คู่มือและคำอธิบายโปรแกรม

โดย ชื่อ **นายศศิศ ศรีรัตน์** รหัสนักศึกษา **6413112** คณะและภาควิชา **EGCO**

ไฟล์ที่แนบมา

1. main.c -> เป็นไฟล์หลัก
2. IncFile1.h -> ไฟล์ที่ใช้ตั้งค่า PIN/PORT รวมถึงการ Include header ต่างๆ
3. Tester -> สำหรับโปรแกรม PICSimLab

หมายเหตุ : Keypad 4x4 บางครั้งก็กดยากและกดไม่ติด วิธีแก้คือกดแช่นานๆ

การตั้งค่า layout ของ keypad 4x4

แบบปกติจะเป็นดังนี้ แบบที่ใช้ในการทำงานบน PICSimLab

อธิบายในส่วนของโปรแกรม

โดยหลัก ๆ แล้ว ในส่วนของคำสั่งของการตั้งค่า Keypad และในส่วนของการแสดงผลออกมาทาง LCD ส่วนใหญ่ได้อ้างอิงและใช้โค้ดคำสั่งจากตัวอย่าง Lab ที่ 9 เป็นหลัก โดยโค้ดที่เหลือจะอธิบายโค้ดโดยใช้ฟังก์ชันในการอธิบาย ดังนี้

ฟังก์ชัน lcd_printNumber() และ **ฟังก์ชัน lcd_printError()** เป็นฟังก์ชันที่ใช้สำหรับการแสดงข้อมูลตัวเลขและข้อผิดพลาดบนหน้าจอ LCD โดยทำงานดังนี้

lcd_printNumber(float num):

1. **รับอินพุต**
    - float num: รับค่าที่ต้องการแสดงบนหน้าจอ LCD
2. **แปลงเลขเป็นข้อความ:**
    - dtostrf(num, 5, 2, buffer): ใช้ฟังก์ชัน dtostrf ใน stdlib.h เพื่อแปลงค่าที่ได้จาก num ให้เป็นสตริง และเก็บไว้ใน buffer โดยกำหนดให้มีทศนิยม 2 ตำแหน่ง
3. **แสดงข้อความบนหน้าจอ LCD:**
    - ใช้ lcd_print() เพื่อแสดงข้อความที่เก็บใน buffer ลงบนหน้าจอ LCD

lcd_printError(const char* errorMessage):

1. **รับอินพุต:**
    - const char* errorMessage: รับข้อความที่ต้องการแสดงเป็นข้อผิดพลาด
2. **ลบหน้าจอ LCD และแสดงข้อความผิดพลาด:**
    - lcd_gotoxy(1, 4): ใช้ lcd_gotoxy() เพื่อเลื่อนตำแหน่งเริ่มแสดงข้อความ
    - lcd_print(errorMessage): ใช้ lcd_print() เพื่อแสดงข้อความผิดพลาดที่ได้รับ
3. **ล้างหน้าจอ LCD และเลื่อนกลับไปที่ตำแหน่งเริ่มต้น:**
    - lcd_clear(): ใช้ lcd_clear() เพื่อล้างหน้าจอ LCD อีกรอบ
    - lcd_gotoxy(1, 1): ใช้ lcd_gotoxy() เพื่อเลื่อนตำแหน่งเริ่มแสดงข้อความในแนวแกน x ไปที่บรรทัดที่ 1

ฟังก์ชัน getPrecedence() เป็นฟังก์ชันที่ใช้เพื่อระบุลำดับความสำคัญของตัวดำเนินการทางคณิตศาสตร์ (operators) ซึ่งมีไว้ในการช่วยในการจัดการการดำเนินการในสมการทางคณิตศาสตร์ โดยจะทำงาน ดังนี้

1. **รับอินพุต:**
    - char op: รับตัวดำเนินการทางคณิตศาสตร์ที่ต้องการระบุลำดับความสำคัญเข้ามา
2. **คืนค่าลำดับความสำคัญ:**
    - โดยใช้ switch เพื่อตรวจสอบ op และคืนค่าลำดับความสำคัญตามที่กำหนดดังนี้ คือ

- ถ้าเป็น ^ ให้คืนค่า 3
- ถ้าเป็น * หรือ / ให้คืนค่า 2
- ถ้าเป็น + หรือ - ให้คืนค่า 1
- สำหรับตัวดำเนินการที่ไม่รู้จักให้คืนค่า 0 (Default สำหรับตัวเลข)

ฟังก์ชัน performOperation() เป็นฟังก์ชันที่ใช้สำหรับทำการคำนวณผลลัพธ์ของการดำเนินการทางคณิตศาสตร์ระหว่าง operand สองตัว โดยดูตามตัวดำเนินการที่ระบุ (operator) ซึ่งจะเป็นอาร์กิวเมนต์ของฟังก์ชันนี้ โดยการทำงานของ performOperation() อธิบายได้ดังนี้ คือ

1. **รับอินพุต:**
    - float operand1: ตัวเลขที่เป็นตัวที่ 1
    - float operand2: ตัวเลขที่เป็นตัวที่ 2
    - char operator: ตัวดำเนินการที่ต้องการทำการคำนวณ
2. **ทำการคำนวณ:**
    - ใช้ switch เพื่อตรวจสอบตัวดำเนินการและทำการคำนวณตามแต่ละกรณี
        - ถ้าเป็น + ให้คืนผลลัพธ์จากการบวก operand1 และ operand2
        - ถ้าเป็น - ให้คืนผลลัพธ์จากการลบ operand2 จาก operand1
        - ถ้าเป็น * ให้คืนผลลัพธ์จากการคูณ operand1 และ operand2
        - ถ้าเป็น / ให้คืนผลลัพธ์จากการหาร operand1 ด้วย operand2 (หาก operand2 ไม่เท่ากับ 0)
        - ถ้าเป็น ^ ให้คืนผลลัพธ์จากการยกกำลัง operand1 กับ operand2
3. **การจัดการข้อผิดพลาด:**
    - ในกรณีที่ operator เป็น / และ operand2 เท่ากับ 0
        - จะแสดงข้อความ "Division by 0!" บนหน้าจอ LCD โดยใช้ฟังก์ชัน lcd_printError()
        - คืนค่า 0 เป็นผลลัพธ์
4. **แสดงผลลัพธ์:**
    - แสดงผลลัพธ์ของการคำนวณทางคณิตศาสตร์

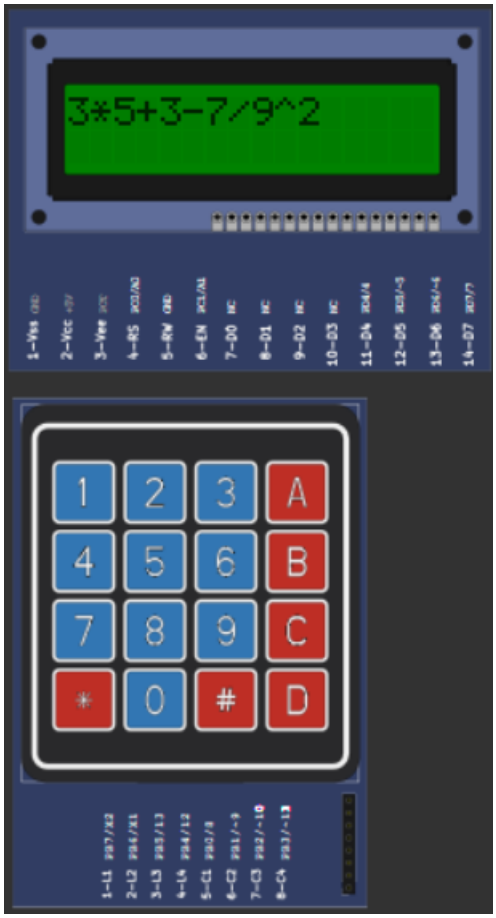
ฟังก์ชัน evaluateExpression() เป็นฟังก์ชันที่ใช้สำหรับคำนวณค่าของนิพจน์ทางคณิตศาสตร์ที่รับเข้ามาในรูปของสตริง (string) แล้วนำแสดงผลลัพธ์ออกทาง LCD โดยในโปรแกรมนี้ การทำงานของ evaluateExpression() สามารถแบ่งเป็นขั้นตอนต่อได้ดังไปนี้

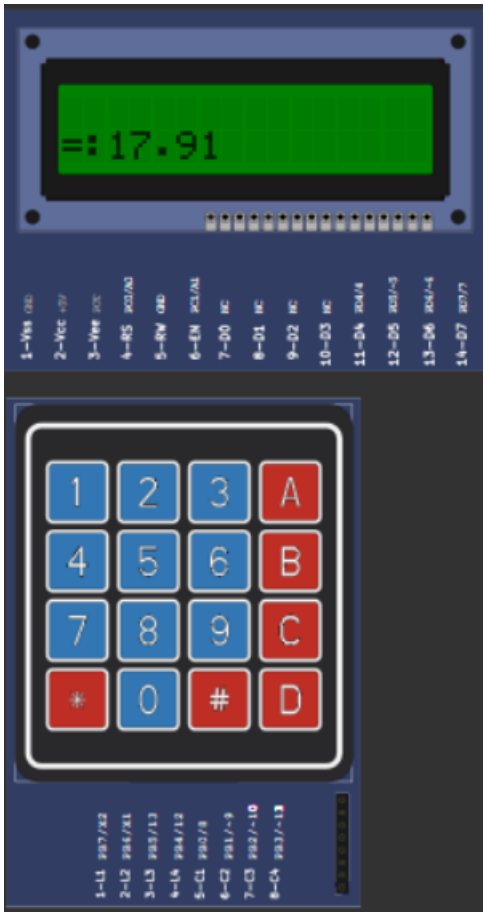
1. **ตรวจสอบตัวเลขและตัวดำเนินการ:**
    - ใน loop ของ evaluateExpression มีการตรวจสอบตัวเลขและตัวดำเนินการที่ตำแหน่งปัจจุบันในสตริง expression ถ้าเป็นตัวเลข โปรแกรมจะดึงค่าตัวเลขนั้นออกมาแล้วเก็บใน operands
    - ถ้าเป็นตัวดำเนินการ โปรแกรมจะตรวจสอบเงื่อนไขพิเศษเพื่อป้องกันกรณีไม่ถูกต้อง เช่น ตัวดำเนินการที่มาติดกัน หรือตัวดำเนินการที่ตำแหน่งแรกของสมการ เช่น */4, *4+3, /4+3 หรือ +/4+3
2. **คำนวณด้วยฟังก์ชัน performOperation:**
    - เมื่อมีตัวเลขและตัวดำเนินการพร้อมที่จะคำนวณ โปรแกรมจะเรียกใช้ฟังก์ชัน performOperation และส่งตัวเลข 2 ตัวและตัวดำเนินการไป
    - ใน performOperation มีการตรวจสอบว่าตัวดำเนินการเป็นอะไร และทำการคำนวณตามนั้น ถ้าเป็นการหาร จะตรวจสอบว่าตัวหารไม่เป็น 0 และจะแสดงข้อความผิดพลาดถ้าหารด้วยศูนย์ (แต่บางทีก็จะไม่ขึ้นอะไร ซึ่งหมายความว่า Error ได้เหมือนกัน)
    - ผลลัพธ์ที่ได้จะถูกส่งกลับไปให้ evaluateExpression เพื่อเก็บไว้ใน operands
3. **การแสดงผลลัพธ์:**
    - เมื่อคำนวณเสร็จสิ้นทั้งหมด โปรแกรมจะแสดงผลลัพธ์บนหน้าจอ LCD โดยใช้ lcd_print และ lcd_printNumber
4. **การรับปุ่มและอัปเดตสตริง expression:**
    - ใน loop หลักของ main โปรแกรมจะรอรับปุ่มจากแป้นพิมพ์
    - ถ้าปุ่มที่ถูกกดเป็นตัวเลข จุดทศนิยม หรือตัวดำเนินการ โปรแกรมจะทำการแสดงปุ่มนั้นบนหน้าจอ LCD และเก็บลงใน expression
    - ถ้าปุ่มที่ถูกกดเป็น "=" โปรแกรมจะทำการส่ง expression ที่ได้ไปให้ evaluateExpression เพื่อทำการคำนวณ

ฟังก์ชันหลักหรือฟังก์ชัน main โดยทำงานดังต่อไปนี้

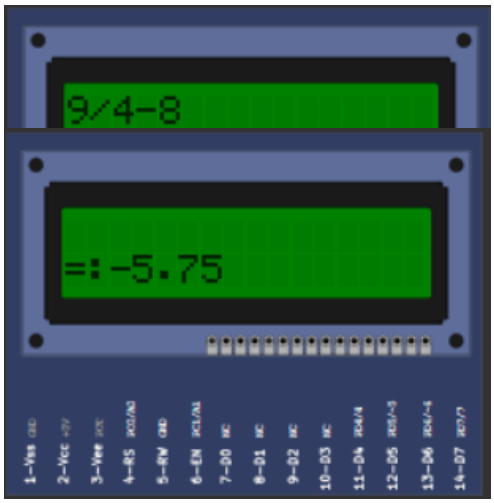
1. **เรียกใช้ฟังก์ชัน โดย:**
    - lcd_init(): เป็นการเริ่มต้น LCD display
    - lcd_hideCursor(): ซ่อนเคอร์เซอร์บน LCD
    - lcd_gotoxy(1, 1): กำหนดตำแหน่งเริ่มต้นของเคอร์เซอร์ LCD ที่มุมบนซ้าย
    - keyboard_init(): เริ่มต้น keypad
2. **ลูปหลัก (while (1)):**
    - keyboard(): อ่านข้อมูลจากแป้นพิมพ์และกำหนดค่า keyvalue
    - หากปุ่มที่ถูกกดเป็นตัวเลข หรือตัวดำเนินการ (+, -, *, /, ^) จะถูกเพิ่มไปยังอาร์เรย์ expression และตัวอักษรที่เข้ากันจะแสดงบน LCD
    - หากปุ่มที่ถูกกดคือ '=' (ก็คือปุ่ม *) การทำงานจะถูกสิ้นสุด และฟังก์ชัน evaluateExpression จะถูกเรียก
3. **โปรแกรมจะดำเนินการแบบนี้ไปเรื่อย ๆ จนกว่าผู้ใช้งานจะออกจากโปรแกรมเอง**

ตัวอย่างผลการทำงาน

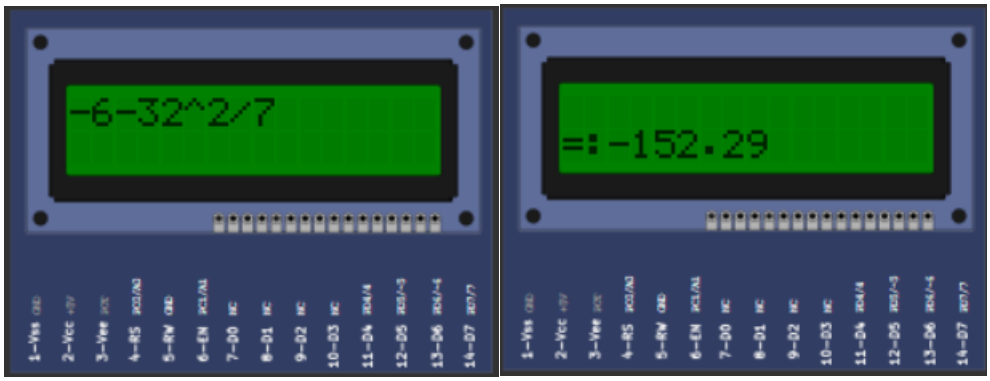




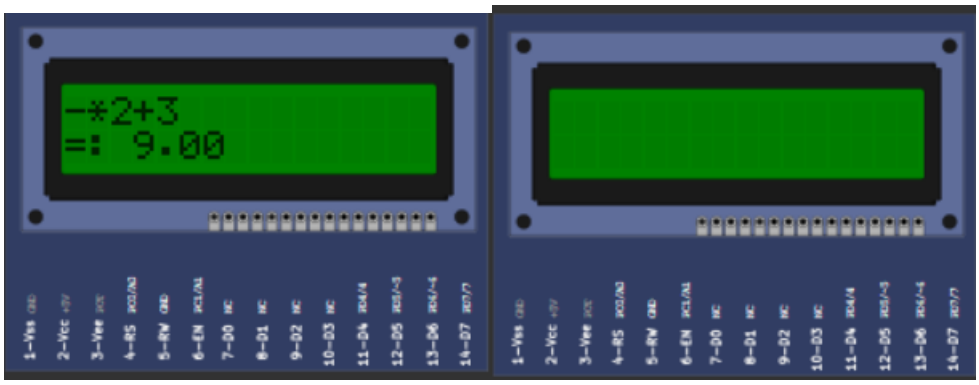
เมื่อลองคำนวณเลข 9^10 ผลลัพธ์ที่ได้จะแสดงตัวเลข 10 หลักพร้อมทศนิยม 2 ตำแหน่ง



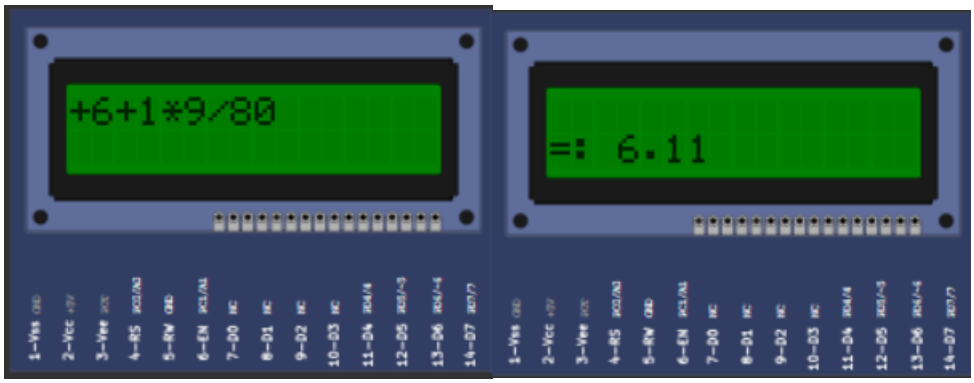
สามารถแสดงผลที่ติดลบออกมาได้



กรณีเมื่อมีเครื่องหมาย – ด้านหน้า



กรณีเมื่อมีเครื่องหมาย * (คูณ) หรือ /(หาร) ด้านหน้า โปรแกรมจะ Error



ถ้าเกิดเครื่องหมาย ด้านหน้าเป็น + หรือ - เฉยๆ โปรแกรมสามารถทำงานได้ตามปกติ