

แบบจำลองเตียงช่วยพลิกตัวผู้ป่วย  
Smart Bed for Bedridden Elderly

ภาวรีย์ มานะรุ่งเรือง  
PHAWAREE MANAROONGRUENG  
สุจิตรา ทรงเจริญ  
SUJITTRA SONGCHAROEN

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมชีวการแพทย์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2560

แบบจำลองเตียงช่วยพลิกตัวผู้ป่วย  
Smart Bed for Bedridden Elderly

ภาวริย์ มานะรุ่งเรือง  
PHAWAREE MANAROONGRUENG  
สุจิตรา ทรงเจริญ  
SUJITTRA SONGCHAROEN

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมชีวการแพทย์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2560

แบบจำลองเตียงช่วยพลิกตัวผู้ป่วย  
Smart Bed for Bedridden Elderly

โดย

ภาวริย์ มานะรุ่งเรือง รหัสนักศึกษา 57010982  
สุจิตรา ทรงเจริญ รหัสนักศึกษา 57011383

อาจารย์ที่ปรึกษา  
ดร.สรินพร วิสิฐส์ทธาพงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมชีวการแพทย์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2560

ปริญญาโท  
สาขาวิชา  
คณะ  
เรื่อง  
ผู้จัดทำ

ปีการศึกษา 2560  
วิศวกรรมชีวการแพทย์  
วิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
แบบจำลองเตียงช่วยพลิกตัวผู้ป่วย  
Smart Bed for Bedridden Elderly  
ภาวริย์ มานะรุ่งเรือง รหัสนักศึกษา 57010982  
สุจิตรา ทรงเจริญ รหัสนักศึกษา 57011383

รายงานนี้ผ่านการตรวจโดยอาจารย์ที่ปรึกษาแล้ว



---

(ดร.สรินพร วิสิฐสัตธาพงศ์)

อาจารย์ที่ปรึกษา

หัวข้อโครงการ  
นักศึกษา

สาขาวิชา  
อาจารย์ที่ปรึกษางานวิจัย  
ปีการศึกษา

แบบจำลองเตียงช่วยพลิกตัวผู้ป่วย  
นางสาวภารีย์ มานะรุ่งเรือง 57010982  
นางสาวสุจิตรา ทรงเจริญ 57011383  
วิศวกรรมชีวการแพทย์  
ดร.สรินพร วิสิษฐัทธพงศ์  
2560

## บทคัดย่อ

แบบจำลองเตียงช่วยพลิกตัวผู้ป่วยมีวัตถุประสงค์ในการออกแบบขึ้นมาเพื่อลดภาระของผู้ดูแลผู้ป่วยติดเตียง ซึ่งมักเกิดขึ้นในผู้สูงอายุ และผู้ป่วยที่ไม่สามารถเคลื่อนไหวได้อย่างอิสระ ภาวะผู้ป่วยนอนติดเตียงมักก่อให้เกิดปัญหาแทรกซ้อนตามมา ทั้งระบบกระดูกและกล้ามเนื้อ ระบบประสาทสัมผัสและอารมณ์ รวมไปถึงระบบหัวใจและหลอดเลือดด้วย เช่น แผลกดทับ อารมณ์แปรปรวน หัวใจเต้นเร็วและสูบฉีดเลือดได้น้อย เป็นต้น โดยออกแบบพื้นเตียงของแบบจำลองเตียงช่วยพลิกตัวผู้ป่วยเป็น 4 ส่วน ใช้ stepper motor 2 ตัวควบคุมการเคลื่อนที่ขึ้นลงของพื้นเตียงทางด้านซ้ายและพื้นเตียงด้านขวา ทำงานร่วมกับ Motor driver board สั่งการผ่าน microcontroller (Arduino) และ push button ทำให้เมื่อกดปุ่ม Arduino จะสั่งการไปยัง Stepper motor และ Motor driver board ทำให้เตียงสามารถเคลื่อนที่ขึ้นและลงได้ ยิ่งไปกว่านั้นแบบจำลองนี้สามารถควบคุมผ่านแอปพลิเคชัน ใน Android และส่งสัญญาณเสียงเตือนเมื่อถึงเวลาที่ต้องพลิกตัวผู้ป่วยอีกครั้ง

Project Title	Smart Bed for Bedridden Elderly
Student	Phawaree Manaroongrueng 57010982 Sujittra Songcharoen 57011383
Program	Biomedical Engineering
Project Advisor	Dr.Sarinporn Visitsattapongse
Year	2017

## Abstract

Smart bed for bedridden elderly is a model created to reduce the burden on caregivers. It also helps prevent the patient with bedridden from complications such as bedsore, which often occur in the elderly.

In this paper the model is designed by divide bed into four parts. The model uses two stepper motors to control the movement up and down left side and right side of bed. Stepper motor works with motor driver board to command through microcontroller (Arduino) and button. When the caregiver pushes button, Arduino will command to stepper motor and motor driver board make the model can move. Moreover the model will work with android by connect android and Arduino together. It makes the caregivers can command through application and alarm on application when the time to flips the patient again.

## กิตติกรรมประกาศ

การทำปริญญานิพนธ์นี้สามารถสำเร็จลุล่วงไปได้ด้วยดีเนื่องจากได้รับความอนุเคราะห์และคำปรึกษาจากบุคคลหลายท่าน ตลอดทั้งให้ความรู้ในการทำโครงการแก่คณะผู้จัดทำมาโดยตลอด คณะผู้จัดทำขอขอบพระคุณหลักสูตรวิศวกรรมชีวการแพทย์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่สนับสนุนเครื่องมือ สถานที่และงบประมาณในการศึกษา ออกแบบ สร้างและทดสอบแบบจำลองเตียงช่วยพลิกตัวผู้ป่วยนี้

ขอขอบพระคุณอาจารย์ที่ปรึกษา ดร.สรินพร วิสิทส์ธาทพงศ์ ที่ให้ความช่วยเหลือ ให้คำชี้แนะ ตรวจสอบความถูกต้องและแก้ไขตลอดจนให้ความรู้และประสบการณ์ที่ดียิ่งขึ้น

ขอขอบคุณนายภาคภูมิ เจริญอารีย์ และนายณัฐนันท์ วันลักษณ์ ที่คอยให้คำปรึกษาและชี้แนะแนวทางการออกแบบและสร้างแบบจำลองเตียงช่วยพลิกตัวผู้ป่วยนี้ขึ้นมา

สุดท้ายนี้ขอขอบพระคุณ บิดา มารดาที่เลี้ยงดู อบรมสั่งสอนและให้การศึกษาย่างดี ทำให้คณะผู้จัดทำวันนี้ได้ และยังให้คำปรึกษาในทุกๆ ด้าน รวมถึงเป็นแรงผลักดันให้คณะผู้จัดทำในการทำสิ่งต่างๆ ให้ประสบความสำเร็จ

ผู้จัดทำ

# สารบัญ

	หน้า
บทคัดย่อ.....	I
Abstract .....	II
กิตติกรรมประกาศ.....	III
สารบัญ .....	IV
สารบัญรูปภาพ .....	V
สารบัญตาราง .....	VII
<b>บทที่ 1 บทนำ.....</b>	<b>1</b>
1.1 ที่มาและความสำคัญของงานวิจัย.....	1
1.2 วัตถุประสงค์ของการวิจัย .....	1
1.3 สมมติฐานในการทำวิจัย.....	1
1.4 ขอบเขตของงานวิจัย .....	2
1.5 แผนการดำเนินงาน .....	2
<b>บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....</b>	<b>3</b>
2.1 ภาวะผู้ป่วยติดเตียง .....	3
2.2 ภาวะเหนื่อยล้าของผู้ดูแล (Caregiver burden) .....	11
2.3 Arduino Uno.....	12
2.4 Stepper motor.....	14
2.5 Liquid Crystal Display (LCD) .....	23
2.6 Push button switch .....	26
2.7 แหล่งจ่ายไฟ .....	29
2.8 ไอซีเร็กกูเลเตอร์ (IC Regulator) .....	33
2.9 MIT App Inventor.....	36
<b>บทที่ 3 วิธีดำเนินงานวิจัย.....</b>	<b>43</b>
3.1 วิธีที่ใช้ศึกษาค้นคว้าและการวิจัยการทดลอง.....	43
3.2 ลักษณะข้อมูล การเลือกข้อมูล และการทดลอง.....	43
3.3 เครื่องมือและวิธีการทดลอง .....	43
3.4 ขั้นตอนออกแบบและสร้างเครื่องมือ .....	43
<b>บทที่ 4 ผลการวิจัย.....</b>	<b>60</b>
4.1 การออกแบบและสร้าง Hardware .....	60
4.2 การออกแบบแอปพลิเคชัน.....	61
4.3 การทดสอบ Hardware และ Software.....	61
<b>บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ .....</b>	<b>63</b>
5.1 สรุปผลการวิจัย .....	63
5.2 ปัญหาและแนวทางแก้ไขปัญหา .....	63
บรรณานุกรม .....	64
ภาคผนวก .....	66

# สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 บริเวณที่อาจเกิดผลกดทับ .....	6
รูปที่ 2.2 ระยะเวลาการเกิดผลกดทับ .....	8
รูปที่ 2.3 บอร์ด Arduino Uno .....	12
รูปที่ 2.4 ขาต่างๆของบอร์ด Arduino Uno .....	13
รูปที่ 2.5 Stepper Motor .....	14
รูปที่ 2.6 โครงสร้าง Stepping Motor .....	15
รูปที่ 2.7 แผนผังการทำงานของ Stepper Motor .....	15
รูปที่ 2.8 โครงสร้างอย่างง่ายของ Stepper Motor ชนิดไฮบริด .....	16
รูปที่ 2.9 ลักษณะการพันขดลวด Stepper Motor ไบโพลาร์.....	16
รูปที่ 2.10 ลักษณะการพันขดลวด Stepper Motor ยูนิโพลาร์.....	17
รูปที่ 2.11 การใช้มอเตอร์วัดค่าความต้านทาน .....	20
รูปที่ 2.12 แสดงการต่อวงจรเพื่อทดสอบโดยการ สวิตช์เพื่อหาลำดับ .....	20
รูปที่ 2.13 การเชื่อมต่อระหว่าง Stepper Motor, Stepper Motor Driver, Arduino Uno .....	22
รูปที่ 2.14 ด้านหน้าจอ LCD 16x2 (Parallel).....	24
รูปที่ 2.15 ด้านหน้าและด้านหลังจอ LCD 16x2 (I2C) .....	25
รูปที่ 2.16 การเชื่อมต่อระหว่าง Arduino Uno และ LCD .....	26
รูปที่ 2.17 Push button switch .....	26
รูปที่ 2.18 โครงสร้างของ สวิตช์.....	27
รูปที่ 2.19 วงจร pull-up.....	28
รูปที่ 2.20 วงจร pull-down .....	29
รูปที่ 2.21 ตัวอย่าง Regulator คมค่า .....	33
รูปที่ 2.22 แผนผังการทำงานของเร็กูเลเตอร์แบบขนาน .....	34
รูปที่ 2.23 วงจรแปลงไฟ .....	35
รูปที่ 2.24 การเขียนโปรแกรมบนมือถือ Android ด้วย .....	37
รูปที่ 2.25 หน้าจอการจัดการโปรเจค.....	38
รูปที่ 2.26 หน้าจอการออกแบบ (Viewer).....	39
รูปที่ 2.27 ตัวอย่างของบล็อกคำสั่งประเภทที่ใช้เรียกค่าคุณสมบัติจาก.....	41
รูปที่ 2.28 ตัวอย่างของบล็อกคำสั่งประเภทที่ใช้กำหนดค่าคุณสมบัติให้กับ .....	41
รูปที่ 2.29 ตัวอย่างของบล็อกคำสั่งประเภทเหตุการณ์ (event handler).....	41
รูปที่ 2.30 ตัวอย่างของบล็อกคำสั่งประเภทกระบวนการทำงาน.....	42
รูปที่ 3.1 พื้นเตียงช่วยรับน้ำหนักของผู้ป่วย.....	44
รูปที่ 3.2 rod shaft .....	44
รูปที่ 3.3 ชิ้นส่วนที่ยึด rod shaft เข้าด้วยกัน .....	45
รูปที่ 3.4 ชิ้นส่วนที่ช่วยยึดสตีปเปอร์มอเตอร์ (stepper motor) ไม่ให้เคลื่อนที่.....	45
รูปที่ 3.5 ประกอบเข้ากับสตีปเปอร์มอเตอร์ (stepper motor).....	45
รูปที่ 3.6 เหล็ก구 .....	46

รูปที่ 3.7 ประกอบรวมกัน .....	46
รูปที่ 3.8 สัตัดเกลียว.....	47
รูปที่ 3.9 coupling ต่อกับสัตัดเกลียว.....	47
รูปที่ 3.10 ชิ้นส่วนที่ใช้เชื่อมกับไม้.....	48
รูปที่ 3.11 ไม้เป็นตัวกลางเพื่อช่วยให้ระบบ scissor lift เคลื่อนที่ได้ .....	48
รูปที่ 3.12 อาร์ดูโน้ อุโน้ (Arduino Uno).....	49
รูปที่ 3.13 สเต็ปเปอร์มอเตอร์ (Stepper Motor) .....	49
รูปที่ 3.14 TB6560 Stepper Motor Driver .....	50
รูปที่ 3.15 LCD I2C.....	50
รูปที่ 3.16 Push button switch.....	51
รูปที่ 3.17 ลิ้มิตสวิตช์ (Limit switch).....	51
รูปที่ 3.18 บลัซเซอร์ (Buzzer) .....	52
รูปที่ 3.19 Bluetooth module HC05 .....	52
รูปที่ 3.20 การต่อstepper motor , motor drive board และ Arduino .....	53
รูปที่ 3.21 การต่อปุ่มกดเข้ากับ Arduino .....	53
รูปที่ 3.22 IC 7805 .....	54
รูปที่ 3.23 Heat sink.....	55
รูปที่ 3.24 Adapter .....	55
รูปที่ 3.25 แผ่นวงจรพิมพ์อเนกประสงค์ (Universal PCB Board).....	56
รูปที่ 3.26 แผนผังแสดงการทำงานของอุปกรณ์ ในการควบคุมการเคลื่อนที่ของเตียงช่วยพลิกตัว ผู้ป่วย.....	56
รูปที่ 3.27 การออกแบบหน้าจอแสดงผล .....	57
รูปที่ 3.28 การออกแบบการทำงานในโปรแกรม Inventor app.....	57
รูปที่ 3.29 หน้าจอแสดงในแอปพลิเคชัน.....	58
รูปที่ 3.30 ตัวอย่างการสั่งการผ่านสวิตช์.....	59
รูปที่ 4.1 พื้นเตียง.....	60
รูปที่ 4.2 ระบบ scissors lift .....	60
รูปที่ 4.3 หน้าต่างแอปพลิเคชัน จาก Android.....	61
รูปที่ 4.4 เตียงเมื่อถูกพลิกไปทางขวา.....	61
รูปที่ 4.5 เตียงเมื่อถูกพลิกไปทางซ้าย.....	62
รูปที่ 4.6 การแสดงผลบนหน้าจอ LCD .....	62

# สารบัญตาราง

	หน้า
ตารางที่ 1.1 แผนการดำเนินงาน.....	2
ตารางที่ 2.1 การควบคุมการหมุนแบบเฟสเดียว.....	18
ตารางที่ 2.2 การควบคุมการหมุนแบบสองเฟส.....	18
ตารางที่ 2.3 การควบคุมการหมุนแบบครึ่งเฟส.....	19
ตารางที่ 2.4 Running Current.....	21
ตารางที่ 2.5 Stop Current.....	21
ตารางที่ 2.6 Excitation Mode.....	21
ตารางที่ 2.7 Decay Setting .....	22
ตารางที่ 2.8 ขาของจอ LCD 16x2 แบบ Parallel .....	24
ตารางที่ 2.9 ขาของจอ LCD 16x2 แบบ I2C.....	25
ตารางที่ 2.10 การเชื่อมต่อขาระหว่าง Arduino Uno กับ LCD (I2C).....	26
ตารางที่ 3.1 แสดงการต่อ Stepper motor กับ Motor driver board และ Arduino.....	54

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญของงานวิจัย

ในช่วงเวลาที่ผ่านมา สัดส่วนประชากรในประเทศไทยได้มีการเปลี่ยนแปลงอย่างรวดเร็วโดยเฉพาะผู้สูงอายุ กล่าวคือ ประชากรผู้สูงอายุหรือประชากรที่มีอายุ 60 ปีขึ้นไป มีจำนวนและสัดส่วนเพิ่มขึ้น และมีแนวโน้มที่จะเพิ่มอย่างรวดเร็วในอนาคต ทำให้ประเทศไทยก้าวเข้าสู่สังคมผู้สูงอายุอย่างหลีกเลี่ยงไม่ได้ ซึ่งการที่ประชากรมีชีวิตยืนยาวขึ้น มิได้หมายถึงประชากรผู้สูงอายุจะมีสุขภาพดีขึ้น แต่กลับกลายเป็นว่าจะเป็นช่วงชีวิตที่มีการเจ็บป่วยเพิ่มขึ้นตามลำดับด้วย ผู้สูงอายุส่วนหนึ่งจำเป็นต้องมีผู้ดูแลจากความเสื่อมสภาพของร่างกายและจากโรคภัยไข้เจ็บที่ทำให้เกิดความพิการจากการสำรวจเป็นระยะ พบว่ามีผู้สูงอายุกลุ่มนี้หรือที่เรียกว่าผู้ป่วยติดเตียงประมาณร้อยละ 15 หรือ 1.5 แสนคน(วิจัย, 2560) ซึ่งก่อให้เกิดภาวะแทรกซ้อนตามมามากมาย เช่น ภาวะข้อติดแข็ง ภาวะกล้ามเนื้ออ่อนแรง และภาวะแผลกดทับ เป็นต้น นำไปสู่การเสียชีวิตในที่สุด อัตราการป่วยด้วยโรคเรื้อรังที่เพิ่มขึ้นนำมาซึ่งจำนวนผู้ป่วยสูงอายุที่ทุพพลภาพและต้องการการดูแลมากขึ้น หน้าที่ของการดูแลผู้ป่วยเหล่านี้ ส่วนใหญ่มักตกหนักที่สมาชิกในครอบครัวของผู้ป่วยเอง

ในบางครั้งผู้ที่ทำหน้าที่ดูแลผู้ป่วยมักถูกมองข้าม ซึ่งจริงๆแล้วการดูแลนั้นบางครั้งยากและเหนื่อยยิ่งกว่าการรักษา เนื่องจากเป็นงานที่ต้องอยู่กับผู้ป่วยตลอดเวลา การที่ผู้ป่วยจะได้รับยาอาหารหรือการดูแลต่างๆตามแพทย์สั่งขึ้นอยู่กับผู้ดูแลเหล่านี้ด้วย จึงไม่ควรละเลยความสำคัญของบุคคลกลุ่มนี้ ภาระการดูแลผู้สูงอายุหรือผู้ป่วยโรคเรื้อรังที่ต้องพึ่งพาผู้ดูแลตลอดเวลา นั้น เป็นงานที่เหน็ดเหนื่อย ใช้เวลามาก และก่อให้เกิดความตึงเครียดสูง อีกทั้งตัวผู้ดูแลเองอาจมีปัญหาของตนเองในหลายๆด้าน ไม่ว่าจะเป็นปัญหาสุขภาพ ชีวิตความเป็นอยู่ ฐานะการเงิน และชีวิตส่วนตัวของตนเองเมื่อเวลาผ่านไป ถ้าผู้ดูแลปราศจากความช่วยเหลือจากสมาชิกคนอื่นๆแล้ว ก็จะทำให้เกิดความเหนื่อยล้าได้ ซึ่งเป็นที่มาของความต้องการการช่วยเหลือเรื่องอุปกรณ์และการรักษา

ดังนั้นการศึกษา ออกแบบและพัฒนาเตียงสำหรับช่วยพลิกตัวผู้ป่วย จึงมีความสำคัญในการนำไปใช้ช่วยลดภาระการพลิกตัวผู้ป่วยของผู้ดูแล เนื่องจากผู้ป่วยจำเป็นต้องพลิกตัวทุกๆ 2 ชั่วโมง เพื่อป้องกันการเกิดปัญหาแทรกซ้อน

### 1.2 วัตถุประสงค์ของการวิจัย

- 1.2.1 ศึกษาปัญหาที่เกิดขึ้นกับผู้ป่วยติดเตียงและผู้ดูแลรวมถึงการแก้ไขปัญหา
- 1.2.2 ออกแบบและพัฒนาแบบจำลองเตียงช่วยพลิกตัวผู้ป่วยเพื่อลดภาระของผู้ดูแล

### 1.3 สมมติฐานในการทำวิจัย

- 1.3.1 แบบจำลองเตียงช่วยพลิกตัวผู้ป่วยสามารถช่วยลดภาระในการดูแลผู้ป่วยได้
- 1.3.2 ระบบแจ้งเตือนเมื่อถึงเวลาพลิกตัว สามารถช่วยลดระยะเวลาการเฝ้าผู้ป่วยได้
- 1.3.3 การแบ่งส่วนเตียงเป็น 4 ส่วนสามารถป้องกันผู้ป่วยตกเตียงเมื่อทำการพลิกตัว



## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

#### 2.1 ภาวะผู้ป่วยติดเตียง

##### 2.1.1 ภาวะแทรกซ้อนจากการนอนติดเตียง

ภาวะแทรกซ้อนจากการนอนติดเตียงเป็นเวลานาน มีดังนี้

##### 2.1.1.1 ระบบกระดูกและกล้ามเนื้อ

- ความแข็งแรงของกล้ามเนื้อลดลง ลดลงวันละ 1-3% ต่อวัน
- มีปัญหาข้อติด ซึ่งเกิดได้ทุกข้อ เนื่องจากอยู่ในท่าเดิมนานๆ และไม่ได้ขยับ

ตัว เช่น ข้อไหล่ติด ไม่สามารถยกขึ้นแนบหูได้, ข้อศอกติดในท่างอ ทำให้ไม่สามารถเหยียดข้อศอกได้ ทุกข้อ, ข้อมือติดในท่ากระดูกงอ, นิ้วมือกำแน่น ไม่สามารถกางอปกได้, ข้อสะโพกและข้อเข่าติดในท่างอ, ข้อเท้าติดในท่าจิกงอ ซึ่งการติดของข้อต่าง ๆ ก่อให้เกิดปัญหาตามมาภายหลังได้ เช่น การติดของข้อไหล่ทำให้ไม่สามารถเอื้อมมือหยิบของได้ การติดของข้อศอกทำให้มีปัญหาการใส่เสื้อผ้า การติดของข้อนิ้วมือทำให้หยิบจับไม่สะดวก การติดของข้อสะโพก ข้อเข่าและข้อเท้า ทำให้เป็นอุปสรรคต่อการเดินในอนาคต เนื่องจากขาสั้นยาวไม่เท่ากัน และการเดินมีการสิ้นเปลืองพลังงานมากกว่าคนปกติ

- มือและเท้าบวม เนื่องจากมือและเท้าไม่ได้ขยับ และอยู่ต่ำกว่าระดับหัวใจ
- เกิดแผลกดทับตามจุดต่าง ๆ ของร่างกาย โดยเฉพาะจุดที่เป็นส่วนนูนของ

กระดูก เช่น บริเวณของปุ่มบริเวณศีรษะด้านหลัง สะบัก ก้นกบ กระดูกสะโพก กระดูกตา ตุ่มด้านนอก เป็นต้น สาเหตุที่มักเกิดตรงบริเวณปุ่มกระดูกเพราะขาดเลือดมาเลี้ยงตรงผิวหนังและเนื้อเยื่อตรงบริเวณนั้น จากการนอนทับเป็นเวลานาน ๆ

- เกิดภาวะกระดูกบาง โดยเฉพาะข้อที่รับน้ำหนัก เช่น ข้อสะโพกและข้อเข่า

##### 2.1.1.2 ระบบหัวใจและหลอดเลือด

พบว่าในผู้ป่วยที่นอนนาน อัตราการเต้นของหัวใจขณะพักจะเต้นเร็วกว่าคนปกติ และหัวใจสูบฉีดเลือดต่อครั้งได้น้อยกว่าคนปกติ ร่วมกับมีภาวะหน้ามืด เป็นลมได้ง่าย เมื่อลุกนั่งในทันที ระบบหายใจพบว่าท่านอน การหายใจต่อครั้งจะได้อากาศน้อยกว่าท่านั่ง เพราะมีลำไส้มาดันกระบังลม ทำให้กระบังลมทำงานได้น้อยกว่าปกติ ผู้ป่วยไอนี้ไม่มีประสิทธิภาพเต็มที่ และการนอนอยู่นิ่ง ๆ ทำให้มีการตกค้างของเสมหะในแขนงของหลอดลม ซึ่งภาวะดังกล่าวทำให้เกิดภาวะปอดบวมติดเชื้อ และปอดแฟบได้ง่าย

##### 2.1.1.3 ระบบประสาทสัมผัสและอารมณ์

มีการศึกษาพบว่า การมองเห็นจะลดลง มีภาวะหูตึงมากขึ้น และมีอารมณ์แปรปรวนง่าย เช่น วิตกกังวลหรือซึมเศร้า เป็นต้น

### 2.1.2 วิธีป้องกันภาวะแทรกซ้อน

จากข้างต้นจะเห็นว่า การป้องกันภาวะแทรกซ้อนจะได้ผลดีกว่ามาแก้ปัญหาเมื่อเกิดภาวะแทรกซ้อนขึ้นแล้ว ซึ่งการป้องกันภาวะแทรกซ้อนดังกล่าวจำเป็นต้องใช้ความร่วมมือในการทำงานและประสานงานของฝ่ายต่าง ๆ ได้แก่ แพทย์เจ้าของไข้ แพทย์เวชศาสตร์ฟื้นฟู พยาบาล ผู้ช่วยพยาบาล นักกายภาพบำบัด และนักกิจกรรมบำบัด เป็นต้น โดยในที่นี้จะกล่าวถึงวิธีการดูแลโดยรวม ดังนี้

- การจัดทำของส่วนต่าง ๆ ของร่างกาย โดยให้ข้อไหล่ ข้อศอก ข้อมือ นิ้วมือ ลำตัว ข้อสะโพก ข้อเข่าและข้อเท้า ให้อยู่ในท่าที่เหมาะสม

- การพลิกตะแคงตัว เพื่อป้องกันแผลกดทับและข้อติด ถ้าผู้ป่วยรู้ตัว และสามารถพลิกตะแคงตัวได้ ให้พยายามตะแคงตัวบ่อย ๆ ทุก 30 นาที ยกเว้นช่วงที่นอนหลับ ในกรณีที่ไม่รู้สีกตัว ควรพลิกตะแคงตัวทุก 2 ชั่วโมง โดยเปลี่ยนท่าสลับกันไป ตั้งแต่ตะแคงซ้าย นอนหงาย ตะแคงขวา นอนคว่ำ (ถ้าทำได้) โดยญาติ พยาบาล และนักกายภาพบำบัด

- การจัดให้ผู้ป่วยอยู่ในท่านั่งบ่อย ๆ โดยการไขหัวเตียง โดยเริ่มให้หัวเตียงสูงครั้งละ 15-30 องศา ถ้าผู้ป่วยสามารถบอกและสื่อสารได้ ถ้าไม่มีอาการมีน็ศีรษะให้ปรับขึ้นได้อีก จนสามารถนั่งได้ตรง 90 องศา แต่ถ้ามีอาการมีน็ศีรษะให้ปรับลงจนไม่มีน็ศีรษะ แล้วรอ 15-20 นาที จึงค่อย ๆ ปรับขึ้นใหม่

- การบริหารข้อต่าง ๆ ถ้าผู้ป่วยสามารถทำได้เอง ให้หมั่นยกข้อไหล่ให้แนบติดหู งอและเหยียดข้อศอกจนสุดมุม กระดกข้อมือขึ้นลง กำมือและแบมือจนสุด งอและเหยียดข้อสะโพกจนสุด กางและหุบข้อสะโพกจนสุด งอและเหยียดข้อเข่าจนสุด กระดกข้อเท้าขึ้นลงจนสุด งอและเหยียดนิ้วเท้าจนสุด แต่ถ้าผู้ป่วยไม่สามารถทำได้ให้พยายามให้ทำมากที่สุด ที่เหลือให้ญาติ พยาบาล หรือนักกายภาพบำบัดช่วยทำต่อ โดยทำข้อละ 3-5 ครั้ง วันละ 2 ครั้ง ในช่วงเช้าและเย็น

- ฝึกหายใจแบบใช้กระบังลม และฝึกหายใจเข้าเต็มที่

- ในรายที่สามารถนั่งได้ ให้พยายามนั่งข้างเตียงบ่อย ๆ โดยมีญาติคอยดูแล

- ในรายที่อาการดีขึ้นและไม่มีสายทางการแพทย์ และปลอดภัยสามารถลงมาฝึกทำกายภาพบำบัดที่ชั้นล่างตึกกระดูกและข้อได้ จะได้ประโยชน์มาก เพราะมีความพร้อมของเครื่องมือมากกว่าบนหอผู้ป่วย

### 2.1.3 การฟื้นฟูผู้ป่วยติดเตียง

การฟื้นฟูผู้ป่วยติดบ้านและติดเตียง ปัจจุบันในสังคมไทยนั้นมีผู้สูงอายุมากขึ้นและเพิ่มขึ้นเรื่อยๆ และอีกหนึ่งโรคที่จะมีมาพร้อมกับผู้สูงอายุ คือ โรคผู้ป่วยติดเตียง วิธีการดูแลรักษาพร้อมกับคำแนะนำสำหรับโรคผู้ป่วยติดเตียง

ลักษณะของผู้ป่วย มี 2 กลุ่ม คือ ผู้ป่วยติดบ้าน และผู้ป่วยติดเตียง

- ผู้ป่วยติดบ้าน มีลักษณะอาการคือ สามารถเคลื่อนไหว เดินภายในบ้านได้ ออกนอกบ้านไม่ได้ต้องอยู่ในพื้นที่จำกัด

- ผู้ป่วยติดเตียง มีลักษณะอาการ คือ ผู้ป่วยที่มีโรค ไม่สามารถเคลื่อนไหวตัวเองได้ หรืออาจจะเป็นกล้ามเนื้ออ่อนแรง

การฟื้นฟู การดูแลผู้ป่วยติดบ้าน มีขั้นตอนดังนี้

- ในกรณีที่มีผู้สูงวัยเป็นผู้ป่วยติดบ้าน คนในครอบครัวต้องดูแลอาการ ให้คงสภาพให้มากที่สุด ช่วยเหลือตัวเองพื้นฐานได้ เช่น ผู้ป่วยสามารถเดินเองได้, เข้าห้องน้ำเองได้ แปรุงฟันเองได้ ต้องให้เขา คงสภาพมากที่สุด ระวังเรื่องการหกล้ม หรืออุบัติเหตุ เล็กๆ น้อยๆ เพื่อไม่ให้อาการทรุดลงไปกว่าเดิม

การฟื้นฟูการดูแล ผู้ป่วยติดเตียง มีขั้นตอนในการดูแลหลักๆ 3 ข้อ และเน้นในส่วนของผู้ดูแล เพื่อปฏิบัติกับผู้ป่วยได้ถูกต้อง

- ผู้ป่วยต้องมีคุณภาพชีวิตให้ดีที่สุด ไม่มีแผลกดทับ ไม่มีข้อติด อาหาร การขับถ่าย การปฏิบัติเหล่านี้ต้องคำนึงถึงความสะอาดและความเหมาะสม

- ลดภาระที่จะนำผู้ป่วยไป โรงพยาบาล เพราะครอบครัวและผู้ดูแลสามารถดูแลได้ดีเหมือนกัน จึงไม่มีความจำเป็นที่จะต้องไปโรงพยาบาล เพราะการเคลื่อนย้ายผู้ป่วยในแต่ละครั้งมีค่าใช้จ่ายและมีอุปสรรคในการเคลื่อนย้ายผู้ป่วย

- ต้องเป็นภาระของผู้ดูแลให้น้อยที่สุด ต้องให้ความร่วมมือกับผู้ดูแลในการขับหรือยกส่วนต่างๆ ของตัวผู้ป่วย

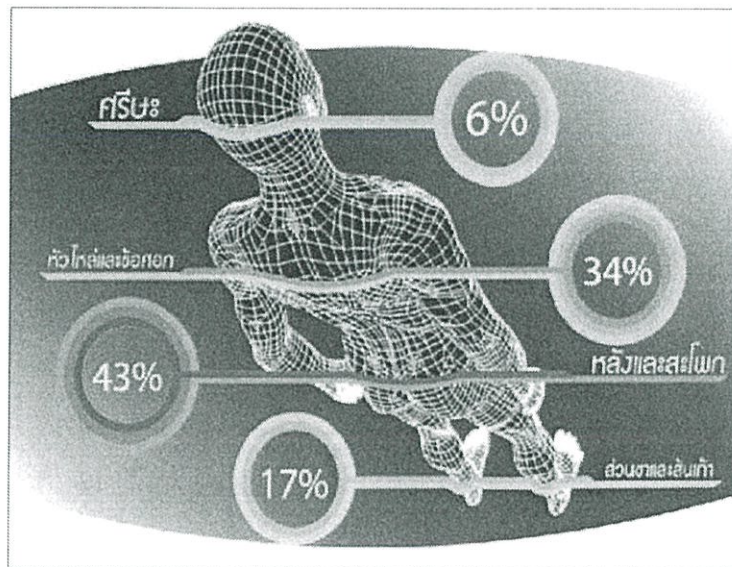
#### 2.1.4 แผลกดทับ อาการที่เกิดจากผู้ป่วยติดเตียง

แผลกดทับ คือ บริเวณเนื้อเยื่อหรือเซลล์ที่มีการตายเนื่องจากขาดเลือดไปเลี้ยง จากการถูกกดทับเป็นเวลานาน สำหรับส่วนที่พบแผลกดทับบ่อย คือ บริเวณเนื้อเยื่อหรือปุ่มกระดูก ที่พบส่วนมากคือบริเวณปุ่มกระดูกก้นกบ ตามตุ่ม และส้นเท้า เพราะโดยปกติคนทั่วไปจะชอบนอนหงาย แต่ที่มากที่สุด คือ บริเวณก้นกบสะโพก ต่อมาก็คือบริเวณสะบักทั้ง 2 ข้าง ส้นเท้าทั้ง 2 ข้าง และบริเวณท้ายทอย สำหรับแผลที่เกิดกับผู้ที่นั่งนอนหงาย แต่สำหรับผู้ที่นั่งนอนตะแคงแผลกดทับที่เกิด คือ บริเวณด้านข้างของศีรษะ ปุ่มกระดูกบริเวณหัวไหล่ บริเวณข้อศอก ต้นขา ข้างเข่าด้านนอกและ ตาตุ่ม

อีกสิ่งหนึ่ง คือ ผู้ที่เป็นเบาหวาน ไทรอยด์ หรือมีการบวมซึ่งจะขัดขวางการไหลเวียนเลือด อาหารที่ไปเลี้ยงเซลล์ต่างๆ เมื่อสารอาหารและออกซิเจนน้อยลงก็จะเกิดส่วนที่เป็นแผลกดทับได้ โดยเฉพาะผู้ที่มีภาวะโลหิตจางก็ควรระวัง สำหรับผู้สูงอายุที่มีปัญหาเกี่ยวกับด้านโภชนาการไม่ดี ในผู้สูงอายุย่อมส่งผลด้วยเช่นเดียวกัน เมื่อผู้ป่วยสูงอายุหรือผู้ที่มีอาการป่วยเรื้อรังทานอาหารได้น้อยลง ได้รับสารอาหารที่น้อยลงตามมา เพราะฉะนั้นความต้านทานของเนื้อเยื่อก็มีน้อยลง ยิ่งถ้าเกิดแผลกดทับแล้วก็จะหายยาก

### 2.1.4.1 สาเหตุและปัจจัยเสี่ยงที่ทำให้เกิดแผลกดทับ

รูปที่ 2.1 แสดงอัตราการเกิดแผลกดทับบริเวณต่างๆในร่างกาย ซึ่งประกอบได้ด้วย ศรีษะ, หัวไหล่และข้อศอก, หลังและสะโพก และส่วนขาและข้อเท้า



รูปที่ 2.1 บริเวณที่อาจเกิดแผลกดทับ

แผลกดทับเกิดจากอวัยวะส่วนใดส่วนหนึ่งในร่างกายได้รับแรงกดเป็นเวลานาน อันส่งผลให้เลือดไหลเวียนไปเลี้ยงบริเวณดังกล่าวไม่เพียงพอ หากเลือดไม่ไปเลี้ยงอวัยวะที่ถูกกดทับ เนื้อเยื่อของอวัยวะดังกล่าวจะถูกทำลายและเริ่มตาย เนื่องจากเลือดจะลำเลียงออกซิเจนและสารอาหารต่าง ๆ ที่จำเป็นและช่วยเสริมสร้างเนื้อเยื่อไปเลี้ยงส่วนต่าง ๆ ของร่างกาย อีกทั้งยังส่งผลให้ผิวหนังไม่ได้รับเซลล์เม็ดเลือดขาวสำหรับต้านทานเชื้อโรค ทำให้เกิดการติดเชื้อที่แผลกดทับได้ ปัจจัยที่ทำให้เกิดแผลกดทับมี ดังนี้

- แรงกด หากส่วนใดส่วนหนึ่งของร่างกายถูกกดทับเป็นเวลานาน จะส่งผลให้เลือดไหลเวียนมาเลี้ยงไม่เพียงพอ เมื่อไม่ได้รับออกซิเจนและสารอาหารที่ลำเลียงมากับเลือดไปหล่อเลี้ยง เนื้อเยื่อส่วนต่าง ๆ จึงถูกทำลายและอาจตายได้ตามที่กล่าวไปข้างต้น ทั้งนี้ ผู้ที่ขยับหรือเคลื่อนไหวร่างกายไม่ค่อยได้ อาจเกิดการกดทับที่กระดูกสันหลัง กระดูกก้นกบ หัวไหล่ สะโพก สันเท้า และข้อศอก

- แรงเสียดทานหรือความฝืด คือ แรงต้านต่อการเคลื่อนที่ของพื้นผิว 2 ด้าน ได้แก่ แรงต้านระหว่างผิวหนังกับวัสดุที่ผิวหนังสัมผัส เช่น ผ้าปูเตียง เบาะรองนั่ง ซึ่งแรงเสียดทานนี้ ถ้ามีมากเกินไปจะทำให้เกิดบาดแผลถลอก (Abrasion) หรือผิวหนังพอง (Blister) หรือฉีกขาดได้ถ้าแผลฉีกหรือผิวหนังมีความชื้นมากเกินไปจะทำให้ผิวหนังบริเวณนั้นติดกับผ้าปูเตียงหรือวัสดุที่สัมผัส ทำให้มีแรงเสียดทานเพิ่มขึ้น

- แรงเฉือน ชั้นผิวหนังถูกรังกันไว้ มักเกิดขึ้นเมื่อผู้ป่วยนอนไกลตัวลงมาในขณะที่เตียงปรับระดับสูง ส่งผลให้ผิวหนังบริเวณก้นกบเกิดการดึงรั้ง

นอกจากนี้ ยังปรากฏปัจจัยเสี่ยงที่ก่อให้เกิดแผลกดทับ ได้แก่ ปัญหาการเคลื่อนไหว โภชนาการไม่ดี ปัญหาสุขภาพบางอย่าง อายุมากขึ้น และปัญหาสุขภาพจิต ดังนี้

- ปัญหาการเคลื่อนไหว ผู้ที่มีปัญหาเกี่ยวกับการเคลื่อนไหวร่างกายเสี่ยงเกิดแผลกดทับได้ โดยปัญหาดังกล่าวอาจเกี่ยวเนื่องกับ
  - การได้รับบาดเจ็บที่ไขสันหลัง อันส่งผลให้ขยับแขนหรือขาไม่ได้
  - สมองถูกทำลายจากโรคหลอดเลือดในสมองหรือได้รับบาดเจ็บที่ศีรษะอย่างรุนแรง ก่อให้เกิดอัมพาต
  - ป่วยเป็นโรคบางอย่างอันทำลายเส้นประสาทที่ใช้ในการเคลื่อนไหวร่างกายอย่างต่อเนื่อง เช่น อัลไซเมอร์ โรคปลอกประสาทเสื่อมแข็ง หรือพาร์กินสัน
  - เกิดอาการเจ็บปวดอย่างรุนแรงจนเคลื่อนไหวร่างกายบางส่วนหรือทั้งหมดได้ไม่ถนัด
  - กระดูกแตกหรือกระดูกหัก
  - พักฟื้นจากการเข้ารับผ่าตัด
  - ประสบภาวะโคม่า
  - ประสบปัญหาสุขภาพที่ทำให้เคลื่อนไหวข้อต่อหรือกระดูกลำบาก เช่น โรคข้ออักเสบรูมาตอยด์
- โภชนาการไม่ดี สาเหตุที่ทำให้ได้รับสารอาหารไม่ครบถ้วนและเพียงพอต่อความต้องการร่างกายอาจเกิดจาก
  - โรคอันอริกเซีย ปัญหาสุขภาพจิตที่ผู้ป่วยยึดติดกับการลดน้ำหนักตัวให้ผอมลงเรื่อย ๆ โดยใช้วิธีลดน้ำหนักที่เป็นอันตรายต่อสุขภาพ
  - ภาวะขาดน้ำ ภาวะที่ร่างกายได้รับน้ำไม่เพียงพอ
  - การกลืนลำบาก ปัญหาเกี่ยวกับการกลืนอาหารที่ทำให้กลืนอาหารได้ยาก
- ปัญหาสุขภาพบางอย่าง ผู้ที่ป่วยเป็นโรคบางอย่างอาจเสี่ยงเกิดแผลกดทับได้ง่าย โดยปัญหาสุขภาพที่เอื้อให้เกิดแผลกดทับนั้นประกอบด้วย
  - เบาหวานชนิดที่ 1 และชนิดที่ 2 ผู้ป่วยเบาหวานที่มีระดับน้ำตาลในเลือดสูงขึ้นอาจส่งผลต่อระบบไหลเวียนโลหิตในร่างกาย
    - เส้นเลือดแดงส่วนปลายอุดตัน (Peripheral Arterial Disease) ผู้ป่วยโรคนี้ประสบภาวะเลือดไปเลี้ยงที่ขาไม่ได้ เนื่องจากเกิดการอุดตันของไขมันที่หลอดเลือดแดง
    - หัวใจวาย ผู้ที่ประสบภาวะหัวใจวายเกิดจากการที่ร่างกายสูบน้ำเลือดไปเลี้ยงอวัยวะส่วนต่าง ๆ ได้ไม่เพียงพอ
    - ไตวาย ผู้ป่วยไตวายจะสูญเสียสมรรถภาพการทำงานของไต ส่งผลให้เกิดสารพิษในเลือด ซึ่งก่อให้เกิดการทำลายเนื้อเยื่อ
    - โรคปอดอุดกั้น ผู้ป่วยโรคนี้จะมีระดับออกซิเจนในเลือดต่ำ ส่งผลให้ผิวหนังถูกทำลายได้ง่าย
  - อายุมากขึ้น ผู้ที่อายุมากขึ้นเสี่ยงเกิดแผลกดทับได้ง่าย เนื่องจากผิวหนังสูญเสียความยืดหยุ่น เลือดไหลเวียนไปเลี้ยงผิวหนังน้อยลง รวมทั้งไขมันใต้ผิวหนังลดลง ทั้งหมดนี้ล้วนเกิดจากการเสื่อมสภาพตามวัย
    - ปัญหาเกี่ยวกับการขับถ่ายและปัสสาวะ ผู้ป่วยที่ประสบภาวะกลั้นปัสสาวะหรืออุจจาระไม่ได้จะมีผิวหนังบางส่วนที่อับชื้น ซึ่งทำให้ติดเชื้อได้ง่าย ทั้งนี้ ผิวหนังที่อับชื้นยังทำให้เกิดแผลกดทับตามมาด้วย

• ปัญหาสุขภาพจิต ผู้ที่ประสบปัญหาสุขภาพจิตร้ายแรง เช่น โรคจิตเภทหรือโรคซึมเศร้าอย่างรุนแรงเสี่ยงเกิดแผลกดทับได้ง่าย เนื่องจากโภชนาการไม่ดีและป่วยเป็นโรคอื่น ๆ ร่วมด้วย เช่น เบาหวาน หรือภาวะกลั้นปัสสาวะไม่ได้ ทั้งนี้ ผู้ที่มีปัญหาสุขภาพจิตบางรายอาจรักษาความสะอาดไม่ดี ส่งผลให้ผิวหนังได้รับบาดเจ็บและติดเชื้อได้ง่าย

#### 2.1.4.2 อาการของแผลกดทับ

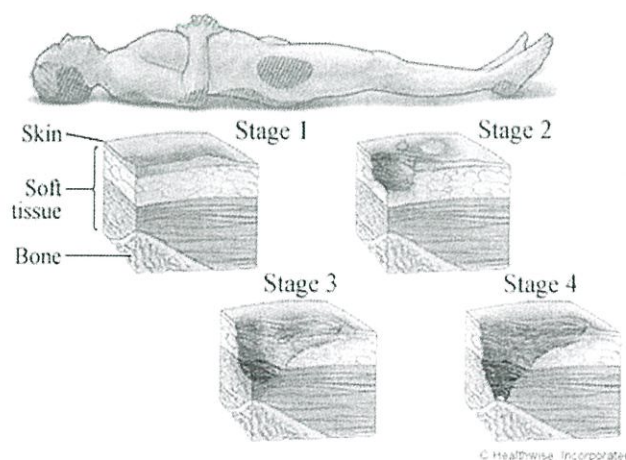
อวัยวะที่เสี่ยงเกิดแผลกดทับได้มากนั้นมักเป็นบริเวณที่ไม่มีไขมันปกคลุมผิวหนังมากและต้องรับแรงกดทับโดยตรง ผู้ป่วยที่เคลื่อนไหวร่างกายไม่ได้และต้องนอนบนเตียงตลอดเวลาเสี่ยงเกิดแผลกดทับที่ไหล่ ข้อศอก ท้ายทอย ข้างไบหู เข่า ข้อเท้า ส้นเท้า เท้า กระดูกสันหลัง หรือกระดูกก้นกบ ส่วนผู้ที่ต้องนั่งรถเข็นเป็นเวลานานเสี่ยงเกิดแผลกดทับที่ก้น หลังแขน หลังต้นขา หรือด้านหลังของกระดูกสะโพก โดยผู้ป่วยจะเกิดอาการหลายอย่าง ได้แก่ สีหรือลักษณะผิวหนังเกิดความผิดปกติ มีอาการบวม มีหนองออกมา เกิดอาการอุ่นหรือเย็นตรงผิวหนังที่เกิดแผลกดทับ และมักกดแล้วเจ็บบริเวณที่เป็นแผลกดทับดังแสดงในรูปที่ 2.2 ทั้งนี้ อาการของแผลกดทับจะรุนแรงขึ้นตามระยะต่าง ๆ ดังนี้

- ระยะที่ 1 แผลกดทับระยะนี้จะไม่เปิดออก มีลักษณะอุ่น นุ่มหรือแข็ง ผู้ป่วยอาจรู้สึกเจ็บและระคายเคือง ผิวหนังบริเวณแผลจะไม่มีสี ผู้ที่มีผิวขาวอาจเกิดรอยแดง ส่วนผู้ที่มีผิวเข้มอาจเกิดสีเขียวม่วง เมื่อกดลงไปบนแผล แผลจะไม่กลายเป็นสีขาว

- ระยะที่ 2 แผลกดทับระยะนี้เป็นแผลเปิดหรือมีแผลตุ่มน้ำพอง เนื่องจากหนังกำพร้าบางส่วนและหนังแท้ถูกทำลาย ส่งผลให้ผิวหนังหลุดลอก ผู้ป่วยอาจรู้สึกเจ็บที่แผลมากขึ้น

- ระยะที่ 3 แผลจะมีลักษณะเป็นโพรงลึก ซึ่งอาจเห็นไขมันที่แผล เนื่องจากผิวหนังทั้งหมดหลุดออกไป รวมทั้งเนื้อเยื่อที่อยู่ลึกลงไปชั้นผิวหนังถูกทำลาย

- ระยะที่ 4 แผลกดทับระยะนี้ถือว่าร้ายแรงที่สุด โดยผิวหนังทั้งหมดถูกทำลายอย่างรุนแรง รวมทั้งเนื้อเยื่อที่อยู่ล้อมรอบเริ่มตายหรือที่เรียกว่าเนื้อเยื่อตายเฉพาะส่วน (Tissue Necrosis) กล้ามเนื้อและกระดูกที่อยู่ลึกลงไปอาจถูกทำลายด้วย



รูปที่ 2.2 ระยะการเกิดแผลกดทับ

### 2.1.4.3 การป้องกันการเกิดแผลกดทับ

ผู้ที่เสี่ยงเกิดแผลกดทับสามารถป้องกันภาวะดังกล่าวได้ โดยดูแลตนเองด้านต่าง ๆ ได้แก่ การจัดทำทาง โภชนาการ ความสะอาดผิวหนัง และพฤติกรรมอื่น ๆ ดังนี้

- การจัดทำทาง การปรับเปลี่ยนท่าทางของผู้ป่วยอย่างสม่ำเสมอช่วยป้องกันการเกิดแผลกดทับได้ดี เนื่องจากวิธีนี้จะช่วยให้ผู้ป่วยเสี่ยงออกแรงกดทับจากการนอนหรือนั่งไปที่อวัยวะส่วนใดส่วนหนึ่งเป็นเวลานาน ซึ่งการจัดท่าทางสำหรับเลี่ยงการเกิดแรงกดทับทำได้ ดังนี้

- ควรปรับเปลี่ยนท่าทางขณะนั่งบนรถเข็นทุก ๆ 15 นาที และพลิกตัวเปลี่ยนท่านอนอย่างน้อยทุก 2 ชั่วโมง เพื่อถ่ายน้ำหนักตัวไม่ให้กดทับอวัยวะส่วนใดส่วนหนึ่งเป็นเวลานาน

- ผู้ป่วยที่ต้องนั่งรถเข็นอาจใช้แขนดันร่างกายส่วนบนให้ยกขึ้นเป็นบางครั้ง

- ควรเลือกรถเข็นที่ปรับระดับได้ เพื่อช่วยผ่อนแรงกดทับ

- ควรเลือกเบาะรองนั่งหรือเตียงนอนที่ช่วยผ่อนแรงกดและปรับท่าทางให้นั่งหรือนอนได้สบาย ปรับเตียงให้สูงขึ้นไม่เกิน 30 องศา เพื่อป้องกันไม่ให้ผิวหนังบริเวณก้นกดตึงรั้งกันจนเกิดแผลกดทับ

- โภชนาการ การรับประทานอาหารที่มีสารอาหารจำพวกโปรตีน วิตามิน และเกลือแร่อย่างหลากหลายจะช่วยป้องกันผิวหนังถูกทำลายและช่วยให้อาการป่วยหายเร็วขึ้น อย่างไรก็ตาม ผู้ป่วยแผลกดทับที่รู้สึกอยากอาหารน้อยลงอันเนื่องมาจากปัญหาสุขภาพที่เกิดขึ้นนั้นสามารถปรับพฤติกรรมการกินเพื่อให้ได้รับสารอาหารครบถ้วน ดังนี้

- แบ่งมื้ออาหารเป็นมื้อย่อย รวมทั้งรับประทานอาหารให้เป็นเวลา ซึ่งจะช่วยให้ได้สารอาหารที่จำเป็นต่อร่างกาย

- เลี่ยงดื่มเครื่องดื่มหรือน้ำเปล่าในปริมาณมากก่อนรับประทานอาหาร เนื่องจากจะทำให้รู้สึกอึดอัดเกินไป

- ดื่มน้ำที่อุณหภูมิเย็นไปสารอาหารหรือรับประทานอาหารอ่อน ๆ ในกรณีที่กลืนอาหารลำบาก

- ผู้ที่รับประทานอาหารมังสวิรัติควรรับประทานอาหารโปรตีนสูงอื่น ๆ ที่ไม่ใช่เนื้อสัตว์ เช่น ซีสโยเกิร์ต เนยถั่ว หรือถั่วต่าง ๆ

- ความสะอาดผิวหนัง ผู้ที่มีความเสี่ยงหรือป่วยเป็นโรคที่เสี่ยงเกิดแผลกดทับ ควรหมั่นตรวจผิวหนังของตนเองว่าเกิดการเปลี่ยนแปลงหรือความผิดปกติใด ๆ หรือไม่ รวมทั้งดูแลรักษาผิวหนังตนเอง ดังนี้

- ควรล้างทำความสะอาดผิวหนังและเช็ดให้แห้งอย่างสม่ำเสมอ ป้องกันไม่ให้ผิวหนังอับชื้น

- ควรดูแลผิวหนังสม่ำเสมอ เช่น ทาโลชั่นสำหรับผิวแห้ง เปลี่ยนเสื้อผ้าและผ้าปูที่นอนอย่างสม่ำเสมอ - ตรวจสอบตุ่มหรือตะเข็บของผ้าปูที่นอนให้ดี เพื่อป้องกันไม่ให้เสียดสีผิวหนัง

- พฤติกรรมอื่น ๆ ผู้ที่สูบบุหรี่หรือควรเลิกพฤติกรรมดังกล่าว เนื่องจากการสูบบุหรี่จะลดระดับออกซิเจนในเลือด และทำให้ระบบภูมิคุ้มกันร่างกายอ่อนแอ อันส่งผลให้เสี่ยงเกิดแผลกดทับได้

#### 2.1.4.4 วิธีการรักษาและดูแลผู้ป่วยแผลกดทับ

การรักษาแผลกดทับคือการช่วยให้ผู้ป่วยเกิดแรงกดทับที่อวัยวะส่วนใดส่วนหนึ่งน้อยลง ดูแลรักษาแผล บรรเทาอาการเจ็บแผล ป้องกันการติดเชื้อ และช่วยให้ผู้ป่วยมีโภชนาการที่ดี ทั้งนี้ แผลกดทับจัดเป็นปัญหาสุขภาพที่ซับซ้อน จำเป็นต้องมีกลุ่มรักษาที่ประกอบด้วยผู้เชี่ยวชาญหลายด้าน ได้แก่ แพทย์ผู้ดูแลแผนการรักษา แพทย์ผิวหนัง ศัลยแพทย์ด้านประสาท กระดูก และศัลยกรรมตกแต่ง พยาบาล นักกายภาพบำบัด นักกิจกรรมบำบัด และนักโภชนาการ

วิธีการรักษาแผลกดทับมีหลายวิธี ขึ้นอยู่กับระยะของอาการที่เป็น โดยผู้ป่วยที่มีแผลกดทับระยะที่ 1 และระยะที่ 2 สามารถหายได้หากได้รับการดูแลอย่างใส่ใจ ส่วนผู้ป่วยระยะที่ 3 และระยะที่ 4 อาจใช้เวลารักษานานกว่า วิธีการรักษาแผลกดทับแบ่งตามการรักษาอาการของโรค ได้แก่ การลดแรงกดทับ การดูแลแผล การรักษาเนื้อเยื่อตาย และการรักษาอื่น ๆ ดังนี้

- การลดแรงกดทับ วิธีการรักษาแผลกดทับขั้นแรกคือลดการกดทับอวัยวะที่เกิดภาวะดังกล่าว เพื่อไม่ให้เกิดแรงกดทับมากขึ้นและลดการเสียดสีของผิวหนัง ซึ่งทำได้ ดังนี้

- ควรปรับเปลี่ยนหรือขยับร่างกายบ่อย ๆ ผู้ที่นั่งรถเข็นควรขยับร่างกายทุก 15 นาที หรือเปลี่ยนท่านั่งทุกชั่วโมง ส่วนผู้ที่นอนบนเตียงควรเปลี่ยนท่านอนทุก 2 ชั่วโมง

- ใช้ที่นอนหรือเบาะรองนั่งที่ช่วยหนุนร่างกายให้นั่งหรือนอน โดยไม่ทำให้ผิวหนังตึงรั้งกันอันก่อให้เกิดแผลกดทับ

- การดูแลแผล การดูแลแผลกดทับขึ้นอยู่กับว่าแผลลึกมากน้อยแค่ไหน โดยทั่วไปแล้ว แผลกดทับสามารถดูแลรักษาได้ ดังนี้

- หากผิวหนังที่เกิดแผลกดทับไม่เปิดออกหรือเป็นแผลปิด ให้ล้างแผลด้วยผลิตภัณฑ์ทำความสะอาดอ่อน ๆ และเช็ดให้แห้ง ส่วนผู้ที่แผลเปิดออกให้ล้างด้วยน้ำเกลือสำหรับล้างแผลทุกครั้งเมื่อต้องทำแผล

- พันแผลเพื่อช่วยให้แผลหายเร็วขึ้น เนื่องจากช่วยให้แผลชุ่มชื้นอยู่เสมอ อีกทั้งยังลดการเสียดสีที่ผิวหนัง

- การรักษาเนื้อเยื่อตาย แผลกดทับจะรักษาให้หายได้นั้นต้องไม่เกิดการติดเชื้อหรือเนื้อเยื่อตาย ผู้ป่วยที่มีเนื้อเยื่อตายจะได้รับการรักษา ดังนี้

- การผ่าตัดเนื้อตาย แพทย์จะทำความสะอาดแผลและตัดเนื้อเยื่อที่ถูกทำลายออกไป

ยาปฏิชีวนะ ผู้ป่วยอาจได้รับยาปฏิชีวนะเพื่อรักษาอาการติดเชื้อและป้องกันไม่ให้เกิดการลุกลาม

- ครีมหรือซีมี้งสำหรับทา รักษา ช่วยให้แผลหายไวขึ้นและป้องกันเนื้อเยื่ออื่นถูกทำลาย

- การดูแลอื่น ๆ ผู้ป่วยอาจต้องได้รับการดูแลรักษาจากผู้เชี่ยวชาญหลายสาขาวิชาชีพ ได้แก่ แพทย์ พยาบาล นักกายภาพบำบัด โภชนาการ และศัลยแพทย์ตกแต่ง

- การรักษาอื่น ๆ ผู้ป่วยแผลกดทับอาจได้รับการรักษาอื่น ๆ ร่วมด้วย เพื่อบรรเทาอาการต่าง ๆ ดังนี้

- ยาแก้ปวดเฉพาะที่ ผู้ป่วยอาจได้รับยาแก้ปวดเฉพาะที่หรือยาบรรเทาอาการปวดที่ไม่ผสมสารสเตียรอยด์ เช่น ไอบูโพรเฟน หรือนาพรอกเซน

- อาหารเสริม แพทย์อาจให้ผู้ป่วยรับประทานอาหารเสริมอื่น ๆ ร่วมด้วย เช่น โปรตีน สังกะสี และวิตามิน เพื่อช่วยให้แผลหายเร็วขึ้น หากร่างกายขาดวิตามินหรือแร่ธาตุดังกล่าว จะส่งผลให้เกิดแผลกดทับที่ผิวหนังได้ง่าย

- การผ่าตัด ผู้ป่วยแผลกดทับที่รักษาแผลให้หายไม่ได้ จำเป็นต้องเข้ารับการผ่าตัด โดยนำส่วนของกล้ามเนื้อ ผิวน้ำ หรือเนื้อเยื่อในร่างกายตนเองมาปิดแผลและใส่รองกระดูกที่ได้รับผลกระทบจากแผลกดทับ

## 2.2 ภาวะเหนื่อยล้าของผู้ดูแล (Caregiver burden)

ในบางครั้งผู้ที่ทำหน้าที่ดูแลผู้ป่วยมักถูกมองข้าม ซึ่งมักลืมว่าจริงๆ แล้วการดูแลนั้นบางครั้งยาก และเหนื่อยยิ่งกว่าการรักษาเสียอีก เพราะเป็นงานที่ต้องอยู่กับผู้ป่วยตลอดเวลา การที่ผู้ป่วย จะได้รับ ยา อาหารหรือ การดูแลด้านต่างๆตามแพทย์สั่งหรือไม่ ก็ขึ้นอยู่กับผู้ดูแลเหล่านี้ด้วย แพทย์จึงไม่ควร ละเลยความสำคัญของบุคคลกลุ่มนี้

ปัจจุบัน ความเจริญด้านการแพทย์ และเทคโนโลยี ทำให้มนุษย์เราอายุยืนมากขึ้น มีประชากร สูงอายุเพิ่มขึ้น อัตราการป่วยด้วยโรคเรื้อรังที่เพิ่มขึ้นนำมาซึ่งจำนวนผู้ป่วยสูงอายุที่ทุพพลภาพและ ต้องการการดูแลมากขึ้น หน้าที่ของการดูแลผู้ป่วยเหล่านี้ ส่วนใหญ่มักตกหนักที่สมาชิกในครอบครัว ของผู้ป่วยเอง ซึ่งมักจะเป็นบทบาททางสังคมของผู้หญิง คนสูงอายุ หรือคนโสด ที่สมาชิกคนอื่นมักจะ มอบหมายภาระนี้ให้(วีจนา, 2561)

ภาระการดูแลผู้สูงอายุหรือผู้ป่วยโรคเรื้อรังที่ต้องพึ่งพาผู้ดูแลตลอดเวลา นั้น เป็นงานที่เหน็ด เหนื่อย ใช้เวลา และก่อให้เกิดความตึงเครียดสูง อีกทั้งตัวผู้ดูแลเองก็อาจมีปัญหาของตนเองใน หลายๆด้าน ไม่ว่าจะเป็นปัญหาสุขภาพ ความเป็นอยู่ ฐานะการเงิน และชีวิตส่วนตัวของตนเอง เมื่อเวลาผ่านไป ถ้าผู้ดูแลปราศจากความช่วยเหลือจากสมาชิกคนอื่นแล้ว ก็จะทำให้เกิดความเหนื่อย ล้า (Caregiver burden) หรือถึงกับหมดไฟในการดูแลไปเลย (Caregiver burnout) เมื่อเกิดภาวะ ดังกล่าว ผู้ป่วยอาจเสี่ยงต่อการถูกทำร้าย หรือทอดทิ้งได้โดยไม่ได้ตั้งใจ ซึ่งถ้าบุคคลากรทางการแพทย์ ไม่เข้าใจเรื่องราวโดยตลอด ก็มักจะกล่าวโทษผู้ดูแลว่า ‘ดูแลไม่ดี ไม่กตัญญู ใจร้าย ทำคนป่วยได้ลง คอ’ ซึ่งก็ยิ่งเป็นการซ้ำเติมความรู้สึกผิด และความรู้สึกเป็นภาระให้กับผู้ดูแลมากขึ้นไปอีก

### 2.2.1 ความเสี่ยงที่จะเกิดภาวะเหนื่อยล้าของผู้ดูแล

ขึ้นกับปัจจัยต่างๆ ดังต่อไปนี้

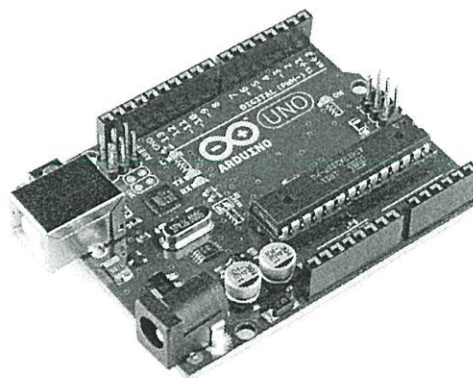
- ระดับความรุนแรงของภาวะทุพพลภาพ และความสามารถในการช่วยเหลือตนเอง ของผู้ป่วย
- ปัญหาสุขภาพของตัวผู้ดูแลเอง
- แหล่งให้ความช่วยเหลืออื่นๆ
- บทบาทและหน้าที่ความรับผิดชอบด้านอื่นๆของผู้ดูแล

## 2.2.2 แนวทางสำหรับผู้ดูแลที่เริ่มเหนื่อยล้า

- วางแผนการดูแลให้ดี อย่าให้เป็นภาระตหนักที่ใครเพียงคนเดียวตลอดเวลา
- ถ้าเป็นไปได้ อาจจ้างผู้อื่นให้มาทำหน้าที่ดูแลชั่วคราวเป็นครั้งๆ เพื่อให้ผู้ดูแลหลักมีเวลาพักผ่อนหรือทำธุระส่วนตัวบ้าง
  - แบ่งหน้าที่ด้านต่างๆ ให้ญาติพี่น้องคนอื่นได้มีส่วนรับผิดชอบร่วมกันบ้าง เช่น ภาระค่าใช้จ่ายในบ้าน การทำความสะอาดบ้าน หรือหน้าที่พาผู้ป่วยมาโรงพยาบาล
  - หาเวลาพักผ่อนไปทำกิจกรรมที่ตนเองชอบบ้างเพื่อผ่อนคลายความเครียด
  - พุดคุยพบปะสังสรรค์ เข้าสังคมบ้าง นอกจากบรรเทาความเครียดแล้วอาจได้รับคำแนะนำในการแก้ปัญหาของตนเองได้
  - ดูแลรักษาสุขภาพกายของตนเองให้แข็งแรงอยู่เสมอ
  - รู้จักปล่อยวางเสียบ้าง อย่าคาดหวังกับสิ่งรอบตัวสูงจนเกินไป เช่น อยากให้ผู้ป่วยหายขาดจนลุกมาเดินได้ กินเองได้ อยากให้ญาติทุกคนมาช่วยดูแลกันพร้อมหน้าตลอดเวลา
  - แบ่งเวลาให้กับบุคคลอันเป็นที่รักและบุคคลในครอบครัวของตนเองบ้าง แทนที่จะทุ่มเทเวลาทั้งหมดให้ผู้ป่วยคนเดียว
  - สำหรับบางกรณี อาจจำเป็นต้องไปฝากผู้ป่วยไว้ในสถานพยาบาลบ้าง หากผู้ดูแลติดธุระหรือรู้สึกเกินกำลังแล้ว

## 2.3 Arduino Uno

Arduino อ่านว่า (อา-ดู-อิ-โน้ หรืออาดูยโน้) เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ตัวอย่างบอร์ด Arduino แสดงในรูปที่ 2.3 ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ดหรือโปรแกรมต่อได้อีกด้วย

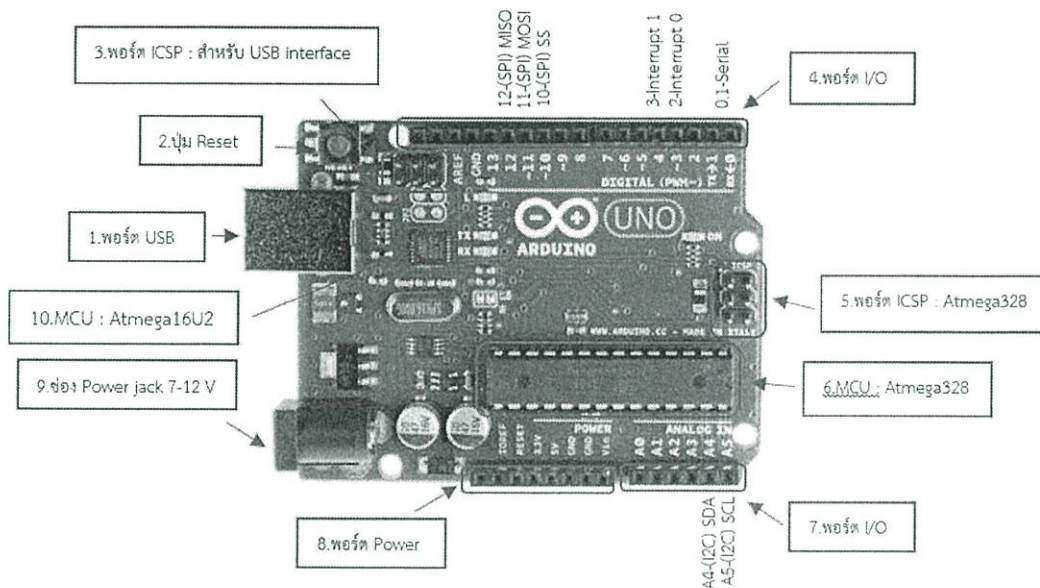


รูปที่ 2.3 บอร์ด Arduino Uno

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ดหรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น

### 2.3.1 ขาต่างๆของ Arduino Uno

รูปที่ 2.4 แสดงขาต่างๆของบอร์ด Arduino Uno ที่ใช้ในการทำงานเพื่อป้อนข้อมูล input หรือส่งข้อมูล output ออกไป

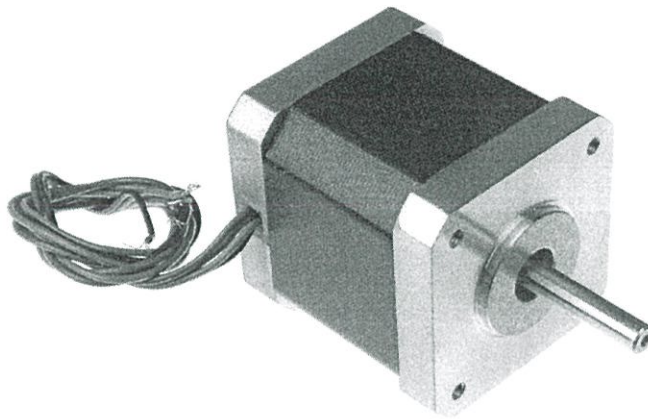


รูปที่ 2.4 ขาต่างๆของบอร์ด Arduino Uno

- USB Port : ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
- Reset Button : เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
- ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2
  - I/O Port : Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆเพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM
  - ICSP Port : Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
  - MCU : Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino
  - I/O Port : นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อกตั้งแต่ขา A0-A5
    - Power Port : ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอกประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, Vin
    - Power Jack : รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
    - MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega 328 จะติดต่อกับ Computer ผ่าน Atmega16U2

## 2.4 Stepper motor

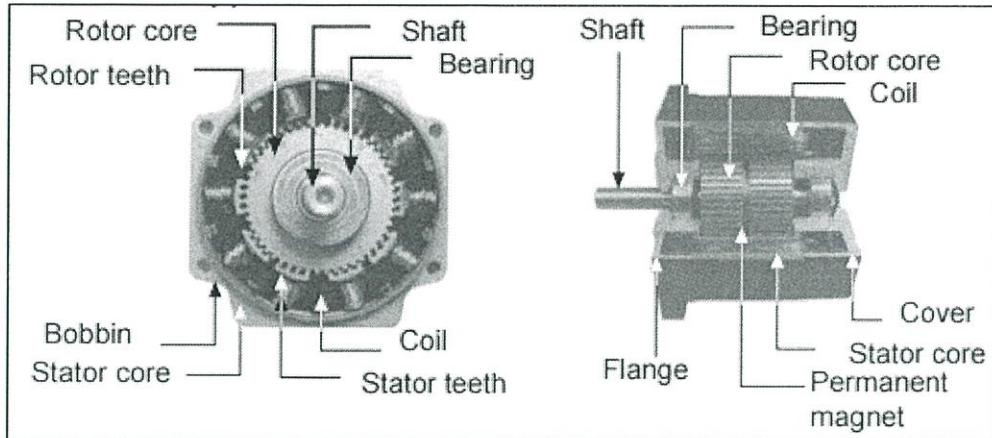
สเต็ปเปอร์มอเตอร์ (Stepper Motor) เป็นอุปกรณ์เอาต์พุตอย่างหนึ่ง ซึ่งสามารถนำไอซี ไมโครคอนโทรลเลอร์ มาทำการควบคุมได้สะดวก และเป็นมอเตอร์ที่เหมาะสมสำหรับใช้ในงาน ควบคุมการหมุน ที่ต้องการตำแหน่ง และทิศทางที่แน่นอน การทำงานของ สเต็ปเปอร์มอเตอร์จะ ขับเคลื่อนทีละขั้นๆละ (Step) 0.9, 1.8, 5, 7.5, 15 หรือ 50 องศา ซึ่งขึ้นอยู่กับคุณสมบัติแต่ละชนิด ของสเต็ปเปอร์มอเตอร์ตัวนั้นๆ Stepper Motor จะแตกต่างจากมอเตอร์กระแสตรงทั่วไป ( DC MOTOR ) โดยการทำงานของมอเตอร์กระแสตรงจะหมุนไปแบบต่อเนื่อง ไม่สามารถหมุนเป็นแบบส เต็ปๆ ได้ดังนั้นในการนำไปกำหนดตำแหน่งจึงควบคุมได้ยากกว่า แต่ในส่วนใหญ่จะใช้สเต็ปเปอร์ มอเตอร์มาทำการการควบคุมโดยใช้วิธีในระบบดิจิทัล เช่น พรินเตอร์ ( Printer ) พล็อตเตอร์ ( X-Y Plotter ) ดิสก์ไดรฟ์ ( Disk drive ) ฯลฯ ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 Stepper Motor

### 2.4.1 หลักการทำงานของ Stepper Motor

Stepper Motor เป็นมอเตอร์ไฟฟ้าที่ขับเคลื่อนด้วยพัลส์ โดยโครงสร้างภายในนั้นจะ ประกอบไปด้วยขั้วแม่เหล็กบนสเตเตอร์ (Stator) ทำมาจากแผ่นเหล็กวงแหวน จะมีซี่ยื่นออกมา ประกอบกันเป็นขั้นๆ โดยแต่ละซี่ที่ยื่นออกมานั้นจะมีขดลวด (คอยล์) พันอยู่ เมื่อมีกระแสผ่านคอยล์ จะเกิดสนามแม่เหล็กไฟฟ้าขึ้น ดังแสดงในรูปที่ 2.6



รูปที่ 2.6 โครงสร้าง Stepping Motor

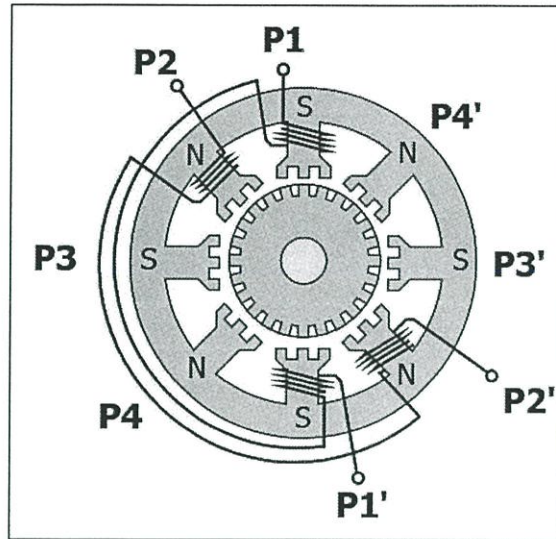
ในการทำงานของ Stepper Motor นั้นจะไม่สามารถขับเคลื่อนหรือทำงานเองได้ จำเป็นต้องมี วงจรอิเล็กทรอนิกส์ที่ใช้ในการสร้างสัญญาณหรือจ่ายพัลส์ไปให้วงจรขับสเต็ปมอเตอร์ (Stepper Motor Drive) ดังแสดงในรูปที่ 2.7 การสร้างสัญญาณนั้นจะต้องสร้างและเรียงลำดับของสัญญาณ ด้วยและอีกสิ่งที่สำคัญคือการดูตำแหน่งของสายที่ทำการต่อเข้ากับตัวสเต็ปมอเตอร์



รูปที่ 2.7 แผนผังการทำงานของ Stepper Motor

#### 2.4.2 ชนิดของ Stepper Motor

ในอดีตมีการแบ่งชนิดของ Stepper Motor ตามลักษณะโครงสร้างซึ่งแบ่งได้เป็น 3 ชนิดคือ แบบแม่เหล็กถาวรหรือ PM (Permanent Magnet), แบบปรับค่าความต้านทานแม่เหล็กได้ หรือ VR (Variable Reluctance) และแบบผสมหรือไฮบริด (Hybrid) ซึ่งเป็นการผสมกันระหว่าง แบบ PM และ VR ในปัจจุบันนี้ Stepper Motor ส่วนใหญ่เป็นแบบไฮบริด เนื่องจากสามารถทำให้มีความละเอียดในการเคลื่อนที่ของแกนได้สูงถึง 0.9 องศาต่อสเต็ป (ซึ่งเป็นข้อดีของแบบ VR) และให้แรงบิดหรือทอร์กที่สูง โดยใช้พลังงานต่ำ (เป็นข้อดีของแบบ PM) ดังแสดงในรูปที่ 2.8 แสดงโครงสร้างอย่างง่ายของ Stepper Motor แบบไฮบริด

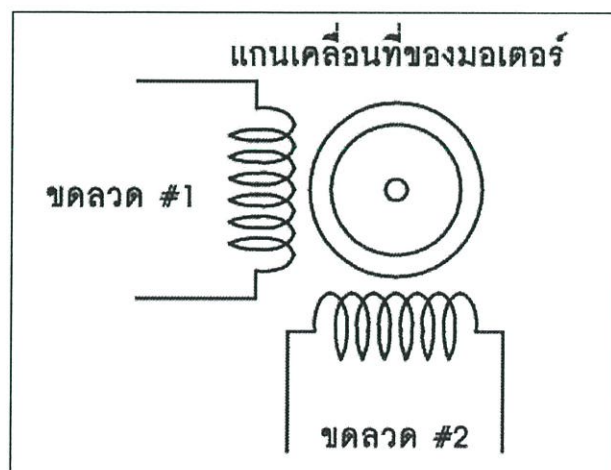


รูปที่ 2.8 โครงสร้างอย่างง่ายของ Stepper Motor ชนิดไฮบริด

ดังนั้นการกำหนดชนิดของ Stepper Motor ในยุคต่อมาจนถึงปัจจุบันจึงพิจารณาที่ลักษณะของการฟันขดลวด, การต่อสายออกมาใช้งาน และวงจรขับ ซึ่งแบ่งเป็น 2 ชนิดหลักๆ คือ ชนิดไบโพลาร์ (bipolar) และชนิดยูนิโพลาร์ (uni-polar)

- Stepper Motor แบบไบโพลาร์

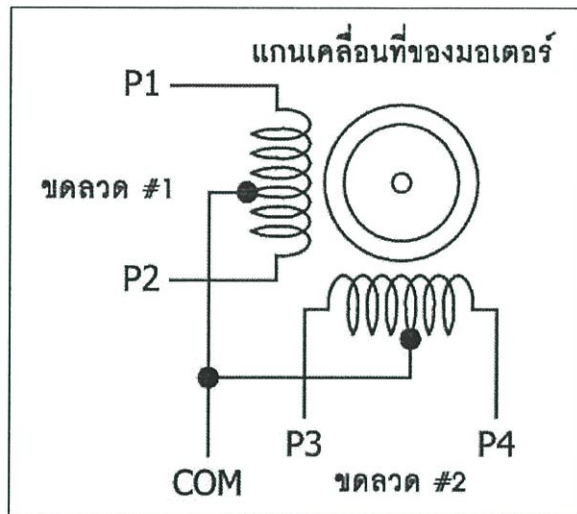
มีลักษณะการฟันขดลวดของมอเตอร์ดังแสดงในรูปที่ 2.9 แบ่งออกเป็น 2 ชุดที่ไม่มีแท็บ กลาง ทำให้บางครั้งจึงเรียก Stepper Motor แบบนี้ว่า เป็น Stepper Motor แบบ 2 เฟส การขับให้มอเตอร์แบบนี้หมุนจะต้องป้อนแรงดันต่างขั้วกันให้แก่ขดลวดแต่ละชุด ทำให้วงจรขับ Stepper Motor แบบนี้ค่อนข้างซับซ้อน



รูปที่ 2.9 ลักษณะการฟันขดลวด Stepper Motor ไบโพลาร์

- Stepper Motor แบบยูนิโพลาร์

มีลักษณะการพันขดลวดของมอเตอร์ดังแสดงในรูปที่ 2.10 มีด้วยกัน 2 แบบคือ แบบ 5 และ 6 สาย บางครั้งเรียก Stepper Motor แบบนี้ว่า เป็น Stepper Motor แบบ 4 เฟส การขับจะต้องป้อนสัญญาณเข้าที่ขั้วหรือเฟสของมอเตอร์ให้เรียงลำดับอย่างถูกต้อง มอเตอร์จึงจะสามารถหมุนได้อย่างราบรื่น Stepper Motor แบบนี้มีการพันขดลวด 2 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์ แต่ละขดแบ่งเป็น 2 เฟส รวมมอเตอร์ทั้งตัวจะมี 4 เฟสคือ เฟส 1, 2, 3 และ 4 มีการต่อสายออกมาจากขดลวดแต่ละขดเพื่อจ่ายไฟเลี้ยง ทำให้ Stepper Motor แบบนี้มีทั้งแบบ 5 สายและ 6 สาย ถ้าเป็นแบบ 5 สาย จะเป็นการนำสายไฟเลี้ยงของขดลวดทั้งสองมาต่อรวมกันเป็นสายเดียว สำหรับในบทความนี้จะเน้นหนักไปที่สตีปเปอร์แบบยูนิโพลาร์นี้ เนื่องจากสามารถหาได้ง่ายกว่า และใช้วงจรขับที่มีความซับซ้อนน้อยกว่ามาก



รูปที่ 2.10 ลักษณะการพันขดลวด Stepper Motor ยูนิโพลาร์

### 2.4.3 การควบคุมการหมุนของ Stepper Motor แบบ Unipolar

สามารถทำได้โดย ควบคุมกระแสไฟที่จ่ายให้กับขดลวดในแต่ละเฟส อย่างเป็นลำดับที่แน่นอน โดยถ้าหากต้องการให้กระแสไหลในเฟสใดๆ ก็จะทำให้สถานะของเฟสนั้นๆ เป็นสถานะลอจิก “1” การควบคุมการหมุนของ Stepper Motor สามารถทำได้ 3 วิธีดังนี้

- การควบคุมการหมุนแบบเฟสเดียว (Single Phase) หรือ Wave Drive

เป็นการป้อนกระแสไฟให้กับขดลวดของ Stepper Motor ทีละขด โดยจะป้อนกระแสเรียงตามลำดับกันไปตามตาราง 1-1 (ตัวเลข 1 หมายถึง มีการจ่ายแรงดันให้กับ ขดลวด 0 คือไม่จ่ายแรงดันให้กับขดลวด) ดังนั้นกระแสที่ไหลในขดลวดจะทำการไหลในทิศทางเดียวกันทุกขด ลักษณะเช่นนี้จึงทำให้แรงขับของ Stepper Motor มีน้อย จึงไม่เป็นที่นิยมใช้

ตารางที่ 2.1 การควบคุมการหมุนแบบเฟสเดียว

Step	Phase A	Phase B	Phase A'	Phase B'
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	Back to step 1			

- การควบคุมการหมุนแบบสองเฟส (Two Phase) หรือ Full Step Drive เป็นการป้อนกระแสไฟให้กับขดลวดของ Stepper Motor ทั้ง 2 ขด พร้อมๆ กันไป และป้อนกระแสเรียงตามลำดับกันไป ตามตาราง 1-2 ดังนั้นจึงมีกระแสไหลในขดลวดของมอเตอร์มากขึ้น และจะทำให้มอเตอร์มีแรงบิดมากขึ้น

ตารางที่ 2.2 การควบคุมการหมุนแบบสองเฟส

Step	Phase A	Phase B	Phase A'	Phase B'
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1
5	Back to step 1			

- การควบคุมการหมุนแบบครึ่งเฟส (Half Step) เป็นการป้อนกระแสแบบ Two Phase และ Single Phase สลับกันไป ตามตาราง 1-3 ซึ่งเป็นการเพิ่มความละเอียดของตำแหน่งในการหมุน

### ตารางที่ 2.3 การควบคุมการหมุนแบบครึ่งเฟส

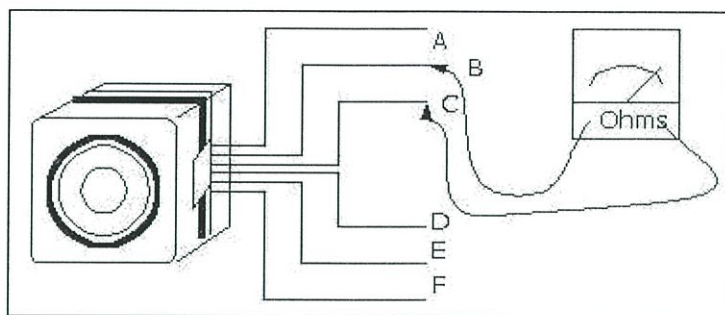
Step	Phase A	Phase B	Phase A'	Phase B'
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	0	0	0	1
9	Back to step 1			

ความเร็วการหมุนของแกน Stepper Motor จะขึ้นอยู่กับกำหนดช่วงเวลา (Time Delay) ในแต่ละลำดับ ถ้า Time Delay มีค่าน้อย แกนจะหมุนเร็วขึ้น ถ้า Time Delay มีค่ามาก แกนจะหมุนช้าและหมุนไม่สม่ำเสมอ

#### 2.4.4 วิธีการตรวจสอบหาเฟสของขดลวด Stepper Motor

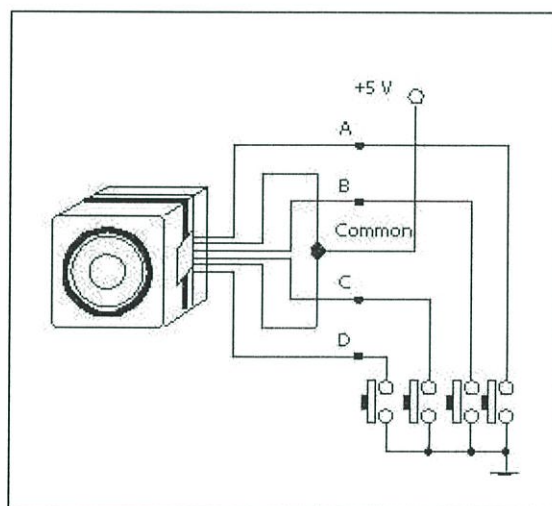
ในขั้นตอนที่ 1 ให้สังเกตว่า Stepper Motor ที่นำมาทดลองที่เป็นแบบยูนิโพลาร์ (Uni-polar stepper motor) จะมีจำนวนสาย 5 เส้น หรือ 6 เส้น

ในขั้นตอนที่ 2 ใช้มิเตอร์วัดค่าความต้านทานของเส้นลวดในแต่ละขด ขั้นตอนการวัด ให้หาสายที่ต่อเป็นจุดร่วมเสียก่อน(common) โดยให้ใช้ มัลติมิเตอร์ตั้งค่าไว้สำหรับการวัดค่าความต้านทานแต่ละเส้น ดังแสดงในรูปที่ 2.11 สังเกตที่ค่าความต้านทาน ถ้าหากไม่ได้วัดระหว่าง จุดต่อร่วม(common) กับสายแต่ละเส้น ค่าความต้านทานจะมีค่าเป็น 2 เท่าของการวัดระหว่างจุดต่อร่วมกับสายที่ใช้งาน ตัวอย่าง เช่น ถ้าให้จุด B เป็นจุดร่วม หากวัดระหว่างที่จุด A กับจุด B จะมีค่าเท่ากับ 60 Ohm แต่ถ้าวัดระหว่างที่จุด A และจุด C ซึ่งไม่ใช่จุดร่วมก็จะได้ค่าเท่ากับ 120 Ohm หากเป็นแบบที่มีสาย 6 เส้นก็จะมีจุดร่วมสองจุด เพราะมีขดลวดคนละชุดกัน และสายที่เป็นจุดร่วมส่วนใหญ่จะมีสีเหมือนกัน ทำนองเดียวกันหากเป็นแบบที่มีสาย 5 เส้นก็จะมีจุดร่วมเพียงจุดเดียวเท่านั้น



รูปที่ 2.11 การใช้มิเตอร์วัดค่าความต้านทาน

ในขั้นตอนที่ 3 หากเป็นแบบที่มีสาย 6 เส้นก็ให้ทำการต่อจุดร่วมเข้าด้วยกันจะได้เป็น 5 เส้น แล้วต่อวงจรในรูปที่ 2.12 หลังจากนั้นให้ทดลองกดสวิตช์ ที่ต่อเข้ากับแต่ละจุดโดยเริ่มที่ จุด A จุด B จุด C และจุด D แล้วให้สังเกตการหมุนของ Stepper Motor ว่าหมุนได้ต่อเนื่องหรือไม่ หากมีการกระโดดข้ามสเต็ปก็ให้ทดลองโดยเรียงลำดับการกด สวิตช์ใหม่ จนหาลำดับของสายได้ถูกต้องคือมอเตอร์เดินตามที่ละสเต็ป อย่างเป็นลำดับ



รูปที่ 2.12 แสดงการต่อวงจรเพื่อทดสอบโดยการ สวิตช์เพื่อหาลำดับ

#### 2.4.5 Stepper Motor Driver Board

Stepper Motor Driver เป็นโมดูลที่ใช้ในการขับ Stepper Motor สามารถกำหนดค่าการทำงานโดยทำการปรับสวิตช์ สามารถทำงานได้ 5-24 VDC, 0.3-3A โดยจะแบ่ง switch setting ออกเป็น 4 กลุ่มดังนี้

- Running Current (switch สำหรับการตั้งค่ากระแสการทำงานให้เหมาะสมกับ stepper motor) โดยจะใช้ switch ทั้งหมด 4 ตัว ได้แก่ SW1,SW2,SW3,S4

ตารางที่ 2.4 Running Current

Running Current														
(A)	0.3	0.5	0.8	1	1.1	1.2	1.4	1.5	1.6	1.9	2	2.2	2.6	3
SW1	OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	ON	ON	ON	ON	ON	ON
SW2	OFF	OFF	ON	ON	ON	OFF	ON	OFF	OFF	ON	OFF	ON	ON	ON
SW3	ON	ON	OFF	OFF	ON	OFF	ON	ON	OFF	OFF	ON	ON	OFF	ON
S1	ON	OFF	ON	OFF	ON	ON	OFF	ON	OFF	ON	OFF	ON	OFF	OFF

ตารางที่ 2.5 Stop Current

Stop Current	
	S2
20%	ON
50%	OFF

- Excitation Mode (เป็น mode ปรับความละเอียดในการหมุนให้กับ stepper motor)

ตารางที่ 2.6 Excitation Mode

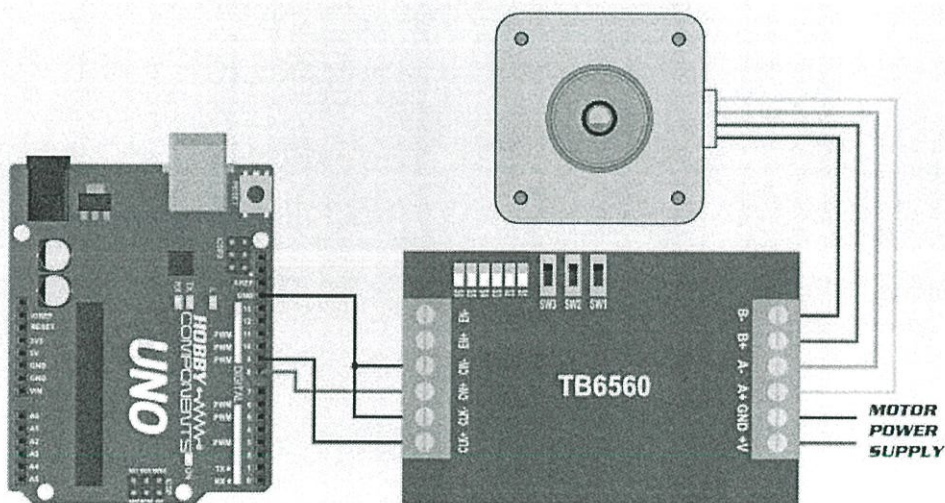
Excitation Mode		
Step	S3	S4
whole	OFF	OFF
half	ON	OFF
1/8	ON	ON
1/16	OFF	ON

- Decay Setting (เป็น mode ควบคุมการ discharge กระแส ในโหมด PWM เพื่อลดการสั่นของ stepper motor มี 4 ระดับ)

ตารางที่ 2.7 Decay Setting

Decay Setting		
	S5	S6
0%	OFF	OFF
25%	ON	OFF
50%	OFF	ON
100%	ON	ON

รูปที่ 2.13 แสดงการเชื่อมต่อระหว่าง Stepper motor ร่วมกับกับ Stepper Motor Driver และ Arduino เพื่อให้ Arduino สั่งการทำงานผ่าน Stepper Motor Driver ในการขับเคลื่อน Stepper Motor



รูปที่ 2.13 การเชื่อมต่อระหว่าง Stepper Motor, Stepper Motor Driver, Arduino Uno

## 2.5 Liquid Crystal Display (LCD)

จอ Liquid Crystal Display (LCD) เป็นจอแสดงผลรูปแบบหนึ่งที่ยิมนำมาใช้งานกับระบบสมอง กลฝังตัวอย่างแพร่หลาย จอ LCD มีทั้งแบบแสดงผลเป็นตัวอักษรเรียกว่า Character LCD ซึ่งมีการกำหนด ตัวอักษรหรืออักขระที่สามารถแสดงผลไว้ได้อยู่แล้ว และแบบที่สามารถแสดงผลเป็นรูปภาพหรือสัญลักษณ์ได้ ตามความต้องการของผู้ใช้งานเรียกว่า Graphic LCD นอกจากนี้บางชนิดเป็นจอที่มีการผลิตขึ้นมาใช้ เฉพาะงาน ทำให้มีรูปแบบและรูปร่างเฉพาะเจาะจงในการแสดงผล เช่น นาฬิกาดิจิตอล เครื่องคิดเลข หรือหน้าปัดวิทยุ เป็นต้น

### 2.5.1 โครงสร้างโดยทั่วไปของ LCD

โครงสร้างของ LCD ทั่วไปจะประกอบขึ้นด้วยแผ่นแก้ว 2 แผ่นประกบกันอยู่ โดยเว้นช่องว่างตรง กลางไว้ 6-10 ไมโครเมตร ผิวด้านในของแผ่นแก้วจะเคลือบด้วยตัวนำไฟฟ้าแบบใสเพื่อใช้แสดงตัวอักษร ตรง กลางระหว่างตัวนำไฟฟ้าแบบใสกับผลึกเหลวจะมีชั้นของสารที่ทำให้โมเลกุลของผลึกรวมตัวกันในทิศทางที่แสง ส่งมากระทบเรียกว่า Alignment Layer และผลึกเหลวที่ใช้โดยทั่วไปจะเป็นแบบ Magnetic โดย LCD สามารถแสดงผลให้มองเห็นได้ทั้งหมด 3 แบบด้วยกันคือ

- แบบใช้การสะท้อนแสง (Reflective Mode) LCD แบบนี้ใช้สารประเภทโลหะเคลือบอยู่ที่แผ่นหลัง ของ LCD ซึ่ง LCD ประเภทนี้เหมาะกับการนำมาใช้งานในที่ที่มีแสงสว่างเพียงพอ

- แบบใช้การส่งผ่าน (Transitive Mode) LCD แบบนี้วางหลอดไฟไว้ด้านหลังจอ เพื่อให้การอ่าน ค่าที่แสดงผลทำได้ชัดเจน

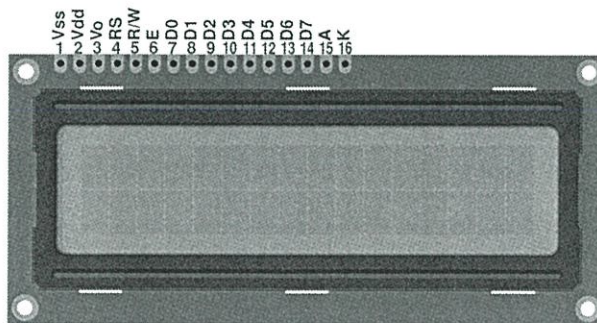
- แบบส่งผ่านและสะท้อน (Transflective Mode) LCD แบบนี้เป็นการนำเอาข้อดีของจอแสดงผล LCD ทั้ง 2 แบบมารวมกับ LCD ที่แสดงผลเป็นอักขระหรือตัวอักษร ตามท้องตลาดทั่วไปจะมีหลายแบบด้วยกัน มีทั้ง 16 ตัวอักษร 20 ตัวอักษรหรือมากกว่า และจำนวนบรรทัดจะมีตั้งแต่ 1 บรรทัด 2 บรรทัด 4 บรรทัดหรือมากกว่า ตามแต่ความต้องการและลักษณะของงานที่ใช้ หรืออาจจะมีแบบสั่งทำเฉพาะงานก็เป็นได้ ยกตัวอย่างจอ LCD ขนาด 16x2 Character หรือที่นิยมเรียกกันว่า จอ LCD 16 ตัวอักษร 2 บรรทัด สามารถ

หาซื้อได้ง่ายและมีราคาไม่สูง เหมาะสมกับการใช้งานแสดงผลไม่มากในหน้าจอดีียว

จอ LCD 16x2 Character ที่นิยมวางจำหน่ายจะมีอยู่ 2 แบบด้วยกันคือ LCD แบบปกติที่เชื่อมต่อ แบบขนาน (Parallel) และ LCD แบบที่เชื่อมต่ออนุกรม (Serial) แบบ I2C โดยทั้ง 2 แบบตัวจอก็มีลักษณะ เดียวกันเพียงแต่แบบ I2C จะมีบอร์ดเสริมทำให้สื่อสารแบบ I2C ได้เชื่อมต่อได้สะดวกขึ้น

- แบบ Parallel มี 16 ขา

รูปที่ 2.14 แสดงตัวอย่างด้านหน้าของ LCD 16 ตัวอักษร 2 บรรทัด โดยจะเชื่อมต่อจอ LCD เข้ากับบอร์ด Arduino โดยตรง แบ่งเป็นการเชื่อมต่อแบบ 4 บิต และการเชื่อมต่อแบบ 8 บิต ใน Arduino จะนิยมเชื่อมต่อแบบ 4 บิต เนื่องจากใช้สายในการเชื่อมต่อน้อยกว่า



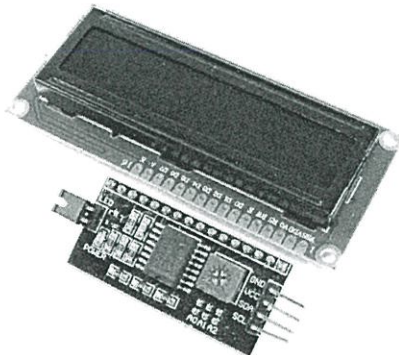
รูปที่ 2.14 ด้านหน้าจอ LCD 16x2 (Parallel)

ตารางที่ 2.8 ขาของจอ LCD 16x2 แบบ Parallel

Pin No	Symbol	Description
1	VSS/GND	Ground
2	VDD	+5VDC
3	VO/VEE	LCD Control สำหรับปรับความเข้มของตัวอักษร
4	RS	Register Select เป็นขาอินพุตสำหรับเลือกเขียนอ่านข้อมูลในรีจิสเตอร์
5	RW	Read/Write เป็นขาอินพุตสำหรับเลือกโหมดเขียนหรืออ่านข้อมูล
6	E/EN	Enable เป็นขาอินพุตสำหรับสัญญาณ Pulse เมื่อต้องการเขียนหรืออ่านข้อมูล
7	DB0	Data Pins 8-Bit
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	A	(LED+) เป็นขา $V_{CC}$ สำหรับ LED backlight (5V)
16	K	(LED-) เป็นขา $Gnd$ สำหรับ LED backlight (Gnd)

- แบบ I2C มี 4 ขา

รูปที่ 2.15 แสดงด้านหน้าและด้านหลังของจอ LCD I2C ซึ่งจะเชื่อมต่อกับจอ LCD ผ่านโมดูลแปลงรูปแบบการเชื่อมต่อกับจอ LCD จากแบบขนาน มาเป็นการเชื่อมต่อแบบอื่นที่ใช้สายน้อยกว่า เช่น การใช้โมดูล I2C Serial Interface ทำให้สามารถใช้งานได้สะดวกขึ้น



รูปที่ 2.15 ด้านหน้าและด้านหลังจอ LCD 16x2 (I2C)

ตารางที่ 2.9 ขาของจอ LCD 16x2 แบบ I2C

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

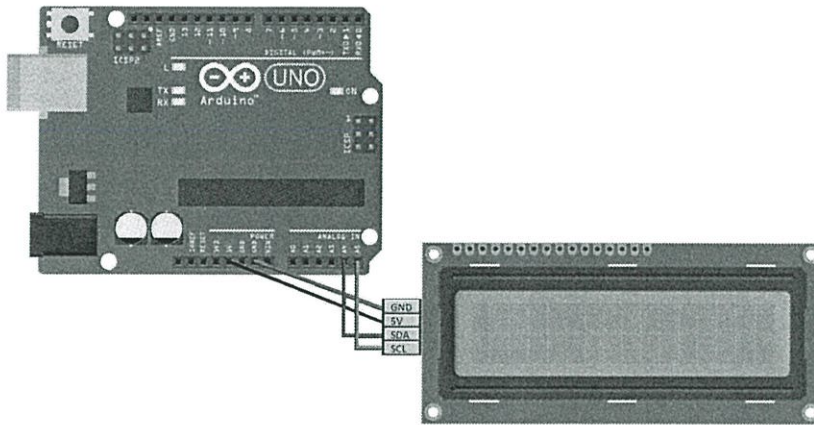
### 2.5.2 การควบคุมการแสดงผลของจอ LCD (I2C)

ในการควบคุมหรือสั่งงาน โดยทั่วไปจอ LCD จะมีส่วนควบคุม (Controller) อยู่ในตัวแล้ว ผู้ใช้ สามารถสั่งรหัสคำสั่งสำหรับควบคุมการทำงานของจอ LCD (I2C) เช่นเดียวกันกับจอ LCD แบบ คีอรหัสคำสั่ง ที่ใช้ในการควบคุมนั้นเหมือนกัน แต่ต่างกันตรงที่รูปแบบในการรับส่งข้อมูล

- GND เป็น Ground ใช้ต่อระหว่าง Ground ของระบบ Microcontroller กับ LCD
- VCC เป็นไฟเลี้ยงวงจรที่ป้อนให้กับ LCD มีขนาด +5VDC
- SDA (Serial Data) เป็นขาที่ใช้ในการรับส่งข้อมูล
- SCL (Serial Clock) เป็นขาสัญญาณนาฬิกาในการรับส่งข้อมูล

### 2.5.3 การเชื่อมต่อระหว่าง Arduino กับจอ LCD (I2C)

รูปที่ 2.16 แสดงการเชื่อมต่อระหว่าง Arduino Uno และหน้าจอ LCD เพื่อให้ LCD สามารถแสดงผลตามคำสั่ง



รูปที่ 2.16 การเชื่อมต่อระหว่าง Arduino Uno และ LCD

ตารางที่ 2.10 การเชื่อมต่อระหว่าง Arduino Uno กับ LCD (I2C)

Arduino UNO R3	LCD (I2C)
GND	GND ( Pin 1)
+5VDC	VCC ( Pin 2 )
A4 (SDA)	SDA ( Pin 3 Serial Data )
A5 (SCL)	SCL ( Pin 4 Serial Clock )

## 2.6 Push button switch

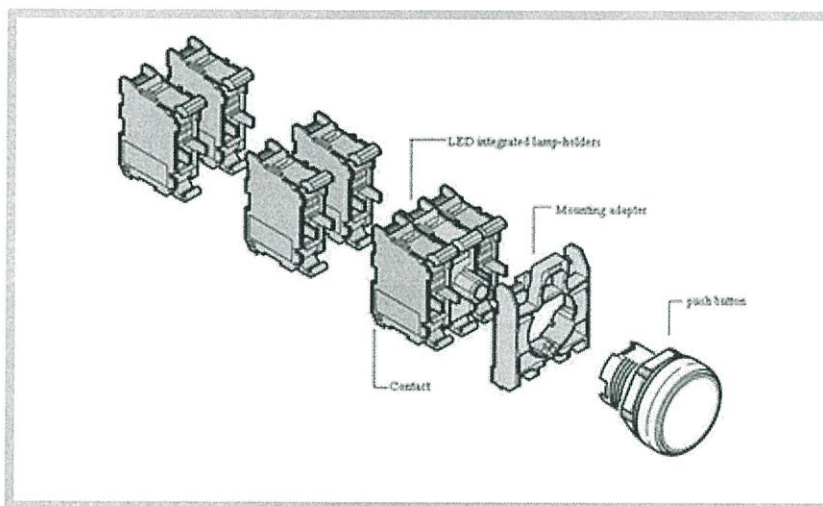
Push Button Switch หรือ สวิตช์ปุ่มกด เป็นอุปกรณ์ทางไฟฟ้าซึ่งทำหน้าที่ตัดและต่อวงจรทางไฟฟ้าและใช้ในการควบคุมการทำงานของมอเตอร์ หรือการทำงานของเครื่องจักรต่างๆ เป็นเหมือนอุปกรณ์พื้นฐาน ใช้ได้กับอุตสาหกรรมทั่วไป มีทั้งแบบมีไฟ และทึบแสง ดังแสดงในรูปที่ 2.17 แสดงตัวอย่าง push button switch



รูปที่ 2.17 Push button switch

โครงสร้างของสวิตช์ปุ่มกดสามารถแยกได้ 4 ส่วน ดังแสดงรูปที่ 2.18 ซึ่งประกอบไปด้วย

- ปุ่มกดทำด้วยโลหะหรือพลาสติกซึ่งจะมีหลายหลายสีให้เลือกใช้งาน
- ฐานยึดระหว่างปุ่มกดและตัวล็อกหน้าสัมผัส โดนจะมีเกลียวที่ฐานเพื่อไว้ สำหรับยึดอุปกรณ์กับชิ้นงานด้วย
- หน้าสัมผัส NO และ NC
- หลอดไฟ LED ที่ใช้แสดงสถานะ



รูปที่ 2.18 โครงสร้างของ สวิตช์

### 2.6.1 หลักการทำงานของ Push Button Switch

Push button Switch สามารถแบ่งได้ 2 ประเภทคือ

- แบบกดติดปล่อยดับ หรือ ที่เรียกว่าแบบสปริงรีเทิร์น เป็นประเภทที่นิยมใช้กันมากที่สุด เนื่องจากมีฟังก์ชันการทำงานที่ไม่ยุ่งยาก นิยมใช้กับ ตู้ MDB ตู้ DB และ ตู้ control เครื่องจักร เป็นต้น

- แบบกดติดกดดับ หรือ แบบ push on / push off เป็นประเภทที่นิยมใช้ในฟังก์ชันการทำงานแบบไม่ซับซ้อน เช่น Start/stop สายพานลำเลียง เป็นต้น

Push button Switch มีทั้งหมด 2 ชนิด ดังนั้นหลักการทำงานจึงมีความแตกต่างกัน

- แบบกดติดปล่อยดับ

เมื่อมีการกด Push button Switch หน้าสัมผัสดังกล่าวจะเปลี่ยนสถานะ

จาก NO เป็น NC หรือ จาก NC จะเป็น NO แต่เมื่อปล่อยมือออกจาก Push button Switch หน้าสัมผัสจะกลับสู่สภาวะปกติในตำแหน่งเดิมโดยมีแรงผลักดันจากสปริงให้ Push button Switch เข้าสู่สภาวะปกติ

- แบบกดติดกดดับ

เมื่อมีการกด Push button Switch หน้าสัมผัสดังกล่าวจะเปลี่ยนสถานะ จาก No

เป็น NC หรือจาก NC จะเป็น NO แต่เมื่อปล่อยมือออกจาก Push button Switch หน้าสัมผัสจะถูกล็อกไว้โดยกลไกของ สวิตช์ ซึ่งสามารถกลับสู่สภาวะปกติในตำแหน่งเดิมได้โดยกด Push button Switch อีกครั้งทำให้คลายล็อก จะมีแรงผลักดันจากสปริงให้ Push button Switch เข้าสู่สภาวะปกติ

## 2.6.2 การนำ switch ไปใช้งาน

### 2.6.2.1 Pull-up, Pull-down

เมื่อนำ สวิตช์หรือปุ่ม button ต่อเข้ากับ Input Pin ของ Arduino นั้น จำเป็นมากที่ต้องมี Pull-up Resistor หรือ Pull-down Resistor เพราะ Input Pin นั้นๆ จะอ่านค่าได้ไม่แน่นอนว่าค่าที่รับมาเป็น HIGH หรือ LOW เนื่องจากหลายๆครั้งที่แรงดันจะอยู่ครึ่งๆกลางๆระหว่าง HIGH กับ LOW จึงต้องมีการใส่ตัวต้านทาน Pull-up หรือ Down ลงไป เพื่อกำหนดสถานะให้ชัดเจน

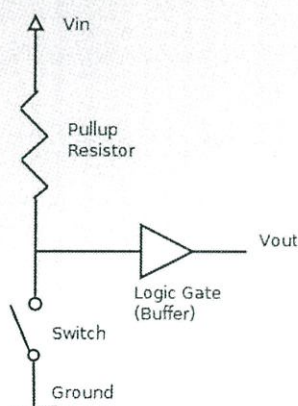
คำว่า Pull จึงหมายถึงการดึงหรือการบังคับให้เป็นสูงหรือต่ำ

- Pull-Up คือการบังคับให้ Input มีสถานะ HIGH ตลอดเวลา จนกว่าจะกดปุ่ม เมื่อกดปุ่ม จะเป็น LOW เมื่อไม่กดจะเป็น HIGH หรือเรียกว่า Active Low

- Pull-Down คือบังคับให้ Input เป็น LOW ตลอดเวลา จนกว่าจะกดปุ่ม เมื่อกดปุ่มจะเป็น HIGH เมื่อไม่กดจะเป็น LOW หรือเรียกว่า Active High

- วงจร Pull-Up

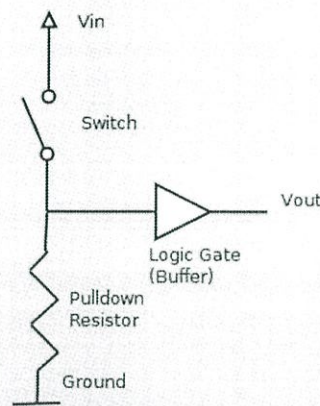
รูปที่ 2.19 แสดงการต่อวงจร pull-up โดยต่อตัวต้านทานค่าหนึ่ง เรียกว่า Pull-UP Resistor ไว้ที่ไฟเลี้ยง  $V_{in}$  เพื่อเป็นการรักษาระดับแรงดันไฟฟ้าให้เป็น HIGH ตลอด ,Arduino Input Pin จึงอ่านค่าได้เป็น HIGH ตลอด แต่เมื่อกดปุ่ม Button ,กระแสไฟฟ้าจะไหลลง Ground ทันที Input Pin จึงอ่านค่าได้เป็น LOW



รูปที่ 2.19 วงจร pull-up

- วงจร Pull-Down

รูปที่ 2.120 แสดงการต่อวงจร pull-down โดยต่อตัวต้านทาน หรือ Pull-Down Resistor ไว้ที่ Ground แทน ส่วนปุ่ม Button จะต่อเข้ากับ Vin เมื่อไม่กดจะมีสถานะเป็น LOW แต่เมื่อกดจะเป็น HIGH เพราะกระแสไฟฟ้าจะเลือกไหลเข้าหา Arduino Pin ที่มีความต้านทานน้อยกว่าแทน



รูปที่ 2.20 วงจร pull-down

## 2.7 แหล่งจ่ายไฟ

แหล่งจ่ายไฟ เป็นอุปกรณ์ที่จ่ายพลังงานไฟฟ้าให้กับโหลดไฟฟ้า ใช้ในการแปลงพลังงานไฟฟ้าจากรูปแบบหนึ่ง ไปเป็นอีกรูปแบบหนึ่ง

แหล่งจ่ายไฟทุกตัวต้องได้รับพลังงานจากแหล่งพลังงานภายนอกเพื่อจ่ายให้โหลดและการบริโภคพลังงานของตัวเองในขณะที่ปฏิบัติงาน แหล่งพลังงานภายนอกจะขึ้นอยู่กับการออกแบบ แหล่งจ่ายไฟอาจจะได้รับพลังงานจาก

- ระบบสายส่งพลังงานไฟฟ้า อาจเป็นกระแสสลับ หรือกระแสตรงที่ได้จากอุปกรณ์ไฟฟ้าที่แปลงแรงดันไฟฟ้า กระแสสลับให้เป็นกระแสตรง
  - อุปกรณ์จัดเก็บพลังงาน เช่น แบตเตอรี่ และ เซลล์เชื้อเพลิง
  - ระบบเครื่องกลไฟฟ้า เช่น เครื่องกำเนิดไฟฟ้าจากพลังงานกล เช่น Generator และ Alternator
  - พลังงานแสงอาทิตย์
  - แหล่งจ่ายไฟอาจถูกนำมาใช้แบบแยกส่วน หรือเป็นอุปกรณ์ส่วนหนึ่งของโหลด เช่น แหล่งจ่ายไฟในคอมพิวเตอร์เดสก์ท็อป และอุปกรณ์อิเล็กทรอนิกส์ทั่วไปสำหรับผู้บริโภค
- คุณลักษณะเฉพาะที่ระบุไว้บนแหล่งจ่ายไฟ ได้แก่
- ปริมาณของแรงดันและกระแสที่จะสามารถจ่ายให้กับโหลดได้

- วิธีการที่จะทำให้แรงดันหรือกระแสเอาต์พุตมีเสถียรภาพ ภายใต้เงื่อนไขที่กระแสไฟฟ้าอินพุตและสถานะของโหลดที่เปลี่ยนแปลง

- ระยะเวลาการใช้งานได้นานเท่าใดโดยไม่ต้องเติมเชื้อเพลิงหรือหรือชาร์จประจุใหม่ (เฉพาะ อุปกรณ์ไฟฟ้าที่ใช้แหล่งพลังงานแบบพกพา)

### 2.7.1 ประเภทของแหล่งจ่ายไฟ

แหล่งจ่ายไฟสำหรับอุปกรณ์อิเล็กทรอนิกส์สามารถแบ่งออกกว้างๆเป็นแบบความถี่ไฟฟ้าขาเข้าและแบบสวิตซ์ซิง แบบความถี่มักจะมีการออกแบบที่ค่อนข้างง่าย แต่จะมีขนาดใหญ่และหนักสำหรับแหล่งจ่ายไฟกระแสสูง เนื่องจากความจำเป็นที่จะต้องใช้หม้อแปลงไฟฟ้าขาเข้าที่มีขนาดใหญ่ และวงจรแบบอิเล็กทรอนิกส์ความถี่สูงที่ต้องการตัวกระจายความร้อนขนาดใหญ่ แหล่งจ่ายไฟแบบธรรมดา นี้ บางครั้งเรียกว่าแบบ "เชิงเส้น" แต่ที่เรียกชื่อผิด เพราะการแปลงจากแรงดันไฟฟ้า AC เป็น DC โดยเนื้อแท้แล้ว ไม่ใช่เชิงเส้น เมื่อ วงจรเรียงกระแสจ่ายไฟเข้าไปใน capacitor. Linear voltage regulators จะผลิตแรงดันไฟฟ้า เอาต์พุตที่จะถูก regulated โดยวิธี voltage divider ที่บริโภคพลังงาน จึงทำให้มีประสิทธิภาพต่ำ แหล่งจ่ายไฟแบบ switched-mode ที่ให้เอาต์พุตเท่ากับแหล่งจ่ายไฟธรรมดาจะมีขนาดเล็กกว่า มีประสิทธิภาพสูงกว่า แต่จะซับซ้อนมากขึ้น

#### • แหล่งจ่ายไฟ DC

แหล่งจ่ายไฟ DCที่ไม่ควบคุมปกติจะใช้หม้อแปลงไฟฟ้าเพื่อแปลงแรงดันจากผนัง (ไฟบ้าน) ให้ต่ำลงให้ได้แรงดันที่ต้องการ ถ้าต้องการผลิต แรงดัน DC, วงจรเรียงกระแส จะใช้ในการแปลงแรงดันไฟฟ้าสลับให้เป็นแรงดันไฟฟ้าตรง(ยังเป็นรูปคลื่นอยู่) ตามด้วยตัวกรองประกอบด้วยตัวเก็บประจุ ตัวต้านทาน อย่างน้อยหนึ่งตัว และบางครั้งมี ตัวเหนี่ยวนำด้วยเพื่อ ทำการกรอง (ทำให้เรียบ) ของคลื่นเหล่านั้น คลื่นขนาดเล็กที่เหลือจากการกรองหรือที่เรียกว่า ripple นี้เป็นสิ่งไม่พึงประสงค์ ซึ่งอาจมีความถี่น้อยกว่าความถี่จาก AC input (ขึ้นอยู่กับวิธีเรียงกระแสว่าเป็นแบบครึ่งคลื่น หรือ เต็มคลื่น) ripple นี้จะขึ้นบนแรงดันไฟฟ้าตรง อย่างหลีกเลี่ยงไม่ได้ ในวัตถุประสงค์ เช่นการชาร์จแบตเตอรี่, ripple ไม่ได้เป็นปัญหาและแหล่งจ่ายไฟที่ง่ายที่สุด อาจเป็นเพียงหม้อแปลงแรงต่ำที่มีไดโอดตัวเดียวกับความต้านทานอีกหนึ่งตัวต่ออนุกรมอยู่ก็ได้ ก่อนที่จะมีการนำอุปกรณ์แบบ solid-state มาใช้, อุปกรณ์อิเล็กทรอนิกส์จะใช้วาล์ว (หลอด สูญญากาศ) ซึ่งต้องการแรงดันสูง แหล่งจ่ายไฟจะใช้หม้อแปลงแบบ step-up, วงจรเรียงกระแส และ ตัวกรอง เพื่อสร้างแรงดันไฟฟ้าตรงหลายระดับ หลายร้อยโวลต์ และแรงดันกระแสสลับเพื่อจุดใส่หลอด เฉพาะอุปกรณ์ที่ทันสมัยที่สุดเท่านั้นที่ต้องใช้แหล่งจ่ายไฟควบคุมที่มีขนาดใหญ่และราคาแพง

#### • แหล่งจ่ายไฟ AC

ปกติจะใช้แรงดันไฟฟ้าจากเต้าเสียบ (ไฟบ้าน) และ ลดแรงดันลงในระดับ แรงดันไฟฟ้าที่ต้องการ บางครั้ง การกรองก็ต้องการเช่นกัน แหล่งจ่ายไฟ AC สามารถผลิตไฟฟ้า AC ได้จากไฟ DC วงจรที่ใช้เปลี่ยนไฟ DC ให้เป็นไฟ AC เรียกว่า Inverter อินเวอร์เตอร์ทำงานด้วยอุปกรณ์ power switching ที่ทำจากสารกึ่งตัวนำที่ควบคุมได้. รูปคลื่น AC ที่เอาต์พุตจึงมีค่าที่ไม่เกาะติดกัน เนื่องจากการแปลงทำงานอย่างรวดเร็วแทนที่จะเป็นการแปลงอย่างราบรื่น ความสามารถในการสร้างรูปคลื่นใกล้เคียงไซน์ที่มีความถี่ใกล้เคียงความถี่พื้นฐานถูกบังคับโดยใช้เทคนิคการมอดดูเลชันที่มีการควบคุมตลอดเวลาที่ เพาเวอร์วาล์วเปิดและปิด เทคนิคการมอดดูเลชันที่ใช้ได้แก่ carrier-based technique, หรือ pulse width modulation, space-vector technique, และ selective-harmonic technique.

## 2.7.2 การประยุกต์ใช้แหล่งจ่ายไฟ

- แหล่งจ่ายไฟคอมพิวเตอร์

แหล่งจ่ายไฟคอมพิวเตอร์ที่ทันสมัยจะเป็นแบบ switch-mode ที่แปลงไฟ AC จากแหล่งจ่ายไฟหลักให้เป็นแรงดันไฟฟ้ากระแสตรงหลายแรงดัน แหล่งจ่ายแบบสวิตซ์ชโหมดมาแทนที่แหล่งจ่ายแบบเชิงเส้น เนื่องจากค่าใช้จ่าย น้ำหนักและ ขนาดที่ต่ำกว่า ความหลากหลายของแรงดันไฟฟ้าที่ส่งออก ยังมีความต้องการของการใช้กระแสที่แตกต่างกันอย่างกว้างขวาง

- แหล่งจ่ายไฟสำหรับการเชื่อม

การเชื่อมอาร์คใช้ไฟฟ้าเพื่อละลายพื้นผิวของโลหะให้หลอมรวมเป็นเนื้อเดียวกัน ไฟฟ้าที่ใช้จะได้จากแหล่งจ่ายไฟสำหรับการเชื่อม และสามารถเป็นได้ทั้งไฟ AC หรือ DC การเชื่อมอาร์คมักจะต้องใช้กระแสสูง ปกติระหว่าง 100 ถึง 350 แอมป์ การเชื่อมบางชนิดสามารถใช้เพียง 10 แอมป์ ในขณะที่การใช้งานแบบ spot welding กระแสที่ใช้อาจสูงที่สุดถึง 60,000 แอมป์ในช่วงเวลาสั้นมาก แหล่งจ่ายไฟสำหรับการเชื่อมแบบเก่าประกอบด้วย หม้อแปลงหรือเครื่องยนต์ที่ขับเคลื่อนด้วยกำเนิดไฟฟ้า ล่าสุด ใช้เซมิคอนดักเตอร์และไมโครโปรเซสเซอร์เพื่อลดขนาดและน้ำหนักของพวกมัน

- AC adapter

แหล่งจ่ายไฟที่สร้างขึ้นภายในเต้าเสียบ AC ตัวผู้ เป็นที่รู้จักกันว่าเป็น "แพ็คเสียบ" หรือ "อะแดปเตอร์ปลั๊กอิน" AC adapter ที่ราคาถูกสุดประกอบด้วยหม้อแปลงขนาดเล็กตัวเดียวเท่านั้น ในขณะที่ อะแดปเตอร์ DC จะเพิ่มไดโอดเข้าไปด้วย ไม่ว่าจะมีความถี่หรือไม้อะแดปเตอร์จะมีสนามแม่เหล็กอย่างต่อเนื่อง และโดยปกติจะไม่สามารถปิดการทำงานได้จนกว่าจะดึงออกจากเต้าเสียบ

## 2.7.3 ชนิดของแหล่งจ่ายไฟ

แหล่งจ่ายไฟมีหลายชนิด ส่วนใหญ่ออกแบบเพื่อแปลงไฟบ้านซึ่งมีแรงดันสูง (เอ.ซี. 220 โวลต์) ให้ได้แรงดันต่ำที่เหมาะสมใช้กับวงจรอิเล็กทรอนิกส์หรืออุปกรณ์อื่นๆ แหล่งจ่ายไฟสามารถแบ่งเขียนเป็นบล็อกอนุกรม ซึ่งแต่ละบล็อกมีหน้าที่เฉพาะต่างกัน

$$\text{อัตราส่วนรอบ} = \frac{V_p}{V_n} = \frac{N_p}{N_s} \quad \text{และ} \quad \text{กำลังออก} = \text{กำลังเข้า}$$

$$V_s \times I_s = V_p \times I_p$$

$V_p$  = แรงดันปฐมภูมิ(อินพุท)

$N_p$  = จำนวนรอบของขดลวดปฐมภูมิ

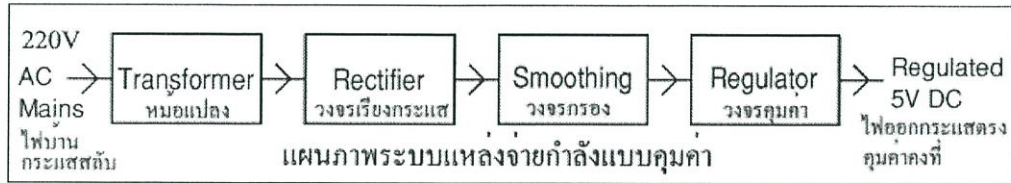
$V_s$  = แรงดันทุติยภูมิ (เอาต์พุท)

$N_s$  = จำนวนรอบของขดลวดทุติยภูมิ

$I_p$  = กระแสปฐมภูมิ(อินพุท)

$I_s$  = กระแสทุติยภูมิ (เอาต์พุท)

## ตัวอย่างแหล่งจ่ายไฟตรงคุมค่า 5V



แต่ละบล็อกมีคำอธิบายรายละเอียดดังนี้

• หม้อแปลง - ลดแรงดันสูงจากไฟกระแสสลับ (AC) บ้านให้มีแรงดันกระแสสลับ(AC)ต่ำ

• วงจรเรียงกระแส - แปลงไฟกระแสสลับ (AC) ต่ำ เป็นไฟกระแสตรง (DC) แต่ไฟตรงยังไม่เรียบ

• วงจรกรอง - กรองไฟตรง DC ให้เรียบมีพริ้ว (ripple) น้อยๆ

• วงจรคุมค่า - ทำให้พริ้วหายไปและกำหนดค่าแรงดัน DC ออกคงที่

• หม้อแปลง(Transformer) หม้อแปลงทำหน้าที่แปลงไฟฟ้ากระแสสลับจากแรงดันค่าหนึ่งเป็นอีกค่าหนึ่งโดยให้มีการสูญเสียกำลังงานน้อยที่สุด หม้อแปลงทำงานเฉพาะกับไฟฟ้ากระแสสลับเท่านั้น และนั่นก็เป็นเหตุผลหนึ่งที่ว่าทำไมไฟฟ้าบ้านจึงเป็นไฟกระแสสลับ หม้อแปลง แปลงขึ้น(step-up) เพิ่มแรงดัน ส่วนหม้อแปลง แปลงลง(step-down) ลดแรงดัน แหล่งจ่ายไฟส่วนใหญ่ใช้หม้อแปลงลดแรงดัน เพื่อลดแรงดัน ไฟบ้านที่มีแรงดันสูง (220V) ซึ่งเป็นอันตรายให้ต่ำลงเพื่อความปลอดภัย

ขดลวดทางเข้าเรียกว่าปฐมภูมิ(primary) และขดลวดทางออกเรียกว่าทุติยภูมิ(secondary) ระหว่างขดทั้งสองไม่มีการต่อกันทางไฟฟ้า แต่ใช้การเชื่อมกัน โดยสนามแม่เหล็กไฟฟ้ากระแสสลับที่เกิดขึ้นในแกนเหล็กของหม้อแปลง ขีดสองเส้นระหว่างขดลวดในรูปสัญลักษณ์แทนแกนเหล็ก

หม้อแปลงเสียพลังงานน้อย จึงถือว่ากำลังงานเข้าเท่ากับกำลังงานออก และสังเกตว่าเมื่อแรงดันแปลงลง กระแสก็จะแปลงขึ้น

อัตราส่วนจำนวนรอบของแต่ละขดลวดเรียกว่า อัตราส่วนรอบ(turns ratio) เป็นตัวกำหนดอัตราส่วนแรงดัน หม้อแปลงลดแรงดัน(step-down) มีขดลวด จำนวนรอบมากคือขดปฐมภูมิต่อกับแรงดันไฟบ้านเป็นอินพุท และทางด้านเอาต์พุทเป็นขดทุติยภูมิมีจำนวนรอบน้อยให้แรงดันออกต่ำ

- วงจรเรียงกระแส(Rectifier)

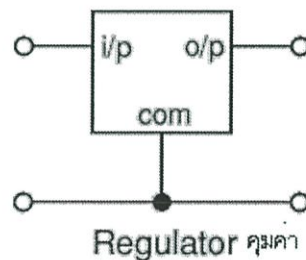
มีหลายวิธีในการต่อไดโอดของวงจรเรียงกระแสเพื่อแปลงไฟกระแสสลับเป็นไฟกระแสตรง วงจรเรียงกระแสแบบบริดจ์ เป็นวิธีที่สำคัญที่สุดสามารถเรียงกระแสแบบเต็มคลื่น(full-wave) และการเรียงกระแสแบบเต็มคลื่นมีอีกวิธีหนึ่งคือใช้ไดโอดเพียงสองตัวแต่ต้องต่อกับหม้อแปลงแบบเซนเตอร์แทป แต่ปัจจุบันไม่นิยมใช้วิธีนี้เพราะไดโอดไม่ได้แพงอะไรมาก ไดโอดตัวเดียวก็สามารถ ต่อเป็นวงจรเรียงกระแสได้ โดยเรียงได้เฉพาะคลื่นไฟฟ้ากระแสสลับทางด้านบวกได้เป็นไฟกระแสตรงครึ่งคลื่น

- วงจรกรอง(Smoothing)

การกรองเกิดขึ้นโดยการต่อ อิเล็กโทรไลติก คาปาซิเตอร์ ค่าสูงคร่อมไฟกระแสตรง ทำหน้าที่เหมือนบ่อเก็บน้ำ, ป้อนกระแสให้อาห์พุทเมื่อแรงดันกระแสสลับจากวงจรกรองกระแสตกลง

- วงจรคุมค่า(Regulator)

ไอซีคุมค่าแรงดันมีชนิดค่าแรงดันคงที่ (เป็นต้นว่า 5, 12 และ 15V) หรือ แรงดันเอาต์พุทปรับได้ มันถูกเรียกตามกระแสสูงสุดที่สามารถผ่านได้ ไอซีคุมค่าแรงดันลบก็มี เหมาะสำหรับใช้กับแหล่งจ่ายไฟแบบคู่ ไอซีคุมค่าส่วนใหญ่จะมีวงจรการป้องกันอัตโนมัติจาก กระแสเกิน (overload protection) และ ความร้อนเกิน (thermal protection) ไอซีคุมค่าแบบคงที่ส่วนมากมี 3 ขา และมองดูเหมือนเพาเวอร์ทรานซิสเตอร์ ดังแสดงในรูปที่ 2.21 เช่นไอซีคุมค่าเบอร์ 7805 +5V 1A ด้านบนมีรูสำหรับยึดติด แผ่นระบายความร้อนหากจำเป็น



รูปที่ 2.21 ตัวอย่าง Regulator คุมค่า

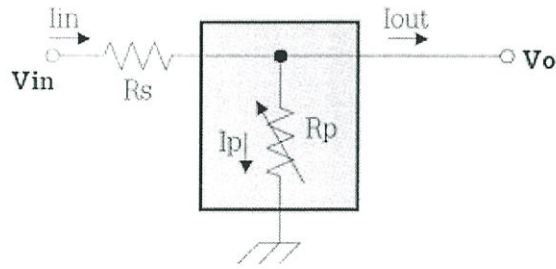
## 2.8 ไอซีเร็กกูเลเตอร์ (IC Regulator)

ประกอบด้วย 3 แบบ ได้แก่

### 2.8.1 เร็กกูเลเตอร์แบบขนาน (Shunt Regulator)

รูปที่ 2.22 แสดงการทำงานของวงจรเร็กกูเลเตอร์แบบขนาน โดยมีแรงดันอินพุท  $V_{IN}$  จ่ายให้กับวงจร มีตัวต้านทาน  $R_S$  ทำหน้าที่ในการจำกัดกระแสที่ไหลผ่านวงจรทั้งหมด ตัวต้านทานที่ปรับค่าได้  $R_p$  จะทำการปรับค่าเองโดยอัตโนมัติเพื่อให้แรงดันที่เอาต์พุทคงที่ตลอดสมการของแรงดันเอาต์พุท  $V_O = V_{IN} - R_S (I_O + I_p)$

### แผนผังการทำงานของเร็กกูเลเตอร์แบบขนาน



$$V_o = V_{IN} - R_s(I_o + I_p)$$

รูปที่ 2.22 แผนผังการทำงานของเร็กกูเลเตอร์แบบขนาน

ตัวอย่างของวงจรประเภทนี้ได้แก่วงจรเร็กกูเลเตอร์ที่ใช้ตัวต้านทานต่อกับซีเนอร์ไดโอด ซึ่ง  $R_p$  ในที่นี้ก็คือนอร์ไดโอดนั่นเอง

### 2.8.2 เร็กกูเลเตอร์แบบอนุกรม (Series Regulator)

หลักการการทำงานของเร็กกูเลเตอร์แบบอนุกรมนี้ แสดงในรูปที่ 2 โดยมีการจ่ายแรงดันที่ยังไม่ได้มีการเร็กกูเลทไปยัง  $R_p$  โดย  $R_p$  จะปรับค่าความต้านทานของตัวเองได้อัตโนมัติ ทำให้เกิดแรงดันตกคร่อมที่  $R_p$  ค่าหนึ่ง จะได้แรงดันเอาต์พุตเท่ากับ แรงดันอินพุตลบด้วยแรงดันตกคร่อมในตัวเร็กกูเลเตอร์ ซึ่งผลของการปรับค่า  $R_p$  ที่ถูกต้อง ก็จะทำให้ได้แรงดันเอาต์พุตตามที่ต้องการ และจากหลักการการทำงานของเร็กกูเลเตอร์ชนิดนี้เองที่ได้นำมาประยุกต์ทำเป็นไอซีเร็กกูเลเตอร์เบอร์ต่างๆ ทั้งเบอร์ 78XX เบอร์ 79XX และอื่นๆ อีก

### 2.8.3 แผนผังวงจรพื้นฐานของเร็กกูเลเตอร์แบบอนุกรม

แผนผังวงจรพื้นฐานของเร็กกูเลเตอร์ชนิดนี้ สามารถแบ่งออกได้ 3 ภาค ประกอบไปด้วย

- วงจรแรงดันอ้างอิง (Voltage Referent) ซึ่งเป็นส่วนที่เป็นอิสระต่อทั้งอุณหภูมิและแรงดันที่จ่ายให้กับเร็กกูเลเตอร์
- วงจรขยายความผิดพลาด (Error Amplifier) ทำหน้าที่คอยเปรียบเทียบแรงดันระหว่างแรงดันอ้างอิงและสัดส่วนของแรงดันเอาต์พุต ที่ป้อนกลับมาที่ขาอินเวอร์ตติ้งของออปแอมป์
- ซีรีส์พาสทรานซิสเตอร์ (Series Transistor) ทำหน้าที่จ่ายกระแสเอาต์พุตให้เพียงพอ กับความต้องการของโหลด

เมื่อป้อนแรงดันอินพุตให้กับไอซีเร็กกูเลเตอร์ แรงดันเอาต์พุตจะถูกป้อนมายังอินพุต โดย  $R_1$  และ  $R_2$  ทำหน้าที่เป็นวงจรแบ่งแรงดัน ซึ่งแรงดันที่ตกคร่อม  $R_2$  จะเป็นสัดส่วนกับแรงดันที่เอาต์พุต วงจรขยายความผิดพลาดจะทำหน้าที่รักษาสัดส่วนของแรงดันอ้างอิงกับแรงดันที่ตกคร่อม  $R_2$  ให้เท่ากัน

ถ้าแรงดัน  $V_{R2}$  มากกว่า  $V_{REF}$  วงจรขยายความผิดพลาดจะลดระดับการขยายสัญญาณเอาต์พุต ทำให้ทรานซิสเตอร์จ่ายกระแสลดลงเป็นผลให้แรงดันเอาต์พุตที่จ่ายให้โหลดลดลงด้วย

ถ้าแรงดัน  $V_{R2}$  น้อยกว่า  $V_{REF}$  วงจรขยายความผิดพลาดจะเพิ่มระดับการขยายสัญญาณเอาต์พุต ทำให้ทรานซิสเตอร์จ่ายกระแสเพิ่มขึ้น เป็นผลให้แรงดันเอาต์พุตที่จ่ายให้โหลดเพิ่มขึ้นด้วย

- ไอซีเร็กกูเลเตอร์สามขาชนิดจ่ายแรงดันคงที่

ไอซีเร็กกูเลเตอร์ภายในประกอบด้วยวงจรเร็กกูเลเตอร์แบบอนุกรม มีขาต่อใช้งาน 3 ขา ประกอบด้วยขา อินพุต เอาต์พุต และกราวด์ ซึ่งจะจ่ายแรงดันค่าใดค่าหนึ่งโดยเฉพาะ โดยรวมเอาส่วนของวงจรป้อนกลับที่ประกอบด้วย  $R_1$  และ  $R_2$  เข้าไว้เป็นส่วนหนึ่งของไอซี ซึ่งจุดนี้เองที่แตกต่างไปจากไอซีเร็กกูเลเตอร์ที่ปรับค่าได้(สุรเดช, 2554)

จุดเด่นของไอซีเร็กกูเลเตอร์ค่าคงที่นี้คือ สามารถต่อวงจรได้ง่ายไม่ต้องต่ออุปกรณ์ภายนอกเพิ่มเติมมากนัก ตัวอย่างวงจรการใช้งาน ดังแสดงในรูปที่ 4 ในการต่อวงจรบางครั้งจำเป็นต้องต่อไอซีเร็กกูเลเตอร์ห่างจากแหล่งจ่ายไฟอินพุตเกิน 5 เซนติเมตร จึงควรใส่ตัวเก็บประจุอิเล็กโทรไลต์ ขนาดประมาณ 10 ไมโครฟารัด สักตัวไว้ด้านอินพุต เพื่อป้องกันการเกิดออสซิลเลตที่ความถี่สูง ซึ่งจะทำให้วงจรขาดเสถียรภาพ เอาต์พุตที่ออกจากไอซีเร็กกูเลเตอร์ จะได้แรงดันเอาต์พุตที่เรียบพอสมควรอยู่แล้ว แต่อาจจะใส่ตัวเก็บประจุที่มีค่าประมาณ 100 ไมโครฟารัด เพื่อช่วยปรับปรุงแรงดันให้เรียบขึ้น ถึงแม้ว่าแรงดันไอซีเร็กกูเลเตอร์ชนิดนี้จะให้แรงดันเอาต์พุตคงที่ มีเบอร์ให้เลือกแรงดันเอาต์พุตได้คงที่หลายเบอร์เช่น 5 V, 5.2 V, 6V, 8V, 10V, 12V, 15V, 18V และ 24V กระแสเอาต์พุตตั้งแต่ 10 มิลลิแอมป์ถึง 3 แอมป์ และมีให้เลือกทั้งชนิดเร็กกูเลเตอร์ไฟบวกและเร็กกูเลเตอร์ไฟลบ

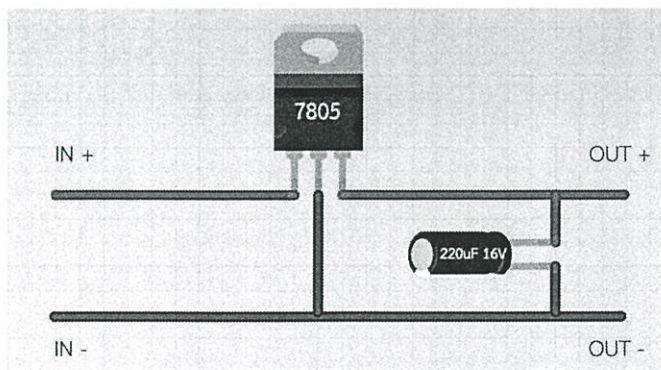
### 2.8.3.1 การแปลงไฟ 12V เป็น 5V

วิธีการแปลงโดยใช้ IC Linear Regulator

อุปกรณ์

- ไอซีเบอร์ 7805 พร้อมแผ่นระบายความร้อน 1 ตัว
- ตัวเก็บประจุขนาด 220 $\mu$ F 16V 1 ตัว
- สายไฟสีดำแดง

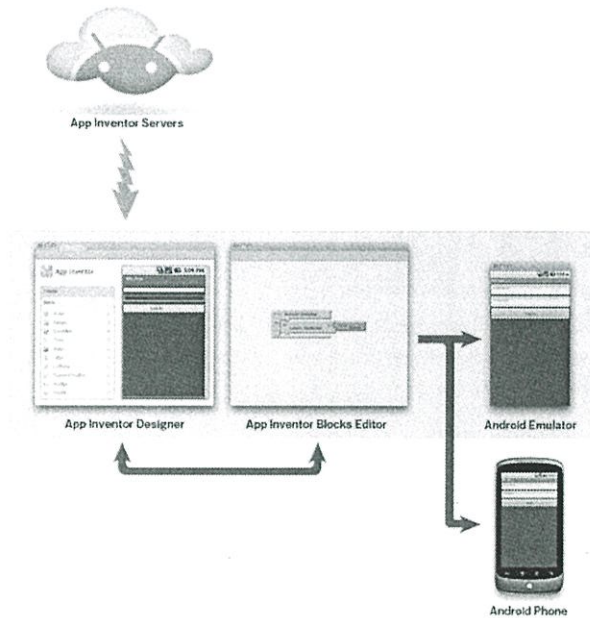
รูปที่ 2.23 แสดงการต่อวงจรแปลงไฟเพื่อแปลงไฟ 12V เป็น 5V โดยนำไอซี 7805 และตัวเก็บประจุมาต่อร่วมกัน



รูปที่ 2.23 วงจรแปลงไฟ

## 2.9 MIT App Inventor

โปรแกรม App Inventor พัฒนาขึ้นโดย MIT โดยโปรแกรม App Inventor เป็นเครื่องมือที่ใช้ในการเขียนโปรแกรมเพื่อพัฒนาเป็นแอปพลิเคชันสำหรับใช้งานบนระบบปฏิบัติการแอนดรอยด์ (Android) ซึ่งเป็นระบบปฏิบัติการที่ใช้ในอุปกรณ์สมาร์ทโฟนและแท็บเล็ต (วิทยา, 2556) โปรแกรม App Inventor พัฒนาขึ้นโดย Professor Hal Abelson และคณะซึ่งเคยเป็นผู้พัฒนาภาษาโลโก้มาก่อน เขาพัฒนาโปรแกรม App Inventor โดยได้รับการสนับสนุนจากบริษัทกูเกิล (Google Inc.) ซึ่งมีแนวคิดในการพัฒนาอยู่ที่ว่าคนที่อยากสร้างแอปพลิเคชันเพื่อใช้งานบนระบบปฏิบัติการแอนดรอยด์แต่ไม่มีความรู้ ไม่มีทักษะในการเขียนโปรแกรมมาก่อนก็สามารถสร้างแอปพลิเคชันขึ้นได้ง่ายๆ โดยโปรแกรม App Inventor พัฒนาขึ้นบนพื้นฐานทฤษฎีการเรียนรู้ตามแนวทางคอนสตรัคชันนิสซึม ซึ่งเน้นให้ใช้การเขียนโปรแกรมเป็นส่วนช่วยส่งเสริมให้เกิดประสิทธิภาพทางความคิดผ่านการเรียนรู้ด้วยการปฏิบัติงานและได้ถูกนำไปใช้สอนนักเรียนในระดับมัธยมศึกษาและในมหาวิทยาลัยหลายแห่งในสหรัฐอเมริกา (Wikipedia, 2012) ด้วยข้อดีของโปรแกรม App Inventor ที่ทำให้ผู้ใช้สามารถพัฒนาแอปพลิเคชันได้ง่ายและสนุกเหมือนการต่อจิ๊กซอว์ (Jigsaw puzzle) หรือการต่อดาวต่อเลโก้ (Lego bricks) App Inventor จึงได้รับความสนใจเป็นอย่างมากสำหรับนักการศึกษาและนักพัฒนาแอปพลิเคชันที่ยังไม่มีประสบการณ์ในการพัฒนาแอปพลิเคชันเพื่อใช้งานบนระบบปฏิบัติการแอนดรอยด์ สอดคล้องกับงานวิจัยของ Pavel Smutny (2011, p.358) ได้กล่าวถึงข้อดีของโปรแกรม App Inventor สำหรับใช้ในการเรียนการสอนว่าการเขียนโปรแกรมด้วย App Inventor ที่มีลักษณะเป็นการต่อบล็อกนั้นง่ายต่อการจำรูปแบบของคำสั่ง ต่างจากการเขียนโปรแกรมด้วยคำสั่งที่เป็นตัวอักษรซึ่งนักเรียนส่วนใหญ่มักจะพิมพ์คำสั่งผิดทำให้เกิดข้อผิดพลาดในการเขียนโปรแกรมจำนวนมาก และนักเรียนยังสับสนกับข้อความแสดงข้อผิดพลาด (error messages) ที่แสดงออกมา ด้วยกลุ่มของคอมโพเนนต์ และ ฟังก์ชันที่โปรแกรม App Inventor จัดเตรียมไว้ให้ ช่วยให้เมื่อเริ่มเขียนโปรแกรมผู้พัฒนาเพียงแค่หาบล็อกที่ต้องการแล้วคลิกลากบล็อกนั้นไปวางไว้ในโปรแกรม ซึ่งไม่จำเป็นต้องจำว่าจะเริ่มต้นเขียนด้วยคำสั่งอย่างไร โครงสร้างของคำสั่งเป็นแบบไหน ด้วยความเป็นบล็อกที่นำมาต่อกันเหมือนจิ๊กซอว์ช่วยลดข้อผิดพลาดในการเขียนโปรแกรมเพราะหากเป็นบล็อกที่เลือกไม่ถูกต้องก็จะไม่สามารถนำมาต่อเข้าด้วยกันได้ และด้วยคอมโพเนนต์ที่มีความสามารถสูง (High-level components) เช่นคอมโพเนนต์ที่เกี่ยวกับระบบ GPS (Global Positioning System) ระบบควบคุมหุ่นยนต์เลโก้ (Lego Mindstorms NXT Robot) ระบบอ่านข้อความเป็นเสียง (text-to-speech) และระบบรู้จำเสียง (speech recognition) เป็นต้น ช่วยลดเวลาในการพัฒนาแอปพลิเคชันเพราะผู้พัฒนาสามารถนำคอมโพเนนต์นั้นมาใช้งานได้เลยทันทีโดยไม่ต้องเสียเวลาเขียนโปรแกรมนานนับเดือน



รูปที่ 2.24 การเขียนโปรแกรมบนมือถือ Android ด้วย MIT App inventor ในภาพรวม

App Inventor servers เป็นเครื่องที่ให้บริการและเก็บงานโปรเจคต่างๆ ที่ผู้ใช้สร้างขึ้นมา ผู้ใช้พัฒนาโปรแกรมมือถือ Android โดยสร้างโปรเจคและเขียนโปรแกรมบนเว็บเบราว์เซอร์ ที่เชื่อมต่อไปยัง App Inventor servers เมื่อได้โปรแกรมมา ก็สามารถทดสอบกับโปรแกรมมือถือจำลอง (Android emulator) หรือโทรศัพท์มือถือ Android จริงๆ ก็ได้

เครื่องมือพัฒนาแอปพลิเคชันระบบปฏิบัติการแอนดรอยด์ที่ชื่อว่า App Inventor เป็นเครื่องมือตัวใหม่ที่ทำให้ใครก็สามารถพัฒนาแอปพลิเคชันบนมือถือแอนดรอยด์ (Android Phone) ได้ โดยไม่จำเป็นต้องเป็นนักพัฒนาโปรแกรมก็สามารถสร้างแอปพลิเคชันได้

App Inventor ช่วยให้ สามารถสร้างโปรแกรม Android โดยไม่ต้องเขียนโค้ดโปรแกรมหรือแทนการเขียนโค้ด และยังสามารถมองเห็นวิธีการออกแบบแอปพลิเคชันผ่านหน้าจอ กำหนดองค์ประกอบของแอปพลิเคชัน แล้วระบุลักษณะหรือเหตุการณ์การทำงานของแอปพลิเคชัน โดยการกำหนดค่าที่ต้องการลงในบล็อก เหมือนกันกับการเขียนโปรแกรมด้วยภาษาใดภาษาหนึ่ง เช่น การกำหนดตัวแปร การสร้างเงื่อนไข ดำเนินงานการทำงานซ้ำและมีเงื่อนไข หรือแม้แต่การคำนวณ มีบล็อกแม้กระทั่งการจัดเก็บข้อมูลในฐานข้อมูลและสร้างการพูดคุยกับบริการเว็บ เช่น Twitter หรือ Facebook ได้

App Inventor เป็นภาษาภาพที่มีอินเตอร์เฟซ แบบลากและวาง แม้กระทั่งคนที่ไม่มีความรู้ด้านการเขียนโปรแกรม หรือไม่รู้ภาษาคอมพิวเตอร์ใดเลย ก็พัฒนาแอปพลิเคชันของตนเองได้ ทั้งนี้เพราะ

- App Inventor มีความผิดพลาดในกรณี ไวยากรณ์ หรือ No syntax เพราะเป็นภาษาบล็อกที่ไม่จำเป็นต้องจำคำสั่งและพิมพ์รหัสลงไป

- ใน App Inventor ทุกสิ่งที่อยู่ตรงหน้าของคุณหรือบนหน้าจอคอมพิวเตอร์ ชิ้นส่วนและคุณสมบัติทางโปรแกรมต่างๆ มีการจัดระเบียบเป็นหมวดหมู่ สามารถใช้งานได้เพียงแค่ลากและวางเท่านั้น

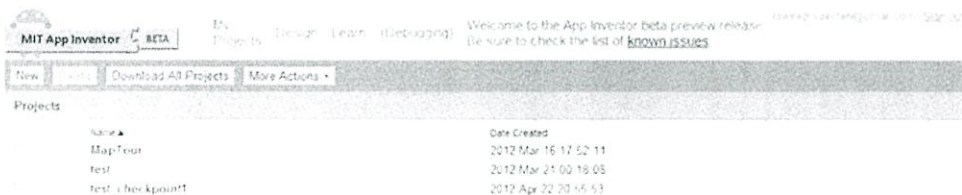
- App Inventor เป็นการออกแบบแอปพลิเคชันแบบระดับขั้น จากบนสู่ล่าง เกิดขึ้นตามลำดับที่ผู้ออกแบบได้กำหนดด้วยตนเอง

- App Inventor มีองค์ประกอบทางโปรแกรมระดับสูง แต่มีความเรียบง่ายในการใช้งาน การสร้างแอปพลิเคชันจะแบ่งการทำงานออกเป็นสองส่วน คือ ส่วนออกแบบ (App Inventor Designer) ที่จะทำให้เราเลือกคอมโพเนนต์ที่ต้องการสำหรับที่จะให้สร้างแอปพลิเคชัน ส่วนที่สองเป็นส่วนการเขียนโค้ด (App Inventor Blocks Editor) ที่ให้เราเขียนโค้ดด้วยการต่อบล็อกต่างๆ เข้าด้วยกันเป็นคำสั่ง ซึ่งจะเป็นการกำหนดพฤติกรรมหรือเหตุการณ์ที่เกิดขึ้นกับคอมโพเนนต์ การเขียนโปรแกรมจะเหมือนการต่อชิ้นส่วนตัวต่อจิ๊กซอว์เข้าด้วยกัน ในแต่ละขั้นตอนการสร้างจะสามารถทำการทดสอบได้ทุกขณะ และเมื่อสร้างเสร็จสมบูรณ์แล้วจะสามารถแพ็คเกจแอปพลิเคชันเพื่อนำไปใช้งานบนโทรศัพท์ระบบปฏิบัติการ Android เครื่องใดก็ได้ หรือหากไม่มีโทรศัพท์ระบบปฏิบัติการ Android ก็สามารถที่จะทดสอบได้บนโทรศัพท์จำลองที่ทำงานอยู่บนคอมพิวเตอร์ซึ่งจะมีลักษณะการทำงานเหมือนโทรศัพท์จริงทุกประการ สภาพแวดล้อมในการพัฒนาด้วย โปรแกรม App Inventor นั้น สนับสนุนระบบปฏิบัติการที่หลากหลาย ไม่ว่าจะเป็นระบบปฏิบัติการ Mac OS X, GNU / Linux และระบบปฏิบัติการ Windows และแอปพลิเคชันที่สร้างขึ้นนั้นสามารถติดตั้งและทำงานได้บนโทรศัพท์ระบบปฏิบัติการ Android หลากหลายรุ่นที่เป็นที่นิยมในปัจจุบัน

## 2.9.1 ส่วนประกอบของโปรแกรม App Inventor

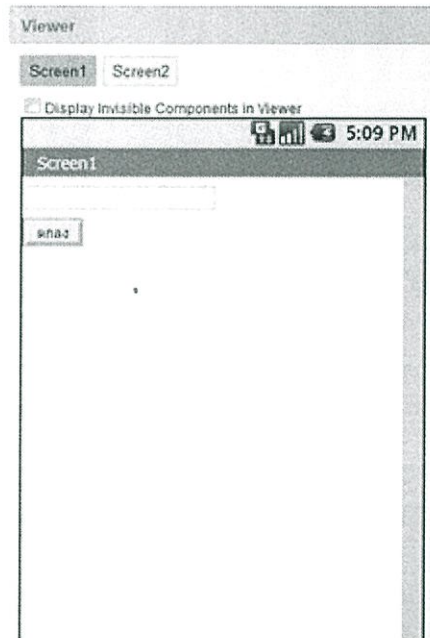
### 2.9.1.1 ส่วนออกแบบ (App Inventor Designer)

ในขั้นตอนแรกของการสร้างแอปพลิเคชันด้วย App Inventor เริ่มจากการเลือกคอมโพเนนต์ที่ต้องการและจัดวางลงในส่วนของการออกแบบ โดยจะทำผ่านส่วนของการออกแบบ (App Inventor Designer) ดังที่แสดงในรูปที่ 2.25 แสดงให้เห็นถึงแอปพลิเคชันที่สร้างขึ้นในหน้าต่างเว็บเบราว์เซอร์ โดยด้านซ้ายจะเป็นส่วนของคอมโพเนนต์ที่ App Inventor เตรียมไว้ให้จัดเรียงเป็นหมวดหมู่ เช่น ปุ่ม (button) ข้อความ (label) กล่องข้อความ (text box) เป็นต้น ผู้ใช้ทำการเพิ่มคอมโพเนนต์ที่เลือกด้วยการคลิกลากลงไปวางไว้ในโปรเจกต์



รูปที่ 2.25 หน้าจอการจัดการโปรเจกต์

อินเตอร์เฟซบนหน้าเว็บ App Inventor นั้นจะประกอบด้วยแท็บที่จะปรากฏในส่วนบนของหน้าเว็บซึ่งจะใช้ในการเข้าไปจัดการโปรเจค (My Projects) ส่วนการออกแบบ (Design) ส่วนการเรียนรู้คำสั่ง (Learn) ในหน้าจการจัดการโปรเจค ดังแสดงในรูปที่ 2.26 จะสามารถเข้าไปจัดการสร้าง ลบ ดาวน์โหลด หรือเลือกโปรเจคที่สร้างและได้ทำการบันทึกไว้เพื่อกลับมาแก้ไขในหน้าจอส่วนการออกแบบได้



รูปที่ 2.26 หน้าจอการออกแบบ (Viewer)

ในส่วนหน้าจอการออกแบบ ปุ่มที่อยู่ทางด้านบนจะใช้เพื่อการบันทึกโปรเจคในลักษณะต่างๆ การเพิ่มและลบหน้าจอ Screen ปุ่มสำหรับการเปิดส่วนการเขียนโค้ด (Open the Blocks Editor) และการจัดแพ็คเกจแอปพลิเคชันเพื่อนำไปใช้งานบนโทรศัพท์ระบบปฏิบัติการ Android ต่อไป ในการสร้างแอปพลิเคชันที่หน้าจอส่วนการออกแบบนี้ ผู้ใช้จะเลือกคอมโพเนนท์เพื่อนำมาวางลงในส่วน Viewer ที่อยู่ตรงกลางหน้าจอ หลังจากนั้นคอมโพเนนท์ที่เลือกนำมาวางจะปรากฏในส่วน Viewer ตามมุมมองของผู้ใช้ซึ่งสามารถเลือกจัดวางลงในตำแหน่งที่เหมาะสมได้ตามต้องการ และคอมโพเนนท์นั้นยังปรากฏในส่วนรายการคอนโพเนนท์ (Components) เรียงกันเป็นรายการเพื่อให้ดูง่ายและสามารถเลือกคอมโพเนนท์ที่ต้องการกำหนดคุณสมบัติจากรายการนี้แล้วกำหนดคุณสมบัติต่างๆ ที่หน้าจอส่วนคุณสมบัติ (Properties) ซึ่งจะเป็นคุณสมบัติเฉพาะของคอมโพเนนท์นั้นๆ

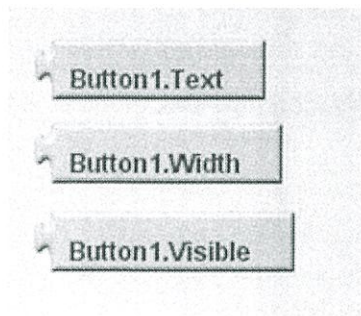
นอกจากในกลุ่มของคอมโพเนนท์ทั่วไปแล้วยังมีคอมโพเนนท์ที่มองไม่เห็น (Non-Visible Components) ซึ่งเมื่อนำมาวางในหน้าจอ Viewer แล้วจะไม่ปรากฏคอมโพเนนท์ดังกล่าวที่หน้าจอ Viewer แต่จะปรากฏที่หน้าจอรายการคอมโพเนนท์แทน คอมโพเนนท์ที่มองไม่เห็นนี้จะประกอบไปด้วยคอมโพเนนท์ในกลุ่ม Sensors ซึ่งประกอบไปด้วยคอมโพเนนท์ที่เกี่ยวข้องกับการเรียกใช้ตัวตรวจจับต่างๆ ที่มีอยู่ในโทรศัพท์ เช่น ระบบ GPS หรือ Accelerometers เป็นต้น กลุ่ม Notifiers ซึ่งเกี่ยวข้องกับความสามารถในการแจ้งเตือนต่างๆ หรือการเขียนบันทึกกิจกรรมของโทรศัพท์ ซึ่งคอมโพเนนท์ในกลุ่ม Notifiers นั้นจะมองไม่เห็นหรือถูกซ่อนไว้ แต่จะสามารถมองเห็นได้

เมื่อเกิดการแจ้งเตือนหรือสอบถามโดยมีการโต้ตอบกับผู้ใช้ในรูปแบบของข้อความ เสียง ปุ่ม หรือช่องสำหรับกรอกข้อมูล ที่จะแสดงให้เห็นเป็นครั้งคราวเท่านั้น กลุ่ม Clocks ซึ่งเกี่ยวข้องกับฟังก์ชันของเวลา ตัวจับเวลา และการตั้งค่าเวลา กลุ่ม ActivityStarters ซึ่งเกี่ยวข้องกับการสั่งให้แอปพลิเคชันอื่นที่ติดตั้งอยู่ในโทรศัพท์ทำงาน เช่น โปรแกรมอ่านบาร์โค้ด (barcode scanner) หรือโปรแกรมอ่านออกเสียงจากข้อความ (text to speech) เป็นต้น กลุ่ม Web Services เช่น คอมโพเนนต์เกี่ยวกับเกมออนไลน์ (Game Client) คอมโพเนนต์เกี่ยวกับฐานข้อมูลบนเว็บ และคอมโพเนนต์เกี่ยวกับการใช้บริการ Twitter เป็นต้น

### 2.9.1.2 ส่วนการเขียนโค้ด (App Inventor Blocks Editor)

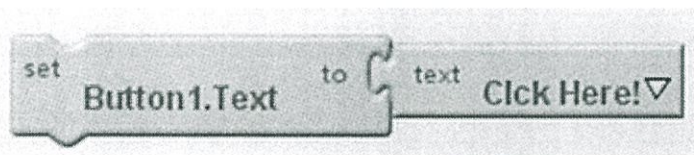
หลังจากที่ทำการเลือกจัดวางคอมโพเนนต์ที่จะใช้สำหรับโปรเจกต์เรียบร้อยแล้ว ผู้ใช้จะสามารถเขียนโค้ดคำสั่งสำหรับแอปพลิเคชันได้ในส่วนการเขียนโค้ด (App Inventor Blocks Editor) สำหรับพื้นที่การทำงานในส่วนหน้าจอการเขียนโค้ด จะประกอบไปด้วยคำสั่งที่อยู่ในรูปของบล็อกรวบรวมไว้บริเวณด้านซ้ายของหน้าจอ ผู้ใช้สามารถเลือกคำสั่งที่ต้องการโดยการคลิกลากบล็อกคำสั่งมาวางไว้ในโปรเจกต์คือบริเวณที่เป็นพื้นที่วางตรงกลางหน้าจอ บล็อกคำสั่ง จะเป็นคำสั่งพื้นฐานที่ผู้ใช้นำมาใช้ในการสร้างแอปพลิเคชันขึ้นมา บล็อกเหล่านี้จะถูกแยกและจัดแบ่งออกเป็นกลุ่มๆ ตามลักษณะของคำสั่ง ตัวอย่างเช่น บล็อกข้อความที่ใช้ในการทำงานที่เกี่ยวข้องกับข้อความที่เป็นสายอักขระ บล็อกทางคณิตศาสตร์ที่ใช้ในการทำงานที่เกี่ยวข้องกับฟังก์ชันทางคณิตศาสตร์ ตัวเลข หรือเครื่องหมายทางคณิตศาสตร์ เป็นต้น App Inventor ยังสามารถสร้างกระบวนการทำงาน (procedure) และตัวแปร (variable) ได้โดยการเลือกใช้บล็อกในส่วนที่เกี่ยวข้องกับการสร้างกระบวนการทำงานและเหตุการณ์ (event handler) ที่เกิดกับคอมโพเนนต์ โดยบล็อกที่เกี่ยวข้องกับคอมโพเนนต์ จะถูกจัดเตรียมไว้ให้ตามคอมโพเนนต์ที่ผู้ใช้เลือกนำมาวางไว้ในโปรเจกต์และจัดเก็บรวมกันไว้ในแท็บ My Blocks แยกไว้ต่างหาก บล็อกที่เกี่ยวข้องกับคอมโพเนนต์เหล่านี้จะแบ่งออกได้เป็น 4 แบบตามประเภทของคำสั่ง คือ ประเภทการเรียกค่าคุณสมบัติจากคอมโพเนนต์ (property getter) ประเภทการกำหนดค่าคุณสมบัติให้กับคอมโพเนนต์ (property setter) ประเภทเหตุการณ์ (event handler) และประเภทการเรียกใช้กระบวนการทำงาน (method call)

- การเรียกค่าคุณสมบัติจากคอมโพเนนต์ (property getter) บล็อกประเภทที่ใช้เรียกค่าคุณสมบัติจากคอมโพเนนต์จะมีลักษณะเป็นช่องต่ออยู่ทางด้านซ้าย โดยการทำงานจะทำการอ่านค่าคุณสมบัติจากคอมโพเนนต์แล้วส่งค่านั้นกลับมาในรูปแบบของข้อความ ตัวเลข หรือค่าทางตรรกศาสตร์ แต่ในบางคอมโพเนนต์อาจมีค่าที่มีรูปแบบที่ซับซ้อนมากกว่าเช่น ค่า GPS จากคอมโพเนนต์ตรวจจับตำแหน่ง (Location Sensor) เป็นต้น ซึ่งจะมีรูปแบบเฉพาะแตกต่างกันไป ตัวอย่างตัวแสดงในรูปที่ 2.27 แต่ทั้งนี้การอ่านค่านั้นทำได้ง่ายมาก ถึงแม้จะเป็นการอ่านค่า GPS ซึ่งโดยปกติมีกระบวนการทำงานที่ซับซ้อน แต่ผู้ใช้สามารถอ่านค่า GPS ได้ผ่านคอมโพเนนต์ตรวจจับตำแหน่งเหมือนอ่านค่าข้อความจากกล่องข้อความ ด้วยกระบวนการนี้ทำให้มั่นใจได้ว่าผู้ใช้จะไม่ต้องกังวลในเรื่องความซับซ้อนของการเข้าถึงค่าและข้อมูลต่างๆ ที่ต้องการ



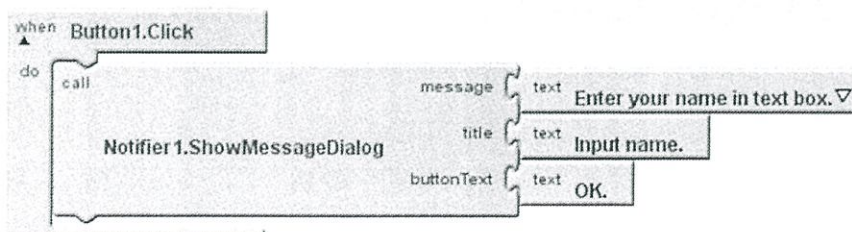
รูปที่ 2.27 ตัวอย่างของบล็อกคำสั่งประเภทที่ใช้เรียกค่าคุณสมบัติจากคอมโพเนนต์

- การกำหนดค่าคุณสมบัติให้กับคอมโพเนนต์ (property setter) บล็อกประเภทที่ใช้กำหนดค่าคุณสมบัติให้กับคอมโพเนนต์จะมีลักษณะเป็นช่องต่ออยู่ทางด้านขวา โดยจะสามารถทำการกำหนดค่าหรือเปลี่ยนแปลงค่าคุณสมบัติให้กับคอมโพเนนต์ที่ต้องการด้วยค่าของบล็อกที่นำมาต่อเข้ากับช่องต่อที่อยู่ทางด้านขวา ช่องต่อนี้จะมีรูปร่างเป็นช่องรับซึ่งจะต่อเข้าได้พอดีกับบล็อกที่มีรูปร่างเหมือนบล็อกประเภทที่ใช้เรียกค่าคุณสมบัติจากคอมโพเนนต์ ตัวอย่างดังแสดงในรูปที่ 2.28 ซึ่งจะทำให้ผู้ใช้เลือกบล็อกที่จะนำมาต่อเข้าด้วยกันได้อย่างง่ายได้และลดข้อผิดพลาดในการเลือกต่อบล็อกที่ไม่ถูกต้อง



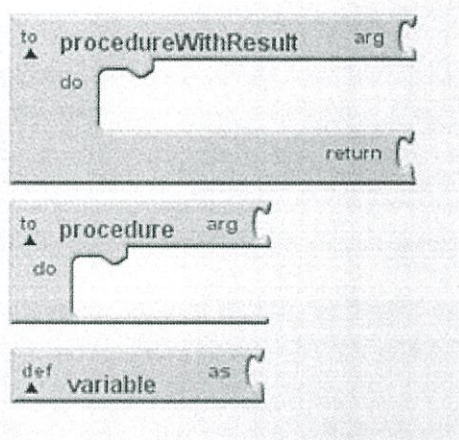
รูปที่ 2.28 ตัวอย่างของบล็อกคำสั่งประเภทที่ใช้กำหนดค่าคุณสมบัติให้กับคอมโพเนนต์

- เหตุการณ์ (event handler) บล็อกประเภทเหตุการณ์จะมีลักษณะเป็นช่องต่ออยู่ทางด้านล่างซึ่งบล็อกประเภtnี้นี้จะทำงานเมื่อเกิดเหตุการณ์ต่างๆ ขึ้นกับคอมโพเนนต์ เช่น การคลิกที่ปุ่ม ซึ่งจะทำงานตามบล็อกคำสั่งที่ต่อลงไปทางด้านล่างภายในบล็อกเหตุการณ์ ตัวอย่าง ดังแสดงในรูปที่ 2.29 แสดงให้เห็นถึงเหตุการณ์เมื่อมีการคลิกปุ่มแล้วให้มีการแสดงหน้าต่างข้อความโต้ตอบขึ้นมาเพื่อเตือนให้ผู้ใช้ทำการป้อนข้อมูลลงในกล่องข้อความ เป็นต้น



รูปที่ 2.29 ตัวอย่างของบล็อกคำสั่งประเภทเหตุการณ์ (event handler)

- การเรียกใช้กระบวนการทำงาน (method call) บล็อกประเภทเรียกใช้กระบวนการทำงานจะมีลักษณะเหมือนกับบล็อกประเภทที่ใช้เรียกค่าคุณสมบัติจากคอมโพเนนต์ที่มีลักษณะเป็นช่องต่ออยู่ทางด้านซ้าย ตัวอย่างดังแสดงในรูปที่ 2.30 บล็อกประเภทนี้จะถูกสร้างขึ้นเมื่อผู้ใช้ได้มีการสร้างกระบวนการทำงานโดยอาศัยบล็อกประเภทกระบวนการทำงานในการสร้างดัง เช่น การสร้างฟังก์ชันการทำงาน การสร้างตัวแปร เป็นต้น และจะมีชื่อเรียกเฉพาะตัวตามที่ผู้ใช้เป็นผู้ตั้งให้ เมื่อมีการสร้างกระบวนการทำงานขึ้น บล็อกประเภทเรียกใช้กระบวนการทำงานก็จะถูกสร้างขึ้นและปรากฏในตัวเลือกเพื่อให้ผู้ใช้เลือกที่มาวางลงในโปรเจกต์เมื่อต้องการให้เกิดการเรียนใช้กระบวนการทำงานดังกล่าว



รูปที่ 2.30 ตัวอย่างของบล็อกคำสั่งประเภทกระบวนการทำงาน

- ส่วนของการแพ็คเกจและการเรียกใช้งานแอปพลิเคชัน เมื่อแอปพลิเคชันได้ถูกออกแบบและทำการเขียนโค้ดคำสั่งเป็นที่เรียบร้อยแล้ว ผู้ใช้สามารถสั่งให้โปรแกรม App Inventor ทำการแพ็คเกจแอปพลิเคชันดังกล่าวให้อยู่รูปของไฟล์ที่พร้อมจะนำไปติดตั้งเพื่อนำไปติดตั้งบนโทรศัพท์ระบบปฏิบัติการ Android ต่อไป ผู้ใช้เพียงเลือกคลิกที่ปุ่ม Package for Phone ที่อยู่ในด้านบนของหน้าจอส่วนออกแบบ โปรแกรม App Inventor จะทำการแพ็คเกจบนเซิร์ฟเวอร์ App Inventor และส่งไฟล์ที่พร้อมจะนำไปติดตั้งออกมาให้ผู้ใช้ทำการดาวน์โหลดเก็บไว้ในเครื่องคอมพิวเตอร์เพื่อจะนำไปติดตั้งและเรียกใช้งานบนโทรศัพท์ระบบปฏิบัติการ Android เครื่องใดก็ได้ หรือหากไม่มีโทรศัพท์ระบบปฏิบัติการ Android ก็สามารถที่จะทดสอบการทำงานของแอปพลิเคชันได้บนโทรศัพท์จำลองที่ทำงานอยู่บนคอมพิวเตอร์ซึ่งจะมีลักษณะการทำงานเหมือนโทรศัพท์จริงทุกประการ(ทวีป, 2559)

## บทที่ 3

### วิธีการดำเนินงานวิจัย

#### 3.1 วิธีที่ใช้ศึกษาค้นคว้าและการวิจัยการทดลอง

ในการศึกษาค้นคว้าการวิจัยนั้น เริ่มต้นด้วยการศึกษาหาข้อมูลเกี่ยวกับผู้ป่วยที่มีภาวะติดเตียง มีอาการอย่างไร ต้องได้รับการดูแลเช่นไร มีผลกระทบต่อผู้ดูแลอย่างไร จากนั้นศึกษางานวิจัยที่เกี่ยวข้องกับผู้ป่วยติดเตียง มีการออกแบบอุปกรณ์เพื่อช่วยเหลือผู้ป่วยอย่างไร เพื่อศึกษาเป็นแนวทาง และนำมาใช้ในการออกแบบต้นแบบเตียงช่วยพลิกตัวผู้ป่วยให้สามารถทดลองใช้งานและสามารถลดภาระแก่ผู้ดูแลได้จริง

#### 3.2 ลักษณะข้อมูล การเลือกข้อมูล และการทดลอง

Arduino ทำงานร่วมกับ Stepper motor , Stepper motor drive โดยเขียนโค้ดภาษาซีเพื่อสั่งการให้เตียงช่วยพลิกตัวผู้ป่วยเคลื่อนที่ได้ร่วมกับอุปกรณ์ดังกล่าว และใช้ Arduino ร่วมกับ Bluetooth และโปรแกรม inventor เพื่อควบคุมผ่านแอปพลิเคชัน

ในการทดลองเพื่อทดสอบการทำงานของอุปกรณ์เพื่อให้เคลื่อนที่ตามที่กำหนดได้ ทำการออกแบบอุปกรณ์ เลือกวัสดุ ออกแบบส่วน hardware และ software ทำการทดสอบว่าอุปกรณ์สามารถเคลื่อนที่ได้หรือไม่ หากเคลื่อนที่ได้ดีแล้วเมื่อกดปุ่มหรือสั่งการในแอปพลิเคชันการเคลื่อนที่เป็นไปอย่างถูกต้องหรือไม่ เมื่อทุกส่วนทำงานสัมพันธ์กันอุปกรณ์จะสามารถเคลื่อนที่ได้ตามที่กำหนดไว้

#### 3.3 เครื่องมือและวิธีการทดลอง

ใช้อุปกรณ์อิเล็กทรอนิกส์ทั้งหมดและส่วน hardware ร่วมกับการเขียนโค้ดภาษาซี เพื่อควบคุมการเคลื่อนที่ของเตียงช่วยพลิกตัวผู้ป่วย โดยแบ่งการเคลื่อนที่ออกเป็น 2 ฝั่ง เมื่อกด สวิตซ์จะส่งค่าไปยัง Arduino จากนั้น Arduino จะประมวลผลและส่งค่าไปยังอุปกรณ์ต่างๆ ทำให้เตียงช่วยพลิกตัวผู้ป่วยสามารถเคลื่อนที่ได้

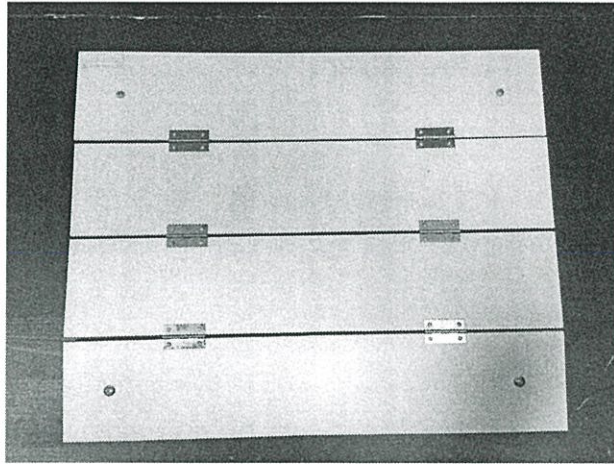
#### 3.4 ขั้นตอนออกแบบและสร้างเครื่องมือ

ในการออกแบบอุปกรณ์แบ่งเป็น 2 ส่วน ดังต่อไปนี้

##### 3.4.1 การออกแบบส่วน Hardware

3.4.1.1 ศึกษาการดูแลผู้ป่วยติดเตียงและงานวิจัยต่างๆ รวบรวมเพื่อนำมาใช้ในการออกแบบเตียงช่วยพลิกตัวผู้ป่วย

3.4.1.2 การออกแบบเตียงช่วยพลิกตัวผู้ป่วย ส่วนพื้นเตียงจะแบ่งออกเป็น 4 ส่วน แต่ละส่วนมีขนาดกว้าง 40 เซนติเมตร ยาว 60 เซนติเมตร วัสดุที่นำมาใช้ คือ plaswood ดังแสดงในรูปที่ 3.1 ซึ่งแสดงลักษณะตัวอย่างของไม้ plaswood ที่ถูกนำมาใช้ทำพื้นเตียง

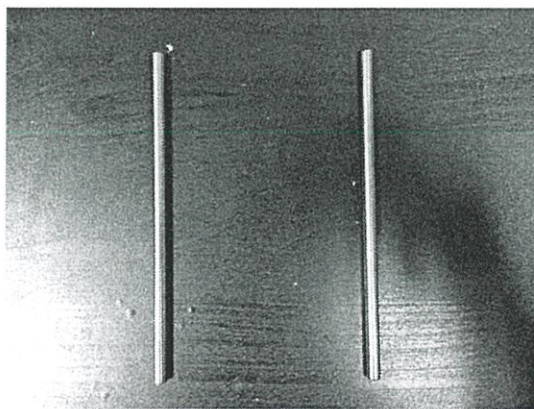


รูปที่ 3.1 พื้นเตียงช่วยรับน้ำหนักของผู้ป่วย

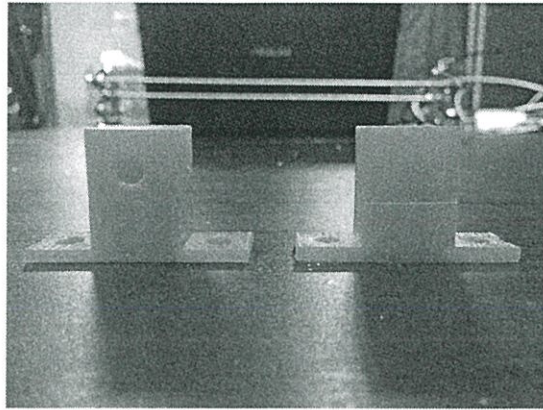
3.4.1.3 การเคลื่อนที่จะแบ่งการควบคุมออกเป็น 2 ฝั่ง คือซ้ายและขวา ใช้กลไกการเคลื่อนที่เดียวกับระบบ scissor lift ใช้ Stepper motor เป็นตัวควบคุม โดยจะทำงานร่วมกับ Motor driver board และ Arduino

3.4.1.4 การควบคุมการเคลื่อนที่ของต้นแบบเตียงช่วยพลิกตัวผู้ป่วย ประกอบไปด้วย

3.4.1.4.1 Linear Guide เพื่อช่วยให้การเคลื่อนที่เป็นได้อย่างพร้อมเพรียง วัสดุที่นำมาใช้ คือ rod shaft มีลักษณะเป็นแท่งเหล็กกลมเรียบ ขนาดเส้นผ่านศูนย์กลางยาว 6 มิลลิเมตร ดังแสดงในรูปที่ 3.2 และชิ้นส่วนที่ช่วยยึด rod shaft เข้าด้วยกัน ดังแสดงในรูปที่ 3.3 จะทำการออกแบบด้วยโปรแกรม Autodesk Inventor 2017 จากนั้นนำไปขึ้นรูปด้วยเครื่องพิมพ์สามมิติ (3D Printer) โดยวัสดุที่ใช้ขึ้นรูปคือ พิลลาเมนต์

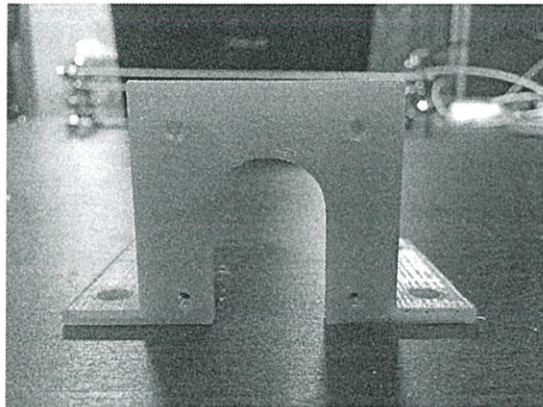


รูปที่ 3.2 rod shaft

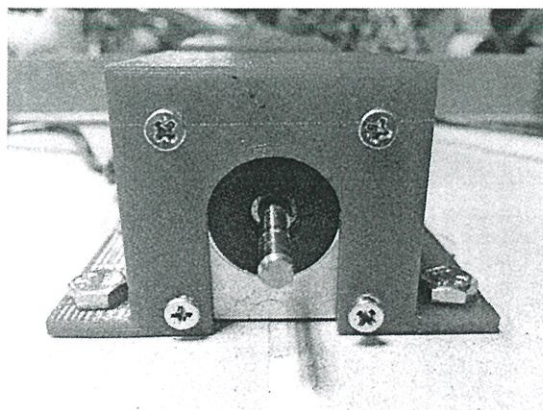


รูปที่ 3.3 ชิ้นส่วนที่ยึด rod shaft เข้าด้วยกัน

3.4.1.4.2 ชิ้นส่วนที่ช่วยยึด stepper motor ไม่ให้เคลื่อนที่ ด้วยกัน จะทำการออกแบบด้วยโปรแกรม Autodesk Inventor 2017 จากนั้นนำไปขึ้นรูปด้วยเครื่องพิมพ์สามมิติ (3D Printer) โดยวัสดุที่ใช้ขึ้นรูปคือ พิลามেন্ট ดังแสดงในรูปที่ 3.4 และ 3.5

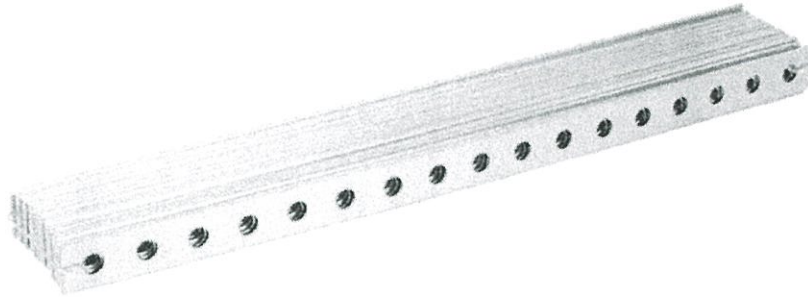


รูปที่ 3.4 ชิ้นส่วนที่ช่วยยึดสเต็ปเปอร์มอเตอร์ (stepper motor) ไม่ให้เคลื่อนที่

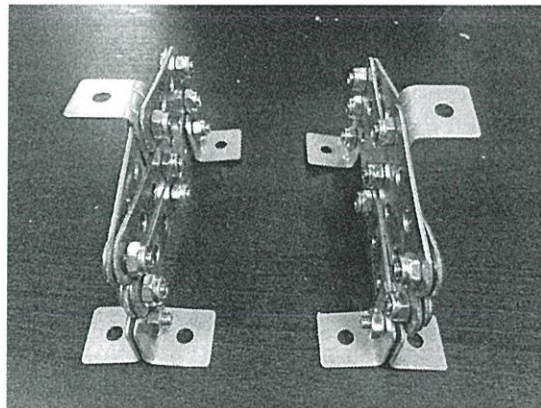


รูปที่ 3.5 ประกอบเข้ากับสเต็ปเปอร์มอเตอร์ (stepper motor)

3.4.1.4.3 ส่วนที่เป็นกลไกช่วยให้เตียงเคลื่อนที่ขึ้นลงได้ ใช้ระบบเดียวกับ scissor lift วัสดุที่นำมาใช้คือ เหล็กชุบ ดึงแสดงในรูปที่ 3.6 โดยนำมาตัดและประกอบกัน มีฉากเหล็ก เจาะรูด้านบนและฐานเพื่อเชื่อมกับพื้นเตียงและไม้ ดึงแสดงในรูปที่ 3.7 มีสตัดเกลียวและชิ้นส่วนที่ปรับสามมิติใส่ไว้ตรงกลางเพื่อช่วยให้ กลไกการขึ้นและลงอยู่ในระนาบเดียวกัน



รูปที่ 3.6 เหล็กชุบ



รูปที่ 3.7 ประกอบรวมกัน

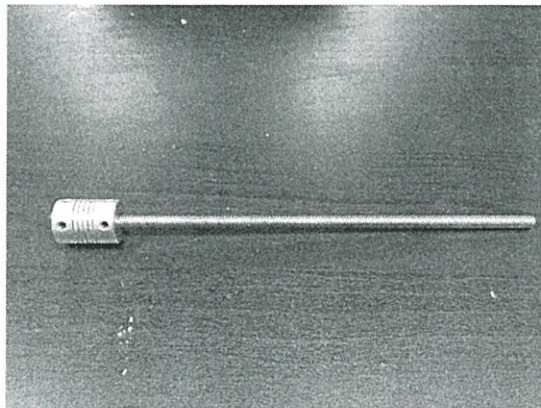
- 3.4.1.4.4 ส่วนที่เชื่อม stepper motor และ scissor lift ประกอบไปด้วย  
 - สตัดเกลียว : ขนาดเส้นผ่านศูนย์กลาง 6 มิลลิเมตร ตัวอย่างดัง

แสดงในรูปที่ 3.8



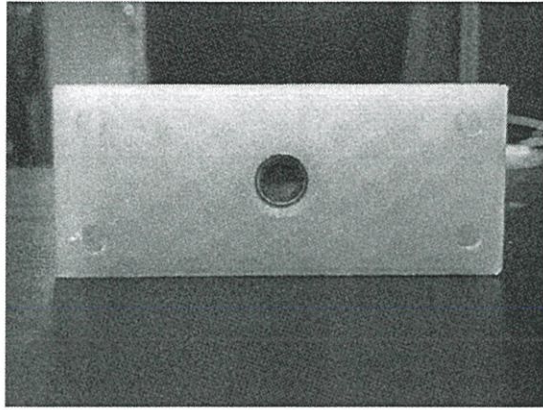
รูปที่ 3.8 สตัดเกลียว

- coupling : ขนาดเส้นผ่านศูนย์กลาง 2.5 มิลลิเมตร ดังแสดงใน  
 รูปที่ 3.9 ช่วยยึดระหว่างสตัดเกลียวและ stepper motor



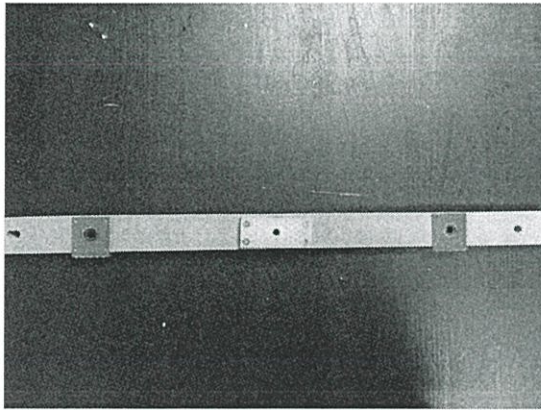
รูปที่ 3.9 coupling ต่อกับสตัดเกลียว

- ชิ้นส่วนที่ใช้เชื่อมกับไม้ : เพื่อให้สตัดเกลียวสามารถเคลื่อนที่ได้ ทำการออกแบบด้วยโปรแกรม Autodesk Inventor 2017 จากนั้นนำไปขึ้นรูปด้วยเครื่องพิมพ์สามมิติ (3D Printer) ดังแสดงในรูปที่ 3.10 โดยวัสดุที่ใช้ขึ้นรูปคือ พลาสติก โดยตรงกลางมีรูเพื่อใส่น็อตตัวเมียขนาดเส้นผ่านศูนย์กลาง 6 มิลลิเมตร



รูปที่ 3.10 ชิ้นส่วนที่ใช้เชื่อมกับไม้

- ไม้ เชื่อมระหว่างสตั๊ดเกลียว , linear guide และ scissor lift  
 ดังแสดงในรูปที่ 3.11 เป็นชิ้นส่วนที่ช่วยให้กลไกการทำงานสามารถเคลื่อนที่ได้

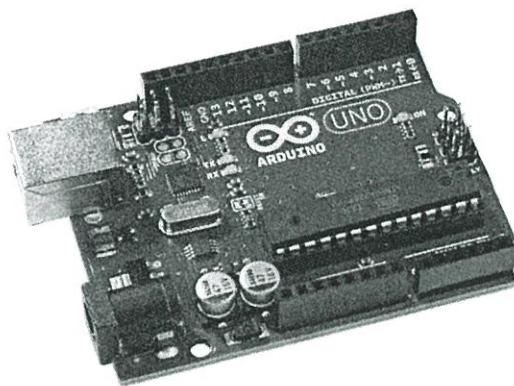


รูปที่ 3.11 ไม้เป็นตัวกลางเพื่อช่วยให้ระบบ scissor lift เคลื่อนที่ได้

### 3.4.2 การออกแบบส่วน Software

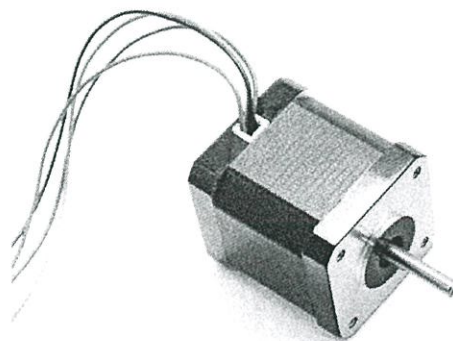
ในการออกแบบส่วน software จะใช้วิธีการเขียนโค้ดภาษาซีร่วมกับการทำงานของ Arduino ต้นแบบเตียงพลิกผู้ป่วยนี้จะใช้ stepper motor ทำงานร่วมกับ Motor Drive Board , push button และ Arduino โดยเริ่มจากการกดปุ่มขึ้นและลง จากนั้น stepper motor ก็จะทำหน้าที่ทำให้เตียงเคลื่อนที่ขึ้นลงได้ และปล่อยปุ่มกดเมื่อต้องการหยุดการเคลื่อนที่ของเตียง โดย Stepper motor , Motor Drive Board และ Arduino อย่างละ 1 ตัว จะควบคุมการเคลื่อนที่ของต้นแบบเตียงพลิกผู้ป่วย 1 ผืน โดยอุปกรณ์อิเล็กทรอนิกส์ทั้งที่ใช้ทั้งหมด มีดังต่อไปนี้

3.4.2.1 Arduino Uno ใช้เป็นไมโครคอนโทรลเลอร์เพื่อประมวลผล ส่งค่าไปยังอุปกรณ์ต่างๆ และแสดงผลออกมา ดังแสดงในรูปที่ 3.12 แสดงตัวอย่างบอร์ด Arduino Uno ที่ใช้ในการสั่งการทำงานอุปกรณ์ต่างๆ



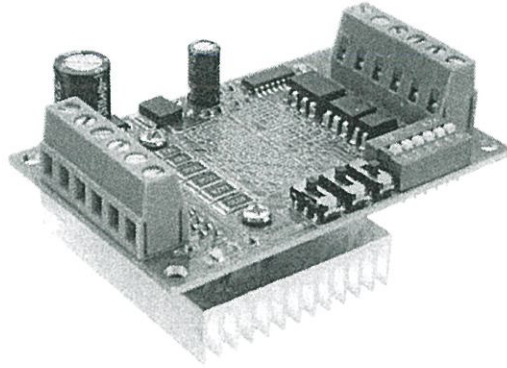
รูปที่ 3.12 อาร์ดูโน้ อูโน่ (Arduino Uno)

3.4.2.2 Stepper Motor แรงบิด 0.55 N.m ดังแสดงในรูปที่ 3.13 สเต็ปปีงมอเตอร์ เป็นโมดูลที่เหมาะสมกับงานที่ต้องการตำแหน่งของการหมุนที่แม่นยำมาก และใช้งานง่ายมอเตอร์ชนิดนี้ จะใช้งานได้หลากหลายแบบในเครื่องจักรหรืออุปกรณ์ที่ต้องการ มุมหรือองศาของการหมุนที่แน่นอน แม่นยำ และควบคุมการหมุนด้วยสัญญาณพัลส์ โดยอุปกรณ์ที่ออกแบบจะใช้โมดูล 2 ตัว ทำงานร่วมกับ Stepper motor driver board และ Arduino เพื่อควบคุมการเคลื่อนที่ของเตียง



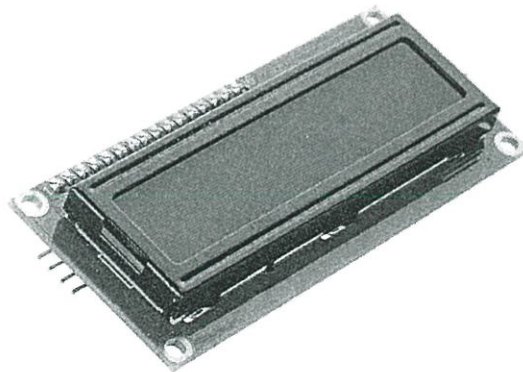
รูปที่ 3.13 สเต็ปเปอร์มอเตอร์ (Stepper Motor)

3.4.2.2 TB6560 Stepper Motor Driver เป็นโมดูลที่ใช้ในการขับ Stepper Motor ดังแสดงในรูปที่ 3.14 สามารถกำหนดค่าการทำงานโดยทำการปรับสวิตช์ส์ สามารถทำงานได้ 5-24 VDC, 0.3-3A โดยอุปกรณ์ที่ออกแบบจะใช้โมดูล 2 ตัว ทำงานร่วมกับ Stepper motor และ Arduino เพื่อควบคุมการเคลื่อนที่ของเตียง



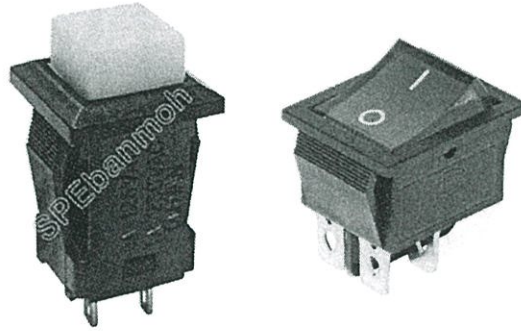
รูปที่ 3.14 TB6560 Stepper Motor Driver

3.4.2.3 LCD I2C เป็นโมดูลที่เป็นจอ LCD ทัวไปที่มาพร้อมกับบอร์ด I2C Bus พร้อมกับ VR สำหรับปรับความเข้มของจอ ดังแสดงในรูปที่ 3.15 ในรูปแบบ I2C จะใช้ขาการเชื่อมต่อกับ Microcontroller เพียง 4 ขา ทำให้ใช้งานได้ง่ายและสะดวกมากขึ้น โดยใช้ LCD I2C ทำงานร่วมกับ Arduino เพื่อจับเวลาในการพลิกตัวผู้ป่วยแต่ละครั้ง



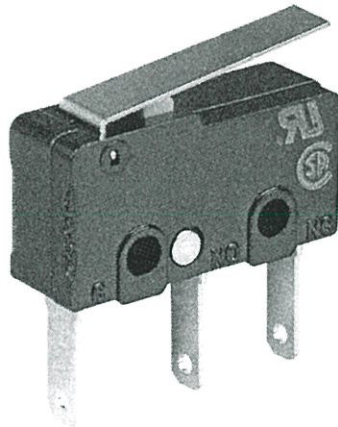
รูปที่ 3.15 LCD I2C

3.4.2.4 Push button switch เป็นโมดูลทางไฟฟ้า ทำหน้าที่ตัดและต่อวงจรทางไฟฟ้า และใช้ในการควบคุมการทำงานของมอเตอร์หรือเครื่องจักรต่างๆ เป็นอุปกรณ์พื้นฐาน ใช้งานได้กับอุตสาหกรรมทั่วไป มีทั้งแบบมีไฟ และทึบแสง โดยจะใช้ push button switch 2 แบบ คือ มีไฟและทึบแสง ดังแสดงในรูปที่ 3.16(ซ้าย) โดยแบบทึบแสงจะใช้ทั้งหมด 3 ปุ่ม คือ ปุ่มซ้าย ปุ่มขวา และปุ่มรีเซต และแบบมีไฟ ดังแสดงในรูปที่ 3.16(ขวา) จะใช้สำหรับเป็น สวิตซ์เปิดปิด



รูปที่ 3.16 Push button switch

3.4.2.5 ลิ้มิตสวิตซ์ซ์ (Limit switch) เป็นสวิตซ์ซ์ที่จำกัดระยะทาง การทำงานอาศัยแรงกดภายนอกมากระทำเช่น วางของทับที่ปุ่มกดหรือลูกเบี้ยวมาชนที่ปุ่มกด และเป็นผลทำให้หน้าสัมผัสที่ต่ออยู่กับก้านชน เปิด-ปิด ตามจังหวะของการชน ดังแสดงในรูปที่ 3.17 ใช้โมดูลนี้เพื่อเป็นการป้องกันหากเกิดความผิดพลาดเมื่อกดปุ่มรีเซต โดยเมื่ออุปกรณ์เคลื่อนที่แตะลิ้มิต สวิตซ์ซ์เมื่อไหร่ มอเตอร์จะหยุดการทำงานทันที



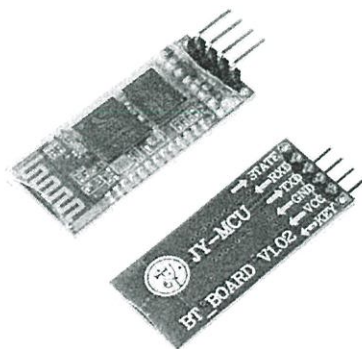
รูปที่ 3.17 ลิ้มิตสวิตซ์ซ์ (Limit switch)

3.4.2.6 บัลซเซอร์ (Buzzer) คือ ลำโพงแบบแม่เหล็กหรือแบบเปียโซที่มีวงจรกำเนิดความถี่ (oscillator) อยู่ภายในตัว ดังแสดงในรูปที่ 3.18 ใช้ไฟเลี้ยง 3.3 – 5V สามารถสร้างเสียงเตือนหรือส่งสัญญาณที่เป็นรูปแบบต่างๆ โดยใช้โมดูลนี้เพื่อเป็นการแจ้งเตือนผ่านเสียงเมื่อถึงเวลาที่กำหนดไว้



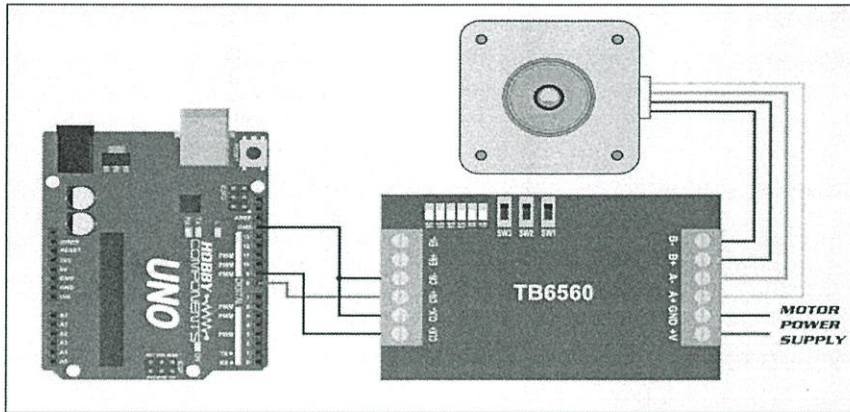
รูปที่ 3.18 บัลซเซอร์ (Buzzer)

3.4.2.7 Bluetooth module HC05 เป็นโมดูล Bluetooth ที่ใช้งานในการเชื่อมต่อกับสมาร์ตโฟนต่างๆ ให้สมาร์ตโฟนสามารถสื่อสารกับไมโครคอนโทรเลอร์ (Arduino AVR PIC etc.) ได้ ผ่าน Serial port ดังแสดงในรูปที่ 3.19 โมดูลรุ่น HC05 สามารถตั้งให้ใช้งานเป็นได้ทั้งโหมด Master (ให้อุปกรณ์อื่นมาเชื่อมต่อ) และโหมด Slave (เชื่อมต่อกับอุปกรณ์อื่น) โดยใช้โมดูลนี้ร่วมกับ Arduino เชื่อมต่อกับแอปพลิเคชันในสมาร์ตโฟน เพื่อแจ้งเตือนบนสมาร์ตโฟนของผู้แล แม้ว่าผู้ดูแลจะไม่ได้นั่งเฝ้าผู้ป่วยในตอนนั้นก็ตาม

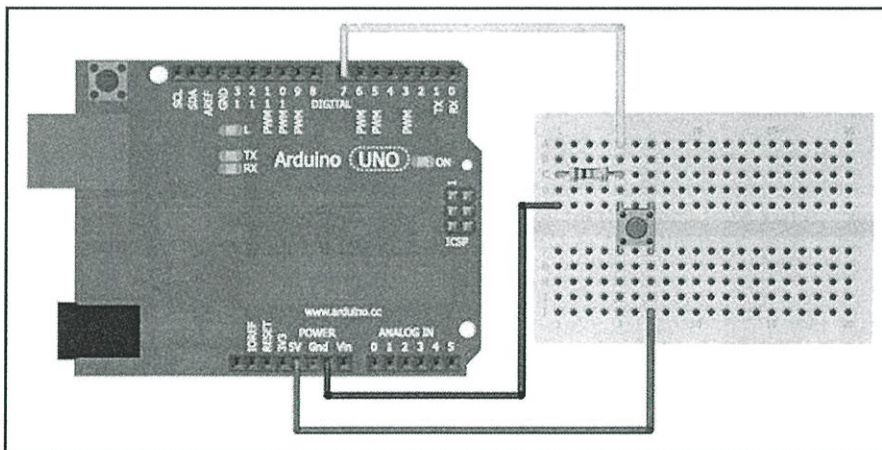


รูปที่ 3.19 Bluetooth module HC05

3.4.2.8 การทำให้ Stepper motor หมุน โดยการต่อ stepper motor เข้ากับ motor drive board และ ต่อปุ่มกด เข้ากับ Arduino ดังแสดงในรูปที่ 3.20 และ 3.21 เป็นการเชื่อมต่อระหว่าง Arduino, Stepper Motor, Motor Driver Board และปุ่มกดเข้าด้วยกัน เพื่อรวมให้สามารถสั่งการทำงานของ motor ผ่านการกดปุ่มได้



รูปที่ 3.20 การต่อ stepper motor , motor drive board และ Arduino



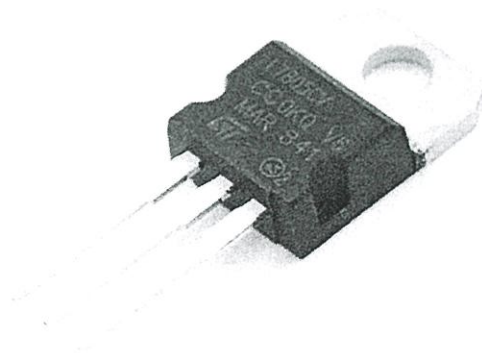
รูปที่ 3.21 การต่อปุ่มกดเข้ากับ Arduino

วิธีการต่อ Stepper motor เข้ากับ Motor driver board และ Arduino

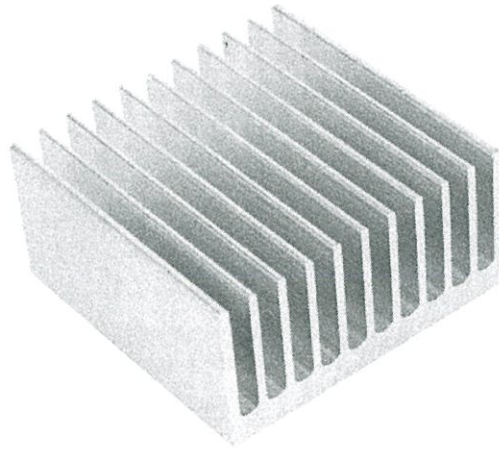
ตารางที่ 3.1 แสดงการต่อ Stepper motor กับ Motor driver board และ Arduino

Stepper Motor	Motor driver board	Arduino
สายสีดำ	B-	-
สายสีเขียว	B+	-
สายสีน้ำเงิน	A-	-
สายสีแดง	A+	-
-	GND	GND
-	+24V	5V
-	CW-	10(Digital)
-	CW+	GND(Digital)
-	CLK-	GND(Digital)
-	CLK+	11(Digital)

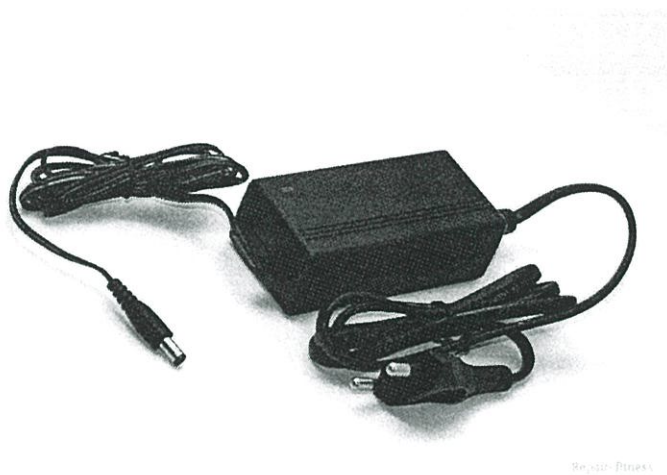
3.4.2.9 แหล่งจ่ายพลังงานให้กับระบบอิเล็กทรอนิกส์ทั้งหมด เนื่องจาก Arduino และ Motor driver board จำเป็นต้องใช้ไฟ 5V และ 12V ตามลำดับ โดยไม่ต้องการใช้ supply ต่อเข้ากับ Motor driver board และ Arduino ต่อเข้ากับ โน้ตบุ๊ก จึงใช้ Adaptor แทน Supply และ โน้ตบุ๊ก โดยจำเป็นต้องใช้อุปกรณ์แปลงไฟเข้ามาช่วย คือ IC Linear Regulator (IC เบอร์ 7805) และ Heat Sink ดังแสดงในรูปที่ 3.22 และ 3.3 นำมาต่อร่วมกัน



รูปที่ 3.22 IC 7805

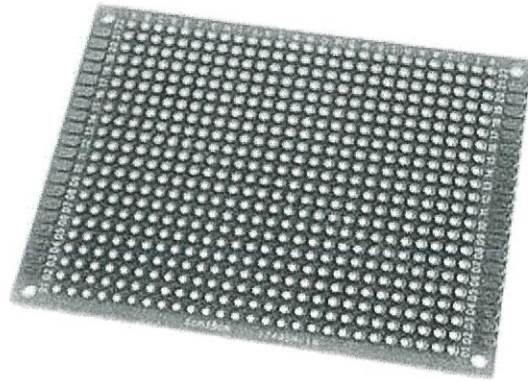


รูปที่ 3.23 Heat sink



รูปที่ 3.24 Adapter

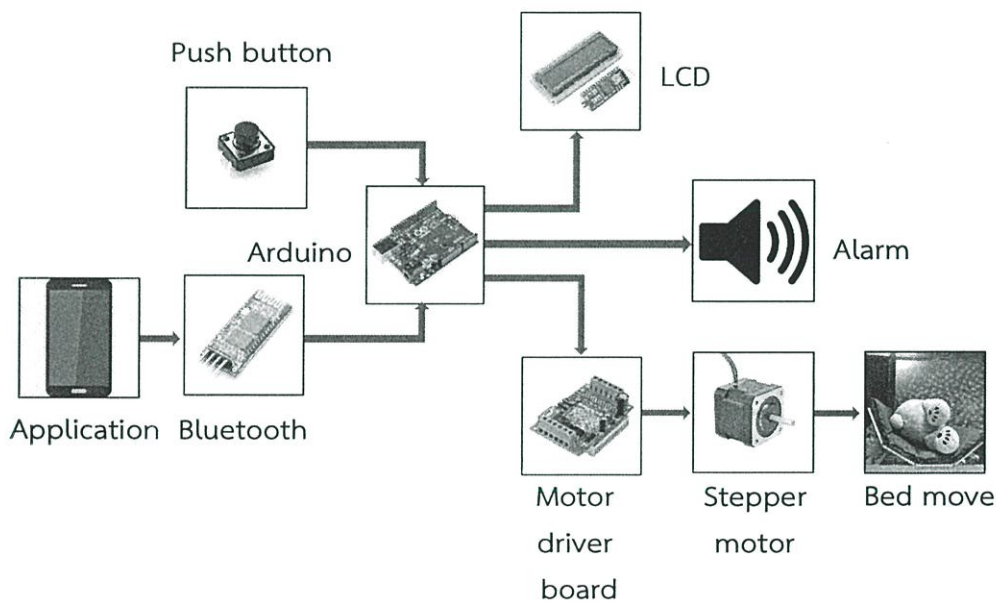
3.4.2.10 PCB (Print Circuit Board) คือ ส่วนประกอบพื้นฐานที่สำคัญของวงจรอิเล็กทรอนิกส์ เป็นทางเดินสัญญาณไฟฟ้าของอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ที่อยู่บนแผงวงจร ทำให้อุปกรณ์ต่าง ๆ เชื่อมต่อกันได้ ดังแสดงในรูปที่ 3.25 และสามารถทำงานได้อย่างถูกต้องตามที่ได้ออกแบบไว้ เลือกใช้ PCB ประเภทแผ่นวงจรพิมพ์อเนกประสงค์ (Universal PCB Board) โดยทำการบัดกรีเพื่อรวมไมโครคอนโทรลเลอร์ อุปกรณ์อิเล็กทรอนิกส์ต่างๆ และแหล่งจ่ายพลังงานเข้าด้วยกัน



รูปที่ 3.25 แผ่นวงจรพิมพ์อเนกประสงค์ (Universal PCB Board)

#### 3.4.2.11 หลักการทำงานของอุปกรณ์

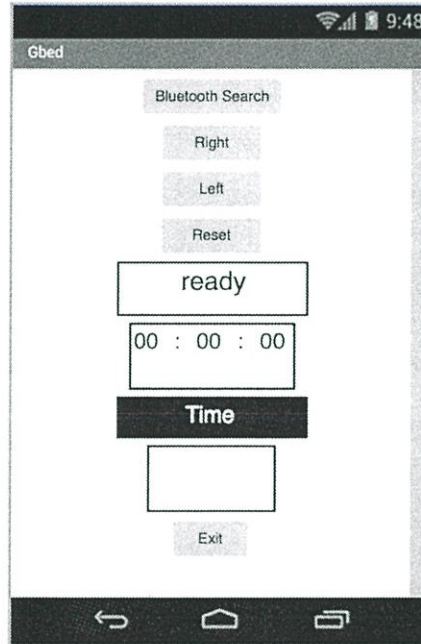
รูปที่ 3.26 แสดงแผนผังการทำงานของอุปกรณ์ทั้งหมด ในการควบคุมและสั่งการทำงานของ motor ทำให้เตียงสามารถเคลื่อนที่ขึ้นและลงเพื่อพลิกตัวผู้ป่วยได้



รูปที่ 3.26 แผนผังแสดงการทำงานของอุปกรณ์ ในการควบคุมการเคลื่อนที่ของเตียงช่วยพลิกตัวผู้ป่วย

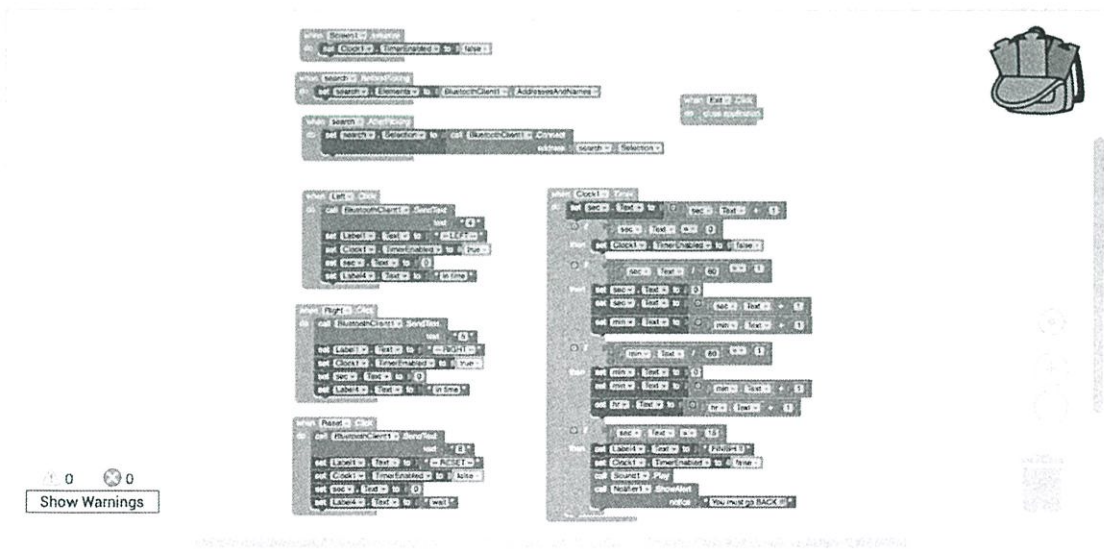
### 3.4.3 การออกแบบส่วนแอปพลิเคชัน

ในการออกแบบแอปพลิเคชัน ออกแบบและสร้างด้วยโปรแกรม Inventor app โดยในแอปพลิเคชันจะประกอบไปด้วย การเลือกกดปุ่มซ้ายขวาหรือรีเซ็ต และการแสดงเวลาในขณะนั้น ดังแสดงในรูปที่ 3.27 เมื่อผู้ดูแลกดเลือกซ้ายหรือขวาจะเริ่มจับเวลา เมื่อถึงเวลาดังไว้ก็จะมีเสียงเตือน โดยโปรแกรมจะทำงานร่วมกับ Arduino และมีการติดต่อกับอุปกรณ์ผ่านทาง Serial Port จากการเชื่อมต่อ Bluetooth



รูปที่ 3.27 การออกแบบหน้าจอแสดงผล

รูปที่ 3.28 แสดงตัวการสร้าง Block คำสั่งในแอปพลิเคชันในการสั่งการทำงานของ motor โดยส่งผ่าน Bluetooth ไปยัง Arduino



รูปที่ 3.28 การออกแบบการทำงานในโปรแกรม Inventor app

รูปที่ 3.29 แสดงหน้าจอที่ใช้แสดงผลในแอปพลิเคชัน โดยจะมีปุ่มควบคุมการทำงาน แสดงเวลาที่จับ และส่งสัญญาณเสียงเตือนเมื่อถึงเวลาที่กำหนด



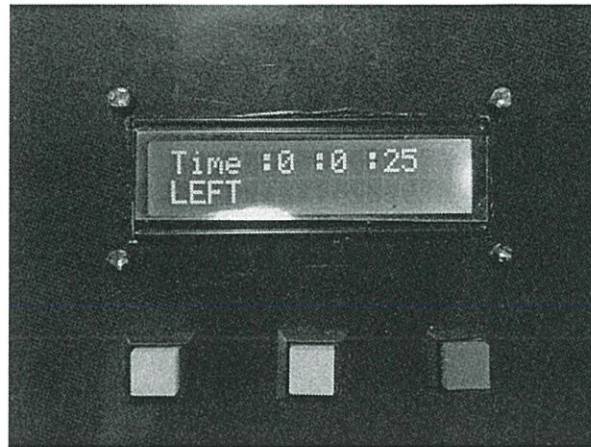
รูปที่ 3.29 หน้าจอแสดงในแอปพลิเคชัน

### 3.5 ขั้นตอนการทดลองการทำงานของอุปกรณ์

การทดลองการเคลื่อนที่ของอุปกรณ์ ออกแบบส่วน hardware และส่วน software ทำงานร่วมกับอุปกรณ์อิเล็กทรอนิกส์ทั้งหมดรวมกับการเขียนโค้ดภาษาซี โดยเพียงพลิกตัวผู้ป่วยจะแบ่งการสั่งการออกเป็น 2 แบบ คือ สั่งการด้วย สวิตช์และสั่งการผ่านแอปพลิเคชัน มีการจับเวลาและการแจ้งเตือนที่เมื่อถึงเวลาที่ต้องพลิกตัวผู้ป่วยอีกครั้งจะมีการแจ้งเตือนด้วยเสียงและแจ้งเตือนผ่านแอปพลิเคชัน

#### 3.5.1 การสั่งการผ่านสวิตช์

ทำการทดลองโดยการติดตั้ง push button switch ทดสอบว่าเมื่อกด สวิตช์แล้วเพียงเคลื่อนที่เป็นไปตามที่กำหนดหรือไม่ โดยจะแบ่งออกเป็น 3 ปุ่ม คือ พลิกทางซ้าย พลิกทางขวา และรีเซ็ตการทำงาน ยกตัวอย่างเช่น เมื่อกดปุ่มพลิกทางซ้ายเพียงทั้งสองฝั่งจะยกขึ้นในองศาที่ต่างกัน แสดงสถานะ ณ ขณะนั้น มีการจับเวลาและแจ้งเตือนเมื่อถึงเวลาที่กำหนด ดังแสดงในรูปที่ 3.30 จากนั้นกดปุ่มรีเซ็ตเพียงจะเคลื่อนที่กลับมาที่จุดเดิม เป็นต้น



รูปที่ 3.30 ตัวอย่างการสั่งการผ่านสวิตช์

### 3.5.2 การสั่งการผ่านแอปพลิเคชัน

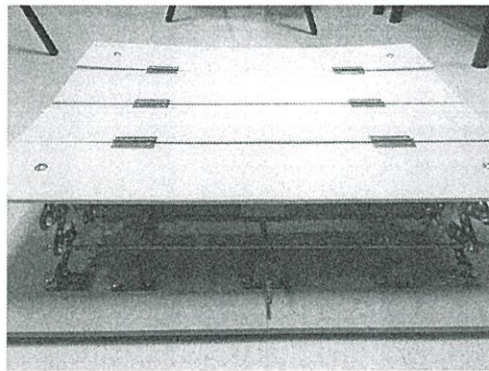
ทำการทดลองโดยการเขียนแอปพลิเคชันผ่านโปรแกรม inventor แสดงผลที่สมาร์ทโฟนผ่านการเชื่อมต่อ Bluetooth เมื่อสั่งการผ่านแอปพลิเคชันหากถึงเวลาที่กำหนดก็จะมีแจ้งเตือนที่สมาร์ทโฟนของผู้ดูแล เป็นต้น

## บทที่ 4 ผลการวิจัย

### 4.1 การออกแบบและสร้าง Hardware

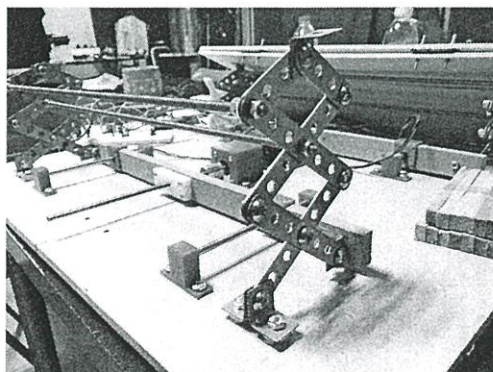
จากการทดลองใช้ไม้ MDF (Medium Density Fiber Board) ในการทำพื้นเตียงพบว่า มีน้ำหนักมาก ทำให้มีปัญหาในการควบคุมการเคลื่อนที่ขึ้นลง ผู้วิจัยจึงตรวจสอบและจัดหาวัสดุใหม่เพื่อนำมาใช้ทำพื้นเตียง ซึ่งวัสดุดังกล่าวต้องมีน้ำหนักเบา มีความทนทานและสามารถรับน้ำหนักได้ดี

และพบว่าไม้ Plaswood สามารถนำมาใช้ทดแทนไม้ MDF ได้ ถูกนำมาใช้ทำพื้นเตียงแทนไม้ MDF เนื่องจากมีลักษณะเบา มีความคงทนต่อการถูกร่อนและไม่ดูดซับความชื้น จึงไม่บวมหรือพองเมื่อโดนน้ำ ดังแสดงในรูปที่ 4.1 นอกจากนี้ยังเป็นวัสดุที่ติดไฟได้ง่าย จึงมีความปลอดภัยและมีความเหมาะสมในการจัดทำเป็นพื้นเตียง



รูปที่ 4.1 พื้นเตียง

ในส่วนของระบบการเคลื่อนที่ขึ้นลงของแบบจำลองเตียงนั้นใช้ระบบ scissors lift ดังแสดงในรูปที่ 4.2 เพื่อลดการใช้พื้นที่ เมื่อทดสอบการใช้งานพบว่า ไม่มีส่วนใดของระบบยื่นออกมาอย่างเดียว ทำให้มีความปลอดภัยเมื่อใช้งาน



รูปที่ 4.2 ระบบ scissors lift

## 4.2 การออกแบบแอปพลิเคชัน

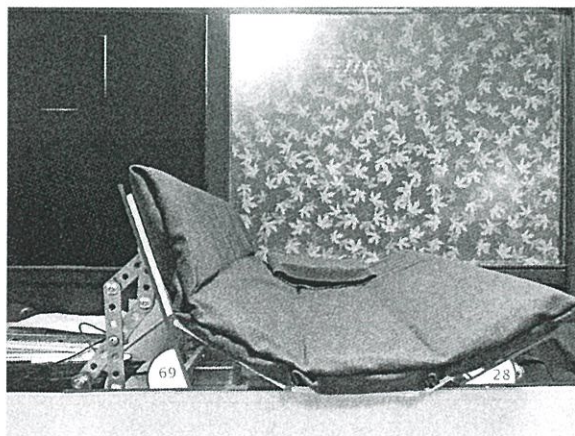
จากการใช้โปรแกรม MIT app inventor ในการออกแบบแอปพลิเคชัน พบว่าสามารถควบคุม Stepper Motor จาก Android ได้โดยส่งข้อมูลผ่าน Bluetooth ไปยัง Arduino หลังจากเพียงถูกยกขึ้นแล้วจะเริ่มจับเวลาทันที โดยจะแสดงสถานะ ณ ช่วงเวลานั้นด้วย ดังแสดงในรูปที่ 4.3 เมื่อถึงเวลาที่กำหนด (ประมาณ 2 ชั่วโมงในการพลิกตัวผู้ป่วยอีกครั้ง) ภายในแอปพลิเคชัน ส่งสัญญาณเสียงเพื่อเตือนให้ผู้ดูแลได้ทำการพลิกตัวผู้ป่วยไปอีกด้านและได้มีโอกาสเข้ามาตรวจสอบความเรียบร้อยของผู้ป่วยอีกด้วย



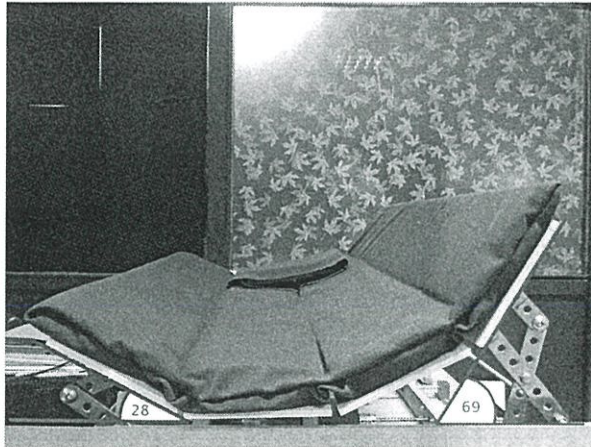
รูปที่ 4.3 หน้าต่างแอปพลิเคชัน จาก Android

## 4.3 การทดสอบ Hardware และ Software

เมื่อนำแบบจำลองเตียงมาที่ถูกสร้างขึ้นมาใช้งานร่วมกับซอฟต์แวร์ที่ถูกออกแบบเพื่อสั่งการทำงานของอุปกรณ์ต่างๆ แล้วทดสอบการเคลื่อนที่ขึ้นได้ผลดังรูป



รูปที่ 4.4 เตียงเมื่อถูกพลิกไปทางขวา



รูปที่ 4.5 เติงเมื่อถูกพลิกไปทางซ้าย

รูปที่ 4.4 และ 4.5 แสดงให้เห็นว่า เมื่อเตียงถูกพลิกไปด้านขวาและด้านซ้าย พื้นเตียงฝั่งที่ถูกยกขึ้นสูงขึ้นเพื่อใช้พลิกตัวผู้ป่วยนั้น ยกสูงเป็นมุม 69 องศา และพื้นเตียงฝั่งด้านที่ยกขึ้นเพื่อป้องกันผู้ป่วยตกเตียง ยกสูงขึ้นเป็นมุม 28 องศา

ในส่วนของการแสดงผล เมื่อเริ่มต้นการทำงานโดยการกดปุ่ม left, right หรือ reset หน้าจอ LCD จะแสดงเวลาและสถานะขณะนั้น ดังแสดงในรูปที่ 4.6 และเมื่อถึงเวลาที่กำหนดไว้ จะส่งสัญญาณเสียงเพื่อแจ้งเตือน



รูปที่ 4.6 การแสดงผลบนหน้าจอ LCD

## บทที่ 5

# สรุปผลการวิจัยและข้อเสนอแนะ

### 5.1 สรุปผลการวิจัย

จากการออกแบบและสร้างแบบจำลองเตียงช่วยพลิกตัวผู้ป่วย พบว่าแบบจำลองเตียงดังกล่าวสามารถทำงานได้ตามที่ตั้งสมมติฐานไว้ โดยพื้นเตียงที่ถูกแบ่งเป็น 4 ส่วนนั้น 2 ส่วนช่วยในการรับน้ำหนักตัวผู้ป่วย อีก 2 ส่วนช่วยในการพลิกตัวและป้องกันผู้ป่วยตกเตียง โดยเมื่อพื้นเตียงถูกยกขึ้นเพื่อพลิกตัวผู้ป่วยจะมีขนาดมุม 69 องศาจากระนาบพื้น และนอกจากการควบคุมผ่านปุ่มกดแล้ว การสั่งการผ่านแอปพลิเคชัน ก็สามารถทำงานได้อย่างดีเยี่ยม แต่จากการออกแบบและการเลือกใช้อุปกรณ์นั้นแสดงให้เห็นว่า งานวิจัยนี้เป็นเพียงการออกแบบเพื่อศึกษาหลักการเคลื่อนที่ของระบบเท่านั้น ไม่สามารถรับน้ำหนักตัวผู้ป่วยที่แท้จริงได้ ซึ่งหากจะสร้างเพื่อนำไปใช้งานจริง จำเป็นต้องศึกษา ออกแบบ และเปลี่ยนชนิดวัสดุที่ใช้ให้เหมาะสม

### 5.2 ปัญหาและแนวทางแก้ไขปัญหา

#### 5.2.1 กลไกการยกขึ้นลงไม่ปกติ

ในขั้นตอนการเลือกวัสดุในการทำพื้นเตียงนั้น พบว่าพื้นเตียงที่ทำจากวัสดุ MDF ไม่สามารถทำให้กลไกการยกขึ้นลงเป็นไปได้อย่างปกติ เนื่องจากไม่ดังกล่าวน้ำหนักที่มากเกินไป และปัญหาการเลือกใช้ Linear guide นั้นพบว่า การวัดที่ผิดพลาดจะทำให้การเคลื่อนที่ไม่เปลี่ยนเส้นตรงเกิดการติดขัด ทำให้ไม่สามารถยกขึ้นลงได้ปกติเช่นเดียวกัน

แนวทางแก้ไข : เปลี่ยนชนิดของไม้ที่ใช้จาก MDF เป็น plaswood ที่มีคุณสมบัติดีกว่า และน้ำหนักเบาลง รวมถึงวัดระยะ ออกแบบ และสร้าง Linear guide ส่วนที่ผิดพลาดใหม่ให้แม่นยำยิ่งขึ้น

#### 5.2.2 การติดขัดระหว่างกลไกการยกขึ้นลงทำงาน

ไม้ MDF ที่ถูกนำมาใช้ในการเชื่อมกับ motor และผลักให้ระบบ scissors lift ยกขึ้นเกิดการแอ่นตรงกลางลำไม้ ซึ่งเป็นอีกสาเหตุหนึ่งที่ทำให้การเคลื่อนที่ขึ้นเกิดการฝืดและติดขัด

แนวทางแก้ไข : เปลี่ยนไม้เป็นวัสดุชนิดอื่นที่มีความแข็งแรง ทนทาน และไม่เกิดการแอ่น แต่เนื่องด้วยราคาของวัสดุอื่นสูง เช่น อะคริลิก ดังนั้นไม้ที่มีขนาดหนาพอสมควรจึงถูกนำมาใช้ทดแทน เพียงแต่เมื่อใช้เป็นเวลานานก็ทำให้เกิดการแอ่นสะสมจนเกิดปัญหาได้

### 5.3 ข้อเสนอแนะ

ในการออกแบบและสร้างชิ้นงานนั้นต้องคำนึงถึงผู้ใช้งานเป็นหลัก ต้องมีความรู้ความเข้าใจในการเลือกใช้อุปกรณ์และวัสดุ รวมทั้งความรู้ทางด้านซอฟต์แวร์ต่างๆที่นำมาใช้ในการควบคุม จะทำให้ผู้ใช้ ใช้งานได้อย่างมีประสิทธิภาพ และมีความปลอดภัยสูงสุด

## บรรณานุกรม

- [1] การควบคุม stepper motor ด้วยบอร์ด TB6560. 2559. [online]. Available : <http://www.tomelectronics1.com/webboard/viewtopic/7>
- [2] การฟื้นฟูดูแลผู้ป่วยติดเตียง. 2559. [online]. Available : <http://www.elifegear.com/การดูแลผู้ป่วยติดเตียง/>
- [3] ทวีป นวคุณานนท์. 2559. ส่วนประกอบของโปรแกรม App Inventor. [online]. Available : <https://programmingappinventor.wordpress.com/รู้จักกับ-app-inventor/ส่วนประกอบของโปรแกรม-app-inventor/>
- [4] ภาควิชาศัลยศาสตร์ออร์โธปิดิกส์ คณะแพทยศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่. 2561. การดูแลผู้ป่วยเพื่อป้องกันภาวะแทรกซ้อนจากการนอนติดเตียงเป็นเวลานาน. [online]. Available : <http://medinfo.psu.ac.th/pr/ortho/know4.html>
- [5] วัจนา ลีละพัฒนและสายพิน หัตถ์รัตน. 2561. เมื่อผู้ดูแลทำไม่ไหวแล้ว. [online]. Available : [https://med.mahidol.ac.th/fammed/th/article/postgrad/writer23/article\\_4](https://med.mahidol.ac.th/fammed/th/article/postgrad/writer23/article_4)
- [6] วิชัย โชควิวัฒน์. 2560. การดูแลผู้สูงอายุติดบ้านติดเตียง. [online]. Available : <http://www.thaihealth.or.th/Content/35697-การดูแลผู้สูงอายุติดบ้านติดเตียง.html>
- [7] วิพรรณ ประจวบเหมาะ. 2542. ผู้สูงอายุในประเทศไทย : แนวโน้ม คุณลักษณะ และปัญหา. [online]. Available : <http://hp.anamai.moph.go.th/soongwai/statics/about/soongwai/topic004.php>
- [8] วิทยา. 2556. App Inventor คืออะไร?. [online]. Available : <http://www.chakkham.ac.th/~krutum/web2013/index.php/2012-11-19-08-58-13/40-app-inventor>
- [9] ศิริกัญญา อุษาทพิริยกุล. 2561. กลไกการเกิดแผลกดทับ (Pathophysiology). [online]. Available : <https://sites.google.com/site/kanom2551/kkai-kar-keid-phael-kd-thab-pathophysiology>
- [10] สุรเดช. 2554. ไอซีเร็กกูเลเตอร์ (IC Regulator). [online]. Available : <http://electhai.blogspot.com/2011/06/ic-regulator.html>
- [11] หลักการทำงานของ Stepping Motor. 2559. [online]. Available : <https://www.factomart.com/th/factomartblog/principle-of-stepping-motor/>
- [12] หลากหลายวิธีการแปลงไฟ 12V to 5V. 2557. [online]. Available : <https://www.thaicconverter.com/article/2/หลากหลายวิธีการแปลงไฟ-12v-to-5v>
- [13] แผลกดทับ. 2558. [online]. Available : <https://www.pobpad.com/แผลกดทับ>
- [14] แผลกดทับ สาเหตุและวิธีดูแลผู้ป่วยแผลกดทับ. 2556. [online]. Available : <https://www.dmc.tv/pages//แผลกดทับ.html>
- [15] Anton Buchgraber. 2017. Tb6560 stepping motor driver V20. [online]. Available : <http://www.ribu.at/PDF/Tb6560.pdf>
- [16] Button Digital Input. 2559. [online]. Available : <http://commandronestore.com/learning/arduino002.php>

- [17] chang. 2554. **Arduino Tutorial 3 : Digital input ,Debounce.** [online]. Available : <http://www.ayarafun.com/2011/04/arduino-tutorial-3-digital-input-and-debounce/>
- [18] **Implementing a Step-Direction Interface-based Stepper Motor Controller using SmartFusion2 Devices.** 2014. [online]. Available : [https://www.microsemi.com/document-portal/doc\\_view/133502-ac413-implementing-a-step-direction-interface-based-stepper-motor-controller-using-sf2-devices](https://www.microsemi.com/document-portal/doc_view/133502-ac413-implementing-a-step-direction-interface-based-stepper-motor-controller-using-sf2-devices)
- [19] James Hewes. 2561. **แหล่งจ่ายไฟ (Power Supplies).** [online]. Available : <http://icelectronic.com/beginner/study/powersup.htm>
- [20] **MIT App Inventor เขียนโปรแกรมบนมือถือ.** 2561. [online]. Available : <https://kidsangsan.com/2012/07/04/mit-app-inventor-เขียนโปรแกรมบนมือถือ/>
- [21] **Push Button Switch คืออะไร ?** 2561. [online]. Available : <https://www.factomart.com/th/main-push-button/>

ภาคผนวก

## Arduino Code

```

#include <SoftwareSerial.h>
SoftwareSerial mySerial(6,5);
int TAPB ;
int TAP ;
const int buttonPinR = 7;
const int buttonPinL = 8;
const int buttonPinS = 9;
int limitL = 3, buttonStateLL;
int limitR = 4, buttonStateRR;
int buttonStateR = 0;
int buttonStateL = 0;
int buttonStateS = 0;
int stp1=11;
int dir1=10;
int stp2=13;
int dir2=12;
int NEWTIME = 0
int minutes2 = 0
int hours2 = 0
int NEWTIMESEC = 0
int NEWTIMEMIN = 0
int NEWTIMEHOUR = 0;
int Distance = 0;
int a=0;
long i=0;
long j=0;
int z=0;
int b = 0, sec = 0, minutes = 0, hours = 0, sec1 = 0;
char start;
int led=13;
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2);
ISR(TIMER1_COMPA_vect)      // timer1 Interrupt ทุกๆ 0.1 วินาที
{

```

```
b++;
if(b >= 10){
    sec++;
    sec1++;
    b = 0;
}
if(sec >= 60){
    minutes++;
    sec = 0;
}
if(minutes >= 60){
    hours++;
    minutes = 0;
}
}

//-----
-----//

void setup()
{
    mySerial.begin(9600);
    pinMode(buttonPinR, INPUT);
    pinMode(buttonPinL, INPUT);
    pinMode(buttonPinS, INPUT);
    pinMode(limitL, INPUT);
    pinMode(limitR, INPUT);
    pinMode(stp1, OUTPUT);
    pinMode(dir1, OUTPUT);
    pinMode(stp2, OUTPUT);
    pinMode(dir2, OUTPUT);
    noInterrupts();
    pinMode(led, OUTPUT);

    TCCR1A = 0;
    TCCR1B = 0;
    TCNT1 = 0;
```

```

OCR1A = 6250;          // 0.1 sec.
TCCR1B |= (1 << WGM12); // CTC mode
TCCR1B |= (1 << CS12); // 256 prescaler
TIMSK1 |= (1 << OCIE1A); // enable timer compare interrupt
interrupts();          // enable all interrupts
}

```

```

//-----
-----//

```

```

void loop()
{
  lcd.init();
  lcd.setCursor(0,0);
  lcd.backlight();
  lcd.print("Time : : ");

  buttonStateR = digitalRead(buttonPinR);
  buttonStateL = digitalRead(buttonPinL);
  buttonStateS = digitalRead(buttonPinS);
  buttonStateR == LOW ;
  buttonStateL == LOW ;
  buttonStateS == LOW ;

  if (mySerial.available()>0)
  {
    int z= mySerial.read();
  }

  if (buttonStateR == HIGH || z=='4' )
  {
    a=1;
    lcd.setCursor(0,1);
    lcd.print("LEFT");
  }

  else if (buttonStateL == HIGH || z=='5')
  {

```

```

a=2;
lcd.setCursor(0,1);
lcd.print("RIGHT");
}

else if (buttonStateS == HIGH || z=='6')
{
a=3;
lcd.setCursor(0,1);
lcd.print("RESET");
}
else a=0;

//-----
-----//

switch (a)
{
case 1:
TAPB = sec1;
for(i=0; i<=80000; i++)
{
Step2();
}
for(j=0; j<=30000; j++)
{
Step1();
}
for (;;)
{
NEWTIME = sec1-TAPB;
Time();
if (sec1-TAPB==50)
{
digitalWrite(led, HIGH);
break;
}}

a=0;

```

```

    break;
case 2:
TAPB = sec1;
for(j=0; j<=80000; j++)
    {
        Step1(); }
for(i=0; i<=30000; i++)
    {
        Step2();
    }
    for(;;)
        {
            NEWTIME = sec1-TAPB;
            Time();
            if (sec1-TAPB==50)
                {
                    digitalWrite(led, HIGH);
                    break;
                }
        }
a=0;
break;

case 3:
if (i>j)
    {
        for(i; i>=0; i--)
            {
                digitalWrite(stp2, LOW);
                delayMicroseconds(50);
                digitalWrite(stp2, HIGH);
                delayMicroseconds(50);
                digitalWrite(dir2,LOW);
            }
    }

for(j; j>=0; j--)
    {
        digitalWrite(stp1, LOW);

```

```

        delayMicroseconds(50);
        digitalWrite(stp1, HIGH);
        delayMicroseconds(50);
        digitalWrite(dir1,LOW);
    }
}

else if (j>i)
{
    for(j; j>=0; j--)
    {
        digitalWrite(stp1, LOW);
        delayMicroseconds(50);
        digitalWrite(stp1, HIGH);
        delayMicroseconds(50);
        digitalWrite(dir1,LOW);
    }
    for(i; i>=0; i--)
    {
        digitalWrite(stp2, LOW);
        delayMicroseconds(50);
        digitalWrite(stp2, HIGH);
        delayMicroseconds(50);
        digitalWrite(dir2,LOW);
    }
}

else
{
    buttonStateLL = digitalRead(limitL);
    while (buttonStateLL!=1 )
    {
        buttonStateLL = digitalRead(limitL);
        Down1();
        if(buttonStateLL==1 )
        {
            Stop1();
        }
    }
}

```

```

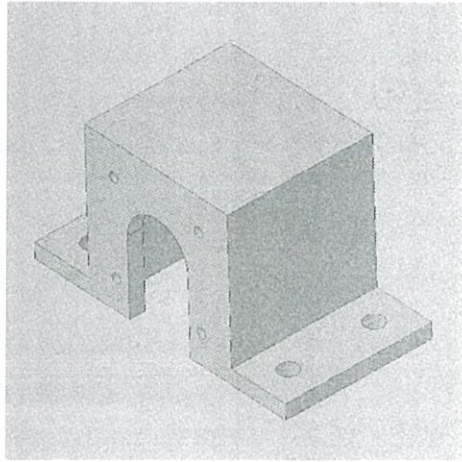
    }
    buttonStateRR = digitalRead(limitR);
    while (buttonStateRR!=1 )
    {
        buttonStateRR = digitalRead(limitR);
        Down2();
        if(buttonStateRR==1 )
        {
            Stop2();
        }
    }
}
a=0;
}
}
void Step1(void)
{
    digitalWrite(stp1, LOW);
    delayMicroseconds(50);
    digitalWrite(stp1, HIGH);
    delayMicroseconds(50);
    digitalWrite(dir1,HIGH);
}
void Step2(void)
{
    digitalWrite(stp2, LOW);
    delayMicroseconds(50);
    digitalWrite(stp2, HIGH);
    delayMicroseconds(50);
    digitalWrite(dir2,HIGH);
}
void Down1(void)
{
    digitalWrite(stp1, LOW);
    delayMicroseconds(50);
    digitalWrite(stp1, HIGH);
    delayMicroseconds(50);
    digitalWrite(dir1,LOW);
}

```

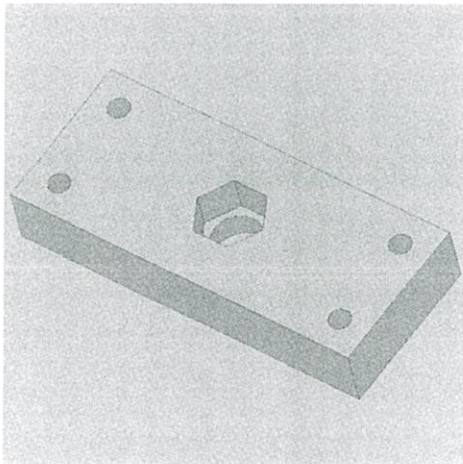
```
}  
void Down2(void)  
{  
    digitalWrite(stp2, LOW);  
    delayMicroseconds(50);  
    digitalWrite(stp2, HIGH);  
    delayMicroseconds(50);  
    digitalWrite(dir2,LOW);  
}  
void Stop1(void)  
{  
    digitalWrite(stp1, LOW);  
    delayMicroseconds(50);  
    digitalWrite(stp1, HIGH);  
    delayMicroseconds(50);  
    digitalWrite(dir1,HIGH);  
}  
void Stop2(void)  
{  
    digitalWrite(stp2, LOW);  
    delayMicroseconds(50);  
    digitalWrite(stp2, HIGH);  
    delayMicroseconds(50);  
    digitalWrite(dir2,HIGH);  
}  
void Time(void)  
{  
    NEWTIMESEC = NEWTIME%60 ;  
    if(NEWTIMESEC<10)  
    {  
        lcd.setCursor(12,0);  
        lcd.print(0);  
        lcd.setCursor(13,0);  
        lcd.print(NEWTIMESEC);  
    }  
  
    else
```

```
{
    lcd.setCursor(12,0);
    lcd.print(NEWTIMESEC);
}
lcd.setCursor(14,0);
lcd.println(" ");
NEWTIMEMIN = NEWTIME/60 ;
    if (NEWTIMEMIN<=59)
    {
        lcd.setCursor(9,0);
        lcd.print(NEWTIMEMIN);
    }
    else
    {
        lcd.setCursor(9,0);
        lcd.print(NEWTIMEMIN-60);
    }
    NEWTIMEHOUR = NEWTIME/3600 ;
    lcd.setCursor(6,0);
    lcd.print(NEWTIMEHOUR);
}
```

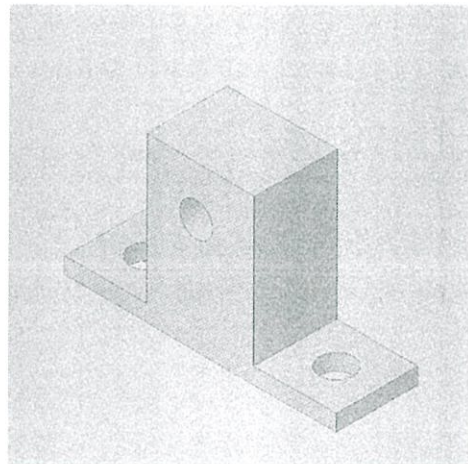
ภาพ CAD ของชิ้นส่วนต่างๆ โดยขึ้นรูปชิ้นงานจากเครื่องพิมพ์สามมิติ (3D Printer)



ชิ้นส่วนที่ช่วยยึดสตีปเปอร์มอเตอร์ (stepper motor)



ชิ้นส่วนที่ใช้เชื่อมกับไม้



ชิ้นส่วนที่ยึด rod shaft เข้าด้วยกัน