ถังขยะอัจฉริยะ

Smart Bin

นายจิตรภานุ วุฒิกรวิภาค
Jitpanu Wutilornwipak

นายธัญพล โต๊ะซา
Tunyapon Tohsa

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2560

ถังขยะอัจฉริยะ

Smart Bin

โดย
นายจิตรภานุ วุฒิกรวิภาค
นายธัญพล โต๊ะซา

อาจารย์ที่ปรึกษา
รศ.ดร.สุรพันธุ์ เอื้อไพบูลย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2560

รายงานวิชาโครงงาน 2 ปีการศึกษา 2560

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์
คณะ วิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง ถังขยะอัจริยะ
Smart Bin
ผู้จัดทำ นายจิตรภานุ วุฒิกรวิภาค รหัสประจำตัว 57010179
นายธัญพล โต๊ะชา รหัสประจำตัว 57010612

รายงานนี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

(รองศาสตราจารย์ ดร. สุรพันธุ์ เอื้อไพบูลย์)
อาจารย์ที่ปรึกษา

บทคัดย่อ

ปริญญานิพนธ์เรื่อง	ถังขยะอัจฉริยะ
นักศึกษา	นายจิตรภานุ วุฒิกรวิภาค รหัสประจำตัว 57010179
	นายธัญพล โต๊ะซา รหัสประจำตัว 57010612
ปริญญา	วิศวกรรมศาสตร์บัณฑิต
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2560
อาจารย์ที่ปรึกษาโครงงาน	รองศาสตราจารย์ ดร. สุรพันธุ์ เอื้อไพบูลย์

บทคัดย่อ

โครงงานนี้มีที่มาจากปัญหาการทิ้งขยะไม่เป็นที่และไม่แยกประเภทขยะทำให้เกิดมลพิษต่อสิ่งแวดล้อมเราจึงได้ทำเครื่องแยกขยะอัจฉริยะขึ้นมาเพื่อเป็นการปลูกฝังให้เด็กรุ่นใหม่ตระหนักถึงการแยกขยะโดยจะมีการคำนวณเงินตามปริมาณขยะประเภทขวดที่กำหนดไว้ซึ่งเครื่องนี้สามารถรับขยะได้เป็น 4ประเภท ขวดแก้ว ขวดพลาสติก กระป๋องน้ำและกล่องนมและจะคิดเงินเฉพาะขวดพลาสติกเมื่อครบตามจำนวนที่ได้กำหนดไว้

ABSTRACT

Thesis title	Smart Bin		
Students	Mr.Jitpanu	Wutikornwipak	Student ID 57010179
	Mr.Tunyapon	Tohsa	Student ID 57010612
Degree	Bachelor of Engineering		
Program	Electronics Engineering		
Academic Year	2017		
Project Advisor	Assac.Prof. Dr.Surapan Airphaiboon		

ABSTRACT

This project is based on the problem of waste. It is not classified waste causes pollution the environment, we have made Smart bin was to teach the children new generation recalled segregated by it.This machine recive waste four type such as glass bottle,plastic bottle,canand milk box but it was calculated plastic bottles only. The calculation of the money of waste bottles, which are scheduled.

กิตติกรรมประกาศ

โครงงานนี้ สำเร็จลุล่วงได้ด้วยดี ต้องขอขอบคุณรองศาสตราจารย์ ดร. สุรพันธุ์ เอื้อไพบูลย์อาจารย์ที่ปรึกษาที่คอยช่วยเหลือในการทดลองวงจรและให้ความรู้และคอยแนะนำต่างๆ คณะผู้จัดทำโครงงานขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณพ่อแม่ของคณะผู้จัดทำที่คอยเป็นกำลังใจหลักและเป็นผู้สนับสนุนเงินทุนหลักในการทำโครงงาน รวมไปถึงเพื่อนๆที่คอยช่วยเหลือ ให้คำแนะนำ รวมทั้งช่วยกันในการแก้ปัญหาที่เกิดขึ้นในระหว่างการทำโครงงานนี้ทั้งหมดทำให้ผลของโครงงานสำเร็จลุล่วงไปด้วยดี

สุดท้ายนี้ คณะผู้จัดทำหวังว่าโครงงานนี้จะเป็นประโยชน์สำหรับผู้สนใจและผู้นำผลงานนี้ไปใช้งานได้

จิตรภานุ วุฒิกรวิภาค

ธัญพล. โต๊ะซา

สารบัญ

สารบัญ(ต่อ)

สารบัญรูป

บทที่ 1
บทนำ

1.1. ที่มาและความสำคัญของปัญหา

เนื่องด้วยในปัจจุบันชอบทิ้งขยะไว้รวมกันโดยที่ไม่ยอมแยกขยะทำให้บางครั้งในการทำลายขยะมีความลำบากเพราะขยะคนละประเภทกันการจะทำลายก็ต้องคนละวิธีกันดังนั้นจึงทำถังขยะนี้ขึ้นมาเพื่อที่จะช่วยให้เด็กรู้จักการแยกขยะ

1.2. ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

ในการทำโครงงานครั้งนี้จัดทำขึ้นเพื่อ

1.2.1. เป็นการปลูกฝังให้เด็กรู้จักการแยกขยะ

1.2.2. นำผลงานจากโครงงานไปใช้ในชีวิตประจำวัน

1.2.3. ทำให้เด็กรู้คุณค่าของขยะที่นำไปทิ้ง

1.3. สมมุติฐานการศึกษา

สามารถแยกขยะเป็นประเภทได้ซึ่งเครื่องตัวนี้จะสามารถแยกประเภทของขวดได้เป็น 4 ประเภทได้แก่ ขวดแก้ว ขวดพลาสติก กล่องนม และกระป๋องแล้วถ้าแยกขวดพลาสติกได้ตามที่กำหนดจะให้เงินออกมา

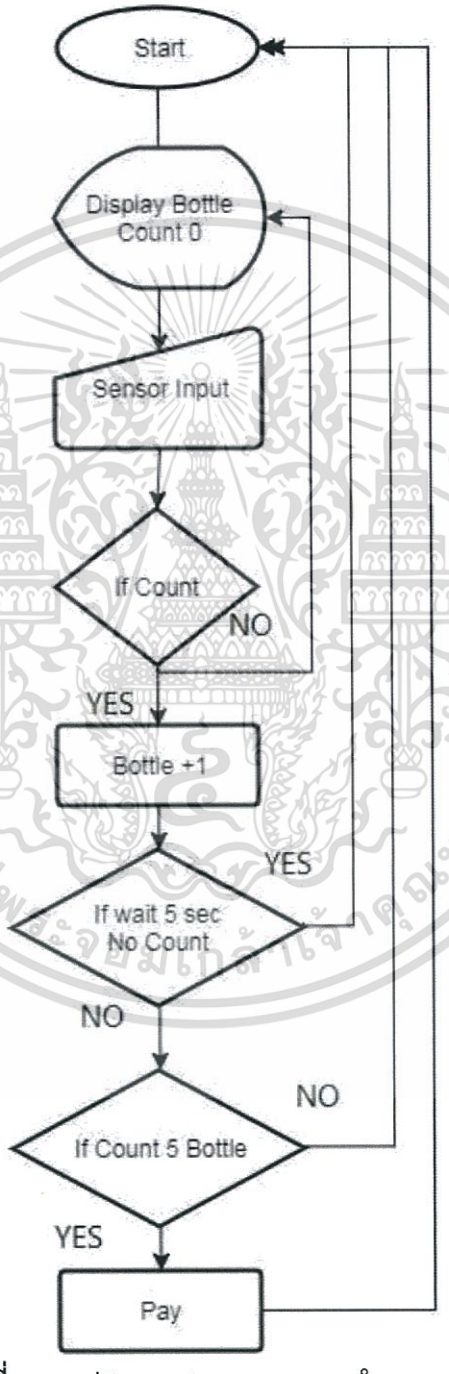
1.4. ขอบเขตของการศึกษา

1.4.1. สามารถแยกขยะออกเป็น4ประเภท

1.4.2. ในกรณีที่เป็นขวดจะมีการแบ่งประเภทคือขวดแก้ว ขวดพลาสติก กล่องนมและกระป๋อง

1.4.3. เมื่อนำขวดพลาสติกไปทิ้งได้ตามจำนวนที่กำหนดจะมีเงินทอนออกมา

1.5 flow chart



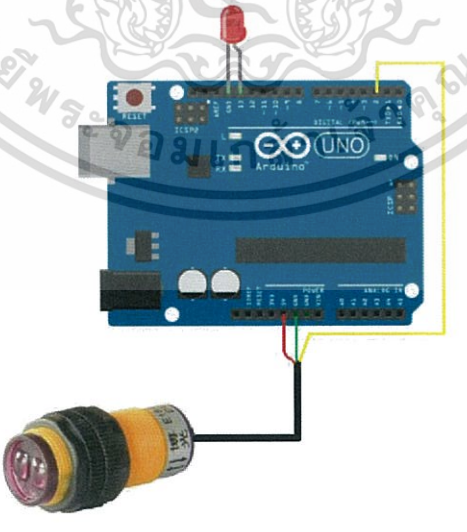
รูปที่1.1 รูปflow chartของการทำงาน

บทที่ 2
หลักการและทฤษฎี

2.1. Infrared Proximity Sensor

รูปที่2.1 Infrared Proximity Sensor

Infrared Proximity Sensor เป็นเซนเซอร์วัดระยะที่ใช้หลักการสะท้อนของคลื่นอินฟาเรด สามารถกำหนดระยะในการทำงานได้โดยปรับค่าที่ Potentiometer ทำงานในช่วง 3-80 CM Output เป็นแบบ Digital ใช้ไฟเลี้ยง 5V ระยะทำงานไม่เกิน 45CM



รูปที่2.2 การต่อ Infrared Proximity Sensor กับ Arduino

2.2. LCD 16x2

รูปที่2.3 LCD 16x2

คำว่า LCD ย่อมาจากคำว่า Liquid Crystal Display ซึ่งเป็นจอที่ทำมาจากผลึกคริสตอลเหลว หลักการคือด้านหลังจอจะมีไฟส่องสว่าง หรือที่เรียกว่า Backlight อยู่ เมื่อมีการปล่อยกระแสไฟฟ้าเข้าไปกระตุ้นที่ผลึก ก็จะทำให้ผลึกโปร่งแสง ทำให้แสงที่มาจากไฟ Backlight แสดงขึ้นมาบนหน้าจอ ส่วนอื่นที่โดนผลึกปิดกั้นไว้ จะมีสีที่แตกต่างกันตามสีของผลึกคริสตอล เช่น สีเขียว หรือ สีฟ้า ทำให้เมื่อมองไปที่จอก็จะพบกับตัวหนังสือสีขาว แล้วพบกับพื้นหลังสีต่างๆกันจอ LCD จะแบ่งเป็น 2 แบบใหญ่ๆตามลักษณะการแสดงผลดังนี้

1. **Character LCD** เป็นจอที่แสดงผลเป็นตัวอักษรตามช่องแบบตายตัว เช่น จอ LCD ขนาด 16x2 หมายถึงใน 1 แถว มีตัวอักษรใส่ได้ 16 ตัว และมีทั้งหมด 2 บรรทัดให้ใช้งาน ส่วน 20x4 จะหมายถึงใน 1 แถว มีตัวอักษรใส่ได้ 20 ตัว และมีทั้งหมด 2 บรรทัด

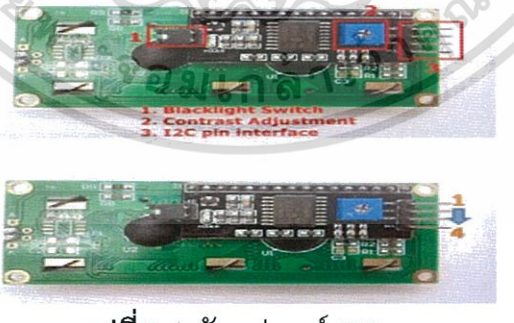
2. **Graphic LCD** เป็นจอที่สามารถกำหนดได้ว่าจะให้แต่ละจุดบนหน้าจอกั้นแสง หรือปล่อยแสงออกไป ทำให้จอนี้สามารถสร้างรูปขึ้นมาบนหน้าจอได้ การระบุขนาดจะระบุในลักษณะของจำนวนจุด (Pixels) ในแต่ละแนว เช่น 128x64 หมายถึงจอที่มีจำนวนจุดตามแนวนอน 128 จุด และมีจุดตามแนวตั้ง 64 จุด

การเชื่อมต่อจะมีด้วยกัน 2 แบบ

- **การเชื่อมต่อแบบขนาน** - เป็นการเชื่อมต่อจอ LCD เข้ากับบอร์ด Arduino โดยตรง โดยจะแบ่งเป็นการเชื่อมต่อแบบ 4 บิต และการเชื่อมต่อแบบ 8 บิต ใน Arduino จะนิยมเชื่อมต่อแบบ 4 บิต เนื่องจากใช้สายในการเชื่อมต่อน้อยกว่า
- **การเชื่อมต่อแบบอนุกรม** - เป็นการเชื่อต่อกับจอ LCD ผ่านโมดูลแปลงรูปแบบการเชื่อมต่อกับจอ LCD จากแบบขนาน มาเป็นการเชื่อมต่อแบบอื่นที่ใช้สายน้อยกว่า เช่น การใช้โมดูล I2C Serial Interface จะเป็นการนำโมดูลเชื่อมเข้ากับตัวจอ LCD แล้วใช้บอร์ด Arduino เชื่อมต่อกับบอร์ดโมดูลผ่านโปรโตคอล I2C ทำให้ใช่สายเพียง 4 เส้น ก็ทำให้หน้าจอแสดงผลข้อความต่างๆออกมาได้

2.2.1 การต่อ LCD โดยใช้ I2C

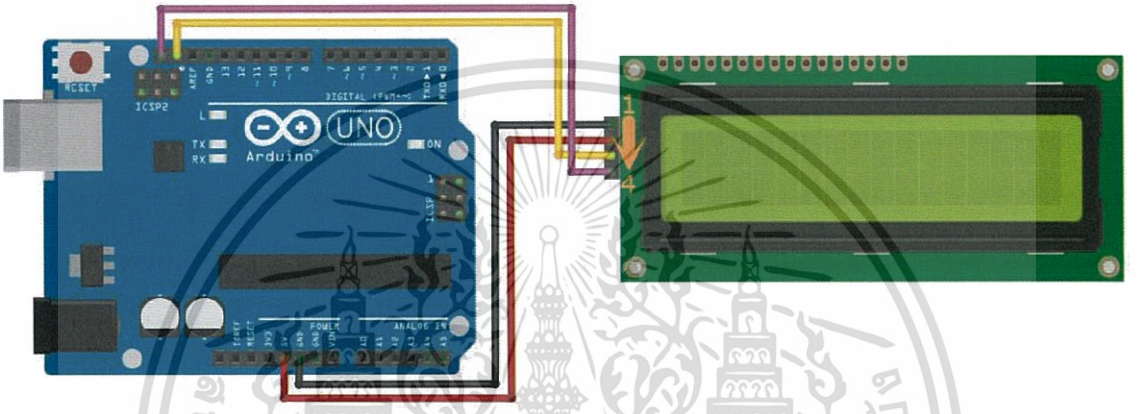
จอ LCD ที่มีการเชื่อมต่อแบบ I2C หรือเรียกอีกอย่างว่าการเชื่อมต่อแบบ Serial จะเป็นจอ LCD ธรรมดาทั่วไปที่มาพร้อมกับบอร์ด I2C Bus ที่ทำให้การใช้งานได้สะดวกยิ่งขึ้นและยังมาพร้อมกับ VR สำหรับปรับความเข้มของจอ ในรูปแบบ I2C จะใช้ขาในการเชื่อมต่อกับ Microcontroller เพียง 4 ขา (แบบ Parallel ใช้ 16 ขา) ซึ่งทำให้ใช้งานได้ง่ายและสะดวกมากยิ่งขึ้น



รูปที่2.4 ตัวอุปกรณ์ I2C

ในการควบคุมหรือสั่งงาน โดยทั่วไปจอ LCD จะมีส่วนควบคุม (Controller) อยู่ในตัวแล้ว ผู้ใช้สามารถส่งรหัสคำสั่งสำหรับควบคุมการทำงานของจอ LCD (I2C) เช่นเดียวกันกับจอ LCD แบบธรรมดา คือรหัสคำสั่งที่ใช้ในการควบคุมนั้นเหมือนกัน แต่ต่างกันตรงที่รูปแบบในการรับส่งข้อมูล

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



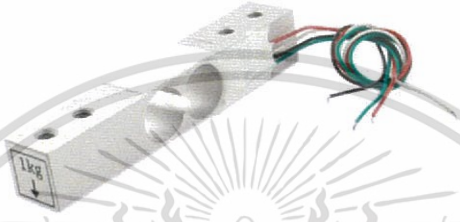
รูปที่2.5 การต่อ LCD กับ I2C

2.3 Servo Motor

รูปที่2.6 Servo Motor

สำหรับ Servo motor นั้น นิยามของมันคือ เป็นมอเตอร์ที่มีการควบคุมการเคลื่อนที่ของมัน (State) ไม่ว่าจะเป็นระยะ ความเร็ว มุมการหมุน โดยใช้การควบคุมแบบป้อนกลับ (Feedback control)

2.4 Load cell



รูปที่2.7 Load cell

โหลดเซล คืออุปกรณ์อิเล็กทรอนิกส์ที่ใช้แปลงค่าของแรงไปเป็นสัญญาณไฟฟ้า (ทรานส์ซีดิวเซอร์) การแปลงค่านี้ไม่ใช่การแปลงค่าโดยตรงหากแต่เกิดขึ้นสองขั้นตอน จากการแปลงค่าทางกลศาสตร์ แรงจะถูกตรวจจับได้จากการเปลี่ยนรูปร่างของสเตนเกจ และสเตนเกจแปลงค่าการเปลี่ยนรูปร่าง (ความเครียด) นี้ไปเป็นสัญญาณไฟฟ้า โหลดเซลมักจะประกอบไปด้วยสเตนเกจสี่ตัวซึ่งจัดเรียงวงจรในรูปแบบของวงจรวิจสโตน บริดจ์ แต่โหลดเซลที่ประกอบด้วยสเตนเกจเพียงหนึ่งหรือสองตัวก็มีใช้เช่นกัน สัญญาณไฟที่จ่ายออกไปนี้มักจะมีขนาดเพียงไม่กี่มิลลิโวลต์และต้องการการขยายสัญญาณด้วยการใช้อุปกรณ์ขยายสัญญาณก่อนที่จะถูกนำไปใช้งานได้ ขั้นตอนวิธีเพื่อคำนวณหาค่าแรงโหลดเซลเกือบ 80% นั้นเป็นชนิดสเตนเกจ โดยโหลดเซลแบบสเตนเกจก็แบ่งเป็นอีก 2 ประเภทใหญ่ คือ โหลดเซลแบบใช้แรงกด ออกแบบมาเพื่อใช้แรงกดลงบนตัวโหลดเซลล์ และ โหลดเซลแบบใช้แรงดึง ออกแบบมาเพื่อใช้แรงดึงตัวโหลดเซลล์ออกจากกัน

2.4.1 Load cell แบบใช้แรงกด

Single End Shear Beam-นิยมใช้ในการชั่งน้ำหนักในถัง น้ำหนักตั้งแต่ 250 กิโลกรัม ถึง 10 ตัน เช่น การชั่งน้ำหนักหิน-ทรายในถัง ก่อนปล่อยลงไปผสมกับซิเมนต์และน้ำในแพลน

คอนกรีต เป็นต้น โหลดเซลประเภทนี้ใช้งานโดยยึดปลายด้านหนึ่งเข้ากับฐาน และนำถังวางลงบนปลายอีกด้านหนึ่ง

Double End Shear Beam-โหลดเซลประเภทนี้เหมือนกับนำ Single End Shear Beam จำนวน 2 ตัวมารวมกัน ซึ่งทำให้มีจำนวนสเตนเกจมากขึ้น ทำให้ได้ความละเอียดมากขึ้น การติดตั้งเป็นการยึดปลายทั้งสองข้างด้วยสกรูติดกับฐาน แล้วนำถังมาวางตรงกลาง โดยมีลูกบอลและเบ้ายึดตัวถังและโหลดเซล เพื่อให้ถังสามารถขยับได้แต่ไม่หลุดหล่นไป โหลดเซลประเภทนี้นิยมใช้ในงานชั่งที่มีน้ำหนักมาก เช่นถังหรือไซโลที่มีขนาดใหญ่ ตั้งแต่ 10 ตัน ถึง 50 ตัน โดยจะติดตั้งไว้ที่ขาของถังหรือไซโล

Single Point-ออกแบบมาสำหรับงานขนาดเล็ก น้ำหนักตั้งแต่ 2 กิโลกรัม ถึง 800 กิโลกรัม ใช้งานโดยยึดโหลดเซลเข้าที่จุดศูนย์กลาง

Bending Beam-โหลดเซลประเภทนี้มีโครงสร้างคล้ายสปริง ทำงานโดยการแปลงแรงบิดที่กดที่ปลายด้านหนึ่ง ซึ่งจะให้สัญญาณได้ดีที่ขนาดแรงกดไม่มาก ตั้งแต่ 25 กิโลกรัม ถึง 500 กิโลกรัม

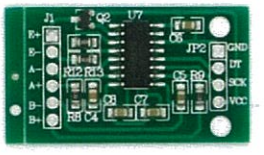
Pancake-ชื่อเรียกก็มาจากรูปทรงกลมของโหลดเซลประเภทนี้ Pancake เป็นโหลดเซลที่ใช้ได้ทั้งกับแรงกดและแรงดึง ขนาดตั้งแต่ 500 กิโลกรัม ถึง 500 ตัน เป็นโหลดเซลที่มีความแม่นยำสูง โดยค่า Linearity และ Hysteresis อยู่ในระดับ 0.05% เนื่องจากมีจำนวนสเตนเกจมากกว่าโหลดเซลชนิดอื่น

Canister-โหลดเซลรูปทรงกระบอก เหมือนกระป๋อง ใช้รับแรงกด มีความแม่นยำสูง โดยค่า Linearity และ Hysteresis อยู่ในระดับ 0.05% จึงนิยมใช้ทำเครื่องชั่งทั่วไปที่ต้องการความแม่นยำสูงๆ ไปจนถึงเครื่องชั่งรถบรรทุก มีขนาดตั้งแต่ 200 กิโลกรัม ถึง 20 ตัน

2.4.2 Load cell แบบใช้แรงดึง

S Beam-โหลดเซลประเภทนี้ใช้งานโดยแขวนถังที่ต้องการชั่งที่ด้านล่าง มีขนาดตั้งแต่ 2 กิโลกรัม ถึง 5 ตัน

2.5 โมดูล HX711



รูปที่2.8 HX711

โมดูล HX711 เป็นโมดูลสำหรับขยายสัญญาณจาก Load Cell เซ็นเซอร์วัดน้ำหนัก ซึ่งปกติมาค่าน้อยมากๆ ตัวโมดูลนี้จะขยายสัญญาณ ออกเป็นสัญญาณดิจิตอล 24bit I2C ทำให้สามารถนำ Arduino NodeMCU Raspberry Pi หรือ MCU อื่นๆมาอ่านค่าน้ำหนักได้ง่ายๆ สายเชื่อมต่อเข้า MCU มี 2 เส้นและไฟเลี้ยง 2 เส้น สามารถใช้ไฟเลี้ยงได้ตั้งแต่ 2.6-5.5v และอีกด้านของโมดูลสามารถต่อกับ LoadCell ได้เลย ตามภาพประกอบ

2.6 โปรแกรม Adruino

Arduino เป็นภาษาอิตาลี อ่านว่า ***อาดุอีโน่*** หรือ จะเรียกว่า ***อาดุยโน่*** ก็ได้ไม่ผิด เพราะไม่ใช่ภาษาบ้านเรา Arduino คือ Open-Source Platform สำหรับการสร้างต้นแบบทางอิเล็กทรอนิกส์ โดยมีจุดมุ่งหมายให้ Arduino Platform เป็น Platform ที่ง่ายต่อการใช้งาน, โดย Arduino Platform ประกอบไปด้วย

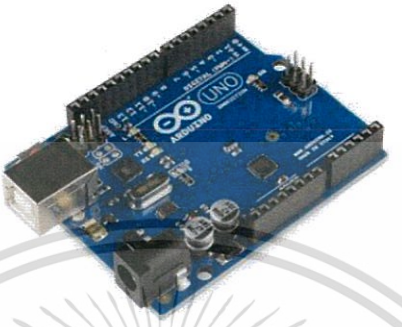
1.ส่วนที่เป็น Hardware คือ

- บอร์ดอิเล็กทรอนิกส์ขนาดเล็กที่มีไมโครคอนโทรลเลอร์ (MCU) เป็นชิ้นส่วนหลัก ถูกนำมาประกอบร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ เพื่อให้ง่ายต่อการใช้งาน หรือที่เรียกกันว่า **บอร์ด Arduino** โดยบอร์ด Arduino เองก็มีหลายรุ่นให้เลือกใช้ โดยในแต่ละรุ่นอาจมีความแตกต่างกันในเรื่องของขนาดของบอร์ด หรือสเปค เช่น จำนวนของขารับส่งสัญญาณ, แรงดันไฟที่ใช้, ประสิทธิภาพของ MCU เป็นต้น

2.ส่วนที่เป็น Software คือ

- ภาษา Arduino เป็นภาษาสำหรับเขียนโปรแกรมควบคุม MCU, มีไวยากรณ์แบบเดียวกับภาษา C/C++

- Arduino IDE เป็นเครื่องมือสำหรับเขียนโปรแกรมด้วยภาษา Arduino, คอมไพล์โปรแกรม (Compile) และอัพโหลดโปรแกรมลงบอร์ด (Upload)



รูปที่2.9 Arduino Uno

บทที่ 3
วิธีการดำเนินงาน

3.1 การต่อขา LCD16X2 กับ arduino

LCD ชนิด I2Cใช้ LCD ชนิด I2C เพราะมีจำนวนขาน้อยประหยัดการต่อพอร์ตที่ไม่จำเป็นซึ่ง LCD ชนิด I2C มี 4 ขาLCD จะใช้ในการแสดงค่าสัญญาณอนาล็อกที่ได้จากเอาพุทของวงจร FSR เพื่อที่จะกำหนด ความดันต่ำสุดและสูงสุดโดยการต่อขา LED

-ขา Vcc ต่อเข้ากับขา +5V

-ขา GND ต่อเข้ากับขา GND

-ขา SDA ต่อเข้ากับขา A4

-ขา SCL ต่อเข้ากับขา A5

3.2 การต่อขา Sensor กับ Arduino

การต่อวงจร Infrared Sensor

Arduino	Sensor
5V	VCC (แดง)
GND	GND (เขียว)
pin 2	A0 (เหลือง)

3.3 การต่อขา Servo กับ Arduino

Servo		Arduino
Output	สีส้ม	pin 9
Vcc	สีแดง	5V
Gnd	สีน้ำตาล	Gnd

3.4 การต่อ HX711 กับ Arduino

HX711	Arduino
Vcc	5V
SCK	A0

GND	GND
DT	A1

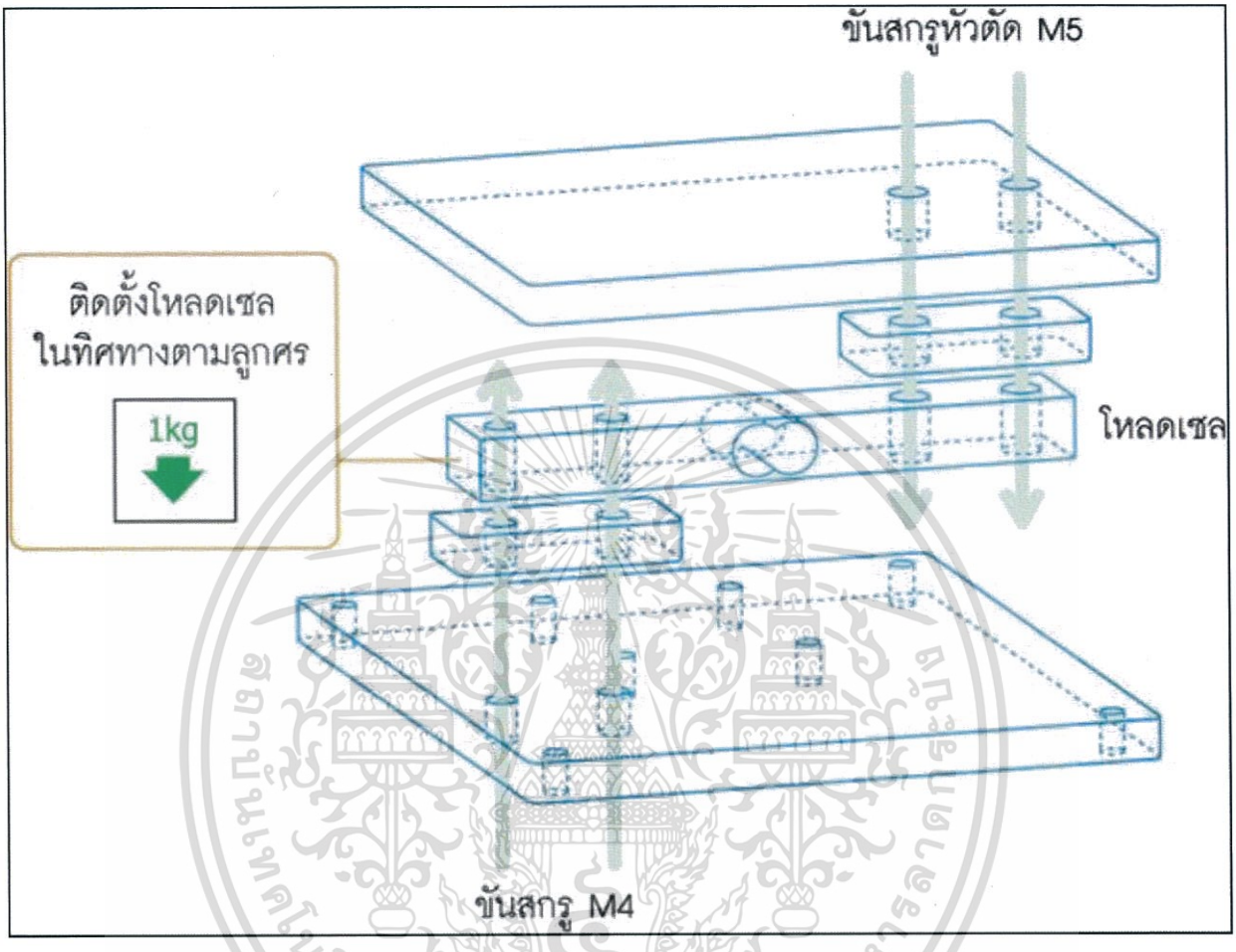
3.5 Arduino UNO

เลือกใช้ Arduino UNO R3 เพราะง่ายต่อการใช้งานและเป็นบอร์ดสำหรับผู้ที่เริ่มต้นใช้งาน Arduino มีไลบรารีที่รองรับมากโดยไม่ต้องเปลี่ยน pin ที่ขา

รายละเอียด Arduino UNO R3

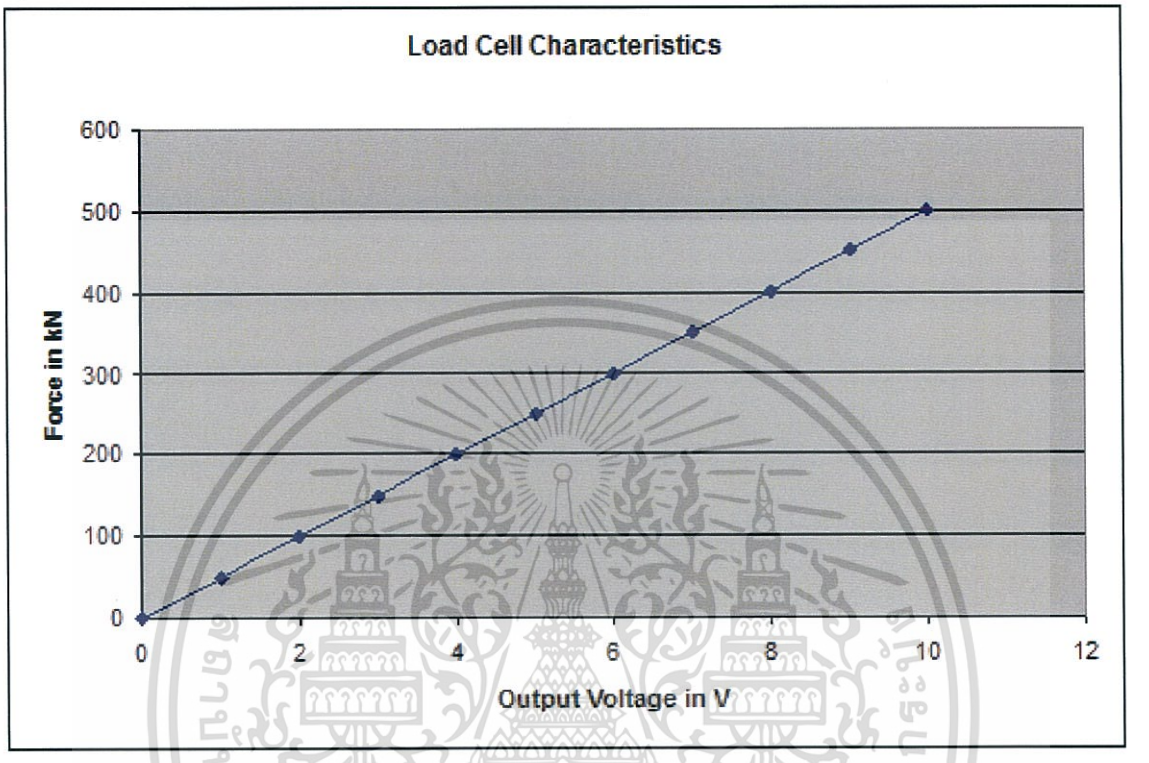
ไมโครคอนโทรนเลอร์	ATmega328
แหล่งจ่ายไฟ	5V
ไฟเข้า(แนะนำ)	7-12V
ไฟเข้า (จำกัดไว้ที่)	6-20V
ขาดิจิตอล I/O	14 ขา (6 รองรับเอาต์พุตแบบ PWM)
ขาอานาล็อกอินพุต	6 ขา
กระแสไฟฟ้า DC ต่อขา I/O	40 mA
กระแสไฟฟ้าออก DC สำหรับขา 3.3V	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

3.6 การต่อHX711กับLoadCell



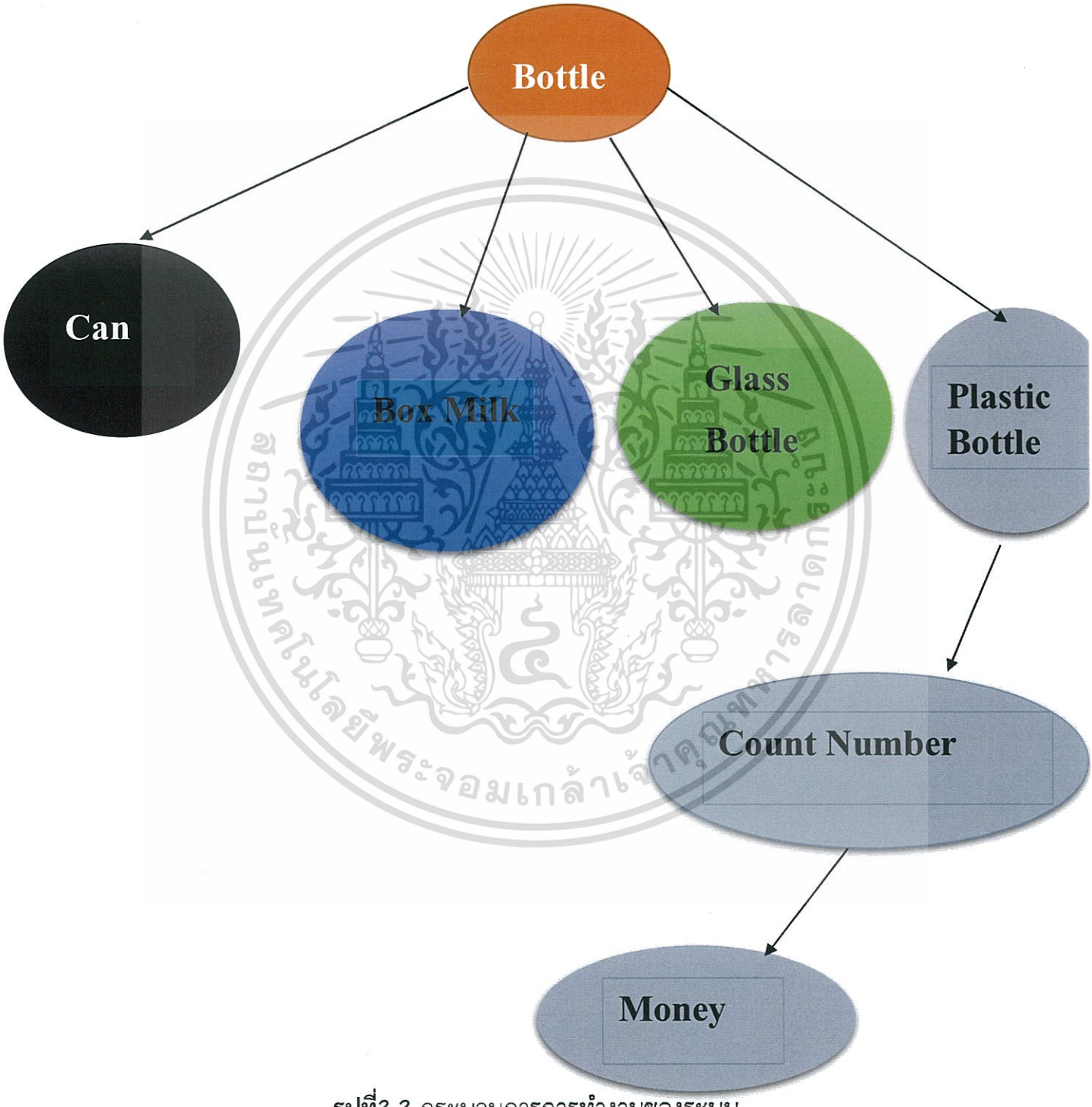
รูปที่3.1 องค์ประกอบของLoad cell

3.7 กราฟ Output ของ Load Cell



รูปที่3.2 กราฟoutputของload cell

3.8 กระบวนการการทำงานของระบบ

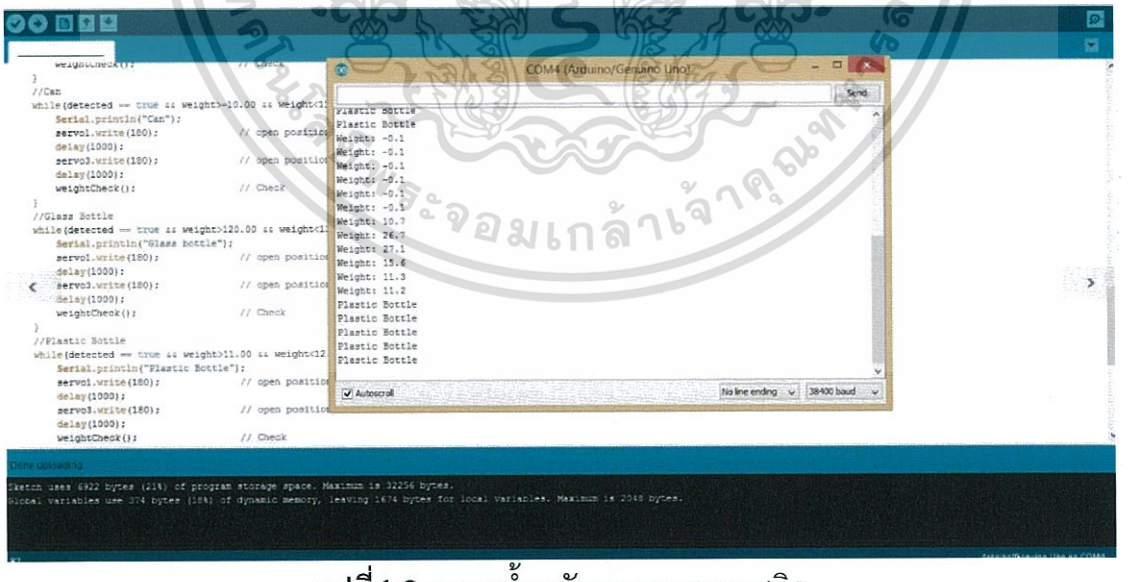


รูปที่3.3 กระบวนการการทำงานของระบบ

บทที่ 4
ผลการทดลอง

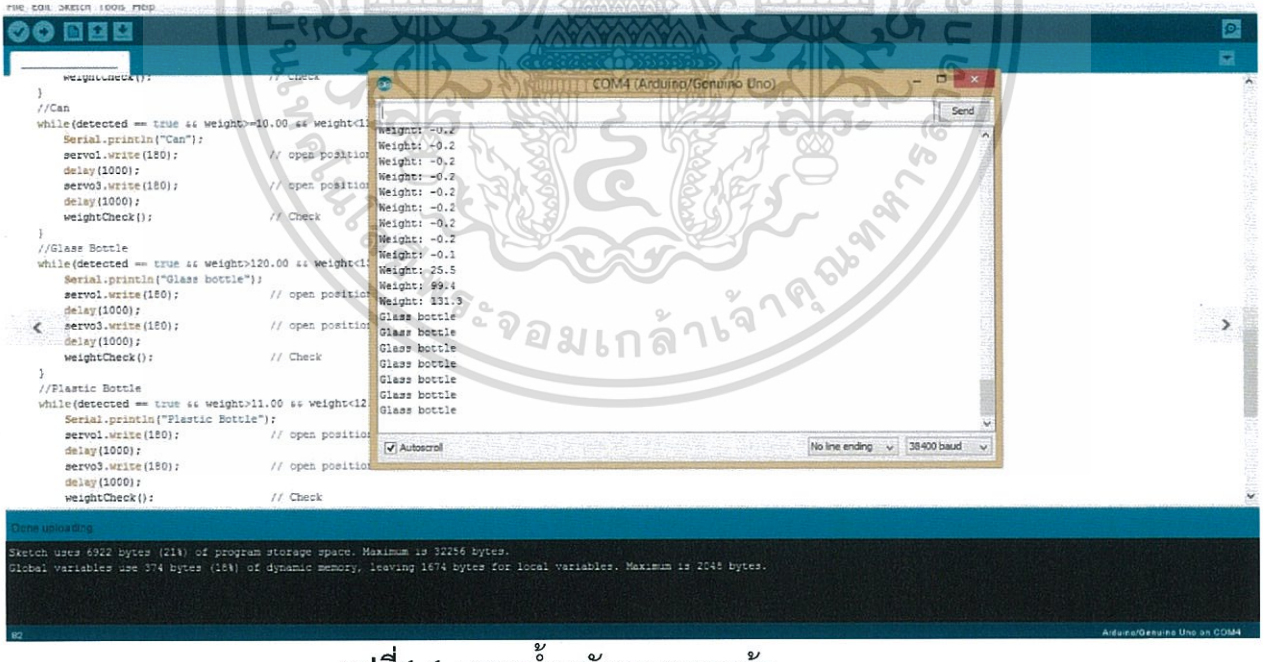
4.1 การทดลองตอนแยกขยะ

รูปที่4.1 นำขวดพลาสติกมาชั่ง



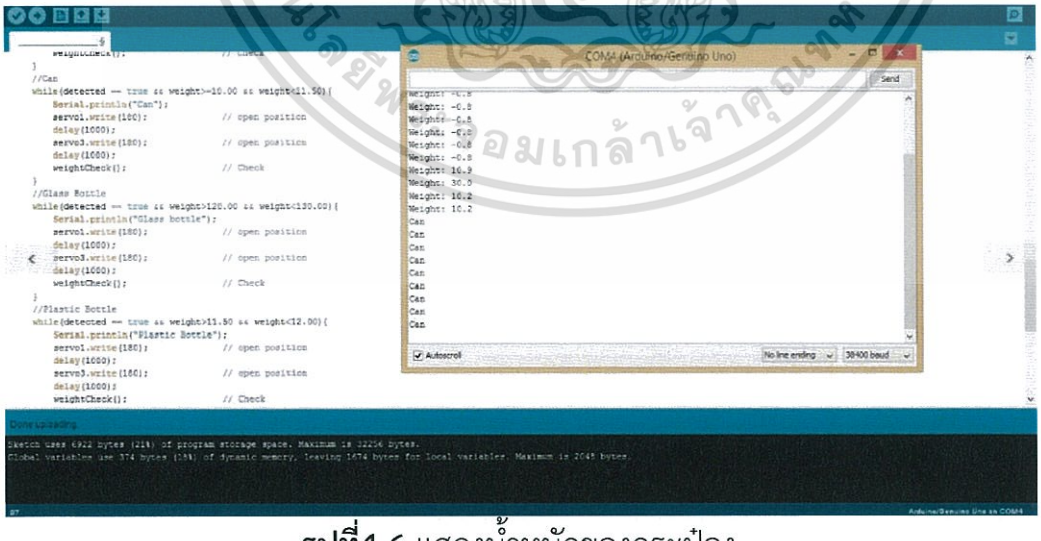
รูปที่4.2 แสดงน้ำหนักของขวดพลาสติก

รูปที่4.3 นำขวดแก้วมาชั่ง



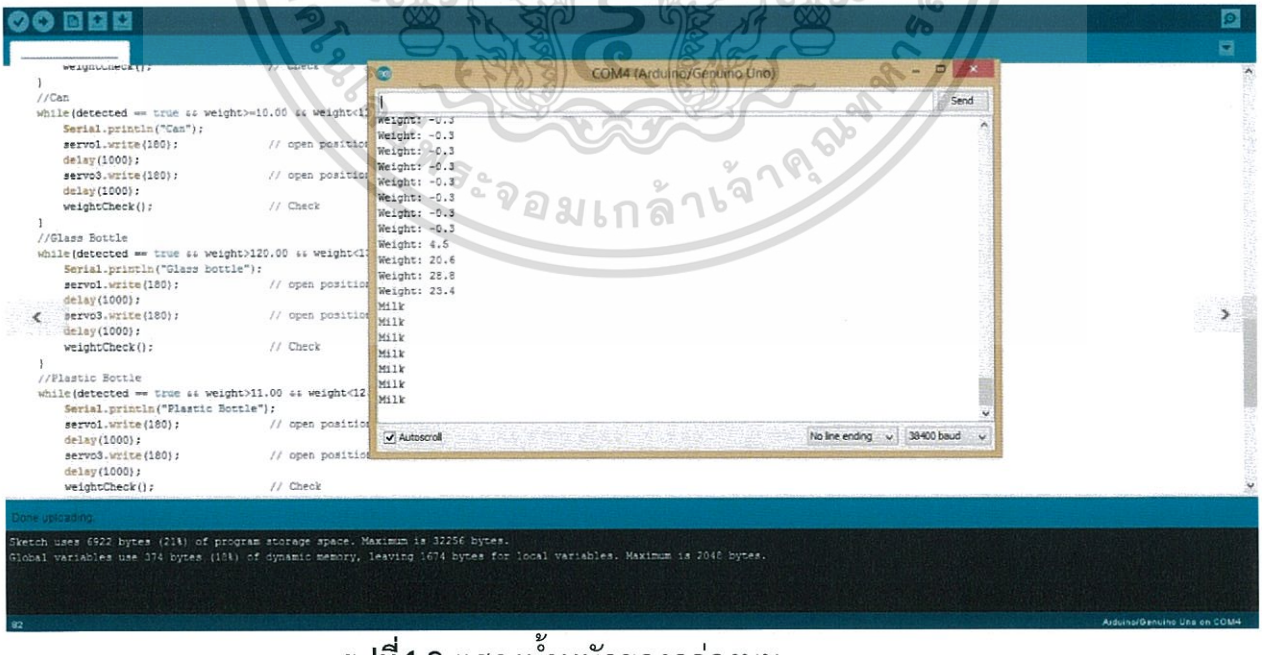
รูปที่4.4 แสดงน้ำหนักของขวดแก้ว

รูปที่4.5 นำกระป๋องมาชั่ง



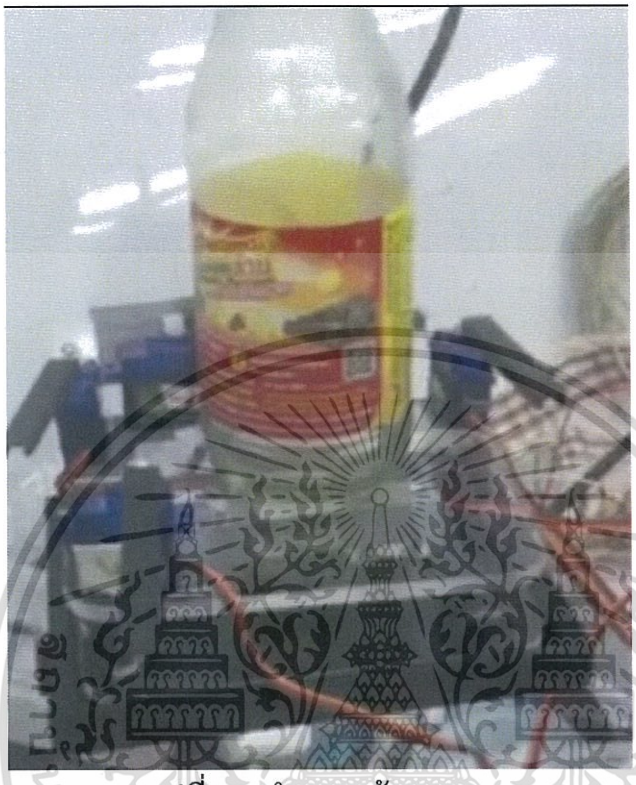
รูปที่4.6 แสดงน้ำหนักของกระป๋อง

รูปที่4.7 นำกล่องนมมาชั่ง



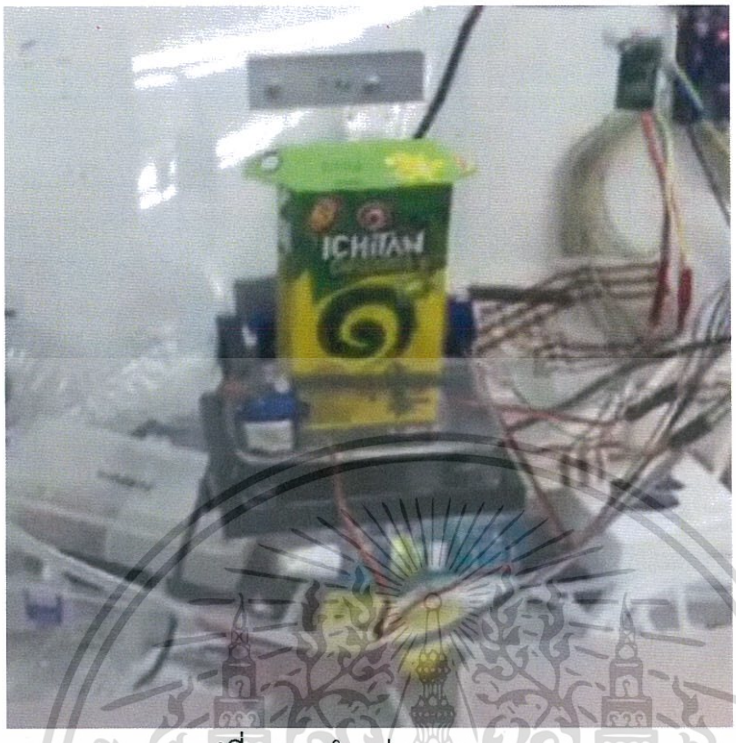
รูปที่4.8 แสดงน้ำหนักของกล่องนม

4.2 ตอนนำขยะมาแยก

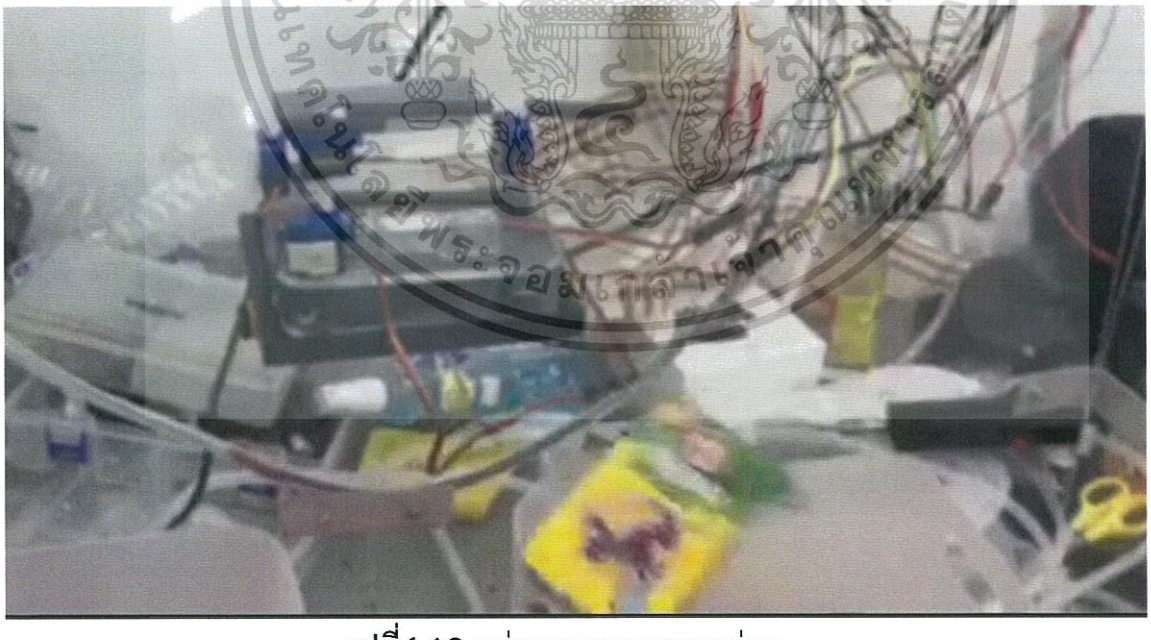


รูปที่4.9 นำขวดแก้วมาแยก

รูปที่4.10 ขวดแก้วถูกนำมาแยกลงกล่อง

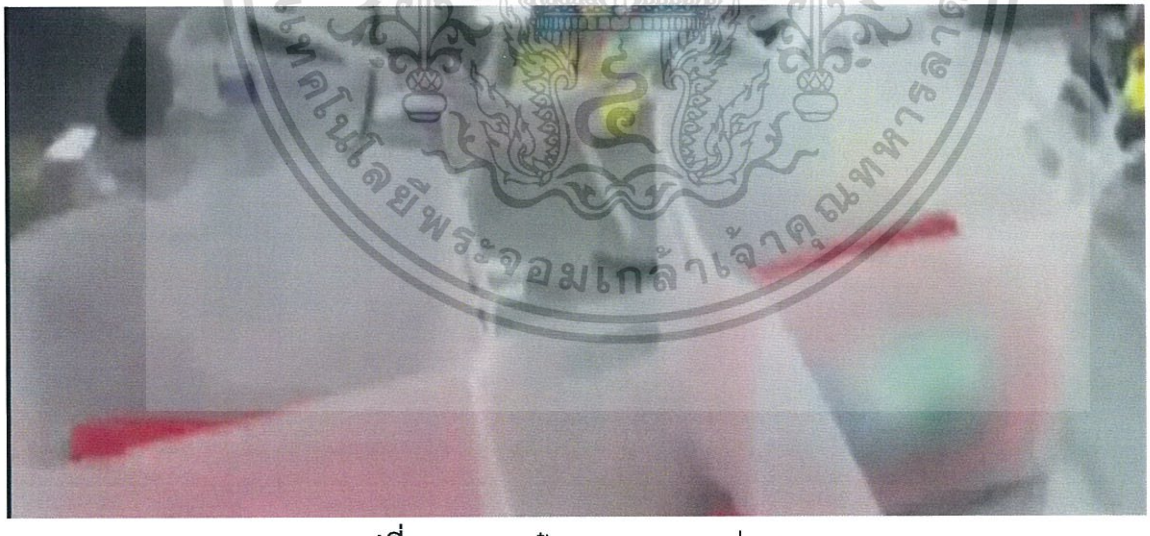


รูปที่4.11 นำกล่องนมมาแยก

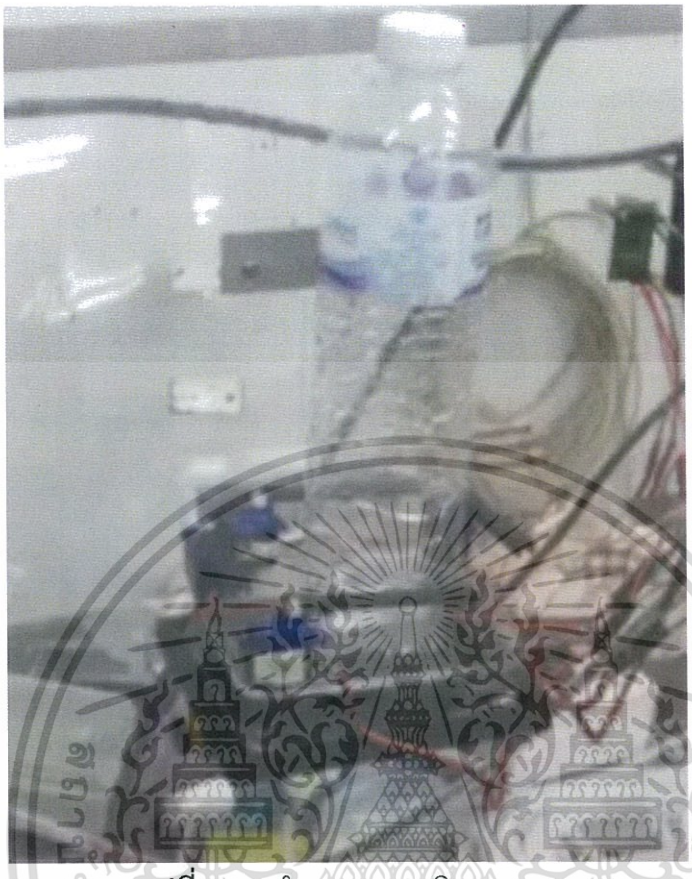


รูปที่4.12 กล่องนมถูกแยกลงกล่อง

รูปที่4.13 นำกระป๋องมาแยก



รูปที่4.14 กระป๋องถูกแยกลงกล่อง



รูปที่4.15 นำขวดพลาสติกมาแยก

รูปที่4.16 ขวดพลาสติกถูกแยกลงกล่อง

รูปที่4.17 ถังขยะให้เงินออกมา

บทที่ 5
สรุปผลการทดลอง

5.1. สรุปผลการทดลอง

ในการทดลองสามารถที่จะแยกประเภทของขยะได้เป็น4ประเภทคือ ขวดน้ำ ขวดแก้ว กล่องนมและกระป๋องโดยจะใช้Load Cellในการแยกประเภทด้วยวิธีการชั่งน้ำหนักแล้วแยกประเภทแล้วเมื่อใส่ขวดพลาสติกครบตามจำนวนที่กำหนดแล้วก็จะให้เงินออกมา

5.2. วิจารณ์ผลการทดลอง

ในการชั่งน้ำหนักเพื่อที่จะแยกประเภทของขวดนั้นอาจจะเกิดความผิดพลาดขึ้นได้เนื่องจากว่าในขยะประเภทที่เราแยกนั้นอาจจะมีหลายขนาดทำให้ตอนที่Load Cellชั่งน้ำหนักอาจจะไปคล้ายกับขยะประเภทอื่นทำให้เครื่องแยกขยะได้คนละประเภทกับที่กำหนดไว้

เอกสารอ้างอิง

[1] https://www.youtube.com/watch?v=f5WPMTv7x7Ms

[2] http://www.myarduino.net/product/60/hx711-weight-sensor-amplifier-module-dual-channel-hx711-for-load-cell

[3] https://circuitdigest.com/microcontroller-projects/arduino-servo-motor-control-code-and-circuit

[4] http://www.myarduino.net/product/919/load-cell-weight-sensor-1-kg

ภาคผนวก

```
#include "HX711.h"
#include <LiquidCrystal_I2C.h> //การเชื่อมต่อ
#include <Servo.h>
#define I2C_ADDR 0x27
#define DOUT A1
#define CLK  A2

HX711 scale(A1, A2);// parameter "gain" is ommited; the default value
128 is used by the library
int sensor = 6; // ประกาศตัวแปรเป็น Sensor
int x = 0; //ค่าจากเซนเซอร์
int i = 0; //จำนวนขวด
unsigned long previousTime = 0;
int setTime = 10; //เวลาจับ
int start = 0; //ตัวแปรการทำงานของฟังก์ชั่นจับเวลาหย่อนขวด
LiquidCrystal_I2C lcd(I2C_ADDR,16,2);
Servo coinServo,servo1, servo2, servo3, servo4;  // create servo object
to control a servo

float weight = 0;
bool detected = false;

void setup() {
  Serial.begin(38400);

  lcd.init();                      // initialize the lcd
  lcd.init();
  // Print a message to the LCD.
  lcd.backlight();
```

```
Serial.println("Setup");
lcd.setCursor(0, 0);
lcd.print("Setup");
delay(500);
servo1.attach(5);  // attaches the servo on pin 5 to the servo object
servo2.attach(9);  // attaches the servo on pin 9 to the servo object
servo3.attach(6);  // attaches the servo on pin 6 to the servo object
servo4.attach(10);  // attaches the servo on pin 10 to the servo object
servoDefault(); // oiginal position
delay(500);
coinServo.attach(11);
coinServo.write(60);
delay(500);
Serial.println("Servo Attached");
lcd.clear();
lcd.setCursor(0, 0);
lcd.println("Servo Attached");
delay(500);
lcd.clear();

lcd.setCursor(3, 0);
lcd.print("57010179");
lcd.setCursor(3, 1);
lcd.print("57010612");
delay(2000);
lcd.setCursor(3, 0);
lcd.print("                ");
lcd.setCursor(3, 1);
lcd.print("                ");
pinMode(sensor,INPUT);
```

```
  lcd.setCursor(5,0); // ไปที่ตัวอักษรที่ 5 แถวที่ 1
  lcd.print("BOTTLE"); // แสดงผลคำว่า Product
  lcd.setCursor(3,1); // ไปที่ตัวอักษรที่ 3 แถวที่ 2
  lcd.print("COUNT = "); // แสดงผลคำว่า Product
  lcd.setCursor(11,1); // ไปที่ตัวอักษรที่ 11 แถวที่ 2
  lcd.print(i); // แสดงค่าของขา Digital ออกหน้าจอ

  scale.set_scale(2280.f);                      // this value is obtained by
calibrating the scale with known weights; see the README for details
  scale.tare();                                 // reset the scale to 0
  Serial.println("Scale Calibrated");
  delay(500);

  Serial.println("Start");
  delay(500);
}

void loop() {

  unsigned long showTime = millis();
  weight = scale.get_units(20);
  Serial.print("Weight:\t");
  Serial.println(weight);
  //x = digitalRead(sensor);

  if(weight>5.00){
    detected = true;
  }else{
    detected = false;
    x = 0;
```

```
  servoDefault();
}

  //Can
  while(detected == true && (weight>=12.00 && weight<15.50)){
     Serial.println("Can");
     //x = 1;
     delay(1000);
     servo2.write(180);          // open position
     delay(1000);
     servo4.write(180);          // open position
     delay(2000);
     weightCheck();              // Check
  }

  //Plastic Bottle
  while(detected == true && (weight>16.00 && weight<17.50)){
     Serial.println("Plastic Bottle");
     x = 1;
     delay(1000);
     servo4.write(0);          // open position
     delay(1000);
     servo2.write(0);          // open position
     delay(2000);
     if (x==1 && i < 2) //ค่า 0 ตรวจจับวัตถุ เริ่มค่า0 แล้วบวกไปเรื่อยๆจนถึง5
ตามจำนวนที่ผ่านเซนเซอร์
        {
          i = i+1;
          lcd.setCursor(11,1); // ไปที่ตัวอักษรที่ 11 แถวที่ 2
          lcd.print(i); // แสดงค่าของขา Digital ออกหน้าจอ
```

```
        Serial.println(i);
        delay(1000);
        //
        start = 1;
        previousTime = showTime;
      }
    weightCheck();              // Check
}

//milk box
while(detected == true && (weight>9.00 && weight<11.50)){
    Serial.println("Milk Box");
    //x = 1;
    delay(1000);
    servo2.write(180);          // open position
    delay(1000);
    servo4.write(180);          // open position
    delay(2000);
    weightCheck();              // Check
}

//glass bottle
while(detected == true && (weight>130.00 && weight<150.50)){
    Serial.println("Glass Bottle");
    //x = 1;
    //delay(1000);
    servo3.write(0);            // open position
    delay(1000);
    //servo1.write(0);          // open position
    //delay(3000);
```

```
    weightCheck();                  // Check
  }

    if(start == 1) //เริ่มจับเวลาหย่อนขวด
    {

      Serial.println((showTime - previousTime) / 1000); //เริ่มนับตั้งแต่
เปิดเครื่องนับเวลาไปลบกับที่ตรวจจับขวดได้ไปหาร1000 เพื่อให้ได้หน่วยมิลลิวินาที

      if((showTime - previousTime) / 1000 >= setTime) //ถ้าเวลาที่จับได้
มากกว่าหรือเท่ากับเวลาที่ตั้งไว้...
      {
        start = 0; //ถ้าไม่ถึง5ในเวลา2วินาทีจะแสดงค่า ตัวเลขนั้นก้บเครื่องหมาย
? แล้วเครื่องจะรีเซ็ตให้เป็นจำนวน 0 ขวด
        lcd.setCursor(12,1); // ไปที่ตัวอักษรที่ 12 แถวที่ 2
        lcd.print('?');
        delay(2000);
        i = 0;
        lcd.setCursor(11,1); // ไปที่ตัวอักษรที่ 12 แถวที่ 2
        lcd.print(" ");
        lcd.setCursor(11,1); // ไปที่ตัวอักษรที่ 12 แถวที่ 2
        lcd.print(i);
        servoDefault();
      }
    }

  if (i == 2) //ถ้าได้เลข2 แล้วจะแสดงเลข5แล้วขึ้น เครื่องหมาย $ แล้วทำให้เซอร์
โวหมุนทำงานด้วยคำสั่ง (0) ครึ่งรอบวงกลม แล้วเซอร์โวจะหยุดทำงานด้วยคำสั่ง
(90)
  {
```

```
      start = 0;
      lcd.setCursor(11,1); // ไปที่ตัวอักษรที่ 11 แถวที่ 2
      lcd.print(i);
      delay(500);
      lcd.setCursor(12,1); // ไปที่ตัวอักษรที่ 12 แถวที่ 2
      lcd.print('$');
      delay(1000);
      coinServo.write(90); //servo ทำงาน
      delay(650);
      coinServo.write(45); //servo หยุด
      i = 0;
      lcd.setCursor(11,1); // ไปที่ตัวอักษรที่ 12 แถวที่ 2
      lcd.print("  ");
      lcd.setCursor(11,1); // ไปที่ตัวอักษรที่ 12 แถวที่ 2
      lcd.print(i);
      servoDefault();
      //return i;
    }
  //delay(500);
  //scale.power_up();
}

void servoDefault(){
      servo1.write(90);          // oiginal position
      servo2.write(90);          // oiginal position
      servo3.write(90);          // oiginal position
      servo4.write(90);          // oiginal position
      delay(1000);
      x = 0;
}
```

```
void weightCheck(){
    weight = scale.get_units();
    if(weight<5.00 && weight<0.00){
        detected = false;
        servoDefault();
        x = 0;
    }
}
```

24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales

DESCRIPTION

Based on Avia Semiconductor's patented technology, HX711 is a precision 24-bit analog-to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor.

The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of ±20mV or ±40mV respectively, when a 5V supply is connected to AVDD analog power supply pin. Channel B has a fixed gain of 32. On-chip power supply regulator eliminates the need for an external supply regulator to provide analog power for the ADC and the sensor. Clock input is flexible. It can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component. On-chip power-on-reset circuitry simplifies digital interface initialization.

There is no programming needed for the internal registers. All controls to the HX711 are through the pins.

FEATURES

- **Two selectable differential input channels**
- **On-chip active low noise PGA with selectable gain of 32, 64 and 128**
- **On-chip power supply regulator for load-cell and ADC analog power supply**
- **On-chip oscillator requiring no external component with optional external crystal**
- **On-chip power-on-reset**
- **Simple digital control and serial interface: pin-driven controls, no programming needed**
- **Selectable 10SPS or 80SPS output data rate**
- **Simultaneous 50 and 60Hz supply rejection**
- **Current consumption including on-chip analog power supply regulator:**
  normal operation < 1.5mA, power down < 1uA
- **Operation supply voltage range: 2.6 ~ 5.5V**
- **Operation temperature range: -40 ~ +85℃**
- **16 pin SOP-16 package**

APPLICATIONS

- **Weigh Scales**
- **Industrial Process Control**

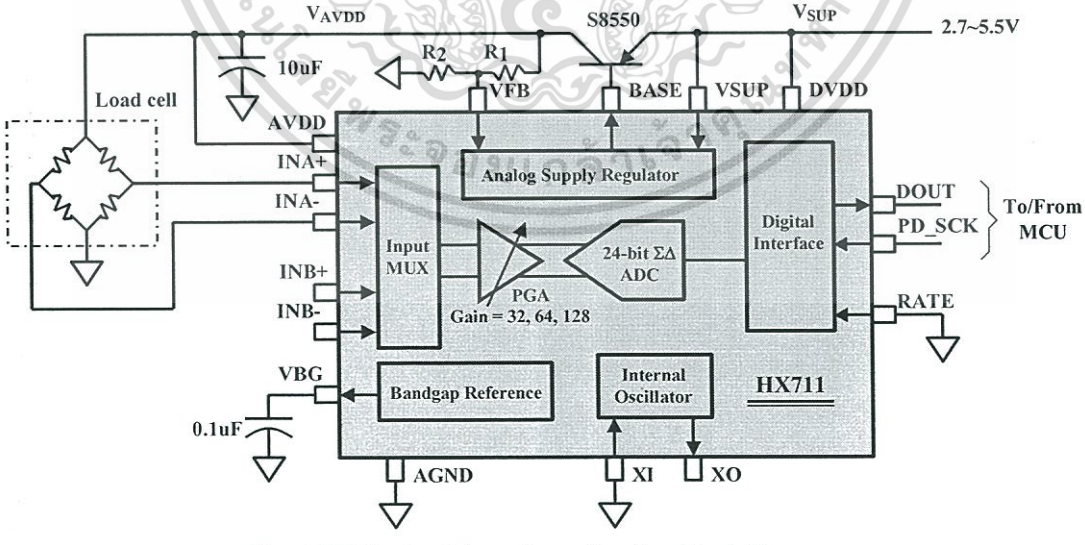
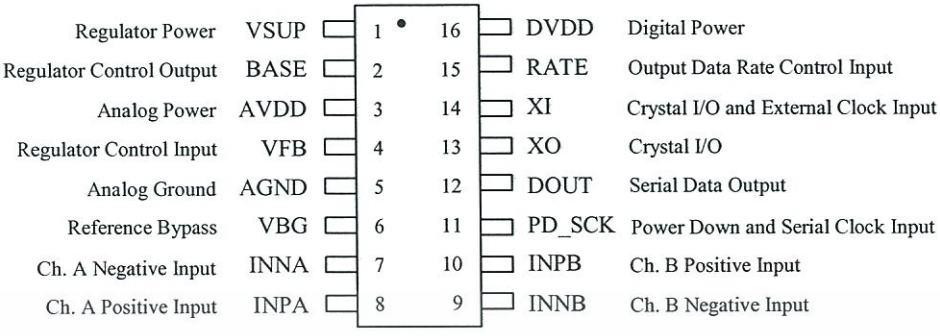


Fig. 1 Typical weigh scale application block diagram

Pin Description



SOP-16L Package

Pin #	Name	Function	Description
1	VSUP	Power	Regulator supply: 2.7 ~ 5.5V
2	BASE	Analog Output	Regulator control output（NC when not used）
3	AVDD	Power	Analog supply: 2.6 ~ 5.5V
4	VFB	Analog Input	Regulator control input（connect to AGND when not used）
5	AGND	Ground	Analog Ground
6	VBG	Analog Output	Reference bypass output
7	INA-	Analog Input	Channel A negative input
8	INA+	Analog Input	Channel A positive input
9	INB-	Analog Input	Channel B negative input
10	INB+	Analog Input	Channel B positive input
11	PD_SCK	Digital Input	Power down control (high active) and serial clock input
12	DOUT	Digital Output	Serial data output
13	XO	Digital I/O	Crystal I/O (NC when not used）
14	XI	Digital Input	Crystal I/O or external clock input, 0: use on-chip oscillator
15	RATE	Digital Input	Output data rate control, 0: 10Hz; 1: 80Hz
16	DVDD	Power	Digital supply: 2.6 ~ 5.5V

Table 1 Pin Description

KEY ELECTRICAL CHARACTERISTICS

Parameter	Notes	MIN	TYP	MAX	UNIT
Full scale differential input range	V(inp)-V(inn)		±0.5(AVDD/GAIN)		V
Common mode input		AGND+1.2		AVDD-1.3	V
Output data rate	Internal Oscillator, RATE = 0		10		Hz
	Internal Oscillator, RATE = DVDD		80		
	Crystal or external clock, RATE = 0		f_{clk}/1,105,920		
	Crystal or external clock, RATE = DVDD		f_{clk}/138,240		
Output data coding	2's complement	800000		7FFFFF	HEX
Output settling time (1)	RATE = 0		400		ms
	RATE = DVDD		50		
Input offset drift	Gain = 128		0.2		mV
	Gain = 64		0.4		
Input noise	Gain = 128, RATE = 0		50		nV(rms)
	Gain = 128, RATE = DVDD		90		
Temperature drift	Input offset (Gain = 128)		±6		nV/℃
	Gain (Gain = 128)		±5		ppm/℃
Input common mode rejection	Gain = 128, RATE = 0		100		dB
Power supply rejection	Gain = 128, RATE = 0		100		dB
Reference bypass (V_{BG})			1.25		V
Crystal or external clock frequency		1	11.0592	20	MHz
Power supply voltage	DVDD	2.6		5.5	V
	AVDD, VSUP	2.6		5.5	
Analog supply current (including regulator)	Normal		1400		μA
	Power down		0.3		
Digital supply current	Normal		100		μA
	Power down		0.2		

(1) Settling time refers to the time from power up, reset, input channel change and gain change to valid stable output data.

Table 2 Key Electrical Characteristics

Analog Inputs

Channel A differential input is designed to interface directly with a bridge sensor's differential output. It can be programmed with a gain of 128 or 64. The large gains are needed to accommodate the small output signal from the sensor. When 5V supply is used at the AVDD pin, these gains correspond to a full-scale differential input voltage of ±20mV or ±40mV respectively.

Channel B differential input has a fixed gain of 32. The full-scale input voltage range is ±80mV, when 5V supply is used at the AVDD pin.

Power Supply Options

Digital power supply (DVDD) should be the same power supply as the MCU power supply.

When using internal analog supply regulator, the dropout voltage of the regulator depends on the external transistor used. The output voltage is equal to $V_{AVDD}=V_{BG}*(R1+R2)/ R1$ (Fig. 1). This voltage should be designed with a minimum of 100mV below VSUP voltage.

If the on-chip analog supply regulator is not used, the VSUP pin should be connected to either AVDD or DVDD, depending on which voltage is higher. Pin VFB should be connected to Ground and pin BASE becomes NC. The external 0.1uF bypass capacitor shown on Fig. 1 at the VBG output pin is then not needed.

Clock Source Options

By connecting pin XI to Ground, the on-chip oscillator is activated. The nominal output data rate when using the internal oscillator is 10 (RATE=0) or 80SPS (RATE=1).

If accurate output data rate is needed, crystal or external reference clock can be used. A crystal can be directly connected across XI and XO pins. An external clock can be connected to XI pin, through a 20pF ac coupled capacitor. This external clock is not required to be a square wave. It can come directly from the crystal output pin of the MCU chip, with amplitude as low as 150 mV.

When using a crystal or an external clock, the internal oscillator is automatically powered down.

Output Data Rate and Format

When using the on-chip oscillator, output data rate is typically 10 (RATE=0) or 80SPS (RATE=1).

When using external clock or crystal, output data rate is directly proportional to the clock or crystal frequency. Using 11.0592MHz clock or crystal results in an accurate 10 (RTE=0) or 80SPS (RATE=1) output data rate.

The output 24 bits of data is in 2's complement format. When input differential signal goes out of the 24 bit range, the output data will be saturated at 800000h (MIN) or 7FFFFFh (MAX), until the input signal comes back to the input range.

Serial Interface

Pin PD_SCK and DOUT are used for data retrieval, input selection, gain selection and power down controls.

When output data is not ready for retrieval, digital output pin DOUT is high. Serial clock input PD_SCK should be low. When DOUT goes to low, it indicates data is ready for retrieval. By applying 25~27 positive clock pulses at the PD_SCK pin, data is shifted out from the DOUT output pin. Each PD_SCK pulse shifts out one bit, starting with the MSB bit first, until all 24 bits are shifted out. The 25th pulse at PD_SCK input will pull DOUT pin back to high (Fig.2).

Input and gain selection is controlled by the number of the input PD_SCK pulses (Table 3). PD_SCK clock pulses should not be less than 25 or more than 27 within one conversion period, to avoid causing serial communication error.

PD_SCK Pulses	Input channel	Gain
25	A	128
26	B	32
27	A	64

Table 3 Input Channel and Gain Selection

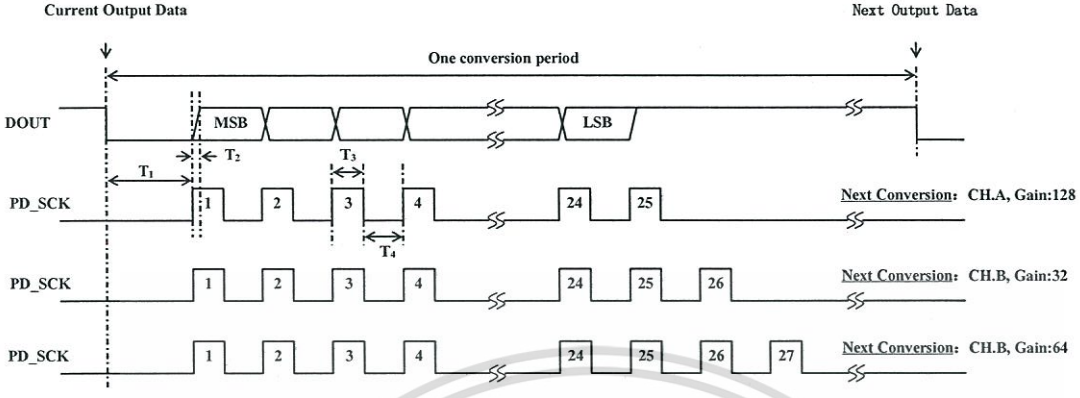


Fig.2 Data output, input and gain selection timing and control

Symbol	Note	MIN	TYP	MAX	Unit
T_1	DOUT falling edge to PD_SCK rising edge	0.1			μs
T_2	PD_SCK rising edge to DOUT data ready			0.1	μs
T_3	PD_SCK high time	0.2	1	50	μs
T_4	PD_SCK low time	0.2	1		μs

Reset and Power-Down

When chip is powered up, on-chip power on rest circuitry will reset the chip.

Pin PD_SCK input is used to power down the HX711. When PD_SCK Input is low, chip is in normal working mode.

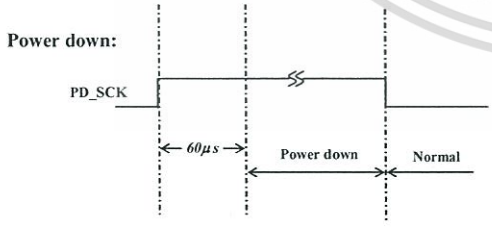


Fig.3 Power down control

When PD_SCK pin changes from low to high and stays at high for longer than 60μs, HX711 enters power down mode (Fig.3). When internal regulator is used for HX711 and the external transducer, both HX711 and the transducer will be powered down. When PD_SCK returns to low, chip will reset and enter normal operation mode.

After a reset or power-down event, input selection is default to Channel A with a gain of 128.

Application Example

Fig.1 is a typical weigh scale application using HX711. It uses on-chip oscillator (XI=0), 10Hz output data rate (RATE=0). A Single power supply （2.7～5.5V） comes directly from MCU power supply. Channel B can be used for battery level detection. The related circuitry is not shown on Fig. 1.

Reference PCB Board (Single Layer)

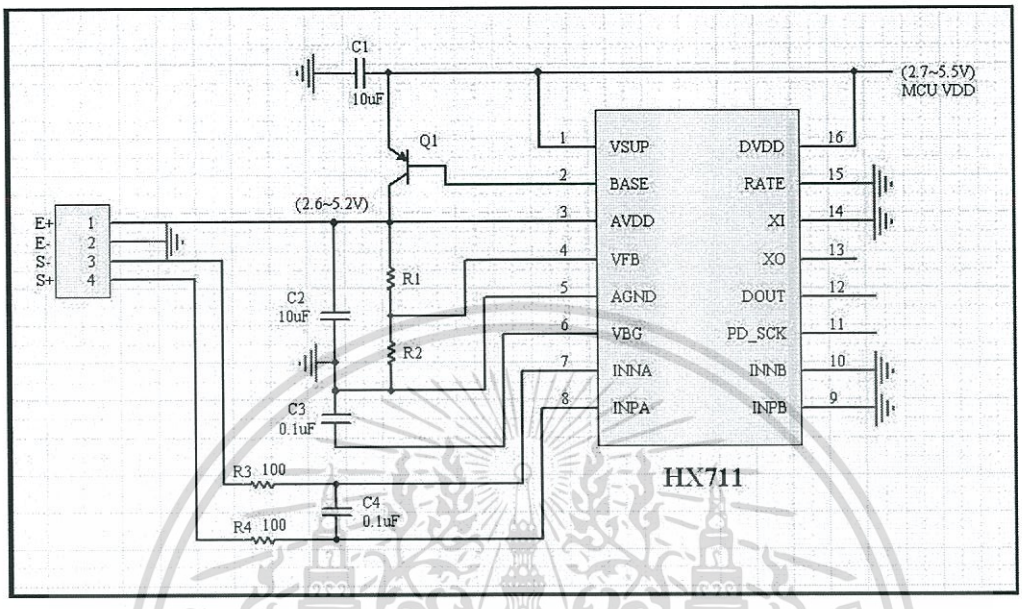


Fig.4 Reference PCB board schematic

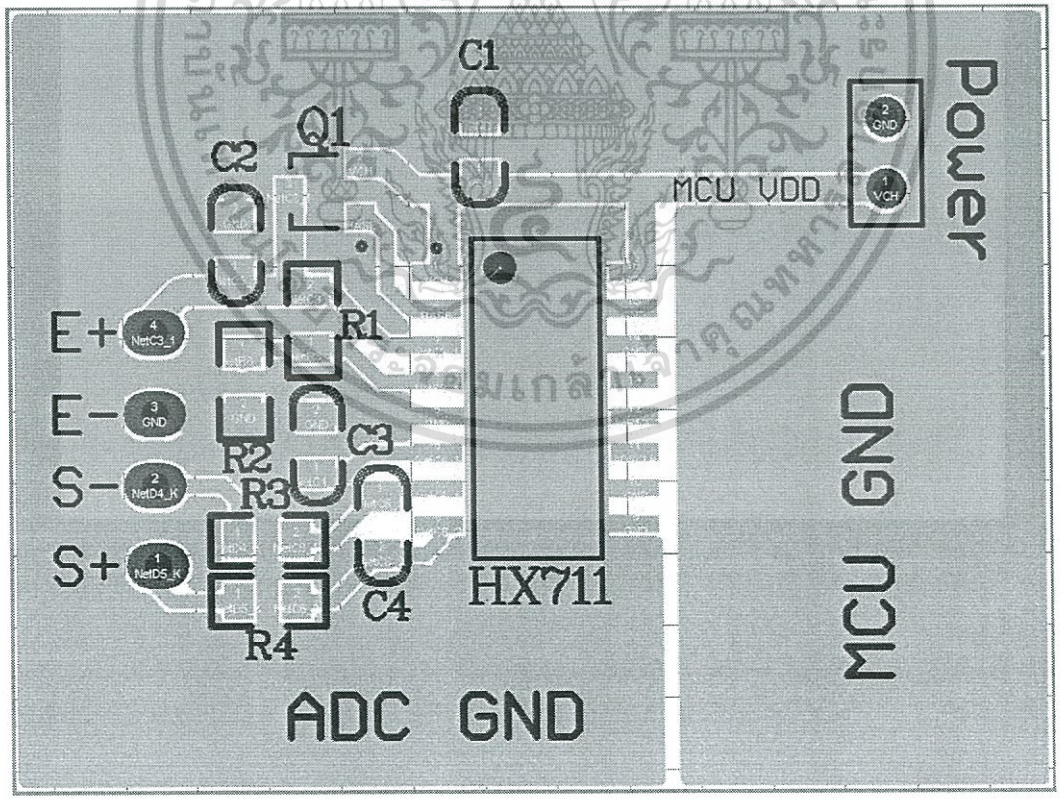


Fig.5 Reference PCB board layout

Reference Driver (Assembly)

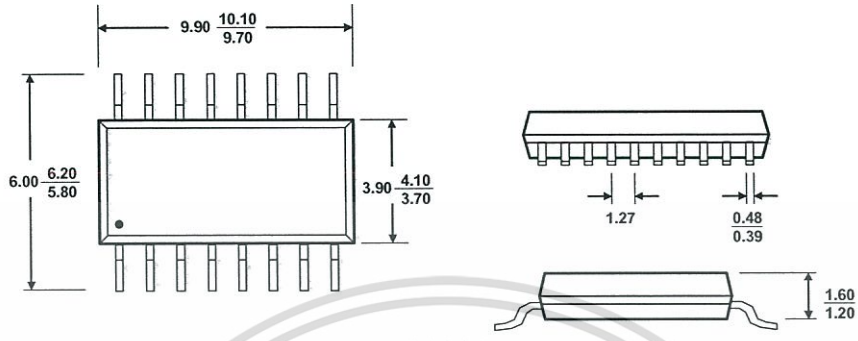
```
/*----------------------------------------------------------------------
Call from ASM:      LCALL   ReaAD
Call from C:    extern unsigned long ReadAD(void);
                  .
                  .
                unsigned long data;
                data=ReadAD();
                 .
                 .
----------------------------------------------------------------------*/
PUBLIC          ReadAD
HX711ROM        segment  code
rseg            HX711ROM

sbit            ADDO = P1.5;
sbit            ADSK = P0.0;
/*-------------------------------------------------
 OUT:   R4, R5, R6, R7   R7=>LSB
-------------------------------------------------*/
ReadAD:
    CLR     ADSK                //AD Enable (PD_SCK set low)
    SETB    ADDO                //Enable 51CPU I/O
    JB      ADDO,$              //AD conversion completed?
    MOV     R4,#24
ShiftOut:
    SETB    ADSK                //PD_SCK set high (positive pulse)
    NOP
    CLR     ADSK                //PD_SCK set low
    MOV     C,ADDO              //read on bit
    XCH     A,R7                //move data
    RLC     A
    XCH     A,R7
    XCH     A,R6
    RLC     A
    XCH     A,R6
    XCH     A,R5
    RLC     A
    XCH     A,R5
    DJNZ    R4,ShiftOut         //moved 24BIT?
    SETB    ADSK
    NOP
    CLR     ADSK
    RET
    END
```

Reference Driver（C）

```
//-------------------------------------------------------------------
sbit   ADDO = P1^5;
sbit   ADSK = P0^0;
unsigned long ReadCount(void){
  unsigned long Count;
  unsigned char i;
  ADDO=1;
  ADSK=0;
  Count=0;
  while(ADDO);
  for (i=0;i<24;i++){
    ADSK=1;
    Count=Count<<1;
    ADSK=0;
    if(ADDO) Count++;
  }
  ADSK=1;
  Count=Count^0x800000;
  ADSK=0;
  return(Count);
}
```

Package Dimensions



Typ MAX/MIN Unit: mm

SOP-16L Package