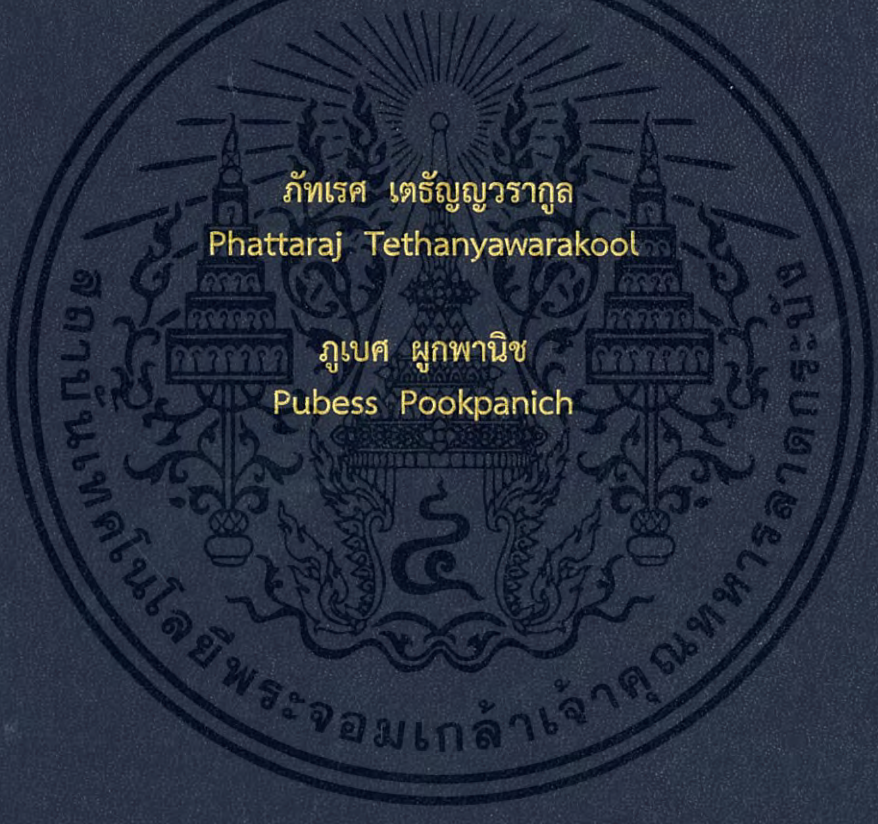
เครื่องวัดความเร่งแบบ 3 แกนไร้สาย
Wireless IMU

ภัทเรศ เตธัญญวรากูล
Phattaraj Tethanyawarakool

ภูเบศ ผูกพานิช
Pubess Pookpanich



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

เครื่องวัดความเร่งแบบ 3 แกนไร้สาย
Wireless IMU

นายภัทเรศ เตธัญญวรากูล
Phattaraj Tethanyawarakool

นายภูเบศ ผูกพานิช
Pubess Pookpanich

อาจารย์ที่ปรึกษา
ดร.สุรเดช ตรีไตรลักษณะ

ปริญญา
วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ **ปีการศึกษา 2560**

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องวัดความเร่ง 3 แกนแบบไร้สาย

Wireless IMU

ผู้จัดทำ นาย ภัทเรศ เตธัญญวรากูล รหัสประจำตัว 57010968

นาย ภูเบศ ผูกพานิช รหัสประจำตัว 57010994

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

(ดร.สุรเดช ตรีไตรลักษณะ)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน	เครื่องวัดความเร่ง 3 แกนแบบไร้สาย		
นักศึกษา	นายภัทเรศ	เตธัญญวรากูล	รหัสประจำตัว 57010968
	นายภูเบศ	ผูกพานิช	รหัสประจำตัว 57010994
ปริญญา	วิศวกรรมศาสตรบัณฑิต		
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์		
ปีการศึกษา	2560		
อาจารย์ที่ปรึกษาโครงงาน	ดร.สุรเดช ตรีไตรลักษณะ		

บทคัดย่อ

วิทยานิพนธ์นี้เป็นการออกแบบสร้างเครื่องวัดความเร่ง 3 แกนแบบไร้สายเพื่อนำไปใช้ประโยชน์ในการศึกษาพฤติกรรมของมนุษย์ในการใช้ชีวิตประจำวัน ในการใช้งานจะติดเครื่องวัดความเร่งบนร่างกายภายนอกเพื่อวัดความเร่งสามแกนของผู้ใช้รวมทั้งวัดความเร็วของการหมุนรอบแกนและทิศทาง ข้อมูลที่วัดได้จะนำมาประเมินเป็นพฤติกรรมเช่นการยืน เดิน นั่ง หรือ นอน เครื่องวัดความเร่ง 3 แกนนี้ใช้ชุดเซ็นเซอร์ MPU6050 6dof IMU ซึ่งจะมีเซ็นเซอร์ในตัว อยู่ 2 อย่าง คือ ADXL345 เป็นตัวตัววัดความเร่ง ITG3205 เป็นตัววัดการหมุนรอบแกน วิทยานิพนธ์นี้ใช้บอร์ด NodeMCU-32s ในการสื่อสารกับเซ็นเซอร์ MPU-6050 6dof IMU แล้วบันทึกผ่าน SD Card เพื่อนำมาประมวลผลในโปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Wireless IMU	
Student	Mr. Phattaraj Tethanyawarakool	Student ID 57010968
	Mr. Pubess Pookpanich	Student ID 57010994
Degree	Bachelor of Engineering	
Program	Electronics Engineering	
Year	2017	
Project advisor	Dr. Suradej Tretriluxana	

ABSTRACT

This thesis is concerned implementation of an initial motion unit for objective to study of human behavior. By attaching the device on the body, acceleration, rotation and also direction of movement are indicated. The data will be then analyzed to indicate continuously the behavior of living such as standing, sitting or lying down. A MPU6050 6dof IMU sensor of which consists three built-in sensors, ADXL345 an accelerometer, ITG3205 gyroscope is used in the thesis. An NodeMCU-32s board is used for transferring the data from the MPU-6050 6dof IMU to Save via SD Card. Firstly we use the program MATLAB as the data monitoring on the PC.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ดร.สุรเดช ตรีไตรลักษณะ ที่สนับสนุนอุปกรณ์และสถานที่ในการทดลอง ให้คำปรึกษา ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประการณ์ที่ดีแก่กลุ่มข้าพเจ้า

ขอขอบคุณ ครอบครัว เตธัญญวรากูล และ ครอบครัว ผูกพานิช สำหรับการสนับสนุนในทุก ๆด้าน และสุดท้ายขอขอบคุณพี่และเพื่อนในห้องปฏิบัติการทุกคนที่คอยให้ความช่วยเหลือและให้กำลังใจสำหรับคุณงามความดีอันใดที่เกิดจากโครงงานนี้ กลุ่มข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

นายภัทเรศ เตธัญญวรากูล
นายภูเบศ ผูกพานิช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....VI
สารบัญรูป.....VII
บทที่ 1 บทนำ.....1
1.1 ความเป็นมาและความสำคัญของปัญหา.....1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....1
1.3 สมมุติฐานของการศึกษา.....1
1.4 ขอบเขตโครงงาน.....1
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....1
1.6 โครงสร้างของรายงาน.....2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....3
2.1 โรคนอนมากเกินไป Hypersomnia.....3
2.2.1 โทษพิษภัยจากการนอนเกิน.....3
2.2.1 สาเหตุ.....4
2.2 ออฟฟิศซินโดรม (office syndrome).....4
2.2.1 อาการทางระบบกระดูกและกล้ามเนื้อ.....4
2.2.2 สาเหตุของการเกิดออฟฟิศซินโดรม.....5
2.2.3 การสังเกตอาการออฟฟิศซินโดรมและแนวทางแก้ไข.....5
2.2.4 แนวทางการรักษากลุ่มอาการออฟฟิศซินโดรม.....6
2.2.5 ความสำคัญของกายภาพบำบัดต่อการรักษากลุ่มอาการออฟฟิศซินโดรม.....6
2.2.6 เป้าหมายในการดูแลทางกายภาพบำบัดในผู้ป่วยกลุ่มอาการออฟฟิศซินโดรม.....7
2.3 โรคเส้นเลือดขอด (Varicose Veins).....7
2.3.1 ลักษณะอาการของ เส้นเลือดขอด ที่พบได้ทั่วไป.....8
2.3.2 สาเหตุที่ทำให้เกิดเส้นเลือดขอด.....8
2.3.2 วิธีรักษา และการป้องกันอย่างถูกต้อง.....9
2.4 การติดต่อสื่อสารแบบ I2C (Inter-Intergrated Circuit).....9
2.5 การติดต่อสื่อสารแบบ SPI (Serial Peripheral Interface).....10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การคำนวณและการออกแบบเครื่องวัดความเร่ง....12
3.1ระบบตรวจวัดสัญญาณความเร่งและความเร็วเชิงมุมจากการเคลื่อนไหว....12
3.1.1 เซ็นเซอร์วัดความเร่งและความเร็วเชิงมุม....13
3.1.2 บอร์ด NodeMCU-32s....14
3.1.3 โมดูล MicroSD Card....15
3.1.4 Real Time Clock DS3231....16
บทที่ 4 วิธีการทดสอบ และผลการทดสอบเครื่องจ่ายไฟตรงสองทางออก....18
4.1 การทดสอบอายุการใช้งานของแบตเตอรี่ขนาด AAA 4 ก้อน....18
4.2 ผลการทดลอง....19
บทที่ 5 วิเคราะห์และสรุปผลการทดสอบคุณสมบัติของวงจร....24
5.1 สรุปผลการทดลอง....24
5.2 ปัญหาและอุปสรรคในการทำงาน....24
5.3 แนวทางการแก้ไข....24
5.4 ประโยชน์และการประยุกต์การใช้งาน....24
5.5 สิ่งที่ได้จากโครงงาน....24
เอกสารอ้างอิง....25
ภาคผนวก....26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

3.1 แสดงรายละเอียดการเชื่อมต่อเซ็นเซอร์วัดความเร่ง กับ บอร์ด NodeMCU-32s 13

3.2 แสดงรายละเอียดการเชื่อมต่อขา SD Card Module กับ บอร์ด NodeMCU-32s 15

3.3 แสดงรายละเอียดการเชื่อมต่อโมดูล Real Time Clock DS3231 กับ บอร์ด NodeMCU-32s 16

4.1 แสดงอายุการใช้งานที่เวลาเป็นหน่วยนาทีต่อแรงดันที่เป็นหน่วยโวลต์ 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า
2.1 แสดงลักษาณะการนั่งที่ถูกต้องและลักษาณะการนั่งที่ไม่ถูกต้อง 4
2.2 รูปแบบสถานะในการรับ-ส่งข้อมูล แบบ i2c 9
2.3 รูปแบบสถานะในการรับ-ส่งข้อมูล แบบ i2c 9
2.4 การเชื่อมต่อการสื่อสารแบบ SPI ระหว่างอุปกรณ์ Master – Slave 10
2.5 การเชื่อมต่อการสื่อสารแบบ SPI ระหว่างอุปกรณ์ Master – Slave หลายตัว 10
3.1 ระบบตรวจวัดสัญญาณความเร่งและความเร็วเชิงมุมจากการเคลื่อนไหว 12
3.2 เซ็นเซอร์วัดความเร่ง (MPU-6050) 13
3.3 บอร์ด NodeMCU-32s 14
3.4 ขาการเชื่อมต่อของบอร์ด NodeMCU-32s 14
3.5 โมดูล MicroSD Card 15
3.6 โมดูล Real Time Clock DS3231 16
3.7 เครื่องวัดความเร่ง 3 แกนแบบไร้สาย 17
4.1 กราฟแสดงเวลา-โวลต์ 18
4.2 สัญญาณ Accelerometer ของการเคลื่อนไหวโดยการเดินลงบันได 19
4.3 สัญญาณ Gyroscope ของการเคลื่อนไหวโดยการเดินลงบันได 19
4.4 สัญญาณ Accelerometer ของการเคลื่อนไหวโดยการเดินขึ้นบันได 20
4.5 สัญญาณ Gyroscope ของการเคลื่อนไหวโดยการเดินขึ้นบันได 20
4.6 สัญญาณ Accelerometer ของการเคลื่อนไหวโดยการนั่งเก้าอี้และลุกจากเก้าอี้ 21
4.7 สัญญาณ Gyroscope ของการเคลื่อนไหวโดยการนั่งเก้าอี้และลุกจากเก้าอี้ 21
4.8 สัญญาณ Accelerometer ของการเคลื่อนไหวโดยการเดิน 22
4.9 สัญญาณ Gyroscope ของการเคลื่อนไหวโดยการเดิน 22
4.10 ชิ้นงาน 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การศึกษาพฤติกรรมการใช้ชีวิตของมนุษย์เพื่อใช้ประเมินทางการแพทย์ สามารถใช้กล้องวิดีโอบันทึกพฤติกรรมอย่างต่อเนื่อง อย่างไรก็ดีการใช้กล้องฯมีข้อจำกัดด้านการติดตามผู้ป่วยอย่างต่อเนื่องในพื้นที่กว้าง ๆ ดังนั้นโครงงานนี้เป็นการสร้างเครื่องติดตามตัวเพื่อวัดความเร่งการเคลื่อนที่ของอวัยวะบ้างจุดเพื่อนำมาใช้ในการประเมินพฤติกรรมเช่นการ นั่ง ยืน หรือนอน โดยใช้เซ็นเซอร์ความเร่ง 3 แกนแบบไร้สายบันทึกข้อมูลเป็นดิจิตอลเพื่อใช้ในการประเมิน

1.2 ความมุ่งหมายและวัตถุประสงค์ของการวิจัย

1.สามารถนำความรู้ที่ได้จากภาคทฤษฎีของวิชาต่าง ๆ ที่ได้ศึกษามาปฏิบัติและประยุกต์ใช้เพื่อสร้างชิ้นงานขึ้นมาและสามารถใช้งานได้จริง

2.ศึกษาความรู้เพิ่มเติมในเรื่องต่าง ๆที่เกี่ยวข้องกับการทำโครงการ เช่น ศึกษาการทำงานของ เซนเซอร์ MPU-6050 และ Bluetooth รวมทั้ง บอร์ด NodeMCU-32s เป็นต้น

1.3 สมมุติฐานของการวิจัย

ข้อมูลการบันทึกความเร่งแบบ 3 แกน บางอวัยวะของร่างกายสามารถนำมาประเมินเป็นพฤติกรรมของมนุษย์เช่นนั่ง ยืน นอน ได้

1.4 ขอบเขตของการวิจัย

ขอบเขตการศึกษาเกี่ยวกับ เครื่องวัดความเร่ง 3 แกนแบบไร้สาย สามารถตรวจพฤติกรรมของมนุษย์ว่าผู้ทดลองกำลังนั่ง ยืน นอน ได้ที่ถูกต้องและแม่นยำและสามารถเก็บค่าผลการทดลองได้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

สามารถนำความรู้ต่าง ๆที่ได้ศึกษามานั้น ไปใช้ในการคิดและออกแบบการทำงานต่าง ๆ และมีความรู้ความเข้าใจในการทำงานของบอร์ด NodeMCU-32s มากขึ้นและสามารถแก้ไขเวลาโครงงานมีปัญหาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 โครงสร้างของวิทยานิพนธ์

รายงานฉบับนี้เป็นผลจากการศึกษาค้นคว้าและทดลองตลอดภาคการศึกษาแล้วนำเสนอเป็นบทตอนดังนี้

บทที่ 1 กล่าวถึง บทนำ ความเป็นมาของโครงงาน และวัตถุประสงค์

บทที่ 2 กล่าวถึง หลักการและทฤษฎีต่าง

บทที่ 3 กล่าวถึง การออกแบบโปรแกรม การออกแบบวงจร คุณสมบัติของวงจร และการเลือกใช้อุปกรณ์

บทที่ 4 กล่าวถึง การทดสอบคุณสมบัติ การทดลอง และผลการทดลอง

บทที่ 5 กล่าวถึง วิเคราะห์และสรุปผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 โรคนอนมากเกินไป Hypersomnia

ปัญหาเรื่องการนอนนั้นมีทั้งการนอนไม่เพียงพอ หรือ การนอนมากเกินไป วันนี้เราจะทำความรู้จักเกี่ยวกับการนอนมากเกินไป (Hypersomnia) ซึ่งเป็นสิ่งที่ผิดปกติ ไม่ว่าจะเป็นเด็กหรือผู้ใหญ่ควรรีบมาพบแพทย์ คนที่มีภาวะนี้จะมีลักษณะที่ต้องการการนอนเพิ่มขึ้นทั้งในช่วงกลางวันและกลางคืน มีภาวะตื่นยากมากจากการนอน เมื่อตื่นแล้วก็รู้สึกว่าต้องการนอนต่อไปอีก ระหว่างวันก็ต้องการงีบหลับหลายๆ ครั้ง ในบางรายอาการงีบก็อาจเกิดขึ้นในเวลาที่ไม่เหมาะสม เช่น ขณะคุยกันก็ยังหลับได้ระหว่างกินอาหาร เป็นต้น ซึ่งหากพบมีอาการหรือมีคนใกล้ตัวที่มีอาการแบบนี้แนะนำควรรีบนำมาพบแพทย์เนื่องจากอาจส่งผลกระทบต่าง ๆ ตามมาได้

2.2.1 โทษพิษภัยจากการนอนเกิน

สมองเฉื่อยชา สมองล้า กลายเป็นคนไร้ชีวิตชีวา การทำงานของกระดูกกล้ามเนื้อและข้อลดประสิทธิภาพลงหากไม่ได้มีการเคลื่อนไหวนาน ๆ อาจมีผลทำให้กระดูกหรือกล้ามเนื้อไม่แข็งแรง มีปัญหาข้อต่าง ๆ เคลื่อนไหวได้ยากตามมาได้ น้ำหนักเกินมากกว่าเดิม เนื่องจากระบบการเผาผลาญไขมันลดลง มีการสะสมไขมันเพิ่มขึ้น อันอาจนำไปสู่โรคต่าง ๆ อาทิเช่น เบาหวาน ความดันโลหิตสูง ไขมันในเลือดสูง เป็นต้น กลายเป็นคนซึมเศร้า การนอนมาก ๆ ทำให้ฮอร์โมนในร่างกายแปรปรวน โดยเฉพาะ “เคมีอารมณ์ สารความสุข” จำพวก “เซโรโทนิน (Serotonin)” และ “เอนดอร์ฟิน (Endorphin)” ลดต่ำลง

2.2.1 สาเหตุ

อดนอนมาเป็นเวลานาน และบ่อย ๆ จนร่างกายพักผ่อนไม่พอ ทำให้นอนเท่าไรก็ไม่รู้จักพอสักที นาฬิกาชีวภาพในร่างกายแปรปรวน ปรับเวลาผิด เช่น เดินทางข้ามประเทศที่ต่างช่วงเวลากันมาก ๆ ฮอร์โมนในร่างกายหรือสารเคมีในสมองไม่ปกติ ทำให้ร่างกายนอนมากผิดปกติ นอนกรน มีภาวะการณ์หยุดหายใจในช่วงหลับ ทำให้ร่างกายรับออกซิเจนไม่พอ สมองได้รับบาดเจ็บหรือโรคเกี่ยวกับทางสมองต่าง ๆ การรับประทานยาบางชนิด เช่น ยาแก้แพ้ ฯลฯ และหมอขอฝากเทคนิคเพื่อให้นอนหลับได้ดีอย่างมีประสิทธิภาพกันค่ะ พยายามตั้งเวลาเข้านอนไม่เกิน 4 ทุ่มเพื่อให้ร่างกายได้รับฮอร์โมนดี ๆ ต่าง ๆ ได้อย่างเต็มที่ซึ่งฮอร์โมนเหล่านี้จะช่วยให้หลับได้ดีรวมไปถึงทำให้ร่างกายมีการฟื้นฟูซ่อมแซมตัวเองได้อย่างมีประสิทธิภาพ กำหนดตารางเข้านอน และตื่นนอนเวลาเดียวกันทุก ๆ วัน ติดต่อกัน 28 วัน ร่างกายจะสร้างระบบนาฬิกาชีวิตของตัวเราเองขึ้นมาใหม่ จะตื่นได้เองอย่างสดชื่น จัดห้องนอนให้โปร่ง อากาศระบายได้ดี ร่างกายที่ได้รับออกซิเจนไม่เพียงพอจะทำให้สมองเซื่องซึม และง่วงนอนตลอดเวลา ออกกำลังกาย นั่งสมาธิ สร้างออกซิเจนในเลือดให้มากขึ้น และยังทำให้ร่างกายแอ็คทีฟด้วย งดอาหาร junk food น้ำอัดลม แป้งขัดขาว เบเกอรี่ ซึ่งทำให้น้ำตาลในเลือดไม่ปกติ ร่างกายคุมไม่ได้ ทำให้ง่วง เหนื่อย ตอนที่ระดับน้ำตาลตก ก่อนเข้านอน 4 ชั่วโมงควรงดอาหารหนักๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ออฟฟิศซินโดรม (office syndrome)

เป็นกลุ่มอาการที่เกิดขึ้นกับคนที่ทำงานในออฟฟิศ เนื่องจากลักษณะงานที่ต้องนั่งหน้าคอมพิวเตอร์หรือทำงานอย่างใดอย่างหนึ่งด้วยท่าทางซ้ำ ๆ ต่อเนื่องเป็นเวลานาน จนอาจส่งผลให้เกิดโรคและอาการผิดปกติในระบบต่าง ๆ ของร่างกาย ไม่ว่าจะเป็นระบบกระดูกและกล้ามเนื้อ ระบบการย่อยอาหารและการดูดซึม ระบบหัวใจและหลอดเลือด ระบบฮอร์โมน นัยน์ตาและการมองเห็น โดยในส่วนที่เกี่ยวข้องกับงานกายภาพบำบัดจะเป็นอาการที่เกิดขึ้นกับระบบกระดูกและกล้ามเนื้อเป็นส่วนใหญ่

2.2.1 อาการทางระบบกระดูกและกล้ามเนื้อ

ที่เกิดจากการทำงานที่พบได้บ่อยและสามารถดูแลโดยนักภาพภาพบำบัด กล้ามเนื้ออักเสบเรื้อรัง (myofascial pain syndrome) เอ็นรัดข้อมืออักเสบกดทับเส้นประสาท (carpal tunnel syndrome) ความผิดปกติของความตึงตัวของเส้นประสาท (nerve tension) กล้ามเนื้อบริเวณแขนท่อนล่างด้านนอกอักเสบ (tennis elbow) ปลอกหุ้มเอ็นกล้ามเนื้อบริเวณฐานนิ้วโป้งอักเสบ (De Quervain's tendonitis) นิ้วล็อก (trigger finger) เอ็นกล้ามเนื้ออักเสบ (tendinitis) ปวดหลังจากท่าทางผิดปกติ (postural back pain) หลังยึดติดในท่าแอ่น (back dysfunction)

2.2.2 สาเหตุของการเกิดออฟฟิศซินโดรม

รูปที่ 2.1 แสดงลักษณะการนั่งที่ถูกต้องและลักษณะการนั่งที่ไม่ถูกต้อง

- การนั่งหน้าคอมพิวเตอร์หรือทำงานอย่างใดอย่างหนึ่งด้วยท่าทางซ้ำ ๆ ต่อเนื่องเป็นเวลานานมากกว่า 6 ชั่วโมงต่อวัน
- ท่าทางในการทำงานที่ไม่เหมาะสม เช่น การนั่งหลังค่อม ท่าก้มหรือเงยคอมากเกินไป
- สภาพแวดล้อมหรืออุปกรณ์ในการทำงานไม่เหมาะสม
- สภาพร่างกายที่อาจส่งผลต่ออาการเจ็บป่วย เช่น ภาวะเครียดจากงาน การอดอาหาร การพักผ่อนไม่เพียงพอ ซึ่งส่งผลให้ร่างกายต้องแบกรับความตึงเครียดปราศจากการผ่อนคลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การสังเกตอาการออฟฟิศซินโดรมและแนวทางแก้ไข

ระดับของอาการ	การสังเกตอาการ	แนวทางแก้ไข
ระดับที่ 1	อาการเกิดขึ้น เมื่อทำงานไประยะหนึ่ง พักแล้วดีขึ้นทันที	• พักสลับทำงานเป็นระยะ ๆ • ยืดกล้ามเนื้อเพื่อผ่อนคลาย • นวดผ่อนคลาย • ออกกำลังกาย
ระดับที่ 2	อาการเกิดขึ้น พักนอนหลับแล้ว แต่ยังคงมีอาการอยู่	• ปรับเปลี่ยนพฤติกรรมการทำงาน • รับการรักษาที่ถูกต้อง
ระดับที่ 3	อาการปวดมากแม้ทำงานเพียงเบาๆ พักแล้วอาการก็ยังไม่ทุเลาลง	• พักงาน/ปรับเปลี่ยนงาน • รับการรักษาที่ถูกต้อง

2.2.4 แนวทางการรักษากลุ่มอาการออฟฟิศซินโดรม

- การรักษาด้วยยา

- การรักษาด้วยวิธีทางเวชศาสตร์ฟื้นฟูและการทำกายภาพบำบัดเพื่อยืดกล้ามเนื้อและปรับอิริยาบถให้ถูกต้อง

- การปรับสถานีงาน พื้นที่การทำงาน สภาพแวดล้อมในการทำงาน และลักษณะงานให้เหมาะสมกับแต่ละบุคคล

- การออกกำลังกายเพื่อเพิ่มสมรรถภาพร่างกายโดยรวม

- การรักษาด้วยศาสตร์ทางเลือกอื่น เช่น การฝังเข็ม การนวดแผนไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 ความสำคัญของกายภาพบำบัดต่อการรักษากลุ่มอาการออฟฟิศซินโดรม

กายภาพบำบัดมีบทบาทสำคัญในการรักษากลุ่มอาการออฟฟิศซินโดรมที่เกี่ยวข้องกับระบบกล้ามเนื้อและกระดูกตั้งแต่ระยะเริ่มแรกจนถึงกรณีที่มีอาการมากจนรบกวนชีวิตประจำวันหรือก่อให้เกิดความบกพร่องทางการเคลื่อนไหว เพื่อฟื้นฟูให้ผู้ป่วยสามารถกลับมาใช้ชีวิตประจำวันได้ตามปกติ
นักกายภาพบำบัดยังมีหน้าที่ในการเสริมสร้างความรู้และความเข้าใจเพื่อป้องกันไม่ให้ผู้ป่วยกลับมามีอาการเหล่านี้อีก โดยนักกายภาพบำบัดจะประเมินโครงสร้างร่างกายพร้อมปรับแก้โครงสร้างร่างกายให้เกิดความสมดุลและปกติ รวมถึงให้คำแนะนำในการปรับเปลี่ยนท่าทางระหว่างการทำงาน การปรับสภาพแวดล้อมของเครื่องมือและสิ่งแวดล้อมในที่ทำงานให้เหมาะสมในแต่ละบุคคล แนะนำการยืดกล้ามเนื้อเพื่อผ่อนคลายในระหว่างการทำงาน รวมทั้งส่งเสริมให้มีการออกกำลังกายอย่างต่อเนื่องเพื่อให้กล้ามเนื้อแข็งแรง พร้อมรับสภาวะการทำงานที่อาจไม่สามารถปรับเปลี่ยนได้ รวมถึงช่วยลดความเสี่ยงของโรคอื่น ๆ ที่เกี่ยวเนื่องกับออฟฟิศซินโดรมด้วย

2.2.6 เป้าหมายในการดูแลทางกายภาพบำบัดในผู้ป่วยกลุ่มอาการออฟฟิศซินโดรม

- ลดอาการปวดที่เกิดจากการอักเสบของกล้ามเนื้อ/เอ็นกล้ามเนื้อ
- ลดการเคลื่อนไหวของข้อต่อนั้น ๆ (immobilization) ด้วยอุปกรณ์ช่วยพยุงต่าง ๆ รวมถึงถ้ายังจำเป็นต้องใช้งานต่อเนื่องจากภาระงาน
- ให้คำแนะนำที่ถูกต้องเพื่อลด/หลีกเลี่ยงการใช้งานบริเวณที่มีการอักเสบ
- ให้ความรู้ในการแก้ไขปัจจัยอื่นที่กระตุ้นให้เกิดอาการ ได้แก่
    - การปรับท่าทางให้ถูกต้อง
    - การปรับหรือแก้ไขความผิดปกติของโครงสร้างร่างกาย
    - การลดการกดทับของกล้ามเนื้อหรือการหดสั้นของกล้ามเนื้อ
    - การปรับสภาพแวดล้อมของเครื่องมือและสิ่งแวดล้อมในที่ทำงานให้เหมาะสมในแต่ละบุคคล
    - การออกกำลังกายเพื่อการป้องกันและส่งเสริมการทำงานของกล้ามเนื้อ
        - การยืดกล้ามเนื้อ
        - การออกกำลังกายเพื่อเพิ่มความแข็งแรงของกล้ามเนื้อ
        - การออกกำลังกายเพื่อเพิ่มความทนทานของกล้ามเนื้อและระบบหัวใจและหลอดเลือด
        - การออกกำลังกายเพื่อการผ่อนคลาย
        - การออกกำลังกายเพื่อปรับการทรงท่า (postural correction)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ดี การรักษากลุ่มอาการออฟฟิศซินโดรมด้วยวิธีต่าง ๆ นั้นเป็นเพียงการรักษาที่ปลายเหตุ เพื่อรักษาอาการกล้ามเนื้ออักเสบหรือรักษาพังผืดในกล้ามเนื้อ วิธีการที่ดีที่สุดที่จะป้องกันอาการจากออฟฟิศซินโดรม จำเป็นต้องทำร่วมกับการปรับเปลี่ยนพฤติกรรม การปรับอิริยาบถในการทำงานให้ถูกต้อง รวมถึงการเสริมสร้างสุขภาพกายและใจให้สมบูรณ์แข็งแรง พักผ่อนให้เพียงพอ ลดความเครียด และออกกำลังกายอย่างสม่ำเสมอ ซึ่งจะช่วยลดการเกิดอาการบาดเจ็บต่าง ๆ ที่จะมาบั่นทอนคุณภาพชีวิตและประสิทธิภาพในการทำงานได้อย่างถาวร

2.3 โรคเส้นเลือดขอด (Varicose Veins)

คือภาวะที่เกิดจากความผิดปกติของหลอดเลือดดำ หากเรากล่าวถึงโรคนี้ ก็จะนึกถึงอาการที่เกิดขึ้นที่หลอดเลือดดำบริเวณขา ซึ่งจะเป็นส่วนของลิ้นเปิดปิดที่ส่งเลือดจากขาขึ้นไปยังหลอดเลือดดำใหญ่ที่ท้อ ความผิดปกติชนิดนี้เป็นตัวการที่ทำให้เลือดไม่สามารถลำเลียงเข้าสู่หัวใจได้ทั้งหมด เกิดการย้อนกลับมาของเลือดดำคั่งค้างอยู่ในหลอดเลือดที่มีความตื้น จากนั้นก็จะโป่งขยายตัวออก จนกลายเป็นภาวะหลอดหลอดขอด

2.3.1 ลักษณะอาการของ เส้นเลือดขอด ที่พบได้ทั่วไป ได้แก่

- ส่วนมากจะไม่แสดงอาการให้เห็น ผู้ป่วยจะเข้ามาพบแพทย์ต่อเมื่อมองเห็นความผิดปกติจากภายนอก โดยเฉพาะในกลุ่มของคนที่รักสวยรักงาม เพราะเรียวขาจะดูไม่สวย มองเห็นเป็นเส้นเลือดสีดำปูดโปนในผิวหนังชัดเจน ซึ่งก็คือหลอดเลือดดำส่วนตื้น
- อาการเริ่มต้นมักจะปรากฏที่น่อง โดยไม่มีอาการใด ๆ เลย แต่บางรายอาจรู้สึกเมื่อยล้า ปวดกล้ามเนื้อ

รู้สึกเมื่อยเมื่อยืนเป็นเวลานาน และอาการจะเป็นมากขึ้นในช่วงสายๆ อาการจะดีขึ้นต่อเมื่อได้นอนราบหรือยกขาให้สูงกว่าศีรษะ

- มองเห็นเส้นเลือดบริเวณขา ซึ่งเป็นหลอดเลือดขนาดเล็กประมาณ 0-5 มิลลิเมตร โยงตัวกันเหมือนใย

แมงมุม เป็นสีม่วงหรือแดง ซึ่งจะสังเกตเห็นตำแหน่งของเส้นเลือดขอดที่มีขนาดใหญ่มากกว่า 1 มิลลิเมตร และเป็นสีเขียวจัด เห็นชัดเจนใต้ผิว ถือว่าเป็นระยะเริ่มต้น

- ผู้ป่วยที่มีอาการเส้นเลือดขอดมานานแล้ว จะพบอันตรายที่เรียกว่าเป็นภาวะแทรกซ้อน คือการอักเสบของหลอดเลือดที่ขอด หรือการอุดตันจนเลือดไม่สามารถผ่านเข้าไปได้ หรือหากเกิดอุบัติเหตุเส้นเลือดบริเวณดังกล่าวฉีกขาด ก็จะทำให้เสียเลือดมาก เพราะอยู่ในตำแหน่งของผิวหนังที่ค่อนข้างบาง เส้นเลือดขอดที่ถูกปล่อยทิ้งไว้นานโดยไม่ทำการรักษา แม้จะพบภาวะแทรกซ้อนได้น้อย แต่สามารถทำให้เกิดอันตรายได้ โดยเฉพาะการแตกตัวของเส้นเลือด การตกเลือดที่อาจถึงขั้นเสียชีวิตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากได้รับการรักษาช้าเกินไป ซึ่งภาวะดังกล่าวมักจะเป็นอาการที่เกิดขึ้นกับเส้นเลือดขอดที่ส่วนอื่นของร่างกาย แต่หากเป็นที่ขา อาการจะไม่น่าวิตกกังวลมากนัก นอกจากอาการอักเสบ เจ็บปวดจนส่งผลกระทบต่อการดำเนินชีวิตประจำวัน เกิดแผลแตกเรื้อรังและเป็นแผลเป็นเพราะความดันผิวหนังที่เพิ่มสูงมากขึ้น

2.3.2 สาเหตุที่ทำให้เกิดเส้นเลือดขอด

- อายุเป็นปัจจัยสำคัญที่ทำให้เกิดความเสี่ยงได้สูงถึง 70 เปอร์เซ็นต์ และพบได้มากในผู้สูงอายุตั้งแต่ 70 ปีขึ้นไป

- ประวัติครอบครัวที่เคยป่วยเป็นโรคเส้นเลือดขอด จะทำให้ลูกหลานมีโอกาสเป็นโรคนี้ได้สูงถึง 2 เท่า

- คนที่ต้องทำงานด้วยการยืนหรือนั่งติดต่อกันเป็นเวลานาน ก็ย่อมเสี่ยงที่จะทำให้เป็นเส้นเลือดขอดได้

- อาหารบางชนิดเป็นตัวกระตุ้นทำให้เกิดความผิดปกติของหลอดเลือดดำ และกลายเป็นปัจจัยทำให้เกิดโรคเส้นเลือดขอดได้เช่นกัน

- ผู้ที่อยู่ในกลุ่มคนที่มีน้ำหนักตัวเกินมาตรฐานจะเสี่ยงต่อการเกิดโรคหลอดเลือดขอดที่ขา

- ความผิดปกติของลิ้นที่ทำหน้าที่เป็นวาล์วเปิดปิดหลอดเลือดดำเสื่อมสภาพหรือทำงานไม่ได้เป็นตัวการทำให้เลือดดำไหลย้อนกลับมาคั่งค้าง สะสมจนโป่งพอง

- การอุดตันของเส้นเลือดที่เกิดจากโรคมะเร็ง

- อุบัติเหตุที่ขาจนเกิดผลแทรกซ้อนต่อการทำงานของเส้นเลือดดำ

2.3.2 วิธีรักษา และการป้องกันอย่างถูกต้อง

แพทย์จะทำการตรวจวินิจฉัย อาการของเส้นเลือดขอดที่เกิดขึ้นก่อน ตั้งแต่การซักประวัติ ตรวจร่างกาย การตรวจทางรังสีวิทยาด้วย Ultrasound กรณีที่แพทย์สงสัยว่าจะพบปัญหาเส้นเลือดดำขอดในส่วนลึก คนที่เคยมีอาการขาบวมหรือแผลที่เท้ามาก่อน ก็จะทำการตรวจเช็คอย่างละเอียด เพราะบางรายก็มีปัญหาเส้นเลือดขอดได้ตั้งแต่แรกเกิด เรียกกันว่าภาวะ "Klippel-Trenaunay syndrome"

- การรักษา ในการรักษาหลอดเลือดขอด ทางการแพทย์ถูกแบ่งออกเป็นหลายวิธีด้วยกัน ซึ่งมีตั้งแต่การใช้ผ้าพันยืด การฉีดยาเข้าสู่หลอดเลือดที่ขอด ไปจนถึงขั้นตอนการผ่าตัด ขึ้นอยู่กับอาการความรุนแรงของโรค ที่จะต้องอยู่ในดุลยพินิจของแพทย์ที่ทำการรักษา

- การป้องกันตัวเองให้ปลอดภัยจากโรคเส้นเลือดขอด แม้จะพบอันตรายเพียงส่วนน้อย แต่หากเกิดขึ้นกับตนเอง ย่อมเป็นความเสี่ยงที่มีความน่ากลัว ดังนั้น จึงควรหลีกเลี่ยงพฤติกรรมที่จะทำให้เลือดคั่งค้างนั่นก็คือ เลี่ยงการเดินและเคลื่อนไหวร่างกายนาน ๆ หลีกเลี่ยงการนั่งหรือยืนเป็นเวลานาน ๆ หากจำเป็นให้ใช้วิธีออกกำลังกายด้วยการเขย่งเท้าอยู่กับที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การติดต่อสื่อสารแบบ I2C หรือ Inter-Intergrated Circuit

คือ เป็นการสื่อสารแบบอนุกรมแบบ Synchronous ด้วยสายสัญญาณเพียง 2 เส้น คือ สายสัญญาณข้อมูล SDA (Serial Data Line) และสายสัญญาณนาฬิกา SCL (Serial Clock Line)

รูปแบบสถานะในการรับ-ส่งข้อมูล

รูปที่ 2.2 รูปแบบสถานะในการรับ-ส่งข้อมูล แบบ i2c

Start เป็นสถานะที่บอกเริ่มต้นการรับ-ส่งข้อมูล โดยการเปลี่ยนสัญญาณของ SDA จาก High ไปเป็น Low โดยที่ SCL ยังคงเป็น High อยู่

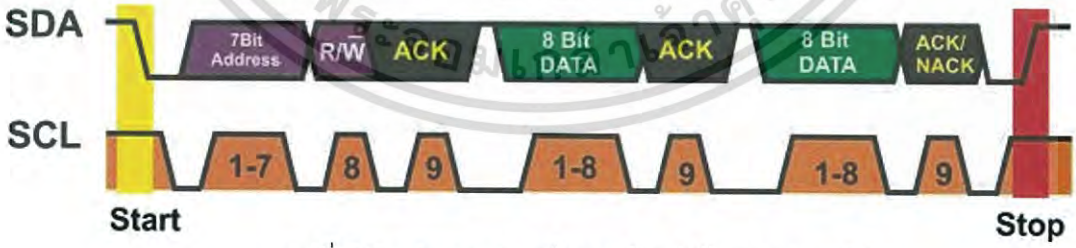
Control Byte ประกอบไปด้วย 3 ส่วน คือ

- ID ของอุปกรณ์จำนวน 4 บิต ซึ่งถูกกำหนดมาโดยผู้ผลิต IC หรืออุปกรณ์ I2C
- Device Address ขนาด 3 บิต สามารถกำหนดได้เองจากการจ่าย Logic หรือต่อขาให้กับ IC
- Mode ขนาด 1 บิต ใช้กำหนดว่าเป็นการ Read หรือ Write Data กับอุปกรณ์ IC

ACK หรือ Acknowledge เป็นบิตที่ใช้บอกว่า IC มีการตอบสนองต่อคำสั่งที่ได้รับมาแล้ว

DATA คือ ข้อมูลที่ต้องการเขียนหรืออ่านออกมาจาก IC ขึ้นกับ Mode ที่เราได้ตั้งค่าเอาไว้

STOP เป็นสถานะที่บอกให้อุปกรณ์รู้ว่าสิ้นสุดการรับส่งข้อมูลแล้ว โดย SDA จะเปลี่ยนจาก Low เป็น High ในขณะที่ SCL ยังเป็น High อยู่



รูปที่ 2.3 รูปแบบสถานะในการรับ-ส่งข้อมูล แบบ I2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

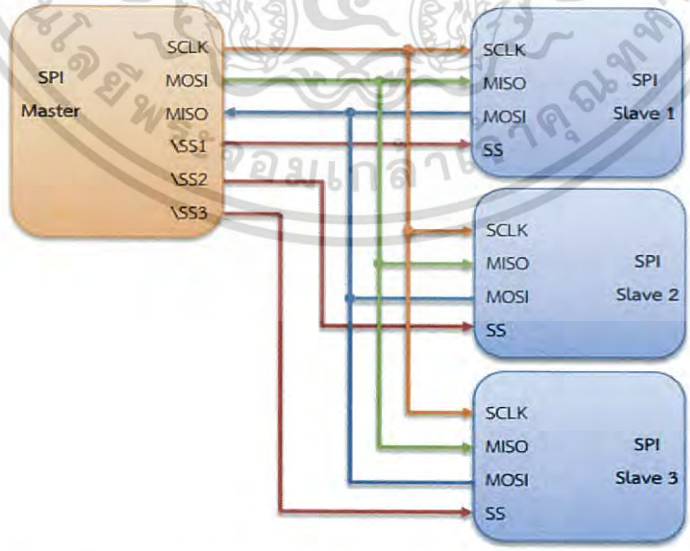
2.5 การติดต่อสื่อสารแบบ SPI (Serial Peripheral Interface)

เป็นการเชื่อมต่อสื่อสารแบบอนุกรมโดยอาศัยสัญญาณนาฬิกาเป็นตัวกำหนดจังหวะการรับส่งข้อมูล (Synchronous) ที่สามารถส่งข้อมูลไปยังปลายทางและรับข้อมูลจากปลายทางกลับมาในครั้งเดียวกัน (Full Duplex) SPI แบ่งอุปกรณ์ออกเป็น 2 ฝั่ง คือ Master เป็นตัวควบคุมการรับส่งข้อมูลโดยในที่นี้คือไมโครคอนโทรลเลอร์ กับ Slave เป็นอุปกรณ์ที่รอรับคำสั่งจาก Master โดย Slave มีได้มากกว่า 1 ตัว

SPI ใช้สายสัญญาณทั้งหมด 4 เส้นดังนี้

1. SCLK (Serial Clock) ใช้ส่งสัญญาณนาฬิกาจากอุปกรณ์ Master ไปยังอุปกรณ์ Slave เพื่อกำหนดจังหวะการรับส่งข้อมูล
2. MOSI (Master Out Slave In) ใช้ส่งข้อมูลจากอุปกรณ์ Master ไปยังอุปกรณ์ Slave
3. MISO (Master In Slave Out) ใช้รับข้อมูลจากอุปกรณ์ Slave
4. SS (Slave Select) หรือ ขา CS (Chip Select) ใช้ส่งสัญญาณ Low ไปยังอุปกรณ์ Slave ที่ต้องการรับส่งข้อมูล

รูปที่ 2.4 การเชื่อมต่อการสื่อสารแบบ SPI ระหว่างอุปกรณ์ Master – Slave



รูปที่ 2.5 การเชื่อมต่อการสื่อสารแบบ SPI ระหว่างอุปกรณ์ Master – Slave หลายตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบสัญญาณใน SPI BUS

รูปแบบสัญญาณ SPI มี 4 รูปแบบ แตกต่างกันที่ขอบสัญญาณนาฬิกา (Clock Polarity) และเฟส (Phase)

เมื่อ CPHA=0 และ CPOL=0 สัญญาณนาฬิกา (Clock) ในสถานะปกติจะเป็น Low และจะรับ-ส่งข้อมูลที่ขอบขาขึ้นของสัญญาณนาฬิกา (Rising Edge Clock)

เมื่อ CPHA=0 และ CPOL=1 สัญญาณนาฬิกา (Clock) ในสถานะปกติจะเป็น High และจะรับ-ส่งข้อมูลที่ขอบขาลงของสัญญาณนาฬิกา (Falling Edge Clock)

เมื่อ CPHA=1 และ CPOL=0 สัญญาณนาฬิกา (Clock) ในสถานะปกติจะเป็น Low และจะรับ-ส่งข้อมูลที่ขอบขาลงของสัญญาณนาฬิกา (Falling Edge Clock)

เมื่อ CPHA=1 และ CPOL=1 สัญญาณนาฬิกา (Clock) ในสถานะปกติจะเป็น High และจะรับ-ส่งข้อมูลที่ขอบขาขึ้นของสัญญาณนาฬิกา (Rising Edge Clock)

ดังนั้น จึงกำหนดเป็น Mode การทำงานได้ 4 โหมด คือ

- o Mode 0 = CPOL=0 และ CPHA=0
- o Mode 1 = CPOL=0 และ CPHA=1
- o Mode 2 = CPOL=1 และ CPHA=0
- o Mode 3 = CPOL=1 และ CPHA=1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและสร้าง

วิทยานิพนธ์ฉบับนี้นำเสนอระบบตรวจจับพฤติกรรมซึ่งมีขั้นตอน 2 ขั้นตอน ขั้นตอนแรกคือระบบตรวจวัดสัญญาณความเร่งและความเร็วเชิงมุมจากการเคลื่อนไหว และ สามารถส่งข้อมูลการเคลื่อนไหวได้ ขั้นตอนที่สองคือบันทึกและแสดงผลข้อมูลจากระบบตรวจวัดสัญญาณความเร่งและความเร็วเชิงมุมจากการเคลื่อนไหว โดยการเชื่อมต่อแบบไร้สาย และบันทึกผ่าน MicroSD Card

3.1ระบบตรวจวัดสัญญาณความเร่งและความเร็วเชิงมุมจากการเคลื่อนไหว

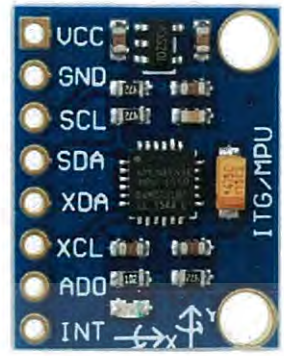
MPU-6050 NodeMPU-32s Bluetooth and MicroSD Card

รูปที่ 3.1 ระบบตรวจวัดสัญญาณความเร่งและความเร็วเชิงมุมจากการเคลื่อนไหว

ผู้ทดลองเคลื่อนไหวได้ค่ารับค่าจากเครื่องวัดความเร่ง (MPU-6050) ส่งข้อมูลให้ บอร์ด NodeMCU-32s รับข้อมูลและส่งข้อมูลไปบันทึกข้อมูลผ่าน MicroSD Card โดย MicroSD Card Module กับ คอมพิวเตอร์ผ่านบูลทธ โดย ชิปเซตภายใน Board NodeMCU-32s และแสดงข้อมูลผ่านโปรแกรม MATLAB และ Serial-Oscilloscope-v1.5 เป็นกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 เซ็นเซอร์วัดความเร่งและความเร็วเชิงมุม



รูปที่ 3.2 เซ็นเซอร์วัดความเร่ง (MPU-6050)

เซ็นเซอร์วัดความเร่ง (MPU-6050) เป็นโมดูล Accelerometer/Gyro Module เชื่อมต่อแบบ I2C ใช้สายสัญญาณ 2 เส้น ใช้ไฟ 3.3 โวลต์ เป็นโมดูลชื่อ GY-521 ใช้ชิปเซต MPU-6050 ต้องการไฟเลี้ยง 3 -5 โวลต์ สามารถกำหนดความรายละเอียดความเร่งได้ 4 ย่าน ± 2 , ± 4 , ± 8 และ ± 16 g และ มีโครงสร้างการเชื่อมต่อขา VCC , GND , SCL , SDA , XDA , XCL , ADO และ INT

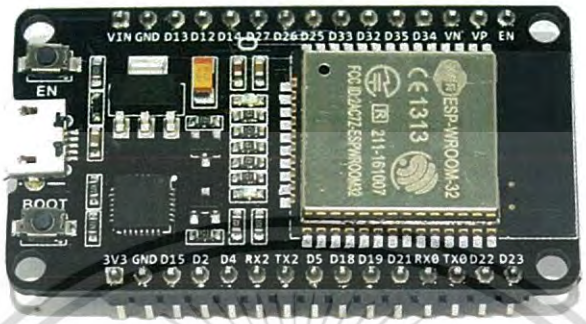
ตารางที่ 3.1 แสดงรายละเอียดการเชื่อมต่อเซ็นเซอร์วัดความเร่ง กับ บอร์ด NodeMCU-32s

ชื่อขา	เชื่อมต่อกับ
VCC	ขาจ่ายไฟเลี้ยง 3 .3 V.
GND	ขาสัญญาณ Ground
SCL	ขาสายสัญญาณนาฬิกา Pin 14
SDA	ขาสายสัญญาณข้อมูล Pin 11

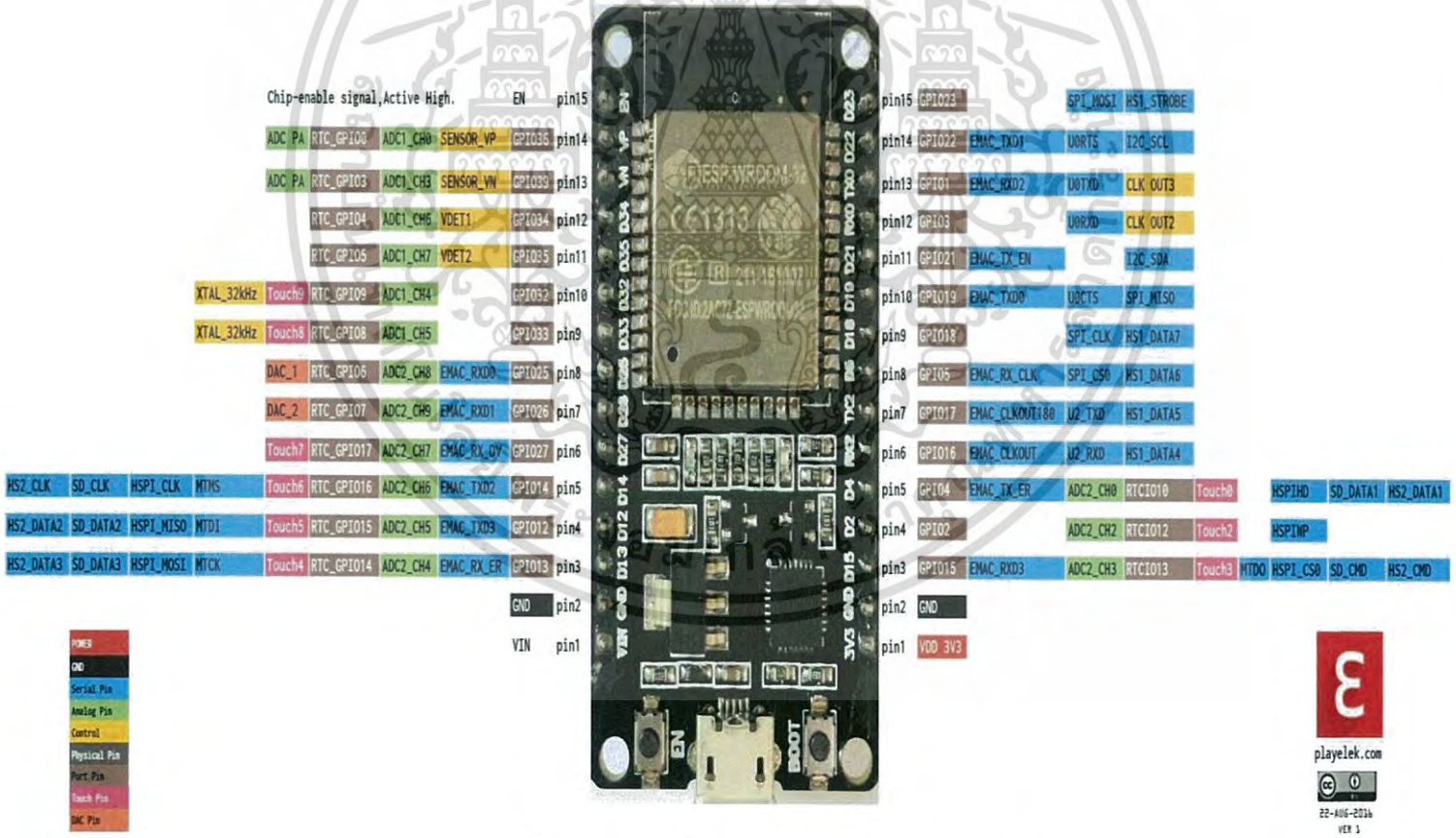
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 บอร์ด NodeMCU-32s

บอร์ด NodeMCU-32s ใช้สำหรับการประมวลผลและอ่านข้อมูลสัญญาณความเร่งจากเซ็นเซอร์วัดความเร่ง และ บอร์ด NodeMCU-32s ยังมีชิปเซตบูลทธภายใน ทำให้สามารถรับส่งข้อมูลได้สะดวก และ มีประสิทธิภาพ



รูปที่ 3.3 บอร์ด NodeMCU-32s



รูปที่ 3.4 ขาการเชื่อมต่อของบอร์ด NodeMCU-32s

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 โมดูล MicroSD Card

รูปที่ 3.5 โมดูล MicroSD Card

โมดูลสำหรับบันทึกข้อมูลลง Micro SD Card และ Micro SD Card Module ยี่ห้อ Catalex สำหรับเพิ่มความสามารถในการบันทึกข้อมูลต่าง ๆ จากบอร์ด NodeMCU-32s ลงบน Micro SD Card มีอินเตอร์เฟสแบบ SPI ใช้งานง่าย มีไลบารีสำเร็จรูปให้พร้อมใช้งาน มีวงจรเรกูเลต 3.3V มีในตัวบอร์ด สามารถใช้ไฟเลี้ยงได้ในช่วง 4.5V - 5.5 V

ตารางที่ 3.2 แสดงรายละเอียดการเชื่อมต่อขา SD Card Module กับ บอร์ด NodeMCU-32s

ชื่อขา	เชื่อมต่อกับ
CS	Pin 8
SCK	Pin 9
MOSI	Pin 15
MISO	Pin 10
VCC	ขาจ่ายไฟเลี้ยง 4.5 – 5.5 V.
GND	ขาสัญญาณ Ground

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 Real Time Clock DS3231

โมดูล Real Time Clock (RTC) คือ อุปกรณ์ที่ให้ค่าเวลาตามจริงโดยมีแบตเตอรี่ในตัวจึงไม่ต้องตั้งเวลาใหม่ทุกครั้ง และ ใช้การสื่อสารแบบ I2C ที่ใช้ SDA SCL VCC และ GND

รูปที่ 3.6 โมดูล Real Time Clock DS3231

ตารางที่ 3.3 แสดงรายละเอียดการเชื่อมต่อโมดูล Real Time Clock DS3231 กับ บอร์ด NodeMCU-32s

ชื่อขา	เชื่อมต่อกับ
VCC	ขาจ่ายไฟเลี้ยง 3 .3 V.
GND	ขาสัญญาณ Ground
SCL	ขาสายสัญญาณนาฬิกา Pin 14
SDA	ขาสายสัญญาณข้อมูล Pin 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.7 เครื่องวัดความเร่ง 3 แกนแบบไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

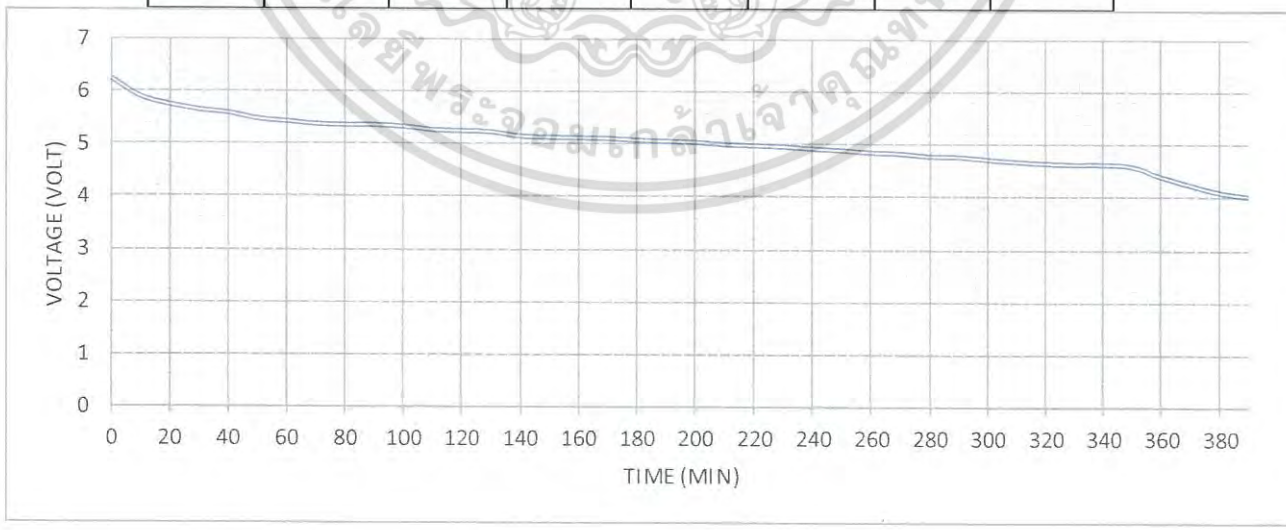
บทที่ 4

วิธีการทดสอบ และผลการทดสอบ

4.1 การทดสอบอายุการใช้งานของแบตเตอรี่ขนาด AAA 4 ก้อน

ตารางที่ 4.1 แสดงอายุการใช้งานที่เวลาเป็นหน่วยนาทีต่อแรงดันที่เป็นหน่วยโวลต์

Time (Min.)	Voltage (Volt)	Time (Min.)	Voltage (Volt)	Time (Min.)	Voltage (Volt)	Time (Min.)	Voltage (Volt)
0	6.24	100	5.33	200	5.04	300	4.72
10	5.90	110	5.27	210	5.00	310	4.68
20	5.75	120	5.25	220	4.98	320	4.65
30	5.65	130	5.23	230	4.96	330	4.63
40	5.59	140	5.15	240	4.92	340	4.63
50	5.48	150	5.13	250	4.89	350	4.59
60	5.43	160	5.12	260	4.85	360	4.40
70	5.38	170	5.11	270	4.83	370	4.24
80	5.36	180	5.08	280	4.78	380	4.10
90	5.36	190	5.06	290	4.77	390	4.02



รูปที่ 4.1 กราฟแสดงเวลา-โวลต์

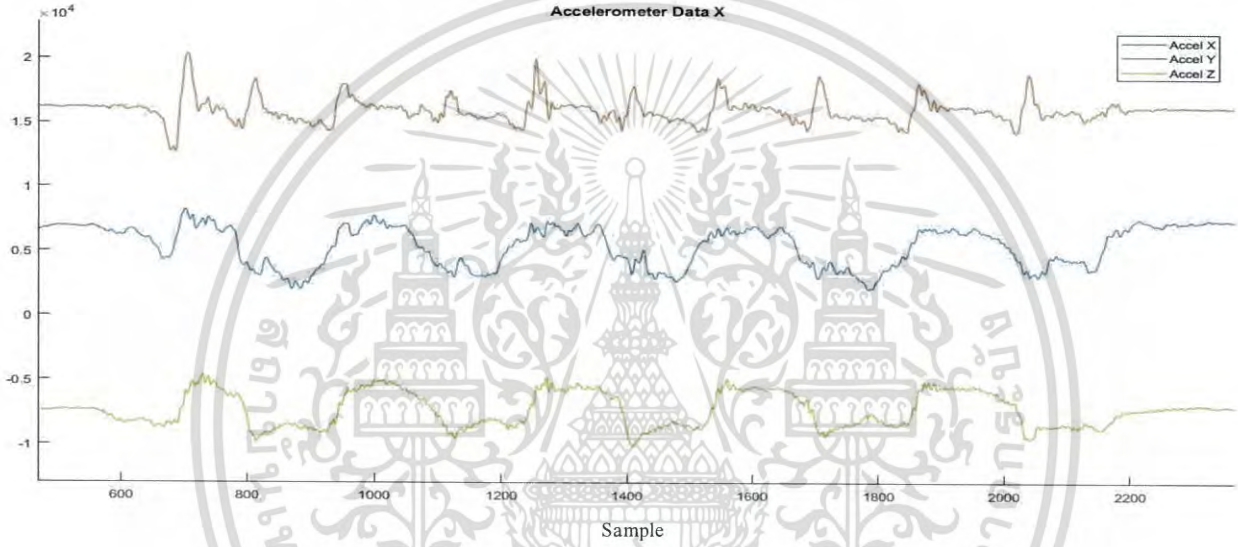
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลอง

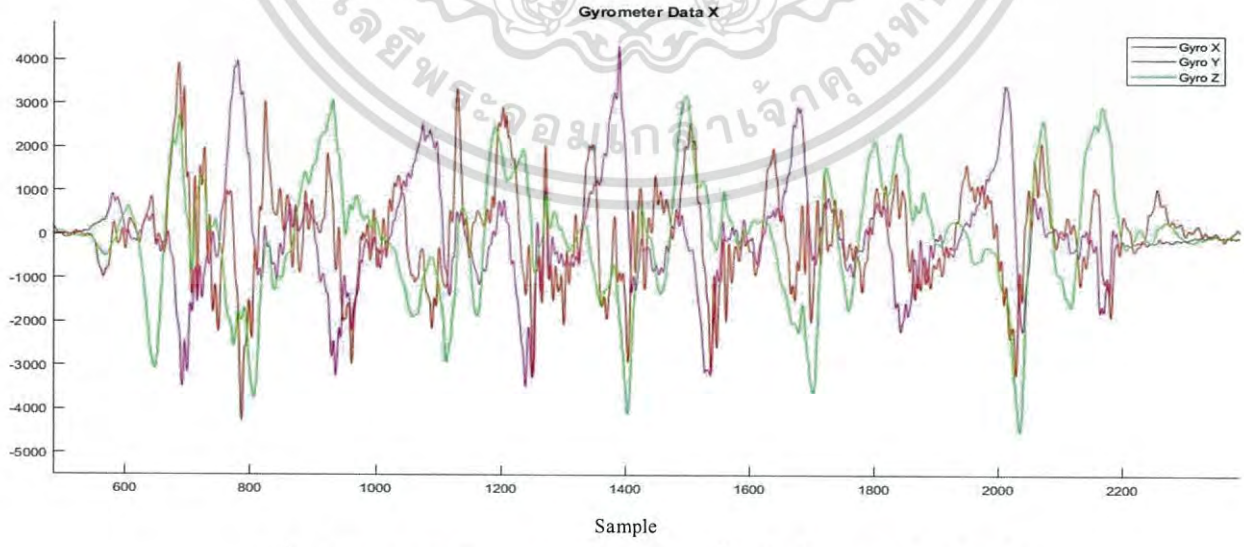
จากการทดลองจะเห็นได้ว่าที่แบตเตอรี่ AAA 4 ก้อนจะมีแรงดันเริ่มที่ 6.24 v. และค่อยๆลดลงเรื่อยๆเมื่อเวลาผ่านไปจนถึงนาที 350 ซึ่งโวลต์มีค่าต่ำกว่า 4.59 V ทำให้บลูทูธไม่ทำงานจึงส่งผลให้ค่าที่ได้หยุดด้วยเช่นกัน

สัญญาณความเร่งจากการเคลื่อนไหว

การเคลื่อนไหวที่ตรวจจับมีอยู่ 4 รูปแบบ ประกอบด้วย การเดินลงบันได , การเดินขึ้นบันได , การนั่งเก้าอี้ และลุกจากเก้าอี้ และ การเดิน

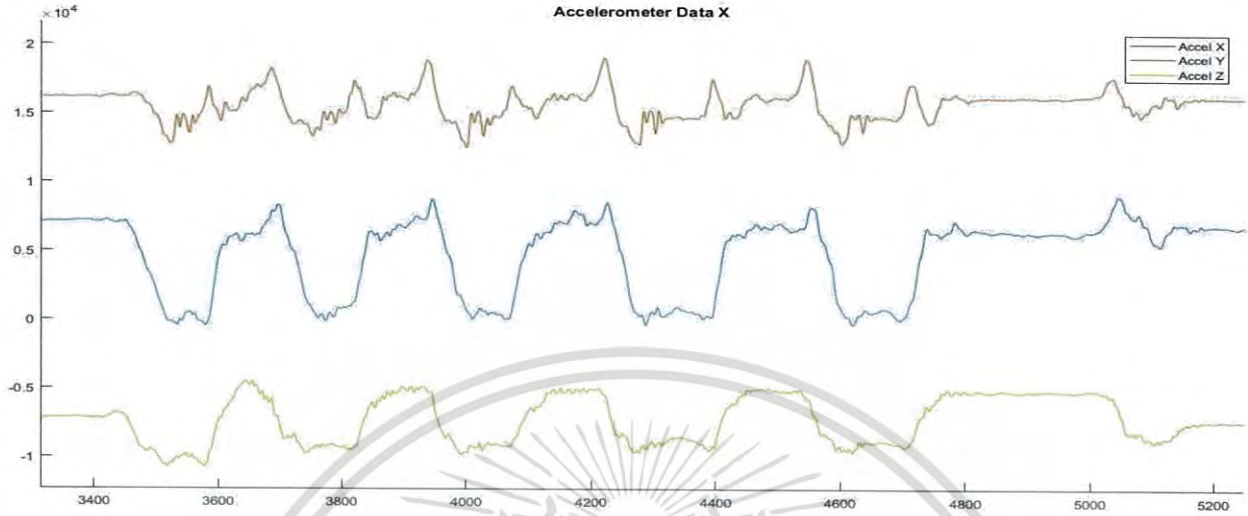


รูปที่ 4.2 สัญญาณ Accelerometer ของการเคลื่อนไหวโดยการเดินลงบันได

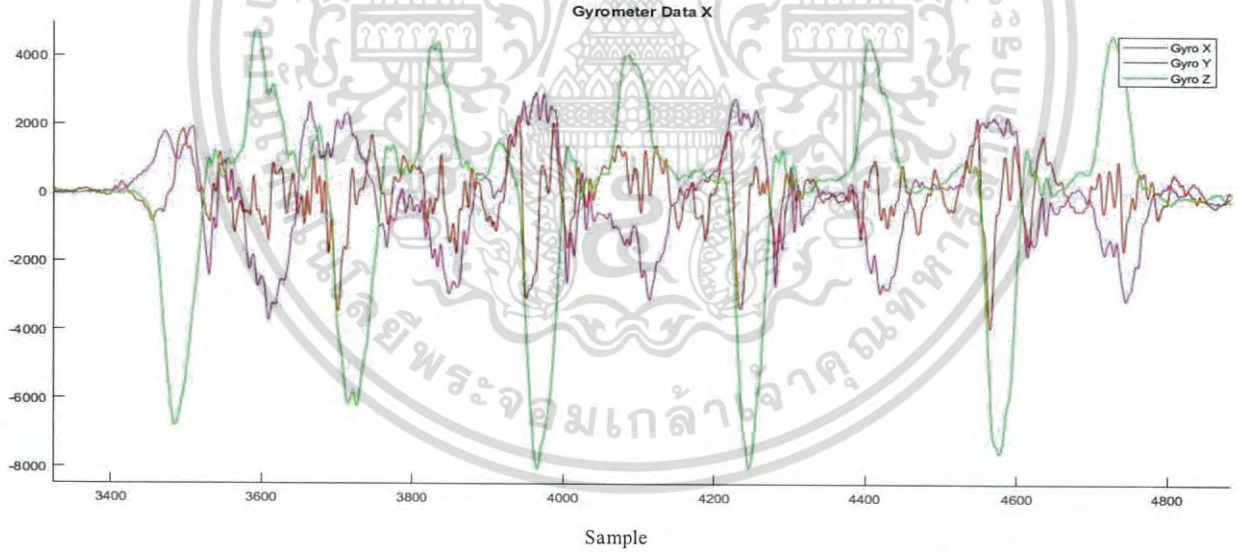


รูปที่ 4.3 สัญญาณ Gyroscope ของการเคลื่อนไหวโดยการเดินลงบันได

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

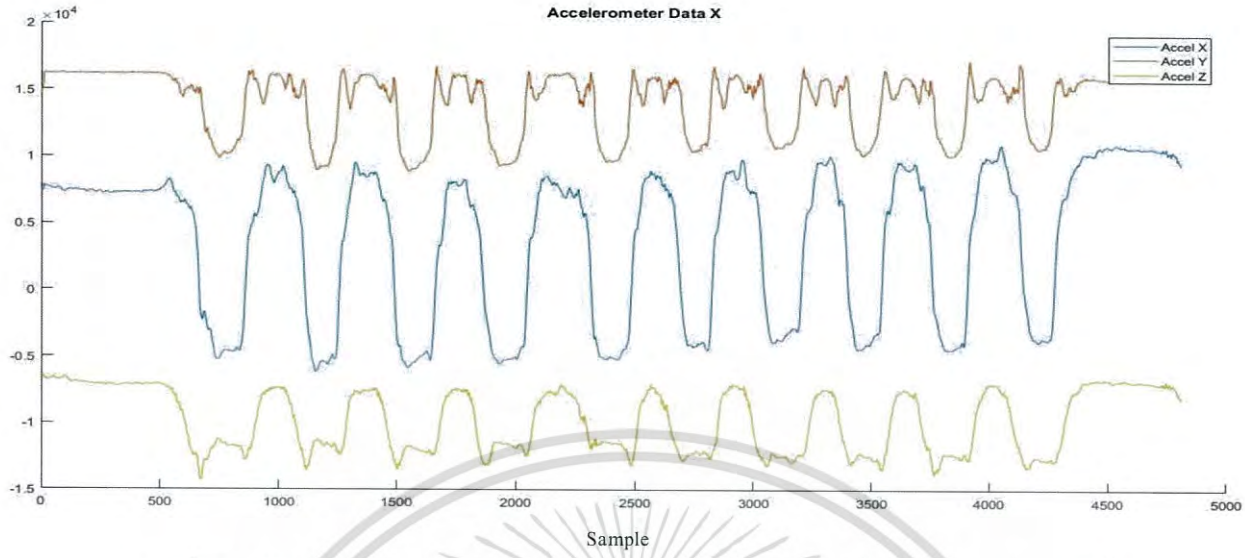


รูปที่ 4.4 สัญญาณ Accelerometer ของการเคลื่อนไหวโดยการเดินขึ้นบันได

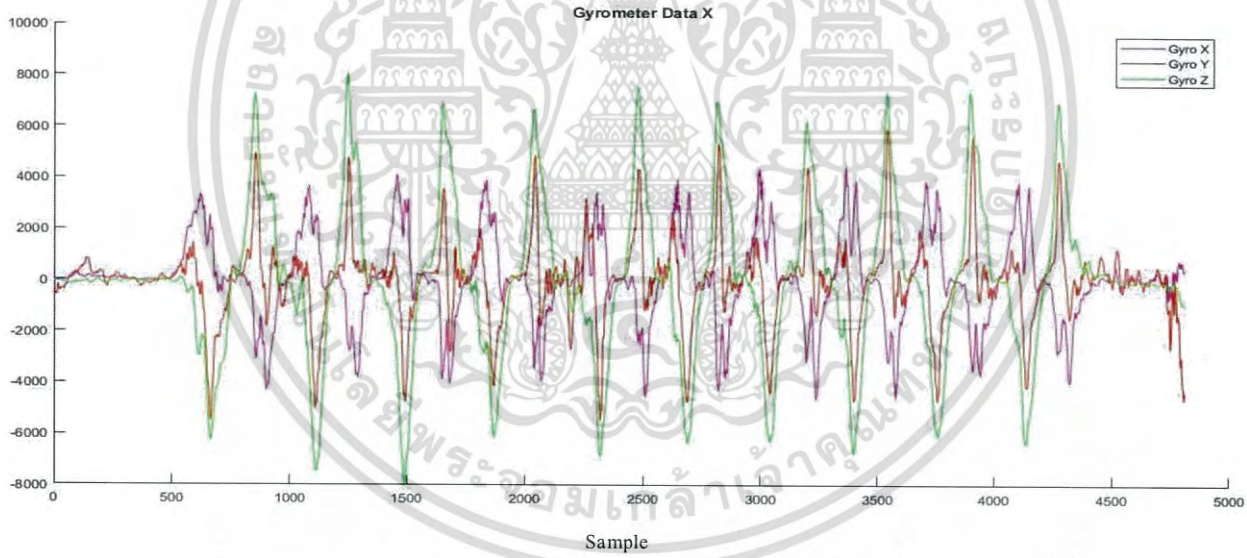


รูปที่ 4.5 สัญญาณ Gyroscope ของการเคลื่อนไหวโดยการเดินขึ้นบันได

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

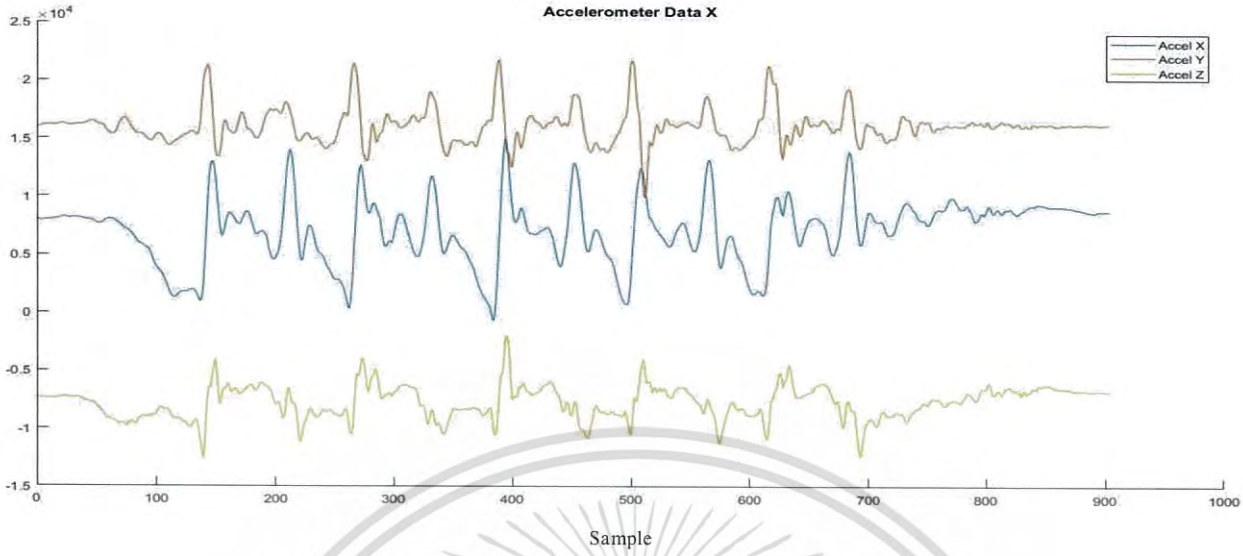


รูปที่ 4.6 สัญญาณ Accelerometer ของการเคลื่อนไหวโดยการนั่งเก้าอี้และลุกจากเก้าอี้

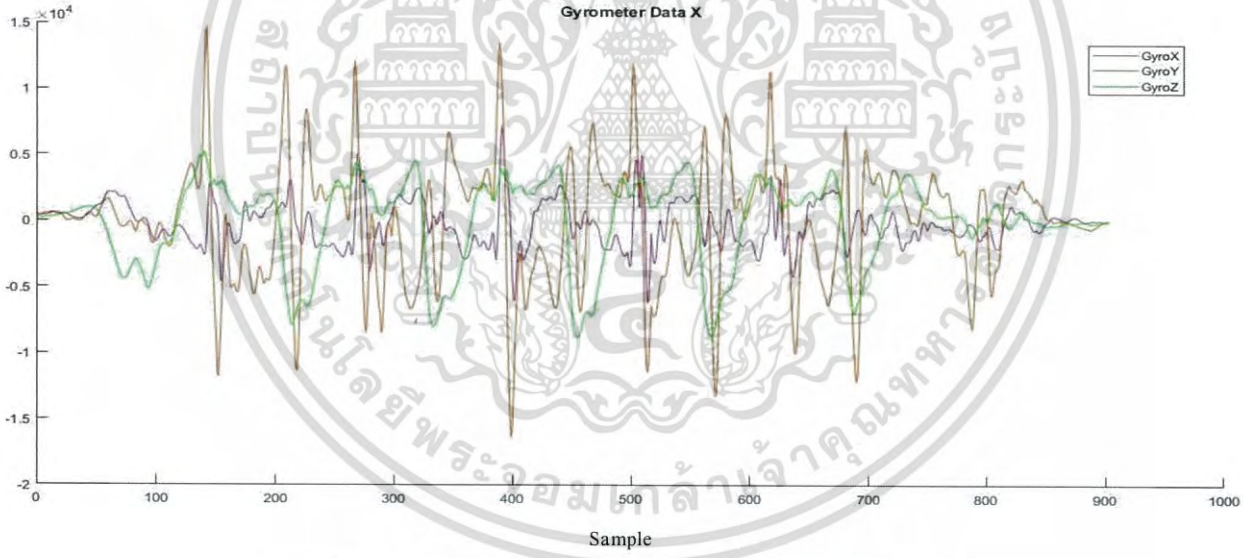


รูปที่ 4.7 สัญญาณ Gyroscope ของการเคลื่อนไหวโดยการนั่งเก้าอี้และลุกจากเก้าอี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

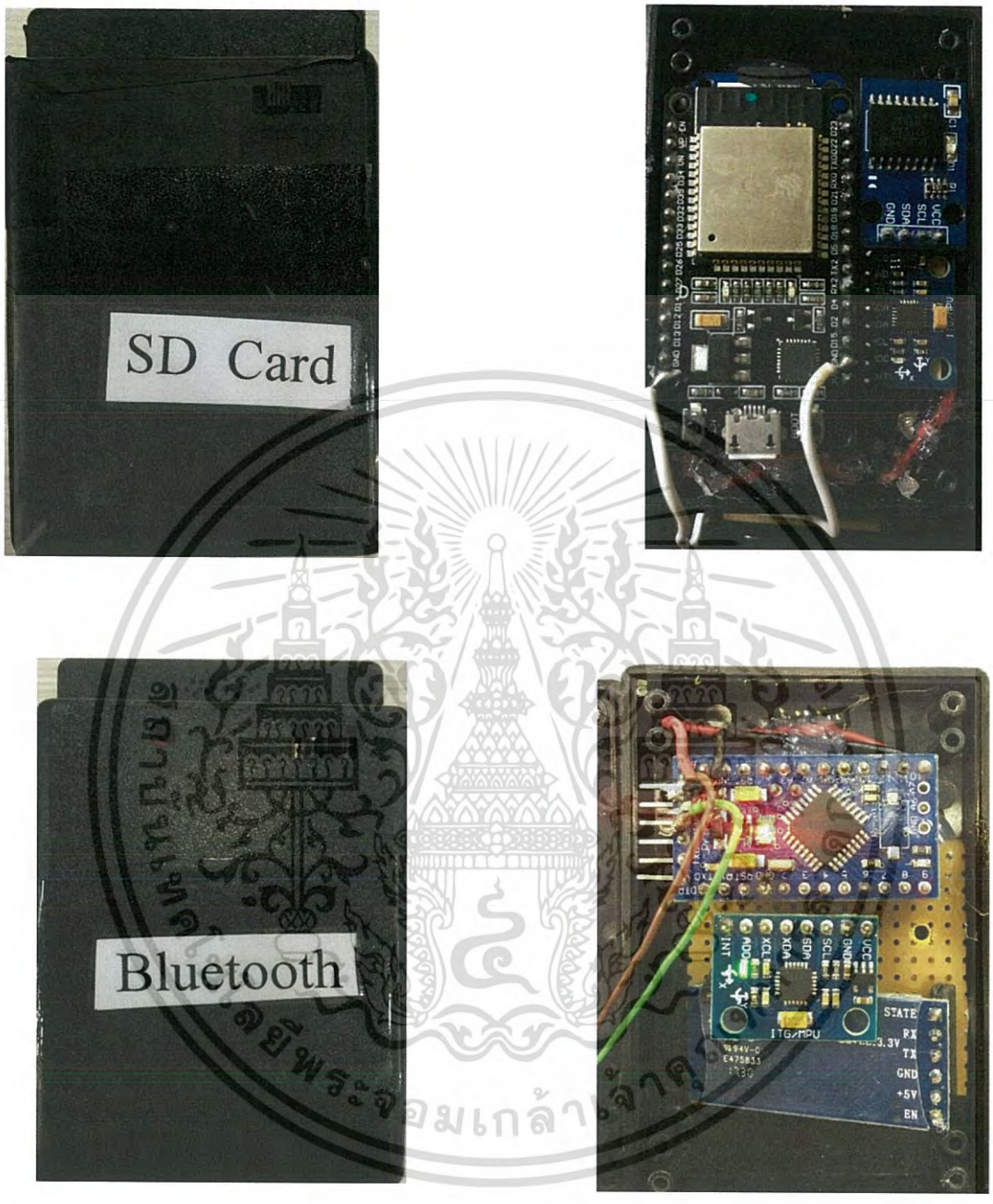


รูปที่ 4.8 สัญญาณ Accelerometer ของการเคลื่อนไหวโดยการเดิน



รูปที่ 4.9 สัญญาณ Gyroscope ของการเคลื่อนไหวโดยการเดิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 ชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

วิเคราะห์และสรุปผลการทดสอบคุณสมบัติของวงจร

5.1 สรุปผลการทดลอง

จากการทดลองเพื่อทดสอบคุณสมบัติของเครื่องวัดความเร่งแบบ 3 แกน แบบไร้สาย สามารถสรุปผลได้สามารถสรุปมาเป็นลักษณะกราฟ 4 รูปแบบ ประกอบด้วย การเดินลงบันได , การเดินขึ้นบันได , การนั่งเก้าอี้และลุกจากเก้าอี้ และ การเดิน

5.2 ปัญหาและอุปสรรคในการทำงาน

1. ความถี่ที่ได้ในการส่งรับข้อมูลมีความช้าเกินไป
2. มีปัญหาในการเขียน Code โปรแกรมของ Arduino IDE 1.8.5
3.ระยะเวลาของแบตเตอรี่ที่ใช้งานไม่นานเท่าที่ควร
4.ขนาดของตัวอุปกรณ์มีขนาดใหญ่
5.อุปกรณ์ไม่สะดวกในการใส่นอน

5.3 แนวทางการแก้ไข

1. เปลี่ยนค่า Baud rate จาก 9600 เป็น 57,600 ซึ่งทำให้ความถี่เพิ่มขึ้นเป็น 64 Hz.
2. ทำการทดลองเขียนโปรแกรมเพิ่มเติมและศึกษาภาษาที่ใช้ในการเขียนโปรแกรมเพิ่มมากขึ้น
3. ลองไปลงพื้นที่จริงเก็บข้อมูลจริงและรับฟังปัญหาจากอาสาสมัครเพื่อนำมาปรับปรุงอุปกรณ์

5.4 ประโยชน์และการประยุกต์การใช้งาน

สามารถนำความรู้ต่าง ๆที่ได้ศึกษามานั้น ไปใช้ในการคิดและออกแบบการทำงานต่าง ๆ และมีความรู้ความเข้าใจในการทำงานของบอร์ด NodeMCU-32s มากขึ้นและสามารถแก้ไขเวลาโครงงานมีปัญหาได้ และสามารถทราบกิจกรรมของอาสาสมัครว่ามีการเคลื่อนไหวในแต่ละวันมีการเคลื่อนไหวมากเกินไปหรือน้อยเกินไปหรืออยู่ในเกินมาตราฐานโดยใช้แบบตรวจสอบการเคลื่อนไหวจากอาจารย์ที่ปรึกษาของคณะกายภาพบำบัดมหาวิทยาลัยมหิดลมาเป็นเกณฑ์กำหนดว่ามีการเคลื่อนไหวมากเกินไปหรือน้อยเกินไปหรืออยู่ในเกินมาตราฐาน

5.5 สิ่งที่ได้จากโครงงาน

ในการศึกษาและออกแบบเครื่องวัดความเร่ง 3 แกนแบบไร้สาย ทำให้ข้าพเจ้าได้รับความรู้เกี่ยวกับการคำนวณ และเข้าใจหลักการทำงานของบอร์ด NodeMCU-32s , SD CARD MODULE , Real Time Clock และเซนเซอร์ MPU-6050 มากยิ่งขึ้น และยังเป็นการเพิ่มทักษะในการเขียนโปรแกรม Arduino IDE 1.8.5 ของผู้ทำและทราบเกี่ยวกับเรื่อง Accelerometers , Gyroscope และ เพิ่มทักษะในการแก้ปัญหาและการแก้ไขสถานการณ์ต่าง ๆ มากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

Accelerometer

[1]https://sites.google.com/site/thaimulticopter/acc

[2]http://accelerometer-gyroscope-sensor-app.blogspot.com/

Compass or Magnetometer

[3]http://guru.sanook.com/23783/

Gyroscope

[4]http://accelerometer-gyroscope-sensor-app.blogspot.com/

Nodemcu-32s

[5]https://www.arduinoall.com/search?q=nodemcu-32s

Bluetooth

[6]https://th.wikipedia.org/wiki/บลูทูธ

[7]https://www.arduinoall.com/product/103/bluetooth-serial-module-hc-05-master-slave-mode-2

I2C

[8]http://www.eclubthai.com/board/index.php?topic=16140.0;wap2

[9]http://www.mindphp.com/forums/viewtopic.php?f=215&t=34304

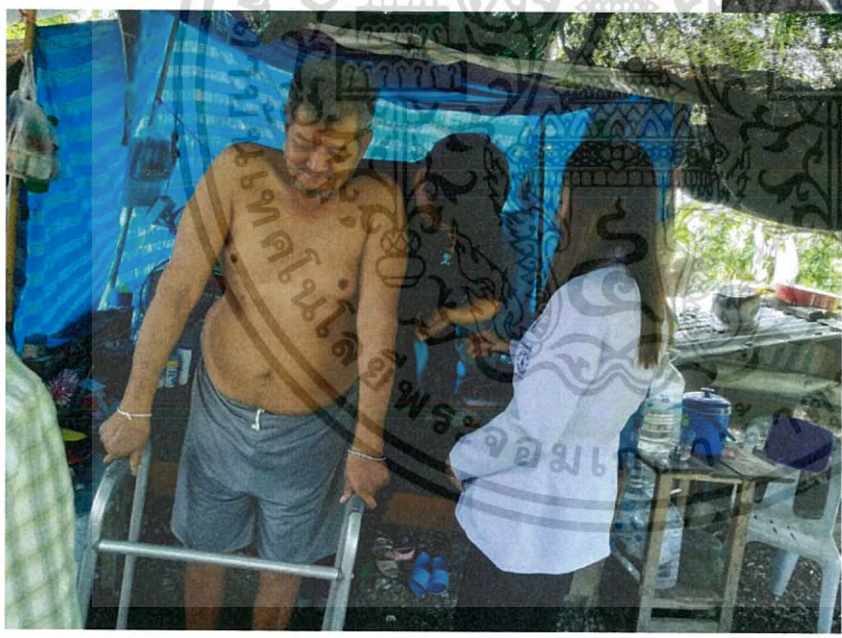
[10]http://cpre.kmutnb.ac.th/esl/learning/index.php?article=i2c-master-slave

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดในโปรแกรม Arduino IDE 1.8.5

การบันทึกข้อมูลแบบ SD Card

```
//#include "SdFat.h"
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include <RtcDS3231.h>
RtcDS3231<TwoWire> Rtc(Wire);
File myFile;
uint8_t k;
int jjj = 0;

const uint32_t RATE_KB_PER_SEC = 1000;

const uint32_t TEST_TIME_SEC = 100;

// Time between printing progress dots
const uint32_t DOT_TIME_MS = 5000UL;

// The name of the sensor is "MPU-6050".
// For program code, I omit the '-',
// therefor I use the name "MPU6050....".



// Register names according to the datasheet.
// According to the InvenSense document
// "MPU-6000 and MPU-6050 Register Map
// and Descriptions Revision 3.2", there are no registers
// at 0x02 ... 0x18, but according other information
// the registers in that unknown area are for gain
// and offsets.
//
#define MPU6050_AUX_VDDIO          0x01   // R/W
#define MPU6050_SMPLRT_DIV         0x19   // R/W
#define MPU6050_CONFIG             0x1A   // R/W
#define MPU6050_GYRO_CONFIG        0x1B   // R/W
#define MPU6050_ACCEL_CONFIG       0x1C   // R/W
#define MPU6050_FF_THR             0x1D   // R/W
#define MPU6050_FF_DUR             0x1E   // R/W
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define MPU6050_MOT_THR          0x1F  // R/W
#define MPU6050_MOT_DUR          0x20  // R/W
#define MPU6050_ZRMOT_THR        0x21  // R/W
#define MPU6050_ZRMOT_DUR        0x22  // R/W
#define MPU6050_FIFO_EN          0x23  // R/W
#define MPU6050_I2C_MST_CTRL     0x24  // R/W
#define MPU6050_I2C_SLV0_ADDR    0x25  // R/W
#define MPU6050_I2C_SLV0_REG     0x26  // R/W
#define MPU6050_I2C_SLV0_CTRL    0x27  // R/W
#define MPU6050_I2C_SLV1_ADDR    0x28  // R/W
#define MPU6050_I2C_SLV1_REG     0x29  // R/W
#define MPU6050_I2C_SLV1_CTRL    0x2A  // R/W
#define MPU6050_I2C_SLV2_ADDR    0x2B  // R/W
#define MPU6050_I2C_SLV2_REG     0x2C  // R/W
#define MPU6050_I2C_SLV2_CTRL    0x2D  // R/W
#define MPU6050_I2C_SLV3_ADDR    0x2E  // R/W
#define MPU6050_I2C_SLV3_REG     0x2F  // R/W
#define MPU6050_I2C_SLV3_CTRL    0x30  // R/W
#define MPU6050_I2C_SLV4_ADDR    0x31  // R/W
#define MPU6050_I2C_SLV4_REG     0x32  // R/W
#define MPU6050_I2C_SLV4_DO      0x33  // R/W
#define MPU6050_I2C_SLV4_CTRL    0x34  // R/W
#define MPU6050_I2C_SLV4_DI      0x35  // R
#define MPU6050_I2C_MST_STATUS   0x36  // R
#define MPU6050_INT_PIN_CFG      0x37  // R/W
#define MPU6050_INT_ENABLE       0x38  // R/W
#define MPU6050_INT_STATUS       0x3A  // R
#define MPU6050_ACCEL_XOUT_H     0x3B  // R
#define MPU6050_ACCEL_XOUT_L     0x3C  // R
#define MPU6050_ACCEL_YOUT_H     0x3D  // R
#define MPU6050_ACCEL_YOUT_L     0x3E  // R
#define MPU6050_ACCEL_ZOUT_H     0x3F  // R
#define MPU6050_ACCEL_ZOUT_L     0x40  // R
#define MPU6050_TEMP_OUT_H       0x41  // R
#define MPU6050_TEMP_OUT_L       0x42  // R
#define MPU6050_GYRO_XOUT_H      0x43  // R
#define MPU6050_GYRO_XOUT_L      0x44  // R
#define MPU6050_GYRO_YOUT_H      0x45  // R
#define MPU6050_GYRO_YOUT_L      0x46  // R
#define MPU6050_GYRO_ZOUT_H      0x47  // R
#define MPU6050_GYRO_ZOUT_L      0x48  // R
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define MPU6050_EXT_SENS_DATA_00 0x49  // R
#define MPU6050_EXT_SENS_DATA_01 0x4A  // R
#define MPU6050_EXT_SENS_DATA_02 0x4B  // R
#define MPU6050_EXT_SENS_DATA_03 0x4C  // R
#define MPU6050_EXT_SENS_DATA_04 0x4D  // R
#define MPU6050_EXT_SENS_DATA_05 0x4E  // R
#define MPU6050_EXT_SENS_DATA_06 0x4F  // R
#define MPU6050_EXT_SENS_DATA_07 0x50  // R
#define MPU6050_EXT_SENS_DATA_08 0x51  // R
#define MPU6050_EXT_SENS_DATA_09 0x52  // R
#define MPU6050_EXT_SENS_DATA_10 0x53  // R
#define MPU6050_EXT_SENS_DATA_11 0x54  // R
#define MPU6050_EXT_SENS_DATA_12 0x55  // R
#define MPU6050_EXT_SENS_DATA_13 0x56  // R
#define MPU6050_EXT_SENS_DATA_14 0x57  // R
#define MPU6050_EXT_SENS_DATA_15 0x58  // R
#define MPU6050_EXT_SENS_DATA_16 0x59  // R
#define MPU6050_EXT_SENS_DATA_17 0x5A  // R
#define MPU6050_EXT_SENS_DATA_18 0x5B  // R
#define MPU6050_EXT_SENS_DATA_19 0x5C  // R
#define MPU6050_EXT_SENS_DATA_20 0x5D  // R
#define MPU6050_EXT_SENS_DATA_21 0x5E  // R
#define MPU6050_EXT_SENS_DATA_22 0x5F  // R
#define MPU6050_EXT_SENS_DATA_23 0x60  // R
#define MPU6050_MOT_DETECT_STATUS 0x61  // R
#define MPU6050_I2C_SLV0_DO      0x63  // R/W
#define MPU6050_I2C_SLV1_DO      0x64  // R/W
#define MPU6050_I2C_SLV2_DO      0x65  // R/W
#define MPU6050_I2C_SLV3_DO      0x66  // R/W
#define MPU6050_I2C_MST_DELAY_CTRL 0x67  // R/W
#define MPU6050_SIGNAL_PATH_RESET 0x68  // R/W
#define MPU6050_MOT_DETECT_CTRL   0x69  // R/W
#define MPU6050_USER_CTRL        0x6A  // R/W
#define MPU6050_PWR_MGMT_1       0x6B  // R/W
#define MPU6050_PWR_MGMT_2       0x6C  // R/W
#define MPU6050_FIFO_COUNTH      0x72  // R/W
#define MPU6050_FIFO_COUNTL      0x73  // R/W
#define MPU6050_FIFO_R_W         0x74  // R/W
#define MPU6050_WHO_AM_I         0x75  // R

#define MPU6050_ACCEL_CONFIG     0xF0  // R/W
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Defines for the bits, to be able to change
// between bit number and binary definition.
// By using the bit number, programming the sensor
// is like programming the AVR microcontroller.
// But instead of using "(1<<X)", or "_BV(X)",
// the Arduino "bit(X)" is used.
#define MPU6050_D0 0
#define MPU6050_D1 1
#define MPU6050_D2 2
#define MPU6050_D3 3
#define MPU6050_D4 4
#define MPU6050_D5 5
#define MPU6050_D6 6
#define MPU6050_D7 7

// AUX_VDDIO Register
#define MPU6050_AUX_VDDIO MPU6050_D7  // I2C high: 1=VDD, 0=VLOGIC

// CONFIG Register
// DLPF is Digital Low Pass Filter for both gyro and accelerometers.
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_DLPF_CFG0     MPU6050_D0
#define MPU6050_DLPF_CFG1     MPU6050_D1
#define MPU6050_DLPF_CFG2     MPU6050_D2
#define MPU6050_EXT_SYNC_SET0 MPU6050_D3
#define MPU6050_EXT_SYNC_SET1 MPU6050_D4
#define MPU6050_EXT_SYNC_SET2 MPU6050_D5

// Combined definitions for the EXT_SYNC_SET values
#define MPU6050_EXT_SYNC_SET_0 (0)
#define MPU6050_EXT_SYNC_SET_1 (bit(MPU6050_EXT_SYNC_SET0))
#define MPU6050_EXT_SYNC_SET_2 (bit(MPU6050_EXT_SYNC_SET1))
#define MPU6050_EXT_SYNC_SET_3 (bit(MPU6050_EXT_SYNC_SET1)|bit(MPU6050_EXT_SYNC_SET0))
#define MPU6050_EXT_SYNC_SET_4 (bit(MPU6050_EXT_SYNC_SET2))
#define MPU6050_EXT_SYNC_SET_5 (bit(MPU6050_EXT_SYNC_SET2)|bit(MPU6050_EXT_SYNC_SET0))
#define MPU6050_EXT_SYNC_SET_6 (bit(MPU6050_EXT_SYNC_SET2)|bit(MPU6050_EXT_SYNC_SET1))
#define MPU6050_EXT_SYNC_SET_7
(bit(MPU6050_EXT_SYNC_SET2)|bit(MPU6050_EXT_SYNC_SET1)|bit(MPU6050_EXT_SYNC_SET0))

// Alternative names for the combined definitions.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define MPU6050_EXT_SYNC_DISABLED    MPU6050_EXT_SYNC_SET_0
#define MPU6050_EXT_SYNC_TEMP_OUT_L  MPU6050_EXT_SYNC_SET_1
#define MPU6050_EXT_SYNC_GYRO_XOUT_L MPU6050_EXT_SYNC_SET_2
#define MPU6050_EXT_SYNC_GYRO_YOUT_L MPU6050_EXT_SYNC_SET_3
#define MPU6050_EXT_SYNC_GYRO_ZOUT_L MPU6050_EXT_SYNC_SET_4
#define MPU6050_EXT_SYNC_ACCEL_XOUT_L MPU6050_EXT_SYNC_SET_5
#define MPU6050_EXT_SYNC_ACCEL_YOUT_L MPU6050_EXT_SYNC_SET_6
#define MPU6050_EXT_SYNC_ACCEL_ZOUT_L MPU6050_EXT_SYNC_SET_7

// Combined definitions for the DLPF_CFG values
#define MPU6050_DLPF_CFG_0 (0)
#define MPU6050_DLPF_CFG_1 (bit(MPU6050_DLPF_CFG0))
#define MPU6050_DLPF_CFG_2 (bit(MPU6050_DLPF_CFG1))
#define MPU6050_DLPF_CFG_3 (bit(MPU6050_DLPF_CFG1)|bit(MPU6050_DLPF_CFG0))
#define MPU6050_DLPF_CFG_4 (bit(MPU6050_DLPF_CFG2))
#define MPU6050_DLPF_CFG_5 (bit(MPU6050_DLPF_CFG2)|bit(MPU6050_DLPF_CFG0))
#define MPU6050_DLPF_CFG_6 (bit(MPU6050_DLPF_CFG2)|bit(MPU6050_DLPF_CFG1))
#define MPU6050_DLPF_CFG_7 (bit(MPU6050_DLPF_CFG2)|bit(MPU6050_DLPF_CFG1)|bit(MPU6050_DLPF_CFG0))

// Alternative names for the combined definitions
// This name uses the bandwidth (Hz) for the accelometer,
// for the gyro the bandwidth is almost the same.
#define MPU6050_DLPF_260HZ   MPU6050_DLPF_CFG_0
#define MPU6050_DLPF_184HZ   MPU6050_DLPF_CFG_1
#define MPU6050_DLPF_94HZ    MPU6050_DLPF_CFG_2
#define MPU6050_DLPF_44HZ    MPU6050_DLPF_CFG_3
#define MPU6050_DLPF_21HZ    MPU6050_DLPF_CFG_4
#define MPU6050_DLPF_10HZ    MPU6050_DLPF_CFG_5
#define MPU6050_DLPF_5HZ     MPU6050_DLPF_CFG_6
#define MPU6050_DLPF_RESERVED MPU6050_DLPF_CFG_7

// GYRO_CONFIG Register
// The XG_ST, YG_ST, ZG_ST are bits for selftest.
// The FS_SEL sets the range for the gyro.
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_FS_SEL0 MPU6050_D3
#define MPU6050_FS_SEL1 MPU6050_D4
#define MPU6050_ZG_ST   MPU6050_D5
#define MPU6050_YG_ST   MPU6050_D6
#define MPU6050_XG_ST   MPU6050_D7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Combined definitions for the FS_SEL values
#define MPU6050_FS_SEL_0 (0)
#define MPU6050_FS_SEL_1 (bit(MPU6050_FS_SEL0))
#define MPU6050_FS_SEL_2 (bit(MPU6050_FS_SEL1))
#define MPU6050_FS_SEL_3 (bit(MPU6050_FS_SEL1)|bit(MPU6050_FS_SEL0))

// Alternative names for the combined definitions
// The name uses the range in degrees per second.
#define MPU6050_FS_SEL_250 MPU6050_FS_SEL_0
#define MPU6050_FS_SEL_500 MPU6050_FS_SEL_1
#define MPU6050_FS_SEL_1000 MPU6050_FS_SEL_2
#define MPU6050_FS_SEL_2000 MPU6050_FS_SEL_3

// ACCEL_CONFIG Register
// The XA_ST, YA_ST, ZA_ST are bits for selftest.
// The AFS_SEL sets the range for the accelerometer.
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_ACCEL_HPF0 MPU6050_D0
#define MPU6050_ACCEL_HPF1 MPU6050_D1
#define MPU6050_ACCEL_HPF2 MPU6050_D2
#define MPU6050_AFS_SEL0  MPU6050_D3
#define MPU6050_AFS_SEL1  MPU6050_D4
#define MPU6050_ZA_ST     MPU6050_D5
#define MPU6050_YA_ST     MPU6050_D6
#define MPU6050_XA_ST     MPU6050_D7

// Combined definitions for the ACCEL_HPF values
#define MPU6050_ACCEL_HPF_0 (0)
#define MPU6050_ACCEL_HPF_1 (bit(MPU6050_ACCEL_HPF0))
#define MPU6050_ACCEL_HPF_2 (bit(MPU6050_ACCEL_HPF1))
#define MPU6050_ACCEL_HPF_3 (bit(MPU6050_ACCEL_HPF1)|bit(MPU6050_ACCEL_HPF0))
#define MPU6050_ACCEL_HPF_4 (bit(MPU6050_ACCEL_HPF2))
#define MPU6050_ACCEL_HPF_7 (bit(MPU6050_ACCEL_HPF2)|bit(MPU6050_ACCEL_HPF1)|bit(MPU6050_ACCEL_HPF0))

// Alternative names for the combined definitions
// The name uses the Cut-off frequency.
#define MPU6050_ACCEL_HPF_RESET MPU6050_ACCEL_HPF_0
#define MPU6050_ACCEL_HPF_5HZ   MPU6050_ACCEL_HPF_1
#define MPU6050_ACCEL_HPF_2_5HZ MPU6050_ACCEL_HPF_2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define MPU6050_ACCEL_HPF_1_25HZ MPU6050_ACCEL_HPF_3
#define MPU6050_ACCEL_HPF_0_63HZ MPU6050_ACCEL_HPF_4
#define MPU6050_ACCEL_HPF_HOLD   MPU6050_ACCEL_HPF_7

// Combined definitions for the AFS_SEL values
#define MPU6050_AFS_SEL_0 (0)
#define MPU6050_AFS_SEL_1 (bit(MPU6050_AFS_SEL0))
#define MPU6050_AFS_SEL_2 (bit(MPU6050_AFS_SEL1))
#define MPU6050_AFS_SEL_3 (bit(MPU6050_AFS_SEL1)|bit(MPU6050_AFS_SEL0))

// Alternative names for the combined definitions
// The name uses the full scale range for the accelerometer.
#define MPU6050_AFS_SEL_2G  MPU6050_AFS_SEL_0
#define MPU6050_AFS_SEL_4G  MPU6050_AFS_SEL_1
#define MPU6050_AFS_SEL_8G  MPU6050_AFS_SEL_2
#define MPU6050_AFS_SEL_16G MPU6050_AFS_SEL_3

#define MPU6050_AFS_SEL_16G 0xF0

// FIFO_EN Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_SLV0_FIFO_EN  MPU6050_D0
#define MPU6050_SLV1_FIFO_EN  MPU6050_D1
#define MPU6050_SLV2_FIFO_EN  MPU6050_D2
#define MPU6050_ACCEL_FIFO_EN MPU6050_D3
#define MPU6050_ZG_FIFO_EN    MPU6050_D4
#define MPU6050_YG_FIFO_EN    MPU6050_D5
#define MPU6050_XG_FIFO_EN    MPU6050_D6
#define MPU6050_TEMP_FIFO_EN  MPU6050_D7

// I2C_MST_CTRL Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_I2C_MST_CLK0  MPU6050_D0
#define MPU6050_I2C_MST_CLK1  MPU6050_D1
#define MPU6050_I2C_MST_CLK2  MPU6050_D2
#define MPU6050_I2C_MST_CLK3  MPU6050_D3
#define MPU6050_I2C_MST_P_NSR MPU6050_D4
#define MPU6050_SLV_3_FIFO_EN MPU6050_D5
#define MPU6050_WAIT_FOR_ES   MPU6050_D6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define MPU6050_MULT_MST_EN  MPU6050_D7

// Combined definitions  for the I2C_MST_CLK
#define MPU6050_I2C_MST_CLK_0 (0)
#define MPU6050_I2C_MST_CLK_1 (bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_2 (bit(MPU6050_I2C_MST_CLK1))
#define MPU6050_I2C_MST_CLK_3 (bit(MPU6050_I2C_MST_CLK1)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_4 (bit(MPU6050_I2C_MST_CLK2))
#define MPU6050_I2C_MST_CLK_5 (bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_6 (bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK1))
#define MPU6050_I2C_MST_CLK_7 (bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK1)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_8 (bit(MPU6050_I2C_MST_CLK3))
#define MPU6050_I2C_MST_CLK_9 (bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_10 (bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK1))
#define MPU6050_I2C_MST_CLK_11 (bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK1)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_12 (bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK2))
#define MPU6050_I2C_MST_CLK_13 (bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_14 (bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK1))
#define MPU6050_I2C_MST_CLK_15
(bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK1)|bit(MPU6050_I2C_MST_CLK0))

// Alternative names for the combined definitions
// The names uses I2C Master Clock Speed in kHz.
#define MPU6050_I2C_MST_CLK_348KHZ MPU6050_I2C_MST_CLK_0
#define MPU6050_I2C_MST_CLK_333KHZ MPU6050_I2C_MST_CLK_1
#define MPU6050_I2C_MST_CLK_320KHZ MPU6050_I2C_MST_CLK_2
#define MPU6050_I2C_MST_CLK_308KHZ MPU6050_I2C_MST_CLK_3
#define MPU6050_I2C_MST_CLK_296KHZ MPU6050_I2C_MST_CLK_4
#define MPU6050_I2C_MST_CLK_286KHZ MPU6050_I2C_MST_CLK_5
#define MPU6050_I2C_MST_CLK_276KHZ MPU6050_I2C_MST_CLK_6
#define MPU6050_I2C_MST_CLK_267KHZ MPU6050_I2C_MST_CLK_7
#define MPU6050_I2C_MST_CLK_258KHZ MPU6050_I2C_MST_CLK_8
#define MPU6050_I2C_MST_CLK_500KHZ MPU6050_I2C_MST_CLK_9
#define MPU6050_I2C_MST_CLK_471KHZ MPU6050_I2C_MST_CLK_10
#define MPU6050_I2C_MST_CLK_444KHZ MPU6050_I2C_MST_CLK_11
#define MPU6050_I2C_MST_CLK_421KHZ MPU6050_I2C_MST_CLK_12
#define MPU6050_I2C_MST_CLK_400KHZ MPU6050_I2C_MST_CLK_13
#define MPU6050_I2C_MST_CLK_381KHZ MPU6050_I2C_MST_CLK_14
#define MPU6050_I2C_MST_CLK_364KHZ MPU6050_I2C_MST_CLK_15

// I2C_SLV0_ADDR Register
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_I2C_SLV0_RW MPU6050_D7

// I2C_SLV0_CTRL Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_I2C_SLV0_LEN0   MPU6050_D0
#define MPU6050_I2C_SLV0_LEN1   MPU6050_D1
#define MPU6050_I2C_SLV0_LEN2   MPU6050_D2
#define MPU6050_I2C_SLV0_LEN3   MPU6050_D3
#define MPU6050_I2C_SLV0_GRP    MPU6050_D4
#define MPU6050_I2C_SLV0_REG_DIS MPU6050_D5
#define MPU6050_I2C_SLV0_BYTE_SW MPU6050_D6
#define MPU6050_I2C_SLV0_EN     MPU6050_D7

// A mask for the length
#define MPU6050_I2C_SLV0_LEN_MASK 0x0F

// I2C_SLV1_ADDR Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_I2C_SLV1_RW MPU6050_D7

// I2C_SLV1_CTRL Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_I2C_SLV1_LEN0   MPU6050_D0
#define MPU6050_I2C_SLV1_LEN1   MPU6050_D1
#define MPU6050_I2C_SLV1_LEN2   MPU6050_D2
#define MPU6050_I2C_SLV1_LEN3   MPU6050_D3
#define MPU6050_I2C_SLV1_GRP    MPU6050_D4
#define MPU6050_I2C_SLV1_REG_DIS MPU6050_D5
#define MPU6050_I2C_SLV1_BYTE_SW MPU6050_D6
#define MPU6050_I2C_SLV1_EN     MPU6050_D7

// A mask for the length
#define MPU6050_I2C_SLV1_LEN_MASK 0x0F

// I2C_SLV2_ADDR Register
// These are the names for the bits.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Use these only with the bit() macro.
#define MPU6050_I2C_SLV2_RW MPU6050_D7

// I2C_SLV2_CTRL Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_I2C_SLV2_LEN0   MPU6050_D0
#define MPU6050_I2C_SLV2_LEN1   MPU6050_D1
#define MPU6050_I2C_SLV2_LEN2   MPU6050_D2
#define MPU6050_I2C_SLV2_LEN3   MPU6050_D3
#define MPU6050_I2C_SLV2_GRP    MPU6050_D4
#define MPU6050_I2C_SLV2_REG_DIS MPU6050_D5
#define MPU6050_I2C_SLV2_BYTE_SW MPU6050_D6
#define MPU6050_I2C_SLV2_EN     MPU6050_D7

// A mask for the length
#define MPU6050_I2C_SLV2_LEN_MASK 0x0F

// I2C_SLV3_ADDR Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_I2C_SLV3_RW MPU6050_D7

// I2C_SLV3_CTRL Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_I2C_SLV3_LEN0   MPU6050_D0
#define MPU6050_I2C_SLV3_LEN1   MPU6050_D1
#define MPU6050_I2C_SLV3_LEN2   MPU6050_D2
#define MPU6050_I2C_SLV3_LEN3   MPU6050_D3
#define MPU6050_I2C_SLV3_GRP    MPU6050_D4
#define MPU6050_I2C_SLV3_REG_DIS MPU6050_D5
#define MPU6050_I2C_SLV3_BYTE_SW MPU6050_D6
#define MPU6050_I2C_SLV3_EN     MPU6050_D7

// A mask for the length
#define MPU6050_I2C_SLV3_LEN_MASK 0x0F

// I2C_SLV4_ADDR Register
// These are the names for the bits.
// Use these only with the bit() macro.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define MPU6050_I2C_SLV4_RW MPU6050_D7

// I2C_SLV4_CTRL Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_I2C_MST_DLY0    MPU6050_D0
#define MPU6050_I2C_MST_DLY1    MPU6050_D1
#define MPU6050_I2C_MST_DLY2    MPU6050_D2
#define MPU6050_I2C_MST_DLY3    MPU6050_D3
#define MPU6050_I2C_MST_DLY4    MPU6050_D4
#define MPU6050_I2C_SLV4_REG_DIS MPU6050_D5
#define MPU6050_I2C_SLV4_INT_EN  MPU6050_D6
#define MPU6050_I2C_SLV4_EN     MPU6050_D7

// A mask for the delay
#define MPU6050_I2C_MST_DLY_MASK 0x1F

// I2C_MST_STATUS Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_I2C_SLV0_NACK MPU6050_D0
#define MPU6050_I2C_SLV1_NACK MPU6050_D1
#define MPU6050_I2C_SLV2_NACK MPU6050_D2
#define MPU6050_I2C_SLV3_NACK MPU6050_D3
#define MPU6050_I2C_SLV4_NACK MPU6050_D4
#define MPU6050_I2C_LOST_ARB  MPU6050_D5
#define MPU6050_I2C_SLV4_DONE MPU6050_D6
#define MPU6050_PASS_THROUGH  MPU6050_D7

// I2C_PIN_CFG Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_CLKOUT_EN       MPU6050_D0
#define MPU6050_I2C_BYPASS_EN   MPU6050_D1
#define MPU6050_FSYNC_INT_EN    MPU6050_D2
#define MPU6050_FSYNC_INT_LEVEL MPU6050_D3
#define MPU6050_INT_RD_CLEAR    MPU6050_D4
#define MPU6050_LATCH_INT_EN    MPU6050_D5
#define MPU6050_INT_OPEN        MPU6050_D6
#define MPU6050_INT_LEVEL       MPU6050_D7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// INT_ENABLE Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_DATA_RDY_EN    MPU6050_D0
#define MPU6050_I2C_MST_INT_EN MPU6050_D3
#define MPU6050_FIFO_OFLOW_EN  MPU6050_D4
#define MPU6050_ZMOT_EN        MPU6050_D5
#define MPU6050_MOT_EN         MPU6050_D6
#define MPU6050_FF_EN          MPU6050_D7

// INT_STATUS Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_DATA_RDY_INT   MPU6050_D0
#define MPU6050_I2C_MST_INT    MPU6050_D3
#define MPU6050_FIFO_OFLOW_INT MPU6050_D4
#define MPU6050_ZMOT_INT       MPU6050_D5
#define MPU6050_MOT_INT        MPU6050_D6
#define MPU6050_FF_INT         MPU6050_D7

// MOT_DETECT_STATUS Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_MOT_ZRMOT MPU6050_D0
#define MPU6050_MOT_ZPOS  MPU6050_D2
#define MPU6050_MOT_ZNEG  MPU6050_D3
#define MPU6050_MOT_YPOS  MPU6050_D4
#define MPU6050_MOT_YNEG  MPU6050_D5
#define MPU6050_MOT_XPOS  MPU6050_D6
#define MPU6050_MOT_XNEG  MPU6050_D7

// IC2_MST_DELAY_CTRL Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_I2C_SLV0_DLY_EN MPU6050_D0
#define MPU6050_I2C_SLV1_DLY_EN MPU6050_D1
#define MPU6050_I2C_SLV2_DLY_EN MPU6050_D2
#define MPU6050_I2C_SLV3_DLY_EN MPU6050_D3
#define MPU6050_I2C_SLV4_DLY_EN MPU6050_D4
#define MPU6050_DELAY_ES_SHADOW MPU6050_D7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// SIGNAL_PATH_RESET Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_TEMP_RESET MPU6050_D0
#define MPU6050_ACCEL_RESET MPU6050_D1
#define MPU6050_GYRO_RESET MPU6050_D2

// MOT_DETECT_CTRL Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_MOT_COUNT0      MPU6050_D0
#define MPU6050_MOT_COUNT1      MPU6050_D1
#define MPU6050_FF_COUNT0       MPU6050_D2
#define MPU6050_FF_COUNT1       MPU6050_D3
#define MPU6050_ACCEL_ON_DELAY0 MPU6050_D4
#define MPU6050_ACCEL_ON_DELAY1 MPU6050_D5

// Combined definitions for the MOT_COUNT
#define MPU6050_MOT_COUNT_0 (0)
#define MPU6050_MOT_COUNT_1 (bit(MPU6050_MOT_COUNT0))
#define MPU6050_MOT_COUNT_2 (bit(MPU6050_MOT_COUNT1))
#define MPU6050_MOT_COUNT_3 (bit(MPU6050_MOT_COUNT1)|bit(MPU6050_MOT_COUNT0))

// Alternative names for the combined definitions
#define MPU6050_MOT_COUNT_RESET MPU6050_MOT_COUNT_0

// Combined definitions for the FF_COUNT
#define MPU6050_FF_COUNT_0 (0)
#define MPU6050_FF_COUNT_1 (bit(MPU6050_FF_COUNT0))
#define MPU6050_FF_COUNT_2 (bit(MPU6050_FF_COUNT1))
#define MPU6050_FF_COUNT_3 (bit(MPU6050_FF_COUNT1)|bit(MPU6050_FF_COUNT0))

// Alternative names for the combined definitions
#define MPU6050_FF_COUNT_RESET MPU6050_FF_COUNT_0

// Combined definitions for the ACCEL_ON_DELAY
#define MPU6050_ACCEL_ON_DELAY_0 (0)
#define MPU6050_ACCEL_ON_DELAY_1 (bit(MPU6050_ACCEL_ON_DELAY0))
#define MPU6050_ACCEL_ON_DELAY_2 (bit(MPU6050_ACCEL_ON_DELAY1))
#define MPU6050_ACCEL_ON_DELAY_3 (bit(MPU6050_ACCEL_ON_DELAY1)|bit(MPU6050_ACCEL_ON_DELAY0))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Alternative names for the ACCEL_ON_DELAY
#define MPU6050_ACCEL_ON_DELAY_0MS MPU6050_ACCEL_ON_DELAY_0
#define MPU6050_ACCEL_ON_DELAY_1MS MPU6050_ACCEL_ON_DELAY_1
#define MPU6050_ACCEL_ON_DELAY_2MS MPU6050_ACCEL_ON_DELAY_2
#define MPU6050_ACCEL_ON_DELAY_3MS MPU6050_ACCEL_ON_DELAY_3

// USER_CTRL Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_SIG_COND_RESET MPU6050_D0
#define MPU6050_I2C_MST_RESET  MPU6050_D1
#define MPU6050_FIFO_RESET     MPU6050_D2
#define MPU6050_I2C_IF_DIS     MPU6050_D4  // must be 0 for MPU-6050
#define MPU6050_I2C_MST_EN     MPU6050_D5
#define MPU6050_FIFO_EN        MPU6050_D6

// PWR_MGMT_1 Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_CLKSEL0      MPU6050_D0
#define MPU6050_CLKSEL1      MPU6050_D1
#define MPU6050_CLKSEL2      MPU6050_D2
#define MPU6050_TEMP_DIS     MPU6050_D3  // 1: disable temperature sensor
#define MPU6050_CYCLE        MPU6050_D5  // 1: sample and sleep
#define MPU6050_SLEEP        MPU6050_D6  // 1: sleep mode
#define MPU6050_DEVICE_RESET MPU6050_D7  // 1: reset to default values

// Combined definitions for the CLKSEL
#define MPU6050_CLKSEL_0 (0)
#define MPU6050_CLKSEL_1 (bit(MPU6050_CLKSEL0))
#define MPU6050_CLKSEL_2 (bit(MPU6050_CLKSEL1))
#define MPU6050_CLKSEL_3 (bit(MPU6050_CLKSEL1)|bit(MPU6050_CLKSEL0))
#define MPU6050_CLKSEL_4 (bit(MPU6050_CLKSEL2))
#define MPU6050_CLKSEL_5 (bit(MPU6050_CLKSEL2)|bit(MPU6050_CLKSEL0))
#define MPU6050_CLKSEL_6 (bit(MPU6050_CLKSEL2)|bit(MPU6050_CLKSEL1))
#define MPU6050_CLKSEL_7 (bit(MPU6050_CLKSEL2)|bit(MPU6050_CLKSEL1)|bit(MPU6050_CLKSEL0))

// Alternative names for the combined definitions
#define MPU6050_CLKSEL_INTERNAL  MPU6050_CLKSEL_0
#define MPU6050_CLKSEL_X         MPU6050_CLKSEL_1
#define MPU6050_CLKSEL_Y         MPU6050_CLKSEL_2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define MPU6050_CLKSEL_Z         MPU6050_CLKSEL_3
#define MPU6050_CLKSEL_EXT_32KHZ  MPU6050_CLKSEL_4
#define MPU6050_CLKSEL_EXT_19_2MHZ MPU6050_CLKSEL_5
#define MPU6050_CLKSEL_RESERVED   MPU6050_CLKSEL_6
#define MPU6050_CLKSEL_STOP       MPU6050_CLKSEL_7

// PWR_MGMT_2 Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_STBY_ZG      MPU6050_D0
#define MPU6050_STBY_YG      MPU6050_D1
#define MPU6050_STBY_XG      MPU6050_D2
#define MPU6050_STBY_ZA      MPU6050_D3
#define MPU6050_STBY_YA      MPU6050_D4
#define MPU6050_STBY_XA      MPU6050_D5
#define MPU6050_LP_WAKE_CTRL0 MPU6050_D6
#define MPU6050_LP_WAKE_CTRL1 MPU6050_D7

// Combined definitions for the LP_WAKE_CTRL
#define MPU6050_LP_WAKE_CTRL_0 (0)
#define MPU6050_LP_WAKE_CTRL_1 (bit(MPU6050_LP_WAKE_CTRL0))
#define MPU6050_LP_WAKE_CTRL_2 (bit(MPU6050_LP_WAKE_CTRL1))
#define MPU6050_LP_WAKE_CTRL_3 (bit(MPU6050_LP_WAKE_CTRL1)|bit(MPU6050_LP_WAKE_CTRL0))

// Alternative names for the combined definitions
// The names uses the Wake-up Frequency.
#define MPU6050_LP_WAKE_1_25HZ MPU6050_LP_WAKE_CTRL_0
#define MPU6050_LP_WAKE_2_5HZ  MPU6050_LP_WAKE_CTRL_1
#define MPU6050_LP_WAKE_5HZ    MPU6050_LP_WAKE_CTRL_2
#define MPU6050_LP_WAKE_10HZ   MPU6050_LP_WAKE_CTRL_3


// Default I2C address for the MPU-6050 is 0x68.
// But only if the AD0 pin is low.
// Some sensor boards have AD0 high, and the
// I2C address thus becomes 0x69.
#define MPU6050_I2C_ADDRESS 0x69


// Declaring an union for the registers and the axis values.
// The byte order does not match the byte order of
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// the compiler and AVR chip.
// The AVR chip (on the Arduino board) has the Low Byte
// at the lower address.
// But the MPU-6050 has a different order: High Byte at
// lower address, so that has to be corrected.
// The register part "reg" is only used internally,
// and are swapped in code.
typedef union accel_t_gyro_union
{
  struct
  {
    uint8_t x_accel_h;
    uint8_t x_accel_l;
    uint8_t y_accel_h;
    uint8_t y_accel_l;
    uint8_t z_accel_h;
    uint8_t z_accel_l;
    uint8_t t_h;
    uint8_t t_l;
    uint8_t x_gyro_h;
    uint8_t x_gyro_l;
    uint8_t y_gyro_h;
    uint8_t y_gyro_l;
    uint8_t z_gyro_h;
    uint8_t z_gyro_l;
  } reg;
  struct
  {
    int16_t x_accel;
    int16_t y_accel;
    int16_t z_accel;
    int16_t temperature;
    int16_t x_gyro;
    int16_t y_gyro;
    int16_t z_gyro;
  } value;
};


void setup()
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  int error;
  uint8_t c;



   Rtc.Begin();
   // RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__);
// Rtc.SetDateTime(compiled);
    Rtc.Enable32kHzPin(false);
    Rtc.SetSquareWavePin(DS3231SquareWavePin_ModeNone);

  // Initialize the 'Wire' class for the I2C-bus.

SD.begin();
Wire.begin();
  Serial.begin(115200);



  // default at power-up:
  //    Gyro at 250 degrees second
  //    Acceleration at 2g
  //    Clock source at internal 8MHz
  //    The device is in sleep mode.
  //

  error = MPU6050_read (MPU6050_WHO_AM_I, &c, 1);
  Serial.print(F("WHO_AM_I : "));
  Serial.print(c,HEX);
  Serial.print(F(", error = "));
  Serial.println(error,DEC);

  // According to the datasheet, the 'sleep' bit
  // should read a '1'.
  // That bit has to be cleared, since the sensor
  // is in sleep mode at power-up.
  error = MPU6050_read (MPU6050_PWR_MGMT_1, &c, 1);
  Serial.print(F("PWR_MGMT_1 : "));
  Serial.print(c,HEX);
  Serial.print(F(", error = "));
  Serial.println(error,DEC);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  // Clear the 'sleep' bit to start the sensor.
  MPU6050_write_reg (MPU6050_PWR_MGMT_1, 0);
}


void loop()
{
  int error;
  double dT;
  accel_t_gyro_union accel_t_gyro;


  //Serial.println(F(""));
 // Serial.println(F("MPU-6050"));

  // Read the raw values.
  // Read 14 bytes at once,
  // containing acceleration, temperature and gyro.
  // With the default settings of the MPU-6050,
  // there is no filter enabled, and the values
  // are not very stable.
  error = MPU6050_read (MPU6050_ACCEL_XOUT_H, (uint8_t *) &accel_t_gyro, sizeof(accel_t_gyro));
  //Serial.print(F("Read accel, temp and gyro, error = "));
 // Serial.println(error,DEC);


  // Swap all high and low bytes.
  // After this, the registers values are swapped,
  // so the structure name like x_accel_l does no
  // longer contain the lower byte.
  uint8_t swap;
  #define SWAP(x,y) swap = x; x = y; y = swap

  SWAP (accel_t_gyro.reg.x_accel_h, accel_t_gyro.reg.x_accel_l);
  SWAP (accel_t_gyro.reg.y_accel_h, accel_t_gyro.reg.y_accel_l);
  SWAP (accel_t_gyro.reg.z_accel_h, accel_t_gyro.reg.z_accel_l);
  SWAP (accel_t_gyro.reg.t_h, accel_t_gyro.reg.t_l);
  SWAP (accel_t_gyro.reg.x_gyro_h, accel_t_gyro.reg.x_gyro_l);
  SWAP (accel_t_gyro.reg.y_gyro_h, accel_t_gyro.reg.y_gyro_l);
  SWAP (accel_t_gyro.reg.z_gyro_h, accel_t_gyro.reg.z_gyro_l);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 // Print the raw acceleration values
 /*
RtcDateTime now = Rtc.GetDateTime();
{
 if(jjj ==0)

{
 k = now.Second();
 jjj = 1;
}

{
 if( k != now.Second())
 {
   k = now.Second();

 RtcDateTime now = Rtc.GetDateTime();
 Serial.print(now.Year(), DEC);
 Serial.print('/');
 Serial.print(now.Month(), DEC);
 Serial.print('/');
 Serial.print(now.Day(), DEC);
 Serial.print(' ');
 Serial.print(now.Hour(), DEC);
 Serial.print(':');
 Serial.print(now.Minute(), DEC);
 Serial.print(':');
 Serial.print(now.Second(), DEC);


 delay(0);
}}}*/



 Serial.print(F(","));
 Serial.print(accel_t_gyro.value.x_accel, DEC);
 Serial.print(F(","));
 Serial.print(accel_t_gyro.value.y_accel, DEC);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Serial.print(F(","));
  Serial.print(accel_t_gyro.value.z_accel, DEC);



  // The temperature sensor is -40 to +85 degrees Celsius.
  // It is a signed integer.
  // According to the datasheet:
  //   340 per degrees Celsius, -512 at 35 degrees.
  // At 0 degrees: -512 - (340 * 35) = -12412



  // Print the raw gyro values.

 Serial.print(F(","));
  Serial.print(accel_t_gyro.value.x_gyro, DEC);
  Serial.print(F(","));
  Serial.print(accel_t_gyro.value.y_gyro, DEC);
  Serial.print(F(","));
  Serial.print(accel_t_gyro.value.z_gyro, DEC);
  Serial.print(F(","));
  Serial.println(F(""));


#define countof(a) (sizeof(a) / sizeof(a[0]))

{
      myFile = SD.open("/2.txt", FILE_APPEND);



       while(myFile) {
        RtcDateTime now = Rtc.GetDateTime();
        {
  if(jjj ==0)

{
  k = now.Second();
  jjj = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

{
  if( k != now.Second())
  {
     k = now.Second();

  RtcDateTime now = Rtc.GetDateTime();

  myFile.print(now.Month(), DEC);
  myFile.print('/');
  myFile.print(now.Day(), DEC);
  myFile.print(' ');
  myFile.print(now.Hour(), DEC);
  myFile.print(':');
  myFile.print(now.Minute(), DEC);
  myFile.print(':');
  myFile.print(now.Second(), DEC);


  delay(0);
}}}


  myFile.print(F(","));
  myFile.print(accel_t_gyro.value.x_accel, DEC);
  myFile.print(F(","));
  myFile.print(accel_t_gyro.value.y_accel, DEC);
  myFile.print(F(","));
  myFile.print(accel_t_gyro.value.z_accel, DEC);

  myFile.print(F(","));
  myFile.print(accel_t_gyro.value.x_gyro, DEC);
  myFile.print(F(", "));
  myFile.print(accel_t_gyro.value.y_gyro, DEC);
  myFile.print(F(", "));
  myFile.print(accel_t_gyro.value.z_gyro, DEC);
  //Serial.print(F(", "));
  myFile.println(F(""));
   myFile.close();
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  delay(0);}
}


// ----------------------------
// MPU6050_read
//
// This is a common function to read multiple bytes
// from an I2C device.
//
// It uses the boolean parameter for Wire.endTransMission()
// to be able to hold or release the I2C-bus.
// This is implemented in Arduino 1.0.1.
//
// Only this function is used to read.
// There is no function for a single byte.
//
int MPU6050_read(int start, uint8_t *buffer, int size)
{
  int i, n, error;

  Wire.beginTransmission(MPU6050_I2C_ADDRESS);
  n = Wire.write(start);
  if (n != 1)
    return (-10);

  n = Wire.endTransmission(false);    // hold the I2C-bus
  if (n != 0)
    return (n);

  // Third parameter is true: relase I2C-bus after data is read.
  Wire.requestFrom(MPU6050_I2C_ADDRESS, size, true);
  i = 0;
  while(Wire.available() && i<size)
  {
    buffer[i++]=Wire.read();
  }
  if ( i != size)
    return (-11);

  return (0);  // return : no error
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}


// ----------------------------
// MPU6050_write
//
// This is a common function to write multiple bytes to an I2C device.
//
// If only a single register is written,
// use the function MPU_6050_write_reg().
//
// Parameters:
//   start : Start address, use a define for the register
//   pData : A pointer to the data to write.
//   size  : The number of bytes to write.
//
// If only a single register is written, a pointer
// to the data has to be used, and the size is
// a single byte:
//   int data = 0;       // the data to write
//   MPU6050_write (MPU6050_PWR_MGMT_1, &c, 1);
//
int MPU6050_write(int start, const uint8_t *pData, int size)
{
  int n, error;

  Wire.beginTransmission(MPU6050_I2C_ADDRESS);
  n = Wire.write(start);        // write the start address
  if (n != 1)
    return (-20);

  n = Wire.write(pData, size);  // write data bytes
  if (n != size)
    return (-21);

  error = Wire.endTransmission(true); // release the I2C-bus
  if (error != 0)
    return (error);

  return (0);         // return : no error
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// ----------------------------
// MPU6050_write_reg
//
// An extra function to write a single register.
// It is just a wrapper around the MPU_6050_write()
// function, and it is only a convenient function
// to make it easier to write a single register.
//
int MPU6050_write_reg(int reg, uint8_t data)
{
  int error;

  error = MPU6050_write(reg, &data, 1);

  return (error);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การบันทึกข้อมูลแบบ Bluetooth

```
// I2C device class (I2Cdev) demonstration Arduino sketch for MPU6050 class

// 10/7/2011 by Jeff Rowberg <jeff@rowberg.net>

// Updates should (hopefully) always be available at https://github.com/jrowberg/i2cdevlib

//

// Changelog:

//      2013-05-08 - added multiple output formats

//                 - added seamless Fastwire support

//      2011-10-07 - initial release

// I2Cdev and MPU6050 must be installed as libraries, or else the .cpp/.h files

// for both classes must be in the include path of your project

#include "I2Cdev.h"

#include "MPU6050.h"

// Arduino Wire library is required if I2Cdev I2CDEV_ARDUINO_WIRE implementation

// is used in I2Cdev.h

#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE

    #include "Wire.h"

    #include <SPI.h>

#include "SdFat.h"

#include <RTClib.h>



#endif

RTC_DS3231 RTC;

// class default I2C address is 0x68
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// specific I2C addresses may be passed as a parameter here
// AD0 low = 0x68 (default for InvenSense evaluation board)
// AD0 high = 0x69
//MPU6050 accelgyro;
MPU6050 accelgyro(0x69); // <-- use for AD0 high

int16_t ax, ay, az;
int16_t gx, gy, gz;
SdFat SD;

const uint32_t RATE_KB_PER_SEC = 100;

const uint32_t TEST_TIME_SEC = 100;
const uint32_t DOT_TIME_MS = 5000UL;

unsigned long ShowTime;
File myFile;
int pinCS = 4; // Pin 10 on Arduino Uno
int jjj = 0;

// uncomment "OUTPUT_READABLE_ACCELGYRO" if you want to see a tab-separated
// list of the accel X/Y/Z and then gyro X/Y/Z values in decimal. Easy to read,
// not so easy to parse, and slow(er) over UART.
#define OUTPUT_READABLE_ACCELGYRO

// uncomment "OUTPUT_BINARY_ACCELGYRO" to send all 6 axes of data as 16-bit
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// binary, one right after the other. This is very fast (as fast as possible
// without compression or data loss), and easy to parse, but impossible to read
// for a human.
//#define OUTPUT_BINARY_ACCELGYRO


#define LED_PIN 13
bool blinkState = false;
uint8_t k;


void setup() {
RTC.begin();

//RTC.adjust(DateTime(__DATE__, __TIME__));

 if (! RTC.isrunning()) {
   Serial.println("RTC is NOT running!");
   // following line sets the RTC to the date & time this sketch was compiled
   //RTC.adjust(DateTime(__DATE__, __TIME__));
 }


   // configure Arduino LED pin for output
   pinMode(LED_PIN, OUTPUT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pinMode(pinCS, OUTPUT);

  if (SD.begin())

{

  Serial.println("SD card is ready to use.");

} else

{

  Serial.println("SD card initialization failed");

  return;

}


  // join I2C bus (I2Cdev library doesn't do this automatically)

  #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE

     Wire.begin();

  #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE

     Fastwire::setup(400, true);

  #endif


  // initialize serial communication

  // (38400 chosen because it works as well at 8MHz as it does at 16MHz, but

  // it's really up to you depending on your project)

  Serial.begin(115200);

Wire.setClock(400000UL); // Set I2C frequency to 400kHz

  // initialize device

//  Serial.println("Initializing I2C devices...");

  accelgyro.initialize();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  // verify connection
 // Serial.println("Testing device connections...");
 // Serial.println(accelgyro.testConnection() ? "MPU6050 connection successful" : "MPU6050 connection failed");

  // use the code below to change accel/gyro offset values

  //Serial.println("Updating internal sensor offsets...");
  // -76  -2359     1688     0        0        0

  accelgyro.setXAccelOffset(-2600);
  accelgyro.setYAccelOffset(2300);
  accelgyro.setZAccelOffset(-900);
  accelgyro.setXGyroOffset(0);
  accelgyro.setYGyroOffset(0);
  accelgyro.setZGyroOffset(0);
 // Serial.print(accelgyro.getXAccelOffset()); Serial.print("\t"); // -76
 // Serial.print(accelgyro.getYAccelOffset()); Serial.print("\t"); // -2359
 // Serial.print(accelgyro.getZAccelOffset()); Serial.print("\t"); // 1688
 // Serial.print(accelgyro.getXGyroOffset()); Serial.print("\t"); // 0
 // Serial.print(accelgyro.getYGyroOffset()); Serial.print("\t"); // 0
 // Serial.print(accelgyro.getZGyroOffset()); Serial.print("\t"); // 0
 // Serial.print("\n");

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void loop() {


    // read raw accel/gyro  measurements from device
    accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);

    // these methods (and a few others) are also available
    //accelgyro.getAcceleration(&ax,  &ay, &az);
    //accelgyro.getRotation(&gx,  &gy, &gz);

  // #ifdef  OUTPUT_READABLE_ACCELGYRO
        // display tab-separated accel/gyro  x/y/z values




    //else{
    //Serial.println("       ");
    //}

  //delay(0);

        Serial.print(",");
        Serial.print(ax); Serial.print(",");
        Serial.print(ay); Serial.print(",");
        Serial.print(az); Serial.print(",");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Serial.print(gx); Serial.print(",");
        Serial.print(gy); Serial.print(",");
        Serial.println(gz);

/*myFile = SD.open("TESTSD.txt", FILE_WRITE);
  if (myFile) {

    DateTime now = RTC.now();
if(jjj ==0)
{
  k = now.second();
  jjj = 1;
}

{
  if( k != now.second())
  {
    k = now.second();
 // myFile.print(now.year(), DEC);
   myFile.print(now.month(), DEC);
  myFile.print('/');
  myFile.print(now.day(), DEC);
  myFile.print(' ');
  myFile.print(now.hour(), DEC);
  myFile.print(':');
  myFile.print(now.minute(), DEC);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
myFile.print(':');
myFile.print(now.second(), DEC);
}
myFile.print(',');
  myFile.print(ax);
  myFile.print(",");

  myFile.print(ay);  myFile.print(",");
  myFile.print(az);  myFile.print(",");
  myFile.print(gx);   myFile.print(",");
  myFile.print(gy);   myFile.print(",");
  myFile.println(gz);
  // myFile.print("\t");
  myFile.close(); // close the file
}}
// if the file didn't open, print an error:
else {
  Serial.println("error  opening .txt");

}
  #endif*/

  #ifdef OUTPUT_BINARY_ACCELGYRO
      Serial.write((uint8_t)(ax >> 8)); Serial.write((uint8_t)(ax & 0xFF));
      Serial.write((uint8_t)(ay >> 8)); Serial.write((uint8_t)(ay & 0xFF));
      Serial.write((uint8_t)(az >> 8)); Serial.write((uint8_t)(az & 0xFF));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Serial.write((uint8_t)(gx >> 8)); Serial.write((uint8_t)(gx & 0xFF));
        Serial.write((uint8_t)(gy >> 8)); Serial.write((uint8_t)(gy & 0xFF));
        Serial.write((uint8_t)(gz >> 8)); Serial.write((uint8_t)(gz & 0xFF));
    #endif

    // blink LED to indicate activity
    blinkState = !blinkState;
    digitalWrite(LED_PIN, blinkState);
    delay(1);

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้