เครื่องยืนยันตัวตนด้วย NFC

ประภากร ศรีวรสาร
พีรดา แสงแป้
ศตคุณ อุตมะ

โครงงานนี้เป็นส่วนหนึ่งของ วิชาโครงงานรหัสวิชา 21909-2023
ตามหลักสูตรประกาศนียบัตรวิชาชีพ พุทธศักราช 2567
ประเภทวิชาอุตสาหกรรมดิจิทัลและเทคโนโลยีสารสนเทศ
สาขาวิชาช่างเทคนิคคอมพิวเตอร์ วิทยาลัยเทคนิคหนองคาย
สำนักงานคณะกรรมการการอาชีวศึกษา
ปีการศึกษา 2568

หัวข้อโครงงาน	เครื่องยืนยันตัวตนด้วย NFC
นักศึกษา	นางสาว ประภาภร ศรีวรสาร รหัสประจำตัว 67219090006
	นางสาว พีรดา แสงแป้น รหัสประจำตัว 67219090007
	นาย ศตคุณ อุตมะ รหัสประจำตัว 67219090014
หลักสูตร	ประกาศนียบัตรวิชาชีพ
สาขาวิชา	ช่างเทคนิคคอมพิวเตอร์
สาขางาน	ช่างเทคนิคคอมพิวเตอร์
พ.ศ.	2568
ครูที่ปรึกษา	นายภาคิน เหรียญทอง
ครูที่ปรึกษาร่วม	นายวสันต์ สารคำ

บทคัดย่อ

โครงงานนี้มีวัตถุประสงค์เพื่อศึกษาและประยุกต์ใช้เทคโนโลยี NFC ในการยืนยันตัวตน ออกแบบและสร้างเครื่องมือที่สามารถตรวจสอบสิทธิ์การเข้าใช้งานได้อย่างมีประสิทธิภาพ รวมถึงป้องกันการบุกรุกจากบุคคลที่ไม่ได้รับอนุญาต และเพิ่มความสะดวกในการยืนยันตัวตนผ่านโทรศัพท์มือถือ โดยมุ่งเน้นให้ระบบมีการทำงานที่ง่าย รวดเร็ว และมีความน่าเชื่อถือ เหมาะสมต่อการนำไปใช้งานจริง

การดำเนินงานเริ่มจากการศึกษาหลักการทำงานของเซนเซอร์ NFC และอุปกรณ์ไมโครคอนโทรลเลอร์ที่เกี่ยวข้อง จากนั้นทำการออกแบบวงจรและโครงสร้างของอุปกรณ์ให้เหมาะสม ผู้จัดทำได้พัฒนาระบบให้สามารถอ่านข้อมูลจากโทรศัพท์มือถือหรือการ์ด NFC ที่ได้ลงทะเบียนไว้ เมื่อมีการนำอุปกรณ์มาสแกนระบบจะทำการตรวจสอบข้อมูลกับฐานข้อมูล หากข้อมูลถูกต้องจะอนุญาตให้เข้าถึงได้ทันที แต่หากไม่ถูกต้องระบบจะปฏิเสธการเข้าถึง เพื่อป้องกันการใช้งานจากบุคคลที่ไม่ได้รับสิทธิ์

นอกจากนี้ยังได้มีการทดสอบประสิทธิภาพของอุปกรณ์ในหลายด้าน เช่น ความสามารถในการอ่านค่า NFC จากอุปกรณ์หลายรูปแบบ ระยะการทำงานของเซนเซอร์ ความถูกต้องในการตรวจสอบข้อมูลและความเสถียรของระบบ ผลการทดสอบแสดงให้เห็นว่าเครื่องมือที่พัฒนาขึ้นสามารถทำงานได้ตามวัตถุประสงค์ สามารถยืนยันตัวตนได้อย่างถูกต้อง มีความรวดเร็วในการประมวลผล และช่วยลดโอกาสการบุกรุกจากบุคคลภายนอกได้จริง

จากผลการดำเนินงานสรุปได้ว่า เครื่องมือยืนยันตัวตนด้วย NFC ที่พัฒนาขึ้นสามารถนำไปประยุกต์ใช้ในด้านการรักษาความปลอดภัยได้หลากหลาย เช่น การควบคุมการเข้า–ออกห้องเรียน ห้องพัก ที่อยู่อาศัย หรือสถานที่ที่ต้องการจำกัดสิทธิ์การเข้าถึง โครงงานนี้จึงเป็นแนวทางหนึ่งในการพัฒนาระบบรักษาความปลอดภัยที่มีต้นทุนไม่สูง ใช้งานง่าย และสามารถต่อยอดพัฒนาเพิ่มเติมในอนาคต เช่น การเชื่อมต่อกับระบบแจ้งเตือนผ่านแอปพลิเคชัน การบันทึกประวัติผู้เข้าใช้งาน หรือการเพิ่มรูปแบบการยืนยันตัวตนร่วมกับเทคโนโลยีอื่น เพื่อเพิ่มประสิทธิภาพและความปลอดภัยให้ดียิ่งขึ้น

Thesis Title	NFC Authentication Machine
Student	Prapakorn Sriworasan Student ID 67219090006
	Peerada Sangpae Student ID 67219090007
	Satakun Utama Student ID 67219090014
Degree	Vocational Certificate Program
Programme	Computer Technology
Year	2568
Thesis Advisor	Mister Pakin Reanthong
Thesis Co-advisor	Mister Wasan Sarakum

Abstract

This project aims to study and apply NFC technology for identity verification, design and create tools for performing authorization, prevent invasion from an unauthorized personnel, and enhance the convenience of mobile phone authentication. The focus is on creating a system that is simple, fast, reliable, and suitable for practical application.

The project began by studying the operating principles of NFC sensors and related microcontrollers. Then, the circuit and structure of the device were designed accordingly. The developers developed a system capable of reading data from registered mobile devices or NFC cards. When a device is scanned, the system verifies the data against a database. If the data is correct, access is granted immediately. If incorrect, the system denies access to prevent unauthorized entry.

Furthermore, the device's performance was tested in several aspects, such as its ability to read NFC data from various devices, the sensor's operating range, verification accuracy, and system stability. The test results showed that the developed tool can function as intended, accurately verifying identities, processing data quickly and effectively reducing the potential for intrusion by unauthorized individuals.

From the project's results, it can be concluded that the developed NFC authentication tool can be applied in various security applications, such as controlling access to classrooms, dorm rooms, residences, or places requiring restricted access. This project therefore represents a pathway for developing a low-cost, easy-to-use security system that can be further improved upon in the future, such as connecting to an application-based notification system, recording user history, or adding authentication methods with other technologies to enhance efficiency and security.

กิตติกรรมประกาศ

ผู้จัดทำขอขอบคุณครูที่ปรึกษาและครูที่ปรึกษาร่วม นายภาคิน เหรียญทอง และนายวสันต์ สารคำ ที่ได้ให้คำปรึกษาและคำแนะนำวิธีการในการจัดทำโครงงานให้แก่กลุ่มของข้าพเจ้า ช่วยให้ชิ้นงานสำเร็จลุล่วงตามจุดประสงค์ที่ได้ตั้งไว้อย่างราบรื่น และขอขอบคุณวิทยาลัยเทคนิคหนองคายที่ได้ให้โอกาศแก่กลุ่มของข้าพเจ้าในการประดิษฐ์เครื่องยืนยันตัวตนด้วยระบบ NFC นี้ขึ้นมา

นางสาวประภากร ศรีวรสาร
นางสาวพีรดา แสงแป้
นายศตคุณ อุตมะ

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญรูป

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ความปลอดภัยนั้นเป็นเรื่องสำคัญสำหรับทุกคนแต่องค์กรแต่ละองค์กรและคนแต่ละคนมักมีความต้องการด้านความปลอดภัยไม่เหมือนกัน แต่ในบางครั้ง เมื่อมีบุคคลหรือองค์กรที่ต้องการเทคโนโลยีด้านความปลอดภัยเหล่านี้ เทคโนโลยีความปลอดภัยนั้นอาจมีราคาสูงเกินกว่าจะเอื้อมถึงได้ ส่งผลให้อาจมีการลดระดับความปลอดภัยลงมา เพิ่มความเสี่ยงของชีวิต ทรัพย์สิน เอกสาร และข้อมูลต่าง ๆ ขององค์กรหรือบุคคลนั้น ๆ

ในโลกปัจจุบัน อินเทอร์เน็ตนั้นก็เป็นสิ่งที่สำคัญมากเช่นกัน และสถานที่ส่วนใหญ่มักจะมีอินเทอร์เน็ต จึงก่อให้เกิดการที่มีอุปกรณ์อินเทอร์เน็ตรอบตัวเพิ่มขึ้นทุกวัน และได้มีสิ่งที่เรียกว่า Internet of Things (IoT) เกิดขึ้น ซึ่งคืออุปกรณ์ที่ถูกปรับปรุงให้ใช้งานได้ดีขึ้นด้วยเทคโนโลยีไร้สายต่าง ๆ เช่น Wi-Fi, Bluetooth, Zigbee, และ Thread

โครงงานนี้จึงมีเป้าหมายที่จะแก้ไขปัญหาที่กล่าวไปข้างต้น พร้อมศึกษาและเรียนรู้เกี่ยวกับเทคโนโลยีไร้สาย Wi-Fi และ NFC เพื่อสร้างอุปกรณ์ยืนยันตัวตนที่ต้นทุนไม่สูงมากและให้ราคาเข้าถึงได้ง่ายขึ้น

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อเพิ่มความปลอดภัยของพื้นที่
1.2.2 เพื่อเพิ่มความไว้วางใจของบุคลากรในองค์กร
1.2.3 เพื่อป้องกันข้อมูลขององค์กรที่อาจรั่วไหล
1.2.4 เพื่อรับมือเหตุการณ์ผู้บุกรุกได้อย่างทันท่วงที

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1.3.1 สามารถประยุกต์ความรู้ด้านอิเล็กทรอนิกส์และเทคโนโลยีมาใช้ในชีวิตประจำวันได้จริง
1.3.2 สามารถตรวจจับผู้บุกรุกได้ ช่วยเพื่อเพิ่มความไว้วางใจของบุคลากรในองค์กร

1.4 ขอบเขตของโครงงาน

1.4.1 สามารถแจ้งเตือนสัญญาณเสียงได้
1.4.2 สามารถแจ้งเตือนผ่านโทรศัพท์มือถือได้
1.4.3 สามารถตรวจจับบุคคลที่ไม่ได้รับอนุญาตได้

1.5 นิยามศัพท์เฉพาะ

เครื่องยืนยันตัวตนด้วย NFC คืออุปกรณ์ความปลอดภัยที่มีหน้าที่ในการยืนยันตัวตนบุคคลที่เข้าออกพื้นที่ โดยใช้เทคโนโลยี NFC เป็นระบบยืนยันตัวตนบุคคลและใช้เซนเซอร์ตรวจจับความเคลื่อนไหวในการตรวจสอบหากมีบุคคลเข้าไปโดยไม่ได้รับอนุญาต

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 ได้รับความรู้ในด้านการรักษาความปลอดภัย

1.6.2 ได้รับประสบการณ์ในการทำงานกับเทคโนโลยีไร้สาย

1.6.3 ได้รับประสบการณ์ในการทำชิ้นงานด้วย ESP32

บทที่ 2
ทฤษฎีและเอกสารที่เกี่ยวข้อง

ผู้จัดทำโครงงาน เครื่องยืนยันตัวตนด้วย NFC ได้ศึกษาทฤษฎีที่เกี่ยวข้องต่าง ๆ และรวบรวมแนวทางและหลักการต่าง ๆ จากเอกสารงานวิจัยที่เกี่ยวข้องดังต่อไปนี้

2.1 ไมโครคอนโทรเลอร์ (Microcontroller)

2.2 เซ็นเซอร์ (Sensors)

2.3 ลำโพงสัญญาณ (Buzzer)

2.4 เกณฑ์วิธีขนส่งข้อความหลายมิติ (HyperText Transfer Protocol; HTTP)

2.5 เกณฑ์วิธีขนส่งข้อความหลายมิติแบบมั่นคง (Hypertext Transfer Protocol Secure; HTTPS)

2.6 เกณฑ์วิธีความมั่นคงของชั้นขนส่ง (Transport Layer Security; TLS)

2.7 การสื่อสารสนามใกล้ (Near-field communication; NFC)

2.8 Flutter

2.9 Git

2.10 ภาษาซี (C Programming Language)

2.1 ไมโครคอนโทรเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ (Microcontroller, MC, uC, หรือ μC) หรือหน่วยไมโครคอนโทรลเลอร์ (Microcontroller Unit; MCU) เป็นคอมพิวเตอร์ขนาดเล็กบนวงจรรวมเดียว (Integrated Circuit; IC) โดยไมโครคอนโทรลเลอร์ประกอบด้วยแกนประมวลผลหนึ่งแกนหรือมากกว่า พร้อมด้วยหน่วยความจำและอุปกรณ์ต่อพ่วงอินพุต/เอาต์พุตที่ตั้งโปรแกรมได้ หน่วยความจำโปรแกรมในรูปแบบของ NOR flash, OTP ROM, หรือ ferroelectric RAM มักจะถูกรวมไว้ในชิปด้วยเช่นกัน รวมถึง RAM จำนวนเล็กน้อย ไมโครคอนโทรลเลอร์ได้รับการออกแบบมาสำหรับการใช้งานแบบฝังตัว ซึ่งแตกต่างจากไมโครโปรเซสเซอร์ที่ใช้ในคอมพิวเตอร์ส่วนบุคคลหรือแอปพลิเคชันทั่วไปอื่น ๆ ที่ประกอบด้วยชิปแยกชิ้นต่าง ๆ

ในศัพท์สมัยใหม่ ไมโครคอนโทรลเลอร์นั้นคล้ายคลึงกับระบบบนชิป (System on a chip; SoC) แต่มีความซับซ้อนน้อยกว่า SoC อาจมีไมโครคอนโทรลเลอร์เป็นส่วนประกอบหนึ่ง แต่โดยทั่วไปแล้วจะรวมเข้ากับอุปกรณ์ต่อพ่วงขั้นสูง เช่น หน่วยประมวลผลกราฟิก (GPU) โมดูล Wi-Fi หรือตัวประมวลผลร่วม (coprocessor) อย่างน้อยหนึ่งตัว

ไมโครคอนโทรลเลอร์ถูกนำไปใช้ในผลิตภัณฑ์และอุปกรณ์ควบคุมอัตโนมัติ เช่น ระบบควบคุมเครื่องยนต์รถยนต์ อุปกรณ์ทางการแพทย์ที่ฝังในร่างกาย รีโมทคอนโทรล เครื่องใช้สำนักงาน เครื่องใช้ไฟฟ้า เครื่องมือไฟฟ้า ของเล่น และระบบฝังตัวอื่น ๆ การลดขนาดและต้นทุนเมื่อเทียบกับการออกแบบที่ใช้ไมโครโปรเซสเซอร์ หน่วยความจำ และอุปกรณ์อินพุต/เอาต์พุตแยกต่างหาก ทำให้การควบคุมแบบดิจิทัลสำหรับอุปกรณ์และกระบวนการต่าง ๆ เป็นไปได้มากขึ้น ไมโครคอนโทรลเลอร์แบบผสมสัญญาณเป็นที่นิยม โดยจะรวมส่วนประกอบอนาล็อกที่จำเป็นในการควบคุมระบบอิเล็กทรอนิกส์ที่ไม่ใช่ดิจิทัล

ในบริบทของ Internet of Things (IoT) ไมโครคอนโทรลเลอร์เป็นวิธีการรวบรวมข้อมูล การตรวจจับและการกระตุ้นโลกทางกายภาพในฐานะอุปกรณ์ปลายทางที่มีราคาประหยัดและเป็นที่นิยม

ไมโครคอนโทรลเลอร์บางตัวอาจใช้คำแบบสี่บิตและทำงานที่ความถี่ต่ำถึง 4 kHz เพื่อการใช้พลังงานต่ำ (มิลลิวัตต์หรือไมโครวัตต์หลักเดียว) โดยทั่วไปแล้ว ไมโครคอนโทรลเลอร์เหล่านี้สามารถคงการทำงานไว้ได้ในขณะที่รอเหตุการณ์ เช่น การกดปุ่มหรือการขัดจังหวะอื่นๆ การใช้พลังงานขณะอยู่ในโหมดสลีป (โดยที่นาฬิกา CPU และอุปกรณ์ต่อพ่วงส่วนใหญ่ปิดอยู่) อาจอยู่ที่ระดับนาโนวัตต์เท่านั้น ทำให้หลายตัวเหมาะสำหรับแอปพลิเคชันที่ใช้แบตเตอรี่ได้นาน ส่วนไมโครคอนโทรลเลอร์อื่นๆ อาจทำหน้าที่ในบทบาทที่สำคัญต่อประสิทธิภาพ ซึ่งอาจต้องทำงานคล้ายกับตัวประมวลผลสัญญาณดิจิทัล (Digital Signal Processor; DSP) โดยมีความเร็วสัญญาณนาฬิกาและการใช้พลังงานที่สูงกว่า

2.1.1 ประวัติ

ไมโครโปรเซสเซอร์แบบหลายชิปตัวแรก ได้แก่ Four-Phase Systems AL1 ในปี 1969 และ Garrett AiResearch MP944 ในปี 1970 ซึ่งถูกพัฒนาขึ้นโดยใช้ชิป MOS LSI หลายตัว ส่วนไมโครโปรเซสเซอร์แบบชิปเดี่ยวตัวแรกคือ Intel 4004 ซึ่งวางจำหน่ายในปี 1971 โดยใช้ชิป MOS LSI เพียงตัวเดียว พัฒนาโดย Federico Faggin โดยใช้เทคโนโลยี MOS แบบซิลิคอนเกต ร่วมกับวิศวกรของ Intel คือ Marcian Hoff และ Stan Mazor และวิศวกรของ Busicom คือ Masatoshi Shima ต่อมาก็มี Intel 4040 แบบ 4 บิต, Intel 8008 แบบ 8 บิต, และ Intel 8080 แบบ 8 บิต โปรเซสเซอร์เหล่านี้ทั้งหมดต้องการชิปภายนอกหลายตัวเพื่อสร้างระบบที่ใช้งานได้ รวมถึงชิปหน่วยความจำและชิปอินเทอร์เฟซอุปกรณ์ต่อพ่วง ส่งผลให้ต้นทุนของระบบโดยรวมสูงถึงหลายร้อยดอลลาร์สหรัฐ (ในทศวรรษ 1970) ทำให้การนำคอมพิวเตอร์มาใช้กับเครื่องใช้ไฟฟ้าขนาดเล็กนั้นไม่คุ้มค่าทางเศรษฐกิจ

บริษัท MOS Technology เปิดตัวไมโครโปรเซสเซอร์ราคาต่ำกว่า 100 ดอลลาร์ในปี 1975 ได้แก่รุ่น 6501 และ 6502 จุดประสงค์หลักคือการลดอุปสรรคด้านราคา แต่ไมโครโปรเซสเซอร์เหล่านี้ยังคงต้องการการสนับสนุนจากภายนอก หน่วยความจำ และชิปอุปกรณ์ต่อพ่วง ซึ่งทำให้ต้นทุนรวมของระบบยังคงอยู่ในระดับหลายร้อยดอลลาร์

2.1.1.1 การพัฒนา

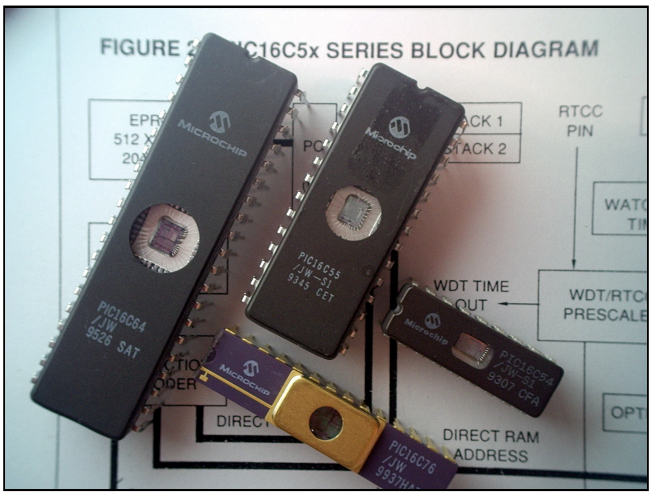
หนังสือเล่มหนึ่งระบุว่า Gary Boone และ Michael Cochran วิศวกรของบริษัท TI ประสบความสำเร็จในการสร้างไมโครคอนโทรลเลอร์ตัวแรกในปี 1971 ผลงานของพวกเขาคือ TMS 1000 ซึ่งวางจำหน่ายในเชิงพาณิชย์ในปี 1974 ไมโครคอนโทรลเลอร์นี้รวมหน่วยความจำแบบอ่านอย่างเดียว หน่วยความจำแบบทั้งอ่านและเขียน โปรเซสเซอร์ และนาฬิกาไว้ในชิปเดียว และมุ่งเป้าไปที่ระบบฝังตัว

ในช่วงต้นถึงกลางทศวรรษ 1970 ผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ของญี่ปุ่นเริ่มผลิตไมโครคอนโทรลเลอร์สำหรับรถยนต์ ซึ่งรวมถึง MCU 4 บิตสำหรับระบบความบันเทิงในรถยนต์ ที่ปัดน้ำฝนอัตโนมัติ ระบบล็อคอิเล็กทรอนิกส์ และแผงหน้าปัด ตลอดจน MCU 8 บิตสำหรับควบคุมเครื่องยนต์

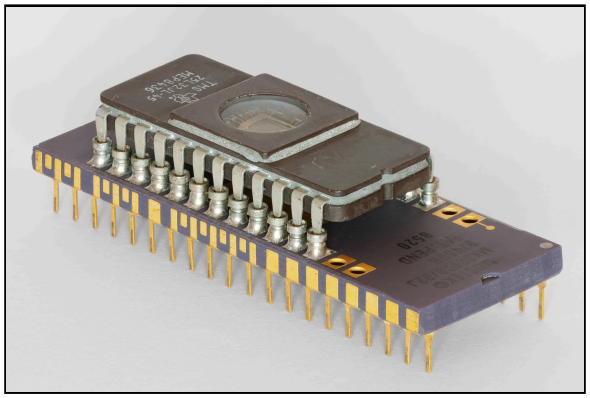
ส่วนหนึ่งเพื่อตอบสนองต่อการมีอยู่ของชิป TMS 1000 แบบชิปเดี่ยว Intel จึงพัฒนาชิปประมวลผลระบบคอมพิวเตอร์บนชิปที่ปรับให้เหมาะสมสำหรับการใช้งานด้านการควบคุม นั่นคือ Intel 8048 โดยเริ่มจัดส่งชิ้นส่วนเชิงพาณิชย์ครั้งแรกในปี 1977 ชิปนี้รวม RAM และ ROM ไว้ในชิปเดียวกันกับไมโครโปรเซสเซอร์ ในบรรดาแอปพลิเคชันมากมาย ชิปนี้ได้ถูกนำไปใช้ในแป้นพิมพ์พีซีมากกว่าหนึ่งพันล้านเครื่องในที่สุด ในเวลานั้น Luke J. Valenter ประธานของ Intel กล่าวว่าไมโคร

คอนโทรลเลอร์เป็นหนึ่งในผลิตภัณฑ์ที่ประสบความสำเร็จมากที่สุดในประวัติศาสตร์ของบริษัท และเขาได้ขยายงบประมาณของแผนกไมโครคอนโทรลเลอร์เพิ่มขึ้นกว่า 25%

ไมโครคอนโทรลเลอร์ส่วนใหญ่ในเวลานั้นมีหลายรุ่นที่ใช้งานพร้อมกัน รุ่นหนึ่งใช้หน่วยความจำโปรแกรม EPROM โดยมีหน้าต่างควอตซ์โปร่งใสอยู่ที่ฝาปิดของตัวชิปเพื่อให้สามารถลบข้อมูลได้โดยการฉายแสงอัลตราไวโอเลต ชิปที่ลบได้เหล่านี้มักใช้สำหรับการสร้างต้นแบบ อีกรุ่นหนึ่งคือ ROM ที่ตั้งโปรแกรมด้วยมาสก์ หรือ PROM ที่ตั้งโปรแกรมได้เพียงครั้งเดียว สำหรับรุ่นหลัง บางครั้งจะใช้คำว่า OTP ซึ่งย่อมาจาก “one-time programmable” (ตั้งโปรแกรมได้ครั้งเดียว) ในไมโครคอนโทรลเลอร์ OTP นั้น PROM มักจะเป็นชนิดเดียวกับ EPROM แต่ตัวชิปไม่มีหน้าต่างควอตซ์ เนื่องจากไม่มีวิธีใดที่จะฉายแสงอัลตราไวโอเลตไปยัง EPROM ได้ จึงไม่สามารถลบข้อมูลได้ เนื่องจากรุ่นที่ลบได้ต้องใช้ตัวชิปเซรามิกที่มีหน้าต่างควอตซ์ จึงมีราคาแพงกว่ารุ่น OTP อย่างมาก ซึ่งสามารถผลิตได้ในตัวชิปพลาสติกทึบแสงที่มีราคาถูกกว่า สำหรับรุ่นที่สามารถลบได้นั้น จำเป็นต้องใช้ควอตซ์แทนกระจกที่มีราคาถูกกว่า เนื่องจากมีความโปร่งใสต่อแสงอัลตราไวโอเลต ซึ่งกระจกส่วนใหญ่ทึบแสง แต่ปัจจัยหลักที่ทำให้ต้นทุนแตกต่างกันคือตัวบรรจุภัณฑ์เซรามิกเอง นอกจากนี้ยังมีการใช้ไมโครคอนโทรลเลอร์แบบ Piggyback ด้วย



รูปที่ 2.1 ไมโครคอนโทรลเลอร์ PIC ต่าง ๆ ที่มี EPROM ภายใน



รูปที่ 2.2 ไมโครคอนโทรลเลอร์ Piggyback จาก MOSTEK

ในปี พ.ศ. 2536 การเปิดตัวหน่วยความจำ EEPROM ทำให้ไมโครคอนโทรลเลอร์ (เริ่มต้นด้วย Microchip PIC16C84) สามารถลบข้อมูลด้วยไฟฟ้าได้อย่างรวดเร็วโดยไม่ต้องใช้แพ็คเกจราคาแพงอย่างที่จำเป็นสำหรับ EPROM ซึ่งช่วยให้สามารถสร้างต้นแบบได้อย่างรวดเร็วและตั้งโปรแกรมในระบบได้ (เทคโนโลยี EEPROM มีมาก่อนหน้านี้ แต่ EEPROM รุ่นก่อนหน้านี้มีราคาแพงกว่าและทนทานน้อยกว่า ทำให้ไม่เหมาะสำหรับไมโครคอนโทรลเลอร์ที่ผลิตจำนวนมากในราคาประหยัด) ในปีเดียวกันนั้น Atmel ได้เปิดตัวไมโครคอนโทรลเลอร์ตัวแรกที่ใช้หน่วยความจำ Flash ซึ่งเป็น EEPROM ชนิดพิเศษ บริษัทอื่น ๆ ก็ได้ดำเนินการตามมาอย่างรวดเร็ว โดยมีทั้งหน่วยความจำทั้งสองประเภท

ปัจจุบันไมโครคอนโทรลเลอร์มีราคาถูกและหาซื้อได้ง่ายสำหรับผู้ที่ชื่นชอบงานอดิเรก โดยมีชุมชนออนไลน์ขนาดใหญ่ที่ให้ความสนใจกับโปรเซสเซอร์บางประเภท

2.1.1.2 ปริมาณและค่าใช้จ่าย

ในปี 2002 ประมาณ 55% ของ CPU ทั้งหมดที่จำหน่ายในโลกเป็นไมโครคอนโทรลเลอร์และไมโครโปรเซสเซอร์ 8 บิต

มีการขายไมโครคอนโทรลเลอร์ 8 บิตมากกว่าสองพันล้านตัวในปี 1997 และจากข้อมูลของ Semico พบว่าไมโครคอนโทรลเลอร์ 8 บิตมากกว่าสี่พันล้านถูกจำหน่ายในปี 2006 ล่าสุด Semico อ้างว่าตลาด MCU เติบโตขึ้น 36.5% ในปี 2010 และ 12% ในปี 2011

บ้านทั่วไปในประเทศที่พัฒนาแล้วมีแนวโน้มที่จะมีไมโครโปรเซสเซอร์อเนกประสงค์เพียงสี่ตัว แต่มีไมโครคอนโทรลเลอร์ประมาณสามโหล รถยนต์ระดับกลางทั่วไปมีไมโครคอนโทรลเลอร์ประมาณ 30 ตัว นอกจากนี้ยังสามารถพบได้ในอุปกรณ์ไฟฟ้าหลายชนิด เช่น เครื่องซักผ้า เตาไมโครเวฟ และโทรศัพท์

ต้นทุนในการผลิตอาจต่ำกว่า 0.10 เหรียญสหรัฐต่อหน่วย

ค่าใช้จ่ายลดลงเมื่อเวลาผ่านไป โดยไมโครคอนโทรลเลอร์ 8 บิตที่ถูกที่สุดมีจำหน่ายในราคาต่ำกว่า 0.03 ดอลลาร์สหรัฐฯ ในปี 2018 และไมโครคอนโทรลเลอร์ 32 บิตบางรุ่นมีราคาประมาณ 1 ดอลลาร์สหรัฐฯ สำหรับปริมาณที่ใกล้เคียงกัน

ในปี 2012 หลังเกิดวิกฤติทั่วโลก ยอดขายลดลงและการฟื้นตัวต่อปีที่เลวร้ายที่สุดเท่าที่เคยมีมา และราคาขายเฉลี่ยเมื่อเทียบเป็นรายปีลดลง 17% ซึ่งถือเป็นการลดลงครั้งใหญ่ที่สุดนับตั้งแต่ทศวรรษ 1980 ราคาเฉลี่ยสำหรับไมโครคอนโทรลเลอร์อยู่ที่ 0.88 เหรียญสหรัฐฯ (0.69 เหรียญสหรัฐฯ สำหรับ 4/8 บิต, 0.59 เหรียญสหรัฐฯ สำหรับ 16 บิต, 1.76 เหรียญสหรัฐฯ สำหรับ 32 บิต)

ในปี 2012 ยอดขายไมโครคอนโทรลเลอร์ 8 บิตทั่วโลกมีมูลค่าประมาณ 4 พันล้านดอลลาร์สหรัฐ ในขณะที่ไมโครคอนโทรลเลอร์ 4 บิตก็มียอดขายที่สำคัญเช่นกัน

ในปี 2015 ไมโครคอนโทรลเลอร์ 8 บิตสามารถซื้อได้ในราคา 0.311 ดอลลาร์สหรัฐฯ (1,000 หน่วย) 16 บิตราคา 0.385 ดอลลาร์สหรัฐฯ (1,000 หน่วย) และ 32 บิตในราคา 0.378 ดอลลาร์สหรัฐฯ (1,000 หน่วย แต่อยู่ที่ 0.35 ดอลลาร์สหรัฐฯ สำหรับ 5,000)

ในปี 2018 ไมโครคอนโทรลเลอร์ 8 บิตสามารถซื้อได้ในราคา 0.03 ดอลลาร์สหรัฐฯ 16 บิตในราคา 0.393 ดอลลาร์สหรัฐฯ (1,000 หน่วย แต่ราคา 0.563 ดอลลาร์สหรัฐฯ สำหรับ 100 หน่วยหรือ 0.349 ดอลลาร์สหรัฐฯ สำหรับม้วนเต็ม 2,000 หน่วย) และ 32 บิตในราคา 0.503 ดอลลาร์สหรัฐฯ (1,000 หน่วย แต่ที่ 0.466 ดอลลาร์สหรัฐสำหรับ 5,000 หน่วย)

ในปี 2018 ไมโครคอนโทรลเลอร์ราคาถูกที่สูงกว่าปี 2015 ทั้งหมดมีราคาแพงกว่า (โดยคำนวณอัตราเงินเฟ้อระหว่างราคาปี 2018 ถึง 2015 สำหรับหน่วยเฉพาะเหล่านั้น) โดยไมโครคอนโทรลเลอร์ 8 บิตสามารถซื้อได้ในราคา 0.319 ดอลลาร์สหรัฐฯ (1,000 หน่วย) หรือสูงกว่า 2.6%

ไมโครคอนโทรลเลอร์ 16 บิตมีราคา 0.464 ดอลลาร์สหรัฐฯ (1,000 หน่วย) หรือ 21% สูงกว่า

แบบ 32 บิตในราคา 0.503 ดอลลาร์สหรัฐฯ (1,000 หน่วย แต่อยู่ที่ 0.466 ดอลลาร์สหรัฐฯ สำหรับ 5,000) หรือสูงกว่า 33%

2.1.1.3 คอมพิวเตอร์ที่เล็กที่สุด

เมื่อวันที่ 21 มิถุนายน 2018 มหาวิทยาลัยมิชิแกนได้ประกาศ “คอมพิวเตอร์ที่เล็กที่สุดในโลก” อุปกรณ์ดังกล่าวเป็น “ระบบเซ็นเซอร์ไร้สายและไร้แบตเตอรี่ขนาด 0.04 ลบ.มม. 16 nW พร้อมด้วยโปรเซสเซอร์ Cortex-M0+ ในตัวและการสื่อสารแบบออปติกสำหรับการวัดอุณหภูมิของเซลล์” “วัดด้านข้างเพียง 0.3 มม. ประมาณขนาดเมล็ดข้าว [...] นอกเหนือจาก RAM และเซลล์แสงอาทิตย์แล้ว อุปกรณ์ คอมพิวเตอร์รุ่นใหม่ยังมีโปรเซสเซอร์และเครื่องส่งและตัวรับสัญญาณไร้สาย เนื่องจากมีขนาดเล็กเกินไปที่จะมีเสาอากาศวิทยุแบบธรรมดา อุปกรณ์จึงรับและส่งข้อมูลด้วยแสงที่มองเห็นได้ สถานีฐานให้แสงสำหรับพลังงานและการเขียนโปรแกรม และรับข้อมูล” อุปกรณ์นี้มีขนาด 1/10 ของขนาดที่ IBM อ้างสิทธิ์ก่อนหน้านี้ คอมพิวเตอร์ที่มีขนาดเป็นสถิติโลกเมื่อหลายเดือนก่อนในเดือนมีนาคม 2018 ซึ่ง “เล็กกว่าเม็ดเกลือ” มีทรานซิสเตอร์หนึ่งล้านตัว ต้นทุนการผลิตน้อยกว่า 0.10 ดอลลาร์ และเมื่อรวมกับเทคโนโลยีบล็อกเชนแล้ว มีไว้สำหรับการขนส่งและ “จุดยึดเข้ารหัสลับ” ซึ่งเป็นแอปพลิเคชันลายนิ้วมือดิจิทัล

2.1.2 ประเภท

ณ ปี 2008 มีผู้ขายและสถาปัตยกรรมไมโครคอนโทรลเลอร์จำนวนมาก รวมไปถึง:

1) หน่วยประมวลผล ARM core โดยเฉพาะคอร์ประเภท ARM Cortex-M

2) Microchip Technology Atmel AVR (8 บิต), AVR32 (32 บิต), และ AT91SAM (32 บิต)

3) คอร์ M8C ของ Cypress Semiconductor’s ที่ถูกใช้ใน Cypress PSoC ของพวกเขา

4) Freescale ColdFire (32 บิต) และ S08 (8 บิต)

5) Freescale 68HC11 (8 บิต) และอื่น ๆ ที่มีรากฐานมาจากครอบครัว Motorola 6800

6) Intel 8051, ซึ่งนอกจาก Intel ก็ถูกผลิตโดย NXP Semiconductors, Infineon, และอื่น ๆ หลายรายการ

7) Infineon: 8 บิต XC800, 16 บิต XE166, 32 บิต XMC4000 (ARM based Cortex M4F), 32 บิต TriCore, และ 32 บิต Aurix Tricore Bit microcontrollers

8) Maxim Integrated MAX32600, MAX32620, MAX32625, MAX32630, MAX32650, MAX32640

9) MIPS

10) Microchip Technology PIC, (8 บิต PIC16, PIC18, 16 บิต dsPIC33 / PIC24), (32 บิต PIC32)

11) NXP Semiconductors LPC1000, LPC2000, LPC3000, LPC4000 (32 บิต),

LPC900, LPC700 (8 บิต)

12) Parallax Propeller

13) PowerPC ISE

14) Rabbit 2000 (8 บิต)

15) Renesas Electronics: RL78 16 บิต MCU; RX 32 บิต MCU; SuperH; V850 32 บิต MCU; H8; R8C 16 บิต MCU

16) Silicon Laboratories ไมโครคอนโทรลเลอร์ Pipelined 8 บิต 8051 และไมโครคอนโทรลเลอร์แบบ ARM-based 32 บิต สัญญาณผสม

17) STMicroelectronics STM8 (8 บิต), ST10 (16 บิต), STM32 (32 บิต), SPC5 (automotive 32 บิต)

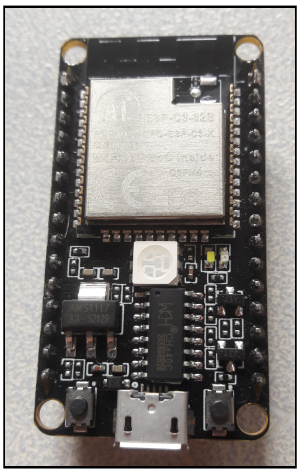
18) Texas Instruments TI MSP430 (16 บิต), MSP432 (32 บิต), C2000 (32 บิต)

19) Toshiba TLCS-870 (8 บิต/16 บิต)

และยังมีอีกมากมาย โดยบางอย่างนั้นถูกใช้ในแอปพลิเคชันที่เจาะจงมาก หรือเหมือนกับหน่วยประมวลผลเฉพาะแอปพลิเคชันมากกว่าไมโครคอนโทรลเลอร์ ตลาดไมโครคอนโทรลเลอร์นั้นกระจัดกระจายเป็นอย่างมาก และมีผู้ขาย เทคโนโลยี และตลาดมากมาย และผู้ขายจำนวนมากขายหลายสถาปัตยกรรม

2.1.3 ESP32

ESP32 คือกลุ่มไมโครคอนโทรลเลอร์ราคาประหยัดและประหยัดพลังงานที่ผสานรวมความสามารถทั้ง Wi-Fi และบลูทูธ ชิปเหล่านี้มีตัวเลือกการประมวลผลที่หลากหลาย รวมถึงไมโครโปรเซสเซอร์ Tensilica Xtensa LX6 ที่มีให้เลือกทั้งแบบ dual-core และ single-core, โปรเซสเซอร์ Xtensa LX7 dual-core หรือไมโครโปรเซสเซอร์ RISC-V แบบ single-core นอกจากนี้ ESP32 ยังรวมส่วนประกอบที่จำเป็นสำหรับการสื่อสารข้อมูลไร้สาย เช่น สวิตช์เสาอากาศในตัว บาลัน RF เครื่องขยายกำลัง เครื่องรับสัญญาณรบกวนต่ำ ตัวกรอง และโมดูลการจัดการพลังงาน



รูปที่ 2.3 บอร์ด NodeMCU ที่มี ESP32-C3-32S

โดยทั่วไปแล้ว ESP32 จะถูกฝังอยู่บนแผงวงจรพิมพ์เฉพาะอุปกรณ์หรือนำเสนอเป็นส่วนหนึ่งของชุดการพัฒนาที่มีพินและตัวเชื่อมต่อ GPIO ที่หลากหลาย โดยมีการกำหนดค่าที่แตกต่างกันไปตามรุ่นและผู้ผลิต ESP32 ได้รับการออกแบบโดย Espressif Systems และผลิตโดย TSMC โดยใช้กระบวนการ 40 นาโนเมตร มันเป็นผู้สืบทอดของไมโครคอนโทรลเลอร์ ESP8266

2.1.3.1 Espressif Systems

บริษัท Espressif Systems (Shanghai) จำกัด (เอสเพรสซิฟ) เป็นบริษัทเซมิคอนดักเตอร์สัญชาติจีนที่จดทะเบียนในตลาดหลักทรัพย์ มีสำนักงานใหญ่ตั้งอยู่ที่เซี่ยงไฮ้ บริษัทมุ่งเน้นการพัฒนาและจำหน่ายชิปและโมดูลสื่อสารไมโครคอนโทรลเลอร์ไร้สายที่ใช้ใน Internet of things (IoT)

ผลิตภัณฑ์เซมิคอนดักเตอร์ที่เป็นที่รู้จักมากที่สุดสองอย่างของ Espressif คือ ESP8266 และ ESP32 ซึ่งเป็นรุ่นต่อจาก ESP8266 ผลิตภัณฑ์เหล่านี้ถูกนำไปใช้ในผลิตภัณฑ์ต่างๆ เช่น เครื่องชงกาแฟและหลอดไฟ รวมถึงผู้ให้บริการโซลูชันเมืองอัจฉริยะและระบบอัตโนมัติ นอกจากนี้ยังถูกใช้โดยผู้ที่ชื่นชอบงาน DIY ด้านเทคโนโลยีอีกด้วย

2.1.4 ตารางพาร์ทิชัน (Partition Table)

ตารางพาร์ทิชันคือสิ่งที่กำหนดการจัดการรูปแบบหน่วยความจำแฟลชและข้อมูลต่าง ๆ จะถูกเก็บไว้ในแต่ละพาร์ทิชัน โดยผู้พัฒนาสามารถใช้รูปแบบตารางพาร์ทิชันที่ถูกกำหนดมาไว้แล้วหรือสามารถกำหนดรูปแบบตารางพาร์ทิชันเองก็ได้ โดยตารางพาร์ทัชันที่ถูกใช้ในโครงงานนี้มีรูปแบบดังนี้

ตารางที่ 2.1 รายการพาร์ทิชัน

Name	Type	SubType	Offset	Size	Flags
nvs	data	nvs	0x9000	0x5000	
otadata	data	ota	0xe000	0x2000	
app0	app	ota_0	0x10000	0x140000	
app1	app	ota_1	0x150000	0x140000	
spiffs	data	spiffs	0x290000	0x160000	
coredump	data	coredump	0x3F0000	0x10000	

ซึ่งคือตารางค่าเริ่มต้นของ ESP32 ใน Arduino platform อย่างไรก็ตามมีการเปลี่ยนแปลงระบบเก็บไฟล์จาก SPIFFS เป็น LittleFS โดยที่

2.1.4.1 Name

Name คือ ชื่อของพาร์ทิชัน ห้ามซ้ำกัน ชื่อนั้นไม่สำคัญต่อระบบและต้องขนาดไม่เกิน 16 ตัวอักษร (ไม่มีอักขระพิเศษ)

2.1.4.2 Type

Type คือ ประเภทของพาร์ทิชัน สามารถเป็น data หรือ app ได้

1) app คือพาร์ทิชันที่ใช้ในการเก็บแอปพลิเคชัน
2) data คือพาร์ทิชันที่ใช้ในการเก็บข้อมูลทั่วไป

2.1.4.3 SubType

SubType คือ ประเภทย่อย ระบุการใช้งานของพาร์ทิชัน data และ app

1) data

ก) ota คือ พาร์ทิชันเก็บข้อมูล OTA (สำหรับการอัพเดททางอากาศ, Over-the-air update) โดยหากไม่ใช้งาน OTA สามารถนำออกได้ โดยขนาดของพาร์ทิชันนี้ควรจะมีขนาดที่แน่นอนอยู่ที่ 8 KiB (0x2000 ไบต์)

ข) nvs คือ พาร์ทิชันเก็บข้อมูลทั่วไปเช่น ข้อมูล Wi-Fi, ข้อมูลการสอบเทียบ PHY ของอุปกรณ์, และข้อมูลอื่น ๆ ที่ต้องถูกเก็บบนหน่วยความจำถาวร (Non-volatile memory) โดยพาร์ทิชันประเภทนี้เหมาะสมสำหรับการเก็บข้อมูลการตั้งค่าเล็กน้อย ใบรองรับคลาวด์ ฯลฯ และการใช้งาน NVS อีกอย่างคือการเก็บข้อมูลที่ละเอียดอ่อน เนื่องจาก NVS รองรับการเข้ารหัส และเป็นสิ่งที่แนะนำอย่างมากที่จะมีพาร์ทิชัน NVS ขนาดขั้นต่ำ 12 KiB (0x3000 ไบต์) และหากจำเป็น คุณสามารถขยายขนาดเพิ่มได้ โดยขนาดที่แนะนำนั้นอยู่ระหว่าง 12 KiB และ 64 KiB ถึงแม้ว่าคุณจะสามารถขยายให้มันใหญ่กว่านี้ได้ การใช้งานระบบไฟล์เช่น FAT หรือ SPIFFS นั้นจะเหมาะสมสำหรับข้อมูลที่ใหญ่กว่า

ค) coredump คือ ประเภทพาร์ทิชันย่อยนี้มีหน้าที่ในการเก็บข้อมูล core dump บนหน่วยความจำแฟลช โดย core dump นั้นคือข้อมูลที่ถูกใช้งานสำหรับการตรวจสอบข้อผิด-พลาดร้ายแรงเช่นการแครชและแพนิค โดยฟังก์ชันนี้จะต้องถูกเปิดในการตั้งค่าโปรเจกต์และตั้งที่หมายในการแฟลช และพาร์ทิชันนี้มีขนาดที่แนะนำอยู่ที่ 64 KiB (0x10000)

ง) nvs_keys คือ พาร์ทิชันที่เป็นประเภทย่อยนี้เก็บคีย์การเข้ารหัสของพาร์ทิชัน NVS เมื่อการเข้ารหัสถูกใช้งาน โดยมีขนาดอยู่ที่ 4 KiB (0x1000)

จ) fat คือ กำหนดพาร์ทิชันสำหรับระบบไฟล์ FAT โดยที่จะเหมาะสมสำหรับข้อมูลใหญ่ ๆ และหากข้อมูลนั้นถูกเปลี่ยนแปลงบ่อย โดยระบบไฟล์ FAT สามารถใช้ฟีเจอร์ wear leveling และการเข้ารหัสได้

ฉ) spiffs คือ กำหนดพาร์ทิชันสำหรับระบบไฟล์ SPIFFS เหมาะสำหรับไฟล์ใหญ่เช่นกันและรองรับ wear leveling อย่างไรก็ตาม ระบบไฟล์นี้ไม่รองรับการเข้ารหัส

2) app

ก) factory คือ พาร์ทิชันเก็บแอปพลิเคชันเริ่มต้น โปรแกรมบูตโหลดเดอร์จะเลือกพาร์ทิชันนี้เป็นแอปพลิเคชันเริ่มต้นหากไม่มีพาร์ทิชัน OTA หรือพาร์ทิชัน OTA นั้นว่างเปล่า หากมีการใช้พาร์ทิชัน OTA พาร์ทิชัน ota_0 สามารถถูกใช้เป็นแอปพลิเคชันเริ่มต้นได้และพาร์ทิชัน factory สามารถถูกนำออกได้

ข) ota_0 ถึง ota_15 คือ พาร์ทิชัน ota_x นั้นถูกใช้สำหรับอัพเดท OTA โดยฟีเจอร์ OTA นั้นจำเป็นต้องใช้พาร์ทิชัน OTA อย่างน้อย 2 พาร์ทิชัน (โดยปกติคือ ota_0 และ ota_1) และจำเป็นต้องใช้พาร์ทิชัน ota ด้วยเช่นกันในการเก็บข้อมูลเกี่ยวกับ OTA โดยสามารถมีพาร์ทิชัน OTA ได้สูงสุด 16 พาร์ทิชัน แต่ 2 พาร์ทิชันคือจำนวนขั้นต่ำที่ต้องใช้สำหรับฟีเจอร์ OTA แบบเบสิค

ค) test คือ ใช้สำหรับการทดสอบในโรงงาน

2.1.4.4 Offset

Offset คือ กำหนดพื้นที่ที่พาร์ทิชันนั้น ๆ เริ่มต้น โดย Offset นั้นถูกกำหนดโดยการรวมค่า Offset และขนาดของพาร์ทิชันก่อนหน้า 0 อย่างไรก็ตาม Offset จะต้องเป็นทวีคูณของ 4 KiB (0x1000) และพาร์ทิชันแอพจะต้องจัดตำแหน่งให้มีขนาด 64 KiB (0x10000) โดยหากปล่อยให้ว่าง ค่า Offset จะถูกคำนวนโดยอัตโนมัติตามตำแหน่งท้ายของพาร์ทิชันก่อนหน้า รวมถึงการจัดตำแหน่งใด ๆ

ที่จำเป็น อย่างไรก็ตาม Offset ของพาร์ทิชันแรกนั้นจะต้องเป็น 0x9000 และ 0x10000 สำหรับพาร์ทิชันแอปพลิเคชันแรก

2.1.4.5 Size

Size คือ ขนาดของพาร์ทิชัน โดยค่านี้สามารถเป็นเลขทศนิยม, ตัวเลข Hex (นำหน้าด้วย 0x), หรือใช้ตัวอักษรต่อท้ายเพื่อบ่งบอกหน่วย K (กิโล) หรือ M (เมกา) เช่น 4096 = 4K = 0x1000

2.1.4.6 Flags

Flags คือ ในปัจจุบันคอลัมน์นี้ใช้เพียงแค่เพื่อบ่งบอกว่าพาร์ทิชันนั้น ๆ ถูกเข้ารหัสหรือไม่

2.1.5 littlefs

littlefs คือระบบไฟล์ขนาดเล็กที่ปลอดภัยต่อความล้มเหลวที่ออกแบบมาสำหรับไมโครคอนโทรลเลอร์

ความยืดหยุ่นในการป้องกันการสูญเสียพลังงาน littlefs ออกแบบมาเพื่อรับมือกับปัญหาไฟฟ้าดับแบบสุ่ม การดำเนินการไฟล์ทั้งหมดมีการรับประกันการคัดลอกข้อมูลเมื่อเขียนข้อมูล (copy-on-write) ที่แข็งแกร่ง และหากไฟฟ้าดับ ระบบไฟล์จะกลับสู่สถานะปกติล่าสุดที่ทราบ

การปรับระดับการสึกหรอแบบไดนามิก littlefs ออกแบบมาเพื่อแฟลชโดยเฉพาะ และมอบการปรับระดับการสึกหรอบนบล็อกแบบไดนามิก นอกจากนี้ littlefs ยังสามารถตรวจจับบล็อกเสียและแก้ไขปัญหาได้

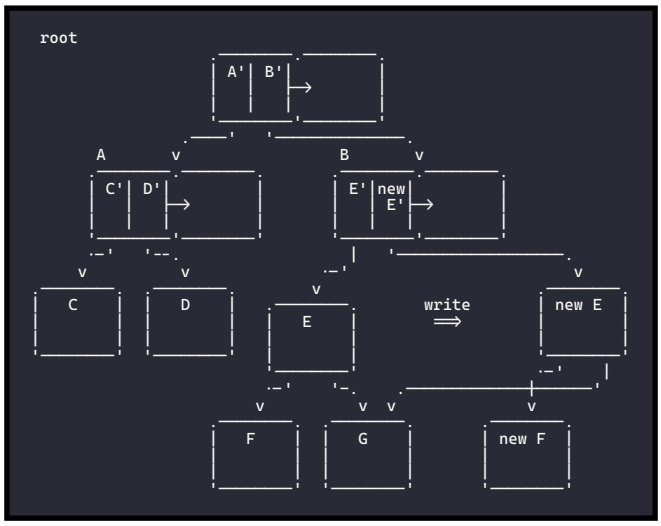
RAM/ROM แบบมีขอบเขต littlefs ออกแบบมาเพื่อทำงานกับหน่วยความจำขนาดเล็ก การใช้งาน RAM ถูกจำกัดอย่างเข้มงวด ซึ่งหมายความว่าการใช้ RAM จะไม่เปลี่ยนแปลงเมื่อระบบไฟล์เติบโตขึ้น ระบบไฟล์ไม่มีการเรียกซ้ำแบบไม่มีขอบเขต และหน่วยความจำแบบไดนามิกถูกจำกัดให้อยู่ในบัฟเฟอร์ที่กำหนดค่าได้ซึ่งสามารถจัดเตรียมแบบคงที่ได้

2.1.6 ออกแบบ

ในระดับสูง littlefs เป็นระบบไฟล์แบบบล็อกที่ใช้ไฟล์บันทึกขนาดเล็กในการจัดเก็บข้อมูลเมตาและโครงสร้าง copy-on-write (COW) ขนาดใหญ่ในการจัดเก็บข้อมูลไฟล์

ใน littlefs ส่วนผสมเหล่านี้ก่อตัวเป็นเค้กสองชั้น โดยที่ท่อนไม้ขนาดเล็ก (เรียกว่าคู่เมตาเดตา) จะให้การอัปเดตเมตาเดตาอย่างรวดเร็วในทุกที่ในที่เก็บข้อมูล ในขณะที่โครงสร้าง COW จะจัดเก็บข้อมูลไฟล์อย่างกะทัดรัดและไม่มีค่าใช้จ่ายในการขยายการสึกหรอใด ๆ

โครงสร้างข้อมูลทั้งสองนี้สร้างขึ้นจากบล็อก ซึ่งถูกป้อนโดยตัวจัดสรรบล็อกร่วม โดยการจำกัดจำนวนการลบข้อมูลที่อนุญาตบนบล็อกต่อการจัดสรรแต่ละครั้ง ตัวจัดสรรจะปรับระดับการสึกหรอแบบไดนามิกทั่วทั้งระบบไฟล์



รูปที่ 2.4 แสดงการทำงานเบื้องต้นของ LittleFS

2.2 เซนเซอร์ (Sensors)

โดยทั่วไปแล้ว เซนเซอร์จะถูกนิยามว่าเป็นอุปกรณ์ที่รับและตอบสนองต่อสัญญาณหรือสิ่งเร้าสิ่งเร้าคือปริมาณ คุณสมบัติ หรือสภาวะที่ถูกตรวจจับและแปลงเป็นสัญญาณไฟฟ้า

ในความหมายกว้างที่สุด เซนเซอร์คืออุปกรณ์ โมดูล เครื่องจักร หรือระบบย่อยที่ตรวจจับเหตุการณ์หรือการเปลี่ยนแปลงในสภาพแวดล้อม และส่งข้อมูลไปยังอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ซึ่งส่วนใหญ่มักจะเป็นหน่วยประมวลผลของคอมพิวเตอร์

เซ็นเซอร์ถูกนำมาใช้ในสิ่งของในชีวิตประจำวัน เช่น ปุ่มลิฟต์แบบสัมผัส (เซ็นเซอร์สัมผัส) และโคมไฟที่หรี่หรือสว่างขึ้นโดยการสัมผัสที่ฐาน และในแอปพลิเคชันมากมายนับไม่ถ้วนซึ่งคนส่วนใหญ่ไม่เคยตระหนักถึง ด้วยความก้าวหน้าในด้านไมโครแมชชีนเนอรี่ และแพลตฟอร์ม ไมโครคอนโทรลเลอร์ที่ใช้งานง่ายการใช้งานเซ็นเซอร์จึงขยายออกไปนอกเหนือจากสาขาแบบดั้งเดิมของการวัดอุณหภูมิ ความดัน และการไหลตัวอย่างเช่น ไปสู่เซ็นเซอร์ MARG

เซ็นเซอร์แบบอนาล็อก เช่นโพเทนชิโอมิเตอร์และตัวต้านทานรับแรงยังคงมีการใช้งานอย่างแพร่หลาย การใช้งานของเซ็นเซอร์เหล่านี้รวมถึงการผลิตและเครื่องจักร เครื่องบินและอวกาศ รถยนต์ การแพทย์ หุ่นยนต์และอีกหลายแง่มุมในชีวิตประจำวันของเรา นอกจากนี้ยังมีเซ็นเซอร์อื่นๆ อีกมากมายที่ใช้วัดคุณสมบัติทางเคมีและกายภาพของวัสดุ รวมถึงเซ็นเซอร์แบบออปติคอลสำหรับการวัดดัชนีหักเห เซ็นเซอร์แบบสั่นสะเทือนสำหรับการวัดความหนืดของของเหลว และเซ็นเซอร์ทางเคมีไฟฟ้าสำหรับการตรวจสอบค่า pH ของของเหลว

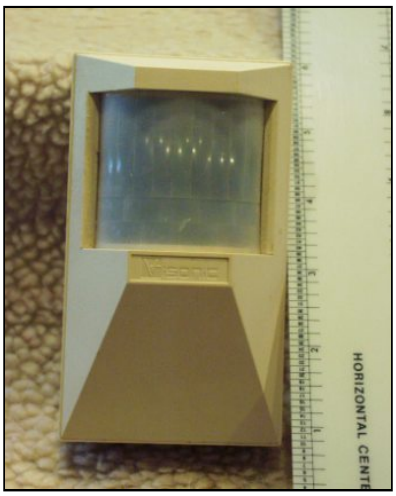
ความไวของเซนเซอร์บ่งชี้ว่าเอาต์พุตเปลี่ยนแปลงไปมากน้อยเพียงใดเมื่อปริมาณอินพุตที่วัดเปลี่ยนแปลง ตัวอย่างเช่น หากปรอทในเทอร์โมมิเตอร์เคลื่อนที่ 1 ซม. เมื่ออุณหภูมิเปลี่ยนแปลง 1 °C ความไวของมันคือ 1 ซม./°C (โดยพื้นฐานแล้วคือความชัน dy/dx โดยสมมติว่ามีลักษณะเชิงเส้น) เซนเซอร์บางชนิดอาจส่งผลต่อสิ่งที่วัดได้เช่นกัน ตัวอย่างเช่น เทอร์โมมิเตอร์วัดอุณหภูมิห้องที่เสียบลงในถ้วยของเหลวร้อนจะทำให้ของเหลวเย็นลงในขณะที่ของเหลวทำให้เทอร์โมมิเตอร์ร้อนขึ้น โดยทั่วไป

เซนเซอร์ได้รับการออกแบบให้มีผลกระทบต่อสิ่งที่วัดน้อยที่สุด การทำให้เซนเซอร์มีขนาดเล็กลงมักจะช่วยปรับปรุงสิ่งนี้และอาจนำมาซึ่งข้อดีอื่นๆ

ความก้าวหน้าทางเทคโนโลยีทำให้สามารถผลิตเซ็นเซอร์ได้มากขึ้นเรื่อย ๆ ในระดับจุลภาคเช่น ไมโครเซ็นเซอร์โดยใช้เทคโนโลยี MEMS ในกรณีส่วนใหญ่ ไมโครเซ็นเซอร์สามารถวัดได้เร็วกว่าและมีความไวสูงกว่าเมื่อเทียบกับวิธีการแบบมหภาคเนื่องจากความต้องการข้อมูลที่รวดเร็ว ราคาไม่แพง และเชื่อถือได้เพิ่มมากขึ้นในโลกปัจจุบัน เซ็นเซอร์แบบใช้แล้วทิ้ง ซึ่งเป็นอุปกรณ์ราคาถูกและใช้งานง่ายสำหรับการตรวจสอบระยะสั้นหรือการวัดแบบครั้งเดียว จึงได้รับความสำคัญเพิ่มมากขึ้น การใช้เซ็นเซอร์ประเภทนี้ทำให้ทุกคนสามารถรับข้อมูลการวิเคราะห์ที่สำคัญได้ทุกที่ ทุกเวลา โดยไม่จำเป็นต้องปรับเทียบใหม่และไม่ต้องกังวลเรื่องการปนเปื้อน

2.2.1 เซ็นเซอร์อินฟราเรดแบบพาสซีฟ (PIR sensor)

เซ็นเซอร์อินฟราเรดแบบพาสซีฟ (PIR sensor) คือ เซ็นเซอร์อิเล็กทรอนิกส์ที่วัดแสงอินฟราเรด (IR) ที่แผ่ออกมาจากวัตถุในระยะการมองเห็น เซ็นเซอร์ชนิดนี้มักใช้ในเครื่องตรวจจับความเคลื่อนไหวแบบ PIR เซ็นเซอร์ PIR มักใช้ในสัญญาณเตือนภัยและระบบไฟส่องสว่างอัตโนมัติ



รูปที่ 2.5 เครื่องตรวจจับการเคลื่อนไหวแบบ PIR ทั่วไปสำหรับที่พักอาศัย/เชิงพาณิชย์

เซ็นเซอร์ PIR ตรวจจับการเคลื่อนไหวทั่วไป แต่ไม่ได้ให้ข้อมูลว่าใครหรือสิ่งใดเคลื่อนไหว ดังนั้น จึงจำเป็นต้องใช้ เซ็นเซอร์ IR แบบสร้างภาพ เซ็นเซอร์ PIR มักเรียกสั้นๆ ว่า “PIR” หรือบางครั้งเรียกว่า “PID” ซึ่งย่อมาจาก “เครื่องตรวจจับอินฟราเรดแบบพาสซีฟ” เซ็นเซอร์ PIR ตรวจจับการเคลื่อนไหวทั่วไป แต่ไม่ได้ให้ข้อมูลว่าใครหรือสิ่งใดเคลื่อนไหว ดังนั้น จึงจำเป็นต้องใช้ เซ็นเซอร์ IR แบบสร้างภาพ เซ็นเซอร์ PIR มักเรียกสั้นๆ ว่า “PIR” หรือบางครั้งเรียกว่า “PID” ซึ่งย่อมาจาก “เครื่องตรวจจับอินฟราเรดแบบพาสซีฟ” คำว่าพาสซีฟหมายถึงข้อเท็จจริงที่ว่าอุปกรณ์ PIR ไม่ได้แผ่พลังงานเพื่อจุดประสงค์ในการตรวจจับ แต่ทำงานโดยการตรวจจับรังสีอินฟราเรด (ความร้อนจากการแผ่รังสี) ที่แผ่ออกมาจากหรือสะท้อนจากวัตถุ เท่านั้นซีฟ“ คำว่าพาสซีฟหมายถึงข้อเท็จจริงที่ว่าอุปกรณ์ PIR ไม่ได้แผ่พลังงานเพื่อจุดประสงค์ในการตรวจจับ แต่ทำงานโดยการตรวจจับรังสีอินฟราเรด (ความร้อนจากการแผ่รังสี) ที่แผ่ออกมาจากหรือสะท้อนจากวัตถุ เท่านั้น

2.2.1.1 หลักการทำงาน

วัตถุทุกชนิดที่มีอุณหภูมิสูงกว่าศูนย์องศาสัมบูรณ์จะปล่อยพลังงานความร้อนออกมาในรูปของรังสีแม่เหล็กไฟฟ้า โดยปกติแล้วรังสีนี้มองไม่เห็นด้วยตาเปล่าเนื่องจากแผ่รังสีในช่วงความยาวคลื่นอินฟราเรด แต่อุปกรณ์อิเล็กทรอนิกส์ที่ออกแบบมาเพื่อจุดประสงค์นี้ สามารถตรวจจับได้

2.2.1.2 เครื่องตรวจจับการเคลื่อนไหวแบบ PIR

รูปที่ 2.6 เครื่องตรวจจับความเคลื่อนไหว PIR ใช้สำหรับควบคุมไฟภายนอกอาคารแบบอัตโนมัติ

รูปที่ 2.7 กล้องดักถ่ายพร้อมระบบตรวจจับความเคลื่อนไหวแบบ PIR

รูปที่ 2.8 สวิตช์ไฟภายในอาคารที่ติดตั้งเซนเซอรตรวจจับการครอบครองแบบ PIR

เครื่องตรวจจับความเคลื่อนไหวแบบ PIR ใช้เพื่อตรวจจับการเคลื่อนไหวของคน สัตว์ หรือวัตถุอื่นๆ มักใช้กับสัญญาณกันขโมยและระบบไฟส่องสว่างแบบอัตโนมัติ

2.2.1.3 การดำเนินการ

เซ็นเซอร์ PIR สามารถตรวจจับการเปลี่ยนแปลงของปริมาณรังสีอินฟราเรดที่กระทบกับวัตถุ ซึ่งจะแตกต่างกันไปขึ้นอยู่กับอุณหภูมิและลักษณะพื้นผิวของวัตถุที่อยู่ด้านหน้าเซ็นเซอร์เมื่อวัตถุ เช่น บุคคล ผ่านด้านหน้าพื้นหลัง เช่น กำแพง อุณหภูมิ ณ จุดนั้นในมุมมองของเซ็นเซอร์จะเพิ่มขึ้นจากอุณหภูมิห้องเป็นอุณหภูมิร่างกายแล้วกลับมาอีกครั้ง เซ็นเซอร์จะแปลงการเปลี่ยนแปลงที่เกิดขึ้นของรังสีอินฟราเรดที่เข้ามาเป็นการเปลี่ยนแปลงของแรงดันไฟฟ้าขาออก และสิ่งนี้จะกระตุ้นการตรวจจับ วัตถุที่มีอุณหภูมิใกล้เคียงกันแต่มีลักษณะพื้นผิวต่างกันอาจมีรูปแบบการปล่อยรังสีอินฟราเรดที่แตกต่างกัน ดังนั้นการเคลื่อนย้ายวัตถุเทียบกับพื้นหลังอาจกระตุ้นเครื่องตรวจจับได้เช่นกัน

PIR มีหลายรูปแบบการใช้งานที่หลากหลาย รุ่นที่นิยมใช้กันมากที่สุดมีเลนส์เฟรสเนลหรือส่วนกระจกจำนวนมาก ระยะการทำงานประมาณ 10 เมตร (30 ฟุต) และมุมมองภาพน้อยกว่า 180° มีรุ่นที่มีมุมมองภาพกว้างกว่า รวมถึง 360° ซึ่งโดยทั่วไปออกแบบมาเพื่อติดตั้งบนเพดาน PIR ขนาดใหญ่บางรุ่นผลิตด้วยกระจกส่วนเดียวและสามารถตรวจจับการเปลี่ยนแปลงของพลังงานอินฟราเรดได้ในระยะ 30 เมตร (100 ฟุต) จาก PIR นอกจากนี้ยังมี PIR ที่ออกแบบด้วยกระจกแบบปรับทิศทางได้ ซึ่งสามารถครอบคลุมพื้นที่ได้กว้าง (110°) หรือครอบคลุมพื้นที่แคบมากแบบ “ม่าน” หรือสามารถเลือกส่วนกระจกแยกแต่ละส่วนเพื่อ “ปรับแต่ง” พื้นที่ครอบคลุมได้

2.2.1.4 การตรวจจับความแตกต่าง

เซ็นเซอร์หลายตัวอาจเชื่อมต่อเป็นอินพุตตรงข้ามกับเครื่องขยายสัญญาณดิฟเฟอเรนเชียล ในรูปแบบนี้ การวัดค่า PIR จะหักล้างกันเอง ทำให้อุณหภูมิเฉลี่ยของระยะการมองเห็นถูกตัดออกจากสัญญาณไฟฟ้า การเพิ่มขึ้นของพลังงานอินฟราเรดทั่วทั้งเซ็นเซอร์จะหักล้างตัวเองและจะไม่กระตุ้นอุปกรณ์ วิธีนี้ช่วยให้อุปกรณ์ต้านทานการเปลี่ยนแปลงที่ผิดพลาดในกรณีที่ได้รับแสงแฟลชสั้นๆ หรือแสงที่ส่องสว่างทั่วทั้งสนาม (การได้รับพลังงานสูงอย่างต่อเนื่องอาจทำให้วัสดุเซ็นเซอร์อิ่มตัวและทำให้เซ็นเซอร์ไม่สามารถบันทึกข้อมูลเพิ่มเติมได้) ในขณะเดียวกัน การจัดเรียงแบบดิฟเฟอเรนเชียลนี้ยังช่วยลดสัญญาณรบกวนโหมดทั่วไปทำให้อุปกรณ์ต้านทานการกระตุ้นเนื่องจากสนามไฟฟ้าใกล้เคียง อย่างไรก็ตาม เซ็นเซอร์แบบดิฟเฟอเรนเชียลคู่ไม่สามารถวัดอุณหภูมิได้ในรูปแบบนี้ ดังนั้นจึงมีประโยชน์เฉพาะสำหรับการตรวจจับการเคลื่อนไหวเท่านั้น

2.2.1.5 การปฏิบัติจริง

เมื่อเซ็นเซอร์ PIR ถูกกำหนดค่าในโหมดดิฟเฟอเรนเชียล เซ็นเซอร์จะสามารถใช้งานได้เฉพาะในฐานะอุปกรณ์ตรวจจับการเคลื่อนไหว ในโหมดนี้ เมื่อตรวจจับการเคลื่อนไหวภายใน “แนวสายตา” ของเซ็นเซอร์ พัลส์เสริมคู่หนึ่งจะถูกประมวลผลที่ขาเอาต์พุตของเซ็นเซอร์ เพื่อนำสัญญาณเอาต์พุตนี้ไปใช้งานจริงในการกระตุ้นโหลด เช่น รีเลย์หรือเครื่องบันทึกข้อมูลหรือสัญญาณเตือน สัญญาณดิฟเฟอเรน-เชียลจะถูกแก้ไขโดยใช้วงจรเรียงกระแสแบบบริดจ์และป้อนเข้าสู่วงจรขับรีเลย์แบบทรานซิสเตอร์ หน้าสัมผัสของรีเลย์นี้จะปิดและเปิดเพื่อตอบสนองต่อสัญญาณจาก PIR โดยกระตุ้นโหลดที่เชื่อมต่ออยู่ผ่านหน้าสัมผัสของมัน รับรู้ถึงการตรวจจับบุคคลภายในพื้นที่จำกัดที่กำหนดไว้ล่วงหน้า

2.2.1.6 การออกแบบผลิตภัณฑ์

รูปที่ 2.9 การออกแบบเซ็นเซอร์ตรวจจับการเคลื่อนไหว PIR

โดยทั่วไปเซ็นเซอร์ PIR จะติดตั้งอยู่บนแผงวงจรพิมพ์ซึ่งมีอุปกรณ์อิเล็กทรอนิกส์ที่จำเป็นสำหรับการตีความสัญญาณจากตัวเซ็นเซอร์เอง โดยทั่วไปแล้วชุดประกอบทั้งหมดจะบรรจุอยู่ภายในตัวเรือน ซึ่งติดตั้งในตำแหน่งที่เซ็นเซอร์สามารถครอบคลุมพื้นที่ที่ต้องการตรวจสอบได้ ตัวเรือนมักจะมี “หน้าต่าง” พลาสติกที่พลังงานอินฟราเรดสามารถผ่านเข้ามาได้ แม้ว่ามักจะโปร่งแสงต่อแสงที่มองเห็น แต่พลังงานอินฟราเรดสามารถผ่านเข้ามายังเซ็นเซอร์ได้ผ่านหน้าต่าง เนื่องจากพลาสติกที่ใช้นั้นโปร่งใสต่อรังสีอินฟราเรด หน้าต่างพลาสติกช่วยลดโอกาสที่วัตถุแปลกปลอม (ฝุ่น แมลง ฝน ฯลฯ) จะบดบังมุมมองของเซ็นเซอร์ ทำให้กลไกเสียหาย และอาจทำให้เกิดสัญญาณเตือนที่ผิดพลาด หน้าต่างนี้สามารถใช้เป็นตัวกรองเพื่อจำกัดความยาวคลื่นให้อยู่ที่ 8-14 ไมโครเมตร ซึ่งใกล้เคียงกับรังสีอินฟราเรดที่มนุษย์ปล่อยออกมามากที่สุด นอกจากนี้ยังสามารถใช้เป็นกลไกโฟกัสได้อีกด้วย (ดูด้านล่าง)

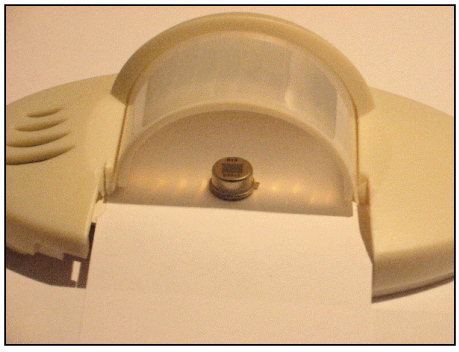
2.2.1.7 การโฟกัส

สามารถใช้กลไกที่แตกต่างกันเพื่อโฟกัสพลังงานอินฟราเรดระยะไกลลงบนพื้นผิวเซ็นเซอร์ได้

2.2.1.8 เลนส์

ม่านพลาสติกอาจหล่อขึ้นรูปหลายเหลี่ยมเพื่อรวมพลังงานอินฟราเรดไปยังเซ็นเซอร์ แต่ละเหลี่ยมคือเลนส์เฟรสเนล

1) เลนส์มัลติเฟรสเนลของ PIR



รูปที่ 2.10 ตัวเรือนเครื่องตรวจจับความเคลื่อนไหว PIR พร้อมช่องหน้าต่างทรงกระบอกเหลี่ยมโดยแต่ละเหลี่ยมเป็นเลนส์เฟรสเนล โฟกัสแสงไปที่ชิ้นส่วนเซ็นเซอร์ไพโรอิเล็กทริกที่อยู่ด้านล่าง

รูปที่ 2.11 ฝาครอบด้านหน้า PIR เท่านั้น (ถอดอุปกรณ์อิเล็กทรอนิกส์ออก) โดยมีแหล่งกำเนิดแสงจุดอยู่ด้านหลัง เพื่อแสดงเลนส์แต่ละตัว

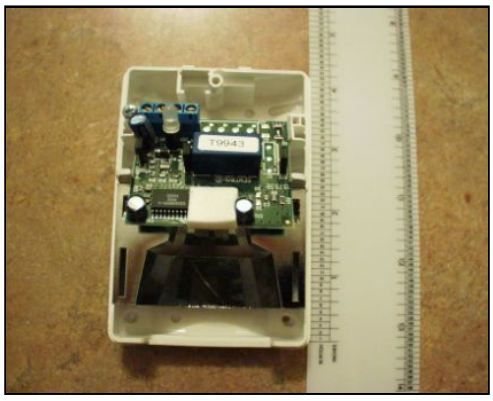
รูปที่ 2.12 PIR ที่ถอดฝาครอบด้านหน้าออก แสดงตำแหน่งของ เซ็นเซอร์ไพโรอิเล็กทริก (ลูกศรสีเขียว)

2.2.1.9 กระจก PIR

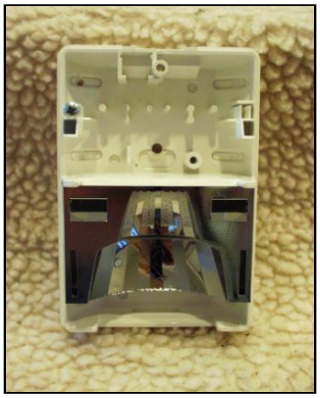
บางรุ่นผลิตขึ้นโดยใช้กระจกพาราโบลา แบบแบ่งส่วนภายใน เพื่อรวมพลังงานอินฟราเรด ในกรณีที่ใช้กระจก ฝาครอบกระจกพลาสติกโดยทั่วไปจะไม่มีเลนส์เฟรสเนลหล่อขึ้นรูป

1) PIR ชนิดกระจกแบ่งส่วน

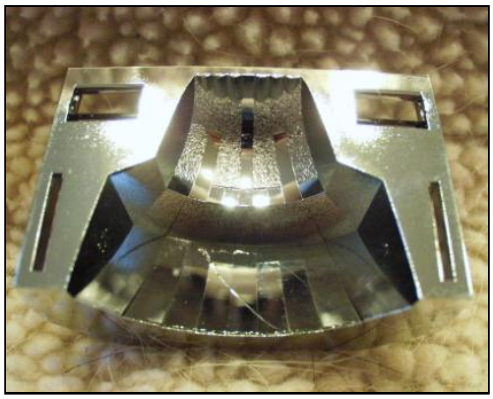
รูปที่ 2.13 PID ทั่วไปสำหรับที่พักอาศัย/เชิงพาณิชย์ที่ ใช้กระจกแบ่งส่วนภายในเพื่อการโฟกัส



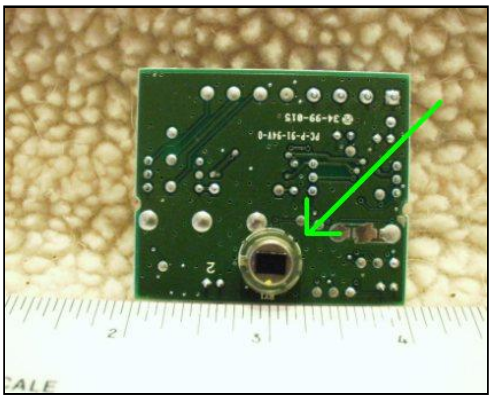
รูปที่ 2.14 ถอดฝาครอบออกแล้ว กระจกแบ่งส่วน ด้านล่างมีแผงวงจรพิมพ์ (PC) อยู่ด้านบน



รูปที่ 2.15 แผงวงจรพิมพ์ถูกถอดออกเพื่อแสดงกระจกแบบแบ่งส่วน

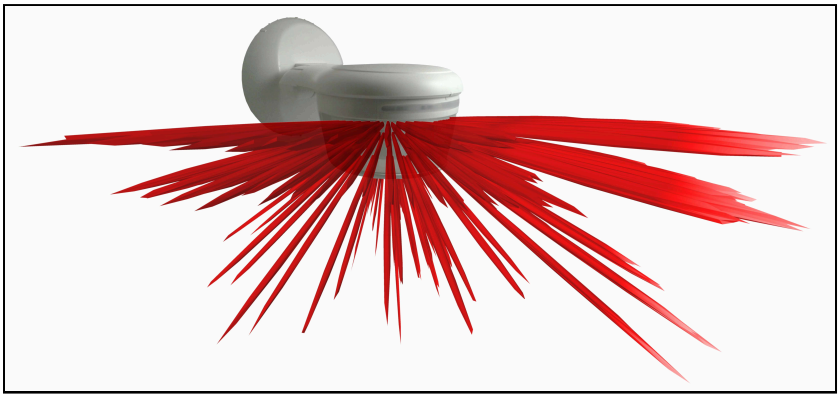


รูปที่ 2.16 กระจกพาราโบลาแบบแบ่งส่วนถอดออกจากตัวเครื่อง



รูปที่ 2.17 ด้านหลังของแผงวงจรที่หันเข้าหากระจกเมื่อติดตั้ง เซ็นเซอร์ไพโรอิเล็กทริกแสดงด้วยลูกศรสีเขียว

2.2.1.10 รูปแบบลำแสง



รูปที่ 2.18 เครื่องตรวจจับความเคลื่อนไหวที่มีรูปแบบลำแสงซ้อนทับ ความยาวของลำแสงเป็นตัวชี้วัดความไวของเครื่องตรวจจับในทิศทางนั้น

จากการโฟกัส ทำให้มุมมองของเครื่องตรวจจับกลายเป็นรูปแบบลำแสง ภายใต้มุมบางมุม (โซน) เซ็นเซอร์ PIR แทบจะไม่ได้รับพลังงานรังสีใด ๆ และภายใต้มุมอื่น ๆ PIR จะได้รับพลังงานอินฟราเรดในปริมาณที่เข้มข้น การแยกนี้ช่วยให้เครื่องตรวจจับความเคลื่อนไหวสามารถแยกแยะระหว่างแสงสว่างที่กว้างและวัตถุที่กำลังเคลื่อนที่ได้

เมื่อบุคคลเดินจากมุมหนึ่ง (ลำแสง) ไปยังอีกมุมหนึ่ง เครื่องตรวจจับจะมองเห็นบุคคลที่กำลังเคลื่อนไหวเป็นระยะ ๆ เท่านั้น ส่งผลให้สัญญาณเซ็นเซอร์เปลี่ยนแปลงอย่างรวดเร็ว ซึ่งระบบอิเล็กทรอนิกส์จะใช้เพื่อส่งสัญญาณเตือนภัยหรือเปิดไฟ ระบบอิเล็กทรอนิกส์จะไม่สนใจสัญญาณที่เปลี่ยนแปลงช้า ๆ

จำนวน รูปร่าง การกระจาย และความไวของโซนเหล่านี้ถูกกำหนดโดยเลนส์และกระจก ผู้ผลิตพยายามอย่างเต็มที่เพื่อสร้างรูปแบบลำแสงความไวที่เหมาะสมที่สุดสำหรับการใช้งานแต่ละประเภท

2.2.1.11 การใช้งานระบบไฟอัตโนมัติ

เมื่อใช้เป็นส่วนหนึ่งของระบบไฟส่องสว่าง ระบบอิเล็กทรอนิกส์ใน PIR มักจะควบคุมรีเลย์ในตัวที่สามารถสลับแรงดันไฟฟ้าหลักได้ ซึ่งหมายความว่า PIR สามารถตั้งค่าให้เปิดไฟที่เชื่อมต่อกับ PIR เมื่อตรวจพบการเคลื่อนไหวได้ วิธีนี้มักใช้ในสถานการณ์กลางแจ้ง ทั้งเพื่อป้องกันอาชญากร (ไฟรักษาความปลอดภัย) หรือเพื่อการใช้งานจริง เช่น การเปิดไฟประตูหน้าบ้านเพื่อให้คุณหากุญแจเจอในความมืด การใช้งานเพิ่มเติมสามารถทำได้ในห้องน้ำสาธารณะ ห้องเตรียมอาหารแบบวอล์กอิน ทางเดิน หรือบริเวณใดก็ตามที่สามารถควบคุมไฟอัตโนมัติได้ วิธีนี้ช่วยประหยัดพลังงานได้ เพราะไฟจะเปิดเฉพาะเมื่อจำเป็นเท่านั้น และผู้ใช้ไม่จำเป็นต้องปิดไฟเมื่อออกจากพื้นที่

2.2.1.12 แอปพลิเคชั่นด้านความปลอดภัย

เมื่อใช้เป็นส่วนหนึ่งของระบบรักษาความปลอดภัย วงจรอิเล็กทรอนิกส์ใน PIR มักจะควบคุมรีเลย์ ขนาดเล็ก รีเลย์นี้จะทำหน้าที่เชื่อมต่อวงจรไฟฟ้าผ่านหน้า สัมผัสไฟฟ้าคู่หนึ่งที่เชื่อมต่อกับโซนอินพุตตรวจจับของแผงควบคุมสัญญาณกันขโมยโดยทั่วไประบบจะออกแบบให้หากไม่มีการเคลื่อนไหว หน้าสัมผัสรีเลย์จะปิดอยู่ ซึ่งเรียกว่ารีเลย์แบบ ‘ปกติปิด’ (NC) หากตรวจพบการเคลื่อนไหว รีเลย์จะเปิดวงจรเพื่อส่งสัญญาณเตือนภัย หรือหากสายไฟถูกตัดการเชื่อมต่อ สัญญาณเตือนภัยก็จะทำงานเช่นกัน

2.2.1.13 การจัดวาง

ผู้ผลิตแนะนำให้วางผลิตภัณฑ์อย่างระมัดระวังเพื่อป้องกันการแจ้งเตือนที่ผิดพลาด (เช่น การตรวจจับใดๆ ที่ไม่ได้เกิดจากผู้บุกรุก)

พวกเขาแนะนำให้ติดตั้ง PIR ในลักษณะที่ PIR ไม่สามารถ “มองเห็น” ออกจากหน้าต่างได้ แม้ว่าความยาวคลื่นของรังสีอินฟราเรดที่ชิปมีความไวต่อแสงจะทะลุผ่านกระจกได้ไม่ดีนัก แต่แหล่งกำเนิดแสงอินฟราเรดที่แรง (เช่น จากไฟหน้ารถยนต์หรือแสงแดด) อาจทำให้เซ็นเซอร์รับภาพเกินพิกัดและทำให้เกิดสัญญาณเตือนภัยผิดพลาดได้ บุคคลที่เคลื่อนไหวอยู่อีกฝั่งของกระจกจะไม่ถูก PID “มองเห็น” ซึ่งอาจเป็นผลดีสำหรับหน้าต่างที่หันหน้าไปทางทางเท้าสาธารณะ หรือเป็นผลเสียสำหรับหน้าต่างในฉากกั้นภายใน

ขอแนะนำว่าไม่ควรติดตั้ง PIR ในตำแหน่งที่ ช่องระบายอากาศ HVAC จะเป่าลมร้อนหรือเย็นลงบนพื้นผิวพลาสติกที่ปิดหน้าต่างของตัวบ้าน แม้ว่าอากาศจะมีค่าการแผ่รังสี ต่ำมาก (ปล่อยพลังงานอินฟราเรดในปริมาณน้อยมาก) แต่ลมที่พัดผ่านฝาครอบหน้าต่างพลาสติกอาจทำให้อุณหภูมิของพลาสติกเปลี่ยนแปลงจนทำให้เกิดสัญญาณเตือนที่ผิดพลาดได้

เซ็นเซอร์มักได้รับการออกแบบมาให้ “เพิกเฉย” สัตว์เลี้ยงในบ้าน เช่น สุนัขหรือแมว โดยการตั้งค่าความไวให้สูงขึ้น หรือทำให้แน่ใจว่าพื้นห้องจะไม่อยู่ในโฟกัส

เนื่องจากเซ็นเซอร์ PIR มีระยะการทำงานสูงสุด 10 เมตร (30 ฟุต) ดังนั้นการติดตั้งเครื่องตรวจจับเพียงตัวเดียวใกล้ทางเข้าจึงเพียงพอสำหรับห้องที่มีทางเข้าเพียงทางเดียว ระบบรักษาความปลอดภัยที่ใช้ PIR ยังใช้งานได้ดีกับระบบรักษาความปลอดภัยภายนอกอาคารและระบบไฟที่ไวต่อการเคลื่อนไหว ข้อดีอย่างหนึ่งคือใช้พลังงานต่ำ ซึ่งทำให้สามารถใช้พลังงานแสงอาทิตย์ได้

2.2.1.14 เทอร์โมมิเตอร์แบบควบคุมระยะไกลด้วย PIR

มีการออกแบบวงจร PIR ที่ใช้วัดอุณหภูมิของวัตถุที่อยู่ห่างไกลในวงจรดังกล่าวจะใช้เอาต์พุต PIR แบบไม่มีค่าความแตกต่าง สัญญาณเอาต์พุตจะถูกประเมินตามการสอบเทียบสเปกตรัม IR ของสสารชนิดเฉพาะที่ต้องการตรวจวัด ด้วยวิธีนี้ การวัดอุณหภูมิจากระยะไกลจึงค่อนข้างแม่นยำและแม่นยำ หากไม่มีการสอบเทียบกับชนิดของวัสดุที่ตรวจวัด อุปกรณ์เทอร์โมมิเตอร์ PIR จะสามารถวัดการเปลี่ยนแปลงของการแผ่รังสี IR ซึ่งสอดคล้องกับการเปลี่ยนแปลงของอุณหภูมิโดยตรงแต่ไม่สามารถคำนวณค่าอุณหภูมิที่แท้จริงได้

2.3 ลำโพงสัญญาณ (Buzzer)

Buzzer เป็นอุปกรณ์ส่งสัญญาณเสียงซึ่งอาจเป็น อุปกรณ์ เชิงกลเครื่องกลไฟฟ้าหรือเพียโซอิเล็กทริก (เรียกสั้น ๆ ว่าเพียโซ) การใช้งานทั่วไปของบัซเซอร์และบี๊บเปอร์ ได้แก่อุปกรณ์แจ้งเตือนตัวตั้งเวลาวงจรและการยืนยันการป้อนข้อมูลของผู้ใช้ เช่น การคลิกเมาส์หรือการกดแป้นพิมพ์

ประเภทของ Buzzer มี 3 ชนิด คือ

2.3.1 ไฟฟ้าเชิงกล (Electromechanical)

อุปกรณ์ในยุคแรกๆ ใช้ระบบไฟฟ้าเครื่องกลแบบเดียวกับกระดิ่งไฟฟ้าโดยไม่มีฆ้องโลหะ ในทำนองเดียวกันรีเลย์อาจเชื่อมต่อเพื่อตัดกระแสไฟฟ้า ที่ทำหน้าที่สั่งการตัวเอง ซึ่งทำให้หน้าสัมผัสส่งเสียงหึ่งๆ (หน้าสัมผัสจะส่งเสียงหึ่งๆ ที่ความถี่สายหากใช้ไฟฟ้ากระแสสลับ) บ่อยครั้งที่อุปกรณ์เหล่านี้ถูกยึดไว้กับผนังหรือเพดานเพื่อใช้เป็นแผงเก็บเสียง คำว่า “buzzer” มาจากเสียงแหบๆ ของ buzzer ระบบไฟฟ้าเครื่องกล

2.3.2 กลไก (Mechanical)

จอยบัซเซอร์เป็นตัวอย่างของบัซเซอร์แบบกลไกล้วนๆ และจำเป็นต้องมีไดรเวอร์ ตัวอย่างอื่นๆ ของบัซเซอร์ประเภทนี้คือกริ่งประตู

2.3.3 เพียโซอิเล็กทริก (Piezoelectric)

องค์ประกอบเพียโซอิเล็กทริกอาจถูกขับเคลื่อนด้วย วงจรอิเล็กทรอนิกส์ แบบสั่นหรือ แหล่งสัญญาณเสียง อื่นๆ ซึ่งขับเคลื่อนด้วยเครื่องขยายเสียงเพียโซอิเล็กทริกเสียงที่มักใช้เพื่อระบุว่ามีการกดปุ่ม ได้แก่ เสียงคลิก เสียงกริ่ง หรือเสียงบี๊บ

2.4 เกณฑ์วิธีขนส่งข้อความหลายมิติ (HyperText Transfer Protocol; HTTP)

HTTP (Hypertext Transfer Protocol) เป็น โปรโตคอลชั้นแอปพลิเคชันในชุดโปรโตคอลอินเทอร์เน็ตสำหรับระบบข้อมูลไฮเปอร์มีเดียแบบกระจายและร่วมมือกัน HTTP เป็นรากฐานของการสื่อสารข้อมูลสำหรับ World Wide Web ซึ่งเอกสารไฮเปอร์เท็กซ์ รวมถึง ไฮเปอร์ลิงก์ไปยังทรัพยากรอื่น ๆ ที่ผู้ใช้สามารถเข้าถึงได้อย่างง่ายดาย เช่น โดยการคลิกเมาส์ หรือโดยการแตะหน้าจอในเว็บเบราว์เซอร์

HTTP เป็น โปรโตคอลแบบคำขอ-การตอบกลับในโมเดลไคลเอนต์-เซิร์ฟเวอร์ ธุรกรรมเริ่มต้นเมื่อไคลเอนต์ส่งคำขอไปยังเซิร์ฟเวอร์ เซิร์ฟเวอร์จะพยายามตอบสนองคำขอและส่งการตอบกลับกลับไปยังไคลเอนต์ ซึ่งอธิบายการจัดการคำขอ และอาจมีทรัพยากรที่ร้องขอ เช่น เอกสาร HTML หรือเนื้อหาอื่นๆ ก็ได้

ในสถานการณ์ทั่วไปเว็บเบราว์เซอร์ทำหน้าที่เป็นไคลเอนต์และเว็บเซิร์ฟเวอร์ที่โฮสต์เว็บไซต์หนึ่งเว็บไซต์หรือมากกว่านั้นคือ เซิร์ฟเวอร์ เว็บเบราว์เซอร์เป็นตัวอย่างของตัวแทนผู้ใช้ (UA) ตัวแทนผู้ใช้ประเภทอื่น ๆ ได้แก่ ซอฟต์แวร์จัดทำดัชนีที่ใช้โดยผู้ให้บริการค้นหา (เว็บครอว์เลอร์) เบราว์เซอร์เสียง แอปพลิเคชันมือถือ และซอฟต์แวร์อื่น ๆ ที่เข้าถึง ใช้ หรือแสดงเนื้อหาเว็บ

HTTP ถูกออกแบบมาเพื่ออนุญาตให้องค์ประกอบเครือข่ายตัวกลางสามารถปรับปรุงหรือเปิดใช้งานการสื่อสารระหว่างไคลเอนต์และเซิร์ฟเวอร์ เว็บไซต์ที่มีปริมาณการใช้งานสูงมักได้รับประโยชน์จากเซิร์ฟเวอร์ แคชเว็บที่ส่งเนื้อหาแทนเซิร์ฟเวอร์ต้นทางเพื่อปรับปรุงเวลาตอบสนอง เว็บเบราว์เซอร์จะแคชทรัพยากรเว็บที่เข้าถึงก่อนหน้านี้และนำกลับมาใช้ซ้ำทุกครั้งที่ทำได้เพื่อลดปริมาณการใช้งานเครือข่าย พร็อกซีเซิร์ฟเวอร์ HTTP ที่ ขอบเขต เครือข่ายส่วนตัวสามารถอำนวยความสะดวกในการสื่อสารสำหรับไคลเอนต์ที่ไม่มีที่อยู่ที่กำหนดเส้นทางได้ทั่วโลก โดยการส่งต่อข้อความไปยังเซิร์ฟเวอร์ภายนอก

เพื่ออนุญาตให้โหนด HTTP ตัวกลาง (พร็อกซีเซิร์ฟเวอร์ แคชเว็บ ฯลฯ) ทำหน้าที่ของตนได้ ส่วนหัว HTTP บางส่วน (พบในคำขอ/การตอบสนอง HTTP) จะได้รับการจัดการแบบฮอปต่อฮอปในขณะที่ส่วนหัว HTTP อื่นๆ จะได้รับการจัดการแบบต้นทางถึงปลายทาง (จัดการโดยไคลเอนต์ต้นทางและเว็บเซิร์ฟเวอร์เป้าหมายเท่านั้น)

ทรัพยากรบนเว็บจะถูกระบุตำแหน่งโดยตัวระบุทรัพยากรแบบสากล (URL) โดยใช้รูปแบบ Uniform Resource Identifier (URI) *http* และ *https* โดย URI จะถูกเข้ารหัสเป็นไฮเปอร์ลิงก์ในเอกสาร HTML เพื่อสร้างเอกสารไฮเปอร์เท็กซ์ที่เชื่อมโยงกัน

2.5 เกณฑ์วิธีขนส่งข้อความหลายมิติแบบมั่นคง (Hypertext Transfer Protocol Secure; HTTPS)

เกณฑ์วิธีขนส่งข้อความหลายมิติแบบมั่นคง (Hypertext Transfer Protocol Secure; HTTPS) คือส่วนต่อขยายของโปรโตคอลเกณฑ์วิธีขนส่งข้อความหลายมิติ (Hypertext Transfer Protocol; HTTP) ซึ่งใช้การเข้ารหัสเพื่อการสื่อสารที่ปลอดภัยผ่านเครือข่ายคอมพิวเตอร์ และถูกใช้อย่างแพร่หลายบนอินเทอร์เน็ต โดยโปรโตคอลเครือข่าย HTTPS จะถูกเข้ารหัสด้วยเกณฑ์วิธีความมั่นคงของชั้นขนส่ง (Transport Layer Security; TLS) หรือก่อนหน้านี้คือเกณฑ์วิธีชั้นซ็อกเก็ตปลอดภัย (Secure Sockets Layer; SSL) ด้วยเหตุนั้น โปรโตคอลนี้สามารถเรียกด้วยชื่อ HTTP over TLS หรือ HTTP over SSL ได้เช่นกัน

แรงจูงใจหลักของ HTTPS คือการยืนยันตัวตนของเว็บไซต์ที่เข้าถึง และการปกป้องความเป็นส่วนตัวและความสมบูรณ์ของข้อมูลที่แลกเปลี่ยนระหว่างการรับส่งข้อมูล HTTPS ป้องกันการโจมตีแบบ man-in-the-middle และการเข้ารหัสบล็อกไซเฟอร์แบบสองทิศทางในการสื่อสารระหว่างไคลเอนต์และเซิร์ฟเวอร์ ช่วยป้องกันการสื่อสารจากการดักฟังและการปลอมแปลง ประเด็นการพิสูจน์ตัวตนของ HTTPS จำเป็นต้องมีบุคคลที่สามที่เชื่อถือได้ลงนามในใบรับรองดิจิทัลฝั่งเซิร์ฟเวอร์ เดิมทีการดำเนินการนี้มีค่าใช้จ่ายสูง ซึ่งหมายความว่าการเชื่อมต่อ HTTPS ที่ผ่านการรับรองความถูกต้องอย่างสมบูรณ์มักจะพบได้เฉพาะในบริการธุรกรรมการชำระเงินที่ปลอดภัยและระบบสารสนเทศขององค์กรที่ปลอดภัยอื่นๆ บนเวิลด์ไวด์เว็บเท่านั้น ในปี 2016 แคมเปญโดยมูลนิธิพรมแดนอิเล็กทรอนิกส์ (Electronic Frontier Foundation; EFF) ด้วยการสนับสนุนจากนักพัฒนาเว็บเบราว์เซอร์ ทำให้โปรโตคอลนี้แพร่หลายมากขึ้น นับตั้งแต่ปี 2018 เป็นต้นมา HTTPS ถูกใช้โดยผู้ใช้เว็บบ่อยกว่า HTTP ดั้งเดิมที่ไม่ปลอดภัย โดย

ส่วนใหญ่เพื่อปกป้องความถูกต้องของหน้าเว็บบนเว็บไซต์ทุกประเภท รักษาความปลอดภัยบัญชี และรักษาความเป็นส่วนตัวของการสื่อสาร การระบุตัวตน และการท่องเว็บของผู้ใช้

2.5.1 โดยรวม

รูปแบบ Uniform Resource Identifier (URI) ของ HTTPS มีรูปแบบการใช้งานที่เหมือนกันกับรูปแบบ HTTP อย่างไรก็ตาม HTTPS จะส่งสัญญาณให้เบราว์เซอร์ใช้ชั้นการเข้ารหัสเพิ่มเติมของ SSL/TLS เพื่อปกป้องการรับส่งข้อมูลซึ่ง SSL/TLS เหมาะอย่างยิ่งสำหรับ HTTP เนื่องจากสามารถให้การป้องกันได้แม้ว่าจะมีการตรวจสอบความถูกต้องเพียงด้านเดียวของการสื่อสาร ในกรณีนี้คือธุรกรรม HTTP บนอินเทอร์เน็ต ซึ่งโดยทั่วไปมีเพียงเซิร์ฟเวอร์เท่านั้นที่ได้รับการรับรองความถูกต้อง (โดยไคลเอนต์ตรวจสอบใบรับรองของเซิร์ฟเวอร์)

HTTPS สร้างช่องทางที่ปลอดภัยบนเครือข่ายที่ไม่ปลอดภัย วิธีนี้ช่วยให้มั่นใจได้ถึงการป้องกันที่เหมาะสมจากผู้ดักฟังและการโจมตีแบบ man-in-the-middle โดยมีเงื่อนไขว่ามีการใช้ชุดการเข้ารหัสที่เหมาะสม และใบรับรองเซิร์ฟเวอร์ได้รับการตรวจสอบและเชื่อถือได้

เนื่องจาก HTTPS เชื่อมโยง HTTP ทั้งหมดเข้ากับ TLS โดยตรงจึงสามารถเข้ารหัสโปรโตคอล HTTP พื้นฐานทั้งหมดได้ ซึ่งรวมถึง URL ของคำขอ พารามิเตอร์การค้นหา ส่วนหัว และคุกกี้ (ซึ่งมักจะมีข้อมูลระบุตัวตนของผู้ใช้) อย่างไรก็ตาม เนื่องจากที่อยู่เว็บไซต์และหมายเลขพอร์ตเป็นส่วนหนึ่งของโปรโตคอล TCP/IP พื้นฐาน HTTPS จึงไม่สามารถปกป้องการเปิดเผยข้อมูลเหล่านี้ได้ ในทางปฏิบัติหมายความว่าแม้แต่บนเว็บเซิร์ฟเวอร์ที่กำหนดค่าอย่างถูกต้อง ผู้ดักฟังก็สามารถอนุมานที่อยู่ IP และหมายเลขพอร์ตของเว็บเซิร์ฟเวอร์ และบางครั้งอาจรวมถึงชื่อโดเมน (เช่น www.example.org แต่ไม่สามารถอนุมานส่วนที่เหลือของ URL) ที่ผู้ใช้กำลังสื่อสารด้วย รวมถึงปริมาณข้อมูลที่ถ่ายโอนและระยะเวลาของการสื่อสาร แต่อย่างไรก็ตามไม่รวมถึงเนื้อหาของการสื่อสาร

เว็บเบราว์เซอร์รู้วิธีเชื่อถือเว็บไซต์ HTTPS โดยอ้างอิงจากผู้ให้บริการออกใบรับรอง (Certificate Authority) ที่ติดตั้งไว้ล่วงหน้าในซอฟต์แวร์ ผู้สร้างเว็บเบราว์เซอร์จึงไว้วางใจผู้ให้บริการออกใบรับรองในการออกใบรับรองที่ถูกต้อง ดังนั้น ผู้ใช้ควรเชื่อถือการเชื่อมต่อ HTTPS ไปยังเว็บไซต์ก็ต่อเมื่อเป็นไปตามเงื่อนไขทั้งหมดต่อไปนี้:

1) ผู้ใช้เชื่อมั่นว่าอุปกรณ์ของตน โฮสต์เบราว์เซอร์ และวิธีการเข้าถึงเบราว์เซอร์นั้นไม่ถูกบุกรุก (กล่าวคือ ไม่มีการโจมตีซัพพลายเชน)

2) ผู้ใช้เชื่อมั่นว่าซอฟต์แวร์เบราว์เซอร์ใช้งาน HTTPS ได้อย่างถูกต้องพร้อมกับผู้ให้บริการออกใบรับรองที่ติดตั้งไว้ล่วงหน้าอย่างถูกต้อง

3) ผู้ใช้เชื่อมั่นว่าผู้ให้บริการออกใบรับรองจะรับรองเฉพาะเว็บไซต์ที่ถูกต้องตามกฎหมายเท่านั้น (กล่าวคือ ผู้ให้บริการออกใบรับรองจะไม่ถูกบุกรุกและไม่มีการออกใบรับรองที่ผิดพลาด)

4) เว็บไซต์มีใบรับรองที่ถูกต้อง ซึ่งหมายความว่าได้รับการลงนามโดยผู้ให้บริการที่เชื่อถือได้

5) ใบรับรองระบุเว็บไซต์ได้อย่างถูกต้อง (เช่น เมื่อเบราว์เซอร์เข้าชม https://example.com ใบรับรองที่ได้รับนั้นถูกต้องสำหรับ example.com และไม่ใช่ของหน่วยงานอื่น)

6) ผู้ใช้เชื่อมั่นว่าเลเยอร์การเข้ารหัสของโปรโตคอล (SSL/TLS) มีความปลอดภัยเพียงพอจากการดักฟัง

HTTPS มีความสำคัญอย่างยิ่งต่อเครือข่ายที่ไม่ปลอดภัยและเครือข่ายที่อาจถูกแทรกแซง เครือข่ายที่ไม่ปลอดภัย เช่น จุดเชื่อมต่อ Wi-Fi สาธารณะ ซึ่งเปิดโอกาสให้ทุกคนในเครือข่ายท้องถิ่นเดียวกันสามารถดักจับแพ็กเก็ตและค้นพบข้อมูลสำคัญที่ไม่ได้รับการป้องกันโดย HTTPS นอกจากนี้ ยังพบว่าเครือข่าย WLAN ทั้งแบบฟรีและแบบเสียเงินบางเครือข่ายได้แทรกแซงหน้าเว็บโดยการแทรกแพ็กเก็ตเพื่อแสดงโฆษณาของตนเองบนเว็บไซต์อื่น การกระทำเช่นนี้สามารถถูกนำไปใช้ในทางที่ผิดได้หลายวิธี เช่น การฉีดมัลแวร์ลงในหน้าเว็บและการขโมยข้อมูลส่วนบุคคลของผู้ใช้

เมื่อมีข้อมูลมากขึ้นเกี่ยวกับการเฝ้าระวังมวลชนทั่วโลกและการขโมยข้อมูลส่วนบุคคลของอาชญากร การใช้ระบบรักษาความปลอดภัย HTTPS บนเว็บไซต์ทั้งหมดจึงมีความสำคัญเพิ่มมากขึ้นเรื่อยๆ โดยไม่คำนึงถึงประเภทของการเชื่อมต่ออินเทอร์เน็ตที่ใช้งาน แม้ว่าข้อมูลเมตาเกี่ยวกับหน้าเว็บแต่ละหน้าที่ผู้ใช้เข้าชมอาจไม่ถือว่ามีความละเอียดอ่อน แต่เมื่อนำมารวมกันแล้ว ข้อมูลเมตาเหล่านี้อาจเปิดเผยข้อมูลเกี่ยวกับผู้ใช้ได้มาก และกระทบต่อความเป็นส่วนตัวของผู้ใช้

การปรับใช้ HTTPS ยังอนุญาตให้ใช้ HTTP/2 และ HTTP/3 (และรุ่นก่อนหน้าอย่าง SPDY และ QUIC) ซึ่งเป็น HTTP เวอร์ชันใหม่ที่ออกแบบมาเพื่อลดเวลา ขนาด และความหน่วงในการโหลดหน้าเว็บ และมีการแนะนำให้ใช้ HTTP Strict Transport Security (HSTS) ร่วมกับ HTTPS เพื่อป้องกันผู้ใช้จากการโจมตีแบบ man-in-the-middle โดยเฉพาะอย่างยิ่ง SSL stripping

2.5.2 ความปลอดภัย

ความปลอดภัยของ HTTPS อยู่ที่ TLS พื้นฐาน ซึ่งโดยทั่วไปจะใช้คีย์สาธารณะและคีย์ส่วนตัวระยะยาวเพื่อสร้างคีย์เซสชันระยะสั้น ซึ่งจะถูกนำไปใช้ในการเข้ารหัสการไหลของข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์ ใบรับรอง X.509 ถูกใช้เพื่อยืนยันตัวตนของเซิร์ฟเวอร์ (และบางครั้งรวมถึงไคลเอนต์ด้วย) ด้วยเหตุนี้ ผู้ให้บริการออกใบรับรองและใบรับรองคีย์สาธารณะจึงจำเป็นต่อการตรวจสอบความสัมพันธ์ระหว่างใบรับรองและเจ้าของ รวมถึงการสร้าง ลงนาม และดูแลความถูกต้องของใบรับรอง แม้ว่าวิธีนี้อาจมีประโยชน์มากกว่าการตรวจสอบตัวตนผ่านเครือข่ายที่เชื่อถือได้ แต่การเปิดเผยข้อมูลการเฝ้าระวังข้อมูลจำนวนมากในปี 2013 ได้ชี้ให้เห็นถึงผู้ให้บริการออกใบรับรองว่าเป็นจุดอ่อนที่อาจนำไปสู่การโจมตีแบบ man-in-the-middle คุณสมบัติที่สำคัญในบริบทนี้คือความลับแบบส่งต่อ (Forward Secrecy) ซึ่งรับประกันว่าการสื่อสารที่เข้ารหัสที่บันทึกไว้ในอดีตจะไม่สามารถดึงข้อมูลและถอดรหัสได้ หากคีย์ลับหรือรหัสผ่านระยะยาวถูกบุกรุกในอนาคต ไม่ใช่ทุกเว็บเซิร์ฟเวอร์ที่จะมีระบบความลับแบบส่งต่อ

เพื่อให้ HTTPS มีประสิทธิภาพ เว็บไซต์จะต้องโฮสต์ผ่าน HTTPS ทั้งหมด หากเนื้อหาบางส่วนของเว็บไซต์ถูกโหลดผ่าน HTTP (เช่น สคริปต์หรือรูปภาพ) หรือหากโหลดเฉพาะหน้าที่มีข้อมูลละเอียดอ่อน เช่น หน้าเข้าสู่ระบบ ผ่าน HTTPS ขณะที่ส่วนอื่น ๆ ของเว็บไซต์ผ่าน HTTP ธรรมดา ผู้ใช้จะเสี่ยงต่อการถูกโจมตีและการเฝ้าระวัง นอกจากนี้ คุกกี้บนเว็บไซต์ที่รันผ่าน HTTPS จะต้องเปิดใช้งานแอตทริบิวต์ secure ในเว็บไซต์ที่มีข้อมูลละเอียดอ่อน ผู้ใช้และเซสชันจะถูกเปิดเผยทุกครั้งที่เข้าถึงเว็บไซต์นั้นด้วย HTTP แทนที่จะเป็น HTTPS

2.5.3 รายละเอียดทางเทคนิค

2.5.3.1 ความแตกต่างจาก HTTP

URL แบบ HTTPS เริ่มต้นด้วย “https://” และใช้พอร์ต 443 ตามค่าเริ่มต้น ในขณะที่ URL แบบ HTTP เริ่มต้นด้วย “http://” และใช้พอร์ต 80 ตามค่าเริ่มต้น

HTTP ไม่ได้เข้ารหัส จึงมีความเสี่ยงต่อการโจมตีแบบ man-in-the-middle และการดักฟังซึ่งอาจทำให้ผู้โจมตีสามารถเข้าถึงบัญชีเว็บไซต์และข้อมูลสำคัญ และแก้ไขหน้าเว็บเพื่อแทรกมัลแวร์หรือโฆษณาได้ HTTPS ได้รับการออกแบบมาให้ทนทานต่อการโจมตีประเภทนี้ และถือว่าปลอดภัย (ยกเว้นการใช้งาน HTTPS ที่ใช้ SSL เวอร์ชันที่ล้าสมัย)

2.5.3.2 ชั้นเครือข่าย

HTTP ทำงานที่เลเยอร์สูงสุดของโมเดล TCP/IP นั่นคือเลเยอร์แอปพลิเคชัน เช่นเดียวกับโปรโตคอลความปลอดภัย TLS (ซึ่งทำงานเป็นเลเยอร์ย่อยที่ต่ำกว่าของเลเยอร์เดียวกัน) ซึ่งเข้ารหัสข้อความ HTTP ก่อนส่ง และถอดรหัสเมื่อข้อความมาถึง โดยเคร่งครัดแล้ว HTTPS ไม่ใช่โปรโตคอลใหม่ที่แยกจากกัน แต่หมายถึงการใช้ HTTP ทั่วไปบนการเชื่อมต่อ SSL/TLS ที่เข้ารหัส (เป็นส่วนต่อขยายจาก HTTP อย่างที่กล่าวไปข้างต้น)

HTTPS เข้ารหัสเนื้อหาข้อความทั้งหมด รวมถึงส่วนหัว HTTP และข้อมูลคำขอ/การตอบกลับ ยกเว้นการโจมตีด้วยการเข้ารหัส CCA ที่อาจเกิดขึ้นตามที่อธิบายไว้ในส่วนข้อจำกัดด้านล่าง ผู้โจมตีควรจะสามารถตรวจพบการเชื่อมต่อระหว่างสองฝ่ายได้มากที่สุด รวมถึงชื่อโดเมนและที่อยู่ IP ของฝ่ายนั้นด้วย

2.5.3.3 การตั้งค่าเซิร์ฟเวอร์

เพื่อเตรียมเว็บเซิร์ฟเวอร์ให้ยอมรับการเชื่อมต่อ HTTPS ผู้ดูแลระบบต้องสร้างใบรับรองคีย์สาธารณะสำหรับเว็บเซิร์ฟเวอร์ ใบรับรองนี้ต้องได้รับการลงนามโดยผู้ออกใบรับรองที่เชื่อถือได้เพื่อให้เว็บเบราว์เซอร์ยอมรับโดยไม่มีการแจ้งเตือน ผู้ออกใบรับรองจะรับรองว่าผู้ถือใบรับรองคือผู้ดำเนินการของเว็บเซิร์ฟเวอร์ที่นำเสนอใบรับรองนั้น โดยทั่วไปเว็บเบราว์เซอร์จะเผยแพร่รายชื่อใบรับรองการลงนามของผู้ออกใบรับรองหลักๆ เพื่อให้สามารถตรวจสอบใบรับรองที่ลงนามโดยผู้ออกใบรับรองเหล่านั้นได้

1) การขอใบรับรอง

มีผู้ให้บริการออกใบรับรองเชิงพาณิชย์จำนวนหนึ่งที่เสนอใบรับรอง SSL/TLS แบบชำระเงินหลายประเภท รวมถึงใบรับรองการตรวจสอบขยาย

Let’s Encrypt เปิดตัวในเดือนเมษายน 2559 ให้บริการใบรับรอง SSL/TLS พื้นฐานแบบอัตโนมัติฟรีแก่เว็บไซต์ มูลนิธิ Electronic Frontier Foundation ระบุว่า Let’s Encrypt จะทำให้การเปลี่ยนจาก HTTP เป็น HTTPS “ง่ายดายเพียงแค่ออกคำสั่งหรือคลิกปุ่ม” ปัจจุบันผู้ให้บริการเว็บโฮสต์และผู้ให้บริการคลาวด์ส่วนใหญ่ใช้ประโยชน์จาก Let’s Encrypt เพื่อมอบใบรับรองฟรีให้กับลูกค้า

2) ใช้เป็นการควบคุมการเข้าถึง

ระบบนี้ยังสามารถใช้สำหรับการตรวจสอบสิทธิ์ไคลเอ็นต์เพื่อจำกัดการเข้าถึงเว็บเซิร์ฟเวอร์เฉพาะผู้ใช้ที่ได้รับอนุญาตเท่านั้น ในการดำเนินการนี้ ผู้ดูแลระบบเว็บไซต์มักจะสร้างใบรับรองสำหรับผู้ใช้แต่ละราย ซึ่งผู้ใช้จะโหลดใบรับรองลงในเบราว์เซอร์ โดยปกติใบรับรองจะมีชื่อและที่อยู่อีเมลของผู้ใช้ที่ได้รับอนุญาต และจะถูกตรวจสอบโดยเซิร์ฟเวอร์โดยอัตโนมัติในแต่ละการเชื่อมต่อเพื่อยืนยันตัวตนของผู้ใช้ ซึ่งอาจไม่จำเป็นต้องใช้รหัสผ่านด้วยซ้ำ

3) ในกรณีที่คีย์ลับถูกบุกรุก

คุณสมบัติที่สำคัญในบริบทนี้คือการเข้ารหัสแบบส่งต่อที่สมบูรณ์แบบ (PFS) การมีคีย์ลับแบบอสมมาตรระยะยาวตัวใดตัวหนึ่งที่ใช้สร้างเซสชัน HTTPS ไม่น่าจะทำให้การได้มาซึ่งคีย์เซสชันระยะสั้นเพื่อถอดรหัสการสนทนาทำได้ง่ายขึ้น แม้ในภายหลังก็ตาม ในปี 2013 มีเพียงการแลกเปลี่ยนคีย์ Diffie–Hellman (DHE) และการแลกเปลี่ยนคีย์ Diffie–Hellman แบบเส้นโค้งวงรี (ECDHE) เท่านั้นที่ทราบว่ามีคุณสมบัตินี้ ในปี 2013 มีเพียง 30% ของเซสชัน Firefox, Opera และ Chromium Browser เท่านั้นที่ใช้คุณสมบัตินี้ และเกือบ 0% ของเซสชัน Safari และ Microsoft Internet Explorer ของ Apple ที่ใช้คุณสมบัตินี้ TLS 1.3 ซึ่งเผยแพร่ในเดือนสิงหาคม 2018 ได้ยกเลิกการสนับสนุนการเข้ารหัสแบบไม่มีการเข้ารหัสแบบส่งต่อ ณ เดือนกุมภาพันธ์ พ.ศ. 2562 เว็บเซิร์ฟเวอร์ที่สำรวจ 96.6% รองรับการรักษาความลับแบบ Forward ในรูปแบบใดรูปแบบหนึ่ง และ 52.1% จะใช้การรักษาความลับแบบ Forward กับเบราว์เซอร์ส่วนใหญ่ ณ เดือนกรกฎาคม พ.ศ. 2566 เว็บเซิร์ฟเวอร์ที่สำรวจ 99.6% รองรับการรักษาความลับแบบ Forward ในรูปแบบใดรูปแบบหนึ่ง และ 75.2% จะใช้การรักษาความลับแบบ Forward กับเบราว์เซอร์ส่วนใหญ่

4) การเพิกถอนใบรับรอง

ใบรับรองอาจถูกเพิกถอนก่อนหมดอายุได้ เช่น เนื่องจากความลับของคีย์ส่วนตัวถูกละเมิด เบราว์เซอร์ยอดนิยมเวอร์ชันที่ใหม่พอเช่น Firefox Opera และ Internet Explorer บน Windows Vista จะใช้ Online Certificate Status Protocol (OCSP) เพื่อตรวจสอบว่าไม่เป็นเช่นนั้น เบราว์เซอร์จะส่งหมายเลขซีเรียลของใบรับรองไปยังผู้ออกใบรับรองหรือผู้แทนผ่าน OCSP และผู้ออกใบรับรองจะตอบกลับ โดยแจ้งให้เบราว์เซอร์ทราบว่าใบรับรองยังคงใช้ได้อยู่หรือไม่ นอกจากนี้ CA อาจออกรายการเพิกถอนใบรับรอง (Certificate Revocation List; CRL) เพื่อแจ้งให้ผู้ใช้ทราบว่าใบรับรองเหล่านี้ถูกเพิกถอนแล้ว อย่างไรก็ตาม CRL ไม่จำเป็นสำหรับฟอรัม CA/Browser (“ฟอรัม CA/Browser” ดังกล่าวคือองค์กร) อีกต่อไป อย่างไรก็ตาม CA ยังคงใช้ CRL กันอย่างแพร่หลาย สถานะการเพิกถอนส่วนใหญ่บนอินเทอร์เน็ตจะหายไปในไม่ช้าหลังจากใบรับรองหมดอายุ

2.5.3.4 ข้อจำกัด

การเข้ารหัส SSL (Secure Sockets Layer) และ TLS (Transport Layer Security) สามารถกำหนดค่าได้สองโหมด ได้แก่ โหมดธรรมดาและโหมด Mutual ในโหมดธรรมดา การตรวจสอบสิทธิ์จะดำเนินการโดยเซิร์ฟเวอร์เท่านั้น โหมด Mutual กำหนดให้ผู้ใช้ต้องติดตั้งใบรับรองไคลเอ็นต์ส่วนบุคคลในเว็บเบราว์เซอร์เพื่อการตรวจสอบสิทธิ์ผู้ใช้ ไม่ว่าในกรณีใด ระดับการป้องกันจะขึ้นอยู่กับความถูกต้องของการใช้งานซอฟต์แวร์และอัลกอริทึมการเข้ารหัสที่ใช้

SSL/TLS ไม่ป้องกันการจัดทำดัชนีของเว็บไซต์โดยเว็บครอว์เลอร์ และในบางกรณี URI ของทรัพยากรที่เข้ารหัสสามารถอนุมานได้โดยการทราบขนาดคำขอ/การตอบสนองที่ถูกสกัดกั้นเท่านั้น วิธีนี้ช่วยให้ผู้โจมตีสามารถเข้าถึงข้อความธรรมดา (เนื้อหาคงที่ที่เปิดเผยต่อสาธารณะ) และข้อความที่เข้ารหัส (เนื้อหาคงที่เวอร์ชันเข้ารหัส) ทำให้สามารถโจมตีด้วยการเข้ารหัสได้

เนื่องจาก TLS ทำงานที่ระดับโปรโตคอลที่ต่ำกว่า HTTP และไม่มีความรู้เกี่ยวกับโปรโตคอลระดับสูงกว่า เซิร์ฟเวอร์ TLS จึงสามารถแสดงใบรับรองได้เพียงใบเดียวสำหรับที่อยู่และพอร์ตที่กำหนดเท่านั้น ในอดีต นั่นหมายความว่าไม่สามารถใช้การโฮสต์เสมือนแบบอิงชื่อกับ HTTPS ได้ มีโซลูชันที่เรียกว่า Server Name Indication (SNI) ซึ่งส่งชื่อโฮสต์ไปยังเซิร์ฟเวอร์ก่อนเข้ารหัสการเชื่อมต่อ แม้ว่าเบราว์เซอร์รุ่นเก่าจะไม่รองรับส่วนขยายนี้ก็ตาม การรองรับ SNI มีให้ใช้งานตั้งแต่ Firefox 2, Opera 8, Apple Safari 2.1, Google Chrome 6 และ Internet Explorer 7 บน Windows Vista

การโจมตีแบบ man-in-the-middle ที่ซับซ้อนประเภทหนึ่งที่เรียกว่า SSL stripping ถูกนำเสนอในงานประชุม Blackhat Conference ปี 2009 การโจมตีประเภทนี้ทำลายความปลอดภัยของ HTTPS โดยการเปลี่ยนลิงก์ https: ให้เป็นลิงก์ http: โดยใช้ประโยชน์จากข้อเท็จจริงที่ว่าผู้ใช้อินเทอร์เน็ตเพียงไม่กี่คนเท่านั้นที่พิมพ์ “https” ลงในอินเทอร์เฟซเบราว์เซอร์ พวกเขาจึงเข้าสู่เว็บไซต์ที่ปลอดภัยได้โดยการคลิกลิงก์ และถูกหลอกให้คิดว่ากำลังใช้ HTTPS ในขณะที่จริงๆ แล้วกำลังใช้ HTTP ผู้โจมตีจึงสื่อสารกับไคลเอ็นต์อย่างชัดเจน สิ่งนี้กระตุ้นให้เกิดการพัฒนามาตรการรับมือใน HTTP ที่เรียกว่า HTTP Strict Transport Security

HTTPS ได้รับการพิสูจน์แล้วว่ามีความเสี่ยงต่อการโจมตีวิเคราะห์ทราฟฟิกหลากหลายรูปแบบ การโจมตีวิเคราะห์ทราฟฟิกเป็นการโจมตีแบบ Side-Channel ประเภทหนึ่งที่อาศัยการเปลี่ยนแปลงเวลาและขนาดของทราฟฟิกเพื่ออนุมานคุณสมบัติของทราฟฟิกที่เข้ารหัส การวิเคราะห์ทราฟฟิกเป็นไปได้เนื่องจากการเข้ารหัส SSL/TLS เปลี่ยนแปลงเนื้อหาของทราฟฟิก แต่มีผลกระทบน้อยมากต่อขนาดและระยะเวลาของทราฟฟิก ในเดือนพฤษภาคม 2553 งานวิจัยโดยนักวิจัยจาก Microsoft Research และ Indiana University ค้นพบว่าข้อมูลผู้ใช้ที่ละเอียดอ่อนโดยละเอียดสามารถอนุมานได้จากช่องทางด้านข้าง เช่น ขนาดแพ็กเก็ต นักวิจัยพบว่าแม้จะมีการป้องกัน HTTPS ในแอปพลิเคชันเว็บชั้นนำที่มีชื่อเสียงหลายตัวในด้านการดูแลสุขภาพ ภาษี การลงทุน และการค้นหาเว็บ แต่ผู้แอบฟังสามารถอนุมานโรค/ยา/การผ่าตัดของผู้ใช้ รายได้ของครอบครัว และความลับในการลงทุนได้

ความจริงที่ว่าเว็บไซต์สมัยใหม่ส่วนใหญ่ รวมถึง Google, Yahoo! และ Amazon ใช้ HTTPS ทำให้เกิดปัญหาสำหรับผู้ใช้จำนวนมากที่พยายามเข้าถึงจุดเชื่อมต่อ Wi-Fi สาธารณะ เนื่องจากหน้าเข้าสู่ระบบจุดเชื่อมต่อ Wi-Fi ของพอร์ทัลแบบแคปทีฟไม่สามารถโหลดได้หากผู้ใช้พยายามเปิดทรัพยากร HTTPS และเว็บไซต์หลายแห่ง เช่น NeverSSL รับประกันว่าเว็บไซต์เหล่านั้นจะสามารถเข้าถึงได้ผ่าน HTTP เสมอ

2.6 เกณฑ์วิธีความมั่นคงของชั้นขนส่ง (Transport Layer Security; TLS)

เกณฑ์วิธีความมั่นคงของชั้นขนส่ง (Transport Layer Security; TLS) เป็นโปรโตคอลการเข้ารหัสที่ออกแบบมาเพื่อรักษาความปลอดภัยการสื่อสารบนเครือข่ายคอมพิวเตอร์เช่นอินเทอร์เน็ตโปรโตคอลนี้ถูกใช้อย่างแพร่หลายในแอปพลิเคชันต่างๆเช่นอีเมลการส่งข้อความโต้ตอบแบบทันทีและบริการเสียงผ่าน IP แต่การใช้งานเพื่อรักษาความปลอดภัย HTTPS ยังคงเป็นที่เปิดเผยต่อสาธารณะมากที่สุด

โปรโตคอล TLS มีวัตถุประสงค์หลักเพื่อรักษาความปลอดภัย รวมถึงความเป็นส่วนตัว (ความลับ) ความสมบูรณ์ และความถูกต้อง ผ่านการใช้การเข้ารหัสเช่น การใช้ใบรับรองระหว่างแอปพลิเคชันคอมพิวเตอร์ที่สื่อสารกันตั้งแต่สองแอปพลิเคชันขึ้นไป โปรโตคอลนี้ทำงานในเลเยอร์การนำเสนอและประกอบด้วยสองชั้น ได้แก่ ระเบียน TLS และโปรโตคอล TLS handshake

Datagram Transport Layer Security (DTLS) ซึ่งเป็นโปรโตคอลการสื่อสารที่เกี่ยวข้องอย่างใกล้ชิดมอบความปลอดภัยให้กับ แอปพลิเคชันที่ใช้ ดาต้าแกรมในงานเขียนทางเทคนิค มักพบการอ้างอิงถึง “(D)TLS” เมื่อใช้กับทั้งสองเวอร์ชัน

TLS เป็นมาตรฐานที่ได้รับการเสนอโดย Internet Engineering Task Force (IETF) ซึ่งกำหนดขึ้นครั้งแรกในปี 1999 และเวอร์ชันปัจจุบันคือ TLS 1.3 ซึ่งกำหนดขึ้นในเดือนสิงหาคม 2018 TLS สร้างขึ้นจาก ข้อกำหนด SSL (Secure Sockets Layer) ที่ไม่รองรับอีกต่อไป (1994, 1995, 1996) ซึ่งพัฒนาโดย Netscape Communications เพื่อเพิ่มโปรโตคอล HTTPS ลงในเว็บเบราว์เซอร์ Netscape Navigator

2.6.1 คำอธิบาย

เนื่องจากแอปพลิเคชันสามารถสื่อสารได้ทั้งแบบมีหรือไม่มี TLS (หรือ SSL) จึงจำเป็นที่ไคลเอนต์จะต้องร้องขอให้เซิร์ฟเวอร์ตั้งค่าการเชื่อมต่อ TLS หนึ่งในวิธีหลักในการทำเช่นนี้คือการใช้หมายเลขพอร์ต อื่น สำหรับการเชื่อมต่อ TLS โดยทั่วไปแล้ว พอร์ต 80 จะใช้สำหรับ การรับส่งข้อมูล HTTP ที่ไม่ได้เข้ารหัส ในขณะที่พอร์ต 443 เป็นพอร์ตทั่วไปที่ใช้สำหรับ การรับส่งข้อมูล HTTPS ที่เข้ารหัส อีกกลไกหนึ่งคือการสร้างคำขอ STARTTLS เฉพาะโปรโตคอลไปยังเซิร์ฟเวอร์เพื่อสลับการเชื่อมต่อกับ TLS ตัวอย่างเช่น เมื่อใช้โปรโตคอลอีเมลและข่าวสารบางอย่าง

เมื่อไคลเอนต์และเซิร์ฟเวอร์ตกลงที่จะใช้ TLS แล้ว พวกเขาจะเจรจา การเชื่อมต่อ แบบมีสถานะโดยใช้ขั้นตอนการจับมือ (ดูการจับมือ TLS) โปรโตคอลใช้การจับมือกับรหัสแบบอสมมาตรเพื่อกำหนดค่าการเข้ารหัสไม่เพียงเท่านั้น แต่ยังรวมถึงคีย์ที่ใช้ร่วมกันเฉพาะเซสชัน ซึ่งการสื่อสารต่อไปจะถูกเข้ารหัสโดยใช้รหัสแบบสมมาตรในระหว่างการจับมือนี้ ไคลเอนต์และเซิร์ฟเวอร์จะตกลงกันเกี่ยวกับพารามิเตอร์ต่างๆ ที่ใช้สร้างความปลอดภัยของการเชื่อมต่อ

1) การจับมือเริ่มต้นเมื่อไคลเอนต์เชื่อมต่อกับเซิร์ฟเวอร์ที่เปิดใช้งาน TLS เพื่อขอการเชื่อมต่อที่ปลอดภัยและไคลเอนต์แสดงรายการชุดรหัสที่รองรับ (รหัสและฟังก์ชันแฮช)

2) จากรายการนี้ เซิร์ฟเวอร์จะเลือกฟังก์ชันรหัสและแฮชที่รองรับ และแจ้งให้ไคลเอนต์ทราบถึงการตัดสินใจ

3) โดยปกติแล้วเซิร์ฟเวอร์จะระบุตัวตนในรูปแบบของใบรับรองดิจิทัลใบรับรองประกอบด้วยชื่อเซิร์ฟเวอร์ผู้ให้บริการออกใบรับรอง (CA) ที่เชื่อถือได้ซึ่งรับรองความถูกต้องของใบรับรอง และคีย์การเข้ารหัสสาธารณะของเซิร์ฟเวอร์

4) ลูกค้าต้องยืนยันความถูกต้องของใบรับรองก่อนดำเนินการต่อ

5) ในการสร้างคีย์เซสชันที่ใช้สำหรับการเชื่อมต่อที่ปลอดภัย ไคลเอนต์จะต้องทำดังนี้

ก) เข้ารหัสตัวเลขสุ่ม (PreMasterSecret) ด้วยคีย์สาธารณะของเซิร์ฟเวอร์และส่งผลลัพธ์ไปยังเซิร์ฟเวอร์ (ซึ่งเฉพาะเซิร์ฟเวอร์เท่านั้นที่จะสามารถถอดรหัสด้วยคีย์ส่วนตัว) จากนั้นทั้งสองฝ่ายใช้ตัวเลขสุ่มเพื่อสร้างคีย์เซสชันเฉพาะสำหรับการเข้ารหัสและถอดรหัสข้อมูลในระหว่างเซสชันในภายหลังหรือ

ข) ใช้การแลกเปลี่ยนคีย์ Diffie–Hellman (หรือรูปแบบ DH ที่เป็นเส้นโค้งวงรี) เพื่อสร้างคีย์เซสชันแบบสุ่มและไม่ซ้ำกันอย่างปลอดภัยสำหรับการเข้ารหัสและถอดรหัส ซึ่งมีคุณสมบัติเพิ่มเติมของการปกปิดแบบส่งต่อ โดยหากคีย์ส่วนตัวของเซิร์ฟเวอร์ถูกเปิดเผยในอนาคต จะไม่สามารถใช้คีย์นั้นเพื่อถอดรหัสเซสชันปัจจุบันได้ แม้ว่าเซสชันนั้นจะถูกดักจับและบันทึกโดยบุคคลที่สามก็ตาม

การดำเนินการนี้จะสิ้นสุดการจับมือและเริ่มการเชื่อมต่อที่ปลอดภัยซึ่งจะถูกเข้ารหัสและถอดรหัสด้วยคีย์เซสชันจนกว่าการเชื่อมต่อจะสิ้นสุดลงหากขั้นตอนใดขั้นตอนหนึ่งข้างต้นล้มเหลวการจับมือ TLS จะล้มเหลวและการเชื่อมต่อจะไม่ถูกสร้างขึ้น

TLS และ SSL ไม่สามารถจัดวางได้อย่างลงตัวในเลเยอร์ใดเลเยอร์หนึ่งของแบบจำลอง OSI หรือแบบจำลอง TCP/IP TLS ทำงาน “บนโปรโตคอลการขนส่งที่เชื่อถือได้ (เช่น TCP)” ซึ่งหมายความว่ามันอยู่เหนือเลเยอร์การขนส่งมันทำหน้าที่เข้ารหัสให้กับเลเยอร์ที่สูงกว่า ซึ่งโดยปกติแล้วเป็นหน้าที่ของเลเยอร์การนำเสนออย่างไรก็ตาม โดยทั่วไปแอปพลิเคชันจะใช้ TLS เหมือนกับเป็นเลเยอร์การขนส่งแม้ว่าแอปพลิเคชันที่ใช้ TLS จะต้องควบคุมการเริ่มต้นการจับมือ TLS และการจัดการใบรับรองการตรวจสอบสิทธิ์ที่แลกเปลี่ยนกัน

เมื่อได้รับการรักษาความปลอดภัยโดย TLS การเชื่อมต่อระหว่างไคลเอนต์ (เช่น เว็บเบราว์เซอร์) และเซิร์ฟเวอร์ (เช่น wikipedia.org) จะมีคุณสมบัติทั้งหมดดังต่อไปนี้

1) การเชื่อมต่อเป็นแบบส่วนตัว (หรือมีความลับ) เนื่องจาก มีการใช้ อัลกอริทึมคีย์แบบสมมาตรในการเข้ารหัสข้อมูลที่ส่ง คีย์สำหรับการเข้ารหัสแบบสมมาตรนี้จะถูกสร้างขึ้นอย่างเฉพาะเจาะจงสำหรับแต่ละการเชื่อมต่อ และอิงจากความลับร่วมที่เจรจากันไว้เมื่อเริ่มต้นเซสชัน เซิร์ฟเวอร์และไคลเอ็นต์จะเจรจารายละเอียดเกี่ยวกับอัลกอริทึมการเข้ารหัสและคีย์การเข้ารหัสที่จะใช้ก่อนที่จะส่งข้อมูลไบต์แรก (ดูด้านล่าง) การเจรจาความลับร่วมนั้นทั้งปลอดภัย (ความลับที่เจรจากันไว้จะไม่สามารถเข้าถึงได้โดยผู้ดักฟังและไม่สามารถได้รับ แม้แต่โดยผู้โจมตีที่วางตัวเองอยู่ตรงกลางการเชื่อมต่อ) และเชื่อถือได้ (ไม่มีผู้โจมตีคนใดสามารถแก้ไขการสื่อสารระหว่างการเจรจาโดยไม่ถูกตรวจพบ)

2) การยืนยันตัวตนของฝ่ายที่สื่อสารสามารถยืนยันได้โดยใช้การเข้ารหัสด้วยคีย์สาธารณะ การยืนยันตัวตนนี้จำเป็นสำหรับเซิร์ฟเวอร์และเป็นทางเลือกสำหรับไคลเอนต์

3) การเชื่อมต่อมีความน่าเชื่อถือ (หรือมีความสมบูรณ์) เนื่องจากข้อความแต่ละข้อความที่ส่งออกจะมีการตรวจสอบความสมบูรณ์ของข้อความโดยใช้รหัสยืนยันข้อความเพื่อป้องกันการสูญหายหรือการเปลี่ยนแปลงข้อมูลที่ไม่ถูกตรวจพบระหว่างการส่งข้อมูล

TLS รองรับวิธีการที่หลากหลายสำหรับการแลกเปลี่ยนคีย์ การเข้ารหัสข้อมูล และการตรวจสอบความถูกต้องของข้อความ ดังนั้น การกำหนดค่า TLS อย่างปลอดภัยจึงเกี่ยวข้องกับพารามิเตอร์ที่กำหนดค่าได้มากมาย และตัวเลือกทั้งหมดไม่ได้มีคุณสมบัติที่เกี่ยวข้องกับความเป็นส่วนตัวทั้งหมดที่อธิบายไว้ในรายการด้านบน (ดูตารางด้านล่าง การแลกเปลี่ยนคีย์ ความปลอดภัยของการเข้ารหัสและความสมบูรณ์ของข้อมูล)

มีการพยายามบ่อนทำลายแง่มุมด้านความปลอดภัยในการสื่อสารที่ TLS มุ่งหวังจะมอบให้ และโปรโตคอลนี้ได้รับการแก้ไขหลายครั้งเพื่อจัดการกับภัยคุกคามด้านความปลอดภัยเหล่านี้ นักพัฒนาเว็บเบราว์เซอร์ได้ปรับปรุงผลิตภัณฑ์ของตนซ้ำแล้วซ้ำเล่าเพื่อป้องกันจุดอ่อนด้านความปลอดภัยที่อาจ

เกิดขึ้นหลังจากค้นพบจุดอ่อนเหล่านี้ (ดูประวัติการสนับสนุน TLS/SSL ของเว็บเบราว์เซอร์) ความปลอดภัยของเลเยอร์การขนส่งดาต้าแกรม

Datagram Transport Layer Security หรือเรียกย่อๆ ว่า DTLS เป็นโปรโตคอลการสื่อสารที่เกี่ยวข้องซึ่งให้ความปลอดภัยแก่ แอปพลิเคชันที่ใช้ Datagram โดยอนุญาตให้แอปพลิเคชันสื่อสารในลักษณะที่ออกแบบมาเพื่อป้องกันการดักฟัง การปลอมแปลงหรือการปลอมแปลงข้อความโปรโตคอล DTLS ใช้ โปรโตคอล Transport Layer Security (TLS) ที่เน้น การสตรีมและมีจุดประสงค์เพื่อให้การรับประกันความปลอดภัยที่คล้ายคลึงกัน อย่างไรก็ตาม โปรโตคอลนี้แตกต่างจาก TLS ตรงที่สามารถใช้งานร่วมกับโปรโตคอลที่เน้น Datagram ส่วนใหญ่ ได้แก่ User Datagram Protocol (UDP), Datagram Congestion Control Protocol (DCCP), Control And Provisioning of Wireless Access Points (CAPWAP), Stream Control Transmission Protocol (SCTP) encapsulation และ Secure Real-time Transport Protocol (SRTP)

เนื่องจากเดตาแกรมของโปรโตคอล DTLS รักษาความหมายของการขนส่งพื้นฐานไว้ แอปพลิเคชันจึงไม่ประสบปัญหาความล่าช้าที่เกี่ยวข้องกับโปรโตคอลสตรีม อย่างไรก็ตาม แอปพลิเคชันต้องจัดการกับการเรียงลำดับแพ็กเก็ตใหม่ การสูญหายของเดตาแกรม และข้อมูลที่มีขนาดใหญ่กว่าขนาดของแพ็กเก็ตเครือข่าย เดตาแกรม เนื่องจาก DTLS ใช้ UDP หรือ SCTP แทน TCP จึงหลีกเลี่ยงปัญหา TCP ล่มเมื่อนำไปใช้สร้างอุโมงค์ VPN

DTLS เวอร์ชัน 1.0 ฉบับดั้งเดิมในปี 2006 ไม่ใช่เอกสารแบบสแตนด์อโลน แต่ได้รับการกำหนดให้เป็นชุดเดลต้าของ TLS 1.1 ทำนองเดียวกัน DTLS เวอร์ชัน 2012 ที่ตามมาก็ถูกกำหนดให้เป็นเดลต้าของ TLS 1.2 โดยได้รับหมายเลขเวอร์ชันของ DTLS 1.2 เพื่อให้ตรงกับเวอร์ชัน TLS สุดท้าย DTLS 1.3 ปี 2022 ก็ถูกกำหนดให้เป็นเดลต้าของ TLS 1.3 เช่นเดียวกับสองเวอร์ชันก่อนหน้า DTLS 1.3 มีวัตถุประสงค์เพื่อให้ “การรับประกันความปลอดภัยที่เทียบเท่า [กับ TLS 1.3] ยกเว้นการป้องกันคำสั่ง/การไม่สามารถเล่นซ้ำได้”

ไคลเอนต์ VPN จำนวนมากรวมถึง Cisco AnyConnect & InterCloud Fabric, OpenConnect, อุโมงค์ ZScaler, F5 Networks Edge VPN Client และ Citrix Systems NetScaler ใช้ DTLS เพื่อรักษาความปลอดภัยการรับส่งข้อมูล UDP นอกจากนี้ เว็บเบราว์เซอร์สมัยใหม่ทั้งหมดยังรองรับ DTLS-SRTP สำหรับ WebRTC

2.6.2 X.509 (รูปแบบใบรับรอง TLS/SSL)

ในการเข้ารหัส X.509 เป็นมาตรฐานของสหภาพโทรคมนาคมระหว่างประเทศ (ITU) ที่กำหนดรูปแบบของใบรับรองคีย์สาธารณะใบรับรอง X.509 ถูกใช้ในโปรโตคอลอินเทอร์เน็ตมากมายรวมถึง TLS/SSL ซึ่งเป็นพื้นฐานของ HTTPS โปรโตคอลที่ปลอดภัยสำหรับการท่องเว็บ นอกจากนี้ยังใช้ในแอปพลิเคชันออฟไลน์เช่น ลาย เซ็นอิเล็กทรอนิกส์ ใบรับรอง X.509 เชื่อมโยงข้อมูลประจำตัวกับคีย์สาธารณะโดยใช้ลายเซ็นดิจิทัล ใบรับรองประกอบด้วยข้อมูลประจำตัว (ชื่อโฮสต์องค์กร หรือบุคคล) และคีย์สาธารณะ (RSA, DSA, ECDSA, ed25519 เป็นต้น) ซึ่งลงนามโดยผู้ออกใบรับรองหรือลงนามด้วยตนเอง เมื่อใบรับรองได้รับการลงนามโดยผู้ออกใบรับรองที่เชื่อถือได้หรือผ่านการตรวจสอบความถูกต้องด้วยวิธีอื่น ผู้ถือใบรับรองนั้นสามารถใช้คีย์สาธารณะที่มีอยู่เพื่อสร้างการสื่อสารที่ปลอดภัยกับบุคคลอื่น หรือตรวจสอบความถูกต้องของเอกสารที่ลงนามดิจิทัลด้วย คีย์ส่วนตัวที่เกี่ยวข้องได้

X.509 ยังกำหนดรายการเพิกถอนใบรับรองซึ่งเป็นวิธีการแจกจ่ายข้อมูลเกี่ยวกับใบรับรองที่ถือว่าไม่ถูกต้องโดยผู้มีอำนาจลงนาม ตลอดจนอัลกอริทึมการตรวจสอบเส้นทางการรับรองซึ่งช่วยให้ใบรับรองได้รับการลงนามโดยใบรับรอง CA ตัวกลาง ซึ่งใบรับรองเหล่านี้จะได้รับการลงนามโดยใบรับรองอื่น ๆ ต่อไปจนไปถึงจุดยึดที่เชื่อถือได้ในที่สุด

X.509 ถูกกำหนดโดย “Standardization Sector” ของ ITU (SG17 ของ ITU-T) ใน ITU-T Study Group 17 และมีพื้นฐานมาจาก Abstract Syntax Notation One (ASN.1) ซึ่งเป็นมาตรฐานอีกประการหนึ่งของ ITU-T

2.6.2.1 โครงสร้างของใบรับรอง

โครงสร้างที่กำหนดไว้โดยมาตรฐานจะแสดงอยู่ในภาษาทางการที่เรียกว่า Abstract Syntax Notation One (ASN.1)

โครงสร้างของใบรับรองดิจิทัล X.509 v3 มีดังนี้

1) ใบรับรอง

ก) หมายเลขเวอร์ชัน

ข) หมายเลขซีเรียล

ค) รหัสอัลกอริทึมลายเซ็น

ง) ชื่อผู้ออก

จ) ระยะเวลาใช้งาน โดยระบุเวลาไม่ก่อนและไม่หลังจากนั้น

ฉ) ชื่อเรื่อง

ช) ข้อมูลคีย์สาธารณะของเรื่อง ได้แก่ อัลกอริทึมคีย์สาธารณะ คีย์สาธารณะของเรื่องเช่นรหัสประจำตัวผู้ออก (ไม่จำเป็น) รหัสประจำตัวเฉพาะเรื่อง (ไม่จำเป็น) ส่วนขยาย (ไม่จำเป็น)

2) อัลกอริทึมลายเซ็นใบรับรอง

3) ลายเซ็นใบรับรอง

ฟิลด์ส่วนขยาย (ถ้ามี) จะเป็นลำดับของส่วนขยายใบรับรองอย่างน้อยหนึ่งรายการ แต่ละส่วนขยายมีรหัสประจำตัวเฉพาะของตัวเอง ซึ่งแสดงเป็นตัวระบุวัตถุ (OID) ซึ่งเป็นชุดค่าพร้อมกับข้อบ่งชี้ที่สำคัญหรือไม่สำคัญ ระบบที่ใช้ใบรับรองต้องปฏิเสธใบรับรองหากพบส่วนขยายที่สำคัญที่ไม่รู้จักหรือส่วนขยายที่สำคัญซึ่งมีข้อมูลที่ไม่สามารถประมวลผลได้ ส่วนขยายที่ไม่สำคัญอาจถูกละเว้นหากไม่รู้จัก แต่จะต้องได้รับการประมวลผลหากรู้จักส่วนขยายใบรับรอง

โครงสร้างของเวอร์ชัน 1 มีอยู่ ใน RFC 1422

รูปแบบภายในของตัวระบุเฉพาะของผู้เผยแพร่และเรื่องที่ระบุไว้ใน X.520 ไดเร็กทอรี :คำแนะนำ ประเภทแอตทริบิวต์ที่เลือก

ITU-T ได้นำตัวระบุเฉพาะของผู้ออกหลักทรัพย์และบุคคลมาใช้ในเวอร์ชัน 2 เพื่ออนุญาตให้นำชื่อผู้ออกหลักทรัพย์หรือบุคคลมาใช้ซ้ำได้หลังจากระยะเวลาหนึ่ง ตัวอย่างหนึ่งของการนำกลับมาใช้ซ้ำคือเมื่อ CA ล้มละลายและชื่อถูกลบออกจากรายชื่อสาธารณะของประเทศ หลังจากนั้น CA อื่นที่มีชื่อเดียวกันอาจลงทะเบียนตัวเองได้ แม้ว่าจะไม่เกี่ยวข้องกับ CA แรกก็ตาม อย่างไรก็ตาม IETF แนะนำว่าไม่ควรนำชื่อผู้ออกหลักทรัพย์และบุคคลมาใช้ซ้ำ ดังนั้นเวอร์ชัน 2 จึงยังไม่แพร่หลายในอินเทอร์เน็ต

ส่วนขยายได้รับการแนะนำในเวอร์ชัน 3 CA สามารถใช้ส่วนขยายเพื่อออกใบรับรองได้เฉพาะสำหรับจุดประสงค์เฉพาะ (เช่น สำหรับการลงนามในวัตถุดิจิทัล เท่านั้น)

ในทุกเวอร์ชันหมายเลขซีเรียลจะต้องไม่ซ้ำกันสำหรับใบรับรองแต่ละใบที่ออกโดย CA เฉพาะ (ดังที่กล่าวถึงใน RFC 5280)

2.6.2.2 นามสกุลไฟล์ใบรับรอง

นามสกุลไฟล์ที่ใช้กันทั่วไปสำหรับใบรับรอง X.509 มีหลายประเภทนามสกุลไฟล์เหล่านี้ยังใช้สำหรับข้อมูลอื่น ๆ เช่น คีย์ส่วนตัวด้วย

1) .pem – (อีเมลอิเล็กทรอนิกส์ที่เพิ่มความเป็นส่วนตัว) ใบรับรอง DER ที่เข้ารหัส Base64 แนบระหว่าง -----BEGIN CERTIFICATE----- และ -----END CERTIFICATE-----

2) .cer, .crt, .der – โดยปกติจะอยู่ในรูปแบบไบนารี DER แต่ใบรับรองที่เข้ารหัส Base64 ก็เป็นเรื่องปกติเช่นกัน (ดู .pem ด้านบน)

3) .p8, .p8e, .pk8 – คีย์ส่วนตัวที่ส่งออกตามที่ระบุไว้ใน PKCS#8 อาจอยู่ในรูปแบบ DER หรือ PEM ที่ขึ้นต้นด้วย -----BEGIN PRIVATE KEY----- คีย์ที่เข้ารหัสจะขึ้นต้นด้วย -----BEGIN ENCRYPTED PRIVATE KEY----- และอาจมี .p8e เป็นนามสกุลไฟล์

4) .p10, .csr – PKCS#10 เป็นคำขอลงนามใบรับรอง (CSR) ในรูปแบบ PEM ขึ้นต้นด้วย -----BEGIN CERTIFICATE REQUEST----- แบบฟอร์มเหล่านี้สร้างขึ้นเพื่อส่งไปยังผู้ออกใบรับรอง (CA) แบบฟอร์มประกอบด้วยรายละเอียดสำคัญของใบรับรองที่ร้องขอ เช่น ชื่อสามัญ (/CN), หัวเรื่อง, องค์กร, รัฐ, ประเทศ รวมถึงคีย์สาธารณะของใบรับรองที่ต้องการให้ลงนาม คีย์เหล่านี้จะได้รับการลงนามโดย CA และใบรับรองจะถูกส่งกลับคืน ใบรับรองที่ส่งคืนคือใบรับรอง สาธารณะ (ซึ่งมีคีย์สาธารณะแต่ไม่มีคีย์ส่วนตัว) ซึ่งตัวใบรับรองเองสามารถอยู่ในรูปแบบต่างๆ ได้หลายรูปแบบ แต่โดยปกติจะเป็น .p7r

5) .p7r – คำตอบ ของ PKCS#7 ต่อ CSR ประกอบด้วยใบรับรองที่เพิ่งลงนาม และใบรับรองของ CA เอง

6) .p7s – ลายเซ็นดิจิทัล PKCS#7 อาจมีไฟล์หรือข้อความที่ลงนามต้นฉบับ ใช้ใน S/MIME สำหรับการลงนามในอีเมลกำหนดไว้ใน RFC 2311

7) .p7m – PKCS#7 (SignedData, EnvelopedData) ข้อความ เช่น ไฟล์ที่เข้ารหัส (“enveloped”) ข้อความ หรือจดหมายอีเมล MIME กำหนดไว้ใน RFC 2311

8) .p7c – โครงสร้าง SignedData แบบ “certs-only” ของ PKCS#7 ที่เสื่อมลงโดยไม่มีข้อมูลใดๆ ให้ลงนาม กำหนดไว้ใน RFC 2311

9) .p7b – โครงสร้าง SignedData ของ PKCS#7 ที่ไม่มีข้อมูล มีเพียงใบรับรองแบบบันเดิลหรือ CRL (ไม่ค่อยเกิดขึ้น) แต่ไม่มีคีย์ส่วนตัว ใช้รูปแบบ DER หรือ BER หรือ PEM ที่ขึ้นต้นด้วย -----BEGIN PKCS7----- รูปแบบที่ Windows ใช้สำหรับการแลกเปลี่ยนใบรับรอง รองรับโดย Java แต่มักใช้นามสกุล .keystore แทน ซึ่งแตกต่างจากใบรับรองแบบ .pem รูปแบบนี้มีวิธีที่กำหนดไว้สำหรับการรวมใบรับรองเส้นทางการรับรอง

10) .p12, .pfx, .pkcs12 – PKCS#12 อาจมีใบรับรอง (สาธารณะ) และคีย์ส่วนตัว (ป้องกันด้วยรหัสผ่าน) ในไฟล์เดียว .pfx - *Personal Information eXchange* PFX ซึ่งเป็นรุ่นก่อนของ PKCS#12 (โดยปกติจะมีข้อมูลในรูปแบบ PKCS#12 เช่น ไฟล์ PFX ที่สร้างใน IIS)

11) .crl – รายการเพิกถอนใบรับรอง (CRL) หน่วยงานที่ออกใบรับรองจะจัดทำรายการเหล่านี้ขึ้นเพื่อใช้ในการเพิกถอนใบรับรองก่อนหมดอายุ

PKCS#7 เป็นมาตรฐานสำหรับการลงนามหรือเข้ารหัสข้อมูล (เรียกอย่างเป็นทางการว่า “enveloping”) เนื่องจากจำเป็นต้องใช้ใบรับรองเพื่อตรวจสอบข้อมูลที่ลงนามแล้วจึงสามารถรวมใบรับรองไว้ในโครงสร้าง SignedData ได้

2.6.3 X.690 (การเข้ารหัส DER)

X.690 เป็น มาตรฐาน ITU-T ที่ระบุรูปแบบการเข้ารหัส ASN.1 หลายรูปแบบ

2.6.3.1 กฎการเข้ารหัสพื้นฐาน (BER)

กฎการเข้ารหัสพื้นฐาน (BER) คือกฎดั้งเดิมที่วางไว้โดยมาตรฐาน ASN.1 สำหรับการเข้ารหัสข้อมูลในรูปแบบไบนารี กฎเหล่านี้ ซึ่งเรียกรวมกันว่าไวยากรณ์การถ่ายโอนในภาษา ASN.1 กำหนดจำนวนอ็อกเท็ต (ไบต์ 8 บิต) ที่ใช้ในการเข้ารหัสข้อมูล

2.6.3.2 กฎการเข้ารหัสที่โดดเด่น (DER)

โดยโครงงานนี้ใช้ใบรับรองในรูปแบบการเข้ารหัส DER ซึ่ง DER (Distinguished Encoding Rules) เป็น BER แบบจำกัดรูปแบบหนึ่งสำหรับการสร้างไวยากรณ์การถ่ายโอนข้อมูลที่ชัดเจนสำหรับโครงสร้างข้อมูลที่อธิบายโดย ASN.1 เช่นเดียวกับ CER การเข้ารหัส DER ถือเป็นการเข้ารหัส BER ที่ถูกต้อง DER เหมือนกับ BER โดยตัดตัวเลือกของผู้ส่งออกทั้งหมด ยกเว้นตัวเลือกเดียว

DER เป็นส่วนย่อยของ BER ที่ให้วิธีการเข้ารหัสค่า ASN.1 เพียงวิธีเดียว DER มีไว้สำหรับสถานการณ์ที่จำเป็นต้องมีการเข้ารหัสเฉพาะ เช่น ในการเข้ารหัสลับและช่วยให้มั่นใจว่าโครงสร้างข้อมูลที่จำเป็นต้องมีการลงนามดิจิทัลจะสร้างการแสดงแบบอนุกรมที่ไม่ซ้ำกัน DER ถือเป็นรูปแบบมาตรฐานของ BER ตัวอย่างเช่นใน BER ค่าบูลีน true สามารถเข้ารหัสเป็นค่าไบต์ที่ไม่ใช่ศูนย์ 255 ค่า ในขณะที่ DER มีวิธีการเข้ารหัสค่าบูลีน true เพียงวิธีเดียว

2.6.4 OpenSSL

OpenSSL คือไลบรารีซอฟต์แวร์สำหรับแอปพลิเคชันที่ให้การสื่อสารที่ปลอดภัยผ่านเครือข่ายคอมพิวเตอร์ป้องกันการดักฟังและระบุตัวบุคคลที่อยู่ปลายทาง OpenSSL ถูกใช้อย่างแพร่หลายในเซิร์ฟเวอร์อินเทอร์เน็ตรวมถึงเว็บไซต์ HTTPS ส่วนใหญ่ OpenSSL ประกอบด้วยการนำโปรโตคอล SSL และ TLS ไปใช้งานแบบโอเพนซอร์สไลบรารีหลักที่เขียนด้วยภาษา C ทำหน้าที่เข้ารหัสข้อมูลพื้นฐานและมีฟังก์ชันยูทิลิตี้ต่างๆมากมาย มีแรปเปอร์ (Wrapper) ที่ช่วยให้สามารถใช้ไลบรารี OpenSSL ในภาษาคอมพิวเตอร์ได้หลากหลายภาษา

มูลนิธิซอฟต์แวร์ OpenSSL (OSF) เป็นตัวแทนของโครงการ OpenSSL ในขอบเขตทางกฎหมายส่วนใหญ่ ซึ่งรวมถึงข้อตกลงสิทธิ์การใช้งานสำหรับผู้สนับสนุนการจัดการการบริจาค และอื่นๆบริการซอฟต์แวร์ OpenSSL (OSS) ยังเป็นตัวแทนของโครงการ OpenSSL สำหรับสัญญาสนับสนุนอีกด้วย

OpenSSL พร้อมใช้งานสำหรับระบบปฏิบัติการประเภท Unix ส่วนใหญ่ (รวมถึง Linux , macOS และ BSD), Microsoft Windows และ OpenVMS

OpenSSL รองรับอัลกอริทึมการเข้ารหัสที่แตกต่างกันจำนวนหนึ่ง โดยฟังก์ชันการเข้ารหัสได้แก่ AES, Blowfish, Camellia, ChaCha20, Poly1305, SEED, CAST-128, DES, IDEA, RC2, RC4, RC5, Triple DES, GOST 28147-89, SM4

ฟังก์ชันแฮชการเข้ารหัสได้แก่ MD5, MD4, MD2, SHA-1, SHA-2, SHA-3, RIPEMD-160, MDC-2, GOST R 34.11-94, BLAKE2, วังวน, SM3

ฟังก์ชันการเข้ารหัสด้วยคีย์สาธารณะได้แก่ RSA, DSA, การแลกเปลี่ยนคีย์ Diffie–Hellman, เส้นโค้งวงรี, X25519, Ed25519, X448, Ed448, GOST R 34.10-2001, SM2
(การปกปิดแบบสมบูรณ์แบบได้รับการสนับสนุนโดยใช้เส้นโค้งวงรี Diffie–Hellman ตั้งแต่เวอร์ชัน 1.0)

2.7 การสื่อสารสนามใกล้ (Near-field communication; NFC)

การสื่อสารแบบใกล้สนาม (NFC) คือชุดโปรโตคอลการสื่อสารที่ทำให้สามารถสื่อสารระหว่างอุปกรณ์อิเล็กทรอนิกส์สองเครื่องในระยะทาง 4 ซม. ($1\frac{1}{2}$ นิ้ว) หรือน้อยกว่า NFC นำเสนอการเชื่อมต่อความเร็วต่ำผ่านการตั้งค่าที่ง่ายดายซึ่งสามารถใช้สำหรับการบูตสแตรปของการเชื่อมต่อไร้สายที่สามารถใช้งานได้เช่นเดียวกับเทคโนโลยีการ์ดระยะใกล้อื่นๆ NFC มีพื้นฐานมาจากการเชื่อมต่อแบบเหนี่ยวนำ ระหว่างขดลวดแม่เหล็กไฟฟ้าสองอันบนอุปกรณ์ที่รองรับ NFC เช่นสมาร์ทโฟนการสื่อสาร NFC ในทิศทางเดียวหรือทั้งสองทิศทางใช้ความถี่ 13.56 MHz ในย่านความถี่วิทยุ ISM ที่ไม่มีใบอนุญาต ซึ่งใช้กันทั่วโลก สอดคล้องกับมาตรฐานอินเทอร์เฟซทางอากาศ ISO/IEC 18000-3 ที่อัตราข้อมูลตั้งแต่ 106 ถึง 848 กิโลบิต/วินาที

ฟอรัม NFC ได้ช่วยกำหนดและส่งเสริมเทคโนโลยีโดยกำหนดมาตรฐานสำหรับการรับรองการปฏิบัติตามข้อกำหนดของอุปกรณ์การสื่อสารที่ปลอดภัยสามารถทำได้โดยใช้ขั้นตอนวิธีการเข้ารหัสเช่นเดียวกับที่ใช้กับบัตรเครดิตและหากตรงตามเกณฑ์สำหรับการพิจารณาให้เป็นเครือข่ายพื้นที่ส่วนบุคคล

2.7.1 มาตราฐาน NFC

มาตรฐาน NFC ครอบคลุมโปรโตคอลการสื่อสารและรูปแบบการแลกเปลี่ยนข้อมูล และอิงตามมาตรฐานการระบุด้วยคลื่นความถี่วิทยุ (RFID) ที่มีอยู่ รวมถึง ISO/IEC 14443 และ FeliCa มาตรฐานเหล่านี้รวมถึง ISO/IEC 18092 และมาตรฐานที่กำหนดโดย NFC Forum นอกจาก NFC Forum แล้วกลุ่ม GSMA ยังได้กำหนดแพลตฟอร์มสำหรับการปรับใช้มาตรฐาน NFC ของ GSMA ภายในโทรศัพท์มือถือ ความพยายามของ GSMA ได้แก่ Trusted Services Manager, Single Wire Protocol, การทดสอบ/การรับรอง และองค์ประกอบความปลอดภัยอุปกรณ์พกพาที่เปิดใช้งาน NFC สามารถมาพร้อมกับซอฟต์แวร์แอปพลิเคชันเช่น เพื่ออ่านแท็กอิเล็กทรอนิกส์หรือชำระเงินเมื่อเชื่อมต่อกับระบบที่รองรับ NFC สิ่งเหล่านี้เป็นมาตรฐานของโปรโตคอล NFC แทนที่เทคโนโลยีที่เป็นกรรมสิทธิ์ที่ใช้ในระบบก่อนหน้านี้

โปรแกรมอนุญาตสิทธิ์สิทธิบัตรสำหรับ NFC กำลังอยู่ระหว่างการใช้งานโดย France Brevets ซึ่งเป็นกองทุนสิทธิบัตรที่จัดตั้งขึ้นในปี 2011 โปรแกรมนี้อยู่ระหว่างการพัฒนาโดย Via Licensing Corporation ซึ่งเป็นบริษัทสาขาอิสระของ Dolby Laboratories และยุติลงในเดือนพฤษภาคม 2012 ไลบรารี NFC แบบโอเพนซอร์สและอิสระต่อแพลตฟอร์ม libnfc มีให้บริการภายใต้ใบอนุญาต GNU Lesser General Public License (LGPL)

แอปพลิเคชันปัจจุบันและที่คาดว่าจะมีในอนาคต ได้แก่ ธุรกรรมแบบไร้สัมผัส การแลกเปลี่ยนข้อมูล และการตั้งค่าการสื่อสารที่ซับซ้อนมากขึ้น เช่น Wi-Fi ที่ง่าย ขึ้นนอกจากนี้เมื่ออุปกรณ์ที่เชื่อมต่อเครื่องหนึ่งมีการเชื่อมต่ออินเทอร์เน็ตอีกเครื่องหนึ่งก็สามารถแลกเปลี่ยนข้อมูลกับบริการออนไลน์ได้

2.7.2 ออกแบบ

NFC เป็นชุดเทคโนโลยีไร้สายระยะสั้น โดยทั่วไปต้องมีระยะห่าง 10 ซม. (3⅞ นิ้ว) หรือน้อยกว่า NFC ทำงานที่ความถี่ 13.56 MHz บนอินเทอร์เฟซทางอากาศ ISO/IEC 18000-3 และที่อัตราตั้งแต่ 106 กิโลบิต/วินาที ถึง 424 กิโลบิต/วินาที NFC มักประกอบด้วยตัวเริ่มต้นและเป้าหมาย ตัวเริ่มต้นจะสร้างสนาม RF ที่สามารถจ่ายพลังงานให้กับเป้าหมายแบบพาสซีฟได้ ซึ่งทำให้เป้าหมาย NFC มีรูปแบบที่เรียบง่ายมาก เช่น แท็ก สติกเกอร์ พวงกุญแจ หรือการ์ดที่ไม่ได้รับพลังงานการสื่อสารแบบเพียร์ทูเพียร์ของ NFC สามารถทำได้หากอุปกรณ์ทั้งสองมีพลังงาน

แท็ก NFC มีข้อมูลและโดยทั่วไปเป็นแบบอ่านอย่างเดียว แต่อาจเขียนได้ ผู้ผลิตสามารถกำหนดรหัสเองได้ หรือใช้ข้อกำหนดของ NFC Forum แท็กสามารถจัดเก็บข้อมูลส่วนบุคคลอย่างปลอดภัย เช่น ข้อมูลบัตรเดบิตและบัตรเครดิต ข้อมูลโปรแกรมสะสมคะแนน รหัส PIN และรายชื่อผู้ติดต่อในเครือข่าย รวมถึงข้อมูลอื่นๆ NFC Forum กำหนดแท็กห้าประเภทที่มีความเร็วและความสามารถในการสื่อสารที่แตกต่างกันในแง่ของความสามารถในการกำหนดค่าหน่วยความจำ ความปลอดภัยการเก็บข้อมูลและความทนทานต่อการเขียน

เช่นเดียวกับ เทคโนโลยี การ์ดแบบ Proximity NFC ใช้การเชื่อมต่อแบบเหนี่ยวนำระหว่างเสาอากาศแบบวงสองต้นที่อยู่ใกล้เคียงกัน ซึ่งก่อตัวเป็นหม้อแปลงแกนอากาศได้อย่างมีประสิทธิภาพ เนื่องจากระยะทางที่เกี่ยวข้องนั้นน้อยมากเมื่อเทียบกับความยาวคลื่นของรังสีแม่เหล็กไฟฟ้า (คลื่นวิทยุ) ของความถี่นั้น (ประมาณ 22 เมตร) ปฏิสัมพันธ์นี้จึงถูกเรียกว่า สนามแม่เหล็กใกล้ (Near Field) สนามแม่เหล็กไฟฟ้าสลับเป็นปัจจัยการเชื่อมต่อหลัก และแทบไม่มีพลังงานแผ่ออกมาในรูปแบบของคลื่นวิทยุ (ซึ่งเป็นคลื่นแม่เหล็กไฟฟ้าที่เกี่ยวข้องกับสนามไฟฟ้าสั่นด้วย) ซึ่งช่วยลดการรบกวนระหว่างอุปกรณ์ดังกล่าวกับการสื่อสารทางวิทยุใดๆ ที่ความถี่เดียวกันหรือกับอุปกรณ์ NFC อื่นๆ ที่อยู่นอกเหนือขอบเขตที่ตั้งใจไว้ NFC ทำงานในย่านความถี่วิทยุ ISM ซึ่งใช้ทั่วโลกและไม่ได้รับอนุญาตที่ 13.56 MHz พลังงาน RF ส่วนใหญ่กระจุกตัวอยู่ในแบนด์วิดท์ ±7 kHz ที่จัดสรรให้กับย่านความถี่นั้น แต่ ความกว้างสเปกตรัมของการแผ่รังสีอาจกว้างได้ถึง 1.8 MHz เพื่อรองรับอัตราข้อมูลสูง

ระยะการทำงานด้วยเสาอากาศมาตรฐานขนาดกะทัดรัดและระดับพลังงานที่สมจริงอาจสูงถึงประมาณ 20 ซม. (7⅞ นิ้ว) (แต่ในทางปฏิบัติ ระยะการทำงานไม่ควรเกิน 10 ซม. หรือ 3⅞ นิ้ว) โปรดทราบว่าเนื่องจากเสาอากาศรับสัญญาณอาจถูกดับในกระแสวนโดยพื้นผิวโลหะที่อยู่ใกล้เคียง แท็กอาจต้องแยกออกจากพื้นผิวดังกล่าวอย่างน้อยที่สุด

มาตรฐาน ISO/IEC 18092 รองรับอัตราข้อมูล 106, 212 หรือ 424 กิโลบิต/วินาที

การสื่อสารเกิดขึ้นระหว่างอุปกรณ์ “ตัวเริ่มต้น” ที่ใช้งานอยู่และอุปกรณ์เป้าหมาย ซึ่งอาจเป็น

1) พาสซีฟ โดยอุปกรณ์ตัวเริ่มต้นจะทำหน้าที่เป็นสนามแม่เหล็กพาหะ และอุปกรณ์เป้าหมายจะสื่อสารโดยการปรับสนามแม่เหล็กตกกระทบ ในโหมดนี้ อุปกรณ์เป้าหมายอาจดึงพลังงานจากสนามแม่เหล็กที่ตัวเริ่มต้นจัดหาให้

2) คล่องแคล่ว โดยทั้งอุปกรณ์เริ่มต้นและอุปกรณ์เป้าหมายสื่อสารกันโดยการสร้างฟิลด์ของตัวเองสลับกัน อุปกรณ์จะหยุดส่งสัญญาณเพื่อรับข้อมูลจากอีกอุปกรณ์หนึ่ง โหมดนี้กำหนดให้อุปกรณ์ทั้งสองต้องมีแหล่งจ่ายไฟ

ตารางที่ 2.2 การเทียบความเร็วและวิธีการสื่อสารที่ใช้

ความเร็ว (กิโลบิต/วินาที)	**อุปกรณ์ที่ใช้งานอยู่**	**อุปกรณ์แบบพาสซีฟ**
424	แมนเชสเตอร์ 10% ASK	แมนเชสเตอร์ 10% ASK
212	แมนเชสเตอร์ 10% ASK	แมนเชสเตอร์ 10% ASK
106	มิลเลอร์ดัดแปลง 100% ASK	แมนเชสเตอร์ 10% ASK

NFC ใช้การเข้ารหัสสองแบบที่แตกต่างกันในการถ่ายโอนข้อมูล หากอุปกรณ์ที่ใช้งานอยู่ถ่ายโอนข้อมูลที่ความเร็ว 106 กิโลบิต/วินาที จะใช้การเข้ารหัสแบบมิลเลอร์ที่ปรับเปลี่ยนแล้วพร้อมการมอดูเลต 100 เปอร์เซ็นต์ในกรณีอื่นๆทั้งหมดจะใช้การเข้ารหัสแบบแมนเชสเตอร์โดยมีอัตราการมอดูเลต 10 เปอร์เซ็นต์

อุปกรณ์ NFC ที่ใช้งานอยู่ทุกเครื่องสามารถทำงานในโหมดใดโหมดหนึ่งหรือหลายโหมดได้

1) การจำลองการ์ด NFC ช่วยให้อุปกรณ์ที่รองรับ NFC เช่น สมาร์ทโฟน ทำหน้าที่เหมือนสมาร์ทการ์ด ช่วยให้ผู้ใช้ทำธุรกรรมต่างๆ เช่น การชำระเงินหรือการออกตั๋วได้ ดูการจำลองการ์ดโฮสต์

2) เครื่องอ่าน/เขียน NFC ช่วยให้อุปกรณ์ที่เปิดใช้งาน NFC สามารถอ่านข้อมูลที่จัดเก็บไว้ในแท็ก NFC ราคาไม่แพงที่ฝังอยู่ในฉลากหรือโปสเตอร์อัจฉริยะได้

3) NFC เพียร์ทูเพียร์ ช่วยให้อุปกรณ์ที่เปิดใช้งาน NFC สองเครื่องสามารถสื่อสารกันเพื่อแลกเปลี่ยนข้อมูลในลักษณะ เฉพาะกิจ

แท็ก NFC คือหน่วยเก็บข้อมูลแบบพาสซีฟที่อุปกรณ์ NFC สามารถอ่านและเขียนข้อมูลได้ในบางกรณี โดยทั่วไปจะมีข้อมูล (ณ ปี 2015 มีขนาดระหว่าง 96 ถึง 8,192 ไบต์) และเป็นแบบอ่านอย่างเดียวในการใช้งานปกติ แต่อาจเขียนซ้ำได้ การใช้งานรวมถึงการจัดเก็บข้อมูลส่วนบุคคลที่ปลอดภัย (เช่นข้อมูลบัตร เดบิตหรือบัตรเครดิตข้อมูลโปรแกรมสะสมคะแนน หมายเลขประจำตัว (PIN) และรายชื่อผู้ติดต่อ) แท็ก NFC สามารถเข้ารหัสแบบกำหนดเองโดยผู้ผลิต หรือใช้ข้อกำหนดเฉพาะของอุตสาหกรรม

2.8 Flutter

Flutter เป็นชุดพัฒนาซอฟต์แวร์ UI แบบโอเพนซอร์สที่สร้างโดย Google สามารถใช้พัฒนาแอปพลิเคชันข้ามแพลตฟอร์มจากฐานโคดเดียวสำหรับเว็บ Fuchsia, Android, iOS, Linux, macOS และ Windows โดย Flutter ได้รับการพูดถึงครั้งแรกในปี 2015 และเปิดตัวในเดือนพฤษภาคม 2017 และ Flutter ถูกใช้งานภายในโดย Google ในแอปพลิเคชันต่างๆ เช่น Google Pay และ Google Earth รวมถึงโดยนักพัฒนาซอฟต์แวร์รายอื่นๆ เช่น ByteDance และ Alibaba

Flutter จะสร้างแอปพลิเคชันที่มีเอ็นจิ้นการเรนเดอร์ของตัวเอง ซึ่งส่งข้อมูลพิกเซลไปยังหน้าจอโดยตรง ซึ่งแตกต่างจากเฟรมเวิร์ก UI อื่น ๆ อีกมากมายที่อาศัยแพลตฟอร์มเป้าหมายเพื่อจัดหาเอ็นจิ้นการเรนเดอร์ เช่น แอป Android พื้นฐานที่ใช้ Android SDK ระดับอุปกรณ์ หรือ iOS SDK ที่ใช้ UI stack ในตัวของแพลตฟอร์มเป้าหมาย การควบคุมขั้นตอนการแสดงผลของ Flutter ช่วยลดความยุ่งยากในการรองรับหลายแพลตฟอร์ม เนื่องจากสามารถใช้โคด UI ที่เหมือนกันได้กับทุกแพลตฟอร์มเป้าหมาย

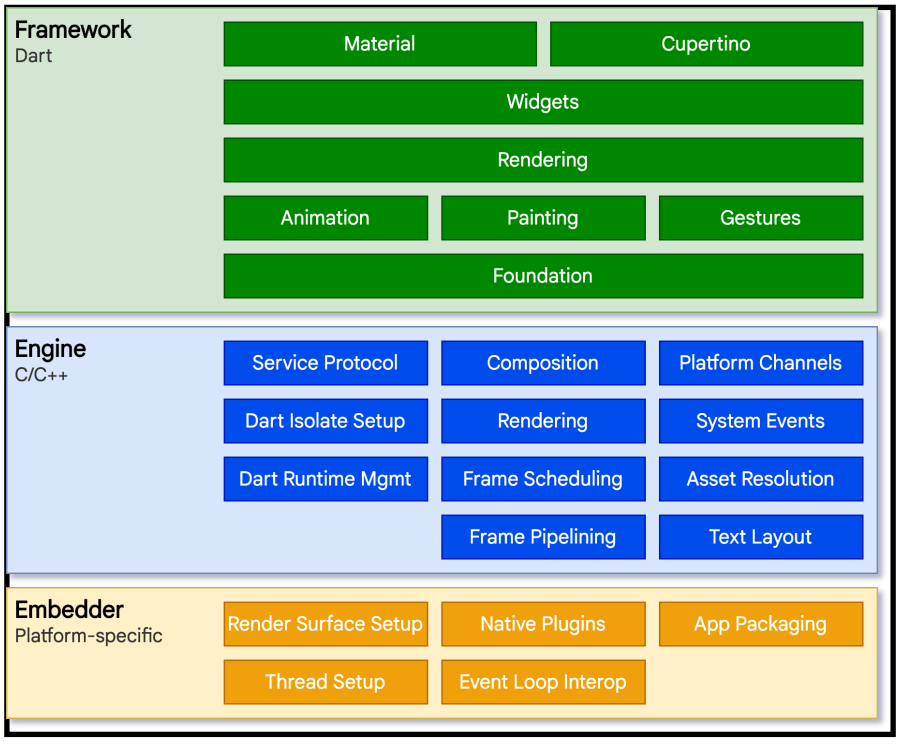
2.8.1 Dart

Dart เป็นภาษาโปรแกรมที่ออกแบบโดย Lars Bak และ Kasper Lund และพัฒนาโดย Google สามารถใช้พัฒนาแอปพลิเคชันบนเว็บ มือถือ เซิร์ฟเวอร์ และเดสก์ท็อปได้ และยังเป็นภาษาหลักที่ใช้ในการพัฒนาแอปพลิเคชัน Flutter

Dart เป็นภาษาเชิงวัตถุ อิงคลาส และรวบรวมขยะ (garbage-collection) ด้วยไวยากรณ์แบบ C สามารถคอมไพล์เป็นโค้ดเครื่อง JavaScript หรือ WebAssembly ได้ รองรับอินเทอร์เฟซ มิกซ์อิน คลาสนามธรรม เจเนอริกแบบรีไฟด์ และการอนุมานชนิดข้อมูล

2.8.2 สถาปัตยกรรม

Flutter ถูกออกแบบมาให้เป็นระบบแบบเลเยอร์ที่ต่อขยายได้ ประกอบด้วยไลบรารีอิสระหลายชุดที่แต่ละชุดพึ่งพาเลเยอร์ที่อยู่ด้านล่าง ไม่มีเลเยอร์ใดที่มีสิทธิ์พิเศษในการเข้าถึงเลเยอร์ด้านล่าง และทุกส่วนของเฟรมเวิร์กถูกออกแบบมาให้เป็นตัวเลือกและสามารถทดแทนได้



รูปที่ 2.19 สถาปัตยกรรม Flutter

สำหรับระบบปฏิบัติการที่อยู่ภายใต้ แอปพลิเคชัน Flutter จะถูกบรรจุในลักษณะเดียวกับแอปพลิเคชันเนทีฟอื่น ๆ โดยตัวฝังตัว (Embedder) เฉพาะแพลตฟอร์มจะทำหน้าที่เป็นจุดเริ่มต้นประสานงานกับระบบปฏิบัติการที่อยู่ภายใต้เพื่อเข้าถึงบริการต่างๆ เช่น พื้นผิวการแสดงผล การเข้าถึงและการป้อนข้อมูล และจัดการลูปเหตุการณ์ข้อความ ตัวฝังตัวเขียนด้วยภาษาที่เหมาะสมกับแพลตฟอร์มปัจจุบันคือ Java และ C++ สำหรับ Android, Swift และ Objective-C/Objective-C++ สำหรับ iOS และ macOS และ C++ สำหรับ Windows และ Linux การใช้ตัวฝังตัว โค้ด Flutter สามารถรวมเข้ากับแอปพลิเคชันที่มีอยู่แล้วในรูปแบบโมดูล หรือโค้ดอาจเป็นเนื้อหาทั้งหมดของแอปพลิเคชันก็ได้ Flutter มีตัวฝังตัวจำนวนมากสำหรับแพลตฟอร์มเป้าหมายทั่วไป แต่ก็ยังมีตัวฝังตัวอื่นๆ อีกด้วย

หัวใจหลักของ Flutter คือ Flutter engine ซึ่งส่วนใหญ่เขียนด้วยภาษา C++ และรองรับฟังก์ชันพื้นฐานที่จำเป็นต่อการทำงานของแอปพลิเคชัน Flutter ทั้งหมด เอนจินนี้มีหน้าที่ในการแปลงฉากที่ประกอบขึ้นเป็นภาพแรสเตอร์ทุกครั้งที่จำเป็นต้องวาดเฟรมใหม่ มันมีหน้าที่ให้การใช้งานระดับต่ำของ API หลักของ Flutter รวมถึงการจัดวางข้อความกราฟิก การรับส่งข้อมูลไฟล์และเครือข่าย รันไทม์ Dart และเครื่องมือคอมไพล์

เอนจินนี้ถูกเปิดเผยสู่เฟรมเวิร์ก Flutter ผ่านทาง dart:ui ซึ่งห่อหุ้มโค้ด C++ ที่อยู่เบื้องหลังด้วยคลาส Dart ไลบรารีนี้เปิดเผยส่วนประกอบพื้นฐานระดับต่ำสุด เช่น คลาสสำหรับควบคุมระบบย่อยการรับข้อมูล กราฟิก และการแสดงผลข้อความ

โดยทั่วไป นักพัฒนาจะโต้ตอบกับ Flutter ผ่านเฟรมเวิร์ก Flutter ซึ่งเป็นเฟรมเวิร์กที่ทันสมัยและตอบสนองต่อสิ่งรอบข้าง เขียนด้วยภาษา Dart เฟรมเวิร์กนี้ประกอบด้วยชุดไลบรารี แพลตฟอร์มเลย์เอาต์ และพื้นฐานที่ครบครัน ซึ่งประกอบด้วยเลเยอร์หลายชั้น เริ่มจากล่างขึ้นบน ได้แก่

1) คลาสพื้นฐานและบริการส่วนประกอบต่างๆ เช่น แอนิเมชัน การวาดภาพ และท่าทางสัมผัส ซึ่งนำเสนอนามธรรมที่ใช้กันทั่วไปเหนือพื้นฐานที่อยู่เบื้องหลัง

2) เลเยอร์การเรนเดอร์ให้นามธรรมสำหรับการจัดการเลย์เอาต์ ด้วยเลเยอร์นี้ คุณสามารถสร้างโครงสร้างแบบต้นไม้ของวัตถุที่เรนเดอร์ได้ คุณสามารถจัดการวัตถุเหล่านี้แบบไดนามิก โดยโครงสร้างแบบต้นไม้จะอัปเดตเลย์เอาต์โดยอัตโนมัติเพื่อสะท้อนการเปลี่ยนแปลงของคุณ

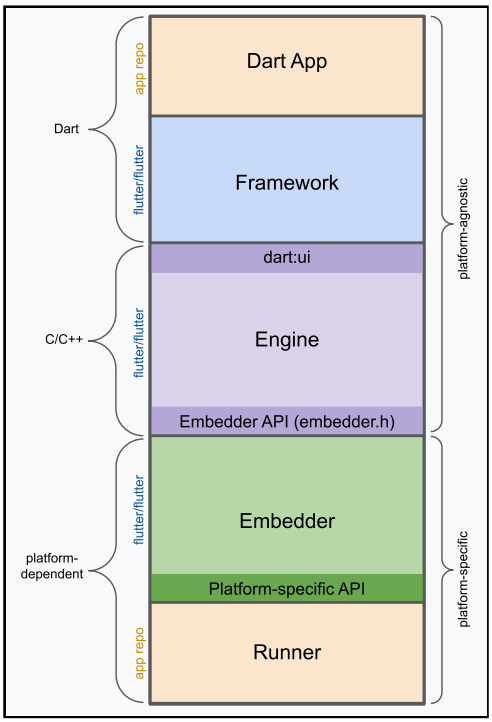
3) เลเยอร์วิดเจ็ตเป็นนามธรรมของการประกอบ วัตถุเรนเดอร์แต่ละชิ้นในเลเยอร์การเรนเดอร์จะมีคลาสที่สอดคล้องกันในเลเยอร์วิดเจ็ต นอกจากนี้ เลเยอร์วิดเจ็ตยังช่วยให้คุณกำหนดการรวมกันของคลาสที่คุณสามารถนำกลับมาใช้ใหม่ได้ นี่คือเลเยอร์ที่แนะนำโมเดลการเขียนโปรแกรมแบบตอบสนอง

4) ไลบรารี Material และ Cupertino นำเสนอชุดควบคุมที่ครอบคลุมซึ่งใช้ส่วนประกอบพื้นฐานของเลเยอร์วิดเจ็ตเพื่อนำภาษาการออกแบบ Material หรือ iOS ไปใช้

เฟรมเวิร์ก Flutter มีขนาดค่อนข้างเล็ก ฟีเจอร์ระดับสูงหลายอย่างที่นักพัฒนาอาจใช้ถูกพัฒนาขึ้นมาในรูปแบบของแพ็กเกจ รวมถึงปลั๊กอินของแต่ละแพลตฟอร์ม เช่น กล้องและเว็บวิว ตลอดจนฟีเจอร์ที่ไม่ขึ้นกับแพลตฟอร์ม เช่น ตัวอักษร, HTTP และแอนิเมชัน ซึ่งสร้างขึ้นจากไลบรารีหลักของ Dart และ Flutter แพ็กเกจบางส่วนมาจากระบบนิเวศที่กว้างกว่า ครอบคลุมบริการต่างๆ เช่น การชำระเงินภายในแอป การตรวจสอบสิทธิ์ของ Apple และแอนิเมชัน

2.8.3 โครงสร้างของแอปพลิเคชัน

แผนภาพต่อไปนี้แสดงภาพรวมของส่วนประกอบต่างๆ ที่ประกอบกันเป็นแอป Flutter ทั่วไปที่สร้างขึ้นโดยคำสั่ง flutter create แผนภาพนี้แสดงตำแหน่งของ Flutter Engine ในโครงสร้างนี้ เน้นขอบเขตของ API และระบุที่เก็บโค้ด (repository) ที่ส่วนประกอบแต่ละส่วนอยู่ คำอธิบายด้านล่างจะอธิบายคำศัพท์บางคำที่ใช้กันทั่วไปในการอธิบายส่วนประกอบของแอป Flutter



รูปที่ 2.20 เลเยอร์ต่าง ๆ ของแอพลิเคชัน Flutter

2.8.3.1 แอปพลิเคชัน Dart (Dart app)

1) ประกอบวิดเจ็ตเข้าด้วยกันเพื่อสร้าง UI ที่ต้องการ

2) ดำเนินการตามตรรกะทางธุรกิจ

3) นักพัฒนาแอปเป็นเจ้าของ

2.8.3.2 เฟรมเวิร์ก (Framework)

1) ให้ API ระดับสูงสำหรับการสร้างแอปคุณภาพสูง (ตัวอย่างเช่น วิดเจ็ต การทดสอบการกด การตรวจจับท่าทาง การเข้าถึงได้ และการอินพุต ข้อความ)

2) ประกอบต้นวิดเจ็ตของแอปพลิเคชันเป็นฉาก

2.8.3.3 เอนจิน (Engine)

1) มีหน้าที่แปลงฉากเป็นรูปแบบแรสเตอร์

2) ให้การทำงานระดับต่ำของแกนกลางของ Flutter API (เช่น กราฟิก การจัดข้อความ และรันไทม์ Dart)

3) เปิดเผยฟังก์ชันระดับนี้ให้แก่เฟรมเวิร์กผ่าน API dart:ui

4) บูรณาการกับแพลตฟอร์มต่าง ๆ ด้วย API ตัวฝังตัว

2.8.3.4 ตัวฝังตัว (Embedder)

1) ประสานงานกับระบบปฏิบัติการภายใต้สำหรับการเข้าถึงบริการต่าง ๆ เช่น พื้นผิวการเรนเดอร์ การเข้าถึง และการป้อนข้อมูล

2) จัดการลูปอิเวนต์

3) เปิดเผย API เฉพาะแพลตฟอร์มเพื่อบูรณาการตัวฝังตัวเข้าไปยังแอป

2.8.3.5 ตัวรัน (Runner)

1) ประกอบชิ้นส่วนที่ถูกเปิดเผยโดยตัวฝังตัวเข้าเป็นแพคเกจแอปพลิเคชันที่สามารถใช้งานได้บนแพลตฟอร์มเป้าหมาย

2) บางส่วนถูกสร้างขึ้นโดย flutter create และมีเจ้าของเป็นผู้พัฒนาแอป

2.8.4 ระบบการดีไซน์

โดยใน Flutter แล้วนั้น ไม่รวมแพคเกจบุคคลที่สาม จะมีระบบการดีไซน์อยู่สองแบบคือ:

1) Material Design คือการดีไซน์ของ Google สำหรับ Android

2) Cupertino Design คือการดีไซน์ของ Apple สำหรับ iOS

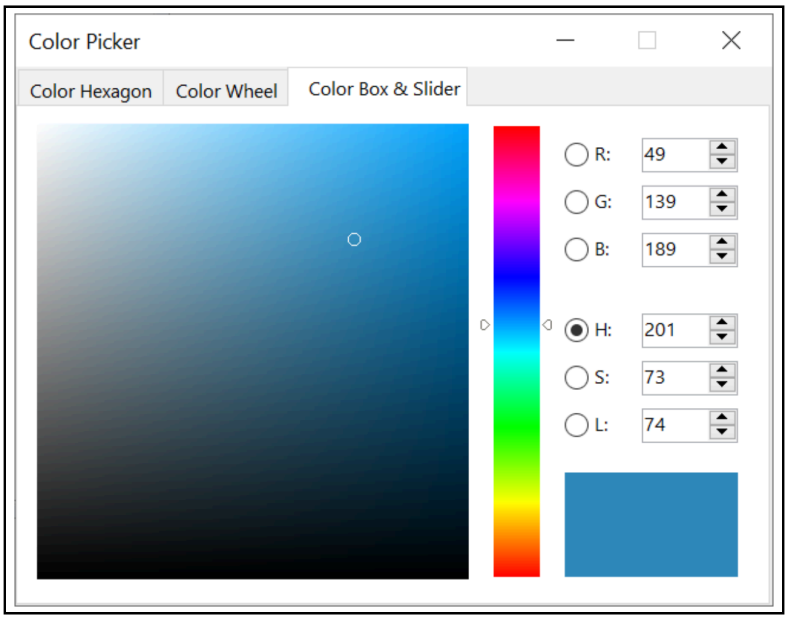
อย่างไรก็ตาม Cupertino Design ถูกแทนที่โดย Liquid Glass แล้ว โดยในปัจจุบันทีม Flutter กำลังทำการตรวจสอบและแก้ไขโครงสร้างระบบดีไซน์ ดังนั้น หากมีผู้พัฒนาต้องการใช้เอฟเฟกต์ Liquid Glass ในแอปพลิเคชัน Flutter จึงจำเป็นต้องพึงพาแพคเกจบุคคลที่สามก่อนในขณะนี้ (Flutter เวอร์ชัน 3.38.5 ณ เวลาที่พิมพ์)

Material Design คือภาษาการดีไซน์ที่ถูกพัฒนาโดย Google และถูกเปิดตัวครั้งแรก 25 มิถุนายน 2014 และมีเวอร์ชันหลัก 3 เวอร์ชันด้วยกัน โดยที่เวอร์ชันที่ 3 ถูกเปิดตัวในงาน Google I/O 2021 และมีชื่อว่า “Material You” (แต่ชื่อธรรมดา “Material Design 3” ก็ยังถูกใช้งานกันอย่างปกติ) และในงาน Google I/O 2025 มีการเปิดตัว “Material 3 Expressive” ซึ่งเป็นการปรับปรุงต่อจาก Material You เดิมสำหรับ Android 16 และ Wear OS 6 และสามารถดูรายละเอียดเพิ่มเติมเกี่ยวกับ Material 3 ได้ที่ https://m3.material.io/

2.8.5 ส่วนติดต่อผู้ใช้ที่มีปฏิกิริยา

บนพื้นผิว Flutter เป็นเฟรมเวิร์ก UI แบบ reactive และ declarative ซึ่งนักพัฒนาเป็นผู้จัดเตรียมการแมปจากสถานะแอปพลิเคชันไปยังสถานะอินเทอร์เฟซ และเฟรมเวิร์กจะทำหน้าที่อัปเดตอินเทอร์เฟซขณะรันไทม์เมื่อสถานะของแอปพลิเคชันเปลี่ยนแปลง โมเดลนี้ได้รับแรงบันดาลใจจากงานที่มาจาก Facebook สำหรับเฟรมเวิร์ก React ของพวกเขา ซึ่งรวมถึงการทบทวนหลักการออกแบบแบบดั้งเดิมหลายประการ

ในเฟรมเวิร์ก UI แบบดั้งเดิมส่วนใหญ่ สถานะเริ่มต้นของอินเทอร์เฟซผู้ใช้จะถูกอธิบายเพียงครั้งเดียว จากนั้นจึงอัปเดตแยกกันด้วยโคดผู้ใช้ ณ รันไทม์ เพื่อตอบสนองต่อเหตุการณ์ ความท้าทายประการหนึ่งของแนวทางนี้คือ เมื่อแอปพลิเคชันมีความซับซ้อนมากขึ้น นักพัฒนาจำเป็นต้องทราบว่าสถานะเปลี่ยนแปลงไปอย่างไรตลอดทั้ง UI ตัวอย่างเช่น พิจารณา UI ต่อไปนี้:



รูปที่ 2.21 หน้าต่างเลือกสี

มีหลายที่ที่สามารถเปลี่ยนสถานะได้ กล่องสี, แถบเลื่อนเฉดสี, ปุ่มตัวเลือก เมื่อผู้ใช้โต้ตอบกับ UI การเปลี่ยนแปลงจะต้องสะท้อนให้เห็นในทุกที่ ที่เลวร้ายยิ่งกว่านั้น เว้นแต่จะได้รับการดูแล การเปลี่ยนแปลงเล็กน้อยในส่วนใดส่วนหนึ่งของอินเทอร์เฟซผู้ใช้อาจทำให้เกิดเอฟเฟกต์คลื่นส่งผลกระทบกับโค้ดที่ดูเหมือนจะไม่เกี่ยวข้องกัน

วิธีแก้ปัญหาอย่างหนึ่งคือแนวทางเช่น MVC โดยที่คุณส่งข้อมูลการเปลี่ยนแปลงไปยังโมเดลผ่านคอนโทรลเลอร์ จากนั้นโมเดลจะพุชสถานะใหม่ไปยังมุมมองผ่านคอนโทรลเลอร์ อย่างไรก็ตามสิ่งนี้ก็เป็นปัญหาเช่นกัน เนื่องจากการสร้างและการอัปเดตองค์ประกอบ UI เป็นขั้นตอนสองขั้นตอนที่แยกจากกันซึ่งอาจไม่ซิงค์กันได้อย่างง่ายดาย

Flutter พร้อมด้วยเฟรมเวิร์กเชิงโต้ตอบอื่น ๆ ใช้แนวทางอื่นในการแก้ไขปัญหานี้ โดยแยกอินเทอร์เฟซผู้ใช้ออกจากสถานะพื้นฐานอย่างชัดเจน ด้วย API สไตล์ React คุณจะสร้างเฉพาะคำอธิบาย UI เท่านั้น และเฟรมเวิร์กจะดูแลการใช้การกำหนดค่านั้นเพื่อสร้างหรืออัปเดตอินเทอร์เฟซผู้ใช้ตามความเหมาะสม

ใน Flutter วิดเจ็ต (คล้ายกับส่วนประกอบใน React) จะแสดงด้วยคลาสที่ไม่เปลี่ยนรูปซึ่งใช้ในการกำหนดค่าแผนผังของวัตถุ วิดเจ็ตเหล่านี้ถูกใช้เพื่อจัดการแผนผังวัตถุที่แยกจากกันสำหรับโครงร่าง ซึ่งจากนั้นจะใช้ในการจัดการแผนผังวัตถุที่แยกจากกันสำหรับการประกอบ หัวใจหลักของ Flutter คือชุดของกลไกในการอัพเดทส่วนที่ดัดแปลงของแผนผังวิดเจ็ตอย่างมีประสิทธิภาพ การแปลงแผนผังหลายแผนผังของวัตถุให้เป็นแผนผังระดับล่างของวัตถุ และการแพร่กระจายการเปลี่ยนแปลงไปยังแผนผังวิดเจ็ตเหล่านี้

วิดเจ็ตประกาศส่วนติดต่อผู้ใช้โดยการเขียนทับเมธอด build() ซึ่งเป็นฟังก์ชันที่แปลงสถานะเป็น UI

```
UI = f(state)
```

รูปที่ 2.22 สูตรแสดงการทำงานอย่างคร่าว

เมธอด build() นั้นตามการออกแบบแล้วเป็นเมธอดที่เร็วและควรที่จะไม่มีผลข้างเคียง ทำให้เมธอดนั้นสามารถถูกเรียกใช้โดยเฟรมเวิร์กเมื่อไหร่ก็ได้ที่จำเป็น (เป็นไปได้ที่จะบ่อยมากและมีการเรียกใช้หนึ่งครั้งต่อหนึ่งเฟรม)

วิธีนี้พึ่งพาลักษณะเฉพาะรันไทม์ภาษา (หากเจาะจงคือการสร้างและทำลายวัตถุอย่างรวดเร็ว) ซึ่ง Dart นั้นเหมาะสำหรับงานนี้เป็นพิเศษ

2.8.6 ประวัติ

Flutter เวอร์ชันแรกรู้จักกันในชื่อ “Sky” และทำงานบนระบบปฏิบัติการ Android มีการเปิดเผยในการประชุมสุดยอดนักพัฒนา Dart ประจำปี 2015 โดยมีจุดประสงค์ที่ระบุไว้คือสามารถแสดงผลได้อย่างสม่ำเสมอที่ 120 เฟรมต่อวินาที เมื่อวันที่ 4 ธันวาคม 2018 Flutter 1.0 เปิดตัวในการประชุม Flutter ที่ลอนดอน

ในวันที่ 6 พฤษภาคม 2020 ชุดพัฒนาซอฟต์แวร์ (SDK) Dart เวอร์ชัน 2.8 และ Flutter 1.17.0 ได้รับการเผยแพร่ โดยเพิ่มการรองรับ Metal API

เมื่อวันที่ 3 มีนาคม 2021 Google ได้เปิดตัว Flutter 2 ระหว่างกิจกรรม Flutter Engage ออนไลน์ ได้เพิ่มตัวเรนเดอร์ที่ใช้ Canvas สำหรับเว็บ นอกเหนือจากตัวเรนเดอร์ที่ใช้ HTML และการสนับสนุนแอปพลิเคชันเดสก์ท็อปแบบทดลองสำหรับ Windows, macOS, และ Linux นอกจากนี้ยังมาพร้อมกับ Dart 2.0 ซึ่งรวมถึงการสนับสนุนด้านความปลอดภัยแบบ null (null-safety) ความปลอดภัยแบบ null เป็นทางเลือกในตอนแรกเนื่องจากเป็นการเปลี่ยนแปลงครั้งใหญ่และบังคับใช้ใน Dart 3 ที่เปิดตัวในปี 2023

ในวันที่ 12 พฤษภาคม 2022 Flutter 3 และ Dart 2.17 ได้รับการเผยแพร่โดยมีการรองรับแพลตฟอร์มเดสก์ท็อปทั้งหมดอย่างเสถียร

เมื่อวันที่ 27 ตุลาคม 2024 นักพัฒนาชุมชน Flutter จำนวนหนึ่งได้ประกาศเปิดตัว Flock ซึ่งเป็นเวอร์ชันแยกของ Flutter ที่มีจุดประสงค์เพื่อให้ง่ายต่อการร่วมพัฒนา ในขณะเดียวกันก็ยังคงรักษาความสอดคล้องกับทุกการเปลี่ยนแปลงที่เกิดขึ้นในโค้ดต้นทาง

ในปี 2025 Google ยังคงพัฒนา Flutter ต่อไปด้วยสถาปัตยกรรมแบบโมดูลาร์ที่ได้รับการปรับปรุง การรองรับอุปกรณ์พับได้ และการเพิ่มประสิทธิภาพ ARM IoT ตามที่ระบุไว้ในแผนงานฉบับปรับปรุง

2.9 Git

Git เป็นระบบซอฟต์แวร์ควบคุมเวอร์ชันแบบกระจาย ที่สามารถจัดการเวอร์ชันของซอร์สโคดหรือข้อมูลได้ มักใช้เพื่อควบคุมซอร์สโคดโดยโปรแกรมเมอร์ที่พัฒนาซอฟต์แวร์ร่วมกัน

2.9.1 Gitea

Gitea เป็นชุดซอฟต์แวร์ forge สำหรับการโฮสต์ระบบควบคุมเวอร์ชันการพัฒนาซอฟต์แวร์ โดยใช้ Git รวมถึงฟีเจอร์การทำงานร่วมกันอื่น ๆ เช่น การติดตามบั๊ก การตรวจสอบโคด การผสานรวมอย่างต่อเนื่อง (Continuous Integration; CI) กระดาน Kanban ระบบรายงานปัญหา และวิกิ รองรับการโฮสต์ด้วยตนเอง และยังมีอินสแตนซ์สาธารณะของบุคคลที่หนึ่งให้ใช้งานฟรีอีกด้วย Gitea เป็นส่วนหนึ่งของ Gogs และเขียนด้วยภาษา Go Gitea สามารถโฮสต์ได้บนทุกแพลตฟอร์มที่รองรับ Go รวมถึง FreeBSD, Linux, macOS และ Windows โครงการนี้ได้รับทุนสนับสนุนจาก Open Collective

2.10 ภาษาซี (C Programming Language)

ภาษาซีเป็นภาษาโปรแกรมสำหรับวัตถุประสงค์ทั่วไปสร้างขึ้นในช่วงทศวรรษ 1970 โดยเดนนิสริตชีและยังคงได้รับความนิยมและใช้งานอย่างกว้างขวางด้วยการออกแบบภาษาซีทำให้โปรแกรมเมอร์สามารถเข้าถึงคุณลักษณะต่างๆของสถาปัตยกรรมซีพียูทั่วไปได้โดยตรง ซึ่งปรับแต่งให้เหมาะกับชุดคำสั่ง เป้าหมาย ภาษาซี ถูกนำมาใช้และยังคงนำมาใช้ในการพัฒนาระบบปฏิบัติการไดรเวอร์อุปกรณ์และสแต็กโปรโตคอลแต่การใช้งานในซอฟต์แวร์แอปพลิเคชั่นกำลังลดลงภาษาซีถูกนำมาใช้ในคอมพิวเตอร์ตั้งแต่ซูเปอร์คอมพิวเตอร์ขนาดใหญ่ที่สุดไปจนถึงไมโครคอนโทรลเลอร์ขนาดเล็กที่สุดและระบบฝังตัว

ภาษาซีเป็นภาษาเชิงกระบวนการที่จำเป็นรองรับการเขียนโปรแกรมแบบมีโครงสร้างขอบเขตตัวแปรเชิงศัพท์และการเรียกซ้ำด้วยระบบชนิดข้อมูลแบบคงที่ภาษาซีถูกออกแบบมาเพื่อการคอมไพล์เพื่อให้สามารถเข้าถึงหน่วยความจำ และโครงสร้างภาษา ในระดับต่ำซึ่งแมปกับคำสั่งเครื่องได้อย่างมีประสิทธิภาพ โดยทั้งหมดนี้รองรับรันไทม์ขั้นต่ำ แม้จะมีความสามารถในระดับต่ำ แต่ภาษาซีก็ถูกออกแบบมาเพื่อสนับสนุนการเขียนโปรแกรมข้ามแพลตฟอร์ม โปรแกรมซี ที่สอดคล้องกับมาตรฐานที่เขียนขึ้นโดยคำนึงถึงความสามารถในการพกพาสามารถคอมไพล์สำหรับแพลตฟอร์มคอมพิวเตอร์และระบบปฏิบัติการที่หลากหลาย โดยมีการเปลี่ยนแปลงซอร์สโคดเพียงเล็กน้อย

แม้ว่าทั้งภาษาซีและไลบรารีมาตรฐานของภาษา ซีจะไม่ได้มีคุณสมบัติยอดนิยมบางอย่างที่พบในภาษาอื่น แต่ก็มีความยืดหยุ่นเพียงพอที่จะรองรับคุณสมบัติเหล่านั้นได้ ตัวอย่างเช่นการวางแนววัตถุและการเก็บขยะนั้นจัดทำโดยไลบรารีภายนอก GLib Object System และ Boehm garbage collector ตามลำดับ

ตั้งแต่ปี 2000 เป็นต้นมาภาษาซี ได้รับการจัดอันดับอย่างต่อเนื่องให้อยู่ในอันดับสี่ภาษาสูงสุดในดัชนี TIOBE ซึ่งเป็นการวัดความนิยมของภาษาการเขียนโปรแกรม

2.10.1 ประวัติ

2.10.1.1 การพัฒนาช่วงแรก

ที่มาของภาษา C มีความเชื่อมโยงอย่างใกล้ชิดกับการพัฒนาระบบปฏิบัติการ Unix ซึ่งเดิมทีเขียนด้วยภาษาแอสเซมบลีบน PDP-7 โดย Dennis Ritchie และ Ken Thompson โดยนำแนวคิดหลายอย่างจากเพื่อนร่วมงานมาใช้ ในที่สุดพวกเขาก็ตัดสินใจย้ายระบบปฏิบัติการไปยัง PDP-11 เวอร์ชัน Unix ดั้งเดิมบน PDP-11 ก็ได้รับการพัฒนาด้วยภาษาแอสเซมบลีเช่นกัน

2.10.1.2 ภาษา B

ทอมป์สันต้องการภาษาโปรแกรมสำหรับการพัฒนายูทิลิตี้สำหรับแพลตฟอร์มใหม่ เขาพยายามเขียนคอมไพเลอร์ Fortran ก่อน แต่ในไม่ช้าเขาก็ล้มเลิกความคิดนั้นและสร้างเวอร์ชันย่อของภาษาโปรแกรมระบบ ที่พัฒนาขึ้นใหม่ชื่อ BCPL แทน คำอธิบายอย่างเป็นทางการของ BCPL ยังไม่พร้อมใช้งานในขณะนั้นและทอมป์สันได้แก้ไขไวยากรณ์ให้ “กระชับ” น้อยลงและคล้ายกับ ALGOL ที่เรียบง่ายกว่า ที่เรียกว่า SMALGOL เขาเรียกผลลัพธ์ นี้ว่า B โดยอธิบายว่าเป็น “ความหมายของ BCPL ที่มีไวยากรณ์ SMALGOL จำนวนมาก” เช่นเดียวกับ BCPL, B มีคอมไพเลอร์บูตสแตรปเพื่ออำนวยความสะดวกในการพอร์ตไปยังเครื่องใหม่ในที่สุด มีการเขียนยูทิลิตี้เพียงไม่กี่ตัวใน B เพราะมันช้าเกินไปและไม่สามารถใช้ประโยชน์จากคุณสมบัติของ PDP-11 เช่นการเข้าถึงที่อยู่ไบต์ได้

2.10.1.3 ภาษา B ใหม่และ C รุ่นแรก

ในปี พ.ศ. 2514 ริชชีเริ่มปรับปรุง B เพื่อใช้คุณสมบัติของ PDP-11 ที่ทรงพลังยิ่งขึ้น การเพิ่มเติมที่สำคัญคือประเภทข้อมูลอักขระ เขาเรียกสิ่งนี้ว่า New B (NB) ทอมป์สันเริ่มใช้ NB เพื่อเขียน เคอร์เนล Unix และข้อกำหนดของเขากำหนดทิศทางการพัฒนาภาษา

จนถึงปี 1972 มีการเพิ่มประเภทข้อมูลที่หลากหลายมากขึ้นให้กับภาษา NB ภาษา NB มีอาร์เรย์ของ int และ char และได้มีการเพิ่มพอยเตอร์ ความสามารถในการสร้างพอยเตอร์ไปยังประเภทอื่นๆ อาร์เรย์ของทุกประเภท และประเภทที่จะส่งคืนจากฟังก์ชัน อาร์เรย์ภายในนิพจน์ได้รับการปฏิบัติเสมือนเป็นพอยเตอร์ มีการเขียนคอมไพเลอร์ใหม่ และเปลี่ยนชื่อภาษาเป็น C

คอมไพเลอร์ C และยูทิลิตี้บางส่วนที่สร้างขึ้นด้วยคอมไพเลอร์นี้ถูกรวมอยู่ใน Unix เวอร์ชัน 2 ซึ่งเรียกอีกอย่างว่า Research Unix

2.10.1.4 โครงสร้างและการเขียน Unix kernel ใหม่

ใน Unix เวอร์ชัน 4 ซึ่งวางจำหน่ายในเดือนพฤศจิกายน พ.ศ. 2516 เคอร์เนลของ Unix ได้รับการเขียนใหม่อย่างกว้างขวางด้วยภาษา C ในเวลานั้น ภาษา C ได้รับคุณสมบัติที่ทรงพลังบางอย่างเช่นประเภท struct

ตัวประมวลผลล่วงหน้าได้รับการแนะนำประมาณปี 1973 ตามคำแนะนำของ Alan Snyder และยังเป็นการยอมรับถึงประโยชน์ของกลไกการรวมไฟล์ที่มีอยู่ใน BCPL และPL/I เวอร์ชันดั้งเดิมให้เฉพาะไฟล์ที่รวมไว้และการแทนที่สตริงแบบง่ายเท่านั้น #include รวม #define ถึงมาโครที่ไม่มีพารามิเตอร์ หลังจากนั้นไม่นาน ก็มีการขยายเพิ่มเติม โดยส่วนใหญ่โดย Mike Lesk และต่อมาโดย John Reiser เพื่อรวมมาโครที่มีอาร์กิวเมนต์และการคอมไพล์แบบมีเงื่อนไข

Unix เป็นหนึ่งในเคอร์เนลระบบปฏิบัติการแรกๆ ที่เขียนด้วยภาษาอื่นที่ไม่ใช่ภาษาแอสเซมบลีตัวอย่างก่อนหน้านี้ได้แก่ ระบบ Multics (ซึ่งเขียนด้วยภาษา PL/I) และ Master Control Program (MCP) สำหรับ Burroughs B5000 (ซึ่งเขียนด้วยภาษา ALGOL) ในปี 1961 ในช่วงปี 1977 Ritchie และ Stephen C. Johnson ได้ทำการเปลี่ยนแปลงเพิ่มเติมให้กับภาษาเพื่ออำนวยความสะดวกในการพกพาระบบปฏิบัติการUnix คอมไพเลอร์ Portable C ของ Johnson เป็นพื้นฐานสำหรับการใช้งาน C บนแพลตฟอร์มใหม่ๆ หลายแพลตฟอร์ม

2.10.1.5 K&R C

ในปี พ.ศ. 2521 Brian KernighanและDennis Ritchie ได้ตีพิมพ์หนังสือ The C Programming Language ฉบับพิมพ์ครั้งแรกหนังสือเล่มนี้รู้จักกันในชื่อย่อ K&R ตามชื่อย่อของผู้เขียนและทำหน้าที่เป็นข้อกำหนดที่ไม่เป็นทางการ ของภาษาเป็นเวลาหลายปีเวอร์ชันของภาษา C ที่อธิบายไว้ในหนังสือเล่มนี้มักเรียกกันว่า “K&R C” เนื่องจาก หนังสือเล่มนี้ได้รับการเผยแพร่ในปี พ.ศ. 2521 จึงเรียกอีกอย่างว่า C78 หนังสือฉบับพิมพ์ครั้งที่สองครอบคลุมมาตรฐาน ANSI C ในภายหลัง ซึ่งจะกล่าวถึงต่อไป

K&R ได้เพิ่มฟีเจอร์ด้านภาษาหลายอย่าง

1) ไลบรารีอินพุต/เอาต์พุตมาตรฐาน
2) long int ประเภทข้อมูล
3) unsigned int ประเภทข้อมูล
4) ตัวดำเนินการกำหนดค่าแบบผสมในรูปแบบ *=op* (เช่น =-) ถูกเปลี่ยนเป็นรูปแบบ *op*= (นั่นคือ -=) เพื่อขจัดความกำกวมทางความหมายที่เกิดจากโครงสร้างเช่น i=-10 ซึ่งถูกตีความว่า i =- 10 (ลด i ลง 10) แทนที่จะเป็นความหมายที่ตั้งใจไว้ (ให้ i เป็น −10)

แม้หลังจากมีการเผยแพร่มาตรฐาน ANSI ปี 1989 แล้วก็ตาม เป็นเวลาหลายปีที่ K&R C ยังคงถูกพิจารณาว่าเป็น “ตัวหารร่วมที่ต่ำที่สุด” ที่โปรแกรมเมอร์ภาษา C ยึดถือเมื่อต้องการความสามารถในการพกพาได้สูงสุด เนื่องจากคอมไพเลอร์รุ่นเก่าจำนวนมากยังคงถูกใช้งานอยู่และเนื่องจากโค้ด K&R C ที่เขียนอย่างระมัดระวังก็สามารถเป็นไปตามมาตรฐาน C ได้เช่นกัน

แม้ว่า C เวอร์ชันต่อมาจะกำหนดให้ฟังก์ชันต้องมีการประกาศประเภทอย่างชัดเจนแต่ C เวอร์ชัน K&R กำหนดให้ฟังก์ชันที่ส่งคืนค่าประเภทอื่นที่ไม่ใช่ประเภท ที่กำหนดไว้เท่านั้น int ที่จะต้องประกาศก่อนใช้งาน ฟังก์ชันที่ใช้โดยไม่มีการประกาศล่วงหน้าจะถือว่าส่งคืนค่าประเภทที่กำหนด int ไว้

2.10.1.6 ANSI C และ ISO C

ในช่วงปลายทศวรรษ 1970 และ 1980 ภาษา C เวอร์ชันต่างๆ ถูกนำไปใช้งานใน คอมพิวเตอร์เมนเฟรมมินิคอมพิวเตอร์และไมโครคอมพิวเตอร์หลากหลายรุ่นรวมถึง IBM PC ด้วยเนื่องจากความนิยมของคอมพิวเตอร์ประเภทนี้เพิ่มขึ้นอย่างมาก

ในปี 1983 สถาบันมาตรฐานแห่งชาติอเมริกัน (ANSI) ได้จัดตั้งคณะกรรมการ X3J11 เพื่อกำหนดมาตรฐานของภาษา C X3J11 ใช้มาตรฐาน C ที่อิงตามการใช้งานบนระบบ Unix เป็นพื้นฐาน อย่างไรก็ตามส่วนที่ไม่สามารถพกพาได้ของไลบรารี C บน Unix ได้ถูกส่งต่อไปยังกลุ่มทำงาน IEEE 1003 เพื่อใช้เป็นพื้นฐานสำหรับ มาตรฐาน POSIX ในปี 1988 ในปี 1989 มาตรฐาน C ได้รับการรับรองเป็น ANSI X3.159-1989 “ภาษาโปรแกรม C” เวอร์ชันนี้ของภาษามักถูกเรียกว่า ANSI C, Standard C หรือบางครั้งเรียกว่า C89

ในปี 1990 มาตรฐาน ANSI C (พร้อมการเปลี่ยนแปลงรูปแบบ) ได้รับการรับรองโดยองค์การมาตรฐานสากล (ISO) ในชื่อ ISO/IEC 9899:1990 ซึ่งบางครั้งเรียกว่า C90 ดังนั้น คำว่า “C89” และ “C90” จึงหมายถึงภาษาโปรแกรมเดียวกัน

เช่นเดียวกับองค์กรมาตรฐานแห่งชาติอื่นๆ ANSI ไม่ได้พัฒนามาตรฐาน C ด้วยตนเองอีกต่อไป แต่จะอ้างอิงถึงมาตรฐาน C สากล ซึ่งดูแลโดยคณะทำงาน ISO/IEC JTC1/SC22 / WG14 การนำมาตรฐานสากลฉบับปรับปรุงมาใช้ในระดับประเทศมักเกิดขึ้นภายในหนึ่งปีหลังจากที่ ISO เผยแพร่มาตรฐานดังกล่าว

หนึ่งในเป้าหมายของกระบวนการกำหนดมาตรฐานภาษา C คือการสร้างซูเปอร์เซ็ตของ K&R C โดยรวมเอาคุณสมบัติที่ไม่เป็นทางการหลายอย่างที่ถูกนำมาใช้ในภายหลัง คณะกรรมการมาตรฐานยังได้เพิ่มคุณสมบัติเพิ่มเติมอีกหลายอย่างเช่นต้นแบบฟังก์ชัน (ยืมมาจาก C++), voidพอยเตอร์, การรองรับชุดอักขระและภาษาท้องถิ่นระหว่างประเทศ และการปรับปรุงพรีโปรเซสเซอร์ แม้ว่าไวยากรณ์สำหรับการประกาศพารามิเตอร์จะได้รับการปรับปรุงให้รวมรูปแบบที่ใช้ใน C++ แต่ก็ยังคงอนุญาตให้ใช้อินเทอร์เฟซ K&R เพื่อความเข้ากันได้กับซอร์สโค้ดที่มีอยู่

C89 ได้รับการสนับสนุนจากคอมไพเลอร์ C ในปัจจุบัน และโค้ด C สมัยใหม่ส่วนใหญ่ก็ใช้ C89 เป็นพื้นฐานโปรแกรมใดๆ ที่เขียนด้วยภาษา C มาตรฐานเท่านั้น และไม่มีข้อสมมติฐานใดๆ ที่ขึ้นอยู่กับฮาร์ดแวร์ จะทำงานได้อย่างถูกต้องบนแพลตฟอร์มใดๆ ที่มีการใช้งาน C ที่สอดคล้องกับมาตรฐาน ภายในขีดจำกัดของทรัพยากร หากไม่ระมัดระวัง โปรแกรมอาจคอมไพล์ได้เฉพาะบนแพลตฟอร์มใดแพลตฟอร์มหนึ่ง หรือด้วยคอมไพเลอร์เฉพาะเท่านั้น ตัวอย่างเช่น เนื่องจากการใช้ไลบรารีที่ไม่เป็นมาตรฐาน เช่น ไลบรารี GUI หรือการพึ่งพาคุณลักษณะเฉพาะของคอมไพเลอร์หรือแพลตฟอร์ม เช่น ขนาดที่แน่นอนของชนิดข้อมูลและลำดับไบต์

ในกรณีที่โค้ดต้องสามารถคอมไพล์ได้ทั้งโดยคอมไพเลอร์ที่สอดคล้องกับมาตรฐานหรือคอมไพเลอร์ที่ใช้ C แบบ K&R นั้น __STDC__ สามารถใช้มาโครเพื่อแบ่งโค้ดออกเป็นส่วนมาตรฐานและส่วน K&R เพื่อป้องกันการใช้คุณสมบัติที่มีเฉพาะใน C มาตรฐานบนคอมไพเลอร์ที่ใช้ C แบบ K&R

หลังจากกระบวนการกำหนดมาตรฐาน ANSI/ISO ข้อกำหนดภาษา C ยังคงค่อนข้างคงที่เป็นเวลาหลายปี ในปี 1995 มีการเผยแพร่การแก้ไขมาตรฐานฉบับที่ 1 ของมาตรฐาน C ปี 1990 (ISO/IEC 9899/AMD1:1995 ซึ่งเรียกกันอย่างไม่เป็นทางการว่า C95) เพื่อแก้ไขรายละเอียดบางประการและเพิ่มการสนับสนุนชุดอักขระสากลที่ครอบคลุมมากขึ้น

2.10.1.7 C99

มาตรฐาน C ได้รับการแก้ไขเพิ่มเติมในช่วงปลายทศวรรษ 1990 ส่งผลให้มีการตีพิมพ์ ISO/IEC 9899:1999 ในปี 1999 ซึ่งโดยทั่วไปเรียกว่า “C99” ต่อมาได้มีการแก้ไขเพิ่มเติมอีกสามครั้งโดย Technical Corrigenda

C99 ได้นำเสนอคุณสมบัติใหม่หลายประการรวมถึงฟังก์ชันอินไลน์ชนิดข้อมูลใหม่หลายชนิด(รวมถึง long long intชนิด ข้อมูล complex ที่ใช้แทนจำนวนเชิงซ้อน) อาร์เรย์ที่มีความยาวแปรผันได้และสมาชิกอาร์เรย์ที่ยืดหยุ่นการสนับสนุนที่ดีขึ้นสำหรับเลขทศลอย IEEE 754 การสนับสนุนมาโครแบบแปรผัน (มาโครที่มีจำนวนอาร์กิวเมนต์ แปรผันได้) และการสนับสนุนความคิดเห็นแบบบรรทัดเดียวที่ขึ้นต้นด้วย @ // เช่นเดียวกับใน BCPL หรือ C++ คุณสมบัติเหล่านี้หลายอย่างได้ถูกนำไปใช้เป็นส่วนขยายในคอมไพเลอร์ C หลายตัวแล้ว

โดยส่วนใหญ่แล้ว C99 สามารถใช้งานร่วมกับ C90 ได้ แต่มีความเข้มงวดมากกว่าในบางด้าน โดยเฉพาะอย่างยิ่ง การประกาศที่ไม่มีตัวระบุประเภทจะไม่ถือว่ามีintการกำหนดโดยปริยาย

อีกต่อไป มีการกำหนดมาโครมาตรฐาน __STDC_VERSION__ พร้อมค่า 199901L เพื่อระบุว่ามีการสนับสนุน C99 คอม ไพเลอร์ C อื่นๆ เช่น GCC, Solaris Studio และคอมไพเลอร์ C อื่นๆ ในปัจจุบันรองรับคุณสมบัติใหม่หลายอย่างหรือทั้งหมดของ C99 อย่างไรก็ตาม คอมไพเลอร์ C ใน Microsoft Visual C++ ใช้มาตรฐาน C89 และส่วนต่างๆ ของ C99 ที่จำเป็นสำหรับการใช้งานร่วมกับ C++11

นอกจากนี้ มาตรฐาน C99 ยังกำหนดให้รองรับตัวระบุที่ใช้ Unicode ในรูปแบบของอักขระพิเศษ (เช่น \u0040 หรือ \U0001f431) และแนะนำให้รองรับชื่อ Unicode ดิบด้วย

2.10.1.8 C11

งานปรับปรุงมาตรฐาน C ฉบับใหม่เริ่มขึ้นในปี 2550 โดยเรียกกันอย่างไม่เป็นทางการว่า “C1X” จนกระทั่งมีการประกาศใช้มาตรฐาน ISO/IEC 9899:2011 อย่างเป็นทางการในวันที่ 8 ธันวาคม 2554 คณะกรรมการมาตรฐาน C ได้กำหนดแนวทางเพื่อจำกัดการนำคุณสมบัติใหม่ๆ ที่ยังไม่ได้รับการทดสอบโดยระบบที่มีอยู่มาใช้

มาตรฐาน C11 เพิ่มคุณสมบัติใหม่มากมายให้กับภาษา C และไลบรารี รวมถึงมาโครแบบเจเนริกชนิดโครงสร้างนิรนามการสนับสนุน Unicode ที่ดีขึ้น การดำเนินการอะตอมิกการทำงานแบบมัลติเธรดและฟังก์ชันตรวจสอบขอบเขต นอกจากนี้ยังทำให้บางส่วนของไลบรารี C99 ที่มีอยู่เป็นตัวเลือก และปรับปรุงความเข้ากันได้กับ C++ มาโครมาตรฐาน __STDC_VERSION__ ถูกกำหนดไว้เพื่อ 201112L ระบุว่ามีการสนับสนุน C11 แล้ว

2.10.1.9 C17

C17 เป็นชื่อเรียกอย่างไม่เป็นทางการของ ISO/IEC 9899:2018 ซึ่งเป็นมาตรฐานสำหรับภาษาโปรแกรม C ที่เผยแพร่ในเดือนมิถุนายน 2018 มาตรฐานนี้ไม่ได้เพิ่มคุณสมบัติใหม่ใดๆ ให้กับภาษา แต่เป็นการแก้ไขทางเทคนิคและการชี้แจงข้อบกพร่องใน C11 เท่านั้น มาโครมาตรฐาน __STDC_VERSION__ ถูกกำหนดขึ้นเพื่อ 201710L ระบุว่ามีการรองรับ C17 แล้ว

2.10.1.10 C23

C23 เป็นชื่อเรียกอย่างไม่เป็นทางการของการแก้ไขมาตรฐานภาษา C หลักในปัจจุบัน ซึ่งในระหว่างการพัฒนาส่วนใหญ่เรียกว่า “C2X” โดยสร้างขึ้นจากเวอร์ชันก่อนหน้า และแนะนำคุณสมบัติใหม่ เช่น คำหลักใหม่ ความหมายเพิ่มเติมสำหรับเพื่อ auto ให้มีการอนุมานประเภทเมื่อประกาศตัวแปรประเภทใหม่รวมถึง nullptr_t และ _BitInt (N) และการขยายไลบรารีมาตรฐาน C23 ได้รับการเผยแพร่ในเดือนตุลาคม 2024 ในชื่อ ISO/IEC 9899:2024 มาโครมาตรฐาน __STDC_VERSION__ ถูกกำหนดไว้ 202311L เพื่อระบุว่ามีการสนับสนุน C23

2.10.1.11 C2Y

C2Y เป็นชื่อเรียกอย่างไม่เป็นทางการของการแก้ไขมาตรฐานภาษา C ครั้งใหญ่ถัดไป หลังจาก C23 (C2X) ซึ่งคาดว่าจะออกในช่วงปลายทศวรรษ 2020 ดังนั้นจึงมีเลข ‘2’ ใน “C2Y” ร่างฉบับแรกของ C2Y ได้รับการเผยแพร่ในเดือนกุมภาพันธ์ 2024 ในชื่อ N3220 โดยกลุ่มทำงาน ISO/IEC JTC1/SC22 /WG14

2.10.1.12 Embedded C

ในอดีตการเขียนโปรแกรม C สำหรับระบบฝังตัวจำเป็นต้องใช้ส่วนขยายที่ไม่เป็นมาตรฐานของภาษา C เพื่อรองรับคุณสมบัติพิเศษ เช่นการคำนวณเลขทศนิยมคงที่ ธนาคารหน่วยความจำหลายชุดที่แตกต่างกันและการดำเนินการอินพุต/เอาต์พุตพื้นฐาน

ในปี 2551 คณะกรรมการมาตรฐาน C ได้เผยแพร่รายงานทางเทคนิคที่ขยายภาษา C เพื่อแก้ไขปัญหาเหล่านี้โดยการจัดหามาตรฐานทั่วไปสำหรับการใช้งานทั้งหมดให้ปฏิบัติตาม ซึ่งรวมถึงคุณสมบัติหลายอย่างที่ไม่มีในภาษา C ปกติ เช่น การคำนวณเลขทศนิยมคงที่ พื้นที่แอดเดรสแบบมีชื่อและการกำหนดแอดเดรสฮาร์ดแวร์ I/O พื้นฐาน

2.10.2 ตัวแปร (Variables)

ตัวแปรในภาษา C เบื้องต้นแล้วประกอบไปด้วยประเภทของข้อมูลและชื่อตัวแปร โดยที่ชื่อตัวแปรนั้นสามารถเป็นรายการที่ถูกแบ่งด้วยเครื่องหมายจุลภาคได้ด้วยเช่นกัน ตัวอย่างคือ

```
int data;
float a, b, c;
```

รูปที่ 2.23 ตัวอย่างการประกาศตัวแปรในภาษา C

2.10.3 ประเภทข้อมูล (Data Types)

ข้อมูลที่เกี่ยวข้องกับตัวเลขมักมีประเภท unsigned และ signed โดยความแตกต่างหากอธิบายสั้น ๆ คือ

1) Signed (มีเครื่องหมาย): ตัวเลขที่สามารถติดลบได้ ระยะข้อมูลตัวอย่างคือ −128 ถึง 127

2) Unsigned (ไม่มีเครื่องหมาย): ตัวเลขที่ไม่สามารถติดลบได้ ระยะข้อมูลตัวอย่างคือ 0 ถึง 255

จะสังเกตได้ว่า ข้อมูลประเภท unsigned นั้นสามารถเก็บตัวเลขบวกได้จำนวนมากกว่า คือสูงสุดที่ 255 แต่หากนำค่าสัมบูรณ์ (absolute value) ของระยะข้อมูลแบบ signed มาบวกกัน เช่น |−128| + |127| จะพบว่าได้ค่า 255 หมายความว่า จริง ๆ แล้วข้อมูลประเภท signed สามารถเก็บข้อมูลได้ 255 ตัวเลขเช่นกัน เพียงแต่ว่าครึ่งหนึ่งของตัวเลขที่สามารถเก็บได้เป็นตัวเลขติดลบ

ดังนั้นโปรดจำไว้ว่า เลขคณิตจำนวนเต็มมีนิยามแตกต่างกันสำหรับชนิดจำนวนเต็มแบบ signed และ unsigned โปรดดูตัวดำเนินการเลขคณิต โดยเฉพาะอย่างยิ่งการโอเวอร์โฟลว์จำนวนเต็ม

2.10.3.1 ประเภทบูลีน (Boolean)

ประเภทบูลีนนั้นถูกนำเสนอครั้งแรกในมาตรฐาน C99 โดยการกล่าวถึงประเภทข้อมูลบูลีนนั้น ในประวัติของภาษา C แล้วมีสองแบบ

1) _Bool (และมีมาโคร bool): จนถึงมาตรฐาน C23
2) bool (ที่ไม่ใช่แค่มาโคร): มีตั้งแต่มาตรฐาน C23

2.10.3.2 ประเภทจำนวนเต็ม (Integer)

1) short int (หรืออีกชื่อหนึ่งคือ short และสามารถใช้คีย์เวิร์ด signed ได้)

2) unsigned short int (หรือ unsigned short)

3) int (หรือ signed int)

คือประเภทข้อมูลตัวเลขที่ปกติที่สุด และจะถูกการันตีว่าจะมีขนาดขั้นต่ำ 16 บิตเสมอ โดยระบบทั่วไปส่วนใหญ่ในปัจจุบันจะเป็น 32 บิต

4) unsigned int (หรือเพียงแค่ unsigned) คือประเภท int ในแบบ unsigned, มี modulo arithmetic, และเหมาะสมสำหรับการเปลี่ยนแปลงบิต

5) long int (หรือ long)

6) unsigned long int (หรือ unsigned long)

7) มีเพิ่มตั้งแต่ C99

ก) long long int (หรือ long long)

ข) unsigned long long int (หรือ unsigned long long)

8) มีเพิ่มตั้งแต่ C23

ก) _BitInt(n) (หรือ signed _BitInt(n)): ประเภทข้อมูล signed แบบมีขนาดชัดเจน โดย n แทนด้วยจำนวนบิต (รวมถึงบิตเครื่องหมาย และ n จะต้องไม่มากกว่า BITINT_MAXWIDTH จากไฟล์ <limits.h>)

ข) unsigned _BitInt(n): เหมือนข้างต้น เพียงแค่เป็นประเภท unsigned (และไม่มีบิตเครื่องหมาย)

และเหมือนประเภทข้อมูลอื่น ๆ คุณสามารถเรียงคีย์เวิร์ดแบบใดก็ได้ เช่น unsigned long long int และ long int unsigned long นั้นเหมือนกัน

ตารางต่อไปนี้สรุปประเภทตัวเลขทั้งหมดและคุณสมบัติของมัน

ตารางที่ 2.3 ขนาดของข้อมูลเป็นบิต

<table>
<tr><th rowspan="2">ชื่อประเภท</th><th rowspan="2">ประเภทเทียบเท่า</th><th colspan="5">จำนวนบิตตามรูปแบบข้อมูล</th></tr>
<tr><th>มาตรฐาน C</th><th>LP32</th><th>ILP32</th><th>LLP64</th><th>LP64</th></tr>
<tr><td>char</td><td>char</td><td rowspan="3">อย่างน้อย 8</td><td rowspan="3">8</td><td rowspan="3">8</td><td rowspan="3">8</td><td rowspan="3">8</td></tr>
<tr><td>signed char</td><td>signed char</td></tr>
<tr><td>unsigned char</td><td>unsigned char</td></tr>
<tr><td>short</td><td rowspan="4">short int</td><td rowspan="6">อย่างน้อย 16</td><td rowspan="6">16</td><td rowspan="6">16</td><td rowspan="6">16</td><td rowspan="6">16</td></tr>
<tr><td>short int</td></tr>
<tr><td>signed short</td></tr>
<tr><td>signed short int</td></tr>
<tr><td>unsigned short</td><td rowspan="2">unsigned short int</td></tr>
<tr><td>unsigned short int</td></tr>
</table>

ตารางที่ 2.4 ขนาดของข้อมูลเป็นบิต (ต่อ)

ชื่อประเภท	ประเภทเทียบเท่า	จำนวนบิตตามรูปแบบข้อมูล				
		มาตรฐาน C	LP32	ILP32	LLP64	LP64
int	int	อย่างน้อย 16	16	32	32	32
signed						
signed int						
unsigned	unsigned int					
unsigned int						
long	long int	อย่างน้อย 32	32	32	32	64
long int						
signed long						
signed long int						
unsigned long	unsigned long int					
unsigned long int						
long long	long long int (C99)	อย่างน้อย 64	64	64	64	64
long long int						
signed long long						
signed long long int						
unsigned long long	unsigned long long int (C99)					
unsigned long long int						

1) รูปแบบข้อมูล (data model)

รูปแบบข้อมูล หรือ data model คือรูปแบบการเก็บข้อมูลของโปรแกรมซึ่งเป็นสิ่งที่กำหนดขนาดของตัวแปร โดยรูปแบบข้อมูลนั้นจะถูกกำหนดโดยแพลตฟอร์มเป้าหมาย ซึ่งมีหน่วยประมวลผลและระบบปฏิบัติการเป็นปัจจัยหลัก โดยตามตารางในหัวข้อก่อนหน้าหลัก ๆ แล้วมีรูปแบบข้อมูลอยู่ 4 รูปแบบ คือ LP32, ILP32, LLP64, และ LP64 ซึ่งหากต้องการหาความหาย L หมายถึง Long, P หมายถึง Pointer, และ I หมายถึง Integer (จำนวนเต็ม) แล้วตามด้วยเลขบิต

2.10.3.3 ประเภทจำนวนทศนิยมจริง (Real floating types)

ภาษา C นั้นมีประเภทข้อมูลสำหรับแทนตัวเลขทศนิยมจริง 3 (หรือ 6 ตั้งแต่ C23) ประเภท

1) float จำนวนทศนิยมความแม่นยำเดี่ยว ตรงกับฟอร์แมตมาตรฐาน IEEE-754 binary32 หากรองรับ

2) double จำนวนทศนิยมความแม่นยำสองเท่า ตรงกับฟอร์แมตมาตรฐาน IEEE-754 binary64 หากรองรับ

3) long double จำนวนทศนิยมความแม่นยำเพิ่มเติม ตรงกับฟอร์แมตมาตรฐาน IEEE-754 binary128 หากรองรับ มิฉะนั้นจะตรงกับ IEEE-754 binary64-extended หากรองรับมิฉะนั้นจะตรงกับรูปแบบจำนวนทศนิยมที่ไม่ตรงกับมาตรฐาน IEEE-754 รูปแบบใดก็ได้ตราบใดที่มี

ความแม่นยำกว่า binary64 และระยะข้อมูลนั้นอย่างน้อยก็ต้องดีเท่า binary64 และหากไม่รองรับทั้งหมดนั้น จะตรงกับฟอร์แมตมาตรฐาน IEEE-754 binary64

ก) รูปแบบ binary128 นั้นถูกใช้โดยระบบ HP-UX, SPARC, MIPS, ARM64, และ z/OS บางระบบ

ข) รูปแบบ IEEE-754 binary64-extended ที่รู้จักกันอย่างแพร่หลายที่สุดคือรูปแบบความแม่นยำเพิ่มเติม 80 บิต x87 ซึ่งถูกใช้โดยสถาปัตยกรรม x86 และ x86-64 บางระบบ (การยกเว้นที่ควรพูดถึงคือ MSVC ที่กำหนดให้ long double อยู่ในรูปแบบเดียวกันกับ double, เช่น binary64)

เมื่อใช้มาตรฐาน C ตั้งแต่ C23 เป็นต้นไปและหากแพลตฟอร์มของคุณใช้งานคอนแสตนต์มาโคร __STDC_IEC_60559_DFP__ ข้อมูลประเภทตัวเลขทศนิยมดังต่อไปนี้จะถูกรองรับด้วย:

1) _Decimal32 แทนรูปแบบมาตรฐาน IEEE-754 decimal32

2) _Decimal64 แทนรูปแบบมาตรฐาน IEEE-754 decimal64

3) _Decimal128 แทนรูปแบบมาตรฐาน IEEE-754 decimal128

มิฉะนั้น ประเภทตัวเลขทศนิยมเพิ่มเติมเหล่านี้จะไม่ถูกรองรับ

ข้อมูลประเภททศนิยมอาจรองรับค่าพิเศษเพิ่มเติมได้แก่

1) อนันต์ (Infinity, ทั้งบวกและลบ)

2) ศูนย์ติดลบ, -0.0 โดยมีค่าเท่ากับศูยน์ที่ติดบวก แต่อาจมีความหมายในบางสมการ เช่น 1.0 / 0.0 == INFINITY แต่ 1.0 / -0.0 == -INFINITY

3) ไม่ใช่ตัวเลข (not-a-number; NaN) ซึ่งไม่เท่ากับอะไรเลย รวมถึงตัวมันเอง

ทศนิยมจำนวนจริงสามารถถูกใช้กับตัวดำเนินการทางคณิตศาสตร์ได้ **+ - / *** และฟังก์ชันทางคณิตศาสตร์จาก <math.h> โดยทั้งตัวดำเนินการและฟังก์ชันจากไลบรารีนั้นสามารถก่อให้เกิดการแสดงข้อผิดพลาดของจำนวนทศนิยมได้และจะตั้งค่า errno

2.10.3.4 ประเภทจำนวนทศนิยมซับซ้อน (Complex floating types)

ประเภทข้อมูลจำนวนทศนิยมซับซ้อนนั้นเป็นประเภทที่แทนตัวเลขเชิงซ้อน (complex number) นั้นคือ ตัวเลขที่สามารถถูกเขียนแทนเป็นผลรวมของจำนวนจริงและจำนวนจริงที่คูณด้วยจำนวนจินตภาพ (a + bi) โดยประเภทจำนวนเชิงซ้อนมีอยู่สามประเภท ได้แก่

1) float _Complex (และสามารถใช้ float complex ได้เช่นกันหากนำเข้า <complex.h>)

2) double _Complex (และสามารถใช้ double complex ได้เช่นกันหากนำเข้า <complex.h>)

3) long double _Complex (และสามารถใช้ long double complex ได้เช่นกันหากนำเข้า <complex.h>)

2.10.3.5 ประเภทจำนวนทศนิยมจินตภาพ (Imaginary floating types)

ประเภทข้อมูลจำนวนทศนิยมจินตภาพนั้นเป็นประเภทที่แทนตัวเลขจินตภาพ (imaginary number) นั้นคือ ตัวเลขที่สามารถถูกเขียนแทนเป็นจำนวนจริงที่คูณด้วยจำนวนจินตภาพ bi โดยประเภทจำนวนเชิงซ้อนมีอยู่สามประเภท ได้แก่

1) float _Imaginary (และสามารถใช้ float imaginary ได้เช่นกันหากนำเข้า <complex.h>)

2) double _Imaginary (และสามารถใช้ double imaginary ได้เช่นกันหากนำเข้า <complex.h>)

3) long double _Imaginary (และสามารถใช้ long double imaginary ได้เช่นกันหากนำเข้า <complex.h>)

2.10.3.6 ประเภทตัวอักษร (Character)

1) signed char คือประเภทสำหรับตัวอักษรแบบ signed

2) unsigned char คือประเภทสำหรับตัวอักษรแบบ unsigned

3) char คือประเภทสำหรับตัวอักษรแบบไม่ระบุระยะข้อมูล ซึ่งสามารถเท่ากับ signed char หรือ unsigned char ก็ได้ขึ้นอยู่กับแพลตฟอร์มและคอมไพเลอร์ แต่อย่างไรก็ตาม char นั้นไม่ใช่เพียงแค่มาโครที่ลิงก์ไปยังประเภทอื่น ๆ แต่ char คือประเภทของมันเอง

2.10.3.7 คีย์เวิร์ด

1) bool, true, false, char, int, short, long, signed, unsigned, float, double

2) _Bool, _BitInt, _Complex, _Imaginary, _Decimal32, _Decimal64, _Decimal128

2.10.3.8 ระยะค่าที่เก็บได้

ก่อนมาตรฐาน C23 มาตรฐาน C อนุญาตการแทนตัวเลขแบบใดก็ได้ และระยะขั้นต่ำของตัวเลข N บิตคือ $-(2^{N-1} - 1)$ ถึง $+2^{N-1} - 1$ (เช่น −127 ถึง 127 สำหรับประเภทตัวเลข 8 บิต) ซึ่งตรงกับขอบเขตของส่วนเติมเต็มหนึ่ง (one's complement) หรือการแทนจำนวนมีเครื่องหมาย (sign-and-magnitude)

อย่างไรก็ตาม รูปแบบข้อมูลที่ใช้กันอย่างแพร่หลายทั้งหมด (รวมถึง ILP32, LP32, LP64, และ LLP64) และคอมไพเลอร์ C เกือบทั้งหมดใช้การแทนตัวเลขแบบส่วนเติมเต็มสอง (two's complement) (มีข้อยกเว้นที่ทราบแค่บางคอมไพเลอร์สำหรับระบบ UNISYS) และตั้งแต่มาตรฐาน C23 มันคือการแทนตัวเลขแบบเดียวที่ถูกอนุญาตให้ใช้โดยมาตรฐาน และมีขอบเขตที่แน่นอนระหว่าง -2^{N-1} ถึง $+2^{N-1}$ (เช่น −128 ถึง 127 สำหรับประเภทตัวเลข 8 บิต)

ตารางต่อไปนี้ให้ข้อมูลเกี่ยวกับขอบเขตของประเภทข้อมูลต่าง ๆ (มีการเพิ่มจุลภาคในทศนิยมเพื่อเพิ่มความสะดวกในการอ่าน)

ตารางที่ 2.5 ตารางแสดงขอบเขตประเภทข้อมูล

ประเภท	ขนาด (บิต)	รูปแบบ	ระยะค่า	
			โดยประมาณ	แน่นอน
ตัวอักษร	8	signed		−128 ถึง 127
		unsigned		0 ถึง 255
	16	UTF-16		0 ถึง 65535
	32	UTF-32		0 ถึง 1114111 (0x10ffff)
จำนวนเต็ม	16	signed	± 3.27 · 104	−32768 ถึง 32767
		unsigned	0 ถึง 6.55 · 104	0 ถึง 65535
	32	signed	± 2.14 · 109	−2,147,483,648 ถึง 2,147,483,647
		unsigned	0 ถึง 4.29 · 109	0 ถึง 4,294,967,295
	64	signed	± 9.22 · 1018	−9,223,372,036,854,775,808 ถึง 9,223,372,036,854,775,807
		unsigned	0 ถึง 1.84 · 1019	0 ถึง 18,446,744,073,709,551,615
ทศนิยมไบนารี	32	IEEE-754	1) min subnormal: $\pm 1.401,298,4 \cdot 10^{-45}$ 2) min normal: $\pm 1.175,494,3 \cdot 10^{-38}$ 3) max: $\pm 3.402,823,4 \cdot 10^{38}$	1) min subnormal: ±0x1p-149 2) min normal: ±0x1p-126 3) max: ±0x1.fffffep+127
	64	IEEE-754	1) min subnormal: $\pm 4.940,656,458,412 \cdot 10^{-324}$ 2) min normal: $\pm 2.225,073,858,507,201, 4 \cdot 10^{-308}$ 3) max: $\pm 1.797,693,134,862,315, 7 \cdot 10^{308}$	1) min subnormal: ±0x1p-1074 2) min normal: ±0x1p-1022 3) max: ±0x1 .fffffffffffffp+1023

ตารางที่ 2.6 ตารางแสดงขอบเขตประเภทข้อมูล (ต่อ)

ประเภท	ขนาด (บิต)	รูปแบบ	ระยะค่า	
			โดยประมาณ	แน่นอน
ทศนิยมไบนารี	80	x86	1) min subnormal: $\pm 3.645,199,531,882,474,602,528 \cdot 10^{-4951}$ 2) min normal: $\pm 3.362,103,143,112,093,506,263 \cdot 10^{-4932}$ 3) max: $\pm 1.189,731,495,357,231,765,021 \cdot 10^{4932}$	1) min subnormal: ±0x1p-16445 2) min normal: ±0x1p-16382 3) max: ±0x1.ffffffffffffffffep+16383
	128	IEEE-754	1) min subnormal: $\pm 6.475,175,119,438,025,110,924,438,958,227,646,552,5 \cdot 10^{-4966}$ 2) min normal: $\pm 3.362,103,143,112,093,506,262,677,817,321,752,602,6 \cdot 10^{-4932}$ 3) max: $\pm 1.189,731,495,357,231,765,085,759,326,628,007,016,2 \cdot 10^{4932}$	1) min subnormal: ±0x1p-16494 2) min normal: ±0x1p-16382 3) max: ±0x1.ffffffffffffffffffffffffffffp+16383
ทศนิยมเดซิมอล	32	IEEE-754		1) min subnormal: $\pm 1 \cdot 10^{-101}$ 2) min normal: $\pm 1 \cdot 10^{-95}$ 3) max: $\pm 9.999'999 \cdot 10^{96}$
	64	IEEE-754		1) min subnormal: $\pm 1 \cdot 10^{-398}$ 2) min normal: $\pm 1 \cdot 10^{-383}$ 3) max: $\pm 9.999'999'999'999'999 \cdot 10^{384}$
	128	IEEE-754		1) min subnormal: $\pm 1 \cdot 10^{-6176}$ 2) min normal: $\pm 1 \cdot 10^{-6143}$ 3) max: $\pm 9.999'999'999'999'999'999'999'999'999'999'999 \cdot 10^{6144}$

2.10.4 ชุดแปลโปรแกรมของกนู (GNU Compiler Collection; GCC)

ในกระบวนการการพัฒนาโครงงานนี้ ชุดแปลโปรแกรมของกนูนั้นถูกใช้เป็นหลักเนื่องจากเป็นชุดแปลโปรแกรม (คอมไพเลอร์; Compiler) ที่ใช้เป็นหลักในการพัฒนาโคดที่สร้างบนพื้นฐาน Arduino และบอร์ดต่าง ๆ รวมถึงบอร์ด ESP32

ชุดคอมไพเลอร์ GNU (GNU Compiler Collection; GCC) (เดิมชื่อ GNU C Compiler) คือชุดคอมไพเลอร์จากโครงการ GNU ที่รองรับภาษาโปรแกรม สถาปัตยกรรมฮาร์ดแวร์ และระบบปฏิบัติการต่าง ๆ มูลนิธิซอฟต์แวร์เสรี (FSF) เผยแพร่ GCC ในฐานะซอฟต์แวร์เสรีภายใต้สัญญาอนุญาตสถูกเรียกาธารณะทั่วไปของ GNU (GNU GPL) GCC เป็นองค์ประกอบสำคัญของชุดเครื่องมือ GNU ซึ่งใช้สำหรับโครงการส่วนใหญ่ที่เกี่ยวข้องกับ GNU และเคอร์เนล Linux ด้วยโคดประมาณ 15 ล้านบรรทัด ในปี 2019 GCC จึงเป็นหนึ่งในโปรแกรมฟรีที่ใหญ่ที่สุดเท่าที่เคยมีมา GCC มีบทบาทสำคัญในการเติบโตของซอฟต์แวร์เสรี ทั้งในฐานะเครื่องมือและตัวอย่าง

นอกจากจะเป็นคอมไพเลอร์อย่างเป็นทางการของระบบปฏิบัติการ GNU แล้ว GCC ยังได้รับการยอมรับให้เป็นคอมไพเลอร์มาตรฐานโดยระบบปฏิบัติการคอมพิวเตอร์สมัยใหม่ที่คล้ายกับ Unix อื่นๆ อีกมากมาย รวมถึงระบบปฏิบัติการ Linux ส่วนใหญ่ ระบบปฏิบัติการตระกูล BSD ส่วนใหญ่ก็เปลี่ยนมาใช้ GCC ไม่นานหลังจากเปิดตัว แม้ว่าหลังจากนั้น FreeBSD และ Apple macOS ได้เปลี่ยนมาใช้คอมไพเลอร์ Clang ส่วนใหญ่เป็นเพราะเหตุผลด้านลิขสิทธิ์ GCC ยังสามารถคอมไพเลอร์โคดสำหรับระบบปฏิบัติการ Windows, Android, iOS, Solaris, HP-UX, AIX และ MS-DOS ได้อีกด้วย

GCC ได้รับการพอร์ตไปยังแพลตฟอร์มและสถาปัตยกรรมชุดคำสั่งต่าง ๆ มากกว่าคอมไพเลอร์อื่น ๆ และถูกนำไปใช้งานอย่างกว้างขวางในฐานะเครื่องมือในการพัฒนาซอฟต์แวร์ทั้งแบบฟรีและแบบที่เป็นกรรมสิทธิ์ นอกจากนี้ GCC ยังพร้อมใช้งานสำหรับระบบฝังตัวมากมาย รวมถึงชิปที่ใช้ ARM และ Power ISA

บทที่ 3
วิธีการดำเนินโครงงาน

ในการดำเนินการศึกษาครั้งนี้ คณะผู้จัดทำโครงงานได้ศึกษาข้อมูลเบื้องต้นในการสร้างเครื่องยืนยันตัวตนด้วย NFC และได้ดำเนินการตามขั้นตอนนี้

1) วางแผนการดำเนินงาน
2) การออกแบบ
3) วัสดุอุปกรณ์
4) ขั้นตอนการประกอบ
5) การทดลอง
6) การวิเคราะห์ข้อมูล

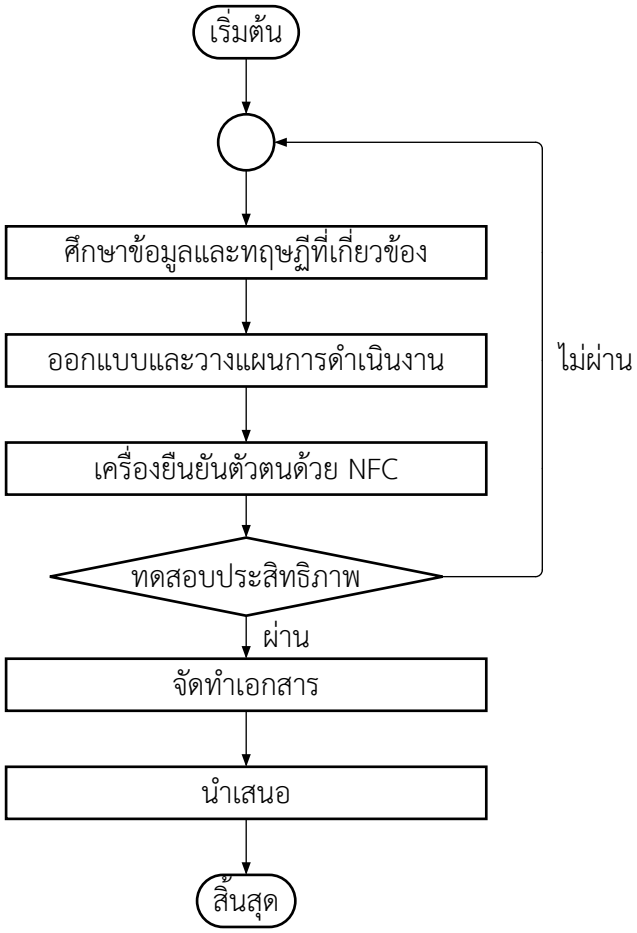
3.1 วางแผนการดำเนินงาน

3.1.1 แผนขั้นตอนและวิธีการดำเนินงาน

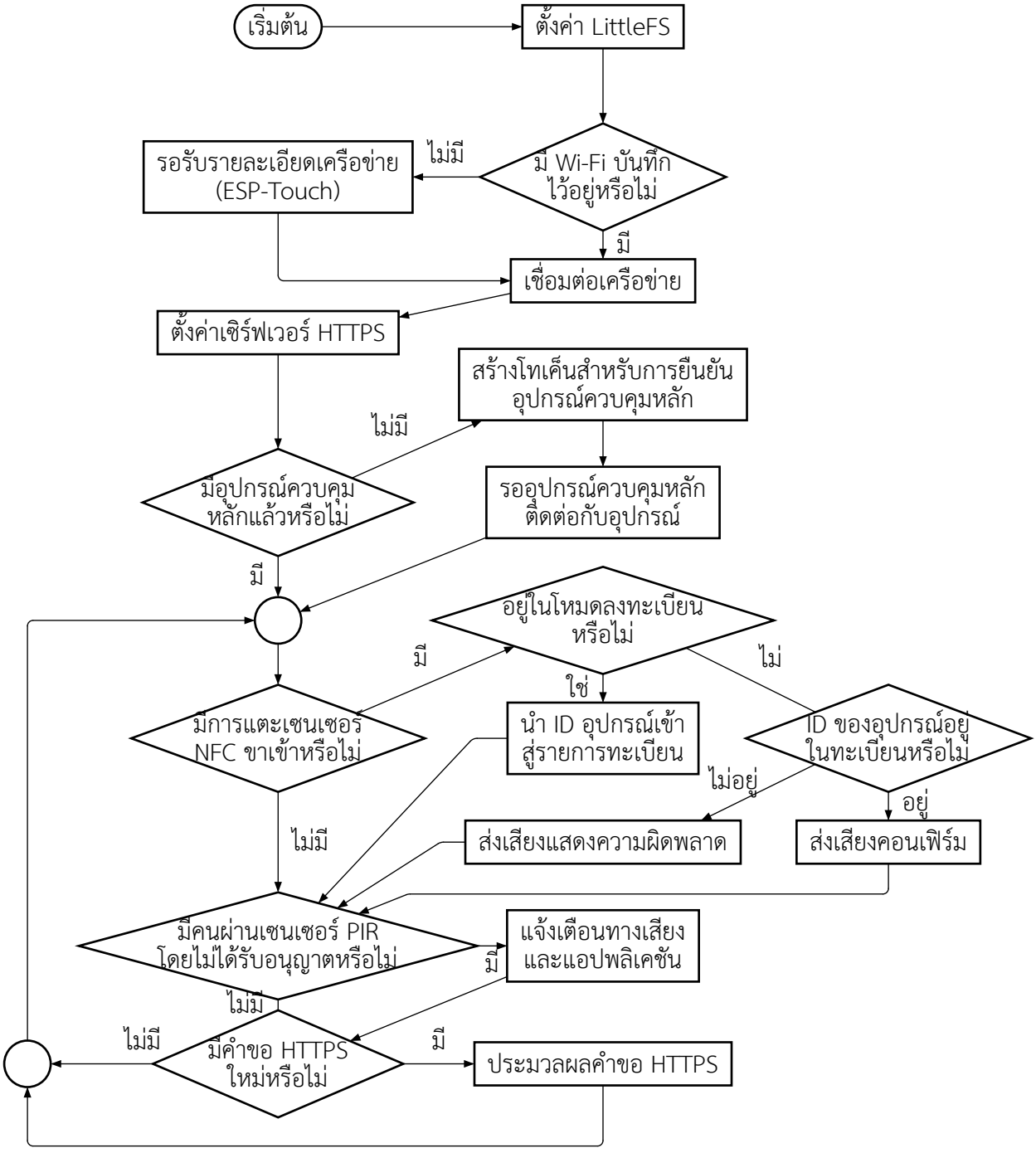
ตาราง 3.1 ระยะเวลาการดำเนินงาน

ขั้นตอนการดำเนินการ	**พ.ศ.2568**			**พ.ศ.2569**	
	ต.ค.	**พ.ย.**	**ธ.ค.**	**ม.ค.**	**ก.พ.**
ศึกษาค้นคว้าข้อมูล	←	—	→		
เสนอหัวข้อ	←	→			
เสนอครั้งที่ 1		←	→		
ออกแบบและสร้าง		←	—	—	→
จัดซื้ออุปกรณ์ทดลอง		←	—	—	→
ทดลองการทำงาน			←	—	→
ปรับปรุงแก้ไข					← — →
เสนอครั้งที่ 2				←	→
จัดทำรูปเล่ม	←	—	—	—	— →
นำเสนอโครงงาน					

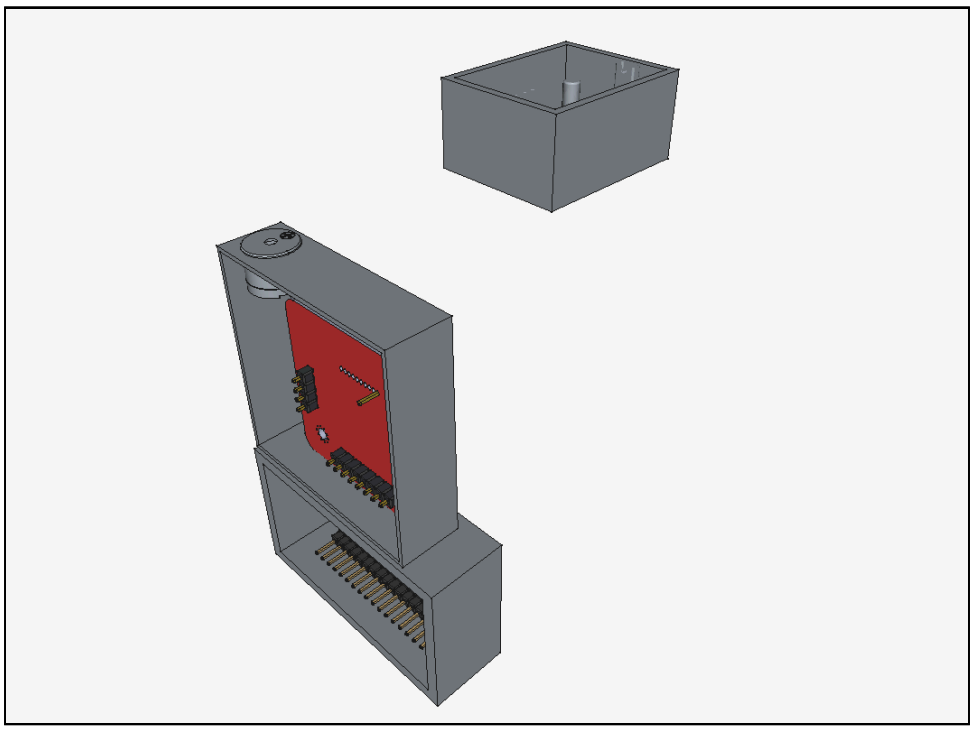
3.1.2 ผังการดำเนินงาน



3.1.3 ผังการทำงาน



3.2 การออกแบบ



รูปที่ 3.1 การออกแบบโครงสร้างเครื่องยืนยันตัวตนด้วย NFC

3.3 วัสดุอุปกรณ์

1) บอร์ด ESP32 (NodeMCU)
2) กล่องพลาสติก
3) บัซเซอร์ (Buzzer)
4) เซนเซอร์ NFC (PN532)

3.4 ขั้นตอนการประกอบ

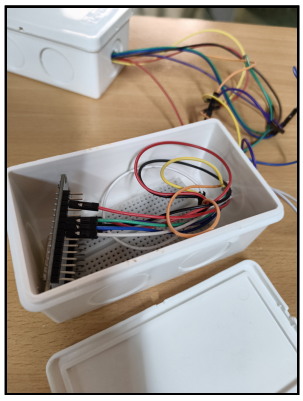
3.4.1 การติดตั้งอุปกรณ์

โครงงานแบ่งออกเป็น 3 โมดูล

1) โมดูลบอร์ด ESP32
2) โมดูลเซนเซอร์ NFC

3.4.1.1 โมดูลบอร์ด ESP32

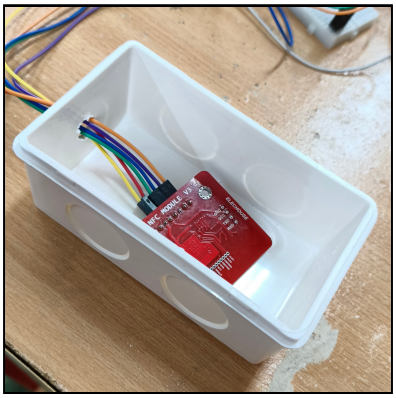
ทำการเจาะรู 2 รูสำหรับสายและร้อยสายเข้าไปในกล่อง



รูปที่ 3.2 ภายในกล่องโมดูล ESP32

3.4.1.2 โมดูลเซนเซอร์ NFC

ดำเนินการเจาะรูบริเวณตัวกล่องเพื่อใช้เป็นช่องสำหรับสายไฟ จากนั้นนำสายไฟร้อยผ่านช่องดังกล่าวและต่อเข้ากับเซนเซอร์ NFC ให้เรียบร้อย



รูปที่ 3.3 โมดูลเซนเซอร์ NFC

3.4.2 การเขียนเฟิร์มแวร์

โครงงานนี้ใช้ซอฟต์แวร์ PlatformIO ในการสร้างและจัดการโปรเจกต์เฟิร์มแวร์ โดยหากต้องการเพียงแค่เขียนเฟิร์มแวร์ลงไปยังบอร์ด ESP32 คุณจำเป็นต้องใช้ซอฟต์แวร์หลัก ๆ คือ PlatformIO Core และ Git (ไม่จำเป็น แต่เพื่อความสะดวกสบาย) อย่างไรก็ตาม PlatformIO จำเป็นต้องใช้ Python เวอร์ชัน 3.6 ขึ้นไปด้วยเช่นกัน ดังนั้นคุณจำเป็นต้องติดตั้ง Python ด้วยหากคุณยังไม่มี

ในขั้นตอนแรก โปรดเปิดเทอร์มินัลของคุณ ซึ่งโดยทั่วไปแล้วคุณสามารถค้นหาแอปพลิเคชัน “Terminal” ได้เลย โดยบน Windows 10 เวอร์ชั่นใหม่ ๆ และ Windows 11 จะมาพร้อมกับแอปพลิเคชัน Windows Terminal อย่างไรก็ตาม เมื่อเปิดแล้ว โปรดตรวจสอบให้แน่ใจว่าคุณกำลังใช้ PowerShell และไม่ใช่ Command Prompt

โดยในปัจจุบัน Python เวอร์ชันล่าสุดคือ Python 3.14.2 โดยคุณสามารถติดตั้ง Python และ Git บน Windows ได้ด้วยการใช้คำสั่งต่อไปนี้

```
winget install Python.Python.3.14 Git.Git -e -s winget
```

รูปที่ 3.4 คำสั่งในการติดตั้ง Python 3.14 และ Git

สำหรับระบบปฏิบัติการอื่นนั้น โดยปกติแล้วจะไม่ต้องติดตั้ง Python เพิ่มเนื่องจากมีติดมากับระบบปฏิบัติการอยู่แล้ว อย่างไรก็ตาม บน Linux อาจต้องมีการติดตั้งการรองรับ Virtual Environment แยก โดยแต่ละระบบจะมีชื่อแพคเกจไม่เหมือนกัน โดยบน Debian, Fedora, และ Arch สามารถใช้คำสั่งต่อไปนี้ในการติดตั้งทั้ง Python Virtual Environment และ Git พร้อมกันได้

```
# Debian
sudo apt install python3-venv git
# Fedora
sudo dnf install python3-virtualenv git
# Arch
sudo pacman -S python-virtualenv git
```

รูปที่ 3.5 คำสั่งในการติดตั้ง Virtual Environment และ Git บนการแจกจ่าย Linux ต่าง ๆ

3.4.2.1 การติดตั้ง PlatformIO Core ผ่านแพคเกจ

หากคุณใช้ Fedora Linux หรือ Arch Linux (หรือลูก ๆ ของมัน) คุณสามารถติดตั้งแพคเกจ PlatformIO ได้โดยตรง โดยมีคำสั่งดังนี้

```
# Fedora Linux
sudo dnf install platformio
# Arch Linux
sudo pacman -S platformio-core
```

รูปที่ 3.6 คำสั่งในการติดตั้ง PlatformIO Core บน Fedora และ Arch Linux

1) หากคุณติดตั้งแพคเกจ Fedora นั้นแล้ว คุณไม่จำเป็นที่จะต้องติดตั้งกฎ udev ด้วยตนเอง
(ที่จะถูกกล่าวถึงในหัวข้อ 3.4.2.3)

2) หากคุณใช้ Arch คุณสามารถติดตั้งแพคเกจกฎ udev ได้โดยตรงโดยไม่ต้องดาวน์โหลดเอง

```
sudo pacman -S --asdeps platformio-core-udev
```

รูปที่ 3.7 คำสั่งในการติดตั้งกฎ udev ของ PlatformIO บน Arch Linux

3.4.2.2 การติดตั้ง PlatformIO Core ผ่านสคริปต์

ถัดไป ในการติดตั้ง PlatformIO Core สามารถทำได้โดยการใช้สคริปต์ติดตั้ง โดยสำหรับ curl สามารถใช้คำสั่งนี้ได้

```
curl -fsSL -o get-platformio.py https://raw.githubusercontent.
com/platformio/platformio-core-installer/master/get-platformio.py
```

รูปที่ 3.8 คำสั่ง curl ที่ใช้ในการดาวน์โหลดสคริปต์ติดตั้ง PlatformIO

หรือหากต้องการใช้ wget

```
wget -O get-platformio.py https://raw.githubusercontent.com/
platformio/platformio-core-installer/master/get-platformio.py
```

รูปที่ 3.9 คำสั่ง wget ที่ใช้ในการดาวน์โหลดสคริปต์ติดตั้ง PlatformIO

หรือสำหรับ PowerShell, สามารถใช้ iwr (หรือชื่อเต็มคือ Invoke-WebRequest) ได้

```
iwr -OutFile get-platformio.py -Uri https://raw.githubusercontent
.com/platformio/platformio-core-installer/master/get-platformio.
py
```

รูปที่ 3.10 คำสั่ง iwr ที่ใช้ในการดาวน์โหลดสคริปต์ติดตั้ง PlatformIO
(มีการเว้นบรรทัดใหม่เนื่องจากพื้นที่ไม่เพียงพอ โปรดอย่าเว้นบรรทัดเมื่อพิมพ์คำสั่งจริง)

แล้วดังนั้นจึงใช้คำสั่ง python3 get-platformio.py ในการรันสคริปต์ติดตั้งที่ได้ทำการดาวน์โหลดมา โดยค่าเริ่มต้นแล้ว PlatformIO จะไม่เพิ่มตนเองเข้าไปยังตัวแปรสิ่งแวดล้อม PATH ซึ่งจำเป็นในการใช้คำสั่งจากที่ไหนก็ได้โดยไม่ต้องกล่าวถึงไฟล์พาธ

โดยสำหรับ Linux แล้วนั้น คุณต้องเพิ่ม $HOME/.local/bin/ เข้าไปยัง PATH ของคุณ โดยหากคุณใช้ Bash คุณสามารถแก้ไข ~/.bash_profile และเพิ่มบรรทัดนี้เข้าไปได้

```
export PATH=$PATH:$HOME/.local/bin
```

รูปที่ 3.11 โคดที่ต้องใช้ในการเพิ่ม ~/.local/bin เข้า PATH

หากคุณใช้ Zsh สามารถใช้โคดเดียวกันได้ เพียงแต่คุณต้องแก้ไขไฟล์ ~/.zprofile หรือ ~/.zshrc แทน

โดยบน Windows มีขั้นตอนดังนี้

1) กด Windows + R
2) พิมพ์ sysdm.cpl และกด Enter
3) ในหน้าต่าง *System Properties* คลิกไปยังแท็บ *Advanced*
4) คลิกปุ่ม *Environment Variables*

จากนั้น เลือกตัวแปร *Path* ในส่วน *User variables* แล้วจึงกด *Edit* แล้วเพิ่ม %USERPROFILE%\.platformio\penv\Scripts\ เข้าไปในรายการ

3.4.2.3 99-platformio-udev.rules

ผู้ใช้ Linux จำเป็นที่จะต้องติดตั้งกฎ udev โดยสามารถดูไฟล์กฎ udev เวอร์ชันล่าสุดได้ที่ https://raw.githubusercontent.com/platformio/platformio-core/develop/

platformio/assets/system/99-platformio-udev.rules และโปรดตรวจสอบว่า PID และ VID ของบอร์ดคุณอยู่ในไฟล์กฎนั้น โดยคุณสามารถดู PID/VID ของบอร์ดคุณได้ผ่านคำสั่ง pio device list

โดยไฟล์นั้นต้องถูกวางอยู่ที่ /etc/udev/rules.d/99-platformio-udev.rules (ตำแหน่งที่ดีที่สุด) หรือ /lib/udev/rules.d/99-platformio-udev.rules (อาจจำเป็นสำหรับบางระบบที่พัง)

โปรดใช้คำสั่งต่อไปนี้ในการดาวน์โหลดและวางไฟล์นั้นไว้ในสถานที่ที่ถูกต้อง

```
curl -fsSL https://raw.githubusercontent.com/platformio/
platformio-core/develop/platformio/assets/system/99-platformio-
udev.rules | sudo tee /etc/udev/rules.d/99-platformio-udev.rules
```

รูปที่ 3.12 คำสั่งในการดาวน์โหลดไฟล์กฎ udev

หรือคุณก็สามารถดาวน์โหลดไฟล์นั้นด้วยตัวเองและคัดลอกมันไปในโฟลเดอร์ที่หมายได้เช่นกัน

```
sudo cp 99-platformio-udev.rules /etc/udev/rules.d/99-platformio-
udev.rules
```

รูปที่ 3.13 คำสั่งในการคัดลอกไฟล์กฎ udev ไปยังสถานที่ที่ถูกต้อง

หลังจากนั้น รีสตาร์ทบริการ udev

```
sudo service udev restart
```

รูปที่ 3.14 คำสั่งในการรีสตาร์ทบริการ udev

หรือทำการรีโหลดและทริกเกอร์กฎ

```
sudo udevadm control --reload-rules
sudo udevadm trigger
```

รูปที่ 3.15 คำสั่งในการรีโหลดกฎ udev

หลังจากติดตั้งไฟล์นี้แล้ว ถอดสายที่เชื่อมต่อระหว่างบอร์ดและคอมพิวเตอร์ของคุณแล้วเสียบมันใหม่

3.4.2.4 การดาวน์โหลดโปรเจกต์

สามารถใช้ Git ในการ clone โปรเจกต์ได้ด้วยคำสั่งต่อไปนี้

```
git clone https://gitskette.dailitation.xyz/linesofcodes/
liteauth-firmware32.git
```

รูปที่ 3.16 คำสั่งในการโคลนโคดสำหรับเฟิร์มแวร์

โดย Git นั้นจะทำการโคลนโปรเจกต์ไปที่โฟลเดอร์ liteauth-firmware32 เนื่องจากเป็นชื่อของ Git repository หรือหากไม่ต้องการใช้ Git กรุณาไปที่ https://gitskette.dailitation.xyz/linesofcodes/liteauth-firmware32 และทำการคลิกปุ่ม *Code* แล้วกด *Download ZIP* หรือ *Download TAR.GZ* แล้วทำการแตกไฟล์ได้ตามปกติ จากนั้นจึงไปที่โฟลเดอร์ของคุณในเทอร์มินัลโดยใช้คำสั่ง cd

3.4.2.5 คำสั่ง PlatformIO เบื้องต้น

1) pio run --list-targets เพื่อดูรายการเป้าหมายคำสั่งรัน
2) pio run upload เพื่อรันเป้าหมายอัพโหลด ซึ่งนี่คือคำสั่งที่คุณควรจะใช้ในการเขียนเฟิร์มแวร์ลงบนบอร์ด
3) pio device monitor เพื่อเปิด Serial Monitor

และโปรดใช้คำสั่งประเภท pio run ในโฟลเดอร์ของโปรเจกต์

3.5 สร้างไฟล์แอปพลิเคชันด้วยตนเอง

โครงงานนี้ใช้แอปพลิเคชันที่สร้างขึ้นมาเอง โดยในการพัฒนาแอปพลิเคชัน อย่างน้อยต้องมีส่วนประกอบดังกล่าวก่อน

1) Flutter
2) Git (ซึ่งคุณจะติดตั้งแล้วหากคุณทำตามหัวข้อ 3.4.2)

อย่างไรก็ตาม Flutter มีข้อจำกัดว่า มีเพียง Android เท่านั้นที่ไม่ว่าแพลตฟอร์มไหนก็จะสามารถคอมไพล์ไฟล์ .apk ออกมาได้ ดังนั้น การสร้างแอปพลิเคชันสำหรับ Linux ต้องทำบน Linux เท่านั้น และการสร้างแอปพลิเคชันสำหรับ Windows ต้องทำบน Windows เท่านั้น

3.5.1 การติดตั้งโปรแกรมเขียนโคด

จริง ๆ แล้วนั้น Flutter สามารถทำงานกับโปรแกรมเขียนโคดใดก็ได้ แต่มีโปรแกรมเหล่านี้ที่อาจมีประสบการณ์การพัฒนาที่ดีกว่าโปรแกรมอื่น

1) Visual Studio Code (VS Code)
2) Android Studio
3) JetBrains IntelliJ
4) Firebase Studio

โครงงานนี้ใช้โปรแกรมเขียนโคด Android Studio เป็นหลักเนื่องจากแอปพลิเคชันโครงงานมี Android เป็นเป้าหมายหลัก และ Android SDK สามารถจัดการได้ง่ายกว่าใน Android Studio

การติดตั้ง Flutter สามารถทำได้สองวิธีด้วยกัน คือการติดตั้งผ่าน Visual Studio Code (VS Code) และการติดตั้งด้วยตนเอง โดยหากต้องการใช้ VS Code เป็นโปรแกรมเขียนโคดอยู่แล้ว สามารถ

ติดตั้งผ่าน VS Code ได้เลย แต่ก่อนอื่น ต้องทำการติดตั้งโปรแกรมและไลบรารีพื้นฐานที่จำเป็นสำหรับ Flutter ก่อน

3.5.2 การติดตั้งโปรแกรมและไลบรารีที่จำเป็น

1) สำหรับ Windows ติดตั้ง Git สำหรับ Windows ซึ่งคุณสามารถดูขั้นตอนการติดตั้งได้ที่ https://git-scm.com/install/windows หรือเพียงแค่ใช้คำสั่งด้านล่าง

```
winget install --id Git.Git -e --source winget
```

รูปที่ 3.17 คำสั่งในการติดตั้ง Git

2) สำหรับ Linux โปรดดูหัวข้อ 3.5.5.2 สำหรับรายละเอียดแพคเกจและคำสั่งที่ต้องใช้สำหรับระบบต่าง ๆ

3) สำหรับ macOS ใช้คำสั่งต่อไปนี้ในการติดตั้งเครื่องมือ Xcode ต่าง ๆ รวมถึง Git

```
xcode-select --install
```

รูปที่ 3.18 คำสั่งในการติดตั้งเครื่องมือ Xcode

3.5.3 การติดตั้งผ่าน Visual Studio Code

1) เปิด VSCode

2) ติดตั้งส่วนขยาย Flutter ซึ่งอยู่ภายใต้ ID Dart-Code.flutter ทั้งบน Visual Studio Marketplace และ OpenVSX

3) ติดตั้ง Flutter ด้วย VS Code

ก) เปิด Command Palette ด้วยเมนู *View* > *Command Palette* หรือกด Ctrl + Shift + P

ข) ใน Command Palette พิมพ์ flutter

ค) เลือก *Flutter: New Project*

ง) VS Code จะให้คุณเลือก Flutter SDK บนคอมพิวเตอร์ของคุณ เลือก *Download SDK*

จ) เมื่อหน้าไดอะลอก *Select Folder for Flutter SDK* แสดงขึ้น เลือกสถานที่ที่คุณอยากติดตั้ง Flutter

ฉ) คลิก *Clone Flutter* โดยในระหว่างการดาวน์โหลด VS Code จะแสดงการแจ้งเตือนนี้

```
Downloading the Flutter SDK. This may take a few minutes.
```

การดาวน์โหลดนี้จะใช้เวลาสองสามนาที หากคุณเชื่อว่าการดาวน์โหลดหยุดชะงัก คุณสามารถคลิก *Cancel* แล้วเริ่มต้นการติดตั้งใหม่ได้

ช) คลิก *Add SDK to PATH* และเมื่อเสร็จสิ้น จะมีการแจ้งเตือน

```
The Flutter SDK was added to your PATH
```

ช) VS Code อาจแสดงการแจ้งเตือนเกี่ยวกับการเก็บข้อมูลของ Google หากคุณยินยอม คลิก *OK*

ฌ) เพื่อความแน่ใจ กรุณาปิดเทอร์มินัลทุกหน้าต่างหรือรีสตาร์ท VS Code เพื่อให้แน่ใจว่า Flutter จะสามารถใช้ผ่านเทอร์มินัลได้

4) เมื่อเสร็จสิ้น ใช้คำสั่ง flutter doctor -v ในเทอร์มินัลที่คุณเลือกเพื่อตรวจสอบการติดตั้ง Flutter ของคุณ หากคำสั่งไม่เจอหรือเกิดข้อผิดพลาดขึ้น ตรวจสอบ https://docs.flutter.dev/install/troubleshoot สำหรับข้อมูลเพิ่มเติม

3.5.4 การติดตั้งด้วยตนเอง

แนะนำให้ทำตาม https://docs.flutter.dev/install/manual#install-flutter เนื่องจากกระบวนการนี้ต้องใช้ข้อมูลที่ใหม่ล่าสุด

1) ดาวน์โหลด Flutter (สามารถหาปุ่มดาวน์โหลดได้จากลิงก์ด้านบน)
2) สร้างโฟลเดอร์สำหรับเก็บ Flutter SDK
3) ทำการแตกไฟล์ที่ดาวน์โหลดมา
4) เพิ่ม Flutter เข้าไปยัง PATH ของคุณ (วิธีการขึ้นอยู่กับระบบปฏิบัติการ)
5) ยืนยันความถูกต้องของการติดตั้งของคุณด้วยคำสั่ง flutter doctor -v

3.5.5 ข้อมูลเฉพาะแพลตฟอร์ม

3.5.5.1 Android

ในการพัฒนาแอปพลิเคชัน Android โดยใช้เฟรมเวิร์ก Flutter จำเป็นต้องใช้ส่วนประกอบเครื่องมือพัฒนา Android ดังนี้

1) Android SDK (API Level 36 ณ เวลาที่พิมพ์)
2) Android SDK Build-Tools
3) Android SDK Command-line Tools
4) Android SDK Platform-Tools
5) Android Emulator (ไม่บังคับ)

โดยแนะนำให้จัดการและติดตั้งเครื่องมือเหล่านี้ผ่าน Android Studio

ในการติดตั้ง Android SDK ควรติดตั้ง Android SDK ล่าสุดถึงแม้ว่าอุปกรณ์ของคุณจะใช้เวอร์ชันที่เก่ากว่านั้น เพื่อความมั่นใจว่าแอปพลิเคชันจะสามารถใช้กับอุปกรณ์ที่ใหม่ล่าสุดได้

แอปพลิเคชัน Android จะมี SDK/API level เป้าหมาย (Target SDK/API level) และ SDK/API level ขั้นต่ำ (Minimum SDK/API level) โครงงานนี้ ณ เวลาที่พิมพ์ ใช้ API level เป้าหมาย 36 (Android 16) และ API level ขั้นต่ำ 24 (Android 7) ซึ่งรวมกันแล้ว แอปพลิเคชัน Android จะสามารถติดตั้งได้บนระบบตั้งแต่ API level ขั้นต่ำจนถึง API level เป้าหมาย หรือก็คือแอปพลิเคชันในโครงงานนี้สามารถติดตั้งได้ตั้งแต่บนระบบ Android 7 ถึง Android 16 นั่นเอง

3.5.5.1.1 Java/Kotlin

Java และ Kotlin เป็นภาษาสำคัญสำหรับการพัฒนาแอปพลิเคชัน Android ถึงอย่างไรก็ตาม แอปพลิเคชัน Flutter นั้นถูกเขียนด้วยภาษา Dart แต่ยังจำเป็นต้องมีโคด Java และ Kotlin เล็กน้อยเพื่อเริ่มต้นแอปพลิเคชัน Flutter

โดยปกติแล้ว Flutter จะสร้างโคดพื้นฐานขึ้นมาให้สำหรับการเริ่มแอปพลิเคชันแบบพื้นฐาน ดังนั้นจึงไม่จำเป็นต้องมีการเขียนโคด Java หรือ Kotlin เอง แต่ในบางกรณี อาจต้องเขียนโคดเพิ่มเองหากมีความต้องการเข้าถึงฟีเจอร์พื้นฐานระบบที่ Flutter ไม่มี API เพื่อให้เข้าถึงได้ และไม่มีแพคเกจเพื่อรองรับฟีเจอร์ที่ต้องการ

โครงการนี้ใช้ Java 21 (JetBrains Runtime/Azul Zulu OpenJDK) เป็นหลักในการทำงานกับ Gradle แต่แอปพลิเคชัน Android ที่ผลิตออกมานั้น เพื่อให้เข้ากับเวอร์ชันที่เก่ากว่าของระบบปฏิบัติการได้ ใช้ Java 11

3.5.5.1.2 Gradle

Gradle เป็นเครื่องมือสร้างระบบอัตโนมัติสำหรับการพัฒนาซอฟต์แวร์หลายภาษา จัดการงานต่าง ๆ เช่น การคอมไพล์ การแพ็คเกจ การทดสอบ การปรับใช้ และการเผยแพร่ ภาษาที่รองรับ ได้แก่ Java (รวมถึงภาษา Kotlin, Groovy, Scala ที่ใช้ JDK), C/C++ และ JavaScript Gradle พัฒนาต่อยอดจากแนวคิดของ Apache Ant และ Apache Maven และนำเสนอภาษาเฉพาะโดเมนที่ใช้ Groovy และ Kotlin ซึ่งต่างจากการกำหนดค่าโครงการที่ใช้ XML ที่ Maven ใช้ Gradle ใช้กราฟแบบอะไซคลิกกำกับทิศทางเพื่อจัดการการอ้างอิง กราฟนี้ใช้เพื่อกำหนดลำดับของงานที่ควรดำเนินการ Gradle ทำงานบน Java Virtual Machine

Gradle คือเครื่องมือหลักที่ใช้ในการจัดการโปรเจกต์ Java ส่วนใหญ่ รวมถึงโปรเจกต์ Android โดยในโครงการนี้ จะใช้ Gradle เวอร์ชัน 8.14.3 เป็นหลัก

โดยปกติแล้ว ผู้พัฒนานั้นไม่มีความจำเป็นที่จะต้องแตะต้อง Gradle ด้วยตนเอง และ Flutter จะทำการจัดการเอง แต่หากมีความจำเป็นต้องใช้คำสั่ง Gradle ด้วยตนเอง จะมีสคริปต์ gradlew (หรือ gradlew.bat สำหรับผู้ใช้ Windows) ภายในโฟลเดอร์ android ของโปรเจกต์ Flutter เสมอเพื่อเรียกใช้ Gradle ที่ถูกดาวน์โหลดมาสำหรับโปรเจกต์นั้น ๆ

3.5.5.2 Linux

เช่นเดียวกับ Android ที่กล่าวไปข้างต้น Flutter มีการสร้างโคดสำหรับการเปิดแอปพลิเคชันแบบพื้นฐาน แต่สำหรับ Linux แล้วนั้น Flutter ใช้โคด C++ และเฟรมเวิร์ก CMake ในการสร้างรากฐานของแอปพลิเคชัน

ในการพัฒนาแอปพลิเคชันสำหรับ Linux ต้องติดตั้งโปรแกรมเพิ่มเติม (build dependencies) ขยายความคือ ด้านบนคือสิ่งที่จำเป็นหากมีระบบอื่นเป็นเป้าหมาย แต่หากต้องการพัฒนาแอปพลิเคชัน Linux ต้องติดตั้งโปรแกรมในรายการหน้าถัดไปเพิ่ม

1) GTK 3 (ไลบรารีสำหรับการพัฒนา)
2) pkg-config
3) ไลบรารี GNU Standard C++ v3
4) Clang
5) CMake
6) Ninja

การติดตั้งไลบรารีและโปรแกรมที่กล่าวไปข้างต้นจะแตกต่างกันไปแต่ละการแจกจ่าย Linux และ Flutter ใช้ไลบรารีพื้นฐานดังกล่าวในการทำงานของแอปพลิเคชัน (runtime dependencies)

1) GTK 3
2) blkid
3) LZMA

แต่โดยทั่วไปแล้ว ไลบรารีเหล่านี้ควรถูกติดตั้งมาอยู่แล้วหากคุณใช้ graphical desktop ทั่วไป

3.5.5.2.1 Debian

```
# Development dependencies:
sudo apt install curl git unzip xz-utils zip libglu1-mesa

# Linux build dependencies:
sudo apt install clang cmake ninja-build pkg-config libgtk-3-dev
libstdc++-12-dev

# Runtime dependencies:
sudo apt install libgtk-3-0 libblkid1 liblzma5
```

รูปที่ 3.19 คำสั่งในการติดตั้งรายการแพคเกจต่าง ๆ บน Debian

3.5.5.2.2 Fedora Linux

```
# Development dependencies:
sudo dnf install curl git unzip xz zip mesa-libglu

# Linux build dependencies:
sudo dnf install clang cmake ninja-build pkgconf gtk3

# Runtime dependencies:
sudo dnf install gtk3 libblkid xz
```

รูปที่ 3.20 คำสั่งในการติดตั้งรายการแพคเกจต่าง ๆ บน Fedora Linux

3.5.5.2.3 Arch Linux

```
# Development dependencies:
sudo pacman -S --needed curl git unzip xz zip glu

# Linux build dependencies:
sudo pacman -S --needed clang cmake ninja pkgconf gtk3

# Runtime dependencies:
sudo pacman -S --needed util-linux-libs xz gtk3
```

รูปที่ 3.21 คำสั่งในการติดตั้งรายการแพคเกจต่าง ๆ บน Arch Linux

3.5.5.3 macOS/iOS

การพัฒนาแอปพลิเคชันสำหรับ macOS และ iOS นั้นต้องทำบน macOS เท่านั้น และจำเป็นต้องพึ่งพาเครื่องมือ Xcode แต่เนื่องจากในโครงงานนี้ไม่มีผู้ใช้ macOS จึงไม่สามารถสร้างไบนารีสำหรับ macOS และ iOS ออกมาได้ และไม่ใช่เป้าหมายของโครงงานนี้เช่นกัน

3.6 การทดสอบ

1) เข้าแอพลิเคชัน liteauthconfig
2) ทำการเชื่อมต่อไวไฟที่ต้องการให้อุปกรณ์เชื่อมต่อ (ต้องไม่ใช่ไวไฟ 2.4 GHz)
3) กดปุ่มบวก
4) กดตั้งค่าอุปกรณ์ใหม่ (Setup new device)
5) ตั้งชื่ออุปกรณ์และใส่รหัสผ่านเครือข่าย
6) กดปุ่มติ๊กถูกเพื่อเริ่มการตั้งค่าอุปกรณ์
7) เข้าสู่โหมดลงทะเบียนอุปกรณ์ NFC
8) ทำการแตะอุปกรณ์ NFC (เช่นโทรศัพท์ที่รองรับหรือแท็ก NFC)
9) ออกจากโหมดลงทะเบียน
10) ทำการแตะอุปกรณ์ NFC ที่ลงทะเบียน
11) รอสังเกตุรายการกิจกรรม (Activity Logs)

3.7 การวิเคราะห์ข้อมูล

วิเคราะห์จากตารางการทดลองตามความเร็วในการทำงานจริงของเครื่องยืนยันตัวตนด้วย NFC

บทที่ 4
ผลการทดสอบ

ผลการทำโครงงานเครื่องยืนยันตัวตนด้วย NFC สามารถตรวจสอบอุปกรณ์ NFC ได้ตามวัตถุประสงค์ที่ตั้งไว้ในข้างต้น และสามารถใช้เป็นต้นแบบในการพัฒนาได้อีกต่อไป โดยการทดสอบจะทำการตรวจสอบค่าความหน่วง (latency) ของการติดต่อสื่อสารระหว่างอุปกรณ์ควบคุมและเครื่องยืนยันตัวตนเป็นหลัก โดยสิ่งที่เหมือนกันในทุกการทดลองคือแอพลิเคชันไคลเอนต์จะทำการขอข้อมูลใหม่ทุก 15 วินาทีโดยไม่รอเวลาคำขอเก่าสำเร็จ

การทดสอบจะใช้อุปกรณ์ไคลเอนต์สองประเภทด้วยกัน ได้แก่ คอมพิวเตอร์และโทรศัพท์มือถือ โดยรายละเอียดทางเทคนิคของอุปกรณ์ไคลเอนต์ที่ใช้มีดังนี้

1) คอมพิวเตอร์ที่ใช้คือโน๊ตบุ๊ค MSI Thin 15 B12UCX

ก) หน่วยประมวลผล Intel® Core™ i5-12450H

ข) ตัวควบคุมเครือข่ายไร้สาย Intel® Wi-Fi 6E AX211

ค) ระบบปฏิบัติการ Arch Linux, Linux Kernel 6.18.9-arch1-2, Avahi 1:0.9rc3-1, nss-mdns 0.15.1-2

ง) มีการปรับแต่ง nss-mdns โดยการแก้ไขไฟล์ /etc/nsswitch.conf ในบรรทัด hosts เป็นดังนี้ (โดยไม่รวมการเว้นบรรทัดใหม่)

```
hosts: mymachines mdns4_minimal [NOTFOUND=return] resolve [!
UNAVAIL=return] files myhostname dns
```

รูปที่ 4.1 บรรทัดที่มีการแก้ไขของไฟล์ /etc/nsswitch.conf

2) โทรศัพท์มือถือที่ใช้คือ CMF by Nothing Phone 2 Pro

ก) หน่วยประมวลผล MediaTek Dimensity 7300 Pro 5G

ข) Android 16 (Kernel 6.1.134, 14 พฤษภาคม 2025), Nothing OS 4.0, Build Galaga-B4.0-260108-1654

และรายละเอียดทางเทคนิคของเซิร์ฟเวอร์ (ESP32) มีดังนี้:

1) บอร์ด 30-pin ESP-WROOM-32 (NodeMCU)
2) หน่วยประมวลผล Xtensa LX6 (240 MHz)
3) หน่วยประมวลผลร่วม (Coprocessor) FSM (20 MHz)
4) Wi-Fi 802.11 b/g/n/e/i

ก) รองรับเครือข่าย Wi-Fi 2 (802.11b), Wi-Fi 3 (802.11g), และ Wi-Fi 4 (802.11n)

ข) รองรับ 802.11n ในคลื่นความถี่ 2.4 GHz และความเร็วสูงสุด 150 Mbps

ค) รองรับมาตรฐานปรับปรุงคุณภาพบริการเครือข่ายไร้สาย (802.11e)

ง) รองรับมาตรฐานความปลอดภัย 802.11i (ซึ่งเป็นส่วนหนึ่งของ WPA2)

5) SRAM 520 KiB

6) ROM 448 KiB

4.1 ระยะเวลาในการเดินทางของข้อมูลทั้งสิ้น

โดยการทดสอบนี้จะทำการเปรียบเทียบและหาระยะเวลาที่ใช้ในการเดินทางของข้อมูลตั้งแต่เซนเซอร์ถูกใช้จนถึงเวลาไคลเอนต์ได้รับข้อมูลบนอุปกรณ์ไคลเอนต์ทั้งสองประเภท

ตารางที่ 4.1 ระยะเวลาในการเดินทางของข้อมูลทั้งสิ้นบนคอมพิวเตอร์

ที่	เวลาที่ตรวจจับ	เวลาที่ไคลเอนต์ได้รับข้อมูล	ความต่าง (วินาที)

ตารางที่ 4.2 ระยะเวลาในการเดินทางของข้อมูลทั้งสิ้นบนโทรศัพท์มือถือ

ที่	เวลาที่ตรวจจับ	เวลาที่ไคลเอนต์ได้รับข้อมูล	ความต่าง (วินาที)

4.2 ระยะเวลาในการส่งคำขอ

ตารางที่ 4.3 เปรียบเทียบระยะเวลาในการส่งคำขอ

ที่	คอมพิวเตอร์	โทรศัพท์มือถือ

บทที่ 5
สรุปผล อภิปรายผลและข้อเสนอแนะ

จากการทำโครงงานเรื่อง เครื่องยืนยันตัวตนด้วย NFC โดยมีวัตถุประสงค์เผื่อการจัดทำโครงงานครั้งนี้ ผู้จัดทำได้ผลสรุปดังนี้

5.1 สรุปผลโครงงาน

เครื่องยืนยันตัวตนด้วย NFC สามารถทำงานได้ตามวัตถุประสงค์ที่ตั้งไว้ โดยเมื่อมีการแตะบัตรหรือสมาร์ตโฟนที่รองรับ NFC เข้ากับตัวอ่าน ระบบจะทำการตรวจสอบข้อมูลกับฐานข้อมูลที่บันทึกไว้ หากข้อมูลถูกต้อง ระบบจะสั่งปลดล็อกและอนุญาตให้เข้าใช้งานได้ แต่หากข้อมูลไม่ถูกต้อง ระบบจะไม่อนุญาตและอาจมีการแจ้งเตือน

5.1.1 โมดูล NFC สามารถอ่านข้อมูลจากบัตรได้อย่างถูกต้องในระยะใกล้

5.1.2 บอร์ดควบคุมสามารถประมวลผลข้อมูลและสั่งงานอุปกรณ์ได้ตามโปรแกรมที่กำหนด

5.1.3 ระบบสามารถลดการใช้กุญแจแบบเดิม และเพิ่มความสะดวกให้กับผู้ใช้งาน

5.2 อภิปรายผล

จากการทดลองใช้งานจริง พบว่าเครื่องสามารถอ่านบัตรได้รวดเร็ว ใช้เวลาเพียงไม่กี่วินาทีในการตรวจสอบสิทธิ์การเข้าใช้งาน ระบบมีความแม่นยำสูงเมื่ออยู่ในระยะที่เหมาะสม อย่างไรก็ตาม หากวางบัตรห่างจากตัวอ่านมากเกินไป ระบบจะไม่สามารถอ่านข้อมูลได้

การใช้งาน NFC ช่วยเพิ่มความปลอดภัย เนื่องจากต้องมีบัตรหรืออุปกรณ์ที่ลงทะเบียนไว้เท่านั้นจึงจะสามารถเข้าใช้งานได้ ทำให้ลดความเสี่ยงจากบุคคลภายนอกได้อย่างมีประสิทธิภาพ

5.3 ข้อเสนอแนะ

5.3.1 ควรพัฒนาให้สามารถบันทึกข้อมูลการเข้า–ออกเพื่อตรวจสอบย้อนหลังได้

5.3.2 ควรเพิ่มระบบแจ้งเตือนผ่านแอปพลิเคชันหรือไลน์เมื่อมีการเข้าใช้งาน

5.3.3 ควรพัฒนาระบบให้รองรับผู้ใช้งานจำนวนมากขึ้น

5.3.4 ควรออกแบบกล่องอุปกรณ์ให้มีความแข็งแรงและสวยงามมากขึ้น

บรรณานุกรม

(1) **NodeMCU**. 15 สิงหาคม 2025. https://en.wikipedia.org/w/index.php?title=NodeMCU&oldid=1306030712 . สืบค้น 8 ธันวาคม 2025

(2) **ESP32**. 27 ธันวาคม 2025. https://en.wikipedia.org/w/index.php?title=ESP32&oldid=1329754183 . สืบค้น 28 ธันวาคม 2025

(3) **Espressif Systems**. 28 พฤศจิกายน 2025. https://en.wikipedia.org/w/index.php?title=Espressif_Systems&oldid=1324514195 . สืบค้น 29 ธันวาคม 2025

(4) **Microcontroller**. 21 ธันวาคม 2025. https://en.wikipedia.org/w/index.php?title=Microcontroller&oldid=1328645390 . สืบค้น 28 ธันวาคม 2025

(5) **Flutter**. 12 พฤศจิกายน 2025. https://en.wikipedia.org/w/index.php?title=Flutter_(software)&oldid=1321794260 . สืบค้น 30 พฤศจิกายน 2025

(6) **Git**. 1 พฤศจิกายน 2025. https://en.wikipedia.org/w/index.php?title=Git&oldid=1319901866 . สืบค้น 30 พฤศจิกายน 2025

(7) **Gitea**. 17 พฤศจิกายน 2025. https://en.wikipedia.org/w/index.php?title=Gitea&oldid=1322631603 . สืบค้น 30 พฤศจิกายน 2025

(8) **HTTPS**. 30 พฤศจิกายน 2025. https://en.wikipedia.org/w/index.php?title=HTTPS&oldid=1324964055 . สืบค้น 30 พฤศจิกายน 2025

(9) **Transport Layer Security**. 24 พฤศจิกายน 2025. https://en.wikipedia.org/w/index.php?title=Transport_Layer_Security&oldid=1323879251 . สืบค้น 30 พฤศจิกายน 2025

(10) **C (programming language)**. 3 มกราคม 2026. https://en.wikipedia.org/w/index.php?title=C_(programming_language)&oldid=1330924334 . สืบค้น 5 มกราคม 2026

(11) **GNU Compiler Collection**. 30 พฤศจิกายน 2025. https://en.wikipedia.org/w/index.php?title=GNU_Compiler_Collection&oldid=1324929423 . สืบค้น 30 พฤศจิกายน 2025

(12) **Dart (programming language)**. 21 พฤศจิกายน 2025. https://en.wikipedia.org/w/index.php?title=Dart_(programming_language)&oldid=1323401675 . สืบค้น 8 ธันวาคม 2025

(13) **X.509**. 11 พฤศจิกายน 2025. https://en.wikipedia.org/w/index.php?title=X.509&oldid=1321610537 . สืบค้น 29 พฤศจิกายน 2025

(14) **X.690**. 6 ตุลาคม 2025. https://en.wikipedia.org/w/index.php?title=X.690&oldid=1315457524#DER_encoding . สืบค้น 3 ธันวาคม 2025

(15) **OpenSSL**. 1 ธันวาคม 2025. https://en.wikipedia.org/w/index.php?title=OpenSSL&oldid=1325138239 . สืบค้น 3 ธันวาคม 2025

(16) **Material Design**. 15 พฤศจิกายน 2025. https://en.wikipedia.org/w/index.php?title=Material_Design&oldid=1322252287 . สืบค้น 6 ธันวาคม 2025

(17) **Buzzer**. 13 พฤศจิกายน 2025. https://en.wikipedia.org/w/index.php?title=Buzzer&oldid=1321902450 . สืบค้น 10 ธันวาคม 2025

(18) **Near-field communication**. 5 พฤศจิกายน 2025. https://en.wikipedia.org/w/index.php?title=Near-field_communication&oldid=1320616102 . สืบค้น 10 ธันวาคม 2025

(19) **Wi-Fi**. 18 กุมภาพันธ์ 2026. https://en.wikipedia.org/w/index.php?title=Wi-Fi&oldid=1338995964 . สืบค้น 19 กุมภาพันธ์ 2026

(20) **IEEE 802.11i-2004**. 22 มีนาคม 2025. https://en.wikipedia.org/w/index.php?title=IEEE_802.11i-2004&oldid=1281726294 . สืบค้น 19 กุมภาพันธ์ 2026

(21) **IEEE 802.11e-2005**. 26 สิงหาคม 2025. https://en.wikipedia.org/w/index.php?title=IEEE_802.11e-2005&oldid=1307917729 . สืบค้น 19 กุมภาพันธ์ 2026

(22) **Java versions in Android builds**. 21 พฤศจิกายน 2025. https://developer.android.com/build/jdks . สืบค้น 26 พฤศจิกายน 2025

(23) **Where is the value of "flutter.minSdkVersion" in Flutter project initialized?**. 26 สิงหาคม 2025. https://stackoverflow.com/a/79746636 . สืบค้น 26 พฤศจิกายน 2025

(24) **Are data models - ILP32 or LP64 decided by OS or the Hardware Architecture?**. 14 ตุลาคม 2020. https://stackoverflow.com/a/79746636 . สืบค้น 9 ธันวาคม 2025

(25) **Install Flutter manually**. 28 ตุลาคม 2025. https://docs.flutter.dev/install/manual . สืบค้น 6 ธันวาคม 2025

(26) **Install Flutter using VS Code**. 28 ตุลาคม 2025. https://docs.flutter.dev/install/with-vs-code . สืบค้น 6 ธันวาคม 2025

(27) **Set up Linux development**. 25 กันยายน 2025. https://docs.flutter.dev/platform-integration/linux/setup . สืบค้น 6 ธันวาคม 2025

(28) **Build Linux apps with Flutter**. 5 กันยายน 2025. https://docs.flutter.dev/platform-integration/linux/building . สืบค้น 6 ธันวาคม 2025

(29) **Flutter architectural overview**. 8 ธันวาคม 2025. https://docs.flutter.dev/resources/architectural-overview . สืบค้น 30 ธันวาคม 2025

(30) **Arch Linux - Package Search**. 15 ธันวาคม 2025. https://archlinux.org/packages/ . สืบค้น 15 ธันวาคม 2025

(31) **Fedora Packages**. 15 ธันวาคม 2025. https://packages.fedoraproject.org/ . สืบค้น 15 ธันวาคม 2025

(32) **Debian -- Packages**. 15 ธันวาคม 2025. https://www.debian.org/distrib/packages . สืบค้น 15 ธันวาคม 2025

(33) **Git - Install for Windows**. 30 พฤศจิกายน 2025. https://git-scm.com/install/windows . สืบค้น 1 ธันวาคม 2025

(34) Material Design [@materialdesign] . **The latest in Material Design is NOW available**. 28 ตุลาคม 2021. https://x.com/materialdesign/status/1453409331697885192 . สืบค้น 1 ธันวาคม 2025

(35) **Declarations**. 9 กุมภาพันธ์ 2025. https://cppreference.com/w/c/language/declarations.html . สืบค้น 8 ธันวาคม 2025

(36) **Arithmetic types**. 9 กุมภาพันธ์ 2025. https://cppreference.com/w/c/language/arithmetic_types.html . สืบค้น 9 ธันวาคม 2025

(37) Brian W. Kernighan และ Dennis M. Ritchie . **The ANSI C Programming Language** . ฉบับที่สอง (Second edition). Prentice Hall. 1988. สืบค้น 11 ธันวาคม 2025. [ออนไลน์]. เข้าถึงได้จาก: https://archive.org/details/the-ansi-c-programming-language-by-brian-w.-kernighan-dennis-m.-ritchie.org

(38) **default.csv**. 15 เมษายน 2024. https://github.com/espressif/arduino-esp32/blob/2ede5ac10923afd1e3a42ce1fb41930a9de05d16/tools/partitions/default.csv . สืบค้น 9 ธันวาคม 2025

(39) **Install Python Interpreter**. 14 กันยายน 2024. https://docs.platformio.org/en/latest/faq/install-python.html . สืบค้น 15 ธันวาคม 2025

(40) **System Requirements**. 30 พฤษภาคม 2022. https://docs.platformio.org/en/latest/core/installation/requirements.html . สืบค้น 15 ธันวาคม 2025

(41) **Installer Script (Recommended)**. 14 สิงหาคม 2023. https://docs.platformio.org/en/latest/core/installation/methods/installer-script.html . สืบค้น 15 ธันวาคม 2025

(42) **ESP32 Datasheet**. 8 ตุลาคม 2016. สืบค้น 19 กุมภาพันธ์ 2026. [ออนไลน์]. เข้าถึงได้จาก: https://www.esp32.dk/esp32_datasheet_en.pdf

บรรณานุกรมภาพ

รูปที่ 2.1 ไมโครคอนโทรลเลอร์ PIC ต่าง ๆ ที่มี EPROM ภายใน

Camillo - Own work, CC BY 2.5, https://commons.wikimedia.org/w/index.php?curid=569240

รูปที่ 2.2 ไมโครคอนโทรลเลอร์ Piggyback จาก MOSTEK

Medvedev - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=33161178

รูปที่ 2.3 บอร์ด NodeMCU ที่มี ESP32-C3-32S

Popolon, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=112634884

รูปที่ 2.4 แสดงการทำงานเบื้องต้นของ LittleFS

เจ้าของ LittleFS (ภายใต้สัญญาอนุญาต BSD-3-Clause)

รูปที่ 2.5 เครื่องตรวจจับการเคลื่อนไหวแบบ PIR ทั่วไปสำหรับที่พักอาศัย/เชิงพาณิชย์

Jack LaRosa, Public Domain, https://commons.wikimedia.org/w/index.php?curid=4479143

รูปที่ 2.6 เครื่องตรวจจับความเคลื่อนไหว PIR ใช้สำหรับควบคุมไฟภายนอกอาคารแบบอัตโนมัติ

CHG, Public Domain, https://commons.wikimedia.org/w/index.php?curid=6087132

รูปที่ 2.7 กล้องดักถ่ายพร้อมระบบตรวจจับความเคลื่อนไหวแบบ PIR

Dariusz Kowalczyk, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=96211951

รูปที่ 2.8 สวิตซ์ไฟภายในอาคารที่ติดตั้งเซนเซอรตรวจจับการครอบครองแบบ PIR

Z22, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=35183184

รูปที่ 2.9 การออกแบบเซ็นเซอร์ตรวจจับการเคลื่อนไหว PIR

Versatile Techno - http://www.sensinova.in/pir-motion-sensor/SNPR11.php, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=48377787

รูปที่ 2.10 ตัวเรือนเครื่องตรวจจับความเคลื่อนไหว PIR พร้อมช่องหน้าต่างทรงกระบอกเหลี่ยมโดยแต่ละเหลี่ยมเป็นเลนส์เฟรสเนล โฟกัสแสงไปที่ชิ้นส่วนเซ็นเซอร์ไพโรอิเล็กทริกที่อยู่ด้านล่าง

CC BY-SA 3.0, https://en.wikipedia.org/w/index.php?curid=14193664

รูปที่ 2.11 ฝาครอบด้านหน้า PIR เท่านั้น (ถอดอุปกรณ์อิเล็กทรอนิกส์ออก) โดยมีแหล่งกำเนิดแสงจุดอยู่ด้านหลัง เพื่อแสดงเลนส์แต่ละตัว

Jack LaRosa, Public Domain, https://commons.wikimedia.org/w/index.php?curid=4463018

รูปที่ 2.12 PIR ที่ถอดฝาครอบด้านหน้าออก แสดงตำแหน่งของ เซ็นเซอร์ไพโรอิเล็กทริก (ลูกศรสีเขียว)

Jack LaRosa, Public Domain, https://commons.wikimedia.org/w/index.php?curid=4478366

รูปที่ 2.13 PID ทั่วไปสำหรับที่พักอาศัย/เชิงพาณิชย์ที่ ใช้กระจกแบ่งส่วนภายในเพื่อการโฟกัส

Jack LaRosa, Public Domain, https://commons.wikimedia.org/w/index.php?curid=4501665

รูปที่ 2.14 ถอดฝาครอบออกแล้ว กระจกแบ่งส่วน ด้านล่างมีแผงวงจรพิมพ์ (PC) อยู่ด้านบน

Deuxdad, Public Domain, https://commons.wikimedia.org/w/index.php?curid=4501724

รูปที่ 2.15 แผงวงจรพิมพ์ถูกถอดออกเพื่อแสดงกระจกแบบแบ่งส่วน

Jack LaRosa, Public Domain, https://commons.wikimedia.org/w/index.php?curid=4502198

รูปที่ 2.16 กระจกพาราโบลาแบบแบ่งส่วนถอดออกจากตัวเครื่อง

Deuxdad, Public Domain, https://commons.wikimedia.org/w/index.php?curid=4502224

รูปที่ 2.17 ด้านหลังของแผงวงจรที่หันเข้าหากระจกเมื่อติดตั้ง เซ็นเซอร์ไพโรอิเล็กทริกแสดงด้วยลูกศรสีเขียว

Jack LaRosa, Public Domain, https://commons.wikimedia.org/w/index.php?curid=4508036

รูปที่ 2.18 เครื่องตรวจจับความเคลื่อนไหวที่มีรูปแบบลำแสงซ้อนทับ ความยาวของลำแสงเป็นตัวชี้วัดความไวของเครื่องตรวจจับในทิศทางนั้น

AndreasCT, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=84066723

รูปที่ 2.19 สถาปัตยกรรม Flutter

Flutter, ภายใต้ CC BY 4.0

รูปที่ 2.20 เลเยอร์ต่าง ๆ ของแอพลิเคชัน Flutter

Flutter, ภายใต้ CC BY 4.0

รูปที่ 3.1 การออกแบบโครงสร้างเครื่องยืนยันตัวตนด้วย NFC

JetBrains เป็นเจ้าของเครื่องหมายการค้า CLion

ภาคผนวก

ภาคผนวก ก
งบประมาณในการจัดทำเครื่องยืนยันตัวตนด้วย NFC

งบประมาณในการจัดทำเครื่องยืนยันตัวตนด้วย NFC

ที่	รายการ	จำนวน	หน่วย	ราคา (บาท)
1	ESP32 NodeMCU ESP-WROOM-32 Wi-Fi and Bluetooth	1	175	175
2	Active Buzzer ลำโพงสัญญาณ 5V TMB12A05 12*9.5mm	1	6	6
3	PIR Motion Sensor HC-SR501	1	40	40
4	เซนเซอร์ NFC PN532 Module Set	2	79	158
5	แผ่นไม้	4	20	80
6	กล่องพลาสติก	3	40	120
รวมค่าใช้จ่าย				579

ภาคผนวก ข
คู่มือการใช้งาน

คู่มือการใช้งานเครื่องยืนยันตัวตนด้วย NFC

ภาคผนวก ค
โปรแกรม

โปรแกรมเครื่องยืนยันตัวตนด้วย NFC

1. Android Studio

รูปที่ 1 Android Studio

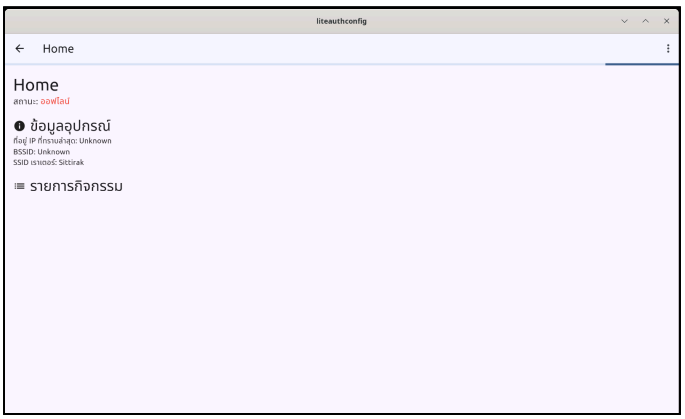
2. CLion

รูปที่ 2 JetBrains CLion

3. PlatformIO

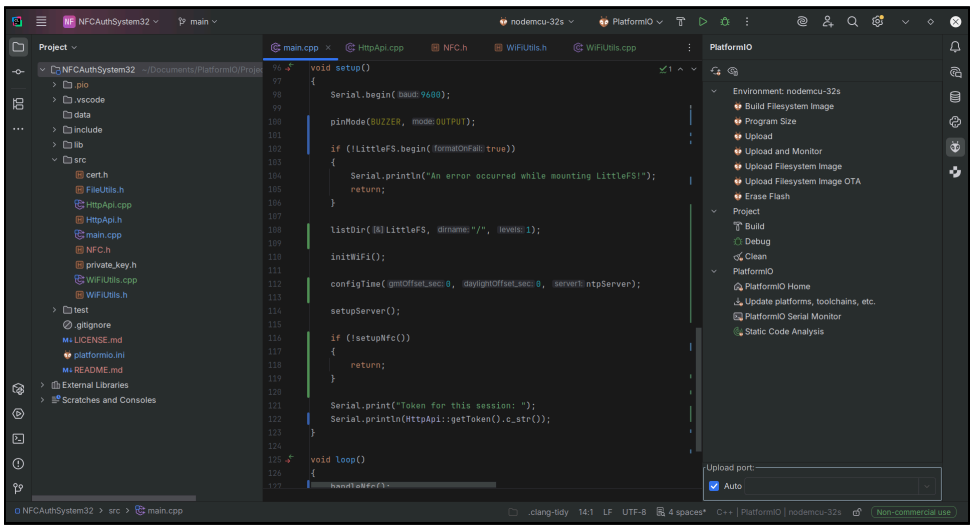
รูปที่ 3 PlatformIO

4. liteauthconfig



รูปที่ 4 liteauthconfig บนเดสก์ท็อป

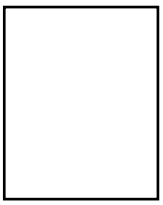
5. liteauth-firmware



รูปที่ 5 โคด liteauth-firmware

ภาคผนวก ง
ประวัติย่อผู้จัดทำ

ประวัติย่อผู้จัดทำ



ชื่อ นางสาวประภากร ศรีวรสาร
เกิด วันที่ 2 ตุลาคม พ.ศ.2551
ที่อยู่ บ้านเลขที่ 192 หมู่ที่ 15 ตำบล ในเมือง
อำเภอ เมืองหนองคาย จังหวัด หนองคาย

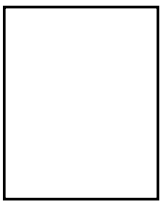
ประวัติการศึกษา

พ.ศ.2564	ป.6	โรงเรียน อนุบาลหนองคาย
พ.ศ.2567	ม.3	โรงเรียน ปทุมเทพวิทยาคาร
พ.ศ.2569	ปวช.	สาขาวิชาช่างเทคนิคคอมพิวเตอร์ วิทยาลัยเทคนิคหนองคาย

ประวัติการฝึกงาน

พ.ศ. 2569 ฝึกงานโรงพยาบาลหนองคาย

ประวัติย่อผู้จัดทำ



ชื่อ นางสาวพีรดา แสงแป้

เกิด วันที่ 8 ธันวาคม พ.ศ.2551

ที่อยู่ บ้านเลขที่ 218 หมู่ที่ 12 ตำบล โพธิ์ชัย
อำเภอ เมือง จังหวัด หนองคาย

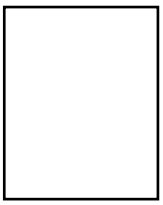
ประวัติการศึกษา

พ.ศ.2564	ป.6	โรงเรียนฮัวเคียวกงฮัก
พ.ศ.2567	ม.3	โรงเรียนปทุมเทพวิทยาคาร
พ.ศ.2569	ปวช.	สาขาวิชาช่างเทคนิคคอมพิวเตอร์ วิทยาลัยเทคนิคหนองคาย

ประวัติการฝึกงาน

พ.ศ. 2569 ฝึกงานโรงพยาบาลหนองคาย

ประวัติย่อผู้จัดทำ



ชื่อ นายศตคุณ อุตมะ
เกิด วันที่ 25 กุมภาพันธ์ พ.ศ.2552
ที่อยู่ บ้านเลขที่ 91/1 หมู่ที่ 1 ตำบล บ้านหม้อ
อำเภอ ศรีเชียงใหม่ จังหวัด หนองคาย

ประวัติการศึกษา

พ.ศ.2564	ป.6	โรงเรียนหัสดีศึกษา
พ.ศ.2567	ม.3	โรงเรียนหัสดีศึกษา
พ.ศ.2569	ปวช.	สาขาวิชาช่างเทคนิคคอมพิวเตอร์ วิทยาลัยเทคนิคหนองคาย

ประวัติการฝึกงาน

พ.ศ. 2569 ฝึกงานโรงพยาบาลหนองคาย

ภาคผนวก จ
ลิขสิทธิ์โครงงาน

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally

incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this

License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its

content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation’s users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others’ Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version”

applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program>  Copyright (C) <year>  <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w’ and `show c’ should show the appropriate parts of the General Public License. Of course, your program’s commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see https://www.gnu.org/licenses/.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read https://www.gnu.org/licenses/why-not-lgpl.html.