



รายงานสหกิจศึกษาฉบับสมบูรณ์

เครื่องบันทึกค่าสั่นสะเทือนในอุปกรณ์ทดสอบหัวอ่าน

Mini Tester Equipment Shock Detector and Logger

นายณัฏชยุต ภูริคณานนท์

สาขาวิชาวิศวกรรมสารสนเทศ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2562

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา	เครื่องมือวัดและบันทึกค่าแรงกระทำในอุปกรณ์ทดสอบหัวอ่าน		
ชื่อ – นามสกุล	นายณัทชยุต	ภูริคณานนท์	
คณะ	วิศวกรรมศาสตร์	ภาควิชา	สารสนเทศ
ชื่อ – นามสกุล อาจารย์นิเทศ	ผศ.บุญยชนะ ภูระหงษ์		
ชื่อ – นามสกุล ผู้นิเทศงาน	นายเอกพัฒน์ เอกนรพันธ์		
สถานประกอบการ	บริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัด		

บทคัดย่อ

อุปกรณ์ทดสอบหัวอ่านประกอบไปด้วยชิ้นส่วนต่างๆ ซึ่งมีขนาดเล็กและไวต่อแรงสั่นสะเทือน ในขณะที่เคลื่อนที่ เกิดแรงสั่นสะเทือนขึ้น ซึ่งเป็นสาเหตุทำให้หัวอ่านและชิ้นส่วนต่างๆเกิดความเสียหาย

โครงการเครื่องมือวัดแรงสั่นสะเทือนในอุปกรณ์ทดสอบหัวอ่านนี้ เป็นเครื่องมือวัดแรงสั่นจากการเคลื่อนที่ที่เกิดขึ้นในเครื่องจักรกระทำต่ออุปกรณ์ทดสอบหัวอ่าน และบันทึกข้อมูลเพื่อนำมาวิเคราะห์หาจุดที่มีแรงกระทำสูงสุด จากนั้นส่งข้อมูลที่ได้ไปยังฝ่ายที่เกี่ยวข้องสำหรับการปรับปรุง ลดแรงสั่นสะเทือนลง

โครงการนี้ขึ้นมาจากพัฒนาจากโปรแกรม Arduino IDE ที่ใช้ภาษา C++ และอุปกรณ์อิเล็กทรอนิกส์ที่สำคัญคือ Inertial measurement unit

คำสำคัญ : อุปกรณ์ทดสอบหัวอ่าน, ตรวจจับ, บันทึก, ความเสียหาย

Cooperative Title : Mini Tester Equipment Shock Detector and Logger

Student intern name : Natchayut Purikananond

Faculty name : Engineering Department : Information Technology

Advisor name : Asst.Prof. Boonchana Purahong

Mentor name : Mr. Akkhatpat Aknoraphan

Company : Seagate Technology (Thailand) ltd.

Abstract

Mini tester equipment has very small and sensitive components. When we transport the equipment, Transportation will create shock that can cause damage to equipment.

Mini Tester Equipment Shock Detector and Logger is vibration measurement and logging device in read tester equipment when testing in machine after that we will send the data to analysis for improve and reduce vibration in process.

So I started develop this project by using C++ program language and Arduino IDE for develop program and using Inertial measurement unit for hardware develop.

Keywords : Mini tester equipment, Detect, Logging, Analysis

กิตติกรรมประกาศ

รายงานสหกิจศึกษาฉบับนี้สามารถเสร็จสมบูรณ์ตามเป้าหมายได้เพราะได้รับความช่วยเหลือ และ คำแนะนำต่างๆที่เป็นประโยชน์ต่อโครงการนี้อย่างยิ่ง จากผู้มีพระคุณต่างๆ อาทิ

คุณกวิน เลขานนท์, คุณเอกพัฒน์ เอกนรพันธ์, คุณชูศักดิ์ ชูแสงสุนทร และพี่ในทีม Software Integration ของแผนก Advance Manufacturing Engineering ที่คอยสนับสนุน ให้คำปรึกษา และช่วยเหลือเรื่องต่างตลอดระยะเวลาที่มาสหกิจศึกษา

คณะอาจารย์คณะวิศวกรรมศาสตร์ จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่มอบความรู้มาตลอด 4 ปี

ผศ.ไพศาล สิทธิโยภาสกุล และผศ.บุญยชนะ ภูระหงษ์ อาจารย์ที่ปรึกษา ที่ให้ข้อเสนอแนะ ข้อคิด และคอยติดต่อ และดูแลตลอดระยะเวลาสหกิจศึกษา

เพื่อนทุกคนที่ทำสหกิจศึกษาที่บริษัท ซีเกท เทคโนโลยี(ประเทศไทย)จำกัด ที่คอยอยู่ด้วยกันมา ตลอดเวลาจนจบ

และขอขอบคุณครอบครัว และบุคคลอื่นๆที่ผมไม่ได้กล่าวถึง ณ ที่นี้

นายณัฏชยุต ภูริคณานนท์

ผู้จัดทำ

สารบัญ

บทคัดย่อ.....	i
Abstract	ii
กิตติกรรมประกาศ	iii
สารบัญตาราง	vi
สารบัญรูปภาพ.....	vii
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมา และความสำคัญ.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตการวิจัย	2
1.4 วิธีการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 IMU (Inertial measurement unit)	3
2.2 Kalman Filter	7
2.3 I ² C Protocol.....	9
2.4 SPI Protocol	11
บทที่ 3 วิธีการดำเนินงาน.....	16
3.1 รับทราบความต้องการของระบบ	16
บทที่ 4 ผลการดำเนินงาน	40
4.1 ผลการทดสอบวิธีการวัด.....	40
4.2 ผลการทดสอบในเปรียบเทียบข้อมูล.....	42

4.3 ผลการทดสอบในเครื่องจักร	43
บทที่ 5 สรุปผล และข้อเสนอแนะ	46
5.1 สรุปผลการดำเนินงาน	46
5.2 ปัญหาและอุปสรรคในการดำเนินงาน	46
5.3 วิธีแก้ไขปัญหา	46
5.4 ข้อเสนอแนะ	47
บรรณานุกรม	48
ภาคผนวก	50



สารบัญตาราง

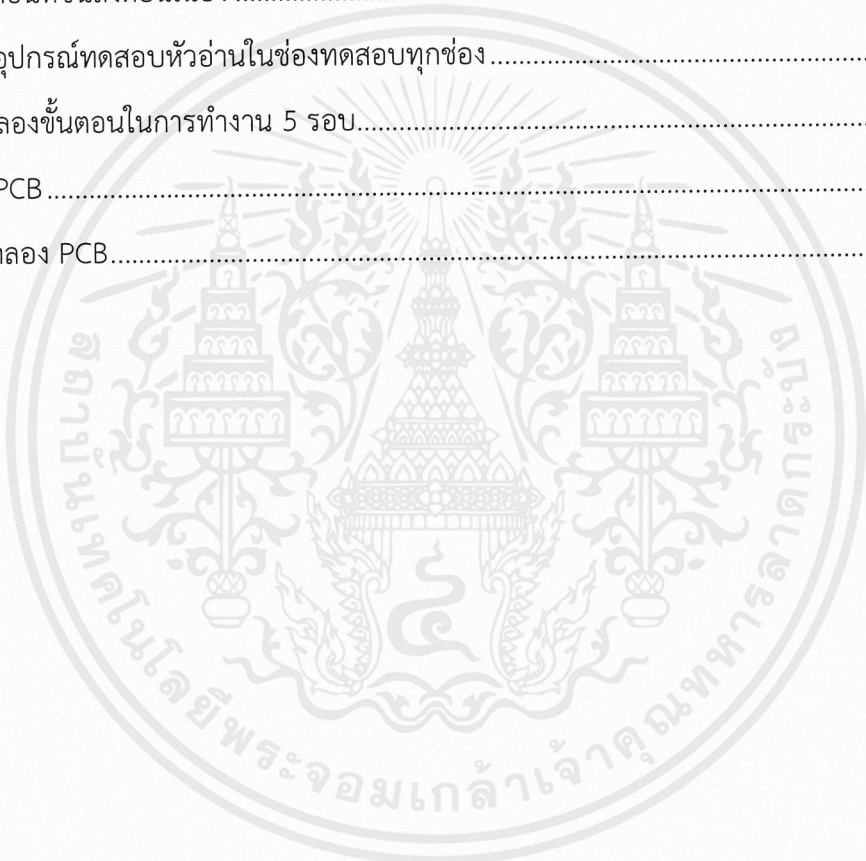
ตารางที่ 1 จุดการเชื่อมต่อ	20
----------------------------------	----



สารบัญรูปภาพ

รูปที่ 1 แผ่น Piezoelectric ที่ใช้ในการเปลี่ยนสัญญาณเสียงเป็นคลื่นเสียง	4
รูปที่ 2 หลักการทำงานของ Piezoelectric accelerometers.....	4
รูปที่ 3 วงจรภายใน Accelerometer	5
รูปที่ 4 หลักการทำงานของ Gyroscope sensor	5
รูปที่ 5 แสดงความสัมพันธ์ของสมการ	8
รูปที่ 6 แสดงการเชื่อมต่ออุปกรณ์แบบ I ² C	9
รูปที่ 7 รูปแบบสัญญาณในโปรโตคอล I ² C	10
รูปที่ 8 การเชื่อมต่อสื่อสารแบบ SPI ระหว่างอุปกรณ์.....	11
รูปที่ 9 การสื่อสารแบบ SPI ระหว่างอุปกรณ์ Master กับ Slave หลายตัว	12
รูปที่ 10 การรับส่งข้อมูลเมื่อเลือก CPOL = 0 และ CPHA = 0.....	14
รูปที่ 11 การรับส่งข้อมูลเมื่อเลือก CPOL = 0 และ CPHA = 1.....	14
รูปที่ 12 การรับส่งข้อมูลเมื่อเลือก CPOL = 1 และ CPHA = 0.....	15
รูปที่ 13 การรับส่งข้อมูลเมื่อเลือก CPOL = 1 และ CPHA = 1.....	15
รูปที่ 14 GY - 86	17
รูปที่ 15 Arduino Mega 2560 R3	17
รูปที่ 16 Micro SD Card SPI Reader Module.....	18
รูปที่ 17 PCF 8563	18
รูปที่ 18 การเชื่อมต่ออุปกรณ์ต่างๆ	19
รูปที่ 19 การเชื่อมต่ออุปกรณ์	20
รูปที่ 20 9DoF Razor IMU M0	28
รูปที่ 21 แบบจำลองการเชื่อมต่ออุปกรณ์	29
รูปที่ 22 การเชื่อมต่ออุปกรณ์ทั้งหมด.....	30
รูปที่ 23 การแสดงผลการวัด.....	38
รูปที่ 24 อุปกรณ์ Inertial Measurement Unit MTi - 30 - 2A8G4	39
รูปที่ 25 คุณสมบัติของ MTi - 30 - 2A8G4.....	39

รูปที่ 26 เปรียบเทียบการตรวจวัดค่าความเร่งแกน X.....	40
รูปที่ 27 เปรียบเทียบการตรวจวัดค่าความเร่งแกน Y.....	41
รูปที่ 28 เปรียบเทียบการตรวจวัดค่าความเร่งแกน Z.....	41
รูปที่ 29 เปรียบเทียบอุปกรณ์กับ MTi – 30 แกน X.....	42
รูปที่ 30 เปรียบเทียบอุปกรณ์กับ MTi – 30 แกน Y.....	42
รูปที่ 31 เปรียบเทียบอุปกรณ์กับ MTi – 30 แกน Z.....	43
รูปที่ 32 การเคลื่อนที่ขึ้นลงตอนเนื่อง	44
รูปที่ 33 การใส่อุปกรณ์ทดสอบหัวอ่านในช่องทดสอบทุกช่อง	44
รูปที่ 34 การจำลองขั้นตอนในการทำงาน 5 รอบ.....	45
รูปที่ 35 ดีไซน์ PCB	47
รูปที่ 36 แบบจำลอง PCB.....	47



บทที่ 1 บทนำ

ในบทนี้เป็นส่วนการเกริ่นที่มาของโครงการนี้ แบ่งเป็นหัวข้อดังต่อไปนี้คือ ความเป็นมาและความสำคัญ วัตถุประสงค์ ขอบเขตการวิจัย วิธีการดำเนินงาน ประโยชน์ที่คาดว่าจะได้รับ

1.1 ความเป็นมา และความสำคัญ

บริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัด ได้จัดโครงการสหกิจศึกษาระหว่าง บริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัด กับ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทำให้เกิดแนวคิดโครงการที่มอบหมายให้นักศึกษา จึงมอบหมายงานให้นักศึกษาทำการศึกษาค้นคว้าเกี่ยวกับเทคโนโลยี และการวัดที่สามารถมาประยุกต์ใช้ในการวัดแรงสั่นสะเทือน เพื่อให้ นักศึกษามีความรู้ความสามารถในการพัฒนาเครื่องมือวัดได้อย่างมีประสิทธิภาพ

โรงงานซีเกท เทพารักษ์ รับผิดชอบในการผลิต และส่งออกหัวอ่านไปประกอบในโรงงานอื่นๆ หลังจากผลิตหัวอ่านเสร็จ ชิ้นส่วนนั้นถูกทดสอบโดยใส่ไว้ในอุปกรณ์ทดสอบหัวอ่าน และนำไปทดสอบผ่านเครื่องจักร

อุปกรณ์ทดสอบหัวอ่านมีชิ้นส่วนต่างๆที่มีขนาดเล็ก และอ่อนไหวง่าย ขณะทำการขนส่งอุปกรณ์ทดสอบหัวอ่านในขั้นตอนต่างๆ เกิดแรงสั่นสะเทือนขึ้นมา หากเกิดแรงสั่นมากเกินไป หัวอ่านเกิดความเสียหายขึ้น

ทำให้เกิดการพัฒนาอุปกรณ์นี้ขึ้นมาโดยใช้ Inertial Measurement Unit เป็นตัวตรวจวัดแรงสั่นสะเทือนในอุปกรณ์ทดสอบหัวอ่าน เพื่อที่เราสามารถตรวจสอบแรงสั่นสะเทือน และนำข้อมูลดังกล่าวมาปรับปรุง โดยคาดว่าจะสามารถนำข้อมูลมาหาจุดที่มีแรงสั่นสะเทือนมากที่สุด และนำข้อมูลนั้นมาปรับปรุงการทำงานของเครื่องจักร

1.2 วัตถุประสงค์

- เพื่อวัดค่าแรงสั่นสะเทือนที่เกิดขึ้นในระหว่างการขนส่ง
- เพื่อหาจุดเสี่ยงที่ทำให้อุปกรณ์ทดสอบหัวอ่านเสียหาย

1.3 ขอบเขตการวิจัย

- จัดทำอุปกรณ์ครั้งนี้ มุ่งเน้นศึกษาเฉพาะการใช้ความเร่งในการเคลื่อนไหวในการตรวจวัดแรงกระทำที่เกิดขึ้น และการประยุกต์การใช้งาน Inertial Measurement Unit ในชีวิตประจำวัน
- ระยะเวลาที่ใช้ในการพัฒนาอุปกรณ์นี้ดำเนินการในภาคเรียนที่ 1 ตั้งแต่วันที่ 5 สิงหาคม 2562 ถึงวันที่ 29 ธันวาคม 2562

1.4 วิธีการดำเนินงาน

- ศึกษาฮาร์ดแวร์ของบริษัทที่เกี่ยวข้องกับโครงการนี้
- ศึกษาขั้นตอนการทำงานเครื่องจักรในสายการผลิต
- ศึกษาขั้นตอนการวัดแรงสั่นสะเทือนภายในอุปกรณ์
- ทำอุปกรณ์วัดและบันทึกแรงสั่นสะเทือนในเครื่องจักร

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- อุปกรณ์ที่พัฒนาขึ้นสามารถวัด และบันทึกค่าแรงสั่นสะเทือนในสายการผลิตได้
- เราพบจุดเสี่ยงที่ทำให้อุปกรณ์ทดสอบหัวอ่านเสียหาย

บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้เป็นส่วนของทฤษฎีที่ และงานวิจัยที่มีความเกี่ยวข้องกับโครงงานนี้ และนำมาใช้งาน แบ่งเป็นหัวข้อดังต่อไปนี้คือ IMU (Inertial measurement unit), Kalman filter, I²C Protocol, SPI Protocol

2.1 IMU (Inertial measurement unit)

IMU (Inertial measurement unit) คือ อุปกรณ์อิเล็กทรอนิกส์ใช้สำหรับการวัด และแสดงกำลังเร่ง, อัตราความเร็วเชิงมุม, และตำแหน่งทิศทางของอุปกรณ์ ณ ขณะเวลานั้น ปกติอุปกรณ์ชนิดนี้มักถูกรวมในระบบ Attitude Heading Reference System หรือ AHRS สำหรับการบังคับเครื่องบิน รวมไปถึงยานอวกาศอีกด้วย ประกอบไปด้วยอุปกรณ์อิเล็กทรอนิกส์ย่อยๆ 3 ชนิด ได้แก่

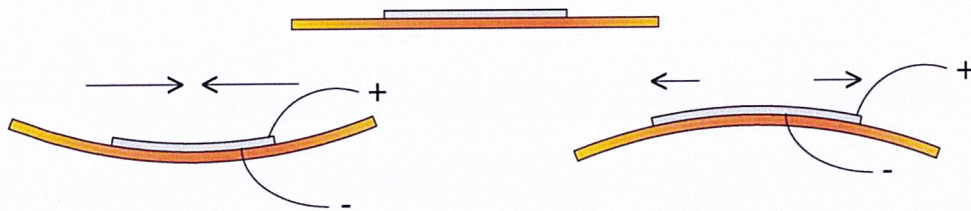
2.1.1 Accelerometer คือ อุปกรณ์สำหรับวัดค่าความเร่งจากอุปกรณ์ที่ถูกกระทำในทันทีทันใด Accelerometer มีแกนได้หลายแกน เพื่อสามารถระบุทิศทางการเร่งได้อย่างแม่นยำ หรือใช้ในการวัดแรงสั่นสะเทือน

Accelerometer ที่วางอยู่บนพื้นผิวโลกแสดงค่าแรงโน้มถ่วงโลกประมาณ 1 จี เนื่องจากพื้นผิวของโลกออกแรงแรงขึ้นไปตามปกติเมื่อเทียบกับกรอบอ้างอิงเฉื่อย เพื่อให้ได้ความเร่งใกล้เคียงกับการเคลื่อนที่ของโลกนั้น ออฟเซตแรงโน้มถ่วงต้องถูกนำออกและแก้ไขปัญหาที่เกิดขึ้นจากเอฟเฟกต์ที่เกิดจากการหมุนของโลกเมื่อเทียบกับกรอบอ้างอิงเฉื่อย

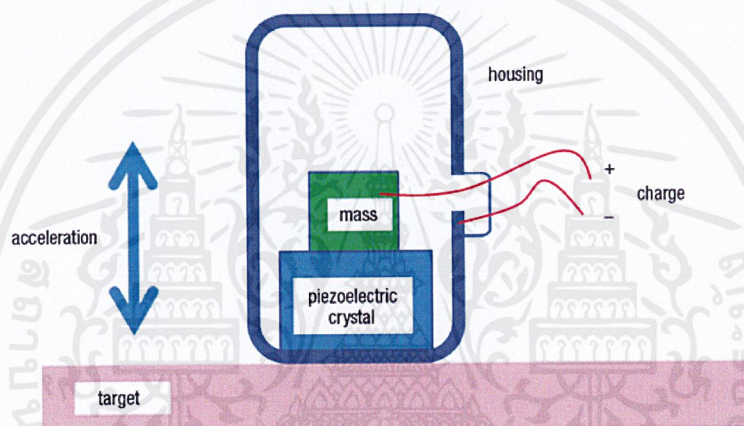
ความเร่งถูกวัดตามหน่วย SI หน่วยต่อเมตรต่อวินาทีกำลังสอง (m / s^2), หน่วย CSG Gal (Gal) หรือที่นิยมในมาตรฐานของแรงโน้มถ่วงมาตรฐาน (จี)

แบ่งได้เป็น 2 ประเภทคือ

2.1.1.1 Piezoelectric accelerometers ใช้วิธีการใช้แรงกระทำต่อวัสดุที่มีสมบัติเป็น Piezoelectric โดยวัสดุเหล่านี้เปลี่ยนพลังงานกลต่างๆเหล่านี้ให้เป็นพลังงานไฟฟ้า ในทางกลับกันเมื่อให้พลังงานไฟฟ้าแก่วัสดุที่มีคุณสมบัติแบบเดียวกัน วัสดุนั้นสามารถเปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกลได้เช่นกัน



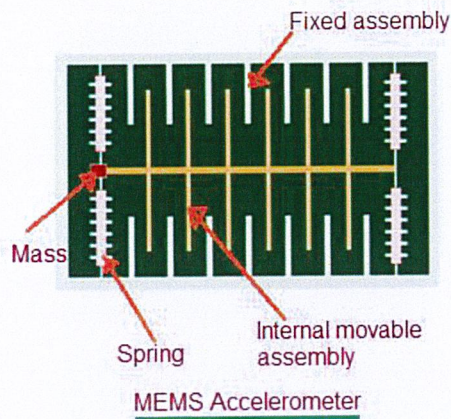
รูปที่ 1 แผ่น Piezoelectric ที่ใช้ในการเปลี่ยนสัญญาณเสียงเป็นคลื่นเสียง



รูปที่ 2 หลักการทำงานของ Piezoelectric accelerometers

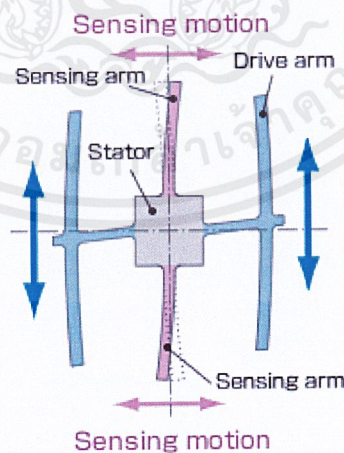
2.1.1.2 Capacitive เป็นหนึ่งในประเภทของ accelerometer ที่วัดความเร่งบนพื้นผิวโดยใช้เทคนิคการตรวจจับค่าประจุไฟฟ้าของช่องระหว่างชิ้นส่วนที่เคลื่อนที่กับชิ้นส่วนที่หยุดนิ่ง

เมื่อส่วนประกอบเหล่านี้เคลื่อนที่หรือเคลื่อนย้าย ความจุหรือพลังงานที่เกิดขึ้นถูกตรวจจับโดยเซ็นเซอร์ และแปลงความเร่งที่เกิดขึ้นให้กลายเป็นกระแสไฟฟ้า



รูปที่ 3 วงจรภายใน Accelerometer

2.1.2 Gyroscope คือ อุปกรณ์ที่อาศัยแรงเฉื่อยของล้อหมุนสำหรับวัดการเปลี่ยนแปลงตำแหน่งทิศทาง และอัตราเร็วเชิงมุมของอุปกรณ์ สามารถวัดความเร็วการหมุนได้ใน 1 2 หรือ 3 ทิศทาง ไจโรสโคปแบบ 3 แกน มีมาตรวัดความเร็ว 3 แกนเพื่อให้ระบบตรวจจับการเคลื่อนไหวมีองศาเสรีถึง 6 ระดับ ไจโรสโคปมักใช้ในเครื่องบินเพื่อช่วยระบุการวางแนวและอัตราการหมุนของเครื่องบินในแกนหมุน เมื่อเครื่องบินหมุน ไจโรสโคปทำการวัดค่าที่ไม่เป็นศูนย์จนกว่าได้แนวระนาบ ซึ่งแสดงค่าเป็นศูนย์เพื่อบอกทิศทางแนวลง



รูปที่ 4 หลักการทำงานของ Gyroscope sensor

2.1.3 Magnetometer คือ อุปกรณ์วัดทิศทาง, ความแรง, และการเปลี่ยนแปลงของสนามแม่เหล็ก ณ ขณะนั้น Magnetometers ใช้กันอย่างแพร่หลายสำหรับการวัดสนามแม่เหล็กของโลกและในการสำรวจธรณีฟิสิกส์เพื่อตรวจจับความผิดปกติของแม่เหล็กในรูปแบบต่าง ๆ ในทัศนคติของเครื่องบินและระบบการอ้างอิงส่วนหัวมักใช้เป็นข้อมูลอ้างอิง ส่วนหัว Magnetometers ยังใช้ในกองทัพเพื่อตรวจจับเรือดำน้ำ ดังนั้นบางประเทศเช่นสหรัฐอเมริกาแคนาดาและออสเตรเลียจึงจัดประเภทเครื่องวัดความไวที่มีความไวสูงกว่าเป็นเทคโนโลยีทางการทหารและควบคุมการกระจายของพวกเขา

ข้อเสียเปรียบที่สำคัญของการใช้ IMU สำหรับการนำทางคือโดยทั่วไปแล้วพวกเขาได้รับข้อผิดพลาดสะสม เนื่องจากระบบนำทางมีการรวมการเร่งความเร็วอย่างต่อเนื่องเมื่อเทียบกับเวลาในการคำนวณความเร็วและตำแหน่งข้อผิดพลาดในการวัดใด ๆ ก็ตามมีขนาดเล็ก แต่สะสมอยู่ตลอดเวลา สิ่งนี้นำไปสู่การ 'ตรีฟท์': ความแตกต่างที่เพิ่มขึ้นอย่างต่อเนื่องระหว่างที่ระบบคิดว่าตั้งอยู่และตำแหน่งที่แท้จริง เนื่องจากการรวมข้อผิดพลาดคงที่ในการเร่งความเร็วส่งผลให้เกิดข้อผิดพลาดเชิงเส้นในความเร็วและการเติบโตข้อผิดพลาดกำลังสองในตำแหน่งข้อผิดพลาดคงที่ในอัตราทัศนคติ (ไจโร) ส่งผลให้เกิดข้อผิดพลาดกำลังสองในความเร็วและการเติบโตข้อผิดพลาดลูกบาศก์ในตำแหน่ง

2.2 Kalman Filter

Kalman filter คือ ตัวประมาณค่าที่เหมาะสมที่สุด หรือกล่าวคือค่าตัวแปรที่น่าสนใจจากข้อมูลที่มมมองอื่นๆ, ทางอ้อม, หรือไม่แน่นอน ตัวแปรนี้เป็นแบบเรียกซ้ำ เพื่อให้การประมวลผลได้ใหม่ทันทีที่ข้อมูลนี้มาถึง

ถ้าสัญญาณรบกวนทั้งหมดที่มีเป็นแบบ Gaussian, ฟังก์ชัน Kalman filter สามารถลดค่าเฉลี่ยของความคลาดเคลื่อนยกกำลังสองของค่าตัวแปรที่เราประเมิน

ถ้าสัญญาณรบกวนไม่ได้เป็นแบบ Gaussian เนื่องจาก Kalman filter เป็นตัวประมาณแบบเชิงเส้นที่ดีที่สุดเพราะเป็นเพียงค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานเท่านั้น ดังนั้นตัวประมาณค่าแบบไม่เชิงเส้นอาจดีกว่า

2.1 ข้อดีของฟังก์ชัน

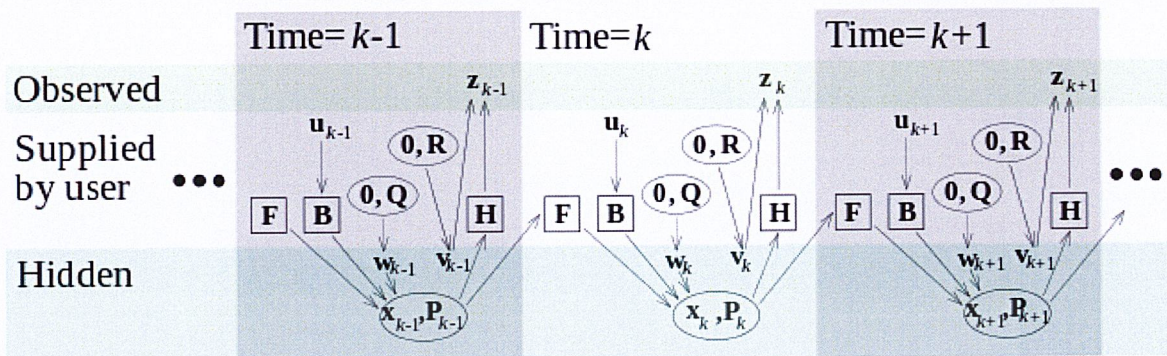
- ได้ผลลัพธ์ที่ดีในทางปฏิบัติเนื่องจากมีโครงสร้างที่ดี และเหมาะสม
- รูปแบบที่สะดวกสำหรับการประมวลผลออนไลน์แบบเรียลไทม์
- ง่ายต่อการกำหนดและนำไปใช้เมื่อความเข้าใจในพื้นฐาน
- สมการการวัดไม่จำเป็นต้องกลับด้าน

2.2 แบบจำลองระบบพลศาสตร์พื้นฐาน

ในการใช้ตัวกรองคาลมานเพื่อประเมินสถานะภายในของกระบวนการที่กำหนดลำดับของการสังเกตที่มีสัญญาณรบกวนเพียงอย่างเดียวเราต้องทำการจำลองกระบวนการตามกรอบของตัวกรองคาลมาน นี่หมายถึงการระบุเมทริกซ์ต่อไปนี้:

- F_k : รูปแบบการเปลี่ยนสถานะ
- H_k : รูปแบบการสังเกต
- Q_k : ความแปรปรวนร่วมของสัญญาณรบกวนในกระบวนการ
- R_k : ความแปรปรวนร่วมของสัญญาณรบกวนจากสังเกต
- B_k : โมเดลควบคุมอินพุตสำหรับแต่ละขั้นตอน

- k : นิยามดังรูปต่อไปนี้



รูปที่ 5 แสดงความสัมพันธ์ของสมการ

แบบจำลองต้นแบบตัวกรองคาลมาน กรอบสี่เหลี่ยมแสดงถึงเมทริกซ์ ส่วนวงรีเป็นการแทนการแจกแจงปกติหลายตัวแปร (รวมถึงค่าเฉลี่ยและความแปรปรวนเมทริกซ์ล้อมรอบ) ค่าที่ไม่ได้ถูกหุ้มไว้ให้ถือว่าเป็นเวกเตอร์ ในกรณีที่เรียกง่ายเมทริกซ์ต่างๆมีค่าคงที่ตามเวลาดังนั้นตัวห้อยถูกตรึง แต่ตัวกรองคาลมานอนุญาตให้เปลี่ยนแปลงใด ๆ ในแต่ละขั้นตอน

แบบจำลองตัวกรองคาลมานสันนิษฐานว่าสถานะที่แท้จริง ณ เวลา k ถูกพัฒนาจากสถานะที่ $(k - 1)$ ตาม

$$x_k = F_k x_{k-1} + B_k u_k + w_k$$

โดย

F_k คือ โมเดลการเปลี่ยนสถานะซึ่งใช้กับสถานะก่อนหน้า x_{k-1}

B_k คือ โมเดลควบคุมอินพุตซึ่งใช้กับเวกเตอร์ควบคุม u_k

w_k คือสัญญาณรบกวนในกระบวนการการทำงานซึ่งสันนิษฐานว่าถูกดึงมาจากการกระจายตัวแบบหลายตัวแปรเฉลี่ยศูนย์ ด้วยความแปรปรวนร่วม

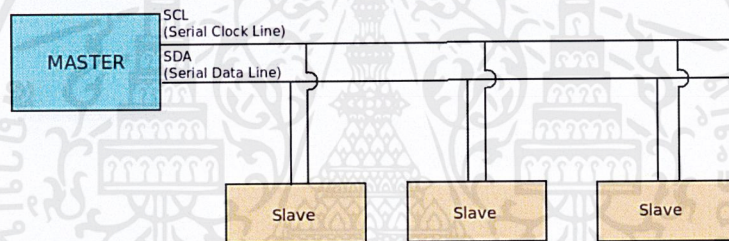
2.3 I²C Protocol

I²C หรือ Inter integrate circuit (IIC) เป็นโปรโตคอลการสื่อสารอนุกรมแบบ Synchronus มีจุดเด่นคือเป็นการเชื่อมต่อแบบระบบบัสรับ - ส่งข้อมูล จึงสามารถเชื่อมต่อกับอุปกรณ์จำนวนมากได้โดยใช้สายเพียงแค่ 2 เส้น และสามารถเชื่อมต่ออุปกรณ์ได้สูงสุด 1,008 ตัว

2.3.1 สัญญาณ

- SCL (Serial Clock Line (SCL) ส่งสัญญาณนาฬิกา เพื่อบอกว่าข้อมูลที่รับ/ส่ง ทำที่ความถี่เท่าใด ความเร็วของการรับส่งข้อมูลนั้นขึ้นอยู่กับความถี่ของสัญญาณนาฬิกาในสาย SCL สำหรับ STM32F4 สามารถส่งสัญญาณนาฬิกาที่มีความถี่ 100kHz (Standard) หรือ 400kHz (Fast)

- SDA (Serial Data Line) ใช้ทำหน้าที่ส่งข้อมูลเพียงอย่างเดียว



รูปที่ 6 แสดงการเชื่อมต่ออุปกรณ์แบบ I²C

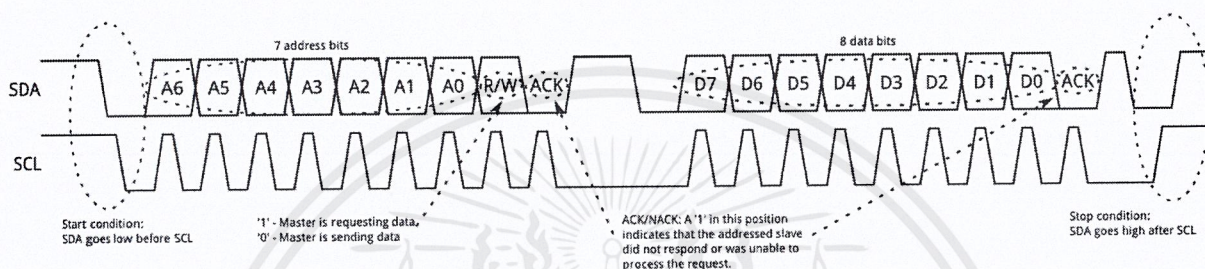
I²C เป็นโปรโตคอลที่มีความคงทนต่อสัญญาณรบกวนพอสมควรและสามารถใช้กับสายสั้น (2-3m) สำหรับระยะยาวหรือระบบที่มีอุปกรณ์จำนวนมากตัวต้านทานขนาดเล็กดีกว่า

2.3.2 ระดับของสัญญาณ

ถึงแม้ว่าอุปกรณ์บนระบบบัสไม่ได้ขับสัญญาณให้สูง I²C ก็อนุญาตให้เชื่อมต่ออุปกรณ์ที่มีแรงดันไฟฟ้า I/O ที่แตกต่างกันได้ โดยทั่วไป อุปกรณ์หนึ่งมีแรงดันไฟฟ้าสูงกว่าอุปกรณ์อื่นอาจเชื่อมต่ออุปกรณ์ทั้งสองผ่าน I²C ได้โดยไม่ต้องมีวงจรเปลี่ยนระดับในระหว่างอุปกรณ์เหล่านั้น มีเคล็ดลับดังนี้คือการเชื่อมต่อตัวต้านทานแบบดึงขึ้น กับแรงดันไฟฟ้าที่ต่ำกว่าของทั้งสอง สามารถใช้งานได้ในบางกรณีที่แรงดันไฟฟ้าต่ำกว่าสองระบบเกินแรงดันอินพุตระดับสูงของระบบไฟฟ้าแรงสูงเช่น Arduino ที่มีแรงดันไฟฟ้า 5V และ Accelerometer ที่มีแรงดันไฟฟ้า 3.3V

2.3.3 โพรโตคอล

การสื่อสารผ่าน I²C นั้นซับซ้อนกว่าการสื่อสารแบบ UART หรือ SPI เพราะการส่งสัญญาณต้องเป็นไปตามโปรโตคอลที่แน่นอนสำหรับอุปกรณ์บนบัสเพื่อรับรู้ว่าเป็นการสื่อสาร I²C ที่ถูกต้อง แต่โชคดีที่อุปกรณ์ส่วนมากจัดการกับรายละเอียดเล็กน้อยๆต่าง ๆ ให้เอง ทำให้เราสามารถจัดการการแลกเปลี่ยนข้อมูลที่เราต้องการได้อย่างราบรื่น



รูปที่ 7 รูปแบบสัญญาณในโปรโตคอล I²C

ข้อความถูกแบ่งออกเป็นเฟรม 2 แบบ คือ Address เฟรม สำหรับให้อุปกรณ์ที่เป็น Master ระบุอุปกรณ์ที่เป็น Slave และต้องส่งข้อมูลไต่ไป ซึ่งเป็นข้อมูล 8 บิตที่ส่งจากอุปกรณ์ที่เป็น Master ไปอุปกรณ์ที่เป็น Slave หรือจาก Slave ไปยัง Master ข้อมูลถูกวางในสัญญาณ SDA หลังจาก สัญญาณ SCL ถูกดึงต่ำลง และถูก Sample หลังจากสัญญาณ SCL ถูกดึงสูงขึ้น เวลาระหว่างนาฬิกาและการอ่าน / เขียนข้อมูลถูกกำหนดโดยอุปกรณ์บนระบบบัสและแตกต่างกันไปตามแต่ละชิป

Address Frame อยู่เป็นกลุ่มบิตแรกเสมอเมื่อมีการเริ่มขั้นตอนการสื่อสารใหม่ทุกครั้ง สำหรับ 7 บิตแรกของ Address Frame เป็นบิตจากการโอเวอร์คล็อกของบิตที่สำคัญที่สุด (Most significant bit หรือ MSB) จากนั้นก็ตามด้วย Read/Write Bit เพื่อบอกว่าขณะนี้ว่าข้อมูลชุดนี้กำลังถูกอ่าน (บิตเป็น 1) หรือถูกเขียนอยู่ (บิตเป็น 0)

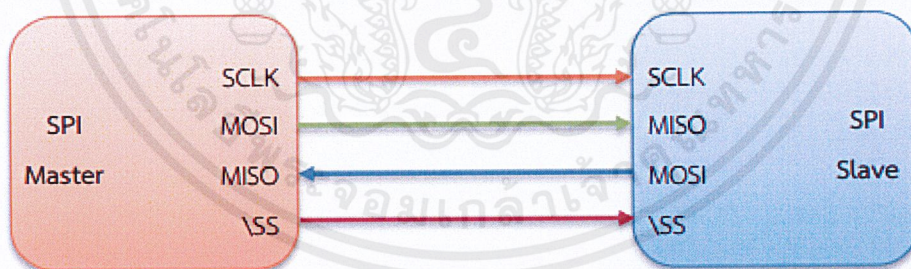
ส่วนบิตที่ 9 เป็น NACK(Negative Acknowledgement)/กับ ACK(Acknowledgement) bit โดยบิตนี้ต้องมีในทุก Address Frame หรือ Data Frame

หลังจากที่ Address Frame ถูกส่งไป Data Frame จึงถูกส่งไปภายหลัง อุปกรณ์หลังคอยสร้างสัญญาณนาฬิกาขึ้นเรื่อยๆในช่วงเวลาปกติ และข้อมูลถูกส่งผ่านสัญญาณ SDA โดยอุปกรณ์หลักหรือ อุปกรณ์ Slave และขึ้นอยู่กับว่าเป็นบิต R/W

หลังจากที่ Data Frame ถูกส่งไปหมดแล้ว อุปกรณ์หลักสร้างเงื่อนไขการหยุดขึ้น เงื่อนไขการหยุดถูกนิยามโดยการเปลี่ยนแปลงของบิต 0 -> 1 (ระดับสัญญาณจากต่ำไปสูง) ของสัญญาณ SDA หลังจากการเปลี่ยนแปลงของสัญญาณ SCL ในช่วงบิต 0 -> 1 โดยระดับสัญญาณ SCL เป็นสูงตลอด ในระหว่างการเขียนข้อมูลปกติ ค่าในสัญญาณ SDA ไม่ควรเกิดการเปลี่ยนแปลงขึ้นขณะที่ระดับสัญญาณ SCL ยังสูงอยู่เพื่อที่ยับยั้งไม่ให้เกิดเงื่อนไขหยุดผิดพลาด

2.4 SPI Protocol

SPI หรือ Serial Peripheral Interface เป็นวิธีการสื่อสารอนุกรมแบบ Synchronous อีกรูปแบบหนึ่ง ซึ่งทำงานในรูปแบบที่ให้อุปกรณ์ตัวหนึ่งทำหน้าที่เป็น Master ในขณะที่อีกตัวหนึ่งทำหน้าที่เป็น Slave และสามารถส่งข้อมูลในโหมด Full-duplex ทำให้เราสามารถรับส่งข้อมูลระหว่างกันได้อย่างต่อเนื่อง รูปแบบการสื่อสารของ SPI ไม่กำหนดมาตรฐานตายตัว ประกอบไปด้วยสัญญาณ 4 สายคือ

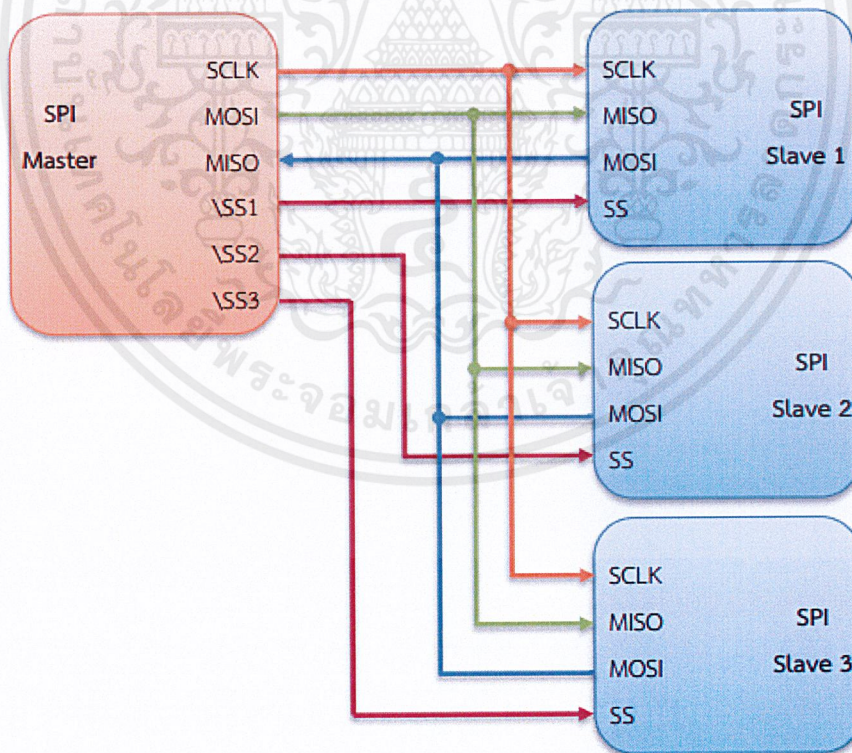


รูปที่ 8 การเชื่อมต่อสื่อสารแบบ SPI ระหว่างอุปกรณ์

- SCLK (Serial Clock) ใช้ส่งสัญญาณนาฬิกาจากอุปกรณ์ Master ไปยังอุปกรณ์ Slave ไว้กำหนดจังหวะรับส่งข้อมูล
- MOSI (Master Out Slave In) ใช้ส่งสัญญาณข้อมูลจาก Master ไปยัง Slave
- MISO (Master In Slave Out) ใช้ส่งสัญญาณข้อมูลจาก Slave ไปยัง Master
- SS (Slave Select) หรือ CS (Chip Select) ใช้ส่งสัญญาณ Low ไปยังอุปกรณ์ Slave ที่ต้องการข้อมูล

อุปกรณ์ Master ทำหน้าที่เป็นตัวควบคุมการสื่อสารทั้งหมด โดยควบคุมการสื่อสารตามสัญญาณนาฬิกา และสายสัญญาณ SS ตัวมาสเตอร์เป็นตัวที่ตัดสินใจเลือก รับ หรือ ส่งข้อมูลไปยังอุปกรณ์ Slave

สัญญาณ SS หรือ Slave select ในกรณีที่มีอุปกรณ์ Slave มากกว่า 1 ตัว โดยการทำให้เส้น SS มีระดับสัญญาณเป็น Low เมื่อต้องการติดต่อกับ Slave ตัวใด จากรูปที่ 9 หากต้องการติดต่อสื่อสารกับอุปกรณ์ Slave ตัวใด ก็เพียงทำให้สัญญาณ SS ของ Slave ตัวนั้น มีระดับสัญญาณเป็น Low



รูปที่ 9 การสื่อสารแบบ SPI ระหว่างอุปกรณ์ Master กับ Slave หลายตัว

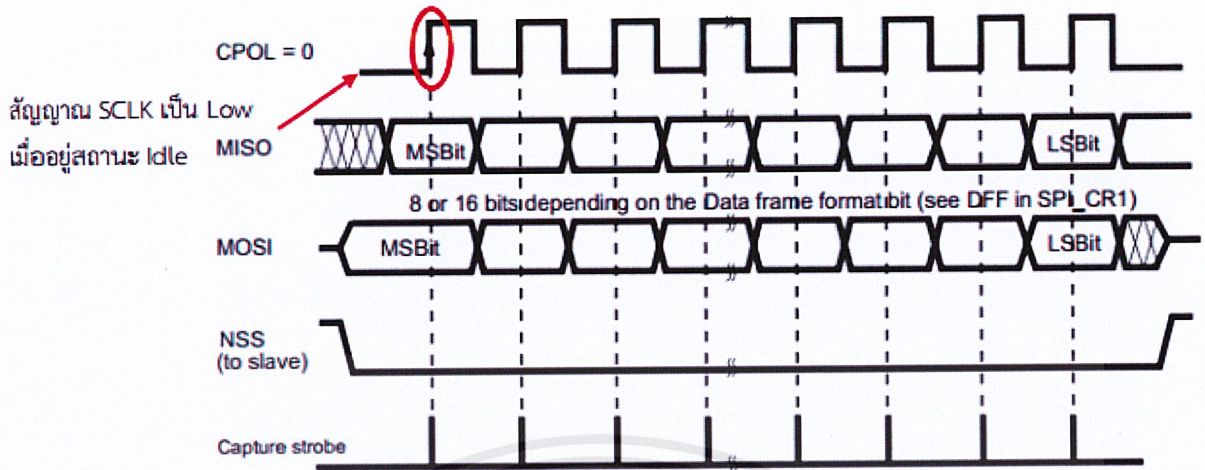
เมื่อเปรียบเทียบระหว่างการสื่อสาร SPI กับ I2C เป็นการสื่อสารอนุกรมแบบ Synchronous เหมาะสำหรับการสื่อสารข้อมูลความเร็วต่ำ เช่น หน่วยความจำ EEPROM หรือโมดูลนาฬิกาดิจิทัล

ข้อดีของการสื่อสารแบบ SPI คือ สามารถสื่อสารแบบ Full Duplex กล่าวคือสามารถรับและส่งข้อมูลได้พร้อมๆ กัน เพราะมีสายสัญญาณรับและส่งข้อมูลโดยเฉพาะ รูปแบบการสื่อสารของ SPI ไม่ต้องกำหนด Address เพื่อระบุอุปกรณ์ที่ต้องการสื่อสารเหมือน I2C เนื่องจากใช้สายสัญญาณ SS เป็นตัวควบคุม จึงมีอัตราการรับส่งข้อมูลสูงกว่า I2C และเหมาะสำหรับการรับส่งข้อมูลแบบต่อเนื่อง หรือ Streaming อย่างไรก็ตาม หากมีอุปกรณ์ Slave หลายตัวดังรูป การสื่อสารแบบ SPI ต้องใช้สายสัญญาณมากกว่า I2C

การสื่อสารแบบ SPI ไม่มีรูปแบบที่แน่นอน ดังนั้นกำหนดรูปแบบการสื่อสารของอุปกรณ์ Master ให้ตรงกับอุปกรณ์ Slave โดยมีรายละเอียดที่สำคัญ ดังนี้

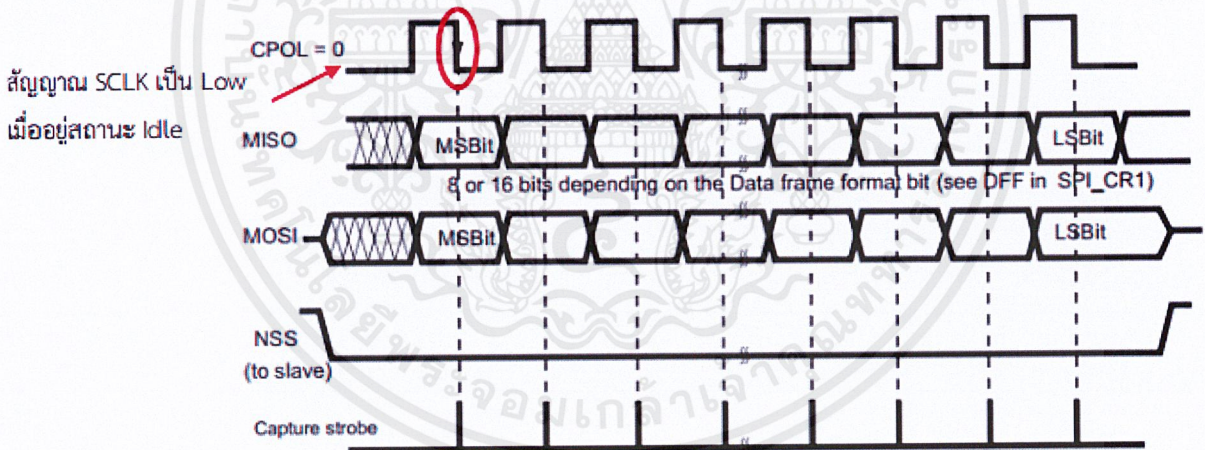
1. กำหนดทิศทางการสื่อสาร Full Duplex (รับส่งข้อมูลได้พร้อมกัน) Half Duplex_Tx ส่งข้อมูลอย่างเดียว Half Duplex_Rx รับข้อมูลอย่างเดียว
2. รูปแบบการรับส่งข้อมูล 8 บิต หรือ 16 บิต โดยสามารถเลือกส่งบิตนัยะสำคัญสูงสุด (MSB) หรือ บิตนัยะสำคัญต่ำสุด (LSB) ก่อน
3. ลักษณะของขอบสัญญาณนาฬิกา เพื่อกำหนดจังหวะการรับ-ส่งข้อมูล สามารถเลือกได้สี่แบบ ดังนี้

Clock polarity Low และ Clock phase 1st edge (CPOL_Low/CPHA_1Edge) ในสถานะเริ่มต้นสัญญาณนาฬิกาอยู่ในสถานะ Low ระบบเริ่มอ่านข้อมูลบิตแรก เมื่อพบขอบแรกของสัญญาณซึ่งเป็นลักษณะขอบขาขึ้น ดังรูป



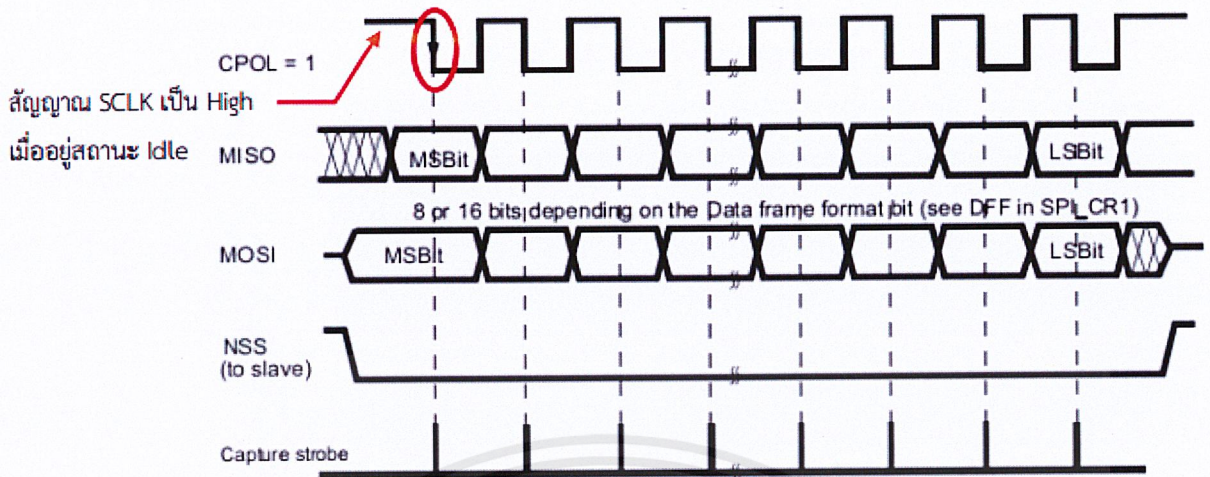
รูปที่ 10 การรับส่งข้อมูลเมื่อเลือก CPOL = 0 และ CPHA = 0

Clock polarity Low และ Clock phase 2st edge (CPOL_Low/CPHA_2Edge) ระบบเริ่มอ่านข้อมูลบิตแรก เมื่อพบขอบที่สองของสัญญาณซึ่งเป็นลักษณะขอบขาลง ดังรูป



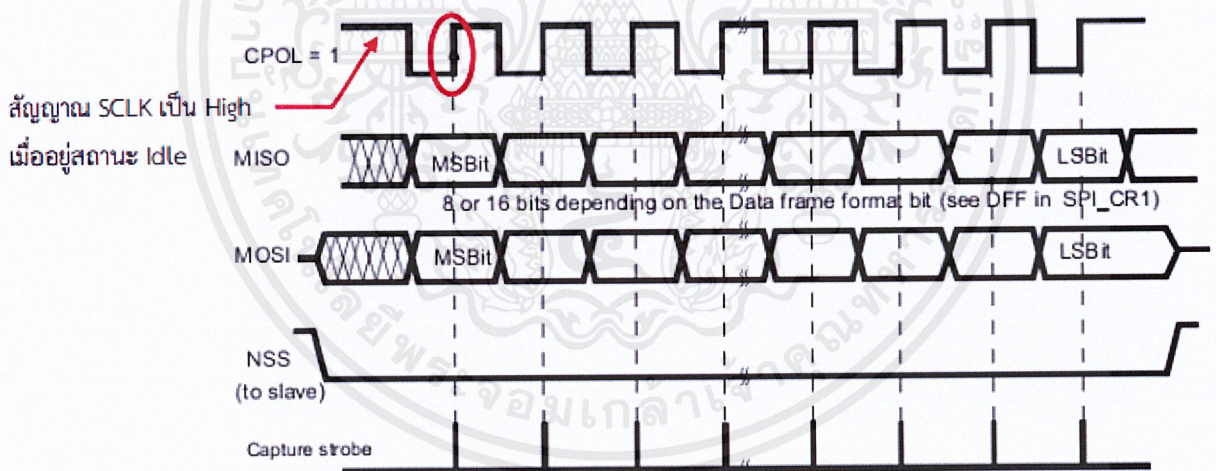
รูปที่ 11 การรับส่งข้อมูลเมื่อเลือก CPOL = 0 และ CPHA = 1

Clock polarity High และ Clock phase 1st edge (CPOL_High/CPHA_1Edge) ในสถานะเริ่มต้นสัญญาณนาฬิกาอยู่ในสถานะ High ระบบเริ่มอ่านข้อมูลบิตแรก เมื่อพบขอบแรกของสัญญาณซึ่งเป็นลักษณะขอบขาลง



รูปที่ 12 การรับส่งข้อมูลเมื่อเลือก CPOL = 1 และ CPHA = 0

Clock polarity High และ Clock phase 2st edge (CPOL_High/CPHA_2Edge) ระบบเริ่มอ่านข้อมูลบิตแรก เมื่อพบขอบที่สองของสัญญาณซึ่งเป็นลักษณะขอบขาขึ้น



รูปที่ 13 การรับส่งข้อมูลเมื่อเลือก CPOL = 1 และ CPHA = 1

บทที่ 3 วิธีการดำเนินงาน

ในบทนี้เป็นการพัฒนาเครื่องวัด และบันทึกแรงสั่นสะเทือนในอุปกรณ์ทดสอบหัวอ่าน ประกอบไปด้วย รับทราบความต้องการของระบบ, ตัวต้นแบบ

3.1 รับทราบความต้องการของระบบ

จากการพูดคุยรายละเอียดกับทาง Project Manager พบว่าบริษัทต้องการเครื่องมือสำหรับการวัดแรงสั่นในอุปกรณ์ทดสอบหัวอ่าน โดยมีฟังก์ชันดังต่อไปนี้

- 1) ใช้งานในอุปกรณ์ทดสอบหัวอ่านได้
- 2) ใช้งานได้นานอย่างน้อย 1 ชั่วโมง
- 3) บันทึกข้อมูลได้
- 4) ข้อมูลที่ได้มีความแม่นยำมากที่สุด

3.1.1 ศึกษาและเลือกวิธีการวัดแรงกระแทกที่เหมาะสมที่สุด

การวัดแรงสั่นสะเทือนที่เลือกมามี 2 วิธีหลักคือ

- ใช้วิธีวัดจากการกระทำต่อตัวเซนเซอร์หรือแรงกด (Piezo)
- ใช้วิธีวัดจากความเร่งที่เซนเซอร์เคลื่อนที่ (MEMS)

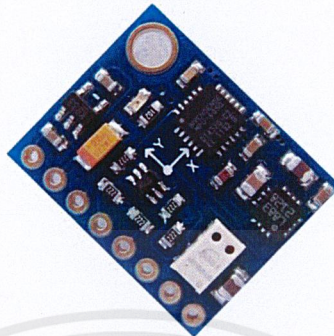
หลังจากที่ได้ศึกษาวิธีการวัดแล้ว มาเทียบกับปัญหาที่ได้รับมา ดังนั้นการใช้วิธีวัดวิธีที่ 2 ที่เหมาะสมที่สุด

3.1.2 ออกแบบ ต่อบางจร และเขียนโปรแกรม

โดยในขั้นตอนนี้แบ่งเป็น 2 ส่วนคือ การพิสูจน์ทฤษฎี และการประกอบตัวต้นแบบ

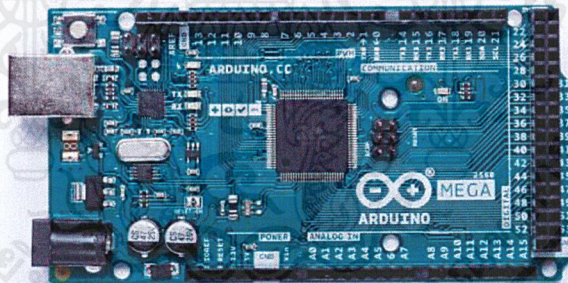
หลังจากเลือกวิธีการวัด เราจึงมองหาอุปกรณ์สำหรับการพิสูจน์ทฤษฎี และเป็นพื้นฐานในการพัฒนาตัวต้นแบบต่อไป โดยมีอุปกรณ์ดังต่อไปนี้

- IMU : GY – 86



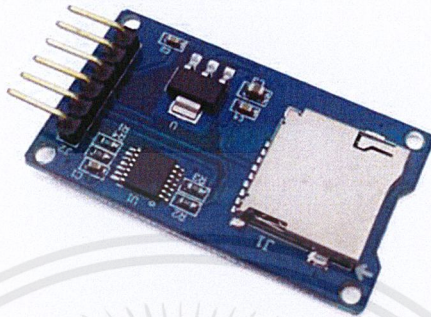
รูปที่ 14 GY – 86

- Microprocessor : Arduino Mega 2560 Rev3



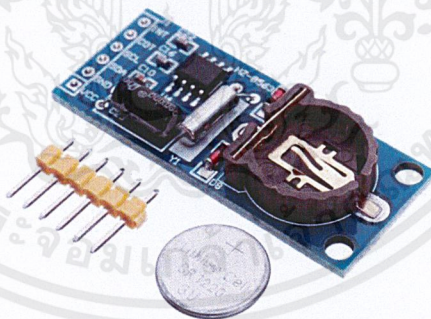
รูปที่ 15 Arduino Mega 2560 R3

- Logging Module : Micro SD Card SPI Reader Module



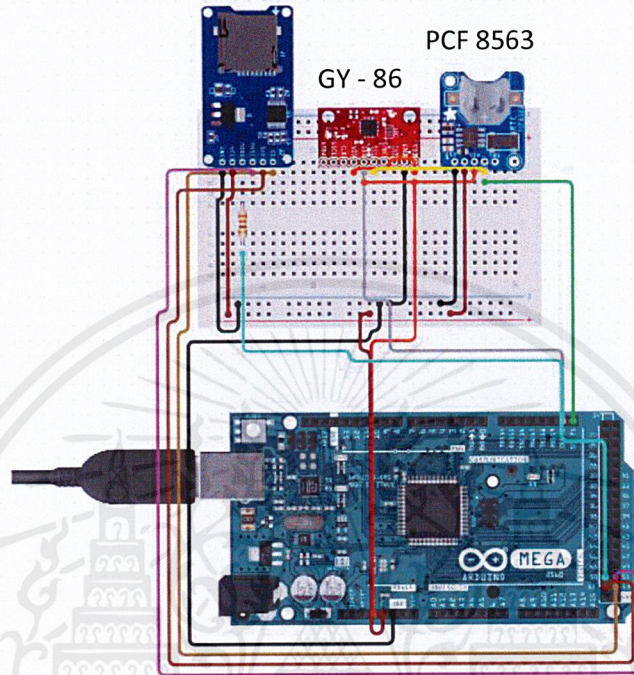
รูปที่ 16 Micro SD Card SPI Reader Module

- Real Time Clock module : PCF 8563



รูปที่ 17 PCF 8563

จากนั้นจึงทำการเชื่อมต่ออุปกรณ์ทั้งหมดเข้าด้วยกันตามรูป

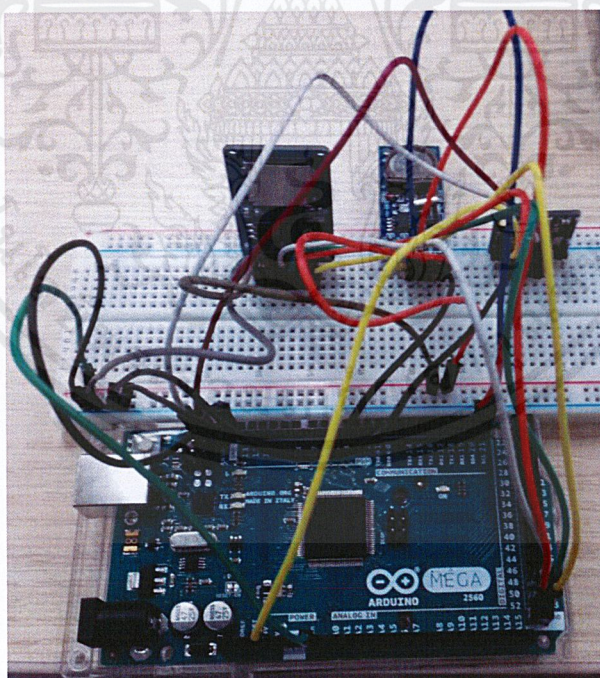


รูปที่ 18 การเชื่อมต่ออุปกรณ์ต่างๆ

โดยมีจุดการเชื่อมต่ออุปกรณ์ทั้งหมดตามตารางต่อไปนี้

Arduino Mega 2560 Rev3	Components
3.3V	3.3V
Vin	Vcc
SDA1,20SDA	SDA
SCL1,21SCL	SCL
GND	GND
50	MISO
51	MOSI
52	SCK
53	CS

ตารางที่ 1 จุดการเชื่อมต่อ



รูปที่ 19 การเชื่อมต่ออุปกรณ์

เมื่อเชื่อมต่อเสร็จแล้ว จึงเขียนชุดคำสั่งลงไปให้อุปกรณ์ทำงานได้ตามต้องการ

```

#include <RTCLib2.h>

#include <Wire.h>

#include <SD.h>

#include <SPI.h>

#include <MPU6050_tockn.h>

File myFile;

int pinCS = 53;

MPU6050 mpu6050(Wire);

long timer = 0;

boolean didMyOneTimeAction = false;

PCF8563 rtc;

void setup ()
{
  SerialUSB.begin(115200);

  SerialUSB.print("compiled: ");

  SerialUSB.print(__DATE__);

  SerialUSB.println(__TIME__);

  Rtc.Begin();

  RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__);

```

```

printData(compiled);

SerialUSB.println();

if (!Rtc.IsDateTimeValid())

{

    if (Rtc.LastError() != 0)

    {

        SerialUSB.print("RTC communications error = ");

        SerialUSB.println(Rtc.LastError());

    }

    else

    {

        Rtc.SetDateTime(compiled);

    }

}

if (!Rtc.GetIsRunning())

{

    SerialUSB.println("RTC was not actively running, starting now");

    Rtc.SetIsRunning(true);

```

```

}

RtcDateTime now = Rtc.GetDateTime();

if (now < compiled)
{
    SerialUSB.println("RTC is older than compile time! (Updating DateTime)");

    Rtc.SetDateTime(compiled);
}
else if (now > compiled)
{
    SerialUSB.println("RTC is newer than compile time. (this is expected)");
}
else if (now == compiled)
{
    SerialUSB.println("RTC is the same as compile time! (not expected but all is
fine)");
}

if (SD.begin())
{

```

```

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

}

else

{

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

return;

}

mpu6050.begin();

mpu6050.calcGyroOffsets(true);SerialUSB.println(" ");

SerialUSB.print("accX(G) : ");SerialUSB.print("\taccY(G) : ");SerialUSB.println("\taccZ(G) :

");

}

void loop ()

{

if (!Rtc.IsDateTimeValid())

{

```

```

if (Rtc.LastError() != 0)

{

    SerialUSB.print("RTC communications error = ");

    SerialUSB.println(Rtc.LastError());

}

else

{

    SerialUSB.println("RTC lost confidence in the DateTime!");

}

}

RtcDateTime now = Rtc.GetDateTime();

printData(now);

myFile = SD.open("sdlog.txt", FILE_WRITE);

if (didMyOneTimeAction == false)

{

    didMyOneTimeAction = true;

    myFile.print("accX(G)");myFile.print("\taccY(G)");myFile.println("\taccZ(G)");

}

```

```

if (myFile) {

    myFile.print("Time = ");

    myFile.println(datestring);

    myFile.print(mpu6050.getAccX());

    myFile.print("\t");

    myFile.print(mpu6050.getAccY());

    myFile.print("\t");

    myFile.println(mpu6050.getAccZ());

    myFile.close(); // close the file
}

else {

    SerialUSB.println("error opening test.txt");

}

SerialUSB.print(mpu6050.getAccX());

SerialUSB.print("\t");

SerialUSB.print(mpu6050.getAccY());

SerialUSB.print("\t");

SerialUSB.println(mpu6050.getAccZ());

```

```

delay(100);

}

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

void printData(const RtcDateTime& dt)
{
char datestring[20];
mpu6050.update();

snprintf_P(datestring,
countof(datestring),
PSTR("%02u/%02u/%04u %02u:%02u:%02u"),
dt.Month(),
dt.Day(),
dt.Year(),
dt.Hour(),
dt.Minute(),
dt.Second() );

```

```

SerialUSB.print("Time = ");

SerialUSB.println(datestring);

}

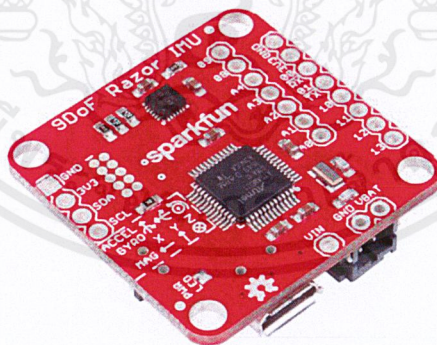
```

แล้วจึงนำมาทดสอบการตรวจวัดความเร่งโดยนำอุปกรณ์ที่เราทำไปติดไว้บนอุปกรณ์ทดสอบหัวอ่าน และติดตั้งบนแกนทรีของเครื่องจักรแล้วสั่งให้เครื่องจักรขยับแกนทรีไปตามแกน X, Y และ Z อย่างละ 5 รอบ

2. ตัวต้นแบบ

หลังจากการทดสอบวิธีการวัดเสร็จสมบูรณ์ จึงเริ่มทำการเริ่มพัฒนาตัวต้นแบบขึ้นมาที่สามารถใส่ลงในอุปกรณ์ทดสอบหัวอ่านได้ ทำให้อุปกรณ์ที่ใช้ทดสอบในช่วงต้นนั้นไม่สามารถนำมาใช้งานต่อได้ ทำให้ต้องหาอุปกรณ์ชิ้นใหม่มา

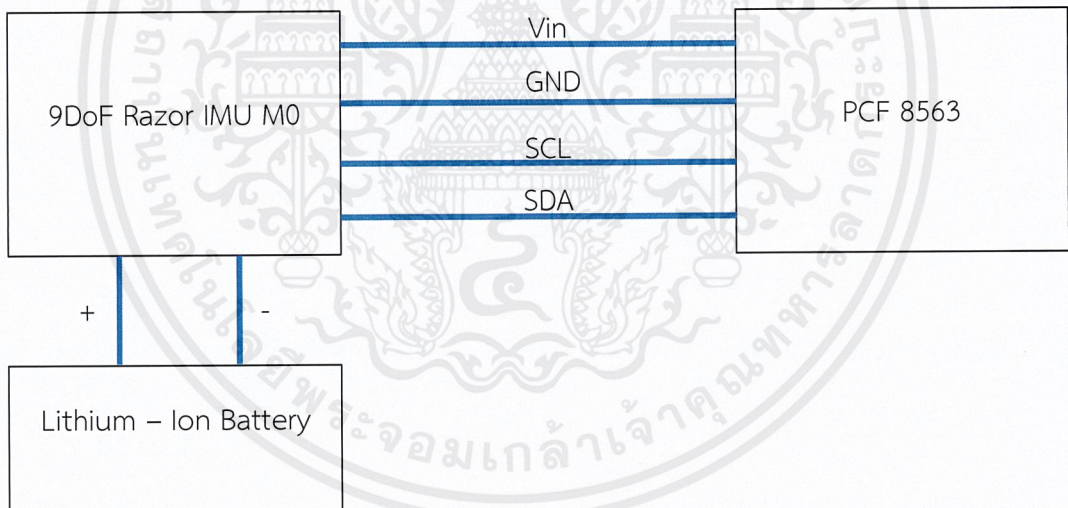
จึงได้อุปกรณ์ในการใช้ในโครงการนี้คือ 9DoF Razor IMU M0 ของ Sparkfun Electronics กับ Lithium – Polymer battery



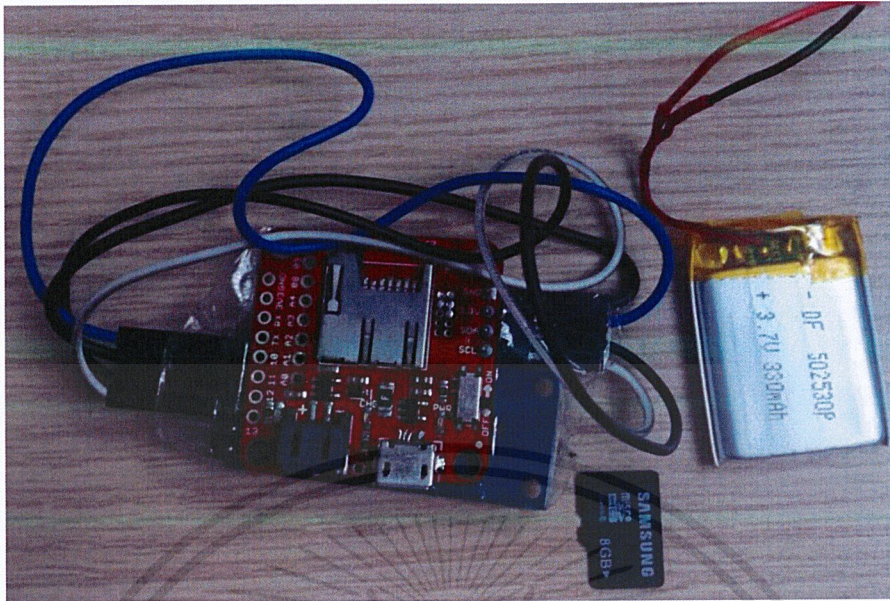
รูปที่ 20 9DoF Razor IMU M0

โดยอุปกรณ์นี้ประกอบไปด้วยไมโครโปรเซสเซอร์, Accelerometer, Micro SD Card Reader, USB Port, และช่องต่อสายแบตเตอรี่ไว้ในวงจรเดียวกัน ทำให้ลดพื้นที่งานเหลือเพียงแค่ประมาณ 30 x 30 x 10 มิลลิเมตร พร้อมทั้งมีฟังก์ชันในการลดสัญญาณรบกวนภายในตัว อุปกรณ์ชิ้นนี้จึงมีความเหมาะสมในการนำมาใช้งาน

จากนั้น เรานำอุปกรณ์นี้ไปต่อกับอุปกรณ์ Real Time Clock : PCF 8563 และแบตเตอรี่ลิเทียม – โพลีเมอร์ดังรูป



รูปที่ 21 แบบจำลองการเชื่อมต่ออุปกรณ์



รูปที่ 22 การเชื่อมต่ออุปกรณ์ทั้งหมด

จากนั้นเราจึงเขียนชุดคำสั่งลงไปในอุปกรณ์

```
#include <Wire.h>
#include <SD.h>
#include <SPI.h>
#include <RTCLib2.h>
#include <MPU9250_RegisterMap.h>
#include <SparkFunMPU9250-DMP.h>
```

```
File myFile;
```

```
const int chipSelect = 38;
```

```
Sd2Card card;
```

```
SdVolume volume;
```

```

SdFile root;

PCF8563 rtc;

MPU9250_DMP imu;

long timer = 0;

boolean didMyOneTimeAction = false;

void setup ()
{
  SerialUSB.begin(115200);
  Wire.begin();
  rtc.begin();
  if (! rtc.isrunning()) {
    SerialUSB.println("RTC is NOT running!");
    rtc.adjust(DateTime(__DATE__, __TIME__));
  }

  SerialUSB.println("Initializing SD card...");

  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
  }
}

```

```

Serial.println("card initialized.");

if (imu.begin() != INV_SUCCESS)

{

while (1)

{

SerialUSB.println("Unable to communicate with MPU-9250");

SerialUSB.println("Check connections, and try again.");

SerialUSB.println();

delay(5000);

}

}

SerialUSB.println(" ");

imu.setSensors(INV_XYZ_GYRO | INV_XYZ_ACCEL | INV_XYZ_COMPASS);

imu.setGyroFSR(2000); // Set gyro to 2000 dps

imu.setAccelFSR(4); // Set accel to +/-2g

imu.setLPF(2); // Set LPF corner frequency to 5Hz

imu.setSampleRate(1000); // Set sample rate to 10Hz

imu.setCompassSampleRate(10); // Set mag rate to 10Hz

pinMode(LED_BUILTIN, OUTPUT);

```

```

delay(1000);

}

void loop ()

{

    DateTime now = rtc.now();

    char buf[100];

    strncpy(buf,"DD.MM.YYYY\thh:mm:ss\0",100);

    myFile = SD.open("sdlog.txt", FILE_WRITE);

    if (didMyOneTimeAction != true)
    {
        didMyOneTimeAction = true;

        SerialUSB.print("Date : ");

        SerialUSB.print("\t");

        SerialUSB.print("Time : ");

        SerialUSB.print("\t");

        SerialUSB.print("AccX(g) : ");

        SerialUSB.print("\t");

        SerialUSB.print("AccY(g) : ");

        SerialUSB.print("\t");
    }
}

```

```

SerialUSB.println("AccZ(g) : ");

myFile.print("Time : ");

myFile.print("\t");

myFile.print("Date : ");

myFile.print("\t");

myFile.print("AccX(g) : ");

myFile.print("\t");

myFile.print("AccY(g) : ");

myFile.print("\t");

myFile.println("AccZ(g) :");
}

if ( imu.dataReady() )
{

imu.update(UPDATE_ACCEL | UPDATE_GYRO | UPDATE_COMPASS);

digitalWrite(LED_BUILTIN, HIGH);

SerialUSB.print(now.format(buf));SerialUSB.print("\t");

printIMUData();

if (myFile)

{

```

```

char buf[100];

strncpy(buf,"DD.MM.YYYY\thh:mm:ss\0",100);

myFile.print(now.format(buf));myFile.print("\t");

printIMUCard();

myFile.close();

}

else
{
  SerialUSB.println("error opening test.txt");
}
}

delay(10);
}

void printIMUData(void)

{

float accelX = imu.calcAccel(imu.ax);

float accelY = imu.calcAccel(imu.ay);

float accelZ = imu.calcAccel(imu.az);

float gyroX = imu.calcGyro(imu.gx);

```

```

float gyroY = imu.calcGyro(imu.gy);

float gyroZ = imu.calcGyro(imu.gz);

float magX = imu.calcMag(imu.mx);

float magY = imu.calcMag(imu.my);

float magZ = imu.calcMag(imu.mz);

float accelX2 = accelX*9.81;

float accelY2 = accelY*9.81;

float accelZ2 = accelZ*9.81;

SerialUSB.print(String(accelX2) + "\t" + String(accelY2) + "\t" + String(accelZ2));

SerialUSB.println();
}

void printIMUCard(void)
{

float accelX = imu.calcAccel(imu.ax);

float accelY = imu.calcAccel(imu.ay);

float accelZ = imu.calcAccel(imu.az);

float gyroX = imu.calcGyro(imu.gx);

float gyroY = imu.calcGyro(imu.gy);

float gyroZ = imu.calcGyro(imu.gz);

```

```
float magX = imu.calcMag(imu.mx);

float magY = imu.calcMag(imu.my);

float magZ = imu.calcMag(imu.mz);

float accelX2 = accelX*9.81;

float accelY2 = accelY*9.81;

float accelZ2 = accelZ*9.81;

myFile.print(String(accelX2) + "\t" + String(accelY2) + "\t" + String(accelZ2));

myFile.println();
}
```



จากนั้นเราจึงทดสอบการตรวจวัด และบันทึกข้อมูลว่าเป็นไปตามที่เราต้องการหรือไม่ ดังรูป

Date	Time	Value 1	Value 2	Value 3
28.11.2019	09:25:41	1.14	-0.05	-8.86
28.11.2019	09:25:41	1.21	-0.02	-9.25
28.11.2019	09:25:41	1.09	-0.18	-9.02
28.11.2019	09:25:41	1.17	-0.02	-9.11
28.11.2019	09:25:41	1.07	-0.11	-9.10
28.11.2019	09:25:41	1.09	-0.09	-9.10
28.11.2019	09:25:41	1.04	-0.10	-9.05
28.11.2019	09:25:41	1.02	0.05	-9.06
28.11.2019	09:25:41	1.08	-0.06	-8.97
28.11.2019	09:25:41	1.08	-0.09	-9.15
28.11.2019	09:25:42	1.14	-0.12	-9.10
28.11.2019	09:25:42	1.10	-0.09	-9.16
28.11.2019	09:25:42	1.07	-0.10	-9.08
28.11.2019	09:25:42	1.05	-0.19	-9.03
28.11.2019	09:25:42	1.18	-0.03	-9.00
28.11.2019	09:25:42	1.04	-0.14	-9.13
28.11.2019	09:25:42	1.09	0.01	-9.04
28.11.2019	09:25:42	1.13	-0.02	-9.07
28.11.2019	09:25:42	1.12	-0.05	-9.05
28.11.2019	09:25:42	1.12	-0.05	-9.06
28.11.2019	09:25:42	1.12	-0.25	-9.37
28.11.2019	09:25:42	1.09	-0.09	-8.99
28.11.2019	09:25:42	1.22	-0.13	-9.05
28.11.2019	09:25:42	1.09	-0.14	-8.99
28.11.2019	09:25:42	1.16	-0.08	-9.03
28.11.2019	09:25:42	1.06	-0.10	-9.05
28.11.2019	09:25:42	1.16	-0.02	-9.01
28.11.2019	09:25:42	1.14	0.03	-9.13
28.11.2019	09:25:42	1.10	-0.01	-9.10
28.11.2019	09:25:42	1.14	-0.04	-9.20
28.11.2019	09:25:42	0.95	-0.06	-8.60
28.11.2019	09:25:42	1.22	0.16	-9.09
28.11.2019	09:25:42	1.21	-0.13	-9.17
28.11.2019	09:25:42	1.23	-0.02	-9.14

รูปที่ 23 การแสดงข้อมูลผลการวัด

เมื่อเราทราบว่าอุปกรณ์ของเราทำงานได้ เราจึงนำมาประเมินหาความเที่ยงตรงของข้อมูลที่เรบันทึก โดยการใช้การเปรียบเทียบข้อมูลระหว่างตัวต้นแบบ กับ Inertial Measurement Unit ที่ใช้ในอุตสาหกรรมคือ MTi – 30 – 2A8G4 จาก Xsens



รูปที่ 24 อุปกรณ์ Inertial Measurement Unit MTi – 30 – 2A8G4

Input voltage	4.5-34V or 3V3	Typical power consumption	400-550 mW
IP-rating	IP 67 (encased)	Temperature	-40 to 85 °C
Vibration and shock	2000 g for 0.5 ms	Sampling frequency	10 kHz/channel (60 kS/s)
Output frequency	Up to 2 kHz	Clock Drift	10ppm or external reference
Latency	< 2 ms	MTBF	300,000 Hours
Standard full range gyro	450 °/s	Standard full range acc	200 m/s ²
In-run bias stability gyro	18°/h	Bandwidth gyro	415 Hz
Bandwidth acc	375 Hz	Interfaces	RS232/RS485/RS422/UART/USB

รูปที่ 25 คุณสมบัติของ MTi – 30 – 2A8G4

โดยแบ่งการทดสอบ และเปรียบเทียบข้อมูลออกเป็น 3 แกนคือ แกน X, Y, และ Z

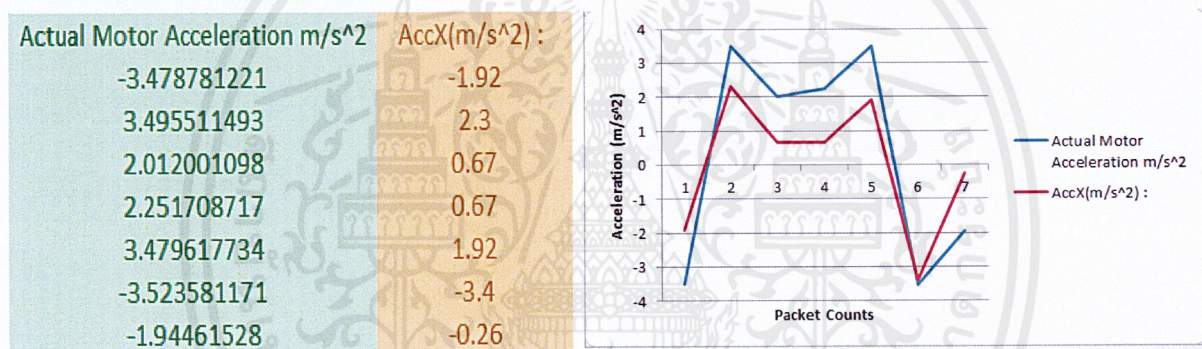
บทที่ 4 ผลการดำเนินงาน

ผลสรุปการปฏิบัติงานโครงการสหกิจศึกษา ณ บริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัด ตั้งแต่ วันที่ 5 สิงหาคม พ.ศ.2562 ถึง 29 ธันวาคม พ.ศ.2562 โดยผมได้ปฏิบัติงานในแผนก Advance Manufacturing Engineering ทีม Software Integration โดยได้รับมอบหมายงานให้พัฒนาเครื่องวัดและบันทึกแรงสั่นสะเทือนในอุปกรณ์ทดสอบหัวอ่าน มีรายละเอียดการดำเนินงานดังต่อไปนี้

4.1 ผลการทดสอบวิธีการวัด

หลังจากการทดสอบการวัดแรงสั่นสะเทือนทั้ง 3 แกน ได้ผลลัพธ์ดังต่อไปนี้

แกน X

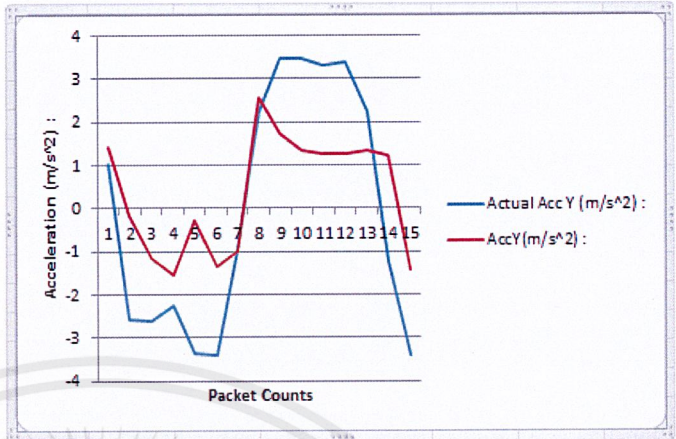


รูปที่ 26 เปรียบเทียบการตรวจวัดค่าความเร่งแกน X

- Root mean square error แกน X = 1.386023 เมตรต่อวินาทีกำลังสอง

แกน Y

Actual Acc Y (m/s ²) :	AccY(m/s ²) :
1.039700255	1.42
-2.563822162	-0.17
-2.596321513	-1.14
-2.241548558	-1.56
-3.346125156	-0.28
-3.401778835	-1.33
-0.953918708	-1
2.235620565	2.58
3.484823072	1.73
3.459995242	1.33
3.315561433	1.26
3.385163047	1.27
2.236806164	1.35
-1.242088915	1.23
-3.403661854	-1.44

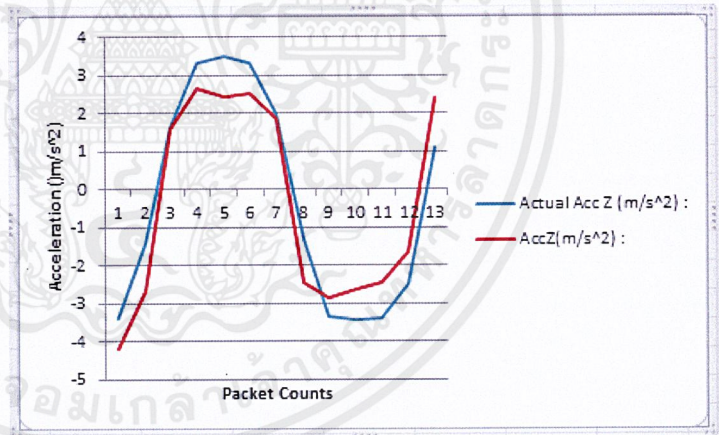


รูปที่ 27 เปรียบเทียบการตรวจวัดค่าความเร่งแกน Y

- Root mean square error แกน Y = 1.813719 เมตรต่อวินาทีกำลังสอง

แกน Z

Actual Acc Z (m/s ²) :	AccZ(m/s ²) :
-3.390533111	-4.2
-1.462261555	-2.68
1.635358954	1.6
3.319745899	2.64
3.486496859	2.42
3.324488293	2.5
2.001918153	0.84
-1.30158473	-2.46
-3.334199367	-2.88
-3.457414822	-2.66
-3.396251881	-2.46
-2.49303495	-1.68
1.096051061	2.4



รูปที่ 28 เปรียบเทียบการตรวจวัดค่าความเร่งแกน Z

- Root mean square error แกน Z = 0.871198 เมตรต่อวินาทีกำลังสอง

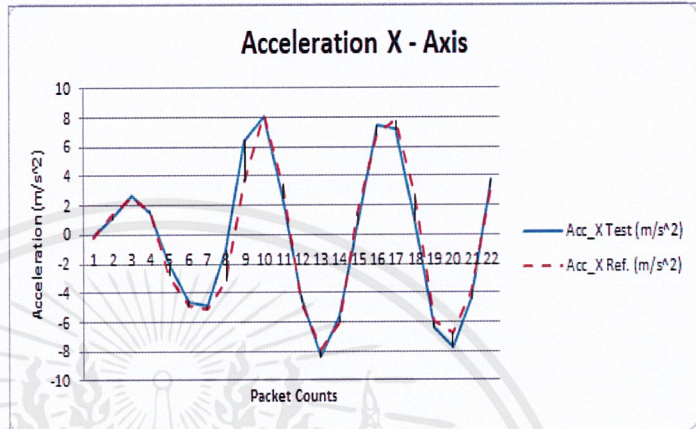
โดยผลที่ได้คือ อุปกรณ์ที่ต่อนั้นสามารถตรวจวัดการเคลื่อนไหวของอุปกรณ์ทดสอบหัวอ่าน ในขณะที่เคลื่อนไหวได้ ถึงแม้ว่าจะไม่เหมือนขั้นตอนในเครื่องจักรจริงๆก็ตาม แต่สามารถพิสูจน์ได้ว่าวิธีตรวจวัดนั้นใช้งานได้จริง

4.2 ผลการทดสอบในเปรียบเทียบข้อมูล

หลังจากการทดสอบการวัดแรงสั่นสะเทือนทั้ง 3 แกน ได้ผลลัพธ์ดังต่อไปนี้

แกน X

Acc_Z Ref. (m/s ²)	Acc_Z Test (m/s ²)
10.145128	10.48
10.037807	9.19
8.489968	8.02
7.118345	7.26
6.508722	8.54
8.388433	10.37
10.355817	12.03
13.061062	14.18
14.184978	14.67
13.706528	14.11
10.304096	11.37

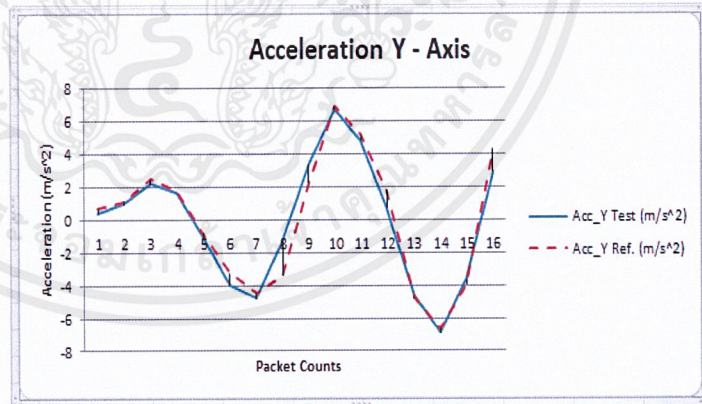


รูปที่ 29 เปรียบเทียบอุปกรณ์กับ MTi – 30 แกน X

- Root mean square error แกน X = 0.955543 เมตรต่อวินาทีกำลังสอง

แกน Y

Acc_Y Test (m/s ²)	Acc_Y Ref. (m/s ²)
0.45	0.661637
0.97	1.076959
2.2	2.473389
1.68	1.706853
-1.03	-0.791283
-3.97	-3.268867
-4.8	-4.503281
-1.28	-3.396304
3.33	2.262156
6.73	6.897528
4.84	5.150455
0.74	1.767853
-4.66	-4.789379
-6.89	-6.662689
-3.62	-3.946755
2.92	4.261251

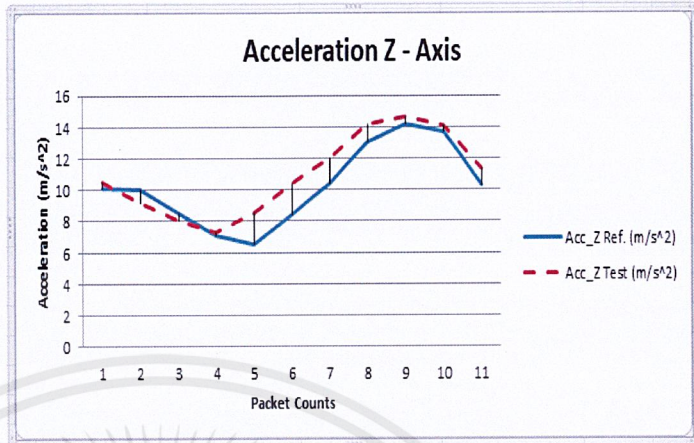


รูปที่ 30 เปรียบเทียบอุปกรณ์กับ MTi – 30 แกน Y

- Root mean square error แกน Y = 0.77227 เมตรต่อวินาทีกำลังสอง

แกน Z

Acc_Z Ref. (m/s ²)	Acc_Z Test (m/s ²)
10.145128	10.48
10.037807	9.19
8.489968	8.02
7.118345	7.26
6.508722	8.54
8.388433	10.37
10.355817	12.03
13.061062	14.18
14.184978	14.67
13.706528	14.11
10.304096	11.37



รูปที่ 31 เปรียบเทียบอุปกรณ์กับ MTi – 30 แกน Z

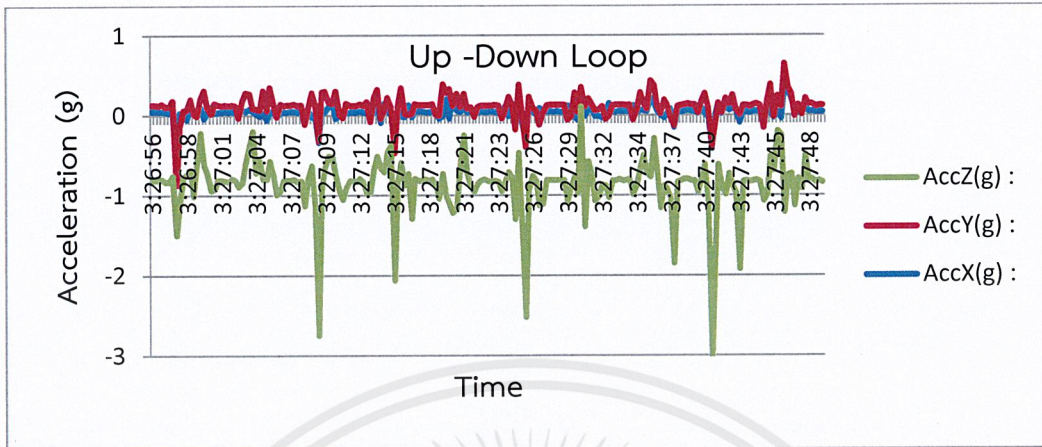
- Root mean square error แกน Z = 1.156548 เมตรต่อวินาทีกำลังสอง

จากข้อมูลที่เปรียบเทียบมานั้น เราสรุปได้ว่า อุปกรณ์ต้นแบบนั้นสามารถตรวจวัดแรงกระแทกได้ใกล้เคียงกับ MTi – 30 ที่ผ่านการทดสอบการวัดค่ามาแล้ว โดยค่า Root mean square error อยู่ในระดับในค่าที่รับได้

4.3 ผลการทดสอบในเครื่องจักร

เมื่อเครื่องวัดเราผ่านการทดสอบตามที่เราคาดหวัง เราจึงนำไปใส่ในอุปกรณ์ทดสอบหัวอ่าน และใช้งานในเครื่องจักรที่สายการผลิต โดยแบ่งการทดสอบเป็น 3 แบบคือ การเคลื่อนที่ขึ้นลงต่อเนื่อง 5 รอบ การใส่อุปกรณ์ทดสอบในช่องทดสอบทุกช่องต่อเนื่อง 5 รอบ และการจำลองขั้นตอนในการทำงาน 5 รอบ

การเคลื่อนที่ขึ้นลงตอนเนื่อง 5 รอบ

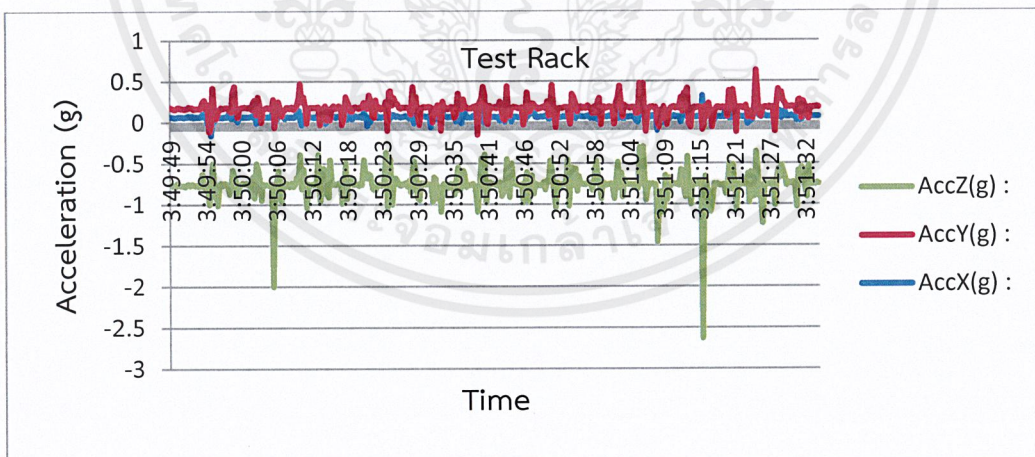


รูปที่ 32 การเคลื่อนที่ขึ้นลงตอนเนื่อง

จากผลการทดสอบนั้น แกนที่แรงสั่นสะเทือนมากที่สุด คือ แกน Z โดยมีค่าอยู่ระหว่าง -2 จี ถึง 0.1 จี ส่วนแกน X, และ Y มีค่าอยู่ระหว่าง - 0.3 จี ถึง 0.3 จี

ส่วนแกน Y และ Z ที่มีค่าประมาณ - 1 จี กับ -3 จี เป็นค่าความเร่งที่ผิดปกติ ตามลำดับ

การใส่อุปกรณ์ทดสอบหัวอ่านในช่องทดสอบทุกช่องต่อเนื่อง 5 รอบ

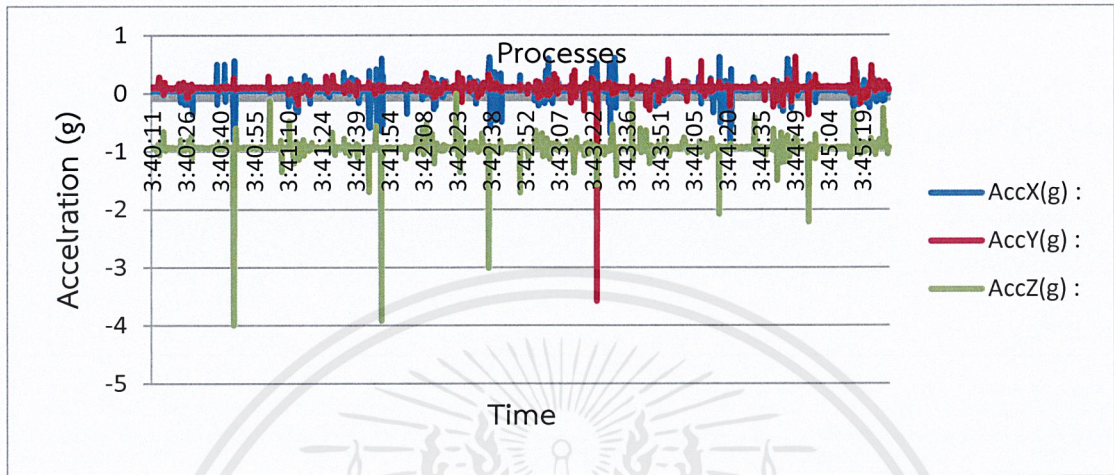


รูปที่ 33 การใส่อุปกรณ์ทดสอบหัวอ่านในช่องทดสอบทุกช่อง

จากผลการทดสอบนั้น แกนที่แรงสั่นสะเทือนมากที่สุด คือ แกน Z โดยมีค่าแรงสั่นสะเทือนอยู่ระหว่าง - 1.5 จี ถึง -0.3 ส่วนแกน X, กับ Y มีค่าแรงสั่นสะเทือนอยู่ระหว่าง - 0.1 จี ถึง 0.6 จี

ค่าแรงสั่นสะเทือนขนาด - 2 จี กับ - 2.6 จี นั้นที่แกน Z เป็นค่าความเร่งที่ผิดปกติ

การจำลองขั้นตอนในการทำงาน 5 รอบ



รูปที่ 34 การจำลองขั้นตอนในการทำงาน 5 รอบ

จากผลการทดสอบนั้น แกนที่แรงสั่นสะเทือนมากที่สุด คือ แกน Z โดยมีค่าสูงสุดอยู่ที่ - 2.3 จี ถึง 0.1 จี รองลงมาเป็นแกน X มีค่าแรงสั่นที่ - 1 จี ถึง 0.5 จี ส่วนแกน Y มีค่าแรงสั่นน้อยที่สุด อยู่ระหว่าง - 0.1 จี ถึง 0.5 จี

ส่วนค่าแรงสั่นสะเทือนขนาด - 4, - 3.8, - 3 จี ที่อยู่ในแกน Y และ Z นั้นเป็นความเร่งที่ผิดปกติ

ในข้อกำหนดของบริษัท แรงสั่นสะเทือนที่เกิดขึ้นกระทำต่ออุปกรณ์ทดสอบหัวอ่านต้องมีค่าน้อยกว่า 5 จี เมื่อนำมาเทียบกับข้อมูลที่เรารับบันทึก ทำให้เห็นได้ว่าเครื่องจักรที่เราทดสอบนั้นผ่านการทดสอบตามข้อกำหนด แต่เราก็ไม่สามารถเพิกเฉยค่าความเร่งที่ผิดปกติ ได้ เพราะค่านี้เป็นการบ่งบอกได้ว่าแรงสั่นที่เกิดขึ้นนี้ เป็นแรงสั่นที่ไม่ควรเกิดขึ้น ถึงแม้ว่าความแรงนั้นยังไม่เกินค่ามาตรฐาน เราจึงต้องนำข้อมูลเหล่านี้ปรับปรุงให้เครื่องจักรให้มีความแข็งแรงในการสั่นให้น้อยลงกว่านี้

บทที่ 5 สรุปผล และข้อเสนอแนะ

ในบทนี้เป็นส่วนการสรุปผลโครงการนี้ทั้งหมดที่ได้พัฒนาขึ้นมา แบ่งเป็นหัวข้อดังต่อไปนี้คือ สรุปผลการดำเนินงาน ปัญหาและอุปสรรคในการดำเนินงาน วิธีการแก้ปัญหา เป้าหมายต่อไป ข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

หลังจากที่พัฒนา ทดสอบ และบันทึกข้อมูลเครื่องวัด และบันทึกค่าแรงสั่นสะเทือนในอุปกรณ์ทดสอบ หัวอ่าน ได้ข้อสรุปดังต่อไปนี้

- 1) เครื่องวัดที่พัฒนาขึ้นมาวัดค่าแรงสั่นสะเทือนได้ใกล้เคียงกับ Inertial Measurement Unit ที่สำหรับใช้ในอุตสาหกรรมขนาดใหญ่ โดยมีค่า Root mean square error สูงสุดเท่ากับ 1.156548 เมตรต่อวินาทีกำลังสอง หรือ 0.1156548 จี นับว่าเป็นค่าที่น้อยมาก
- 2) เครื่องวัดนี้สามารถใช้งานได้นานกว่า 8 ชั่วโมง ทำให้สามารถทดสอบได้อย่างต่อเนื่อง
- 3) เครื่องวัดนี้มีขนาดเล็กพอที่สามารถใส่ในอุปกรณ์ทดสอบหัวอ่าน ทำให้ทราบถึงผลของแรงสั่นสะเทือนที่กระทำต่อหัวอ่านภายในการทดสอบได้

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

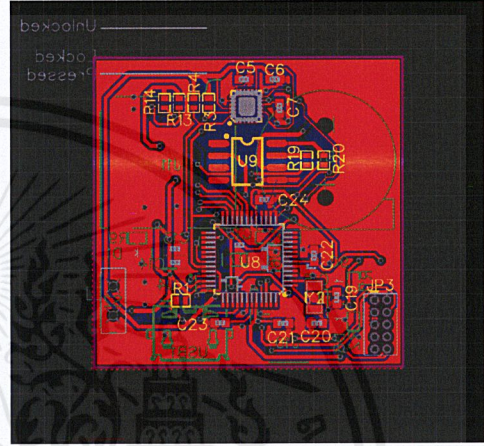
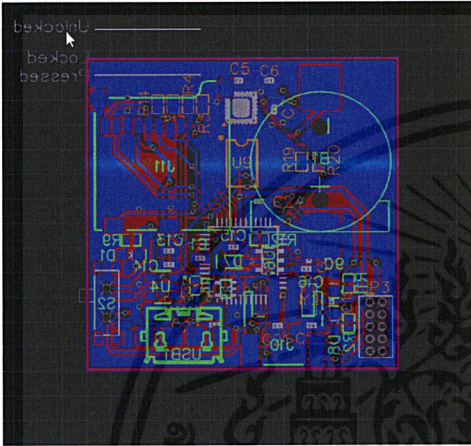
- 1) โครงการที่ทำนี้เป็นโครงการที่เกี่ยวข้องกับการสร้าง และพัฒนาอุปกรณ์ฮาร์ดแวร์เป็นส่วนมาก ซึ่งเป็นส่วนที่มีความเข้าใจน้อย ทำให้การพัฒนาเป็นไปได้ช้า
- 2) ระบบบริษัทมีความซับซ้อนจึงต้องใช้เวลาในการปรับตัวให้เข้ากับวัฒนธรรมขององค์กรในช่วงแรก จึงทำให้งานในช่วงแรกพัฒนาได้ช้า

5.3 วิธีแก้ไขปัญหา

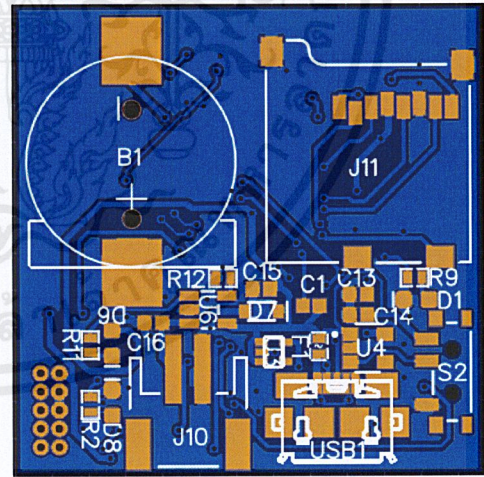
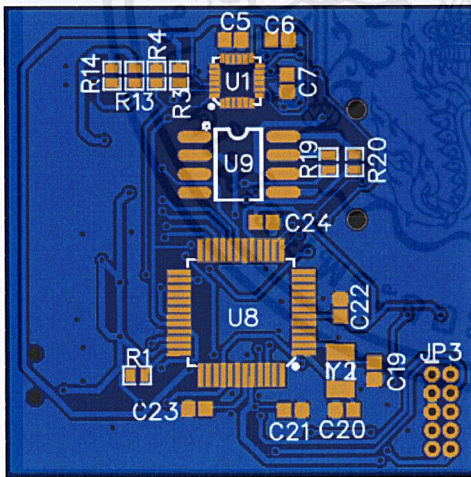
- 1) ศึกษาข้อมูลด้วยตนเอง พร้อมสอบถาม หรือขอข้อเสนอแนะกับพี่ในทีมเมื่อเกิดปัญหานั้นๆ
- 2) ฝึกพัฒนาบุคคลิกภาพ เพิ่มพูนทักษะในการสื่อสาร จนมีผลให้ทักษะทางด้านอารมณ์ (soft skill) ที่ดีขึ้น

5.4 ข้อเสนอแนะ

หลังจากที่พัฒนาเครื่องวัดจนสามารถใช้งานได้จริง ในขั้นตอนต่อไปเป็นการนำอุปกรณ์ทั้งหมดมารวมอยู่ในแผ่นปริ้นท์ PCB 1 แผ่น เพื่อลดพื้นที่ของเครื่องวัด และเปลี่ยนให้เครื่องมือวัดนี้เหมาะสมกับการใช้งานในสายการผลิต



รูปที่ 35 ดีไซน์ PCB



รูปที่ 36 แบบจำลอง PCB

บรรณานุกรม

- [1] Inertial Measurement Unit [ระบบออนไลน์] แหล่งที่มา https://en.wikipedia.org/wiki/Inertial_measurement_unit (24 ตุลาคม 2562)
- [2] Accelerometer [ระบบออนไลน์] แหล่งที่มา <https://en.wikipedia.org/wiki/Accelerometer> (24 ตุลาคม 2562)
- [3] What is a capacitive accelerometer? [ระบบออนไลน์] แหล่งที่มา <https://www.techopedia.com/definition/24745/capacitive-accelerometer> (24 ตุลาคม 2562)
- [4] Gyroscope [ระบบออนไลน์] แหล่งที่มา <https://en.wikipedia.org/wiki/Gyroscope> (24 ตุลาคม 2562)
- [5] Magnetometer [ระบบออนไลน์] แหล่งที่มา <https://en.wikipedia.org/wiki/Magnetometer> (24 ตุลาคม 2562)
- [6] Kalman filter [ระบบออนไลน์] แหล่งที่มา https://en.wikipedia.org/wiki/Kalman_filter (24 ตุลาคม 2562)
- [7] I²C [ระบบออนไลน์] แหล่งที่มา <https://learn.sparkfun.com/tutorials/i2c> (24 ตุลาคม 2562)
- [8] SPI [ระบบออนไลน์] แหล่งที่มา <http://aimagin.com/blog/spi/?lang=th> (24 ตุลาคม 2562)
- [9] C++ คืออะไร [ระบบออนไลน์] แหล่งที่มา <https://mindphp.com/คู่มือ/73-คืออะไร/2183-c++-คืออะไร.html> (24 ตุลาคม 2562)
- [10] Xsens features [ระบบออนไลน์] แหล่งที่มา <https://www.xsens.com/products/mti-10-series> (24 ตุลาคม 2562)

[11] ดร. อนุรักษ์ ศุภกา. MEMS & NEMS [ระบบออนไลน์] แหล่งที่มา

http://www.tpa.or.th/publisher/pdfFileDownloadS/tn227b_p37-42.pdf. (18 กันยายน 2562)

[12] Dr. Scott Smith รูปหลักการการทำงานของ Piezoelectric accelerometers [ระบบออนไลน์] แหล่งที่มา

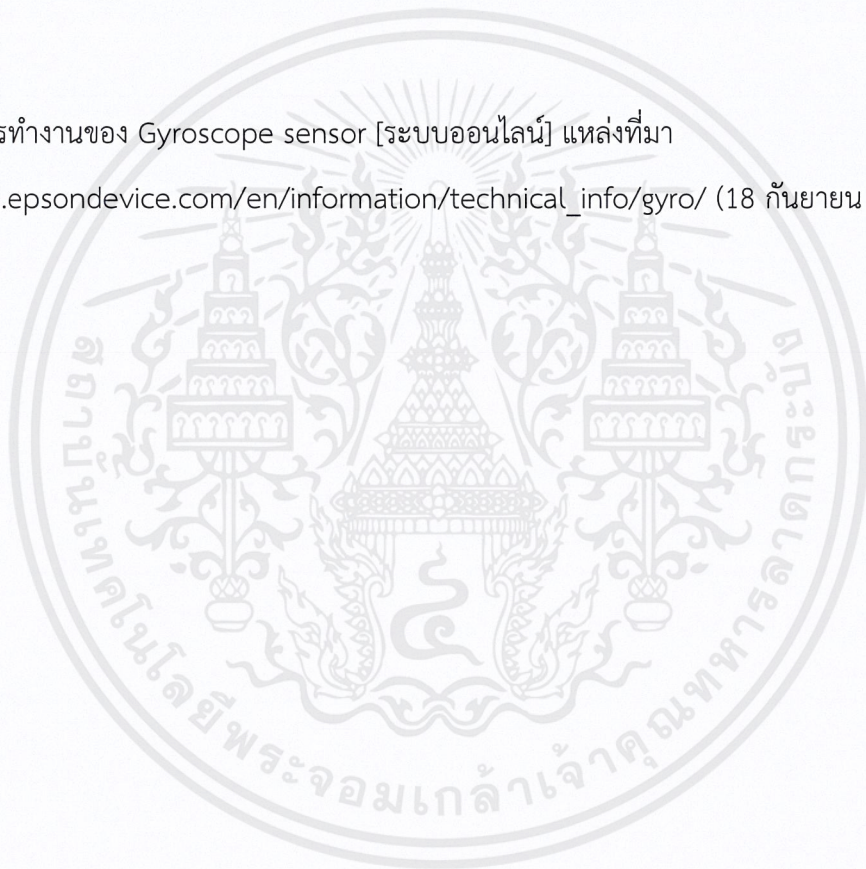
<https://www.ctemag.com/news/articles/measuring-machine-vibration> (18 กันยายน 2562)

[13] รูปวงจรภายใน MEMS [ระบบออนไลน์] แหล่งที่มา <https://www.rfwireless->

[world.com/Terminology/Difference-between-Accelerometer-and-Gyroscope.html](https://www.rfwireless-world.com/Terminology/Difference-between-Accelerometer-and-Gyroscope.html) (18 กันยายน 2562)

[14] รูปหลักการการทำงานของ Gyroscope sensor [ระบบออนไลน์] แหล่งที่มา

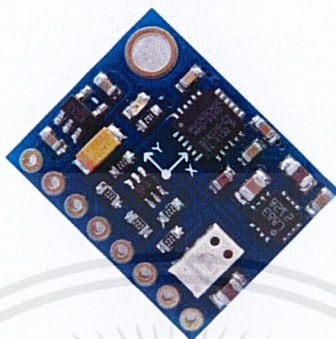
https://www5.epsondevice.com/en/information/technical_info/gyro/ (18 กันยายน 2562)



ภาคผนวก

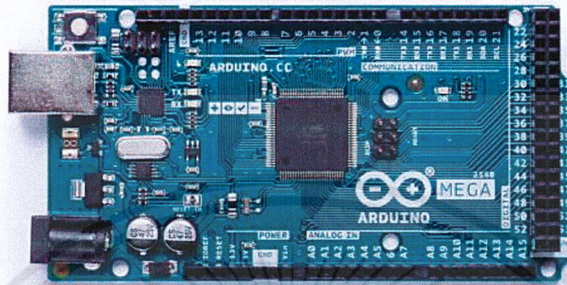


GY - 86



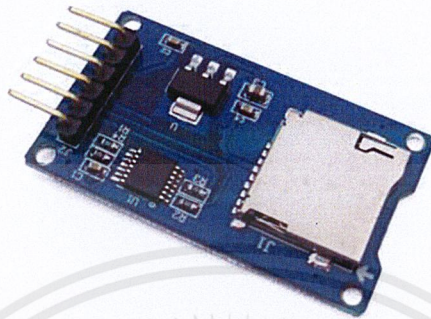
	GY - 86
ความกว้างการวัดความเร่ง	2,4,8,16 จี
สัญญาณรบกวนที่เกิดขึ้น	10 เฮิร์ต ในความเร่ง 10 g
คุณสมบัติอื่นๆเพิ่มเติม	มีไจโรสโคป
การจัดการกับ Bias จากการใช้งานเป็นเวลานาน	มีไจโรสโคปรวมในไอซีแล้ว จึงสามารถใช้ Kalman's Filter ได้เลย
ขนาด	22 x 17 x 3mm
ราคา	25.56 \$

Arduino Mega 2560 Rev 3

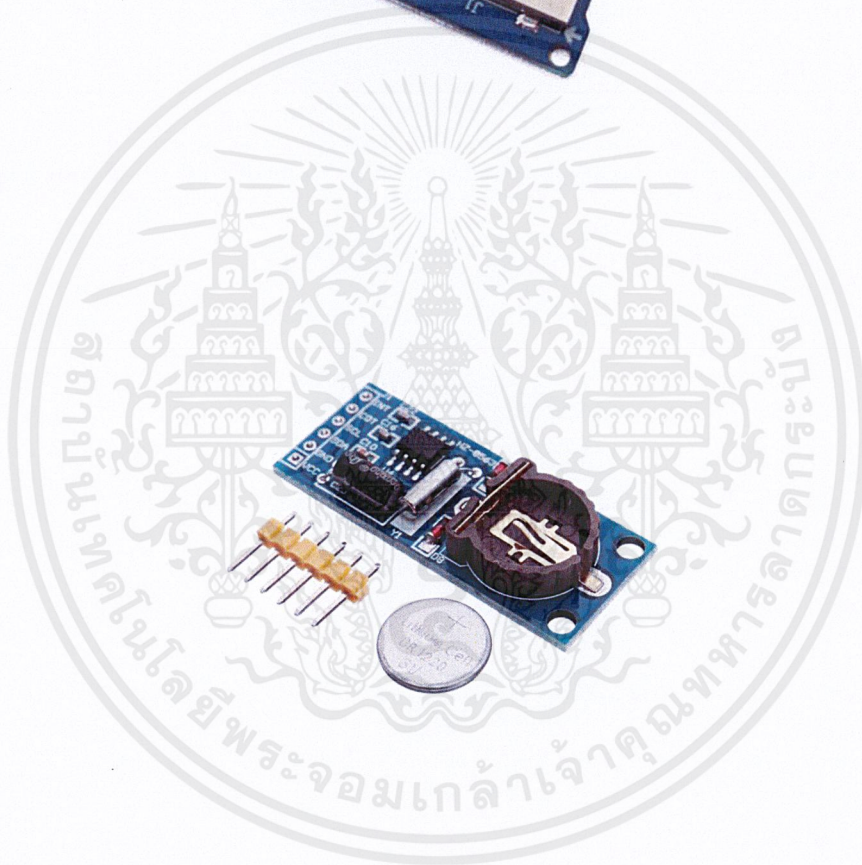


	Arduino Mega 2560 Rev 3
ไมโครคอนโทรลเลอร์	ATmega2560
กำลังไฟสำหรับการทำงาน	5V
จำนวนดิจิทัล I/O pin	54 pins
SRAM	8KB
Flash Memory	256KB
EEPROM	4KB
ความเร็วสัญญาณนาฬิกา	16MHz

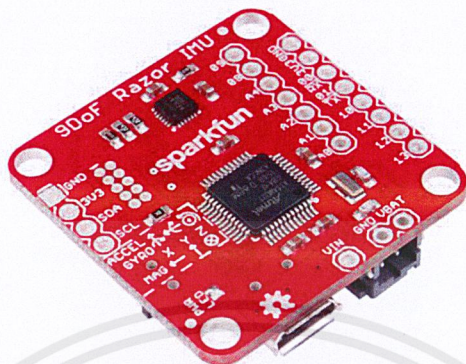
Micro SD Card SPI Reader



PCF 8563



9 DoF Razor IMU M0+



	9 DoF Razor IMU M0+
ไมโครโปรเซสเซอร์	SAMD21G18A
Inertial Measurement Unit	MPU - 9250
กำลังไฟฟ้าสำหรับทำงาน	3.3 โวลต์
ค่าความเร่งที่วัดได้สูงสุด	16 จี
คุณสมบัติอื่นๆ	มีไจโรสโคป, Magnetometer, ช่องจ่ายไฟกับแบตเตอรี่, ช่อง Micro SD Card SPI Reader
ขนาด	30 x 30 x 10 mm
ราคา	35.95 \$