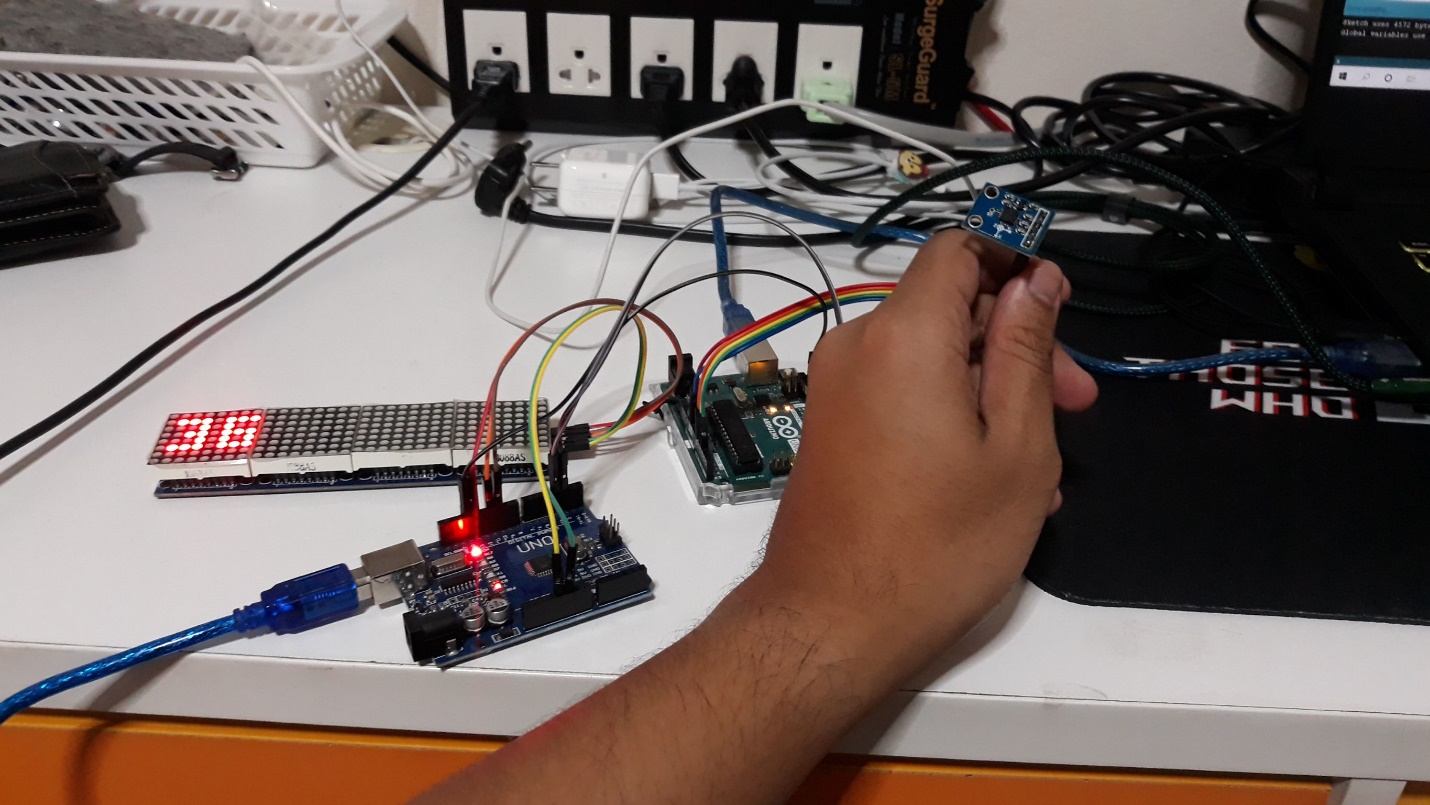
**LEVEL METER PROJECT**

**ขายตรงแบบ 300 %**

****

**LEVEL METER PROJECT’S DETAILED**

**การทดลองนี้ เป็นการวัดมุม(องศา)จากอุปกรณ์**

**Accelerometer(ADXL 335) โดยนำผลของการวัดมุม**

**จะนำมาแสดงในอุปกรณ์ LED Dot Matrix 8x32**

**หมายเหตุ :**

**องศาที่แสดงจะอยู่ในช่วง 0๐-180๐ และวัดองศาในแกน x**

**PICTURE AND DISCUSSION**

**A computer sitting on a desk

Description automatically generated**

**เราจะใช้ Notebook สองเครื่อง/Arduinoในการส่งและรับข้อมูล โดยฝั่งขวาจะเป็นการส่งข้อมูลที่ได้จากการวัดค่ามุม ส่วนฝั่งซ้ายจะเป็นการรับข้อมูลที่ได้จากการวัด แล้วนำข้อมูลนั้นไปแสดงผลใน LED Dot Matrix 8 x 32**

**Accelerometer(ค่ามุม) > Arduino(ตัวส่งค่ามุม)**

**V**

**Transfer data to**

**V**

**Arduino(ตัวรับค่ามุม) -> แสดงค่ามุมใน LED Dot Matrix**

**A screen shot of a computer

Description automatically generated**

**โดยภาพนี้จะเป็นการเอียงไปฝั่งซ้าย จะแสดงค่ามุมสูงสุด (180๐)**

**A person sitting at a desk in front of a computer

Description automatically generated**

**โดยภาพนี้จะเป็นการเอียงไปฝั่งขวา จะแสดงค่ามุม**

**ต่ำสุด (0๐)**

**MAIN PROCESS OF PROJECT**

**ก่อนอื่นเราจะแบ่งการทำงานหลักๆเป็น 2ส่วน ก็คือ**

**1.การวัดค่ามุมจาก ADXL 335**

**2.การรับค่ามุม เพื่อที่จะไปแสดงใน LED Dot Matrix**

**1.การวัดค่ามุมจาก ADXL 335**

**A close up of text on a white background

Description automatically generated**

**1.1 ก่อนอื่นเราจะทำการนำเข้าไรบรารี SoftwareSerial.h เข้าไป เนื่องจากไรบรารีนี้จะเป็นไรบรารีในการส่งข้อมูลจากพอร์ตแรกไปยังพอร์ตที่สอง**

**1.2 ต่อมาเราจะกำหนดให้ PIN ที่วัดค่ามุมเป็น A0 เนื่องจากเป็นตัว Analog วัดค่ามุม และเมื่อเราสังเกตในการวัดค่าดิบครั้งแรก เราจะเห็นว่า ค่าที่เราวัดไม่คงที่และวัดได้ยาก ค่ากระโดดนั้นเรียกว่า ค่า Jitter ดั้งนั้นเราต้องแก้ปัญหานี้ด้วยการ Buffer ค่าที่รับเข้ามา**

**\*\*\* Buffer หมายถึง การเฉลี่ยค่าคลาดเคลื่อนให้คงที่ จากนั้นเราจะวัดค่าได้สะดวกและง่ายขึ้นครับ\*\*\***

**1.3 พอเราวัดค่าได้ เราก็จะทำการกำหนดช่วงโดยใช้ Map();**

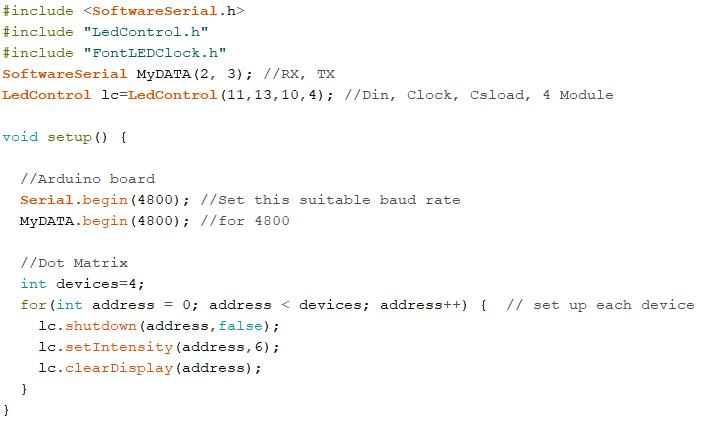
**Map(value, min, max, newmin1, newmax2) คือการปรับช่วงของค่า**

**1.4 ก่อนที่เราจะส่งค่าเราจะกำหนดฟังก์ชัน SoftwareSerial (RX,TX); หมายความว่า มีตัวรับและตัวส่งในแต่ละ PIN และเราก็จะต้องกำหนดฟังก์ชั่นนี้ให้กับอีกพอร์ตด้วย**

**(ชื่อฟังก์ชั่นต้องเป็นชื่อเดี่ยวกัน เช่น MySerial และ MySerial)**

**จบการทำงานในส่วนที่ 1**

**2.การรับค่ามุมเพื่อที่จะไปแสดงใน LED Dot Matrix**

****

**2.1 เราจะป้อนไรบรารีสำคัญที่ใช้งานในส่วนที่สองคือ**

**2.1.1 SoftwareSerial.h -> เพื่อที่จะรับค่าข้อมูลจากพอร์ตแรก**

**2.1.2 LedControl.h -> เพื่อที่จะควบคุมการแสดง LED MATRIX**

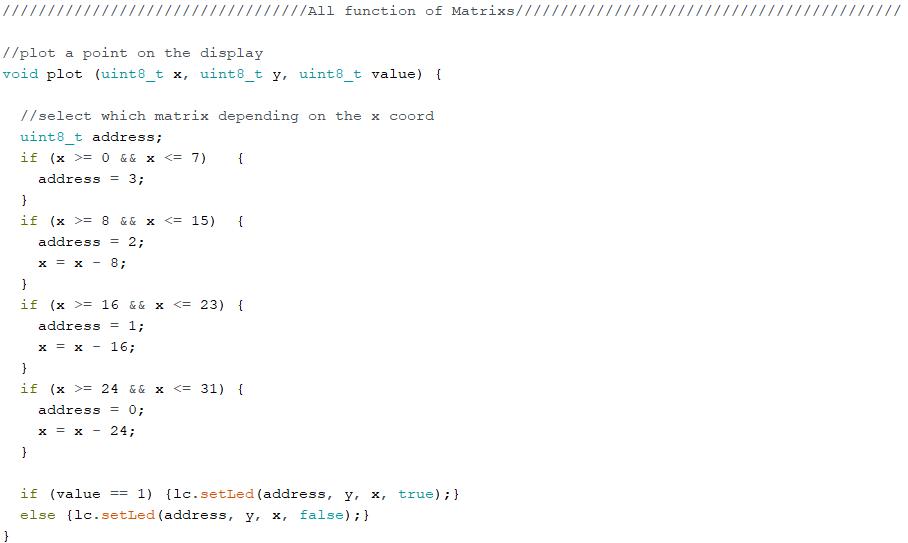
**2.1.3 FontLEDClock.h -> เพื่อที่จะกำหนดขนาดตัวอักษรใน**

**บอร์ด LED Dot Matrix**

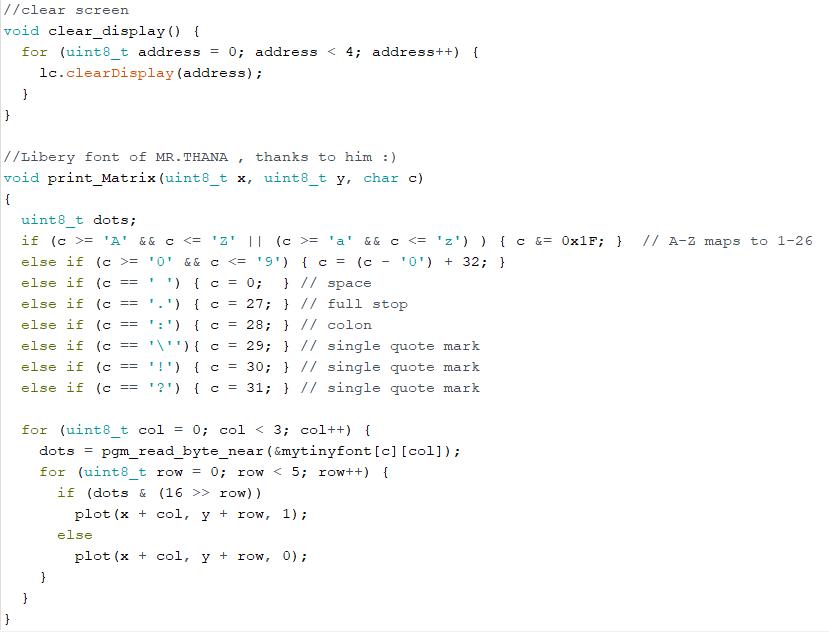
**2.2 เราจะใช้ Baud rate 4800 เนื่องจากถ้าเราส่งข้อมูลเร็วเกินไป**

**โปรแกรมจะเกิดอาการ loss data หรือ การสูญเสียข้อมูลนั่นเอง**

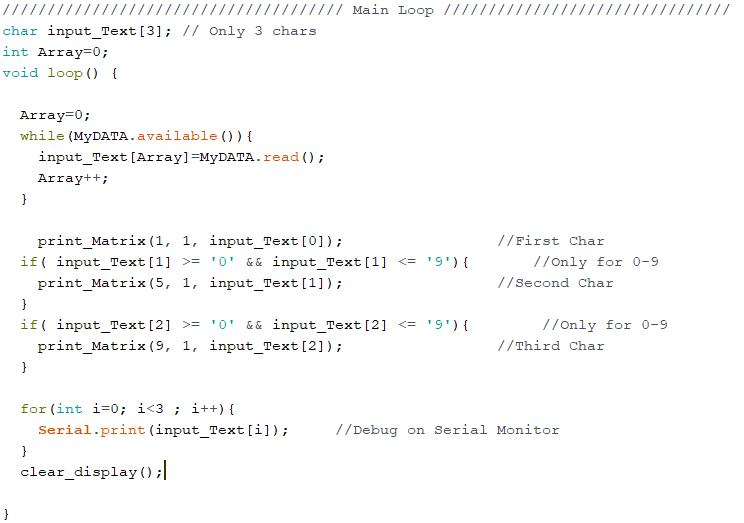
**2.3 เราจะกำหนดว่าเราจะควบคุม module 4 ตัว (8x32) และใช้ setIntensity เพื่อปรับแสง และใช้ clearDisplay เพื่อล้างหน้าจอก่อนที่จะแสดงใหม่**

****

**2.4 เราจะทำการเปลี่ยนพิกัดของ LED Dot Matrix โดยการใช้ฟังก์ชั่นจากรูปด้านบน และทำให้เราสะดวกในการแสดงตัวอักษรอีกด้วย (คล้ายๆ Assignment 4 LED Dot Matrix GAME)**

****

**2.5 ไรบรารี่ FontLEDClock.h เป็นการกำหนดขนาดตัวอักษรที่เราจะนำมาแสดงใน LED Dot Matrix และสะดวกต่อการใช้งานในการแสดงผลด้วย**

****

**2.6 ต่อมาเป็นการทำงานของฟังก์ชั่นหลัก(main) โดยเราจะกำหนดตัวอักษรให้มี 1-3 หลัก เนื่องจากเรากำหนดให้มีองศาระหว่าง 0๐-180(ค่าสูงสุดแสดง 3 หลัก)**

**2.6.1 เมื่อตัวรับข้อมูลพร้อมรับข้อมูล (MyDATA.available()) ก็จะทำการรับข้อมูลมา1ชุด(1ค่ามุม) แล้วค่อยๆป้อนตัวอักษร**

**2.6.2 ต่อมาเราจะทำการป้อนค่ามุมทีละตัวอักษรโดยจะป้อนเป็นชุดๆ ชุดละ1ค่ามุม และใน1ชุดนั้นก็ค่อยๆป้อนทีละหลักจนครบจำนวน เช่น ค่ามุม1ชุดคือ 142 องศา ชุดนี้จะมี3หลัก เราจะป้อน 1ก่อนแล้วค่อยป้อน 4 แล้วตามด้วย 2 พอป้อนข้อมูลครบค่าแล้วพอร์ตตัวนั้นก็จะพร้อมรับข้อมูลชุดถัดไป(ค่าองศาที่วัดได้ใหม่)**

**2.7 หลังจากที่เรารับค่าและป้อนค่าองศาได้แล้ว เราก็จะใช้ฟังก์ชั่นแสดงข้อมูลทีละหลัก และเราก็จะได้เอาท์พุธเป็นค่ามุมที่เราวัดได้จากพอร์ตแรก**

**จบการทำงานส่วนที่ 2**

**END OF PROJECT**