

เครื่องลงคะแนนเสียงอัจฉริยะด้วย Arduino/ESP32

บทนำ

เครื่องลงคะแนนเสียงอัจฉริยะคือระบบเลือกตั้งอิเล็กทรอนิกส์ที่ผสมผสานเทคโนโลยีหลายอย่างเพื่อให้การลงคะแนนมีความปลอดภัย โปร่งใส และรวดเร็ว ระบบนี้ใช้ไมโครคอนโทรลเลอร์ (เช่น Arduino หรือ ESP32) ควบคุมอุปกรณ์ต่าง ๆ เช่น เครื่องอ่านบัตร RFID, เซ็นเซอร์ลายนิ้วมือ, จอแสดงผล LCD, ปุ่มกดสำหรับเลือกผู้สมัคร และโมดูลสื่อสาร Wi-Fi เพื่อส่งผลการโหวตไปยังคอมพิวเตอร์แบบเรียลไทม์ นอกจากนี้ยังสามารถต่อเครื่องพิมพ์ความร้อนขนาดเล็กเพื่อพิมพ์ใบเสร็จยืนยันการลงคะแนนให้ผู้ใช้ได้ ระบบได้รับการออกแบบให้รองรับทั้งโหมดการลงคะแนนแบบเลือกได้คนเดียว (Single Vote) และแบบเลือกได้หลายคนพร้อมกัน (Multi-Choice Vote) สูงสุด 5 คน โดยมีมาตรการป้องกันไม่ให้ผู้ใดลงคะแนนซ้ำด้วยการยืนยันตัวบุคคลสองขั้นตอน (บัตร RFID และลายนิ้วมือ) ¹ ² ทั้งนี้ ตัวเครื่องต้นแบบถูกประกอบในกล่องหรือโครงสร้างที่ทำจากกระดาษลัง DIY เพื่อความสวยงามและสะดวกในการนำเสนอผลงาน

คุณสมบัติหลักของระบบ:

- **โหมดการลงคะแนน:** สามารถสลับระหว่างโหมดเลือกได้คนเดียว กับโหมดเลือกได้หลายคน (สูงสุด 5 คน) ตามประเภทของการเลือกตั้ง
- **การยืนยันตัวตนสองขั้น:** ผู้มีสิทธิ์ลงคะแนนต้องแตะบัตร RFID และสแกนลายนิ้วมือก่อนลงคะแนน เพื่อป้องกันการปลอมแปลงและลงคะแนนซ้ำ ผู้ใช้แต่ละคนจะลงคะแนนได้เพียงครั้งเดียวเท่านั้น หากมีการพยายามลงคะแนนซ้ำ ระบบจะปฏิเสธทันที ²
- **ส่วนติดต่อผู้ใช้:** มีจอ LCD สำหรับแสดงคำแนะนำ/สถานะการทำงาน และปุ่มกดสูงสุด 8 ปุ่มสำหรับเลือกผู้สมัครที่ต้องการลงคะแนน (รองรับผู้สมัครได้สูงสุด 8 คน)
- **การส่งข้อมูลผลโหวต:** ใช้ความสามารถ Wi-Fi ของ ESP32 ในการส่งข้อมูลผลการลงคะแนนไปยังคอมพิวเตอร์ (เช่น ODROID หรือ PC) แบบไร้สาย เพื่อแสดงผลคะแนนบนแดชบอร์ดแบบเรียลไทม์ ผู้ควบคุมสามารถดูผลคะแนนสะสมได้ทันทีที่มีการลงคะแนนใหม่เข้ามา
- **การพิมพ์ใบยืนยัน:** หลังจากผู้ใช้ลงคะแนนสำเร็จ ระบบสามารถพิมพ์ใบเสร็จยืนยันการลงคะแนนผ่านเครื่องพิมพ์ความร้อน (thermal printer) ขนาดเล็ก เพื่อให้ผู้ลงคะแนนตรวจสอบว่าเสียงของตนถูกบันทึกเรียบร้อยแล้ว (แนวคิดคล้าย VVPAT ในการเลือกตั้งจริง)
- **การออกแบบโครงสร้าง:** ชุดอุปกรณ์ทั้งหมดติดตั้งอยู่ในกล่องหรือคอกาทำจากวัสดุอย่างกระดาษลัง ซึ่งออกแบบ DIY ให้ดูเหมือนตู้ลงคะแนนเสียงขนาดเล็ก มีความแข็งแรงพอประมาณและตกแต่งให้สวยงาม เหมาะแก่การนำไปแสดงในงานโครงงานนักศึกษา

ด้านล่างนี้เป็นรายละเอียดเชิงลึกของระบบ ตั้งแต่ส่วนประกอบฮาร์ดแวร์ การต่อวงจร ฝังงานการทำงาน โค้ดโปรแกรมทั้งฝั่งไมโครคอนโทรลเลอร์และฝั่งคอมพิวเตอร์ รวมถึงวิธีเตรียมฐานข้อมูลผู้มีสิทธิ์และขั้นตอนการทำงานของระบบแบบเป็นลำดับ

อุปกรณ์และการต่อวงจร

ระบบประกอบด้วยฮาร์ดแวร์สำคัญหลายรายการดังต่อไปนี้:

- **ไมโครคอนโทรลเลอร์:** สามารถใช้ Arduino UNO/MEGA หรือ ESP32 เป็นหน่วยประมวลผลกลางของระบบ (แนะนำใช้ ESP32 เพื่อความสะดวกในการเชื่อมต่อ Wi-Fi ในตัว)
- **เครื่องอ่าน RFID:** โมดูล MFRC522 (RFID RC522) ใช้สำหรับอ่านบัตรประจำตัวผู้มีสิทธิ์ลงคะแนน ความถี่ 13.56 MHz สื่อสารผ่านอินเทอร์เฟซแบบ SPI ³

- **เซ็นเซอร์ลายนิ้วมือ:** โมดูล AS608 (หรือรุ่นใกล้เคียงเช่น R305) ใช้สแกนลายนิ้วมือผู้ลงคะแนน ตัวโมดูลมีหน่วยความจำภายในสำหรับเก็บเทมเพลตลายนิ้วมือประมาณ 128 ลายนิ้วมือ และสื่อสารผ่านพอร์ตอนุกรม (UART) กับไมโครคอนโทรลเลอร์
- **จอ LCD:** จอแสดงผลตัวอักษร 16x2 (16 คอลัมน์ 2 แถว) หรือ 20x4 เพื่อใช้แสดงข้อความสถานะต่าง ๆ เช่น พร้อมให้ลงคะแนน, ยืนยันตัวตน, ข้อผิดพลาด เป็นต้น (ในที่นี้จะอ้างอิง 16x2 เพื่อความง่าย)
- **ปุ่มกด (Push Buttons):** ใช้สำหรับเลือกผู้สมัครที่ต้องการลงคะแนน โดยสามารถมีได้สูงสุด 8 ปุ่ม (รองรับผู้สมัคร 8 หมายเลข) การต่อวงจรจะใช้ปุ่มแบบ Momentary (กดติดปล่อยดับ) ต่อลงไปยังขาพอร์ตดิจิทัลของไมโครคอนโทรลเลอร์ พร้อมตัวต้านทาน Pull-down หรือใช้ Internal Pull-up ของ MCU เพื่อให้ตรวจจับสถานะการกดได้
- **บัสเซอร์/LED (อุปกรณ์เสริม):** บัสเซอร์ใช้ส่งเสียงแจ้งเตือนเมื่อมีเหตุการณ์บางอย่าง เช่น สแกนบัตรผ่านหรือเกิดข้อผิดพลาด LED สีเขียว/แดงสามารถใช้แสดงสถานะพร้อมลงคะแนนหรือการยืนยันตัวตนผ่าน (อุปกรณ์เสริมนี้ไม่ได้ระบุในคุณสมบัติหลัก แต่สามารถเพิ่มได้เพื่อเพิ่มประสบการณ์ผู้ใช้)
- **โมดูล Wi-Fi:** หากใช้ ESP32 จะมี Wi-Fi ในตัว แต่หากใช้ Arduino อาจต้องต่อโมดูล Wi-Fi แยกต่างหาก (เช่น ESP8266) อย่างไรก็ตาม ในที่นี้เราเลือกใช้ ESP32 เป็นหลักเพราะรวม Wi-Fi กับ MCU ไว้ในตัวเดียว
- **เครื่องพิมพ์ความร้อน (Thermal Printer):** เป็นเครื่องพิมพ์ใบเสร็จขนาดเล็กที่ใช้กระดาษความร้อน 58 มม. เชื่อมต่อผ่าน Serial TTL (ต้องการไฟเลี้ยง 5V แยกต่างหาก) เพื่อพิมพ์ข้อความยืนยันการลงคะแนนเมื่อโหวตสำเร็จ (การต่อพ่วงเป็นทางเลือกเพิ่มเติม)

สำหรับการต่อวงจร **Arduino/ESP32** กับอุปกรณ์ข้างต้น มีรายละเอียดดังนี้:

- **การต่อโมดูล RFID RC522:** โมดูล RC522 ทำงานที่ไฟ 3.3V (ควรใช้ 3.3V เพื่อความปลอดภัยของตัวโมดูล) และสื่อสารด้วย SPI ขาของ RC522 ได้แก่ SDA (SS), SCK, MOSI, MISO, IRQ, RST, 3.3V, GND โดยการต่อกับ ESP32 (ตัวอย่างใช้ ESP32 DevKit V1) จะต่อ SDA → GPIO 5, SCK → GPIO 18, MOSI → GPIO 23, MISO → GPIO 19, RST → GPIO 21, และใช้ไฟ 3.3V, GND ตามลำดับ ⁴ (ESP32 สามารถปรับเปลี่ยนขา SPI ได้ แต่ตัวอย่างนี้ใช้ SPI ชุดเริ่มต้น (VSPI) ของ ESP32)
- **การต่อเซ็นเซอร์ลายนิ้วมือ:** โมดูล AS608 มีสายเชื่อมต่อ 4 เส้นหลักคือ VCC (ไฟเลี้ยง 3.3V-5V), GND, TX, RX โดย TX/RX ของเซ็นเซอร์จะต่อเข้ากับ RX2 และ TX2 ของ ESP32 (หรือหากใช้ Arduino UNO สามารถใช้ SoftwareSerial ในการรับส่งข้อมูลได้) การเชื่อมต่อคือเซ็นเซอร์ TX → RX ของ MCU, เซ็นเซอร์ RX → TX ของ MCU และจ่ายไฟให้เซ็นเซอร์ที่ 3.3V หรือ 5V ตามสเปค (AS608 รองรับ 3.3V)
- **การต่อจอ LCD 16x2:** ถ้าใช้จอ LCD พร้อม I2C backpack จะง่ายขึ้น โดยต่อเพียง 4 เส้นคือ VCC (5V), GND, SDA, SCL จากโมดูล I2C ไปยังขา SDA/SCL ของ MCU (เช่น SDA → GPIO 21, SCL → GPIO 22 บน ESP32 หรือ A4, A5 บน Arduino UNO ตามลำดับ) ⁵ แต่หากใช้การต่อแบบขนาน 4-bit (ไม่มี I2C backpack) จะต้องต่อสายหลายเส้น: ขา RS, EN, D4-D7 ของ LCD ไปยังขาดิจิทัลของ MCU และต่อไฟเลี้ยง/ปรับแสงตามปกติ (ในตัวอย่าง Arduino UNO: RS→D13, EN→D12, D4→D11, D5→D10, D6→D9, D7→D8) ⁶
- **การต่อปุ่มกด:** ปุ่มกดแต่ละตัวมีขา 2 ขา (หรือ 4 ขาสำหรับปุ่มแบบมีขาเพื่อ) การต่อวงจรนิยมต่อแบบหนึ่งขาเข้าขาอินพุตของ MCU และอีกขาหนึ่งลง GND จากนั้นใช้ Pull-up ภายใน MCU (INPUT_PULLUP) เพื่อให้สถานะปกติของปุ่มเป็น HIGH และเมื่อกดจะต่อขาเข้าลง GND (อ่านค่า LOW เมื่อกด) ซึ่งวิธีนี้ลดจำนวนอุปกรณ์เพราะไม่ต้องใช้ตัวต้านทานภายนอก (หรือจะต่อแบบหนึ่งขาเข้าขาอินพุต หนึ่งขาไป VCC และใช้ Pull-down resistor ก็ได้) แต่หลักการคือเมื่อกดปุ่มให้เกิดการเปลี่ยนสถานะโลจิกที่ตรวจจับได้
- **การต่อบัสเซอร์/LED:** (ถ้ามี) บัสเซอร์แอคทีฟ 5V สามารถต่อขา + ไปขา MCU (ผ่านทรานซิสเตอร์ถ้ากระแสเกิน) และขา - ลง GND เพื่อให้ MCU ส่งเสียงได้ สำหรับ LED ให้ต่อขา Anode ผ่านตัวต้านทาน ~220Ω ไปยังขา MCU และขา Cathode ลง GND (ไฟ 3.3V ของ ESP32 สามารถไต่ SW LED ได้โดยตรง)

ด้านล่างเป็นแผนภาพบล็อกและแผนผังวงจรเพื่อความเข้าใจในการต่อระบบทั้งหมด:

แผนภาพบล็อกของระบบเครื่องลงคะแนนเสียงแบบ *biometric* (ยืนยันตัวบุคคลด้วยลายนิ้วมือและ *RFID*) โดยมีไมโครคอนโทรลเลอร์ *Arduino* เป็นศูนย์กลางเชื่อมต่อกับจอ *LCD* (สำหรับแสดงผลข้อความ), โมดูลสแกนลายนิ้วมือ, เครื่องอ่านบัตร *RFID* และปุ่มกดสำหรับลงคะแนน ⁷ แต่ละองค์ประกอบทำงานร่วมกันเพื่อให้การลงคะแนนมีความปลอดภัยและแม่นยำ

จากแผนภาพบล็อกข้างต้น จะเห็นว่าหลังจากผู้ใช้ได้รับการตรวจสอบสิทธิ์ผ่าน RFID และลายนิ้วมือ ไมโครคอนโทรลเลอร์จะอนุญาตให้กดปุ่มลงคะแนนเลือกผู้สมัครที่ต้องการ จากนั้นจึงบันทึกและส่งผลไปยังส่วนประมวลผลผลโหวตผ่าน Wi-Fi โดยจะแสดงผลบนคอมพิวเตอร์แบบเรียลไทม์ ขั้นตอนเหล่านี้จะถูกอธิบายอย่างละเอียดในส่วนถัดไป

แผนผังวงจรการเชื่อมต่ออุปกรณ์ทั้งหมด (ตัวอย่างการใช้ *Arduino Uno*): โมดูลสแกนลายนิ้วมือต่อเข้ากับ Tx/Rx ของ *Arduino*, จอ LCD ต่อในโหมด 4-bit (ขา RS, EN, D4-D7 ต่อเข้ากับ D13, D12, D11, D10, D9, D8 ของ *Arduino*), และปุ่มกดหลายตัวต่อเข้ากับอินพุตสำหรับผู้สมัคร CAN1, CAN2, CAN3 (ในภาพตัวอย่างมี 3 ผู้สมัคร) พร้อมทั้งปุ่มสำหรับฟังก์ชันอื่นๆ เช่น *Enroll*, *Delete* เป็นต้น (ต่อเข้ากับ A0-A4) ⁶ ในการใช้งานจริงสามารถเพิ่มจำนวนปุ่มผู้สมัครได้ถึง 8 ปุ่มตามที่ระบบต้องการ

หมายเหตุ: หากใช้ **ESP32** แทน *Arduino* ในการต่อวงจรจริง ขา I/O บน ESP32 จะต่างออกไป แต่หลักการต่อเหมือนกัน (RFID ใช้ SPI, เซ็นเซอร์ลายนิ้วมือใช้ UART, จอ LCD ใช้ I2C หรือ 4-bit parallel, ปุ่มกดต่อเข้า GPIO ทั่วไป) ต้องระมัดระวังเรื่องแรงดันไฟ: ESP32 ทำงานที่ระดับตรรกะ 3.3V ดังนั้นโมดูลทั้งหมดควรใช้ 3.3V หรือต่อวงจรแปลงระดับแรงดัน (logic level shifter) ถ้าจำเป็น (เช่น จอ LCD 16x2 ถ้าไม่มี I2C module อาจต้องใช้ 5V แต่ว่าสัญญาณจาก ESP32 3.3V ก็มักเพียงพอในการขับให้ LCD รับรู้เป็น High ได้)

เมื่อฮาร์ดแวร์ทุกส่วนต่อครบถ้วนดังผังวงจรแล้ว ก็พร้อมสำหรับการเขียนโปรแกรมควบคุมส่วนต่าง ๆ และดำเนินการตามขั้นตอนการลงคะแนนเสียง ซึ่งจะอธิบายในลำดับถัดไป

ผังงานของระบบ

ด้านล่างเป็นผังงาน (Flowchart) โดยสรุปขั้นตอนการทำงานของเครื่องลงคะแนนเสียงอัจฉริยะ ตั้งแต่การเริ่มต้นระบบไปจนถึงจบการลงคะแนนในแต่ละคน

ผังงานแสดงลำดับกระบวนการของระบบลงคะแนนเสียง: เริ่มจากรอให้ผู้ใช้สแกนบัตร *RFID* และลายนิ้วมือเพื่อยืนยันสิทธิ์ หากตรวจสอบผ่านและยังไม่เคยลงคะแนนมาก่อน ระบบจะอนุญาตให้ลงคะแนนได้ แต่หากลายนิ้วมือไม่ตรงกับข้อมูลที่ลงทะเบียนไว้ หรือผู้ใช้คนเดิมพยายามลงคะแนนซ้ำ ระบบจะปฏิเสธการลงคะแนน (ยกเลิกกระบวนการ) ² จากนั้นจึงวนกลับไปรอผู้ใช้คนถัดไป

จากผังงานจะเห็นว่าระบบมีเงื่อนไขที่สำคัญคือ **การตรวจสอบสิทธิ์** และ **การตรวจจบการลงคะแนนซ้ำ** เพื่อรักษาความถูกต้องของการเลือกตั้ง กระบวนการละเอียดจะขยายความในหัวข้อถัดไป

ขั้นตอนการลงคะแนน (Step-by-Step Workflow)

เพื่อให้เข้าใจการทำงานของระบบอย่างเป็นลำดับต่อเนื่อง ขอนำเสนอขั้นตอนการลงคะแนนของผู้ใช้หนึ่งคน ดังนี้:

- ผู้ใช้แสดงตัวตนด้วยบัตร RFID:** ผู้ลงคะแนนนำบัตรประจำตัว (ที่มีแท็ก RFID) และที่เครื่องอ่าน RFID RC522 ที่หน้าตู้ เครื่องอ่านจะตรวจจบบัตรและอ่านรหัสประจำตัว (UID) ออกมาได้สำเร็จ (บัตร RFID ทุกใบจะมี UID ที่ไม่ซ้ำกันเพื่อระบุผู้ถือบัตรแต่ละคน) ⁸
- ตรวจสอบ UID กับฐานข้อมูลผู้มีสิทธิ์:** ไมโครคอนโทรลเลอร์จะนำ UID ที่อ่านได้ไปค้นหาในรายชื่อผู้มีสิทธิ์ในหน่วยความจำ/ฐานข้อมูล หากไม่พบหมายเลขบัตรนี้ในระบบ แสดงว่าเป็นบัตรที่ไม่มีสิทธิ์หรือไม่ถูกต้อง ระบบจะแสดงผลบน LCD ว่า “ไม่มีสิทธิ์” หรือ “Unauthorized” และกระบวนการจะสิ้นสุดลงทันทีสำหรับบุคคลนั้น
- ยืนยันตัวตนด้วยลายนิ้วมือ:** หาก UID บัตรตรงกับรายชื่อผู้มีสิทธิ์ ระบบจะแสดงข้อความบน LCD ให้ผู้ใช้ “วางนิ้วเพื่อตรวจสอบ” จากนั้นผู้ใช้ต้องวางนิ้วที่ลงทะเบียนไว้บนเซ็นเซอร์ลายนิ้วมือ AS608 ระบบจะสแกนลายนิ้วมือและเทียบกับข้อมูลลายนิ้วมือในฐานข้อมูลที่ผูกกับหมายเลขบัตร RFID นั้นๆ เพื่อยืนยันว่าบุคคลที่ถือบัตรคือเจ้าของสิทธิ์ตัวจริง (การใช้สองปัจจัยนี้ช่วยรับประกันว่าบัตรเป็นของเจ้าของที่แท้จริง ไม่ใช่หยิบยืมผู้อื่นมา) ⁹
- ตรวจสอบสถานะการลงคะแนน:** หากลายนิ้วมือตรงกับข้อมูลในระบบ (ผ่านการยืนยันตัวบุคคล) โปรแกรมจะตรวจเช็คต่อว่าผู้มีสิทธิ์รายนี้เคยลงคะแนนไปแล้วหรือยังโดยการดูสถานะ “hasVoted” ในฐานข้อมูลของบุคคลนั้นๆ – ถ้าพบว่าเคยลงคะแนนแล้ว ระบบจะปฏิเสธการลงคะแนนซ้ำทันที จะแจ้งเตือนบนหน้าจอว่า “คุณได้ลงคะแนนแล้ว” และสิ้นสุด

กระบวนการสำหรับผู้ใช้นี้ (มาตรการนี้ทำให้หนึ่งคนลงคะแนนได้เพียงหนึ่งครั้งตามหลักการ *one person, one vote*

2)

5. **เลือกรูปแบบการลงคะแนน (Single/Multi):** กรณีผู้ใช้ยังไม่เคยลงคะแนน ระบบจะเข้าสู่โหมดลงคะแนน โดยจะแสดงข้อความให้เลือกผู้สมัครบนจอ LCD (เช่น “เลือกผู้สมัคร”) โหมดการลงคะแนนขึ้นอยู่กับค่าของระบบในขณะนั้น:
6. โหมดเลือกคนเดียว: ระบบจะอนุญาตให้กดปุ่มเลือกผู้สมัครได้เพียงคนเดียว เมื่อผู้ใช้กดปุ่มหมายเลขผู้สมัคร (เช่น ปุ่มหมายเลข 3 เพื่อเลือกผู้สมัครหมายเลข 3) ระบบจะบันทึกตัวเลือกนั้นแล้วถือว่าลงคะแนนเสร็จสิ้น
7. โหมดเลือกหลายคน: ระบบจะอนุญาตให้เลือกผู้สมัครได้หลายคนภายในการลงคะแนนครั้งเดียว (เช่น เลือกได้สูงสุด 5 คน) กรณีนี้ผู้ใช้สามารถกดปุ่มผู้สมัครได้ทีละคน ระบบจะนับจำนวนที่เลือกไป และแสดงผลจำนวนที่เลือกแล้ว/จำนวนสูงสุดบน LCD (เช่น “เลือกแล้ว 2/5”) ผู้ใช้จะกดจนครบตามจำนวนที่อนุญาต ระบบจะไม่อนุญาตให้เลือกเกินที่กำหนด (ถ้าครบ 5 คนแล้ว ปุ่มที่กดเพิ่มเติมจะไม่มีผล หรืออาจต้องบีบปุ่ม “ยืนยัน” แยกต่างหากเพื่อสิ้นสุดการเลือก)
8. **บันทึกผลโหวต:** เมื่อผู้ใช้เลือกผู้สมัครครบตามที่ระบบกำหนดในโหมดนั้นๆ แล้ว ไมโครคอนโทรลเลอร์จะทำการบันทึกคะแนนเสียงลงในตัวแปรหรือหน่วยความจำที่เก็บ **ตัวนับคะแนน** ของผู้สมัครแต่ละคน โดยจะเพิ่มค่า +1 ให้แก่ผู้สมัครที่ถูกเลือก (สำหรับโหมดเลือกหลายคน ผู้สมัครแต่ละคนที่ถูกเลือกจะได้ +1 ในตัวนับคะแนนของตน)
9. **อัปเดตสถานะผู้ใช้:** ระบบทำเครื่องหมายในฐานข้อมูลว่าผู้ใช้คนนี้ “ลงคะแนนแล้ว” (เช่น ตั้งค่าตัวแปร `hasVoted = true` สำหรับผู้ใช้นี้) เพื่อป้องกันไม่ให้เขากลับมาลงคะแนนซ้ำอีกในภายหลัง 2
10. **ส่งข้อมูลผลการเลือกตั้งไปยังเซิร์ฟเวอร์:** หลังการบันทึกคะแนนในเครื่องแล้ว ไมโครคอนโทรลเลอร์ (ESP32) จะทำการส่งข้อมูลผลการลงคะแนนไปยังคอมพิวเตอร์แม่ข่าย (Server) ผ่านทาง Wi-Fi ทั้งนี้ ข้อมูลที่ส่งอาจเป็นรายละเอียดการโหวตครั้งนี้หรือยอดรวมคะแนนล่าสุดของทุกผู้สมัครก็ได้ ข้อมูลจะถูกส่งด้วยรูปแบบที่ตกลงกันระหว่างไมโครคอนโทรลเลอร์กับโปรแกรมฝั่งคอมพิวเตอร์ (อาจใช้ HTTP request, WebSocket หรือ MQTT แล้วแต่การออกแบบในที่นี้จะยกตัวอย่างการส่งผ่าน HTTP) เมื่อเซิร์ฟเวอร์ได้รับข้อมูลแล้ว จะนำไปปรับปรุงแดชบอร์ดแสดงผลคะแนนแบบเรียลไทม์บนหน้าจอคอมพิวเตอร์
11. **การพิมพ์ยืนยัน (ถ้ามี):** หากระบบมีการต่อเครื่องพิมพ์ความร้อนไว้ ไมโครคอนโทรลเลอร์จะสั่งพิมพ์ใบเสร็จยืนยันการลงคะแนนให้แก่ผู้ใช้ เช่น พิมพ์ข้อความว่า “การลงคะแนนเสร็จสมบูรณ์”, รหัสของผู้ใช้ (หรือหมายเลขบัตร) และเวลาที่ลงคะแนน เป็นต้น ทั้งนี้เพื่อเป็นหลักฐานให้ผู้ลงคะแนนสามารถตรวจสอบความถูกต้องในภายหลัง (ข้อความในใบเสร็จจะไม่ระบุชื่อผู้สมัครที่เลือก เพื่อรักษาความลับของบัตรเลือกตั้ง)
12. **สิ้นสุดและพร้อมรับคนถัดไป:** ระบบแสดงข้อความ “ลงคะแนนสำเร็จ” หรือข้อความขอบคุณบน LCD สักครู่ จากนั้นจะรีเซ็ตกลับไปสู่สถานะพร้อมใช้งานสำหรับผู้มีสิทธิ์คนถัดไป (กลับไปขึ้นตอนรอบัตร RFID) โดยหากมีไฟ LED สถานะก็อาจกระพริบแจ้งเตือนสั้น ๆ ร่วมด้วยว่ากระบวนการสำหรับคนนี้เสร็จสิ้นแล้ว

ขั้นตอนข้างต้นจะเกิดขึ้นซ้ำ ๆ ทุกครั้งที่ผู้มีมาใช้สิทธิ์ลงคะแนนจนกว่าจะครบทุกคนหรือจนกว่าจะสิ้นสุดเวลาลงคะแนน ข้อมูลผลการลงคะแนนทุกครั้งจะถูกบันทึกไว้ทั้งในตัวเครื่องและส่งไปยังระบบศูนย์กลางตลอดเวลาเพื่อความโปร่งใสและปลอดภัย

การเตรียมฐานข้อมูลผู้มีสิทธิ์

ระบบนี้ต้องมี **ฐานข้อมูลรายชื่อผู้มีสิทธิ์** ที่ประกอบด้วยข้อมูลสองส่วนที่เชื่อมโยงกัน ได้แก่ **รหัสบัตร RFID** และ **เทมเพลตลายนิ้วมือ** ของผู้มีสิทธิ์แต่ละคน การเตรียมฐานข้อมูลนี้ควรทำเป็นขั้นตอนก่อนวันใช้งานจริง (pre-enrollment) โดยอาจดำเนินการดังนี้:

- **การลงทะเบียนบัตร RFID:** แจกจ่ายบัตร RFID ให้แก่ผู้มีสิทธิ์แต่ละคน (เช่น นักศึกษาแต่ละคนในกรณีเลือกตั้งองค์การนักศึกษา) จากนั้นอ่าน UID ของบัตรแต่ละใบและบันทึกลงในระบบพร้อมเชื่อมโยงกับข้อมูลบุคคล (เช่น ชื่อ หรือรหัสนักศึกษา เพื่อทราบว่า UID ใดเป็นของใคร)
- **การเก็บลายนิ้วมือ:** ใช้เซ็นเซอร์ลายนิ้วมือ AS608 เชื่อมต่อกับ Arduino/ESP32 รันโปรแกรม enrollment (มีอยู่ในไลบรารีของ Adafruit) เพื่อบันทึกลายนิ้วมือของผู้มีสิทธิ์แต่ละคนลงในหน่วยความจำของโมดูล โดยจะกำหนด **รหัสประจำลายนิ้วมือ (Finger ID)** ให้แต่ละคน (เช่น 001, 002, ... ตามลำดับ) ในขั้นตอนนี้ผู้ลงทะเบียนจะต้องวางนิ้ว 2-3 ครั้งตามที่ระบบแจ้งเพื่อบันทึกลายนิ้วมือให้คมชัดครบถ้วน เมื่อบันทึกแล้ว เทมเพลตลายนิ้วมือจะถูกจัดเก็บในโมดูล AS608 และสามารถค้นหาหรือเปรียบเทียบได้ภายหลัง
- **เชื่อมโยงข้อมูลสองส่วน:** ในโปรแกรมของไมโครคอนโทรลเลอร์ต้องมี **โครงสร้างข้อมูล** สำหรับเก็บรายชื่อผู้มีสิทธิ์ อาจทำเป็นอาร์เรย์ของโครงสร้าง (struct) ที่มีฟิลด์เก็บ UID ของบัตร (ความยาว 4 ไบต์สำหรับบัตร Mifare 1K ทั่วไป), หมายเลข ID ลายนิ้วมือที่บันทึกไว้ในเซ็นเซอร์ และสถานะการลงคะแนน (ยังไม่ได้โหวต/โหวตแล้ว) ดังตัวอย่าง:

```

struct Voter {
    uint8_t rfidUid[4];
    uint16_t fingerId;
    bool hasVoted;
};

Voter votersDB[] = {
    {{0xDE,0xAD,0xBE,0xEF}, 1, false}, // ตัวอย่าง: UID = DE AD BE EF, Finger ID = 1
    {{0xA1,0xB2,0xC3,0xD4}, 2, false}, // ตัวอย่าง: UID = A1 B2 C3 D4, Finger ID = 2
    // ... (เพิ่มตามจำนวนผู้มีสิทธิ์)
};

int totalVoters = sizeof(votersDB)/sizeof(votersDB[0]);

```

ในตัวอย่างข้างต้น เราเตรียมข้อมูลผู้มีสิทธิ์ 2 คน (ในทางปฏิบัติอาจมีหลายสิบหรือหลักร้อยคนก็ได้) โดยคนแรกมี UID ของบัตร RFID = DEADBEEF (เขียนในรูป hexadecimal) และมีลายนิ้วมือเก็บอยู่ในเซ็นเซอร์ที่ตำแหน่ง ID หมายเลข 1 ส่วนคนที่สอง UID = A1B2C3D4, Finger ID = 2 เป็นต้น ค่า `hasVoted` เริ่มต้นเป็น `false` แปลว่ายังไม่ได้ใช้สิทธิ์

- **นำเข้าข้อมูลสู่หน่วยความจำ:** ในที่นี้เลือกเก็บฐานข้อมูลในหน่วยความจำของไมโครคอนโทรลเลอร์เพื่อความง่าย (เนื่องจากจำนวนผู้มีสิทธิ์ไม่มากนัก) แต่หากต้องการขยาย สามารถใช้หน่วยความจำเสริม (เช่น EEPROM, microSD) หรือเชื่อมต่อกับฐานข้อมูลภายนอก (ผ่าน Wi-Fi ไปยังเซิร์ฟเวอร์กลาง) ก็ได้ตามความเหมาะสม ² ระบบควรโหลดรายชื่อผู้มีสิทธิ์นี้ในขั้นตอนเริ่มต้น (`setup()`) เพื่อเตรียมพร้อมสำหรับการตรวจสอบขณะลงคะแนนจริง
- **รักษาความปลอดภัยของข้อมูล:** ควรจำกัดการเข้าถึง/แก้ไขฐานข้อมูลผู้มีสิทธิ์เฉพาะผู้ดูแลระบบเท่านั้น เพื่อป้องกันการเพิ่มชื่อหรือลบชื่อโดยไม่ได้รับอนุญาต นอกจากนี้ เมื่อการเลือกตั้งสิ้นสุด ควรล้างข้อมูลลายนิ้วมือออกจากเซ็นเซอร์เพื่อลดความเสี่ยงด้านความเป็นส่วนตัวของข้อมูลชีวมิติ

การเตรียมฐานข้อมูลที่ถูกต้องและเชื่อถือได้เป็นรากฐานสำคัญของระบบนี้ เพราะหากข้อมูลผิดพลาด (เช่น ลายนิ้วมือไม่ตรงกับบัตร, รายชื่อขาดตก ฯลฯ) จะทำให้ผู้มีสิทธิ์บางคนไม่สามารถลงคะแนนหรือเกิดปัญหาในขั้นตอนยืนยันตัวตนได้

โปรแกรมบนไมโครคอนโทรลเลอร์ (Arduino/ESP32)

ต่อไปจะเป็นส่วนของซอฟต์แวร์ที่รันบนไมโครคอนโทรลเลอร์ เพื่อควบคุมอุปกรณ์และดำเนินการตามขั้นตอนที่ได้อธิบายไปแล้ว โค้ดต่อไปนี้จะจัดทำขึ้นสำหรับแพลตฟอร์ม **ESP32** (สามารถปรับใช้กับ Arduino ได้โดยแก้ไขไลบรารี Wi-Fi และพอร์ต Serial ตามความเหมาะสม) โค้ดนี้ครอบคลุมการทำงานตั้งแต่การอ่าน RFID, สแกนลายนิ้วมือ, รับอินพุตจากปุ่ม, ส่งข้อมูลผ่าน Wi-Fi และควบคุม LCD รวมถึงการพิมพ์ใบเสร็จ (ในรูปแบบความคิดเห็น) โดยมีคำอธิบายเป็นตอน ๆ ภายใต้อีก:

```

#include <SPI.h>
#include <MFRC522.h>           // ไลบรารีสำหรับโมดูล RFID RC522
#include <WiFi.h>               // ไลบรารี Wi-Fi สำหรับ ESP32
#include <HTTPClient.h>         // ไลบรารีสำหรับส่ง HTTP requests
#include <Adafruit_Fingerprint.h> // ไลบรารีเซ็นเซอร์ลายนิ้วมือ (Adafruit)
#include <LiquidCrystal_I2C.h> // ไลบรารีจอ LCD ผ่าน I2C

// กำหนดขาสำหรับ RC522 (ใช้ชุด SPI พื้นฐานของ ESP32)
#define RFID_SDA 5           // SS (SDA) ของ RC522 ต่อกับ GPIO5

```

```

#define RFID_RST 21 // RST ของ RC522 ต่อกับ GPIO21
MFRC522 mfrc522(RFID_SDA, RFID_RST); // สร้างอ็อบเจกต์ RC522

// กำหนด UART สำหรับเซ็นเซอร์ลายนิ้วมือ (ใช้ UART2 บน ESP32)
HardwareSerial FingerSerial(2);
// หมายเหตุ: ต้องกำหนดขา RX, TX ของ UART2 ตอน begin (ค่าเริ่ม RX2=16, TX2=17 บนบางบอร์ด)
Adafruit_Fingerprint finger(&FingerSerial);

// กำหนดจอ LCD I2C (ที่อยู่ 0x27 ขนาด 16x2)
LiquidCrystal_I2C lcd(0x27, 16, 2);

// กำหนดขาปุ่มผู้สมัคร (ตัวอย่างใช้ 8 ปุ่มกับขา GPIO ตามลำดับ)
int candidateButtons[8] = {13, 12, 14, 27, 26, 25, 33, 32};
int numCandidates = 8;

// กำหนดโหมดการลงคะแนน (false = โหมดเลือกคนเดียว, true = โหมดเลือกหลายคน)
bool multiMode = false;
int maxSelections = 5; // จำนวนสูงสุดที่เลือกได้ในโหมดหลายคน (เริ่มต้น 5)

// โครงสร้างและฐานข้อมูลผู้มีสิทธิ (ตัวอย่าง)
struct Voter {
    uint8_t rfidUid[4];
    uint16_t fingerId;
    bool hasVoted;
} votersDB[] = {
    {{0xDE, 0xAD, 0xBE, 0xEF}, 1, false},
    {{0xAA, 0xBB, 0xCC, 0xDD}, 2, false}
    // สามารถเพิ่มรายการตามจำนวนผู้มีสิทธิ
};
int totalVoters = sizeof(votersDB) / sizeof(votersDB[0]);

// ตัวแปรเก็บผลคะแนนของผู้สมัคร (เริ่มต้น 0 ทุกคน)
int voteCounts[8] = {0};

// ข้อมูลการเชื่อมต่อ Wi-Fi และเซิร์ฟเวอร์ (แก้ไขตามสภาพแวดล้อมจริง)
const char* ssid = "YourWiFiSSID";
const char* password = "YourWiFiPassword";
const char* serverURL = "http://192.168.1.100:5000/update"; // URL สำหรับส่งผล
คะแนน (ตัวอย่าง)

// ฟังก์ชันช่วยเหลือ: เปรียบเทียบ UID ที่อ่านได้กับ UID ในฐานข้อมูล (คืน -1 ถ้าไม่พบ)
int findVoterByUID(uint8_t *uid, byte uidSize) {
    for(int i = 0; i < totalVoters; i++) {
        bool match = true;
        for(byte j = 0; j < uidSize; j++) {
            if(votersDB[i].rfidUid[j] != uid[j]) {
                match = false;
                break;
            }
        }
    }
}

```

```

    if(match) return i;
}
return -1;
}

void setup() {
    Serial.begin(115200);    // เริ่มต้น Serial สำหรับ debug
    lcd.init(); lcd.backlight(); // เริ่มต้นจอ LCD
    lcd.print("Smart Voting"); // แสดงข้อความเปิดระบบ
    // เริ่มต้นโมดูล RFID
    SPI.begin();
    mfrc522.PCD_Init();
    // เริ่มต้นโมดูลลายนิ้วมือ
    FingerSerial.begin(57600, SERIAL_8N1, 16, 17); // กำหนด UART2 RX=GPI016,
    TX=GPI017
    finger.begin(57600);
    if(finger.verifyPassword()) {
        Serial.println("Fingerprint sensor OK");
    } else {
        Serial.println("Fingerprint sensor NOT found");
    }
    // กำหนดขา I/O ของปุ่มเป็น input pullup
    for(int i = 0; i < numCandidates; i++) {
        pinMode(candidateButtons[i], INPUT_PULLUP);
    }
    // เชื่อมต่อ Wi-Fi
    WiFi.begin(ssid, password);
    lcd.setCursor(0,1); lcd.print("WiFi Connecting");
    while(WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\nWiFi connected");
    lcd.clear();
}

void loop() {
    // 1. รอการสแกนบัตร RFID
    if(!mfrc522.PICC_IsNewCardPresent() || !mfrc522.PICC_ReadCardSerial()) {
        return; // วนลูปไปเรื่อย ๆ จนกว่าจะมีบัตรเข้ามาใหม่
    }
    // อ่าน UID ของบัตรที่เพิ่งสแกนได้
    uint8_t *uid = mfrc522.uid.uidByte;
    byte uidSize = mfrc522.uid.size;
    Serial.print("Card UID: ");
    for(byte i = 0; i < uidSize; i++) {
        Serial.printf("%02X ", uid[i]);
    }
    Serial.println();
    // 2. ตรวจสอบ UID กับฐานข้อมูลผู้มีสิทธิ์

```

```

int voterIndex = findVoterByUID(uid, uidSize);
if(voterIndex == -1) {
    lcd.clear(); lcd.print("Unauthorized!"); // แสดงว่าไม่พบในฐานข้อมูล
    Serial.println("Unauthorized card.");
    delay(3000);
    mfrc522.PICC_HaltA(); // หยุดการอ่านบัตร
    return;
}
// 3. ตรวจสอบว่าผู้ใช้คนนี้เคยโหวตหรือยัง
if(votersDB[voterIndex].hasVoted) {
    lcd.clear(); lcd.print("Already Voted"); // แจ้งว่าโหวตแล้ว
    Serial.println("Voter already voted.");
    delay(3000);
    mfrc522.PICC_HaltA();
    return;
}
// 4. ขอให้สแกนลายนิ้วมือ
lcd.clear(); lcd.print("Scan Finger...");
Serial.println("Place finger for scanning...");
// พยายามอ่านลายนิ้วมือ (ให้โอกาส 3 ครั้ง)
int fingerprintResult = -1;
for(int attempt = 0; attempt < 3; attempt++) {
    if(finger.getImage() == FINGERPRINT_OK) {
        if(finger.image2Tz() == FINGERPRINT_OK) {
            // ค้นหารอยนิ้วมือในฐานข้อมูลของโมดูล
            if(finger.fingerSearch() == FINGERPRINT_OK) {
                fingerprintResult = finger.fingerID; // ได้ ID ลายนิ้วมือที่ตรง
                break;
            }
        }
    }
}
delay(1000);
}
// ตรวจสอบผลการสแกน
if(fingerprintResult != votersDB[voterIndex].fingerId) {
    lcd.clear(); lcd.print("Fingerprint Err"); // ลายนิ้วมือไม่ตรง
    Serial.println("Fingerprint mismatch or not found.");
    delay(3000);
    mfrc522.PICC_HaltA();
    return;
}
// 5. ผ่านการยืนยันตัวตน แสดงข้อความให้ลงคะแนน
lcd.clear();
lcd.print("Select Candidate");
if(multiMode) {
    lcd.setCursor(0,1);
    lcd.print("Choose up to ");
    lcd.print(maxSelections);
}
Serial.println("Authentication successful, ready for voting.");

```



```

// 6. รับอินพุตการลงคะแนนจากปุ่ม
if(!multiMode) {
    // --- โหมดเลือกได้คนเดียว ---
    int chosenIndex = -1;
    // รอจนกว่าจะมีการกดปุ่มใดปุ่มหนึ่ง
    while(chosenIndex == -1) {
        for(int i = 0; i < numCandidates; i++) {
            if(digitalRead(candidateButtons[i]) == LOW) { // ปุ่มถูกกด (เนื่องจากใช้
INPUT_PULLUP, LOW = กด)
                chosenIndex = i;
                // หน่วงรอจนปล่อยปุ่ม เพื่อ debounce
                while(digitalRead(candidateButtons[i]) == LOW) { delay(10); }
                break;
            }
        }
    }
    // เพิ่มคะแนนให้ผู้สมัครที่เลือก
    voteCounts[chosenIndex] += 1;
    Serial.printf("Voted for candidate %d\n", chosenIndex+1);
} else {
    // --- โหมดเลือกได้หลายคน ---
    bool selectedFlags[8] = {false};
    int countSelected = 0;
    // วนรับการกดจนกว่าจะครบจำนวนที่กำหนด
    while(countSelected < maxSelections) {
        for(int i = 0; i < numCandidates; i++) {
            if(digitalRead(candidateButtons[i]) == LOW && !selectedFlags[i]) {
                // หากปุ่ม i ถูกกดและยังไม่เคยถูกเลือกมาก่อน
                selectedFlags[i] = true;
                countSelected++;
                voteCounts[i] += 1;
                Serial.printf("Selected candidate %d\n", i+1);
                // อัปเดต LCD แจ้งจำนวนที่เลือกแล้ว
                lcd.clear();
                lcd.print("Selected #");
                lcd.print(i+1);
                lcd.setCursor(0,1);
                lcd.print("Count: ");
                lcd.print(countSelected);
                lcd.print("/");
                lcd.print(maxSelections);
                // รอจนปล่อยปุ่ม
                while(digitalRead(candidateButtons[i]) == LOW) { delay(10); }
                if(countSelected >= maxSelections) break;
            }
        }
    }
}
// 7. ปรับสถานะว่าโหวตแล้ว
votersDB[voterIndex].hasVoted = true;

```

```

// 8. ส่งข้อมูลผลโหวตไปยัง Server ผ่าน Wi-Fi
if(WiFi.status() == WL_CONNECTED) {
  HTTPClient http;
  // สร้าง URL สำหรับส่งคำขอ (ส่งคะแนนทั้งหมดไปด้วย)
  String url = String(serverURL) + "?";
  for(int i = 0; i < numCandidates; i++) {
    url += "c" + String(i+1) + "=" + String(voteCounts[i]);
    if(i < numCandidates-1) url += "&";
  }
  http.begin(url);
  int httpCode = http.GET(); // ส่ง HTTP GET request
  if(httpCode == 200) {
    Serial.println("Results updated to server.");
  } else {
    Serial.printf("Failed to send (HTTP %d)\n", httpCode);
  }
  http.end();
}

// 9. พิมพ์ใบเสร็จยืนยัน (ถ้าเชื่อมต่อเครื่องพิมพ์ได้)
// *โค้ดตัวอย่าง:*
// Serial2.begin(19200); // เชื่อมต่อเครื่องพิมพ์ที่ Serial2 (ต้องทราบว่าต่อ TX, RX ขาไหน)
// Serial2.println("Vote recorded successfully!");
// Serial2.print("Voter ID: ");
Serial2.println(votersDB[voterIndex].fingerId);
// Serial2.println("Thank you for voting.");
// Serial2.write(0x0C); // สั่งตัดกระดาษ (ขึ้นอยู่กับรุ่นเครื่องพิมพ์)

// 10. แจ้งบน LCD และเตรียมพร้อมสำหรับคนถัดไป
lcd.clear();
lcd.print("Vote Recorded!");
delay(2000);
mfrc522.PICC_HaltA(); // จบการทำงานกับบัตรใบนี้
}

```

คำอธิบายโค้ด: โค้ดด้านบนเริ่มต้นด้วยการ include ไลบรารีต่าง ๆ ที่จำเป็น จากนั้นตั้งค่าขาอุปกรณ์และตัวแปรต่าง ๆ ใน `setup()` เช่น เริ่มต้น RFID, Wi-Fi, เซ็นเซอร์ลายนิ้วมือ และ LCD ใน `loop()` จะเข้าสู่ขั้นตอนวนซ้ำที่คอยตรวจสอบการทาบบัตร RFID (ถ้าไม่มีบัตรเข้ามาใหม่ก็วนรอไปเรื่อย ๆ) เมื่อได้ UID ของบัตรแล้วจะเรียกฟังก์ชัน `findVoterByUID()` เพื่อค้นหาในฐานข้อมูล ถ้าไม่พบก็จะขึ้น "Unauthorized" บนหน้าจอ LCD และยกเลิกกระบวนการ ถ้าพบก็จะตรวจสอบสถานะ `hasVoted` เพื่อป้องกันการโหวตซ้ำ จากนั้นสั่งให้ผู้ใช้นิ้วสแกนลายนิ้วมือ เปรียบเทียบกับ ID ที่ลงทะเบียนไว้ หากไม่ตรงก็ขึ้น "Fingerprint Err" แล้วจบกระบวนการ ถ้าตรงก็ไปยังขั้นตอนการลงคะแนนโดยให้กดปุ่มเลือกผู้สมัคร โค้ดได้แยกกรณี `multiMode` เพื่อรองรับทั้งสองโหมดตามที่ระบุ เมื่อบันทึกตัวเลือกลงในตัวแปร `voteCounts` แล้ว ก็ทำการส่งข้อมูลผ่าน Wi-Fi โดยในตัวอย่างใช้ HTTP GET ที่ใส่พารามิเตอร์คะแนนของผู้สมัครแต่ละคน (c1..c8) ส่งไปยัง URL ของเซิร์ฟเวอร์ จากนั้น (ถ้ามีเครื่องพิมพ์) จะสั่งพิมพ์ใบเสร็จยืนยัน สุดท้ายแสดงผล "Vote Recorded!" ให้ผู้ใช้ทราบและ reset การอ่านบัตร RFID เพื่อพร้อมรับคนถัดไป

การปรับใช้บน Arduino UNO: หากใช้ Arduino UNO/MEGA แทน ESP32 ส่วนโค้ดหลักจะคล้ายกัน ยกเว้นการเชื่อมต่อ Wi-Fi ที่ต้องพึ่งพาโมดูลภายนอก (เช่น ESP8266) หรือการสื่อสารทางอนุกรมกับคอมพิวเตอร์แทน (Serial ผ่าน USB) รวมถึงการใช้ไลบรารี WiFi ของ Arduino หากใช้ Arduino Uno WiFi. นอกจากนี้ Arduino UNO มีข้อจำกัดที่ Serial อย่างเดียว

อาจต้องใช้ SoftwareSerial (เช่น ขา D2,D3) สำหรับเซ็นเซอร์ลายนิ้วมือ เพื่อไม่ให้ชนกับ Serial หลักที่ใช้ติดต่อกับคอมพิวเตอร์

การส่งข้อมูลผ่าน Wi-Fi และแดชบอร์ดแสดงผล

ในการส่งข้อมูลผลการลงคะแนนไปยังคอมพิวเตอร์แบบเรียลไทม์ เราสามารถเลือกใช้วิธีการสื่อสารได้หลายรูปแบบ เช่น HTTP, WebSocket หรือ MQTT สำหรับโครงงานนี้จะยกตัวอย่างการใช้ **HTTP Request** เนื่องจากเข้าใจง่าย โดยหลังจากที่ไมโครคอนโทรลเลอร์ (ESP32) เชื่อมต่อเข้าเครือข่าย Wi-Fi เดียวกับเครื่องคอมพิวเตอร์แล้ว จะสามารถส่งคำร้องขอ HTTP ไปยัง **เซิร์ฟเวอร์เว็บ (Web Server)** ที่รันอยู่บนเครื่องคอมฯ ได้

ฝั่งคอมพิวเตอร์เราสามารถเขียนโปรแกรมเซิร์ฟเวอร์อย่างง่ายเพื่อรับข้อมูลและแสดงผลได้ อาจเลือกใช้ภาษา Python (ผ่านเฟรมเวิร์ก Flask) หรือ Node.js (ผ่านเฟรมเวิร์ก Express หรือ Socket.io) ทั้งสองแบบสามารถทำให้เกิดแดชบอร์ดแบบเรียลไทม์ได้ทั้งคู่ ในที่นี้จะยกตัวอย่างด้วย **Python Flask** ซึ่งจะรับค่าคะแนนที่ ESP32 ส่งมาแล้วแสดงผลบนหน้าเว็บ

ขั้นตอนการตั้งค่า: ให้เครื่องคอมพิวเตอร์ (หรือ ODROID) รันสคริปต์ Python เพื่อเป็นเซิร์ฟเวอร์รับข้อมูล ก่อนอื่นต้องติดตั้ง Flask (ด้วยคำสั่ง `pip install flask`) จากนั้นใช้โค้ดดังต่อไปนี้:

```
from flask import Flask, request

app = Flask(__name__)

# ตัวแปรเก็บคะแนนล่าสุดของผู้สมัคร (เริ่มต้น 0 ทุกคน 8 คน)
vote_counts = [0]*8

@app.route('/update')
def update_results():
    # ดึงค่าคะแนนจากพารามิเตอร์ URL (c1..c8)
    global vote_counts
    for i in range(1, 9):
        param = request.args.get(f'c{i}')
        if param is not None:
            vote_counts[i-1] = int(param)
    print("Updated counts:", vote_counts) # แสดงใน console (ตรวจสอบ)
    return "OK", 200

@app.route('/')
def dashboard():
    # สร้างหน้า HTML แสดงผลคะแนนปัจจุบัน
    html = "<h1>ผลการเลือกตั้ง (เรียลไทม์)</h1><ul>"
    for idx, count in enumerate(vote_counts, start=1):
        html += f"<li>ผู้สมัคร #{idx}: {count} votes</li>"
    html += "</ul>"
    return html

if __name__ == "__main__":
    # รันเซิร์ฟเวอร์ Flask บนโฮสต์เครื่อง (ให้ ESP32 ส่งเข้า IP เครื่องนี้)
    app.run(host='0.0.0.0', port=5000)
```

โค้ดด้านบนสร้างเซิร์ฟเวอร์เว็บที่มีสองเส้นทาง (route) คือ `/update` สำหรับรับข้อมูลอัปเดตผลคะแนน และ `/` สำหรับแสดงแดชบอร์ด ในฟังก์ชัน `update_results()` จะรับพารามิเตอร์ `c1` ถึง `c8` จาก URL (ซึ่ง ESP32 ส่งมา) แล้วอัปเดตค่าในอาร์เรย์ `vote_counts` จากนั้นส่งคำตอบ "OK" กลับไป ส่วนฟังก์ชัน `dashboard()` จะสร้าง HTML อย่างง่ายโดยแสดงรายชื่อผู้สมัคร (#1 ถึง #8) พร้อมจำนวนคะแนนที่สะสมจนถึงล่าสุด

ในการใช้งานจริง ให้รันสคริปต์นี้ (`python server.py`) บนเครื่องคอมพิวเตอร์ที่เชื่อมต่อเน็ตเวิร์กเดียวกับ ESP32 แล้วเปิดเว็บเบราว์เซอร์ไปที่ `http://<IP_เครื่อง>:5000/` (เช่น `http://192.168.1.100:5000/`) จะเห็นหน้าเว็บรายงานผลคะแนนแบบเรียลไทม์ ซึ่งทุกครั้งที่มีการลงคะแนนใหม่และ ESP32 ส่งข้อมูลมายัง `/update`, ข้อมูลบนหน้าแดชบอร์ด (ถ้ารีเฟรชหน้าเว็บ) จะแสดงตัวเลขที่เปลี่ยนไปทันที

ในการทำให้แดชบอร์ดอัปเดตแบบเรียลไทม์โดยไม่ต้องรีเฟรชหน้า อาจพัฒนาเพิ่มเติมโดยใช้เทคนิค **AJAX** หรือ **WebSocket** (เช่น ใช้ Flask-SocketIO หรือใน Node.js ใช้ Socket.io) เพื่อ push ข้อมูลไปที่หน้าเว็บอัตโนมัติเมื่อมีการโหวตใหม่ อย่างไรก็ตาม ในบริบทโครงงานนักเรียน การรีเฟรชหน้าเว็บหรือกดปุ่มแสดงผลใหม่หลังการลงคะแนนก็อาจเพียงพอแล้ว

หมายเหตุ: หากเลือกใช้ Node.js สามารถเขียนเซิร์ฟเวอร์ด้วย Express ที่รับ `req.query` คล้าย ๆ กัน แล้วใช้ Socket.io ส่งข้อมูลไปยังหน้าเว็บ HTML ที่ทำเป็นแดชบอร์ด ซึ่งจะให้เห็นผลทันทีโดยไม่ต้องรีเฟรช ส่วนการทำ UI สามารถใช้ HTML/JS สร้างเป็นกราฟหรือ bar chart ให้ดูเข้าใจง่ายขึ้น

ด้วยระบบเครือข่ายนี้ การสื่อสารข้อมูลผลโหวตจะเกิดขึ้นแบบเรียลไทม์จากเครื่องลงคะแนนไปยังศูนย์กลาง ซึ่งช่วยเพิ่มความโปร่งใส ผู้เกี่ยวข้องสามารถติดตามจำนวนคะแนนของผู้สมัครแต่ละคนได้ทันที ลดขั้นตอนการนับคะแนนด้วยมือและความผิดพลาดที่อาจเกิดขึ้นได้ ¹⁰ และหากต้องการขยายระบบให้รองรับหลายเครื่องลงคะแนน (หลายบูธ) ก็สามารถทำได้โดยให้แต่ละเครื่องส่งข้อมูลเข้ามายังเซิร์ฟเวอร์กลางเดียวกันเพื่อรวมคะแนน

ฟังก์ชันการพิมพ์ใบเสร็จยืนยัน

หนึ่งในฟีเจอร์เพิ่มเติมของระบบนี้คือการพิมพ์ใบยืนยันการลงคะแนน (Vote receipt) เมื่อผู้ใช้ลงคะแนนสำเร็จ โมดูลเครื่องพิมพ์ความร้อนที่ใช้ในโครงงานสามารถเป็นรุ่นที่รองรับ TTL Serial (เช่น เครื่องพิมพ์ความร้อนขนาดเล็กของ AdaFruit หรือยี่ห้ออื่นที่คล้ายกัน) ซึ่งมีขา TX, RX สำหรับรับส่งข้อมูลการพิมพ์ วิธีการต่อคือใช้ขา TX ของไมโครคอนโทรลเลอร์ต่อเข้า RX ของเครื่องพิมพ์ (ส่วน TX ของเครื่องพิมพ์สามารถต่อกลับเข้ากับ RX ของ MCU เพื่อรับสถานะได้ แต่ไม่จำเป็นสำหรับงานนี้) พร้อมทั้งจ่ายไฟเลี้ยงให้เครื่องพิมพ์ (ส่วนใหญ่ใช้ 5V 2A)

ในการใช้งานกับ Arduino/ESP32 สามารถใช้ไลบรารีเช่น **Adafruit Thermal Printer library** เพื่อสะดวกในการพิมพ์ข้อความและจัดรูปแบบ หรือจะส่งชุดคำสั่ง ESC/POS โดยตรงก็ได้ ในโค้ดตัวอย่างที่แสดงข้างต้น (ข้อ 9 ในโค้ด) มีการคอมเมนต์ให้เห็นแนวทางการพิมพ์อย่างง่ายโดยใช้ `Serial2` ส่งข้อความตัวอักษรไปยังเครื่องพิมพ์ (โดยต้องเปิด `Serial2` ด้วย baud rate ที่ตรงกับเครื่อง เช่น 19200 หรือ 9600 แล้วแต่รุ่น ¹¹ ¹²) ตัวอย่างเช่น:

```
Serial2.begin(19200);
Serial2.println("Vote recorded successfully!");
Serial2.println("Voter ID: 001");
Serial2.println("Thank you for voting.");
Serial2.write(0x0C); // คำสั่งตัดกระดาษ (พับกระดาษและตัด)
```

เมื่อเครื่องพิมพ์ได้รับข้อความเหล่านี้ก็จะพิมพ์ออกมาตามบรรทัด ตัวอย่าง output ที่ได้บนกระดาษอาจเป็น:

Vote recorded successfully!
Voter ID: 001
Thank you for voting.

จากนั้นตัวคำสั่ง 0x0C (หรือคำสั่งที่เครื่องพิมพ์แต่ละรุ่นกำหนด) จะสั่งให้เครื่องพิมพ์ตัดกระดาษอัตโนมัติ ผู้ลงคะแนนก็จะได้รับใบเสร็จนี้ออกมาจากเครื่องเพื่อเก็บไว้เป็นหลักฐาน

ข้อควรระวัง: เครื่องพิมพ์ความร้อนใช้กระแสค่อนข้างสูงช่วงพิมพ์ (โดยเฉพาะตอนตัดกระดาษและเลื่อนกระดาษ) จึงควรใช้แหล่งจ่าย 5V ที่จ่ายกระแสได้พอเพียง (เช่น อะแดปเตอร์ 5V 2A) แยกต่างหาก ไม่ดึงไฟจากบอร์ดไมโครคอนโทรลเลอร์โดยตรง เพื่อป้องกันปัญหาแรงดันตกหรือรีเซ็ตเครื่องไมโครคอนโทรลเลอร์

การพิมพ์ใบเสร็จนี้เป็นเพียงตัวเลือกเสริม แต่มีประโยชน์ในการเสริมความน่าเชื่อถือของระบบ ถ้านำระบบนี้ไปใช้ในการเลือกตั้งจริง (เช่น การเลือกตั้งสมาชิกสภาหรือองค์การนักศึกษา) ใบเสร็จเหล่านี้สามารถใช้ในการตรวจสอบภายหลังได้ว่าสิทธิของแต่ละคนถูกใช้อย่างถูกต้องหรือไม่ อย่างไรตาม ควรคำนึงถึงความลับในการลงคะแนนด้วย จึงไม่ควรพิมพ์รายละเอียดว่าโหวตให้ใคร ควรพิมพ์เพียงว่าผู้ใช้ ID ใดได้ลงคะแนนแล้ว เวลาใด เป็นต้น

การติดตั้งและโครงสร้างทางกายภาพ

ในส่วนของโครงสร้างตู้ลงคะแนน สามารถสร้างขึ้นจากวัสดุที่หาได้ง่าย เช่น กล่องกระดาษลัง แผ่นโฟมกระดาน หรืออะคริลิก ตามความคิดสร้างสรรค์ของผู้จัดทำ แนวคิดคือทำเป็นลักษณะคูหาลงคะแนนขนาดย่อมที่มีฝาปิดบางส่วนเพื่อความเป็นส่วนตัว โดยมีองค์ประกอบดังนี้:

- **แผงด้านหน้าสำหรับผู้ใช้:** เจาะช่องสำหรับหน้าจอ LCD ให้อยู่ในระดับสายตา และเจาะรูขนาดพอเหมาะสำหรับติดตั้งเซ็นเซอร์ลายนิ้วมือ (ควรยึดโมดูลลายนิ้วมือจากด้านในให้ผิวหน้าของเซ็นเซอร์เสมอกับผิวกล่อง), ติดตั้งเครื่องอ่าน RFID (โดยอาจนำแท็ก RFID มาแตะที่ด้านนอกกล่องในบริเวณที่มีเครื่องอ่านอยู่ภายใน เนื่องจากคลื่น RFID ทะลุวัสดุบางได้ หรือเจาะช่องและติดตั้งเครื่องอ่านโพลีออกไซด์ด้านนอกเลยก็ได้)
- **แผงปุ่มกด:** จัดวางปุ่มกดหมายเลขผู้สมัคร 1-8 เรียงตามลำดับ อาจติดตารางหรือพิมพ์สติกเกอร์หมายเลขกำกับไว้ชัดเจน ขนาดปุ่มควรใหญ่พอที่กดง่าย เว้นระยะห่างพอประมาณเพื่อไม่ให้กดผิด ผู้ใช้จะเลือกกดปุ่มเหล่านี้ตามผู้สมัครที่ต้องการโหวต
- **ไฟแสดงสถานะ (ถ้ามี):** อาจติด LED สีแดง/เขียวบนแผงเพื่อแสดงสถานะ “พร้อมใช้งาน” หรือ “ตรวจสอบผ่าน” เป็นต้น เช่น LED เขียวสว่างเมื่อลายนิ้วมือผ่านการยืนยัน และ LED แดงกระพริบเมื่อเกิดข้อผิดพลาด
- **ตำแหน่งเครื่องพิมพ์ (ถ้ามี):** เครื่องพิมพ์ความร้อนสามารถติดตั้งไว้ด้านข้างหรือด้านหน้าล่างของตู้ โดยเจาะช่องสำหรับให้กระดาษใบเสร็จเลื่อนออกมาได้สะดวก หลังเครื่องพิมพ์ควรเพื่อช่องให้สามารถเปลี่ยนม้วนกระดาษได้ง่าย
- **การจัดวางภายใน:** ติดตั้งบอร์ดไมโครคอนโทรลเลอร์บนแผ่นกระดานหรือพลาสติกภายในกล่องยึดให้แน่นหนา วางโมดูล RFID และสายอากาศในตำแหน่งที่ใกล้กับจุดที่ผู้ใช้จะแตะบัตร (เพื่อให้อ่านบัตรได้ง่าย สัญญาณไม่ถูกบัง) เดินสายไฟและสายสัญญาณต่าง ๆ ให้เป็นระเบียบ ใช้ที่รัดสายหรือเทปพันสายยึดเพื่อลดความยุ่งเหยิง
- **แหล่งจ่ายไฟ:** หากใช้แบตเตอรี่หรือ Power Bank ควรวางไว้ในกล่องด้วยและทำช่องสำหรับชาร์จ หรือถ้าใช้หม้อแปลง AC->DC ต่อเข้ากับปลั๊กไฟ ให้ทำรูให้สายไฟออกจากตัวกล่องได้อย่างปลอดภัย ไม่เกะกะทางเดิน

เมื่อประกอบทุกส่วนเข้าด้วยกัน จะได้ตู้ลงคะแนนเสียงอัจฉริยะที่มีลักษณะคล้ายเครื่องลงคะแนนอิเล็กทรอนิกส์ขนาดเล็ก ผู้ใช้งานจะเห็นหน้าจอข้อความ เห็นช่องให้ใส่แกนบัตรและลายนิ้วมือ และมีปุ่มเลือกผู้สมัครให้กดชัดเจน ตัวอย่างรูปลักษณะอาจคล้ายตู้ EVM ของงานประกวดโครงงานที่ทำจากวัสดุเรียบง่ายแต่ฟังก์ชันครบถ้วน

การออกแบบโครงสร้าง DIY เช่นนี้ช่วยลดต้นทุนและสามารถปรับแต่งรูปร่าง สี สัน ได้ตามต้องการ แต่ผู้จัดทำต้องคำนึงถึงความแข็งแรงในการใช้งานจริงด้วย (เช่น ปุ่มกดต้องติดแน่น, โมดูลต่าง ๆ ไม่โยกหรือหลุดง่าย) รวมถึงความปลอดภัย (ไม่มีสายไฟเปลือยหรือจุดที่ผู้ใช้จะโดนกระแส/บาดเจ็บ) เพื่อให้การสาธิตเป็นไปอย่างราบรื่นและมีอาชีพ

สรุปการทำงานของระบบ

เครื่องลงคะแนนเสียงอัจฉริยะด้วย Arduino/ESP32 นี้รวมข้อดีของระบบลงคะแนนอิเล็กทรอนิกส์เข้ากับมาตรการรักษาความปลอดภัยแบบสองชั้น ผลที่ได้คือระบบที่ผู้มีสิทธิ์ต้องยืนยันตัวตนด้วย บางสิ่งที่พวกเขาใช้ (บัตร RFID) และ บางสิ่งที่พวกเขามี (ลายนิ้วมือ) ก่อนจะลงคะแนนเสียง ซึ่งช่วยป้องกันการสวมสิทธิ์และการลงคะแนนซ้ำได้อย่างมีประสิทธิภาพ ¹ หลังการยืนยันระบบจะให้ผู้ใช้ลงคะแนนผ่านอินเทอร์เฟซที่เรียบง่าย (ปุ่มกดหมายเลขผู้สมัคร) ซึ่งผลลัพธ์จะถูกส่งไปนับรวมและแสดงผลแบบเรียลไทม์ ทำให้ผู้จัดการเลือกตั้งสามารถติดตามความคืบหน้าของการลงคะแนนได้ทันทีโดยไม่ต้องรอนับมือ นอกจากนี้ การพิมพ์ใบเสร็จยืนยันยังเพิ่มความโปร่งใส เพราะผู้ลงคะแนนสามารถตรวจสอบได้ว่าเสียงของตนถูกบันทึกแล้วจริงๆ

ระบบนี้ถูกออกแบบมาให้ใกล้เคียงการใช้งานจริงในระดับหนึ่ง แต่ยังคงความง่ายในการสร้างสำหรับนักศึกษา โดยใช้อุปกรณ์ที่มีจำหน่ายทั่วไปและซอฟต์แวร์โอเพนซอร์ส การประกอบลงในตู้กระดาษลัง DIY ช่วยให้ต้นทุนต่ำและนำเสนอแนวคิดได้ดี อย่างไรก็ตาม หากจะนำไปต่อยอดเป็นระบบลงคะแนนจริงที่ใช้ในวงกว้าง อาจต้องพัฒนาเพิ่มเติมในหลายด้าน เช่น การเข้ารหัสข้อมูลเมื่อส่งผ่านเครือข่าย, การป้องกันการฉ้อโกงเครื่อง (tamper-proof), การสำรองข้อมูลคะแนนลงหน่วยความจำถาวร, หรือการรองรับจำนวนผู้มีสิทธิ์และผู้สมัครที่มากขึ้น เป็นต้น

โดยสรุป เครื่องลงคะแนนเสียงอัจฉริยะนี้แสดงถึงการผสมผสานฮาร์ดแวร์และซอฟต์แวร์เพื่อแก้ปัญหาของระบบลงคะแนนแบบเดิม ไม่ว่าจะเป็นการป้องกันการโกงคะแนนหรือการนับคะแนนที่ล่าช้า ระบบนี้เน้นที่ความถูกต้อง แม่นยำ และความโปร่งใส ซึ่งเป็นหัวใจสำคัญของการเลือกตั้งในทุกๆ ระดับ ¹⁰ แม้จะเป็นต้นแบบสำหรับโครงงานนักศึกษา แต่แนวคิดและเทคโนโลยีที่ใช้ก็สามารถต่อยอดไปสู่ระบบ E-Voting ที่ใช้งานจริงได้ในอนาคต หากได้รับการพัฒนาเพิ่มเติมอย่างเหมาะสม

แหล่งอ้างอิง:

1. Arduino Based Authenticated Voting Machine (AVM) using RFID and Fingerprint for the Student Elections ^{1 9}
2. Fingerprint Based Biometric Voting Machine using Arduino – CircuitDigest ^{13 6}
3. Fingerprint & RFID Based Voting Machine – IEEE Maker Project ⁷
4. A Proposed Framework for Fingerprint-based Voting System – ResearchGate ²
5. ESP32 with MFRC522 RFID Reader – Random Nerd Tutorials ⁴
6. RFID & Fingerprint Advanced Security System – electronicclinic.com ^{5 14}
7. Arduino Code for Thermal Printer – Adafruit ^{11 12}
8. Other references and resources on electronic voting machine design and IoT connectivity ^{10 15}

^{1 2 3 8 9 10 15} (PDF) Arduino Based Authenticated Voting Machine (AVM) using RFID and Fingerprint for the Student Elections
https://www.researchgate.net/publication/347968409_Arduino_Based_Authenticated_Voting_Machine_AVM_using_RFID_and_Fingerprint_for_the_Student_Elections

⁴ ESP32 with MFRC522 RFID Reader/Writer (Arduino IDE) | Random Nerd Tutorials
<https://randomnerdtutorials.com/esp32-mfrc522-rfid-reader-arduino/>

^{5 14} RFID & Fingerprint based Advanced Security System using Arduino
<https://www.electronicclinic.com/rfid-fingerprint-based-advanced-security-system-using-arduino/>

^{6 13} Fingerprint Based Biometric Voting Machine Project using Arduino
<https://circuitdigest.com/microcontroller-projects/fingerprint-based-biometric-voting-machine-arduino>

⁷ Fingerprint & RFID Based Voting Machine - The IEEE Maker Project
<https://transmitter.ieee.org/makerproject/view/01ccd>

^{11 12} Arduino Code | Mini Thermal Receipt Printers | Adafruit Learning System
<https://learn.adafruit.com/mini-thermal-receipt-printer/microcontroller>