ระบบแจ้งเตือนความปลอดภัยในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์

Server Room Security Alert System via Line Application

ซะห์ฉัน เหสามี

รุสฉัน หะมะ

โครงงานหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

โปรแกรมวิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยราชภัฏสงขลา

2560

ชื่อโครงงาน ระบบแจ้งเตือนความปลอดภัยในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์

ชื่อผู้จัดทำ นายซะห์ลัน เหสามี รหัสนักศึกษา 574235046

นายรุสลัน หะมะ รหัสนักศึกษา 574235065

อาจารย์ที่ปรึกษา

……...……...

(อาจารย์ภาณุกร ภูริปัญญานันท์)

อาจารย์ประจำวิชา

……..……

(ผู้ช่วยศาสตราจารย์พิกุล สมจิตต์)

คณะกรรมการสอบ

……..……ประธาน

(ผู้ช่วยศาสตราจารย์ ดร.ศศลักษณ์ ทองขาว)

……..……กรรมการ

(อาจารย์โชติธรรม ธารรักษ์)

……..……กรรมการ

(อาจารย์ญาณพัฒน์ ชูชื่น)

……..……กรรมการและเลขานุการ

(อาจารย์จักสิทธิ์ โอฬาริกชาติ)

มหาวิทยาลัยราชภัฏสงขลา SONGKHLA RAJABHAT UNIVERSITY

โปรแกรมวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏสงขลา อนุมัติโครงงานเล่มนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

……..

(อาจารย์โชติธรรม ธารรักษ์)

ประธานโปรแกรมวิชาคอมพิวเตอร์

ชื่อโครงงาน ระบบแจ้งเตือนความปลอดภัยในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์
(Server Room Security Alert System via Line Application)

ชื่อผู้จัดทำ นายซะห์ลัน เหสามี รหัสนักศึกษา 574235046
นายรุสลัน หะมะ รหัสนักศึกษา 574235065

ระดับการศึกษา วิทยาศาสตรบัณฑิต

ปีการศึกษา 2560

บทคัดย่อ

องค์กรต่าง ๆ อาทิเช่น โรงพยาบาล มหาวิทยาลัย บริษัทเอกชน รัฐวิสาหกิจ และส่วนงานราชการต่าง ๆ มักจะมีห้องเซิร์ฟเวอร์เป็นที่จัดเก็บเซิร์ฟเวอร์ต่าง ๆ ซึ่งมีข้อมูลสำคัญอยู่ภายในเครื่องเซิร์ฟเวอร์ ดังนั้นการรักษาความปลอดภัยให้กับข้อมูลดังกล่าวจึงเป็นสิ่งจำเป็นอย่างยิ่งเพราะหากเกิดข้อผิดพลาด ผู้ดูแลห้องเซิร์ฟเวอร์ควรจะทราบทันทีว่าเมื่อมีการทำงานผิดพลาดหรือเกิดความผิดปกติกับเครื่องคอมพิวเตอร์จะได้ทำการแก้ไขปัญหาอย่างรวดเร็วเพื่อลดความเสียหายให้กับอุปกรณ์ โดยปกติแล้วเมื่อมีการทำงานผิดพลาดใด ๆ ก็ตามเกิดขึ้นบนเซิร์ฟเวอร์จะมีการแจ้งเตือนผ่าน Nagios หรือซอฟต์แวร์อื่น ๆ ให้ผู้ดูแลระบบทราบ ซึ่งผู้ดูแลระบบจะต้องเรียกใช้งานผ่านคอมพิวเตอร์ ซึ่งบางครั้งผู้ดูแลไม่ได้อยู่ที่หน้าคอมพิวเตอร์ ทำให้ไม่สามารถทราบการแจ้งเตือนดังกล่าวได้ ซึ่งในปัจจุบันได้มีการเปิด API ต่าง ๆ ซึ่งจะทำให้ สามารถเขียนโปรแกรมเพื่อเชื่อมต่อกับแอปพลิเคชันไลน์ได้ หนึ่งในนั้นคือ Notification API หรือการแจ้งเตือนผ่านแอปพลิเคชันไลน์ เพื่อให้ผู้ใช้งานสามารถรับการแจ้งเตือนต่าง ๆ ผ่านแอปพลิเคชันไลน์ได้ทันที

ดังนั้นผู้พัฒนาจึงมีแนวคิดในการพัฒนาการแจ้งเตือนความปลอดภัยต่าง ๆ ที่เกิดขึ้นภายในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์เพื่ออำนวยความสะดวก ในการแจ้งเตือนสิ่งผิดปกติต่าง ๆ อาทิเช่น เมื่อมีเหตุการณ์ไฟดับภายในห้องเซิร์ฟเวอร์หรืออุณหภูมิภายในห้องมีความผิดปกติ ระบบจะแจ้งเตือนให้กับผู้ดูแลผ่านแอปพลิเคชันไลน์ให้ทราบทันที โดยจะทำงานบนสมาร์ทโฟนที่สั่งการผ่านบอร์ดอาดูโน่ ใช้ภาษา C++ ในการพัฒนาโดยเขียนผ่านซอฟต์แวร์ที่มีชื่อว่า Arduino IDE1.6.12 และเซนเซอร์ต่าง ๆ ได้แก่ DHT-22 เพื่อรับค่าของอุณหภูมิและความชื้น MQ-2 เพื่อรับค่าของควัน และ Adapter เพื่อวัดสถานะของไฟฟ้า เพื่อที่จะได้แก้ไขปัญหาดังกล่าวได้

กิตติกรรมประกาศ

การพัฒนาต้นแบบระบบเตือนภัยควบคุมโดยบอร์ดอาดูโน่(The Development Of Alarm Systems Prototype On Arduino) สำเร็จได้ด้วยความช่วยเหลือจากบุคคลหลายฝ่ายดังนี้

อาจารย์ภาณุกร ภูริปัญญานันท์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน ที่คอยให้คำแนะนำคำปรึกษา และเสียสละเวลาส่วนตัว เพื่อตรวจสอบความถูกต้องในการจัดทำโครงงาน รวมถึงให้ความช่วยเหลือในการแก้ปัญหาต่าง ๆ ที่เกิดขึ้นในระหว่างการดำเนินโครงงาน

คณะกรรมการตรวจสอบโครงงานวิทยาศาสตรบัณฑิตทุกท่าน ที่กรุณาตรวจสอบความถูกต้อง และให้คำปรึกษาโครงงานครั้งนี้

อาจารย์โปรแกรมวิชาคอมพิวเตอร์ทุกท่าน ที่คอยประสิทธิ์ประสาทวิชาความรู้ทางวิชาการ ซึ่งทางผู้จัดทำโครงงานได้นำความรู้มาใช้เป็นพื้นฐานในการทำโครงงาน

โครงงานนี้สำเร็จลุล่วงไปได้ โดยการสนับสนุนจากบุคคล ดังรายนามข้างต้น ทางผู้พัฒนาจึงขอขอบคุณทุก ๆ ท่านเป็นอย่างสูงมา ณ โอกาสนี้

คณะผู้จัดทำ

19 กันยายน 2560

มหาวิทยาลัยราชภัฏสงขลา SONGKHLA RAJABHAT UNIVERSITY

สารบัญ

บทคัดย่อ ..... **I**

กิตติกรรมประกาศ ..... **II**

สารบัญ ..... **III**

สารบัญภาพ ..... **VI**

บทที่ 1 บทนำ **10**

หลักการและเหตุผล 10

วัตถุประสงค์ของโครงงาน 11

ขอบเขตและความสามารถของระบบ 11

แผนการดำเนินงาน 11

1. ขั้นตอนและระยะเวลาการดำเนินงาน 11

2. ระยะเวลาการดำเนินงาน 12

เครื่องมือและอุปกรณ์ที่ใช้ในการพัฒนา 12

1. ด้านฮาร์ดแวร์ 12

2. ด้านซอฟต์แวร์ 13

ประโยชน์ที่คาดว่าจะได้รับ 13

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง **14**

อาดูโน่ 14

1. บอร์ดอาดูโน่ 15

2. อาดูโน่ไอดีอี (Arduino IDE) 17

ภาษา C++ 19

1. ตัวแปรและชนิดข้อมูล 20

2. การประกาศใช้งานสายอักขระด้วยแถวลำดับอักขระ 22

เซนเซอร์ 23

1. เซนเซอร์ตรวจจับควัน (Smoke Detector) 23

2. เซนเซอร์ตรวจจับก๊าซ (Gas Sensor) 24

3. เซนเซอร์วัดอุณหภูมิ/ความชื้น 27

สารบัญ(ต่อ)

งานวิจัยที่เกี่ยวข้อง.....30
บทที่ 3 การวิเคราะห์ และออกแบบระบบ.....31
การวิเคราะห์ระบบ.....31
การออกแบบระบบ.....31
การออกแบบวงจร.....41
บทที่ 4 การพัฒนาระบบ และการทดสอบ.....42
การพัฒนาระบบ.....42
การควบคุมการทำงานของ บอร์ด Arduino.....42
การควบคุมการทำงานของจอ Character LCD.....46
การควบคุมการทำงานของเซนเซอร์ตรวจจับควัน DHT-22.....49
การควบคุมการทำงานของเซนเซอร์ตรวจจับควัน MQ-2.....52
การควบคุมการทำงานของ Module SIM 900A.....54
การควบคุมการทำงานของ Adapter เช็คสถานะไฟฟ้า.....57
การควบคุมการทำงานของ Ethernet shield – W5100.....59
การทำงานของ Line API.....62
การควบคุมการทำงานของ Module DC-to-DC Step down Converter.....67
การแก้ไขค่าโค้ดโปรแกรม.....69
การทดสอบ.....74
บทที่ 5 บทสรุปและข้อเสนอแนะ.....79
สรุปผลการดำเนินงาน.....79
ปัญหาและอุปสรรคในการดำเนินงาน.....79
ข้อจำกัดของระบบ.....79
ข้อเสนอแนะ.....80
บรรณานุกรม.....81
ภาคผนวก ก.....82
วิธีติดตั้งโปรแกรม ARDUINO IDE.....82

สารบัญ(ต่อ)

ภาคผนวก ข. 89
โค้ดระบบแจ้งเตือนความปลอดภัยในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์ 89
ภาคผนวก ค. 108
ประวัติผู้จัดทำโครงงาน.....108

สารบัญภาพ

ภาพที่ **หน้า**

2.1 บอร์ดอาดูโน่ UNO R3 15
2.2 บอร์ดอาดูโน่รุ่น LEONARDO 16
2.3 บอร์ดอาดูโน่รุ่น MEGA 16
2.4 บอร์ดรีเลย์ 17
2.5 แสดงลักษณะของแถวลำดับที่มีสมาชิก 8 ตัว 21
2.6 อุปกรณ์ตรวจจับควันชนิดไอออนไนเซชั่น 23
2.7 อุปกรณ์ตรวจจับควันชนิดโฟโต้อิเลคตริก 24
2.8 เซนเซอร์ตรวจจับก๊าซ MQ-7 24
2.9 เซนเซอร์ตรวจจับก๊าซ MQ-2 25
2.10 เซนเซอร์ตรวจวัดก๊าซ MQ-5 26
2.11 เซนเซอร์ตรวจจับก๊าซ MQ-9 26
2.12 เซนเซอร์ตรวจวัดคุณภาพอากาศ MQ-135 27
2.13 ภาพแสดงข้อมูลเชิงเทคนิค 28
2.14 รูปแสดงลำดับของข้อมูลบิต 29
2.15 รูปแสดงลำดับของข้อมูลบิตในการอ่านค่าจากไอซีและความกว้างของช่วง LOW และ HIGH 29
2.16 เซนเซอร์ตรวจวัดอุณหภูมิ/ความชื้น DHT-22 29
3.1 รูปแบบการทำงานของระบบ 32
3.2 บอร์ด ARDUINO UNOMEGA 2560 R3 32
3.3 ARDUINO NANO 3.0 MINI USB ชิฟ CH340 33
3.4 MODULE DC-TO-DC STEP DOWN CONVERTER LM2596 33
3.5 จอ LCD CHARACTER 20X4 34
3.6 UPS 34
3.7 ADAPTER 9V 2A 35
3.8 โมดูลตรวจจับก๊าซ MQ-2 35
3.9 DHT-22 36

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

3.10 ETHERNET SHIELD.....37

3.11 บอร์ด SIM 900A MODULE.....37

3.12 LINE APPLICATION.....38

3.13 . LINE API.....38

3.14 BREADBOARD 170 HOLES.....39

3.15 POWER CONNECTOR 3.5 MM.....39

3.16 สายแพ JUMPER.....40

3.17 รูปแบบการต่อวงจรของระบบ.....41

4.1 บอร์ด Arduino MEGA 2560.....42

4.2 รูปแบบการเขียนโปรแกรมบน ARDUINO.....43

4.3 เลือกรุ่นบอร์ด ARDUINO ที่ต้องการ UPLOAD.....43

4.4 SERIAL PORT ของคอมพิวเตอร์.....44

4.5 UPLOAD โค้ด โปรแกรมไปยังบอร์ด ARDUINO.....44

4.6 LAYOUT ของ BOARD ARDUINO.....45

4.7 จอ LCD CHARACTER 20X4.....46

4.8 การต่อวงจร LCD.....47

4.9 ตัวอย่างโค้ดแสดงข้อความบน LCD.....48

4.10 เซนเซอร์ DHT-22.....49

4.11 การต่อวงจร DHT-22.....50

4.12 ตัวอย่างโค้ดวัดค่าอุณหภูมิและความชื้นสำหรับเซนเซอร์ DHT-22.....51

4.13 โมดูลตรวจจับก๊าซ MQ-2.....52

4.14 วงจรการทำงานของ MQ-2.....52

4.15 การต่อวงจร MQ-2.....53

4.16 ตัวอย่างโค้ดวัดค่าควันสำหรับเซนเซอร์ MQ-2.....54

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

4.17 บอร์ด SIM 900A MODULE....54
4.18 การต่อวงจร MODULE SIM 900A....55
4.19 ตัวอย่างโค้ดการโทรออกสำหรับ MODULE SIM 900A....56
4.20 การเชื่อมต่อ ADAPTER กับ POWER CONNECTER....57
4.21 การต่อวงจรเข้ากับ ADAPTER....58
4.22 ตัวอย่างโค้ดสำหรับการเช็คสถานะไฟ....58
4.23 ETHERNET SHIELD – W5100....59
4.24 วงจรของ ETHERNET SHILED W5100....60
4.25 การเชื่อมต่อ ETHERNET SHIELD กับ ARDUINO....60
4.26 ตัวอย่างโค้ดการเชื่อมต่ออินเทอร์เน็ตสำหรับ ETHERNET SHIELD – W5100....61
4.27 LINE NOTIFY....62
4.28 กรอก USER PASSWORD เพื่อการใช้งาน....63
4.29 ออก ACCESS TOKEN....63
4.30 กรอกข้อมูลที่จะให้แสดง....64
4.31 แสดง ACCESS TOKEN....65
4.32 แจ้งเตือนว่าออก ACCESS TOKEN สำเร็จ....65
4.33 แก้ไขข้อความที่ต้องการส่ง....66
4.34 MODULE DC-TO-DC STEP DOWN CONVERTER LM2596....67
4.35 การต่อใช้งาน MODULE DC-TO-DC STEP DOWN CONVERTER....68
4.36 การกำหนดค่าความชื้นและค่าอุณหภูมิ....69
4.37 การกำหนดค่า TOKEN และ ข้อความที่ต้องการแจ้งเตือน....70
4.38 การกำหนดหมายเลขโทรศัพท์....71
4.39 ตรวจสอบความถูกต้องของโค้ดโปรแกรม....71
4.40 แสดงคำว่า DONE COMPILING.หรือสถานะถูกต้อง....72
4.41 ทำการอัพโหลดโค้ดข้อมูลไปยังระบบ....73

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

4.42 ทดสอบไฟฟ้าดับ 74
4.43 ทดสอบการเช็คอินเทอร์เน็ต 74
4.44 ทดสอบการเชื่อมต่ออินเทอร์เน็ต 75
4.45 ทดสอบเซนเซอร์ MQ-2 75
4.46 การทดสอบเซนเซอร์ DHT-22 76
4.47 ข้อความแจ้งเตือนเกิดไฟไหม้ 76
4.48 ข้อความแจ้งเตือนความชื้น 77
4.49 ข้อความแจ้งเตือนควันไฟ 77
4. 50 การทดลองเปลี่ยนกระแสไฟฟ้า 78
4. 51 การทดลองการโทร 78
ก.1 ดาวน์โหลด Arduino IDE 83
ก. 2 เลือกระบบปฏิบัติการ 83
ก. 3 กด JUST DOWNLOAD 84
ก. 4 บันทึกไฟล์ที่ติดตั้ง 84
ก. 5 UNZIP ไฟล์ติดตั้ง 85
ก. 6 เปิดโปรแกรม 85
ก. 7 หน้าต่างโปรแกรม 86
ก. 8 สร้าง SHORTCUT 86
ก. 9 ไอคอนโปรแกรม 87
ก. 10 เลือกบอร์ดที่ต้องการใช้ 87
ก. 11 เลือก PORT 88
ก. 12 หน้าต่างการเขียนโปรแกรม 88

บทที่ 1

บทนำ

หลักการและเหตุผล

องค์กรต่าง ๆ อาทิเช่น โรงพยาบาล มหาวิทยาลัย บริษัทเอกชน รัฐวิสาหกิจ และส่วนงานราชการต่าง ๆ มักจะมีห้องเซิร์ฟเวอร์เป็นที่จัดเก็บเซิร์ฟเวอร์ต่าง ๆ ซึ่งมีข้อมูลสำคัญอยู่ภายในเครื่องเซิร์ฟเวอร์ ดังนั้นการรักษาความปลอดภัยให้กับข้อมูลดังกล่าวจึงเป็นสิ่งจำเป็นอย่างยิ่งเพราะหากเกิดข้อผิดพลาดผู้ดูแลห้องเซิร์ฟเวอร์ควรจะทราบทันทีว่าเมื่อมีการทำงานผิดพลาดหรือเกิดความผิดปกติกับเครื่องคอมพิวเตอร์จะได้ทำการแก้ไขปัญหาอย่างรวดเร็วเพื่อลดความเสียหายให้กับอุปกรณ์ โดยปกติแล้วเมื่อมีการทำงานผิดพลาดใด ๆ ก็ตามเกิดขึ้นบนเซิร์ฟเวอร์จะมีการแจ้งเตือนผ่าน Nagios หรือซอฟต์แวร์อื่น ๆ ให้ผู้ดูแลระบบทราบ ซึ่งผู้ดูแลระบบจะต้องเรียกใช้งานผ่านคอมพิวเตอร์ ซึ่งบางครั้งผู้ดูแลไม่ได้อยู่ที่หน้าคอมพิวเตอร์ ทำให้ไม่สามารถทราบการแจ้งเตือนดังกล่าวได้ ซึ่งในปัจจุบันได้มีการเปิด API ต่าง ๆ ซึ่งจะทำให้ สามารถเขียนโปรแกรมเพื่อเชื่อมต่อกับแอปพลิเคชันไลน์ได้ หนึ่งในนั้นคือ Notification API หรือการแจ้งเตือนผ่านแอปพลิเคชันไลน์เพื่อให้ผู้ใช้งานสามารถรับการแจ้งเตือนต่าง ๆ ผ่านแอปพลิเคชันไลน์ได้ทันที

ดังนั้นผู้พัฒนาจึงมีแนวคิดในการพัฒนาการแจ้งเตือนความปลอดภัยต่าง ๆ ที่เกิดขึ้นภายในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์เพื่ออำนวยความสะดวก ในการแจ้งเตือนสิ่งผิดปกติต่าง ๆ อาทิเช่น เมื่อมีเหตุการณ์ไฟดับภายในห้องเซิร์ฟเวอร์หรืออุณหภูมิภายในห้องมีความผิดปกติ ระบบจะแจ้งเตือนให้กับผู้ดูแลผ่านแอปพลิเคชันไลน์ให้ทราบทันที โดยจะทำงานบนสมาร์ทโฟนที่สั่งการผ่าน บอร์ดอาดูโน่ ใช้ภาษา C++ ในการพัฒนาโดยเขียนผ่านซอฟต์แวร์ที่มีชื่อว่า Arduino IDE1.6.12 และเซนเซอร์ต่าง ๆ ได้แก่ DHT-22 เพื่อรับค่าของอุณหภูมิและความชื้น MQ-2เพื่อรับค่าของควัน และ Adapter เพื่อวัดสถานะของไฟฟ้า เพื่อที่จะได้แก้ไขปัญหาดังกล่าวได้

วัตถุประสงค์ของโครงงาน

เพื่อพัฒนาระบบแจ้งเตือนความปลอดภัยในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์

ขอบเขตและความสามารถของระบบ

ระบบนี้จะเป็นระบบที่ตรวจพบสิ่งที่ผิดปกติที่เกิดขึ้นในห้องเซิร์ฟเวอร์ เพื่อที่จะได้ส่งการแจ้งเตือนต่าง ๆ ผ่านแอปพลิเคชันไลน์ไปยังผู้เกี่ยวข้อง โดยทำงานผ่านบอร์ด Arduino และมีเซนเซอร์สำหรับวัดค่าต่าง ๆได้แก่ DHT 22 และ MQ-2 แล้วส่งการแจ้งเตือนมายังผู้เกี่ยวข้องผ่านแอปพลิเคชันไลน์ ซึ่งการแจ้งเตือนมีดังนี้

8.1 ระบบแจ้งเตือนไฟดับ

- แจ้งเตือนผ่านแอปพลิเคชันไลน์เมื่อมีเหตุขัดข้องเกี่ยวกับระบบไฟฟ้า ได้แก่ เมื่อมีเหตุการณ์ไฟฟ้าดับ

8.2 ระบบแจ้งเตือนอุณหภูมิ

- แจ้งเตือนผ่านแอปพลิเคชันไลน์เมื่ออุณหภูมิมีค่าสูงหรือต่ำกว่าค่าที่ตั้งไว้

8.3 ระบบแจ้งเตือนความชื้น

- แจ้งเตือนผ่านแอปพลิเคชันไลน์เมื่อมีความชื้นเกินค่าที่กำหนด

8.4 ระบบแจ้งเตือนควัน

- แจ้งเตือนผ่านแอปพลิเคชันไลน์เมื่อเกิดควันภายในห้อง

8.4 ระบบการโทร

- ทำการโทรออกเมื่อการแจ้งเตือนทางแอปพลิเคชันไลน์มีการทำงานผิดพลาด

แผนการดำเนินงาน

1. ขั้นตอนและระยะเวลาการดำเนินงาน

1.1 เสนอเค้าโครงโครงงาน

1.2 ศึกษาการทำงานของระบบเดิม งานวิจัยที่เกี่ยวข้อง และรวบรวมข้อมูล

1.3 วิเคราะห์ความต้องการของผู้ใช้

1.4 ออกแบบระบบงาน

1.5 พัฒนาระบบงาน

1.6 ทดสอบการทำงานของระบบ

1.7 จัดทำเอกสารประกอบโครงงาน

2. ระยะเวลาการดำเนินงาน

ขั้นตอนและระยะเวลาการดำเนินงาน

แผนการดำเนินงาน	มกราคม 2560 – กันยายน 2560											
	มกราคม	กุมภาพันธ์	มีนาคม	เมษายน	พฤษภาคม	มิถุนายน	กรกฎาคม	สิงหาคม	กันยายน	ตุลาคม	พฤศจิกายน	ธันวาคม
1. เสนอเค้าโครงโครงงาน	↔											
2. ศึกษาการทำงานของระบบเดิม งานวิจัยที่เกี่ยวข้อง และรวบรวมข้อมูล		↔	↔									
3. วิเคราะห์ความต้องการของผู้ใช้		↔	↔	↔	↔	↔	↔	↔				
4. ออกแบบและพัฒนาระบบงาน			↔	↔	↔	↔	↔	↔	↔			
5. ทดสอบการทำงานของระบบ				↔	↔	↔	↔	↔	↔			
6. จัดทำเอกสารประกอบโครงงาน			↔	↔	↔	↔	↔	↔	↔			

ภาพที่ 1.1 แผนภาพการดำเนินงาน

เครื่องมือและอุปกรณ์ที่ใช้ในการพัฒนา

1. ด้านฮาร์ดแวร์

1.1 คอมพิวเตอร์

- CPU intel Core i7
- RAM 8 GB
- HARDDISK 1 TB

1.2 Ethernet Shield W5100 อุปกรณ์สำหรับเชื่อมต่อเครือข่าย

1.3 Arduino Mega 2560 บอร์ดอาดูโน่ Mega 2560

1.4 Arduino Nano 3.0 Mini บอร์ดอาดูโน่ Mini 3.0

1.5 MQ-2 เซนเซอร์ตรวจจับควัน

1.6 Adapter 5V 1A	อุปกรณ์สำหรับจ่ายไฟขนาด 5 โวลต์
1.7 Adapter 9V 2A	อุปกรณ์สำหรับจ่ายไฟขนาด 9 โวลต์
1.8 LCD Character 20x4	หน้าจอ LCD Character 20x4
1.9 SIM 900A Module	โมดูล SIM 900A
1.10 SIM Card	ซิมการ์ดสำหรับโทร
1.11 Step down Converter Module	โมดูลแปลงกระแสไฟฟ้า
1.12 DHT 22	เซนเซอร์วัดอุณหภูมิ/ความชื้น
1.13 Breadboard 170 holes	บอร์ดทดลอง ขนาดเล็ก 170 ช่องเสียบ
1.14 Power connector 3.5 mm	แจ็คสำหรับเชื่อมต่อกับ adapter
1.15 Jumper	สาย jumper สำหรับบอร์ดและเซนเซอร์ต่าง ๆ

2. ด้านซอฟต์แวร์

2.1 C++	เป็นภาษาที่ใช้ในการพัฒนา
2.2 Arduino IDE 1.6.12	ซอฟต์แวร์ในการเขียนโปรแกรม
2.3 Line Application	แอปพลิเคชันไลน์
2.4 Line API	ช่องทางการสื่อสารระหว่างบริการกับผู้ใช้
2.5 Microsoft Windows 10	ระบบปฏิบัติการ
2.6 Microsoft Word 2015	โปรแกรมสำหรับจัดทำเอกสาร
2.7 Microsoft PowerPoint 2015	โปรแกรมสำหรับจัดทำเอกสารนำเสนอ

ประโยชน์ที่คาดว่าจะได้รับ

ได้ระบบแจ้งเตือนความปลอดภัยในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์

อาจารย์ที่ปรึกษาโครงงาน

อาจารย์ภาณุกร ภูริปัญญานันท์ อาจารย์ที่ปรึกษา
โปรแกรมวิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสงขลา

อาจารย์ประจำวิชา

ผู้ช่วยศาสตราจารย์พิกุล สมจิตต์ อาจารย์ประจำวิชา
โปรแกรมวิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสงขลา

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การจัดทำโครงงานเรื่องการพัฒนาต้นแบบระบบเตือนภัยควบคุมโดยบอร์ดอาดูโน่ ผู้จัดทำได้ศึกษาค้นคว้า รวบรวมเอกสาร แนวคิดทฤษฎีและงานวิจัยที่เกี่ยวข้อง ดังต่อไปนี้

- อาดูโน่
- ภาษา C++
- เซนเซอร์
- งานวิจัยที่เกี่ยวข้อง

อาดูโน่

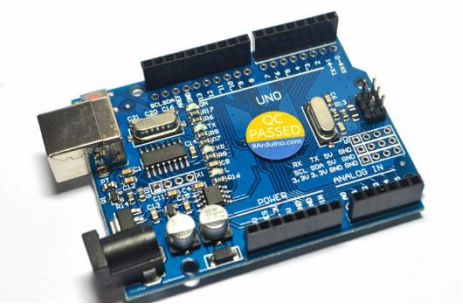
อาดูโน่ (Arduino) เป็นภาษาอิตาลีใช้เป็นชื่อของโครงการพัฒนาไมโครคอนโทรลเลอร์ตระกูล AVR (Automatic Voltage Regulator) แบบเปิดแหล่งต้นทาง (Open Source) ได้รับการปรับปรุงมาจากโครงการพัฒนาเปิดแหล่งต้นทางของ AVR อีกโครงการหนึ่งชื่อว่า Wiring ใช้ AVR เบอร์ ATMEGA128 เป็นชิปที่มีตัวถังแบบ SMD (Surface Mount Device) ทำให้เป็นอุปสรรคสำหรับผู้เริ่มต้นสร้างบอร์ดและต่อวงจรขึ้นมาใช้งานกันเอง และบอร์ดมีขนาดใหญ่เกินความจำเป็นสำหรับผู้เริ่มต้นจึงไม่ได้รับความนิยม แต่หลังจากที่ทีมงานอาดูโน่นำรหัสต้นทาง (Source Code) ของ Wiring มาพัฒนาปรับปรุงใหม่โดยสามารถใช้กับไมโครคอนโทรลเลอร์ AVR ขนาดเล็กทั้ง MEGA8 และ MEGA168 ได้ จึงทำให้ระบบวงจรของบอร์ดลดลงกว่า Wiring มากและยังใช้อุปกรณ์น้อยชิ้นทำให้ง่ายต่อการต่อวงจรใช้งานกันเองและประหยัดต้นทุนในการสร้างบอร์ด ด้วยเหตุนี้เองทำให้อาดูโน่ได้รับความนิยมจากผู้ใช้งานทั่วโลก

1. บอร์ดอาดูโน่

อาดูโน่เป็นบอร์ดไมโครคอนโทรลเลอร์ (Microcontroller) ใช้ไมโครคอนโทรลเลอร์ AVR ขนาดเล็กเป็นตัวประมวลผลและสั่งงานเหมาะสำหรับนำไปใช้ในการศึกษาและเรียนรู้ระบบไมโครคอนโทรลเลอร์ และนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมอุปกรณ์รับเข้า/ส่งออกต่าง ๆ ได้มากมายทั้งในแบบที่เป็นการทำงานตัวเดียวอิสระ หรือเชื่อมต่อสั่งงานร่วมกับอุปกรณ์อื่น ๆ เช่น คอมพิวเตอร์ส่วนบุคคล ทั้งนี้เนื่องมาจากว่าอาดูโน่สนับสนุนการเชื่อมต่อกับอุปกรณ์รับเข้า/ส่งออกต่าง ๆ ได้มากมาย ทั้งแบบดิจิทัล (Digital) และอนาล็อก (Analog) เช่น การรับค่าจากสวิตซ์หรืออุปกรณ์ตรวจจับ (Sensor) แบบต่าง ๆ รวมไปถึง การควบคุมอุปกรณ์ส่งออกตั้งแต่ แอลอีดี หลอดไฟ มอเตอร์ รีเลย์ ฯลฯ โดยระบบฮาร์ดแวร์ของอาดูโน่สามารถสร้างและประกอบขึ้น ในกรณีที่ผู้ใช้มีความรู้ด้านอิเล็กทรอนิกส์

ไมโครคอนโทรลเลอร์เป็นอุปกรณ์ชิปไอซีพิเศษ ชนิดหนึ่งที่สามารถเขียนโปรแกรมเพื่อควบคุมการทำงานตามที่ต้องการได้ ภายในไมโครคอนโทรลเลอร์จะประกอบไปด้วย หน่วยประมวลผล, หน่วยความจำชั่วคราว (RAM), หน่วยความจำถาวร (ROM), พอร์ตอินพุต/เอาท์พุต

1.1 บอร์ดอาดูโน่รุ่น UNO ใช้ไมโครคอนโทรลเลอร์ ATMEGA328 เป็นไมโครคอนโทรลเลอร์หลักใช้แพ็คเกจแบบ DIP (Dual In-Line Package) มีช่องทางรับเข้า/ส่งออกทั้งหมด 20 ขา แบ่งเป็นดิจิทัลรับเข้า/ส่งออก 16 ขา และอินพุตแบบอนาล็อกอีก 6 ขา ใช้คริสตัลความถี่ 16 เมกะเฮิรตซ์ มี USB (Universal Serial Bus) เป็นตัวเชื่อมต่อแบบ B หัวต่อไฟฟ้ากระแสตรง พอร์ตสำหรับโปรแกรมแบบ ICSP (In-Circuit Serial Programming) และสวิตซ์สำหรับตั้งค่าใหม่ ตัวบอร์ดสามารถเลือกแหล่งจ่ายไฟได้อัตโนมัติระหว่างช่องทาง USB กับแหล่งจ่ายภายนอก บนบอร์ดอาดูโน่รุ่น UNO มีไมโครคอนโทรลเลอร์อีกตัวคือ ATMEGA16U2 ทำหน้าที่เป็นแปลงช่องทาง USB เป็นช่องทางอนุกรม



ภาพที่ 2.1 บอร์ดอาดูโน่ UNO R3

ที่มา : https://www.9arduino.com/product/2/arduino-uno-r3-

1.2 บอร์ดอาดูโน่รุ่น Leonardo ใช้ไมโครคอนโทรลเลอร์ ATMEGA32U4 มีช่องทางรับเข้า/ส่งออกดิจิทัลทั้งหมด 20 ขา สามารถใช้ PWM (Pulse-Width Modulation) ได้ 7 ขา และสามารถใช้อนาล็อก 12 ขา ใช้คริสตันความถี่ 16 เมกะเฮิรตซ์ ใช้ไมโคร USB เป็นตัวเชื่อมต่อ หัวต่อไฟฟ้ากระแสตรงช่องทางสำหรับโปรแกรมแบบ ICSP และสวิตซ์สำหรับตั้งค่าใหม่ เชื่อมต่อเข้ากับเครื่องคอมพิวเตอร์ด้วยสาย USB หรือต่อกับตัวแปลงไฟฟ้ากระแสตรง

ภาพที่ 2.2 บอร์ดอาดูโน่รุ่น Leonardo

ที่มา : https://www.9arduino.com/product/2/arduino-uno-r3-

1.3 บอร์ดอาดูโน่รุ่น MEGA เป็นบอร์ดอาดูโน่รุ่น MEGA ที่ออกแบบมาสำหรับงานที่ต้องใช้ I/O มากกว่า Arduino Uno R3 เช่น งานที่ต้องการรับสัญญาณจาก Sensor หรือควบคุมมอเตอร์ Servo หลาย ๆ ตัว ทำให้ Pin I/O ของบอร์ด Arduino Uno R3 ไม่สามารถรองรับได้ ทั้งนี้บอร์ด Mega 2560 R3 ยังมีความหน่วยความจำแบบ Flash มากกว่า Arduino Uno R3 ทำให้สามารถเขียนโค้ดโปรแกรมเข้าไปได้มากกว่า ในความเร็วของ MCU ที่เท่ากัน ในภาพที่ 2.3

ภาพที่ 2.3 บอร์ดอาดูโน่รุ่น MEGA

ที่มา : https://www.arduinoall.com/product/507/arduino-mega-2560-r3-

1.4 บอร์ดรีเลย์ (Relay Shield) เป็นอุปกรณ์เสริมที่ติดรีเลย์เข้าไปทำหน้าที่ ตัดและต่อวงจรคล้ายกับสวิตซ์ โดยใช้หลักการหน้าสัมผัส และการทำงานของรีเลย์ต้องจ่ายไฟให้ตาม ที่กำหนด เพราะเมื่อจ่ายไฟให้กับตัวรีเลย์ มันจะทำให้หน้าสัมผัสติดกัน กลายเป็นวงจรปิด และ ตรงข้ามทันทีที่ไม่ได้จ่ายไฟให้มัน มันก็จะกลายเป็นวงจรเปิด ไฟฟ้าที่เราใช้ป้อนให้กับตัวรีเลย์ ก็จะเป็นไฟฟ้าที่มาจากแหล่งจ่ายไฟฟ้าของบอร์ดอาดูโน่ ดังนั้นทันทีที่เปิดเครื่องทำให้รีเลย์ทำงาน บอร์ดรีเลย์ยังสามารถติดอุปกรณ์เสริมเข้าไปได้อีก เช่น บลูทูธ (Bluetooth) NRF24L01 Wireless Interface และยังสามารถต่อตัวรับรู้ (Sensor) ได้ 6 ช่องทาง ดังแสดงตัวอย่างบอร์ดรีเลย์

ภาพที่ 2.4 บอร์ดรีเลย์

ที่มา : https://www.9arduino.com/product/195/module-

2. อาดูโน่ไอดีอี (Arduino IDE)

เป็นโปรแกรมภาษาซีประยุกต์ที่มีโครงสร้างของตัวภาษาใกล้เคียงกันกับภาษาซีมาตรฐาน (ANSI-C) ซึ่งมีการปรับปรุงรูปแบบการเขียนโปรแกรมและให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้ง่ายและสะดวกมากกว่าการเขียนภาษาซีตามแบบมาตรฐานของ ANSI-C

ภาษาซีของอาดูโน่จัดแบ่งรูปแบบโครงสร้างของการเขียนโปรแกรมออกเป็นส่วนย่อยหลายส่วน โดยเรียกแต่ละส่วนว่าฟังก์ชัน และเมื่อนำฟังก์ชันมารวมเข้าด้วยกันก็จะเรียกว่าโปรแกรมโดยโครงร้างการเขียนโปรแกรมของอาดูโน่นั้น ทุก ๆ โปรแกรมจะต้องประกอบไปด้วยฟังก์ชันจำนวนเท่าใดก็ได้ แต่อย่างน้อยที่สุดต้องมีฟังก์ชันจำนวน 2 ฟังก์ชันคือ setup() และ loop() ดังตัวอย่าง

โครงสร้างพื้นฐานของภาษาซีที่ใช้กับอาดูโน่ประกอบด้วย 2 ส่วน คือ setup และ loop

- header คือส่วนที่นำเข้าไลบรารี ที่ใช้ในการเขียนโปรแกรมและประกาศตัวแปรที่ใช้ในการเขียนโปรแกรม
- setup คือส่วนฟังก์ชันบังคับที่ต้องกำหนดทุกโปรแกรม ใช้เป็นตัวกำหนดขอบเขตของฟังก์ชัน ใช้สำหรับบรรจุคำสั่งในส่วนที่ต้องการให้โปรแกรมทำงานเพียงรอบเดียวช่วงเริ่มต้นทำงานของโปรแกรมครั้งแรก ได้แก่ คำสั่งเดียวกับการตั้งค่าการทำงาน เช่น การกำหนดหน้าที่การใช้งานของช่องทางและการกำหนดค่าอัตราบอด (Baud Rate) สำหรับใช้งานช่องทางสื่อสารอนุกรม เป็นต้น
- loop คือส่วนฟังก์ชันบังคับที่ต้องกำหนดทุกโปรแกรมเช่นเดียวกันกับฟังก์ชัน setup() ฟังก์ชัน loop() ใช้บรรจุคำสั่งที่ต้องการให้โปรแกรมทำงานเป็นวงรอบซ้ำกัน

ตัวอย่างการ Loop ของโปรแกรม

```
#include<>                                   header
void setup(){
        pinMode(13, OUTPUT);                setup
}
void loop(){
        digitalWrite(13, HIGH);
      delay(1000);                          loop
      digitalWrite(13, LOW);
      delay(1000); }
```

ภาษา C++

ภาษา C++ เป็นภาษาโปรแกรมคอมพิวเตอร์อเนกประสงค์ มีโครงสร้างภาษาที่มีการจัดชนิดข้อมูลแบบสแตติก (Statically Typed) และสนับสนุนรูปแบบการเขียนโปรแกรมที่หลากหลาย (Multi-Paradigm Language) ได้แก่ การโปรแกรมเชิงกระบวนคำสั่ง การนิยามข้อมูล การโปรแกรมเชิงวัตถุ และการโปรแกรมแบบเจเนริก (Generic Programming) ภาษา C++ เป็นภาษาโปรแกรมเชิงพาณิชย์ที่นิยมมากภาษาหนึ่งนับตั้งแต่ช่วงทศวรรษ 1990

เบียเนอสเดราสดร็อบ (Bjarne Stroustroup) จากเบลล์แล็บส์ (Bell Labs) เป็นผู้พัฒนาภาษา C++ (เดิมใช้ชื่อ “C with classes”) ในปี ค.ศ. 1983 เพื่อพัฒนาภาษาซีดั้งเดิม สิ่งที่พัฒนาขึ้นเพิ่มเติมนั้นเริ่มจากการเพิ่มเติมการสร้างคลาส จากนั้นก็เพิ่มคุณสมบัติต่าง ๆ ตามมาได้แก่เวอร์ชวลฟังก์ชัน การโอเวอร์โหลดโอเปอเรเตอร์ การสืบทอดหลายสาย เทมเพลต และการจัดการเอกเซพชัน มาตรฐานของภาษา C++ ได้รับการรับรองในปี ค.ศ. 1998 เป็นมาตรฐาน ISO/IEC 14882:1998 เวอร์ชันล่าสุดคือเวอร์ชันในปี ค.ศ. 2003 ซึ่งเป็นมาตรฐาน ISO/IEC 14882:2003 ในปัจจุบันมาตรฐานของภาษาในเวอร์ชันใหม่ (รู้จักกันในชื่อ C++0X) กำลังอยู่ในขั้นพัฒนา

รูปแบบของการออกแบบภาษา C++

- ภาษา C++ ได้ถูกออกแบบมาเพื่อเป็นภาษาสำหรับการเขียนโปรแกรมทั่วไป สามารถรองรับการเขียนโปรแกรมในระดับภาษาเครื่องได้เช่นเดียวกับภาษาซี
- ในทางทฤษฎีภาษา C++ ควรจะมีความเร็วเทียบเท่าภาษาซี แต่ในการเขียนโปรแกรมจริงนั้นภาษา C++ เป็นภาษาที่มีการเปิดกว้างให้โปรแกรมเมอร์เลือกรูปแบบการเขียนโปรแกรม ซึ่งทำให้มีแนวโน้มที่โปรแกรมเมอร์อาจใช้รูปแบบที่ไม่เหมาะสม ทำให้โปรแกรมที่เขียน มีประสิทธิภาพต่ำกว่าที่ควรจะเป็น และภาษา C++ นั้นเป็นภาษาที่มีความซับซ้อนมากกว่าภาษาซี จึงทำให้มีโอกาสเกิดบั๊กขณะคอมไพล์มากกว่า
- ภาษา C++ ได้รับการออกแบบเพื่อเพื่อเข้ากันได้กับภาษาซีในเกือบทุกกรณี
- มาตรฐานของภาษา C++ ถูกออกแบบมาเพื่อให้ไม่มีการเจาะจงแพลตฟอร์มคอมพิวเตอร์
- ภาษา C++ ถูกออกแบบมาให้รองรับรูปแบบการเขียนโปรแกรมที่หลากหลาย

1. ตัวแปรและชนิดข้อมูล

ชนิดของข้อมูลที่ C++ จัดเตรียมไว้ให้ใช้งานแบ่งได้ 2 ประเภทคือ ชนิดข้อมูลพื้นฐาน เช่น บูลีน (Boolean), จำนวนเต็ม (Integer) และอักขระ (Character) เป็นต้น และชนิดข้อมูลผสม เช่น สายอักขระ (String), Structure และการแจ้งนับ (Enumeration) เป็นต้น ชนิดข้อมูลดังกล่าว จะถูกกำหนดให้กับตัวแปร ซึ่งใช้ในการระบุพื้นที่ที่ใช้ในการเก็บข้อมูล โดยภายในตัวแปร จะเกี่ยวข้องกับข้อมูล 3 ส่วนนี้

- ตำแหน่งหรือสถานที่ที่ใช้ในการเก็บข้อมูล
- ค่าที่เก็บไว้ในตัวแปรคืออะไร
- ชนิดของข้อมูลที่ถูกเก็บไว้

ตัวอย่างการกำหนดตัวแปร

```
int my_number; //ประกาศตัวแปรชนิดจ้านวนเต็ม

my_number=10; //กำหนดค่าให้กับตัวแปร
```

1.1 ข้อมูลชนิดอักขระ (Character)

อักขระ หรือ Char เป็นข้อมูลที่ใช้เก็บข้อมูลตัวอักขระต่าง ๆ เช่น ตัวอักษรและตัวเลข เป็นต้น โดยข้อมูลชนิดอักขระจะใช้เลขรหัส (Number Code) เพื่อแทนตัวอักษร ตัวเลข และสัญลักษณ์ต่าง ๆ โดยทั่วไปตัวเลขที่นำมาใช้จะต้องสามารถแทนตัวอักขระต่าง ๆ ได้อย่างครบถ้วน รหัสที่นิยมนำมาใช้แทนตัวอักษรกนอย่างแพร่หลาย ได้แก่รหัสแอสกี (ASCII) โดยจะกำหนดเลขรหัสแทนอักขระแต่ละตัว เช่น เลขรหัส “65” ใช้แทนตัวอักษร A เป็นต้น

ตัวอย่างการกำหนดตัวแปรชนิดอักขระ

```
char ch1=97;//ch1 เก็บตัวอักษร a

char ch2='a';//ch2 เก็บตัวอักษร a
```

จากโค้ดข้างต้นจะเห็นว่ามีการกำหนดค่าตัวแปร ch1 ด้วยตัวเลข 97 ซึ่งในรหัสแอสกีคือตัวอักษร a ลักษณะดังกล่าวเป็นการระบุค่าของตัวแปรชนิดอักขระด้วยเลขรหัส ส่วนตัวแปร ch2 ใช้วิธีกำหนดตัวอักษร a ซึ่งตัวแปรทั้งสองจะมีค่าเท่ากัน

1.2 ข้อมูลชนิดจำนวนเต็ม (Integer)

จำนวนเต็มเป็นข้อมูลชนิดตัวเลขจำนวนเต็ม เช่น 0,8,99 และ -2388 เป็นต้น ในระบบคอมพิวเตอร์มีข้อจำกัดในเรื่องจำนวนหน่วยความจำที่ใช้เก็บข้อมูล ทำให้แทนตัวเลขได้ในช่วงที่จำกัดโดย C++ ได้จัดเตรียมข้อมูลชนิดจำนวนเต็ม ซึ่งใช้ขนาดของหน่วยความจำต่างกันไว้เลือกใช้หลายชนิด ทำให้แทนจำนวนข้อมูลได้ไม่เท่ากัน นอกจากนี้ข้อมูลบางชนิดยังเป็นได้ทั้งค่าบวกและลบ (Signed Type) แต่บางชนิดไม่สามารถแทนค่าลบได้ (Unsingned Type) โดย C++ ได้จัดเตรียมชนิดข้อมูลพื้นฐานสำหรับแทนข้อมูลชนิดจำนวนเต็มไว้ 3 ชนิดคือ shot int และ long

1.3 ข้อมูลชนิดแถวลำดับ (Array)

แถวลำดับเป็นรูปแบบของข้อมูลที่สามารถเก็บชุดของตัวแปรที่แสดงอยู่ในรูปของลำดับที่เพื่อเก็บค่าของข้อมูลที่เป็นชนิดเดียวกันแถวลำดับถือเป็นชนิดข้อมูลที่เรียกว่าชนิดผสม (Compound Type) เนื่องจากถูกสร้างข้อมูลชนิดอื่น โดยสมาชิกที่อยู่ในแถวลำดับจะถูกเก็บไว้ในหน่วยความจำแบบต่อเนื่องกัน ส่วนตัวแปรทั่วไปจะถูกเก็บไว้ในหน่วยความจำในตำแหน่งที่ไม่ต่อเนื่องกัน ให้พิจารณารูปแบบการเก็บข้อมูลของแถวลำดับ ดังตัวอย่าง

Int my_array[8];

ภาพที่ 2.5 แสดงลักษณะของแถวลำดับที่มีสมาชิก 8 ตัว

คำสั่งที่ใช้ในการประกาศใช้งานแถวลำดับจะประกอบด้วยข้อมูล 3 ส่วน ดังนี้

- ชนิดข้อมูลของแถวลำดับที่จะกำหนดให้กับสมาชิก
- ชื่อของแถวลำดับ
- จำนวนสมาชิกของแถวลำดับ

1.4 ข้อมูลชนิดสายอักขระ (String)

สายอักขระเป็นการเก็บข้อมูลในรูปชุดของตัวอักษรของตัวอักษรเรียงต่อกัน แนวคิดดังกล่าวจะนำตัวอักษรแต่ละไบต์มาเก็บไว้ในลักษณะแถวลำดับของอักขระ การเก็บตัวอักษรของสายอักขระทำให้การจัดการข้อมูลง่ายขึ้น การประกาศใช้ข้อมูลชนิดสายอักขระใน C++ ทำได้ 2 แบบ คือ การใช้แถวลำดับอักขระและการใช้คลาสไลบรารีที่เรียกว่า String

2. การประกาศใช้งานสายอักขระด้วยแถวลำดับอักขระ

การประกาศใช้สายอักขระด้วยแถวลำดับอักขระ (Character Array) เป็นวิธีแบบเดิมที่ใช้ในภาษา C โดยคอมไพเลอร์ (Compiler) จะใส่สมาชิกสุดท้ายเป็น \0 (ในรหัส ASCII คือ 0) ให้โดยอัตโนมัติเพื่อใช้แทนอักขระว่างซึ่งแสดงแสดงจุดสิ้นสุดของสายอักขระ ข้อความที่ต้องการกำหนดให้กับสายอักขระจะอยู่ในเครื่องหมาย “ ” ให้พิจารณาตัวอย่างการประกาศใช้งานสายอักขระดังต่อไปนี้

```
Char string1[6] = “tiger” // ประกาศใช้งานสายอักขระโดยระบุสมาชิก

Char string2[] = “lion”//ประกาศใช้งานสายอักขระโดยไม่ระบุจำนวน
```

คำสั่งข้างต้นเป็นการประกาศใช้สายอักขระโดยบรรทัดแรกระบุจำนวนสมาชิกของสายอักขระไว้จำนวน 6 ตัว และกำหนดข้อมูลเป็น “tiger” ซึ่งประกอบด้วยตัวอักษรทั้งหมด 5 ตัว โดยคอมไพเลอร์จะเพิ่มอักขระว่างให้โดยอัตโนมัติ ทำให้สายอักขระนี้เก็บข้อมูลตัวอักษรทั้งหมด 6 ตัวพอดี ส่วนการประกาศใช้งานสายอักระในบรรทัดที่ 2 จะไม่มีการระบุจำนวนสมาชิกของสายอักขระไว้ คอมไพเลอร์จะตรวจสอบจำนวนสมาชิกของสายอักขระจากจำนวนตัวอักษรที่กำหนดไว้และเพิ่มอักขระว่างเข้าไปอีก 1 ตัวให้โดยอัตโนมัติ

เซนเซอร์

เซนเซอร์ (Sensor) เป็นอุปกรณ์ตรวจจับสัญญาณหรือปริมาณทางฟิสิกส์ต่าง ๆ เช่น อุณหภูมิ เสียง แสง การสัมผัส เป็นต้น ปัจจุบันมีการนำระบบ Sensor มาใช้ในหลายรูปแบบ เช่น Motion Sensor ระบบตรวจจับความเคลื่อนไหว, Accelerometer Sensor เซนเซอร์วัดความเร่ง, Orientation Sensor เซนเซอร์วัดความเอียง, Sound Sensor เซนเซอร์ตรวจวัดระดับเสียง, Magnetic Sensor เซนเซอร์ตรวจจับความเข้มสนามแม่เหล็ก, Light Sensor เซนเซอร์ตรวจจับแสงสว่างสำหรับการปรับแสงบนหน้าจออัตโนมัติ และ Proximity Sensor ระบบเปิด/ปิดหน้าจออัตโนมัติขณะสนทนาแนบหู เป็นต้น

1. เซนเซอร์ตรวจจับควัน (Smoke Detector)

แบ่งออกเป็น 2 แบบดังนี้

1.1 อุปกรณ์ตรวจจับควันชนิดไอออนไนเซชั่น (Ionization Smoke Detector)

อุปกรณ์ชนิดนี้เหมาะสำหรับใช้ตรวจจับสัญญาณควันในระยะเริ่มต้นที่มีอนุภาคของควันเล็กมาก Ionization Detector ทำงานโดยใช้หลักการเปลี่ยนแปลงคุณลักษณะทางไฟฟ้า โดยใช้สารกัมมันตภาพรังสีปริมาณน้อยมากซึ่งอยู่ใน Chamber ซึ่งจะทำปฏิกิริยากับอากาศที่อยู่ระหว่างขั้วบวกและลบ ทำให้ความนำไฟฟ้า (Conductivity) เพิ่มขึ้นมีผลให้กระแสสามารถไหลผ่านได้โดยสะดวก เมื่อมีอนุภาคของควันเข้ามาใน Sensing Chamber นี้ อนุภาคของควันจะไปรวมตัวกับอิออน จะมีผลทำให้การไหลของกระแสลดลงด้วย ซึ่งทำให้ตัวตรวจจับควันแจ้งสถานะ Alarm ทันที

ภาพที่ 2.6 อุปกรณ์ตรวจจับควันชนิดไอออนไนเซชั่น

ที่มา : https://ae01.alicdn.com/kf/HTB1Oj5WRXXXXXX6XpXXq6xXFXXXV.jpg_220x220.jpg

1.2 อุปกรณ์ตรวจจับควันชนิดโฟโต้อิเลคตริก (Photoelectric Smoke Detector)

เหมาะสำหรับใช้ตรวจจับสัญญาณควันในระยะที่มีอนุภาคของควันที่ใหญ่ขึ้น Photoelectric Smoke Detector ทำงานโดยใช้หลักการสะท้อนของแสง เมื่อมีควันเข้ามาในตัวตรวจจับควันจะไปกระทบกับแสงที่ออกมาจาก Photoemiter ซึ่งไม่ได้ส่องตรงไปยังอุปกรณ์รับแสง Photo Receptor แต่แสงดังกล่าว บางส่วนจะสะท้อนอนุภาคควัน และหักเหเข้าไปที่ Photo Receptor ทำให้วงจรตรวจจับของตัวตรวจจับควันส่งสัญญาณแจ้ง Alarm

ภาพที่ 2.7 อุปกรณ์ตรวจจับควันชนิดโฟโต้อิเลคตริก

ที่มา : http://img.tarad.com/shop/t/thaimakro/img-lib/spd_20110703213200_b.jpg

2. เซนเซอร์ตรวจจับก๊าซ (Gas Sensor)

เซนเซอร์ก๊าซ (Gas Sensor) ทำหน้าที่คล้ายจมูกของหุ่นยนต์ที่แยกความหนาแน่นของแก๊สที่อยู่ในอากาศ โดยโครงสร้างภายในจะมีลักษณะพิเศษที่สามารถเปลี่ยนความหนาแน่นของแก๊สเป็นสัญญาณไฟฟ้าได้

ภาพที่ 2.8 เซนเซอร์ตรวจจับก๊าซ MQ-7

ที่มา : https://i.lnwfile.com/_/i/_raw/f0/tk/5j.jpg

โมดูลตรวจจับก๊าซ MQ-2 ให้ค่าของปริมาณก๊าซในย่าน 100ถึง 5000 ppm (pieces per million - เศษในล้านส่วน) ตามชนิดของก๊าซหรือไอระเหยที่ตรวจจับได้ ประกอบด้วย ก๊าซโพรเพน

(propane) ได้ค่า 200 ถึง 5000ppm, มีเธน (methane) ได้ค่า5000 ถึง 20000, ไฮโดรเจน (hydrogen) ได้ค่า 300 ถึง 5000,แอลกอฮอล์ (alcohol) 100 ถึง 20000 และควัน ถึงแม้ว่า ตัว MQ-2 จะตรวจจับก๊าซได้หลากหลาย แต่ถ้าสังเกตจะเห็นว่าค่าของการตรวจจับที่ได้มีช่วงที่ทับซ้อนกันอยู่ ซึ่ง MQ-2 ไม่ได้มีความสามารถในการแยกแยะชนิดของก๊าซได้ ดังนั้นผู้ใช้งานจึงต้องกำหนดค่าอ้างอิงของก๊าซที่ต้องการตรวจจับ และไม่แนะนำให้ใช้ในการตรวจจับและแยกแยะชนิดของก๊าซ จากการทดสอบขอแนะนำให้นำไปใช้การตรวจจับควันหรือตรวจจับว่ามีก๊าซอื่น ๆ เข้ามาปะปนในพื้นที่ตรวจจับหรือไม่ จะได้ผลการทำงานที่ดีมากกว่า

ภาพที่ 2.9 เซนเซอร์ตรวจจับก๊าซ MQ-2

ที่มา : https://www.mysensors.org/uploads/57c3ebeb071cb0e34c90057a/image/mq2.png

โมดูลตรวจวัดก๊าซ ที่ไวต่อก๊าซไวไฟในกลุ่ม LPG, CH4, Natural Gas, Hydrogen จึงเป็นเซนเซอร์ที่นิยมนำมาใช้ในการตรวจจับการรั่วของก๊าซต่าง ๆ เพื่อป้องกันอันตรายที่จะเกิดขึ้นจากการรั่วไหลนั้นได้ โดยเซนเซอร์ MQ-2 นี้มีจุดเด่นที่มีความไวต่อ ไอระเหยของแอลกอฮอลและควันต่ำกว่าเซนเซอร์ MQ-2 จึงสามารถนำไปติดตั้งในบริเวณที่อาจมีควันจากการปรุงอาหาร/สูบบุหรี่โดยที่ไม่ถูกรบกวนได้

- ใช้แรงดัน5V
- ให้เอาท์พุตทั้งสัญญาณอนาล็อกซึ่งเป็นค่าที่วัดได้จริง และสัญญาณดิจิตอลสามารถปรับตั้งระดับแจ้งเตือนได้ (ใช้ LM393 เป็นวงจรเปรียบเทียบแรงดัน)
- เมื่อป้อนแรงดันให้แก่เซนเซอร์ ต้องรอการอุ่นชิพอย่างน้อย 20 วินาที ก่อนทำการวัดค่า

ภาพที่ 2.10 เซนเซอร์ตรวจวัดก๊าซ MQ-5

ที่มา : https://fb1-o.lnwfile.com/ /o/ raw/tv/aq/by.jpg

MQ-9 ตรวจจับก๊าซ Carbon Monoxide(CO) และ ก๊าซที่ติดไฟได้นี้ จะตรวจจับความเข้มข้นของ CO และ ก๊าซที่ติดไฟได้ ในอากาศ และ output ที่ออกมาจะเป็นค่า analog voltage ตัวเซนเซอร์เองจะวัดค่าความเข้มข้นของ CO ตั้งแต่ 10 ถึง 10,000 ppm และก๊าซติดไฟได้ตั้งแต่ 100 ถึง 10,000 ppm ซึ่งตัวเซนเซอร์จะทำงานที่อุณหภูมิตั้งแต่ -10 ถึง 50 องศาเซลเซียส และกินไฟน้อยกว่า 150 mA 5V

ภาพที่ 2.11 เซนเซอร์ตรวจจับก๊าซ MQ-9

ที่มา : https://ae01.alicdn.com/kf/HTB1F1LGOpXXXXXwaXXXq6xXFXXX1.jpg_220x220.jpg

MQ-135 ตรวจวัดคุณภาพของอากาศ ซึ่งใช้การตรวจจับก๊าซที่หลากหลายในอากาศ ซึ่งประกอบด้วย NH3, NOx, alcohol, benzene, ควัน และ CO2 ซึ่งเหมาะกับการใช้ใน Office หรือโรงงาน ใช้วงจร drive และ วงจรตรวจสอบ ที่ง่าย

- ช่วงการตรวจจับ 10 - 300 ppm สำหรับ NH3 10 - 1,000 ppm สำหรับ Benzene และ 10 - 300 สำหรับ alcohol
- ใช้ไฟเลี้ยง 5 V
- ขนาดเส้นผ่านศูนย์กลาง 18 mm สูงรวมความสูงของ pin 17 mm และ pin สูง 6 mm

ภาพที่ 2.12 เซนเซอร์ตรวจวัดคุณภาพอากาศ MQ-135

ที่มา : https://potentiallabs.com/cart/image/cache/catalog/New%20Components-17/Mq-135-

3. เซนเซอร์วัดอุณหภูมิ/ความชื้น

อุปกรณ์เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์ (Temperature & Relative Humidity Sensor) เป็นอุปกรณ์ที่สามารถนำมาประยุกต์ใช้งานทางด้านระบบสมองกลฝังตัวได้หลากหลาย เช่น การวัดและควบคุมอุณหภูมิและความชื้น ระบบบันทึกข้อมูลเกี่ยวกับอุณหภูมิและความชื้นในห้อง เป็นต้น อุปกรณ์ประเภทนี้แตกต่างกันตามผู้ผลิต ราคา ความแม่นยำ ความละเอียดในการวัด การให้ค่าแบบดิจิทัลหรือแบบแอนะล็อก เป็นต้น

บทความนี้กล่าวถึง การทดลองใช้งานโมดูล DHT22 / AM2302 ซึ่งมีราคาถูก ให้ค่าเป็นแบบดิจิทัล ใช้ขาสัญญาณดิจิทัลเพียงเส้นเดียวในการเชื่อมต่อแบบบิตอนุกรมสองทิศทาง (serial data, bi-directional) โดยนำมาเชื่อมต่อกับ Arduino เพื่ออ่านค่าจากเซนเซอร์

ข้อมูลเชิงเทคนิค (Techinal details):

-ใช้แรงดันไฟเลี้ยงได้ในช่วง: 3.3V ถึง 5.5V DC (ดังนั้นจึงใช้ได้กับ 3.3V และ 5V)

-วัดอุณหภูมิได้ในช่วง: -40 to 80 °C (±0.5 °C accuracy)

-วัดความชื้นสัมพัทธ์ได้ในช่วง: 0 - 100 RH% (2 - 5% accuracy)

-อัตราการวัดสูงสุด: 0.5Hz

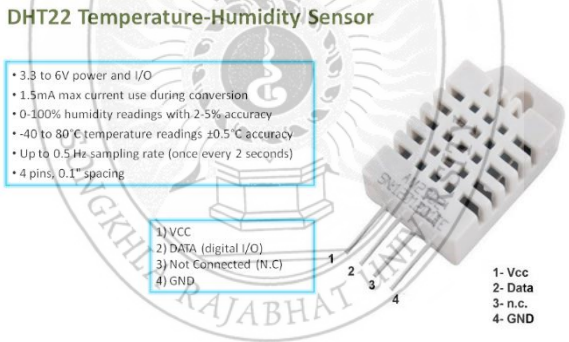
-คอนเนกเตอร์แบบ 4 ขา (0.1" / 2.54mm spacing)

Pin 1 = VCC

Pin 2 = SDA (Serial data, bidirectional)

Pin 3 = N.C. (Not Connect)

Pin 4 = GND

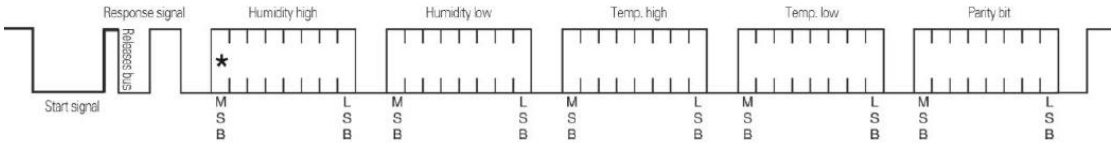


ภาพที่ 2.13 ภาพแสดงข้อมูลเชิงเทคนิค

ที่มา : http://cpre.kmutnb.ac.th/esl/learning/dht22_am2302/dht22_sensor_info-sheet.png

ในการอ่านข้อมูลจากไอซีนั้น จะใช้ขาสัญญาณเพียงเส้นเดียวคือ DATA (หรือ SDA) แบบสองทิศทาง และในสถานะปรกติสัญญาณ DATA จะเป็น HIGH ในการอ่านข้อมูลแต่ละครั้ง ไมโครคอนโทรลเลอร์จะต้องกำหนดให้ขา DATA เป็นเอาต์พุต และสร้างบิต START ซึ่งจะต้องเป็น LOW อย่างน้อย 800 μsec จากนั้นจึงให้เป็น HIGH อย่างน้อย 20 μsec หลังจากนั้นเป็นการรอการตอบกลับ (response) และจากไอซี ขา DATA จะถูกต้องเปลี่ยนเป็นอินพุต

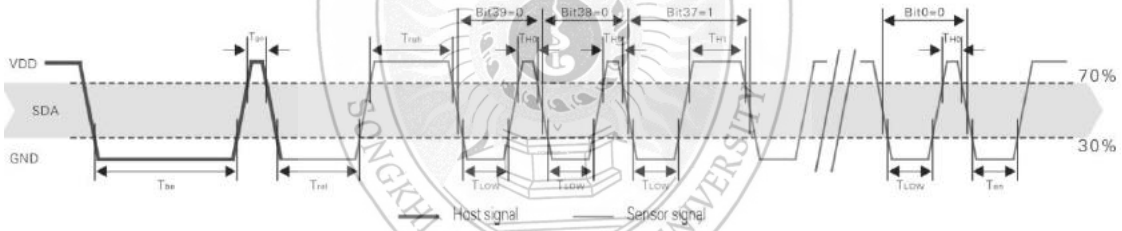
เริ่มต้นของการตอบกลับไอซี จะดึงสัญญาณลงเป็น LOW และปล่อยให้เป็น HIGH ช่วงละ 80 μsec โดยประมาณ (เรียกว่า Response Bit) จากนั้นจึงจะเป็นการส่งข้อมูลทีละบิต รวม 40 บิต (ช่วง LOW ตามด้วยช่วง HIGH) ช่วง LOW ของแต่ละบิต จะกว้างเท่ากัน แต่จะต่างกันในช่วง HIGH สำหรับบิตที่มีค่าเป็น 0 หรือ 1 (ใช้ความกว้างช่วง HIGH ในการจำแนกค่าของบิต)



ภาพที่ 2.14 รูปแสดงลำดับของข้อมูลบิต

ที่มา : http://cpre.kmutnb.ac.th/esl/learning/dht22_am2302/bit_sequence.png

ลำดับของข้อมูลบิตในการอ่านค่าจากไอซีทั้งหมด 5 ไบต์ (40 บิต) สองไบต์แรกสำหรับความชื้น สองไบต์ต่อมาสำหรับอุณหภูมิ และไบต์สุดท้ายเป็น checksum หรือ parity bits



ภาพที่ 2.15 รูปแสดงลำดับของข้อมูลบิตในการอ่านค่าจากไอซีและความกว้างของช่วง LOW และ HIGH

ที่มา : http://cpre.kmutnb.ac.th/esl/learning/dht22_am2302/bit_sequence_timing.png

ภาพที่ 2.16 เซนเซอร์ตรวจวัดอุณหภูมิ/ความชื้น DHT-22

ที่มา : https://netpie.io/public/netpieio/assets/images/projects/piesensor/dht22.jpg

งานวิจัยที่เกี่ยวข้อง

จากการที่ได้ศึกษาถึงการพัฒนาต้นแบบระบบเตือนภัยควบคุมโดยบอร์ดอาดูโน่ ผู้พัฒนาระบบได้ทำการศึกษางานวิจัยที่เกี่ยวข้องไว้หลายระบบดังต่อไปนี้

กายรัฐ เจริญราษฎร์ (2557 : 30-33) ได้ศึกษาเรื่อง “ระบบแจ้งเตือนผู้บุกรุกผ่านเครื่องแม่ข่ายโดยใช้เซนเซอร์ตรวจจับแบบไร้สาย” พบว่าปัจจุบันนี้ผู้คนในสังคมเมืองส่วนมากมักต้องออกไปทำงานนอกบ้านจึงไม่มีเวลาในการระมัดระวัง ความปลอดภัยของทรัพย์สินภายในบ้าน จึงเป็นเรื่องง่ายของกลุ่มมิจฉาชีพที่จะบุกรุกเข้ามาขโมยทรัพย์สินภายในบ้าน ผู้วิจัยจึงพัฒนาระบบกันขโมย ไร้สายโดยใช้ซิกบีขึ้นเพื่อแก้ปัญหาดังกล่าวข้างต้น โดยระบบนี้เป็นระบบที่พัฒนาขึ้นเพื่อช่วยแจ้งเจ้าของบ้านเมื่อมีโจรบุกรุกเข้ามาในบ้าน ซึ่งระบบนี้ใช้เทคโนโลยีซิกบีที่เป็นเครือข่ายเซนเซอร์ ไร้สาย และสามารถติดตั้งตามบริเวณต่าง ๆ ของบ้านได้ง่าย ดังนั้นแม้เจ้าของบ้านจะไม่อยู่ก็สามารถรับรู้ถึงสิ่งผิดปกติจากการแจ้งเตือนที่เครื่องแม่ข่าย และส่งข้อความเตือนเข้าที่สมาร์ทโฟนผ่านทางเครือข่ายอินเทอร์เน็ต ทำให้สามารถป้องกันเหตุการณ์ต่าง ๆ ได้อย่างทันท่วงที

เดโช เพ็งเหล็ง และปรีชา สมหวัง (2554 : 43) ได้ศึกษาเรื่อง “ระบบเตือนภัยอัจฉริยะสำหรับบ้านพักอาศัย” พบว่าการเกิดอาชญากรรมและโจรกรรมตามบ้านพัก ซึ่งบางครั้งเกิดจากระบบรักษาความปลอดภัยบ้านไม่ดีเพียงพอ เมื่อเกิดการโจรกรรมขึ้นแต่ละครั้งต้องใช้เวลานานในการติดตามผู้กระทำผิด อีกทั้งสิ้นเปลืองค่าใช้จ่ายจำนวนมากในการดำเนินคดีแต่ละครั้ง งานวิจัยนี้วัตถุประสงค์เพื่อสร้างระบบรักษาความปลอดภัยบ้านพักอาศัยแบบอัจริยะ โดยประยุกต์ใช้อุปกรณ์ตรวจจับวัตถุทำงานร่วมกับสัญญาณโทรศัพท์ เพื่อแจ้งเตือนการบุกรุกให้ทราบผ่านระบบเครือข่ายไปยังสถานีตำรวจ หรือปลายทางเป้าหมาย ซึ่งเป็นศูนย์รักษาความปลอดภัย ที่เก็บรายละเอียดข้อมูลโดยการสมัครสมาชิกบ้านพักอาศัยไว้ ในระบบฐานข้อมูล แล้วแสดงผลเป็นแผนที่ตั้งตำแหน่งบ้านเวลาเกิดเหตุ ผลจากการทดลองและจำลองเหตุการณ์ต่างๆทำให้ทราบว่าระบบมีประสิทธิภาพเพียงพอที่จะเป็นส่วนหนึ่งของระบบรักษาความปลอดภัยบ้านพักอาศัยสามารถป้องกันการโจรกรรมต่างๆพร้อมระบุตำแหน่งที่เกิดเหตุได้แม่นยำ

บทที่ 3

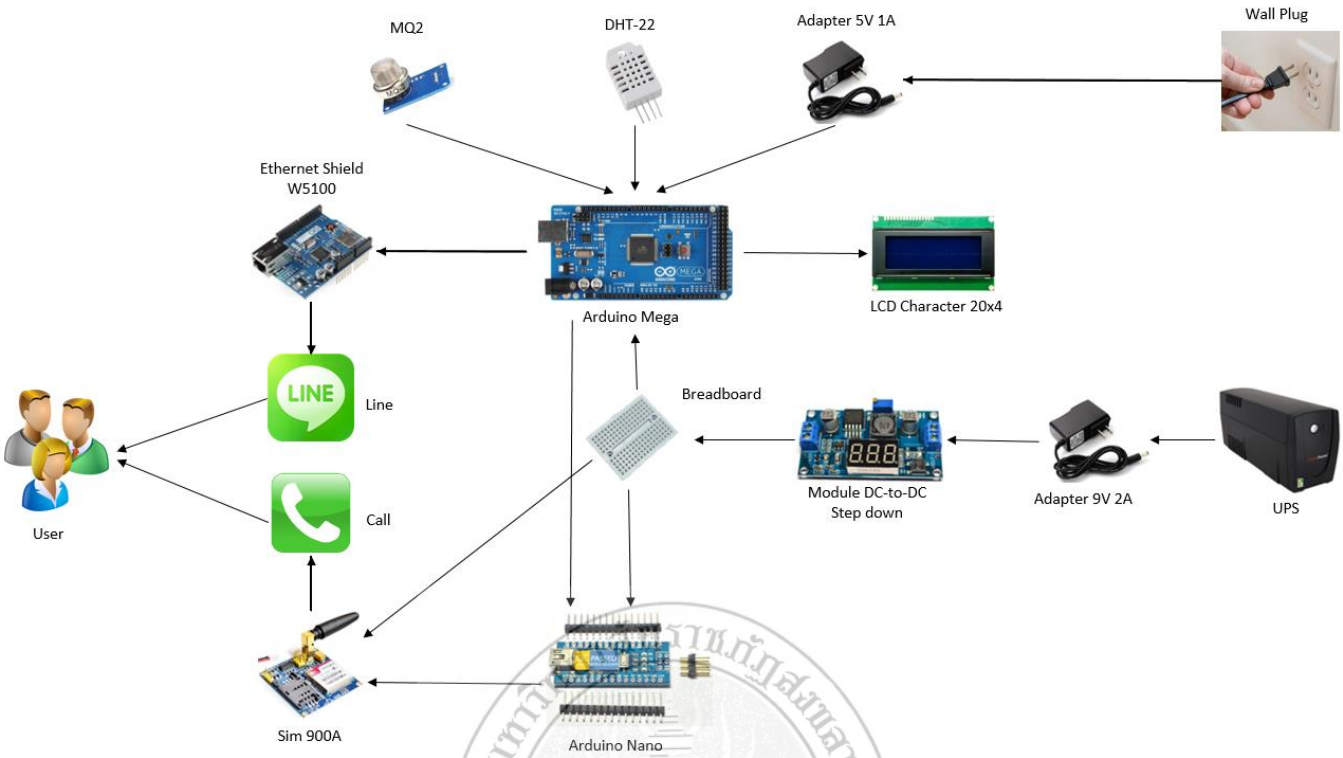
การวิเคราะห์ และออกแบบระบบ

การวิเคราะห์ระบบ

การวิเคราะห์ระบบแจ้งเตือนความปลอดภัยในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์ จะศึกษาถึงปัญหาของระบบเก่า แล้วนำมาวิเคราะห์ความต้องการและปัญหาต่าง ๆ ที่เกิดขึ้น

การออกแบบระบบ

ในการทำงานของระบบการพัฒนาต้นแบบระบบเตือนภัยควบคุมโดยบอร์ดอาดูโน่จะมีเซนเซอร์อยู่ 2 ชนิด และ อุปกรณ์สำหรับตรวจจับสถานะของไฟฟ้า 1 ชนิด ได้แก่ เซนเซอร์วัดอุณหภูมิ/ความชื้น(DHT22)สำหรับตรวจวัดอุณหภูมิและความชื้นที่เกิดขึ้นในห้องเซิร์ฟเวอร์ เซนเซอร์ตรวจวัดควัน (MQ-2) สำหรับตรวจจับควันภายในห้องเซิร์ฟเวอร์ อุปกรณ์สำหรับจ่ายไฟ (Adapter) สำหรับตรวจวัดสถานะของไฟ เมื่อเซนเซอร์เหล่านี้พบสิ่งผิดปกติภายในอาคารจะทำการส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ที่ทำหน้าที่ประมวลผลและทำการแจ้งเตือนผ่านทางแอปพลิเคชันไลน์และหากการแจ้งเตือนดังกล่าวมีความผิดพลาดก็จะทำการโทรแจ้งเตือนให้ผู้ดูแลผ่านเครือข่าย GSM ได้ทราบว่ามีสิ่งผิดปกติในห้องเซิร์ฟเวอร์



ภาพที่ 3.1 รูปแบบการทำงานของระบบ

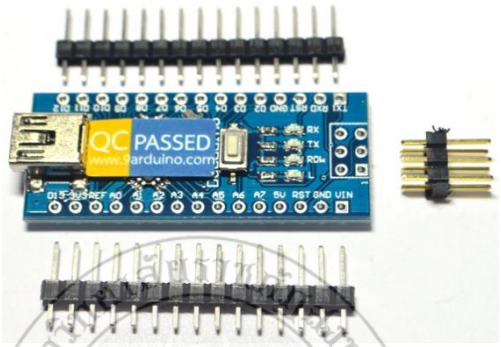
อุปกรณ์ที่ใช้ในการพัฒนา มีดังนี้

1. บอร์ด Arduino UnoMEGA 2560 R3 เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา และสามารถนำไปดัดแปลงเพิ่มเติม หรือพัฒนาต่อยอดได้อีกด้วย

ภาพที่ 3.2 บอร์ด Arduino UnoMEGA 2560 R3

ที่มา : https://cz.lnwfile.com/_/cz/_raw/79/m8/w8.jpg

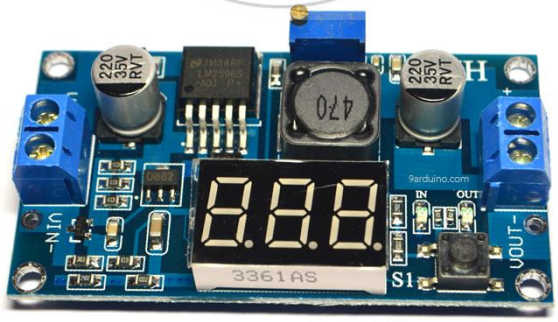
2. Arduino Nano 3.0 Mini USB ชิฟ CH340 Arduino Nano 3.0 Mini USB ชิฟ CH340G เป็น Arduino รุ่นที่ใช้ชิฟประมวล ATmega328 ตัวเดียวกันกับ Arduino Uno ความสามารถเท่ากันกับ Arduino Uno ทุกอย่างแต่ความแตกต่างของ Arduino Nano 3.0 คือ ได้ถูกผลิตออกแบบให้มีขนาดเล็ก และยังคงความเป็น Arduino ที่สาย USB กับเครื่องคอมพิวเตอร์ก็สามารถเขียนโปรแกรมได้ทันที



ภาพที่ 3.3 Arduino Nano 3.0 Mini USB ชิฟ CH340

ที่มา : https://www.9arduino.com/product/14/arduino-nano-3-0-

3. Module DC-to-DC Step down Converter LM2596 โมดูล LM2596 ออกแบบมาให้ปรับค่าโวลต์ได้แบบละเอียด ดังนั้นหมุนที่ตัวต้านทานปรับค่าได้บนบอร์ด หลาย ๆ รอบ โดยการปรับแรงดันจะทำได้โดยการใช้ไขควงหมุนที่ตัวปรับแรงดัน หากหมุนตามเข็มนาฬิกาเป็นการปรับไฟขึ้น และหมุนทวนเข็มนาฬิกาเป็นการปรับไฟลง



ภาพที่ 3.4 Module DC-to-DC Step down Converter LM2596

ทมา : https://do.lnwfile.com/_/do/_raw/be/th/f6.png

4. จอ LCD Character 20x4 จอ Liquid Crystal Display (LCD) เป็นจอแสดงผลรูปแบบหนึ่งที่นิยมนำมาใช้งานร่วมกับระบบสมองกลฝังตัวอย่างแพร่หลาย จอ LCD มีทั้งแบบแสดงผลเป็นตัวอักขระเรียกว่า Character LCD ซึ่งมีการกำหนดตัวอักษรหรืออักขระที่สามารถแสดงผลไว้ อยู่แล้ว และแบบแสดงเป็นจุดเรียกว่า Graphic LCD

ภาพที่ 3.5 จอ LCD Character 20x4

ที่มา : https://www.gravitechthai.com/upload/product/real/pic-515-9711998684.jpg

5. UPS คือ อุปกรณ์ไฟฟ้าชนิดหนึ่งที่สามารถทำการจ่ายพลังงานไฟฟ้าให้กับอุปกรณ์ไฟฟ้าและอุปกรณ์อิเล็กทรอนิคส์ได้อย่างต่อเนื่องแม้ในเวลาที่เกิดไฟดับหรือเกิดปัญหาแรงดันไฟฟ้าผันผวนผิดปกติ โดย UPS จะทำการปรับระดับแรงดันไฟฟ้าให้คงที่อยู่ในระดับที่ปลอดภัยต่ออุปกรณ์ไฟฟ้าและอุปกรณ์อิเล็กทรอนิกส์

UPS มีหน้าที่หลัก คือ ป้องกันความเสียหายที่สามารถเกิดขึ้นกับอุปกรณ์ไฟฟ้าและอุปกรณ์อิเล็กทรอนิคส์ (โดยเฉพาะคอมพิวเตอร์และอุปกรณ์เชื่อมต่อ) อันมีสาเหตุจากความผิดปกติของพลังงานไฟฟ้า เช่น ไฟตก ไฟดับ ไฟกระชากและไฟเกิน เป็นต้น รวมถึงมีหน้าที่ในการจ่ายพลังงานไฟฟ้าสำรองจากแบตเตอรี่ให้แก่อุปกรณ์ไฟฟ้าหรืออุปกรณ์อิเล็กทรอนิคส์เมื่อเกิดปัญหาทางไฟฟ้า

ภาพที่ 3. 6 UPS

ที่มา : http://www.sysnetcenter.com/1512-thickbox_default/-ups-

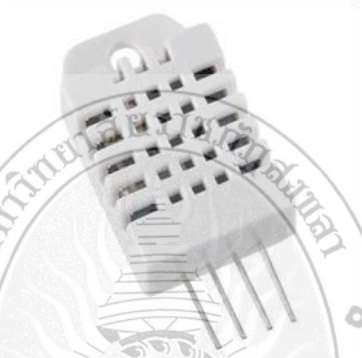
6. Adapter 9V 2A คือ หม้อแปลงไฟฟ้า จากไฟฟ้ากระแสสลับ(ไฟบ้าน)ที่มีค่าความต่างศักย์ 220 โวลต์ ให้เป็นไฟฟ้ากระแสตรง (DC) ที่มีค่าความต่างศักย์ต่ำลง เพื่อให้สามารถจ่ายกระแสไฟฟ้าให้กับเครื่องใช้ไฟฟ้าได้

ภาพที่ 3.7 Adapter 9V 2A
ที่มา : http://img.dxcdn.com/productimages/sku_126288_1.jpg

7. โมดูลตรวจจับก๊าซ MQ-2 ให้ค่าของปริมาณก๊าซในย่าน 100 ถึง 5000 ppm (pieces per million - เศษในล้านส่วน) ตามชนิดของก๊าซหรือไอระเหยที่ตรวจจับได้ ประกอบด้วย ก๊าซโพรเพน (propane) ได้ค่า 200 ถึง 5000ppm, มีเธน (methane) ได้ค่า 5000 ถึง 20000, ไฮโดรเจน (hydrogen) ได้ค่า 300 ถึง 5000,แอลกอฮอล์ (alcohol) 100 ถึง 20000 และควัน ถึงแม้ว่า ตัว MQ-2 จะตรวจจับก๊าซได้หลากหลาย แต่ถ้าสังเกตจะเห็นว่าค่าของการตรวจจับที่ได้มีช่วงที่ทับซ้อนกันอยู่ ซึ่ง MQ-2 ไม่ได้มีความสามารถในการแยกแยะชนิดของก๊าซได้ ดังนั้นจึงต้องกำหนดค่าอ้างอิงของก๊าซที่ต้องการตรวจจับ และไม่แนะนำให้ใช้ในการตรวจจับและแยกแยะชนิดของก๊าซ จากการทดสอบแนะนำให้นำไปใช้การตรวจจับควันหรือตรวจจับว่ามีก๊าซอื่น ๆ เข้ามาปะปนในพื้นที่ตรวจจับหรือไม่ จะได้ผลการทำงานที่ดีมากกว่า

ภาพที่ 3.8 โมดูลตรวจจับก๊าซ MQ-2
ที่มา : https://www.mysensors.org/uploads/57c3ebeb071cb0e34c90057a/image/mq2.png

8. DHT-22 คือ อุปกรณ์เซนเซอร์สำหรับวัดอุณหภูมิและความชื้นสัมพัทธ์ (Temperature & Relative Humidity Sensor) เป็นอุปกรณ์ที่สามารถนำมาประยุกต์ใช้งานทางด้านระบบสมองกลฝังตัวได้หลากหลาย เช่น การวัดและควบคุมอุณหภูมิและความชื้น ระบบบันทึกข้อมูลเกี่ยวกับอุณหภูมิและความชื้นในห้อง เป็นต้น และยังเป็น เซนเซอร์ที่วัดอุณหภูมิได้แบบแม่นยำ ใช้สายข้อมูลเพียง 1 เส้น สามารถใช้งานร่วมกับ Arduino ได้ รองรับแรงดันไฟฟ้าตั้งแต่ 3.3V - 6V ใช้กระแสสูงสุดเพียง 1.5mA วัดค่าความชื้นได้ตั้งแต่ 0 - 100% วัดอุณหภูมิได้ตั้งแต่ -40 ถึง 80 องศา C ค่าความผิดพลาดในการวัดความชื้นเพียง 2% และ ค่าความผิดพลาดในการวัดอุณหภูมิเพียง 0.5

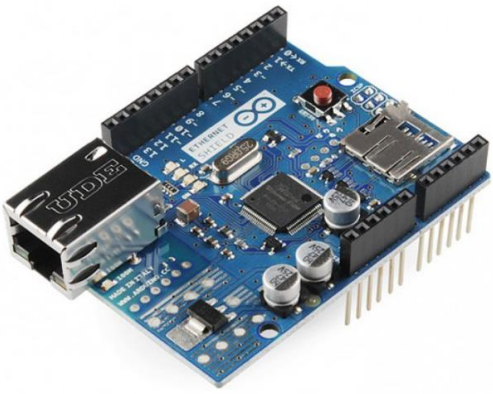


ภาพที่ 3.9 DHT-22

ที่มา : https://netpie.io/public/netpieio/assets/images/projects/piesensor/dht22.jpg

9. ETHERNET SHIELD นี้จะใช้ประกอบกับ Arduino Uno เพื่อที่จะทำให้สามารถติดต่อกับระบบเครือข่ายได้ โดยใช้ Ethernet Library ซึ่งเวอร์ชั่นล่าสุดจะมี ช่องอ่าน Micro SD Card ติดมาด้วย สามารถใช้กับ SD Library ของ ARDUINO ได้เลย

การเชื่อมต่อกับ ETHERNET SHIELD นี้จะใช้สาย RJ45 อาจจะใช้ CAT5 หรือ CAT6 โดยสามารถใช้ DHCP ได้ แต่การเชื่อมต่อระหว่างสองจุดยังต้องใช้สาย Cross Over อยู่ เพราะไม่มีวงจร Cross Over ภายใน ความเร็วในการสื่อสารของบอร์ดนี้จะอยู่ที่ 10/50 Mbps หรือ 10/100 Mbps แล้วแต่แหล่งผลิต



ภาพที่ 3.10 ETHERNET SHIELD

ที่มา:http://www.robotop.lv/66-thickbox/ethernet-shield-w5100-r3.jpg

10. บอร์ด SIM 900A Module มาพร้อมเสาอากาศใช้งานกับ micro sim โมดูล SIM900A Module สำหรับใช้ในการรับส่ง SMS โทรศัพท์หาเบอร์ที่ต้องการ เชื่อมต่ออินเตอร์เน็ตและอื่น ๆ ได้เหมือนบอร์ด GSM รุ่นใหญ่ ๆ เลย แต่ไม่สามารถคุยได้ เพราะตัดขาลำโพงกับไมค์ออก จึงเหมาะกับงานเฉพาะด้านที่ต้องการความคุ้มค่า การใช้งานไลบารีเหมือน SIM900A มีไลบารีมาให้พร้อมใช้งาน กระแสสูงสุด 2A ถ้าบอร์ด Arduino ที่ใช้ไฟจาก usb กระแสไฟอาจจะไม่พอ ต้องต่อไฟเพิ่ม ทำงานทันทีที่จ่ายไฟ มี LED แสดงผลสัญญาณ ถ้าจับสัญญาณโทรศัพท์ได้จะกระพริบช้า ๆ แต่ถ้าจับไม่ได้จะกระพริบถี่ ๆ บอร์ด SIM900A Module สามารถต่อกับ Battery Li-ion ได้โดยตรง สามารถต่อไฟ 5V ได้โดยตรง

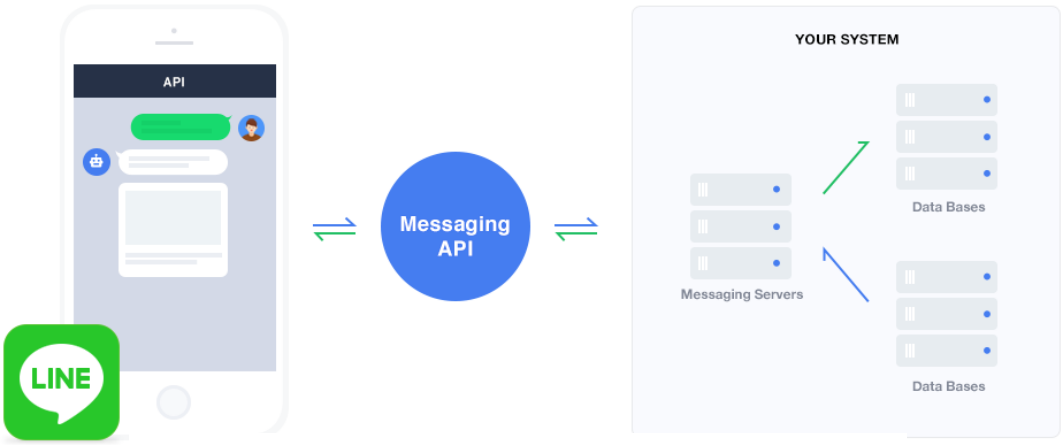
ภาพที่ 3.11 บอร์ด SIM 900A Module

ที่มา : http://bdspeedytech.com/image/cache/catalog/SIM800L-750x750.png

11. LINE คือแอพพลิเคชั่นที่ผสมผสานบริการ Messaging และ Voice Over IP นำมาผนวกเข้าด้วยกัน จึงทำให้เกิดเป็นแอปพลิชันที่สามารถแชท สร้างกลุ่ม ส่งข้อความ โพสต์รูปต่าง ๆ หรือจะโทรคุยกันแบบเสียงก็ได้ โดยข้อมูลทั้งหมดไม่ต้องเสียเงิน หากเราใช้งานโทรศัพท์ที่มีแพคเกจอินเทอร์เน็ตอยู่แล้ว แถมยังสามารถใช้งานร่วมกันระหว่าง iOS และ Android รวมทั้งระบบปฏิบัติการอื่น ๆ ได้อีกด้วย

ภาพที่ 3.12 LINE Application

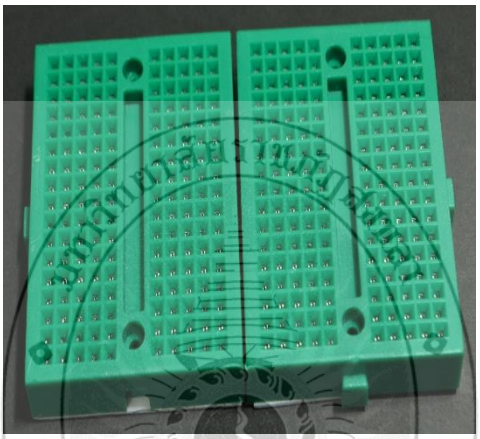
12. Line API หรือ Messaging API คือ การสื่อสารระหว่างบริการและผู้ใช้ LINE เป็นการสื่อสารแบบสองฝ่าย โดย จะส่งและรับข้อมูลระหว่างเซิร์ฟเวอร์ของคุณและแอปพลิเคชัน LINE ผ่านทางเซิร์ฟเวอร์ของ LINE สามารถส่งและรับข้อความกับผู้ใช้ที่เป็นเพื่อนกับบัญชีนั้น ๆ และสามารถเพิ่มเป็นสมาชิกในกลุ่มเพื่อส่งข้อความได้เช่นกัน



ภาพที่ 3.13 . Line API

ที่มา : https://scdn.line-apps.com/n/_5/partner-center/img/lp/msgapi-

13. Breadboard 170 holes เป็นบอร์ดที่ใช้ทดลองวงจรอิเล็กทรอนิกส์ ลักษณะเป็นแผ่นพลาสติกหนาสีขาวหรือสีเขียว บนแผ่นมีรูเรียงกันจำนวนมาก ภายในรูมีตัวนำไฟฟ้าซึ่งเชื่อมต่อกันในรูปแบบที่มีการกำหนดไว้ เวลาทดลองก็เสียบขาของอุปกรณ์อิเล็กทรอนิกส์ลงไปให้ตัวนำภายในเชื่อมวงจรถึงกัน และอาจใช้สายไฟเสียบลงรูเพื่อเชื่อมวงจรไฟฟ้าได้เช่นกัน ข้อดีของเบรดบอร์ดคือไม่ต้องออกแบบแผงวงจรและไม่ต้องบัดกรี แต่มีข้อเสียคือใช้ทดลองวงจรที่ทำงานที่ความถี่สูง ๆ ไม่ได้เนื่องมีปัญหาเรื่องสัญญาณรบกวนในวงจร



ภาพที่ 3. 14 Breadboard 170 holes

ที่มา : https://www.9arduino.com/product/61/breadboard-170-holes

14. Power connector 3.5 mm หัวแจ็คสำหรับ ติดตั้งลงกล่องอเนกประสงค์ เพื่อเชื่อมต่อกับ Adapter ต่าง ๆ สามารถเจาะสว่านแล้วยึดกับ กล่องได้ทันที

ภาพที่ 3. 15 Power connector 3.5 mm

ที่มา : https://www.9arduino.com/product/168/power-connector-3-5-mm-

15. สายแพ Jumper เป็นสายแพที่สามารถฉีกออกได้ 40 เส้น มีความยาว 20 ซม. เป็นสายที่ใช้ต่อร่วมกับ บอร์ด Arduino Sensor module ต่าง ๆ เพื่อให้ใช้งานได้ง่าย

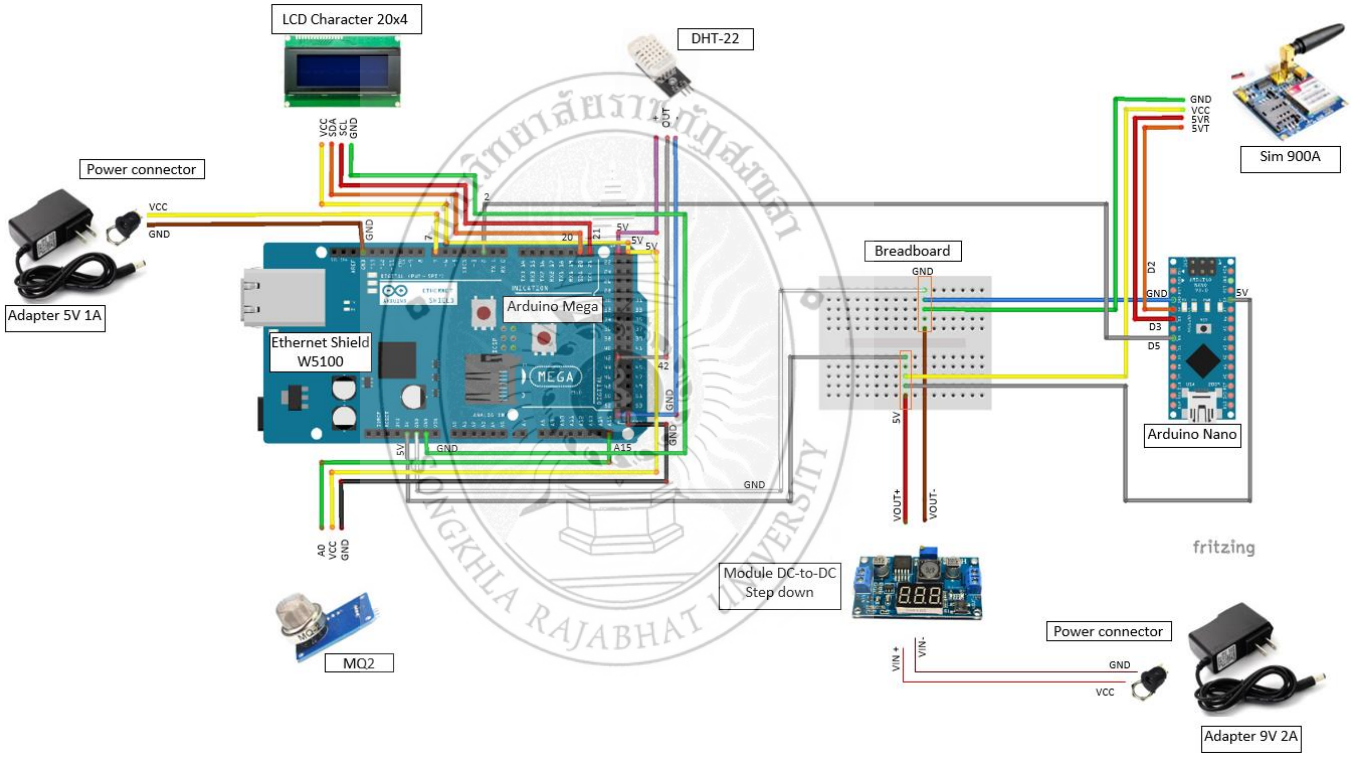
ภาพที่ 3. 16 สายแพ Jumper

ที่มา : https://www.9arduino.com/product/

มหาวิทยาลัยราชภัฏสงขลา SONGKHLA RAJABHAT UNIVERSITY

การออกแบบวงจร

วงจรของการพัฒนาระบบแจ้งเตือนความปลอดภัยในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์ โดยใช้บอร์ดอาดูโน่ทำหน้าที่เป็นหน่วยประมวลผล และมีเซนเซอร์ต่าง ๆ ที่เป็นตัววัดค่าเพื่อจะส่งการแจ้งเตือนไปยังแอปพลิเคชั่น ไลน์ให้แก่ผู้ใช้ได้แก่ เซนเซอร์ DHT–22 ใช้สำหรับวัดความชื้นและอุณหภูมิ MQ-2 ใช้สำหรับวัดค่าของควัน และ Adapter เพื่อใช้ตรวจเช็คสถานะของไฟฟ้า โดยมีการออกแบบวงจรภาพต่อไปนี้



ภาพที่ 3.17 รูปแบบการต่อวงจรของระบบ

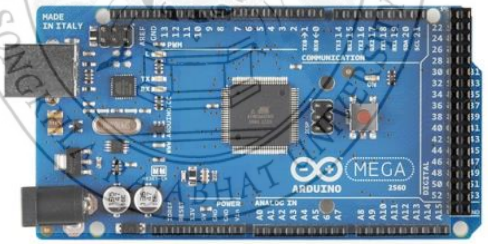
บทที่ 4

การพัฒนาระบบ และการทดสอบ

การพัฒนาระบบ

การพัฒนาระบบแจ้งเตือนความปลอดภัยในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์ จะทำงานเมื่อมีสิ่งผิดปกติเกิดขึ้นภายในห้องเซิร์ฟเวอร์ ระบบนี้จะประกอบไปด้วยเซนเซอร์อยู่ 2 ชนิด ได้แก่ เซนเซอร์ตรวจจับควันไฟ (MQ-2) สำหรับตรวจจับควันไฟภายในห้องเซิร์ฟเวอร์ และเซนเซอร์วัดอุณหภูมิและความชื้น (DHT-22) สำหรับการตรวจวัดความชื้นและอุณหภูมิภายในห้องเซิร์ฟเวอร์ และ Adapter (5V 1A) 1 ตัว เพื่อตรวจเช็คสถานะของไฟในห้องเซิร์ฟเวอร์ เมื่อเซนเซอร์หรืออุปกรณ์เหล่านี้ตรวจจับสิ่งผิดปกติภายในห้องเซิร์ฟเวอร์ได้จะทำการส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ที่ทำหน้าที่ประมวลผลและสั่งการ เพื่อแจ้งเตือนผ่านทางแอปพลิเคชันไลน์ หรือเมื่อเกิดข้อผิดพลาดในการส่งระบบจะทำการโทรไปยังผู้ใช้แทน

การควบคุมการทำงานของ บอร์ด Arduino

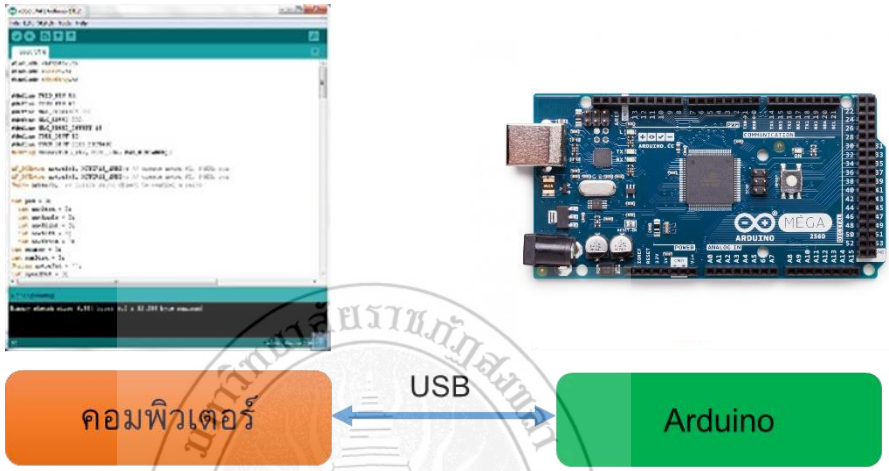


ภาพที่ 4.1 บอร์ด Arduino MEGA 2560
ที่มา : https://cz.lnwfile.com/_/cz/_raw/79/m8/w8.jpg

Arduino เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัว บอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย ผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิคส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ดหรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม ประเภทต่าง ๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless

Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาผ่าน Arduino IDE

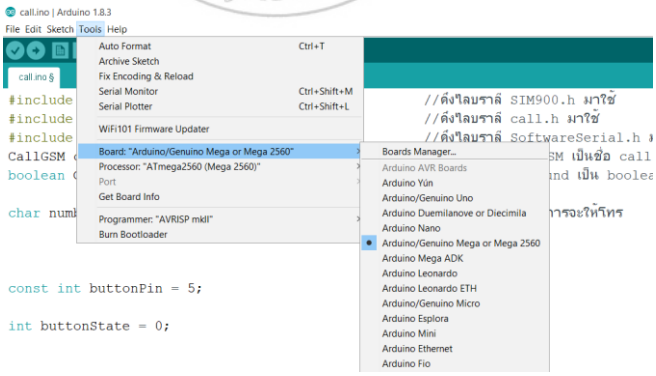
รูปแบบการเขียนโปรแกรมบน Arduino



ภาพที่ 4.2 รูปแบบการเขียนโปรแกรมบน Arduino

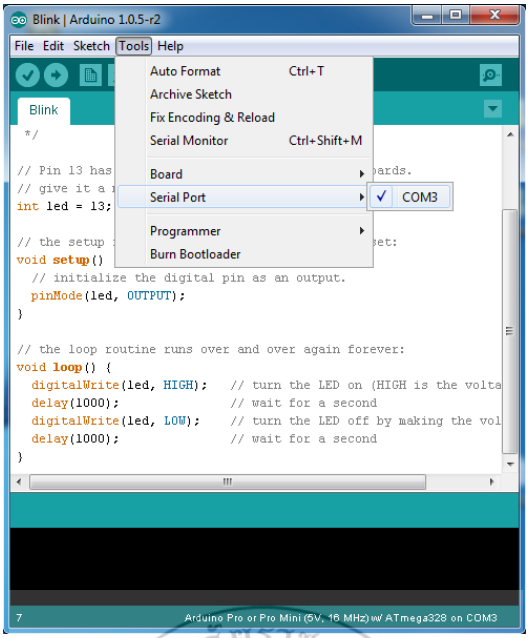
ที่มา : http://thaieasyelec.com/images/basic-electronics/interview-arduino-

เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม ArduinoIDE ซึ่งสามารถดาวน์โหลดได้จาก Arduino.cc/en/main/software หลังจากที่เขียนโค้ดโปรแกรมเรียบร้อยแล้ว ให้ผู้ใช้งานเลือกรุ่นของบอร์ด Arduino ที่ใช้ในที่นี้เลือกรุ่น Arduino Mega และเลือกหมายเลข Serial port



ภาพที่ 4.3 เลือกรุ่นบอร์ด Arduino ที่ต้องการ upload

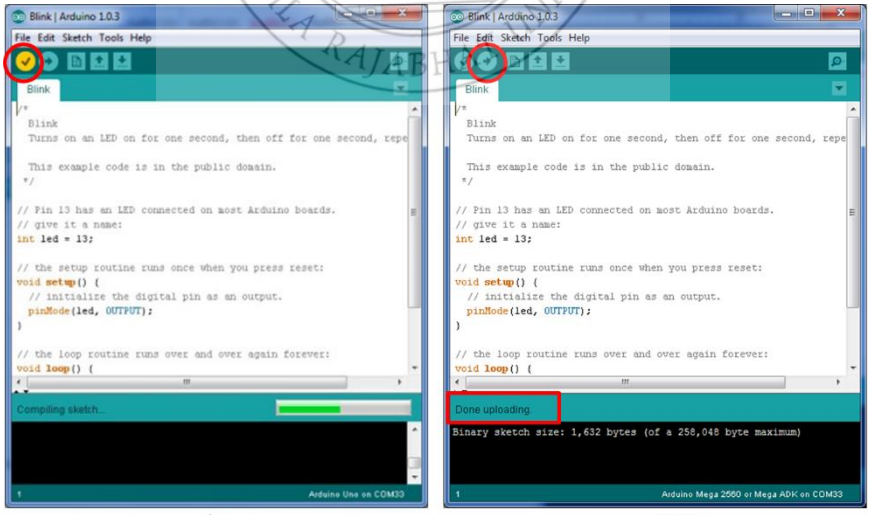
ที่มา : http://cu.lnwfile.com/_/cu/_raw/7z/vp/ss.pngpart1/3.png



ภาพที่ 4.4 Serial port ของคอมพิวเตอร์

ที่มา : http://cu.lnwfile.com/_/cu/_raw/hz/1w/ti.png

เขียนชุดคำสั่งจากนั้นกดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม จากนั้นกดปุ่ม Upload โค้ด โปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่ออับโหลดเรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที



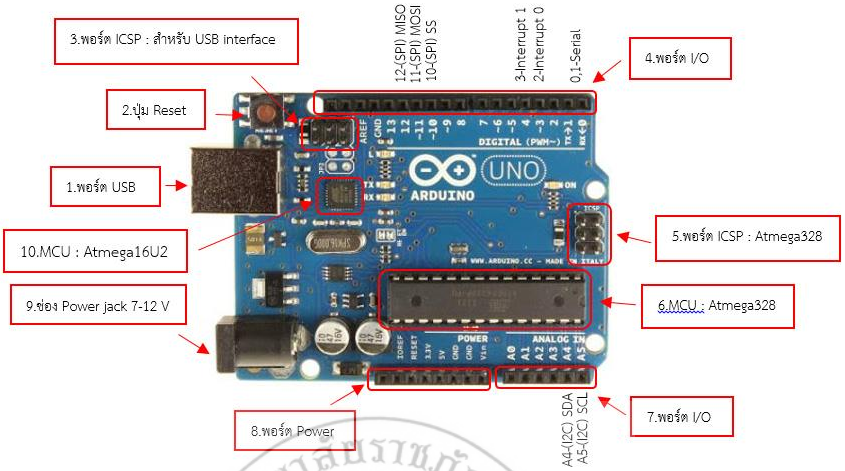
กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม

Upload โค้ดโปรแกรม

ภาพที่ 4.5 Upload โค้ด โปรแกรมไปยังบอร์ด Arduino

ที่มา : http://cu.lnwfile.com/_/cu/_raw/og/7e/l7.png

รูปแบบ Layout ของ Board Arduino (Model : Arduino UNO R3)



ภาพที่ 4.6 Layout ของ Board Arduino

1. USB Port: ใช้สำหรับต่อกับ Computer เพื่ออับโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด

2. Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่

3. ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2

4. I/O Port: Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx , Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM

5. ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader

6. MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino

7. I/O Port: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5

8. Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, V in

9. Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V

10. MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

การควบคุมการทำงานของจอ Character LCD

จอ LCD 20 ตัวอักษร จำนวน 4 บรรทัด แสดงผลตัวอักษรสีขาวบนพื้น (Background) สีน้ำเงิน ใช้งานง่าย ด้วย interface แบบ I2C โดยใช้สายแค่ 2 เส้นเท่านั้นในการรับส่งข้อมูลเพื่อแสดงตัวอักษรบนจอ LCD ทำให้ประหยัด port ที่ต่อใช้งานบนบอร์ดไมโครคอนโทลเลอร์ และยังสามารถปรับ contrast เพื่อเพิ่มความคมชัดบนหน้าจอ LCD ได้อีกด้วย

ภาพที่ 4.7 จอ LCD Character 20x4

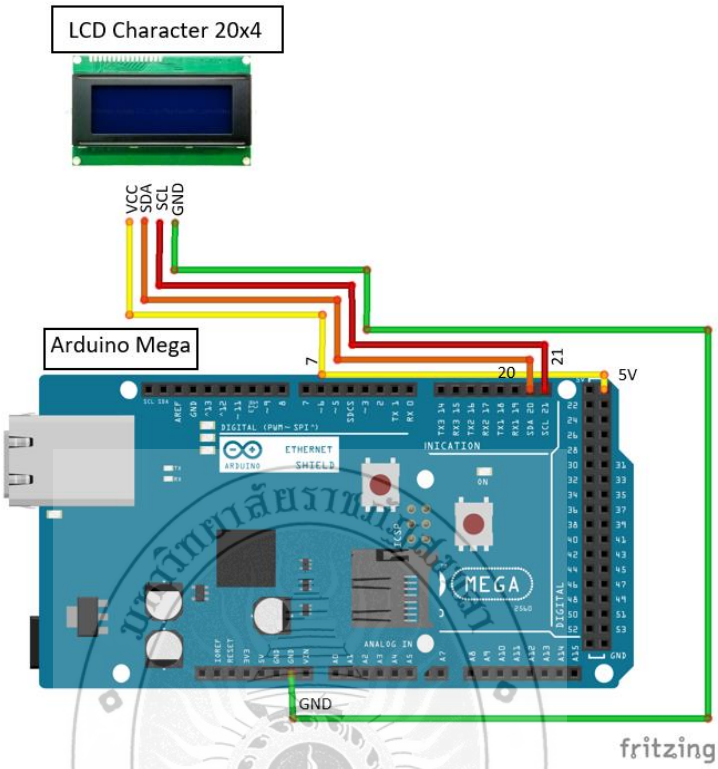
ที่มา : https://www.gravitechthai.com/upload/product/real/pic-515-9711998684.jpg

การควบคุมการแสดงผลของจอ LCD

ในการควบคุมหรือสั่งงาน โดยทั่วไปจอ LCD จะมีส่วนควบคุม (Controller) อยู่ในตัวแล้ว ผู้ใช้สามารถส่งรหัสคำสั่งสำหรับควบคุมการทำงานของจอ LCD กับ Controller ว่าต้องการใช้แสดงผลอย่างไร โดย LCD Controller ขาในการเชื่อมต่อระหว่าง LCD กับ Microcontroller มีดังนี้

1. GND เป็น Ground ใช้ต่อระหว่าง Ground ของระบบ Microcontroller กับ LCD
2. VCC เป็นไฟเลี้ยงวงจรที่ป้อนให้กับ LCD มีขนาด +5VDC
3. SDA (Serial Data) เป็นขาที่ใช้ในการรับส่งข้อมูล
4. SCL (Serial Clock) เป็นขาสัญญาณนาฬิกาในการรับส่งข้อมูล

การต่อใช้งานจอ LCD Character 20x4 กับ Arduino Mega



ภาพที่ 4.8 การต่อวงจร LCD

วิธีการต่อขา ใช้งาน

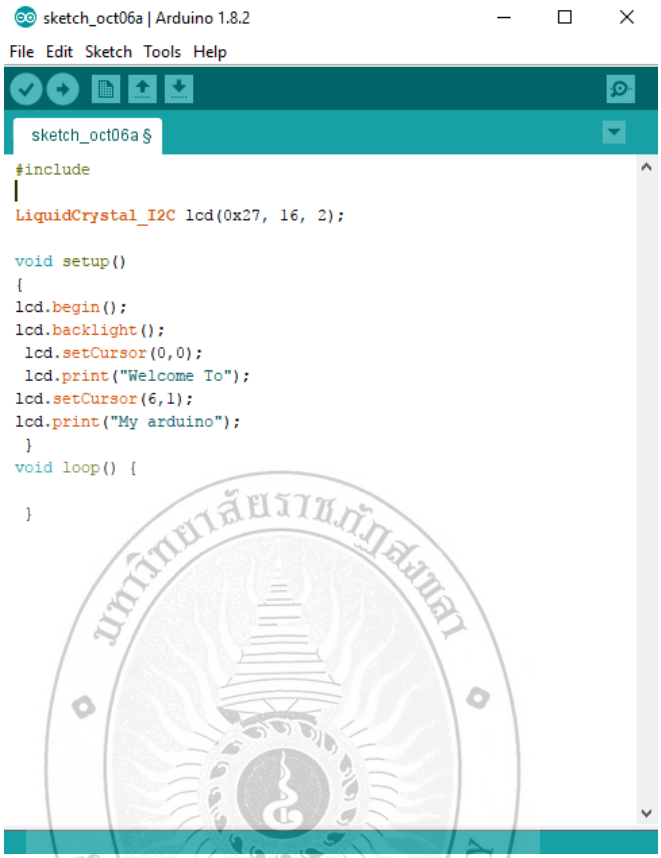
VCC ต่อเข้ากับบอร์ดขา 5V

GND ต่อเข้ากับบอร์ดขา GND

SCL ต่อเข้ากับบอร์ดขา 21

SDA ต่อเข้ากับบอร์ดขา 20

ตัวอย่างโค้ดสำหรับแสดงข้อความบนจอ LCD



ภาพที่ 4.9 ตัวอย่างโค้ดแสดงข้อความบน LCD

มหาวิทยาลัยราชภัฏสงขลา SONGKHLA RAJABHAT UNIVERSITY

การควบคุมการทำงานของเซนเซอร์ตรวจจับควัน DHT-22

DHT-22 (Digital Temperature and Humidity Sensor) เป็นโมดูลวัดอุณหภูมิและความชื้นสัมพัทธ์ โดยใช้ชิพ DHT-22 ให้ Output ออกมาเป็นแบบ Digital Output ใช้ไฟ DC ขนาด 3.3 – 6 โวลต์

ภาพที่ 4.10 เซนเซอร์ DHT-22

ที่มา : https://netpie.io/public/netpieio/assets/images/projects/piesensor/dht22.jpg

คุณสมบัติของ DHT-22

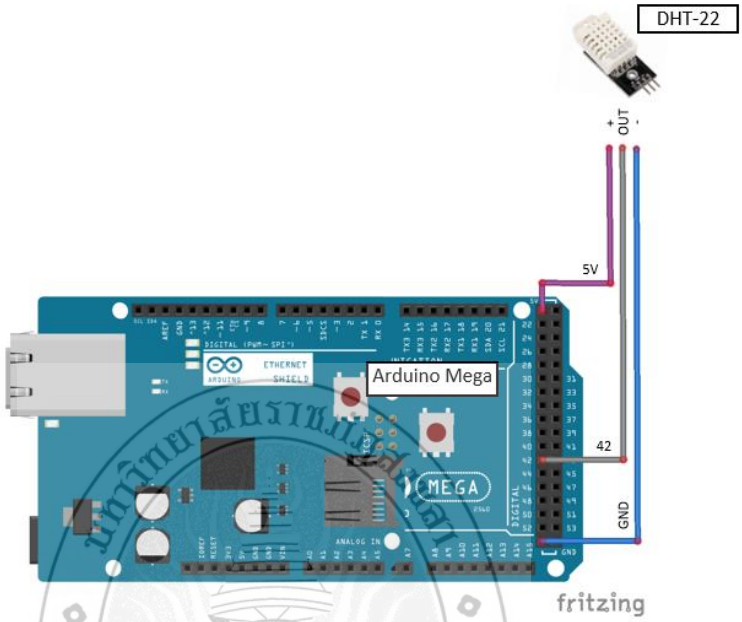
ไฟเลี้ยงโมดูล VCC = 3.3-6V DC

สามารถตรวจสอบค่าความชื้นที่ 0-100 RH

สามารถตรวจสอบค่าอุณหภูมิช่วง -40 C ถึง 80 C

ให้สัญญาณเอาท์พุทออกมาเป็นแบบ Digital Output คือ 0 กับ 1

การต่อใช้งาน เซนเซอร์ตรวจจับควัน DHT-22 กับ Arduino Mega



ภาพที่ 4.11 การต่อวงจร DHT-22

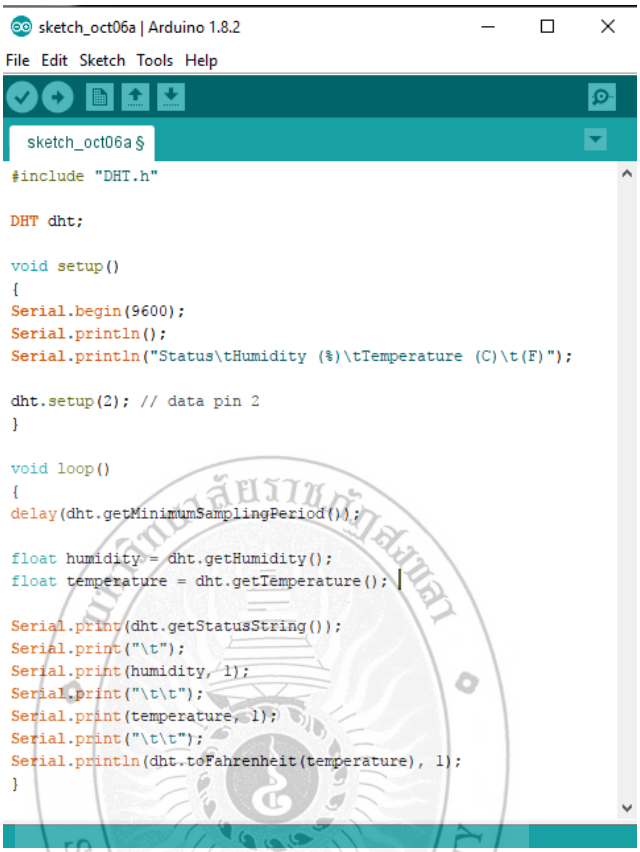
วิธีการต่อขา ใช้งาน

+ ต่อเข้ากับบอร์ดขา 5V

- ต่อเข้ากับบอร์ดขา GND

OUT ต่อเข้ากับบอร์ดขา 42

ตัวอย่างโค้ดวัดค่าอุณหภูมิและความชื้นสำหรับเซนเซอร์ DHT22



ภาพที่ 4.12 ตัวอย่างโค้ดวัดค่าอุณหภูมิและความชื้นสำหรับเซนเซอร์ DHT-22

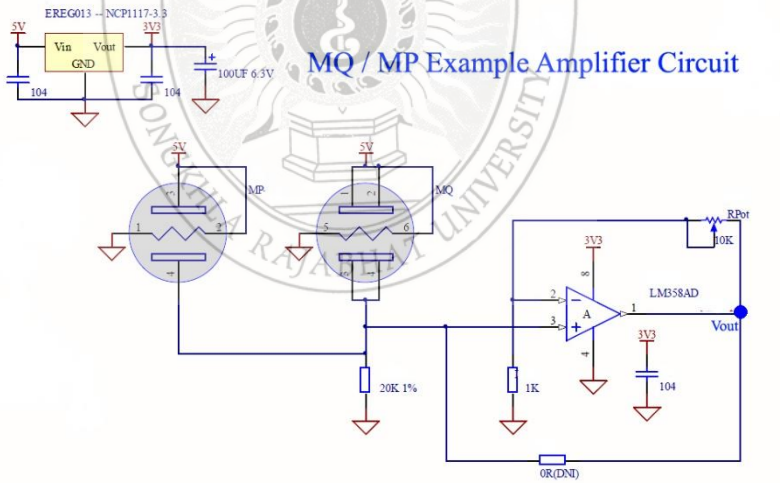
การควบคุมการทำงานของเซนเซอร์ตรวจจับควัน MQ-2

MQ-2 เป็น Sensor ตรวจสอบปริมาณ ก๊าซไวไฟ และ ควัน เช่น LPG, i-butane, propane, methane , alcohol, Hydrogen, smokeในอากาศ ซึ่งเมื่อเราเริ่มจ่ายพลังงานให้ MQ-2 ที่ขา H ทำให้เกิดความร้อนขึ้นที่ขดลวด เมื่อก๊าซไวไฟต่างๆ เข้ามาทำปฏิกิริยาจะทำให้ ค่าความต้านทานที่เกิดขึ้นระหว่าง ขา A และ B (RS)ลดลง

ภาพที่ 4.13 โมดูลตรวจจับก๊าซ MQ-2

ที่มา : https://www.mysensors.org/uploads/57c3ebeb071cb0e34c90057a/image/mq2.png

วงจรการทำงานของโมดูล MQ-2



ภาพที่ 4.14 วงจรการทำงานของ MQ-2

ที่มา : http://www.thaieasyelec.com/downloads/ESEN087/MP-MQ-Example-Amp-Circuit.jpg

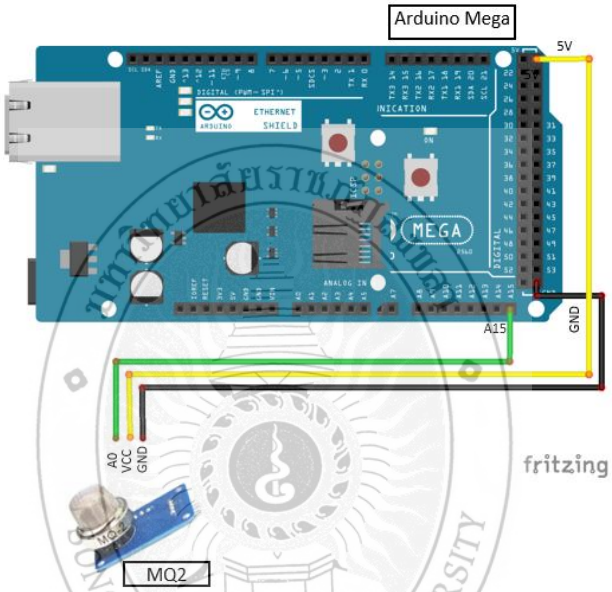
คุณสมบัติของ MQ-2

ไฟเลี้ยงโมดูล 5 VDC @ ~160mA

รองรับการทำงานที่ค่าอุณหภูมิช่วง -4 to +122 °F (-20 to +50 °C)

ให้สัญญาณเอาท์พุทออกมาเป็นแบบ Analog

การต่อใช้งาน เซนเซอร์ตรวจจับควัน DHT-22 กับ Arduino Mega



ภาพที่ 4.15 การต่อวงจร MQ-2

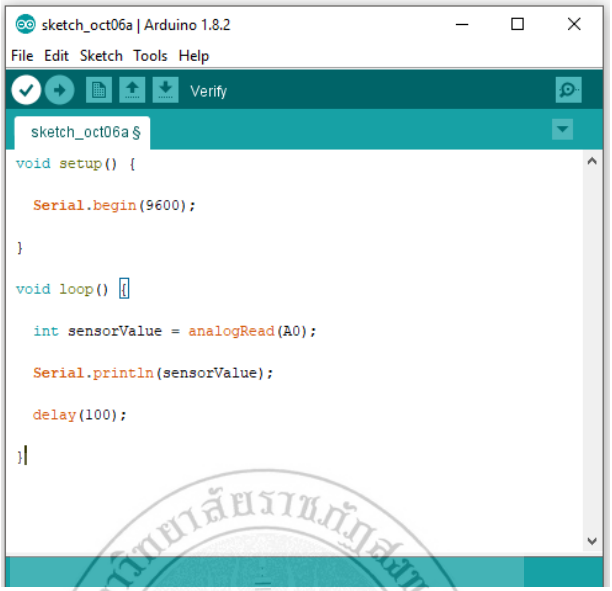
วิธีการต่อขา ใช้งาน

VCC ต่อเข้ากับบอร์ดขา 5V

GND ต่อเข้ากับบอร์ดขา GND

A0 ต่อเข้ากับบอร์ดขา A15

ตัวอย่างโค้ดวัดค่าควันสำหรับเซนเซอร์ MQ-2



ภาพที่ 4.16 ตัวอย่างโค้ดวัดค่าควันสำหรับเซนเซอร์ MQ-2

การควบคุมการทำงานของ Module SIM 900A

Module SIM 900A สามารถโทรเข้า/โทรออก รับสาย/วางสาย ส่ง SMS เชื่อมต่ออินเตอร์เน็ต หรือทำงานต่าง ๆ ได้เหมือนโทรศัพท์

ภาพที่ 4.17 บอร์ด SIM 900A Module

ที่มา : http://bdspeedytech.com/image/cache/catalog/SIM800L-750x750.png

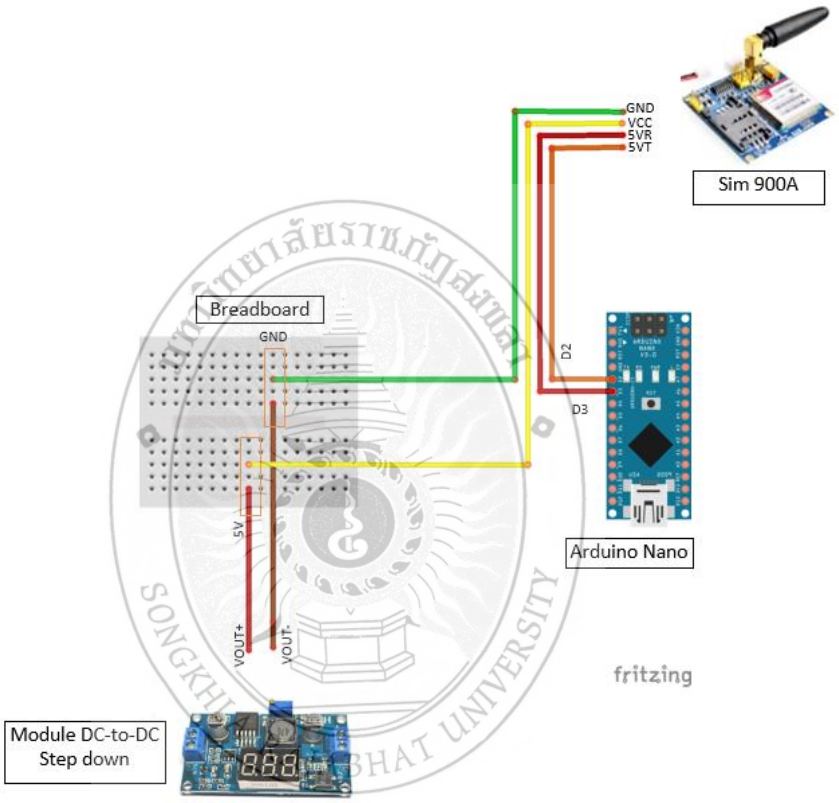
คุณสมบัติของ Module SIM 900A

สามารถโทรเข้า/โทรออก รับสาย/วางสาย

สามารถส่ง SMS

ไฟเลี้ยงโมดูล 4.1-4.5 V (ใช้ไฟเลี้ยงจาก USB ไม่ได้)

การต่อใช้งาน Module SIM 900A กับ Arduino



ภาพที่ 4.18 การต่อวงจร Module SIM 900A

วิธีการต่อขา ใช้งาน

การเชื่อมต่อ Module Sim 900A จำเป็นต้องใช้ไฟแยกในการจ่าย โดยจะใช้ Breadboard เป็นแหล่งพักไฟที่ได้มาจากการแปลงกระแสไฟของ Module Step down เนื่องจาก Module Sim 900A ใช้กระแสไฟมาก เสียบกับบอร์ด Arduino โดยตรงบอร์ดจะไม่ทำงาน วิธีการต่อมีดังต่อไปนี้

VCC ต่อเข้ากับ Breadboard แถว 5V

GND ต่อเข้ากับแถว Breadboard GND

5VT ต่อเข้ากับบอร์ดขา D2

5VR ต่อเข้ากับบอร์ดขา D3

VIN+ ต่อเข้ากับ Breadboard แถว 5V

VIN- ต่อเข้ากับแถว Breadboard GND

ตัวอย่างโค้ดการโทรออกสำหรับ Module Sim 900A

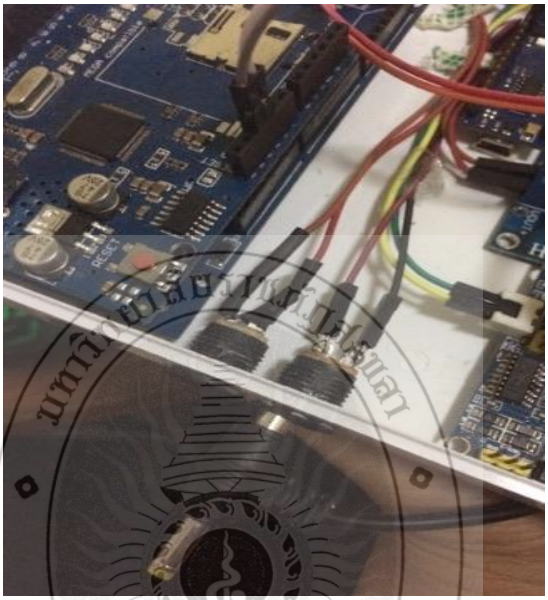
call.ino §

```
#include "SIM900.h"
#include "call.h"
#include <SoftwareSerial.h>
CallGSM call;
boolean One_Round = false;
char number[] ="+668242819O8";
const int buttonPin = 5;
int buttonState = 0;

int CDCall = 0;
int TimerCall = 3;
void setup()

{
  Serial.begin(9600);
     if (gsm.begin(9600))
          Serial.println("\nstatus=READY");
     else Serial.println("\nstatus=IDLE");
       pinMode(buttonPin, INPUT);
}

void loop()
{
    buttonState = digitalRead(buttonPin);
    Serial.print("digitalRead...");

  if (buttonState == HIGH) {

     if (CDCall <= 0) {
      Serial.println("Calling...");
      call.Call(number);
    } else {
      TimerCall = CDCall - 1;
    }
  }

  else{
    Serial.println("Is LOW DON'T CALL");
    TimerCall = 0;
    }
    CDCall = TimerCall;
}
```

ภาพที่ 4.19 ตัวอย่างโค้ดการโทรออกสำหรับ Module Sim 900A

การควบคุมการทำงานของ Adapter เช็คสถานะไฟฟ้า

ในการเช็คค่าสถานะไฟนั้นจะใช้ Adapter ขนาด 5V 1A เสียบที่ Wall Plug เพื่อเช็คสถานะของไฟฟ้าว่ามีสถานะติดหรือดับและมาต่อเข้ากับบอร์ดโดยใช้การบัดกรีเข้ากับตัวแปลง Power Connecter เพื่อที่จะสามารถแยกสายออกเป็น VCC และ GND

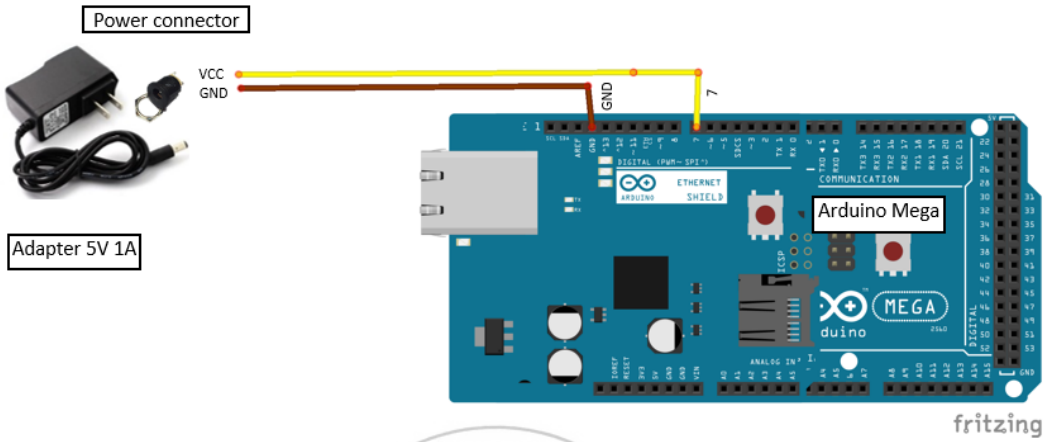


ภาพที่ 4.20 การเชื่อมต่อ Adapter กับ Power connecter

การทำงานของ Adapter

บอร์ดจะทำการเช็คค่าสถานะต่างๆของไฟผ่านกระแสไฟที่ไหลเข้ามาในบอร์ดโดยจะรับค่าเป็นแบบ Digital หรือ 0 กับ 1 เช่น สถานะมีไฟแทนด้วย 1 เมื่อเกิดเหตุการณ์ไฟดับกระแสไฟก็จะเปลี่ยนเป็น 0

การต่อใช้งาน Adapter เพื่อวัดสถานะไฟ กับ Arduino Mega



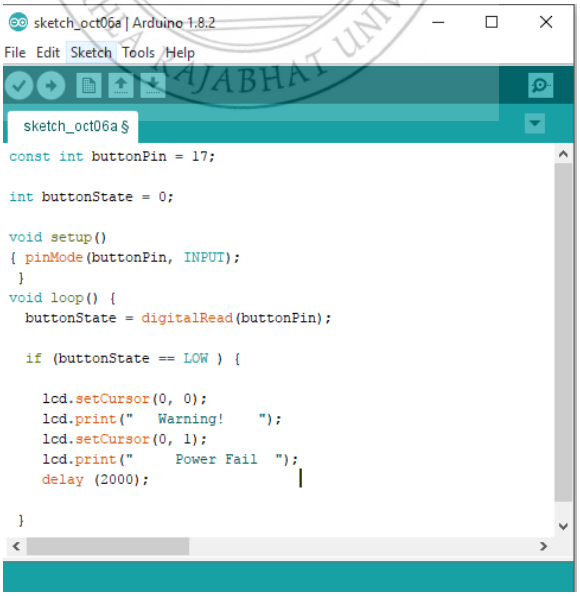
ภาพที่ 4.21 การต่อวงจรเข้ากับ Adapter

วิธีการต่อขา ใช้งาน

VCC ต่อเข้ากับบอร์ดขา 7

GND ต่อเข้ากับบอร์ดขา GND

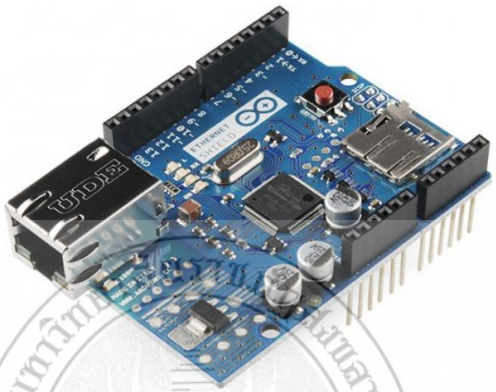
ตัวอย่างโค้ดสำหรับการเช็คสถานะไฟ



ภาพที่ 4.22 ตัวอย่างโค้ดสำหรับการเช็คสถานะไฟ

การควบคุมการทำงานของ Ethernet shield – W5100

Ethernet shield สามารถทำให้ Arduino ติดต่อกับเครือข่าย หรือเชื่อมต่อกับอินเตอร์เน็ตได้ เพื่อที่จะได้สะดวกในการควบคุมและติดตามอุปกรณ์ต่าง ๆ เช่น การดึงค่าเซนเซอร์ต่าง ๆ ที่ติดตั้งไว้ มาดูบนโทรศัพท์มือถือหรือการเปิดปิดเครื่องใช้ไฟฟ้าผ่านทางอินเตอร์เน็ต



ภาพที่ 4.23 Ethernet shield – W5100

ที่มา : http://www.robotop.lv/66-thickbox/ethernet-shield-w5100-r3.jpg

คุณสมบัติของ Ethernet shield – W5100

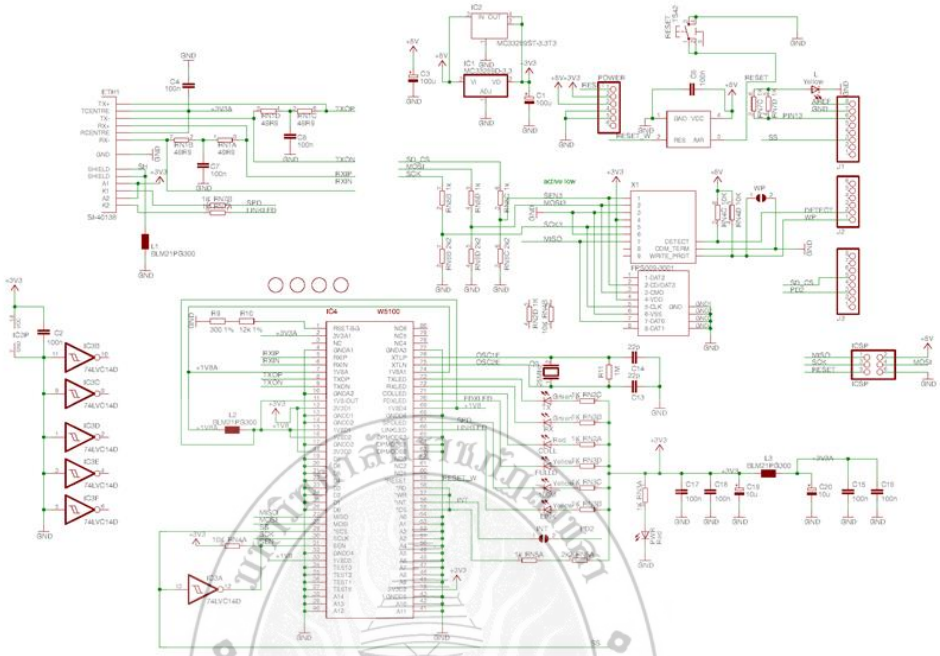
ไฟเลี้ยงโมดูล VCC = 5V รองรับจากไฟบอร์ดของอาดูโน่โดยตรง

ความเร็วการเชื่อมต่อ 10/100Mb

ทำงานผ่านขา SPI port

มีช่องสำหรับเพิ่ม SD Card

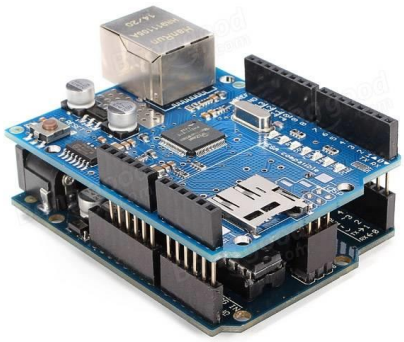
วงจรการทำงาน Ethernet shield – W5100



ภาพที่ 4.24 วงจรของ Ethernet shiled W5100

ที่มา : http://eezone.co.uk/wp-

การต่อใช้งาน การเชื่อมต่อ Ethernet shield กับ Arduino



ภาพที่ 4.25 การเชื่อมต่อ Ethernet shield กับ Arduino

ที่มา : https://www.google.co.th/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja

วิธีการต่อขา ใช้งาน

เสียบ Ethernet Shield W5100 ลงบนบอร์ด Arduino จากนั้นเสียบสายแลนที่มีอินเตอร์เน็ตเข้ากับตัวบอร์ด Ethernet Shield W5100

ตัวอย่างโค้ดการเชื่อมต่ออินเทอร์เน็ตสำหรับ Ethernet shield – W5100

ภาพที่ 4.26 ตัวอย่างโค้ดการเชื่อมต่ออินเทอร์เน็ตสำหรับ Ethernet shield – W5100

การทำงานของ Line API

LINE Notify เป็นบริการของทาง LINE สามารถส่งความ การแจ้งเตือนต่าง ๆ ไปยังแอ๊คเค้าเองได้ ผ่านการใช้ API ซึ่งเรียกผ่าน HTTP POST แต่สามารถส่งการแจ้งเตือนได้เฉพาะผู้ที่ขอใช้หรือกลุ่มที่ผู้ขอใช้เป็นสมาชิกเท่านั้น ไม่สามารถส่งข้อความเข้าห้องสนทนา

ภาพที่ 4.27 Line Notify

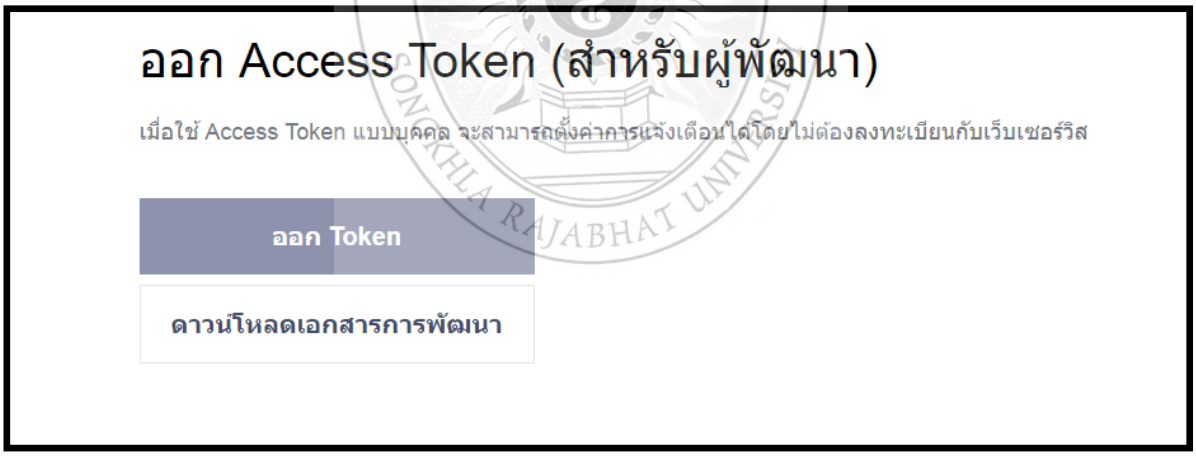
ที่มา : notify-bot.line.me

การใช้งาน API ในทุก ๆ บริการ จะมีสิ่งที่เรียกว่า Access Token ไว้สำหรับเป็นรหัสที่ใช้สำหรับเข้าใช้งาน API โดยรหัสนี้จะเป็นข้อความแทนอีเมล์ และรหัสผ่าน โดยเข้าไปที่หน้าเว็บ https://notify-bot.line.me/my/ จากนั้นระบบจะให้ทำการล็อกอินด้วย Account LINE โดยกรอกอีเมล์ และรหัสผ่านที่ได้ตั้งไว้ลงไป

ภาพที่ 4.28 กรอก User Password เพื่อการใช้งาน

ที่มา : notify-bot.line.me

เมื่อล็อกอินสำเร็จแล้วให้เลื่อนลงมาด้านล่างจะพบออก Access Token (สำหรับผู้พัฒนา) ให้กดปุ่ม ออก Token



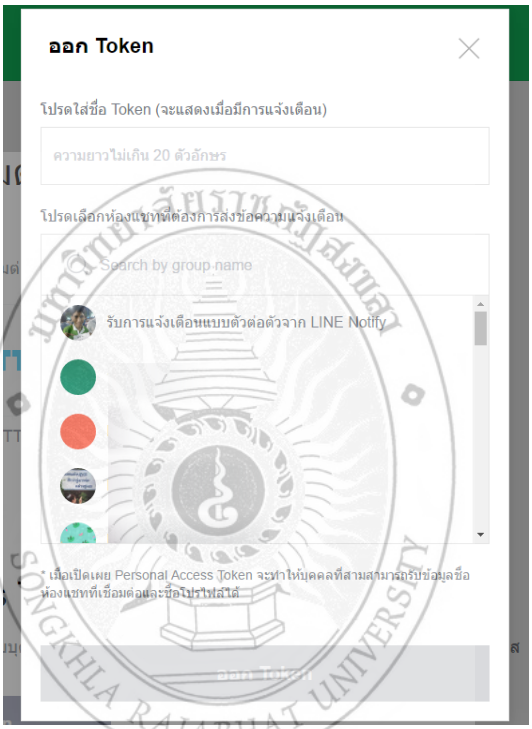
ภาพที่ 4.29 ออก Access Token

ที่มา : http://cz.lnwfile.com/_/cz/_raw/od/7a/dk.png

เมื่อกดออก Token เรียบร้อยแล้วระบบจะให้กรอกข้อมูลที่จะให้แสดงดังนี้

ช่องที่ 1 สามารถกรอกเป็นอะไรก็ได้ และสิ่งที่กรอกนั้นจะติดไปพร้อมกับข้อความเสมอ เช่น หากกรอกว่า Arduino เมื่อใช้ API ส่งข้อความว่า "สวัสดี" ข้อความจะขึ้นว่า "Arduino:สวัสดี"

ช่องที่ 2 จะให้เลือกว่าจะส่งข้อความเข้าไปในกลุ่มไหนหรือส่งให้ตัวเองเท่านั้นเมื่อกรอกครบแล้วให้กดปุ่มออก Token



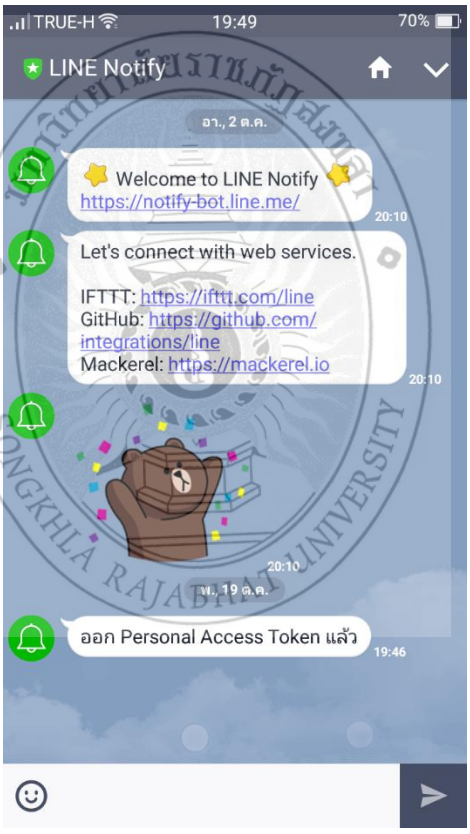
ภาพที่ 4.30 กรอกข้อมูลที่จะให้แสดง

ที่มา : http://cz.lnwfile.com/_/cz/_raw/bi/7d/di.png

เมื่อกดปุ่มแล้วจะปรากฏรหัส Token ให้เก็บรหัสนี้ไว้ให้ดีเพราะจะออกให้เพียงครั้งเดียว ส่วนใน LINE ก็จะมีการแจ้งเตือนว่าออก Access Token ใหม่แล้ว

ภาพที่ 4.31 แสดง Access Token

ที่มา : http://cz.lnwfile.com/_/cz/_raw/7e/af/sd.png



ภาพที่ 4.32 แจ้งเตือนว่าออก Access Token สำเร็จ

ที่มา : http://cz.lnwfile.com/_/cz/_raw/sy/8i/uu.png

วิธีการใช้งานร่วมกับ Arduino

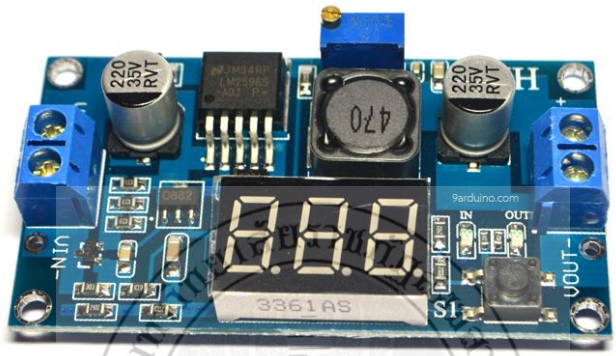
นำ Token ที่ได้มาใส่ใน Code บรรทัดที่ระบุไว้ในรูปภาพด้านล่าง แล้วแก้ไขข้อความที่ต้องให้ Line ส่งไปยังกลุ่มที่ต้องการ แล้วอัพโหลดไปยัง Board Arduino

ภาพที่ 4.33 แก้ไขข้อความที่ต้องการส่ง

ที่มา : http://cz.lnwfile.com/_/cz/_raw/sy/8i/uu.png

การควบคุมการทำงานของ Module DC-to-DC Step down Converter

Module DC-to-DC Step down Converter ทำหน้าที่แปลงระดับแรงดันไฟฟ้าตรงลง โดยสามารถปรับค่าแรงดัน output ได้โดย Potentiometer ที่มีอยู่บนบอร์ด สามารถจ่ายกระแสได้ถึง 3 A และใช้หลักการแปลงโดยวงจร Buck Converter ความถี่ 150 kHz ทำให้ทำงานเงียบ และแรงดันเรียบ



ภาพที่ 4.34 Module DC-to-DC Step down Converter LM2596

ที่มา : https://do.lnwfile.com/_/do/_raw/be/th/f6.png

คุณสมบัติของ Module DC-to-DC Step down Converter

- รองรับแรงดันเข้าที่ 4V-35V
- รองรับแรงดันออกที่ 1.23V-30V
- รองรับกระแสออกสูงสุดที่ 3A
- มีหน้าจอแสดงสถานะแรงดันไฟ

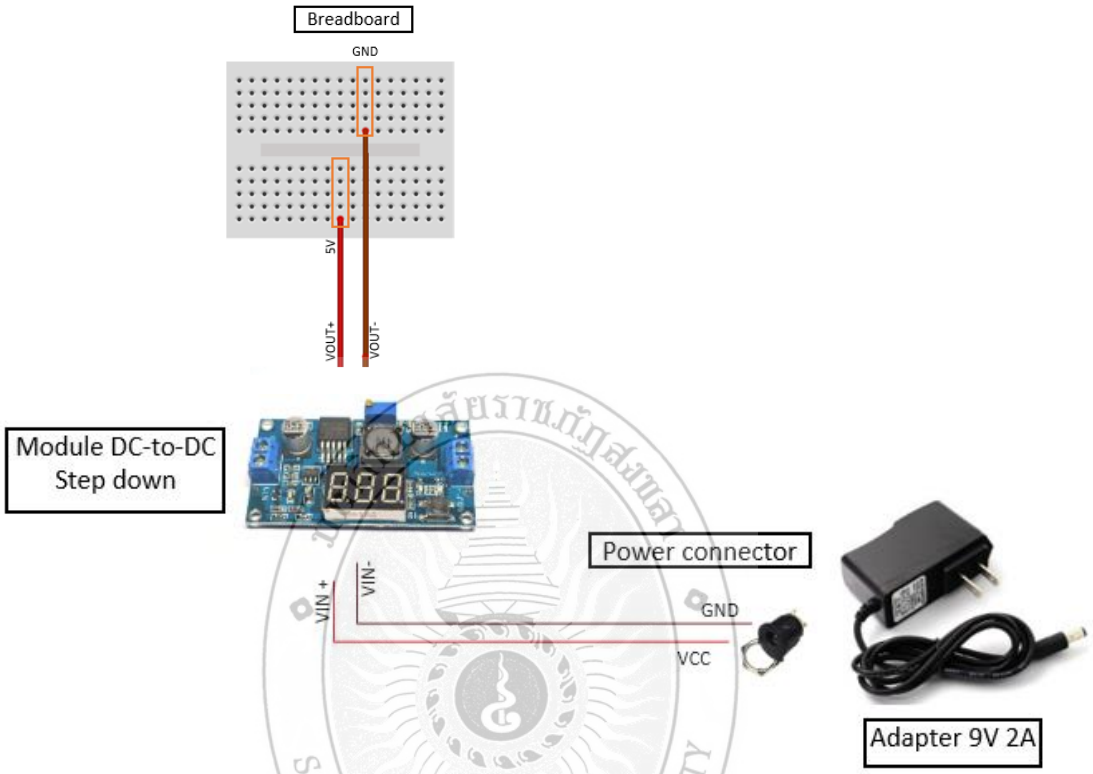
วิธีการใช้งาน

โมดูลตัวนี้ออกแบบมาให้ปรับค่าโวลต์ได้แบบละเอียด ดังนั้นหมุนที่ตัวต้านทานปรับค่าได้บนบอร์ด หลายๆรอบดังนี้

-หมุนทวนเข็มนาฬิกา : ปรับไฟลง

- หมุนตามเข็มนาฬิกา : ปรับไฟขึ้น

การต่อใช้งาน Module DC-to-DC Step down Converter



ภาพที่ 4.35 การต่อใช้งาน Module DC-to-DC Step down Converter

วิธีการต่อขาใช้งาน

ทำการต่อ Adapter เข้ากับ Power Connecter เพื่อแยกสาย VCC และ GND

VCC ต่อเข้ากับ VIN+

GND ต่อเข้ากับ VIN-

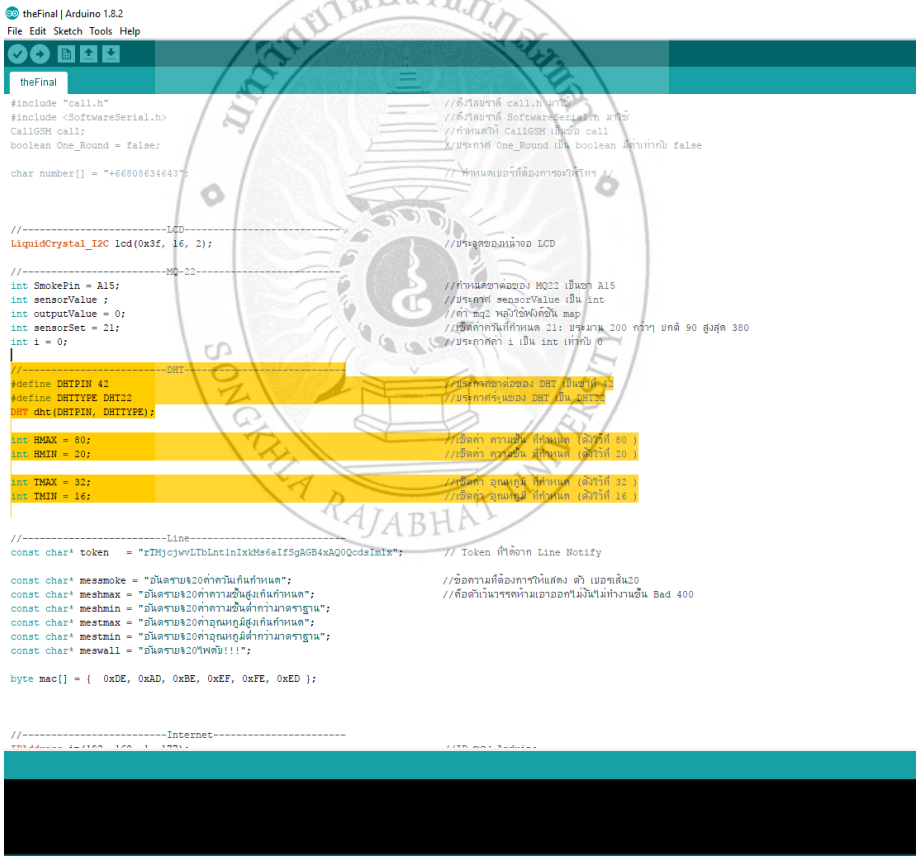
VOUT+ ต่อเข้ากับ Breadboard แถว 5V

VOUT- ต่อเข้ากับ Breadboard แถว GND

การแก้ไขค่าโค้ดโปรแกรม

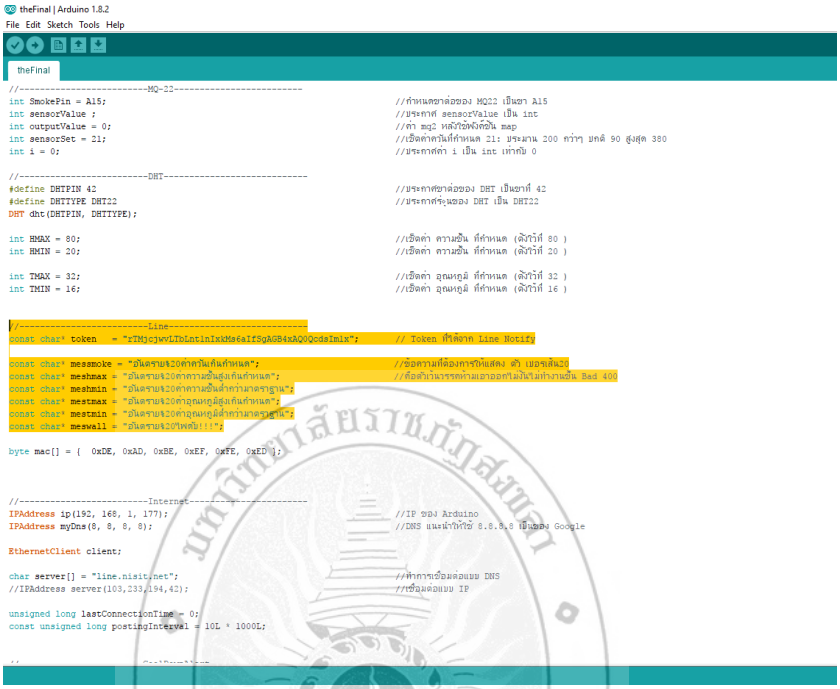
เป็นขั้นตอนการอัปโหลด ข้อมูลการทำงานอัตโนมัติตามที่กำหนด เช่น กำหนดค่าอุณหภูมิและความชื้น,Token ของ Line ,ข้อความที่ต้องการแจ้งเตือน และ เบอร์โทรศัพท์มือถือที่ต้องการให้โทร เพื่อให้ระบบทำงานได้ตรงตามความต้องการและแม่นยำมากที่สุด ซึ่งจะทำให้ระบบนั้นมีความถูกต้องตามที่กำหนดไว้ซึ่งมีขั้นตอนพอสังเขป

1. การเขียนโค้ดกำหนดค่า อุณหภูมิ สูง/ต่ำ และ ค่าความชื้น สูง/ต่ำ เพื่อตั้งค่าระบบให้ทำงานเหมาะสมตามความต้องการในปัจจุบัน ตามผู้ใช้กำหนด เช่น กำหนดค่าความชื้นสูงสุดเท่ากับ 80 เปอร์เซ็น ถ้าค่ามากกว่านี้ระบบก็จะทำการแจ้งเตือน



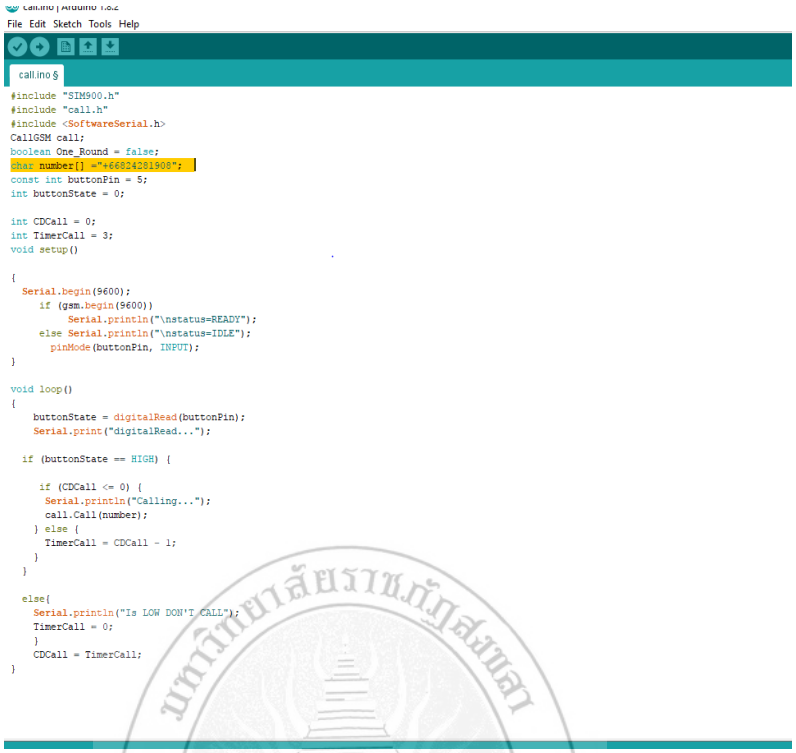
ภาพที่ 4.36 การกำหนดค่าความชื้นและค่าอุณหภูมิ

2. การเขียนโค้ดกำหนดค่า Token ของ Line และข้อความที่ต้องการให้ระบบส่งการแจ้งเตือนไปตามที่ผู้ใช้กำหนด



ภาพที่ 4.37 การกำหนดค่า Token และ ข้อความที่ต้องการแจ้งเตือน

3. การเขียนโค้ดกำหนดหมายเลขโทรศัพท์ที่ต้องการให้ระบบโทรเมื่อเกิดข้อผิดพลาดในการเชื่อมต่ออินเทอร์เน็ต เช่น char number[] ="+66824281908"; กำหนดให้ระบบโทรไปหาหมายเลขโทรศัพท์ : 0824281908



```
File  Edit  Sketch  Tools  Help

call.ino §

#include "SIM900.h"
#include "call.h"
#include <SoftwareSerial.h>
CallGSM call;
boolean One_Round = false;
char number[] ="+66824281908";
const int buttonPin = 5;
int buttonState = 0;

int CDCall = 0;
int TimerCall = 3;
void setup()

{
  Serial.begin(9600);
     if (gsm.begin(9600))
          Serial.println("\nstatus=READY");
     else Serial.println("\nstatus=IDLE");
       pinMode(buttonPin, INPUT);
}

void loop()
{
    buttonState = digitalRead(buttonPin);
    Serial.print("digitalRead...");

  if (buttonState == HIGH) {

     if (CDCall <= 0) {
      Serial.println("Calling...");
      call.Call(number);
    } else {
      TimerCall = CDCall - 1;
    }
  }

  else{
    Serial.println("Is LOW DON'T CALL");
    TimerCall = 0;
    }
    CDCall = TimerCall;
}
```

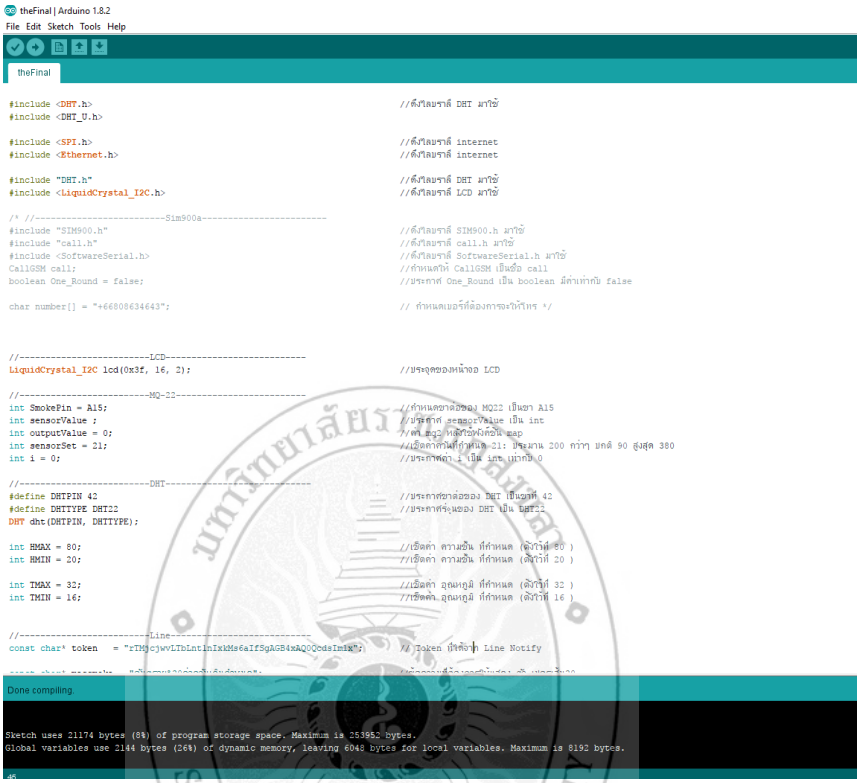
ภาพที่ 4.38 การกำหนดหมายเลขโทรศัพท์

4.หลังจากเขียนโค้ดกำหนดค่าต่างๆเสร็จเรียบร้อย กดที่ปุ่มเครื่องหมายถูกสีเหลืองดังภาพ เพื่อตรวจสอบโค้ดว่าถูกต้องหรือไม่

```
theFinal | Arduino 1.8.2
File  Edit  Sketch  Tools  Help

theFinal

#include <DHT.h>                                        //ดึงไลบรารี่ DHT มาใช้
#include <DHT_U.h>

#include <SPI.h>                                        //ดึงไลบรารี่ internet
#include <Ethernet.h>                                   //ดึงไลบรารี่ internet

#include "DHT.h"                                        //ดึงไลบรารี่ DHT มาใช้
#include <LiquidCrystal_I2C.h>                          //ดึงไลบรารี่ LCD มาใช้

/* //------------------------Sim900a-----------------------
#include "SIM900.h"                                     //ดึงไลบรารี่ SIM900.h มาใช้
#include "call.h"                                       //ดึงไลบรารี่ call.h มาใช้
#include <SoftwareSerial.h>                             //ดึงไลบรารี่ SoftwareSerial.h มาใช้
CallGSM call;                                           //กำหนดให้ CallGSM เป็นชื่อ call
boolean One_Round = false;                              //ประกาศ One_Round เป็น boolean มีค่าเท่ากับ false

char number[] = "+66808634643";                         // กำหนดเบอร์ที่ต้องการจะให้โทร */


//------------------------LCD-----------------------
LiquidCrystal_I2C lcd(0x3f, 16, 2);                     //ประกาศตัวแปรของหน้าจอ LCD

//------------------------MQ-22-----------------------
int SmokePin = A15;                                     //กำหนดขาต่อของ MQ22 เป็นขา A15
int sensorValue ;                                       //ประกาศ sensorValue เป็น int
int outputValue = 0;                                    //ค่า mq2 หลังใช้ฟังก์ชัน map
int sensorSet = 21;                                     //เรียกความไวที่กำหนด 21: ประมาณ 200 ถ่วๆ มากสุด 90 สู่งสุด 380
int i = 0;                                              //ประกาศตัวแ i เป็น int เท่ากับ 0

//------------------------DHT-----------------------
#define DHTPIN 42                                       //ประกาศขาต่อของ DHT เป็นขาที่ 42
#define DHTTYPE DHT22                                   //ประกาศรุ่นของ DHT เป็น DHT22
DHT dht(DHTPIN, DHTTYPE);

int HMAX = 80;                                          //เรียกว่า ความชื้น ที่กำหนด (ตัวโรที่ 80 )
int HMIN = 20;                                          //เรียกว่า ความชื้น ที่กำหนด (ตัวโรที่ 20 )

int TMAX = 32;                                          //เรียกว่า อุณหภูมิ ที่กำหนด (ตัวโรที่ 32 )
int TMIN = 16;                                          //เรียกว่า อุณหภูมิ ที่กำหนด (ตัวโรที่ 16 )

//------------------------Line-----------------------
const char* token   = "rTMjcjwvLTbLntlnIxkMs6alfSgAGB4xAQOodsImlx";      // Token ที่ใช้ส่ง Line Notify

Compiling sketch...
```

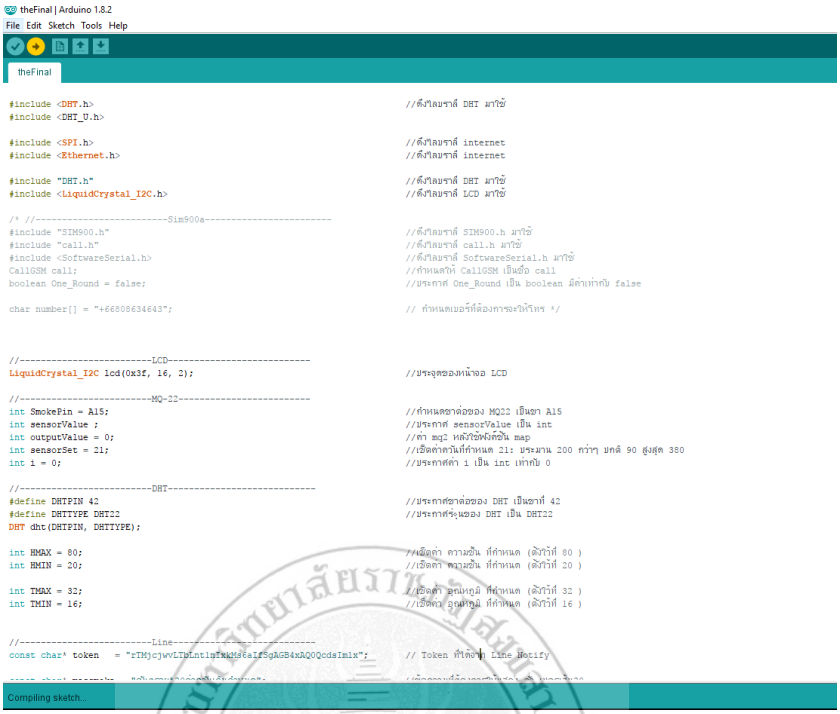
ภาพที่ 4.39 ตรวจสอบความถูกต้องของโค้ดโปรแกรม

5.เมื่อโปรแกรมตรวจสอบแล้วว่าโค้ดถูกต้อง จะขึ้นแสดงผลว่า Done Compiling. หรือ ถ้าโปรแกรมตรวจสอบว่าโค้ดผิดจะขึ้นแจ้งว่าผิดที่บรรทัดไหนเป็นสีแดง



ภาพที่ 4.40 แสดงคำว่า Done Compiling.หรือสถานะถูกต้อง

6.จากนั้นให้กดที่ปุ่ม Upload เพื่ออัปโหลดโค้ดโปรแกรมไปยังบอร์ดอาดูโน่ เป็นอันเสร็จสิ้นการอัปโหลดโค้ดโปรแกรม

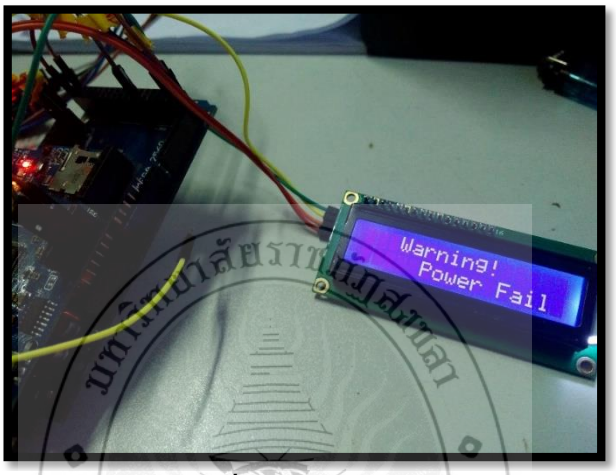


ภาพที่ 4.41 ทำการอัพโหลดโค้ดข้อมูลไปยังระบบ

การทดสอบ

หลังจากที่ผู้พัฒนาได้ทำการพัฒนาต้นแบบระบบแจ้งเตือนความปลอดภัยของห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์ แล้ว ผู้พัฒนาได้ทำการทดสอบระบบ ดังนี้

1. การจำลองไฟฟ้าดับโดยการดึงสาย Adapter ที่เป็นตัววัดสถานะของไฟฟ้าออก ที่หน้าจอ LCD ก็จะแสดงสถานะแจ้งเตือนไฟฟ้าดับ และมีการส่งการแจ้งเตือนไปยังแอปพลิเคชันไลน์ด้วย



ภาพที่ 4.42 ทดสอบไฟฟ้าดับ

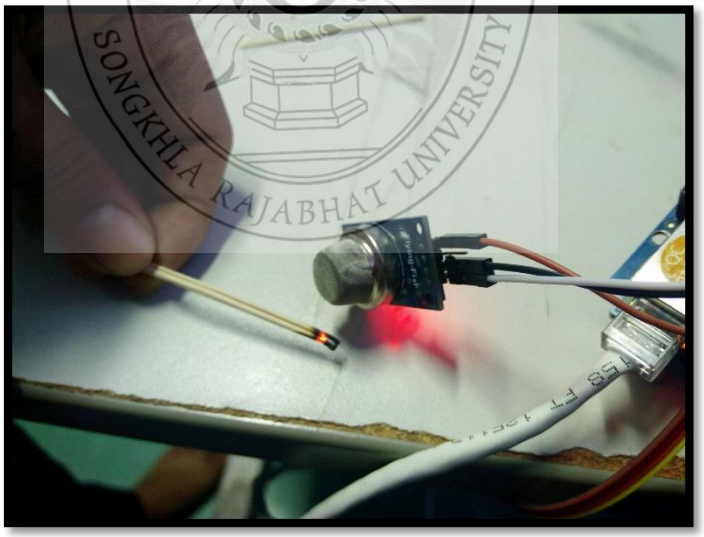
2. หากไม่มีการเสียบสาย LAN เพื่อเชื่อมต่อกับเครื่องข่าย ที่หน้าจอ LCD ก็จะแสดงข้อความ Checking Network เพื่อทำการเช็คสถานะของเครือข่าย จนกว่าจะมีการเชื่อมต่อกับเครือข่ายจึงจะใช้งานได้

ภาพที่ 4.43 ทดสอบการเช็คอินเทอร์เน็ต

3. เมื่อทำการเชื่อต่อกับเครือข่ายอินเทอร์เน็ตได้แล้วหน้าจอ LCD จะแสดงหมายเลข IP Address ของเครือข่าย

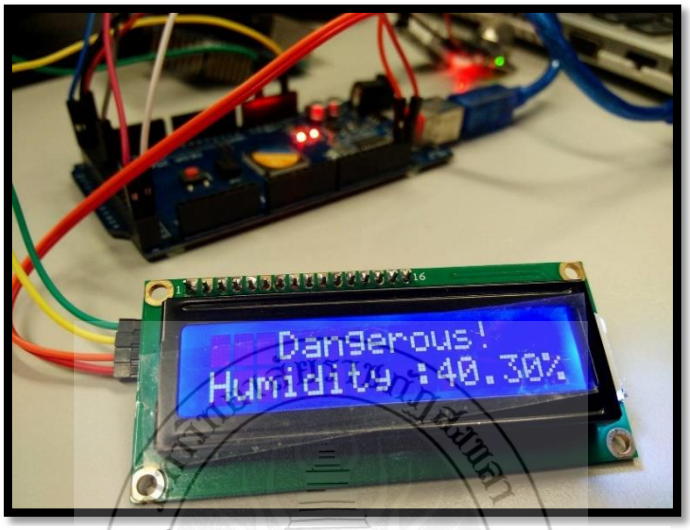
ภาพที่ 4.44 ทดสอบการเชื่อมต่ออินเทอร์เน็ต

4. จากที่ได้ทำการทดลองกับเซนเซอร์ MQ-2 เมื่อมีควันเกิดในระยะของเซนเซอร์และเกิดค่าที่กำหนด ระบบจะส่งการแจ้งเตือนไปยังแอปพลิเคชันไลน์ของผู้ใช้



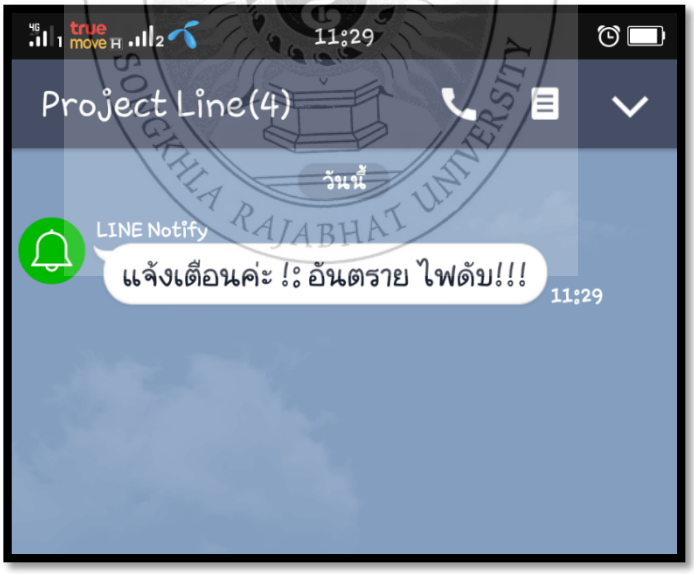
ภาพที่ 4.45 ทดสอบเซนเซอร์ MQ-2

5. เมื่ออุณหภูมิภายในห้องมีค่าสูงกว่าที่กำหนดบนหน้าจอ LCD จะมีข้อความบอกสถานะในขณะเดียวกันระบบก็จะทำการส่งข้อความแจ้งเตือนไปยังแอปพลิเคชันไลน์ให้กับผู้ใช้ได้ทราบเช่นกัน



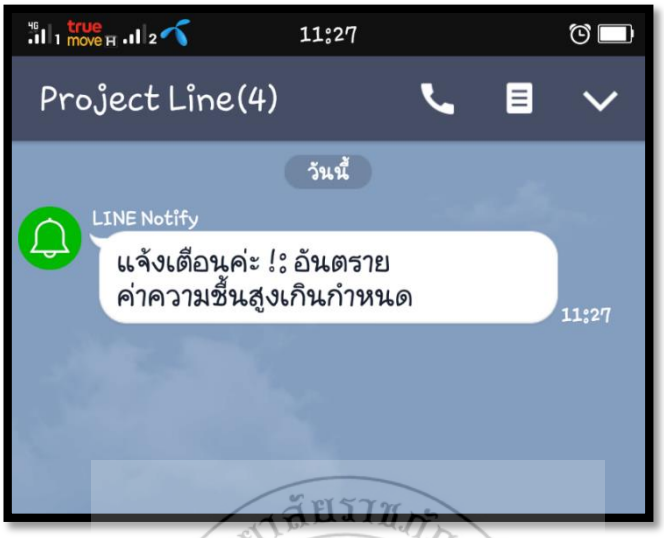
ภาพที่ 4.46 การทดสอบเซนเซอร์ DHT-22

6. ข้อความแจ้งเตือนเมื่อเกิดไฟฟ้าดับ



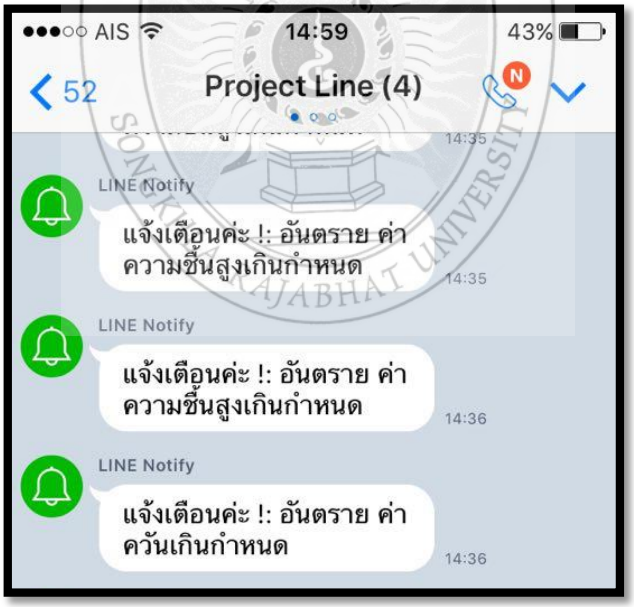
ภาพที่ 4.47 ข้อความแจ้งเตือนเกิดไฟไหม้

7. ข้อความแจ้งเตือนเมื่อค่าความชื้นสูงเกิดกำหนด



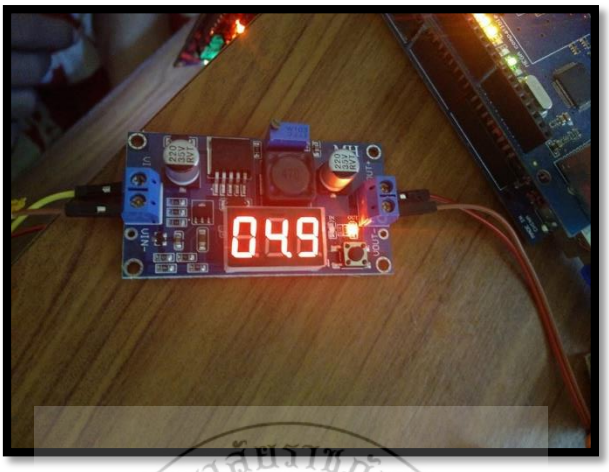
ภาพที่ 4.48 ข้อความแจ้งเตือนความชื้น

8. ข้อความแจ้งเตือนเมื่อเกิดควันไฟ



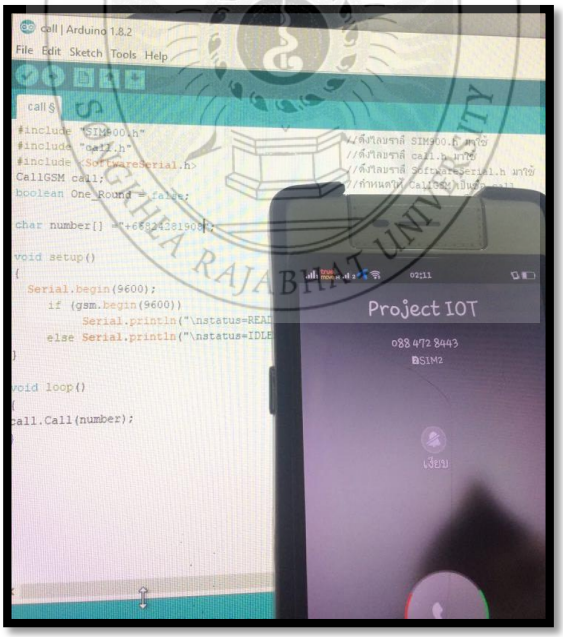
ภาพที่ 4.49 ข้อความแจ้งเตือนควันไฟ

9. การทดลองเปลี่ยนกระแสไฟฟ้าจาก 9V เป็น 5V



ภาพที่ 4. 50 การทดลองเปลี่ยนกระแสไฟฟ้า

10. การทดลองระบบการโทร เมื่อขาดการเชื่อมต่ออินเทอร์เน็ตและไม่สามารถส่งข้อความแจ้งเตือนผ่านทางแอปพลิเคชันไลน์ได้ระบบจะทำการโทรเข้าไปยังโทรศัพท์ของผู้ใช้ระบบ



ภาพที่ 4. 51 การทดลองการโทร

บทที่ 5

บทสรุปและข้อเสนอแนะ

สรุปผลการดำเนินงาน

ผลการดำเนินงานเป็นไปตามวัตถุประสงค์ที่ได้ตั้งไว้ คือ สามารถพัฒนาระบบแจ้งเตือนความปลอดภัยในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์ กล่าวคือ

1. สามารถตรวจจับสถานะของไฟฟ้าได้

2. สามารถตรวจจับค่าของควันได้เมื่อเกิดควันภายในห้อง

3. สามารถตรวจวัดอุณหภูมิและความชื้นได้

4. สามารถส่งส่งข้อความแจ้งเตือนผ่านแอปพลิเคชันไลน์ได้ เมื่อมีสิ่งผิดปกติเกิดขึ้นภายในห้องเซิร์ฟเวอร์

ปัญหาและอุปสรรคในการดำเนินงาน

1. อุปกรณ์เกิดความเสียหายได้ง่าย

2. SIM 900A เป็นอุปกรณ์ที่หายากในปัจจุบันเนื่องจากมีรุ่นใหม่มาทดแทน

3. SIM 900A มีความเสียหายจึงเกิดความล่าช้าในการทำงานเนื่องจากต้องมีการสั่งอุปกรณ์มาใหม่

4. กำลังไฟไม่สม่ำเสมอ ทำให้ SIM 900A ไม่ทำงาน

5. ในการต่ออุปกรณ์ต่าง ๆ เข้ากับบอร์ดควบคุมควรจะตรวจเช็คให้เรียบร้อยก่อนจะเปิดระบบทำงาน เพราะเมื่อเกิดปัญหาแล้วจะตรวจหาจุดที่เสียหายยาก และเสียเวลา

ข้อจำกัดของระบบ

ระบบที่พัฒนานี้ ยังมีข้อกัดในด้านกำลังไฟฟ้า เนื่องจาก บอร์ด SIM 900A ต้องใช้ความคงที่ของฟ้าไฟและต้องมีไฟฟ้าเข้ามาเลี้ยงบอร์ดที่เพียงพอ บอร์ดจึงจะทำงาน หากภายในห้องมีไฟฟ้าที่ไม่คงที่ อาจทำให้บอร์ดไม่ทำงานได้ และเซนเซอร์ MQ-2 ที่ใช้ตรวจจับค่าของควันก็มีขอบเขตที่จำกัดในการที่จะรับค่าของควัน เมื่อเกิดควันในห้อง

ข้อเสนอแนะ

1. ควรศึกษาหลักการทำงานของระบบแจ้งเตือนความปลอดภัยในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์ให้เข้าใจอย่างลึกซึ้งเพื่อการนำไปใช้กับอุปกรณ์ต่าง ๆตามที่เราต้องการอย่างถูกต้องสมราคาและเต็มประสิทธิภาพ

2. ควรศึกษารายละเอียดของบอร์ด อุปกรณ์ต่าง ๆ ก่อนที่จะทำการประกอบ การเชื่อมต่อสายต่าง ๆ เพื่อป้องกันความผิดพลาดที่จะเกิดขึ้น

3. สามารถนำระบบแจ้งเตือความปลอดภัยในห้องเซิร์ฟเวอร์ผ่านแอปพลิเคชันไลน์ไปพัฒนาต่อยอด เพื่อรองรับห้องเซิร์ฟเวอร์ที่มีขนาดใหญ่ขึ้นได้ และยังสามารถพัฒนาต่อให้มีการรองรับสัญญาณ Wifi เพื่อสะดวกต่อผู้ใช้

4. ขนาดของชิ้นงานเป็นขนาดทดลอง หากต้องการนำไปพัฒนาเพื่อรองรับห้องเซิร์ฟเวอร์ที่มีขนาดใหญ่ขึ้น สามารถแยกเซนเซอร์ออกเป็น Node เพื่อกระจายเซนเซอร์ได้ทั่วห้องเซิร์ฟเวอร์ได้

5. ปรับเปลี่ยนอุปกรณ์ให้มีประสิทธิภาพมากขึ้น เพื่อความแม่นยำในการตรวจจับค่าต่าง ๆ

มหาวิทยาลัยราชภัฏสงขลา SONGKHLA RAJABHAT UNIVERSITY

บรรณานุกรม

ประจิน พลังสันติกุล. (2549). **การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ AVR ด้วยภาษา C กับWinAVR (C Compiler)**. แอพซอฟต์เทคกรุงเทพฯ.

My arduino. (2560). **บทความ My arduino**. จากhttp://www.myarduino.net/article

Engineering Engineering Educational Solution. (2560). **ZX-MQ-2 แผงวงจรตรวจจับควัน**. จาก http://inex.co.th/shop/zx-mq2.html

ThaiSensorModule. (2560). **โมดูลตรวจวัดแก๊สไวไฟ (MQ-5)**. จาก http://thaisensormodule.com/index.php/gas-sensor/mq5

ThaiEasyElec. (2560). **Carbon Monoxide & Flammable Gas Sensor - MQ-9**. จาก http://www.thaieasyelec.com/products/sensors/gas/carbon-monoxide-flammable-gas-sensor-mq-9-detail.html

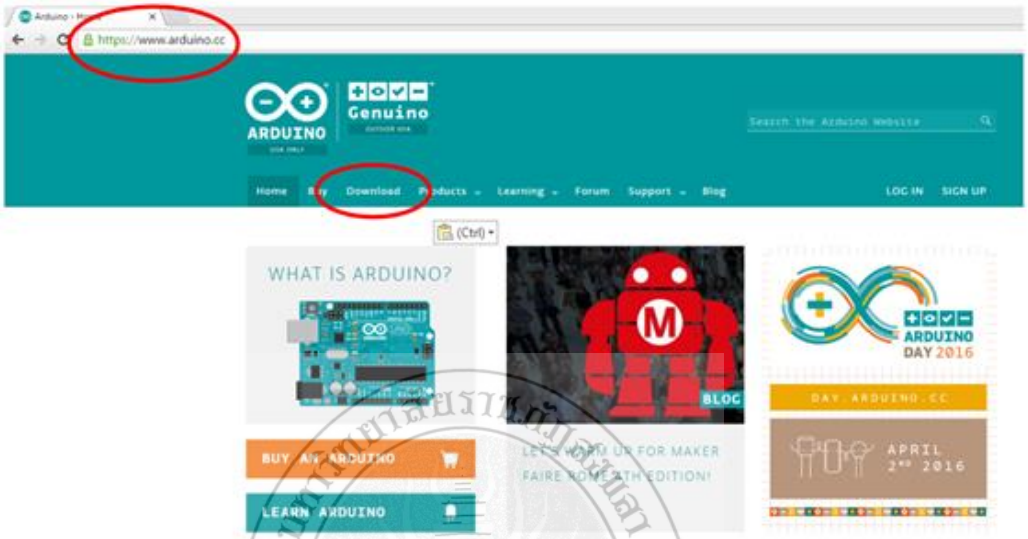
This work is licensed under: Creative Commons Attribution-NonCommercial 3.0 Unported. 2560. **DHT22 / AM2302 Temperature & Relative Humidity Sensor**. จาก http://cpre.kmutnb.ac.th/esl/learning/index.php?article=dht22_am2302

ภาคผนวก ก

วิธีติดตั้งโปรแกรม Arduino IDE

คู่มือการติดตั้งซอฟต์แวร์ Arduino IDE

1. ดาวน์โหลด Arduino IDE จาก https://www.arduino.cc โดยคลิกที่เมนู Download



ภาพที่ ก.1 ดาวน์โหลด Arduino IDE

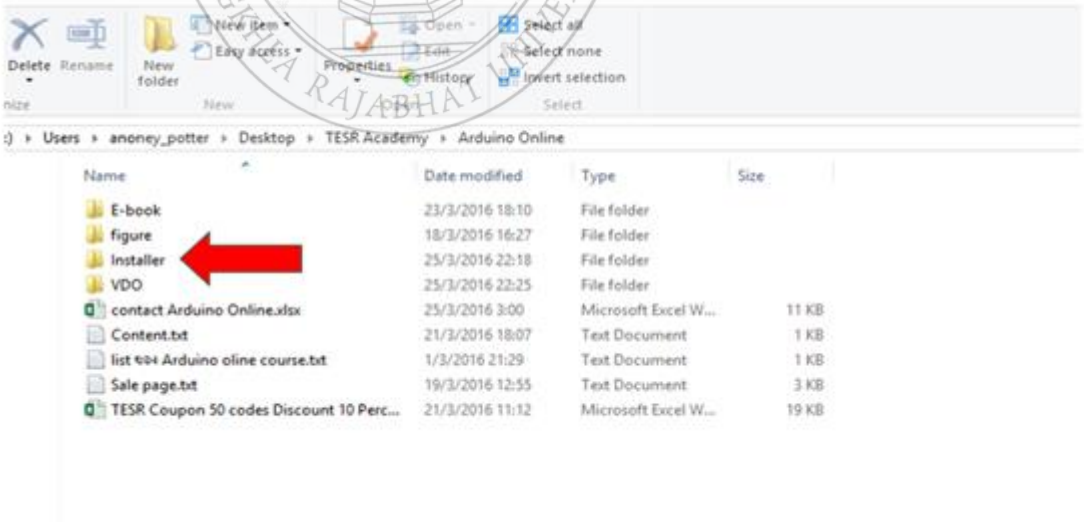
2. เลือกระบบปฏิบัติการของเครื่องคอมพิวเตอร์ที่จะใช้ในการเขียนโปรแกรม Arduino

ภาพที่ ก. 2 เลือกระบบปฏิบัติการ

3. กด JUST DOWNLOAD

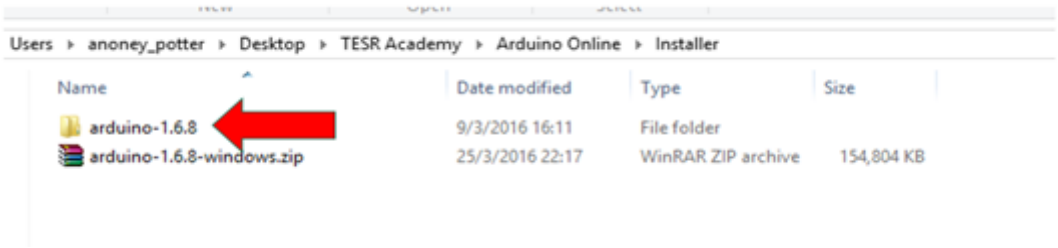
ภาพที่ ก. 3 กด JUST DOWNLOAD

4. บันทึกไฟล์ติดตั้งโปรแกรมไปยังโฟลเดอร์ที่ต้องการ



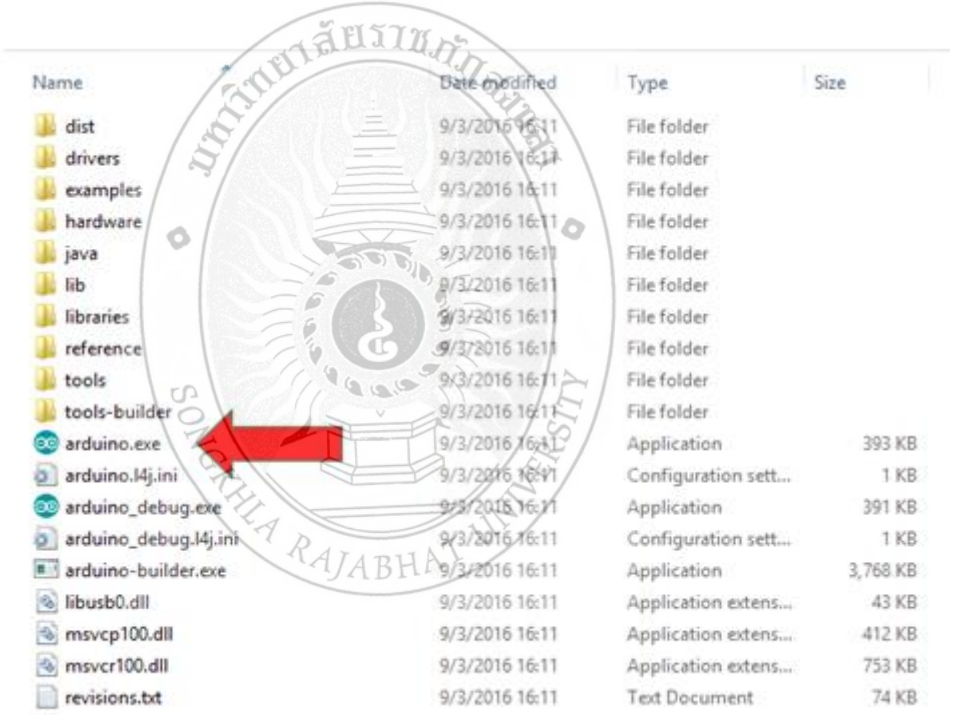
ภาพที่ ก. 4 บันทึกไฟล์ที่ติดตั้ง

5. Unzip ไฟล์ติดตั้ง



ภาพที่ ก. 5 Unzip ไฟล์ติดตั้ง

6. ดับเบิ้ลคลิกที่ไฟล์ arduino.exe

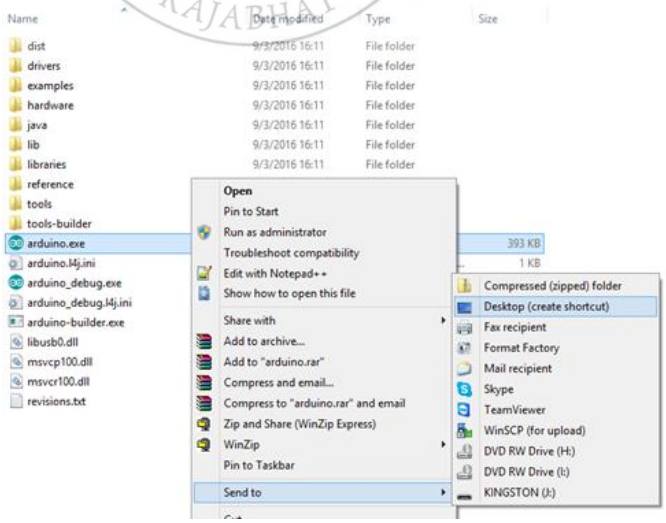


ภาพที่ ก. 6 เปิดโปรแกรม

7.หน้าต่างโปรแกรม Arduino จะปรากฏขึ้น ดังรูป

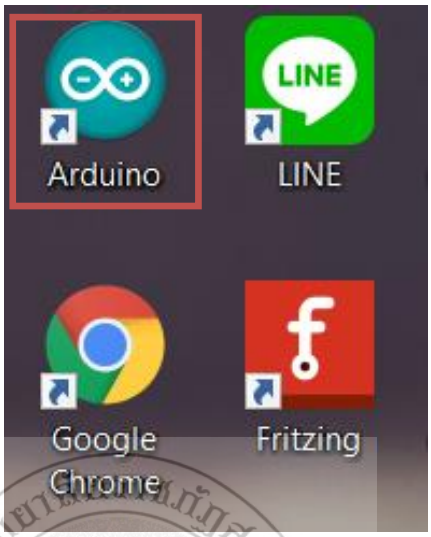
ภาพที่ ก. 7 หน้าต่างโปรแกรม

8. สร้าง Shortcut บน Desktop เพื่อความสะดวกในการเปิดโปรแกรมในครั้งต่อๆไป



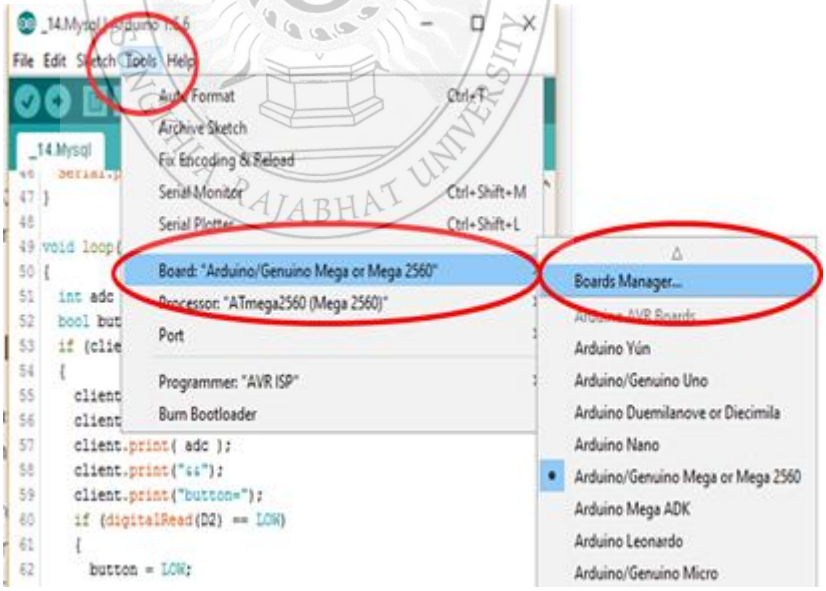
ภาพที่ ก. 8 สร้าง Shortcut

9. จะปรากฏไอคอนของโปรแกรม Arduino บน Desktop ดังแสดงในรูป



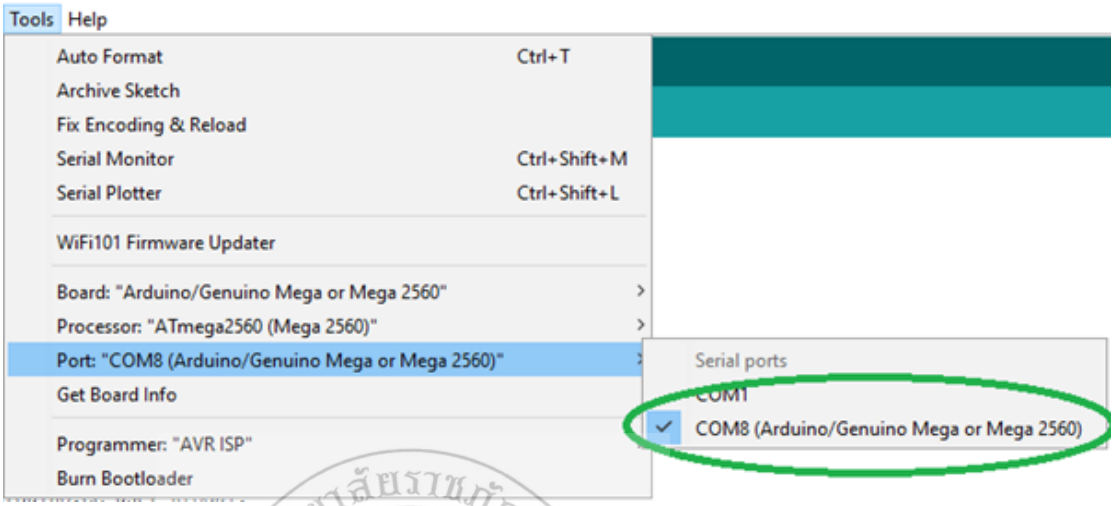
ภาพที่ ก. 9 ไอคอนโปรแกรม

10. ในหน้าต่างโปรแกรม Arduino ให้คลิกไปที่เมนู Tools -> Board -> Arduino/Genuino Mega or Mega 2560



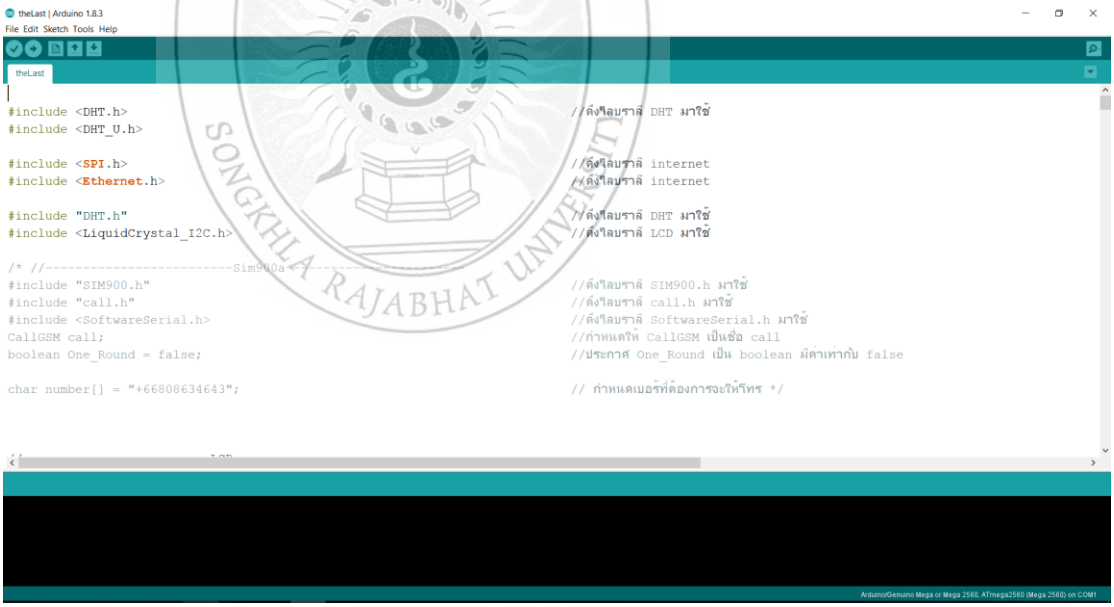
ภาพที่ ก. 10 เลือกบอร์ดที่ต้องการใช้

11. ทำการเลือก Port ให้ตรงกับช่องที่ได้ทำการเสียบบอร์ดไว้



ภาพที่ ก. 11 เลือก Port

14. เริ่มต้นการเขียนโปรแกรมลงบอร์ดได้ตามต้องการ



ภาพที่ ก. 12 หน้าต่างการเขียนโปรแกรม

ภาคผนวก ข

โค้ดระบบแจ้งเตือนความปลอดภัยในห้องเซิร์ฟเวอร์ผ่าน แอปพลิเคชันไลน์

โค้ดสำหรับอัปโหลดเข้าบอร์ด Arduino MEGA 2560

```
#include <DHT.h>                    //ดึงไลบรารี DHT มาใช้
#include <DHT_U.h>
#include <SPI.h>                    //ดึงไลบรารี internet
#include <Ethernet.h>               //ดึงไลบรารี internet

#include "DHT.h"                    //ดึงไลบรารี DHT มาใช้
#include <LiquidCrystal_I2C.h>      //ดึงไลบรารี LCD มาใช้

//------------------------LCD-------------------------
LiquidCrystal_I2C lcd(0x3f, 16, 2);   //ประจุดของหน้าจอ LCD

//------------------------MQ-22-----------------------
int SmokePin = A15;                 //กำหนดขาต่อของ MQ22 เป็นขา A15
int sensorValue ;                   //ประกาศ sensorValue เป็น int
int outputValue = 0;                //ค่า mq2 หลังใช้ฟังค์ชั้น map
int sensorSet = 21
int i = 0;                          //ประกาศค่า i เป็น int เท่ากับ 0

//------------------------DHT-------------------------
#define DHTPIN 42                   //ประกาศขาต่อของ DHT เป็นขาที่ 42
#define DHTTYPE DHT22               //ประกาศรุ่นของ DHT เป็น DHT22
DHT dht(DHTPIN, DHTTYPE);
int HMAX = 80;                      //เซ็ตค่า ความชื้น ที่กำหนด (ตั้งไว้ที่ 80 )
int HMIN = 20;                      //เซ็ตค่า ความชื้น ที่กำหนด (ตั้งไว้ที่ 20 )

int TMAX = 32;                      //เซ็ตค่า อุณหภูมิ ที่กำหนด (ตั้งไว้ที่ 32 )
int TMIN = 16;                      //เซ็ตค่า อุณหภูมิ ที่กำหนด (ตั้งไว้ที่ 16 )
```

```
28.
29. //------------------------Line--------------------------
30. const char* token   = "rTMjcjwvLTbLnt1nIxkMs6aIfSgAGB4xAQ0QcdsIm1x";
31. const char* messmoke = "อันตราย%20ค่าควันเกินกำหนด";
32. const char* meshmax = "อันตราย%20ค่าความชื้นสูงเกินกำหนด";
33. const char* meshmin = "อันตราย%20ค่าความชื้นต่ำกว่ามาตราฐาน";
34. const char* mestmax = "อันตราย%20ค่าอุณหภูมิสูงเกินกำหนด";
35. const char* mestmin = "อันตราย%20ค่าอุณหภูมิต่ำกว่ามาตราฐาน";
36. const char* meswall = "อันตราย%20ไฟดับ!!!";
37. byte mac[] = {  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
38. //------------------------Internet----------------------
39. IPAddress ip(192, 168, 1, 177);       //IP ของ Arduino
40. IPAddress myDns(8, 8, 8, 8);          //DNS แนะนำให้ใช้ 8.8.8.8 เป็นของ Google
41. EthernetClient client;
42. char server[] = "line.nisit.net";          //ทำการเชื่อมต่อแบบ DNS
43. //IPAddress server(103,233,194,42);    //เชื่อมต่อแบบ IP
44. unsigned long lastConnectionTime = 0;
45. const unsigned long postingInterval = 10L * 1000L;
46.
47. //------------------------CoolDownAlert-----------------
48. int CDhmax = 0;                  //ประกาศค่า CDhmax เป็น int เท่ากับ 0
49. int Timerhmax = 3;                //ประกาศค่า Timerhmax เป็น int เท่ากับ 0
50. int CDhmin = 0;                  //ประกาศค่า CDhmin เป็น int เท่ากับ 0
51. int Timerhmin = 3;                //ประกาศค่า Timerhmin เป็น int เท่ากับ 0
52. int CDTmax = 0;                  //ประกาศค่า CDTmax เป็น int เท่ากับ 0
53. int TimerTmax = 3;                //ประกาศค่า TimerTmax เป็น int เท่ากับ 0
54. int CDTmin = 0;                  //ประกาศค่า CDTmin เป็น int เท่ากับ 0
55. int TimerTmin = 3;                //ประกาศค่า TimerTmin เป็น int เท่ากับ 0
56. int CDSmoke = 0;                 //ประกาศค่า CDSmoke เป็น int เท่ากับ 0
57. int TimerSmoke = 3;               //ประกาศค่า TimerSmoke เป็น int เท่ากับ 0
```

```
58. int CDWall = 0;                    //ประกาศค่า CDWall เป็น int เท่ากับ 0
59. int TimerWall = 3;                 //ประกาศค่า TimerWall เป็น int เท่ากับ 0
60.
61. //------------------------wallPlug--------------------
62. const int buttonPin = 17;        // กำหนดค่า buttonPin ให้รับค่าจากขาที่ 7
63. int buttonState = 0;              // กำหนดค่า buttonState มีค่าเท่ากับ 0
64.
65. //------------------------CallOutPut-------------------
66. const int CallbuttonPin = 2;          // กำหนดค่า CallbuttonPin เป็นขาที่ 2
67.
68. int CallbuttonState = 0;
69. //------------------------SetUp-----------------------
70. void setup() {
71.
72.   Serial.begin(9600);                 //สั่งให้ Serial เริ่มทำงาน
73.   pinMode(buttonPin, INPUT);      // กำหนดให้ buttonPin เป็นตัว INPUT หรือตัวรับค่า
74.   pinMode(CallbuttonPin, OUTPUT); // กำหนดให้ CallbuttonPin เป็นตัว OUTPUT หรือ
    ตัวส่งค่า
75.   lcd.begin();                                  //LCD เริ่มทำงาน
76.   delay(4000);                                  //รอเวลา 4 วิ
77.
78.   lcd.print("Checking Network");                 //สั่งหน้าจอ
    แสดง Checking Network
79.   if (Ethernet.begin(mac) == 0) {               //สั่งถ้า เมื่อทำการเช็ค
    อินเตอร์เน็ทแล้วค่า mac = 0 จะทำการรีบูตเครื่องใหม่
80.     softReset();                                //รีบูทเครื่องใหม่
81.     Ethernet.begin(mac, ip);
82.   }
83.   delay(1000);                                  //รอเวลา 1 วิ
84.
```

```
85.  Serial.println("connecting...");
86.  lcd.setCursor(0, 0);                    //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 1
87.  lcd.print("  Connecting... ");          //สั่งหน้าจอแสดง Connecting...
88.  delay(2000);
89.
90.  if (client.connect(server, 80)) {
91.    Serial.println("  connected");
92.    lcd.setCursor(0, 0);                  //สั่งกำหนดค่าหน้าจอทีตัวที่ 1 แถวที่ 1
93.    lcd.print("   Connected   ");         //สั่งหน้าจอแสดง Connected
94.    delay(1000);                          //รอเวลา 1 วิ
95.
96.    client.println("GET /search?q=arduino HTTP/1.1");          // Make a HTTP request:
97.    client.println("Host: www.google.com");
98.    client.println("Connection: close");
99.    client.println();
100.
101.           lcd.setCursor(0, 0);
102.           lcd.print("My IP address:  ");
103.           lcd.setCursor(3, 1);
104.           lcd.print(Ethernet.localIP());
105.
106.         } else {
107.
108.           Serial.println("connection failed");
109.           lcd.setCursor(0, 0);           //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 1
110.           lcd.println("   Connection   "); //สั่งหน้าจอแสดง Connected
111.
112.           lcd.setCursor(5, 1);           //สั่งกำหนดค่าหน้าจอที่ตัวที่ 6 แถวที่ 2
113.           lcd.println("failed     ");     //สั่งหน้าจอแสดง failed
114.           lcd.println("Check your Network");
```

```
115.          delay(2000);                    //รอเวลา 2 วิ
116.        }
117.        delay(5000);                      //รอเวลา 5 วิ
118.        }
119.
120.      //-----------------------Loop-------------------------
121.      void loop() {                       //ฟังก์ชั่น Loop
122.        Wall();                           //เรียกใช้งานฟังก์ชั่น Wall มาใช้งาน
123.        Temps();                          //เรียกใช้งานฟังก์ชั่น Temps มาใช้งาน
124.        Smoke();                          //เรียกใช้งานฟังก์ชั่น Smoke มาใช้งาน
125.        delay (4000);                     //รอเวลา 4 วิ
126.      }
127.      //-----------------------Smoke------------------------
128.      void Smoke()                        //ฟังก์ชั่น Smoke
129.      {
130.        sensorValue = analogRead(SmokePin);
131.        outputValue = map(sensorValue, 150, 380, 0, 100);
132.        if (sensorValue <= 150 || i != 0) {
133.          if (outputValue <= 0 ) {
134.            outputValue = 0;              //ประกาศ outputValue เท่ากับ 0
135.          }
136.          if (outputValue >= 100) {
137.            outputValue = 100;            //ประกาศ outputValue เท่ากับ 100
138.          }
139.          lcd.setCursor(0, 0);
140.          lcd.print("  Smoke/GAS     ");  //สั่งหน้าจอแสดง Smoke/GAS
141.          lcd.setCursor(0, 1);            //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 2
142.          lcd.print("     Value : ");    //สั่งหน้าจอแสดง  Value :
143.          lcd.print(outputValue);         //สั่งหน้าจอแสดงค่า outputValue
144.          if (outputValue > sensorSet)
```

```
145.            {
146.              lcd.setCursor(1, 0);              //สั่งกำหนดค่าหน้าจอที่ตัวที่ 2 แถวที่ 1
147.              lcd.print("  Warning!");           //สั่งหน้าจอแสดง  Warning!
148.              lcd.setCursor(0, 1);              //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 2
149.              lcd.print(" Smoke/GAS : ");       //สั่งหน้าจอแสดง  Smoke/GAS :
150.              lcd.print(outputValue);           //สั่งหน้าจอแสดงค่า  outputValue
151.
152.              if (CDSmoke <= 0) {
153.                httpRequestsmoke();             //เรียกใช้งานฟังก์ชั่น httpRequestsmoke
154.                delay (1000);                   //รอเวลา 1 วิ
155.                TimerSmoke = 3;                 //กำหนดให้ TimerSmoke เท่ากับ 3 เพื่อ
    ไม่ให้มันทำการแจ้งเตือนซ้ำจะเว้นช่วงให้มันประมาณ 40 วินาที
156.
157.              } else {
158.                TimerSmoke = CDSmoke - 1;
159.              }
160.            }
161.
162.            delay (1000);                       //รอเวลา 1 วิ
163.            i = 1;                              //กำหนดให้ i เท่ากับ 1
164.          }
165.          else {
166.            TimerSmoke = 0;                   //กำหนดให้ TimerSmoke = 0
167.            Serial.print("Failed no ");        //สั่งหน้าจอ Serial Moniter แสดง Failed no
168.            Serial.println(sensorValue);      //สั่งหน้าจอ Serial Moniter แสดง
    ค่า sensorValue
169.          }
170.          CDSmoke = TimerSmoke;
171.        }
172.        //-----------------------CallOutPut----------------------
```

```
173.        void CallOutPut()
174.        {
175.         digitalWrite(CallbuttonPin, HIGH);
176.         delay(2000);
177.         digitalWrite(CallbuttonPin, LOW);
178.          }
179.        //------------------------WallPlug-------------------------
180.        void Wall()                                  //ฟังก์ชั่น Wall
181.        {
182.         buttonState = digitalRead(buttonPin);
183.
184.         if (buttonState == LOW ) {
185.
186.          lcd.setCursor(0, 0);                      //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 1
187.          lcd.print("   Warning!    ");             //สั่งหน้าจอแสดง  Warning!
188.          lcd.setCursor(0, 1);                      //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 2
189.          lcd.print("    Power Fail  ");           //สั่งหน้าจอแสดง   Power Fail
190.          delay (2000);                             //รอเวลา 2 วิ
191.
192.          if (CDWall <= 0) {                        //ถ้าค่า CDWall น้อยกว่าหรือเท่ากับ 0  จะ
    ทำการเรียกใช้ฟังก์ชั่น httpRequestwall เพื่อทำการแจ้งเตือนไปยังไลน์หรือโทรออก
193.            httpRequestwall();                      //เรียกใช้ฟังก์ชั่น httpRequestwall
194.            TimerWall = 3;                          //กำหนดให้ TimerWall เท่ากับ 3 เพื่อไม่ให้
    มันทำการแจ้งเตือนซ้ำจะเว้นช่วงให้มันประมาณ 40 วินาที
195.
196.          } else {                                  //ถ้าค่า CDWall มากกว่า 0 จะ
    กำหนดให้ TimerWall = CDWall - 1 เพื่อให้มันลดค่าไปเรื่อยๆให้เท่ากับ 0 เพื่อทำการแจ้ง
    เตือนอีกครั้ง
197.            TimerWall = CDWall - 1;         //กำหนดให้ TimerWall = CDWall - 1
198.          }
```

```
199.          } else {
200.            TimerWall = 0;          //กำหนดให้  TimerWall = 0
201.          }
202.          CDWall = TimerWall;          //กำหนดให้ CDWall = TimerWall เพื่อรีเซ็ทค่า
    ให้เท่ากับ 0
203.        }
204.        //------------------------Temps-------------------------
205.        void Temps()                    //ฟังก์ชั่น Wall
206.        {
207.          float h = dht.readHumidity();     //ประกาศค่า h เป็น float ให้อ่านค่าความชื้น
    จากคำสั่ง dht.readHumidity();
208.          float t = dht.readTemperature();   //ประกาศค่า t เป็น float ให้อ่านค่าอุณหภูมิ
    จากคำสั่ง dht.readTemperature();
209.
210.          lcd.setCursor(0, 0);              //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 1
211.          lcd.print("Humidity: ");          //สั่งหน้าจอแสดง Humidity:
212.          lcd.print(h);                     //สั่งหน้าจอแสดงค่า h ที่ได้รับจาก sensor
213.          lcd.print("%");                   //สั่งหน้าจอแสดง %
214.          lcd.setCursor(0, 1);              //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 2
215.          lcd.print(" Temp: ");             //สั่งหน้าจอแสดง Temp:
216.          lcd.print(t);                     //สั่งหน้าจอแสดงค่า t ที่ได้รับจาก sensor
217.          lcd.print(" *C");                 //สั่งหน้าจอแสดง *C
218.          delay(5000);                      //รอเวลา 5 วิ
219.        if (h > HMAX) {                     //ถ้าค่า h มากกว่าค่า HMAX จะทำการแจ้ง
    เตือนเมื่อค่าเกินกำหนด
220.          lcd.setCursor(0, 0);              //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 1
221.          lcd.print("   Warning!   ");      //สั่งหน้าจอแสดง Warning!
222.          lcd.setCursor(0, 1);              //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 2
223.          lcd.print("Humidity :");          //สั่งหน้าจอแสดง Humidity:
224.          lcd.print(h);                     //สั่งหน้าจอแสดงค่า h ที่ได้รับจาก sensor
```

```
225.            lcd.print("%");                    //สั่งหน้าจอแสดง %
226.            if (CDhmax <= 0) {                 //ถ้าค่า CDWall น้อยกว่าหรือเท่ากับ 0  จะ
    ทำการเรียกใช้ฟังก์ชั่น httpRequestwall เพื่อทำการแจ้งเตือนไปยังไลน์หรือโทรออก
227.              httpRequesthmax();               //เรียกใช้ฟังก์ชั่น httpRequesthmax
228.              delay(5000);                     //รอเวลา 5 วิ
229.              Timerhmax = 3;                   //กำหนดให้ Timerhmax เท่ากับ 3 เพื่อไม่ให้
    มันทำการแจ้งเตือนซ้ำจะเว้นช่วงให้มันประมาณ 40 วินาที
230.
231.            } else {                           //ถ้าค่า CDhmax มากกว่า 0 จะ
    กำหนดให้ Timerhmax = CDhmax - 1 เพื่อให้มันลดค่าไปเรื่อยๆให้เท่ากับ 0 เพื่อทำการ
    แจ้งเตือนอีกครั้ง
232.              Timerhmax = CDhmax - 1;          //กำหนดให้ Timerhmax = CDhmax - 1
233.            }
234.           } else {                            //ถ้าค่า h น้อยกว่าค่า HMAX จะ
    กำหนดให้ Timerhmax = 0 เพื่อให้ลดการดูลดาวน์แล้วกลับมาแจ้งเตือนอีกครั้ง
235.            Timerhmax = 0;                     //กำหนดให้  Timerhmax = 0
236.           }
237.           if (h < HMIN) {                     //ถ้าค่า h มากกว่าค่า HMIN จะทำการแจ้ง
    เตือนเมื่อค่าเกินกำหนด
238.            lcd.setCursor(0, 0);               //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 1
239.            lcd.print("    Warning!    ");     //สั่งหน้าจอแสดง Warning!
240.            lcd.setCursor(0, 1);               //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 2
241.            lcd.print("Humidity :");           //สั่งหน้าจอแสดง Humidity:
242.            lcd.print(h);                      //สั่งหน้าจอแสดงค่า h ที่ได้รับจาก sensor
243.            lcd.print("%");                    //สั่งหน้าจอแสดง %
244.          if (CDhmin <= 0) {                   //ถ้าค่า CDhmin น้อยกว่าหรือเท่ากับ 0  จะ
    ทำการเรียกใช้ฟังก์ชั่น httpRequesthmin เพื่อทำการแจ้งเตือนไปยังไลน์หรือโทรออก
245.              httpRequesthmin();               //เรียกใช้ฟังก์ชั่น httpRequesthmin
246.              delay(5000);                      //รอเวลา 5 วิ
```

```
247.              Timerhmin = 3;                //กำหนดให้ Timerhmin เท่ากับ 3 เพื่อไม่ให้มัน
    ทำการแจ้งเตือนซ้ำจะเว้นช่วงให้มันประมาณ 40 วินาที
248.            } else {                        //ถ้าค่า CDhmin มากกว่า 0 จะ
    กำหนดให้ Timerhmin = CDhmin - 1 เพื่อให้มันลดค่าไปเรื่อยๆให้เท่ากับ 0 เพื่อทำการแจ้ง
    เตือนอีกครั้ง
249.              Timerhmin = CDhmin - 1;       //กำหนดให้ Timerhmax = CDhmax - 1
250.            }
251.          } else {                          //ถ้าค่า h น้อยกว่าค่า HMAX จะ
    กำหนดให้ Timerhmax = 0 เพื่อให้ลดการดูลดาวน์แล้วกลับมาแจ้งเตือนอีกครั้ง
252.            Timerhmin = 0;                  //กำหนดให้ Timerhmin = 0;
253.          }
254.          if (t > TMAX) {                   //ถ้าค่า t มากกว่าค่า TMAX จะทำการแจ้ง
    เตือนเมื่อค่าเกินกำหนด
255.            lcd.setCursor(0, 0);            //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 1
256.            lcd.print("    Warning!    ");  //สั่งหน้าจอแสดง Warning!
257.            lcd.setCursor(0, 1);            //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 2
258.            lcd.print(" Temp: ");           //สั่งหน้าจอแสดง Temp:
259.            lcd.print(t);                   //สั่งหน้าจอแสดงค่า t ที่ได้รับจาก sensor
260.            lcd.print(" *C ");              //สั่งหน้าจอแสดง *C
261.
262.            if (CDTmax <= 0) {              //ถ้าค่า CDTmax น้อยกว่าหรือเท่ากับ 0  จะทำ
    การเรียกใช้ฟังก์ชั่น httpRequesttmax เพื่อทำการแจ้งเตือนไปยังไลน์หรือโทรออก
263.              httpRequesttmax();            //เรียกใช้ฟังก์ชั่น httpRequesttmax
264.              delay(5000);                  //รอเวลา 5 วิ
265.              TimerTmax = 3;                //กำหนดให้ TimerTmax เท่ากับ 3 เพื่อไม่ให้
    มันทำการแจ้งเตือนซ้ำจะเว้นช่วงให้มันประมาณ 40 วินาที
266.            } else {
267.              TimerTmax = CDTmax - 1;       //กำหนดให้ TimerTmax = CDTmax - 1
268.            }
269.          } else {
```

```
270.            TimerTmax = 0;                    //กำหนดให้ Timerhmin = 0
271.          }
272.          if (t < TMIN) {                  //ถ้าค่า t มากกว่าค่า TMIN จะทำการแจ้งเตือนเมื่อ
    ค่าเกินกำหนด
273.            lcd.setCursor(0, 0);          //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 1
274.            lcd.print("   Warning!   ");      //สั่งหน้าจอแสดง Warning!
275.            lcd.setCursor(0, 1);            //สั่งกำหนดค่าหน้าจอที่ตัวที่ 1 แถวที่ 2
276.            lcd.print(" Temp: ");            //สั่งหน้าจอแสดง Temp:
277.            lcd.print(t);                    //สั่งหน้าจอแสดงค่า t ที่ได้รับจาก sensor
278.            lcd.print(" *C ");               //สั่งหน้าจอแสดง *C
279.
280.            if (CDTmin <= 0) {              //ถ้าค่า CDTmin น้อยกว่าหรือเท่ากับ 0  จะทำ
    การเรียกใช้ฟังก์ชั่น httpRequesttmax เพื่อทำการแจ้งเตือนไปยังไลน์หรือโทรออก
281.              httpRequesttmin();            //เรียกใช้ฟังก์ชั่น httpRequesttmin
282.              delay(5000);                  //รอเวลา 5 วิ
283.              TimerTmin = 3;                //กำหนดให้ TimerTmin เท่ากับ 3 เพื่อไม่ให้มัน
    ทำการแจ้งเตือนซ้ำจะเว้นช่วงให้มันประมาณ 40 วินาที
284.            } else {
285.              TimerTmin = CDTmin - 1;       //กำหนดให้ TimerTmin = CDTmin - 1
286.            }
287.          } else {
288.            TimerTmin = 0;                  //กำหนดให้ TimerTmin = 0
289.          }
290.         CDhmax = Timerhmax;
291.          CDhmin = Timerhmin;
292.          CDTmax = TimerTmax;
293.          CDTmin = TimerTmin;
294.         }
295.         //-------------------------http-wallplug-------------------------
296.         void httpRequestwall() {          //ฟังก์ชั่น httpRequestwall
```

```
297.            client.stop();
298.            if (client.connect(server, 80)) {
299.              Serial.println("wallplug--connecting...");
300.              client.print("GET /~aline/index.php?token=");
301.              client.print(token);
302.              client.print("&mes=");
303.              client.println(meswall);
304.              client.println(" HTTP/1.1");
305.              client.println("Host: www.arduino.cc");
306.              client.println("User-Agent: arduino-ethernet");
307.              client.println("Connection: close");
308.              client.println();
309.              lastConnectionTime = millis();
310.            } else {
311.              Serial.println("connection failed");
312.              Serial.println("Calling .. ");
313.              CallOutPut();
314.              Serial.println("Called");
315.            }
316.          }
317.          //------------------------http-smoke-------------------------
318.          void httpRequestsmoke() {            //ฟังก์ชั่น httpRequestsmoke
319.            client.stop();
320.            if (client.connect(server, 80)) {     //ถ้า client ทำการเชื่อมต่ออินเตอร์เน็ทได้จะ
    ทำการเปลี่ยน Token ที่ได้รับมาเพื่อทำการแจ้งเตือนไปยังไลน์
321.              Serial.println("smoke--connecting...");
322.              client.print("GET /~aline/index.php?token=");
323.              client.print(token);
324.              client.print("&mes=");
325.              client.println(messmoke);
```

```
326.              client.println(" HTTP/1.1");
327.              client.println("Host: www.arduino.cc");
328.              client.println("User-Agent: arduino-ethernet");
329.              client.println("Connection: close");
330.              client.println();
331.              lastConnectionTime = millis();
332.            } else {
333.              Serial.println("smoke-connection failed");
334.              Serial.println("Calling .. ");
335.              CallOutPut();
336.              Serial.println("Called");
337.             }
338.          }
339.          //-------------------------http-temperatureMin-------------------------
340.          void httpRequesttmin() {              //ฟังก์ชั่น httpRequesttmin
341.            client.stop();
342.            if (client.connect(server, 80)) {      //ถ้า client ทำการเชื่อมต่ออินเตอร์เน็ทได้จะ
    ทำการเปลี่ยน Token ที่ได้รับมาเพื่อทำการแจ้งเตือนไปยังไลน์
343.              Serial.println("temperatureMin--connecting...");
344.              digitalWrite(CallbuttonPin, LOW);
345.              client.print("GET /~aline/index.php?token=");
346.              client.print(token);
347.              client.print("&mes=");
348.              client.println(mestmin);
349.              client.println(" HTTP/1.1");
350.              client.println("Host: www.arduino.cc");
351.              client.println("User-Agent: arduino-ethernet");
352.              client.println("Connection: close");
353.              client.println();
354.              lastConnectionTime = millis();
```

```
355.          } else {
356.            Serial.println("temperatureMin--connection failed");
357.            Serial.println("Calling .. ");       //สั่งหน้าจอ Serial Moniter แสดง Calling ..
358.            CallOutPut();
359.            Serial.println("Called");           //สั่งหน้าจอ Serial Moniter แสดง Calling ..
360.          }
361.        }
362.        //------------------------http-temperatureMax-----------------------
363.        void httpRequesttmax() {          //ฟังก์ชั่น httpRequesttmax
364.          client.stop();
365.          if (client.connect(server, 80)) {
366.            Serial.println("temperatureMax--connecting...");
367.            digitalWrite(CallbuttonPin, LOW);
368.            client.print("GET /~aline/index.php?token=");
369.            client.print(token);
370.            client.print("&mes=");
371.            client.println(mestmax);
372.            client.println(" HTTP/1.1");
373.            client.println("Host: www.arduino.cc");
374.            client.println("User-Agent: arduino-ethernet");
375.            client.println("Connection: close");
376.            client.println();
377.            lastConnectionTime = millis();
378.          } else {
379.            Serial.println("temperatureMax-connection failed");
380.            Serial.println("Calling .. ");
381.            CallOutPut();
382.            Serial.println("Called");
383.            }
384.        }
```

```
385.          //------------------------http-humidityMin-------------------------
386.          void httpRequesthmin() {
387.            client.stop();
388.            if (client.connect(server, 80)) {
389.              Serial.println("humidityMin--connecting...");
390.              digitalWrite(CallbuttonPin, LOW);
391.              client.print("GET /~aline/index.php?token=");
392.              client.print(token);
393.              client.print("&mes=");
394.              client.println(meshmin);
395.              client.println(" HTTP/1.1");
396.              client.println("Host: www.arduino.cc");
397.              client.println("User-Agent: arduino-ethernet");
398.              client.println("humidityMin-Connection: close");
399.              client.println();
400.              lastConnectionTime = millis();
401.            } else {
402.              Serial.println("connection failed");
403.              Serial.println("Calling .. ");
404.              CallOutPut();
405.              Serial.println("Called");
406.              }
407.          }
408.
409.          //------------------------http-humidityMax-------------------------
410.          void httpRequesthmax() {              //ฟังก์ชั่น httpRequesthmax
411.            client.stop();
412.            if (client.connect(server, 80)) {
413.              Serial.println("humidityMax--connecting...");
414.              digitalWrite(CallbuttonPin, LOW);
```

```
415.            client.print("GET /~aline/index.php?token=");
416.            client.print(token);
417.            client.print("&mes=");
418.            client.println(meshmax);
419.            client.println(" HTTP/1.1");
420.            client.println("Host: www.arduino.cc");
421.            client.println("User-Agent: arduino-ethernet");
422.            client.println("Connection: close");
423.            client.println();
424.            lastConnectionTime = millis();
425.          } else {
426.            Serial.println("humidityMax-connection failed");
427.            Serial.println("Calling .. ");
428.            CallOutPut();
429.            Serial.println("Called");
430.             }
431.        }
432.        void softReset() {
433.          asm volatile ("  jmp 0");
434.        }
```

มหาวิทยาลัยราชภัฏสงขลา SONGKHLA RAJABHAT UNIVERSITY

โค้ดสำหรับอัปโหลดเข้าบอร์ด Arduino Nano

```
1.  #include "SIM900.h"                //ดึงไลบรารี SIM900.h มาใช้
2.  #include "call.h"                  //ดึงไลบรารี call.h มาใช้
3.  #include <SoftwareSerial.h>        //ดึงไลบรารี SoftwareSerial.h มาใช้
4.  CallGSM call;                       //กำหนดให้ CallGSM เป็นชื่อ call
5.  boolean One_Round = false;          //ประกาศ One_Round เป็น boolean มีค่าเท่ากับ false
6.
7.  char number[] ="+66824281908";          //กำหนดเบอร์ที่ต้องการจะให้โทร
8.  const int buttonPin = 5;
9.  int buttonState = 0;
10.
11. int CDCall = 0;                                 //ประกาศค่า CDWall เป็น int เท่ากับ 0
12. int TimerCall = 3;
13.
14. void setup()
15. {
16.   Serial.begin(9600);                           //สั่งให้ Serial เริ่มทำงาน
17.     if (gsm.begin(9600))                        //ถ้า GSM เริ่มทำงานให้เช็ค
    ว่า Sim900a ตอบสนองไหม ถ้าใช่ให้แสดงว่า READY ถ้าไม่ แสดงว่า IDLE
18.         Serial.println("\nstatus=READY");
19.     else Serial.println("\nstatus=IDLE");        //สั่งหน้าจอ Serial Moniter แสดง IDLE
20.       pinMode(buttonPin, INPUT);
21. }
22. void loop()
23. {
24.    buttonState = digitalRead(buttonPin);
25.    Serial.print("digitalRead...");
26.
27.  if (buttonState == HIGH) {                     //เปลียนเป็น HIGH จะโทรออก
```

```
28.
29.     if (CDCall <= 0) {                  //ถ้าค่า CDWall น้อยกว่าหรือเท่ากับ 0  จะทำการ
    เรียกใช้ฟังก์ชั่น httpRequestwall เพื่อทำการแจ้งเตือนไปยังไลน์หรือโทรออก
30.       Serial.println("Calling...");
31.       call.Call(number);                //ทำการโทรออกไปยังเบอร์ที่กำหนด
32.       TimerCall = 10;                   //กำหนดให้ TimerWall เท่ากับ 3 เพื่อไม่ให้มันทำการ
    แจ้งเตือนซ้ำจะเว้นช่วงให้มันประมาณ 40 วินาที
33.
34.     } else {                            //ถ้าค่า CDWall มากกว่า 0 จะ
    กำหนดให้ TimerWall = CDWall - 1 เพื่อให้มันลดค่าไปเรื่อยๆให้เท่ากับ 0 เพื่อทำการแจ้ง
    เตือนอีกครั้ง
35.       TimerCall = CDCall - 1;           //กำหนดให้ TimerWall = CDWall - 1
36.     }
37.   }
38.   else{
39.     Serial.println("Is LOW DON'T CALL");
40.     TimerCall = 0;
41.     }
42.     CDCall = TimerCall;
43. }
```

มหาวิทยาลัยราชภัฏสงขลา SONGKHLA RAJABHAT UNIVERSITY

ภาคผนวก ค

ประวัติผู้จัดทำโครงงาน

ประวัติผู้จัดทำโครงงาน

ชื่อ- สกุล นางซะห์ลัน เหสามี

วันเดือนปี เกิด 1 กรกฎาคม 2538

ที่อยู่ 1 หมู่ 1 ตำบลปะลุรู อำเภอสุไหงปาดี จังหวัดนราธิวาส 96140

โทรศัพท์ 0-8242-81908

อีเมล์ sahlanhaysamee@gmail.com

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนบ้านโคกตา อ.สุไหงปาดี จ.นราธิวาส
มัธยมศึกษาตอนต้น	โรงเรียนอัตตัรกียะห์อิสลามียะห์ อ.เมือง จ.นราธิวาส
มัธยมศึกษาตอนปลาย	โรงเรียนอัตตัรกียะห์อิสลามียะห์ อ.เมือง จ.นราธิวาส
ปริญญาตรี	โปรแกรมวิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสงขลา อำเภอเมืองสงขลา จังหวัดสงขลา

มหาวิทยาลัยราชภัฏสงขลา SONGKHLA RAJABHAT UNIVERSITY

ประวัติผู้จัดทำโครงงาน

ชื่อ- สกุล นางรุสลัน หะมะ

วันเดือนปี เกิด 29 มีนาคม 2539

ที่อยู่ 48/1 หมู่ 1 บ้านป่าหวัง ตำบลบันนังสตา อำเภอบันนังสตา จังหวัดยะลา 95130

โทรศัพท์ 0-8847-28443

อีเมล์ 9ruslan.hama@gmail.com

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนบ้านป่าหวัง อ.บันนังสตา จ.ยะลา
มัธยมศึกษาตอนต้น	โรงเรียนดำรงวิทยา อ.บันนังสตา จ.ยะลา
มัธยมศึกษาตอนปลาย	โรงเรียนดำรงวิทยา อ.บันนังสตา จ.ยะลา
ปริญญาตรี	โปรแกรมวิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสงขลา อำเภอเมืองสงขลา จังหวัดสงขลา

มหาวิทยาลัยราชภัฏสงขลา SONGKHLA RAJABHAT UNIVERSITY

มหาวิทยาลัยราชภัฏสงขลา
SONGKHLA RAJABHAT UNIVERSITY