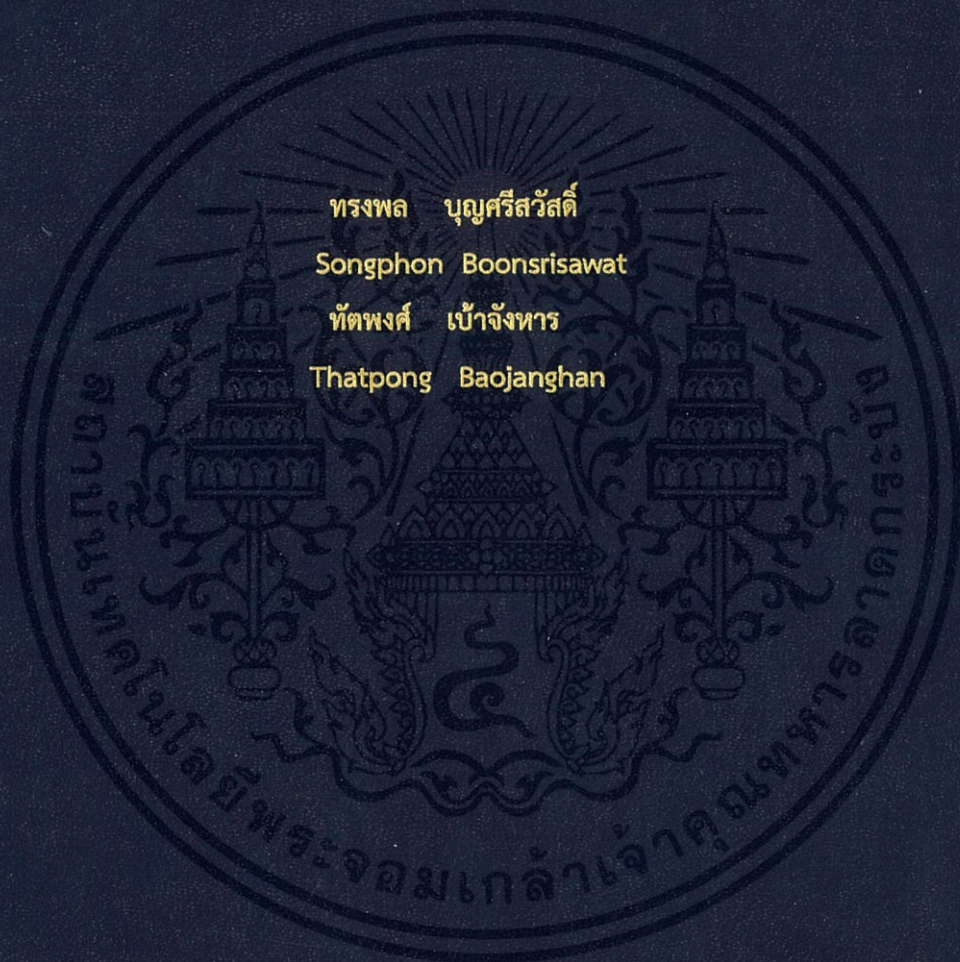


อุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูป

Gimbal Stabilizer for camera



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

อุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูป

Gimbal Stabilizer for camera



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูป

Gimbal Stabilizer for camera

โดย



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ ปีการศึกษา 2561

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง อุปกรณ์กันการสั่นสะเทือน

Gimbal Stabilizer for camera

ผู้จัดทำ นายทรงพล บุญศรีสวัสดิ์ รหัสประจำตัว 58010456

นายทัตพงศ์ เบ้าจันทาร รหัสประจำตัว 58010470

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



(ผศ.ดร.แสงระวี บัวแก้ว)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท	อุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูป
นักศึกษา	นายทรงพล บุญศรีสวัสดิ์ รหัสประจำตัว 58010456 นายทัตพงศ์ เบ้าจันทาร รหัสประจำตัว 58010470
ปริญญา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2561
อาจารย์ที่ปรึกษาโครงการ	ผศ.ดร.แสงระวี บัวแก้ว

### บทคัดย่อ

โครงการนี้นำเสนออุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูป (Gimbal Stabilizer for camera) โดยมีจุดประสงค์ที่ทำให้ตัวกล้องถ่ายรูปไม่เอียงไปตามแนวแกน X, Y และ Z เพื่อปรับสมดุลตัวกล้องถ่ายรูปโดยอุปกรณ์ที่สร้างขึ้นอาศัยหลักการการทำงานของไมโครคอนโทรลเลอร์, เซนเซอร์วัดความเร่งทั้ง 3 แกนและเซอร์โวมอเตอร์ทั้ง 3 ตัว การทำงานคือเมื่อไมโครคอนโทรลเลอร์ได้รับข้อมูลจากเซนเซอร์วัดความเร่ง จึงนำข้อมูลในรูปแบบดิจิทัลมาเพื่อคำนวณหามุมในแกนต่างๆ และทำการส่งชุดคำสั่งเพื่อไปควบคุมการทำงานของเซอร์โวมอเตอร์มอเตอร์ทั้ง 3 ตัว โครงการนี้เลือกใช้เซอร์โวมอเตอร์เพราะสามารถรับน้ำหนักได้มากและสามารถควบคุมองศาได้อย่างแม่นยำ

Thesis Title	Gimbal Stabilizer for camera	
Students	Mr.Songphon Boonsrisawat	Student ID 58010456
	Mr.Thatpong Baojanghan	Student ID 58010470
Degree	Bachelor of Engineering	
Program	Electronics Engineering	
Academic Year	2018	
Project Advisor	Asst.Prof.Dr.Seangrawee Buakeaw	

### ABSTRACT

This project presents a vibration protector devices for a camera. It known as a Gimbal stabilizer. The purpose of device is to protect the tilting along the x, y, and z axes of camera. The balance angle of camera can adjust and control by microprocessor. The acceleration sensor for all 3 axes, and all 3 servo motors is working when the microcontroller receives data from the acceleration sensor. Therefore the information in digital form is transfer to calculate the angle in various axes. Then the microcontroller send the instruction set to control the operation of all 3 servo motors. This project uses a servo motor because it can handle a lot of weight and can control degrees precisely.

### กิตติกรรมประกาศ

โครงการ “อุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายภาพ” สำเร็จลุล่วงได้ด้วยดี จากความช่วยเหลือและ ให้คำปรึกษาจาก ผศ.ดร.แสงระวี บัวแก้ว อาจารย์ที่ปรึกษา ที่คอยช่วยเหลือในการทดลองวงจรและให้ความรู้รวมถึงการแก้ปัญหา เมื่อผลิตชิ้นงานออกมา และรวมถึงเอื้อเพื่ออุปกรณ์และสถานที่ในการทดสอบและผลิตชิ้นงาน คณะผู้จัดทำโครงการ ขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณพ่อแม่ของคณะผู้จัดทำที่คอยเป็นกำลังใจหลักและเป็นผู้สนับสนุนเงินทุนหลัก ในการทำโครงการ รวมไปถึงเพื่อนๆเป็นกำลังหลักคอยช่วยเหลือ ให้คำแนะนำ รวมทั้งช่วยกันในการแก้ปัญหาที่เกิดขึ้นในระหว่างการทำโครงการนี้ทั้งหมดทำให้ผลของโครงการสำเร็จลุล่วงไปด้วยดี

สุดท้ายนี้ คณะผู้จัดทำหวังว่าโครงการนี้จะเป็นประโยชน์สำหรับผู้สนใจและผู้นำผลงานนี้ไปใช้งานได้

ทรงพล บุญศรีสวัสดิ์

ทัตพงศ์ เป้าจันหาร



## สารบัญ

## หน้า

บทคัดย่อภาษาไทย.....	A
บทคัดย่อภาษาอังกฤษ.....	B
กิตติกรรมประกาศ.....	C
สารบัญ.....	D,E
สารบัญรูป.....	F,G
สารบัญตาราง.....	H
บทที่ 1 บทนำ.....	1
1.1. ที่มาและความสำคัญของปัญหา.....	2
1.2. ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	2
1.3. สมมุติฐานการศึกษา.....	2
1.4. ขอบเขตของการศึกษา.....	2
บทที่ 2 หลักการและทฤษฎี.....	3
2.1. หลักการเบื้องต้น.....	3
2.2. ทฤษฎีที่เกี่ยวข้อง.....	3
2.2.1. Servo.....	5
2.2.2. Arduino.....	5
2.2.3. Accelerometers(MPU6050).....	11
2.2.4. ATmega328P.....	12
2.2.5. I2C.....	14
บทที่ 3 วิธีการดำเนินงาน.....	21

3.1.คุณสมบัติของอุปกรณ์กันการสั่นสะเทือนสำหรับกล้อง.....	21
3.2.ส่วนประกอบของวงจรไมโครคอนโทรลเลอร์.....	21
3.3.ออกแบบการทำงานของอุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูป.....	22
3.4.ออกแบบวงจร Microcontroller.....	22
3.5.โปรแกรมควบคุม.....	23
3.6.ส่วนของโครงสร้าง.....	25
บทที่ 4 ผลการทดลอง.....	28
4.1.อุปกรณ์ที่ใช้ในการทดลอง.....	28
4.2.ผลการทดลอง.....	28
บทที่ 5 สรุปผลการทดลอง.....	33
5.1.สรุปผลการทดลอง.....	33
5.2.ปัญหาและอุปสรรค.....	33
5.3.แนวทางแก้ไขปัญหา.....	33
เอกสารอ้างอิง.....	34
ภาคผนวก.....	35

## สารบัญรูป

## หน้า

รูปที่ 2.1 รายละเอียดของ ATmega328P.....	4
รูปที่ 2.2 Aduino UNO R3.....	5
รูปที่ 2.3 Layout & Pin Out Arduino Board (Model : Aduino UNO R3).....	6
รูปที่ 2.4 การเชื่อมต่อ computer กับบอร์ด Aduino.....	7
รูปที่ 2.5 Accelerometers (MPU6050).....	11
รูปที่ 2.6 Block Daigram(MPU6050).....	12
รูปที่ 2.7 ส่วนประกอบของ ATmega328P.....	12
รูปที่ 2.8 การเชื่อมต่ออุปกรณ์แบบ I2C.....	14
รูปที่ 2.9 SCL SDA.....	15
รูปที่ 2.10 ภาวะที่เกิดขึ้นบนบัส I2C.....	15
รูปที่ 2.11 สภาวะบนระบบบัส.....	16
รูปที่ 2.12 รูปแบบการสื่อสารแบบ I2C.....	17
รูปที่ 2.13 การเข้าถึงแบบ 7 บิต.....	18
รูปที่ 2.14 การเข้าถึงแบบ 10 บิต.....	19
รูปที่ 2.15 ลำดับการทำงาน (Timing Diagram).....	20
รูปที่ 3.1 วงจรบอร์ดควบคุม (skemetic).....	21
รูปที่ 3.2 วงจรบอร์ดควบคุม (PCB).....	21
รูปที่ 3.3 บล็อกไดอะแกรมแสดงการทำงาน.....	22
รูปที่ 3.4 วงจร Microcontroller.....	22

รูปที่ 3.5 Flowchart โปรแกรมควบคุม.....	23
รูปที่ 3.6 ข้อต่อ 3 ทาง (แผ่นอะคริลิก).....	25
รูปที่ 3.7 ข้อต่อ 90 องศา (แผ่นอะคริลิก).....	25
รูปที่ 3.8 ข้อต่อเซอร์โว (แผ่นอะคริลิก).....	26
รูปที่ 3.9 อลูมิเนียมโปรไฟล์กับแผ่นอะคริลิกตอนประกอบ.....	26
รูปที่ 3.10 แขนของ Gimbal Stabilizer for camera.....	27
รูปที่ 3.11 Servo Motor.....	27
รูปที่ 3.12 อุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูป.....	27
รูปที่ 4.1 สัญญาณ PWM ที่ส่งให้ Servo Motor.....	28
รูปที่ 4.2 กราฟค่าที่ได้จากเซนเซอร์ผ่านไลบรารี.....	32

## สารบัญตาราง

	หน้า
ตารางที่ 2.1 รายละเอียดของ ATmega328P.....	13
ตารางที่ 4.1 แกน X.....	29
ตารางที่ 4.2 แกน Y.....	30
ตารางที่ 4.3 แกน Z.....	31



## บทที่ 1

### บทนำ

#### 1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบันเทคโนโลยีในด้านกล้องถ่ายวิดีโอมีการพัฒนาอย่างต่อเนื่อง โดยทำให้กล้องถ่ายวิดีโอถ่ายวิดีโอได้ชัดเจนมากขึ้น หลักการทำงานของกล้องถ่ายภาพ คือ การที่แสงสะท้อนจากวัตถุเดินทางเป็นเส้นตรงผ่านช่องเล็กๆ ของกล้องสี่เหลี่ยม เกิดภาพของวัตถุบนฉากรองรับด้านตรงกันข้ามเป็นภาพหัวกลับ อันเป็นหลักการของการสร้างกล้องรูเข็มในสมัยโบราณ ปัจจุบัน กล้องถ่ายภาพได้พัฒนาโดยลำดับ เช่น มีการนำเอาเลนส์นูนไปติดตั้งที่ช่องรับแสงที่มีขนาดเล็ก เพื่อช่วยรวมแสงให้เข้าไปในตัวกล้องให้มากขึ้น ทางด้านตรงกันข้ามของเลนส์เป็นตำแหน่งที่ตั้งวัสดุไวแสงหรือฟิล์ม สามารถปรับตัวเลนส์เพื่อให้เกิดภาพที่ชัดเจนบนฟิล์มได้ มีการติดตั้งไดอะแฟรมปรับให้เกิดช่องรับแสงขนาดต่างๆ รวมทั้งมีส่วนที่เรียกว่าชัตเตอร์ ทำหน้าที่ควบคุมเวลาในการเปิด-ปิด ม่าน เพื่อให้ปริมาณแสงตกกระทบกับฟิล์มตามความเหมาะสม และยังมีช่องเล็งภาพเพื่อช่วยในการจัดองค์ประกอบของภาพถ่ายให้เกิดความสวยงาม แต่ไม่ได้พัฒนาให้ตัวกล้องถ่ายวิดีโอไม่ให้สั่นไหวเกิดขึ้น จึงเป็นการประดิษฐ์ อุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูป (Gimbal Stabilizer for Camera) โดยอาศัยหลักการ กิมบอล (Gimbal) คือ อุปกรณ์มอเตอร์ 3 แกน หมุนอิสระ ทำให้ตัวกล้องตั้งตรงคงที่อยู่ในทิศทางเดียวกับแนวเส้นขอบฟ้า หรือตั้งให้หันไปในทิศทางอื่นตามต้องการ ควบคุมกล้องไม่ให้สั่นไหวเคลื่อนที่ไปตามมือที่ถือ หรือวัตถุอื่นๆ ที่มีการเคลื่อนที่ ป้องกันอาการสั่นไหวของภาพ ทำให้ได้ภาพที่นิ่งและตั้งตรงเสมอ อุปกรณ์นี้เป็นอุปกรณ์เสริม ติดกับตัวกล้อง DSLR ช่วยทำให้การถ่ายวิดีโอ เป็นไปอย่างราบรื่นขึ้น ไม่มีการสั่นไหวของภาพ Gimbal Stabilizer for Camera มากับ มอเตอร์ 3 ตัว ถูกติดตั้งในแนวแกน x, y, z และ sensor วัดระดับการสั่นของตัวเฟรมกล้อง ซึ่งมอเตอร์ถูกควบคุมด้วยวงจรไมโครคอนโทรลเลอร์ และพลังงานไฟฟ้าที่ได้จาก battery Li-Fe เป็นตัวขับเคลื่อนมอเตอร์กันสั่น เมื่อทำการถ่ายวิดีโอที่ต้องมีการเคลื่อนไหวของตัวผู้ถ่าย เช่น การเดิน การวิ่ง หรือแม้แต่ตั้งมอเตอร์ไฮดรอลิก อุปกรณ์ Gimbal Stabilizer for Camera คือ หัวใจของการป้องกันการสั่นไหว ความนิ่ง การถ่ายโดยไม่มีอุปกรณ์เสริม ก็จะทำให้ได้ภาพที่มีการสั่นไหวของวิดีโอที่ได้ ทำให้ง่ายแก่การตัดต่อ และทำให้เลือกมุมกล้องที่ถ่ายได้น้อยลง หากมีอุปกรณ์ Camera Gimbal Stabilizer แล้วปัญหาการสั่นไหวของวิดีโอจะหมดไป ทำให้วิดีโอที่ได้มีคุณภาพ ดังนั้นอุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูปจึงมีความสำคัญที่เข้ามามีบทบาทในการถ่ายวิดีโอให้มีคุณภาพที่ดีขึ้น

## 1.2 วัตถุประสงค์ของการศึกษา

- 1.2.1 นำความรู้ที่ได้จากการสอนภาคทฤษฎีมาใช้ในทางปฏิบัติ
- 1.2.2 ได้ศึกษาหาความรู้เพิ่มเติมเกี่ยวกับหลักการทำงานของวงจรไมโครคอนโทรลเลอร์
- 1.2.3 เป็นการฝึกปฏิบัติของคณะผู้จัดทำเพื่อเป็นพื้นฐานในภาคปฏิบัติต่อไป

## 1.3 ขอบเขตการศึกษา

- 1.3.1 สามารถทำให้ตัวกล้องตั้งตรงคงที่อยู่ในทิศทางเดียวกับแนวเส้นขอบฟ้า
- 1.3.2 ใช้ Microcontroller ในการควบคุมการทำงาน
- 1.3.3 ตัวอุปกรณ์มีความแข็งแรง และ รับน้ำหนักได้มาก

## 1.4 ผลที่คาดว่าจะได้รับ

- 1.4.1 อุปกรณ์สามารถควบคุมกล้องไม่ให้สั่นไหว
- 1.4.2 อุปกรณ์สามารถทำให้ได้ภาพที่นิ่งและตั้งตรงเสมอ
- 1.4.3 อุปกรณ์สามารถรับน้ำหนักกล้องถ่ายรูปได้



## บทที่ 2

### หลักการและทฤษฎี

#### 2.1 หลักการเบื้องต้น

หลักการงานเบื้องต้นของอุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูปนั้น จะถูกควบคุมด้วยบอร์ด Microcontroller โดยการเขียนชุดคำสั่งการทำงานควบคุมให้อุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูป ขยับกล้องเข้าสู่สมดุล ผ่านการส่งข้อมูลจาก MPU6050 อุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูปอาศัยพลังงานจากแบตเตอรี่ในการทำงานโดยไม่ต้องเสียบปลั๊กขณะใช้งาน และชาร์ตแบตเตอรี่เก็บไว้เมื่อหยุดการใช้งาน อุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูปเป็นเครื่องทุ่นแรงในการทำให้กล้องถ่ายรูปสมดุลเมื่อมีการสั่นสะเทือนหรือเคลื่อนที่

#### 2.2 ทฤษฎีที่เกี่ยวข้อง

2.2.1 Servo เป็นคำศัพท์ที่ใช้กันทั่วไปในระบบควบคุมอัตโนมัติ มาจากภาษาละตินคำว่า Sevus หมายถึง “ทาส” (Slave) ในเชิงความหมายของ Servo Motor ก็คือ Motor ที่เราสามารถสั่งงานหรือตั้งค่าแล้วตัว Motor จะหมุนไปยังตำแหน่งองศาที่เราสั่งได้เองอย่างถูกต้อง โดยใช้การควบคุมแบบป้อนกลับ (Feedback Control) ในบทความนี้จะกล่าวถึง RC Servo Motor ซึ่งนิยมนำมาใช้ในเครื่องเล่นที่บังคับด้วยคลื่นวิทยุ (RC = Radio - Controlled) เช่น เรือบังคับวิทยุ รถบังคับวิทยุ เฮลิคอปเตอร์บังคับวิทยุ เป็นต้น

2.2.1.1 Feedback Control คือ ระบบควบคุมที่มีการวัดค่าเอาต์พุตของระบบนำมาเปรียบเทียบกับค่าอินพุตเพื่อควบคุมและปรับแต่งให้ค่าเอาต์พุตของระบบให้มีค่า เท่ากับ หรือ ใกล้เคียงกับค่าอินพุต

##### 2.2.1.2 ส่วนประกอบภายนอก RC Servo Motor

- Case ตัวถัง หรือ กรอบของตัว Servo Motor
- Mounting Tab ส่วนจับยึดตัว Servo กับชิ้นงาน
- Output Shaft เพลาส่งกำลัง
- Servo Horns ส่วนเชื่อมต่อกับ Output shaft เพื่อสร้างกลไก
- Cable สายเชื่อมต่อเพื่อ จ่ายไฟฟ้า และ ควบคุม Servo Motor จะประกอบด้วยสายไฟ 3 เส้น และ ใน RC Servo Motor จะมีสีของสายแตกต่างกันไปดังนี้
  - o สายสีแดง คือ ไฟเลี้ยง (4.8-6V)
  - o สายสีดำ หรือ น้ำตาล คือ กราวด์
  - o สายสีเหลือง (ส้ม ขาว หรือฟ้า) คือ สายส่งสัญญาณพัลส์ควบคุม (3-5V)
- Connector จุดเชื่อมต่อสายไฟ

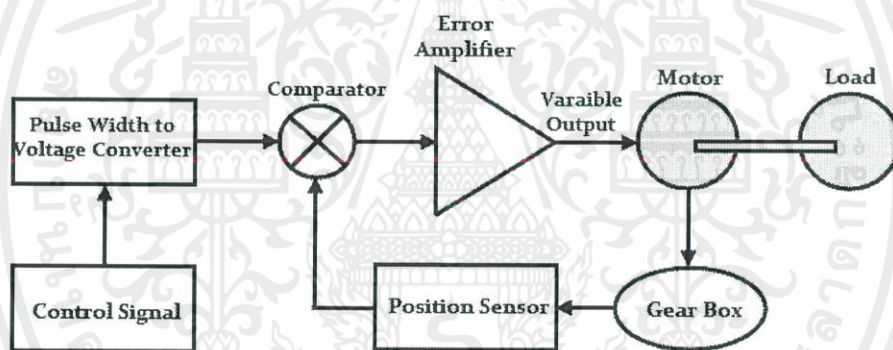
### 2.2.1.3 ส่วนประกอบภายใน RC Servo Motor

1. Motor เป็นส่วนของตัวมอเตอร์
2. Gear Train หรือ Gearbox เป็นชุดเกียร์ทดแรง
3. Position Sensor เป็นเซ็นเซอร์ตรวจจับตำแหน่งเพื่อหาค่าองศาในการหมุน
4. Electronic Control System เป็นส่วนที่ควบคุมและประมวลผล

### 2.2.1.4 Servo Motor Block Diagram

#### หลักการการทำงานของ RC Servo Motor

เมื่อจ่ายสัญญาณพัลส์เข้ามายัง RC Servo Motor ส่วนวงจรควบคุม (Electronic Control System) ภายใน Servo จะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลส์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่งองศาที่ต้องการให้ Motor หมุนเคลื่อนที่ไปยังตำแหน่งนั้น แล้วส่งคำสั่งไปทำการควบคุมให้ Motor หมุนไปยังตำแหน่งที่ต้องการ โดยมี Position Sensor เป็นตัวเซ็นเซอร์คอยวัดค่ามุมที่ Motor กำลังหมุน เป็น Feedback กลับมาให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้องแม่นยำ



รูปที่ 2.1 Block Diagram Servo Motor

### 2.2.1.5 สัญญาณ RC ในรูปแบบ PWM

ตัว RC Servo Motor ออกแบบมาใช้สำหรับรับคำสั่งจาก Remote Control ที่ใช้ควบคุมของเล่น ด้วยสัญญาณวิทยุต่างๆ เช่น เครื่องบินบังคับ รถบังคับ เรือบังคับ เป็นต้น ซึ่ง Remote จำพวกนี้ที่ภาครับจะแปลงความถี่วิทยุออกมาในรูปแบบสัญญาณ PWM (Pulse Width Modulation)

มุมหรือองศาจะขึ้นอยู่กับความกว้างของสัญญาณพัลส์ ซึ่งโดยส่วนมากความกว้างของพัลส์ที่ใช้ใน RC Servo Motor จะอยู่ในช่วง 1-2 ms หรือ 0.5-2.5 ms

ยกตัวอย่างเช่นหากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1 ms ตัว Servo Motor จะหมุนไปทางซ้ายสุด ในทางกลับกันหากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 2 ms ตัว Servo Motor จะหมุนไปยังตำแหน่งขวาสุด แต่หากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1.5 ms ตัว Servo Motor ก็ จะหมุนมาอยู่ที่ตำแหน่งตรงกลางพอดี

ดังนั้นสามารถกำหนดองศาการหมุนของ RC Servo Motor ได้โดยการเทียบค่า เช่น RC Servo Motor สามารถหมุนได้ 180 องศา โดยที่ 0 องศาใช้ความกว้างพัลส์เท่ากับ 1000 us ที่ 180 องศาความกว้างพัลส์เท่ากับ 2000 us เพราะฉะนั้นค่าที่เปลี่ยนไป 1 องศาจะใช้ความกว้างพัลส์ต่างกัน  $(2000-1000)/180$  เท่ากับ 5.55 us

จากการหาค่าความกว้างพัลส์ที่มุม 1 องศาข้างต้น หากต้องกำหนดให้ RC Servo Motor หมุนไปที่มุม 45 องศาจะหาค่าพัลส์ที่ต้องการได้จาก  $5.55 \times 45$  เท่ากับ 249.75 us แต่ที่มุม 0 องศาเราเริ่มที่ความกว้างพัลส์ 1ms หรือ 1000 us เพราะฉะนั้นความกว้างพัลส์ที่ใช้กำหนดให้ RC Servo Motor หมุนไปที่ 45 องศา คือ  $1000 + 249.75$  เท่ากับประมาณ 1250 us

### 2.2.2 Arduino

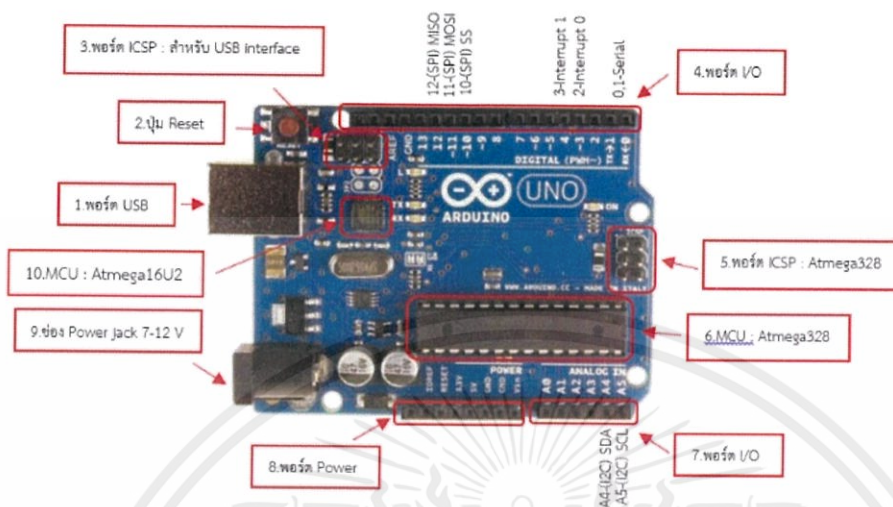
Arduino คือ บอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software โดยด้าน Hardware คือ บอร์ดอิเล็กทรอนิกส์ขนาดเล็กสามารถเคลื่อนย้ายพกพาได้สะดวกโดยมีไมโครคอนโทรลเลอร์เป็นอุปกรณ์หลักที่สำคัญและมีอุปกรณ์อิเล็กทรอนิกส์ต่างๆมาประกอบรวมกัน ต่อมาในด้าน Software ตัวบอร์ด Arduino เป็นบอร์ดที่ใช้ไมโครคอนโทรลเลอร์ดังนั้นจึงมีลักษณะภาษาแบบเดียวกับ ภาษา C/C++ ซึ่งเป็นภาษาสำหรับเขียนโปรแกรมที่แพร่หลาย ดังนั้นตัวบอร์ด Arduino จึงสามารถใช้งานได้ง่าย จึงเหมาะสำหรับผู้เริ่มต้นศึกษาทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติมพัฒนาต่อยอดทั้งตัวบอร์ดหรือโปรแกรมต่อได้อีกด้วย



รูปที่ 2.2 Arduino UNO R3

ความสะดวกและไม่ซับซ้อนของบอร์ด Arduino ทำให้สามารถต่ออุปกรณ์เสริมต่างๆได้อย่างสะดวก นั่นคือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์ต่างๆจากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ดหรือสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้

### 2.2.2.1 โครงสร้างของบอร์ด Arduino UNO R3



รูปที่ 2.3 Layout & Pin out Arduino Board (Model: Arduino UNO R3)

หมายเลข 1 USB Port : ใช้ในการต่อเข้ากับComputer เพื่ออัปโหลดโปรแกรมเข้าไมโครคอนโทรลเลอร์และจ่ายไฟ ให้กับบอร์ด

หมายเลข 2 Reset Button : ปุ่ม Reset ใช้เพื่อให้ไมโครคอนโทรลเลอร์เริ่มการทำงานใหม่

หมายเลข 3 ICSP Port : เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2

หมายเลข 4 I/O Port : Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่น

เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM

หมายเลข 5 ICSP Port : Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader

หมายเลข 6 MCU : Atmega328 เป็นไมโครคอนโทรลเลอร์ ที่ใช้นบนบอร์ด Arduino

หมายเลข 7 I/O Port : คือ Digital I/O และยังสามารถเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5

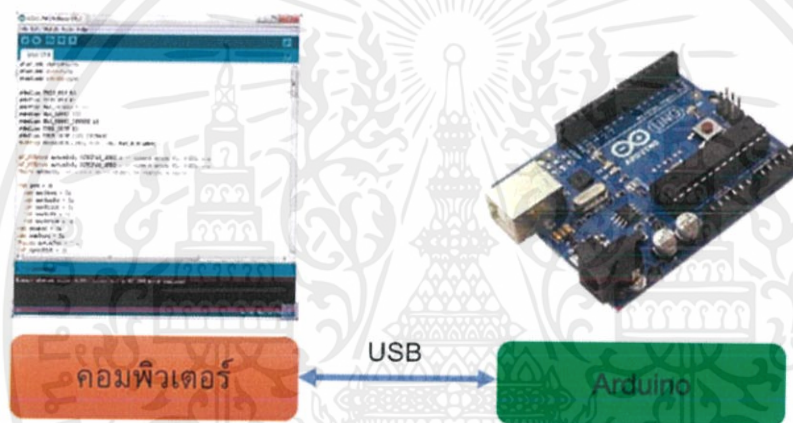
หมายเลข 8 Power Port : ไฟเลี้ยงของตัวบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอกประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, Vin

หมายเลข 9 Power Jack : ส่วนที่รับไฟจาก Adapter และมีแรงดันอยู่ระหว่าง 7-12 V

หมายเลข 10 MCU ของ Atmega16U2 : เป็นไมโครคอนโทรลเลอร์ ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

## 2.2.2.2 การเขียนโปรแกรมพื้นฐาน Arduino

Arduino ใช้พื้นฐานลักษณะภาษาแบบเดียวกับภาษา C/C++ ดังนั้นจึงมีโครงสร้างเหมือนกับภาษา C/C++ ทั่วไปนั่นเองทำให้ง่ายต่อการศึกษาใช้งานสำหรับผู้ที่มีความรู้ภาษา C/C++ สำหรับผู้เริ่มใช้ Arduino มีหลักสำคัญที่ต้องรู้เกี่ยวกับคำสั่งพื้นฐาน Arduino ดังนี้  
 1. **พรีโพรเซสเซอร์ไดเรกทีฟ (Preprocessor directives)** คือส่วนที่คอมไพเลอร์จะทำการประมวลผลและทำตามคำสั่งก่อนที่จะมีการคอมไพล์โปรแกรม ส่วนนี้จะเริ่มด้วยเครื่องหมายไดเรกทีฟ (directive) หรือสัญลักษณ์เครื่องหมายสี่เหลี่ยม # แล้วตามด้วยชื่อคำสั่งที่ต้องการใช้งาน หรือกำหนด ซึ่งในส่วนนี้จะอยู่ในส่วนบนสุด หรือส่วนหัวของโปรแกรม และต้องอยู่นอกฟังก์ชันหลักใดๆก็ตามถ้าให้เข้าใจง่ายๆนั้นคือ #include เป็นคำสั่งที่ใช้อ้างอิงไฟล์ภายนอก เพื่อเรียกใช้ฟังก์ชัน หรือตัวแปรที่มีการสร้างหรือกำหนดไว้ในไฟล์นั้น



รูปที่ 2.4 การเชื่อมต่อ Computer กับบอร์ด Arduino

## 2.2.2.3 รูปแบบการใช้งานคือ #include <ชื่อไฟล์.h>

ตัวอย่างเช่น

```
#include <Wire.h> //อ้างอิงไฟล์ Wire.h โดยเป็นไลบรารีพื้นฐานที่มีอยู่ใน Arduino
#include <Time.h> // อ้างอิงไฟล์ Time.h โดยเป็นไลบรารีพื้นฐานที่มีอยู่ใน Arduino
```

การอ้างอิงไฟล์จากภายใน หรือไฟล์ไลบรารีที่มีอยู่แล้วใน Arduino หรือเป็นไลบรารีที่เราเพิ่มเข้าไปเอง จะใช้เครื่องหมาย <> ในการคร่อมชื่อไฟล์ไว้ โดยโปรแกรมคอมไพเลอร์จะเข้าใจว่าจะสามารถไปหาไฟล์ที่สั่งนี้จากในโฟลเดอร์ไลบรารีนั่นเอง และในอีกกรณีถ้าต้องการอ้างอิงไฟล์ที่อยู่ในโฟลเดอร์โปรเจกต์ จะใช้เครื่องหมาย "" คร่อมแทน ซึ่งคอมไพเลอร์จะวิ่งไป หาไฟล์นี้โดย อ้างอิงจากไฟล์โปรแกรมที่คอมไพเลอร์อยู่ ต่อมาเป็นมีความหมายคือ #define เป็นคำสั่งที่ใช้แทนข้อความที่กำหนดไว้ ด้วยข้อความที่กำหนดไว้ ซึ่งการใช้คำสั่งนี้ ข้อดีคือไม่มีการอ้างอิงกับตัวโปรแกรมเลย

### 2.2.2.4 รูปแบบการใช้งานคือ #define NAME VALUE

ตัวอย่างเช่น

#define LEDPIN 14 // ไม่ว่าคำว่า LEDPIN จะอยู่ส่วนใดของโค้ดโปรแกรมก็ตาม คอมไพเลอร์จะแทนคำว่า LEDPIN ด้วยเลข 14 แทนซึ่งข้อดีคือเราไม่ต้องสร้างเป็นตัวแปรขึ้นมาเพื่อเปลืองพื้นที่แรม และยังช่วยให้โปรแกรมทำงานเร็วขึ้นอีกด้วยเพราะซีพียูไม่ต้องไปขอข้อมูลมาจากแรมหลายๆทอดส่วนของการกำหนดค่า (Global declarations) เป็นส่วนที่ใช้ในการกำหนดชนิดตัวแปรแบบนอกฟังก์ชัน หรือประกาศฟังก์ชัน เพื่อให้ฟังก์ชันที่ประกาศสามารถกำหนด หรือเรียกใช้ได้จากทุกส่วนของโปรแกรม เช่น int pin = 13; // ประกาศตัวแปรนอกฟังก์ชัน ฟังก์ชัน setup() และฟังก์ชัน loop() คือคำสั่งที่ถูกบังคับให้ต้องมีในทุกโปรแกรม โดยฟังก์ชัน setup() เป็นฟังก์ชันแรกที่ถูกเรียกใช้ ใช้กำหนดค่าต่างๆ กำหนดขาในการใช้งานให้เป็น input หรือ output หรือเริ่มต้นใช้งานไลบรารีต่างๆ เช่น ในฟังก์ชัน setup() จะมีคำสั่ง pinMode() เพื่อกำหนดให้ขาใดๆก็ตามเป็น ดิจิตอลอินพุต หรือเอาต์พุตโดยการกำหนดในส่วนนี้จะกำหนดเพียงครั้งเดียวเท่านั้น ส่วนฟังก์ชัน loop() จะเป็นฟังก์ชันที่ทำงานหลังจากฟังก์ชัน setup() ได้ทำงานเสร็จสิ้นไปแล้ว โดยการทำงานจะมีการวนรูปแบบไม่รู้จบไปเรื่อยๆหรือกล่าวได้ว่าเมื่อฟังก์ชัน loop() ทำงานตามคำสั่งครบแล้ว ฟังก์ชัน loop() ก็จะถูกเรียกขึ้นมาใช้อีกเป็นแบบนี้ไปเรื่อยๆ

ตัวอย่างเช่น

```
int pin = 13;           // ประกาศตัวแปรนอกฟังก์ชัน
void setup() {
  pinMode(pin, OUTPUT); // ฟังก์ชันsetup()กำหนดให้ขาที่ 13 เป็นดิจิตอลเอาต์พุต
}
void loop() {
  digitalWrite(pin, HIGH); // ฟังก์ชัน loop() มีการกำหนดให้พอร์ต 13 มีลอจิกเป็น 1
  delay(1000);             // ฟังก์ชัน delay() ในการหน่วงเวลา 1 วินาที
  digitalWrite(pin, LOW);  // ให้พอร์ต 13 มีสถานะลอจิกเป็น 0
  delay(1000);             // หน่วงเวลา 1 วินาที
}
```

จากตัวอย่างแสดงว่าในฟังก์ชัน loop() เมื่อจบคำสั่ง จะเริ่มทำฟังก์ชัน loop() ใหม่ไปเรื่อยๆ ผลที่ได้คือไฟกระพริบบนบอร์ด Arduino Uno ในพอร์ตที่ 13 ทำงานแบบไม่รู้จบนั่นคือความหมายของ loop การสร้างฟังก์ชัน และการใช้งานฟังก์ชัน (Users-defined function) คำสั่งต่างๆที่อยู่ภายในฟังก์ชัน ต้องอยู่ภายใต้เครื่องหมายปีกกาเปิด { และปีกกาปิด } เท่านั้น ภายใต้เครื่องหมาย {} เราสามารถนำฟังก์ชันหรือคำสั่งใดๆก็ได้มาใส่ไว้ แต่จะต้องคั่นแต่ละคำสั่งด้วยเครื่องหมายเซมิโคลอน ; โดยจะนำคำสั่งทั้งหมดไว้บรรทัดเดียวกันเลย หรือแยกบรรทัดกันก็ได้เพื่อความสวยงามของโค้ด

#### ตัวอย่างเช่น

```
void Mode(int pin) {
pinMode(pin, OUTPUT);    // ฟังก์ชันsetup()กำหนดให้pinที่กำหนดเป็นดิจิตอลเอาต์พุต
}

void setup() {
Mode(13);
}
```

ส่วนอธิบายโปรแกรม (Program comments) มีความหมายอีกอย่างว่าการคอมเม้นโปรแกรมเป็นส่วนที่สำคัญอย่างมากที่จะช่วยให้ผู้ที่ไม่ได้เขียนโปรแกรม หรือเป็นผู้เขียนโปรแกรมเข้าใจในตัวโปรแกรมได้ง่ายขึ้นโดยสามารถอ่านได้จากคอมเม้น แทนการทำความเข้าใจโปรแกรมโดยอ่านแต่ละฟังก์ชัน โดยส่วนอธิบายโปรแกรม หรือส่วนคอมเม้นนี้ จะไม่มีผลกระทบต่อใดๆกับขนาดของโปรแกรมหลังคอมไพล์ นั่นเพราะส่วนนี้จะถูกตัดทิ้งทั้งหมดเนื่องจากไม่ได้ถูกนำไปใช้งาน โดยจะมีผลเพียงแค่ว่าไฟล์โค้ดของโปรแกรมจะใหญ่ขึ้นมา ถ้ามีการคอมเม้นโค้ดเยอะๆ ขนาดก็จะเพิ่มขึ้นตามตัวอักษร ดังนั้นการคอมเม้นโค้ดจึงไม่คิดพื้นที่มากนัก แต่ผู้เขียนแนะนำให้คอมเม้นโค้ดให้สั้น และกระชับเพื่อให้เกิดความรวดเร็วในการทำความเข้าใจ การคอมเม้นโค้ดมีอยู่ 3 รูปแบบ คือเปิดด้วย /\* และปิดด้วย \*/ และแบบสุดท้าย // เป็นการคอมเม้นโค้ดแบบข้ามบรรทัด คือตราบใดที่ยังไม่มี \*/ตรงส่วนนั้นจะเป็นคอมเม้นทั้งหมด

#### 2.2.2.5 คำสั่งพื้นฐานในชิป

- pinMode (ขาที่จะใช้ ,OUTPUT) = ตั้งค่าให้ขาเป็นเอาต์พุตใช้ได้ทั้งอะนาล็อกและดิจิตอล
- pinMode (ขาที่จะใช้ ,INPUT) = ตั้งค่าให้ขาเป็นอินพุตใช้ได้ทั้งอะนาล็อกและดิจิตอล
- digitalWrite (ขาที่จะใช้,สถานะที่จะแสดง) = ใช้สั่งขาเป็นเอาต์พุตดิจิตอล เช่น LED หน้าจอ
- digitalRead (ขาที่จะใช้) = ใช้สั่งขาอ่านอินพุตดิจิตอล เช่น สวิตช์กด เช่น เซอร์วาล์ว
- analogWrite (ขาที่จะใช้) = ใช้สั่งขาเป็นเอาต์พุตอนาล็อก เช่น ICควบคุมความเร็วมอเตอร์

- analogRead ( ขาที่จะใช้ ) = ใช้ส่งขานั้นรับอินพุทอนาลอก เช่น Rเก็อกมา LDR เซ็นเซอร์วัดระยะทาง
- delay ( เวลาหน่วยเป็นมิลลิวินาที ) = ใช้หน่วงเวลาทำงานก่อนทำงานคำสั่งต่อไป
- delayMicroseconds ( เวลาหน่วยเป็นไมโครวินาที ) = ใช้หน่วงเวลาทำงานก่อนทำงานคำสั่งต่อไป
- int = ใช้ตั้งชื่อตัวแปรสำหรับตั้งค่าเริ่มต้นที่เป็นตัวเลขหรือจองพื้นที่เก็บตัวเลขหรือตั้งชื่อให้ขา
- char = ใช้ตั้งชื่อตัวแปรที่ใช้เก็บตัวอักษร(จองพื้นที่ใช้เก็บอักษร)
- Serial.begin (9600) = ตั้งค่าเริ่มต้นเพื่อติดต่อสื่อสารกับคอมพิวเตอร์อัตราเร็ว9600บิตต่อวินาที
- Serial.print ( “ คำพูด ” ) = ใช้พิมพ์คำพูดเพื่อให้แสดงผลบนจอคอมแบบไม่เว้นบรรทัด
- Serial.println ( “ คำพูด ” ) = ใช้พิมพ์คำพูดเพื่อให้แสดงผลบนจอคอมแบบเว้นบรรทัด
- Serial.available () = ใช้ตรวจสอบว่ามีการกดคีย์บอร์ดหรือไม่
- Serial.Read () = ใช้อ่านค่าปุ่มคีย์บอร์ดคำสั่งที่เกี่ยวข้องกับคณิตศาสตร์
- ชื่อตัวแปร++ = เพิ่มค่าตัวแปรนั้นทีละหนึ่ง = +1
- ชื่อตัวแปร-- = ลดค่าตัวแปรนั้นทีละหนึ่ง = -1
- / = ทหาร
- = คูณ
- + = บวก
- \_ = ลบ
- = = เท่ากับ
- == = เท่ากันกับ(สามารถใช้เปรียบเทียบคำสั่งกับคำสั่งก็ได้ไม่จำเป็นต้องใช้กับตัวเลข)หรือเทียบความเหมือนกัน
- % = ทหารเอาเศษ เช่น 6%3 =0 เพราะ 6 ทหาร 3 ลงตัว ,6%4 เพราะ 6 ทหาร 4 ได้ 1 เหลือเศษ
- != = ไม่เท่ากับ
- pi = 22/7
- pow(x,y) = ยกกำลัง เช่น pow(2,3) = 23
- sqrt(x) = รากที่สอง
- = มากกว่า
- < = น้อยกว่า
- >= = มากกว่าหรือเท่ากับ
- <= = น้อยกว่าหรือเท่ากับ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.2.6. ชนิดข้อมูลพื้นฐานสำหรับ Arduino C/C++

- Byte ใช้สำหรับข้อมูลที่เป็นเลขจำนวนเต็มตั้ง 0 ถึง 255
- Int ใช้สำหรับข้อมูลที่เป็นเลขจำนวนเต็มได้ตั้งแต่ -32768 ถึง +32767
- Long ใช้สำหรับข้อมูลที่เป็นเลขจำนวนเต็มได้ตั้งแต่ -2,147,483,648 ถึง +2,147,483,647
- Float ใช้สำหรับข้อมูลที่เป็นเลขทศนิยม เป็นค่าที่เป็นบวกหรือลบในช่วงที่กว้างกว่าชนิดข้อมูลแบบ byte และ int และเป็นตัวเลขทศนิยมได้ด้วยแต่มีความละเอียดเพียง 6-7 ตำแหน่งหลังจุดทศนิยมในเลขฐานสิบ
- Boolean ใช้สำหรับข้อมูลที่เป็นค่าทางลอจิก true (จริง) หรือ false (เท็จ) เท่านั้น

### 2.2.3 Accelerometers (MPU6050)

#### 2.2.3.1 ข้อมูลเบื้องต้น (Introduction / Overview)

GY-521 เป็นโมดูล Accelerometers ใช้ในการตรวจสอบทิศทางการเคลื่อนที่ และสามารถใช้ในการตรวจสอบความเร็วในการเปลี่ยนแปลงทิศทาง ของแกน XYZ ได้ ยกตัวอย่าง ถ้าวัตถุเกิดการเคลื่อนที่หรือเอียง output ของ Accelerometer จะบอกค่าของการเอียงว่าสถานะปัจจุบันค่าของ XYZ อยู่ที่เท่าไร

#### 2.2.3.2 คุณสมบัติ (Features)

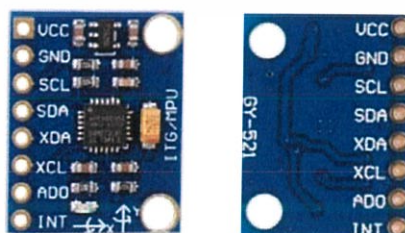
- ใช้ไฟเลี้ยง +3.3V ถึง +5V
- ชิพ MPU6050

#### 2.2.3.3 การนำไปประยุกต์ใช้งาน (Application Ideas)

ตรวจสอบการเคลื่อนที่ เคลื่อนไหวต่างๆของวัตถุ

#### 2.2.3.4 ข้อควรระวังการใช้งาน (Caution / Warning)

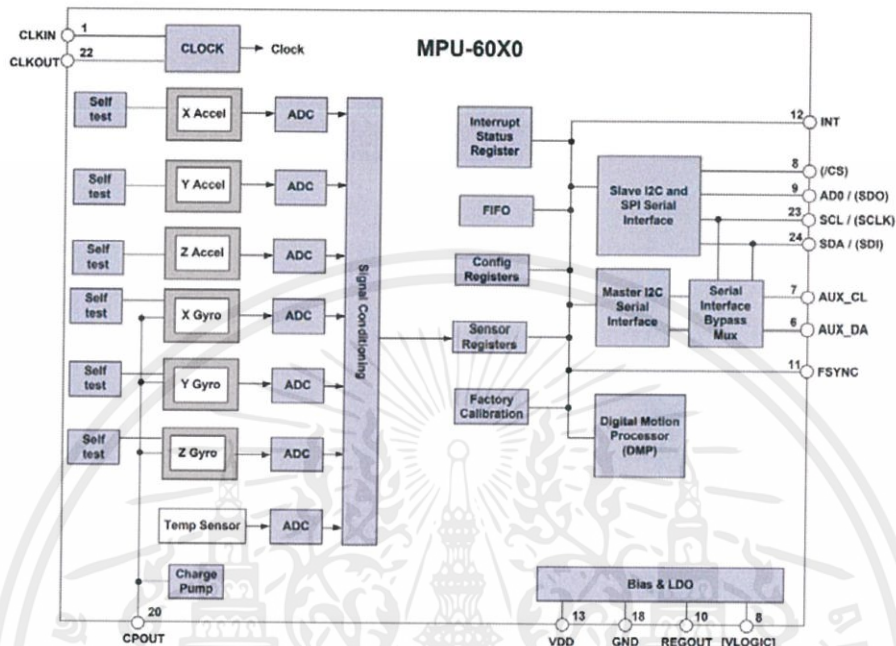
- ควรหลีกเลี่ยงการต่อวงจรให้เกิดการลัดวงจร
- ควรอ่านเอกสารก่อนการต่อวงจรจริง
- ไม่ควรใช้ไฟเกินตามที่เอกสารกำหนด



รูปที่ 2.5 Accelerometers (MPU6050)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.3.4 Block Diagram



รูปที่ 2.6 Block Diagram (MPU6050)

### 2.2.4. ATmega328P



รูปที่ 2.7 ส่วนประกอบของ ATmega328P

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

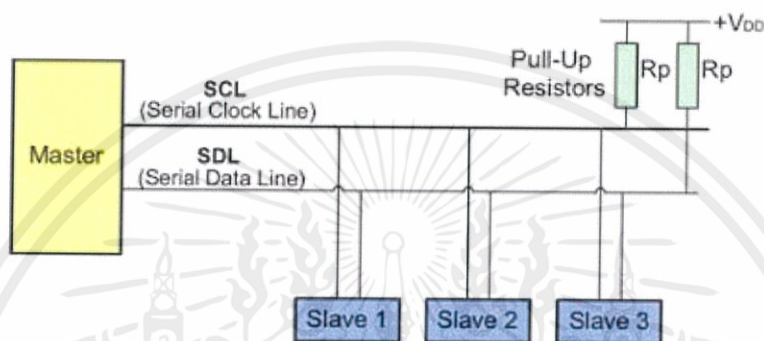
ตารางที่ 2.1 รายละเอียดของ ATmega328P

ชื่อ	รายละเอียด	ขา
GND	ขากราวด์ต่อสาดิน	8,22
VCC	ไฟเลี้ยง 1.8 ถึง 5.5volt	7
Port B(PB 7:0) XTAL 1/XTAL2/TOSC1/ TOSC2	- เป็นพอร์ต 2 ทิศทาง ขนาด 8 บิต โดยสามารถ กำหนดให้ขาของ แต่ละพอร์ตสามารถตั้งค่าให้ Pull up Resistor ได้(ภายในเป็นอิสระ แยกจากกัน เพื่อดึงแรงดันของลอจิก1 ให้เท่ากับ5volt) - สามารถใช้งานพิเศษตามความต้องการของ ATmega328 โดย ขึ้นอยู่กับค่าสัญญาณนาฬิกา ที่ขา PB6 ที่ใช้เป็นแรงดัน Oscillator และขาอินพุตของวงจรถ่ายสัญญาณ Clock Oscillator	9,10, 14-19
PC6/RESET	ขา Reset	1
Port D (PD7:0)	- เป็นพอร์ตสองทิศทางขนาด 8 บิต โดย สามารถ กำหนดให้ขาของ แต่ละพอร์ตสามารถตั้งค่าให้ Pull up Resistor ได้ - สามารถใช้งานพิเศษตามความต้องการของ ATmega328	1-6, 11-13
AVCC	ใช้จ่ายไฟให้กับวงจรถ่ายสัญญาณอนาล็อกเป็นดิจิตอล มักจะต่อกับขา VCC	20
AREF	แรงดันอ้างอิงใช้แปลงสัญญาณอนาล็อกเป็นดิจิตอลมักต่อกับขา VCC	21
ADC7:6 (TQFP and QFN/MLF Package Only)	ขากำลังงานใช้แปลงสัญญาณอนาล็อกเป็นดิจิตอล	23-28
Port (PC5)	- เป็นพอร์ตสองทิศทางขนาด 8 บิต โดยสามารถกำหนดให้ขาของ แต่ละพอร์ตสามารถให้ค่า Pull up Resistor ได้ - สามารถใช้งานพิเศษตามความต้องการของ ATmega328	23-28

## 2.2.5. I2C

### 2.2.5.1 I2C ย่อมาจาก Inter IC Communication คิดค้นโดย Philip

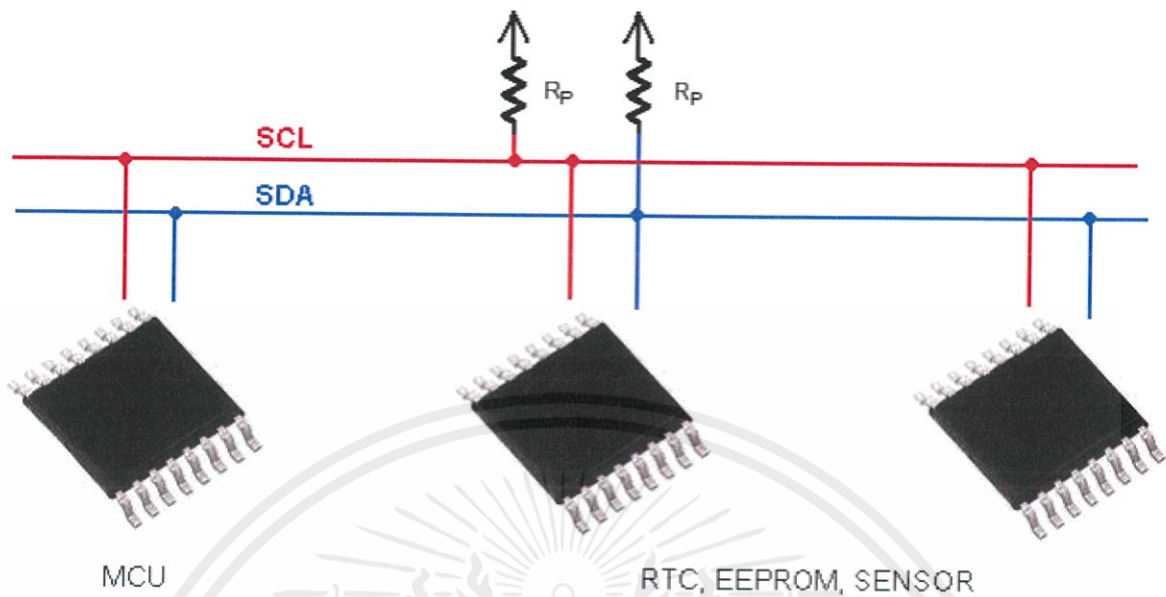
semiconductor หรือ NXP semiconductor จุดประสงค์เพื่อรับส่งข้อมูลความเร็วต่ำระหว่างอุปกรณ์ต่างๆ เช่น เครื่องบันทึกเสียง โทรศัพท์มือถือ รวมถึงหน่วยความจำอย่าง EEPROM เป็นต้น การสื่อสารแบบ I2C มีจุดเด่นคือ การเชื่อมต่อกันเป็นระบบบัสและรับส่งข้อมูลระหว่างอุปกรณ์ได้เป็นจำนวนมาก โดยใช้สายสัญญาณเพียงสองเส้นดังรูป 1-1 ซึ่งสามารถลดจำนวนสายสัญญาณและขนาดของอุปกรณ์ จึงเป็นที่นิยมในระบบสมองกลฝังตัวขนาดเล็ก



รูปที่ 2.8 การเชื่อมต่ออุปกรณ์ แบบ I2C

เป็นการสื่อสารแบบอนุกรมแบบ Synchronous ด้วยสายสัญญาณเพียง 2 เส้น คือ สายสัญญาณข้อมูล SDA (Serial Data Line) และสายสัญญาณนาฬิกา SCL (Serial Clock Line) โดย I2C แบ่งการทำงานออกเป็น 4 โหมดตามความเร็วในการรับส่งข้อมูลดังนี้

1. Normal Mode: 100Kbps
2. Fast Mode: 400Kbps
3. Fast Mode Plus: 1Mbps
4. High Speed Mode: 3.4 Mbps



รูปที่ 2.9 SCL SDA

2.2.5.2. ตัวอย่างการต่อใช้งาน I2C

จากภาพแสดงให้เห็นว่าในการใช้งาน I2C สามารถต่อใช้งานกับอุปกรณ์ได้มากกว่าหนึ่งตัวบนสายสัญญาณ 1 ชุด (2 เส้น) โดยเลือกติดต่อกับอุปกรณ์ใดได้ด้วยการกำหนดแอดเดรสทางฮาร์ดแวร์ให้กับอุปกรณ์ตัวนั้นและต้องมีการ Pull-Up ให้กับสายสัญญาณทั้ง 2 เส้น

ภาวะที่เกิดขึ้นบนบัส I2C



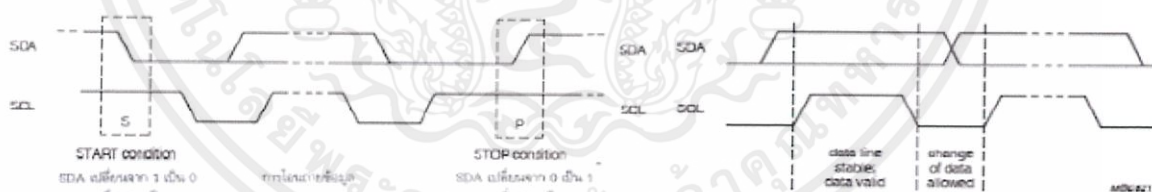
รูปที่ 2.10 ภาวะที่เกิดขึ้นบนบัส I2C

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีด้วยกัน 5 สถานะดังนี้

1. บัสว่าง เป็นสถานะที่มาสเตอร์ทำให้ SDA และ SCL เป็น "1" ทั้งคู่ เพื่อที่จะเริ่มต้นการสื่อสาร
2. สถานะเริ่มต้น Start เกิดขึ้นเมื่อมาสเตอร์ทำให้ SDA เปลี่ยนจาก "1" เป็น "0" ก่อน แล้วจึงเปลี่ยน SCL จาก "1" เป็น "0"
3. สถานะบิตข้อมูล เกิดขึ้นโดยตัวส่งทำให้ SDA เป็นบิตข้อมูล "0" หรือ "1" ก่อน แล้วมาสเตอร์ทำให้เกิดสัญญาณนาฬิกาขึ้นที่ SCL ตัวรับจึงรับข้อมูลในช่วงดังกล่าว
4. สถานะบิตรับรู้ ACK หรือ Acknowledge เกิดขึ้นโดยตัวส่งทำให้ SDA เป็น "1" ก่อนแล้วมาสเตอร์ทำให้เกิดสัญญาณนาฬิกาขึ้นที่ SCL เมื่อตัวรับรับรู้แล้วก็จะทำให้ SDA เป็น "0"
5. สถานะหยุด Stop เกิดขึ้นเมื่อมาสเตอร์ทำให้ SCL เปลี่ยนจาก "0" เป็น "1" ก่อน แล้วจึงเปลี่ยน SDA จาก "0" เป็น "1"

สถานะบนระบบบัส	SDA	SCL
1. Bus not busy	Hi	Hi
2. Start data transfer		Hi
3. Stop		Hi
4. Data valid การรับส่งข้อมูล 1 บิตใช้ clock 1 ลูก ขณะรับ/ส่ง ข้อมูลบน SDA ต้องคงที่และ SCL Hi ข้อมูลบน SDA เปลี่ยนแปลงได้เมื่อ SCL Lo	X Change	Hi Lo
5. Acknowledge ตัวรับจะ ส่งสัญญาณ Ack เมื่อรับข้อมูลครบ 1 byte แล้ว	Lo	Hi



รูปที่ 2.11 สถานะบนระบบบัส

### 2.2.5.3. รูปแบบสถานะในการรับ-ส่งข้อมูล

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

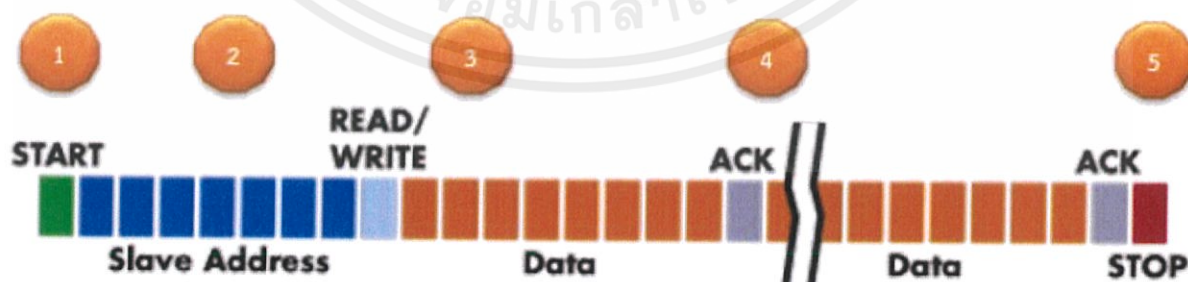
ACK หรือ Acknowledge เป็นบิตที่ใช้บอกว่า IC มีการตอบสนองต่อคำสั่งที่ได้รับมาแล้ว

DATA คือ ข้อมูลที่ต้องการเขียนหรืออ่านออกมาจาก IC ขึ้นกับ Mode ที่เราได้ตั้งค่าเอาไว้

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

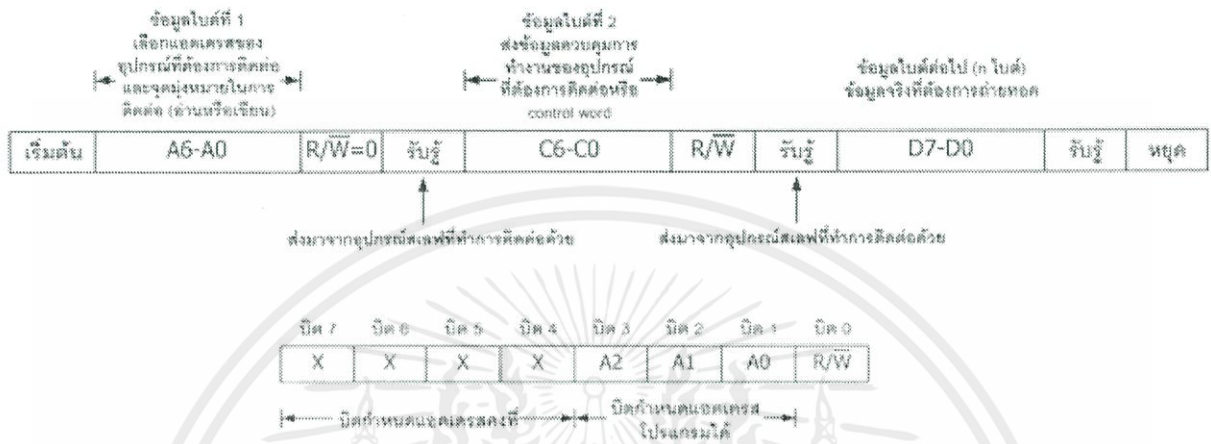
การสื่อสารแบบ I2C มีรูปแบบดังรูป 1-2 และมีขั้นตอนการรับส่งข้อมูลเป็นลำดับ ดังนี้

1. เพื่อเริ่มต้นสถานะการสื่อสารแบบ I2C อุปกรณ์ Master เริ่มจะส่งคำสั่ง START ซึ่งจะให้อุปกรณ์ Slave ทุกตัวที่อยู่ใน Bus เตรียมพร้อมรับข้อมูลจากสายส่งข้อมูล (SDA)
2. อุปกรณ์ Master ส่ง Address ขนาด 7 บิตและตามด้วยคำสั่งอ่านหรือเขียน (Read/Write) อีก 1 บิต
3. อุปกรณ์ Slave ทุกตัวใน Bus จะตรวจสอบ Address ในสายส่งข้อมูลว่าตรงกับ Address ของตนหรือไม่หากตรงกัน จะส่งสัญญาณ ACK ขนาด 1 บิต กลับไปยังอุปกรณ์ Master เพื่อเตรียมพร้อมดำเนินการต่อไป
4. การรับส่งข้อมูลระหว่าง Master และ Slave ที่ระบุ Address จะดำเนินการอย่างต่อเนื่อง ในกรณีที่ Master ส่งคำสั่ง Read อุปกรณ์ Slave จะส่งข้อมูลเป็นชุดๆ ชุดละ 8 บิต (1 ไบต์) เมื่ออุปกรณ์ Master จะส่งสัญญาณ ACK เมื่อได้รับทุกๆ ไบต์
5. อุปกรณ์ Master จะส่งคำสั่ง STOP เพื่อสิ้นสุดสถานะการสื่อสารแบบ I2C



รูปที่ 2.12 รูปแบบการสื่อสารแบบ I2C

2.2.5.4. การเข้าถึงแบบ 7 บิต(7-bit addressing)



รูปที่ 2.13 การเข้าถึงแบบ 7 บิต

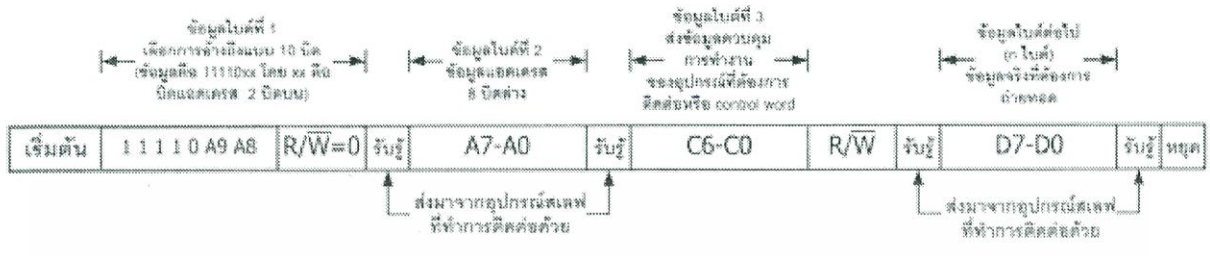
ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสถานะเริ่มต้นคือ ข้อมูลที่ใช้อ้างอุปกรณ์ที่ต้องการติดต่อโดยมีรูปแบบแสดงในรูป

ใน 7 บิตขบวนทั้งบิต LSB ด้วยจะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็นบิตกำหนดแอดเดรสคงที่ (fix address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้ อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิตเป็นบิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (programmable address bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกให้แก่ขา A0-A2 ของอุปกรณ์ที่มีการเชื่อมต่อแบบบัส I2C ส่วนในบิต LSB ที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้น ๆ หากบิต LSB เป็น "0" หมายถึงต้องการเขียนข้อมูลไปยังอุปกรณ์นั้น ถ้าเป็น "1" จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ

ข้อมูลในไบต์ต่อมาคือ ข้อมูลควบคุม(control byte) ในอุปกรณ์แต่ละตัวจะมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไป ยกตัวอย่างเช่น ไอซีเมมโมรี่ของทีวีตระกูล 24Cxx จะต้องส่งข้อมูลแอดเดรสของหน่วยความจำก่อนที่จะทำการส่งข้อมูลไป

ข้อมูลในไบต์ต่อมาคือ ข้อมูลที่ทำการถ่ายทอดจริง(data) หลังจากการถ่ายทอดข้อมูลในแต่ละไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง

2.2.5.5. การเข้าถึงแบบ 10 บิต(10-bit addressing)



รูปที่ 2.14 การเข้าถึงแบบ 10 บิต

จะมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในไบต์แรกหลักจากสถานะเริ่มต้น ต้องกำหนดให้ 5 บิตบนมีข้อมูลเป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อ ในบิต LSB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่า ต้องการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวที่ต้องการติดต่อด้วย ข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อด้วย ข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม ข้อมูลหลังจากนี้ก็จะเป็นข้อมูลจริงที่ใช้ในการติดต่อด้วย เช่นเดียวกัน กับการเข้าถึงแบบ 7 บิตหลังจากถ่ายทอดข้อมูลครบทุกไบต์ ต้องมีสถานะรับรู้เกิดขึ้น เพื่อให้ขบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้

2.2.5.6. ลำดับการทำงาน (Timing Diagram) ก็จะเริ่มต้นดังนี้ครับ

1. ในสถานะที่ยังไม่เริ่มทำงาน อุปกรณ์ทุกตัวจะมีแรงดันระดับสูง (H)
2. เมื่อเริ่มจะส่งข้อมูล SDA จะลดลงไปที่ระดับแรงดันศูนย์ (L) ในขณะที่ SCL ยังอยู่ที่ H จะเห็นในช่องที่แรงเงาสีเหลืองซ้ายสุด โดย SCL จะค้างที่ H นาน 0.6 ไมโครวินาที (tHDSTA: Hold Time to Start Condition) ผ่านช่วงเวลานี้ไปเป็นอันรู้กันว่า Master จะเริ่มส่งการ (แต่ตอนนี้ยังไม่รู้แน่ครึบว่าจะส่ง Slave ตัวไหน เพราะอาจมีหลายตัวต่ออยู่บนบัส)
3. ตอนนี้ SCL จะลงไปที่ L จาก นั้น ระหว่างที่รอสัญญาณขาขึ้นของ Clock แรกอีก 1.3 ไมโครวินาที SDA จะเริ่มกำหนดค่า Bit แรก โดย Master จะส่งค่าบิตแรกไปพร้อมกับสัญญาณ Clock และไอซีที่เป็น Slave บนบัสจะเริ่มอ่านค่าในจังหวะที่ SCL เป็น H
4. จากนั้นก็จะเป็นอย่างนี้ไปอีกเพื่อส่งค่า Address ของไอซีที่ต้องการจะติดต่อด้วยไปรวมทั้งสิ้น 7 บิตและตามด้วยบิตที่ 8 ซึ่งจะระบุว่า จะสั่งให้ Slave รอคำสั่ง (Write ระบุโดย SDA เป็น H) หรือจะรออ่านค่าที่ส่งมาจาก Slave (Read ระบุค่าโดย SDA เป็น L)
5. บิตที่ 9 จะเป็นการตอบรับจาก Slave ที่มี Address ตรงกับที่ Master ส่งไป ถ้ามี Slave ตอบรับ

(Acknowledge) โดยการดึงสัญญาณ SDA ลง L ก็แปลได้ว่ามี Slave พร้อมจะสื่อสารด้วย แต่ถ้าไม่มี คือ SDA ค้างที่ H แปลว่า (Not Acknowledge) Master ไม่มีใครคุยด้วย

6. หลังจาก Acknowledge กันเรียบร้อยแล้ว (สมมุติว่ามีสัญญาณตอบรับ) จะเป็นช่วงเวลาที่ SCL ถูกดึงลง L เป็นเวลาสั้นๆ และ SDA จะถูกปล่อยว่าง ก่อนที่ Slave ที่ติดต่อด้วยจะเริ่มส่งค่า Bit แรกมาที่ SDA

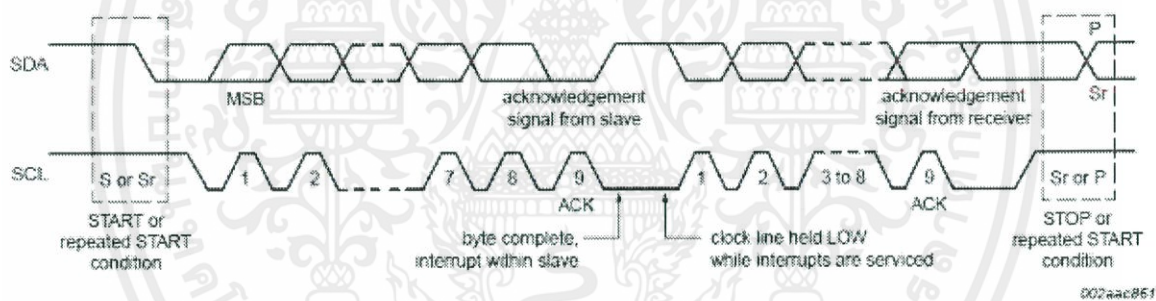
7. จากนั้นสัญญาณ SCL จาก Master จะปล่อยออกมาเพื่ออ่านค่าสัญญาณบนสาย SDA ในจังหวะที่ SCL เป็น H และรับค่าเช่นนี้ไปเรื่อยๆ จน Data ครบ 8 บิตแรก ตามจังหวะสัญญาณ SCL

8. เมื่อ Data ครบ 8 บิตแรกแล้ว Master จะให้สัญญาณ Acknowledge โดยดึงสัญญาณเป็น L เพื่อแสดงให้ Slave รู้ว่า Master จะรอ Data อีก 8 บิตต่อไป (ในกรณีที่ข้อมูลที่ Slave ต้องการส่งเป็นแบบ 16 บิต)

9. จะกลับเข้าการทำงานในข้อ 6 และอ่านข้อมูลไปจนครบบิตที่ 9 - 16

10. จากนั้นเมื่อถึงจังหวะการ Acknowledge ในบิตที่ 9 คราวนี้ Slave ไม่ต้องการส่งข้อมูลต่อแล้ว เพราะครบ 16 บิตแล้ว Slave จะไม่ตอบสนองแล้ว สาย SDA จะถูกปล่อยให้เป็น H

11. จากนั้น Master จะรู้แล้วว่าข้อมูลมาครบถ้วน 16 บิต Master ก็จะสั่งหยุดโดยการส่งสัญญาณ ขาขึ้นในสาย SDA พร้อมกับ SCL เป็น H ก็เป็นอันรู้กันว่าเป็น Stop bit หมายถึงส่งข้อมูลครบ Byte แรก ซึ่งเป็น Address ของ Slave ตัวที่ Master ต้องการคุยด้วย



รูปที่ 2.15 ลำดับการทำงาน (Timing Diagram)

## บทที่ 3

### วิธีการดำเนินงาน

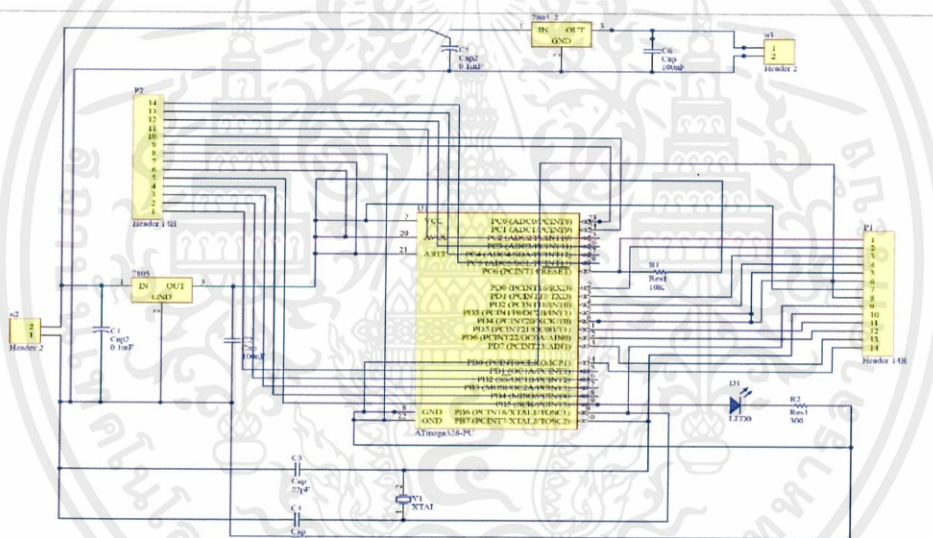
#### 3.1 คุณสมบัติของอุปกรณ์กันการสั่นสะเทือนสำหรับกล่อง

- ทำให้กล่องไม่เอียงไปตามแนวแกนใดๆ โดยอุปกรณ์ที่สร้างขึ้นอาศัยหลักการการทำงานของไมโครคอนโทรลเลอร์

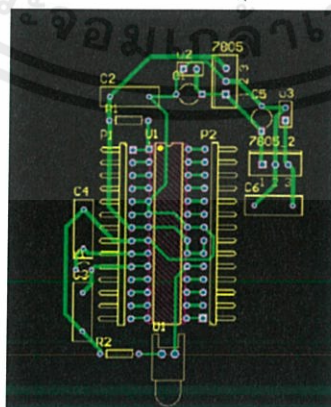
- ควบคุมการทำงานของมอเตอร์ทั้ง 3 ตัวเพื่อให้กล่องอยู่นิ่งโดยใช้ไมโครคอนโทรลเลอร์

#### 3.2 ส่วนประกอบของวงจรไมโครคอนโทรลเลอร์

- Arduino standalone เป็นส่วนของวงจรควบคุม โดยรับสัญญาณอินพุตมาจาก MPU6050 และทำการประมวลผลเพื่อส่งค่าไปยังเซอร์โวมอเตอร์ทั้ง 3 ตัว เพื่อกำหนดองศาในการทำให้กล่องสมดุล



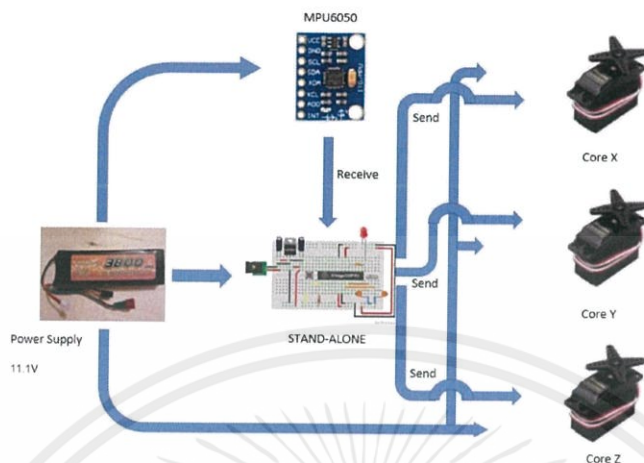
รูปที่ 3.1 วงจรบอร์ดควบคุม (skemetic)



รูปที่ 3.2 วงจรบอร์ดควบคุม (PCB)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 ออกแบบการทำงานของอุปกรณ์กันการสั้นสะเทือนสำหรับกล้องถ่ายรูป

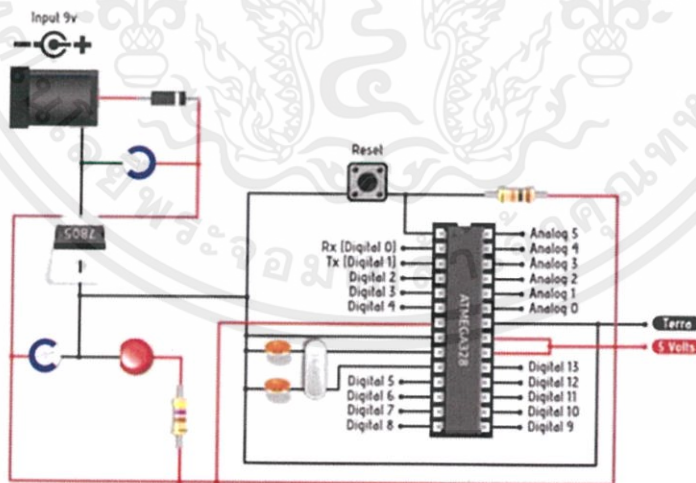


รูปที่ 3.3 บล็อกไดอะแกรมแสดงการทำงาน

จากบล็อกไดอะแกรม แสดงให้เห็นว่าอุปกรณ์กันการสั้นสะเทือนสำหรับกล้องจะควบคุมการทำงาน โดยวงจรควบคุมจากวงจร Microcontroller ผ่าน MPU6050 สั่งให้มอเตอร์ทำงาน

### 3.4 ออกแบบวงจร Microcontroller

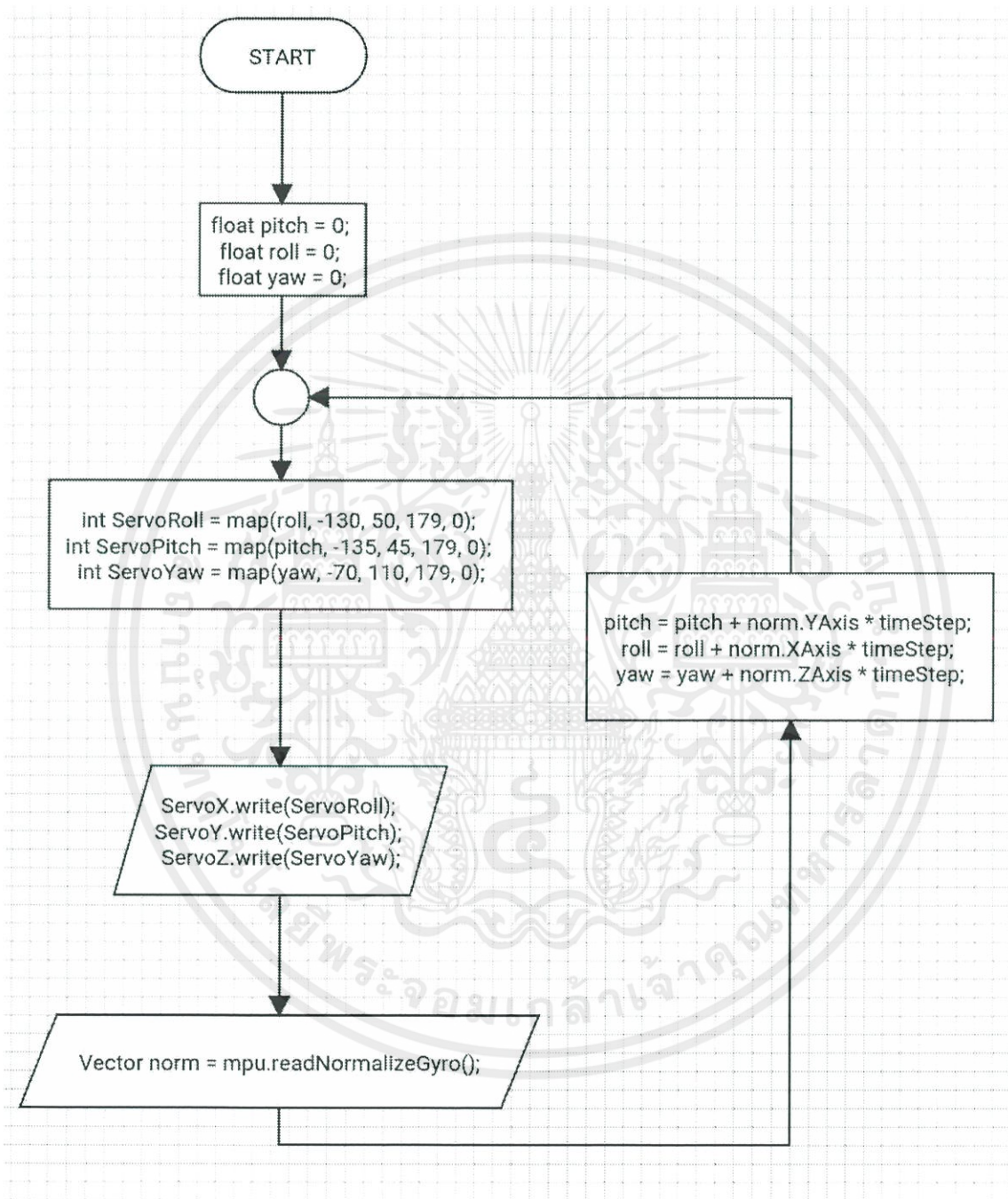
ออกแบบวงจร Microcontroller สำหรับควบคุมการทำงานส่วนต่างๆของอุปกรณ์กันการสั้นสะเทือน สำหรับกล้องถ่ายรูป โดยใช้ ATmega328P



รูปที่ 3.4 วงจร Microcontroller

### 3.5 โปรแกรมควบคุม

#### 3.5.1. Flowchart



รูปที่ 3.5 Flowchart โปรแกรมควบคุม

### 3.5.2. วิธีการคำนวณหาค่ามุม

#### 3.5.2.1. กำหนดตัวแปร

Roll is core X

Pitch is core Y

Yaw is core Z

timeStep=0.01.

#### 3.5.2.2. สูตรคำนวณหาค่ามุม

pitch = pitch + norm.YAxis \* timeStep;

roll = roll + norm.XAxis \* timeStep;

yaw = yaw + norm.ZAxis \* timeStep;

#### 3.5.2.3. โค้ดในการอ่านค่าจากเซนเซอร์

Vector norm = mpu.readNormalizeGyro();

ซึ่งมีค่าระหว่าง 18000 ถึง -18000

#### 3.5.2.4. ยกตัวอย่าง

มุม 45 องศา ที่แกน Y (Pitch)

เริ่มต้นกำหนดค่า Pitch = 0 เมื่อตัวกล้องมีการหมุนเกิดขึ้น ทำให้เซนเซอร์ส่งค่าไปยัง

ไมโครคอนโทรลเลอร์เพื่อคำนวณหาค่ามุม โดยใช้โค้ดดังนี้

pitch = pitch + norm.YAxis \* timeStep;

pitch = 0 + 4500 \* 0.01

pitch = 45 องศา

เมื่อตัวกล้องหมุนกลับ 45 องศา โค้ดเป็นดังนี้

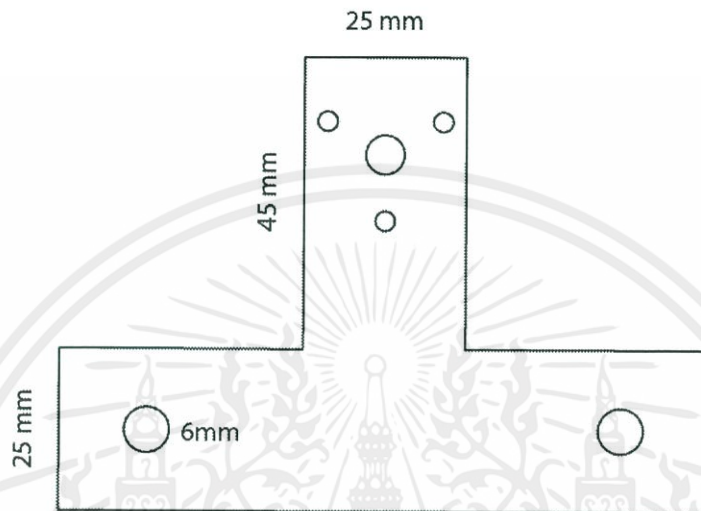
pitch = 45 + (-4500) \* 0.01

pitch = 0 องศา

### 3.6 ส่วนของโครงสร้าง

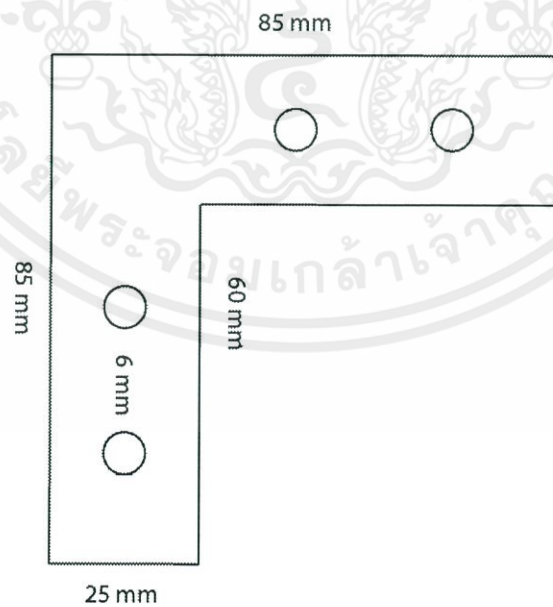
#### 3.6.1. การออกแบบข้อต่อ

3.6.1.1. ข้อต่อ 3 ทาง เพื่อเชื่อมเหล็กโปรไฟล์ในแนวแกน X โดยใช้แผ่นอะคริลิก



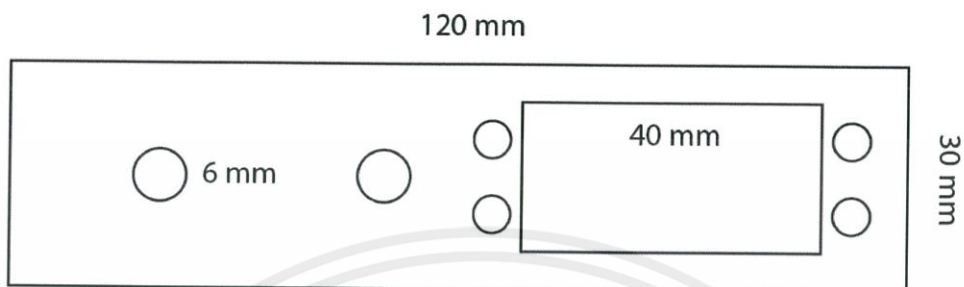
รูปที่ 3.6 ข้อต่อ 3 ทาง (แผ่นอะคริลิก)

3.6.1.2. ข้อต่อ 90 องศา เพื่อเชื่อมเหล็กโปรไฟล์ทำเป็นโครงอุปกรณ์โดยใช้แผ่นอะคริลิก



รูปที่ 3.7 ข้อต่อ 90 องศา (แผ่นอะคริลิก)

3.6.1.3. ข้อต่อเซอร์โว เพื่อเชื่อมเหล็กโปรไฟล์กับเซอร์โวโดยใช้แผ่นอะคริลิก

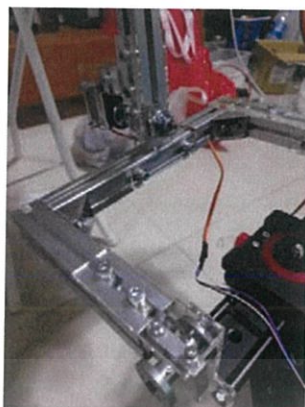


รูปที่ 3.8 ข้อต่อเซอร์โว (แผ่นอะคริลิก)

3.6.2. แขน gimbal stabilizer - ทำมาจากอะลูมิเนียมโปรไฟล์ต่อกับแผ่นอะคริลิก เนื่องจากมีน้ำหนักเบา และแข็งแรง



รูปที่ 3.9 อะลูมิเนียมโปรไฟล์กับแผ่นอะคริลิกก่อนประกอบ



รูปที่ 3.10 แขนของ Gimbal Stabilizer for camera

3.6.3. Servo Moter – เนื่องจากกำหนดองศาได้ชัดเจนจึงเหมาะสำหรับการสร้างอุปกรณ์กันการสั่นสะเทือนสำหรับกล้อง



รูปที่ 3.11 Servo Moter

3.6.4. ประกอบ Gimbal Stabilizer – เมื่อได้ส่วนประกอบการทำงานครบแล้ว จึงประกอบทุกส่วนเข้าด้วยกันและทดสอบการทำงาน



รูปที่ 3.12 อุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

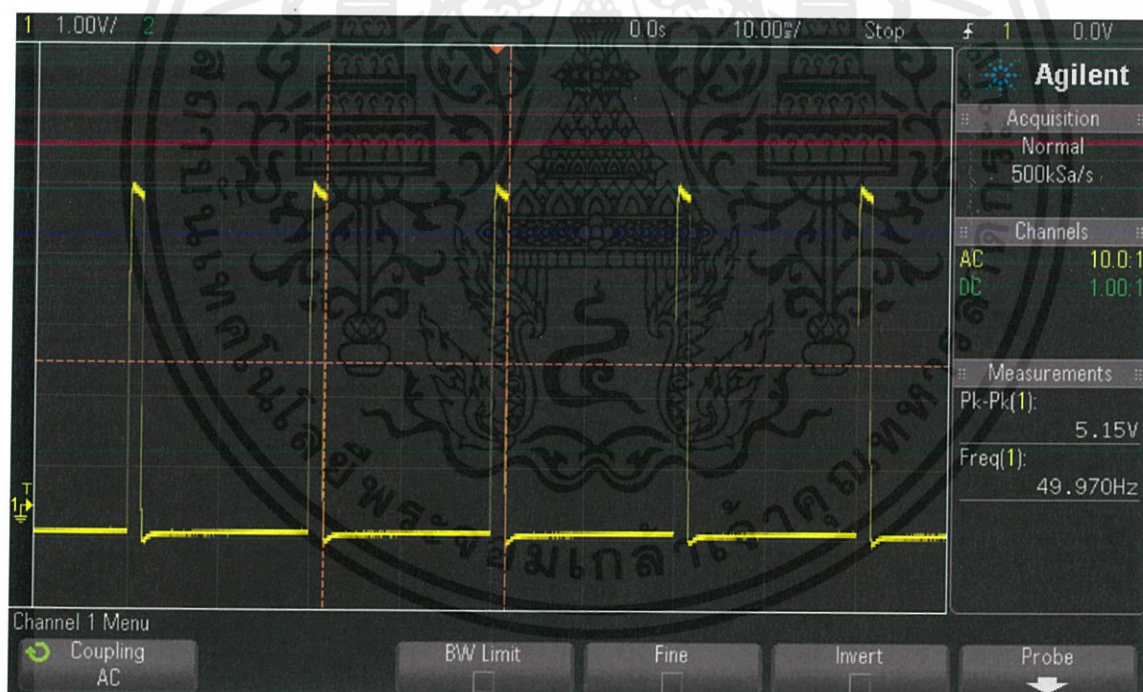
## ผลการทดลอง

## 4.1 อุปกรณ์ที่ใช้ในการทดลอง

- บอร์ด Arduino
- MPU6050
- Servo motor

## 4.2 ผลการทดลอง

1. ทำการทดลองโดยทำการเขียนโค้ดโปรแกรมลงในบอร์ดคอนโทรล Arduino ให้ควบคุม Servo motor หมุน 90 องศา ซึ่งวัดที่ขาเอาต์พุทของบอร์ดคอนโทรล Arduino มี Duty Cycle เท่ากับ 10%



รูปที่ 4.1 สัญญาณ PWM ที่ส่งให้ Servo Motor

2. ทดลอง MPU6050 โดยทำการเอียงเซนเซอร์เป็นมุม 45 องศา และ -45 องศา ในแต่ละแกนได้แก่ แกน X (Roll), แกน Y (Pitch) และแกน Z (Yaw) แล้วทำการเก็บผลการทดลองที่ได้จากค่าของเซนเซอร์ผ่านไลบรารี และค่าที่ได้จากเอาร์ทพุทให้กับเซอร์ไวโมเตอร์

ตารางที่ 4.1 : แกน X

ค่าที่ได้จากเซนเซอร์ผ่านไลบรารี		
45° ROLL		
X	Y	Z
45	15	1
45	15	1
45	15	1
45	15	1
45	15	1
45	15	1
45	15	1
45	15	1
45	15	1
45	15	1
45	15	1
45	15	1
45	15	1
45	15	1
45	15	1

ค่าเอาร์ทพุทที่ส่งให้กับเซอร์ไวโมเตอร์		
45° ROLL		
X	Y	Z
5	30	110
5	30	110
5	30	110
5	30	110
5	30	110
5	30	110
5	30	110
5	30	110
5	30	110
5	30	110
5	30	110
5	30	110
5	30	110
5	30	110
5	30	110
5	30	110

จากตารางพบว่าเมื่อทำการเอียงเซนเซอร์ไปตามแนวแกน X (Roll) เป็นมุม 45 องศา ค่าที่ได้จากเซนเซอร์ แกน X เท่ากับ 45, แกน Y เท่ากับ 15 และ แกน Z เท่ากับ 1

จากตารางพบว่าเมื่อทำการเอียงเซนเซอร์ไปตามแนวแกน X (Roll) เป็นมุม 45 องศา ค่าที่ได้เซอร์ไวโมเตอร์ แกน X เท่ากับ 5, แกน Y เท่ากับ 30 และ แกน Z เท่ากับ 110

ค่าที่ได้จากเซนเซอร์ผ่านไลบรารี		
-45° ROLL		
X	Y	Z
-45	-120	-15
-45	-120	-15
-45	-120	-15
-45	-120	-15
-45	-120	-15
-45	-120	-15
-45	-120	-15
-45	-120	-15
-45	-120	-15
-45	-120	-15
-45	-120	-15
-45	-120	-15
-45	-120	-15
-45	-120	-15
-45	-120	-15

ค่าเอาร์ทพุทที่ส่งให้กับเซอร์ไวโมเตอร์		
-45° ROLL		
X	Y	Z
95	160	125
95	160	125
95	160	125
95	160	125
95	160	125
95	160	125
95	160	125
95	160	125
95	160	125
95	160	125
95	160	125
95	160	125
95	160	125
95	160	125
95	160	125

จากตารางพบว่าเมื่อทำการเอียงเซนเซอร์ไปตามแนวแกน X (Roll) เป็นมุม -45 องศา ค่าที่ได้จากเซนเซอร์ แกน X เท่ากับ -45, แกน Y เท่ากับ -120 และ แกน Z เท่ากับ -15

จากตารางพบว่าเมื่อทำการเอียงเซนเซอร์ไปตามแนวแกน X (Roll) เป็นมุม -45 องศา ค่าที่ได้เซอร์ไวโมเตอร์ แกน X เท่ากับ 95, แกน Y เท่ากับ 160 และ แกน Z เท่ากับ 125

## ตารางที่ 4.2 : แกน Y

ค่าที่ได้จากเซนเซอร์ผ่านโลบรารี		
45° PITCH		
X	Y	Z
-31	44	4
-31	44	4
-31	44	4
-31	44	4
-31	45	4
-31	45	4
-31	45	4
-31	45	4
-31	45	4
-31	45	4
-31	45	4
-31	44	4

จากตารางพบว่าเมื่อทำการเอียงเซนเซอร์ไปตามแนวแกน Y (Pitch) เป็นมุม 45 องศา ค่าที่ได้จากเซนเซอร์ แกน X เท่ากับ -31, แกน Y เท่ากับ 45 และ แกน Z เท่ากับ 4

ค่าที่ได้จากเซนเซอร์ผ่านโลบรารี		
-45° PITCH		
X	Y	Z
70	-45	5
70	-44	5
70	-45	5
70	-45	5
70	-44	5
70	-44	5
70	-44	5
70	-44	5
70	-44	5
70	-45	5
70	-45	5
70	-45	5

จากตารางพบว่าเมื่อทำการเอียงเซนเซอร์ไปตามแนวแกน Y (Pitch) เป็นมุม -45 องศา ค่าที่ได้จากเซนเซอร์ แกน X เท่ากับ 70, แกน Y เท่ากับ -45 และ แกน Z เท่ากับ 5

ค่าเอาต์พุตที่ส่งให้กับเซอร์โวมอเตอร์		
45° PITCH		
X	Y	Z
81	0	106
81	0	106
81	0	106
81	0	106
81	0	106
81	0	106
81	0	106
81	0	106
81	0	106
81	0	106
81	0	106
81	0	106
81	0	106

จากตารางพบว่าเมื่อทำการเอียงเซนเซอร์ไปตามแนวแกน Y (Pitch) เป็นมุม 45 องศา ค่าที่ได้ เซอร์โวมอเตอร์ แกน X เท่ากับ 81, แกน Y เท่ากับ 0 และ แกน Z เท่ากับ 106

ค่าเอาต์พุตที่ส่งให้กับเซอร์โวมอเตอร์		
-45° PITCH		
X	Y	Z
50	90	105
50	90	105
50	90	105
50	90	105
50	90	105
50	90	105
50	90	105
50	90	105
50	90	105
50	90	105
50	90	105
50	90	105
50	90	105

จากตารางพบว่าเมื่อทำการเอียงเซนเซอร์ไปตามแนวแกน Y (Pitch) เป็นมุม -45 องศา ค่าที่ได้ เซอร์โวมอเตอร์ แกน X เท่ากับ 50, แกน Y เท่ากับ 90 และ แกน Z เท่ากับ 105

### ตารางที่ 4.3 : แกน Z

ค่าที่ได้จากเซนเซอร์ผ่านไลบรารี		
45° YAW		
X	Y	Z
5	-12	45
5	-12	45
5	-12	45
5	-12	45
5	-12	45
5	-12	45
5	-12	45
5	-12	45
5	-12	45
5	-12	45
5	-12	45
5	-12	45

จากตารางพบว่าเมื่อทำการเอียงเซนเซอร์ไปตามแนวแกน Z (Yaw) เป็นมุม 45 องศา ค่าที่ได้จากเซนเซอร์ แกน X เท่ากับ 5, แกน Y เท่ากับ -12 และ แกน Z เท่ากับ 45

ค่าเอาต์พุตที่ส่งให้กับเซอร์โวมอเตอร์		
45° YAW		
X	Y	Z
57	46	65
57	46	65
57	46	65
57	46	65
57	46	65
57	46	65
57	46	65
57	46	65
57	46	65
57	46	65
57	46	65
57	46	65

จากตารางพบว่าเมื่อทำการเอียงเซนเซอร์ไปตามแนวแกน Z (Yaw) เป็นมุม 45 องศา ค่าที่ได้เซอร์โวมอเตอร์ แกน X เท่ากับ 57, แกน Y เท่ากับ 46 และ แกน Z เท่ากับ 65

ค่าที่ได้จากเซนเซอร์ผ่านไลบรารี		
-45° YAW		
X	Y	Z
-35	-6	-45
-35	-6	-45
-35	-6	-45
-35	-6	-45
-35	-6	-45
-35	-6	-45
-35	-6	-45
-35	-6	-45
-35	-6	-45
-35	-6	-45
-35	-6	-45
-35	-6	-45

จากตารางพบว่าเมื่อทำการเอียงเซนเซอร์ไปตามแนวแกน Z (Yaw) เป็นมุม -45 องศา ค่าที่ได้จากเซนเซอร์ แกน X เท่ากับ -35, แกน Y เท่ากับ -6 และ แกน Z เท่ากับ -45

ค่าเอาต์พุตที่ส่งให้กับเซอร์โวมอเตอร์		
-45° YAW		
X	Y	Z
51	86	155
51	86	155
51	86	155
51	86	155
51	86	155
51	86	155
51	86	155
51	86	155
51	86	155
51	86	155
51	86	155
51	86	155

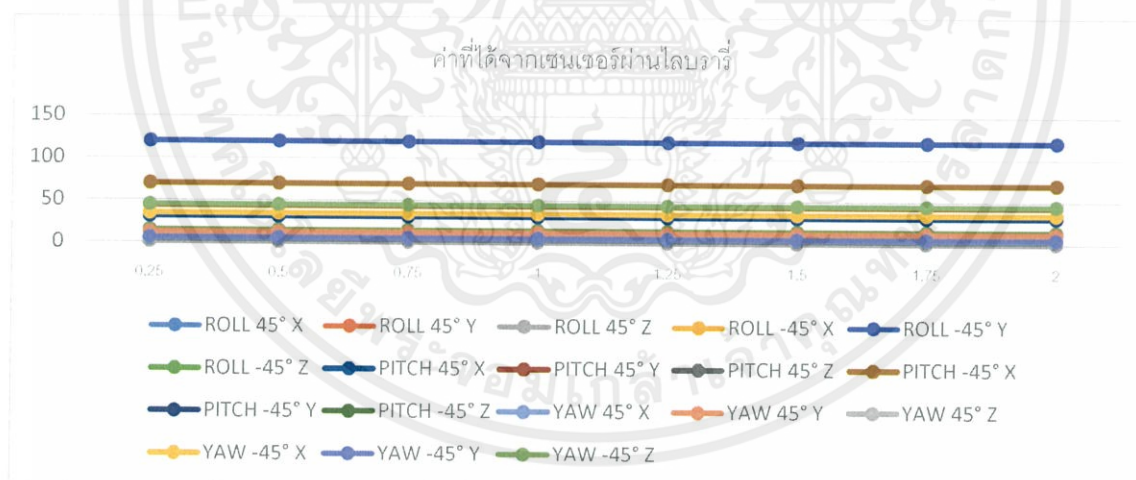
จากตารางพบว่าเมื่อทำการเอียงเซนเซอร์ไปตามแนวแกน Z (Yaw) เป็นมุม -45 องศา ค่าที่ได้เซอร์โวมอเตอร์ แกน X เท่ากับ 51, แกน Y เท่ากับ 86 และ แกน Z เท่ากับ 155

#### ตารางที่ 4.4 : ตารางเทียบมุมของแกน XYZ เทียบเวลา

โดยทำการเอียงเซนเซอร์เป็นมุม 45 องศา และ -45 องศา ในแต่ละแกนได้แก่ แกน X (Roll), แกน Y (Pitch) และแกน Z (Yaw) แล้วทำการเก็บผลการทดลองที่ได้จากค่าของเซนเซอร์ผ่านไลบรารี เทียบกับเวลา (Time)

Time	ค่าที่ได้จากเซนเซอร์ผ่านไลบรารี																	
	ROLL 45°			ROLL -45°			PITCH 45°			PITCH -45°			YAW 45°			YAW -45°		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
0.25	45	15	1	45	120	15	31	44	4	70	45	5	5	12	45	35	6	45
0.5	45	15	1	45	120	15	31	44	4	70	45	5	5	12	45	35	6	45
0.75	45	15	1	45	120	15	31	44	4	70	45	5	5	12	45	35	6	45
1	45	15	1	45	120	15	31	44	4	70	45	5	5	12	45	35	6	45
1.25	45	15	1	45	120	15	31	44	4	70	45	5	5	12	45	35	6	45
1.5	45	15	1	45	120	15	31	44	4	70	45	5	5	12	45	35	6	45
1.75	45	15	1	45	120	15	31	44	4	70	45	5	5	12	45	35	6	45
2	45	15	1	45	120	15	31	44	4	70	45	5	5	12	45	35	6	45

จากตารางพบว่าแกน x, y, และ z แต่ละแกนมีการเปลี่ยนแปลงเล็กน้อยตามเวลาที่เปลี่ยนไป และเมื่อแกนอื่นเปลี่ยนแปลง ทำให้แกนอื่นมีการเปลี่ยนแปลงไปด้วย



รูปที่ 4.2 กราฟค่าที่ได้จากเซนเซอร์ผ่านไลบรารี

## บทที่ 5

### สรุปผลการทดลอง

#### 5.1 สรุปผลการทดลอง

จากการทดลองของอุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูปได้นำ accelerometer sensor มาวัดมุม  $45^{\circ}$  และ  $-45^{\circ}$  ทั้งหมด 3 แกนโดยการอ่านค่าตัวเลขผ่าน Serial Monitor แล้วพบว่าเมื่อวัดค่า แกน X ค่าของแกนอื่นจะมีค่าเปลี่ยนแปลงไปด้วยเช่นเดียวกับแกน Y และ Z ซึ่งตัวเลขที่ไม่คงที่เนื่องจากเซนเซอร์มีความไวสูง จึงสรุปได้ว่าอุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูปสามารถทำให้กล้องเข้าสู่สมดุลและอยู่นิ่งได้ แต่จะมีอาการสั่นเล็กน้อยเพราะเซนเซอร์มีค่าเปลี่ยนแปลงตลอดเวลา

#### 5.2 ปัญหาและอุปสรรค

จากการทดลองของอุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูปพบว่า

- เซอร์โวมอเตอร์ง่ายต่อการใช้งานแต่มีความเร็วไม่เพียงพอสำหรับตัวอุปกรณ์
- อุปกรณ์มีน้ำหนักมากเนื่องจากทำการอะลูมิเนียมแต่ได้ความแข็งแรงสูง

#### 5.3 แนวทางการแก้ไข

- ควรเปลี่ยนมอเตอร์ให้มีความเร็วการหมุนเพิ่มขึ้น
- ควรทำชุดรับน้ำหนักของอุปกรณ์

## เอกสารอ้างอิง

Block Diagram Servo Motor.

ข้อมูลจาก : <https://www.electronicshub.org/servo-motors/>

Arduino.

ข้อมูลจาก : [https://www.thaieasyelec.com/article-wiki/basic-electronics/บทความ-arduino-คืออะไร-เริ่มต้น](https://www.thaieasyelec.com/article-wiki/basic-electronics/บทความ-arduino-คืออะไร-เริ่มต้นใช้งาน-arduino.html)

ใช้งาน-arduino.html

Accelerometers (MPU6050).

ข้อมูลจาก : [https://www.thaieasyelec.com/downloads/ESEN247/GY521\\_USG.pdf](https://www.thaieasyelec.com/downloads/ESEN247/GY521_USG.pdf)

MPU-6050 Datasheet

ข้อมูลจาก : <http://www.alldatasheet.com/datasheet-pdf/pdf/517744/ETC1/MPU-6050.html?fbclid=IwAR1X6NQ0HZwej-NQ1EUdTIVDrWny1BFVjiblKYoe1s-5YLrOoxOYqJj4i1g>

ATMEGA328P Datasheet

ข้อมูลจาก : <http://www.alldatasheet.com/datasheet-pdf/pdf/241077/ATMEL/ATMEGA328P.html?fbclid=IwAR1zFtUG5PPJU0sYJXhxFC5Vp-x3DMyslJ3PiCjNuwxz3hUVDB-j35z1ekg>



## ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งที่ใช้ในการควบคุม

```
#include <MPU6050.h>
```

```
#include <Wire.h>
```

```
#include <Servo.h>
```

```
Servo ServoX, ServoY, ServoZ;
```

```
MPU6050 mpu;
```

```
int16_t ax, ay, az;
```

```
float timeStep = 0.01;
```

```
float pitch = 0;
```

```
float roll = 0;
```

```
float yaw = 0;
```

```
void setup()
```

```
{
```

```
  ServoX.attach(9);
```

```
  ServoY.attach(10);
```

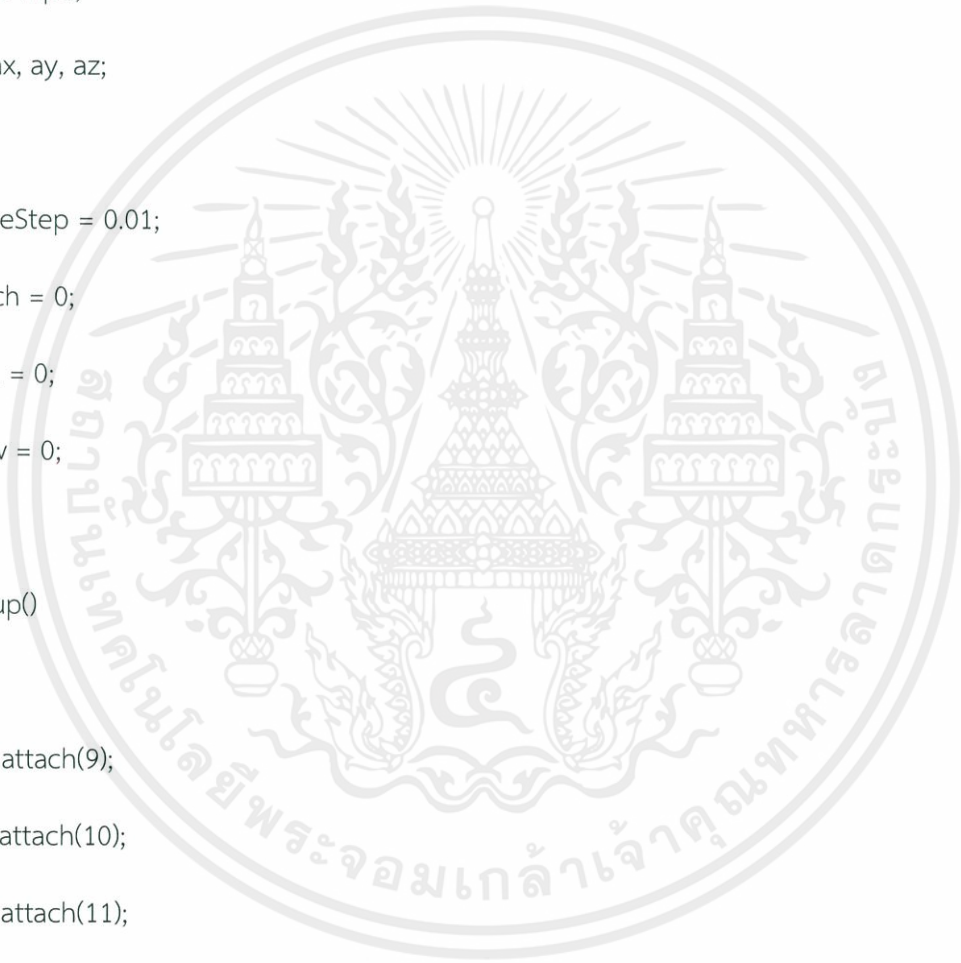
```
  ServoZ.attach(11);
```

```
  ServoX.write(50);
```

```
  ServoY.write(45);
```

```
  ServoZ.write(110);
```

```
  Serial.begin(115200);
```



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))
{
    Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
    delay(5000);
}

mpu.calibrateGyro();
}

void loop()
{
    int ServoRoll = map(roll, -130, 50, 179, 0);
    int ServoPitch = map(pitch, -135, 45, 179, 0);
    int ServoYaw = map(yaw, -111, 69, 0, 179);

    ServoX.write(ServoRoll);
    ServoY.write(ServoPitch);
    ServoZ.write(ServoYaw);

    Vector norm = mpu.readNormalizeGyro();

    if (-2 > norm.YAxis* timeStep >2){
        pitch = pitch ;
        roll = roll ;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    yaw = yaw;
}

else {
    pitch = pitch + norm.YAxis * timeStep;
    roll = roll ;
    yaw = yaw;
}

if (-2 > norm.XAxis* timeStep >2){
    pitch = pitch ;
    roll = roll ;
    yaw = yaw;
}
else {
    pitch = pitch;
    roll = roll + norm.XAxis * timeStep;
    yaw = yaw ;
}

if (-2 > norm.ZAxis* timeStep >2){
    pitch = pitch ;
    roll = roll ;
    yaw = yaw;
```



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

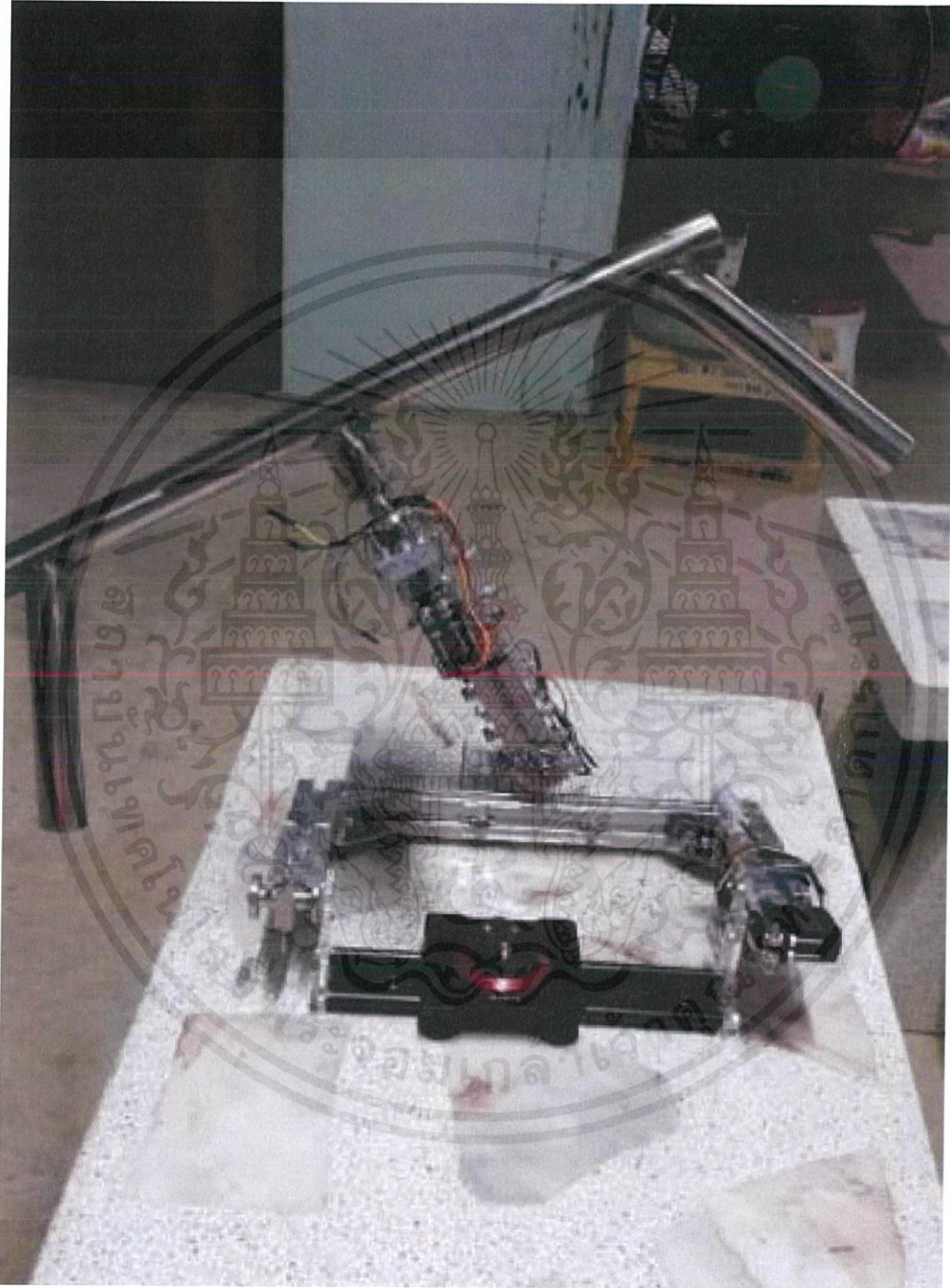
```
}  
  
else {  
  
    pitch = pitch ;  
  
    roll = roll ;  
  
    yaw = yaw + norm.ZAxis * timeStep;  
  
}
```

```
Serial.print(" Pitch = ");  
  
Serial.print(pitch);  
  
Serial.print(" Roll = ");  
  
Serial.print(roll);  
  
Serial.print(" Yaw = ");  
  
Serial.print(yaw);  
  
Serial.print(" ServoPitch = ");  
  
Serial.print(ServoPitch);  
  
Serial.print(" ServoRoll = ");  
  
Serial.print(ServoRoll);  
  
Serial.print(" ServoYaw = ");  
  
Serial.println(ServoYaw);  
  
delay(50);  
  
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# อุปกรณ์กันการสั่นสะเทือนสำหรับกล้องถ่ายรูป

## Gimbal Stabilizer for camera



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**MPU-6000 and MPU-6050**  
**Product Specification**  
**Revision 3.3**

## CONTENTS

<b>1</b>	<b>REVISION HISTORY .....</b>	<b>5</b>
<b>2</b>	<b>PURPOSE AND SCOPE .....</b>	<b>6</b>
<b>3</b>	<b>PRODUCT OVERVIEW .....</b>	<b>7</b>
3.1	MPU-60X0 OVERVIEW .....	7
<b>4</b>	<b>APPLICATIONS.....</b>	<b>9</b>
<b>5</b>	<b>FEATURES .....</b>	<b>10</b>
5.1	GYROSCOPE FEATURES.....	10
5.2	ACCELEROMETER FEATURES .....	10
5.3	ADDITIONAL FEATURES .....	10
5.4	MOTION PROCESSING.....	11
5.5	CLOCKING .....	11
<b>6</b>	<b>ELECTRICAL CHARACTERISTICS .....</b>	<b>12</b>
6.1	GYROSCOPE SPECIFICATIONS.....	12
6.2	ACCELEROMETER SPECIFICATIONS.....	13
6.3	ELECTRICAL AND OTHER COMMON SPECIFICATIONS.....	14
6.4	ELECTRICAL SPECIFICATIONS, CONTINUED .....	15
6.5	ELECTRICAL SPECIFICATIONS, CONTINUED .....	16
6.6	ELECTRICAL SPECIFICATIONS, CONTINUED .....	17
6.7	I <sup>2</sup> C TIMING CHARACTERIZATION.....	18
6.8	SPI TIMING CHARACTERIZATION (MPU-6000 ONLY) .....	19
6.9	ABSOLUTE MAXIMUM RATINGS .....	20
<b>7</b>	<b>APPLICATIONS INFORMATION .....</b>	<b>21</b>
7.1	PIN OUT AND SIGNAL DESCRIPTION .....	21
7.2	TYPICAL OPERATING CIRCUIT.....	22
7.3	BILL OF MATERIALS FOR EXTERNAL COMPONENTS.....	22
7.4	RECOMMENDED POWER-ON PROCEDURE .....	23
7.5	BLOCK DIAGRAM .....	24
7.6	OVERVIEW .....	24
7.7	THREE-AXIS MEMS GYROSCOPE WITH 16-BIT ADCs AND SIGNAL CONDITIONING.....	25
7.8	THREE-AXIS MEMS ACCELEROMETER WITH 16-BIT ADCs AND SIGNAL CONDITIONING .....	25
7.9	DIGITAL MOTION PROCESSOR.....	25
7.10	PRIMARY I <sup>2</sup> C AND SPI SERIAL COMMUNICATIONS INTERFACES .....	25
7.11	AUXILIARY I <sup>2</sup> C SERIAL INTERFACE .....	26

7.12	SELF-TEST.....	27
7.13	MPU-60X0 SOLUTION FOR 9-AXIS SENSOR FUSION USING I <sup>2</sup> C INTERFACE.....	28
7.14	MPU-6000 USING SPI INTERFACE.....	29
7.15	INTERNAL CLOCK GENERATION .....	30
7.16	SENSOR DATA REGISTERS.....	30
7.17	FIFO .....	30
7.18	INTERRUPTS.....	30
7.19	DIGITAL-OUTPUT TEMPERATURE SENSOR .....	31
7.20	BIAS AND LDO .....	31
7.21	CHARGE PUMP .....	31
<b>8</b>	<b>PROGRAMMABLE INTERRUPTS.....</b>	<b>32</b>
8.1	MOTION INTERRUPT.....	33
<b>9</b>	<b>DIGITAL INTERFACE .....</b>	<b>34</b>
9.1	I <sup>2</sup> C AND SPI (MPU-6000 ONLY) SERIAL INTERFACES.....	34
9.2	I <sup>2</sup> C INTERFACE .....	34
9.3	I <sup>2</sup> C COMMUNICATIONS PROTOCOL.....	34
9.4	I <sup>2</sup> C TERMS .....	37
9.5	SPI INTERFACE (MPU-6000 ONLY).....	38
<b>10</b>	<b>SERIAL INTERFACE CONSIDERATIONS (MPU-6050).....</b>	<b>39</b>
10.1	MPU-6050 SUPPORTED INTERFACES.....	39
10.2	LOGIC LEVELS .....	39
10.3	LOGIC LEVELS DIAGRAM FOR AUX_VDDIO = 0.....	40
10.4	LOGIC LEVELS DIAGRAM FOR AUX_VDDIO = 1.....	41
<b>11</b>	<b>ASSEMBLY .....</b>	<b>42</b>
11.1	ORIENTATION OF AXES .....	42
11.2	PACKAGE DIMENSIONS .....	43
11.3	PCB DESIGN GUIDELINES.....	44
11.4	ASSEMBLY PRECAUTIONS .....	45
11.5	STORAGE SPECIFICATIONS.....	48
11.6	PACKAGE MARKING SPECIFICATION.....	48
11.7	TAPE & REEL SPECIFICATION .....	49
11.8	LABEL .....	50
11.9	PACKAGING.....	51
11.10	REPRESENTATIVE SHIPPING CARTON LABEL.....	52

**12 RELIABILITY .....53**  
12.1 QUALIFICATION TEST POLICY .....53  
12.2 QUALIFICATION TEST PLAN .....53  
**13 ENVIRONMENTAL COMPLIANCE.....54**



## 1 Revision History

Revision Date	Revision	Description
11/24/2010	1.0	Initial Release
05/19/2011	2.0	For Rev C parts. Clarified wording in sections (3.2, 5.1, 5.2, 6.1-6.4, 6.6, 6.9, 7, 7.1-7.6, 7.11, 7.12, 7.14, 8, 8.2-8.4, 10.3, 10.4, 11, 12.2)
07/28/2011	2.1	Edited supply current numbers for different modes (section 6.4)
08/05/2011	2.2	Unit of measure for accelerometer sensitivity changed from LSB/mg to LSB/g
10/12/2011	2.3	Updated accelerometer self test specifications in Table 6.2. Updated package dimensions (section 11.2). Updated PCB design guidelines (section 11.3)
10/18/2011	3.0	For Rev D parts. Updated accelerometer specifications in Table 6.2. Updated accelerometer specification note (sections 8.2, 8.3, & 8.4). Updated qualification test plan (section 12.2).
10/24/2011	3.1	Edits for clarity Changed operating voltage range to 2.375V-3.46V Added accelerometer Intelligence Function increment value of 1mg/LSB (Section 6.2) Updated absolute maximum rating for acceleration (any axis, unpowered) from 0.3ms to 0.2ms (Section 6.9) Modified absolute maximum rating for Latch-up to Level A and $\pm 100\text{mA}$ (Section 6.9, 12.2)
11/16/2011	3.2	<b>Updated self-test response specifications for Revision D parts dated with date code 1147 (YYWW) or later.</b> Edits for clarity Added Gyro self-test (sections 5.1, 6.1, 7.6, 7.12) Added Min/Max limits to Accel self-test response (section 6.2) Updated Accelerometer low power mode operating currents (Section 6.3) Added gyro self test to block diagram (section 7.5) Updated packaging labels and descriptions (sections 11.8 & 11.9)
5/16/2012	3.3	Updated Gyro and Accelerometer self test information (sections 6.1, 6.2, 7.12) Updated latch-up information (Section 6.9) Updated programmable interrupts information (Section 8) Changed shipment information from maximum of 3 reels (15K units) per shipper box to 5 reels (25K units) per shipper box (Section 11.7) Updated packing shipping and label information (Sections 11.8, 11.9) Updated reliability references (Section 12.2)

## 2 Purpose and Scope

This product specification provides advanced information regarding the electrical specification and design related information for the MPU-6000™ and MPU-6050™ MotionTracking™ devices, collectively called the MPU-60X0™ or MPU™.

Electrical characteristics are based upon design analysis and simulation results only. Specifications are subject to change without notice. Final specifications will be updated based upon characterization of production silicon. For references to register map and descriptions of individual registers, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document.

**The self-test response specifications provided in this document pertain to Revision D parts with date codes of 1147 (YYWW) or later. Please see Section 11.6 for package marking description details.**



### 3 Product Overview

#### 3.1 MPU-60X0 Overview

MotionInterface™ is becoming a “must-have” function being adopted by smartphone and tablet manufacturers due to the enormous value it adds to the end user experience. In smartphones, it finds use in applications such as gesture commands for applications and phone control, enhanced gaming, augmented reality, panoramic photo capture and viewing, and pedestrian and vehicle navigation. With its ability to precisely and accurately track user motions, MotionTracking technology can convert handsets and tablets into powerful 3D intelligent devices that can be used in applications ranging from health and fitness monitoring to location-based services. Key requirements for MotionInterface enabled devices are small package size, low power consumption, high accuracy and repeatability, high shock tolerance, and application specific performance programmability – all at a low consumer price point.

The MPU-60X0 is the world's first integrated 6-axis MotionTracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor™ (DMP) all in a small 4x4x0.9mm package. With its dedicated I<sup>2</sup>C sensor bus, it directly accepts inputs from an external 3-axis compass to provide a complete 9-axis MotionFusion™ output. The MPU-60X0 MotionTracking device, with its 6-axis integration, on-board MotionFusion™, and run-time calibration firmware, enables manufacturers to eliminate the costly and complex selection, qualification, and system level integration of discrete devices, guaranteeing optimal motion performance for consumers. The MPU-60X0 is also designed to interface with multiple non-inertial digital sensors, such as pressure sensors, on its auxiliary I<sup>2</sup>C port. The MPU-60X0 is footprint compatible with the MPU-30X0 family.

The MPU-60X0 features three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs and three 16-bit ADCs for digitizing the accelerometer outputs. For precision tracking of both fast and slow motions, the parts feature a user-programmable gyroscope full-scale range of  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ , and  $\pm 2000^\circ/\text{sec}$  (dps) and a user-programmable accelerometer full-scale range of  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ , and  $\pm 16g$ .

An on-chip 1024 Byte FIFO buffer helps lower system power consumption by allowing the system processor to read the sensor data in bursts and then enter a low-power mode as the MPU collects more data. With all the necessary on-chip processing and sensor components required to support many motion-based use cases, the MPU-60X0 uniquely enables low-power MotionInterface applications in portable applications with reduced processing requirements for the system processor. By providing an integrated MotionFusion output, the DMP in the MPU-60X0 offloads the intensive MotionProcessing computation requirements from the system processor, minimizing the need for frequent polling of the motion sensor output.

Communication with all registers of the device is performed using either I<sup>2</sup>C at 400kHz or SPI at 1MHz (MPU-6000 only). For applications requiring faster communications, the sensor and interrupt registers may be read using SPI at 20MHz (MPU-6000 only). Additional features include an embedded temperature sensor and an on-chip oscillator with  $\pm 1\%$  variation over the operating temperature range.

By leveraging its patented and volume-proven Nasiri-Fabrication platform, which integrates MEMS wafers with companion CMOS electronics through wafer-level bonding, InvenSense has driven the MPU-60X0 package size down to a revolutionary footprint of 4x4x0.9mm (QFN), while providing the highest performance, lowest noise, and the lowest cost semiconductor packaging required for handheld consumer electronic devices. The part features a robust 10,000g shock tolerance, and has programmable low-pass filters for the gyroscopes, accelerometers, and the on-chip temperature sensor.

For power supply flexibility, the MPU-60X0 operates from VDD power supply voltage range of 2.375V-3.46V. Additionally, the MPU-6050 provides a VLOGIC reference pin (in addition to its analog supply pin: VDD), which sets the logic levels of its I<sup>2</sup>C interface. The VLOGIC voltage may be  $1.8V \pm 5\%$  or VDD.

The MPU-6000 and MPU-6050 are identical, except that the MPU-6050 supports the I<sup>2</sup>C serial interface only, and has a separate VLOGIC reference pin. The MPU-6000 supports both I<sup>2</sup>C and SPI interfaces and has a single supply pin, VDD, which is both the device's logic reference supply and the analog supply for the part. The table below outlines these differences:

**Primary Differences between MPU-6000 and MPU-6050**

Part / Item	MPU-6000	MPU-6050
VDD	2.375V-3.46V	2.375V-3.46V
VLOGIC	n/a	1.71V to VDD
Serial Interfaces Supported	I <sup>2</sup> C, SPI	I <sup>2</sup> C
Pin 8	/CS	VLOGIC
Pin 9	AD0/SDO	AD0
Pin 23	SCL/SCLK	SCL
Pin 24	SDA/SDI	SDA



## 4 Applications

- *BlurFree™* technology (for Video/Still Image Stabilization)
- *AirSign™* technology (for Security/Authentication)
- *TouchAnywhere™* technology (for “no touch” UI Application Control/Navigation)
- *MotionCommand™* technology (for Gesture Short-cuts)
- Motion-enabled game and application framework
- InstantGesture™ iG™ gesture recognition
- Location based services, points of interest, and dead reckoning
- Handset and portable gaming
- Motion-based game controllers
- 3D remote controls for Internet connected DTVs and set top boxes, 3D mice
- Wearable sensors for health, fitness and sports
- Toys



## 5 Features

### 5.1 Gyroscope Features

The triple-axis MEMS gyroscope in the MPU-60X0 includes a wide range of features:

- Digital-output X-, Y-, and Z-Axis angular rate sensors (gyroscopes) with a user-programmable full-scale range of  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ , and  $\pm 2000^\circ/\text{sec}$
- External sync signal connected to the FSYNC pin supports image, video and GPS synchronization
- Integrated 16-bit ADCs enable simultaneous sampling of gyros
- Enhanced bias and sensitivity temperature stability reduces the need for user calibration
- Improved low-frequency noise performance
- Digitally-programmable low-pass filter
- Gyroscope operating current: 3.6mA
- Standby current: 5 $\mu$ A
- Factory calibrated sensitivity scale factor
- User self-test

### 5.2 Accelerometer Features

The triple-axis MEMS accelerometer in MPU-60X0 includes a wide range of features:

- Digital-output triple-axis accelerometer with a programmable full scale range of  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  and  $\pm 16g$
- Integrated 16-bit ADCs enable simultaneous sampling of accelerometers while requiring no external multiplexer
- Accelerometer normal operating current: 500 $\mu$ A
- Low power accelerometer mode current: 10 $\mu$ A at 1.25Hz, 20 $\mu$ A at 5Hz, 60 $\mu$ A at 20Hz, 110 $\mu$ A at 40Hz
- Orientation detection and signaling
- Tap detection
- User-programmable interrupts
- High-G interrupt
- User self-test

### 5.3 Additional Features

The MPU-60X0 includes the following additional features:

- 9-Axis MotionFusion by the on-chip Digital Motion Processor (DMP)
- Auxiliary master I<sup>2</sup>C bus for reading data from external sensors (e.g., magnetometer)
- 3.9mA operating current when all 6 motion sensing axes and the DMP are enabled
- VDD supply voltage range of 2.375V-3.46V
- Flexible VLOGIC reference voltage supports multiple I<sup>2</sup>C interface voltages (MPU-6050 only)
- Smallest and thinnest QFN package for portable devices: 4x4x0.9mm
- Minimal cross-axis sensitivity between the accelerometer and gyroscope axes
- 1024 byte FIFO buffer reduces power consumption by allowing host processor to read the data in bursts and then go into a low-power mode as the MPU collects more data
- Digital-output temperature sensor
- User-programmable digital filters for gyroscope, accelerometer, and temp sensor
- 10,000 g shock tolerant
- 400kHz Fast Mode I<sup>2</sup>C for communicating with all registers
- 1MHz SPI serial interface for communicating with all registers (MPU-6000 only)
- 20MHz SPI serial interface for reading sensor and interrupt registers (MPU-6000 only)

- MEMS structure hermetically sealed and bonded at wafer level
- RoHS and Green compliant

#### 5.4 MotionProcessing

- Internal Digital Motion Processing™ (DMP™) engine supports 3D MotionProcessing and gesture recognition algorithms
- The MPU-60X0 collects gyroscope and accelerometer data while synchronizing data sampling at a user defined rate. The total dataset obtained by the MPU-60X0 includes 3-Axis gyroscope data, 3-Axis accelerometer data, and temperature data. The MPU's calculated output to the system processor can also include heading data from a digital 3-axis third party magnetometer.
- The FIFO buffers the complete data set, reducing timing requirements on the system processor by allowing the processor burst read the FIFO data. After burst reading the FIFO data, the system processor can save power by entering a low-power sleep mode while the MPU collects more data.
- Programmable interrupt supports features such as gesture recognition, panning, zooming, scrolling, zero-motion detection, tap detection, and shake detection
- Digitally-programmable low-pass filters
- Low-power pedometer functionality allows the host processor to sleep while the DMP maintains the step count.

#### 5.5 Clocking

- On-chip timing generator  $\pm 1\%$  frequency variation over full temperature range
- Optional external clock inputs of 32.768kHz or 19.2MHz

## 6 Electrical Characteristics

### 6.1 Gyroscope Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
<b>GYROSCOPE SENSITIVITY</b>						
Full-Scale Range	FS_SEL=0		±250		%/s	
	FS_SEL=1		±500		%/s	
	FS_SEL=2		±1000		%/s	
	FS_SEL=3		±2000		%/s	
Gyroscope ADC Word Length			16		bits	
Sensitivity Scale Factor	FS_SEL=0		131		LSB/(°/s)	
	FS_SEL=1		65.5		LSB/(°/s)	
	FS_SEL=2		32.8		LSB/(°/s)	
	FS_SEL=3		16.4		LSB/(°/s)	
Sensitivity Scale Factor Tolerance	25°C	-3		+3	%	
Sensitivity Scale Factor Variation Over Temperature			±2		%	
Nonlinearity	Best fit straight line; 25°C		0.2		%	
Cross-Axis Sensitivity			±2		%	
<b>GYROSCOPE ZERO-RATE OUTPUT (ZRO)</b>						
Initial ZRO Tolerance	25°C		±20		%/s	
ZRO Variation Over Temperature	-40°C to +85°C		±20		%/s	
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		%/s	
Power-Supply Sensitivity (10 - 250Hz)	Sine wave, 100mVpp; VDD=2.5V		0.2		%/s	
Power-Supply Sensitivity (250Hz - 100kHz)	Sine wave, 100mVpp; VDD=2.5V		4		%/s	
Linear Acceleration Sensitivity	Static		0.1		%/s/g	
<b>SELF-TEST RESPONSE</b>						
Relative	Change from factory trim	-14		14	%	1
<b>GYROSCOPE NOISE PERFORMANCE</b>						
Total RMS Noise	FS_SEL=0 DLPFCFG=2 (100Hz)		0.05		%/s-rms	
Low-frequency RMS noise	Bandwidth 1Hz to 10Hz		0.033		%/s-rms	
Rate Noise Spectral Density	At 10Hz		0.005		%/s/√Hz	
<b>GYROSCOPE MECHANICAL FREQUENCIES</b>						
X-Axis		30	33	36	kHz	
Y-Axis		27	30	33	kHz	
Z-Axis		24	27	30	kHz	
<b>LOW PASS FILTER RESPONSE</b>						
	Programmable Range	5		256	Hz	
<b>OUTPUT DATA RATE</b>						
	Programmable	4		8,000	Hz	
<b>GYROSCOPE START-UP TIME</b>						
ZRO Settling (from power-on)	DLPFCFG=0 to ±1%/s of Final		30		ms	

- Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*

## 6.2 Accelerometer Specifications

VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
<b>ACCELEROMETER SENSITIVITY</b>						
Full-Scale Range	AFS_SEL=0		±2		g	
	AFS_SEL=1		±4		g	
	AFS_SEL=2		±8		g	
	AFS_SEL=3		±16		g	
ADC Word Length	Output in two's complement format		16		bits	
Sensitivity Scale Factor	AFS_SEL=0		16,384		LSB/g	
	AFS_SEL=1		8,192		LSB/g	
	AFS_SEL=2		4,096		LSB/g	
	AFS_SEL=3		2,048		LSB/g	
Initial Calibration Tolerance			±3		%	
Sensitivity Change vs. Temperature	AFS_SEL=0, -40°C to +85°C		±0.02		%/°C	
Nonlinearity	Best Fit Straight Line		0.5		%	
Cross-Axis Sensitivity			±2		%	
<b>ZERO-G OUTPUT</b>						
Initial Calibration Tolerance	X and Y axes		±50		mg	1
	Z axis		±80		mg	
Zero-G Level Change vs. Temperature	X and Y axes, 0°C to +70°C		±35			
	Z axis, 0°C to +70°C		±60		mg	
<b>SELF TEST RESPONSE</b>						
Relative	Change from factory trim	-14		14	%	2
<b>NOISE PERFORMANCE</b>						
Power Spectral Density	@10Hz, AFS_SEL=0 & ODR=1kHz		400		μg/√Hz	
<b>LOW PASS FILTER RESPONSE</b>						
	Programmable Range	5		260	Hz	
<b>OUTPUT DATA RATE</b>						
	Programmable Range	4		1,000	Hz	
<b>INTELLIGENCE FUNCTION INCREMENT</b>			32		mg/LSB	

1. Typical zero-g initial calibration tolerance value after MSL3 preconditioning
2. Please refer to the following document for further information on Self-Test: *MPU-6000/MPU-6050 Register Map and Descriptions*

**6.3 Electrical and Other Common Specifications**

 VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	Units	Notes
<b>TEMPERATURE SENSOR</b>						
Range			-40 to +85		°C	
Sensitivity	Untrimmed		340		LSB/°C	
Temperature Offset	35°C		-521		LSB	
Linearity	Best fit straight line (-40°C to +85°C)		±1		°C	
<b>VDD POWER SUPPLY</b>						
Operating Voltages		2.375		3.46	V	
Normal Operating Current	Gyroscope + Accelerometer + DMP		3.9		mA	
	Gyroscope + Accelerometer (DMP disabled)		3.8		mA	
	Gyroscope + DMP (Accelerometer disabled)		3.7		mA	
	Gyroscope only (DMP & Accelerometer disabled)		3.6		mA	
	Accelerometer only (DMP & Gyroscope disabled)		500		µA	
Accelerometer Low Power Mode Current	1.25 Hz update rate		10		µA	
	5 Hz update rate		20		µA	
	20 Hz update rate		70		µA	
	40 Hz update rate		140		µA	
Full-Chip Idle Mode Supply Current			5		µA	
Power Supply Ramp Rate	Monotonic ramp. Ramp rate is 10% to 90% of the final value			100	ms	
<b>VLOGIC REFERENCE VOLTAGE</b>						
Voltage Range	MPU-6050 only	1.71		VDD	V	
Power Supply Ramp Rate	VLOGIC must be ≤VDD at all times			3	ms	
Normal Operating Current	Monotonic ramp. Ramp rate is 10% to 90% of the final value		100		µA	
<b>TEMPERATURE RANGE</b>						
Specified Temperature Range	Performance parameters are not applicable beyond Specified Temperature Range	-40		+85	°C	

**6.4 Electrical Specifications, Continued**

 VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	Units	Notes
<b>SERIAL INTERFACE</b>						
SPI Operating Frequency, All Registers Read/Write	MPU-6000 only, Low Speed Characterization		100 ±10%		kHz	
	MPU-6000 only, High Speed Characterization		1 ±10%		MHz	
SPI Operating Frequency, Sensor and Interrupt Registers Read Only	MPU-6000 only		20 ±10%		MHz	
	All registers, Fast-mode			400	kHz	
I <sup>2</sup> C Operating Frequency	All registers, Standard-mode			100	kHz	
<b>I<sup>2</sup>C ADDRESS</b>						
	AD0 = 0		1101000			
	AD0 = 1		1101001			
<b>DIGITAL INPUTS (SDI/SDA, AD0, SCLK/SCL, FSYNC, /CS, CLKIN)</b>						
V <sub>IH</sub> , High Level Input Voltage	MPU-6000	0.7*VDD			V	
	MPU-6050	0.7*VLOGIC			V	
V <sub>IL</sub> , Low Level Input Voltage	MPU-6000			0.3*VDD	V	
	MPU-6050			0.3*VLOGIC	V	
C <sub>i</sub> , Input Capacitance			< 5		pF	
<b>DIGITAL OUTPUT (SDO, INT)</b>						
V <sub>OH</sub> , High Level Output Voltage	R <sub>LOAD</sub> =1MΩ; MPU-6000	0.9*VDD			V	
	R <sub>LOAD</sub> =1MΩ; MPU-6050	0.9*VLOGIC			V	
V <sub>OL1</sub> , LOW-Level Output Voltage	R <sub>LOAD</sub> =1MΩ; MPU-6000			0.1*VDD	V	
	R <sub>LOAD</sub> =1MΩ; MPU-6050			0.1*VLOGIC	V	
V <sub>OLINT1</sub> , INT Low-Level Output Voltage	OPEN=1, 0.3mA sink Current			0.1	V	
Output Leakage Current	OPEN=1		100		nA	
t <sub>INT</sub> , INT Pulse Width	LATCH_INT_EN=0		50		μs	
<b>DIGITAL OUTPUT (CLKOUT)</b>						
V <sub>OH</sub> , High Level Output Voltage	R <sub>LOAD</sub> =1MΩ	0.9*VDD			V	
V <sub>OL1</sub> , LOW-Level Output Voltage	R <sub>LOAD</sub> =1MΩ			0.1*VDD	V	

**6.5 Electrical Specifications, Continued**

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, TA = 25°C

Parameters	Conditions	Typical	Units	Notes
<b>Primary I<sup>2</sup>C I/O (SCL, SDA)</b>				
V <sub>IL</sub> , LOW-Level Input Voltage	MPU-6000	-0.5 to 0.3*VDD	V	
V <sub>IH</sub> , HIGH-Level Input Voltage	MPU-6000	0.7*VDD to VDD + 0.5V	V	
V <sub>hys</sub> , Hysteresis	MPU-6000	0.1*VDD	V	
V <sub>IL</sub> , LOW Level Input Voltage	MPU-6050	-0.5V to 0.3*VLOGIC	V	
V <sub>IH</sub> , HIGH-Level Input Voltage	MPU-6050	0.7*VLOGIC to VLOGIC + 0.5V	V	
V <sub>hys</sub> , Hysteresis	MPU-6050	0.1*VLOGIC	V	
V <sub>OL1</sub> , LOW-Level Output Voltage	3mA sink current	0 to 0.4	V	
I <sub>OL</sub> , LOW-Level Output Current	V <sub>OL</sub> = 0.4V	3	mA	
	V <sub>OL</sub> = 0.6V	5	mA	
Output Leakage Current		100	nA	
t <sub>of</sub> , Output Fall Time from V <sub>IHmax</sub> to V <sub>ILmax</sub>	C <sub>b</sub> bus capacitance in pF	20+0.1C <sub>b</sub> to 250	ns	
C <sub>i</sub> , Capacitance for Each I/O pin		< 10	pF	
<b>Auxiliary I<sup>2</sup>C I/O (AUX_CL, AUX_DA)</b>				
<b>MPU-6050: AUX_VDDIO=0</b>				
V <sub>IL</sub> , LOW-Level Input Voltage		-0.5V to 0.3*VLOGIC	V	
V <sub>IH</sub> , HIGH-Level Input Voltage		0.7*VLOGIC to VLOGIC + 0.5V	V	
V <sub>hys</sub> , Hysteresis		0.1*VLOGIC	V	
V <sub>OL1</sub> , LOW-Level Output Voltage	VLOGIC > 2V; 1mA sink current	0 to 0.4	V	
V <sub>OL3</sub> , LOW-Level Output Voltage	VLOGIC < 2V; 1mA sink current	0 to 0.2*VLOGIC	V	
I <sub>OL</sub> , LOW-Level Output Current	V <sub>OL</sub> = 0.4V	1	mA	
	V <sub>OL</sub> = 0.6V	1	mA	
Output Leakage Current		100	nA	
t <sub>of</sub> , Output Fall Time from V <sub>IHmax</sub> to V <sub>ILmax</sub>	C <sub>b</sub> bus capacitance in pF	20+0.1C <sub>b</sub> to 250	ns	
C <sub>i</sub> , Capacitance for Each I/O pin		< 10	pF	
<b>Auxiliary I<sup>2</sup>C I/O (AUX_CL, AUX_DA)</b>				
<b>MPU-6050: AUX_VDDIO=1; MPU-6000</b>				
V <sub>IL</sub> , LOW-Level Input Voltage		-0.5 to 0.3*VDD	V	
V <sub>IH</sub> , HIGH-Level Input Voltage		0.7*VDD to VDD+0.5V	V	
V <sub>hys</sub> , Hysteresis		0.1*VDD	V	
V <sub>OL1</sub> , LOW-Level Output Voltage	1mA sink current	0 to 0.4	V	
I <sub>OL</sub> , LOW-Level Output Current	V <sub>OL</sub> = 0.4V	1	mA	
	V <sub>OL</sub> = 0.6V	1	mA	
Output Leakage Current		100	nA	
t <sub>of</sub> , Output Fall Time from V <sub>IHmax</sub> to V <sub>ILmax</sub>	C <sub>b</sub> bus cap. in pF	20+0.1C <sub>b</sub> to 250	ns	
C <sub>i</sub> , Capacitance for Each I/O pin		< 10	pF	

**6.6 Electrical Specifications, Continued**

 Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

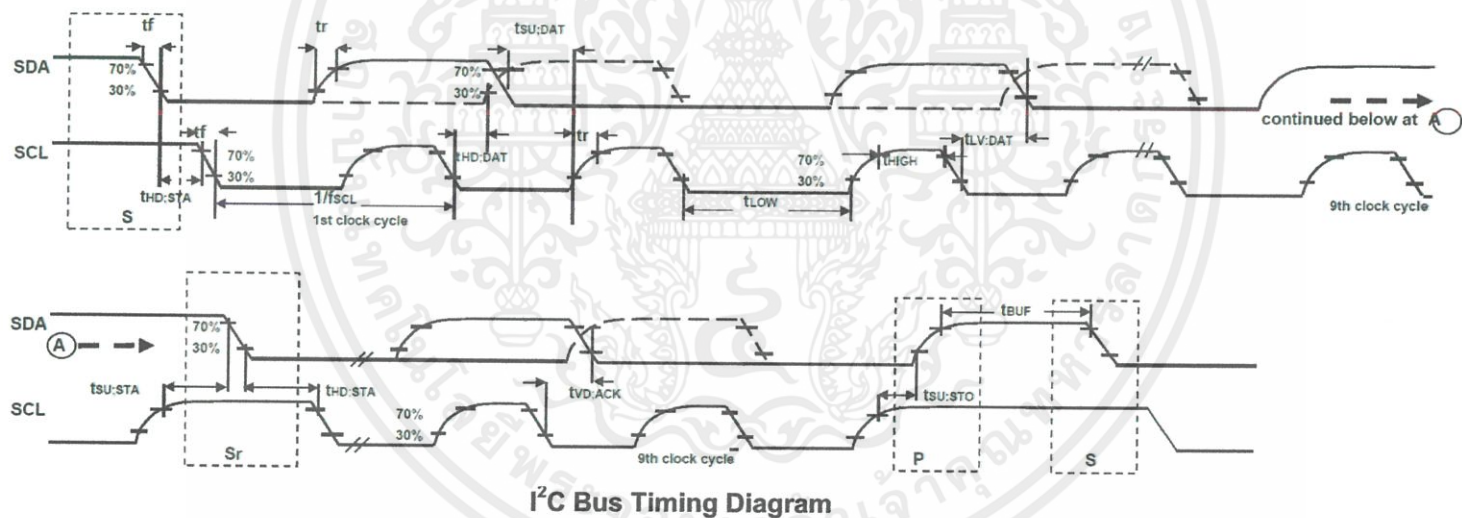
Parameters	Conditions	Min	Typical	Max	Units	Notes
<b>INTERNAL CLOCK SOURCE</b>						
Gyroscope Sample Rate, Fast	CLK_SEL=0,1,2,3 DLPFCFG=0 SAMPLERATEDIV = 0		8		kHz	
Gyroscope Sample Rate, Slow	DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0		1		kHz	
Accelerometer Sample Rate			1		kHz	
Reference Clock Output	CLKOUTEN = 1		1.024		MHz	
Clock Frequency Initial Tolerance	CLK_SEL=0, 25°C	-5		+5	%	
	CLK_SEL=1,2,3; 25°C	-1		+1	%	
Frequency Variation over Temperature	CLK_SEL=0		-15 to +10		%	
	CLK_SEL=1,2,3		±1		%	
PLL Settling Time	CLK_SEL=1,2,3		1	10	ms	
<b>EXTERNAL 32.768kHz CLOCK</b>						
External Clock Frequency	CLK_SEL=4		32.768		kHz	
External Clock Allowable Jitter	Cycle-to-cycle rms		1 to 2		µs	
Gyroscope Sample Rate, Fast	DLPFCFG=0 SAMPLERATEDIV = 0		8.192		kHz	
Gyroscope Sample Rate, Slow	DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0		1.024		kHz	
Accelerometer Sample Rate			1.024		kHz	
Reference Clock Output	CLKOUTEN = 1		1.0486		MHz	
PLL Settling Time			1	10	ms	
<b>EXTERNAL 19.2MHz CLOCK</b>						
External Clock Frequency	CLK_SEL=5		19.2		MHz	
Gyroscope Sample Rate	Full programmable range	3.9		8000	Hz	
Gyroscope Sample Rate, Fast Mode	DLPFCFG=0 SAMPLERATEDIV = 0		8		kHz	
Gyroscope Sample Rate, Slow Mode	DLPFCFG=1,2,3,4,5, or 6 SAMPLERATEDIV = 0		1		kHz	
Accelerometer Sample Rate			1		kHz	
Reference Clock Output	CLKOUTEN = 1		1.024		MHz	
PLL Settling Time			1	10	ms	

### 6.7 I<sup>2</sup>C Timing Characterization

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C

Parameters	Conditions	Min	Typical	Max	Units	Notes
<b>I<sup>2</sup>C TIMING</b>						
<b>I<sup>2</sup>C FAST-MODE</b>						
f <sub>SCL</sub> , SCL Clock Frequency				400	kHz	
t <sub>HD,STA</sub> , (Repeated) START Condition Hold Time		0.6			μs	
t <sub>LOW</sub> , SCL Low Period		1.3			μs	
t <sub>HIGH</sub> , SCL High Period		0.6			μs	
t <sub>SU,STA</sub> , Repeated START Condition Setup Time		0.6			μs	
t <sub>HD,DAT</sub> , SDA Data Hold Time		0			μs	
t <sub>SU,DAT</sub> , SDA Data Setup Time		100			ns	
t <sub>r</sub> , SDA and SCL Rise Time	C <sub>b</sub> bus cap. from 10 to 400pF	20+0.1C <sub>b</sub>		300	ns	
t <sub>f</sub> , SDA and SCL Fall Time	C <sub>b</sub> bus cap. from 10 to 400pF	20+0.1C <sub>b</sub>		300	ns	
t <sub>SU,STO</sub> , STOP Condition Setup Time		0.6			μs	
t <sub>BUF</sub> , Bus Free Time Between STOP and START Condition		1.3			μs	
C <sub>b</sub> , Capacitive Load for each Bus Line			< 400		pF	
t <sub>VD,DAT</sub> , Data Valid Time				0.9	μs	
t <sub>VD,ACK</sub> , Data Valid Acknowledge Time				0.9	μs	

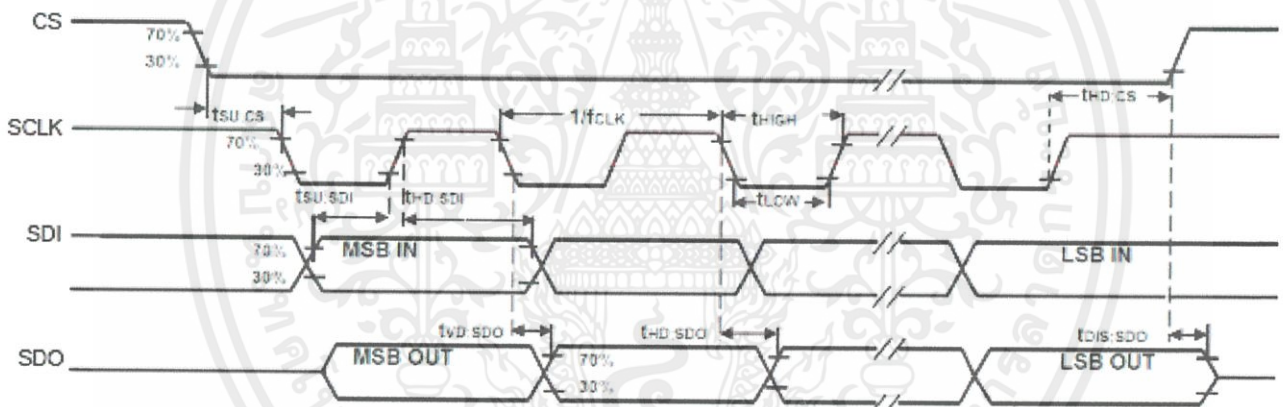
**Note:** Timing Characteristics apply to both Primary and Auxiliary I<sup>2</sup>C Bus



**6.8 SPI Timing Characterization (MPU-6000 only)**

 Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T<sub>A</sub> = 25°C, unless otherwise noted.

Parameters	Conditions	Min	Typical	Max	Units	Notes
<b>SPI TIMING</b>						
f <sub>SCLK</sub> , SCLK Clock Frequency				1	MHz	
t <sub>LOW</sub> , SCLK Low Period		400			ns	
t <sub>HIGH</sub> , SCLK High Period		400			ns	
t <sub>SU,CS</sub> , CS Setup Time		8			ns	
t <sub>HD,CS</sub> , CS Hold Time		500			ns	
t <sub>SU,SDI</sub> , SDI Setup Time		11			ns	
t <sub>HD,SDI</sub> , SDI Hold Time		7			ns	
t <sub>VD,SDO</sub> , SDO Valid Time	C <sub>load</sub> = 20pF			100	ns	
t <sub>HD,SDO</sub> , SDO Hold Time	C <sub>load</sub> = 20pF	4			ns	
t <sub>DIS,SDO</sub> , SDO Output Disable Time				10	ns	


**SPI Bus Timing Diagram**

### 6.9 Absolute Maximum Ratings

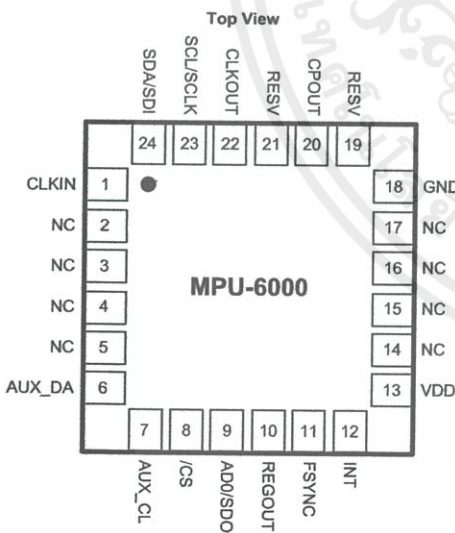
Stress above those listed as “Absolute Maximum Ratings” may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these conditions is not implied. Exposure to the absolute maximum ratings conditions for extended periods may affect device reliability.

Parameter	Rating
Supply Voltage, VDD	-0.5V to +6V
VLOGIC Input Voltage Level (MPU-6050)	-0.5V to VDD + 0.5V
REGOUT	-0.5V to 2V
Input Voltage Level (CLKIN, AUX_DA, AD0, FSYNC, INT, SCL, SDA)	-0.5V to VDD + 0.5V
CPOUT (2.5V ≤ VDD ≤ 3.6V )	-0.5V to 30V
Acceleration (Any Axis, unpowered)	10,000g for 0.2ms
Operating Temperature Range	-40°C to +105°C
Storage Temperature Range	-40°C to +125°C
Electrostatic Discharge (ESD) Protection	2kV (HBM); 200V (MM)
Latch-up	JEDEC Class II (2), 125°C ±100mA

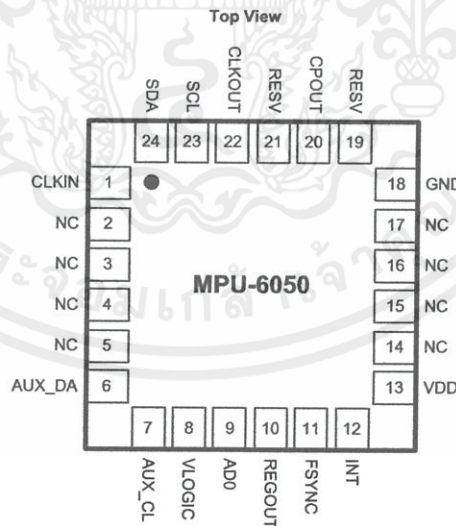
## 7 Applications Information

### 7.1 Pin Out and Signal Description

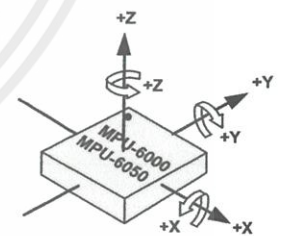
Pin Number	MPU-6000	MPU-6050	Pin Name	Pin Description
1	Y	Y	CLKIN	Optional external reference clock input. Connect to GND if unused.
6	Y	Y	AUX_DA	I <sup>2</sup> C master serial data, for connecting to external sensors
7	Y	Y	AUX_CL	I <sup>2</sup> C Master serial clock, for connecting to external sensors
8	Y		/CS	SPI chip select (0=SPI mode)
8		Y	VLOGIC	Digital I/O supply voltage
9	Y		AD0 / SDO	I <sup>2</sup> C Slave Address LSB (AD0); SPI serial data output (SDO)
9		Y	AD0	I <sup>2</sup> C Slave Address LSB (AD0)
10	Y	Y	REGOUT	Regulator filter capacitor connection
11	Y	Y	FSYNC	Frame synchronization digital input. Connect to GND if unused.
12	Y	Y	INT	Interrupt digital output (totem pole or open-drain)
13	Y	Y	VDD	Power supply voltage and Digital I/O supply voltage
18	Y	Y	GND	Power supply ground
19, 21	Y	Y	RESV	Reserved. Do not connect.
20	Y	Y	CPOUT	Charge pump capacitor connection
22	Y	Y	CLKOUT	System clock output
23	Y		SCL / SCLK	I <sup>2</sup> C serial clock (SCL); SPI serial clock (SCLK)
23		Y	SCL	I <sup>2</sup> C serial clock (SCL)
24	Y		SDA / SDI	I <sup>2</sup> C serial data (SDA); SPI serial data input (SDI)
24		Y	SDA	I <sup>2</sup> C serial data (SDA)
2, 3, 4, 5, 14, 15, 16, 17	Y	Y	NC	Not internally connected. May be used for PCB trace routing.



QFN Package  
24-pin, 4mm x 4mm x 0.9mm

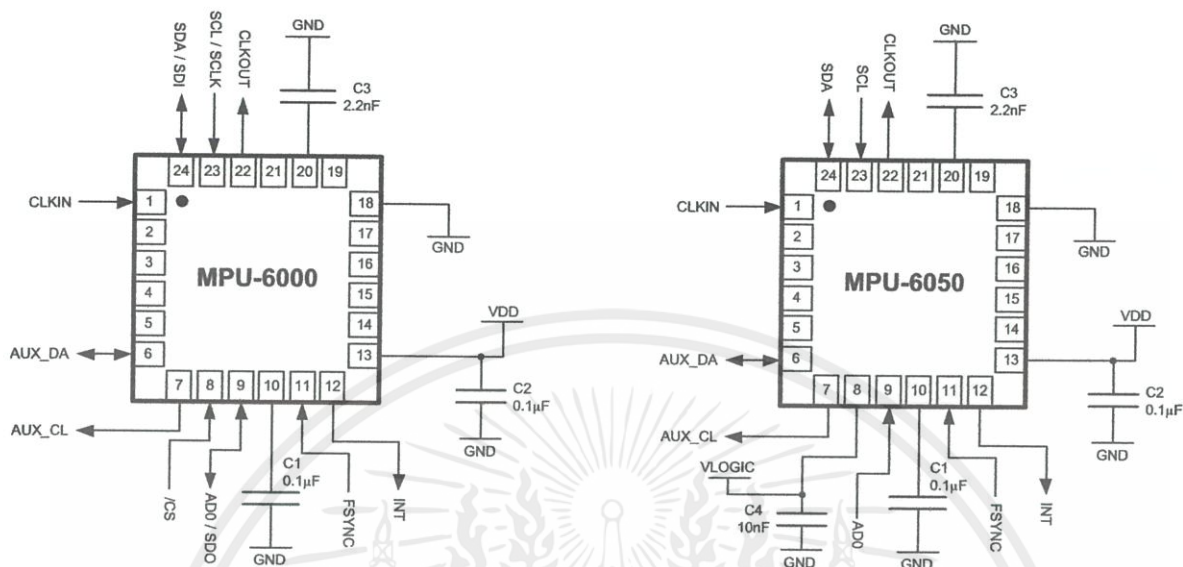


QFN Package  
24-pin, 4mm x 4mm x 0.9mm



Orientation of Axes of Sensitivity and  
Polarity of Rotation

## 7.2 Typical Operating Circuit

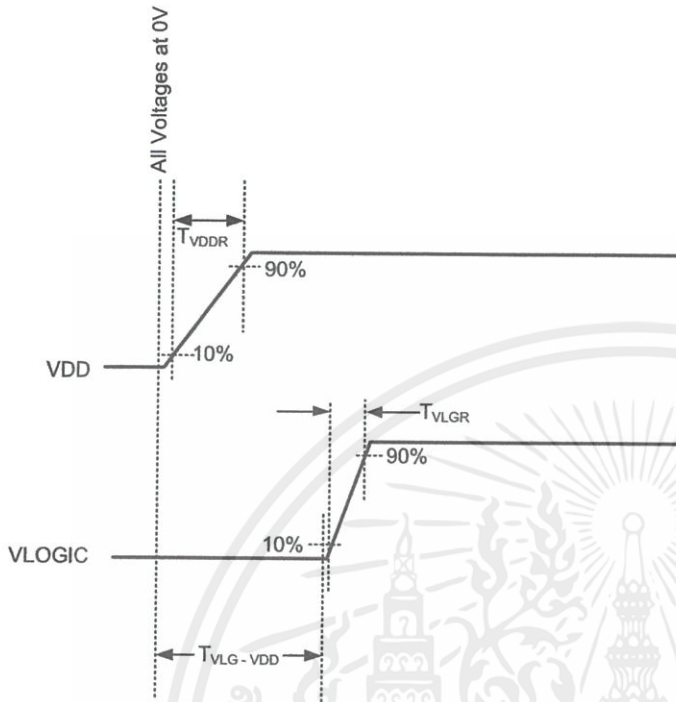


Typical Operating Circuits

## 7.3 Bill of Materials for External Components

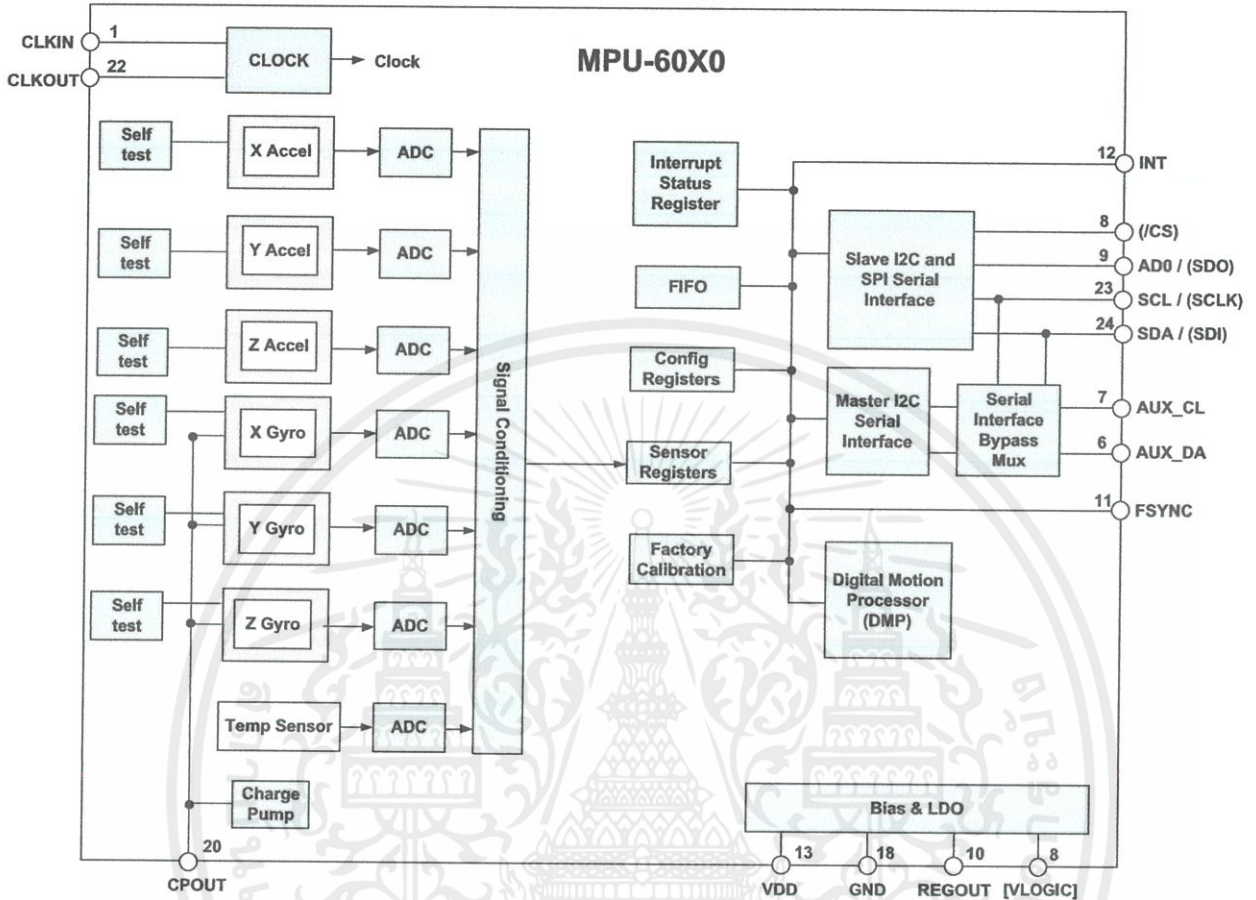
Component	Label	Specification	Quantity
Regulator Filter Capacitor (Pin 10)	C1	Ceramic, X7R, 0.1µF ±10%, 2V	1
VDD Bypass Capacitor (Pin 13)	C2	Ceramic, X7R, 0.1µF ±10%, 4V	1
Charge Pump Capacitor (Pin 20)	C3	Ceramic, X7R, 2.2nF ±10%, 50V	1
VLOGIC Bypass Capacitor (Pin 8)	C4*	Ceramic, X7R, 10nF ±10%, 4V	1

\* MPU-6050 Only.

**7.4 Recommended Power-on Procedure**

**Power-Up Sequencing**

1. VLOGIC amplitude must always be  $\leq$ VDD amplitude
2.  $T_{VDDR}$  is VDD rise time: Time for VDD to rise from 10% to 90% of its final value
3.  $T_{VDDR}$  is  $\leq 100\text{ms}$
4.  $T_{VLGR}$  is VLOGIC rise time: Time for VLOGIC to rise from 10% to 90% of its final value
5.  $T_{VLGR}$  is  $\leq 3\text{ms}$
6.  $T_{VLG-VDD}$  is the delay from the start of VDD ramp to the start of VLOGIC rise
7.  $T_{VLG-VDD}$  is  $\geq 0$
8. VDD and VLOGIC must be monotonic ramps

### 7.5 Block Diagram



Note: Pin names in round brackets ( ) apply only to MPU-6000  
Pin names in square brackets [ ] apply only to MPU-6050

### 7.6 Overview

The MPU-60X0 is comprised of the following key blocks and functions:

- Three-axis MEMS rate gyroscope sensor with 16-bit ADCs and signal conditioning
- Three-axis MEMS accelerometer sensor with 16-bit ADCs and signal conditioning
- Digital Motion Processor (DMP) engine
- Primary I<sup>2</sup>C and SPI (MPU-6000 only) serial communications interfaces
- Auxiliary I<sup>2</sup>C serial interface for 3<sup>rd</sup> party magnetometer & other sensors
- Clocking
- Sensor Data Registers
- FIFO
- Interrupts
- Digital-Output Temperature Sensor
- Gyroscope & Accelerometer Self-test
- Bias and LDO
- Charge Pump

### 7.7 Three-Axis MEMS Gyroscope with 16-bit ADCs and Signal Conditioning

The MPU-60X0 consists of three independent vibratory MEMS rate gyroscopes, which detect rotation about the X-, Y-, and Z- Axes. When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a capacitive pickoff. The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate. This voltage is digitized using individual on-chip 16-bit Analog-to-Digital Converters (ADCs) to sample each axis. The full-scale range of the gyro sensors may be digitally programmed to  $\pm 250$ ,  $\pm 500$ ,  $\pm 1000$ , or  $\pm 2000$  degrees per second (dps). The ADC sample rate is programmable from 8,000 samples per second, down to 3.9 samples per second, and user-selectable low-pass filters enable a wide range of cut-off frequencies.

### 7.8 Three-Axis MEMS Accelerometer with 16-bit ADCs and Signal Conditioning

The MPU-60X0's 3-Axis accelerometer uses separate proof masses for each axis. Acceleration along a particular axis induces displacement on the corresponding proof mass, and capacitive sensors detect the displacement differentially. The MPU-60X0's architecture reduces the accelerometers' susceptibility to fabrication variations as well as to thermal drift. When the device is placed on a flat surface, it will measure  $0g$  on the X- and Y-axes and  $+1g$  on the Z-axis. The accelerometers' scale factor is calibrated at the factory and is nominally independent of supply voltage. Each sensor has a dedicated sigma-delta ADC for providing digital outputs. The full scale range of the digital output can be adjusted to  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$ , or  $\pm 16g$ .

### 7.9 Digital Motion Processor

The embedded Digital Motion Processor (DMP) is located within the MPU-60X0 and offloads computation of motion processing algorithms from the host processor. The DMP acquires data from accelerometers, gyroscopes, and additional 3<sup>rd</sup> party sensors such as magnetometers, and processes the data. The resulting data can be read from the DMP's registers, or can be buffered in a FIFO. The DMP has access to one of the MPU's external pins, which can be used for generating interrupts.

The purpose of the DMP is to offload both timing requirements and processing power from the host processor. Typically, motion processing algorithms should be run at a high rate, often around 200Hz, in order to provide accurate results with low latency. This is required even if the application updates at a much lower rate; for example, a low power user interface may update as slowly as 5Hz, but the motion processing should still run at 200Hz. The DMP can be used as a tool in order to minimize power, simplify timing, simplify the software architecture, and save valuable MIPS on the host processor for use in the application.

### 7.10 Primary I<sup>2</sup>C and SPI Serial Communications Interfaces

The MPU-60X0 communicates to a system processor using either a SPI (MPU-6000 only) or an I<sup>2</sup>C serial interface. The MPU-60X0 always acts as a slave when communicating to the system processor. The LSB of the of the I<sup>2</sup>C slave address is set by pin 9 (AD0).

The logic levels for communications between the MPU-60X0 and its master are as follows:

- **MPU-6000:** The logic level for communications with the master is set by the voltage on VDD
- **MPU-6050:** The logic level for communications with the master is set by the voltage on VLOGIC

For further information regarding the logic levels of the MPU-6050, please refer to Section 10.

### 7.11 Auxiliary I<sup>2</sup>C Serial Interface

The MPU-60X0 has an auxiliary I<sup>2</sup>C bus for communicating to an off-chip 3-Axis digital output magnetometer or other sensors. This bus has two operating modes:

- **I<sup>2</sup>C Master Mode:** The MPU-60X0 acts as a master to any external sensors connected to the auxiliary I<sup>2</sup>C bus
- **Pass-Through Mode:** The MPU-60X0 directly connects the primary and auxiliary I<sup>2</sup>C buses together, allowing the system processor to directly communicate with any external sensors.

#### Auxiliary I<sup>2</sup>C Bus Modes of Operation:

- **I<sup>2</sup>C Master Mode:** Allows the MPU-60X0 to directly access the data registers of external digital sensors, such as a magnetometer. In this mode, the MPU-60X0 directly obtains data from auxiliary sensors, allowing the on-chip DMP to generate sensor fusion data without intervention from the system applications processor.

For example, In I<sup>2</sup>C Master mode, the MPU-60X0 can be configured to perform burst reads, returning the following data from a magnetometer:

- X magnetometer data (2 bytes)
- Y magnetometer data (2 bytes)
- Z magnetometer data (2 bytes)

The I<sup>2</sup>C Master can be configured to read up to 24 bytes from up to 4 auxiliary sensors. A fifth sensor can be configured to work single byte read/write mode.

- **Pass-Through Mode:** Allows an external system processor to act as master and directly communicate to the external sensors connected to the auxiliary I<sup>2</sup>C bus pins (AUX\_DA and AUX\_CL). In this mode, the auxiliary I<sup>2</sup>C bus control logic (3<sup>rd</sup> party sensor interface block) of the MPU-60X0 is disabled, and the auxiliary I<sup>2</sup>C pins AUX\_DA and AUX\_CL (Pins 6 and 7) are connected to the main I<sup>2</sup>C bus (Pins 23 and 24) through analog switches.

Pass-Through Mode is useful for configuring the external sensors, or for keeping the MPU-60X0 in a low-power mode when only the external sensors are used.

In Pass-Through Mode the system processor can still access MPU-60X0 data through the I<sup>2</sup>C interface.

#### Auxiliary I<sup>2</sup>C Bus IO Logic Levels

- **MPU-6000:** The logic level of the auxiliary I<sup>2</sup>C bus is VDD
- **MPU-6050:** The logic level of the auxiliary I<sup>2</sup>C bus can be programmed to be either VDD or VLOGIC

For further information regarding the MPU-6050's logic levels, please refer to Section 10.2.

## 7.12 Self-Test

Please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document for more details on self test.

Self-test allows for the testing of the mechanical and electrical portions of the sensors. The self-test for each measurement axis can be activated by means of the gyroscope and accelerometer self-test registers (registers 13 to 16).

When self-test is activated, the electronics cause the sensors to be actuated and produce an output signal. The output signal is used to observe the self-test response.

The self-test response is defined as follows:

Self-test response = Sensor output with self-test enabled – Sensor output without self-test enabled

The self-test response for each accelerometer axis is defined in the accelerometer specification table (Section 6.2), while that for each gyroscope axis is defined in the gyroscope specification table (Section 6.1).

When the value of the self-test response is within the min/max limits of the product specification, the part has passed self test. When the self-test response exceeds the min/max values, the part is deemed to have failed self-test. Code for operating self test code is included within the MotionApps software provided by InvenSense.

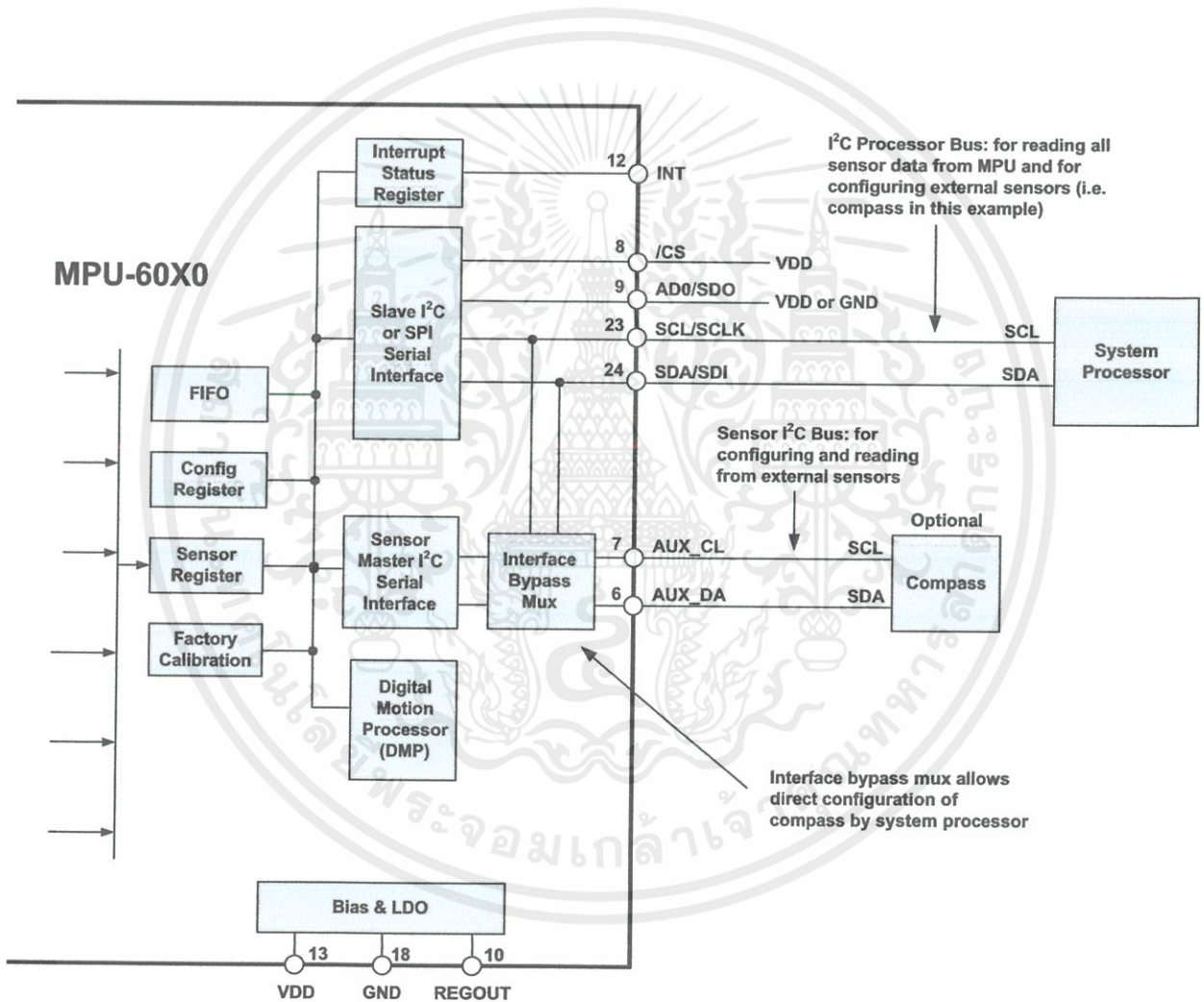


### 7.13 MPU-60X0 Solution for 9-axis Sensor Fusion Using I<sup>2</sup>C Interface

In the figure below, the system processor is an I<sup>2</sup>C master to the MPU-60X0. In addition, the MPU-60X0 is an I<sup>2</sup>C master to the optional external compass sensor. The MPU-60X0 has limited capabilities as an I<sup>2</sup>C Master, and depends on the system processor to manage the initial configuration of any auxiliary sensors. The MPU-60X0 has an interface bypass multiplexer, which connects the system processor I<sup>2</sup>C bus pins 23 and 24 (SDA and SCL) directly to the auxiliary sensor I<sup>2</sup>C bus pins 6 and 7 (AUX\_DA and AUX\_CL).

Once the auxiliary sensors have been configured by the system processor, the interface bypass multiplexer should be disabled so that the MPU-60X0 auxiliary I<sup>2</sup>C master can take control of the sensor I<sup>2</sup>C bus and gather data from the auxiliary sensors.

For further information regarding I<sup>2</sup>C master control, please refer to Section 10.



### 7.14 MPU-6000 Using SPI Interface

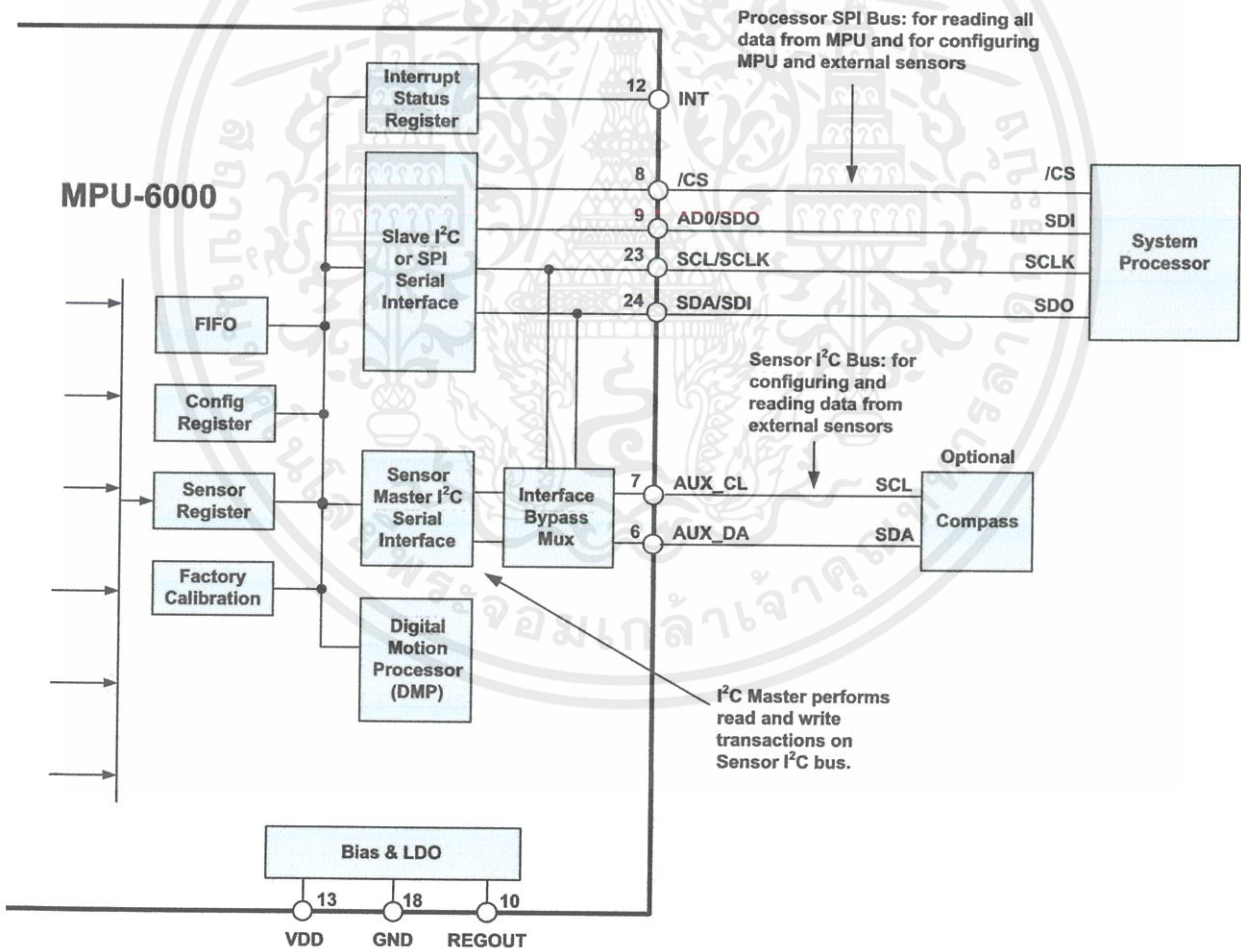
In the figure below, the system processor is an SPI master to the MPU-6000. Pins 8, 9, 23, and 24 are used to support the /CS, SDO, SCLK, and SDI signals for SPI communications. Because these SPI pins are shared with the I<sup>2</sup>C slave pins (9, 23 and 24), the system processor cannot access the auxiliary I<sup>2</sup>C bus through the interface bypass multiplexer, which connects the processor I<sup>2</sup>C interface pins to the sensor I<sup>2</sup>C interface pins.

Since the MPU-6000 has limited capabilities as an I<sup>2</sup>C Master, and depends on the system processor to manage the initial configuration of any auxiliary sensors, another method must be used for programming the sensors on the auxiliary sensor I<sup>2</sup>C bus pins 6 and 7 (AUX\_DA and AUX\_CL).

When using SPI communications between the MPU-6000 and the system processor, configuration of devices on the auxiliary I<sup>2</sup>C sensor bus can be achieved by using I<sup>2</sup>C Slaves 0-4 to perform read and write transactions on any device and register on the auxiliary I<sup>2</sup>C bus. The I<sup>2</sup>C Slave 4 interface can be used to perform only single byte read and write transactions.

Once the external sensors have been configured, the MPU-6000 can perform single or multi-byte reads using the sensor I<sup>2</sup>C bus. The read results from the Slave 0-3 controllers can be written to the FIFO buffer as well as to the external sensor registers.

For further information regarding the control of the MPU-60X0's auxiliary I<sup>2</sup>C interface, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document.



### 7.15 Internal Clock Generation

The MPU-60X0 has a flexible clocking scheme, allowing a variety of internal or external clock sources to be used for the internal synchronous circuitry. This synchronous circuitry includes the signal conditioning and ADCs, the DMP, and various control circuits and registers. An on-chip PLL provides flexibility in the allowable inputs for generating this clock.

Allowable internal sources for generating the internal clock are:

- An internal relaxation oscillator
- Any of the X, Y, or Z gyros (MEMS oscillators with a variation of  $\pm 1\%$  over temperature)

Allowable external clocking sources are:

- 32.768kHz square wave
- 19.2MHz square wave

Selection of the source for generating the internal synchronous clock depends on the availability of external sources and the requirements for power consumption and clock accuracy. These requirements will most likely vary by mode of operation. For example, in one mode, where the biggest concern is power consumption, the user may wish to operate the Digital Motion Processor of the MPU-60X0 to process accelerometer data, while keeping the gyros off. In this case, the internal relaxation oscillator is a good clock choice. However, in another mode, where the gyros are active, selecting the gyros as the clock source provides for a more accurate clock source.

Clock accuracy is important, since timing errors directly affect the distance and angle calculations performed by the Digital Motion Processor (and by extension, by any processor).

There are also start-up conditions to consider. When the MPU-60X0 first starts up, the device uses its internal clock until programmed to operate from another source. This allows the user, for example, to wait for the MEMS oscillators to stabilize before they are selected as the clock source.

### 7.16 Sensor Data Registers

The sensor data registers contain the latest gyro, accelerometer, auxiliary sensor, and temperature measurement data. They are read-only registers, and are accessed via the serial interface. Data from these registers may be read anytime. However, the interrupt function may be used to determine when new data is available.

For a table of interrupt sources please refer to Section 8.

### 7.17 FIFO

The MPU-60X0 contains a 1024-byte FIFO register that is accessible via the Serial Interface. The FIFO configuration register determines which data is written into the FIFO. Possible choices include gyro data, accelerometer data, temperature readings, auxiliary sensor readings, and FSYNC input. A FIFO counter keeps track of how many bytes of valid data are contained in the FIFO. The FIFO register supports burst reads. The interrupt function may be used to determine when new data is available.

For further information regarding the FIFO, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document.

### 7.18 Interrupts

Interrupt functionality is configured via the Interrupt Configuration register. Items that are configurable include the INT pin configuration, the interrupt latching and clearing method, and triggers for the interrupt. Items that can trigger an interrupt are (1) Clock generator locked to new reference oscillator (used when switching clock

sources); (2) new data is available to be read (from the FIFO and Data registers); (3) accelerometer event interrupts; and (4) the MPU-60X0 did not receive an acknowledge from an auxiliary sensor on the secondary I<sup>2</sup>C bus. The interrupt status can be read from the Interrupt Status register.

For further information regarding interrupts, please refer to the MPU-60X0 Register Map and Register Descriptions document.

For information regarding the MPU-60X0's accelerometer event interrupts, please refer to Section 8.

### 7.19 Digital-Output Temperature Sensor

An on-chip temperature sensor and ADC are used to measure the MPU-60X0 die temperature. The readings from the ADC can be read from the FIFO or the Sensor Data registers.

### 7.20 Bias and LDO

The bias and LDO section generates the internal supply and the reference voltages and currents required by the MPU-60X0. Its two inputs are an unregulated VDD of 2.375 to 3.46V and a VLOGIC logic reference supply voltage of 1.71V to VDD (MPU-6050 only). The LDO output is bypassed by a capacitor at REGOUT. For further details on the capacitor, please refer to the Bill of Materials for External Components (Section 7.3).

### 7.21 Charge Pump

An on-board charge pump generates the high voltage required for the MEMS oscillators. Its output is bypassed by a capacitor at CPOUT. For further details on the capacitor, please refer to the Bill of Materials for External Components (Section 7.3).

## 8 Programmable Interrupts

The MPU-60X0 has a programmable interrupt system which can generate an interrupt signal on the INT pin. Status flags indicate the source of an interrupt. Interrupt sources may be enabled and disabled individually.

### Table of Interrupt Sources

Interrupt Name	Module
Motion Detection	Motion
FIFO Overflow	FIFO
Data Ready	Sensor Registers
I <sup>2</sup> C Master errors: Lost Arbitration, NACKs	I <sup>2</sup> C Master
I <sup>2</sup> C Slave 4	I <sup>2</sup> C Master

For information regarding the interrupt enable/disable registers and flag registers, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document. Some interrupt sources are explained below.

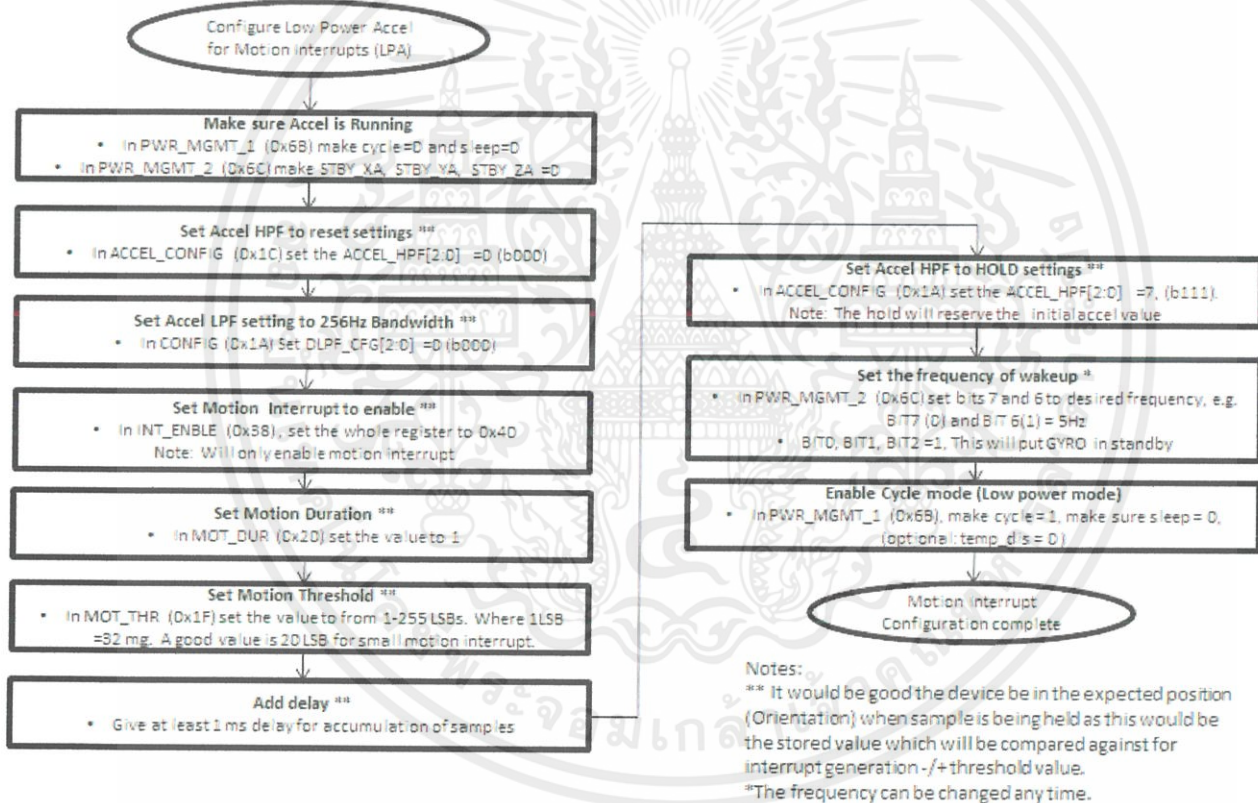


### 8.1 Motion Interrupt

The MPU-60X0 provides Motion detection capability. Accelerometer measurements are passed through a configurable digital high pass filter (DHPF) in order to eliminate bias due to gravity. A qualifying motion sample is one where the high passed sample from any axis has an absolute value exceeding a user-programmable threshold. A counter increments for each qualifying sample, and decrements for each non-qualifying sample. Once the counter reaches a user-programmable counter threshold, a motion interrupt is triggered. The axis and polarity which caused the interrupt to be triggered is flagged in the MOT\_DETECT\_STATUS register.

Motion detection has a configurable acceleration threshold MOT\_THR specified in 1 mg increments. The counter threshold MOT\_DUR is specified in 1 ms increments. The decrement rate for non-qualifying samples is also configurable. The MOT\_DETECT\_CTRL register allows the user to specify whether a non-qualifying sample makes the counter reset to zero, or decrement in steps of 1, 2, or 4.

The flow chart below explains how the motion interrupt should be used. Please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document for descriptions of the registers referenced in the flow chart.



## 9 Digital Interface

### 9.1 I<sup>2</sup>C and SPI (MPU-6000 only) Serial Interfaces

The internal registers and memory of the MPU-6000/MPU-6050 can be accessed using either I<sup>2</sup>C at 400 kHz or SPI at 1MHz (MPU-6000 only). SPI operates in four-wire mode.

#### Serial Interface

Pin Number	MPU-6000	MPU-6050	Pin Name	Pin Description
8	Y		/CS	SPI chip select (0=SPI enable)
8		Y	VLOGIC	Digital I/O supply voltage. VLOGIC must be ≤ VDD at all times.
9	Y		AD0 / SDO	I <sup>2</sup> C Slave Address LSB (AD0); SPI serial data output (SDO)
9		Y	AD0	I <sup>2</sup> C Slave Address LSB
23	Y		SCL / SCLK	I <sup>2</sup> C serial clock (SCL); SPI serial clock (SCLK)
23		Y	SCL	I <sup>2</sup> C serial clock
24	Y		SDA / SDI	I <sup>2</sup> C serial data (SDA); SPI serial data input (SDI)
24		Y	SDA	I <sup>2</sup> C serial data

#### Note:

To prevent switching into I<sup>2</sup>C mode when using SPI (MPU-6000), the I<sup>2</sup>C interface should be disabled by setting the *I2C\_IF\_DIS* configuration bit. Setting this bit should be performed immediately after waiting for the time specified by the “Start-Up Time for Register Read/Write” in Section 6.3.

For further information regarding the *I2C\_IF\_DIS* bit, please refer to the MPU-6000/MPU-6050 Register Map and Register Descriptions document.

### 9.2 I<sup>2</sup>C Interface

I<sup>2</sup>C is a two-wire interface comprised of the signals serial data (SDA) and serial clock (SCL). In general, the lines are open-drain and bi-directional. In a generalized I<sup>2</sup>C interface implementation, attached devices can be a master or a slave. The master device puts the slave address on the bus, and the slave device with the matching address acknowledges the master.

The MPU-60X0 always operates as a slave device when communicating to the system processor, which thus acts as the master. SDA and SCL lines typically need pull-up resistors to VDD. The maximum bus speed is 400 kHz.

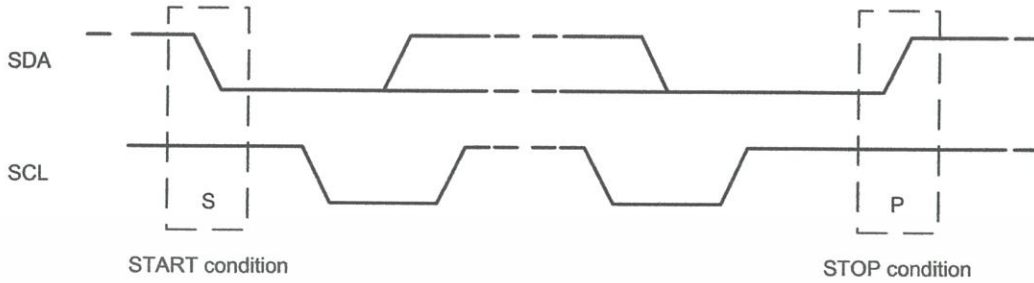
The slave address of the MPU-60X0 is b110100X which is 7 bits long. The LSB bit of the 7 bit address is determined by the logic level on pin AD0. This allows two MPU-60X0s to be connected to the same I<sup>2</sup>C bus. When used in this configuration, the address of the one of the devices should be b1101000 (pin AD0 is logic low) and the address of the other should be b1101001 (pin AD0 is logic high).

### 9.3 I<sup>2</sup>C Communications Protocol

#### START (S) and STOP (P) Conditions

Communication on the I<sup>2</sup>C bus starts when the master puts the START condition (S) on the bus, which is defined as a HIGH-to-LOW transition of the SDA line while SCL line is HIGH (see figure below). The bus is considered to be busy until the master puts a STOP condition (P) on the bus, which is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH (see figure below).

Additionally, the bus remains busy if a repeated START (Sr) is generated instead of a STOP condition.

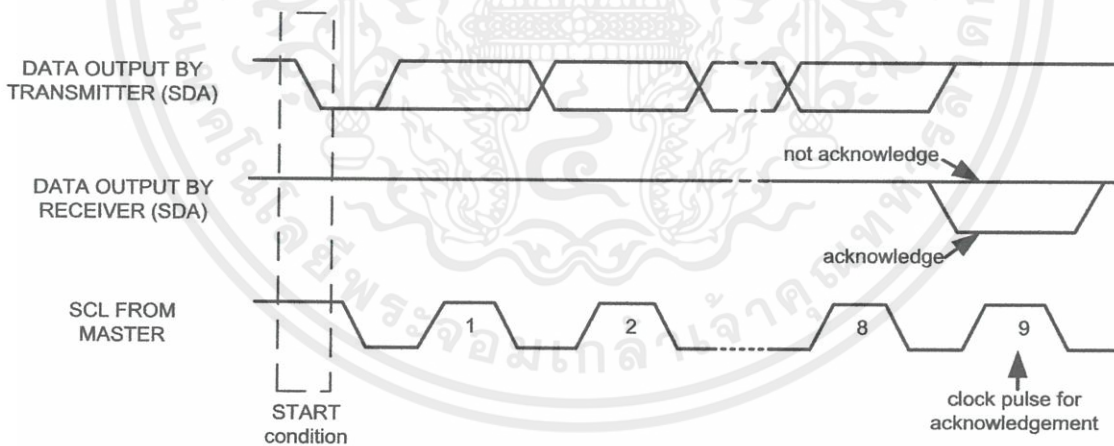


**START and STOP Conditions**

*Data Format / Acknowledge*

I<sup>2</sup>C data bytes are defined to be 8-bits long. There is no restriction to the number of bytes transmitted per data transfer. Each byte transferred must be followed by an acknowledge (ACK) signal. The clock for the acknowledge signal is generated by the master, while the receiver generates the actual acknowledge signal by pulling down SDA and holding it low during the HIGH portion of the acknowledge clock pulse.

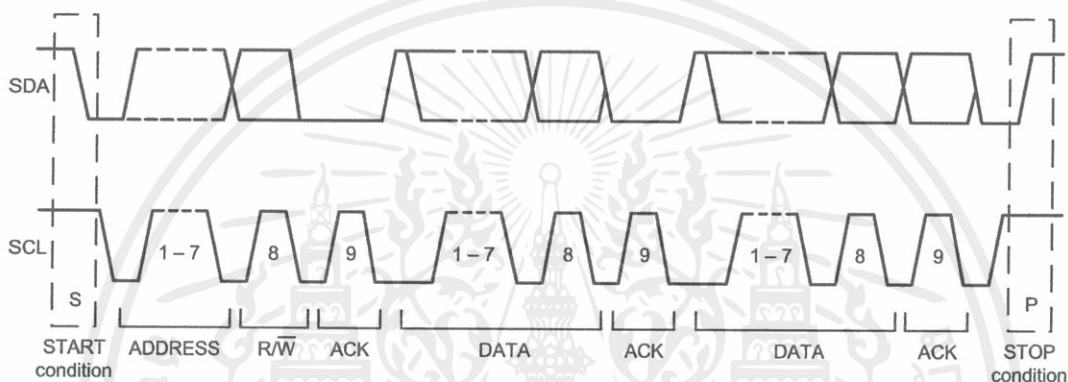
If a slave is busy and cannot transmit or receive another byte of data until some other task has been performed, it can hold SCL LOW, thus forcing the master into a wait state. Normal data transfer resumes when the slave is ready, and releases the clock line (refer to the following figure).



**Acknowledge on the I<sup>2</sup>C Bus**

### Communications

After beginning communications with the START condition (S), the master sends a 7-bit slave address followed by an 8<sup>th</sup> bit, the read/write bit. The read/write bit indicates whether the master is receiving data from or is writing to the slave device. Then, the master releases the SDA line and waits for the acknowledge signal (ACK) from the slave device. Each byte transferred must be followed by an acknowledge bit. To acknowledge, the slave device pulls the SDA line LOW and keeps it LOW for the high period of the SCL line. Data transmission is always terminated by the master with a STOP condition (P), thus freeing the communications line. However, the master can generate a repeated START condition (Sr), and address another slave without first generating a STOP condition (P). A LOW to HIGH transition on the SDA line while SCL is HIGH defines the stop condition. All SDA changes should take place when SCL is low, with the exception of start and stop conditions.



**Complete I<sup>2</sup>C Data Transfer**

To write the internal MPU-60X0 registers, the master transmits the start condition (S), followed by the I<sup>2</sup>C address and the write bit (0). At the 9<sup>th</sup> clock cycle (when the clock is high), the MPU-60X0 acknowledges the transfer. Then the master puts the register address (RA) on the bus. After the MPU-60X0 acknowledges the reception of the register address, the master puts the register data onto the bus. This is followed by the ACK signal, and data transfer may be concluded by the stop condition (P). To write multiple bytes after the last ACK signal, the master can continue outputting data rather than transmitting a stop signal. In this case, the MPU-60X0 automatically increments the register address and loads the data to the appropriate register. The following figures show single and two-byte write sequences.

#### Single-Byte Write Sequence

Master	S	AD+W		RA		DATA		P
Slave			ACK		ACK		ACK	

#### Burst Write Sequence

Master	S	AD+W		RA		DATA		DATA		P
Slave			ACK		ACK		ACK		ACK	

To read the internal MPU-60X0 registers, the master sends a start condition, followed by the I<sup>2</sup>C address and a write bit, and then the register address that is going to be read. Upon receiving the ACK signal from the MPU-60X0, the master transmits a start signal followed by the slave address and read bit. As a result, the MPU-60X0 sends an ACK signal and the data. The communication ends with a not acknowledge (NACK) signal and a stop bit from master. The NACK condition is defined such that the SDA line remains high at the 9<sup>th</sup> clock cycle. The following figures show single and two-byte read sequences.

#### Single-Byte Read Sequence

Master	S	AD+W		RA		S	AD+R			NACK	P
Slave			ACK		ACK			ACK	DATA		

#### Burst Read Sequence

Master	S	AD+W		RA		S	AD+R			ACK		NACK	P
Slave			ACK		ACK			ACK	DATA		DATA		

### 9.4 I<sup>2</sup>C Terms

Signal	Description
S	Start Condition: SDA goes from high to low while SCL is high
AD	Slave I <sup>2</sup> C address
W	Write bit (0)
R	Read bit (1)
ACK	Acknowledge: SDA line is low while the SCL line is high at the 9 <sup>th</sup> clock cycle
NACK	Not-Acknowledge: SDA line stays high at the 9 <sup>th</sup> clock cycle
RA	MPU-60X0 internal register address
DATA	Transmit or received data
P	Stop condition: SDA going from low to high while SCL is high

### 9.5 SPI Interface (MPU-6000 only)

SPI is a 4-wire synchronous serial interface that uses two control lines and two data lines. The MPU-6000 always operates as a Slave device during standard Master-Slave SPI operation.

With respect to the Master, the Serial Clock output (SCLK), the Serial Data Output (SDO) and the Serial Data Input (SDI) are shared among the Slave devices. Each SPI slave device requires its own Chip Select (/CS) line from the master.

/CS goes low (active) at the start of transmission and goes back high (inactive) at the end. Only one /CS line is active at a time, ensuring that only one slave is selected at any given time. The /CS lines of the non-selected slave devices are held high, causing their SDO lines to remain in a high-impedance (high-z) state so that they do not interfere with any active devices.

#### SPI Operational Features

1. Data is delivered MSB first and LSB last
2. Data is latched on the rising edge of SCLK
3. Data should be transitioned on the falling edge of SCLK
4. The maximum frequency of SCLK is 1MHz
5. SPI read and write operations are completed in 16 or more clock cycles (two or more bytes). The first byte contains the SPI Address, and the following byte(s) contain(s) the SPI data. The first bit of the first byte contains the Read/Write bit and indicates the Read (1) or Write (0) operation. The following 7 bits contain the Register Address. In cases of multiple-byte Read/Writes, data is two or more bytes:

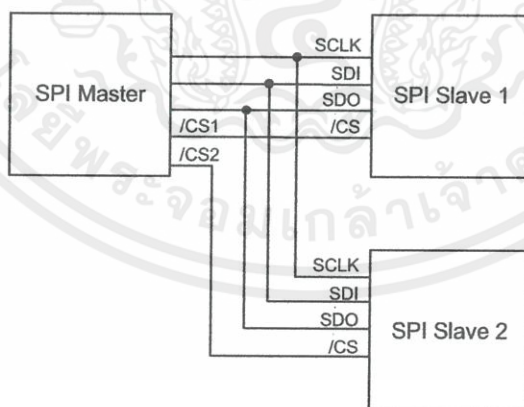
#### SPI Address format

MSB							LSB
R/W	A6	A5	A4	A3	A2	A1	A0

#### SPI Data format

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0

6. Supports Single or Burst Read/Writes.



**Typical SPI Master / Slave Configuration**

## 10 Serial Interface Considerations (MPU-6050)

### 10.1 MPU-6050 Supported Interfaces

The MPU-6050 supports I<sup>2</sup>C communications on both its primary (microprocessor) serial interface and its auxiliary interface.

### 10.2 Logic Levels

The MPU-6050's I/O logic levels are set to be either VDD or VLOGIC, as shown in the table below.

**I/O Logic Levels vs. AUX\_VDDIO**

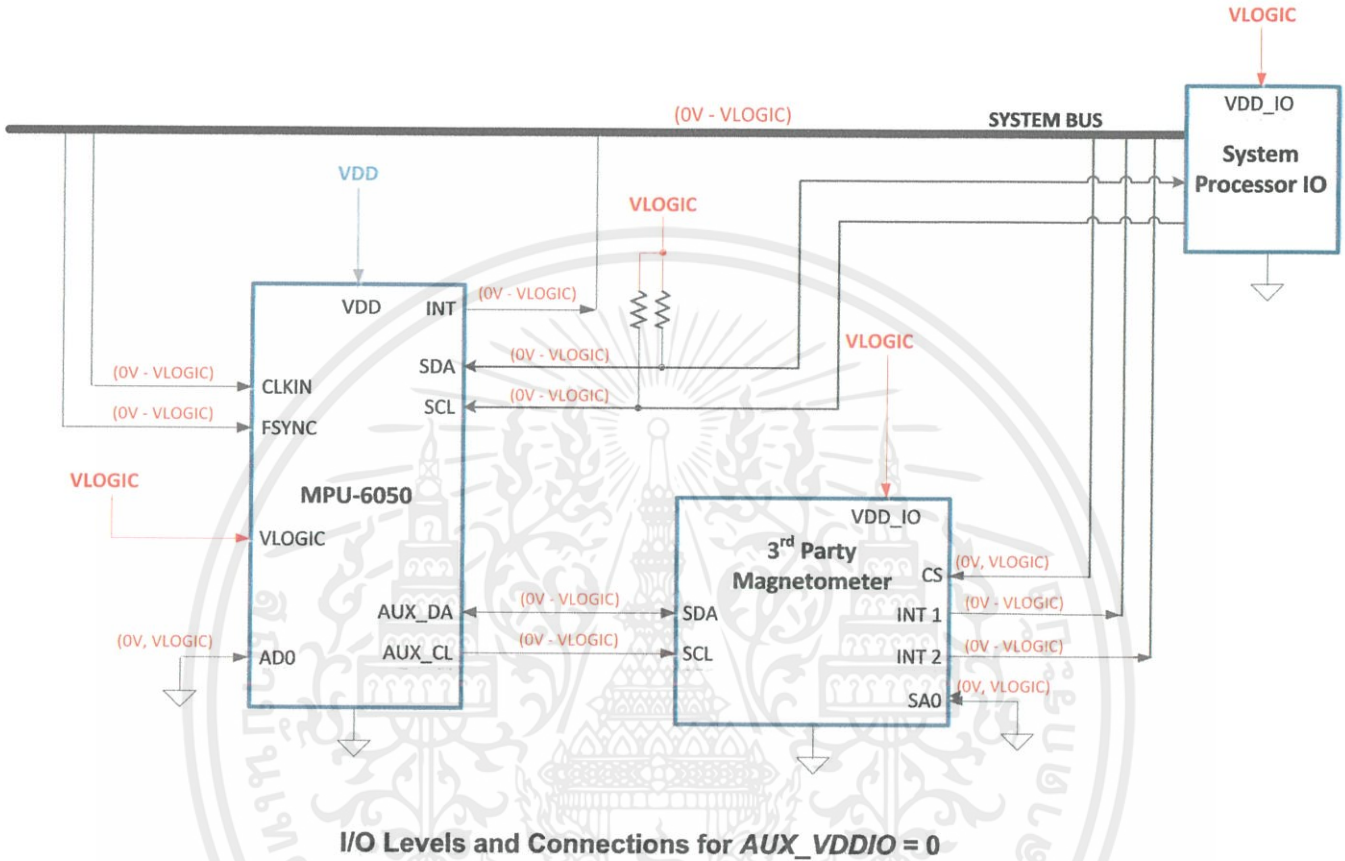
<b>AUX_VDDIO</b>	<b>MICROPROCESSOR LOGIC LEVELS</b> (Pins: SDA, SCL, AD0, CLKIN, INT)	<b>AUXILIARY LOGIC LEVELS</b> (Pins: AUX_DA, AUX_CL)
0	VLOGIC	VLOGIC
1	VLOGIC	VDD

Note: The power-on-reset value for AUX\_VDDIO is 0.

VLOGIC may be set to be equal to VDD or to another voltage. However, VLOGIC must be  $\leq$  VDD at all times. When AUX\_VDDIO is set to 0 (its power-on-reset value), VLOGIC is the power supply voltage for both the microprocessor system bus and the auxiliary I<sup>2</sup>C bus, as shown in the figure of Section 10.3. When AUX\_VDDIO is set to 1, VLOGIC is the power supply voltage for the microprocessor system bus and VDD is the supply for the auxiliary I<sup>2</sup>C bus, as shown in the figure of Section 10.4.

### 10.3 Logic Levels Diagram for AUX\_VDDIO = 0

The figure below depicts a sample circuit with a third party magnetometer attached to the auxiliary I<sup>2</sup>C bus. It shows logic levels and voltage connections for AUX\_VDDIO = 0. Note: Actual configuration will depend on the auxiliary sensors used.

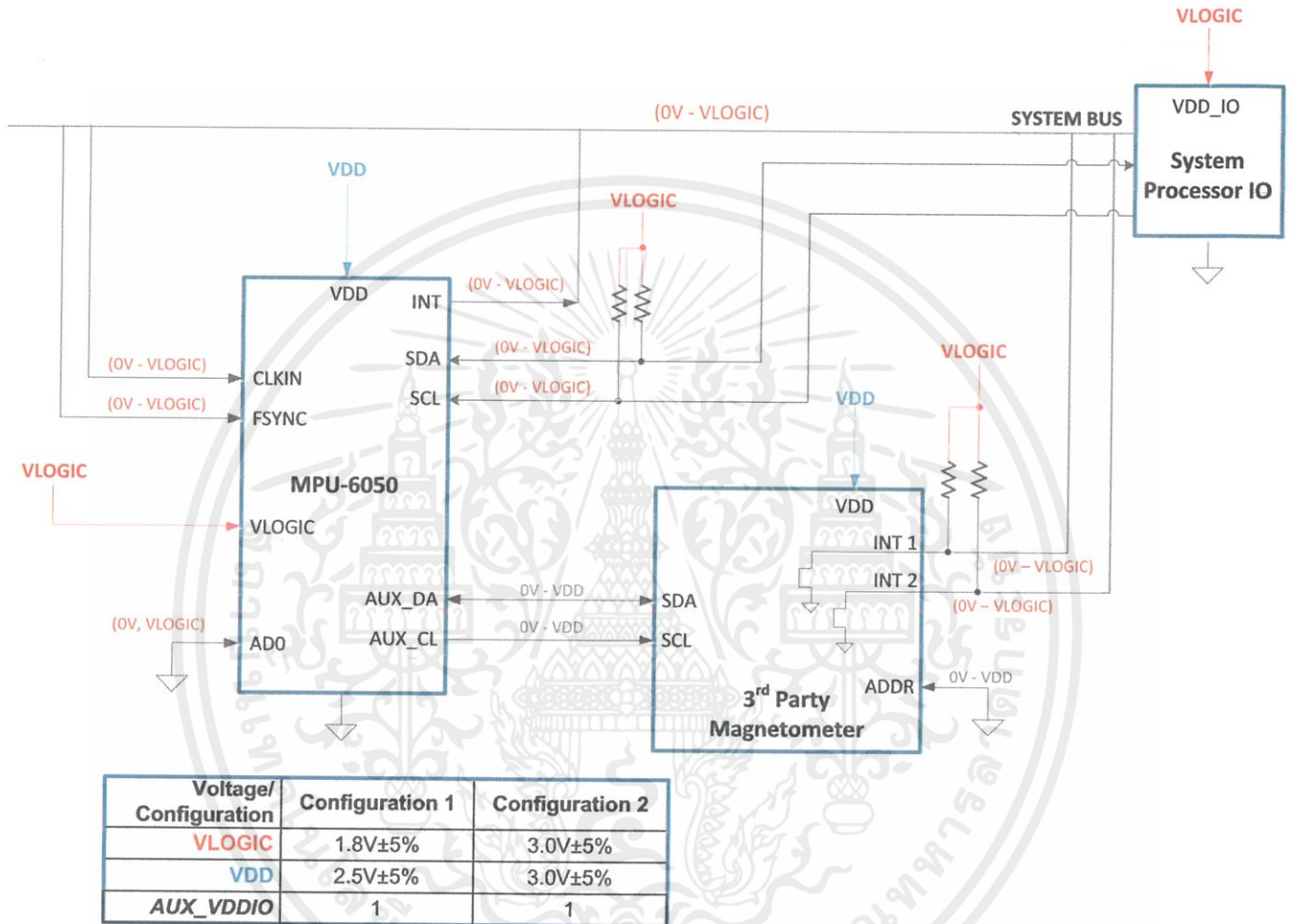


**Notes:**

1. AUX\_VDDIO determines the IO voltage levels of AUX\_DA and AUX\_CL (0 = set output levels relative to VLOGIC)
2. CLKOUT is referenced to VDD.
3. All other MPU-6050 logic IOs are referenced to VLOGIC.

**10.4 Logic Levels Diagram for AUX\_VDDIO = 1**

The figure below depicts a sample circuit with a 3<sup>rd</sup> party magnetometer attached to the auxiliary I<sup>2</sup>C bus. It shows logic levels and voltage connections for AUX\_VDDIO = 1. This configuration is useful when the auxiliary sensor has only one supply for logic and power. Note: Actual configuration will depend on the auxiliary sensors used.



**I/O Levels and Connections for Two Example Power Configurations (AUX\_VDDIO = 1)**

**Notes:**

1. AUX\_VDDIO determines the IO voltage levels of AUX\_DA and AUX\_CL. AUX\_VDDIO = 1 sets output levels relative to VDD.
2. 3<sup>rd</sup>-party auxiliary device logic levels are referenced to VDD. Setting INT1 and INT2 to open drain configuration provides voltage compatibility when VDD ≠ VLOGIC. When VDD = VLOGIC, INT1 and INT2 may be set to push-pull outputs, and external pull-up resistors are not needed.
3. CLKOUT is referenced to VDD.
4. All other MPU-6050 logic IOs are referenced to VLOGIC.

## 11 Assembly

This section provides general guidelines for assembling InvenSense Micro Electro-Mechanical Systems (MEMS) gyros packaged in Quad Flat No leads package (QFN) surface mount integrated circuits.

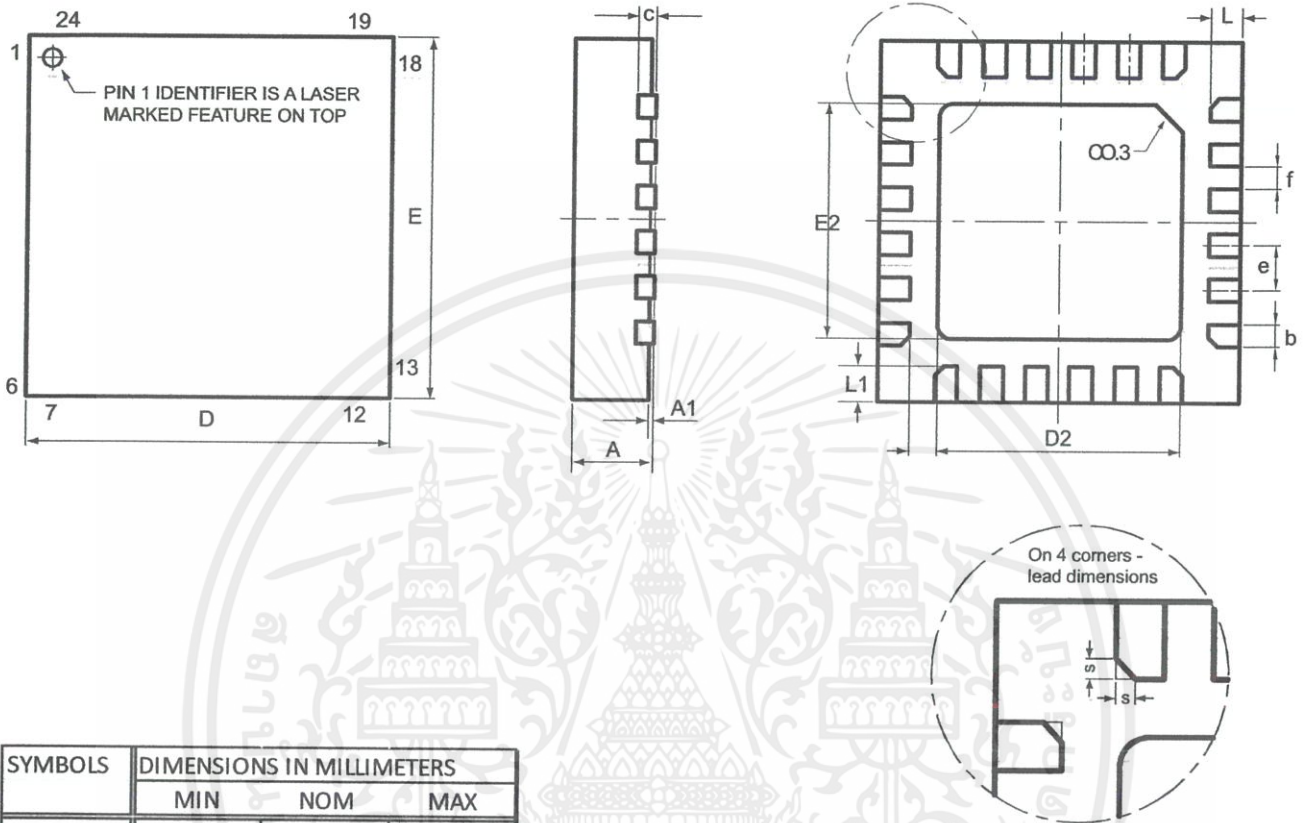
### 11.1 Orientation of Axes

The diagram below shows the orientation of the axes of sensitivity and the polarity of rotation. Note the pin 1 identifier (•) in the figure.



## 11.2 Package Dimensions

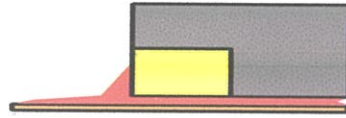
### 24 Lead QFN (4x4x0.9) mm NiPdAu Lead-frame finish



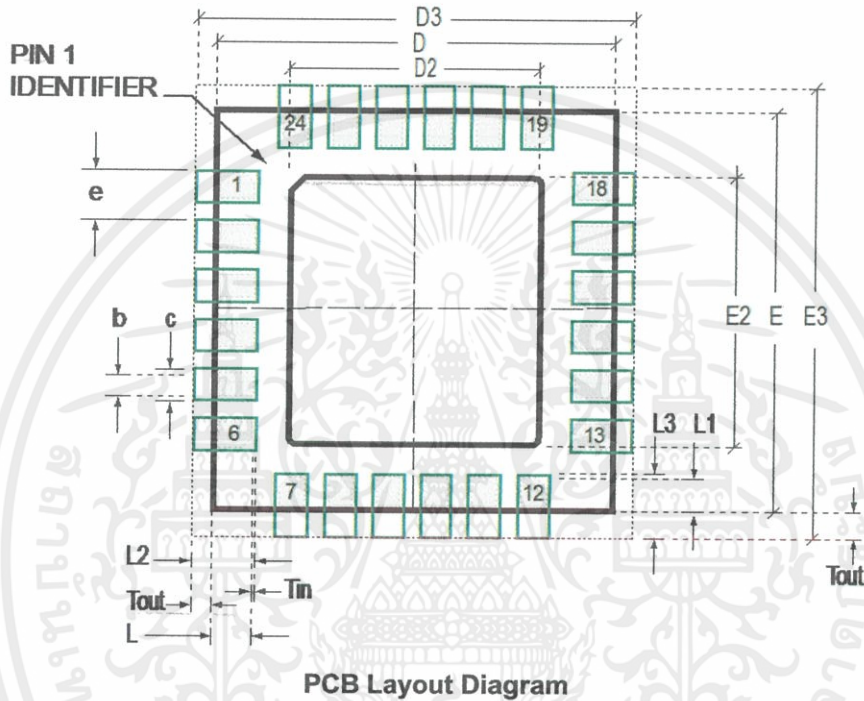
SYMBOLS	DIMENSIONS IN MILLIMETERS		
	MIN	NOM	MAX
A	0.85	0.90	0.95
A1	0.00	0.02	0.05
b	0.18	0.25	0.30
c	---	0.20 REF	---
D	3.90	4.00	4.10
D2	2.65	2.70	2.75
E	3.90	4.00	4.10
E2	2.55	2.60	2.65
e	---	0.50	---
f (e-b)	---	0.25	---
K	0.25	0.30	0.35
L	0.30	0.35	0.40
L1	0.35	0.40	0.45
s	0.05	---	0.15

### 11.3 PCB Design Guidelines

The Pad Diagram using a JEDEC type extension with solder rising on the outer edge is shown below. The Pad Dimensions Table shows pad sizing (mean dimensions) recommended for the MPU-60X0 product.



JEDEC type extension with solder rising on outer edge



PCB Layout Diagram

SYMBOLS	DIMENSIONS IN MILLIMETERS	NOM
<b>Nominal Package I/O Pad Dimensions</b>		
e	Pad Pitch	0.50
b	Pad Width	0.25
L	Pad Length	0.35
L1	Pad Length	0.40
D	Package Width	4.00
E	Package Length	4.00
D2	Exposed Pad Width	2.70
E2	Exposed Pad Length	2.60
<b>I/O Land Design Dimensions (Guidelines)</b>		
D3	I/O Pad Extent Width	4.80
E3	I/O Pad Extent Length	4.80
c	Land Width	0.35
Tout	Outward Extension	0.40
Tin	Inward Extension	0.05
L2	Land Length	0.80
L3	Land Length	0.85

PCB Dimensions Table (for PCB Lay-out Diagram)

## 11.4 Assembly Precautions

### 11.4.1 Gyroscope Surface Mount Guidelines

InvenSense MEMS Gyros sense rate of rotation. In addition, gyroscopes sense mechanical stress coming from the printed circuit board (PCB). This PCB stress can be minimized by adhering to certain design rules:

When using MEMS gyroscope components in plastic packages, PCB mounting and assembly can cause package stress. This package stress in turn can affect the output offset and its value over a wide range of temperatures. This stress is caused by the mismatch between the Coefficient of Linear Thermal Expansion (CTE) of the package material and the PCB. Care must be taken to avoid package stress due to mounting.

Traces connected to pads should be as symmetric as possible. Maximizing symmetry and balance for pad connection will help component self alignment and will lead to better control of solder paste reduction after reflow.

Any material used in the surface mount assembly process of the MEMS gyroscope should be free of restricted RoHS elements or compounds. Pb-free solders should be used for assembly.

### 11.4.2 Exposed Die Pad Precautions

The MPU-60X0 has very low active and standby current consumption. The exposed die pad is not required for heat sinking, and should not be soldered to the PCB. Failure to adhere to this rule can induce performance changes due to package thermo-mechanical stress. There is no electrical connection between the pad and the CMOS.

### 11.4.3 Trace Routing

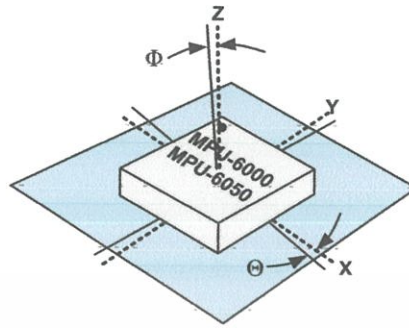
Routing traces or vias under the gyro package such that they run under the exposed die pad is prohibited. Routed active signals may harmonically couple with the gyro MEMS devices, compromising gyro response. These devices are designed with the drive frequencies as follows: X =  $33\pm 3\text{KHz}$ , Y =  $30\pm 3\text{KHz}$ , and Z =  $27\pm 3\text{KHz}$ . To avoid harmonic coupling don't route active signals in non-shielded signal planes directly below, or above the gyro package. Note: For best performance, design a ground plane under the e-pad to reduce PCB signal noise from the board on which the gyro device is mounted. If the gyro device is stacked under an adjacent PCB board, design a ground plane directly above the gyro device to shield active signals from the adjacent PCB board.

### 11.4.4 Component Placement

Do not place large insertion components such as keyboard or similar buttons, connectors, or shielding boxes at a distance of less than 6 mm from the MEMS gyro. Maintain generally accepted industry design practices for component placement near the MPU-60X0 to prevent noise coupling and thermo-mechanical stress.

### 11.4.5 PCB Mounting and Cross-Axis Sensitivity

Orientation errors of the gyroscope and accelerometer mounted to the printed circuit board can cause cross-axis sensitivity in which one gyro or accel responds to rotation or acceleration about another axis, respectively. For example, the X-axis gyroscope may respond to rotation about the Y or Z axes. The orientation mounting errors are illustrated in the figure below.



**Package Gyro & Accel Axes ( - - - ) Relative to PCB Axes ( — ) with Orientation Errors ( $\Theta$  and  $\Phi$ )**

The table below shows the cross-axis sensitivity as a percentage of the gyroscope or accelerometer's sensitivity for a given orientation error, respectively.

**Cross-Axis Sensitivity vs. Orientation Error**

Orientation Error ( $\theta$ or $\Phi$ )	Cross-Axis Sensitivity ( $\sin\theta$ or $\sin\Phi$ )
0°	0%
0.5°	0.87%
1°	1.75%

The specifications for cross-axis sensitivity in Section 6.1 and Section 6.2 include the effect of the die orientation error with respect to the package.

#### 11.4.6 MEMS Handling Instructions

MEMS (Micro Electro-Mechanical Systems) are a time-proven, robust technology used in hundreds of millions of consumer, automotive and industrial products. MEMS devices consist of microscopic moving mechanical structures. They differ from conventional IC products, even though they can be found in similar packages. Therefore, MEMS devices require different handling precautions than conventional ICs prior to mounting onto printed circuit boards (PCBs).

The MPU-60X0 has been qualified to a shock tolerance of 10,000g. InvenSense packages its gyroscopes as it deems proper for protection against normal handling and shipping. It recommends the following handling precautions to prevent potential damage.

- Do not drop individually packaged gyroscopes, or trays of gyroscopes onto hard surfaces. Components placed in trays could be subject to g-forces in excess of 10,000g if dropped.
- Printed circuit boards that incorporate mounted gyroscopes should not be separated by manually snapping apart. This could also create g-forces in excess of 10,000g.
- Do not clean MEMS gyroscopes in ultrasonic baths. Ultrasonic baths can induce MEMS damage if the bath energy causes excessive drive motion through resonant frequency coupling.

#### 11.4.7 ESD Considerations

Establish and use ESD-safe handling precautions when unpacking and handling ESD-sensitive devices.

- Store ESD sensitive devices in ESD safe containers until ready for use. The Tape-and-Reel moisture-sealed bag is an ESD approved barrier. The best practice is to keep the units in the original moisture sealed bags until ready for assembly.

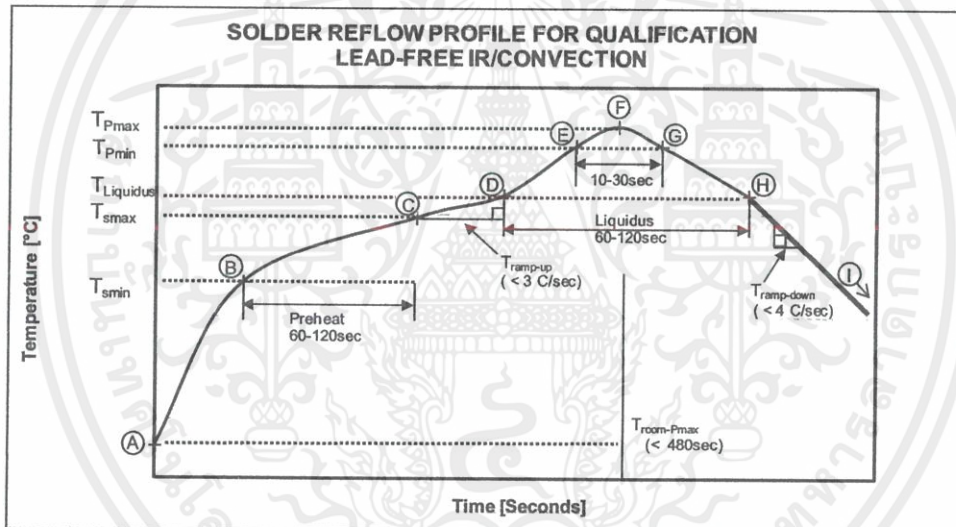
Restrict all device handling to ESD protected work areas that measure less than 200V static charge. Ensure that all workstations and personnel are properly grounded to prevent ESD.

**11.4.8 Reflow Specification**

**Qualification Reflow:** The MPU-60X0 was qualified in accordance with IPC/JEDEC J-STD-020D.01. This standard classifies proper packaging, storage and handling in order to avoid subsequent thermal and mechanical damage during the solder reflow attachment phase of PCB assembly.

The qualification preconditioning process specifies a sequence consisting of a bake cycle, a moisture soak cycle (in a temperature humidity oven), and three consecutive solder reflow cycles, followed by functional device testing.

The peak solder reflow classification temperature requirement for package qualification is (260 +5/-0°C) for lead-free soldering of components measuring less than 1.6 mm in thickness. The qualification profile and a table explaining the set-points are shown below:



**Temperature Set Points Corresponding to Reflow Profile Above**

Step	Setting	CONSTRAINTS		
		Temp (°C)	Time (sec)	Max. Rate (°C/sec)
A	T <sub>room</sub>	25		
B	T <sub>Smin</sub>	150		
C	T <sub>Smax</sub>	200	60 < t <sub>BC</sub> < 120	
D	T <sub>Liquidus</sub>	217		Γ(T <sub>Liquidus</sub> -T <sub>Pmax</sub> ) < 3
E	T <sub>Pmin</sub> [255°C, 260°C]	255		Γ(T <sub>Liquidus</sub> -T <sub>Pmax</sub> ) < 3
F	T <sub>Pmax</sub> [260°C, 265°C]	260	t <sub>AF</sub> < 480	Γ(T <sub>Liquidus</sub> -T <sub>Pmax</sub> ) < 3
G	T <sub>Pmin</sub> [255°C, 260°C]	255	10 < t <sub>EG</sub> < 30	Γ(T <sub>Pmax</sub> -T <sub>Liquidus</sub> ) < 4
H	T <sub>Liquidus</sub>	217	60 < t <sub>DH</sub> < 120	
I	T <sub>room</sub>	25		

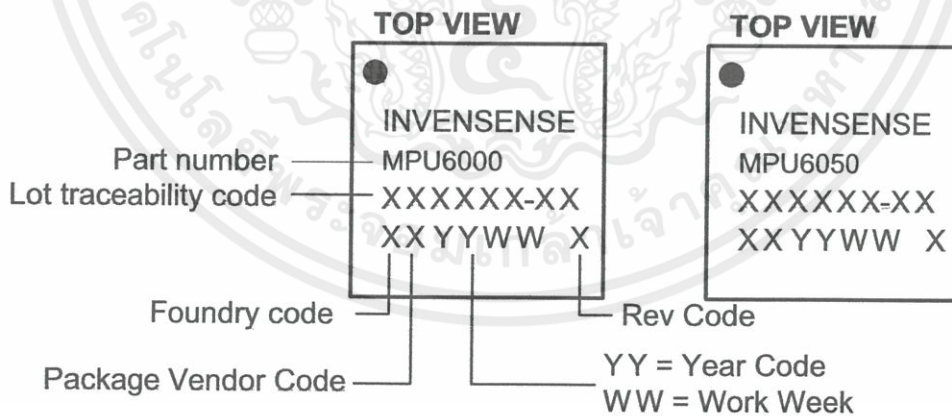
**Notes:** Customers must never exceed the Classification temperature (T<sub>Pmax</sub> = 260°C).  
 All temperatures refer to the topside of the QFN package, as measured on the package body surface.

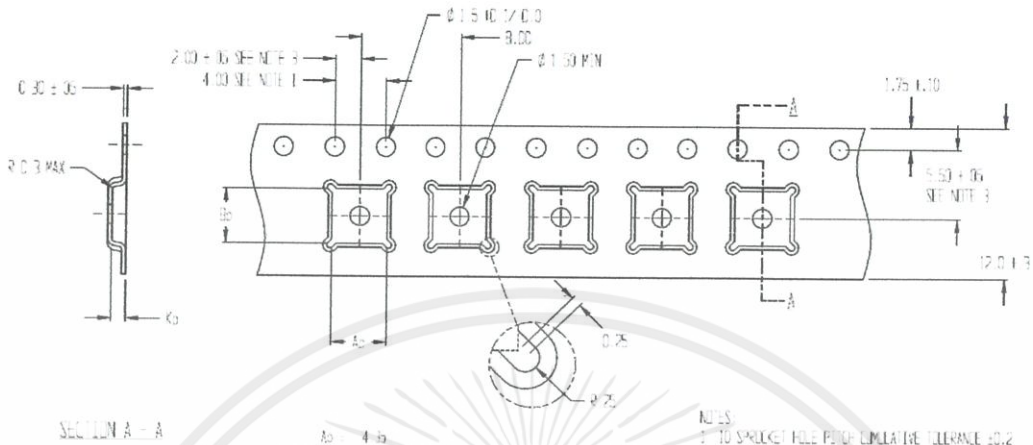
**Production Reflow:** Check the recommendations of your solder manufacturer. For optimum results, use lead-free solders that have lower specified temperature profiles (T<sub>pmax</sub> ~ 235°C). Also use lower ramp-up and ramp-down rates than those used in the qualification profile. Never exceed the maximum conditions that we used for qualification, as these represent the maximum tolerable ratings for the device.

**11.5 Storage Specifications**

The storage specification of the MPU-60X0 conforms to IPC/JEDEC J-STD-020D.01 Moisture Sensitivity Level (MSL) 3.

Calculated shelf-life in moisture-sealed bag	12 months -- Storage conditions: <40°C and <90% RH
After opening moisture-sealed bag	168 hours -- Storage conditions: ambient ≤30°C at 60%RH

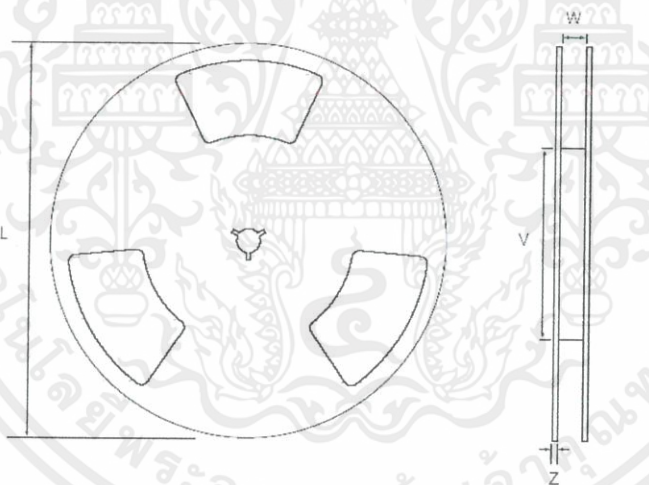
**11.6 Package Marking Specification**

**Package Marking Specification**

**11.7 Tape & Reel Specification**


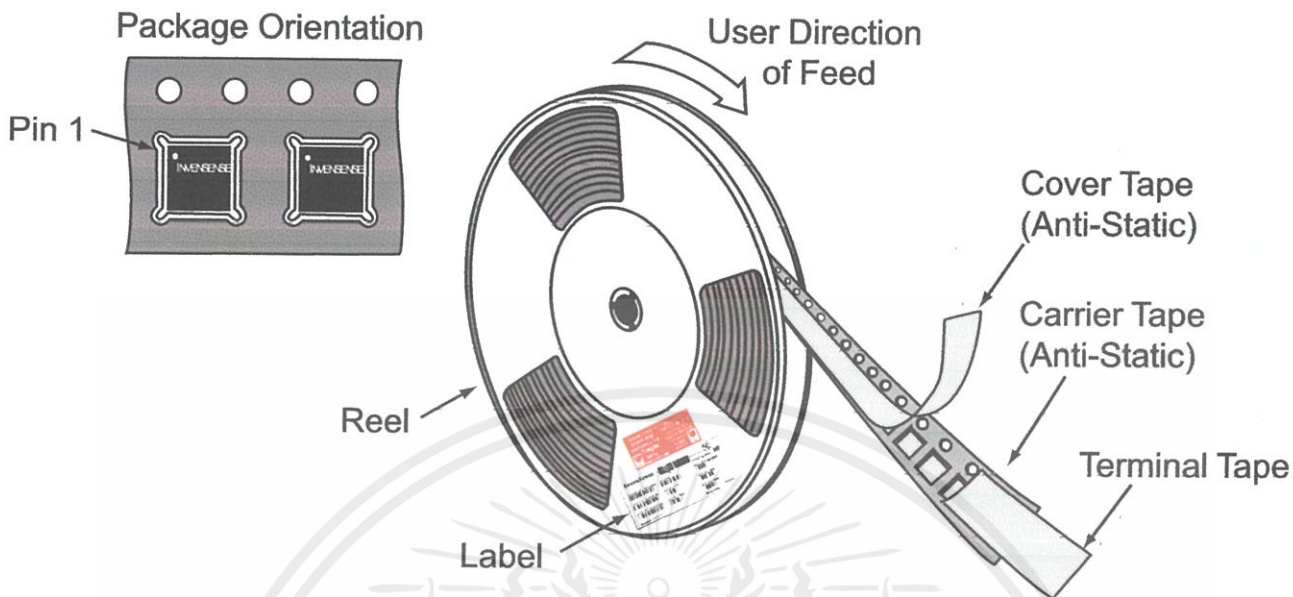
NOTES:  
 1. TO SPAGLET HOLE PITCH: CUMULATIVE TOLERANCE ±0.2  
 2. CAMBER IN COMPLIANCE WITH EIA-481  
 3. PITCH POSITION TOLERANCE: SPAGLET HOLE POSITION AS PER POSITION TOLERANCE, ALL PITCH HOLE

TOLERANCE: ±0.05  
 UNLESS OTHERWISE SPECIFIED

A<sub>0</sub> = 4.45  
 A<sub>1</sub> = 4.75  
 K<sub>0</sub> = 1.15

**Tape Dimensions**

**Reel Outline Drawing**
**Reel Dimensions and Package Size**

PACKAGE SIZE	REEL (mm)			
	L	V	W	Z
4x4	330	100	13.2	2.2



**Tape and Reel Specification**

**Reel Specifications**

Quantity Per Reel	5,000
Reels per Box	1
Boxes Per Carton (max)	5
Pcs/Carton (max)	25,000

**11.8 Label**

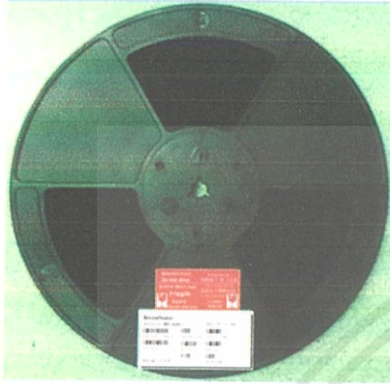


Barcode Label

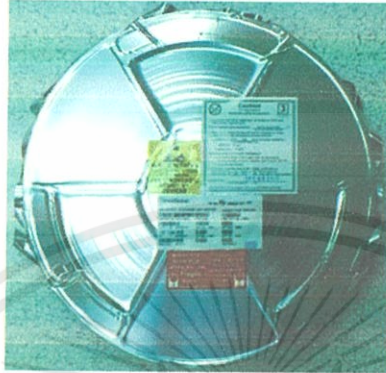


Location of Label on Reel

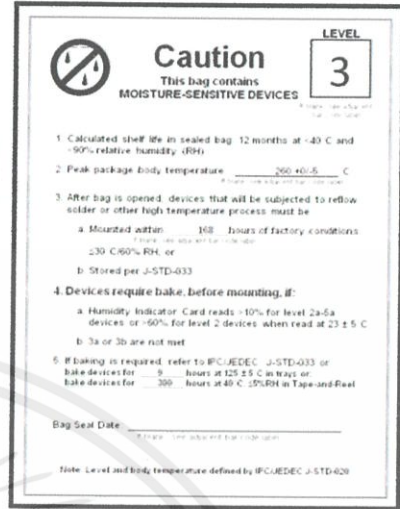
### 11.9 Packaging



**REEL – with Barcode & Caution labels**



**Vacuum-Sealed Moisture Barrier Bag with ESD, MSL3, Caution, and Barcode Labels**



**MSL3 Label**



**Caution Label**



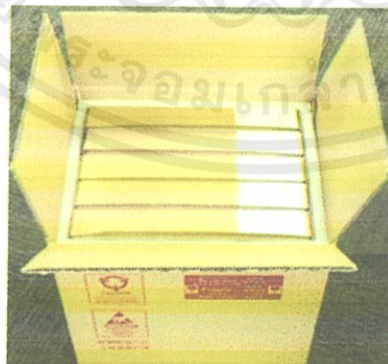
**ESD Label**



**Inner Bubble Wrap**



**Pizza Box**



**Pizza Boxes Placed in Foam-Lined Shipper Box**



**Outer Shipper Label**

**11.10 Representative Shipping Carton Label**

		<b>INV. NO:</b> <b>111013-99</b>	
<b>From:</b> InvenSense Taiwan, Ltd. 1F, 9 Prosperity 1st Road, Hsinchu Science Park, HsinChu City, 30078, Taiwan TEL: +886 3 6686999 FAX: +886 3 6686777		<b>Ship To:</b> Customer Name Street Address City, State, Country ZIP Attn: Buyer Name Phone: Buyer Phone Number	
<b>SUPP PROD ID:</b> MPU-6050			
<b>LOT#: Q2R994-F1</b>	<b>LOT#:</b>		
<b>QTY: 5615</b>	<b>QTY: 0</b>		
<b>LOT#: Q3X785-G1</b>	<b>LOT#:</b>		
<b>QTY: 4385</b>	<b>QTY: 0</b>		
<b>LOT#: Q3Y196-02</b>	<b>LOT#:</b>		
<b>QTY: 5000</b>	<b>QTY: 0</b>		
<b>LOT#:</b>	<b>LOT#:</b>		
<b>QTY: 0</b>	<b>QTY: 0</b>		
<b>Total Quantity/Carton</b> 15000		<b>Weight: (KG)</b> 4.05	
<b>Pb-free</b>	<b>Shipping Carton:</b> 1	3	<b>Category (e4) HF</b>
<b>MSL3</b>	<b>OF</b>		

## 12 Reliability

### 12.1 Qualification Test Policy

InvenSense's products complete a Qualification Test Plan before being released to production. The Qualification Test Plan for the MPU-60X0 followed the JESD 47H.01 Standards, "Stress-Test-Driven Qualification of Integrated Circuits," with the individual tests described below.

### 12.2 Qualification Test Plan

#### Accelerated Life Tests

TEST	Method/Condition	Lot Quantity	Sample / Lot	Acc / Reject Criteria
(HTOL/LFR) High Temperature Operating Life	JEDEC JESD22-A108D, Dynamic, 3.63V biased, Tj>125°C [read-points 168, 500, 1000 hours]	3	77	(0/1)
(HAST) Highly Accelerated Stress Test <sup>(1)</sup>	JEDEC JESD22-A118A Condition A, 130°C, 85%RH, 33.3 psia., unbiased, [read-point 96 hours]	3	77	(0/1)
(HTS) High Temperature Storage Life	JEDEC JESD22-A103D, Cond. A, 125°C Non-Bias Bake [read-points 168, 500, 1000 hours]	3	77	(0/1)

#### Device Component Level Tests

TEST	Method/Condition	Lot Quantity	Sample / Lot	Acc / Reject Criteria
(ESD-HBM) ESD-Human Body Model	JEDEC JS-001-2010, (1.5KV)	1	3	(0/1)
(ESD-MM) ESD-Machine Model	JEDEC JESD22-A115C, (200V)	1	3	(0/1)
(LU) Latch Up	JEDEC JESD-78D Class II (2), 125°C; ±100mA	1	6	(0/1)
(MS) Mechanical Shock	JEDEC JESD22-B104C, Mil-Std-883, Method 2002.5, Cond. E, 10,000g's, 0.2ms, ±X, Y, Z – 6 directions, 5 times/direction	3	5	(0/1)
(VIB) Vibration	JEDEC JESD22-B103B, Variable Frequency (random), Cond. B, 5-500Hz, X, Y, Z – 4 times/direction	3	5	(0/1)
(TC) Temperature Cycling <sup>(1)</sup>	JEDEC JESD22-A104D Condition N [-40°C to +85°C], Soak Mode 2 [5], 100 cycles	3	77	(0/1)

#### Board Level Tests

TEST	Method/Condition	Lot Quantity	Sample / Lot	Acc / Reject Criteria
(BMS) Board Mechanical Shock	JEDEC JESD22-B104C, Mil-Std-883, Method 2002.5, Cond. E, 10000g's, 0.2ms, +X, Y, Z – 6 directions, 5 times/direction	1	5	(0/1)
(BTC) Board Temperature Cycling <sup>(1)</sup>	JEDEC JESD22-A104D Condition N [-40°C to +85°C], Soak mode 2 [5], 100 cycles	1	40	(0/1)

(1) Tests are preceded by MSL3 Preconditioning in accordance with JEDEC JESD22-A113F