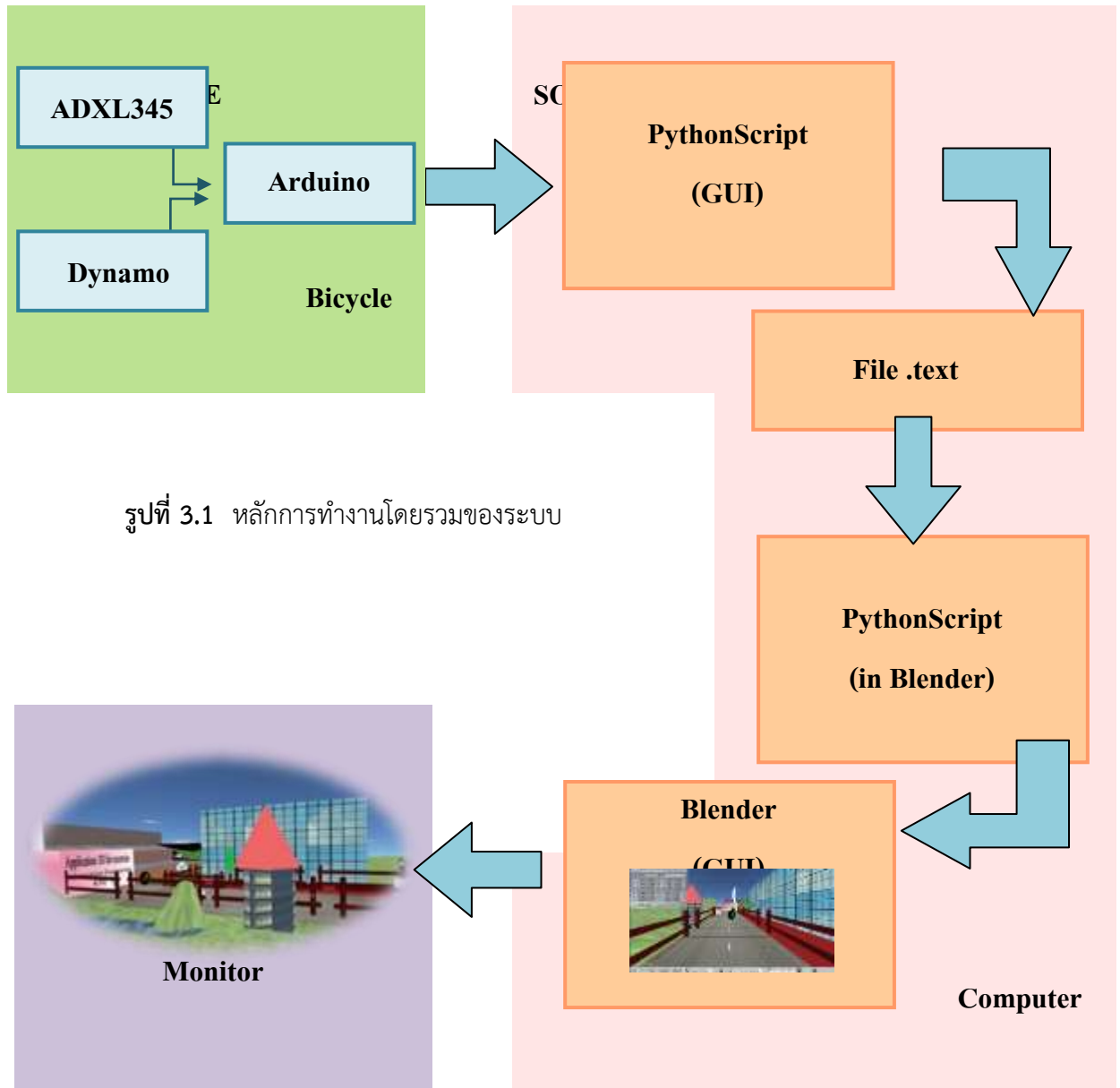


บทที่ 3

การออกแบบและการสร้าง

3.1 หลักการทำงานของระบบโดยรวม



รูปที่ 3.1 หลักการทำงานของระบบโดยรวม

จากรูปที่ 3.1 การทำงานของระบบโดยรวมจะแบ่งเป็น 2 ส่วนใหญ่ๆ คือ ส่วนของชุดอุปกรณ์และส่วนของโปรแกรมกราฟิกสามมิติ

โดยการทำงานคือ ชุดโมดูลบอร์ดไมโครคอนโทรลเลอร์ Arduino จะทำหน้าที่ควบคุมการทำงานของอุปกรณ์ ADXL345(Accelerometer) เซนเซอร์วัดความเร่งแนว 3 แกน (แกน X, Y, Z) ที่ใช้สำหรับตรวจจับการเคลื่อนไหวศีรษะของผู้ออกกำลังกายเพื่อให้สามารถบังคับซ้ายขวาตามความ

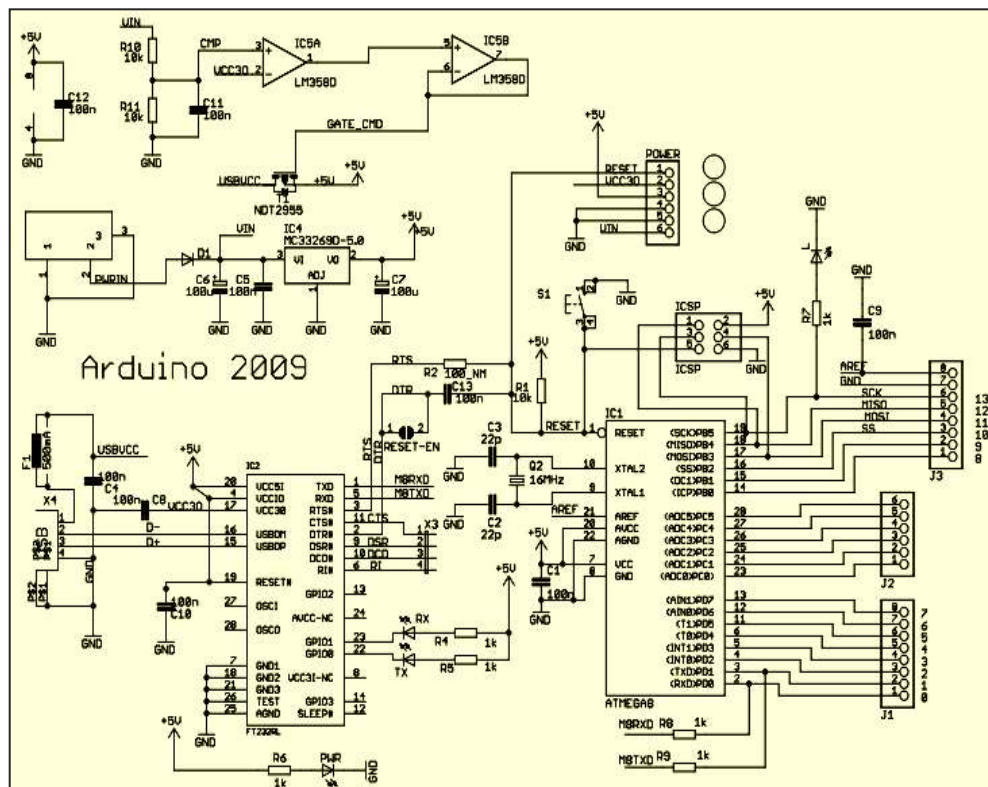
ต้องการของผู้เล่นได้ และสำหรับการเคลื่อนที่ภายในเกมจะเคลื่อนที่ตามการปั่นจักรยานของผู้ออกกำลังกาย โดยมีไดนามิจักรยานทำหน้าที่ส่งค่าอนาล็อกให้กับชุดโมดูลบอร์ดไมโครคอนโทรลเลอร์ Arduino เมื่อชุดโมดูลบอร์ดดังกล่าวได้รับค่าข้อมูลจึงจะส่งค่าไป(โดยซอฟต์แวร์ที่ใช้เป็นภาษา PythonScript และค่าที่ได้รับมาจะมี 4 ค่า นั่นคือ X, Y, Z และ Dynamo) เก็บไว้ในรูปแบบไฟล์ .text เพื่อให้สามารถนำค่าข้อมูลดังกล่าวไปประมวลผลในส่วนของโปรแกรมกราฟิกสามมิติที่ถูกออกแบบขึ้น โดยใช้โปรแกรม Blender3D และควบคุมการใช้งานต่างๆของเกมด้วยโปรแกรมภาษา PythonScript และส่งค่าให้กับ Monitor แสดงผลการทำงานของโปรแกรมกราฟิกสามมิติต่อไป

3.2 ส่วนของชุดอุปกรณ์

อุปกรณ์จะต้องอาศัยการติดต่อกันระหว่าง ชุดโมดูลบอร์ดไมโครคอนโทรลเลอร์ Arduino, ADXL345(Accelerometer) และไดนามิจักรยาน โดยหลักการต่างๆมีดังต่อไปนี้

3.2.1 ชุดโมดูลบอร์ดไมโครคอนโทรลเลอร์ Arduino

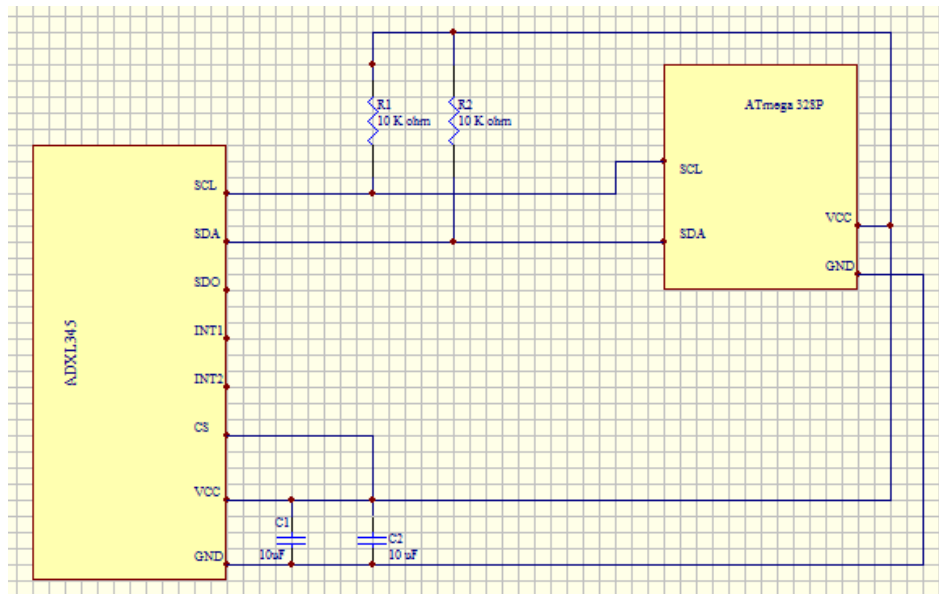
ชุดโมดูลบอร์ดไมโครคอนโทรลเลอร์ Arduino เป็นบอร์ดทดลองสำเร็จรูปโดยใช้ MCU ในตระกูล AVR เบอร์ ATmega 328 สามารถส่งข้อมูลได้รวดเร็วและมีช่องสำหรับต่อกับอุปกรณ์ภายนอกได้อย่างสะดวก อีกทั้งสามารถใช้ไฟเลี้ยงจากการต่อพอร์ทUSB ได้อีกด้วย



รูปที่ 3.2 วงจรภายในชุดโมดูลบอร์ดไมโครคอนโทรลเลอร์ Arduino Duemilanove [1]

3.2.2 ADXL345 (Accelerometer)

การติดต่อเซนเซอร์ ADXL345 กับบอร์ด Arduino Duemilanove ใช้หลักการส่งข้อมูลแบบ I2C Bus ดังนั้นการออกแบบวงจรของเซนเซอร์จึงใช้เพียงสาย SDA กับ SCL ในการส่งข้อมูลระหว่างบอร์ด Arduino Duemilanove โดยใช้ไฟเลี้ยง 3.3 v จากบอร์ดได้ด้วยเช่นกัน

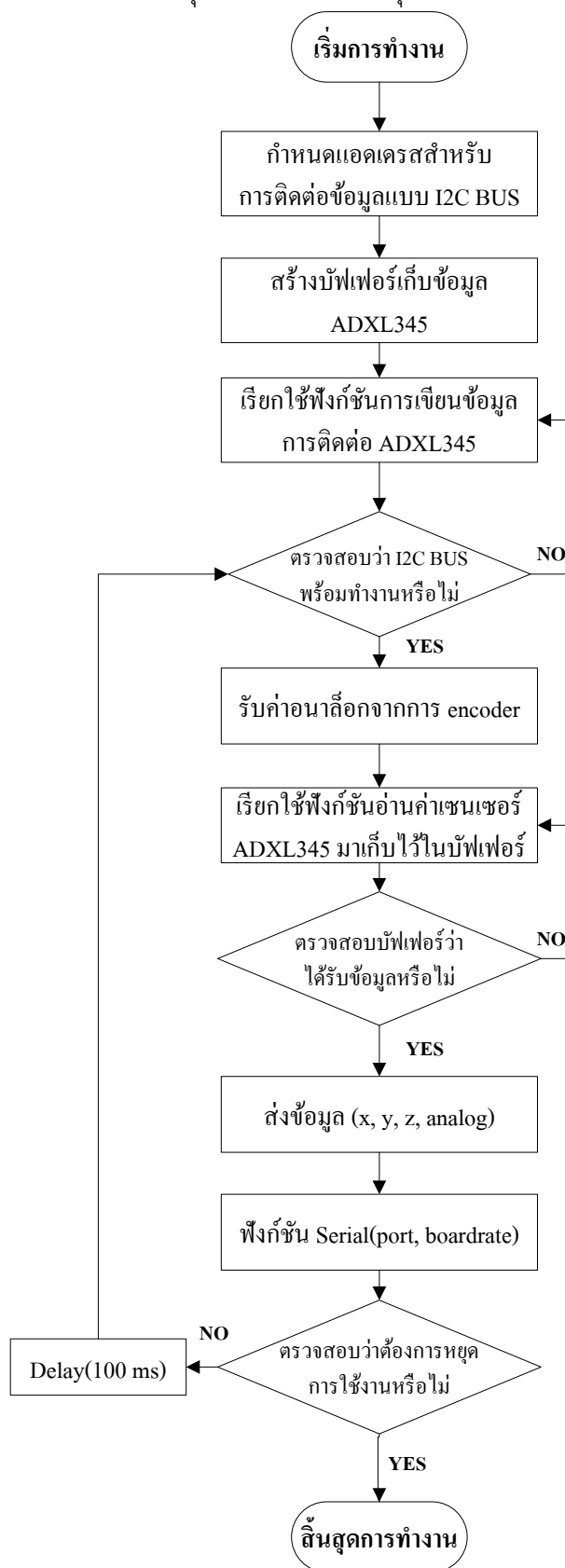


รูปที่ 3.3 การออกแบบการเชื่อมต่อข้อมูลแบบ I2C Bus กับบอร์ดทดลอง Arduino

3.2.3 หลักการทำงานของ Arduino ในการส่งข้อมูล

1. กำหนดแอดเดรสสำหรับการติดต่อข้อมูลของ Arduino โดยอาศัยการส่งแบบ I2C Bus ให้กำหนดพอร์ตนาล็อกสำหรับรับค่าอินพุตจากไดนาโมจิกรยาน
2. สร้างบัฟเฟอร์ขึ้นเพื่อสำหรับเก็บข้อมูลที่รับจาก ADXL345(Accelerometer) โดยข้อมูลดังกล่าวจะถูกนำไปเก็บเพื่อนำไปใช้งาน
3. เรียกใช้ฟังก์ชันการเขียนข้อมูลเพื่อให้อุปกรณ์โมดูลบอร์ด Arduino ติดต่อกับ ADXL345
4. ตรวจสอบว่าการส่งข้อมูลในรูปแบบ I2C BUS พร้อมทำงานหรือไม่
 - ถ้าการส่งข้อมูลในรูปแบบ I2C BUS ไม่พร้อมทำงานจะทำการเรียกใช้ฟังก์ชันการเขียนข้อมูลติดต่อ ADXL345 (ในข้อ 3) อีกครั้ง
 - ถ้าการส่งข้อมูลในรูปแบบ I2C BUS พร้อมทำงานจะรับค่าอนาล็อกจากไดนาโมจิกรยาน เพื่อส่งข้อมูลดังกล่าวต่อไป
5. ทำการรับข้อมูลชนิดอนาล็อกจากไดนาโมจิกรยาน
6. เรียกใช้ฟังก์ชันการอ่านค่าข้อมูลจากเซนเซอร์ ADXL345 (ในหัวข้อ 3.2.4) และนำค่าข้อมูลดังกล่าวมาเก็บไว้ในบัฟเฟอร์
7. ตรวจสอบบัฟเฟอร์ว่าได้รับข้อมูลเรียบร้อยแล้วหรือไม่
 - ถ้าตรวจสอบบัฟเฟอร์พบว่ายังไม่มีกรับข้อมูล จะย้อนกลับไปเพื่อเรียกใช้ฟังก์ชันการอ่านค่าข้อมูลจากเซนเซอร์ ADXL345 มาเก็บในบัฟเฟอร์ (ในข้อ 6) อีกครั้ง
 - ถ้าตรวจสอบบัฟเฟอร์พบว่ามีกรส่งข้อมูลเกิดขึ้น จะส่งข้อมูล (x, y, z, Analog) และฟังก์ชัน Serial (Port, Board rate) ออกไปใช้งาน
8. ตรวจสอบว่าต้องการหยุดการใช้งานหรือไม่
 - ถ้าตรวจสอบว่าไม่ต้องการหยุดการใช้งาน จะเพิ่มค่า delay เป็นขนาด 100 ms

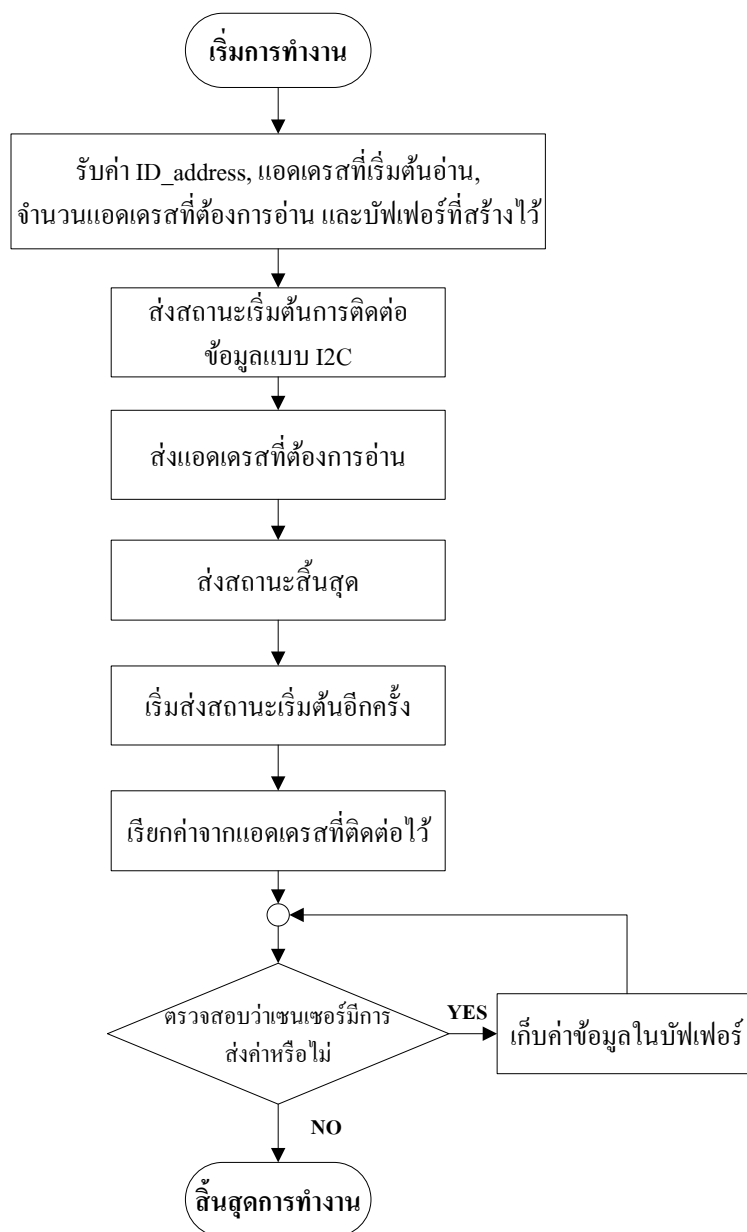
- ถ้าตรวจสอบว่าต้องการหยุดการใช้งาน จะสิ้นสุดการทำงาน



รูปที่ 3.4 Flow chat หลักการทำงานของ Arduino ในการส่งข้อมูล

3.2.4 หลักการทำงานของ Arduino ในการอ่านค่าข้อมูลจากเซนเซอร์ ADXL345

1. เริ่มต้นการทำงาน โดยรับค่า ID_Address, แอดเดรสที่เริ่มต้นอ่าน, จำนวนแอดเดรสที่ต้องการอ่าน และบัฟเฟอร์ที่สร้างไว้เพื่อเข้ามารองรับแอดเดรสที่อ่านได้
2. ส่งสถานะเริ่มต้นการติดต่อข้อมูลแบบ I2C
3. ส่งแอดเดรสที่ต้องการอ่าน
4. ส่งสถานะสิ้นสุด
5. เริ่มส่งสถานะเริ่มต้นใหม่อีกครั้ง
6. เรียกค่าจากแอดเดรสที่ติดต่อไว้
7. ตรวจสอบว่าเซนเซอร์ส่งค่าข้อมูลหรือไม่
 - ถ้าตรวจสอบเซนเซอร์พบว่ามีค่าส่งข้อมูล จะทำการส่งค่าไปเก็บในบัฟเฟอร์ที่สร้างไว้ และกลับไปตรวจสอบเซนเซอร์อีกครั้ง
 - ถ้าตรวจสอบเซนเซอร์พบว่าไม่มีการส่งค่าข้อมูล จะสิ้นสุดการทำงาน



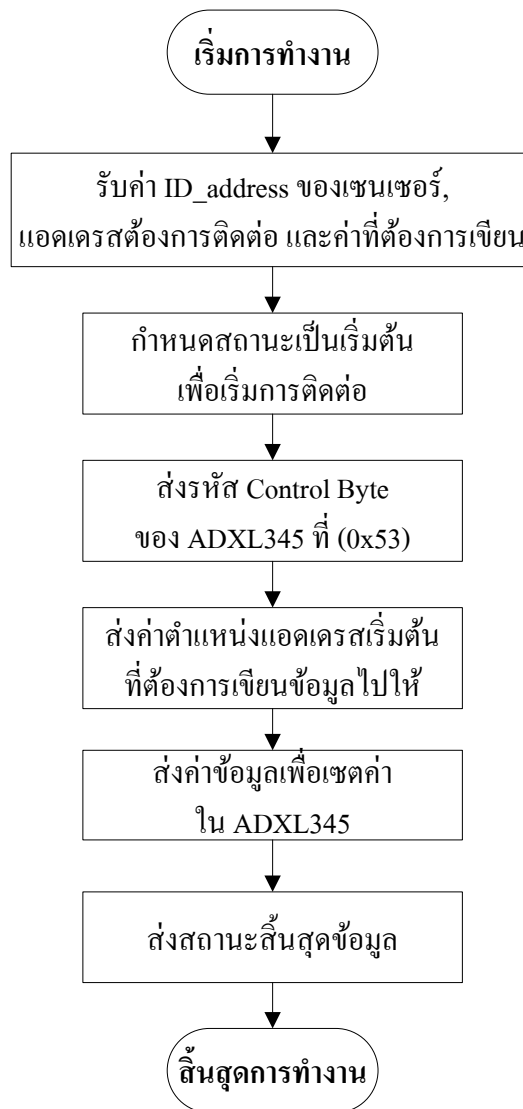
รูปที่ 3.5 Flow chat หลักการอ่านค่าข้อมูลของเซนเซอร์ ADXL345

3.2.5 หลักการทำงานของ Arduino ในการเขียนค่าข้อมูลไปยังเซนเซอร์ ADXL345

1. เริ่มต้นการทำงานโดยรับค่า ID_address ของเซนเซอร์, แอดเดรสที่ต้องการติดต่อ และค่าที่

ต้องการเขียน

2. กำหนดสถานะเป็นเริ่มต้น เพื่อเริ่มการติดต่อสื่อสารระหว่าง Arduino และ ADXL345
3. ส่งรหัส Control Byte ของ ADXL345 ไปที่ (0x53)
4. ส่งค่าตำแหน่งแอดเดรสที่ต้องการเขียนข้อมูลติดต่อภายใน slave
5. ส่งค่าข้อมูลเพื่อปรับค่าใน ADXL345
6. ส่งสถานะสิ้นสุดการทำงาน

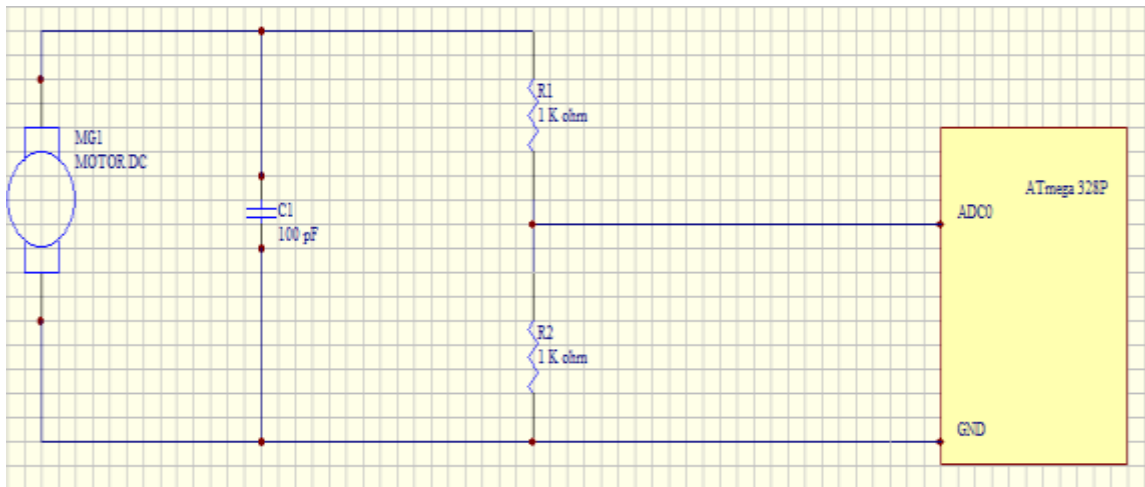


รูปที่ 3.6 Flow chat หลักการเขียนค่าข้อมูลของเซนเซอร์ ADXL345

3.2.6 หลักการทำงานของ Arduino ในการอ่านค่าข้อมูลที่ได้รับจากไดนาโมจักรยาน

ในส่วนนี้เป็นการออกแบบการติดต่อระหว่าง Arduino และไดนาโม หลักการคือ ใช้ตัวเก็บประจุแบบไม่มีขั้วต่อคร่อมกันระหว่างไดนาโมจักรยาน เพื่อป้องกันแรงดันที่อาจมีค่าสูง และใช้ตัวต้านทานสองตัวซึ่งมีค่าตัวละ 1K ต่อแบบอนุกรม เพื่อช่วยในการลดค่าแรงดันจากไดนาโมจักรยานให้เหลือเพียง 5 โวลต์ เหตุผลที่สร้างวงจรดังกล่าว เนื่องจากหากปั่นจักรยานจะเกิดการหมุนของไดนาโมจักรยานเกิดขึ้น เมื่อความเร็วในการปั่นสูงขึ้นจะเกิดแรงดันที่ทำให้ส่งเข้าไปในบอร์ด

ไมโครคอนโทรลเลอร์ Arduino สูงเกินกว่าบอร์ดไมโครคอนโทรลเลอร์จะรับได้ และส่งผลต่ออาจทำให้บอร์ดดังกล่าวเสียหาย



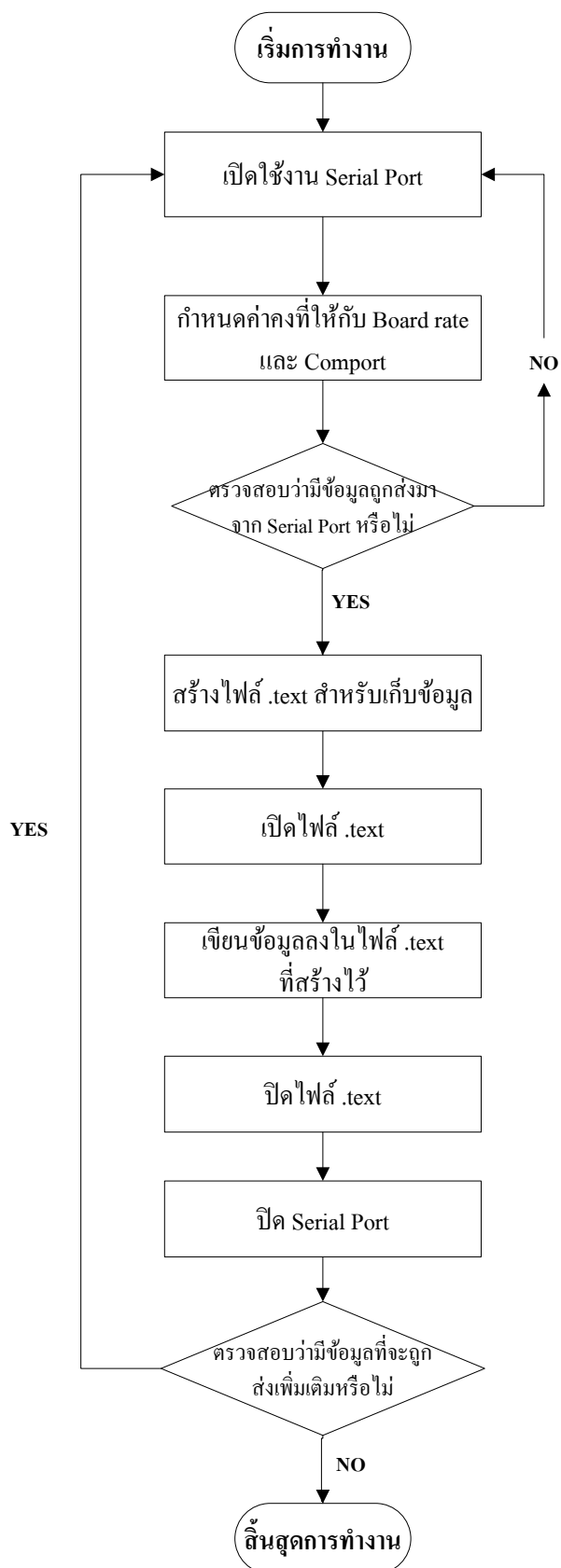
รูปที่ 3.7 วงจรไดนาโมจักรยานต่อเข้าสู่บอร์ด Arduino

3.3 ส่วนของโปรแกรมกราฟิกสามมิติ

3.3.1 ส่วนของการรับข้อมูลของโปรแกรมภาษา Python จากไมโครคอนโทรลเลอร์บอร์ด

Arduino โดยส่งผ่านทาง Serial Port

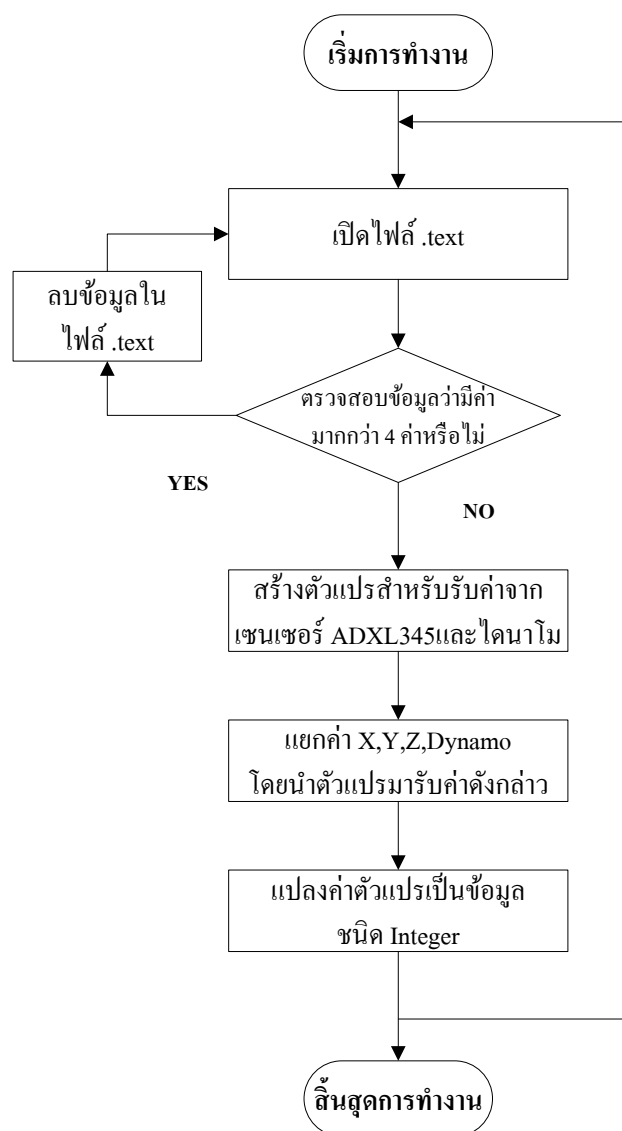
1. เริ่มต้นการทำงานโดยเปิดใช้งาน Serial Port
2. กำหนดค่าคงที่ให้กับ Board rate และ Comport สำหรับรับค่าจาก Serial Port
3. ตรวจสอบว่ามีข้อมูลถูกส่งมาจาก Serial Port หรือไม่
 - ถ้าตรวจสอบพบว่าส่งข้อมูลไม่ถูกส่งมา จะทำการส่งค่าว่างเปล่า(ค่า null)กลับไปเพื่อรอรับข้อมูลใหม่ที่จะเข้ามา (ในข้อ 1) อีกครั้ง
 - ถ้าตรวจสอบพบว่าการส่งข้อมูลเข้ามา จะนำค่าข้อมูลดังกล่าวเก็บไว้ในไฟล์ .text
4. สร้างไฟล์ .text เพื่อเก็บค่าข้อมูลที่ถูกส่งมาจาก Serial Port
5. เปิดไฟล์ .text
6. เขียนข้อมูลลงในไฟล์ .text ที่สร้างไว้
7. ปิดไฟล์ .text
8. ปิด Serial Port
9. ตรวจสอบว่าต้องการรับค่าจาก Serial Port เพิ่มเติมหรือไม่
 - ถ้าตรวจสอบพบว่าต้องการรับค่าข้อมูลเพิ่มเติมจะทำการวนลูป(ไปที่ข้อ 1) เพื่อทำการรับค่าข้อมูลใหม่
 - ถ้าตรวจสอบพบว่าไม่ต้องการรับค่าข้อมูลเพิ่มเติม จะสิ้นสุดการทำงาน



รูปที่ 3.8 Flow chat ส่วนของการรับข้อมูลโปรแกรมภาษา Python รับค่าข้อมูลจาก ไมโครคอนโทรลเลอร์บอร์ด Arduino โดยส่งผ่านทาง Serial Port

3.3.2 ส่วนของการแยกค่าข้อมูลที่ส่งมาไว้ในไฟล์ .text

1. เปิดไฟล์ .text ที่ได้จัดเก็บค่าไว้
2. ตรวจสอบข้อมูลที่เข้ามาว่ามีค่ามากกว่า 4 ค่าต่อการส่งข้อมูล 1 ครั้งหรือไม่
 - ถ้าตรวจสอบพบว่าค่าที่ส่งมามีค่ามากกว่า 4 ค่า จะทำการลบข้อมูลในไฟล์ .text และรอค่าข้อมูลใหม่
 - ถ้าตรวจสอบพบว่าค่าที่ส่งมามีค่าเท่ากับ 4 ค่า จะทำการกำหนดตัวแปรมารับค่าดังกล่าวเป็นค่า dX, dY, dZ, Dynamo
3. สร้างตัวแปรสำหรับรับค่าข้อมูลจากเซนเซอร์และไดนามิกจรรยา
4. แยกค่า X, Y, Z, Dynamo ออกจากกันและนำตัวแปรที่สร้างขึ้นมารับค่าข้อมูล
5. แปลงค่าข้อมูลดังกล่าวเป็น Integer

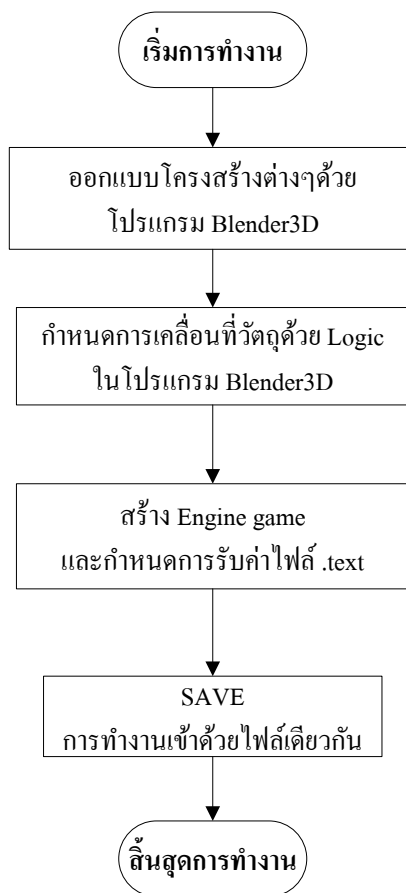


รูปที่ 3.9 Flow chat ส่วนของการแยกค่าข้อมูลที่ส่งมาไว้ในไฟล์ .text

3.3.3 ส่วนของการออกแบบรูปทรงสามมิติ

1. ออกแบบโครงสร้างต่างๆด้วยโปรแกรม Blender3D
2. กำหนดการเคลื่อนที่เบื้องต้นด้วย Logic ในโปรแกรม Blender3D

3. สร้าง Engine game และกำหนดให้มีการรับค่าจากไฟล์ .text ด้วยโปรแกรมภาษา Python
4. Save โปรแกรม Blender3D และโปรแกรมภาษา Python ลงในไฟล์เดียวกัน



รูปที่ 3.10 Flow chat ส่วนของการออกแบบรูปทรงสามมิติ