



EE04-แบบขอสอบป้องกัน

นักศึกษา ☐ ปกติ ☒ สมทบ

รายงานความก้าวหน้าโครงการวิศวกรรมไฟฟ้า

หลักสูตรสาขาวิชาวิศวกรรมไฟฟ้า

รหัสโครงการ : S1-634-CS2

ชื่อโครงการ

(ภาษาไทย) การพัฒนาลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ

(ภาษาอังกฤษ) Development of treadmills for patients and the elderly

ผู้จัดทำ

นายธีรพงษ์ รอดทองอ่อน รหัสนักศึกษา 160204132005 โทร. 0918265456

นายฟายัด ภูแห่ม รหัสนักศึกษา 160204132039 โทร. 0992279133

ความเห็นอาจารย์ที่ปรึกษาเข้าร่วม	ความเห็นอาจารย์ที่ปรึกษาหลัก
ได้ตรวจโครงการและรายงานฉบับสมบูรณ์แล้วเห็นควร ☐ อนุมัติสอบได้ ☐ ปรับปรุงเนื้อหาหรือรูปแบบใหม่ ลงชื่อ..... (.....)	ได้ตรวจโครงการและรายงานฉบับสมบูรณ์แล้วเห็น ☒ อนุมัติสอบได้ ☐ ปรับปรุงเนื้อหาหรือรูปแบบ ลงชื่อ..... (.....)

สาขาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย



การพัฒนาลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ
Development of treadmills for patients and the elderly

นายธีรพงษ์ รอดทองอ่อน

Teerapong Rodtongon

นายฟาฮัด กูแหม

Fahad Koomae

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

หลักสูตรสาขาวิชาวิศวกรรมไฟฟ้า

สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์

มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE BACHELOR DEGREE IN ELECTRICAL ENGINEERING

FACULTY OF ENGINEERING

RAJAMANGALA UNIVERSITY OF TECHNOLOGY SRIVIJAYA

2568

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

การพัฒนาลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ
Development of treadmills for patients and the elderly

นายธีรพงษ์ รอดทองอ่อน

Teerapong Rodtongon

นายฟาฮัด กูแหม

Fahad Koomae

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
หลักสูตรสาขาวิชาวิศวกรรมไฟฟ้า
สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE BACHELOR DEGREE IN ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING

RAJAMANGALA UNIVERSITY OF TECHNOLOGY SRIVIJAYA

2568

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย

ปริญญานิพนธ์ : การพัฒนาสู่วัยสำหรับผู้ป่วยและผู้สูงอายุ
ชื่อ : นายธีรพงษ์ รอดทองอ่อน
: นายฟายัด ภูแห่ม
สาขาวิชา : วิศวกรรมไฟฟ้า

ประธานที่ปรึกษา

กรรมการสอบ

.....
(ผู้ช่วยศาสตราจารย์ชาญณรงค์ พงศ์รักธรรม)

.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์พิทักษ์ สติธรรมธนะ)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ธีรพงษ์ ฉิมเพชร)

.....กรรมการ
(อาจารย์ธรรมนงค์ ทวีชื่น)

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย สงขลา อนุมัติให้ปริญญานิพนธ์นี้
เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า

.....
(ผู้ช่วยศาสตราจารย์ชาญณรงค์ พงศ์รักธรรม)
หัวหน้าหลักสูตร / ประธานหลักสูตร

Project Report Title : Fault Indicator in High Voltage Distribution System
Name : Mr.Teerapong Rodtongon
: Mr.Fahad Koomae
Major Field : Electrical Engineering

Major Advisor

.....
(Asst.Prof. Channarong Phongraktham)

Examining Committee

..... Chairperson
(Asst.Prof. Phithak Sathiwantana)

..... Committee
(Asst.Prof. Teerapong Chimphe)

..... Committee
(Lecturer Thamnong Taweechuen)

Accepted by the Faculty of Engineering, Rajamangala University of Technology
Srivijayain Partial Fulfillment of the Requirements for the Degree of Bachelor
ofElectrical Engineering

.....
(Asst.Prof. Channarong Phongraktham)
Chief Curriculum/Head of Curriculum

หัวข้อปริญญานิพนธ์	การพัฒนาลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ
โดย	นายธีรพงษ์ รอดทองอ่อน นายฟาฮัด กูแหม
อาจารย์ที่ปรึกษา	ผศ.ชาญณรงค์ พงศ์ภักดิ์ธรรม
ภาควิชา	วิศวกรรมไฟฟ้า
ปีการศึกษา	2568

บทคัดย่อ

โครงงานนี้นำเสนอการพัฒนาลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุโดยใช้ไมโครคอนโทรลเลอร์ควบคุมการทำงาน เพื่อแก้ปัญหาด้านการเคลื่อนไหวและการเดินเป็นอุปสรรคสำคัญที่ส่งผลกระทบต่อคุณภาพชีวิตของผู้ป่วยและผู้สูงอายุ โดยเฉพาะกลุ่มที่ยังคงมีความสามารถในการเดินได้ในระดับหนึ่งแต่ต้องการการฟื้นฟูสมรรถภาพ โครงงานนี้จึงมีวัตถุประสงค์ในการพัฒนาลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ โดยได้ทำการศึกษาข้อดีและข้อเสียของอุปกรณ์ช่วยเดินที่มีอยู่ในปัจจุบัน และนำมาต่อยอดเพื่อออกแบบระบบการทำงานให้มีประสิทธิภาพสูงสุด ตอบสนองความต้องการของผู้ป่วยที่จำเป็นต้องได้รับการฟื้นฟูร่างกายอย่างเหมาะสมและปลอดภัยครอบคลุมการพัฒนาอุปกรณ์ลู่วิ่งพร้อมจอแสดงผลจำนวน 1 เครื่อง ซึ่งมีความสามารถในการปรับความเร็วได้สูงสุด 5 กิโลเมตรต่อชั่วโมง เพื่อให้เหมาะสมกับระดับความสามารถของผู้ใช้งานแต่ละประเภทของผู้ใช้งาน นอกจากนี้อุปกรณ์ยังรองรับการฝึกเดิน 2 โหมด ได้แก่ โหมดแมนนวล ที่ผู้ใช้สามารถปรับเพิ่มหรือลดความเร็วได้ตามต้องการจากหน้าจอ และ โหมดอัตโนมัติ ที่ผู้ใช้สามารถเลือกฝึกตามระยะทางที่กำหนดไว้ระยะ 15 cm, ระยะ 20 cm, ระยะ 25 cm เพื่อความปลอดภัยสูงสุดของผู้ใช้งาน อุปกรณ์ยังถูกออกแบบมาเพื่อให้สามารถสั่งหยุดการทำงานของเครื่องได้ทันทีหากเกิดเหตุการณ์ผู้ป่วยล้ม พร้อมทั้งมีระบบแจ้งเตือนด้วยเสียง และเพื่อการติดตามผลการฟื้นฟู ระบบจะสามารถบันทึกข้อมูลระยะทางการเดินของผู้ป่วยได้สูงสุด 5 คน

(ปริญญานิพนธ์นี้มีจำนวนทั้งสิ้น 110 หน้า)

Project Report Title	Development of treadmills for patients and the elderly
By	Mr.Teerapong Rodtongon Mr.Fahad Koomae
Project Report Advisor	Asst.Prof. Channarong Phongraktham
Department of	Electrical Engineering
Academic Year	2025

Abstract

This project presents the development of a treadmill for patients and the elderly using a microcontroller to control the operation. To solve the problem of movement and walking, which are major obstacles affecting the quality of life of patients and the elderly, especially those who still have the ability to walk to a certain level but need rehabilitation. Therefore, this project aims to develop a treadmill for patients and the elderly by studying the advantages and disadvantages of current walking aids and developing them to design a system that is most efficient, meeting the needs of patients who need to receive proper and safe rehabilitation. It covers the development of 1 treadmill with a display screen, which can adjust the speed up to 5 kilometers per hour to suit the ability level of each type of user. In addition, the device supports 2 walking training modes: manual mode, which allows users to increase or decrease the speed as desired from the screen, and automatic mode, which allows users to choose to train at a specified distance of 15 cm, 20 cm, and 25 cm for maximum user safety. The device is also designed to be able to stop the machine immediately if a patient falls, with an audible alert system. For the purpose of monitoring the rehabilitation results, the system can record walking distance data for up to 5 patients.

(Total 110 pages)

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จสมบูรณ์ได้ด้วยความอนุเคราะห์และการสนับสนุนจาก ผู้ช่วยศาสตราจารย์ ชาญณรงค์ พงศ์รักรธรรม ซึ่งเป็นอาจารย์ที่ปรึกษาที่กรุณาให้คำปรึกษาข้อชี้แนะอันมีคุณค่าและก่อให้เกิดประโยชน์ในการดำเนินโครงการ ตลอดจนแนะแนวทางการแก้ปัญหาต่าง ๆ ในการดำเนินโครงการอย่างดียิ่ง อีกทั้งความเมตตาในการช่วยเหลือด้านต่างๆ ตลอดโครงการนี้ ผู้จัดทำปริญญานิพนธ์ขอกราบขอบพระคุณอาจารย์ด้วยความซาบซึ้งใจและความเคารพอย่างสูง

ผู้จัดทำปริญญานิพนธ์ขอขอบพระคุณคณาจารย์ทุกท่านที่ได้กรุณาให้ความรู้ และคำปรึกษาแนะนำด้านการออกแบบโครงการชุดนี้ให้ปริญญานิพนธ์มีความเรียบร้อยสมบูรณ์

ประโยชน์อันใดที่เกิดจากปริญญานิพนธ์เล่มนี้ผู้จัดทำโครงการ ขออุทิศความดีและกุศลนั้นแด่ผู้ที่กล่าวไว้ข้างต้น จึ่งใคร่ขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ธีรพงษ์ รอดทองอ่อน

ฟาฮัด ภูแห่ม

สารบัญ

	หน้า
บทคัดย่อ	ค
Abstract	ง
กิตติกรรมประกาศ	จ
สารบัญ	ฉ
สารบัญตาราง	ช
สารบัญภาพ	ณ
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ	3
1.4 ขอบเขตของโครงการ	3
1.5 ขั้นตอนและวิธีการดำเนินโครงการ	3
1.6 แผนการดำเนินงาน	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้องและอุปกรณ์ที่ใช้ในโครงการ	5
2.1 ปัญหาและความสำคัญ	5
2.2 การทบทวนวรรณกรรมที่เกี่ยวข้อง	5
2.3 อุปกรณ์ที่ใช้ในโครงการ	14
บทที่ 3 วิธีการดำเนินงาน	23
3.1 การศึกษาอุปกรณ์ลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุที่นำมาพัฒนา	23
3.2 การสร้างอุปกรณ์ลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ	26
3.3 วงจรการทำงานของลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ	31
3.4 ขั้นตอนการประกอบรายละเอียดชิ้นงาน	32
บทที่ 4 การทดลองและวิจารณ์	44
4.1 วิธีทดสอบ	44
4.2 ผลการทดลอง	44
4.3 สรุปและวิเคราะห์ผลการทดลอง	52

สารบัญ(ต่อ)

	หน้า
บทที่ 5 สรุปผลและข้อเสนอแนะ	53
5.1 สรุปผลที่ได้จากการทำโครงการ	53
5.2 ปัญหาและอุปสรรค	54
5.3 ข้อควรระวังและข้อเสนอแนะ	54
5.4 แนวทางในการพัฒนา	54
เอกสารอ้างอิง	55
ภาคผนวก ก	57
ภาคผนวก ข	59
ภาคผนวก ค	64
ประวัติผู้จัดทำปริญญานิพนธ์	108

สารบัญตาราง

ตารางที่		หน้า
1-1	แผนดำเนินงาน	4
3-1	อุปกรณ์ที่ใช้ในการสั่งการควบคุมและต้นกำลัง	42
4-1	การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ บุคคลที่ 1	44
4-2	การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ บุคคลที่ 2	45
4-3	การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ บุคคลที่ 3	46
4-4	การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ บุคคลที่ 4	47
4-5	การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ บุคคลที่ 5	48
4-6	การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 1	49
4-7	การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 2	49
4-8	การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 3	50
4-9	การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 4	51
4-10	การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 5	51
ก-1	งบประมาณของการสร้างพัฒนาลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ	58

สารบัญภาพ

ภาพที่		หน้า
1-1	อัตราตายของผู้ป่วยโรคหลอดเลือดสมอง (Stroke : 160 – 169)	1
2-1	กำหนดจุดอ้างอิงในการวัด	6
2-2	เครื่องช่วยเดินอัตโนมัติ นายจรูญ คนแรง คณะเทคโนโลยีอุตสาหกรรม	8
2-3	เครื่องช่วยเดินต้านแรงโน้มถ่วงแบบอัตโนมัติ	9
2-4	โปรเจคเครื่องช่วยหัดเดินอัตโนมัติสำหรับผู้ป่วย	10
2-5	งานวิจัยรถเข็นช่วยเดินชนิดเคลื่อนที่ได้รอบทิศทาง	11
2-6	โลโก้โปรแกรม Arduino IDE	15
2-7	ภาษาซี (C Programming Language)	15
2-8	ESP32 NodeMCU ESP-WROOM-32 Wi-Fi and Bluetooth	16
2-9	LCD 128x64 Dots LCD Graphic Matrix Display	16
2-10	LCD IIC/I2C Interface	17
2-11	IR Sensor Module	18
2-12	Switching Power Supply AC to DC	18
2-13	DC-to-DC Step Down Module	19
2-14	Micro SD Card Module	19
2-15	Relay module Arduino 5V	20
2-16	Micro switch tact switch button switch	21
2-17	Transistor BC547	22
3-1	ลู่วิ่งรุ่น Kee Going Max	23
3-2	หน้าจอลู่วิ่งที่นำมาพัฒนา	24
3-3	มอเตอร์ที่ใช้ในการขับเคลื่อนสายพานเป็นแบบกระแสตรง	25
3-4	บอร์ดที่ใช้ควบคุมการทำงานของมอเตอร์กระแสตรง	25
3-5	การออกแบบโครงสร้างของลู่วิ่งสำหรับผู้ป่วยและสูงอายุ	26
3-6	ออกแบบวงจรใหม่ใช้ในการคอลโทรลคำสั่ง Input	27
3-7	ออกแบบวงจรใหม่ใช้ในส่วนเมนูบนจอควบคุม	27

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
3-8	ออกแบบวงจรใหม่ใช้ในส่งสัญญาณไปแสดงยังหน้าจอ LCD	28
3-9	ออกแบบวงจรใหม่โดยแปลงแรงดันจาก Ac to Dc และ stepdown	28
3-10	วงจร Push button switch	29
3-11	วงจร Save Data	30
3-12	วงจรการทำงานของลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ	31
3-13	วงจรโมดูลปุ่มกดป้อนค่า Input	32
3-14	วงจรของแผงวงจรหลักและตำแหน่งอุปกรณ์	32
3-15	วงจรโมดูลปุ่มกดป้อนค่า Input ที่ลอกลายเสร็จ	33
3-16	วงจรโมดูลปุ่มกดป้อนค่า Input ที่ลงอุปกรณ์เสร็จ	33
3-17	วงจรโมดูลหลักของเครื่อง ที่ลอกลายเสร็จ	34
3-18	วงจรโมดูลหลักของเครื่อง ที่ลงอุปกรณ์เสร็จ	34
3-19	ติดตั้งหน้าจอแสดงผลและชุดคำสั่ง Input	35
3-20	ติดตั้งวงจรชุดคำสั่งหลักเข้ากับกล่อง Box electric	35
3-21	ติดตั้งชุดบอร์ดคำสั่งของเครื่องเดิมเข้ากับบอร์ดที่สร้างขึ้นใหม่	36
3-22	ติดตั้ง IR sensor1 และ IR sensor2	36
3-23	เครื่องลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุที่เสร็จสมบูรณ์	37
3-24	การทำงานของอุปกรณ์แบบ Flow Chart	38
3-25	ไดอะแกรมการต่อใช้งานลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ	40
ข-1	เปิดสวิตซ์เครื่อง	60
ข-2	ทำการติด Safety switch	60
ข-3	ทำการป้อนรหัสเข้าใช้งาน	61
ข-4	กดโหมด Auto	61
ข-5	โหมด Auto เลือกระยะใช้งาน	62
ข-6	การเริ่ม-หยุดเครื่อง โหมด Auto	62
ข-7	กดโหมด Manal	63
ข-8	การเริ่ม-หยุดเครื่องโหมด Manal	63

บทที่ 1

บทนำ

ในบทนี้จะกล่าวถึงความสำคัญและที่มาของโครงการ วัตถุประสงค์ของโครงการ ประโยชน์ที่คาดว่าจะได้รับ ขอบเขตของโครงการ ขั้นตอนและวิธีการดำเนินงานของโครงการ แผนการดำเนินงานดังต่อไปนี้

1.1 ความสำคัญและที่มาของโครงการ

เนื่องจากสภาพแวดล้อมในปัจจุบันมีจำนวนประชากรในกลุ่มผู้สูงอายุและผู้ป่วยทางการเคลื่อนไหวเพิ่มขึ้นอย่างต่อเนื่อง และยังมีแนวโน้มมากขึ้น ซึ่งเกิดจากหลายสาเหตุด้วยกัน ไม่ว่าจะเป็นเกิดจากการเสื่อมสภาพของร่างกายจากการที่มีอายุมากขึ้น จากอุบัติเหตุที่ทำให้เกิดการสูญเสียความสามารถในการเคลื่อนไหว รวมถึงผลพวงที่มาจากโรคร้ายต่างๆ ซึ่งเครื่องช่วยเดินแบบลู่วิ่งก็ยังเป็นอุปกรณ์ทางการแพทย์ที่ใช้ในการฟื้นฟูสภาพร่างกายของผู้ที่มีปัญหาเหล่านั้น เช่น ผู้สูงอายุ, ผู้ป่วยในระยะพักฟื้น, ผู้ป่วยกล้ามเนื้ออ่อนแรง, ผู้ป่วยโรคหลอดเลือดสมอง [1] เป็นต้น

24.1. อัตราตายของผู้ป่วยโรคหลอดเลือดสมอง (Stroke : I60-I69)

ตัวแปร
 A หมายถึง จำนวนครั้งของการสำนายสถานะตายของผู้ป่วยโรคหลอดเลือดสมอง (Stroke : I60-I69) จากทุกหอผู้ป่วย
 B หมายถึง จำนวนครั้งของการสำนายผู้ป่วยโรคหลอดเลือดสมองจากทุกหอผู้ป่วยในช่วงเวลาเดียวกัน (Stroke : I60-I69)
 สูตรคำนวณ (A/B)X100

Copy CSV Print Excel

โรงพยาบาล	รวมทั้งปีงบประมาณ		
	I60-I69		
	A	B	ร้อยละ
รวม *	512	6822	7.51
11400-โรงพยาบาลเลิดสิน	90	650	9.47
11470-โรงพยาบาลนพรัตนราชธานี	143	1211	11.81
11471-โรงพยาบาลสงฆ์	7	317	2.21
11472-โรงพยาบาลราชวิถี	211	1813	11.64
11473-โรงพยาบาลอปรราชภัฏ (วัดโสม)	0	12	0.00
12250-สถาบันพระปกเกล้า	1	1	100.00
12256-สถาบันวิจัยและพัฒนาสุขภาพภาคเหนือ	4	58	6.90
12258-สถาบันวิจัยและพัฒนาสุขภาพภาคเหนือ	0	16	0.00
12281-โรงพยาบาลพระนครศรีอยุธยา	31	1382	2.24
12283-โรงพยาบาลระยอง	0	5	0.00
12438-สถาบันสุขภาพเด็กแห่งชาติมหาราชินี	0	33	0.00
12439-สถาบันประสาทวิทยา	24	693	2.42
12442-โรงพยาบาลสมเด็จพระสังฆราชญาณสังวรเพื่อผู้สูงอายุ จ.ชลบุรี	0	17	0.00
14100-โรงพยาบาลระยอง	0	2	0.00
14201-โรงพยาบาลระยอง	1	1	100.00
	0	11	0.00

ภาพที่ 1-1 อัตราตายของผู้ป่วยโรคหลอดเลือดสมอง (Stroke : I60 – I69) [1]

โดยงานวิจัยชิ้นนี้จะมุ่งเน้นเพื่อช่วยเหลือผู้ที่มีปัญหาการเคลื่อนไหวด้านการเดินที่มีความสามารถเดินที่ดีในระดับหนึ่ง ซึ่งหมายถึง ผู้ป่วยพอที่จะช่วยเหลือตัวเอง หรือก้าวขาได้ด้วยตัวเองได้โดยอาจจะต้องการผู้ช่วยพยุงเดิน 1 คนตลอดเวลาหรือบางครั้ง (Functional Ambulatory Category; FAC ระดับ 2-4) เช่น ผู้สูงอายุ ผู้ป่วยในระยะพักฟื้น ผู้สูงอายุ, ผู้ป่วยในระยะพักฟื้น, ผู้ป่วยกล้ามเนื้ออ่อนแรง, ผู้ป่วยโรคหลอดเลือดสมอง ที่มีปัญหาด้านการเดินในรูปแบบต่าง ๆ ที่ต้องได้รับการฟื้นฟูสภาพโดยการเริ่มหัดเดินใหม่ด้วยตนเองหรือมีอุปกรณ์บางชนิดช่วยในการเดินในกรณีผู้ป่วยไม่มีกำลังแขนหรือผู้ป่วยสูงอายุที่ไม่สามารถเดินตามปกติ

จากข้อมูลนพ.ศักรินทร์ วงศ์เลิศศิริ ผอ.สถาบันสิรินธรเพื่อการฟื้นฟูสมรรถภาพทางการแพทย์แห่งชาติ (สำนักงานกองทุนสนับสนุนการส่งเสริมสุขภาพ, 2562) [2] สาเหตุของการเดินที่ผิดปกติสามารถแบ่งได้ดังนี้ โครงสร้างมีความผิดปกติ เช่น ข้อติด ข้อหรือกระดูกผิดรูป กล้ามเนื้ออ่อนแรง ระบบประสาทรับความรู้สึกผิดปกติ การมองเห็นผิดปกติหรือตาบอด ระบบประสาทส่วนกลางผิดปกติ การบำบัดรักษาฟื้นฟู คือ ทำสิ่งที่เป็นอยู่ให้ดีขึ้น ถ้าสิ่งที่เกิดขึ้นไม่สามารถแก้ไขได้มีความพิการถาวรควรใช้อุปกรณ์ช่วยสามารถทรงตัวอยู่ได้ในช่วงที่เท้าข้างเดียวมีการสัมผัสพื้น ขาอีกข้างจะต้องเหวี่ยงไปยังตำแหน่งที่เหมาะสมเพื่อทำหน้าที่รับน้ำหนักต่อไป และต้องมีกำลังเพียงพอในการเคลื่อนไหวและลำตัวไปข้างหน้า หากพบว่าการเดินที่ผิดปกติจะเป็นอาการแสดงอย่างหนึ่งของภาวะหรือโรคที่เกี่ยวข้องกับระบบกระดูก ข้อ กล้ามเนื้อ และโรคทางระบบประสาท ดังนั้นในการฟื้นฟูผู้ป่วยที่มีปัญหาการเดินที่ผิดปกติ แพทย์จะวินิจฉัยเพื่อหาแนวทางรักษาและฟื้นฟูอย่างเหมาะสมต่อไป

ดังนั้น ทางผู้จัดทำโครงการจึงมีแนวคิดที่จะประดิษฐ์อุปกรณ์ลู่วงสำหรับผู้ป่วยและผู้สูงอายุจากการนำข้อดีและข้อเสียของอุปกรณ์ช่วยหัดเดินแต่ละชนิดที่มีอยู่ในปัจจุบัน จึงนำมาพัฒนาต่อเพื่อให้ตอบสนองในระบบการทำงานของอุปกรณ์ให้มีประสิทธิภาพสูงสุด และเกิดประโยชน์มากที่สุดสำหรับผู้ป่วยที่ต้องได้รับการฟื้นฟูสภาพร่างกาย

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาเกี่ยวกับการพัฒนาอุปกรณ์ไฟฟ้าและอิเล็กทรอนิกส์
2. เพื่อส่งเสริมให้เกิดแนวคิดสร้างสรรค์และนวัตกรรมในการออกแบบ ตลอดจนการนำวัสดุมาประยุกต์ใช้ได้อย่างมีประสิทธิภาพ
3. เพื่อศึกษาและสร้างนวัตกรรมที่สามารถช่วยเหลือสังคมได้

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ความรู้เกี่ยวกับอุปกรณ์ลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ
2. ช่วยเพิ่มความมั่นคงในการเคลื่อนไหว
3. เป็นอุปกรณ์ช่วยเดินสำหรับผู้ป่วยอัมพฤกษ์ อัมพาต และผู้สูงอายุ
4. สะดวก เหมาะสม กะทัดรัด สวยงาม เคลื่อนย้ายสะดวก
5. สามารถพัฒนาเข้าสู่ระบบพาณิชย์ได้
6. สามารถนำความรู้ที่ได้มาประดิษฐ์ช่วยเหลือสังคม

1.4 ขอบเขตของโครงการ

1. สร้างเครื่องช่วยหัดเดินอัตโนมัติพร้อมจอแสดงผลสำหรับผู้ป่วยและผู้สูงอายุจำนวน 1 เครื่อง
2. สามารถปรับความเร็วในการช่วยเดิน ได้สูงสุด 5 กิโลเมตร/ชั่วโมง
3. สามารถเลือกการฝึกโหมดการเดิน 2 โหมด คือโหมดแมนนวลและโหมดอัตโนมัติ
4. สามารถส่งเครื่องหยุดทำงานทันทีหากผู้ป่วยล้ม และแจ้งเตือนด้วยเสียง
5. สามารถบันทึกระยะทางการเดินของผู้ป่วยจำนวน 5 คน

1.5 ขั้นตอนและวิธีการดำเนินของโครงการ

1. กำหนดหัวข้อ
2. ค้นหาข้อมูลที่เกี่ยวข้องและทบทวนวรรณกรรม
3. การจัดทำโครงการเพื่อนำเสนอและวางแผนส่วนที่เกี่ยวข้อง
4. ศึกษาอุปกรณ์ที่นำมาทำโครงการและอุปกรณ์ที่นำมาควบคุมการทำงาน
5. จัดทำโครงสร้างอุปกรณ์ช่วยหัดเดินสำหรับผู้ป่วย เขียนโปรแกรมการทำงาน
6. ทดสอบการทำงานและแก้ไขส่วนที่ผิดพลาด
7. ประเมินผลที่ได้และสรุปผล

1.6 แผนการดำเนินงาน

ตารางที่ 1-1 แผนการดำเนินงาน

ระยะเวลาดำเนินงาน	ก.ย. 66	ต.ค. 66	พ.ย. 66	ธ.ค. 66	ม.ค. 67	ก.พ. 67	มี.ค. 67
1. กำหนดหัวข้อ รวบรวมข้อมูล ทบทวนวรรณกรรม							
2. การจัดทำโครงงานเพื่อนำเสนอและวางแผนส่วนที่เกี่ยวข้อง							
3. ศึกษาอุปกรณ์ที่นำมาใช้งาน							
4. จัดทำโครงสร้างอุปกรณ์และเขียนโปรแกรมการทำงาน							
5. สอบก้าวหน้าโครงการ							
6. ทดสอบการทำงานและแก้ไขส่วนที่ผิดพลาด							
7. ประเมินผลที่ได้และสรุปผล							
8. จัดทำรูปเล่มโครงงาน							
9. สอบป้องกันโครงงาน							

บทที่ 2

ทฤษฎีที่เกี่ยวข้องและอุปกรณ์ที่ใช้ในโครงการ

ในบทนี้จะกล่าวถึงปัญหาและความสำคัญ การทบทวนวรรณกรรมที่เกี่ยวข้อง การเคลื่อนที่ในแนวตรง อุปกรณ์ที่ใช้ในโครงการ ดังต่อไปนี้

2.1 ปัญหาและความสำคัญ

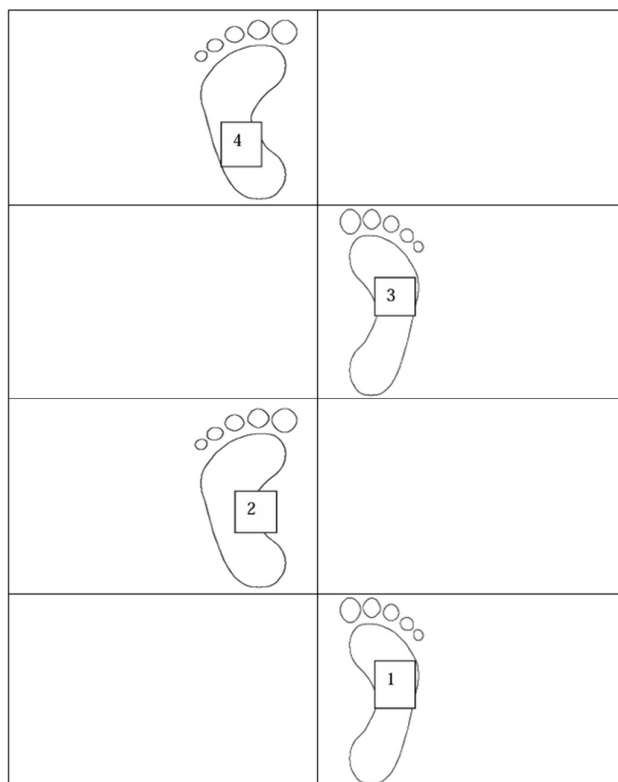
ปัจจุบันมีผู้ป่วยทางการเคลื่อนไหวเป็นจำนวนมาก และยังมีแนวโน้มเพิ่มมากขึ้นซึ่งเกิดจากหลายสาเหตุด้วยกัน ไม่ว่าจะเป็นเกิดจากการเสื่อมสภาพของร่างกายจากการที่มีอายุมากขึ้นจากอุบัติเหตุที่ทำให้สูญเสียความสามารถในการเคลื่อนไหว รวมถึงผลพวงที่มาจากโรคภัยต่าง ๆ เช่น ผู้สูงอายุ, ผู้ป่วยในระยะพักฟื้น, ผู้ป่วยกล้ามเนื้ออ่อนแรง, ผู้ป่วยโรคหลอดเลือดสมอง ก็จะมีภาวะกล้ามเนื้ออ่อนแรง โดยในการทำกายภาพบำบัดผู้ป่วยต้องเดินให้มากที่สุดเพื่อให้กล้ามเนื้อแข็งแรง แต่ปัญหาที่เกิดขึ้นคือการใช้งานของอุปกรณ์ โดยงานวิจัยที่เกี่ยวข้องนี้ จะศึกษาผลของระบบ ในกรณีต่าง ๆ จากอุปกรณ์ใกล้เคียงที่มีอยู่ในท้องตลาด และเทคนิคและอุปกรณ์ที่เกี่ยวข้อง

2.2 การทบทวนวรรณกรรมที่เกี่ยวข้อง

2.2.1 งานวิจัยการประมาณความสูงจากความยาวของการก้าวขณะเดิน Estimation of Stature from Stride Length while Walking [3]

งานวิจัยนี้มีวัตถุประสงค์ในการประมาณความสูงของบุคคลจากความยาวของการก้าวเดิน โดยการหาความสัมพันธ์ของระยะการก้าวเดินกับความสูงของบุคคล รวมทั้งศึกษาวิธีการในการวัดความยาว ของการก้าวเดิน เพื่อใช้ประโยชน์ในการทำนายความสูงของบุคคล ทำการทดลองโดยการเก็บตัวอย่าง รอยพื้นรองเท้าซึ่งได้จากการก้าวเดินในท่าเดินปกติเป็นระยะทาง 1.00 เมตร แล้วทำการวัดระยะห่างของ การก้าวเพื่อหาความสัมพันธ์ร่วมกับส่วนสูง ด้วยการวิเคราะห์ผลทางสถิติ ซึ่งกลุ่มตัวอย่างที่ใช้ในการ ทดลองคืออาสาสมัครเพศชายจำนวน 100 คน ผลการวิจัยพบว่า ความสัมพันธ์ระหว่างระยะการก้าวเดินและความสูง ทำให้ได้สมการถดถอย คือ $\text{ความสูง} = 155.720 + .212(\text{ระยะก้าว})$ สามารถอธิบายความสูงของบุคคลได้ 25.1% เมื่อนำผล ที่ได้มาทำการวิเคราะห์ความแปรปรวนโดยใช้สถิติ F-test ค่าสถิติ F เท่ากับ 32.763 พบว่าความสูงมี ความสัมพันธ์เชิงเส้นกับระยะก้าวด้วยความเชื่อมั่น 95% ผู้วิจัยได้ทำการทดสอบสมการที่ได้โดยการแทน ค่าระยะการก้าวเดินของบุคคลจากการทดลองเพื่อคำนวณหาค่าความสูง จำนวน 100 คน ผลการทดลอง พบว่า

ค่าที่ได้จากการแทนค่าในสมการมีค่าเฉลี่ยของผลต่างเท่ากับ 1.46% ทำให้ระบุได้ว่าวิธีการดังกล่าวมีความน่าเชื่อถือได้ ดังภาพที่ 2-1



ภาพที่ 2-1 กำหนดจุดอ้างอิงในการวัด [3]

ความยาวของระยะการก้าวเดินของอาสาสมัครจำนวน 100 คน ที่ทำการเก็บข้อมูลมาจากการกำหนดให้อาสาสมัครทำการก้าวเดินในท่าเดินปกติ เป็นระยะทาง 1.00 เมตร ผู้วิจัยได้นำข้อมูลระยะการก้าวเดินที่ได้จากการทดลองมาทำการวิเคราะห์ข้อมูลโดยใช้สถิติเข้ามาช่วยในการหาความสัมพันธ์ระหว่างระยะการก้าวเดินกับความสูงของบุคคลจากรอยพิมพ์พื้นรองเท้าเพื่อใช้ประโยชน์ในการประมาณ ความสูงของบุคคลจากระยะการก้าวเดินในท่าเดินปกติ นำไปสู่การหาวิธีการใหม่ๆ ในการพิสูจน์เอกลักษณ์บุคคลเชื่อมโยงถึงตัวบุคคลผู้กระทำความผิด ซึ่งผลจากการวิเคราะห์ผลข้อมูลแสดงผลการ วิเคราะห์การถดถอย (Regression Analysis) ของข้อมูลเพื่อหาความสัมพันธ์ระหว่างระยะการก้าวเดิน และความสูง ทำให้ได้ค่าสัมประสิทธิ์สหสัมพันธ์ (Correlation Coefficient : r) มีค่า .501 ซึ่งมีค่าเป็น บวก แสดงว่า ระยะการก้าวเดินแปรผันตรงกับส่วนสูงของบุคคล นั่นคือ

ถ้าพบว่าระยะการก้าวเดินเพิ่มมากขึ้น แสดงว่าส่วนสูงของบุคคลก็จะเพิ่มขึ้น และค่า Sig. เท่ากับ .000 มีค่าน้อยกว่า 0.05 แสดงว่า ระยะการก้าวเดินและความสูงมีความสัมพันธ์กันอย่างมีนัยสำคัญที่ระดับ 0.05 และจากค่าความสัมพันธ์ที่ได้นี้ ผู้วิจัยนำไปสร้างสมการถดถอยอย่างง่ายได้เป็นความสูง = $155.720 + .212$ (ระยะก้าว) จากสมการดังกล่าวสามารถระบุได้ว่า ถ้าระยะการก้าวเดินเพิ่มขึ้น 10 เซนติเมตร สามารถพยากรณ์ความสูง 783 Veridian E-Journal, SU Vol.5 No. 2 May - August 2012 กลุ่มวิทยาศาสตร์ของคนที่เดินได้ เพิ่มขึ้น 21.2 เซนติเมตร และเมื่อพิจารณาค่าคงที่ (b0) และค่าสัมประสิทธิ์ของ ระยะ ก้าว (b1) พบว่า ทั้ง 2 ค่านี้ มีค่า Sig. เท่ากับ .000 มีค่าน้อยกว่า 0.05 แสดงว่า ความสัมพันธ์ระหว่าง ระยะการก้าวเดินและความสูงมีความสัมพันธ์เชิงเส้นในทางบวก ที่ระดับนัยสำคัญ 0.05 หรือความชันของ สมการถดถอยนี้มีค่าเป็นบวก ซึ่งจากผลการวิเคราะห์ข้างต้น สมการการถดถอยมีค่า R² จากสมการที่ได้มีค่าเท่ากับ .251 แสดงว่า สมการถดถอยที่ได้สามารถอธิบายความสูงของบุคคลได้ 25.1% เมื่อมา ทดสอบ ทดสอบการวิเคราะห์ความแปรปรวนโดยใช้สถิติ F-test ค่าสถิติ F เท่ากับ 32.763 ซึ่งมีค่า Sig. เท่ากับ .000 มีค่าน้อยกว่า 0.05 แสดงว่าระยะการก้าวเดินของอาสาสมัครสามารถอธิบายความผันแปรของความ สูงได้ หรือความสูงมีความสัมพันธ์เชิงเส้นกับระยะก้าวด้วยความเชื่อมั่น 95% รากที่สองของ ค่าสถิติ F เท่ากับ 5.724 ซึ่งเท่ากับค่าสถิติ t สำหรับทดสอบความชันของสมการถดถอย ดังนั้นสามารถใช้สถิติ F ทดสอบความชันหรือค่าสัมประสิทธิ์ของ ระยะก้าว (b1) ได้ด้วย ผู้วิจัยได้ทำการทดสอบสมการที่ได้โดยการ แทนค่าระยะการก้าวเดินของบุคคลจากการทดลองเพื่อคำนวณหาความสูง จำนวน 100 คน ผลการ ทดลองพบว่าค่าความสูงที่จากการทดสอบมีค่าความแตกต่างกับค่าความสูงจริงของอาสาสมัครพบว่า ค่าที่ได้จากการแทนค่าในสมการมีค่าเฉลี่ยของผลต่างเท่ากับ 1.46% ทำให้ระบุได้ว่าวิธีการดังกล่าวมีความน่าเชื่อถือได้

2.2.2 ในปีพ.ศ. 2559 นายจรัญ คนแรง คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏเชียงราย ได้นำเสนองานวิจัยเครื่องช่วยเดิน (Walker) [4] เป็นอุปกรณ์สำหรับผู้ป่วยที่มีปัญหาเกี่ยวกับการเคลื่อนที่ เช่น ผู้ป่วยเกี่ยวกับกระดูกเท้า กระดูกขา ผู้ป่วยกล้ามเนื้อแขนขาอ่อนแรง และผู้สูงอายุ เพื่อให้ผู้ป่วยเคลื่อนที่ได้ด้วยตนเองอย่างเป็นปกติ แต่ทั้งนี้ถ้าผู้ป่วยไม่มีกำลังแขนและกำลังขาที่ยืนพยุงตัวเองเพียงลำพังได้ ก็ไม่สามารถที่จะยกเครื่องช่วยเดินได้ ทำให้ผู้ที่ดูแลผู้ป่วยจะต้องคอยประคองผู้ป่วยอยู่ตลอดเวลา งานวิจัยได้พัฒนาคิดค้น เครื่องช่วยเดินอัตโนมัติ สำหรับผู้ป่วยหัดเดินและคนชรา โดยการติดตั้งที่ขาของเครื่องช่วยเดินและขับเคลื่อนด้วยมอเตอร์ไฟฟ้ากระแสตรง พร้อมทั้งติดตั้งเซ็นเซอร์ที่อาศัยการก้าวขาของคนป่วยหัดเดินหรือคนชรามาเป็นตัวควบคุมให้เครื่องช่วยเดินอัตโนมัติเคลื่อนที่แทนการยก อุปกรณ์มีประสิทธิภาพ สะดวก ปลอดภัย และเป็นที่ยอมรับ ช่วยให้ผู้ป่วยหรือคนชราเคลื่อนที่ได้สะดวกมากขึ้น ดังภาพที่ 2-2



ภาพที่ 2-2 เครื่องช่วยเดินอัตโนมัติ นายจรูญ คนแรง คณะเทคโนโลยีอุตสาหกรรม [4]

2.2.3 ในปีพ.ศ. 2558 นายพงศ์พิสุทธิ์ จอมวุฒิ และ นายสุทธิพงษ์ ชาชงโดยตีพิมพ์ในวารสารของ ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ได้นำเสนองานวิจัยเครื่องช่วยหัดเดินสำหรับผู้ป่วย (The Patient Assistant Walker) [5] งานวิจัยนี้ทำขึ้นเพื่อคิดค้นและทำการออกแบบสร้างเครื่องช่วยหัดเดินสำหรับผู้สูงอายุที่ไม่สามารถเดินเองได้สะดวกและผู้ป่วยที่ต้องการการรักษาทางกายภาพบำบัดเกี่ยวกับการเดินโดยเครื่องช่วยหัดเดินจะควบคุมด้วยไมโครคอนโทรลเลอร์ หลักการทำงานคือ เครื่องช่วยหัดเดินสามารถปรับความเร็วได้ 3 ระดับ แสดงเวลาในการทำงานของเครื่องในแต่ละครั้งสามารถวัดระยะทางในการเดินแต่ละครั้งมีปุ่มควบคุมทิศทางตรงแขนจับทั้งสองข้าง ข้อดี ช่วยในการอำนวยความสะดวกในการทำกายภาพบำบัดได้ดี ข้อเสีย ใช้งานได้เฉพาะพื้นที่ที่ผิวเรียบใช้ในพื้นที่ผิวขรุขระไม่ได้

2.2.4 ในปีพ.ศ. 2558 พันโท นพ.บุระ สันธูภากร สำนักวิชาแพทยศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ได้นำเสนองานวิจัยเครื่องช่วยเดินด้านแรงโน้มถ่วงแบบอัตโนมัติ [6] งานวิจัยนี้ทำขึ้นเพื่อที่ไม่สามารถเดินหรือขยับตัวได้อย่างคล่องแคล่วจะต้องมีจำนวนสูงมากในอนาคตอันใกล้นี้ ร่างกายช่วงล่างเป็นปัจจัยสำคัญของการเคลื่อนไหว การขยับขาทั้งสองข้างไม่ได้จากภาวะอัมพาตหรือกล้ามเนื้ออ่อนแรงที่เกิดจากการบาดเจ็บที่กระดูกสันหลังมีความรุนแรงที่แตกต่างกันไปในแต่ละบุคคล แต่ต้องใช้เวลาในการฟื้นฟูกล้ามเนื้อและหมั้นขยับข้อทุกส่วนเพื่อป้องกันการติดขัด บางคนสามารถฝึกนั่งได้บ้างเพื่อฟื้นฟูกล้ามเนื้อบางส่วนให้มีการขยับได้บ้าง แม้ผู้ป่วยกลุ่มนี้ยังจำเป็นต้องมีผู้ช่วยเหลือในการฟื้นฟูกล้ามเนื้ออยู่ตลอด แต่สมองของผู้ป่วยกลุ่มนี้สามารถคิดและสั่งการได้อย่างปกติ หากสมองนึกคิดที่จะออกกำลังกาย ก็จำเป็นต้องมีเครื่องออกกำลังกายที่เหมาะสมกับความพิการนี้ เพื่อให้พึ่งผู้ช่วยเหลือน้อย

สำหรับหลักการทำงานของอุปกรณ์นี้ คือ การใช้แรงดันลมจากชุดควบคุมแรงดันลม ซึ่งประกอบไปด้วยมอเตอร์และปั๊มอัดอากาศ แล้วอัดอากาศเข้าไปยังถุงลม เมื่ออากาศถูกอัดเข้ามาในถุงลมเป็นปริมาณมากจะทำให้เกิดแรงยกขึ้น เพื่อช่วยลดแรงเนื่องจากน้ำหนักของผู้ทำกายภาพบำบัดที่จะกระทำต่ออวัยวะส่วนล่างของร่างกาย (เช่น เข่า และข้อเท้า) เพราะว่าแรงยกที่เกิดขึ้นนั้นจะทำให้ทำกายภาพบำบัดอยู่ในสภาวะเหมือนไร้น้ำหนัก ทำให้เกิดความปลอดภัยในการกายภาพบำบัด และลดความเสี่ยงเนื่องจากแรงกระทำที่กระทำกับอวัยวะส่วนล่างที่จะเกิดขึ้นขณะทำการกายภาพบำบัด ช่วยให้เกิดความปลอดภัยมากยิ่งขึ้น ดังภาพที่ 2-3



ภาพที่ 2-3 เครื่องช่วยเดินต้านแรงโน้มถ่วงแบบอัตโนมัติ [6]

2.2.5 ในปีพ.ศ. 2553 นายสาโรจน์ แนวบรรทัด และคณะ สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัยได้นำเสนอโปรเจกต์อุปกรณ์ช่วยหัดเดินอัตโนมัติสำหรับผู้ป่วย งานวิศวกรรมซ่อมบำรุง โรงพยาบาลสงขลานครินทร์ [7] โปรเจกต์นี้ทำขึ้นเพื่อการสร้างอุปกรณ์เพื่ออำนวยความสะดวกในการเดินให้กับผู้ป่วย เป็นเครื่องมือทางกายภาพบำบัดเพื่อใช้ในการช่วยให้ผู้ป่วยที่มีปัญหากับขาหรือกระดูกสันหลังได้หัดเดิน ผู้ป่วยสามารถใช้เครื่องมือดังกล่าวโดยไม่ต้องมีผู้อื่นช่วย อุปกรณ์ดังกล่าวนี้ใช้พลังงานจากแบตเตอรี่ ซึ่งมีอายุการใช้งานได้นานถึงสามชั่วโมงต่อการชาร์จหนึ่งครั้ง สามารถปรับระดับตามความสูงของผู้ป่วย และใช้ระบบไมโครคอนโทรลเลอร์ควบคุมการทำงาน มีตัวอินฟาเรดเซ็นเซอร์ควบคุมการเคลื่อนไหวในแต่ละก้าว ซึ่งทันทีที่มีการเคลื่อนไหวผ่านเซ็นเซอร์สัญญาณจะถูกส่งไปยังไมโครคอนโทรลเลอร์และตัวมอเตอร์ที่ติดอยู่ตรงล้อจะทำงาน ช่วงเวลาทำงานของเครื่องจะสั้น และสามารถตั้งได้สี่ระดับ คือ 15 ซม, 20 ซม, 25 ซม

และ 30 ซม. การทำงานของอุปกรณ์ช่วยเดินสำหรับผู้ป่วย เมื่อมีก้าวทำผ่านตัวตรวจจับทั้งสองขาตัวตรวจจับจะส่งสัญญาณไปยังไมโครคอนโทรลเลอร์ ประมวลผลการทำงาน จึงส่งสัญญาณไปยังวงจรขับเคลื่อนมอเตอร์ จึงทำให้ตัวอุปกรณ์เดินหน้าและแสดงการทำงานมายังหน้าจอLCD ดังภาพที่ 2-4

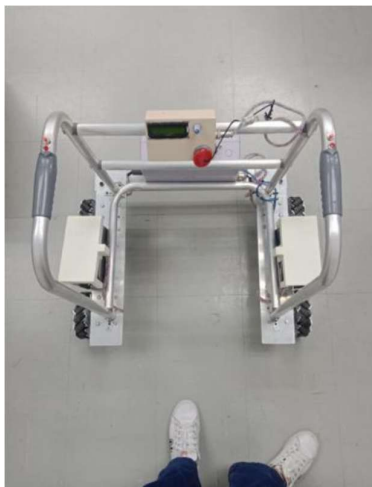


ภาพที่ 2-4 โปรเจคเครื่องช่วยหัดเดินอัตโนมัติสำหรับผู้ป่วย [7]

2.2.6 ในปีพ.ศ. 2561 ร.ต.พิชิตพล โชติกุลนันท์ และคณะวิจัยวิทยาลัยวิศวกรรมชีวการแพทย์ มหาวิทยาลัยรังสิต ได้นำเสนองานวิจัยรถเข็นช่วยเดินชนิดเคลื่อนที่ได้รอบทิศทาง [8]

งานวิจัยนี้ทำขึ้นเพื่อเป็นเครื่องมือที่ช่วยเหลือการเดินของผู้ที่มีปัญหาในการเดินจากอุบัติเหตุ ผู้ป่วย หรือผู้สูงอายุ ปัญหาที่มักพบในการใช้อุปกรณ์เหล่านี้โดยทั่วไปก็คือ การที่ต้องยกอุปกรณ์ช่วยเดินซึ่งทำให้ผู้ใช้เกิดอาการเหนื่อยล้าและไม่สะดวกในการใช้งาน นอกจากนี้ ระบบล้อทั่วไปซึ่งมักมีการเคลื่อนที่โดยไม่ได้ตั้งใจซึ่งอาจก่อให้เกิดอันตรายต่อผู้ที่ใช้งานได้

รถเข็นช่วยเดินชนิดเคลื่อนที่ได้รอบทิศทาง ได้รับการพัฒนาคิดค้นขึ้นเพื่อตอบโจทย์การใช้งานของผู้ที่ต้องการฝึกเดิน อาศัยการเคลื่อนที่ของล้อแมคคานัมที่สามารถเคลื่อนที่รอบทิศทาง โดยใช้รีโมตหรือปุ่มบังคับ ใช้งานง่าย มีความปลอดภัยสูง โดยระบบการทำงานจะถูกตัดโดยอัตโนมัติเมื่อเกิดเหตุฉุกเฉินหรือกรณีมีปัญหาการเคลื่อนที่ นอกจากนี้ยังสามารถตั้งโปรแกรมต่างๆ เช่น การกำหนดระยะการก้าวเดินได้ ผลิตจากวัสดุที่หาได้ง่ายในประเทศ ต้นทุนการผลิตต่ำ ผ่านการทดสอบด้านความพึงพอใจจากผู้ใช้แล้ว ดังภาพที่ 2-5



ภาพที่ 2-5 งานวิจัยรถเข็นช่วยเดินชนิดเคลื่อนที่ได้รอบทิศทาง [8]

2.3 การเคลื่อนที่ในแนวตรง [9]

นิยามการเคลื่อนที่

2.3.1 ระยะทาง (Distance)

จากการศึกษาการเคลื่อนที่ของวัตถุ เช่น รถยนต์ สัตว์ วัตถุตกในอากาศ พบว่าตำแหน่งของวัตถุมีการเปลี่ยนไปจากเดิม หรือกล่าวได้ว่าวัตถุจะเลื่อนจากตำแหน่งเดิมไปยังตำแหน่งใหม่ ซึ่งจะเรียกการเคลื่อนที่เช่นนี้ว่า การเคลื่อนที่แบบเลื่อนตำแหน่ง (translation motion) ถ้าเราทราบตำแหน่งเริ่มต้นเส้นทางการเคลื่อนที่และตำแหน่งสุดท้ายของการเคลื่อนที่ ก็จะได้ระยะทางจากความยาวตามเส้นทางการเคลื่อนที่นั้น

ระยะทางใช้สัญลักษณ์ “s” เป็นปริมาณสเกลาร์ คือมีแต่ขนาดเพียงอย่างเดียว แต่ไม่บอกทิศทางมีหน่วยเป็น เมตร (m)

- การกระจัด (Displacement)

เมื่อวัตถุมีการเคลื่อนที่จากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่ง การบอกตำแหน่งใหม่เทียบกับตำแหน่งเดิม เพื่อให้เข้าใจได้ชัดเจนต้องบอกทั้งระยะห่างและทิศทาง ปริมาณที่บอกให้ทราบถึงการเปลี่ยนตำแหน่ง เรียกว่า การกระจัด

การกระจัด ใช้สัญลักษณ์ “ s^v ” เป็นปริมาณเวกเตอร์ มีหน่วยเป็น เมตร (m)

การกระจัดหาได้จากเส้นตรง ที่เขียนหัวลูกศรกำกับโดยลากจากจุดเริ่มต้นไปยังจุดสุดท้ายของการเคลื่อนที่ ความยาวของเส้นตรงแทนขนาดของการกระจัดและทิศที่หัวลูกศรชี้จะแทนทิศของการกระจัด

2.3.2 อัตราเร็ว (Speed)

เมื่อวัตถุมีการเคลื่อนที่ ปริมาณที่บอกได้ว่าวัตถุนั้นเคลื่อนที่ได้เร็วมากหรือน้อยนั้น คือ อัตราเร็ว ซึ่งโดยทั่วไป อัตราเร็ว หมายถึง ระยะทางที่วัตถุเคลื่อนที่ได้ในหนึ่งเวลา หรือ อัตราการเปลี่ยนระยะทาง อัตราเร็วเป็นปริมาณสเกลาร์ ใช้สัญลักษณ์ “ v ” มีหน่วย เมตรต่อวินาที (m/s)

$$\text{อัตราเร็ว} = \frac{\text{ระยะทาง}}{\text{เวลา}} \quad \text{หรือ} \quad v = \frac{s}{t} \quad (2.1)$$

เมื่อ v คือ อัตราเร็วของวัตถุ มีหน่วย เมตรต่อวินาที (m / s)

s คือ ระยะทางของวัตถุ มีหน่วย เมตร (m)

t คือ เวลาที่วัตถุใช้ในการเคลื่อนที่ มีหน่วย วินาที (s)

อัตราเร็วสามารถแบ่งออกได้ ดังนี้

1. อัตราเร็วเฉลี่ย (V_{av}) คือ ระยะทางที่วัตถุเคลื่อนที่ได้ในหนึ่งหน่วยเวลา (ในช่วงเวลาหนึ่งที่กำลังพิจารณาเท่านั้น) สามารถเขียนเป็นสมการได้

$$V_{av} = \frac{\Delta s}{\Delta t} \quad (2.2)$$

เมื่อ

V_{av} คือ ความเร็วเฉลี่ย มีหน่วยเป็น เมตรต่อวินาที

Δs คือ ระยะที่วัตถุเคลื่อนที่ได้ มีหน่วยเป็น เมตร

Δt คือ ช่วงเวลาที่วัตถุใช้เคลื่อนที่ มีหน่วยเป็น วินาที

2. อัตราเร็วขณะใดขณะหนึ่ง (V_t) คือ ระยะทางที่เคลื่อนที่ได้ในหนึ่งหน่วยเวลา เมื่อช่วงเวลาที่เคลื่อนที่น้อยมากๆ สามารถเขียนสมการได้

$$V_t = \frac{\Delta s}{\Delta t} \quad (2.3)$$

เมื่อ V_t คือ อัตราเร็วขณะใดขณะหนึ่ง มีหน่วยเป็น เมตรต่อวินาที (m / s)

3. อัตราเร็วคงที่ (V) คือ การบอกให้ทราบว่าวัตถุมีการเคลื่อนที่อย่างสม่ำเสมอ ไม่ว่าจะพิจารณาในช่วงเวลาใด

2.3.3 ความเร็ว (velocity)

เมื่อวัตถุมีการเคลื่อนที่ ตำแหน่งของวัตถุจะเปลี่ยนแปลง การเปลี่ยนแปลงตำแหน่งของวัตถุอาจทำให้ทราบว่าวัตถุเคลื่อนที่มีลักษณะอย่างไร โดยมีการกำหนดว่า อัตราการเปลี่ยนแปลงการกระจัด หรือ การกระจัดที่เปลี่ยนแปลงไปในหนึ่งหน่วยเวลา เรียกว่า ความเร็ว

เนื่องจากการกระจัดเป็นปริมาณเวกเตอร์ ความเร็วจึงเป็นปริมาณเวกเตอร์ โดยหน่วยของความเร็วจึงคือ เมตรต่อวินาที (m / s) เขียนแทนด้วยสัญลักษณ์ “ V ”

$$\bar{v} = \frac{\Delta \bar{s}}{\Delta t} \quad (2.4)$$

เมื่อ

$\bar{v}$ คือ ความเร็วของวัตถุ มีหน่วย เมตรต่อวินาที (m/s)

$\Delta \bar{s}$ คือ การกระจัดของวัตถุ มีหน่วย เมตร (m)

Δt คือ เวลาที่วัตถุใช้ในการเคลื่อนที่ มีหน่วย วินาที (s)

ความเร็วสามารถแบ่งออกได้ คือ

1. ความเร็วขณะใดขณะหนึ่ง คือ ความเร็วที่เกิดขึ้น ณ เวลาหนึ่งของการเคลื่อนที่หรือถ้าพูดในอีกความหมายหนึ่ง คือ อัตราการเปลี่ยนแปลงการกระจัดในช่วงเวลานั้น ๆ

2. ความเร็วเฉลี่ย คือ อัตราการเปลี่ยนแปลงการกระจัดต่อหนึ่งหน่วยเวลา

ใช้สัญลักษณ์ “ V_{av} ”

$$\bar{v}_{av} = \frac{\Delta \bar{s}}{\Delta t} \quad (2.5)$$

เมื่อ

$\bar{v}_{av}$ คือ ความเร็วของวัตถุ มีหน่วย เมตรต่อวินาที (m/s)

$\Delta \bar{s}$ คือ การกระจัดของวัตถุ มีหน่วย เมตร (m)

Δt คือ เวลาที่วัตถุใช้ในการเคลื่อนที่ มีหน่วย วินาที (s)

3. ความเร็วคงที่ (V) คือ วัตถุที่มีการเคลื่อนที่อย่างสม่ำเสมอในแนวเส้นตรง ไม่ว่าจะพิจารณาในช่วงเวลาใด ๆ หาได้จากการกระจัดที่เปลี่ยนไปในเวลาหนึ่งหน่วย

2.3.4 ความเร่ง (Acceleration)

จากการศึกษาพบว่า การเคลื่อนที่ของวัตถุพบว่า ในบางครั้งอาจมีการเปลี่ยนแปลงทิศทาง การเคลื่อนที่ การเคลื่อนที่ที่มีการเปลี่ยนแปลงขนาดหรือทิศทางของความเร็ว เรียกว่า ความเร่ง โดยทั่วไปความเร่งหมายถึง อัตราการเปลี่ยนแปลงความเร็ว หรือ ความเร็วที่เปลี่ยนไปในหนึ่งหน่วยเวลา

เนื่องจากความเป็นปริมาตรเวกเตอร์ ความเร่งจึงเป็นปริมาณเวกเตอร์ หน่วยของความเร่งคือ เมตรต่อวินาที² (m/s²)

2.4 อุปกรณ์ที่ใช้ในโครงงาน

2.4.1 อาร์ดูอิโน (Arduino)

Arduino คือ โครงการที่นำชิปไอซีไมโครคอนโทรลเลอร์ตระกูลต่างๆ มาใช้ร่วมกันในภาษา C ซึ่งภาษา C นี้เป็นลักษณะเฉพาะคือมีการเขียนไวยากรณ์ของ Arduino ขึ้นมาเพื่อให้การสั่งงานไมโครคอนโทรลเลอร์ที่แตกต่างกัน สามารถใช้งานโค้ดตัวเดียวกันได้ โดยตัวโครงการได้ออกบอร์ดทดลองมาหลายรูปแบบ เพื่อใช้งานกับ IDE ของตนเอง สาเหตุหลักที่ทำให้ Arduino เป็น นิยมมากเป็นเพราะซอฟต์แวร์ที่ใช้งานร่วมกันสามารถโหลดได้ฟรี เป็นบอร์ดมาตรฐานที่นิยมใช้งานมากที่สุด ด้วยเหตุนี้เองทำให้อาร์ดูอิโนได้รับความนิยมจากผู้ใช้งานทั่วโลก อาร์ดูอิโนสามารถนำไปประยุกต์ใช้งานเกี่ยวกับการเชื่อมต่อกับอุปกรณ์รับเข้าและส่งออกต่างๆ ได้มากมายทั้งแบบดิจิทัล (Digital) และอนาล็อก (Analog) เช่น การรับค่าจากสวิตช์ หรืออุปกรณ์ตรวจจับ (Sensor) แบบต่าง ๆ รวมไปถึงการควบคุมอุปกรณ์ส่งออกตั้งแต่ แอลอีดี หลอดไฟ มอเตอร์ รีเลย์ เป็นต้น

จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

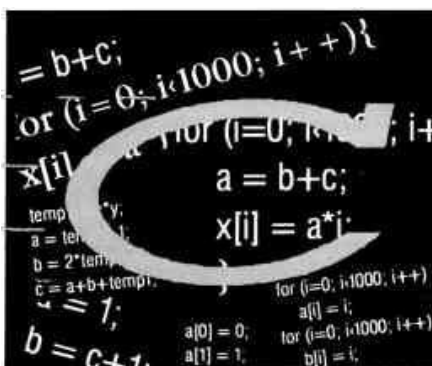
- ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
- มี Arduino Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแกร่ง
- Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
- ราคาไม่แพง
- Cross Platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้

2.4.2 โปรแกรมภาษาของอาร์ดูอิโน (Arduino IDE) [10] Arduino IDE คือเครื่องมือการเขียนโปรแกรมที่มีใช้งานได้กับอาดูอิโนได้ทุกรุ่นโดยภายใน จะมีเครื่องมือที่จำเป็นสำหรับติดต่ออาดูอิโน เช่น การค้นหาอาดูอิโนที่ติดต่อกับเครื่องคอมพิวเตอร์การ เลือกรุ่นอาดูอิโนที่ต่ออยู่เพื่อตรวจสอบว่าขนาดของโปรแกรมที่เขียนหรือไบบารีต่าง ๆ เข้าพอร์ทกับอาดูอิโนรุ่นนั้นๆหรือไม่อีกทั้งยังมีโปรแกรมติดต่อผ่านซีเรียลโดยตรงสำหรับคอมพิวเตอร์ทำความรู้จักกับอาดูอิโน ดังภาพที่ 2-6



ภาพที่ 2-6 โลโก้โปรแกรม Arduino IDE [10]

2.4.3 ภาษาซี (C Programming Language) [11]



ภาพที่ 2-7 ภาษาซี (C Programming Language) [11]

จากภาพที่ 2-6 ภาษาซี (C Programming Language) [11] คือ ภาษาคอมพิวเตอร์ใช้สำหรับพัฒนาโปรแกรมทั่วไป ถูกพัฒนาครั้งแรกเพื่อใช้เป็นภาษาสำหรับพัฒนาระบบปฏิบัติการยูนิกซ์ (Unix Operating System) แทนภาษาแอสเซมบลี ซึ่งเป็นภาษาระดับต่ำที่สามารถกระทำในระบบฮาร์ดแวร์ได้ด้วยความรวดเร็ว แต่จุดอ่อนของภาษาแอสเซมบลีก็คือความยุ่งยากในการโปรแกรม ความเป็นเฉพาะตัว และความแตกต่างกันไปในแต่ละเครื่อง เดนนิส ริตชี (Dennis Ritchie) จึงได้คิดค้นพัฒนาภาษาใหม่นี้ขึ้นมาเมื่อประมาณต้นปี ค.ศ. 1970

2.4.4 ESP32 Node MCU ESP-WROOM-32 Wi-Fi and Bluetooth [12] โมดูล Wifi ESP-32 รุ่น ESP-WROOM-32 โมดูล Wifi + Bluetooth 4.2 + Touch/Temp Sensor ทำงานแบบ Dual Core ที่ความเร็ว 160Mhz มี SRAM 512K หน่วยความจำ Flash สำหรับอัปโหลดโปรแกรม

ขนาด 16M มีขา GPIO 36 ขา ความละเอียดในการอ่านค่า ADC 12 Bit สามารถเขียนโปรแกรม ผ่าน Arduino IDE เหมือนเขียน Arduino ได้

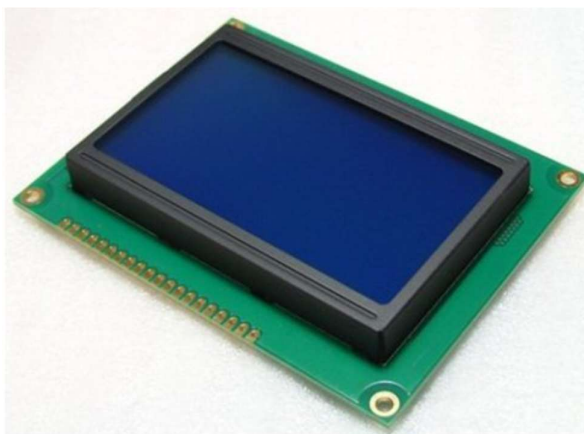
โมดูลรวม USB TTL และ ESP-32 ไว้ในตัวแล้วคล้ายกับ Node MCU จึงไม่ต้องบัดกรีหรือต่อวงจรเพิ่ม ติดตั้งบอร์ดใน Arduino IDE แล้วเสียบสาย USB โปรแกรมได้เลย ดังภาพที่ 2-8



ภาพที่ 2-8 ESP32 NodeMCU ESP-WROOM-32 Wi-Fi and Bluetooth [12]

2.4.5 LCD 128x64 Dots LCD Graphic Matrix Display จอ LCD monochrome [13]

ขนาด 128x64 มี Backlight ในตัว สามารถใช้กับ I2C LCD module ได้ ดังภาพที่ 2-9



ภาพที่ 2-9 LCD 128x64 Dots LCD Graphic Matrix Display [13]

2.4.6 LCD IIC/I2C Interface Module Port IIC/I2C/TWI/SPI Interface Module for 1602 LCD Display [14] เป็นโมดูลที่ใช้สำหรับแปลงการติดต่อสื่อสารระหว่างจอ LCD กับบอร์ด Arduino ให้อยู่ในรูปแบบ I2C ซึ่งข้อดีก็คือสามารถลด pin ใช้งานจาก 6 ให้เหลือ 2 pin ดังภาพที่ 2-10



ภาพที่ 2-10 LCD IIC/I2C Interface [14]

2.4.7 IR Sensor Module E18-D80NK Infrared Reflectance Sensor Photoelectric Sensor [15] เป็น เซ็นเซอร์สวิตช์ อินฟราเรด มีระยะการทำงานที่ 3 - 80 ซม. สามารถปรับระยะทางได้จากตัวต้านทานปรับค่าได้ ใช้ไฟ 5 โวลต์ ทำงานแบบแสงอินฟราเรดจึงไม่ถูกรบกวนได้ง่าย สะดวกในการนำไปประกอบวงจร สามารถนำไปประยุกต์ใช้กับงานหุ่นยนต์หลบสิ่งกีดขวาง และงาน automation อื่นๆ เมื่อมีวัตถุผ่านเซ็นเซอร์จะให้ค่าออกมาเป็น 1 และเมื่อไม่มีวัตถุมาบังส่งค่าออกเป็น 0



ภาพที่ 2-11 IR Sensor Module [15]

2.4.8 Switching Power Supply AC to DC Switching Power Supply HIECUBE (HE05P15LRN) 220VAC to 5VDC 3A 15W [16] แปลงไฟ 220VAC เป็น 5VDC สามารถจ่ายกระแสได้ 3A ดังภาพที่ 2-12



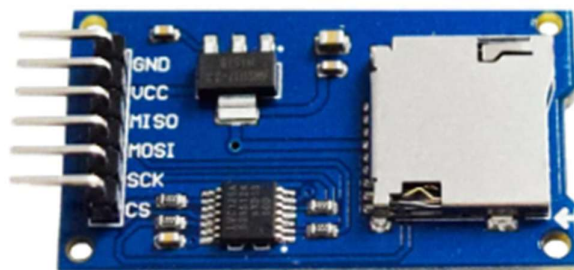
ภาพที่ 2-12 Switching Power Supply AC to DC [16]

2.4.9 DC-to-DC Step Down Module DC-DC Step-Down Ultra-Small Module [17] 7V-28V to 5V 3A Fixed Output โมดูล Step Down ขนาดเล็ก 7V-28V to 5V 3A ดังภาพที่ 2-13



ภาพที่ 2-13 DC-to-DC Step Down Module [17]

2.4.10 Micro SD Card Module โมดูล SD card รุ่นนี้รองรับการใช้งานการ์ดขนาดเล็ก (Micro SD) [18] ทำหน้าที่บันทึกข้อมูลลง SD Card เก็บ Data logger โดยใช้ร่วมกับ Arduino มี library มาตรฐานพร้อมใช้งาน สามารถใช้งานได้ง่าย ดังภาพที่ 2-14



ภาพที่ 2-14 Micro SD Card Module [18]

2.4.11 Relay module Arduino 5V [19] รีเลย์โมดูล Arduino 5V เป็นอุปกรณ์ที่ใช้ในการควบคุมอุปกรณ์ไฟฟ้าที่มีแรงดันไฟฟ้าและกระแสสูง โดยใช้สัญญาณควบคุมจากไมโครคอนโทรลเลอร์ที่มีแรงดันต่ำ เช่น Arduino รีเลย์ทำหน้าที่เป็นสวิตช์ทางไฟฟ้าที่สามารถเปิดหรือปิดวงจรไฟฟ้ากำลังได้ด้วยสัญญาณควบคุมเล็กๆ ดังภาพที่ 2-15



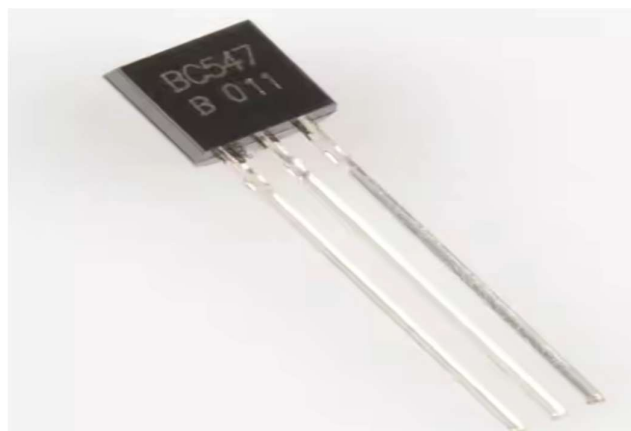
ภาพที่ 2-15 Relay module Arduino 5V [19]

2.4.12 Micro switch tact switch button switch [20] เป็นสวิตช์ขนาดเล็กที่รู้จักกันในชื่อ snap-action switch เนื่องจากมีกลไกภายในที่ทำให้หน้าสัมผัสเปลี่ยนสถานะอย่างรวดเร็วและเด็ดขาด (เปิด/ปิด) เมื่อถูกกดด้วยแรงเพียงเล็กน้อย หรือมีการเคลื่อนที่เพียงเล็กน้อยของกลไก ดังภาพที่ 2-16



ภาพที่ 2-16 Micro switch tact switch button switch [20]

2.4.13 Transistor BC547 [21] ทรานซิสเตอร์ BC547 เป็นทรานซิสเตอร์ชนิด NPN Bipolar Junction Transistor (BJT) ที่นิยมใช้กันอย่างแพร่หลายในวงจรอิเล็กทรอนิกส์สำหรับงานทั่วไป เช่น วงจรขยายสัญญาณและวงจรสวิตช์ มีคุณสมบัติเด่นที่เหมาะสมสำหรับงานพลังงานต่ำและสัญญาณขนาดเล็ก ดังภาพที่ 2-17



ภาพที่ 2-17 Transistor BC547 [21]

บทที่ 3

วิธีการดำเนินงาน

ในบทนี้จะกล่าวถึง การศึกษาอุปกรณ์ลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุที่นำมาพัฒนาการสร้างอุปกรณ์ลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ วงจรการทำงานของลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ ขั้นตอนการประกอบรายละเอียดชิ้นงาน หลักการทำงานของอุปกรณ์ลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ รายการอุปกรณ์ที่ใช้ในการจัดทำลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ การเขียนโปรแกรม Arduino ดังต่อไปนี้

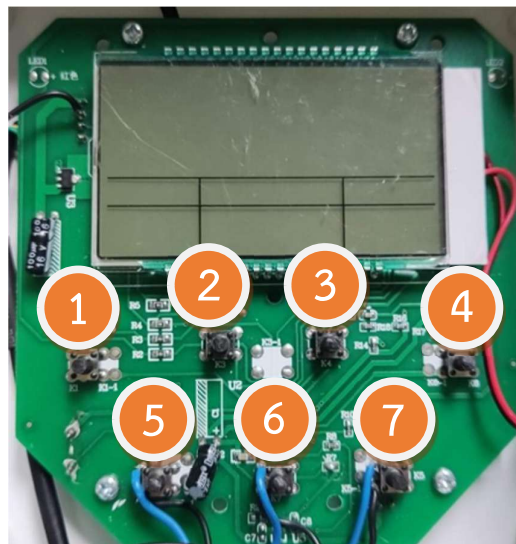
3.1 การศึกษาอุปกรณ์ลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุที่นำมาพัฒนา

3.1.1 ศึกษาลู่วิ่งรุ่น Kee Going Max นำมาใช้ในการพัฒนาเครื่อง พารามิเตอร์ แรงม้าสูงสุด 2.0 hp แรงม้าคงที่ 1.0 hp ขนาดพับ 36*61.5*117.5 cm ขนาดขยาย 124*61.5*117.5 cm น้ำหนักเครื่อง 28 kg รองรับน้ำหนักสูงสุด 120 kg



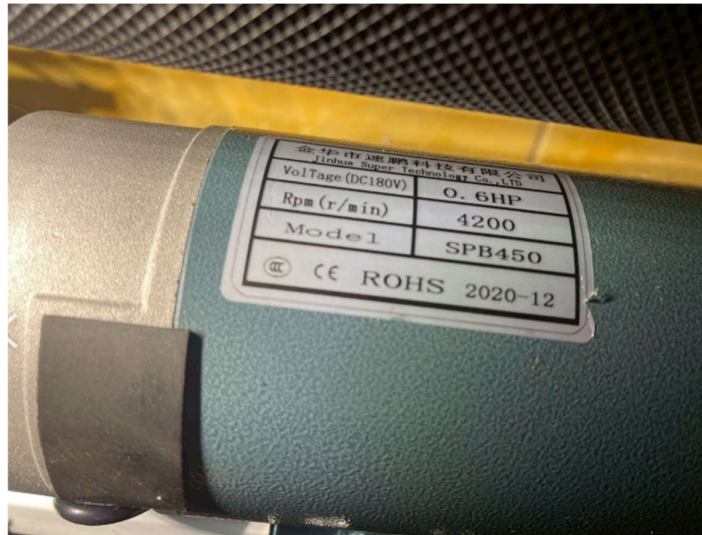
ภาพที่ 3-1 ลู่วิ่งรุ่น Kee Going Max

3.1.2 ศึกษาหน้าจอลู่วิ่งที่นำมาพัฒนา หน้าจอลู่วิ่งของเดิมจะประกอบไปด้วยฟังก์ชันของปุ่มดังนี้ ปุ่มที่ 1 ความเร็ว 2 KM/H, ปุ่มที่ 2 ความเร็ว 4 KM/H, ปุ่มที่ 3 ความเร็ว 6 KM/H, ปุ่มที่ 4 การทำงานของโหมด MODE, ปุ่มที่ 5 เพิ่มความเร็วสายพาน, ปุ่มที่ 6 การทำงาน START/STOP และปุ่มที่ 7 ลดความเร็วสายพาน ดังภาพที่ 3-2



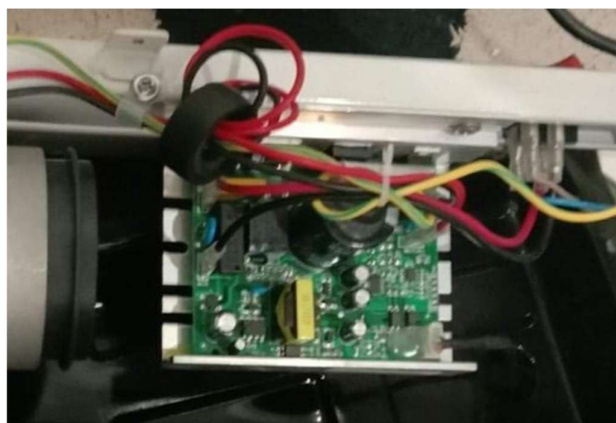
ภาพที่ 3-2 หน้าจอลู่วิ่งที่นำมาพัฒนา

3.1.3 ศึกษาหน้าจอลู่วิ่งที่นำมาพัฒนา มอเตอร์ที่ใช้ในการขับเคลื่อนสายพานเป็นแบบกระแสตรง (DC motor) แรงดัน 180 V ขนาด 0.6 HP ความเร็วรอบ 4200 Rpm(r/min) ดังภาพที่ 3-3



ภาพที่ 3-3 มอเตอร์ที่ใช้ในการขับเคลื่อนสายพานเป็นแบบกระแสตรง

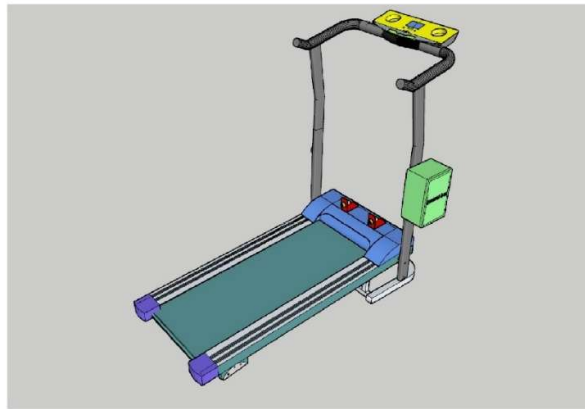
3.1.4 ศึกษาบอร์ดที่ใช้ควบคุมการทำงานของมอเตอร์ การควบคุมมอเตอร์กระแสตรงมักเกี่ยวข้องกับการควบคุมอย่างน้อยหนึ่งในสามปริมาณหลักนี้ (ความเร็ว, ทิศทาง, แรงบิด) โดยใช้วิธีการต่างๆ ขึ้นอยู่กับชนิดของมอเตอร์ (เช่น มอเตอร์แบบ Shunt, Series, Compound) ใช้งานนั้นๆ เทคนิค PWM ผ่านวงจร DC Chopper หรือ H-Bridge Driver เป็นวิธีการที่ทันสมัยและนิยมใช้กันอย่างแพร่หลายในการควบคุมความเร็วและทิศทางของมอเตอร์กระแสตรงในปัจจุบัน เนื่องจากประสิทธิภาพและความยืดหยุ่นในการควบคุม ดังภาพที่ 3-4



ภาพที่ 3-4 บอร์ดที่ใช้ควบคุมการทำงานของมอเตอร์กระแสตรง

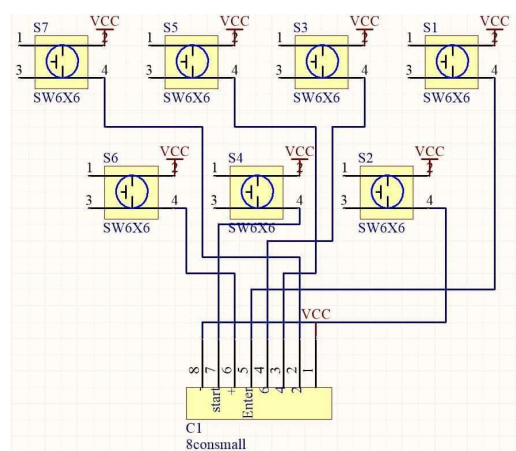
3.2 การสร้างอุปกรณ์ลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ

3.2.1 การออกแบบโครงสร้างของลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ จะใช้โปรแกรม Sketch Up 2020 ในการออกแบบเพื่อที่จะนำไปเป็นแบบในการใช้ทำอุปกรณ์ที่นำไปใช้งานจริง ดังภาพที่ 3-5



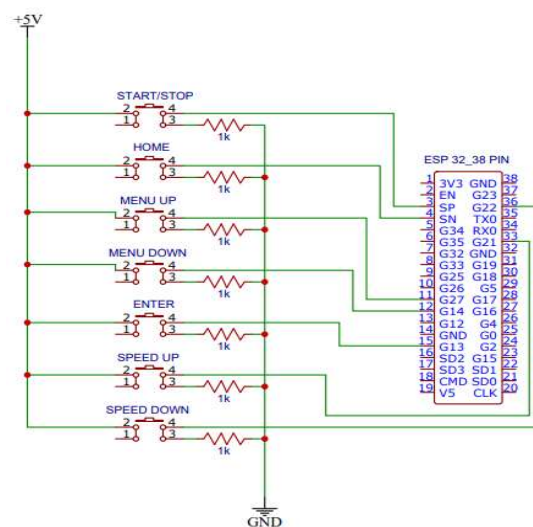
ภาพที่ 3-5 การออกแบบโครงสร้างของลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ

3.2.2 ออกแบบวงจรใหม่ใช้ในการคอลโทรลคำสั่ง Input ของเครื่องลู่วิ่งจะประกอบไปด้วย ฟังก์ชันของปุ่มดังนี้ ปุ่มที่ S7 คำสั่ง HOME, ปุ่มที่ S5 คำสั่งเลื่อนขึ้น, ปุ่มที่ S3 คำสั่งเลื่อนลง, ปุ่มที่ S1 คำสั่ง ENTER, ปุ่มที่ S6 เพิ่มความเร็วสายพาน, ปุ่มที่ S4 การทำงาน START/STOP และ ปุ่มที่ S2 ลดความเร็วสายพาน โดยเชื่อมต่อผ่านขั้วเชื่อมเทอร์มินอล ดังภาพที่ 3-6



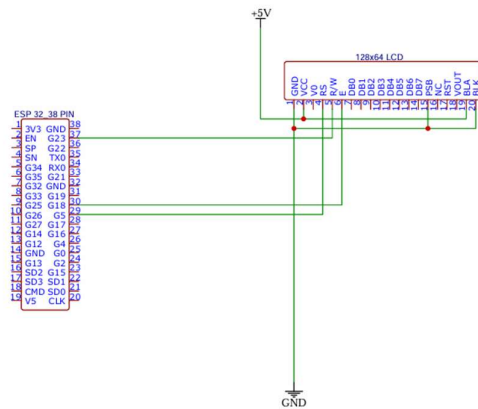
ภาพที่ 3-6 ออกแบบวงจรใหม่ใช้ในการคอลโทรลคำสั่ง Input

3.2.3 ออกแบบวงจรใหม่เป็นวงจรสำหรับเชื่อมต่อปุ่มกด (Push Buttons) จำนวน 7 ปุ่ม เข้ากับ ESP32 (38-pin) โดยใช้การเชื่อมต่อแบบ Pull-down resistors เพื่ออ่านสถานะของ ปุ่ม START/STOP (ต่อกับ ขา3) HOME (ต่อกับ ขา4) MENU UP (ต่อกับ ขา11) MENU DOWN (ต่อกับ ขา12) ENTER (ต่อกับ ขา15) SPEED UP (ต่อกับ ขา33) และ SPEED DOWN (ต่อกับ ขา36) ดังภาพที่ 3-7



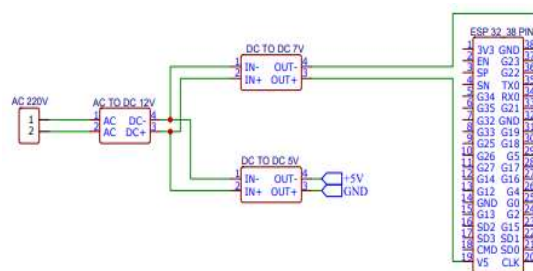
ภาพที่ 3-7 ออกแบบวงจรใหม่ใช้ในส่วนเมนูบนจอควบคุม

3.2.4 ออกแบบวงจรใหม่ เป็นการเชื่อมต่อ ESP32 (38-pin) เข้ากับจอ LCD ขนาด 128x64 พิกเซล โดยใช้การสื่อสารแบบขนาน (Parallel Interface) ซึ่งเป็นวิธีการส่งข้อมูลที่ค่อนข้างรวดเร็วสำหรับ LCD ประเภทนี้ ESP32 ทำหน้าที่ควบคุมการแสดงผลบนจอ LCD โดยส่งข้อมูลและคำสั่งไปยัง LCD ดังภาพที่ 3-8



ภาพที่ 3-8 ออกแบบวงจรใหม่ใช้ในส่งสัญญาณไปแสดงยังหน้าจอ LCD

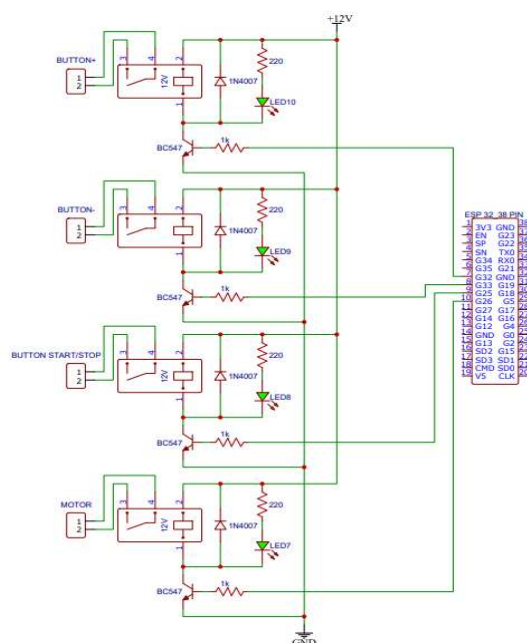
3.2.5 ออกแบบวงจรใหม่โดยแปลงแรงดันแปลงไฟ AC เป็น DC 12V ไฟบ้าน AC 220V ถูกแปลงเป็นไฟ DC 12V โดยโมดูล AC TO DC 12V แยกแรงดันสำหรับไฟ DC 12V นี้ถูกส่งต่อไปยังโมดูล DC TO DC 7V เพื่อลดแรงดันเป็น 7V ซึ่งเป็นแรงดันที่เหมาะสมสำหรับป้อนเข้าขาของ ESP32 แยกแรงดันสำหรับอุปกรณ์เสริม ในขณะเดียวกัน ไฟ DC 12V ก็ถูกส่งต่อไปยังโมดูล DC TO DC 5V เพื่อลดแรงดันเป็น 5V สำหรับจ่ายไฟให้กับอุปกรณ์ภายนอกที่ต้องการไฟ 5V ดังภาพที่ 3-9

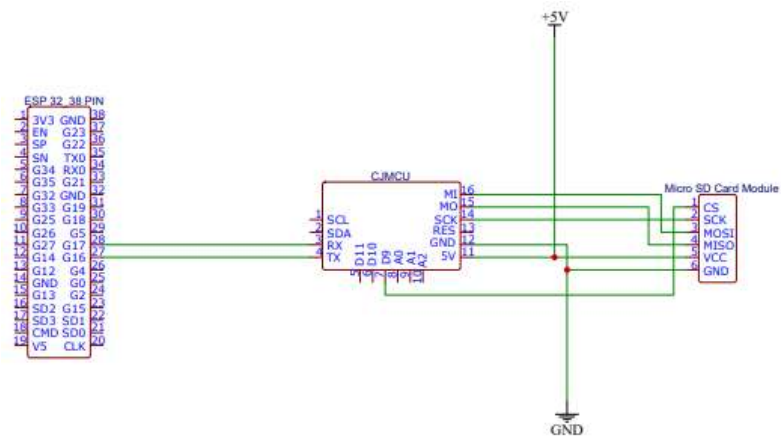


ภาพที่ 3-9 ออกแบบวงจรใหม่โดยแปลงแรงดันจาก Ac to Dc และ stepdown

3.2.6 วงจร Push button switch เป็นการเชื่อมต่อ ESP32 (38-pin) กับรีเลย์ 4 ตัว ซึ่งแต่ละตัวควบคุมด้วยทรานซิสเตอร์ BC547 และมี LED แสดงสถานะ เพื่อสั่งงานชุดปั๊มคำสั่งของบอร์ดหน้าจอเดิมทำงาน ปุ่ม Button+ , ปุ่ม Button- , ปุ่ม Start/Stop และ Motor ESP32 -สัญญาณควบคุม: เมื่อ ESP32 ต้องการเปิดรีเลย์ตัวใดตัวหนึ่ง จะส่งสัญญาณดิจิทัล HIGH (ประมาณ 3.3V) ไปยังขา GPIO ที่เชื่อมต่อกับ Base ของทรานซิสเตอร์ที่เกี่ยวข้อง

- ทรานซิสเตอร์ทำงาน: สัญญาณ HIGH นี้จะไป Bias ให้ทรานซิสเตอร์ BC547 ทำงาน (เหมือนสวิตช์ ปิด)ทำให้กระแสไฟฟ้าจากแหล่งจ่าย +12V ไหลผ่านขดลวดของรีเลย์และผ่านทรานซิสเตอร์ลงสู่ GND
- รีเลย์ทำงาน: การไหลของกระแสผ่านขดลวดรีเลย์จะสร้างสนามแม่เหล็กที่ดึงดูดหน้าสัมผัสภายในรีเลย์ให้เปลี่ยนสถานะ (จากปกติเปิดเป็นปิด หรือปกติปิดเป็นเปิด) เพื่อเชื่อมต่อวงจรภายนอก
- LED สว่าง: ในขณะเดียวกัน LED แสดงสถานะที่ต่อขนานอยู่ก็จะสว่างขึ้นด้วย
- ไดโอดป้องกัน: เมื่อ ESP32 สั่งปิดรีเลย์ (ส่งสัญญาณ LOW ไปยังทรานซิสเตอร์) ทรานซิสเตอร์จะหยุดทำงาน สนามแม่เหล็กในขดลวดจะยุบตัวลงและเกิดแรงดันย้อนกลับ แต่ไดโอด 1N4007 จะทำหน้าที่ระบายแรงดันนี้ออกไป ป้องกันไม่ให้ทรานซิสเตอร์เสียหาย ดังภาพที่ 3-10

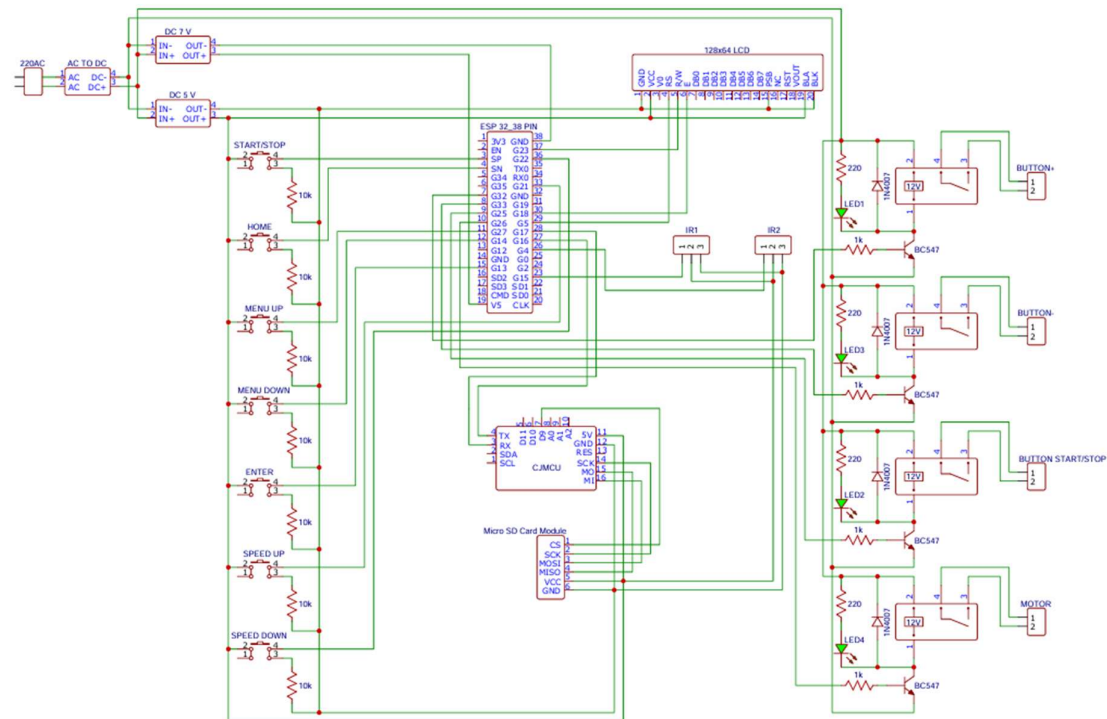




ภาพที่ 3-11 วงจร Save Data

3.3 วงจรการทำงานของลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ

วงจรที่สร้างมาคอลลโทรลบอร์ดเดิมของลู่วิ่ง ดังภาพที่ 3-12

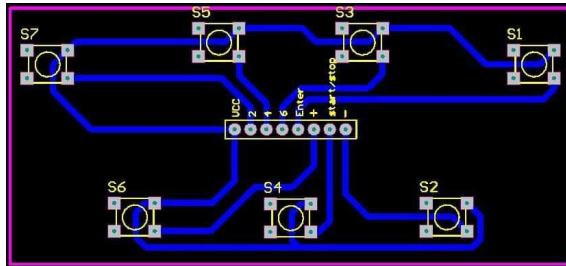


ภาพที่ 3-12 วงจรการทำงานของลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ

3.4 ขั้นตอนการประกอบรายละเอียดชิ้นงาน

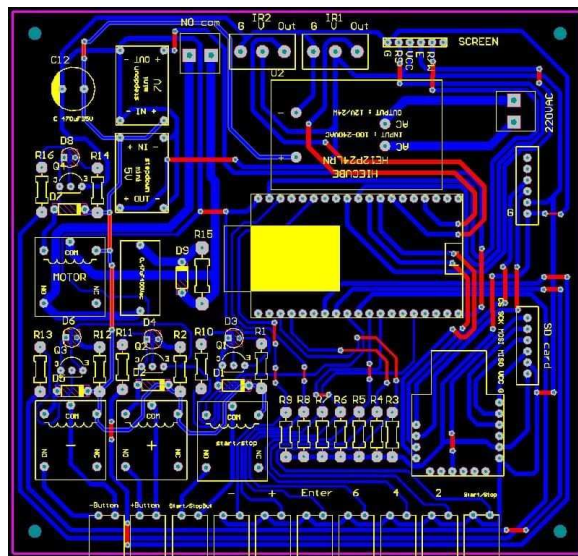
3.4.1 การออกแบบวงจร PCB ตำแหน่งอุปกรณ์ใช้งาน

ภาพ Layout นี้แสดงถึงการออกแบบแผงวงจรสำหรับโมดูลปุ่มกด 7 ปุ่ม โดยมีคอนเนคเตอร์ 8 pin เป็นจุดเชื่อมต่อหลัก ดังภาพที่ 3-13



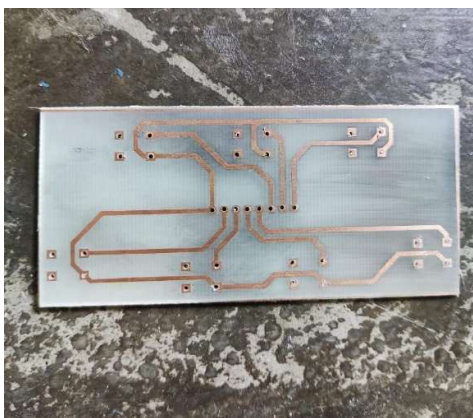
ภาพที่ 3-13 วงจรโมดูลปุ่มกดปุ่มค่า Input

ภาพ PCB Layout ของแผงวงจรหลัก ซึ่งเป็นภาพรวมของการจัดวางส่วนประกอบอุปกรณ์อิเล็กทรอนิกส์และลายทองแดง ดังภาพที่ 3-14

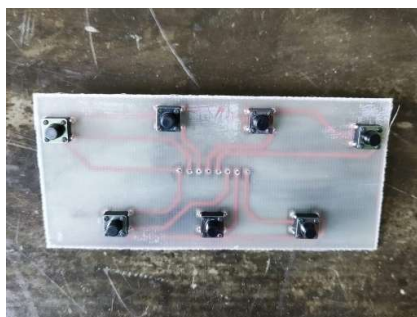


ภาพที่ 3-14 วงจรของแผงวงจรหลักและตำแหน่งอุปกรณ์

3.4.2 ออกแบบแผงวงจรสำหรับโมดูลปุ่มกด 7 ปุ่ม โดยมีคอนเนคเตอร์ 8 pin เป็นจุดเชื่อมต่อหลัก เป็นการลงอุปกรณ์บนแผ่น PCB ที่ลอกลายเสร็จแล้วนำมาบัดกรีพร้อมทั้งลงอุปกรณ์ ดังภาพที่ 3-15 และภาพที่ 3-16

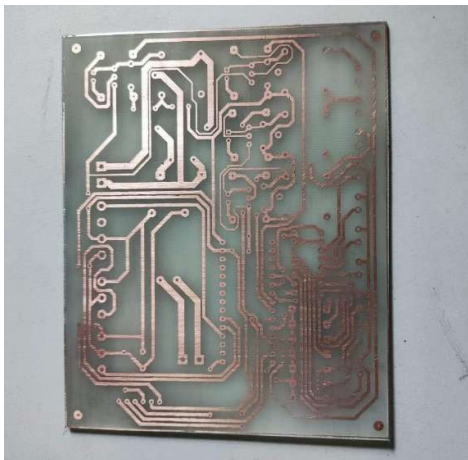


ภาพที่ 3-15 วงจรโมดูลปุ่มกดป้อนค่า Input ที่ลอกลายเสร็จ

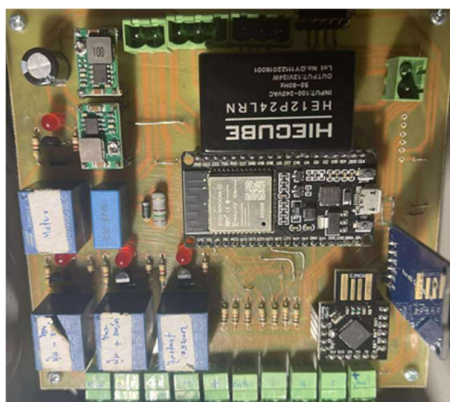


ภาพที่ 3-16 วงจรโมดูลปุ่มกดป้อนค่า Input ที่ลงอุปกรณ์เสร็จ

3.4.3 ออกแบบแผงวงจรวงจรหลักของเครื่อง และเป็นการลงอุปกรณ์บนแผ่น PCB ที่ลอกลายเสร็จแล้วนำมาบัดกรีพร้อมทั้งลงอุปกรณ์ ดังภาพที่ 3-17 และภาพที่ 3-18



ภาพที่ 3-17 วงจรโมดูลหลักของเครื่อง ที่ลอกลายเสร็จ



ภาพที่ 3-18 วงจรโมดูลหลักของเครื่อง ที่ลงอุปกรณ์เสร็จ

3.4.4 การประกอบหน้าจอLCD กับชุดคำสั่งป้อนค่าInputของเครื่อง พร้อมจับยึดอุปกรณ์
ตำแหน่งเดียวกันกับหน้าจอเดิม พร้อมจัดเก็บสาย ดังภาพที่ 3-19



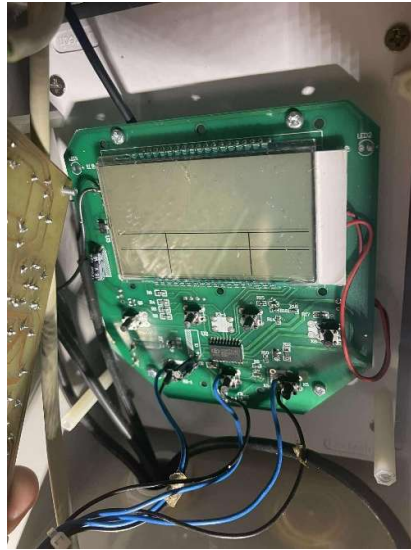
ภาพที่ 3-19 ติดตั้งหน้าจอแสดงผลและชุดคำสั่ง Input

3.4.5 การประกอบวงจรชุดคำสั่งหลัก ติดตั้งแผ่น PCB ที่ลงอุปกรณ์เสร็จนำมาติดตั้งใน Box electric พร้อมจับยึดให้แน่นและทำการต่อสายตามวงจรให้ถูกต้อง ดังภาพที่ 3-20



ภาพที่ 3-20 ติดตั้งวงจรชุดคำสั่งหลักเข้ากับกล่อง Box electric

3.4.6 การประกอบติดตั้งชุดบอร์ดคำสั่งของเครื่องเดิมเข้ากับบอร์ดที่สร้างขึ้นมาใหม่ เพื่อใช้ในการควบคุมความเร็วโหมคแมนนวล ดังภาพที่ 3-21



ภาพที่ 3-21 ติดตั้งชุดบอร์ดคำสั่งของเครื่องเดิมเข้ากับบอร์ดที่สร้างขึ้นมาใหม่

3.4.7 การประกอบติดตั้ง IR sensor1, IR sensor2 ตำแหน่งหน้าสายพานของลู่วิ่ง เพื่อควบคุมการช่วยเดินในโหมดอัตโนมัติ ดังภาพที่ 3-22



ภาพที่ 3-22 ติดตั้ง IR sensor1 และ IR sensor2

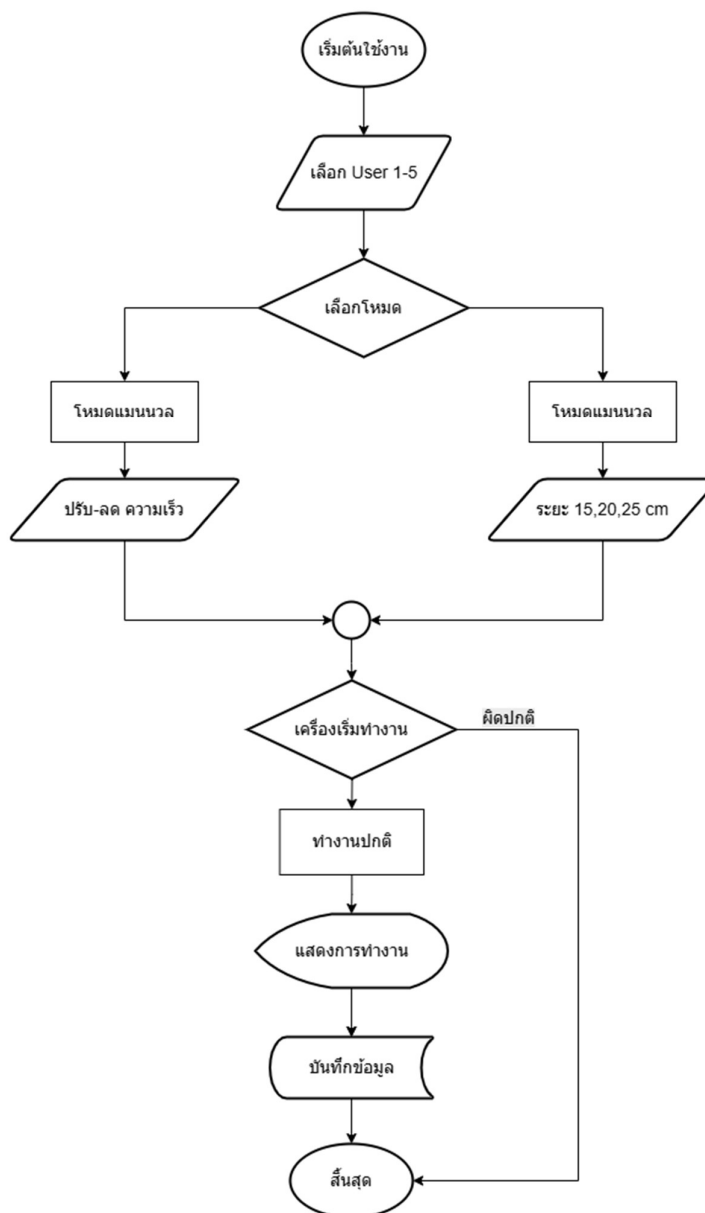
3.4.8 ประกอบเครื่องตามการออกแบบและทำการต่อสายระหว่างชุดป้อนคำสั่ง Input จากปุ่มด้านบนหน้าจอแสดงผล ต่อสายการทำงานระหว่าง IR sensor1,IR sensor2 เข้ากับชุดคำสั่งหลักของเครื่องพร้อมจัดเก็บสายไฟให้เรียบร้อย ดังภาพภาพที่ 3-23



ภาพที่ 3-23 เครื่องลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุที่เสร็จสมบูรณ์

3.5 หลักการทำงานของอุปกรณ์สำหรับผู้ป่วยและสูงอายุ

3.5.1 ขั้นตอนการทำงานของอุปกรณ์แบบ Flow Chart

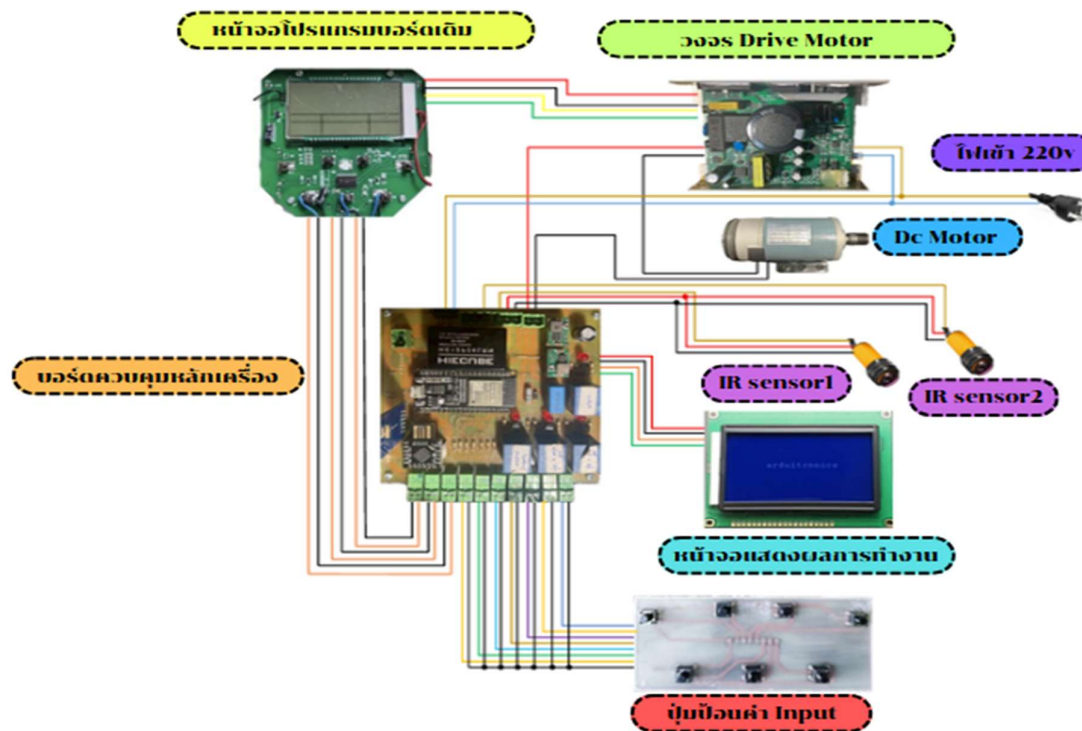


ภาพที่ 3-24 การทำงานของอุปกรณ์แบบ Flow Chart

หลักการทำงานของอุปกรณ์จะแบ่งการทำงานออกเป็น 2 โหมด คือ โหมดแมนนวลและโหมดอัตโนมัติโดยโหมดแมนนวลผู้ใช้สามารถเลือกความเร็วตามต้องการได้โดย การเพิ่ม - ลดจากหน้าจอของเครื่องและโหมดอัตโนมัติผู้ใช้สามารถเลือกโหมดตามระยะที่ต้องการ

- Adaptor 220vac to 12vdc เป็นอุปกรณ์แปลงไฟจาก 220vac เป็น 12vdc เพื่อจ่ายไปยังส่วนต่างๆ
- Dc to Dc stepdown converter 5V / 7V เป็นอุปกรณ์ลดแรงดันเพื่อไปใช้งานยังอุปกรณ์ต่าง แรงดันเอาต์พุต 5V / 7V
- IR sensor 1,2 เป็นอุปกรณ์ตรวจจับวัตถุชนิดNPN ใช้ในการตรวจจับตำแหน่งเท้าของผู้ใช้งาน
- บอร์ด ESP32 เป็นอุปกรณ์ประมวลผลหลักซึ่งรับสัญญาณInput เข้ามาประมวลผล และส่งสัญญาณ Output ไปควบคุมส่วนต่างๆ ที่กำหนดไว้ตามโปรแกรม
- Relay module อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่เป็นสวิตช์ไฟฟ้าที่ควบคุมด้วยสัญญาณไฟฟ้าขนาดเล็ก เช่น จากไมโครคอนโทรลเลอร์
- Motor เป็นอุปกรณ์ที่เปลี่ยนจากพลังงานไฟฟ้าเป็นพลังงานกลเพื่อขับเคลื่อนให้หมุน
- SD card module เป็นอุปกรณ์ที่ช่วยให้ไมโครคอนโทรลเลอร์ เช่น Arduino หรือ Raspberry Pi สามารถเชื่อมต่อการ์ด SD เพื่อจัดเก็บข้อมูลหรืออ่านข้อมูลได้ง่ายขึ้น
- LCD screen หน้าจอแสดงค่าทั้งหมดที่ใช้งานของเครื่อง
- Buzzer อุปกรณ์อิเล็กทรอนิกส์ที่ใช้สร้างเสียง โดยเปลี่ยนสัญญาณไฟฟ้าเป็นเสียงด้วยการสั่น

ไดอะแกรมการต่อใช้งานเครื่องลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ ดังภาพที่ 3-25



ภาพที่ 3-25 ไดอะแกรมการต่อใช้งานเครื่องลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ

3.5.2 กระบวนการทำงานของโหมดแมนนวล

- เลือก User โดยใส่รหัสผ่าน ด้วยการกดปุ่ม Enter จำนวน4ครั้ง
- เลือก login กดปุ่ม Enter เข้าสู่หน้าโหมดการทำงาน
- เลือกโหมดการทำงานแมนนวล
- เลือกโหมดแมนนวลกดปุ่ม Enter
- กด Start เริ่มการทำงานของเครื่อง
- เลือกปุ่มเพิ่ม/ลด ความเร็วสายพานของลู่วิ่งจาก 0.5 km/hr.
- หน้าจอจะแสดง User,ระยะทาง,เวลา,ความเร็ว
- เมื่อหยุดใช้งานเครื่องกดปุ่ม Stop เครื่องจะทำการบันทึกข้อมูลของ User ลงหน่วยความจำเครื่อง หน่วย Km.
- หากเกิดเหตุฉุกเฉินผู้ใช้สามารถกดปุ่ม Stop หรือดึงสาย Safety เพื่อหยุดกาทำงานเครื่อง กรณีดึงสาย Safety เครื่องจะหยุดการทำงานทันทีและมีเสียงแจ้งเตือน

3.5.3 กระบวนการทำงานของโหมดอัตโนมัติ

- เลือก User ใส่รหัสผ่าน โดยการกดปุ่ม Enter จำนวน4ครั้ง
- เลือก login กดปุ่ม Enter เข้าสู่หน้าโหมด
- เลือกโหมดการทำงานอัตโนมัติ/แมนนวล
- เลือกโหมดอัตโนมัติกดปุ่ม Enter
- เลือกระยะการเลื่อนของลู่วิ่ง
 - โหมด1 ระยะ 15 cm.
 - โหมด2 ระยะ 20 cm.
 - โหมด3 ระยะ 25 cm.
- กดปุ่ม Start เพื่อเริ่มการทำงานของเครื่อง
- ก้าวเท้าทั้งสองอยู่ในระยะเซนเซอร์
- เครื่องจะท้าวานช้าๆจนกว่าเท้าทั้งสองจะอยู่นอกระยะเซนเซอร์และผู้ใช้หยุดเครื่อง

- เมื่อกดปุ่ม Stop เครื่องจะบันทึกการใช้งานของ User นั้นลงในหน่วยความจำเครื่อง หน่วย Km.
- หากเกิดเหตุฉุกเฉินผู้ใช้สามารถกดปุ่ม Stop หรือดึงสาย Safety เพื่อหยุดการทำงานของเครื่อง กรณีดึงสาย Safety เครื่องจะหยุดการทำงานทันทีและมีเสียงแจ้งเตือน

3.6 รายการอุปกรณ์ที่ใช้ในการจัดทำตู้วิ่งสำหรับผู้ป่วยและผู้สูงอายุ

อุปกรณ์ที่ใช้ในการจัดทำโครงสร้างและอุปกรณ์ที่ใช้ในการสั่งการควบคุมและต้นกำลัง มีรายการดังต่อไปนี้

ตารางที่ 3-1 อุปกรณ์ที่ใช้ในการสั่งการควบคุมและต้นกำลัง

ลำดับ	ชื่ออุปกรณ์	จำนวน
1	LCD 128x64 Dots LCD Graphic Matrix Display	1 ชิ้น
2	LCD IIC/I2C Interface	1 ชิ้น
3	IR Sensor Module	2 ชิ้น
4	ESP32 NodeMCU ESP-WROOM-32 Wi-Fi and Bluetooth	2 ตัว
5	Switching Power Supply AC To DC	1 ชิ้น
6	DC-to-DC Step Down Module	2 ชิ้น
7	Micro SD Card Module	1 ชิ้น
8	Micro SD Card	1 ชิ้น
9	Relay module Arduino 5V	4 ชิ้น
10	CJMCU	1 ชิ้น
11	Resistor	15 ชิ้น
12	Transistor	4 ชิ้น
13	Capacitor	2 ชิ้น
14	Micro switch button	7 ชิ้น
15	LED	4 ชิ้น
16	Electrical Box	1 กล่อง

3.7 การเขียนโปรแกรม Arduino

เขียนโค้ดในโปรแกรมควบคุมการทำงานของลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ ด้วยบอร์ด ESP32 NodeMCU เพื่อควบคุมการทำงานของมอเตอร์และเซนเซอร์ โดยมีการออกแบบวงจรและทดลองวงจรไปด้วยกัน เพื่อให้ได้โปรแกรมและวงจรใช้งานได้ตามขอบเขตที่ตั้งไว้

```

CODE_EE15
1 #include <U8g2lib.h> //ประกาศไลบรารีสำหรับควบคุมหน้าจอ
2 #include <SoftwareSerial.h> //ประกาศไลบรารีสำหรับสร้าง Serial Port เพิ่ม
3 #include <EEPROM.h> //ประกาศไลบรารีสำหรับอ่านและเขียนข้อมูลลง EEPROM
4 #ifndef U8X8_HAVE_HW_SPI //ประกาศไลบรารีสำหรับเลือกอุปกรณ์ SPI
5 #include <SPI.h> //ประกาศไลบรารีสำหรับเลือกอุปกรณ์ SPI
6 #endif
7 #ifndef U8X8_HAVE_HW_I2C //ประกาศไลบรารีสำหรับเลือกอุปกรณ์ I2C
8 #include <Wire.h> //ประกาศไลบรารีสำหรับเลือกอุปกรณ์ I2C
9 #endif
10
11 #define button_plus 32 //Relay ปุ่ม Up
12 #define button_minus 33 //Relay ปุ่ม Down
13 #define button_start_stop 25 //Relay ปุ่ม Start/Stop
14 #define button_Motor_Out 26 //Relay Control Motor
15 #define IR1_left 15 //เซ็นเซอร์หัว ซ้าย
16 #define IR2_right 4 //เซ็นเซอร์หัว ขวา
17 #define SW_2 12 //ปุ่ม Home
18 #define SW_4 27 //ปุ่ม Up
19 #define SW_6 14 //ปุ่ม Down
20 #define SW_Enter 13 //ปุ่ม Enter
21 #define SW_Minut 21 //ปุ่ม Speed -
22 #define SW_Pluse 22 //ปุ่ม Speed +
23 #define SW_Start_Stop 36 //ปุ่ม Start/Stop
24 #define TIME_INTERVAL 1000
25
26 int t_current = 0; //สร้างตัวแปร t_current ให้มีค่าเท่ากับ 0
27 int t_last = 0; //สร้างตัวแปร t_last ให้มีค่าเท่ากับ 0
  
```

ภาพที่ 3-26 โค้ดโปรแกรม Arduino

บทที่ 4

ผลการทดลองและวิจารณ์

ในบทนี้จะเป็นการกล่าวถึงการทดลอง และผลการทดลองการใช้งานจริงของเครื่องที่ได้ ออกแบบไว้ ซึ่งทำการทดลองเลือกการใช้งานโหมดการเดิน 2 โหมด โหมดอัตโนมัติและโหมดแมนนวล ตามขอบเขตที่กำหนดไว้

4.1 วิธีการทดสอบ

- 4.1.1 ติดตั้งและเชื่อมต่ออุปกรณ์ต่าง ๆ สำหรับการทดสอบลู่วิ่งโหมดอัตโนมัติ
- 4.1.2 ติดตั้งและเชื่อมต่ออุปกรณ์ต่าง ๆ สำหรับการทดสอบลู่วิ่งโหมดแมนนวล

4.2 ผลการทดลอง

4.2.1 การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ

การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ บุคคลที่ 1 ประเภทผู้สูงอายุ โดยทำการทดลอง ทั้งหมด 5 ครั้ง เพื่อหาค่าเฉลี่ย ซึ่งจะได้ผลการทดลองดังแสดงในตารางที่ 4-1

ตารางที่ 4-1 การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ บุคคลที่ 1

ชื่อ-สกุล	บุคคลที่ 1					
อายุ	87 ปี					
น้ำหนัก	51 กิโลกรัม					
ส่วนสูง	158 เซนติเมตร					
โรคประจำตัว	ความดัน, ไขมันในเลือดสูง					
ระยะการเดิน (cm.)	การทดลองครั้งที่					ค่าเฉลี่ย
	1	2	3	4	5	
15	17.0	15.2	16.0	15.4	16.0	15.92
20	20.5	22.0	23.2	21.1	21.8	21.92
25	27.0	26.0	24.2	27.5	26.8	26.30

จากตารางที่ 4-1 บุคคลที่ 1 ประเภทผู้สูงอายุ โดยทำการทดลองทั้งหมด 5 ครั้ง เมื่อปรับระยะการเดินโหมดัดโนมิติ ระยะ 15 cm ค่าเฉลี่ย 15.92 cm, ระยะ 20 cm ค่าเฉลี่ย 21.92 cm และระยะ 25 cm ค่าเฉลี่ย 26.30 cm

การทดสอบลู่วิ่งโดยเลือกโหมดัดโนมิติ บุคคลที่ 2 ประเภทผู้สูงอายุ โดยทำการทดลองทั้งหมด 5 ครั้ง เพื่อหาค่าเฉลี่ย ซึ่งจะได้ผลการทดลองดังแสดงในตารางที่ 4-2

ตารางที่ 4-2 การทดสอบลู่วิ่งโดยเลือกโหมดัดโนมิติ บุคคลที่ 2

ชื่อ-สกุล	บุคคลที่ 2					
อายุ	82 ปี					
น้ำหนัก	59 กิโลกรัม					
ส่วนสูง	150 เซนติเมตร					
โรคประจำตัว	ความดัน, เบาหวาน, ไขมันในเลือดสูง					
ระยะการเดิน (cm.)	การทดลองครั้งที่					ค่าเฉลี่ย
	1	2	3	4	5	
15	15.0	15.3	16.2	15.8	16.0	15.66
20	21.2	20.8	22.0	20.4	20.6	21.00
25	28.0	26.0	27.5	26.3	25.8	26.72

จากตารางที่ 4-2 บุคคลที่ 2 ประเภทผู้สูงอายุ โดยทำการทดลองทั้งหมด 5 ครั้ง เมื่อปรับระยะการเดินโหมดัดโนมิติ ระยะ 15 cm ค่าเฉลี่ย 15.66 cm, ระยะ 20 cm ค่าเฉลี่ย 21.00 cm และระยะ 25 cm ค่าเฉลี่ย 26.72 cm

การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ บุคคลที่ 3 ประเภทผู้ป่วย โดยทำการทดลองทั้งหมด 5 ครั้ง เพื่อหาค่าเฉลี่ย ซึ่งจะได้ผล การทดลองดังแสดงในตารางที่ 4-3

ตารางที่ 4-3 การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ บุคคลที่ 3

ชื่อ-สกุล	บุคคลที่ 3					
อายุ	67 ปี					
น้ำหนัก	57 กิโลกรัม					
ส่วนสูง	161 เซนติเมตร					
โรคประจำตัว	ความดัน, เบาหวาน, ไขมันในเลือดสูง, เก๊าท์					
ระยะการเดิน (cm.)	การทดลองครั้งที่					ค่าเฉลี่ย
	1	2	3	4	5	
15	16.0	16.5	16.8	17.0	16.3	16.52
20	22.0	23.5	23.0	22.0	22.5	22.60
25	27.0	28.0	26.0	28.3	27.2	27.30

จากตารางที่ 4-3 บุคคลที่ 3 ประเภทผู้สูงอายุ โดยทำการทดลองทั้งหมด 5 ครั้ง เมื่อปรับระยะการเดินโหมดอัตโนมัติ ระยะ 15 cm ค่าเฉลี่ย 16.52 cm, ระยะ 20 cm ค่าเฉลี่ย 22.60 cm และระยะ 25 cm ค่าเฉลี่ย 27.30 cm

การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ บุคคลที่ 4 ประเภทผู้ป่วย โดยทำการทดลองทั้งหมด 5 ครั้ง เพื่อหาค่าเฉลี่ย ซึ่งจะได้ผล การทดลองดังแสดงในตารางที่ 4-4

ตารางที่ 4-4 การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ บุคคลที่ 4

ชื่อ-สกุล	บุคคลที่ 4					
อายุ	58 ปี					
น้ำหนัก	62 กิโลกรัม					
ส่วนสูง	151 เซนติเมตร					
โรคประจำตัว	ความดัน, ไขมันในเลือดสูง, เข้าเสื่อม					
ระยะการเดิน (cm.)	การทดลองครั้งที่					ค่าเฉลี่ย
	1	2	3	4	5	
15	15.0	14.0	16.0	16.8	15.2	15.40
20	22.1	22.0	21.6	23.0	21.4	22.02
25	27.0	28.0	26.0	28.0	27.2	27.24

จากตารางที่ 4-4 บุคคลที่ 4 ประเภทผู้สูงอายุ โดยทำการทดลองทั้งหมด 5 ครั้ง เมื่อปรับระยะการเดินโหมดอัตโนมัติ ระยะ 15 cm ค่าเฉลี่ย 15.40 cm, ระยะ 20 cm ค่าเฉลี่ย 22.02 cm และระยะ 25 cm ค่าเฉลี่ย 27.24 cm

การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ บุคคลที่ 5 ประเภทผู้ป่วย โดยทำการทดลองทั้งหมด 5 ครั้ง เพื่อหาค่าเฉลี่ย และค่าความคลาดเคลื่อนเฉลี่ย (%) ซึ่งจะได้ผล การทดลองดังแสดงในตารางที่ 4-5

ตารางที่ 4-5 การทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ บุคคลที่ 5

ชื่อ-สกุล	บุคคลที่ 5					
อายุ	61 ปี					
น้ำหนัก	65 กิโลกรัม					
ส่วนสูง	154 เซนติเมตร					
โรคประจำตัว	ไขมันในเลือดสูง, ผ่าตัดเปลี่ยนเข้า					
ระยะการเดิน (cm.)	การทดลองครั้งที่					ค่าเฉลี่ย
	1	2	3	4	5	
15	16.0	15.6	15.2	15.8	15.6	15.64
20	22	21.5	21.2	22.1	21.6	21.68
25	25.2	25.7	26	25.3	25.6	25.56

จากตารางที่ 4-5 บุคคลที่ 5 ประเภทผู้สูงอายุ โดยทำการทดลองทั้งหมด 5 ครั้ง เมื่อปรับระยะการเดินโหมดอัตโนมัติ ระยะ 15 cm ค่าเฉลี่ย 15.64 cm, ระยะ 20 cm ค่าเฉลี่ย 21.68 cm และระยะ 25 cm ค่าเฉลี่ย 25.56 cm

4.2.2 การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล

การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 1 โดยทำการทดลองทั้งหมด 5 ครั้ง เพื่อหาค่าเฉลี่ย ซึ่งจะได้ผลการทดลองดังแสดงในตารางที่ 4-6

ตารางที่ 4-6 การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 1

ชื่อ-สกุล	บุคคลที่1					
น้ำหนัก	62 กิโลกรัม					
ความเร็ว (km/h)	การทดลองครั้งที่ เวลา 5 นาที					ค่าเฉลี่ย
	1	2	3	4	5	
1.3	0.1	0.1	0.1	0.2	0.1	0.12
2.4	0.2	0.2	0.2	0.2	0.3	0.22
3.5	0.3	0.3	0.4	0.3	0.3	0.32

จากตารางที่ 4-6 บุคคลที่ 1 โดยทำการทดลองโหมดแมนนวลทั้งหมด 5 ครั้ง เมื่อปรับความเร็ว ที่ 1.3 , 2.4 , 3.5 (หน่วย km/h) ตามที่กำหนดไว้ พบว่าค่าเฉลี่ยของที่เคลื่อนที่ได้ความเร็ว 1.3 km/h ค่าเฉลี่ยที่ได้ 0.12 km, ความเร็ว 2.4 km/h ค่าเฉลี่ยที่ได้ 0.22 km และความเร็ว 3.5 km/h ค่าเฉลี่ยที่ได้ 0.32 km

การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 2 โดยทำการทดลองทั้งหมด 5 ครั้ง เพื่อหาค่าเฉลี่ย ซึ่งจะได้ผลการทดลองดังแสดงในตารางที่ 4-7

ตารางที่ 4-7 การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 2

ชื่อ-สกุล	บุคคลที่ 2					
น้ำหนัก	67 กิโลกรัม					
ความเร็ว (km/h)	การทดลองครั้งที่ เวลา 5 นาที					ค่าเฉลี่ย
	1	2	3	4	5	
1.3	0.1	0.1	0.1	0.1	0.1	0.10
2.4	0.2	0.3	0.2	0.2	0.2	0.22
3.5	0.3	0.3	0.3	0.4	0.3	0.32

จากตารางที่ 4-7 บุคคลที่ 2 โดยทำการทดลองโหมดแมนนวลทั้งหมด 5 ครั้ง เมื่อปรับความเร็ว ที่ 1.3 , 2.4 , 3.5 (หน่วย km/h) ตามที่กำหนดไว้ พบว่าค่าเฉลี่ยของที่เคลื่อนที่ได้ความเร็ว 1.3 km/h ค่าเฉลี่ยที่ได้ 0.10 km, ความเร็ว 2.4 km/h ค่าเฉลี่ยที่ได้ 0.22 km และความเร็ว 3.5 km/h ค่าเฉลี่ยที่ได้ 0.32 km

การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 3 โดยทำการทดลองทั้งหมด 5 ครั้ง เพื่อหาค่าเฉลี่ย ซึ่งจะได้ผลการทดลองดังแสดงในตารางที่ 4-8

ตารางที่ 4-8 การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 3

ชื่อ-สกุล	บุคคลที่ 3					
น้ำหนัก	55 กิโลกรัม					
ความเร็ว (km/h)	การทดลองครั้งที่ เวลา 5 นาที					ค่าเฉลี่ย
	1	2	3	4	5	
1.3	0.1	0.2	0.2	0.1	0.2	0.16
2.4	0.2	0.2	0.3	0.3	0.2	0.26
3.5	0.3	0.3	0.3	0.4	0.4	0.34

จากตารางที่ 4-8 บุคคลที่ 3 โดยทำการทดลองโหมดแมนนวลทั้งหมด 5 ครั้ง เมื่อปรับความเร็ว ที่ 1.3 , 2.4 , 3.5 (หน่วย km/h) ตามที่กำหนดไว้ พบว่าค่าเฉลี่ยของที่เคลื่อนที่ได้ความเร็ว 1.3 km/h ค่าเฉลี่ยที่ได้ 0.16 km, ความเร็ว 2.4 km/h ค่าเฉลี่ยที่ได้ 0.26 km และความเร็ว 3.5 km/h ค่าเฉลี่ยที่ได้ 0.34 km

การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 4 โดยทำการทดลองทั้งหมด 5 ครั้ง เพื่อหาค่าเฉลี่ย ซึ่งจะได้ผลการทดลองดังแสดงในตารางที่ 4-9

ตารางที่ 4-9 การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 4

ชื่อ-สกุล	บุคคลที่ 4					
น้ำหนัก	70 กิโลกรัม					
ความเร็ว (km/h)	การทดลองครั้งที่ เวลา 5 นาที					ค่าเฉลี่ย
	1	2	3	4	5	
1.3	0.1	0.1	0.1	0.1	0.1	0.10
2.4	0.2	0.2	0.2	0.2	0.2	0.20
3.5	0.3	0.3	0.3	0.3	0.3	0.30

จากตารางที่ 4-9 บุคคลที่ 4 โดยทำการทดลองโหมดแมนนวลทั้งหมด 5 ครั้ง เมื่อปรับความเร็ว ที่ 1.3 , 2.4 , 3.5 (หน่วย km/h) ตามที่กำหนดไว้ พบว่าค่าเฉลี่ยของที่เคลื่อนที่ได้ความเร็ว 1.3 km/h ค่าเฉลี่ยที่ได้ 0.10 km, ความเร็ว 2.4 km/h ค่าเฉลี่ยที่ได้ 0.20 km และความเร็ว 3.5 km/h ค่าเฉลี่ยที่ได้ 0.30 km

การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 5 โดยทำการทดลองทั้งหมด 5 ครั้ง เพื่อหาค่าเฉลี่ย ซึ่งจะได้ผลการทดลองดังแสดงในตารางที่ 4-10

ตารางที่ 4-10 การทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล บุคคลที่ 5

ชื่อ-สกุล	บุคคลที่ 5					
น้ำหนัก	65 กิโลกรัม					
ความเร็ว (km/h)	การทดลองครั้งที่ เวลา 5 นาที					ค่าเฉลี่ย
	1	2	3	4	5	
1.3	0.1	0.1	0.2	0.1	0.1	0.12
2.4	0.2	0.2	0.3	0.2	0.2	0.22
3.5	0.3	0.4	0.3	0.3	0.3	0.32

จากตารางที่ 4-10 บุคคลที่ 5 โดยทำการทดลองโหมดแมนนวลทั้งหมด 5 ครั้ง เมื่อปรับความเร็ว ที่ 1.3 , 2.4 , 3.5 (หน่วย km/h) ตามที่กำหนดไว้ พบว่าค่าเฉลี่ยของที่เคลื่อนที่ได้ความเร็ว 1.3 km/h ค่าเฉลี่ยที่ได้ 0.12 km, ความเร็ว 2.4 km/h ค่าเฉลี่ยที่ได้ 0.22 km และความเร็ว 3.5 km/h ค่าเฉลี่ยที่ได้ 0.32 km

4.3 อภิปรายผลการทดลอง

4.3.1 จากการทดสอบลู่วิ่งโดยเลือกโหมดอัตโนมัติ ประเภทผู้สูงอายุ จำนวน 5 คน โดยทำการทดลองทั้งหมด 5 ครั้ง เมื่อปรับระยะการเดิน ระยะ 15 cm ค่าเฉลี่ยที่ได้ 15.82 cm, ระยะ 20 cm ค่าเฉลี่ยที่ได้ 21.84 cm และระยะ 25 cm ได้ค่าเฉลี่ย 26.62 cm

โดยการใช้ทฤษฎีการเคลื่อนที่ในแนวตรง การหาอัตราเร็ว สูตร $v = \frac{s}{t}$ (4.1)

โดย v คือ อัตราเร็วของวัตถุ

s คือ ระยะทางของวัตถุ

t คือ เวลาที่วัตถุใช้เคลื่อนที่

คำนวณโหมดอัตโนมัติ ระยะ 15 cm โดยใช้อัตราเร็ว 0.5 km/h แปลงระยะทาง 15 cm

เป็น 0.00015 km จากสูตร $v = \frac{s}{t}$ หาค่าเวลาที่วัตถุใช้เคลื่อนที่ จะได้ $t = \frac{s}{v}$

$$\text{แทนค่าสมการ } t = \frac{0.00015}{0.5} \times 3600$$

$$\begin{aligned}\text{จะได้ } t &= 1.08 \times 1000 \\ &= 1080 \text{ S}\end{aligned}$$

คำนวณโหมดอัตโนมัติ ระยะ 20 cm โดยใช้อัตราเร็ว 0.5 km/h แปลงระยะทาง 20 cm เป็น 0.00020 km จากสูตร $v = \frac{S}{t}$ หาค่าเวลาที่วัตถุใช้เคลื่อนที่ จะได้ $t = \frac{S}{v}$

$$\begin{aligned}\text{แทนค่าสมการ } t &= \frac{0.00020}{0.5} \times 3600 \\ \text{จะได้ } t &= 1.44 \times 1000 \\ &= 1440 \text{ S}\end{aligned}$$

คำนวณโหมดอัตโนมัติ ระยะ 25 cm โดยใช้อัตราเร็ว 0.5 km/h แปลงระยะทาง 25 cm เป็น 0.00025 km จากสูตร $v = \frac{S}{t}$ หาค่าเวลาที่วัตถุใช้เคลื่อนที่ จะได้ $t = \frac{S}{v}$

$$\begin{aligned}\text{แทนค่าสมการ } t &= \frac{0.00025}{0.5} \times 3600 \\ \text{จะได้ } t &= 1.8 \times 1000 \\ &= 1800 \text{ S}\end{aligned}$$

ผลการทดสอบการใช้งานเครื่องจริงตามระยะทางที่กำหนด ในโหมดอัตโนมัติค่าที่เคลื่อนที่ได้จริงมีความใกล้เคียงกับการใช้ทฤษฎีการเคลื่อนที่แนวตรงมาประยุกต์ใช้งานจริง

4.3.2 จากการทดสอบลู่วิ่งโดยเลือกโหมดแมนนวล เนื่องจากผู้ใช้งานในโหมดอัตโนมัติไม่สามารถทดลองในโหมดแมนนวล ได้ทำการทดสอบจากผู้ที่มีความพร้อมในการทดสอบ จำนวน 5 คน โดยทำการทดลองทั้งหมด 5 ครั้ง เมื่อปรับความเร็ว 1.3 km/h ค่าเฉลี่ยที่ได้ 0.12 km, ความเร็ว 2.4 km/h ค่าเฉลี่ยที่ได้ 0.22 km และความเร็ว 3.5 km/h ได้ค่าเฉลี่ย 0.32 km

คำนวณโหมดแมนนวล ความเร็ว 1.3 km/h โดยใช้เวลา 5 นาที แปลง m/s เป็น km/3.6

จากสูตร $v = \frac{S}{t}$ หาค่าระยะทางของวัตถุ จะได้ $S = vt$

$$\text{แทนค่าสมการ } S = 0.36 \times 3600$$

$$\text{จะได้ } S = 1296 \text{ m}$$

หาในเวลา 5 นาที

$$2.3 \text{ km} = 1300 \text{ m}$$

$$v = 1300/60 = 21.66 \text{ m/s}$$

$$\text{หา } S = 21.66 \times 5 = 108.3 \text{ m}$$

คำนวณโหมดแมนนวล ความเร็ว 2.4 km/h โดยใช้เวลา 5 นาที แปลง m/s เป็น km/3.6

จากสูตร $v = \frac{s}{t}$ หาค่าระยะทางของวัตถุ จะได้ $s = vt$

$$\text{แทนค่าสมการ } s = 0.66 \times 3600$$

$$\text{จะได้ } s = 2376 \text{ m}$$

หาในเวลา 5 นาที

$$2.4 \text{ km} = 2400 \text{ m}$$

$$v = 2400/60 = 40 \text{ m/s}$$

$$\text{หา } s = 40 \times 5 = 200 \text{ m}$$

คำนวณโหมดแมนนวล ความเร็ว 3.5 km/h โดยใช้เวลา 5 นาที แปลง m/s เป็น km/3.6

จากสูตร $v = \frac{s}{t}$ หาค่าระยะทางของวัตถุ จะได้ $s = vt$

$$\text{แทนค่าสมการ } s = 0.97 \times 3600$$

$$\text{จะได้ } s = 3492 \text{ m}$$

หาในเวลา 5 นาที

$$3.5 \text{ km} = 3500 \text{ m}$$

$$v = 3500/60 = 40 \text{ m/s}$$

$$\text{หา } s = 58.33 \times 5 = 291 \text{ m}$$

ผลการทดสอบการใช้งานเครื่องจริงตามระยะทางที่กำหนด ในโหมดแมนนวลค่าที่เคลื่อนที่ได้จริง มีความใกล้เคียงกับการใช้ทฤษฎีการเคลื่อนที่แนวตรงมาประยุกต์ใช้งานจริง

บทที่ 5

สรุปผลและข้อเสนอแนะ

ปริญญานิพนธ์นี้การพัฒนาตู้วิ่งสำหรับผู้ป่วยและผู้สูงอายุ โดยมี วัตถุประสงค์เพื่อเพื่อช่วยเหลือผู้ที่มีปัญหาด้านการเคลื่อนไหว กลุ่มเป้าหมายผู้ป่วยและผู้สูงอายุ ซึ่งในบทนี้จะกล่าวถึงบทสรุปผลของโครงการ ปัญหาและอุปสรรคใน การทำโครงการ รวมทั้ง ข้อเสนอแนะ เพื่อเป็นแนวทางในการนำไปทำโครงการรุ่นต่อไป ดังนี้

5.1 สรุปผลที่ได้จากการทำโครงการ

จากการศึกษาและทำการทดลองโครงการเรื่องการพัฒนาตู้วิ่งสำหรับผู้ป่วยและผู้สูงอายุโดยการนำตู้วิ่งที่มีอยู่ในท้องตลาดมาพัฒนาด้วยไมโครคอนโทรลเลอร์ ทำให้ได้เรียนรู้เกี่ยวกับการเขียนโปรแกรม Arduino IDE ในการใช้ควบคุมการทำงานของอุปกรณ์ไมโครคอนโทรลเลอร์ต่างๆ ได้เรียนรู้การออกแบบโครงสร้างด้วยโปรแกรม SketchUp เรียนรู้การทำงานของตู้วิ่งรุ่นที่นำมาพัฒนา เพื่อสร้างโหมดการใช้งาน 2 โหมด โหมดอัตโนมัติ ด้วยระยะ 15 cm, 20 cm และ 25 cm โหมดแมนนวล จะสามารถเลือกใช้ความเร็วได้เหมาะกับผู้ใช้งานผู้ป่วยและผู้สูงอายุ รองรับการใช้งานสูงสุด 5 คนมีการบันทึกค่าการใช้งานของผู้ใช้งาน เครื่องหยุดการทำงานหากเกิดเหตุฉุกเฉินและมีเสียงแจ้งเตือน

5.2 ปัญหาและอุปสรรค

5.2.1 การปรับระยะห่างระหว่างเท้า จากการทดลองพบว่าผู้ใช้งานที่อยู่ในช่วงเท้ากว้าง 19-21 เซนติเมตร เครื่องนี้เหมาะสมกับผู้ใช้งานช่วงระยะห่างเท้า

5.2.2 การใช้งานเครื่องเมื่อขณะผู้ใช้งานใช้เครื่องหากเกิดไฟฟ้าดับเครื่องไม่สามารถบันทึกค่าการใช้งานขณะนั้นได้

5.2.3 หากผู้ใช้งานในโหมดอัตโนมัติเท้าไม่ได้อยู่ในตำแหน่งเซนเซอร์เครื่องไม่สามารถเคลื่อนสายพานได้

5.3 ข้อเสนอแนะ

5.3.1 พัฒนากลไกที่สามารถปรับระยะห่างของฐานรองเท้าหรือจุดรองรับเท้าได้ เพื่รองรับผู้ใช้งานที่มีช่วงความกว้างเท้าแตกต่างกัน กลไกนี้อาจเป็นแบบเลื่อนเข้า-ออกพร้อมตัวล็อก หรือมีอุปกรณ์เสริมที่สามารถถอดเปลี่ยนได้สำหรับผู้ใช้งานแต่ละราย

5.3.2 ตั้งค่าให้ระบบบันทึกข้อมูลการใช้งาน (เช่น ระยะทาง, เวลา) ลงในหน่วยความจำแบบไม่ลบเลือนทุกๆ ช่วงเวลาที่กำหนด (เช่น ทุก 1 นาที หรือทุก 100 เมตรที่เดิน) หรือเมื่อมีการเปลี่ยนแปลงสถานะสำคัญ วิธีนี้จะช่วยให้ข้อมูลไม่สูญหายทั้งหมด หากเกิดไฟฟ้าดับกะทันหัน

5.3.3 ใช้ IMU หรือกล้องขนาดเล็กที่ติดตั้งบริเวณใกล้เคียงเพื่อตรวจจับการเคลื่อนไหวของเท้าและขาของผู้ใช้งานโดยรวม ไม่ใช่แค่ตำแหน่งการวางเท้าเท่านั้น วิธีนี้จะช่วยให้ระบบเข้าใจเจตนาในการเดินของผู้ใช้งานได้ดีขึ้นแม้เท้าจะไม่อยู่บนจุดเซ็นเซอร์

5.4 แนวทางในการพัฒนา

จากการสร้าง และทดลอง เพื่อสร้างเครื่องลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุข้อเสนอแนะที่จะนำไปปรับปรุงคือ

5.4.1 ออกแบบวงจรสั่งการทำงานใช้ IC ไปควบคุมการทำงานแทน Relay

5.4.2 การออกแบบโครงสร้าง สามารถปรับระยะห่างระหว่างเซนเซอร์ได้

5.4.3 การออกแบบโปรแกรมการทำงาน สามารถปรับระยะการเคลื่อนที่ของสายพานได้ละเอียด

เอกสารอ้างอิง

- [1] ระบบสารสนเทศด้านการแพทย์และสุขภาพ. 2567. อัตราตายของผู้ป่วยโรคหลอดเลือดสมอง. วารสารสำนักดิจิทัลการแพทย์. 68(1), 1-16. https://medinfo.dms.go.th/public-health/kpi_reports.php?kpi_year=2567&id=1.
- [2] นพ.ศักรินทร์ วงศ์เลิศศิริ. 2562. พื้นฟูผู้ป่วยเดินผิดปกติ. สถาบันสิรินธร. <https://www.thaihealth.or.th/?p=292119>.
- [3] ดวงภรณ์ แดงจัน. 2012. การประมาณความสูงจากความยาวของการก้าวขณะเดิน. กลุ่มวิทยาศาสตร์. <https://he02.tci-thaijo.org/index.php/Veridian-E-Journal/article/download/28231/24263>.
- [4] นายจรัญ คนแรง. 2559. งานวิจัยเครื่องช่วยเดิน (Walker). มหาวิทยาลัยราชภัฏเชียงราย. https://i4biz.nrct.go.th/NRCT_PLATINUM58/home/show_product.php?research_id=2851.
- [5] นายพงศ์พิสุทธิ์ จอมวุฒิและนายสุทธิพงษ์ ช่าง. 2558. งานวิจัยเครื่องช่วยหัดเดินสำหรับผู้ป่วย (The Patient Assistant Walker). มหาวิทยาลัยนเรศวร. <https://nuir.lib.nu.ac.th/dspace/bitstream/123456789/1668/1/Pongphisut%20Jomwut.pdf>.
- [6] นพ.บุระ สันธูภากร. 2558. งานวิจัยเครื่องช่วยเดินต้านแรงโน้มถ่วงแบบอัตโนมัติ. มหาวิทยาลัยเทคโนโลยีสุรนารี. https://www.thailandtechshow.com/view_techno.php?id=477.
- [7] นายสาโรจน์ แนวบรรทัด และคณะ. 2553. อุปกรณ์ช่วยหัดเดินอัตโนมัติสำหรับผู้ป่วย. [วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ]. มหาวิทยาลัยเทคโนโลยีราชมงคลศรีวิชัย.
- [8] ร.ต.พิชิตพล โชติกุลนันท์ และคณะ. 2561. งานวิจัยรถเข็นช่วยเดินชนิดเคลื่อนที่ได้รอบทิศทาง. https://www.thailandtechshow.com/view_techno.php?id=1260.
- [9] สมพงษ์ ใจดี. 2540. ฟิสิกส์ มหาวิทยาลัย 1 (พิมพ์ครั้งที่ 2). สำนักพิมพ์ จุฬาลงกรณ์มหาวิทยาลัย.
- [10] Softonic. 2567. โปรแกรมภาษาของอาร์ดูอิโน (Arduino IDE). <https://arduino-ide.softonic-th.com/>.
- [11] Gurusnook. 2556. ภาษาซี (C Programming Language). <https://guru.Snook.com/6394/>.

- [12] Cybertice. 2560. *ESP32 NodeMCU ESP-WROOM-32*. <https://www.cybertice.com/product/1149/esp32-nodemcu-esp-wroom-32-wi-fi-and-bluetooth-module-dual-core-consumption-cp2102>.
- [13] Arduino engineer. 2560. *LCD 128x64 Dots LCD Graphic Matrix Display*. <https://www.arduitronics.com/product/486/lcd-display-module>.
- [14] นายพิชัยพร บ่มไธ้. 2560. *LCD IIC/I2C Interface*. <https://www.analogread.com/product/256>.
- [15] นายไกรสร สืบบุญ. 2560. *IR Sensor Module*. <https://www.allnewstep.com/product/371>.
- [16] Tectonics. 2561. *Switching Power Supply AC to DC*. <https://www.analogread.com/product/2213>.
- [17] Tectonics. 2559. *DC-to-DC Step Down Module*. <https://techtonics.in/product/dc-dc-7v-26v-24v-to-5v-3a-step-down>.
- [18] Code mobiles. 2562. *Micro SD Card Module*. <http://www.iot.codemobiles.com/product/222/micro-sd-card-module>.
- [19] นายสนธยา ใจผ่อง. 2560. *Relay module Arduino 5V*. <https://www.as99shop.com/product/199/5v-1-channel-relay>.
- [20] นายไกรสร สืบบุญ. 2559. *Micro switch tact switch button switch*. <https://www.allnewstep.com/product/2515>.
- [21] Mahachaielectronics. (2558). *Transistor BC547*. <https://www.allnewstep.com/product/2515>.

ภาคผนวก ก

งบประมาณของการพัฒนาสู่วิ้งสำหรับผู้ป่วยและสูงอายุ

งบประมาณของการสร้างการพัฒนาตู้วิ่งสำหรับผู้ป่วยและสูงอายุเครื่องมีราคาต้นทุนในการสร้างประมาณ 6,985 บาท ดังแสดงในตารางที่ ก-1

ตารางที่ ก-1 งบประมาณของการสร้างพัฒนาตู้วิ่งสำหรับผู้ป่วยและผู้สูงอายุ

ลำดับ	รายการ	จำนวน	ราคาต่อหน่วย (บาท)	ราคารวม (บาท)
1	LCD 128x64 Dots LCD Graphic Matrix Display	1 ชิ้น	230	230
2	LCD IIC/I2C Interface	1 ชิ้น	140	140
3	IR Sensor Module	2 ชิ้น	130	260
4	ESP32 NodeMCU ESP-WROOM-32 Wi-Fi and Bluetooth	1 ตัว	325	325
5	Switching Power Supply AC To DC	1 ชิ้น	250	250
6	DC-to-DC Step Down Module	2 ชิ้น	100	200
7	Micro SD Card Module	1 ชิ้น	30	30
8	Micro SD Card	1 ชิ้น	180	180
9	Relay module Arduino 5V	4 ชิ้น	15	60
10	CJMCU	1 ชิ้น	175	175
11	Resistor	15 ชิ้น	2	30
12	Transistor	4 ชิ้น	25	100
13	Capacitor	6 ตัว	20	120
14	ชุดสี	1ชุด	300	300
15	อุปกรณ์เก็บสาย	1ชุด	85	85
16	เครื่องตู้วิ่ง	1ชุด	4,500	4,500
รวม	6,507	6,985		

ภาคผนวก ข คู่มือแนะนำการใช้งาน

คู่มือแนะนำการใช้งาน เครื่องลู่วิ่งสำหรับผู้ป่วยและผู้สูงอายุ

วิธีใช้งาน

- 1) ทำการเปิดสวิตช์เครื่องก่อนการใช้งาน



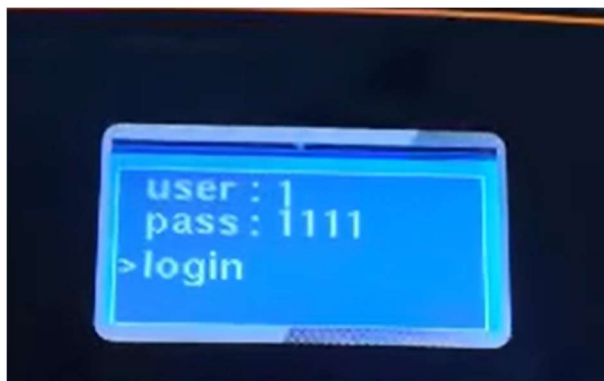
ภาพที่ ข-1 เปิดสวิตช์เครื่อง

- 2) ทำการติด Safety switch (2) ที่เสื่อของผู้ใช้งาน เลื่อนปุ่ม UP , ปุ่ม DOWN เพื่อเลื่อนเลือก User ที่ต้องการใช้งานเครื่อง



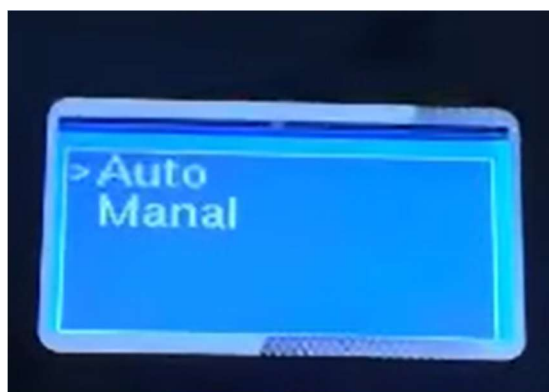
ภาพที่ ข-2 ทำการติด Safety switch

- 3) ทำการป้อนรหัสผ่านโดยกดปุ่ม Enter จำนวน 4 ครั้ง และกด Login



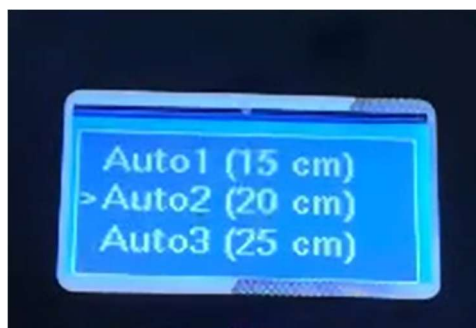
ภาพที่ ข-3 ทำการป้อนรหัสเข้าใช้งาน

- 4) การใช้งานโหมด Auto กดเลือก กด Enter



ภาพที่ ข-4 กดโหมด Auto

5) โหมด Auto จะมีให้เลือกระยะการเคลื่อนที่ของสายพาน ระยะ 15 ซม., ระยะ 20 ซม.และ ระยะ 25 ซม.



ภาพที่ ข-5 โหมด Auto เลือกระยะใช้งาน

6) เมื่อเมื่อพร้อมที่จะเริ่มการทำงานกดปุ่ม Start (6) และหากต้องการหยุดเครื่องกดปุ่ม Stop



ภาพที่ ข-6 การเริ่ม-หยุดเครื่อง โหมด Auto

7) การใช้งานโหมด Auto กดเลือก กด Enter



ภาพที่ ข-7 กดโหมด Manal

8) การใช้งานโหมด Auto กดเลือก กด Enter จากนั้น กดปุ่ม Start เริ่มใช้งานเครื่อง หากต้องการหยุดเครื่อง กดปุ่ม Stop หยุดการใช้งาน



ภาพที่ ข-8 การเริ่ม-หยุดเครื่อง โหมด Manal

ภาคผนวก ค SOURCE CODE

SOURCE CODE การทำงานเครื่อง

```

#include <U8g2lib.h> //ประกาศไลบรารีสำหรับควบคุมหน้าจอ
#include <SoftwareSerial.h> //ประกาศไลบรารีสำหรับสร้าง Serial Port เพิ่ม
#include <EEPROM.h> //ประกาศไลบรารีสำหรับอ่านและเขียนข้อมูลลง EEPROM
#ifdef U8X8_HAVE_HW_SPI //ประกาศไลบรารีสำหรับสื่อสารกับอุปกรณ์ SPI
#include <SPI.h> //ประกาศไลบรารีสำหรับสื่อสารกับอุปกรณ์ SPI
#endif

#ifdef U8X8_HAVE_HW_I2C //ประกาศไลบรารีสำหรับสื่อสารกับอุปกรณ์ I2C
#include <Wire.h> //ประกาศไลบรารีสำหรับสื่อสารกับอุปกรณ์ I2C
#endif

#define button_pluse      32 //Relay ปุ่ม Up
#define button_minus      33 //Relay ปุ่ม Down
#define button_start_stop 25 //Relay ปุ่ม Start/Stop
#define button_Moter_Out  26 //Relay Control Motor
#define IR1_left          15 //เซ็นเซอร์เท้า ซ้าย
#define IR2_right          4 //เซ็นเซอร์เท้า ขวา
#define SW_2              12 //ปุ่ม Home
#define SW_4              27 //ปุ่ม Up
#define SW_6              14 //ปุ่ม Down
#define SW_Enter          13 //ปุ่ม Enter
#define SW_Minut          21 //ปุ่ม Speed -
#define SW_Pluse          22 //ปุ่ม Speed +
#define SW_Start_Stop     36 //ปุ่ม Start/Stop
#define TIME_INTERVEL     1000 //กำหนดตัวแปร TIME_INTERVEL ให้มีค่าเท่ากับ 1000
int t_current = 0; //สร้างตัวแปร t_current ให้มีค่าเท่ากับ 0
int t_last = 0; //สร้างตัวแปร t_last ให้มีค่าเท่ากับ 0
int t_setting = 0; //สร้างตัวแปร t_setting ให้มีค่าเท่ากับ 0
unsigned int Min = 0; //สร้างตัวแปร Min ให้มีค่าเท่ากับ 0

```

```

unsigned int Sec = 0; //สร้างตัวแปร Sec ให้มีค่าเท่ากับ 0
float Speed = 0.0; //สร้างตัวแปร Speed ให้มีค่าเท่ากับ 0.0
float disten_sum = 0.0; //สร้างตัวแปร disten_sum ให้มีค่าเท่ากับ 0.0
float T = 0.0; //สร้างตัวแปร T ให้มีค่าเท่ากับ 0.0
float S = 0.0; //สร้างตัวแปร S ให้มีค่าเท่ากับ 0.0
float V = 0.0; //สร้างตัวแปร V ให้มีค่าเท่ากับ 0.0
float DISTEN_USER_1; //สร้างตัวแปร DISTEN_USER_1
float DISTEN_USER_2; //สร้างตัวแปร DISTEN_USER_2
float DISTEN_USER_3; //สร้างตัวแปร DISTEN_USER_3
float DISTEN_USER_4; //สร้างตัวแปร DISTEN_USER_4
float DISTEN_USER_5; //สร้างตัวแปร DISTEN_USER_5
float disten_u1, disten_u2, disten_u3, disten_u4, disten_u5; //สร้างตัวแปร disten_u1, dis-
ten_u2, disten_u3, disten_u4, disten_u5
String SPEED = ""; //สร้างตัวแปร SPEED
String User_data = ""; //สร้างตัวแปร User_data
String Password_data = ""; //สร้างตัวแปร Password_data
boolean state_SP = false; //สร้างตัวแปร state_SP มีค่าเป็น false
boolean User_step1 = true; //สร้างตัวแปร User_step1 มีค่าเป็น true
boolean Password_step1 = true; //สร้างตัวแปร Password_step1
boolean Login_step = true; //สร้างตัวแปร Login_step มีค่าเป็น true
boolean Auto_state = true; //สร้างตัวแปร Auto_state มีค่าเป็น true
boolean Manal_state = true; //สร้างตัวแปร Manal_state มีค่าเป็น true
boolean state_setup = true; //สร้างตัวแปร state_setup มีค่าเป็น true
boolean Auto_disten = true; //สร้างตัวแปร Auto_disten มีค่าเป็น true
boolean state_sdcard = false; //สร้างตัวแปร state_sdcard มีค่าเป็น false
boolean state_sdcard_Auto = false; //สร้างตัวแปร state_sdcard_Auto มีค่าเป็น false
int x = 0; //สร้างตัวแปร x ให้มีค่าเท่ากับ 0
int y = 0; //สร้างตัวแปร y ให้มีค่าเท่ากับ 0

```

```

int user_1 = 0; //สร้างตัวแปร user_1 ให้มีค่าเท่ากับ 0
int selec = 0; //สร้างตัวแปร selec ให้มีค่าเท่ากับ 0
int Disten_Auto = 0; //สร้างตัวแปร Disten_Auto ให้มีค่าเท่ากับ 0
int address_user1 = 0; //สร้างตัวแปร address_user1 ให้มีค่าเท่ากับ 0
int address_user2 = 7; //สร้างตัวแปร address_user2 ให้มีค่าเท่ากับ 7
int address_user3 = 14; //สร้างตัวแปร address_user3 ให้มีค่าเท่ากับ 14
int address_user4 = 21; //สร้างตัวแปร address_user4 ให้มีค่าเท่ากับ 21
int address_user5 = 28; //สร้างตัวแปร address_user5 ให้มีค่าเท่ากับ 28

U8G2_ST7920_128X64_F_SW_SPI u8g2(U8G2_R0, /* clock=*/ 18, /* data=*/ 23, /* CS=*/
5); //ตั้งค่าจอ

void setup(){
  Serial.begin(115200); //ติดต่อกับคอมพิวเตอร์ผ่าน USB
  Serial2.begin(9600); //สื่อสารระหว่างบอร์ด SD card
  EEPROM.begin(1000); //เริ่มต้นการใช้งาน EEPROM บนบอร์ด
  pinMode(SW_2, INPUT); //กำหนดให้ขา SW_2 เป็น INPUT
  pinMode(SW_4, INPUT); //กำหนดให้ขา SW_4 เป็น INPUT
  pinMode(SW_6, INPUT); //กำหนดให้ขา SW_6 เป็น INPUT
  pinMode(SW_Enter, INPUT); //กำหนดให้ขา SW_Enter เป็น INPUT
  pinMode(SW_Pluse, INPUT); //กำหนดให้ขา SW_Pluse เป็น INPUT
  pinMode(SW_Minut, INPUT); //กำหนดให้ขา SW_Minut เป็น INPUT
  pinMode(SW_Start_Stop, INPUT); //กำหนดให้ขา SW_Start_Stop เป็น INPUT
  pinMode(button_pluse, OUTPUT); //กำหนดให้ขา button_pluse เป็น OUTPUT
  pinMode(button_minus, OUTPUT); //กำหนดให้ขา button_minus เป็น OUTPUT
  pinMode(button_start_stop, OUTPUT); //กำหนดให้ขา button_start_stop เป็น OUTPUT
  pinMode(button_Moter_Out, OUTPUT); //กำหนดให้ขา button_Moter_Out เป็น OUTPUT
  digitalWrite(button_Moter_Out, HIGH); ////กำหนดให้ขา Control Motor มีสถานะเป็น HIGH
  u8g2.begin(); //เริ่มต้นการทำงานของจอ

```

```

u8g2.enableUTF8Print(); //เปิดให้รับรอง UTF-8 เพื่อพิมพ์สัญลักษณ์พิเศษ
u8g2.setFont(u8g2_font_helvB10_tf); //กำหนดฟอนต์ที่ใช้แสดงบนหน้าจอ
u8g2.setColorIndex(1); //ตั้งค่าสีสำหรับวาดบนจอวาดด้วยสีขาวบนพื้นหลังดำ
}

void loop(){
u8g2.firstPage(); //ฟังก์ชันการทำงานของหน้าจอ
do{
if(User_step1 == true){ //ถ้าตัวแปร User_step เป็น true
Userlogin(); //ให้ทำในฟังก์ชัน Userlogin
}else if(User_step1 == false && Password_step1 == true){ //หรือถ้าตัวแปร User_step1 เป็น
false และตัวแปร Password_step1 เป็น true
Password(); //ให้ทำในฟังก์ชัน Password
}else if (Password_step1 == false && Login_step == true){ //หรือถ้าตัวแปร Pass-
word_step1 เป็น false และตัวแปร Login_step เป็น true
Login(); //ให้ทำในฟังก์ชัน Login
}else if((Login_step == false) && (Auto_state == true) && (Manal_state == true)){ //หรือถ้า
ตัวแปร Login_step เป็น false และตัวแปร Auto_state เป็น true และตัวแปร Manal_state เป็น
true
Select_Mode(); //ให้ทำในฟังก์ชัน Select_Mode
}else if(Auto_state == false && Manal_state == true && Auto_disten == true){ //หรือถ้าตัว
แปร Auto_state เป็น false และตัวแปร Manal_state เป็น true และตัวแปร Auto_disten เป็น true
Auto_select(); //ให้ทำในฟังก์ชัน Auto_select
}else if(Auto_state == true && Manal_state == false){ //หรือถ้าตัวแปร Auto_state เป็น false
และตัวแปร Manal_state เป็น false
Mode_manal(); //ให้ทำในฟังก์ชัน Mode_manal
}else if(Auto_disten == false){ //หรือถ้าตัวแปร Auto_disten เป็น false
Mode_Auto(); //ให้ทำในฟังก์ชัน Mode_Auto
}
}

```

```

}while(u8g2.nextPage()); //ออกจากการทำงานของหน้าจอ
}

void Userlogin(){
u8g2.firstPage(); //ฟังก์ชันการทำงานของหน้าจอ
do{
u8g2.setFont(u8g2_font_helvB10_tf); //กำหนดฟอนต์สำหรับการแสดงผลบนหน้าจอ
u8g2.drawFrame(0,0,128,64); //วาดกรอบสี่เหลี่ยมล้อมรอบจอทั้งหมด
u8g2.setCursor(1, 13); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(">"); //แสดงข้อความ > บนหน้าจอ
u8g2.drawStr(10,13, "user"); //แสดงข้อความ user บนหน้าจอ
u8g2.setCursor(45, 13); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(":"); //แสดงข้อความ : บนหน้าจอ
u8g2.setCursor(55, 15); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(User_data); //แสดงค่าของตัวแปร User_data บนหน้าจอ
u8g2.drawStr(10,27, "pass"); //แสดงข้อความ pass บนหน้าจอ
u8g2.setCursor(45,27); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(":"); //แสดงข้อความ : บนหน้าจอ
u8g2.drawStr(10,45, "login"); //แสดงข้อความ login บนหน้าจอ
switch(x){ //นำตัวแปรของค่า x ใช้งาน
case 0: //x เป็น 0
u8g2.setCursor(55, 15); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(user_1); //แสดงค่าของตัวแปร user_1 บนหน้าจอ
break;
}
if(digitalRead(SW_4) == HIGH){ //ถ้ามีการกดปุ่ม Up
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_4) == HIGH){ //ถ้ายังมีการกดปุ่ม Up อยู่
user_1++; //กำหนดให้ตัวแปร user_1 เพิ่มขึ้นครั้งละ 1

```

```

if(user_1 > 5){ //ถ้าตัวแปร user_1 มีค่ามากกว่า 5
user_1 = 1; //กำหนดให้ตัวแปร user_1 มีค่าเท่ากับ 1
    }
}

}else if(digitalRead(SW_6) == HIGH){ //หรือถ้ามีการกดปุ่ม Down
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_6) == HIGH){ //ถ้ายังมีการกดปุ่ม Down อยู่
user_1--; //กำหนดให้ตัวแปร user_1 ลดลงครั้งละ 1
if(user_1 < 0){ //ถ้าตัวแปร user_1 มีค่าน้อยกว่า 0
user_1 = 1; //กำหนดให้ตัวแปร user_1 มีค่าเท่ากับ 1
    }
}

}else if(digitalRead(SW_Enter) == HIGH){ //หรือถ้ามีการกดปุ่ม Enter
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_Enter) == HIGH){ //ถ้ายังมีการกดปุ่ม Enter อยู่
if(x == 0){ //ถ้าตัวแปร x มีค่าเท่ากับ 0
u8g2.setCursor(55, 13); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(user_1); //แสดงค่าของตัวแปร user_1 บนหน้าจอ
User_data = "" + String(user_1); //นำค่าของตัวแปร user_1 แปลงเป็นค่าของ String แล้วนำค่าของ
ตัวแปร user_1 ไปเก็บไว้ในตัวแปร User_data
}else if(x > 1){ //หรือถ้าตัวแปร x มากกว่า 1
User_step1 = false; //กำหนดให้ตัวแปร User_step1 เท่ากับ false
    }
x++; //กำหนดให้ตัวแปร x เพิ่มขึ้นครั้งละ 1
    }
}

}while(u8g2.nextPage()); //ออกจากการทำงานของหน้าจอ
}

```

```

void Password(){
u8g2.firstPage(); //ฟังก์ชันการทำงานของหน้าจอ
do{
u8g2.drawFrame(0, 0, 128, 64); //วาดกรอบสี่เหลี่ยมล้อมรอบจอทั้งหมด
u8g2.drawStr(10, 13, "user"); //แสดงข้อความ user บนหน้าจอ
u8g2.setCursor(45, 13); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print("."); //แสดงข้อความ : บนหน้าจอ
u8g2.setCursor(55, 15); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(User_data); //แสดงค่าของตัวแปร User_data บนหน้าจอ (นำมาโชว์ในรูป 2)
u8g2.setCursor(1, 27); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(">"); //แสดงข้อความ > บนหน้าจอ
u8g2.drawStr(10, 27, "pass"); //แสดงข้อความ pass บนหน้าจอ
u8g2.setCursor(45, 27); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print("."); //แสดงข้อความ : บนหน้าจอ
u8g2.setCursor(55, 27); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print>Password_data); //แสดงค่าของตัวแปร Password_data บนหน้าจอ (ค่าที่โชว์ในรูปนี้)
u8g2.drawStr(10, 45, "login"); //แสดงข้อความ login บนหน้าจอ
switch(y){ //นำตัวแปรของค่า y ใช้งาน
case 0: //y เป็น 0
u8g2.setCursor(55, 27); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(user_1); //แสดงค่าของตัวแปร user_1 บนหน้าจอ
break;
case 1: //y เป็น 1
u8g2.setCursor(62, 27); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(user_1); //แสดงค่าของตัวแปร user_1 บนหน้าจอ
break;
case 2: //y เป็น 2

```

```

u8g2.setCursor(72, 27); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(user_1); //แสดงค่าของตัวแปร user_1 บนหน้าจอ
break;
case 3: //y เป็น 3
u8g2.setCursor(80, 27); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(user_1); //แสดงค่าของตัวแปร user_1 บนหน้าจอ
break;
}

if(digitalRead(SW_4) == HIGH){ //ถ้ามีการกดปุ่ม Up
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_4) == HIGH){ //ถ้ายังมีการกดปุ่ม Up อยู่
user_1++; //กำหนดให้ตัวแปร user_1 เพิ่มขึ้นครั้งละ 1
if(user_1 > 9){ //ถ้าตัวแปร user_1 มากกว่า 9
user_1 = 0; //กำหนดให้ตัวแปร user_1 มีค่าเท่ากับ 0
}
}
}

}else if(digitalRead(SW_6) == HIGH){ //หรือถ้ามีการกดปุ่ม Down
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_6) == HIGH){ //ถ้ายังมีการกดปุ่ม Down อยู่
user_1--; //กำหนดให้ตัวแปร user_1 ลดลงครั้งละ 1
if(user_1 < 0){ //ถ้าตัวแปร user_1 มีค่าน้อยกว่า 0
user_1 = 0; //กำหนดให้ตัวแปร user_1 มีค่าเท่ากับ 0
}
}
}

}else if(digitalRead(SW_Enter) == HIGH){ //หรือถ้ามีการกดปุ่ม Enter
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_Enter) == HIGH){ //ถ้ายังมีการกดปุ่ม Enter อยู่
switch(y){ //นำตัวแปรของค่า y ใช้งาน

```

```

case 0: //y เป็น 0
u8g2.setCursor(55, 27); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(user_1); //แสดงค่าของตัวแปร user_1 บนหน้าจอ
Password_data = "" + String(user_1); //นำค่าของตัวแปร user_1 แปลงเป็นค่าของ String แล้วนำ
ค่าของตัวแปร user_1 ไปเก็บไว้ในตัวแปร Password_data
break;
case 1: //y เป็น 1
Password_data = "" + Password_data + String(user_1); //นำค่าของตัวแปร user_1 แปลงเป็น
ค่าของ String แล้วนำค่าของตัวแปร user_1 บวกกับตัวแปร Password_data ไปเก็บไว้ในตัวแปร
Password_data
break;
case 2: //y เป็น 2
Password_data = "" + Password_data + String(user_1); //นำค่าของตัวแปร user_1 แปลงเป็นค่า
ของ String แล้วนำค่าของตัวแปร user_1 บวกกับตัวแปร Password_data ไปเก็บไว้ในตัวแปร Pass-
word_data
break;
case 3: //y เป็น 3
Password_data = "" + Password_data + String(user_1); //นำค่าของตัวแปร user_1 แปลงเป็นค่า
ของ String แล้วนำค่าของตัวแปร user_1 บวกกับตัวแปร Password_data ไปเก็บไว้ในตัวแปร Pass-
word_data
break;
    }
y++; //กำหนดให้ตัวแปร y เพิ่มขึ้นครั้ง 1
if(y > 4){ //ถ้าตัวแปร y มีค่ามากกว่า 4
Password_step1 = false; //กำหนดให้ตัวแปร Password_step1 เท่ากับ false
    }
}
}

```

```

}while(u8g2.nextPage()); //ออกจากการทำงานของหน้าจอ
}

void Login(){
u8g2.firstPage(); //ฟังก์ชันการทำงานของหน้าจอ
do{
u8g2.drawFrame(0,0,128,64); //วาดกรอบสี่เหลี่ยมล้อมรอบจอทั้งหมด
u8g2.drawStr(10,13, "user"); //แสดงข้อความ user บนหน้าจอ
u8g2.setCursor(45, 13); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(":"); //แสดงข้อความ : บนหน้าจอ
u8g2.setCursor(55, 15); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(User_data); //แสดงค่าของตัวแปร User_data บนหน้าจอ (นำมาโชว์ในรูป 2)
u8g2.drawStr(10,27, "pass"); //แสดงข้อความ pass บนหน้าจอ
u8g2.setCursor(45,27); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(":"); //แสดงข้อความ : บนหน้าจอ
u8g2.setCursor(55, 27); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print>Password_data); //แสดงค่าของตัวแปร Password_data บนหน้าจอ (ค่าที่โชว์ในรูปนี้)
u8g2.setCursor(1, 45); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(">"); //แสดงข้อความ > บนหน้าจอ
u8g2.drawStr(10,45, "login"); //แสดงข้อความ login บนหน้าจอ
if(digitalRead(SW_Enter)== HIGH){ //หรือถ้ามีการกดปุ่ม Enter
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_Enter)== HIGH){ //ถ้ายังมีการกดปุ่ม Enter อยู่ (ใช้คำสั่งเปรียบเทียบระหว่าง
User with password เพื่อเข้าสู่ เมนูเลือกโหมด Auto , Manal)
if(User_data == "1" && Password_data == "1111"){ //ถ้าตัวแปร User_data มีค่าเท่ากับ 1 และ
ตัวแปร Password_data มีค่าเท่ากับ 1111
Login_step = false; //กำหนดให้ตัวแปร Login_step เท่ากับ false
}else if(User_data == "2" && Password_data == "2222"){ //ถ้าตัวแปร User_data มีค่าเท่ากับ 2
และตัวแปร Password_data มีค่าเท่ากับ 2222

```

```

Login_step = false; //กำหนดให้ตัวแปร Login_step เท่ากับ false
}else if(User_data == "3" && Password_data == "3333"){ //ถ้าตัวแปร User_data มีค่าเท่ากับ 3
และตัวแปร Password_data มีค่าเท่ากับ 3333
Login_step = false; //กำหนดให้ตัวแปร Login_step เท่ากับ false
}else if(User_data == "4" && Password_data == "4444"){ //ถ้าตัวแปร User_data มีค่าเท่ากับ 4
และตัวแปร Password_data มีค่าเท่ากับ 4444
Login_step = false; //กำหนดให้ตัวแปร Login_step เท่ากับ false
}else if(User_data == "5" && Password_data == "5555"){ //ถ้าตัวแปร User_data มีค่าเท่ากับ 5
และตัวแปร Password_data มีค่าเท่ากับ 5555
Login_step = false; //กำหนดให้ตัวแปร Login_step เท่ากับ false
}else{
Login_step = true; //กำหนดให้ตัวแปร Login_step เท่ากับ true
User_step1 = true; //กำหนดให้ตัวแปร User_step1 เท่ากับ true
Password_step1 = true; //กำหนดให้ตัวแปร Password_step1 เท่ากับ true
Password_data = ""; //กำหนดให้ตัวแปร Password_data เป็นค่าว่าง
User_data = ""; //กำหนดให้ตัวแปร User_data เป็นค่าว่าง
user_1 = 0; x = 0; y = 0; //กำหนดให้ตัวแปร user_1, x, y มีค่าเท่ากับ 0
    }
    }
    }
}while(u8g2.nextPage()); //ออกจากการทำงานของหน้าจอ
}

void Select_Mode(){
u8g2.firstPage(); //ฟังก์ชันการทำงานของหน้าจอ
do{
u8g2.drawFrame(0,0,128,64); //วาดกรอบสี่เหลี่ยมล้อมรอบจอทั้งหมด
u8g2.drawStr(10, 13, "Auto"); //แสดงข้อความ Auto บนหน้าจอ
u8g2.drawStr(10, 27, "Manal"); //แสดงข้อความ Manal บนหน้าจอ

```

```

if(selec == 1){ //ถ้าตัวแปร selec มีค่าเท่ากับ 1
u8g2.setCursor(1, 13); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(">"); //แสดงข้อความ > บนหน้าจอ
if(digitalRead(SW_Enter) == HIGH){ //ถ้ามีการกดปุ่ม ENTER
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_Enter) == HIGH){ //ถ้ายังมีการกดปุ่ม ENTER อยู่
Auto_state = false; //กำหนดให้ตัวแปร Auto_state เป็น false
Manal_state = true; //กำหนดให้ตัวแปร Manal_state เป็น true
    }
}
}else if(selec == 2){ //หรือถ้าตัวแปร selec มีค่าเท่ากับ 2
u8g2.setCursor(1, 27); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(">"); //แสดงข้อความ > บนหน้าจอ
if(digitalRead(SW_Enter) == HIGH){ //ถ้ามีการกดปุ่ม ENTER
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_Enter) == HIGH){ //ถ้ายังมีการกดปุ่ม ENTER อยู่
Manal_state = false; //กำหนดให้ตัวแปร Manal_state เป็น false
Auto_state = true; //กำหนดให้ตัวแปร Auto_state เป็น true
    }
}
}

if(digitalRead(SW_4) == HIGH){ //ถ้ามีการกดปุ่ม Up
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_4) == HIGH){ //ถ้ายังมีการกดปุ่ม Up อยู่
selec = 1; //ให้ตัวแปร selec เท่ากับ 1
    }
}

}else if(digitalRead(SW_6) == HIGH){ //หรือถ้ามีการกดปุ่ม Down
delay(10); //หน่วงเวลา 10 ms

```

```

if(digitalRead(SW_6) == HIGH){ //ถ้ายังมีการกดปุ่ม Down อยู่
selec = 2; //ให้ตัวแปร selec เท่ากับ 2
}
}else if(digitalRead(SW_2) == HIGH){ //หรือถ้ามีการกดปุ่ม HOME
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_2) == HIGH){ //ถ้ายังมีการกดปุ่ม HOME อยู่
Login_step = true ; //กำหนดให้ตัวแปร Login_step เป็น true
Password_data = ""; //กำหนดให้ตัวแปร Password_data เป็นค่าว่าง
User_data = ""; //กำหนดให้ตัวแปร User_data เป็นค่าว่าง
user_1 = 0; x = 0; y = 0; //กำหนดให้ตัวแปร user_1, x, y มีค่าเท่ากับ 0
}
}
}while(u8g2.nextPage()); //ออกจากการทำงานของหน้าจอ
}

void Auto_select(){
u8g2.firstPage(); //ฟังก์ชันการทำงานของหน้าจอ
do{
u8g2.drawFrame(0,0,128,64); //วาดกรอบสี่เหลี่ยมล้อมรอบจอทั้งหมด
u8g2.drawStr(10, 18, "Auto1 (15 cm)"); //แสดงข้อความ Auto1 (15 cm) บนหน้าจอ
u8g2.drawStr(10, 35, "Auto2 (20 cm)"); //แสดงข้อความ Auto1 (20 cm) บนหน้าจอ
u8g2.drawStr(10, 53, "Auto3 (25 cm)"); //แสดงข้อความ Auto1 (25 cm) บนหน้าจอ
if(selec == 0){ //ถ้าตัวแปร selec มีค่าเท่ากับ 0
u8g2.setCursor(1, 18); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(">"); //แสดงข้อความ > บนหน้าจอ
if(digitalRead(SW_Enter) == HIGH){ //ถ้ามีการกดปุ่ม ENTER
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_Enter) == HIGH){ //ถ้ายังมีการกดปุ่ม ENTER อยู่
Disten_Auto = 15; //กำหนดให้ตัวแปร Disten_Auto มีค่าเท่ากับ 15

```

```

Auto_disten = false; //กำหนดให้ตัวแปร Auto_disten เป็น false
    }
}
}else if(selec == 1){ //หรือถ้าตัวแปร selec มีค่าเท่ากับ 1
u8g2.setCursor(1, 35); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(">"); //แสดงข้อความ > บนหน้าจอ
if(digitalRead(SW_Enter) == HIGH){ //ถ้ามีการกดปุ่ม ENTER
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_Enter) == HIGH){ //ถ้ายังมีการกดปุ่ม ENTER อยู่
Disten_Auto = 20; //กำหนดให้ตัวแปร Disten_Auto มีค่าเท่ากับ 20
Auto_disten = false; //กำหนดให้ตัวแปร Auto_disten เป็น false
    }
}
}else if(selec == 2){ //หรือถ้าตัวแปร selec มีค่าเท่ากับ 2
u8g2.setCursor(1, 53); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(">"); //แสดงข้อความ > บนหน้าจอ
if(digitalRead(SW_Enter) == HIGH){ //ถ้ามีการกดปุ่ม ENTER
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_Enter) == HIGH){ //ถ้ายังมีการกดปุ่ม ENTER อยู่
Disten_Auto = 25; //กำหนดให้ตัวแปร Disten_Auto มีค่าเท่ากับ 25
Auto_disten = false; //กำหนดให้ตัวแปร Auto_disten เป็น false
    }
}
}

if(selec > 2 || selec < 0){ //ถ้าตัวแปร selec มากกว่า 2 หรือ ถ้าตัวแปร selec น้อยกว่า 0
selec = 0; //ให้ตัวแปร selec เท่ากับ 0
}

```

```

if(digitalRead(SW_4) == HIGH){ //ถ้ามีการกดปุ่ม Up
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_4) == HIGH){ //ถ้ายังมีการกดปุ่ม Up อยู่
selec++; //ให้ตัวแปร selec เพิ่มขึ้น 1
    }
}else if(digitalRead(SW_6) == HIGH){ //หรือถ้ามีการกดปุ่ม Down
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_6) == HIGH){ //ถ้ายังมีการกดปุ่ม Down อยู่
selec--; //ให้ตัวแปร selec ลดลง 1
    }
}else if(digitalRead(SW_2) == HIGH){ //หรือถ้ามีการกดปุ่ม HOME
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_2) == HIGH){ //ถ้ายังมีการกดปุ่ม HOME อยู่
User_step1 = true; //กำหนดให้ตัวแปร User_step1 เป็น true
Password_step1 = true; //กำหนดให้ตัวแปร Password_step1 เป็น true
Login_step = true; //กำหนดให้ตัวแปร Login_step เป็น true
Auto_state = true ; //กำหนดให้ตัวแปร Auto_state เป็น true
Manal_state = true; //กำหนดให้ตัวแปร Manal_state เป็น true
Password_data = ""; //กำหนดให้ตัวแปร Password_data เป็นค่าว่าง
User_data = ""; //กำหนดให้ตัวแปร User_data เป็นค่าว่าง
user_1 = 0; x = 0; y = 0; //กำหนดให้ตัวแปร user_1, x, y มีค่าเท่ากับ 0
    }
    }
}while(u8g2.nextPage()); //ออกจากการทำงานของหน้าจอ
}

void Mode_manal(){
u8g2.firstPage(); //ฟังก์ชันการทำงานของหน้าจอ

```

```

do{
EEPROM.get(address_user1, DISTEN_USER_1); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user1 มาเก็บในตัวแปร DISTEN_USER_1
EEPROM.get(address_user2, DISTEN_USER_2); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user2 มาเก็บในตัวแปร DISTEN_USER_2
EEPROM.get(address_user3, DISTEN_USER_3); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user3 มาเก็บในตัวแปร DISTEN_USER_3
EEPROM.get(address_user4, DISTEN_USER_4); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user4 มาเก็บในตัวแปร DISTEN_USER_4
EEPROM.get(address_user5, DISTEN_USER_5); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user5 มาเก็บในตัวแปร DISTEN_USER_5
couter(); //ให้ทำในฟังก์ชัน couter
u8g2.drawFrame(0, 0, 128, 64); //วาดกรอบสี่เหลี่ยมล้อมรอบจอทั้งหมด
u8g2.drawStr(1,13, "user :"); //แสดงข้อความ user : บนหน้าจอ
u8g2.setCursor(55, 13); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(User_data); //แสดงค่าของตัวแปร User_data บนหน้าจอ
u8g2.drawStr(1, 40, "Time :"); //แสดงข้อความ Time : บนหน้าจอ
u8g2.setCursor(55, 40); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(Min); //แสดงค่าของตัวแปร Min บนหน้าจอ
u8g2.drawUTF8(100, 40, "min"); //แสดงข้อความ min บนหน้าจอ
u8g2.drawStr(1, 58, "speed:"); //แสดงข้อความ speed: บนหน้าจอ
u8g2.setCursor(55, 58); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(Speed, 1); //แสดงค่าของตัวแปร Speed เป็นตัวเลขทศนิยม 1 ตำแหน่งบนหน้าจอ
u8g2.drawUTF8(86, 58, "km/hr"); //แสดงข้อความ km/hr บนหน้าจอ
if(User_data == "1"){ //ถ้าตัวแปร User_data มีค่าเท่ากับ 1
u8g2.drawStr(1, 26, "disten:"); //แสดงข้อความ disten: บนหน้าจอ
u8g2.setCursor(55, 26); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
if(state_SP == true){ //ถ้าตัวแปร state_SP มีค่าเท่ากับ true

```

```

u8g2.print(DISTEN_USER_1 + S, 1); //แสดงค่าของตัวแปร DISTEN_USER_1 บวกกับค่าของตัวแปร
S เป็นตัวเลขทศนิยม 1 ตำแหน่งบนหน้าจอ
}else{
u8g2.print(DISTEN_USER_1, 1); //หรือแสดงค่าของตัวแปร DISTEN_USER_1 เป็นตัวเลขทศนิยม 1
ตำแหน่งบนหน้าจอ
    }
u8g2.drawUTF8(105, 26, "km"); //แสดงข้อความ km บนหน้าจอ
Serial.println(DISTEN_USER_1); //ปริ้นค่าของตัวแปร DISTEN_USER_1 ออกทาง Serial Monitor
}else if(User_data == "2"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 2
u8g2.drawStr(1, 26, "disten:"); //แสดงข้อความ disten: บนหน้าจอ
u8g2.setCursor(55, 26); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
if(state_SP == true){ //ถ้าตัวแปร state_SP มีค่าเท่ากับ true
u8g2.print(DISTEN_USER_2 + S, 1); //แสดงค่าของตัวแปร DISTEN_USER_2 บวกกับค่าของตัวแปร
S เป็นตัวเลขทศนิยม 1 ตำแหน่งบนหน้าจอ
}else{
u8g2.print(DISTEN_USER_2, 1); //หรือแสดงค่าของตัวแปร DISTEN_USER_2 เป็นตัวเลขทศนิยม 1
ตำแหน่งบนหน้าจอ
    }
u8g2.drawUTF8(105, 26, "km"); //แสดงข้อความ km บนหน้าจอ
Serial.println(DISTEN_USER_2); //ปริ้นค่าของตัวแปร DISTEN_USER_2 ออกทาง Serial Monitor
}else if(User_data == "3"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 3
u8g2.drawStr(1, 26, "disten:"); //แสดงข้อความ disten: บนหน้าจอ
u8g2.setCursor(55, 26); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
if(state_SP == true){ //ถ้าตัวแปร state_SP มีค่าเท่ากับ true
u8g2.print(DISTEN_USER_3 + S, 1); //แสดงค่าของตัวแปร DISTEN_USER_3 บวกกับค่าของตัวแปร
S เป็นตัวเลขทศนิยม 1 ตำแหน่งบนหน้าจอ
}else{

```

```

u8g2.print(DISTEN_USER_3, 1); //หรือแสดงค่าของตัวแปร DISTEN_USER_3 เป็นตัวเลขทศนิยม 1
ตำแหน่งบนหน้าจอ
    }
u8g2.drawUTF8(105, 26, "km"); //แสดงข้อความ km บนหน้าจอ
Serial.println(DISTEN_USER_3); //ปรี้นค่าของตัวแปร DISTEN_USER_3 ออกทาง Serial Monitor
}else if(User_data == "4"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 4
u8g2.drawStr(1, 26, "disten:"); //แสดงข้อความ disten: บนหน้าจอ
u8g2.setCursor(55, 26); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
if(state_SP == true){ //ถ้าตัวแปร state_SP มีค่าเท่ากับ true
u8g2.print(DISTEN_USER_4 + S, 1); //แสดงค่าของตัวแปร DISTEN_USER_4 บวกกับค่าของตัวแปร
S เป็นตัวเลขทศนิยม 1 ตำแหน่งบนหน้าจอ
}else{
u8g2.print(DISTEN_USER_4, 1); //หรือแสดงค่าของตัวแปร DISTEN_USER_4 เป็นตัวเลขทศนิยม 1
ตำแหน่งบนหน้าจอ
    }
u8g2.drawUTF8(105, 26, "km"); //แสดงข้อความ km บนหน้าจอ
Serial.println(DISTEN_USER_4); //ปรี้นค่าของตัวแปร DISTEN_USER_4 ออกทาง Serial Monitor
}else if(User_data == "5"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 5
u8g2.drawStr(1, 26, "disten:"); //แสดงข้อความ disten: บนหน้าจอ
u8g2.setCursor(55, 26); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
if(state_SP == true){ //ถ้าตัวแปร state_SP มีค่าเท่ากับ true
u8g2.print(DISTEN_USER_5 + S, 1); //แสดงค่าของตัวแปร DISTEN_USER_5 บวกกับค่าของตัวแปร
S เป็นตัวเลขทศนิยม 1 ตำแหน่งบนหน้าจอ
}else{
u8g2.print(DISTEN_USER_5, 1); //หรือแสดงค่าของตัวแปร DISTEN_USER_5 เป็นตัวเลขทศนิยม 1
ตำแหน่งบนหน้าจอ
    }
u8g2.drawUTF8(105, 26, "km"); //แสดงข้อความ km บนหน้าจอ

```

```

Serial.println(DISTEN_USER_5); //ปริ้นค่าของตัวแปร DISTEN_USER_5 ออกทาง Serial Monitor
}

}while(u8g2.nextPage()); //ออกจากการทำงานของหน้าจอ
}

void couter(){
digitalWrite(button_Moter_Out, HIGH); //กำหนดให้ขา Control Motor มีสถานะเป็น HIGH
if(digitalRead(SW_Pluse) == HIGH && state_SP == true){ //ถ้ามีการกดปุ่ม Speed + และตัวแปร
state_SP มีค่าเท่ากับ true
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_Pluse) == HIGH && state_SP == true){ //ถ้ายังมีการกดปุ่ม Speed + และตัว
แปร state_SP มีค่าเท่ากับ true อยู่
if(Speed == 0){ //ถ้าตัวแปร Speed มีค่าเท่ากับ 0
Speed = Speed + 0.5; //ให้ตัวแปร Speed บวก 0.5
}else if(Speed > 0){ //หรือถ้าตัวแปร Speed มีค่ามากกว่า 0
Speed = Speed + 0.1; //ให้ตัวแปร Speed บวก 0.1
}else if(Speed >= 5){ //หรือถ้าตัวแปร Speed มีค่ามากกว่าหรือเท่ากับ 5
Speed = 5; //ให้ตัวแปร Speed เท่ากับ 5
}
digitalWrite(button_pluse, HIGH); //กำหนดขา Relay ปุ่ม Up มีสถานะเป็น HIGH
delay(200); //หน่วงเวลา 200 ms
digitalWrite(button_pluse, LOW); //กำหนดขา Relay ปุ่ม Up มีสถานะเป็น LOW
}
}else if(digitalRead(SW_Minut) == HIGH && state_SP == true){ //หรือถ้ามีการกดปุ่ม Speed -
และตัวแปร state_SP มีค่าเท่ากับ true
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_Minut) == HIGH && state_SP == true){ //ถ้ายังมีการกดปุ่ม Speed - และตัว
แปร state_SP มีค่าเท่ากับ true อยู่
if(Speed > 0.5){ //ถ้าตัวแปร Speed มีค่ามากกว่า 0.5

```

```

Speed = Speed - 0.1; //ให้ตัวแปร Speed ลบ 0.1
}else if(Speed == 0.5){ //หรือถ้าตัวแปร Speed เท่ากับ 0.5
Speed = Speed - 0.5; //ให้ตัวแปร Speed ลบ 0.5
}else if(Speed < 0.0){ //หรือถ้าตัวแปร Speed น้อยกว่า 0.0
Speed = 0.5; //ให้ตัวแปร Speed เท่ากับ 0.5
}

digitalWrite(button_minus, HIGH); //กำหนดขา Relay ปุ่ม Down มีสถานะเป็น HIGH
delay(200); //หน่วงเวลา 200 ms
digitalWrite(button_minus, LOW); //กำหนดขา Relay ปุ่ม Down มีสถานะเป็น LOW
}

}else if(digitalRead(SW_Start_Stop) == HIGH ){ //หรือถ้ามีการกดปุ่ม Start/Stop
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_Start_Stop) == HIGH){ //ถ้ายังมีการกดปุ่ม Start/Stop อยู่
state_SP = !state_SP; //กลับสถานะของตัวแปร state_SP จาก false เป็น true
Speed = 0.5; //ให้ตัวแปร Speed เท่ากับ 0.5
digitalWrite(button_start_stop, HIGH); //กำหนดขา Relay Control Motor มีสถานะเป็น HIGH
delay(200); //หน่วงเวลา 200 ms
digitalWrite(button_start_stop, LOW); //กำหนดขา Relay Control Motor มีสถานะเป็น LOW
}

}else if(digitalRead(SW_2) == HIGH){ //หรือถ้ามีการกดปุ่ม Home
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_2) == HIGH){ //ถ้ายังมีการกดปุ่ม Home อยู่
User_step1 = true; //กำหนดให้ตัวแปร User_step1 เป็น true
Password_step1 = true; //กำหนดให้ตัวแปร Password_step1 เป็น true
Login_step = true; //กำหนดให้ตัวแปร Login_step เป็น true
Auto_state = true; //กำหนดให้ตัวแปร Auto_state เป็น true
Manal_state = true; //กำหนดให้ตัวแปร Manal_state เป็น true
state_SP = false; //กำหนดให้ตัวแปร state_SP เป็น false

```

```

Speed = 0.0; //กำหนดให้ตัวแปร Speed มีค่าเท่ากับ 0.0
Password_data = ""; //กำหนดให้ตัวแปร Password_data เป็นค่าว่าง
User_data = ""; //กำหนดให้ตัวแปร User_data เป็นค่าว่าง
user_1 = 0; x = 0; y = 0; //กำหนดให้ตัวแปร user_1, x, y มีค่าเท่ากับ 0
selec = 0; //กำหนดให้ตัวแปร selec มีค่าเท่ากับ 0
delay(2000); //หน่วงเวลา 2 s
    }
}

if(Speed >= 5.0){ //ถ้าตัวแปร Speed มีค่ามากกว่าหรือเท่ากับ 5.0
    Speed = 5.0; //กำหนดให้ตัวแปร Speed มีค่าเท่ากับ 5.0
}

if (Speed <= 0.0){ //ถ้าตัวแปร Speed มีค่าน้อยกว่าหรือเท่ากับ 0.0
    Speed = 0.5; //กำหนดให้ตัวแปร Speed มีค่าเท่ากับ 0.5
}

if(digitalRead(SW_4) == HIGH && digitalRead(SW_6) == HIGH){ //ถ้ามีการกดปุ่ม Up และ ปุ่ม
Down
    delay(3000); //หน่วงเวลา 3 s
    if(digitalRead(SW_4) == HIGH && digitalRead(SW_6) == HIGH){ //ถ้ายังมีการกดปุ่ม Up และ ปุ่ม
        Down อยู่
        if(User_data == "1"){ //ถ้าตัวแปร User_data มีค่าเท่ากับ 1
            DISTEN_USER_1 = 0.0; //กำหนดให้ตัวแปร DISTEN_USER_1 มีค่าเท่ากับ 0.0
            EEPROM.put(address_user1, DISTEN_USER_1); //บันทึกค่าของตัวแปร DISTEN_USER_1 ลงใน
            EEPROM ที่ตำแหน่ง address_user1
            EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
            ESP32
        }else if(User_data == "2"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 2
            DISTEN_USER_2 = 0.0; //กำหนดให้ตัวแปร DISTEN_USER_2 มีค่าเท่ากับ 0.0

```

```

EEPROM.put(address_user2, DISTEN_USER_2); //บันทึกค่าของตัวแปร DISTEN_USER_2 ลงใน
EEPROM ที่ตำแหน่ง address_user2
EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
ESP32
}else if(User_data == "3"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 3
DISTEN_USER_3 = 0.0; //กำหนดให้ตัวแปร DISTEN_USER_3 มีค่าเท่ากับ 0.0
EEPROM.put(address_user3, DISTEN_USER_3); //บันทึกค่าของตัวแปร DISTEN_USER_3 ลงใน
EEPROM ที่ตำแหน่ง address_user3
EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
ESP32
}else if(User_data == "4"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 4
DISTEN_USER_4 = 0.0; //กำหนดให้ตัวแปร DISTEN_USER_4 มีค่าเท่ากับ 0.0
EEPROM.put(address_user4, DISTEN_USER_4); //บันทึกค่าของตัวแปร DISTEN_USER_4 ลงใน
EEPROM ที่ตำแหน่ง address_user4
EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
ESP32
}else if(User_data == "5"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 5
DISTEN_USER_5 = 0.0; //กำหนดให้ตัวแปร DISTEN_USER_5 มีค่าเท่ากับ 0.0
EEPROM.put(address_user5, DISTEN_USER_5); //บันทึกค่าของตัวแปร DISTEN_USER_5 ลงใน
EEPROM ที่ตำแหน่ง address_user5
EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
ESP32
    }
    delay(1000); //หน่วงเวลา 1 s
    }
}

if(state_SP == true){ //ถ้าตัวแปร state_SP มีค่าเท่ากับ true

```

```

digitalWrite(button_Moter_Out, HIGH); //กำหนดให้ขา Relay Control Motor มีสถานะ HIGH
MILLIS(); //เรียกใช้งานฟังก์ชัน นับเวลา และ คำนวณอัตราเร็ว
state_sdcard = true; //กำหนดให้ตัวแปร state_sdcard มีค่าเท่ากับ true
}else if(state_SP == false){ //หรือถ้าตัวแปร state_SP เท่ากับ false
Serial.print("State SD CARD = "); //ปรี้นข้อความ State SD CARD = ออกทาง Serial Monitor
Serial.println(state_sdcard); //ปรี้นค่าของตัวแปร state_sdcard ออกทาง Serial Monitor
if(state_sdcard == true){ //ถ้าตัวแปร state_sdcard มีค่าเท่ากับ true
delay(2000); //หน่วงเวลา 2 s
Serial2.begin(9600); //เริ่มต้นการสื่อสารแบบอนุกรม(Serial) ที่พอร์ต Serial2 โดยกำหนด baud rate
= 9600
if(User_data == "1"){ //ถ้าตัวแปร User_data มีค่าเท่ากับ 1 (user1)
EEPROM.put(address_user1, DISTEN_USER_1 + S); //บันทึกค่าของตัวแปร DISTEN_USER_1
บวกกับค่าของตัวแปร S ลงใน EEPROM ที่ตำแหน่ง address_user1
EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
ESP32
EEPROM.get(address_user1, DISTEN_USER_1); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user1 มาเก็บในตัวแปร DISTEN_USER_1
delay(100); //หน่วงเวลา 100 ms
if(DISTEN_USER_1 < 10){ //ถ้าตัวแปร DISTEN_USER_1 มีค่าน้อยกว่า 10
Serial2.print("User1-00"); //ส่งข้อความ User1-00 ออกทางพอร์ตอนุกรม Serial2 ไปยังอุปกรณ์
ภายนอกที่เชื่อมต่อผ่าน UART (TX/RX) (ส่งไปยัง sd card)
Serial2.println(DISTEN_USER_1); //ส่งค่าของตัวแปร DISTEN_USER_1 ออกทางพอร์ตอนุกรม Se-
rial2
}else if(10 < DISTEN_USER_1 < 100){ //หรือถ้าตัวแปร DISTEN_USER_1 มีค่ามากกว่า 10 น้อย
กว่า 100
Serial2.print("User1-0"); //ส่งข้อความ User1-0 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_1); //ส่งค่าของตัวแปร DISTEN_USER_1 ออกทางพอร์ตอนุกรม Se-
rial2

```

```

}else if(DISTEN_USER_1 >= 100){ //หรือถ้าตัวแปร DISTEN_USER_1 มีค่ามากกว่าหรือเท่ากับ 100
Serial2.print("User1-"); //ส่งข้อความ User1- ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_1); //ส่งค่าของตัวแปร DISTEN_USER_1 ออกทางพอร์ตอนุกรม Serial2
}

Serial.println(DISTEN_USER_1); //ปริ้นค่าของตัวแปร DISTEN_USER_1 ออกทาง Serial Monitor
delay(2000); //หน่วงเวลา 2 s

}else if(User_data == "2"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 2 (user2)
EEPROM.put(address_user2, DISTEN_USER_2 + S); //บันทึกค่าของตัวแปร DISTEN_USER_2
บวกกับค่าของตัวแปร S ลงใน EEPROM ที่ตำแหน่ง address_user2
EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
ESP32
EEPROM.get(address_user2, DISTEN_USER_2); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง address_user2
มาเก็บในตัวแปร DISTEN_USER_2
delay(100); //หน่วงเวลา 100 ms
if(DISTEN_USER_2 < 10){ //ถ้าตัวแปร DISTEN_USER_2 มีค่าน้อยกว่า 10
Serial2.print("User2-00"); //ส่งข้อความ User2-00 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_2); //ส่งค่าของตัวแปร DISTEN_USER_2 ออกทางพอร์ตอนุกรม Serial2
}else if(10 < DISTEN_USER_2 < 100){ //หรือถ้าตัวแปร DISTEN_USER_2 มีค่ามากกว่า 10 น้อยกว่า 100
Serial2.print("User2-0"); //ส่งข้อความ User2-0 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_2); //ส่งค่าของตัวแปร DISTEN_USER_2 ออกทางพอร์ตอนุกรม Serial2
}else if(DISTEN_USER_2 >= 100){ //หรือถ้าตัวแปร DISTEN_USER_2 มีค่ามากกว่าหรือเท่ากับ 100
Serial2.print("User2-"); //ส่งข้อความ User2- ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_2); //ส่งค่าของตัวแปร DISTEN_USER_2 ออกทางพอร์ตอนุกรม Serial2
}

```

```

    }

    Serial.println(DISTEN_USER_2); //ปริ้นค่าของตัวแปร DISTEN_USER_2 ออกทาง Serial Monitor
    delay(2000); //หน่วงเวลา 2 s

    }else if(User_data == "3"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 3 (user3)
        EEPROM.put(address_user3, DISTEN_USER_3 + 5); //บันทึกค่าของตัวแปร DISTEN_USER_3
        บวกกับค่าของตัวแปร 5 ลงใน EEPROM ที่ตำแหน่ง address_user3
        EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
        ESP32

        EEPROM.get(address_user3, DISTEN_USER_3); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
        dress_user3 มาเก็บในตัวแปร DISTEN_USER_3

        delay(100); //หน่วงเวลา 100 ms

        if(DISTEN_USER_3 < 10){ //ถ้าตัวแปร DISTEN_USER_3 มีค่าน้อยกว่า 10
            Serial2.print("User3-00"); //ส่งข้อความ User3-00 ออกทางพอร์ตอนุกรม Serial2
            Serial2.println(DISTEN_USER_3); //ส่งค่าของตัวแปร DISTEN_USER_3 ออกทางพอร์ตอนุกรม Se-
            rial2

        }else if(10 < DISTEN_USER_3 < 100){ //หรือถ้าตัวแปร DISTEN_USER_3 มีค่ามากกว่า 10 น้อย
        กว่า 100

            Serial2.print("User3-0"); //ส่งข้อความ User3-0 ออกทางพอร์ตอนุกรม Serial2
            Serial2.println(DISTEN_USER_3); //ส่งค่าของตัวแปร DISTEN_USER_3 ออกทางพอร์ตอนุกรม Se-
            rial2

        }else if(DISTEN_USER_3 >= 100){ //หรือถ้าตัวแปร DISTEN_USER_3 มีค่ามากกว่าหรือเท่ากับ 100
            Serial2.print("User3-"); //ส่งข้อความ User3- ออกทางพอร์ตอนุกรม Serial2
            Serial2.println(DISTEN_USER_3); //ส่งค่าของตัวแปร DISTEN_USER_3 ออกทางพอร์ตอนุกรม Se-
            rial2

        }

        Serial.println(DISTEN_USER_3); //ปริ้นค่าของตัวแปร DISTEN_USER_3 ออกทาง Serial Monitor
        delay(2000); //หน่วงเวลา 2 s

    }else if(User_data == "4"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 4 (user4)

```

```

EEPROM.put(address_user4, DISTEN_USER_4 + S); //บันทึกค่าของตัวแปร DISTEN_USER_4
บวกกับค่าของตัวแปร S ลงใน EEPROM ที่ตำแหน่ง address_user4
EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
ESP32
EEPROM.get(address_user4, DISTEN_USER_4); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user4 มาเก็บในตัวแปร DISTEN_USER_4
delay(100); //หน่วงเวลา 100 ms
if(DISTEN_USER_4 < 10){ //ถ้าตัวแปร DISTEN_USER_4 มีค่าน้อยกว่า 10
Serial2.print("User4-00"); //ส่งข้อความ User4-00 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_4); //ส่งค่าของตัวแปร DISTEN_USER_4 ออกทางพอร์ตอนุกรม Se-
rial2
}else if(10 < DISTEN_USER_4 < 100){ //หรือถ้าตัวแปร DISTEN_USER_4 มีค่ามากกว่า 10 น้อย
กว่า 100
Serial2.print("User4-0"); //ส่งข้อความ User4-0 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_4); //ส่งค่าของตัวแปร DISTEN_USER_4 ออกทางพอร์ตอนุกรม Se-
rial2
}else if(DISTEN_USER_4 >= 100){ //หรือถ้าตัวแปร DISTEN_USER_4 มีค่ามากกว่าหรือเท่ากับ 100
Serial2.print("User4-"); //ส่งข้อความ User4- ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_4); //ส่งค่าของตัวแปร DISTEN_USER_4 ออกทางพอร์ตอนุกรม Se-
rial2
}
Serial.println(DISTEN_USER_4); //ปรี้นค่าของตัวแปร DISTEN_USER_4 ออกทาง Serial Monitor
delay(2000); //หน่วงเวลา 2 s
}else if(User_data == "5"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 5 (user5)
EEPROM.put(address_user5, DISTEN_USER_5 + S); //บันทึกค่าของตัวแปร DISTEN_USER_5
บวกกับค่าของตัวแปร S ลงใน EEPROM ที่ตำแหน่ง address_user5
EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
ESP32

```

```

EEPROM.get(address_user5, DISTEN_USER_5); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user5 มาเก็บในตัวแปร DISTEN_USER_5
delay(100); //หน่วงเวลา 100 ms
if(DISTEN_USER_5 < 10){ //ถ้าตัวแปร DISTEN_USER_5 มีค่าน้อยกว่า 10
Serial2.print("User5-00"); //ส่งข้อความ User5-00 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_5); //ส่งค่าของตัวแปร DISTEN_USER_5 ออกทางพอร์ตอนุกรม Se-
rial2
}else if(10 < DISTEN_USER_5 < 100){ //หรือถ้าตัวแปร DISTEN_USER_5 มีค่ามากกว่า 10 น้อย
กว่า 100
Serial2.print("User5-0"); //ส่งข้อความ User5-0 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_5); //ส่งค่าของตัวแปร DISTEN_USER_5 ออกทางพอร์ตอนุกรม Se-
rial2
}else if( DISTEN_USER_5 >= 100){ //หรือถ้าตัวแปร DISTEN_USER_5 มีค่ามากกว่าหรือเท่ากับ 100
Serial2.print("User5-"); //ส่งข้อความ User5- ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_5); //ส่งค่าของตัวแปร DISTEN_USER_5 ออกทางพอร์ตอนุกรม Se-
rial2
}
Serial.println(DISTEN_USER_5); //ปรี้นค่าของตัวแปร DISTEN_USER_5 ออกทาง Serial Monitor
delay(2000); //หน่วงเวลา 2 s
}

state_sdcard = false; //กำหนดให้ตัวแปร state_sdcard มีค่าเท่ากับ false
S = 0.0; //กำหนดให้ตัวแปร S มีค่าเท่ากับ 0.0
}

if(state_sdcard == false){ //ถ้าตัวแปร state_sdcard มีค่าเท่ากับ false
int t_current = 0; //กำหนดให้ตัวแปร t_current มีค่าเท่ากับ 0
int t_last = 0; //กำหนดให้ตัวแปร t_last มีค่าเท่ากับ 0
int t_setting = 0; //กำหนดให้ตัวแปร t_setting มีค่าเท่ากับ 0
Min = 0 ; //กำหนดให้ตัวแปร Min มีค่าเท่ากับ 0

```

```

Speed = 0.0 ; //กำหนดให้ตัวแปร Speed มีค่าเท่ากับ 0
    }
}
}

void Mode_Auto(){
u8g2.firstPage(); //ฟังก์ชันการทำงานของหน้าจอ
do{
EEPROM.get(address_user1, DISTEN_USER_1); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user1 มาเก็บในตัวแปร DISTEN_USER_1
EEPROM.get(address_user2, DISTEN_USER_2); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user2 มาเก็บในตัวแปร DISTEN_USER_2
EEPROM.get(address_user3, DISTEN_USER_3); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user3 มาเก็บในตัวแปร DISTEN_USER_3
EEPROM.get(address_user4, DISTEN_USER_4); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user4 มาเก็บในตัวแปร DISTEN_USER_4
EEPROM.get(address_user5, DISTEN_USER_5); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user5 มาเก็บในตัวแปร DISTEN_USER_5
u8g2.drawFrame(0,0,128,64); //วาดกรอบสี่เหลี่ยมล้อมรอบจอทั้งหมด
u8g2.drawStr(1,13, "user"); //แสดงข้อความ user : บนหน้าจอ
u8g2.setCursor(45, 13); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(":"); //แสดงข้อความ : บนหน้าจอ
u8g2.setCursor(55, 13); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(User_data); //แสดงค่าของตัวแปร User_data บนหน้าจอ
if(Disten_Auto == 15){ //ถ้าตัวแปร Disten_Auto มีค่าเท่ากับ 15
u8g2.drawStr(2, 26, "Auto1"); //แสดงข้อความ Auto1 บนหน้าจอ
u8g2.setCursor(45, 26); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(":"); //แสดงข้อความ : บนหน้าจอ
u8g2.setCursor(55, 26); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ

```

```

u8g2.print(Disten_Auto); //แสดงค่าของตัวแปร Disten_Auto บนหน้าจอ
}else if(Disten_Auto == 20){ //หรือถ้าตัวแปร Disten_Auto เท่ากับ 20
u8g2.drawStr(2, 26, "Auto2"); //แสดงข้อความ Auto2 บนหน้าจอ
u8g2.setCursor(45, 26); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(":"); //แสดงข้อความ : บนหน้าจอ
u8g2.setCursor(55, 26); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(Disten_Auto); //แสดงค่าของตัวแปร Disten_Auto บนหน้าจอ
}else if(Disten_Auto == 25){ //หรือถ้าตัวแปร Disten_Auto เท่ากับ 25
u8g2.drawStr(2, 26, "Auto3"); //แสดงข้อความ Auto3 บนหน้าจอ
u8g2.setCursor(45, 26); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(":"); //แสดงข้อความ : บนหน้าจอ
u8g2.setCursor(55, 26); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(Disten_Auto); //แสดงค่าของตัวแปร Disten_Auto บนหน้าจอ
}

u8g2.drawStr(1, 42, "disten"); //แสดงข้อความ disten บนหน้าจอ
u8g2.setCursor(45, 42); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(":"); //แสดงข้อความ : บนหน้าจอ

if(User_data == "1"){ //ถ้าตัวแปร User_data มีค่าเท่ากับ 1
u8g2.setCursor(55, 42); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print((DISTEN_USER_1) + (disten_sum/100000), 2); //แสดงค่าผลรวมของตัวแปร DIS-
TEN_USER_1 กับตัวแปร disten_sum เป็นหน่วย km โดยมีทศนิยม 2 ตำแหน่ง บนหน้าจอ
}else if(User_data == "2"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 2
u8g2.setCursor(55, 42); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print((DISTEN_USER_2) + (disten_sum/100000), 2); //แสดงค่าผลรวมของตัวแปร DIS-
TEN_USER_2 กับตัวแปร disten_sum เป็นหน่วย km โดยมีทศนิยม 2 ตำแหน่ง บนหน้าจอ
}else if(User_data == "3"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 3
u8g2.setCursor(55, 42); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ

```

```

u8g2.print((DISTEN_USER_3) + (disten_sum/100000), 2); //แสดงค่าผลรวมของตัวแปร DIS-
TEN_USER_3 กับตัวแปร disten_sum เป็นหน่วย km โดยมีทศนิยม 2 ตำแหน่ง บนหน้าจอ
}else if(User_data == "4"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 4
u8g2.setCursor(55, 42); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print((DISTEN_USER_4) + (disten_sum/100000), 2); //แสดงค่าผลรวมของตัวแปร DIS-
TEN_USER_4 กับตัวแปร disten_sum เป็นหน่วย km โดยมีทศนิยม 2 ตำแหน่ง บนหน้าจอ
}else if(User_data == "5"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 5
u8g2.setCursor(55, 42); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print((DISTEN_USER_5) + (disten_sum/100000), 2); //แสดงค่าผลรวมของตัวแปร DIS-
TEN_USER_5 กับตัวแปร disten_sum เป็นหน่วย km โดยมีทศนิยม 2 ตำแหน่ง บนหน้าจอ
}

u8g2.drawUTF8(105, 42, "km"); //แสดงข้อความ km บนหน้าจอ
u8g2.setCursor(45, 40); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(":"); //แสดงข้อความ : บนหน้าจอ
u8g2.drawStr(1, 58, "speed"); //แสดงข้อความ speed บนหน้าจอ
u8g2.setCursor(45, 58); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(":"); //แสดงข้อความ : บนหน้าจอ
u8g2.setCursor(55, 58); //กำหนดตำแหน่งสำหรับพิมพ์ข้อความ
u8g2.print(Speed, 1); //แสดงค่าของตัวแปร Speed เป็นตัวเลขทศนิยม 1 ตำแหน่งบนหน้าจอ
u8g2.drawUTF8(86, 58, "km/hr"); //แสดงข้อความ km/hr บนหน้าจอ
if(digitalRead(SW_2) == HIGH){ //ถ้ามีการกดปุ่ม Home
delay(10); //หน่วงเวลา 10 ms
if(digitalRead(SW_2) == HIGH){ //ถ้ายังมีการกดปุ่ม Home อยู่
User_step1 = true; //กำหนดให้ตัวแปร User_step1 เป็น true
Password_step1 = true; //กำหนดให้ตัวแปร Password_step1 เป็น true
Login_step = true; //กำหนดให้ตัวแปร Login_step เป็น true
Auto_state = true; //กำหนดให้ตัวแปร Auto_state เป็น true
Manal_state = true; //กำหนดให้ตัวแปร Manal_state เป็น true

```

```

Auto_disten = true ; //กำหนดให้ตัวแปร Auto_disten เป็น true
selec = 0; //กำหนดให้ตัวแปร selec มีค่าเท่ากับ 0
Disten_Auto = 0; //กำหนดให้ตัวแปร Disten_Auto มีค่าเท่ากับ 0
Password_data = ""; //กำหนดให้ตัวแปร Password_data เป็นค่าว่าง
User_data = ""; //กำหนดให้ตัวแปร User_data เป็นค่าว่าง
user_1 = 0; x = 0; y = 0; //กำหนดให้ตัวแปร user_1, x, y มีค่าเท่ากับ 0
Speed = 0.0; //กำหนดให้ตัวแปร Speed มีค่าเท่ากับ 0.0
state_SP = LOW; //กำหนดให้ตัวแปร state_SP เท่ากับ LOW
delay(2000); //หน่วงเวลา 2 s
    }
}

}else if(digitalRead (SW_Start_Stop) == HIGH){ //หรือถ้ามีการกดปุ่ม Start/Stop
delay(10); //หน่วงเวลา 10 ms
if(digitalRead (SW_Start_Stop) == HIGH){ //ถ้ายังมีการกดปุ่ม Start/Stop อยู่
state_SP = !state_SP; //กลับสถานะของตัวแปร state_SP จาก false เป็น true
Speed = 0.5 ; //กำหนดให้ตัวแปร Speed มีค่าเท่ากับ 0.5
digitalWrite(button_start_stop ,HIGH); //กำหนดให้ขา Relay Control Motor มีสถานะ HIGH
delay(200); //หน่วงเวลา 200 ms
digitalWrite(button_start_stop ,LOW); //กำหนดให้ขา Relay Control Motor มีสถานะ LOW
delay(2000); //หน่วงเวลา 2 s
    }
}

if(state_SP == true){ //ถ้าตัวแปร state_SP มีค่าเท่ากับ true
digitalWrite(button_Moter_Out, LOW); //กำหนดให้ขา Relay Control Motor มีสถานะ LOW
if(analogRead(IR1_left) == LOW && analogRead(IR2_right) == LOW){ //ถ้า IR เซ็นเซอร์ทางซ้าย
และทางขวาจับวัตถุได้
delay(10); //หน่วงเวลา 10 ms
if(analogRead(IR1_left) == LOW && analogRead(IR2_right) == LOW){ //ถ้า IR เซ็นเซอร์ทางซ้าย
และทางขวาจับวัตถุได้อยู่

```

```

if(Disten_Auto == 15){ //ถ้าตัวแปร Disten_Auto มีค่าเท่ากับ 15
digitalWrite (button_Moter_Out, HIGH ); //กำหนดให้ขา Relay Control Motor มีสถานะ HIGH
delay(1080); //หน่วงเวลา 1080 ms (0.0003*60*60)
digitalWrite (button_Moter_Out, LOW ); //กำหนดให้ขา Relay Control Motor มีสถานะ LOW
disten_sum += 15 ; //ให้ตัวแปร disten_sum บวกเพิ่มครั้งละ 15
}else if(Disten_Auto == 20){ //หรือถ้าตัวแปร Disten_Auto มีค่าเท่ากับ 20
digitalWrite (button_Moter_Out, HIGH ); //กำหนดให้ขา Relay Control Motor มีสถานะ HIGH
delay(1440); //หน่วงเวลา 1440 ms (0.0004*60*60)
digitalWrite (button_Moter_Out, LOW ); //กำหนดให้ขา Relay Control Motor มีสถานะ LOW
disten_sum += 20 ; //ให้ตัวแปร disten_sum บวกเพิ่มครั้งละ 20
    }
else if(Disten_Auto == 25){ //หรือถ้าตัวแปร Disten_Auto มีค่าเท่ากับ 25
digitalWrite (button_Moter_Out, HIGH ); //กำหนดให้ขา Relay Control Motor มีสถานะ HIGH
delay(1800); //หน่วงเวลา 1800 ms (0.0005*60*60)
digitalWrite (button_Moter_Out, LOW ); //กำหนดให้ขา Relay Control Motor มีสถานะ LOW
disten_sum += 25; //ให้ตัวแปร disten_sum บวกเพิ่มครั้งละ 25
    }
    }
    }

state_sdcard_Auto = true; //กำหนดให้ตัวแปร state_sdcard_Auto มีค่าเท่ากับ true
}else{
digitalWrite(button_Moter_Out , HIGH ); //กำหนดให้ขา Relay Control Motor มีสถานะ HIGH
Serial.print(" state_sdcard_Auto  = "); //ป้อนข้อความ state_sdcard_Auto = ออกทาง Serial
Monitor
Serial.println(state_sdcard_Auto); //ป้อนค่าของตัวแปร state_sdcard_Auto ออกทาง Serial
Monitor
if(state_sdcard_Auto == HIGH){ //ถ้าตัวแปร state_sdcard_Auto มีค่าเท่ากับ HIGH
delay(2000); //หน่วงเวลา 2 s

```

```

Serial2.begin(9600); //เริ่มต้นการสื่อสารแบบอนุกรม(Serial) ที่พอร์ต Serial2 โดยกำหนด baud rate
= 9600

if(User_data == "1"){ //ถ้าตัวแปร User_data มีค่าเท่ากับ 1
EEPROM.put(address_user1, DISTEN_USER_1 + (disten_sum / 100000)); //บันทึกค่าของตัว
แปร DISTEN_USER_1 บวกกับผลรวมของค่าของตัวแปร disten_sum ลงใน EEPROM ที่ตำแหน่ง ad-
dress_user1
EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
ESP32
EEPROM.get(address_user1, DISTEN_USER_1); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user1 มาเก็บในตัวแปร DISTEN_USER_1
delay(100); //หน่วงเวลา 100 ms
if(DISTEN_USER_1 < 10){ //ถ้าตัวแปร DISTEN_USER_1 มีค่าน้อยกว่า 10
Serial2.print("User1-00"); //ส่งข้อความ User1-00 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_1); //ส่งค่าของตัวแปร DISTEN_USER_1 ออกทางพอร์ตอนุกรม Se-
rial2
}else if(10 < DISTEN_USER_1 < 100){ //หรือถ้าตัวแปร DISTEN_USER_1 มีค่ามากกว่า 10 น้อย
กว่า 100
Serial2.print("User1-0"); //ส่งข้อความ User1-0 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_1); //ส่งค่าของตัวแปร DISTEN_USER_1 ออกทางพอร์ตอนุกรม Se-
rial2
}else if(DISTEN_USER_1 >= 100){ //หรือถ้าตัวแปร DISTEN_USER_1 มีค่ามากกว่าหรือเท่ากับ 100
Serial2.print("User1-"); //ส่งข้อความ User1- ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_1); //ส่งค่าของตัวแปร DISTEN_USER_1 ออกทางพอร์ตอนุกรม Se-
rial2
}

Serial.println(DISTEN_USER_1); //ปริ้นค่าของตัวแปร DISTEN_USER_1 ออกทาง Serial Monitor
delay(2000); //หน่วงเวลา 2 s
}else if(User_data == "2"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 2 (user2)

```

```

EEPROM.put(address_user2, DISTEN_USER_2 + (disten_sum/100000)); //บันทึกค่าของตัวแปร
DISTEN_USER_2 บวกกับผลรวมของค่าของตัวแปร disten_sum ลงใน EEPROM ที่ตำแหน่ง ad-
dress_user2
EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
ESP32
EEPROM.get(address_user2, DISTEN_USER_2); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user2 มาเก็บในตัวแปร DISTEN_USER_2
delay(100); //หน่วงเวลา 100 ms
if(DISTEN_USER_2 < 10){ //ถ้าตัวแปร DISTEN_USER_2 มีค่าน้อยกว่า 10
Serial2.print("User2-00"); //ส่งข้อความ User2-00 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_2); //ส่งค่าของตัวแปร DISTEN_USER_2 ออกทางพอร์ตอนุกรม Se-
rial2
}else if(10 < DISTEN_USER_2 < 100){ //หรือถ้าตัวแปร DISTEN_USER_2 มีค่ามากกว่า 10 น้อย
กว่า 100
Serial2.print("User2-0"); //ส่งข้อความ User2-0 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_2); //ส่งค่าของตัวแปร DISTEN_USER_2 ออกทางพอร์ตอนุกรม Se-
rial2
}else if(DISTEN_USER_2 >= 100){ //หรือถ้าตัวแปร DISTEN_USER_2 มีค่ามากกว่าหรือเท่ากับ 100
Serial2.print("User2-"); //ส่งข้อความ User2- ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_2); //ส่งค่าของตัวแปร DISTEN_USER_2 ออกทางพอร์ตอนุกรม Se-
rial2
}
Serial.println(DISTEN_USER_2); //ปริ้นค่าของตัวแปร DISTEN_USER_2 ออกทาง Serial Monitor
delay(2000); //หน่วงเวลา 2 s
}else if(User_data == "3"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 3 (user3)
EEPROM.put(address_user3, DISTEN_USER_3 + (disten_sum/100000)); //บันทึกค่าของตัวแปร
DISTEN_USER_3 บวกกับผลรวมของค่าของตัวแปร disten_sum ลงใน EEPROM ที่ตำแหน่ง ad-
dress_user3

```

```

EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
ESP32
EEPROM.get(address_user3, DISTEN_USER_3); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user3 มาเก็บในตัวแปร DISTEN_USER_3
delay(100); //หน่วงเวลา 100 ms
if(DISTEN_USER_3 < 10){ //ถ้าตัวแปร DISTEN_USER_3 มีค่าน้อยกว่า 10
Serial2.print("User3-00"); //ส่งข้อความ User3-00 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_3); //ส่งค่าของตัวแปร DISTEN_USER_3 ออกทางพอร์ตอนุกรม Se-
rial2
}else if(10 < DISTEN_USER_3 < 100){ //หรือถ้าตัวแปร DISTEN_USER_3 มีค่ามากกว่า 10 น้อย
กว่า 100
Serial2.print("User3-0"); //ส่งข้อความ User3-0 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_3); //ส่งค่าของตัวแปร DISTEN_USER_3 ออกทางพอร์ตอนุกรม Se-
rial2
}else if(DISTEN_USER_3 >= 100){ //หรือถ้าตัวแปร DISTEN_USER_3 มีค่ามากกว่าหรือเท่ากับ 100
Serial2.print("User3-"); //ส่งข้อความ User3- ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_3); //ส่งค่าของตัวแปร DISTEN_USER_3 ออกทางพอร์ตอนุกรม Se-
rial2
}
Serial.println(DISTEN_USER_3); //ปรี้นค่าของตัวแปร DISTEN_USER_3 ออกทาง Serial Monitor
delay(2000); //หน่วงเวลา 2 s
}else if(User_data == "4"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 4 (user4)
EEPROM.put(address_user4, DISTEN_USER_4 + (disten_sum/100000)); //บันทึกค่าของตัวแปร
DISTEN_USER_4 บวกกับผลรวมของค่าของตัวแปร disten_sum ลงใน EEPROM ที่ตำแหน่ง ad-
dress_user4
EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
ESP32

```

```

EEPROM.get(address_user4, DISTEN_USER_4); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user4 มาเก็บในตัวแปร DISTEN_USER_4
delay(100); //หน่วงเวลา 100 ms
if(DISTEN_USER_4 < 10){ //ถ้าตัวแปร DISTEN_USER_4 มีค่าน้อยกว่า 10
Serial2.print("User4-00"); //ส่งข้อความ User4-00 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_4); //ส่งค่าของตัวแปร DISTEN_USER_4 ออกทางพอร์ตอนุกรม Se-
rial2
}else if(10 < DISTEN_USER_4 < 100){ //หรือถ้าตัวแปร DISTEN_USER_4 มีค่ามากกว่า 10 น้อย
กว่า 100
Serial2.print("User4-0"); //ส่งข้อความ User4-0 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_4); //ส่งค่าของตัวแปร DISTEN_USER_4 ออกทางพอร์ตอนุกรม Se-
rial2
}else if(DISTEN_USER_4 >= 100){ //หรือถ้าตัวแปร DISTEN_USER_4 มีค่ามากกว่าหรือเท่ากับ 100
Serial2.print("User4-"); //ส่งข้อความ User4- ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_4); //ส่งค่าของตัวแปร DISTEN_USER_4 ออกทางพอร์ตอนุกรม Se-
rial2
}
Serial.println(DISTEN_USER_4); //ปริ้นค่าของตัวแปร DISTEN_USER_4 ออกทาง Serial Monitor
delay(2000); //หน่วงเวลา 2 s
}else if(User_data == "5"){ //หรือถ้าตัวแปร User_data มีค่าเท่ากับ 5 (user5)
EEPROM.put(address_user5, DISTEN_USER_5 + (disten_sum/100000)); //บันทึกค่าของตัวแปร
DISTEN_USER_5 บวกกับผลรวมของค่าของตัวแปร disten_sum ลงใน EEPROM ที่ตำแหน่ง ad-
dress_user5
EEPROM.commit(); //ยืนยันการบันทึกข้อมูลลง EEPROM ใช้เฉพาะในบอร์ดที่ใช้ ESP8266 หรือ
ESP32
EEPROM.get(address_user5, DISTEN_USER_5); //อ่านข้อมูลจาก EEPROM ในตำแหน่ง ad-
dress_user5 มาเก็บในตัวแปร DISTEN_USER_5
delay(100); //หน่วงเวลา 100 ms

```

```

if(DISTEN_USER_5 < 10){ //ถ้าตัวแปร DISTEN_USER_5 มีค่าน้อยกว่า 10
Serial2.print("User5-00"); //ส่งข้อความ User5-00 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_5); //ส่งค่าของตัวแปร DISTEN_USER_5 ออกทางพอร์ตอนุกรม Serial2
}else if(10 < DISTEN_USER_5 < 100){ //หรือถ้าตัวแปร DISTEN_USER_5 มีค่ามากกว่า 10 น้อยกว่า 100
Serial2.print("User5-0"); //ส่งข้อความ User5-0 ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_5); //ส่งค่าของตัวแปร DISTEN_USER_5 ออกทางพอร์ตอนุกรม Serial2
}else if(DISTEN_USER_5 >= 100){ //หรือถ้าตัวแปร DISTEN_USER_5 มีค่ามากกว่าหรือเท่ากับ 100
Serial2.print("User5-"); //ส่งข้อความ User5- ออกทางพอร์ตอนุกรม Serial2
Serial2.println(DISTEN_USER_5); //ส่งค่าของตัวแปร DISTEN_USER_5 ออกทางพอร์ตอนุกรม Serial2
}
Serial.println(DISTEN_USER_5); //ปริ้นค่าของตัวแปร DISTEN_USER_5 ออกทาง Serial Monitor
delay(2000); //หน่วงเวลา 2 s
}

state_sdcard_Auto = false; //กำหนดให้ตัวแปร state_sdcard_Auto มีค่าเท่ากับ false
}

if(state_sdcard == false{ //ถ้าตัวแปร state_sdcard มีค่าเท่ากับ false
disten_sum = 0.0; //กำหนดให้ตัวแปร disten_sum มีค่าเท่ากับ 0.0
Speed = 0.0; //กำหนดให้ตัวแปร Speed มีค่าเท่ากับ 0.0
disten_u1 = 0; //กำหนดให้ตัวแปร disten_u1 มีค่าเท่ากับ 0
disten_u2 = 0; //กำหนดให้ตัวแปร disten_u2 มีค่าเท่ากับ 0
disten_u3 = 0; //กำหนดให้ตัวแปร disten_u3 มีค่าเท่ากับ 0
disten_u4 = 0; //กำหนดให้ตัวแปร disten_u4 มีค่าเท่ากับ 0
disten_u5 = 0; //กำหนดให้ตัวแปร disten_u5 มีค่าเท่ากับ 0
} }

```

```

}while(u8g2.nextPage()); //ออกจากการทำงานของหน้าจอ
}

void MILLIS(){
t_current = millis(); //เก็บฟังก์ชัน millis ไว้ในตัวแปร t_current
if((t_current - t_last) >= (TIME_INTERVEL)){ //ถ้าตัวแปร t_current ลบกับ ตัวแปร t_last แล้วมี
ค่ามากกว่าหรือเท่ากับตัวแปร TIME_INTERVEL (กำหนดค่าไว้คือ 1000)
Sec++; //ตัวแปร Sec เพิ่มขึ้น 1
Serial.print("Time:"); //ปริ้นข้อความ Time: ออกทาง Serial Monitor
Serial.print(Min); //ปริ้นค่าของตัวแปร Min ออกทาง Serial Monitor
Serial.print(":"); //ปริ้นข้อความ : ออกทาง Serial Monitor
Serial.println(Sec); //ปริ้นค่าของตัวแปร Sec ออกทาง Serial Monitor
if(Sec >= 60){ //ถ้าตัวแปร Sec มีค่ามากกว่าหรือเท่ากับ 60
Min++; //ตัวแปร Min เพิ่มขึ้น 1
Sec = 0 ; //ตัวแปร Sec มีค่าเท่ากับ 0
}
t_last = t_current; //ให้ตัวแปร t_last เท่ากับตัวแปร t_current
}

V = Speed; //เก็บค่าของตัวแปร Speed ไว้ในตัวแปร V
Serial.print("T_min "); //ปริ้นข้อความ T_min ออกทาง Serial Monitor
Serial.println(Min); //ปริ้นค่าของตัวแปร Min ออกทาง Serial Monitor
T = Min / 60.0; //เอาค่าของตัวแปร Min หารด้วย 60 แล้วนำค่าที่ได้เก็บไว้ในตัวแปร T
Serial.print("T_min "); //ปริ้นข้อความ T_min ออกทาง Serial Monitor
Serial.println(T); //ปริ้นค่าของตัวแปร T ออกทาง Serial Monitor
S = V * T; //เอาค่าของตัวแปร V คูณกับค่าของตัวแปร T แล้วนำค่าที่ได้เก็บในตัวแปร S
}

```

SOURCE CODE การบันทึกข้อมูล

```

#include <SoftwareSerial.h> //ประกาศไลบรารี
#include <SPI.h> //ประกาศไลบรารี

```

```

#include <SD.h> //ประกาศไลบรารี

const int chipSelect = 9; //ขา CS ของ SD card module

SoftwareSerial mySerial(11, 10); //ประกาศ Serial เสริมชื่อ mySerial โดยใช้ขา D11 รับข้อมูลและ
D10 ส่งข้อมูล

File dataFile ; //ตัวแปรสำหรับเก็บไฟล์

void setup(){
  Serial.begin(9600); //เปิด Serial Monitor

  pinMode(LED_BUILTIN, OUTPUT); //กำหนดให้ขา LED_BUILTIN เป็น OUTPUT

  mySerial.begin(9600); //เปิด Software Serial สำหรับรับข้อมูล

  Serial.print("Initializing SD card..."); //ปรี้นข้อความ Initializing SD card... ออกทาง Serial Moni-
  tor

  if(!SD.begin(chipSelect)){ //ถ้าการเริ่มต้นล้มเหลว (เริ่มต้นใช้งาน SD card)

    Serial.println("Card failed, or not present"); //ปรี้นข้อความ Card failed, or not present ออก
    ทาง Serial Monitor

    }

  Serial.println("card initialized.KO"); //ปรี้นข้อความ card initialized.KO ออกทาง Serial Monitor

  digitalWrite(LED_BUILTIN, HIGH); //กำหนดให้ขา LED_BUILTIN มีสถานะเป็น HIGH

  delay(3000); //หน่วงเวลา 3 s

}

void loop(){

  if(!SD.begin(chipSelect)){ //ถ้า SD card ไม่พร้อมใช้งาน

    Serial.println("Card failed, or not present"); //ปรี้นข้อความ Card failed, or not present ออก
    ทาง Serial Monitor

    while(!SD.begin(chipSelect)){ //เมื่อ SD card ไม่พร้อมใช้งาน

      Serial.println("Card failed, or not present"); //ปรี้นข้อความ Card failed, or not present ออก
      ทาง Serial Monitor

    }

  }

}

```

```

digitalWrite(LED_BUILTIN, LOW); //กำหนดให้ขา LED_BUILTIN มีสถานะเป็น LOW
}
}

digitalWrite(LED_BUILTIN, HIGH); //กำหนดให้ขา LED_BUILTIN มีสถานะเป็น HIGH
String teststr = ""; //ประกาศตัวแปรชนิด String ชื่อ teststr และกำหนดค่าเริ่มต้นเป็นสตริงว่าง
if(mySerial.available() > 0){ //ถ้ามีข้อมูลเข้ามา
digitalWrite(LED_BUILTIN, HIGH); //กำหนดให้ขา LED_BUILTIN มีสถานะเป็น HIGH
delay(100); //หน่วงเวลา 100 ms
digitalWrite(LED_BUILTIN, LOW); //กำหนดให้ขา LED_BUILTIN มีสถานะเป็น LOW
delay(100); //หน่วงเวลา 100 ms
digitalWrite(LED_BUILTIN, HIGH); //กำหนดให้ขา LED_BUILTIN มีสถานะเป็น HIGH
teststr = mySerial.readString(); //อ่านข้อมูลทั้งหมดที่เข้ามาแล้วเก็บไว้ในตัวแปร teststr
Serial.print(teststr); //ปริ้นค่าของตัวแปร teststr ออกทาง Serial Monitor
teststr.trim(); //ลบช่องว่างด้านหน้า/ท้ายของ teststr
Serial.println(teststr.substring(0,5)); //แสดงอักขระ 5 ตัวแรกของ teststr
if(teststr.substring(0,5)== "User1"){ //ถ้าตรวจสอบว่าข้อความ 5 ตัวแรกที่รับมาคือ User1 หรือไม่
File dataFile = SD.open("User1.txt", FILE_WRITE); //เปิดไฟล์ User1.txt เพื่อเขียนข้อมูลต่อ
if(dataFile){ //ถ้าเปิดไฟล์สำเร็จ (ตรวจสอบว่าเปิดไฟล์สำเร็จหรือไม่)
dataFile.print(teststr.substring(0,5)); //เขียนข้อความ 5 ตัวแรกของ teststr ลงในไฟล์ SD ที่ชื่อว่า
dataFile
dataFile.print(","); //เขียนเครื่องหมายจุลภาค , ลงในไฟล์
dataFile.println(teststr.substring(6,12)); //เขียนข้อความจากตำแหน่งที่ 6 ถึงก่อนตำแหน่งที่ 12 ใน
teststr ลงในไฟล์
dataFile.close(); //ปิดไฟล์ที่เปิดอยู่บน SD card หลังจากเขียนข้อมูลเสร็จ
Serial.println(teststr); //ปริ้นค่าของตัวแปร teststr ออกทาง Serial Monitor
Serial.println(teststr.substring(6,12)); //แสดงข้อมูลที่บันทึกลง SD card ผ่าน Serial Monitor

```

```

}else{
  Serial.println("error opening User1.txt"); //หรือเปิดไฟล์ไม่สำเร็จ ปรี้นข้อความ error opening
  User1.txt ออกทาง Serial Monitor
  }

  else if(teststr.substring(0,5) == "User2"){ //หรือถ้าตรวจสอบว่าข้อความ 5 ตัวแรกที่รับมาคือ User2
  หรือไม่
  File dataFile = SD.open("User2.txt", FILE_WRITE); //เปิดไฟล์ User2.txt เพื่อเขียนข้อมูลต่อ
  if(dataFile){ //ถ้าเปิดไฟล์สำเร็จ
  dataFile.print(teststr.substring(0,5)); //เขียนข้อความ 5 ตัวแรกของ teststr ลงในไฟล์ SD ที่ชื่อว่า
  dataFile
  dataFile.print(","); //เขียนเครื่องหมายจุลภาค , ลงในไฟล์
  dataFile.println(teststr.substring(6,12)); //เขียนข้อความจากตำแหน่งที่ 6 ถึงก่อนตำแหน่งที่ 12 ใน
  teststr ลงในไฟล์
  dataFile.close(); //ปิดไฟล์ที่เปิดอยู่บน SD card หลังจากเขียนข้อมูลเสร็จ
  Serial.println(teststr); //ปรี้นค่าของตัวแปร teststr ออกทาง Serial Monitor
  Serial.println(teststr.substring(6,12)); //แสดงข้อมูลที่บันทึกลง SD card ผ่าน Serial Monitor
  }else{
  Serial.println("error opening User2.txt"); //หรือเปิดไฟล์ไม่สำเร็จ ปรี้นข้อความ error opening
  User2.txt ออกทาง Serial Monitor
  }

  }else if(teststr.substring(0,5) == "User3"){ //ถ้าตรวจสอบว่าข้อความ 5 ตัวแรกที่รับมาคือ User3
  หรือไม่
  File dataFile = SD.open("User3.txt", FILE_WRITE); //เปิดไฟล์ User3.txt เพื่อเขียนข้อมูลต่อ
  if(dataFile){ //ถ้าเปิดไฟล์สำเร็จ
  dataFile.print(teststr.substring(0,5)); //เขียนข้อความ 5 ตัวแรกของ teststr ลงในไฟล์ SD ที่ชื่อว่า
  dataFile

```

```

dataFile.print(","); //เขียนเครื่องหมายจุลภาค , ลงในไฟล์
dataFile.println(teststr.substring(6,12)); //เขียนข้อความจากตำแหน่งที่ 6 ถึงก่อนตำแหน่งที่ 12 ใน
teststr ลงในไฟล์
dataFile.close(); //ปิดไฟล์ที่เปิดอยู่บน SD card หลังจากเขียนข้อมูลเสร็จ
Serial.println(teststr); //ปริ้นค่าของตัวแปร teststr ออกทาง Serial Monitor
Serial.println(teststr.substring(6,12)); //แสดงข้อมูลที่บันทึกลง SD card ผ่าน Serial Monitor
}else{
Serial.println("error opening User3.txt");//หรือเปิดไฟล์ไม่สำเร็จ ปริ้นข้อความ error opening
User3.txt ออกทาง Serial Monitor
}
}else if(teststr.substring(0,5) == "User4"){ //ถ้าตรวจสอบว่าข้อความ 5 ตัวแรกที่ได้รับมาคือ User4
หรือไม่
File dataFile = SD.open("User4.txt", FILE_WRITE); //เปิดไฟล์ User4.txt เพื่อเขียนข้อมูลต่อ
if(dataFile){ //ถ้าเปิดไฟล์สำเร็จ
dataFile.print(teststr.substring(0,5)); //เขียนข้อความ 5 ตัวแรกของ teststr ลงในไฟล์ SD ที่ชื่อว่า
dataFile
dataFile.print(","); //เขียนเครื่องหมายจุลภาค , ลงในไฟล์
dataFile.println(teststr.substring(6,12)); //เขียนข้อความจากตำแหน่งที่ 6 ถึงก่อนตำแหน่งที่ 12 ใน
teststr ลงในไฟล์
dataFile.close(); //ปิดไฟล์ที่เปิดอยู่บน SD card หลังจากเขียนข้อมูลเสร็จ
Serial.println(teststr); //ปริ้นค่าของตัวแปร teststr ออกทาง Serial Monitor
Serial.println(teststr.substring(6,12)); //แสดงข้อมูลที่บันทึกลง SD card ผ่าน Serial Monitor
}else{
Serial.println("error opening User4.txt");//หรือเปิดไฟล์ไม่สำเร็จ ปริ้นข้อความ error opening
User4.txt ออกทาง Serial Monitor
}
}

```

```

}else if(teststr.substring(0,5) == "User5"){ //ถ้าตรวจสอบว่าข้อความ 5 ตัวแรกที่ได้รับมาคือ User5
หรือไม่
File dataFile = SD.open("User5.txt", FILE_WRITE); //เปิดไฟล์ User5.txt เพื่อเขียนข้อมูลต่อ
if(dataFile){ //ถ้าเปิดไฟล์สำเร็จ
dataFile.print(teststr.substring(0,5)); //เขียนข้อความ 5 ตัวแรกของ teststr ลงในไฟล์ SD ที่ชื่อว่า
dataFile
dataFile.print(","); //เขียนเครื่องหมายจุลภาค , ลงในไฟล์
dataFile.println(teststr.substring(6,12)); //เขียนข้อความจากตำแหน่งที่ 6 ถึงก่อนตำแหน่งที่ 12 ใน
teststr ลงในไฟล์
dataFile.close(); //ปิดไฟล์ที่เปิดอยู่บน SD card หลังจากเขียนข้อมูลเสร็จ
Serial.println(teststr); //ปริ้นค่าของตัวแปร teststr ออกทาง Serial Monitor
Serial.println(teststr.substring(6,12)); //แสดงข้อมูลที่บันทึกลง SD card ผ่าน Serial Monitor
}else{
Serial.println("error opening User5.txt"); //หรือเปิดไฟล์ไม่สำเร็จ ปริ้นข้อความ error opening
User5.txt ออกทาง Serial Monitor
    }
} } }

```

ประวัติผู้จัดทำปริญญานิพนธ์

ประวัติผู้แต่ง



- ชื่อ : นายธีรพงษ์ รอดทองอ่อน
- สาขา : ภาควิชาวิศวกรรมไฟฟ้า
- วัน-เดือน-ปีเกิด : วันที่ 16 พฤศจิกายน พ.ศ. 2532
- สถานที่เกิด : กรุงเทพมหานคร
- ที่อยู่ : 121/1 ม.6 ตำบลเชียรเขมา อำเภอลำปลายมาศ จังหวัด นครศรีธรรมราช
รหัสไปรษณีย์ 80190
- ประวัติการศึกษา : ระดับประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) วิทยาลัยเทคนิคนครศรีธรรมราช
- ปีสำเร็จ : พ.ศ. 2553

ประวัติผู้แต่ง



ชื่อ : นายฟาฮัด กูแหม

สาขา : ภาควิชาวิศวกรรมไฟฟ้า

วัน-เดือน-ปีเกิด : วันที่ 13 มีนาคม พ.ศ. 2540

สถานที่เกิด : นครศรีธรรมราช

ที่อยู่ : 238 ม.15 ตำบลท่าศาลา อำเภอท่าศาลา จังหวัดนครศรีธรรมราช

รหัสไปรษณีย์ 80160

ประวัติการศึกษา : ระดับประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) วิทยาลัยเทคนิคนครศรีธรรมราช

ปีสำเร็จ : พ.ศ. 2559