ระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม

ALERTING SYSTEM FOR HUNCHBACK TO PREVENT OFFICE SYNDROME

โดย

นางสาวคุณัญญา ทองศิริไทย

นางสาวบัวชมพู ทวีเดช

นางสาวปภาวดี สุขแนบ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม

ALERTING SYSTEM FOR HUNCHBACK TO PREVENT OFFICE SYNDROME

โดย

นางสาวคุณัญญา ทองศิริไทย 61010118

นางสาวบัวชมพู ทวีเดช 61010600

นางสาวปภาวดี สุขแนบ 61010621

อาจารย์ที่ปรึกษา

ผศ. มนต์ชัย แช่มช้อย

ผศ. ดร. พิชญ สุพรรณกูล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2564

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม

ALERTING SYSTEM FOR HUNCHBACK TO PREVENT OFFICE SYNDROME

ผู้จัดทำ

1. นางสาวคุณัญญา ทองศิริไทย 61010118
2. นางสาวบัวชมพู ทวีเดช 61010600
3. นางสาวปภาวดี สุขแนบ 61010621

..... **อาจารย์ที่ปรึกษา**

(ผศ. มนต์ชัย แช่มช้อย)

..... **อาจารย์ที่ปรึกษาร่วม**

(ผศ. ดร. พิชญ สุพรรณกูล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การดำเนินปริญญานิพนธ์เรื่อง “ระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม” จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความอนุเคราะห์อย่างดียิ่งจาก ผศ. มนต์ชัย แช่มช้อย อาจารย์ที่ปรึกษา และ ผศ.ดร. พิชญ สุพรรณกูล อาจารย์ที่ปรึกษาร่วม ที่กรุณาให้คำแนะนำ คำปรึกษา และแนวทางการแก้ไขปัญหาที่เป็นประโยชน์ต่อการศึกษา ค้นคว้าวิจัยให้ปริญญานิพนธ์นี้สำเร็จสมบูรณ์ยิ่งขึ้น รวมถึงสนับสนุนสถานที่ เครื่องมือ และอุปกรณ์ต่าง ๆ ที่ใช้ระหว่างการจัดปริญญานิพนธ์

ขอขอบคุณคณาจารย์และเจ้าหน้าที่ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอน ประสิทธิ์ประสาทวิชา ความรู้ และประสบการณ์ให้แก่ผู้จัดทำ

ขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่ให้ความรัก ความห่วงใย และเป็นกำลังใจที่สำคัญเสมอมาและที่สำคัญคือสนับสนุนให้โอกาสทางด้านการศึกษาอันมีค่ายิ่งแก่ผู้จัดทำ

นางสาวคุณัญญา ทองศิริไทย
นางสาวบัวชมพู ทวีเดช
นางสาวปภาวดี สุขแนบ
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม

ALERTING SYSTEM FOR HUNCHBACK TO PREVENT OFFICE SYNDROME

โดย นางสาวคุณัญญา ทองศิริไทย 61010118
นางสาวบัวชมพู ทวีเดช 61010600
นางสาวปภาวดี สุขแนบ 61010621

อาจารย์ที่ปรึกษา ผศ. มนต์ชัย แช่มช้อย
อาจารย์ที่ปรึกษาร่วม ผศ.ดร. พิชญ สุพรรณกูล

บทคัดย่อ

ในปัจจุบันได้มีการแพร่ระบาดของเชื้อโควิด-19 ทำให้คนต้องปรับเปลี่ยนการทำงานมาเป็นแบบทำงานที่บ้านอย่างหลีกเลี่ยงมิได้ ซึ่งการนั่งทำงานเป็นเวลานานเกินไปส่งผลให้เกิดโรคออฟฟิศซินโดรม (Office Syndrome) ปริญญานิพนธ์นี้เสนอระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม เป็นการออกแบบร่วมกันระหว่างฮาร์ดแวร์และซอฟต์แวร์ โดยการเขียนโปรแกรมควบคุมผ่านบอร์ดไมโครคอนโทรลเลอร์ในการวิเคราะห์การนั่งหลังค่อมตั้งแต่ระยะ 20-45 องศา ด้วยโมดูลวัดความเอียง (Gyro Module) และการจับเวลาการนั่งเป็นระยะเวลาตามที่กำหนดไว้ด้วยเซนเซอร์วัดแรงกด (Force Sensor) จากผลการทดสอบพบว่าระบบสามารถตรวจจับผู้สวมใส่ที่นั่งหลังค่อมได้ ถ้าผู้สวมใส่มีแนวโน้มนั่งหลังค่อมและนั่งเป็นเวลานานโดยไม่เปลี่ยนท่าทาง ระบบจะแจ้งเตือนด้วยเสียงและมีการสั่นด้วยมอเตอร์ เพื่อแจ้งเตือนให้ผู้ใช้เปลี่ยนท่าทางในการนั่ง ไมโครคอนโทรลเลอร์นำค่าที่ได้เก็บไว้ในฐานข้อมูลและแสดงผลผ่านเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

Nowadays, the epidemic of COVID-19 has forced people to inevitably change to work from home, where sitting for too long results in Office Syndrome. This thesis proposes alerting system for hunchback to prevent Office Syndrome that is joint design between hardware and software. The microcontroller board is programmed to analyze hunchback from 20 to 45 degrees with gyro module and to measure the sitting time for specific time with the force sensor. From the test results, it is found that the system is able to detect hunchback. If the wearer tends to sit hunched over and sit for a long time without changing the posture, the system will alert with sound and motor vibration to alert the wearer to change the posture of sitting. The microcontroller takes the values stored in the database and shows the results via a web page.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		IV
สารบัญรูป		VII
สารบัญตาราง		XVI
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของปริญญานิพนธ์	2
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**3**
	2.1 ไมโครคอนโทรลเลอร์ (MICROCONTROLLER)	3
	2.2 ชุดฐานรองขยายช่องต่อสำหรับ ARDUINO NANO 3.0 (NANO 3.0 PROTOTYPE SHIELD I/O EXTENSION BOARD)	9
	2.3 โมดูลวัดความเอียง (GY-521 MPU6050 3AXIS GYROSCOPE ACCELEROMETER SENSOR MODULE – GYRO MODULE)	10
	2.4 เซนเซอร์วัดแรงกด (FORCE SENSITIVE RESISTOR - FSR402)	12
	2.5 ลำโพง BUZZER	12
	2.6 โมดูลรีเลย์ 5 V	13
	2.7 NODEMCU	14
	2.8 โมดูลไวไฟ NODEMCU-ESP32S	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
	2.9 ฐานข้อมูล (DATABASE)	19
	2.10 ภาษาที่เกี่ยวข้อง	23
	2.11 โรคออฟฟิศซินโดรม	29
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**37**
	3.1 การออกแบบ	37
	3.2 เครื่องมือที่ใช้ในการทดลอง	90
	3.3 การจัดเก็บผลการทดลอง	91
บทที่ 4	**ผลการทดลอง**	**93**
	4.1 การทดสอบอุปกรณ์โมดูลวัดความเอียง (GYRO MODULE)	93
	4.2 การทดสอบเซนเซอร์วัดแรงกด (FORCE SENSITIVE RESISTOR - FSR402)	99
	4.3 การทดสอบอุปกรณ์โมดูลเสียงเตือน (ACTIVE BUZZER)	104
	4.4 การทดสอบต่ออุปกรณ์โมดูลวัดความเอียง (GYRO MODULE) เซนเซอร์วัดแรงกด และโมดูลเสียงเตือน (ACTIVE BUZZER) กับบอร์ดไมโครคอนโทรลเลอร์ (ARDUINO NANO)	107
	4.5 ทำการทดสอบเขียนโปรแกรมระบบการแจ้งเตือนผู้นั่งหลังค่อมและผู้ที่นั่งเกินเวลาที่กำหนด	109
	4.6 ทำการทดสอบชิ้นงาน และระบบทั้งหมด	113
	4.7 ทำการทดลองจัดเก็บข้อมูลลงในฐานข้อมูล FIREBASE	123
	4.8 ทำการสร้างหน้าเว็บเพจเพื่อแสดงผลการแจ้งเตือน	132

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**142**
	5.1 สรุปผล	142
	5.2 ข้อเสนอแนะ	142
บรรณานุกรม		**143**
ภาคผนวก ก	คำสั่งการทำงานระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม	147
ภาคผนวก ข	คำสั่งการเชื่อมต่อระบบแจ้งเตือนสำหรับผู้ที่นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรมเข้าสู่ฐานข้อมูล	153
ภาคผนวก ค	การแสดงผลบน WEB PAGE ของระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม	160

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่

2.1 ARDUINO NANO 5

2.2 PIN OUT ของ ARDUINO NANO 3.0 6

2.3 PIN OUT ของ ARDUINO NANO 3.0 7

2.4 ชุดฐานรองขยายช่องต่อสำหรับ ARDUINO NANO 3.0 9

2.5 โมดูล ACCELEROMETER/GYRO MODULE (GY-521 MPU6050) 10

2.6 เซนเซอร์วัดแรงกด 12

2.7 โมดูลเสียงเตือน (ACTIVE BUZZER) 13

2.8 โมดูลรีเลย์ 5 V 13

2.9 NODEMCU DEVKIT เวอร์ชั่น 1 15

2.10 ตำแหน่งขาของ NODEMCU DEVKIT เวอร์ชั่น 1 15

2.11 NODEMCU DEVKIT เวอร์ชั่น 2 16

2.12 ตำแหน่งขาของ NODEMCU DEVKIT เวอร์ชั่น 2 16

2.13 NODEMCU-32S 38 PINS 17

2.14 ตำแหน่งขาของ NODEMCU-32S จำนวน 38 ขา 18

2.15 ฐานข้อมูล (FIREBASE) 22

2.16 ออฟฟิศซินโดรม 30

2.17 การนั่งทำงานที่เหมาะสม 32

2.18 ลักษณะการหลังค่อม 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

3.1 บล็อกไดอะแกรมของโครงงาน 37

3.2 วงจรการเชื่อมต่ออุปกรณ์โมดูลวัดความเอียง (GYRO MODULE) กับบอร์ดไมโครคอนโทรลเลอร์ (ARDUINO NANO) 38

3.3 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ โมดูลวัดความเอียง สาย MINI USB สายจัมพ์ และโพรโทบอร์ดทดลอง ตามลำดับ 39

3.4 การต่อขาของอุปกรณ์โมดูลวัดความเอียง และบอร์ดไมโครคอนโทรลเลอร์ ตามลำดับ 40

3.5 โปรแกรมทดลองการทำงานของโมดูลวัดความเอียง (GYRO MODULE) 41

3.6 การแสดงผลของโมดูลเมื่อทำการอัพโหลด และรันโปรแกรม 42

3.7 การแสดงผลค่าแกน X Y และ Z จาก SERIAL MONITOR 43

3.8 ทำการทดลองอีกครั้งโดยทำการเอนโมดูลวัดความเอียง 43

3.9 การแสดงผลค่าแกน X Y และ Z จาก SERIAL MONITOR 44

3.10 วงจรการเชื่อมต่อเซนเซอร์วัดแรงกด (FORCE SENSITIVE RESISTOR - FSR402) กับบอร์ดไมโครคอนโทรลเลอร์ (ARDUINO NANO) 45

3.11 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์เซนเซอร์วัดแรงกด และสาย MINI USB ตามลำดับ 45

3.12 การต่อเซนเซอร์วัดแรงกด และบอร์ดไมโครคอนโทรลเลอร์ ตามลำดับ 46

3.13 โปรแกรมทดลองการทำงานของเซนเซอร์วัดแรงกด (FORCE SENSITIVE RESISTOR - FSR402) 47

3.14 การแสดงผลของเซนเซอร์เมื่อทำการอัพโหลด และรันโปรแกรม 48

3.15 การแสดงผลค่าวัดแรงกดจาก SERIAL MONITOR 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

3.16 ทำการทดลองอีกครั้งโดยทำการกดที่เซนเซอร์ 49

3.17 การแสดงผลค่าวัดแรงกดจาก SERIAL MONITOR 49

3.18 วงจรการเชื่อมต่ออุปกรณ์โมดูลเสียงเตือน (ACTIVE BUZZER) กับบอร์ดไมโครคอนโทรลเลอร์ (ARDUINO NANO) 50

3.19 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ โมดูลเสียงเตือน สาย MINI USB สายจัมพ์ และโพรโทบอร์ดทดลอง ตามลำดับ 51

3.20 การต่อขาของอุปกรณ์โมดูลเสียงเตือน และบอร์ดไมโครคอนโทรลเลอร์ตามลำดับ 52

3.21 โปรแกรมทดลองการทำงานของโมดูลเสียงเตือน (ACTIVE BUZZER) 52

3.22 การแสดงผลของโมดูลเมื่อทำการอัพโหลด และรันโปรแกรม 53

3.23 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ โมดูลเสียงเตือน โมดูลวัดความเอียง เซนเซอร์วัดแรงกดสายจัมพ์ สายMINI USB และโพรโทบอร์ดทดลองตามลำดับ 54

3.24 การบัดกรีอุปกรณ์มอเตอร์สั่นกับสายไฟ และเซนเซอร์วัดแรงกดกับสายไฟ 55

3.25 ประกอบอุปกรณ์ลงกล่อง 55

3.26 การต่ออุปกรณ์เข้ากับบอร์ดไมโครคอนโทรลเลอร์ ARDUINO NANO 56

3.27 วงจรระบบการแจ้งเตือนผู้ที่นั่งหลังค่อม และผู้ที่นั่งเกินเวลาที่กำหนด 57

3.28 โปรแกรมทดลองการทำงานโดยรวมของโมดูลวัดความเอียง (GYRO MODULE) เซนเซอร์วัดแรงกด และโมดูลเสียงเตือน (ACTIVE BUZZER) 59

3.29 เขียนโปรแกรมเพื่อทดลอง 59

3.30 ทำการอัพโหลดและรันโปรแกรม 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

3.31 ขาที่ใช้เชื่อมต่อกับ ARDUINO NANO ของ NODEMCU-32S 60

3.32 การเชื่อมต่อ ARDUINO NANO กับ NODEMCU-32S 61

3.33 โปรแกรมเชื่อมต่อ ARDUINO NANO กับ NODEMCU-32S 63

3.34 ค่าที่แสดงผ่าน SERIAL MONITOR 63

3.35 โปรแกรมการเชื่อมต่อ NODEMCU-32S กับ FIREBASE 66

3.36 ค่าที่แสดงผ่าน SERIAL MONITOR 67

3.37 ค่าต่าง ๆ ที่แสดง และเก็บข้อมูลไว้ใน FIREBASE 67

3.38 การเก็บข้อมูลผู้ใช้งาน และ NODEMCU 68

3.39 ทำการประกอบ และใส่อุปกรณ์ลงกล่อง 68

3.40 นำกล่องติดกับสายคาดหลัง 69

3.41 ทดลองสวมใส่ชิ้นงาน 69

3.42 แผนผังการทำงานของระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อม เพื่อป้องกันโรคออฟฟิศซินโดรม 71

3.43 คำสั่งการเปิดใช้งาน APP 72

3.44 คำสั่งการเชื่อมต่อระหว่างฐานข้อมูล FIREBASE กับ NODE.JS 72

3.45 คำสั่งการเชื่อมต่อฐานข้อมูล REALTIME DATABASE ในฐานข้อมูล FIREBASE 72

3.46 ข้อมูลผู้ใช้งานในฐานข้อมูล FIREBASE 73

3.47 ข้อมูลจาก NODEMCU ที่ส่งเข้ามาในฐานข้อมูล FIREBASE 74

3.48 คำสั่งเรียกใช้ข้อมูล LOG DEVICE ในฐานข้อมูล FIREBASE 74

3.49 คำสั่งการตั้งชื่อที่อยู่ของเว็บไซต์ 75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

3.50 หน้า SIGN IN เมื่อกรอกข้อมูลไม่ครบหรือกรอกข้อมูลผิด 76

3.51 การเก็บค่า ID USER ใน LOCAL STORAGE 77

3.52 การเก็บค่า GUEST ใน LOCAL STORAGE 77

3.53 หน้าต่างการลงทะเบียน 78

3.54 เมื่อ USERNAME ตรงกับ USERNAME ในฐานข้อมูล 79

3.55 LABEL เมื่อค่าที่รับมานั้นถูกกำหนดให้เป็น REQUIRED จะแสดง * สีแดงขึ้น 80

3.56 ข้อความที่แสดงขึ้นเมื่อผู้ใช้กรอกข้อมูลไม่ครบ 80

3.57 คำสั่งการประมวลผลหลังจากกด SUBMIT 81

3.58 คำสั่งการอัพเดตข้อมูลของผู้ใช้งาน 81

3.59 คำสั่งการเรียกใช้ LOADING ICON 82

3.60 ออกแบบโลโก้ และชื่อ TITLE ของ WEBSITE TITLE 82

3.61 WEBSITE TITLE 82

3.62 การออกแบบส่วนของ HEADER 83

3.63 HEAD TITLE ของ WEB PAGE 83

3.64 เมนู DROP DOWN เมื่อเข้าใช้ด้วย USER ID 83

3.65 เมนู DROP DOWN เมื่อเข้าใช้ด้วย GUEST 84

3.66 ส่วนการแสดงผล และการแบ่งหน้าการแสดงผล 85

3.67 ส่วนของหน้าการแก้ไขข้อมูล (EDIT INFORMATION) 86

3.68 หน้าการเปลี่ยนรหัสผ่าน (CHANGE PASSWORD) 86

3.69 หน้าการลบข้อมูลผู้ใช้งาน (DELETE USER) 87

3.70 คำสั่งส่วนของการลงทะเบียนเมื่อใช้ ID GUEST 88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

3.71 คำสั่งส่วนของการออกจากระบบ 88

3.72 แผนผังของระบบการทำงานการแสดงผลผ่านเว็บเพจ 89

4.1 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ โมดูลวัดความเอียง สาย MINI USB สายจัมพ์ และโพรโทบอร์ดทดลอง ตามลำดับ 93

4.2 การต่อขาของอุปกรณ์โมดูลวัดความเอียง และบอร์ดไมโครคอนโทรลเลอร์ ตามลำดับ 94

4.3 โปรแกรมทดลองการทำงานของโมดูลวัดความเอียง (GYRO MODULE) 95

4.4 การแสดงผลของโมดูลเมื่อทำการอัพโหลด และรันโปรแกรม 96

4.5 การแสดงผลค่าแกน X Y และ Z จาก SERIAL MONITOR 96

4.6 ทำการทดลองอีกครั้งโดยทำการเอนโมดูลวัดความเอียง 97

4.7 การแสดงผลค่าแกน X Y และ Z จาก SERIAL MONITOR 97

4.8 กราฟแสดงความสัมพันธ์ระหว่างค่าที่อ่านได้จากเซนเซอร์และค่ามุมที่วัดได้ 98

4.9 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์เซนเซอร์วัดแรงกด และสาย MINI USB ตามลำดับ 99

4.10 การต่อเซนเซอร์วัดแรงกด และบอร์ดไมโครคอนโทรลเลอร์ ตามลำดับ 100

4.11 โปรแกรมทดลองการทำงานของเซนเซอร์วัดแรงกด (FORCE SENSITIVE RESISTOR - FSR402) 100

4.12 การแสดงผลของเซนเซอร์เมื่อทำการอัพโหลด และรันโปรแกรม 101

4.13 การแสดงผลค่าวัดแรงกดจาก SERIAL MONITOR 101

4.14 ทำการทดลองอีกครั้งโดยทำการกดที่เซนเซอร์ 102

4.15 การแสดงผลค่าวัดแรงกดจาก SERIAL MONITOR 102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

4.16 กราฟแสดงความสัมพันธ์ระหว่างค่าที่อ่านได้จากเซนเซอร์วัดแรงกด และน้ำหนักแรงกด 104

4.17 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์โมดูลเสียงเตือน สาย MINI USB สายจัมพ์ และโพรโทบอร์ดทดลอง ตามลำดับ 105

4.18 การต่อขาของอุปกรณ์โมดูลเสียงเตือน และบอร์ดไมโครคอนโทรลเลอร์ ตามลำดับ 106

4.19 โปรแกรมทดลองการทำงานของโมดูลเสียงเตือน (ACTIVE BUZZER) 106

4.20 การแสดงผลของโมดูลเมื่อทำการอัพโหลด และรันโปรแกรม 107

4.21 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ โมดูลเสียงเตือน โมดูลวัดความเอียง เซนเซอร์วัดแรงกดสายจัมพ์ สายMINI USB และ โพรโทบอร์ดทดลอง ตามลำดับ 108

4.22 โปรแกรมทดลองการทำงานโดยรวมของโมดูลวัดความเอียง (GYRO MODULE) เซนเซอร์วัดแรงกด และโมดูลเสียงเตือน (ACTIVE BUZZER) 110

4.23 กราฟแสดงความสัมพันธ์ระหว่างค่าที่อ่านได้จากเซนเซอร์ และค่ามุมที่วัดได้ ขณะนั่งหลังค่อม 111

4.24 กราฟแสดงความสัมพันธ์ระหว่างค่าที่อ่านได้จากเซนเซอร์วัดแรงกด และเวลา 112

4.25 ทดสอบโดยให้ผู้สวมใส่นั่งหลังค่อม 113

4.26 ค่าที่แสดงบน SERIAL MONITOR 114

4.27 ทดสอบโดยให้ผู้สวมใส่นั่งเป็นระยะเวลามากกว่า 10 วินาที 115

4.28 ผู้สวมใส่เปลี่ยนอิริยาบถจากการนั่งเป็นยืน 116

4.29 ค่าที่แสดงบน SERIAL MONITOR 117

4.30 ทดสอบโดยให้ผู้สวมใส่นั่งหลังค่อม และนั่งเป็นระยะเวลามากกว่า 10 วินาที 118

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.31	ผู้สวมใส่เปลี่ยนจากนั่งหลังค่อมเป็นหลังตรง	119
4.32	ผู้สวมใส่เปลี่ยนอิริยาบถจากการนั่งเป็นยืน	119
4.33	ค่าที่แสดงบน SERIAL MONITOR	120
4.34	กราฟแสดงความสัมพันธ์ระหว่างแกน X, Y และ Z ของ GYRO MODULE เมื่อตรวจพบว่าผู้สวมใส่นั่งหลังค่อมและนั่งหลังตรง	122
4.35	เขียนโปรแกรมเพื่อทดสอบ	123
4.36	ทำการอัพโหลดและรันโปรแกรม	123
4.37	ขาที่ใช้เชื่อมต่อกับ ARDUINO NANO ของ NODEMCU-32S	124
4.38	การเชื่อมต่อ ARDUINO NANO กับ NODEMCU-32S	124
4.39	โปรแกรมเชื่อมต่อ ARDUINO NANO กับ NODEMCU-32S	126
4.40	ค่าที่แสดงผ่าน SERIAL MONITOR	127
4.41	โปรแกรมการเชื่อมต่อ NODEMCU-32S กับ FIREBASE	130
4.42	ค่าที่แสดงผ่าน SERIAL MONITOR	131
4.43	ค่าต่าง ๆ ที่แสดง และเก็บข้อมูลไว้ใน FIREBASE	131
4.44	การเก็บข้อมูลผู้ใช้งาน และ NODEMCU	132
4.45	คำสั่งการเชื่อมต่อระหว่างฐานข้อมูล FIREBASE กับ NODE.JS	132
4.46	คำสั่งการเชื่อมต่อฐานข้อมูล REALTIME DATABASE ในฐานข้อมูล FIREBASE	133
4.47	การสร้างหน้า SIGN IN	134
4.48	WEB PAGE ในหน้า SIGN IN	134
4.49	การเก็บค่า ID USER ใน LOCAL STORAGE	135

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

4.50 การเก็บค่า GUEST ใน LOCAL STORAGE 135

4.51 หน้าต่างการลงทะเบียน 136

4.52 เมื่อ USERNAME ตรงกับ USERNAME ในฐานข้อมูล 137

4.53 การแบ่งหน้าการแสดงผล 138

4.54 ส่วนของหน้าการแก้ไขข้อมูล (EDIT INFORMATION) 139

4.55 หน้าการเปลี่ยนรหัสผ่าน 139

4.56 หน้าการลบข้อมูลผู้ใช้งาน 140

4.57 คำสั่งส่วนของการลงทะเบียนเมื่อใช้ ID GUEST 140

4.58 คำสั่งส่วนของการออกจากระบบ 141

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

2.1 ข้อมูลจำเพาะบอร์ด ARDUINO NANO 5

2.2 รายละเอียดการใช้งานของบอร์ด ARDUINO NANO 3.0 7

2.3 การเชื่อมต่อระหว่าง GYRO MODULE กับบอร์ด ARDUINO 11

4.1 ค่ามุมที่วัดได้ ค่าแกน X, Y และ Z ของ GYRO MODULE และค่าความผิดพลาดของค่ามุมที่วัดได้เทียบกับค่าแกน Y จากเซนเซอร์ 98

4.2 ค่าน้ำหนักแรงกด และค่าที่อ่านได้จากเซนเซอร์วัดแรงกด 103

4.3 ค่ามุมที่วัดได้ ค่าแกน X, Y และ Z ของ GYRO MODULE และค่าความผิดพลาดของค่ามุมที่วัดได้เทียบกับค่าแกน Y จากเซนเซอร์ ขณะนั่งหลังค่อม 111

4.4 เวลา และค่าที่อ่านได้จากเซนเซอร์วัดแรงกด 112

4.5 ค่าแกน X, Y และ Z ของ GYRO MODULE เมื่อตรวจพบว่าผู้สวมใส่นั่งหลังค่อมและนั่งหลังตรง 121

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากเกิดการแพร่ระบาดของเชื้อโควิด-19 ทำให้ทุกคนต้องเปลี่ยนมาเป็นการทำงานแบบ Work From Home อย่างหลีกเลี่ยงมิได้ ซึ่งปัญหาที่ตามมาก็คือ อาการปวดเมื่อยกล้ามเนื้อ เช่น คอ บ่า ไหล่ เพราะนั่งทำงานหน้าจอคอมพิวเตอร์เป็นระยะเวลาที่นานเกินไปและไม่ได้เปลี่ยนท่า จนบางคนอาจมีอาการลุกลามเรื้อรังไปจนถึงขนาดชาตามแขนและมือ อาการเหล่านี้เรียกว่า ออฟฟิศซินโดรม (Office syndrome) ซึ่งเป็นกลุ่มอาการปวดกล้ามเนื้อ (Myofascial Pain syndrome) ที่มักจะเกิดในผู้ที่ต้องทำงานต่อเนื่องเป็นเวลานาน มีอาการปวดที่พบมากสุดคือบริเวณคอ บ่า ไหล่ ส่วนของบั้นเอว สะโพก และก้น การมีระบบที่สามารถแจ้งเตือนสำหรับผู้นั่งหลังค่อม และนั่งเป็นระยะเวลานานทำให้สามารถยับยั้งและป้องกันโรคออฟฟิศซินโดรม หากมีแนวโน้มว่ากำลังนั่งหลังค่อม ซึ่งระบบจะสามารถแจ้งเตือนด้วยเสียงและมีการสั่นด้วยมอเตอร์ เพื่อแจ้งเตือนให้ผู้สวมใส่เปลี่ยนท่าทางในการนั่ง

1.2 วัตถุประสงค์

1) เพื่อศึกษาเซนเซอร์ที่ใช้ในการตรวจจับท่าทางการนั่งเพื่อป้องกันการนั่งหลังค่อมและการนั่งนานเกินระยะเวลาที่กำหนด

2) เพื่อเสริมสร้างบุคลิกภาพและสุขภาพที่ดีแก่ผู้ใช้งาน

3) เพื่อป้องกันการเป็นโรคออฟฟิศซินโดรม

4) เพื่อศึกษาและออกแบบฐานข้อมูลของการแจ้งเตือนผู้ที่นั่งหลังค่อมและผู้ที่นั่งเกินระยะเวลาที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปริญญานิพนธ์

1) ระบบสามารถแจ้งเตือนด้วยเสียง และการสั่น หากผู้สวมใส่มีแนวโน้มว่ากำลังนั่งหลังค่อม

2) ระบบสามารถแจ้งเตือนผู้สวมใส่หากนั่งนานเกินระยะเวลาที่กำหนด

3) ระบบสามารถเก็บบันทึกข้อมูลลงฐานข้อมูลและแสดงผลผ่านเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ปริญญานิพนธ์เรื่อง “ระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม” ได้ทำการออกแบบและพัฒนาระบบจัดการเพื่อตรวจสอบและแจ้งเตือนผู้ที่มีแนวโน้มว่ากำลังนั่งหลังค่อมและนั่งนานเกินระยะเวลาที่กำหนด ทำงานผ่านไมโครคอนโทรลเลอร์โดยเชื่อมต่อกับโมดูลวัดความเอียง (Gyro Module) และเซนเซอร์วัดแรงกด (Force Sensitive Resistor) โดยจะแจ้งเตือนด้วยเสียงผ่านโมดูลเสียงเตือน (Active Buzzer) และการสั่นด้วยมอเตอร์สั่น (Motor Vibration) ซึ่งมีการเก็บข้อมูลลงในฐานข้อมูล และสามารถเรียกดูค่าต่าง ๆ ผ่านเว็บเพจ ดังนั้นปริญญานิพนธ์ที่นำเสนอจึงมีหลักการที่เกี่ยวข้องดังต่อไปนี้

2.1 ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ (Microcontroller) คือ อุปกรณ์ควบคุมขนาดเล็ก ที่มีความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียู, หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวถังเดียวกัน โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์นั้น สามารถแบ่งออกได้เป็น 5 ส่วนใหญ่ ๆ ดังต่อไปนี้

1. หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)

2. หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือหน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เช่น Flash Memory ลักษณะการทำงานของหน่วยความจำนี้ เป็นหน่วยความจำที่อ่าน-เขียนได้ด้วยไฟฟ้า เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใด ๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกับกระดาษทดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยงในการทำงานข้อมูลจะหายไปคล้ายกับหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่ว ๆ ไป แต่สำหรับ ไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยงก็ตาม ในอดีตเป็นหน่วยความจำโปรแกรมแบบ EPROM หน่วยความจำที่ลบด้วยแสง

3. ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ซึ่งถือว่าเป็นส่วนที่สำคัญมาก พอร์ตอินพุตรับสัญญาณเพื่อนำไปประมวลผลและส่งไปแสดงผลที่พอร์ตเอาต์พุต เช่น การติดสว่างของหลอดไฟ เป็นต้น

4. ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณจำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus) บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

5. วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์จะขึ้นอยู่กับการกำหนดจังหวะ หากสัญญาณนาฬิกามีความถี่สูงจังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้นมีความเร็วในการประมวลผลสูงตามไปด้วย สำหรับไมโครคอนโทรลเลอร์ที่ผู้จัดทำเลือกใช้งาน คือ Arduino

Arduino คือ โครงการที่นำชิปไอซีไมโครคอนโทรลเลอร์ตระกูลต่าง ๆ มาใช้ร่วมกันในภาษา C ซึ่งภาษา C นี้เป็นลักษณะเฉพาะ คือมีการเขียนไลบารี่ของ Arduino ขึ้นมาเพื่อให้การสั่งงานไมโครคอนโทรลเลอร์ที่แตกต่างกัน โดยตัวโครงการได้นำเสนอบอร์ดทดลองมาหลาย ๆ รูปแบบ เพื่อใช้งานกับ IDE ของตนเอง ซอฟต์แวร์ที่ใช้งานสามารถใช้ร่วมกันได้ ซึ่งในการทดลองนี้ได้เลือกใช้ Arduino Nano แสดงดังรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.1 Arduino Nano [1]

บอร์ด Arduino Nano ออกแบบมาให้มีขนาดเล็ก และใช้กับงานทั่ว ๆ ไป ใช้ชิปไอซีไมโครคอนโทรลเลอร์เบอร์ ATmega168 หรือเบอร์ ATmega328 โปรแกรมผ่านโปรโตคอล UART มีชิป USB to UART มาให้ ใช้ Mini USB เชื่อมต่อกับคอมพิวเตอร์ มีพอร์ตดิจิตอลอินพุตเอาต์พุต 14 พอร์ต มีพอร์ตอนาล็อกอินพุต 8 พอร์ต บนบอร์ดยังมีเรกกูเลเตอร์ สามารถจ่ายไฟได้ตั้งแต่ 7–12 V เพื่อให้บอร์ดทำงานได้ (จ่ายไฟที่ขา V_{IN}) กรณีมีแหล่งจ่ายไฟ 5 V อยู่สามารถจ่ายเข้าได้เลยที่ขา 5 V ซึ่งข้อมูลจำเพาะแสดงได้ดังตารางที่ 2.1

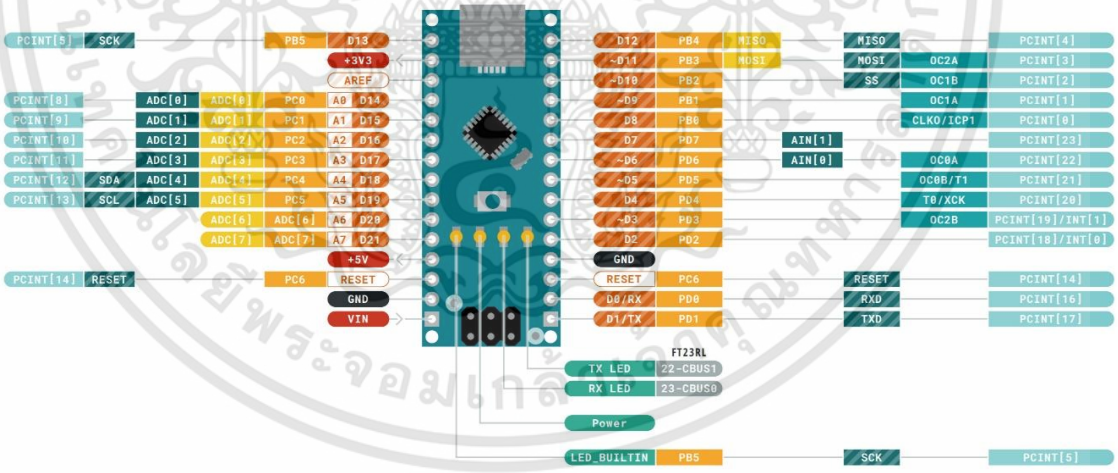
ตารางที่ 2.1 ข้อมูลจำเพาะบอร์ด Arduino Nano [1]

ชิปไอซีไมโครคอนโทรลเลอร์	ATmega168 หรือ ATmega328
ใช้แรงดันไฟฟ้า	5 V
รองรับการจ่ายแรงดันไฟฟ้า	7 – 12 V
รองรับการจ่ายแรงดันไฟฟ้า (ที่จำกัด)	6 – 20 V
พอร์ต Digital I/O	14 พอร์ต (มี 6 พอร์ต PWM output)
พอร์ต Analog Input	6 พอร์ต
กระแสไฟที่จ่ายได้ในแต่ละพอร์ต	40 mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

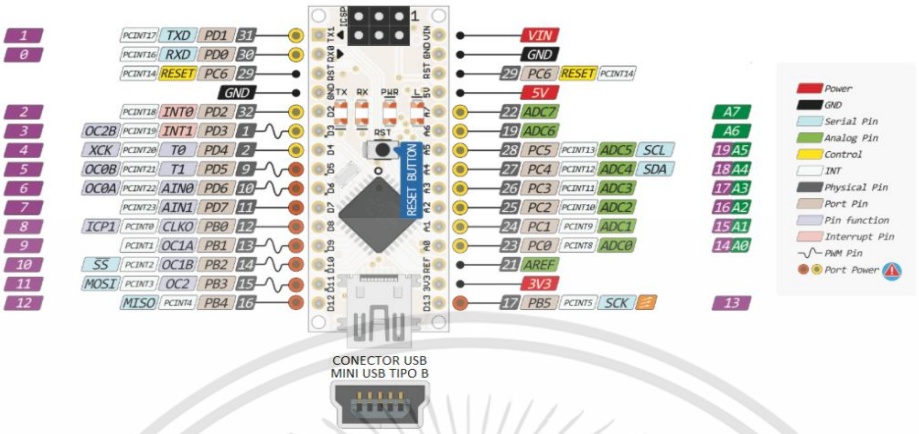
ตารางที่ 2.1 ข้อมูลจำเพาะบอร์ด Arduino Nano (ต่อ) [1]

กระแสที่จ่ายได้ในพอร์ต 3.3 V	50 mA
พื้นที่โปรแกรมภายใน	16 KB หรือ 32 KB พื้นที่โปรแกรม และ 500 B ใช้โดย Bootloader
พื้นที่แรม	1 หรือ 2 KB
พื้นที่หน่วยความจำถาวร (EEPROM)	512 B หรือ 1 KB
ความถี่คริสตัล	16 MHz
ขนาด	45 x 18 mm.
น้ำหนัก	5 g.



รูปที่ 2.2 Pin out ของ Arduino nano 3.0 [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 Pin out ของ Arduino nano 3.0 [2]

จากรูปที่ 2.2 และ 2.3 รายละเอียดการใช้งานของบอร์ด Arduino Nano 3.0 ได้แสดงไว้ในตารางที่ 2.2

ตารางที่ 2.2 รายละเอียดการใช้งานของบอร์ด Arduino Nano 3.0 [2]

Pin Category	ชื่อ Pin	รายละเอียด
Power	Vin 5V 3.3V และ GND	V_{in}: Nano 3.0 ขาใช้งานนี้สำหรับ ไฟเข้าสำหรับจ่ายไฟให้บอร์ด Arduino โดยใช้ไฟจากภายนอกไม่ผ่านช่อง USB รับไฟได้ช่วง 6-12 V 5 V: Nano 3.0 ขาใช้งานนี้คือ ขาไฟออก 5 V สำหรับจ่าย ให้อุปกรณ์ Arduino เซนเซอร์ต่าง ๆ 3.3 V: ขาไฟออก 3.3 V จ่ายกระแสได้สูงสุดเพียง 50 mA GND: ขากราวน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

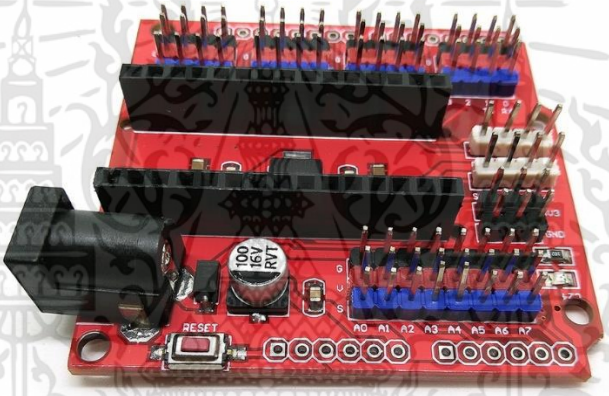
ตารางที่ 2.2 รายละเอียดการใช้งานของบอร์ด Arduino Nano 3.0 (ต่อ) [2]

Pin Category	ชื่อ Pin	รายละเอียด
Analog Pins	A0 – A7	Nano 3.0 ขาใช้งานแบบ Analog สำหรับรับค่าแบบ Analog เป็นไฟในช่วง 0-5V
Input/Output Pins	Digital Pins D0–D13	Nano 3.0 ขาใช้งาน แบบ Digital สามารถเป็นได้ทั้งแบบ Input และ Output 0V (low) และ 5 V (high)
Serial	Rx และ Tx	Nano 3.0 ขาใช้งาน สำหรับรับและส่งข้อมูลสื่อสารแบบ Serial TTL
External Interrupts	2 และ 3	Nano 3.0 ขานี้ใช้ฟังก์ชั่น Interrupt ได้
PWM	3, 5, 6, 9 และ 11	Nano 3.0 ขาสำหรับเอาต์พุตแบบ PWM ความละเอียด 8 บิต หรือช่วงเลข 0-255
SPI	10 (SS) 11 (MOSI) 12 (MISO) และ 13 (SCK)	ใช้สำหรับสื่อสารแบบ SPI
Inbuilt LED	13	ขานี้มี LED ที่ติดตั้งมาให้ในบอร์ด ให้เราทดลองได้รวดเร็ว
IIC	A4 (SDA) และA5 (SCA)	ใช้สำหรับสื่อสารแบบ I2C
AREF	AREF	ใช้เป็นแรงดันไฟฟ้าอ้างอิง หรับไฟอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ชุดฐานรองขยายช่องต่อสำหรับ Arduino Nano 3.0 (Nano 3.0 Prototype Shield I/O Extension Board) [3]

จากรูปที่ 2.4 ชุดฐานรองนี้เป็นฐานรองที่สามารถนำมาใช้แก้ปัญหาการต่อสายไฟยุ่งยากบนโพรโทรบอร์ดได้ โดยตัวฐานรองนี้จะจัดกลุ่มช่องต่อกับเซนเซอร์ต่าง ๆ อย่างเป็นระเบียบ และมีช่องสำหรับต่อไฟเลี้ยงแยกกันทุกจุด นอกจากนี้ตัวบอร์ดยังมาพร้อมกับช่องปลั๊ก DC ทำให้สามารถเสียบสายไฟเลี้ยงช่วง 7 – 12 V เข้ามาที่ช่องปลั๊ก DC ได้เลย และบอร์ดได้ออกแบบให้มีช่องบัดกรี Pin Header เพิ่ม เพื่อให้สามารถนำ Shield ของ Arduino Uno มาต่อเพิ่มได้



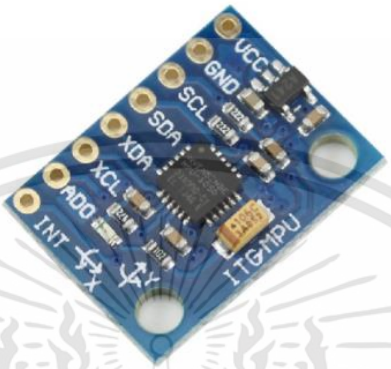
รูปที่ 2.4 ชุดฐานรองขยายช่องต่อสำหรับ Arduino Nano 3.0 [3]

รายละเอียด

- ใช้ขยายช่องใช้งานสำหรับบอร์ด Arduino Nano 3.0
- มีช่องต่อไฟแยกในแต่ละจุด
- ขนาดเท่าบอร์ด Arduino Uno และสามารถนำ Pin Header มาบัดกรีเพิ่ม เพื่อให้สามารถเสียบ Shield ของ Arduino Uno ได้
- มีปลั๊ก DC รองรับแรงดัน 7 – 12 V จ่ายไฟเลี้ยงให้กับทั้งบอร์ด
- ขนาด 60 x 35 x 13 mm.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 โมดูลวัดความเอียง (GY-521 MPU6050 3Axis Gyroscope Accelerometer Sensor Module – Gyro Module)



รูปที่ 2.5 โมดูล Accelerometer/Gyro Module (GY-521 MPU6050) [4]

- GY-521 MPU6050 3 Axis Gyroscope Accelerometer Sensor Module [5]

สามารถวัดความเอียงได้แบบ 3 ทิศทาง ความละเอียดในการวัด 16 บิต ใช้การเชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่าน I2C โมดูลนี้ได้รวมเอาเซนเซอร์ Gyroscope และ Accelerometer เข้าไว้ด้วยกัน

- Accelerometer [5]

ใช้สำหรับวัดความเร่ง หรือความเอียง เมื่อเราจับเซนเซอร์วางขนานกับพื้นโลก ค่าที่ได้จาก Accelerometer ในแกน z จะประมาณ 10 เนื่องจากถูกแรงโน้มถ่วงของโลกดึงไว้ แต่เมื่อเราเอียงเซนเซอร์ไปในทิศทางต่าง ๆ ค่าในแต่ละแกนจะเปลี่ยนไป เช่น เอียงเซนเซอร์ไปทางซ้ายที่ 90 องศา และหยุดนิ่ง ค่าที่ได้แกน z จะเป็น 0 แต่แกน x หรือ y จะเป็น 10 เนื่องจากแรงโน้มถ่วงดึงให้ค่าในแกน x หรือ y ลงด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Gyroscope [5]

ใช้วัดความเร่งในการเคลื่อนที่ ความสามารถของ Accelerometer จะให้ค่าที่คงที่อยู่เสมอ แตกต่างจาก Gyroscope ที่จะให้ค่าเมื่อมีการเคลื่อนที่เท่านั้น หน่วยวัดเป็น องศาเรเดียนต่อวินาที (rad/s)

การใช้งานวัดความเอียง การหมุนต่าง ๆ นิยมใช้งานทั้ง Accelerometer และ Gyroscope คู่กัน เนื่องจาก Accelerometer ไม่สามารถวัดการหมุนในทิศทางที่เซนเซอร์วางขนานกับพื้นโลกได้ (เมื่อหมุนเซนเซอร์ที่วางขนานกับพื้นโลก ค่า x y และ z จะไม่เปลี่ยนแปลง) จึงจำเป็นต้องใช้ Gyroscope มาใช้วัดร่วมด้วย

จากรูปที่ 2.5 เซนเซอร์นี้ สามารถนำไปประยุกต์ใช้ร่วมกับโดรนเพื่อควบคุมความสมดุล หรือนำไปใช้กับ บาลานซ์โรบอท เพื่อให้หุ่นยนต์สามารถตั้งตรงได้อยู่เสมอ นอกจากนี้ยังนำไปวัดความลาดเอียงเพื่อประยุกต์ใช้อื่น ๆ เช่น นำไปติดกับไม้เท้าเพื่อตรวจจับการล้มของผู้สูงอายุ เป็นต้น ซึ่งการเชื่อมต่อกันระหว่าง Gyro Module กับบอร์ด Arduino แสดงได้ดังตารางที่ 2.3

ตารางที่ 2.3 การเชื่อมต่อระหว่าง Gyro Module กับบอร์ด Arduino [4]

	MPU6050	Arduino
ขาที่ใช้เชื่อมต่อกัน	VCC	5V
	GND	GND
	SCL	A5
	SDA	A4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 เซนเซอร์วัดแรงกด (Force Sensitive Resistor - FSR402)

รูปที่ 2.6 เซนเซอร์วัดแรงกด [6]

ตัวต้านทานแปรผันแบบแรงกด (Force Sensitive Resistor) ดังรูปที่ 2.6 ขนาด 0.5 นิ้ว รุ่น FSR402 หรือ Force Sensitive Resistor (FSR402) เป็นเซนเซอร์วัดแรงกดในช่วง 0.2 N-20 N ค่าที่วัดได้จะเป็นแบบ Analog 0-5 V_{DC} (0-1023) ใช้ไฟเลี้ยง 5VDC FSR402 จึงเหมาะสำหรับงานที่ต้องการเซนเซอร์วัดแรงกดได้หลายระดับ

2.5 ลำโพง Buzzer

ลำโพง Buzzer เป็นอุปกรณ์ที่ให้กำเนิดเสียงทำหน้าที่เปลี่ยนสัญญาณไฟฟ้าให้อยู่ในรูปสัญญาณเสียง ลำโพง Buzzer มีอยู่ 2 ประเภท ได้แก่

1. แบบแอคทีฟ (Active buzzer) ลำโพงชนิดนี้มีวงจรกำเนิดความถี่อยู่ภายใน สามารถสร้างสัญญาณเสียง เตือนได้ทันทีเมื่อจ่ายแรงดันไฟฟ้าเข้าไป

2. แบบพาสซีฟ (Passive buzzer) ลำโพงชนิดนี้ทำงานเหมือนลำโพงขนาดเล็ก คือ ถ้าป้อนแรงดันไฟฟ้า กระแสตรงเข้าจะไม่มีเสียงถ้าต้องการให้มีสัญญาณเสียงต้องทำการป้อนสัญญาณความถี่เข้าไป ลำโพงชนิดนี้ สามารถกำเนิดเสียงที่มีความแตกต่างกันตามความถี่ที่ป้อนเข้ามา ในปริญญานิพนธ์นี้ผู้จัดทำเลือกใช้ Active buzzer 3.3-5 V_{DC} แสดงดังรูปที่ 2.7 มาใช้เป็นอุปกรณ์แจ้งเตือนโดยความดังจะเปลี่ยนตามขนาดแรงดันไฟฟ้าที่จ่ายให้กับ Active buzzer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.7 โมดูลเสียงเตือน (Active Buzzer) [8]

Active Buzzer เป็นตัวสร้างเสียง 2,300 Hz สามารถสร้างเสียงเตือนได้โดยการจ่ายไฟ 3.3–5 V_{DC} ดังรูปที่ 2.7

คุณสมบัติ

- ออดไฟฟ้ากำเนิดเสียงสูง
- ไฟเลี้ยงทำงาน 5 V (สูงสุด 3 – 7 V)
- กระแส 25 – 30 mA
- ความดัง 35 dBA
- ความถี่ 2,300 ± 300 Hz (เสียงสูง)
- เส้นผ่านศูนย์กลาง 12 mm

2.6 โมดูลรีเลย์ 5 V

รูปที่ 2.8 โมดูลรีเลย์ 5 V [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมดูลรีเลย์ 5 V - Active Low ใช้งานในการควบคุมอุปกรณ์ไฟฟ้า รับกระแสได้สูงถึง 10 A 250 V_{AC} และ 10 A 30 V_{DC} มี LED แสดงสถานะการทำงานของรีเลย์ ดังรูปที่ 2.8

คุณสมบัติ

- ทำงานที่แรงดัน 5 V
- รองรับแรงดันอินพุต 3.3 V – 5 V
- ใช้ทรานซิสเตอร์ขับรีเลย์โดยตรง
- รีเลย์ขนาด 10 A 250 V_{AC} และ 10 A 30 V_{DC}
- มีหลอด LED ขนาด 3 mm. 2 หลอด สีแดงบอกสถานะการจ่ายไฟ สีเขียวบอกสถานะการทำงานของรีเลย์

2.7 NodeMCU [10]

NodeMCU คือแพลตฟอร์มหนึ่งที่ใช้ช่วยในการสร้างโปรเจค Internet of Things (IoT) ที่ประกอบไปด้วย Development Kit (ตัวบอร์ด) และ Firmware (Software บนบอร์ด) ที่เป็น open source สามารถเขียน โปรแกรมด้วยภาษา Lau ได้ทำให้ใช้งานได้ง่ายขึ้น มาพร้อมกับโมดูล WiFi ซึ่งใช้เชื่อมต่อกับอินเตอร์เน็ต

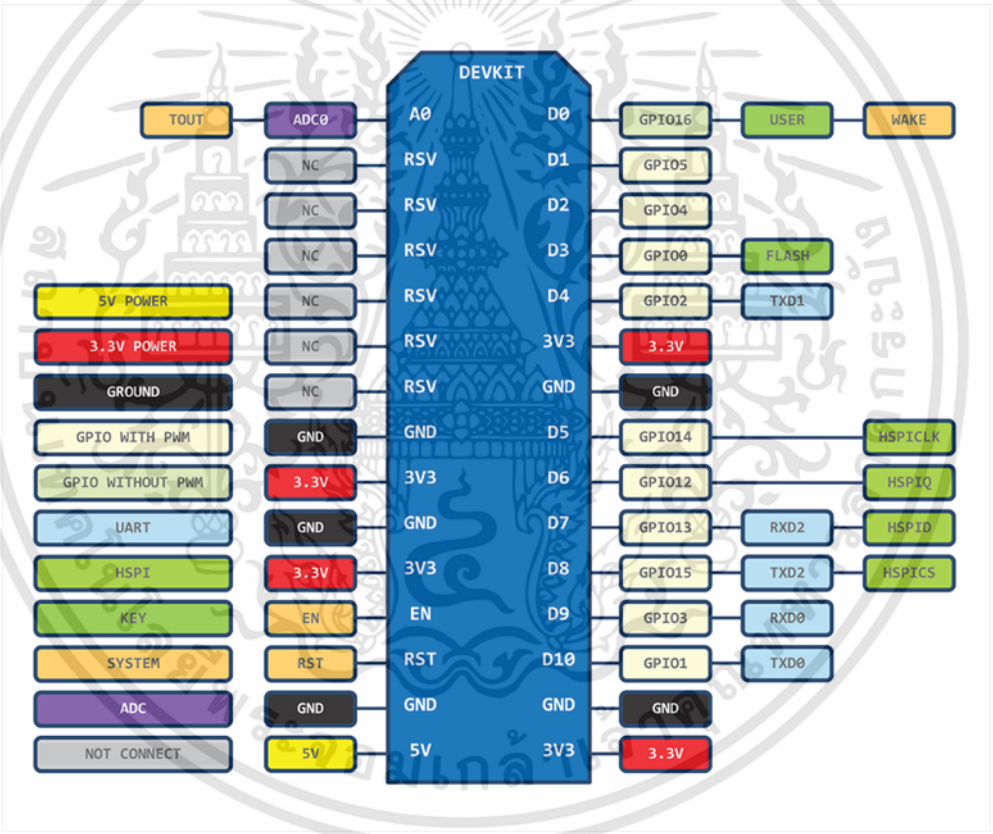
NodeMCU Development Kit

ชุดพัฒนาบอร์ด NodeMCU หรือเรียกว่า NodeMCU DevKit มี 2 เวอร์ชั่น ได้แก่

1) NodeMCU Devkit 0.9 (ESP-12) เวอร์ชั่น 1 แสดงดังรูปที่ 2.9 และแสดงตำแหน่งขาดังรูปที่ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

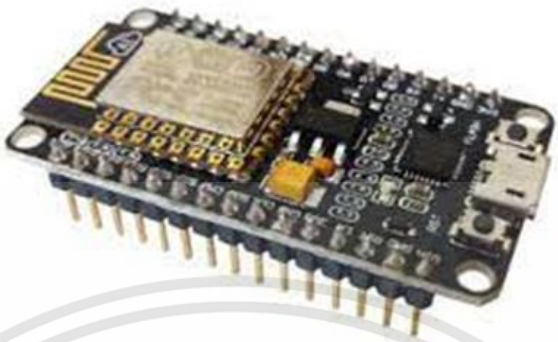
รูปที่ 2.9 NodeMCU Devkit เวอร์ชั่น 1 [10]



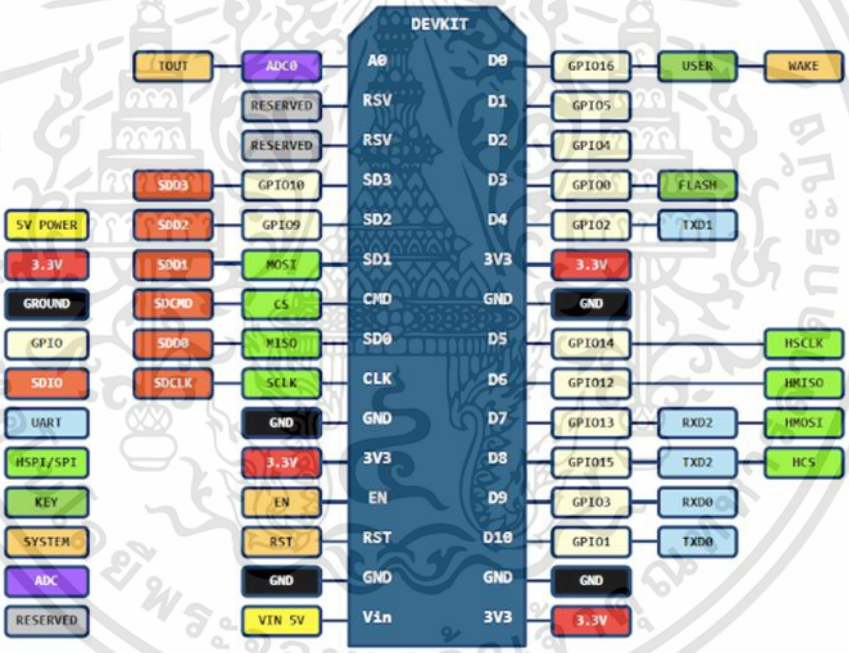
รูปที่ 2.10 ตำแหน่งขาของ NodeMCU Devkit เวอร์ชั่น 1 [10]

2) NodeMCU Devkit 1.0 (ESP-12E) เวอร์ชั่น 2 แสดงดังรูปที่ 2.11 และแสดงตำแหน่งขาดังรูปที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 NodeMCU Devkit เวอร์ชั่น 2 [10]



รูปที่ 2.12 ตำแหน่งขาของ NodeMCU Devkit เวอร์ชั่น 2 [10]

รายละเอียดของ NodeMCU Devkit เวอร์ชั่น 2

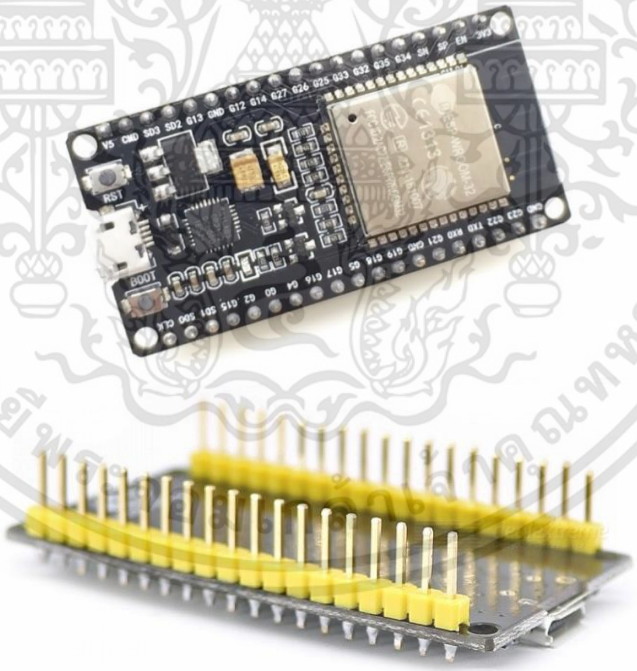
- ชุดพัฒนานี้ based on โมดูล WiFi ที่ชื่อ ESP8266
- มี GPIO PWM I2C 1-Wire และ ADC รวมอยู่บนบอร์ดเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มี USB-TTL มาในตัวไม่ต้องซื้อแยกเหมือนกับการใช้ ESP8266 ปกติ ทำให้ใช้งานได้สะดวก
- มีขา GPIO 10 ขา ทุก ๆ ขาสามารถเป็น PWM I2C และ 1-wire ได้
- มี PCB antenna สำหรับรับส่งสัญญาณไร้สาย
- ใช้คอนเนกเตอร์แบบ micro-USB สำหรับจ่ายแรงดันไฟเลี้ยงหรือเท่ากับ +5V และสำหรับดาวน์โหลดเฟิร์มแวร์

2.8 โมดูลไวไฟ NodeMCU-ESP32S [11]

โมดูลไวไฟ ESP32S แสดงดังรูปที่ 2.13 และ 2.14 (NodeMCU-32S esp32 DevKit Module WiFi Bluetooth IoT Development Board - 38 pins)

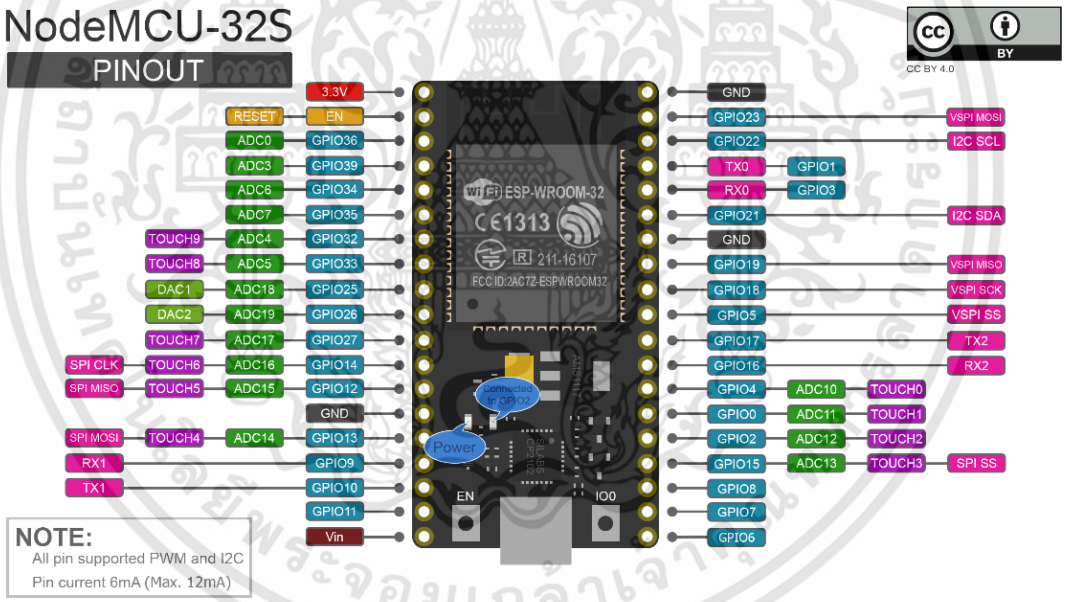


รูปที่ 2.13 NodeMCU-32S 38 pins [11]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียด

- มีจำนวน 38 ขา (สีเหลือง)
- หน่วยประมวลผลสองแกน Tensilica LX 6 ความเร็วสูงสุด 240MHz
- แรมภายใน 520 Kb และ หน่วยความจำขนาด 16 Mb
- ใช้แรงดันไฟ 2.6V -3V กินกระแส 2.5 µA
- WiFi 2.4 GHz แบบ Low Power
- Bluetooth Dual-mode classic และ BLE
- มี GPIO UART SPI I2C ADC DAC I2S และ PWM



รูปที่ 2.14 ตำแหน่งขาของ NodeMCU-32S จำนวน 38 ขา [11]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 ฐานข้อมูล (Database) [12]

2.9.1 ความหมายของฐานข้อมูล

ข้อมูล คือ ข้อเท็จจริงของสิ่งที่เราสนใจ ข้อเท็จจริงที่เป็นตัวเลข ข้อความ หรือรายละเอียด ซึ่งอาจอยู่ในรูปแบบต่าง ๆ เช่น ภาพ เสียง วีดิโอไม่ว่าจะเป็นคน สัตว์ สิ่งของ หรือเหตุการณ์ที่เกี่ยวข้องกับสิ่งต่าง ๆ ข้อมูลเป็นเรื่องเกี่ยวกับเหตุการณ์ที่เกิดขึ้นอย่างต่อเนื่องและต้องถูกต้องแม่นยำครบถ้วน ซึ่งขึ้นอยู่กับผู้ดำเนินการที่ให้ความสำคัญของความรวดเร็วของการเก็บข้อมูล ดังนั้นการเก็บข้อมูลจึงเป็นการเก็บรวบรวมเกี่ยวกับข้อเท็จจริงของสิ่งที่เราสนใจนั่นเอง ข้อมูลจึงหมายถึงตัวแทนของข้อเท็จจริง หรือความเป็นไปของสิ่งของที่เราสนใจ

Database หรือ ฐานข้อมูล คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลาย ๆ แฟ้มข้อมูล

ระบบฐานข้อมูล (Database System) คือ ระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบมีความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่ชัดเจน ในระบบฐานข้อมูลจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูลเกี่ยวข้องสัมพันธ์กันเข้าไว้ด้วยกันอย่างเป็นระบบและเปิดโอกาสให้ผู้ใช้สามารถใช้งานและดูแลรักษาป้องกันข้อมูลเหล่านี้ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (data base management system) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งคำถามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล

ลักษณะข้อมูลในฐานข้อมูล

ระบบฐานข้อมูล (Database System) หมายถึง โครงสร้างสารสนเทศที่ประกอบด้วยรายละเอียดของข้อมูลที่เกี่ยวข้องกันที่จะนำมาใช้ในระบบต่าง ๆ ร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลเป็นการจัดเก็บข้อมูลอย่างเป็นระบบ ทำให้ผู้ใช้สามารถใช้ข้อมูลที่เกี่ยวข้องในระบบงาน ต่าง ๆ ร่วมกันได้ โดยที่จะไม่เกิดความซ้ำซ้อนของข้อมูล และยังสามารถหลีกเลี่ยงความขัดแย้งของข้อมูลด้วย อีกทั้งข้อมูลในระบบก็จะถูกต้องเชื่อถือได้ และเป็นมาตรฐานเดียวกัน โดยจะมีการกำหนดระบบความปลอดภัยของข้อมูลขึ้น

1) ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

เป็นการเก็บข้อมูลในรูปแบบที่เป็นตาราง (Table) หรือเรียกว่า รีเลชั่น (Relation) มีลักษณะเป็น 2 มิติ คือเป็นแถว (row) และเป็นคอลัมน์ (column) การเชื่อมโยงข้อมูลระหว่างตาราง จะเชื่อมโยงโดยใช้ แอททริบิวต์ (attribute) หรือคอลัมน์ที่เหมือนกันทั้งสองตารางเป็นตัวเชื่อมโยงข้อมูล ฐานข้อมูลเชิงสัมพันธ์นี้จะเป็นรูปแบบของฐานข้อมูลที่นิยมใช้ในปัจจุบัน

2) ฐานข้อมูลแบบเครือข่าย (Network Database)

ฐานข้อมูลแบบเครือข่ายจะเป็นการรวมระเบียนต่าง ๆ และความสัมพันธ์ระหว่างระเบียน แต่จะต่างกับฐานข้อมูลเชิงสัมพันธ์ คือ ในฐานข้อมูลเชิงสัมพันธ์จะแฝงความสัมพันธ์เอาไว้ โดยระเบียนที่มีความสัมพันธ์กันจะต้องมีค่าของข้อมูลในแอททริบิวต์ใดแอททริบิวต์หนึ่งเหมือนกัน แต่ฐานข้อมูลแบบเครือข่าย จะแสดงความสัมพันธ์อย่างชัดเจน

3) ฐานข้อมูลแบบลำดับชั้น (Hierarchical Database)

ฐานข้อมูลแบบลำดับชั้น เป็นโครงสร้างที่จัดเก็บข้อมูลในลักษณะความสัมพันธ์แบบพ่อ-ลูก (Parent-Child Relationship Type : PCR Type) หรือเป็นโครงสร้างรูปแบบต้นไม้ (Tree) ข้อมูลที่จัดเก็บในที่นี้ คือ ระเบียน (Record) ซึ่งประกอบด้วยค่าของเขตข้อมูล (Field) ของเอนทิตี้หนึ่ง ๆ ฐานข้อมูลแบบลำดับชั้นนี้คล้ายคลึงกับฐานข้อมูลแบบเครือข่าย แต่ต่างกันที่ฐานข้อมูลแบบลำดับชั้น มีกฎเพิ่มขึ้นมาอีกหนึ่งประการ คือ ในแต่ละกรอบจะมีลูกศรวิ่งเข้าหาได้ไม่เกิน 1 หัวลูกศร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์ของฐานข้อมูล

1) ลดการเก็บข้อมูลที่ซ้ำซ้อน ข้อมูลบางชุดที่อยู่ในรูปของแฟ้มข้อมูลอาจมีปรากฏอยู่หลาย ๆ แห่ง เพราะมีผู้ใช้ข้อมูลชุดนี้หลายคน เมื่อใช้ระบบฐานข้อมูลแล้วจะช่วยให้ความซ้ำซ้อนของข้อมูลลดน้อยลง

2) รักษาความถูกต้องของข้อมูล เนื่องจากฐานข้อมูลมีเพียงฐานข้อมูลเดียว ในกรณีที่มีข้อมูลชุดเดียวกันปรากฏอยู่หลายแห่งในฐานข้อมูล ข้อมูลเหล่านี้จะต้องตรงกัน ถ้ามีการแก้ไขข้อมูลนี้ทุก ๆ แห่งที่ข้อมูลปรากฏอยู่จะแก้ไขให้ถูกต้องตามกันหมดโดยอัตโนมัติด้วยระบบจัดการฐานข้อมูล

3) การป้องกันและรักษาความปลอดภัยให้กับข้อมูลทำได้อย่างสะดวก การป้องกันและรักษาความปลอดภัยกับข้อมูลระบบฐานข้อมูลจะให้เฉพาะผู้ที่เกี่ยวข้องเท่านั้น ซึ่งก่อให้เกิดความปลอดภัย (security) ของข้อมูลด้วย

ในปริญญานิพนธ์นี้ได้เลือกโปรแกรม Firebase ซึ่งเป็นโปรแกรมจัดการฐานข้อมูลแบบเรียลไทม์ (Realtime Database)

2.9.2 ฐานข้อมูล Firebase [13]

Firebase ดังรูปที่ 2.15 คือบริการหนึ่งของ Google เป็นการบริการข้อมูลแบบออนไลน์ในรูปแบบ Realtime Database สำหรับ Application และ Web Application โดยที่ Firebase สามารถนำมาเป็นตัวกลางในการเชื่อมต่ออุปกรณ์เข้าด้วยกัน เช่น NodeMCU ESP8266 ส่งข้อมูลไปยัง แอพพลิเคชั่นบนระบบปฏิบัติการ Android เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.15 ฐานข้อมูล (Firebase) [13]

Firebase คือฐานข้อมูลแบบ NoSQL โดยจะไม่ใช้ภาษา SQL ในการจัดการข้อมูล แต่ออกแบบให้มีความยืดหยุ่นและเน้นความเร็วในการใช้งาน โดย NoSQL ที่นิยมใช้งานมากที่สุดในปัจจุบันคือ MongoDB ซึ่งมีการเก็บข้อมูลแบบ JSON โดยที่มีตารางเหมือนกับ SQL แต่ไม่มีคอลัมน์ ในหนึ่งแถวสามารถเก็บข้อมูลได้ทั้งข้อความ (String) ตัวเลข (Number) และอื่น ๆ รวมไปถึงอาเรย์ และ Object

ฐานข้อมูลแบบเรียลไทม์ (Realtime Database)

- ฐานข้อมูลแบบเรียลไทม์ของ Firebase: ในฐานข้อมูลแบบเรียลไทม์ของ Firebase จะเก็บข้อมูลไว้ในโครงสร้าง JSON ขนาดใหญ่ นั่นเป็นเหตุผลว่าทำไมข้อมูลขนาดเล็กหรือเรียบง่ายจึงง่ายต่อการจัดเก็บ แต่ข้อมูลที่ซับซ้อนและตามลำดับชั้นนั้นยากหรือจัดระเบียบเมื่อมีการปรับขนาด

- ฐานข้อมูลแบบเรียลไทม์ : รองรับออฟไลน์สำหรับอุปกรณ์มือถือ (Android และ iOS) เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฐานข้อมูลแบบเรียลไทม์ : ฐานข้อมูลแบบเรียลไทม์บันทึกสถานะการเชื่อมต่อไคลเอ็นต์และให้ข้อมูลอัปเดตแก่เราเมื่อใดก็ตามที่การเชื่อมต่อไคลเอ็นต์ออนไลน์หรือออฟไลน์

2.10 ภาษาที่เกี่ยวข้อง

2.10.1 ภาษาซี [14]

- โครงสร้างโปรแกรมภาษาซี บน Arduino

ประกอบด้วย 5 ลักษณะ ดังนี้

1. พรีโปรเซสเซอร์ไดเร็กทีฟ (Preprocessor directives)

โดยปกติแล้วเกือบทุกโปรแกรมต้องมี โดยส่วนนี้จะเป็นส่วนที่คอมไพลเลอร์จะมีการประมวลผลและทำตามคำสั่งก่อนที่จะมีการคอมไพล์โปรแกรม ซึ่งจะเริ่มต้นด้วยเครื่องหมายไดเร็กทีฟ (directive) หรือเครื่องหมายสี่เหลี่ยม # แล้วจึงตามด้วยชื่อคำสั่งที่ต้องการเรียกใช้ หรือกำหนด โดยปกติแล้วส่วนนี้จะอยู่ในส่วนบนสุด หรือส่วนหัวของโปรแกรม และต้องอยู่นอกฟังก์ชั่นหลักใด ๆ ก็ตาม

#include เป็นคำสั่งที่ใช้อ้างอิงไฟล์ภายนอก เพื่อเรียกใช้ฟังก์ชั่น หรือตัวแปรที่มีการสร้างหรือกำหนดไว้ในไฟล์นั้น รูปแบบการใช้งานคือ #include <ชื่อไฟล์.h>

การอ้างอิงไฟล์จากภายใน หรือการอ้างอิงไฟล์ไลบารี่ที่มีอยู่แล้วใน Arduino หรือเป็นไลบารี่ที่เราเพิ่มเข้าไปเอง จะใช้เครื่องหมาย <> ในการคร่อมชื่อไฟล์ไว้ เพื่อให้โปรแกรมคอมไพลเลอร์เข้าใจว่าควรไปหาไฟล์เหล่านี้จากในโฟลเดอร์ไลบารี่ แต่หากต้องการอ้างอิงไฟล์ที่อยู่ในโฟลเดอร์โปรเจค จะต้องใช้เครื่องหมาย " " คร่อมแทน ซึ่งคอมไพล์เลอร์จะวิ่งไปหาไฟล์นี้โดยอ้างอิงจากไฟล์โปรแกรมที่คอมไพล์เลอร์อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนของการกำหนดค่า (Global declarations)

ส่วนนี้จะเป็นส่วนที่ใช้ในการกำหนดชนิดตัวแปรแบบนอกฟังก์ชั่น หรือประกาศฟังก์ชั่น เพื่อให้ฟังก์ชั่นที่ประกาศสามารถกำหนด หรือเรียกใช้ได้จากทุกส่วนของโปรแกรม

3. ฟังก์ชั่น setup() และฟังก์ชั่น loop()

- ฟังก์ชั่น setup() และฟังก์ชั่น loop() เป็นคำสั่งที่ถูกบังคับให้ต้องมีในทุกโปรแกรม
- ฟังก์ชั่น setup() จะเป็นฟังก์ชั่นแรกที่ถูกเรียกใช้ นิยมใช้กำหนดค่า หรือเริ่มต้นใช้งานไลบารี่ต่าง ๆ
- ฟังก์ชั่น loop() จะเป็นฟังก์ชั่นที่ทำงานหลังจากฟังก์ชั่น setup() ได้ทำงานเสร็จสิ้นไปแล้ว และมีการวนรอบแบบไม่รู้จบ เมื่อฟังก์ชั่น loop() งานครบตามคำสั่งแล้ว ฟังก์ชั่น loop() ก็จะถูกเรียกขึ้นมาใช้อีก

4. การสร้างฟังก์ชั่น และการใช้งานฟังก์ชั่น (Users-defined function)

ในการสร้างฟังก์ชั่นขึ้นมา คำสั่งต่าง ๆ ที่อยู่ภายในฟังก์ชั่น ต้องอยู่ภายใต้เครื่องหมายปีกกาเปิด { และปีกกาปิด } เท่านั้น ภายใต้เครื่องหมาย {} เราสามารถนำฟังก์ชั่นหรือคำสั่งใด ๆ ก็ได้มาใส่ไว้ แต่จะต้องคั่นแต่ละคำสั่งด้วยเครื่องหมายเซมิโคล่อน ;

5. ส่วนอธิบายโปรแกรม (Program comments)

ส่วนอธิบายโปรแกรม หรือการคอมเม้นต์โปรแกรมเป็นส่วนที่สำคัญอย่างมากที่จะช่วยให้ผู้ที่ไม่ได้เขียนโปรแกรม หรือเป็นผู้เขียนโปรแกรมเข้าใจโปรแกรมได้ง่ายขึ้นโดยอ่านจากคอมเม้นต์ แทนการทำความเข้าใจโปรแกรมโดยอ่านแต่ละฟังก์ชั่น

การคอมเม้นโค้ดมีอยู่ 2 รูปแบบ คือเปิดด้วย /* และปิดด้วย */ เป็นการคอมเม้นต์โค้ดแบบข้ามบรรทัด คือตราบใดที่ยังไม่มี */ตรงส่วนนั้นจะเป็นคอมเม้นต์ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โครงสร้างพื้นฐานของภาษาซีที่ใช้กับ Arduino

ประกอบไปด้วย 3 ส่วนใหญ่ ๆ ด้วยกัน คือ

1) header : ในส่วนนี้จะมีหรือไม่มีก็ได้ ถ้ามีต้องกำหนดไว้ในส่วนเริ่มต้นของโปรแกรม ซึ่งส่วนของ Header ได้แก่ ส่วนที่เป็น Compiler Directive ต่าง ๆ รวมไปถึงส่วนของการประกาศตัวแปร และค่าคงที่ต่าง ๆ ที่จะใช้ในโปรแกรม

2) setup() : ในส่วนนี้เป็นฟังก์ชั่นบังคับที่ต้องกำหนดให้มีในทุก ๆ โปรแกรม ถึงแม้ว่าในบางโปรแกรมจะไม่ต้องการใช้งานก็ยังจำเป็นต้องประกาศไว้ด้วยเสมอ เพียงแต่ไม่ต้องเขียนคำสั่งใด ๆ ไว้ในระหว่าง วงเล็บปีกกา { } ที่ใช้เป็นตัวกำหนดขอบเขตของฟังก์ชั่น โดยฟังก์ชั่นนี้จะใช้สำหรับบรรจุคำสั่งใน ส่วนที่ต้องการให้โปรแกรมทำงานเพียงรอบเดียวตอนเริ่มต้นทำงานของโปรแกรมครั้งแรกเท่านั้น

3) loop() : เป็นส่วนฟังก์ชั่นบังคับที่ต้องกำหนดให้มีในทุก ๆ โปรแกรม เช่นเดียวกันกับฟังก์ชั่น setup() โดยฟังก์ชั่น loop() นี้จะใช้บรรจุคำสั่งที่ต้องการให้โปรแกรมทำงานเป็นวงรอบซ้ำ ๆ กันไม่รู้จบ

2.10.2 ภาษา JavaScript [15]

JavaScript คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ที่กำลังได้รับความนิยมอย่างสูง

JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ ที่เรียกกันว่า ‘สคริปต์ (script)’ ซึ่งในการสร้างและพัฒนาเว็บไซต์ใช้ร่วมกับ HTML เพื่อให้เว็บไซต์มีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ แปลความและดำเนินงานไปทีละคำสั่ง (interpret) หรือเรียกว่า อ็อบเจ็กโอเรียลเต็ด (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JavaScript ถูกพัฒนาขึ้นโดย เน็ตสเคปคอมมิวนิเคชันส์ (Netscape Communications Corporation) โดยใช้ชื่อว่า Live Script ออกมาพร้อมกับ Netscape Navigator 2.0 เพื่อใช้สร้างเว็บเพจโดยติดต่อกับเซิร์ฟเวอร์แบบ Live Wire ต่อมา เน็ตสเคปจึงได้ร่วมมือกับ บริษัทซันไมโครซิสเต็มส์ปรับปรุงระบบของเบราวเซอร์เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวาได้ และได้ปรับปรุง LiveScript ใหม่เมื่อ ปี 2538 แล้วตั้งชื่อใหม่ว่า ‘JavaScript’

JavaScript สามารถทำให้การสร้างเว็บเพจมีลูกเล่นต่าง ๆ มากมาย และยังสามารถโต้ตอบกับผู้ใช้ได้อย่างทันที เช่น การใช้เมาส์คลิกหรือการกรอกข้อความในฟอร์ม เป็นต้น

เนื่องจาก JavaScript ช่วยให้ผู้พัฒนาสามารถสร้างเว็บเพจได้ตรงกับความต้องการและมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิดจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง

การทำงานของ JavaScript จะต้องมีการแปลคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยเบราวเซอร์ (เรียกว่าเป็น client-side script) ดังนั้น JavaScript จึงสามารถทำงานได้ เฉพาะบนเบราวเซอร์ที่สนับสนุนเท่านั้น

โปรแกรมที่ใช้ร่วมกับ ภาษา JavaScript

- **โปรแกรม Node.js** [16]

Node.js ถูกพัฒนาและทำงานด้วยใช้ Chrome V8 engine สำหรับคอมไพล์ภาษา JavaScript ให้เป็นภาษาเครื่องด้วยการคอมไพล์แบบ Just-in-time (JIT) เพื่อเพิ่มประสิทธิภาพการทำงานของภาษา JavaScript จากที่แต่เดิมมันเป็นภาษาที่มีการทำงานแบบ Interpreted

Node.js เป็นโปรแกรมที่สามารถใช้ได้ทั้งบน Windows Linux และ Mac OS X ซึ่งสามารถเขียนโปรแกรมในภาษา JavaScript และนำไปรันได้ทุกระบบปฏิบัติการที่สนับสนุนโดย Node.js นี่เป็นแนวคิดของการเขียนครั้งเดียวแต่ทำงานได้ทุกที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในช่วงเริ่มแรกภาษา JavaScript ถูกพัฒนาขึ้นมาสำหรับเขียน Script บนเว็บเบราว์เซอร์เพื่อทำให้หน้าเว็บสามารถตอบสนองได้แบบเรียลไทม์

ในปัจจุบัน ภาษา JavaScript ไม่ได้ถูกจำกัดสำหรับการเขียนโปรแกรมเพียงบนเว็บเบราว์เซอร์อีกต่อไป ในการมาของ Node.js เราสามารถเขียนโปรแกรม Command line สร้างเว็บเซิร์ฟเวอร์ หรือเขียนโปรแกรมบน Desktop โดยการใช้ Framework อย่าง Electron หรือโปรแกรมทุกรูปแบบเหมือนกับภาษาอื่น ๆ ได้

2.10.3 ภาษา HTML [17]

HTML ย่อมาจาก HyperText Markup Language เป็นภาษาคอมพิวเตอร์ที่ใช้สร้างหน้าเว็บ (Web Page) ในรูปแบบของไฟล์ HTML (คือไฟล์ที่มีนามสกุลเป็น .htm หรือ.html) ซึ่งมีเว็บเบราว์เซอร์ (Web Browser) เป็นโปรแกรมที่ใช้แปลงไฟล์ HTML เพื่อแสดงผลในรูปของหน้าเว็บ ไฟล์ HTML เป็นไฟล์รหัสแอสกี (ASCII) ถูกบันทึกในรูปของไฟล์เอกสาร (Text File) ที่สามารถถูกสร้างจากโปรแกรมสร้างไฟล์ข้อความ (Text Editor) เช่น Notepad หรือ Word Processing ทั่ว ๆ ไป ซึ่งลักษณะของไฟล์ HTML ประกอบไปด้วยแท็ก (Tag) ต่าง ๆ ที่เป็นคำ สั่งของ HTML ซึ่งแท็กจะอยู่ภายในเครื่องหมาย < และ >

แท็กใน HTML แบ่งเป็น 2 ประเภท คือ

1) คอนเทนเนอร์แท็ก (Container Tag) และ โดยที่คอนเทนเนอร์แท็ก ประกอบไปด้วยแท็กเปิด และแท็กปิด โดยที่แท็กปิดจะมีเครื่องหมาย/ นำหน้าแท็ก เช่น <H1>. . .</H1>

2) แท็กเปล่า (Empty Tag) จะมีแท็กเปิดอย่างเดียว เช่น<HR> ซึ่งแท็กจะถูกเขียนด้วยตัวอักษรพิมพ์ใหญ่หรือพิมพ์เล็กก็ได้จะไม่มีผลต่อการแสดงผลของเว็บเบราว์เซอร์ เช่น <BR>, <br>,
 หรือ
 เว็บเบราว์เซอร์จะแปลความหมายเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างไฟล์ HTML แบ่งออกเป็น 2 ส่วน คือ

1) ส่วนหัวเรื่อง (Head Section) โดยจะมีแท็ก <HTML> และ </HTML> เป็นตัวกำหนดขอบเขตไฟล์ซึ่งส่วนหัวเรื่อง มีไว้กำหนดข้อมูลเฉพาะของหน้าเว็บ เช่น ชื่อเรื่องของเว็บภายในแท็ก <HEAD> และ </HEAD>

2) ส่วนเนื้อหา (Body Section) มีไว้กำหนดรายละเอียดต่าง ๆ ที่ต้องการแสดงบนหน้าเว็บ เช่น ข้อความ และรูปภาพภายในแท็ก <BODY> และ </BODY>

คำศัพท์ที่เกี่ยวข้องกับ HTML

1. Internet คือ เครือข่ายคอมพิวเตอร์ที่ใหญ่ที่สุดในโลก เกิดจากการเชื่อมโยงของเครือข่ายต่าง ๆ เข้าด้วยกัน

2. Hypertext คือ รูปแบบเอกสารที่บรรจุการเชื่อมโยงไปยังเอกสารอื่น ๆ ซึ่งสามารถใช้ข้อความ หรือรูป เป็นจุดเชื่อมโยง

3. WWW ย่อมาจาก World Wide Web คือ การสื่อสารด้วยการเชื่อมโยงเครือข่ายข่าวสารแบบใยแมงมุม (Web) แสดงผลด้วยเอกสารไฮเปอร์เท็กซ์

4. HTTP คือ ย่อมาจาก Hypertext Transfer Protocol เป็นรูปแบบการสื่อสารที่ใช้ในการรับส่งข้อมูลไฮเปอร์เท็กซ์ในเครือข่าย อินเทอร์เน็ต

5. Web Browser คือ โปรแกรมสำหรับแสดงผลหน้าเว็บ เช่น Internet Explorer, Mozilla Firefox และ Google Chrome เป็นต้น

6. Web Page คือ หน้าเอกสารที่อยู่ในรูปของไฮเปอร์เท็กซ์

7. Web Site คือ กลุ่มของหน้าเว็บหลาย ๆ หน้ารวมเข้าด้วยกัน

8. Home Page คือ หน้าเว็บ หน้าแรกของเว็บไซต์

9. Web Site คือ เครื่องให้บริการที่เป็นที่เก็บข้อมูลของ เว็บไซต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ร่วมกับ ภาษา HTML

- **โปรแกรม Vue.js** [18]

Vue.js ใช้เป็นเหมือนส่วนเพิ่มความสามารถที่เอาไปใช้งานกับ HTML ซึ่งชื่อ Vue นั้นก็มีที่มาจากการที่ library นี้จะโฟกัสที่การจัดการในส่วนของ view เท่านั้น เพื่อให้สามารถนำ Vue ไปใช้งานร่วมกับ Library อื่น ๆ ได้สะดวก และยังเหมาะกับการทำ Single-Page Applications (SPA) หรือเว็บที่ไม่ต้องเปลี่ยนหน้าบ่อย ๆ

- Vue.js ช่วยให้สามารถจัดการกับ element ที่นำมา render บนหน้าเว็บได้ง่าย
- โดยปกติถ้าใช้เพียง JavaScript เวลาต้องการจัดการกับ element ในหน้าเว็บจะต้องเขียน JS ไปแก้ไข DOM โดยตรงเพื่อจัดการกับ element ตาม id หรือ class
- แต่ใน Vue.js สามารถใช้ “component” จัดการกับ element ได้โดยไม่ต้องเข้าไปแก้ไข DOM โดยตรง

2.11 โรคออฟฟิศซินโดรม [21]

จากรูปที่ 2.16 ออฟฟิศซินโดรม (office syndrome) คือ กลุ่มอาการปวดกล้ามเนื้อและเยื่อพังผืด รวมถึงอาการปวดหรือชาจากอาการอักเสบจากเนื้อเยื่อและเอ็น มักเกิดขึ้นกับกลุ่มวัยทำงานที่ต้องนั่งทำงานในออฟฟิศหรือทำงานโดยใช้คอมพิวเตอร์และมือถือเป็นประจำ

ออฟฟิศซินโดรม มักเกิดจากการใช้งานกล้ามเนื้อมัดเดิม ๆ ซ้ำ ๆ เป็นเวลานาน ไม่ค่อยได้เปลี่ยนท่าทาง รวมถึงอิริยาบถที่ไม่เหมาะสม เช่น การนั่งไขว่ห้างเป็นประจำ การนั่งตัวงอ หรือการก้มหน้านาน ๆ เป็นต้น ทำให้กล้ามเนื้อที่ถูกใช้งานซ้ำ ๆ มีการหดเกร็ง หรือยืดค้างในรูปแบบเดิมบ่อย ๆ จนกล้ามเนื้อมัดนั้น ๆ เกิดการบาดเจ็บ หรืออาจขมวดเป็นก้อนตึง และเกิดอาการปวดตามมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล้ามเนื้อส่วนต่าง ๆ ในร่างกายของเรามีลักษณะเป็นเส้นใยร้อยโยงต่อเนื่องกันหลายส่วน เมื่อกล้ามเนื้อเริ่มมีการขมวดกันเป็นปมขึ้น ก็ดึงรั้งกันไปมา ตอนแรกอาการปวดตึงอาจจะเริ่มจากจุดหนึ่ง แต่พอนานวันเข้าก็จะร้าวไปปวดอีกจุดหนึ่ง เพราะถูกดึงรั้งจากกล้ามเนื้อส่วนที่หดเกร็ง รู้ตัวอีกทีก็จะปวดเป็นบริเวณกว้าง ๆ ระบุหาตำแหน่งที่ปวดจริง ๆ ไม่ได้

รูปที่ 2.16 ออฟฟิศซินโดรม [21]

ธรรมชาติของคนทำงานประจำในยุคนี้ มักจะปรับเปลี่ยนพฤติกรรมการทำงานได้ยาก เพราะต้องโฟกัสกับงานที่ทำ หรือยุ่งจนลืมปรับเปลี่ยนท่าทางและหยุดพัก บ่อยครั้งจึงปล่อยให้อาการของโรคนี้มีอาการหนักมากขึ้น หรือลุกลามไปยังกล้ามเนื้อและระบบประสาทส่วนอื่น ๆ ที่เกี่ยวข้องกัน

หากปล่อยไว้โดยไม่รักษา หรือไม่เปลี่ยนแปลงพฤติกรรม ก็อาจก่อให้เกิดอันตรายตามมา เช่น เสี่ยงต่อการเกิดหมอนรองกระดูกทับเส้นประสาท กระดูกสันหลังคด และแขนขาอ่อนแรงอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.1 สาเหตุของออฟฟิศซินโดรม [23]

ออฟฟิศซินโดรม มีสาเหตุมาจาก 2 ปัจจัยหลัก ๆ ดังนี้

1) สภาพแวดล้อมที่ทำงานไม่ถูกสุขลักษณะ เช่น มีแสงสว่างน้อย อุปกรณ์ที่ใช้ไม่เหมาะสม โต๊ะทำงานอยู่ในระดับที่ไม่พอดี ตำแหน่งหน้าจอคอมพิวเตอร์สูงหรือต่ำจนเกินไป เก้าอี้ที่นั่งไม่มีพนักพิง

2) อิริยาบถในการนั่งทำงานไม่เหมาะสม เช่น นั่งหลังค่อม หลังงอ นั่งบนเก้าอี้ที่ไม่มีพนักพิง หรือนั่งไม่เต็มก้น นั่งกอดอก นั่งไขว่ห้าง นั่งอยู่ในท่าเดิมนาน ๆ ไม่ขยับไปไหน

2.11.2 อาการออฟฟิศซินโดรม [23]

อาการออฟฟิศซินโดรม มักจะมีอาการหลากหลาย เริ่มตั้งแต่ปวดกล้ามเนื้อเบา ๆ ไปจนถึงกระดูกทับเส้น อาการออฟฟิศซินโดรมนั้นสามารถบรรเทาลงได้ด้วยการทำท่าบริหาร office syndrome ซึ่งหากใครที่เริ่มมีอาการปวดเมื่อยจากการนั่งทำงาน และสงสัยว่าจะเป็นออฟฟิศซินโดรม สามารถสังเกตอาการได้ดังนี้

1) ปวดกล้ามเนื้อบริเวณใดบริเวณหนึ่งของร่างกาย เช่น ไหล่ สะบัก คอ บ่า ท้ายทอย ปวดหลังส่วนบนหรือส่วนล่าง ปวดมือ ข้อมือ ปวดข้อศอก ปวดเข่าหรือข้อเท้า ปวดสะโพกหรือต้นขา มักจะมีอาการปวดเป็นบริเวณกว้าง ปวดร้าวไปบริเวณอื่นใกล้เคียง มีลักษณะการปวดแบบล้า ๆ ไม่สามารถระบุอาการหรือตำแหน่งที่ชัดเจนได้ โดยจะมีอาการปวดตั้งแต่เล็กน้อยไปจนถึงรุนแรงและทรมานอย่างมาก

2) มีอาการของระบบประสาทร่วมด้วย เช่น ชา ปวดร้าว หรืออาจมีอาการหูอื้อ มึนงง ตาพร่ามัว ปวดไมเกรน หรือปวดศีรษะอย่างรุนแรง

3) อาการทางระบบประสาทที่ถูกกดทับ เช่น ชาบริเวณมือและแขน และหากมีการกดทับเส้นประสาทนานเกินไปอาจมีอาการอ่อนแรงร่วมด้วย

4) อาการทางตา เช่น ปวดตา เมื่อยล้าตา มีอาการแสบตา ระคายเคือง ตราพร่ามัว สู้แสงไม่ได้ น้ำตาไหล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) อาการทางผิวหนัง เช่น คันตามลำตัว เป็นผดผื่น แพ้ ผิวหนังแดง

6) อาการทางระบบทางเดินหายใจ เช่น คัดจมูก ไอ จาม คล้ายเป็นภูมิแพ้ หายใจลำบาก แน่นหน้าอก แสบคอ คอแห้ง

2.11.3 วิธีป้องกันออฟฟิศซินโดรม [23]

1) ออกกำลังกายยืดกล้ามเนื้ออย่างสม่ำเสมอ เช่น เล่นโยคะ

2) ปรับสภาพแวดล้อมในการทำงานให้เหมาะสม แสดงดังรูปที่ 2.17 ดังนี้

- จอคอมพิวเตอร์แนวตรงกับหน้า และอยู่เหนือกว่าระดับสายตาเล็กน้อย โดยขอบบนของจอคอมพิวเตอร์อยู่ตรงกับระดับสายตา ในท่านั่งที่รู้สึกสบาย
- จอคอมตั้งห่างกับความยาวเท่ากับความยาวแขน ซึ่งเป็นระยะที่สายตาอ่านได้สบาย
- แป้นพิมพ์วางอยู่ในระดับศอก ทำมุม 90 องศา
- เบาะของเก้าอี้ต่ำกว่าระดับเข่า ปรับให้มีช่องว่างระหว่างขอบเก้าอี้กับขาด้านหลัง
- ปรับเก้าอี้ให้เท้าวางบนพื้นได้พอดี ประมาณ 90 องศา
- ปรับพนักพิงหลังของเก้าอี้ให้รับกับหลังส่วนล่าง หากเก้าอี้ทำไม่ได้ให้ใช้หมอนหนุนหลังแทน

รูปที่ 2.17 การนั่งทำงานที่เหมาะสม [23]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ปรับเปลี่ยนพฤติกรรมการทำงาน

- ยืดเหยียดกล้ามเนื้อระหว่างวัน
- เปลี่ยนอิริยาบถเพื่อให้กล้ามเนื้อได้ผ่อนคลายอย่างน้อย ทุก ๆ 1 ชั่วโมง
- พักสายตาจากจอคอมทุก ๆ 10 นาที
- เปลี่ยนท่านั่งทำงานทุก ๆ 20 นาที
- นั่งตัวตรง หลังชิดขอบด้านในของเก้าอี้
- จับเม้าส์ในตำแหน่งตรง ไม่บิด งอ ข้อมือขึ้นหรือลง

2.11.4 หลังค่อม (Kyphosis) [24]

หลังค่อม (Kyphosis) คือ อาการที่กระดูกสันหลังโค้งนูนมากผิดปกติ มีลักษณะดังรูปที่ 2.18 ในบางรายอาจสังเกตเห็นได้ด้วยตาเปล่า โดยอาจมีอาการอื่น ๆ ร่วมด้วย เช่น รู้สึกปวดหรือตึงบริเวณหลัง เจ็บที่กระดูกสันหลัง ในผู้ที่มีอาการหลังค่อมมากอาจลำบากเวลารับประทานอาหารหรือหายใจ สาเหตุเกิดจากการพัฒนาที่ผิดปกติของกระดูกสันหลังของทารกในครรภ์มารดา หรือการทำร้ายสุขภาพของกระดูกสันหลังด้วยการทำท่าทางหรืออยู่ในอิริยาบถที่ไม่ถูกต้อง หรือจากความเสื่อมของกระดูกสันหลังตามวัย พบได้ในเด็ก วัยรุ่น และในผู้สูงอายุ

รูปที่ 2.18 ลักษณะการหลังค่อม [24]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.4.1 อาการของหลังค่อม

หลังค่อมจะมีอาการที่สังเกตได้ชัดคือ ลักษณะของหลังที่โค้งนูนมากกว่าปกติหรือมีลักษณะที่แปลกไป โดยปกติกระดูกสันหลังตอนบนจะมีลักษณะโค้งตามธรรมชาติอยู่ที่ประมาณ 20-45 องศา แต่ในผู้ที่มีอาการหลังค่อมจะมีความโค้งตั้งแต่ 50 องศาขึ้นไป อาจมีอาการอื่น ๆ ร่วมด้วยขึ้นอยู่กับสาเหตุและองศาความโค้งนูนของหลังในผู้ป่วยแต่ละราย เช่น ไหล่ที่ห่อไปทางด้านหน้า รู้สึกปวดหรือตึงที่บริเวณหลัง รู้สึกเจ็บที่บริเวณกระดูกสันหลังหรือเมื่อยล้า ในผู้ที่มีอาการรุนแรงอาจยากลำบากเวลารับประทานอาหารหรือหายใจ หลังค่อมเกิดขึ้นได้ทุกวัย โดยเฉพาะในเด็กหรือวัยรุ่นในช่วงที่มีการพัฒนาของกระดูกอย่างรวดเร็ว หากพบว่าบุตรหลานมีกระดูกสันหลังตอนบนที่โค้งนูนผิดปกติควรพาไปพบแพทย์

2.11.4.2 สาเหตุของหลังค่อม

หลังค่อมเป็นความผิดปกติที่เกิดขึ้นบริเวณกระดูกสันหลังส่วนอกที่โค้งนูนมากผิดปกติ อาจเป็นผลมาจากหลายสาเหตุ หลังค่อมที่พบทั่วไปแบ่งได้ 3 ประเภทโดยมีรายละเอียดดังต่อไปนี้

- **Postural Kyphosis** มักพบได้ในเด็กผู้หญิงมากกว่าเด็กผู้ชาย เป็นผลมาจากการทำท่าทางที่ไม่ถูกต้องเป็นเวลานานและส่งผลต่อสุขภาพของกระดูกสันหลัง เช่น การงอหลัง การนั่งพิงพนัก การนั่งเอนหลัง หรือการสะพายกระเป๋าที่มีน้ำหนักมาก เป็นต้น ทำให้เส้นเอ็นและกล้ามเนื้อที่ทำหน้าที่ประคองกระดูกสันหลังยืดหรือถ่างออก และดึงให้กระดูกสันหลังผิดรูป แก้ไขได้ด้วยการทำท่าทางที่ถูกต้อง เช่น การยืนตัวตรง การยืดหลัง เป็นต้น และไม่ทำร้ายสุขภาพของกระดูกสันหลัง อาการหลังค่อมประเภทนี้ไม่ทำให้รู้สึกเจ็บ และมักไม่พบอาการหลังค่อมมากขึ้นเมื่อเป็นผู้ใหญ่

- **Scheuermann's Kyphosis** มักพบอาการที่สังเกตเห็นได้ชัดเจนคือ หลังจะงอหรือค่อมเป็นมุมที่ชัดเจน ไม่สามารถแก้ไขได้แม้จะทำท่าทางที่ถูกต้อง เช่น การยืนตัวตรง การยืดหลัง เป็นต้น เหมือนเช่นในผู้ป่วยที่มีอาการหลังค่อมแบบ Postural Kyphosis อาการนี้พบได้ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เด็กวัยรุ่นชายมากกว่าในเด็กวัยรุ่นหญิง โดยเฉพาะในผู้ที่มีลักษณะผอม ซึ่งเป็นผลมาจากปริมาณเลือดไปเลี้ยงกระดูกสันหลังไม่เพียงพอ ทำให้กระดูกสันหลังเจริญเติบโตได้ไม่เต็มที่และเส้นเอ็นรอบบริเวณกระดูกสันหลังมีความหนากว่าปกติ ซึ่งอาจเป็นลักษณะหนึ่งทางพันธุกรรมที่ถ่ายทอดกันภายในครอบครัว พบความผิดปกติที่เกิดขึ้นบริเวณกระดูกสันหลังตอนบน หรือตอนล่าง ในผู้ป่วยบางรายอาจรู้สึกเจ็บที่บริเวณส่วนที่โค้งนูนของกระดูกสันหลัง รวมถึงบริเวณเอว อาการเจ็บจะรุนแรงขึ้นโดยเฉพาะในเวลาทำกิจกรรมต่าง ๆ เช่น การนั่งหรือการยืนเป็นเวลานาน

- **Congenital Kyphosis** เกิดจากภาวะกระดูกสันหลังพัฒนาผิดปกติตั้งแต่ทารกอยู่ในครรภ์มารดา ส่วนมากจะพบว่ามีการเชื่อมรวมกันของกระดูกสันหลังตั้งแต่ 2 ชิ้นขึ้นไป อาจพบอาการที่รุนแรงได้และจำเป็นต้องเข้ารับการรักษาด้วยวิธีการผ่าตัดตั้งแต่ในวัยเด็ก เพื่อยับยั้งการเจริญเติบโตของกระดูกสันหลังที่อาจโค้งนูนไปมากกว่าปกติ ซึ่งอาจเป็นลักษณะหนึ่งทางพันธุกรรมที่ถ่ายทอดกันภายในครอบครัว หรืออาจพบว่าผู้ป่วยมีปัญหาเกี่ยวกับหัวใจหรือไตร่วมด้วย นอกจากนี้ ยังรวมไปถึงปัจจัยอื่น ๆ ที่อาจเป็นสาเหตุของอาการหลังค่อม ได้แก่

- โรคกระดูกพรุน (Osteoporosis) ทำให้กระดูกไม่แข็งแรงและแตกหักได้ง่าย
- โรคกระดูกเสื่อม (Spondylosis) รวมถึงหมอนรองกระดูกและเส้นเอ็นของกระดูกสันหลังที่เสื่อมไปตามวัย
- อาการสไปนา ไบฟิดา (Spina Bifida) เป็นอาการที่กระดูกสันหลังพัฒนาได้ไม่ถูกต้อง
- โรคพาเจท (Paget's Disease) เป็นอาการที่เกิดจากปัญหาของการฟื้นฟูกระดูก ส่งผลให้กระดูกไม่แข็งแรงหรืออาจทำให้ผิดรูปได้
- โรคนิวโรไฟโบรมาโตซิส (Neurofibromatosis) เป็นโรคทางพันธุกรรมชนิดหนึ่งที่มีผลต่อระบบประสาท
- โรคกล้ามเนื้อเสื่อม (Muscular Dystrophy) เป็นภาวะทางพันธุกรรมที่ทำให้กล้ามเนื้อเสื่อมสภาพ และอาจมีโอกาสพิการได้
- วัณโรค เกิดจากการติดเชื้อแบคทีเรียที่มักเกิดขึ้นบริเวณปอดและที่กระดูกสันหลัง
- โรคมะเร็ง ที่เกิดขึ้นบริเวณกระดูกสันหลังหรือลุกลามมาจากเชื้อมะเร็งในอวัยวะอื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การบาดเจ็บที่เกิดขึ้นกับกระดูกสันหลัง
- อายุ เมื่ออายุมากขึ้น กระดูกสันหลังจะโค้งมากขึ้น

นั่งหลังค่อมส่งผลต่อแนวกระดูกสันหลัง การนั่งหลังค่อมเป็นพฤติกรรมที่ไม่ควรมองข้าม นอกจากจะทำให้เสียบุคลิกแล้ว การอยู่ในอิริยาบถนี้นาน ๆ จนติดเป็นนิสัย ยังพาให้แนวกระดูกสันหลังเกิดการเปลี่ยนแปลงโครงสร้างไปจากปกติ (Posture) อาจทำให้กล้ามเนื้ออ่อนแรง นอกจากนี้ยังพาอีกหลายโรคตามมาด้วย เช่น ปวดหลัง หมอนรองกระดูกทับเส้นประสาท โรคกระดูกสันหลังโค้งผิดปกติ (Kyphosis) โรคกระดูกสันหลังคด (Scoliosis) [25]

สัญญาณเตือนก่อนหลังค่อมถาวร หากร่างกายเริ่มมีอาการเจ็บ ปวด ชา ร้าว นั่น อาจเป็นสัญญาณเตือนว่าร่างกายคุณต้องการการบำบัดที่ถูกต้อง ที่สำคัญต้องรู้จักปรับการนั่ง ยืน เดิน นอน ปรับอิริยาบถต่าง ๆ ตามระยะเวลาที่เหมาะสม มีช่วงพักผ่อนบ้าง ไม่นั่งทำงานอย่างต่อเนื่องทั้งวันทั้งคืน และหมั่นออกกำลังกายอย่างสม่ำเสมอ เพื่อให้กล้ามเนื้อได้มีการเคลื่อนไหว จะทำให้บรรเทาอาการปวดได้เป็นอย่างดี [25]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

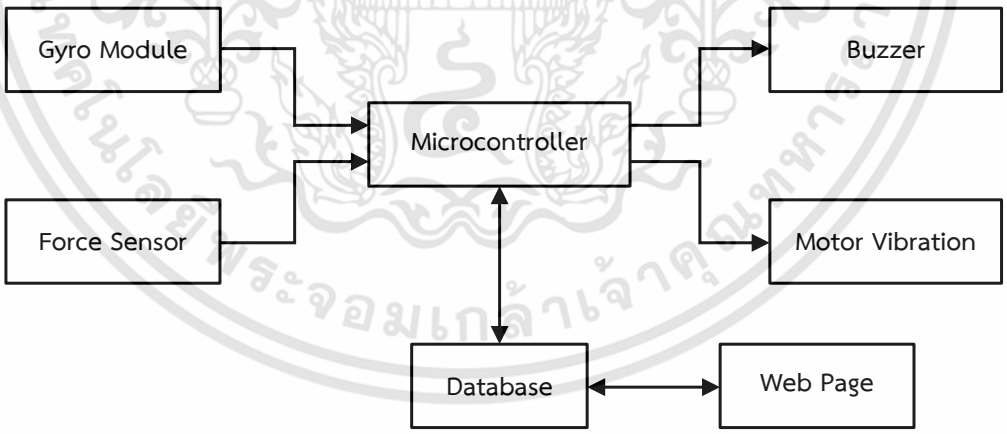
บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

3.1.1 การออกแบบการทำงานของระบบ

ระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม ทำงานผ่านไมโครคอนโทรลเลอร์โดยเชื่อมต่อกับโมดูลวัดความเอียง (Gyro Module) และเซนเซอร์วัดแรงกด (Force Sensitive Resistor) โดยจะแจ้งเตือนหากพบผู้ที่มีแนวโน้มในการนั่งหลังค่อม และนั่งเป็นเวลานานเกินเวลาที่กำหนด ด้วยเสียงผ่านโมดูลเสียงเตือน (Active Buzzer) และการสั่นด้วยมอเตอร์สั่น (Motor Vibration) และไมโครคอนโทรลเลอร์จะนำค่าที่ได้ไปเก็บไว้ที่ฐานข้อมูล และแสดงผลผ่านเว็บเพจ (Web Page) แสดงดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

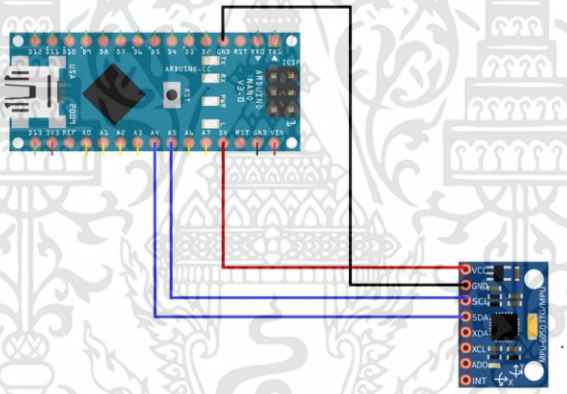
3.1.2 ทำการทดลองต่ออุปกรณ์โมดูลวัดความเอียง (Gyro Module) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

3.1.2.1 วงจรการเชื่อมต่ออุปกรณ์โมดูลวัดความเอียง (Gyro Module) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano) แสดงดังรูปที่ 3.2

- Vcc (Gyro Module) เชื่อมต่อกับ 5V (Arduino Nano)

- SDA (Gyro Module) เชื่อมต่อกับ A4 (Arduino Nano)

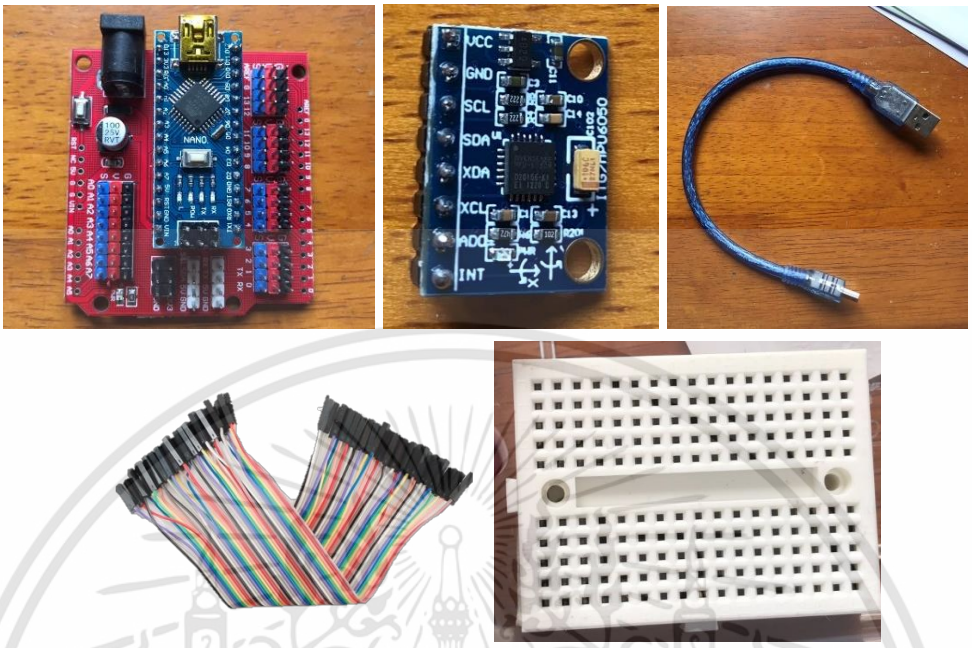
- SCL (Gyro Module) เชื่อมต่อกับ A5 (Arduino Nano)



รูปที่ 3.2 วงจรการเชื่อมต่ออุปกรณ์โมดูลวัดความเอียง (Gyro Module) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

3.1.2.2 ขั้นตอนต่อไปเป็นการต่อโมดูลต่าง ๆ ที่จะใช้ในการทดลองการทำงานของโมดูล โดยอุปกรณ์ที่ใช้มีดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

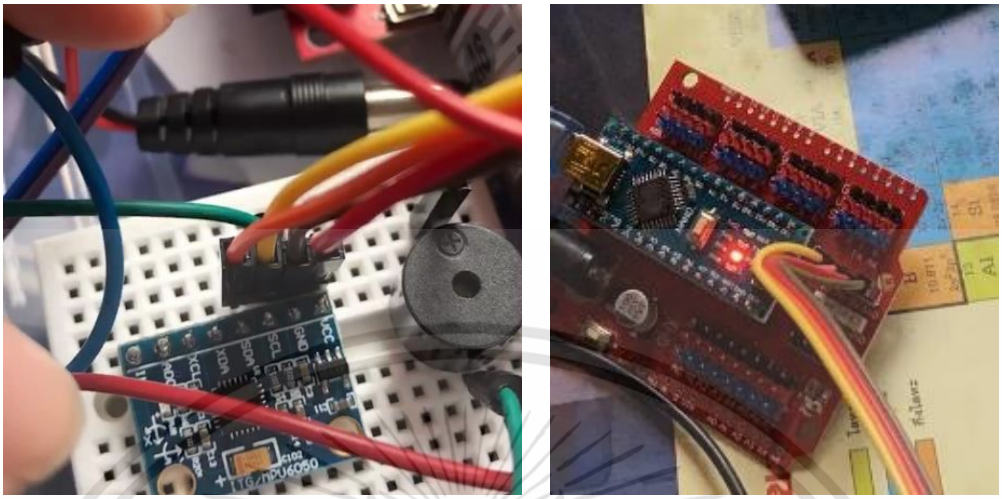


รูปที่ 3.3 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ โมดูลวัดความเอียง สาย Mini USB สายจัมพ์ และโพรโทบอร์ดทดลอง ตามลำดับ

- บอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)
- โมดูลวัดความเอียง (Gyro Module)
- สาย Mini USB
- สายจัมพ์
- โพรโทบอร์ดทดลอง

3.1.2.3 ทำการต่อสาย USB จากบอร์ดไมโครคอนโทรลเลอร์ Arduino Nano เข้าสู่คอมพิวเตอร์ และทำการเชื่อมต่อขาของโมดูลวัดความเอียง สู่บอร์ดไมโครคอนโทรลเลอร์ โดยที่ขาเชื่อมต่อตามตารางที่ 2.3 ซึ่งออกมาเป็นดังรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 การต่อขาของอุปกรณ์โมดูลวัดความเอียง และบอร์ดไมโครคอนโทรลเลอร์ ตามลำดับ

3.1.2.4 เขียนโปรแกรมให้ไมโครคอนโทรลเลอร์

- การเขียนโปรแกรมนั้นจะใช้ไลบรารี่อ้างอิงของ I2CdeV.h [26] ที่ได้มาจาก Github และไลบรารี่ MPU6050.h [26] แสดงดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

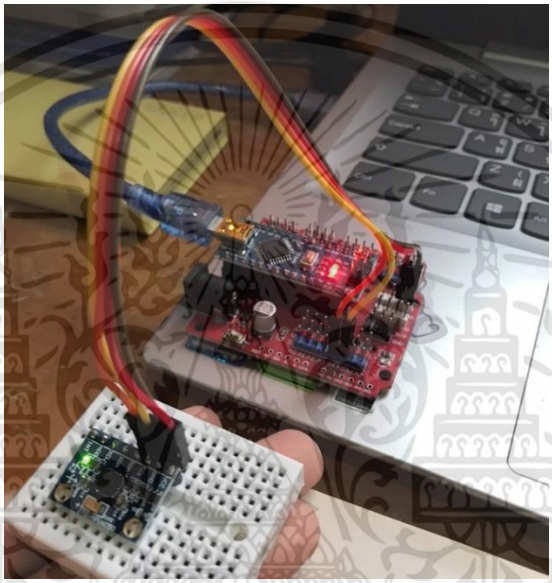
```
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

MPU6050 mpu;

int16_t ax, ay, az;
int16_t gx, gy, gz;
int valx , valy , valz;
char rd;

int prevVal;
int led = 13 ;
int pin11 = 11 , pin10 = 10 ;
int val1 , val2 ;
int valgy1 = 0 , valgy2 = 0;

void setup() {

pinMode(led,OUTPUT) ;
Wire.begin();
Serial.begin(9600);
Serial.println("Initialize MPU");
mpu.initialize();
Serial.println(mpu.testConnection() ? "Connected" : "Connection failed");

}

void loop()
 {

 mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
 valx = map(ax, -17000, 17000, 0, 179);
 valy = map(ay, -17000, 17000, 0, 179);
 valz = map(az, -17000, 17000, 0, 179);

 Serial.print("axis x = ") ;
 Serial.print(valx) ;
 Serial.print(" axis y = ") ;
 Serial.print(valy) ;
 Serial.print(" axis z = ") ;
 Serial.println(valz) ;
 delay(1000);
 }
```

รูปที่ 3.5 โปรแกรมทดลองการทำงานของโมดูลวัดความเอียง (Gyro Module)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.5 ทำการทดลองโปรแกรมที่เขียนไว้

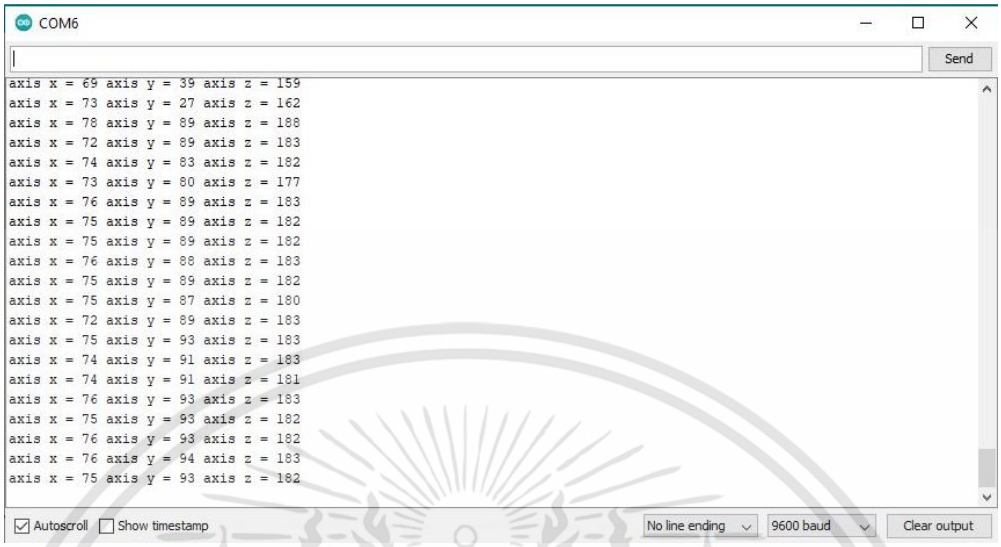
- เมื่อทำการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ จากนั้นนำสาย Mini USB เสียบต่อเข้ากับไมโครคอนโทรลเลอร์ และคอมพิวเตอร์ แล้วทำการอัพโหลด และรันโปรแกรม โมดูลจะแสดงผลดังรูปที่ 3.6



รูปที่ 3.6 การแสดงผลของโมดูลเมื่อทำการอัพโหลด และรันโปรแกรม

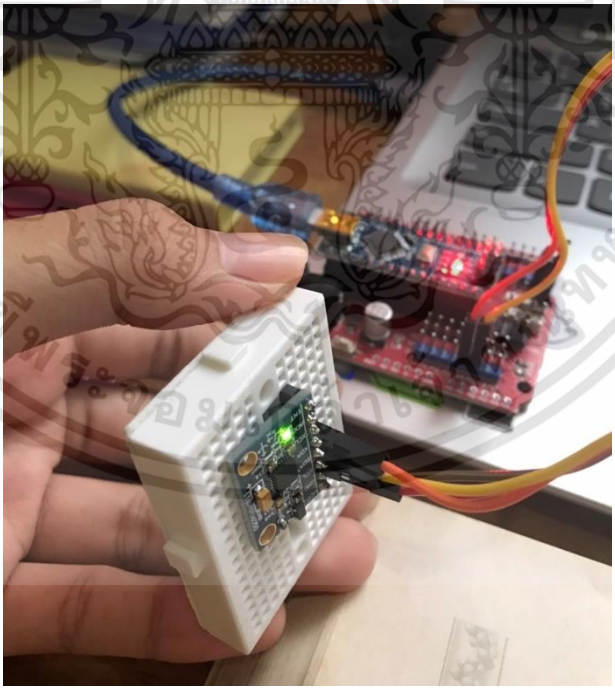
- เมื่อทำการอัพโหลด และรันโปรแกรมสำเร็จโมดูลจะแสดงผลดังรูปที่ 3.6 จากนั้นเรียกดูค่าแกน x y และ z จาก Serial Monitor แสดงดังรูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



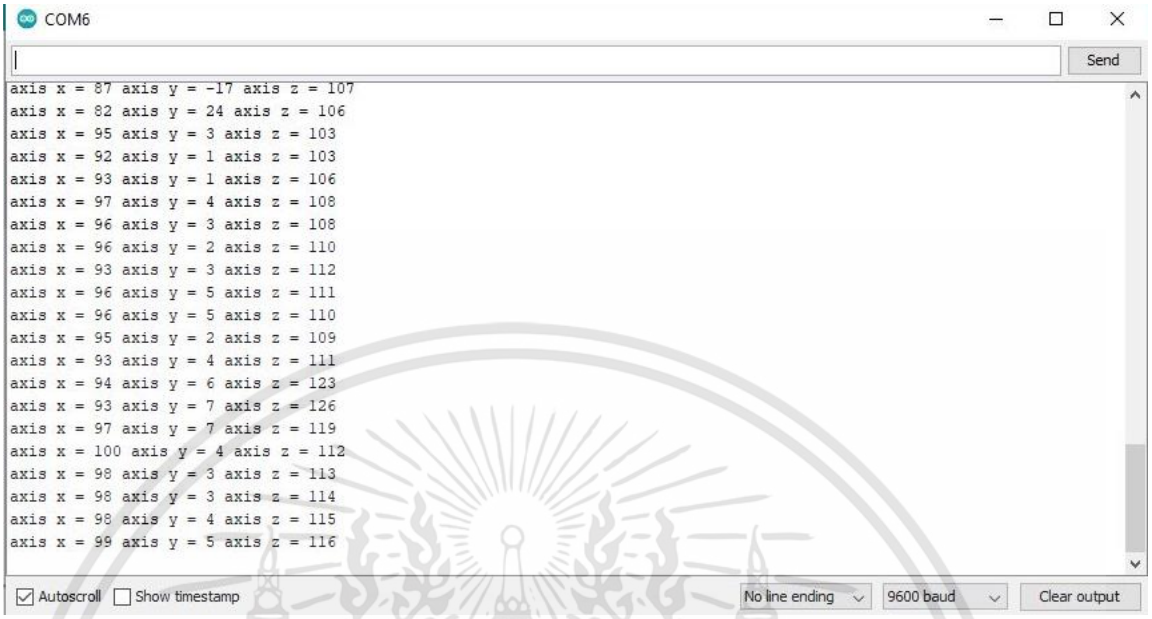
รูปที่ 3.7 การแสดงผลค่าแกน x y และ z จาก Serial Monitor

- ทำการทดลองอีกครั้งโดยทำการเอนโมดูลวัดความเอียง ดังรูปที่ 3.8 และเรียกดูค่าแกน x y และ z จาก Serial Monitor แสดงดังรูปที่ 3.9



รูปที่ 3.8 ทำการทดลองอีกครั้งโดยทำการเอนโมดูลวัดความเอียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



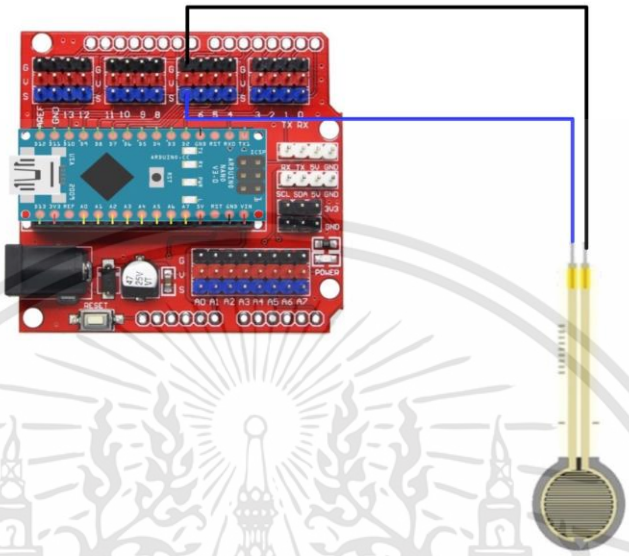
รูปที่ 3.9 การแสดงผลค่าแกน x y และ z จาก Serial Monitor

3.1.3 ทำการทดลองต่อเซนเซอร์วัดแรงกด (Force Sensitive Resistor - FSR402) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

3.1.3.1 วงจรการเชื่อมต่อเซนเซอร์วัดแรงกด (Force Sensitive Resistor - FSR402) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano) แสดงดังรูปที่ 3.10

- ขาบวกของเซนเซอร์วัดแรงกด เชื่อมต่อกับ S7 (Arduino Nano)
- ขาลบของเซนเซอร์วัดแรงกด เชื่อมต่อกับ G7 (Arduino Nano)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 วงจรการเชื่อมต่อเซนเซอร์วัดแรงกด (Force Sensitive Resistor - FSR402) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

3.1.3.2 ขั้นตอนต่อไปเป็นการต่อโมดูลต่าง ๆ ที่จะใช้ในการทดลองการทำงานของเซนเซอร์ โดยอุปกรณ์ที่ใช้มีดังรูปที่ 3.11

รูปที่ 3.11 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ เซนเซอร์วัดแรงกด และสาย Mini USB ตามลำดับ

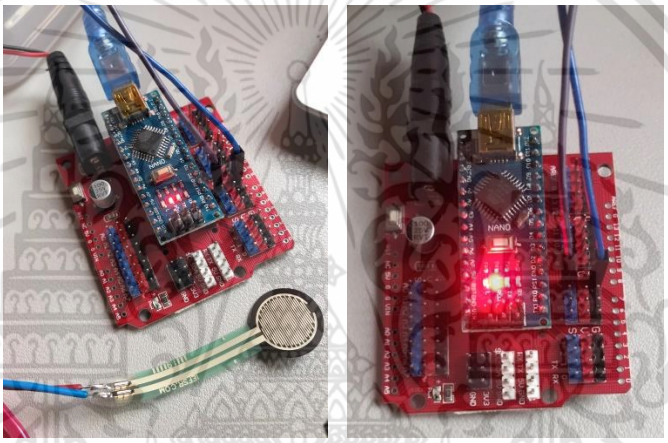
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

- เซนเซอร์วัดแรงกด (Force Sensitive Resistor - FSR402)

- สาย Mini USB

3.1.3.3 ทำการต่อสาย USB จากบอร์ดไมโครคอนโทรลเลอร์ Arduino Nano เข้าสู่คอมพิวเตอร์ และทำการเชื่อมต่อเซนเซอร์วัดแรงกด สู่บอร์ดไมโครคอนโทรลเลอร์ ซึ่งออกมาเป็นดังรูปที่ 3.12



รูปที่ 3.12 การต่อเซนเซอร์วัดแรงกด และบอร์ดไมโครคอนโทรลเลอร์ ตามลำดับ

3.1.3.4 เขียนโปรแกรมให้ไมโครคอนโทรลเลอร์

การเขียนโปรแกรมทดลองการทำงานของเซนเซอร์วัดแรงกดกับไมโครคอนโทรลเลอร์ แสดงดังรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int fsrPin = 0;     // the FSR and 10K pulldown are connected to a0
int fsrReading;     // the analog reading from the FSR resistor divider

void setup(void) {
  // We'll send debugging information via the Serial monitor
  Serial.begin(9600);
}

void loop(void) {
  fsrReading = analogRead(fsrPin);

  Serial.print("Analog reading = ");
  Serial.print(fsrReading);     // the raw analog reading

  // We'll have a few threshholds, qualitatively determined
  if (fsrReading < 10) {
    Serial.println(" - No pressure");
  } else if (fsrReading < 200) {
    Serial.println(" - Light touch");
  } else if (fsrReading < 500) {
    Serial.println(" - Light squeeze");
  } else if (fsrReading < 800) {
    Serial.println(" - Medium squeeze");
  } else {
    Serial.println(" - Big squeeze");
  }
  delay(1000);
}
```

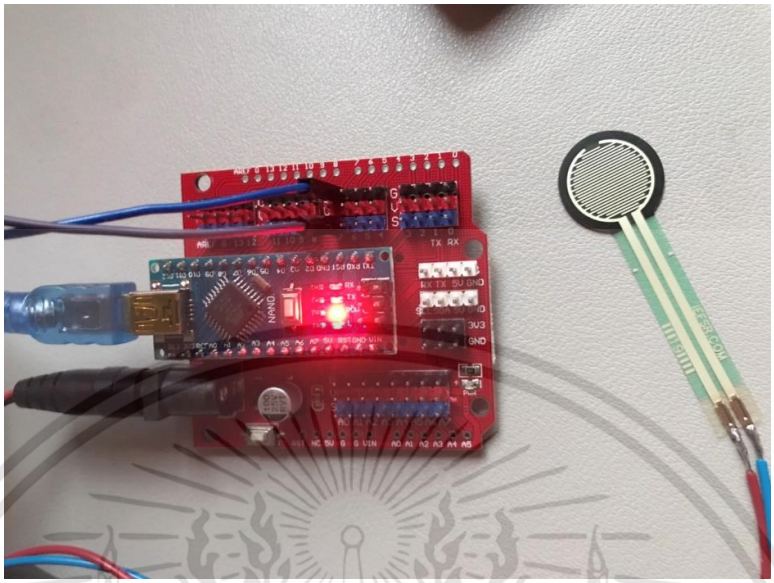
รูปที่ 3.13 โปรแกรมทดลองการทำงานของเซนเซอร์วัดแรงกด

(Force Sensitive Resistor - FSR402)

3.1.3.5 ทำการทดลองโปรแกรมที่เขียนไว้

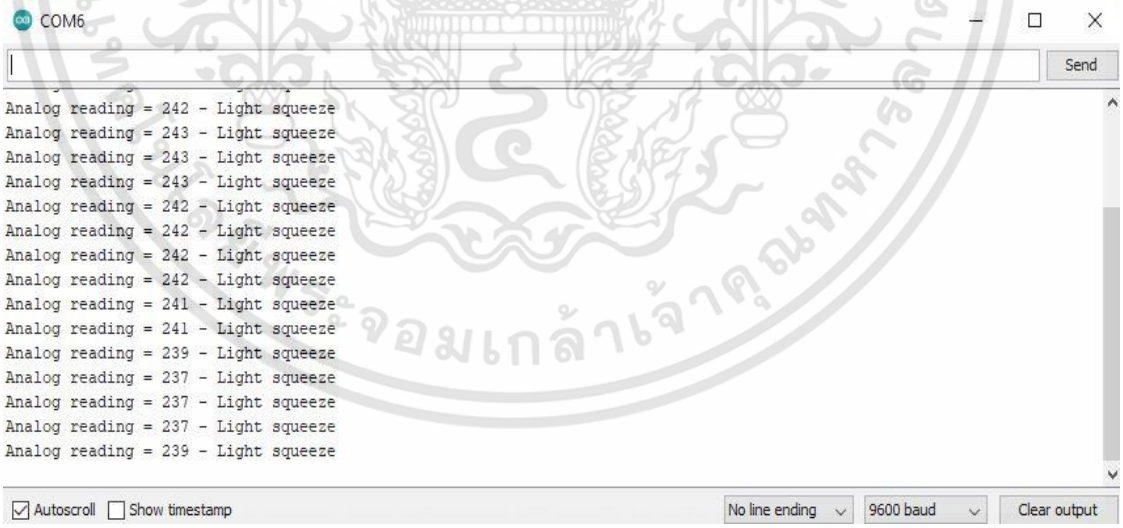
- เมื่อทำการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ จากนั้นนำสาย Mini USB เสียบต่อเข้ากับไมโครคอนโทรลเลอร์ และคอมพิวเตอร์ แล้วทำการอัพโหลด และรันโปรแกรม จะแสดงผลดังรูปที่ 3.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 การแสดงผลของเซนเซอร์เมื่อทำการอัพโหลด และรันโปรแกรม

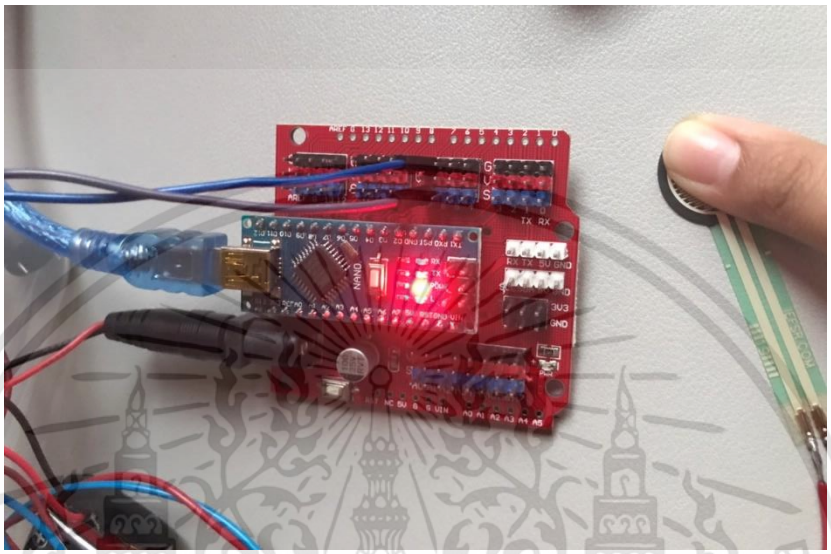
- เมื่อทำการอัพโหลด และรันโปรแกรมสำเร็จจะแสดงผลดังรูปที่ 3.14 จากนั้นเรียกดูค่าวัดแรงกดจาก Serial Monitor แสดงดังรูปที่ 3.15



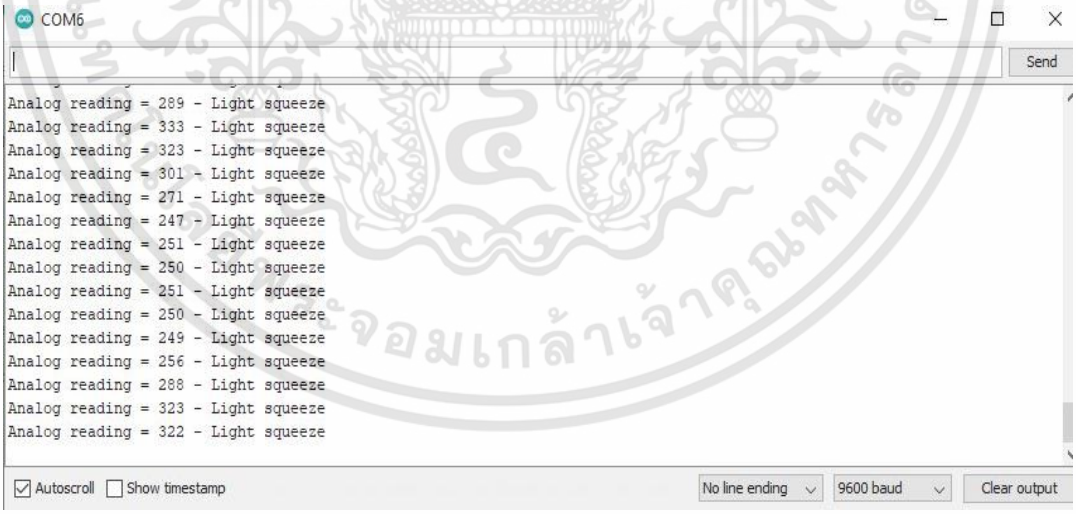
รูปที่ 3.15 การแสดงผลค่าวัดแรงกดจาก Serial Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการทดลองอีกครั้งโดยทำการกดที่เซนเซอร์ ดังรูปที่ 3.16 และเรียกดูค่าวัดแรงกดจาก Serial Monitor แสดงดังรูปที่ 3.17



รูปที่ 3.16 ทำการทดลองอีกครั้งโดยทำการกดที่เซนเซอร์



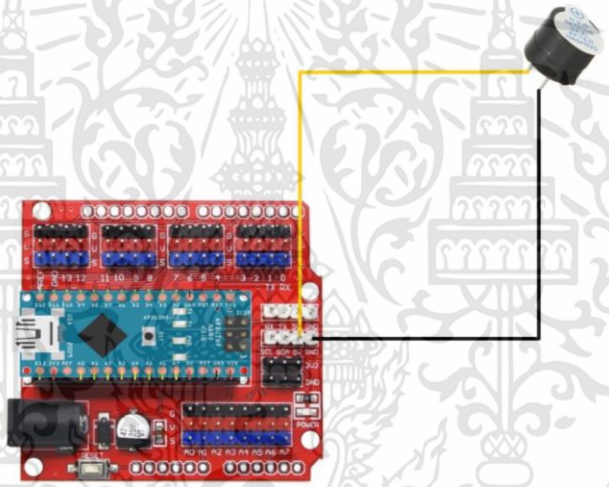
รูปที่ 3.17 การแสดงผลค่าวัดแรงกดจาก Serial Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 ทำการทดลองต่ออุปกรณ์โมดูลเสียงเตือน (Active Buzzer) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

3.1.4.1 วงจรการเชื่อมต่ออุปกรณ์โมดูลเสียงเตือน (Active Buzzer) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano) แสดงดังรูปที่ 3.18

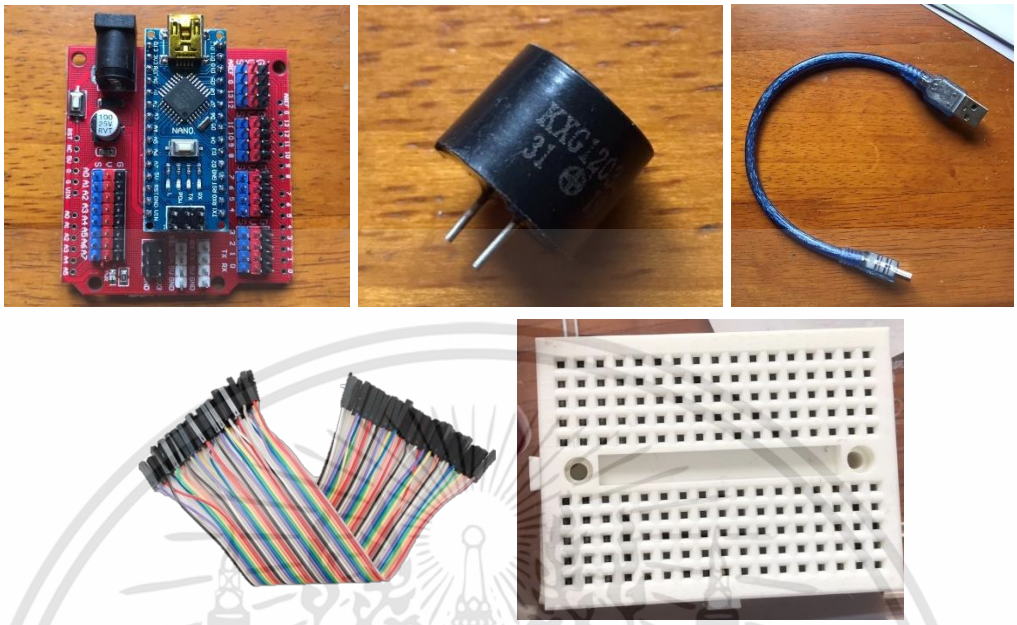
- ขั้วบวกโมดูลเสียงเตือน (Active Buzzer) เชื่อมต่อ 5V (Arduino Nano)
- ขั้วลบโมดูลเสียงเตือน (Active Buzzer) เชื่อมต่อ GND (Arduino Nano)



รูปที่ 3.18 วงจรการเชื่อมต่ออุปกรณ์โมดูลเสียงเตือน (Active Buzzer) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

3.1.4.2 ขั้นตอนต่อไปเป็นการต่อโมดูลต่าง ๆ ที่จะใช้ในการทดลองการทำงานของโมดูล โดยอุปกรณ์ที่ใช้มีดังรูปที่ 3.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

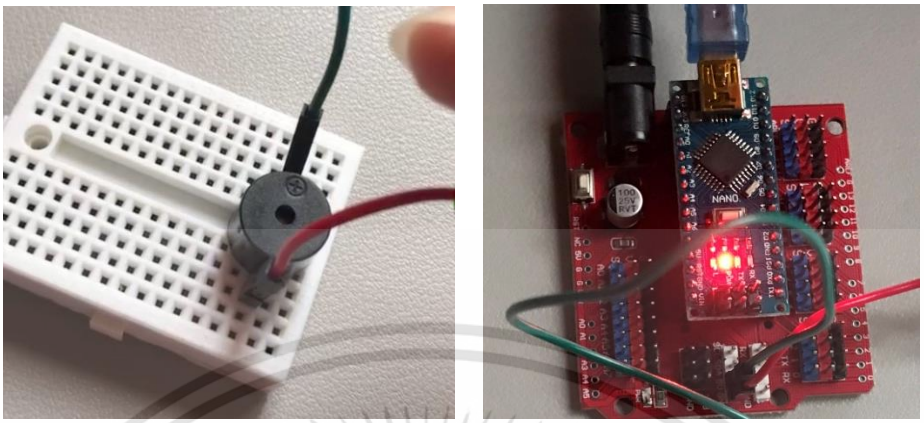


รูปที่ 3.19 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ โมดูลเสียงเตือน สาย Mini USB สายจัมพ์ และโพรโทบอร์ดทดลอง ตามลำดับ

- บอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)
- โมดูลเสียงเตือน (Active Buzzer)
- สาย Mini USB
- สายจัมพ์
- โพรโทบอร์ดทดลอง

3.1.4.3 ทำการต่อสาย USB จากบอร์ดไมโครคอนโทรลเลอร์ Arduino Nano เข้าสู่คอมพิวเตอร์ และทำการเชื่อมต่อโมดูลเสียงเตือนกับโพรโทบอร์ด จากนั้นทำการเชื่อมต่อเข้าสู่บอร์ดไมโครคอนโทรลเลอร์ ซึ่งออกมาเป็นดังรูปที่ 3.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 การต่อขาของอุปกรณ์โมดูลเสียงเตือน และบอร์ดไมโครคอนโทรลเลอร์ ตามลำดับ

3.1.4.4 เขียนโปรแกรมให้ไมโครคอนโทรลเลอร์

การเขียนโปรแกรมทดลองการทำงานของโมดูลเสียงเตือนกับไมโครคอนโทรลเลอร์ แสดงดังรูปที่ 3.21

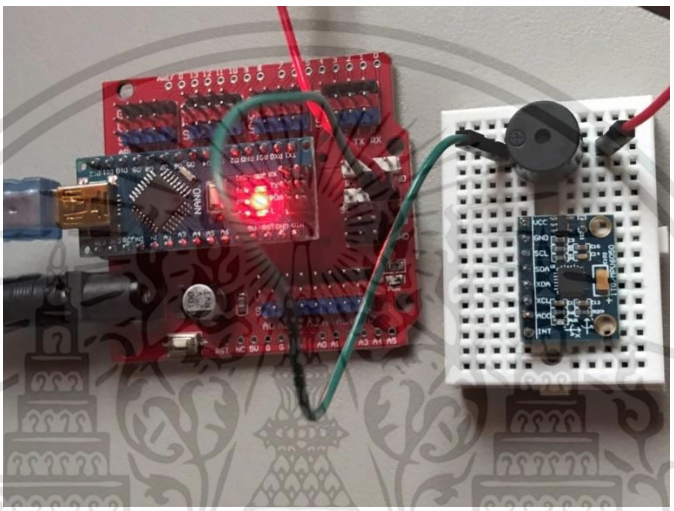
```
int buzzer = 13;

void setup() {
pinMode(buzzer, OUTPUT); }
void loop() {
digitalWrite(buzzer , HIGH); // สร้างเสียงเตือน
delay(1000); // ร้องค้างไว้ 1 วินาที
digitalWrite(buzzer , LOW); //หยุดร้อง
delay(1000); // หยุดร้อง 1 วินาที
}
```

รูปที่ 3.21 โปรแกรมทดลองการทำงานของโมดูลเสียงเตือน (Active Buzzer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4.5 ทำการทดลองโปรแกรมที่เขียนไว้

- เมื่อทำการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ จากนั้นนำสาย Mini USB เสียบต่อเข้ากับไมโครคอนโทรลเลอร์ และคอมพิวเตอร์ แล้วทำการอัพโหลด และรันโปรแกรม โมดูลจะแสดงผลดังรูปที่ 3.22



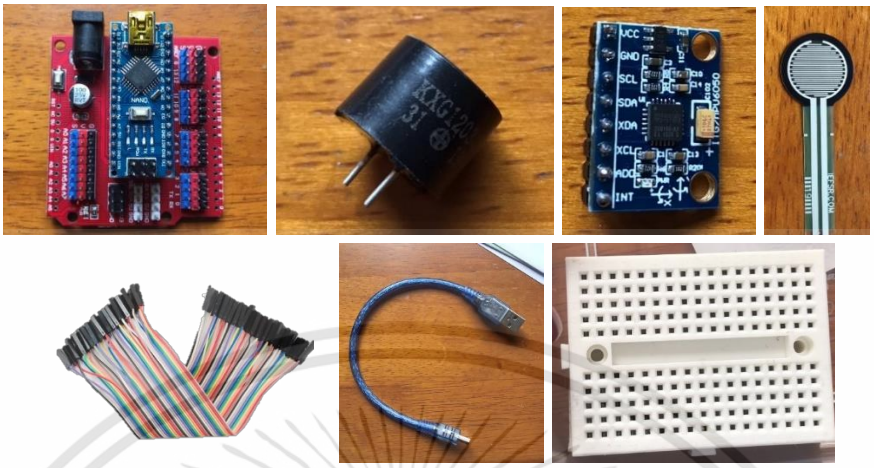
รูปที่ 3.22 การแสดงผลของโมดูลเมื่อทำการอัพโหลด และรันโปรแกรม

- เมื่อทำการอัพโหลด และรันโปรแกรมสำเร็จโมดูลจะแสดงผลดังรูปที่ 3.22 และจะมีเสียงแจ้งเตือนจากตัวโมดูลนาน 1 วินาทีตามที่ได้เขียนในโปรแกรมไว้ ซึ่งสามารถปรับเปลี่ยนเวลาเสียงแจ้งเตือนได้ตามต้องการ

3.1.5 ทำการต่ออุปกรณ์โมดูลวัดความเอียง (Gyro Module) เซนเซอร์วัดแรงกด และโมดูลเสียงเตือน (Active Buzzer) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

3.1.5.1 ขั้นตอนต่อไปเป็นการต่อโมดูลต่าง ๆ ที่จะใช้ในการทำงานของระบบ โดยอุปกรณ์ที่ใช้มีดังรูปที่ 3.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

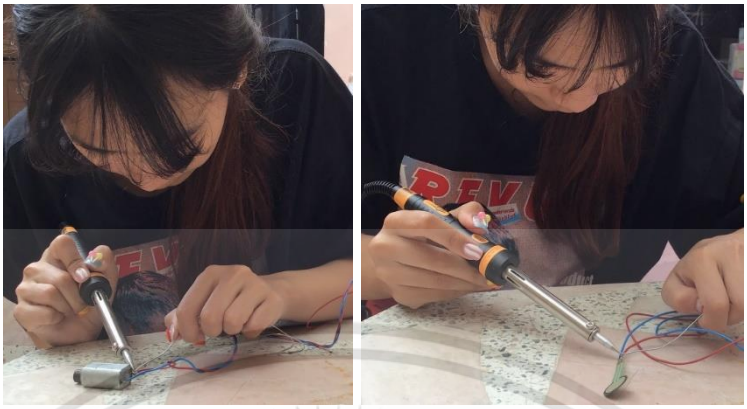


รูปที่ 3.23 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ โมดูลเสียงเตือน โมดูลวัดความเอียง
เซนเซอร์วัดแรงกดสายจัมพ์ สายMini USB
และโพรโทบอร์ดทดลอง ตามลำดับ

- บอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)
- โมดูลวัดความเอียง (Gyro Module)
- โมดูลเสียงเตือน (Active Buzzer)
- เซนเซอร์วัดแรงกด (Force Sensitive Resistor - FSR402)
- สาย Mini USB
- สายจัมพ์
- โพรโทบอร์ดทดลอง

3.1.5.2 ทำการบัดกรีอุปกรณ์ เซนเซอร์วัดแรงดันกับสายไฟ มอเตอร์สั่นกับสายไฟ และอุปกรณ์อื่น ๆ แสดงดังรูปที่ 3.24

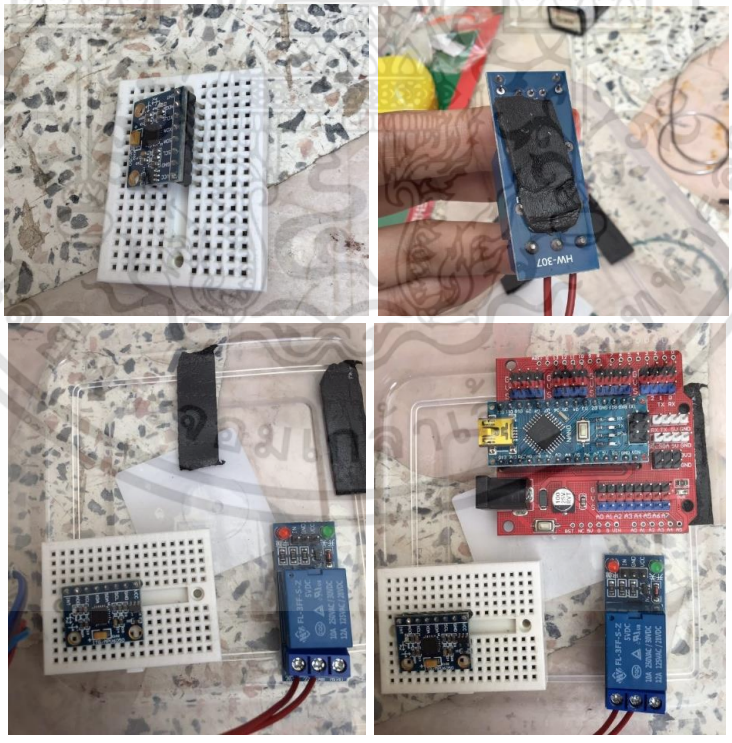
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 การบัดกรีอุปกรณ์ มอเตอร์สั่นกับสายไฟ และเซนเซอร์วัดแรงกดกับสายไฟ

3.1.6 ทำการประกอบอุปกรณ์ต่าง ๆ ลงกล่อง

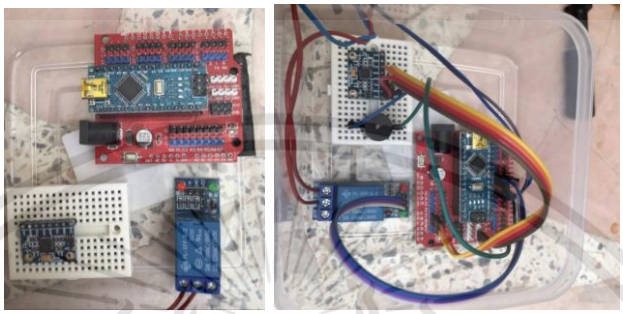
- นำอุปกรณ์ต่าง ๆ มาประกอบใส่ลงในกล่อง โดยใช้กาวสองหน้ามาติดที่อุปกรณ์เพื่อให้อุปกรณ์ยึดติดอยู่กับตัวกล่อง ดังรูปที่ 3.25



รูปที่ 3.25 ประกอบอุปกรณ์ลงกล่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6.1 ทำการต่อสาย USB จากบอร์ดไมโครคอนโทรลเลอร์ Arduino Nano เข้าสู่คอมพิวเตอร์ และทำการเชื่อมต่อโมดูลวัดความเอียง และโมดูลเสียงเตือนกับโพรโทบอร์ด จากนั้นทำการเชื่อมต่อเข้าสู่บอร์ดไมโครคอนโทรลเลอร์ ซึ่งออกมาเป็นดังรูปที่ 3.26



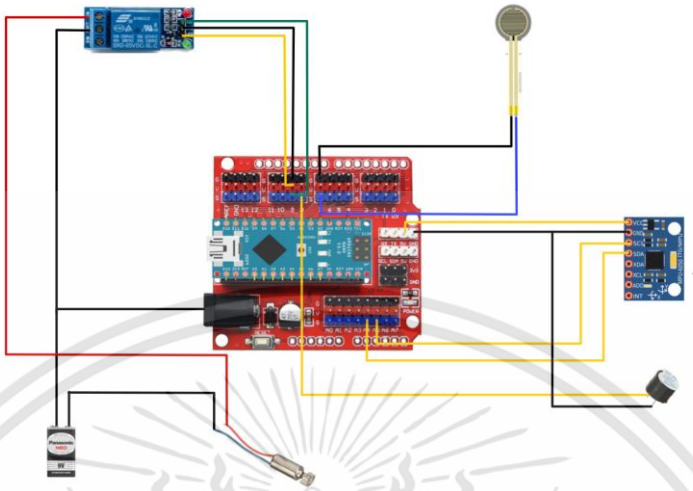
รูปที่ 3.26 การต่ออุปกรณ์เข้ากับบอร์ดไมโครคอนโทรลเลอร์ Arduino Nano

3.1.7 ทำการทดลองเขียนโปรแกรมระบบการแจ้งเตือนผู้นั่งหลังค่อม และผู้ที่นั่งเกินเวลาที่กำหนด

3.1.7.1 การออกแบบวงจรระบบการแจ้งเตือนผู้ที่นั่งหลังค่อม และผู้ที่นั่งเกินเวลาที่กำหนด แสดงดังรูปที่ 3.27

- Vcc (Gyro Module) เชื่อมต่อกับ 5V (Arduino Nano)
- SDA (Gyro Module) เชื่อมต่อกับ A4 (Arduino Nano)
- SCL (Gyro Module) เชื่อมต่อกับ A5 (Arduino Nano)
- ขาบวกของเซนเซอร์วัดแรงกด เชื่อมต่อกับ S7 (Arduino Nano)
- ขาลบของเซนเซอร์วัดแรงกด เชื่อมต่อกับ G7 (Arduino Nano)
- ขั้วบวกโมดูลเสียงเตือน (Active Buzzer) เชื่อมต่อ S8 (Arduino Nano)
- ขั้วลบโมดูลเสียงเตือน (Active Buzzer) เชื่อมต่อ GND (Arduino Nano)
- IN (Relay 5V) เชื่อมต่อ S9 (Arduino Nano)
- Vcc (Relay 5V) เชื่อมต่อ V9 (Arduino Nano)
- GND (Relay 5V) เชื่อมต่อ G9 (Arduino Nano)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.27 วงจรระบบการแจ้งเตือนผู้ที่นั่งหลังค่อม และผู้ที่นั่งเกินเวลาที่กำหนด

3.1.7.2 ทำการทดลองเขียนโปรแกรมโดยรวมของโมดูลวัดความเอียง (Gyro Module) เซนเซอร์วัดแรงกด และโมดูลเสียงเตือน (Active Buzzer) แสดงดังรูปที่ 3.28

```
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#include "Wire.h"

MPU6050 mpu;
int16_t ax, ay, az;
int16_t gx, gy, gz;
int valx , valy , valz;
char rd;
int prevVal;


int val1 , val2 ;
int valgy1 = 0 , valgy2 = 0;


#define sit 7
#define buzzer 8
#define motor 9

int alarm1 = 0;
int alarm2 = 0;
int count1 = 0;
int count2 = 0;
void setup() {
  Serial.begin(9600);
  pinMode(sit, INPUT);

  Serial.begin(9600);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Serial.begin(9600);
  Serial.println("Initialize MPU");
  mpu.initialize();
  Serial.println(mpu.testConnection() ? "Connected" : "Connection failed");

  pinMode(motor, OUTPUT);
  pinMode(buzzer, OUTPUT);
  digitalWrite(motor, HIGH);
  digitalWrite(buzzer, LOW);
}

void loop() {


  Serial.println( digitalRead(sit));

  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  valx = map(ax, -17000, 17000, 0, 179);
  valy = map(ay, -17000, 17000, 0, 179);
  valz = map(az, -17000, 17000, 0, 179);
  Serial.print("axis x = ") ;
  Serial.print(valx) ;
  Serial.print(" axis y = ") ;
  Serial.print(valy) ;
  Serial.print(" axis z = ") ;
  Serial.println(valz) ;


if ( valy >=  7) {
  Serial.println("          y > !!! ");
  count1 += 1;
  Serial.println("                    count 1 = " + String(count1));
  if ( count1 >= 10 )
  {
    alarm1 = 1;
  }
}
if (valy < 7 ) {
  Serial.println("          < y");
  count1 = 0;
  alarm1 = 0;
}
/*
  if (valx < 50) {
   Serial.println("          <  x ");
  }
  if (valz > 150) {
   Serial.println("          z > ");
  }
  //-------------------------------------------------------------
*/
if (  digitalRead(sit) == LOW )
{
  count2 += 1;
  Serial.println("               count 2 = " + String(count2));
  if ( count2 >= 10 )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      alarm2 = 1;
    }
  }
  else
  {
    count2 = 0;
    alarm2 = 0;
  }
  //--------------------------------------------------------------------------
  if ( alarm1 == 1 | alarm2 == 1 )
  {
    Serial.println("                          !!!!!!!!!!!!!! ");
    digitalWrite(buzzer, HIGH);
    digitalWrite(motor, LOW);
  }
  else
  {
    Serial.println("                          ..............  ");
    digitalWrite(buzzer, LOW);
    digitalWrite(motor, HIGH);
  }

  //--------------------------------------------------------------------------
  delay(1000);
}
```

รูปที่ 3.28 โปรแกรมทดลองการทำงานโดยรวมของโมดูลวัดความเอียง (Gyro Module) เซนเซอร์วัดแรงกด และโมดูลเสียงเตือน (Active Buzzer)

3.1.8 การออกแบบฐานข้อมูลในการจัดเก็บข้อมูล

1) ทำการทดลองใช้งาน NodeMCU-32S 38 pins

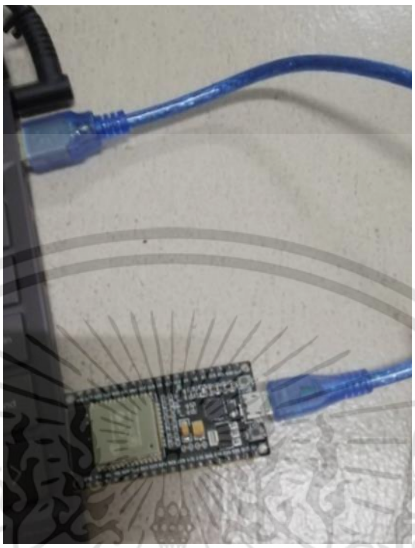
- ทำการเขียนโปรแกรมทดสอบการทำงานดังรูปที่ 3.29 ว่า NodeMCU-32S 38 pins สามารถใช้งานได้หรือไม่

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);     // Initialize the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, LOW);   // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  delay(1000);                      // Wait for a second
  digitalWrite(LED_BUILTIN, HIGH);  // Turn the LED off by making the voltage HIGH
  delay(2000);                      // Wait for two seconds (to demonstrate the active low LED)
}
```

รูปที่ 3.29 เขียนโปรแกรมเพื่อทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

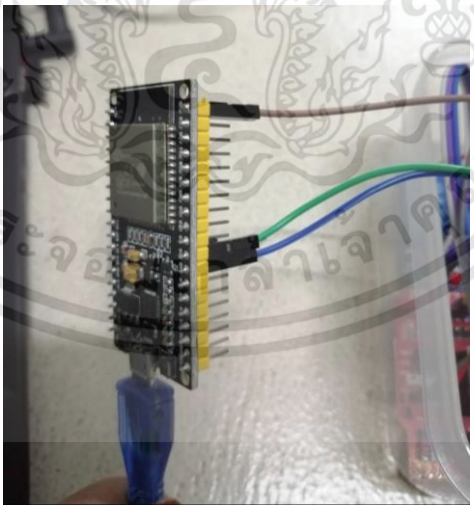
- เมื่อเขียนโปรแกรมเสร็จ ทำการอัพโหลดและรันโปรแกรม แสดงดังรูปที่ 3.30



รูปที่ 3.30 ทำการอัพโหลดและรันโปรแกรม

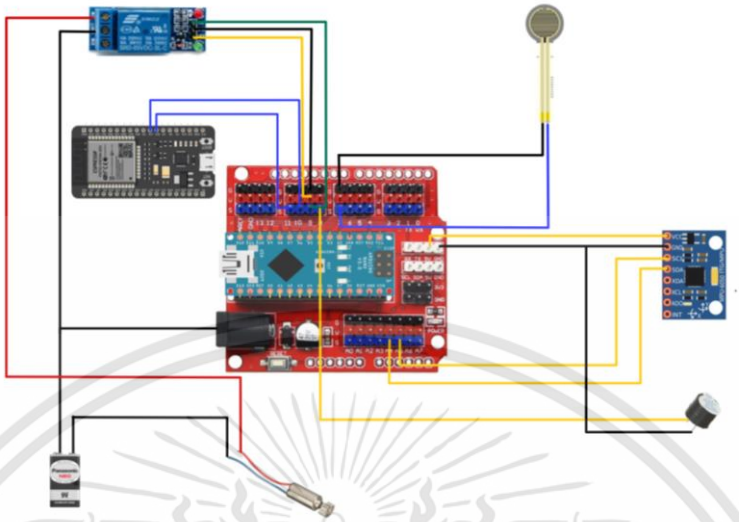
2) ทำการเชื่อมต่อ Arduino Nano กับ NodeMCU-32S

- ทำการเชื่อมต่อ Arduino Nano กับ NodeMCU-32S แสดงดังรูปที่ 3.31 และ 3.32



รูปที่ 3.31 ขาที่ใช้เชื่อมต่อกับ Arduino Nano ของ NodeMCU-32S

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.32 การเชื่อมต่อ Arduino Nano กับ NodeMCU-32S

3) ทำการทดลองเขียนโปรแกรมเชื่อมต่อรับส่งข้อมูลระหว่าง Arduino Nano และ NodeMCU-32S

- ทำการเขียนโปรแกรมลงในบอร์ด Arduino Nano เพื่อเชื่อมต่อ Arduino Nano กับ NodeMCU-32S แสดงดังรูปที่ 3.33

```
#include<SoftwareSerial.h>
SoftwareSerial mySerial (10, 11);
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#include "Wire.h"

MPU6050 mpu;
int16_t ax, ay, az;
int16_t gx, gy, gz;
int valx , valy , valz;
char rd;
int prevVal;


int val1 , val2 ;
int valgy1 = 0 , valgy2 = 0;


#define sit 7
#define buzzer 8
#define motor 9

int alarm1 = 0;
int alarm2 = 0;
int count1 = 0;
int count2 = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

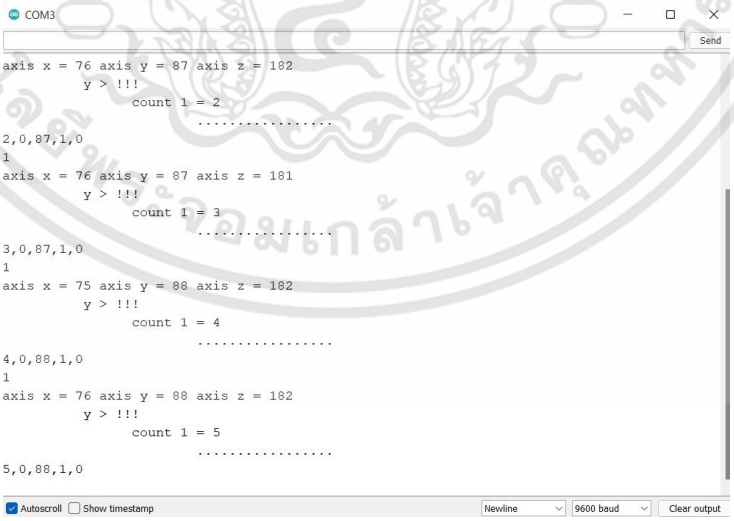
```
int counts = 0;
int county = 0;
int cmotor = 0;
void setup() {
  Serial.begin(9600);
  pinMode(sit, INPUT_PULLUP);

  Serial.begin(9600);
  Serial.println("Initialize MPU");
  mpu.initialize();
  Serial.println(mpu.testConnection() ? "Connected" : "Connection failed");

  pinMode(motor, OUTPUT);
  pinMode(buzzer, OUTPUT);
  digitalWrite(motor, HIGH);
  digitalWrite(buzzer, LOW);
  mySerial.begin(9600);
}

void loop() {


  Serial.println( digitalRead(sit));

  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  valx = map(ax, -17000, 17000, 0, 179);
  valy = map(ay, -17000, 17000, 0, 179);
  valz = map(az, -17000, 17000, 0, 179);
  Serial.print("axis x = ") ;
  Serial.print(valx) ;
  Serial.print(" axis y = ") ;
  Serial.print(valy) ;
  Serial.print(" axis z = ") ;
  Serial.println(valz) ;


  if ( valy >=  7) {
    Serial.println("          y > !!! ");
    county1 += 1;
    Serial.println("                count 1 = " + String(county1));
    county = county1;
    if ( county1 >= 10 )
    {
      alarm1 = 1;
    }
  }
  if (valy < 7 ) {
    Serial.println("       < y");
    county1 = 0;
    county = county1;
    alarm1 = 0;
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (  digitalRead(sit) == LOW )
{
  count2 += 1;
  Serial.println("               count 2 = " + String(count2));
  counts = count2;
  if ( count2 >= 10 )
  {
    alarm2 = 1;
  }
}
else
{
  count2 = 0;
  counts = count2;
  alarm2 = 0;
}
//--------------------------------------------------------------------------
if ( alarm1 == 1 | alarm2 == 1 )
{
  Serial.println("                        !!!!!!!!!!!!! ");
  digitalWrite(buzzer, HIGH);
  digitalWrite(motor, LOW);
  cmotor = 1;
}
else
{
  Serial.println("                        .............  ");
  digitalWrite(buzzer, LOW);
  digitalWrite(motor, HIGH);
  cmotor = 0;
}

//--------------------------------------------------------------------------
//count, &gyro, &sit, &motor
Serial.println( String(county) + "," +String(counts) + "," + String(valy)  + "," + String(digitalRead(sit))   + "," + String(cmotor));
mySerial.println( String(county) + "," +String(counts) + "," + String(valy)  + "," + String(digitalRead(sit))   + "," + String(cmotor));
delay(1000);
}
```

รูปที่ 3.33 โปรแกรมเชื่อมต่อ Arduino Nano กับ NodeMCU-32S

- ทำการอัพโหลด และรันโปรแกรม จากนั้นเรียกดูค่าจาก Serial Monitor แสดงดังรูปที่ 3.34



รูปที่ 3.34 ค่าที่แสดงผ่าน Serial Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ทำการเชื่อมต่อ NodeMCU-32S กับ Firebase เพื่อจัดเก็บข้อมูลลงในฐานข้อมูล

- ทำการเขียนโปรแกรมลงในบอร์ด ESP32 เพื่อเชื่อมต่อ NodeMCU-32S กับ Firebase แสดงดังรูปที่ 3.35

```
#include <WiFi.h>
#include <WiFiClientSecure.h>


#include <IOXhop_FirebaseESP32.h>
#include <time.h>
// Set these to run example.
#define FIREBASE_HOST "https://database1-bba99-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "DvX1MDc7xhKYfZzShihiuyPnrTL1KKSnd13fDV6f"
#define WIFI_SSID "M" // ชื่อ wifi
#define WIFI_PASSWORD "mmmmmmmm" // รหัส wifi

//Time
char ntp_server1[20] = "pool.ntp.org";
char ntp_server2[20] = "time.nist.gov";
char ntp_server3[20] = "time.uni.net.th";
int timezone = 7 * 3600;

int dst = 0;
int HH, MM, sec;
int county = 0;
int counts = 0;

int starts = 0;
int gyro;
int sit;
int motor;

int se;
char datas[50];
String getdata;
bool stringComplete = false;  // whether the string is complete

void setup()
{
  Serial.begin(9600);


  // เริ่มเชื่อมต่ออายฟาย
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  // แสดงคำว่า "Start WiFi Connecting.." ในซีเรียลคอนโซล
  Serial.print("Start WiFi Connecting..");

  // ถ้าไม่สามารถเชื่อมต่ออายฟายได้ให้แสดง "." ในซีเรียลคอนโซล
  // ลูปเช็คสถานะวายฟายทุกๆ ครึ่งวินาที (500 Millisecound)
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  // แสดงคำว่า "WiFi Connected:" และหมายเลข IP ที่ได้รับในซีเรียลคอนโซล
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
  Serial.println();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  configTime(timezone, dst, ntp_server1, ntp_server2, ntp_server3);
  Serial.println("\nWaiting for time");
  while (!time(nullptr)) {
    Serial.print(".");
    delay(1000);
  }
  Serial.println("");

  // เริ่มเชื่อมต่อ Firebase
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

  Serial2.begin(9600);
}


void loop ()
{

  if ( Serial2.available())
  {
    String getdata = Serial2.readString();
    Serial.println(" getdata =" + String(getdata));
    getdata.toCharArray(datas, 50); //count gyro sit motor
    int n = sscanf(datas, "%d,%d,%d,%d,%d", &county, &counts, &gyro, &sit, &motor );
    Serial.print(F("n="));
    Serial.println(n);
    Serial.print(F("        county="));
    Serial.print(county);
    Serial.print(F("        counts="));
    Serial.print(counts);
    Serial.print(F("        gyro="));
    Serial.print(gyro);
    Serial.print(F("        sit="));
     if ( sit == 0 )
    {
      se=1;
    }
    else
    {
      se=0;
    }
    Serial.print(se);
    Serial.print(F("        motor="));
    Serial.print(motor);

    Serial.println();
```

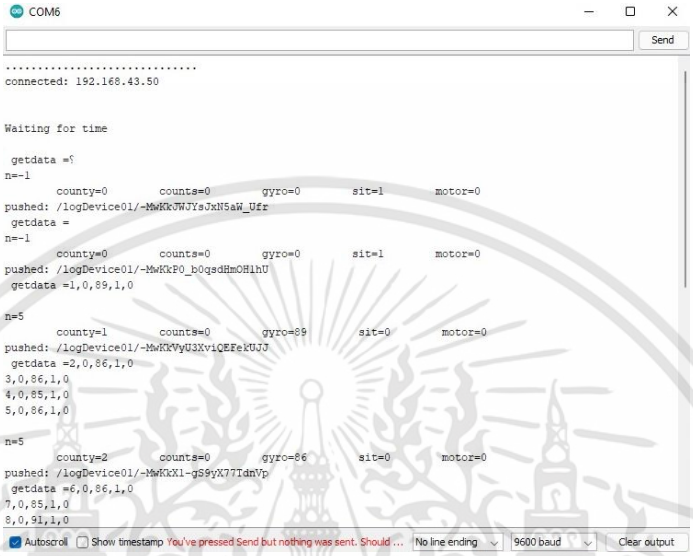
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    starts = 1;
  } // end Serial2

  NowString();
  if ( starts == 1 )
  {

    StaticJsonBuffer<200> jsonBuffer;
    JsonObject& root = jsonBuffer.createObject();
    root["count_gyro"] = county;
    root["count_sit"] = counts;
    root["gyro"] = gyro;
    root["sit"] = se;
    root["motor"] = motor;
    root["time"] = NowString();


    // set value
    String name = Firebase.push("logDevice01", root);
    // handle error
    if (Firebase.failed()) {
      Serial.print("setting /number failed:");
      Serial.println(Firebase.error());
      return;
    }
    Serial.print("pushed: /logDevice01/");
    Serial.println(name);


   delay(1000);
   starts = 0;
  }


}// end loop

String NowString() {
  time_t now = time(nullptr);
  struct tm* p_tm = localtime(&now);
  String timeNow = "";
  HH = String(p_tm->tm_hour).toInt();
  MM = String(p_tm->tm_min).toInt();
  sec = String(p_tm->tm_sec).toInt();
  timeNow += String(p_tm->tm_hour);
  timeNow += ":";
  timeNow += String(p_tm->tm_min);
  timeNow += ":";
  timeNow += String(p_tm->tm_sec);
  timeNow += " ";
  timeNow += String(p_tm->tm_mday);
  timeNow += "-";
  timeNow += String(p_tm->tm_mon + 1);
  timeNow += "-";
  timeNow += String(p_tm->tm_year + 1900);
  return timeNow;
}
```

รูปที่ 3.35 โปรแกรมการเชื่อมต่อ NodeMCU-32S กับ Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

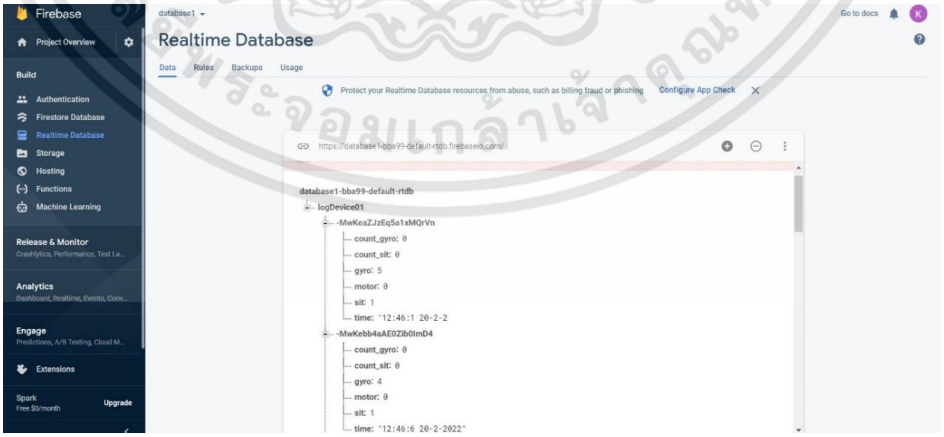
- ทำการอัพโหลด และรันโปรแกรม จากนั้นเรียกดูค่าจาก Serial Monitor แสดงดังรูปที่ 3.36



รูปที่ 3.36 ค่าที่แสดงผ่าน Serial Monitor

5) ส่งข้อมูล และจัดเก็บข้อมูลไว้ในฐานข้อมูล

- เมื่อทำการเขียนโปรแกรมจากรูปที่ 3.35 สำเร็จ จะสามารถส่งข้อมูลที่ได้เขียนโปรแกรมไว้ไปเก็บข้อมูลค่าต่าง ๆ ไว้ใน Firebase ซึ่งเป็นฐานข้อมูลแบบเรียลไทม์ ซึ่งจะแสดงค่า และจัดเก็บข้อมูลไว้ดังรูปที่ 3.37

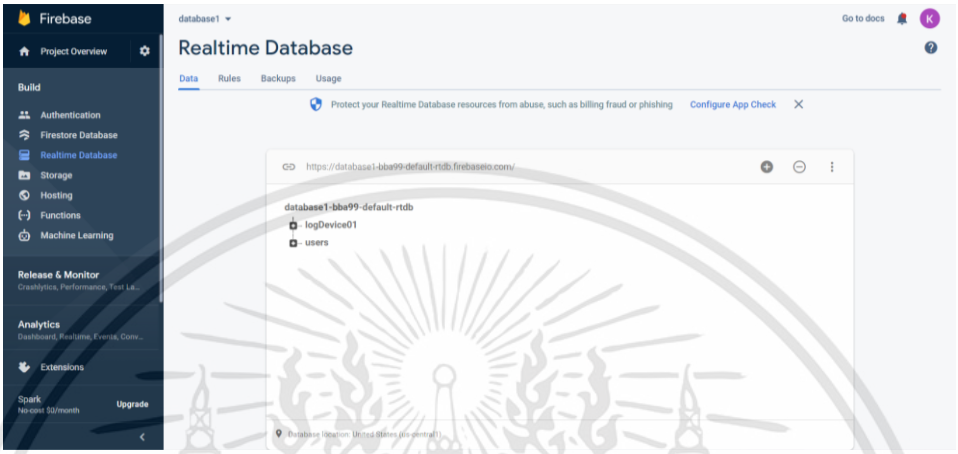


รูปที่ 3.37 ค่าต่าง ๆ ที่แสดง และเก็บข้อมูลไว้ใน Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) เก็บข้อมูลผู้ใช้งาน (Users) และ NodeMCU

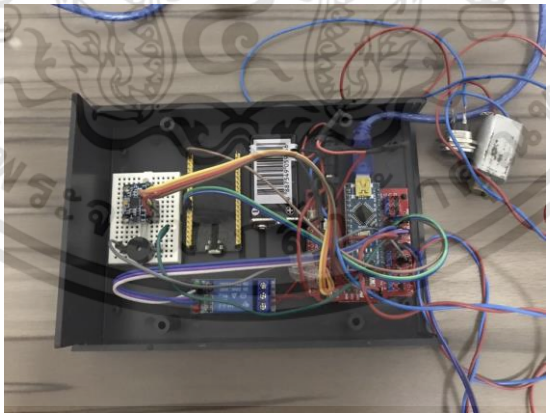
- เก็บข้อมูลผู้ใช้งาน และ NodeMCU ที่ ฐานข้อมูล Firebase แสดงดังรูปที่ 3.38



รูปที่ 3.38 การเก็บข้อมูลผู้ใช้งาน และ NodeMCU

3.1.9 ทำการติดกล่องที่ใส่อุปกรณ์ต่าง ๆ เข้ากับสายคาดหลัง

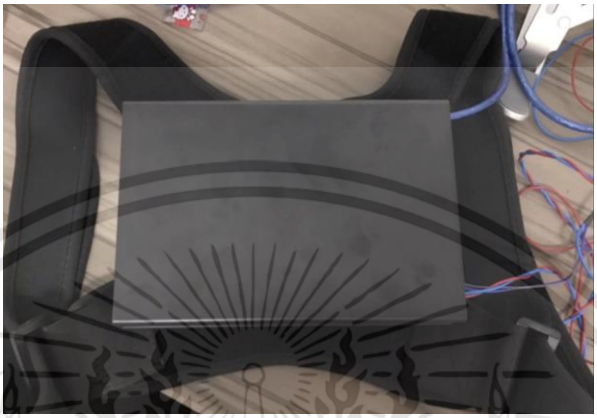
- ทำการประกอบ และใส่อุปกรณ์ทั้งหมดลงในกล่อง แสดงดังรูปที่ 3.39



รูปที่ 3.39 ทำการประกอบ และใส่อุปกรณ์ทั้งหมดลงในกล่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- นำกล่องที่ใส่อุปกรณ์ต่าง ๆ เรียบร้อยแล้วมาติดกับสายคาดหลังเพื่อสะดวกต่อการทดลอง และใช้งาน แสดงดังรูปที่ 3.40



รูปที่ 3.40 นำกล่องติดกับสายคาดหลัง

3.1.10 ทำการทดลองสวมใส่ชิ้นงาน

- ทำการทดลองสวมใส่ชิ้นงานที่ได้ทำการประกอบเสร็จสิ้นแล้ว ดังรูปที่ 3.41

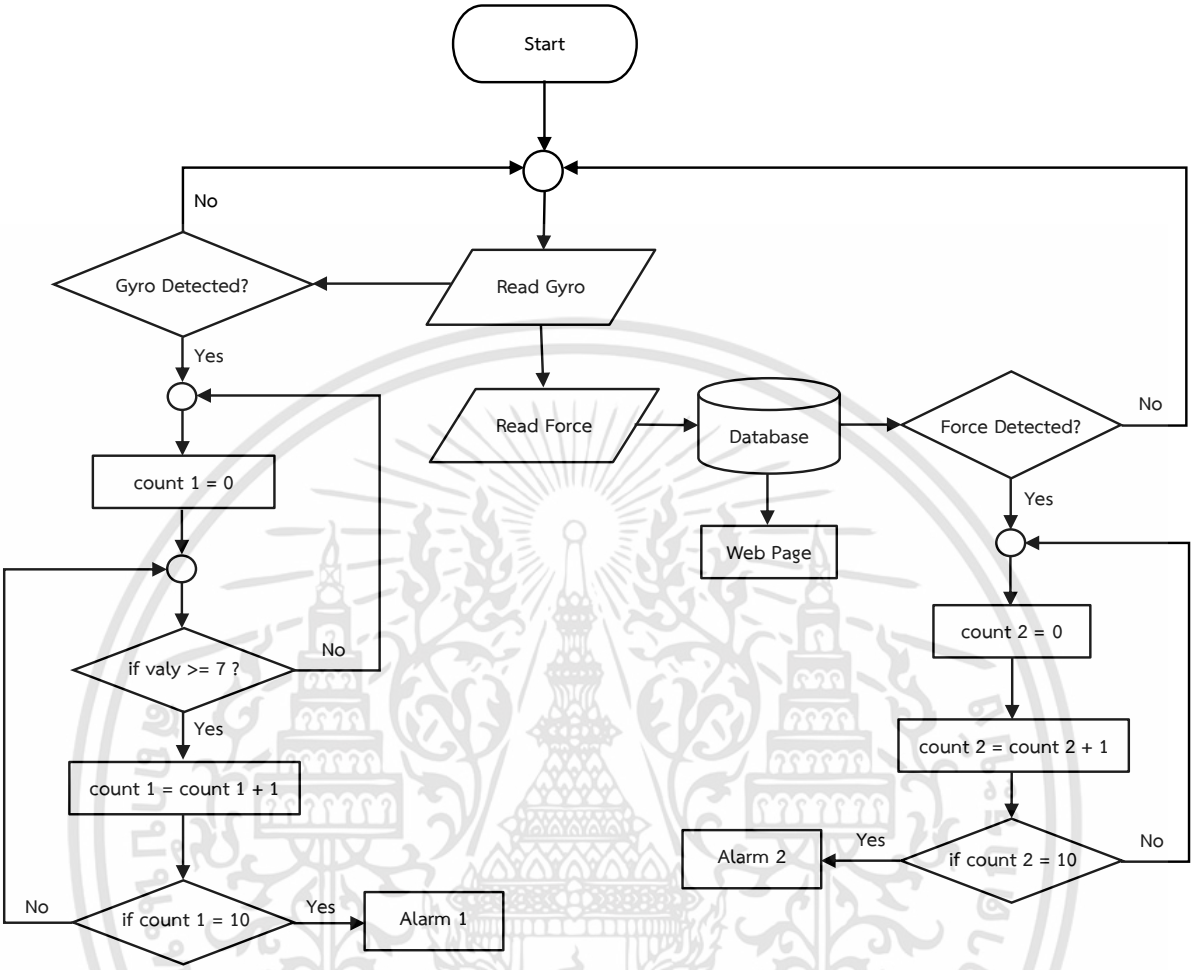
รูปที่ 3.41 ทดลองสวมใส่ชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.11 การออกแบบการทำงานของระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม

การทำงานของระบบเริ่มต้นจากการอ่านค่าของเซนเซอร์ โดยจะอ่านค่าจากโมดูลวัดความเอียง (Gyro Module) และเซนเซอร์วัดแรงกด (Force Sensitive Resistor) ซึ่งเมื่อเซนเซอร์เริ่มทำงานจะทำการส่งข้อมูลไปยังฐานข้อมูล (Database) และแสดงผลค่าต่าง ๆ ที่อ่านได้ผ่านเว็บเพจ (Web Page) และเมื่อเริ่มอ่านค่าเซนเซอร์แล้วจะสร้างเงื่อนไขว่าตรวจพบหรือไม่ ถ้าตรวจพบว่าผู้สวมใส่เริ่มนั่งหลังค่อมระบบจะทำการเก็บข้อมูลและเริ่มนับเวลา เมื่อนับถึงเวลาตามที่กำหนดไว้ระบบจะแจ้งเตือนด้วยเสียงและการสั่นของมอเตอร์ให้ผู้สวมใส่นั่งหลังตรง เมื่อนั่งหลังตรงแล้วการแจ้งเตือนด้วยเสียงและการสั่นของมอเตอร์จะหยุดทำงาน แต่หากผู้สวมใส่ยังนั่งหลังค่อมต่อไประบบการแจ้งเตือนด้วยเสียงและการสั่นของมอเตอร์จะทำงานต่อไปเรื่อย ๆ จนกว่าผู้สวมใส่จะนั่งหลังตรง ถ้าตรวจพบว่าเมื่อผู้สวมใส่นั่งระบบจะทำการเก็บข้อมูลและเริ่มนับเวลา เมื่อนับเวลาถึงระยะเวลานานเกินที่กำหนดไว้ระบบจะแจ้งเตือนด้วยเสียงและการสั่นของมอเตอร์ให้ผู้สวมใส่เปลี่ยนอิริยาบถเป็นยืนขึ้น เมื่อผู้สวมใส่ยืนขึ้นการแจ้งเตือนด้วยเสียงและการสั่นของมอเตอร์จะหยุดทำงาน แต่ถ้าผู้สวมใส่ยังคงนั่งต่อไประบบการแจ้งเตือนด้วยเสียงและการสั่นของมอเตอร์จะทำงานต่อไปเรื่อย ๆ จนกว่าผู้สวมใส่จะยืนขึ้น ถ้าระบบตรวจจับได้ทั้งสองกรณีคือ เมื่อผู้สวมใส่นั่งหลังค่อมและนั่งระบบจะทำการเก็บข้อมูลและเริ่มนับเวลา เมื่อนับถึงเวลาตามที่กำหนดไว้ระบบจะแจ้งเตือนด้วยเสียงและการสั่นของมอเตอร์ เมื่อผู้สวมใส่นั่งหลังตรงแล้วแต่ยังคงนั่งนานเกินระยะเวลาที่กำหนดไว้ระบบจะยังคงแจ้งเตือนด้วยเสียงและการสั่นของมอเตอร์อยู่จนกว่าผู้สวมใส่จะยืนขึ้นการแจ้งเตือนด้วยเสียงและการสั่นของมอเตอร์จะหยุดทำงาน ซึ่งแผนผังการทำงานของระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรมแสดงดังรูปที่ 3.42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.42 แผนผังการทำงานของระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อม
เพื่อป้องกันโรคออฟฟิศซินโดรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.12 การออกแบบเว็บเพจเพื่อแสดงผล

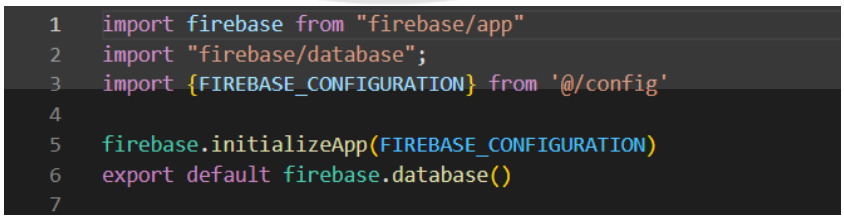
1) ทำการเปิดใช้งาน โดยที่คำสั่งแสดงดังรูปที่ 3.43

รูปที่ 3.43 คำสั่งการเปิดใช้งาน app

2) การเชื่อมต่อฐานข้อมูล Firebase กับ Node.js โดยที่คำสั่งแสดงดังรูปที่ 3.44

รูปที่ 3.44 คำสั่งการเชื่อมต่อระหว่างฐานข้อมูล Firebase กับ Node.js

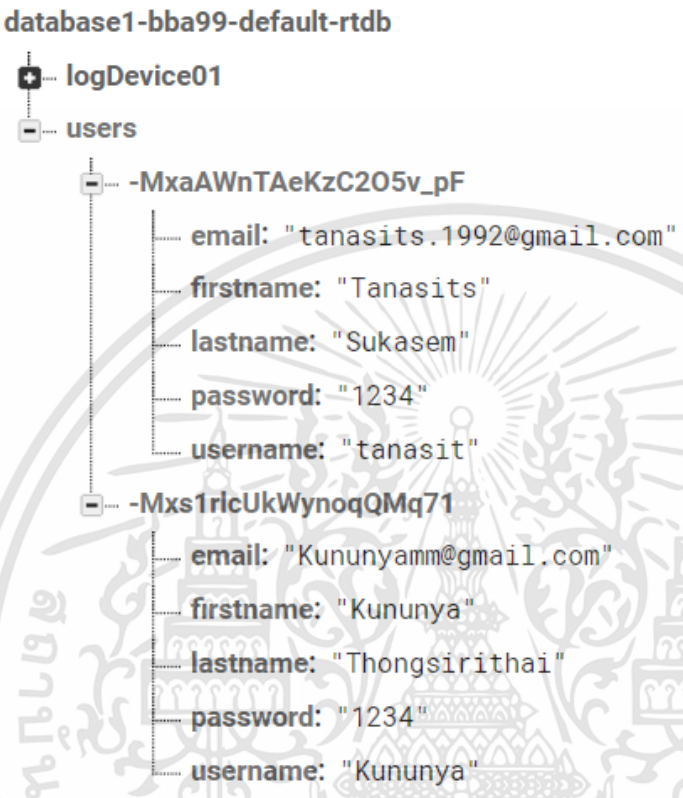
3) การเชื่อมต่อฐานข้อมูล Realtime Database ในฐานข้อมูล Firebase โดยที่คำสั่งแสดงดังรูปที่ 3.45



รูปที่ 3.45 คำสั่งการเชื่อมต่อฐานข้อมูล Realtime Database ในฐานข้อมูล Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ข้อมูลผู้ใช้งานในฐานข้อมูล Firebase แสดงดังรูปที่ 3.46



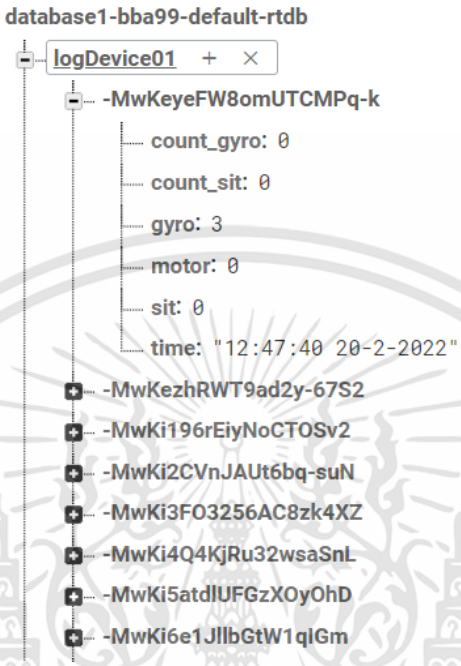
รูปที่ 3.46 ข้อมูลผู้ใช้งานในฐานข้อมูล Firebase

5) การนำข้อมูลของผู้ใช้งาน (User) ในฐานข้อมูล Firebase มาใช้งาน

การเรียกใช้ข้อมูลทั้งหมดของ User ในฐานข้อมูล Firebase โดยจะสามารถสร้างหรือเพิ่ม User ใหม่ลงฐานข้อมูล แก้ไขและลบข้อมูลของ User ได้ และถ้าส่ง ID เข้าไปดูในฐานข้อมูลสามารถรู้ข้อมูลของ ID นั้นได้ โดยจะนำข้อมูลนี้ไปใช้ในหน้า Sign in Sign up Edit และ Change password

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) ข้อมูลจาก NodeMCU ที่ส่งเข้ามาในฐานข้อมูล Firebase แสดงดังรูปที่ 3.47



รูปที่ 3.47 ข้อมูลจาก NodeMCU ที่ส่งเข้ามาในฐานข้อมูล Firebase

7) การเรียกใช้ข้อมูล Log Device ในฐานข้อมูล Firebase

จากรูปที่ 3.48 แสดงคำสั่งการนำข้อมูล Log Device ในฐานข้อมูล Firebase ทั้งหมดเข้ามา เพื่อนำไปเรียกใช้ในหน้าต่าง ๆ

```
import firebase from "../firebase";

const db = firebase.ref("/logDevice01");

class LogDeviceDataService {
    getAll() {
        return db;
    }
    off(event) {
        return db.off(event)
    }
}
export default new LogDeviceDataService();
```

รูปที่ 3.48 คำสั่งเรียกใช้ข้อมูล Log Device ในฐานข้อมูล Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) การตั้งชื่อที่อยู่ของเว็บไซต์ในหน้าต่าง ๆ ซึ่งคำสั่งแสดงดังรูปที่ 3.49

```
import { createRouter, createWebHistory } from 'vue-router'
import SignInView from '../views/SignInView.vue'
import SignUpView from '../views/SignUpView.vue'
import DetailView from '../views/DetailView.vue'
import EditView from '../views/EditView.vue'
import ChangePasswordView from '../views/ChangePasswordView.vue'

const routes = [
  {
    path: '/',
    name: 'signin',
    component: SignInView
  },
  {
    path: '/signup',
    name: 'signup',
    component: SignUpView
  },
  {
    path: '/detail',
    name: 'deatil',
    component: DetailView
  },
  {
    path: '/edit/:id',
    name: 'edit',
    component: EditView
  },
  {
    path: '/changePassword/:id',
    name: 'changePassword',
    component: ChangePasswordView
  },
]

const router = createRouter({
  history: createWebHistory(process.env.BASE_URL),
  routes
})

export default router
```

รูปที่ 3.49 คำสั่งการตั้งชื่อที่อยู่ของเว็บไซต์

9) ระบบการเข้าสู่ระบบ (Sign In)

การสร้างหน้าการเข้าสู่ระบบ ซึ่งประกอบไปด้วย Username และ Password ในการเข้าระบบ โดยจะเก็บค่า input ของ Username เป็นประเภท text และเก็บค่า input ของ Password เป็นประเภท Password เพื่อให้เมื่อเวลากรอก Password จะไม่แสดงขึ้นจะแสดงเป็น dot และในหน้าของ Sign In จะทำการเชื่อมต่อไปยังหน้าของ Sign Up สำหรับบุคคลที่ยังไม่ได้เป็นสมาชิกเพื่อให้สามารถลงทะเบียนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10) การประมวลผลเมื่อกด Submit แล้วในระบบการเข้าสู่ระบบ

- การประมวลผล เมื่อกด submit แล้วจะทำการตรวจสอบโดยสร้างเงื่อนไขเมื่อกรอกข้อมูลเข้าระบบไม่ครบ จะแสดงข้อความ “Invalid username or password” ขึ้น และหากกรอก Username และ Password แล้วกด Sign In ระบบจะทำการนำข้อมูลที่กรอกไว้ไปตรวจกับฐานข้อมูลว่าตรงกันไหมถ้าพบว่าไม่มี Username นี้ในฐานข้อมูล Firebase จะแสดงข้อความ “Invalid username or password” ขึ้น และหากมี User ในฐานข้อมูลแต่ใส่ Password ไม่ถูกต้อง ก็จะแสดงข้อความ “Invalid username or password” ขึ้นเช่นกัน แสดงดังรูปที่ 3.50 และถ้าตรวจสอบข้อมูล Username และ Password ตรงกับฐานข้อมูลจะสามารถเข้าระบบ และสามารถไปยังหน้าหลักได้

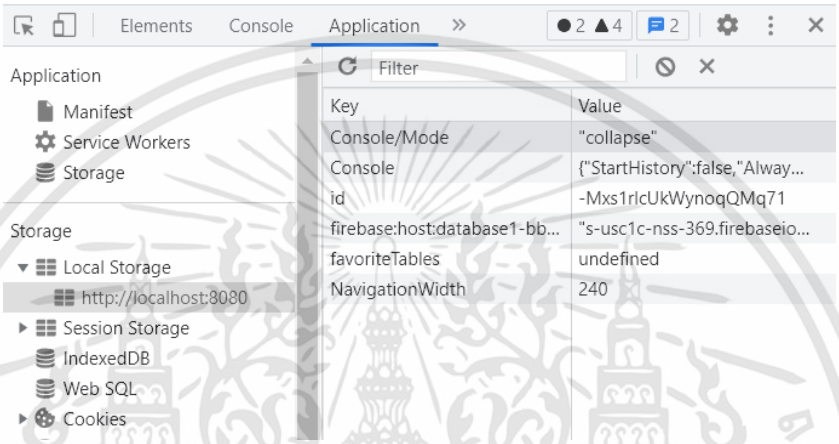
รูปที่ 3.50 หน้า Sign In เมื่อกรอกข้อมูลไม่ครบหรือกรอกข้อมูลผิด

- เมื่อกด submit ระบบจะทำการตรวจสอบว่ามี ID User นี้ใน Local Storage หรือไม่ ถ้ามี ID User นี้ จะสามารถเข้าไปยังหน้าหลักได้ หรือถ้าเข้าระบบ Guest ก็จะทำการเข้าไปยังหน้าหลักได้เลยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

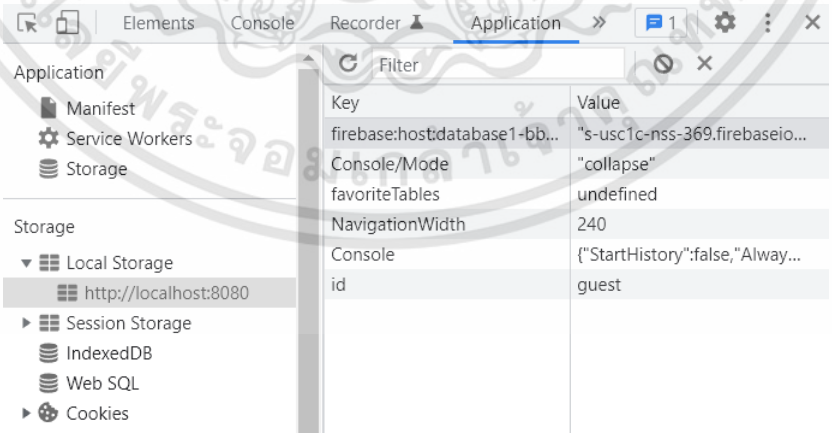
11) การเก็บค่า ID ใน Local Storage เมื่อ Sign In

- เมื่อเข้าระบบแล้ว จะทำการตรวจสอบ ID ของ user ใน Firebase แล้วนำ ID User ไปเก็บค่าใน Local Storage โดยเป็น storage ของเบราวเซอร์ เพื่อที่เมื่อรีเฟรชแล้วจะได้ไม่ต้องไปเข้าหน้า Sign In ใหม่ ดังรูปที่ 3.51



รูปที่ 3.51 การเก็บค่า ID User ใน Local Storage

- ถ้าเข้าระบบด้วย Guest ก็จะนำค่า Guest ไปเก็บค่าเป็นค่า ID ใน Local Storage แสดงดังรูปที่ 3.52 โดยเป็น Storage ของเบราวเซอร์เพื่อที่เมื่อรีเฟรชแล้วจะได้ไม่ต้องไปเข้าหน้า Sign In ใหม่



รูปที่ 3.52 การเก็บค่า Guest ใน Local Storage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12) ระบบลงทะเบียน (Sign Up)

โดยหน้าลงทะเบียนที่ได้ออกแบบไว้ จะประกอบไปด้วย Username Password First name Last name และ Email Address โดยจะเก็บค่า Username First name และ Last name เป็นประเภท Text Password เป็นประเภท Password และ Email Address เป็นประเภท Email โดยจะบังคับให้ผู้ใช้ลงทะเบียนว่าต้องกรอกข้อมูล Username Password First name และ Last name ส่วนของ Email Address ใส่หรือไม่ก็ได้

โดยในหน้าลงทะเบียนนี้ยังเชื่อมต่อการ Login with Guest ไว้เพื่อเข้าระบบโดยที่ไม่ต้องลงทะเบียน แสดงดังรูปที่ 3.53

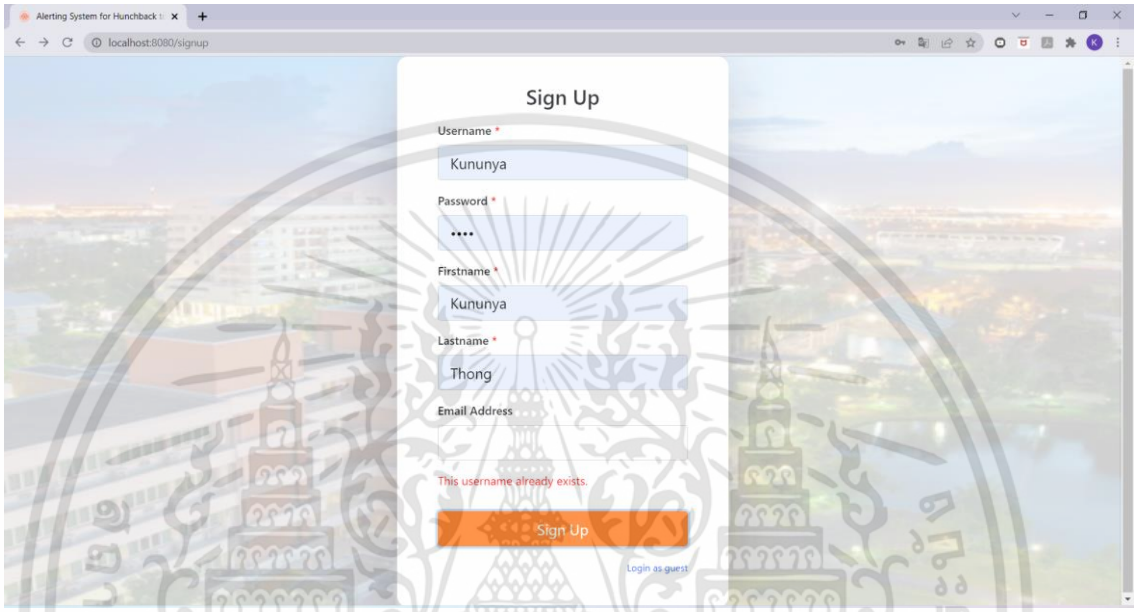
รูปที่ 3.53 หน้าต่างการลงทะเบียน

13) หน้าต่างการลงทะเบียน

เมื่อใส่ข้อมูลในหน้าลงทะเบียนครบแล้วกด submit จะทำการตรวจสอบ Username ในฐานข้อมูล Firebase โดยการสร้างเงื่อนไขว่าถ้า Username ที่ลงทะเบียนใหม่ตรงกับฐานข้อมูล Firebase จะไม่สามารถสร้าง ID User ได้ และจะแสดงผล “This username already exists” ดังรูปที่ 3.54 ถ้า Username ไม่ตรงกับ Username ในฐานข้อมูล Firebase จะทำการสร้าง User ID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นโดยการส่งค่า Username First name Last name และ Email address ไปยังฐานข้อมูล Firebase และเก็บค่า ID ใน Local Storage เมื่อสร้าง User ID แล้วจะสามารถเข้าไปยังหน้าหลักได้

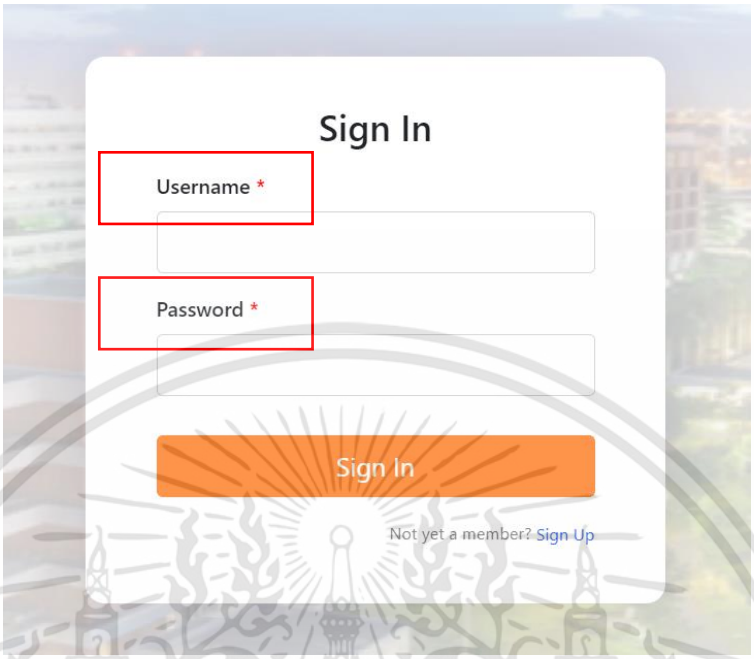


รูปที่ 3.54 เมื่อ Username ตรงกับ Username ในฐานข้อมูล

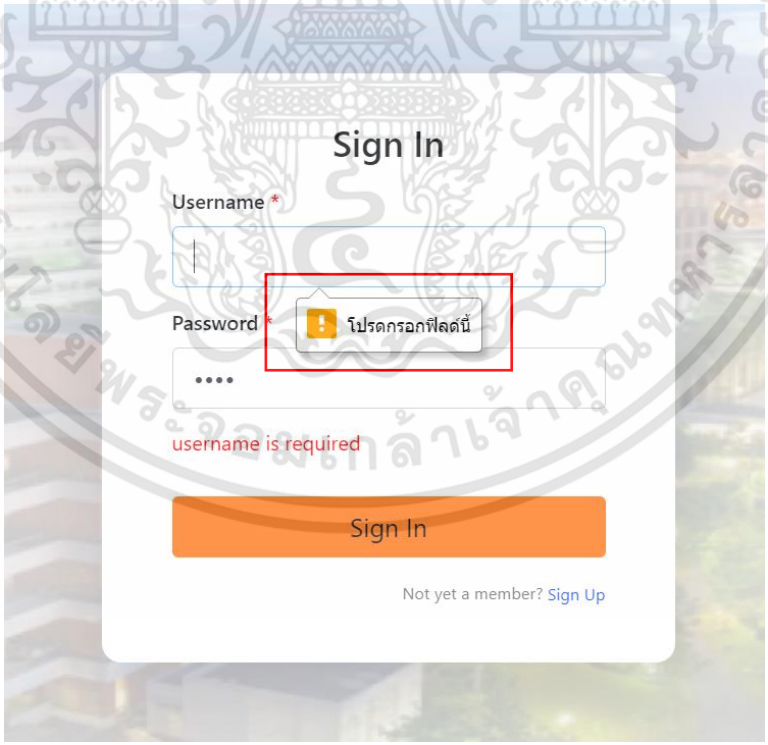
14) ส่วนของ Input form ที่เรียกใช้ในหน้าอื่น ๆ

ส่วนของ Input form ที่เรียกใช้ในหน้าอื่น ๆ โดยจะเก็บค่า input ที่ได้มาตามลำดับ จะให้แสดง label ก่อน โดยถ้าค่าที่รับมานั้นถูกกำหนดให้เป็น required จะแสดง * สีแดงขึ้น ดังรูปที่ 3.55 ในส่วน input จะกำหนดให้เป็นประเภทไหนตามที่เรากำหนด เช่น Username เก็บค่า input เป็นประเภท text และ Password เก็บค่า input เป็นประเภท Password เป็นต้น ถ้าผู้ใช้งานไม่กรอกข้อมูลช่องใดช่องหนึ่งแล้วกด submit ระบบจะแสดงข้อความขึ้นโดยทันที แสดงดังรูปที่ 3.56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.55 Label เมื่อค่าที่รับมานั้นถูกกำหนดให้เป็น required จะแสดง * สีแดงขึ้น



รูปที่ 3.56 ข้อความที่แสดงขึ้นเมื่อผู้ใช้กรอกข้อมูลไม่ครบ

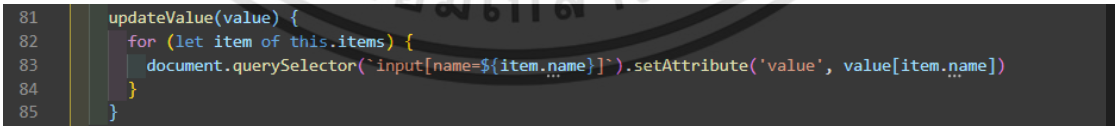
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกด submit จะประมวลผล input ที่ผู้ใช้ส่งเข้ามาในระบบ แสดงดังรูปที่ 3.57 โดยจะนำค่า input text มานำค่า value ของ input นั้นมา โดยมีเงื่อนไขว่าถ้า value มีค่าจะกำหนดค่า result เป็นค่าที่ผู้ใช้ใส่เข้ามา แต่ถ้า value นั่นไม่มีค่าแล้ว field นั่น required จะการเก็บ error ไว้ใน error message จากนั้นจะทำการตรวจสอบว่าถ้าไม่มี error message จะทำการส่งค่าออกไป

```
submit() {
  this.errorMesages = []

  for (let item of this.items) {
    let value = document.querySelector(`input[name=${item.name}]`).value

    if (![null, undefined, ""].includes(value)) {
      this.result[item.name] = value
    } else {
      if (item.required) this.errorMesages.push(`${item.name} is required`)
    }
  }
  if (this.errorMesages.length === 0) {
    this.$emit('submit', this.result)
  }
},
cancel() {
  this.$emit('cancel')
},
showErrorMessage(messages) {
  this.errorMesages = messages
},
updateValue(value) {
  for (let item of this.items) {
    document.querySelector(`input[name=${item.name}]`).setAttribute('value', value[item.name])
  }
}
```

รูปที่ 3.57 คำสั่งการประมวลผลหลังจากกด submit

การอัพเดตข้อมูลของผู้ใช้งาน ว่ามีข้อมูลอะไรบ้างมาแสดง โดยจะเรียกใช้ในหน้าการแก้ไขข้อมูล ซึ่งคำสั่งการประมวลผลแสดงดังรูปที่ 3.58



```
updateValue(value) {
  for (let item of this.items) {
    document.querySelector(`input[name=${item.name}]`).setAttribute('value', value[item.name])
  }
}
```

รูปที่ 3.58 คำสั่งการอัพเดตข้อมูลของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15) ส่วนของ Loading Icon ที่เรียกใช้ในหน้าอื่น ๆ

- Loading Icon เมื่อรับข้อมูลมาแล้วเป็น true จะแสดงสัญลักษณ์ spinner border ขึ้น เพื่อให้ผู้ใช้รอการประมวลผลในหน้านั้น ๆ ซึ่งคำสั่งการใช้งานแสดงดังรูปที่ 3.59

```
<template>
  <div v-show="showLoading" class="loading">
    <span class="spinner-border"></span>
  </div>
</template>

<script>

export default{
  name: 'LoadingIcon',
  props: {
    showLoading: {
      type: Boolean,
      required: true
    },
  }
}
</script>
```

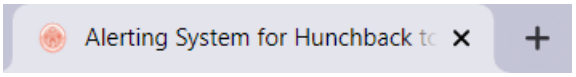
รูปที่ 3.59 คำสั่งการเรียกใช้ Loading Icon

16) การออกแบบหน้าหลักที่ใช้แสดงผล

- การออกแบบโลโก้ และชื่อ Title ของ Website Title ใน HTML ดังรูปที่ 3.60 และ 3.61 ตามลำดับ

```
<!-- TODO: logo and title-->
<link rel="icon" href="<%= BASE_URL %>logo.png">
<title>Alerting System for Hunchback to Prevent Office Syndrome</title>
```

รูปที่ 3.60 ออกแบบโลโก้ และชื่อ Title ของ Website Title



รูปที่ 3.61 Website Title

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

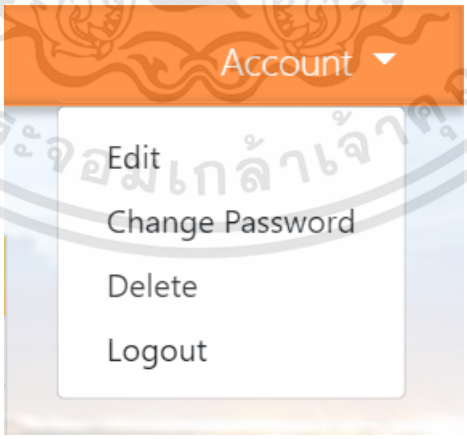
17) ส่วนของ Header

จากรูปที่ 3.62 แสดงการออกแบบส่วนของ Header โดยให้ Header ชื่อ “Alerting System for Hunchback to Prevent Office Syndrome” และสร้างเมนู Drop down ชื่อ Account แสดงดังรูปที่ 3.63 โดยหาก login ใช้งานโดยใช้ User ID เมนูที่เชื่อมไปยังหน้าอื่น ๆ ได้แก่ Edit Change Password Delete และ Log out แสดงดังรูปที่ 3.64 แต่หากผู้ใช้ ใช้งานโดยใช้ Guest เมนูที่เชื่อมไปยังหน้าอื่น ๆ ได้แก่ Sign Up และ Logout แสดงดังรูปที่ 3.65

```
<template>
  <div class="nav-header">
    <h3 class="text-left text-header">Alerting System for Hunchback to Prevent Office Syndrome</h3>
    <button class="btn dropdown-toggle btn-manage-user" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
      Account
    </button>
    <div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
      <a v-if="!guest" class="dropdown-item" @click="editUser">Edit</a>
      <a v-if="!guest" class="dropdown-item" @click="changePassword">Change Password</a>
      <a v-if="!guest" class="dropdown-item" @click="deleteUser">Delete</a>
      <a v-if="guest" class="dropdown-item" @click="signup">Sign Up</a>
      <a class="dropdown-item" @click="logout">Logout</a>
    </div>
  </div>
```

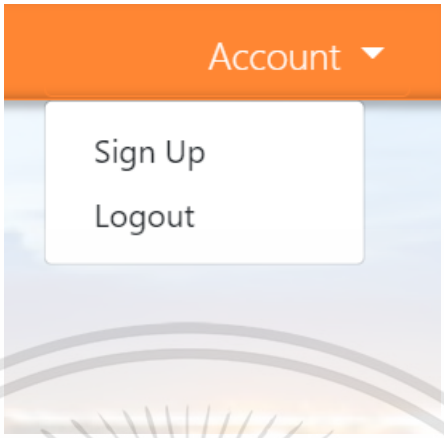
รูปที่ 3.62 การออกแบบส่วนของ Header

รูปที่ 3.63 Head title ของ Web Page



รูปที่ 3.64 เมนู Drop down เมื่อเข้าใช้ด้วย User Id

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.65 เมนู Drop down เมื่อเข้าใช้ด้วย Guest

18) ส่วนของการแสดงผล และการแบ่งหน้าแสดงผล Pagination

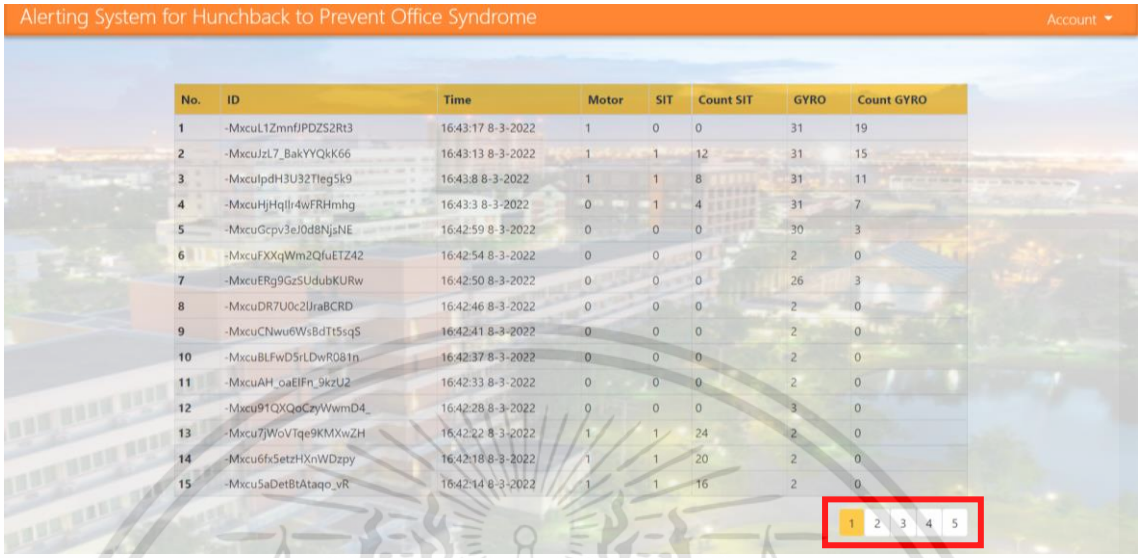
- ทำการสร้างตารางโดยสร้างตาราง header เป็นคอลัมน์แสดง ค่า No. ค่า ID ค่า Time ค่า Moter ค่า Sit ค่า Count Sit ค่า Gyro และ ค่า Count gyro ในส่วนของตาราง body จะนำข้อมูล Log Device ในฐานข้อมูล Firebase มาแสดง

- การเรียกใช้ข้อมูล Log Device จากฐานข้อมูลมาใช้ โดยจะไม่ปิดการใช้งานหากยังอยู่ในหน้าการแสดงผล เพื่อที่ข้อมูลจะอัพเดตตลอดเวลาแล้วแสดงผลแบบเรียลไทม์ และจะปิดการเรียกใช้ข้อมูล Log Device ในฐานข้อมูล เมื่อออกจากหน้านี้ไป

- ในฐานข้อมูล ในตอนแรกข้อมูลเวลาไม่ได้อยู่ในรูปแบบของ 00:00:00 และวันที่ 00-00-0000 หากนำค่าในฐานข้อมูลไปใช้เลยจะทำให้เกิดข้อผิดพลาดขึ้นได้ใน JavaScript จึงต้องทำข้อมูลเวลาให้อยู่ในรูปแบบ 00:00:00 และวันที่ 00-00-0000 เสียก่อน แล้วจึงจะนำข้อมูลเวลามาเปรียบเทียบ โดยที่กำหนดให้เวลาล่าสุดขึ้นก่อน

- การเปลี่ยนหน้าการแสดงผล Pagination เมื่อได้จำนวนข้อมูลมาทั้งหมด จะทำการแบ่งหน้าโดยการกำหนดให้ข้อมูลแต่ละหน้ามี 15 loops จากนั้นจะนำจำนวนข้อมูลทั้งหมดมาหารด้วย 15 แล้วปัดเศษขึ้นจะทำให้เราได้จำนวนหน้าดังรูปที่ 3.66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



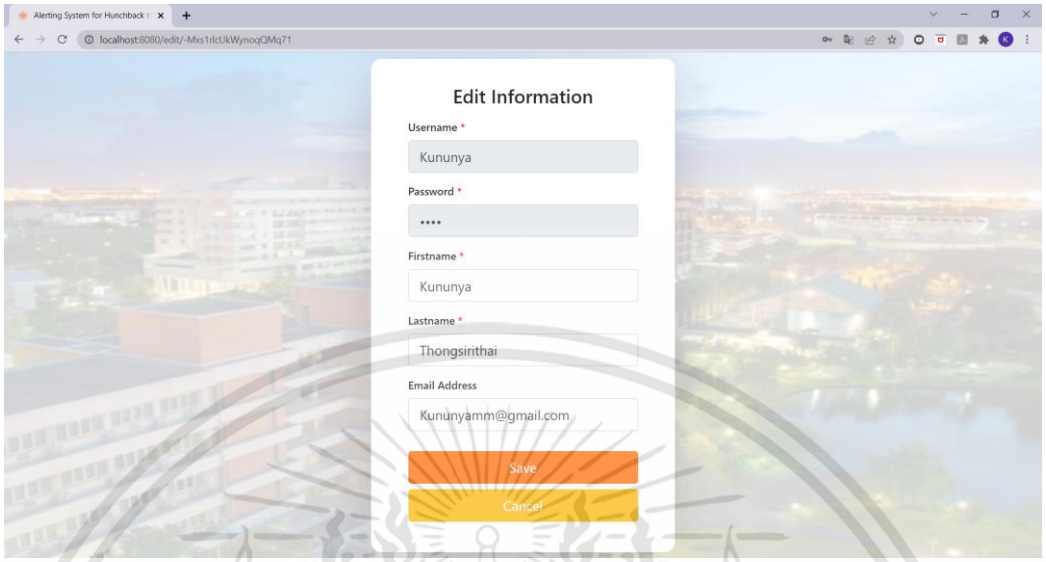
รูปที่ 3.66 ส่วนการแสดงผล และการแบ่งหน้าการแสดงผล

19) ส่วนของเมนู Drop down

- ส่วนของการแก้ไขข้อมูล (Edit Information)

การออกแบบของหน้าการแก้ไขข้อมูล ซึ่งส่วนของการแก้ไขข้อมูลนั้นได้ออกแบบเหมือนกันกับหน้าลงทะเบียน (Sign Up) เมื่อผู้ใช้กดแก้ไข (Edit) ในเมนู Drop down จะเข้ามายังหน้าการแก้ไขข้อมูล (Edit Information) ระบบจะทำการดึงข้อมูลเดิมมาแสดงให้เห็น และการแก้ไขข้อมูลกำหนดให้แก้ไขข้อมูลได้แค่ First name Last name และ Email address เท่านั้น ถ้าผู้ใช้แก้ไขข้อมูลแล้วกดบันทึก (Save) ระบบจะทำการแก้ไขข้อมูลในฐานข้อมูลด้วยและจะกลับไปยังหน้าแสดงผล ถ้าผู้ใช้กดยกเลิก (Cancel) ข้อมูลในฐานข้อมูลจะไม่มีการเปลี่ยนแปลง และจะกลับไปยังหน้าแสดงผล โดยที่ส่วนการแสดงผลของหน้าการแก้ไขข้อมูลแสดงได้ดังรูปที่ 3.67

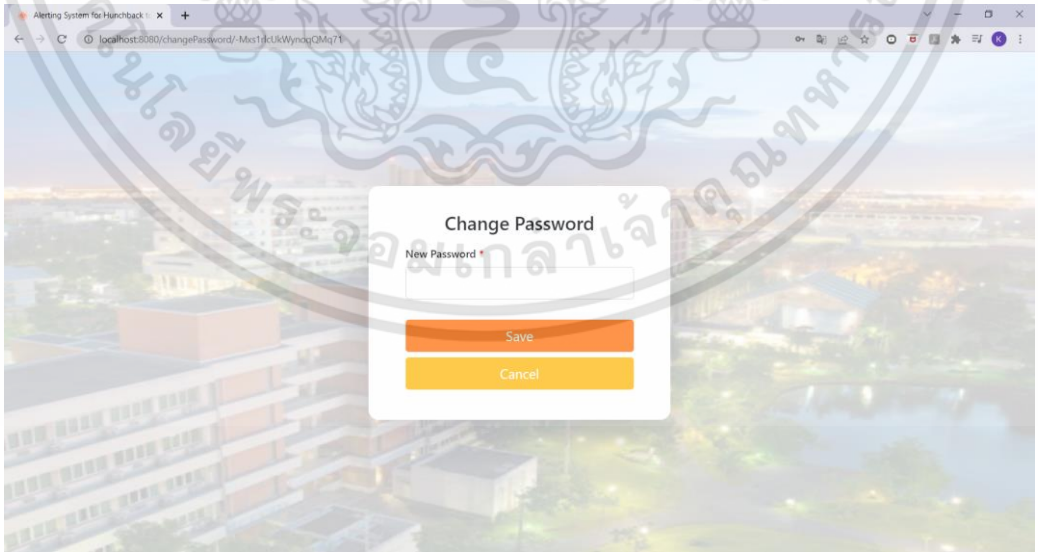
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.67 ส่วนของหน้าการแก้ไขข้อมูล (Edit Information)

- ส่วนของการเปลี่ยนรหัสผ่าน (Change Password)

การออกแบบหน้าการเปลี่ยนรหัสผ่าน โดยจะตั้งชื่อ Label ว่า New Password รับค่ามาในประเภทของ Password และตั้งชื่อของปุ่ม submit ว่า ‘Save’ และปุ่มการยกเลิกว่า ‘Cancel’ แสดงดังรูปที่ 3.68



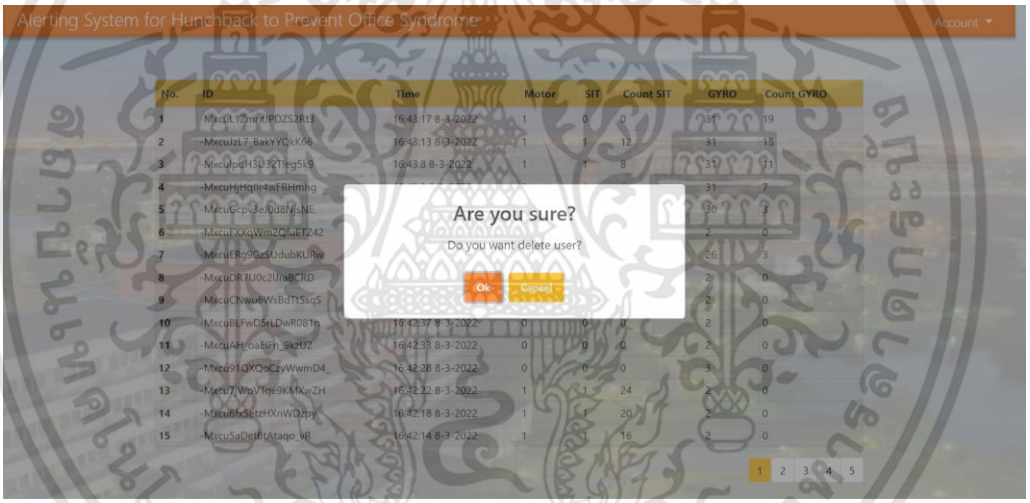
รูปที่ 3.68 หน้าการเปลี่ยนรหัสผ่าน (Change Password)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้ทำการเปลี่ยนรหัสผ่านใหม่แล้วกด Save ระบบจะทำการส่งข้อมูลไปยังฐานข้อมูลของผู้ใช้งาน แล้วทำการเปลี่ยนรหัสผ่านของผู้ใช้งาน แล้วจะกลับไปยังหน้าแสดงผล แต่ถ้าผู้ใช้กด Cancel ระบบจะกลับมายังหน้าแสดงผล

- ส่วนของการลบข้อมูลผู้ใช้งาน (Delete User)

เมื่อผู้ใช้งานกดลบ (Delete) ที่เมนู Drop down หน้า modal จะขึ้นมา แล้วแสดงข้อความขึ้นว่า “Do you want delete user?” ขึ้นแสดงในรูปที่ 3.69 ถ้าผู้ใช้เลือก ‘Ok’ ระบบจะลบข้อมูล ID ของผู้ใช้ใน Local storage และลบข้อมูลทั้งหมดของผู้ใช้ในฐานข้อมูลออก แล้วจะไปยังหน้า Sign In แต่ถ้าผู้ใช้เลือก ‘Cancel’ จะยังคงอยู่หน้าแสดงผลอยู่



รูปที่ 3.69 หน้าการลบข้อมูลผู้ใช้งาน (Delete User)

- ส่วนของการลงทะเบียนเมื่อใช้ ID Guest

เมื่อ Guest กด Sign In ในเมนู Drop down ระบบจะทำการนำ ID Guest ออกจาก Local Storage แล้วไปยังหน้า Sign Up โดยที่คำสั่งส่วนของการลงทะเบียนเมื่อใช้ ID Guest แสดงดังรูปที่ 3.70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
223    signup() {
224      localStorage.removeItem("id")
225      this.$router.push('/signup')
226    }
```

รูปที่ 3.70 คำสั่งส่วนของการลงทะเบียนเมื่อใช้ ID Guest

- ส่วนของการออกจากระบบ

เมื่อผู้ใช้กด Log out ในเมนู Drop down ระบบจะทำการนำ ID ออกจาก Local Storage แล้วไปยังหน้า Sign In โดยที่คำสั่งส่วนของการออกจากระบบ แสดงดังรูปที่ 3.71

```
219    logout() {
220      localStorage.removeItem("id")
221      this.$router.push('/')
222    },
```

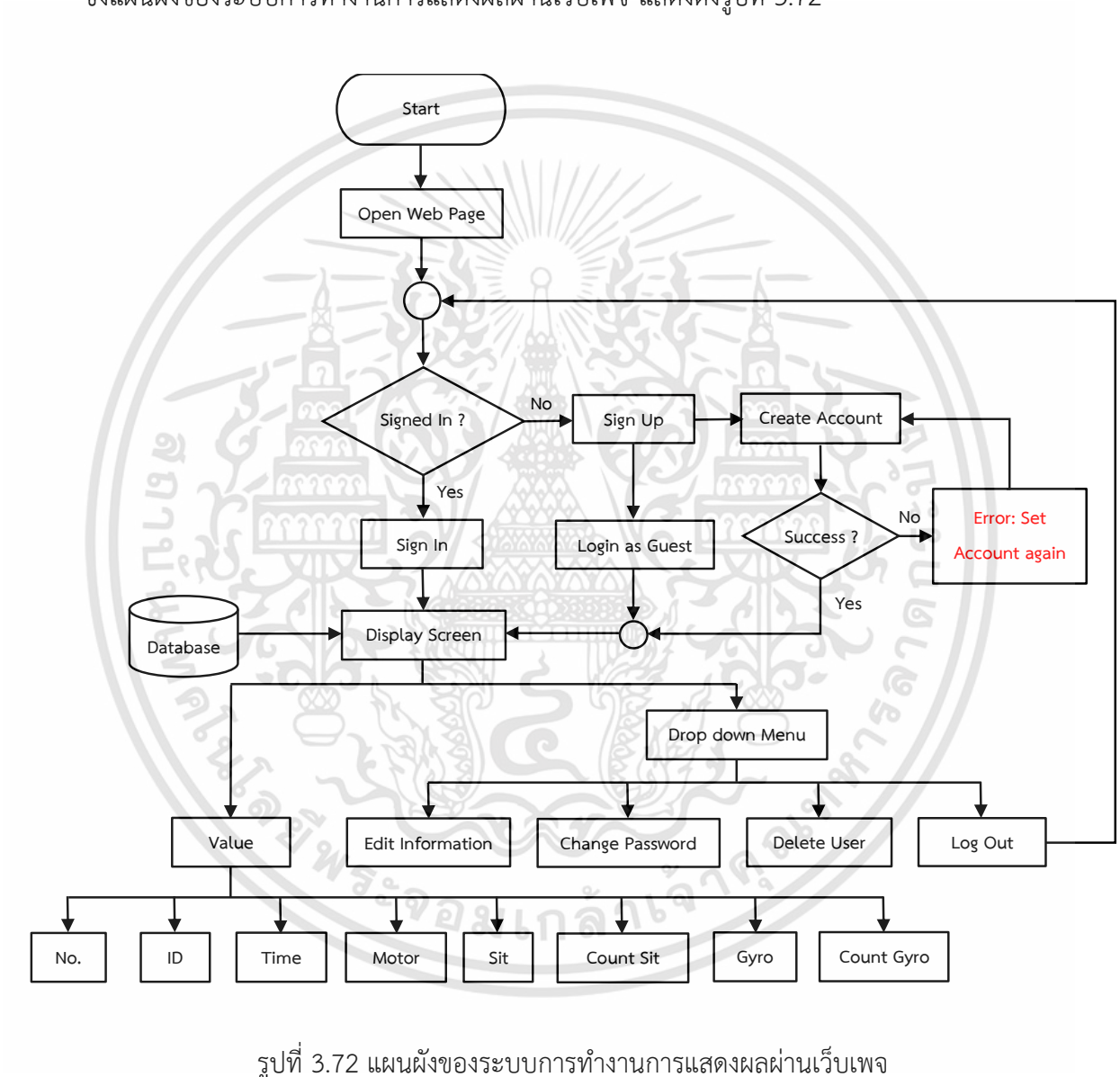
รูปที่ 3.71 คำสั่งส่วนของการออกจากระบบ

3.1.13 การออกแบบระบบการทำงานการแสดงผลผ่านเว็บเพจ

การทำงานการแจ้งเตือนผ่านเว็บเพจโดยจะเริ่มจากผู้ใช้งานเข้าสู่เว็บเพจจะแสดงหน้าเข้าสู่ระบบ (Sign In) เมื่อกรอกข้อมูลครบ และกด Sign In ระบบจะตรวจสอบว่ากรอกข้อมูลครบหรือไม่ ถ้าข้อมูลครบ และถูกต้อง จะเข้าสู่หน้าแสดงผล แต่หากไม่ครบ และผู้ใช้ไม่มีข้อมูลอยู่ในระบบจะต้องไปที่หน้าลงทะเบียน (Sign Up) เมื่อกรอกข้อมูลแล้วกด Sign Up ระบบจะทำการตรวจสอบว่ากรอกข้อมูลครบหรือไม่ ถ้ากรอกไม่ครบจะให้กรอกใหม่ แต่หากกรอกครบจะทำการตรวจสอบว่ามี Username ซ้ำกันหรือไม่ ถ้าซ้ำให้ตั้งใหม่อีกครั้ง ถ้าไม่ซ้ำจะบันทึกข้อมูลของผู้ใช้แล้วจะทำการเข้าสู่หน้าแสดงผล หรือถ้าหาก login as guest ก็จะส่งไปยังหน้าแสดงผล เมื่อเข้าสู่หน้าแสดงผลระบบจะนำข้อมูลในฐานข้อมูลออกมาแสดงผลเป็นตาราง โดยที่ข้อมูลที่แสดงจะแสดงข้อมูลล่าสุดไว้ข้างบนสุด และในแต่ละหน้าจะแสดงผลทั้งหมด 15 ข้อมูล ซึ่งประกอบไปด้วย ค่า No. ค่า ID ค่า Time ค่า Motor ค่า Sit ค่า Count Sit ค่า Gyro และ ค่า Count gyro และอีกส่วนของหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงผล คือ เมนู Drop down ซึ่งประกอบไปด้วย ส่วนของการแก้ไขข้อมูล (Edit Information) ส่วนของการเปลี่ยนรหัสผ่าน (Change Password) ส่วนของการลบข้อมูลผู้ใช้งาน (Delete User) ส่วนของการลงทะเบียนเมื่อใช้ ID Guest และส่วนของการออกจากระบบ ซึ่งแผนผังของระบบการทำงานการแสดงผลผ่านเว็บเพจ แสดงดังรูปที่ 3.72



รูปที่ 3.72 แผนผังของระบบการทำงานการแสดงผลผ่านเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เครื่องมือที่ใช้ในการทดลอง

ในปริญญานิพนธ์นี้ มีอุปกรณ์และเครื่องมือที่ใช้ในการทดลอง ดังนี้

3.2.1 ไมโครคอนโทรลเลอร์

สำหรับไมโครคอนโทรลเลอร์ที่เลือกใช้ในปริญญานิพนธ์นี้ คือ Arduino Nano แสดงดังรูปที่ 2.1 ออกแบบมาให้มีขนาดเล็ก และใช้กับงานทั่ว ๆ ไป ใช้ชิปไอซีไมโครคอนโทรลเลอร์เบอร์ ATmega168 หรือเบอร์ ATmega328 โปรแกรมผ่านโปรโตคอล UART มีชิป USB to UART มาให้ใช้ Mini USB เชื่อมต่อกับคอมพิวเตอร์

3.2.2 โมดูลวัดความเอียง

สำหรับโมดูลวัดความเอียงที่ผู้จัดทำเลือกใช้งาน คือ GY-521 MPU6050 3 Axis Gyroscope Accelerometer Sensor Module แสดงดังรูปที่ 2.5 สามารถวัดความเอียงได้แบบ 3 ทิศทาง ความละเอียดในการวัด 16 บิต ใช้การเชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่าน I2C โมดูลนี้ได้รวมเอาเซนเซอร์ Gyroscope และ Accelerometer เข้าไว้ด้วยกัน

3.2.3 เซนเซอร์วัดแรงกด

ปริญญานิพนธ์นี้เลือกใช้เซนเซอร์วัดแรงกดดังรูปที่ 2.6 ขนาด 0.5 นิ้ว รุ่น FSR402 หรือ Force Sensitive Resistor (FSR402) เป็นเซนเซอร์วัดแรงกดในช่วง 0.2 N-20 N ค่าที่วัดได้จะเป็นแบบ Analog 0-5 V_{DC} (0-1023) ใช้ไฟเลี้ยง 5VDC FSR402 จึงเหมาะสำหรับงานที่ต้องการเซนเซอร์วัดแรงกดได้หลายระดับ

3.2.4 ลำโพง Buzzer

ปริญญานิพนธ์นี้เลือกใช้ active buzzer เป็นตัวสร้างเสียง 2,300 Hz สามารถสร้างเสียงเตือนได้โดยการจ่ายไฟ 3.3 – 5 V_{DC} ดังรูปที่ 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 โมดูลรีเลย์

ปริญญานิพนธ์นี้เลือกใช้โมดูลรีเลย์ 5 V - Active Low ใช้งานในการควบคุมอุปกรณ์ไฟฟ้า รับกระแสได้สูงถึง 10 A 250 V_{AC} และ 10 A 30 V_{DC} มี LED แสดงสถานะการทำงานของรีเลย์ ดังรูปที่ 2.8

3.2.6 โมดูลไวไฟ NodeMCU-ESP32S

ปริญญานิพนธ์นี้เลือกใช้ โมดูลไวไฟ NodeMCU-ESP32S ในการเชื่อมต่อระหว่าง Arduino Nano เข้าสู่เซิร์ฟเวอร์เพื่อจัดเก็บข้อมูลต่าง ๆ ลงในฐานข้อมูล ดังรูปที่ 2.13

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทดสอบอุปกรณ์โมดูลวัดความเอียง (Gyro Module)

3.3.1.1 การทดสอบอุปกรณ์โมดูลวัดความเอียง (Gyro Module) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

3.3.2 การทดสอบเซนเซอร์วัดแรงกด (Force Sensitive Resistor - FSR402)

3.3.2.1 การทดสอบเซนเซอร์วัดแรงกด (Force Sensitive Resistor - FSR402) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

3.3.3 การทดสอบอุปกรณ์โมดูลเสียงเตือน (Active Buzzer)

3.3.3.1 การทดสอบอุปกรณ์โมดูลเสียงเตือน (Active Buzzer) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

3.3.4 การทดสอบต่ออุปกรณ์โมดูลวัดความเอียง (Gyro Module) เซนเซอร์วัดแรงกด และโมดูลเสียงเตือน (Active Buzzer) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

3.3.4.1 ทำการทดสอบต่ออุปกรณ์โมดูลวัดความเอียง (Gyro Module) เซนเซอร์วัดแรงกด และโมดูลเสียงเตือน (Active Buzzer) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 ทำการทดสอบเขียนโปรแกรมระบบการแจ้งเตือนผู้นั่งหลังค่อม และผู้ที่นั่งเกินเวลาที่กำหนด

3.3.6 ทำการทดสอบชิ้นงาน และระบบทั้งหมด

3.3.6.1 ทำการทดสอบเมื่อผู้สวมใส่มีแนวโน้มในการนั่งหลังค่อม

3.3.6.2 ทำการทดสอบเมื่อผู้สวมใส่นั่งเป็นระยะเวลานานเกินที่กำหนด

3.3.6.3 ทำการทดสอบเมื่อผู้สวมใส่มีแนวโน้มในการนั่งหลังค่อม และนั่งเป็นระยะเวลานานเกินที่กำหนด

3.3.6.4 การทดสอบ Gyro Module เมื่อตรวจพบว่าผู้สวมใส่นั่งหลังค่อมและนั่งหลังตรง

3.3.7 ทำการทดลองจัดเก็บข้อมูลลงในฐานข้อมูล Firebase

3.3.7.1 ทำการทดสอบการใช้งาน NodeMCU-32S 38 pins

3.3.7.2 ทำการเชื่อมต่อ Arduino Nano กับ NodeMCU-32S

3.3.7.3 ทำการทดลองเขียนโปรแกรมเชื่อมต่อรับส่งข้อมูลระหว่าง Arduino Nano และ NodeMCU-32S

3.3.7.4 ทำการเชื่อมต่อ NodeMCU-32S กับ Firebase เพื่อจัดเก็บข้อมูลลงในฐานข้อมูล

3.3.7.5 ทำการส่งข้อมูล และจัดเก็บข้อมูลไว้ในฐานข้อมูล

3.3.7.6 ทำการเก็บข้อมูลผู้ใช้งาน (Users) และ NodeMCU

3.3.8 ทำการสร้างหน้าเว็บเพจเพื่อแสดงผลการแจ้งเตือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

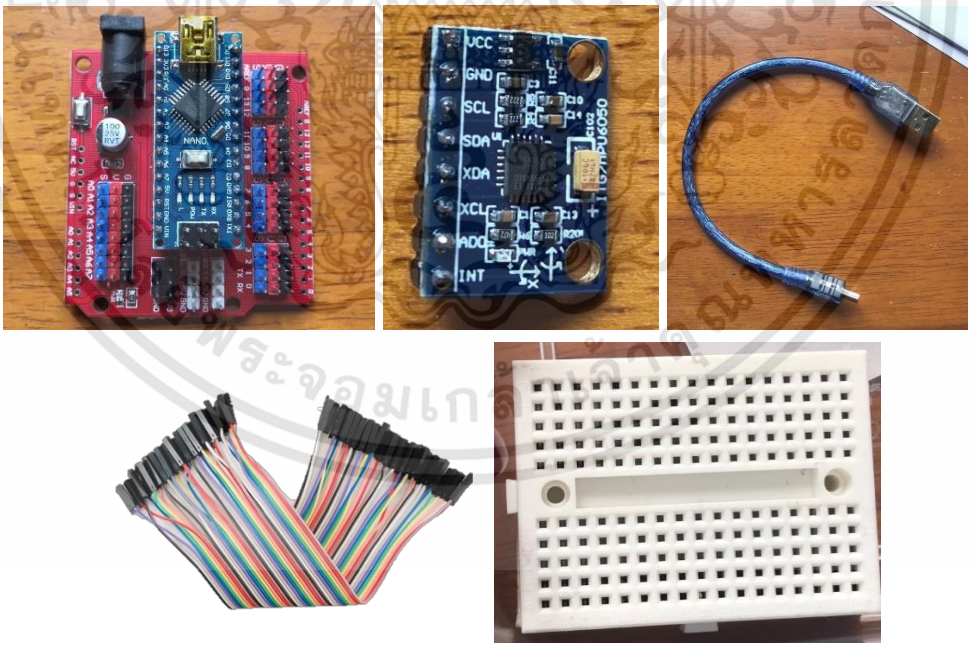
ผลการทดลอง

ผู้จัดทำได้ทำการเก็บผลการทำงานของระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม โดยแบ่งการทดลองเป็นส่วน ๆ ดังต่อไปนี้

4.1 การทดสอบอุปกรณ์โมดูลวัดความเอียง (Gyro Module)

4.1.1 การทดสอบอุปกรณ์โมดูลวัดความเอียง (Gyro Module) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

4.1.1.1 ขั้นตอนต่อไปเป็นการต่อโมดูลต่าง ๆ ที่จะใช้ในการทดลองการทำงานของโมดูล โดยอุปกรณ์ที่ใช้มีดังรูปที่ 4.1



รูปที่ 4.1 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ โมดูลวัดความเอียง สาย Mini USB สายจัมพ์ และโพรโทบอร์ดทดลอง ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

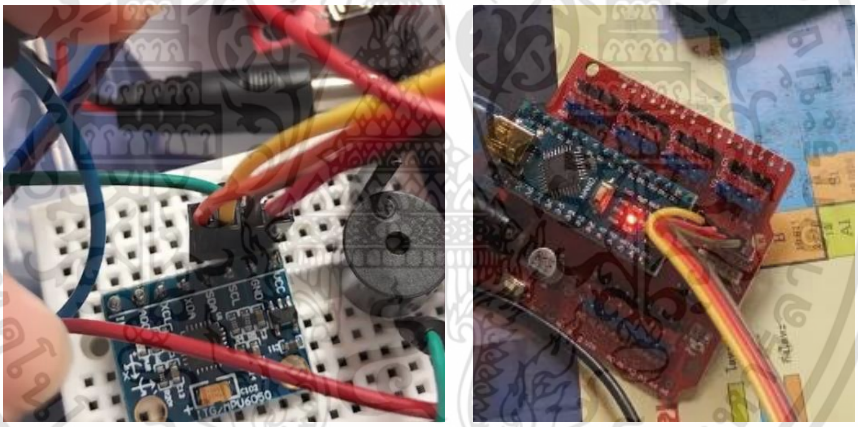
- โมดูลวัดความเอียง (Gyro Module)

- สาย Mini USB

- สายจัมพ์

- โพรโทบอร์ดทดลอง

4.1.1.2 ทำการต่อสาย USB จากบอร์ดไมโครคอนโทรลเลอร์ Arduino Nano เข้าสู่คอมพิวเตอร์ และทำการเชื่อมต่อขาของโมดูลวัดความเอียง สู่บอร์ดไมโครคอนโทรลเลอร์ โดยที่ขาเชื่อมต่อตามตารางที่ 2.3 ซึ่งออกมาเป็นดังรูปที่ 4.2



รูปที่ 4.2 การต่อขาของอุปกรณ์โมดูลวัดความเอียง และบอร์ดไมโครคอนโทรลเลอร์ ตามลำดับ

4.1.1.3 เขียนโปรแกรมให้ไมโครคอนโทรลเลอร์

- การเขียนโปรแกรมนั้นจะใช้ไลบรารี่อ้างอิงของ I2CdeV.h [26] ที่ได้มาจาก Github และไลบรารี่ MPU6050.h [26] แสดงดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

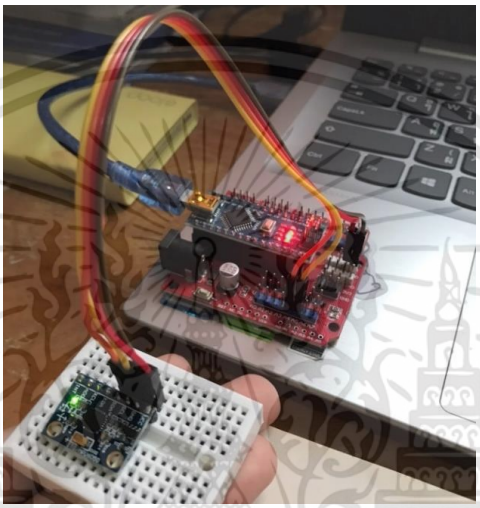
```
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

MPU6050 mpu;

int16_t ax, ay, az;
int16_t gx, gy, gz;
int valx , valy , valz;
char rd;

int prevVal;
int led = 13 ;
int pin11 = 11 , pin10 = 10 ;
int val1 , val2 ;
int valgy1 = 0 , valgy2 = 0;

void setup() {

pinMode(led,OUTPUT) ;
Wire.begin();
Serial.begin(9600);
Serial.println("Initialize MPU");
mpu.initialize();
Serial.println(mpu.testConnection() ? "Connected" : "Connection failed");

}

void loop()

{

mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
valx = map(ax, -17000, 17000, 0, 179);
valy = map(ay, -17000, 17000, 0, 179);
valz = map(az, -17000, 17000, 0, 179);

Serial.print("axis x = ") ;
Serial.print(valx) ;
Serial.print(" axis y = ") ;
Serial.print(valy) ;
Serial.print(" axis z = ") ;
Serial.println(valz) ;
delay(1000);
}
```

รูปที่ 4.3 โปรแกรมทดลองการทำงานของโมดูลวัดความเอียง (Gyro Module)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

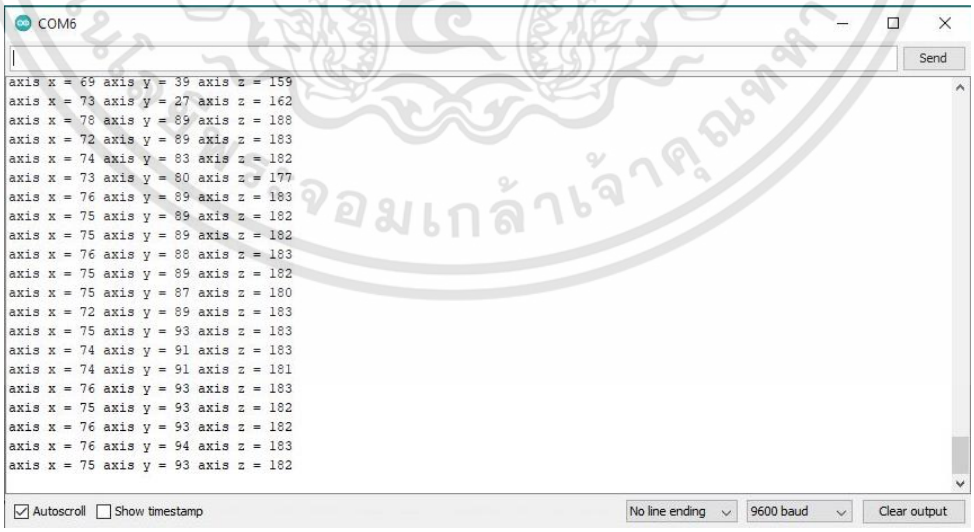
4.1.1.4 ทำการทดลองโปรแกรมที่เขียนไว้

- เมื่อทำการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ จากนั้นนำสาย Mini USB เสียบต่อเข้ากับไมโครคอนโทรลเลอร์ และคอมพิวเตอร์ แล้วทำการอัพโหลด และรันโปรแกรม โมดูลจะแสดงผลดังรูปที่ 4.4



รูปที่ 4.4 การแสดงผลของโมดูลเมื่อทำการอัพโหลด และรันโปรแกรม

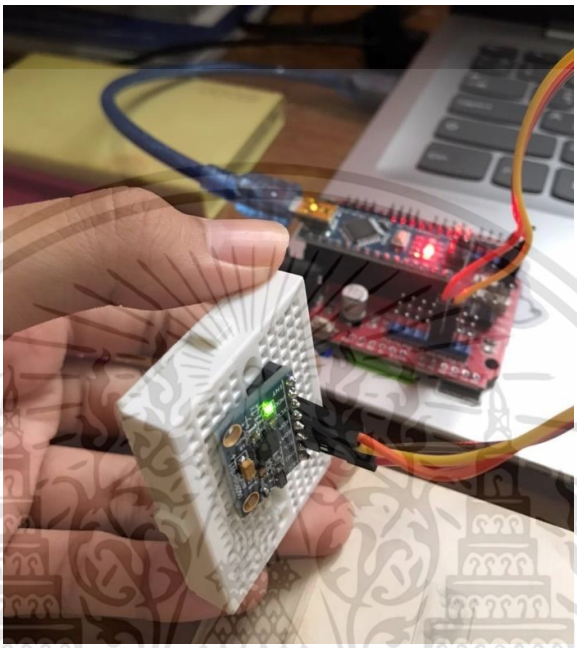
- เมื่อทำการอัพโหลด และรันโปรแกรมสำเร็จโมดูลจะแสดงผลดังรูปที่ 4.4 จากนั้นเรียกดูค่าแกน x y และ z จาก Serial Monitor แสดงดังรูปที่ 4.5



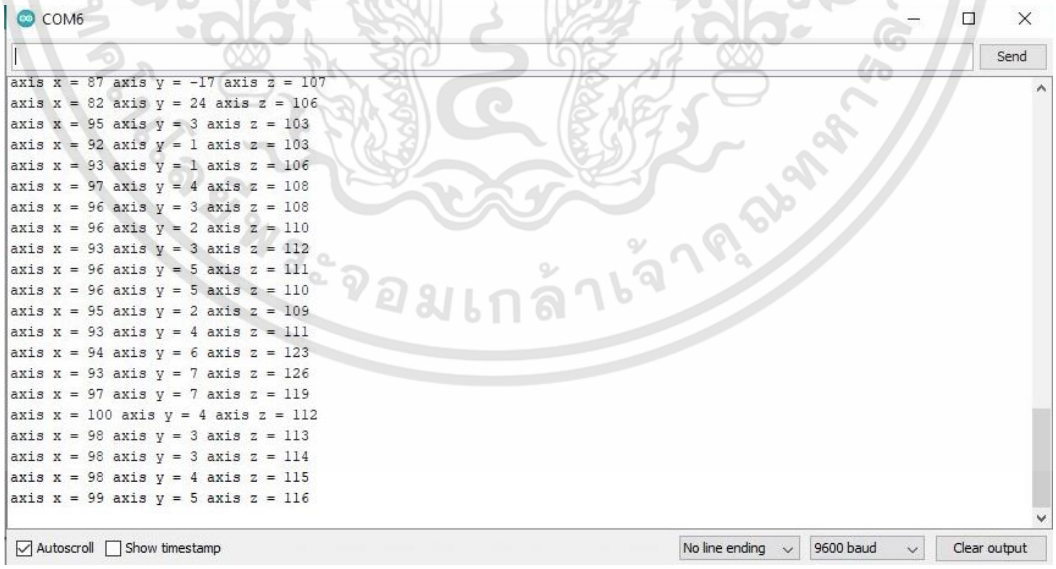
รูปที่ 4.5 การแสดงผลค่าแกน x y และ z จาก Serial Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการทดลองอีกครั้งโดยทำการเอนโมดูลวัดความเอียง ดังรูปที่ 4.6 และเรียกดูค่าแกน x y และ z จาก Serial Monitor แสดงดังรูปที่ 4.7



รูปที่ 4.6 ทำการทดลองอีกครั้งโดยทำการเอนโมดูลวัดความเอียง



รูปที่ 4.7 การแสดงผลค่าแกน x y และ z จาก Serial Monitor

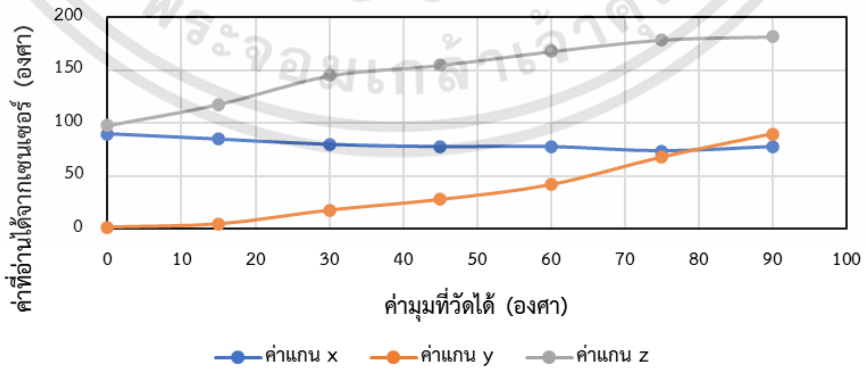
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากการทดสอบสามารถแสดงได้ดังตารางที่ 4.1

ตารางที่ 4.1 ค่ามุมที่วัดได้ ค่าแกน x, y และ z ของ Gyro Module และค่าความผิดพลาดของค่ามุมที่วัดได้เทียบกับค่าแกน y จากเซนเซอร์

ค่ามุมที่วัดได้ (องศา)	ค่าแกน x (องศา)	ค่าแกน y (องศา)	ค่าแกน z (องศา)	ค่าความผิดพลาดของค่ามุมที่วัดได้เทียบกับค่าแกน y จากเซนเซอร์
0	90	2	98	1
15	85	5	118	2
30	80	18	145	0.67
45	78	28	155	0.61
60	78	42	168	0.43
75	74	68	179	0.10
90	78	90	182	0

จากตารางที่ 4.1 แสดงค่ามุมที่วัดได้ และค่าแกน x, y และ z ของ Gyro Module โดยทดสอบการเอนโมดูลแล้วบันทึกผลหาค่าความผิดพลาดของค่ามุมที่วัดได้เทียบกับค่าแกน y จากเซนเซอร์ (อ้างอิงค่ามุมที่วัดได้ โดยเริ่มต้นที่ 0 องศา) สามารถนำมาพล็อตเป็นกราฟแสดงความสัมพันธ์ระหว่างค่าที่อ่านได้จากเซนเซอร์และค่ามุมที่วัดได้ แสดงดังรูปที่ 4.8



รูปที่ 4.8 กราฟแสดงความสัมพันธ์ระหว่างค่าที่อ่านได้จากเซนเซอร์และค่ามุมที่วัดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดสอบเซนเซอร์วัดแรงกด (Force Sensitive Resistor - FSR402)

4.2.1 การทดสอบเซนเซอร์วัดแรงกด (Force Sensitive Resistor - FSR402) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

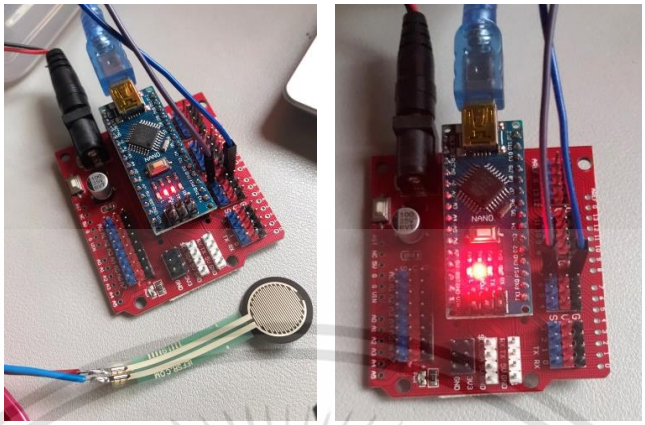
4.2.1.1 ขั้นตอนต่อไปเป็นการต่อโมดูลต่าง ๆ ที่จะใช้ในการทดลองการทำงานของเซนเซอร์ โดยอุปกรณ์ที่ใช้มีดังรูปที่ 4.9

รูปที่ 4.9 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ เซนเซอร์วัดแรงกด และสาย Mini USB ตามลำดับ

- บอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)
- เซนเซอร์วัดแรงกด (Force Sensitive Resistor - FSR402)
- สาย Mini USB

4.2.1.2 ทำการต่อสาย USB จากบอร์ดไมโครคอนโทรลเลอร์ Arduino Nano เข้าสู่คอมพิวเตอร์ และทำการเชื่อมต่อเซนเซอร์วัดแรงกด สู่บอร์ดไมโครคอนโทรลเลอร์ ซึ่งออกมาเป็นดังรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 การต่อเซนเซอร์วัดแรงกด และบอร์ดไมโครคอนโทรลเลอร์ ตามลำดับ

4.2.1.3 เขียนโปรแกรมให้ไมโครคอนโทรลเลอร์

- การเขียนโปรแกรมทดลองการทำงานของเซนเซอร์วัดแรงกดกับไมโครคอน-โทรลเลอร์ แสดงดังรูปที่ 4.11

```
int fsrPin = 0;     // the FSR and 10K pulldown are connected to a0
int fsrReading;     // the analog reading from the FSR resistor divider

void setup(void) {
  // We'll send debugging information via the Serial monitor
  Serial.begin(9600);
}

void loop(void) {
  fsrReading = analogRead(fsrPin);

  Serial.print("Analog reading = ");
  Serial.print(fsrReading);     // the raw analog reading

  // We'll have a few threshholds, qualitatively determined
  if (fsrReading < 10) {
    Serial.println(" - No pressure");
  } else if (fsrReading < 200) {
    Serial.println(" - Light touch");
  } else if (fsrReading < 500) {
    Serial.println(" - Light squeeze");
  } else if (fsrReading < 800) {
    Serial.println(" - Medium squeeze");
  } else {
    Serial.println(" - Big squeeze");
  }
  delay(1000);
}
```

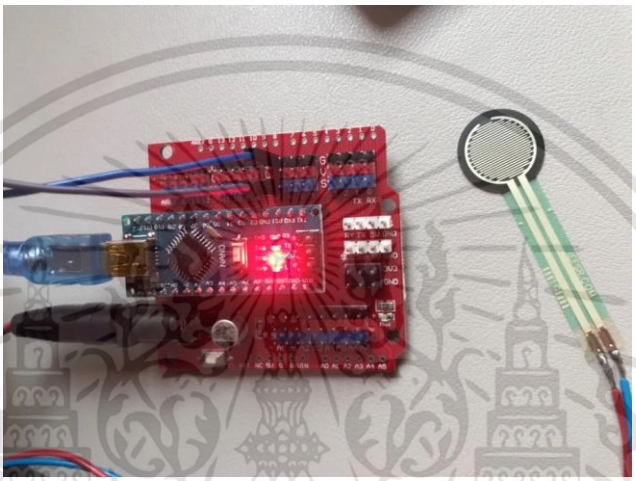
รูปที่ 4.11 โปรแกรมทดลองการทำงานของเซนเซอร์วัดแรงกด

(Force Sensitive Resistor - FSR402)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

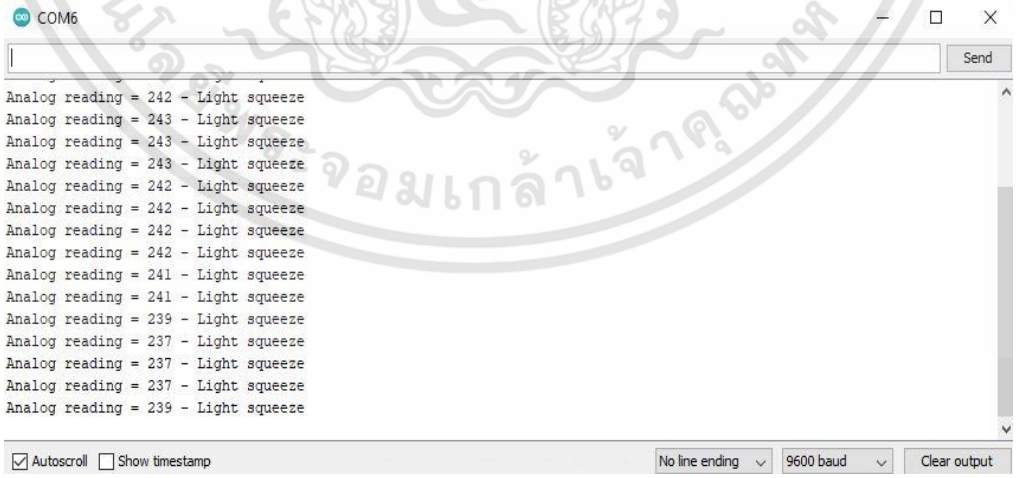
4.2.1.4 ทำการทดลองโปรแกรมที่เขียนไว้

- เมื่อทำการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ จากนั้นนำสาย Mini USB เสียบต่อเข้ากับไมโครคอนโทรลเลอร์ และคอมพิวเตอร์ แล้วทำการอัพโหลด และรันโปรแกรม จะแสดงผลดังรูปที่ 4.12



รูปที่ 4.12 การแสดงผลของเซนเซอร์เมื่อทำการอัพโหลด และรันโปรแกรม

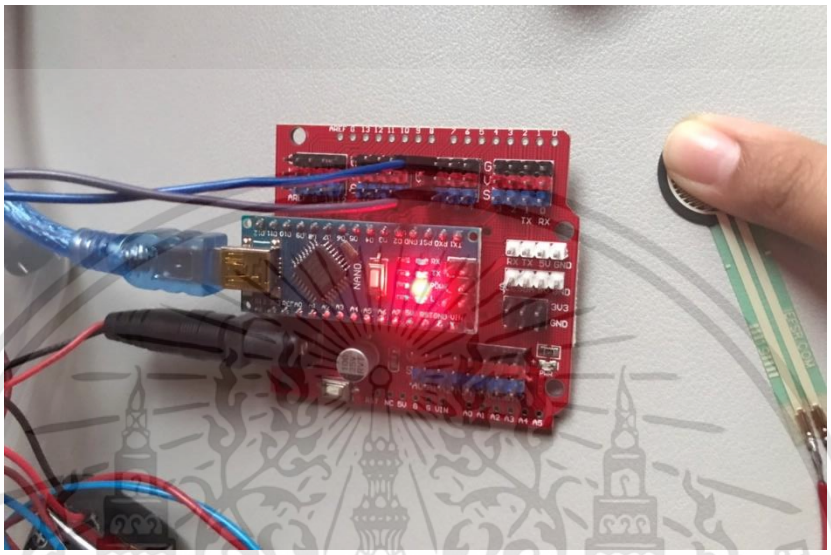
- เมื่อทำการอัพโหลด และรันโปรแกรมสำเร็จจะแสดงผลดังรูปที่ 4.12 จากนั้นเรียกดูค่าวัดแรงกดจาก Serial Monitor แสดงดังรูปที่ 4.13



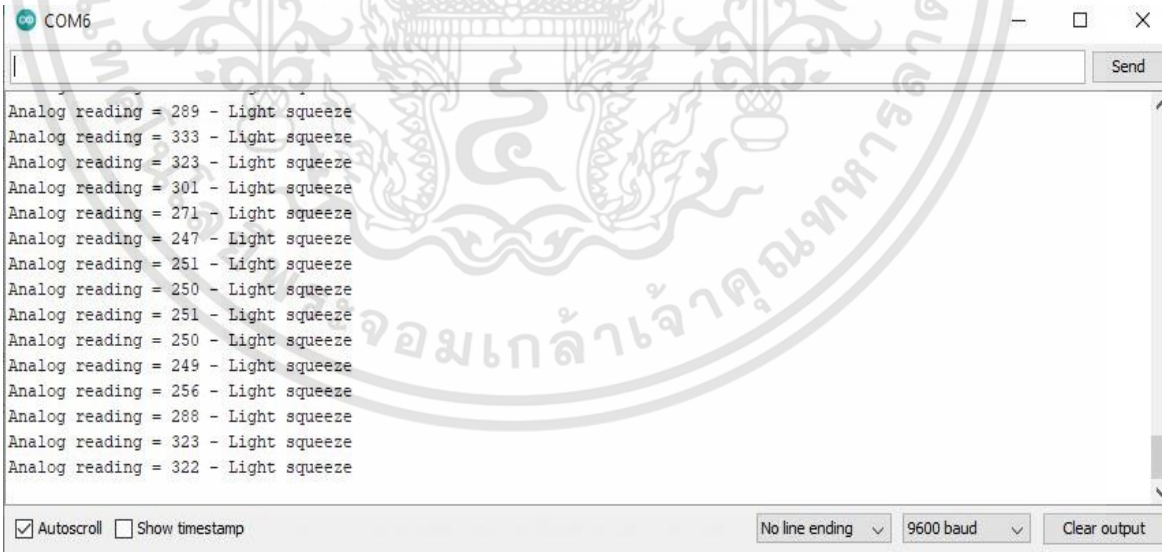
รูปที่ 4.13 การแสดงผลค่าวัดแรงกดจาก Serial Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการทดลองอีกครั้งโดยทำการกดที่เซนเซอร์ ดังรูปที่ 4.14 และเรียกดูค่าวัดแรงกดจาก Serial Monitor แสดงดังรูปที่ 4.15



รูปที่ 4.14 ทำการทดลองอีกครั้งโดยทำการกดที่เซนเซอร์



รูปที่ 4.15 การแสดงผลค่าวัดแรงกดจาก Serial Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

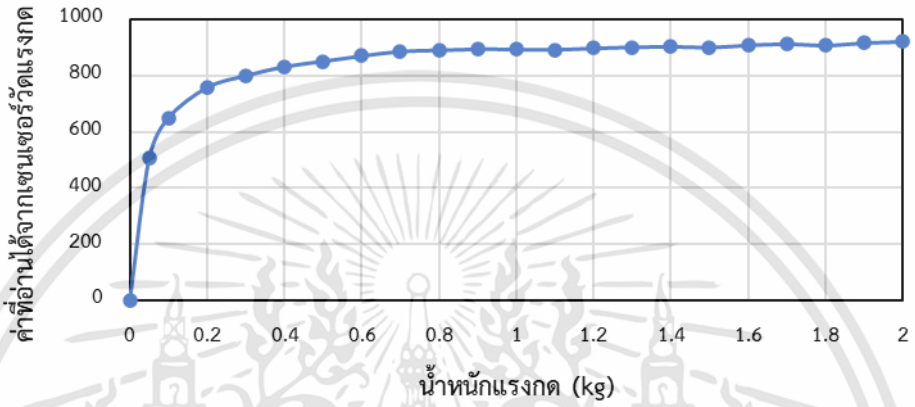
- จากการทดสอบสามารถแสดงได้ดังตารางที่ 4.2

ตารางที่ 4.2 ค่าน้ำหนักแรงกด และค่าที่อ่านได้จากเซนเซอร์วัดแรงกด

น้ำหนักแรงกด (kg)	ค่าที่อ่านได้จากเซนเซอร์วัดแรงกด
0	0
0.05	505
0.1	650
0.2	758
0.3	800
0.4	832
0.5	851
0.6	870
0.7	886
0.8	890
0.9	894
1	893
1.1	892
1.2	897
1.3	901
1.4	903
1.5	900
1.6	908
1.7	912
1.8	907
1.9	916
2	920

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.2 แสดงค่าน้ำหนัก และค่าที่อ่านได้จากเซนเซอร์วัดแรงกดโดยทดสอบแล้วบันทึกผล สามารถนำมาพล็อตเป็นกราฟแสดงความสัมพันธ์ระหว่างค่าที่อ่านได้จากเซนเซอร์วัดแรงกด และน้ำหนักแรงกด แสดงดังรูปที่ 4.16



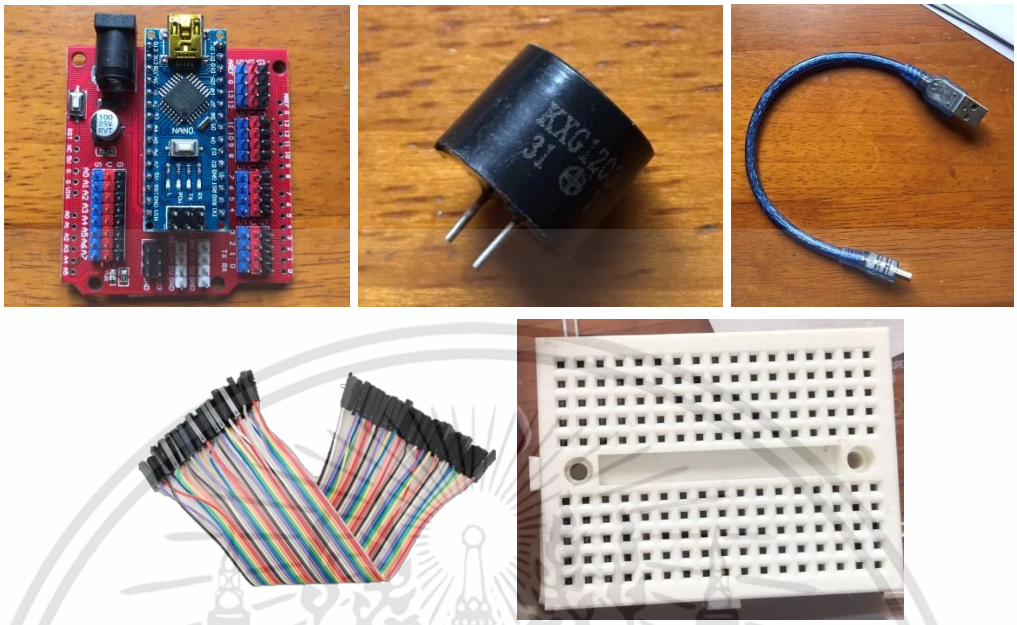
รูปที่ 4.16 กราฟแสดงความสัมพันธ์ระหว่างค่าที่อ่านได้จากเซนเซอร์วัดแรงกด และน้ำหนักแรงกด

4.3 การทดสอบอุปกรณ์โมดูลเสียงเตือน (Active Buzzer)

4.3.1 การทดสอบอุปกรณ์โมดูลเสียงเตือน (Active Buzzer) กับบอร์ดไมโครคอน-โทรลเลอร์ (Arduino Nano)

4.3.1.1 ขั้นตอนต่อไปเป็นการต่อโมดูลต่าง ๆ ที่จะใช้ในการทดลองการทำงานของโมดูล โดยอุปกรณ์ที่ใช้มีดังรูปที่ 4.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

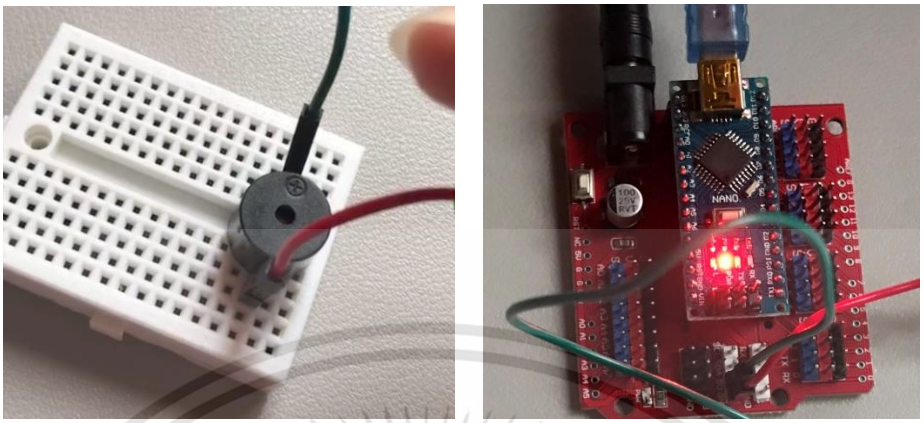


รูปที่ 4.17 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ โมดูลเสียงเตือน สาย Mini USB สายจัมพ์ และโพรโทบอร์ดทดลอง ตามลำดับ

- บอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)
- โมดูลเสียงเตือน (Active Buzzer)
- สาย Mini USB
- สายจัมพ์
- โพรโทบอร์ดทดลอง

4.3.1.2 ทำการต่อสาย USB จากบอร์ดไมโครคอนโทรลเลอร์ Arduino Nano เข้าสู่คอมพิวเตอร์ และทำการเชื่อมต่อโมดูลเสียงเตือนกับโพรโทบอร์ด จากนั้นทำการเชื่อมต่อเข้าสู่บอร์ดไมโครคอนโทรลเลอร์ ซึ่งออกมาเป็นดังรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 การต่อขาของอุปกรณ์โมดูลเสียงเตือน และบอร์ดไมโครคอนโทรลเลอร์ ตามลำดับ

4.3.1.3 เขียนโปรแกรมให้ไมโครคอนโทรลเลอร์

การเขียนโปรแกรมทดลองการทำงานของโมดูลเสียงเตือนกับไมโครคอนโทรลเลอร์ แสดงดังรูปที่ 4.19

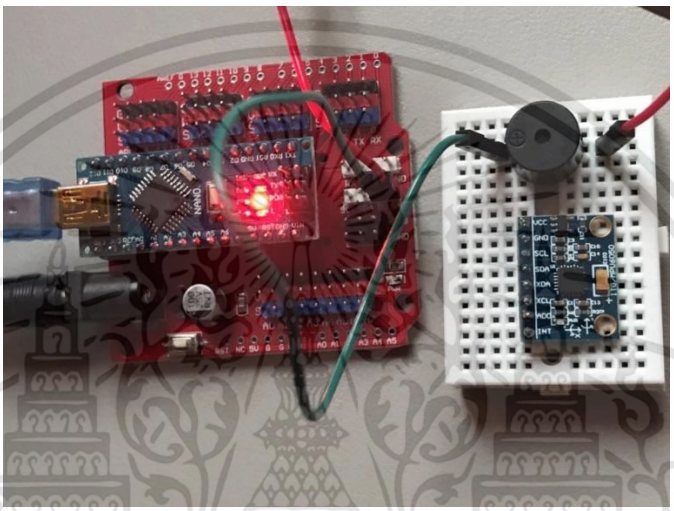
```
int buzzer = 13;

void setup() {
pinMode(buzzer, OUTPUT); }
void loop() {
digitalWrite(buzzer , HIGH); // สร้างเสียงเตือน
delay(1000); // ร้องค้างไว้ 1 วินาที
digitalWrite(buzzer , LOW); //หยุดร้อง
delay(1000); // หยุดร้อง 1 วินาที
}
```

รูปที่ 4.19 โปรแกรมทดลองการทำงานของโมดูลเสียงเตือน (Active Buzzer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.1.4 ทำการทดลองโปรแกรมที่เขียนไว้

- เมื่อทำการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ จากนั้นนำสาย Mini USB เสียบต่อเข้ากับไมโครคอนโทรลเลอร์ และคอมพิวเตอร์ แล้วทำการอัพโหลด และรันโปรแกรม โมดูลจะแสดงผลดังรูปที่ 4.20



รูปที่ 4.20 การแสดงผลของโมดูลเมื่อทำการอัพโหลด และรันโปรแกรม

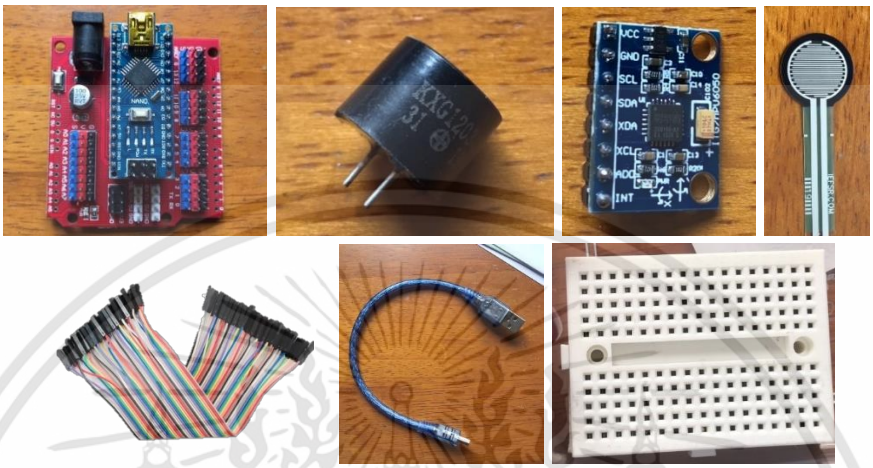
- เมื่อทำการอัพโหลด และรันโปรแกรมสำเร็จโมดูลจะแสดงผลดังรูปที่ 4.20 และจะมีเสียงแจ้งเตือนจากตัวโมดูลนาน 1 วินาทีตามที่ได้เขียนในโปรแกรมไว้ ซึ่งสามารถปรับเปลี่ยนเวลาเสียงแจ้งเตือนได้ตามต้องการ

4.4 การทดสอบต่ออุปกรณ์โมดูลวัดความเอียง (Gyro Module) เซนเซอร์วัดแรงกด และโมดูลเสียงเตือน (Active Buzzer) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

4.4.1 ทำการทดสอบต่ออุปกรณ์โมดูลวัดความเอียง (Gyro Module) เซนเซอร์วัดแรงกด และโมดูลเสียงเตือน (Active Buzzer) กับบอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.1.1 ขั้นตอนต่อไปเป็นการต่อโมดูลต่าง ๆ ที่จะใช้ในการทำงานของระบบ โดยอุปกรณ์ที่ใช้มีดังรูปที่ 4.21



รูปที่ 4.21 อุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ โมดูลเสียงเตือน โมดูลวัดความเอียง เซนเซอร์วัดแรงกดสายจัมพ์ สายMini USB และโพรโทบอร์ดทดลอง ตามลำดับ

- บอร์ดไมโครคอนโทรลเลอร์ (Arduino Nano)
- โมดูลวัดความเอียง (Gyro Module)
- โมดูลเสียงเตือน (Active Buzzer)
- เซนเซอร์วัดแรงกด (Force Sensitive Resistor - FSR402)
- สาย Mini USB
- สายจัมพ์
- โพรโทบอร์ดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ทำการทดสอบเขียนโปรแกรมระบบการแจ้งเตือนผู้นั่งหลังค่อม และผู้ที่นั่งเกินเวลาที่กำหนด

- ทำการทดลองเขียนโปรแกรมโดยรวมของโมดูลวัดความเอียง (Gyro Module) เซนเซอร์วัดแรงกด และโมดูลเสียงเตือน (Active Buzzer) แสดงดังรูปที่ 4.22

```
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#include "Wire.h"

MPU6050 mpu;
int16_t ax, ay, az;
int16_t gx, gy, gz;
int valx , valy , valz;
char rd;
int prevVal;


int val1 , val2 ;
int valgy1 = 0 , valgy2 = 0;


#define sit 7
#define buzzer 8
#define motor 9

int alarm1 = 0;
int alarm2 = 0;
int count1 = 0;
int count2 = 0;
void setup() {
  Serial.begin(9600);
  pinMode(sit, INPUT);

  Serial.begin(9600);
  Serial.println("Initialize MPU");
  mpu.initialize();
  Serial.println(mpu.testConnection() ? "Connected" : "Connection failed");

  pinMode(motor, OUTPUT);
  pinMode(buzzer, OUTPUT);
  digitalWrite(motor, HIGH);
  digitalWrite(buzzer, LOW);
}

void loop() {


  Serial.println( digitalRead(sit));

  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  valx = map(ax, -17000, 17000, 0, 179);
  valy = map(ay, -17000, 17000, 0, 179);
  valz = map(az, -17000, 17000, 0, 179);
  Serial.print("axis x = ") ;
  Serial.print(valx) ;
  Serial.print(" axis y = ") ;
  Serial.print(valy) ;
  Serial.print(" axis z = ") ;
  Serial.println(valz) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if ( valy >=  7) {
    Serial.println("          y > !!! ");
    count1 += 1;
    Serial.println("                count 1 = " + String(count1));
    if ( count1 >= 10 )
    {
      alarm1 = 1;
    }
  }
  if (valy < 7 ) {
    Serial.println("         < y");
    count1 = 0;
    alarm1 = 0;
  }
  /*
    if (valx < 50) {
     Serial.println("         <  x ");
    }
    if (valz > 150) {
     Serial.println("         z > ");
    }
    //-------------------------------------------------------------
  */
  if (  digitalRead(sit) == LOW )
  {
    count2 += 1;
    Serial.println("                count 2 = " + String(count2));
    if ( count2 >= 10 )
    {
      alarm2 = 1;
    }
  }
  else
  {
    count2 = 0;
    alarm2 = 0;
  }
  //-------------------------------------------------------------------------
  if ( alarm1 == 1 | alarm2 == 1 )
  {
    Serial.println("                !!!!!!!!!!!!! ");
    digitalWrite(buzzer, HIGH);
    digitalWrite(motor, LOW);
  }
  else
  {
    Serial.println("                ..............  ");
    digitalWrite(buzzer, LOW);
    digitalWrite(motor, HIGH);
  }

  //-------------------------------------------------------------------------
  delay(1000);
}
```

รูปที่ 4.22 โปรแกรมทดลองการทำงานโดยรวมของโมดูลวัดความเอียง (Gyro Module)

เซนเซอร์วัดแรงกด และโมดูลเสียงเตือน (Active Buzzer)

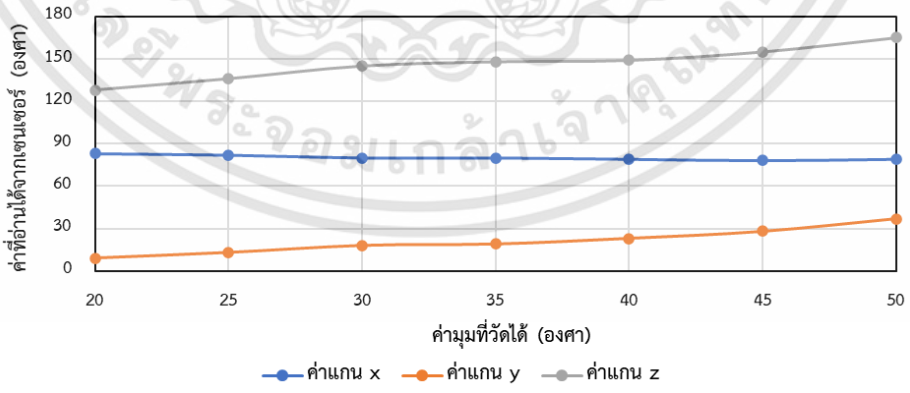
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากการทดสอบสามารถแสดงได้ดังตารางที่ 4.3

ตารางที่ 4.3 ค่ามุมที่วัดได้ ค่าแกน x, y และ z ของ Gyro Module และค่าความผิดพลาดของค่ามุมที่วัดได้เทียบกับค่าแกน y จากเซนเซอร์ ขณะนั่งหลังค่อม

ค่ามุมที่วัดได้ (องศา)	ค่าแกน x (องศา)	ค่าแกน y (องศา)	ค่าแกน z (องศา)	ค่าความผิดพลาดของค่ามุมที่วัดได้เทียบกับค่าแกน y จากเซนเซอร์
20	83	9	128	1.22
25	82	13	136	0.93
30	80	18	145	0.67
35	80	19	148	0.84
40	79	23	149	0.74
45	78	28	155	0.61
50	79	37	165	0.35

จากตารางที่ 4.3 แสดงค่ามุมที่วัดได้ และค่าแกน x, y และ z ของ Gyro Module โดยทดสอบการเอนโมดูลแล้วบันทึกผลหาค่าความผิดพลาดของค่ามุมที่วัดได้เทียบกับค่าแกน y จากเซนเซอร์ (อ้างอิงค่ามุมที่วัดได้ โดยเริ่มต้นที่ 20 องศา) สามารถนำมาพล็อตเป็นกราฟแสดงความสัมพันธ์ระหว่างค่าที่อ่านได้จากเซนเซอร์และค่ามุมที่วัดได้ขณะนั่งหลังค่อม แสดงดังรูปที่ 4.23



รูปที่ 4.23 กราฟแสดงความสัมพันธ์ระหว่างค่าที่อ่านได้จากเซนเซอร์และค่ามุมที่วัดได้ขณะนั่งหลังค่อม

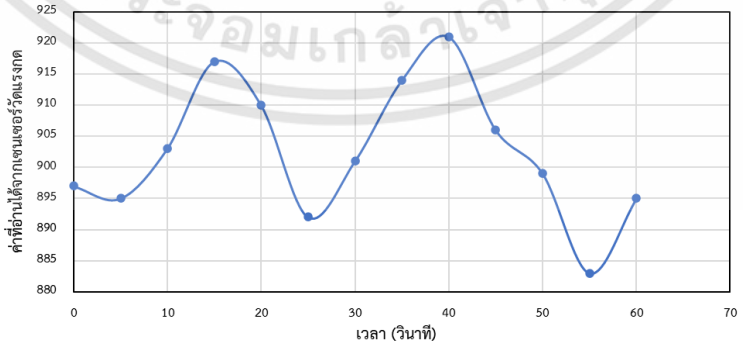
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากการทดสอบผู้ที่นั่งเกินเวลาที่กำหนดสามารถแสดงได้ดังตารางที่ 4.4

ตารางที่ 4.4 เวลา และค่าที่อ่านได้จากเซนเซอร์วัดแรงกด

เวลา (วินาที)	ค่าที่อ่านได้จากเซนเซอร์วัดแรงกด
0	897
5	895
10	903
15	917
20	910
25	892
30	901
35	914
40	921
45	906
50	899
55	883
60	895

จากตารางที่ 4.4 แสดงค่าที่อ่านได้จากเซนเซอร์วัดแรงกด และเวลา โดยทดสอบแล้วบันทึกผล สามารถนำมาพล็อตเป็นกราฟแสดงความสัมพันธ์ระหว่างค่าที่อ่านได้จากเซนเซอร์วัดแรงกด และเวลา แสดงดังรูปที่ 4.24



รูปที่ 4.24 กราฟแสดงความสัมพันธ์ระหว่างค่าที่อ่านได้จากเซนเซอร์วัดแรงกด และเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ทำการทดสอบชิ้นงาน และระบบทั้งหมด

4.6.1 ทำการทดสอบเมื่อผู้สวมใส่มีแนวโน้มในการนั่งหลังค่อม

ทำการทดสอบโดยมีขั้นตอนดังนี้

1) ทำการทดสอบโดยให้ผู้สวมใส่นั่งหลังค่อม แสดงดังรูปที่ 4.25

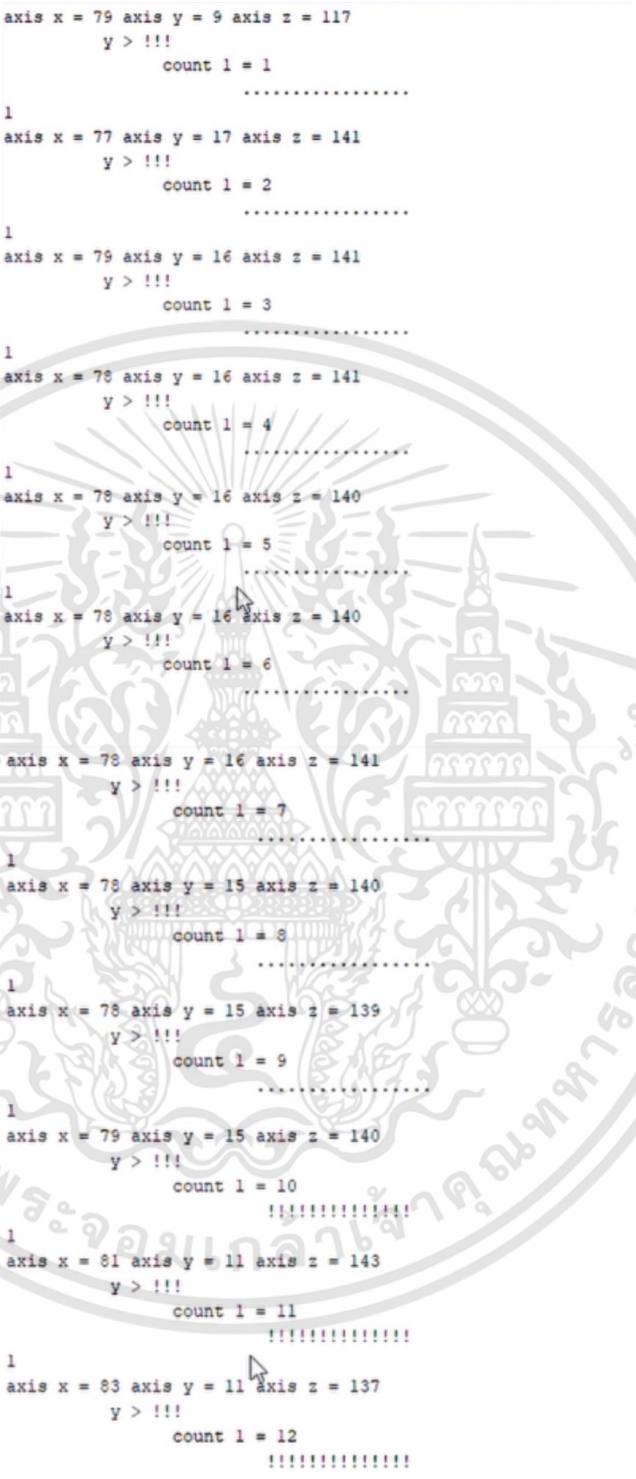
รูปที่ 4.25 ทดสอบโดยให้ผู้สวมใส่นั่งหลังค่อม

2) เมื่อระบบตรวจพบว่าในระยะเวลา 10 วินาที ผู้สวมใส่มีแนวโน้มในการนั่งหลังค่อม ระบบจะมีเสียงแจ้งเตือน และมอเตอร์มีการสั่น เพื่อแจ้งเตือนให้ผู้สวมใส่นั่งหลังตรง

3) เมื่อผู้สวมใส่นั่งหลังตรงเสียงแจ้งเตือน และการสั่นของมอเตอร์จะหยุดทันที

4) และสามารถเรียกดูค่าได้จาก Serial Monitor แสดงดังรูปที่ 4.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



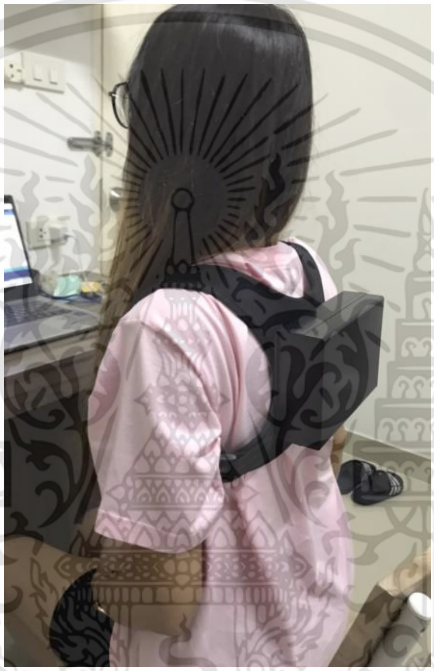
รูปที่ 4.26 ค่าที่แสดงบน Serial Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.2 ทำการทดสอบเมื่อผู้สวมใส่นั่งเป็นระยะเวลานานเกินที่กำหนด

ทำการทดสอบโดยมีขั้นตอนดังนี้

1) ทำการทดสอบโดยให้ผู้สวมใส่นั่งเป็นระยะเวลามากกว่า 10 วินาที โดยที่นั่งหลังตรง แสดงดังรูปที่ 4.27



รูปที่ 4.27 ทดสอบโดยให้ผู้สวมใส่นั่งเป็นระยะเวลามากกว่า 10 วินาที

2) เมื่อระบบตรวจพบว่านั่งเป็นระยะเวลามากกว่า 10 วินาที โดยที่นั่งหลังตรง ระบบจะมีเสียงแจ้งเตือน และมอเตอร์มีการสั่น เพื่อแจ้งเตือนให้ผู้สวมใส่เปลี่ยนอิริยาบถเป็นแบบอื่น เช่น ยืน หรือการเปลี่ยนท่าทางในการนั่ง เป็นต้น

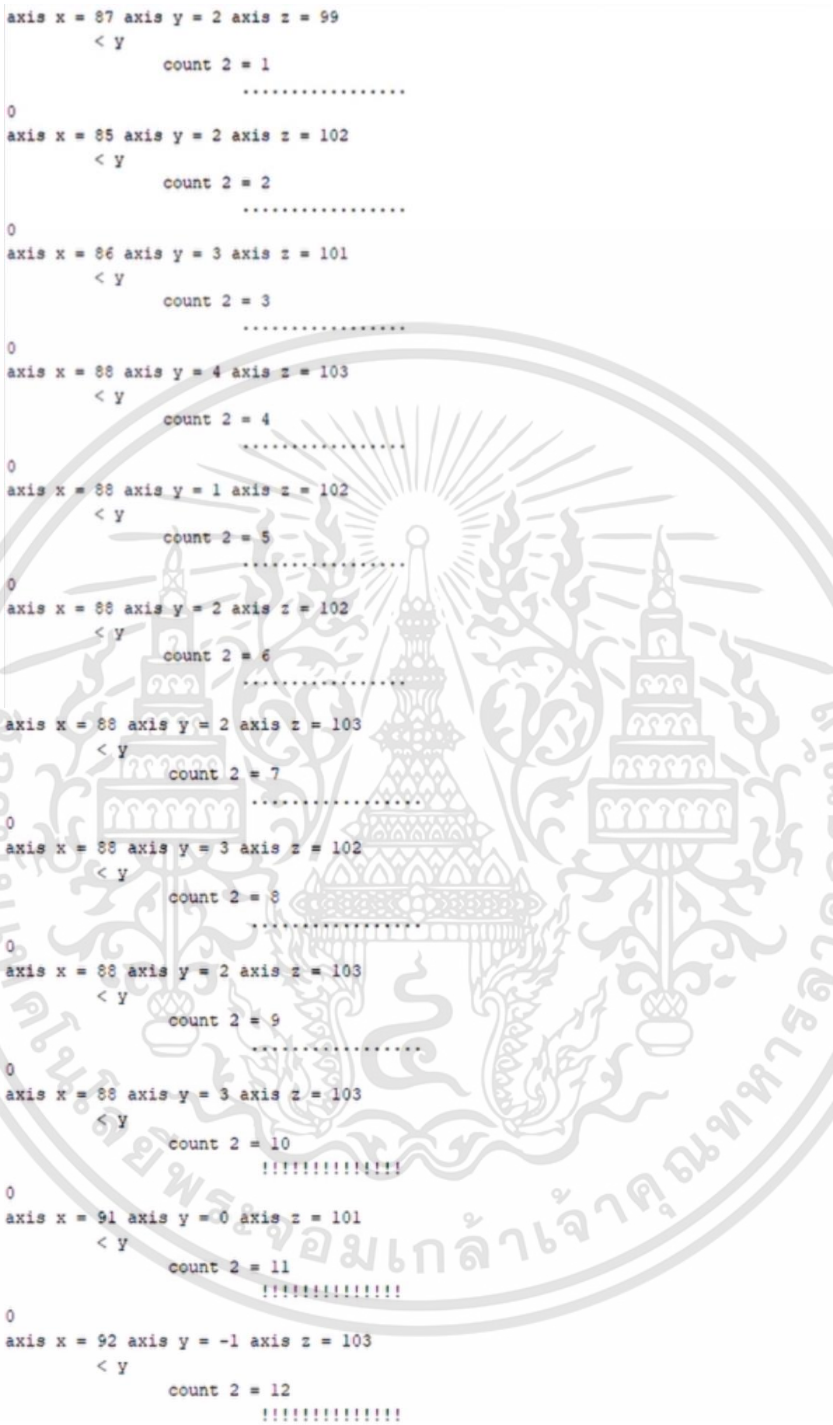
3) เมื่อผู้สวมใส่เปลี่ยนอิริยาบถจากการนั่งเป็นยืน แสดงดังรูปที่ 4.28 เสียงแจ้งเตือน และการสั่นของมอเตอร์จะหยุดทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.28 ผู้สวมใส่เปลี่ยนอิริยาบถจากการนั่งเป็นยืน

4) และสามารถเรียกดูค่าได้จาก Serial Monitor แสดงดังรูปที่ 4.29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.29 ค่าที่แสดงบน Serial Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.3 ทำการทดสอบเมื่อผู้สวมใส่มีแนวโน้มในการนั่งหลังค่อม และนั่งเป็นระยะเวลานานเกินที่กำหนด

ทำการทดสอบโดยมีขั้นตอนดังนี้

1) ทำการทดสอบโดยให้ผู้สวมใส่นั่งหลังค่อม และนั่งเป็นระยะเวลามากกว่า 10 วินาที แสดงดังรูปที่ 4.30

รูปที่ 4.30 ทดสอบโดยให้ผู้สวมใส่นั่งหลังค่อม และนั่งเป็นระยะเวลามากกว่า 10 วินาที

2) เมื่อระบบตรวจพบว่าผู้สวมใส่นั่งหลังค่อม และนั่งเป็นระยะเวลามากกว่า 10 วินาที ระบบจะมีเสียงแจ้งเตือน และมอเตอร์มีการสั่น เพื่อแจ้งเตือนให้ผู้สวมใส่นั่งหลังตรง และเปลี่ยนอิริยาบถเป็นแบบอื่น เช่น ยืน หรือการเปลี่ยนท่าทางในการนั่ง เป็นต้น

3) เมื่อผู้สวมใส่เปลี่ยนจากนั่งหลังค่อมเป็นหลังตรง แสดงดังรูปที่ 4.31 แต่ยังมีการนั่งเป็นระยะเวลามากกว่า 10 วินาที ระบบก็ยังมีเสียงแจ้งเตือน และมอเตอร์มีการสั่นอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.31 ผู้สวมใส่เปลี่ยนจากนั่งหลังค่อมเป็นหลังตรง

4) แต่เมื่อพบว่าผู้สวมใส่นั่งหลังตรง และเปลี่ยนอิริยาบถจากการนั่งเป็นยืน แสดงดังรูปที่ 4.32 เสียงแจ้งเตือน และการสั่นของมอเตอร์จะหยุดทันที

รูปที่ 4.32 ผู้สวมใส่เปลี่ยนอิริยาบถจากการนั่งเป็นยืน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) และสามารถเรียกดูค่าได้จาก Serial Monitor แสดงดังรูปที่ 4.33

```
axis x = 77 axis y = 16 axis z = 141
         y > !!!
               count 1 = 1
               count 2 = 1
                       ..................
0
axis x = 79 axis y = 14 axis z = 138
         y > !!!
               count 1 = 2
               count 2 = 2
                       ..................
0
axis x = 78 axis y = 14 axis z = 138
         y > !!!
               count 1 = 3
               count 2 = 3
                       ..................
0
axis x = 78 axis y = 13 axis z = 137
         y > !!!
               count 1 = 4
               count 2 = 4
                       ..................
0
axis x = 79 axis y = 14 axis z = 139
         y > !!!
               count 1 = 5
               count 2 = 5
                       ..................
axis x = 78 axis y = 14 axis z = 138
         y > !!!
               count 1 = 6
               count 2 = 6
                       ..................
0
axis x = 78 axis y = 14 axis z = 137
         y > !!!
               count 1 = 7
               count 2 = 7
                       ..................
0
axis x = 78 axis y = 14 axis z = 137
         y > !!!
               count 1 = 8
               count 2 = 8
                       ..................
0
axis x = 79 axis y = 14 axis z = 138
         y > !!!
               count 1 = 9
               count 2 = 9
                       ..................
0
axis x = 78 axis y = 14 axis z = 138
         y > !!!
               count 1 = 10
               count 2 = 10
                        !!!!!!!!!!!!!!
```

รูปที่ 4.33 ค่าที่แสดงบน Serial Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.4 การทดสอบ Gyro Module เมื่อตรวจพบว่าผู้สวมใส่นั่งหลังค่อมและนั่งหลังตรง

จากการทดสอบสามารถแสดงได้ดังตารางที่ 4.5

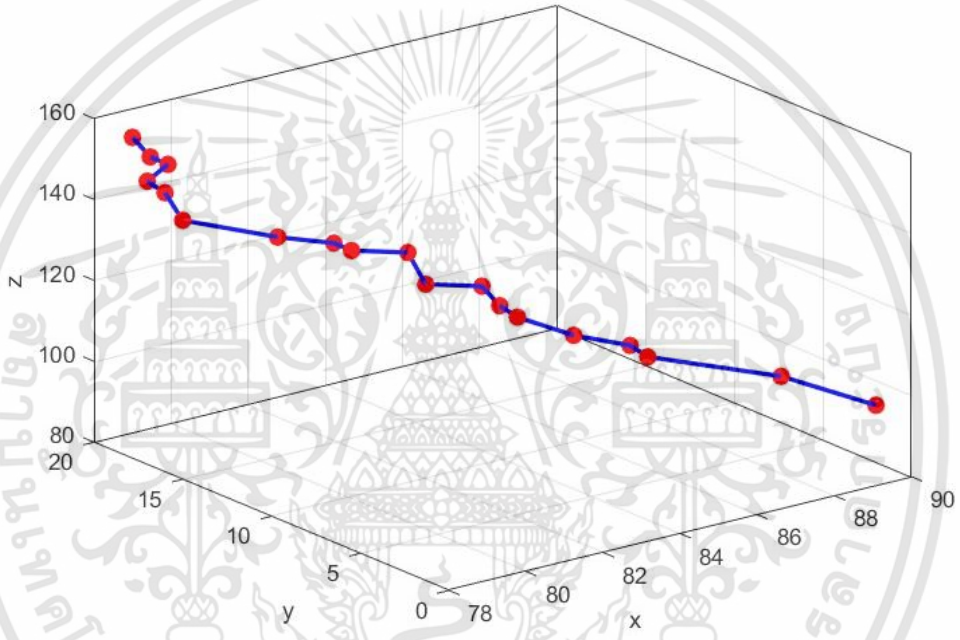
ตารางที่ 4.5 ค่าแกน x, y และ z ของ Gyro Module เมื่อตรวจพบว่าผู้สวมใส่นั่งหลังค่อมและนั่งหลังตรง

ค่าแกน x (องศา)	ค่าแกน y (องศา)	ค่าแกน z (องศา)	ตรวจจับได้หรือไม่ได้
90	2	94	ไม่ได้
88	3	104	ไม่ได้
85	4	114	ไม่ได้
85	5	115	ไม่ได้
84	6	118	ไม่ได้
83	7	123	ได้
83	8	124	ได้
83	9	127	ได้
82	10	128	ได้
82	11	134	ได้
81	12	135	ได้
81	13	135	ได้
80	14	137	ได้
78	15	144	ได้
78	16	149	ได้
78	17	150	ได้
79	18	150	ได้
79	19	150	ได้
79	20	153	ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้สวมใส่มีแนวโน้มในการนั่งหลังค่อม โดยกำหนดให้ Gyro Module ในแกน y มากกว่าหรือเท่ากับ 7 (โดยอ้างอิงจากแกน x ที่ 90 องศา คือผู้สวมใส่นั่งหลังตรง) จะตรวจจับว่าผู้สวมใส่นั่งหลังค่อม เมื่อแกน x น้อยกว่าหรือเท่ากับ 87 องศา

จากตารางที่ 4.5 สามารถนำมาพล็อตเป็นกราฟแสดงความสัมพันธ์ระหว่างแกน x, y และ z ของ Gyro Module เมื่อตรวจพบว่าผู้สวมใส่นั่งหลังค่อมและนั่งหลังตรง แสดงดังรูปที่ 4.34



รูปที่ 4.34 กราฟแสดงความสัมพันธ์ระหว่างแกน x, y และ z ของ Gyro Module เมื่อตรวจพบว่าผู้สวมใส่นั่งหลังค่อมและนั่งหลังตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 ทำการทดลองจัดเก็บข้อมูลลงในฐานข้อมูล Firebase

4.7.1 ทำการทดสอบการใช้งาน NodeMCU-32S 38 pins

- ทำการเขียนโปรแกรมทดสอบการทำงานดังรูปที่ 4.35 ว่า NodeMCU-32S 38 pins สามารถใช้งานได้หรือไม่

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);     // Initialize the LED_BUILTIN pin as an output
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, LOW);   // Turn the LED on (Note that LOW is the voltage level
  // but actually the LED is on; this is because
  // it is active low on the ESP-01)
  delay(1000);                      // Wait for a second
  digitalWrite(LED_BUILTIN, HIGH);  // Turn the LED off by making the voltage HIGH
  delay(2000);                      // Wait for two seconds (to demonstrate the active low LED)
}
```

รูปที่ 4.35 เขียนโปรแกรมเพื่อทดสอบ

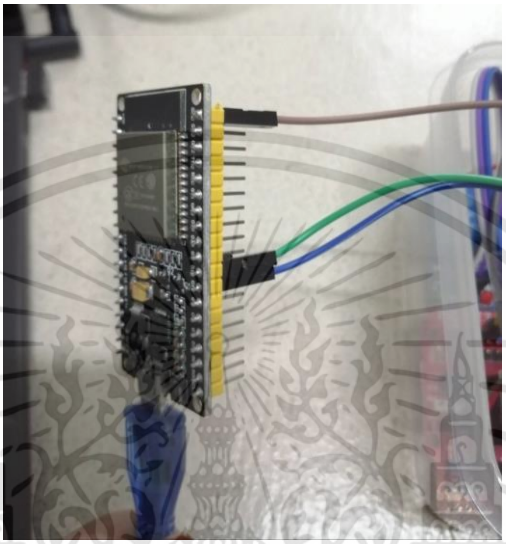
- เมื่อเขียนโปรแกรมเสร็จ ทำการอัพโหลดและรันโปรแกรม แสดงดังรูปที่ 4.36

รูปที่ 4.36 ทำการอัพโหลดและรันโปรแกรม

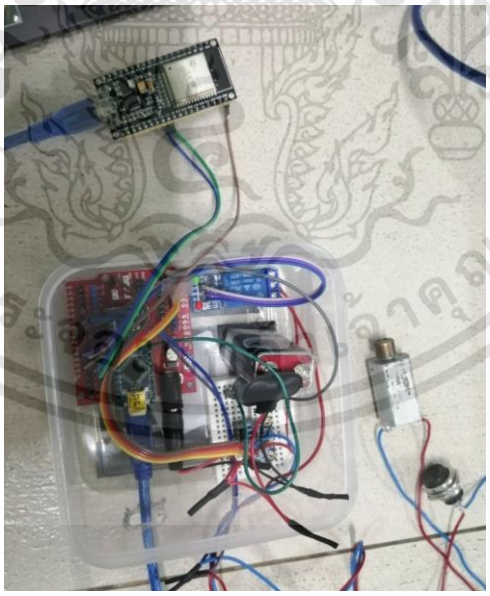
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7.2 ทำการเชื่อมต่อ Arduino Nano กับ NodeMCU-32S

- ทำการเชื่อมต่อ Arduino Nano กับ NodeMCU-32S แสดงดังรูปที่ 4.37 และ 4.38



รูปที่ 4.37 ขาที่ใช้เชื่อมต่อกับ Arduino Nano ของ NodeMCU-32S



รูปที่ 4.38 การเชื่อมต่อ Arduino Nano กับ NodeMCU-32S

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7.3 ทำการทดลองเขียนโปรแกรมเชื่อมต่อรับส่งข้อมูลระหว่าง Arduino Nano และ NodeMCU-32S

- ทำการเขียนโปรแกรมลงในบอร์ด Arduino Nano เพื่อเชื่อมต่อ Arduino Nano กับ NodeMCU-32S แสดงดังรูปที่ 4.39

```
#include<SoftwareSerial.h>
SoftwareSerial mySerial (10, 11);
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#include "Wire.h"

MPU6050 mpu;
int16_t ax, ay, az;
int16_t gx, gy, gz;
int valx , valy , valz;
char rd;
int prevVal;


int val1 , val2 ;
int valgy1 = 0 , valgy2 = 0;


#define sit 7
#define buzzer 8
#define motor 9

int alarm1 = 0;
int alarm2 = 0;
int count1 = 0;
int count2 = 0;

int counts = 0;
int county = 0;
int cmotor = 0;
void setup() {
  Serial.begin(9600);
  pinMode(sit, INPUT_PULLUP);

  Serial.begin(9600);
  Serial.println("Initialize MPU");
  mpu.initialize();
  Serial.println(mpu.testConnection() ? "Connected" : "Connection failed");

  pinMode(motor, OUTPUT);
  pinMode(buzzer, OUTPUT);
  digitalWrite(motor, HIGH);
  digitalWrite(buzzer, LOW);
  mySerial.begin(9600);
}

void loop() {


  Serial.println( digitalRead(sit));

  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  valx = map(ax, -17000, 17000, 0, 179);
  valy = map(ay, -17000, 17000, 0, 179);
  valz = map(az, -17000, 17000, 0, 179);
```

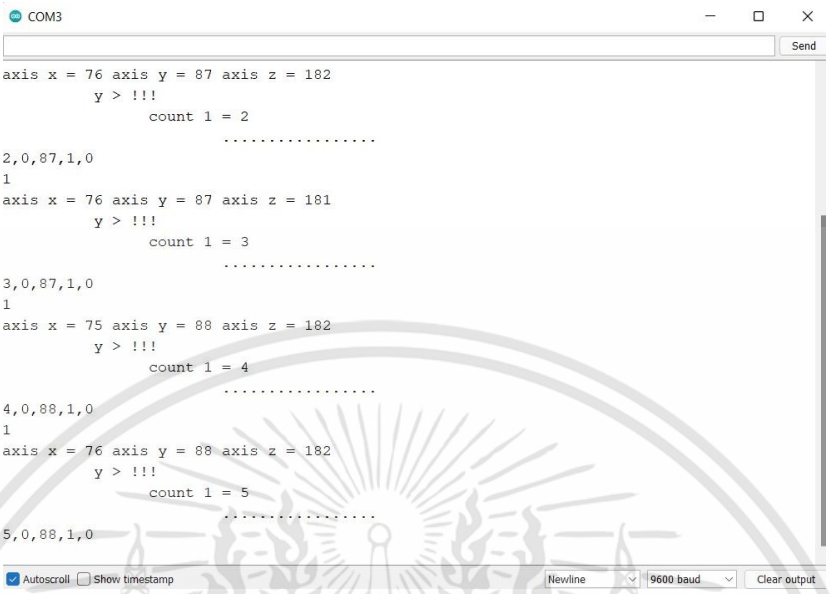
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Serial.print("axis x = ") ;
Serial.print(valx) ;
Serial.print(" axis y = ") ;
Serial.print(valy) ;
Serial.print(" axis z = ") ;
Serial.println(valz) ;


if ( valy >=  7) {
  Serial.println("          y > !!! ");
  count1 += 1;
  Serial.println("                count 1 = " + String(count1));
  county = count1;
  if ( count1 >= 10 )
  {
    alarm1 = 1;
  }
}
if (valy < 7 ) {
  Serial.println("         < y");
  count1 = 0;
  county = count1;
  alarm1 = 0;
}
if (  digitalRead(sit) == LOW )
{
  count2 += 1;
  Serial.println("              count 2 = " + String(count2));
  counts = count2;
  if ( count2 >= 10 )
  {
    alarm2 = 1;
  }
}
else
{
  count2 = 0;
  counts = count2;
  alarm2 = 0;
}
//---------------------------------------------------------------------------
if ( alarm1 == 1 | alarm2 == 1 )
{
  Serial.println("                    !!!!!!!!!!!!! ");
  digitalWrite(buzzer, HIGH);
  digitalWrite(motor, LOW);
  cmotor = 1;
}
else
{
  Serial.println("                    ..............  ");
  digitalWrite(buzzer, LOW);
  digitalWrite(motor, HIGH);
  cmotor = 0;
}

//---------------------------------------------------------------------------
//count, &gyro, &sit, &motor
Serial.println( String(county) + "," +String(counts) + "," + String(valy)  + "," + String(digitalRead(sit))   + "," + String(cmotor));
mySerial.println( String(county) + "," +String(counts) + "," + String(valy)  + "," + String(digitalRead(sit))   + "," + String(cmotor));
delay(1000);
}
```

รูปที่ 4.39 โปรแกรมเชื่อมต่อ Arduino Nano กับ NodeMCU-32S

- ทำการอัพโหลด และรันโปรแกรม จากนั้นเรียกดูค่าจาก Serial Monitor แสดงดังรูปที่ 4.40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.40 ค่าที่แสดงผ่าน Serial Monitor

4.7.4 ทำการเชื่อมต่อ NodeMCU-32S กับ Firebase เพื่อจัดเก็บข้อมูลลงในฐานข้อมูล

- ทำการเขียนโปรแกรมลงในบอร์ด ESP32 เพื่อเชื่อมต่อ NodeMCU-32S กับ Firebase แสดงดังรูปที่ 4.41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <WiFi.h>
#include <WiFiClientSecure.h>


#include <IOXhop_FirebaseESP32.h>
#include <time.h>
// Set these to run example.
#define FIREBASE_HOST "https://database1-bba99-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "DvX1MDc7xhKYfZzShihiuyPnrTL1KKSnd13fDV6f"
#define WIFI_SSID "M" // ชื่อ wifi
#define WIFI_PASSWORD "mmmmmmmm" // รหัส wifi

//Time
char ntp_server1[20] = "pool.ntp.org";
char ntp_server2[20] = "time.nist.gov";
char ntp_server3[20] = "time.uni.net.th";
int timezone = 7 * 3600;

int dst = 0;
int HH, MM, sec;
int county = 0;
int counts = 0;

int starts = 0;
int gyro;
int sit;
int motor;

int se;
char datas[50];
String getdata;
bool stringComplete = false;  // whether the string is complete

void setup()
{
  Serial.begin(9600);


  // เริ่มเชื่อมต่ออายฟาย
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  // แสดงคำว่า "Start WiFi Connecting.." ในซีเรียลคอนโซล
  Serial.print("Start WiFi Connecting..");

  // ถ้าไม่สามารถเชื่อมต่ออายฟายได้ให้แสดง "." ในซีเรียลคอนโซล
  // ลูปเช็คสถานะวายฟายทุกๆ ครึ่งวินาที (500 Millisecound)
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  // แสดงคำว่า "WiFi Connected:" และหมายเลข IP ที่ได้รับในซีเรียลคอนโซล
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
  Serial.println();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  configTime(timezone, dst, ntp_server1, ntp_server2, ntp_server3);
  Serial.println("\nWaiting for time");
  while (!time(nullptr)) {
    Serial.print(".");
    delay(1000);
  }
  Serial.println("");

  // เริ่มเชื่อมต่อ Firebase
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

  Serial2.begin(9600);
}


void loop ()
{

  if ( Serial2.available())
  {
    String getdata = Serial2.readString();
    Serial.println(" getdata =" + String(getdata));
    getdata.toCharArray(datas, 50); //count gyro sit motor
    int n = sscanf(datas, "%d,%d,%d,%d,%d", &county, &counts, &gyro, &sit, &motor );
    Serial.print(F("n="));
    Serial.println(n);
    Serial.print(F("        county="));
    Serial.print(county);
    Serial.print(F("        counts="));
    Serial.print(counts);
    Serial.print(F("        gyro="));
    Serial.print(gyro);
    Serial.print(F("        sit="));
     if ( sit == 0 )
    {
      se=1;
    }
    else
    {
      se=0;
    }
    Serial.print(se);
    Serial.print(F("        motor="));
    Serial.print(motor);

    Serial.println();
```

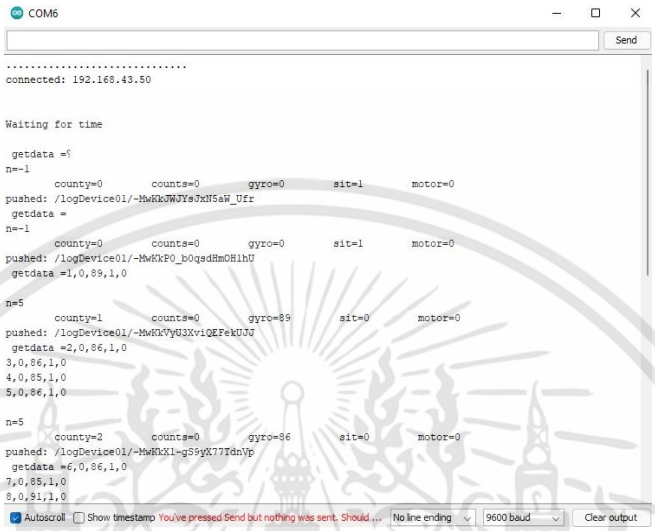
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    starts = 1;
  } // end Serial2

  NowString();
  if ( starts == 1 )
  {

    StaticJsonBuffer<200> jsonBuffer;
    JsonObject& root = jsonBuffer.createObject();
    root["count_gyro"] = county;
    root["count_sit"] = counts;
    root["gyro"] = gyro;
    root["sit"] = se;
    root["motor"] = motor;
    root["time"] = NowString();


    // set value
    String name = Firebase.push("logDevice01", root);
    // handle error
    if (Firebase.failed()) {
      Serial.print("setting /number failed:");
      Serial.println(Firebase.error());
      return;
    }
    Serial.print("pushed: /logDevice01/");
    Serial.println(name);

    delay(1000);
    starts = 0;
  }


}// end loop

String NowString() {
  time_t now = time(nullptr);
  struct tm* p_tm = localtime(&now);
  String timeNow = "";
  HH = String(p_tm->tm_hour).toInt();
  MM = String(p_tm->tm_min).toInt();
  sec = String(p_tm->tm_sec).toInt();
  timeNow += String(p_tm->tm_hour);
  timeNow += ":";
  timeNow += String(p_tm->tm_min);
  timeNow += ":";
  timeNow += String(p_tm->tm_sec);
  timeNow += " ";
  timeNow += String(p_tm->tm_mday);
  timeNow += "-";
  timeNow += String(p_tm->tm_mon + 1);
  timeNow += "-";
  timeNow += String(p_tm->tm_year + 1900);
  return timeNow;
}
```

รูปที่ 4.41 โปรแกรมการเชื่อมต่อ NodeMCU-32S กับ Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการอัพโหลด และรันโปรแกรม จากนั้นเรียกดูค่าจาก Serial Monitor แสดงดังรูปที่ 4.42



รูปที่ 4.42 ค่าที่แสดงผ่าน Serial Monitor

4.7.5 ทำการส่งข้อมูล และจัดเก็บข้อมูลไว้ในฐานข้อมูล

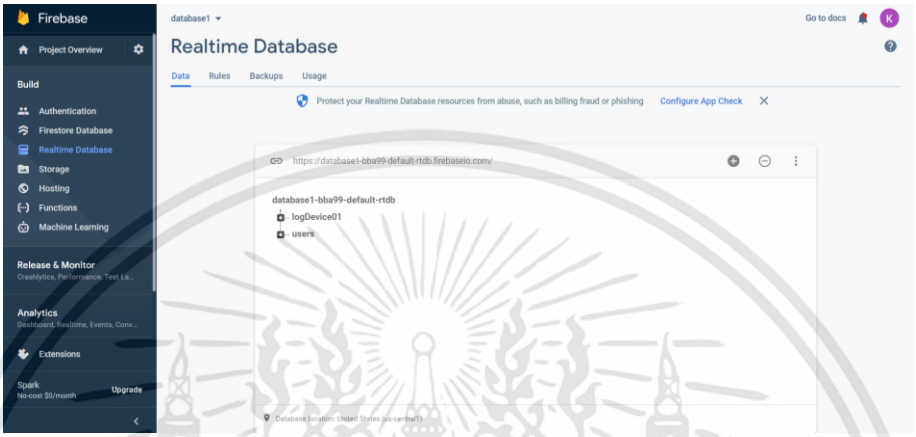
- เมื่อทำการเขียนโปรแกรมจากรูปที่ 4.41 สำเร็จ จะสามารถส่งข้อมูลที่ได้เขียนโปรแกรมไว้ไปเก็บข้อมูลค่าต่าง ๆ ไว้ใน Firebase ซึ่งเป็น Realtime Database ซึ่งจะแสดงค่า และเก็บข้อมูลไว้ดังรูปที่ 4.43

รูปที่ 4.43 ค่าต่าง ๆ ที่แสดง และเก็บข้อมูลไว้ใน Firebase

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7.6 ทำการเก็บข้อมูลผู้ใช้งาน (Users) และ NodeMCU

- เก็บข้อมูลผู้ใช้งาน และ NodeMCU ที่ ฐานข้อมูล Firebase แสดงดังรูปที่ 4.44



รูปที่ 4.44 การเก็บข้อมูลผู้ใช้งาน และ NodeMCU

4.8 ทำการสร้างหน้าเว็บเพจเพื่อแสดงผลการแจ้งเตือน

1) การเชื่อมต่อฐานข้อมูล Firebase กับ Node.js โดยที่คำสั่งแสดงดังรูปที่ 4.45

```
export const FIREBASE_CONFIGURATION = {
  apiKey: "AIzaSyB-zS0LHod0r_in9wig4qV6wDlmBzErVn8",
  authDomain: "database1-bba99.firebaseapp.com",
  databaseURL: "https://database1-bba99-default-rtdb.firebaseio.com",
  projectId: "database1-bba99",
  storageBucket: "database1-bba99.appspot.com",
  messagingSenderId: "951870625573",
  appId: "1:951870625573:web:65296a475155a9751deccc",
  measurementId: "G-STWBWCVH6X"
}
```

รูปที่ 4.45 คำสั่งการเชื่อมต่อระหว่างฐานข้อมูล Firebase กับ Node.js

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การเชื่อมต่อฐานข้อมูล Realtime Database ในฐานข้อมูล Firebase โดยที่คำสั่งแสดงดังรูปที่ 4.46

```
1  import firebase from "firebase/app"
2  import "firebase/database";
3  import {FIREBASE_CONFIGURATION} from '@/config'
4
5  firebase.initializeApp(FIREBASE_CONFIGURATION)
6  export default firebase.database()
7
```

รูปที่ 4.46 คำสั่งการเชื่อมต่อฐานข้อมูล Realtime Database ในฐานข้อมูล Firebase

3) การนำข้อมูลของผู้ใช้งาน (User) ในฐานข้อมูล Firebase มาใช้งาน

- การเรียกใช้ข้อมูลทั้งหมดของ User ในฐานข้อมูล Firebase โดยจะสามารถสร้างหรือเพิ่ม User ใหม่ลงฐานข้อมูล แก้ไขและลบข้อมูลของ User ได้ และถ้าส่ง ID เข้าไปดูในฐานข้อมูลสามารถรู้ข้อมูลของ ID นั้นได้ โดยจะนำข้อมูลนี้ไปใช้ในหน้า Sign in Sign up Edit และ Change password

4) การเรียกใช้ข้อมูล Log Device ในฐานข้อมูล Firebase

- การนำข้อมูล Log Device ในฐานข้อมูล Firebase ทั้งหมดเข้ามา เพื่อนำไปเรียกใช้ในหน้าต่าง ๆ

5) ทำการตั้งชื่อที่อยู่ของเว็บไซต์ในหน้าต่าง ๆ

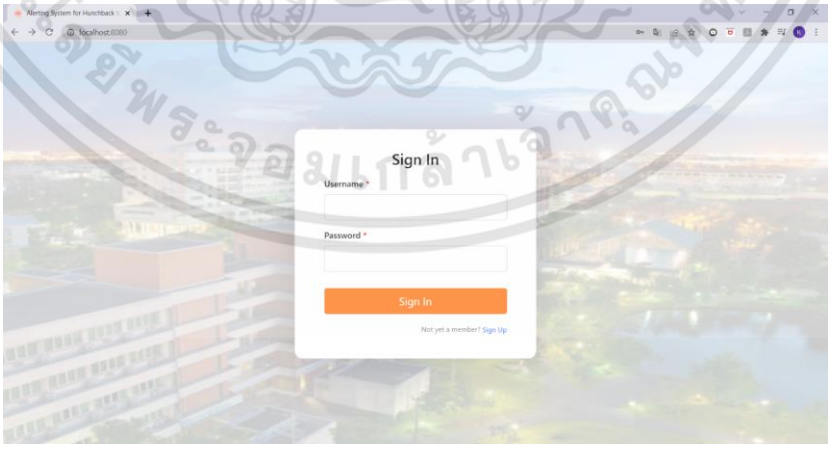
6) การเข้าสู่ระบบ (Sign In)

- จากรูปที่ 4.47 แสดงการสร้างหน้าการเข้าสู่ระบบ ซึ่งประกอบไปด้วย Username และ Password ในการเข้าระบบ โดยจะเก็บค่า input ของ Username เป็นประเภท text และเก็บค่า input ของ Password เป็นประเภท Password เพื่อให้เมื่อเวลากรอก Password จะไม่แสดงขึ้นจะแสดงเป็น dot และในหน้าของ Sign In ดังรูปที่ 4.48 จะทำการเชื่อมต่อไปยังหน้าของ Sign Up สำหรับบุคคลที่ยังไม่ได้เป็นสมาชิกเพื่อให้สามารถลงทะเบียนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```vue
<template>
  <div class="vertical-center">
    <div class="inner-block">
      <div class="vue-tempalte">
        <div class="home">
          <InputForm
            ref="signInRef"
            headerName="Sign In"
            sumitButtonName="Sign In"
            :items="formItems"
            @submit="submit"
          ></InputForm>
          <p class="signup text-right mt-2">
            Not yet a member? <router-link to="/signup">Sign Up</router-link>
          </p>
          <LoadingIcon :showLoading="showLoading" ></LoadingIcon>
        </div>
      </div>
    </div>
  </div>
</template>

<script>
import UsersDataService from "../services/UsersDataService";
import LoadingIcon from '@/components/LoadingIcon.vue'
import InputForm from '@/components/InputForm.vue'

export default {
  name: 'SignInView',
  components: {
    LoadingIcon,
    InputForm
  },
  data () {
    return {
      formItems: [
        { label: "Username", type: "text", name: "username", required: true },
        { label: "Password", type: "password", name: "password", required: true },
      ],
      showLoading: false,
      checked: false,
    }
  },
```

รูปที่ 4.47 การสร้างหน้า Sign In

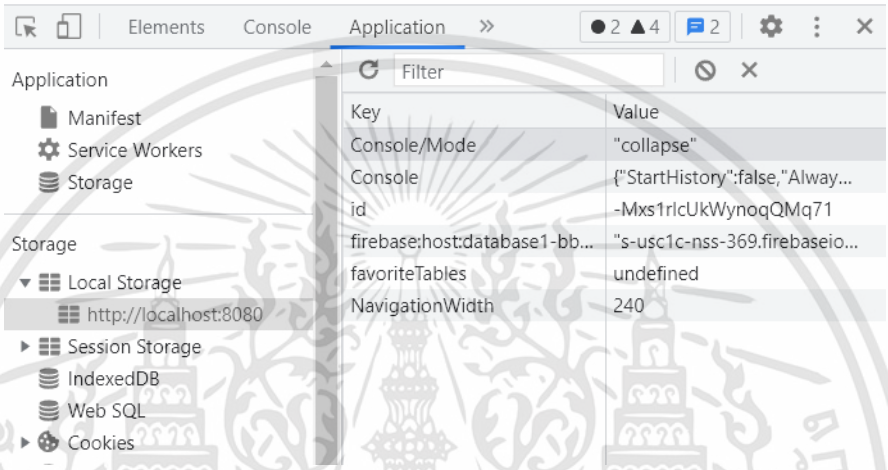


รูปที่ 4.48 Web Page ในหน้า Sign In

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

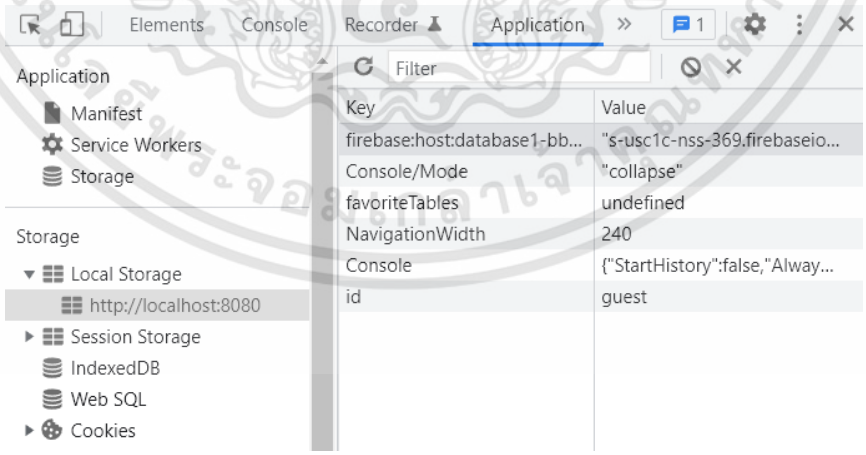
7) การเก็บค่า ID ใน Local Storage เมื่อ Log in

- เมื่อเข้าระบบแล้ว จะทำการตรวจสอบ ID ของ user ใน Firebase แล้วนำ ID User ไปเก็บค่าใน Local Storage โดยเป็น storage ของเบราวเซอร์ เพื่อที่เมื่อรีเฟรชแล้วจะได้ไม่ต้องไปเข้าหน้า Sign In ใหม่ ดังรูปที่ 4.49



รูปที่ 4.49 การเก็บค่า ID User ใน Local Storage

- ถ้าเข้าระบบด้วย Guest ก็จะนำค่า Guest ไปเก็บค่าเป็นค่า ID ใน Local Storage แสดงดังรูปที่ 4.50 โดยเป็น Storage ของเบราวเซอร์เพื่อที่เมื่อรีเฟรชแล้วจะได้ไม่ต้องไปเข้าหน้า Sign In ใหม่



รูปที่ 4.50 การเก็บค่า Guest ใน Local Storage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) ระบบลงทะเบียน (Sign Up)

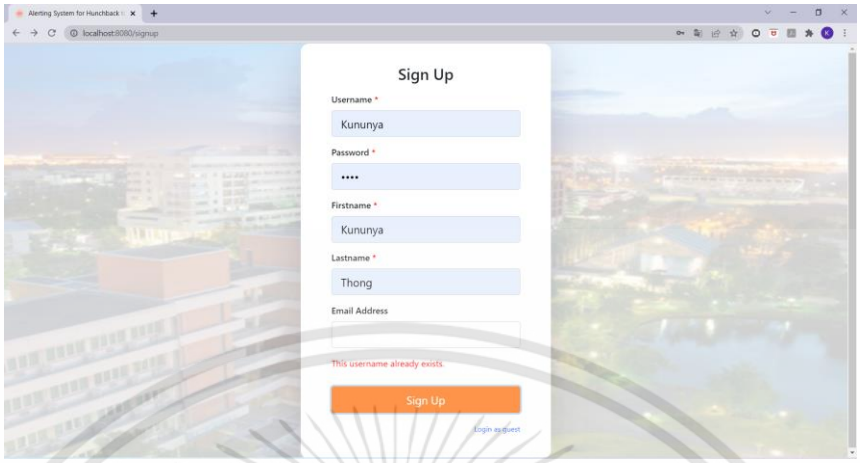
- โดยหน้าลงทะเบียน จะประกอบไปด้วย Username Password First name Last name และ Email Address โดยจะเก็บค่าUsername First name และ Last name เป็นประเภท Text Password เป็นประเภท Password และ Email Address เป็นประเภท Email โดยจะบังคับให้ผู้ใช้ลงทะเบียนว่าต้องกรอกข้อมูล Username Password First name และ Last name ส่วนของ Email Address ใส่หรือไม่ก็ได้

- โดยในหน้าลงทะเบียนนี้ยังเชื่อมต่อการ Login with Guest ไว้เพื่อเข้าระบบโดยที่ไม่ต้องลงทะเบียน แสดงดังรูปที่ 4.51

รูปที่ 4.51 หน้าต่างการลงทะเบียน

- เมื่อใส่ข้อมูลในหน้าลงทะเบียนครบแล้วกด submit จะทำการตรวจสอบ Username ในฐานข้อมูล Firebase โดยการสร้างเงื่อนไขว่าถ้า Username ที่ลงทะเบียนใหม่ตรงกับฐานข้อมูล Firebase จะไม่สามารถสร้าง ID User ได้ และจะแสดงผล “This username already exists” ดังรูปที่ 4.52 ถ้า Username ไม่ตรงกับ Username ในฐานข้อมูล Firebase จะทำการสร้าง User ID ขึ้นโดยการส่งค่า Username First name Last name และ Email address ไปยังฐานข้อมูล Firebase และเก็บค่า ID ใน Local Storage เมื่อสร้าง User ID แล้วจะสามารถเข้าไปยังหน้าหลักได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.52 เมื่อ Username ตรงกับ Username ในฐานข้อมูล

9) ส่วนของการแสดงผล และการแบ่งหน้าการแสดงผล Pagination

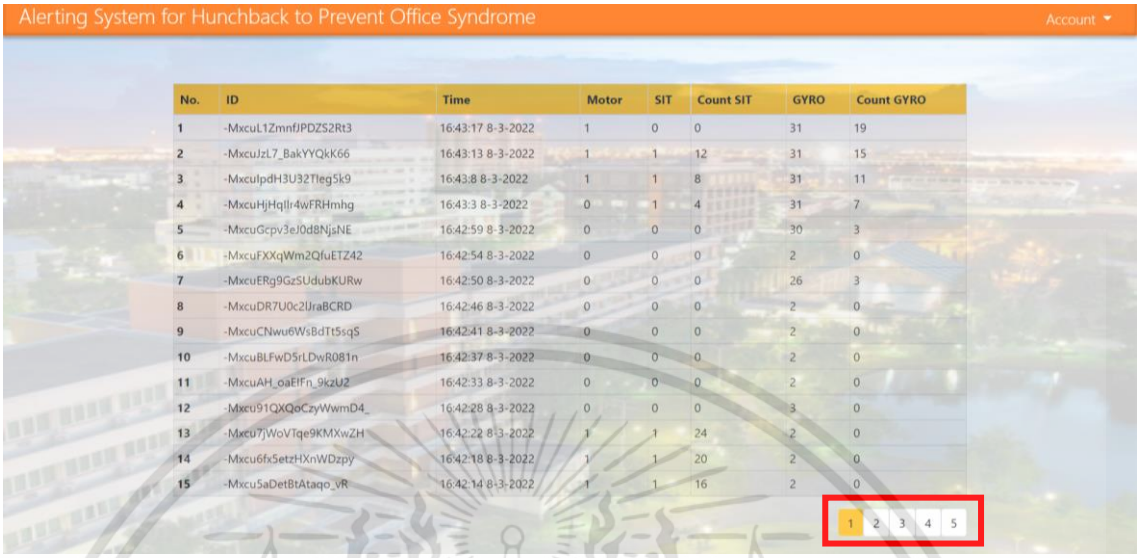
- ทำการสร้างตารางโดยสร้างตาราง header เป็นคอลัมน์แสดง ค่า No. ค่า ID ค่า Time ค่า Moter ค่า Sit ค่า Count Sit ค่า Gyro และ ค่า Count gyro ในส่วนของตาราง body จะนำข้อมูล Log Device ในฐานข้อมูล Firebase มาแสดง

- การเรียกใช้ข้อมูล Log Device จากฐานข้อมูลมาใช้ โดยจะไม่ปิดการใช้งานหากยังอยู่ในหน้าการแสดงผล เพื่อที่ข้อมูลจะอัพเดตตลอดเวลาแล้วแสดงผลแบบเรียลไทม์ และจะปิดการเรียกใช้ข้อมูล Log Device ในฐานข้อมูล เมื่อออกจากหน้านี้ไป

- ในฐานข้อมูล ในตอนแรกข้อมูลเวลาไม่ได้อยู่ในรูปแบบของ 00:00:00 และวันที่ 00-00-0000 หากนำค่าในฐานข้อมูลไปใช้เลยจะทำให้เกิดข้อผิดพลาดขึ้นได้ใน JavaScript จึงต้องทำข้อมูลเวลาให้อยู่ในรูปแบบ 00:00:00 และวันที่ 00-00-0000 เสียก่อน แล้วจึงจะนำข้อมูลเวลามาเปรียบเทียบ โดยที่กำหนดให้เวลาล่าสุดขึ้นก่อน

- การแบ่งหน้า และเปลี่ยนหน้าการแสดงผล Pagination เมื่อได้จำนวนข้อมูลมาทั้งหมดจะทำการแบ่งหน้าโดยการกำหนดให้ข้อมูลแต่ละหน้ามี 15 loops จากนั้นจะนำจำนวนข้อมูลทั้งหมดมาหารด้วย 15 แล้วปัดเศษขึ้นจะทำให้เราได้จำนวนหน้าดังรูปที่ 4.53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



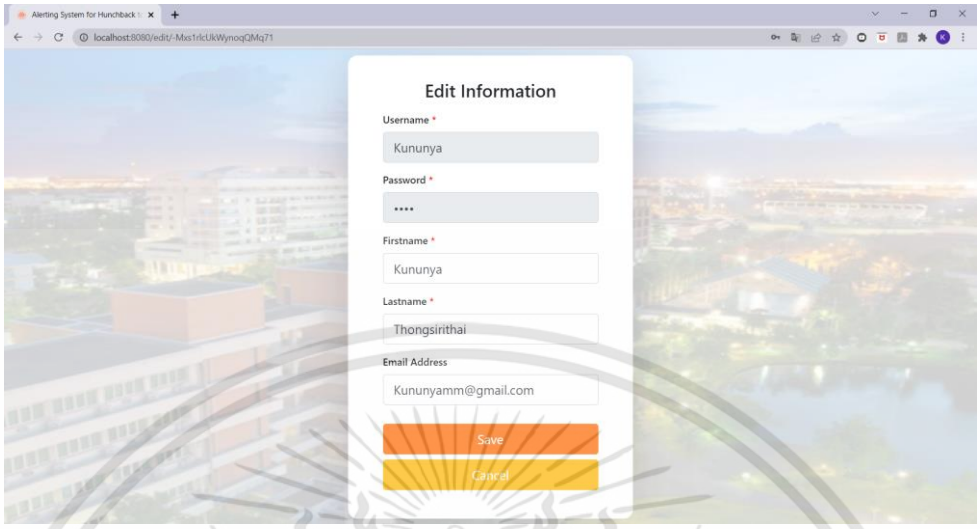
รูปที่ 4.53 การแบ่งหน้าการแสดงผล

10) ส่วนของเมนู Drop down

- ส่วนของการแก้ไขข้อมูล (Edit Information)

ส่วนของการแก้ไขข้อมูลนั้นได้ออกแบบเหมือนกันกับหน้าลงทะเบียน (Sign Up) เมื่อผู้ใช้กดแก้ไข (Edit) ในเมนู Drop down จะเข้ามายังหน้าการแก้ไขข้อมูล (Edit Information) ระบบจะทำการดึงข้อมูลเดิมมาแสดงให้เห็น และการแก้ไขข้อมูลกำหนดให้แก้ไขข้อมูลได้แค่ First name Last name และ Email address เท่านั้น ถ้าผู้ใช้แก้ไขข้อมูลแล้วกดบันทึก (Save) ระบบจะทำการแก้ไขข้อมูลในฐานข้อมูลด้วยและจะกลับไปยังหน้าแสดงผล ถ้าผู้ใช้กดยกเลิก (Cancel) ข้อมูลในฐานข้อมูลจะไม่มีการเปลี่ยนแปลง และจะกลับไปยังหน้าแสดงผล ซึ่งส่วนการแสดงผลของหน้าการแก้ไขข้อมูลแสดงได้ดังรูปที่ 4.54

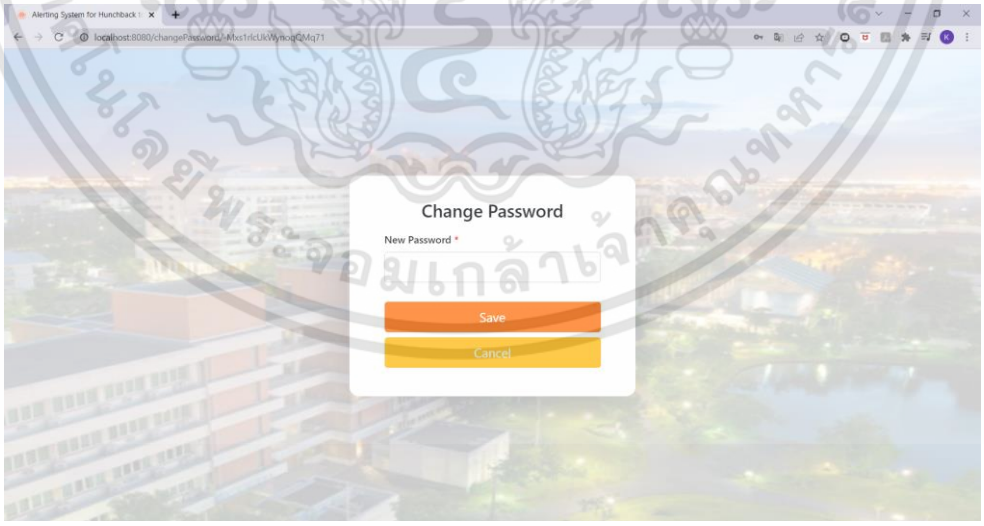
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.54 ส่วนของหน้าการแก้ไขข้อมูล (Edit Information)

- ส่วนของการเปลี่ยนรหัสผ่าน (Change Password)

หน้าการเปลี่ยนรหัสผ่าน โดยจะตั้งชื่อ Label ว่า New Password รับค่ามาในประเภทของ Password และตั้งชื่อของปุ่ม submit ว่า ‘Save’ และปุ่มการยกเลิกว่า ‘Cancel’ แสดงดังรูปที่ 4.55

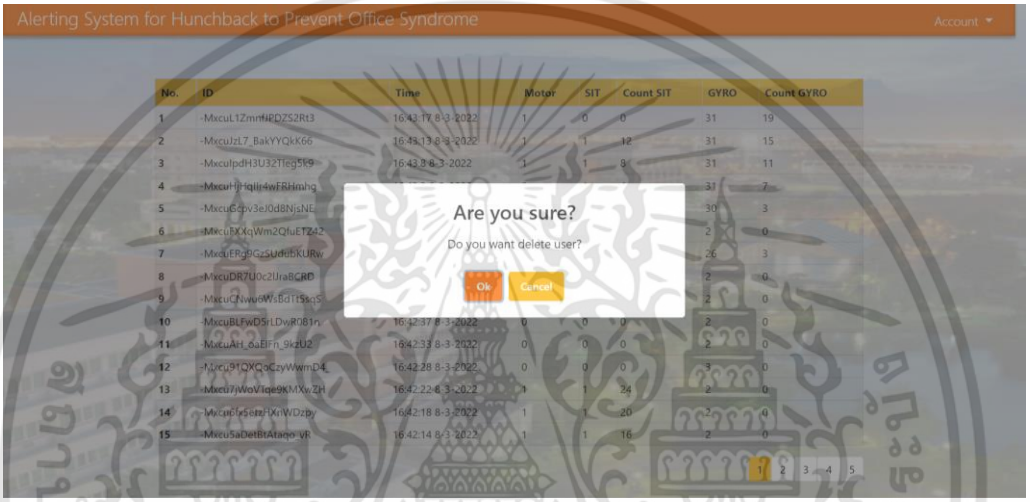


รูปที่ 4.55 หน้าการเปลี่ยนรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนของการลบข้อมูลผู้ใช้งาน (Delete User)

เมื่อผู้ใช้งานกดลบ (Delete) ที่เมนู Drop down หน้า modal จะขึ้นมา แล้วแสดงข้อความขึ้นว่า “Do you want delete user?” ขึ้นแสดงในรูปที่ 4.56 ถ้าผู้ใช้เลือก ‘Ok’ ระบบจะลบข้อมูล ID ของผู้ใช้ใน Local storage และลบข้อมูลทั้งหมดของผู้ใช้ในฐานข้อมูลออก แล้วจะไปยังหน้า Sign In แต่ถ้าผู้ใช้เลือก ‘Cancel’ จะยังคงอยู่หน้าแสดงผลอยู่



รูปที่ 4.56 หน้าการลบข้อมูลผู้ใช้งาน

- ส่วนของการลงทะเบียนเมื่อใช้ ID Guest

เมื่อ Guest กด Sign In ในเมนู Drop down ระบบจะทำการนำ ID Guest ออกจาก Local Storage แล้วไปยังหน้า Sign Up โดยที่คำสั่งส่วนของการลงทะเบียนเมื่อใช้ ID Guest แสดงดังรูปที่ 4.57

```
223   signup() {
224     localStorage.removeItem("id")
225     this.$router.push('/signup')
226   }
```

รูปที่ 4.57 คำสั่งส่วนของการลงทะเบียนเมื่อใช้ ID Guest

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนของการออกจากระบบ

เมื่อผู้ใช้กด Log out ในเมนู Drop down ระบบจะทำการนำ ID ออกจาก Local Storageแล้วไปยังหน้า Sign In โดยที่คำสั่งส่วนของการออกจากระบบ แสดงดังรูปที่ 4.58

```
219     logout() {
220       localStorage.removeItem("id")
221       this.$router.push('/')
222     },
```

รูปที่ 4.58 คำสั่งส่วนของการออกจากระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์นี้มีวัตถุประสงค์เพื่อตรวจจับท่าทางการนั่งเพื่อป้องกันการนั่งหลังค่อมและการนั่งนานเกินระยะเวลาที่กำหนด สามารถแจ้งเตือนด้วยเสียงและการสั่นด้วยเสียงหากผู้ที่สวมใส่นั่งหลังค่อมหรือนั่งนานเกินระยะเวลาที่กำหนด เพื่อให้ผู้สวมใส่เปลี่ยนท่าทางในการนั่ง ซึ่งปริญญานิพนธ์นี้ใช้บอร์ด Arduino Nano เป็นไมโครคอนโทรลเลอร์ควบคุมการทำงานของระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม จากการทดสอบระบบพบว่าเมื่อผู้สวมใส่นั่งหลังค่อมหรือนั่งนานเกินระยะเวลาที่กำหนดระบบจะแจ้งเตือนด้วยเสียงและการสั่นมอเตอร์ โดยที่ระบบสามารถเก็บบันทึกข้อมูลลงฐานข้อมูล และแสดงผลผ่านเว็บเพจได้ ซึ่งเป็นไปตามขอบเขตที่กำหนดไว้ข้างต้น และไมโครคอนโทรลเลอร์สามารถควบคุมการทำงานของระบบและอุปกรณ์ทุกอย่างได้

5.2 ข้อเสนอแนะ

ระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรมที่ได้ออกแบบขึ้นสามารถปรับปรุงและพัฒนาได้ในอนาคตให้สามารถใช้งานได้ดีขึ้น ซึ่งในขณะนี้ระบบนี้ยังมีข้อบกพร่องคือ อุปกรณ์ที่นำมาใช้ไม่เหมาะสมในการทำงาน เช่น การนำ force senser มาใช้ในระบบเนื่องจากข้อจำกัดของ force senser รองรับน้ำหนักได้น้อยทำให้เกิดข้อผิดพลาดขึ้น การติดมอเตอร์การสั่นในอุปกรณ์ที่จัดทำเมื่อมอเตอร์สั่นทำให้ระบบอ่านข้อมูลผิดพลาด และการนำโพรโทบอร์ดมาใช้ทำให้การเชื่อมต่อของสายไฟไม่แน่นหนาพอจึงควรเปลี่ยนเป็นการบัดกรีวงจรแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] “บทความ Arduino คืออะไร? ตอนที่2 (แนะนำ Arduino รุ่นต่าง ๆ กัน).” https://www.myarduino.net/article/4/%E0%B8%9A%E0%B8%97%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1-arduino-.

[2] “Arduino บอร์ด Nano 3.0 รุ่นใหม่ CH340 พร้อมสาย USB.” https://www.allnewstep.com/product/462/%E0%B8%82%E0%B8%B2%E0%B8%A2%E0%B8%9A%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B8%94-nano.

[3] “Nano 3.0 Prototype Shield I/O Extension Board.” https://www.ioxhop.com/product/628/nano-3-0-prototype-shield-i-o-extension-board.

[4] “สอนใช้งาน GY-521 IMU 3-axis Accelerometer/Gyro Module (MPU6050) กับ arduino.”https://www.myarduino.net/article/42/%E0%B8%AA%E0%B8%AD%E0%B8%99%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99.

[5] “GY-521 MPU6050 3 Axis Gyroscope Accelerometer Sensor Module.” https://www.ioxhop.com/product/484/gy-521-mpu6050-3-axis-gyroscope-accelerometer-sensor-module.

[6] “Force Sensitive Resistor 0.5".” https://electropeak.com/forse-sensitive-resistor-fsr-402#description.

[7] “FSR402 เซนเซอร์วัดแรงกด Force Sensing Resistor.” https://www.arduino4.com/product/685/fsr402-force-sensing-resistor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ)

[8] “5 V Active Buzzer.”
https://commandronestore.com/products/bi001.php.

[9] “โมดูลรีเลย์ 5V 1 ช่อง Relay Module 5V 1 Channel 250V 10A Active Low.”
https://www.spmicrotech.com/product/relay-module-5v-1-channel-low-level-trigger/.

[10] “Arduino ESP8266 (NodeMCU).”
http://narong.ece.engr.tu.ac.th/ei444/document/ESP8226.

[11] “ESP32S NodeMCU-32S esp32 DevKit Module WiFi Bluetooth IoT Development Board 38 pins บอร์ดพัฒนาโปรแกรมควบคุมวงจร ไวไฟ บลูทูธ 38 ขา.” https://www.tido.tech/index. php/product/esp32s-nodemcu-32s-esp32/.

[12] “ฐานข้อมูล (Database).”
https://sites.google.com/site/thekhnoloyisarsnthesit/xngkh-prakxb-khxng-thekhnoloyi-sarsnthes/than-khxmul-database.

[13] “Firebase Basic [EP1] : Firebase คืออะไร ?.”
http://www.arduino-makerzone.com/article/54/firebase-basic-ep1-firebase%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.

[14] “Arduino ตอนที่ 5 โครงสร้างภาษา C Arduino เบื้องต้น.”
https://www.arduinoone.com/index.php?module=knowledge&id=28.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ)

[15] “JavaScript คืออะไร.”
https://www.seibottech.co.th/news/javascript-%E0%B8.

[16] “ทำความรู้จักกับ Node.js.”
http://marcuscode.com/tutorials/nodejs/introducingnodejs#:~:text=Node.js%20%E0%B8%84%.

[17] “HTML: ภาษาเขียนเว็บ.”
https://www.bu.ac.th/knowledgecenter/executive_journal/july_sep_11/pdf/aw32.pdf.

[18] “Vue.Js 101 ฉบับมาเร็ว ไปเร็ววว.”
https://www.borntodev.com/2020/07/14/vue-js-101/.

[19] “โพรโทบอร์ด.”
https://sites.google.com/site/somyongregina/academic/electronic/prototoboard.

[20] “สายไฟ / สายจั้มเปอร์.”
https://www.ab.in.th/category/40/accessories/%E0.

[21] “ออฟฟิศซินโดรม (office syndrome).”
https://www.praram9.com/officesyndrome/.

[22] “office syndrome.”
https://s.isanook.com/he/0/ud/3/17973/office-syndrome.jpg.

[23] “อาการออฟฟิศซินโดรม โรคยอดฮิตของพนักงานออฟฟิศที่ไม่ควรมองข้าม.”
https://allwellhealthcare.com/office-syndrome/.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (ต่อ)

[24] “ความหมาย หลังค่อม.”

https://www.pobpad.com/%E0%B8%AB%E0%B8%A5%E0%B8%B1%E0%B8%87%E0%B8%84%E0%B9%88%E0%B8%AD%E0%B8%A1.

[25] “ใครที่นั่งหลังค่อมบ่อยๆ ระวังจะค่อมถาวร.”

https://www.sarirarak.com/single-post/2017/06/07/%E0%B8%99-.

[26] Pkourany. “I2CDEV_MPU6050.”

https://github.com/pkourany/I2 CDEV_MPU6 0 5 0 ?fbclid=IwAR1 K6 - CpSXtQBD5WmoFGy _81gPgAiNhboUmUEvaV5Aj2ro FaTwjLfufc5uc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

คำสั่งการทำงานระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรมมีโปรแกรมการทำงานซึ่งประกอบไปด้วยส่วนที่ 1 ไลบรารี และกำหนดตัวแปรเพื่อเก็บค่า ส่วนที่ 2 ส่วนการทำงานของ ระบบ (Setup) ส่วนที่ 3 การทำงานของ Gyro Module และ Force Senser สามารถแสดงได้ดังต่อไปนี้

```
#include<SoftwareSerial.h>
SoftwareSerial mySerial (10, 11);
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#include "Wire.h"

MPU6050 mpu;
int16_t ax, ay, az;
int16_t gx, gy, gz;
int valx , valy , valz;
char rd;
int prevVal;
int val1 , val2 ;
int valgy1 = 0 , valgy2 = 0;

#define sit 7
#define buzzer 8
#define motor 9

int alarm1 = 0;
int alarm2 = 0;
int count1 = 0;
int count2 = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int counts = 0;
int county = 0;
int cmotor = 0;
void setup() {
  Serial.begin(9600);
  pinMode(sit, INPUT_PULLUP);

  Serial.begin(9600);
  Serial.println("Initialize MPU");
  mpu.initialize();
  Serial.println(mpu.testConnection() ? "Connected" : "Connection failed");

  pinMode(motor, OUTPUT);
  pinMode(buzzer, OUTPUT);
  digitalWrite(motor, HIGH);
  digitalWrite(buzzer, LOW);
  mySerial.begin(9600);
}
void loop() {
  Serial.println( digitalRead(sit));
  mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

  valx = map(ax, -17000, 17000, 0, 179);
  valy = map(ay, -17000, 17000, 0, 179);
  valz = map(az, -17000, 17000, 0, 179);
  Serial.print("axis x = ") ;
  Serial.print(valx) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Serial.print(" axis y = ") ;
Serial.print(valy) ;
Serial.print(" axis z = ") ;
Serial.println(valz) ;


if ( valy >=  7) {
  Serial.println("          y > !!! ");
  count1 += 1;
  Serial.println("                count 1 = " + String(count1));
  county = count1;
  if ( count1 >= 10 )
  {
   alarm1 = 1;
  }
}
if (valy < 7 ) {
  Serial.println("         < y");
  count1 = 0;
  county = count1;
  alarm1 = 0;
}
if (  digitalRead(sit) == LOW )
{
  count2 += 1;
  Serial.println("                count 2 = " + String(count2));
  counts = count2;
  if ( count2 >= 10 )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  {
    alarm2 = 1;
  }
}
else
{
  count2 = 0;
  counts = count2;
  alarm2 = 0;
}
if ( alarm1 == 1 | alarm2 == 1 )
{
  Serial.println("                    !!!!!!!!!!!!! ");
  digitalWrite(buzzer, HIGH);
  digitalWrite(motor, LOW);
  cmotor = 1;
}
else
{
  Serial.println("                    ..............  ");
  digitalWrite(buzzer, LOW);
  digitalWrite(motor, HIGH);
  cmotor = 0;
}

//---------------------------------------------------------------------
//count, &gyro, &sit, &motor
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Serial.println( String(county) + "," +String(counts) + "," + String(valy)  + "," +
String(digitalRead(sit))   + "," + String(cmotor));
  mySerial.println( String(county) + "," +String(counts) + "," + String(valy)  + "," +
String(digitalRead(sit))   + "," + String(cmotor));
  delay(1000);
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

คำสั่งการเชื่อมต่อระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อม เพื่อป้องกันโรคออฟฟิศซินโดรมเข้าสู่ฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรมเข้าสู่ฐานข้อมูลมีโปรแกรมการทำงานซึ่งประกอบไปด้วยส่วนที่ 1 การเชื่อมต่อระหว่าง Arduino Nano กับ NodeMCU-ESP32S และส่วนที่ 2 ส่วนการการเชื่อมต่อระหว่าง NodeMCU-ESP32S กับฐานข้อมูล Firebase สามารถแสดงได้ดังต่อไปนี้

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <IOXhop_FirebaseESP32.h>
#include <time.h>
#define FIREBASE_HOST "https://database1-bba99-default-rtdb.firebaseio.com/"
#define FIREBASE_AUTH "DvX1MDc7xhKYfZzShihiuyPnrTLlKKSndl3fDV6f"
#define WIFI_SSID "M" // ชื่อ wifi
#define WIFI_PASSWORD "mmmmmmmm" // รหัส wifi

char ntp_server1[20] = "pool.ntp.org";
char ntp_server2[20] = "time.nist.gov";
char ntp_server3[20] = "time.uni.net.th";
int timezone = 7 * 3600;
int dst = 0;
int HH, MM, sec;
int county = 0;
int counts = 0;
int starts = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int gyro;
int sit;
int motor;
int se;
char datas[50];
String getdata;
bool stringComplete = false;  // whether the string is complete
void setup()
{
  Serial.begin(9600);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Start WiFi Connecting..");

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());
  Serial.println();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  configTime(timezone, dst, ntp_server1, ntp_server2, ntp_server3);
  Serial.println("\nWaiting for time");
  while (!time(nullptr)) {
    Serial.print(".");
    delay(1000);
  }
  Serial.println("");
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Serial2.begin(9600);
}
void loop ()
{
  if ( Serial2.available())
  {
    String getdata = Serial2.readString();
    Serial.println(" getdata =" + String(getdata));
    getdata.toCharArray(datas, 50); //count gyro sit motor
    int n = sscanf(datas, "%d,%d,%d,%d,%d", &county, &counts, &gyro, &sit, &motor );
    Serial.print(F("n="));
    Serial.println(n);
    Serial.print(F("         county="));
    Serial.print(county);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Serial.print(F("          counts="));
  Serial.print(counts);
  Serial.print(F("          gyro="));
  Serial.print(gyro);
  Serial.print(F("          sit="));
   if ( sit == 0 )
  {
    se=1;
  }
  else
  {
    se=0;
  }
  Serial.print(se);
  Serial.print(F("          motor="));
  Serial.print(motor);
  Serial.println();
  starts = 1;
}
NowString();
if ( starts == 1 )
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    StaticJsonBuffer<200> jsonBuffer;
    JsonObject& root = jsonBuffer.createObject();
    root["count_gyro"] = county;
    root["count_sit"] = counts;
    root["gyro"] = gyro;
    root["sit"] = se;
    root["motor"] = motor;
    root["time"] = NowString();
    String name = Firebase.push("logDevice01", root);
    if (Firebase.failed()) {
     Serial.print("setting /number failed:");
     Serial.println(Firebase.error());
     return;
    }
    Serial.print("pushed: /logDevice01/");
    Serial.println(name);
    delay(1000);
    starts = 0;
  }
}
String NowString() {
  time_t now = time(nullptr);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  struct tm* p_tm = localtime(&now);
  String timeNow = "";
  HH = String(p_tm->tm_hour).toInt();
  MM = String(p_tm->tm_min).toInt();
  sec = String(p_tm->tm_sec).toInt();
  timeNow += String(p_tm->tm_hour);
  timeNow += ":";
  timeNow += String(p_tm->tm_min);
  timeNow += ":";
  timeNow += String(p_tm->tm_sec);
  timeNow += " ";
  timeNow += String(p_tm->tm_mday);
  timeNow += "-";
  timeNow += String(p_tm->tm_mon + 1);
  timeNow += "-";
  timeNow += String(p_tm->tm_year + 1900);
  return timeNow;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

การแสดงผลบน Web Page ของระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของระบบแจ้งเตือนสำหรับผู้นั่งหลังค่อมเพื่อป้องกันโรคออฟฟิศซินโดรมที่แสดงบน Web Page ประกอบไปด้วย 14 หน้าต่าง คือ คำสั่งการเชื่อมต่อระหว่าง Node.js กับฐานข้อมูล Firebase, คำสั่งการเปิดใช้งานแอพพลิเคชั่น, คำสั่งการเรียกใช้ข้อมูล Log Device ในฐานข้อมูล Firebase, คำสั่งการนำข้อมูลของผู้ใช้งานในฐานข้อมูล Firebase มาใช้งาน, คำสั่งการตั้งชื่อที่อยู่ของเว็บไซต์ในหน้าต่าง ๆ , ส่วนของ Input form ที่เรียกใช้ในหน้าอื่น ๆ , ส่วนของ Loading Iconที่เรียกใช้ในหน้าอื่น ๆ , คำสั่งการเรียกใช้โลโก้ และชื่อ Title ของ Website title, การเรียกใช้พื้นหลัง, คำสั่งส่วนของระบบการลงทะเบียน, คำสั่งของการเข้าสู่ระบบ, คำสั่งหน้าการแสดงผล, คำสั่งสั่งหน้าแก้ไขข้อมูล และคำสั่งหน้าการเปลี่ยนรหัสผ่าน สามารถแสดงได้ดังต่อไปนี้

1. คำสั่งการเชื่อมต่อระหว่าง Node.js กับฐานข้อมูล Firebase

```
export const FIREBASE_CONFIGURATION = {
  apiKey: "AIzaSyB-zS0LHod0r_in9wig4qV6wDlmBzErVn8",
  authDomain: "database1-bba99.firebaseapp.com",
  databaseURL: "https://database1-bba99-default-rtdb.firebaseio.com",
  projectId: "database1-bba99",
  storageBucket: "database1-bba99.appspot.com",
  messagingSenderId: "951870625573",
  appId: "1:951870625573:web:65296a475155a9751deccc",
  measurementId: "G-STWBWCVH6X"
}
import firebase from "firebase/app"
import "firebase/database";
import {FIREBASE_CONFIGURATION} from '@/config'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
firebase.initializeApp(FIREBASE_CONFIGURATION)

export default firebase.database()
```

2. คำสั่งการเปิดใช้งานแอพพลิเคชั่น

```
import { createApp } from 'vue'

import App from './App.vue'

import router from './router'

import 'sweetalert2/dist/sweetalert2.min.css';

const app = createApp(App)

app.use(router)

app.mount('#app')
```

3. คำสั่งการเรียกใช้ข้อมูล Log Device ในฐานข้อมูล Firebase

```
import firebase from "../firebase";

const db = firebase.ref("/logDevice01");

class LogDeviceDataService {

    getAll() {

        return db;

    }

    off(event) {

        return db.off(event)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
}
export default new LogDeviceDataService();
```

4. คำสั่งการนำข้อมูลของผู้ใช้งานในฐานข้อมูล Firebase มาใช้งาน

```
import firebase from "../firebase";
const db = firebase.ref("/users");
class UserDataService {
    getAll() {
        return db
    }
    getById(id) {
        return db.orderByKey().equalTo(id)
    }
    getByField(fieldname, value) {
        return db.orderByChild(fieldname).equalTo(value)
    }
    create(user) {
        return db.push(user)
    }
    update(key, value) {
        return db.child(key).update(value)
    }
    delete(key) {
        return db.child(key).remove()
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    deleteAll() {
        return db.remove()
    }
    off(event) {
        return db.off(event)
    }
}
export default new UserDataService()
```

5. คำสั่งการตั้งชื่อที่อยู่ของเว็บไซต์ในหน้าต่าง ๆ

```
import { createRouter, createWebHistory } from 'vue-router'
import SignInView from '../views/SignInView.vue'
import SignUpView from '../views/SignUpView.vue'
import DetailView from '../views/DetailView.vue'
import EditView from '../views/EditView.vue'
import ChangePasswordView from '../views/ChangePasswordView.vue'
const routes = [
  {
    path: '/',
    name: 'signin',
    component: SignInView
  },
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    path: '/signup',
    name: 'signup',
    component: SignUpView
  },
  {
    path: '/detail',
    name: 'deatil',
    component: DetailView
  },
  {
    path: '/edit/:id',
    name: 'edit',
    component: EditView
  },
  {
    path: '/changePassword/:id',
    name: 'changePassword',
    component: ChangePasswordView
  },
}
const router = createRouter({
  history: createWebHistory(process.env.BASE_URL),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  routes
})
export default router
```

6. ส่วนของ Input form ที่เรียกใช้ในหน้าอื่น ๆ

```
<template>
  <form v-on:submit.prevent="onEnter">
    <h3>{{headerName}}</h3>
    <div v-for="(item, index) in items" :key="index">
      <div class="form-group">
        <label>{{item.label}} <span v-if="item.required" class="text-
required">*</span></label>
        <input :type="item.type" class="form-control form-control-lg"
:name="item.name" :required="item.required" :disabled="item.disabled"/>
      </div>
    </div>
    <div v-if="errorMesages.length > 0" class="errorMessage">
      <p>{{errorMesages.join(', ')}}</p>
    </div>
    <button class="btn btn-lg btn-block submit"
@click="submit">{{sumitButtonName}}</button>
    <button v-show="cancelButtonName" class="btn btn-lg btn-block cancel"
@click="cancel">{{cancelButtonName}}</button>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  </form>
</template>
<script>
export default {
  name: 'InputForm',
  props: {
    headerName: {
      type: String,
      required: true
    },
    items: {
      type: Array,
      required: true
    },
    sumitButtonName: {
      type: String,
      default: 'Submit'
    },
    cancelButtonName: {
      type: String
    },
  },
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mounted() {
  for (let item of this.items) {
    this.result[item.name] = null
  }
},
data () {
  return {
    result: {},
    errorMesages: []
  }
},
methods: {
  onEnter(e) {
    if (e.keyCode === 13) {
      this.submit(e)
    }
  },
  submit() {
    this.errorMesages = []
    for (let item of this.items) {
      let value = document.querySelector(`input[name=${item.name}]`).value
      if (![null, undefined, ""].includes(value)) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            this.result[item.name] = value
          } else {
            if (item.required) this.errorMesages.push(`${item.name} is required`)
          }
        }
        if (this.errorMesages.length === 0) {
          this.$emit('submit', this.result)
        }
      },
      cancel() {
        this.$emit('cancel')
      },
      showErrorMessage(messages) {
        this.errorMesages = messages
      },
      updateValue(value) {
        for (let item of this.items) {
          document.querySelector(`input[name=${item.name}]`).setAttribute('value',
value[item.name])
        }
      }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
</script>
<style scoped>
.form-group {
  margin-bottom: 15px;
}
.text-required {
  color: red;
}
.submit {
  margin-top: 30px;
}
.cancel {
  background-color: #FFC433 ;
}
</style>
```

7. ส่วนของ Loading Icon ที่เรียกใช้ในหน้าอื่น ๆ

```
<template>
  <div v-show="showLoading" class="loading">
    <span class="spinner-border"></span>
  </div>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
</template>
<script>
export default{
  name: 'LoadingIcon',
  props: {
    showLoading: {
      type: Boolean,
      required: true
    },
  }
}
</script>
<style scoped>
.loading {
  position: fixed;
  z-index: 999;
  height: 2em;
  width: 2em;
  margin: auto;
  top: 0;
  left: 0;
  bottom: 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  right: 0;
}
.loading:before {
  content: '';
  display: block;
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: radial-gradient(rgba(20, 20, 20,.8), rgba(0, 0, 0, .8));
  background: -webkit-radial-gradient(rgba(20, 20, 20,.8), rgba(0, 0, 0,.8));
}
.spinner-border {
  height: 3em;
  width: 3em;
  color: #FF8633;
  z-index: 1000
}
</style>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. คำสั่งการเรียกใช้โลโก้ และชื่อ Title ของ Website title

```
<!DOCTYPE html>
<html lang="">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <!-- TODO: logo and title-->
    <link rel="icon" href="<%= BASE_URL %>logo.png">
    <title>Alerting System for Hunchback to Prevent Office Syndrome</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.11.2/css/all.css" />
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.2.1/dist/css/bootstrap.min.css"
integrity="sha384-
GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS"
crossorigin="anonymous">
  </head>
  <body>
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.6/dist/umd/popper.min.js"
integrity="sha384-
wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut"
crossorigin="anonymous"></script>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.2.1/dist/js/bootstrap.min.js"
integrity="sha384-
B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k"
crossorigin="anonymous"></script>
    <noscript>
      <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work
properly without JavaScript enabled. Please enable it to continue.</strong>
    </noscript>
    <div id="app"></div>
    <!-- built files will be auto injected -->
  </body>
</html>
```

9. การเรียกใช้พื้นหลัง

```
<template>
  <div class="vue-tempalte">
    <img class="background" src="/background.jpeg" />
    <div class="App">
      <router-view />
    </div>
  </div>
</template>
<script>
export default {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  name: 'SignIn',
  mounted() {
    let id = localStorage.getItem("id")
    if (![undefined, null, 'guest'].includes(id)) {
      this.$router.push('/detail')
    }
  }
}
</script>>
<style>
* {
  box-sizing: border-box;
}
body,
html,
.App,
.vue-tempalte,
.vertical-center {
  width: 100%;
  height: 100%;
}
.vertical-center {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  display: flex;
  text-align: left;
  justify-content: center;
  flex-direction: column;
  min-height: 100vh;
  display: flex;
  font-weight: 400;
  justify-content: center;
}
.inner-block {
  width: 450px;
  margin: auto;
  background: #ffffff;
  box-shadow: 0px 14px 80px rgba(34, 35, 58, 0.2);
  padding: 40px 55px 45px 55px;
  border-radius: 15px;
  transition: all .3s;
  z-index: 1;
}
.vertical-center .form-control:focus {
  /* border-color: #fcb885; */
  box-shadow: none;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
.vertical-center h3 {
  text-align: center;
  margin: 0;
  line-height: 1;
  padding-bottom: 20px;
}
label {
  font-weight: 500;
}
.signup,
.signup a {
  text-align: right;
  font-size: 13px;
  padding-top: 10px;
  color: #7a7a7a;
  margin: 0;
}
.signup a {
  color: #2554FF;
}
.background {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  opacity: 0.2;
  position: fixed;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  z-index: 0;
}
.errorMessage {
  color: red;
}
.btn {
  background: #FF8633;
  color: white;
}
</style>
```

10. คำสั่งระบบลงทะเบียน

```
<template>
  <div class="vertical-center">
    <div class="inner-block">
      <div class="vue-tempalte">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        <div class="home">
          <InputForm
            ref="signUpRef"
            headerName="Sign Up"
            sumitButtonName="Sign Up"
            :items="formItems"
            @submit="submit"
          ></InputForm>
          <p class="signup text-right mt-2">
            <a href="" @click="loginWithGuest">Login as guest</a>
          </p>
          <LoadingIcon :showLoading="showLoading" ></LoadingIcon>
        </div>
      </div>
    </div>
  </div>
</template>
<script>
import UsersDataService from "../services/UsersDataService";
import LoadingIcon from '@/components/LoadingIcon.vue'
import InputForm from '@/components/InputForm.vue'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
export default {
  name: 'SignInView',
  components: {
    LoadingIcon,
    InputForm
  },
  data () {
    return {
      formItems: [
        { label: "Username", type: "text", name: "username", required: true },
        { label: "Password", type: "password", name: "password", required: true },
        { label: "Firstname", type: "text", name: "firstname", required: true },
        { label: "Lastname", type: "text", name: "lastname", required: true },
        { label: "Email Address", type: "email", name: "email", required: false },
      ],
      showLoading: false,
      checked: false
    }
  },
  methods: {
    submit(form) {
      this.showLoading = true
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
this.checked = false
UsersDataService.getByField('username', form.username).on('value', snapshot => {
  if (!this.checked) {
    if (snapshot.val()) {
      this.showLoading = false
      this.$refs.signUpRef.showErrorMessage(['This username already exists.'])
    } else {
      UsersDataService.create({
        firstname: form.firstname,
        lastname: form.lastname,
        email: form.email,
        username: form.username,
        password: form.password,
      })
      .then(response => {
        this.showLoading = false
        localStorage.setItem("id", response.key)
        this.$router.push('/detail')
      })
      .catch(err => {
        this.showLoading = false
        this.$refs.signUpRef.showErrorMessage([`Cannot connect to service (${err})`])
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            })
          }
          this.checked = true
        }
        UsersDataService.off('value')
      })
    },
    cancel() {
      this.$router.push('/')
    },
    loginWithGuest() {
      localStorage.setItem('id', 'guest')
      this.$router.push('/detail')
    }
  }
}
</script>
<style scoped>
.previos {
  float: left;
}
</style>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. คำสั่งการเข้าระบบ

```
<template>
  <div class="vertical-center">
    <div class="inner-block">
      <div class="vue-tempalte">
        <div class="home">
          <InputForm
          ref="signInRef"
          headerName="Sign In"
          sumitButtonName="Sign In"
          :items="formItems"
          @submit="submit"
          ></InputForm>
          <p class="signup text-right mt-2">
          Not yet a member? <router-link to="/signup">Sign Up</router-link>
          </p>
          <LoadingIcon :showLoading="showLoading" ></LoadingIcon>
        </div>
      </div>
    </div>
  </div>
</template>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<script>
import UsersDataService from "../services/UsersDataService";
import LoadingIcon from '@/components/LoadingIcon.vue'
import InputForm from '@/components/InputForm.vue'
export default {
  name: 'SignInView',
  components: {
    LoadingIcon,
    InputForm
  },
  data () {
    return {
      formItems: [
        { label: "Username", type: "text", name: "username", required: true },
        { label: "Password", type: "password", name: "password", required: true },
      ],
      showLoading: false,
      checked: false,
    }
  },
  methods: {
    submit(form) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
this.showLoading = true
this.checked = false
UsersDataService.getByField("username", form.username).on('value', snapshot => {
  this.showLoading = false
  if (!this.checked) {
    if (!snapshot.val()) {
      this.$refs.signInRef.showErrorMessage(['Invalid username or password.'])
    } else {
      let key = Object.keys(snapshot.val())[0]
      let password = snapshot.val()[key].password
      if (password !== form.password) {
        this.$refs.signInRef.showErrorMessage(['Invalid username or password.'])
      } else {
        localStorage.setItem("id", key)
        this.$router.push('/detail')
      }
    }
    this.checked = true
  }
  UsersDataService.off('value')
 })
},
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    cancel() {
      this.showLoading = true
      console.log('Press cancel')
    }
  }
}
</script>
```

12. คำสั่งหน้าการแสดงผล

```
<template>
 <div class="nav-header">
   <h3 class="text-left text-header">Alerting System for Hunchback to Prevent Office Syndrome</h3>
   <button class="btn dropdown-toggle btn-manage-user" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
       Account
   </button>
   <div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
       <a v-if="!guest" class="dropdown-item" @click="editUser">Edit</a>
       <a v-if="!guest" class="dropdown-item" @click="changePassword">Change Password</a>
       <a v-if="!guest" class="dropdown-item" @click="deleteUser">Delete</a>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      <a v-if="guest" class="dropdown-item" @click="signup">Sign Up</a>
      <a class="dropdown-item" @click="logout">Logout</a>
  </div>
</div>
<LoadingIcon :showLoading="showLoading" ></LoadingIcon>
<div class="space"></div>
<table class="table table-bordered table-hover table-striped">
  <thead class="">
    <tr>
      <th class="table-header" scope="col">No.</th>
      <th class="table-header" scope="col">ID</th>
      <th class="table-header" scope="col">Time</th>
      <th class="table-header" scope="col">Motor</th>
      <th class="table-header" scope="col">SIT</th>
      <th class="table-header" scope="col">Count SIT</th>
      <th class="table-header" scope="col">GYRO</th>
      <th class="table-header" scope="col">Count GYRO</th>
    </tr>
  </thead>
  <tbody v-for="v in items" :key="v.id">
    <tr>
      <th class="table-body" scope="row">{{v.item_no}}</th>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          <td class="table-body">{{v.key}}</td>
          <td class="table-body">{{v.time}}</td>
          <td class="table-body">{{v.motor}}</td>
          <td class="table-body">{{v.sit}}</td>
          <td class="table-body">{{v.count_sit}}</td>
          <td class="table-body">{{v.gyro}}</td>
          <td class="table-body">{{v.count_gyro}}</td>
        </tr>
      </tbody>
    </table>
    <nav>
      <ul class="pagination justify-content-end pagination-position">
        <span v-for="index in totalPage" :key="index">
          <li class="page-item"><a class="page-link" :class="{ 'item-active': page === index
}" @click="changePage(index)">{{index}}</a></li>
        </span>
      </ul>
    </nav>
    <div class="space"></div>
  </template>
  <script>
  import UsersDataService from "../services/UsersDataService"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
import LogDeviceDataService from "../services/LogDeviceDataService";
import LoadingIcon from '@/components/LoadingIcon.vue'
import Swal from 'sweetalert2'
export default {
  name: 'DetailView',
  components: {
    LoadingIcon,
  },
  async mounted() {
   let id = localStorage.getItem('id')
   if ([undefined, null, ""].includes(id)) {
       this.$router.push('/')
   }
    if (id === 'guest') this.guest = true
    if (!this.guest) await this.getUserById(id)
    this.showLoading = true
    LogDeviceDataService.getAll().on('value', this.onDataChange);
  },
  beforeUnmount() {
    LogDeviceDataService.off('value')
  },
  data () {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return {
      showLoading: false,
      contents: [],
      items: [],
      itemPerPage: 15,
      totalPage: 0,
      page: 1,
      guest: false
    }
  },
  methods: {
    getDate(dateTimeStr) {
      let timeStr = dateTimeStr.split(' ')[0]
      let dateStr = dateTimeStr.split(' ')[1]
      let hour = timeStr.split(':')[0].padStart(2, '0')
      let minute = timeStr.split(':')[1].padStart(2, '0')
      let second = timeStr.split(':')[2].padStart(2, '0')
      let date = dateStr.split('')[0].padStart(2, '0')
      let month = dateStr.split('-')[1].padStart(2, '0')
      let year = dateStr.split('-')[2]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      return new Date(`${year}-${month}-
${date}T${hour}:${minute}:${second}+0700`).getTime()
    },
    compareDate(a, b) {
      let aDateTimeStr = this.getDate(a.time)
      let bDateTimeStr = this.getDate(b.time)

      return bDateTimeStr - aDateTimeStr
    },
    onDataChange(items) {
      let _results = [];
      items.forEach(item => {
        let key = item.key
        let data = item.val()
        _results.push({
          key: key,
          ...data,
        })
      })
      _results.sort(this.compareDate)
      let index = 1
      for (let result of _results) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      result.item_no = index
      index++
    }
    this.contents = _results
    this.totalPage = Math.ceil(this.contents.length / this.itemPerPage)
    this.changePage(1)
    this.showLoading = false
  },
  async getUserById(id) {
    this.showLoading = true
    this.checked = false
    UsersDataService.getById(id).on('value', snapshot => {
      this.showLoading = false
      if (!this.checked) {
        if (!snapshot.val()) {
          this.$router.push('/')
        } else {
          let key = Object.keys(snapshot.val())[0]
          let body = snapshot.val()[key]
          this.id = id
          this.firstname = body.firstname
          this.lastname = body.lastname
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          this.email = body.email
          this.username = body.username
          this.password = body.password
        }
        this.checked = true
      }
      UsersDataService.off('value')
    })
  },
  changePage(page) {
    let start = this.itemPerPage * (page - 1)
    let end = this.itemPerPage * page
    this.items = this.contents.slice(start, end)
    this.page = page
  },
  editUser() {
    this.$router.push({ name: 'edit', params: { id: this.id }})
  },
  changePassword() {
    this.$router.push({ name: 'changePassword', params: { id: this.id }} )
  },
  deleteUser() {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Swal.fire({
  title: 'Are you sure?',
  text: 'Do you want delete user?',
  type: 'warning',
  showCancelButton: true,
  confirmButtonColor: '#FF8633',
  cancelButtonColor: '#FFC433',
  confirmButtonText: 'Ok',
  cancelButtonText: 'Cancel',
  allowOutsideClick: false
})
.then((result) => {
  if (result.isConfirmed) {
    this.showLoading = true

    UsersDataService.delete(this.id)
    .then(() => {
      this.loadingIcon = false
      localStorage.removeItem("id")
      this.$router.push('/')
    })
    .catch(err => {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        this.showLoading = false
        Swal.fire({
          title: `Cannot connect to service (${err}).`,
          type: 'error',
          confirmButtonColor: '#FF8633',
          confirmButtonText: 'Ok',
          allowOutsideClick: false
        })
      })
      this.showLoading = false
    }
  })

},
logout() {
  localStorage.removeItem("id")
  this.$router.push('/')
},
signup() {
  localStorage.removeItem("id")
  this.$router.push('/signup')
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
}
</script>
<style scoped>
.nav-header {
  position: relative;
  min-height: 50px;
  border: 1px solid transparent;
  background: #FF8633;
  box-shadow: 0 0.25rem 0.25rem rgb(0 0 0 / 25%), inset 0 -1px 5px rgb(0 0 0 / 25%);
}
.btn-manage-user {
  float: right;
  margin-top: -42px;
  margin-right: 20px;
  width: 12%;
  border-color: #FF8633;
  color: white;
  background-color: transparent;
  font-weight: 200;
  font-size: 1.2rem;
  text-align: right;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
.text-header {
  margin-left: 20px;
  margin-top: 5px;
  color: white;
  background-color: transparent;
  font-weight: 200;
}
.table {
  width: 70%;
  margin: auto;
  background-color: white;
}
.table-header {
  padding: 0.5em;
  background: #FFC433;
}
.table-body {
  padding: 0.25em;
  margin: 0px;
}
.space {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  padding: 2em;
}
.pagination-position {
  margin-top: 1em;
  width: 70%;
  margin-right: auto;
  margin-left: auto;
}
.item-active {
  background-color: #FFC433;
}
</style>
```

13. คำสั่งการหน้าการแก้ไขข้อมูล

```
<template>
  <div class="vertical-center">
    <div class="inner-block">
      <div class="vue-tempalte">
        <div class="home">
          <InputForm
            ref="editInfoRef"
            headerName="Edit Information"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            sumitButtonName="Save"
            cancelButtonName="Cancel"
            :items="formItems"
            @submit="submit"
            @cancel="cancel"
          ></InputForm>
          <LoadingIcon :showLoading="showLoading" ></LoadingIcon>
        </div>
      </div>
    </div>
  </div>
</template>
<script>
import UsersDataService from "../services/UsersDataService";
import LoadingIcon from '@/components/LoadingIcon.vue'
import InputForm from '@/components/InputForm.vue'
export default {
  name: 'SignInView',
  components: {
    LoadingIcon,
    InputForm
  },
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  async mounted() {
    let id = this.$route.params.id
    await this.getUserById(id)
  },
  data () {
    return {
      formItems: [
        { label: "Username", type: "text", name: "username", required: true, disabled: true
},
        { label: "Password", type: "password", name: "password", required: true, disabled:
true },
        { label: "Firstname", type: "text", name: "firstname", required: true },
        { label: "Lastname", type: "text", name: "lastname", required: true },
        { label: "Email Address", type: "email", name: "email", required: false },
      ],
      showLoading: false,
      checked: false,
      id: '',
      firstname: '',
      lastname: '',
      email: '',
      username: '',
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    password: '',
  }
},
methods: {
  async getUserById(id) {
    this.showLoading = true
    this.checked = false
    UsersDataService.getById(id).on('value', snapshot => {
     this.showLoading = false
     if (!this.checked) {
      if (!snapshot.val()) {
        this.$router.push('/')
      } else {
        let key = Object.keys(snapshot.val())[0]
        let body = snapshot.val()[key]
        this.$refs.editInfoRef.updateValue({ ...body })
        this.id = id
        this.firstname = body.firstname
        this.lastname = body.lastname
        this.email = body.email
        this.username = body.username
        this.password = body.password
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        this.checked = true
      }
      UsersDataService.off('value')
    })
  },
  submit(form) {
    this.showLoading = true
    UsersDataService.update(this.id, {
      firstname: form.firstname,
      lastname: form.lastname,
      email: form.email,
      username: form.username,
      password: form.password,
    })
    .then(() => {
      this.showLoading = false
      this.$router.push('/detail')
    })
    .catch(err => {
      this.showLoading = false
      this.$refs.editInfoRef.showErrorMessage([`Cannot connect to service (${err})`])
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      })
    },
    cancel() {
      this.$router.push('/detail')
    }
  }
}
</script>
```

14. คำสั่งหน้าการเปลี่ยนรหัสข้อมูล

```
<template>
  <div class="vertical-center">
    <div class="inner-block">
      <div class="vue-tempalte">
        <div class="home">
          <InputForm
            ref="changePasswordRef"
            headerName="Change Password"
            sumitButtonName="Save"
            cancelButtonName="Cancel"
            :items="formItems"
            @submit="submit"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            @cancel="cancel"
          ></InputForm>
          <LoadingIcon :showLoading="showLoading" ></LoadingIcon>
        </div>
      </div>
    </div>
  </div>
</template>
<script>
import UsersDataService from "../services/UsersDataService";
import LoadingIcon from '@/components/LoadingIcon.vue'
import InputForm from '@/components/InputForm.vue'
export default {
  name: 'SignInView',
  components: {
    LoadingIcon,
    InputForm
  },
  async mounted() {
    let id = this.$route.params.id
    await this.getUserById(id)
  },
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
data () {
  return {
    formItems: [
      { label: "New Password", type: "password", name: "password", required: true },
    ],
    showLoading: false,
    checked: false,
    id: '',
    firstname: '',
    lastname: '',
    email: '',
    username: '',
    password: '',
  }
},
methods: {
  async getUserById(id) {
    this.showLoading = true
    this.checked = false

    UsersDataService.getById(id).on('value', snapshot => {
      this.showLoading = false
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (!this.checked) {
      if (!snapshot.val()) {
          this.$router.push('/detail')
      } else {
        let key = Object.keys(snapshot.val())[0]
        let body = snapshot.val()[key]
        this.id = id
        this.firstname = body.firstname
        this.lastname = body.lastname
        this.email = body.email
        this.username = body.username
        this.password = body.password
      }
      this.checked = true
    }
    UsersDataService.off('value')
  })
},
submit(form) {
  this.showLoading = true
  UsersDataService.update(this.id, {
    firstname: this.firstname,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        lastname: this.lastname,
        email: this.email,
        username: this.username,
        password: form.password,
      })
      .then(() => {
        this.showLoading = false
        this.$router.push('/detail')
      })
      .catch(err => {
        this.showLoading = false
        this.$refs.changePasswordRef.showErrorMessage([`Cannot connect to service
(${err}`])
      })
    },
    cancel() {
      this.$router.push('/detail')
    }
  }
}
</script>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้