

หน่วยที่ 6 การแสดงผลด้วยจอ LCD ของ ARDUINO

สาระสำคัญ

คำว่า LCD ย่อมาจากคำว่า Liquid Crystal Display ซึ่งเป็นจอที่ทำมาจากผลึกคริสตอลเหลว หลักการคือ ด้านหลังจอจะมีไฟส่องสว่าง หรือที่เรียกว่า Backlight อยู่ เมื่อมีการปล่อยกระแสไฟฟ้าเข้าไปกระตุ้นที่ผลึก ก็จะทำให้ผลึกโปร่งแสง ทำให้แสงที่มาจากไฟ Backlight แสดงขึ้นมาบนหน้าจอ ส่วนอื่นที่โดนผลึกปิดกั้นไว้จะไม่สว่าง ผลึกมีสีที่แตกต่างกันตามสีของผลึกคริสตอล เช่นสีเขียว หรือ สีฟ้า ฯลฯ ทำให้เมื่อมองไปที่จอก็จะพบกับตัวหนังสือแล้วพบกับพื้นหลังสีต่างๆกัน

เนื้อหาสาระการเรียนรู้

- 6.1 ความรู้ทั่วไปเกี่ยวกับ LCD Display
- 6.2 โครงสร้างโดยทั่วไปของ LCD
- 6.3 การเชื่อมต่อสัญญาณขาข้อมูลระหว่าง Arduino กับ LCD Controller
- 6.4 รายละเอียดคำสั่งในการสั่งงานระหว่าง Arduino กับจอ LCD
- 6.5 ขั้นตอนการติดตั้งไลบรารีจอแสดงผล LCD
- 6.6 ฟังก์ชันสั่งงานจอ LCD
- 6.7 การสร้างตัวอักษรหรือใส่รูปภาพลงจอ LCD

จุดประสงค์การเรียนรู้

จุดประสงค์ทั่วไป

1. เพื่อให้มีความรู้ความเข้าใจเกี่ยวกับลักษณะการแสดงผลด้วยจอ LCD ของ Arduino
2. เพื่อให้สามารถนำความรู้ไปประยุกต์ใช้ในการเขียนโปรแกรมกำหนดการทำงานของ Arduino
3. เพื่อให้ตระหนักถึงความสำคัญของลักษณะการแสดงผลด้วยจอ LCD ของ Arduino

จุดประสงค์เชิงพฤติกรรม

1. อธิบายความรู้ทั่วไปเกี่ยวกับ LCD Display ได้
2. อธิบายโครงสร้างโดยทั่วไปของ LCD ได้
3. เชื่อมต่อจอ Character LCD ได้
4. เชื่อมต่อสัญญาณขาข้อมูลระหว่างArduinoกับ LCD (I2C) ได้
5. บอกรายละเอียดคำสั่งในการสั่งงานระหว่าง Arduino กับจอ LCD ได้

แบบทดสอบก่อนเรียน หน่วยที่ 6

เรื่อง การแสดงผลด้วยจอ LCD ของ Arduino

เรื่อง การแสดงผลด้วยจอ LCD ของ Arduino

ใช้เวลา 20 นาที

วิชา ไมโครคอนโทรลเลอร์เบื้องต้น

รหัสวิชา (2127-2107)

ระดับชั้น ประกาศนียบัตรวิชาชีพ (ปวช.)

สาขาวิชา เมคคาทรอนิกส์

คำชี้แจง 1. แบบทดสอบมีทั้งหมด 10 ข้อ (10 คะแนน)

2. ให้ผู้เรียนเลือกคำตอบที่ถูกต้องที่สุดแล้วกาเครื่องหมายกากบาท (X) ลงในกระดาษคำตอบ

1. คำว่า LCD ย่อมาจากคำว่า

- ก. Liquid Computer Display
- ข. Liquid Center Display
- ค. Liquid Communication Display
- ง. Liquid Crystal Display

2. จอ LCD จะแบ่งเป็น 2 แบบใหญ่ๆตามลักษณะการแสดงผลดังนี้

- ก. Character LCD/ Graphic LCD
- ข. Center LCD/ Graphic LCD
- ค. Computer LCD/ Graphic LCD
- ง. Crystal LCD/ Graphic LCD

3. โครงสร้างของ LCD ทั่วไปจะประกอบขึ้นด้วยแผ่นแก้ว 2 แผ่นประกบกันอยู่ โดยเว้นช่องว่างตรงกลางไว้

- ก. 20-26 ไมโครเมตร
- ข. 16-20 ไมโครเมตร
- ค. 10-16 ไมโครเมตร
- ง. 6-10 ไมโครเมตร

4. จอ LCD สามารถแสดงผลให้เรามองเห็นได้ทั้งหมด 3 แบบด้วยกันคือ

- ก. สะท้อนแสง/ส่งเสริมแสง/สะท้อนและส่งเสริมแสง
- ข. สะสมแสง/ส่งผ่านแสง/สะสมและส่งผ่านแสง
- ค. สะสมแสง/สะท้อนแสง/สะสมและสะท้อนแสง
- ง. สะท้อนแสง/ส่งผ่านแสง/สะท้อนแสงและส่งผ่านแสง

5. ในการเชื่อมต่อแบบขนาน Pin 7 – 14 จอ Character LCD ใช้สำหรับ
- ก. RW
 - ข. RS
 - ค. Data
 - ง. E/EN
6. ขา E เป็นสัญญาณ Enable เมื่อมีค่าเป็น “1” หมายความว่า
- ก. ไม่สนใจสัญญาณ RS R/W และ DB7-DB0
 - ข. digitalWrite
 - ค. pinMode
 - ง. อุปกรณ์ภายนอกต้องการติดต่ออ่านหรือเขียนข้อมูล
7. R/W เป็นสัญญาณบอกให้ LCD ทราบว่าอุปกรณ์ภายนอกต้องการอ่านหรือเขียนกับ LCD โดย
- ก. ใช้แจ้งว่าได้รับข้อมูลตัวอักษร (characters) แล้ว และพร้อมสำหรับการอ่านไปใช้งาน
 - ข. R/W = “1” หมายถึง เขียน และ R/W = “0” หมายถึงอ่าน
 - ค. ส่งค่ากลับจากฟังก์ชัน
 - ง. R/W = “0” หมายถึง เขียน และ R/W = “1” หมายถึง อ่าน
8. ขาสัญญาณข้อมูล DB0-DB7 เป็นสัญญาณแบบ 2 ทิศทาง โดยจะสัมพันธ์กับสัญญาณ
- ก. E
 - ข. RS
 - ค. VDD
 - ง. R/W
9. lcd.clear() คือฟังก์ชันอะไร
- ก. เมื่อมีตัวอักษรใดๆอยู่บนหน้าจอ จะถูกล้างออกทั้งหมด
 - ข. ใช้สำหรับเขียนข้อมูลออกไปที่ละตัวอักษรใช้ล้างหน้าจอ
 - ค. ใช้สั่งให้แสดงเคอเซอร์บนหน้าจอ
 - ง. ใช้เขียนข้อมูลออกไปทั้งข้อความ
10. RS ใช้บอกให้ LCD Controller ทราบว่า
- ก. Code ที่ส่งมาทางขา Data เป็นคำสั่งหรือข้อมูล
 - ข. Code ที่ส่งมาทางขา Data เป็นคำสั่ง
 - ค. Code ที่ส่งมามีค่าเท่ากับ 11
 - ง. Code ที่ส่งมาทางขา Data เป็นข้อมูล

หน่วยที่ 6

การแสดงผลด้วยจอ LCD ของ Arduino

6.1 ความรู้ทั่วไปเกี่ยวกับ LCD Display

คำว่า LCD ย่อมาจากคำว่า Liquid Crystal Display ซึ่งเป็นจอที่ทำมาจากผลึกคริสตอลเหลว หลักการคือด้านหลังจอจะมีไฟส่องสว่าง หรือที่เรียกว่า Backlight อยู่ เมื่อมีการปล่อยกระแสไฟฟ้าเข้าไปกระตุ้นที่ผลึก ก็จะทำให้ผลึกโปร่งแสง ทำให้แสงที่มาจากไฟ Backlight แสดงขึ้นมาบนหน้าจอ ส่วนอื่นที่โดนผลึกปิดกั้นไว้จะไม่สว่าง ผลึกมีสีที่แตกต่างกันตามสีของผลึกคริสตอล เช่น สีเขียว หรือ สีฟ้า ฯลฯ ทำให้เมื่อมองไปที่จอ ก็จะพบกับตัวหนังสือ แล้วพบกับพื้นหลังสีต่างๆกัน

จอ Liquid Crystal Display (LCD) เป็นจอแสดงผลรูปแบบหนึ่งที่ยิมนำมาใช้งานกันกับระบบสมองกลฝังตัวอย่างแพร่หลาย จอ LCD มีทั้งแบบแสดงผลเป็นตัวอักษรเรียกว่า Character LCD ซึ่งมีการกำหนดตัวอักษรหรืออักขระที่สามารถแสดงผลไว้ได้อยู่แล้ว และแบบที่สามารถแสดงผลเป็นรูปภาพหรือสัญลักษณ์ได้ตามความต้องการของผู้ใช้งานเรียกว่า Graphic LCD นอกจากนี้บางชนิดเป็นจอที่มีการผลิตขึ้นมาใช้เฉพาะงาน ทำให้มีรูปแบบและรูปร่างเฉพาะเจาะจงในการแสดงผล เช่น นาฬิกาดิจิตอล เครื่องคิดเลข หรือ หน้าปัดวิทยุ เป็นต้น

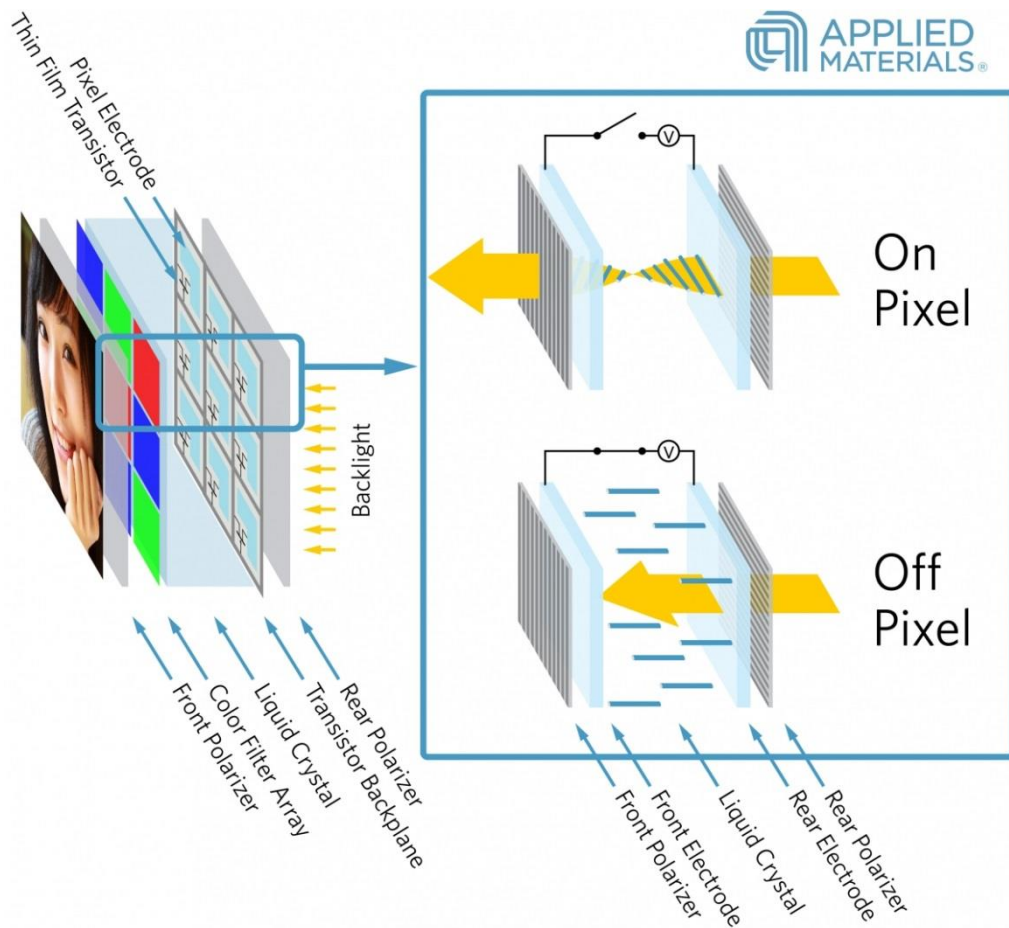
จอ LCD แบ่งเป็น 2 แบบใหญ่ๆตามลักษณะการแสดงผลดังนี้

1. Character LCD เป็นจอที่แสดงผลเป็นตัวอักษรตามช่องแบบตายตัว เช่น จอ LCD ขนาด 16x2 หมายถึงใน 1 แถวมีตัวอักษรใส่ได้ 16 ตัว และมีทั้งหมด 2 บรรทัดให้ใช้งาน ส่วน 20x4 จะหมายถึงใน 1 แถวมีตัวอักษรใส่ได้ 20 ตัว และมีทั้งหมด 2 บรรทัด

2. Graphic LCD เป็นจอที่สามารถกำหนดได้ว่าจะให้แต่ละจุดบนหน้าจอขึ้นแสง หรือปล่อยแสงออกไป ทำให้อาจสามารถสร้างรูปขึ้นมาบนหน้าจอได้ การระบุขนาดจะระบุในลักษณะของจำนวนจุด (Pixels) ในแต่ละแนว เช่น 128x64 หมายถึงจอที่มีจำนวนจุดตามแนวนอน 128 จุด และมีจุดตามแนวตั้ง 64 จุด

หลักการทำงานอาศัย-ของเหลวพิเศษที่มีคุณสมบัติการบิดแกนโพลาไรซ์ของแสง ถ้าจ่ายแรงดันไฟฟ้าเข้าไประหว่างสารเหล่านี้ โมเลกุลจะบิดตัวและทำให้แสงไม่สามารถผ่านกระจกออกมาได้ ถ้าไม่มีการจ่ายแรงดันไฟฟ้า แสงจะทะลุผ่านออกมาได้

รูปที่ 6.1 อธิบายการทำงานเกิดจากกระจกโพลาไรซ์ 2 แผ่น ที่มีแกนตั้งฉากกัน แสงจะไม่สามารถลอดผ่านออกมาได้ เปรียบเหมือนเอาแว่นตาโพลาไรซ์สองอันมาบิดทำมุมตั้งฉากกัน แสงจะไม่ลอดผ่าน และมีการค้นพบว่า มีของเหลวชนิดหนึ่งที่ปกติแล้วจะสามารถ "บิดแกนโพลาไรซ์ของแสง" ได้ จึงเกิดความคิดเอามาทำจอ LCD



รูปที่ 6.1 โครงสร้างภายในจอ LCD
(ที่มา www.Applied Materials.com)



รูปที่ 6.2 นาฬิกาจอ LCD
(ที่มา www.Thaieasyelec.com)



รูปที่ 6.3 จอ LCD
(ที่มาจาก www.Thaieasyelec.com)



รูปที่ 6.4 จอคอมพิวเตอร์ LCD
(ที่มาจาก www.Thaieasyelec.com)

6.2 โครงสร้างโดยทั่วไปของ LCD

โครงสร้างของ LCD ทั่วไปจะประกอบขึ้นด้วยแผ่นแก้ว 2 แผ่นประกบกันอยู่ โดยเว้นช่องว่างตรงกลางไว้ 6-10 ไมโครเมตร ผิวด้านในของแผ่นแก้วจะเคลือบด้วยตัวนำไฟฟ้าแบบใสเพื่อใช้แสดงตัวอักษร ตรงกลางระหว่างตัวนำไฟฟ้าแบบใสกับผลึกเหลว มีชั้นของสารที่ทำให้โมเลกุลของผลึกรวมตัวกันในทิศทางที่แสงส่องมากระทบ เรียกว่า Alignment Layer และผลึกเหลวที่ใช้โดยทั่วไปจะเป็นแบบ Magnetic โดย LCD สามารถแสดงผลให้เรา มองเห็นได้ทั้งหมด 3 แบบด้วยกันคือ

6.2.1 แบบใช้การสะท้อนแสง (Reflective Mode) LCD

แบบนี้ใช้สารประเภทโลหะเคลือบอยู่ที่แผ่นหลังของ LCD ซึ่ง LCD ประเภทนี้เหมาะกับการนำมาใช้งานในที่ที่มีแสงสว่างเพียงพอ

6.2.2 แบบใช้การส่งผ่าน (Transitive Mode) LCD

แบบนี้วางหลอดไฟไว้ด้านหลังจอ เพื่อให้การอ่านค่าที่แสดงผลทำได้ชัดเจน

6.2.3 แบบส่งผ่านและสะท้อน (Transflective Mode) LCD

แบบนี้เป็นการนำเอาข้อดีของจอแสดงผล LCD ทั้ง 2 แบบมารวมกัน

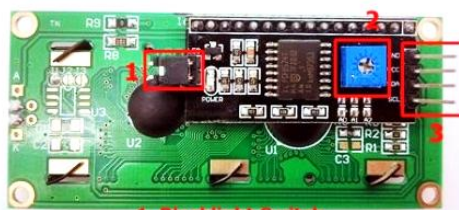
ในบทนี้จะกล่าวถึงจอ LCD ที่แสดงผลเป็นอักขระหรือตัวอักษร ตามท้องตลาดทั่วไปจะมีหลายแบบด้วยกัน มีทั้ง 16 ตัวอักษร 20 ตัวอักษรหรือมากกว่า และจำนวนบรรทัดจะมีตั้งแต่ 1 บรรทัด 2 บรรทัด 4 บรรทัดหรือมากกว่า ตามแต่ความต้องการและลักษณะของงานที่ใช้ หรืออาจจะมีแบบสั่งทำเฉพาะงานก็เป็นได้ ในบทนี้จะยกตัวอย่างจอ LCD ขนาด 16x2 Character หรือที่นิยมเรียกกันว่าจอ LCD 16 ตัวอักษร 2 บรรทัด สามารถหาซื้อได้ง่ายและมีราคาไม่สูง เหมาะสมกับการใช้งานแสดงผลไม่มากในหน้าจอเดียว

จอ LCD 16x2 Character ที่นิยมวางจำหน่ายจะมีอยู่ 2 แบบด้วยกันคือ LCD แบบปกติที่เชื่อมต่อแบบขนาน (Parallel) และ LCD แบบที่เชื่อมต่ออนุกรม (Serial) แบบ I2C โดยทั้ง 2 แบบตัวจอมีลักษณะเดียวกัน เพียงแต่แบบ I2C จะมีบอร์ดเสริมทำให้สื่อสารแบบ I2C ได้เชื่อมต่อได้สะดวกขึ้น



รูปที่ 6.5 จอ LCD 16x2 Character (Parallel)

(ที่มาจาก www.Thaieasyelec.com)



1. Blacklight Switch
2. Contrast Adjustment
3. I2C pin interface

รูปที่ 6.6 จอ LCD 16x2 Character (I2C)

(ที่มาจาก www.Thaieasyelec.com)

6.3 การเชื่อมต่อสัญญาณขาข้อมูลระหว่าง Arduino กับ LCD Controller

ในการควบคุมหรือสั่งงาน โดยทั่วไปจอ LCD จะมีส่วนควบคุม (Controller) อยู่ในตัวแล้ว ผู้ใช้สามารถส่งรหัสคำสั่งสำหรับควบคุมการทำงานของจอ LCD (I2C) เช่นเดียวกันกับจอ LCD แบบธรรมดา พุดต่างๆ คือรหัสคำสั่งที่ใช้ในการควบคุมนั้นเหมือนกัน แต่ต่างกันตรงที่รูปแบบในการรับส่งข้อมูล ในบทความนี้เราจะมาพูดถึงจอ LCD 16x2 ที่มีการส่งข้อมูลรูปแบบ I2C ที่ใช้ขาเพียง 4 ขาที่ใช้ในการเชื่อมต่อเท่านั้น

ในการควบคุมหรือสั่งงานจอ LCD Controller ของจอนี้เป็น Hitachi เบอร์ HD44780 และขาในการเชื่อมต่อระหว่าง LCD กับ Arduino มีดังนี้

1. GND เป็นกราวด์ใช้ต่อระหว่าง Ground ของ Arduino กับ LCD
2. VCC เป็นไฟเลี้ยงวงจรที่ป้อนให้กับ LCD ขนาด +5VDC
3. VO ใช้ปรับความสว่างของหน้าจอ LCD
4. RS ใช้บอกให้ LCD Controller ทราบว่า Code ที่ส่งมาทางขา Data เป็นคำสั่งหรือข้อมูล
5. R/W ใช้กำหนดว่าจะอ่านหรือเขียนข้อมูลกับ LCD Controller
6. E เป็นขา Enable หรือ Chips Select เพื่อกำหนดการทำงานให้กับ LCD Controller
7. DB0-DB7 เป็นขาสัญญาณ Data ใช้สำหรับเขียนหรืออ่านข้อมูล/คำสั่ง กับ LCD Controller

วิธีการสั่งงานจะแตกต่างกันไปโดย LCD Controller สามารถรับรหัสคำสั่งจาก Arduino ได้จากสัญญาณ RS R/W และ DB0-DB7 ในขณะที่สัญญาณ E มีค่า Logic เป็น "1" ซึ่งสัญญาณเหล่านี้จะใช้ร่วมกันเพื่อกำหนดเป็นรหัสคำสั่งสำหรับสั่งงาน LCD โดยหน้าที่ของแต่ละสัญญาณสรุปได้ดังนี้

1. E เป็นสัญญาณ Enable เมื่อมีค่าเป็น

"1" เป็นการบอกให้ LCD ทราบว่าอุปกรณ์ภายนอกต้องการติดต่ออ่านหรือเขียนข้อมูล

"0" ให้ LCD ไม่สนใจสัญญาณ RS R/W และ DB7-DB0

2. RS เป็นสัญญาณสำหรับกำหนดให้ LCD ทราบว่าอุปกรณ์ภายนอกต้องการติดต่อกับ LCD ในขณะนั้น เป็นรหัสคำสั่งหรือข้อมูล โดย

3. R/W เป็นสัญญาณสำหรับบอกให้ LCD ทราบว่าอุปกรณ์ภายนอกต้องการอ่านหรือเขียนกับ LCD โดย

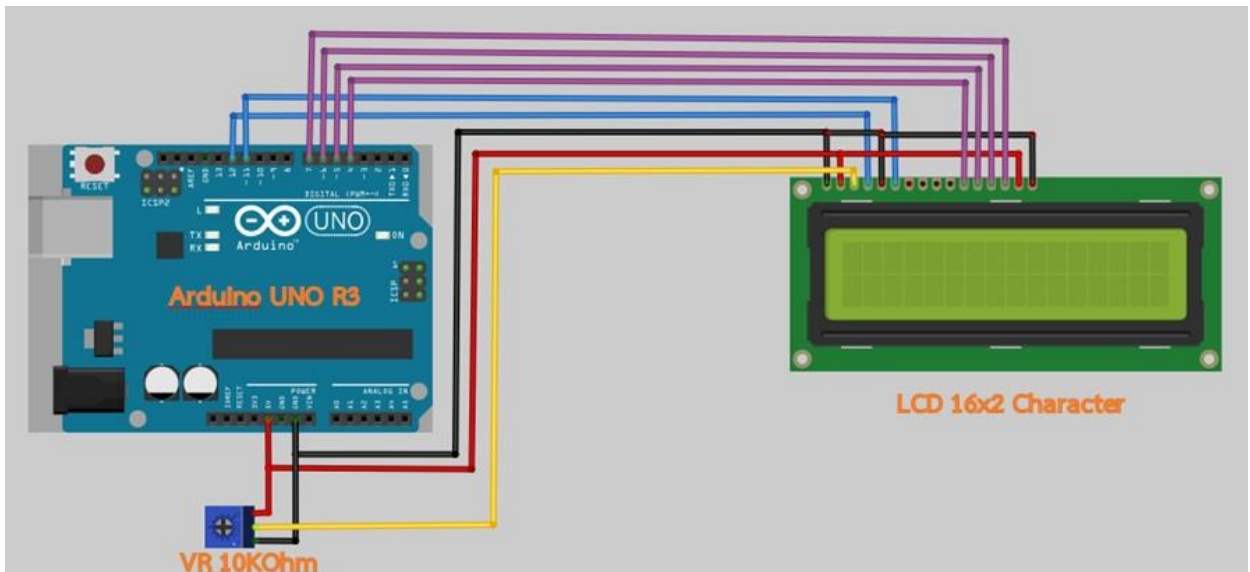
R/W = "0" หมายถึง เขียน

R/W = "1" หมายถึง อ่าน

4. DB0-DB7 เป็นสัญญาณแบบ 2 ทิศทาง โดยจะสัมพันธ์กับสัญญาณ R/W ใช้สำหรับรับคำสั่งและข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอก โดยถ้า R/W = "0" สัญญาณ DB7-DB0 จะส่งจากอุปกรณ์ภายนอกมาที่ LCD แต่ถ้า R/W = "1" สัญญาณ DB7-DB0 จะส่งจาก LCD ไปยังอุปกรณ์ภายนอก

การเชื่อมต่อสัญญาณขาข้อมูลระหว่าง Arduino กับ LCD Controller สามารถทำได้ 2 ลักษณะ คือ การเชื่อมต่อแบบ 8 บิต (DB0-DB7) และการเชื่อมต่อแบบ 4 บิต (DB4-DB7) ทั้งสองแบบแตกต่างกันเพียงจำนวนขาที่ใช้คือ 8 หรือ 4 ขา และยังสามารถทำงานได้เหมือนกัน อย่างที่แน่นอนในการส่งข้อมูลแบบ 4 ขา ผลย่อมทำได้ช้ากว่า 8 ขา แต่ไม่ได้ช้ามากจนสังเกตได้ด้วยสายตา

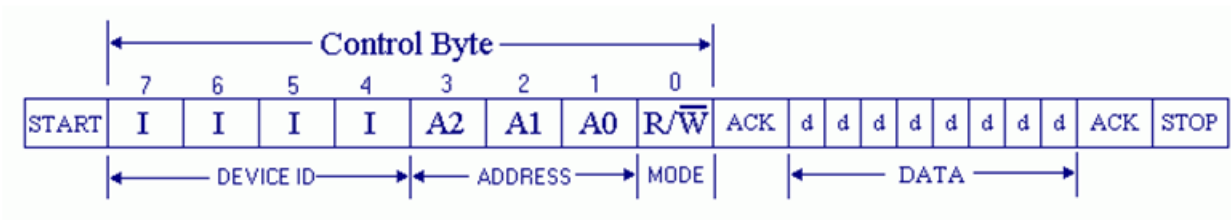
ในการต่อกับ Arduino นั้นจึงนิยมต่อเพียง 4 ขาหรือ 4 บิตเท่านั้น เพื่อเป็นการประหยัดขาในการต่อใช้งานไปไว้ต่อกับอุปกรณ์อื่น ตัวอย่างเช่น Arduino UNO R3 นั้นมีขาให้ใช้งานค่อนข้างน้อย



รูปที่ 6.7 ขาที่ใช้ในการเชื่อมต่อกับ Arduino UNO R3
(ที่มา www.Thaieasyelec.com)

| <u>VR 10 KOhm</u> | <u>LCD</u> | <u>Arduino</u> |
|-------------------|------------|-------------------------------|
| GND | VSS/GND | Ground |
| VCC | VDD | +5VDC |
| Signal | VO/VEE | - |
| - | RS | Digital Pin 12 |
| - | RW | Ground (เพราะเราต้องการเขียน) |
| - | E/EN | Digital Pin 11 |
| - | DB4 | Digital Pin 4 |
| - | DB5 | Digital Pin 5 |
| - | DB6 | Digital Pin 6 |
| - | DB7 | Digital Pin 7 |
| - | A | +5VDC |
| - | K | Ground |

ตารางที่ 6.1 ขาที่ใช้ในการเชื่อมต่อกับ Arduino UNO R3
(ที่มา www.Thaieasyelec.com)



รูปที่ 6.8 การรับ-ส่งข้อมูลแบบ I2C BUS

(ที่มา www.loxhop.com)

ลำดับขั้นตอนการการรับ-ส่งข้อมูลแบบ I2C BUS มีดังนี้

1. MCU จะทำการส่งสถานะเริ่มต้น (START Conditions) เพื่อแสดงการขอใช้บัส
2. แล้วตามด้วยรหัสควบคุม (Control Byte) ซึ่งประกอบด้วยรหัสประจำตัวอุปกรณ์ Device ID, Device Address และ Mode ในการเขียนหรืออ่านข้อมูล
3. เมื่ออุปกรณ์รับทราบว่า MCU ต้องการจะติดต่อกับ ก็ต้องส่งสถานะรับรู้ (Acknowledge) หรือแจ้งให้ MCU รับรู้ว่าข้อมูลที่ได้ส่งมามีความถูกต้อง
4. เมื่อสิ้นสุดการส่งข้อมูล MCU จะต้องส่งสถานะสิ้นสุด (STOP Conditions) เพื่อบอกกับอุปกรณ์ว่าสิ้นสุดการใช้บัส

6.3 การเชื่อมต่อ Arduino กับจอ Character LCD

1. การเชื่อมต่อแบบขนาน - เป็นการเชื่อมต่อจอ LCD เข้ากับบอร์ด Arduino โดยตรง โดยจะแบ่งเป็นเชื่อมต่อแบบ 4 บิต และการเชื่อมต่อแบบ 8 บิต ใน Arduino จะนิยมเชื่อมต่อแบบ 4 บิต เนื่องจากใช้ต่อน้อยกว่า
2. การเชื่อมต่อแบบอนุกรม - เป็นการเชื่อมต่อกับจอ LCD ผ่านโมดูลแปลงรูปแบบการเชื่อมต่อกับจอ LCD จากแบบขนานมาเป็นการเชื่อมต่อแบบอื่นที่ใช้สายน้อยกว่า เช่น การใช้โมดูล I2C Serial Interface จะเป็นการนำโมดูลเชื่อมเข้ากับตัวจอ LCD แล้วใช้บอร์ด Arduino เชื่อมต่อกับบอร์ดโมดูลผ่านโปรโตคอล I2C ทำให้ใช้สายเพียง 4 เส้นก็สามารถทำให้หน้าจอแสดงผลข้อความต่างๆออกมาได้

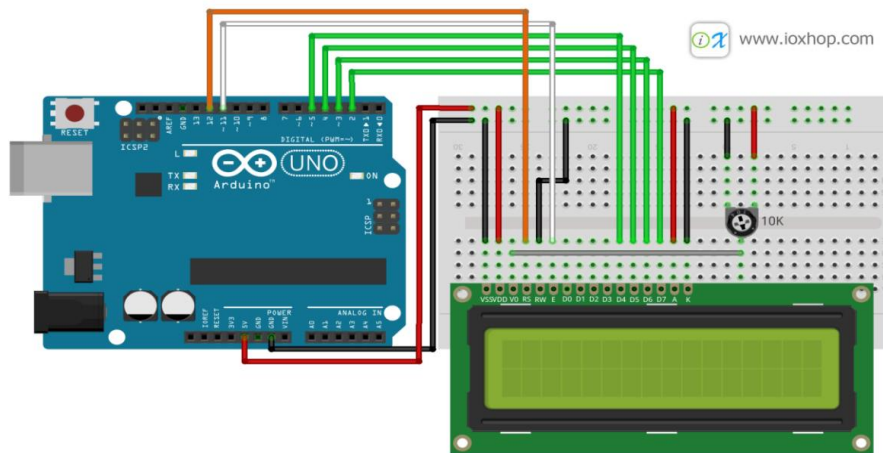


รูปที่ 6.9 รายละเอียดการต่อขาจอ LCD 16x2 Character (I2C)

(ที่มา www.Thaieasyelec.com)

6.3.1 การเชื่อมต่อ Arduino กับจอ LCD แบบขนาน

การเชื่อมต่อแบบขนาน 4 บิต สามารถทำได้ตามวงจรด้านล่างนี้



รูปที่ 6.10 การใช้งาน Character LCD กับ Arduino การเชื่อมต่อแบบขนาน (ที่มา www.ioxhop.com)

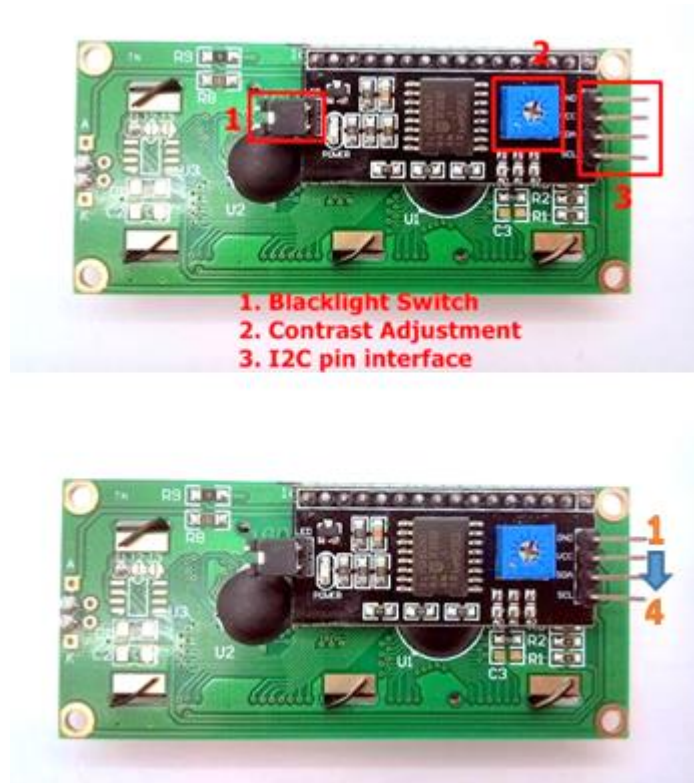
เมื่อต่อวงจรเรียบร้อยแล้วต่อสาย USB เข้ากับบอร์ด Arduino จะเห็นกล่องสี่เหลี่ยมทั้งหมด 16 ตัว (หากเป็นจอ 16x2) ในบรรทัดแรกหากไม่พบกล่องให้ปรับความชัดได้จาก VR ที่ต่ออยู่กับขา V0

| Pin No | Symbol | Description |
|--------|---------|---|
| 1 | VSS/GND | Ground |
| 2 | VDD | +5VDC |
| 3 | VO/VEE | LCD Control สำหรับปรับความเข้มของตัวอักษร |
| 4 | RS | Register Select เป็นขาอินพุตสำหรับเลือกเขียนอ่านข้อมูลในรีจิสเตอร์ |
| 5 | RW | Read/Write เป็นขาอินพุตสำหรับเลือกโหมดเขียนหรืออ่านข้อมูล |
| 6 | E/EN | Enable เป็นขาอินพุตสำหรับสัญญาณ Pulse เมื่อต้องการเขียนหรืออ่านข้อมูล |
| 7 | DB0 | Data Pins 8-Bit |
| 8 | DB1 | |
| 9 | DB2 | |
| 10 | DB3 | |
| 11 | DB4 | |
| 12 | DB5 | |
| 13 | DB6 | |
| 14 | DB7 | |
| 15 | A | (LED+) เป็นขา <u>Vcc</u> สำหรับ LED backlight (5V) |
| 16 | K | (LED-) เป็นขา <u>Gnd</u> สำหรับ LED backlight (Gnd) |

ตารางที่ 6.2 ตารางขาของจอ LCD 16x2 แบบขนาน (ที่มา www.ioxhop.com)

6.3.2 การเชื่อมต่อ Arduino กับจอ LCD แบบอนุกรม

จอ LCD ที่มีการเชื่อมแบบ I2C หรือเรียกอีกอย่างว่าการเชื่อมต่อแบบ Serial เป็นจอ LCD ธรรมดาทั่วไปที่มาพร้อมกับบอร์ด I2C Bus ที่ทำให้การใช้งานได้สะดวกยิ่งขึ้นและมาพร้อมกับ VR สำหรับปรับความเข้มของจอในรูปแบบ I2C ใช้ขาในการเชื่อมต่อกับ Arduino เพียง 4 ขา (แบบ Parallel ใช้ 16 ขา) ซึ่งทำให้ใช้งานได้ง่ายและสะดวกมากยิ่งขึ้น

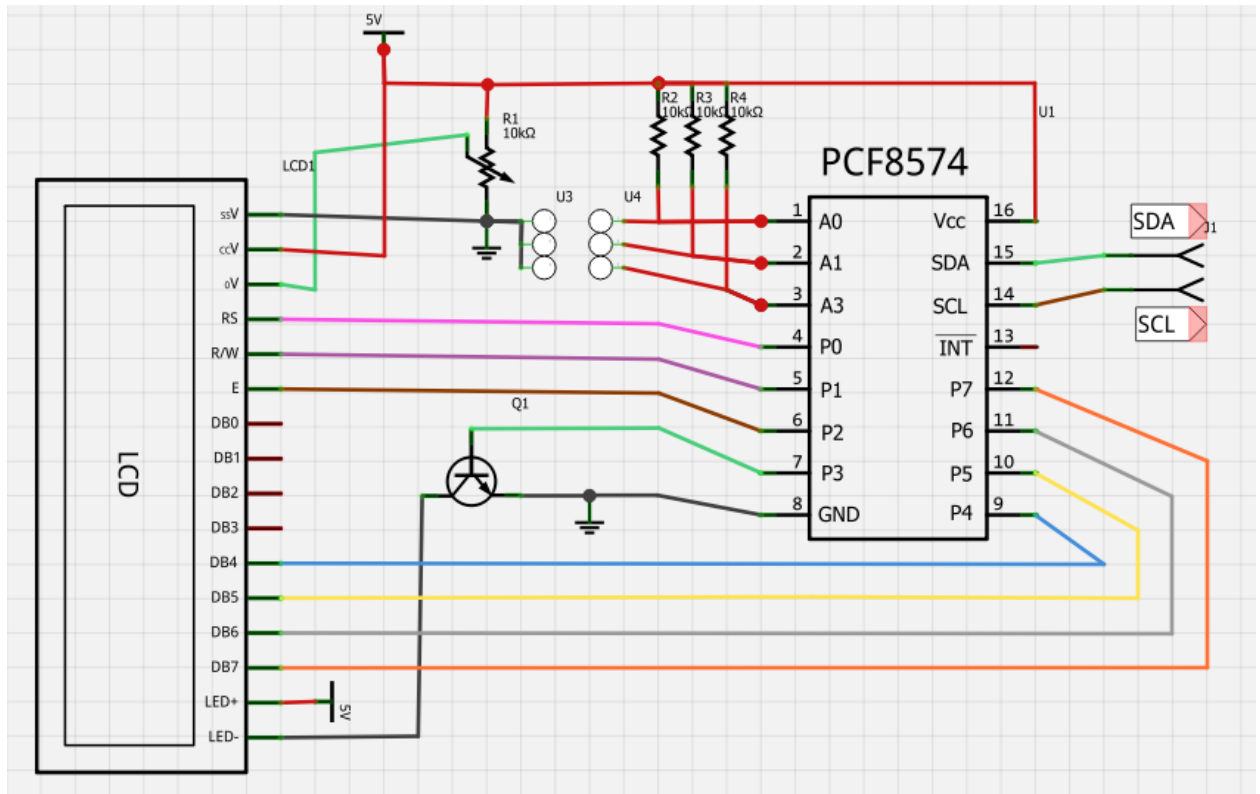


ตารางที่ 6.11 แสดงขาของจอ LCD 16x2 แบบอนุกรม (I2C)
(ที่มา www.loxhop.com)

| Arduino UNO R3 | LCD (I2C) |
|----------------|----------------------------|
| GND | GND (Pin 1) |
| +5VDC | VCC (Pin 2) |
| A4 (SDA) | SDA (Pin 3 Serial Data) |
| A5 (SCL) | SCL (Pin 4 Serial Clock) |

ตารางที่ 6.3 ตารางขาของจอ LCD 16x2 แบบขนาน
(ที่มา www.loxhop.com)

สำหรับการเชื่อมต่อสัญญาณระหว่าง Arduino กับ LCD ที่มีบอร์ด I2C อยู่แล้วนั้น การส่งข้อมูลจาก Arduino ถูกส่งออกมาในรูปแบบ I2C ไปยังบอร์ด I2C และบอร์ดจะมีหน้าที่จัดการข้อมูลให้ออกมาในรูปแบบปกติ หรือแบบ Parallel เพื่อใช้ในการติดต่อไปยังจอ LCD โดยที่รหัสคำสั่งที่ใช้ในการสั่งงานจอ LCD ยังคงไม่ต่างกับจอ LCD ที่เป็นแบบ Parallel โดยส่วนใหญ่บอร์ด I2C จะเชื่อมต่อกับตัวควบคุมของจอ LCD เพียง 4 บิตเท่านั้น วงจรภายในระหว่างจอ LCD กับบอร์ด I2C นั้น มีการต่อไว้ดังนี้



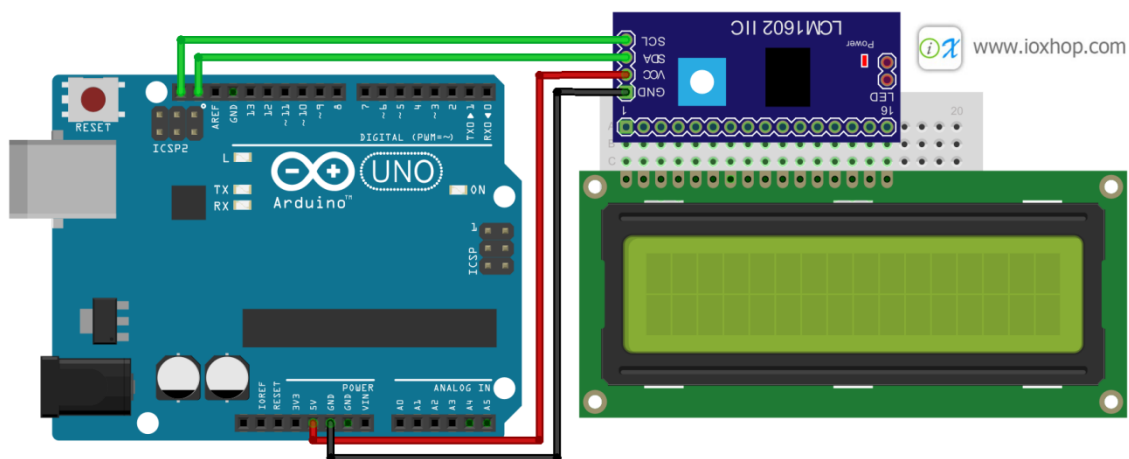
รูปที่ 6.12 การเชื่อมต่อระหว่าง Arduino กับ LCD (I2C)
(ที่มา www.loxhop.com)

จากรูปที่ 6.12 วงจรจอ LCD และบอร์ด I2C ได้มีการเชื่อมต่อขาสำหรับการรับส่งข้อมูลเป็นแบบ 4 บิต ขาที่เชื่อมต่อไว้คือ ขา P4 > DB4, P5 > DB5, P6 > DB6, P7 > DB7 และขา P2 > E (Enable), P1 > R/W, P0 > RS รวมไปถึงตัวต้านทานสำหรับปรับค่าความเข้มของตัวอักษร และ Switch Backlight จากวงจรขาที่จำเป็นในการใช้งานถูกเชื่อมต่อเข้ากับตัวบอร์ด I2C และอุปกรณ์อิเล็กทรอนิกส์เรียบร้อยแล้ว



รูปที่ 6.13 โมดูล I2C Serial Interface Board Module
(ที่มาจาก www.ioxhop.com)

อีกลักษณะหนึ่งของการเชื่อมต่อระหว่าง Arduino กับจอ LCD (I2C) แบบอนุกรมใช้งานโมดูล I2C Serial Interface Board Module มาเชื่อมต่อระหว่าง Arduino กับจอ LCD วงจรที่เชื่อมต่อจะเป็นไปตามรูปที่ 6.13 (กรณีใช้บอร์ดรุ่นอื่นต้องต่อ SDA เข้า A4 และ SCL เข้ากับ A5)



รูปที่ 6.14 การเชื่อมต่อโมดูล I2C Serial Interface Board Module
(ที่มาจาก www.ioxhop.com)

6.4 รายละเอียดคำสั่งในการสั่งงานระหว่าง Arduino กับ จอ LCD

คำสั่งในการควบคุมจอ LCD ของ Arduino นั้น ทาง Arduino.cc เขียนเป็น Library มาให้เพื่อสะดวกในการนำไปใช้งาน หลังจากต่อสายเสร็จเรียบร้อยแล้ว ขั้นตอนแรกในการเริ่มเขียนโปรแกรมคือการเรียกใช้ Library ของ LCD จากไฟล์ชื่อ LiquidCrystal.h

ฟังก์ชัน LiquidCrystal(); ใช้ประกาศขาที่ต้องการส่งข้อมูลไปยังจอ LCD รูปแบบในการสั่งงานคือ

- LiquidCrystal lcd(rs, enable, d4, d5, d6, d7)
<<< ในกรณีใช้งานแบบ 4 บิต
- LiquidCrystal lcd(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)
<<< ในกรณีใช้งานแบบ 8 บิต
- ใช้แบบ 4 บิต คือ LiquidCrystal lcd(12, 11, 4, 5, 6, 7); ก็หมายถึงการเชื่อมต่อ rs ที่ขา 12 , Enable ที่ขา 11 , และ DB4-DB7 ที่ขา 4-7 ของ Arduino ตามลำดับ

ฟังก์ชัน begin(); ใช้กำหนดขนาดของจอ ใช้ขนาด 16 ตัวอักษร 2 บรรทัด จึงประกาศเป็น lcd.begin(16, 2);

ฟังก์ชัน setCursor(); ใช้กำหนดตำแหน่งและบรรทัดของ Cursor เช่น lcd.setCursor(0, 1); คือให้เคอร์เซอร์ไปที่ตำแหน่งที่ 0 บรรทัดที่ 1 การนับตำแหน่งเริ่มจาก 0 ดังนั้น LCD 16x2 มีตำแหน่ง 0 – 15 บรรทัดคือ 0 กับ 1

ฟังก์ชัน print(); ใช้กำหนดข้อความที่ต้องการแสดง เช่น lcd.print("TEST"); คือให้แสดงข้อความ "TEST" ออกทางหน้าจอ LCD

โปรแกรมที่ 6.1

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // RS, E, D4, D5, D6, D7
void setup() {
  lcd.begin(16, 2); // จอกว้าง 16 ตัวอักษร 2 บรรทัด
  lcd.print("LCDisplay"); // แสดงผลคำว่า Hello, world! ออกหน้าจอ
  lcd.setCursor(0, 1); // เลื่อนเคอร์เซอร์ไปบรรทัดที่ 2 ลำดับที่ 0 (ก่อนหน้าตัวอักษรแรก)
  lcd.print("TEST"); // แสดงผลคำว่า TEST
  delay(3000); // หน่วงเวลา 3 วินาที
  lcd.clear(); // ล้างหน้าจอ
}
void loop() {
  lcd.setCursor(0, 0);
  lcd.print(" InFunction ");
}
```



```

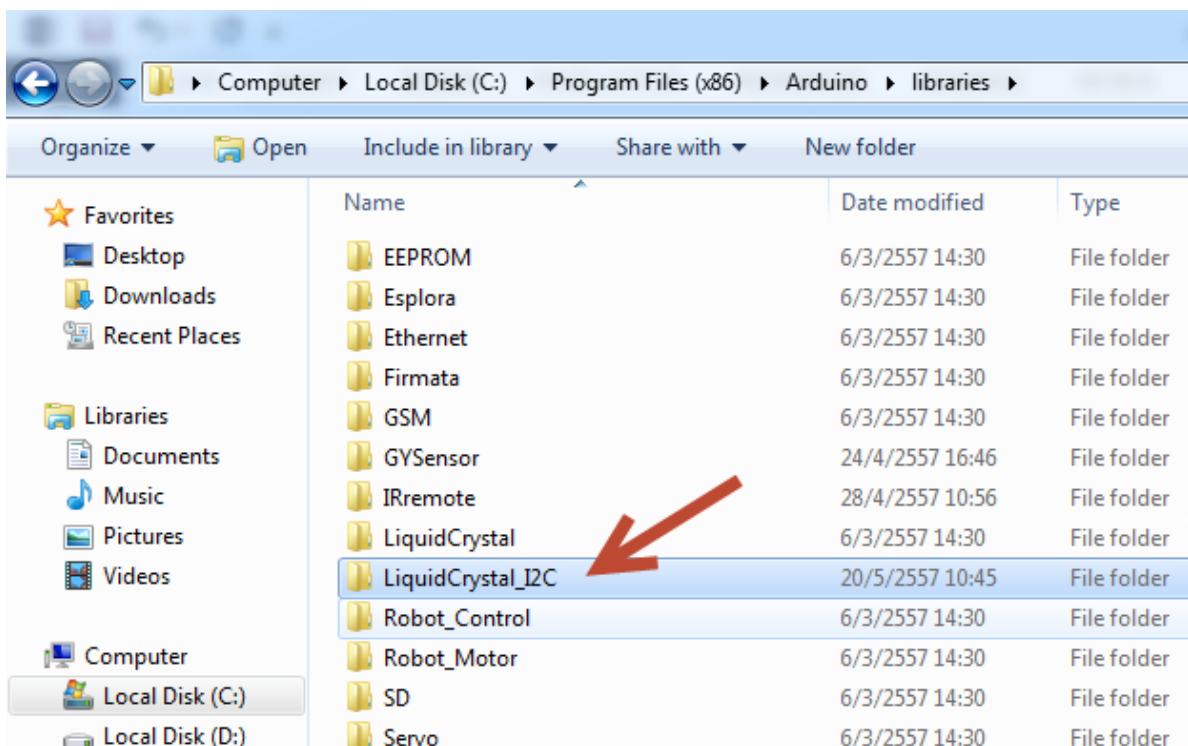
lcd.setCursor(0, 1);
lcd.print(" void loop(){ ");
delay(500); // หน่วงเวลา 0.5 วินาที
lcd.clear(); // ล้างหน้าจอ
delay(500); // หน่วงเวลา 0.5 วินาที
}

```

รายละเอียดชุดคำสั่งที่ใช้ในการสั่งงานระหว่าง Arduino กับจอ LCD คำสั่งที่ใช้ในการควบคุมจอ LCD ของ Arduino แบบ I2C นั้นไม่ต่างจากจอ LCD แบบธรรมดา (Parallel) มากนัก ทั้งนี้ยังได้มีการพัฒนา Library I2C มาให้ใช้งานได้อย่างสะดวกมากยิ่งขึ้น

6.5 ขั้นตอนการติดตั้งไลบรารีจอแสดงผล LCD

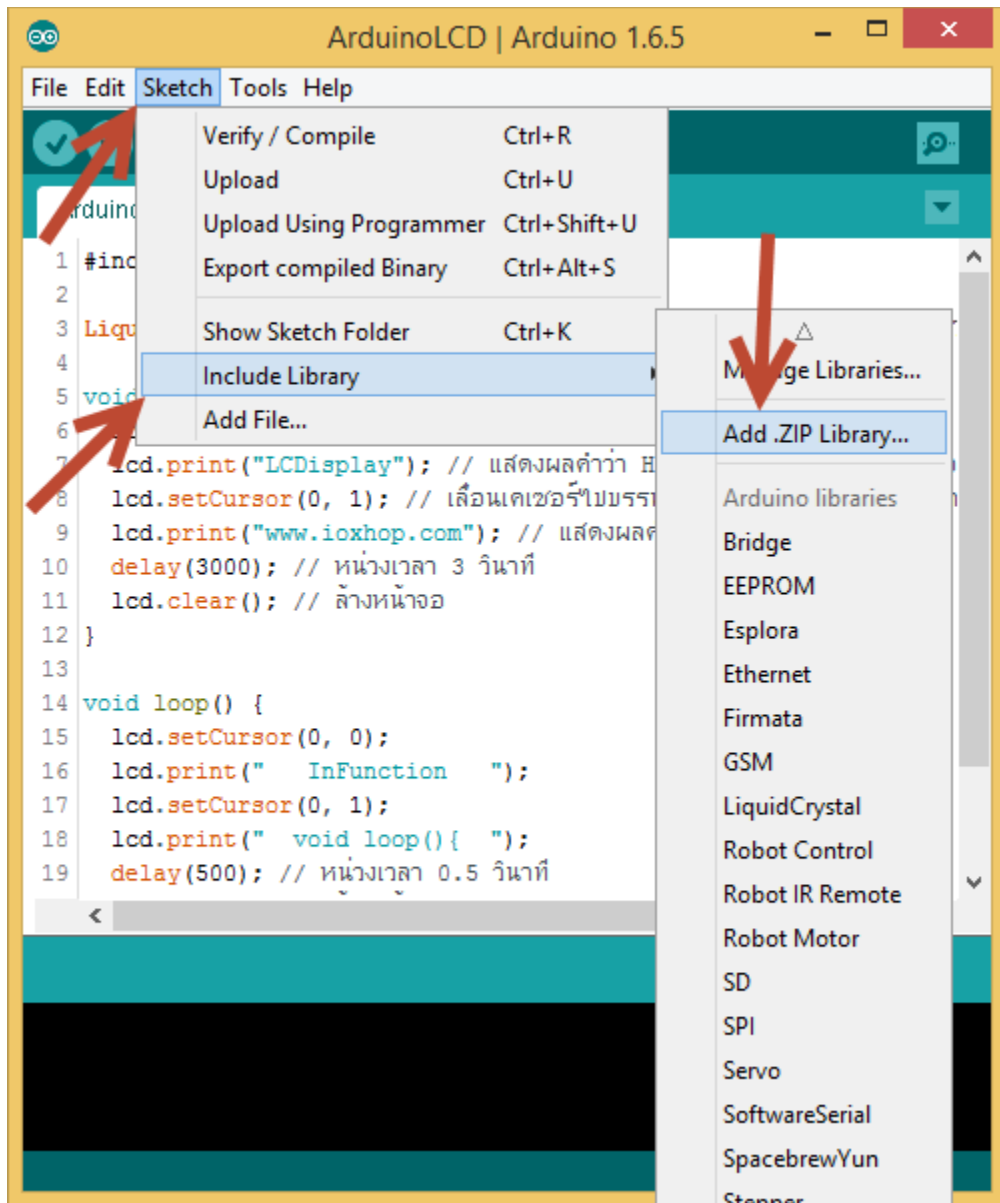
- เริ่มต้นจากการนำไฟล์ Library LCD (I2C) ไปไว้ใน Library ของ Arduino ก่อนเพื่อให้ง่ายต่อการเรียกใช้ Library และทำการแยกไฟล์ที่โหลดมาข้างต้น แล้วนำไปเก็บยัง
C:\ProgramFiles(x86)\Arduino\libraries (Drive ที่ได้ทำการติดตั้งโปรแกรม Arduino IDE)



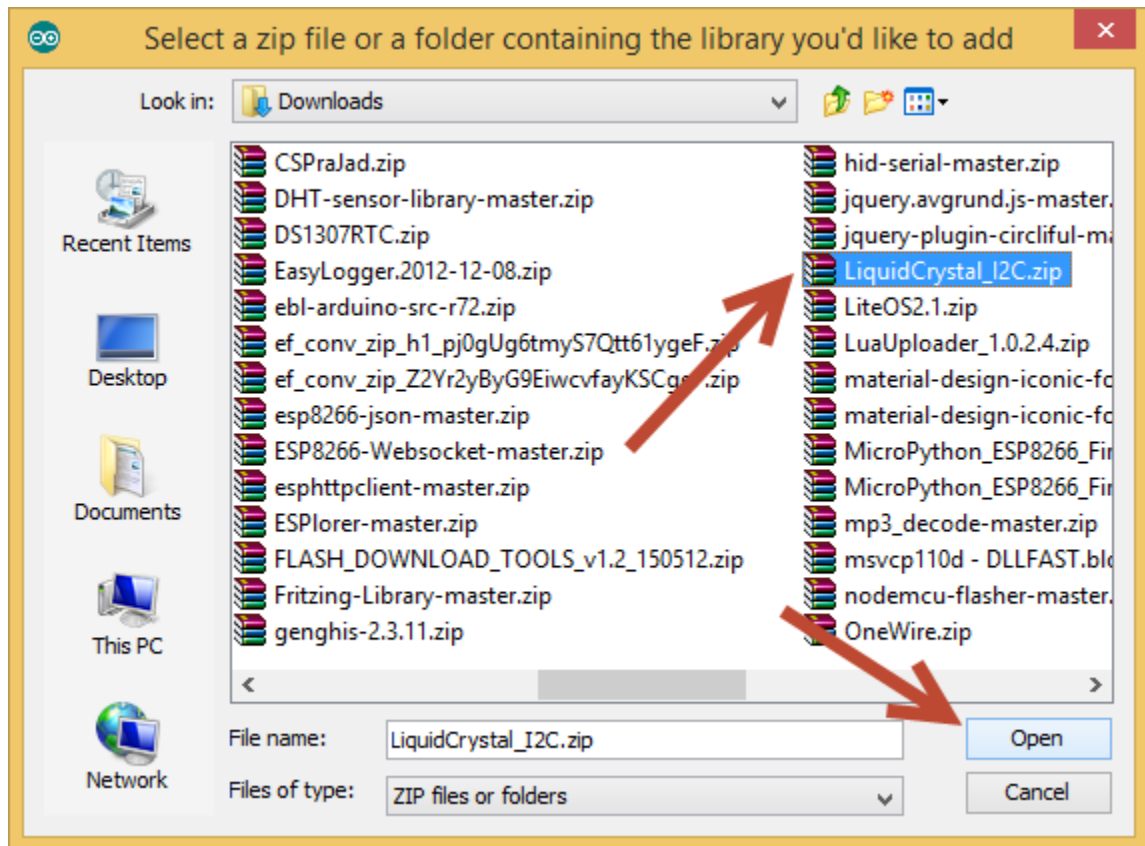
รูปที่ 6.14 การโหลด Library LCD (I2C)

- เรียก Library ของการสื่อสารแบบ I2C และ Library ของจอ LCD (I2C) เปิดโปรแกรม Arduino

IDE ขึ้นมา จากนั้นกดไปที่ Tool > Include Library > Add .ZIP Library



รูปที่ 6.15 เรียก Library ของการสื่อสารแบบ I2C และ Library ของจอ LCD (I2C)



รูปที่ 6.16 การเลือกไฟล์ที่ได้ดาวน์โหลดไว้

เลือกไฟล์ที่ได้ดาวน์โหลดไว้จากนั้นกดปุ่ม Open

6.6 ฟังก์ชันสั่งงานจอ LCD

lcd.clear()

> ใช้ล้างหน้าจอ เมื่อมีตัวอักษรใดๆอยู่บนหน้าจอ จะถูกล้างออกทั้งหมด

lcd.home()

> ใช้ปรับให้เคเซอร์กลับไปอยู่ที่ตำแหน่งแรกด้านซ้าย เมื่อใช้คำสั่ง lcd.print() จะไปเริ่มแสดงผลทางด้านบนซ้าย

lcd.setCursor(ลำดับตัวอักษรนับจากทางซ้าย, บรรทัด)

> ใช้ตั้งค่าเคเซอร์ เช่น lcd.setCursor(2, 0); หมายถึงเซตเคเซอร์ไปตัวอักษรที่ 2 นับจากทางซ้าย และอยู่บรรทัดแรก เมื่อใช้คำสั่ง lcd.print() ตัวอักษรตัวแรกจะอยู่ลำดับที่ 3 นับจากทางซ้าย

lcd.write(ข้อมูลที่ต้องการเขียนออกไป)

> ใช้สำหรับเขียนข้อมูลออกไปทีละตัวอักษร

lcd.print(ข้อมูลที่ต้องการให้เขียนออกไป [, รูปแบบข้อมูล])

> ใช้เขียนข้อมูลออกไปทั้งข้อความ

lcd.cursor()

> ใช้สั่งให้แสดงเคอร์เซอร์บนหน้าจอ

lcd.noCursor()

> ใช้สั่งให้ไม่แสดงเคอร์เซอร์บนหน้าจอ

lcd.display()

> แสดงตัวอักษรบนหน้าจอ

lcd.noDisplay()

> ปิดการแสดงตัวอักษรในหน้าจอ

lcd.scrollDisplayLeft()

> เลื่อนตัวอักษรไปทางซ้าย 1 ตัว

lcd.scrollDisplayRight()

> เลื่อนตัวอักษรไปทางขวา 1 ตัว

lcd.autoscroll()

> เลื่อนตัวอักษรไปทางขวาอัตโนมัติหากใช้คำสั่ง lcd.print() หรือ lcd.write() เมื่อตัวอักษรเต็มหน้าจอ

lcd.noAutoscroll()

> ปิดการเลื่อนตัวอักษรอัตโนมัติ

lcd.leftToRight()

> เมื่อใช้คำสั่ง lcd.print() หรือ lcd.write() ตัวอักษรจะเขียนจากซ้ายไปขวา

lcd.rightToLeft()

> เมื่อใช้คำสั่ง lcd.print() หรือ lcd.write() ตัวอักษรจะเขียนจากขวาไปซ้าย

โปรแกรมที่ 6.2

```
#include <LiquidCrystal.h>
#define BacklightPin 7
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
  pinMode(BacklightPin, OUTPUT);
  digitalWrite(BacklightPin, HIGH);
  lcd.begin(16, 2);
  lcd.print("LCD Safe Mode"); // แสดงคำว่า LCD Safe Mode ออกทางหน้าจอ
}
void loop() {
```

```

LCD_ON(); // เปิดหน้าจอ
delay(1000); // หน่วงเวลา 1 วินาที
LCD_OFF(); // ปิดหน้าจอ
delay(1000); // หน่วงเวลา 1 วินาที
}

void LCD_ON() {
  lcd.display(); // เปิดการแสดงผลตัวอักษร
  digitalWrite(BacklightPin, HIGH); // เปิดไฟแบล็กไลท์
}

void LCD_OFF() {
  lcd.noDisplay(); // ปิดการแสดงผลตัวอักษร
  digitalWrite(BacklightPin, LOW); // ปิดไฟแบล็กไลท์
}

```

โปรแกรมที่ 6.3

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Set the LCD address to 0x27 in PCF8574 by NXP and Set to 0x3F in PCF8574A by Ti
LiquidCrystal_I2C lcd(0x3F, 16, 2);

void setup() {
  lcd.begin();
  lcd.print("LCD Safe Mode"); // แสดงคำว่า LCD Safe Mode ออกทางหน้าจอ
}

void loop() {
  LCD_ON(); // เปิดหน้าจอ
  delay(1000); // หน่วงเวลา 1 วินาที
  LCD_OFF(); // ปิดหน้าจอ
  delay(1000); // หน่วงเวลา 1 วินาที
}

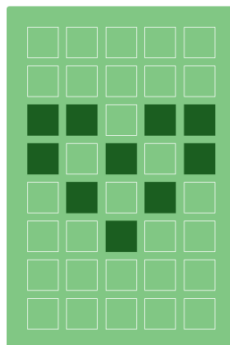
void LCD_ON() {
  lcd.display(); // เปิดการแสดงผลตัวอักษร
  lcd.backlight(); // เปิดไฟแบล็กไลท์
}

```

```
void LCD_OFF() {  
    lcd.noDisplay(); // ปิดการแสดงผลตัวอักษร  
    lcd.noBacklight(); // ปิดไฟแบล็กไลท์  
}
```

6.7 การสร้างตัวอักษรหรือใส่รูปภาพลงจอ LCD

นอกจากจะใช้ตัวอักษร ABCD ตัวเลข เครื่องหมายต่างๆแล้ว ยังสามารถสร้างตัวอักษรเองให้เป็นลักษณะของรูปภาพได้ เพื่อเพิ่มความเข้าใจ และความสวยงามของการแสดงผลข้อมูลบนหน้าจอ การสร้างตัวอักษรสามารถสร้างได้จากลิง : [Maxpromer.github.io/LCD-Character-Creator/](https://github.com/Maxpromer/LCD-Character-Creator) อัปเดตเข้า Arduino



รูปที่ 6.17 การสร้างตัวอักษรหรือใส่รูปภาพลงจอ LCD
(ที่มา www.Maxpromer.github.io/LCD-Character-Creator/)

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Set the LCD address to 0x27 in PCF8574 by NXP and Set to 0x3F
LiquidCrystal_I2C lcd(0x3F, 16, 2);

byte customChar[] = {
  B00000,
  B00000,
  B11011,
  B10101,
  B01010,
  B00100,
  B00000,
  B00000
};

void setup() {
  lcd.begin();
  lcd.createChar(0, customChar);
  lcd.home();
  lcd.write(0);
}

void loop() { }
```

สรุปเนื้อหาสาระสำคัญ

การเชื่อมต่อสัญญาณขาข้อมูลระหว่าง Arduino กับ LCD Controller สามารถทำได้ 2 ลักษณะ คือการเชื่อมต่อแบบ 8 บิต (DB0-DB7) และการเชื่อมต่อแบบ 4 บิต (DB4-DB7) ทั้งสองแบบแตกต่างกันเพียงจำนวนขาที่ใช้คือ 8 หรือ 4 ขา สามารถทำงานได้เหมือนกัน อย่างที่แน่นอนในการส่งข้อมูลแบบ 4 ขา ผลย่อมทำได้ช้ากว่า 8 ขา แต่ไม่ได้ช้ามากจนสังเกตได้ด้วยสายตา ในการต่อกับ Arduino นั้นจึงนิยมต่อเพียง 4 ขา หรือ 4 บิตเท่านั้น เพื่อเป็นการประหยัดขาในการต่อใช้งานไปไว้ต่อกับอุปกรณ์อื่น ตัวอย่างเช่น Arduino UNO R3 นั้นมีขาให้ใช้งานค่อนข้างน้อย สำหรับการเชื่อมต่อสัญญาณระหว่าง Arduino กับ LCD ที่มีบอร์ด I2C อยู่แล้วนั้น การส่งข้อมูลจาก Arduino จะถูกส่งออกมาในรูปแบบ I2C ไปยังบอร์ด I2C และบอร์ดจะมีหน้าที่จัดการข้อมูลให้ออกมาในรูปแบบปกติ หรือแบบ Parallel เพื่อใช้ในการติดต่อไปยังจอ LCD โดยที่รหัสคำสั่งที่ใช้ในการสั่งงานจอ LCD ยังคงไม่ต่างกับจอ LCD ที่เป็นแบบ Parallel โดยส่วนใหญ่บอร์ด I2C จะเชื่อมต่อกับตัวควบคุมของจอ LCD เพียง 4 บิตเท่านั้น



แบบฝึกหัดหน่วยที่ 6

เรื่อง การแสดงผลด้วยจอ LCD ของ Arduino

ใช้เวลา 20 นาที

คำชี้แจง แบบฝึกหัดมีทั้งหมด 2 ตอน ประกอบด้วยตอนที่ 1 และตอนที่ 2 (20 คะแนน)

2. แบบฝึกหัดตอนที่ 1 เป็นคำถามแบบถูก-ผิด มีทั้งหมด 20 ข้อ (10 คะแนน)
3. แบบฝึกหัดตอนที่ 2 เป็นคำถามแบบปรนัย มีทั้งหมด 10 ข้อ (10 คะแนน)



แบบฝึกหัดตอนที่ 1

คำชี้แจง ให้ผู้เรียนกาเครื่องหมายถูก ✓ ในข้อที่คิดว่าถูก และกาเครื่องหมายผิด ✗ ในข้อที่คิดว่าผิด

- 1. lcd.home() ใช้ปรับให้เคอร์เซอร์กลับไปอยู่ที่ตำแหน่งแรกด้านขวา เริ่มแสดงผลทางด้านบนซ้าย
- 2. lcd.setCursor (ลำดับตัวอักษรนับจากทางขวา, บรรทัด) ใช้ตั้งค่าเคอร์เซอร์
- 3. lcd.write (ข้อมูลที่ต้องการเขียนออกไป) ใช้สำหรับเขียนข้อมูลออกไปทีละตัวอักษร
- 4. lcd.print (ข้อมูลที่ต้องการให้เขียนออกไป [, รูปแบบข้อมูล]) ใช้เขียนข้อมูลออกไปทั้งข้อความ
- 5. lcd.cursor() ใช้สั่งให้แสดงเคอร์เซอร์บนหน้าจอ
- 6. lcd.noCursor() ใช้สั่งให้ไม่แสดงเคอร์เซอร์บนหน้าจอ
- 7. lcd.display() แสดงตัวอักษรบนหน้าจอ
- 8. lcd.scrollDisplayLeft() เลื่อนตัวอักษรไปทางซ้าย 2 ตัว
- 9. lcd.autoscroll() เลื่อนตัวอักษรไปทางขวาอัตโนมัติ
- 10. lcd.leftToRight() เมื่อใช้คำสั่ง lcd.print() หรือ lcd.write() ตัวอักษรจะเขียนจากขวาไปซ้าย

 **แบบฝึกหัดตอนที่ 2**

คำชี้แจง ให้ผู้เรียนเลือกคำตอบที่ถูกต้องที่สุดแล้วกาเครื่องหมายกากบาท (X) ให้ครบทุกข้อ

- เมื่อมีการปล่อยกระแสไฟฟ้าเข้าไปกระตุ้นที่ผลึกเหลวทำให้เกิด
 - ผลึกทึบแสง ทำให้แสงที่มาจากไฟ Backlight แสดง ขึ้นมาบนหน้าจอ
 - ผลึกโปร่งแสง ทำให้แสงที่มาจากไฟ Backlight แสดง ขึ้นมาบนหน้าจอ
 - ผลึกทึบแสง ทำให้แสงที่มาจากไฟ Backlight ไม่แสดง ขึ้นมาบนหน้าจอ
 - ผลึกโปร่งแสง ทำให้แสงที่มาจากไฟ Backlight ไม่แสดง ขึ้นมาบนหน้าจอ
- การระบุขนาดจอ Graphic LCD ระบุในลักษณะอย่างไร
 - ตัวอักษรใส่ได้
 - จำนวนจุด (Pixels)
 - สีที่แตกต่างกันตามสีของผลึกคริสตอล
 - สีพื้นหลังที่ต่างกัน
- ขา D0-D7 คือ
 - LCD Control Pin
 - Read/Write Pin
 - Data Pin
 - Enable Pin
- ผลึกเหลวที่ใช้โดยทั่วไปจะเป็นแบบ
 - Magnetic
 - Resistance
 - Inductance
 - Capacitance
- จอ LCD 16x2 Character ที่นิยามวางจำหน่ายจะมีอยู่ 2 แบบด้วยกันคือ
 - แบบขนาน และ ผสม I2C
 - แบบผสม และ อนุกรม I2C
 - แบบขนาน และ อนุกรม I2C
 - แบบอนุกรม และ อนุกรม I2C
- การเชื่อมต่อแบบขนาน นิยมเชื่อมต่อกี่แบบ
 - 4 บิต และ 8 บิต
 - 1 บิต และ 2 บิต
 - 2 บิต และ 4 บิต
 - 1 บิต และ 8 บิต

7. RS เป็นสัญญาณกำหนดให้ LCD ทราบว่าอุปกรณ์ภายนอกต้องการติดต่อกับ LCD โดย
- ก. RS = “1” หมายถึง คำสั่ง , RS = “0” หมายถึง ข้อมูล
 - ข. RS = “0” หมายถึง คำสั่ง , RS = “1” หมายถึง ข้อมูล
 - ค. RS = “0” หมายถึง เขียน , RS = “1” หมายถึง อ่าน
 - ง. RS = “0” หมายถึง อ่าน , RS = “1” หมายถึง เขียน
8. ถ้า R/W = “0” หมายความว่า
- ก. สัญญาณ DB3-DB0 จะส่งจากอุปกรณ์ภายนอกมาที่ LCD
 - ข. สัญญาณ DB7-DB0 จะส่งจากอุปกรณ์ภายนอกมาที่ LCD
 - ค. สัญญาณ DB3-DB0 จะส่งไปยังอุปกรณ์ภายนอกมาที่ LCD
 - ง. สัญญาณ DB7-DB0 จะส่งไปยังอุปกรณ์ภายนอกมาที่ LCD
9. ถ้า R/W = “1” หมายความว่า
- ก. สัญญาณ DB3-DB0 จะส่งจากอุปกรณ์ภายนอกมาที่ LCD
 - ข. สัญญาณ DB7-DB0 จะส่งจากอุปกรณ์ภายนอกมาที่ LCD
 - ค. สัญญาณ DB3-DB0 จะส่งไปยังอุปกรณ์ภายนอกมาที่ LCD
 - ง. สัญญาณ DB7-DB0 จะส่งไปยังอุปกรณ์ภายนอกมาที่ LCD
10. การเชื่อมต่อแบบอนุกรม ใช้โมดูลอะไรในการเชื่อมเข้ากับตัวจอ LCD
- ก. ICC Serial Interface
 - ข. I2C Serial Interface
 - ค. IIC Serial Interface
 - ง. C2I Serial Interface



คำชี้แจง ให้ผู้เรียนทุกคนทำการทดลองตามปฏิบัติการทดลองหน่วยที่ 6 เรื่อง การแสดงผลด้วยจอ LCD ของ Arduino ใช้เวลา 180 นาที (20 คะแนน)

จุดประสงค์เชิงพฤติกรรม

1. สามารถใช้ฟังก์ชันแสดงผลด้วยจอ LCD ของ Arduino ได้ถูกต้อง
2. สามารถแก้ปัญหาในการทำงานของบอร์ด Arduino Uno R3 ได้
3. สามารถต่อใช้งานและอัปโหลดโปรแกรมให้กับบอร์ด Arduino Uno R3 ได้

อุปกรณ์การทดลอง

| | | |
|--|----|---------|
| 1. เครื่องคอมพิวเตอร์และโปรแกรม Arduino IDE 1.6.9 | 1 | เครื่อง |
| 2. USB Cable Arduino Uno R3 | 1 | เส้น |
| 3. Arduino Uno R3 Board | 1 | บอร์ด |
| 4. Breadboard | 1 | แผง |
| 5. LCD Screen (compatible with Hitachi HD44780 driver) | 1 | ตัว |
| 6. Hook-up Wires | 10 | เส้น |
| 7. 10k ohm potentiometer | 1 | ตัว |
| 8. pin headers to solder to the LCD display pins | 16 | ตัว |

ข้อควรระวัง

1. ควรระวังไม่วางบอร์ด Arduino Uno R3 หรือซีลต่างๆ บนโต๊ะโลหะหรือที่วางที่เป็นโลหะเพราะอาจเกิดการลัดวงจรของภาคจ่ายไฟได้
2. ไม่ควรต่อสายต่อวงจรในบอร์ด Arduino Uno R3 ทิ้งไว้ ควรถอดสายต่อวงจรออกให้หมด เพราะผลการทดลองอาจเกิดการผิดพลาดไม่เป็นไปตามทฤษฎีได้
3. ไม่ควรถอดสายสายโหลด USB เข้าออกตลอดเวลา เพราะอาจทำให้ภาคจ่ายไฟของบอร์ด Arduino Uno R3 เสียหายได้

การทดลองที่ 6.1 autscroll

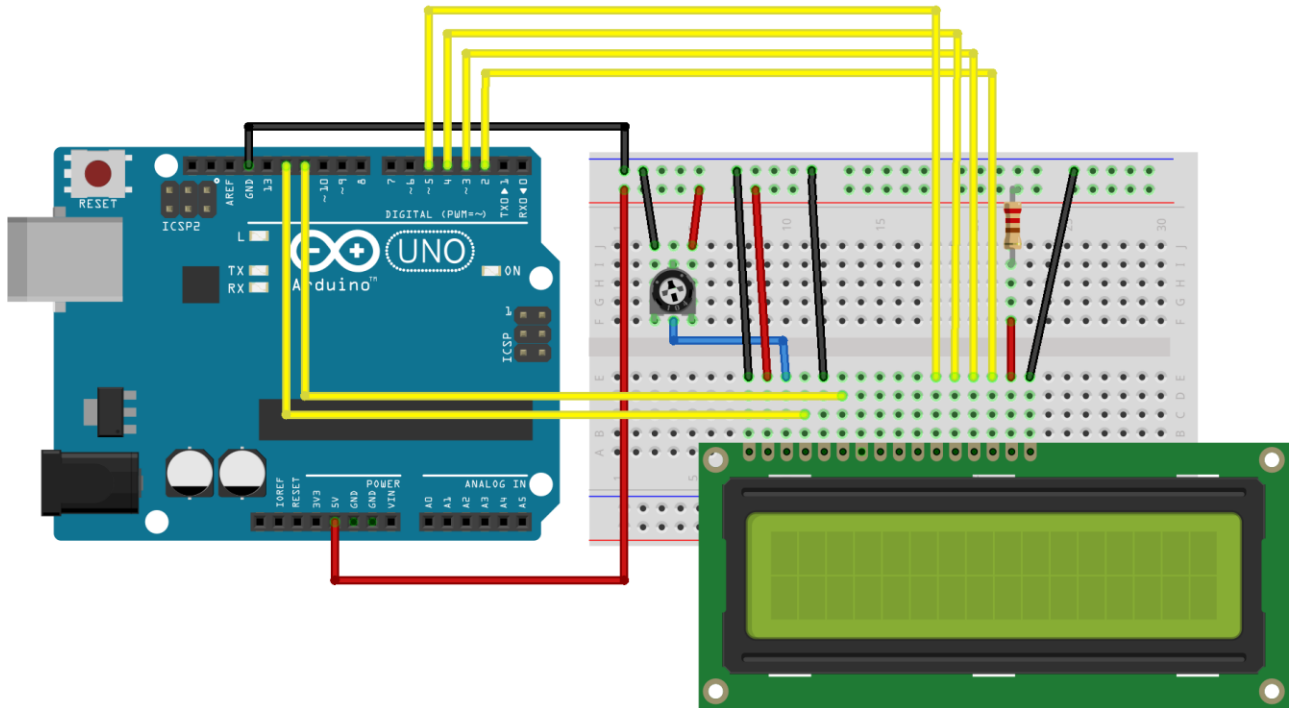
ไลบรารี LiquidCrystal ช่วยให้ควบคุมจอแสดงผล ที่มีความเข้ากันได้กับไอซีของฮิตาชิ HD44780 และมีขาอินเตอร์เฟสที่ 16 ขา ใบงานนี้แสดงให้เห็นถึงวิธีการใช้ Autscroll () และ noAutscroll () วิธีที่จะย้ายข้อความทั้งหมดบนหน้าจอด้านซ้ายหรือขวา และเคอร์เซอร์ไปยังด้านล่างขวา

Hardware Required

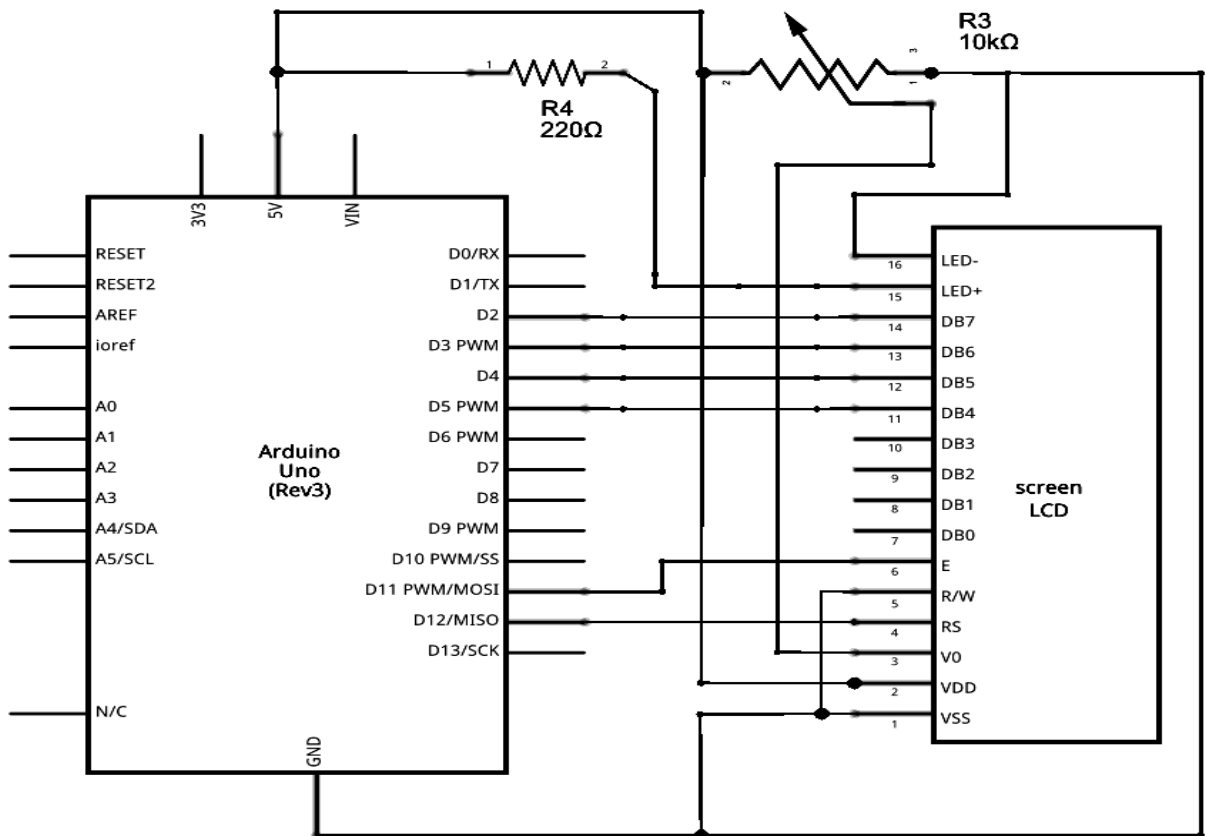
1. Arduino or Genuino Board
2. LCD Screen (compatible with Hitachi HD44780 driver)
3. pin headers to solder to the LCD display pins
4. 10k ohm potentiometer
5. 220 ohm resistor
6. hook-up wires
7. breadboard

Circuit

1. LCD RS pin to digital pin 12
2. LCD Enable pin to digital pin 11
3. LCD D4 pin to digital pin 5
4. LCD D5 pin to digital pin 4
5. LCD D6 pin to digital pin 3
6. LCD D7 pin to digital pin 2



Schematic



Code

```
// http://www.arduino.cc/en/Tutorial/LiquidCrystalAutoscroll
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd (12, 11, 5, 4, 3, 2);
void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
}
void loop() {
  // set the cursor to (0,0):
  lcd.setCursor(0, 0);
  // print from 0 to 9:
  for (int thisChar = 0; thisChar < 10; thisChar++) {
    lcd.print(thisChar);
    delay(500);
  }
  // set the cursor to (16,1):
  lcd.setCursor(16, 1);
  // set the display to automatically scroll:
  lcd.autoscroll();
  // print from 0 to 9:
  for (int thisChar = 0; thisChar < 10; thisChar++) {
    lcd.print(thisChar);
    delay(500);
  }
  // turn off automatic scrolling
  lcd.noAutoscroll();
  // clear screen for the next loop:
  lcd.clear();
}
```

ผลการทดลอง

.....

.....

.....

.....

.....

.....

การทดลองที่ 6.2 blink() and noBlink()

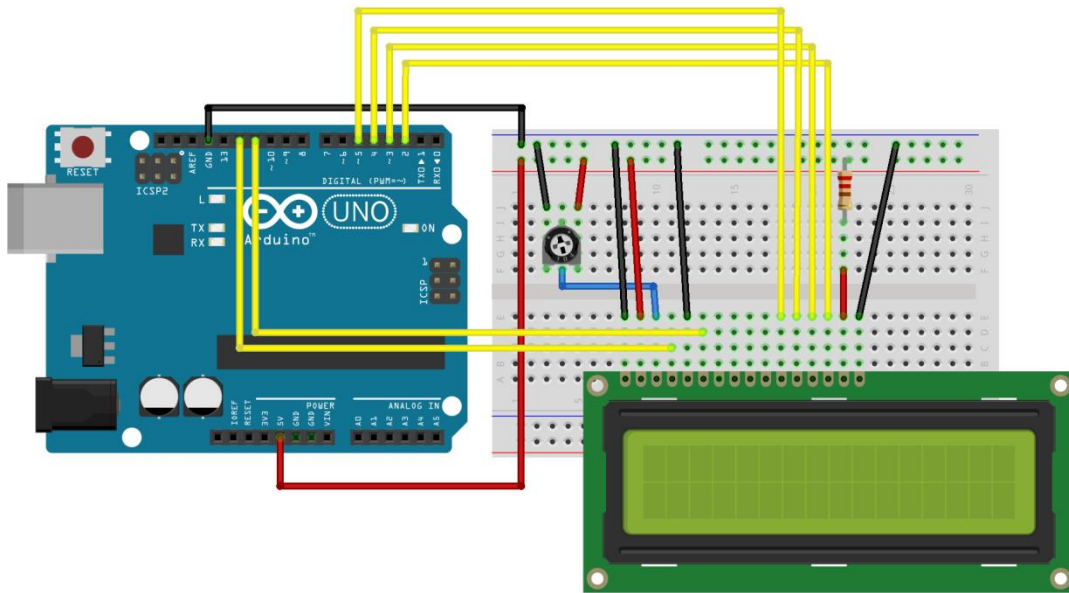
ไลบรารี LiquidCrystal ช่วยให้ควบคุมจอแสดงผลที่มีความเข้ากันได้กับไอซีของฮิตาชิ HD44780 โดยอินเทอร์เฟซที่ 16 ขา ใบงานนี้แสดงให้เห็นถึงวิธีการกระพริบข้อความ () และ การไม่กระพริบข้อความ ()

Hardware Required

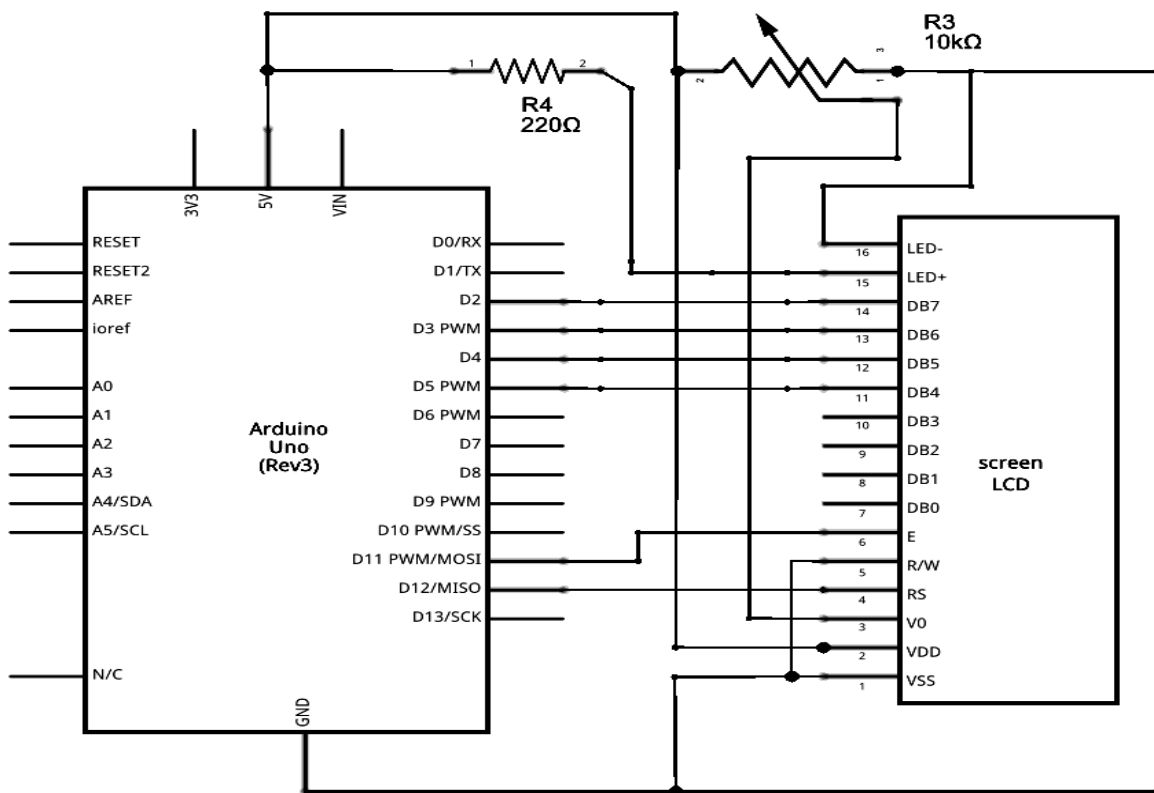
1. Arduino or Genuino Board
2. LCD Screen (compatible with Hitachi HD44780 driver)
3. pin headers to solder to the LCD display pins
4. 10k ohm potentiometer
5. 220 ohm resistor
6. hook-up wires
7. breadboard

Circuit

1. LCD RS pin to digital pin 12
2. LCD Enable pin to digital pin 11
3. LCD D4 pin to digital pin 5
4. LCD D5 pin to digital pin 4
5. LCD D6 pin to digital pin 3
6. LCD D7 pin to digital pin 2



Schematic



Code

```
/*  
  http://www.arduino.cc/en/Tutorial/LiquidCrystalBlink  
*/  
  
// include the library code:  
#include <LiquidCrystal.h>  
  
// initialize the library with the numbers of the interface pins  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  
  
void setup() {  
  // set up the LCD's number of columns and rows:  
  lcd.begin(16, 2);  
  // Print a message to the LCD.  
  lcd.print("hello, world!");  
}  
  
void loop() {  
  // Turn off the blinking cursor:  
  lcd.noBlink();  
  delay(3000);  
  // Turn on the blinking cursor:  
  lcd.blink();  
  delay(3000);  
}
```

ผลการทดลอง

.....

.....

.....

.....

.....

.....

การทดลองที่ 6.3 "Hello World!"

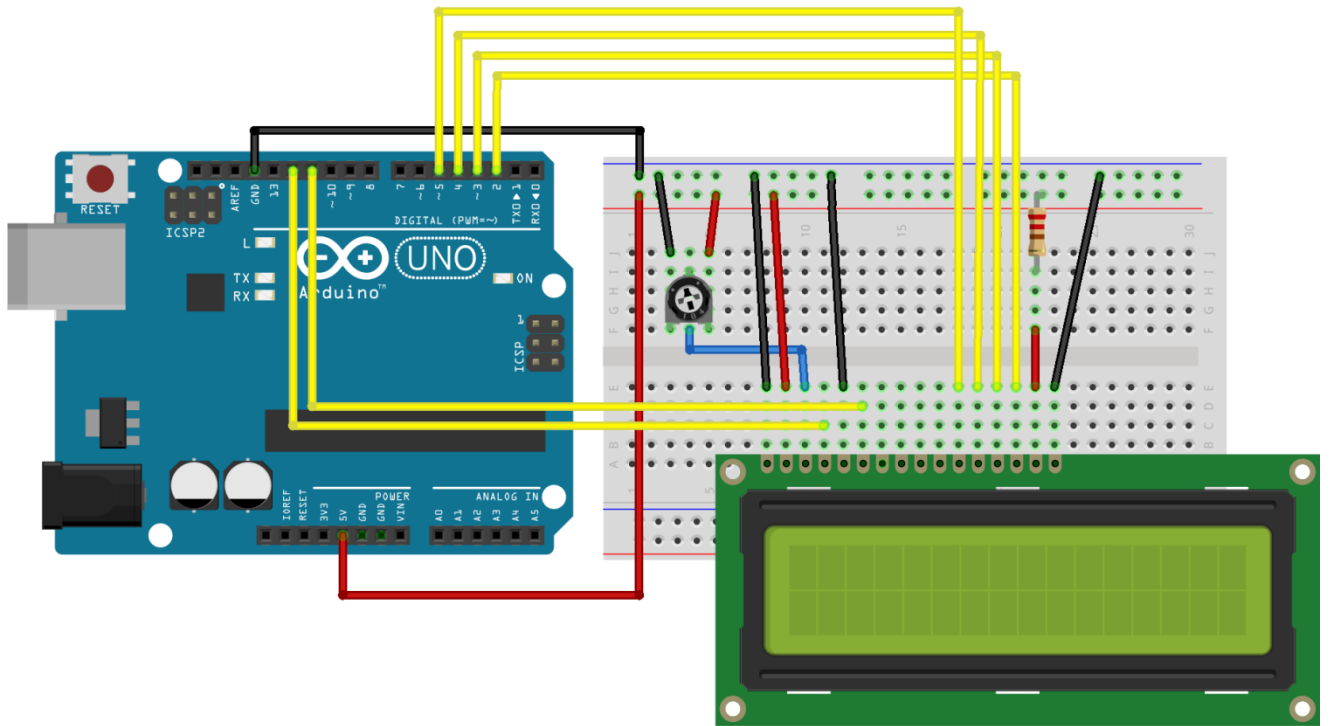
ไลบรารี LiquidCrystal ช่วยให้ควบคุมจอแสดงผลที่มีความเข้ากันได้กับไอซีขับของฮิตาชิ HD44780 โดยอินเทอร์เฟซที่ 16 ขา ใบงานนี้พิมพ์ข้อความ "Hello World!" จอแอลซีดีและแสดงเวลาเป็นวินาที ตั้งแต่ Arduino ถูกเริ่มให้ทำงาน



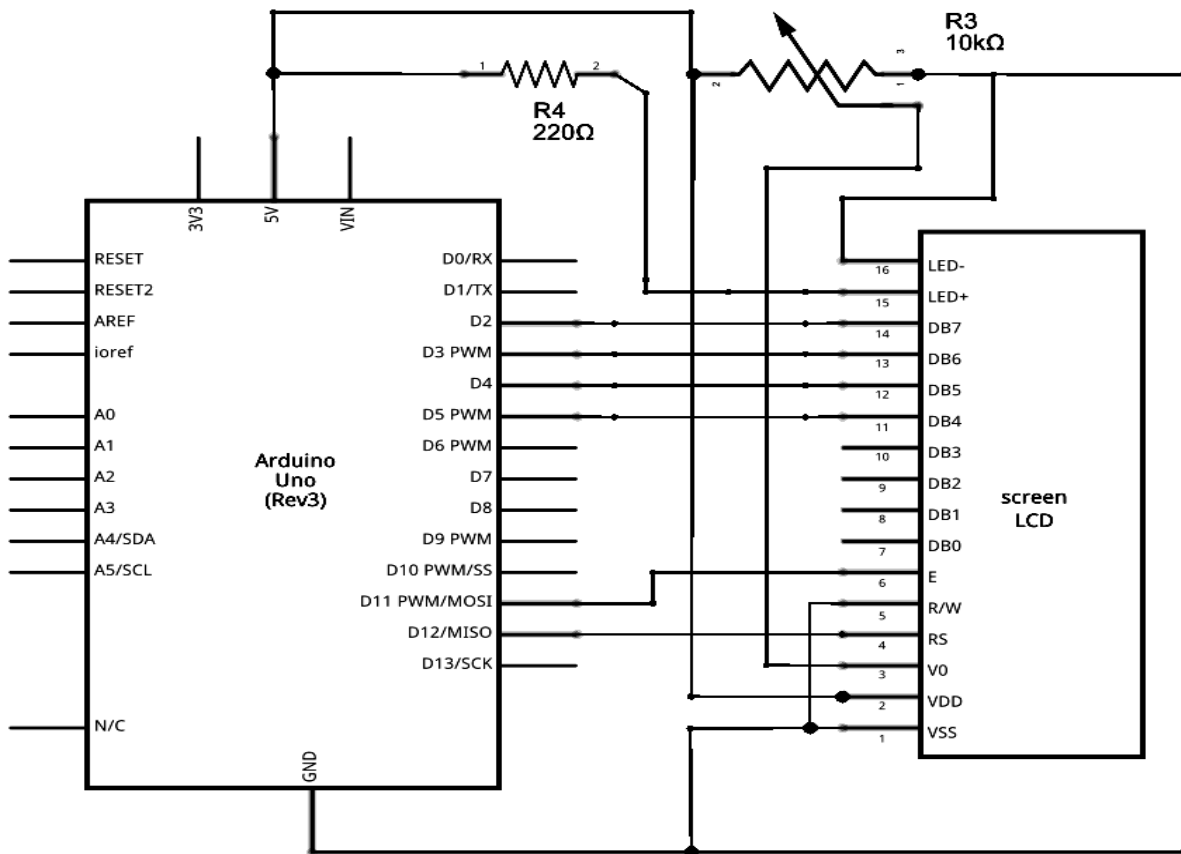
Hardware Required

1. Arduino or Genuino Board
2. LCD Screen (compatible with Hitachi HD44780 driver)
3. pin headers to solder to the LCD display pins
4. 10k ohm potentiometer
5. 220 ohm resistor
6. hook-up wires
7. breadboard

Circuit



Schematic



Code

```
/*  
  LiquidCrystal Library - Hello World  
  http://www.arduino.cc/en/Tutorial/LiquidCrystal  
  */  
  
// include the library code:  
#include <LiquidCrystal.h>  
  
// initialize the library with the numbers of the interface pins  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  
  
void setup() {  
  // set up the LCD's number of columns and rows:  
  lcd.begin(16, 2);  
  // Print a message to the LCD.  
  lcd.print("hello, world!");  
}  
  
void loop() {  
  // set the cursor to column 0, line 1  
  // (note: line 1 is the second row, since counting begins with 0):  
  lcd.setCursor(0, 1);  
  // print the number of seconds since reset:  
  lcd.print(millis() / 1000);  
}
```

ผลการทดลอง

.....
.....
.....
.....

การทดลองที่ 6.4 Serial Input

Liquid Crystal Library ช่วยให้ควบคุมจอแสดงผลที่มีความเข้ากันได้กับไอซีขับของฮิตาชิ HD44780 โดยอินเตอร์เฟซที่ 16 ขา ใบงานนี้รับการป้อนข้อมูลแบบอนุกรมจากโฮสต์คอมพิวเตอร์และแสดงบนหน้าจอ LCD ที่ใช้มันอัปโหลดข้อมูล จากนั้นเปิด Serial Monitor ข้อความปรากฏบนหน้าจอ LCD

Hardware Required

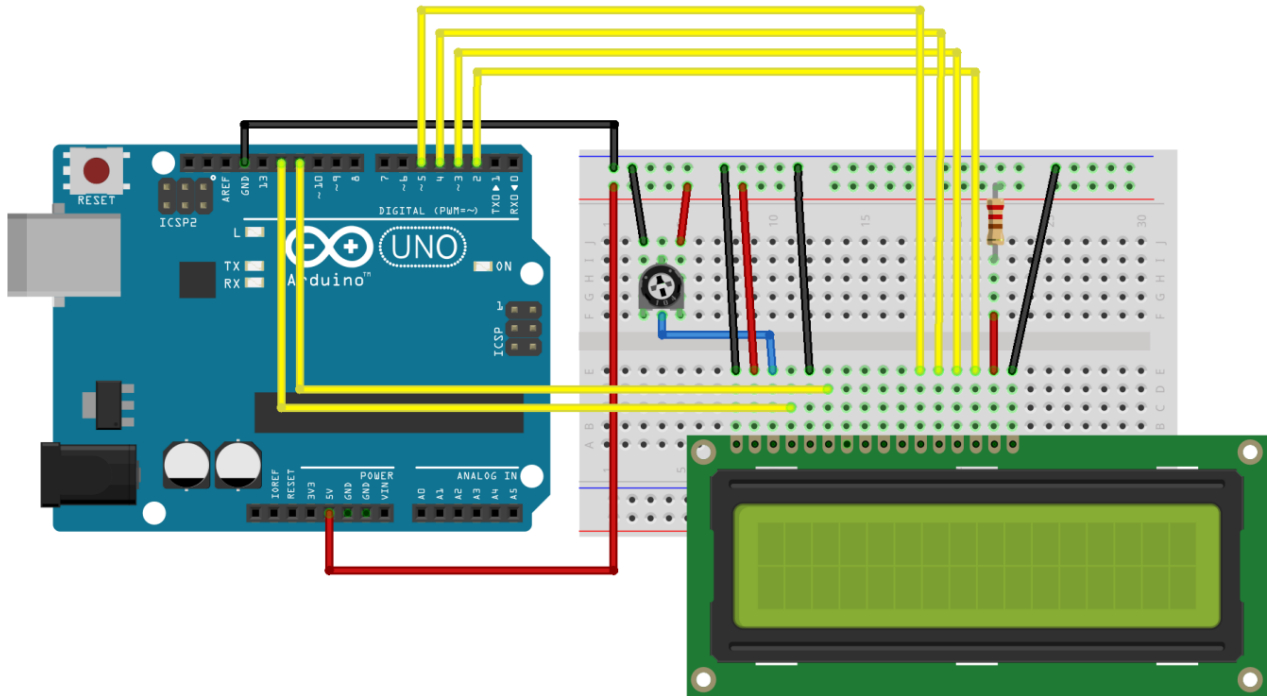
1. Arduino or Genuino Board
2. LCD Screen (compatible with Hitachi HD44780 driver)
3. pin headers to solder to the LCD display pins
4. 10k ohm potentiometer
5. 220 ohm resistor
6. hook-up wires
7. breadboard

Software

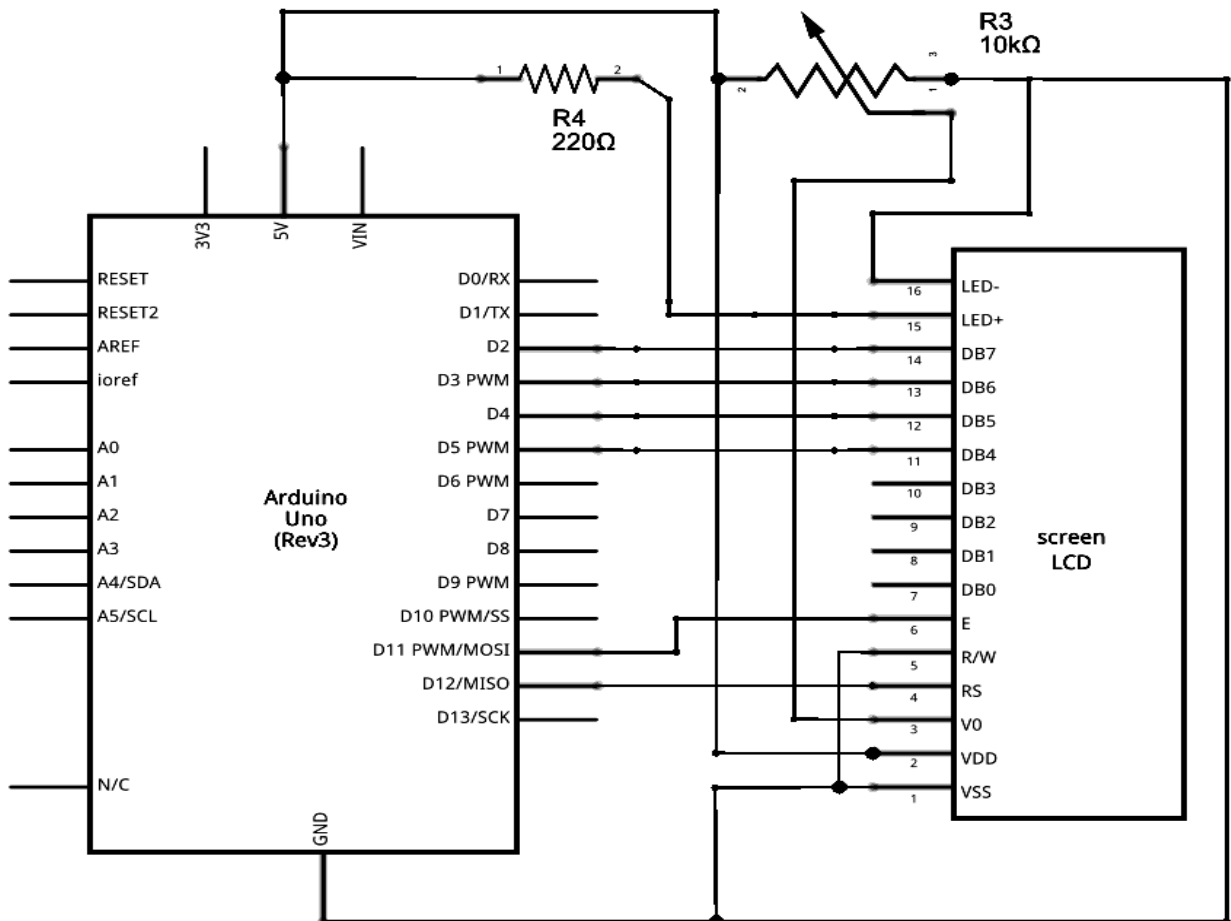
1. A computer running the Arduino Software (IDE)

Circuit

1. LCD RS pin to digital pin 12
2. LCD Enable pin to digital pin 11
3. LCD D4 pin to digital pin 5
4. LCD D5 pin to digital pin 4
5. LCD D6 pin to digital pin 3
6. LCD D7 pin to digital pin 2



Schematic



Code

```

/*
  http://www.arduino.cc/en/Tutorial/LiquidCrystalSerial
*/
// include the library code:
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // initialize the serial communications:
  Serial.begin(9600);
}
void loop() {
  // when characters arrive over the serial port...
  if (Serial.available()) {
    // wait a bit for the entire message to arrive
    delay(100);
    // clear the screen
    lcd.clear();
    // read all the available characters
    while (Serial.available() > 0) {
      // display each character to the LCD
      lcd.write(Serial.read());
    }
  }
}

```

ผลการทดลอง

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

ปัญหาอุปสรรคหรือข้อเสนอแนะ

.....

.....

ตารางการประเมินผลคะแนนภาคปฏิบัติ

| หัวข้อการพิจารณาภาคปฏิบัติ | ระดับคะแนน |
|---------------------------------------|------------|
| การทดลองที่ 6.1 autoscroll | 5 คะแนน |
| การทดลองที่ 6.2 blink() and noBlink() | 5 คะแนน |
| การทดลองที่ 6.3 Hello World! | 5 คะแนน |
| การทดลองที่ 6.4 Serial Input | 5 คะแนน |
| | |
| | |
| | |
| รวมคะแนนภาคปฏิบัติ |คะแนน |

แบบทดสอบหลังเรียน หน่วยที่ 6

เรื่อง การแสดงผลด้วยจอ LCD ของ Arduino

เรื่อง การแสดงผลด้วยจอ LCD ของ Arduino

ใช้เวลา 20 นาที

วิชา Arduino เบื้องต้น

รหัสวิชา (2127-2107)

ระดับชั้น ประกาศนียบัตรวิชาชีพ (ปวช.)

สาขาวิชา เมคคาทรอนิกส์

คำชี้แจง 1. แบบทดสอบมีทั้งหมด 10 ข้อ (10 คะแนน)

2. ให้ผู้เรียนเลือกคำตอบที่ถูกต้องที่สุดแล้วกาเครื่องหมายกากบาท (X) ลงในกระดาษคำตอบ

1. คำว่า LCD ย่อมาจากคำว่า
 - ก. Liquid Center Display
 - ข. Liquid Computer Display
 - ค. Liquid Crystal Display
 - ง. Liquid Communication Display
2. จอ LCD จะแบ่งเป็น 2 แบบใหญ่ๆตามลักษณะการแสดงผลดังนี้
 - ก. Center LCD/ Graphic LCD
 - ข. Character LCD/ Graphic LCD
 - ค. Crystal LCD/ Graphic LCD
 - ง. Computer LCD/ Graphic LCD
3. โครงสร้างของ LCD ทั่วไปจะประกอบขึ้นด้วยแผ่นแก้ว 2 แผ่นประกบกันอยู่ โดยเว้นช่องว่างตรงกลางไว้
 - ก. 6-10 ไมโครเมตร
 - ข. 10-16 ไมโครเมตร
 - ค. 16-20 ไมโครเมตร
 - ง. 20-26 ไมโครเมตร
4. จอ LCD สามารถแสดงผลให้เรามองเห็นได้ทั้งหมด 3 แบบด้วยกันคือ
 - ก. แบบใช้การสะสมแสง/แบบใช้การส่งผ่านแสง/แบบสะสมและส่งผ่านแสง
 - ข. แบบใช้การสะท้อนแสง/แบบใช้การส่งเสริมแสง/แบบสะท้อนและส่งเสริมแสง
 - ค. แบบใช้การสะท้อนแสง/แบบใช้การส่งผ่านแสง/แบบส่งผ่านและสะท้อนแสง
 - ง. แบบใช้การสะสมแสง/แบบใช้การส่งผ่านแสง/แบบสะสมและส่งผ่านแสง

5. ในการเชื่อมต่อแบบขนาน Pin 7 – 14 จอ Character LCD ใช้สำหรับ
- RS
 - RW
 - E/EN
 - Data
6. ขา E เป็นสัญญาณ Enable เมื่อมีค่าเป็น “1” หมายความว่า
- digitalWrite
 - ไม่สนใจสัญญาณ RS RW และ DB7-DB0
 - อุปกรณ์ภายนอกต้องการติดต่ออ่านหรือเขียนข้อมูล
 - pinMode
7. RW เป็นสัญญาณบอกให้ LCD ทราบว่าอุปกรณ์ภายนอกต้องการอ่านหรือเขียนกับ LCD โดย
- RW = “1” หมายถึง เขียน และ RW = “0” หมายถึง อ่าน
 - ใช้แจ้งว่าได้รับข้อมูลตัวอักษร (characters) แล้ว และพร้อมสำหรับการอ่านไปใช้งาน
 - RW = “0” หมายถึง เขียน และ RW = “1” หมายถึง อ่าน
 - ส่งค่ากลับจากฟังก์ชัน
8. ขาสัญญาณข้อมูล DB0-DB7 เป็นสัญญาณแบบ 2 ทิศทาง โดยจะสัมพันธ์กับสัญญาณ
- RS
 - E
 - RW
 - VDD
9. lcd.clear() คือฟังก์ชันอะไร
- ใช้สำหรับเขียนข้อมูลออกไปทีละตัวอักษร
 - ใช้ล้างหน้าจอ เมื่อมีตัวอักษรใดๆอยู่บนหน้าจอ จะถูกล้างออกทั้งหมด
 - ใช้เขียนข้อมูลออกไปทั้งข้อความ
 - ใช้สั่งให้แสดงเคอร์เซอร์บนหน้าจอ
10. RS ใช้บอกให้ LCD Controller ทราบว่า
- Code ที่ส่งมาทางขา Data เป็นคำสั่ง
 - Code ที่ส่งมาทางขา Data เป็นคำสั่งหรือข้อมูล
 - Code ที่ส่งมาทางขา Data เป็นข้อมูล
 - Code ที่ส่งมามีค่าเท่ากับ 11