



รายงานสหกิจศึกษาบับสมบูรณ์

เครื่องชั่งน้ำหนักเปิดอัตโนมัติ

Automatic Duck weighting machine

นายจิรภาส ไชยเนตร

สาขาวิชาระบบอุตสาหกรรมเกษตร

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจ	เครื่องชั่งน้ำหนักเปิดอัตโนมัติ
นักศึกษาผู้จัดทำ	นายจิรภาส ไชยเนตร
คณะ วิศวกรรมศาสตร์	ภาควิชาวิศวกรรมเกษตร
ชื่อ-สกุล อาจารย์ปรึกษา	ผศ.ดร.ประสันต์ ชุ่มใจหาญ
ชื่อ-สกุล ผู้นิเทศงาน	นายอุดมศักดิ์ โชติช่วง
สถานประกอบการ	บริษัท เอวิลอน อินโนเวชั่น จำกัด

### บทคัดย่อ

โครงการนี้เป็นการออกแบบเครื่องชั่งน้ำหนักเปิด เพื่อเพิ่มผลผลิตและลดเวลาการทำงานในการชั่งน้ำหนักเปิด จากการศึกษาค้นคว้าที่เกี่ยวข้องกับการออกแบบด้วย Solidwork และการเขียนโปรแกรมเขียนด้วยโปรแกรม Arduino IDE ซึ่งส่วนประกอบที่สำคัญของเครื่องมีดังนี้ 1) โครงสร้างของเครื่อง 2) เซ็นเซอร์โหลดเซลล์ 3) โมดูลขยายสัญญาณโหลดเซลล์ และ 4) ESP 8266 ในการทำงานของเครื่อง เซ็นเซอร์โหลดเซลล์จะรับค่าน้ำหนักส่งสัญญาณไปยังโมดูลและโมดูลจะขยายสัญญาณส่งให้ ESP 8266 และเชื่อมต่อสัญญาณ Wifi เพื่อส่งและข้อมูลไว้บน Cloud แล้วแสดงค่าผ่าน FEEDs ของ NETPEI ซึ่งประหยัดเวลาในการชั่งเปิด และสามารถติดตามการเจริญเติบโตของเปิดในแต่ละโรงเรือนได้แบบเรียลไทม์ ความแม่นยำไม่เกิน 0.05

คำสำคัญ : เครื่องชั่งเปิด, เซ็นเซอร์โหลดเซลล์, โมดูลขยายสัญญาณโหลดเซลล์, esp8266 และ cloud

Cooperative Title : Duck scales

Student Intern name : Mr.Jirapad chaiyanate

Faculty : Engineer Department : Agricultural

Advisor name : Asst.Prof.Dr. Prasan Choomjaihan

Mentor name : Mr.Udomsak Chotchung

Company : Avilon Innovation Co., Ltd

## ABSTRACT

This cooperative education project is to Duck scales design. Increase productivity and reduce working time in duck weighing. From studies related to design with Solidwork and programming using Arduino IDE. The key components of the machine are as follows: 1) the structure of the machine, 2) the load cell sensor, 3) load cell amplifiers and 4) ESP 8266. In the operation of the machine The load cell sensor receives the weight, sends the signal to the module, and the module expands the signal to ESP 8266 and connects the Wifi signal to send and data on the Cloud. Show values via NETPEI FEEDs which save time when weighing ducks. And can track the growth of ducks in each house in real time.

**Keywords:** duck scales, load cell sensors, load cell amplifiers, esp8266 and cloud

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

การที่ข้าพเจ้าได้มาปฏิบัติงานสหกิจศึกษา ณ บริษัท เวิร์ดออนไลน์ จำกัด ตั้งแต่วันที่ 5 เดือน สิงหาคม พ.ศ. 2562 ถึงวันที่ 21 เดือนพฤศจิกายน พ.ศ. 2562 ส่งผลให้ข้าพเจ้าได้รับความรู้และประสบการณ์ต่าง ๆ มากมาย สำหรับรายงานวิชาสหกิจศึกษาระดับนี้สำเร็จลุล่วงไปได้ด้วยดี เกิดจากความร่วมมือและสนับสนุนจากหลายฝ่าย ดังนี้

1. นายอุดมศักดิ์ โชติช่วง                      พี่เลี้ยงนักศึกษาฝึกงาน
2. ผศ.ดร.ประสันต์ ชุ่มใจหาญ                    อาจารย์ที่ปรึกษา

และบุคคลท่านอื่นๆ ที่ไม่ได้กล่าวชื่อนามทุกท่านที่ได้ให้คำแนะนำช่วยเหลือในการจัดทำรายงาน

ข้าพเจ้าใคร่ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่าน ที่มีส่วนร่วมในการให้ข้อมูล

เป็นที่ปรึกษาในการทำรายงานฉบับนี้จนเสร็จสมบูรณ์

ตลอดจนให้ทราบดูแลและให้ความเข้าใจเกี่ยวกับชีวิตของการทำงานจริง ข้าพเจ้าขอขอบพระคุณไว้ ณ ที่นี้

นายจิรภาส                      ไชยเนตร  
ผู้จัดทำ

# สารบัญ

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูปภาพ.....	VIII
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญ .....	1
1.2 วัตถุประสงค์ .....	1
1.3 ขอบเขต.....	1
1.4 วิธีดำเนิน.....	1
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 แนวคิด ทฤษฎี.....	3
2.1 การทำฟาร์มเลี้ยงเป็ด.....	3
2.2 การออกแบบเครื่องชั่งน้ำหนักเป็ดโดยใช้โปรแกรม.....	4
2.3 NETPIE.....	5
2.3.1 ประโยชน์ของ NETPIE.....	6
2.4 MICROGEAR .....	7
2.5 NETPIE FEED.....	9
2.6 Load Cell .....	12
บทที่ 3 ขั้นตอนการดำเนินงาน .....	14
3.1 ศึกษาปัญหาเพื่อทำโครงการ.....	14
3.2 กำหนดที่มาและความสำคัญ, วัตถุประสงค์, ขอบเขต, แผนการดำเนินงาน.....	14
3.3 ศึกษาและวิเคราะห์ปัญหา.....	14
3.3.1 ศึกษาปัญหา ที่มาของโครงการ.....	14
3.3.2 วิเคราะห์ปัญหา.....	15
3.4. ศึกษาข้อมูลเพื่อพัฒนาเครื่องชั่งน้ำหนักเป็ด.....	15

## สารบัญ (ต่อ)

3.4.1 การพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์.....	15
3.4.2 ระบบ Cloud ที่เลือกใช้.....	15
3.5 อุปกรณ์.....	16
3.5.1 เซ็นเซอร์ที่เลือกใช้.....	16
3.6 ขั้นตอนการออกแบบ.....	17
3.7 การเขียนโปรแกรม.....	18
3.7.1 Calibration Load Cell.....	18
3.8 การออกแบบวงจรควบคุม.....	23
บทที่ 4 สรุปผลการดำเนินงาน.....	24
4.1 สรุปผลการดำเนินงาน.....	24
4.1.1 การชั่งน้ำหนักของเบ็ด.....	24
4.1.2 การส่งข้อมูลไปยัง NETPIE.....	24
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....	25
5.1 สรุปผลการดำเนินการ.....	25
5.2 ข้อเสนอแนะ.....	25
บรรณานุกรม.....	26
ภาคผนวก ก.....	27

## สารบัญตาราง

ตาราง 3.1 แผนการดำเนินงาน .....	14
---------------------------------	----



## สารบัญรูปภาพ

รูปที่ 2.1	แบบเครื่องชั่งน้ำหนักเบ็ด สามมิติ.....	4
รูปที่ 2.2	NETPIE.....	5
รูปที่ 2.3	ลำดับการเรียกฟังก์ชันพื้นฐานของ Microgear.....	8
รูปที่ 2.4	การทำงาน Load cell.....	12
รูปที่ 2.5	วิธีติดตั้ง Load Cell แบบ Straight Bar เข้ากับแผ่นชั่งน้ำหนัก.....	12
รูปที่ 2.6	การติดตั้ง load cell เข้ากับแผ่นตัวอย่าง.....	13
รูปที่ 3.1	แผนภูมิการปลาวิเคราะห์ปัญหา .....	15
รูปที่ 3.2	Node MCU-12E และผังคำอธิบายขาต่อ.....	16
รูปที่ 3.3	Load cell.....	17
รูปที่ 3.4	HX711.....	18
รูปที่ 3.5	แบบเครื่องชั่งน้ำหนักเบ็ด สามมิติ.....	19
รูปที่ 3.6	ตุ้มน้ำหนักมาตรฐาน.....	19
รูปที่ 3.7	คำสั่งกำหนดค่าน้ำหนักอ้างอิง.....	19
รูปที่ 3.8	การตั้งค่า Baud rate.....	20
รูปที่ 3.9	การส่งค่า a รูปแบบ command.....	21
รูปที่ 3.10	แสดงค่า Zero Factor.....	21
รูปที่ 3.11	ตัวอย่างวัตถุในการชั่งน้ำหนัก.....	22
รูปที่ 3.12	การส่งค่า b รูปแบบ command.....	22
รูปที่ 3.13	ตัวอย่างการหาค่า calibration factor.....	23
รูปที่ 3.12	การส่งค่า c รูปแบบ command.....	23
รูปที่ 3.13	ตัวอย่าง การนำค่า Zero Factor และ calibration factor ไปใช้งาน.....	24
รูปที่ 3.14	ตัวอย่างการแสดงค่าลูกตุ้มน้ำหนักรวม 720 กรัม.....	25
รูปที่ 3.15	วงจรควบคุมสำหรับเครื่องชั่งเบ็ด.....	25
รูปที่ 4.1	แสดงค่าผ่าน FEEDs ของ NETPIE.....	26
รูปที่ ก 1	Specification Load cell.....	27

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญ

โครงการนี้จัดทำขึ้นเพื่อพัฒนาเครื่องชั่งน้ำหนักเปิด โดยมี บริษัท เอวิลอน อินโนเวชั่น จำกัด ในการเข้ามามีส่วนร่วมในการแก้ไขปัญหาต่าง ๆ ในระหว่างการดำเนินโครงการ ซึ่งเป็นบริษัทที่พัฒนาเทคโนโลยีในด้าน IoT เกี่ยวกับการเกษตร และสำรวจพื้นที่การเกษตร เพื่อตอบสนองการเกษตรยุค 4.0 ซึ่งช่วยในเรื่องการประหยัดเวลา ต้นทุนการผลิต และเพื่อเพิ่มผลกำไรให้กับเกษตรกร

ในการชั่งน้ำหนักเปิดในฟาร์ม จำเป็นต้องทำส้อมเปิดมาทำการวัดน้ำหนักเพื่อจดบันทึกการเจริญเติบโตของเปิดในแต่ละโรงเรือน ซึ่งในการตรวจวัดในแต่ละครั้งต้องใช้เวลาในการตรวจวัดเป็นเวลานาน และยังเป็นการทำลายพฤติกรรมตามธรรมชาติของเปิด

เทคโนโลยีในด้าน IoT สามารถช่วยในอำนวยความสะดวกการตรวจวัดน้ำหนักเปิด เช่น ด้านเวลา และยังสามารถตรวจวัดน้ำหนักเปิดได้ทั้งโรงเรือน จากการสำรวจ จึงได้มีแนวคิดในการติดตั้งเครื่องตรวจวัดเป็นบริเวณทางเดินที่เปิดจะเดินไปกินน้ำ เพื่อให้มั่นใจว่า เปิดทุกตัวจะได้ทำการชั่งน้ำหนัก โดยน้ำหนักที่ได้จากการตรวจวัด จะนำค่าน้ำหนักมาเก็บไว้บน Cloud และแสดงผลผ่านทาง Web server เพื่อผู้ประกอบการสามารถติดตามการเจริญเติบโตของเปิดในโรงเรือนได้แบบเรียลไทม์

### 1.2 วัตถุประสงค์ของการทดลอง

1. พัฒนาเครื่องชั่งน้ำหนักสำหรับเปิดแบบเดินผ่าน
2. ส่งข้อมูลไปเก็บไว้บน Cloud

### 1.3 ขอบเขตของการทดสอบ

1. ศึกษาและออกแบบเครื่องชั่งน้ำหนักเปิด
2. พัฒนาเครื่องชั่งเปิดร่วมกับเทคโนโลยี IoT เพื่อสะดวกต่อการใช้งาน

### 1.4 วิธีการดำเนิน

1. ศึกษาปัญหาเพื่อทำโครงการ
2. กำหนดที่มาและความสำคัญ, วัตถุประสงค์, ขอบเขต, แผนการดำเนินงาน
3. ศึกษาและวิเคราะห์ของปัญหา
4. ออกแบบเครื่อง
5. ดำเนินการตามแผนที่วางไว้
6. ตรวจสอบผลลัพธ์และปรับปรุงเครื่องชั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 7.สรุปผลโครงการ

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.สามารถส่งข้อมูลไปเก็บไว้ใน Cloud
- 2.ได้ชุดตัวอย่างในการชั่งน้ำหนักเปิดแบบอัตโนมัติรายตัวแบบแม่นยำ



## บทที่ 2

### แนวคิด ทฤษฎี

การซึ่กน้ำหนักเปิดในปัจจุบันมีการซึ่กแบบเครื่องซึ่กสปริงแบบดิจิทัลแต่ไม่สามารถส่งข้อมูลเก็บไปไว้ใน Cloud ได้ ทำให้มีอุปสรรคในการเก็บข้อมูลการเจริญเติบโตของเปิด

วิธีการซึ่กโดยเปิดจะเดินผ่านเครื่องซึ่กและเครื่องซึ่กจะส่งข้อมูลน้ำหนักเปิดขึ้นไปยัง NETPIE แสดงผลบน FEEDS และสามารถนำข้อมูลใน FEEDS มาทำเป็นกราฟเพื่อหาค่าเฉลี่ยของน้ำหนักเปิดได้มา การแก้ปัญหา โดยใช้เครื่องซึ่กอัตโนมัติ แบบซึ่กรายตัวระหว่างเปิดเดิน พร้อมส่งสัญญาณผ่าน Netpie

#### 2.1 การทำฟาร์มเลี้ยงเปิด

เปิดเป็นสัตว์ปีกที่เลี้ยงกันแพร่หลายทั่วไปในประเทศไทย ยังไม่มีหลักฐานว่าการเลี้ยง เปิดในประเทศไทยเริ่มต้นตั้งแต่เมื่อใด แต่คาดกันว่า ผู้ที่นำเปิดเข้ามาเลี้ยงเป็นครั้งแรกคงจะเป็นคนจีน เพราะคนจีนนิยมบริโภคทั้งเนื้อเปิดและไข่เปิดกันมากกว่าคนไทย คนจีนรู้วิธีทำไข่เปิด เค็มและไข่เยี่ยวม้า ซึ่งคนไทยสมัยก่อนทำไม่เป็น การเลี้ยงเปิดพันธุ์และการฟักไข่เปิดเป็น อาชีพที่คนจีนมีความเชี่ยวชาญเป็นพิเศษ โดยนำเอาวิธีการจากประเทศจีน แม้จะเป็นวิธีการที่ใช้กันมานานับพันปี แต่ก็ยังใช้ได้ผลกันอยู่จนปัจจุบัน

วิวัฒนาการของการเลี้ยงเปิด เมื่อเปรียบเทียบกับวิวัฒนาการของการเลี้ยงไก่แล้ว จะเห็นว่า การเลี้ยงเปิดก้าวหน้าอย่างเชื่องช้ามาก ทั้งนี้เพราะไม่มีการส่งเสริมกันอย่างจริงจัง เหมือนการเลี้ยงไก่

การเลี้ยงเปิด ถ้าจะทำเป็นอุตสาหกรรมเช่นเดียวกับไก่ก็อาจทำได้ ทั้งนี้เพราะราคา ไข่เปิดมักจะดีกว่าไข่ไก่ ในด้านการใช้ไข่ประกอบอาหารหรือทำขนม ไข่เปิดก็ยังได้รับความนิยม จากผู้บริโภคสูงกว่าไข่ไก่ แม้ว่าราคาตัวเปิดเมื่อหมดอายุการไข่จะต่ำกว่าไก่ แต่ก็แตกต่างกันไม่มากนัก

โดยเหตุที่ทั้งไข่เปิดและเนื้อเปิดสามารถนำไปประกอบอาหารได้อย่างดีนั่นเอง จึง ทำให้ความต้องการของตลาดทั่วไปมีอยู่เป็นประจำ และทวีปริมาณขึ้นทุกวันตามจำนวนประชากร ของประเทศที่เพิ่มขึ้น ขณะนี้ไม่เฉพาะแต่จังหวัดภาคกลางและภาคใต้เท่านั้นที่เลี้ยงเปิดกันเป็น จำนวนมาก ในภาคตะวันออกเฉียงเหนือ และภาคเหนือก็มีการเลี้ยงเปิดเป็นจำนวนมากด้วยเหมือนกัน แม้การเลี้ยงเปิดในประเทศในประเทศจะมีจำนวนเพิ่มขึ้นมากกว่าแต่ก่อนอย่างมากมายก็ตาม แต่ก็ยังไม่ปรากฏว่าทำให้การเลี้ยงเปิดต้องหยุดชะงัก เพราะไม่มีตลาด

เปิดเป็นสัตว์เลี้ยงง่าย โรคของเปิดมีไม่มากเหมือนไก่ โรคที่นับว่าร้ายแรงมากของ เปิดไข่ ได้แก่ โรคคอหิวาต์และโรคเพล็ก (PLAQUE) ซึ่งปัจจุบันทางกรมปศุสัตว์ก็ได้ผลิตวัคซีน ป้องกันโรคคอหิวาต์และโรคเพล็กขึ้นไว้สำหรับช่วยเหลือเจ้าของฟาร์มเปิดอยู่แล้ว ทำให้การเลี้ยง เปิดมีความปลอดภัยยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การเลี้ยงเป็ดในประเทศไทย

การเลี้ยงเป็ดในประเทศไทยในปัจจุบันมีการเลี้ยงเป็น 3 แบบคือ

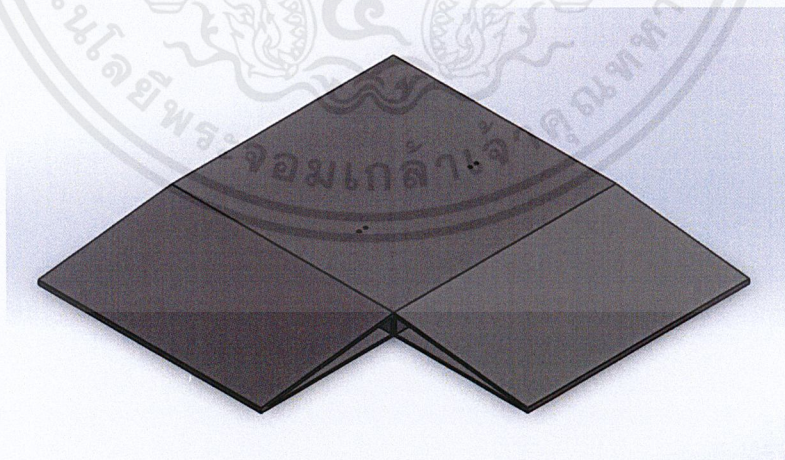
1. การเลี้ยงเป็ดแบบเป็นการค้า จะจัดทำเป็นฟาร์มขนาดใหญ่ มีเป็นจำนวน 10,000ตัวขึ้นไป เป็นการเลี้ยงไว้ในคอกภายในโรงเรือนขนาดใหญ่แบบการเลี้ยงไก่ ไม่มีลานหรือสระน้ำ ให้เป็ดลง แต่มีรางหรือภาชนะใส่อาหารและน้ำไว้ให้เป็ดกินตลอดเวลา มีรั้วสำหรับให้เป็ดเข้า ไซในตอนเช้ามีดี

2. การเลี้ยงเป็ดแบบถึงการค้า เป็นการเลี้ยงเป็ดแบบฟาร์มเป็ดขนาดใหญ่ แต่มี จำนวนเป็ดน้อยกว่า มีโรงเรือนให้เป็ดอยู่แบบง่าย ๆ มีลานปล่อยเป็ด ให้น้ำและอาหารในลาน มีบ่อให้เป็ดลงไปเล่นน้ำ การเลี้ยงแบบนี้เป็ดจะได้อาหารธรรมชาติจากบ่อและตามขอบบ่อเป็น อาหารเสริมด้วย แต่ต้องใช้เนื้อที่มาก

3. การเลี้ยงเป็ดแบบหลังบ้าน เป็นการเลี้ยงจำนวนน้อยคล้ายเป็นงานอดิเรก โดย มีจุดประสงค์ให้มีไข่ไว้พอบริโภคภายในครอบครัวเท่านั้น การให้อาหารจะให้เท่าที่มีเศษอาหาร เหลือ เช่น เศษผัก เศษปลา เนื้อ แล้วปล่อยเป็ดออกไปหาอาหารกินเองในแหล่งน้ำธรรมชาติ หรือในบ่อเป็นบางเวลาหรือตลอดทั้งวัน บางแห่งอาจเลี้ยงรวมกับการเลี้ยงปลาในบ่อ

### 2.2 การออกแบบเครื่องชั่งน้ำหนักเป็ดโดยใช้โปรแกรม

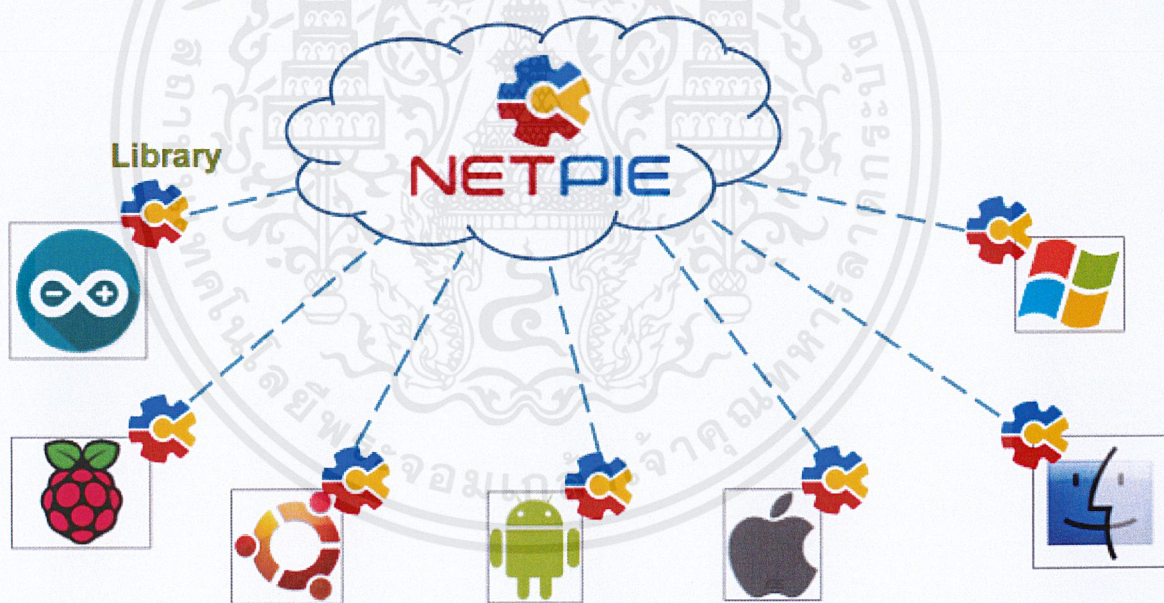
โครงสร้างถูกออกแบบจากโปรแกรม Solid works ใช้หลักการคือการนำเอาขนาดของทางเดินของเป็ด ซึ่งเครื่องชั่งน้ำหนักมีความกว้าง 30 เซนติเมตร มีความยาว 30 เซนติเมตร และมีความสูง 7 เซนติเมตร



รูปที่ 2.1 แบบเครื่องชั่งน้ำหนักเป็ด สามมิติ

## 2.3 NETPIE

NETPIE NETPIE เป็น IoT (Internet of Things) Cloud Platform ที่พัฒนาขึ้นโดยทีมงานวิจัยและเปิดให้บุคคลทั่วไปใช้งานโดยมี Web Portal ที่ให้สามารถลงทะเบียนและจัดการตัวตนและสิทธิ์ของแอปพลิเคชันและอุปกรณ์ได้ที่เว็บไซต์ <https://netpie.io> ตั้งแต่เดือนกันยายน 2558 เป็นต้นมา NETPIE เป็น Middleware ที่มีหัวใจหลัก (นอกเหนือจากส่วนอื่นๆ) เป็น Distributed MQTT Brokers ซึ่งเป็นเสมือนจุดนัดพบให้สิ่งต่างๆ (Things) มาติดต่อสื่อสารและทำงานร่วมกันผ่านวิธีการส่งข้อความแบบ Publish/Subscribe NETPIE มีโครงสร้างสถาปัตยกรรมเป็นคลาวด์อย่างแท้จริงในทุกองค์ประกอบ ทำให้สามารถขยายตัวได้อย่างอัตโนมัติ (Auto-scale) สามารถดูแลและซ่อมแซมตัวเองได้อัตโนมัติเมื่อส่วนหนึ่งส่วนใดในระบบมีปัญหา (Self-healing, Self-recovery) โดยไม่ต้องพึ่งผู้ดูแลระบบ การบริหารจัดการระบบ เป็นแบบ Plug-and-Play ไม่ต้อง Configure หรือปรับแต่ง ในฝั่งอุปกรณ์ NETPIE มี Client Library หรือที่ เรียกว่า Microgear ซึ่งทำหน้าที่สร้างและดูแลช่องทางสื่อสารระหว่างอุปกรณ์กับ NETPIE รวมไปถึงรักษา ความปลอดภัยในการส่งข้อมูล Microgear เป็น Open Source และสามารถดาวน์โหลดได้



จาก <https://github.com/netpieio> โดย

รูปที่ 2.2 NETPIE

ณ ปัจจุบันมี Microgear สำหรับ OS และ Embedded Board หลักๆ ที่ เป็นที่นิยมในหมู่นักพัฒนาเกือบทุกชนิด โมเดลการสื่อสารของ NETPIE แสดงไว้ในรูปที่ 2.2

## ประโยชน์ของ NETPIE

1. ช่วยลดการใช้ทรัพยากรของการเชื่อมต่อ NETPIE ช่วยให้อุปกรณ์สามารถสื่อสารกันได้ โดยผู้ใช้ไม่ต้องกังวลว่า อุปกรณ์นั้นจะอยู่ที่ใด เพียงแค่ นำ Microgear Library ไปติดตั้งในอุปกรณ์ NETPIE จะรับหน้าที่ดูแลเชื่อมต่อให้ทั้งหมด ไม่ว่าจะอุปกรณ์นั้น จะอยู่ในเครือข่ายชนิดใด ลักษณะใด หรือแม้กระทั่ง เคลื่อนย้ายไปอยู่ที่ใด ผู้ใช้สามารถตัดปัญหาในการ เข้าถึงอุปกรณ์จากระยะไกล (Remote Access) ด้วยวิธีการแบบเดิมๆ เช่น การใช้ Fixed Public IP Address หรือการตั้ง Port Forwarding ใน เราท์เตอร์และ การต้องไปลงทะเบียนกับผู้ให้บริการ Dynamic DNS ซึ่งทั้งหมดล้วนมีความยุ่งยาก ลดความยืดหยุ่นของระบบ ไม่เพียงเท่านั้น NETPIE ยังช่วยให้การ เริ่มต้นใช้งานเป็นไปโดยง่าย โดยออกแบบให้อุปกรณ์ถูก ค้นพบและเข้าสู่บริการโดยอัตโนมัติ (Automatic Discovery, Plug-and-Play) 2. ช่วยลดภาระด้านความปลอดภัยของข้อมูล NETPIE ถูกออกแบบให้มีระดับและสิทธิ์ในการเข้าถึงในระดับ Fine Grain กล่าวคือ ผู้ใช้สามารถ ออกแบบได้เองทั้งหมดว่า สิ่งใดมีสิทธิ์คุยกับสิ่งใด สิ่งใดมีสิทธิ์หรือไม่ - เพียงใดในการอ่านหรือ เขียนข้อมูล และสิทธิ์เหล่านี้จะมีอายุการใช้งานนานเท่าใด หรือจะถูกเพิกถอนภายใต้เงื่อนไขใด เป็นต้น 3. ยืดหยุ่นต่อการขยายระบบ NETPIE มีสถาปัตยกรรมเป็นคลาวด์เซิร์ฟเวอร์อย่างแท้จริงในทุกองค์ประกอบ ของระบบ ทำให้เกิด ความยืดหยุ่นและคล่องตัวสูงในการขยายตัว นอกจากนี้ไมดูลต่างๆ ยังถูกออกแบบให้ ทำงานแยกจากกัน เพื่อให้เกิดสภาวะ Loose Coupling และสื่อสารกันด้วยวิธี Asynchronous Messaging ช่วยให้ แพลตฟอร์มมีความน่าเชื่อถือได้สูง นำไปใช้ซ้ำและพัฒนาต่อได้ง่าย ดังนั้นผู้พัฒนาไม่ จำเป็นต้องกังวลกับ การขยายตัวเพื่อรับโหลดที่เพิ่มขึ้นในระบบอีกต่อไปราบรื่น นอกจากนี้ Microgear ยัง ช่วยอำนวยความสะดวก ในการสร้างช่องทางการสื่อสารแบบ เข้ารหัสในกรณีที่ใช้ต้องการ ส่วนการ แลกเปลี่ยนข้อมูลระหว่าง Microgear และคลาวด์ของ NETPIE จะใช้โพรโทคอล MQTT ในการสื่อสาร
2. ด้านการยืนยันตัวตน (Authentication) ในขั้นตอนการสร้างการเชื่อมต่อ Microgear จะช่วย ยืนยันตัวตนของอุปกรณ์กับคลาวด์ของ NETPIE โดยการพิสูจน์ตัวตน (Identity) ของอุปกรณ์จะ ใช้ ข้อมูลประกอบกันสามส่วนคือ App ID, App Key และ Token
3. ด้านการขออนุญาตสิทธิ์ (Authorization) การขออนุญาตสิทธิ์ในการสื่อสารจะเกิดขึ้น ใน ขั้นตอนการสร้างการเชื่อมต่อ ควบคู่กับการยืนยันตัวตน คลาวด์ของ NETPIE จะเป็นผู้ออก ใบอนุญาต (Token) ที่ระบุว่าอุปกรณ์ตัวนี้ สามารถสื่อสารได้กับอุปกรณ์ตัวใดบ้าง ในกรณีปกติ อุปกรณ์ที่อยู่ภายใต้ กลุ่ม App ID เดียวกันเท่านั้น จึงจะมีสิทธิ์สื่อสารกันได้ (ยกเว้นในกรณีการใช้ Freeboard Microgear ที่ อนุญาตให้สื่อสารข้าม App ID ได้
4. ด้านการประสานงาน (Coordination) Microgear มีฟังก์ชันที่ช่วยให้อุปกรณ์ต่างๆ ภายในกลุ่ม App ID เดียวกันทราบสถานะของกันและกัน เช่น ทราบว่ามีอุปกรณ์ใดออนไลน์เข้ามาใหม่ใน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่ม หรือมีอุปกรณ์ได้ออกไปจากกลุ่ม รวมถึงทราบการเปลี่ยนแปลงสถานะของอุปกรณ์ที่สนใจ ติดตาม จากข้อมูลดังกล่าวผู้ใช้สามารถกำหนดบทบาทหน้าที่ให้อุปกรณ์ในกลุ่มตามสถานะของ อุปกรณ์อื่นๆ ใน กลุ่ม เช่น หากเป็นอุปกรณ์ตัวแรกในกลุ่มให้ทำหน้าที่เป็นหัวหน้ากลุ่ม เป็นต้น

## 2.4 .MICROGEAR

Microgear คือซอฟต์แวร์ไลบรารีของ NETPIE ที่ติดตั้งอยู่บนอุปกรณ์ที่ต้องการเชื่อมต่อสื่อสาร ผ่าน คลาวด์ของ NETPIE Microgear เปรียบเสมือนตัวกลางและผู้ช่วยในการสร้างและดูแลการเชื่อมต่อ ให้ มีความเสถียร ปลอดภัย ให้การสื่อสารแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์เป็นไปอย่างราบรื่น บทบาทหน้าที่ ของ Microgear สามารถแบ่งออกเป็น 4 ด้านคือ 1. ด้านการสื่อสาร (Communication) Microgear จะ เป็นผู้ช่วยในการสร้างการเชื่อมต่อ (Connection) ไปยังคลาวด์ของ NETPIE และคอยตรวจสอบสถานะ ของการเชื่อมต่อ หากการ เชื่อมต่อมีปัญหา Microgear สามารถช่วยเชื่อมต่อให้ใหม่เพื่อให้การสื่อสาร เป็นไปได้อย่างราบรื่น นอกจากนี้ Microgear ยังช่วยอำนวยความสะดวก ในการสร้างช่องทางการสื่อสาร แบบ เข้ารหัสในกรณีที่ใช้ต้องการ ส่วนการแลกเปลี่ยนข้อมูลระหว่าง Microgear และคลาวด์ของ NETPIE จะใช้โพรโทคอล MQTT ในการสื่อสาร 2. ด้านการยืนยันตัวตน (Authentication) ในขั้นตอนการ สร้างการเชื่อมต่อ Microgear จะช่วย ยืนยันตัวตนของอุปกรณ์กับคลาวด์ของ NETPIE โดยการพิสูจน์ ตัวตน (Identity) ของอุปกรณ์จะ ใช้ข้อมูลประกอบกันสามส่วนคือ App ID, App Key และ Token 3. ด้านการขออนุญาตสิทธิ์ (Authorization) การขออนุญาตสิทธิ์ในการสื่อสารจะเกิดขึ้นใน ขั้นตอนการสร้าง การเชื่อมต่อ ควบคู่กับการยืนยันตัวตน คลาวด์ของ NETPIE จะเป็นผู้ออก ใบอนุญาต (Token) ที่ระบุว่า อุปกรณ์ตัวนี้สามารถสื่อสารได้กับอุปกรณ์ตัวใดบ้าง ในกรณีปกติ อุปกรณ์ที่อยู่ภายใต้กลุ่ม App ID เดียวกัน เท่านั้น จึงจะมีสิทธิ์สื่อสารกันได้ (ยกเว้นในกรณีการใช้ Freeboard Microgear ที่อนุญาตให้สื่อสารข้าม App ID ได้ ซึ่งจะอธิบายในบทที่ 5) 4. ด้านการประสานงาน (Coordination) Microgear มีฟังก์ชันที่ช่วย ให้อุปกรณ์ต่างๆ ภายในกลุ่ม App ID เดียวกันทราบสถานะของกันและกัน เช่น ทราบว่ามีอุปกรณ์ใด ออนไลน์เข้ามาใหม่ใน กลุ่ม หรือมีอุปกรณ์ได้ออกไปจากกลุ่ม รวมถึงทราบการเปลี่ยนแปลงสถานะของ อุปกรณ์ที่สนใจ ติดตาม จากข้อมูลดังกล่าวผู้ใช้สามารถกำหนดบทบาทหน้าที่ให้อุปกรณ์ในกลุ่มตามสถานะ ของ อุปกรณ์อื่นๆ ในกลุ่ม เช่น หากเป็นอุปกรณ์ตัวแรกในกลุ่มให้ทำหน้าที่เป็นหัวหน้ากลุ่ม เป็นต้น Microgear ถูกพัฒนาขึ้นเพื่อให้ทำงานได้กับอุปกรณ์ที่หลากหลาย ในส่วนของซอฟต์แวร์มี Microgear ให้ เลือกใช้กับ Programming Language ได้แก่ Node.js Python HTML5 Java Android C# สำหรับ อุปกรณ์ฮาร์ดแวร์ประเภทไมโครคอนโทรลเลอร์ Microgear เปรียบเสมือน Firmware ซึ่งมี Microgear ที่ รองรับ Arduino with Ethernet Shield (ใช้ได้กับ Arduino Mega) และ Microgear สำหรับ Wi-Fi ไมโครคอนโทรลเลอร์ ESP8266 Microgear ที่พัฒนาขึ้นทั้งหมดถูกรวบรวมไว้ที่ <https://github.com/netpieio> โดยมีสัญญาอนุญาตให้ใช้สิทธิ์แบบเปิดประเภท ISC License ซึ่งอนุญาต

ให้ทำซ้ำ ดัดแปลง และ/หรือส่งต่อไลบรารีนี้ได้ ทั้งในการใช้งานเชิงสาธารณประโยชน์และเชิงพาณิชย์  
รายละเอียดของสัญญาอนุญาตให้ใช้สิทธิ์แบบ ISC License มีดังนี้

ฟังก์ชันหลักของ Microgear แต่ละชนิดอาจมีชื่อและชนิดของฟังก์ชันแตกต่างกันตามลักษณะของการเขียน  
โปรแกรม ในภาษานั้นๆ ในที่นี้ขอยกตัวอย่างฟังก์ชันที่มีเหมือนกันอยู่ในหลาย Microgear โดยขออ้างอิงชื่อ  
ฟังก์ชัน จาก HTML5 Microgear สำหรับรายละเอียดฟังก์ชันของแต่ละชนิด Microgear สามารถดูได้จาก  
ภาคผนวก หรือเอกสาร Readme ใน <https://github.com/netpie.io>

- create สร้าง Microgear เพื่อเริ่มต้นใช้งาน
- connect เชื่อมต่อ Microgear เข้ากับคลาวด์ของ NETPIE
- set Alias กำหนดชื่อเล่นของอุปกรณ์เพื่อใช้ระบุตัวตนของอุปกรณ์ภายใน NETPIE
- chat ส่งข้อความแบบเจาะจงผู้รับ
- publish ส่งข้อความแบบไม่เจาะจงผู้รับไปยังหัวข้อสนทนาที่กำหนด
- subscribe ระบุความสนใจในหัวข้อสนทนา บอกรับข้อความที่เกิดขึ้นบนหัวข้อนั้นๆ
- unsubscribe ยกเลิกการบอกรับข้อความในหัวข้อสนทนาที่เคย subscribe ไว้
- reset Token ยกเลิกใบอนุญาต (Token) และลบใบอนุญาตออกจาก cache บนอุปกรณ์
- useTLS ระบุว่าต้องการสร้างการเชื่อมต่อแบบเข้ารหัสระหว่าง Microgear กับคลาวด์ของ NETPIE
- on ตอบสนองต่อเหตุการณ์ที่สนใจผ่านการเรียก Callback Function



รูปที่ 2.3 ลำดับการเรียกฟังก์ชันพื้นฐานของ Microgear

Events ของ Microgear การทำงานของ Microgear เป็นแบบ Event-driven จึงต้องตอบสนองต่อ  
เหตุการณ์ต่างๆ ด้วยการ เขียน Callback Function ซึ่งชนิดของเหตุการณ์ที่สามารถเกิดขึ้น มีดังนี้

- connected เกิดขึ้นเมื่อ Microgear เชื่อมต่อกับ NETPIE สำเร็จ
- closed เกิดขึ้นเมื่อ Microgear ปิดการเชื่อมต่อกับ NETPIE
- error เกิดขึ้นเมื่อมีความผิดพลาดเกิดขึ้นกับ Microgear
- message เกิดขึ้นเมื่อมีข้อความเข้ามาที่อุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- present เกิดขึ้นเมื่อมีอุปกรณ์ใน App ID เดียวกันเชื่อมต่อเข้ามาบน NETPIE
- absent เกิดขึ้นเมื่อมีอุปกรณ์ใน App ID เดียวกันหายไปจากการเชื่อมต่อกับ NETPIE

## 2.5 NETPIE FEED

NETPIE FEED ข้อมูลที่เกิดจากการอ่านค่าของอุปกรณ์เซนเซอร์ในบอทที่แล้ว สามารถนำมาใช้ให้เกิดประโยชน์ได้ อย่างหลากหลาย เช่น ใช้สำหรับการตรวจสอบ (Monitoring) หรือการแสดงผลของข้อมูล (Data Visualization) โดยเฉพาะอย่างยิ่งข้อมูลแบบทันเวลา (Real-time Data) ซึ่งจะต้องเก็บรวบรวมด้วยอัตรา ความถี่ที่เหมาะสมเพื่อให้การตรวจสอบหรือการแสดงผลตรงตามความต้องการ ในปัจจุบันแพลตฟอร์ม NETPIE มีบริการที่สามารถเก็บข้อมูลและแสดงผลที่เรียกว่า Feed ซึ่งทำหน้าที่เสมือนถังเก็บข้อมูล ประเภท Time-Series กล่าวคือ เป็นชุดข้อมูลหรือค่าตัวแปร ณ เวลาต่างๆ เช่นอุณหภูมิอากาศในช่วงเวลา ต่างๆ ของวัน เป็นต้น ข้อมูลเหล่านี้จะถูกเก็บแบบต่อเนื่องสะสมกันไปตลอด และสามารถเรียกออกมาดูใน ช่วงเวลาใดก็ได้ในบอทนี้จะอธิบายรายละเอียดการใช้งาน NETPIE Feed ตั้งแต่การตั้งค่าการเก็บข้อมูลจาก Datasource

### การสร้าง FEED

1. เลือก FEEDS จากเมนู RESOURCES
2. สร้าง Feed ใหม่ หรือเพิ่ม Feed ด้วยการคลิกที่เครื่องหมาย + ที่มุมบนของ Feed
3. ตั้งชื่อ Feed โดยที่ชื่อจะต้องไม่ซ้ำกับที่เคยมีมาก่อน และห้ามซ้ำกับของผู้ใช้อื่น เมื่อตั้งชื่อแล้วให้ คลิก CREATE เพื่อสร้าง Feed
4. เมื่อสร้าง Feed สำเร็จ จะเข้าสู่หน้าตั้งค่า Feed ตามภาพข้างล่าง ในตัวอย่างนี้ได้สร้างสร้าง Feed ที่มีชื่อว่า “my sensor” ซึ่งจะปรากฏในหน้ารวม Feed เมื่อเราเข้ามาในครั้งต่อไปเราสามารถกลับมาที่หน้าตั้งค่า นี้ได้อีกจากหน้ารวมของ Feed โดยคลิกที่เครื่องหมายรูปประแจด้าน ขวามือหลังชื่อ Feed ที่ต้องการ
5. เนื่องจาก Feed ที่สร้างขึ้นใหม่นี้ยังไม่สามารถรับค่าใดๆ เข้ามาได้ จึงต้องสร้าง Field ขึ้นมารับ ข้อมูล ให้กดปุ่ม + ADD จะปรากฏหน้าต่างให้ตั้งค่าต่างๆ ในตัวอย่างนี้ เป็นการสร้าง Field จำนวน 2 Field ที่มีชื่อว่า temp และ humid (หรือจะตั้งเป็นชื่ออะไรก็ได้) หากมีหน่วยก็ระบุไว้เพื่อ การแสดงผลต่อไป โดย ณ ขณะนี้ยังรองรับเฉพาะข้อมูลชนิดตัวเลข (Number) เท่านั้น โดยในอนาคตมีอาจมีการพัฒนาเพิ่มเติมให้รองรับข้อมูลชนิดอื่นๆ

6. คลิก SAVE จะได้ Feed ที่พร้อมรับค่ารายละเอียดการปรับแต่งอยู่ในลำดับถัดไป

### การกำหนดสิทธิ์ในการเข้าถึง FEED

เนื่องจาก Feed แต่ละอันที่สร้างขึ้นมานั้น เป็น Resource อิสระที่ไม่ขึ้นกับ App ID ดังนั้นในการ นำไปใช้งานจำเป็นต้องมีเรื่องของข้อกำหนดสิทธิ์การให้สิทธิ์กับอุปกรณ์ที่จะเข้ามาอ่านหรือเขียน Feed มีด้วยกัน 2 วิธี ได้แก่

1. การใช้ API Key ที่ปรากฏอยู่ในแท็บ Permission วิธีนี้จะอนุญาตให้ Client ที่มี API key สามารถอ่านเขียน Feed นี้ได้ วิธีการ 1 API key ไปใช้จะกล่าวถึงในต่อไป

2. การให้สิทธิ์กับ App ID ใช้ได้กับเฉพาะ App ID ของเจ้าของ Feed เท่านั้นวิธีนี้มีไว้เพื่ออำนวยความสะดวกในขั้นตอนการพัฒนาเนื่องจากการเป็นการให้สิทธิ์กันผ่านเว็บ จึงไม่ต้องเข้าไปแก้โค้ดโปรแกรมการให้สิทธิ์กับ App ID จะทำให้ทุกอุปกรณ์ใน App ID นั้นมีสิทธิ์อ่านหรือเขียน Feed นี้เหมือนกันหมด การตั้งค่า ทำได้โดยคลิกที่แท็บ Permission และคลิกที่ EDIT ตามภาพ

จากนั้นพิมพ์ App ID ที่ต้องการให้สิทธิ์หรือเลือกจากเมนู Drop-Down และคลิก SAVE ในตัวอย่าง ข้างล่าง เป็นการให้สิทธิ์กับ App ID ที่ชื่อ “IoT demo” ดังนั้นทุกอุปกรณ์ภายใต้ App ID นี้ จะสามารถอ่าน เขียน Feed ได้โดยอัตโนมัติ การอนุญาตแบบนี้จะเป็นการแจกสิทธิ์ในรูปแบบเดียวกับการ 1 Default API key ของ Feed ไปใช้

### การเขียนข้อมูลลงใน FEED

การเขียนข้อมูลเพื่อเก็บใน Feed มี 2 วิธีดังนี้วิธีที่ 1 ใช้ REST API (ข้อมูลและการใช้งานโดยละเอียดของ REST API อยู่ในบทที่ 7) REST API สามารถเรียกผ่าน Command Line ได้โดยใช้โปรแกรม curl ซึ่งเป็น HTTP Client แบบ Command Line ที่นักพัฒนามานิยมใช้กันมาก ในระบบปฏิบัติการ Mac OSX และ Linux โปรแกรมนี้จะถูก ติดตั้งมาแบบพร้อมใช้งานอยู่แล้ว เพียงแค่เปิด Terminal ก็สามารถพิมพ์คำสั่งได้เลย แต่สำหรับ Windows อาจต้องติดตั้งโปรแกรมเพิ่ม โดยสามารถดาวน์โหลดได้จาก <https://curl.haxx.se/download.html> การเขียน Feed ผ่าน REST API จะต้องกระทำโดยใช้ API Key ควบคู่กันไปด้วยเสมอ และมีรูปแบบ คำสั่งที่เรียกผ่าน curl บน Terminal ดังนี้ (เครื่องหมาย\$ ไม่ต้องพิมพ์ลงไป ใส่ไว้เพื่อสื่อว่าเป็น Command Line)

เมื่อกลับไปดูที่หน้า Feed เราจะเห็นจุดข้อมูลปรากฏขึ้นมาแบบนี้ ข้อที่ควรระวังคือ ชื่อ Field ข้อมูล ในคำสั่ง (temp และ humid)จะต้องตรงกับชื่อField ที่ได้สร้างรอเอาไว้บน Feed เพื่อให้ข้อมูลถูกจัดเก็บ สำเร็จ

วิธีที่ 2 ใช้ Microgear Function สำหรับอุปกรณ์ที่ต่อ NETPIE ผ่าน Microgear Library อยู่แล้ว สามารถส่งค่าเข้าไปใน Feed ผ่าน Real-time Message ได้เลยโดยใช้ฟังก์ชัน `microgear.writeFeed()` ซึ่งจะมีใน Library ปัจจุบันทุกภาษา ของ NETPIE อยู่แล้ว (หากไม่มีโปรดอัปเดตให้เป็นเวอร์ชันล่าสุด) การเขียน Feed ด้วย Microgear นี้ ใน กรณีที่เราได้ให้สิทธิ์กับ App ID ไว้แล้ว ก็ไม่จำเป็นต้องใส่ API Key ลงไปรูปแบบการใช้ฟังก์ชันมีดังนี้ `microgear.writeFeed("", "")` โดยที่ DATA เป็น String ที่มีลักษณะเดียวกับ REST API ดังตัวอย่างนี้ `microgear.writeFeed("mysensor","temp:25.2,humid:62.5")` ส่วนของ DATA นอกจากกรอกรับ String รูปแบบดังกล่าวแล้ว เรายังสามารถส่ง String ที่แปลงมา จาก json ได้ด้วย เช่น `"{\`temp\`:25.2,\`humid\`:62.5}"` หรือหากเป็น Node.js หรือ Java script เรายัง สามารถส่งตัวแปรที่มี Type เป็น Object ได้เลยโดยไม่ต้องแปลงเป็น String ก่อน

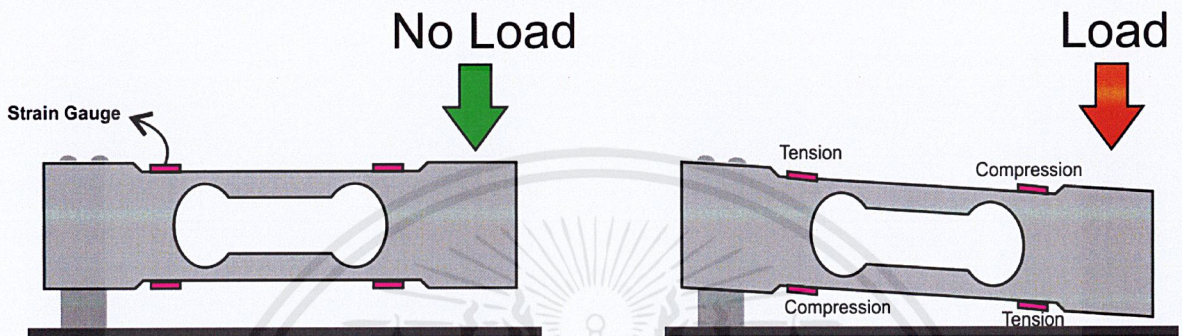
ในกรณีที่เรายังไม่ได้ให้สิทธิ์กับ App ID ฟังก์ชันข้างต้นจะได้รับข้อความ Error ว่าไม่มีสิทธิ์เขียน Feed จึงต้องใส่ API Key ลงไปในฟังก์ชันด้วยซึ่งจะเหมาะกับกรณีที่ต้องการให้มีเฉพาะอุปกรณ์เพียงบางตัวเท่านั้นที่สามารถเขียน Feed ได้รูปแบบการเรียกจะเป็นตามนี้ `microgear.writeFeed("", "", "")` ซึ่งตัวอย่างการใช้จริงจะอยู่ในรูปแบบดังนี้ `microgear.writeFeed("mysensor","temp:25.2,humid:62.5","5DunZ38nKP5dGC0h4Bj7mXyeMURdoXOo")` และทุกครั้งที่เราส่งค่าเข้าไปที่ Feed ไม่ว่าจะสำเร็จหรือไม่ จะมี Message ตอบกลับมาทาง Event Info หรือ Error เสมอ เราสามารถเช็คสถานะการเขียน Feed ได้จากการรับ Event info และ Error ซึ่งลักษณะ การเรียกใช้จะแตกต่างกันเล็กน้อยตามรูปแบบภาษาของ Library ที่ใช้งาน รายละเอียดเพิ่มเติมสามารถ ศึกษาได้ในส่วนอธิบาย Events ของเอกสาร readme ใน Microgear Library แต่ละภาษา 6.4 การดึงข้อมูลจาก FEED มาใช้งาน NETPIE มีวิธีใช้งานข้อมูลจาก Feed อยู่ 3 แบบดังนี้ วิธีที่ 1 การแสดงผลผ่านหน้าการจัดการของ Feed เราสามารถให้หน้า Feed ของ NETPIE.io ในการดูค่าย้อนหลังได้ในลักษณะเส้นกราฟและ สามารถตั้งค่าการแสดงผลได้จากแท็บ Data Display ดังนี้

- GRANULARITY คือความละเอียดของจุดข้อมูล ตัวอย่างนี้ตั้งไว้ที่ 12 นาที หมายความว่าจุดข้อมูลต่างๆ ที่เก็บมาในช่วงเวลาหน้าต่างละ 12 นาทีจะถูกนำมาเฉลี่ยเป็นจุดเดียว
- SINCE เป็นเวลาตั้งต้นย้อนหลังที่จะเรียกดูข้อมูล
- BEGIN AT 0 ตั้งเป็น TRUE จะเป็นการตั้งค่าแกน Y เริ่มต้นที่ 0
- MARKER แสดงวงกลมที่จุด
- AUTO GAP แทรกช่องว่างอัตโนมัติ (ในกรณีไม่มีข้อมูลในช่วงเวลาที่ตั้งไว้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

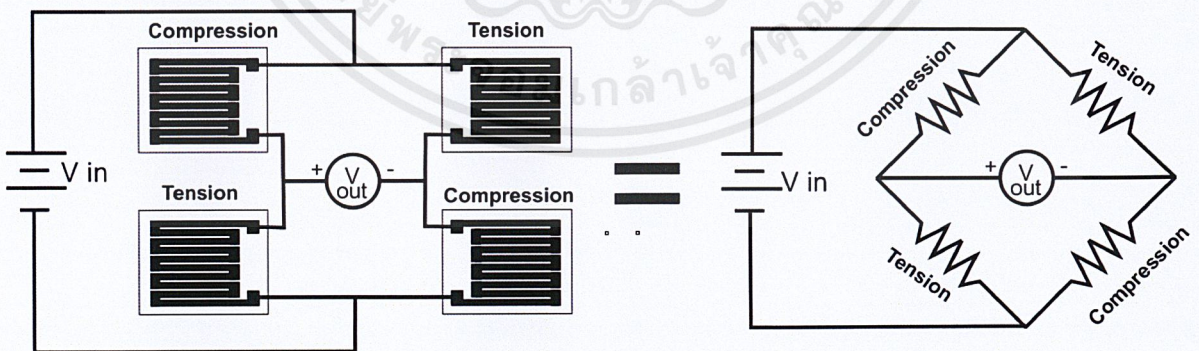
## 2.6 Load Cell

Load Cell คือ Sensor สำหรับตรวจวัดน้ำหนัก แรงกระทำทางกล หรือปริมาณของ Load ที่ต้องการทราบค่า โดยใช้ Strain Gauge มาติดตั้งในบริเวณที่มีการเปลี่ยนแปลงรูปร่างของ Load Cell เมื่อมีแรงมากระทำกับตัว Load Cell จะทำให้ Strain Gauge ที่ติดอยู่ในบริเวณที่มีการเปลี่ยนแปลงรูปร่าง ยืด หรือ หด ตัว ทำให้ค่าความต้านทานที่ตัว Strain Gauge เปลี่ยนไป



รูปที่ 2.4 การทำงาน Load cell

ตามรูปภาพ ในจุดที่ Strain Gauge ได้รับแรงกด (Compression) จะทำให้ Strain Gauge หดตัวเข้าหากัน และในจุดที่ได้รับแรงดึง (tension) จะทำให้ strain gauge ถูกยืดออก จึงทำให้ค่าความต้านทานของ Strain Gauge เปลี่ยนแปลงไป Strain Gauge ทั้ง 4 ตัวที่อยู่บน Load Cell แบบ Straight Bar จะถูกต่ออยู่ด้วยกันในลักษณะของวงจร Wheatstone Bridge

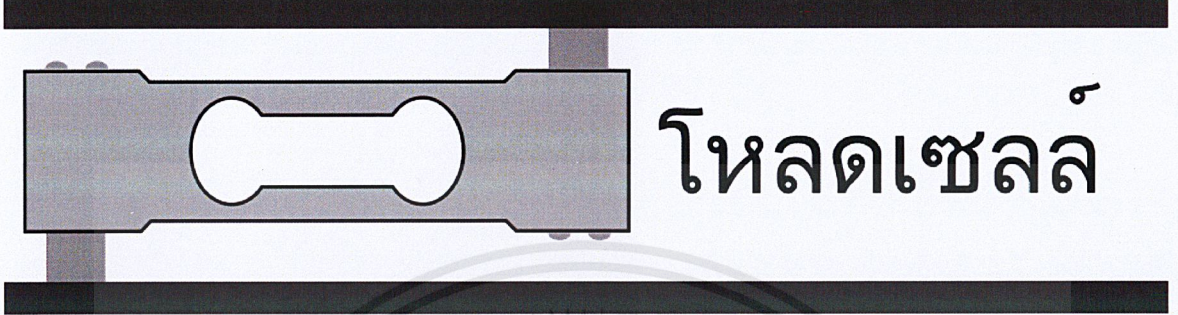


รูปที่ 2.5 วิธีติดตั้ง Load Cell แบบ Straight Bar เข้ากับแผ่นชั่งน้ำหนัก

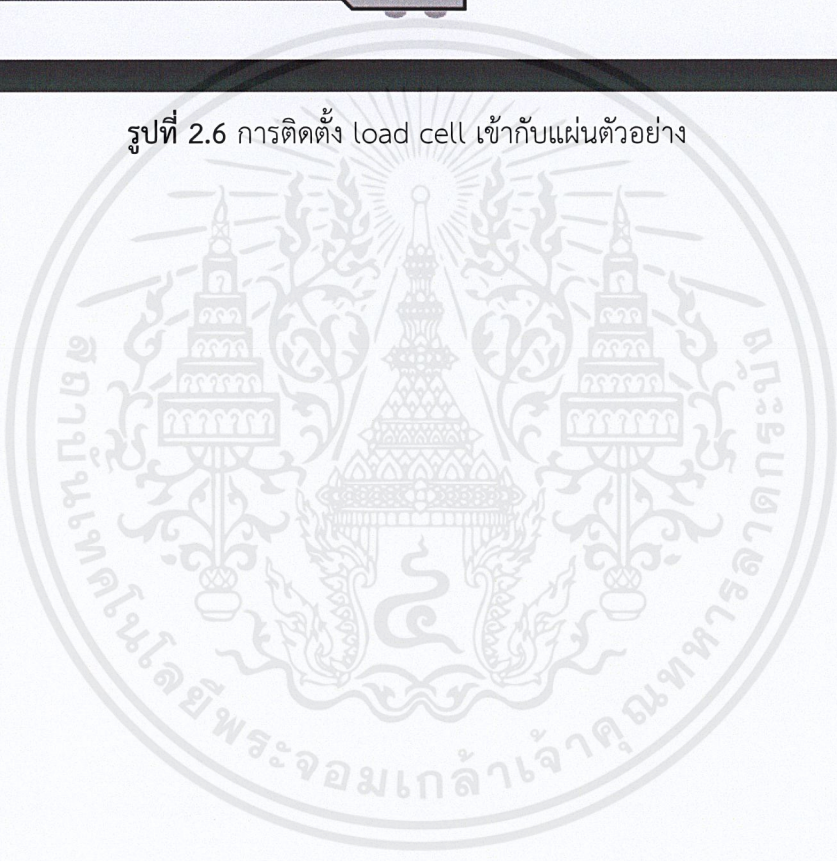
Load



แผ่นไม้ , แผ่นอะคริลิกหนา , แผ่นโลหะ



รูปที่ 2.6 การติดตั้ง load cell เข้ากับแผ่นตัวอย่าง



## บทที่ 3

### วิธีการดำเนินงาน

#### 3.1 ศึกษาปัญหาเพื่อทำโครงการ

ศึกษาข้อมูลและเอกสารอ้างอิงที่เกี่ยวข้องกับการพัฒนาเครื่องชั่งน้ำหนักร่วมกับเทคโนโลยี IoT เพื่อส่งข้อมูลที่ได้จากเซ็นเซอร์ Load cell เก็บไว้บน Cloud และแสดงผลผ่านทาง NETPIE FEEDs

#### 3.2 กำหนดที่มาและความสำคัญ, วัตถุประสงค์, ขอบเขต, แผนการดำเนินงาน

ตาราง 3.1 แผนการดำเนินงาน

รายละเอียด	ระยะเวลาดำเนินงาน			
	เดือนที่ 1	เดือนที่ 2	เดือนที่ 3	เดือนที่ 4
ศึกษาข้อมูลและสั่งซื้อเซ็นเซอร์	←→			
เขียนโปรแกรมและออกแบบ เครื่องมือวัด	←→	←→		
ติดตั้งและทดสอบการทำงาน		←→	←→	
บันทึกผล			←→	
ทำเล่มโครงการ				←→

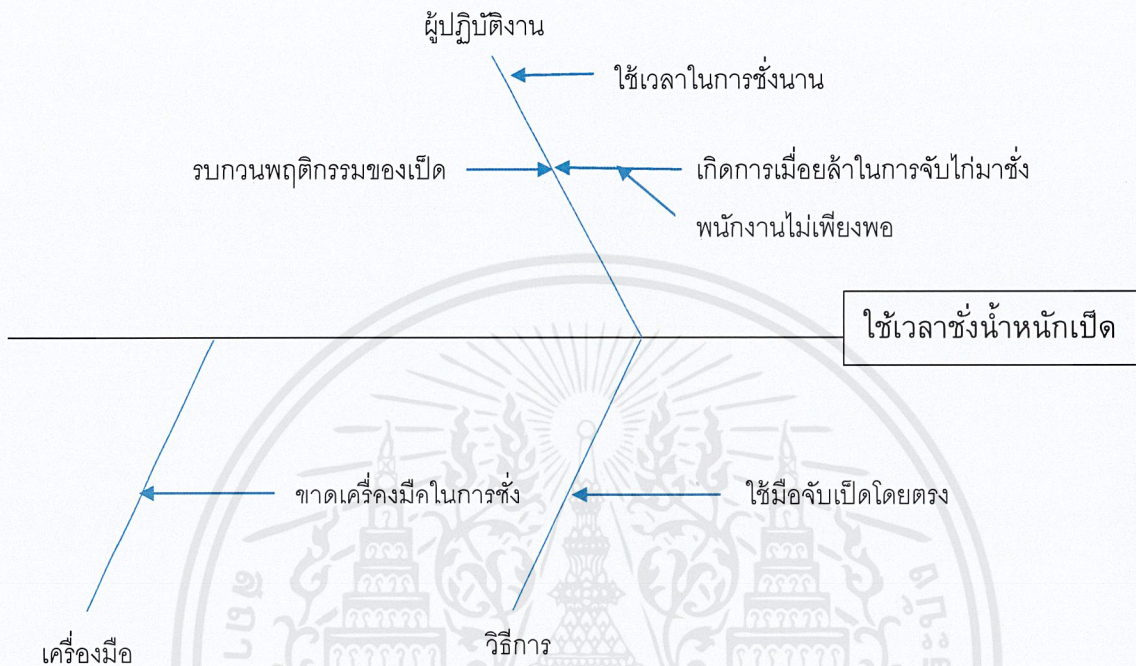
#### 3.3 ศึกษาและวิเคราะห์ปัญหา

##### 3.3.1 ศึกษาปัญหา ที่มาของโครงการ

จากการศึกษา พบว่าในเวลาชั่งน้ำหนักเปิดจำเป็นต้องใช้คนงานอุ้มเบ็ดทีละตัวมาชั่งน้ำหนัก ทำให้เกิดการล่าช้าในกระบวนการ และยังเป็นกระบวนการบวกรวมเบ็ดทำให้ได้ผลผลิตชิ้นได้น้อยลงจำเป็นต้องมีการวัดและติดตามค่าที่เกิดขึ้นอย่างต่อเนื่อง เพื่อให้เกษตรกรสามารถรับทราบความผิดปกติและแก้ไขปัญหาที่เกิดขึ้นได้ทันเวลา การพัฒนาระบบเซนเซอร์ร่วมกับเทคโนโลยี IoT จะช่วยให้สามารถเก็บข้อมูลน้ำหนักเบ็ดก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 วิเคราะห์ปัญหา



รูปที่ 3.1 แผนภูมิการวิเคราะห์ปัญหา

### 3.4. ศึกษาข้อมูลเพื่อพัฒนาเครื่องชั่งน้ำหนักเปิด

#### 3.4.1 การพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์

โครงการนี้เลือกใช้ Arduino esp8266 เป็นไมโครคอนโทรลเลอร์หลักในการพัฒนาระบบ และใช้ภาษา C บน Arduino IDE ในการพัฒนาโปรแกรม Arduino เป็นที่นิยมเนื่องจากมีไลบรารีที่หลากหลาย สามารถรองรับการใช้งานเซ็นเซอร์หลากหลายชนิด และการเชื่อมโยงกับระบบ Internet of Things เช่น การสื่อสารผ่านระบบ Wifi เพื่อส่งข้อมูลไปยัง Cloud server

#### 3.4.2 ระบบ Cloud ที่เลือกใช้

ระบบ Cloud ที่เลือกใช้คือ NETPIE สามารถแสดงผลของค่าคุณสมบัติต่างๆผ่าน Web server และ สามารถแสดงผลได้ตลอดเวลา โดยจะส่งข้อมูลทุกๆ 1 นาที ไปยัง FEED ของ NETPIE เพื่อแสดงค่าในรูปแบบของกราฟเส้น เพื่อที่ผู้พัฒนาสามารถนำข้อมูลจาก FEED มาวิเคราะห์ข้อมูล เพื่อพัฒนาคุณภาพของเครื่องมือตรวจวัด

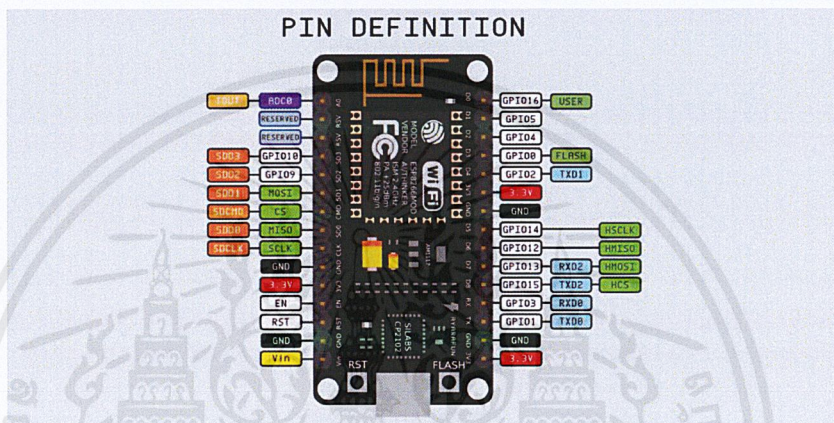
### 3.5 อุปกรณ์

#### 3.5.1 เซ็นเซอร์ที่เลือกใช้

อุปกรณ์ที่สำคัญเพื่อพัฒนาเครื่องชั่งน้ำหนักเบ็ดมีทั้งหมด 3 อย่าง ได้แก่

#### ESP8266/NODEMCU

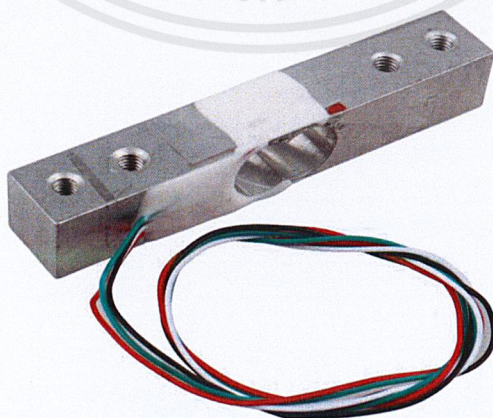
NodeMCU รุ่น 12E หรือ NodeMCU V2 หรือ NodeMCU Dev kit v1.0 ซึ่งใช้โมดูล ESP8266-SoC (ESP-12E) ซึ่งเป็น 32-bit Microcontroller และมี Built-in WiFi Node MCU-12E รวมทั้งผังคำอธิบายขาต่อแสดงไว้ดังรูปที่



รูปที่ 3.2 Node MCU-12E และผังคำอธิบายขาต่อ

#### Load Cell

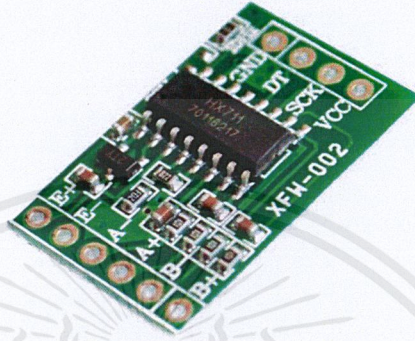
Load Cell คือ Sensor สำหรับตรวจวัดน้ำหนัก แรงกระทำทางกล หรือปริมาณของ Load ที่ต้องการทราบค่า โดยใช้ Strain Gauge มาติดตั้งในบริเวณที่มีการเปลี่ยนแปลงรูปทรงของ Load Cell เมื่อมีแรงมากระทำกับตัว Load Cell จะทำให้ Strain Gauge ที่ติดอยู่ในบริเวณที่มีการเปลี่ยนแปลงรูปทรง ยืด หรือ หด ตัว ทำให้ค่าความต้านทานที่ตัว Strain Gauge เปลี่ยนไป



รูปที่ 3.3 Load cell

## โมดูล HX711

การเปรียบเทียบความต่างศักย์ไฟฟ้าที่ขาอินพุตที่ขา A และ B ของโมดูลสัญญาณที่ได้จะอยู่ในรูปแบบมิลลิโวลต์ และทำการขยายสัญญาณเพื่อเพิ่มความแม่นยำในการวัด อีกทั้งยังใช้สำหรับการแปลงสัญญาณ Analog เป็นสัญญาณ Digital เพื่อส่งข้อมูลไปยังบอร์ดไมโครคอนโทรลเลอร์



รูปที่ 3.4 HX711

### 3.6 ขั้นตอนการออกแบบ

โครงสร้างถูกออกแบบจากโปรแกรม Solid works ใช้หลักการคือการนำเอาขนาดของเบ็ด เพื่อให้เป็นชิ้นเครื่องชั่งทีละตัว โดยเครื่องชั่งน้ำหนักมีความกว้าง 30 เซนติเมตร มีความยาว 30 เซนติเมตร และมีความสูง 7 เซนติเมตร เพื่อให้เปิดชิ้นชั่งน้ำหนักได้ทีละตัว



รูปที่ 3.5 แบบเครื่องชั่งน้ำหนักเบ็ด สามมิติ

### 3.7 การเขียนโปรแกรม

#### 3.7.1 Calibration Load Cell

การ Calibration นั้นเราจำเป็นจะต้องมี Load หรือ น้ำหนักที่เราทราบค่าอยู่แล้ว เพื่อทำการ Calibration โดยทั่วไปมักจะใช้ ต้มน้ำหนักมาตรฐานเพื่อทำการ Calibration โดยมีวิธีการดังนี้



รูปที่ 3.6 ต้มน้ำหนักมาตรฐาน

#### Calibration

- >> ติดตั้ง Load Cell เข้ากับแผ่นชั่งน้ำหนัก
- >> เชื่อมต่อ Load Cell และ HX711 เข้ากับบอร์ด Arduino
- >> Download Library สำหรับใช้กับ HX711
- >> <https://github.com/bogde/HX711>
- >> ติดตั้ง Library HX711 ให้กับ Arduino IDE (Copy Folder Library HX711 ลงใน Folder libraries ที่อยู่ใน Folder ของ Arduino IDE)
- >> Download Code Auto Calibrate >> File แนบ (Arduino\_Auto\_Cal)
- >> เปิด Arduino IDE และ เปิด File Arduino\_Auto\_Cal.ino ขึ้นมา
- >> แก้ค่าในตัวแปร real\_weight ให้เป็นค่าน้ำหนักจริงของ load ที่จะใช้นำมา Calibrate (หน่วยเป็น กิโลกรัม) ในที่นี้คือค่าน้ำหนักของขวดน้ำ 1.5 ลิตร น้ำหนักประมาณ 1.56 กิโลกรัม

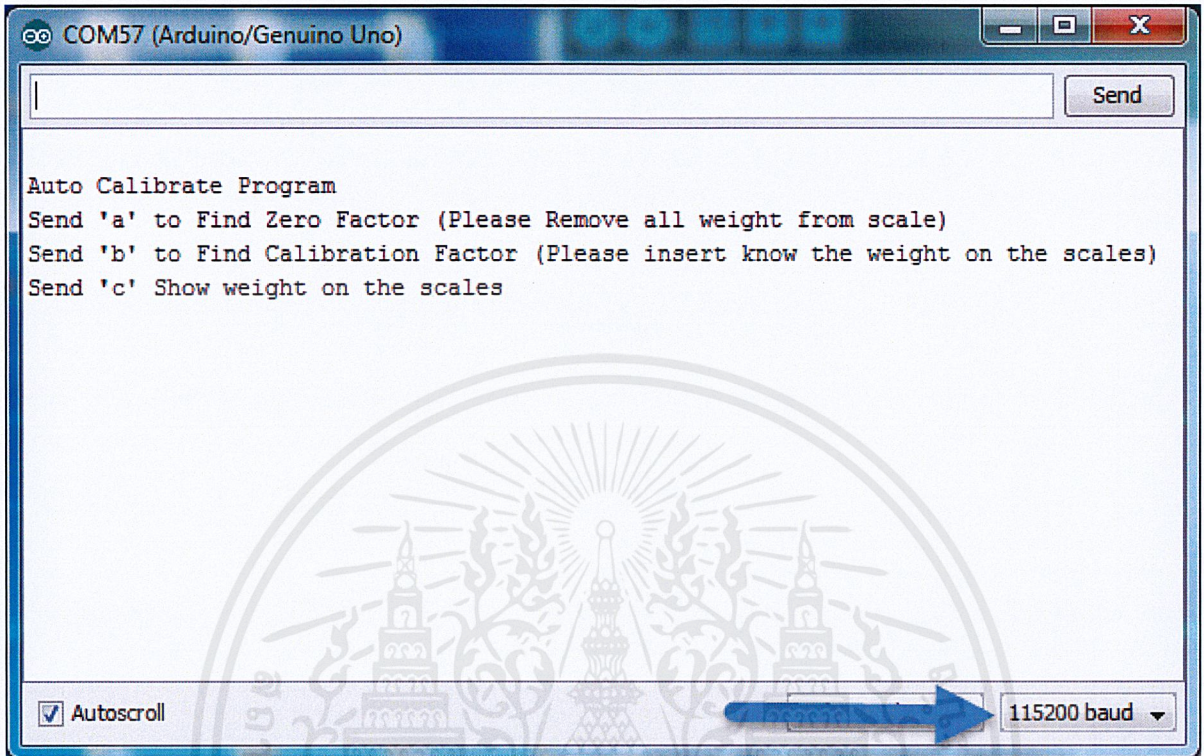
```
10 float offset=0;  
11 float calibration factor = 1;  
12 float real_weight = 1.560;//kg
```

รูปที่ 3.7 คำสั่งกำหนดค่าน้ำหนักอ้างอิง

- >> Upload โปรแกรม Arduino\_Auto\_Cal ลงไปยัง board Arduino

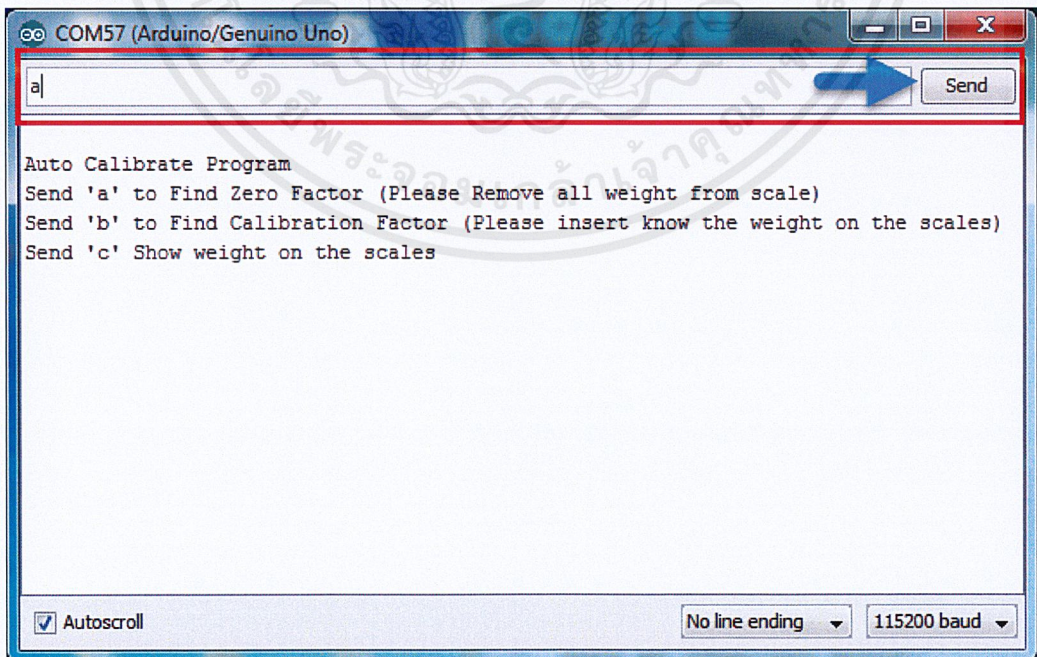
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- >> เปิด หน้าต่าง Serial Monitor ขึ้นมา
- >> ตั้งค่า Baud rate เป็น 115200



รูปที่ 3.8 การตั้งค่า Baud rate

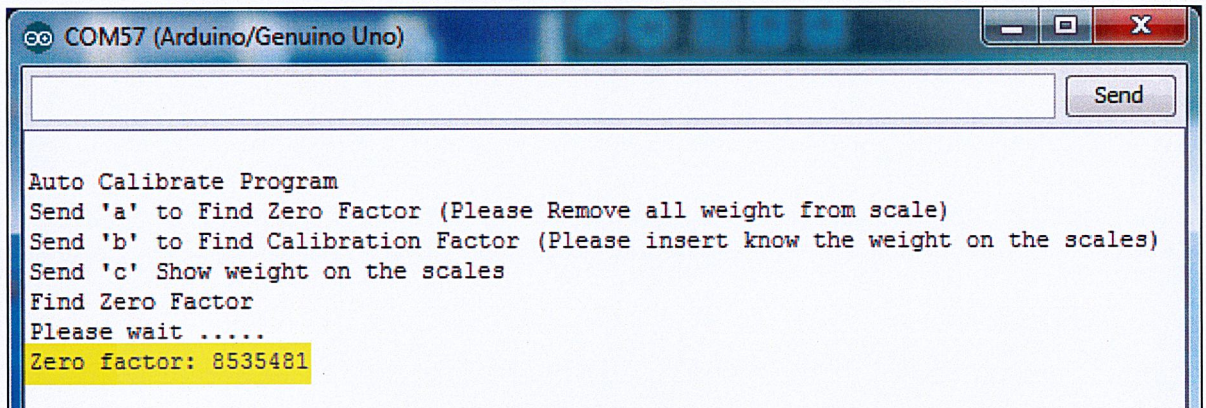
- >> นำสิ่งของทุกอย่างออกจาก แผ่นชั่งน้ำหนัก
- >> ส่ง ตัว 'a' ไป เพื่อหาค่า zero



รูปที่ 3.9 การส่งค่า a รูปแบบ command

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

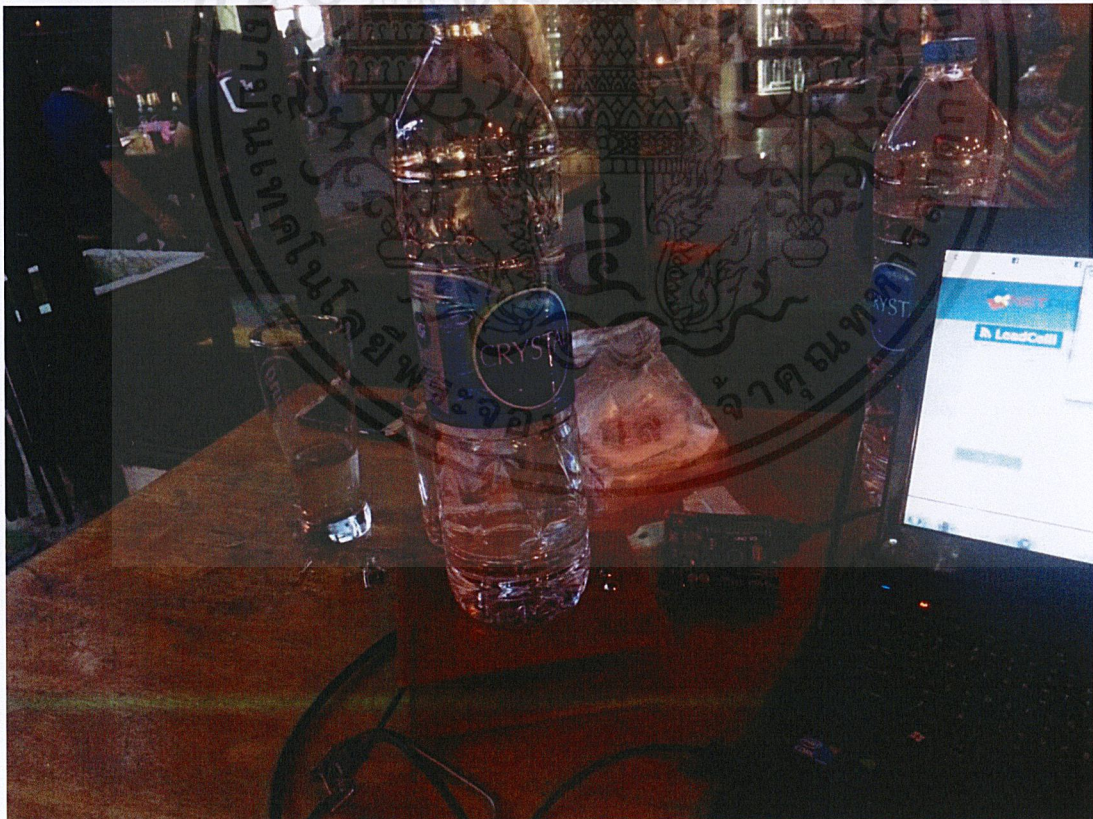
>> รอสักครู่โปรแกรม จะแสดงค่า Zero Factor ออกมาให้ ให้เราจดบันทึกค่านี้เอาไว้ใช้งาน (ในการทดลองนี้คือ 8535481)



```
COM57 (Arduino/Genuino Uno)
Auto Calibrate Program
Send 'a' to Find Zero Factor (Please Remove all weight from scale)
Send 'b' to Find Calibration Factor (Please insert know the weight on the scales)
Send 'c' Show weight on the scales
Find Zero Factor
Please wait .....
Zero factor: 8535481
```

รูปที่ 3.10 แสดงค่า Zero Factor

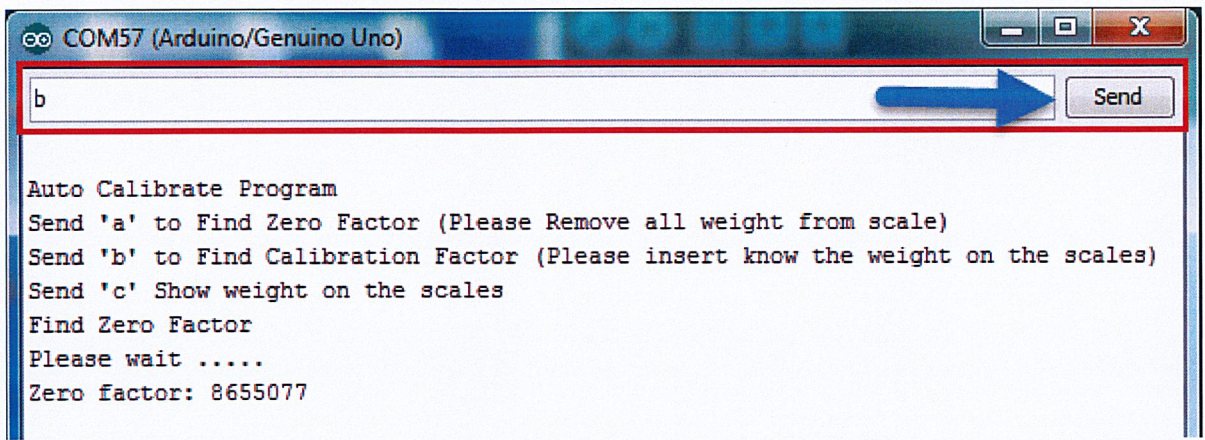
>> นำวัตถุที่ทราบค่าน้ำหนักที่แน่นอนมาวางลงบนแผ่นชั่งน้ำหนัก (น้ำหนักของวัตถุที่ใส่ลงในตัวแปร `real_weight` ใน `Arduino_Auto_Calino`)



รูปที่ 3.11 ตัวอย่างวัตถุในการชั่งน้ำหนัก

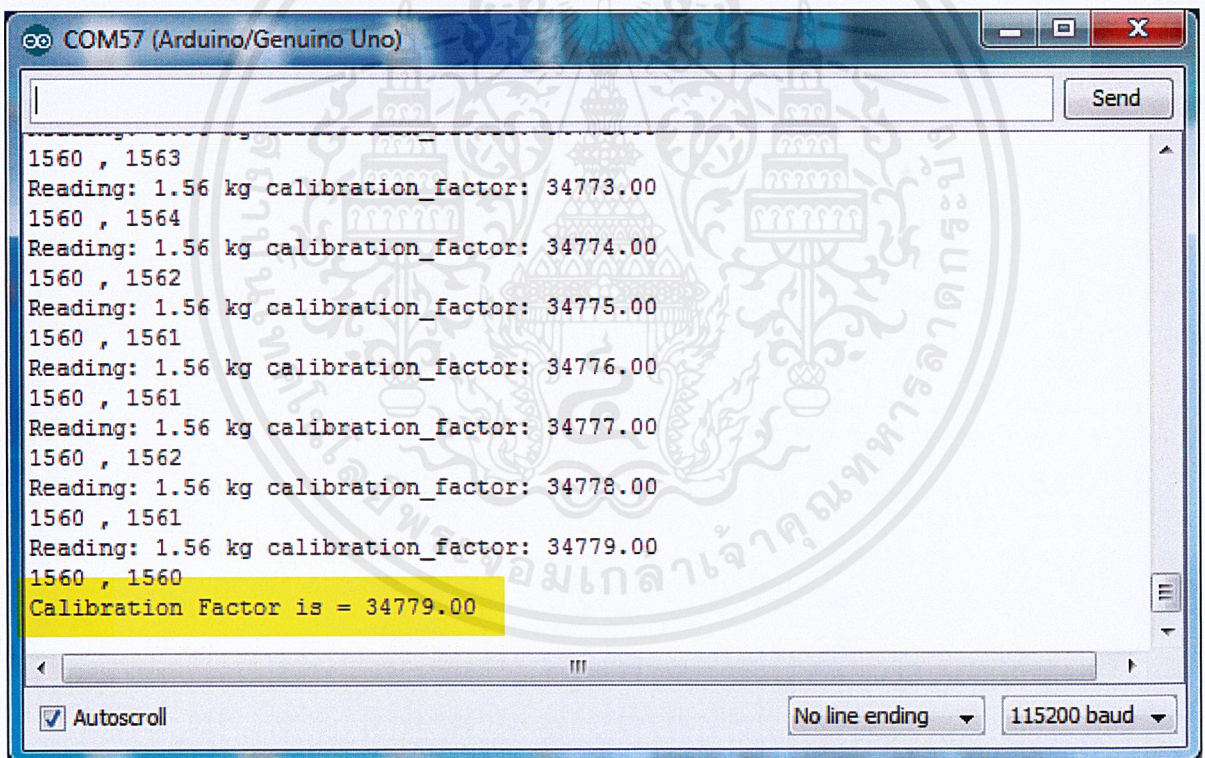
>> ส่ง ตัว 'b' ไป เพื่อหาค่า calibration factor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 การส่งค่า b รูปแบบ command

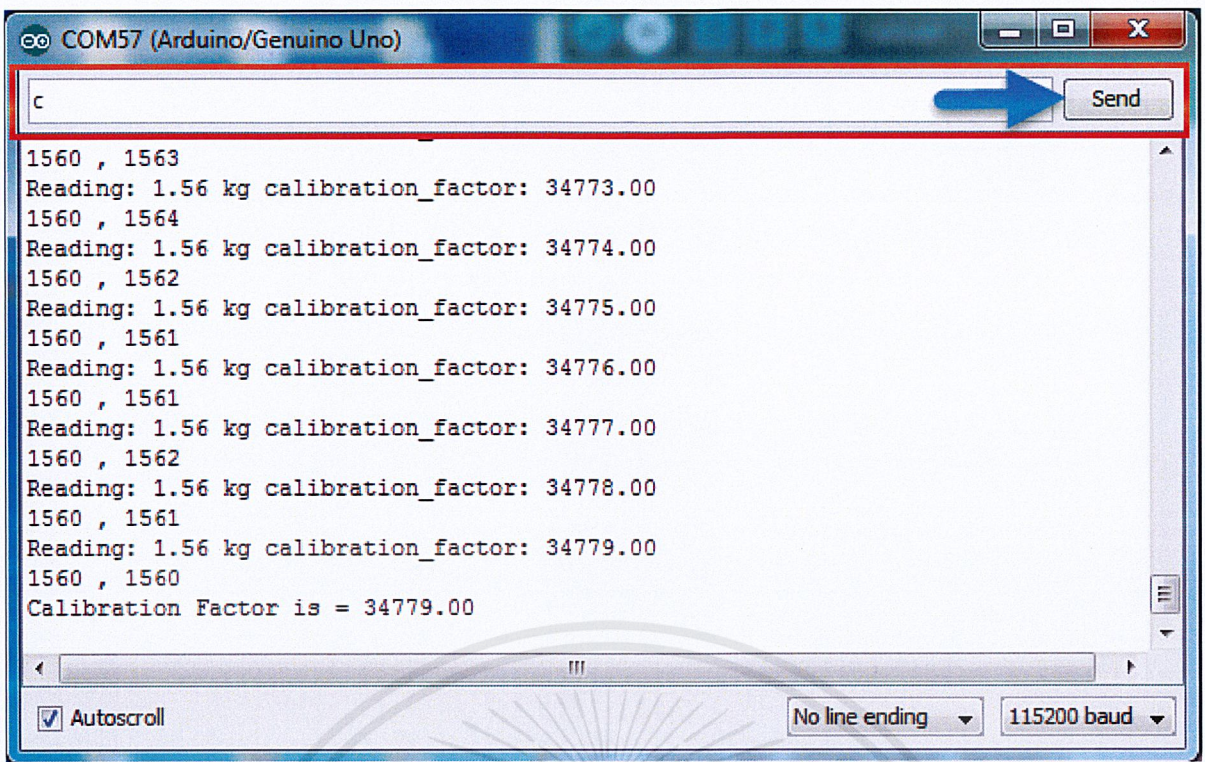
>> รอนโปรแกรมสามารถหาค่า calibration factor ได้สำเร็จ ให้เราดบันทึกค่านี้เอาไว้ใช้งาน (ในการทดลองนี้คือ 34779.00)



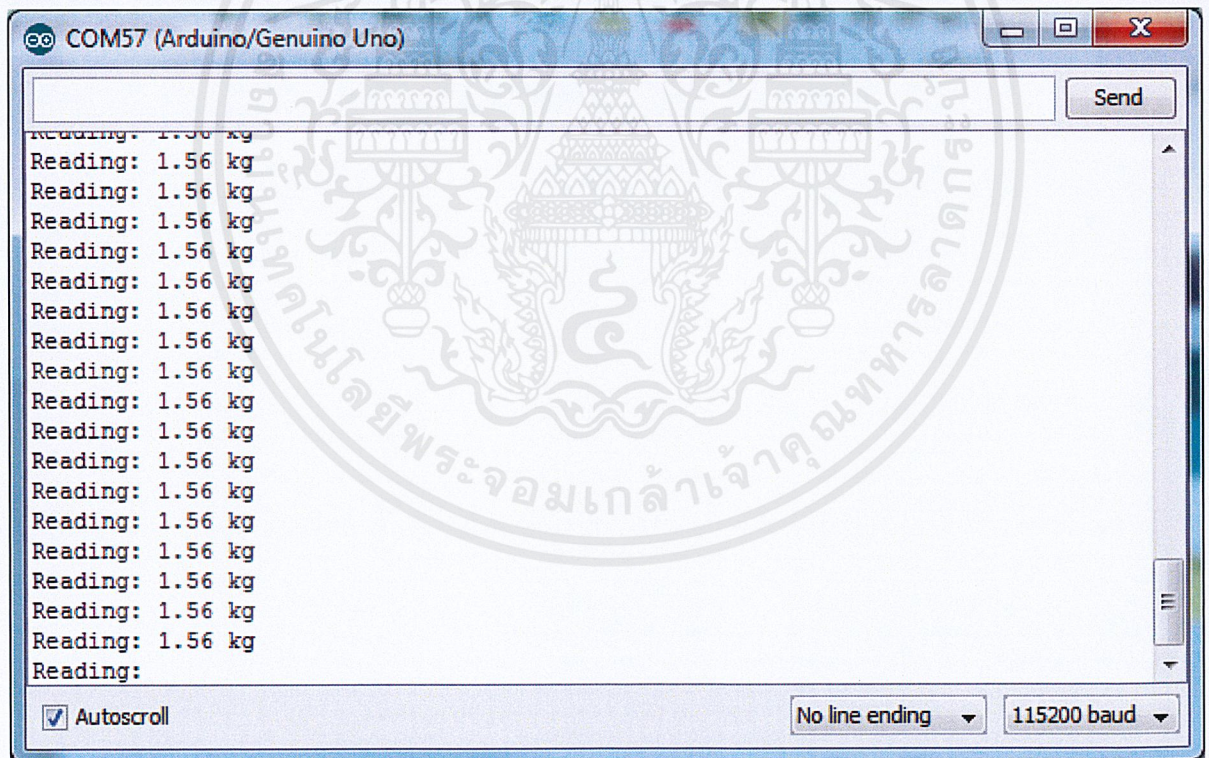
รูปที่ 3.13 ตัวอย่างการหาค่า calibration factor

>> ส่ง ตัว 'c' ไป เพื่อทดลองอ่านค่าน้ำหนักโดยใช้ Zero Factor และ calibration factor ที่ใช้โปรแกรมหาค่ามา

>> ผลการทดลองอ่านค่า



รูปที่ 3.12 การส่งค่า c รูปแบบ command



รูปที่ 3.13 ตัวอย่าง การนำค่า Zero Factor และ calibration factor ไปใช้งาน

>> โหลดตัวอย่าง Code >> File แนบ (ex\_loadcell)

>> จากตัวอย่างเรื่อง วิธี Calibrate Load Cell ทำให้เราทราบค่า Zero Factor และ calibration factor

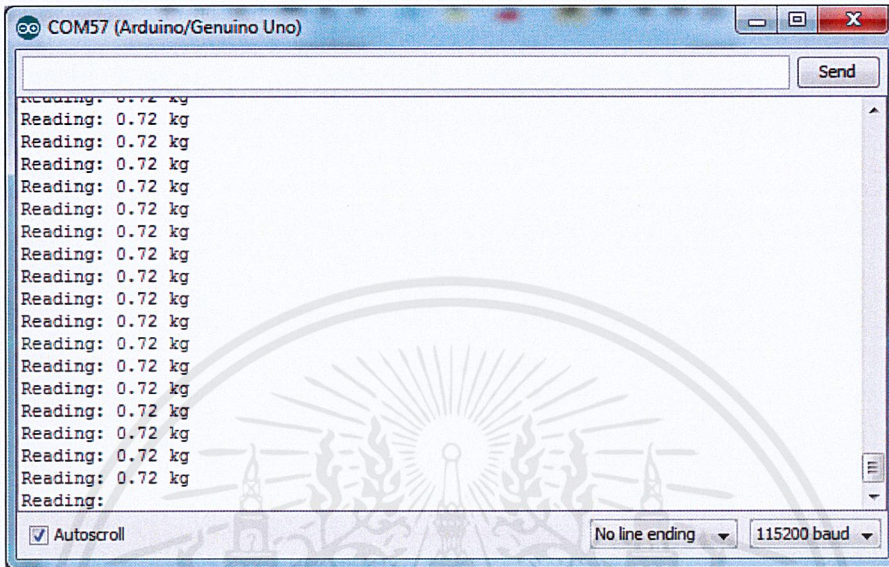
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

>> ให้เรานำเอาค่า Zero Factor และ calibration factor มาใส่ในตัวแปรแทนของเดิม

>> ทดลอง Run Program โดยชั่งขวดน้ำ (1.56 Kg)

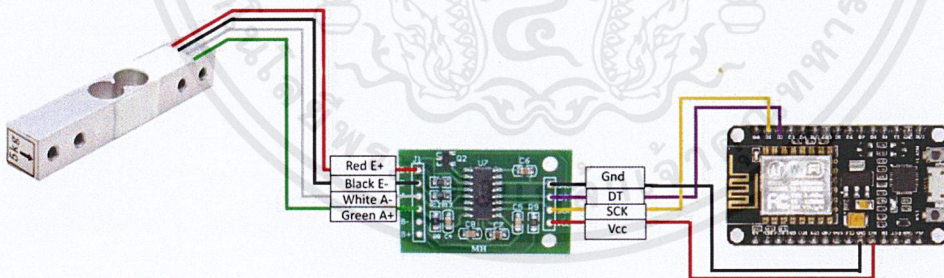
>> ทดลอง Run Program โดยชั่งตุ้มน้ำหนักมาตรฐานน้ำหนัก 500,200,20 กรัม รวมกัน

เท่ากับ 720 กรัม ดังรูปที่ 3.14



รูปที่ 3.14 ตัวอย่างการแสดงผลค่าลูกตุ้มน้ำหนักรวม 720 กรัม

### 3.8 การออกแบบวงจรควบคุม



รูปที่ 3.15 วงจรควบคุมสำหรับเครื่องชั่งเปิด

## บทที่ 4

### สรุปผลการดำเนินงาน

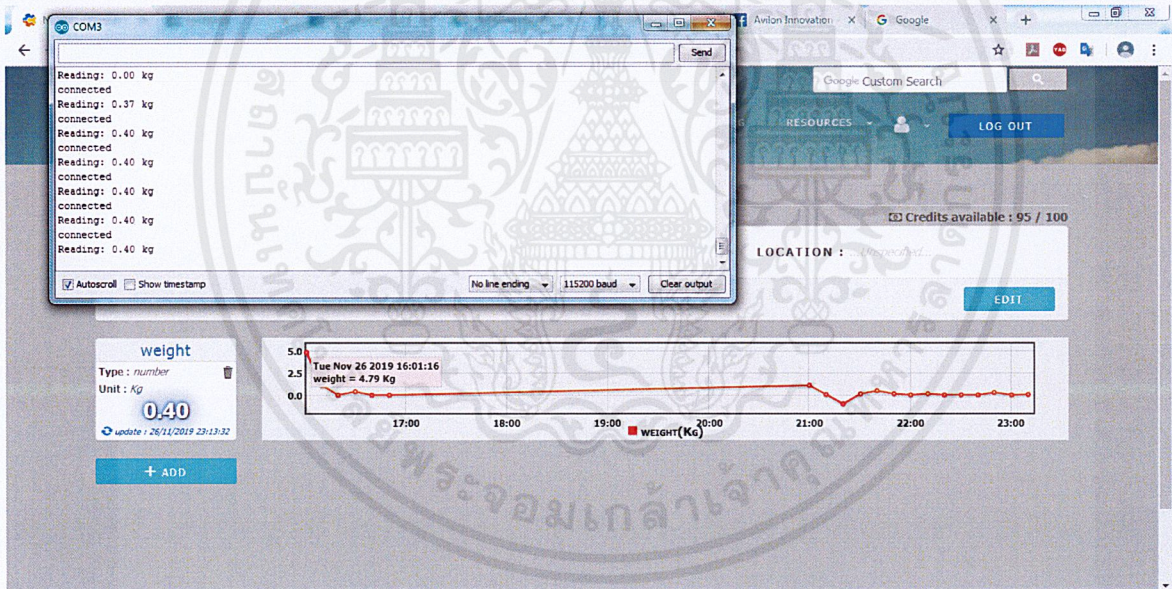
#### 4.1 สรุปผลการดำเนินงาน

##### 4.1.1 การชั่งน้ำหนักของเบ็ด

จากการออกแบบเครื่องชั่งน้ำหนักเบ็ด ผลปรากฏว่า เครื่องชั่งน้ำหนักเบ็ดส่งผลทำให้เป็นขึ้นชั่งวัดน้ำหนักได้ทีละตัวเพราะทางเดินของเบ็ดเดินได้ทีละตัว เพื่อลดการที่เบ็ดจะขึ้นพร้อมกันหลาย ๆ ตัว เพื่อช่วยความสะดวกทางผู้ประกอบการฟาร์มเบ็ดในการอ่านค่าน้ำหนักที่ได้ความแม่นยำ  $\pm 0.03$

##### 4.1.2 การส่งข้อมูลไปยัง NETPIE

เครื่องชั่งเบ็ดที่พัฒนาร่วมกับเทคโนโลยี IoT สามารถส่งข้อมูลของน้ำหนักที่วัดได้ไปยัง Server ของ NETPIE เพื่อทำการแสดงผลได้ตรงตามวัตถุประสงค์ ดังรูปที่ 4.1 โดยค่าที่ส่งไปยัง NETPIE มีค่าที่สอดคล้องกับค่าที่วัดได้จาก Load cell จากการแสดงค่าผ่าน Serial Monitor ของ Arduino



รูปที่ 4.1 แสดงค่าผ่าน FEEDs ของ NETPIE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลการทดลองและข้อเสนอแนะ

#### 5.1 สรุปผลการดำเนินการ

โครงการนี้เป็นการพัฒนาเครื่องชั่งน้ำหนักเปิดร่วมกับเทคโนโลยี Internet of Things เข้ามามีส่วนช่วยในการเก็บข้อมูลไว้บน Cloud และแสดงน้ำหนักของเปิดแบบเรียลไทม์ ผ่านทาง Web server ของ NETPIE

จากการทดลองส่งค่าน้ำหนักจากตัวอย่างทดลองในแต่ละน้ำหนัก พบว่า ค่าที่วัดได้ตรงกับเครื่องวัดชั่งน้ำหนักมาตรฐาน และสามารถส่งข้อมูลไปยัง NETPIE และแสดงค่าน้ำหนักได้ตามวัตถุประสงค์ และในการออกแบบเครื่องชั่งเพื่อให้เปิดขึ้นเครื่องชั่งน้ำหนักทีละตัวสามารถทำได้จริงตามที่ต้องการ ความแม่นยำ  $\pm 0.03$

#### 5.2 ข้อเสนอแนะ

1. ในการส่งข้อมูลไปยัง NETPIE ยังมีข้อจำกัดในเรื่องเวลาของการส่งข้อมูล คือ ส่งข้อมูลทุก ๆ 1 นาที
2. ในการพัฒนาเครื่องตรวจชั่งน้ำหนักเปิด ยังเป็นเพียงการทดสอบการส่งข้อมูลไปยัง NETPIE และพฤติกรรมการเดินขึ้นไปชั่งน้ำหนักของเปิด ยังไม่ได้ทำการติดตั้งจริง เนื่องจากการเจรจาด้านการซื้อ-ขาย ของทางบริษัท

## บรรณานุกรม

[1] การศึกษาเปรียบเทียบต้นทุนและผลตอบแทน จากการทำฟาร์มเลี้ยงปลา ฟาร์มเลี้ยงเป็ด และฟาร์มเลี้ยงปลาและเป็ดแบบผสมผสาน

Available : [https://cuir.car.chula.ac.th/handle/123456789/62567?fbclid=IwAR0xB0lYADGXZH5GW0byuxdY1Q9MylzF9\\_nNzpWBV6ifZxQx8UUOofg6pzs](https://cuir.car.chula.ac.th/handle/123456789/62567?fbclid=IwAR0xB0lYADGXZH5GW0byuxdY1Q9MylzF9_nNzpWBV6ifZxQx8UUOofg6pzs)

[2] คู่มือการใช้งาน NETPIE

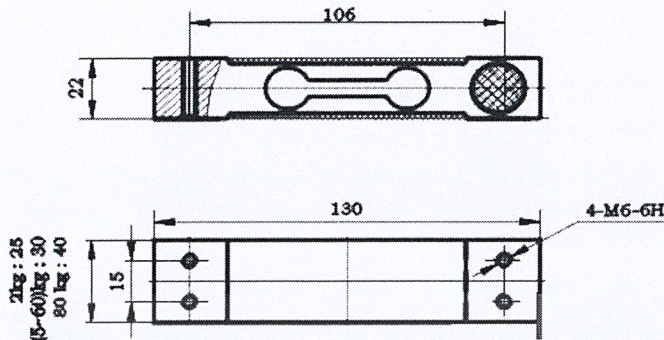
Available : [http://203.159.154.241/innogoth/wp-content/uploads/2017/09/NETPIE-WS\\_v23.pdf](http://203.159.154.241/innogoth/wp-content/uploads/2017/09/NETPIE-WS_v23.pdf)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

## ภาคผนวก ก



### 技术指标

中文	Application	计价秤 Price computing scales												
型号	Model	Y2C-1B												
容量	Capacity	Kg	2	3	5	6	8	10	30	35	40	50	60	80
输出灵敏度	Rated output	MV/V	2.0±0.15											
精度等级	Accuracy class		C2											
最大称定分度值	Maximum number of verification intervals	N <sub>max</sub>	2000										3000	
最小称定分度值	Minimum load cell verification interval	V <sub>min</sub>	E <sub>Max</sub> /5000										E <sub>Max</sub> /7500	
综合误差	Combined error	%RO	< ±0.030										< ±0.020	
蠕变	Creep	%RO/30min	0.03										0.0167	
温度灵敏度漂移	Temperature effect on sensitivity	%RO/°C	0.003										0.0016	
温度零点漂移	Temperature effect on zero	%RO/°C	0.003										0.002	
零点平衡	Zero balance	%RO	±1.0											
输入阻抗	Input resistance	Ω	402±6										1066±10	
输出阻抗	Output resistance	Ω	350±3										1000±10	
绝缘电阻	Insulation resistance	MΩ 50V	5000											
推荐激励电压	Recommended excitation voltage	V	10-15											
温度补偿范围	Compensated temperature range	°C	-10 ~ +40											
工作温度范围	Operating temperature range	°C	-35 ~ +80											
安全超载能力	Safe overload	%RO	150											
极限超载能力	Ultimate overload	%RO	200											
传感器材料	Load cell material		铝合 Aluminim											
平台台面尺寸	Platform size		350×350											
接线电缆	Connecting cable		∅4.2×350mm											
接线方式	Method of connecting wire		红+ 绿+ 蓝- 黄- 棕+ 黑-											

รูปที่ ก 1 Specification Load cell

### Specification :

- Capacity: 20KG
- Rated output(MV/V): 2.0C2±0.15
- Accuracy class: C2
- Maximum number of laod cell verification intervals(N max): 2000
- Minimum number of laod cell verification intervals(Vmin): EMax/5000
- Combined erro(%RO): <±0.030
- Creep(%RO/30min): 0.03
- Temperature effect on sensitivity(%RO/°C): 0.0016
- Temperature effect on zero(%RO/°C): 0.003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Zero balance(%RO): 1.0
- Input resistance( $\Omega$ ): 402±6
- Output resistance( $\Omega$ ): 350±3
- Insulation resistance(M $\Omega$ <50V>): 5000
- Recommended excitation voltage(V): 10~15
- Compensated temperature range(°C): -10~+40
- Operating temperature range(°C): -35~+80
- Safe overload(%RO): 150
- Ultimate overload(%RO): 200
- Load cell material: Aluminium
- Platform size: 350x350mm
- Connecting cable:  $\phi$ 4.2x350mm
- Method of connecting wire: Red(+),Black(-),Green(+),White(-)

