อุปกรณ์ลดการสั่นสะเทือนสำหรับการขนส่ง

Self-Balancing for Transportation

ชนะชนม์ เศรษฐแสงศรี

Chanachon Srattasangsri

ภูมรินทร์ ปุริสพันธ์

Phumarin Purisapun

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2563

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use. Forbidden to modify the content, and cite the document when use.

อุปกรณ์ลดการสั่นสะเทือนสำหรับการขนส่ง

Self-Balancing for Transportation

จัดทำโดย

นาย ชนะชนม์ เศรษฐแสงศรี

นาย ภูมรินทร์ ปุริสพันธ์

อาจารย์ที่ปรึกษา

รศ. จิรวัฒน์ ปานกลาง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2563

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

ปริญญานิพนธ์ปีการศึกษา 2563

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง อุปกรณ์ลดการสั่นสะเทือนสำหรับการขนส่ง

Self-Balancing for Transportation

ผู้จัดทำ นาย ชนะชนม์ เศรษฐแสงศรี รหัสประจำตัว 60010187

นาย ภูมรินทร์ ปุริสพันธ์ รหัสประจำตัว 60011291

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

(รศ. จิรวัฒน์ ปานกลาง)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

หัวข้อปริญญานิพนธ์ อุปกรณ์ลดการสั่นสะเทือนสำหรับการขนส่ง
นักศึกษา นาย ชนะชนม์ เศรษฐแสงศรี รหัสประจำตัว 60010187
นาย ภูมรินทร์ ปุริสพันธ์ รหัสประจำตัว 60011291
ปริญญา วิศวกรรมศาสตรบัณฑิต
ภาควิชา วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา 2563
อาจารย์ที่ปรึกษาโครงงาน รศ. จิรวัฒน์ ปานกลาง

บทคัดย่อ

โครงงานนี้นำเสนออุปกรณ์ลดการสั่นสะเทือนสำหรับการขนส่ง (Self-Balancing for Transportation) โดยมีจุดประสงค์ ที่จะลดการความเสียหายของพัสดุที่ต้องการขนส่งที่เกิดจากการสั่นสะเทือนหรือกระแทก โดยอุปกรณ์ที่สร้างขึ้นอาศัยหลักการการงานของไมโครคอนโทรเลอร์ และเซนเซอร์วัดองศาทั้ง 2 แกน พร้อมทั้ง มอเตอร์จำนวน 3 ตัว การทำงานของมอเตอร์มีการทำงานร่วมกับเซนเซอร์ไจโรสโคป ชิ้นงานนี้เป็นชิ้นงานที่ช่วยให้พัสดุอยู่ขนานกับพื้นโลก หรือก็คือวางอยู่อย่างสมดุล เพื่อเกิดความไม่สมดุลขึ้นจะเกิดค่าความผิดพลาด โดยไมโครคอนโทรเลอร์ได้รับข้อมูลจากเซนเซอร์และทำการประมวณผลค่าความผิดพลาดโดยใช้ความรู้จากทฤษฎี PID Control เพื่อต้องการแก้ความผิดพลาดนั้น โดยจะทำการส่งชุดคำสั่งไปควบคุม การทำงานของมอเตอร์ทั้ง 3 ตัวทำให้มอเตอร์หมุน และพยายามให้พัสดุอยู่นิ่งและขนานกับพื้นโลกที่สุด โดยชิ้นงานนี้สามารถลดความเสียหายของพัสดุที่จะเกิดขึ้นจากการสั่นสะเทือนหรือการกระแทกได้ และยังช่วยลดวัสดุกันกระแทกที่เป็นขยะจากการขนส่ง เพื่อลดปัญหาขยะมูลฝอย จึงมีแนวคิดในการทำอุปกรณ์นี้ขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Project Title	Self-Balancing for Transportation	
Student	Mr. Chanachon Srattasangsri	Student ID 60010187
	Mr. Phumarin Purisapun	Student ID 60011291
Degree	Bachelor of Engineering	
Program	Electronics Engineering	
Year	2020	
Project Advisor	Associate Professor Jirawat Parnklang	

ABSTRACT

This project presents vibration dampers for transportation. (Self-Balancing for Transportation) with the purpose That will reduce the damage of the parcel that needs to be transported caused by vibration or shock. The device was created based on the principle of operation of the microcontroller. And angle sensors, all 2 axes, along with many 3 motors, the motor is working with a gyroscope sensor. This work is the work that keeps supplies parallel to the earth. Or is to be placed in balance to cause an imbalance, there will be an error value. The microcontroller receives data from the sensor and calculates the error value using the knowledge of PID Control theory to correct the error. Which will send a set of commands to control. The operation of three motors makes the motor turn. And try to keep the parcel as still and parallel to the earth as possible By this work piece can reduce the damage of parcels that can be caused by vibration or shock. And also reduce the cushioning material that is transport waste To reduce the problem of solid waste So there was an idea to make this device.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

กิตติกรรมประกาศ

ในการทำโครงงานอุปกรณ์ลดการสั่นสะเทือนสำหรับการขนส่งนี้ผู้จัดทำสามารถทำสำเร็จลุล่วงด้วยดี ผู้จัดทำขอขอบคุณ รศ.จิรวัฒน์ ปานกลาง อาจารย์ที่ปรึกษา ที่ให้ความช่วยเหลือ และให้คำปรึกษาให้คำแนะนำต่าง ๆ ในการเลือกซื้ออุปกรณ์ การทำงานของอุปกรณ์บางตัว การออกแบบวงจร และการจัดวางต่าง ๆ ขอขอบคุณรุ่นที่จบการศึกษาไปแล้ว ที่ได้ให้ความช่วยเหลือตลอดการทำงาน รวมทั้งอธิบายการทำงานของ วรจรบางส่วนเพิ่มเติมให้ ตลอดจนช่วยตรวจสอบหาข้อผิดพลาดต่าง ๆ ขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณคุณพ่อและคุณแม่ที่ได้คอยกำลังใจในการทำงาน และให้เงินทุนสนับสนุนในการทำโครงงานครั้งนี้ และขอขอบคุณเพื่อนๆทุกคนที่ได้ให้การช่วยเหลือ เอื้อเฟื้ออุปกรณ์ต่าง ๆในการทำงานเพื่อความสะดวกสบายมากขึ้น หากประโยชน์ใดเกิดจากรายงานฉบับนี้ผู้จัดทำขอมอบเป็นเครื่องบูชาแด่บุพการี ครู-อาจารย์ และผู้มีพระคุณที่ได้ให้การช่วยเหลือทุกท่าน และสุดท้ายผู้จัดทำหวังเป็นอย่างยิ่งว่าโครงงานนี้จะสามารถเป็นประโยชน์แก่ผู้สนใจ

ชนะชนม์ เศรษฐแสงศรี
ภูมรินทร์ ปุริสพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญ

เรื่อง **หน้า**

บทคัดย่อ I

ABSTRACT II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 ความสำคัญของรายงาน 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 1

1.3 สมมุติฐานการศึกษา 1

1.4 ขอบเขตในการจัดทำโคงงาน 1

1.5 ประโยชน์ที่คาดว่าจะได้รับ 1

บทที่ 2 หลักการและทฤษฎี 2

2.1 หลักการทำงานของเบื้องต้น 2

2.2 ทฤษฎีอุปกรณ์ 2

2.2.1 MPU6050 GY-87 2

2.2.2 Arduino DUE 5

2.2.3 Servo Motor 10

2.3 ทฤษฎีที่เกี่ยวข้อง 13

2.3.1 PID 13

2.3.2 Arduino IDE 16

2.3.2 Matlab Simulink 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 3 วิธีดำเนินการวิจัย 19
3.1 ศึกษาหาข้อมูล 19
3.2 ขั้นตอนการดำเนินงาน 20
3.2.1 วางแผนการทำตัวงาน 20
3.2.2 ออกแบบต่อวงจรของตัวอุปกรณ์ 20
3.2.3 ออกแบบการทำงานของตัวอุปกรณ์ 21
3.3.3 ออกแบบและประกอบตัวอุปกรณ์ตัวอุปกรณ์ 25
บทที่ 4 การทดลอง และผลการทดลอง 26
4.1 คุณสมบัติของอุปกรณ์ลดการสั่นสะเทือนสำหรับการขนส่งตามที่ตั้งไว้ 26
4.2 การทดลองการอ่านค่าเซ็นเซอร์ 26
4.2.1 วิธีทำการทดลอง 26
4.2.2 ผลการทดลอง 27
4.3 การทดลองการวัดสัญญาณควบคุมเซอร์โวมอเตอร์ 30
4.3.1 วิธีการทดลอง 30
4.3.2 ผลการทดลอง 31
4.4 การทดลอง PID Tune 36
4.4.1 วิธีการทดลอง 36
4.4.2 ผลการทดลอง 37
บทที่ 5 สรุปผลการวิจัย และข้อเสนอแนะ 40
5.1 สรุปผลการทดลอง 40
5.2 วิจารณ์ผลการทดลอง 40
5.3 ข้อเสนอแนะ 40
เอกสารอ้างอิง 41
ภาคผนวก 42
คำสั่งที่ใช้ควบคุม 43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญตาราง

ตารางที่ **หน้า**

1 แสดงการต่อกันของ Servo Motor, MPU6050 GY-87 และ Arduino Due 21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญรูป

รูปที่ **หน้า**

2.1 บล็อกไดอะแกรม 2

2.2 MPU6050 GY-87 3

2.3 MPU6050 Pin Configuration 3

2.4 Arduino DUE 5

2.5 Pin out Arduino Board DUE 7

2.6 รูปแบบการเขียนโปรแกรมบน Arduino 8

2.7 ลักษณะทั่วไปของ Servo motor 10

2.8 RC Servo Motor 10

2.9 มุมหรือองศาหมุนขึ้นอยู่กับความกว้างของสัญญาณพัลซ์ 11

2.10 RC Servo Motor หมุนไปที่ 45 องศา 12

2.11 Proportional 13

2.12 Integral 14

2.13 Derivative 15

2.14 Derivative 15

2.15 Derivative 16

2.15 โปรแกรม Arduino IDE 17

2.16 โปรแกรม Matlab Simulink 18

3.1 บล็อกไดอะแกรมแสดงขั้นตอนการดำเดินงาน 20

3.2 การต่อวงจรรวม 20

3.3 แผนผังการทำงานเบื้องต้นของตัวโปรแกรมการทำงาน 21

3.4 code1 22

3.5 code2 22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

รูปที่ **หน้า**

3.6 code3 22

3.7 code4 23

3.8 code5 23

3.9 code6 24

3.10 หลังประกอบอุปกรณ์ 25

4.1 การต่อ GY-87 กับบอร์ด Arduino 26

4.2 Code MPU6050 27

4.3 ค่าที่แสดงของแกน X ย่าน1g 27

4.4 ค่าที่แสดงของแกน Y ย่าน1g 28

4.5 ค่าที่แสดงของแกน Z ย่าน1g 28

4.6 ค่าที่แสดงของแกนย่าน2g 29

4.7 การต่อ MG995 กับบอร์ด Arduino 30

4.8 Code MG995 30

4.9 ความกว้างสัญญาณ PWM คือ 540µs 31

4.10 มอเตอร์อยู่ที่ 0องศา 31

4.11 ความกว้างสัญญาณ PWM คือ 1020µs 32

4.12 มอเตอร์อยู่ที่ 45องศา 32

4.13 ความกว้างสัญญาณ PWM คือ 1460µs 33

4.14 มอเตอร์อยู่ที่ 90องศา 33

4.16 มอเตอร์อยู่ที่ 135องศา 34

4.17 ความกว้างสัญญาณ PWM คือ 2420µs 35

4.18 มอเตอร์อยู่ที่ 0องศา 35

4.19 ไดอะแกรมใน Simulink 36

4.20 กราฟในแกน X ก่อนปรับ (เอียง30องศา) 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

รูปที่ **หน้า**

4.21 กราฟในแกน X หลังปรับ (เอียง30องศา)..... 37

4.22 กราฟในแกน y ก่อนปรับ (เอียง30องศา)..... 38

4.23 กราฟในแกน y หลังปรับ (เอียง30องศา)..... 38

4.24 PID Tuning 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 1
บทนำ

1.1 ความสำคัญของรายงาน

เนื่องด้วยทางภาควิชาวิศวกรรมอิเล็กทรอนิกส์ได้มีนโยบายให้นักศึกษาคิดค้นสิ่งประดิษฐ์ซึ่งทางคณะผู้จัดทำได้คิดค้นทำ อุปกรณ์ลดการสั่นสะเทือนสำหรับการขนส่ง โดยการนำเอาความรู้จากภาคการเรียนมาประยุกต์ใช้ในทางปฏิบัติ เพื่อลดปัญหาสิ้นค้าเสียหายจากการขนส่ง และลดวัสดุกันกระแทกซึ่งเป็นขยะมูลฝอยจากการขนส่งที่ทำให้โลกร้อน

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

- นำความรู้ที่ได้จากการสอนภาคทฤษฎีมาใช้ในทางปฏิบัติ
- ได้ศึกษาหาความรู้เพิ่มเติมเกี่ยวกับหลักการทำงานของวงจรไมโครคอนโทรลเลอร์
- ได้ศึกษาหาความรู้เพิ่มเติมเกี่ยวกับหลักการทำงานของ PID control
- ออกแบบและสร้างอุปกรณ์ลดการสั่นสะเทือน
- เป็นการฝึกปฏิบัติของคณะผู้จัดทำเพื่อเป็นพื้นฐานในภาคปฏิบัติต่อไป

1.3 สมมุติฐานการศึกษา

สามารถเข้าใจวิธีการทำงานของวงจรไมโครคอนโทรลเลอร์ ขั้นตอนการจัดทำ และเข้าใจหลักการทำงานของ PID control สามารถนำความรู้ที่ได้รับจากภาคการสอนในห้องเรียนไปใช้ในภาคปฏิบัติจนทำให้สามารถสร้างอุปกรณ์ลดการสั่นสะเทือนสำหรับการขนส่งมาใช้งานได้จริง

1.4 ขอบเขตในการจัดทำโครงงาน

- ปรับสมดุลได้สองแกน ได้แก่ แกนX และแกนY
- สามารถทำงานได้โดย ระบบอัตโนมัติ

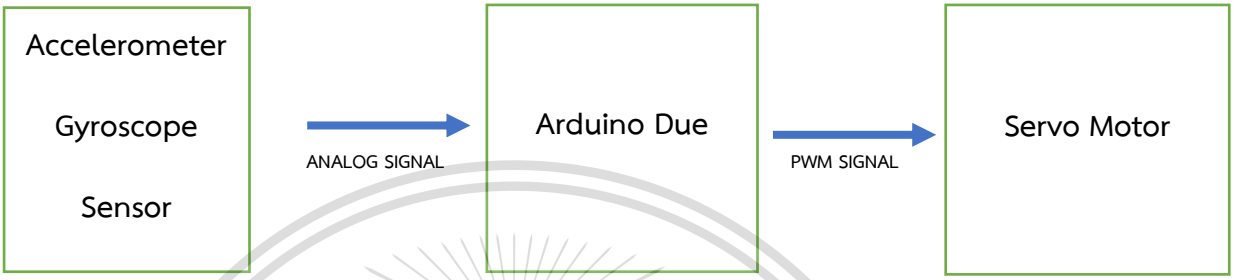
1.5 ประโยชน์ที่คาดว่าจะได้รับ

- มีความรู้ความเข้าใจทั้งด้านทฤษฎีและด้านปฏิบัติในการใช้งานไมโครคอนโทรลเลอร์และการเขียนคำสั่งของไมโครคอนโทรลเลอร์
- อุปกรณ์ชิ้นนี้สามารถช่วยให้การขนส่งมีประสิทธิภาพมากขึ้น โดยลดการสั่นสะเทือนที่ทำให้พัสดุที่ต้องการขนส่งเกิดความเสียหาย
- สามารถนำความรู้เข้าใจที่ได้ศึกษาไปประยุกต์ใช้กับงานแบบอื่น ๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 2
หลักการและทฤษฎี

2.1 หลักการทำงานของเบื้องต้น



รูปที่ 2.1 บล็อกไดอะแกรม

หลักการทำงานเบื้องต้นของตัวอุปกรณ์ จะถูกควบคุมด้วยบอร์ด ไมโครคอนโทรลเลอร์ (Microcontroller) ซึ่งบอร์ดไมโครคอนโทรลเลอร์ นั้นจะใช้ Arduino DUEโดยการเขียนชุดคำสั่งการทำงานควบคุมให้ตัว Servo Motor สามารถหมุนซ้าย และหมุนขวาได้ โดยอัตโนมัติ ถ้าหากตัวแผ่นวางเอียงไปตามพื้นถนนจะสามารถปรับสมดุลได้ และสามารถทราบว่าแผ่นวางนั้นเอียงไปตามพื้นถนนไปเป็นดีกรีมากน้อยเพียงใดจากตัวเซ็นเซอร์ MPU6050 GY-87 ซึ่งเป็นเซ็นเซอร์ที่วัดความเร่งและไจโรสโคป อุปกรณ์ลดการสั่นสะเทือนสำหรับการขนส่งอาศัยพลังงานจากแบตเตอร์รี่ในการทำงานโดยไม่ต้องเสียบปลั๊กขณะใช้งาน และ สามารถชาร์ตแบตเตอร์รี่เก็บไว้เมื่อหยุดการใช้งาน

2.2 ทฤษฎีอุปกรณ์

ส่วนประกอบพื้นฐานของระบบ

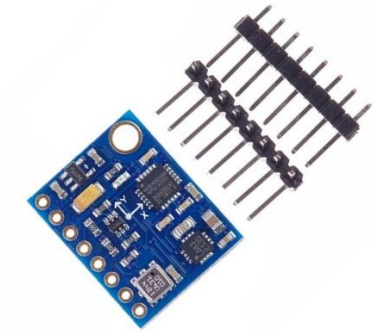
- Accelerometer Gyroscope Sensor
- Arduino DUE
- Servo Motor

2.2.1 MPU6050 GY-87

GY-87 หรือที่หลายๆคนเรียกว่า ไจโร MPU6050 (MPU6050 คือชื่อชิพประมวลผลบนตัว GY-87) คือโมดูลเซ็นเซอร์ที่ตรวจจับการเคลื่อนไหวและความเอียงของวัตถุ โดยตรวจวัดจากความเร่งเชิงเส้น (Linear Acceleration) และ ความเร็วเชิงมุม (Angular Velocity หรือใช้ Gyroscope) ถือเป็นอุปกรณ์แบบ 10 DOF (10 Degrees of Freedom) คืออุปกรณ์ที่สามารถตรวจวัดค่าได้ทั้ง 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

แกนคือ Ax ,Ay ,Az ,Gx ,Gy และ Gz) และ Compass พร้อมทั้งการหาค่าของความกดอากาศได้อีกด้วย

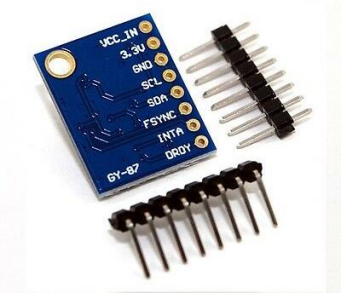


รูปที่ 2.2 MPU6050 GY-87

(รูปภาพจาก https://www.arduino4.com/product/610/gy-87-hw-290-10-dof-gyro-3axis-acceleration-3axis-magnetic-field-and-air-pressure-sensor-imu-mpu6050)

องค์ประกอบของ PIN

VCC_IN	ให้พลังงานสำหรับโมดูล โดยทั่วไปจะใช้ + 5V
3.3V	ให้พลังงานสำหรับโมดูลสามารถ + 3.3V
Ground	เชื่อมต่อกับกราวด์ของระบบ
Serial Clock (SCL)	ใช้สำหรับจ่ายสัญญาณ clock พัลส์สำหรับการสื่อสาร I2C
Serial Data (SDA)	ใช้สำหรับส่งข้อมูลสำหรับการสื่อสาร I2C
FSYNC	หากใช้ MPU6050 มากกว่าหนึ่งตัวใน MCU เดียวก็สามารถใช้พินนี้เพื่อเปลี่ยนที่อยู่ได้,อินพุตดิจิตอลซิงโครไนซ์เฟรม
Interrupt (INT)	เพื่อระบุว่ามีข้อมูลให้ MCU อ่าน
DRDY	การเชื่อมต่อเสริม ต่ำสำหรับ 250 µsec เมื่อข้อมูลถูกวางลงในรีจิสเตอร์เอาต์พุตข้อมูลของ HMC5883L



รูปที่ 2.3 MPU6050 Pin Configuration

(รูปภาพจาก https://picclick.co.uk/GY-87-10DOF-GYRO-Accelerometer-Compass-Barometric-Sensor-HMC5883L-161798809800.html)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

คุณสมบัติ Gyroscope

- ดิจิตอลเอาท์พุท X-, Y- และเซ็นเซอร์อัตราเชิงมุมแกน Z (ไจโรสโคป) ที่มีช่วงเต็มสเกลที่ผู้ใช้ตั้งโปรแกรมได้ที่± 250, ± 500, ± 1000 และ± 2000 ° / วินาที
- สัญญาณซิงค์ภายนอกที่เชื่อมต่อกับพิน FSYNC รองรับการซิงโครไนซ์รูปภาพวิดีโอและ GPS
- ADC แบบ 16 บิตในตัวช่วยให้สามารถ sampling ไจโรได้พร้อมกัน
- ความเสถียรของอุณหภูมิที่เพิ่มขึ้นและความไวช่วยลดความจำเป็นในการสอบเทียบของผู้ใช้
- ปรับปรุงประสิทธิภาพสัญญาณรบกวนความถี่ต่ำ
- ตัวกรองความถี่ต่ำที่ตั้งโปรแกรมได้แบบดิจิทัล
- Gyroscope ปฏิบัติการที่กระแส: 3.6mA
- กระแสไฟสแตนด์บาย: 5μA

คุณสมบัติ Accelerometer

- เครื่องวัดความเร่งแบบสามแกนเอาต์พุตดิจิตอลพร้อมช่วงเต็มสเกลที่ตั้งโปรแกรมได้ที่± 2g, ± 4g, ± 8g และ ± 16 g
- ADC แบบ 16 บิตในตัวช่วยให้สามารถ sampling accelerometers พร้อมกันโดยไม่ต้องต่อภายนอก มัลติเพล็กเซอร์
- Accelerometer กระแสไฟฟ้าทำงานปกติ: 500μA
- โหมดพลังงานต่ำปัจจุบัน: 10μA ที่ 1.25Hz, 20μA ที่ 5Hz, 60μA ที่ 20Hz, 110μA ที่ 40 Hz
- การตรวจจับทิศทางและการส่งสัญญาณ
- Tap detection
- interrupts ที่ผู้ใช้ตั้งโปรแกรมได้
- High-G interrupt
-

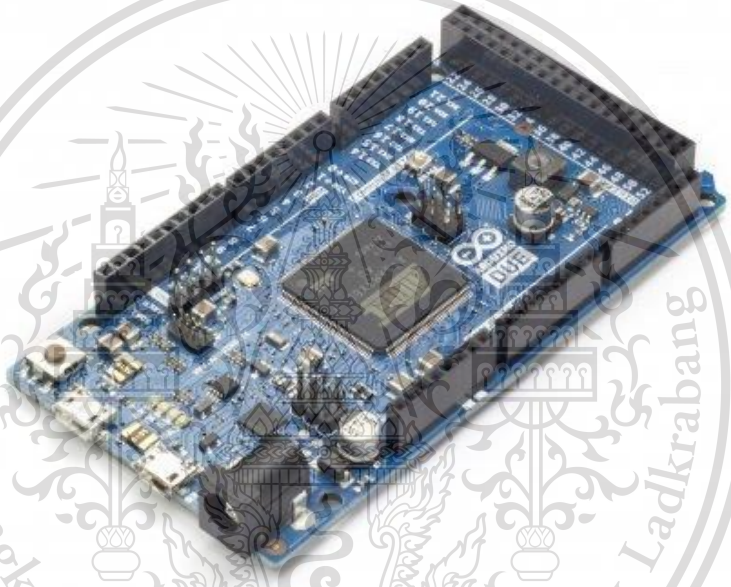
คุณลักษณะเพิ่มเติม

- MEMS 3-aixs accelerometerและ3-axis gyroscope รวมกันและเซ็นเซอร์อุณหภูมิในตัว
- แหล่งจ่ายไฟ: 3-5V
- การสื่อสาร: โปรโตคอล I2C
- ADC 16 บิตในตัวให้ความแม่นยำสูง
- สามารถใช้เพื่อเชื่อมต่อกับอุปกรณ์ I2C อื่น ๆ เช่น magnetometer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.2.2 Arduino DUE

Arduino คือ บอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR มีกำรพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software โดยด้าน Hardware คือ บอร์ดอิเล็กทรอนิกส์ขนาดเล็กสามารถเคลื่อนย้ายพกพาได้สะดวกโดยมีไมโครคอนโทรลเลอร์เป็นอุปกรณ์หลักที่สำคัญและมีอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆมาประกอบรวมกัน ต่อมาในด้าน Software ตัวบอร์ด Arduino เป็นบอร์ดที่ใช้ไมโครคอนโทรลเลอร์ดังนั้นจึงมีลักษณะภาษาแบบเดียวกับ ภาษา C/C++ซึ่งเป็นภาษาสำหรับเขียนโปรแกรมที่แพร่หลาย ดังนั้นตัวบอร์ด Arduino จึงสามารถใช้งานได้ง่าย จึงเหมาะสำหรับผู้เริ่มต้นศึกษาทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ดหรือโปรแกรมต่อได้อีกด้วย



รูปที่ 2.4 Arduino DUE

(รูปภาพจาก http://www.cei-store.com/product/24/arduino-due-2)

Arduino Due เป็นบอร์ดที่ใช้ชิป Atmel AT91SAM3X8E ที่อยู่ในตระกูล ARM Cortex-M3 ซึ่งแตกต่างจากบอร์ด Arduino ตัวอื่น ๆ ที่ใช้ Micro-controller (chip) ตระกูล AVR ทำให้การประมวลผลของ Arduino Due เร็ว แต่ยังคงรูปแบบโค้ดโปรแกรมของ Arduino Arduino Due เป็นบอร์ดไมโครคอนโทรลเลอร์ รุ่นแรกของ Arduino ที่ใช้ไมโครคอนโทรลเลอร์ 32 บิต โมดูล CAN และสัญญาณนาฬิกาที่มีความถี่ 84 MHz มี digital I/O 54 ขา, analog input 12 ขา, UART จำนวน 4 ช่อง, DAC จำนวน 2 ช่อง, TWI จำนวน 2 ช่อง, มีการรองรับการทำงานของอุปกรณ์ USB สำหรับเชื่อมต่ออุปกรณ์ภายนอก โดย Arduino Due ต้องใช้กับ ซอฟแวร์ Arduino IDE เวอร์ชั่น 1.5 ขึ้นไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

<u>องค์ประกอบของ PIN</u>

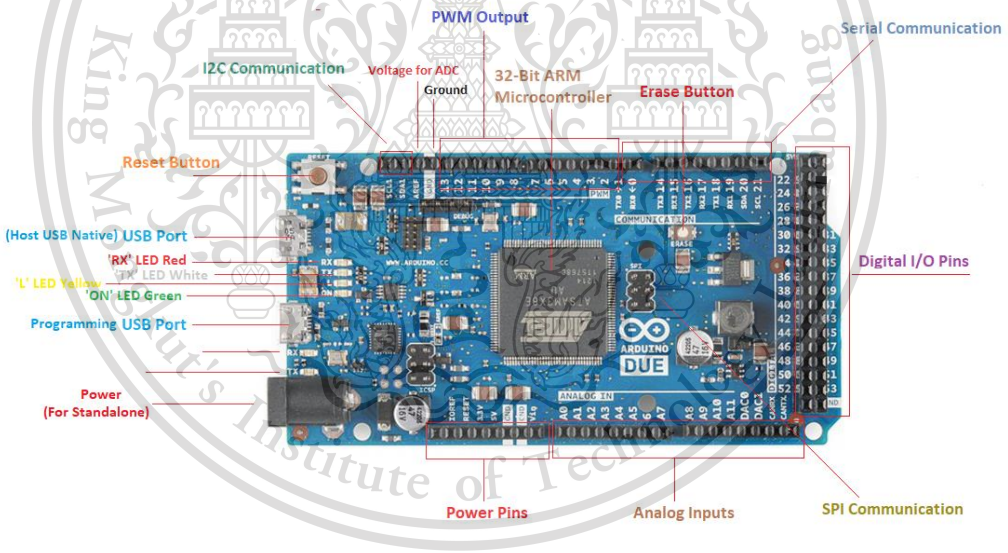
VIN:	แรงดันไฟฟ้าภายนอก (ไม่เกี่ยวข้องกับ 5V จาก USB หรือแรงดันไฟฟ้าเสถียรอื่น ๆ) ด้วยข้อสรุปนี้คุณสามารถส่งได้อย่างไร แหล่งจ่ายไฟภายนอกและใช้กระแสไฟเมื่ออุปกรณ์ใช้พลังงานจากอะแดปเตอร์ภายนอก
5V:	เอาต์พุตจะรับแรงดันไฟฟ้า 5 V จากตัวควบคุมแรงดันไฟฟ้าบนบอร์ดไม่ว่าอุปกรณ์จะใช้พลังงานอย่างไร: จากอะแดปเตอร์ (7-12 V) จาก USB (5 V) หรือผ่านขา VIN (7-12 V). ไม่แนะนำให้จ่ายไฟอุปกรณ์ผ่านขา 5V - ในกรณีนี้จะไม่ใช้ตัวควบคุมแรงดันไฟฟ้าซึ่งอาจทำให้บอร์ดล้มเหลวได้
3.3V:	3.3 V จากตัวควบคุมแรงดันไฟฟ้าของบอร์ด กระแสสูงสุดคือ 800 mA โคลงยังให้พลังงานแก่ไมโครคอนโทรลเลอร์ SAM3X
GND:	การค้นพบภาคพื้นดิน
IOREF:	พินนี้ให้ข้อมูลเกี่ยวกับแรงดันไฟฟ้าที่ใช้งานของไมโครคอนโทรลเลอร์แก่บอร์ดส่วนขยาย บอร์ดขยายสามารถเปลี่ยนไปใช้แหล่งจ่ายไฟที่เหมาะสมหรือใช้ตัวแปลงระดับซึ่งจะช่วยให้สามารถทำงานกับอุปกรณ์ 5V และ 3.3V ได้ขึ้นอยู่กับแรงดันไฟฟ้า
I/OPort:Digital I/O:	หมุด 0 - 53 ระดับลอจิกของหนึ่งคือ 3.3 V ศูนย์คือ 0 V กระแสเอาต์พุตสูงสุดคือ 3 หรือ 15 mA ขึ้นอยู่กับเอาต์พุตของไมโครคอนโทรลเลอร์หรือกระแสอินพุตคือ 6 หรือ 9 mA ตัวต้านทานแบบดึงขึ้น 100 kΩ เชื่อมต่อกับหน้าสัมผัสซึ่งจะปิดโดยค่าเริ่มต้น แต่ซอฟต์แวร์สามารถเปิดได้
PWM:	หมุด 2 - 13 อนุญาตให้เอาต์พุตค่าอะนาล็อกเป็นสัญญาณ PWM ความละเอียด PWM ช่วยให้คุณสามารถเปลี่ยนฟังก์ชัน analogWriteResolution ()
ADC:	หมุด A0 - A11 อินพุตอะนาล็อก 12 ตัวซึ่งแต่ละตัวสามารถแทนแรงดันไฟฟ้าแบบอนาล็อกเป็นตัวเลข 12 บิต (ค่า 4096) ตามค่าเริ่มต้นความกว้างของบิตจะถูกตั้งค่าเป็น 10 บิตเพื่อให้เข้ากันได้กับบอร์ดต่าง ๆ ซึ่งเรื่องของความละเอียด ADC สามารถเปลี่ยนแปลงได้โดยใช้ฟังก์ชัน analogReadResolution () อินพุตอะนาล็อกของบอร์ดวัดจาก 0 ถึงค่าสูงสุด 3.3 V. หากใช้แรงดันไฟฟ้าที่สูงกว่า 3.3 V กับอินพุตจะทำให้คริสตัล SAM3X เสียหาย
TWI / I²C:	หมุด 20 (SDA), 21 (SCL) และ TWI 2 / I²C 2: SDA1 และ SCL1 สำหรับการสื่อสารกับอุปกรณ์ต่อพ่วงโดยใช้โปรโตคอลแบบซิงโครนัสผ่านสายไฟ 2 สาย ใช้ไลบรารี Wire เพื่อทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

SPI: หมุด SPI ถูกส่งไปยังขั้วต่อ 6 ขากลางซึ่งเข้ากันได้กับ Uno, Leonardo และ Mega2560 ใช้ไลบรารี SPI สำหรับการสลับ SPI

UART: อนุกรม: หมุด 0 (RX) และ 1 (TX);
Serial1: หมุด 19 (RX1) และ 18 (TX1);
Serial2: หมุด 17 (RX2) และ 16 (TX2); Serial3: หมุด 15 (RX3) และ 14 (TX3) พินเหล่านี้ใช้เพื่อรับ (RX) และส่งข้อมูล (TX) ผ่านอินเทอร์เฟซแบบอนุกรม พิน 0 (RX) และ 1 (TX) เชื่อมต่อกับพินที่เกี่ยวข้องของ ATmega16U2 ซึ่งทำหน้าที่เป็นตัวแปลง USB-UART ในการสื่อสารกับ Arduino เนื่องจากคอมพิวเตอร์ผ่านพอร์ตการเขียนโปรแกรมให้ใช้คลาส Serial

DAC1 / DAC2: หมุด DAC1 และ DAC2 DAC ให้เอาต์พุตแบบอะนาล็อกที่มีความละเอียด 12 บิต (4096 ระดับ) โดยใช้ฟังก์ชัน analogWrite () พินเหล่านี้สามารถใช้เป็นเอาต์พุตเสียงโดยใช้ไลบรารีเสียง



รูปที่ 2.5 Pin out Arduino Board DUE

(รูปภาพจาก https://www.myarduino.net/product/672/arduino-due)

<u>ข้อมูลจำเพาะ</u>

ไมโครคอนโทรลเลอร์: AT91SAM3X8E

ความถี่นาฬิกา: 84 เมกะเฮิรตซ์

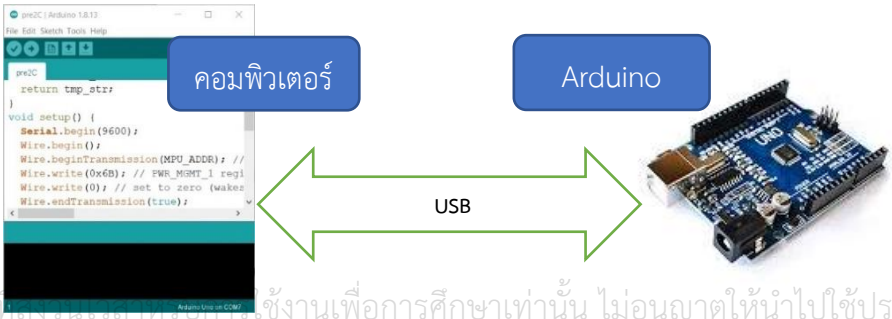
ความลึกของบิต: 32 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

แรงดันไฟฟ้า:	3.3V
แรงดันไฟฟ้าขาเข้า (แนะนำ):	7-12V
แรงดันไฟฟ้าขาเข้า (ขีด จำกัด):	6-16V
พินดิจิตอล I / 0:	54 เส้น (12 อันคือ PWM)
อินพุตแบบอนาล็อก:	12 (ADC)
เอาต์พุตอนาล็อก:	2 (DAC)
หน่วยความจำแฟลช:	512 KB (มีหน่วยความจำทั้งหมดสำหรับโปรแกรม)
หน่วยความจำ SRAM:	96 KB
กระแสสูงสุดที่ขา I / 0:	130 mA
กระแสสูงสุดที่ขา 3.3V:	800 mA
กระแสสูงสุดที่ขา 5V:	800 mA
ความยาว:	101.52 มม.
ความกว้าง:	53.3 มม.
น้ำหนัก:	36 กรัม

การเขียนโปรแกรมพื้นฐาน Arduino Arduino

ใช้พื้นฐานลักษณะภาษาแบบเดียวกับภาษา C/C++ ดังนั้นจึงมีโครงสร้างเหมือนกับ ภาษา C/C++ ทั่วไปนั่นเองทำให้ง่ายต่อการศึกษาใช้งานสำหรับผู้ที่มีความรู้ภาษา C/C++ สำหรับผู้เริ่มใช้Arduino มีหลักสำคัญที่ต้องรู้เกี่ยวกับคำสั่งพื้นฐาน Arduino ดังนี้พรีโปรเซสเซอร์ไดเร็กทีฟ (Preprocessor directives) คือส่วนที่คอมไพลเลอร์จะทำการประมวลผลและทำตามคำสั่งก่อนที่จะมีการคอมไพล์โปรแกรม ส่วนนี้จะเริ่มด้วยเครื่องหมายไดเร็กทีฟ (directive) หรือสัญลักษณ์เครื่องหมายสี่เหลี่ยม # แล้วตามด้วยชื่อคำสั่งที่ต้องการใช้งาน หรือกำหนด ซึ่งในส่วนนี้จะอยู่ใน ส่วนบนสุด หรือส่วนหัวของโปรแกรมและต้องอยู่นอกฟังก์ชั่นหลักใดๆก็ตามถ้าให้เข้าใจง่ายๆนั่นคือ #include เป็นคำสั่งที่ใช้อ้างอิงไฟล์ภายนอกเพื่อเรียกใช้ฟังก์ชั่นหรือตัวแปรที่มีการสร้าง



รูปที่ 2.6 รูปแบบการเขียนโปรแกรมบน Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

กระบวนการโหลดโปรแกรมลงในไมโครคอนโทรลเลอร์ SAM3X แตกต่างจากกระบวนการเฟิร์มแวร์สำหรับไมโครคอนโทรลเลอร์ AVR ที่ใช้ในบอร์ด Arduino อื่น ๆ ความผิดปกติของ SAM3X คือในการรีแฟลชคุณต้องล้างหน่วยความจำแฟลชของคอนโทรลเลอร์ก่อน นี่เป็นเพราะกระบวนการโหลดโปรแกรมถูกควบคุมโดย bootloader ใน SAM3X ROM ซึ่งจะเริ่มทำงานก็ต่อเมื่อไม่มีโปรแกรมในหน่วยความจำแฟลชของไมโครคอนโทรลเลอร์ ดังนั้นจึงสามารถใช้พอร์ต USB ใดก็ได้ในการแฟลชบอร์ด อย่างไรก็ตามขอแนะนำให้ใช้พอร์ต USB สำหรับการเขียนโปรแกรม ("Programming Port" ในรูป) เนื่องจากลักษณะเฉพาะบางประการของกระบวนการล้างหน่วยความจำไมโครคอนโทรลเลอร์:

Programming Port: หากต้องการใช้พอร์ตนี้ใน Arduino IDE ให้เลือก "Arduino Due (Programming Port)" เป็นบอร์ดทำงาน เชื่อมต่อเนื่องจากคอมพิวเตอร์ของคุณโดยเชื่อมต่อสาย USB เข้ากับขั้วต่อการเขียนโปรแกรม (ใกล้กับขั้วต่อสายไฟ) พอร์ตการเขียนโปรแกรมโต้ตอบกับไมโครวงจร 16U2 ซึ่งทำหน้าที่เป็นตัวแปลง USB-UART ในทางกลับกันไมโครวงจร 16U2 เชื่อมต่อกับ UART แรกของไมโครคอนโทรลเลอร์ SAM3X (พิน RX0 และ TX0) และยังควบคุมพินรีเซ็ตและลบด้วย เมื่อพอร์ตเปิดและปิดที่ 1200 บอดระดับแอคทีฟจะถูกสร้างขึ้นที่พินลบและรีเซ็ตซึ่งจะล้างหน่วยความจำของไมโครคอนโทรลเลอร์ ดังนั้นขั้นตอนที่เรียกว่า "การทำความสะอาดฮาร์ดแวร์" จึงถูกเรียกใช้ก่อนที่จะโต้ตอบกับ SAM3X UART วิธีนี้มีความน่าเชื่อถือมากกว่า "การทำความสะอาดซอฟต์แวร์" เมื่อใช้พอร์ต USB มาตรฐานและใช้งานได้แม้ว่าโปรเซสเซอร์จะหยุดทำงาน นี่คือเหตุผลที่แนะนำให้ใช้พอร์ตการเขียนโปรแกรมสำหรับเฟิร์มแวร์ Arduino Due

Native Port: หากต้องการใช้พอร์ตนี้ใน Arduino IDE เป็นบอร์ดทำงานให้เลือก "Arduino Due (Naive USB Port)" พอร์ต USB มาตรฐานเชื่อมต่อโดยตรงกับไมโครคอนโทรลเลอร์ SAM3X เชื่อมต่อเนื่องจากคอมพิวเตอร์ของคุณโดยเชื่อมต่อสาย USB เข้ากับขั้วต่อ USB ของ OEM (อยู่ใกล้กับปุ่มรีเซ็ต) การเปิดและปิดพอร์ตที่ 1200 baud จะทริกเกอร์ขั้นตอน "software cleanup" ซึ่งจะล้างหน่วยความจำแฟลชรีบูตไมโครคอนโทรลเลอร์และเริ่ม bootloader เนื่องจากขั้นตอนนี้ดำเนินการโดยโปรแกรมของไมโครคอนโทรลเลอร์ SAM3X เท่านั้นหากหลังค้างกระบวนการทำความสะอาดอาจไม่เกิดขึ้น ในขณะเดียวกันการเปิด / ปิดพอร์ตมาตรฐานด้วยความเร็วที่ต่างกันจะไม่ช่วยรีสตาร์ทไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.2.3 Servo Motor

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

รูปที่ 2.7 ลักษณะทั่วไปของ Servo motor
(รูปภาพจาก www.Pyroelectro.com/)

ใน RC Servo Motor จะมีสีของสายแตกต่างกันไปดังนี้
สายสีน้ำตาล คือ กราวด์
สายสีแดง คือ ไฟเลี้ยง (4.8-6V)
สายสีส้ม คือ สายส่งสัญญาณพัลซ์ควบคุม (3-5V)

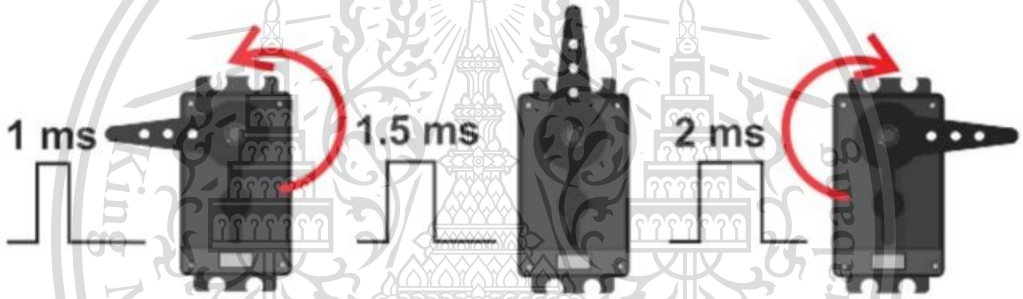
รูปที่ 2.8 RC Servo Motor
(รูปภาพจาก https://shopee.co.th/เกียร์ มอเตอร์ MG995Tower Pro Copper Servo สำหรับ R/C Carสีดำ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

หลักการทำงานของ RC Servo Motor

เมื่อจ่ายสัญญาณพัลซ์เข้ามายัง RC Servo Motor ส่วนวงจรควบคุม (Electronic Control System) ภายใน Servo จะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลซ์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่ง องศาที่ต้องการให้ Motor หมุนเคลื่อนที่ไปยังตำแหน่งนั้น แล้วส่งคำสั่งไปทำการควบคุมให้ Motor หมุนไปยังตำแหน่งที่ต้องการ โดยมี Position Sensor เป็นตัวเซ็นเซอร์คอยวัดค่ามุมที่ Motor กำลังหมุนเป็น Feedback กลับมาให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้องแม่นยำ สัญญาณ RC ในรูปแบบ PWM

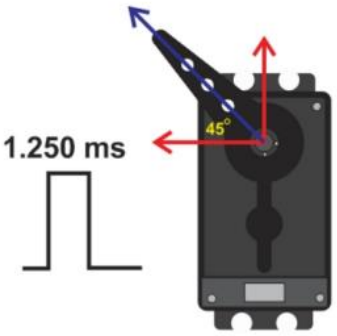
ตัวอย่างเช่นหากกำหนดความกว้างของสัญญาณพัลซ์ไว้ที่ 1 ms ตัว Servo Motor จะหมุนไปทางด้านซ้ายจนสุดในทางกลับกันหากกำหนดความกว้างของสัญญาณพัลซ์ไว้ที่ 2 ms ตัว Servo Motor จะหมุนไปยังตำแหน่งขวาสุด แต่หากกำหนดความกว้างของสัญญาณพัลซ์ไว้ที่ 1.5 ms ตัว Servo Motor ก็จะหมุนมาอยู่ที่ตำแหน่งตรงกลางพอดี



รูปที่ 2.9 มุมหรือองศาหมุนขึ้นอยู่กับความกว้างของสัญญาณพัลซ์
(รูปภาพจาก www.Learn.parallax.com/KickStart/900-00008)

ดังนั้นสามารถกำหนดองศาการหมุนของ RC Servo Motor ได้โดยการเทียบค่า เช่นRC Servo Motor สามารถหมุนได้ 180 องศา โดยที่ 0 องศาใช้ความกว้างพัลซ์เท่ากับ 1000 us ที่ 180 องศาความกว้างพัลซ์เท่ากับ 2000 us เพราะฉะนั้นค่าที่เปลี่ยนไป 1 องศาจะใช้ความกว้างพัลซ์ต่างกัน (2000-1000)/180 เท่ากับ 5.55 us จากการหาค่าความกว้างพัลซ์ที่มุม 1 องศาข้างต้น หากต้องกำหนดให้ RC Servo Motor หมุนไปที่มุม 45 องศาจะหาค่าพัลซ์ที่ต้องการได้จาก 5.55 x 45 เท่ากับ 249.75 us แต่ที่มุม 0 องศาเราเริ่มที่ความกว้างพัลซ์ 1ms หรือ 1000 us เพราะฉะนั้นความกว้างพัลซ์ที่ใช้กำหนดให้ RC Servo Motor หมุนไปที่ 45 องศา คือ 1000 + 249.75 เท่ากับประมาณ 1250 us

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 2.10 RC Servo Motor หมุนไปที่ 45 องศา

(รูปภาพจาก www.Learn.parallax.com/KickStart/900-00008)

วิธีควบคุม RC Servo Motor ด้วย Arduino

Arduino มีไลบรารี่สำหรับสั่งงาน RC Servo Motor มาให้ใช้งานอยู่แล้ว เป็นฟังก์ชั่นสำเร็จรูปและใช้งาน ได้ง่าย ในหน้าเว็บไซต์ http://arduino.cc/en/reference/servo ได้ให้ข้อมูลไว้ว่า Servo Library ของ Arduino สามารถสั่งงาน RC Servo Motor ได้ทั้งแบบหมุนไป-กลับได้ 0-180 องศา (ที่กล่าวถึงตามตัวอย่างข้างต้น) และ แบบต่อเนื่องที่หมุนครบรอบได้เรียกว่าเป็น Continuous Rotation Servo โดยสามารถรองรับการเชื่อมต่อ RC Servo Motor ได้ถึง 12 ตัวกับบอร์ด Arduino DUEและรองรับ สูงสุดถึง 48 ตัวหากใช้บอร์ด Arduino Mega ฟังก์ชั่นภายใน Servo Library

attach() คือฟังก์ชั่นที่ใช้ในการกำหนดขาสัญญาณที่ Servo Motor ต่อกับ Arduino และกำหนดความกว้างของพัลซ์ที่ 0 องศาและ 180 องศา

write() คือฟังก์ชั่นที่ใช้ควบคุมตำแหน่งที่ต้องการให้ Servo Motor หมุนไปยังองศาที่กำหนดสามารถ กำหนดเป็นค่าองศาได้คือ 0-180 องศา แต่ใน Servo Motor ที่เป็น Full Rotation คำสั่ง write จะเป็นการกำหนดความเร็วในการหมุน โดยค่าเท่ากับ 90 คือคำสั่งให้ Servo Motor หยุดหมุน ค่าเท่ากับ 0 คือการหมุนด้วยความเร็วสูงสุดในทิศทางหนึ่ง ค่าเท่ากับ 180 คือการหมุนด้วยความเร็วสูงสุดในทิศทางตรงกันข้าม

writeMicroseconds() คือฟังก์ชั่นที่ใช้ควบคุมตำแหน่งที่ให้ Servo Motor หมุนไปยังตำแหน่งองศาที่กำหนดโดยกำหนด เป็นค่าความกว้างของพัลซ์ในหน่วย us ปกติแล้ว RC Servo Motor จะใช้ความกว้างของพัลซ์อยู่ที่ 1000-2000 us ตามที่ได้กล่าวไปข้างต้นแล้ว แต่ RC Servo Motor บางรุ่นหรือบางยี่ห้อไม่ได้ใช้ช่วงความกว้างของพัลซ์ตามที่ได้ กล่าวเอาไว้นี้ อาจจะใช้ช่วง 700-2300 แทนก็สามารถใช้ฟังก์ชั่น writeMicroseconds นี้เพื่อกำหนดความกว้างพัลซ์ได้เอง การใช้ฟังก์ชั่น writeMicroseconds สามารถกำหนดค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ได้อิสระ ต้องระวังในการใช้งานหากสั่งงาน RC Servo Motor (แบบ 0 - 180 องศา) จนหมุนไปเกินจุดสิ้นสุดคือเกินทั้งฝั่ง 0 หรือ 180 องศา จะทำให้เกิดเสียง คร่างดังจากการหมุนไปต่อไม่ได้และมอเตอร์จะกินกระแสสูงขึ้นด้วยในเวลาเดียวกันนั้น ซึ่งอาจทำให้ RC Servo Motor เกิดความเสียหายได้

read() คือฟังก์ชั่นอ่านค่าองศาที่สั่งเข้าไปด้วยฟังก์ชั่น write() เพื่อให้รู้ว่าตำแหน่งองศาสุดท้าย ที่สั่งเข้าไปนั้นมีค่าเท่าไหร่ ซึ่งค่าที่อ่านออกมานั้นจะมีค่าอยู่ในช่วง 0 - 180

attached() คือฟังก์ชั่นตรวจสอบว่า Servo ที่ต้องการใช้กำลังต่ออยู่กับขาสัญญาณของ Arduino หรือไม่

detach() คือฟังก์ชั่นคืนสถานะของขาที่เรากำหนดให้เป็นขาควบคุม Servo Motor ด้วยคำสั่ง attached() ให้กลับคือสู่การใช่งานปกติ

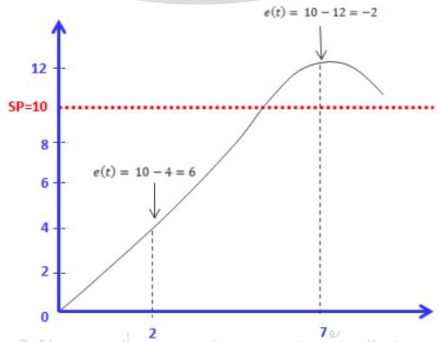
2.3 ทฤษฎีที่เกี่ยวข้อง

2.3.1 PID

P (Proportional) เป็นตัวที่เข้าใจง่ายที่สุด คือนำค่า Error ณ เวลานั้นไปคูณกับ gain K_p นั่นเอง

$$P_{out}=K_p e(t)$$

จากสมการข้างล่าง สังเกตว่าที่เวลา t = 2 เมื่อกราฟยังวิ่งไม่ถึง SP ค่า Error จะเป็นบวก เมื่อนำไปคูณกับ K_p จะทำให้ค่า P_{out} เป็นบวกด้วย จึงไปทำการเพิ่ม Output ให้ถึงค่า SP นั่นเอง ที่เวลา t = 7 เมื่อกราฟวิ่งสูงกว่า SP ค่า Error จะเป็นลบ เมื่อนำไปคูณกับ K_p จะทำให้ค่า P_{out} เป็นลบด้วย จึงไปทำการลดค่า Output นั่นเอง



รูปที่ 2.11 Proportional
(รูปภาพจาก https://automation360blog.wordpress.com/2018/01/01/pid-by-graph/)

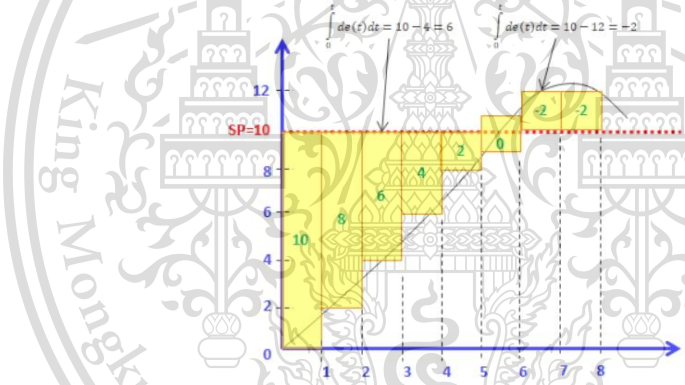
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ดังนั้น Proportional gain ที่มีค่ามากจึงส่งผลให้ Rise Time เปลี่ยนแปลงเร็ว (เพิ่มเร็ว ลดเร็ว) แต่ถ้าใส่มากเกินไปก็อาจจะให้เกิน Overshoot ได้ง่ายด้วยเช่นกัน

I (Integral) พื้นฐานของการ Integrate คือการหาพื้นที่ใต้กราฟนั่นเอง กรณีนี้เราจะหาพื้นที่ใต้กราฟของ error แทนการหาพื้นที่ใต้กราฟแบบปกติ

$$I_{out}=K_i \int_0^t e(\tau)d\tau$$

จากรูปข้างล่างจะเห็นว่า ระหว่างที่กราฟยังไม่ถึงจุด SP ผลรวมการ Integrate ซึ่งก็คือค่าใต้กราฟของตัว error จะมีค่าสะสมเพิ่มขึ้นเรื่อย ๆ ซึ่งเมื่อนำไปคูณกับ gain K_i ก็จะทำให้ไปเพิ่ม Output นั่นเอง แต่ควรระวังเพราะถ้าใส่ gain K_i มากเกินไปก็จะทำให้เกิด Overshoot ได้ง่ายขึ้น



รูปที่ 2.12 Integral

(รูปภาพจาก https://automation360blog.wordpress.com/2018/01/01/pid-by-graph/)

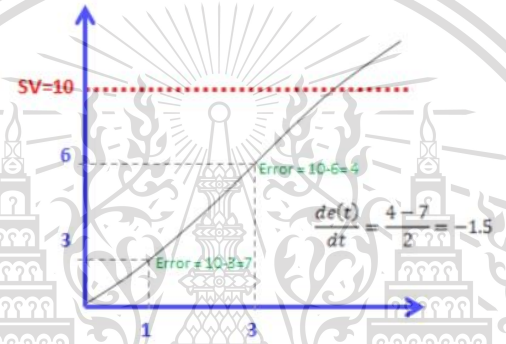
ตราบใดที่ค่า Output ยังอยู่ใต้ SP ค่า Integral ของ Error ก็จะมีรวมกันและเพิ่มขึ้นไปเรื่อย ๆ ซึ่งเมื่อนำไปคูณกับ K_i แล้วจะทำให้ค่าเป็นบวกและจะไปเพิ่ม Output ซึ่งทำให้กราฟยังคงวิ่งไปหา SP ต่อไป แต่ตราบใดที่ค่า Output อยู่เหนือ SP ค่า Integral ของ Error ก็จะมีค่าเป็นลบ ซึ่งเมื่อรวมกับผลรวมของ error ก่อนหน้าแล้ว ก็จะทำให้ผลรวมลดลงไปนั่นเอง และจะไปลด gain รวมให้ลดลง สมมติว่ากราฟของ output ยังคงสูงกว่า SP ไปเรื่อย ๆ ก็จะทำให้ gain รวมลดลงไปเรื่อย ๆเช่นกัน (เพราะตัวสูตรเป็นการ Integral ตั้งแต่เวลา 0 -> t ซึ่งแตกต่างจากตัวอื่นที่พิจารณาผล ณ เวลานั้น ๆ) ทำให้ Integral ส่งผลต่อการลด Steady State Error แต่ก็ส่งผลต่อ Overshoot ด้วยเช่นกันถ้าใส่ gain มากเกินไป (แต่ก็ลด rise time ได้ คือเพิ่ม-ลดได้เร็ว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

D (Derivative) พื้นฐานของการ Derivative คือการหา slope หรือความชันของกราฟนั่นเอง

$$D_{out} = K_d \frac{d}{dt} e(t)$$

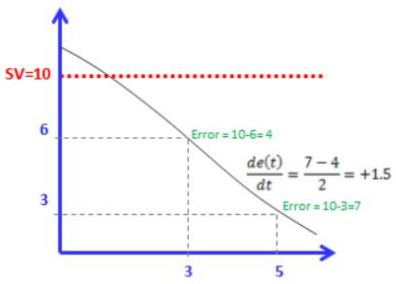
สังเกตว่าในขณะที่กราฟกำลังพุ่งขึ้น slope ของ Output จะเป็น + แต่เนื่องจากเรามองที่ค่า error จึงทำให้ค่า de(t)/dt ตอนกราฟพุ่งขึ้นมีค่าติดลบแทน เมื่อนำไปคูณกับ K_d จึงไปทำหน้าที่ให้ลด Output ลง ดังนั้นส่วนของ D_{out} จึงนำไปช่วยในการลดอัตราการเร่งของกราฟได้เมื่อกราฟกำลังเพิ่มขึ้น



รูปที่ 2.13 Derivative

(รูปภาพจาก https://automation360blog.wordpress.com/2018/01/01/pid-by-graph/)

สังเกตว่าในขณะที่กราฟกำลังลดลง slope ของ Output จะเป็น – แต่เนื่องจากเรามองที่ค่า error จึงทำให้ค่า de(t)/dt ตอนกราฟพุ่งขึ้นมีค่าเป็นบวกแทน เมื่อ feedback กลับไปจึงทำหน้าที่ไปเพิ่มค่าให้ Output ดังนั้นส่วนของ D_{out} จึงนำไปช่วยในการเพิ่มอัตราการเร่งของ graph ได้เมื่อกราฟกำลังลดลง

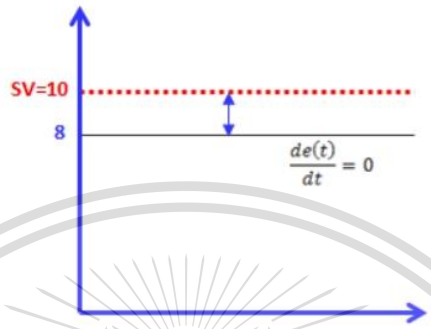


รูปที่ 2.14 Derivative

(รูปภาพจาก https://automation360blog.wordpress.com/2018/01/01/pid-by-graph/)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

กรณีที่กราฟราบเรียบ ความชัน หรือ de(t)/dt จะมีค่าเป็นศูนย์ parameter D จึงไม่ส่งผล กรณีที่กราฟนิ่งแล้วอย่างไรก็ตาม จากรูปจะเห็นว่าแม้ว่ากราฟจะนิ่งแต่ก็ยังมี error อยู่ จึงเห็นได้ว่าตัว Derivative จะไม่ช่วยลด error ได้เลย



รูปที่ 2.15 Derivative

(รูปภาพจาก https://automation360blog.wordpress.com/2018/01/01/pid-by-graph/)

กราฟยิ่งเพิ่มเร็ว D_{out} จะเป็นลบเยอะ จึงไปช่วยห้ามการเพิ่มขึ้นหรือ Overshoot ได้ดี กราฟยิ่งลดเร็ว D_{out} จะเป็นบวกเยอะ จึงไปช่วยห้ามการลดลงหรือ Undershoot ได้ดี หรือพูดง่ายๆคือ Derivative ช่วยต่อต้านการเพิ่มขึ้นหรือลดลงอย่างรวดเร็วได้ดี แต่พอกราฟนิ่ง D จะตรวจสอบไม่ได้เลยเพราะมีค่าเป็นศูนย์ จึงไม่ส่งผลต่อ Steady State Error

2.3.2 Arduino IDE

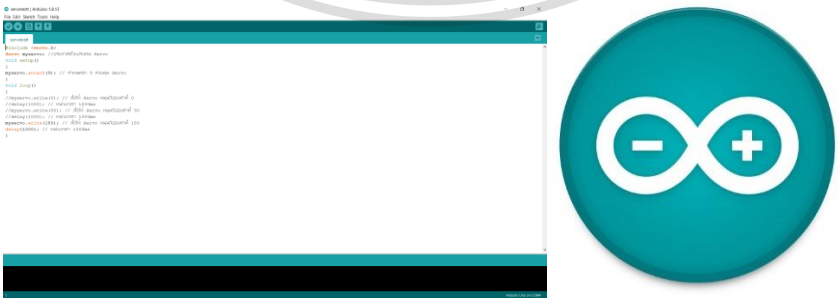
Arduino Integrated Development Environment (IDE) เป็นโปรแกรมที่ออกแบบมาเพื่อให้ง่ายต่อการเขียนซอฟต์แวร์สำหรับแพลตฟอร์มโอเพ่นซอร์สนี้ แพลตฟอร์ม Arduino เป็นแพลตฟอร์มอิเล็กทรอนิกส์ยอดนิยมที่ออกแบบมาเพื่อลดความซับซ้อนของกระบวนการออกแบบอุปกรณ์อิเล็กทรอนิกส์ การใช้งานทั่วไป ได้แก่ หุ่นยนต์เทคโนโลยีการปรับปรุงบ้านคอมพิวเตอร์ที่สวมใส่ได้และแอปพลิเคชันอิเล็กทรอนิกส์ที่แปลกใหม่ สิ่งประดิษฐ์ Arduino ส่วนใหญ่ได้รับการพัฒนาโดยใช้ Arduino IDE

IDE มักใช้โดยโปรแกรมเมอร์เพื่อเร่งกระบวนการเขียนโปรแกรม คุณสมบัติทั่วไปของ IDE รวมถึงการกำหนดหมายเลขบรรทัดอัตโนมัติการเน้นไวยากรณ์และการรวบรวมแบบรวม แม้ว่าจะเป็นไปได้ในทางเทคนิคที่จะเขียนซอฟต์แวร์โดยใช้โปรแกรมแก้ไขข้อความอย่างง่าย แต่กระบวนการนั้นง่ายกว่ามากเมื่อเขียนโค้ดใน IDE ภาษาการเขียนโปรแกรมจำนวนมากมี IDEs ของตนเองและมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

การพัฒนา IDE สำหรับวัตถุประสงค์ทั่วไปหลายอย่าง IDE วัตถุประสงค์ทั่วไปเหล่านี้สามารถใช้กับภาษาการเขียนโปรแกรมที่รองรับได้หลากหลาย Arduino IDE มีสภาพแวดล้อมที่อนุญาตให้โปรแกรมเมอร์ใช้โปรแกรมเดี่ยวตั้งแต่ต้นจนจบ มันสามารถติดตามไฟล์หลาย ๆ ไฟล์ในโครงการซึ่งช่วยให้โปรแกรมเมอร์เขียนโปรแกรมที่ซับซ้อนมากขึ้นหรือโมดูลาร์เพื่อจัดการโครงการของพวกเขา IDE ยังรวบรวมรหัสตัวเองทำการดีบักพื้นฐานและส่งรหัสโดยตรงไปยังบอร์ด Arduino ซึ่งจะใช้ bootloader Arduino เพื่อเขียนโปรแกรมใหม่ลงในหน่วยความจำ

แม้จะมีคุณสมบัติเพิ่มเติมเหล่านี้โปรแกรมเมอร์บางคนก็บ่นว่า Arduino IDE นั้นขาดความแจ่มใสเมื่อเทียบกับ IDE อื่น ๆ ที่ก้าวหน้ากว่า นี่เป็นเพราะมันขาดคุณสมบัติทั่วไปหลายประการรวมถึงการกำหนดหมายเลขบรรทัดที่มองเห็นโดยอัตโนมัติซึ่งจะช่วยให้โปรแกรมเมอร์สามารถอ้างถึงส่วนที่เฉพาะเจาะจงของซอร์สโค้ดได้ง่ายเมื่อประเมินข้อความแสดงข้อผิดพลาดหรือการสื่อสารกับโปรแกรมเมอร์อื่น ๆ คุณลักษณะที่ขาดหายไปอื่น ๆ ได้แก่ ข้อความแสดงข้อผิดพลาดโดยละเอียดซึ่งมีประโยชน์สำหรับการวินิจฉัยและแก้ไขข้อผิดพลาดในการเขียนโค้ดและการพับโค้ดซึ่งทำให้โปรแกรมเมอร์สามารถตรวจสอบเฉพาะบางส่วนของซอร์สโค้ดที่เกี่ยวข้องโดยการซ่อน pars ซึ่งไม่ได้รับผลกระทบ เพื่อจัดการกับข้อ จำกัด เหล่านี้โปรแกรมเมอร์ Arduino บางคนใช้ IDE อื่น ๆ เพื่อเขียนโปรแกรม ผู้ใช้เหล่านี้มีปลั๊กอินซอฟต์แวร์ที่เขียนขึ้นสำหรับ IDE วัตถุประสงค์ทั่วไปที่เพิ่มการสนับสนุนสำหรับการเขียนโปรแกรมเฉพาะArduino นี่เป็นการเพิ่มคุณสมบัติมากมายที่โปรแกรมเมอร์พลาดใน Arduino IDE แต่โซลูชันยังมาพร้อมกับข้อ จำกัด หลายประการ เพื่อรักษาความสามารถในการใช้ IDE ทั่วไปสำหรับรหัสArduinoโปรแกรมเมอร์จะต้องอัพเดตปลั๊กอินของตนเป็นประจำด้วยซอฟต์แวร์ Arduino รุ่นใหม่แต่ละตัว นอกจากนี้ IDE ทั่วไปเหล่านี้ไม่สามารถเชื่อมต่อกับบอร์ด Arduino ได้ดังนั้นจึงไม่สามารถใช้อัปโหลดซอฟต์แวร์ที่สมบูรณ์ไปยังสิ่งประดิษฐ์ Arduino ได้



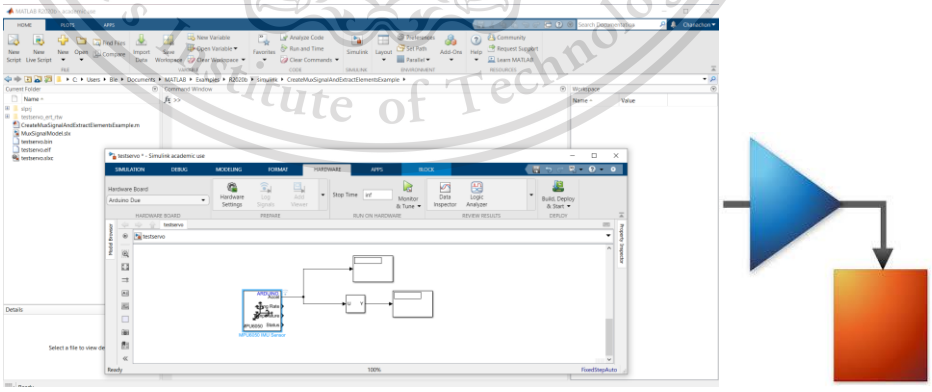
รูปที่ 2.15 โปรแกรม Arduino IDE
(รูปภาพจาก https://www.arduitronics.com/article/1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.3.2 Matlab Simulink

โปรแกรม Matlab สามารถจำลอง ทดสอบ และวิเคราะห์การทำงานของระบบพลศาสตร์ ในเชิงเวลา ได้โดยการใช้ Simulink ซึ่งเป็นเครื่องมือ Toolbox ที่อยู่ในโปรแกรม Matlab โดยจะทำงานภายใต้ หน้าต่างที่เป็นการเชื่อมต่อทางรูปภาพ (GUI) ของ Simulink เท่านั้น คำว่า Simulink มาจากคำสองคำคือ Simulation และ Link การใช้งาน Simulinkจะกระทำโดยการนำ Block ใน หน้าต่าง Library-Simulink มาต่อกัน ตามที่เราต้องการและสามารถจำลองระบบได้ทั้งระบบที่เป็นเชิงเส้น ไม่เป็นเชิงเส้น ระบบเวลาต่อเนื่องและไม่ต่อเนื่อง การจำลองระบบสามารถกระทำได้โดย ป้อนอินพุตที่ป้อนเข้าไป

Simulink เป็นซอฟต์แวร์ที่ทำงานอยู่บน Matlab ใช้ในการจำลองแบบทางคณิตศาสตร์ด้วยรูปภาพ สนับสนุนเครื่องมือสร้างแบบจำลอง การเลียนแบบ และเครื่องมือสำหรับวิเคราะห์ข้อมูล คุณสามารถทำแบบจำลองด้วยรูปภาพได้อย่างรวดเร็วเพื่อแสดงถึงการออกแบบแนวความคิดของระบบด้วยความพยายามเพียงเล็กน้อย นั่นคือ Simulink ติดต่อกับผู้ใช้ผ่านทางรูปภาพ หรือ GUI (Graphic User Interface) ในการสร้างไดอะแกรมของแบบจำลอง นอกจากนั้น Simulink ยังประกอบด้วยไลบรารีบล็อกพื้นฐานและขั้นสูงเฉพาะสาขาวิชา ทั้งระบบเชิงเส้น (Linear System) ระบบไม่เชิงเส้น (Nonlinear System) ระบบเวลาต่อเนื่อง (Continuous-time) ระบบที่เวลาแบบแซมปิ้ง (Sample time) ระบบไฮบริด (Hybrid) Simulink สนับสนุนการเปลี่ยนแปลงค่าพารามิเตอร์ได้ขณะที่เรากำลังเลียนแบบระบบอยู่ ทำให้เราเห็นการเปลี่ยนแปลงที่เกิดขึ้นได้ทันที ว่าอะไร ที่ไหน อย่างไร กับแบบจำลองระบบของเรา สุดท้าย Simulink สามารถเชื่อมต่อข้อมูลกับ Matlab ได้โดยตรง ทำให้เราสามารถเชื่อมต่อกับซอฟต์แวร์และฮาร์ดแวร์ภายนอกได้



รูปที่ 2.16 โปรแกรม Matlab Simulink

(รูปภาพจาก https://www.cleanpng.com/png-simulink-matlab-mathworks-computer-software-logo-c-1203429/)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 3
วิธีดำเนินการวิจัย

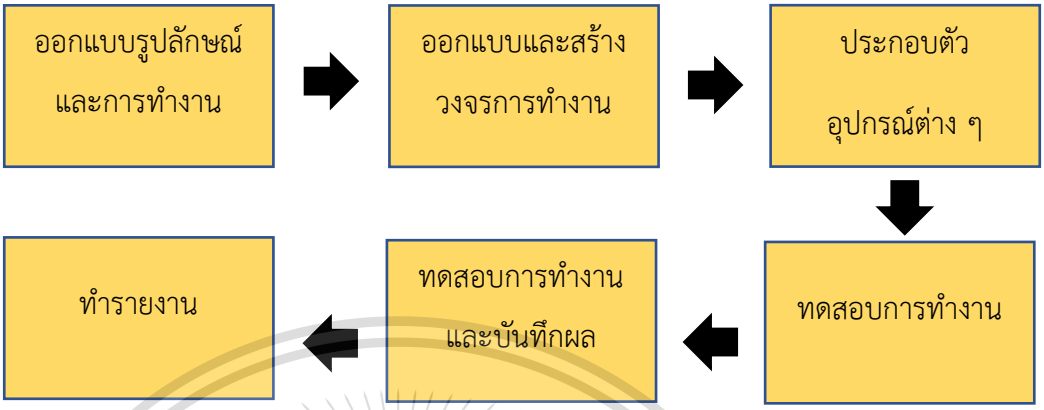
3.1 ศึกษาหาข้อมูล

ศึกษาข้อมูลการสร้างโครงสร้างของชิ้นงานเครื่องช่วยลดการสั่นสะเทือนสำหรับการขนส่ง ว่าสามารถใช้ข้อต่อหรือจุดที่ใช้สำหรับการหมุนของมอเตอร์ (servo) ได้มีแบบไหนบ้าง และทำการสรุปวิธีที่จะใช้ในการสร้างชิ้นงาน โดยการควบคุมการหมุนของมอเตอร์ (servo) จำเป็นต้องใช้ไมโครคอนโทรลเลอร์เข้ามาใช้ในการควบคุม เราจึงเลือกใช้ Arduino ในการออกคำสั่งควบคุมการทำงานของมอเตอร์ (servo) ทำการศึกษาเกี่ยวกับการเขียนคำสั่งของ Arduino เพื่อให้ชิ้นงานมีความนิ่ง สมดุล และสามารถทราบได้ว่าแผ่นวางวัตถุมีความลาดเอียงมากน้อยแค่ไหน จึงทำการศึกษาหาเซ็นเซอร์เพื่อมาวัดค่าความลาดเอียง โดยได้เลือกเซ็นเซอร์ MPU6050 GY-87 ซึ่งเป็นเซ็นเซอร์ที่สามารถวัดความลาดเอียงและสามารถวัดความเร่งของการเปลี่ยนแปลงของการลาดเอียง ได้ทำการศึกษาคำสั่งการทำงานของ MPU6050 GY-87 ว่ามีการทำงานอย่างไรบ้างและศึกษาของเขตการทำงานของเซ็นเซอร์ในโปรแกรม Arduino IDE ศึกษาคุณสมบัติการทำงานของมอเตอร์ (servo)ในโปรแกรม Arduino IDE และ Oscilloscope โดยเลือกใช้มอเตอร์ MG-995 ซึ่งเป็น servo motor ที่สามารถหมุนได้ 0 – 180 องศา โดยมอเตอร์ตัวนี้สามารถรับน้ำหนักได้ประมาณ 9 kg-cm ที่ไฟเลี้ยง 4.8 V ที่กระแสไฟตรง 0.5 A ต่อ 1 ตัว ชิ้นงานนี้ใช้มอเตอร์ 3 ตัวจึงจำเป็นต้องเปลี่ยนบอร์ด Arduino UNO R3 เป็น Arduino Due เนื่องจากพอร์ตเอาต์พุตสามารถจ่ายกระแสออกได้ถึง 800mA ซึ่งเพียงพอ ทำการศึกษาหลักการทำงานของ PID control ซึ่งเป็นระบบควบคุมแบบป้อนกลับโดยนำค่าความผิดพลาด (error) ที่เกิดขึ้น โดยค่าความผิดพลาดในชิ้นงานนี้เกิดจากมุมที่เซ็นเซอร์อ่านได้ลบกับมุมที่ขนานกับพื้นโลก โดยค่า PID จะช่วยให้ค่าความผิดพลาดเหลือน้อยที่สุดและใช้เวลาน้อยที่สุดในการเข้าสู่จุดสมดุล ได้ทำการศึกษาการจูนค่าคงที่ K_p , K_i , K_d โดยใช้โปรแกรม Matlab Simulink เพื่อได้ค่าที่สามารถทำให้ชิ้นงานนี้มีประสิทธิภาพในการทำงานมากที่สุด คือ ชิ้นงานนี้สามารถเข้าสู่จุดสมดุลและสามารถรักษาเสถียรภาพได้ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.2 ขั้นตอนการดำเนินงาน

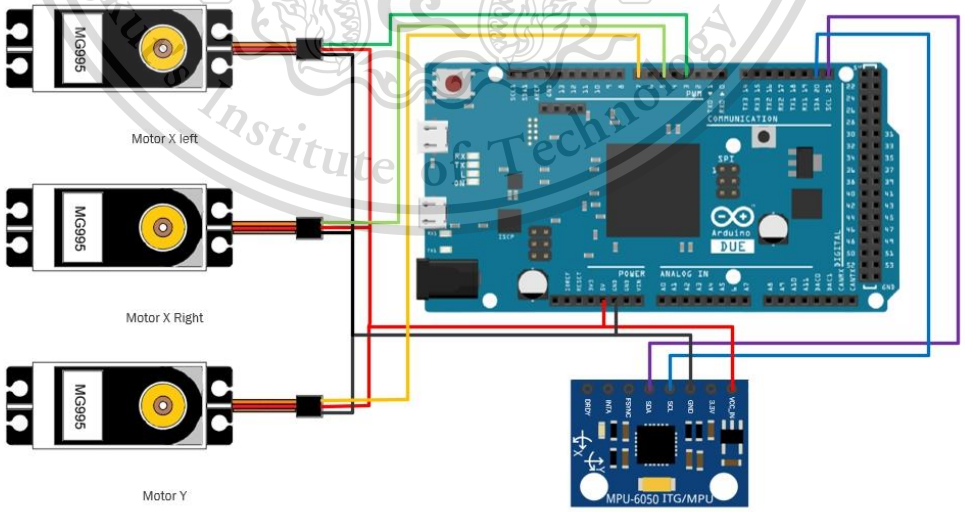
3.2.1 วางแผนการทำตัวงาน



รูปที่ 3.1 บล็อกไดอะแกรมแสดงขั้นตอนการดำเนินงาน

เริ่มออกแบบโดยเขียนบล็อกไดอะแกรมวางแผนงาน โดยทุก ๆ ขั้นตอนนั้นจะเข้าพบอาจารย์ที่ปรึกษาทุกครั้งเพื่อให้อาจารย์ตรวจสอบและแก้ไขข้อผิดพลาดที่เกิดขึ้นก่อนจะดำเนินงานในขั้นตอนต่อไป

3.2.2 ออกแบบต่อวงจรของตัวอุปกรณ์



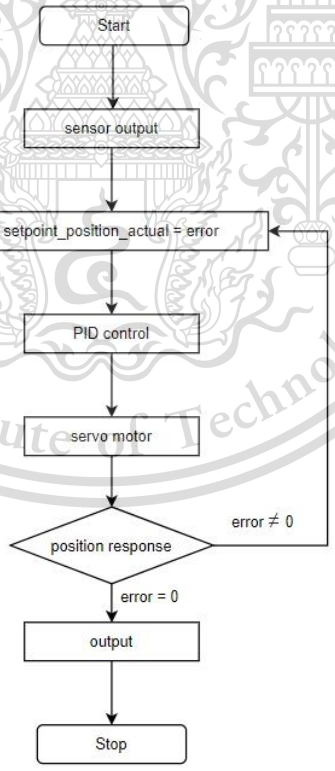
รูปที่ 3.2 การต่อวงจรรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ตารางที่ 1 แสดงการต่อกันของ Servo Motor, MPU6050 GY-87 และ Arduino Due

Servo Motor1 X axis left	Servo Motor X axis right	Servo Motor Y axis	MPU6050 GY-87	Arduino DUE
VCC	VCC	VCC	VCC_IN	5V
GND	GND	GND	GND	GND
-	-	-	SCL	SCL21
-	-	-	SCA	SCA20
PWM Input	-	-	-	PWM3
-	PWM Input	-	-	PWM5
-	-	PWM Input	-	PWM7

3.2.3 ออกแบบการทำงานของตัวอุปกรณ์



รูปที่ 3.3 แผนผังการทำงานเบื้องต้นของตัวโปรแกรมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

1. โดยเริ่มแรกที่จะทำการเรียกใช้ไลบารี่หรือฟังก์ชั่นที่เกี่ยวข้องได้แก่ มอเตอร์Servo ,การสื่อสารแบบ I2C และ เซ็นเซอร์MPU6050

```
#include <Servo.h>
#include "Wire.h"
#include "I2Cdev.h"
#include "MPU6050.h"
MPU6050 mpu;
```

รูปที่ 3.4 code1

2. เป็นการประกาศตัวแปรชื่อต่างๆ เช่น ชื่อเรียกเซอร์โวมอเตอร์ตัวบน-ล่าง และกำหนดค่าต่าง ๆ เช่น ค่าตำแหน่งจุดset_point หรือค่าสัมประสิทธิ์ของ PID

```
int16_t ax, ay, az;
int16_t gx, gy, gz;
Servo BON;
Servo LANG_LEFT;
Servo LANG_RIGHT;
double kPX = 0.54385, kIX = 45.7933 , kDX = 0.015 ;
double kPY = 1.2156, kIY = 44.6973 , kDY = 0.0089 ;
double PX = 0 , IX = 0 , DX = 0 , PY = 0 , IY = 0 , DY = 0 , PIDX , PIDY ;
double error_X = 0 , previous_error_X = 0 , previous_error_Y = 0, previous_I = 0 , error_Y = 0  ;
double setpoint_X = 0 , setpoint_Y = 0 , setpoint_Z = 168 ;
double actual_position_X = 0 , actual_position_Y = 0 , actual_position_Z = 0 ;
```

รูปที่ 3.5 code2

3. เป็นการกำหนดค่าต่าง ๆ คืออัตราบอดของการรับส่งข้อมูล หน่วยเป็นบิตต่อวินาทีอยู่ที่ความเร็วในการรับส่งสูงสุดที่ 115200 BaudRate จากนั้นประกาศเลือกขาสำหรับต่อสัญญาณการทำงานคือ เซอร์โวมอเตอร์ตัวบนต่อกับที่ขา 7 ของบอร์ดอาดูโน่, เซอร์โวมอเตอร์ตัวล่างขวาต่อกับที่ขา 5 ของบอร์ดอาดูโน่และเซอร์โวมอเตอร์ตัวล่างซ้ายต่อกับที่ขา 3 ของบอร์ดอาดูโน่ แล้วปรับมุมการหมุนของเซอร์โวมอเตอร์ให้อยู่ตรงกลาง 90 องศาโดยกำหนดจากการกำหนดค่าความกว้างพัลซ์ไว้ที่ 1500 ms

```
void setup()
{
  Wire.begin();
  Serial.begin(115200);
  BON.attach(7);
  LANG_LEFT.attach(3);
  LANG_RIGHT.attach(5);
  BON.writeMicroseconds(1500);
  LANG_LEFT.writeMicroseconds(1500);
  LANG_RIGHT.writeMicroseconds(1500);
```

รูปที่ 3.6 code3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4. เป็นส่วนของการอ่านค่าจากตัวเซ็นเซอร์ 16bits มาแปลงเป็นค่ามุมที่เปลี่ยนแปลงในหน่วยองศา และ delay ในการอ่านค่าของเซ็นเซอร์ มีความช้ากว่า delay การทำงานของมอเตอร์ เนื่องจากต้องการให้มอเตอร์ทำงานก่อนจึงค่อยอ่านค่า error ตามภายหลัง

```
void loop()
{
  Vector normAccel = mpu.readNormalizeAccel();
  int ax = -(atan2(normAccel.XAxis, sqrt(normAccel.YAxis * normAccel.YAxis + normAccel.ZAxis * normAccel.ZAxis)) * 180.0) / M_PI;
  int ay = (atan2(normAccel.YAxis, normAccel.ZAxis) * 180.0) / M_PI;
  delay(180);
```

รูปที่ 3.7 code4

5. โดยจะอธิบายจากบรรทัดแรกคือมุมองศาปัจจุบันที่เซ็นเซอร์แกน X สามารถอ่านได้แล้วบรรทัดถัดมาคือ error = ค่าสุดสมดุลที่ตั้งค่าไว้ในแกน X (มอเตอร์ตัวบน) - มุมองศาปัจจุบันที่เซ็นเซอร์แกน X อ่านได้ ต่อมาเป็นส่วนของสมการของ P (Proportional) คือค่าคงที่ k_pX ที่ตั้งค่าไว้ * errorที่เกิดขึ้นระหว่างค่าสุดสมดุลที่ตั้งค่าไว้ในแกน X กับ มุมองศาปัจจุบันที่เซ็นเซอร์แกน X ถัดมาคือสมการของ I (Integral) เพื่อลด offset ที่เกิดขึ้นจากค่า k_pX ที่มากเกิน ต่อมาสมการของ D (Derivative) คือการลดค่า error ที่เกิดขึ้นโดยการนำ $error_{ปัจจุบัน}$ - $error_{ก่อนหน้า}$ เพื่อลดการแกว่ง ในส่วนสุดท้ายของสมการหลักPID คือ นำค่า P I D มาบวกกันเพื่อไปควบคุมมอเตอร์

```
actual_position_X = ax ;
error_X = setpoint_X - actual_position_X ;
PX = kPX * error_X;
IX = IX + (kIX * error_X);
DX = kDX * (error_X - previous_error_X);
PIDX = PX + IX + DX;
if (PIDX < -550)
{
  PIDX = -550;
}
if (PIDX > 550)
{
  PIDX = 550;
}
```

```
actual_position_Y = ay ;
error_Y = setpoint_Y - actual_position_Y ;
PY = kPY * error_Y;
IY = IY + (kIY * error_Y);
DY = kDY * (error_Y - previous_error_Y);
PIDY = PY + IY + DY;
if (PIDY < -550)
{
  PIDY = -550;
}
if (PIDY > 550)
{
  PIDY = 550;
}
```

รูปที่ 3.8 code5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

<u>หมายเหตุ</u> การทำงานของมอเตอร์เกิดจาก (1500+PID) 1500ms คือมุม 90 องศาและองศามากที่สุดคือ 950ms และ 2050ms ซึ่งอุปกรณ์สามารถหมุนได้มากสุดที่จะไม่ทำให้อุปกรณ์เสียหายอยู่ที่

ไม่เกิน 57องศา และ 123 องศาการกำหนดค่ามากสุดของ PID เพื่อไม่ให้เกิดการหมุนมากเกินองศาที่อาจทำให้อุปกรณ์เสียหาย

6. เป็นส่วนกานำค่าที่ได้จากPID ไปควบคุมเซอร์โวมอเตอร์โดยใช้การกำหนดการส่งความกว้างของพัลส์มากำหนดมุมการหมุน โดยค่ามอเตอร์ตัวบน 1500ms(ค่า$_{สมดุลมอเตอร์}$) + PID $_{ที่คำนวณแกนX}$ และค่ามอเตอร์ตัวล่าง 1500ms(ค่า$_{สมดุลมอเตอร์}$) - PID $_{ที่คำนวณแกนY}$ (ที่ต้องลบค่าPIDเนื่องจากวางเซ็นเซอร์ในทางตรงกันข้าม)กำหนดให้ค่าerror$_{ก่อนหน้า}$ ให้เท่ากับ error $_{ปัจจุบัน}$ เพื่อ Loop กลับไปคำนวณ FEED BACK ในแกนทั้งสองแกน

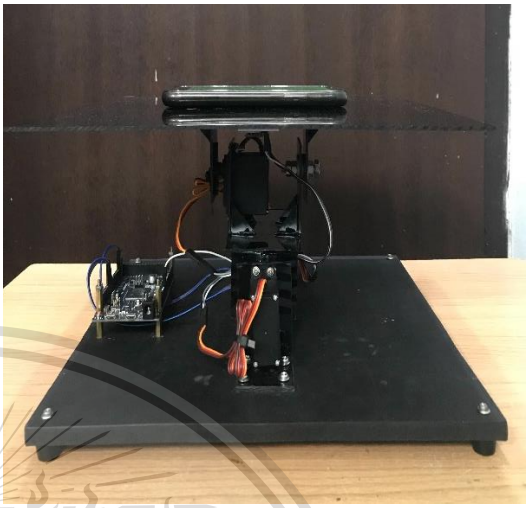
```
BON.writeMicroseconds(1450 - PIDY);
LANG_LEFT.writeMicroseconds(1450 - PIDX);
LANG_RIGHT.writeMicroseconds(1380 + PIDX);
previous_error_X = error_X ;
previous_error_Y = error_Y ;
```

รูปที่ 3.9 code6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.3.3 ออกแบบและประกอบตัวอุปกรณ์ตัวอุปกรณ์

ก.



ข.

ค.

รูปที่ 3.10 หลังประกอบอุปกรณ์

ก. ด้านหน้า

ข. ด้านข้าง

ค. ด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 4

การทดลองและผลการทดลอง

4.1 คุณสมบัติของอุปกรณ์ลดการสั่นสะเทือนสำหรับการขนส่งตามที่ตั้งไว้

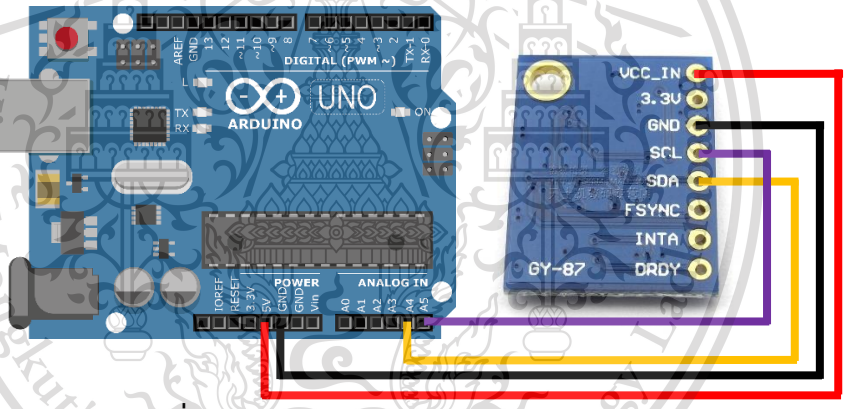
- สามารถปรับสมดุลได้ 2 แกน คือ แกนX และแกนY
- ศึกษาส่วนประกอบต่าง ๆ ที่นำมาใช้

4.2 การทดลองการอ่านค่าเซ็นเซอร์

4.2.1 วิธีทำการทดลอง

ทำการทดลองเพื่อหาคุณสมบัติของค่าที่อ่านได้ของIC MPU6050 วัดความเร่งและไจโรสโคป

โดย 1. ต่อบอร์ด Arduino กับตัวเซ็นเซอร์ตามรูปที่ 4.1



รูปที่ 4.1 การต่อ GY-87 กับบอร์ด Arduino

(ข้อมูลจาก : https://pixabay.com , Seven_auและ

https://www.yoycart.com/Product/27358144796/)

2. จากนั้นทำการใส่ code ดังรูปที่ 4.2 กด RUN แล้วเปิด Serial Monitor
3. สังเกตค่าเอาต์พุตที่ได้จากตัวเซ็นเซอร์ และวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

```
#include <Wire.h>
#include <MPU6050.h>
MPU6050 mpu;
void setup()
{ Serial.begin(115200);
  Serial.println("Initialize MPU6050");
  while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))
  {Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
    delay(500);}
void checkSettings()
{ Serial.println();
  Serial.print(" * Accelerometer:          ");
  switch(mpu.getRange())
  { case MPU6050_RANGE_16G:            Serial.println("+/- 16 g"); break;
    case MPU6050_RANGE_8G:             Serial.println("+/- 8 g"); break;
    case MPU6050_RANGE_4G:             Serial.println("+/- 4 g"); break;
    case MPU6050_RANGE_2G:             Serial.println("+/- 2 g"); break;
  }  }
void loop()
{ Vector rawAccel = mpu.readRawAccel();
  Serial.print(" Xraw = "); Serial.print(rawAccel.XAxis);
  Serial.print(" Yraw = "); Serial.print(rawAccel.YAxis);
  Serial.print(" Zraw = "); Serial.println(rawAccel.ZAxis);
  delay(1000);}
```

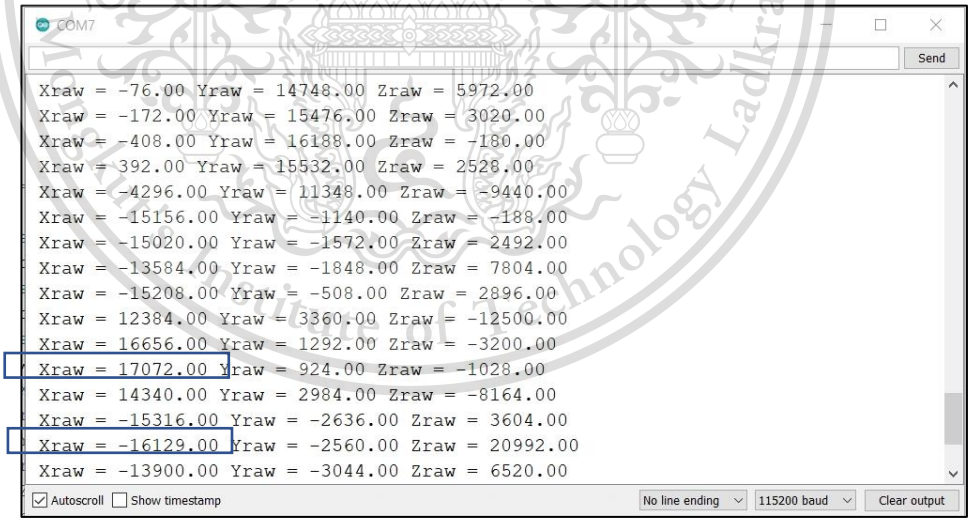
รูปที่ 4.2 Code MPU6050

4.2.2 ผลการทดลอง

โดยจากคำสั่งที่ส่งไปคือ วัดค่าที่ออกมาจากตัวเซ็นเซอร์บน Serial Monitor ได้ค่าดังนี้

ย่าน1g หันปกติ

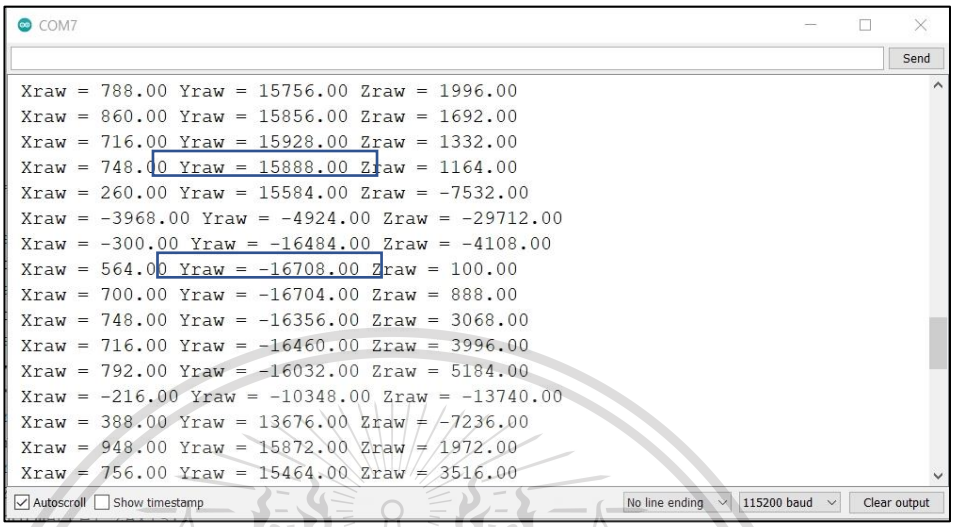
ค่าที่อ่านได้ในแกนX มีช่วงอยู่ที่ประมาณ 17000 ถึง -16200



รูปที่ 4.3 ค่าที่แสดงของแกน X ย่าน1g

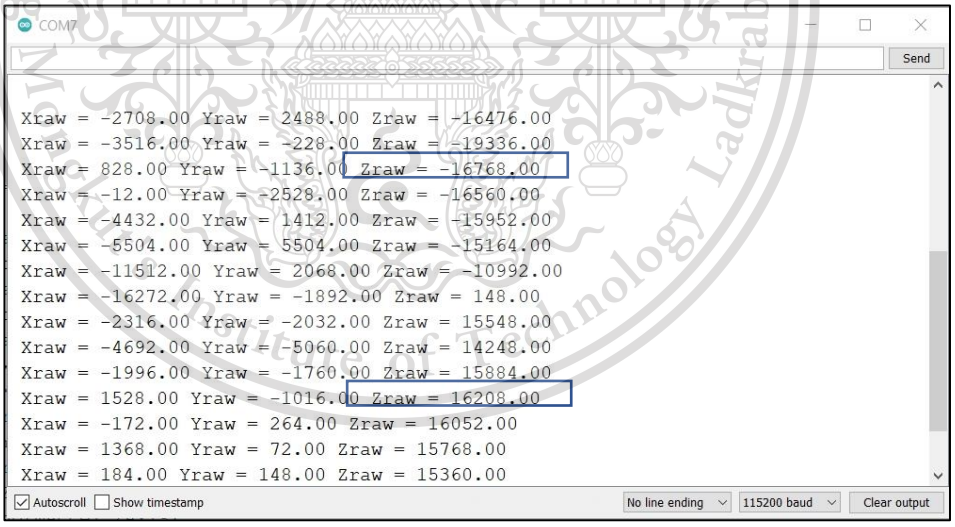
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ค่าที่อ่านได้ในแกนY มีช่วงอยู่ที่ประมาณ 15900 ถึง -16800



รูปที่ 4.4 ค่าที่แสดงของแกน Y ย่าน1g

ค่าที่อ่านได้ในแกนZ มีช่วงอยู่ที่ประมาณ 16200 ถึง -16700



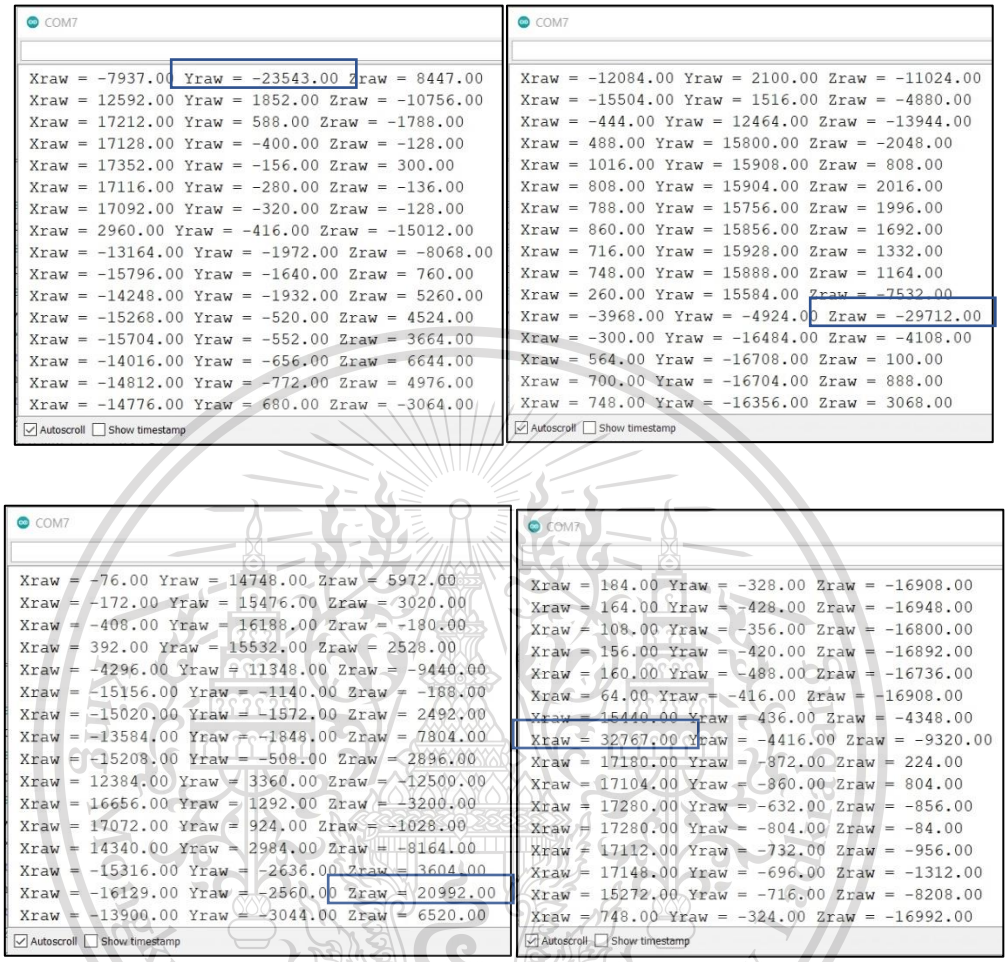
รูปที่ 4.5 ค่าที่แสดงของแกน Z ย่าน1g

จากการทดลองพบช่วงของค่าที่อ่านค่าได้ดังนี้

ช่วงแกนX คือ 33200 ช่วงแกนY คือ 33700 ช่วงแกนZ คือ 32900

และในการทดลองในย่าน 1g พบค่าที่อยู่ในช่วง 2g ด้วยเนื่องจากหมุนตัวเซ็นเซอร์ด้วยความเร็วที่สูงกว่าปกติมากได้ค่าดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 4.6 ค่าที่แสดงของแกนย่าน2g

จากทดลองนี้พบว่าค่าที่อ่านได้ของเซ็นเซอร์อยู่ในช่วง 16bits (2^{16}=65536) หรือในช่วง 32768 ถึง -32767 และได้แบ่งออกเป็น2ย่านคือ 1g กับ 2g ย่านละ 15bits (2^{15}=32768) ซึ่งค่าช่วงที่ได้จากการทดลองต่างจากค่าของ 15 bits คือ แกนX=432 แกนY=932 แกนZ=132 ค่าช่วงของค่าความคลาดเคลื่อนซึ่งอยู่ในช่วงพอรับได้เมื่อนำมาแปลงเป็นองศา

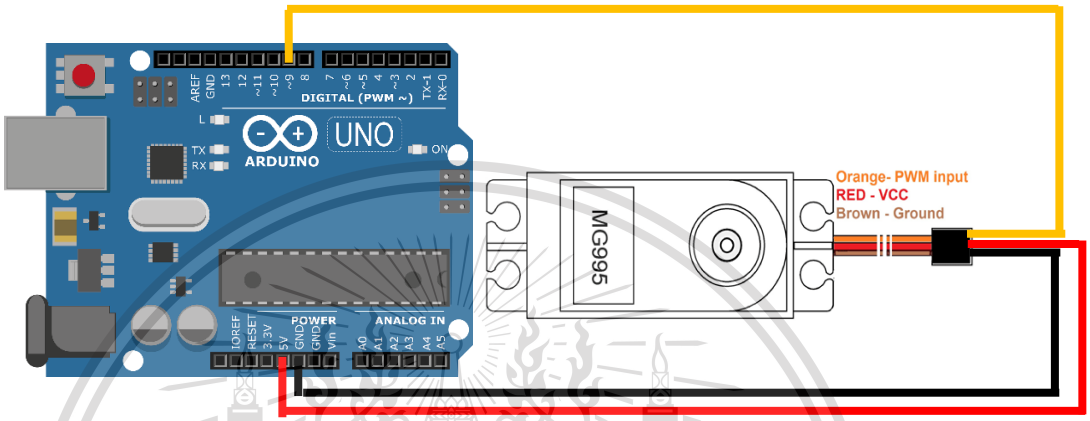
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.3 การทดลองการวัดสัญญาณควบคุมเซอร์โวมอเตอร์

4.3.1 วิธีการทดลอง

ทำการทดลองเพื่อหา คุณสมบัติของตัวมอเตอร์เพื่อการควบคุมที่มีความละเอียดมากขึ้น

โดย 1. ต่อบอร์ด Arduino กับตัวมอเตอร์ตามรูปที่ 3.3



รูปที่ 4.7 การต่อ MG995 กับบอร์ด Arduino

(ข้อมูลจาก : https://pixabay.com , Seven_au

และ https://components101.com/motors/mg995-servo-motor)

2. จากนั้นต่อสโคปเพื่อทำการวัดที่ขา 9 ของ Arduino ขนาดของ PWM input
3. จากนั้นทำการใส่ code ดังรูปที่ 4.8
4. บันทึกรูปกราฟที่แสดง และดูค่า +Width ของกราฟมาเปรียบเทียบ

```
#include <Servo.h>
Servo myservo;
void setup()
{ myservo.attach(9); }
void loop()
{
  myservo.write(0); delay(1000);
  myservo.write(45); delay(1000);
  myservo.write(90); delay(1000);
  myservo.write(135); delay(1000);
  myservo.write(180); delay(1000);
}
```

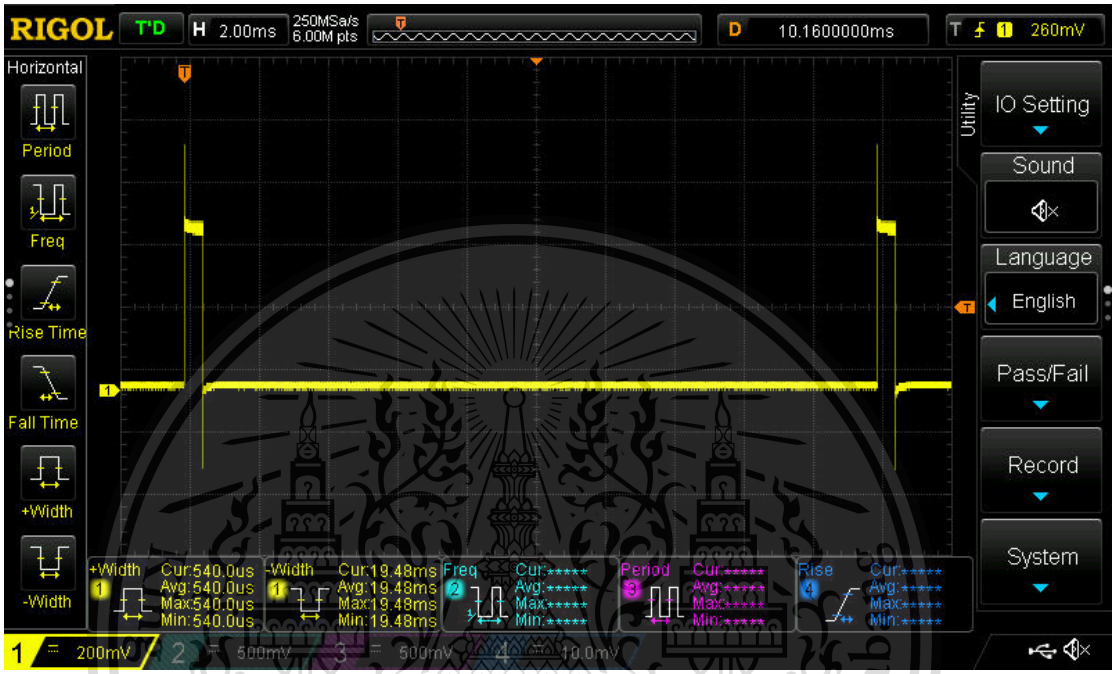
รูปที่ 4.8 Code MG995

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

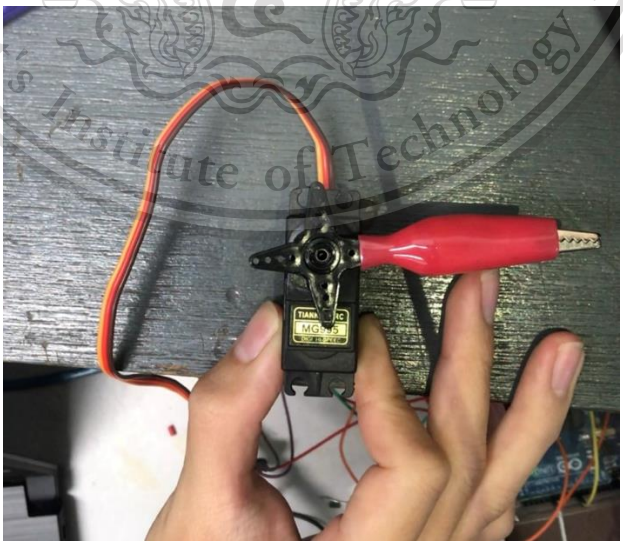
4.3.2 ผลการทดลอง

สัญญาณที่ใช้ควบคุมมุมของมอเตอร์ไฟฟ้ากระแสตรงนี้ มีความถี่ 50 Hz นั่นคือมี คาบเวลาเท่ากับ 20ms ซึ่งป้อนเป็นสัญญาณ PWM ที่มี Duty cycle แตกต่างกันไป ดังนี้

<u>ที่มุม 0 องศา</u>



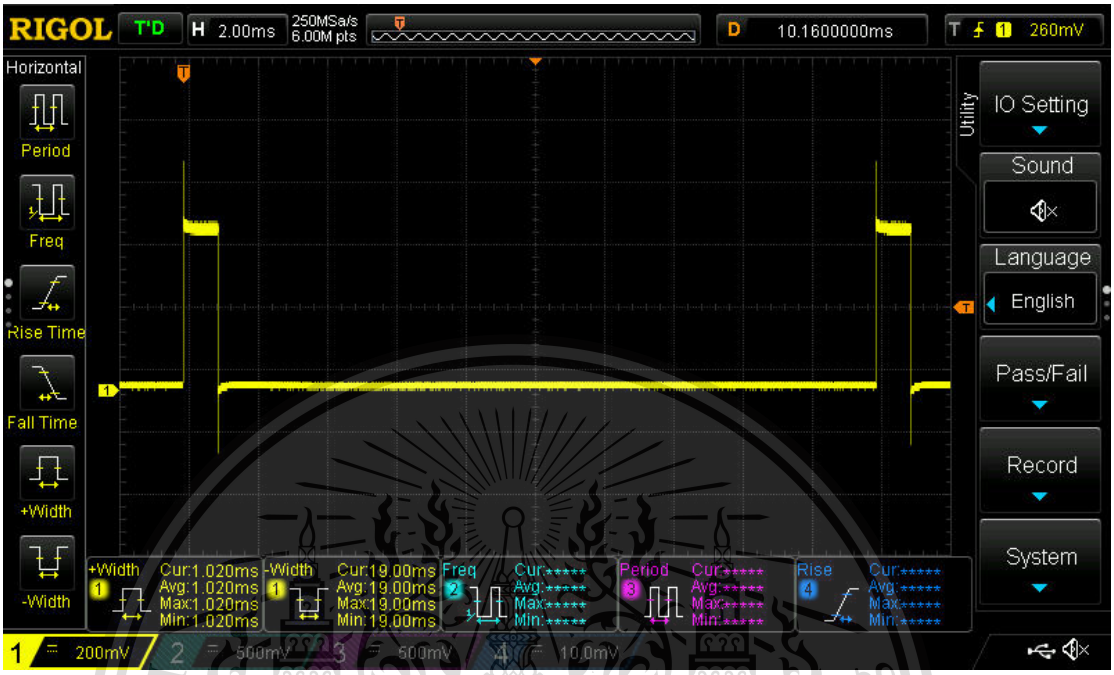
รูปที่ 4.9 ความกว้างสัญญาณ PWM คือ 540µs



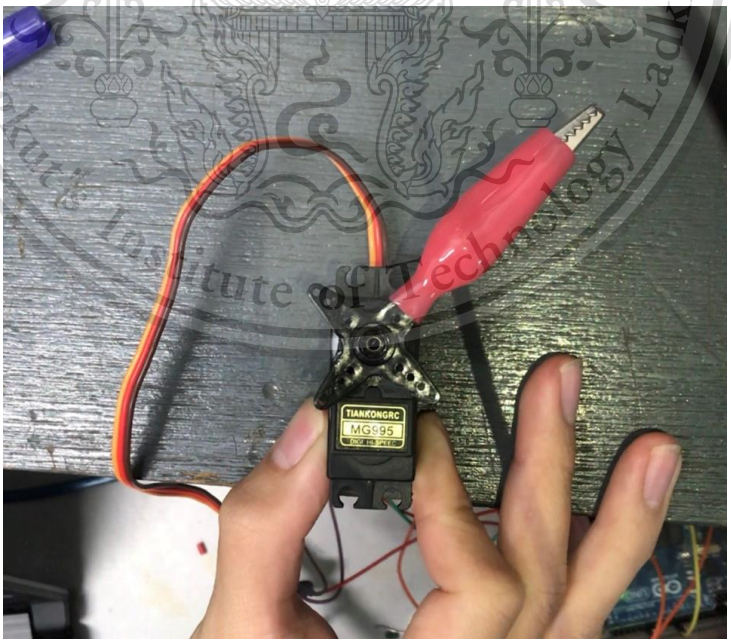
รูปที่ 4.10 มอเตอร์อยู่ที่ 0องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ที่มุม 45 องศา



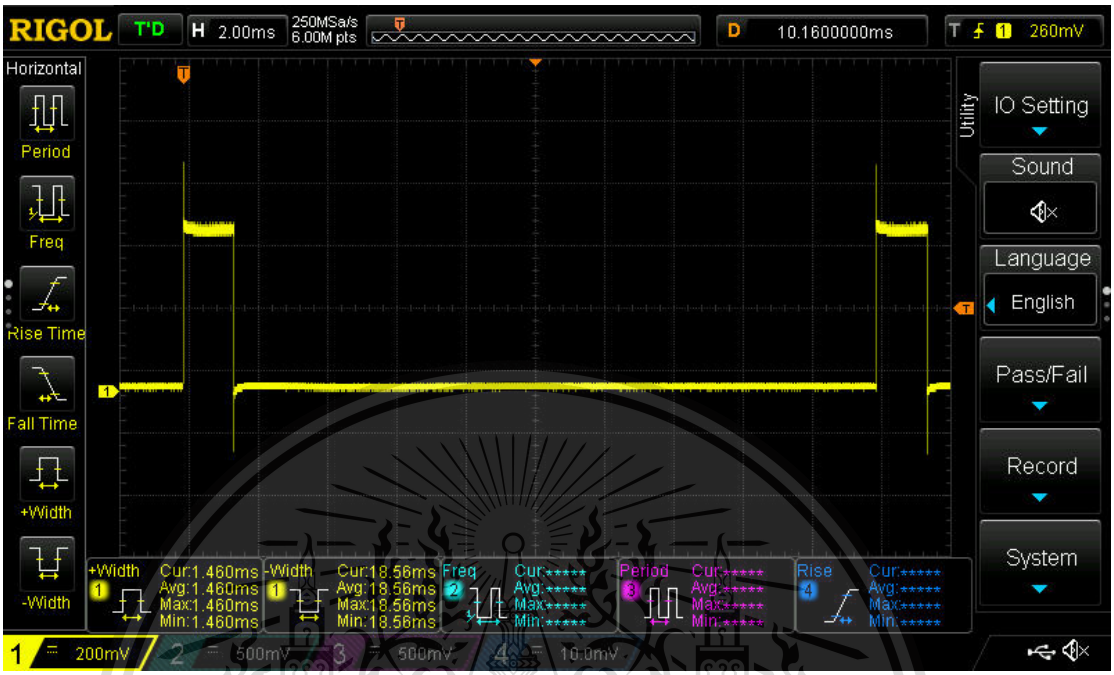
รูปที่ 4.11 ความกว้างสัญญาณ PWM คือ 1020μs



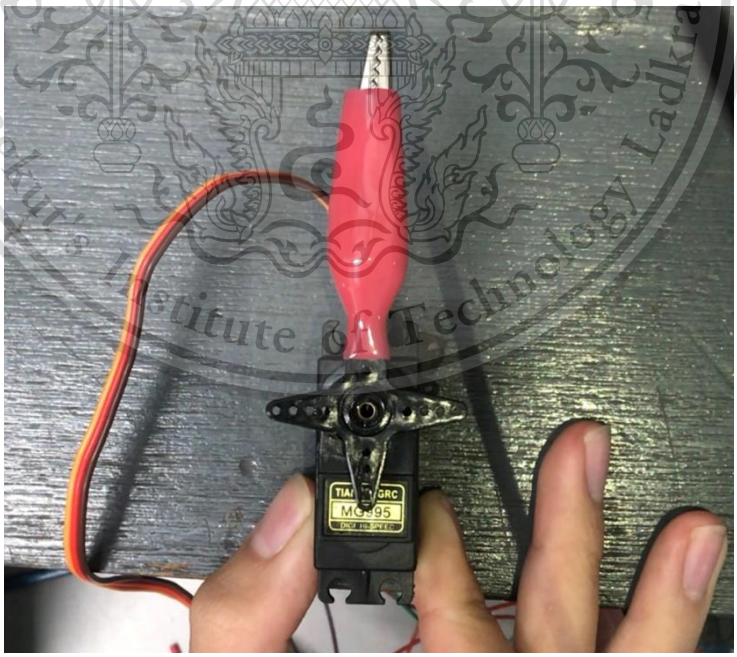
รูปที่ 4.12 มอเตอร์อยู่ที่ 45องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ที่มุม 90 องศา



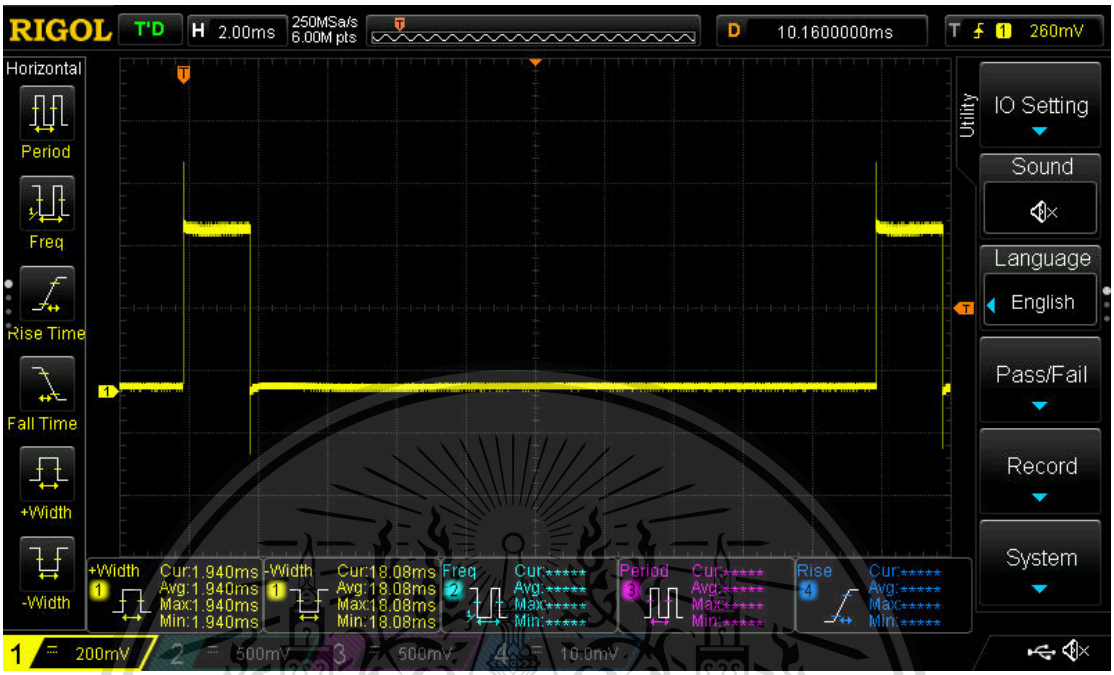
รูปที่ 4.13 ความกว้างสัญญาณ PWM คือ 1460µs



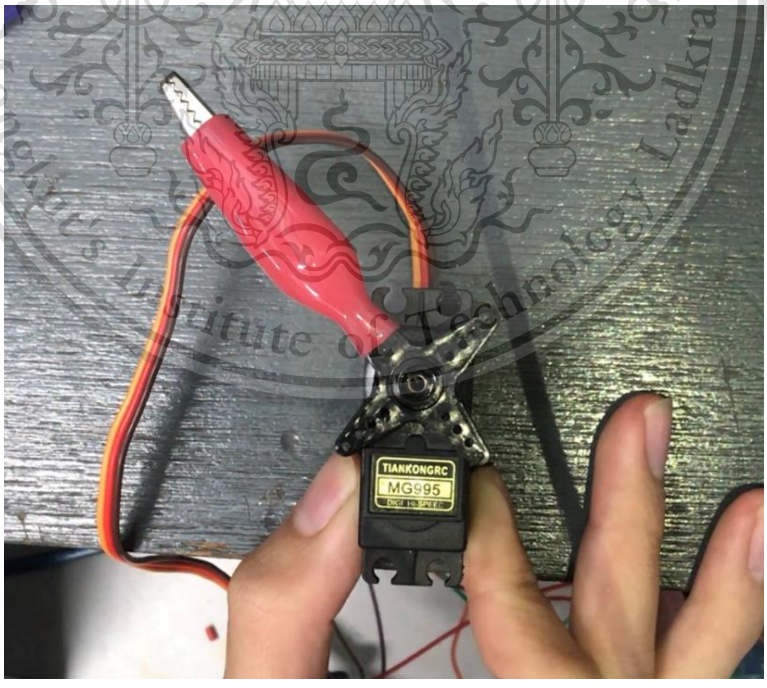
รูปที่ 4.14 มอเตอร์อยู่ที่ 90องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ที่มุม 135 องศา



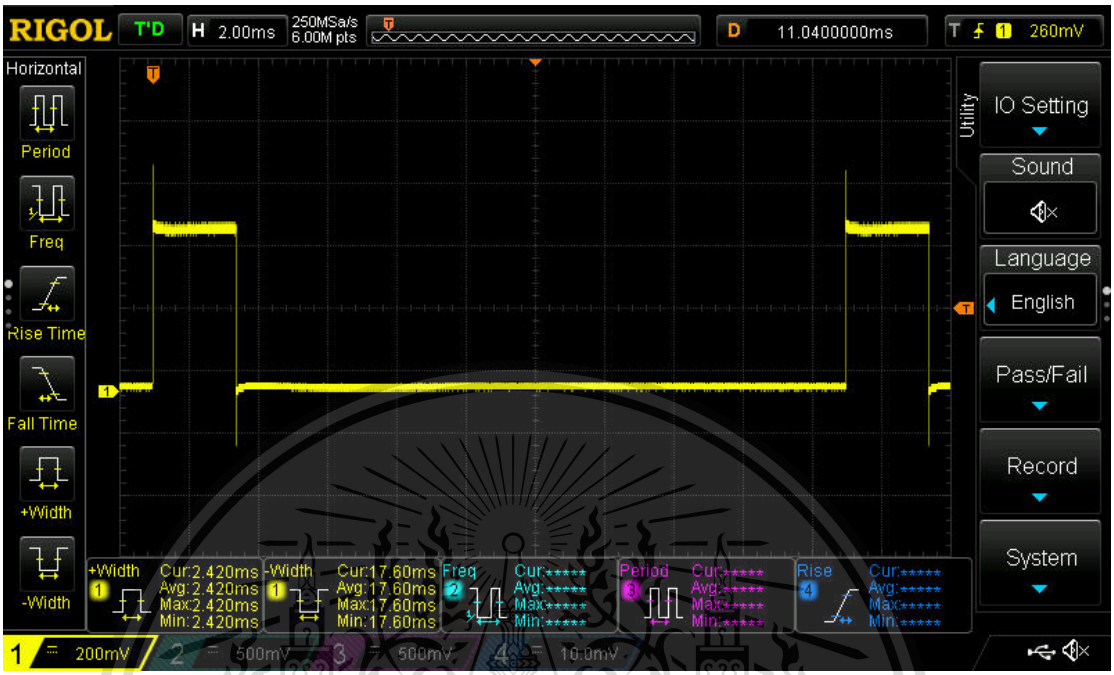
รูปที่ 4.15 ความกว้างสัญญาณ PWM คือ 1940µs



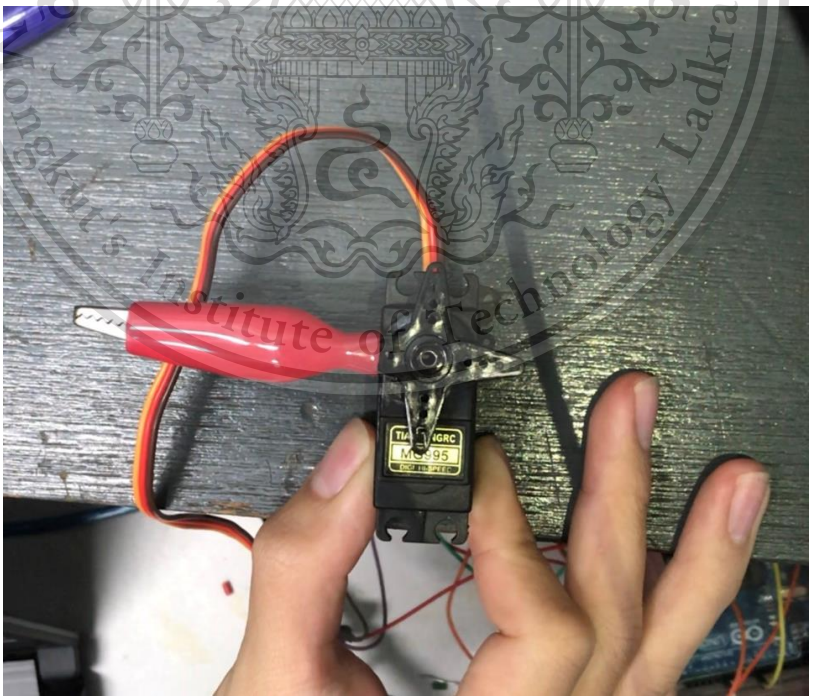
รูปที่ 4.16 มอเตอร์อยู่ที่ 135องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ที่มุม 180 องศา



รูปที่ 4.17 ความกว้างสัญญาณ PWM คือ 2420μs



รูปที่ 4.18 มอเตอร์อยู่ที่ 0องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.4 การทดลอง PID Tune

4.4.1 วิธีการทดลอง

ทำการทดลองเพื่อหา การเลือกค่าที่เหมาะสมจะทำให้ระบบสมดุลและได้ผลลัพธ์ตรงตามต้องการ

โดย 1. เปิดโปรแกรม Matlab แล้วไปเลือกเปิด Simulink ที่แทบด้านบน

2. กด Blank Model เพื่อสร้างไฟล์และกดบันทึก

3. กด Add on สำหรับส่วนของ Arduino หรือส่วนต่าง ๆ เพื่อให้โปรแกรมรับรองการเชื่อม

4. ไปที่ Model Settings แล้วไปตั้งค่าบอร์ดที่เชื่อมละพอร์ทที่เชื่อมและstop time เป็น inf

5. เลือก Library browser แล้วลากอุปกรณ์ให้มาตามภาพที่ 4.19

6. กด run on hardware เพื่อนำโค้ดไดอะแกรมไปลงบอร์ด

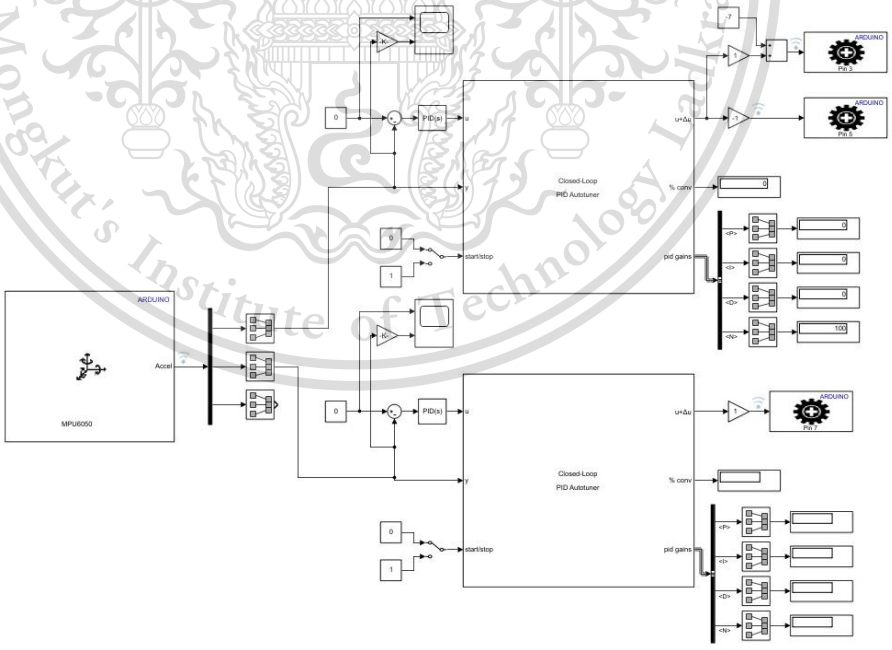
7. ดับเบิ้ลคลิ๊กที่ Switch ให้ไปที่ 1 เพื่อให้โปรแกรมคำนวณค่า K_P, K_I , K_D

8. รอที่ display ของพอร์ต %conv มีค่ามากกว่า90 ขึ้นไปไม่ลงเพื่อความชัวร์ จากนั้นสลับ Switch ไปที่ 0

9. ค่า K_P, K_I , K_D จะไปปรากฏที่ display ของพอร์ต pid gains

10. ดับเบิ้ลคลิ๊กที่ Closed-loop PID Autotuner ไปที่ Block แล้วกด Update PID Block

11. ทำไปเรื่อยๆ จนค่า K_P, K_I , K_D ที่ได้ นั้นทำให้ตัวชิ้นงานสมดุลที่สุดและนิ่งที่สุด



รูปที่ 4.19 ไดอะแกรมใน Simulink

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.4.2 ผลการทดลอง

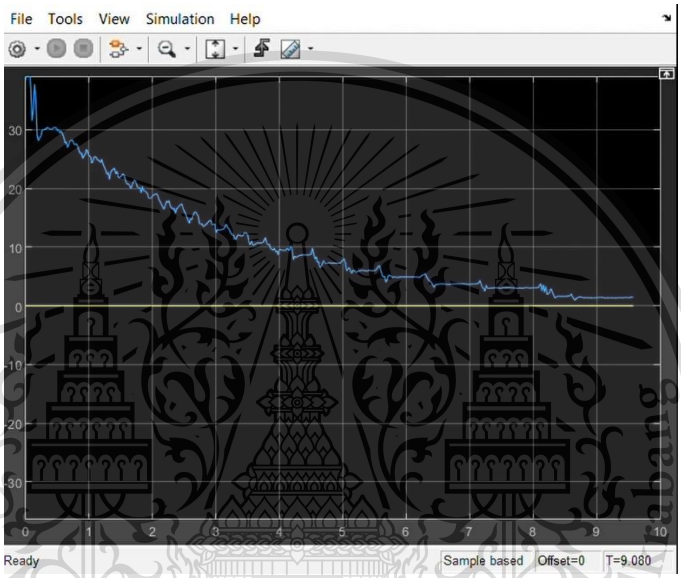
การคำนวณและพิจารณาค่า PID จากการทำลองการพิจารณาค่า K_P K_I K_D จาก

$$P + I\frac{1}{s} + D\frac{N}{1+\frac{1}{s}}$$

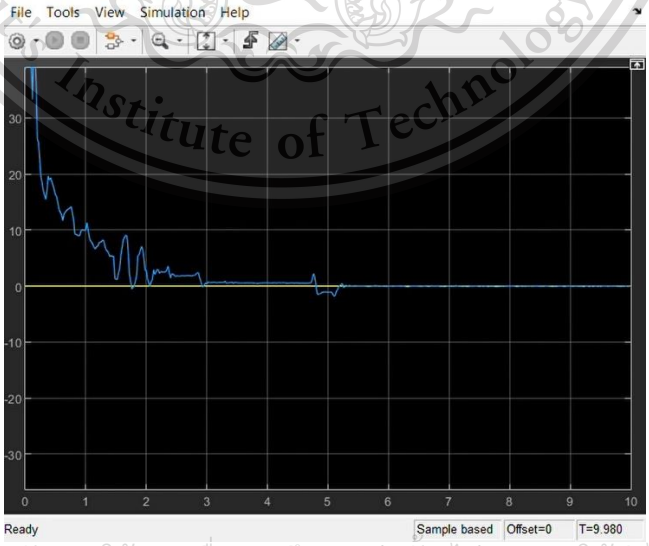
ใน Continuous-time และค่าที่เหมาะสมที่ทำให้สมดุลและนิ่งที่สุดคือ

ในแกนX K_P = 0.54385 , K_I = 45.7933 , K_D = 0.015

ในแกนY K_P = 1.2156 , K_I = 44.6973 , K_D = 0.0089

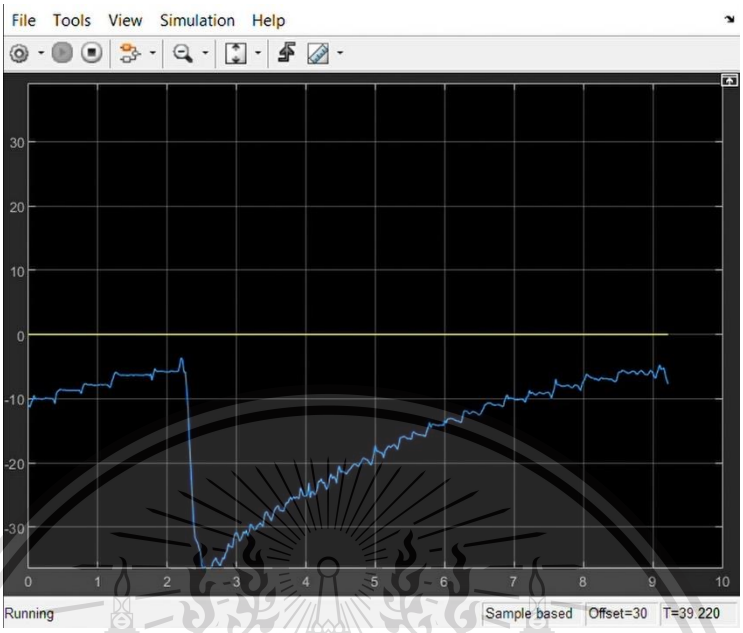


รูปที่ 4.20 กราฟในแกน X ก่อนปรับ (เอียง30องศา)

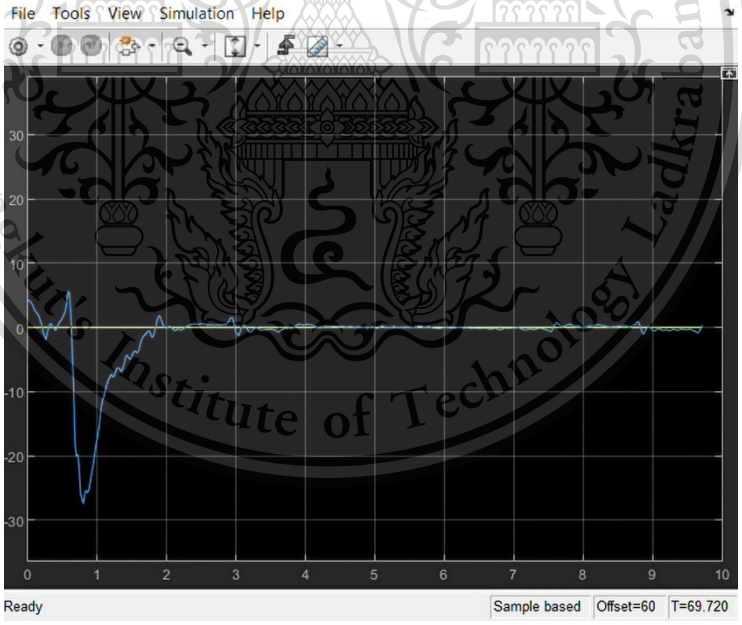


รูปที่ 4.21 กราฟในแกน X หลังปรับ (เอียง30องศา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 4.22 กราฟในแกน y ก่อนปรับ (เอียง30องศา)



รูปที่ 4.23 กราฟในแกน y หลังปรับ (เอียง30องศา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

รูปที่ 4.24 PID Tuning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการทดลองเพื่อทดสอบการทำงานของอุปกรณ์ลดการสั่นสะเทือนสำหรับการขนส่ง พบว่าอุปกรณ์สามารถทำงานได้ตามที่ต้องการทั้งในขณะที่ไม่มีภาระ และขณะที่มีภาระต่าง ๆ ที่มีน้ำหนักไม่เกิน 5 กิโลกรัม คืออุปกรณ์สามารถควบคุมฐานวางวัตถุให้อยู่ในสภาวะสมดุลได้ และไม่ทำให้วัตถุหล่นจากอุปกรณ์

จากการทดลองขอบเขตการทำงานของโมดูลเซ็นเซอร์ที่ตรวจจับการเคลื่อนไหวและความเอียงของวัตถุ (GY-87) มีเสถียรภาพในการส่งค่ามุมต่าง ๆ และสามารถกำหนดช่วงความกว้างของการอ่านค่ามุมได้

จากการทดลองเพื่อทำการควบคุมการหมุนของมอเตอร์ โดยการจ่ายพัลส์เพื่อควบคุมการหมุนของมอเตอร์(PWM) ได้ทำการทดลองหาขอบเขตของพัลส์มอเตอร์ พบว่าที่ 0 องศาจะได้พัลส์เท่ากับ 600 ms และ 180 องศาที่พัลส์เท่ากับ 2400 ms ทำให้ที่ 90 องศามีพัลส์เท่ากับ 1500 ms ที่ความถี่ 50 Hz

จากการทดลองปรับค่าการควบคุมแบบ PID Control หรือ Proportional Integral Derivative Control ทำการปรับค่า K_P , K_I , K_D จากการทดลองพบว่าค่าที่ทำให้อุปกรณ์เกิดเสถียรภาพที่สุด คือ สามารถเค้าสู่จุดสมดุลได้เร็วที่สุดและเกิดการแกว่งน้อยที่สุด พบว่าต้องใช้ค่า ในแกนX K_P = 0.54385 , K_I = 45.7933 , K_D = 0.015 ในแกนY K_P = 1.2156 , K_I = 44.6973 , K_D = 0.0089

5.2 วิจารณ์ผลการทดลอง

จากการทดลองพบปัญหาที่มีผลทำให้การวิจัยเกิดการผิดพลาดคือ

- การจัดวางอุปกรณ์ในตัวชิ้นงาน ในขณะที่อุปกรณ์ทำงานและเกิดการหมุนของมอเตอร์มาก ๆ ทำให้อุปกรณ์เกิดการคลายตัวของน็อตที่ใช้ยึด
- การหมุนของมอเตอร์ทำให้ตัวอุปกรณ์ไปโดนสายไฟ ทำให้สายไฟหลุด

5.3 ข้อเสนอแนะ

- ควรออกแบบการวางอุปกรณ์ให้เหมาะสม
- ใช้ความยาวของสายไฟให้เหมาะสม
- มีการเตรียมความพร้อมสำหรับการแก้ไขปัญหาที่อาจจะเกิดขึ้นได้ตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

เอกสารอ้างอิง

[1] SUPPORT THAIEASYELEC,. (2017). บทความ Arduino คืออะไร ตอนที่1 แนะนำเพื่อนใหม่ที่ชื่อ Arduino. Retrieved 07 October 2020, from https://blog.thaieasyelec.com/what-is-arduino-ch1/

[2] Arduino DUE compatible แบบ SMD. Retrieved 07 October 2020, from http://www.ioteshop.com/product/191/arduino-uno-r3-compatible-แบบ-smd

[3] Module MPU6050 GY-521. Retrieved 07 October 2020, from https://commandronestore.com/products/bo100.php

[4] Module MPU6050 GY-521. Retrieved 07 October 2020, from https://www.modulemore.com/product/66/เซนเซอร์วัดความเร่ง-ความเร็วเชิงมุม-gy-521-for-arduino-imu-3-axis-accelerometer-gyro-module-mpu6050

[5] Thanpong Phurak,. (2017). การควบคุมเซอร์โวมอเตอร์ด้วย ARDUINO. Retrieved 07 October 2020, from http://www.sbt.ac.th/new/sites/default/files/TNP_Unit_9.pdf

[6] automation360blog,. (2018). อธิบายผลของ PID อย่างง่ายๆด้วยกราฟ. Retrieved 07 October 2020, from https://automation360blog.wordpress.com/2018/01/01/pid-by-g

[7] MPU6050 Datasheet

[8] subcase.ru,. (2021). คำอธิบายเนื่องจาก Arduino เริ่มต้นกับ Arduino Due การป้องกันกระแสเกินUSB. Retrieved 24 March 2021, from https://subcase.ru/th/arduino-due-opisanie-nachalo-raboty-s-arduino-due-zashchita-usb-ot-peregruzki-po.html

[9] Arduino® IDE คืออะไร. Retrieved 24 March 2021, from https://www.netinbag.com/th/internet/what-is-an-arduinoreg-ide.html

[10] Anonymous,. (2013). Simulink คือ อะไร. Retrieved 24 March 2021, from http://whatsimulink.blogspot.com/

[11] SIMULINKเบื้องต้น. Retrieved 24 March 2021, from http://www.research-system.siam.edu/images/coop/STUDY_AND_DESIGN_OF_MICROCONTROLLER_CONTROL_SYSTEM_USING_SIMULINK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

คำสั่งที่ใช้ควบคุม

```
#include <Servo.h>

#include "Wire.h"

#include "I2Cdev.h" // การประกาศใช้ฟังก์ชันวัดความกดอากาศ

#include "MPU6050.h" // การประกาศใช้ฟังก์ชันวัดมุม

MPU6050 mpu;


 // code by Chanachon Srattasangsri 60010187

 //         Phmarin Purisapun 60011291


int16_t ax, ay, az;

int16_t gx, gy, gz;

Servo BON;

Servo LANG_LEFT;

Servo LANG_RIGHT;


double kPX = 0.54385, kIX = 45.7933 , kDX = 0.015 ;

double kPY = 1.2156, kIY = 44.6973 , kDY = 0.0089 ;

double PX = 0 , IX = 0 , DX = 0 , PY = 0 , IY = 0 , DY = 0 , PIDX , PIDY ;

double error_X = 0 , previous_error_X = 0 , previous_error_Y = 0, previous_I = 0 ,
error_Y = 0  ;

double setpoint_X = 0 , setpoint_Y = 0 , setpoint_Z = 168 ;

double actual_position_X = 0 , actual_position_Y = 0 , actual_position_Z = 0 ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

```
void setup()
{
  Wire.begin();
  Serial.begin(115200);
  //BON คือ Y
  //LANG คือ X
  BON.attach(7); // ประกาศเลือกขาสำหรับต่อสัญญาณการทำงาน
  LANG_LEFT.attach(3); // ประกาศเลือกขาสำหรับต่อสัญญาณการทำงาน
  LANG_RIGHT.attach(5); // ประกาศเลือกขาสำหรับต่อสัญญาณการทำงาน
  Serial.println("Initialize MPU6050");
  while (!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))
  {
    Serial.println("Could not find a valid MPU6050 sensor, check wiring!");
    delay(20);
  }

  BON.writeMicroseconds(1400);
  LANG_LEFT.writeMicroseconds(1400);
  LANG_RIGHT.writeMicroseconds(1400);

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

```
void loop()

{

  Vector normAccel = mpu.readNormalizeAccel();


  int ax = -(atan2(normAccel.XAxis, sqrt(normAccel.YAxis * normAccel.YAxis +
normAccel.ZAxis * normAccel.ZAxis)) * 180.0) / M_PI;

  int ay = (atan2(normAccel.YAxis, normAccel.ZAxis) * 180.0) / M_PI;


  //  Serial.print(" ax = ");

  //  Serial.print(ax);

  //  Serial.print(" ay = ");

  //  Serial.print(ay);

  //

  //  Serial.println();


  delay(180); // delay ในการอ่านค่าของเซ็นเซอร์ มีความช้ากว่า delay การทำงานของมอเตอร์
เนื่องจากต้องการให้มอเตอร์ทำงานก่อนจึงค่อยอ่านค่า error ตามภายหลัง

  ////////บน///////////////


  actual_position_X = ax ; // คือมุมองศาปัจจุบันที่เซ็นเซอร์แกน X สามารถอ่านได้


  error_X = setpoint_X - actual_position_X ; // error = ค่าสุดสมดุลที่ตั้งค่าไว้ในแกน X
(มอเตอร์ตัวบน) - มุมองศาปัจจุบันที่เซ็นเซอร์แกน X อ่านได้
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

PX = kPX * error_X; // ค่าคงที่ kPX ที่ตั้งค่าไว้ * errorที่เกิดขึ้นระหว่างค่าสุดสมดุลที่ตั้งค่าไว้ในแกน X กับ มุมองศาปัจจุบันที่เซ็นเซอร์แกน X

IX = IX + (kIX * error_X); // เพื่อลด offset ที่เกิดขึ้นจากค่า kPX ที่มากเกิน

DX = kDX * (error_X - previous_error_X); // คือการลดค่า error ที่เกิดขึ้นโดยการนำ error ปัจจุบัน - errorก่อนหน้า เพื่อลดการแกว่ง

PIDX = PX + IX + DX; // นำค่า P I D มาบวกกันเพื่อไปควบคุมมอเตอร์

if (PIDX < -350) //

{ //

PIDX = -350; // การทำงานของมอเตอร์เกิดจาก (1500+PID) 1500 ms คือมุม 90 องศา และองศามากที่สุดคือ 950ms และ 2050ms

} // ซึ่งอุปกรณีสามารถหมุนได้มากสุดที่จะไม่ทำให้อุปกรณีเสียหายอยู่ที่ไม่เกิน 57 องศา และ 123 องศา

if (PIDX > 350) // การกำหนดค่ามากสุดของ PID เพื่อไม่ให้เกิดการหมุนมากเกินองศา ที่อาจทำให้อุปกรณีเสียหาย

{ //

PIDX = 350; //

}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

/////////ล่าง//////////

actual_position_Y = ay ; // คือมุมองศาปัจจุบันที่เซ็นเซอร์แกน Y สามารถอ่านได้

error_Y = setpoint_Y - actual_position_Y ; // error = ค่าสุดสมดุลที่ตั้งค่าไว้ในแกน Y (มอเตอร์ตัวล่าง) - มุมองศาปัจจุบันที่เซ็นเซอร์แกน Y อ่านได้

PY = kPY * error_Y; // ค่าคงที่ kPY ที่ตั้งค่าไว้ * errorที่เกิดขึ้นระหว่างค่าสุดสมดุลที่ตั้งค่าไว้ในแกน Y กับ มุมองศาปัจจุบันที่เซ็นเซอร์แกน Y

IY = IY + (kIY * error_Y); // เพื่อลด offset ที่เกิดขึ้นจากค่า kPY ที่มากเกิน

DY = kDY * (error_Y - previous_error_Y); // คือการลดค่า error ที่เกิดขึ้นโดยการนำ error ปัจจุบัน - errorก่อนหน้า เพื่อลดการแกว่ง

PIDY = PY + IY + DY; //

if (PIDY < -350) //

{ //

PIDY = -350; // การทำงานของมอเตอร์เกิดจาก (1500+PID) 1500 ms คือมุม 90 องศา และองศามากที่สุดคือ 950ms และ 2050ms

} // ซึ่งอุปกรณีสามารถหมุนได้มากสุดที่จะไม่ทำให้อุปกรณีเสียหายอยู่ที่ไม่เกิน 57 องศา และ 123 องศา

if (PIDY > 350) // การกำหนดค่ามากสุดของ PID เพื่อไม่ให้เกิดการหมุนมากเกินองศา ที่อาจทำให้อุปกรณีเสียหาย

{ //

PIDY = 350; //

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

```
  }

  BON.writeMicroseconds(1450 - PIDY);  // การควบคุมมอเตอร์ตัวบน ค่าสมดุลมอเตอร์ 1500 msที่90องศา + PID ที่คำนวณได้

  LANG_LEFT.writeMicroseconds(1450 - PIDX); // การควบคุมมอเตอร์ตัวบน ค่าสมดุลมอเตอร์ 1500 msที่90องศา - PID ที่คำนวณได้ (ที่ต้องลบค่าPIDเนื่องจากวางเซ็นเซอร์ในทางตรงกันข้าม)

  LANG_RIGHT.writeMicroseconds(1380 + PIDX);

  previous_error_X = error_X ; // กำหนดให้ค่าerrorก่อนหน้า ให้เท่ากับ error ปัจจุบัน เพื่อ Loop กลับไปคำนวณ FEED BACK ในแกน X

  previous_error_Y = error_Y ; // กำหนดให้ค่าerrorก่อนหน้า ให้เท่ากับ error ปัจจุบัน เพื่อ Loop กลับไปคำนวณ FEED BACK ในแกน Y


//    Serial.print("error_X = ");

//    Serial.print(error_X);

//    Serial.print("error_Y = ");

//    Serial.print(error_Y);

  //    Serial.print("previous_error_X = ");

  //    Serial.print(previous_error_X);

  //  Serial.print("previous_error = ");

  //  Serial.print(previous_error);

  //  Serial.print("P = ");

  //  Serial.print(P);

  //  Serial.print("I = ");

  //  Serial.print(I);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

```
  //  Serial.print("D = ");

  //  Serial.print(D);

  //  Serial.print("previous_I = ");

  //  Serial.print(previous_I);

  Serial.print("PIDX = ");

  Serial.print(PIDX);

  Serial.print("PIDY = ");

  Serial.println(PIDY);

  //  Serial.print(setpoint_Y);

  //  Serial.print("actual_position_Y = ");

  //  Serial.print(actual_position_Y);

  delay (20);


}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.