

เครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์
COIN COUNTING MACHINE USING MICROCONTROLLER

โดย
นางสาวจริยา นิ่มนวล
นางสาวทิพวรรณ สวัสดิ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์
COIN COUNTING MACHINE USING MICROCONTROLLER

โดย

นางสาวจรรยา

นางสาวทิพวรรณ

นิมมวล

สวัสดี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์
COIN COUNTING MACHINE USING MICROCONTROLLER

	โดย	
นางสาวจรรยา	นิมมวล	56010142
นางสาวทิพวรรณ	สวัสดิ์	56010487

อาจารย์ที่ปรึกษา
ผศ.ดร. มนตรี คำเงิน
รศ.ดร. พิพัฒน์ พรหมมี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

ปริญญาานิพนธ์ปีการศึกษา 2559

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์

COIN COUNTING MACHINE USING MICROCONTROLLER

ผู้จัดทำ

1 นางสาวจริยา นิ่มนวล 56010142

2 นางสาวทิพวรรณ สวัสดิ์ 56010487


..... อาจารย์ที่ปรึกษา
(ผศ.ดร.มนตรี คำเงิน)


..... อาจารย์ที่ปรึกษาร่วม
(รศ.ดร.พิพัฒน์ พรหมมี)

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ดำเนินงานสำเร็จลุล่วงได้เป็นอย่างดีโดยได้รับความร่วมมือและความช่วยเหลือจากหลายท่านในการให้คำแนะนำแนวทางและวิธีการแก้ไขปัญหา

คณะผู้จัดทำขอกราบขอบพระคุณ รศ.ดร.พิพัฒน์ พรหมมี และ ผศ.ดร.มนตรี คำเงิน อาจารย์ที่ปรึกษา ที่ให้คำปรึกษาและแนะนำทางด้านทฤษฎีและด้านปฏิบัติพร้อมทั้งยังให้ความอนุเคราะห์ในด้านเครื่องมือและอุปกรณ์ที่ใช้ในการทดสอบ ตลอดจนชี้แนะแนวทางในการแก้ไขปัญหาดังกล่าวที่เกิดขึ้นในการทำปริญญานิพนธ์ครั้งนี้ได้เป็นอย่างดีมาโดยตลอด

คณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้และพี่เพื่อนนักศึกษาทุกท่านที่คอยช่วยเหลือให้คำปรึกษาและแนะนำในหลายๆด้าน คณะผู้จัดทำขอขอบพระคุณมา ณ ที่นี้

นางสาวจรรยา นิ่มนวล
นางสาวทิพวรรณ สวัสดิ์
ผู้จัดทำ

เครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์
COIN COUNTING MACHINES USING
MICROCONTROLLER

โดย นางสาวจริยา นิมนวล 56010142
นางสาวทิพวรรณ สวัสดิ์ 56010487

อาจารย์ที่ปรึกษา ผศ.ดร.มนตรี คำเงิน
รศ.ดร.พิพัฒน์ พรหมมี

บทคัดย่อ

ปริญญาานิพนธ์ฉบับนี้ได้จัดทำเครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์วัตถุประสงค์เพื่อใช้ร่วมกับระบบขายของอัตโนมัติแบบหยอดเหรียญต่างๆ ในท้องตลาด และศึกษาการทำงานของไมโครคอนโทรลเลอร์ พร้อมกับการทำงานร่วมกันกับ เซ็นเซอร์ชนิดต่างๆ โดยเครื่องนับเหรียญที่ออกแบบนั้น สามารถแยกชนิดของ เหรียญบาท, เหรียญสองบาท เหรียญห้าบาท และเหรียญสิบบาท และ คำนวณจำนวนเงินของเหรียญที่ได้ทำการนับทั้งหมด ซึ่งเครื่องนับเหรียญแบ่งการทำงานเป็น 2 ส่วน คือ ส่วนของกลไกการแยกชนิดของเหรียญ โดยทำงานร่วมกับมอเตอร์ และส่วนของการนับจำนวนเงินของเหรียญจะใช้เซนเซอร์แสง เพื่อนับจำนวนเหรียญแต่ละชนิด โดยเครื่องนับเหรียญที่สร้างขึ้นนี้ใช้ ARDUINO ทำหน้าที่ประมวลผล และควบคุมการทำงานของเซนเซอร์ต่างๆ และทำการแสดงผลผ่านทางจอ LCD

ABSTRACT

This project introduces a coin counting machine using microcontroller. The purposes of this project are co-operated with the commercial available vending machine and studying the microcontroller operation and its interface as well as different sensors. The coin counting machine can be classified the types of coins, one Baht coins, two Bath coins, five Baht coins, ten Baht coins and counting all of them. This machine can be divided for three parts, coin classification, coin sensors and software. First part, the coin classification based on motor and mechanism to spin the coins into the relative slot. Second part, the coin sensors use numerous

serves the processing, control, calculating functions and displays the result through the LCD.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	IX
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	4
2.1 ไมโครคอนโทรลเลอร์ AVR	4
2.2 Arduino	5
2.3 Infrared Sensor Optocoupler Module (เซ็นเซอร์ก้ามปู)	9
2.4 รังสีอินฟราเรด (Infrared)	11
2.5 Liquid Crystal Display (LCD)	13
2.6 โปรแกรม Solidworks	18
2.7 มอเตอร์ไฟฟ้ากระแสตรง (Direct Current Motor)	23
2.8 สวิตชิงเพาเวอร์ซัพพลาย (Output Switching Power Supply)	25
2.9 โมดูลแปลงไฟ 6-24V เป็น 5V แบบ USB	27
2.10 โพรโทบอร์ด (Protoboard)	29
2.11 สายไฟจัมป์เปอร์	29
บทที่ 3	
การออกแบบและการจัดทำปริญญานิพนธ์	31
3.1 การออกแบบ	32
3.2 เครื่องมือที่ใช้ในการทดลอง	46
3.3 การจัดเก็บผลการทดลอง	46

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการทดลอง	49
4.1 ผลการทดลองส่วนของซอฟต์แวร์	49
4.2 ผลการทดลองส่วนของฮาร์ดแวร์	57
บทที่ 5 สรุปผลและข้อเสนอแนะ	58
5.1 สรุปผล	58
5.2 ข้อเสนอแนะ	58
บรรณานุกรม	59
ภาคผนวก ก CODE ภาษา C	60
ภาคผนวก ข DATASHEET	67

สารบัญรูป

รูปที่	หน้า	
1.1	บล็อกไดอะแกรมรวมของเครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์	3
2.1	ส่วนประกอบของ ARDUINO BOARD รุ่น UNO R3 [2]	6
2.2	ARDUINO UNO R3 BOARD [2]	7
2.3	INFRARED SENSOR OPTOCOUPLER MODULE [3]	10
2.4	INFRARED SENSOR OPTOCOUPLER MODULE กับ ARDUINO	11
2.5	ELECTROMAGNETIC SPECTRUM [4]	12
2.6	จอแสดงผล LCD [5]	13
2.7	การรับ-ส่งข้อมูลแบบ I2C BUS [5]	15
2.8	การเชื่อมต่อ ARDUINO กับจอ LCD [5]	16
2.9	ไฟล์ LIQUIDCRYSTAL_I2C.7Z [5]	16
2.10	โปรแกรม SOLIDWORKS [6]	18
2.11	หน้าต่างโปรแกรม SOLIDWORKS (1) [6]	19
2.12	โปรแกรม SOLIDWORKS (2) [6]	20
2.13	หน้าต่างโปรแกรม SOLIDWORKS (3) [6]	21
2.14	ทูลบาร์มาตรฐานของโปรแกรม SOLIDWORKS [6]	21
2.15	ชิ้นงานจากการออกแบบด้วย SOLIDWORKS [6]	23
2.16	มอเตอร์ไฟฟ้ากระแสตรง [7]	25
2.17	วงจรสวิตชิงเพาเวอร์ซัพพลาย [8]	25
2.18	แผนผังสวิตชิงเพาเวอร์ซัพพลาย [8]	26
2.19	โมดูลแปลงไฟ 5V	27
2.20	โพโรโทบอร์ด [8]	29
2.21	สายไฟจัมพ์เปอร์สำเร็จรูป [11]	30
3.1	บล็อกไดอะแกรมรวมของเครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์	32
3.2	การออกแบบส่วนการแยกเหรียญแบบ 2 มิติ	33
3.3	การออกแบบส่วนการแยกเหรียญแบบ 3 มิติ	34
3.4	การออกแบบส่วนการแยกเหรียญแบบ 3 มิติ แบบมีขอบกั้นเหรียญ	34
3.5	การออกแบบส่วนการแยกเหรียญทั้ง 4 ชนิด คือ เหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท	35
3.6	ชิ้นส่วนแบบจำลองอุปกรณ์การแยกเหรียญในส่วนของงานหมุน (1)	36

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.5	35
การออกแบบส่วนการแยกเหรียญทั้ง 4 ชนิด คือ เหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท	
3.6	36
ชิ้นส่วนแบบจำลองอุปกรณ์การแยกเหรียญในส่วนของจานหมุน (1)	
3.7	36
ชิ้นส่วนแบบจำลองอุปกรณ์การแยกเหรียญในส่วนของจานหมุน (2)	
3.8	37
ส่วนแสดงผลของเครื่องนับเหรียญ	
3.9	37
ส่วนการทำงานของเครื่องนับเหรียญ	
3.10	38
ส่วนการประมวลผลของเครื่องนับเหรียญ และส่วนของการแยกเหรียญ	
3.11	38
คำสั่งในการกำหนดค่าตัวแปรเริ่มต้น	
3.12	39
คำสั่งกำหนดค่าเริ่มต้นของการแสดงผล	
3.13	39
คำสั่งกำหนดขาอินพุตของ arduino	
3.14	40
คำสั่งในการทำงานเมื่อเซ็นเซอร์นับเหรียญหนึ่งบาท แล้วคำนวณค่า โดยจำนวนเหรียญคูณกับ 1	
3.15	41
คำสั่งในการทำงานเมื่อเซ็นเซอร์นับเหรียญสองบาท แล้วคำนวณค่า โดยจำนวนเหรียญคูณกับ 2	
3.16	42
คำสั่งในการทำงานเมื่อเซ็นเซอร์นับเหรียญห้าบาท แล้วคำนวณค่า โดยจำนวนเหรียญคูณกับ 5	
3.17	43
คำสั่งในการทำงานเมื่อเซ็นเซอร์นับเหรียญสิบบาท แล้วคำนวณค่าโดย จำนวนเหรียญคูณกับ 10	
3.18	44
คำสั่งในการกำหนดดีเลย์ของการทำงาน	
3.19	44
คำสั่งการ Reset การนับ	
3.20	45
Flowchart การทำงานของเครื่องนับเหรียญ	
3.21	46
การทำงานของเซ็นเซอร์กับ Arduino	
3.22	47
ทดสอบการทำงานของเซ็นเซอร์กับ Arduino โดยใช้ Oscilloscope	
3.23	48
เชื่อมต่อเซ็นเซอร์อินฟราเรด, Arduino และจอ LCD	
3.24	48
ผลการทำงานเบื้องต้นของเซ็นเซอร์อินฟราเรดกับ Arduino	
4.1	49
ค่าของจำนวนเหรียญแต่ละชนิดและมูลค่าของเหรียญที่ได้ทำการนับที่ แสดงบนจอมอนิเตอร์ของโปรแกรม ARDUINO	
4.2	50
สัญญาณจากการทดสอบการทำงานของเซ็นเซอร์กับ Arduino โดยใช้	

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.7	36
3.8	37
3.9	37
3.10	38
3.11	38
3.12	39
3.13	39
3.14	40
3.15	41
3.16	42
3.17	43
3.18	44
3.19	44
3.20	45
3.21	46
3.22	47
3.23	48
3.24	48
4.1	49
4.2	50
4.3	50
4.4	51

สารบัญรูป(ต่อ)

รูปที่		หน้า
4.5	ร่างเหรียญที่มีการชะลอความเร็ว และเวลาที่ใช้ในการนับเหรียญบาท จำนวน 40 เหรียญ	51
4.6	ผลการทดสอบความแม่นยำของเซนเซอร์เมื่อนับเหรียญบาท จำนวน 40 เหรียญ (2)	52
4.7	ค่ากระแสไฟฟ้าที่มอเตอร์ใช้ในหมุนก่อนใส่เหรียญ	52
4.8	ค่ากระแสไฟฟ้าที่มอเตอร์ใช้ในหมุนหลังใส่เหรียญ	53
4.9	หน้าจอ LCD ของเครื่องนับเหรียญ	53
4.10	ปุ่มรีเซ็ตและสวิตช์เปิด-ปิด ควบคุมมอเตอร์	55
4.11	ส่วนของการแยกเหรียญทั้ง 4 ชนิด	56
4.12	ส่วนของการนับจำนวนเหรียญแต่ละชนิด และค่าของเหรียญที่ได้ทำการนับ	56
4.13	ช่องของเหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท	57

สารบัญตาราง

ตารางที่	หน้า	
2.1	คุณลักษณะของบอร์ด ARDUINO UNO R3	7
2.2	เชื่อมต่อจอ CHARACTER LCD DISPLAY กับ ARDUINO ในรูปแบบ I2C	14
2.3	การเชื่อมต่อระหว่าง ARDUINO กับจอ LCD (I2C)	15
2.4	ความหมายและวิธีการใช้ไอคอนมุมมองต่างๆ (VIEWTOOL) บนทุลบาร์ มาตรฐาน	22
4.1	ทดสอบการทำงานของเครื่องนับเหรียญแบบแยกชนิดเหรียญ	54
4.2	ทดสอบการทำงานของเครื่องนับเหรียญแบบรวมชนิดเหรียญ	55

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในสภาวะปัจจุบันประเทศไทยมีความก้าวหน้าทางด้านเศรษฐกิจ ทางด้านการศึกษา และ ทางด้านเทคโนโลยี ทำให้มีผู้ประดิษฐ์คิดค้นเครื่องอำนวยความสะดวกในการดำเนินชีวิตประจำวันของ มนุษย์เป็นอย่างมาก เนื่องจากสภาวะเศรษฐกิจที่มีความเจริญก้าวหน้านี้เองจึงทำให้มีการสร้างงาน สร้าง อาชีพให้แก่ชุมชนต่างๆทั้งในตัวเมืองและชนบทไม่ว่าจะเป็นสถานประกอบการรายใหญ่และราย ย่อย ล้วนแล้วแต่มีรายจ่ายและรายรับในวงเงินที่สูงมากในแต่ละวันในบางครั้งจึงเป็นปัญหาการตรวจเช็ค จำนวนเงินโดยเฉพาะเงินที่เป็นเหรียญบาท, เหรียญสองบาท, เหรียญห้าบาทและเหรียญสิบ บาทปนกันอยู่เป็นจำนวนมาก ทำให้ต้องเสียเวลาในการแยกเหรียญ การนับจำนวนเหรียญและการนับจำนวนเงิน อีกทั้งการใช้งานของระบบขายของอัตโนมัติแบบหยอดเหรียญต่างๆ ในท้องตลาดที่มีการใช้ งานอย่างแพร่หลาย ดังนั้นผู้จัดทำจึงได้เกิดแนวคิดที่จะทำเครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์ ซึ่งเป็นเครื่องที่จะอำนวยความสะดวกในการแยกเหรียญนับจำนวนเหรียญและการนับจำนวนเงินเพื่อ ช่วยลดภาระและความยุ่งยากของสถานประกอบการที่มีวงเงินหมุนเวียนเป็นจำนวนมากได้และสามารถนับจำนวนเหรียญจำนวนมากได้อย่างถูกต้อง

โดยการทำงานทั้งหมดของเครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์ทำงานภายใต้การควบคุมของชุดคำสั่งไมโครคอนโทรลเลอร์ที่เขียนโค้ดเก็บไว้บน Arduino เพื่อที่จะสามารถทำงาน ให้เป็นไปตามเงื่อนไขที่กำหนดไว้

1.2 วัตถุประสงค์

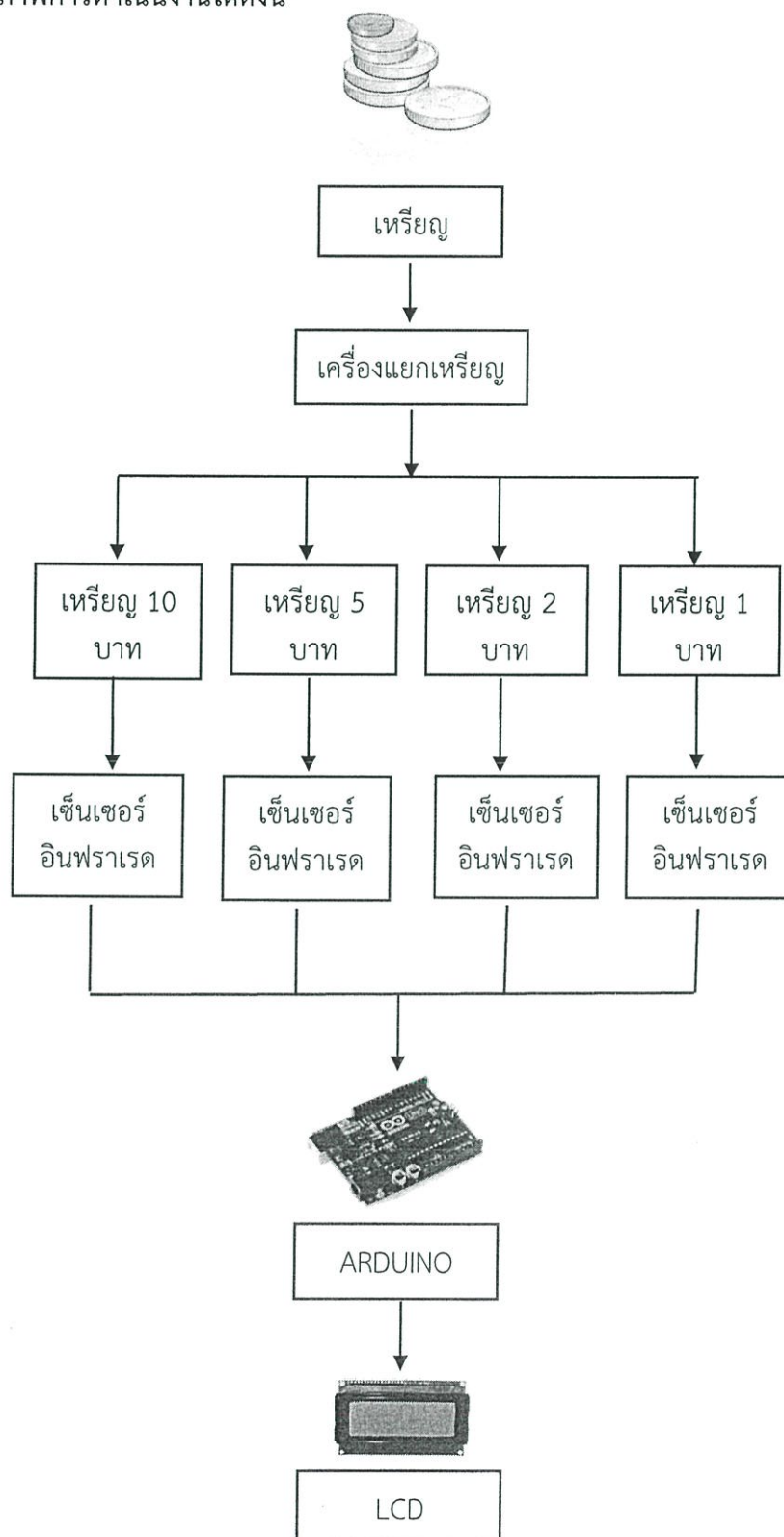
- 1) เพื่อทำการแยกชนิดของเหรียญบาท, เหรียญสองบาท เหรียญห้าบาทและเหรียญสิบบาท
- 2) เพื่อศึกษาลักษณะการเขียนโปรแกรมคำสั่งของไมโครคอนโทรลเลอร์ Arduino และศึกษาหลักการทำงานของเซ็นเซอร์อินฟราเรด
- 3) เพื่อทำการนับค่าของเหรียญทั้งหมด
- 4) เพื่อนำไปประยุกต์ใช้ในการการเงิน

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์นี้คณะผู้จัดทำได้มีการนำ Arduino มาประยุกต์ใช้ในการทำงานร่วมกับเซ็นเซอร์อินฟราเรด เพื่อใช้เป็นระบบการแยกชนิดของเหรียญและนับเหรียญ ในการจัดทำเครื่อง

นับเหรียญนั้นจะมี 2 ส่วนหลักคือ ส่วนของการแยกชนิดของเหรียญ ซึ่งส่วนนี้จะเป็นฮาร์ดแวร์ เป็น การแยกชนิดของเหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท โดยการแยก ชนิดของเหรียญจะเป็นช่องตามขนาดของเหรียญแต่ละชนิด เมื่อทำการแยกเหรียญได้แล้ว ก็ จะเข้าสู่ส่วนที่สอง คือส่วนของการนับจำนวนของเหรียญแต่ละชนิดเป็นส่วนของซอฟต์แวร์ ซึ่งการ นับจำนวนของเหรียญแต่ละชนิดนั้น จะใช้เซ็นเซอร์อินฟราเรดเป็นตัวตรวจจับจำนวนของเหรียญที่ ผ่านการแยกชนิด โดยเซ็นเซอร์อินฟราเรดจะทำการงานร่วมกับระบบประมวลผลของ Arduino จากนั้น Arduino จะทำการประมวลผลคิดค่าของเงินที่ได้ทำการแยก แล้วแสดงผลออกทางจอ LCD โดยทำการแสดงค่าของจำนวนเหรียญแต่ละชนิด และค่าของเหรียญที่ได้ทำการนับ

เขียนแผนภาพการดำเนินงานได้ดังนี้



รูปที่ 1.1 บล็อกไดอะแกรมรวมของเครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

สำหรับโครงการ “เครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์” จะเป็นเครื่องสำหรับการแยกชนิดของเหรียญและนับเหรียญ ในการจัดทำเครื่องนับเหรียญนั้นจะมี 2 ส่วนหลักคือ ส่วนของการแยกชนิดของเหรียญได้แก่ เหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท เมื่อทำการแยกเหรียญได้แล้วก็จะเข้าสู่ส่วนที่สองคือส่วนของการนับจำนวนของเหรียญแต่ละชนิด ซึ่งการนับจำนวนของเหรียญแต่ละชนิดนั้นจะใช้เซ็นเซอร์อินฟราเรด โดยเซ็นเซอร์อินฟราเรดจะทำงานร่วมกันกับระบบประมวลผลของ Arduino จากนั้น Arduino จะทำการประมวลผลคิดค่าของเงินที่ได้ทำการแยก แล้วแสดงผลออกทางจอ LCD โดยทำการแสดงค่าของจำนวนเหรียญแต่ละชนิด และค่าของเหรียญที่ได้ทำการนับ

2.1 ไมโครคอนโทรลเลอร์ AVR

ไมโครคอนโทรลเลอร์ AVR เป็นหนึ่งในไมโครคอนโทรลเลอร์ที่ผลิตโดยบริษัท ATMEL (ผู้นำทางด้านไมโครคอนโทรลเลอร์ตระกูล MCS-51) เป็นไมโครคอนโทรลเลอร์ตระกูลใหม่จาก ATMEL มีสถาปัตยกรรมแบบ RISC (Reduce Instruction Set Computer) คือหนึ่งคำสั่งทำงานใช้สัญญาณนาฬิกาเพียง 1 ลูก (Instruction in a single clock cycle) เป็นไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพและความสามารถสูง แบ่งออกเป็นหลายอนุกรม ในแต่ละอนุกรมยังแบ่งออกเป็นหลายเบอร์ เพื่อรองรับความต้องการที่แตกต่างของผู้งาน ในขณะที่ยังคงประสิทธิภาพเท่ากัน ซึ่ง Arduino นั้นเป็นไมโครคอนโทรลเลอร์ชนิดหนึ่งที่พัฒนามาแล้ว ทำให้สามารถเขียนคำสั่งโปรแกรมได้เสมือนโปรแกรมขั้นสูงทั่วไปโครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์นั้นสามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆ ดังต่อไปนี้

2.1.1 หน่วยประมวลผลกลาง

2.1.2 หน่วยความจำ

สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ คือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกระดาษทดในการคำนวณของซีพียูและเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยงข้อมูลก็จะหายไปคล้ายกับหน่วยความจำแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่วไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรมซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยงและเป็นอีอีพรอม (Erasable Electrically Read-Only Memory, EEPROM) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

2.1.3 ส่วนติดต่อกับอุปกรณ์ภายนอกหรือพอร์ต

มี 2 ลักษณะคือพอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุตเพื่อรับสัญญาณอาจจะด้วยการกดสวิตช์เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุตเพื่อแสดงผล เช่น การติดสว่างของหลอดไฟ เป็นต้น

2.1.4 ช่องทางเดินของสัญญาณหรือบัส

เส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่างซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณจำนวนมากอยู่ในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus), บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

2.1.5 วงจรกำเนิดสัญญาณนาฬิกา

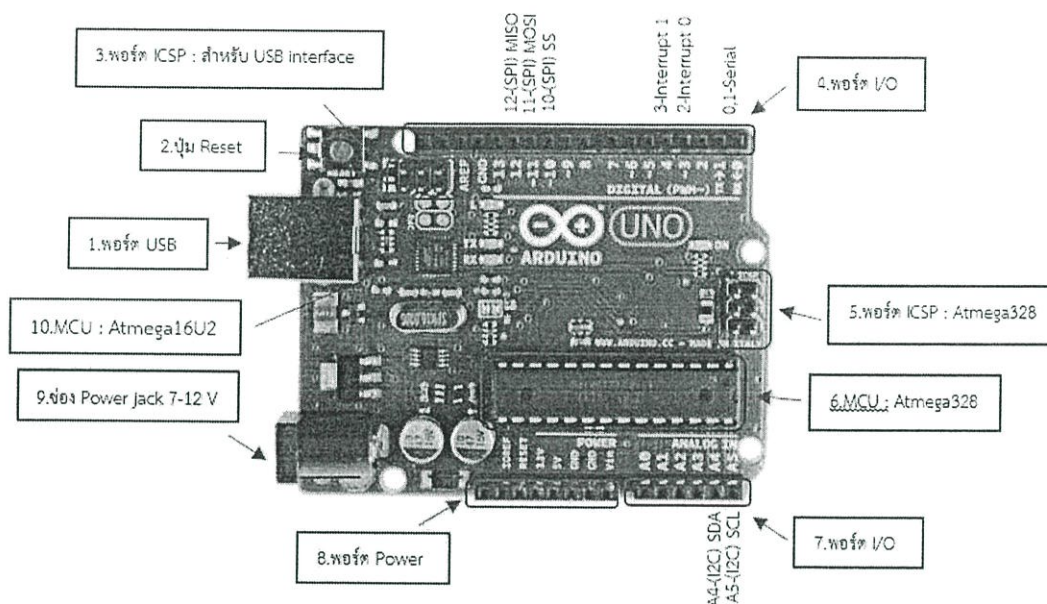
นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับข้อกำหนดจังหวะหากสัญญาณนาฬิกามีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้นมีความเร็วในการประมวลผลสูงตามไปด้วย

2.2 Arduino

Arduino เป็นภาษาอิตาลี ซึ่งเป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software โดยพบว่าได้รับการปรับปรุงมาจากอีกโครงการ “Wiring” ที่ใช้ ATmega128 มีหน่วยความจำและ I/O ค่อนข้างมาก ที่ตัวถังของ ATmega128 เป็นอุปสรรคกับผู้ใช้งานในเบื้องต้น จึงไม่เป็นที่นิยม ทีมงานของ Arduino ได้นำมาพัฒนาต่อยอดโดยให้สามารถใช้งาน AVR ขนาดเล็กทำให้วงจรของบอร์ดมีขนาดเล็กลงอีกทั้งยังใช้อุปกรณ์จำนวนน้อยอีกด้วย ทำให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ดหรือโปรแกรมต่อได้อีกด้วย สำหรับในการทดลองดังกล่าวจะเป็นการเริ่มต้นศึกษาการใช้งานบอร์ดไมโครคอนโทรลเลอร์ Arduino Model: Arduino UNO R3 ใช้ชิพ ATmega328 รั้นที่ความถี่ 16 MHz หน่วยความจำแฟลช 32 KB แรม 2 KB บอร์ดใช้ไฟเลี้ยง 7 ถึง 12 V มีระดับแรงดันไฟฟ้าในการทำงานและขาสัญญาณอยู่ที่ 5 V (TTL) มี Digital Input / Output 14 ขา (เป็น PWM ได้ 6 ขา) มี Analog Input 6 ขา Serial UART 1 ชุด I2C 1 ชุด SPI 1 ชุด เขียนโปรแกรม บนซอฟต์แวร์ Arduino IDE และโปรแกรมผ่านพอร์ต USB

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือ ผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วสามารถเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield และ Arduino GPRS Shield มาเสียบบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้

2.2.1 Layout & Pin out Arduino Board



รูปที่ 2.1 ส่วนประกอบของ Arduino board รุ่น UNO R3 [2]

2.2.2 รายละเอียดอุปกรณ์

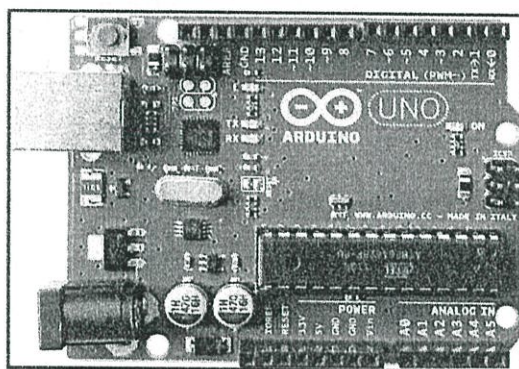
1. USB Port ใช้สำหรับต่อกับ Computer เพื่ออัปเดตโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด Arduino
2. Reset Button เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
3. ICSP Port เป็นพอร์ตที่ใช้โปรแกรม Visual Comport บน Atmega 16U2
4. I/o Port, digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะ ทำหน้าที่อื่นเพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx, Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM
5. ICSP Port Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
6. MCU Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino
7. I/O Port นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็นช่องรับสัญญาณอนาล็อกได้อีกด้วยตั้งแต่ ขา A0-A5
8. Power Port ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอกประกอบด้วยขาไฟเลี้ยง +3.3V,+5V , GND , Vin
9. Power Jack รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
10. MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega 328 จะติดต่อ ผ่าน Atmega16U2

2.2.3 จุดเด่นของบอร์ด Arduino

- 1.ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อน เหมาะสำหรับผู้เริ่มต้นงาน
- 2.มี community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแรง
- 3.Open hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
- 4.ราคาไม่แพง
- 5.Cross platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้

2.2.4 รายละเอียดคุณลักษณะ

บอร์ด Arduino ที่ได้เลือกศึกษาเพื่อนำมาใช้งานคือรุ่น Arduino Uno R3 ดังรูปที่ 2.2 ซึ่งเป็นรุ่นพื้นฐาน และมีรายละเอียดคุณลักษณะดังตารางที่ 2.1



รูปที่ 2.2 Arduino Uno R3 Board [2]

ตารางที่ 2.1 คุณลักษณะของบอร์ด Arduino Uno R3

ไมโครคอนโทรลเลอร์	ATmega328
แรงดันที่ใช้งาน	5V
แรงดันอินพุต	7-12V
แรงดันอินพุตสูงสุด	6-20V
พิน Digital I/O	14 พิน (6 พินให้เอาต์พุตแบบ PWN)
พิน Analog Input	6 พิน
แรงดันไฟตรงต่อพิน I/O	40 mA
แรงดันไฟตรงที่พิน 3.3V	50 mA
Flash Memory	32 KB (ATmega328) โดยโปรแกรม bootloader ใช้พื้นที่แล้ว 0.5 KB

ไมโครคอนโทรลเลอร์	ATmega328
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

2.2.5 ชนิดข้อมูลพื้นฐานของ Arduino

1. Byte ใช้สำหรับข้อมูลที่เป็นเลขจำนวนเต็มได้ตั้งแต่ 0 ถึง 255
2. Int ใช้สำหรับข้อมูลที่เป็นเลขจำนวนเต็มได้ตั้งแต่ -32768 ถึง +32767
3. Float ใช้สำหรับข้อมูลที่เป็นเลขทศนิยมเป็นค่าเป็นบวกหรือลบ ในช่วงที่กว้างกว่าชนิดข้อมูลแบบ byte หรือ int และเป็นเลขทศนิยมได้ด้วย แต่มีความละเอียดเพียง 6-7 ตำแหน่ง หลังจุดทศนิยมในเลขฐานสิบ

4. Boolean ใช้สำหรับข้อมูลที่เป็นค่าลอจิก true(จริง) หรือ false(ผิด) เท่านั้น

2.2.6 คำสั่งพื้นฐาน

Int คือการประกาศตัวแปร เมื่อเราประกาศตัวแปรเช่น int a = 13 แปลว่าเราจะต้องใช้ตัวแปร a แทน port 13 ตลอดทั้งการเขียนโปรแกรม

pinMode() ใช้กำหนดทิศทางสัญญาณ input/output ของขา digital

digitalRead() ใช้อ่านค่าจากขา digital ที่ถูกกำหนดให้เป็นอินพุต

digitalWrite() ใช้เขียนค่า LOW, HIGH ให้ขา digital ที่ถูกกำหนดให้เป็นเอาต์พุต

analogWrite() ใช้สร้างสัญญาณ PWM เป็นเอาต์พุต

analogRead() ใช้อ่านค่าจากขา analog-input

delay() รอเวลาให้ผ่านไปตามระยะเวลาที่กำหนด(มิลลิวินาที) ก่อนที่จะทำขั้นตอน

ต่อไป

delayMicroseconds() รอเวลาให้ผ่านไปตามระยะเวลาที่กำหนด (ไมโครวินาที)

random() ให้ค่าเป็นเลขสุ่ม

min() ให้ค่าน้อยสุดระหว่างตัวเลข 2 ค่าที่นำมาเปรียบเทียบกัน

max() ให้ค่ามากที่สุดระหว่างตัวเลข 2 ค่าที่นำมาเปรียบเทียบกัน

2.2.7 ตัวอย่างโปรแกรมพื้นฐานของ arduino

ในการเขียนโปรแกรม Arduino นั้นจะมีส่วนประกอบอยู่ 2 ส่วนคือส่วนที่หนึ่งสำหรับการกำหนดค่าต่างๆและส่วนที่สองคือการสั่งโปรแกรมทำงาน โดย แบ่งออกได้ดังนี้

```
void setup ()
{
}
```

ในส่วนนี้จะเป็นการกำหนดค่าต่างๆ เช่น จะใช้กำหนดขาในการใช้งานให้เป็น INPUT หรือ OUTPUT ซึ่งการกำหนดส่วนนี้จะกำหนดเพียงครั้งเดียวเท่านั้น เช่น

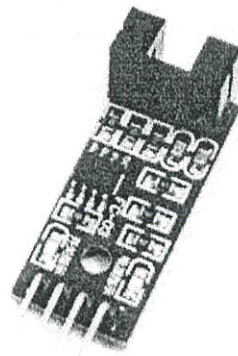
```
void setup()
{
  pinMode(led1, OUTPUT);
}
Void loop ()
{
}
```

ในส่วนนี้จะเป็นส่วนการเขียนโปรแกรมและสั่งให้โปรแกรมทำงาน ซึ่งในการทำงานจะเป็นแบบวนลูปไปเรื่อยๆ เช่น

```
void loop()
{
  digitalWrite(led1,HIGH); )
  delay(1000);
  digitalWrite(led1,LOW); );
  delay(1000);
}
```

2.3 Infrared Sensor Optocoupler Module (เซ็นเซอร์กัมพู)

Infrared Sensor Optocoupler Modules หรือเซ็นเซอร์กัมพู มีหลักการทำงานของเซ็นเซอร์โดยใช้แสงส่องผ่านจากตัวส่งเป็นอินฟราเรดมาที่ตัวรับซึ่งเป็นโฟโตทรานซิสเตอร์โดยจะมีช่องว่างระหว่างตัวส่งและตัวรับ เซ็นเซอร์จะทำงานเปลี่ยนสถานะเมื่อมีอุปสรรคผ่านช่องว่าง โดยจะให้สถานะที่ต่างกัน เมื่อถูกอุปสรรคตัดผ่าน และเมื่อไม่มีอุปสรรคตัดผ่านพร้อมหลอดไฟ LED แสดงการทำงาน โดยเอาท์พุทจะเป็น 1 เมื่อมีวัตถุมาบังที่ร่องตรวจนับทำให้หลอดไฟ LED ติด



รูปที่ 2.3 Infrared Sensor Optocoupler Module [3]

2.3.1 คุณสมบัติของเซ็นเซอร์

1. การทำงานของเซ็นเซอร์เกิดขึ้นบริเวณช่องว่างของเซ็นเซอร์
2. เซ็นเซอร์มีช่องว่างสำหรับอุปกรณ์ ในการตัดผ่าน 5 มิลลิเมตร
3. สถานะการทำงานของเอ้าท์พุทจะมีหลอดไฟ LED แสดงการทำงาน
4. เมื่อมีสิ่งกีดขวางสถานะของเอ้าท์พุทจะเป็น high output และถ้าไม่มีสิ่งกีดขวางสถานะของเอ้าท์พุทจะเป็น low output
5. การใช้งานจะใช้แรงดันไฟฟ้า(Input) 3.3V - 5V
6. รูปแบบการแสดงผลของเอ้าท์พุท จะเป็นการส่งออกเป็นสัญญาณดิจิตอลสลับกัน (0 และ 1) กล่าวคือ DO (Digital Out) และ AO (Analog Out)
7. ร่องของเซ็นเซอร์มีขนาดคงที่และง่ายต่อการติดตั้ง
8. แผ่น PCB มีขนาดเล็กคือ 3.2cm x 1.4cm
9. มีไฟแสดงสถานะแหล่งจ่าย และสถานะเมื่อเซ็นเซอร์ทำงาน

2.3.2 ตัวอย่างการเขียนโค้ดใช้งาน

สามารถอ่านค่าโดยต่อเข้ากับขา DO และเขียนการใช้งานแบบเดียวกับการอ่านสวิตช์ เมื่อต้องการให้ Active LOW หรือ Active HIGH สามารถออกแบบได้

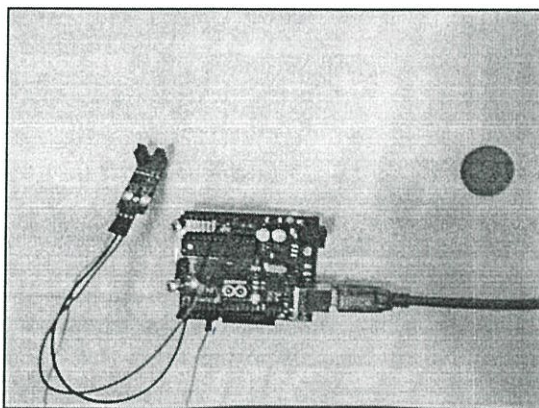
```
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}
// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
```

```

Serial.println(sensorValue);
delay(100);
// delay in between reads for stability
}

```

2.3.3 ตัวอย่างการเชื่อมต่อเซ็นเซอร์กัมพูกับ Arduino

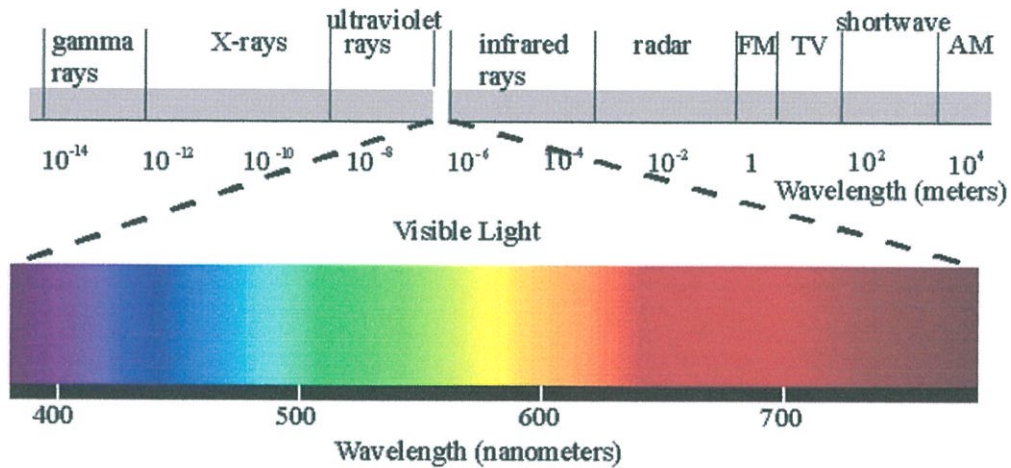


รูปที่ 2.4 Infrared Sensor Optocoupler Module กับ ARDUINO

2.4 รังสีอินฟราเรด (Infrared)

2.4.1 ความหมายของรังสีอินฟราเรด

รังสีอินฟราเรดหรือรังสีใต้แดงมีคลื่นความถี่สั้น (Millimeter waves) เป็นคลื่นแม่เหล็กไฟฟ้าชนิดหนึ่งที่มีความถี่อยู่ระหว่าง 1011 - 1014 เฮิรตซ์ และมีความยาว 0.75 ไมโครเมตรถึง 1000 ไมโครเมตร สสารที่มีอุณหภูมิมากกว่า 0 องศาเซลวิน จะปล่อยรังสีอินฟราเรดออกมาจากตัวเองเสมอ ลำแสงอินฟราเรดเดินทางเป็นเส้นตรงไม่สามารถผ่านวัตถุทึบแสง และสามารถสะท้อนแสงในวัสดุผิวเรียบได้เหมือนกับแสงทั่วไปใช้มากในการสื่อสารระยะใกล้ โดยคลื่นสั้นนี้จะไม่มีปัญหาในเรื่องการ Interference (การรบกวนของสัญญาณ)



รูปที่ 2.5 Electromagnetic Spectrum [4]

จากรูปแสดงคลื่นแม่เหล็กไฟฟ้าในช่วงความยาวคลื่นต่างๆจะเห็นได้ว่ารังสีอินฟราเรด จะมีความยาวคลื่นที่มากกว่าแสงที่ตามองเห็นที่ยาวที่สุดก็คือสีแดง รังสีอินฟราเรดสามารถแบ่งความยาวคลื่นได้เป็น 3 ช่วง คือ รังสีอินฟราเรดช่วงคลื่นสั้น (NIR) มีความยาวประมาณ 0.7 ไมโครเมตรถึง 1.5 ไมโครเมตรมักประยุกต์ใช้ในภาพถ่ายความร้อน รังสีอินฟราเรดช่วงคลื่นกลาง(MIR) มีความยาวประมาณ 1.5 ไมโครเมตรถึง 5.6 ไมโครเมตรมักประยุกต์ใช้กับระบบวิถีจรวด MISSILE และรังสีอินฟราเรดช่วงคลื่นยาว(FIR)) มีความยาวประมาณ 5.6 ไมโครเมตรขึ้นไป มักประยุกต์ในการบำบัดผู้ป่วย เช่น อาการปวดเมื่อยเรื้อรัง ความดันโลหิตสูง รวมถึงการควบคุมน้ำหนัก

2.4.2 การค้นพบรังสีอินฟราเรด

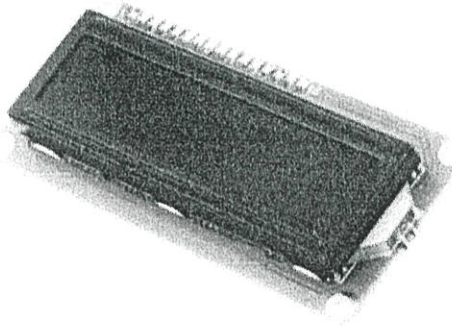
Sir William Herschel เป็นผู้ค้นพบเมื่อปี ค.ศ. 1800 ซึ่งได้ทำการแยกแสงสีต่างๆ ออกจากแสงสีขาวโดยใช้ปริซึมเพื่อต้องการวัดความร้อนของแสงสีต่างๆโดยการนำเทอร์โมมิเตอร์ ทาสีดำไปวางที่แถบสีต่างๆเพื่อวัดอุณหภูมิ แต่เพื่อควบคุมการทดลองจึงได้นำเทอร์โมมิเตอร์ไปวางไว้บนแสงสีดำ จากการทดลองนี้ทำให้ทราบว่า อุณหภูมิของแสงเพิ่มขึ้นจากสีม่วงไปหาสีแดง และพบว่าแถบสีที่มองไม่เห็นเหนือสีแดงมีอุณหภูมิสูงที่สุดซึ่งต่อมาเป็นที่รู้จักในชื่อ รังสีอินฟราเรด

2.4.3 การประยุกต์ใช้รังสีอินฟราเรด

รังสีอินฟราเรดถูกนำมาประยุกต์ใช้งานอย่างหลากหลาย ตัวอย่างเช่น กล้องถ่ายภาพความร้อน(Thermal camera) ซึ่งเป็นกล้องที่ใช้ถ่ายภาพความร้อนในย่านแสงอินฟราเรด เพื่อแสดงความร้อนของวัตถุโดยไม่ต้องสัมผัสกับวัตถุทำให้ทราบค่าอุณหภูมิได้อย่างสะดวก ม่านแสงอินฟราเรด(Barrier Curtain Sensor) ซึ่งเป็นอุปกรณ์กันขโมยประเภทหนึ่งติดตั้งไว้กับบริเวณหน้าต่างหรือบริเวณทางเข้าประตู โดยม่านแสงอินฟราเรดจะประกอบด้วยเสาส่งอินฟราเรดจำนวน 2 เสา

และระบบติดตามเป้าหมายด้วยกล้องอินฟราเรด(Infrared Homing) มักติดตั้งไว้หน้าจรวดMissile ของเครื่องบิน โดยที่กล้องอินฟราเรดจะตรวจจับความร้อนที่ออกมาจากเครื่องบินไอพ่น

2.5 Liquid Crystal Display (LCD)



รูปที่ 2.6 จอแสดงผล LCD [5]

จอ Liquid Crystal Display (LCD) เป็นจอแสดงผลรูปแบบหนึ่งที่นิยมนำมาใช้ งานกันกับระบบสมองกลฝังตัวอย่างแพร่หลาย จอ LCD มีทั้งแบบแสดงผลเป็นตัวอักษรเรียกว่า Character LCD ซึ่งมีการกำหนดตัวอักษรหรืออักขระที่สามารถแสดงผลไว้ได้อยู่แล้ว และแบบที่สามารถแสดงผลเป็นรูปภาพหรือสัญลักษณ์ได้ตามความต้องการของผู้ใช้งานเรียกว่า Graphic LCD นอกจากนี้บางชนิดเป็นจอที่มีการผลิตขึ้นมาใช้เฉพาะงาน ทำให้มีรูปแบบและรูปร่างเฉพาะเจาะจงใน การแสดงผล เช่น นาฬิกาดิจิตอล เครื่องคิดเลข หรือ หน้าปัดวิทยุ เป็นต้น

2.5.1 โครงสร้างโดยทั่วไปของ LCD

โครงสร้างของ LCD ทั่วไปจะประกอบขึ้นด้วยแผ่นแก้ว 2 แผ่นประกบกันอยู่ โดย เว้นช่องว่างตรงกลางไว้ 6-10 ไมโครเมตร ผิวด้านในของแผ่นแก้วจะเคลือบด้วยตัวนำไฟฟ้าแบบใส เพื่อใช้แสดงตัวอักษร ตรงกลางระหว่างตัวนำไฟฟ้าแบบใสกับผลึกเหลว จะมีชั้นของสารที่ทำให้ โมเลกุลของผลึกรวมตัวกันในทิศทางที่แสงส่องมากระทบเรียกว่า Alignment Layer และผลึกเหลว ที่ใช้โดยทั่วไปจะเป็นแบบ Magnetic โดย LCD สามารถแสดงผลให้เรามองเห็นได้ทั้งหมด 3 แบบ ด้วยกันคือ

1. แบบใช้การสะท้อนแสง (Reflective Mode) LCD แบบนี้ใช้สารประเภทโลหะ เคลือบอยู่ที่แผ่นหลังของ LCD ซึ่ง LCD ประเภทนี้เหมาะกับการนำมาใช้งานในที่ที่มีแสงสว่างเพียงพอ
2. แบบใช้การส่องผ่าน (Transitive Mode) LCD แบบนี้วางหลอดไฟไว้ด้านหลังจอ เพื่อให้การอ่านค่าที่แสดงผลทำได้ชัดเจน
3. แบบส่องผ่านและสะท้อน (Transflective Mode) LCD แบบนี้เป็นการนำเอา ข้อดีของจอแสดงผล LCD ทั้ง 2 แบบมารวมกัน

จอ LCD ที่แสดงผลเป็นอักขระหรือตัวอักษร ตามท้องตลาดทั่วไปจะมีหลายแบบด้วยกัน มีทั้ง 16 ตัวอักษร 20 ตัวอักษรหรือมากกว่า และจำนวนบรรทัดจะมีตั้งแต่ 1 บรรทัด 2 บรรทัด 4 บรรทัดหรือมากกว่าตามแต่ความต้องการและลักษณะของงานที่ใช้ หรืออาจจะมีแบบสั่งทำเฉพาะงานก็เป็นได้ ในที่นี้ยกตัวอย่างจอ LCD ขนาด 16x2 Character หรือที่นิยมเรียกกันว่าจอ LCD 16 ตัวอักษร 2 บรรทัด สามารถหาซื้อได้ง่ายและมีราคาไม่สูง เหมาะสมกับการใช้งานแสดงผลไม่มากในหน้าจอดีียว

จอ LCD 16x2 Character ที่นิยมวางจำหน่ายจะมีอยู่ 2 แบบด้วยกันคือ LCD แบบปกติที่เชื่อมต่อแบบขนาน (Parallel) และ LCD แบบที่เชื่อมต่ออนุกรม (Serial) แบบ I2C โดยทั้ง 2 แบบตัวจอมีลักษณะเดียวกันเพียงแต่แบบ I2C จะมีบอร์ดเสริมทำให้สื่อสารแบบ I2C ได้เชื่อมต่อได้สะดวกขึ้น

2.5.2 การเชื่อมต่อจอ Character LCD Display 16x2 กับ Arduino ในรูปแบบ I2C (Serial)

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

ตารางที่ 2.2 การเชื่อมต่อจอ Character LCD Display กับ Arduino ในรูปแบบ I2C

PIN NO	Symbol	Description
1	GND	Ground
2	VCC	+5VDC
3	SDA	Serial Data
4	SCL	Serial Clock

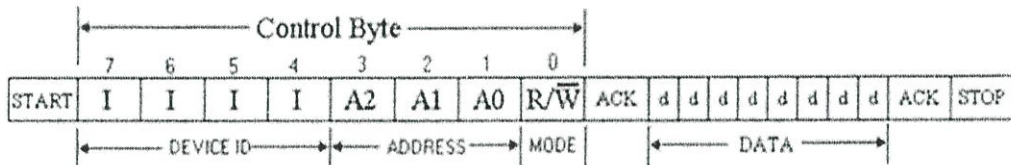
2.5.3 การควบคุมการแสดงผลของจอ LCD (I2C)

ในการควบคุมหรือสั่งงาน โดยทั่วไปจอ LCD จะมีส่วนควบคุม (Controller) อยู่ในตัว ผู้ใช้สามารถส่งรหัสคำสั่งสำหรับควบคุมการทำงานของจอ LCD (I2C) เช่นเดียวกันกับจอ LCD แบบธรรมดา คือรหัสคำสั่งที่ใช้ในการควบคุมนั้นเหมือนกัน แต่ต่างกันตรงที่รูปแบบในการรับส่ง จอ LCD 16x2 ที่มีการส่งข้อมูลรูปแบบ I2C ที่ใช้ขาเพียง 4 ขาที่ใช้ในการเชื่อมต่อเท่านั้น

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

2.5.4 การรับ-ส่งข้อมูลแบบ I2C BUS



รูปที่ 2.7 การรับ-ส่งข้อมูลแบบ I2C BUS [5]

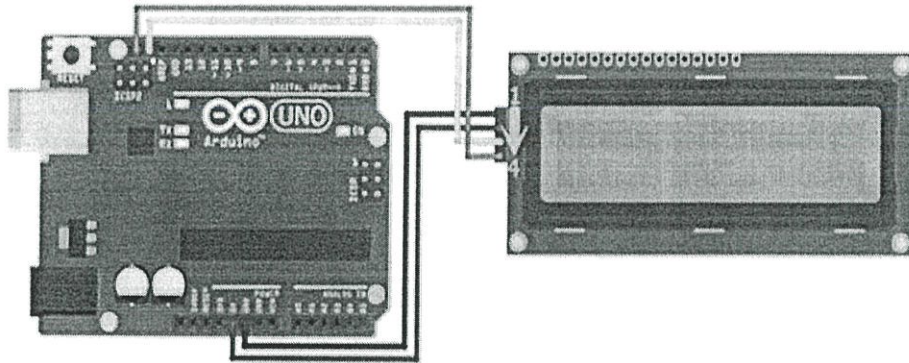
MCU จะทำการส่งสถานะเริ่มต้น (START Conditions) เพื่อแสดงการขอใช้บัส แล้วตามด้วย รหัสควบคุม (Control Byte) ซึ่งประกอบด้วยรหัส ประจำตัวอุปกรณ์ Device ID, Device Address และ Mode ในการเขียนหรืออ่านข้อมูล เมื่ออุปกรณ์รับทราบว่า MCU ต้องการจะติดต่อด้วย ก็ต้องส่งสถานะรับรู้ (Acknowledge) หรือแจ้งให้ MCU รับรู้ว่าข้อมูลที่ได้ส่งมามีความถูกต้อง และเมื่อสิ้นสุดการส่งข้อมูล MCU จะต้องส่งสถานะสิ้นสุด (STOP Conditions) เพื่อบอกกับอุปกรณ์ว่า สิ้นสุดการใช้บัส

2.5.5 การเชื่อมต่อระหว่าง Arduino กับจอ LCD (I2C)

ตารางที่ 2.3 การเชื่อมต่อระหว่าง Arduino กับจอ LCD (I2C)

Arduino UNO R3	LCD (I2C)
GND	GND (PIN1)
+5VDC	VCC (PIN2)
A4 (SDA)	SDA (PIN3 Serial Data)
A5 (SCL)	SCL (PIN4 Serial Clock)

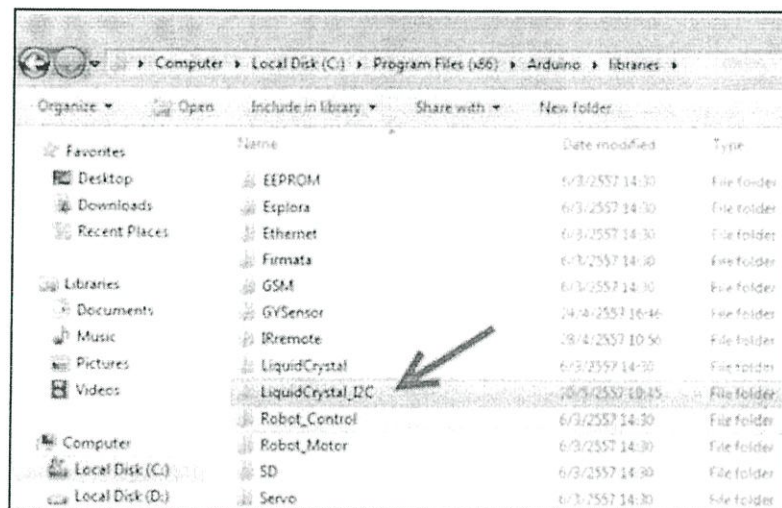
2.5.6 รายละเอียดชุดคำสั่งที่ใช้ในการสั่งงานระหว่าง Arduino กับจอ LCD



รูปที่ 2.8 การเชื่อมต่อ Arduino กับจอ LCD [5]

คำสั่งที่ใช้ในการควบคุมจอ LCD ของ Arduino แบบ I2C นั้นไม่ต่างจากจอ LCD แบบธรรมดา (Parallel) มากนัก ทั้งนี้ยังได้มีการพัฒนา Library I2C มาให้ใช้งานได้อย่างสะดวกมากยิ่งขึ้น ขั้นตอนแรกคือการนำไฟล์ Library LCD (I2C) ไปไว้ใน Library ของ Arduino ก่อน เพื่อให้ง่ายต่อการเรียกใช้ Library ผู้ใช้สามารถทำตามขั้นตอนด้านล่างได้

1. Download library LCD (I2C) คือ ไฟล์ LiquidCrystal_I2C.7z



รูปที่ 2.9 ไฟล์ LiquidCrystal_I2C.7z [5]

2. ทำการแยกไฟล์ที่โหลด

ทำการแยกไฟล์แล้วนำไปไว้ยัง C:\ProgramFiles(x86)\Arduino\libraries (Drive ที่ได้ทำการติดตั้งโปรแกรม Arduino IDE ไว้) หลังจากที่ทำนำไฟล์ Library ไปไว้เรียบร้อยแล้ว เริ่มการเขียนโปรแกรม ก่อนอื่นให้เรียก Library ของการสื่อสารแบบ I2C และ

Library ของจอ LCD (I2C) คือ Wire.h และ LiquidCrystal_I2C.h หลังจากทำการเรียก Library ฟังก์ชันแรก LiquidCrystal_I2C(); ใช้ในการประกาศ Address และ ขนาดของจอ รูปแบบของคำสั่งคือ LiquidCrystal_I2C lcd(Address, columns, rows); ตัวอย่างนี้จะกำหนดดังนี้

2.1 LiquidCrystal_I2C lcd(0x27, 16, 2);

หมายถึง Address ของจอคือ 27 ขนาดของจอคือ 16 ตัวอักษร 2 บรรทัด

2.2 begin();

หมายถึง ฟังก์ชันใช้ในการกำหนดการเริ่มต้นในการใช้งานจอ

2.3 setCursor();

หมายถึง ฟังก์ชันใช้ในการกำหนดตำแหน่งของ Cursor เช่น
lcd.setCursor (0, 1) ;

2.4 print();

หมายถึง ฟังก์ชันใช้ในการกำหนดข้อความที่ต้องการแสดง

2.5.7 ตัวอย่างโค้ดโปรแกรม

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h> //ประกาศ Library ของจอ I2C
// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup()
{
  // initialize the LCD
  lcd.begin();
  // Print a message to the LCD.
  lcd.print("Hello !!!"); //ฟังก์ชันในการกำหนดข้อความที่ต้องการแสดงผล
  lcd.setCursor(0, 1); //ฟังก์ชันในการกำหนดตำแหน่ง Cursor
  lcd.print("ThaiEasyElec");
}
void loop()
```

{
}

2.6 โปรแกรม Solidworks



รูปที่ 2.10 โปรแกรม Solidworks [6]

2.6.1 ทำความรู้จักกับโปรแกรม Solidworks

โปรแกรม Solidworks เป็นโปรแกรมที่ใช้ในการออกแบบงานด้าน 3 มิติ (3D) ซึ่งเป็นโปรแกรมที่มีความสามารถในการออกแบบได้อย่างหลากหลายและเป็นโปรแกรมที่ใช้งานได้ง่ายมาก ซึ่งใช้งานกันอย่างแพร่หลายทั้งในระดับการศึกษาและระดับอุตสาหกรรม มีการเรียนการสอนในมหาวิทยาลัย ทำให้โปรแกรม SolidWorks มีผู้ใช้จำนวนมากเพราะมีแหล่งศึกษาที่หาได้ง่าย

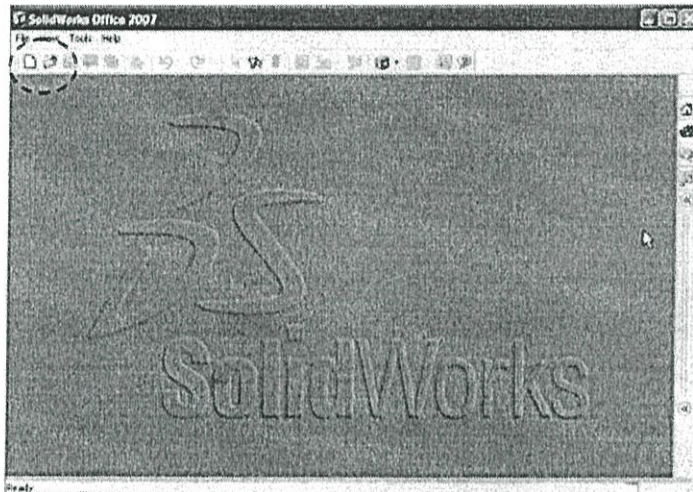
SolidWorks พัฒนาขึ้นในปี 1995 โดยบริษัท Dassault System ในฝรั่งเศสเป็นซอฟต์แวร์เพื่อให้นักออกแบบใช้ เป็นเครื่องมือในการออกแบบทางวิศวกรรม เพื่อสร้างตัวอย่างผลิตภัณฑ์จำลองใน Computer ก่อนที่จะสร้างผลิตภัณฑ์ต้นแบบจริง โดยตัวซอฟต์แวร์จะจัดอยู่ในตระกูล CAD (Computer Aided Design) ซึ่งสามารถสร้างชิ้นงานจำลองในรูปแบบ 3D Solid Models เป็นแบบงานแยกชิ้น (Part) และแบบงานประกอบ (Assembly) เพื่อนำไปสร้างเป็น 2D Standard Engineering ซึ่งมีฟังก์ชันการใช้งานดังต่อไปนี้

- 1.การสร้าง Part Solid ใช้วิธีการและเทคโนโลยี Surface Modeling (NURBS)
- 2.Assembly Modeling สามารถประกอบชิ้นส่วน 3 มิติได้เร็วขึ้น โดยมีขนาดไฟล์เล็กและใช้หน่วยความจำน้อย
- 3.Drawing สร้าง Drawing 2 มิติจาก 3 มิติ โดยอัตโนมัติและบันทึกไฟล์เป็น .dwg ได้
- 4.Simulation ใช้ทดสอบการเคลื่อนที่และตรวจสอบชิ้นงานส่วนที่ขัดกัน

5. Animator สร้างภาพเคลื่อนไหวแสดงการทำงานของชิ้นส่วนหรือเครื่องจักรกล และสามารถบันทึกไฟล์เป็น .AVI(ไฟล์วีดีโอ)ได้

6. Sheet Metal สามารถสร้างงานพับแบบต่างๆ และทำแผ่นคลี่งานโลหะแผ่นได้
Module การใช้งานอื่นๆ เช่น การวิเคราะห์ไฟไนต์เอลิเมนต์เบื้องต้น

เมื่อเปิดโปรแกรม solidworks จะมีวินโดวหรือหน้าต่างดังรูปที่ 2.11 ซึ่งมีเมนูหลักคล้ายโปรแกรม Microsoft office



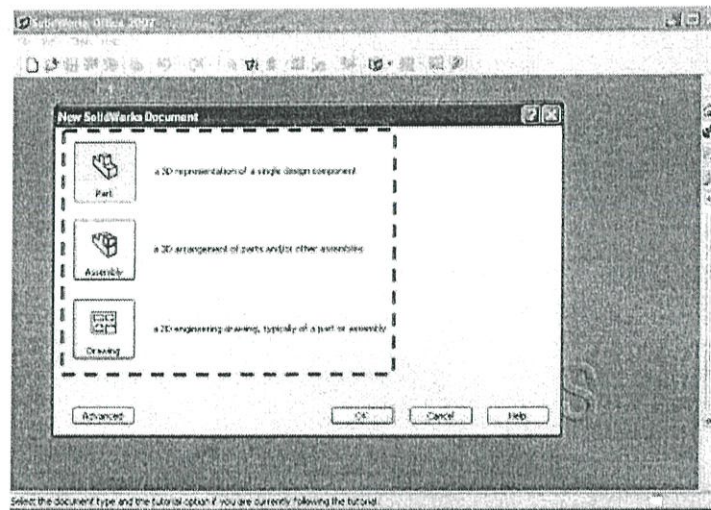
รูปที่ 2.11 หน้าต่างโปรแกรม Solidworks (1) [6]

New เป็นไอคอนที่คลิกเมื่อต้องการเริ่มสร้างงานใหม่ จะประกอบด้วยไฟล์หลายๆไฟล์ให้เลือกใช้ตามวัตถุประสงค์ของงานที่จะทำ

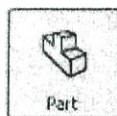
Open เป็นไอคอนที่คลิกเมื่อต้องการเปิดไฟล์ที่สร้างแล้วบันทึกเก็บไว้

2.6.1.1 ชนิดของไฟล์ในโปรแกรม solidworks

เมื่อคลิกไอคอน New ก็จะปรากฏหน้าต่างรูปที่ 2.12 ซึ่งประกอบด้วยไฟล์ที่มีนามสกุลต่างๆตามลักษณะการใช้งานดังนี้



รูปที่ 2.12 หน้าต่างโปรแกรม Solidworks (2) [6]



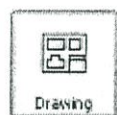
Part

เป็นไฟล์ที่ใช้สร้างชิ้นส่วน(Part) หนึ่งไฟล์จะมีเพียงชิ้นส่วนประกอบเดียวเท่านั้นสามารถสร้างให้เป็น 2 มิติ หรือ 3 มิติก็ได้



Assembly

เป็นไฟล์ที่ใช้สำหรับนำชิ้นส่วนเดียวหรือไฟล์ชิ้นส่วนประกอบย่อย (Sub assemblies) มาประกอบกัน

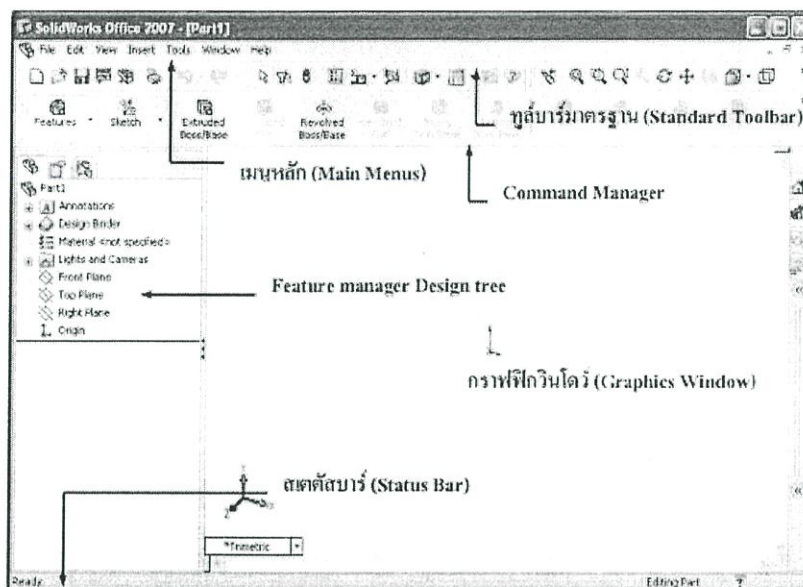


Drawing

เป็นไฟล์ที่ใช้สร้างงานเขียนแบบสั่งงานผลิต(Drawing) โดยการนำไฟล์ part หรือไฟล์ Assembly มาวางในไฟล์นี้ สามารถกำหนดขนาดและสัญลักษณ์ในการเขียนแบบต่างๆเพื่อนำไปสั่งงานผลิตชิ้นงานตามที่ได้ออกแบบไว้

2.6.1.2 หน้าจอติดต่อกับผู้ใช้งาน

โปรแกรม Solidworks จะรันอยู่บนระบบปฏิบัติการ ซึ่งมีเมนูบาร์ทูลบาร์ต่างๆให้ใช้งานอย่างง่ายๆเหมือนโปรแกรมอื่นๆที่รันบนระบบปฏิบัติการวินโดวส์ทั่วไป จะมีส่วนประกอบต่างๆ ดังรูปที่ 2.13



รูปที่ 2.13 หน้าต่างโปรแกรม Solidworks (3) [6]

Main Menus เป็นแถบคำสั่งที่เรียกใช้งาน โดยใช้เมาส์เลือกจะแสดงชื่อ Standard toolbar เป็นคำสั่งมาตรฐานของวินโดวและโปรแกรม Solidworks และเป็นที่อยู่

Command Manager ของแถบเครื่องมือของโปรแกรม Solidworks สามารถปรับเปลี่ยนได้ตามผู้ใช้ต้องการ

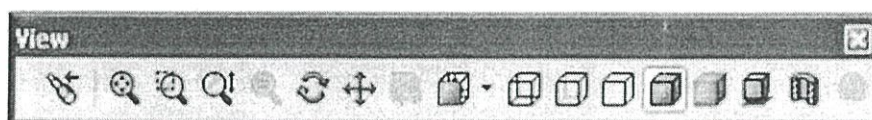
Feature Manager Design Tree เป็นที่อยู่ของ Browser จะแสดงประวัติขั้นตอนการทำงานว่าใช้เครื่องมือชนิดใด วิธีใดมาบ้าง และพร้อมที่จะแก้ไขได้ตลอดเวลา

Graphics Window เป็นพื้นที่ที่ใช้เขียนแบบ 2 มิติ และ 3 มิติ

Status Bar เป็นตำแหน่งที่แสดงข้อความบอกให้ทราบว่า您当前ทำงานถึงขั้นตอนไหนและเป็นสิ่งที่คอยอธิบายให้คุณทำอะไรต่อไป














2.6.1.3 มุมมองของวัตถุในโปรแกรม Solidworks

มุมมองหรือการมองภาพแบบต่างๆในโปรแกรม solidworks จะช่วยอำนวยความสะดวกในการมองภาพและการเขียนแบบคำสั่งที่ใช้แสดงมุมมองแบบต่างๆ (Viewtool) จะอยู่บนทูลบาร์มาตรฐานดังรูปที่ 2.14



รูปที่ 2.14 ทูลบาร์มาตรฐานของโปรแกรม Solidworks [6]

ต่อไปนี้จะเป็นการอธิบายถึงความหมายและวิธีการใช้ไอคอนมุมมองต่างๆ (Viewtool) จะอยู่บนทูลบาร์มาตรฐานดังตารางต่อไปนี้
 ตารางที่ 2.4 ความหมายและวิธีการใช้ไอคอนมุมมองต่างๆ (Viewtool) บนทูลบาร์มาตรฐาน

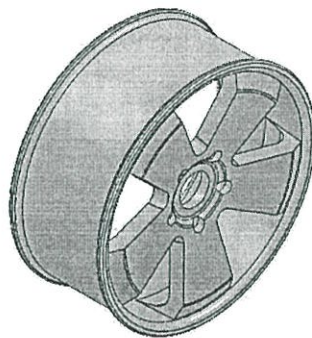
ไอคอน	คำสั่ง	ความหมายและวิธีการใช้งาน
	Previous view	แสดงภาพของวัตถุในมุมมองก่อนหน้าที่แสดงอยู่ในปัจจุบัน โดยการคลิกที่ไอคอน
	Zoom to fit	แสดงภาพของวัตถุทั้งหมด ตามขอบเขตของจอภาพโดยการคลิกที่ไอคอน
	Zoom to area	ขยายภาพของวัตถุบางส่วนตามกรอบของหน้าต่างที่เลือก โดยการคลิกที่ไอคอนเลือกตำแหน่งที่ต้องการขยายโดยการสร้างกรอบหน้าต่างครอบตำแหน่งนั้นๆ
	Zoom In-out	ขยายหรือย่อภาพของวัตถุ โดยคลิกที่ไอคอนหรือหมุนปุ่มกลางของเมาส์เข้า-ออก
	Zoom to selection	ขยายพื้นผิวที่เลือก โดยการเลือกพื้นผิวของภาพแล้วคลิกที่ไอคอน
	Rolate view	การหมุนวัตถุโดยการ กดปุ่มกลางของเมาส์ค้างไว้แล้วหมุนได้ตามความต้องการ
	Pan	เคลื่อนย้ายวัตถุไปยังตำแหน่งที่ต้องการโดยการคลิกเมาส์ปุ่มซ้ายที่วัตถุค้างไว้เลื่อนไปยังตำแหน่งที่ต้องการ
	Standard view	การเปลี่ยนมุมมองไปยังมุมมองมาตรฐาน
	Normal to	การเปลี่ยนมุมมองของภาพให้ขนานกับระนาบหรือพื้นผิวที่ราบเรียบ
	Wire Frame	การให้วัตถุแสดงเฉพาะเส้นเต็ม
	Hidden lines visible	การให้วัตถุแสดงเส้นที่มองเห็นเฉพาะเส้นเต็ม ส่วนเส้นที่มองไม่เห็นจะแสดงเป็นเส้นประ
	Hidden lines removed	แสดงเฉพาะเส้นที่มองเห็นเท่านั้น ส่วนเส้นที่มองไม่เห็นจะแสดงเป็นเส้นประ
	Shaded with Edges	การให้แสงเงาแก่วัตถุและให้แสดงเส้นที่ขอบรูปทุกเส้น

2.6.1.4 วิธีการสร้างชิ้นส่วน 3 มิติด้วยโปรแกรม Solidworks

การสร้างชิ้นส่วน 3 มิติด้วยโปรแกรม solidworks ส่วนใหญ่จะมีขั้นตอนในการสร้างอยู่ 4 ขั้นตอนคือ

1. สเกตซ์ภาพด้วยเส้นหรือภาพรูปทรงเรขาคณิตเป็นภาพ 2 มิติ
2. กำหนดความสัมพันธ์ทางเรขาคณิตให้ภาพสเกตซ์นั้นๆ
3. กำหนดขนาดให้ภาพสเกตซ์
4. สร้างเนื้อของชิ้นส่วนให้เป็น 3 มิติโดยใช้ฟีเจอร์ต่างๆ (Feature)

2.6.2 ตัวอย่างชิ้นงานที่ได้จากการออกแบบด้วยโปรแกรม Solidwork



รูปที่ 2.15 ชิ้นงานจากการออกแบบด้วย Solidworks [6]

2.7 มอเตอร์ไฟฟ้ากระแสตรง (Direct Current Motor)

2.7.1 ความหมายของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรง เป็นต้นกำลังขับเคลื่อนที่สำคัญอย่างหนึ่งในโรงงานอุตสาหกรรมเพราะมีคุณสมบัติที่โดดเด่นในด้านการปรับความเร็วได้ตั้งแต่ความเร็วต่ำสุดจนถึงสูงสุด นิยมใช้กันมากในโรงงานอุตสาหกรรม เช่น โรงงานทอผ้า โรงงานเส้นใยโพลีเอสเตอร์ โรงงานถลุงโลหะหรือให้ เป็นต้นกำลังในการขับเคลื่อนรถไฟไฟฟ้า เป็นต้นในการศึกษาเกี่ยวกับมอเตอร์ไฟฟ้ากระแสตรงจึงควรรู้จัก อุปกรณ์ต่าง ๆ ของมอเตอร์ไฟฟ้ากระแสตรงและเข้าใจถึงหลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบต่าง ๆ แบ่งออกเป็น 3 ชนิดคือ มอเตอร์แบบอนุกรมหรือที่เรียกว่าซีรีย์มอเตอร์ (Series Motor), มอเตอร์แบบอนุขนานหรือที่เรียกว่าชันทมอเตอร์ (Shunt Motor) และมอเตอร์ไฟฟ้าแบบผสมหรือที่เรียกว่าคอมปาวด์มอเตอร์ (Compound Motor)

2.7.1.1 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงมีส่วนประกอบที่สำคัญ 2 ส่วนดังนี้

1. ส่วนที่อยู่กับที่หรือที่เรียกว่าสเตเตอร์ (Stator) ประกอบด้วย

1.1 เฟรมหรือโยค (Frame or Yoke) เป็นโครงภายนอกทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กจากขั้วเหนือไปขั้วใต้ให้ครบวงจรและยึดส่วนประกอบอื่นๆให้แข็งแรงทำด้วยเหล็กหล่อหรือเหล็กแผ่นหนาหมุนเป็นรูปทรงกระบอก

1.2 ขั้วแม่เหล็ก (Pole) ประกอบด้วย 2 ส่วนคือแกนขั้วแม่เหล็กและขดลวด ส่วนแรกแกนขั้ว (Pole Core) ทำด้วยแผ่นเหล็กบางๆ กั้นด้วยฉนวนประกบกันเป็นแท่งยึดติดกับเฟรม ส่วนปลายที่ทำเป็นรูปโค้งนั้นเพื่อโค้งรับรูปกลมของตัวโรเตอร์เรียกว่า ขั้วแม่เหล็ก (Pole Shoes) มีวัตถุประสงค์ให้ขั้วแม่เหล็กและโรเตอร์ใกล้ชิดกันมากที่สุดเพื่อให้เกิดช่องอากาศน้อยที่สุด เพื่อให้เกิดช่องอากาศน้อยที่สุดจะมีผลให้เส้นแรงแม่เหล็กจากขั้วแม่เหล็กจากขั้วแม่เหล็กผ่านไปยังโรเตอร์มากที่สุดแล้วทำให้เกิดแรงบิดหรือกำลังบิดของโรเตอร์มากเป็นการทำให้มอเตอร์ มีกำลังหมุน (Torque) และส่วนที่สอง ขดลวดสนามแม่เหล็ก (Field Coil) จะพันอยู่รอบๆแกนขั้วแม่เหล็กขดลวดนี้ทำหน้าที่รับกระแสจากภายนอกเพื่อสร้างเส้นแรงแม่เหล็กให้เกิดขึ้น และเส้นแรงแม่เหล็กนี้จะเกิดการหักล้างและเสริมกันกับสนามแม่เหล็กของอาเมเจอร์ทำให้เกิดแรงบิดขึ้น

2. ตัวหมุน (Rotor) ตัวหมุนหรือเรียกว่าโรเตอร์ตัวหมุนนี้ทำให้เกิดกำลังงานมีแกนวางอยู่ในตลับลูกปืน (Ball Bearing) ซึ่งประกอบอยู่ในแผ่นปิดหัวท้าย (End Plate) ของมอเตอร์ประกอบด้วย 4 ส่วนด้วยกัน คือ

2.1 แกนเพลลา (Shaft)

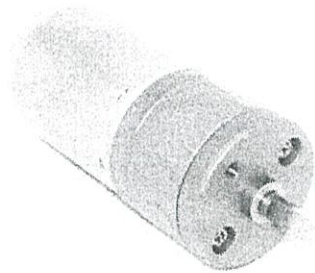
2.2 แกนเหล็กอาร์มาเจอร์ (Armature Core)

2.3 คอมมิวเตเตอร์ (Commutator)

2.4 ขดลวดอาร์มาเจอร์ (Armature Winding)

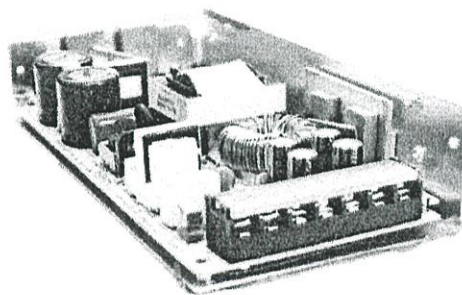
2.7.1.2 หลักการของมอเตอร์ไฟฟ้ากระแสตรง

หลักการของมอเตอร์ไฟฟ้ากระแสตรง (Motor Action) เมื่อเป็นแรงดันกระแสไฟฟ้าตรงเข้าไปในมอเตอร์ ส่วนหนึ่งจะ แปรงผ่านผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์มาเจอร์สร้างสนามแม่เหล็กขึ้น และกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field coil) สร้างขั้วเหนือ-ใต้ขึ้น จะเกิดสนามแม่เหล็ก 2 สนาม ในขณะเดียวกัน ตามคุณสมบัติของเส้นแรงแม่เหล็ก จะไม่ตัดกันทิศทางตรงข้ามจะหักล้างกัน และทิศทางเดียวจะเสริมแรงกัน ทำให้เกิดแรงบิดในตัวอาร์มาเจอร์ ซึ่งวางแกนเพลลาและแกนเพลลานี้ สวมอยู่กับตลับลูกปืนของมอเตอร์ ทำให้อาร์มาเจอร์นี้หมุนได้ ขณะที่ตัวอาร์มาเจอร์ทำหน้าที่หมุนได้นี้เรียกว่า โรเตอร์ (Rotor) ซึ่งหมายความว่าตัวหมุน การที่อำนาจเส้นแรงแม่เหล็กทั้งสองมีปฏิกิริยาต่อกัน ทำให้ขดลวดอาร์มาเจอร์ หรือโรเตอร์หมุนไปนั้นเป็นไปตามกฎซ้ายของเฟลมมิ่ง (Fleming's left hand rule)



รูปที่ 2.16 มอเตอร์ไฟฟ้ากระแสตรง [7]

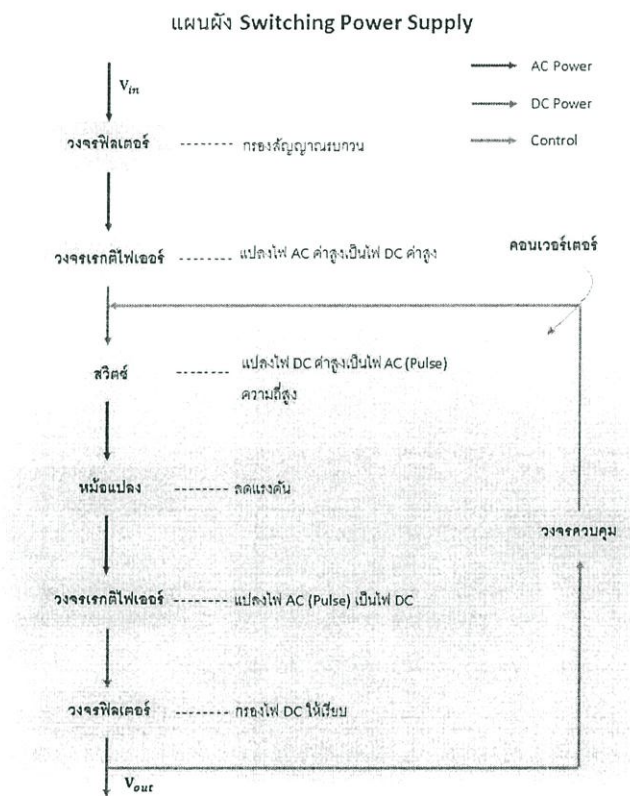
2.8 สวิตซ์เพาเวอร์ซัพพลาย (Output Switching Power Supply)



รูปที่ 2.17 วงจรสวิตซ์เพาเวอร์ซัพพลาย [8]

สวิตซ์เพาเวอร์ซัพพลายนั้นถูกสร้างขึ้นมาเพื่อใช้ในงานอิเล็กทรอนิกส์ ซึ่งเป็นแหล่งจ่ายไฟให้กับอุปกรณ์ต่างๆ และสามารถเปลี่ยนแรงดันไฟจากไฟสลับโวลต์สูงให้เป็นแรงดันไฟตรงโวลต์ต่ำได้ ซึ่งองค์ประกอบพื้นฐานของสวิตซ์เพาเวอร์ซัพพลายนั้นโดยทั่วไปจะคล้ายกันและสิ่งที่สำคัญที่สุดขององค์ประกอบนี้คือ คอนเวอร์เตอร์ สวิตซ์เพาเวอร์ซัพพลาย จะประกอบด้วย 3 ส่วนใหญ่ๆ คือ

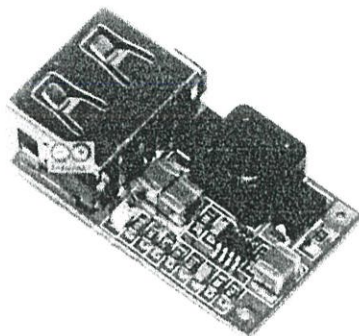
- 1) วงจรฟิลเตอร์และเรกติไฟเออร์ ทำหน้าที่แปลงแรงดันไฟสลับเป็นไฟตรง
- 2) คอนเวอร์เตอร์ ทำหน้าที่แปลงไฟตรงเป็นไฟสลับความถี่สูง และแปลงกลับเป็นไฟตรงโวลต์ต่ำ
- 3) วงจรควบคุม ทำหน้าที่ควบคุมการทำงานของคอนเวอร์เตอร์ เพื่อให้ได้แรงดันเอาต์พุตตามต้องการ



รูปที่ 2.18 แผนผังสวิตชิงเพาเวอร์ซัพพลาย [8]

การคงค่าแรงดันจะทำได้โดยการป้อนค่าแรงดันที่ Output กลับมายังวงจรควบคุม เพื่อควบคุมให้การนำกระแสมากขึ้นหรือน้อยลงตามการเปลี่ยนแปลงของแรงดันที่ Output ซึ่งจะมีผลทำให้แรงดัน Output คงที่ได้

2.9 โมดูลแปลงไฟ 6-24V เป็น 5V แบบ USB



รูปที่ 2.19 โมดูลแปลงไฟ 5V [9]

โมดูลแปลงไฟ 6-24V เป็น 5V แบบ USB จ่ายไฟต่อเนื่องที่ 2A กระแสสูงสุด 3A มี วงจรป้องกันจ่ายไฟ input ผิดขั้ว และเอาต์พุตลัดวงจร

2.9.1 Module information

1) มีการออกแบบใหม่, ออกแบบวงจรไฟฟ้าซึ่งไม่ต้องใช้ตัวเก็บประจุ electrolytic, Tantalum (electrolytic อายุใช้งานต่ำและ Tantalum เสี่ยงต่อการระเบิด) ปัจจุบันด้วยการใช้งาน วงจรไฟฟ้าแบบ high-end ของตัวเก็บประจุ solid ceramic ที่ใหญ่ 3 Ke

2) Ripple ต่ำมาก หลายๆบริษัทสามารถทำ Ripple ได้ถึง 500 mV หรือมากกว่า อุปกรณ์นี้จะให้ผลลัพธ์ในรูปแบบที่เห็นได้ชัด สาเหตุที่อุปกรณ์ไม่ทำงานหรือได้รับความเสียหาย

3) โปรแกรมใช้วงจร synchronous rectification ซึ่งประสิทธิภาพการแปลงค่อนข้าง สูง ในรอบ 5V 0.7A 6.5V เมื่อประสิทธิภาพสูงถึง 97.5% และ ในรอบ 12V 5V 1A จะไปถึง 94% ซึ่งแตกต่างจาก peers ซึ่งมีประสิทธิภาพเพียง 60-70% (ซึ่งประสิทธิภาพนี้ได้กำจัดการรับค่า การ ป้องกัน Diode แบบ Reverse polarity และ Fuse drop)

4) การใช้งาน Flat Copper ต่อกับตัวเหนี่ยวนำจะช่วยเพิ่มประสิทธิภาพการแปลง และลดความร้อน

5) ระบบเครือข่ายทั้งหมดที่ปราศจากการป้องกันแรงดันเกิน มี input fuse มี 1.5A, 300w, output 5V จะป้องกัน TVS ด้วย Clamp protection เพื่อป้องกัน USB

6) buck peek USB Module ไม่มีการป้องกันแรงดันเกิน, ซึ่งเป็นการป้องกันความ เสี่ยงอย่างมากให้กับ USB

7) ใช้ USB ล่าสุดเพื่อที่จะจำแนกวงจรไฟฟ้า

8) แผ่นสีทอง ของ USB ขอบที่ short head , มีการอายุการใช้งานที่ยาวนาน ซึ่ง USB interface ธรรมดาไม่สามารถเปรียบเทียบได้

9) แรงดัน output เป็นตัวชี้วัดสีแดง

10) ขนาดเล็กพิเศษ 26.4x15x7.4 มม.

11) รับค่า input ด้วย reverse polarity protection ไดโอด, reverse ด้านบวกด้านลบจะไม่ได้รับความเสียหาย

12) กระแส Quiescent ต่ำมาก, ต่อเข้ากับแบตเตอรี่รถยนต์โดยไม่ต้องใช้สวิตช์สามารถต่อสายหรือ ACC power cable ได้

2.9.2 Module parameters

1) คุณลักษณะของ Module : ไม่ได้แยกแรงดัน (BUCK) Synchronous rectifier

2) แรงดัน input : 6-24 VDC

3) แรงดัน output : 5.1-5.2 V

4) กระแส Output : สูงสุดที่ 3 A (มีการทำความเย็นร่วมด้วย) , แรงดันทดสอบ 12 V จะให้กระแส 2 A โดยไม่มีการทำความเย็น

5) ประสิทธิภาพการแปลงสูงสุดถึง 97.5 % (ค่าการทดสอบหลังจาก shorting และ reverse polarity protection fuse)

6) ความถี่สวิตซิ่ง 500 kHz

7) Ripple output 10mV (12V รอบ 5V 3A) 20M Bandwidth

8) สถานะแรงดัน output คือสีแดง

9) ย่านการทำงาน : มาตรฐานอุตสาหกรรม -40 oC ถึง +85 oC (อุณหภูมิยิ่งสูง power output ยิ่งต่ำ

10) แรงดัน over-voltage output: ใช้ input fuse 1.5A ,5V และ output 300 W การป้องกัน Clamp TVS tube

11) ภาระทางไฟฟ้าสูงสุดทำให้อุณหภูมิเพิ่มไปถึง 30 องศาเซลเซียส

12) กระแส static 0.85 Haoan

13) Load Regulation: +-1%

14) Voltage Regulation

15) ความเร็วการตอบสนอง Dynamic : 5% 200 us

16) output ป้องกันการลัดวงจร, ไม่ควรป้องกันนานมาก

17) input การป้องกัน reverse polarity: มีการป้องกัน reverse polarity Diode

18) การเชื่อมต่อ : สายไฟ

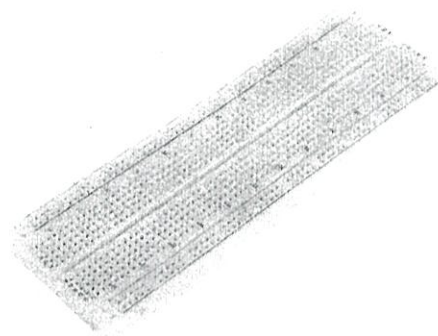
19) รับค่า input : สายไฟ

20) Output : Short head boundless USB

2.10 โพรโทบอร์ด (Protoboard)

Protoboard หรืออาจเรียกว่า Breadboard คือบอร์ดที่มีลักษณะเป็นแผ่นพลาสติกหนา สีขาวบนแผ่นมีรูเรียงกันจำนวนมาก ภายในรูมีตัวนำไฟฟ้าซึ่งเชื่อมต่อกันในรูปแบบที่มีการกำหนดไว้มากมาย โดยแต่ละรูที่อยู่ในแถวเดียวกันจะมีการเชื่อมต่อกันอยู่ภายในเมื่อเรานำสายไฟสองเส้นมาเสียบลงบน Protoboard ตรงตำแหน่งของรูที่อยู่ในแถวแนวอนเดียวกัน จะทำให้สายไฟทั้งสองเส้นนั้นเชื่อมต่อกัน ถือว่าเป็นสายไฟเส้นเดียวกัน เวลาทดลองก็เสียบขาของอุปกรณ์อิเล็กทรอนิกส์ลงไปเพื่อให้ตัวนำภายในเชื่อมวงจรถึงกัน และอาจใช้สายไฟเสียบลงรูเพื่อเชื่อมวงจรไฟฟ้าได้เช่นกัน

Protoboard ถือเป็นอุปกรณ์สำคัญที่จะช่วยให้เราออกแบบวงจรอิเล็กทรอนิกส์ได้ง่ายขึ้น เพราะการใช้งาน Protoboard ไม่ต้องใช้การบัดกรีวงจร เพียงแค่ใช้สายไฟเสียบลงไปที่รูบนบอร์ดก็สามารถเชื่อมต่อจุดในวงจรได้ตามที่เราต้องการ แต่มีข้อเสียคือเมื่อใช้ทดลองวงจรที่ทำงานที่ความถี่สูงๆ จะทำให้เกิดปัญหาเรื่องสัญญาณรบกวนในวงจร



รูปที่ 2.20 โพรโทบอร์ด [10]

2.11 สายไฟจัมพ์เปอร์

สายไฟสำเร็จรูปเป็นอุปกรณ์ที่ใช้ส่งพลังงานไฟฟ้าจากที่หนึ่งไปยังอีกที่หนึ่งโดยกระแสไฟฟ้าจะเป็นตัวนำพลังงานไฟฟ้าผ่านไปตามสายไฟจนถึงอุปกรณ์อีกชนิดหนึ่ง สายไฟทำด้วยสารที่ยอมให้กระแสไฟฟ้าผ่านได้ เรียกว่าตัวนำไฟฟ้า และตัวนำไฟฟ้าที่ใช้ทำสายไฟเป็นโลหะที่ยอมให้กระแสไฟฟ้าผ่านได้ดี ลวดตัวนำแต่ละชนิดยอมให้กระแสไฟฟ้าผ่านได้ต่างกัน ตัวนำไฟฟ้าที่ยอมให้กระแสไฟฟ้าผ่านได้มากเรียกว่ามีความนำไฟฟ้ามาก ลวดตัวนำจะมีความต้านทานไฟฟ้าอยู่ด้วย โดยลวดตัวนำที่มีความต้านทานไฟฟ้ามากจะยอมให้กระแสไฟฟ้าผ่านได้น้อยเป็นสายไฟที่ใช้กับแรงดันไม่เกิน 750 V เป็นสายหุ้มฉนวน ทำด้วยทองแดงหรืออลูมิเนียม โดยทั่วไปเป็นสายทองแดงสายขนาดเล็กจะเป็นตัวนำเดี่ยววัสดุฉนวนที่ใช้กับสายแรงดันต่ำคือ POLYVINYL CHLORIDE (PVC) และ CROSS-LINKED POLYETHYLENE (XLPE)

ขนาดเล็กจะเป็นตัวนำเดี่ยววัสดุฉนวนที่ใช้กับสายแรงดันต่ำคือ POLYVINYL CHLORIDE (PVC) และ CROSS-LINKED POLYETHYLENE (XLPE)



รูปที่ 2.21 สายไฟจัมพ์เปอร์สำเร็จรูป [11]

บทที่ 3

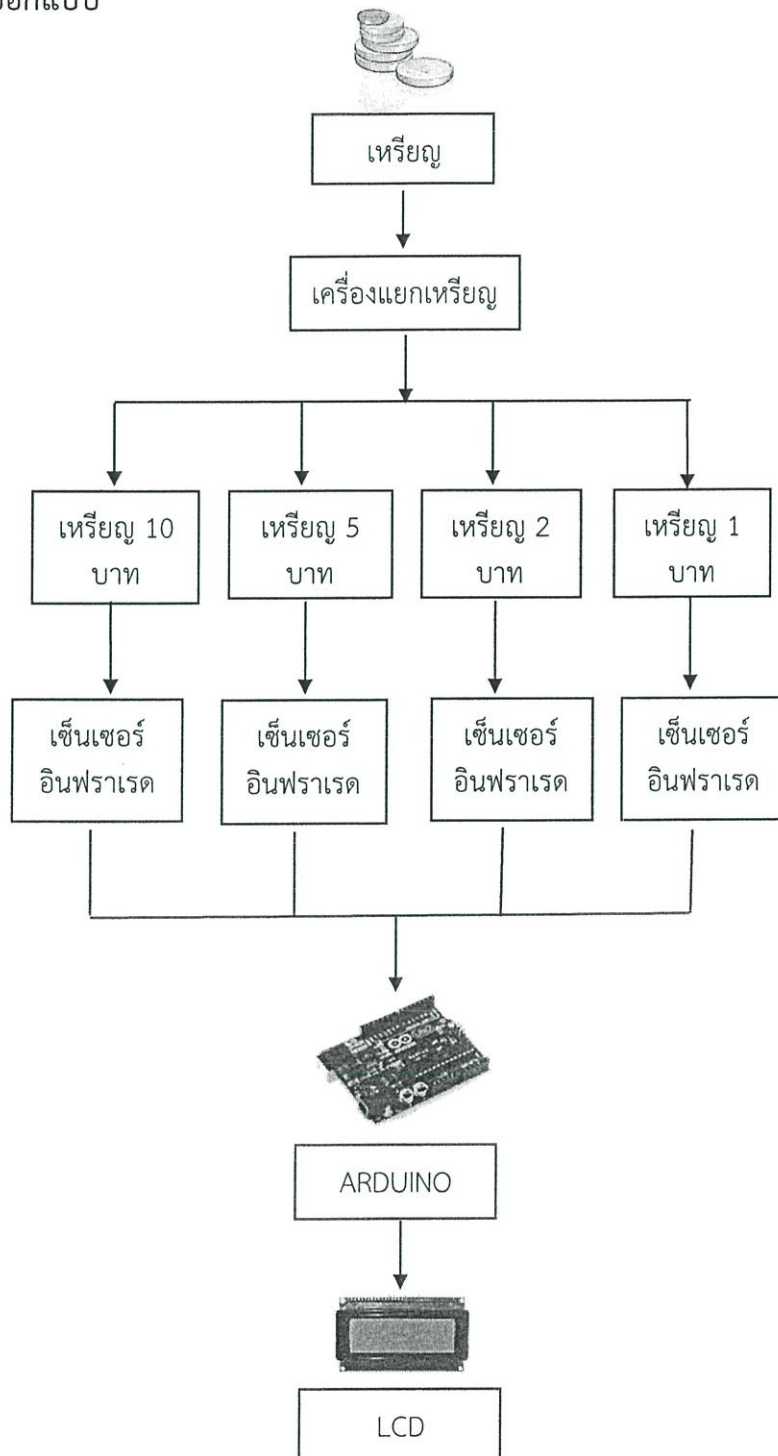
การออกแบบและการจัดทำปฏิญานินพธ์

การออกแบบและจัดทำปฏิญานินพธ์“เครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์”
จะแบ่งการออกแบบเป็นสองส่วนหลัก คือ

1. ฮาร์ดแวร์ (HARDWARE)
2. ซอฟต์แวร์ (SOFTWARE)

เครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์ มีวัตถุประสงค์เพื่อทำการแยกชนิดของเหรียญและนับเหรียญ โดยมีจอ LCD แสดงค่าของจำนวนเหรียญแต่ละชนิดและค่าของเหรียญที่ได้ทำการนับ ทำให้สามารถทราบจำนวนของเหรียญแต่ละชนิด และมูลค่าของเหรียญที่ได้ทำการนับทั้งหมด

3.1 การออกแบบ



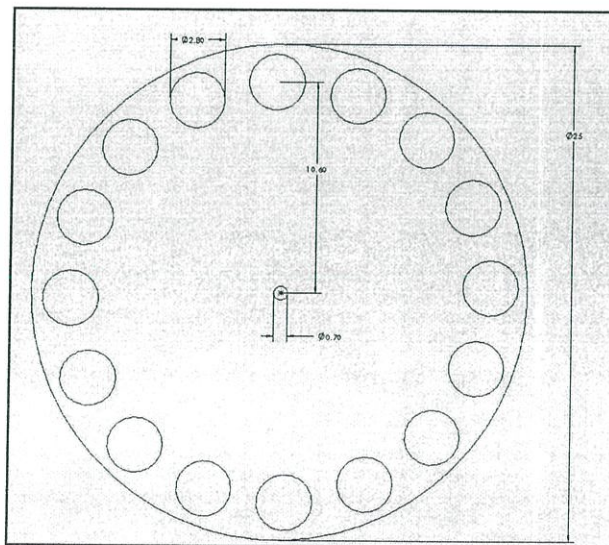
รูปที่ 3.1 บล็อกไดอะแกรมรวมของเครื่องนับเหรียญที่ใช้ไมโครคอนโทรลเลอร์

จาก Block Diagram มีการนำ Arduino มาประยุกต์ใช้ในการทำงานร่วมกับเซ็นเซอร์อินฟราเรด เพื่อใช้เป็นระบบการแยกชนิดของเหรียญและนับเหรียญ โดยจะทำการแยกชนิดของเหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท ส่วนของการนับจำนวนของเหรียญจะใช้เซ็นเซอร์อินฟราเรดเป็นตัวตรวจจับจำนวนของเหรียญที่ผ่านการแยกชนิด โดยเซ็นเซอร์อินฟราเรดจะทำงานร่วมกันกับระบบประมวลผลของ Arduino จากนั้น Arduino จะทำการประมวลผลคิดค่าของเงินที่ได้ทำการแยก แล้วแสดงผลออกทางจอ LCD โดยทำการแสดงค่าของจำนวนเหรียญแต่ละชนิด และค่าของเหรียญที่ได้ทำการนับ

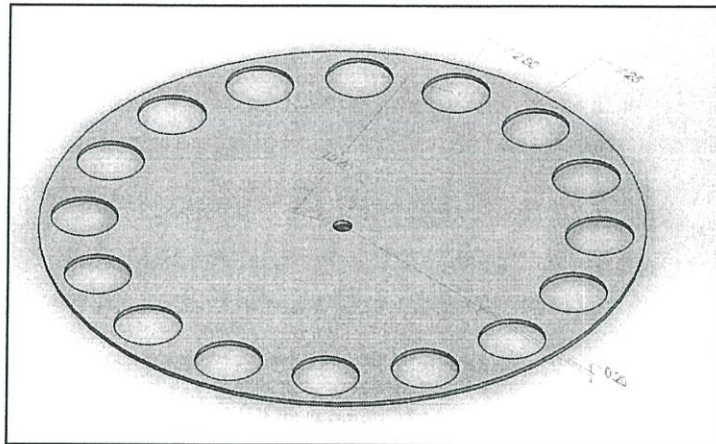
3.1.1 การออกแบบด้านฮาร์ดแวร์

การออกแบบด้านฮาร์ดแวร์ของปริญญาานิพนธ์นี้ เป็นการออกแบบส่วนของเครื่องนับเหรียญ โดยเครื่องนับเหรียญจะทำการแยกชนิดของเหรียญแต่ละชนิดคือ เหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท

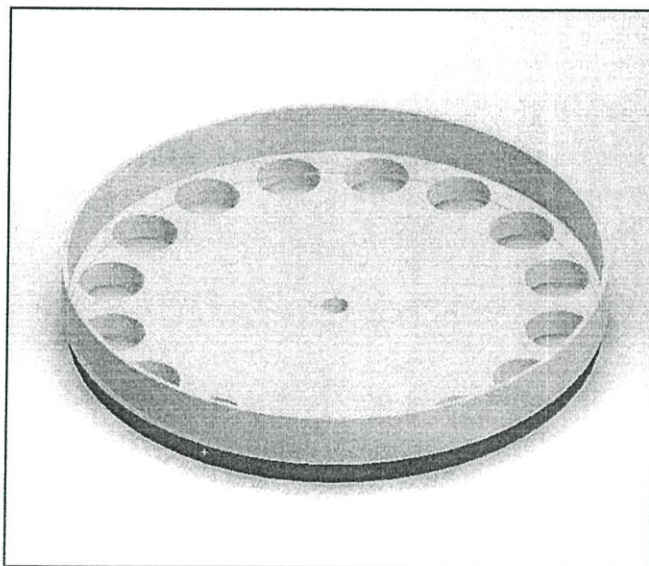
ทำการออกแบบจำลองอุปกรณ์ในส่วนของการแยกเหรียญ โดยใช้โปรแกรมที่เรียกว่า "Solidworks" โดยจะออกแบบเป็น 2 ส่วนคือ ในส่วนของช่องที่ไว้สำหรับจัดให้เหรียญอยู่ในรูปแบบที่เป็นลักษณะแนวนอน และอีกส่วนสำหรับการแยกชนิดของเหรียญทั้ง 4 ชนิดคือ เหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท โดยกำหนดเส้นผ่านศูนย์กลางของจานหมุนคือ 25 เซนติเมตร เส้นผ่านศูนย์กลางของช่องเล็กๆที่ไว้แยกเหรียญออกจากกันและเหรียญอยู่ในรูปแบบที่เป็นลักษณะแนวนอนคือ 2.80 เซนติเมตร



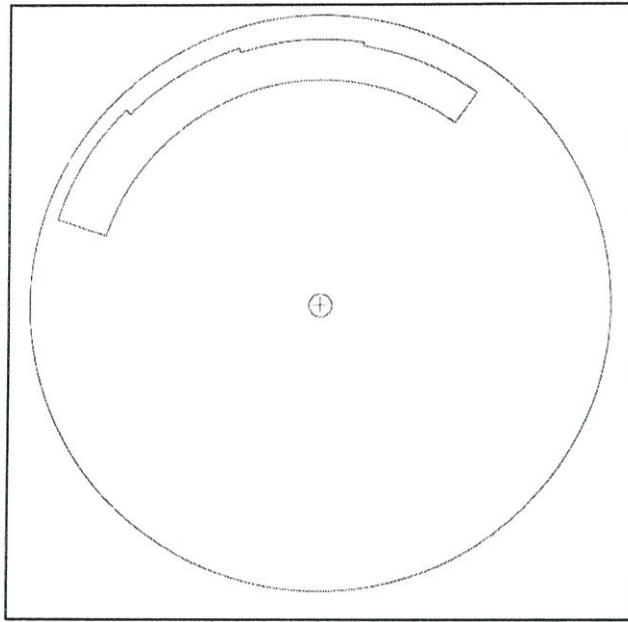
รูปที่ 3.2 การออกแบบส่วนการแยกเหรียญแบบ 2 มิติ



รูปที่ 3.3 การออกแบบส่วนการแยกเหรียญแบบ 3 มิติ

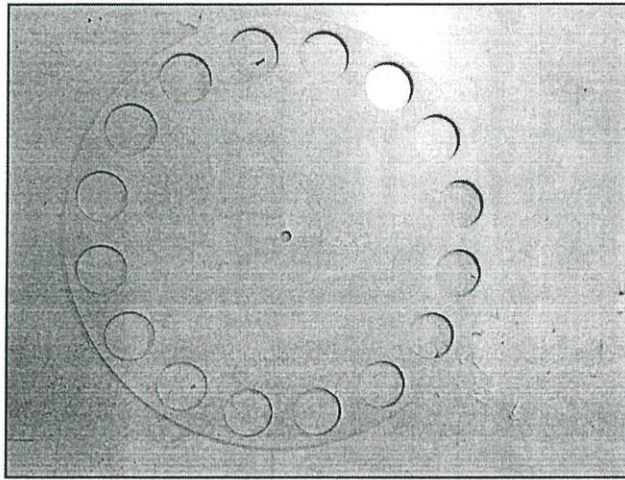


รูปที่ 3.4 การออกแบบส่วนการแยกเหรียญแบบ 3 มิติ แบบมีขอบกั้นเหรียญ

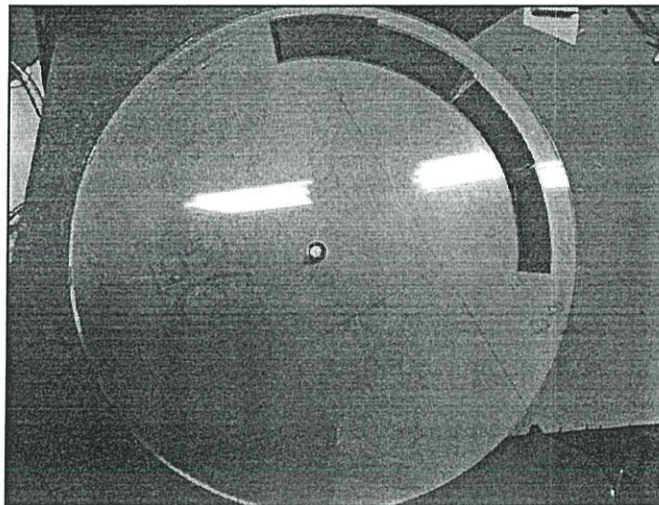


รูปที่ 3.5 การออกแบบส่วนการแยกเหรียญทั้ง 4 ชนิด คือ เหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท

สร้างแบบจำลองอุปกรณ์ในส่วนของการแยกเหรียญ โดยเริ่มทำในส่วนของงานหมุนแบ่งออกเป็น 2 ส่วนหลักๆ คือ ในส่วนของช่องที่ไว้สำหรับจัดให้เหรียญอยู่ในรูปแบบที่เป็นลักษณะแนวนอน และอีกส่วนสำหรับการแยกชนิดของเหรียญทั้ง 4 ชนิดคือเหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท

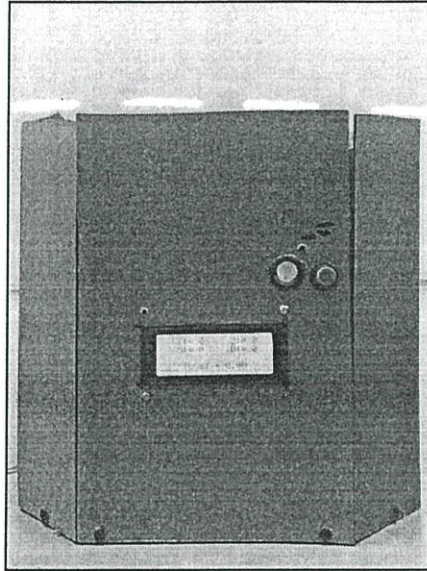


รูปที่ 3.6 ชิ้นส่วนแบบจำลองอุปกรณ์การแยกเหรียญในส่วนของจานหมุน (1)

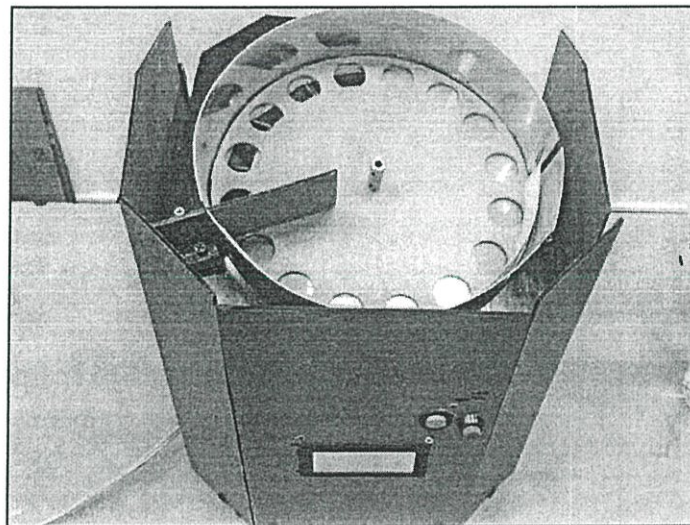


รูปที่ 3.7 ชิ้นส่วนแบบจำลองอุปกรณ์การแยกเหรียญในส่วนของจานหมุน (2)

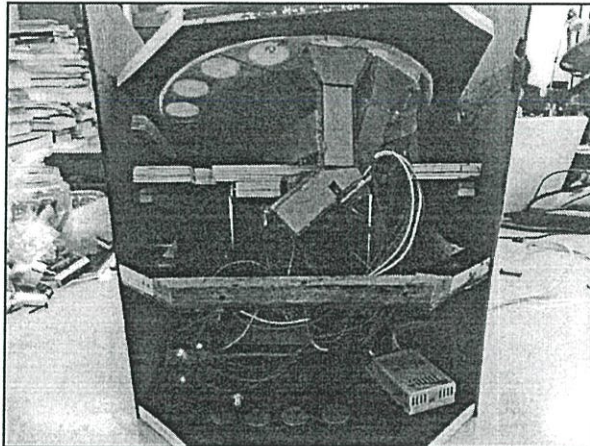
ทำเครื่องนับเหรียญ เริ่มทำในส่วนของฐานเครื่องนับเหรียญออกแบบเป็นลักษณะของฐานทรงแปดเหลี่ยม ทำส่วนของรางเหรียญโดยทำทั้งหมด 4 อันแบ่งตามขนาดของเหรียญแต่ละชนิด คือเหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท ทำส่วนของลิ้นชักที่เก็บเหรียญแต่ละชนิดไว้ และประกอบเครื่องนับเหรียญ



รูปที่ 3.8 ส่วนแสดงผลของเครื่องนับเหรียญ



รูปที่ 3.9 ส่วนการทำงานของเครื่องนับเหรียญ



รูปที่ 3.10 ส่วนการประมวลผลของเครื่องนับเหรียญ และส่วนของการแยกเหรียญ

3.1.2 การออกแบบด้านซอฟต์แวร์

การออกแบบด้านซอฟต์แวร์ใช้โปรแกรม ARDUINO เป็นโปรแกรมที่ใช้เขียนภาษา C ลงในตัวไมโครคอนโทรลเลอร์ที่เรียกว่า ARDUINO ซึ่งการเขียนโปรแกรมนั้นจะเป็นการควบคุมการทำงานทั้งหมดของเครื่องนับเหรียญในส่วนของการนับค่าของเหรียญให้ทำงานเป็นไปตามที่กำหนดไว้

```
int LED = 13;
int isObstaclePin3 = 4, isObstaclePin4 = 5, isObstaclePin5 = 6, isObstaclePin6 = 7; // This is our input pin
int isObstacle3 = HIGH, isObstacle4 = HIGH, isObstacle5 = HIGH, isObstacle6 = HIGH; // HIGH MEANS NO OBSTACLE
int Coin1=0, Coin2=0, Coin5=0, Coin10=0, coin1=0, coin2=0, coin5=0, coin10=0;
float total=0.00;
int ledPin = 12;
int buttonPin = 9;
int buttonState = LOW;
int button=0;
LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
```

รูปที่ 3.11 คำสั่งในการกำหนดค่าตัวแปรเริ่มต้น

```

void loop() {
  isObstacle3 = digitalRead(isObstaclePin3);
  isObstacle4 = digitalRead(isObstaclePin4);
  isObstacle5 = digitalRead(isObstaclePin5);
  isObstacle6 = digitalRead(isObstaclePin6);
  buttonState = digitalRead(buttonPin);
  lcd.setCursor(0,0); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 0
  lcd.print("  1=");lcd.print(Coin1);
  lcd.setCursor(10,0); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 0
  lcd.print("  2=");lcd.print(Coin2);
  lcd.setCursor(0,1); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 1
  lcd.print("  5=");lcd.print(Coin5);
  lcd.setCursor(10,1); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 1
  lcd.print(" 10=");lcd.print(Coin10);
  lcd.setCursor(4,3); // เริ่มทำงานที่ตัวอักษรที่ 4 บรรทัดที่ 3
  lcd.print("Total = ");lcd.print((float)total);
}

```

รูปที่ 3.12 คำสั่งกำหนดค่าเริ่มต้นของการแสดงผล

```

void setup() {
  lcd.begin(20,4);
  pinMode(LED, OUTPUT);
  pinMode(isObstaclePin3, INPUT);
  pinMode(isObstaclePin4, INPUT);
  pinMode(isObstaclePin5, INPUT);
  pinMode(isObstaclePin6, INPUT);
  pinMode(ledPin, OUTPUT); //กำหนดให้ Pin 12 เป็น Output
  pinMode(buttonPin, INPUT); //กำหนดให้ Pin 9 เป็น Input
  Serial.begin(9600);
}

```

รูปที่ 3.13 คำสั่งกำหนดขาอินพุตของ arduino

คำสั่งที่ควบคุมการทำงานของเครื่องนับเหรียญในส่วนของการนับค่าของเหรียญนั้น เมื่อมีเหรียญผ่านเซ็นเซอร์กัมปูจะทำให้เซ็นเซอร์กัมปูทำงาน โดยการทำงานนั้นจะใช้คำสั่งไมโครคอนโทรลเลอร์และสามารถเขียนโปรแกรมคำสั่งภาษา C ได้ดังนี้

```

if(isObstacle3 == HIGH&& buttonState == LOW)
{
digitalWrite(ledPin, LOW);
Coin1=Coin1+1;
coin1=Coin1;

Serial.print("Count 1 : ");Serial.print(Coin1);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin1);

Serial.print("Count 2 : ");Serial.print(Coin2);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin2);

Serial.print("Count 5 : ");Serial.print(Coin5);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin5);

Serial.print("Count 10 : ");Serial.print(Coin10);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin10);

total=coin1+coin2+coin5+coin10;
Serial.print("Total : ");Serial.println(total);

lcd.setCursor(0,0); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 0
lcd.print(" 1:= ");lcd.print(Coin1);
lcd.setCursor(10,0); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 0
lcd.print(" 2:= ");lcd.print(Coin2);
lcd.setCursor(0,1); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 1
lcd.print(" 5:= ");lcd.print(Coin5);
lcd.setCursor(10,1); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 1
lcd.print(" 10:= ");lcd.print(Coin10);
lcd.setCursor(4,3); // เริ่มทำงานที่ตัวอักษรที่ 4 บรรทัดที่ 3
lcd.print("Total = ");lcd.print((float)total);
}

```

รูปที่ 3.14 คำสั่งในการทำงานเมื่อเซ็นเซอร์นับเหรียญหนึ่งบาท
แล้วคำนวณค่าโดยจำนวนเหรียญคูณกับ 1

```

if(isObstacle4 == HIGH && buttonState == LOW)
{
digitalWrite(ledPin, LOW);
Coin2=Coin2+1;
coin2=Coin2*2;

Serial.print("Count 1 : ");Serial.print(Coin1);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin1);

Serial.print("Count 2 : ");Serial.print(Coin2);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin2);

Serial.print("Count 5 : ");Serial.print(Coin5);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin5);

Serial.print("Count 10 : ");Serial.print(Coin10);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin10);

total=coin1+coin2+coin5+coin10;
Serial.print("Total : ");Serial.println(total);

lcd.setCursor(0,0); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 0
lcd.print(" 1:= ");lcd.print(Coin1);
lcd.setCursor(10,0); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 0
lcd.print(" 2:= ");lcd.print(Coin2);
lcd.setCursor(0,1); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 1
lcd.print(" 5:= ");lcd.print(Coin5);
lcd.setCursor(10,1); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 1
lcd.print(" 10:= ");lcd.print(Coin10);
lcd.setCursor(4,3); // เริ่มทำงานที่ตัวอักษรที่ 4 บรรทัดที่ 3
lcd.print("Total = ");lcd.print((float)total);
}

```

รูปที่ 3.15 คำสั่งในการทำงานเมื่อเซ็นเซอร์นับเหรียญสองบาท
แล้วคำนวณค่าโดยจำนวนเหรียญคูณกับ 2

```

if(isObstacle5 == HIGH && buttonState == LOW)
{
digitalWrite(ledPin, LOW);
Coin5=Coin5+1;
coin5=Coin5*5;

Serial.print("Count 1 : ");Serial.print(Coin1);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin1);

Serial.print("Count 2 : ");Serial.print(Coin2);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin2);

Serial.print("Count 5 : ");Serial.print(Coin5);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin5);

Serial.print("Count 10 : ");Serial.print(Coin10);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin10);

total=coin1+coin2+coin5+coin10;
Serial.print("Total : ");Serial.println(total);

lcd.setCursor(0,0); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 0
lcd.print(" 1:= ");lcd.print(Coin1);
lcd.setCursor(10,0); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 0
lcd.print(" 2:= ");lcd.print(Coin2);
lcd.setCursor(0,1); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 1
lcd.print(" 5:= ");lcd.print(Coin5);
lcd.setCursor(10,1); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 1
lcd.print(" 10:= ");lcd.print(Coin10);
lcd.setCursor(4,3); // เริ่มทำงานที่ตัวอักษรที่ 4 บรรทัดที่ 3
lcd.print("Total = ");lcd.print((float)total);
}

```

รูปที่ 3.16 คำสั่งในการทำงานเมื่อเซ็นเซอร์นับเหรียญห้าบาท
แล้วคำนวณค่าโดยจำนวนเหรียญคูณกับ 5

```

if(isObstacle6 == HIGH && buttonState == LOW)
{
digitalWrite(ledPin, LOW);
Coin1=Coin1+1;
coin1=Coin1*10;

Serial.print("Count 1 : ");Serial.print(Coin1);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin1);

Serial.print("Count 2 : ");Serial.print(Coin2);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin2);

Serial.print("Count 5 : ");Serial.print(Coin5);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin5);

Serial.print("Count 10 : ");Serial.print(Coin10);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin10);

total=coin1+coin2+coin5+coin10;
Serial.print("Total : ");Serial.println(total);

lcd.setCursor(0,0); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 0
lcd.print(" 1:= ");lcd.print(Coin1);
lcd.setCursor(10,0); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 0
lcd.print(" 2:= ");lcd.print(Coin2);
lcd.setCursor(0,1); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 1
lcd.print(" 5:= ");lcd.print(Coin5);
lcd.setCursor(10,1); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 1
lcd.print(" 10:= ");lcd.print(Coin10);
lcd.setCursor(4,3); // เริ่มทำงานที่ตัวอักษรที่ 4 บรรทัดที่ 3
lcd.print("Total = ");lcd.print((float)total);
}

```

รูปที่ 3.17 คำสั่งในการทำงานเมื่อเซ็นเซอร์นับเหรียญสิบบาท
แล้วคำนวณค่าโดยจำนวนเหรียญคูณกับ 10

```

else
{
  Serial.println("Clear");
  digitalWrite(LED, HIGH);
}

delay(3);
}

```

รูปที่ 3.18 คำสั่งในการกำหนดดีเลย์ของการทำงาน

```

if(buttonState == HIGH)
{

  Serial.print("Count 1 : ");Serial.print(Coin1);Serial.print("\t");
  digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin1);

  Serial.print("Count 2 : ");Serial.print(Coin2);Serial.print("\t");
  digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin2);

  Serial.print("Count 5 : ");Serial.print(Coin5);Serial.print("\t");
  digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin5);

  Serial.print("Count 10 : ");Serial.print(Coin10);Serial.print("\t");
  digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin10);

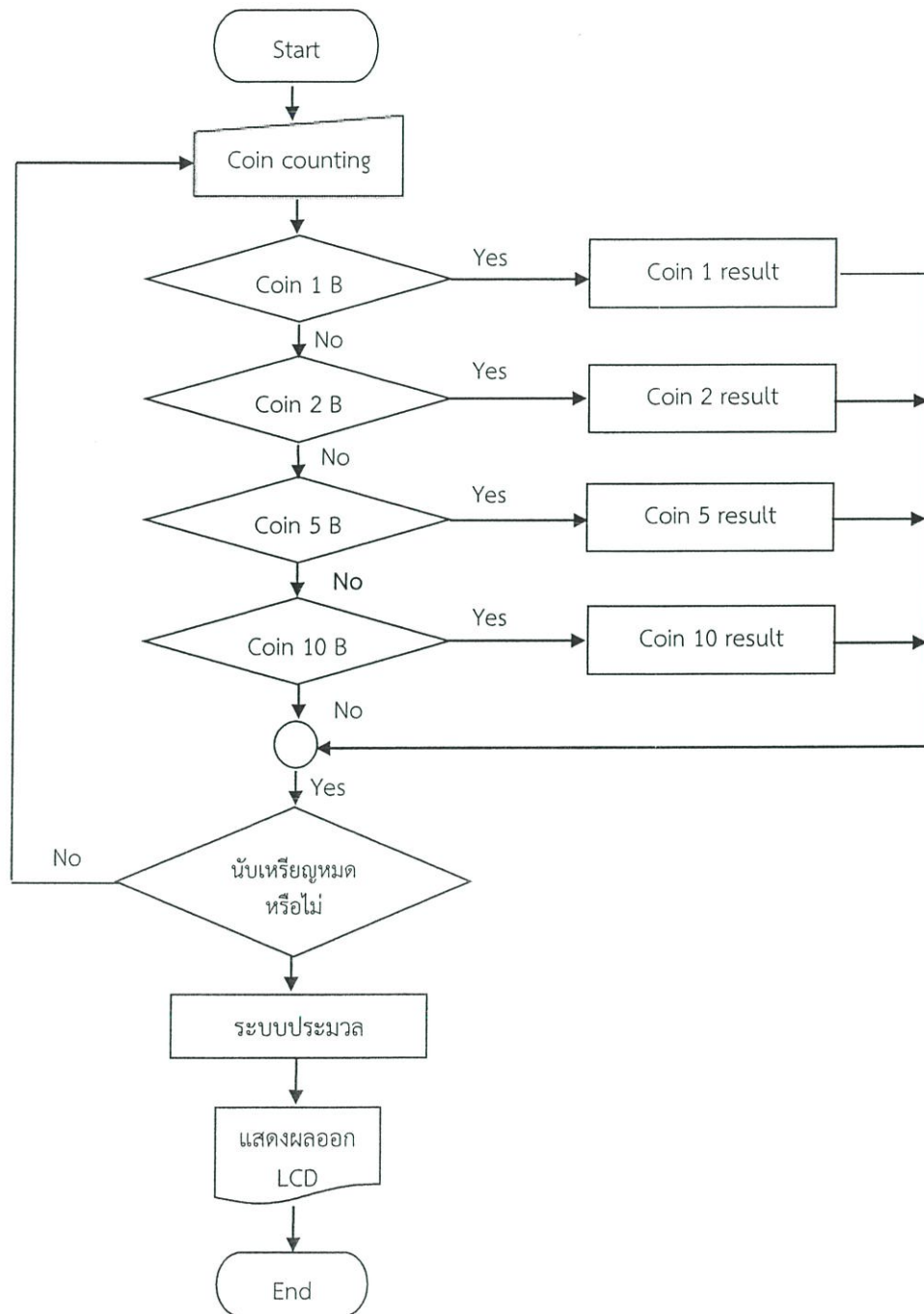
  total=coin1+coin2+coin5+coin10;
  Serial.print("Total : ");Serial.println(total);
}

```

รูปที่ 3.19 คำสั่งการ Reset การนับ

Flowchart การทำงานของเครื่องนับเหรียญ

การทำงานของ Flowchart เริ่มต้นการทำงานที่การนับจำนวนของเหรียญแต่ละชนิด โดย Infrared Sensor Optocoupler Modules หรือเซ็นเซอร์กัมพู เหรียญแต่ละเหรียญที่ผ่านเซ็นเซอร์จะทำการนับจำนวนไว้ เมื่อเซ็นเซอร์ทำการนับจำนวนของเหรียญเสร็จสิ้นแล้ว จากนั้นระบบจะทำการคำนวณมูลค่าของเหรียญรวมทั้งหมดที่นับได้ โดยระบบปฏิบัติการและประมวลผลของ Arduino จากนั้นจะทำการแสดงผลผ่านทางจอแสดงผล LCD แต่ถ้าหากการนับจำนวนของเหรียญแต่ละชนิดยังไม่เสร็จ ระบบก็จะทำการนับเหรียญต่อไปจนกว่าจะเสร็จ เมื่อเสร็จถึงจะเข้าสู่ระบบประมวลผลคำนวณมูลค่า และแสดงผลผ่านจอแสดงผล LCD เช่นเดียวกัน



รูปที่ 3.20 Flowchart การทำงานของเครื่องนับเหรียญ

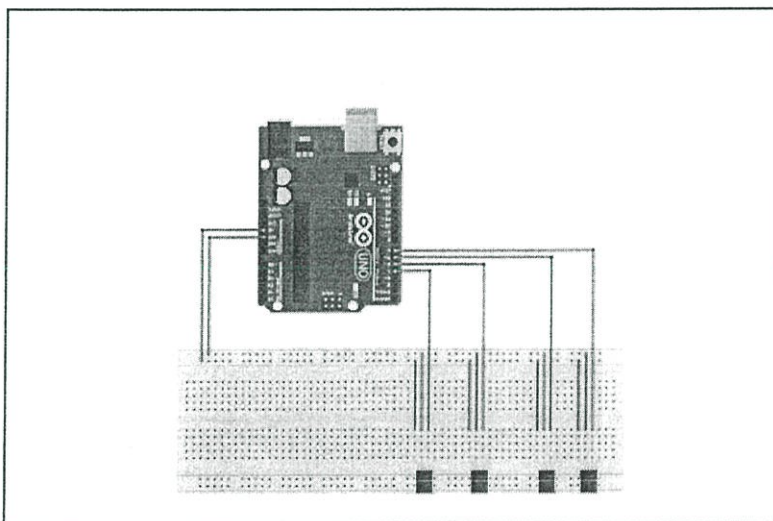
3.2 เครื่องมือที่ใช้ในการทดลอง

1. Arduino
2. Infrared Sensor Optocoupler Modules หรือเซ็นเซอร์กัมพู
3. จอ Liquid Crystal Display (LCD)
4. มอเตอร์ไฟฟ้ากระแสตรง
5. Protoboard
6. สายไฟจัมป์เปอร์
7. โมดูลแปลงไฟ 6-24V เป็น 5V แบบ USB
8. Switching Power Supply (12V, 2A)

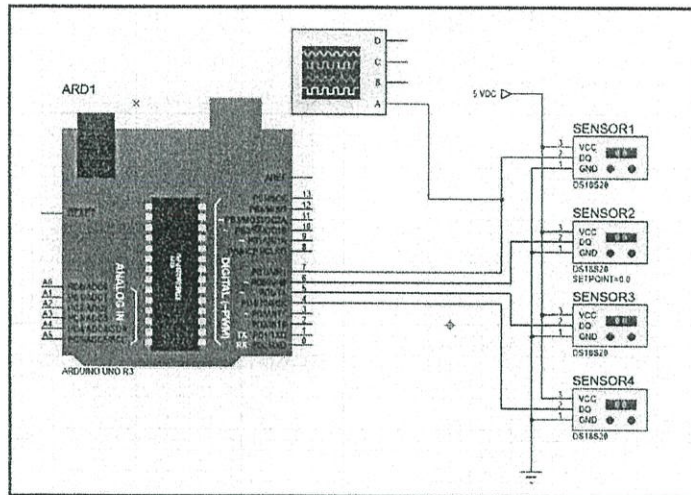
3.3 การจัดเก็บผลการทดลอง

3.3.1 ทดสอบการทำงานของเซ็นเซอร์กับ Arduino

ส่วนนี้จะทำการเชื่อมต่อ Infrared Sensor Optocoupler Modules หรือเซ็นเซอร์กัมพู กับ Arduino ทดสอบการทำงานของเซ็นเซอร์โดยใช้คำสั่ง สั่งการให้ ARDUINO รับค่าที่เซ็นเซอร์นับค่าได้มาประมวลผลแล้วแสดงผลออก ที่ serial monitor โดยจะแสดงจำนวนของเหรียญค่าของจำนวนเหรียญแต่ละชนิด คือ เหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท และมูลค่าของเหรียญทั้งหมดที่ได้ทำการนับ



รูปที่ 3.21 ทดสอบการทำงานของเซ็นเซอร์กับ Arduino



รูปที่ 3.22 ทดสอบการทำงานของเซ็นเซอร์กับ Arduino โดยใช้ Oscilloscope

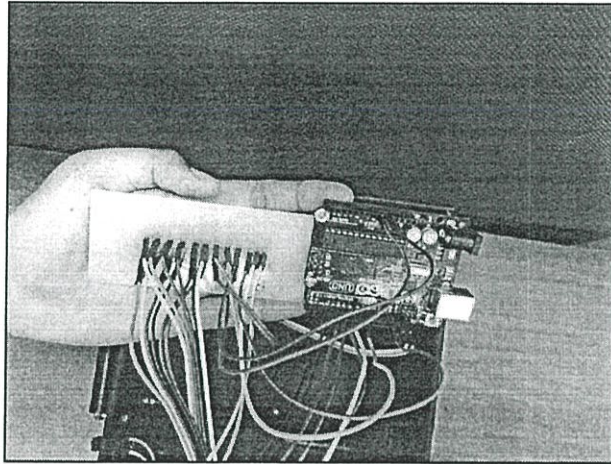
ทำการทดสอบความเร็วของเซ็นเซอร์เมื่อทำการนับเหรียญ โดยทดสอบความแม่นยำของ Infrared Sensor Optocoupler Modules กับการหมุนของมอเตอร์ไฟฟ้ากระแสตรง 12 V, 10 rpm และทำการทดสอบกระแสไฟฟ้าของมอเตอร์ว่าใช้กระแสไฟฟ้าเท่าไร

3.3.2 ทดสอบการทำงานของ จอ LCD

ส่วนนี้จะทำการเชื่อมต่อ Infrared Sensor Optocoupler Modules กับ Arduino ทำการทดสอบการทำงานของ LCD โดยใช้เซ็นเซอร์กัมพูทั้ง 4 ตัว จากนั้นใช้โปรแกรมเขียนคำสั่งสั่งการให้ Arduino รับค่าที่เซ็นเซอร์นับค่าได้นำค่าที่ได้มาประมวลผลแล้วแสดงผลออกทางจอ LCD

3.3.3 ทดสอบของระบบโดยรวม

ส่วนนี้จะเป็นการทดสอบส่วนสุดท้ายโดยนำส่วนต่างๆ มารวมกันให้สามารถทำงานร่วมกันได้ว่าเป็นระบบตามวัตถุประสงค์ที่ได้วางไว้ ซึ่งประกอบไปด้วยเชื่อมต่อ Infrared Sensor Optocoupler Modules จำนวน 4 ตัว, Arduino และจอ LCD



รูปที่ 3.23 เชื่อมต่อเซ็นเซอร์อินฟราเรด, Arduino และจอ LCD

```

COM3 (Arduino/Genuino Uno)

Clear
Clear
Clear
Clear
Clear
Clear
Count 1 : 3      Value: 3
Count 2 : 5      Value: 10
Count 5 : 3      Value: 15
Count 10 : 3     Value: 30
Total : 58.00
Clear
Clear

```

รูปที่ 3.24 ผลการทำงานเบื้องต้นของเซ็นเซอร์อินฟราเรดกับ Arduino

บทที่ 4

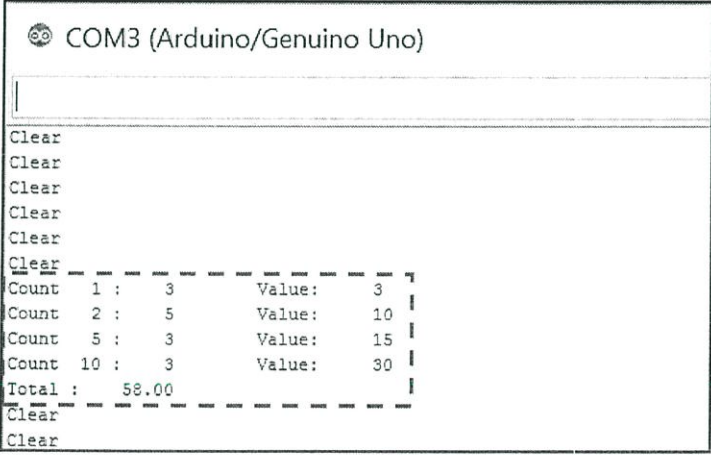
ผลการทดลอง

จากการออกแบบทั้งหมดในบทที่ 3 เมื่อทดสอบการทำงานของเครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์สามารถแสดงผลได้ในส่วน 2 ส่วน คือ ส่วนของซอฟต์แวร์และส่วนของฮาร์ดแวร์

4.1 ผลการทดลองส่วนของซอฟต์แวร์

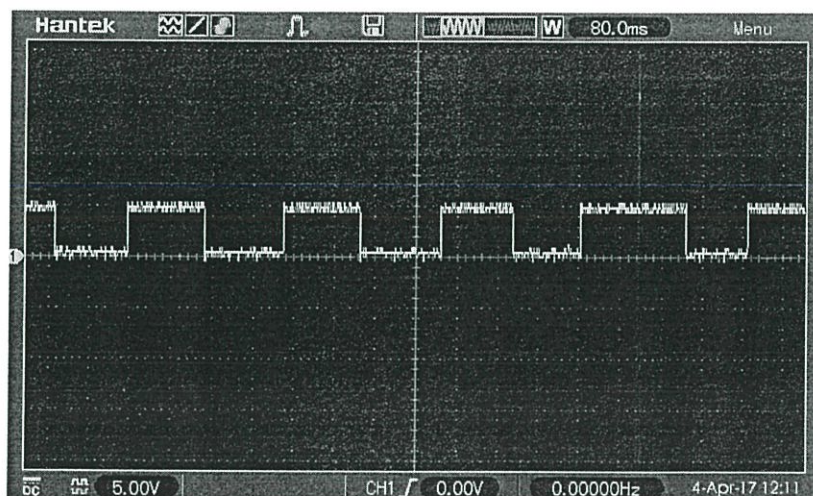
ผลการทดลองในส่วนของซอฟต์แวร์เป็นตัวแสดงผลการทำงานของโปรแกรมว่าเป็นไปตามหลักการที่ได้เขียนโปรแกรมลงในไมโครคอนโทรลเลอร์ ซึ่งผลลัพธ์การทำงานจะแสดงผลบนหน้าจอจอมอนิเตอร์ของโปรแกรม Arduino

เริ่มต้นจากการทดสอบการทำงานของ Infrared Sensor Optocoupler Modules หรือเซ็นเซอร์กัมพูแต่ละตัว จากนั้นทำการทดสอบการทำงานเซ็นเซอร์ทั้ง 4 ตัว โดยผลลัพธ์ที่แสดงบนจอมอนิเตอร์ของโปรแกรม ARDUINO คือแสดงจำนวนของเหรียญแต่ละชนิดค่าของจำนวนเหรียญแต่ละชนิดและมูลค่าของเหรียญทั้งหมดที่ได้ทำการนับ ดังตัวอย่างที่แสดงด้านล่าง



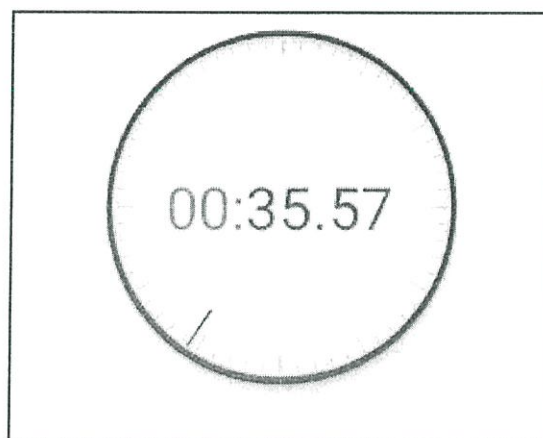
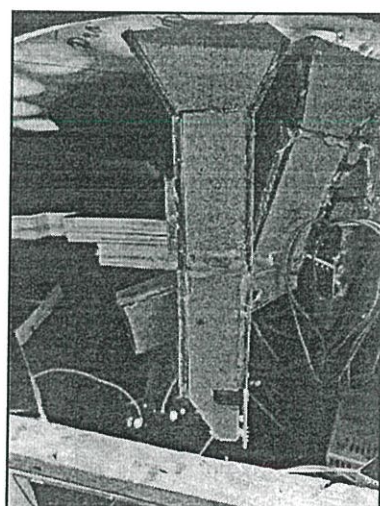
```
COM3 (Arduino/Genuino Uno)
Clear
Clear
Clear
Clear
Clear
Clear
Count 1 : 3      Value: 3
Count 2 : 5      Value: 10
Count 5 : 3      Value: 15
Count 10 : 3     Value: 30
Total : 58.00
Clear
Clear
```

รูปที่ 4.1 ค่าของจำนวนเหรียญแต่ละชนิดและมูลค่าของเหรียญที่ได้ทำการนับที่แสดงบนจอมอนิเตอร์ของโปรแกรม ARDUINO

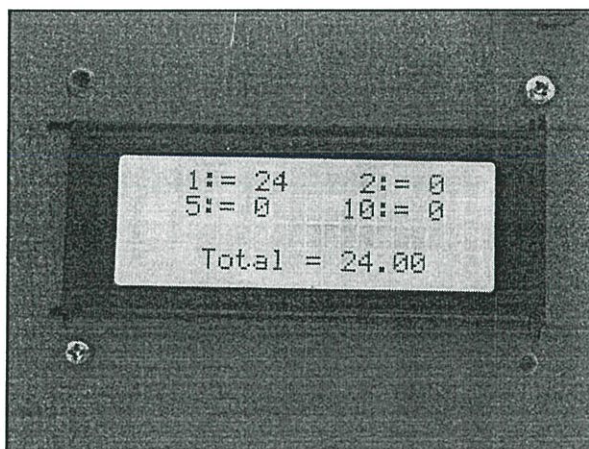


รูปที่ 4.2 สัญญาณจากการทดสอบการทำงานของเซ็นเซอร์กับ Arduino โดยใช้ Oscilloscope

ทดสอบความแม่นยำของ Infrared Sensor Optocoupler Modules หรือเซ็นเซอร์ กัมพู โดยทดสอบนับเหรียญบาทจำนวน 40 เหรียญกับการหมุนของมอเตอร์ไฟฟ้ากระแสตรง 12 V, 10 rpm พบว่าลักษณะของรางเหรียญมีผลต่อความแม่นยำของเซ็นเซอร์ เนื่องจากความเร็วของเหรียญที่ผ่านเซ็นเซอร์ ดังนั้นจึงออกแบบให้รางเหรียญมีลักษณะโค้งงอเพื่อชะลอความเร็วที่เหรียญแต่ละเหรียญจะไหลผ่านเซ็นเซอร์ได้ ดังตัวอย่างที่แสดงด้านล่าง

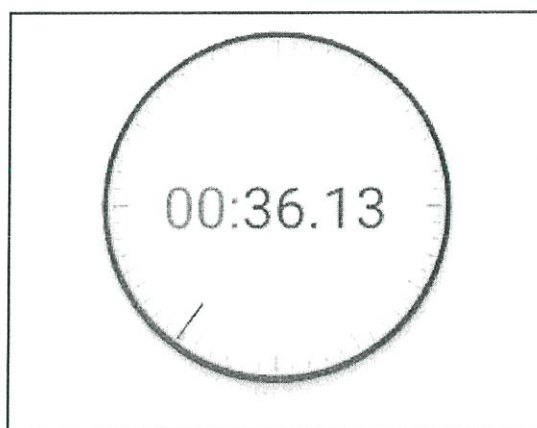


รูปที่ 4.3 รางเหรียญที่ไม่มีการชะลอความเร็ว และเวลาที่ใช้ในการนับเหรียญบาทจำนวน 40 เหรียญ

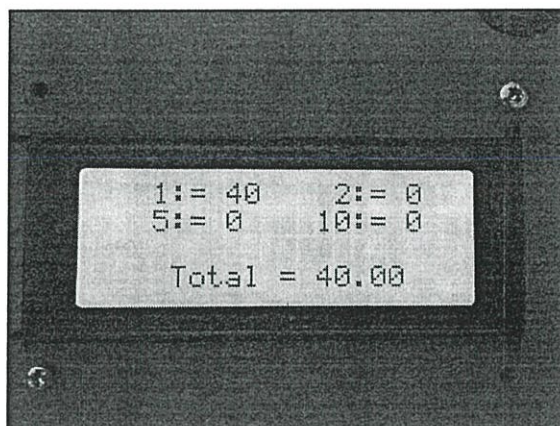


รูปที่ 4.4 ผลการทดสอบความแม่นยำของเซนเซอร์เมื่อนับ
เหรียญบาทจำนวน 40 เหรียญ (1)

จากรูปที่ 4.3 และ 4.4 พบว่าเมื่อทำการนับเหรียญบาท จำนวน 40 เหรียญ รางเหรียญที่ไม่มีการชะลอความเร็วสามารถนับเหรียญบาทได้เพียง 24 เหรียญเท่านั้น ใช้เวลาในการนับเหรียญ 35.57 วินาที ผลที่ได้จากการนับของเซนเซอร์มีค่าความผิดพลาด 40 เปอร์เซ็นต์



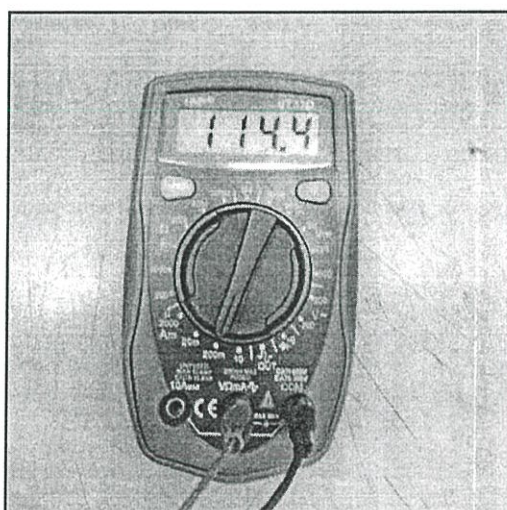
รูปที่ 4.5 รางเหรียญที่มีการชะลอความเร็ว และเวลาที่ใช้ในการนับ
เหรียญบาทจำนวน 40 เหรียญ



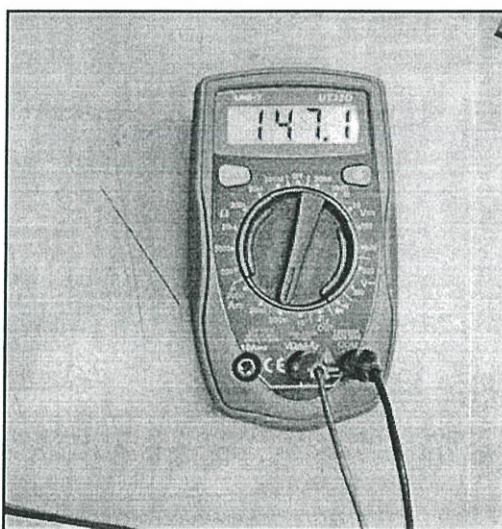
รูปที่ 4.6 ผลการทดสอบความแม่นยำของเซนเซอร์เมื่อนับเหรียญบาทจำนวน 40 เหรียญ (2)

จากรูปที่ 4.5 และ 4.6 พบว่าเมื่อทำการนับเหรียญบาท จำนวน 40 เหรียญ รางเหรียญที่มีการชะลอความเร็วสามารถนับเหรียญบาทได้จำนวนทั้งหมด 40 เหรียญ ใช้เวลาในการนับเหรียญ 36.13 วินาที ผลที่ได้จากการนับของเซนเซอร์ไม่มีค่าความผิดพลาด 0 เปอร์เซ็นต์

ทำการวัดกระแสไฟฟ้าของมอเตอร์เครื่องนับเหรียญ โดยทำการวัดค่ากระแสไฟฟ้าทั้งก่อนใส่เหรียญและหลังใส่เหรียญ โดยเปรียบเทียบค่ากระแสไฟฟ้าที่มอเตอร์ใช้ในขณะก่อนใส่เหรียญ พบว่า ก่อนใส่เหรียญมอเตอร์ใช้กระแสไฟฟ้าประมาณ 115-120 mA. และหลังใส่เหรียญบาทจำนวน 40 เหรียญ มอเตอร์ใช้กระแสไฟฟ้าประมาณ 140-150 mA.

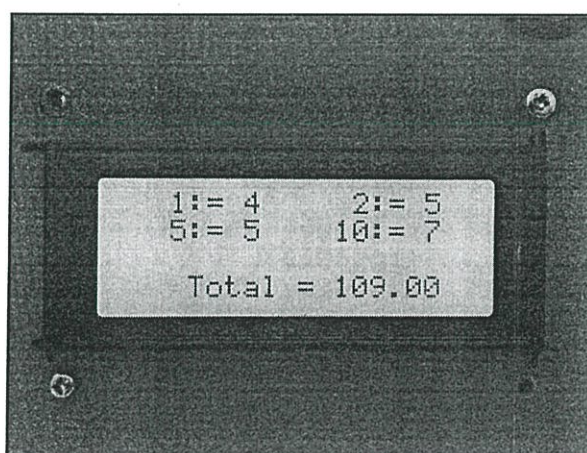


รูปที่ 4.7 ค่ากระแสไฟฟ้าที่มอเตอร์ใช้ในขณะก่อนใส่เหรียญ



รูปที่ 4.8 ค่ากระแสไฟฟ้าที่มอเตอร์ใช้ในหมุนหลังใส่เหรียญ

ทำการนับจำนวนเหรียญ โดยใช้ Infrared Sensor Optocoupler Modules หรือ เซ็นเซอร์กัมพูทำงานร่วมกับ ARDUINO โดยที่ ARDUINO จะทำการประมวลผลจากจำนวนเหรียญคูณกับมูลค่าของเหรียญแต่ละชนิด และทำการรวมมูลค่าของเหรียญทั้งหมดที่นับได้ โดยค่าทั้งหมดจะทำการแสดงผลผ่านทางจอ LCD ดังนั้นข้อมูลที่แสดงผลผ่านทางจอ LCD คือ จำนวนของเหรียญแต่ละชนิดและมูลค่าของเหรียญที่ได้ทำการนับทั้งหมด



รูปที่ 4.9 หน้าจอ LCD ของเครื่องนับเหรียญ

แต่ละชนิด ทำการเก็บผลการทดลองของ Infrared Sensor Optocoupler Modules และการนับค่าที่ผิดพลาดลงในตารางที่ 4.1

ตารางที่ 4.1 ทดสอบการทำงานของเครื่องนับเหรียญแบบแยกชนิดเหรียญ

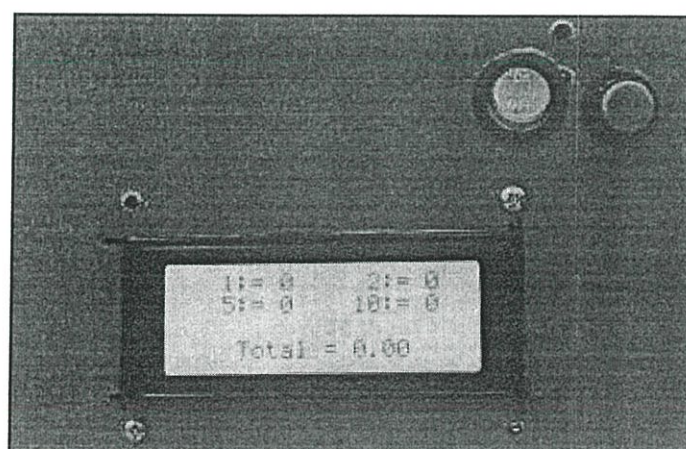
ชนิดเหรียญ	จำนวนเหรียญที่ทดสอบ (เหรียญ)		จำนวนเหรียญผิดพลาด (เหรียญ)
	จำนวนที่ใส่	จำนวนที่นับ	
เหรียญบาท	5	5	-
	10	10	-
	15	15	-
	20	20	-
เหรียญสองบาท	5	5	-
	10	10	-
	15	15	-
	20	20	-
เหรียญห้าบาท	5	5	-
	10	10	-
	15	15	-
	20	20	-
เหรียญสิบบาท	5	5	-
	10	9	1
	15	15	-
	20	19	1

ทดสอบการทำงานของเครื่องนับเหรียญ โดยทำการใส่เหรียญลงในเครื่องนับเหรียญจำนวน 20 เหรียญคือ เหรียญบาท 5 เหรียญ, เหรียญสองบาท 5 เหรียญ, เหรียญห้าบาท 5 เหรียญ และเหรียญสิบบาท 5 เหรียญ ทำการเก็บผลการทดลองของ Infrared Sensor Optocoupler Modules และการนับค่าที่ผิดพลาดลงในตารางที่ 4.2

ตารางที่ 4.2 ทดสอบการทำงานของเครื่องนับเหรียญแบบรวมชนิดเหรียญ

ชนิดเหรียญ		จำนวนเหรียญที่ทดสอบ (เหรียญ)		จำนวนเหรียญ ผิดพลาด (เหรียญ)
		จำนวนที่ใส่	จำนวนที่นับ	
ครั้งที่ 1	เหรียญบาท	5	5	-
	เหรียญสองบาท	5	5	-
	เหรียญห้าบาท	5	5	-
	เหรียญสิบบาท	5	4	1
ครั้งที่ 2	เหรียญบาท	5	5	-
	เหรียญสองบาท	5	5	-
	เหรียญห้าบาท	5	5	-
	เหรียญสิบบาท	5	5	-
ครั้งที่ 3	เหรียญบาท	5	5	-
	เหรียญสองบาท	5	5	-
	เหรียญห้าบาท	5	5	-
	เหรียญสิบบาท	5	5	-

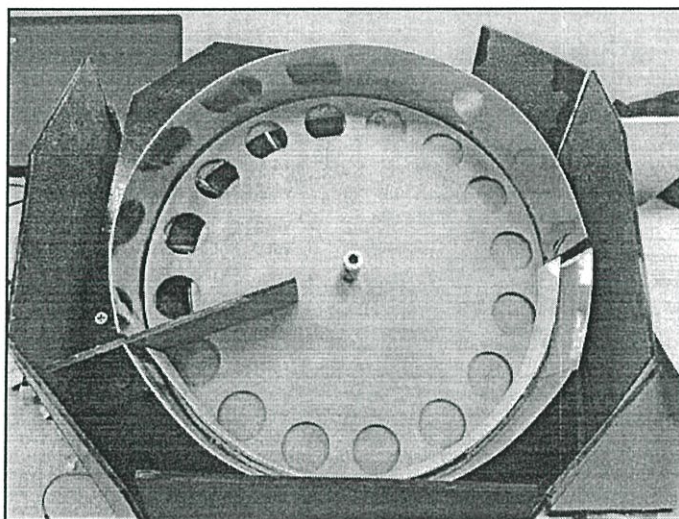
ส่วนของการนับเหรียญนั้นจะแสดงการรีเซ็ตค่าของเครื่องนับเหรียญเมื่อทำการกดปุ่มรีเซ็ต เพื่อพร้อมในการนับเหรียญครั้งต่อไป และสวิตช์เปิด-ปิด ควบคุมการทำงานของมอเตอร์ของเครื่องนับเหรียญ ดังแสดงในรูปที่ 4.10



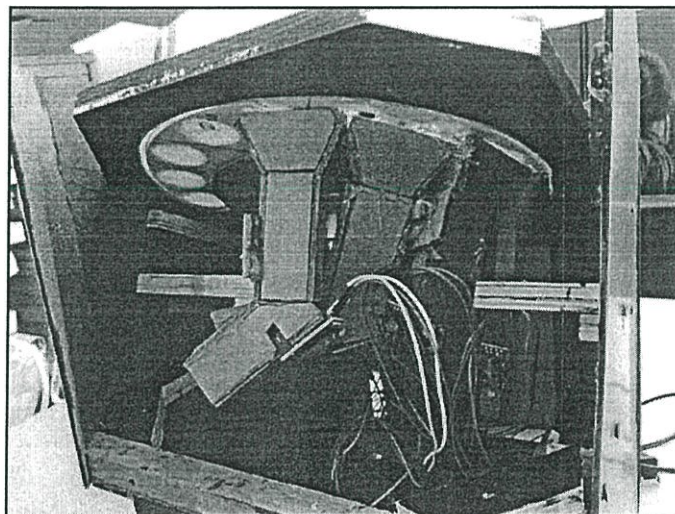
รูปที่ 4.10 ปุ่มรีเซ็ตและสวิตช์เปิด-ปิด ควบคุมมอเตอร์

4.2 ผลการทดลองส่วนของฮาร์ดแวร์

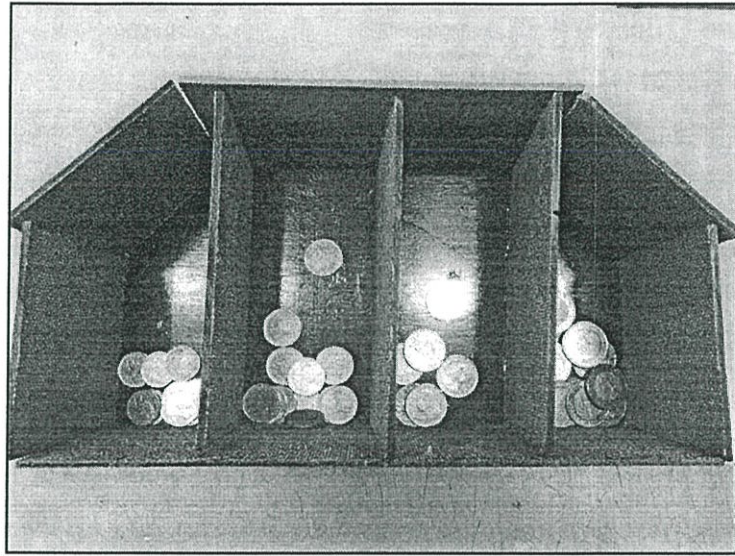
ผลการทดลองในส่วนของฮาร์ดแวร์เมื่อทำการสร้างเครื่องนับเหรียญในส่วนของการแยกเหรียญ โดยจานหมุนจะแบ่งออกเป็น 2 ส่วนหลักๆคือในส่วนของช่องที่ไว้สำหรับจัดให้ เหรียญอยู่ในรูปแบบที่เป็นลักษณะแวนอน และอีกส่วนสำหรับการแยกชนิดของเหรียญทั้ง 4 ชนิด คือ เหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท โดยเครื่องนับเหรียญสามารถแยกเหรียญแต่ละชนิดได้ตามช่องของเหรียญที่ได้กำหนดไว้



รูปที่ 4.11 ส่วนของการแยกเหรียญทั้ง 4 ชนิด



รูปที่ 4.12 ส่วนของการนับจำนวนเหรียญแต่ละชนิด และค่าของเหรียญที่ได้ทำการนับ



รูปที่ 4.13 ช่องของเหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์ชิ้นนี้นำเสนอเครื่องนับเหรียญโดยใช้ Infrared Sensor Optocoupler Modules หรือเซ็นเซอร์กัมพูทำงานร่วมกับไมโครคอนโทรลเลอร์ แบ่งเป็นสองส่วน คือส่วนของฮาร์ดแวร์และส่วนของซอฟต์แวร์ เริ่มจากส่วนของฮาร์ดแวร์ ทำการแยกชนิดของเหรียญทั้ง 4 ชนิด คือ เหรียญบาท, เหรียญสองบาท, เหรียญห้าบาทและเหรียญสิบบาท โดยใช้งานแยกชนิดของเหรียญ และในส่วนซอฟต์แวร์จะใช้ Infrared Sensor Optocoupler Modules หรือเซ็นเซอร์กัมพูทำหน้าที่นับจำนวนของเหรียญที่ผ่านรางเหรียญมาโดยถูกแยกชนิดของเหรียญมาแล้ว เมื่อ Infrared Sensor Optocoupler Modules ทำการนับจำนวนเหรียญ จากนั้น ARDUINO จะทำการประมวลผล ซึ่งทำการคำนวณจากจำนวนเหรียญคูณกับมูลค่าของเหรียญแต่ละชนิด และทำการรวมมูลค่าของเหรียญทั้งหมดที่นับได้ โดยค่าทั้งหมดจะทำการแสดงผลผ่านทางจอ LCD

จากการทดสอบการทำงานของเครื่องนับเหรียญ สามารถทำการแยกชนิดของเหรียญทั้ง 4 ชนิด คือ เหรียญบาท, เหรียญสองบาท, เหรียญห้าบาท และเหรียญสิบบาท สามารถนับจำนวนของเหรียญที่ผ่าน Infrared Sensor Optocoupler Modules และรวมมูลค่าของเหรียญทั้งหมดที่ทำการนับได้

5.2 ข้อเสนอแนะ

เครื่องนับเหรียญโดยใช้ไมโครคอนโทรลเลอร์ที่จัดทำขึ้นในโครงการนี้ เนื่องจากเป็นอุปกรณ์ที่มีราคาถูกและมีประสิทธิภาพต่ำ จึงส่งผลให้เกิดความผิดพลาดเล็กน้อย กล่าวคือในบางครั้ง Infrared Sensor Optocoupler Modules หรือเซ็นเซอร์กัมพูทำการนับเหรียญผิดพลาดเนื่องจากช่วงเวลาการตกของเหรียญทำให้เกิดการนับจำนวนเหรียญที่ผิดพลาดจึงส่งผลให้มูลค่ารวมของเหรียญผิดพลาดตามไปด้วย ดังนั้นควรเลือกใช้ Infrared Sensor Optocoupler Modules หรือเซ็นเซอร์กัมพูที่มีประสิทธิภาพสูงและความแม่นยำสูง และในส่วนของวัสดุโครงสร้างที่มีราคาถูกทำให้ไม่ได้มาตรฐานตามที่ควร จึงควรออกแบบโครงสร้างและวัสดุที่ได้มาตรฐาน แต่ก็จะมีราคาที่สูงขึ้นด้วย

บรรณานุกรม

- [1] Hestore. “ไมโครคอนโทรลเลอร์.”
http://www.hestore.hu/files/ARDUNIO_01.pdf.
- [2] Thaieasyelec. “Arduino.”
<http://www.thaieasyelec.com/article-wiki/basic-electronics/>.
- [3] Myarduino. “Infrared Sensor Optocoupler Modules.”
<http://www.myarduino.net/product/182/counter-module-motor-speed-sensor>.
- [4] Csscsmutt. “รังสีอินฟราเรด.”
http://www.csscsmutt.in.th/cssc/cssc_classroom/Solarenergy/Assignment/SolEn54/SolEn54_Doc/5_Infrared.pdf.
- [5] Thaieasyelec. “Liquid Crystal Display (LCD).”
<http://www.thaieasyelec.com/article-wiki/review-product-article>.
- [6] docs.google. “โปรแกรม Solidworks .”
<https://docs.google.com/>.
- [7] Tronixlabs. “การควบคุมมอเตอร์. ”
<http://tronixlabs.com/news/tutorial-l298n-dual-motor-controller-module-2a-and-arduino/>.
- [8] thaipowersupply “สวิตชิงเพาเวอร์ซัพพลาย.”
<http://www.thaipowersupply.com/manual/s-15-spec.pdf>.
- [9] arduinoall “โมดูลแปลงไฟ 6-24V เป็น 5V แบบ USB.”
<https://www.arduinoall.com/product>.
- [10] บ้านอิเล็กทรอนิกส์. “โปรโทบอร์ด.”
<http://www.semi-shop.com/knowledge/knowledge.php>.
- [11] Arduitrronics. “สายไฟจัมป์เปอร์สำเร็จรูป.”
<http://www.arduitronics.com/product/jumper-male-to-female>.

ภาคผนวก ก

CODE ภาษา C

```

#include "Wire.h"
#include "LCD.h"
#include "LiquidCrystal_I2C.h" //ประกาศ Library ของจอ I2C
#define I2C_ADDR 0x27
#define BACKLIGHT_PIN 3

int LED = 13;
int isObstaclePin3 = 4,isObstaclePin4 = 5,isObstaclePin5 = 6,isObstaclePin6 = 7;
// This is our input pin
int isObstacle3 = HIGH,isObstacle4 = HIGH,isObstacle5 = HIGH,isObstacle6 = HIGH; //
HIGH MEANS NO OBSTACLE
int Coin1=0,Coin2=0,Coin5=0,Coin10=0,coin1=0,coin2=0,coin5=0,coin10=0;
float total=0.00;
int ledPin = 12;
int buttonPin = 9;
int buttonState = LOW;
int button=0;
LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
void setup() {
  lcd.begin(20,4);
  pinMode(LED, OUTPUT);
  pinMode(isObstaclePin3, INPUT);
  pinMode(isObstaclePin4, INPUT);
  pinMode(isObstaclePin5, INPUT);
  pinMode(isObstaclePin6, INPUT);
  pinMode(ledPin, OUTPUT); //กำหนดให้ Pin 12 เป็น Output
  pinMode(buttonPin, INPUT); //กำหนดให้ Pin 9 เป็น Input
  Serial.begin(9600);
}
void loop() {
  isObstacle3 = digitalRead(isObstaclePin3);
  isObstacle4 = digitalRead(isObstaclePin4);
  isObstacle5 = digitalRead(isObstaclePin5);
  isObstacle6 = digitalRead(isObstaclePin6);

```

```

buttonState = digitalRead(buttonPin);
lcd.setCursor(0,0); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 0
lcd.print(" 1:= ");lcd.print(Coin1);
lcd.setCursor(10,0); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 0
lcd.print(" 2:= ");lcd.print(Coin2);
lcd.setCursor(0,1); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 1
lcd.print(" 5:= ");lcd.print(Coin5);
lcd.setCursor(10,1); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 1
lcd.print(" 10:= ");lcd.print(Coin10);
lcd.setCursor(4,3); // เริ่มทำงานที่ตัวอักษรที่ 1 บรรทัดที่ 3
lcd.print("Total = ");lcd.print((float)total);

if(isObstacle3 == HIGH&& buttonState == LOW)
{
digitalWrite(ledPin, LOW);
Coin1=Coin1+1;
coin1=Coin1;

Serial.print("Count 1 : ");Serial.print(Coin1);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin1);

Serial.print("Count 2 : ");Serial.print(Coin2);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin2);

Serial.print("Count 5 : ");Serial.print(Coin5);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin5);

Serial.print("Count 10 : ");Serial.print(Coin10);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin10);

total=coin1+coin2+coin5+coin10;
Serial.print("Total : ");Serial.println(total);

lcd.setCursor(0,0); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 0

```

```

lcd.print(" 1:= ");lcd.print(Coin1);
lcd.setCursor(10,0); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 0
lcd.print(" 2:= ");lcd.print(Coin2);
lcd.setCursor(0,1); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 1
lcd.print(" 5:= ");lcd.print(Coin5);
lcd.setCursor(10,1); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 1
lcd.print(" 10:= ");lcd.print(Coin10);
lcd.setCursor(4,3); // เริ่มทำงานที่ตัวอักษรที่ 4 บรรทัดที่ 3
lcd.print("Total = ");lcd.print((float)total);
}
if(isObstacle4 == HIGH && buttonState == LOW)
{
digitalWrite(ledPin, LOW);
Coin2=Coin2+1;
coin2=Coin2*2;

Serial.print("Count 1 : ");Serial.print(Coin1);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin1);

Serial.print("Count 2 : ");Serial.print(Coin2);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin2);

Serial.print("Count 5 : ");Serial.print(Coin5);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin5);

Serial.print("Count 10 : ");Serial.print(Coin10);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin10);

total=coin1+coin2+coin5+coin10;
Serial.print("Total : ");Serial.println(total);

lcd.setCursor(0,0); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 0
lcd.print(" 1:= ");lcd.print(Coin1);
lcd.setCursor(10,0); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 0

```

```

lcd.print(" 2:= ");lcd.print(Coin2);
lcd.setCursor(0,1); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 1
lcd.print(" 5:= ");lcd.print(Coin5);
lcd.setCursor(10,1); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 1
lcd.print(" 10:= ");lcd.print(Coin10);
lcd.setCursor(4,3); // เริ่มทำงานที่ตัวอักษรที่ 4 บรรทัดที่ 3
lcd.print("Total = ");lcd.print((float)total);
}
if(isObstacle5 == HIGH && buttonState == LOW)
{
digitalWrite(ledPin, LOW);
Coin5=Coin5+1;
coin5=Coin5*5;

Serial.print("Count 1 : ");Serial.print(Coin1);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin1);

Serial.print("Count 2 : ");Serial.print(Coin2);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin2);

Serial.print("Count 5 : ");Serial.print(Coin5);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin5);

Serial.print("Count 10 : ");Serial.print(Coin10);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin10);

total=coin1+coin2+coin5+coin10;
Serial.print("Total : ");Serial.println(total);

lcd.setCursor(0,0); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 0
lcd.print(" 1:= ");lcd.print(Coin1);
lcd.setCursor(10,0); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 0
lcd.print(" 2:= ");lcd.print(Coin2);
lcd.setCursor(0,1); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 1

```

```

lcd.print(" 5:= ");lcd.print(Coin5);
lcd.setCursor(10,1); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 1
lcd.print(" 10:= ");lcd.print(Coin10);
lcd.setCursor(4,3); // เริ่มทำงานที่ตัวอักษรที่ 4 บรรทัดที่ 3
lcd.print("Total = ");lcd.print((float)total);
}
if(isObstacle6 == HIGH && buttonState == LOW)
{
digitalWrite(ledPin, LOW);
Coin10=Coin10+1;
coin10=Coin10*10;

Serial.print("Count 1 : ");Serial.print(Coin1);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin1);

Serial.print("Count 2 : ");Serial.print(Coin2);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin2);

Serial.print("Count 5 : ");Serial.print(Coin5);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin5);

Serial.print("Count 10 : ");Serial.print(Coin10);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin10);

total=coin1+coin2+coin5+coin10;
Serial.print("Total : ");Serial.println(total);

lcd.setCursor(0,0); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 0
lcd.print(" 1:= ");lcd.print(Coin1);
lcd.setCursor(10,0); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 0
lcd.print(" 2:= ");lcd.print(Coin2);
lcd.setCursor(0,1); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 1
lcd.print(" 5:= ");lcd.print(Coin5);
lcd.setCursor(10,1); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 1

```

```

lcd.print(" 10:= ");lcd.print(Coin10);
lcd.setCursor(4,3); // เริ่มทำงานที่ตัวอักษรที่ 4 บรรทัดที่ 3
lcd.print("Total = ");lcd.print((float)total);
}

if(buttonState == HIGH)
{

Serial.print("Count 1 : ");Serial.print(Coin1);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin1);

Serial.print("Count 2 : ");Serial.print(Coin2);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin2);

Serial.print("Count 5 : ");Serial.print(Coin5);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin5);

Serial.print("Count 10 : ");Serial.print(Coin10);Serial.print("\t");
digitalWrite(LED, LOW);Serial.print("Value: ");Serial.println(coin10);

total=coin1+coin2+coin5+coin10;
Serial.print("Total : ");Serial.println(total);

lcd.setCursor(0,0); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 0
lcd.print(" 1:= ");lcd.print(Coin1);
lcd.setCursor(10,0); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 0
lcd.print(" 2:= ");lcd.print(Coin2);
lcd.setCursor(0,1); // เริ่มทำงานที่ตัวอักษรที่ 0 บรรทัดที่ 1
lcd.print(" 5:= ");lcd.print(Coin5);
lcd.setCursor(10,1); // เริ่มทำงานที่ตัวอักษรที่ 10 บรรทัดที่ 1
lcd.print(" 10:= ");lcd.print(Coin10);
lcd.setCursor(4,3); // เริ่มทำงานที่ตัวอักษรที่ 4 บรรทัดที่ 3
lcd.print("Total = ");lcd.print((float)total);
Coin1=0;Coin2=0;Coin5=0;Coin10=0;

```

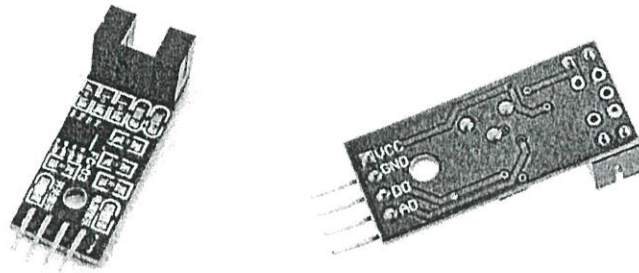
```
    coin1=0;coin2=0;coin5=0;coin10=0;total=00.00;
}
else
{
    Serial.println("Clear");
    digitalWrite(LED, HIGH);
}

delay(5);
}
```

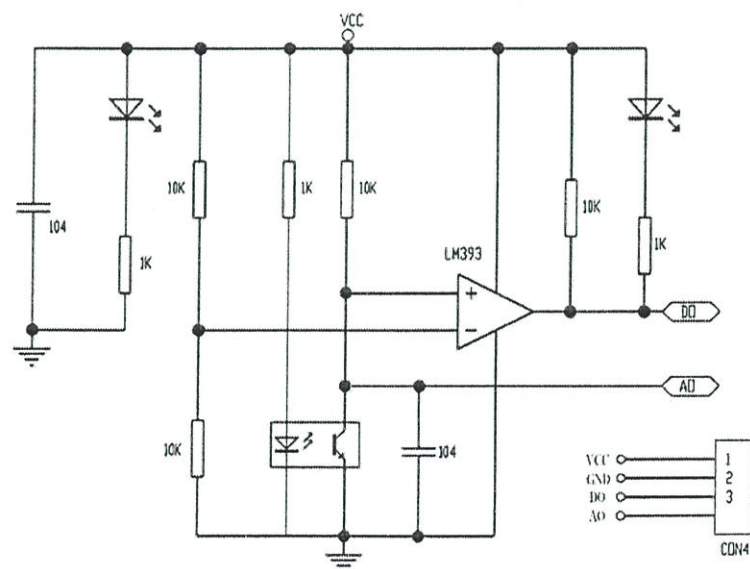
ภาคผนวก ข

DATASHEET

Infrared speed sensor module



1. Motor speed sensor or work piece counting
2. Using infrared detection, strong anti-jamming capability
3. Schmidt trigger, high sensitivity, stable performance
4. 5V Working voltage, black line output low level and white lines output high level
5. 74HC14 Hex inverting high performance Schmitt trigger chip
6. Photo interrupter: MOCH25A; slotted width : 1 cm



The LED (left) indicates the presence of voltage applied. When there is no object between the photo interrupter, the LED (right) is ON; otherwise is OFF. If slot unobstructed, DO output low; otherwise, DO output high.

Pins Connection:

VCC: positive power supply 3.3-5V;

GND: Ground;

DO: Output frequency pulses;

AO: analog output, real-time output voltage signal.

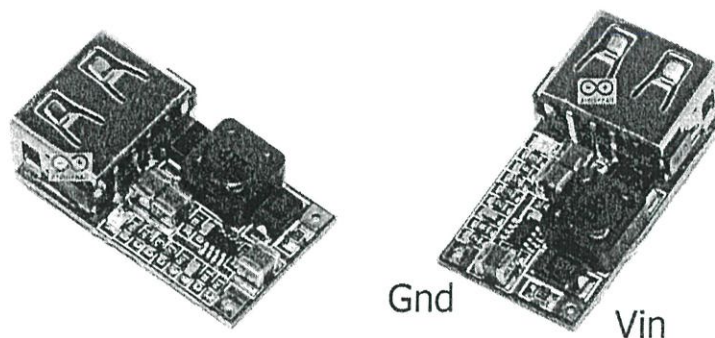
Module Features:

- the use of imported groove coupler sensor
- the slot width 5mm.
- the output state indicator lamp output high, output low lights.
- with cover, high output; unobstructed, output low.
- the comparator output signal clean waveform is good, driving ability, than 15mA.
- Operating Voltage 3.3V-5V
- the output format: Digital switching output (0 and 1)
- a fixed bolt holes for easy installation
- small plates PCB size: 3.2cm x 1.4cm
- 10, using a wide voltage LM393 comparator

Module for use:

- module slot unobstructed, the receiver tube is turned on, the module DO output low, cover when, DO output high;
- DO module can be connected to the relay, consisting limit switches and other functions, can also be connected with the active buzzer module, composed of alarm.

Regulator Step Down



Module Information

- Adopt new design , circuit design does not require electrolytic capacitors and tantalum capacitors (electrolytic capacitor life is short, tantalum risk of explosion). Now with the use of high-end circuit of large-size solid ceramic capacitors (MCC) 3 Ke .
- Very low ripple, 12V turn 5V 3A only about 10 millivolts, many colleagues have reached a ripple 500 mV or more, the device will result in an invisible injury, a serious cause device is not working or is damaged, the IPHONE but very delicate .
- The new program uses synchronous rectification circuit conversion efficiency is very high, in turn 5V 0.7A 6.5V when efficiency up to% 97.5 , 12V turn 5V 1A, also reached 94%, unlike some peers efficiency is only 60-70%. (Efficiency test removed the input reverse polarity protection diode and fuse drop).
- Using flat copper wire inductance , maximize conversion efficiency. Reduce heat.
- First the whole network with output overvoltage protection , input fuse has 1.5A, 300W output 5V have done the TVS tube voltage clamp protection, to protect USB devices.
- Now the buck peer USB modules do not have overvoltage protection, there is a very big security risk for USB devices, if you think about the step-down

module breakdown, 12V directly added to your IPHONE above your IPHONE can survive it?

- Using the latest USB identification circuit, perfectly compatible IPHONE, the vast majority of Android phones, IPAD .
- Gold-plated USB boundless short head , long service life, not ordinary USB interfaces can be compared.
- Output voltage indicator (red)
- Ultra-small size, 26.4 (long) * 15 (width) * 7.4 (height) mm, the whole network the minimum .
- Input with reverse polarity protection diode , the positive and negative reversed input will not be damaged, (We ignore the input diode test efficiency and pressure drop fuse)
- Ultra-low quiescent current, quiescent current of 0.85 mA . Connected to the car battery without adding switches can be connected to the cigarette lighter line or the ACC power cable line.

Module parameters

- Module Properties: non-isolated step-down module (BUCK) synchronous rectifier
- Input voltage: the DC 6-24 V
- Output voltage: 5.1-5.2V (You can choose to choose between 3.3-6V, if other voltages Please let the treasurer, the default shipping is 5.1V-5.2)
- Output current: Maximum 3 the A (Please note that full cooling), the actual test input 12V output 2A without special treatment within the heat.
- Conversion efficiency: Up 97.5% (6.5 0.7A switch 5V) (test value after shorting and reverse polarity protection fuse)
- Switching frequency: 5 00KHz
- Output ripple: 10mV around (12V turn 5V3A) 20M-bandwidth
- Output indication: output voltage LED is red
- Operating temperature: Industrial grade (-40 ℃ to + 85 ℃) (the higher the temperature, the smaller the output power)
- Output over-voltage protection: Yes, input fuse 1.5A, 5V output has 300W TVS tube clamp protection.

- Full load temperature rise: 30 deg.] C
- Static current: 0.85 Haoan
- Load regulation: $\pm 1 \%$
- Voltage regulation: $\pm 0.5\%$
- Dynamic response speed: 5% 200uS
- Output short circuit protection: Yes, do not short-circuit a long time
- Input reverse polarity protection: Yes, there are input reverse polarity protection diode
- Connection: Welding
- Input: Welding
- Output: short head boundless USB