

อุปกรณ์จำกัดและวัดมุมงอของเข่า
Knee Support and Braces for Rehabilitation Patient



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมชีวการแพทย์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

อุปกรณ์จำกัดและวัดมุมงอของเข่า

Knee Support and Braces for Rehabilitation Patient

โดย



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมชีวการแพทย์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ ปีการศึกษา 2559
สาขาวิชา วิศวกรรมชีวการแพทย์
คณะ วิศวกรรมศาสตร์
เรื่อง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 อุปกรณ์จำกัดและวัดมุมของเข่า
 Knee Support and Braces for Rehabilitation Patient
ผู้จัดทำ ภาคภูมิ เจริญอารีย์ รหัส 56010920
 รายงานนี้ผ่านการตรวจโดยอาจารย์ที่ปรึกษาแล้ว



(รศ.ดร.ชูชาติ ปิณฑวิรุจน์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์	อุปกรณ์จำกัดและวัดมุมของเข่า	
นักศึกษา	นายภาคภูมิ เหริยญอารีย์	รหัสประจำตัว 56010920
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมชีวการแพทย์	
ปีการศึกษา	2559	
อาจารย์ที่ปรึกษาปริญญาานิพนธ์	รศ.ดร.ชูชาติ พิณฑวิรุจน์	

บทคัดย่อ

ในปัจจุบัน อาการบาดเจ็บที่หัวเข่าพบมากขึ้นในประเทศไทย ทั้งจากการเล่นกีฬา หรือจากการประสบอุบัติเหตุ ซึ่งการรักษาส่วนใหญ่จะเป็นการผ่าตัด ดังนั้น หลังการผ่าตัดผู้ป่วยต้องได้รับการดูแลอย่างใกล้ชิดเพื่อป้องกันผลแทรกซ้อนต่างๆ และการบาดเจ็บซ้ำซ้อน อีกทั้งยังต้องได้รับการกายภาพบำบัดเพื่อฟื้นฟูการทำงานของเขา เพื่อให้กลับมาใช้งานได้อย่างปกติ และสามารถใช้ชีวิตประจำวันได้ปกติดังเดิม ด้วยเหตุนี้จึงออกแบบและสร้างอุปกรณ์ในการศึกษาครั้งนี้ขึ้น เพื่อเป็นอุปกรณ์ที่สามารถจำกัดมุมงอเข่าของผู้ป่วยได้โดยกลไกภายในอุปกรณ์ เพื่อป้องกันการเกิดอาการบาดเจ็บซ้ำซ้อน และวัดองศาการงอของเข่าผู้ป่วยได้ โดยใช้ 3-axis Accelerometer/Gyro Module (IMU) และนำข้อมูลนั้นมาแสดงผลผ่านโปรแกรมคอมพิวเตอร์ที่สร้างขึ้น เพื่อให้แพทย์สามารถปรับเปลี่ยนการกายภาพบำบัดให้เหมาะสมต่อผู้ป่วยเป็นเฉพาะรายบุคคลได้ จะส่งผลให้ผู้ป่วยกลับไปใช้ชีวิตได้อย่างปกติโดยเร็ว โดยตัวอุปกรณ์ออกแบบโดยโปรแกรมคอมพิวเตอร์ช่วยในการออกแบบ (CAD) และมีการจำลองการทำงาน โดยอุปกรณ์จำกัดและวัดมุมงอของเข่ามีวงจรรีเลย์ทรานซิสเตอร์ติดต่อกับคอมพิวเตอร์ด้วยระบบไร้สาย (Bluetooth) เพื่อส่งข้อมูลมาแสดงผลบนโปรแกรมที่สร้างขึ้น ซึ่งสามารถแสดงค่ามุมงอของเข่าได้แบบทันที ซึ่งสามารถวัดค่ามุมงอเข่าได้ในขณะที่ผู้ป่วยเดิน หรือทำกิจกรรมต่างๆ ได้ และสามารถบันทึกค่ามุมงอของเข่าไว้ใช้วินิจฉัยอาการ และการฟื้นตัวของผู้ป่วยได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Title	Knee Support and Braces for Rehabilitation Patient	
Student	Parkbhum Rean-aree	Student ID 56010920
Degree	Bachelor of Engineering	
Program	Biomedical Engineering	
Year	2016	
Thesis Advisor	Assoc.Prof.Dr. Chuchart Pintavirooj	

ABSTRACT

Today, knee injuries have become more common in Thailand, either from the sport or from the accident, the treatment is mostly a surgery. Hence, after the surgery, the patient must be closely monitored to prevent complications and repetitive injury. It also requires physical therapy to restore the knee function to resume a normal use. For this reason, the idea of creating a device that can restrict the patient's knee bend by internal mechanisms to prevent repetitive injury was brought out by measuring the degree of bending of the patient's knee using 3-axis Accelerometer / Gyro Module (IMU). The information is then displayed through a computer program created so that the doctor can adjust the therapy to suit a particular patient individually, resulting in the patient returning to normal life as quickly as possible. The device is designed by computer aided design (CAD) software and is simulated. The device has an electronic circuit that communicates with the computer wirelessly (Bluetooth) to send data to the display on the generated program. The bending angle can be displayed immediately and recorded for a diagnosis and recovery of the patient.

กิตติกรรมประกาศ

การทำวิทยานิพนธ์นี้สามารถสำเร็จลุล่วงไปได้ด้วยดีเนื่องจากได้รับความอนุเคราะห์และคำปรึกษาจากบุคคลหลายท่าน ตลอดทั้งให้ความรู้ในการทำโครงการแก่คณะผู้จัดทำมาโดยตลอด คณะผู้จัดทำขอขอบพระคุณหลักสูตรวิศวกรรมชีวการแพทย์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่สนับสนุนเครื่องมือ สถานที่และงบประมาณในการศึกษา ออกแบบ สร้างและทดสอบต้นแบบเครื่องในวิทยานิพนธ์นี้

คณะผู้จัดทำขอขอบพระคุณ รศ.ดร. ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษาโครงการ ที่ให้ความสนับสนุน แนะนำทาง ให้ความกรุณาตรวจผลงานและแก้ไขข้อบกพร่องในเอกสารและชิ้นงานให้มีความสมบูรณ์มากยิ่งขึ้น รวมถึงให้ความช่วยเหลือในเรื่องต่างๆ อย่างเต็มที่ ขอขอบพระคุณอาจารย์ประจำหลักสูตรที่ถ่ายทอดวิชา ความรู้ต่างๆ แก่คณะผู้จัดทำ

สุดท้ายนี้ขอขอบพระคุณ บิดา มารดาที่เลี้ยงดู อบรมสั่งสอนและให้การศึกษาย่างดี ทำให้คณะผู้จัดทำมีวันนี้ได้ และยังให้คำปรึกษาในทุกๆ ด้าน รวมถึงเป็นแรงผลักดันให้คณะผู้จัดทำในการทำสิ่งต่างๆ ให้ประสบความสำเร็จ

ภาคภูมิ เจริญอารีย์
ผู้จัดทำ

สารบัญ

หน้า

บทคัดย่อ.....	I
Abstract.....	II
กิตติกรรมประกาศ.....	III
สารบัญ	IV
สารบัญรูปภาพ.....	VI
สารบัญตาราง.....	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมุติฐานของการศึกษา.....	1
1.4 ขอบเขตการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 หลักการและทฤษฎี.....	3
2.1 การบาดเจ็บที่เข้า.....	3
2.2 ไมโครคอนโทรลเลอร์ (Microcontroller).....	12
2.3 Inertial measurement unit (IMU).....	15
2.4 การสื่อสารอนุกรมแบบ I2C (Inter Integrate Circuit Bus)	15
2.5 Accelerometer และ Gyro Module.....	18
2.6 ภาษา C#	26
2.7 งานวิจัยที่เกี่ยวข้อง.....	30
บทที่ 3 วิธีดำเนินการวิจัย	33
3.1 วิธีที่ใช้ศึกษาค้นคว้าและการวิจัยทดลอง.....	33
3.2 ลักษณะข้อมูล การเลือกข้อมูล และการทดลอง.....	33
3.3 เครื่องมือและวิธีการวิจัยทดลอง	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.4 ขั้นตอนออกแบบและสร้างเครื่องมือ.....	34
3.5 ขั้นตอนการทดลองการทำงานของอุปกรณ์.....	44
บทที่ 4 การทดลองและผลการทดลอง.....	45
4.1 อุปกรณ์จำกัดและวัดมุมของเข่า.....	45
4.2 โปรแกรมแสดงผลการวัดค่ามุมของเข่า.....	46
4.3 การทดสอบการทำงานของการวัดมุมของเข่า.....	47
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	48
5.1 สรุปผลการวิจัย.....	48
5.2 ปัญหาและแนวทางการแก้ปัญหา.....	48
5.3 ข้อเสนอแนะ.....	48
บรรณานุกรม.....	49
ภาคผนวก.....	51

สารบัญรูป

หน้า

รูปที่ 2.1	กายวิภาคของเข่า	3
รูปที่ 2.2	เส้นเอ็นต่างๆ ของข้อเข่า (ligament).....	4
รูปที่ 2.4	เส้นเอ็นของข้อเข่า และหมอนรองกระดูก (menisci)	6
รูปที่ 2.5	Range of motion ของเข่า	7
รูปที่ 2.6	การบริหารกล้ามเนื้อบริเวณเข่า.....	10
รูปที่ 2.7	Arduino UNO.....	14
รูปที่ 2.8	Arduino Micro	14
รูปที่ 2.9	Arduino IDE	15
รูปที่ 2.10	การสื่อสารอนุกรมแบบ I2C.....	16
รูปที่ 2.11	การเขียน-อ่านข้อมูลกับอุปกรณ์แบบ I2C BUS.....	16
รูปที่ 2.12	การกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I2C BUS.....	17
รูปที่ 2.13	รหัสควบคุมของ I2C BUS.....	17
รูปที่ 2.14	แสดงการรับส่งบิตข้อมูลของ I2C BUS.....	18
รูปที่ 2.15	โครงสร้างของ Accelerometer อาศัยแรงเฉื่อย.....	19
รูปที่ 2.16	โครงสร้างของ Accelerometer เพียโซอิเล็กทริก.....	19
รูปที่ 2.17	โครงสร้างพื้นฐานของมิเตอร์วัดอัตราเร่งแบบโซลิดสเตต.....	20
รูปที่ 2.18	โครงสร้างพื้นฐานของมิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริก.....	22
รูปที่ 2.19	Roll, Pitch, Yaw angles	24
รูปที่ 2.20	Mechanic gyroscope	24
รูปที่ 2.21	ไดอะแกรมแสดงโครงสร้างของ Gyroscope รูปร่างแบบ comb fingers	25
รูปที่ 2.22	โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน.....	27
รูปที่ 2.23	โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน กรณีไม่เขียนในส่วนของ namespace	28

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 2.24 ตัวอย่าง กรณีเขียนในส่วนของ namespace.....	29
รูปที่ 2.25 ตัวอย่าง กรณีไม่เขียนในส่วนของ namespace.....	29
รูปที่ 2.26 วงจรการเดิน (Gait Cycle).....	30
รูปที่ 2.27 การทำงานของข้อต่างๆ และกล้ามเนื้อในขณะเดิน.....	31
รูปที่ 3.1 ส่วนประกอบของอุปกรณ์ที่ออกแบบโดย โปรแกรม Autodesk Inventor 2017.....	34
รูปที่ 3.2 การขึ้นรูปชิ้นงานต้นแบบโดยใช้ เครื่องพิมพ์สามมิติ (3D Printer).....	35
รูปที่ 3.3 ส่วนประกอบของอุปกรณ์ต้นแบบ	35
รูปที่ 3.4 อุปกรณ์ต้นแบบ รุ่น 1.....	36
รูปที่ 3.5 อุปกรณ์ต้นแบบ รุ่น 2	36
รูปที่ 3.6 อุปกรณ์ต้นแบบ รุ่น 3	37
รูปที่ 3.7 อุปกรณ์ต้นแบบ รุ่น 4	37
รูปที่ 3.8 ผ้ารัดขาที่ใส่ติดตั้งอุปกรณ์.....	38
รูปที่ 3.9 Arduino Nano 3.0	38
รูปที่ 3.10 GY-521 IMU 3-axis Accelerometer/Gyro Module	39
รูปที่ 3.11 Lithium Battery (3.7V 400mAh Rechargeable Polymer Li-Po).....	39
รูปที่ 3.12 Step Up Booster Power (MT3608).....	40
รูปที่ 3.13 TP4056 1A Mini USB Battery Charger Board.....	40
รูปที่ 3.14 ชุด Power Supply	40
รูปที่ 3.15 Bluetooth module HC05.....	41
รูปที่ 3.16 แผ่น PCB.....	41
รูปที่ 3.18 การออกแบบ และเขียนโปรแกรมเพื่อสร้างระบบแสดงผล	43
รูปที่ 3.19 การออกแบบหน้าต่างโปรแกรมที่ติดต่อกับผู้ใช้	43
รูปที่ 3.20 การติดตั้งเซนเซอร์เข้ากับ goniometer	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.21 การเก็บผลจาก Serial Port.....	44
รูปที่ 4.1 การติดตั้งชุดอุปกรณ์เข้ากับขาของผู้ป่วย.....	45
รูปที่ 4.2 การรับค่ามุมมองเข้าผ่าน Arduino IDE.....	45
รูปที่ 4.4 หน้าต่างของโปรแกรมแสดงผล.....	46
รูปที่ 4.5 การบันทึกข้อมูลของโปรแกรม.....	46



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 4.1 แสดงค่ามุมมองเข้าจากอุปกรณ์จำกัดและวัดมุมมองของเข้า.....47



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

อาการบาดเจ็บที่หัวเข่าพบมากขึ้นในประเทศไทย ทั้งจากการเล่นกีฬา หรือจากการประสบอุบัติเหตุ ซึ่งการรักษาส่วนใหญ่จะเป็นการผ่าตัด ดังนั้นหลังการผ่าตัดผู้ป่วยต้องได้รับการดูแลอย่างใกล้ชิดเพื่อป้องกันผลแทรกซ้อนต่างๆ หรือการบาดเจ็บซ้ำซ้อน อีกทั้งยังต้องได้รับการกายภาพบำบัดเพื่อฟื้นฟูการทำงานของเข่า เพื่อให้กลับมาใช้งานได้อย่างปกติ และสามารถใช้ชีวิตประจำวันได้ปกติดังเดิม ซึ่งในปัจจุบันการกายภาพบำบัดนั้นจะมีการออกแบบได้เป็นแบบปฏิบัติที่ใช้กันทั่วไป ซึ่งสามารถใช้ได้กับผู้ป่วยทุกคน แต่จะทำให้ผลในการกายภาพบำบัดนั้นได้ไม่เต็มประสิทธิภาพ เนื่องจากการฟื้นฟูร่างกายของผู้ป่วยแต่ละคนนั้นไม่เหมือนกัน ทำให้ช่วงเวลาในการกายภาพบำบัดนั้นแต่ละคนต้องใช้ไม่เท่ากัน ตัวอย่างเช่น ผู้ป่วยบางคนอาจจะใช้เวลาฟื้นฟูในแต่ละขั้นของการบำบัดน้อย เนื่องจากร่างกายสามารถฟื้นฟูได้เร็ว ดังนั้น ผู้ป่วยสามารถกลับมาใช้ชีวิตประจำวันได้อย่างปกติแล้ว แต่หากต้องปฏิบัติตามแบบปฏิบัติที่ใช้กันทั่วไปแล้วจะทำให้ผู้ป่วยต้องเสียเวลาไปโดยเปล่าประโยชน์ นั่นคือเหตุผลที่ทำให้เกิดการศึกษาค้นคว้าและสร้างอุปกรณ์ในการศึกษาครั้งนี้ขึ้น เป็นอุปกรณ์ที่สามารถจำกัดมุมมองของผู้ป่วยได้ เพื่อป้องกันการเกิดอาการบาดเจ็บแทรกซ้อนและวัดองศาการงอของเข่าผู้ป่วยได้ โดยสามารถแสดงผลแบบตามเวลาจริงในขณะที่ผู้ป่วยกำลังเคลื่อนไหวหรือทำกิจกรรมได้ และนำข้อมูลนั้นมาแสดงผลเพื่อให้แพทย์สามารถปรับเปลี่ยนการกายภาพบำบัดให้เหมาะสมต่อผู้ป่วยเป็นเฉพาะรายบุคคลได้ เพื่อให้ผู้ป่วยกลับไปใช้ชีวิตได้อย่างปกติโดยเร็ว

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 สามารถออกแบบ และสร้างอุปกรณ์ที่สามารถจำกัดมุมมอง และวัดองศาการงอของเข่าผู้ป่วยได้

1.2.2 สามารถนำค่าองศาการงอของเข่าของผู้ป่วยมาแสดงบนจอแสดงผลได้ เพื่อให้แพทย์สามารถนำมาใช้ในการรักษา และกายภาพบำบัดของผู้ป่วยแบบเฉพาะบุคคลได้อย่างรวดเร็ว

1.3 สมมุติฐานของการศึกษา

อุปกรณ์สามารถวัด และแสดงผลมุมมองของเข่าได้ พร้อมทั้งจำกัดมุมมองของเข่าผู้ป่วยได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขอบเขตการวิจัย

อุปกรณ์ที่ออกแบบใช้สำหรับผู้ป่วยหลังผ่าตัดรักษาเข้า โดยสามารถวัดค่ามุมมองของเข้า และแสดงผลได้ โดยอุปกรณ์ออกแบบมาสำหรับการทำงานกับขาขวาของผู้ป่วยเท่านั้น และมีขนาดเส้นรอบวงของต้นขาและแข้งไม่เกิน 55 เซนติเมตร และ 38 เซนติเมตร ตามลำดับ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 สามารถจำกัดมุมมองของเข้าผู้ป่วย เพื่อป้องกันการอาการบาดเจ็บซ้ำซ้อนได้

1.5.2 ทำให้แพทย์สามารถปรับเปลี่ยนการรักษา และกายภาพบำบัดของผู้ป่วยที่มีอาการบาดเจ็บที่เข้า แบบเฉพาะบุคคลได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

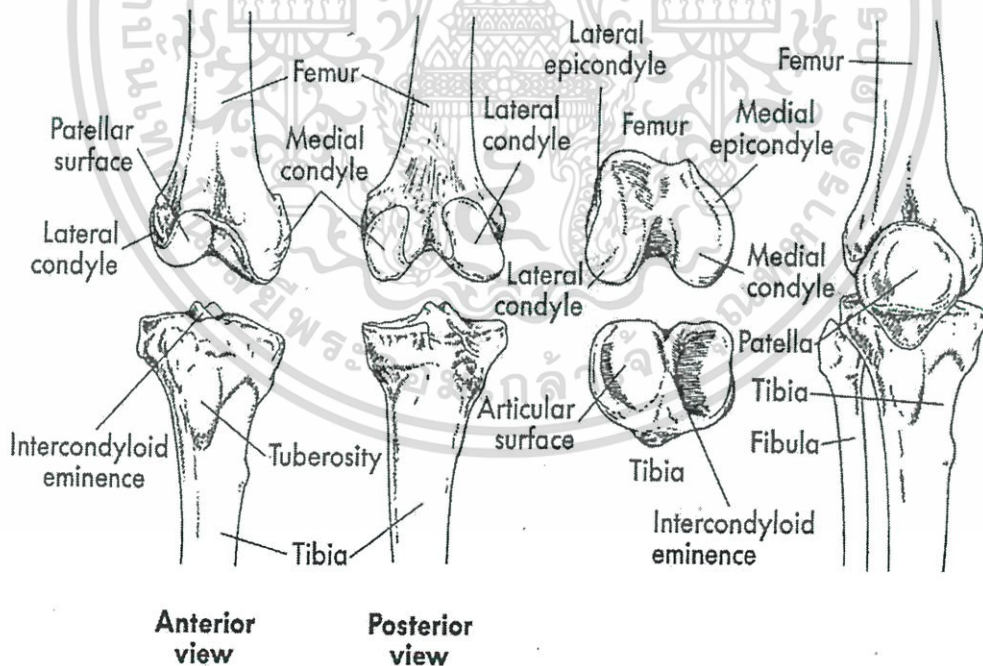
บทที่ 2

หลักการและทฤษฎี

2.1 การบาดเจ็บที่เข่า

2.1.1 กายวิภาคของเข่า

โครงสร้างของข้อเข่าเป็นส่วนที่แบกรับแรงหรือภาระได้มาก ตลอดจนเคลื่อนไหวตามกิจกรรมต่างๆ โดยข้อเข่าประกอบด้วยปลายกระดูก femur และกระดูก tibia มาต่อเชื่อมกันโดยมีกระดูกอ่อนอยู่ส่วนปลาย 2 ส่วนประกบกัน ได้แก่ Tibiofemoral joint ซึ่งเป็นข้อต่อแบบวงรี 2 วง และมีลักษณะคล้ายข้อต่อแบบเบ้า แต่จะมีการจำกัดการเคลื่อนไหวในทิศทางใดทิศทางหนึ่ง และอีกส่วนหนึ่งคือ patellofemoral joint ซึ่งเกิดจากกระดูกสะบ้า (patella) เชื่อมต่อกับผิวกระดูก femur และมีเยื่อหุ้มอยู่โดยรอบ (joint capsule) ซึ่งเป็นพื้นที่รอยต่อระหว่างกระดูกที่ทำให้เกิดการเคลื่อนไหวได้ ภายในข้อมีน้ำหล่อเลี้ยงข้อ (synovial fluid) บริเวณใดที่มีการหนาตัวของเยื่อหุ้มข้อก็จะช่วยเสริมความแข็งแรงของข้อบริเวณนั้น



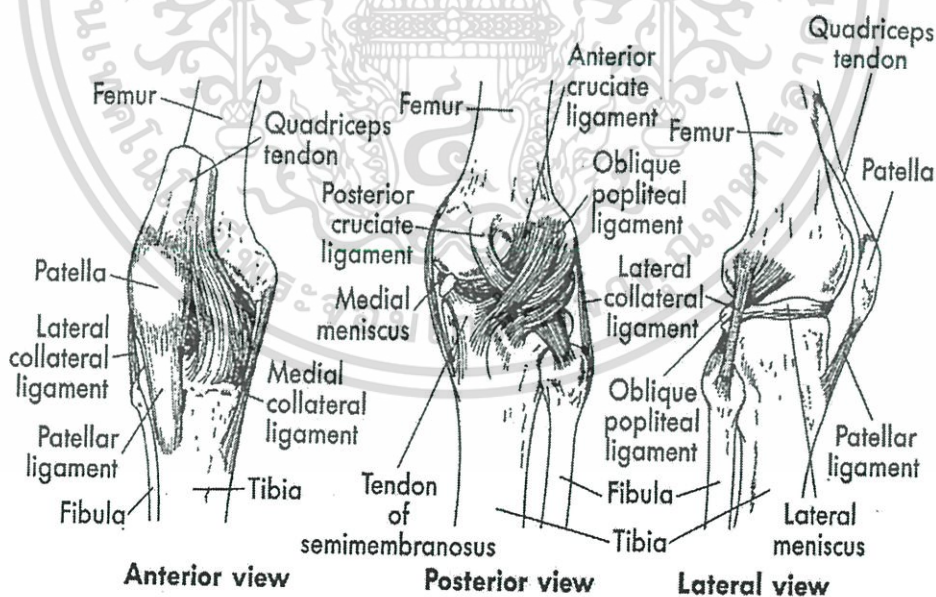
รูปที่ 2.1 กายวิภาคของเข่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tibiofemoral joint เป็นข้อต่อระหว่างกระดูก femur และกระดูก tibia ซึ่งเป็นข้อต่อแบบวงรี โดยมีการทำงานเหมือนบานพับ และมีการจำกัดการงอด้วยเส้นเอ็น ที่ปลายของกระดูก femur จะมี condyle 2 ชิ้น คือ medial femoral condyle และ lateral femoral condyle ซึ่งเป็นส่วนที่ช่วยในการงอของเข่า โดยเมื่อมีการเหยียดขาสุดโดยไม่มีกรงอเข่า (full extension) กระดูก tibia จะปิดออกด้านนอกเล็กน้อย เนื่องจาก medial femoral condyle มีความยาวมากกว่า lateral femoral condyle ซึ่งเรียกว่า screw-home mechanism

ในการเคลื่อนไหวของข้อเข่านั้น กระดูก femur และกระดูก tibia ไม่ได้เสียดสีกันโดยตรง แต่จะมีหมอนของกระดูก (menisci) ซึ่งมีรูปร่างเป็นพระจันทร์เสี้ยว ติดอยู่กับด้านบนของกระดูก tibia โดยมีส่วนช่วยการรับแรงของข้อเข่า

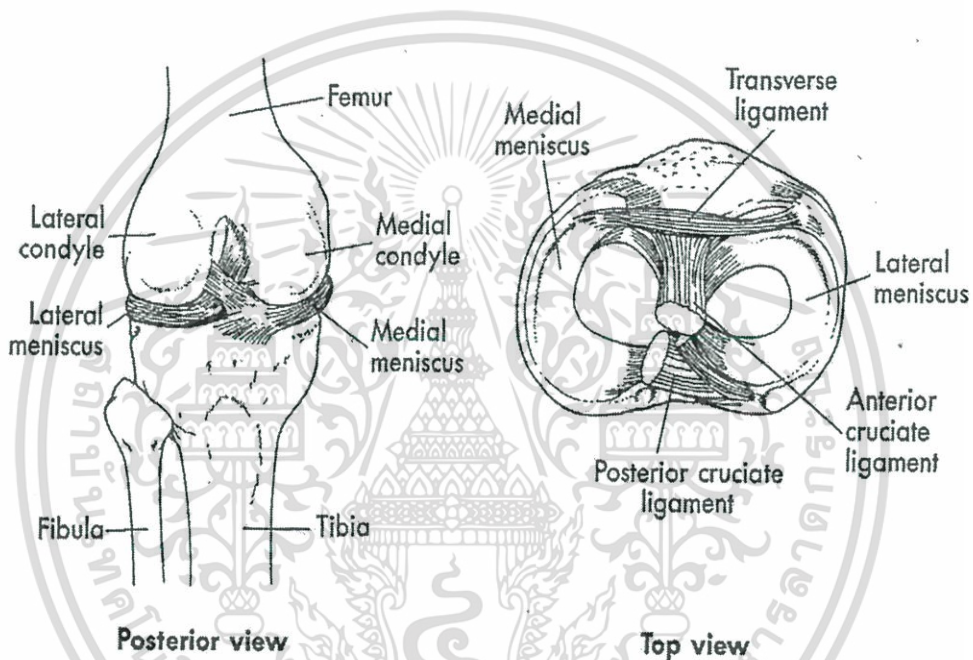
นอกจากนี้ ข้อเข่ายังประกอบด้วยเส้นเอ็นหลายชิ้นจะช่วยในการเสริมความแข็งแรงของข้อเข่า อีกทั้งเอ็นแต่ละตำแหน่งจะเป็นตัวช่วยกำหนด หรือจำกัดทิศทางของการเคลื่อนที่ของเข่าในทิศทางต่างกันไป เอ็นที่อยู่นอกข้อ (extra-articular ligament) ซึ่งประกอบด้วย medial และ lateral collateral ligament หรืออาจเรียกว่า tibia และ fibular collateral ligament ซึ่งเรียกตามกระดูกที่เส้นเอ็นนี้ไปยึดติด โดย medial collateral ligament จะยึดติดกับกระดูก femur และกระดูก tibia ด้านในของเข่า และ lateral collateral ligament จะยึดติดกับกระดูก femur และกระดูก fibular ช่วยเสริมความแข็งแรงทั้งด้านในและด้านนอกข้อเข่า



รูปที่ 2.2 เส้นเอ็นต่างๆ ของข้อเข่า (ligament)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

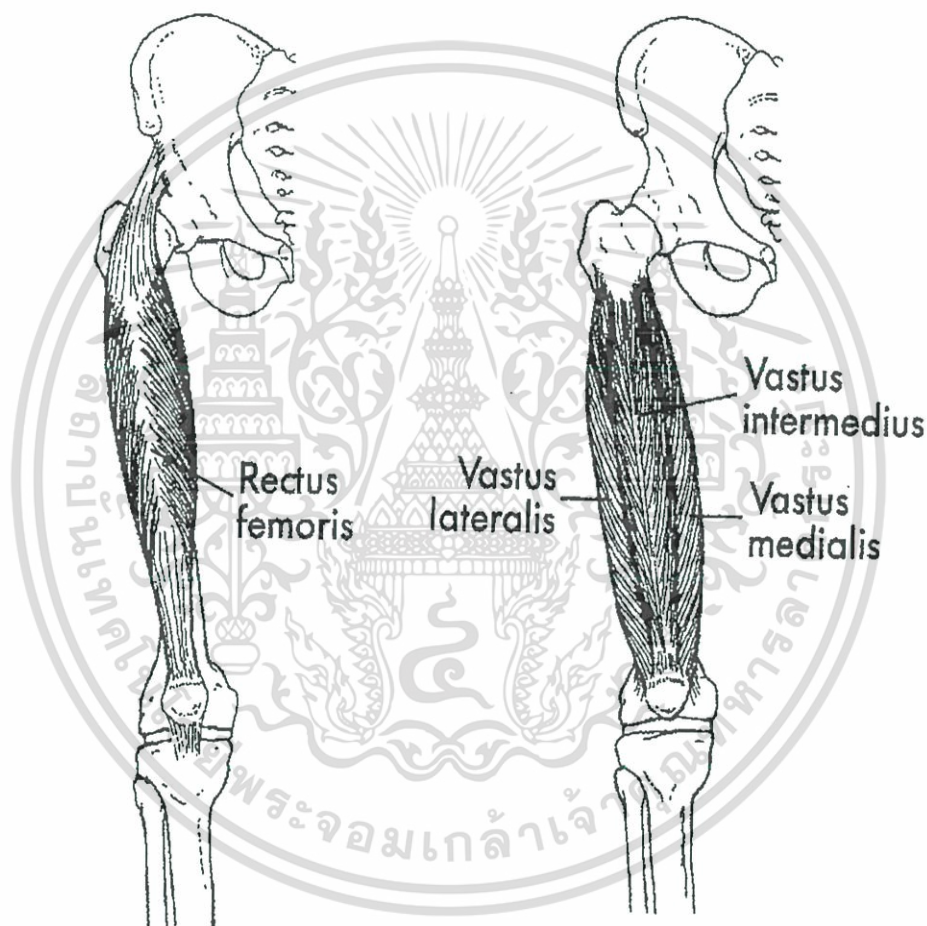
เอ็นในข้อ (intra-articular ligament) อันได้แก่ เอ็นไขว้หน้า Anterior cruciate ligament (ACL) และ เอ็นไขว้หลัง Posterior cruciate ligament (PCL) โดยเอ็นทั้งสองนี้จะช่วยจำกัดการเคลื่อนที่แบบสไลด์ไปด้านหน้า และหลังของกระดูก femur บนกระดูก tibia ในขณะที่มีการงอและเหยียดเข่า (flexion และ extension) อีกทั้งเป็นตัวจำกัดการเหยียดของเข่าแบบ hyperextension ซึ่งข้อของเส้นเอ็นนั้นตั้งตามจุดที่ยึดติดกับกระดูก tibia ซึ่งทั้งหมดที่กล่าวมานี้ล้วนแต่เป็นโครงสร้างเสริมความแข็งแรงให้กับข้อทั้งสิ้น (static stabilization) นอกจากนี้เอ็นส่วนปลายกล้ามเนื้อ (tendon) ที่ทอดข้ามข้อเข่า จะช่วยเสริมความแข็งแรงของข้อในขณะที่เคลื่อนไหวด้วยเช่นกัน (dynamic stabilization)



รูปที่ 2.3 เส้นเอ็นของข้อเข่า และหมอนรองกระดูก (menisci)

patellofemoral joint เป็นข้อต่อที่ประกอบด้วยกระดูกสะบ้า (patella) ที่มีรูปร่างคล้ายสามเหลี่ยม ซึ่งถูกห่อหุ้มด้วยเส้นเอ็น และ trochlear groove โดย patellofemoral joint จะอยู่ตรงตำแหน่งระหว่าง medial femoral condyle และ lateral femoral condyle โดยด้านหลังของกระดูกสะบ้า (patella) จะถูกครอบด้วยกระดูกอ่อนเพื่อลดแรงเสียดทานระหว่างกระดูกสะบ้า (patella) กับกระดูก femur ซึ่งในขณะที่กำลังก้าวขึ้นบันไดนั้น patella ต้องรับแรงมากถึง 3 เท่าของน้ำหนักตัว อย่างไรก็ตามการทำงานของ patella ยังช่วยการได้เปรียบเชิงกลในขณะที่เหยียดเข่าออก โดยช่วยลดแรงดึงที่เกิดจากกล้ามเนื้อ Quadriceps ได้อย่างมาก โดยทั่วไป patella ประกอบด้วย articular facet 2 ข้าง คือ medial และ lateral facet ซึ่งแต่ละองศาของการงอหรือเหยียดข้อเข่านั้นกระดูกสะบ้า (patella) จะประกบกับกระดูก femur อยู่น้อยต่างกันไปในท่าเหยียดเข่าสุดกระดูกสะบ้า (patella) จะประกบกับ Intercondylar trochlea fossa ทั้งทาง medial เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้เห็นใบเซอร์เวชันนี้ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ lateral facet ซึ่งเป็นการช่วยลด force per surface area ลงไป เพราะในขณะ flexion นั้น กระดูกสะบ้า (patella) ต้องรับแรงที่กดต่อผิวของ femoral สูงมาก ทางกลศาสตร์แนวแรงของ กล้ามเนื้อ Quadriceps ที่ผ่านสะบ้าจะทำมุมกับแนวแรงดึงของ patellar tendon ขนาดหนึ่ง เรียก Q-angle แต่แรงต่างๆ นี้จะสมดุลย์กับแรงดึงจาก Vastus medialis และ Vastus lateralis รวมทั้ง patello-femoral ligaments และ retinaculum ทั้งทาง lateral และ medial โดยทั่วไปมุม Q-angle นี้มีค่าปกติไม่ควรเกิน 10 องศา หากมุมมีค่าเกินค่าดังกล่าวอาจเป็นปัญหาทำให้กระดูกสะบ้า (patella) หลุดหรือเคลื่อนที่จาก trochlear groove ของ distal femur ขณะงอข้อเข้าได้ง่าย



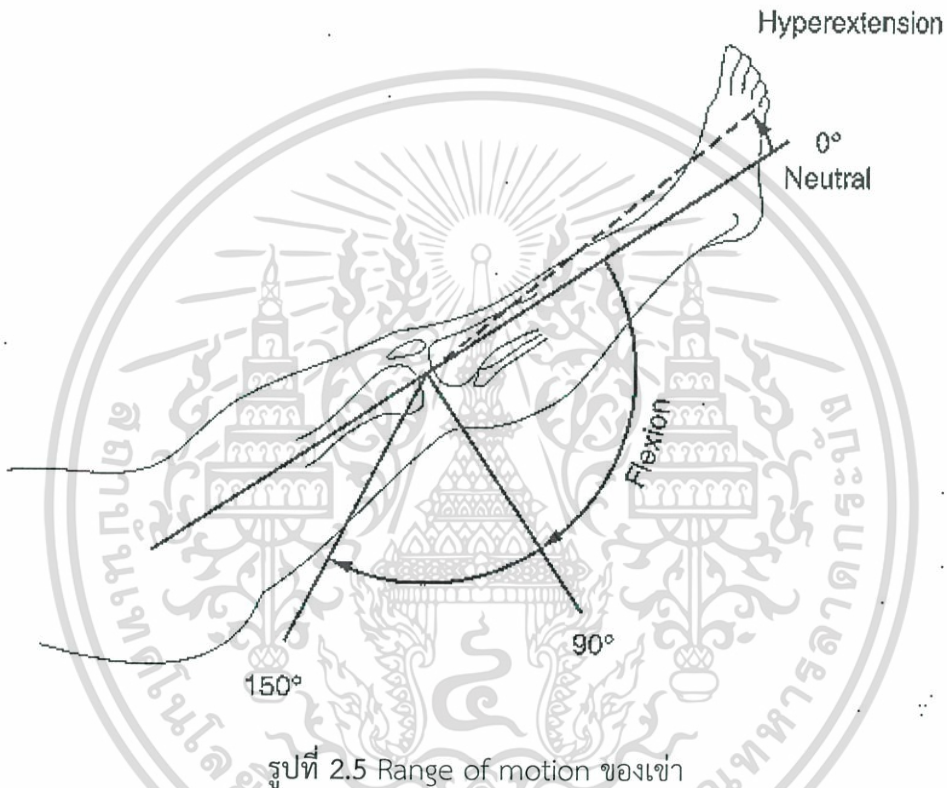
Anterior view

รูปที่ 2.4 มัดกล้ามเนื้อบริเวณต้นขา (กล้ามเนื้อ Quadriceps)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 Range of motion

โดยทั่วไปการเคลื่อนไหวของข้อเข่าควรเหยียดได้ตรง (ในคนทั่วไปบางคนอาจมีเข่าแอ่น (genu recurvatum) ได้เล็กน้อย หรือ hyperextension) ซึ่งโดยปกติมุมการเคลื่อนไหวจากท่าเข่าเหยียดตรงควรเริ่มจาก 0 องศา และเมื่องอข้อเข่าได้จนสุดของข้อพับ ควรได้มุมตั้งแต่ 120 องศาเป็นต้นไปจนถึงมุม 155 ทั้งนี้ขึ้นอยู่กับ ความหนาหรือความตึงตัวของเนื้อเยื่อใต้ข้อพับของแต่ละบุคคล การเคลื่อนไหวที่ได้น้อยกว่าปกติ โดยเฉพาะร่วมกับอาการปวด น่าจะบ่งบอกว่าเป็นปัญหาของข้อเข่า



2.1.3 อาการบาดเจ็บ

อาการบาดเจ็บของข้อเข่าเป็นสิ่งที่พบได้บ่อยในชีวิตประจำวัน ซึ่งอาจจะเกิดจากอุบัติเหตุอย่างรุนแรง เช่น อุบัติเหตุรถยนต์ หรือตกจากที่สูง ผู้ป่วยมักได้รับบาดเจ็บทางระบบกระดูกและกล้ามเนื้อเป็นเรื่องของกระดูกหัก เลือดออกภายในข้อ กล้ามเนื้อระบมหรือฉีกขาด เอ็นฉีกขาดและข้อสูญเสียความมั่นคง โดยเอ็นที่มีการฉีกขาดบ่อย คือ เอ็นเอ็นไขว้หน้าข้อเข่า (anterior cruciate ligament, ACL) ซึ่งผลก็คือมีการสูญเสียการทำงานของข้อ และมีผลกระทบต่อการใช้งานข้อเข่าอย่างมีประสิทธิภาพ

นอกจากนี้การบาดเจ็บของข้อเข่ายังอาจพบได้บ่อยในหมู่นักกีฬาที่มีการปะทะกันรุนแรงหรือการเคลื่อนไหวผิดจังหวะจึงทำให้บาดเจ็บ เช่น ฟุตบอล บาสเก็ตบอล วอลเลย์บอลล์ เทนนิส ซึ่งพบว่าก่อให้เกิดการบาดเจ็บต่อข้อเข่าได้ กลไกของการบาดเจ็บอาจมีได้หลายลักษณะ เช่น การเหยียดเอกสทรันเป็นเอกสทรันทลงวนเวลาหรือการเข้านเพื่อการศึกษาเท่านั้น เมื่อผู้เฒ่าเห็นไปเซบระยะขึ้นต้นการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าหรือการงอเข้าอย่างรุนแรง หรือมีแรงกระทำทางด้านนอกของข้อเข่า แต่กลไกการบาดเจ็บของข้อเข่าที่พบได้บ่อยมากที่สุดในการเล่นกีฬา คือ การบิดหมุนของข้อเข่า และมักจะทำให้มีการเสียหายของโครงสร้างที่สำคัญของข้อเข่า เช่น เส้นเอ็น ซึ่งนำไปสู่อาการบาดเจ็บของข้อเข่า

โดยพื้นฐานข้อมีหน้าที่และการทำงาน คือ เป็นส่วนที่เคลื่อนไหวของระบบโครงร่างของร่างกาย ปัจจัยที่จะทำให้ข้อทำงานได้อย่างสมบูรณ์ ได้แก่

2.1.3.1 ความมั่นคงของข้อ (stability)

2.1.3.2 ข้อสามารถเคลื่อนไหวในทิศทางต่างๆ ได้ (mobility)

2.1.3.3 ไม่มีความเจ็บปวดมารบกวนการทำงานของข้อ (painless)

ความมั่นคงของข้อเข่า (Stability of the knee joint) ปัจจัยที่มีส่วนเกี่ยวข้องกับความมั่นคงข้อเข่า ได้แก่

1. ลักษณะและรูปร่างของกระดูก (configuration) ตัวอย่างข้อที่ความมั่นคงแข็งแรงมาจากโครงสร้าง หรือรูปร่างของข้อนั้นๆ ได้แก่ ข้อสะโพก เนื่องจากข้อสะโพกมีรูปร่างกลมและข้อฝังอยู่ในเบ้าลึก (ball and socket) ความแข็งแรงของข้อสะโพกจึงได้มาจากความได้เปรียบในเรื่องโครงสร้างเป็นส่วนใหญ่ ในขณะที่ข้อเข่าที่มีรูปร่าง femoral side เป็น condyle และทาง tibia เป็นรูปแบน (plateau) นั้น จึงทำให้ข้อเข่าได้รับความแข็งแรงจากส่วนนี้ค่อนข้างน้อยกว่าเมื่อเทียบกับข้อสะโพก

2. เยื่อหุ้มข้อที่อยู่โดยรอบ รวมทั้งบริเวณของเยื่อหุ้มข้อที่หนาตัวขึ้น จะช่วยเสริมความแข็งแรงของข้อ

3. กระดูกอ่อน เช่น meniscus หรือ labrum จะเป็นส่วนที่ช่วยเสริมให้พื้นที่ผิวสัมผัสของข้อระหว่างกระดูกแต่ละชิ้นให้เพิ่มมากขึ้น ช่วยให้รูปร่างของข้อสอดคล้องเข้ากันได้มากขึ้น และทำให้ความลึกของข้อมากขึ้น

4. เอ็นที่อยู่ทั้งในและนอกข้อ (cruciate และ collateral ligaments) เป็นตัวช่วยยึดให้ข้อมีความมั่นคง และในข้อเข่าที่ด้านโครงสร้างนั้นมีความเสถียรเปรียบเทียบในเรื่องความมั่นคงอยู่แล้ว เอ็นที่อยู่ทั้งในและนอกข้อยังมีความสำคัญมากขึ้น

5. กล้ามเนื้อที่ทอดผ่านข้อ กล้ามเนื้อเหล่านี้จะมีความสำคัญในแง่ของการเสริมสร้างความแข็งแรงให้แก่ข้อ ซึ่งการฝึกออกกำลังกายจะช่วยให้กล้ามเนื้อเหล่านี้มีความแข็งแรงมากขึ้น และมีผลช่วยให้ความตึงของเอ็นเพิ่มขึ้น

6. Hydrostatic pressure ซึ่งส่วนใหญ่ข้อต่อของร่างกายเรามีความดันติดลบอยู่แล้ว (negative pressure) ทำให้ข้อต่อมีความมั่นคงมากขึ้น

2.1.4 วิธีการตรวจ และการรักษา

ผู้ป่วยที่มีเส้นเอ็นไขว้หน้าข้อเข่าฉีกขาดจำเป็นต้องจำกัดกิจกรรมและทำการรักษา หากปล่อยทิ้งไว้ อาจทำให้ข้อเข่าแย่งเรื่อยๆ และทำให้ผู้ป่วยมีความเสี่ยงที่จะได้รับบาดเจ็บที่หัวเข่ามากขึ้น นอกจากนี้อาการกระดูกอ่อนเสื่อมก็อาจทำให้เกิดข้อเข่าเสื่อมได้ในที่สุด อย่างไรก็ตาม ผู้ป่วยไม่จำเป็นต้องเข้ารับการผ่าตัดเสมอไป การรักษาขั้นแรกจะใช้หลักการ RICE ซึ่งก็คือ Rest = การพัก Ice = น้ำแข็ง Compression = การประคบ และ Elevation = การยกสูง ร่วมกับการใช้ยา ทั้งนี้ผู้ป่วยสามารถใช้วิธีการบริหารเพื่อฝึกให้ข้อต่อ และกล้ามเนื้อสามารถเคลื่อนไหวได้เหมือนเดิม และออกกำลังกายด้วยวิธีการเพิ่มการเคลื่อนไหว (range-of-motion) ให้เร็วที่สุด โดยตั้งเป้าหมายในการกลับมาใช้เข่าได้เหมือนเดิม เมื่ออาการทุเลาลงและมีความแข็งแรงเพิ่มขึ้นแล้ว แพทย์จึงจะทำการตรวจวิเคราะห์เข่าอีกครั้ง เพื่อประเมินว่าผู้ป่วยสามารถทำกิจกรรมได้ในระดับใด หากผู้ป่วยยังไม่สามารถใช้งานเข่าได้เหมือนเดิม ก็จำเป็นต้องลดกิจกรรมลง ถ้ายังไม่สามารถแก้ไขได้ อาจจำเป็นต้องได้รับการผ่าตัดต่อไป

การผ่าตัดจัดเป็นวิธีการรักษาที่ใช้บ่อยในกรณีเส้นเอ็นไขว้หน้าข้อเข่าฉีกขาด มีเป้าหมายหลักเพื่อป้องกันไม่ให้เกิดกระดูกหน้าแข้งเคลื่อนที่ไปข้างหน้ามากเกินไป เพราะจะส่งผลเสียต่อกระดูกอ่อน และเพื่อให้หัวเข่ากลับมาทำงานได้ตามปกติ โดยวิธีการรักษา แพทย์จะนำเส้นเอ็นส่วนที่ขาดออกไป และนำเส้นใหม่มาใส่แทนที่ ซึ่งในช่วง 10 ปีที่ผ่านมา เทคนิคการผ่าตัดเส้นเอ็นไขว้หน้าข้อเข่ามีการเปลี่ยนแปลงอย่างมาก โดยมีการปรับปรุงเทคนิคและทำให้ผู้ป่วยฟื้นตัวได้ง่ายขึ้น ทั้งนี้ผู้ป่วยหลายรายอาจมีความสงสัยว่าทำไมจึงไม่สามารถเย็บเส้นเอ็นที่ขาดให้ติดเข้าด้วยกัน ความคิดเช่นนี้ได้มีการทดลองมาแล้วในอดีต ซึ่งได้ผลที่แตกต่างเป็นอย่างมากหรืออาจเป็นไปได้เลยที่จะทำให้เส้นเอ็นสมานตัวเองจนถึงระดับที่แข็งแรงพอจนทำงานตามปกติได้ อย่างไรก็ตามมีบางกรณีที่เส้นเอ็นไขว้หน้าข้อเข่าฉีกขาดแล้วเกิดสมานตัวเองโดยไม่ต้องเปลี่ยน แต่เป็นเหตุการณ์ที่ไม่เกิดขึ้นบ่อย และจะเกิดขึ้นต่อเมื่อเส้นเอ็นมีการฉีกขาดเพียงเล็กน้อยเท่านั้น

ทั้งนี้มีการศึกษาที่ทำการเปรียบเทียบการรักษาแบบไม่ผ่าตัด และการรักษาโดยการผ่าตัด พบว่า ผู้ป่วยร้อยละ 94 ที่เข้ารับการผ่าตัดเส้นเอ็นไขว้หน้าข้อเข่าจะสามารถใช้ข้อเข่าได้ตามปกติต่อไปเป็นเวลา 15-20 ปี และมีเปอร์เซ็นต์การเกิดข้อเข่าเสื่อมน้อยลงอย่างมีนัยสำคัญเมื่อเทียบกับผู้ป่วยที่รักษาด้วยวิธีดั้งเดิม

2.1.5 การกายภาพบำบัด และการฟื้นฟูผู้ป่วยหลังการรักษาเข่าด้วยการผ่าตัด

หลังผ่าตัดเข้าผู้ป่วยจะได้รับการพันเข่าด้วยสำลีหนาๆ แล้วรัดด้วยผ้ายึดให้แน่นเพื่อไม่ให้บวม ประมาณ 3 – 4 วัน หลังจากนั้นให้ผู้ป่วยเริ่มบริหารกล้ามเนื้อหัวเข่าได้ การบริหารกล้ามเนื้อ การทำกายภาพบำบัดหลังผ่าตัด

1. หลังจากแผลหายดี ผู้ป่วยเริ่มบริหารกล้ามเนื้อเหยียดหัวเข่าได้
2. เริ่มบริหารโดยไม่รับน้ำหนักที่เข่า เช่น การว่ายน้ำ, การขี่จักรยาน
3. การบริหารกล้ามเนื้อโดยใช้น้ำหนัก (Weight training)

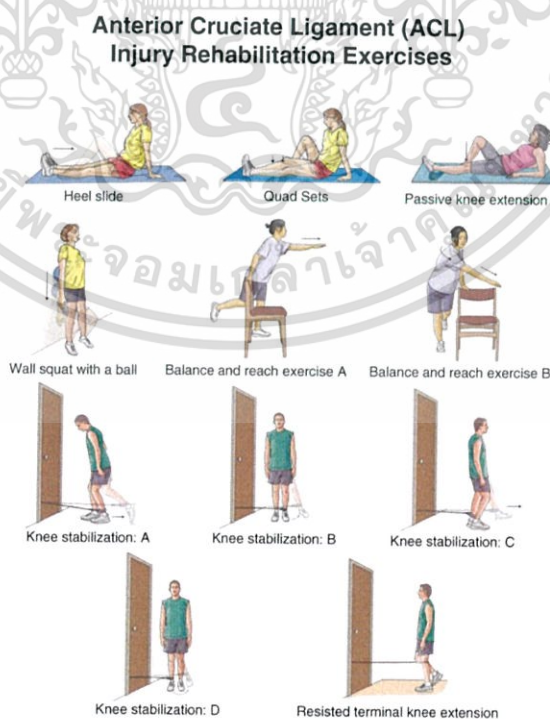
2.1.5.1 การบริหารยืดเหยียดกล้ามเนื้อ

การยืดเหยียดกล้ามเนื้อ (Quadriceps) ยืนเกาะกำแพงหรือโต๊ะ งอเข่าดึงเท้าไปด้านหลังให้รู้สึกตึงต้นขาด้านหน้าให้มากที่สุดนาน 5 – 7 วินาที ทำซ้ำ 6 – 10 ครั้งทำทั้ง 2 ข้าง

การยืดเหยียดกล้ามเนื้อ (Hamstrings) ฝึกงอเข่าข้างหนึ่ง เหยียดขาอีกข้างหนึ่ง โน้นตัวไปข้างหลัง และข้างหน้ารู้สึกตึงด้านหลังต้นขาข้างที่เหยียด นาน 5-7 วินาที ทำซ้ำ 6-10 ครั้งทำทั้ง 2 ข้าง

2.1.5.2 การบริหารให้กล้ามเนื้อแข็งแรง

กล้ามเนื้อ Quadriceps นั่งเก้าอี้สูง งอเข่า ห้อยเท้า เหยียดเข่า ต้านกับแรงต้าน อาจจะใช้ถุงทราย ยางยืด



รูปที่ 2.6 การบริหารกล้ามเนื้อบริเวณเข่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5.3 โปรแกรมกายภาพบำบัด (Recovery After ACL Reconstruction)

ระยะที่ 1 (สัปดาห์ที่ 1-2)

ปรับส้นเข่าให้งอได้ 0-60 องศา เดินลงน้ำหนักบางส่วนด้วยไม้ค้ำยันในท่าเหยียดเข่าตรง สัปดาห์ที่ 2 ควรจะสามารถนอนหงายเหยียดเข่าสุดแล้วกระดกข้อเท้าเข้าหาตัวได้ซึ่งจะทำให้สามารถเหยียดเข่าได้สุด บริหารโดยการกระดกข้อเท้าขึ้นลง วันแรกๆ หลังจากนั้นเมื่อปวดเข่าลดลงให้นอนหงาย และค่อยๆงอเข่าในลักษณะที่ส้นเท้าสัมผัสกับพื้นหลังจากนั้นเกร็งขึ้นค้างไว้ประมาณ 10 วินาทีปล่อยขาลง ให้ทำบ่อยๆเท่าที่จะทำได้

ระยะที่ 2 (สัปดาห์ที่ 3-4)

ปรับส้นเข่ามาที่ 90 องศา เดินลงน้ำหนักได้บางส่วนบริหารเข่าเหมือนระยะที่ 1 แต่เพิ่มจำนวนรอบให้มากขึ้นหลังสัปดาห์ที่ 4 ฝึกขี่จักรยานอยู่กับที่ บริหารกล้ามเนื้อโดยการยื่นหลังชิดกำแพงย่อตัวขึ้นลงให้เข่าเคลื่อนที่ 0-45 องศา ปลายสัปดาห์ที่ 4 เดินลงน้ำหนักได้เต็มที่โดยไม่ต้องใช้ไม้ค้ำยันแต่ต้องเดินในท่าเข่าเหยียดตรง

ระยะที่ 3 (สัปดาห์ที่ 5-6)

ผู้ป่วยที่ใส่ส้นเข่าควรจะสามารถเหยียดเข่าได้สุด และงอเข่าได้เต็มที่ แพทย์อาจให้หยุดใช้ส้นเข่าได้

ระยะที่ 4 (สัปดาห์ที่ 7-12)

ฝึกบริหารมากขึ้นโดยปั่นจักรยานอยู่กับที่ปรับระยะ ผืนให้มากขึ้น ยืนเขย่ง ปลายเท้ายืนลงน้ำหนักเหยียดเข่าสุดแล้วกระดกข้อเท้าขึ้นฝึกขึ้นลงบันได ยืนลงน้ำหนักข้างที่ผ่าย่อตัวลงว้ายน้ำท่ากบ วิ่งในน้ำได้

ระยะที่ 5 (เดือนที่ 4-6)

ผู้ป่วยควรงอเหยียดเข่าได้สุด และปราศจากความเจ็บปวดบริหารโดยการเดิน และวิ่งเหยาะๆ เริ่มวิ่งไปด้านหน้า และถอยหลัง ค่อยๆเพิ่มความเร็วตามลำดับ

ระยะที่ 6 (เดือนที่ 6 เป็นต้นไป)

สามารถเล่นกีฬาตามปกติได้ประมาณเดือนที่ 6 โดยกล้ามเนื้อต้นขาควรได้ประมาณ 80-90 % ของข้างปกติ

2.2 ไมโครคอนโทรลเลอร์ (Microcontroller)

เป็นระบบที่ประกอบด้วยหน่วยความจำทั้งแบบชั่วคราว(RAM) และแบบถาวร (ROM) หน่วยประมวลผล และพอร์ตอินพุต เอาท์พุตภายในตัว สามารถเขียนโปรแกรมเพื่อควบคุม ระบบอื่นๆตามต้องการได้ จัดเป็นระบบฝังตัวชนิดหนึ่ง

2.2.1 ประเภทของไมโครคอนโทรลเลอร์

มีเกณฑ์การจำแนกหลายชนิดเช่น จำนวนอินพุต และเอาท์พุต ขนาดของหน่วยความจำชั่วคราว แต่ส่วนมากจะแบ่งตามสถาปัตยกรรม ตัวอย่างที่พบได้ส่วนมาก เช่น

1. ไมโครคอนโทรลเลอร์ตระกูล PIC (บริษัทผู้ผลิต Microchip ไมโครชิป)
2. ไมโครคอนโทรลเลอร์ตระกูล MCS51 (บริษัทผู้ผลิต Atmel,Phillips)
3. ไมโครคอนโทรลเลอร์ตระกูล AVR (บริษัทผู้ผลิต Atmel)
4. ไมโครคอนโทรลเลอร์ตระกูล ARM7,ARM9 (บริษัทผู้ผลิต Atmel, Phillips, Analog Device, Sumsung, STMicroelectronics)
5. ไมโครคอนโทรลเลอร์ตระกูล Basic Stamp (บริษัทผู้ผลิต Parallax)
6. ไมโครคอนโทรลเลอร์ตระกูล PSOC (บริษัทผู้ผลิต CYPRESS)
7. ไมโครคอนโทรลเลอร์ตระกูล MSP (บริษัทผู้ผลิต Texas Instruments)
8. ไมโครคอนโทรลเลอร์ตระกูล 68HC (บริษัทผู้ผลิต MOTOROLA)
9. ไมโครคอนโทรลเลอร์ตระกูล H8 (บริษัทผู้ผลิต Renesas)
10. ไมโครคอนโทรลเลอร์ตระกูล RABBIT (บริษัทผู้ผลิต RABBIT SEMICONDUCTOR)
11. ไมโครคอนโทรลเลอร์ตระกูล Z80 (บริษัทผู้ผลิต Zilog)

2.2.2 ภาษาที่ใช้ในการเขียนไมโครคอนโทรลเลอร์

ภาษาที่ใช้ในเบื้องต้น ไมโครคอนโทรลเลอร์บางตระกูล จะใช้ได้ครบทุกภาษา แต่บางตระกูลจะใช้ได้บางภาษา ขึ้นอยู่กับบริษัทผู้ผลิต Software (โดยทั่วไปจะเรียกว่า Editor And Compiler) ที่ใช้เขียนภาษาไมโครคอนโทรลเลอร์นั้น ได้แก่

- 1.ภาษา Assembly
- 2.ภาษา Basic
- 3.ภาษา C
- 4.ภาษา Pascal

2.2.3 การประยุกต์ใช้ไมโครคอนโทรลเลอร์

นำไปใช้ฝังในระบบของอุปกรณ์อื่น ๆ เพื่อใช้ควบคุมการทำงานบางอย่าง เช่น ใช้ในรถยนต์, เต้าอบไมโครเวฟ, เครื่องปรับอากาศ, เครื่องซักผ้าอัตโนมัติ เป็นต้น เพราะว่าไมโครคอนโทรลเลอร์มีข้อดีเหมาะสมต่อการใช้งานควบคุมหลายประการ เช่น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชิพไอซีและระบบที่ได้มีขนาดเล็ก
- ระบบที่ได้มีราคาถูกกว่าการใช้ชิพไมโครโพรเซสเซอร์
- วงจรที่ได้จะมีความซับซ้อนน้อย ช่วยลดข้อผิดพลาดที่อาจจะเกิดขึ้นได้ในการต่อวงจร
- มีคุณสมบัติเพิ่มเติมสำหรับงานควบคุมโดยเฉพาะซึ่งใช้งานได้ง่าย
- ช่วยลดระยะเวลาในการพัฒนาระบบได้

2.2.4 ไมโครคอนโทรลเลอร์ Arduino

เป็นไมโครคอนโทรลเลอร์ตระกูล AVR ขนาดเล็กเป็นตัวประมวลผล และสั่งงาน ซึ่งมีการพัฒนาแบบ open source มีจุดเด่นในเรื่องของความง่ายในการใช้งานเนื่องจากการออกแบบคำสั่งต่างๆ ขึ้นมาสนับสนุนการใช้งาน ด้วยรูปแบบที่ไม่ซับซ้อน รวมถึงมีการพัฒนาแบบ open source คือการเปิดเผยข้อมูลด้าน Hardware และ Software ทำให้มีการนำไปพัฒนาและประยุกต์ใช้งานกันอย่างแพร่หลาย นอกจากนี้ยังมีราคาไม่แพง มี source code สำเร็จจำนวนมาก ซึ่งเหมาะสมสำหรับผู้เริ่มต้น

ผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด ทั้งแบบดิจิทัล และแบบอนาล็อกเพื่อนำไปรับค่าจากอุปกรณ์ตรวจจับ (sensor) แบบต่างๆ รวมไปถึงการควบคุมอุปกรณ์เอาต์พุตต่างๆ เช่น หลอดไฟ มอเตอร์ เป็นต้น หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น

2.2.4.1 รุ่นต่างๆของ Arduino

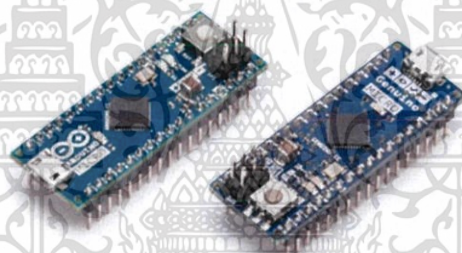
ปัจจุบันมีผลิออกมามากกว่า 20 รุ่น โดยมีความแตกต่างกันทั้งทางด้านขนาด จำนวนพิน ชนิดของหน่วยประมวลผล รวมถึงคุณสมบัติพิเศษที่เพิ่มเข้ามาเพื่อการทำงาน โดยเฉพาะ ซึ่งเพิ่มความสะดวกในการใช้งานยิ่งขึ้น เช่น

Arduino UNO เป็นบอร์ดพื้นฐานที่ใช้งานง่าย สามารถนำไปประยุกต์ใช้กับงานต่างๆได้ เหมาะสำหรับผู้เริ่มต้น



รูปที่ 2.7 Arduino UNO

Arduino Micro เป็นบอร์ดขนาดเล็ก มีจำนวนพินที่น้อยกว่ารุ่นอื่นๆ เหมาะสำหรับใช้ในงาน built-in ภายในอุปกรณ์ที่มีขนาดเล็ก เช่น เม้าส์ คีย์บอร์ด เป็นต้น

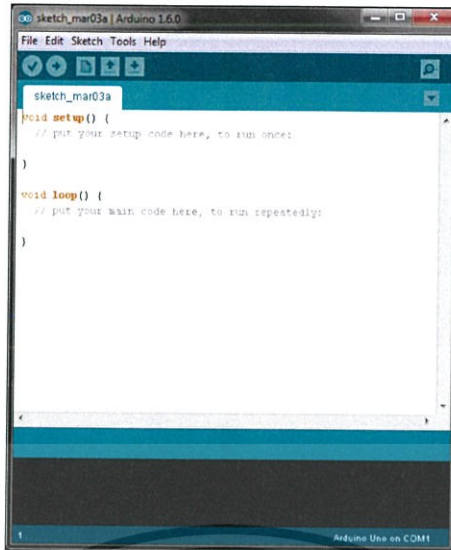


รูปที่ 2.8 Arduino Micro

2.2.4.2 การโปรแกรม Arduino

โปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม ซึ่งเป็นโปรแกรมแบบ open-source ทำให้การเขียน และการอัปโหลดโปรแกรมเป็นไปอย่างง่ายดาย Arduino IDE นั้นสามารถใช้งานได้บนระบบปฏิบัติการทั้ง Windows Mac OS X และ Linux ภาษาที่ใช้ในการเขียนโปรแกรมคือ C++ ที่มีโครงสร้างของตัวภาษาโดยรวมใกล้เคียงกับภาษาซีมาตรฐาน (ANSI-C) เพียงแต่มีการปรับปรุงรูปแบบในการเขียนโปรแกรมบางส่วนให้ต่างไปจากเดิมเล็กน้อย เพื่อลดความยุ่งยากในการเขียนโปรแกรม และเพิ่มความง่าย สะดวกกว่าการเขียนแบบมาตรฐานโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 Arduino IDE

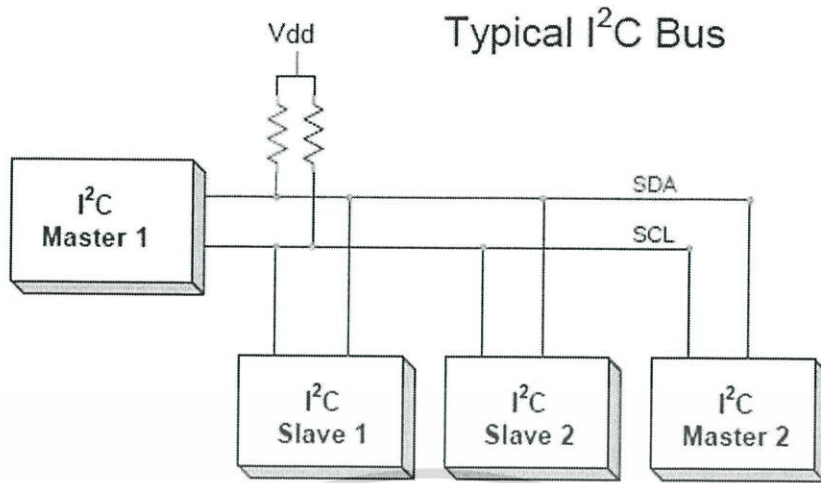
2.3 Inertial measurement unit (IMU)

Inertial Measurement Unit (IMU) เซนเซอร์หรืออุปกรณ์ที่ใช้ใน INS (Inertial Navigation System) ถูกเรียกว่า Inertial Measurement Units (IMU) ซึ่งเป็นส่วนประกอบหลักของ INSs ที่ใช้ในเครื่องบิน, ยานอวกาศ และเรือ เช่นเดียวกับจรวดขีปนาวุธ IMU ประกอบด้วย 2 ส่วนหลักคือ accelerometers 3 ทิศทาง และ gyroscopes 3 ทิศทาง ซึ่งรับค่าความเร่งยานพาหนะและความเร็วเชิงมุมตามลำดับ โปรดทราบว่า mechanic gyroscope และ accelerometer ถูกนำมากล่าว ณ ที่นี้เพียงเพื่อแสดงหลักการพื้นฐานของ IMU สำหรับ MEMS นั้น จะไม่กล่าวถึงอุปกรณ์ mechanic gyroscope และ accelerometer แต่จะขึ้นอยู่กับระบบ nanoelectromechanical และนาโนเทคโนโลยี

2.4 การสื่อสารอนุกรมแบบ I2C (Inter Integrate Circuit Bus)

I2C ย่อมาจาก Inter Integrate Circuit Bus (IIC) นิยมเรียกสั้นๆ ว่า I2C bus เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) เพื่อใช้ ติดต่อสื่อสาร ระหว่าง ไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอก ซึ่งถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors ซึ่งสามารถเชื่อมต่ออุปกรณ์จำนวนมากๆ ตัวเข้าด้วยกันได้ ทำให้ไมโครคอนโทรลเลอร์ใช้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ SCL ,SDA สำหรับติดกับอุปกรณ์แบบ 2 ทิศทาง โดยที่ขาสัญญาณทั้ง 2 จะต้องต่อกับตัวต้านทานแบบ pull up 2-10K เนื่องจากเอาต์พุตมีลักษณะเป็น แบบ Open Darin หรือเป็นแบบ Open Collector เพื่อให้เอาต์พุตเชื่อมต่อกันได้หลายตัวดังรูปที่ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

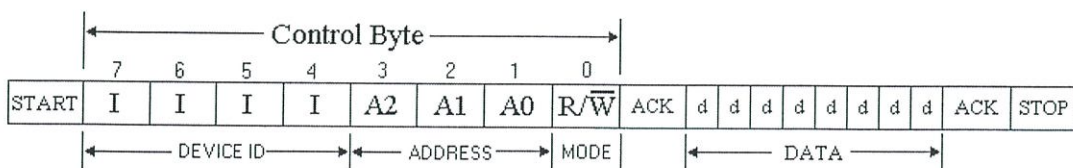


รูปที่ 2.10 การสื่อสารอนุกรมแบบ I2C

เนื่องจากรูปแบบการเชื่อมต่ออุปกรณ์ด้วยระบบบัสเช่นนี้ คือใช้สายเชื่อมต่อเพียงสองเส้นแต่สามารถเชื่อมต่ออุปกรณ์จำนวนหลายๆตัวร่วมกันในบัสเดียวได้ จึงนิยมนำมาใช้กับไมโครคอนโทรลเลอร์ เนื่องจากระบบการทำงานของวงจรถองการให้มีขนาดเล็กและใช้งานได้หลากหลาย จึงมีผู้นำระบบการเชื่อมต่อแบบนี้ไปใช้เป็นมาตรฐานในการสื่อสารกันระหว่างอุปกรณ์แต่ละประเภท

2.4.1 การเขียน-อ่านข้อมูลกับอุปกรณ์แบบ I2C BUS

การรับ-ส่งข้อมูลแบบ I2C BUS ไมโครคอนโทรลเลอร์จะเริ่มต้นการส่งข้อมูลด้วยการส่งสถานะเริ่มต้น (START Conditions) เพื่อแสดงการขอใช้บัส แล้วตามด้วยรหัสควบคุม (Control Byte) ซึ่งประกอบ ด้วยรหัส ประจำตัวอุปกรณ์ Device ID ,Device Address และ Mode ในการเขียนหรืออ่านข้อมูล เมื่ออุปกรณ์ รับทราบ ว่า ไมโครคอนโทรลเลอร์ต้องการ จะติดต่อด้วยก็ต้งส่งสถานะรับรู้ (Acknowledge) หรือแจ้งให้ ไมโครคอนโทรลเลอร์รับรู้ ว่าข้อมูลที่ได้ส่งมามีความถูกต้อง เมื่ออุปกรณ์ รับทราบ ว่า ไมโครคอนโทรลเลอร์ต้องการ จะติดต่อด้วยก็ต้งส่งสถานะรับรู้ (Acknowledge) หรือแจ้งให้ ไมโครคอนโทรลเลอร์รับรู้ ว่าข้อมูลที่ได้ส่งมามีความถูกต้อง และเมื่อสิ้นสุดการส่งข้อมูล ไมโครคอนโทรลเลอร์จะต้งส่งสถานะสิ้นสุด (STOP Conditions) เพื่อบอกกับอุปกรณ์ว่า สิ้นสุดการใช้บัส

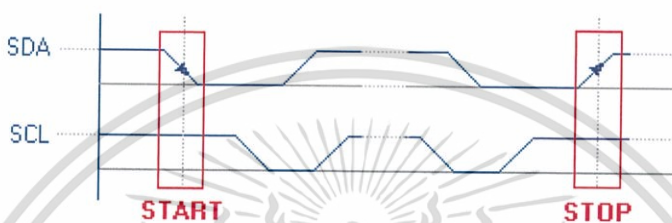


รูปที่ 2.11 การเขียน-อ่านข้อมูลกับอุปกรณ์แบบ I2C BUS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 การกำหนดสถานะเริ่มต้น และสถานะสิ้นสุดของ I2C BUS

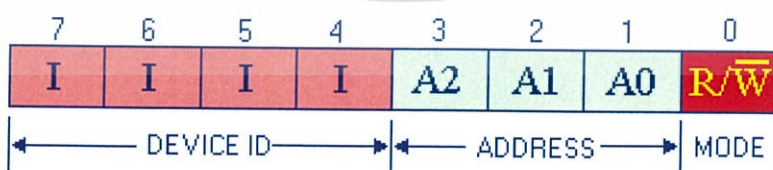
ลักษณะการกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I2C BUS เกิดเมื่อต้องการส่งข้อมูลไมโครคอนโทรลเลอร์ทำการส่งสถานะเริ่มต้น (START Conditions) คือให้ SDA เปลี่ยนจาก 1 มาเป็น 0 ในขณะที่ SCL มีค่าเป็น 1 และเมื่อสิ้นสุดการการใช้บัส ไมโครคอนโทรลเลอร์จะต้องส่งสถานะสิ้นสุด (STOP Conditions) คือให้ SDA เปลี่ยนจาก 0 มาเป็น 1 ในขณะที่ SCL มีค่าเป็น 1



รูปที่ 2.12 การกำหนดสถานะเริ่มต้น และสถานะสิ้นสุดของ I2C BUS

2.4.3 รหัสควบคุมของ I2C BUS (Control Byte)

รหัสควบคุมของ I2C BUS ประกอบด้วยรหัสประจำตัวของอุปกรณ์ (Device ID) ประกอบด้วยบิต 1-7 และบิต 0 เป็นบิตควบคุมการเขียนอ่าน โดยรหัสประจำตัวของอุปกรณ์ ประกอบด้วยรหัสประจำตัวจากผู้ผลิต Product ID 4 บิต (บิต 4-7) ที่เปลี่ยนแปลงแก้ไขไม่ได้ และ Device Address 3 บิต (บิต 1-3) ซึ่งผู้ใช้สามารถกำหนดเองได้ รวมแล้วเป็นรหัส 7 บิต ใช้ระบุตัวอุปกรณ์ ที่ต่ออยู่บนบัส จะมีค่าซ้ำกันไม่ได้ สำหรับบิตควบคุมการเขียนอ่าน (Mode) บิต 0 เมื่อไมโครคอนโทรลเลอร์ต้องการเขียนข้อมูลไปยังอุปกรณ์ก็กำหนดให้บิตนี้เป็น 0 และเมื่อต้องการอ่านข้อมูล จากอุปกรณ์ก็กำหนดให้บิตนี้เป็น 1

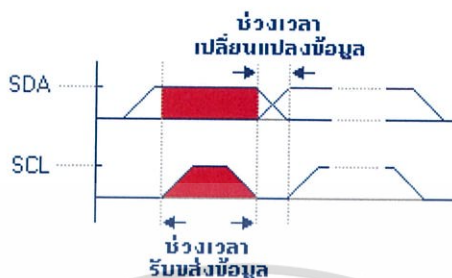


รูปที่ 2.13 รหัสควบคุมของ I2C BUS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 ช่วงเวลารับส่งบิตข้อมูลของ I2C BUS

ในสภาวะการรับ-ส่งข้อมูล จะกระทำในขณะที่ขา SCL เป็น 1 และการเปลี่ยนแปลงข้อมูล จะกระทำในขณะที่ขา SCL เป็น 0



รูปที่ 2.14 แสดงการรับส่งบิตข้อมูลของ I2C BUS

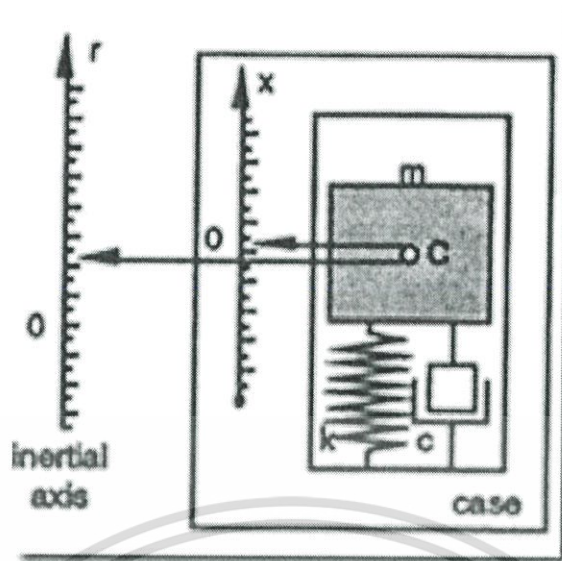
2.5 Accelerometer และ Gyro Module

Inertial Measurement Unit (IMU) หรืออุปกรณ์ที่ใช้ใน INS (Inertial Navigation System) ถูกเรียกว่า Inertial Measurement Units (IMU) ซึ่งเป็นส่วนประกอบหลักของ INSs ที่ใช้ในเครื่องบิน ยานอวกาศ และ เรือ เช่นเดียวกับจรวดขีปนาวุธ IMU ประกอบด้วย 2 ส่วนหลักคือ Accelerometers 3 ทิศทาง และ Gyroscopes 3 ทิศทาง ซึ่งรับความเร่งยานพาหนะ และความเร็วเชิงมุมตามลำดับ โดย mechanic gyroscope และ accelerometer เป็นอุปกรณ์ที่นำมาเพื่อแสดงหลักการพื้นฐานของ IMU และสำหรับ MEMS นั้นจะขึ้นอยู่กับระบบ Nanoelectromechanical และนาโนเทคโนโลยี เซนเซอร์ใน IMU

2.5.1 Accelerometer

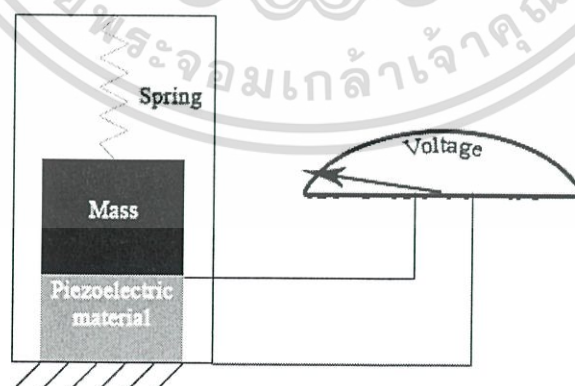
Accelerometer เป็นเซนเซอร์ที่ใช้วัดความเร่งเชิงเส้นในแนวแกน x, y, และ z ยกตัวอย่างเช่น เซนเซอร์อยู่บนพื้นโลกโดยปกติแล้วจะต้องอ่านค่าได้ 9.81 m/s^2 คือค่าความเร่งเนื่องจากแรงโน้มถ่วงของโลก หรือมีค่าเท่ากับ $1g$ หรืออาจเรียกเซอร์เซอร์เหล่านี้ว่า IMU (Inertial Measurement Unit) ซึ่ง Accelerometer เป็นอุปกรณ์ที่ใช้วัดความเร่งตามแนวแกนที่เฉพาะเจาะจงสังเกตได้ว่า Accelerometers ใน IMU รับเพียง specific forces แนวคิดนี้เป็นสิ่งสำคัญใน inertial navigation หรือระบบนำทางอาศัยแรงเฉื่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 โครงสร้างของ Accelerometer อาศัยแรงเฉื่อย

นั่นคือ โครงสร้างของ accelerometer จะประกอบด้วยสปริงและลูกตุ้มน้ำหนัก เมื่อมีการเคลื่อนที่ด้วยความเร่งลูกตุ้มน้ำหนักจะถูกกดไปอีกฝั่งตรงข้ามกับการเคลื่อนที่ สปริงก็ทำหน้าที่ในการดึงลูกตุ้มน้ำหนักกลับเข้าที่อีกครั้งเมื่อหยุดการสั่นของตัวเซ็นเซอร์ภายใน โดยในการตรวจวัดความเร่งของลูกตุ้มที่อยู่ในระบบนั้นมีหลายชนิด เช่น เพียโซอิเล็กทริก สเตรนเกจ ชนิดใช้แสง ตรวจวัด วัดแรงเฉื่อย เป็นต้น การเคลื่อนที่ด้วยความเร่งที่คือความเร่งเท่ากับ 0 ค่าที่วัดได้ก็จะไม่เปลี่ยนแปลง



รูปที่ 2.16 โครงสร้างของ Accelerometer เพียโซอิเล็กทริก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่สามารถแบ่งลักษณะการตรวจวัดได้ 2 ลักษณะ

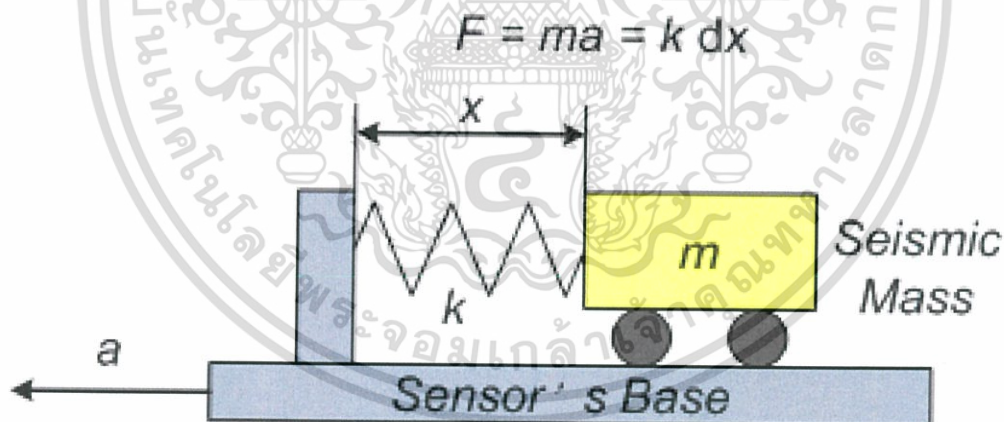
1. การตรวจวัดการช็อก (shock) และการสั่นสะเทือน (vibration) ซึ่งการช็อก คือ อัตราเร่งขนาดมหาศาลที่เกิดขึ้นในช่วงเวลาสั้นๆ การสั่นสะเทือน คือ อัตราเร่งขนาดเล็กที่เกิดขึ้นซ้ำกันไปเรื่อยๆ

2. การตรวจวัดอัตราเร่งของวัตถุ เพื่อนำข้อมูลไปใช้ในการระบุตำแหน่ง ความเร็ว และระยะทางที่ได้จากการเคลื่อนที่

มิเตอร์วัดความเร่งนี้โดยหลักๆแล้วจะแบ่งเป็น 2 ชนิด

1. มิเตอร์วัดอัตราเร่งแบบไซซมิกแมส (seismic mass accelerometer) มิเตอร์ชนิดนี้อาศัยหลักการตรวจวัดระยะขจัดเชิงเส้นแล้วนำไปคำนวณหาอัตราเร่งที่เกิดขึ้น โดยเทคนิคดังกล่าวสามารถอธิบายได้ก็คือ วัตถุชิ้นหนึ่งจะมีความเร่งได้ ก็จะต้องมีแรงมากกระทำยิ่งมีแรงมากกระทำมาก ก็จะมี ความเร่งมาก ในขณะที่เดียวกันแรงต้านการเคลื่อนที่ก็จะมากด้วย

นอกจากนี้เมื่อมีแรงมาทำให้วัตถุเกิดการเคลื่อนที่ ก็จะมีระยะขจัด ซึ่งก็จะแปรผันตรงกับแรงที่มากกระทำที่วัตถุ ยิ่งแรงมากระยะขจัดยิ่งมากจากความสัมพันธ์ดังกล่าวได้นำไปใช้เป็นหลักการพื้นฐานของมิเตอร์วัดอัตราเร่งแบบไซซมิกแมสในการตรวจวัดอัตราเร่งของวัตถุในเทอมของระยะขจัดที่เกิดขึ้น



รูปที่ 2.17 โครงสร้างพื้นฐานของมิเตอร์วัดอัตราเร่งแบบไซซมิกแมส

โครงสร้างนี้มีมวล m ที่เรียกว่ามวลตรวจการสั่นไหว (seismic mass) ยึดติดอยู่กับสปริงที่มีค่า spring constant เท่ากับ k และมวลนี้สามารถเคลื่อนที่ในแนวระดับได้ ซึ่งหลักการทำงานคือเมื่อตัวเซนเซอร์ตัวนี้ถูกทำให้มีอัตราเร่งเกิดขึ้นจะส่งผลให้มวล m เคลื่อนที่ซึ่งระยะที่เคลื่อนที่ออกไปจะเป็นระยะขจัดเท่ากับ x และมีทิศทางตรงกันข้ามกับการเคลื่อนที่ของตัวมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น อัตราเร่ง a ของวัตถุสามารถคำนวณหาค่าได้จากความสัมพันธ์ต่อไปนี้

$$a = xk/m$$

โดยที่

a คือ อัตราเร่งของวัตถุ หน่วย เมตร/วินาที

x คือ ระยะขจัดของมวล m หน่วย เมตร

k คือ ค่าคงที่ของสปริง หน่วย นิวตัน/เมตร

m คือ น้ำหนักของมวล m หน่วย กิโลกรัม

จากสมการดังกล่าวจะแสดงให้เห็นว่า

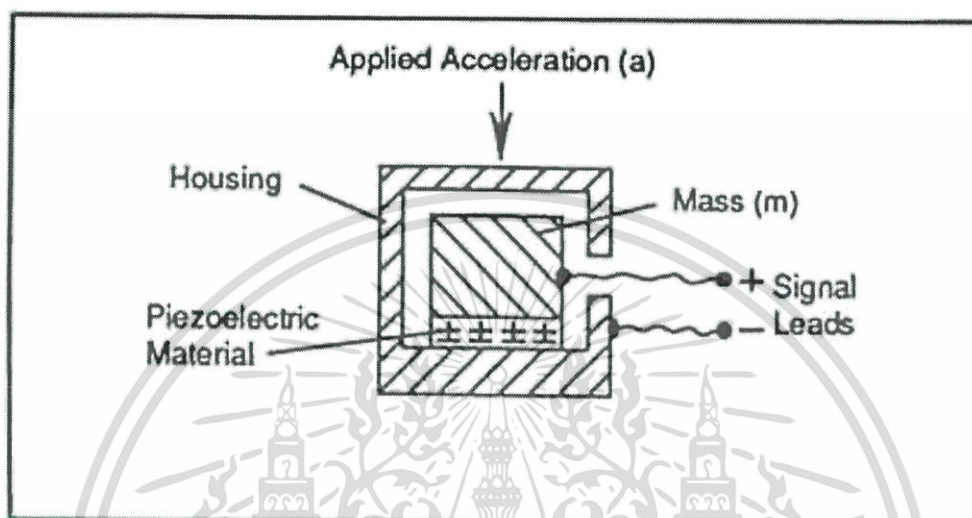
เมื่ออัตราเร่งของวัตถุมีค่าเพิ่มขึ้น ทำให้ระยะขจัดของมวล m มีค่าเพิ่มขึ้นตามไปด้วย เมื่ออัตราเร่งของวัตถุมีค่าลดลง ทำให้มวล m เคลื่อนที่ไปดันสปริง เมื่ออัตราเร่งของวัตถุหยุดลง ก็จะทำให้มวล m เคลื่อนที่กลับมาอยู่ตำแหน่งเดิม (ตำแหน่งอ้างอิง) แต่ในทางปฏิบัติเราสามารถวัดระยะขจัดของมวล m ได้โดยอาศัยมิเตอร์อีกชนิดหนึ่ง คือ มิเตอร์วัดระยะขจัดเชิงเส้น (LVDT, potentiometer)

ส่วนการวิเคราะห์หาอัตราเร่งที่เกิดขึ้นเราสามารถคำนวณหาได้โดยใช้คอมพิวเตอร์ มิเตอร์วัดอัตราเร่งแบบไซซมิกแมสนี้จะนิยมใช้ในการตรวจวัดลักษณะการสั่น และลักษณะการสั่นสะเทือนที่มีความถี่ต่ำมากๆ เช่น ในเครื่องมือตรวจวัดแผ่นดินไหว หรือในเครื่องมือตรวจวัดการปะทุใต้ดินของภูเขาไฟ ฯลฯ

2. มิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริก (piezoelectric accelerometer)

คุณสมบัติพื้นฐานทางไฟฟ้าของผลึกเพียโซอิเล็กทริก (piezoelectric crystal) ถูกค้นพบโดย Pierre และ Jacques Curie ในราวปี ค.ศ.1880 ซึ่ง piezoelectric crystal นี้มันมีคุณสมบัติพิเศษ คือ เมื่อถูกแรงทางกลมากระทำ จะสร้างประจุไฟฟ้าขึ้นมา โดยเป็นส่วนสัดส่วนกับแรงกระทำนั้น ซึ่งจากคุณสมบัติพิเศษนี้ได้ถูกดัดแปลงนำไปใช้สร้างอุปกรณ์ต่างๆ มากมาย เช่น ใช้เป็นแบตเตอรี่จ่ายพลังงานไฟฟ้าให้กับนาฬิกาข้อมือดิจิตอลที่ใช้ทั่วไป และใช้สร้างมิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริกอีกด้วย

โครงสร้างของมอเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริกจะประกอบด้วย seismic mass ยึดติดกับ piezoelectric crystal และบรรจุอยู่ในตัวถังป้องกัน โดย piezoelectric crystal ที่นิยมนำมาใช้งาน ได้แก่ ผลึกควอตซ์ และผลึกโซเดียมโปตัสเซียมตาเตรต (sodium potassium tartrate) เพราะมีความทนทานต่อแรงกระทำ และราคาไม่แพงมากนัก



รูปที่ 2.18 โครงสร้างพื้นฐานของมิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริก (piezoelectric accelerometer)

สามารถอธิบายการทำงานได้ ดังนี้ เมื่อ seismic mass (m) ถูกทำให้เกิดอัตราเร่งขึ้น (ถูกกด) จะส่งผ่านแรงกดไปกระทำกับ piezoelectric crystal ที่ถูกยึดติดอยู่ด้วยกัน ด้วยคุณสมบัติพิเศษจะทำให้ประจุไฟฟ้าถูกสร้างขึ้น และถูกสายนำสัญญาณออกไปยังเอาต์พุตของวงจร โดยที่ด้านเอาต์พุตจะต้องมีวงจรขยายประจุไฟฟ้า (charge amplifier) เพื่อขยายค่าประจุไฟฟ้าที่ได้ ให้เป็นแรงดันเอาต์พุตตามสัดส่วนของอัตราเร่งที่เกิด จะสามารถแสดงผลได้ด้วยโวลต์มิเตอร์

มิเตอร์วัดอัตราเร่งแบบเพียโซอิเล็กทริกตอบสนองต่อทางด้านความถี่สูงได้ดี แต่ในทางกลับกันก็จะมีผลตอบสนองทางด้านความถี่ต่ำที่ไม่ดีนัก มีขนาดค่อนข้างเล็ก น้ำหนักเบา และสามารถใช้งานที่มีอัตราเร่งได้สูงถึง $250,000 \text{ m}^*\text{s}^{-2}$

โดยทั่วไปถ้าใช้แค่ accelerometer อย่างเดียวมุม yaw จะไม่ถูกนำมาใช้ ดังนั้นในบทความนี้จะพูดถึงแค่ roll และ pitch เท่านั้น วิธีคำนวณ roll และ pitch นั้นจะขึ้นอยู่กับชนิดของ accelerometer ในที่นี้จะอ้างถึงตระกูล ADXL345 เนื่องจากมี Library ช่วยแปลงค่าสัญญาณ analog ที่อ่านได้จาก accelerometer ให้อยู่ในรูปของแรง “g” (หรือความเร่ง) ถ้าไม่ใช้ library นี้ก็จะต้องเขียนโปรแกรมในส่วนของการแปลงนี้เองโดยใช้สูตรด้านล่าง

$$G_{acc} = Raw_{acc} \times \frac{Range}{2^{resolution}-1} \quad (2.1)$$

โดยที่ค่า Range และ resolution เป็นค่าเฉพาะของแต่ละ accelerometer เช่น สำหรับ ADXL345 ค่าทั้งสองนี้คือ resolution = 10-bit และ Range = ± 2 g โดยทั่วไปค่าความเร่งที่วัดได้ในแต่ละแกน (Xg, Yg, Zg) จะมีสัญญาณรบกวนผสมอยู่ จึงควรทำการลดปริมาณสัญญาณรบกวนนี้ออกไป โดยใช้ Low-Pass Filter ซึ่ง implement ได้ดังนี้

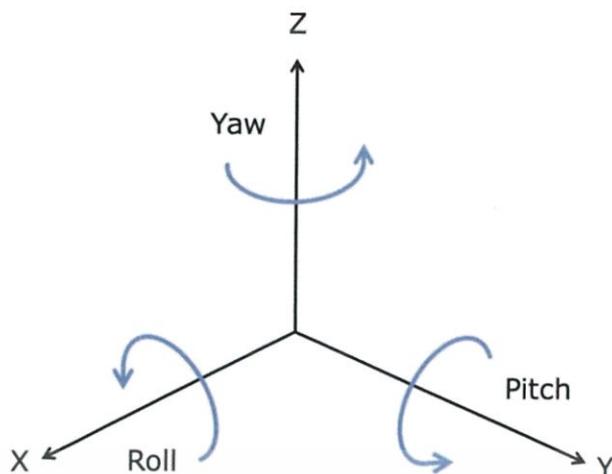
$$y(k) = \alpha \times x(k) + (1 - \alpha) \times y(k - 1) \quad (2.2)$$

ค่าแอลฟา เป็นค่าคงที่ของ Low pass filter โดยจะมีค่าน้อยกว่าหรือเท่ากับ 1 ค่าแอลฟาที่เหมาะสมขึ้นอยู่กับระบบและรูปแบบการใช้งาน ส่วน $y(k)$ และ $x(k)$ คือค่าความเร่งหลัง และก่อน filter ตามลำดับ $y(k-1)$ คือค่าความเร่งที่ถูก filter ของ loop ที่แล้ว หลังจากที่ได้ความเร่งที่ถูก filter ของทั้ง 3 แกนมา (fXg, fYg, fZg) สามารถคำนวณหา pitch และ roll ได้ดังนี้

$$Pitch = \arctan \left(\frac{fYg}{\sqrt{fXg^2 + fZg^2}} \right) \quad (2.3)$$

$$Roll = \arctan \left(\frac{-fXg}{fZg} \right) \quad (2.4)$$

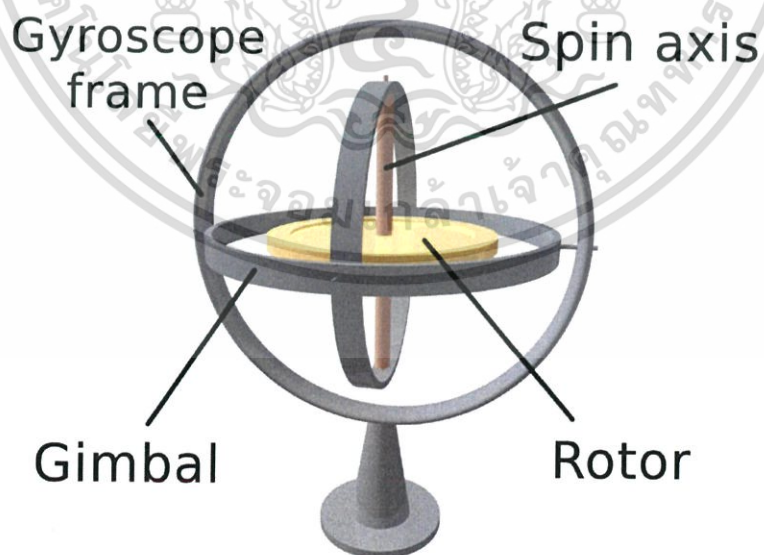
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 Roll, Pitch, Yaw angles

2.5.2 Gyroscopes

Gyroscope เป็นอุปกรณ์สำหรับการวัด หรือการรักษาการปรับทิศทาง ขึ้นอยู่กับหลักการของการอนุรักษ์โมเมนตัมเชิงมุม ซึ่งอุปกรณ์นี้นำมาใช้สำหรับตรวจเช็คสภาพการเอียงการโคลงเคลง ของวัตถุที่ติดตั้งอยู่ เช่น เครื่องบิน เรือ หุ่นยนต์ จรวด จานดาวเทียม ซึ่งของเหล่านี้ล้วนต้องการที่จะทราบสถานะความสมดุลของตัวเองอยู่ตลอดเวลา รูปที่ 2.17 Mechanic gyroscope ซึ่งมี two-degree of freedom (TDF)



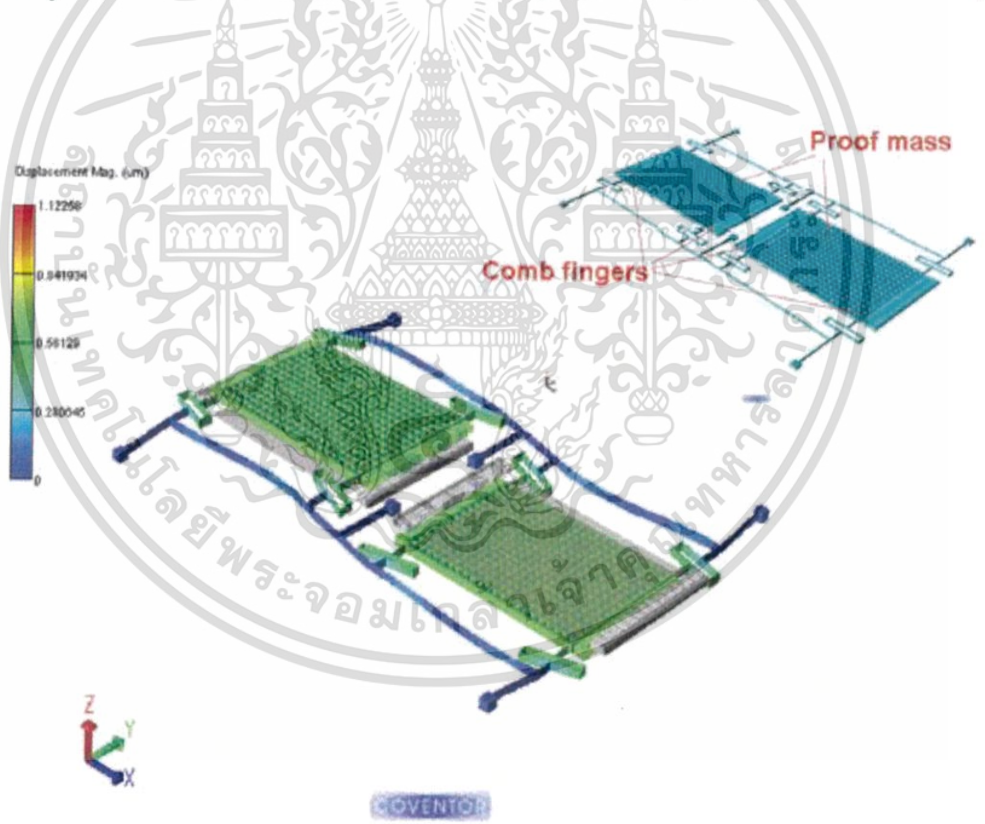
รูปที่ 2.20 Mechanic gyroscope

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Gyroscope นั้น จะเป็นไปตามกฎ ของนิวตันคือ มวลจะเคลื่อนที่เป็นเส้นตรงด้วยความเร็วคงที่ ถ้าไม่มีแรงภายนอกมากระทำ เมื่อตัวไจโรหมุนไป 90 องศา จุดบนจะหมุนเปลี่ยนตำแหน่งไป 90 องศา และยังเคลื่อนที่ไปทางซ้าย เช่นเดียวกับจุดล่าง เมื่อหมุนขึ้นมา 90 องศา ยังคงเคลื่อนที่ไปทางขวา ทำให้ล่อเกิดการหมุนควง ขณะที่จุดบนและจุดล่างเปลี่ยนตำแหน่งไป 90 องศา การเคลื่อนที่ในครั้งแรก จะถูกยกเลิกไป ไม่เกิดการพลิกของล้อ ดังนั้นแกนหมุนของไจโรจะเหมือนกับห้อยอยู่กับที่ตลอดเวลา

ดังนั้นเมื่อเอาคุณสมบัตินี้ไปสร้างอุปกรณ์วัดระดับก็จะสามารถรู้ได้โดยที่อุปกรณ์ที่ติดตั้ง Gyro เอียงไปจากเดิมเท่าไร ซึ่งเมื่อ โรเตอร์หมุนแล้ว ก็จะคงตำแหน่งอยู่แบบนั้น กรอบที่เป็นแหวนซึ่งอยู่ล้อมรอบเมื่อเราได้ติดตั้งตัววัดระยะ วัดการหมุนก็สามารถเช็คมุมที่เปลี่ยนไปได้

MEMS gyroscope ซึ่งเป็นอุปกรณ์สำหรับการตรวจสอบการเคลื่อนที่และการหมุนของวัตถุ ซึ่งเป็นชิ้นส่วนที่สำคัญในเครื่องเล่นเกมส์ กลุมนิรภัย และโทรศัพท์มือถือ เป็นต้น ตัวอย่างหนึ่งของระบบแสดงในรูปที่



รูปที่ 2.21 ไดอะแกรมแสดงโครงสร้างของ Gyroscope รูปร่างแบบ comb fingers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยอุปกรณ์เหล่านี้จะถูกนำไปติดตั้งบนวัตถุที่ต้องการตรวจสอบการเคลื่อนที่ เมื่อวัตถุเกิดการเคลื่อนที่ มวล (proof mass) ของ gyroscope จะเคลื่อนที่เช่นกัน แต่เคลื่อนที่ช้ากว่า เนื่องจากความเฉื่อย และทำให้ตำแหน่งของปลายทั้งสองด้านของมวลเปลี่ยนแปลงไป ตำแหน่งที่เปลี่ยนไปนี้สามารถวัดได้ด้วยเซนเซอร์แบบตัวเก็บประจุไฟฟ้า (capacitive sensing) ทำให้รู้ค่าความเร่งเชิงเส้นและความเร่งเชิงมุมของการเคลื่อนที่นั้นได้ ข้อดีของ MEMS gyroscope คือมีขนาดเล็กทำให้มีความไวในการตอบสนองสูง และสามารถขยายช่วงความถี่ในการทำงานให้ครอบคลุมความถี่ที่สูงขึ้นได้อีกด้วย

โดยในงานวิจัยนี้เลือกใช้ Accelerometer และ Gyro Module รุ่น GY-521 หรือที่เรียกว่า ไรโร MPU6050 (MPU6050 คือชื่อชิพประมวลผลบนตัว GY-521) คือ โมดูลเซนเซอร์ที่ตรวจจับการเคลื่อนไหวและความเอียงของวัตถุ โดยตรวจวัดจากความเร่งเชิงเส้น (Linear Acceleration) และความเร่งเชิงมุม (Angular Velocity หรือใช้ Gyroscope) ถือเป็นอุปกรณ์แบบ 6 DOF (6 Degrees of Freedom) คืออุปกรณ์ที่สามารถตรวจวัดค่าได้ทั้ง 6 แกนคือ A_x , A_y , A_z , G_x , G_y และ G_z

2.6 ภาษา C#

ภาษา C# (ซี-ชาร์ป) เป็นภาษาโปรแกรมคอมพิวเตอร์ระดับสูงที่ใช้สำหรับเขียนโปรแกรมคอมพิวเตอร์ ที่ได้รับความนิยมเพิ่มมากขึ้นเรื่อยๆ ในปัจจุบัน และเป็นภาษาโปรแกรมคอมพิวเตอร์ที่เหมาะสมสำหรับผู้ที่ยังเริ่มต้นสนใจที่จะเขียนโปรแกรมคอมพิวเตอร์เป็นอย่างยิ่ง ซึ่งภาษา C# ถูกพัฒนามาจากภาษา C++ (ซี-พลัสพลัส) และมีโครงสร้างแบบเชิงวัตถุ (object-oriented programming) โดยใช้ Visual Studio (วิซวล-สตูดิโอ) เป็นเครื่องมือสำหรับพัฒนาโปรแกรมคอมพิวเตอร์ ซึ่ง Visual Studio เป็นเครื่องมือที่คอยอำนวยความสะดวกในการพัฒนาโปรแกรมคอมพิวเตอร์ ทำให้ผู้เขียนโปรแกรมสามารถพัฒนาโปรแกรมคอมพิวเตอร์ได้ไม่ยาก

2.6.1 การพัฒนาของภาษา C#

ภาษา C# ได้รวบรวมข้อดีของภาษาต่างๆ เข้าไว้ด้วยกัน ไม่ว่าจะเป็นภาษา Java ภาษา C และ ภาษา C++ โดยมีข้อดีดังนี้

1. เป็นภาษาที่เขียนง่าย ไม่ซับซ้อนและเรียบง่าย เพราะคล้ายภาษา Java ภาษา C และ ภาษา C++ ทำให้หลายคนเข้าใจได้ไม่ยาก
2. เป็นภาษาโปรแกรมคอมพิวเตอร์ยุคใหม่ที่ถูกสร้างขึ้นมาสำหรับการพัฒนาโปรแกรมคอมพิวเตอร์ ภายใต้แนวคิด .NET Framework ซึ่งเป็นแนวคิดที่ได้รับความนิยมสูงที่สุดในปัจจุบัน
3. เป็นภาษาที่ถูกออกแบบมาให้ทำงานบน .NET Framework (ดอตเน็ต-เฟรมเวิร์ก) โดย .NET Framework เป็นรูปแบบในการพัฒนาโปรแกรมคอมพิวเตอร์สมัยใหม่ ซึ่งบริษัทไมโครซอฟท์

เป็นผู้พัฒนา ซึ่งคุณสมบัติที่สำคัญของ .NET Framework ก็คือ ผู้ใช้งานสามารถใช้งานบนระบบไม่จำกัดทุกที่ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฮาร์ดแวร์ (Hardware) หรือระบบปฏิบัติการ (Operating System) ที่แตกต่างกันได้อย่างไม่มีปัญหา เช่น เครื่องพีซีกับเครื่องแมค หรือระบบปฏิบัติการวินโดวส์กับระบบปฏิบัติการแมคอินทอช เป็นต้น ดังนั้น ผู้เขียนโปรแกรมจึงสามารถเขียนโปรแกรมคอมพิวเตอร์ใหม่ๆ ได้โดยง่าย รวดเร็ว และไม่ต้องติดข้อจำกัดต่างๆ อย่างเช่นการเขียนโปรแกรมคอมพิวเตอร์ในสมัยก่อนอีกต่อไป

4. เป็นภาษาที่แข็งแกร่ง เพราะเป็นภาษาที่ได้มีการแก้ไขข้อบกพร่องบางอย่างของภาษา Java ภาษา C และ ภาษา C++ เหล่านั้น ทำให้ ภาษา C# เป็นภาษาที่มีความสมบูรณ์ตามแบบฉบับของโครงสร้างแบบเชิงวัตถุ (object-oriented programming)

2.6.2 เครื่องมือสำหรับพัฒนาโปรแกรม

การเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษา C# นั้น จะมีเครื่องมือที่ช่วยคอยอำนวยความสะดวกสบายให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมคอมพิวเตอร์ได้อย่างง่ายดาย รวดเร็ว และ ผู้เขียนโปรแกรมสามารถแก้ไขข้อผิดพลาดในการเขียนโปรแกรมได้ง่ายขึ้นอีกด้วย ซึ่งเครื่องมือดังกล่าวก็คือ โปรแกรม Visual Studio นั่นเอง

Visual Studio เป็นซอฟต์แวร์ประเภท IDE (Integrated Development Environment) ซึ่งเป็นการนำแนวความคิดการทำงานแบบรวมศูนย์มาใช้ คือ การทำให้วงจรการพัฒนาระบบทั้งหมดทำงานได้อย่างสะดวก รวดเร็ว และ ง่ายดาย เริ่มตั้งแต่การวิเคราะห์ ออกแบบจนถึงการนำไปปรับใช้ให้เหมาะสมกับวัตถุประสงค์ของการเขียนโปรแกรมคอมพิวเตอร์นั้นๆ

2.6.3 โครงสร้างโปรแกรมภาษา C#

โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐานจะประกอบด้วยส่วนของโปรแกรมหลักแต่จะไม่มีส่วนของโปรแกรมย่อย (subroutine) โดยแสดงดังรูปที่ 2.22

```
namespace (1)
{
    class (2)
    {
        static void Main ( )
        {
            (3)
        }
    }
}
```

รูปที่ 2.22 โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.22 แสดงโครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน โดยมีรายละเอียดดังนี้

1. หมายเลข (1) เป็นการระบุชื่อของ namespace ซึ่งใช้ในการกำหนดขอบเขตให้กับคลาสต่างๆ รวมถึงใช้ในการจัดโครงสร้างของโปรแกรมขนาดใหญ่ให้เป็นสัดส่วนอีกด้วย โดยเฉพาะอย่างยิ่งในการเขียนโปรแกรมคอมพิวเตอร์ที่ซับซ้อนโดยมีผู้เขียนโปรแกรมหลายคน นอกจากนี้ การกำหนด namespace ยังช่วยป้องกันปัญหาการตั้งชื่อคลาสหรือค่าคงที่อื่นๆ ซ้ำกันได้
2. หมายเลข (2) เป็นการระบุชื่อของ class
3. หมายเลข (3) เป็นการระบุพื้นที่สำหรับคำสั่งต่างๆ ที่ผู้เขียนโปรแกรมต้องการให้คอมพิวเตอร์ปฏิบัติตาม

นอกจากนี้ ในบางกรณี ผู้เขียนโปรแกรมสามารถที่จะไม่เขียนในส่วนของ namespace ได้ ถ้าโปรแกรมคอมพิวเตอร์นั้นมีขนาดเล็ก และไม่ซับซ้อนมากนัก ซึ่งการที่ไม่เขียนในส่วนของ namespace จะถือว่า class ที่ถูกสร้างขึ้นมาอยู่ใน namespace กลาง โดยแสดงดังรูปที่ 2.23

```

class (2)
{
    static void Main ( )
    {
        (3)
    }
}

```

รูปที่ 2.23 โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน กรณีไม่เขียนในส่วนของ namespace

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง โครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน ซึ่งจะแสดงข้อความ Hello C# ออกทางจอภาพ และจากนั้นรอนจนกว่าผู้ใช้งานจะกด Enter แล้วจบการทำงาน
กรณีนี้ที่ 1 เขียนในส่วนของ namespace โดยแสดงดังรูปที่ 2.24

```
namespace HelloApp
{
    class HelloC#
    {
        static void Main ( )
        {
            System.Console.WriteLine ("Hello C#");
            System.Console.ReadLine ( );
        }
    }
}
```

รูปที่ 2.24 ตัวอย่างโครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน กรณีเขียนในส่วนของ namespace

กรณีนี้ที่ 2 ไม่เขียนในส่วนของ namespace โดยแสดงดังรูปที่ 2.25

```
class HelloC#
{
    static void Main ( )
    {
        System.Console.WriteLine ("Hello C#");
        System.Console.ReadLine ( );
    }
}
```

รูปที่ 2.25 ตัวอย่างโครงสร้างโปรแกรมภาษา C# ขั้นพื้นฐาน กรณีไม่เขียนในส่วนของ namespace

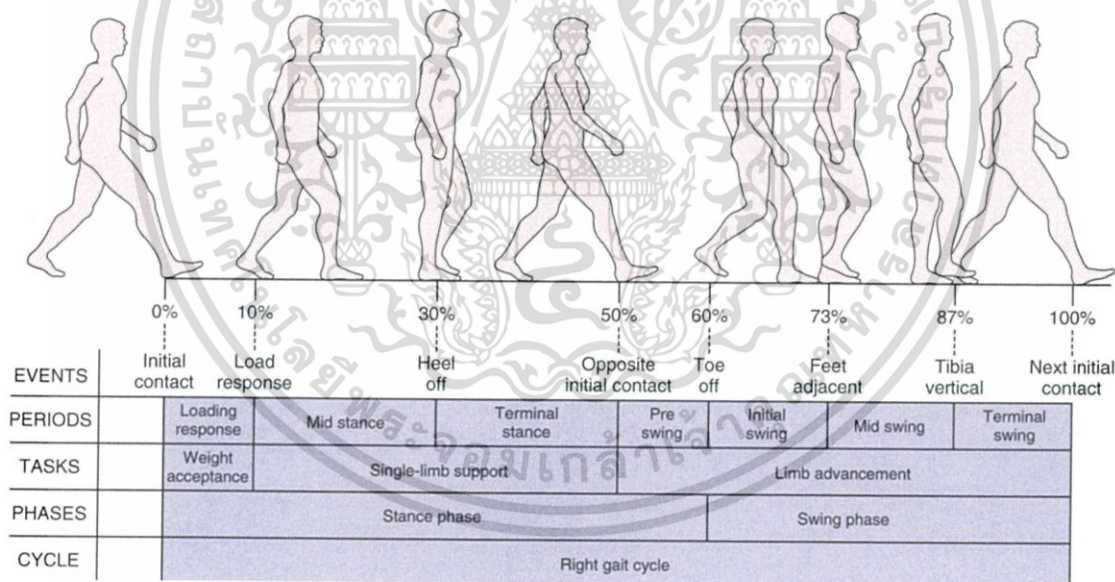
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 งานวิจัยที่เกี่ยวข้อง

เฉลิมพล คงเขียว (2553) ระบบภาพเคลื่อนไหวสำหรับการวิเคราะห์การเดินของมนุษย์ โดยงานวิจัยนี้ทำการศึกษาการวิเคราะห์การเดินของมนุษย์ ซึ่งมีข้อมูลเกี่ยวกับองศาการงอ และความเร็วในการเคลื่อนไหวของอวัยวะ และข้อต่อต่างๆ ของร่างกายมนุษย์ ซึ่งในงานวิจัยนี้เป็นกระบวนการเพื่อหาวิธีการแก้ปัญหาการเดิน หรือลักษณะความผิดปกติของการเดิน โดยการใช้ระบบภาพเคลื่อนไหวแบบไดนามิกในการวิเคราะห์การเดินของมนุษย์ โดยทำการทดลองกับอาสาสมัครจำนวน 15 คน โดยในงานวิจัยได้ข้อมูลที่มีความจำเป็นต่อการคำนวณของผู้จัดทำ ได้แก่ ค่ามุมงอของเข่าที่มากที่สุด ในขณะที่เดินคือ 65 องศา และมีความเร็วเชิงมุมอยู่ในช่วง 0.7-0.9 องศา/วินาที ซึ่งเป็นข้อมูลสำคัญต่อการใช้ในการคำนวณความแข็งแรงของอุปกรณ์จำกัดและวัดมุมงอของเข่า

การเดินของมนุษย์นั้นเกิดจากการเคลื่อนไหวของขา 2 ข้าง โดยแบ่งวงจรการเดิน (Gait Cycle) ออกเป็น 2 ช่วง ได้แก่

1. Stance phase คือ ในช่วงที่เท้าทั้ง 2 ข้างยังอยู่บนพื้น
2. Swing phase คือ ในช่วงที่เท้าลอยจากพื้นในขณะที่เดิน



รูปที่ 2.26 วงจรการเดิน (Gait Cycle)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Phases	Hip(degree)	Knee(degree)	Ankle(degree)	Muscles
Heel strike	Flex 25	Fully extend	Neutral	Gluteus maximus, quadriceps, hamstrings, foot and toe dorsiflexors
Foot Flat	Flex 23	Flex 20	Plantarflex 15	Same Heel strike
Mid stance	Flex 10	Flex 10	Dorsiflex 3	Iliopsoas, gluteus minimus, tensor fascia lata, quadriceps, plantar flexors
Heel off	Hyperextend 10	Flex 2	Dorsiflex 15	Iliopsoas, hamstring, plantar flexors
Toe off	Flex 10	Flex 40	Plantarflex 20	Iliopsoas, quadriceps, plantar flexors
Acceleration	Flex 5	Flex 65	Neutral	Iliopsoas, quadriceps, foot and toe dorsiflexors
Mid swing	Flex 25	Flex 65	Neutral	Iliopsoas
Deceleration	Flex 25	Fully extend	Neutral	Gluteus maximus, hamstring, foot and toe dorsiflexors

รูปที่ 2.27 การทำงานของข้อต่างๆ และกล้ามเนื้อในขณะเดิน

Grant Elliott, Gregory S. Sawicki, Andrew Marecki, and Hugh Herr ตีพิมพ์ในการประชุมวิชาการ 2013 IEEE International Conference on Rehabilitation Robotics โดยงานวิจัยนี้เสนอเกี่ยวกับการติดตั้งอุปกรณ์โครงร่างภายนอกเกี่ยวกับการสนับสนุนการเดิน ในขณะที่มีข้อสนับสนุนอย่างชัดเจนว่า ผลของการใช้สปริงที่ต่อแบบอนุกรมในชีวกลศาสตร์ของการวิ่งมากมาย แต่ในการใช้สปริงแบบขนานนั้นมีเพียงข้อมูลเฉพาะกรณีศึกษาต่างๆ ของการกระโดดนั้น งานจำนวนมากในการออกแบบกระดูกโครงร่างภายนอก หรือกายอุปกรณ์เสริม จะออกแบบให้อุปกรณ์นั้นขนานไปกับขาของผู้ทดลอง เพื่อให้เข้าใจถึงผลจากสภาวะแทรกซ้อนต่างๆ แรงของสปริงเป็นแรงจากธรรมชาติของวัสดุ และเป็นทางเลือกในการก่อกวนกระบวนการทำงานของร่างกายในการวิ่ง ซึ่งมันถูกกระตุ้นทางชีวภาพได้ และมีราคาถูกเหมาะสมแก่การนำมาใช้ ด้วยเหตุนี้ เราจึงทำการพิสูจน์สมมติฐานที่ว่า การเพิ่มองค์ประกอบยืดหยุ่นภายนอกเข้าไปที่เข้า ในช่วงท่าทางต่างๆ ของการวิ่ง สามารถลดการทำงานของกล้ามเนื้อยืดบริเวณเข้า โดยทำให้ quasi-stiffness รวมของข้อเข้ามีค่าคงที่ อุปกรณ์กระดูกโครงร่างภายนอกจะประกอบไปด้วย ส่วนประกอบที่มีความยืดหยุ่นที่ทำงานร่วมกับคลัทช์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อใช้ในการสร้างแรงบิดแก่กล้ามเนื้อยึดในท่าทางต่างๆ ของการวิ่ง และมีการตรวจจับการเคลื่อนไหวของวัตถุ 5 จุด เพื่อใช้ในการตรวจสอบผลกระทบต่อการวิ่งของอุปกรณ์ โดยแรงดึงของขาหรือแรงดึงรวมของเข่าไม่มีการเปลี่ยนแปลงอย่างมีนัยสำคัญ สังเกตได้จากการทำงานของสปริงที่ขนานกับเข่า อย่างไรก็ตาม ข้อมูลที่กล่าวมานี้แสดงให้เห็นความแตกต่างของการตอบสนองระหว่างการวิ่งปกติ กับการวิ่งแข่งขันระยะไกล ซึ่งแรงบิดรวมของเข่าถูกทำให้เพิ่มขึ้นด้วยอุปกรณ์ ความสัมพันธ์ระหว่างการศึกษาที่ผ่านมา และการใช้ประโยชน์อย่างมีประสิทธิภาพของแรงภายนอกชี้ให้เห็นข้อจำกัดของการประยุกต์ใช้อุปกรณ์อำนวยความสะดวกต่างๆ ซึ่งในงานวิจัยยังแสดงถึงตำแหน่งที่เหมาะสมต่อการติดตั้งอุปกรณ์กับตัวผู้ใช้อีกด้วย เพื่อป้องกันไม่ให้เกิดภาวะต่อกล้ามเนื้อของผู้ใช้ และทำให้ผู้ใช้สามารถใช้งานร่างกายได้อย่างเต็มประสิทธิภาพอีกด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีดำเนินการวิจัย

3.1 วิธีที่ใช้ศึกษาค้นคว้าและการวิจัยทดลอง

ในการศึกษาค้นคว้าการวิจัยนั้น เริ่มด้วยการศึกษาหาข้อมูลเกี่ยวกับอาการของผู้ป่วยที่ได้รับบาดเจ็บที่เข่า และวิธีการรักษาด้วยการผ่าตัด จากนั้นศึกษาวิธีการบำบัด และกายภาพฟื้นฟูร่างกายของผู้ป่วย เพื่อศึกษาลักษณะการเคลื่อนไหวของขา และเข่าซึ่งนำมาใช้ในการออกแบบอุปกรณ์ที่ใช้ในการวัดมุมงอของเข่าให้ได้ถูกต้องตามการเคลื่อนไหวของร่างกายผู้ป่วย

3.2 ลักษณะข้อมูล การเลือกข้อมูล และการทดลอง

Inertial Measurement Unit (IMU) ค่าการเคลื่อนไหวของขา 2 ส่วน คือ ขาท่อนบน และขาท่อนล่าง โดยใช้ค่า Accelerometer/Gyroscope ในแนวแกน X ของ Module มาคำนวณหาค่ามุมงอของเข่า

ในการทดลองเพื่อทดสอบการทำงานของอุปกรณ์ในการคำนวณหาค่ามุมงอของเข่า ทำการวัดค่าในช่วง 0-165 องศา ซึ่งเป็นช่วงของมุม Flexion ของเข่ามนุษย์โดยทั่วไป โดยทำการเก็บผลการทดลองจาก 0-165 องศา โดยเพิ่มทีละ 15 องศา ซึ่งสัมพันธ์กับการทำงานของอุปกรณ์ในการจำกัดมุมงอเข่าที่สามารถปรับการจำกัดได้ที่ละ 15 องศา

3.3 เครื่องมือและวิธีการทดลอง

ใช้ Inertial Measurement Unit (IMU) จำนวน 2 ตัว ในการเก็บค่าการเคลื่อนไหวของขา 2 ส่วน คือ ขาท่อนบน และขาท่อนล่าง ซึ่งข้อมูลที่ได้จะเป็นค่าการเปลี่ยนแปลงมุมของเซนเซอร์ ที่เทียบกับพื้นโลก จากนั้นนำค่าการเปลี่ยนแปลงจากทั้ง 2 เซนเซอร์มาคำนวณหามุมงอของเข่า

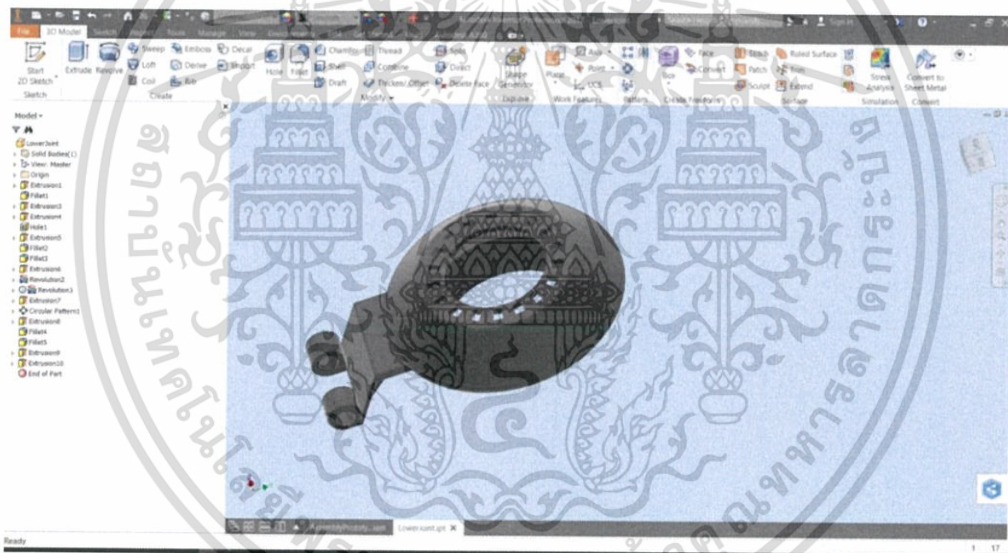
3.4 ขั้นตอนออกแบบและสร้างเครื่องมือ

ในการออกแบบอุปกรณ์แบ่งเป็น 3 ส่วน ดังต่อไปนี้

3.4.1 การออกแบบโครงสร้าง และกลไกภายใน

3.4.1.1 ศึกษาการเคลื่อนไหวของเข้า เพื่อใช้ในการออกแบบจุดหมุน และจุดยึดต่ออุปกรณ์กับร่างกายผู้ป่วย

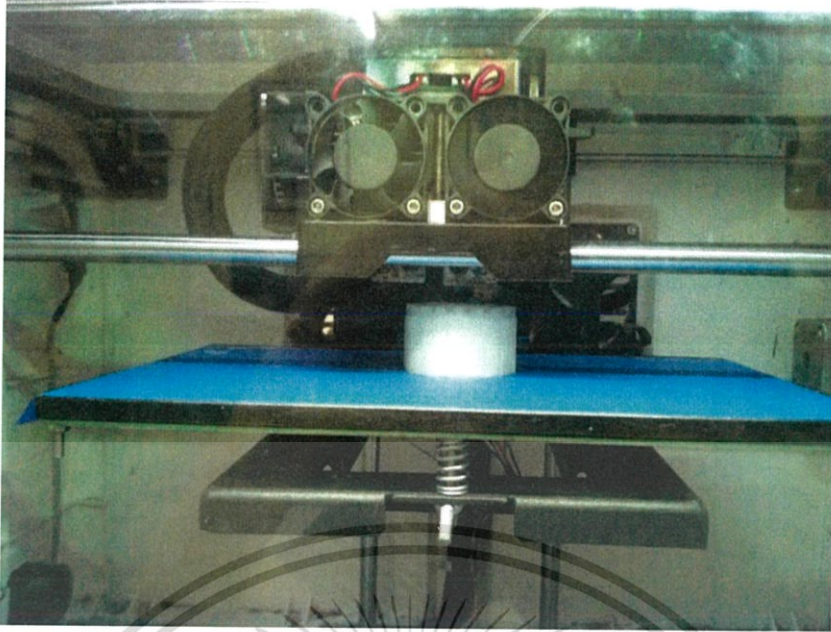
3.4.1.2 ทำการวาดต้นแบบของอุปกรณ์บนกระดาษ ใช้โปรแกรม Autodesk Inventor 2017 ออกแบบโครงสร้าง รวมทั้งใช้โปรแกรมตรวจสอบการทำงานของระบบกลไกที่คิดขึ้น ซึ่งในการออกแบบมีการพัฒนา และปรับปรุงรูปแบบของชิ้นงาน เพื่อถ่ายทอดของขึ้นรูปชิ้นงานซึ่งจะส่งผลถึงราคาการทำชิ้นงาน การวาดแบบของอุปกรณ์ต้นแบบรุ่น 1 นั้นการกำหนดขนาดต่างๆ อ้างอิงจากขนาดของลูกปืน (Bearing) ที่เป็นขนาดมาตรฐานมีจำหน่ายในท้องตลาดทั่วไป โดยเลือกแบบที่มีขนาดเล็กเหมาะสมต่อการใช้งาน และมีน้ำหนักเบา



รูปที่ 3.1 ส่วนประกอบของอุปกรณ์ที่ออกแบบโดย โปรแกรม Autodesk Inventor 2017

3.4.1.3 นำไฟล์ CAD ของอุปกรณ์ที่ออกแบบสมบูรณ์แล้ว นำมาขึ้นรูปเป็นต้นแบบ เพื่อตรวจสอบ และสังเกตการทำงานของกลไกที่คิดขึ้น โดยใช้เครื่องพิมพ์สามมิติ (3D Printer) โดยใช้วัสดุเป็นพลาสติก ABS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 การขึ้นรูปชิ้นงานต้นแบบโดยใช้เครื่องพิมพ์สามมิติ (3D Printer)

3.4.1.4 ประกอบอุปกรณ์ตัวต้นแบบที่ได้จากเครื่องพิมพ์สามมิติ (3D Printer) เพื่อทดสอบการทำงานของกลไกที่ออกแบบ

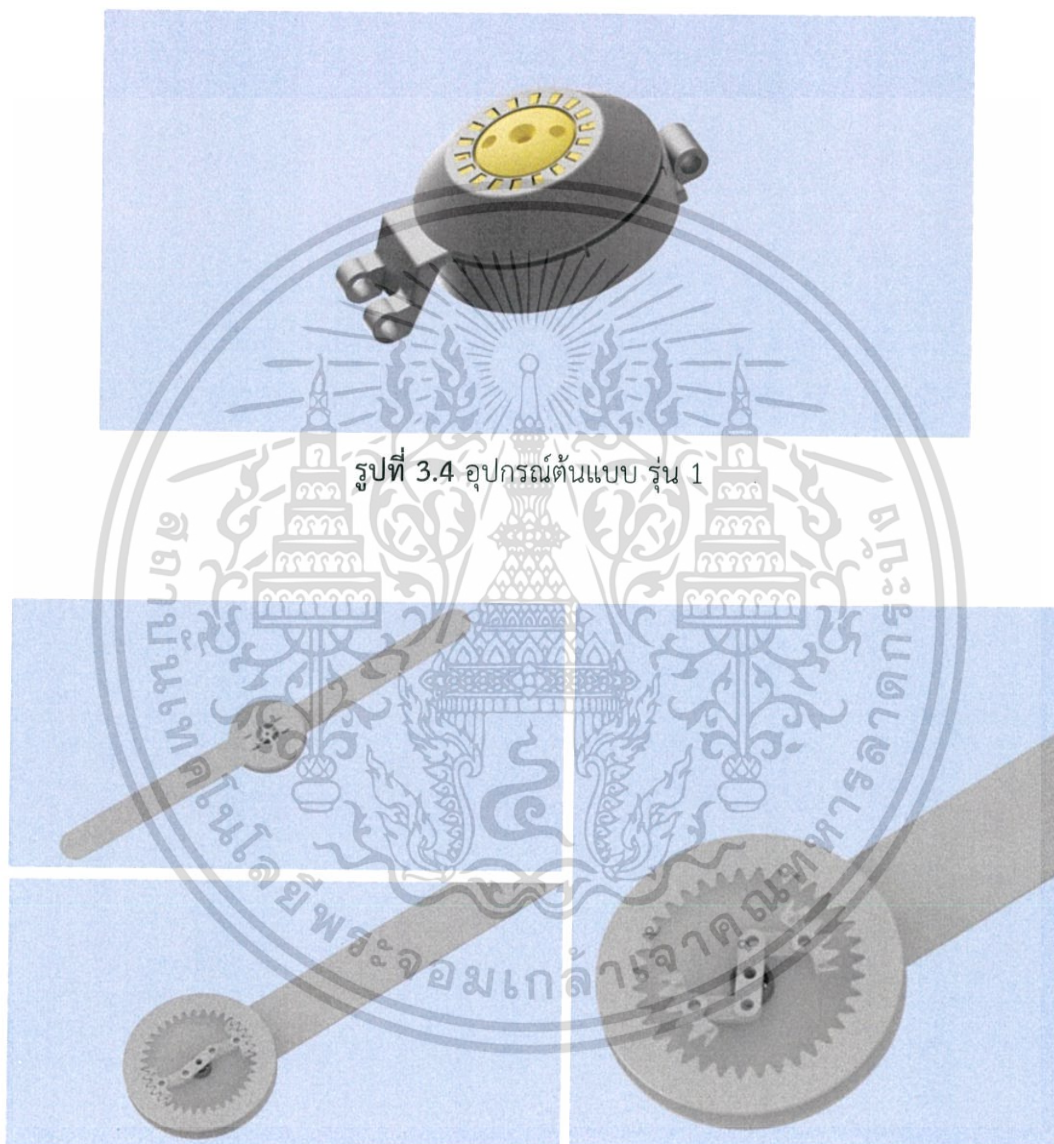


รูปที่ 3.3 ส่วนประกอบของอุปกรณ์ต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

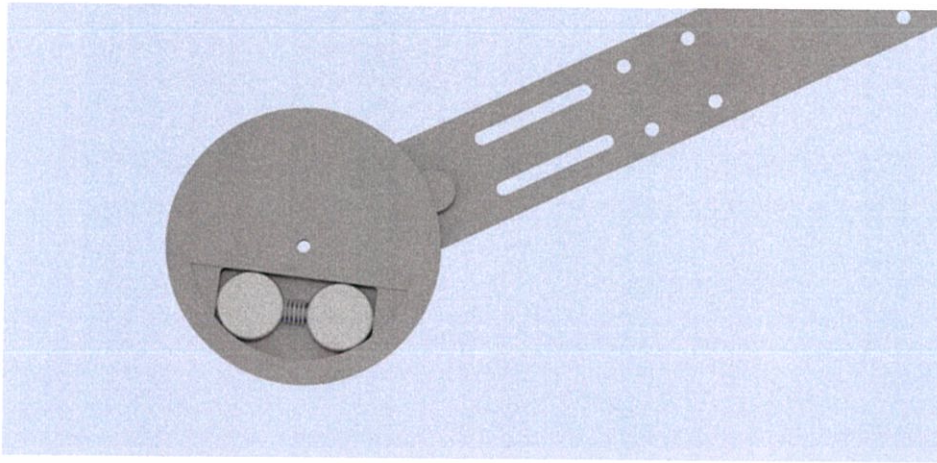
3.4.1.5 ทำการปรับปรุงโครงสร้าง และการออกแบบตัวอุปกรณ์ให้สามารถขึ้นรูปได้ง่ายโดยใช้เครื่องมือที่หาได้ง่าย และกระบวนการผลิตที่มีราคาต่ำ เพื่อประหยัดค่าใช้จ่าย โดยใช้วัสดุเป็นอลูมิเนียม ที่มีน้ำหนักเบาแต่ความแข็งแรง และทนทานต่อการใช้งาน

โดยการออกแบบอุปกรณ์ต้นแบบนั้นมีการพัฒนาออกมา 4 รุ่น เพื่อหาแบบที่สามารถขึ้นรูปชิ้นงานได้ง่าย และประหยัดที่สุด ดังนี้



รูปที่ 3.5 อุปกรณ์ต้นแบบ รุ่น 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 อุปกรณ์ต้นแบบ รุ่น 3

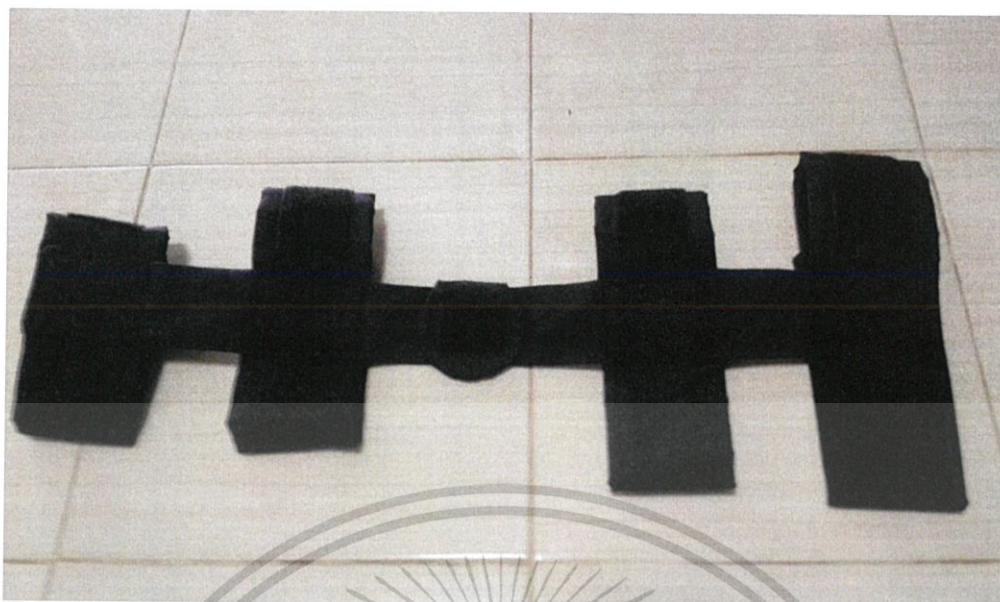


รูปที่ 3.7 อุปกรณ์ต้นแบบ รุ่น 4

ต้นแบบของอุปกรณ์ที่เลือกใช้คือ รุ่น 4 ซึ่งมีขนาดเล็ก และมีน้ำหนักเบา (0.070 กิโลกรัม) โดยวัสดุที่ใช้ในการทำชิ้นงานจริงคือ อลูมิเนียม เนื่องจากเป็นวัสดุที่มีความแข็งแรง และมีน้ำหนักเบา ในการออกแบบอุปกรณ์รุ่น 4 นั้นจะเห็นได้ว่าการพัฒนามาจากอุปกรณ์ต้นแบบรุ่น 1 โดยการลดขนาด และไม่ใส่ลูกปืน (Bearing) เนื่องจากในการเคลื่อนไหวของเข้านั้นมีความเร็วต่ำจึงไม่มีความจำเป็นในการใช้ลูกปืน เพื่อช่วยในการหมุน ซึ่งใช้การ CNC ในการขึ้นรูปชิ้นงาน

3.4.1.6 ออกแบบอุปกรณ์ในส่วนที่จะใช้รัดติดอุปกรณ์กับขาผู้ป่วย โดยใช้ผ้าสีดำ และใส่ผ้าสีดำด้านในเพื่อเพิ่มความนุ่มให้กับผ้า และใช้เวลโคร (velcro) เป็นตัวยึด ซึ่งทำให้สามารถปรับขนาดได้ตามขนาดขาของผู้สวมใส่ เพื่อให้สะดวกสบายต่อผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

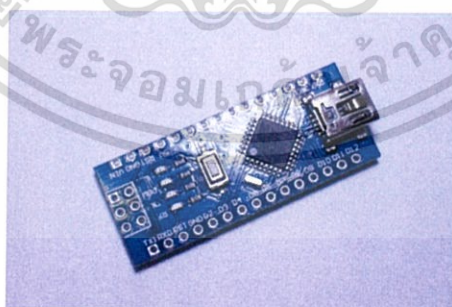


รูปที่ 3.8 แผงรีดชาที่ใส่ติดตั้งอุปกรณ์

3.4.2 การออกแบบวงจรไฟฟ้าในการเก็บข้อมูล

ในการออกแบบวงจรไฟฟ้าเพื่อเก็บข้อมูลของการเคลื่อนไหวของเขานั้น มีส่วนประกอบของชิ้นส่วนอิเล็กทรอนิกส์ และระบบการทำงานในการคำนวณ รับ-ส่งข้อมูล รวมถึงการเก็บบันทึกข้อมูลลงใน Micro SD Card ดังนี้

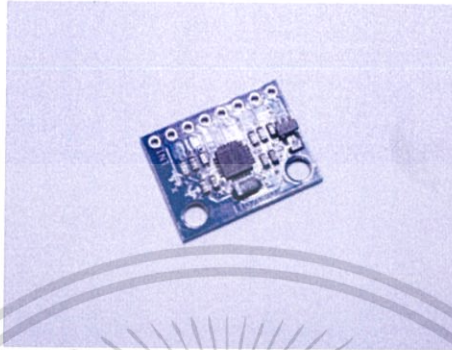
3.4.2.1 Arduino Nano 3.0 ใช้เป็นไมโครคอนโทรลเลอร์ เพื่อประมวลผล และส่งค่ามุมมองของเข่าไปยังตัวรับเพื่อนำไปแสดงผลบนจอแสดงผลในคอมพิวเตอร์



รูปที่ 3.9 Arduino Nano 3.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2.2 GY-521 IMU 3-axis Accelerometer/Gyro Module เป็นโมดูลสำหรับการวัดค่าความเร่งในแนวแกน x, y, z ทั้ง 3 แนวแกน (Accelerometer) รวมถึงการวัดการหมุนรอบแกนใน 3 แนวแกน (Gyroscope) โดยอุปกรณ์ที่ออกแบบจะใช้โมดูล 2 ตัว เพื่อการเคลื่อนไหวของขาที่อ่อน และขาที่อ่อนล่าง จากนั้นนำค่าจากเซ็นเซอร์ทั้ง 2 มาคำนวณหาค่ามุมองศาของผู้ป่วย



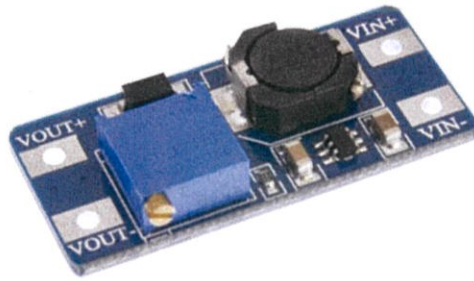
รูปที่ 3.10 GY-521 IMU 3-axis Accelerometer/Gyro Module

3.4.2.3 แหล่งจ่ายพลังงานให้กับระบบอิเล็กทรอนิกส์ทั้งหมด โดยเลือกใช้ Lithium Battery ขนาด 3.7V 400mAh Rechargeable Polymer Li-Po ซึ่งมีขนาดเล็ก และมีการเชื่อมต่อกับ Step Up Booster Power (MT3608) เพื่อปรับค่าแรงดันเป็น 5.5 โวลต์ เพียงพอต่อการจ่ายไฟฟ้าให้ระบบ และมีการเชื่อมต่อกับ TP4056 1A Mini USB Battery Charger Board ใช้ในการชาร์จพลังงานให้แก่แบตเตอรี่

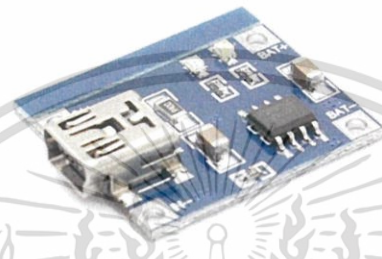


รูปที่ 3.11 Lithium Battery (3.7V 400mAh Rechargeable Polymer Li-Po)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 Step Up Booster Power (MT3608)



รูปที่ 3.13 TP4056 1A Mini USB Battery Charger Board

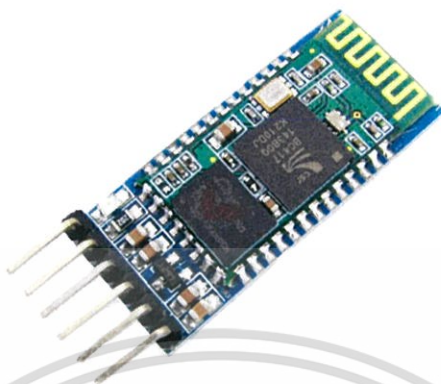
เมื่อนำอุปกรณ์ทั้ง 3 ชิ้นมาติดตั้งเข้าด้วยกัน โดยมีสวิตซ์ที่ใช้ในการสลับการทำงาน ระหว่าง การชาร์จแบตเตอรี่ และการใช้งานแบตเตอรี่ เพื่อความปลอดภัย และออกแบบกล่องเพื่อ บรรจุอุปกรณ์ทั้ง 3 ชิ้น จะได้ดังรูปที่ 3.14



รูปที่ 3.14 ชุด Power Supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2.4 Bluetooth module HC05 โมดูลสำหรับการส่งข้อมูลจากอุปกรณ์ที่เก็บ
ค่ามวงอเข้าของผู้ป่วย ไปยังเครื่องคอมพิวเตอร์เพื่อแสดงผล



รูปที่ 3.15 Bluetooth module HC05

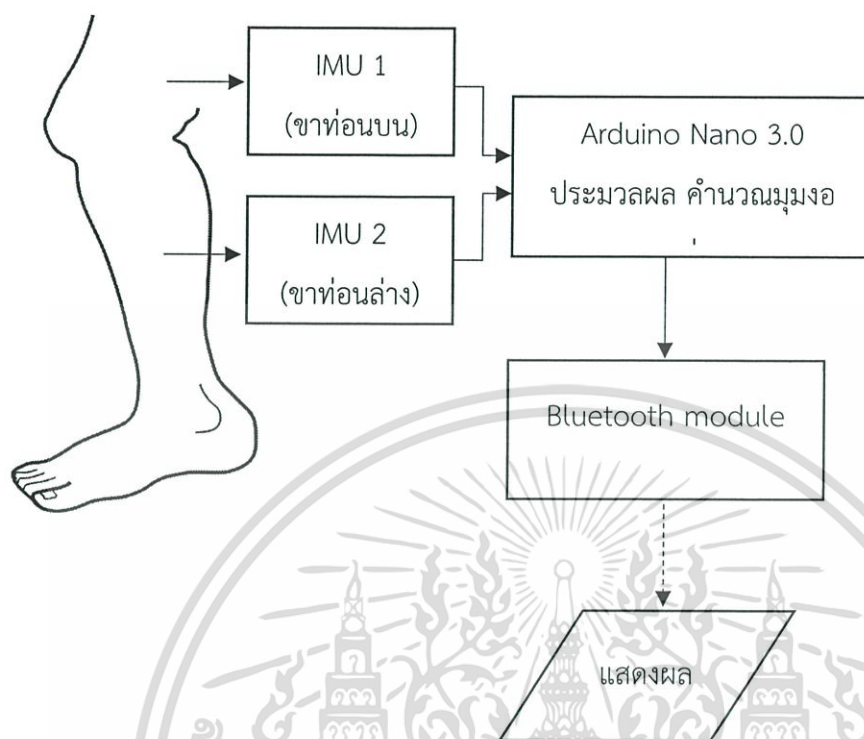
3.4.2.5 ออกแบบแผ่น PCB เพื่อรวมไมโครคอนโทรลเลอร์ และเซนเซอร์เข้าด้วยกัน
โดยออกแบบให้มีขนาดเล็ก เพื่อสะดวกต่อการติดตั้งและการใช้งานอุปกรณ์



รูปที่ 3.16 แผ่น PCB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยหลักการทำงานของอุปกรณ์ จะทำงานตามแผนผังต่อไปนี้



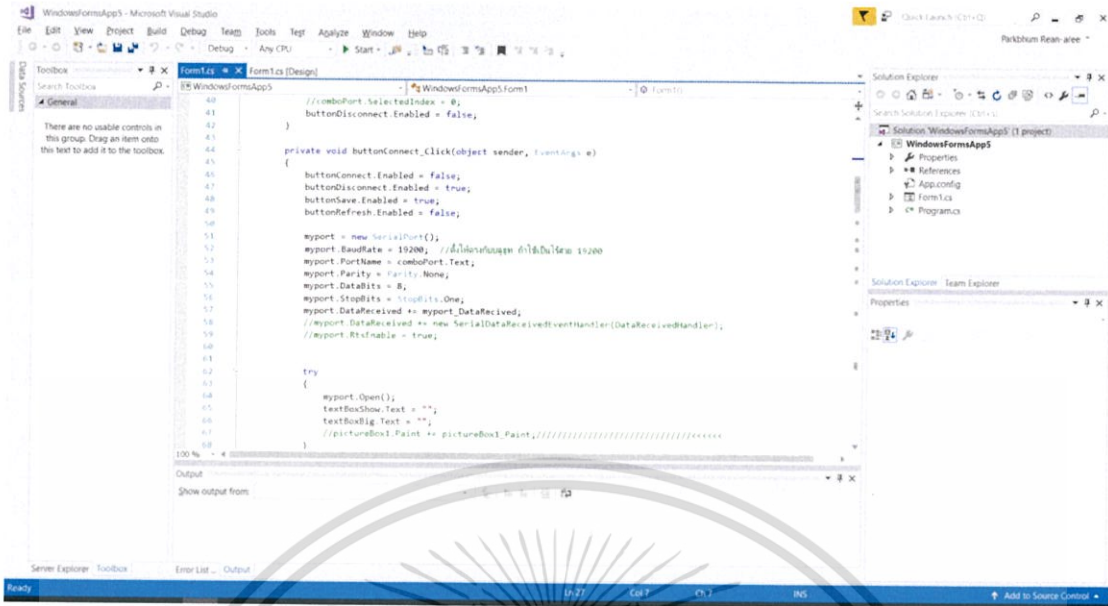
รูปที่ 3.17 แผนผังแสดงการทำงานของอุปกรณ์ ในการวัดมุมองของเขา

นอกจากนี้ การวางตำแหน่งของชิ้นส่วนอุปกรณ์อิเล็กทรอนิกส์ ยังใช้โปรแกรม Autodesk Inventor 2017 ในการออกแบบและตำแหน่งการวางที่เหมาะสมเพื่อให้ได้ชิ้นงานที่มีขนาดเล็ก การเขียนโปรแกรมเพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์ (Arduino Nano 3.0) โดยใช้โปรแกรม Arduino IDE ซึ่งมี Code ที่จะแสดงในภาคผนวกต่อไป

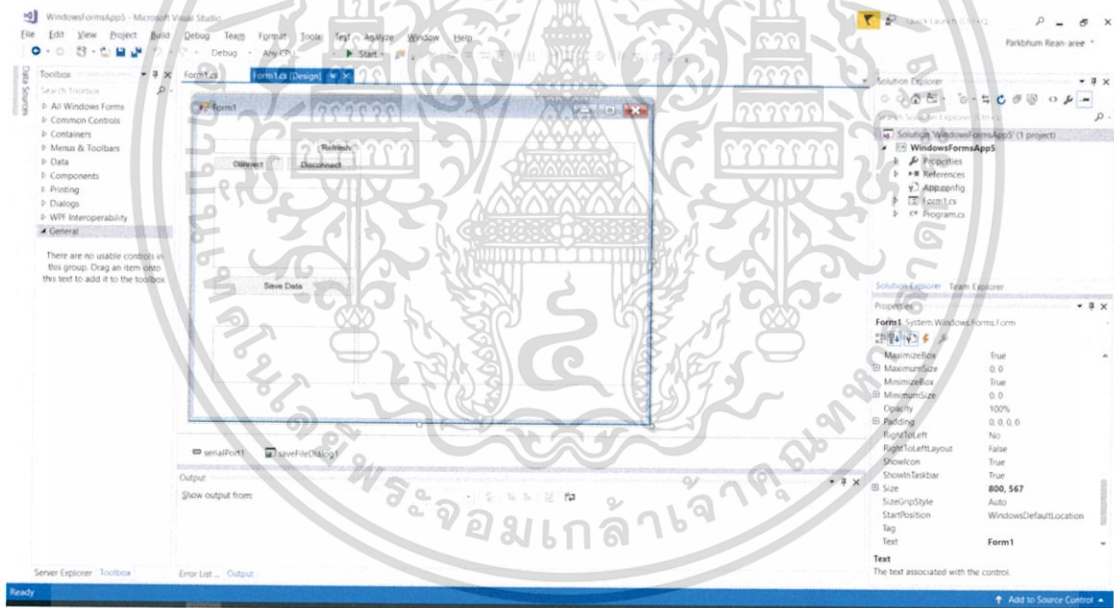
3.4.3 การออกแบบโปรแกรมแสดงผล

ในการออกแบบโปรแกรมแสดงผล ออกแบบและสร้างด้วยโปรแกรม Visual Studio 2017 โดยหน้าต่างแสดงผลจะประกอบไปด้วย การแสดงค่าแบบตัวเลขซึ่งจะบอกองศาการงอเข้าของผู้ป่วยในขณะนั้น มีการแสดงผลเป็นภาพจำลองการงอเข้าของผู้ป่วย อีกทั้งสามารถสร้างระบบบันทึกข้อมูลเพื่อนำมาใช้วิเคราะห์การทำงานของเข้าผู้ป่วยในภายหลังได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 การออกแบบ และเขียนโปรแกรมเพื่อสร้างระบบแสดงผล



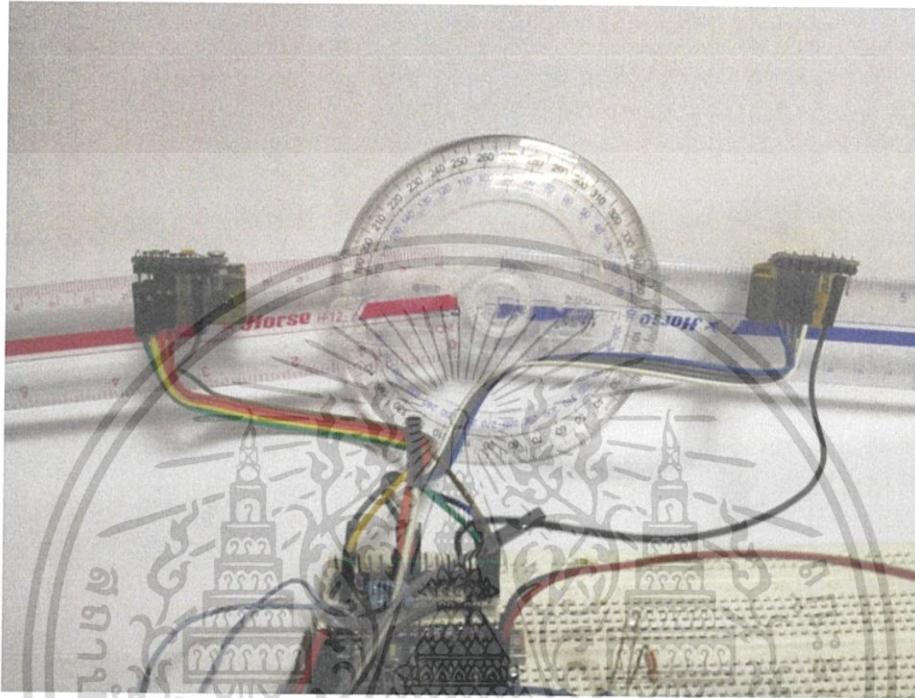
รูปที่ 3.19 การออกแบบหน้าต่างโปรแกรมที่ติดต่อกับผู้ใช้

จากรูปที่ 3.16 และ 3.17 แสดงการออกแบบ และเขียนโปรแกรมเพื่อสร้างระบบแสดงผล และการออกแบบหน้าต่างโปรแกรมที่ติดต่อกับผู้ใช้ โดยโปรแกรมจะมีการติดต่อกับอุปกรณ์จำกัดและวัดมุมองของเช่า ผ่านทาง Serial Port จากการเชื่อมต่อ Bluetooth ระหว่างอุปกรณ์ และคอมพิวเตอร์

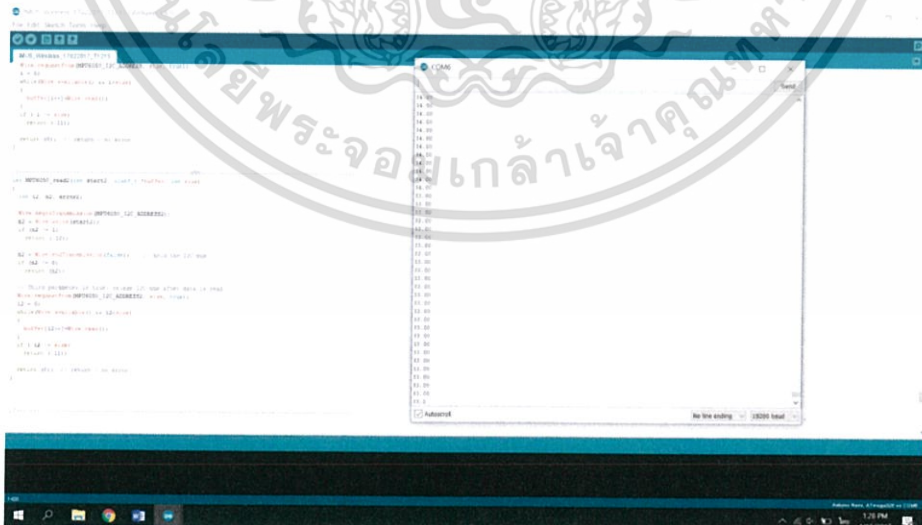
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ขั้นตอนการทดลองการทำงานของอุปกรณ์

การทดลองการวัดมุมของอุปกรณ์ ทำการติดตั้งเซนเซอร์เข้ากับ goniometer และทำการเชื่อมต่อกับคอมพิวเตอร์ผ่าน Serial Port โดยใช้ Bluetooth ทำการทดลองที่มุม 0-165 องศา ซึ่งจะปรับมุมเพิ่มขึ้นทุกๆ 15 องศา



รูปที่ 3.20 การติดตั้งเซนเซอร์เข้ากับ goniometer



รูปที่ 3.21 การเก็บผลจาก Serial Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

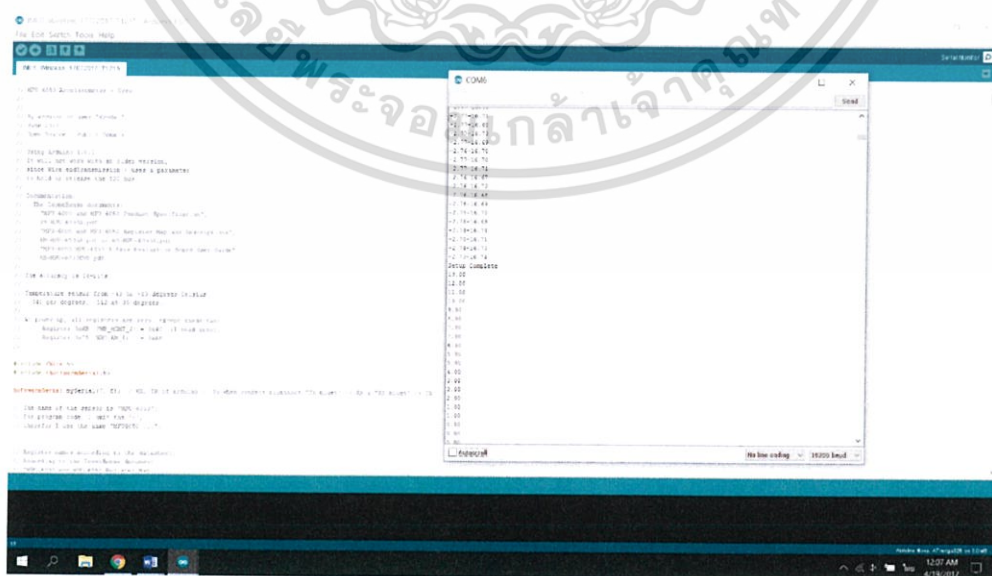
การทดลองและผลการทดลอง

4.1 อุปกรณ์จำกัดและวัดมุงของเข่า

จากการออกแบบอุปกรณ์ และทำการสร้างอุปกรณ์จำกัดและวัดมุงของเข่า และสามารถสวมใส่กับผู้ใช้งานได้โดยสามารถปรับขนาดให้เหมาะสมกับการสวมใส่ของผู้ใช้แต่ละคนได้ ดังรูปที่ 4.1 และในส่วนของกระบวนการการคำนวณเพื่อวัดมุงของเข่านั้น สามารถทำการคำนวณและส่งค่ามุงของเข่าผ่าน Bluetooth ได้ ดังรูปที่ 4.2



รูปที่ 4.1 การติดตั้งชุดอุปกรณ์เข้ากับขาของผู้ป่วย



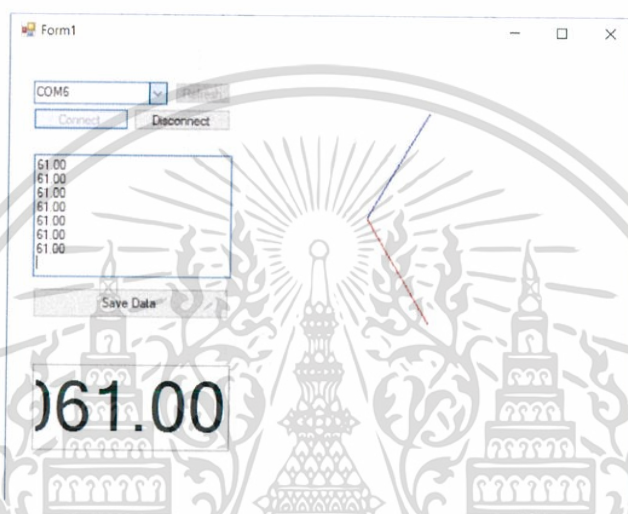
รูปที่ 4.2 การรับค่ามุงของเข่าผ่าน Arduino IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 โปรแกรมแสดงผลการวัดค่ามุมงอเข่า

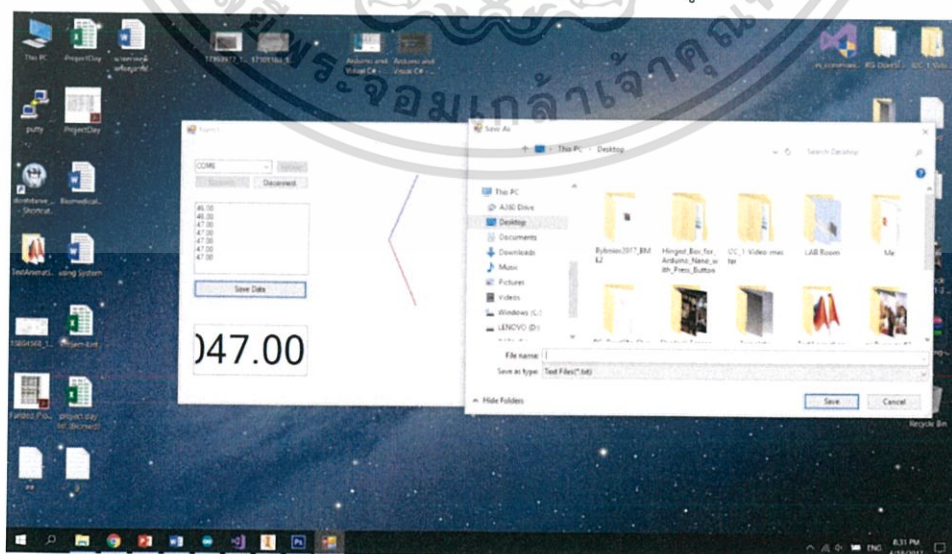
จากการสร้างโปรแกรมสำหรับแสดงผลค่ามุมงอเข่าที่รับจากอุปกรณ์ โดยใช้โปรแกรม Visual Studio 2017 สามารถทำการเชื่อมต่อ และรับค่าจากอุปกรณ์ได้ โดยสามารถแสดงผลมุมงอเข่าของผู้สวมใส่ได้ 2 รูปแบบ ได้แก่ แบบตัวเลขที่แสดงให้เห็นค่ามุมงอเข่าในหน่วย องศา (degree) และแบบภาพเคลื่อนไหว ดังรูปที่ 4.4

จากรูปที่ 4.4 ในส่วนการแสดงผลแบบภาพเคลื่อนไหว เส้นสีน้ำเงินใช้แทนขาที่นอนบน และเส้นสีแดงใช้แทนขาที่นอนล่างของผู้ใช้อุปกรณ์



รูปที่ 4.4 หน้าต่างของโปรแกรมแสดงผล

อีกทั้งโปรแกรมที่สร้างขึ้นมีคำสั่งในการบันทึกข้อมูลของมุมงอเข่า เพื่อใช้ในการวินิจฉัยอาการ การฟื้นตัวของร่างกายผู้ป่วยได้ โดยจะบันทึกออกมาเป็นไฟล์ .txt ดังรูปที่ 4.5



รูปที่ 4.5 การบันทึกข้อมูลของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบการทำงานของกรวัดมุมของเข่า

จากการทดลองการวัดค่ามุมของอุปกรณ์ โดยทำการติดตั้งเซนเซอร์เข้ากับ goniometer และทำการเชื่อมต่อกับคอมพิวเตอร์ผ่าน Serial Port โดยใช้ Bluetooth ทำการทดลองที่มุม 0–165 องศา ซึ่งจะปรับมุมเพิ่มขึ้นทุกๆ 15 องศา โดยทำการทดลองซ้ำทั้งหมด 5 ครั้ง ได้ผลการทดลองตามตารางที่ 4.1

ตารางที่ 4.1 แสดงค่ามุมงอเข่าจากอุปกรณ์จำกัดและวัดมุมงอของเข่า

Angle (degree)	The measured results bent angle of the device					Average (degree)	Error (%)
	Trial 1 (degree)	Trial 2 (degree)	Trial 3 (degree)	Trial 4 (degree)	Trial 5 (degree)		
0	2	6	0	3	4	3	3.00
15	19	19	15	19	15	17.4	16.00
30	33	37	30	32	32	32.8	9.33
45	49	52	42	46	47	47.2	4.89
60	61	68	61	60	65	63	5.00
75	79	75	74	77	76	76.2	1.60
90	91	92	91	91	92	91.4	1.56
105	107	109	103	107	106	106.4	1.33
120	120	124	116	117	124	120.2	0.17
135	133	138	128	132	135	133.2	1.33
150	148	152	147	144	148	147.8	1.47
165	164	166	159	158	160	161.4	2.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากงานวิจัยการออกแบบอุปกรณ์สามารถจำกัดมุมมองของเข้า โดยกลไกที่ออกแบบมานั้นสามารถทำงานได้ ซึ่งสามารถปรับความละเอียดของการจำกัดมุมได้ที่ละ 15 องศา และสามารถคำนวณมุมมองของเข้าได้ถูกต้องแม่นยำ และแสดงผลได้แบบเวลาจริงผ่านระบบไร้สาย โดยมีเปอร์เซ็นต์ความผิดพลาดไม่เกิน 16 เปอร์เซ็นต์ ซึ่งเกิดจากความสามารถของเซนเซอร์ และความคลาดเคลื่อนขณะทำการทดลอง ซึ่งผลที่ได้แสดงให้เห็นว่าการออกแบบอุปกรณ์เป็นประโยชน์ต่อการติดตามอาการผู้ป่วย และใช้ปรับแก้แผนการกายบำบัดให้เหมาะกับการฟื้นตัวของผู้ป่วยแบบเฉพาะรายบุคคลได้ ในการใช้งานอุปกรณ์ยังมีข้อจำกัดในส่วนของระยะห่างระหว่างอุปกรณ์กับคอมพิวเตอร์ต้องไม่เกิน 5 เมตร อุปกรณ์ถูกออกแบบมาสำหรับผู้ใช้ที่มีขนาดเส้นรอบวงของต้นขาและขาท่อนล่าง ที่มีขนาดไม่เกิน 55 เซนติเมตร และ 38 เซนติเมตร ตามลำดับ และอุปกรณ์ต้นแบบชิ้นนี้ใช้สำหรับชาวเขาเท่านั้น

5.2 ปัญหาและแนวทางการแก้ปัญหา

5.2.1 ปัญหาที่พบ

ความถูกต้อง แม่นยำของเซนเซอร์ต่ำ จึงทำให้ผลการทดลองและการใช้งานนั้นมีความผิดพลาดในการวัดค่ามุมมองเข้าเกิดขึ้น และการขึ้นรูปชิ้นงานในส่วนกลไกมีราคาสูง

5.2.2 แนวทางการแก้ปัญหา

เลือกใช้เซนเซอร์ที่มีประสิทธิภาพ ความแม่นยำสูงขึ้น ในส่วนการขึ้นรูปชิ้นงานกลไกนั้นเลือกวัสดุที่มีราคาต่ำ และหาวิธีการขึ้นรูปชิ้นงานที่มีราคาต่ำลง

5.3 ข้อเสนอแนะ

ในการออกแบบชิ้นงานนั้น ต้องคำนึงถึงการใช้งานเป็นหลัก อีกทั้งผู้ออกแบบต้องมีความรู้ความเข้าใจในกระบวนการขึ้นรูป และการเลือกวัสดุ เนื่องจากหากไม่มีความรู้ในส่วนนี้จะทำให้ชิ้นงานอาจขึ้นรูปไม่ได้ หรืออาจมีค่าใช้จ่ายสูงในการขึ้นรูปชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- ____ “Accelerometer” [Online] Available: <http://bme231metrology.blogspot.com/2011/07/accelerometer-iphone-accelerometer.html>. 2554
- ____ “Gyroscope” [Online] Available: <http://zygomatrica.exteen.com/20100811/gyro-sensor-gyroscope>. 2553
- ____ “Inertial measurement unit (IMU) and Global Positioning System (GPS)” [Online] Available: <http://www.slideshare.net/hunchxx/inertial-measurement-unit.2013>
- เฉลิมมพล คงเขียว. 2553. “ระบบภาพเคลื่อนไหวสำหรับการวิเคราะห์การเดินของมนุษย์.” สาขาวิชาวิศวกรรมไฟฟ้า วิศวกรรมศาสตรมหาบัณฑิต.
- ยุทธนันท์ กามอ้อย. “ภาษา C#” [Online] Available: <https://sites.google.com/site/programmingm42/phas-a-c>
- รศ.นพ.สมศักดิ์ คุปต์นิรัติศัยกุล. “กายวิภาคของเข่า” [Online] Available: <http://ortho.md.chula.ac.th/student/SHEET/somsak/3016702.html>. 2559
- รศ.นพ.สมศักดิ์ คุปต์นิรัติศัยกุล. “หลักการดูแลรักษาผู้ป่วยบาดเจ็บบริเวณข้อเข่า” [Online] Available: <http://ortho.md.chula.ac.th/student/SHEET/somsak/3016610.html>. 2559
- รศ.นพ.พิสิฎฐ์ เลิศวานิช. “การรักษาเอ็นเข่าฉีกจากการเล่นกีฬา” [Online] Available: <http://www.thaihealth.or.th/Content/31572/การรักษาเอ็นเข่าฉีกจากการเล่นกีฬา>. 2559
- นพ.วิฑูร บัญณอมวงศ์ “การรักษาการรักษาเส้นเอ็นไขว้หน้าข้อเข่า (ACL) ฉีกขาด” [Online] Available: <https://www.bumrungrad.com/healthspot/May-2015/acl-treatment-orthopedic-center-best-bangkok-thailand>. 2559
- วรพงษ์ ณรงค์วงวัฒนา และพลเชษฐ์ ทนันทนะ. 2558. “การดูแลสุขภาพเชิงรุกด้วยระบบวัดคลื่นสมองอัจฉริยะ.” วิศวกรรมศาสตรบัณฑิต.
- ดร.อลงกรณ์ พิมพ์พิณ. “MEMS : Micro Electro Mechanical Systems” [Online] Available: <http://www.vcharkarn.com/varticle/41941.2010>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Dr. GREG McLATCHIE. Injuries in Combat Sports. London: Typeset by Oxford Publishing Services, 1982

Grant Elliott, Gregory S. Sawicki, Andrew Marecki, and Hugh Herr. : “The Biomechanics and Energetics of Human Running Using an Elastic Knee Exoskeleton”
Electrical Engineering and Computer Science Department Massachusetts Institute of Technology Cambridge, Biomedical Engineering North Carolina State University University of North Carolina Raleigh, and Mechanical Engineering, Department Massachusetts Institute of Technology Cambridge. 2013.

Susan J. Hall. Basic Biomechanics. Singapore: The Publishing Services Group, 1999



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการใช้งานอุปกรณ์จำกัดและวัดมุมของเข่า

1. สวมใส่อุปกรณ์กับขาผู้ป่วย โดยให้จุดหมุนของอุปกรณ์ตรงกับเข่าของผู้สวมใส่



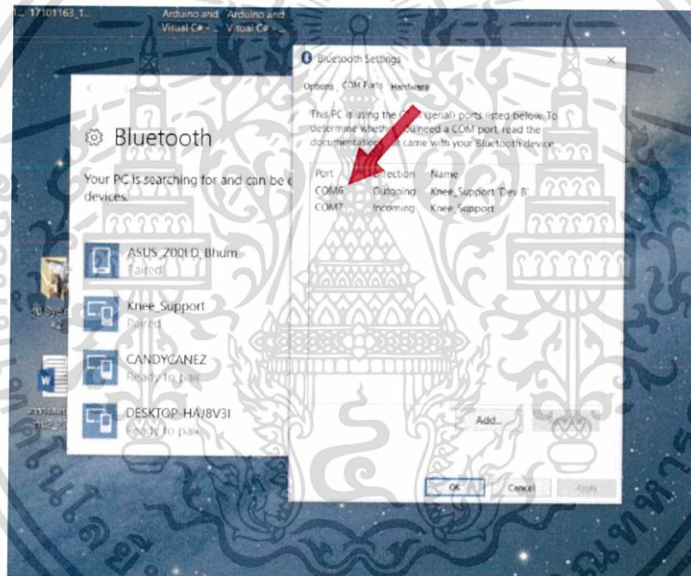
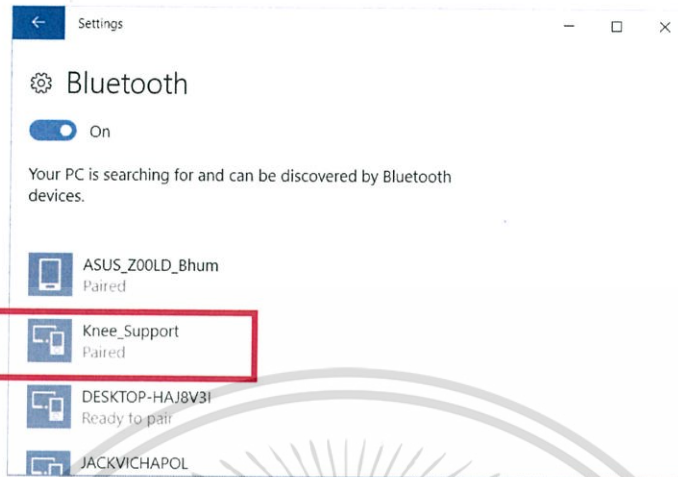
2. ให้ผู้ป่วยยืนตรง หรือนอนเหยียดขาตรง (0 องศา)
3. เปิดการทำงานของอุปกรณ์ และกดปุ่มเพื่อทำการบันทึกค่าเริ่มต้น แล้วจะมีไฟสีแดงสว่างขึ้น เป็นสัญลักษณ์ของการตั้งค่าเริ่มต้นสมบูรณ์แล้ว

ไฟสีแดงสัญลักษณ์ของการตั้งค่าเริ่มต้นสมบูรณ์



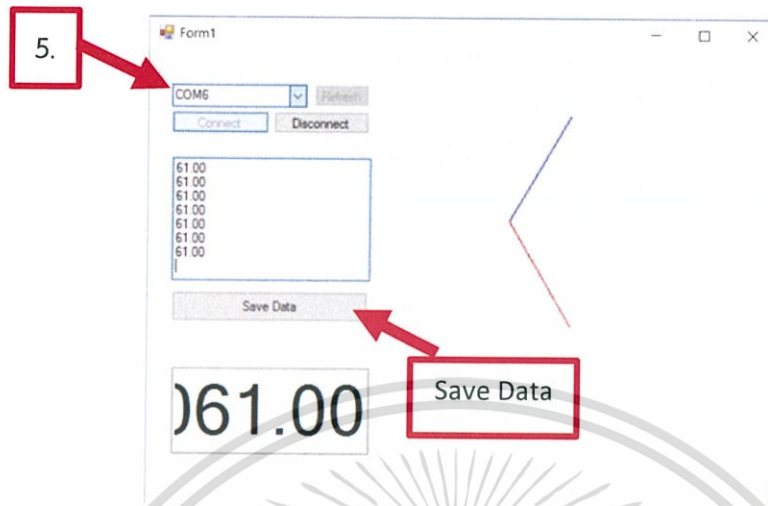
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทำการเชื่อมต่อคอมพิวเตอร์ กับอุปกรณ์จำกัดและวัดมุมของเข่าผ่านทาง Bluetooth โดยชื่อของอุปกรณ์ คือ Knee_Support



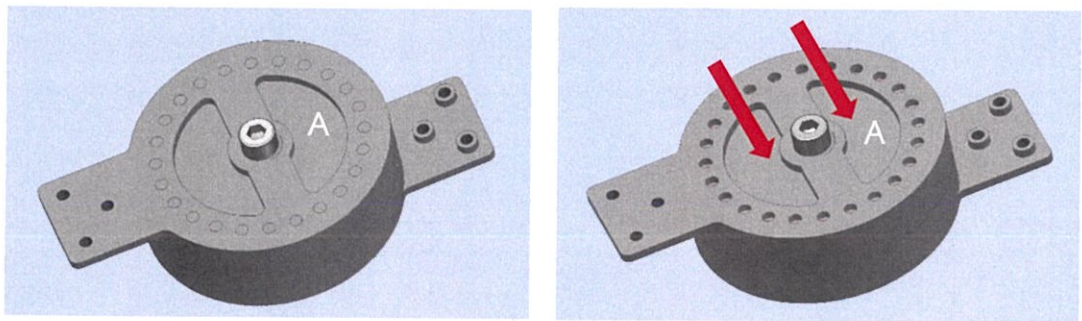
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เปิดโปรแกรมแสดงผล และทำการเลือก Port การเชื่อมต่อ และกดปุ่ม Connect



หากต้องการบันทึกข้อมูล เพียงกดปุ่ม Save Data จากนั้นเลือกตำแหน่งที่ต้องการบันทึก
ไฟล์ในคอมพิวเตอร์ เมื่อบันทึกสำเร็จจะมีกล่องข้อความแสดงว่าการบันทึกสมบูรณ์

ขั้นตอนการปรับการจำกัดองศาการงอเข้า
กดชิ้นส่วน A ลงและหมุน



หากต้องการเพิ่มองศาของการงอเข้าให้ “หมุนตามเข็มนาฬิกา”

หากต้องการลดองศาของการงอเข้าให้ “หมุนทวนเข็มนาฬิกา”



จากนั้นปล่อยมือชิ้นส่วน A จะดันกลับขึ้นมาเอง และทำการล็อกตำแหน่งของการจำกัดองศาการงอเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino Code

```
#include <Wire.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(7, 8); // RX, TX of arduino || So when connect bluetooth "TX bluet" => RX & "RX
bluet" => TX

#define MPU6050_AUX_VDDIO      0x01 // R/W
#define MPU6050_SMPLRT_DIV     0x19 // R/W
#define MPU6050_CONFIG         0x1A // R/W
#define MPU6050_GYRO_CONFIG    0x1B // R/W
#define MPU6050_ACCEL_CONFIG   0x1C // R/W
#define MPU6050_FF_THR         0x1D // R/W
#define MPU6050_FF_DUR         0x1E // R/W
#define MPU6050_MOT_THR        0x1F // R/W
#define MPU6050_MOT_DUR        0x20 // R/W
#define MPU6050_ZRMOT_THR      0x21 // R/W
#define MPU6050_ZRMOT_DUR      0x22 // R/W
#define MPU6050_FIFO_EN        0x23 // R/W
#define MPU6050_I2C_MST_CTRL   0x24 // R/W
#define MPU6050_I2C_SLV0_ADDR  0x25 // R/W
#define MPU6050_I2C_SLV0_REG   0x26 // R/W
#define MPU6050_I2C_SLV0_CTRL  0x27 // R/W
#define MPU6050_I2C_SLV1_ADDR  0x28 // R/W
#define MPU6050_I2C_SLV1_REG   0x29 // R/W
#define MPU6050_I2C_SLV1_CTRL  0x2A // R/W
#define MPU6050_I2C_SLV2_ADDR  0x2B // R/W
#define MPU6050_I2C_SLV2_REG   0x2C // R/W
#define MPU6050_I2C_SLV2_CTRL  0x2D // R/W
#define MPU6050_I2C_SLV3_ADDR  0x2E // R/W
#define MPU6050_I2C_SLV3_REG   0x2F // R/W
#define MPU6050_I2C_SLV3_CTRL  0x30 // R/W
#define MPU6050_I2C_SLV4_ADDR  0x31 // R/W
#define MPU6050_I2C_SLV4_REG   0x32 // R/W
#define MPU6050_I2C_SLV4_DO    0x33 // R/W
#define MPU6050_I2C_SLV4_CTRL  0x34 // R/W
#define MPU6050_I2C_SLV4_DI    0x35 // R
#define MPU6050_I2C_MST_STATUS  0x36 // R
#define MPU6050_INT_PIN_CFG    0x37 // R/W
#define MPU6050_INT_ENABLE     0x38 // R/W
#define MPU6050_INT_STATUS     0x3A // R
#define MPU6050_ACCEL_XOUT_H    0x3B // R
#define MPU6050_ACCEL_XOUT_L    0x3C // R
#define MPU6050_ACCEL_YOUT_H    0x3D // R
#define MPU6050_ACCEL_YOUT_L    0x3E // R
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


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

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

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

```
// Combined definitions for the DLPF_CFG values
```

```
#define MPU6050_DLPF_CFG_0 (0)
#define MPU6050_DLPF_CFG_1 (bit(MPU6050_DLPF_CFG0))
#define MPU6050_DLPF_CFG_2 (bit(MPU6050_DLPF_CFG1))
#define MPU6050_DLPF_CFG_3 (bit(MPU6050_DLPF_CFG1)|bit(MPU6050_DLPF_CFG0))
#define MPU6050_DLPF_CFG_4 (bit(MPU6050_DLPF_CFG2))
#define MPU6050_DLPF_CFG_5 (bit(MPU6050_DLPF_CFG2)|bit(MPU6050_DLPF_CFG0))
#define MPU6050_DLPF_CFG_6 (bit(MPU6050_DLPF_CFG2)|bit(MPU6050_DLPF_CFG1))
#define MPU6050_DLPF_CFG_7
(bit(MPU6050_DLPF_CFG2)|bit(MPU6050_DLPF_CFG1)|bit(MPU6050_DLPF_CFG0))
```

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

```
// This name uses the bandwidth (Hz) for the accelerometer,
```

```
// for the gyro the bandwidth is almost the same.
```

```
#define MPU6050_DLPF_260HZ MPU6050_DLPF_CFG_0
#define MPU6050_DLPF_184HZ MPU6050_DLPF_CFG_1
#define MPU6050_DLPF_94HZ MPU6050_DLPF_CFG_2
#define MPU6050_DLPF_44HZ MPU6050_DLPF_CFG_3
#define MPU6050_DLPF_21HZ MPU6050_DLPF_CFG_4
#define MPU6050_DLPF_10HZ MPU6050_DLPF_CFG_5
#define MPU6050_DLPF_5HZ MPU6050_DLPF_CFG_6
#define MPU6050_DLPF_RESERVED MPU6050_DLPF_CFG_7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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

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

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

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

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

```

```

#define MPU6050_ACCEL_HPF_7
(bit(MPU6050_ACCEL_HPF2)|bit(MPU6050_ACCEL_HPF1)|bit(MPU6050_ACCEL_HPF0))

// Alternative names for the combined definitions
// The name uses the Cut-off frequency.
#define MPU6050_ACCEL_HPF_RESET MPU6050_ACCEL_HPF_0
#define MPU6050_ACCEL_HPF_5HZ MPU6050_ACCEL_HPF_1
#define MPU6050_ACCEL_HPF_2_5HZ MPU6050_ACCEL_HPF_2
#define MPU6050_ACCEL_HPF_1_25HZ MPU6050_ACCEL_HPF_3
#define MPU6050_ACCEL_HPF_0_63HZ MPU6050_ACCEL_HPF_4
#define MPU6050_ACCEL_HPF_HOLD MPU6050_ACCEL_HPF_7

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

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

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

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

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define MPU6050_I2C_MST_P_NSR MPU6050_D4
#define MPU6050_SLV_3_FIFO_EN MPU6050_D5
#define MPU6050_WAIT_FOR_ES MPU6050_D6
#define MPU6050_MULT_MST_EN MPU6050_D7

// Combined definitions for the I2C_MST_CLK
#define MPU6050_I2C_MST_CLK_0 (0)
#define MPU6050_I2C_MST_CLK_1 (bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_2 (bit(MPU6050_I2C_MST_CLK1))
#define MPU6050_I2C_MST_CLK_3 (bit(MPU6050_I2C_MST_CLK1)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_4 (bit(MPU6050_I2C_MST_CLK2))
#define MPU6050_I2C_MST_CLK_5 (bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_6 (bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK1))
#define MPU6050_I2C_MST_CLK_7
(bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK1)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_8 (bit(MPU6050_I2C_MST_CLK3))
#define MPU6050_I2C_MST_CLK_9 (bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_10 (bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK1))
#define MPU6050_I2C_MST_CLK_11
(bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK1)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_12 (bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK2))
#define MPU6050_I2C_MST_CLK_13
(bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK0))
#define MPU6050_I2C_MST_CLK_14
(bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK1))
#define MPU6050_I2C_MST_CLK_15
(bit(MPU6050_I2C_MST_CLK3)|bit(MPU6050_I2C_MST_CLK2)|bit(MPU6050_I2C_MST_CLK1)|bit(MPU6050_I2C_
MST_CLK0))

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

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define MPU6050_I2C_MST_CLK_400KHZ MPU6050_I2C_MST_CLK_13
#define MPU6050_I2C_MST_CLK_381KHZ MPU6050_I2C_MST_CLK_14
#define MPU6050_I2C_MST_CLK_364KHZ MPU6050_I2C_MST_CLK_15
```

```
// I2C_SLV0_ADDR Register
```

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

```
// Use these only with the bit() macro.
```

```
#define MPU6050_I2C_SLV0_RW MPU6050_D7
```

```
// I2C_SLV0_CTRL Register
```

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

```
// Use these only with the bit() macro.
```

```
#define MPU6050_I2C_SLV0_LEN0 MPU6050_D0
```

```
#define MPU6050_I2C_SLV0_LEN1 MPU6050_D1
```

```
#define MPU6050_I2C_SLV0_LEN2 MPU6050_D2
```

```
#define MPU6050_I2C_SLV0_LEN3 MPU6050_D3
```

```
#define MPU6050_I2C_SLV0_GRP MPU6050_D4
```

```
#define MPU6050_I2C_SLV0_REG_DIS MPU6050_D5
```

```
#define MPU6050_I2C_SLV0_BYTE_SW MPU6050_D6
```

```
#define MPU6050_I2C_SLV0_EN MPU6050_D7
```

```
// A mask for the length
```

```
#define MPU6050_I2C_SLV0_LEN_MASK 0x0F
```

```
// I2C_SLV1_ADDR Register
```

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

```
// Use these only with the bit() macro.
```

```
#define MPU6050_I2C_SLV1_RW MPU6050_D7
```

```
// I2C_SLV1_CTRL Register
```

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

```
// Use these only with the bit() macro.
```

```
#define MPU6050_I2C_SLV1_LEN0 MPU6050_D0
```

```
#define MPU6050_I2C_SLV1_LEN1 MPU6050_D1
```

```
#define MPU6050_I2C_SLV1_LEN2 MPU6050_D2
```

```
#define MPU6050_I2C_SLV1_LEN3 MPU6050_D3
```

```
#define MPU6050_I2C_SLV1_GRP MPU6050_D4
```

```
#define MPU6050_I2C_SLV1_REG_DIS MPU6050_D5
```

```
#define MPU6050_I2C_SLV1_BYTE_SW MPU6050_D6
```

```
#define MPU6050_I2C_SLV1_EN MPU6050_D7
```

```
// A mask for the length
```

```
#define MPU6050_I2C_SLV1_LEN_MASK 0x0F
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// I2C_SLV2_ADDR Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_I2C_SLV2_RW MPU6050_D7

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

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

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

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

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

```

```

// I2C_SLV4_ADDR Register

```

```

// These are the names for the bits.

```

```

// Use these only with the bit() macro.

```

```

#define MPU6050_I2C_SLV4_RW MPU6050_D7

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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

```

```

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

```

```

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

```

```

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

```

```

// INT_ENABLE Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_DATA_RDY_EN   MPU6050_D0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define MPU6050_I2C_MST_INT_EN MPU6050_D3
#define MPU6050_FIFO_OFLOW_EN MPU6050_D4
#define MPU6050_ZMOT_EN MPU6050_D5
#define MPU6050_MOT_EN MPU6050_D6
#define MPU6050_FF_EN MPU6050_D7

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

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

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

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

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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

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

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

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

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

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

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

```

```

// USER_CTRL Register

```

```

// These are the names for the bits.

```

```

// Use these only with the bit() macro.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define MPU6050_SIG_COND_RESET MPU6050_D0
#define MPU6050_I2C_MST_RESET MPU6050_D1
#define MPU6050_FIFO_RESET MPU6050_D2
#define MPU6050_I2C_IF_DIS MPU6050_D4 // must be 0 for MPU-6050
#define MPU6050_I2C_MST_EN MPU6050_D5
#define MPU6050_FIFO_EN MPU6050_D6

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

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

// Alternative names for the combined definitions
#define MPU6050_CLKSEL_INTERNAL MPU6050_CLKSEL_0
#define MPU6050_CLKSEL_X MPU6050_CLKSEL_1
#define MPU6050_CLKSEL_Y MPU6050_CLKSEL_2
#define MPU6050_CLKSEL_Z MPU6050_CLKSEL_3
#define MPU6050_CLKSEL_EXT_32KHZ MPU6050_CLKSEL_4
#define MPU6050_CLKSEL_EXT_19_2MHZ MPU6050_CLKSEL_5
#define MPU6050_CLKSEL_RESERVED MPU6050_CLKSEL_6
#define MPU6050_CLKSEL_STOP MPU6050_CLKSEL_7

// PWR_MGMT_2 Register
// These are the names for the bits.
// Use these only with the bit() macro.
#define MPU6050_STBY_ZG MPU6050_D0
#define MPU6050_STBY_YG MPU6050_D1
#define MPU6050_STBY_XG MPU6050_D2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

uint8_t y_accel_l;
uint8_t z_accel_h;
uint8_t z_accel_l;
uint8_t t_h;
uint8_t t_l;
uint8_t x_gyro_h;
uint8_t x_gyro_l;
uint8_t y_gyro_h;
uint8_t y_gyro_l;
uint8_t z_gyro_h;
uint8_t z_gyro_l;
} reg;
struct
{
    int x_accel;
    int y_accel;
    int z_accel;
    int temperature;
    int x_gyro;
    int y_gyro;
    int z_gyro;
} value;
};

//-----VW-----//
typedef union accel_t_gyro_union2
{
    struct
    {
        uint8_t x_accel2_h;
        uint8_t x_accel2_l;
        uint8_t y_accel2_h;
        uint8_t y_accel2_l;
        uint8_t z_accel2_h;
        uint8_t z_accel2_l;
        uint8_t t2_h;
        uint8_t t2_l;
        uint8_t x_gyro2_h;
        uint8_t x_gyro2_l;
        uint8_t y_gyro2_h;
        uint8_t y_gyro2_l;
        uint8_t z_gyro2_h;
        uint8_t z_gyro2_l;
    } reg2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

struct
{
    int x_accel2;
    int y_accel2;
    int z_accel2;
    int temperature2;
    int x_gyro2;
    int y_gyro2;
    int z_gyro2;
} value2;
};

//-----^^^-----//

// Use the following global variables and access functions to help store the overall
// rotation angle of the sensor
unsigned long last_read_time;
float    last_x_angle; // These are the filtered angles
float    last_y_angle;
float    last_z_angle;
float    last_gyro_x_angle; // Store the gyro angles to compare drift
float    last_gyro_y_angle;
float    last_gyro_z_angle;

//-----VVV-----//

unsigned long last_read_time2;
float    last_x_angle2; // These are the filtered angles
float    last_y_angle2;
float    last_z_angle2;
float    last_gyro_x_angle2; // Store the gyro angles to compare drift
float    last_gyro_y_angle2;
float    last_gyro_z_angle2;

//-----^^^-----//

void set_last_read_angle_data(unsigned long time, float x, float y, float z, float x_gyro, float y_gyro, float
z_gyro) {
    last_read_time = time;
    last_x_angle = x;
    last_y_angle = y;
    last_z_angle = z;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

last_gyro_x_angle = x_gyro;
last_gyro_y_angle = y_gyro;
last_gyro_z_angle = z_gyro;
}

//-----VVV-----//
void set_last_read_angle_data2(unsigned long time, float x2, float y2, float z2, float x_gyro2, float y_gyro2,
float z_gyro2) {
last_read_time2 = time;
last_x_angle2 = x2;
last_y_angle2 = y2;
last_z_angle2 = z2;
last_gyro_x_angle2 = x_gyro2;
last_gyro_y_angle2 = y_gyro2;
last_gyro_z_angle2 = z_gyro2;
}

//-----^^^-----//

inline unsigned long get_last_time() {return last_read_time;}
inline float get_last_x_angle() {return last_x_angle;}
inline float get_last_y_angle() {return last_y_angle;}
inline float get_last_z_angle() {return last_z_angle;}
inline float get_last_gyro_x_angle() {return last_gyro_x_angle;}
inline float get_last_gyro_y_angle() {return last_gyro_y_angle;}
inline float get_last_gyro_z_angle() {return last_gyro_z_angle;}

//-----VVV-----//

inline unsigned long get_last_time2() {return last_read_time2;}
inline float get_last_x_angle2() {return last_x_angle2;}
inline float get_last_y_angle2() {return last_y_angle2;}
inline float get_last_z_angle2() {return last_z_angle2;}
inline float get_last_gyro_x_angle2() {return last_gyro_x_angle2;}
inline float get_last_gyro_y_angle2() {return last_gyro_y_angle2;}
inline float get_last_gyro_z_angle2() {return last_gyro_z_angle2;}
//-----^^^-----//

// Use the following global variables and access functions
// to calibrate the acceleration sensor
float base_x_accel;
float base_y_accel;
float base_z_accel;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float base_x_gyro;
float base_y_gyro;
float base_z_gyro;

//-----VVV-----//
float base_x_accel2;
float base_y_accel2;
float base_z_accel2;

float base_x_gyro2;
float base_y_gyro2;
float base_z_gyro2;
//-----^^^-----//

int read_gyro_accel_vals(uint8_t* accel_t_gyro_ptr) {
    // Read the raw values.
    // Read 14 bytes at once,
    // containing acceleration, temperature and gyro.
    // With the default settings of the MPU-6050,
    // there is no filter enabled, and the values
    // are not very stable. Returns the error value

    accel_t_gyro_union* accel_t_gyro = (accel_t_gyro_union *) accel_t_gyro_ptr;

    int error = MPU6050_read (MPU6050_ACCEL_XOUT_H, (uint8_t *) accel_t_gyro, sizeof(*accel_t_gyro));

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

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

    return error;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//-----VVV-----//

int read_gyro_accel_vals2(uint8_t* accel_t_gyro_ptr2) {

    accel_t_gyro_union2* accel_t_gyro2 = (accel_t_gyro_union2 *) accel_t_gyro_ptr2;

    int error2 = MPU6050_read2(MPU6050_ACCEL_XOUT_H, (uint8_t *) accel_t_gyro2, sizeof(*accel_t_gyro2));
    //////////////////////////////////////

    uint8_t swap2;
    #define SWAP2(x2,y2) swap2 = x2; x2 = y2; y2 = swap2

    SWAP2((*accel_t_gyro2).reg2.x_accel2_h, (*accel_t_gyro2).reg2.x_accel2_l);
    SWAP2 ((*accel_t_gyro2).reg2.y_accel2_h, (*accel_t_gyro2).reg2.y_accel2_l);
    SWAP2 ((*accel_t_gyro2).reg2.z_accel2_h, (*accel_t_gyro2).reg2.z_accel2_l);
    SWAP2 ((*accel_t_gyro2).reg2.t2_h, (*accel_t_gyro2).reg2.t2_l);
    SWAP2 ((*accel_t_gyro2).reg2.x_gyro2_h, (*accel_t_gyro2).reg2.x_gyro2_l);
    SWAP2 ((*accel_t_gyro2).reg2.y_gyro2_h, (*accel_t_gyro2).reg2.y_gyro2_l);
    SWAP2 ((*accel_t_gyro2).reg2.z_gyro2_h, (*accel_t_gyro2).reg2.z_gyro2_l);

    return error2;
}

//-----^^^-----//
// The sensor should be motionless on a horizontal surface
// while calibration is happening
void calibrate_sensors() {
    int          num_readings = 10;
    float        x_accel = 0;
    float        y_accel = 0;
    float        z_accel = 0;
    float        x_gyro = 0;
    float        y_gyro = 0;
    float        z_gyro = 0;
    accel_t_gyro_union  accel_t_gyro;
    //Serial.println("Starting Calibration");
    // Discard the first set of values read from the IMU
    read_gyro_accel_vals((uint8_t *) &accel_t_gyro);
    // Read and average the raw values from the IMU
    for (int i = 0; i < num_readings; i++) {
        read_gyro_accel_vals((uint8_t *) &accel_t_gyro);
        x_accel += accel_t_gyro.value.x_accel;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับวิชาการซึ่งงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    y_accel += accel_t_gyro.value.y_accel;
    z_accel += accel_t_gyro.value.z_accel;
    x_gyro += accel_t_gyro.value.x_gyro;
    y_gyro += accel_t_gyro.value.y_gyro;
    z_gyro += accel_t_gyro.value.z_gyro;
    delay(100);
}

x_accel /= num_readings;
y_accel /= num_readings;
z_accel /= num_readings;
x_gyro /= num_readings;
y_gyro /= num_readings;
z_gyro /= num_readings;

// Store the raw calibration values globally
base_x_accel = x_accel;
base_y_accel = y_accel;
base_z_accel = z_accel;
base_x_gyro = x_gyro;
base_y_gyro = y_gyro;
base_z_gyro = z_gyro;

//Serial.println("Finishing Calibration");
}
//-----VV-----//

void calibrate_sensors2() {
    int        num_readings2 = 10;
    float      x_accel2 = 0;
    float      y_accel2 = 0;
    float      z_accel2 = 0;
    float      x_gyro2 = 0;
    float      y_gyro2 = 0;
    float      z_gyro2 = 0;
    accel_t_gyro_union2  accel_t_gyro2;
    read_gyro_accel_vals2((uint8_t *) &accel_t_gyro2);
    // Read and average the raw values from the IMU
    for (int i = 0; i < num_readings2; i++) {
        read_gyro_accel_vals2((uint8_t *) &accel_t_gyro2);
        x_accel2 += accel_t_gyro2.value2.x_accel2;
        y_accel2 += accel_t_gyro2.value2.y_accel2;
        z_accel2 += accel_t_gyro2.value2.z_accel2;
        x_gyro2 += accel_t_gyro2.value2.x_gyro2;
        y_gyro2 += accel_t_gyro2.value2.y_gyro2;
        z_gyro2 += accel_t_gyro2.value2.z_gyro2;
    }
}

```



```

float accel_x = accel_t_gyro.value.x_accel;
float accel_y = accel_t_gyro.value.y_accel;
float accel_z = accel_t_gyro.value.z_accel;

// Get angle values from accelerometer
float RADIANS_TO_DEGREES = 180/3.14159;
// float accel_vector_length = sqrt(pow(accel_x,2) + pow(accel_y,2) + pow(accel_z,2));
float accel_angle_y = atan(-1*accel_x/sqrt(pow(accel_y,2) + pow(accel_z,2)))*RADIANS_TO_DEGREES;
float accel_angle_x = atan(accel_y/sqrt(pow(accel_x,2) + pow(accel_z,2)))*RADIANS_TO_DEGREES;

float accel_angle_z = 0;

// Compute the (filtered) gyro angles
float dt =(t_now - get_last_time())/1000.0;
float gyro_angle_x = gyro_x*dt + get_last_x_angle();
float gyro_angle_y = gyro_y*dt + get_last_y_angle();
float gyro_angle_z = gyro_z*dt + get_last_z_angle();

// Compute the drifting gyro angles
float unfiltered_gyro_angle_x = gyro_x*dt + get_last_gyro_x_angle();
float unfiltered_gyro_angle_y = gyro_y*dt + get_last_gyro_y_angle();
float unfiltered_gyro_angle_z = gyro_z*dt + get_last_gyro_z_angle();

// Apply the complementary filter to figure out the change in angle - choice of alpha is
// estimated now. Alpha depends on the sampling rate...
float alpha = 0.96;
float angle_x = alpha*gyro_angle_x + (1.0 - alpha)*accel_angle_x;
float angle_y = alpha*gyro_angle_y + (1.0 - alpha)*accel_angle_y;
float angle_z = gyro_angle_z; //Accelerometer doesn't give z-angle

// Update the saved data with the latest values
set_last_read_angle_data(t_now, angle_x, angle_y, angle_z, unfiltered_gyro_angle_x,
unfiltered_gyro_angle_y, unfiltered_gyro_angle_z);

//-----VVV-----//
float accel_x2 = accel_t_gyro2.value2.x_accel2;
float accel_y2 = accel_t_gyro2.value2.y_accel2;
float accel_z2 = accel_t_gyro2.value2.z_accel2;
// Get angle values from accelerometer
float RADIANS_TO_DEGREES2 = 180/3.14159;
// float accel_vector_length = sqrt(pow(accel_x,2) + pow(accel_y,2) + pow(accel_z,2));
float accel_angle_y2 = atan(-1*accel_x2/sqrt(pow(accel_y2,2) +
pow(accel_z2,2)))*RADIANS_TO_DEGREES2;
float accel_angle_x2 = atan(accel_y2/sqrt(pow(accel_x2,2) + pow(accel_z2,2)))*RADIANS_TO_DEGREES2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับผู้ใดที่นำไปใช้โดยไม่ได้รับอนุญาต
 ไม่ว่าการณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float accel_angle_z2 = 0;
// Compute the (filtered) gyro angles
//float dt =(t_now - get_last_time())/1000.0;
float gyro_angle_x2 = gyro_x2*dt + get_last_x_angle2();
float gyro_angle_y2 = gyro_y2*dt + get_last_y_angle2();
float gyro_angle_z2 = gyro_z2*dt + get_last_z_angle2();

// Compute the drifting gyro angles
float unfiltered_gyro_angle_x2 = gyro_x2*dt + get_last_gyro_x_angle2();
float unfiltered_gyro_angle_y2 = gyro_y2*dt + get_last_gyro_y_angle2();
float unfiltered_gyro_angle_z2 = gyro_z2*dt + get_last_gyro_z_angle2();

// Apply the complementary filter to figure out the change in angle - choice of alpha is
// estimated now. Alpha depends on the sampling rate...
float alpha2 = 0.96;
float angle_x2 = alpha*gyro_angle_x2 + (1.0 - alpha2)*accel_angle_x2 ;
float angle_y2 = alpha*gyro_angle_y2+ (1.0 - alpha2)*accel_angle_y2 ;
float angle_z2 = gyro_angle_z2; //Accelerometer doesn't give z-angle

// Update the saved data with the latest values
set_last_read_angle_data2(t_now, angle_x2, angle_y2, angle_z2, unfiltered_gyro_angle_x2,
unfiltered_gyro_angle_y2, unfiltered_gyro_angle_z2);
//-----^-----//
float KneeAngle;
//-----//

buttonState = digitalRead(buttonPin);
if (SetupState == 0)
{
  Serial.print(angle_x);
  Serial.println(angle_x2);
  delay(10);

if (buttonState == HIGH)
{
  digitalWrite(ledPin, HIGH);
  initialX1=angle_x;
  initialX2=angle_x2;
  Serial.println("Setup Complete");
  delay(300);
  SetupState=1;
}
} else
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DataX1 = initialX1 - angle_x;
DataX2 = initialX2 - angle_x2;

KneeAngle = abs(DataX1-DataX2);

Serial.println(KneeAngle);

}
} //---End void loop---//
//-----
// MPU6050_read
int MPU6050_read(int start, uint8_t *buffer, int size)
{
  int i, n, error;

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

  n = Wire.endTransmission(false); // hold the I2C-bus

  if (n != 0)
    return (n);

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

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

//-----VVV-----//
int MPU6050_read2(int start2, uint8_t *buffer, int size)
{
  int i2, n2, error2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Wire.beginTransmission(MPU6050_I2C_ADDRESS2);
n2 = Wire.write(start2);
if (n2 != 1)
    return (-10);

n2 = Wire.endTransmission(false); // hold the I2C-bus
if (n2 != 0)
    return (n2);

// Third parameter is true: release I2C-bus after data is read.
Wire.requestFrom(MPU6050_I2C_ADDRESS2, size, true);
i2 = 0;
while(Wire.available() && i2<size)
{
    buffer[i2++]=Wire.read();
}
if ( i2 != size)
    return (-11);

return (0); // return : no error
}
//-----^-----//
//-----^-----//
// MPU6050_write
int MPU6050_write(int start, const uint8_t *pData, int size)
{
    int n, error;

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

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

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

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

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//-----VVV-----//
```

```
int MPU6050_write2(int start2, const uint8_t *pData2, int size)
```

```
{  
    int n2, error2;  
  
    Wire.beginTransaction(MPU6050_I2C_ADDRESS2);  
    n2 = Wire.write(start2);    // write the start address  
    if (n2 != 1)  
        return (-20);  
  
    n2 = Wire.write(pData2, size); // write data bytes  
    if (n2 != size)  
        return (-21);  
  
    error2 = Wire.endTransmission(true); // release the I2C-bus  
    if (error2 != 0)  
        return (error2);  
  
    return (0);    // return : no error  
}
```

```
//-----^^^-----//
```

```
// MPU6050_write_reg
```

```
int MPU6050_write_reg(int reg, uint8_t data)
```

```
{  
    int error;  
  
    error = MPU6050_write(reg, &data, 1);  
  
    return (error);  
}
```

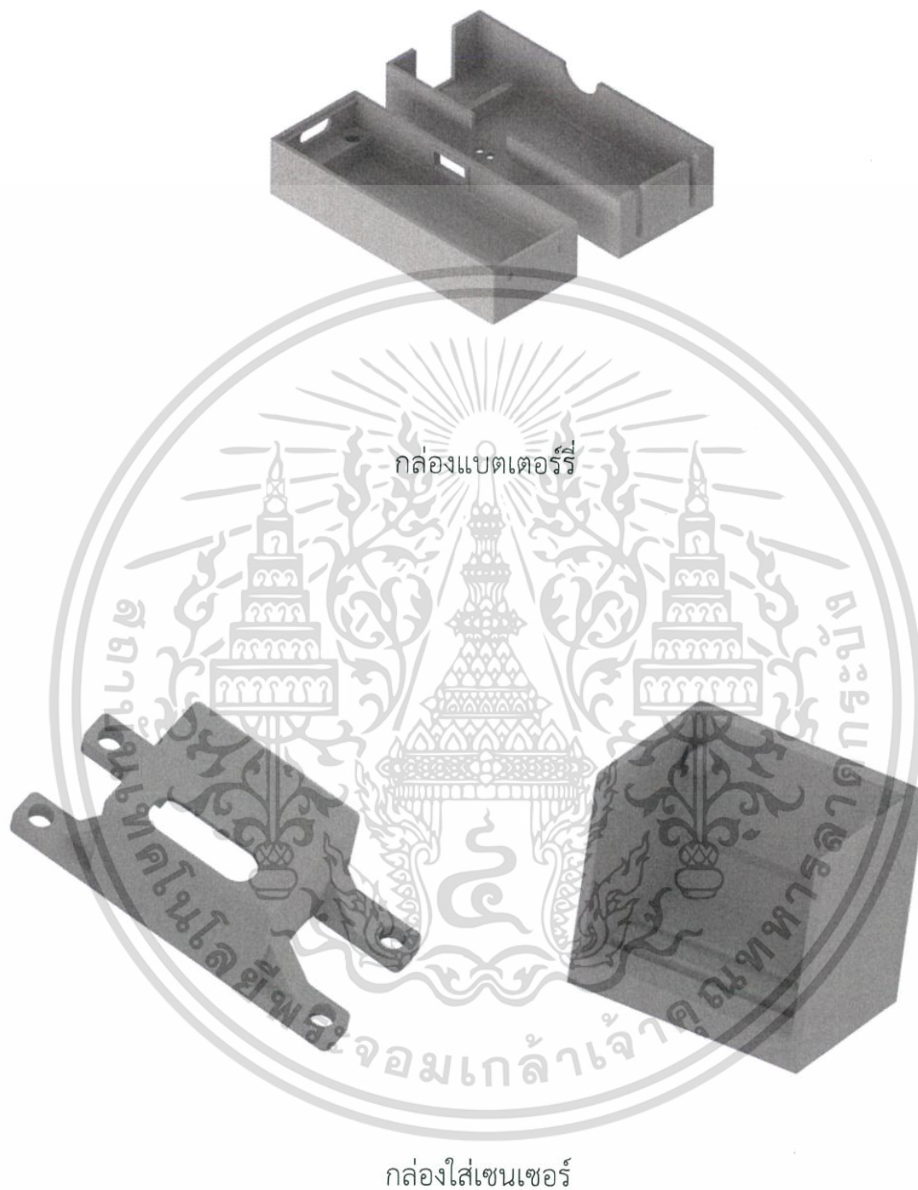
```
//-----VVV-----//
```

```
int MPU6050_write_reg2(int reg2, uint8_t data2)
```

```
{  
    int error2;  
  
    error2 = MPU6050_write2(reg2, &data2, 1);  
  
    return (error2);  
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพ CAD ในส่วนกล่องใส่อุปกรณ์อิเล็กทรอนิกส์ต่างๆ โดยขึ้นรูปชิ้นงานจาก
3D Printer

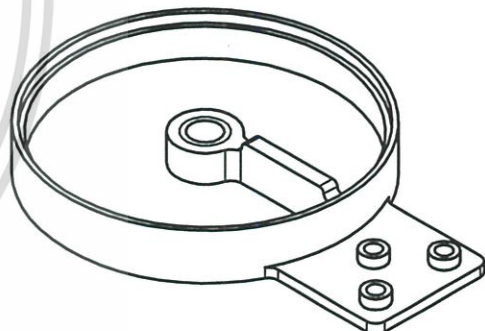
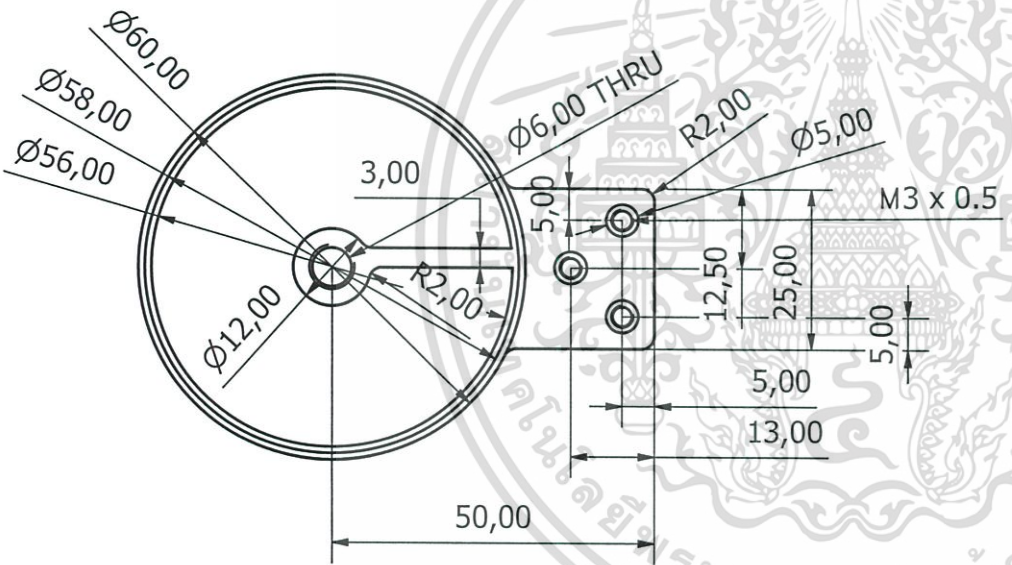
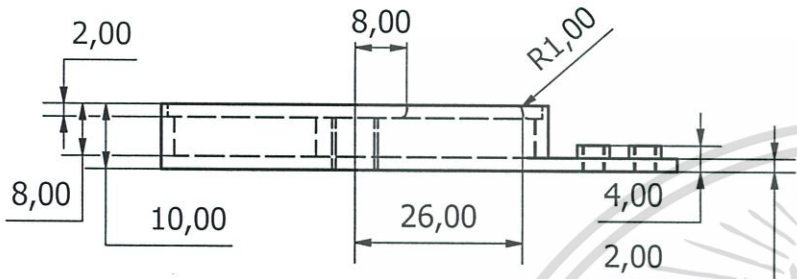


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



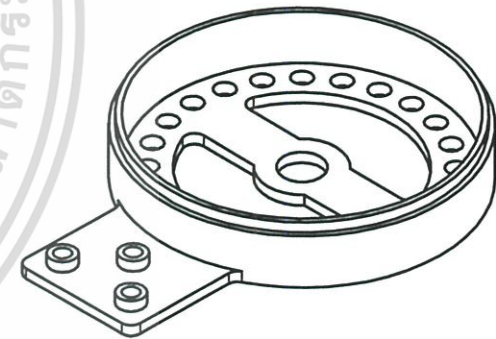
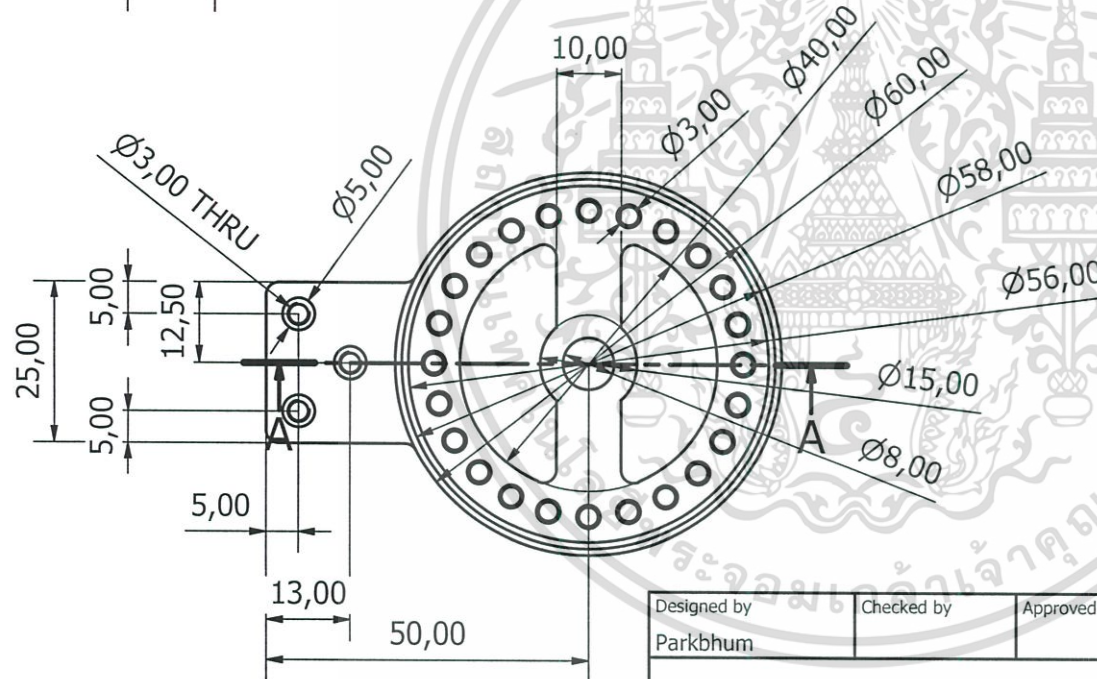
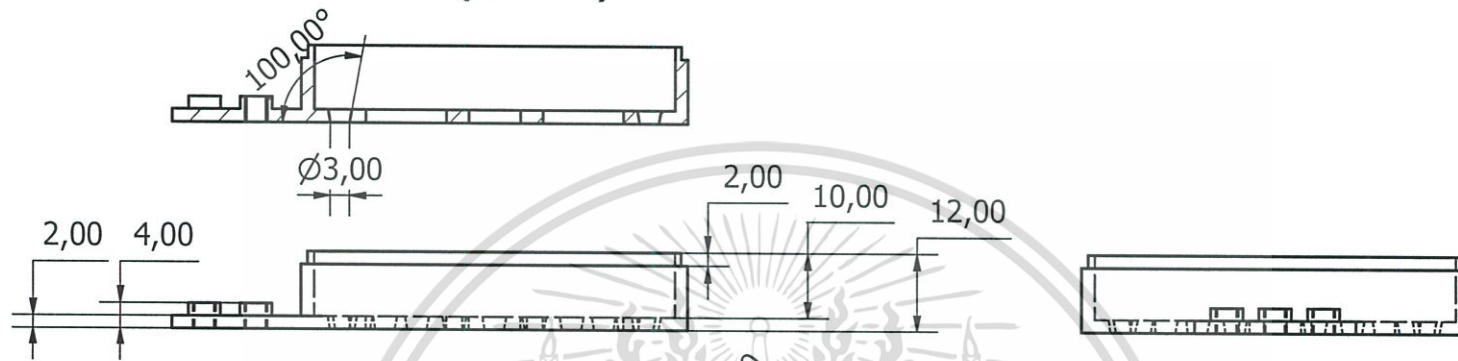
Drawing ของอุปกรณ์จำกัดและวัดมุมมองของเข่า
(กลไกการจำกัดมุมมองของเข่า)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

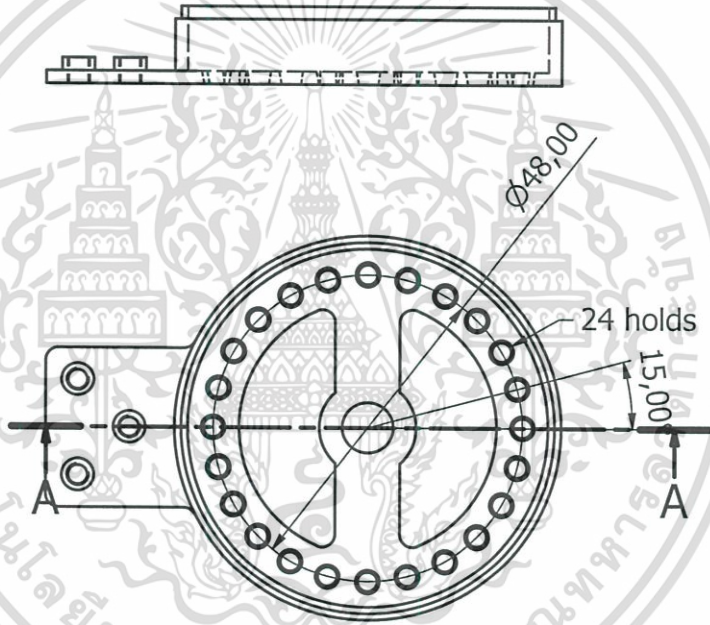
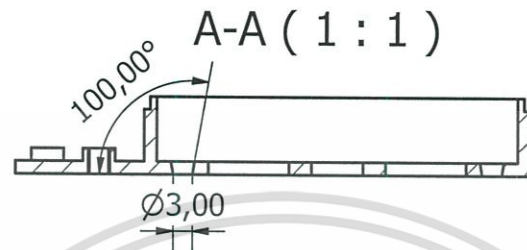


Designed by Parkbhum	Checked by	Approved by	Date	Date 4/6/2017
			Part1	
			Edition	Sheet 2 / 2

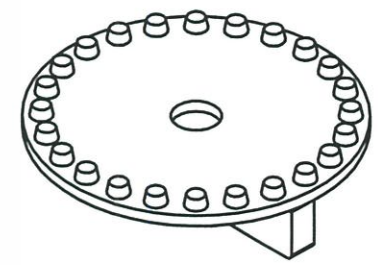
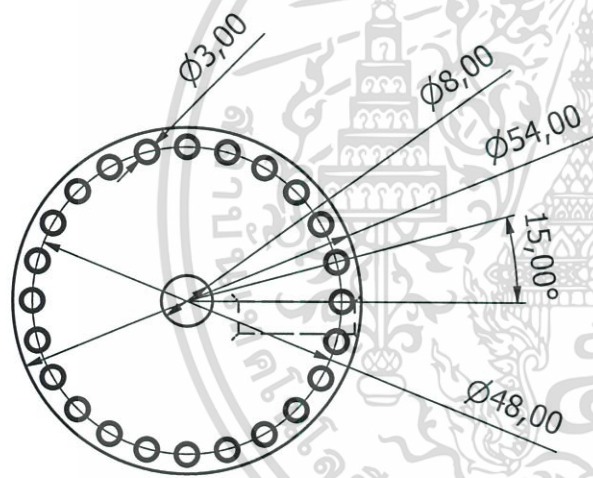
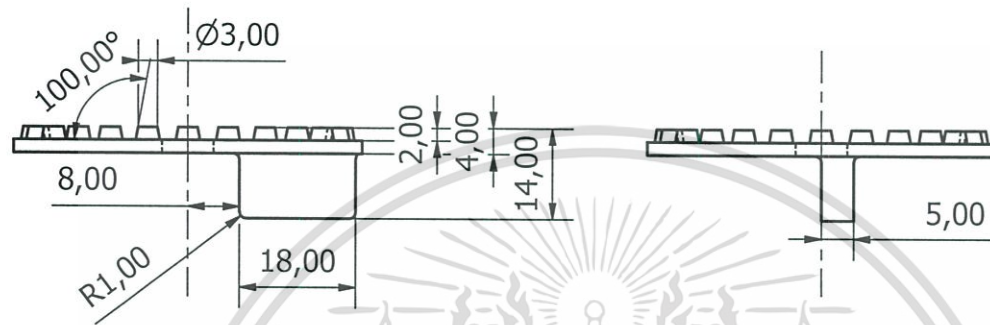
A-A (1:1)



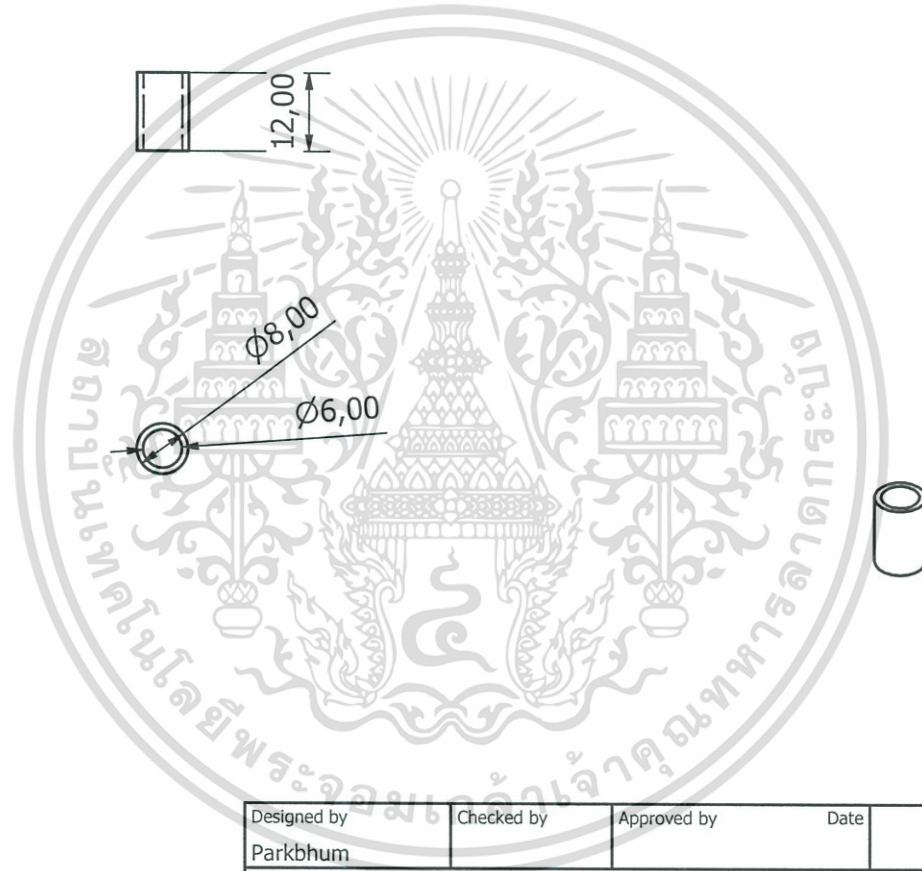
Designed by Parkbhum	Checked by	Approved by	Date	Date 4/6/2017
Part2			Edition	Sheet 2 / 2



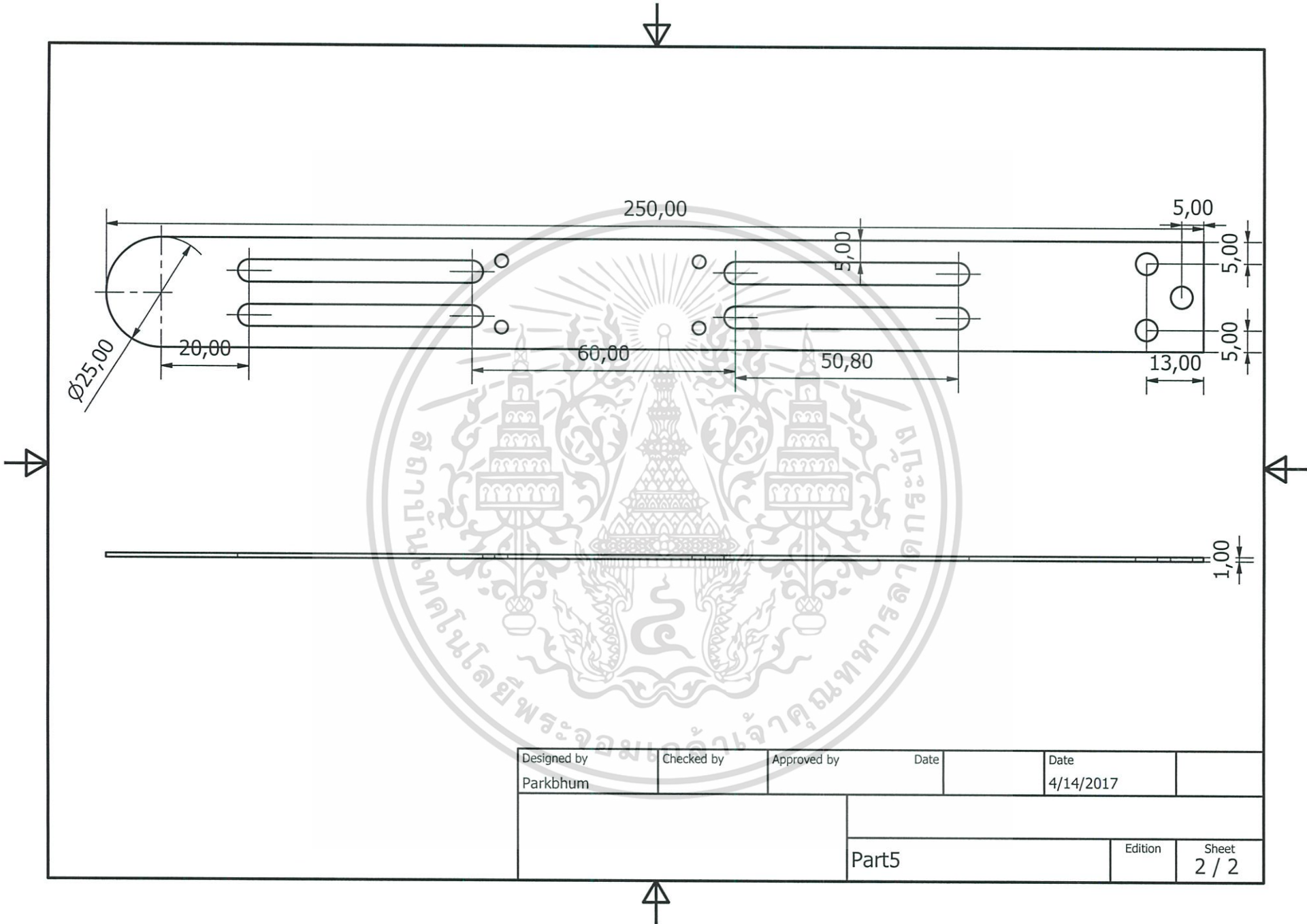
Designed by Parkbhum	Checked by	Approved by	Date	Date 4/6/2017
			Edition	
Part2-1			Sheet 2 / 2	



Designed by Parkbhum	Checked by	Approved by	Date	Date 4/6/2017
			Part3	Edition
			Sheet 2 / 2	

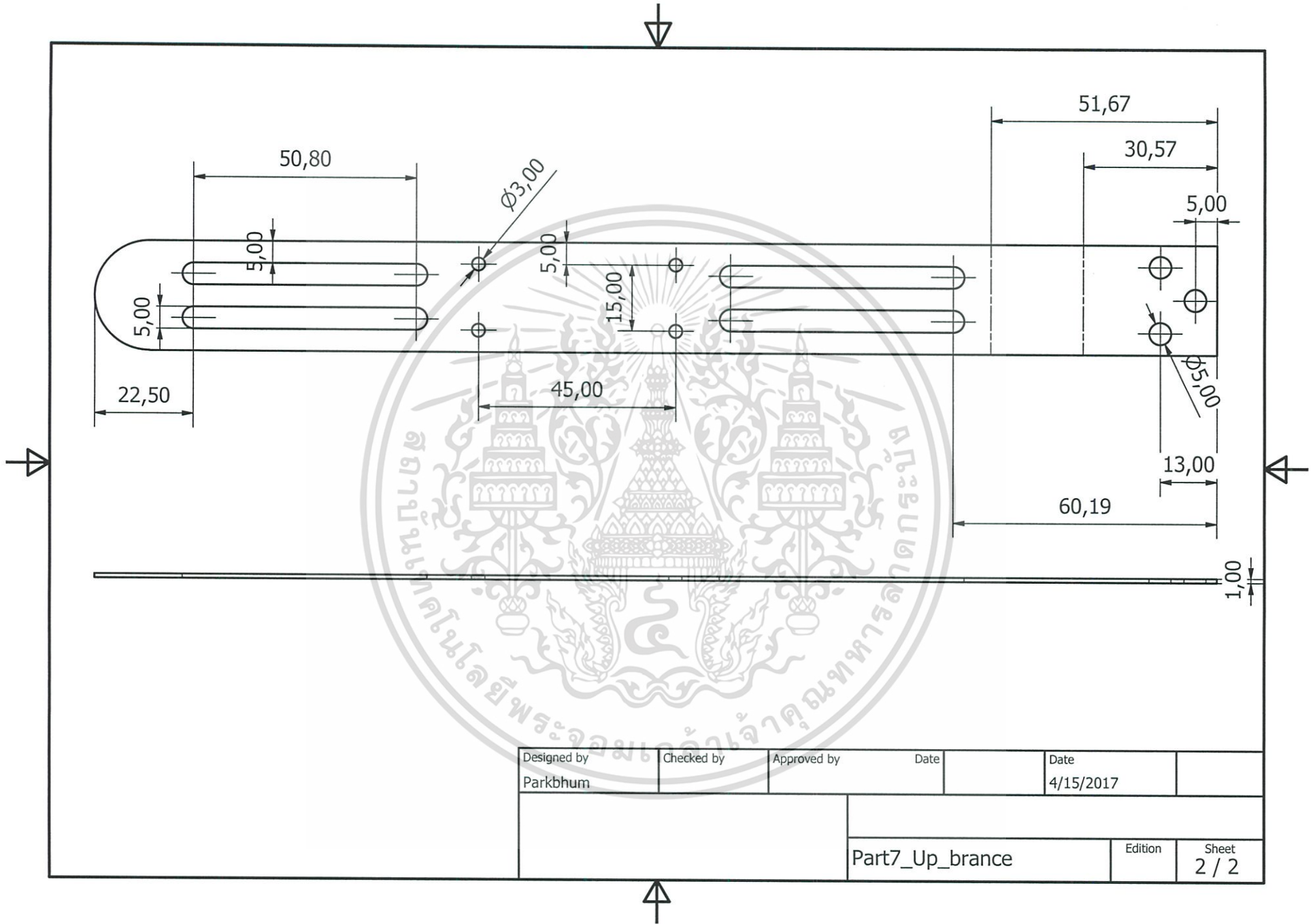


Designed by Parkbhum	Checked by	Approved by	Date	Date 4/6/2017	
			Part4	Edition	Sheet 2 / 2



Designed by Parkbhum	Checked by	Approved by	Date	Date 4/14/2017
			Edition	
			Sheet 2 / 2	

Part5



Designed by Parkbhum	Checked by	Approved by	Date	Date 4/15/2017
			Part7_Up_brance	
			Edition	Sheet 2 / 2