



## โครงการ

เรื่อง การออกแบบเครื่องคิดเลข

เสนอ

ผศ.ดร.สิรภพ ตู๊ประกาย

Asst.Prof.Dr. Siraphop Tooprakai

และ

ดร.สมปอง วิเศษพานิชกิจ

Dr. Sompong Wisetpanichkij

จัดทำโดย

นายวงศธร มาตรฐาน 63010835

วิชา Embedded System And Application Laboratory 01016261

ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

# บทที่ 1

## บทนำ

### ที่มาและความสำคัญ

ปัจจุบันความก้าวหน้าทางเทคโนโลยีสารสนเทศก่อให้เกิดการเปลี่ยนแปลงครั้งยิ่งใหญ่ต่อทุกวงการทั่วโลก รวมทั้งวงการศึกษาไทยด้วยและผลพวงที่ติดตามมาในแง่เทคนิควิธีการเกี่ยวกับกระบวนการเรียนรู้ในการพัฒนาเทคโนโลยีสารสนเทศเพื่อให้เกิดประโยชน์สูงสุดแก่ผู้เรียนในทุกระดับ มีการใช้เครื่องคอมพิวเตอร์และเครื่องมืออิเล็กทรอนิกส์มากขึ้นในการศึกษา

จะเห็นได้ว่าการจัดการศึกษาในยุคของการใช้เทคโนโลยีสารสนเทศสามารถเปลี่ยนแปลงกระบวนการเรียนรู้และความต้องการของการศึกษาในอนาคต สื่อและอุปกรณ์การศึกษารูปแบบใหม่จะเข้ามาแทนที่สื่อแบบเก่า มีแหล่งทรัพยากรการเรียนรู้ที่หลากหลายนับเป็นสิ่งที่ช่วยสนับสนุนการเรียนรู้ในสภาพแวดล้อมทางการศึกษาแบบใหม่ ทำให้การเรียนการสอนไม่จำกัดอยู่เฉพาะในห้องเรียนและอยู่ภายใต้การควบคุมของผู้สอนเท่านั้น แต่ผู้เรียนสามารถเรียนรู้ได้จากแหล่งความรู้ที่หลากหลาย โดยเฉพาะเครือข่ายอินเทอร์เน็ต ในการศึกษาหาข้อมูลในรายวิชา EMBEDDED SYSTEMS AND APPLICATIONทำให้เป็นที่มาของการทำโครงการเรื่องเครื่องคิดเลขนี้

### วัตถุประสงค์

- 1.เพื่อศึกษาค้นคว้าด้วยตนเอง
- 2.เพื่อฝึกฝนการเขียน code ในการใช้ Arduino
- 3.เพื่อนำเครื่องคิดเลข จาก Arduino uno

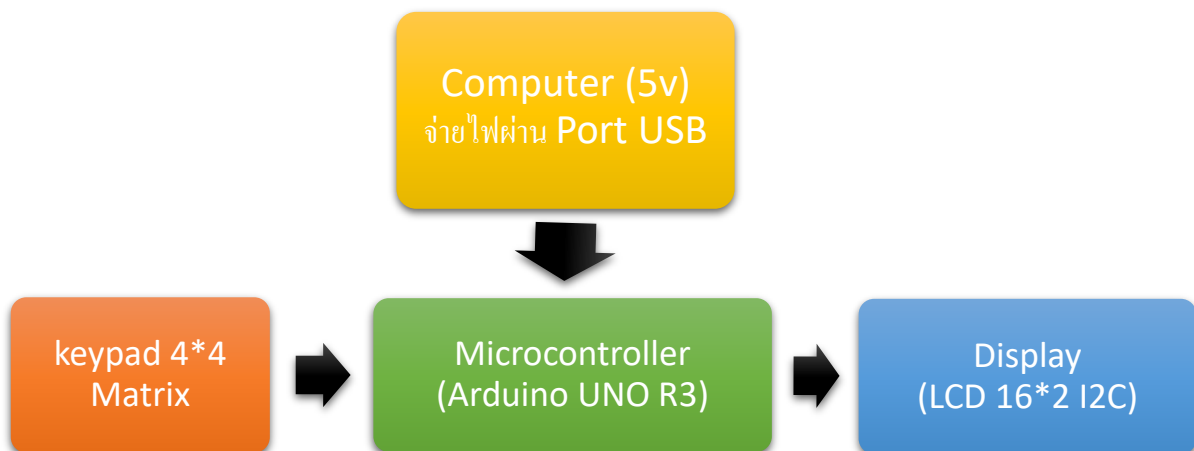
### ขอบเขตการศึกษาค้นคว้า

- 1.ศึกษาค้นคว้าตัว code จากอินเทอร์เน็ต
- 2.ศึกษาการทำเครื่องคิดเลขจาก YouTube
- 3.ระยะเวลาในการทำ 2 สัปดาห์

### ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ความรู้เบื้องต้นเกี่ยวกับการใช้จอ LCD 16\*2 แบบ I2C
2. ได้ความรู้เบื้องต้นเกี่ยวกับการใช้ Arduino UNO R3
3. ได้รู้หลักการในการทำเครื่องคิดเลขเบื้องต้น
4. ได้รู้วิธีใช้ Keypad 4\*4 matrix แบบ I2C

### Block diagram



## บทที่ 2

### การทบทวนเอกสาร/วรรณกรรมที่เกี่ยวข้อง

รายงานการจัดทำโครงงานนี้ทำเพื่อศึกษาการทำเครื่องคิดเลขโดยใช้ Arduino UNO R3 ดังนั้นผู้ศึกษาจึงได้รวบรวมหาแนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้องซึ่งมีรายละเอียดดังนี้

#### 2.1 Arduino

##### 2.1.1 Arduino (Hardware)

##### 2.1.2 Arduino IDE (Software)

##### 2.1.3 โครงสร้างโปรแกรมของ Arduino ที่เกี่ยวข้อง

##### 2.1.4 Arduino Uno

#### 2.2 การใช้งาน Keypad 4\*4

#### 2.3 การใช้งานจอ LCD 16\*2 I2C ที่เกี่ยวข้อง

### 2.1 Arduino

Arduino อ่านว่า (อาดูโน้ หรือ อา-ดู-อิ-โน้) เป็นแพลตฟอร์มอิเล็กทรอนิกส์แบบโอเพนซอร์สบนพื้นฐานของฮาร์ดแวร์และซอฟต์แวร์ที่ง่ายต่อการใช้งาน สามารถนำไปประยุกต์ใช้งานได้หลายอย่าง เช่น ควบคุมอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก อ่านค่าเซ็นเซอร์วัดสภาพแวดล้อมต่างๆแล้วแสดงค่าที่เซ็นเซอร์สามารถอ่านได้ออกมาทางจอแสดงผล นำไปประยุกต์ใช้เป็นชิ้นงานทางอิเล็กทรอนิกส์เพื่ออำนวยความสะดวกในการใช้ชีวิตประจำวัน เป็นต้น

ปัจจุบัน Arduino ถือได้ว่าเป็นแพลตฟอร์มที่ได้รับความนิยมสูงจากทั่วโลก เนื่องมาจากราคาของตัวบอร์ด Arduino ไม่ค่อยสูงมาก เป็นโอเพนซอร์สทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ คอมมิวนิตีและฟอรัมน์ในการถามตอบเรื่องเกี่ยวกับการใช้งานร่วมกับอุปกรณ์อิเล็กทรอนิกส์ต่างๆ ตัวอย่างโปรแกรมเบื้องต้นและไลบรารีสำหรับใช้งาน และอื่นๆ ในแพลตฟอร์มของ Arduino ประกอบไปด้วยส่วนหลักๆ 2 ส่วน คือ

### 2.1.1 Arduino (Hardware)

บอร์ด Arduino เป็นวงจรอิเล็กทรอนิกส์ขนาดเล็กที่มีไอซีหรือชิปประมวลผลของไมโครคอนโทรลเลอร์ (Micro Controller Unit:MCU) ประกอบอยู่กับวงจรอิเล็กทรอนิกส์อื่นๆ ตัวบอร์ดของ Arduino จะมีอยู่หลายโมเดลให้เลือกใช้ โดยบอร์ดแต่ละโมเดลจะมีความแตกต่างกันในเรื่องของสเป็ค จำนวนพอร์ตต่างๆสำหรับใช้งาน พื้นที่ในการเขียนโปรแกรม ขนาดของบอร์ด เป็นต้น (เปรียบเทียบบอร์ด Arduino แต่ละโมเดลได้จาก



### 2.1.2 Arduino IDE (Software)

Arduino IDE เป็นโปรแกรมที่ “แจกฟรี” ในการใช้งานลักษณะ Open source ซึ่ง Arduino IDE จะทำหน้าที่ติดต่อระหว่างคอมพิวเตอร์ ไม่ว่าจะเป็นระบบ Windows, Mac OS X หรือ Linux กับ บอร์ด Arduino ซึ่งโปรแกรมนี้ออกแบบให้ง่ายต่อการเขียนโค้ดและอัปโหลดโปรแกรมที่เราเขียนเข้าสู่บอร์ด Arduino

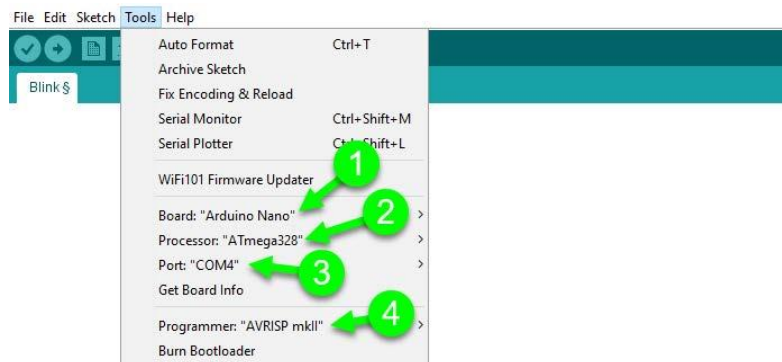
Arduino IDE ส่วน IDE ย่อมาจาก (Integrated Development Environment) คือ ส่วนเสริมของระบบการพัฒนา หรือตัวช่วยต่าง ๆ ที่จะคอยช่วยเหลือ Developer หรือช่วยเหลือคนที่พัฒนา Application เพื่อเสริมให้เกิดความรวดเร็ว ถูกต้อง แม่นยำ ตรวจสอบระบบที่จัดทำได้ ทำให้การพัฒนา งานต่าง ๆ เร็วมากขึ้น

ส่วนในการเขียนโปรแกรมและคอมไพล์ลงบอร์ด โดยขนาดของโปรแกรม Arduino โดยปกติแล้วจะใหญ่กว่าโค้ด AVR ปกติเนื่องจากโค้ด AVR เป็นการเข้าถึงจากรีจิสเตอร์โดยตรง แต่โค้ด Arduino เข้าถึง

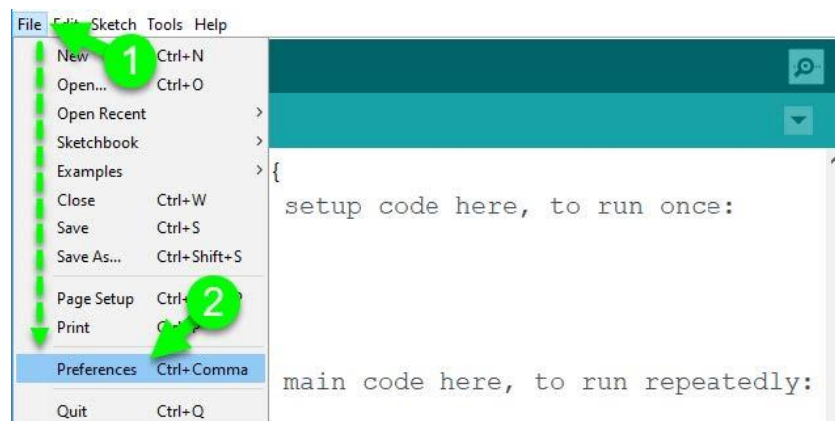
ผ่านฟังก์ชัน เพื่อให้สามารถเขียนโค้ดได้ง่ายมากกว่าการเขียนโค้ดแบบ AVR หรือเวอร์ชันอื่นๆ ของ Arduino

### การใช้งานโปรแกรม Arduino IDE

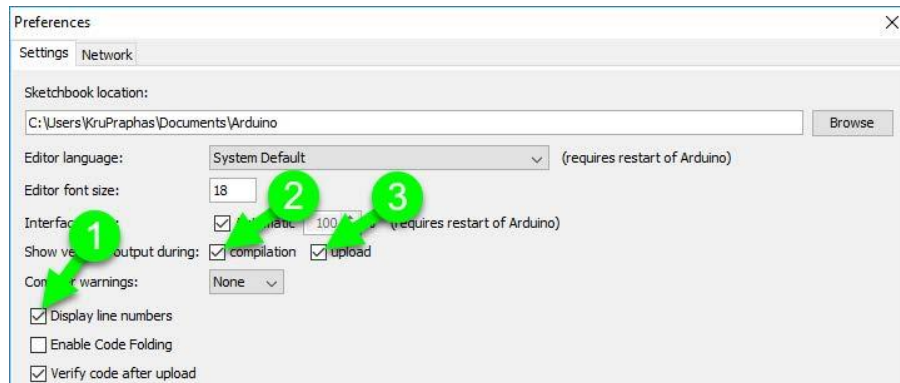
1. เรียกใช้โปรแกรมด้วยการดับเบิลคลิกที่ไอคอน Arduino.exe ในตำแหน่งที่ติดตั้งโปรแกรมไว้
2. ตั้งค่าบอร์ดให้ตรงกับที่ใช้งานซึ่งมีการตั้งค่าอยู่ด้วยกัน 4 รายการดังนี้
  - 1) ชนิดของบอร์ด Arduino ที่ใช้งาน (กำลังเชื่อมต่ออยู่)
  - 2) ชนิดของProcessor เลือกให้ตรงกับ Processor ของบอร์ด (บอร์ดบางชนิดไม่ต้องเลือก เนื่องจากมีใช้งานเพียงเบอร์เดียว)
  - 3) คอมพอร์ตที่กำลังเชื่อมต่อ เลือกให้ตรงกับที่บอร์ด Arduino กำลังเชื่อมต่อ (ดูจาก Device manager)
  - 4) ชนิดของเครื่องโปรแกรม เลือก AVRISP mkII



3. ตั้งค่าในโปรแกรม Arduino IDE ให้แสดงตำแหน่งของไฟล์ภาษาเครื่อง (HEX file) หลังจากการคอมไพล์ผ่าน การตั้งค่าดังรูป

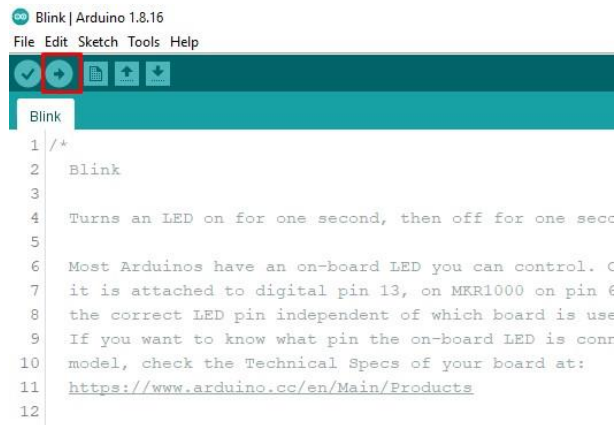


#### 4. คลิกเลือกแสดงผลทั้ง 3 รายการดังรูป



#### 5. เขียนโปรแกรมหรือเรียกใช้งานโปรแกรมตัวอย่าง

#### 6. คลิกปุ่ม Verify ตรวจสอบความถูกต้องของโค้ดที่เขียนและคลิกปุ่ม Upload ลงบอร์ด (กรณีต่อบอร์ดจริง)



### 2.1.3 โครงสร้างโปรแกรมของ Arduino ที่เกี่ยวข้อง

ในการเขียนโปรแกรมสำหรับบอร์ด Arduino จะต้องเขียนโปรแกรมโดยใช้ภาษาของ Arduino (Arduino Programming Language) ซึ่งตัวภาษาของ Arduino ก็นำเอาโอเพ่นซอร์สโปรเจกต์ชื่อ Wiring มาพัฒนาต่อภาษาของ Arduino แบ่งได้เป็น 2 ส่วนหลักคือ

1. โครงสร้างภาษา (Structure) ตัวแปรและค่าคงที่
2. ฟังก์ชัน (Function)

ภาษาของ Arduino จะอ้างอิงตามภาษา C/C++ จึงอาจกล่าวได้ว่าการเขียนโปรแกรมสำหรับ Arduino (ซึ่งก็รวมถึงบอร์ด Arduino) ก็คือการเขียนโปรแกรมภาษา C โดยเรียกใช้ฟังก์ชันและไลบรารีที่ทาง Arduino ได้เตรียมไว้ให้แล้ว ซึ่งสะดวกและทำให้ผู้ที่ไม่มีความรู้ด้านไมโครคอนโทรลเลอร์อย่างลึกซึ้งสามารถเขียนโปรแกรม ทำงานได้ในบทรนี้จะอธิบายถึงโครงสร้างโปรแกรมของ Arduino แบ่งได้เป็นสองส่วนคือ void setup() และ void loop()

#### ส่วนของฟังก์ชัน setup()

ฟังก์ชันนี้จะเขียนที่ส่วนต้นของโปรแกรมทำงานเมื่อโปรแกรมเริ่มต้นเพียงครั้งเดียวใช้เพื่อกำหนดค่าของตัวแปร โหมดการทำงานของขาต่างๆ เริ่มต้นเรียกใช้ไลบรารี ฯลฯ

#### ส่วนของฟังก์ชัน loop()

หลังจากที่เขียนฟังก์ชัน setup() ที่กำหนดค่าเริ่มต้นของโปรแกรมแล้ว ส่วนถัดมาคือฟังก์ชัน loop() ซึ่งมีการทำงานตรงตามชื่อ คือจะทำงานตามฟังก์ชันวนต่อเนื่องตลอดเวลา ภายในฟังก์ชันจะมีโปรแกรมของผู้ใช้เพื่อรับค่าจากพอร์ต ประมวลผลแล้วส่งเอาต์พุตออกขาต่างๆ เพื่อควบคุมการทำงานของบอร์ด ในฟังก์ชันนี้จะยกตัวอย่างคำสั่งที่ใช้มา



## คำสั่งและเงื่อนไขต่างๆ

### 1. คำสั่ง if

ใช้ทดสอบเพื่อกำหนดเงื่อนไขการทำงานของโปรแกรม เช่นถ้าอินพุตมีค่ามากกว่าค่าที่กำหนดไว้จะให้ทำอะไร โดยมีรูปแบบการเขียนดังนี้

```
if (somevariable > 50)
```

```
{ // do something Here }
```

2. คำสั่ง if...else ใช้ทดสอบเพื่อกำหนดเงื่อนไขการทำงานของโปรแกรมได้มากกว่าคำสั่ง if ธรรมดา โดยสามารถกำหนดได้ ว่าถ้าเงื่อนไขเป็นจริงให้ทำอะไร ถ้าเป็นเท็จให้ทำอะไร เช่นถ้าค่าอินพุตแอนะล็อกที่อ่านได้น้อยกว่า 500 ให้ทำอะไร ถ้าค่ามากกว่าหรือเท่ากับ 500 ให้ทำอีกอย่าง จะเขียนคำสั่งได้ดังนี้

```
if (pinFiveInput < 500)
```

```
{ // do thing A }
```

Else

```
{ // do thing B }
```

### 3. ส่วนของตัวกระทำทางคณิตศาสตร์

ประกอบด้วยตัวกระทำ 4 ตัวคือ + (บวก), - (ลบ), \* (คูณ), / (หาร)

#### รูปแบบคำสั่ง

```
result = value1 + value2;
```

```
result = value1 - value2;
```

```
result = value1 * value2;
```

```
result = value1 / value2;
```

## พารามิเตอร์

value1 : เป็นค่าของตัวแปรหรือค่าคงที่ใดๆ

value2: เป็นค่าของตัวแปรหรือค่าคงที่ ใดๆ

## 4. ส่วนของตัวกระทำเปรียบเทียบ

ใช้ประกอบกับคำสั่ง if() และ while() เพื่อทดสอบเงื่อนไขหรือเปรียบเทียบค่าตัวแปรต่างๆ โดยจะเขียนเป็นนิพจน์ อยู่ภายในเครื่องหมาย ()

$x == y$  (x เท่ากับ y)

$x != y$  (x ไม่เท่ากับ y)

$x < y$  (x น้อยกว่า y)

$x > y$  (x มากกว่า y)

$x <= y$  (x น้อยกว่าหรือเท่ากับ y)

$x >= y$  (x มากกว่าหรือเท่ากับ y)

ใช้ในการเปรียบเทียบของคำสั่ง if() มี 3 ตัวคือ &&, || และ !

### && (ตรรกะ และ)

ให้ค่าเป็นจริงเมื่อผลการเปรียบเทียบทั้งสองข้างเป็นจริงทั้งคู่ ตัวอย่าง

```
if (x > 0 && x < 5)
```

```
{ // ... }
```

ให้ค่าเป็นจริงเมื่อ x มากกว่า 0 และน้อยกว่า 5 (มีค่า 1 ถึง 4)

### || (ตรรกะ หรือ)

ให้ค่าเป็นจริงเมื่อผลการเปรียบเทียบพบว่า มีตัวแปรใดเป็นจริงหรือเป็นจริงทั้งคู่ ตัวอย่าง

```
if (x > 0 || y > 0)
```

```
{ // ... }
```

ให้ผลเป็นจริงเมื่อ x หรือ y มีค่ามากกว่า 0

**! (ใช้กลับผลเป็นตรงกันข้าม)**

ให้ค่าเป็นจริงเมื่อผลการเปรียบเทียบเป็นเท็จ ตัวอย่าง

```
if (!x)
```

```
{ // ... }
```

ให้ผลเป็นจริงถ้า x เป็นเท็จ (เช่น ถ้า x = 0 ให้ผลเป็นจริง)

#### 2.1.4 Arduino Uno

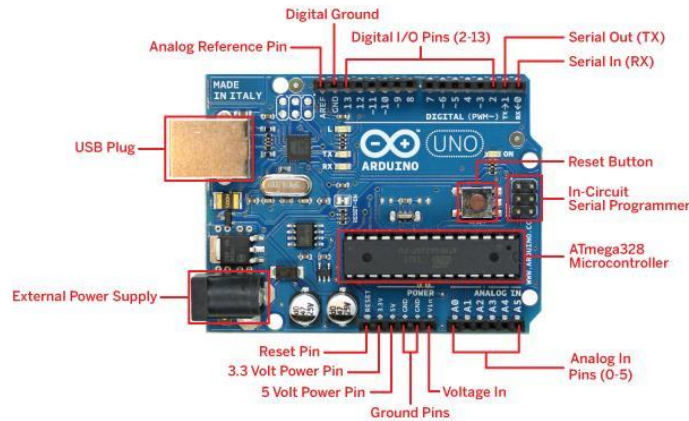


Arduino UNO เป็นบอร์ดไมโครคอนโทรลเลอร์ open-source ใช้กันอย่างแพร่หลายบนพื้นฐานของไมโครคอนโทรลเลอร์ ATmega328P บอร์ดมีชุด Pin อินพุต / เอาต์พุต (I / O) แบบดิจิทัลและอนาล็อกที่สามารถเชื่อมต่อเข้ากับบอร์ดขยาย (shields) ต่างๆ และวงจรอื่นๆ ได้

บอร์ดนี้มี Analog pin 6 ขา และ Digital pin 14 ขา โดยมี 6 ขา สามารถใช้เป็น output แบบ PWM ได้ สามารถโปรแกรมด้วย Arduino IDE (Integrated Development Environment) สามารถเชื่อมต่อกับคอมพิวเตอร์ด้วยสายเคเบิล USB หรือใช้ adaptor AC-to-DC เพื่อเริ่มต้นใช้งาน

"Uno" หมายถึง "หนึ่ง" ในภาษาอิตาลี และต้องการตั้งชื่อให้เป็นเครื่องหมายการค้าของ Arduino 1.0 โดย Uno และ version 1.0 จะเป็นรุ่นที่ใช้อ้างอิงสำหรับ Arduino รุ่นอื่นๆ ในอนาคต และ Uno ยังเป็นรุ่นล่าสุดในชุดของ USB board Arduino

นอกจากนี้ Arduino Uno ยังเป็นบอร์ดรุ่นที่นิยมมากที่สุดเนื่องจากราคาที่ไม่สูงและมีผู้พัฒนาโปรเจกต์และ library ทำให้ใช้งานได้ง่ายและมีความหลากหลาย



บอร์ดสามารถทำงานได้ในช่วงแรงดัน 6 ถึง 20 V หากแหล่งจ่ายมีค่าต่ำกว่า 7 V อาจส่งผลให้ pin ที่จ่าย 5 V ออก มีแรงดันที่ต่ำกว่า 5V และ บอร์ดอาจจะไม่เสถียร แต่ถ้าหากแรงดันมีค่าสูงกว่า 12 V อาจทำให้บอร์ดร้อนเกินไป ซึ่งทำให้บอร์ดเสียหายได้ ดังนั้นช่วงแรงดันที่เหมาะสมกับบอร์ดคือ 7 V ถึง 12 V

ฟังก์ชัน Pin ทั่วไป

LED: built-in ที่ digital pin 13 เมื่อส่งค่า High จะทำให้ LED ติด เมื่อส่งค่า Low จะทำให้ LED ดับ

VIN: ใช้เมื่อจ่ายแรงดันไฟฟ้าภายนอกเข้ากับบอร์ด

5V: เป็น output pin โดยจ่ายแรงดันออกจากบอร์ดขนาด 5V

3V3: เป็น output pin โดยจ่ายแรงดันออกจากบอร์ดขนาด 3.3V และให้กระแสสูงสุด 50 mA

GND: ขา Ground

IOREF: เป็น pin ที่ให้ voltage reference กับไมโครคอนโทรลเลอร์ เพื่อเลือกค่าแรงดันให้กับ shield ที่มาเชื่อมต่อกับบอร์ด

Reset: ใช้ในการ reset ไมโครคอนโทรลเลอร์ แต่โดยทั่วไปจะใช้โดยการเพิ่มปุ่ม reset ไว้บน shield เพื่อป้องกันปุ่มที่อยู่บนบอร์ด

## 2.2 การใช้งาน Keypad 4\*4

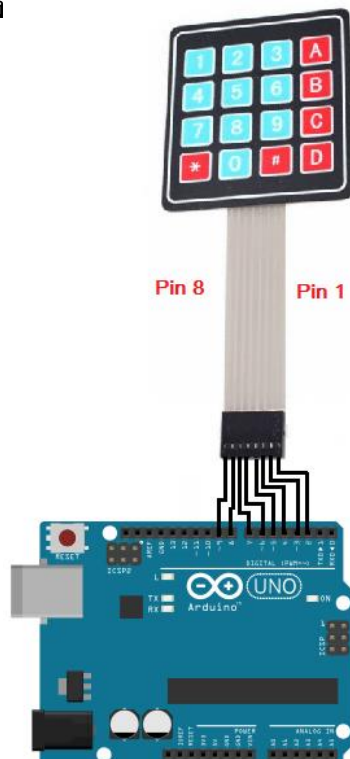
4x4 Matrix Membrane Keypad คือชุดโมดูลปุ่มกดที่สำเร็จรูป สำหรับ Input ข้อมูลหรือป้อนคำสั่งไปยังไมโครคอนโทรลเลอร์ เช่น บอร์ด Arduino

เหมาะสำหรับผู้ที่ต้องการทำงานวงจรที่มีการ Input ค่าต่างๆ เช่น วงจรจับเวลา , กรอกรหัสผ่าน และอื่นๆอีกมากมาย

### ประโยชน์ของ Matix Membrane Keypad

- ใช้งานง่ายและรวดเร็ว ไม่ต้องทำงานวงจรปุ่มเอง ทำให้ประหยัดเวลาในการทดลองวงจร หรือนำไปใช้จริงก็ได้
- ประหยัดจำนวน Pin เป็นจุดเด่นของ Matrix Keypad เพราะจำนวน Pin ที่ใช้ มักจะน้อยกว่าจำนวนปุ่มที่ได้ เช่น ในที่นี้ 4x4 ใช้ 8 Pin แต่ได้ถึง 16 ปุ่ม จะได้เหลือ Pin Arduino เอาไปทำอย่างอื่นได้อีก
- มี Library ครบครัน
- ใช้งานง่าย เนื่องจากภายในปุ่มมีพลาสติกบางๆ เรียกว่า Membrane ซึ่งจะสามารถกดและขยายตัวได้ตามแรงกด ซึ่งจะมีแผ่นโลหะเล็กๆติดอยู่ด้วย ทำหน้าที่แทนหน้าสัมผัสของ Push Button ทั่วไป แต่กดง่ายกว่ามาก

### การต่อในการทดลอง



## 2.3 การใช้งานจอ LCD 16\*2 I2C ที่เกี่ยวข้อง

### 1. ความรู้ทั่วไปเกี่ยวกับ LCD Display

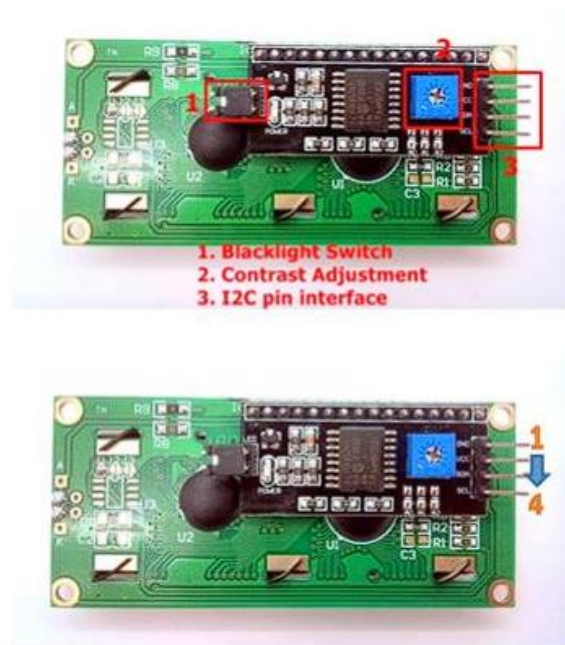
คำว่า LCD ย่อมาจากคำว่า Liquid Crystal Display ซึ่งเป็นจอที่ทำมาจากผลึกคริสตัลเหลว หลักการคือด้านหลังจอจะมีไฟส่องสว่าง หรือที่เรียกว่า Backlight อยู่เมื่อมีการปล่อยกระแสไฟฟ้าเข้าไปกระตุ้นที่ผลึกก็จะทำให้ผลึกโปร่งแสง ทำให้แสงที่มาจากไฟ Backlight แสงขึ้นมามบนหน้าจอ ส่วนอื่นที่โดนผลึกปิดกั้นไว้จะไม่สว่าง ผลึกมีสีที่แตกต่างกันตามสีของผลึกคริสตัล เช่นสีเขียว หรือ สีฟ้า ฯลฯ ทำให้เมื่อมองไปที่จอก็จะพบกับตัวหนังสือ แล้วพบกับพื้นหลังสีต่างๆกัน จอ Liquid Crystal Display (LCD) เป็นจอแสดงผลรูปแบบหนึ่งที่นิยมนำมาใช้งานกันกับระบบสมองกลฝังตัวอย่างแพร่หลาย จอ LCD มีทั้งแบบแสดงผลเป็นตัวอักษรเรียกว่า Character LCD ซึ่งมีการกำหนดตัวอักษร หรืออักขระที่สามารถแสดงผลไว้ได้อยู่แล้ว และแบบที่สามารถแสดงผลเป็นรูปภาพหรือสัญลักษณ์ได้ตามความต้องการของผู้ใช้งานเรียกว่า Graphic LCD นอกจากนี้บางชนิดเป็นจอที่มีการผลิตขึ้นมาใช้เฉพาะงาน ทำให้มีรูปแบบและรูปร่างเฉพาะเจาะจงในการแสดงผล เช่นนาฬิกา ดิจิตอล เครื่องคิดเลข หรือ หน้าปัดวิทยุ เป็นต้น จอ LCD แบ่งเป็น 2 แบบใหญ่ๆตามลักษณะการแสดงผลดังนี้

1. Character LCD เป็นจอที่แสดงผลเป็นตัวอักษรตามช่องแบบตายตัว เช่นจอ LCD ขนาด 16x2 หมายถึงใน 1 แถวมีตัวอักษรใส่ได้ 16 ตัว และมีทั้งหมด 2 บรรทัดให้ใช้งาน ส่วน 20x4 จะหมายถึงใน 1 แถวมี ตัวอักษรใส่ได้ 20 ตัว และมีทั้งหมด 2 บรรทัด

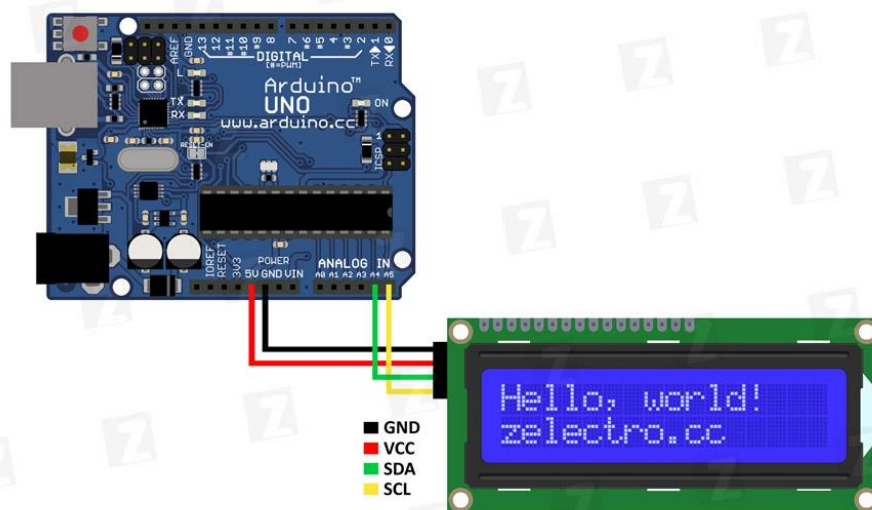
2. Graphic LCD เป็นจอที่สามารถกำหนดได้ว่าจะให้แต่ละจุดบนหน้าจอขึ้นแสงหรือปล่อยแสงออกไป ทำให้อจอนี้สามารถสร้างรูปขึ้นมบนหน้าจอได้ การระบุขนาดจะระบุในลักษณะของจำนวนจุด (Pixels) ในแต่ละแนว เช่น 128x64 หมายถึงจอที่มีจำนวนจุดตามแนวนอน 128 จุด และมีจุดตามแนวตั้ง 64 จุด หลักการทำงานของเหลวพิเศษที่มีคุณสมบัติการบิดแกนโพลาไรซ์ของแสง ถ้าจ่ายแรงดันไฟฟ้าเข้าไป ระหว่างสารเหลวนั้น โมเลกุลจะบิดตัวและทำให้แสงไม่สามารถผ่านกระจกออกมาได้ ถ้าไม่มีการจ่ายแรงดันไฟฟ้า แสงจะทะลุผ่านออกมาได้

## 2.การเชื่อมต่อสัญญาณขาข้อมูลระหว่าง Arduino กับ LCD Controller

การเชื่อมต่อ Arduino กับจอ LCD แบบอนุกรม จอ LCD ที่มีการเชื่อมแบบ I2C หรือเรียกอีกอย่างว่าการเชื่อมต่อแบบ Serial เป็นจอ LCD ธรรมดาทั่วไป ที่มาพร้อมกับบอร์ด I2C Bus ที่ทำให้การใช้งานได้สะดวกยิ่งขึ้นและมาพร้อมกับ VR สำหรับปรับความเข้มของจอใน รูปแบบ I2C ใช้ขาในการเชื่อมต่อกับ Arduino เพียง 4 ขา (แบบ Parallel ใช้ 16 ขา) ซึ่งทำให้ใช้งานได้ง่ายและ สะดวกมากยิ่งขึ้น



การต่อขาที่ใช้ในการทดลอง



## ฟังก์ชันสั่งงานจอ LCD

lcd.clear()

> ใช้ล้างหน้าจอ เมื่อมีตัวอักษรใดๆอยู่บนหน้าจอ จะถูกล้างออกทั้งหมด

lcd.home()

> ใช้ปรับให้เคเซอร์กลับไปอยู่ที่ตำแหน่งแรกด้านซ้าย เมื่อใช้คำสั่ง lcd.print() จะไปเริ่มแสดงผล ทางด้านบนซ้าย lcd.setCursor(ลำดับตัวอักษรนับจากทางซ้าย, บรรทัด)

> ใช้ตั้งค่าเคเซอร์ เช่น lcd.setCursor(2, 0); หมายถึงเคเซอร์ไปตัวอักษรที่ 2 นับจากทางซ้าย และอยู่บรรทัดแรกเมื่อใช้คำสั่ง lcd.print() ตัวอักษรตัวแรกจะอยู่ลำดับที่ 3 นับจากทางซ้าย

lcd.write(ข้อมูลที่ต้องการเขียนออกไป)

> ใช้สำหรับเขียนข้อมูลออกไปทีละตัวอักษร

lcd.print (ข้อมูลที่ต้องการให้เขียนออกไป [ รูปแบบข้อมูล])

> ใช้เขียนข้อมูลออกไปทั้งข้อความ

lcd.cursor()

> ใช้สั่งให้แสดงเคเซอร์บนหน้าจอ

lcd.noCursor()

> ใช้สั่งให้ไม่แสดงเคเซอร์บนหน้าจอ

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

### บทที่ 3

#### การออกแบบโปรแกรม

ออกแบบเครื่องคิดเลข บวก ลบ คูณ หาร โดยใช้ Arduino UNO R3 ในการทำโครงงานนี้โดยมี Keypad 4\*4 ในการป้อนข้อมูล และแสดงผลผ่านจอ LCD 16\*2 I2C



โดยกำหนดปุ่มละเงื่อนไขต่างๆ คือ

**ปุ่ม**

A เป็นการบวก

B เป็นการลบ

C เป็นการคูณ

D เป็นการหาร

# ใช้เป็นจุดทศนิยม

\* 1 ครั้ง แทนการใช้เครื่องหมาย =

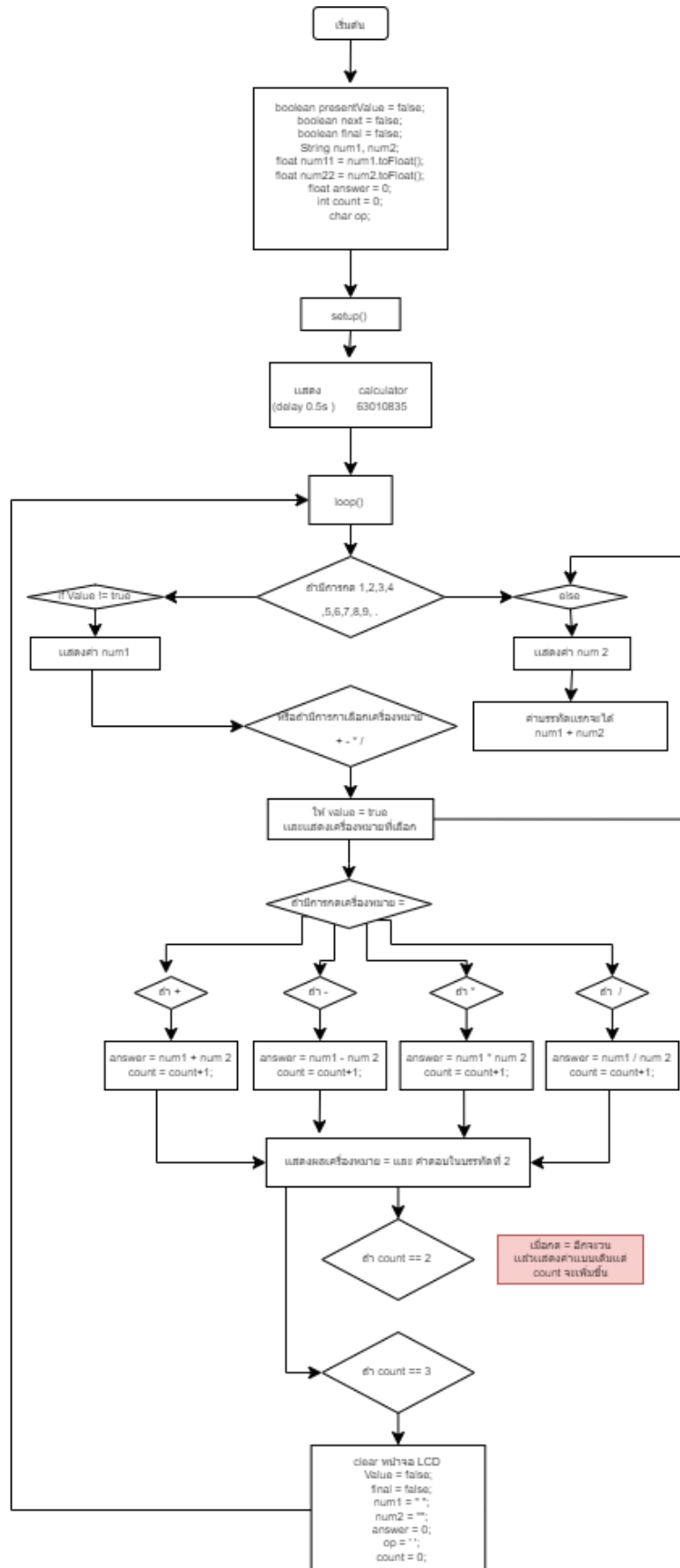
\* 2 ครั้ง หรือ \*\* แทนการ clear ค่าทุกอย่างแล้วเริ่มคำนวณใหม่

**เงื่อนไข**

1.แบบเลื่อนไปทางซ้ายเมื่อกดเพิ่ม

2.คำนวณเลข 3 หลัก และทศนิยม 2 ตำแหน่ง

## Flowchart



## Code

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 20, 2); // set the LCD address to 0x27 for a 16 chars and 2 line display

#include <Keypad.h>

const byte ROWS = 4;
const byte COLS = 4;

char keys [ROWS] [COLS] = {

  {'1', '2', '3', '+'},
  {'4', '5', '6', '-'},
  {'7', '8', '9', '*'},
  {'=', '0', '.', '/'};

byte rowPins[ROWS] = {10, 9, 7, 6};
byte colPins[COLS] = {5, 4, 3, 2};

Keypad myKeypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

boolean Value = false;
boolean next = false;
boolean final = false;

String num1, num2;

float num11 = num1.toFloat();
float num22 = num2.toFloat();

float answer = 0;

int count = 0;

char op;

void setup()
{
  lcd.init();

  lcd.backlight();

  lcd.setCursor(3,0);

  lcd.print("Calculator");

  lcd.setCursor(4,1);

  lcd.print("63010835");
```

```

    delay(500);

    lcd.clear();
}

void loop() {

    char key = myKeypad.getKey();

    if (key != NO_KEY && (key == '1' || key == '2' || key == '3' || key == '4' || key == '5' || key == '6' || key == '7' || key == '8' ||
key == '9' || key == '0' || key == '.'))
    {

        if (Value != true)

        {   num1 = num1 + key;

            int numLength1 = num1.length();

            int numLength = num1.length();

            lcd.setCursor(16 - numLength1, 0);

            lcd.print(num1);  }

        else

        {   num2 = num2 + key;

            int numLength = num2.length();

            int numLength1 = num1.length();

            lcd.setCursor(17, 0);

            lcd.print(num2);

            final = true;

            lcd.scrollDisplayLeft();  }

    }

    else if (Value == false && key != NO_KEY && (key == '/' || key == '*' || key == '-' || key == '+'))
    {

        if (Value == false)

        {   lcd.scrollDisplayLeft();

            int numLength = num1.length();

            Value = true;

```

```

    op = key;

    lcd.setCursor(16 , 0);

    lcd.print(op);  }

}

```

```

else if (final == true && key != NO_KEY && key == '=') {
if (op == '+') {

    answer = num1.toFloat() + num2.toFloat(); count = count+1;  }

else if (op == '-') {

    answer = num1.toFloat() - num2.toFloat(); count = count+1;  }

else if (op == '*') {

    answer = num1.toFloat() * num2.toFloat(); count = count+1;  }

else if (op == '/') {

    answer = num1.toFloat() / num2.toFloat(); count = count+1;  }

int numLength = num1.length();

    lcd.setCursor(1+numLength , 1);

    lcd.print("=");

    lcd.print(answer);

}

if (key != NO_KEY && count == 2){

}

if (key != NO_KEY && count == 3){

    lcd.clear();

    Value = false;  final = false;

    num1 = " "; num2 = "";

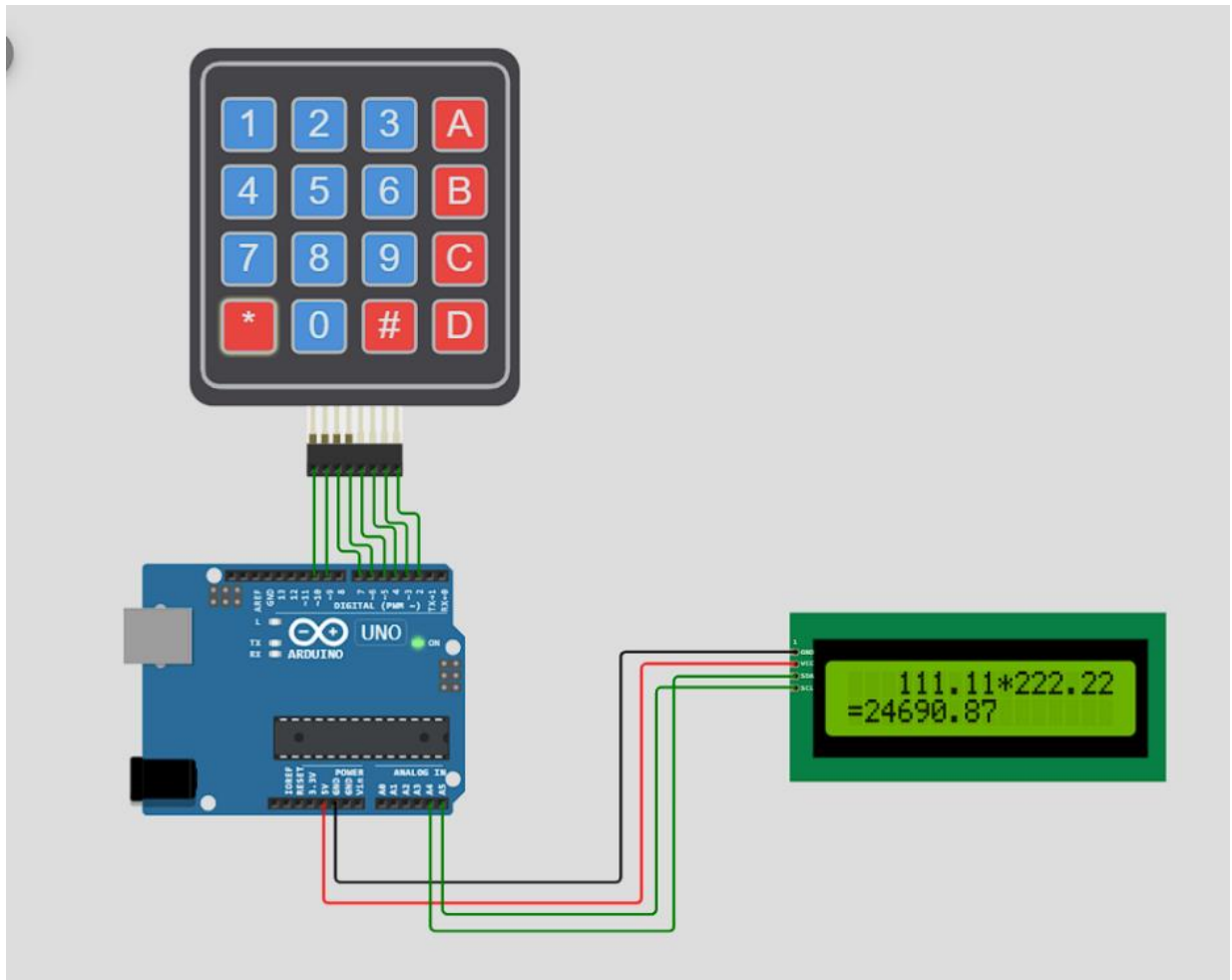
    answer = 0;

    op = ' ';

    count = 0;  }}

```

## การต่อวงจร

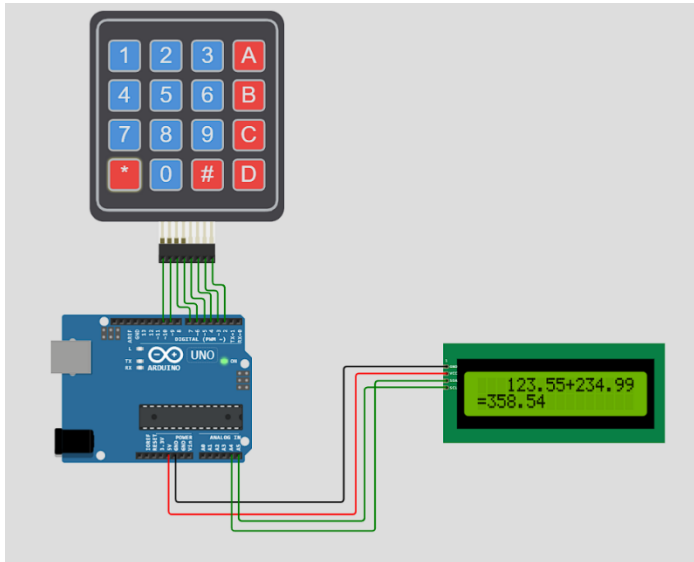


## บทที่ 4

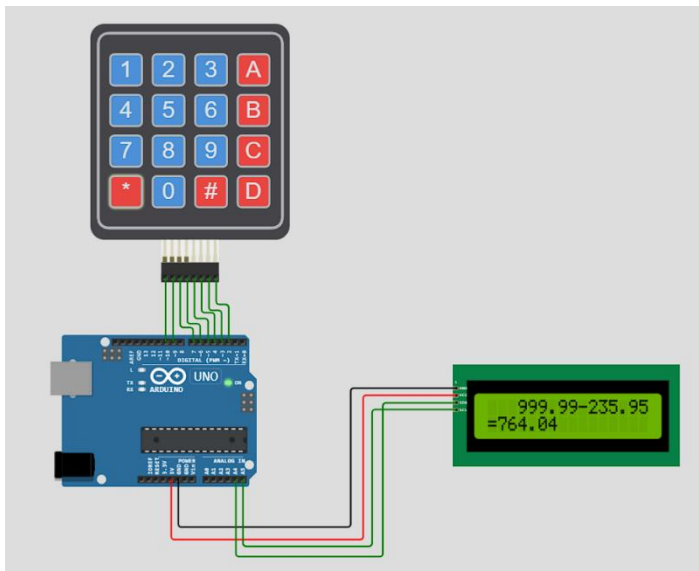
### ผลการทดลอง

รูปภาพผลการทดลองใช้การ simulation วงจรจริงจำทำการแนบ video ใน classroom

#### 1.การบวก

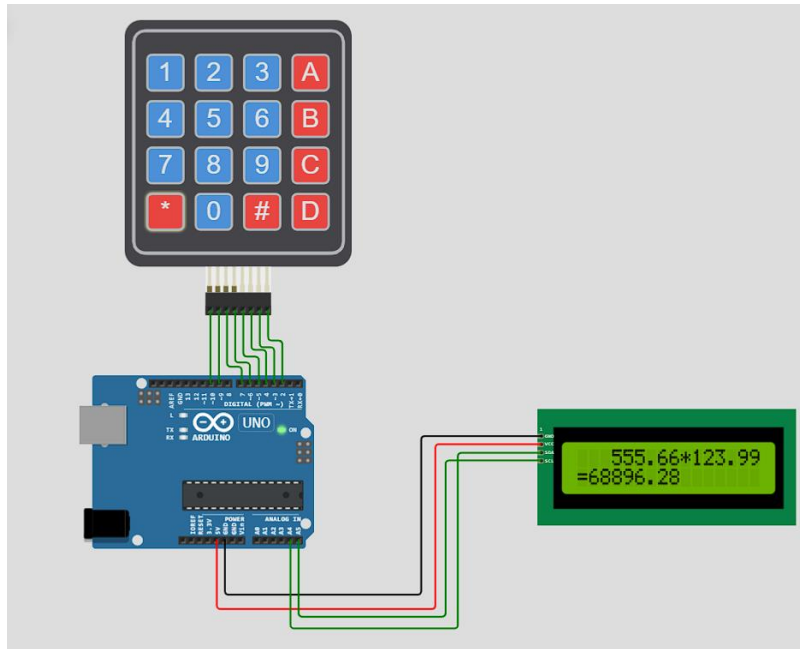


#### 2.การลบ

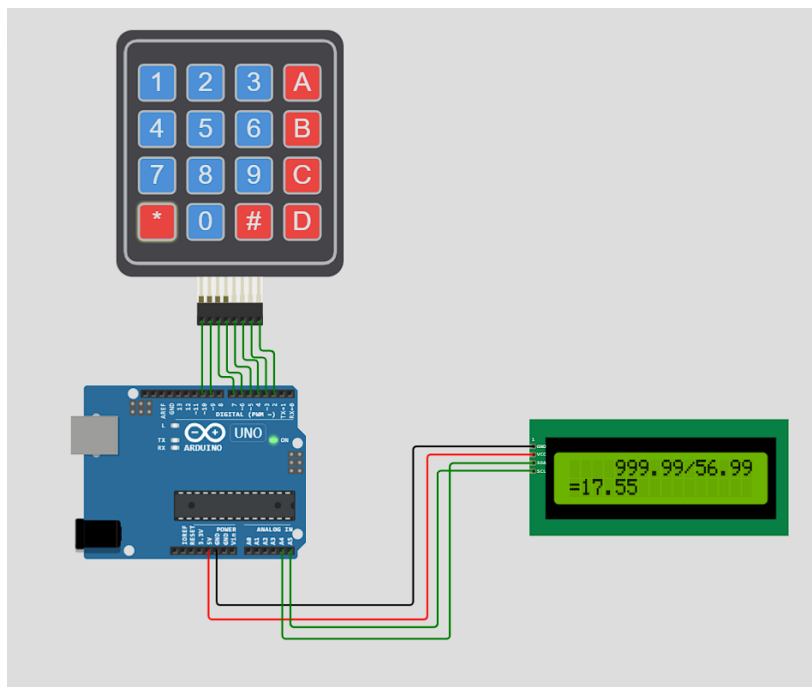




### 3.การคูณ



### 4.การหาร



## บทที่ 5

### สรุปและวิจารณ์

#### สรุปผลการทำงาน

การออกแบบเครื่องคิดเลขโดยใช้ Arduino UNO R3 ได้สำเร็จตามวัตถุประสงค์ที่ตั้งไว้ โดยมีการใช้ Keypad 4\*4 ซึ่งสามารถทำงานได้เป็นอย่างดี และ การแสดงผลของจอ LCD 16\*2 I2C สามารถแสดงผลตัวเลขและคำตอบได้อย่างชัดเจน แต่ไม่สามารถแสดงคำตอบให้ออกมาชัดเจนได้เนื่องจากมีบางกรณีที่ทำให้เกิดช่องว่างทำให้เลือกใช้การแสดงผลออกให้มาชัดเจน

#### วิจารณ์ผลการทำงาน

เนื่องจากการแสดงคำตอบในการคำนวณตัวเลขบางตัวที่มีหลักไม่เท่ากันทำให้มีการเว้นว่างในการแสดงผลคำตอบที่ออกมาจากจอ LCD ทำให้ผู้จัดทำได้เลือกแสดงผลคำตอบเป็นการชัดเจนผู้จัดทำไม่สามารถแก้ไขออกมาให้ทันเวลาขอบเขตการศึกษา