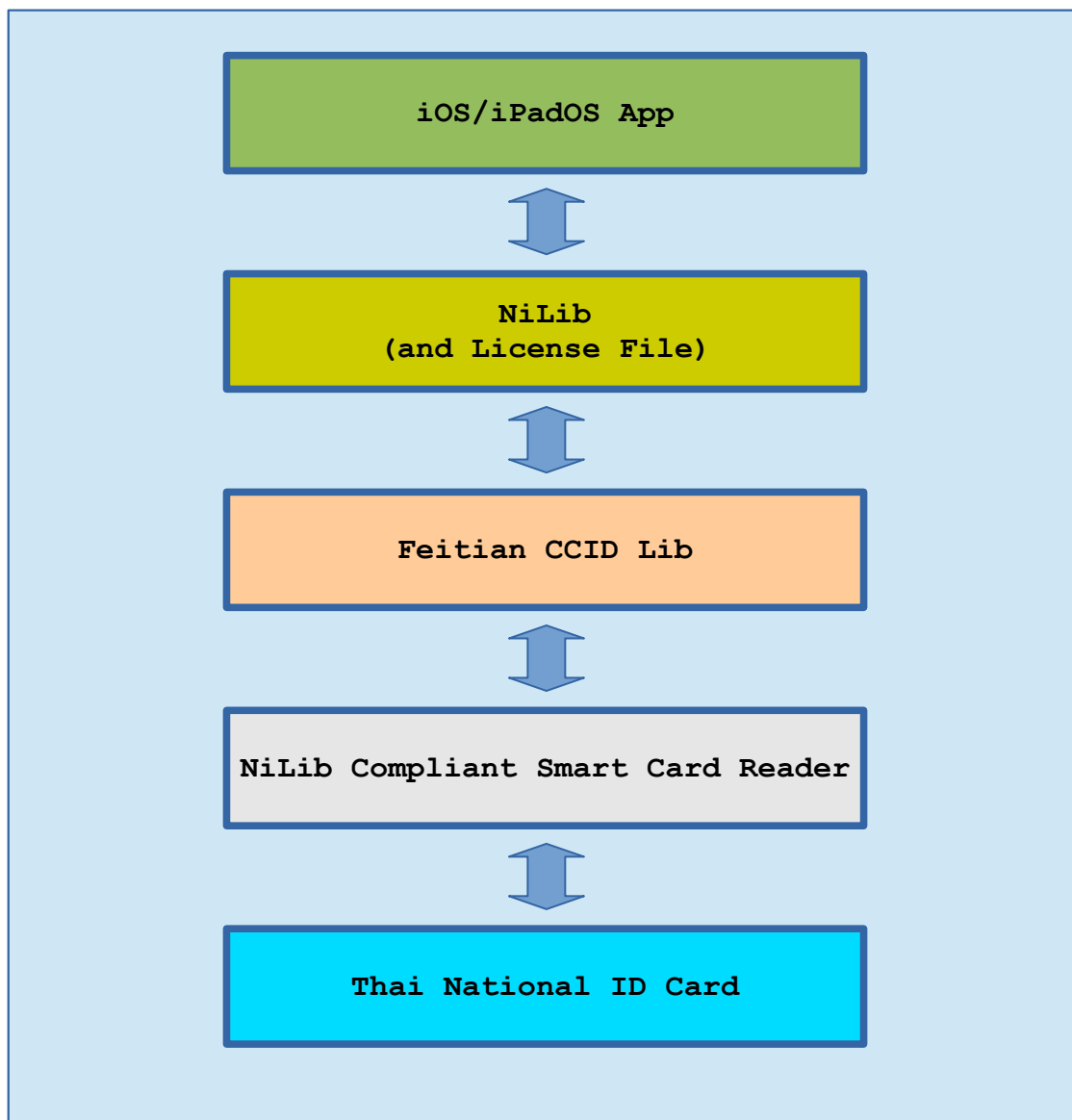
- ผู้พัฒนาจะต้องมีความรู้ในการพัฒนาแอปพลิเคชันสำหรับระบบ iOS แล้วเป็นอย่างดี ก่อนที่จะเริ่มใช้ชุดพัฒนา HawkEye TDi SDK ได้



## 5. การนำไปใช้งานหลังจากพัฒนาแอปพลิเคชันเสร็จ

- 5.1 ให้ผู้ใช้ (End Users) จัดซื้อ-จัดหาชุดเครื่องอ่านบัตรประชาชน HawkEye TDAi ตามรุ่นที่ต้องการ
- 5.2 หากเลือกเครื่องอ่านแบบ Bluetooth 3.0 (TDAi301BT) ขอให้ผู้ใช้ตั้งจับคู่เครื่องอ่านบลูทูธกับอุปกรณ์มือถือก่อน โดยเข้าที่เมนู การตั้งค่า ของระบบ iOS เลือกที่ **บลูทูธ** สั่งเปิด แล้วตั้งจับคู่ (Pair) กับเครื่องอ่าน
  - **เครื่องอ่านแบบ BLE ไม่ต้องจับคู่ก่อนใช้งาน**
- 5.3 ให้ผู้ใช้ติดตั้งแอปพลิเคชันจาก App Store หรือจากไฟล์ของผู้พัฒนา
- 5.4 ให้ผู้ใช้เปิดใช้งานแอปพลิเคชัน

## 6. ฝั่งการทำงานของแอปพลิเคชัน



### iOS/iPadOS App

คือแอปพลิเคชันบนระบบ iOS รวมทั้ง iPadOS ที่จะพัฒนาให้สามารถอ่านบัตรประจำตัวประชาชนได้

### NiLib (and License File)

คือ R&D NiLib Library ที่มีไว้ให้เรียกใช้งาน ผู้พัฒนาแอปพลิเคชัน (iOS App) จะต้องใช้ฟังก์ชันต่าง ๆ ตามที่มีให้  
นี้เพื่อส่งอ่านข้อมูลจากบัตรประชาชน

สำหรับไฟล์ไลเซนส์ (License File) หรือแฟ้มใบอนุญาต มีไว้เพื่อใช้ในการตรวจสอบเครื่องอ่านแบบ FL ที่ใช้ว่า  
เป็นเครื่องอ่านที่ผ่านการลงทะเบียนและมีสิทธิ์ในการใช้งานหรือไม่

หมายเหตุ กรณีเครื่องอ่านเป็นแบบ Virtual Licensing จะไม่มีการใช้แฟ้มใบอนุญาต จึงไม่จำเป็นต้องมีไฟล์  
ไลเซนส์นี้

- เครื่องอ่านแบบ BLE เครื่องอ่านเป็นแบบ Virtual Licensing จึงไม่ต้องใช้แฟ้มใบอนุญาต

**Feitian CCID Lib**

คือไลบรารีมาตรฐานที่จัดทำโดย Feitian Technologies เพื่อใช้ในการติดต่อกับเครื่องอ่านบัตรสมาร์ทการ์ด

**NiLib Compliant Smart Card Reader**

คือเครื่องอ่านบัตรสมาร์ทการ์ดที่มาพร้อมกับชุดสินค้า ซึ่งต้องเป็นเครื่องอ่านที่ผ่านการลงทะเบียนไว้แล้วเท่านั้น จึงจะใช้งานได้ ส่วนเครื่องอ่านเปล่าที่ยังไม่ได้ลงทะเบียน จะไม่สามารถใช้งานได้

**Thai National ID Card**

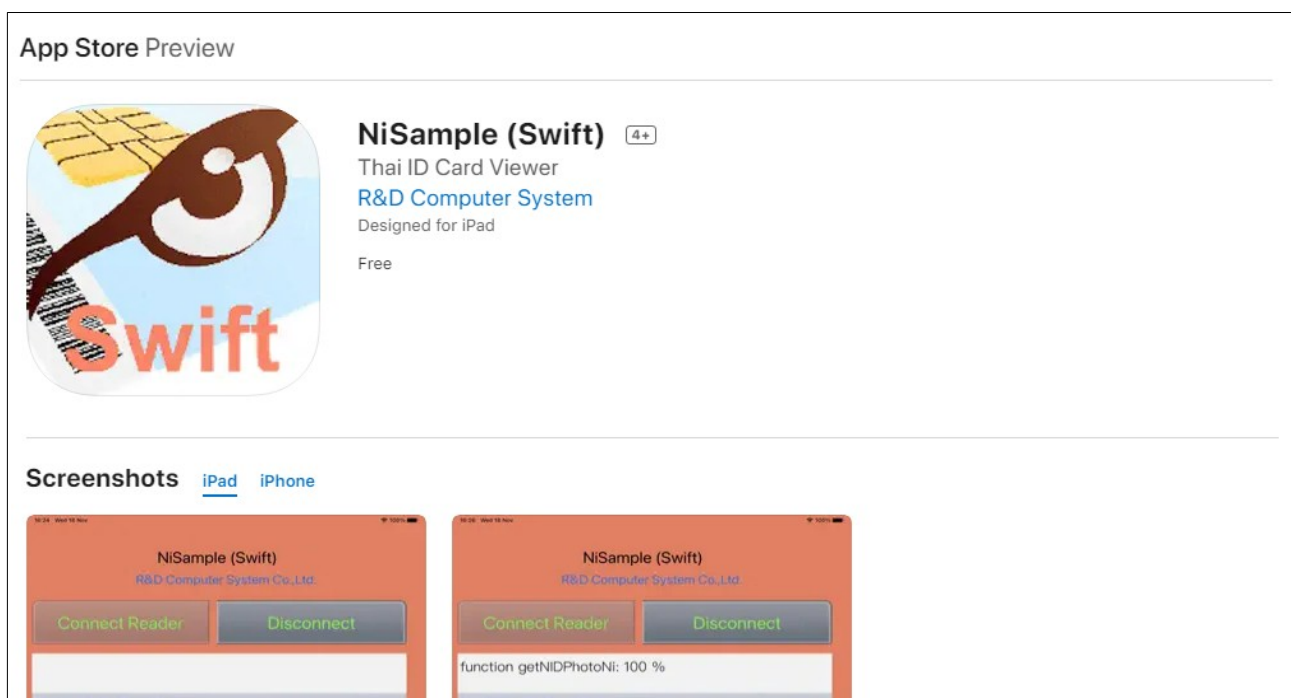
คือบัตรประจำตัวประชาชนไทยแบบสมาร์ทการ์ด

## 7. การทดสอบแอปพลิเคชันตัวอย่าง NiSample

### 7.1 การทดสอบแอปพลิเคชันตัวอย่างที่พัฒนามาจากภาษา Swift

- เครื่องอ่านแบบ BLE ไม่สามารถใช้กับแอปพลิเคชันตัวอย่างนี้ได้ ให้ข้ามหัวข้อนี้ได้ทั้งหมด

7.1.1 ให้ติดตั้งแอปพลิเคชัน Ni Sample (Swift) – Thai ID Card Viewer จาก App Store ที่ <https://itunes.apple.com/us/app/nios-sample-thai-id-card-viewer/id1101615070?mt=8> หรือ ค้นหา จากในแอปพลิเคชัน App Store ใช้คำค้นหาว่า “NiSample” หรือสแกน QR Code นี้



### 7.1.2 การเชื่อมต่อกับเครื่องอ่าน

#### 7.1.2.1 กรณีเป็นเครื่องอ่านบลูทูธ 3 (TDAi301BT)

- เปิดใช้งานเครื่องอ่านบลูทูธ โดยการกดปุ่มที่เครื่องอ่าน 1 ครั้ง สังเกตไฟสีฟ้าจะติดกะพริบ
- บนอุปกรณ์ iDevice (iPhone หรือ iPad) เข้าเมนู การตั้งค่า , บลูทูธ แล้วเปิดใช้งานบลูทูธ
- ดูรายชื่ออุปกรณ์ที่ค้นหพบ เครื่องอ่านบลูทูธจะมีชื่อขึ้นต้นด้วยตัวอักษร FT ในลักษณะนี้ FT\_xxxxxxxxxxxx เช่น FT\_8CDE521A64CB ให้แตะที่ชื่อเครื่องอ่านนั้นเพื่อเชื่อมต่อ (สัักพักจะมีสถานะเป็น “เชื่อมต่ออยู่”)
- ออกจากเมนู การตั้งค่า

#### 7.1.2.2 กรณีเป็นเครื่องอ่านแบบ Lightning หรือ USB Type-C

- เสียบเครื่องอ่านเข้ากับพอร์ต Lightning หรือ USB Type-C หันด้านที่เป็นสีฟ้าขึ้น

- สังเกตจะมีไฟสีขาวติดกระพริบ ให้รอสักพักประมาณ 2 วินาที จนกว่าไฟจะติดสว่างค้าง

#### 7.1.3 เข้าใช้งานแอปพลิเคชัน NiSample

- แตะที่ **Connect Reader** รอจนหน้าจอแสดงชื่อเครื่องอ่านที่เชื่อมต่อได้
- เลียบบัตรประชาชนที่เครื่องอ่านบัตร จะเห็นไฟสีน้ำเงินติดกระพริบ
- แตะที่ **Read Card** จะเห็นการอ่านบัตร 2 จังหวะ จังหวะแรกเป็นการอ่านข้อมูลตัวอักษร จังหวะที่ 2 เป็นการอ่านรูปภาพ ที่เครื่องอ่านจะเห็นไฟกระพริบตามจังหวะการอ่าน
- เมื่อจะหยุดการทดสอบ
  - แตะที่ **Disconnect**
  - แตะที่ **Exit**

#### 7.1.4 กรณีเกิดปัญหาระหว่างการทดสอบ NiSample

- อาจเกิดจากเพิ่มใบอนุญาตล้ำสมัย ให้ลองโหลดเพิ่มใหม่โดยการแตะที่ปุ่ม **Update LIC**
- สำหรับเครื่องอ่านแบบบลูทูธ
  - ตรวจสอบในเมนู บลูทูธ ของระบบ iOS ว่า อุปกรณ์ FT\_xxxxxxxxxxxxx ยังคง “เชื่อมต่ออยู่” หากไม่ใช่ ให้ปิดการใช้งานบลูทูธของ iDevice ก่อน แล้วเปิดใหม่ และจับคู่เครื่องอ่านกันใหม่
  - สังเกตว่าไฟสีน้ำเงินของเครื่องอ่านบัตร ยังติดสว่างค้าง หรือกระพริบ 2 ครั้งอยู่หรือไม่ หากดับไป ให้กดปุ่มเปิดใหม่ และเริ่มต้นการจับคู่กับ iDevice ใหม่
  - หากเป็นไปได้ ในระหว่างการทดสอบควรปิดการทำงานของบลูทูธของเครื่อง iDevice หรือ Android เครื่องอื่น ๆ ที่เคยจับคู่กับเครื่องอ่านบลูทูธนี้ไว้ทั้งหมด เพราะอาจมีการแย่งการเชื่อมต่อโดยที่เราไม่รู้ตัว
  - หากยังอ่านบัตรไม่ได้ ให้ทำตามขั้นตอนต่อไปนี้ตามลำดับ
    - ปิดแอปพลิเคชัน NiSample (แตะที่ปุ่ม **Exit**)
    - ปิดการทำงานของบลูทูธบนเครื่อง iDevice
    - กดปุ่มปิดไฟเครื่องอ่านบัตรบลูทูธ แล้วเปิดใหม่
    - เปิดการทำงานของบลูทูธบนเครื่อง iDevice
    - จับคู่เครื่องอ่านบัตรในเมนู บลูทูธ จนได้สถานะ “เชื่อมต่ออยู่”
    - เปิดใช้งานแอปพลิเคชัน NiSample
    - แตะที่ **Connect Reader** ซ้ำ ๆ จนกว่าจะได้ชื่อเครื่องอ่าน
- สำหรับเครื่องอ่านแบบ Lightning และ USB Type-C
  - สังเกตว่าไฟสีขาวที่เครื่องอ่านยังคงติดสว่างอยู่ หากดับไป ให้ดึงเครื่องอ่านออกแล้วเสียบใหม่ รอจนไฟสีขาวหยุดกระพริบ และติดสว่างค้าง แล้วลองอีกครั้งตั้งแต่ต้น
  - หากยังอ่านบัตรไม่ได้ ให้ทำตามขั้นตอนต่อไปนี้ตามลำดับ
    - ปิดแอปพลิเคชัน NiSample (แตะที่ปุ่ม **Exit**)
    - ดึงเครื่องอ่านบัตรออกจากพอร์ต แล้วเสียบใหม่
    - เปิดใช้งานแอปพลิเคชัน NiSample

- แตะที่ **Connect Reader** ซ้ำ ๆ จนกว่าจะได้ชื่อเครื่องอ่าน

#### หมายเหตุ

เมื่อเสร็จสิ้นการทดสอบแล้ว ควรปิดแอปพลิเคชันโดยการแตะที่ปุ่ม **Exit** ทุกครั้ง และหากเป็นการทดสอบกับเครื่องอ่านแบบบลูทูธ ควรปิดการใช้งานบลูทูธของอุปกรณ์ iDevice ด้วย เพื่อเป็นการบังคับให้ iDevice หยุดการเชื่อมต่อกับเครื่องอ่านบลูทูธ

## 8. รายละเอียดการใช้งาน NiLib API

คำสั่งต่าง ๆ ที่เรียกใช้งานผ่าน API ของ NiLib จะมีดังต่อไปนี้

คำสั่ง	หน้าที่
<b>openLibNi</b> <b>openNiOSLibNi</b>	เปิดการใช้งานไลบรารี NiLib
<b>closeLibNi</b> <b>closeNiOSLibNi</b>	ปิดการใช้งาน NiLib
<b>getReaderListNi</b>	ขอรายชื่อเครื่องอ่านบัตรสมาร์ทการ์ดที่กำลังต่ออยู่กับอุปกรณ์ iDevice
<b>selectReaderNi</b>	เลือกเครื่องอ่านบัตรสมาร์ทการ์ดที่จะใช้ในการทำงานต่อไป
<b>deselectReaderNi</b>	ยกเลิกการเลือกเครื่องอ่านบัตรสมาร์ทการ์ด
<b>connectCardNi</b>	สั่งให้เชื่อมต่อกับบัตรประจำตัวประชาชน
<b>disconnectCardNi</b>	ยกเลิกการเชื่อมต่อกับบัตรประจำตัวประชาชน
<b>getNIDNumberNi</b>	อ่านเลขประจำตัวประชาชนจากบัตร
<b>getNIDTextNi</b>	อ่านข้อมูลตัวอักษรหน้าบัตรประจำตัวประชาชน
<b>getNIDPhotoNi</b>	อ่านรูปภาพถ่ายใบหน้าจากบัตรประจำตัวประชาชน
<b>getCardStatusNi</b>	ขอสถานะการเสียบบัตรที่เครื่องอ่าน
<b>getRidNi</b>	ขอข้อมูลเลขประจำตัวของเครื่องอ่านบัตร (RID)
<b>getSoftwareInfoNi</b>	ขอข้อมูลของไลบรารี
<b>getLicenseInfoNi</b>	ขอข้อมูลของแฟ้มใบอนุญาตที่ใช้งานอยู่
<b>updateLicenseFileNi</b>	ตรวจสอบและปรับเปลี่ยนแฟ้มใบอนุญาต (License File) ให้เป็นปัจจุบัน
<b>getContextNi</b>	อ่านค่า Context Handle ของ NiLib
<b>getCardHandleNi</b>	อ่านค่า Handle ของเครื่องอ่านบัตรที่ถูกเชื่อมต่อ
<b>attachContextNi</b>	ผูกค่า (binding) ของ Context Handle
<b>detachContextNi</b>	ยกเลิกการผูกค่า (binding) ของ Context Handle
<b>getReaderInfoNi</b>	ขอข้อมูลของเครื่องอ่านที่กำลังเชื่อมต่ออยู่
<b>getATextNi</b>	อ่านข้อมูลตัวอักษรทั้งหมดของบัตรประจำตัวประชาชน
<b>setReaderTypeNi</b>	กำหนดประเภทเครื่องอ่านที่จะใช้งาน BLE
<b>scanReaderListBleNi</b>	ค้นหาเครื่องอ่านชนิด BLE
<b>stopReaderListBleNi</b>	หยุดการค้นหาเครื่องอ่านชนิด BLE

### 8.1 ฟังก์ชัน (SInt32) openLibNi: (NSMutableString \*)LICfile

Description	เปิดการใช้งานไลบรารี NiLib
Input parameter	<b>LICfile:</b> ชื่อและ Path ของแฟ้มใบอนุญาต (License File) ควรใช้ชื่อแฟ้มเป็น rdnidlib.dlt เสมอ เพื่อป้องกันการสับสน (FL+VL Mode) กรณีจะใช้เครื่องอ่านแบบ Virtual Licensing เพียงอย่างเดียว (VL Mode) ให้ส่งค่า <b>LICfile</b> เป็น String ที่ความยาวเป็นศูนย์ (" ") ไม่ต้องส่งชื่อแฟ้มใบอนุญาต (เพราะไม่มีการใช้แฟ้มใบอนุญาต) <ul style="list-style-type: none"><li>• เครื่องอ่านแบบ BLE ควรใช้งานแบบ VL Mode เท่านั้น</li></ul>
Return type	SInt32
Return value	Return Code
หมายเหตุ	ใน Objective-C จะใช้ชื่อฟังก์ชันเป็น <b>openNiOSLibNi</b>

### 8.2 ฟังก์ชัน (SInt32) closeLibNi

Description	ปิดการใช้งาน NiLib ให้เรียกฟังก์ชันนี้เมื่อจะหยุดการใช้งาน NiLib
Input parameter	ไม่มี
Return type	SInt32
Return value	Return Code
หมายเหตุ	ใน Objective-C จะใช้ชื่อฟังก์ชันเป็น <b>closeNiOSLibNi</b>

### 8.3 ฟังก์ชัน (SInt32) getReaderListNi: (NSMutableArray \*)readerList

Description	ขอรายชื่อเครื่องอ่านบัตรสมาร์ทการ์ดที่กำลังต่ออยู่กับอุปกรณ์ iDevice
Input parameter	ไม่มี
Output	<b>readerList:</b> จะคืนชื่อของเครื่องอ่านบัตรสมาร์ทการ์ดแต่ละเครื่องที่พบในระบบไว้ใน Array นี้ โดยเครื่องอ่านที่พบเครื่องแรกจะอยู่ใน <b>readerList[0]</b> หากมีเครื่องที่สองก็จะอยู่ใน <b>readerList[1]</b> และต่อ ๆ ไป จนครบทุกเครื่องอ่าน ชื่อของเครื่องอ่านเป็นข้อมูลชนิด NSMutableString
Return type	SInt32
Return value	จำนวนเครื่องอ่านที่ต่ออยู่ (กรณีเป็นเลขมากกว่า 0) Return Code (กรณีเป็นเลขลบ)
หมายเหตุ	เครื่องอ่านแบบ BLE ให้ใช้คำสั่ง <b>scanReaderListBleNi</b> แทน

### 8.4 ฟังก์ชัน (SInt32) selectReaderNi: (NSMutableString \*)reader

Description	เลือกเครื่องอ่านบัตรสมาร์ทการ์ดที่จะใช้ในการทำงานต่อไป
Input parameter	<b>reader:</b> ชื่อของเครื่องอ่าน
Return type	SInt32
Return value	Return Code



#### 8.5 ฟังก์ชัน (SInt32) **deselectReaderNi**

Description	ยกเลิกการเลือกเครื่องอ่านบัตรสมาร์ทการ์ดที่เกิดจากฟังก์ชัน <b>selectReaderNi</b> เพื่อคืนค่าหน่วยความจำ
Input parameter	ไม่มี
Return type	SInt32
Return value	Return Code

#### 8.6 ฟังก์ชัน (SInt32) **connectCardNi**

Description	สั่งให้เชื่อมต่อกับบัตรประจำตัวประชาชน จะต้องเรียกฟังก์ชันนี้ออย่างน้อย 1 ครั้ง ก่อนใช้ฟังก์ชันอื่นในการอ่านบัตร
Input parameter	ไม่มี
Return type	SInt32
Return value	Return Code

#### 8.7 ฟังก์ชัน (SInt32) **disconnectCardNi**

Description	ยกเลิกการเชื่อมต่อกับบัตรประจำตัวประชาชนที่เกิดจากฟังก์ชัน <b>connectCardNi</b> เพื่อคืนค่าหน่วยความจำให้ระบบ
Input parameter	ไม่มี
Return type	SInt32
Return value	Return Code

#### 8.8 ฟังก์ชัน (SInt32) **getNIDNumberNi: (NSMutableString \*) strCardData**

Description	อ่านเลขประจำตัวประชาชนจากบัตร
Input parameter	ไม่มี
Output	<b>strCardData</b> : เลขประจำตัวประชาชน
Return type	SInt32
Return value	Return Code

#### 8.9 ฟังก์ชัน (SInt32) **getNIDTextNi: (NSMutableString \*) strCardText**

Description	อ่านข้อมูลตัวอักษรหน้าบัตรประจำตัวประชาชน
Input parameter	ไม่มี
Output	<b>strCardText</b> : ข้อมูลบัตรประชาชนทุกฟิลด์ แต่ละฟิลด์จะถูกคั่นด้วยเครื่องหมาย # ในลักษณะนี้ “เลขประจำตัวประชาชน#คำนำหน้านาม#ชื่อไทย#ชื่อรองไทย#นามสกุลไทย#คำนำหน้านามอังกฤษ#ชื่ออังกฤษ#ชื่อกลางอังกฤษ#นามสกุลอังกฤษ#เลขที่#หมู่บ้าน#ตรอก#ซอย#ถนน#ตำบล-แขวง#อำเภอ-เขต#จังหวัด#เพศ#วันเกิด#หน่วยงานที่ออกบัตร#วันออกบัตร#วันบัตรหมดอายุ#เลขหมายคำขอ” หากฟิลด์ใดไม่มีข้อมูลจะเป็นสตริงที่ความยาวเท่ากับศูนย์ (หรือ “”)
Return type	SInt32

Return value      Return Code

หมายเหตุ      ขณะอ่านข้อมูล ฟังก์ชันนี้จะมีการสร้าง Notification

	ข้อมูล	ความยาวข้อมูล	หมายเหตุ
1	เลขประจำตัวประชาชน	13 ตัวอักษร	“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		
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

	ข้อมูล	ความยาวข้อมูล	หมายเหตุ
22	วันบัตรหมดอายุ	8 ตัวอักษร	ตัวเลข 8 หลักติดกันในรูปแบบ YYYYMMDD YYYY คือปี พ.ศ. MM คือเดือน DD คือวันที่ กรณีเป็นบัตรตลอดชีพ จะมีข้อมูลเป็น “99999999”
23	เลขหมายคำขอ	14 ตัวอักษร	เป็นอักขระได้รูปถ่าย อาจเป็นตัวเลขหรือตัวอักษรก็ได้ บัตรประชาชนบางรุ่นจะไม่มีข้อมูลนี้

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

#### 8.10 ฟังก์ชัน (SInt32) getNIDPhotoNi: (NSMutableData \*) strCardPhoto

Description อ่านรูปภาพถ่ายใบหน้าจากบัตรประจำตัวประชาชน

Input parameter ไม่มี

Output **strCardPhoto:** ข้อมูลรูปภาพบัตรประชาชน รูปแบบไฟล์ JPEG ความยาวไม่เกิน 6144 ไบต์ ขนาดภาพ 297x355 จุด หรือ 148x178 จุด แล้วแต่รุ่นของบัตร

Return type SInt32

Return value Return Code

หมายเหตุ ขณะอ่านข้อมูล ฟังก์ชันนี้จะมีการสร้าง Notification

#### 8.11 ฟังก์ชัน (SInt32) getCardStatusNi

Description ขอสถานะการเสียบบัตรที่เครื่องอ่าน

Input parameter ไม่มี

Return type SInt32

Return value 1 คือมีบัตรเสียบอยู่ที่เครื่องอ่าน

-16 คือไม่มีบัตรเสียบอยู่

ค่าอื่น ๆ เป็น Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

#### 8.12 ฟังก์ชัน (SInt32) getRidNi: (unsigned char \*) strRid

Description ขอข้อมูลเลขประจำตัวของเครื่องอ่านบัตร (Reader ID หรือ RID)

Input parameter **strRid:** พื้นที่ว่างอย่างน้อย 256 ไบต์

Output **strRid:** ข้อมูล Reader ID ที่อ่านได้ เป็นข้อมูลแบบ Binary

Return type SInt32

Return value กรณีเป็นเลขบวก จะเป็นความยาว (จำนวนไบต์) ของ Reader ID

กรณีเป็นเลขลบ จะเป็น Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด

Note คำสั่งนี้จะเรียกใช้ได้ ในขณะที่เครื่องอ่านถูกเลือกแล้วเท่านั้น (จากคำสั่ง **selectReaderNi**)

#### 8.13 ฟังก์ชัน (SInt32) getSoftwareInfoNi: (NSMutableString \*) strData

Description ขอข้อมูลของไลบรารี

Input parameter ไม่มี

Output **strData:** ข้อมูลของไลบรารีที่ได้ ประกอบด้วย ชื่อ, รุ่นของไลบรารี และข้อมูลลิขสิทธิ์ โดยมีตัวอักษร # คั่น ในลักษณะนี้  
“ชื่อและรุ่นของไลบรารี#ข้อมูลลิขสิทธิ์” เช่น  
“NiLib 0.40#Copyright 2017-2022 R&D Computer System Co., Ltd.”

Return type SInt32

Return value Return Code

#### 8.14 ฟังก์ชัน (SInt32) getLicenseInfoNi: (NSMutableString \*) strData

Description ขอข้อมูลของใบอนุญาตที่ใช้งานอยู่

Input parameter ไม่มี

Output **strData:** ข้อมูลใบอนุญาต ประกอบด้วย ชื่อแฟ้มใบอนุญาต (License File) และวันที่ของแฟ้ม (รูปแบบ YYYY-MM-DD) และข้อมูลใบอนุญาตแบบ Virtual Licensing โดยมีตัวอักษร # คั่น ในลักษณะนี้  
“ชื่อแฟ้มใบอนุญาต#วันที่ของแฟ้มใบอนุญาต#ข้อมูลใบอนุญาตแบบ VL” เช่น  
“/var/Document/LIC/rdnidlib.dlt#2020-11-02#V|R1.02|IR01”

Return type SInt32

Return value Return Code

หมายเหตุ ต้องมีการเรียกใช้งานฟังก์ชัน **openLibNi** มาก่อนแล้วเท่านั้น จึงจะใช้งานฟังก์ชันนี้ได้

#### 8.15 ฟังก์ชัน (SInt32) updateLicenseFileNi

Description ตรวจสอบและปรับเปลี่ยนแฟ้มใบอนุญาต (License File) ให้เป็นปัจจุบัน โดยโหลดจากอินเทอร์เน็ตมาไว้ที่ตำแหน่งและชื่อตามที่ระบุไว้ในตอนเรียกใช้ฟังก์ชัน **openLibNi**

Input parameter ไม่มี

Return type SInt32

Return value 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 แต่ไม่ได้มีการอัปเดตเพราะแฟ้มที่มีอยู่ในเครื่อง iPhone, iPad เป็นรุ่นล่าสุดอยู่แล้ว  
กรณีเป็นเลขลบ แสดงว่าทำงานไม่สำเร็จ ได้ค่าเป็น Return Code มีความหมายตามค่าในตารางรหัสส่งกลับและรหัสข้อผิดพลาด เช่น

-15: ไม่สามารถติดต่ออินเทอร์เน็ตได้

-18: อัปเดตแฟ้มไบอณุญาตไม่สำเร็จ

หมายเหตุ

ให้เรียกฟังก์ชันนี้เมื่อพบว่าเครื่องอ่านที่นำมาใช้งานยังไม่ได้ลงทะเบียน หรือพบการแจ้งปัญหาเกี่ยวกับแฟ้มไบอณุญาต เช่น ไม่พบแฟ้มไบอณุญาต (Return Code = -12) หรือ ไบอณุญาตใช้ไม่ได้ (Return Code = -2) เพื่อให้ระบบทำการดาวน์โหลดแฟ้มไบอณุญาตรุ่นล่าสุดจากอินเทอร์เน็ตมาใช้งานแทน

หลังจากเรียกฟังก์ชันนี้แล้ว ควรปิดการใช้งาน NiLib ทั้งหมด (**disconnectCardNi**, **deselectReaderNi**, **closeLibNi**) แล้วเริ่มสั่ง **openLibNi** ใหม่ตั้งแต่ต้น ก่อนจะเรียกใช้งานฟังก์ชันนี้ ควรเปิดให้อุปกรณ์ iDevice เชื่อมต่อกับเครือข่ายอินเทอร์เน็ตให้ได้ ก่อน มิฉะนั้น ฟังก์ชันนี้จะแจ้งคืนค่า -15 (ไม่สามารถติดต่ออินเทอร์เน็ตได้) หรือ -18 (อัปเดตแฟ้มไบอณุญาตไม่สำเร็จ)

สำหรับการใช้งานกับเครื่องอ่านแบบ Virtual Licensing ไม่จำเป็นต้องเรียกใช้งานฟังก์ชันนี้ เพราะเครื่องอ่านแบบ Virtual Licensing ไม่มีการใช้แฟ้มไบอณุญาต จะมีแฟ้มหรือไม่ได้

#### 8.16 ฟังก์ชัน (UInt32) getContextNi

Description	อ่านค่า Context Handle ของ NiLib ซึ่งเกิดมาจากที่มีการใช้ฟังก์ชัน SCardEstablishContext ภายใน NiLib ผู้พัฒนาสามารถนำค่า Context Handle นี้ไปใช้งานกับฟังก์ชันหรือคำสั่งต่าง ๆ ของ PC/SC ได้ เช่น SCardIsValidContext , SCardListReaders ผู้พัฒนาไม่จำเป็นต้องเรียกใช้ฟังก์ชันนี้เลย หากไม่มีความต้องการใช้คำสั่งของ PC/SC
Input parameter	ไม่มี
Return type	UInt32
Return value	Context Handle
หมายเหตุ	ค่า Context Handle ของ NiLib จะมีหลังจากเรียกใช้ฟังก์ชัน <b>openLibNi</b> แล้วเท่านั้น

#### 8.17 ฟังก์ชัน (UInt32) getCardHandleNi

Description	อ่านค่า Handle ของเครื่องอ่านบัตรที่ถูกเชื่อมต่อกับฟังก์ชัน <b>connectCardNi</b> ไว้แล้ว ค่า Handle นี้จะเป็นค่าเดียวกับค่า Handle ของฟังก์ชัน SCardConnect ผู้พัฒนาสามารถนำค่านี้ไปใช้งานกับฟังก์ชันหรือคำสั่งต่าง ๆ ของ PC/SC ได้ เช่น SCardStatus, SCardGetAttrib , SCardTransmit
Input parameter	ไม่มี
Return type	UInt32
Return value	Handle
หมายเหตุ	ค่า Handle นี้ จะมีหลังจากเรียกใช้ฟังก์ชัน <b>connectCardNi</b> แล้วเท่านั้น

#### 8.18 ฟังก์ชัน (SInt32) attachContextNi: (UInt32)\_hContextHandle

Description	ผูกค่า (binding) ของ Context Handle ที่ได้จากฟังก์ชัน SCardEstablishContext (ของ PC/SC) กับ NiLib กรณีโปรแกรมมีการสร้างค่า Context Handle เองก่อนเรียกใช้ฟังก์ชัน <b>openLibNi</b> หรือ กรณีหลังจากการเรียกฟังก์ชัน <b>DetachContext</b>
-------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Input parameter **\_hContextHandle**: คือค่าของ **Context Handle** ที่ได้จากฟังก์ชัน **SCardEstablishContext** ของ PC/SC

Return type **SInt32**

Return value **Return Code**

Example `SCARDCONTEXT hContextHandle;  
SCardEstablishContext(SCARD_SCOPE_SYSTEM,NULL,NULL, & hContextHandle);  
[ mNiOS attachContextNi:hContextHandle ];  
int nres = [ mNiOS openLibNi: path ];`

#### 8.19 ฟังก์ชัน (**SInt32**) **detachContextNi**

Description ยกเลิกการผูกค่า (binding) ของ Context Handle ที่ได้จากฟังก์ชัน **SCardEstablishContext** (ของ PC/SC) กับ NiLib

Input parameter ไม่มี

Return type **SInt32**

Return value **Return Code**

Example `- (void)applicationDidEnterBackground:(UIApplication *)application  
{  
 FtDidEnterBackground(1);  
 SCARDCONTEXT cardContext = [nios getContextNi];  
 if(cardContext!=0)  
 {  
 [nios detachContextNi];  
 SCardReleaseContext( cardContext);  
 }  
}  
  
- (void)applicationDidBecomeActive:(UIApplication *)application  
{  
 SCARDCONTEXT cardContext;  
 SCardEstablishContext(SCARD_SCOPE_SYSTEM,NULL,NULL,&cardContext);  
 [nios attachContextNi: cardContext ];  
}`

#### 8.20 ฟังก์ชัน (**SInt32**) **getReaderInfoNi: (NSMutableString \*)strData**

Description ขอข้อมูลของเครื่องอ่านที่กำลังเชื่อมต่ออยู่

Input parameter ไม่มี

Output **strData**: ข้อมูลของเครื่องอ่านที่ขอได้ ประกอบด้วย ชนิดของการเชื่อมต่อ, ชนิดของ License และชื่อรุ่นของเครื่องอ่าน โดยมีตัวอักษร # คั่น ในลักษณะนี้

*“Interface#LicenseType#Model#Firmware Version”*

Interface	USB: เป็นเครื่องอ่านแบบ USB
	LTN: เป็นเครื่องอ่านแบบ Lightning
	BLT: เป็นเครื่องอ่านแบบบลูทูธ 3

LicenseType	BLE: เป็นเครื่องอ่านบลูทูธแบบ BLE VL: เป็นเครื่องอ่านที่ใช้ใบอนุญาตแบบ Virtual Licensing FL: เป็นเครื่องอ่านที่ใช้ใบอนุญาตแบบ File Licensing NL: เป็นเครื่องอ่านที่ไม่มีใบอนุญาต (Non-Licensing) UL: เป็นเครื่องอ่านที่ยังไม่ทราบว่ามีใบอนุญาตหรือไม่ (Unknown Licensing)
Model	ชื่อรุ่นของเครื่องอ่าน
Firmware Version	รุ่นเฟิร์มแวร์ของเครื่องอ่าน ตัวอย่างเช่น “BLE#VL#TDAi301BL#2.22”

Return type SInt32

Return value Return Code

หมายเหตุ ต้องมีการเรียกใช้งานฟังก์ชัน **selectReaderNi** มาก่อนแล้วเท่านั้น จึงจะใช้งานฟังก์ชันนี้ได้

#### 8.21 ฟังก์ชัน (SInt32) getATextNi: (NSMutableString \*) strAText

Description อ่านข้อมูลตัวอักษรทั้งหมดของบัตรประจำตัวประชาชน

Input parameter ไม่มี

Output **strAText:** ข้อมูลบัตรประจำตัวประชาชนที่อ่านได้ทั้งหมดทุกรายการ แต่ละรายการจะถูกค้นด้วยเครื่องหมาย # ในลักษณะนี้  
“เลขประจำตัวประชาชน#คำนำหน้านามไทย#ชื่อไทย#ชื่อรองไทย#นามสกุลไทย#คำนำหน้านามอังกฤษ#ชื่ออังกฤษ#ชื่อกลางอังกฤษ#นามสกุลอังกฤษ#เลขที่#หมู่ที่#ตรอก#ซอย#ถนน#ตำบลแขวง#อำเภอเขต#จังหวัด#เพศ#วันเกิด#หน่วยงานที่ออกบัตร#วันออกบัตร #วันบัตรหมดอายุ#เลขหมายคำขอ#หมายเลขบัตร/คำร้อง#รหัสผู้ออกบัตร#รุ่นของโครงสร้าง#รหัสประเภทบัตร#รหัสคำนำหน้านาม”

Return type SInt32

Return value Return Code

หมายเหตุ ขณะอ่านข้อมูล ฟังก์ชันนี้จะมีการสร้าง Notification

	ข้อมูล	ความยาวข้อมูล	หมายเหตุ
1-23	ข้อมูลหน้าบัตรประจำตัวประชาชน		ข้อมูล 23 รายการแรกจะเหมือนกับข้อมูลที่ได้จากฟังก์ชัน getNIDTextNi ดูรายละเอียดได้จากในฟังก์ชัน getNIDTextNi
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”

	ข้อมูล	ความยาวข้อมูล	หมายเหตุ
26	รุ่นของโครงสร้าง	4 ตัวอักษร	หมายเลขรุ่นของโครงสร้างข้อมูลบัตร เช่น “0003” หรือ “0004”
27	รหัสประเภทบัตร	2 ตัวอักษร	“01” 01 คือ บัตรประจำตัวประชาชน
28	รหัสคำนำหน้านาม	1 ตัวอักษร	Space = ไม่มีข้อมูล “0” = ปกติ “1” = ยศทหารเรือ (ร.น.) “2” = สมณศักดิ์

### ตัวอย่าง

“3650800011234#นาย#สมชาย##สามรักสกุล#Mr.#Somchai##Samruksakul#123/45#หมู่ที่ 12####ตำบลบึงพระ#อำเภอเมืองพิษณุโลก#จังหวัดพิษณุโลก#1#25240119#เทศบาลนครพิษณุโลก#25521026#25590118#12345678901234#12345678901/12345678#1234567890123#0004#01#0”

**หมายเหตุ** สามารถใช้งานฟังก์ชันนี้แทน **getNIDTextNi** ได้ ต่างกันที่ **getNIDTextNi** คืนข้อมูล 23 รายการแรก ส่วน **getATextNi** จะเพิ่มข้อมูลต่อท้ายให้อีก 5 รายการ รวมเป็น 28 รายการ จำนวนรายการขณะนี้อยู่ที่ 28 รายการ แต่ในอนาคตอาจปรับเปลี่ยนมากขึ้นได้ ในการเขียนโปรแกรมควรเผื่อกรณีไว้ด้วย

### 8.22 ฟังก์ชัน (void) setReaderTypeNi: (SInt32) readerType

Description	กำหนดประเภทของเครื่องอ่านที่ต้องการให้ทำงานร่วมกับ NiLib โดยฟังก์ชันนี้จำเป็นต้องถูกเรียกใช้งานเป็นลำดับแรกเพียงครั้งเดียวก่อนการเรียกใช้ฟังก์ชันต่างๆของ NiLib และก่อนคำสั่ง SCardEstablishContext
Input parameter	<b>readerType</b> ประเภทเครื่องอ่านที่ต้องการทำงานร่วมกับ NiLib โดยส่งเป็นค่าคงที่ตามที่ถูกกำหนดไว้แล้วในไฟล์ .h <b>READERTYPE_BLE</b> เลือกเครื่องอ่านเป็น BLE (Bluetooth 4) <b>READERTYPE_BUL</b> เลือกเครื่องอ่านเป็น Bluetooth 3.0, USB และ Lightning
Output	ไม่มี
Return type	ไม่มี
Return value	ไม่มี

### 8.23 ฟังก์ชัน (SInt32) scanReaderListBleNi: (NSMutableArray \*) strReaderList : (SInt32) timeout

Description	ให้ค้นหาเครื่องอ่านบัตรสมาร์ทการ์ดประเภท BLE ที่เปิดอยู่พร้อมใช้งาน และคืนชื่อเครื่องอ่านทั้งหมดที่พบ
Input parameter	<b>timeout</b> คือระยะเวลาที่กำหนดให้ระบบค้นหาเครื่องอ่านแบบ BLE มีหน่วยเป็นวินาที



Output	<b>strReaderList</b> : จะคืนชื่อของเครื่องอ่านบาร์โค้ดแต่ละเครื่องที่พบในระบบไว้ใน Array นี้ โดยเครื่องอ่านที่พบเครื่องแรกจะอยู่ใน <b>strReaderList[0]</b> หากมีเครื่องที่สองก็จะอยู่ใน <b>strReaderList[1]</b> และต่อ ๆ ไป จนครบทุกเครื่องอ่าน ชื่อของเครื่องอ่านเป็นข้อมูลชนิด NSMutableString
Return type	SInt32
Return value	จำนวนเครื่องอ่านที่ต่ออยู่ (กรณีเป็นเลขมากกว่า 0) Return Code (กรณีเป็นเลขลบ)
หมายเหตุ	ขณะค้นหาเครื่องอ่าน ฟังก์ชันนี้จะมีการสร้าง Notification เมื่อพบเครื่องอ่านแต่ละเครื่อง

#### 8.24 ฟังก์ชัน (SInt32) stopReaderListBleNi

Description	ยกเลิกและหยุดการค้นหาเครื่องอ่านแบบ BLE ของฟังก์ชัน <b>scanReaderListBleNi</b>
Input parameter	ไม่มี
Output	ไม่มี
Return type	SInt32
Return value	Return Code

### 8.25 รหัสส่งกลับและรหัสข้อผิดพลาด (Return Code) ที่ได้จากฟังก์ชันต่างๆ

Return Code	Name	ความหมาย
0	NI_SUCCESS	ทำงานได้สำเร็จเสร็จสิ้น
-1	NI_INTERNAL_ERROR	เกิดข้อผิดพลาดภายในระบบ
-2	NI_INVALID_LICENSE	เครื่องอ่านนี้ใช้ไม่ได้เพราะไม่มีทะเบียนในใบอนุญาต
-3	NI_READER_NOT_FOUND	ไม่พบเครื่องอ่านบัตร
-4	NI_CONNECTION_ERROR	ไม่สามารถติดต่อกับบัตรได้
-5	NI_GET_PHOTO_ERROR	ไม่สามารถอ่านรูปภาพถ่ายใบหน้าได้
-6	NI_GET_TEXT_ERROR	ไม่สามารถอ่านข้อมูลตัวอักษรได้
-7	NI_INVALID_CARD	บัตรที่อ่านไม่ใช่บัตรประชาชน
-8	NI_UNKNOWN_CARD_VERSION	ไม่รองรับการใช้งานกับบัตรประชาชนรุ่นนี้
-9	NI_DISCONNECTION_ERROR	ไม่สามารถยกเลิกการเชื่อมต่อกับเครื่องอ่านบัตรได้
-10	NI_INIT_ERROR	ยังไม่ได้เรียกใช้งานฟังก์ชัน <b>openLibNi</b> เพื่อเริ่มต้นการทำงาน
-11	NI_READER_NOT_SUPPORTED	ไม่รองรับการใช้งานกับเครื่องอ่านเครื่องนี้ หรือไม่พบเครื่องอ่านบัตร
-12	NI_LICENSE_FILE_ERROR	ไม่พบแฟ้มใบอนุญาต หรือแฟ้มใบอนุญาตเสียหาย
-13	NI_PARAMETER_ERROR	พารามิเตอร์หรือตัวแปรผิดพลาด
-15	NI_INTERNET_ERROR	ไม่สามารถติดต่ออินเทอร์เน็ตได้
-16	NI_CARD_NOT_FOUND	ไม่พบบัตรในเครื่องอ่าน
-18	NI_LICENSE_UPDATE_ERROR	อัปเดตแฟ้มใบอนุญาตไม่สำเร็จ

#### หมายเหตุ :

กรณีรหัสส่งกลับและรหัสข้อผิดพลาด (Return Code) เป็นค่าอื่นที่ไม่ได้อยู่ในช่วง 0 ถึง -18 จะเป็นค่าผิดพลาดของไลบรารีเครื่องอ่านบัตรสมาร์ทการ์ด CCID ของ Feitan Technologies (CCID Lib) ซึ่งจะมี ความหมายตามที่แสดงในแฟ้มใน wintypes.h

## 9. ตัวอย่างการใช้งาน NiLib และการพัฒนาแอปพลิเคชัน

### 9.1 ขั้นตอนการเรียกใช้ฟังก์ชันต่าง ๆ ของ NiLib สำหรับเครื่องอ่านแบบ BLE ให้ทำตามลำดับดังนี้

- เรียก **setReaderTypeNi (READERTYPE\_BLE)** เพื่อกำหนดประเภทของเครื่องอ่าน BLE ให้ทำงานร่วมกับ NiLib
- เรียก **scanReaderListBleNi** เพื่อค้นหาเครื่องอ่านบัตรประเภท BLE
  - กรณีที่ต้องการให้ระบบหยุดการค้นหาก่อนหมดเวลาให้เรียกใช้ฟังก์ชัน **stopReaderListBleNi**
- สร้างหน้าจอ Pop-up เพื่อรับ Notification จากคำสั่ง **scanReaderListBleNi** สำหรับแสดงรายชื่อเครื่องอ่านที่ค้นพบ และให้ผู้ใช้เลือก โดยอาจทำตามตัวอย่างคลาส **scanBleViewCtrl** ในไฟล์ **scanBleViewCtrl.swift** หรือจะพัฒนาหน้าจอ Pop-up นี้เองในรูปแบบอื่นก็ได้
- เรียก **openLibNi** เพื่อเลือกโหมดการทำงาน โดยต้องเลือกว่าจะให้ทำงานแบบ VL อย่างเดียว หรือจะให้ทำงานแบบผสม FL+VL (ปกติแนะนำให้ใช้แบบ FL+VL)
  - ถ้าใช้โหมด FL+VL แล้วพบปัญหาเกี่ยวกับแฟ้มใบอนุญาต เช่น ไม่พบแฟ้มใบอนุญาต หรือแฟ้มใบอนุญาตเสียหาย (Return Code -2 หรือ -12) ให้เรียก **updateLicenseFileNi** เพื่อดาวน์โหลดแฟ้มใบอนุญาตลงมาใหม่ แล้วจึงสั่ง **openLibNi** ใหม่อีกครั้ง
  - เครื่องอ่านแบบ BLE ควรเลือกโหมด VL เท่านั้น
- เรียก **getReaderListNi** เพื่อขอรายชื่อเครื่องอ่านบัตรสมาร์ทการ์ดที่ต่ออยู่กับระบบ
  - เครื่องอ่านแบบ BLE ไม่ต้องเรียกใช้ฟังก์ชันนี้
- เรียก **selectReaderNi** เพื่อกำหนดเครื่องอ่านที่จะใช้ในการอ่านบัตรประชาชน ตามชื่อที่อ่านมาได้จากคำสั่ง **scanReaderListNi** หรือคำสั่ง **getReaderListNi**
  - หากพบปัญหาเกี่ยวกับแฟ้มใบอนุญาต เช่น เครื่องอ่านใช้ไม่ได้เพราะไม่มีทะเบียนในใบอนุญาต (Return Code -2) หรือแฟ้มใบอนุญาตเสียหาย (Return Code -12) ให้เรียก **updateLicenseFileNi** เพื่ออัปเดตแฟ้มใบอนุญาตให้เป็นรุ่นล่าสุด แล้วปิดการใช้งาน NiLib ทั้งหมด (สั่ง **deselectReaderNi** และ **closeLibNi**) แล้วเริ่มสั่ง **openLibNi** ใหม่ตั้งแต่ต้น
    - เครื่องอ่านแบบ BLE ไม่ต้องทำตามข้อความข้างต้น และไม่ต้องสั่ง **updateLicenseFileNi** เพราะไม่มีการใช้แฟ้มใบอนุญาต
- เรียก **connectCardNi** เพื่อเริ่มติดต่อกับบัตรประชาชน และตรวจสอบว่ามีการเสียบบัตรที่เครื่องอ่านหรือไม่
- อ่านข้อมูลจากบัตรประชาชน โดยใช้ ฟังก์ชันต่อไปนี้ ตามที่ต้องการ
  - **getNIDNumberNi** อ่านเฉพาะเลขประจำตัวประชาชน
  - **getNIDTextNi** หรือ **getATextNi** อ่านข้อมูลตัวอักษรของบัตรประจำตัวประชาชนทั้งหมด ยกเว้นรูปภาพถ่าย
  - **getNIDPhotoNi** อ่านรูปภาพถ่ายใบหน้าของบัตร
- เมื่ออ่านข้อมูลจากบัตรประชาชนครบแล้ว ให้สั่ง **disconnectCardNi** เพื่อยกเลิกการติดต่อกับบัตรประชาชนฉบับนี้

- หากต้องการอ่านบัตรประชาชนฉบับใหม่ ให้กลับไปขั้นตอนการเรียกใช้ **connectCardNi** อีกครั้งหนึ่ง
- กรณีต้องการเปลี่ยนไปเลือกเครื่องอ่านบัตรสมาร์ตการ์ดเครื่องอื่น หรือต้องการอ่านรายชื่อเครื่องอ่านใหม่ ให้สั่งยกเลิกการเลือกเครื่องอ่านเดิมก่อน โดยเรียก **deselectReaderNi** แล้วค่อยเรียกใช้คำสั่ง **scanReaderListNi** หรือคำสั่ง **getReaderListNi** และเรียก **selectReaderNi** อีกครั้ง
- กรณีต้องการยกเลิกการใช้งานไลบรารีหรือต้องการรีเซ็ตไลบรารีใหม่ทั้งหมดเพื่อจะเริ่มต้นการอ่านใหม่ตั้งแต่ต้น ให้เรียก **closeLibNi** เพื่อปิดการใช้งานและคืนหน่วยความจำให้กับระบบ
- สำหรับฟังก์ชัน **getSoftwareInfoNi** และ **getLicenseInfoNi** จะเรียกใช้งานหรือไม่ก็ได้

## 9.2 การพัฒนาแอปพลิเคชันด้วย Objective-C และ Swift ใน Xcode

### 9.2.1 ให้แก้ไขไฟล์ Info.plist ดังนี้

- สำหรับเครื่องอ่านแบบ BLE ให้เพิ่มบรรทัดต่อไปนี้ ในแฟ้ม Info.plist ด้วย

```
<key>NSBluetoothAlwaysUsageDescription</key>
<string>Our app uses Bluetooth for Thai ID card reading</string>
```

```
<key>NSBluetoothPeripheralUsageDescription</key>
<string>Our app uses Bluetooth for Thai ID card reading</string>
```

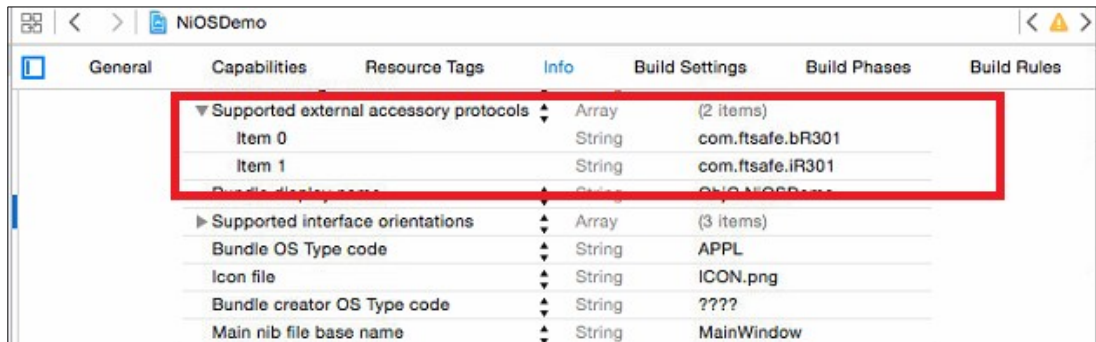
- การกำหนดค่า UISupportedExternalAccessoryProtocols ในแฟ้ม Info.plist กรณีมีการใช้งานเครื่องอ่านรุ่น TDAi301U8A หรือ TDAi301VC หรือ TDAi301BT ให้เพิ่ม Protocol ที่ใช้งานลงใน array (เครื่องอ่านแบบ BLE ไม่ต้องทำ) เช่น

```
<key>UISupportedExternalAccessoryProtocols</key>
<array>
  <string>com.ftsafe.bR301</string>
  <string>com.ftsafe.iR301</string>
</array>
```

### หมายเหตุ

- ให้เพิ่มเฉพาะบรรทัด <string>com.ftsafe.bR301</string> เมื่อต้องการใช้งานกับเครื่องอ่านรุ่น TDAi301BT
- ให้เพิ่มเฉพาะบรรทัด <string>com.ftsafe.iR301</string> เมื่อต้องการใช้งานกับเครื่องอ่านแบบเสียบพอร์ต USB หรือ Lightning รุ่น TDAi301VC และ TDAi301U8A
- หรือให้เพิ่มทั้ง 2 บรรทัดหากจะใช้เครื่องอ่านทั้ง 2 แบบ
- หากมีการเพิ่มบรรทัด UISupportedExternalAccessoryProtocols เหล่านี้แล้ว จะต้องขอ PPID จากทางบริษัท เพื่อให้สามารถอัปโหลดขึ้น Apple App Store ได้

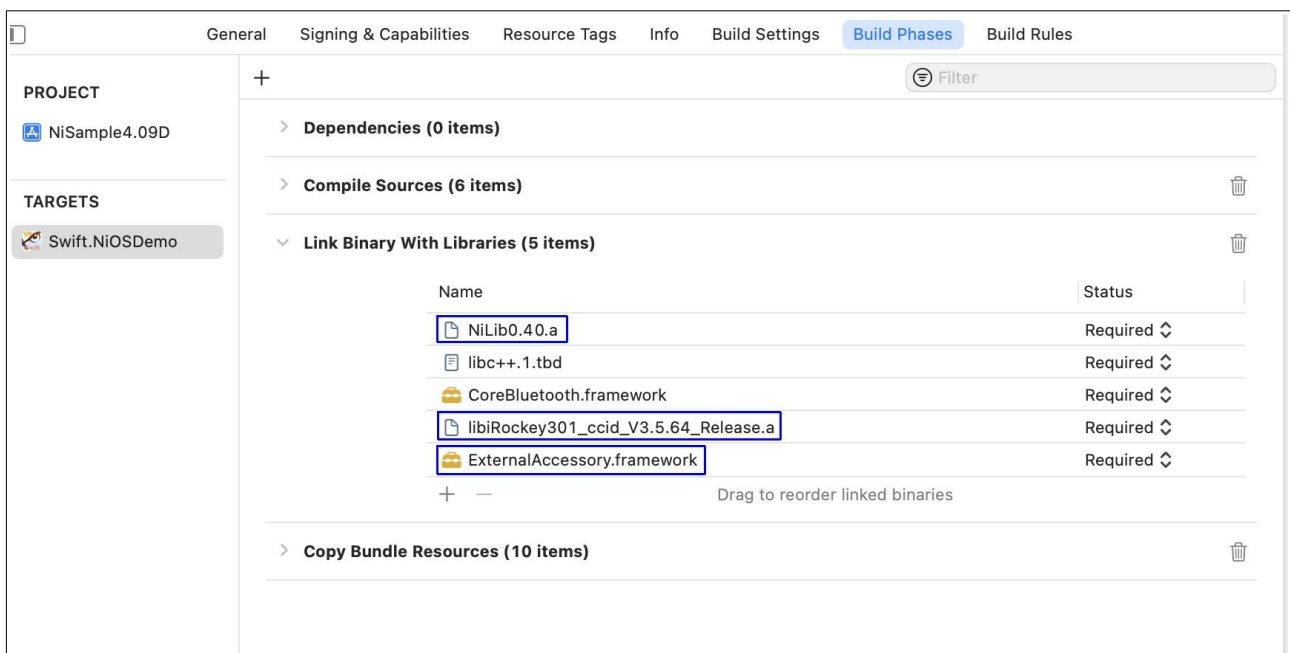
- การเพิ่ม UISupportedExternalAccessoryProtocols อาจทำผ่าน Xcode configuration ได้ดังนี้



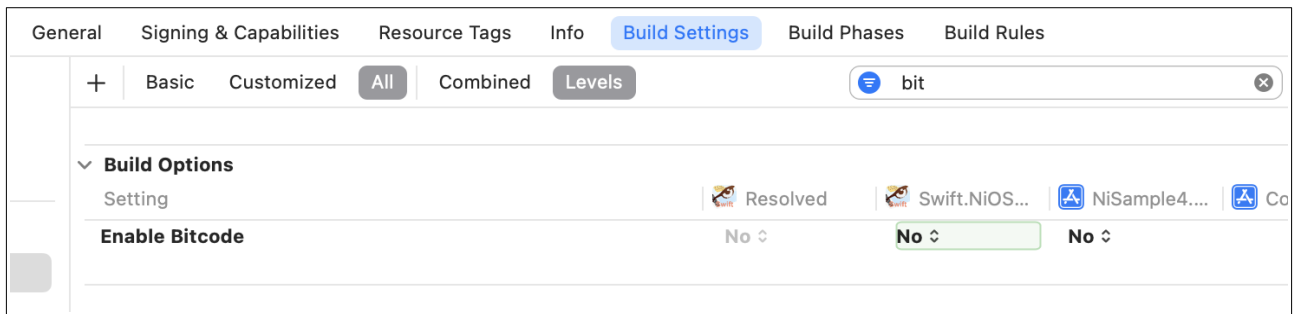
- กรณีมีการใช้งานเครื่องอ่าน BLE เพียงแบบเดียว (TDAi301BL, TDAi301BLM) ให้ไม่ต้องเพิ่มบรรทัด UISupportedExternalAccessoryProtocols เหล่านี้เลย หากเพิ่มไว้แล้วให้ลบออก (ไม่เช่นนั้น จะถูกบังคับในต้องขอ PPID ก่อนจึงจะสามารถนำแอปพลิเคชันขึ้น Apple App Store ได้)

#### 9.2.2 ให้เพิ่มรายการของไลบรารีให้กับ Project ดังนี้

- ไลบรารีของ NiLib (NiLibX.xx.a)
- ไลบรารี CCID ของเครื่องอ่านบัตร (libiRockey301\_ccid\_Vx.xx.x\_a)
- ไลบรารี ExternalAccessory.framework ของ iDevice
  - หากใช้งานเครื่องอ่านแบบ BLE เพียงแบบเดียว (TDAi301BL, TDAi301BLM) ไม่ต้องเพิ่มไลบรารีตัวนี้



### 9.2.3 ให้กำหนดค่าให้กับ Compiler Option : Enable Bitcode = NO



### 9.2.4 กรณีเลือกการทำงานแบบ FL+VL ให้วางเพิ่มใบอนุญาต rdnidlib.dlt ไว้ที่โฟลเดอร์

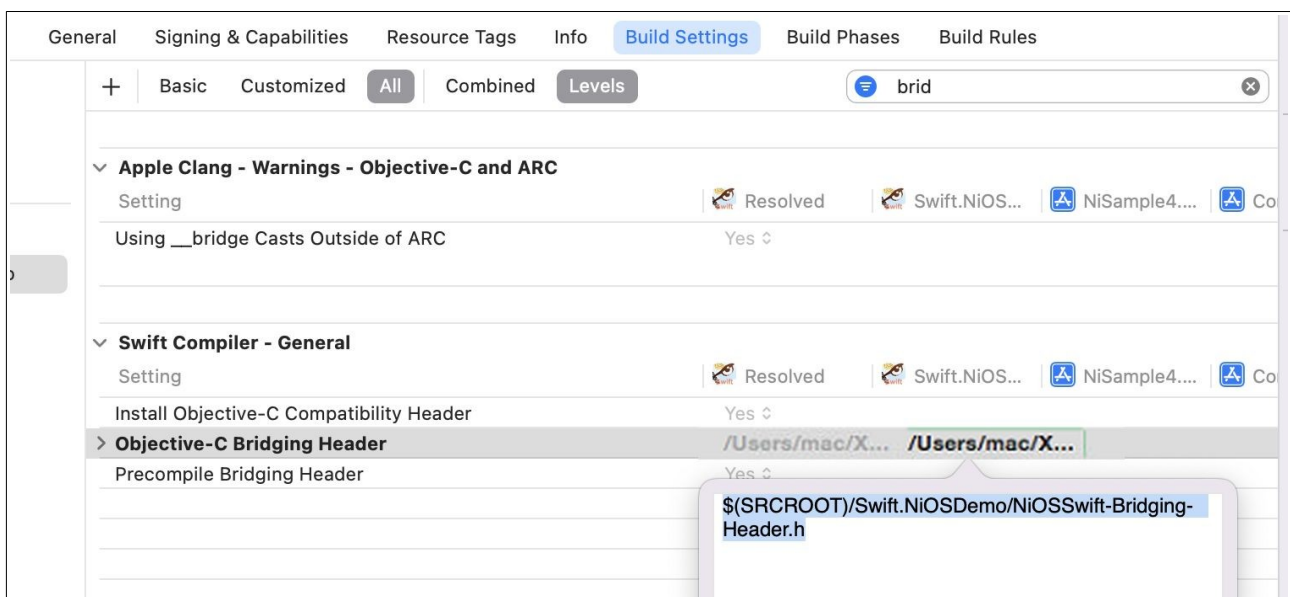
Document/LIC/ ของอุปกรณ์ iDevice ที่จะใช้ทดสอบด้วย

- สำหรับแอปพลิเคชันตัวอย่าง (NiSample) จะมีกลไกในการดาวน์โหลดเพิ่ม rdnidlib.dlt จากอินเทอร์เน็ตแบบอัตโนมัติไว้ให้แล้ว ดังนั้นอาจไม่ต้องวางเพิ่มใบอนุญาตไว้ก่อนก็ได้
- เครื่องอ่านแบบ BLE ไม่ต้องทำตามข้อความข้างต้น เพราะทำงานแบบ VL เท่านั้น

### 9.2.5 เฉพาะ Swift ให้สร้างเพิ่ม Bridging ให้กับ Swift เป็นชื่อที่ต้องการ เช่น NiOSSwift-Bridging-Header.h

```
// include ccidlib
#import "CCIDLib/include/winscard.h"
#import "CCIDLib/include/ReaderInterface.h"
#import "CCIDLib/include/ft301u.h"
// include NiLib
#include "NiLib/include/NiOS.h"
```

- แล้วกำหนดชื่อไฟล์ของเพิ่ม Bridging ให้กับ Project ดังนี้

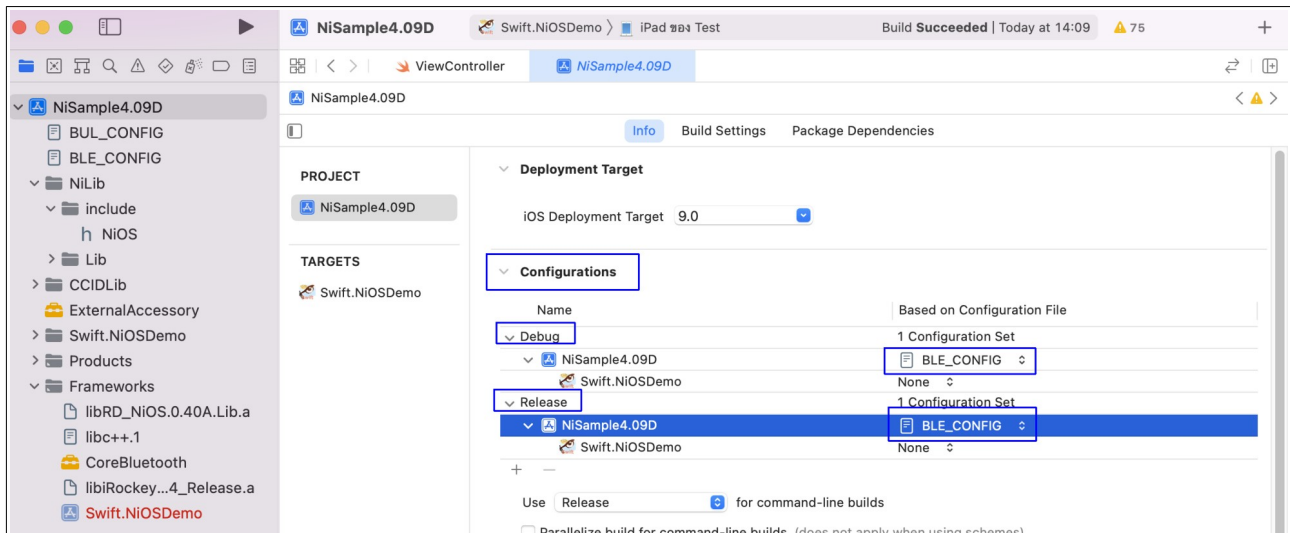


## 9.2.6 ตัวอย่างโปรแกรมสำหรับ Swift ให้ศึกษาได้จากไฟล์ในโฟลเดอร์

NiBLESample\_Swift\_xx.xx ในชุดพัฒนา TDi NiBLE SDK

## 9.3 การกำหนดให้ใช้เครื่องอ่านเป็นแบบ BLE ในโปรเจกต์

NiLib รุ่นนี้รองรับการใช้งานเครื่องอ่าน BLE เป็นหลัก โดยการกำหนดค่าไว้ในไฟล์ Configurations ของโปรเจกต์ เป็นไฟล์ BLE\_CONFIG.xcconfig จึงขอให้ผู้พัฒนา กำหนด Configurations ของโปรเจกต์ดังนี้



### หมายเหตุ

- ชุดพัฒนา NiBLE นี้ออกแบบมาให้เหมาะสำหรับการพัฒนาแอปพลิเคชันที่มีการใช้งานเครื่องอ่านแบบ BLE เพียงรุ่นเดียวเป็นหลัก แต่หากต้องการ นักพัฒนาสามารถ Build แอปพลิเคชันสำหรับการใช้งานกับเครื่องอ่านรุ่นเดิมที่เป็น Bluetooth 3, USB-C และ Lightning (หรือเรียกย่อ ๆ ว่าเครื่องอ่าน BUL) ได้จาก Source Code ตัวอย่างนี้ได้ด้วย โดยการเลือกไฟล์ Configurations ของโปรเจกต์ เป็นไฟล์ BUL\_CONFIG.xcconfig แทน
- หรือหากต้องการใช้เฉพาะเครื่องอ่านแบบเสียบ USB-C และ Lightning เท่านั้น ไม่ใช่เครื่องอ่านแบบ Bluetooth และ BLE เลย ให้เลือกไฟล์ Configurations ของโปรเจกต์ เป็นไฟล์ UL\_CONFIG.xcconfig แทน

## 9.4 การใช้งานระบบ Notification ของ NiLib

NiLib มีความสามารถในการสร้าง Event ผ่านทางระบบ Notification Center ของ iOS/iPadOS เพื่อแจ้งข้อมูลต่าง ๆ ให้กับระบบได้ เช่นแจ้งความคืบหน้าในการอ่านข้อมูล หรือแจ้งว่าขณะนี้ มี Event การเสียบ หรือถอดบัตร ซึ่งผู้พัฒนาสามารถนำเอาข้อมูลเหล่านี้ไปแสดงให้กับผู้ใช้ทราบว่าแอปพลิเคชันกำลังอ่านข้อมูลอยู่

### 9.4.1 ขั้นตอนการเขียนระบบ Notification

- ให้ลงทะเบียน (Registration) notification id ด้วยชื่อ NiOS\_NotifyMessage ตามที่นิยามไว้ในแฟ้ม NiOS.h

#### ตัวอย่าง ภาษา Swift

```
NotificationCenter.default.addObserver(self,
    selector: #selector(ViewController.OnRDNID_NotifyMessage(_:)),
    name: NSNotification.Name(rawValue: NiOS_NotifyMessage), object: nil)
```

#### ตัวอย่าง ภาษา Objective-C

```
[[NSNotificationCenter defaultCenter] addObserver:self
    selector:@selector(OnRDNID_NotifyMessage:)
    name: NiOS_NotifyMessage object:NULL];
```

โดยกำหนดให้ใช้ฟังก์ชัน **OnRDNID\_NotifyMessage**: ทำหน้าที่รับค่า Notification จาก NiLib

โดย **ViewController.OnRDNID\_NotifyMessage** คือ ฟังก์ชันบนคลาส ViewController ทำหน้าที่รับค่า Notification จาก NiLib

- สร้างฟังก์ชันสำหรับรับ Notification จากระบบ Notification Center ที่มีโครงสร้างฟังก์ชันดังนี้

#### ตัวอย่าง ภาษา Swift

```
func onRDNID_NotifyMessage(_ parmNotification: Notification) {
    let userInfo: [AnyHashable: Any]? = parmNotification.userInfo
    let NotifyId: String? = (userInfo?["NotifyId"] as? String)
    let MessageType: String? = (userInfo?["MessageType"] as? String)
    let Caller: String? = (userInfo?["Caller"] as? String)
    let perc = (userInfo?["Percent"] as? NSNumber)
    let arg: NSObject? = (userInfo?["ArgData"] as? NSObject)
    let floatValue = CFLOAT(perc)
    return
}
```

#### ตัวอย่าง ภาษา Objective-C

```
(void) OnRDNID_NotifyMessage:(NSNotification *) parmNotification {
    NSDictionary *userInfo = parmNotification.userInfo;
    NSString *NotifyId = [userInfo objectForKey:@"NotifyId" ];
    NSString *MessageType = [userInfo objectForKey:@"MessageType" ];
    NSString *Caller = [userInfo objectForKey:@"Caller" ];
    NSNumber *perc = [userInfo objectForKey:@"Percent" ];
    NSObject *arg = [userInfo objectForKey:@"ArgData" ];

    float floatValue = [perc floatValue];
    return ;
}
```

โดย

**OnRDNID\_NotifyMessage**: คือ ฟังก์ชันที่รับค่า Notification จากระบบ iOS



**parmNotification**

คือ ค่า notification ที่ระบบ Notification Center ของ iOS ส่งให้  
โดยการอ่านข้อมูลของ NiLib ดำเนินการผ่านพรอเพอร์ตี้

**userInfo****userInfo**

มีชนิดข้อมูลเป็น NSDictionary ซึ่งมีรายละเอียดดังนี้

โครงสร้าง NSDictionary บน <b>userInfo</b> ของฟังก์ชัน <b>getNIDTextNi, getATextNi และ getNIDPhotoNi</b>		
ชื่อฟิลด์ (Field name)	ความหมาย	ประเภทข้อมูล (Data Type)
NotifyId	ค่า notification id มีค่า คือ NiOS_NotifyMessage	NSString
MessageType	ประเภทของ message ค่าคือ ReadSmartCardIPercent	NSString
Caller	ชื่อฟังก์ชันที่อ่านข้อมูล ได้แก่ getNIDTextNi หรือ getATextNi หรือ getNIDPhotoNi	NSString
Percent	ค่าปัจจุบันของการอ่านข้อมูล มีค่าตั้งแต่ 1 ถึง 100 (1% ถึง 100%)	NSNumber
ArgData	ข้อมูลที่อ่านปัจจุบัน	NSObject

โครงสร้าง NSDictionary บน <b>userInfo</b> ของฟังก์ชัน <b>scanReaderListBleNi</b>		
ชื่อฟิลด์ (Field name)	ความหมาย	ประเภทข้อมูล (Data Type)
NotifyId	ค่า notification id มีค่า คือ NiOS_NotifyMessage	NSString
Caller	ชื่อฟังก์ชันที่ส่งชื่อเครื่องอ่าน scanReaderListBleNi	NSString
ArgData	ชื่อเครื่องอ่าน BLE ที่ระบบค้นหพบ	NSObject

โครงสร้าง NSDictionary บน <b>userInfo</b> ของ <b>Library</b>		
ชื่อฟิลด์ (Field name)	ความหมาย	ประเภทข้อมูล (Data Type)
NotifyId	ค่า notification id มีค่า คือ NiOS_NotifyMessage	NSString
Caller	ชื่อข้อมูล (สถานะของบัตร) คือ CardStatus	NSString
Status	สถานะของบัตรที่เครื่องอ่าน 0: ไม่มีบัตรเสียบที่เครื่องอ่าน 1: มีบัตรเสียบที่เครื่องอ่าน	NSObject

โครงสร้าง NSDictionary บน userInfo ของ Library		
ชื่อฟิลด์ (Field name)	ความหมาย	ประเภทข้อมูล (Data Type)
NotifyId	ค่า notification id มีค่า คือ NiOS_NotifyMessage	NSString
Caller	ชื่อข้อมูล (สถานะของเครื่องอ่าน) คือ ReaderStatus	NSString
Status	สถานะของเครื่องอ่าน 0: ระบบไม่เชื่อมต่อกับเครื่องอ่าน 1: ระบบกำลังเชื่อมต่อกับเครื่องอ่าน	NSObject

### ตัวอย่าง ภาษา Swift

```

@objc func updateProgressView(_ notification: Notification) {
    var userInfo: [AnyHashable: Any] = notification.userInfo!
    var NotifyId: String? = (userInfo["NotifyId"] as? String)
    var MessageType: String? = (userInfo["MessageType"] as? String)
    let Caller: String? = (userInfo["Caller"] as! String)
    let arg: NSObject? = (userInfo["ArgData"] as? NSObject)

    if ( Caller!.caseInsensitiveCompare("getNIDTextNi") == ComparisonResult.orderedSame)
    {
        let perc: Int? = (userInfo["Percent"] as? Int)
        :
    }
    if( Caller!.compare("getNIDPhotoNi") == ComparisonResult.orderedSame )
    {
        let perc: Int? = (userInfo["Percent"] as? Int)
        :
    }
    if( Caller!.compare("CardStatus") == ComparisonResult.orderedSame )
    {
        let Status: Int? = (userInfo["Status"] as? Int)
        :
    }
    if( Caller!.compare("ReaderStatus") == ComparisonResult.orderedSame )
    {
        let Status: Int? = (userInfo["Status"] as? Int)
        :
    }
    if Caller!.compare("scanReaderListBleNi") == ComparisonResult.orderedSame {
        let readerName:String! = arg as! String
        detectReaderList.add(readerName)
        self.tableView.reloadData()
    }
}

```

## 10. การจัดการแฟ้มใบอนุญาต (.dlt)

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

สำหรับเครื่องอ่านแบบ Virtual Licensing (VL) เช่นเครื่องอ่าน BLE และเครื่องอ่าน TDAi301VC จะไม่มีการใช้งานแฟ้มใบอนุญาต ดังนั้นจึงไม่ต้องสนใจและสามารถข้ามเนื้อหาในบทนี้ไปได้ทั้งหมด

แฟ้มใบอนุญาตจะเป็นที่รวบรวมรายการของเครื่องอ่านที่ได้ลงทะเบียนให้ใช้งานกับ TDi SDK ไว้ ดังนั้นแม้เครื่องอ่านที่มีรูปร่างเหมือนกัน ชื่อจากบริษัท อาร์แอนดีดี คอมพิวเตอร์ ซิสเต็ม มาเหมือนกัน แต่หากเป็นเครื่องเปล่าที่ยังไม่ได้ลงทะเบียน ก็จะใช้งานกับ TDi SDK ไม่ได้ เครื่องอ่านที่ลงทะเบียนแล้ว จะขายอยู่ในชุดสินค้าที่มีชื่อรุ่นขึ้นต้นด้วย TDAi เท่านั้น

- แฟ้มใบอนุญาตของ TDi SDK จะมีชื่อเป็น `rdnidlib.dlt` เสมอ
- ในแอปพลิเคชันตัวอย่าง NiSample สามารถดูข้อมูลของแฟ้มใบอนุญาตได้จากปุ่ม “License Info” ซึ่งจะมีชื่อโฟลเดอร์ ชื่อแฟ้ม และวันที่ของแฟ้ม
- ภายในแฟ้มใบอนุญาตจะมีรายการของหมายเลขเครื่อง (S/N) ของเครื่องอ่านที่ลงทะเบียนไว้แล้ว
- ในการทำงานของ NiLib จะมีการตรวจสอบ S/N ของเครื่องอ่านว่ามีลงทะเบียนอยู่ในแฟ้มใบอนุญาตหรือไม่ หากมีก็จะสามารถใช้งานเครื่องอ่านนั้นได้ หากไม่มีก็จะใช้ไม่ได้
- ต้นฉบับแฟ้มใบอนุญาตจะมีอยู่บนเซิร์ฟเวอร์ที่สามารถติดต่อได้ผ่านทางอินเทอร์เน็ต ตัวแฟ้มจะมีการปรับปรุงเพิ่มเติม S/N ของเครื่องอ่านโดยการสะสมหมายเลข S/N เพิ่มขึ้นเรื่อย ๆ ตามจำนวนการผลิต แฟ้มรุ่นใหม่มากกว่าจะมีจำนวนเครื่องอ่านที่มากกว่า และใช้งานแทนแฟ้มรุ่นเก่าได้เสมอ
- ในครั้งแรกที่แอปพลิเคชันทำงาน ควรมีการเตรียมแฟ้ม `rdnidlib.dlt` นี้ไว้ในโฟลเดอร์ของแอปพลิเคชันก่อน โดยอาจใช้วิธี Copy แฟ้ม `rdnidlib.dlt` นี้มาวางไว้เอง หรือให้แอปพลิเคชันส่งดาวน์โหลดจากเครื่องแม่ข่ายผ่านทางอินเทอร์เน็ตก็ได้ (ส่งด้วยฟังก์ชัน `updateLicenseFileNi`)
- เมื่อพบการเชื่อมต่อกับเครื่องอ่านที่ยังไม่ได้ลงทะเบียน อาจเป็นเพราะแฟ้มใบอนุญาตที่มีอยู่ในเครื่องเป็นแฟ้มรุ่นเก่า ให้ลองสั่งอัปเดตด้วยฟังก์ชัน `updateLicenseFileNi` หากอัปเดตแล้วยังใช้ไม่ได้ แสดงว่าเป็นเครื่องอ่านที่ยังไม่ได้ลงทะเบียนจริง ๆ ให้แจ้งผู้ใช้ถอดเปลี่ยนเครื่องอ่านเป็นเครื่องใหม่ที่ลงทะเบียนแล้วแทน
- ในการพัฒนาแอปพลิเคชัน ผู้พัฒนาควรเขียนโปรแกรมให้มีการโหลดและอัปเดตแฟ้มใบอนุญาตได้หลาย ๆ วิธี เพื่อความสะดวกในการใช้งานของผู้ใช้งาน ได้แก่
  - ให้มีการสั่งโหลดหรืออัปเดตแฟ้มใบอนุญาตทันทีหลังจากติดตั้งแอปพลิเคชันในครั้งแรก
  - ให้มีการสั่งอัปเดตแฟ้มใบอนุญาตทุกครั้งที่พบ Return Code -2 (เครื่องอ่านนี้ใช้ไม่ได้เพราะไม่มีทะเบียนในใบอนุญาต) หรือ Return Code -12 (ไม่พบแฟ้มใบอนุญาตหรือแฟ้มใบอนุญาตเสียหาย) (ดูตัวอย่างได้จาก Source Code ของ NiSample)
  - ทำปุ่ม “Update License” ไว้ที่ในหน้าจอใดหน้าจอหนึ่งของแอปพลิเคชัน เพื่อให้ผู้ใช้สามารถสั่งอัปเดตได้ด้วยตัวเอง เพื่อให้ End Users ใช้แก้ไขปัญหาแฟ้มใบอนุญาตที่ไม่ยอมอัปเดตหรือแฟ้มสูญหายได้เอง (ดูตัวอย่างได้จาก Source Code ของ NiSample)
- ในการสั่งอัปเดตแฟ้มใบอนุญาต จะต้องมีการใช้งานอินเทอร์เน็ตด้วย หากสั่งอัปเดตแล้ว พบข้อผิดพลาด Return Code -15 (ไม่สามารถติดต่ออินเทอร์เน็ตได้) หรือ Return Code -18 (อัปเดตแฟ้มใบอนุญาตไม่สำเร็จ) อาจต้องร้องขอให้ผู้ใช้เปิดการใช้งานอินเทอร์เน็ตของโทรศัพท์มือถือแล้วลองสั่งอัปเดตอีกครั้ง

## 11. ความสามารถในการเชื่อมต่อกับเครื่องอ่านของ NiLib

- เครื่องอ่านแบบ BLE จะไม่สามารถใช้งานคู่กับเครื่องอ่านแบบอื่นได้
- NiLib สามารถรองรับการสั่งอ่านบัตรประชาชนในขณะใดขณะหนึ่ง ได้เพียงเครื่องเดียวเท่านั้น โดยผู้พัฒนาจะต้องเลือกเครื่องอ่านที่จะใช้โดยให้ใช้ฟังก์ชัน `scanReaderListBleNi` ในการขอรายชื่อเครื่องอ่าน และให้เพิ่มโปรแกรม Pop-up การแสดงผลการค้นหาเพื่อให้ผู้ใช้เลือกเครื่องอ่านที่ต้องการ ดูจากตัวอย่างคลาส `scanBleViewCtrl` ในไฟล์ `scanBleViewCtrl.swift` หรือจะพัฒนาหน้าจอ Pop-up นี้ในรูปแบบอื่นก็ได้ จากนั้นให้สั่งเลือกเครื่องอ่านผ่านฟังก์ชัน `selectReaderNi`

## 12. การลงทะเบียนแอปพลิเคชันและขอรหัส MFi PPID

ในการพัฒนาแอปพลิเคชันที่เกี่ยวข้องกับฮาร์ดแวร์ ทาง Apple Inc. มีข้อกำหนดให้ผู้พัฒนาทุกรายจะต้องลงทะเบียนแอปพลิเคชันกับทาง Apple ก่อนที่จะนำแอปพลิเคชันขึ้น App Store ได้ โดยให้ผู้พัฒนาติดต่อกับผู้ผลิตฮาร์ดแวร์ที่ได้รับใบรับรองแล้วเท่านั้น

สำหรับเครื่องอ่านแบบ BLE จะได้รับการยกเว้น ไม่ต้องขอรหัส PPID ดังนั้นจึงไม่ต้องสนใจและสามารถข้ามเนื้อหาในบทนี้ไปได้ทั้งหมด

ในการลงทะเบียนนี้ ทางบริษัทอาร์แอนดีดี คอมพิวเตอร์ ซิสเต็ม จำกัด จะเป็นตัวแทนในการดำเนินการลงทะเบียนให้ เมื่อลงทะเบียนเสร็จก็จะได้รับรหัส MFi Product Plan ID (PPID) เพื่อนำไปใช้ในการนำแอปพลิเคชันขึ้น App Store ต่อไป

ในการขอรหัส PPID จะใช้เวลาประมาณ 2-3 สัปดาห์ (ขึ้นกับทางผู้ผลิตและ Apple Inc.) หากผู้พัฒนาแอปพลิเคชันที่ต้องการลงทะเบียนหลายแอปพลิเคชัน ก็ต้องแจ้งลงทะเบียนทุกโปรแกรมตามจำนวนแอปพลิเคชัน โดยให้ขอเมื่อพัฒนาแอปพลิเคชันเสร็จแล้วเท่านั้น วิธีขอ ให้ผู้พัฒนาแจ้งข้อมูลต่อไปนี้มายังบริษัทอาร์แอนดีดี คอมพิวเตอร์ ซิสเต็ม จำกัด (เป็นข้อมูลภาษาอังกฤษทั้งหมด)

1. Developer name: (ชื่อนามสกุลของผู้พัฒนา)
2. Developer organization: (ชื่อหน่วยงานหรือชื่อบริษัทของผู้พัฒนา)
3. Developer e-mail/phone: (อีเมลและหมายเลขโทรศัพท์ในการติดต่อกับผู้พัฒนา)
4. Copyright owner name: (ชื่อนามสกุลเจ้าของลิขสิทธิ์ หรือผู้ว่าจ้างในการพัฒนา)
5. Copyright owner organization: (ชื่อหน่วยงานหรือชื่อบริษัทของเจ้าของลิขสิทธิ์ หรือผู้ว่าจ้างในการพัฒนา)
6. Copyright owner email/phone: (อีเมลและหมายเลขโทรศัพท์ในการติดต่อกับเจ้าของลิขสิทธิ์)
7. Name of app as it will appear in App Store: (ชื่อแอปพลิเคชันบน App Store)
8. App version: (รุ่นของแอปพลิเคชัน)
9. Planned release date: (วันที่ต้องการปล่อยแอปพลิเคชันโดยประมาณ)
10. Apple Store category: (ประเภทของแอปพลิเคชันใน App Store)
11. Bundle identifier: (ชื่อ Bundle ID ที่ใช้)
12. Protocol names: (ชื่อโปรโตคอล ปกติให้ระบุมาเป็น com.ftsafe.bR301 สำหรับเครื่องอ่านแบบบลูทูธ (TDAi301BT) หรือให้ระบุมาเป็น com.ftsafe.iR301 สำหรับเครื่องอ่านแบบเสียบ (TDAi301U8A และ TDAi301VC) หรือระบุชื่อโปรโตคอลทั้ง 2 แบบ หากต้องการให้ใช้ได้กับเครื่องอ่านทั้งแบบบลูทูธและแบบเสียบ)
13. Functional overview: (คำอธิบายการทำงานของแอปพลิเคชัน)
14. Devices and S/N: (ระบุชื่อรุ่นเครื่องอ่านที่ใช้งานในแอปพลิเคชันนี้ และ S/N ของสินค้าที่ได้ซื้อมาเพื่อพัฒนา ทางบริษัท อาร์แอนดีดี คอมพิวเตอร์ ซิสเต็ม จำกัด จะใช้ข้อมูลนี้ตรวจสอบว่าเป็นลูกค้าของบริษัท ฯ จริงหรือไม่ หากซื้อมาหลายเครื่อง ให้ระบุ S/N หนึ่งหมายเลขต่อเครื่องอ่าน 1 รุ่น)

เมื่อได้รับรหัส MFi PPID แล้ว ให้นำข้อมูลนี้กรอกลงในช่อง Review Notes ในระหว่างการส่งแอปพลิเคชันขึ้น App Store โดยให้กรอกในรูปแบบ “MFi PPID \*\*\*\*\*” ตามรูป

<b>Review Notes (optional)</b>	MFI PPID 123456-7890
--------------------------------	----------------------

### 13. คำแนะนำการใช้งานกับเครื่องอ่านบลูทูธ (ควรแจ้งให้ผู้ใช้งาน End Users ทราบดีด้วย)

#### 13.1 การใช้งานกับเครื่องอ่านแบบบลูทูธพลังงานต่ำ (BLE) ได้แก่สินค้ารุ่น TDAi301BL และ TDAi301BLM

##### 13.1.1 การใช้งานเครื่องอ่านแบบ BLE ที่เครื่อง iPhone, iPad ให้ทำดังนี้

- เปิดการใช้งานบลูทูธของเครื่อง iPhone, iPad
- **ไม่ต้องจับคู่เครื่องอ่าน BLE กับโทรศัพท์มือถือ** แอปพลิเคชันจะแสดงชื่อเครื่องอ่านให้เลือกเองตอนใช้งาน

##### 13.1.2 ความหมายของดวงไฟของเครื่องอ่านบลูทูธ (BLE) รุ่น TDAi301BL และ TDAi301BLM

- ไฟสีน้ำเงินกระพริบ: รอการเชื่อมต่อ, ไฟสีน้ำเงินติดสว่างค้าง: เชื่อมต่อสำเร็จ สามารถส่งอ่านข้อมูลได้
- ไฟสีเหลือง ติดสว่างหรือกระพริบเมื่อมีการติดต่อกับบัตร
- ไฟสีแดง(ดวงขวา) ติดสว่างเมื่อมีการอัปเดตประจุ (ชาร์จ) ไฟให้กับแบตเตอรี่ และจะดับเมื่อแบตเตอรี่เต็ม
  - หากเสียบสายชาร์จแล้วไฟสีแดง ไม่ติดสว่าง แสดงว่าแบตเตอรี่เต็ม หรือยังเกือบเต็มอยู่ ไม่จำเป็นต้องอัปเดตประจุกระแสไฟอีก (นั่นคือ ถ้าเสียบสายแล้วไฟสีแดงไม่ติด ไม่ได้แปลว่าเครื่องอ่านเสียบหรือชาร์จไฟไม่เข้า แต่แปลว่าไม่จำเป็นต้องชาร์จ และเครื่องอ่านยังทำงานคืออยู่ตามปกติ)

##### 13.1.3 การเปิดปิดเครื่องอ่าน BLE

- การเปิดใช้งาน ให้กดสวิทช์ด้านล่าง 1 ครั้ง สังเกตจะเห็นไฟสีน้ำเงินกระพริบ
- การปิด ให้กดสวิทช์ด้านล่าง ค้างประมาณ 3 วินาที สังเกตไฟจะดับหมด
- เครื่องจะปิดเองอัตโนมัติเพื่อประหยัดพลังงาน เมื่อ
  - ถูกยกเลิกการเชื่อมต่อ หรือหลุดจากการเชื่อมต่อ (เช่น ปิดสัญญาณบลูทูธของโทรศัพท์มือถือ หรือนำเครื่องอ่านออกจากโทรศัพท์มือถือเกินระยะทำงานของสัญญาณบลูทูธ)
  - รอแล้วไม่มีการเชื่อมต่อนานกว่า 3 นาที

##### 13.1.4 ปัญหาเครื่องอ่าน BLE ปิดเครื่องเอง หรือการเชื่อมต่อบลูทูธหลุด

- อาจเกิดจากแบตเตอรี่อ่อน (สังเกตไฟจะดับทันทีที่สั่งอ่านบัตร) ทดสอบโดยการเสียบสายชาร์จกับอะแดปเตอร์สำหรับชาร์จจะใช้งานได้ทันที (อย่าชาร์จโดยการเสียบสายกับคอมพิวเตอร์หรือโทรศัพท์มือถือ ให้เสียบกับอะแดปเตอร์เท่านั้น)
- เมื่อมีการเชื่อมต่อกับเครื่องอ่านบลูทูธแล้ว แต่ไม่ได้มีการสั่งอ่านเป็นเวลานาน ๆ บางครั้งจะพบว่าการเชื่อมต่อหลุด เนื่องจากมีสัญญาณรบกวน หรือ เครื่องอ่านอยู่ใกล้กับคลื่นรบกวนจากการของสัญญาณบลูทูธ
- เมื่อการเชื่อมต่อสัญญาณบลูทูธหลุด ไม่ว่าจะจากกรณีใดก็ตาม เครื่องอ่านบลูทูธจะปิดตัวเอง ดังนั้นผู้ใช้จะต้องกดปุ่มเปิดเครื่องอ่านบลูทูธใหม่เองทุกครั้งที่จะใช้งานต่อ

#### 13.2 การใช้งานกับเครื่องอ่านแบบบลูทูธธรรมดา (Bluetooth 3.0) ได้แก่สินค้ารุ่น TDAi301BT

##### 13.2.1 การจับคู่อุปกรณ์บลูทูธ (สำหรับสินค้ารุ่น TDAi301BT)

โดยปกติ การจะนำอุปกรณ์บลูทูธทุกชนิดมาใช้งานกับโทรศัพท์มือถือ จะต้องมีการ “จับคู่” (Pair) ก่อนเสมอ การจับคู่ระหว่างเครื่องอ่านบัตรแบบบลูทูธกับโทรศัพท์มือถือ iPhone หรือ iPad ให้ทำดังนี้

- หากเครื่องอ่านบัตรสมาร์ทการ์ดยังไม่ได้อัปเดตเฟิร์มแวร์ ให้ดาวน์โหลดเฟิร์มแวร์มาใส่ให้เรียบร้อยก่อน
- กดปุ่มที่ด้านหน้าเครื่องอ่านบัตร สังเกตจะมีไฟสีน้ำเงินกระพริบเป็นจังหวะ
- ที่โทรศัพท์หรือแท็บเล็ต เข้าแอปพลิเคชัน การตั้งค่า ของระบบ iOS
- และที่ **Bluetooth (บลูทูธ)** ตั้ง “เปิด” จะพบว่ามีคำค้นหาอุปกรณ์ เมื่อพบเครื่องอ่านจะมีรายชื่อเครื่องอ่านแสดงขึ้นมาบนจอ ในรูปแบบ FT\_123456789ABC ให้ดูเฉพาะรหัส 4 ตัวท้าย เทียบกับรหัส 4 ตัวที่ด้านหลังเครื่องอ่านบัตรว่าตรงกันหรือไม่ หากตรงกัน ให้แตะชื่อเครื่องอ่านนั้น สักพักจะมีการแสดงว่าเครื่องอ่านนั้น “Connected” หรือ “เชื่อมต่ออยู่” และถ้าสังเกตที่เครื่องอ่าน จะเห็นว่าไฟสีน้ำเงินจะกระพริบเป็นจังหวะ 2 ครั้งติดกันหรือติดสว่างค้าง
- ในบางกรณี เมื่อ iPhone หรือ iPad พบเครื่องอ่าน จะมีหน้าต่างแสดงขึ้นมาว่า “คำขอจับคู่บลูทูธ” ให้แตะที่ปุ่ม “จับคู่” ได้เลย
- หากไม่พบเครื่องอ่านหรือจับคู่ไม่ได้ ให้ลองปิดเครื่องอ่าน โดยการกดปุ่มค้างจนไฟดับ แล้วกดปุ่มอีกครั้ง ส่วนที่ iPhone/iPad ให้สั่งปิด และเปิด บลูทูธใหม่อีกครั้ง

#### 13.2.2 ควรตั้ง Pair เครื่องอ่านบลูทูธก่อนเรียกใช้งานแอปพลิเคชันในครั้งแรก หากเรียก

แอปพลิเคชันไปก่อน แล้วมีการ Pair ภายหลัง อาจทำให้แอปพลิเคชันไม่เห็นเครื่องอ่าน และจะไม่ยอมเชื่อมต่อกับเครื่องอ่านนั้น แก้ไขโดยให้ออกจากแอปพลิเคชัน ปิดสัญญาณบลูทูธของ iPhone/iPad แล้วทำตามข้อ 13.1 ใหม่อีกครั้ง จากนั้นค่อยเรียกใช้งานแอปพลิเคชัน

#### 13.2.3 ความหมายของดวงไฟของเครื่องอ่านบลูทูธรุ่น TDAi301BT

- ไฟสีน้ำเงินกระพริบ 1 ครั้ง: รอการเชื่อมต่อ
- ไฟสีน้ำเงินกระพริบ 2 ครั้ง: เชื่อมต่อสำเร็จ สามารถส่งอ่านข้อมูลได้
- ไฟสีน้ำเงินกระพริบ 3 ครั้ง: กำลังตรวจสอบตัวเองหลังจากเปิดเครื่อง และรอการเชื่อมต่อ
- ไฟสีน้ำเงินติดสว่างค้าง: เชื่อมต่อสำเร็จ สามารถส่งอ่านข้อมูลได้
- ไฟสีขาว ติดสว่างหรือกระพริบเมื่อมีการติดต่อกับบัตร
- ไฟสีเหลือง ติดสว่างเมื่อแบตเตอรี่อ่อน ควรนำเครื่องอ่านไปประจุไฟฟ้าให้เต็ม
- ไฟสีแดง ติดสว่างเมื่อมีการประจุ (ชาร์จ) ไฟให้กับแบตเตอรี่ และจะดับเมื่อแบตเตอรี่เต็ม
  - หากเสียบสายชาร์จแล้วไฟสีแดงไม่ติดสว่าง แสดงว่าแบตเตอรี่เต็ม หรือยังเกือบเต็มอยู่ ไม่จำเป็นต้องประจุกระแสไฟอีก (นั่นคือ ถ้าเสียบสายแล้วไฟสีแดงไม่ติด ไม่ได้แปลว่าเครื่องอ่านเสียบหรือชาร์จไฟไม่เข้า แต่แปลว่าไม่จำเป็นต้องชาร์จ และเครื่องอ่านยังทำงานอยู่ตามปกติ)

#### 13.2.4 การเปิดปิดเครื่องอ่านบลูทูธ

- การเปิดใช้งาน ให้กดสวิทช์ด้านหน้าเครื่อง 1 ครั้ง สังเกตจะเห็นไฟสีน้ำเงินกระพริบ
- การปิด ให้กดสวิทช์ด้านหน้าค้างประมาณ 3 วินาที สังเกต ไฟจะดับหมด
- เครื่องจะปิดเองอัตโนมัติเพื่อประหยัดพลังงาน เมื่อ
  - ถูกยกเลิกการเชื่อมต่อ หรือหลุดจากการเชื่อมต่อ (เช่น ปิดสัญญาณบลูทูธของโทรศัพท์มือถือ หรือนำเครื่องอ่านออกจากโทรศัพท์มือถือเกินระยะเวลาทำงานของสัญญาณบลูทูธ)



- รอแล้วไม่มีการเชื่อมต่อนานกว่า 3 นาที

13.2.5 ไม่ควรจับคู่ (Pair) เครื่องอ่านบลูทูธเครื่องเดียวกับอุปกรณ์โทรศัพท์มือถือหลาย ๆ เครื่อง เพราะจะทำให้สับสนได้ง่าย และจะมีการแย่งจับคู่จนไม่รู้วากำลังจับคู่กับเครื่องใด

13.2.6 อุปกรณ์โทรศัพท์มือถือควรจับคู่กับเครื่องอ่านแบบบลูทูธเพียงเครื่องเดียว เพื่อลดปัญหาการสับสน หากเป็นไปได้ ควรสั่งให้อุปกรณ์ iPhone หรือ iPad ขกเลิกการจับคู่ (**ลบอุปกรณ์นี้** หรือ **Forget This Device**) กับเครื่องอ่านบลูทูธที่จะไม่ได้ใช้งานกับ iPhone หรือ iPad เครื่องนั้นแล้ว

13.2.7 การตรวจสอบว่าขณะนี้อุปกรณ์ iPhone หรือ iPad กำลังเชื่อมต่อกับเครื่องอ่านบลูทูธเครื่องไหนอยู่บ้าง ให้เข้าที่ **Settings (การตั้งค่า)**, และที่ **Bluetooth (บลูทูธ)** ดูที่ **MY DEVICES (อุปกรณ์ของฉัน)** จะมีรายการอุปกรณ์บลูทูธที่เกี่ยวข้องแสดงอยู่ ให้ดูรหัส 4 ตัวท้ายของชื่อเครื่องที่ขึ้นต้นด้วย FT เพื่อนำไปเปรียบเทียบกับรหัส 4 ตัวที่อยู่ด้านหลังเครื่องอ่าน สถานะจะมี 2 แบบคือ

- **Connected (เชื่อมต่ออยู่)** คือเครื่องอ่านนั้นกำลังเชื่อมต่อและใช้งานอยู่
- **Not Connected (ไม่ได้เชื่อมต่อ)** คือเครื่องอ่านนั้นไม่ได้เชื่อมต่อกับอุปกรณ์ iPhone หรือ iPad แล้ว แต่เมื่อใดที่มีการเปิดไฟของเครื่องอ่าน ก็จะมีการเชื่อมต่อกันใหม่แบบอัตโนมัติทันที

13.2.8 ปัญหาเครื่องอ่านบลูทูธปิดเครื่องเอง หรือการเชื่อมต่อบลูทูธหลุด

- เมื่อมีการเชื่อมต่อกับเครื่องอ่านบลูทูธแล้ว แต่ไม่ได้มีการสั่งอ่านเป็นเวลานาน ๆ บางครั้งจะพบว่าการเชื่อมต่อหลุด เนื่องจากมีสัญญาณรบกวน หรือ เครื่องอ่านอยู่ไกลเกินรัศมีทำการของสัญญาณบลูทูธ
- เมื่อการเชื่อมต่อสัญญาณบลูทูธหลุด ไม่ว่าจะจากกรณีใดก็ตาม เครื่องอ่านบลูทูธจะปิดตัวเอง ดังนั้นผู้ใช้จะต้องกดปุ่มเปิดเครื่องอ่านบลูทูธใหม่เองทุกครั้งที่จะใช้งานต่อ

## 14. รายการปรับปรุงของ TDi NiBLE SDK

### R2400502 (NiLib 0.43, NiBLE Sample 4.3)

1. เพิ่มการรองรับกับ Xcode รุ่น 14 และ 15 ได้
2. เพิ่มการรองรับการทดสอบแอปพลิเคชันด้วย Firebase App Distribution
3. เพิ่ม UL Configuration สำหรับ Build ให้ใช้กับเครื่องอ่านแบบเสียบ USB-C และ Lightning เท่านั้น
4. แก้ไขจุดบกพร่องต่าง ๆ

### R221129 (NiLib 0.42, NiBLE Sample 4.20)

5. เพิ่มฟังก์ชันใหม่ **getATextNi** อ่านข้อมูลบัตรประชาชนได้เพิ่มอีก 5 ข้อมูล
6. รองรับการดาวน์โหลดไฟล์ไลเซนส์ได้จากหลายแหล่งแบบอัตโนมัติ (License File Auto Switch Downloading)
7. แก้ไขชื่อฟังก์ชัน (**setReaderTypeNi**) และ Notification Caller ID (ReaderStatus และ CardStatus) ให้ถูกต้อง
8. โปรแกรมตัวอย่างรองรับการ Build แยกเป็น 2 แอปพลิเคชัน (NiBLE Sample และ NiBUL Sample) จาก Source Code เดียวกัน

### R221004 (NiLib 0.40)

1. รองรับการใช้งานกับเครื่องอ่านแบบ BLE รุ่น TDAi301BL และ TDAi301BLM

## 15. คำถามที่ถามบ่อยของชุดพัฒนาซอฟต์แวร์อ่านบัตรประชาชน HawkEye TDAi

### คำถาม

อะไรคือชุดพัฒนาซอฟต์แวร์ HawkEye TDAi NiBLE SDK

### คำตอบ

ชุดพัฒนาซอฟต์แวร์อ่านบัตรประชาชน HawkEye TDAi NiBLE SDK คือผลิตภัณฑ์ที่จัดชุดสำหรับการพัฒนาแอปพลิเคชัน โปรแกรมบนระบบ Android และ iOS ให้สามารถอ่านข้อมูลบัตรประจำตัวประชาชนได้ โดยภายในชุดจะประกอบด้วยเครื่องอ่านบัตรสมาร์ทการ์ด Bluetooth แบบ BLE และชุดซอฟต์แวร์สำหรับการพัฒนา (SDK) จำนวน 2 ชุด ชุดหนึ่งสำหรับการพัฒนาแอปพลิเคชันให้กับระบบแอนดรอยด์ (ซึ่งเป็นชุดเดียวกับที่อยู่ในสินค้า HawkEye TDA) และอีกชุดเป็นซอฟต์แวร์สำหรับการพัฒนาสำหรับระบบ iOS และ iPadOS

ชุดพัฒนา NiBLE SDK นี้ใช้งานกับเครื่องอ่านแบบ BLE รุ่น TDAi301BL และ TDAi301BLM เท่านั้น

### คำถาม

ชุดเครื่องอ่านบัตรประจำตัวประชาชน TDAi ต่างจากสินค้าชุดเครื่องอ่านบัตรประจำตัวประชาชน TDA, TRA และสินค้าเครื่องอ่านบัตรประชาชนอื่น ๆ อย่างไร

### คำตอบ

TDAi คือชุดสินค้าสำหรับการพัฒนาแอปพลิเคชัน ที่รองรับระบบปฏิบัติการ 2 ระบบ คือ Android และ iOS เหมาะสำหรับโปรแกรมเมอร์เพื่อนำไปใช้พัฒนาแอปพลิเคชันบน Android หรือ iOS หรือทั้ง 2 ระบบก็ได้

TDA คือชุดสินค้าสำหรับการพัฒนาแอปพลิเคชัน ที่รองรับเฉพาะระบบปฏิบัติการ Android เพียงระบบเดียว เหมาะสำหรับโปรแกรมเมอร์เพื่อนำไปใช้พัฒนาแอปพลิเคชันบน Android เท่านั้น

TRA คือชุดเครื่องอ่านบัตรประจำตัวประชาชนสำเร็จรูปสำหรับระบบ Android ภายในชุดจะมีทั้งตัวเครื่องอ่านและซอฟต์แวร์ (แอปพลิเคชัน) สำหรับอ่านบัตรประจำตัวประชาชน ชื้อไปแล้วใช้งานได้ทันที (เหมาะสำหรับผู้ซื้อทั่วไป)

ส่วนผลิตภัณฑ์เครื่องอ่านบัตรเปล่าเช่น bR301, iR301-U-8A เป็นเพียงเครื่องอ่านบัตรสมาร์ทการ์ด (เครื่องเปล่า) ไม่มีซอฟต์แวร์ให้ใช้งาน ผู้ใช้ต้องหาซอฟต์แวร์มาเอง (เหมาะสำหรับหน่วยงานที่มีแอปพลิเคชันของตัวเองแล้วเท่านั้น)

### คำถาม

สามารถซื้อเฉพาะเครื่องอ่านบัตรสมาร์ทการ์ด **bR301BLE** หรือ iR301-U-8A (เครื่องเปล่า) มาใช้งานกับ TDAi ได้หรือไม่ และสามารถใช้งานร่วมกับเครื่องอ่านของ TDA และ TRA ได้หรือไม่

### คำตอบ

ชุดพัฒนาซอฟต์แวร์ HawkEye TDAi จะใช้งานได้กับเครื่องอ่านที่อยู่ในชุดสินค้าตระกูล TDAi เท่านั้น เพราะเครื่องอ่านทุกเครื่องในชุดสินค้า TDAi จะผ่านการลงทะเบียนให้ใช้งานกับ TDAi มาก่อนแล้ว ส่วนเครื่องอ่านอื่น ที่ยังไม่ได้ลงทะเบียน หรือเป็นเครื่องเปล่า หรือเป็นเครื่องอ่านในกลุ่มสินค้า TRA จะใช้งานกับ TDAi และแอปพลิเคชันที่พัฒนามาจาก TDAi ไม่ได้

เฉพาะเครื่องอ่านจากชุด TDA สามารถนำมาใช้งานกับแอปพลิเคชันระบบ Android ของ TDAi ได้ (เพราะเป็นชุดพัฒนา Android ชุดเดียวกัน) แต่จะใช้กับระบบ iOS ไม่ได้

### คำถาม

ขั้นตอนการพัฒนาแอปพลิเคชันสำหรับ iOS ให้สามารถอ่านบัตรประชาชน จนถึงการนำไปใช้งาน ต้องทำอย่างไร

### คำตอบ

ทำตามขั้นตอนดังนี้

- ซื้อชุดพัฒนาซอฟต์แวร์ HawkEye TDAi มาอย่างน้อย 1 ชุด
- ศึกษาการใช้งาน TDAi จากคู่มือการใช้งาน ซึ่งจะอธิบายถึงวิธีการเรียกใช้ (API) และศึกษาตัวอย่างโปรแกรมจากในชุด SDK
- พัฒนาแอปพลิเคชัน และทดสอบการใช้งานกับเครื่องอ่านบัตรประชาชนที่ได้มาพร้อมกับชุดสินค้า
- เมื่อพัฒนาแอปพลิเคชันจนเสร็จ ให้ติดต่อบริษัท R&D เพื่อขอรหัส PPID และนำแอปพลิเคชันขึ้นเผยแพร่ผ่าน Apple App Store ด้วยรหัส PPID ที่ได้รับ (สำหรับเครื่องอ่านแบบ BLE ไม่ต้องขอ PPID)
- สั่งซื้อสินค้า TDAi เพิ่ม ตามจำนวนและชนิดของเครื่องอ่านที่จะใช้งาน
- ติดตั้งแอปพลิเคชัน ลงในอุปกรณ์ iPhone หรือ iPad ที่จะใช้งาน และมอบเครื่องอ่านที่ได้มากับชุดสินค้า TDAi ให้กับผู้ใช้งาน

### คำถาม

ซื้อชุดพัฒนาซอฟต์แวร์ HawkEye TDAi มา 1 ชุด แล้วนำแอปพลิเคชันที่พัฒนาได้ไปใช้งานกับเครื่องอ่านรุ่นอื่นได้หรือไม่

### คำตอบ

ไม่ได้ เพราะไลบรารีของชุดพัฒนา TDAi นี้ใช้งานได้กับเครื่องอ่านบัตรที่มากับชุดสินค้า TDAi เท่านั้น เครื่องอ่านบัตรรุ่นอื่นใช้ไม่ได้ และบริษัท ฯ ไม่อนุญาตให้นำสิทธิ์นี้ไปใช้งานกับเครื่องอ่านบัตรรุ่นอื่นด้วย

เฉพาะเครื่องอ่านจากชุด TDA สามารถนำมาใช้งานกับแอปพลิเคชันระบบ Android ของ TDAi ได้ (เพราะเป็นชุดพัฒนา Android ชุดเดียวกัน) แต่จะใช้กับระบบ iOS ไม่ได้

### คำถาม

หากเครื่องอ่านบัตรประชาชนในชุดเสีย สามารถนำสิทธิ์ (License) ไปใช้งานต่อได้หรือไม่

### คำตอบ

หากเครื่องอ่านบัตรเสียหายจนไม่สามารถใช้งานได้ จะถือว่าสิทธิ์ในการใช้งาน (License) นั้นหมดไปด้วย อย่างไรก็ตามเครื่องอ่านบัตรทุกเครื่องจะมีการรับประกันการใช้งาน หากเครื่องเสียในระยะเวลาประกันทางบริษัท ฯ มีบริการซ่อมหรือแก้ไขให้ใช้งานเครื่องอ่านได้จนเป็นปกติ ดังนั้นจะยังคงมีสิทธิ์ในการใช้งานได้อยู่เหมือนเดิม

### คำถาม

ชุด TDAi นี้สามารถขอข้อมูลแบบ Online จากเซิร์ฟเวอร์ของกรมการปกครองได้หรือไม่

### คำตอบ

TDAi จะอ่านข้อมูลจากชิปภายในบัตรประชาชนเท่านั้น และทำงานแบบ Offline ไม่มีการขอข้อมูลจากเซิร์ฟเวอร์ของกรมการปกครอง

### คำถาม

มีข้อกำหนดในการใช้งานอย่างไรบ้าง

### คำตอบ

ชุดพัฒนานี้สามารถใช้งานได้ทั่วไป แต่ทั้งนี้ก็มีข้อกำหนดในการใช้งานบางประการดังนี้

- ห้ามนำไปใช้ในทางที่ผิดกฎหมาย
- ใช้เพื่อการทำงาน, ให้บริการหรือทำธุรกรรมเพื่อเจ้าของบัตรหรือผู้ถือบัตรเท่านั้น
- ในการนำไปอ่านข้อมูลจากบัตรประจำตัวประชาชน จะต้องได้รับความยินยอมจากเจ้าของบัตรหรือผู้ถือบัตรเท่านั้น
- ห้ามนำข้อมูลภาพและข้อความที่อ่านได้จากบัตรประจำตัวประชาชนไปเปิดเผยหรือเผยแพร่ให้ผู้อื่นโดยไม่ได้รับอนุญาตจากเจ้าของบัตรหรือผู้ถือบัตร
- กรณีมีความเสียหายใด ๆ เกิดขึ้น อันเนื่องมาจากการนำซอฟต์แวร์นี้ หรือข้อมูลที่อ่านได้จากบัตรประจำตัวประชาชนไปใช้งาน ผู้ใช้ซอฟต์แวร์นี้ต้องรับผิดชอบในผลของความเสี่ยงที่เกิดขึ้นทั้งทางแพ่งและอาญาและอื่น ๆ โดยทางบริษัทอาร์แอลดีดี คอมพิวเตอร์ ซิสเต็ม จำกัด ไม่ต้องร่วมรับผิดชอบด้วย

### คำถาม

สามารถดาวน์โหลดซอฟต์แวร์ชุดพัฒนาสำหรับระบบแอนดรอยด์ (TDA NALib SDK) และชุดพัฒนาสำหรับระบบ iOS (TDi SDK) ได้จากที่ใด

### คำตอบ

ผู้พัฒนาสามารถดาวน์โหลดซอฟต์แวร์ชุด SDK ได้จากเว็บไซต์ของบริษัท (www.rd-comp.com) โดยการเข้าไปที่หน้าสินค้า HawkEye TDAi รุ่น TDAi301BL แล้วคลิกดาวน์โหลดจาก Link ในหน้านั้น

### คำถาม

สามารถดาวน์โหลดแอปพลิเคชันตัวอย่าง NiSample สำหรับภาษา Objective-C และ Swift ได้จากที่ใด

### คำตอบ

สามารถดาวน์โหลดแอปพลิเคชันตัวอย่าง NiSample ทั้งสำหรับภาษา Objective-C และ Swift เพื่อลองทดสอบได้จาก App Store โดยค้นหาคำว่า NiOS หรือสแกน QR Code นี้

**เครื่องอ่านแบบ BLE ไม่สามารถใช้กับแอปพลิเคชันตัวอย่างนี้ได้**



NiSample (Obj-C)



NiSample (Swift)



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

02-694-3133

[sales@rd-comp.com](mailto:sales@rd-comp.com)

<https://www.rd-comp.com>

*TDAi\_NiBLE\_Manual\_R240502.odt*