

4

พื้นฐานของ Arduino

4.1 ความรู้เบื้องต้นเกี่ยวกับ Arduino

Arduino ถูกคิดค้นขึ้นที่ Ivrea Interaction Design Institute มันถูกออกแบบมาสำหรับการสร้างต้นแบบอย่างรวดเร็ว โดยมุ่งเป้าไปที่นักอดิเรกโดยไม่ต้องมีพื้นฐานการเขียนโปรแกรมใดๆ ในไม่ช้า แพลตฟอร์มที่ใช้งานง่ายดึงดูดผู้ชมที่ครอบคลุมชุมชนในวงกว้าง และเริ่มเปลี่ยนแปลงเพื่อปรับแนวโน้มล่าสุดในตลาด ตั้งแต่บอร์ด 8 บิตไปจนถึงผลิตภัณฑ์ IoT อุปกรณ์สวมใส่ และสภาพแวดล้อมแบบฝังตัว บอร์ด Arduino เป็นโอเพ่นซอร์สอย่างสมบูรณ์และสามารถใช้สำหรับการพัฒนาแอปพลิเคชันที่มีความต้องการเฉพาะ ซอฟต์แวร์ Arduino นั้นใช้งานง่ายและง่ายต่อการเริ่มต้นด้วยสภาพแวดล้อมที่ยืดหยุ่นสำหรับผู้ใช้งานสูง สามารถใช้งานได้บนแพลตฟอร์ม Mac, Linux และ Window Arduino สามารถเรียนรู้สิ่งใหม่ได้

ข้อดีของ Arduino:

คำใช้จำเพาะ:บอร์ด Arduino มีราคาถูกกว่าเมื่อเทียบกับไมโครอื่น ๆ
บอร์ดควบคุม

แพลตฟอร์ม:ซอฟต์แวร์ Arduino (IDE) เข้ากันได้กับระบบปฏิบัติการ เช่น Macintosh OSX, Windows และ Linux

เป็นมิตรกับผู้ใช้:ซอฟต์แวร์ Arduino (IDE) ใช้งานง่ายและง่ายต่อการเริ่มต้นและมีความยืดหยุ่นสำหรับโปรแกรมเมอร์ที่มีทักษะ

ไอเฟนบอร์ด:Arduino เป็นซอฟต์แวร์ไอเฟนบอร์ดที่สามารถ
โปรแกรมด้วยภาษา C, C++ หรือ AVR-C ผู้ใช้จึงสามารถออกแบบโมดูลได้
หลากหลาย

4.1.1 Arduino Uno

Arduino/Genuino Uno มีไมโครคอนโทรลเลอร์ ATmega328 ออนบอร์ด มีพอร์ตอินพุตแบบอะนาล็อก 6 พอร์ต (A0–A5) และพอร์ต I/O ดิจิตอล 14 พอร์ต โดย 6 พอร์ตเป็นพิน PWM แต่ละพินสามารถทำงานด้วยแรงดันไฟฟ้า 0–5 V ทำงานที่ความถี่ 16 MHz **รูปที่ 4.1** แสดงบอร์ด Arduino Uno ([ตาราง 4.1](#)).

ตาราง 4.4

ตารางเปรียบเทียบสำหรับบอร์ด Arduino บางตัว

ชื่อ	โปรเซสเซอร์	ซีพียู ความเร็ว	ปฏิบัติการ/ ป้อนข้อมูล แรงดันไฟฟ้า	ดิจิทัลไอโอ/ PWM	อนาล็อก เข้า/ออก	UART	แฟลช [kB]
LilyPad	ATmega168V ATmega328P	8 MHz	2.4–5.5 โวลต์/ 2.4–5.5 V	14/6	6/0	—	16
เมก้า 2560	ATmega2560	16 MHz	5 โวลต์/4–12 โวลต์	54/15	16/0	4	256
ไมโคร	ATmega32U4	16 MHz	5 โวลต์/4–12 โวลต์	20/4	12/0	1	32
จูโนะ	ATmega328P	16 MHz	5 โวลต์/4–12 โวลต์	14/6	6/0	1	32
เลโอนาร์โด	ATmega32U4	16 MHz	5 โวลต์/4–12 โวลต์	20/4	12/0	1	32
ยูน	ATmega32U4 AR9331 Linux	16 MHz 400 MHz	5 V	20/4	12/0	1	32
อีเธอร์เน็ต	ATmega328P	16 MHz	5 โวลต์/4–12 โวลต์	14/4	6/0	—	32
เจมปา	ATtiny85	8 MHz	3.3 โวลต์/ 4–16 V	3/2	1/0	—	8
MKRZero	SAMD21 Cortex-M0+ ต่ำ 32 บิต พลัง ARM MCU	48 MHz	3.3 V	22/12	4 (ADC 8/10/ 12 บิต)/1 (DAC 10 บิต)	1	256

4.2 Arduino IDE

Arduino integrated development environment (IDE) เป็นซอฟต์แวร์โอเพ่นซอร์ส และทำให้ง่ายต่อการเขียนโค้ดและอัปโหลดไปยังบอร์ด

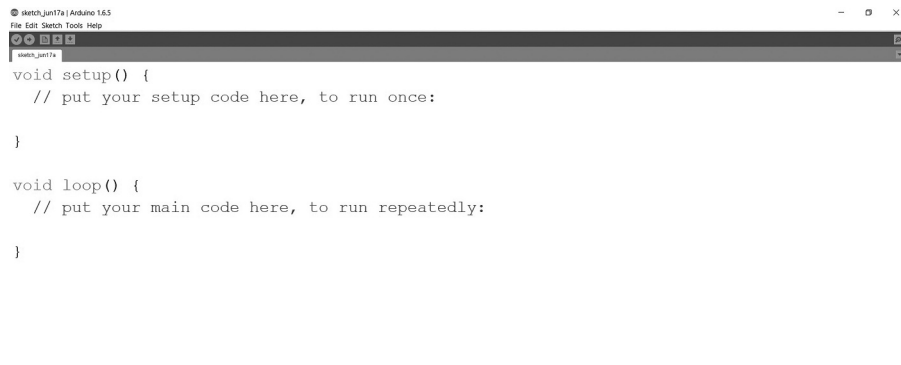
4.2.1 ขั้นตอนในการติดตั้ง Arduino IDE

ขั้นตอนที่ 1: ติดตั้ง Arduino IDE และเปิดหน้าต่าง

ในการเริ่มต้น ให้ติดตั้ง Arduino IDEที่4.4แสดงหน้าต่างของ Arduino IDE

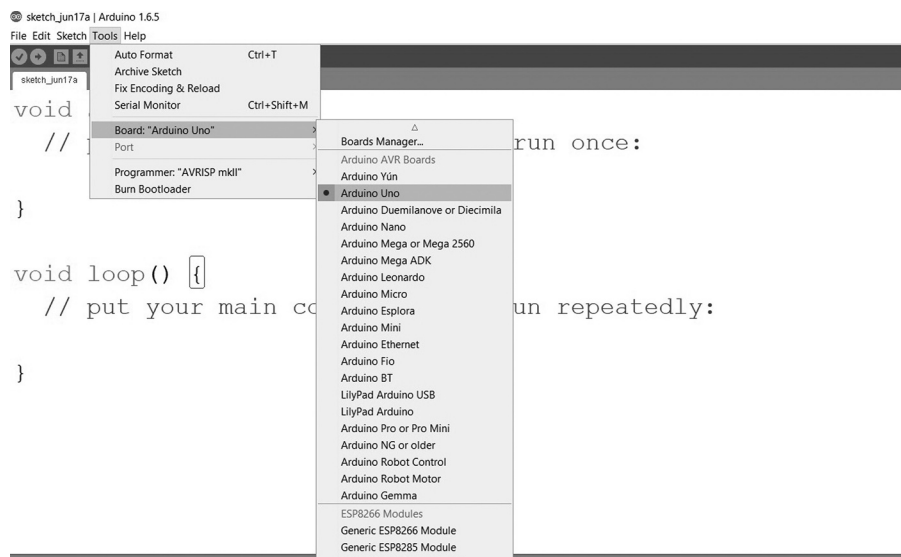
ขั้นตอนที่ 2: เลือกเวอร์ชันของบอร์ด Arduino

Arduino มีหลายเวอร์ชัน เช่น UNO, MEGA, NANO เป็นต้น ก่อนเริ่ม-
ในโครงการค้นหารุ่นที่เหมาะสมโดยเลือกพารามิเตอร์ตามความต้องการ บอร์ดทั่วไป
สำหรับผู้เริ่มต้นคือ Arduino UNO เลือกบอร์ดและพอร์ตอนุกรม



şu 4.4

หน้าต่าง Arduino IDE



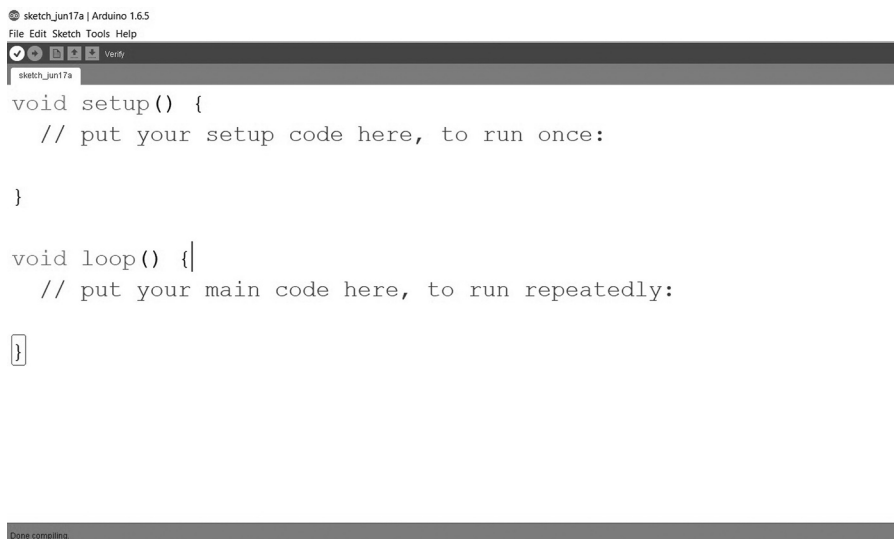
รูปที่ 4.5

การเลือกบอร์ด Arduino

ใน Arduino IDE ในการเลือกบอร์ด Arduino ให้คลิกที่ "เครื่องมือ" จากนั้นคลิกที่ "บอร์ด"**รูปที่4.5**แสดงการเลือก "Arduino Uno"

ขั้นตอนที่ 3: เขียนและคอมไพล์โปรแกรม

เขียนโปรแกรมในหน้าต่าง Arduino IDE จากนั้น "เรียกใช้" โปรแกรม. [รูปที่ 4.6](#) แสดงหน้าต่างเพื่อคอมไพล์โปรแกรม



§4.6

รวบรวมโปรแกรม

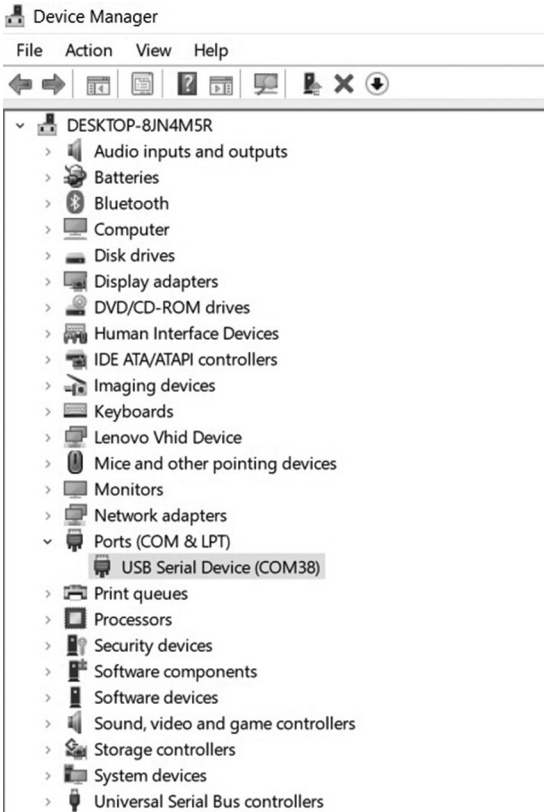
ขั้นตอนที่ 4: เชื่อมต่อ Arduino กับ PC

เชื่อมต่อ Arduino กับพอร์ต USB ของพีซีด้วยสาย USB ทั้งหมด
บอร์ด Arduino มีที่อยู่พอร์ตอนุกรมที่แตกต่างกัน (COM2, COM4 ฯลฯ) ดังนั้น
จึงจำเป็นต้องกำหนดค่าพอร์ตใหม่สำหรับ Arduino แต่ละตัวและเลือกใน IDE ใน
การตรวจสอบพอร์ตที่เชื่อมต่อ Arduino ให้คลิกขวาที่ "PC" จากนั้นเลือก
"manager"; หน้าต่างจะเปิดขึ้น จากนั้นดับเบิลคลิกที่ "ตัวจัดการอุปกรณ์"
หน้าต่างตามที่แสดงในรูปที่ 4.7 จะเปิด คลิกที่พอร์ต COM และ LPT และพบพอร์ตที่
อุปกรณ์เชื่อมต่ออยู่

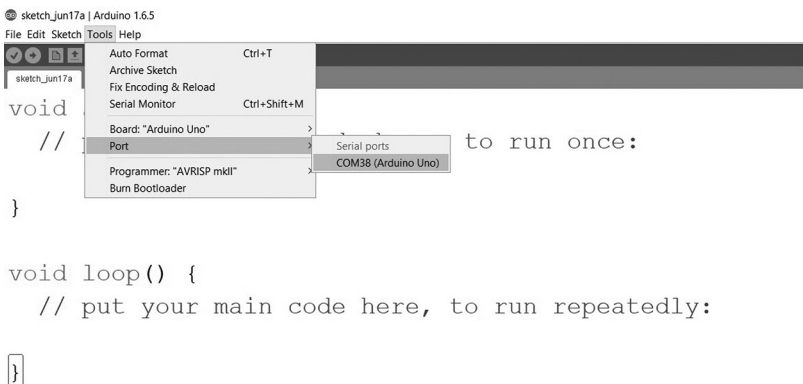
ตอนฝึกที่หัว "เครื่องมือ" ที่หน้าต่าง Arduino IDE ไปที่
พอร์ตและเลือกหมายเลขพอร์ตเดียวกัน ซึ่งพบได้ในโปรแกรมจัดการอุปกรณ์ (เลือก
COM1 หรือ COM2 เป็นต้น)รูปที่ 4.8แสดง “COM38” เป็นพอร์ตอนุกรมของ
บอร์ด

ขั้นตอนที่ 5: อัปโหลดโปรแกรมไปยังบอร์ด Arduino

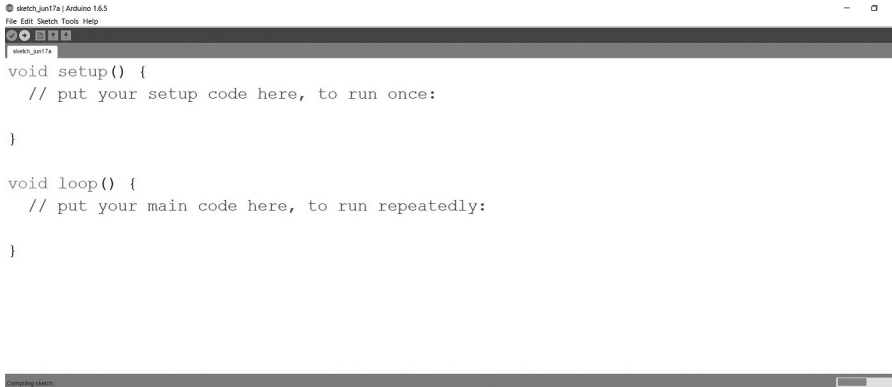
อัปโหลดโปรแกรมไปยังบอร์ด Arduino **รูปที่ 4.9** แสดงวิธีการ
ลงโปรแกรม.



รูปที่ 4.7
หน้าต่างตรวจสอบพอร์ต Arduino



รูป 4.8
พอร์ตอนุกรมของบอร์ด



รูป 4.9

หน้าต่างสำหรับอัปโหลดโปรแกรม

4.3 คำสั่งพื้นฐานสำหรับ Arduino

1. **pinMode(x, เอาต์พุต);**//กำหนดหมายเลขพิน x เป็นพินเอาต์พุต โดยที่ x คือจำนวนพินดิจิทัล
2. **digitalWrite(x, สูง);**//เปิดหมายเลขพิน x เป็น HIGH หรือ ON โดยที่ x คือหมายเลขพินดิจิทัล
3. **pinMode(x, อินพุต);**//กำหนดหมายเลขพิน x เป็นพินอินพุต โดยที่ x คือจำนวนพินดิจิทัล
4. **digitalRead (พินดิจิทัล);**//อ่านพินดิจิทัลเช่น 13 หรือ 12 หรือ 11 เป็นต้น
5. **analogRead (ขาอะนาล็อก);**//อ่านพินอะนาล็อกเช่น A0 หรือ A1 หรือ A2 เป็นต้น

4.4 คำสั่ง LCD

1. **lcd.begin(16, 2);**//เริ่มต้น LCD 16*2 หรือ 20*4
2. **lcd.print("ราชา");**//พิมพ์สตริง "RAJESH" บน LCD
3. **lcd.setCursor(x, y);**//ตั้งค่าเคอร์เซอร์ของ LCD ในตำแหน่งที่ต้องการ โดยที่ x คือจำนวน COLUMN และ y
4. **lcd.print(LPU);**//พิมพ์ LPU เป็นจำนวนเต็มบน LCD
5. **จอแอลซีดีที่ชัดเจน();**//ล้างเนื้อหาของ LCD

4.5 คำสั่งการสื่อสารแบบอนุกรม

- 1.**Serial.begin(รับส่งข้อมูล);**//เริ่มต้นการสื่อสารแบบอนุกรมเพื่อกำหนดอัตราบอดเป็น 600/1200/2400/4800/9600
- 2.**Serial.print("ราเชช");**//สตริงคงที่สำหรับการพิมพ์แบบอนุกรมพร้อมกำหนดอัตราบอดบน Tx line
- 3.**Serial.println("ราเชช");**//สตริงคงที่สำหรับการพิมพ์แบบอนุกรมพร้อมกำหนดอัตราบอดและป้อนคำสั่งบน Tx line
- 4.**Serial.print ("LPU");**//สตริง int พิมพ์แบบอนุกรมพร้อมกำหนดอัตราบอดบน Tx line
- 5.**Serial.print ("LPU");**//สตริง int พิมพ์แบบอนุกรมพร้อมกำหนดอัตราบอดและป้อนคำสั่งบน Tx line
- 6.**อนุกรม.เขียน (BYTE);**//serial โอนหนึ่งไบต์บน Tx line
- 7.**Serial.read();**//อ่านหนึ่งไบต์ซีเรียลจาก Rx line

4.6 เล่นกับ LED และ Arduino

ไดโอดเปล่งแสง (LED) เป็นอุปกรณ์ที่สามารถใช้เป็นตัวบ่งชี้ได้ LED มีขั้ว 2 ขั้ว คือ แอโนดและแคโทด 1W LED มีให้เลือกหลายสี [รูปที่ 4.10](#)แสดง LED



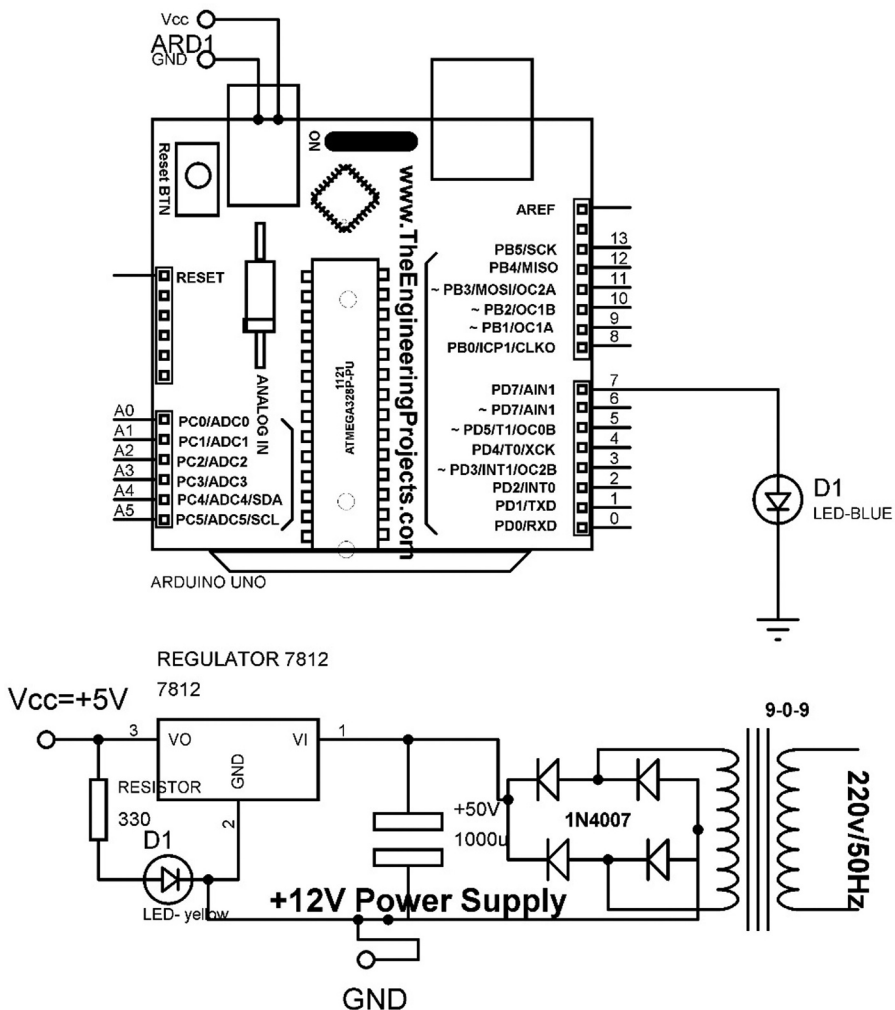
รูป 4.10

ไดโอดเปล่งแสง

สามารถใช้สีต่างๆ เพื่อแสดงเงื่อนไขต่างๆ ได้ สีของ LED เกิดจากการเปล่งแสงในบริเวณเฉพาะของสเปกตรัมแสงที่มองเห็นได้จากสารประกอบต่างๆ

เพื่อให้เข้าใจการทำงานของ LED ให้เชื่อมต่อขั้วบวกของ LED กับขา 4 ของ Arduino และขั้วลบกับกราวด์ อับโหลดภาพร่างที่อธิบายไว้ในหัวข้อ 4.4.1 ไปยัง Arduino และสังเกตการกะพริบของ LED

รูปที่ 4.11 แสดงแผนภาพวงจรของ Arduino ที่เชื่อมต่อกับ LED



รูป 4.11

แผนภาพวงจรเพื่อเชื่อมต่อ LED กับ Arduino

4.6.1 ร่าง

```
int LED_CONTROL=4;

การตั้งค่าเป็นโมฆะ ()

{
    โหมดพืน (LED_CONTROL, OUTPUT); // เริ่มต้ณพืน 4 เป็นพืนเอาต์พุต
}

วงเป็นโมฆะ ()

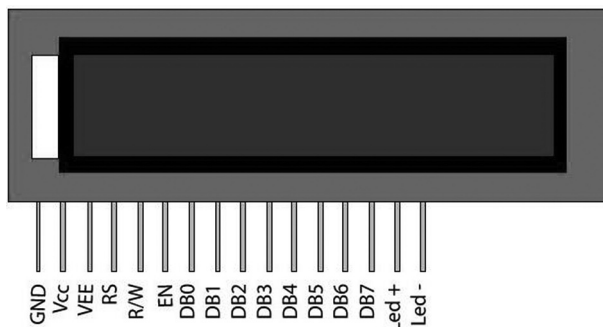
{
    digitalWrite (LED_CONTROL, สูง); // ทำให้พืน 4 ตีเล่ย์สูง
    (1000); // 1,000 มิลลิวินาทีล่าช้า
    digitalWrite (LED_CONTROL, ต่ำ); // ทำให้พืน 4 ตีเล่ย์สูง
    (1000); // 1,000 มิลลิวินาทีล่าช้า
}
```

4.7 เล่นกับ LCD ด้วย Arduino

จอแสดงผลคริสตัลเหลว (LCD) เป็นโมดูลแสดงผลที่ใช้กันทั่วไป A 16×2 จอ LCD ใช้เป็นอุปกรณ์แสดงผลในวงจร โมดูลนี้เป็นการมากกว่าเจ็ดส่วน เนื่องจากไม่มีข้อจำกัดในการแสดงอักขระพิเศษ หรือแม้แต่กำหนดเอง และประหยัด

A 16×2 LCD สามารถแสดงอักขระได้ 16 ตัวต่อแถว และมี 2 แถว ใน LCD นี้ 5×เมทริกซ์ 4 พิกเซลแสดงอักขระ มีการลงทะเบียณสองรายการคือการลงทะเบียณข้อมูลและการลงทะเบียณคำสั่ง [รูปที่ 4.12](#)แสดง 16×2 จอแอลซีดี

A 20×4 LCD มี 4 แถวและสามารถแสดงได้ 20 ตัวอักษรต่อแถว A 5×ใช้เมทริกซ์ 4 พิกเซลเพื่อแสดงอักขระ คำอธิบายพืนเหมือนกับ LCD (16×2). [รูปที่ 4.13](#)แสดง 20×4 จอแอลซีดี ([ตาราง4.5](#)).



รูป 4.12

จอแสดงผลคริสตัลเหลว (16×2).



รูป 4.13
จอแสดงผลคริสตัลเหลว (20×4).

ตาราง 4.5
LCD Pin Description

เขียนชุด	คำอธิบาย
พิน (1) กราวด์	กราวด์ (0 V)
พิน (2) Vcc	แหล่งจ่ายไฟ (5 V)
พิน (3) VEE	ตัวต้านทานปรับค่าได้ใช้เพื่อปรับความคมชัด
ปักหมุด (4) ลงกะเบียนเลือก	เมื่อต่ำก็จะเลือกคำสั่งรีจิสเตอร์และถ้าสูงก็จะเลือกการลงกะเบียนข้อมูล
ปักหมุด (5) อ่าน/เขียน	สูงเพื่ออ่านรีจิสเตอร์และต่ำเพื่อเขียนบนรีจิสเตอร์ ส่งข้อมูลไปยังสายข้อมูลเมื่อได้รับพัลส์สูงไปต่ำ
ปักหมุด (6) เปิดใช้งาน	
พิน (7) DB0	
พิน (8) DB1	
พิน (9) DB2	สายข้อมูล 8 บิต
พิน (10) DB3	
พิน (11) DB4	
พิน (12) DB5	
พิน (13) DB6	
พิน (14) DB7	
พิน (15) LED+	แบ็คไลท์ Vcc (5 V) แบ็คไลท์
ขา (16) LED-	กราวด์ (0 V)

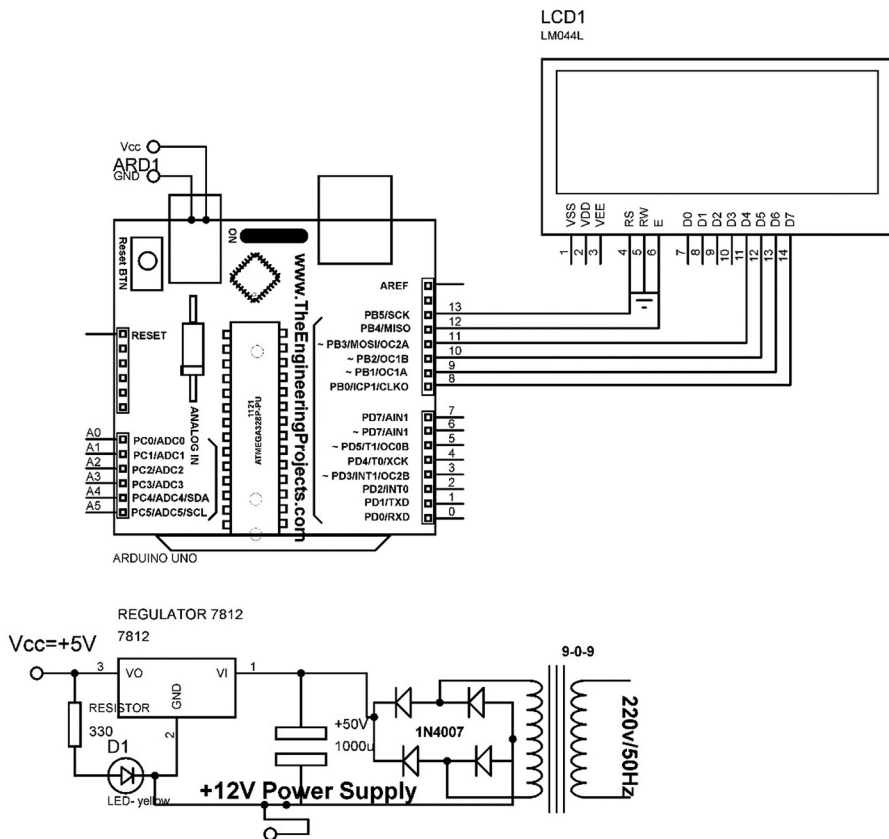
การเชื่อมต่อ LCD

เชื่อมต่อส่วนประกอบดังต่อไปนี้:

- ขาดิจิตอล Arduino (13) ถึง ขา RS (4) ของ LCD
- Arduino ดิจิตอลพิน (GND) ถึง RW พิน (5) ของ LCD
- ขาดิจิตอล Arduino (12) ถึง E ขา (6) ของ LCD

- Arduino ดิจิตอลพิน (11) ถึง D4 พิน (11) ของ LCD
- ขาดิจิตอล Arduino (10) ถึง D5 ขา (12) ของ LCD
- ขาดิจิตอล Arduino (9) ถึง D6 ขา (13) ของ LCD
- ขาดิจิตอล Arduino (8) ถึงขา D7 (14) ของ LCD

รูปที่ 4.14แสดงแผนภาพวงจรของ Arduino ที่เชื่อมต่อกับ LCD



ŞÜ 4.14

แผนภาพวงจรสำหรับอ่าน LCD

4.7.1 ร่าง

```
# รวม <LiquidCrystal.h> LiquidCrystal
LCD (13, 12, 11, 10, 9, 8);

การตั้งค่าเป็นโมดูล ()
{
  lcd.begin(20, 4); // เริ่มต้น LCD lcd.print ("ยินดีต้อนรับ"); // พิมพ์สตริงบน LCD ลำช้า (2000); // หน่วยเวลา 2000mS
  lcd.clear();
}

วงเป็นโมดูล ()
{
  lcd.setCursor(0, 1); // ตั้งค่าเคอร์เซอร์ของ LCD
  lcd.print("ECE Department"); // พิมพ์สตริงบน LCD ลำช้า (2000); // หน่วยเวลา 2000mS
  lcd.setCursor(0, 2); // ตั้งค่าเคอร์เซอร์ของ LCD
  lcd.print("Rajesh Singh"); // พิมพ์สตริงบน LCD ลำช้า (2000); // หน่วยเวลา 2000mS
}
```