เครื่องคัดแยกขวดอัตโนมัติ

Automatic Bottle Separator

นายณัฐวัฒน์ อ้วนตรงต้น

นายภาณุพงศ์ พลีน้อย

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม

คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

พ.ศ. 2563

Automatic Bottle Separator

Nattawat Ountongton

Panupong Pleenoi

This Project Report Submitted in Partial Fulfillment of

The Requirements for the Degree of Bachelor of Industrial Technology Education

Program (Electrical Engineering)

Department of Electronics and Telecommunication Engineering

Faculty of Industrial Education

Rajamangala University of Technology phra Nakhon

2020

ชื่อโครงการ เครื่องคัดแยกขวดอัตโนมัติ

ชื่อนักศึกษา นายณัฐวัฒน์ อ้วนตรงต้น รหัส 0361605002-6

ชื่อนักศึกษา นายภาณุพงศ์ พลีน้อย รหัส 0361605005-9

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม

อาจารย์ที่ปรึกษา อาจารย์สุปัญญา สิงห์กรณ์

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร อนุมัติให้รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

.....

(อาจารย์ภาวนา ชูศิริ)

หัวหน้าสาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม

คณะกรรมการสอบโครงการ

.....ประธาน
(อาจารย์สุปัญญา สิงห์กรณ์)

.....กรรมการ
(อาจารย์ภาวนา ชูศิริ)

.....กรรมการ
(อาจารย์อนุชา ไชยชาญ)

.....กรรมการ
(ผศ.ดร.รุ่งอรุณ พรเจริญ)

.....กรรมการ
(อาจารย์วารินี วีระสินธุ์)

.....กรรมการ
(อาจารย์นิคม ดิษฐคลี)

ชื่อโครงการ เครื่องคัดแยกขวดอัตโนมัติ
ชื่อนักศึกษา นายณัฐวัฒน์ อ้วนตรงต้น รหัส 0361605002-6
ชื่อนักศึกษา นายภาณุพงศ์ พลีน้อย รหัส 0361605005-9
สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม
อาจารย์ที่ปรึกษา อาจารย์สุปัญญา สิงห์กรณ์

บทคัดย่อ

โครงการนี้จัดทำขึ้นเพื่อคัดแยกขวดอัตโนมัติ โดยได้กำหนดแยกขวดจำนวน 3 ประเภท คือ ขวดพลาสติก ขวดแก้ว และกระป๋อง ในการช่วยอำนวยความสะดวกในการคัดแยกขวดแล้วนำขวดที่คัดแยกกลับมารีไซเคิลใหม่ได้ตามที่ต้องการ โดยผู้จัดทำได้ศึกษาการใช้งานโปรแกรม Arduino มาประยุกต์ควบคุมการทำงานในส่วนของการคัดแยกขวดตามประเภทที่กำหนดนั้น ได้ศึกษาการทำงานระบบไมโครคอนโทรลเลอร์ ระบบการควบคุม และการใช้งานโปรแกรม Arduino ระบบการทำงานแบ่งออกเป็น 2 ส่วน คือ ส่วนเซนเซอร์ตรวจจับโลหะ และเซนเซอร์น้ำหนักแล้วส่งข้อมูลไปยัง Arduino ภาครับของเซนเซอร์ และเซนเซอร์จะตรวจจับวัตถุตลอดเวลาที่ช่องทิ้งขวด

ผลการทดลอง พบว่าการทำงานของระบบเครื่องคัดแยกขวดอัตโนมัติ เมื่อทิ้งขวดผ่านช่องเซ็นเซอร์ ประสิทธิภาพในการทำงานของโปรแกรม Arduino ที่ได้เขียนกำหนดแยกประเภทขวดไว้นั้น มีค่าเปอร์เซ็นต์การทำงานตามที่กำหนดร้อยละ 90 และมีเปอร์เซ็นต์ค่าความผิดพลาดร้อยละ 10

(รายงานมีจำนวนทั้งสิ้น 73 หน้า)

Title Automatic bottle separator
Students Mr.Nattawat Ountongton No. 0361605002-6
Students Mr.Panupong Pleenoi No. 0361605005-9
Major Electronics and Telecommunication Engineering
Advisor Mr.Supanya Singkorn

Abstract

This project was created to sort bottles automatically. By specifying 3 types of bottles which are plastic bottles, glass bottles and cans To help facilitate the separation of the bottles and then recycle the extracted bottles as needed. The organizer has studied the use of the Arduino program to control the operation of the sorting bottles according to the specified category. Studied the microcontroller system operation Control system And how to use the Arduino program The operating system is divided into 2 parts, namely the metal detection sensor section And weight sensor and then send data to the sensor receiver Arduino And the sensor will detect the object all the time in the empty bottle compartment

The results showed that the operation of the automatic bottle sorter system When throwing the bottle through the sensor compartment. The performance of the Arduino program that has been written to define the classification of the bottles. Have the percentage of work as specified by 90% and the error percentage of 10%

(Total 73 Pages)

กิตติกรรมประกาศ

รายงานนี้สำเร็จลุล่วงไปได้ด้วยการสนับสนุนจากอาจารย์สุปัญญา สิงห์กรณ์ อาจารย์ที่ปรึกษา รวมทั้งอาจารย์ประจำสาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคมทุกท่านที่ได้ให้คำแนะนำให้ความช่วยเหลือจากการทำโครงการ และให้ความอนุเคราะห์ด้านวัสดุอุปกรณ์และสถานที่ปฏิบัติงานระหว่างการทดลอง การแก้ไขปัญหาอุปสรรคต่างๆ และได้รับทุนอุดหนุนงบประมาณจากโครงการส่งเสริมสิ่งประดิษฐ์และนวัตกรรมเพื่อคนรุ่นใหม่ ประจำปีงบประมาณ พ.ศ. 2563 มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร จนโครงการนี้ประสบผลสำเร็จ คณะผู้จัดทำขอขอบพระคุณเป็นอย่างสูง

สุดท้ายนี้คุณค่าและประโยชน์ใดๆ อันพึงมีจากโครงการนี้ คณะผู้จัดทำขอมอบให้เพื่อทดแทนคุณ บิดา มารดา ครูอาจารย์ ครอบครัว ญาติ เพื่อนๆ ทุกคนที่คอยช่วยเหลือให้การสนับสนุนด้านต่างๆ และเป็นกำลังใจมาโดยตลอด

ณัฐวัฒน์ อ้วนตรงต้น
ภาณุพงศ์ พลีน้อย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญตาราง	ช
สารบัญภาพ	ซ
บทที่	
1. บทนำ	1
1.1 ปัญหาและความเป็นมาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 แผนการดำเนินงาน	2
1.6 ประโยชน์ที่คาดว่าจะได้รับ	3
2. เอกสารและทฤษฎีที่เกี่ยวข้อง	4
2.1 โปรแกรมภาษาอาดุยโน่	4
2.2 Arduino IDE	6
2.3 บอร์ดอาดุยโน่	10
2.4 servo	12
2.5 sensor	13
2.6 load cell	14
2.7 จอ LCD	26
3. ขั้นตอนและวิธีการดำเนินงาน	28
3.1 ขั้นตอนการดำเนินงาน	28
3.2 การทำงานของเครื่องคัดแยกขวดอัตโนมัติโดยใช้โปรแกรม Arduino	30
3.3 ขั้นตอนการประกอบชิ้นงาน	31
4. ผลการดำเนินงาน	38
4.1 ขั้นตอนการทดสอบโครงการ	38
4.2 ภาพรวมแสดงผลการทดสอบประสิทธิภาพของระบบ	40
4.3 สรุปผลการทดสอบโครงการ	44

สารบัญ (ต่อ)

	หน้า
5. สรุปผลและข้อเสนอแนะ	45
5.1 สรุปผลการดำเนินงาน	45
5.2 ปัญหาและอุปสรรค	45
5.3 ข้อเสนอแนะ	45
บรรณานุกรม	46
ภาคผนวก	47
ภาคผนวก ก โปรแกรมการทำงานเครื่องคัดแยกขวดอัตโนมัติ	48
ภาคผนวก ข โครงสร้าง และการต่อวงจร	53
ภาคผนวก ค คู่มือการใช้งานเครื่องคัดแยกขวดอัตโนมัติ	57
ภาคผนวก ง Data Sheet	60
ประวัติผู้เขียน	71

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินการ	2
4.1 แสดงค่าความถูกต้องของการทดสอบระบบเซนเซอร์ของระบบ รอบที่ 1	38
4.2 แสดงค่าความถูกต้องของการทดสอบระบบเซนเซอร์ของระบบ รอบที่ 2	38
4.3 แสดงค่าความถูกต้องของการทดสอบระบบเซนเซอร์ของระบบ รอบที่ 3	39
4.4 แสดงค่าความถูกต้องของการทดสอบระบบเซนเซอร์ของระบบ รอบที่ 4	39
4.5 ผลรวมแสดงผลการทดสอบประสิทธิภาพในการคัดแยกขวดอัตโนมัติ	40

สารบัญภาพ

ภาพที่		หน้า
2.1	Header โปรแกรม	5
2.2	Setup โปรแกรม	5
2.3	Loop โปรแกรม	6
2.4	เปิดโปรแกรม Arduino IDE	6
2.5	สร้าง Sketch ใหม่	7
2.6	เขียนโปรแกรม	7
2.7	การบันทึกโปรแกรม	8
2.8	การอัพโหลดโปรแกรมลงบอร์ด	9
2.9	การอัพโหลดบอร์ดเสร็จสิ้น	9
2.10	บอร์ด Arduino Uno R3	10
2.11	บอร์ด Arduino Uno R2	10
2.12	บอร์ด Arduino Uno SMD	11
2.13	MG995 Servo Motor 0-180 องศา	12
2.14	เซนเซอร์ตรวจจับโลหะ	13
2.15	Load cell	15
2.16	การวัดโดยเกจตัวต้านทานทั้งสี่	16
2.17	Load cell 5 kg	16
2.18	Load cell สำหรับตรวจวัดน้ำหนัก	17
2.19	Wheatstone Bridge	17
2.20	วิธีเชื่อมต่อ Load Cell และ HX711 เข้ากับ Arduino	18
2.21	ขาอุปกรณ์ HX711	18
2.22	ตุ้มน้ำหนักมาตรฐานเพื่อทำการ Calibrate	18
2.23	ขวดน้ำเพื่อทำการ Calibrate	19
2.24	ค่าน้ำหนักขวดน้ำ	19
2.25	โปรแกรม Arduino Auto Cal ลงไปยัง board Arduino	20
2.26	เปิดหน้าต่าง Serial Monitor ขึ้นมา	20
2.27	ตั้งค่า Baud rate เป็น 115200	20
2.28	นำสิ่งของทุกอย่างออกจากแผ่นชั่งน้ำหนัก	21
2.29	ส่งตัว ‘a’ ไปเพื่อหาค่า zero	21
2.30	ค่า Zero Factor ออกมาให้เราจดบันทึกค่านี้เอาไว้ใช้งาน	22
2.31	นำวัตถุที่ทราบค่าน้ำหนักที่แน่นอนมาวางลงบนแผ่นชั่งน้ำหนัก	22
2.32	ส่งตัว ‘b’ ไปเพื่อหาค่า calibration factor	23

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
2.33	ค่า calibration factor ได้สำเร็จให้เราจดบันทึกค่านี้เอาไว้ใช้งาน	23
2.34	ส่งตัว ‘c’ ไปเพื่อทดลองอ่านค่าน้ำหนัก	24
2.35	ผลการทดลองอ่านค่า	24
2.36	ให้เรานำเอาค่ามาใส่ในตัวแปรแทนของเดิม	25
2.37	ทดลอง Run Program โดยชั่งขวดน้ำ (1.56 Kg)	25
2.38	ผลทดลอง Run Program โดยชั่งขวดน้ำ (1.56 Kg)	26
2.39	การเชื่อมต่อแบบอนุกรมจะใช้งานโมดูล	27
3.1	แผนการดำเนินงาน	28
3.2	บล็อกไดอะแกรมแสดงการทำงานของเครื่องคัดแยกขวดอัตโนมัติ	30
3.3	วงจรเครื่องคัดแยกขวดอัตโนมัติ	30
3.4	วงจร Load cell ต่อกับ Arduino	31
3.5	อลูมิเนียมฉาก	31
3.6	แผ่นอะคริลิกรู	32
3.7	อลูมิเนียมฉาก 2 ขนาด	32
3.8	ประกอบอลูมิเนียมเข้าฉากและเจาะรู	33
3.9	ยิงรีเวทยึดฉาก 4 มุม	33
3.10	ขึ้นโครง	34
3.11	ตัดแผ่นอะคริลิกยึดกับโครง	34
3.12	ติดตาข่ายพลาสติก	35
3.13	ติดแผ่นอะคริลิกเป็นประตูเข้ากับโครง	35
3.14	บัดกรีวงจรและสายไฟที่ใช้กับส่วนต่างๆภายใน	36
3.15	เก็บสายให้เรียบร้อยแล้วใส่กล่อง	36
3.16	จอแสดงผล LCD	37
4.1	ใส่ขวดพลาสติกเพื่อทดสอบเซนเซอร์น้ำหนัก	40
4.2	ใส่กระป๋องเพื่อทดสอบเซนเซอร์ตรวจจับโลหะ	41
4.3	ใส่ขวดแก้วเพื่อทดสอบเซนเซอร์น้ำหนัก	41
4.4	หน้าจอแสดงผลจำนวนขวดแต่ละประเภท	41
4.5	ใส่ขวดพลาสติกในช่องเซนเซอร์	42
4.6	เซนเซอร์ไม่ทำงานตามเงื่อนไข	42
4.7	ใส่ขวดพลาสติกในช่องเซนเซอร์	43
4.8	เซนเซอร์ทำงานตามเงื่อนไข	43

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ข.1 โครงสร้างเครื่องคัดแยกขวดอัตโนมัติ	54
ข.2 โครงสร้างเครื่องคัดแยกขวดอัตโนมัติ (ต่อ)	54
ข.3 โครงสร้างเครื่องคัดแยกขวดอัตโนมัติ (ต่อ)	55
ข.4 โครงสร้างเครื่องคัดแยกขวดอัตโนมัติ (ต่อ)	55
ข.5 วงจรเครื่องคัดแยกขวดอัตโนมัติ	56
ข.6 วงจรเครื่องคัดแยกขวดอัตโนมัติ(ต่อ)	56
ค.1 ช่องปลั๊กควบคุมอุปกรณ์ต่างๆ	58
ค.2 ช่องปลั๊กควบคุมอุปกรณ์ต่างๆ (ต่อ)	58
ค.3 นำขวดใส่ช่องคัดแยกขวดอัตโนมัติ	59
ค.4 หน้าจอแสดงผลจำนวนขวดแต่ละประเภท	59

บทที่ 1
บทนำ

1.1 ปัญหาและความเป็นมาของโครงการ

ในปัจจุบันความนิยมในการใช้ขวดพลาสติกขวดแก้วและขวดอลูมิเนียมในการบรรจุเครื่องดื่มมีเป็นจำนวนมาก เนื่องจากสะดวกในการใช้งาน หาง่ายและมีราคาถูก ผลิตภัณฑ์เหล่านี้จะทำให้เกิดขยะมูลฝอยพลาสติก ซึ่งเป็นสาเหตุประการหนึ่งที่ก่อให้เกิดปัญหาสิ่งแวดล้อมและมีผลต่อสุขภาพ อนามัยขยะมีปริมาณเพิ่มขึ้นทุกปี นับเป็นปัญหาที่สำคัญของชุมชนซึ่งต้องได้รับการแก้ไข การกำจัดขยะโดยการเอาไปเผา ก็เป็นสาเหตุหนึ่งที่ทำให้เกิดมลพิษ เกิดแก๊ส ภาวะเรือนกระจก หรือภาวะโลกร้อน การกำจัดอีกทางหนึ่ง คือ การนำขยะไปฝัง แต่ต้องเป็นขยะที่สามารถย่อยสลายได้ และต้องใช้เวลานาน

จากการสำรวจขยะในบริเวณมหาวิทยาลัย พบว่าขยะส่วนใหญ่เป็นขวดพลาสติกขวดแก้วและขวดอลูมิเนียม เพื่อช่วยลดปัญหาขยะ ปัญหาภาวะโลกร้อน และยังเป็นการนำเอาของที่เหลือใช้มาทำให้มีค่า ให้เกิดประโยชน์ และทำการรีไซเคิลนำ กลับมาใช้ใหม่ได้อีกด้วย เครื่องแยกขวดอัตโนมัติจะสามารถรับขวดรีไซเคิลได้ตามประเภทของขวด ได้แก่ ขวดพลาสติก ขวดแก้ว กระป๋อง และเป็นการปลูกจิตสำนึก และเห็นถึงความสำคัญการเข้ามามีส่วนร่วมในการแก้ไขปัญหาขยะ จึงเป็นรูปแบบหนึ่งของการดำเนินงานเพื่อส่งเสริมให้เกิดความรู้ความเข้าใจในการคัดแยกขยะให้ถูกวิธีและนำมาใช้ในมหาวิทยาลัยมีระบบการจัดการขยะอย่างมีประสิทธิภาพ และมี สภาพแวดล้อมที่ดีคณะผู้จัดทำได้เล็งเห็นถึงความสำคัญและที่มาของปัญหา จากข้อมูลในข้างต้น จึงมีแนวคิด ความคิดที่จะสร้างเครื่องแยกขวดอัตโนมัติขึ้นเพื่อคัดแยกขวดและนำกลับมารีไซเคิลใหม่ และจะเป็นประโยชน์ในการแยกประเภท ของขวด ลดเวลาในการคัดแยกขวด และนำกลับมารีไซเคิลใหม่ได้ตามต้องการต่อไป

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อคัดแยกขวดอัตโนมัติ

1.2.2 เพื่อคัดแยกขวดให้ตามประเภทที่กำหนด

1.2.3 เพื่ออำนวยความสะดวกในการคัดแยกขวด

1.2.4 เพื่อนำขวดที่คัดแยกกลับมารีไซเคิล

1.3 ขอบเขตของโครงการ

1.3.1 สร้างเครื่องคัดแยกขวดอัตโนมัติ

1.3.2 เครื่องคัดแยกได้ 3 ประเภท คือ 1. ขวดพลาสติก, 2. ขวดแก้ว, 3. กระป๋อง

1.3.3 ใช้อาดุยโน่ควบคุมจอแสดงผล LCD

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาทฤษฎีและการวิจัยที่เกี่ยวข้อง

1.4.2 เก็บรวบรวมข้อมูลเกี่ยวกับเซ็นเซอร์ในการตรวจจับวัตถุเพื่อเป็นแนวคิดที่สร้างเครื่องคัดแยกขวดอัตโนมัติ

1.4.3 ศึกษาข้อมูลการใช้อาดุยโน่ควบคุมการทำงานของเซ็นเซอร์

1.4.4 ออกแบบระบบและเขียนโปรแกรมสั่งงานอาดุยโน่ควบคุมการทำงานของอุปกรณ์ต่างๆ

1.4.5. ทดสอบเครื่องคัดแยกขวดอัตโนมัติโดยใช้เซ็นเซอร์ หาข้อผิดพลาดที่เกิดขึ้นและแก้ไขข้อผิดพลาด

1.4.6. แผนการดำเนินงาน

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	พ.ศ. 2562							พ.ศ. 2563	
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1. ศึกษาข้อมูลที่เกี่ยวข้อง		←	—	—	—	→			
2. เสนอโครงการ				←	—	→			
3. จัดหาอุปกรณ์	←	—	—	—	—	—	—	→	
4. สร้างเครื่องคัดแยกขวดอัตโนมัติ					←	—	—	→	
5. สร้างและเขียนโปรแกรม					←	—	—	→	
6. ทดสอบและแก้ไขโปรแกรม							←	→	
7. ทดสอบเครื่องคัดแยกขวดอัตโนมัติควบคุมด้วยอาดุยโน่							←	—	→
8. จัดทำรูปเล่มรายงาน					←	—	—	—	→

หมายเหตุ ←——→ ระยะเวลาการทำงานโดยประมาณ

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ได้คัดแยกขวดให้แบ่งตามประเภท

1.6.2 ได้ความสะดวกในการคัดแยกขวด

1.6.3 ได้นำกลับมารีไซเคิลไปใช้งานใหม่

บทที่ 2
เอกสารและทฤษฎีที่เกี่ยวข้อง

การศึกษาทฤษฎีที่เกี่ยวข้อง ผู้จัดทำโครงการได้ศึกษาทฤษฎีและเอกสารที่เกี่ยวข้องดังต่อไปนี้

2.1 โปรแกรมภาษาอาดุยโน่

2.2 Arduino IDE

2.3 บอร์ดอาดุยโน่

2.4 Servo

2.5 sensor

2.6 load cell

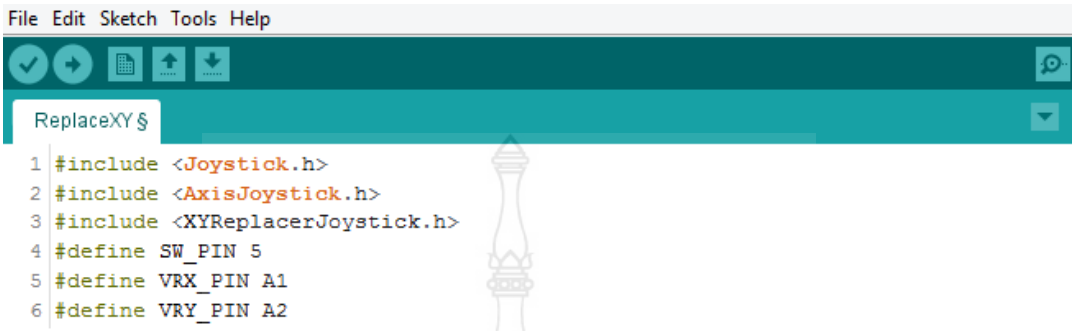
2.7 จอ LCD

2.1 โปรแกรมภาษาอาดุยโน่

โปรแกรมภาษาของ Arduino จะได้ภาษา C++ ซึ่งเป็นรูปแบบของโปรแกรมภาซีประยุกต์แบบหนึ่งที่มีโครงสร้างของตัวภาษาโดยรวมใกล้เคียงกันกับภาษาซีมาตรฐาน (ANSI-C) อื่นๆ เพียงแต่ได้มีการปรับปรุงแบบในการเขียนโปรแกรมบางส่วนที่ผิดเพี้ยนไปจาก ANSI-C เล็กน้อย เพื่อช่วยลดความยุ่งยากในการเขียน โปรแกรมและให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้ง่ายและสะดวกมากขึ้นกว่าการเขียนภาษาซีตาม แบบมาตรฐานของ ANSI-C สำหรับการเขียนโปรแกรมของ Arduino นั้นจะใช้ภาษา C++ ซึ่งเป็นรูปแบบภาษาซีประยุกต์แบบหนึ่ง ที่มีโครงสร้างการทำงานของตัวภาษาโดยรวมคล้ายกับภาษาซีมาตรฐาน (ANSI-C) ทั่วๆ ไป เพียงแต่ได้มีการปรับปรุงเพื่อลดความยุ่งยากในการใช้งานลง เพื่อให้ผู้ใช้สามารถใช้งานเขียนโปรแกรมได้ง่าย และสะดวกมากกว่าเขียนภาษาซี แบบมาตรฐาน โดยตรงซึ่งในความเป็นจริงแล้วในการเขียนโปรแกรมของ Arduino เราสามารถใช้คำสั่งต่างๆ ที่เป็นคำสั่งมาตรฐานของ ANSI-C เข้ามาใช้ในการเขียนโปรแกรมได้ทันที โดยรูปแบบการเขียนโปรแกรม และการใช้คำสั่งต่างๆ โดยโครงสร้างการเขียนโปรแกรมของ Arduino นั้นทุกๆ โปรแกรมจะต้องประกอบไปด้วยฟังก์ชันจำนวนเท่าใดก็ได้ แต่อย่างน้อยที่สุดต้องมีฟังชั่น จำนวน 2 ฟังก์ชั่น คือ Setup() และLoop() โครงสร้างพื้นฐานของภาษาซีที่ใช้กับ Arduino นั้นจะประกอบไปด้วย 3 ส่วนใหญ่ๆ ด้วยกัน คือ

1. Header ในส่วนนี้จะมีหรือไม่มีก็ได้ ถ้ามีต้องกำหนดไว้ในส่วนเริ่มต้นของโปรแกรมซึ่งส่วนของ Header ได้แก่ ส่วนที่เป็น Compiler Directive ต่างๆ รวมไปถึงส่วนของการประกาศตัวแปรและค่าคงที่ต่างๆ ที่จะใช้ในโปรแกรมเมื่อพบคำสั่ง #include ตัวแปลภาษาของ Arduino จะไปค้นหาไฟล์ที่ระบุไว้ใน เครื่องหมาย <> หลังคำสั่ง #include จากตำแหน่ง Directory ที่เก็บไฟล์ Library ของโปรแกรม Arduino ไว้ ซึ่งแน่นอนว่าส่วนของ Header จะนับรวมไปถึงคำสั่งส่วนที่ใช้ประการ

สร้างตัวแปร (Variable Declaration) และค่าคงที่ (Constant Declaration) รวมทั้งฟังก์ชั่นต่างๆ (Function Declaration) ด้วยซึ่งภาพที่ 2.1 ได้แก่ส่วนที่เป็นคำสั่ง

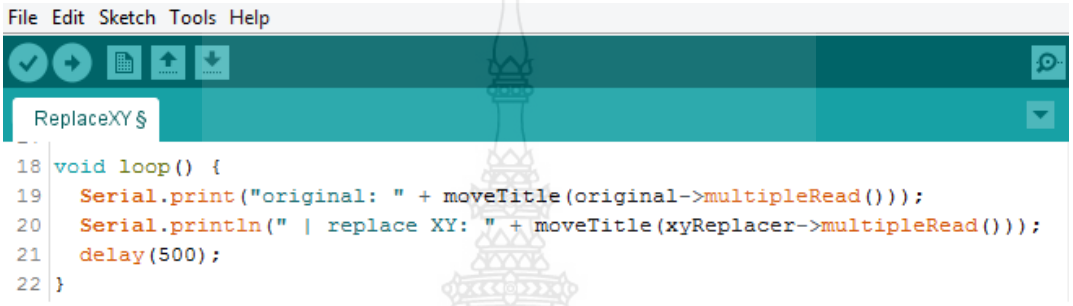


ภาพที่ 2.1 Header โปรแกรม
ที่มา : https://poundxi.com/

2. setup ในส่วนนี้เป็นฟังก์ชั่นบังคับที่ต้องกำหนดให้มีในทุกๆ โปรแกรม ถึงแม้ว่าในภาพเขียนคำสั่งใดๆ ไว้ในระหว่างวงเล็บปีกกา {} ที่ใช้เป็นตัวกำหนดของเขตของฟังก์ชั่น โดยฟังก์ชั่นนี้จะใช้สำหรับบรรจุคำสั่ง ในส่วนที่ต้องกาให้โปรแกรมทำงานเพียงรอบเดียวตอนเริ่มต้นทำงานของโปรแกรมครั้งแรกเท่านั้น ซึ่งได้แก่ คำสั่งเกี่ยวกับการ Setup ค่าการทำงานต่างๆ เช่น การกำหนดหน้าที่การใช้งานของ Pin Mode และการกำหนดค่า Baud rate สำหรับใช้งานพอร์ตสื่อสารอนุกรม เป็นต้น หน้าที่ของฟังก์ชั่น setup() ใน Arduino คือ ใช้ทำหน้าที่เป็นส่วนของโปรแกรมย่อยสำหรับใช้บรรจุคำสั่งต่างๆ ที่ใช้สำหรับกำหนดการทำงานของระบบ หรือ กำหนดคุณสมบัติการทำงานให้กับอุปกรณ์ต่างๆ ซึ่งคำสั่งทั้งหมดที่บรรจุไว้ภายใต้ฟังก์ชั่นของ Setup() นี้ จะถูกเรียกขึ้นมาทำงานเพียงรอบเดียวคือตอนเริ่มต้นการทำงานของโปรแกรม โดยคำสั่งที่นิยมบรรจุไว้ในฟังก์ชั่นส่วน ดังภาพที่ 2.2

ภาพที่ 2.2 Setup โปรแกรม
ที่มา : https://poundxi.com/

3. loop เป็นส่วนฟังก์ชั่นบังคับที่ต้องกำหนดให้มีในทุกๆ โปรแกรมเช่นเดียวกันกับฟังก์ชั่น setup() โดยฟังก์ชั่น loop() นี้จะใช้บรรจุคำสั่งที่ต้องการให้โปรแกรมทำงานเป็นวงรอบซ้ำๆ กันไปไม่รู้จบ ซึ่งถ้าเปรียบเทียบกับรูปแบบของ ANSI-C ส่วนนี้ก็คือ ฟังก์ชั่น main() นั่นเอง หน้าที่ของฟังก์ชั่น loop() ใน Arduino คือใช้ทำหน้าที่เป็นส่วนของโปรแกรมหลัก สำหรับใช้บรรจุคำสั่งควบคุมการทำงานต่างๆ ของ 18 โปรแกรม ที่ต้องการใช้โปรแกรมทำงาน โดยคำสั่งที่บรรจุไว้ในฟังก์ชั่นนี้จะถูกเรียกขึ้นมาทำงานซ้ำๆ กัน ตามลำดับและเงื่อนไขที่กำหนดไว้



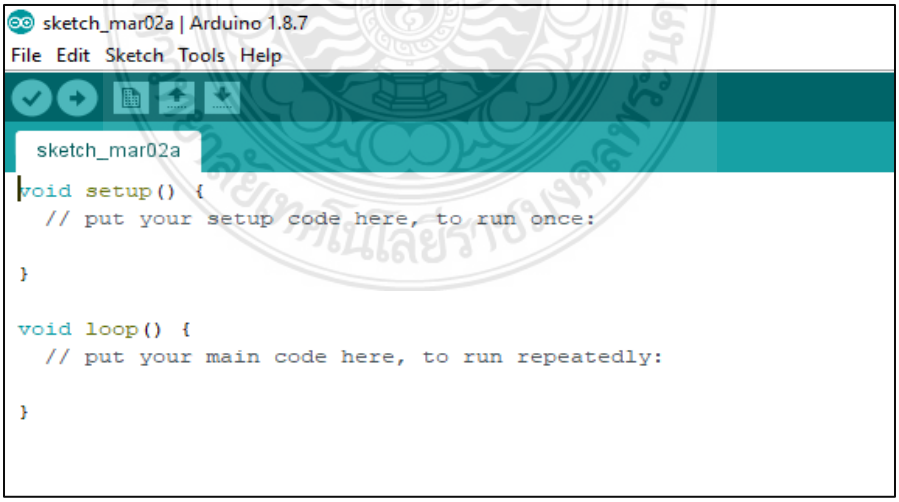
ภาพที่ 2.3 Loop โปรแกรม
ที่มา : https://poundxi.com/

2.2 Arduino IDE

Arduino IDE คือโปรแกรมสำหรับใช้เขียนโปรแกรม, คอมไพล์ และอัพโหลดโปรแกรมลงบอร์ด Arduino หรือบอร์ดตัวอื่นๆ ที่คล้ายกัน เช่น NodeMCU หรือ WeMos D1 เป็นต้น

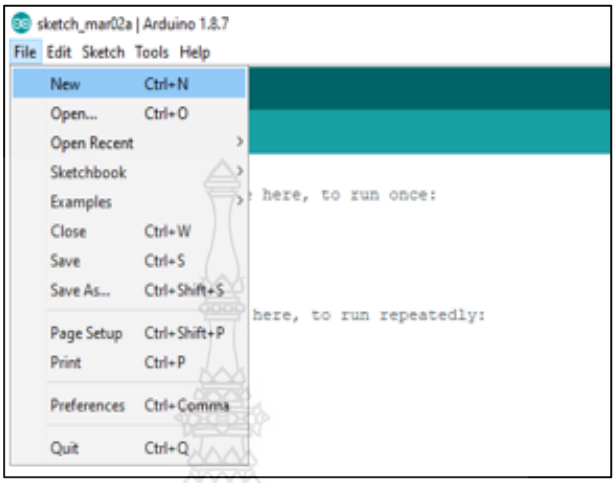
วิธีการใช้งาน Arduino IDE

ขั้นตอนที่ 1 เปิดโปรแกรม Arduino IDE ขึ้นมา ดังภาพที่ 2.4



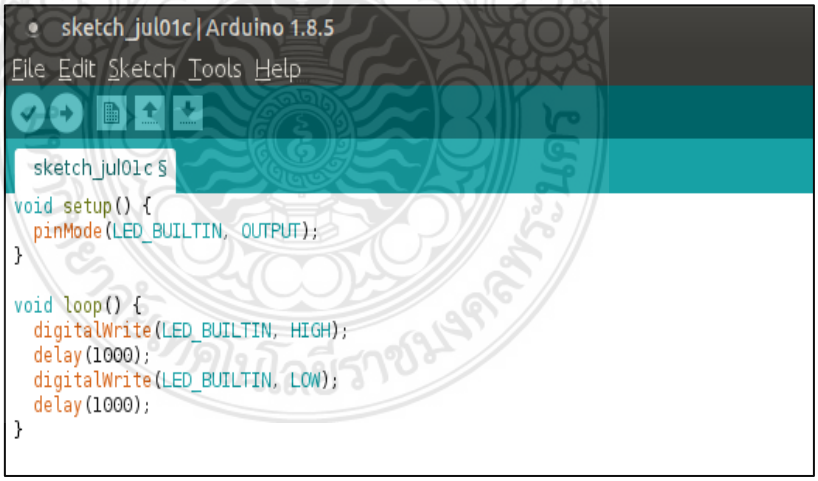
ภาพที่ 2.4 เปิดโปรแกรม Arduino IDE
ที่มา : https://poundxi.com/

ขั้นตอนที่ 2 สร้าง Sketch ใหม่ โดยคลิกที่เมนู File > New ดังภาพที่ 2.5



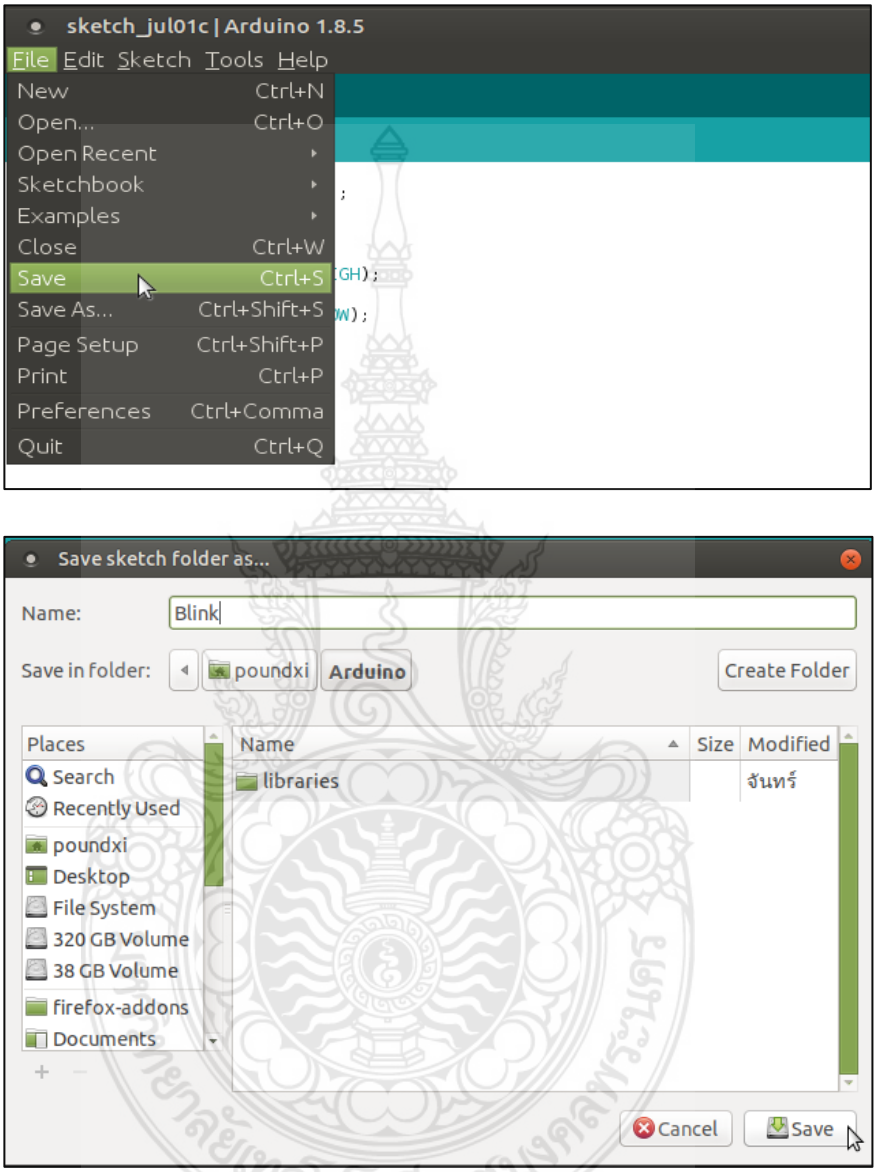
ภาพที่ 2.5 สร้าง Sketch ใหม่
ที่มา : https://poundxi.com

ขั้นตอนที่ 3 เขียนโปรแกรมลงไปตรงพื้นที่สีขาวๆ ซึ่งการเขียนโปรแกรมสำหรับ Arduino จะประกอบไปด้วยฟังก์ชั่น void setup และ ฟังก์ชั่น void loop ดังภาพที่ 2.6



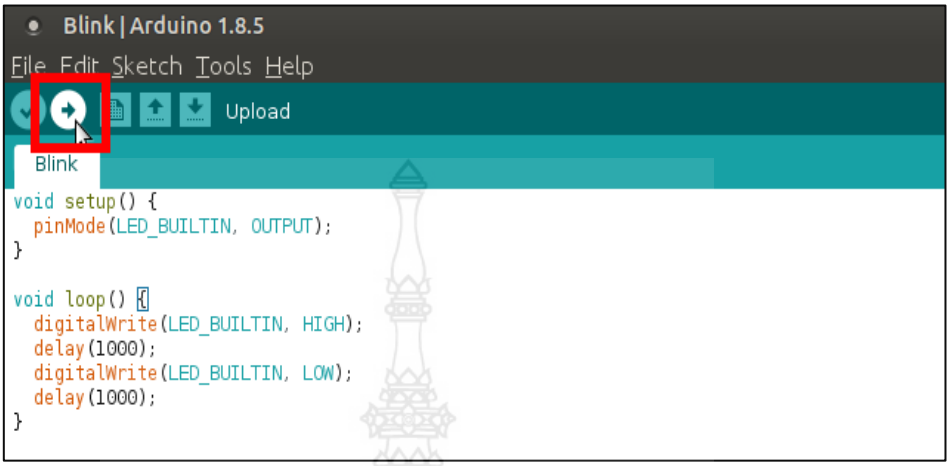
ภาพที่ 2.6 เขียนโปรแกรม
ที่มา : https://poundxi.com

ขั้นตอนที่ 4 เมื่อเขียนโปรแกรมเสร็จแล้วให้บันทึกเก็บไว้โดยคลิกที่เมนู File > Save จากนั้นจะมีหน้าต่างขึ้นมาให้เราเลือกว่าจะบันทึกไว้ที่ไหน และตั้งชื่อ Sketch ดังภาพที่ 2.7



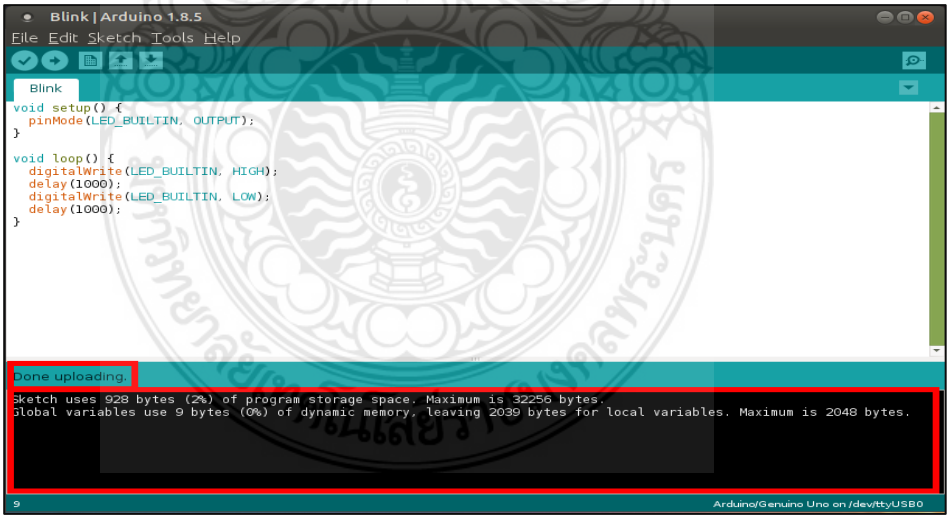
ภาพที่ 2.7 การบันทึกโปรแกรม
ที่มา : https://poundxi.com

ขั้นตอนที่ 5 คลิกปุ่มอัพโหลด หรือจะคลิกที่เมนู Sketch > Upload ดังภาพที่ 2.8



ภาพที่ 2.8 การอัพโหลดโปรแกรมลงบอร์ด
ที่มา : https://poundxi.com

ขั้นตอนที่ 6 หากเขียนโปรแกรมถูกต้อง และไม่ได้มีปัญหาระหว่างการอัพโหลด จะมีข้อความขึ้นว่า Done uploading และจะมีข้อความรายงานเป็นข้อความสีขาวๆ ดังภาพที่ 2.9



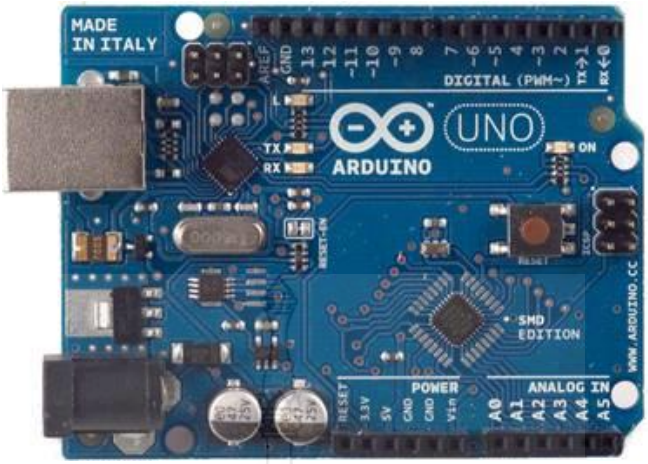
ภาพที่ 2.9 การอัพโหลดบอร์ดเสร็จสิ้น
ที่มา : https://poundxi.com

2.3 บอร์ดอาดุยโน่

ภาพที่ 2.10 บอร์ด Arduino Uno R3
ที่มา : https://sites.google.com/site/karanwinatktech/unit1

คำว่า Uno เป็นภาษาอิตาลี ซึ่งแปลว่าหนึ่งเป็นบอร์ด Arduino รุ่นแรกที่ออกมามีขนาดประมาณ 68.6x53.4 mm เป็นบอร์ดมาตรฐานที่นิยมใช้งานมากที่สุด เนื่องจากเป็นขนาดที่เหมาะสำหรับการเริ่มต้นเรียนรู้ Arduino และมี Shields ให้เลือกใช้งานได้มากกว่าบอร์ด Arduino รุ่นอื่นๆ ที่ออกแบบมาเฉพาะมากกว่า โดยบอร์ด Arduino Uno ได้มีการพัฒนาเรื่อยมา ตั้งแต่ R2 R3 และรุ่นย่อยที่เปลี่ยนชิปไอซีเป็นแบบ SMD

ภาพที่ 2.11 บอร์ด Arduino Uno R2
ที่มา : https://sites.google.com/site/karanwinatktech/unit1



ภาพที่ 2.12 บอร์ด Arduino Uno SMD
ที่มา : https://sites.google.com/site/karanwinatktech/unit1

ข้อมูลจำเพาะ

ชิปไอซีไมโครคอนโทรเลอร์	ATmega328
ใช้แรงดันไฟฟ้า	5V
รองรับการจ่ายแรงดันไฟฟ้า (ที่แนะนำ)	7 – 12V
รองรับการจ่ายแรงดันไฟฟ้า (ที่จำกัด)	6 – 20V
พอร์ต Digital I/O	14 พอร์ต (มี 6 พอร์ต PWM output)
พอร์ต Analog Input	6 พอร์ต
กระแสไฟที่จ่ายได้ในแต่ละพอร์ต	40mA
กระแสไฟที่จ่ายได้ในพอร์ต 3.3V	50mA
พื้นที่โปรแกรมภายใน	32KB พื้นที่โปรแกรม, 500B ใช้โดย Booloader
พื้นที่แรม	2KB
พื้นที่หน่วยความจำถาวร (EEPROM)	1KB
ความถี่คริสตัล	16MHz
ขนาด	68.6x53.4 mm
น้ำหนัก	25 กรัม

2.4 Servo

ภาพที่ 2.13 MG995 Servo Motor 0-180 องศา
ที่มา : https://www.myarduino.net/

MG995 Tower Pro MG995 DIGI Hi-Speed Servo
Servo แรงบิดสูงเฟืองเป็นโลหะแข็งแรง แรงบิดสูงถึง 12 Kg หมุน 0-180องศา

Specification

- Dimension : 40mm x 19mm x 43mm
- Weight : 55g
- Operating Speed : 0.17sec / 60 degrees (4.8V no load)
- Operating Speed : 0.13sec / 60 degrees (6.0V no load)
- Stall Torque : 9 kg-cm (180.5 oz-in) at 4.8V
- Stall Torque : 12 kg-cm (208.3 oz-in) at 6V
- Operation Voltage : 4.8 - 7.2Volts
- Gear Type: All Metal Gears
- Original box: NO
- Color: Black
- Connector Wire: Heavy Duty, 11.81" (300mm)

ตัวอย่างการใช้งาน

- Servo -> Arduino
- สายสีน้ำตาล -> GND
- สายสีแดง ->5V
- สายสีส้ม -> Pin9

ตัวอย่างcode

```
#include <Servo.h>
Servo myservo;
void setup()
{
  myservo.attach(9);
}
void loop()
{       myservo.write(0);
        delay(1000);
        myservo.write(90);
        delay(1000);
        myservo.write(180);
        delay(1000);
}
```

2.5 sensor

ภาพที่ 2.14 เซนเซอร์ตรวจจับโลหะระยะ 8mm Inductive proximity switch sensor

ที่มา : https://www.arduinothai.com/

เซนเซอร์ตรวจจับโลหะระยะ 8 mm Inductive proximity switch sensor เป็นเซนเซอร์ตรวจจับโลหะได้ในระยะสูงสุด 8 mm เมื่อสามารถตรวจจับโลหะได้ จะมีไฟ LED สีแดงแสดงที่ด้านหลังเซนเซอร์ ให้ค่าเอาท์พุตเป็น 1 ถ้าตรวจจับไม่เจอโลหะไฟ LED ก็จะไม่ติดเอาท์พุตเป็น 0 เซนเซอร์เป็นชนิด PNP สายไฟยาว 1.2 เมตร

Specification

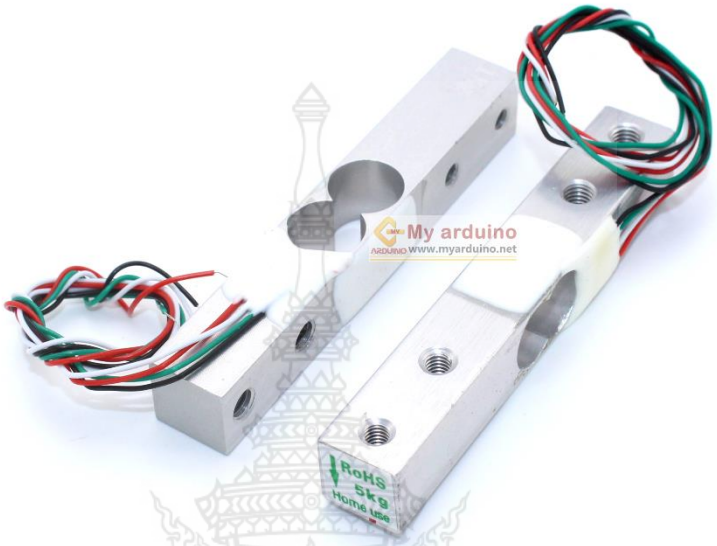
Switch Appearance Type : Cylinder Type
Theory : Capacitance Sensor;
Output Type : NO(Normal Open);
Diameter of Head : 18mm;
Detecting Distance : 8mm
Detecting Object : Metal or Nonmetal Material;
Supply Voltage : DC 6-36V;
Current Output : 300mA;
Response Frequency : 100Hz
Size : 7.5 x 1.8cm (L*Max. Dia);
Material : Plastic, Alloy

การต่อใช้งาน

สีน้ำเงิน - GND
สีน้ำตาล - VCC ไฟเลี้ยง 5-36V
สีดำ - OUTPUT สายสัญญาณ

2.6 load cell

โหลดเซลล์ Load Cell คืออุปกรณ์ที่ใช้ในการเปลี่ยนจากแรงหรือน้ำหนักที่กระทำต่อตัวโหลดเซลล์ เป็นสัญญาณทางไฟฟ้า ทางเราสามารถนำสัญญาณทางไฟฟ้านี้ไปจ่ายเข้าจอแสดงผล Display แสดงค่าเป็นน้ำหนักหรือแรงที่กระทำให้คนเห็นได้ Load cell สร้างมาจาก Strain Gauge ที่จัดเรียงวงจรในรูปแบบของวงจรวิจสโตนบริดจ์ (Wheatstone Bridge) ส่งค่า output ออกมาเป็นสัญญาณทางไฟฟ้า Load cell สามารถเอาไปประยุกต์ทำเครื่องชั่งตวงในอุตสาหกรรมได้ (วัดแรงกด Compression) หรือ ใช้ทดสอบวัสดุ (วัดแรงดึง Tensile) ได้อีกด้วย

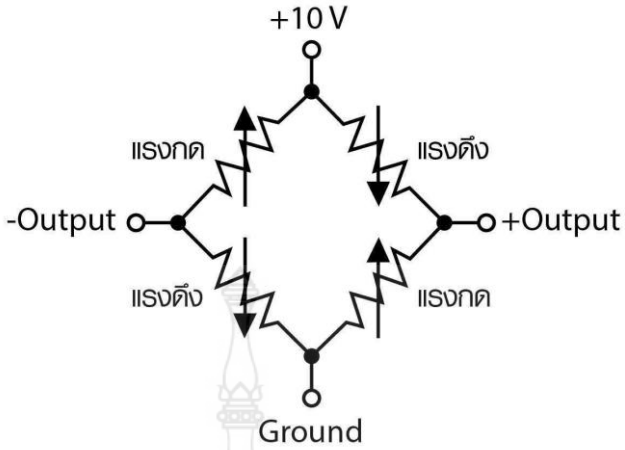


ภาพที่ 2.15 Load cell

ที่มา : https://www.factomart.com/th/load-cell-what-is-it.html

หลักการทำงานของโหลดเซลล์

โหลดเซลล์ คือ เซนเซอร์ที่สามารถแปลงค่าแรงกด หรือแรงดึง ให้เป็นสัญญาณทางไฟฟ้าได้ โดยการเปลี่ยนแปลงค่าความต้านทานตาม แรงกด หรือ แรงดึง โหลดเซลล์ถูกนำไปใช้ในอุตสาหกรรมหลากหลายประเภท ได้แก่ การชั่งน้ำหนัก การทดสอบแรงกดของชิ้นงาน การทดสอบความแข็งแรงของชิ้นงาน การทดสอบการเข้ารูปชิ้นงาน (Press fit) ใช้สำหรับงานทางด้านวัสดุ โลหะ ทดสอบโลหะชิ้นส่วนรถยนต์ วิศวกรรมโยธา ทดสอบคอนกรีต ทดสอบไม้ฯลฯ โหลดเซลล์แบบสเตรนเกจ (Strain Gauge Load cell) หลักการของโหลดเซลล์ประเภทนี้ก็คือ เมื่อมีน้ำหนักมากระทำความเครียด (Strain) จะเปลี่ยนเป็นความต้านทานทางไฟฟ้าในสัดส่วนโดยตรงกับแรงที่มากระทำปรกติแล้วมักจะใช้เกจวัดความเครียด 4 ตัว



ภาพที่ 2.16 การวัดโดยเกจตัวต้านทานทั้งสี่

ที่มา : https://www.factomart.com/th/load-cell-what-is-it.html

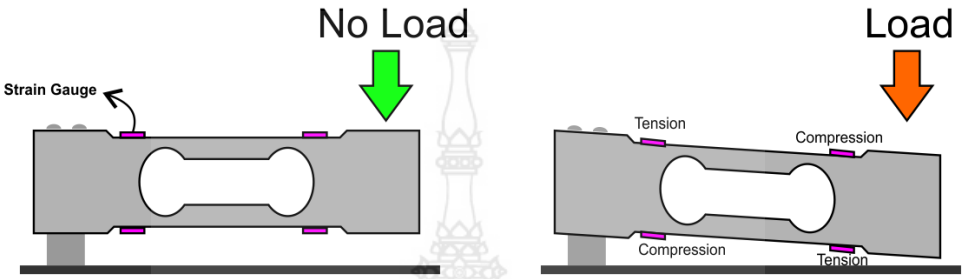
การวัดโดยเกจตัวต้านทานทั้งสี่จะเชื่อมต่อเข้าด้วยกันเพื่อใช้แปลงแรงที่กระทำกับตัวของมันไม่ว่าจะเป็นแรงกด หรือแรงดึงส่งสัญญาณออกมาเป็นแรงดันไฟฟ้า โดยที่แรงดันไฟฟ้าจะมีหน่วยเป็น mV/V หมายความว่าถ้าจ่าย แรงดัน 10 V ให้กับ Load cell ที่มี Spec 2 mV/V ที่ Full load สมมุติว่าถ้าน้ำหนัก Full Load เป็น 100 กิโลกรัม ดังนั้นเมื่อมีแรงกระทำต่อ Load cell ที่น้ำหนัก 100 กิโลกรัม สัญญาณที่จะได้ก็จะได้เท่ากับ 20 mV

วิธีการใช้งาน Load Cell กับ HX711 Amplifier Module

ภาพที่ 2.17 Load cell

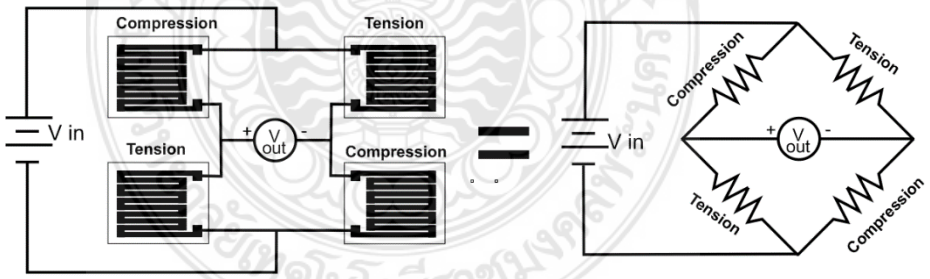
ที่มา : https://www.thaieasyelec.com/

Load Cell คือ Sensor สำหรับตรวจวัดน้ำหนักแรงกระทำทางกลหรือปริมาณของ Load ที่ต้องการทราบค่า โดยใช้ Strain Gauge มาติดตั้งในบริเวณที่มีการเปลี่ยนแปลงรูปทรงของ Load Cell เมื่อมีแรงมากระทำกับตัว Load Cell จะทำให้ Strain Gauge ที่ติดอยู่ในบริเวณที่มีการเปลี่ยนรูปทรง ยืด หรือ หดตัว ทำให้ค่าความต้านทานที่ตัว Strain Gauge เปลี่ยนไป



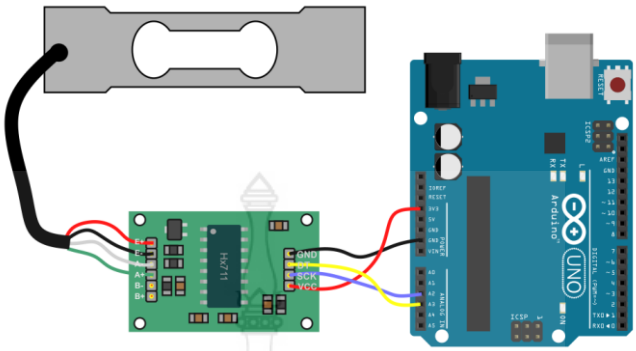
ภาพที่ 2.18 Load cell สำหรับตรวจวัดน้ำหนัก
ที่มา : https://www.thaieasyelec.com/

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



ภาพที่ 2.19 Wheatstone Bridge
ที่มา : https://www.thaieasyelec.com/

วิธีเชื่อมต่อ Load Cell และ HX711 เข้ากับ Arduino



ภาพที่ 2.20 วิธีเชื่อมต่อ Load Cell และ HX711 เข้ากับ Arduino
ที่มา : https://www.thaieasyelec.com/

Load Cell	HX711(China)	HX711(SparkFun)
RED	E+	RED/E+
BLACK	E-	BLK/E-
WHITE	A-	WHT/A-
GREEN	A+	GRN/A+

ภาพที่ 2.21 ขาอุปกรณ์ HX711
ที่มา : https://www.thaieasyelec.com/

วิธี Calibrate Load Cell

การ Calibrate นั้นเราจำเป็นจะต้องมี Load หรือ น้ำหนักที่เราทราบค่าอยู่แล้ว เพื่อทำการ Calibrate โดยทั่วไปมักจะใช้ตุ้มน้ำหนักมาตรฐานเพื่อทำการ Calibrate

ภาพที่ 2.22 ตุ้มน้ำหนักมาตรฐานเพื่อทำการ Calibrate
ที่มา : https://www.thaieasyelec.com/

แต่สำหรับผู้ที่ไม่มี และอยากจะทดลองในเบื้องต้นก่อนเราก็สามารถใช้สิ่งของรอบๆตัวเราที่พอจะทราบน้ำหนักมาใช้ทดลองก่อนได้ เช่น น้ำเปล่า 1 ลิตร มีน้ำหนักเท่ากับ 1 กิโลกรัม ในการทดลองนี้เราจึงนำขวดน้ำขนาด 1.5 ลิตรที่มีขายตามท้องตลาดมาทดลอง ซึ่งจากการชั่งด้วยเครื่องชั่งที่มีพบว่าน้ำขวดนี้มีน้ำหนักประมาณ 1.56x กิโลกรัม

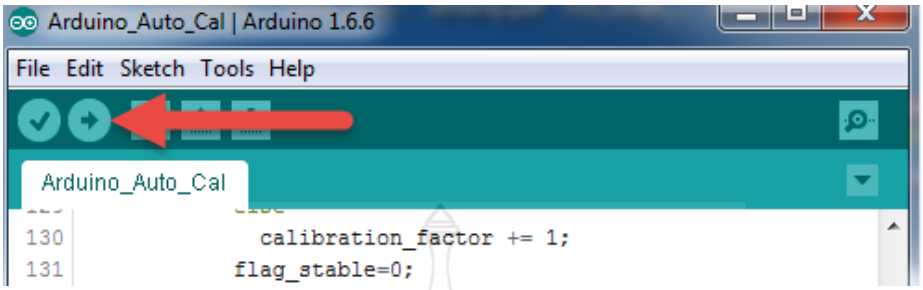
ภาพที่ 2.23 ขวดน้ำเพื่อทำการ Calibrate
ที่มา : https://www.thaieasyelec.com/

เริ่มทดลอง Calibrate
>> 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
```

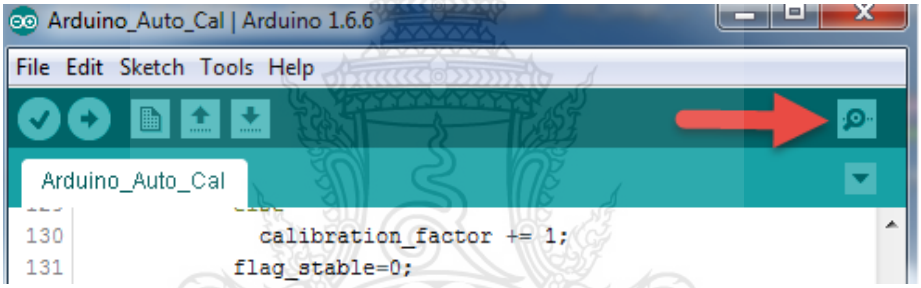
ภาพที่ 2.24 ค่าน้ำหนักขวดน้ำ
ที่มา : https://www.thaieasyelec.com/

>> โปรแกรม Arduino_Auto_Cal ลงไปยัง board Arduino



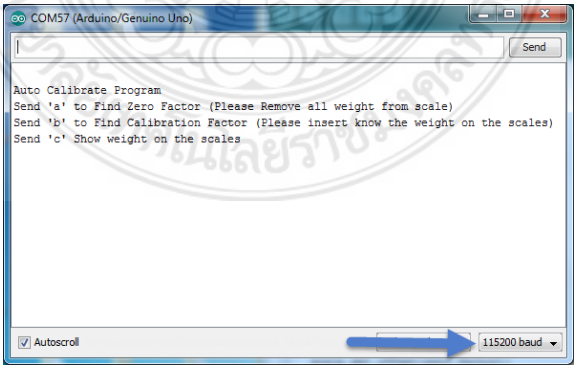
ภาพที่ 2.25 โปรแกรม Arduino_Auto_Cal ลงไปยัง board Arduino
ที่มา : https://www.thaieasyelec.com/

>> เปิดหน้าต่าง Serial Monitor ขึ้นมา



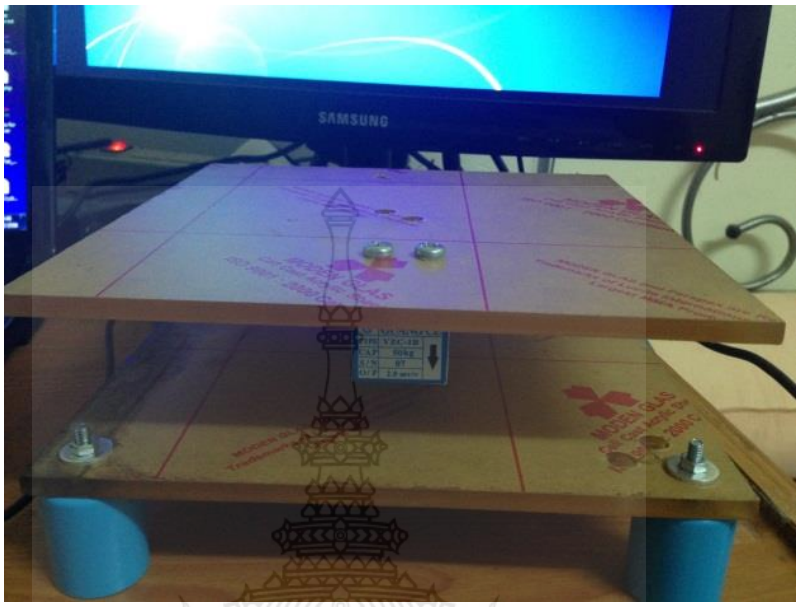
ภาพที่ 2.26 เปิด หน้าต่าง Serial Monitor ขึ้นมา
ที่มา : https://www.thaieasyelec.com/

>> ตั้งค่า Baud rate เป็น 115200



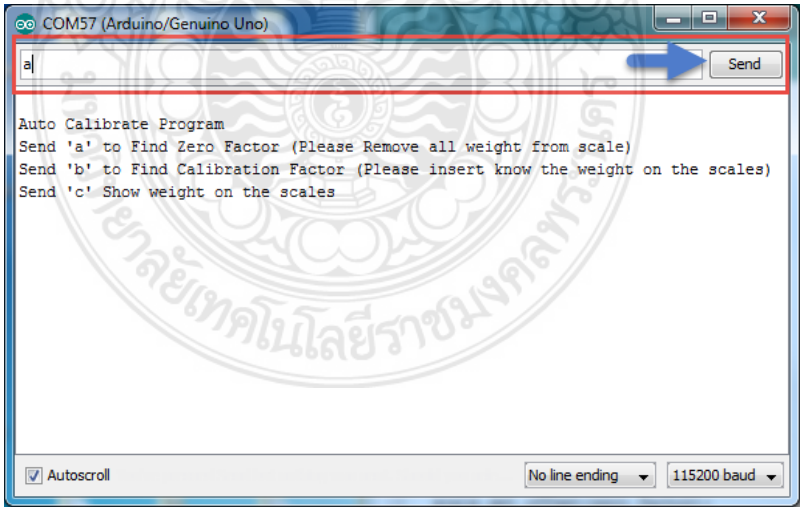
ภาพที่ 2.27 ตั้งค่า Baud rate เป็น 115200
ที่มา : https://www.thaieasyelec.com/

>> นำสิ่งของทุกอย่างออกจากแผ่นชั่งน้ำหนัก



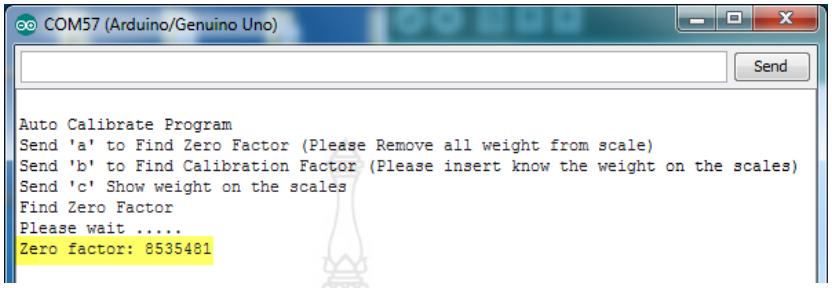
ภาพที่ 2.28 นำสิ่งของทุกอย่างออกจากแผ่นชั่งน้ำหนัก
ที่มา : https://www.thaieasyelec.com/

>> ส่งตัว ‘a’ ไปเพื่อหาค่า zero



ภาพที่ 2.29 ส่ง ตัว ‘a’ ไป เพื่อหาค่า zero
ที่มา : https://www.thaieasyelec.com/

>> รอสักครู่โปรแกรมจะแสดงค่า Zero Factor ออกมาให้เราจดบันทึกค่านี้เอาไว้ใช้งาน (ในการทดลองนี้คือ 8535481)

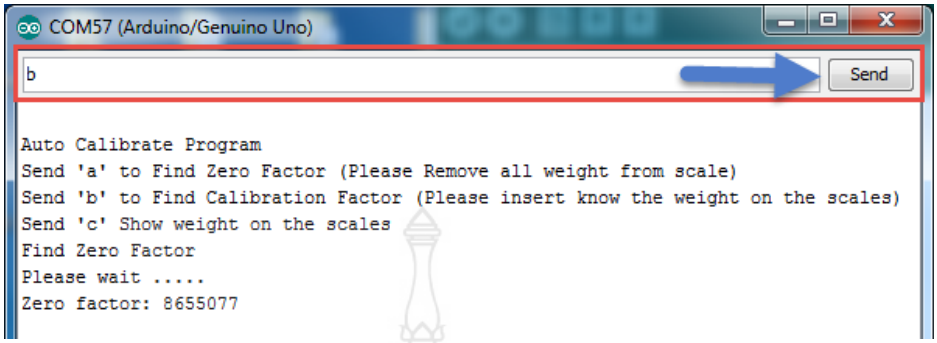


ภาพที่ 2.30 ค่า Zero Factor ออกมาให้ ให้เราจดบันทึกค่านี้เอาไว้ใช้งาน
ที่มา : https://www.thaieasyelec.com/

>> นำวัตถุที่ทราบค่าน้ำหนักที่แน่นอนมาวางลงบนแผ่นชั่งน้ำหนัก (น้ำหนักของวัตถุที่ใส่ลงในตัวแปร real_weight ใน Arduino_Auto_Cal.ino)

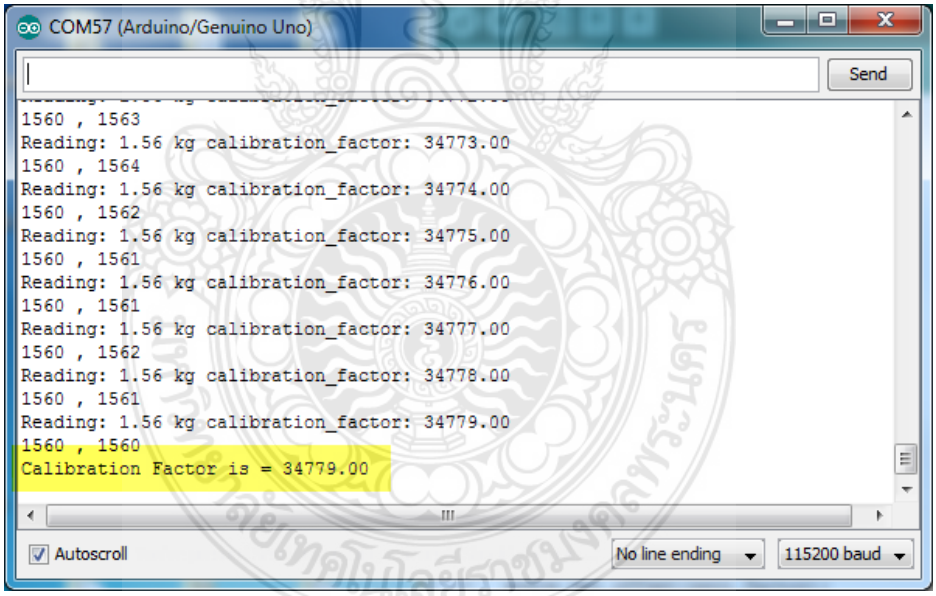
ภาพที่ 2.31 นำวัตถุที่ ทราบค่าน้ำหนักที่แน่นอนมาวางลงบนแผ่นชั่งน้ำหนัก
ที่มา : https://www.thaieasyelec.com/

>> ส่งตัว ‘b’ ไปเพื่อหาค่า calibration factor



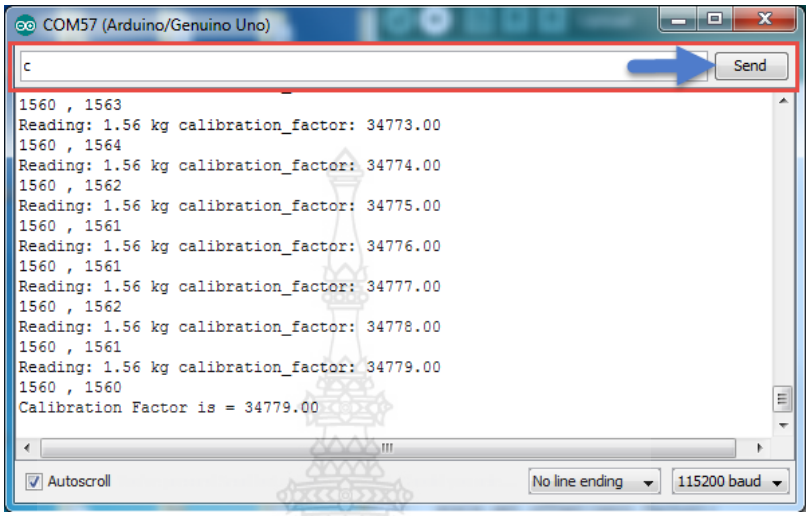
ภาพที่ 2.32 ส่ง ตัว ‘b’ ไป เพื่อหาค่า calibration factor
ที่มา : https://www.thaieasyelec.com/

>> รอจนโปรแกรมสามารถหาค่า calibration factor ได้สำเร็จให้เราจดบันทึกค่านี้เอาไว้ใช้งาน (ในการทดลองนี้คือ 34779.00)



ภาพที่ 2.33 ค่า calibration factor ได้สำเร็จ ให้เราจดบันทึกค่านี้เอาไว้ใช้งาน
ที่มา : https://www.thaieasyelec.com/

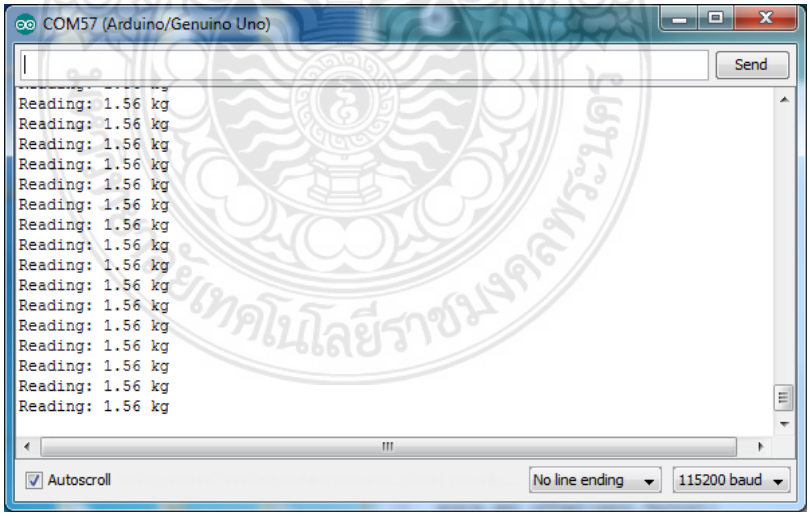
>> ส่งตัว ‘c’ ไปเพื่อทดลองอ่านค่าน้ำหนักโดยใช้ Zero Factor และ calibration factor ที่ใช้โปรแกรมหาค่ามา



ภาพที่ 2.34 ส่ง ตัว ‘c’ ไป เพื่อทดลองอ่านค่าน้ำหนักโดยใช้ Zero Factor และ calibration factor ที่ใช้โปรแกรมหาค่ามา

ที่มา : https://www.thaieasyelec.com/

>> ผลการทดลองอ่านค่า



ภาพที่ 2.35 ผลการทดลองอ่านค่า

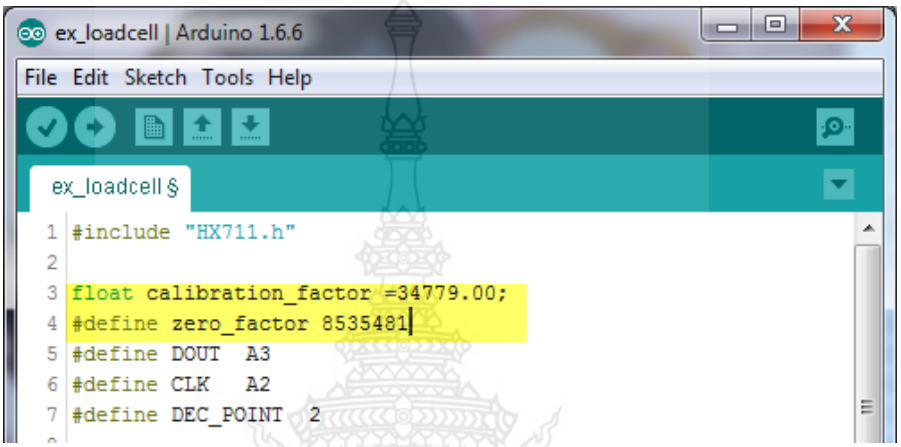
ที่มา : https://www.thaieasyelec.com/

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

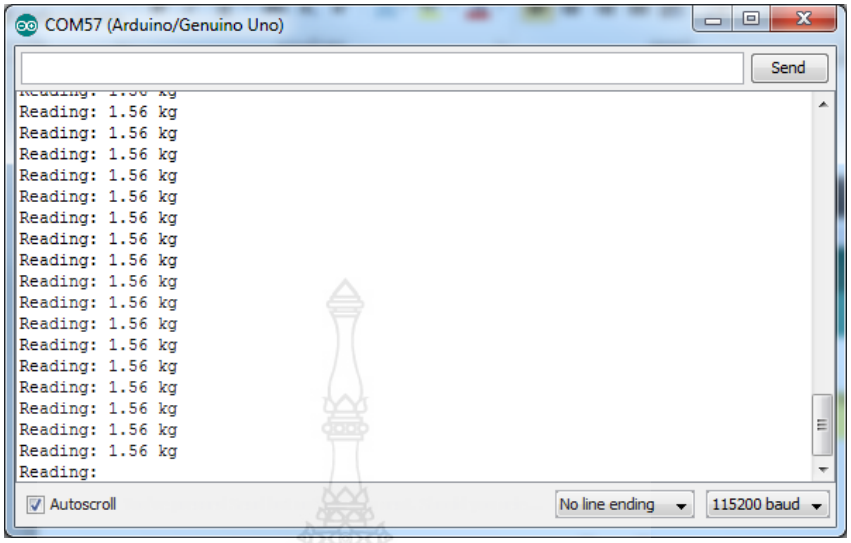
>> ให้เรานำเอาค่า Zero Factor และ calibration factor มาใส่ในตัวแปรแทนของเดิม



ภาพที่ 2.36 ให้เรานำเอาค่า Zero Factor และ calibration factor มาใส่ในตัวแปรแทนของเดิม
ที่มา : https://www.thaieasyelec.com/

>> ทดลอง Run Program โดยชั่งขวดน้ำ (1.56 Kg)

ภาพที่ 2.37 ทดลอง Run Program โดยชั่งขวดน้ำ (1.56 Kg)
ที่มา : https://www.thaieasyelec.com/



ภาพที่ 2.38 ผลทดลอง Run Program โดยชั่งขวดน้ำ (1.56 Kg)
ที่มา : https://www.thaieasyelec.com/

2.7 จอ LCD

คำว่า LCD ย่อมาจากคำว่า Liquid Crystal Display ซึ่งเป็นจอที่ทำมาจากผลึกคริสตอลเหลว หลักการคือด้านหลังจอจะมีไฟส่องสว่างหรือที่เรียกว่า Backlight อยู่เมื่อมีการปล่อยกระแสไฟฟ้าเข้าไปกระตุ้นที่ผลึกก็จะทำให้ผลึกโปร่งแสงทำให้แสงที่มาจากไฟ Backlight แสดงขึ้นมาบนหน้าจอส่วนอื่นที่โดนผลึกปิดกั้นไว้ จะมีสีที่แตกต่างกันตามสีของผลึกคริสตอล เช่น สีเขียว หรือ สีฟ้า ทำให้เมื่อมองไปที่จอก็จะพบกับตัวหนังสือสีขาวแล้วพบกับพื้นหลังสีต่างๆ กันจอ LCD จะแบ่งเป็น 2 แบบใหญ่ๆตามลักษณะการแสดงผลดังนี้

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

2. Graphic LCD เป็นจอที่สามารถกำหนดได้ว่าจะให้แต่ละจุดบนหน้าจอกั้นแสงหรือปล่อยแสงออกไป ทำให้จอนี้สามารถสร้างรูปขึ้นมาบนหน้าจอได้การระบุขนาดจะระบุในลักษณะของจำนวนจุด (Pixels) ในแต่ละแนว เช่น 128x64 หมายถึงจอที่มีจำนวนจุดตามแนวนอน 128 จุด และมีจุดตามแนวตั้ง 64 จุด

ในบทความนี้จะกล่าวถึง Character LCD เพียงอย่างเดียว เนื่องจากใช้งานได้ง่าย และนิยมใช้งานในโปรเจคทั่วๆไปมากกว่าครับการเชื่อมต่อกับจอ Character LCD

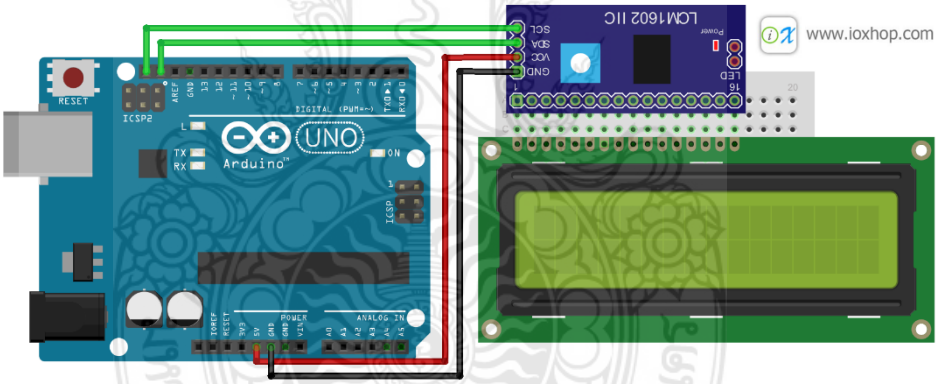
การเชื่อมต่อจะมีด้วยกัน 2 แบบ คือ

- การเชื่อมต่อแบบขนาน - เป็นการเชื่อมต่อจอ LCD เข้ากับบอร์ด Arduino โดยตรง โดยจะแบ่งเป็นการเชื่อมต่อแบบ 4 บิต และการเชื่อมต่อแบบ 8 บิต ใน Arduino จะนิยมเชื่อมต่อแบบ 4 บิต เนื่องจากใช้สายในการเชื่อมต่อน้อยกว่า
- การเชื่อมต่อแบบอนุกรม - เป็นการเชื่อต่อกับจอ LCD ผ่านโมดูลแปลงรูปแบบการเชื่อมต่อกับจอ LCD จากแบบขนานมาเป็นการเชื่อมต่อแบบอื่นที่ใช้สายน้อยกว่า เช่น การใช้โมดูล I2C Serial Interface จะเป็นการนำโมดูลเชื่อมเข้ากับตัวจอ LCD แล้วใช้บอร์ด Arduino เชื่อมต่อกับบอร์ดโมดูลผ่านโปรโตคอล I2C ทำให้ใช่สายเพียง 4 เส้นก็ทำให้หน้าจอแสดงผลข้อความต่างๆออกมาได้

การเชื่อมต่อแบบอนุกรม (LCD I2C)

การเชื่อมต่อแบบอนุกรม จะใช้งานโมดูล I2C Serial Interface Board Module มาเชื่อมต่อระหว่าง Arduino กับจอ LCD

วงจรที่เชื่อมต่อจะเป็นไปตามรูปนี้ (กรณีใช้บอร์ดรุ่นอื่น จะต้องต่อ SDA เข้า A4 และ SCL เข้ากับ A5)



ภาพที่ 2.39 การเชื่อมต่อแบบอนุกรม จะใช้งานโมดูล I2C Serial Interface Board Module มาเชื่อมต่อระหว่าง Arduino กับจอ LCD

ที่มา : https://www.ioxhop.com/

บทที่ 3
ขั้นตอนและวิธีดำเนินงาน

การจัดทำโครงการ เครื่องคัดแยกขวดอัตโนมัติ โดยใช้โปรแกรม Arduino โดยการควบคุม เข้ามาช่วยในการควบคุมการคัดแยกขวด กระป๋อง และแก้ว โดยทางผู้จัดทำได้จัดลำดับการดำเนินการตามขั้นตอนและวิธีการดังนี้

3.1 ขั้นการดำเนินงาน

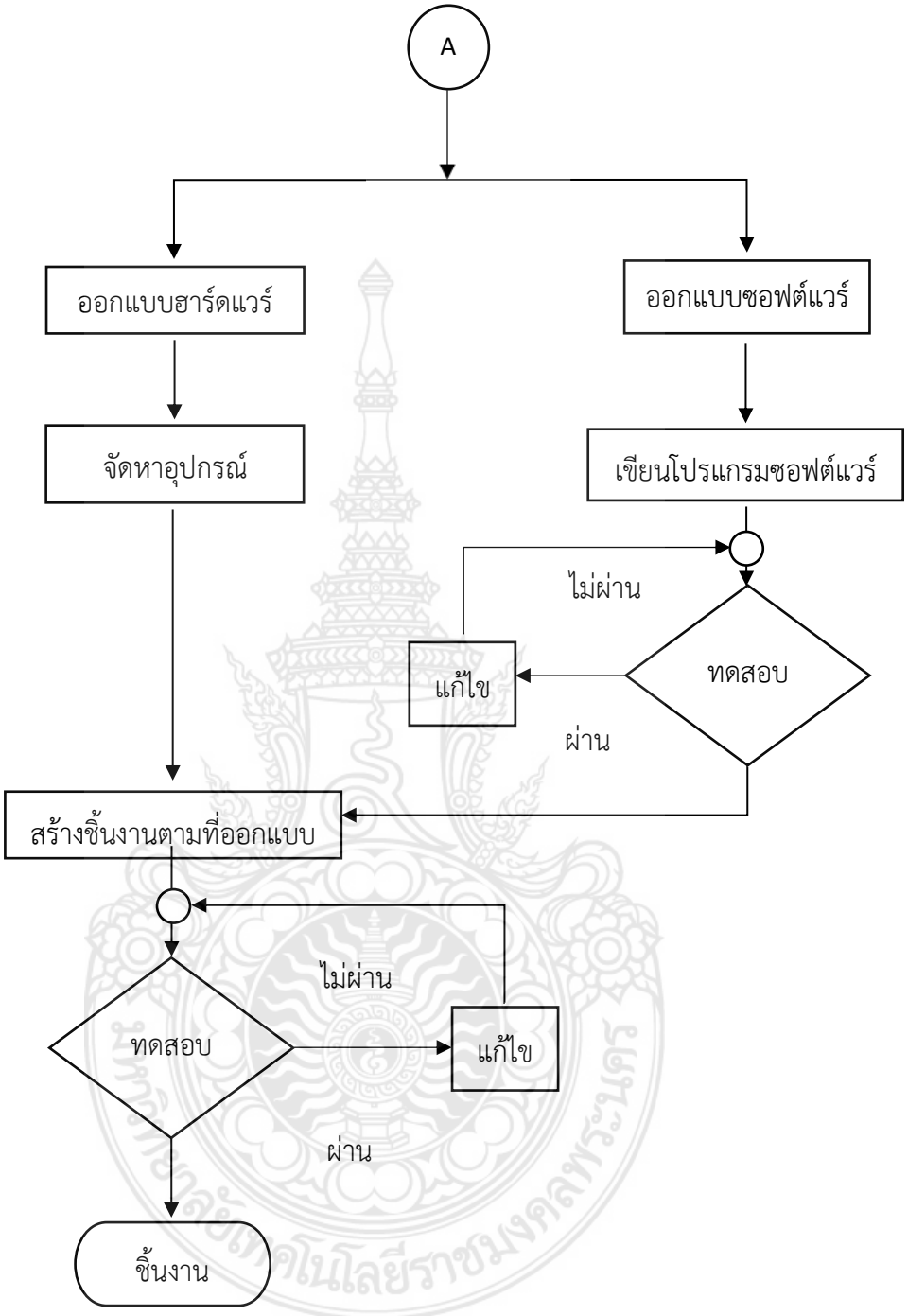
3.2 การทำงานของเครื่องคัดแยกขวดอัตโนมัติ

3.3 ขั้นตอนการประกอบชิ้นงาน

3.1 ขั้นตอนการดำเนินงาน

จากการศึกษาทฤษฎีที่เกี่ยวข้องสามารถเขียนแผนผังแสดงขั้นตอนและวิธีการดำเนินงานต่าง ๆ เป็นแผนภาพการทำงานได้ดังภาพที่ 3.1

ภาพที่ 3.1 แผนการดำเนินงาน

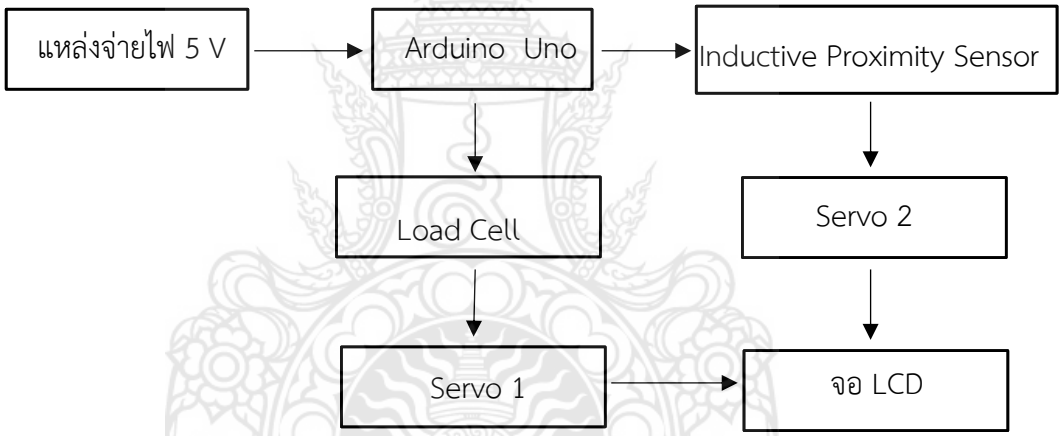


ภาพที่ 3.1 แผนการดำเนินงาน (ต่อ)

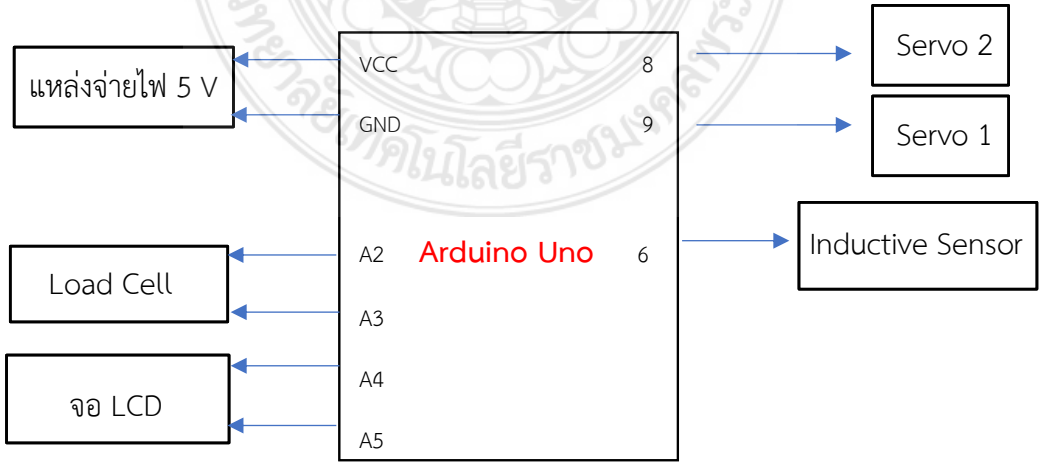
ดังภาพที่ 3.1 เป็นผังแสดงขั้นตอนในการออกแบบเครื่องคัดแยกขวดอัตโนมัติ การแยกขวดพลาสติก ขวดแก้ว และกระป๋อง โดยศึกษาหลักการทำงานของโปรแกรมและเซนเซอร์ต่างๆที่ใช้ในการชั่งน้ำหนักและการตรวจจับโลหะ รวมไปถึงหลักการทำงานของบอร์ด Arduino UNO ใช้ในการควบคุม Sensor และ Load cell ในการควบคุม Servo และนอกจากนี้ยังศึกษาการทำงานและการเขียนโปรแกรม Arduino เพื่อเป็นตัวกลางในการควบคุม และแสดงผลผ่านจอ LCD 20x4

3.2 การทำงานของเครื่องคัดแยกขวดอัตโนมัติโดยใช้โปรแกรม Arduino

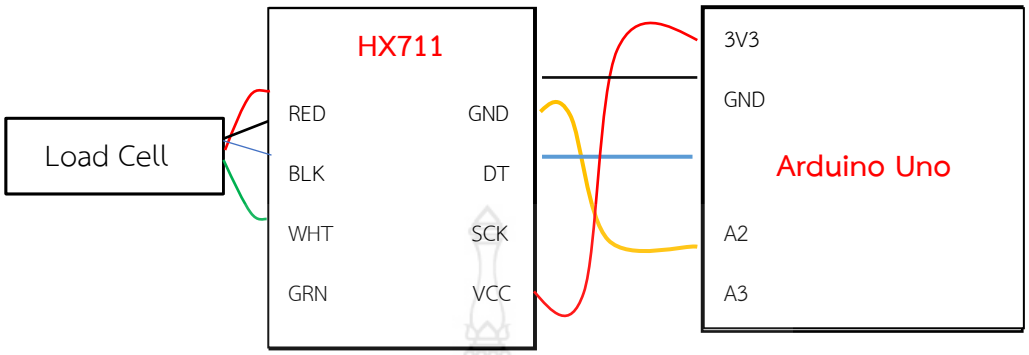
การทำงานของเครื่องคัดแยกขวดอัตโนมัติด้วยระบบไมโครคอนโทรลเลอร์ โดยจำลองเป็นบล็อกไดอะแกรมการทำงานของอุปกรณ์แต่ละตัวเริ่มต้นด้วยการจ่ายไฟเลี้ยงให้กับบอร์ดควบคุม เซนเซอร์ตรวจจับโลหะและเซนเซอร์ชั่งน้ำหนักบอร์ด Arduino UNO ในภาพซึ่งเซนเซอร์จะเป็นตัวตรวจจับโลหะและน้ำหนักส่งมายังบอร์ด Arduino UNO เป็นตัวควบคุมการทำงานทั้งหมดและมีจอแสดง LCD แสดงการนับจำนวนขวดและแบ่งเป็นแต่ละชนิด ดังภาพที่ 3.2



ภาพที่ 3.2 บล็อกไดอะแกรมของเครื่องคัดแยกขวดอัตโนมัติ



ภาพที่ 3.3 วงจรเครื่องคัดแยกขวดอัตโนมัติ



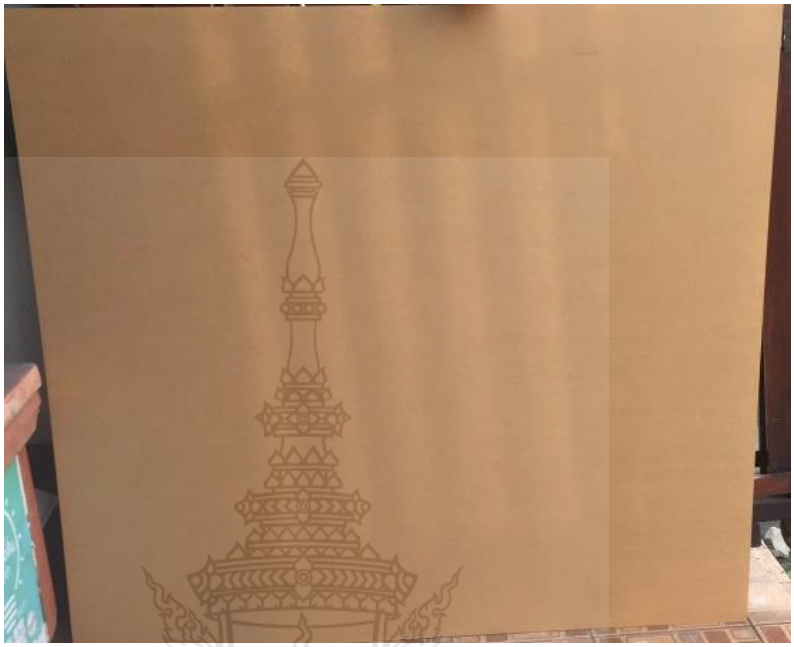
ภาพที่ 3.4 วงจร Load cell ต่อกับ Arduino

3.3 ขั้นตอนการประกอบชิ้นงาน

ขั้นตอนที่ 1 เตรียมอลูมิเนียมฉากเพื่อตัดทำโครง ดังภาพที่ 3.5

ภาพที่ 3.5 อลูมิเนียมฉาก

ขั้นตอนที่ 2 เตรียมแผ่นอะคริลิกเพื่อทำโครง ดังภาพที่ 3.6



ภาพที่ 3.6 แผ่นอะคริลิก

ขั้นตอนที่ 3 ตัดอลูมิเนียมฉากเป็น 2 ขนาด 90cm และ 70cm ดังภาพที่ 3.7

ภาพที่ 3.7 อลูมิเนียมฉาก 2 ขนาด

ขั้นตอนที่ 4 ประกอบอลูมิเนียมเข้าฉากและเจาะรู ดังภาพที่ 3.8

ภาพที่ 3.8 ประกอบอลูมิเนียมเข้าฉากและเจาะรู

ขั้นตอนที่ 5 ยิงรีเวทยึดฉาก 4 มุม ดังภาพที่ 3.9

ภาพที่ 3.9 ยิงรีเวทยึดฉาก 4 มุม

ขั้นตอนที่ 6 ยึดฉากเข้ากับฉากขั้นตอนที่ 5 ขึ้นโครง ดังภาพที่ 3.10

ภาพที่ 3.10 ขึ้นโครง

ขั้นตอนที่ 7 ตัดแผ่นอะคริลิกยึดกับโครง ดังภาพที่ 3.11

ภาพที่ 3.11 ตัดแผ่นอะคริลิกยึดกับโครง

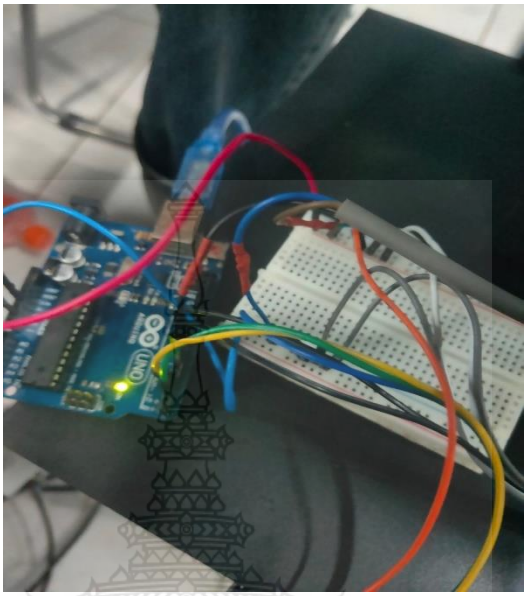
ขั้นตอนที่ 8 ติดตาข่ายพลาสติก ดังภาพที่ 3.12

ภาพที่ 3.12 ติดตาข่ายพลาสติก

ขั้นตอนที่ 8 ติดอะคริลิกเป็นประตูเข้ากับโครง ดังภาพที่ 3.13

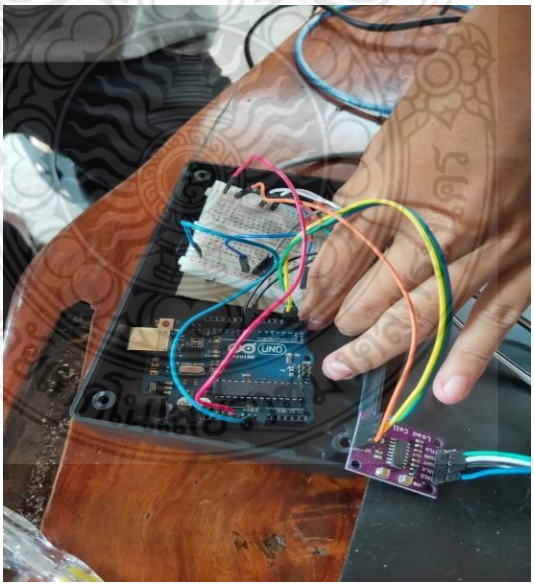
ภาพที่ 3.13 ติดแผ่นอะคริลิกเป็นประตูเข้ากับโครง

ขั้นตอนที่ 9 บัดกรีวงจรและสายไฟที่ใช้กับส่วนต่างๆภายใน ดังภาพที่ 3.14



ภาพที่ 3.14 บัดกรีวงจรและสายไฟที่ใช้กับส่วนต่างๆภายใน

ขั้นตอนที่ 10 เก็บสายให้เรียบร้อยแล้วใส่กล่อง ดังภาพที่ 3.15



ภาพที่ 3.15 เก็บสายให้เรียบร้อยแล้วใส่กล่อง

ขั้นตอนที่ 11 แสดงถึงหน้าของชิ้นงานโดยจะบอก ชื่อโครงการ ชื่อผู้จัด และจอแสดงผล LCD จะทำงานเมื่อจ่ายไฟเข้า Vin5vDc เมื่อเปิดเครื่องหน้าจอแสดงผล LCD จะบอกถึงจำนวนขวด ดังภาพที่ 3.16

ภาพที่ 3.16 จอแสดงผล LCD

มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

บทที่ 4
ผลการดำเนินงาน

ในบทนี้จะกล่าวถึงการทดลองการทำงานของระบบเครื่องคัดแยกขวดอัตโนมัติโดยใช่โปรแกรม Arduino ว่าสามารถควบคุมเครื่องคัดแยกขวดอัตโนมัติตามที่เรากำหนดได้หรือไม่และการทดสอบโครงการนี้จะมีการทดสอบตามขั้นตอนดังนี้

4.1 ขั้นตอนการทดสอบโครงการ

4.2 ภาพรวมแสดงผลการทดสอบประสิทธิภาพของระบบเครื่องคัดแยกขวดอัตโนมัติ

4.3 สรุปผลการทดสอบโครงการ

4.1. ขั้นตอนการทดสอบโครงการ

4.1.1 การทดสอบระบบเครื่องคัดแยกขวดอัตโนมัติโดยใช้โปรแกรม Arduino

แสดงค่าความถูกต้องของคำสั่งที่ใช้ทดสอบสั่งงานของระบบเครื่องคัดแยกขวดอัตโนมัติโดยใช้โปรแกรม Arduino โดยทำการทดสอบโดยแยกประเภทขวดเป็น 3 ประเภท คือ 1.ขวดพลาสติก 2.ขวดแก้ว 3.กระป๋อง เพื่อทดสอบระบบและประสิทธิภาพ ดังตารางที่ 4.1

ตารางที่ 4.1 แสดงค่าความถูกต้องของการทดสอบระบบเซนเซอร์ของระบบเครื่องคัดแยกขวดอัตโนมัติโดยใช้โปรแกรม Arduino

รอบที่ 1

ลำดับที่	การทดสอบ	ความถูกต้อง (ครั้ง)	ความผิดพลาด (ครั้ง)
1	ขวดพลาสติก	10	0
2	ขวดแก้ว	10	3
3	กระป๋อง	10	5

ตารางที่ 4.2 แสดงค่าความถูกต้องของการทดสอบระบบเซนเซอร์ของระบบเครื่องคัดแยกขวดอัตโนมัติโดยใช้โปรแกรม Arduino

รอบที่ 2

ลำดับที่	การทดสอบ	ความถูกต้อง (ครั้ง)	ความผิดพลาด (ครั้ง)
1	ขวดพลาสติก	10	0
2	ขวดแก้ว	10	2
3	กระป๋อง	10	3

ตารางที่ 4.3 แสดงค่าความถูกต้องของการทดสอบระบบเซนเซอร์ของระบบเครื่องคัดแยกขวดอัตโนมัติโดยใช้โปรแกรม Arduino

รอบที่ 3

ลำดับที่	การทดสอบ	ความถูกต้อง (ครั้ง)	ความผิดพลาด (ครั้ง)
1	ขวดพลาสติก	10	0
2	ขวดแก้ว	10	1
3	กระป๋อง	10	2

ตารางที่ 4.4 แสดงค่าความถูกต้องของการทดสอบระบบเซนเซอร์ของระบบเครื่องคัดแยกขวดอัตโนมัติโดยใช้โปรแกรม Arduino

รอบที่ 4

ลำดับที่	การทดสอบ	ความถูกต้อง (ครั้ง)	ความผิดพลาด (ครั้ง)
1	ขวดพลาสติก	10	0
2	ขวดแก้ว	10	0
3	กระป๋อง	10	1

สรุปผลการทดลอง

ค่าความถูกต้องของการทดสอบสั่งงานของระบบเครื่องคัดแยกขวดอัตโนมัติโดยใช้โปรแกรม Arduino โดยทำการทดสอบ 1วัน วันละ 4 รอบ รอบละ 10 ครั้ง เพื่อทดสอบระบบและประสิทธิภาพ โดยการทดสอบการเปิดผ่านโปรแกรม Arduino มีค่าความถูกต้องร้อย 90 และมีค่าความผิดพลาดร้อยละ 10 โดยการทดสอบการปิดผ่านโปรแกรม Arduino การทดสอบการคัดแยกขวดที่ในแต่ละรอบของการคัดแยกขวดจะมีจำนวนขวดที่แตกต่างกัน โดยแสดงดังตารางที่ 4.5

ตารางที่ 4.5 ผลรวมแสดงผลการทดสอบประสิทธิภาพในการคัดแยกขวดอัตโนมัติ

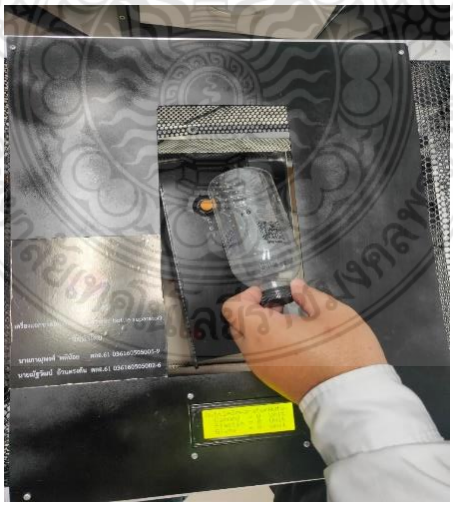
ลำดับที่	การทดสอบ	ความถูกต้อง		ความผิดพลาด	
		จำนวนที ทดสอบ 4ครั้ง	ร้อยละ	จำนวนที่ ทดสอบ 4 ครั้ง	ร้อยละ
1	การทดสอบคัดแยกขวดพลาสติก	1	90	10	10
2	การทดสอบคัดแยกขวดแก้ว	1	90	10	10
3	การทดสอบคัดแยกกระป๋อง	1	90	10	10

ค่าความถูกต้องของจำนวนขวดที่เครื่องคัดแยกขวดได้ โดยทำการทดสอบ 1วัน วันละ 4 รอบ รอบละ10 ขวดเพื่อทดสอบประสิทธิภาพในการคัดแยกขวดในแต่ละรอบ

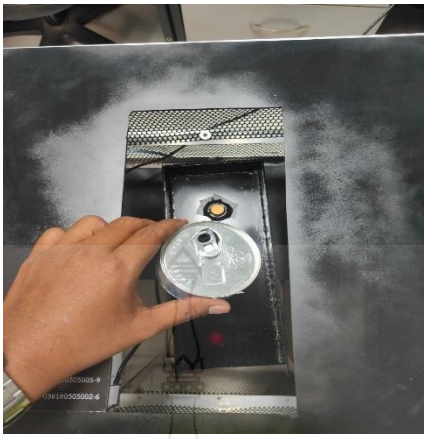
4.2 ภาพรวมแสดงผลการทดสอบประสิทธิภาพของระบบเครื่องคัดแยกขวดอัตโนมัติ

แสดงค่าความถูกต้องของคำสั่งที่ใช้ทดสอบสั่งงานของระบบเครื่องคัดแยกขวดอัตโนมัติโดยใช้โปรแกรม Arduino โดยทำการทดสอบโดยแยกประเภทขวดเป็น 3 ประเภท คือ 1.ขวดพลาสติก 2.ขวดแก้ว 3.กระป๋อง เพื่อทดสอบระบบและประสิทธิภาพ

4.2.1 ทำการทดสอบสั่งงานของระบบเครื่องคัดแยกขวดอัตโนมัติโดยใช้โปรแกรม Arduino



ภาพที่ 4.1 ใส่ขวดพลาสติกเพื่อทดสอบเซนเซอร์น้ำหนัก



ภาพที่ 4.2 ใส่กระป๋องเพื่อทดสอบเซนเซอร์ตรวจจับโลหะ

ภาพที่ 4.3 ใส่ขวดแก้วเพื่อทดสอบเซนเซอร์น้ำหนัก

ภาพที่ 4.4 หน้าจอแสดงผลจำนวนขวดแต่ละประเภท

4.2.2 การทดลองใช้งานเครื่องคัดแยกขวดอัตโนมัติตามขอบเขตที่กำหนด

การทดลองใช้งานเครื่องคัดแยกขวดอัตโนมัติตามขอบเขตที่กำหนดจากภาพ

ภาพที่ 4.1 ใส่ขวดพลาสติกเพื่อทดสอบเซนเซอร์น้ำหนัก

ภาพที่ 4.2 ใส่กระป๋องเพื่อทดสอบเซนเซอร์ตรวจจับโลหะ

ภาพที่ 4.3 ใส่ขวดแก้วเพื่อทดสอบเซนเซอร์น้ำหนัก

ภาพที่ 4.4 หน้าจอแสดงผลจำนวนขวดแต่ละประเภท

4.2.3ทำการทดสอบเพื่อหาความแม่นยำของเซนเซอร์ที่เหมาะสมเพื่อนำมาแก้ไข

ภาพที่ 4.5ใส่ขวดพลาสติกในช่องเซนเซอร์

ภาพที่ 4.6 เซนเซอร์ไม่ทำงานตามเงื่อนไข

ภาพที่ 4.7 ใส่ขวดพลาสติกในช่องเซนเซอร์

ภาพที่ 4.8 เซนเซอร์ทำงานตามเงื่อนไข

4.2.4ทำการทดสอบเพื่อหาความแม่นยำของเซนเซอร์ที่เหมาะสมเพื่อนำมาแก้ไข

การทดลองใช้งานเครื่องคัดแยกขวดอัตโนมัติตามขอบเขตที่กำหนดจากภาพ

ภาพที่ 4.5 ใส่ขวดพลาสติกในช่องเซนเซอร์

ภาพที่ 4.6 เซนเซอร์ไม่ทำงานตามเงื่อนไข

ภาพที่ 4.7 ใส่ขวดพลาสติกในช่องเซนเซอร์

ภาพที่ 4.8 เซนเซอร์ทำงานตามเงื่อนไข

4.3 สรุปผลการทดสอบโครงการ

การทดสอบสั่งงานของระบบเครื่องคัดแยกขวดอัตโนมัติโดยใช้โปรแกรม Arduino ในการแสดงผลมีความถูกต้องของการทำงานร้อยละ 90% มีค่าผิดพลาด 10% โดยค่าผิดพลาดเกิดจากสาเหตุวางภาชนะไม่ตรงกับเซนเซอร์ทำให้เซนเซอร์ทำงานผิดปกติและไม่สามารถทำงานตามคำสั่งได้ ส่วนความผิดพลาดของการควบคุม Arduino มีค่าความถูกต้องร้อยละ 100 ไม่มีค่าความผิดพลาด สามารถนำไปใช้งานได้จริง

มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

บทที่ 5
สรุปผลและข้อเสนอแนะ

ในการจัดทำโครงการเรื่องเครื่องคัดแยกขวดอัตโนมัติการทำงานสามารถสรุปผลการดำเนินงานดังนี้

5.1 สรุปผลการดำเนินงาน

5.2 ปัญหาและอุปสรรค

5.3 ข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการทำเครื่องคัดแยกขวดอัตโนมัติใช้Arduinoในการควบคุมการทำงานของเครื่องโดยไมโครคอนโทรลเลอร์ทำการเปิด-ปิด เซนเซอร์ตรวจจับวัตถุที่กำหนด โปรแกรมจะสั่งงานให้เซนเซอร์ตรวจจับวัตถุตามเงื่อนไขที่กำหนดจากนั้นเครื่องนี้ยังสามารถนำไปพัฒนาต่อยอดได้ในอนาคต

5.2 ปัญหาและอุปสรรค

5.2.1 เซนเซอร์ตรวจจับโลหะไม่ค่อยเสถียร

5.2.2 หน้าสัมผัสของเซนเซอร์ไม่ค่อยดี

5.3 ข้อเสนอแนะ

5.3.1 เครื่องคัดแยกขวดอัตโนมัติสามารถพัฒนาให้มีโหมดบีบอัดกระป๋องและขวดพลาสติก

5.3.2 เครื่องคัดแยกขวดอัตโนมัติสามารถนำไปพัฒนาต่อยอดให้สามารถนำไปใช้ในที่ต่างๆ ที่ไม่มี ไฟฟ้า ได้โดยติดตั้งแผงโซล่าเซล แบตเตอร์รี่และตัวแปลงไฟฟ้า

5.3.3 เพิ่มขนาดของช่องเก็บขวดเพื่อสะดวกต่อการเก็บขวดที่มากขึ้น

บรรณานุกรม

บุญเกิด สนธิพันธ์ พ.ศ. 2563. “โปรแกรมภาษาอาดุยโน่”.13 พฤศจิกายน 2562,
http://www.ayuttech.ac.th/2017/images/pdf/boonkoed/microcontroller_9.pdf
ราชิน กะมินสิน. “เครื่องคัดแยกขวดอัตโนมัติ”. ปริญญานิพนธ์ หลักสูตรวิศวกรรมศาสตร์ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์, สงขลา, 2557.
เด่น ดอกพิมาย. “การควบคุมนิวเมติกส์สำหรับอุตสาหกรรมอัตโนมัติ”. พิมพ์ครั้งที่ 1,กรุงเพมหานคร : สำนักพิมพ์ ท้อป จำกัด, 2552.
Pound XI. พ.ศ. 2563. “โปรแกรมภาษาอาดุยโน่”.11 พฤศจิกายน 2562,
https://poundxi.com/arduino-
Thaiesyelec. Node Mcu Esp8266. 16 มกราคม 2563,
https://www.thaieasyelec.com/development-boards/esp.

ภาคผนวก

ภาคผนวก ก
โปรแกรมการทำงานเครื่องคัดแยกขวดอัตโนมัติ

โปรแกรมการทำงานเครื่องคัดแยกขวดอัตโนมัติ

```
#include "HX711.h"
#define calibration_factor 378000 // เอาค่าที่ได้จากการปรับ มาใส่ตรงนี้
#define DOUT  3
#define CLK  2
HX711 scale;
#include <Servo.h>
Servo myservo1; //ประกาศตัวแปรแทน Servo
Servo myservo2; //ประกาศตัวแปรแทน Servo
int sensor=6;  //ใช้ขา D6
#include <Servo.h>
int valuesesor = 0;
int Glass  = 0 ;
int Plastic = 0 ;
#include <LiquidCrystal_PCF8574.h>
#include <Wire.h>
LiquidCrystal_PCF8574 lcd(0x27); // set the LCD address to 0x27 for a 16 chars and 2
line display

void setup() {
        myservo1.attach(9); // กำหนดขา 9 ควบคุม Servo
        myservo2.attach(8); // กำหนดขา 8 ควบคุม Servo
        Serial.begin(9600);
        erial.println("ArduinoAll Calibrating...");
        scale.begin(DOUT, CLK);
        scale.set_scale(calibration_factor); // ปรับค่า calibration factor
        scale.tare(); //รีเซตน้ำหนักเป็น 0

       Serial.println("OK Start :");
       while (!Serial);
       Serial.println("Dose: check for LCD");
       Wire.begin();
       Wire.beginTransmission(0x27);
       Serial.println("LCD...");
```

```
        lcd.begin(20,4); // initialize the lcd
        lcd.print("Hello LCD");
        lcd.clear();
        lcd.setBacklight(255);
        lcd.setCursor(0, 0); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่0 แถวที่ 0 เตรียมพิมพ์ข้อความ
        lcd.print("BottleSeparatorAuto. ");
        cd.setCursor(2, 1); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่2 แถวที่ 1 เตรียมพิมพ์ข้อความ
        lcd.print("Canned  = ");
        lcd.setCursor(2, 2); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่2 แถวที่ 2 เตรียมพิมพ์ข้อความ
        lcd.print("Plastic = ");
        lcd.setCursor(2, 3); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่2 แถวที่ 3 เตรียมพิมพ์ข้อความ
        lcd.print("Glass   = ");
        lcd.setCursor(15, 1); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่ 15 แถวที่ 1 เตรียมพิมพ์ข้อความ
        lcd.print("Unit");
        lcd.setCursor(15, 2); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่ 15 แถวที่ 2 เตรียมพิมพ์ข้อความ
        lcd.print("Unit");
        lcd.setCursor(15, 3); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่ 15 แถวที่ 3 เตรียมพิมพ์ข้อความ
        lcd.print("Unit");
}

void loop() {
        Serial.print("Reading: ");
        Serial.print(scale.get_units(), 3); //แสดงผลน้ำหนัก 3 ตำแหน่ง
        Serial.println(" kg");
        delay (500);
        int x = digitalRead(sensor);
        Serial.println(x);
        delay (100);
```

```
if (x==1){
  myservo2.write(90); // สั่งให้ Servo หมุนไปองศาที่ 90
  delay(1000); // หน่วงเวลา 1000ms
 }
if (x==0){
   x = valuesesor ;
   valuesesor ++ ;

myservo2.write(135); // สั่งให้ Servo หมุนไปองศาที่ 135
delay(1000); // หน่วงเวลา 1000ms
}
if (scale.get_units()>= 0.041) {
  Plastic ++ ;
  myservo1.write(180); // สั่งให้ Servo หมุนไปองศาที่ 180
  delay(1000); // หน่วงเวลา 1000ms
  myservo2.write(135); // สั่งให้ Servo หมุนไปองศาที่ 135
  delay(1000); // หน่วงเวลา 1000ms
  delay(2000);
  myservo2.write(90); // สั่งให้ Servo หมุนไปองศาที่ 90
  delay(1000); // หน่วงเวลา 1000ms
  myservo1.write(90); // สั่งให้ Servo หมุนไปองศาที่ 90
  delay(1000); // หน่วงเวลา 1000ms
  }

else if ((scale.get_units()> 0.005) and (scale.get_units()< 0.040)) {
  Glass ++;
  myservo1.write(0); // สั่งให้ Servo หมุนไปองศาที่ 0
  delay(1000); // หน่วงเวลา 1000ms
  myservo2.write(135); // สั่งให้ Servo หมุนไปองศาที่ 135
  delay(1000); // หน่วงเวลา 1000ms
  delay(2000);
  myservo2.write(90); // สั่งให้ Servo หมุนไปองศาที่ 90
  delay(1000); // หน่วงเวลา 1000ms
  myservo1.write(90); // สั่งให้ Servo หมุนไปองศาที่ 90
  delay(1000); // หน่วงเวลา 1000ms
}
```

```
        lcd.setCursor(2, 1); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่2 แถวที่ 1 เตรียมพิมพ์ข้อความ
        lcd.print("Canned  = ");
        lcd.setCursor(2, 2); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่2 แถวที่ 2 เตรียมพิมพ์ข้อความ
        lcd.print("Plastic = ");
        lcd.setCursor(2, 3); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่2 แถวที่ 3 เตรียมพิมพ์ข้อความ
        lcd.print("Glass   = ");
        lcd.setCursor(15, 1); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่ 15 แถวที่ 1 เตรียมพิมพ์ข้อความ
        lcd.print("Unit");
        lcd.setCursor(15, 2); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่ 15 แถวที่ 2 เตรียมพิมพ์ข้อความ
        lcd.print("Unit");
        lcd.setCursor(15, 3); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่ 15 แถวที่ 3 เตรียมพิมพ์ข้อความ
        lcd.print("Unit");

        lcd.setCursor(12, 1); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่ 12 แถวที่ 1 เตรียมพิมพ์ข้อความ

        lcd.print(Canned);
        Serial.print("Canned"); Serial.println(Canned);
        lcd.setCursor(12, 3); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่ 12 แถวที่ 3 เตรียมพิมพ์ข้อความ
        lcd.print(Plastic);
        Serial.print("Plastic");Serial.println(Plastic);
        lcd.setCursor(12, 2); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่ 12 แถวที่ 2 เตรียมพิมพ์ข้อความ
        lcd.print(Glass);
        Serial.print("Glass");Serial.println(Glass);
        lcd.setCursor(0, 0); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่0 แถวที่ 0 เตรียมพิมพ์ข้อความ
            lcd.print("BottleSeparatorAuto. ");
}
```

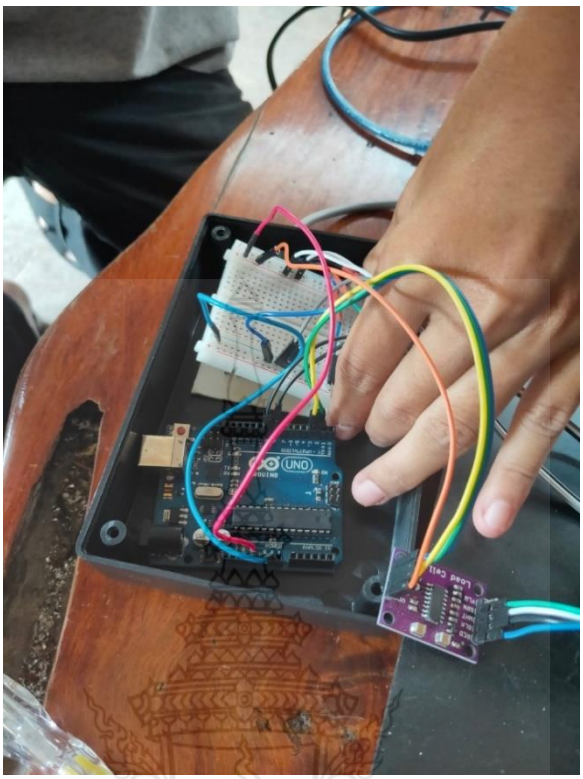
ภาคผนวก ข
โครงสร้างและการต่อวงจร

ภาพ ข.1 โครงสร้างเครื่องคัดแยกขวดอัตโนมัติ

ภาพ ข.2 โครงสร้างเครื่องคัดแยกขวดอัตโนมัติ(ต่อ)

ภาพ ข.3 โครงสร้างเครื่องคัดแยกขวดอัตโนมัติ (ต่อ)

ภาพ ข.4 โครงสร้างเครื่องคัดแยกขวดอัตโนมัติ (ต่อ)



ภาพ ข.5 วงจรเครื่องคัดแยกขวดอัตโนมัติ

ภาพ ข.6 วงจรเครื่องคัดแยกขวดอัตโนมัติ(ต่อ)

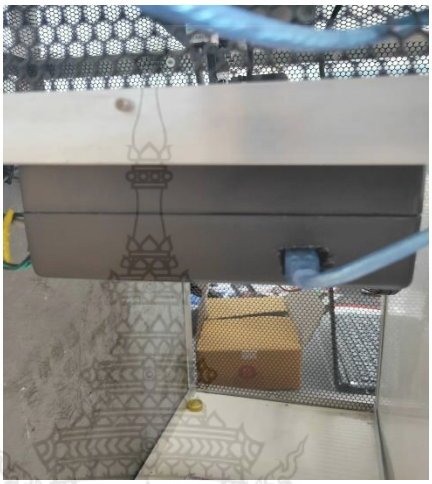
ภาคผนวก ค
คู่มือการใช้งาน

คู่มือการใช้งานเครื่องคัดแยกขวดอัตโนมัติ

รายละเอียดเครื่องคัดแยกขวดอัตโนมัติ

เครื่องคัดแยกขวดอัตโนมัติ ใช้ไฟเลี้ยงวงจรควบคุม Arduino (DC 12 V)

1. จ่ายไฟให้เครื่องคัดแยกขวดอัตโนมัติ เลือกช่องปลั๊กใช้งานกับอุปกรณ์ที่จะควบคุม



ภาพ ค.1 ช่องปลั๊กควบคุมอุปกรณ์ต่างๆ

ภาพ ค.2 ช่องปลั๊กควบคุมอุปกรณ์ต่างๆ (ต่อ)

เซนเซอร์ตรวจจับโลหะ (DC 5 V) ใช้ตรวจจับโลหะเช่น กระป๋องน้ำ

Arduino ช่องปลั๊ก (DC 12 V) ใช้ควบคุมเซนเซอร์และเซอร์โวมอเตอร์

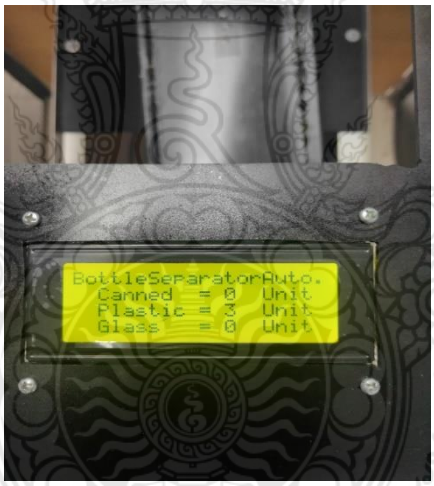
เซนเซอร์น้ำหนัก (DC 5 V) ใช้ตรวจจับขวดพลาสติกและขวดแก้ว

เซอร์โวมอเตอร์ (DC 5 V) ใช้ควบคุมเซนเซอร์

USB ช่องเสียบเพื่ออัพโหลดโปรแกรมการทำงาน

2. ทำการนำขวดต่างๆใส่ช่องเพื่อคัดแยกขวดตามข้อกำหนด

ภาพ ค.3 นำขวดใส่ช่องคัดแยกขวดอัตโนมัติ



ภาพ ค.4 หน้าจอแสดงผลจำนวนขวดแต่ละประเภท

หน้าจอการควบคุมเครื่องคัดแยกขวดอัตโนมัติด้วย Arduino จะแสดงผลการทำงานของเครื่องคัดแยกขวดอัตโนมัติและบอกจำนวนของขวดที่คัดแยก

การแก้ไขปัญหาเบื้องต้นกรณีเครื่องไม่ทำงาน

1. ไม่มีไฟเข้าเครื่องคัดแยกขวดอัตโนมัติ
   - ตรวจสอบแหล่งจ่ายไฟว่ามีไฟเข้ามาหรือไม่
   - ตรวจสอบสายไฟ
2. กรณีเซนเซอร์ตรวจจับโลหะไม่ตรวจจับภาชนะที่ต้องการ
   - ตรวจสอบอุปกรณ์เซนเซอร์ว่าเสียบแน่นหรือไม่

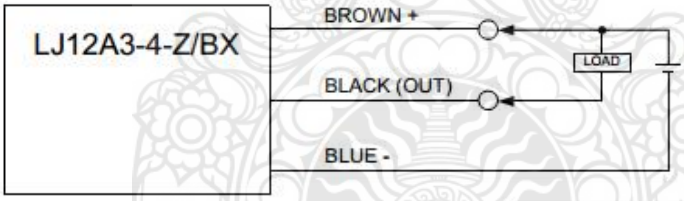
ภาคผนวก ง

Data Sheet

LJ12A3-4-Z/BX (A-RF-C-00014) * 140.-

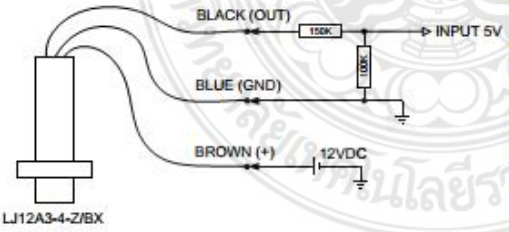
INDUCTIVE PROXIMITY SENSOR LJ12A3-4-Z/BX
NPN, 3 WIRE ON, DIAMETER 12 mm.,
PROXIMITY SW. (ใช้ได้กับโลหะเท่านั้น ในการตรวจจับ),
ระยะตรวจจับ 4 mm.,
พร้อม LED สีแดง แสดงผลสถานะการตรวจจับ

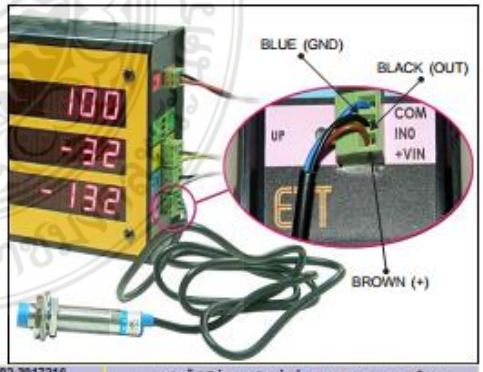
MODEL	LJ12A3-4-Z/BX
OUTPUT	NPN, 3 WIRE, NO (NORMAL OPEN)
WORK VOLTAGE	6VDC - 36VDC
OUTPUT CURRENT	300mA. (SINK)
SIZE	DIA. 12 mm. L : 64 mm.(BODY)
CABLE LENGTH	100 cm.
DETECTION RANGE	4 mm.
DETECTION OF BOJECT	METAL (COPPER, IRON, ALUMINUIM ETC.)



NPN
NO NORMALLY OPEN
4 mm.

- การต่อกับระบบ 5V TTL ในกรณีใช้ไฟเลี้ยง SENSOR 12VDC





ETT บริษัท อีทีที จำกัด ETT CO., LTD.
Tel : 02-7121120
e-mail : sale@etteam.com
www.etteam.com
Fax : 02-3917216
Line ID : etteam1
* ราคาสินค้ายังไม่รวมภาษีมูลค่าเพิ่ม 7% บริษัทขอสงวนสิทธิ์ในการเปลี่ยนแปลงราคาและเงื่อนไขโดยไม่จำเป็นต้องแจ้งให้ทราบล่วงหน้า

Arduino Uno

Arduino Uno R3 Front

Arduino Uno R3 Back

Arduino Uno R2 Front

Arduino Uno SMD

Arduino Uno Front

Arduino Uno Back

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.
Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.
Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the Index of Arduino boards.

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer)
Schematic: arduino-uno-Rev3-schematic.pdf
Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.
External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.
The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.
The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).
A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.
The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.
The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.
The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

Datasheet

3133 - Micro Load Cell (0-5kg) - CZL635

Contents

1 What do you have to know?
1 How does it work - For curious people
1 Installation
2 Calibration
2 Product Specifications
3 Glossary

What do you have to know?

A load cell is a force sensing module - a carefully designed metal structure, with small elements called strain gauges mounted in precise locations on the structure. Load cells are designed to measure a specific force, and ignore other forces being applied. The electrical signal output by the load cell is very small and requires specialized amplification. Fortunately, **the 1046 PhidgetBridge will perform all the amplification and measurement of the electrical output.**

Load cells are designed to measure force in one direction. They will often measure force in other directions, but the sensor sensitivity will be different, since parts of the load cell operating under compression are now in tension, and vice versa.

How does it work - For curious people

Strain-gauge load cells convert the load acting on them into electrical signals. The measuring is done with very small resistor patterns called strain gauges - effectively small, flexible circuit boards. The gauges are bonded onto a beam or structural member that deforms when weight is applied, in turn deforming the strain-gauge. As the strain gauge is deformed, it's electrical resistance changes in proportion to the load.

The changes to the circuit caused by force is much smaller than the changes caused by variation in temperature. Higher quality load cells cancel out the effects of temperature using two techniques. By matching the expansion rate of the strain gauge to the expansion rate of the metal it's mounted on, undue strain on the gauges can be avoided as the load cell warms up and cools down. The most important method of temperature compensation involves using multiple strain gauges, which all respond to the change in temperature with the same change in resistance. Some load cell designs use gauges which are never subjected to any force, but only serve to counterbalance the temperature effects on the gauges that measuring force. Most designs use 4 strain gauges, some in compression, some under tension, which maximizes the sensitivity of the load cell, and automatically cancels the effect of temperature.

Installation

This Single Point Load Cell is used in small jewelry scales and kitchen scales. It's mounted by bolting down the end of the load cell where the wires are attached, and applying force on the other end **in the direction of the arrow.** Where the force is applied is not critical, as this load cell measures a shearing effect on the beam, not the bending of the beam. If you mount a small platform on the load cell, as would be done in a small scale, this load cell provides accurate readings regardless of the position of the load on the platform.

Calibration

A simple formula is usually used to convert the measured mv/V output from the load cell to the measured force:

$$Measured\ Force = A * Measured\ mV/V + B\ (\text{offset})$$

It's important to decide what unit your measured force is - grams, kilograms, pounds, etc.

This load cell has a rated output of 1.0±0.15mv/v which corresponds to the sensor's capacity of 5kg.

To find A we use

$$Capacity = A * Rated\ Output$$

$$A = Capacity / Rated\ Output$$

$$A = 5 / 1.0$$

$$A = 5$$

Since the Offset is quite variable between individual load cells, it's necessary to calculate the offset for each sensor. Measure the output of the load cell with no force on it and note the mv/V output measured by the PhidgetBridge.

$$\text{Offset} = 0 - 5 * \text{Measured Output}$$

Product Specifications

Mechanical	
Housing Material	Aluminum Alloy
Load Cell Type	Strain Gauge
Capacity	5kg
Dimensions	55.25x12.7x12.7mm
Mounting Holes	M5 (Screw Size)
Cable Length	550mm
Cable Size	30 AWG (0.2mm)
Cable - no. of leads	4
Electrical	
Precision	0.05%
Rated Output	1.0±0.15 mv/V
Non-Linearity	0.05% FS
Hysteresis	0.05% FS
Non-Repeatability	0.05% FS
Creep (per 30 minutes)	0.1% FS
Temperature Effect on Zero (per 10°C)	0.05% FS
Temperature Effect on Span (per 10°C)	0.05% FS
Zero Balance	±1.5% FS
Input Impedance	1130±10 Ohm
Output Impedance	1000±10 Ohm
Insulation Resistance (Under 50VDC)	≥5000 MOhm
Excitation Voltage	5 VDC
Compensated Temperature Range	-10 to ~+40°C
Operating Temperature Range	-20 to ~+55°C
Safe Overload	120% Capacity
Ultimate Overload	150% Capacity

Glossary

Capacity
The maximum load the load cell is designed to measure within its specifications.

Creep
The change in sensor output occurring over 30 minutes, while under load at or near capacity and with all environmental conditions and other variables remaining constant.

FULL SCALE or FS
Used to qualify error - FULL SCALE is the change in output when the sensor is fully loaded. If a particular error (for example, Non-Linearity) is expressed as 0.1% F.S., and the output is 1.0mV/V, the maximum non-linearity that will be seen over the operating range of the sensor will be 0.001 mV/V. An important distinction is that this error doesn't have to only occur at the maximum load. If you are operating the sensor at a maximum of 10% of capacity, for this example, the non-linearity would still be 0.001mV/V, or 1% of the operating range that you are actually using.

Hysteresis
If a force equal to 50% of capacity is applied to a load cell which has been at no load, a given output will be measured. The same load cell is at full capacity, and some of the force is removed, resulting in the load cell operating at 50% capacity. The difference in output between the two test scenarios is called hysteresis.

Excitation Voltage
Specifies the voltage that can be applied to the power/ground terminals on the load cell. In practice, if you are using the load cell with the PhidgetBridge, you don't have to worry about this spec.

Input Impedance
Determines the power that will be consumed by the load cell. The lower this number is, the more current will be required, and the more heating will occur when the load cell is powered. In very noisy environments, a lower input impedance will reduce the effect of Electromagnetic interference on long wires between the load cell and PhidgetBridge.

Insulation Resistance
The electrical resistance measured between the metal structure of the load cell, and the wiring. The practical result of this is the metal structure of the load cells should not be energized with a voltage, particularly higher voltages, as it can arc into the PhidgetBridge. Commonly the load cell and the metal framework it is part of will be grounded to earth or to your system ground.

Maximum Overload
The maximum load which can be applied without producing a structural failure.

Non-Linearity
Ideally, the output of the sensor will be perfectly linear, and a simple 2-point calibration will exactly describe the behaviour of the sensor at other loads. In practice, the sensor is not perfect, and Non-linearity describes the maximum deviation from the linear curve. Theoretically, if a more complex calibration is used, some of the non-linearity can be calibrated out, but this will require a very high accuracy calibration with multiple points.

Non-Repeatability
The maximum difference the sensor will report when exactly the same weight is applied, at the same temperature, over multiple test runs.

Operating Temperature
The extremes of ambient temperature within which the load cell will operate without permanent adverse change to any of its performance characteristics.

Output Impedance
Roughly corresponds to the input impedance. If the Output Impedance is very high, measuring the bridge will distort the results. The PhidgetBridge carefully buffers the signals coming from the load cell, so in practice this is not a concern.

Rated Output
Is the difference in the output of the sensor between when it is fully loaded to its rated capacity, and when it's unloaded. Effectively, it's how sensitive the sensor is, and corresponds to the gain calculated when calibrating the sensor. More expensive sensors have an exact rated output based on an individual calibration done at the factory.

Safe Overload
The maximum axial load which can be applied without producing a permanent shift in performance characteristics beyond those specified.

Compensated Temperature
The range of temperature over which the load cell is compensated to maintain output and zero balance within specified limits.

Temperature Effect on Span
Span is also called rated output. This value is the change in output due to a change in ambient temperature. It is measured over 10 degree C temperature interval.

Temperature Effect on Zero
The change in zero balance due to a change in ambient temperature. This value is measured over 10 degree C temperature interval.

Zero Balance
Zero Balance defines the maximum difference between the +/- output wires when no load is applied. Realistically, each sensor will be individually calibrated, at least for the output when no load is applied. Zero Balance is more of a concern if the load cell is being interfaced to an amplification circuit - the PhidgetBridge can easily handle enormous differences between +/-. If the difference is very large, the PhidgetBridge will not be able to use the higher Gain settings.

S3003 FUTABA SERVO

...S3003 FUTABA SERVO...

Detailed Specifications

Control System:	+Pulse Width Control 1520usec Neutral	Current Drain (4.8V):	7.2mA/idle
Required Pulse:	3-5 Volt Peak to Peak Square Wave	Current Drain (6.0V):	8mA/idle
Operating Voltage:	4.8-6.0 Volts	Direction:	Counter Clockwise/Pulse Traveling 1520-1900usec
Operating Temperature Range:	-20 to +60 Degree C	Motor Type:	3 Pole Ferrite
Operating Speed (4.8V):	0.23sec/60 degrees at no load	Potentiometer Drive:	Indirect Drive
Operating Speed (6.0V):	0.19sec/60 degrees at no load	Bearing Type:	Plastic Bearing
Stall Torque (4.8V):	44 oz/in. (3.2kg.cm)	Gear Type:	All Nylon Gears
Stall Torque (6.0V):	56.8 oz/in. (4.1kg.cm)	Connector Wire Length:	12"
Operating Angle:	45 Deg. one side pulse traveling 400usec	Dimensions:	1.6" x 0.8"x 1.4" (41 x 20 x 36mm)
360 Modifiable:	Yes	Weight:	1.3oz. (37.2g)

ประวัติผู้เขียน

ประวัติผู้เขียน

ชื่อ-สกุล	นายณัฐวัฒน์ อ้วนตรงต้น
วัน-เดือน-ปีเกิด	14 มกราคม 2541
ที่อยู่ปัจจุบัน	1/43 หมู่ที่ 5 ตำบลบางเมือง อำเภอเมือง จังหวัดสมุทรปราการ
เบอร์โทรศัพท์	099-7656922
ประวัติการศึกษา	
พ.ศ. 2558	ประกาศนียบัตรวิชาชีพ แผนกช่างอิเล็กทรอนิกส์ วิทยาลัยเทคนิคสมุทรปราการ จังหวัดสมุทรปราการ
พ.ศ. 2560	ประกาศนียบัตรวิชาชีพชั้นสูง แผนกช่างอิเล็กทรอนิกส์ สาขาวิชาอิเล็กทรอนิกส์อุตสาหกรรม วิทยาลัยเทคนิคสมุทรปราการ จังหวัดสมุทรปราการ
พ.ศ. 2561	ปริญญาตรี อุตสาหกรรมศาสตรบัณฑิต (อสบ.) สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร ศูนย์เทเวศร์

ประวัติผู้เขียน

ชื่อ-สกุล	นายภาณุพงศ์ พลีน้อย
วัน-เดือน-ปีเกิด	5 กุมภาพันธ์ 2541
ที่อยู่ปัจจุบัน	85/1 หมู่ที่ 13 ตำบลบางเสาธง อำเภอบางเสาธง จังหวัดสมุทรปราการ
เบอร์โทรศัพท์	096-6210861
ประวัติการศึกษา	
พ.ศ. 2558	ประกาศนียบัตรวิชาชีพ แผนกช่างอิเล็กทรอนิกส์ วิทยาลัยเทคนิคกาญจนาภิเษกสมุทรปราการ จังหวัดสมุทรปราการ
พ.ศ. 2560	ประกาศนียบัตรวิชาชีพชั้นสูง แผนกช่างอิเล็กทรอนิกส์ สาขาวิชาอิเล็กทรอนิกส์อุตสาหกรรม วิทยาลัยเทคนิคสมุทรปราการ จังหวัดสมุทรปราการ
พ.ศ. 2561	ปริญญาตรี อุตสาหกรรมศาสตรบัณฑิต (อสบ.) สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร ศูนย์เทเวศร์