ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

โบว์ลิ่งเสมือนจริง

VIRTUAL STRIKE BOWL

โดย

ธีรวัฒน์ ลือวงษ์

ภูเบศ กร้ำมาตร

อาจารย์ที่ปรึกษา

รศ.ดร. นพพร โชติกกำธร

อาจารย์ที่ปรึกษาร่วม

อาจารย์ ณฐพล พันธุวงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2551

เลขหมู่.....
เลขทะเบียน 06042
วัน,เดือน,ปี 10 มิ.ย. 2553
b 12176067
i.....

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VIRTURL STRIKE BOWL

THEERAWAT LUEWONG

PUBASE KRAMMART

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/2008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2009

FACULTY ON INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญานิพนธ์ ประจำปีการศึกษา 2551

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โบว์ลิ่งเสมือนจริง

VIRTUAL STRIKE BOWL

ผู้จัดทำ

1. นายธีรวัฒน์ ลือวงษ์ รหัสนักศึกษา 48070127
2. นายภูเบศ กร้ำมาตร รหัสนักศึกษา 48070146

.....อาจารย์ที่ปรึกษา

(รศ.ดร.นพพร โชติกกำธร)

.....อาจารย์ที่ปรึกษาร่วม

(อาจารย์ณฐพล พันธุวงศ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน	โบว์ลิ่งเสมือนจริง
นักศึกษา	ธีรวัฒน์ ลือวงษ์
	ภูเบศ กร้ำมาตร
รหัสนักศึกษา	48070127
	48070146
ปริญญา	วิทยาศาสตรบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
ปีการศึกษา	2551
อาจารย์ที่ปรึกษา	รศ.ดร. นพพร โชติกกำธร
อาจารย์ที่ปรึกษาร่วม	อาจารย์ณฐพล พันธุวงศ์

บทคัดย่อ

กีฬาโบว์ลิ่งเป็นกีฬาที่กำลังได้รับความนิยมเป็นอย่างมากทั้งในเด็กและผู้ใหญ่ แต่เนื่องด้วยกีฬาโบว์ลิ่งเป็นกีฬาที่ต้องใช้พื้นที่มากในการเล่น และเลนที่ใช้จริงมีขนาดใหญ่ ดังนั้นผู้จัดทำจึงคิดสร้างโบว์ลิ่งเสมือนจริงขึ้นมาให้มีขนาดเล็กลง และสามารถเคลื่อนย้ายได้ แต่ยังสามารถโยนได้เหมือนเล่นโบว์ลิ่งจริง โดยได้สร้างเลนโบว์ลิ่งขึ้นมาให้มีความยาวประมาณ 6 เมตร เพื่อให้ผู้ใช้ได้โยนลูกโบว์ลิ่งจริง รอบเลนโบว์ลิ่งที่สร้างขึ้นจะมีการติดตัวเซนเซอร์ และเลเซอร์ไว้ เพื่อรับค่าส่งให้ไมโครคอนโทรเลอร์ เมื่อลูกโบว์ลิ่งที่ผู้เล่นโยนเคลื่อนที่มาถึงจุดปลายเลนก็จะเป็นการส่งค่าต่าง ๆ ให้กับโปรแกรมเบลนเดอร์ จากนั้นจะเป็นการจำลองส่วนความยาวที่เหลือของเลน พิน และลูกโบว์ลิ่งที่โยน ด้วยโปรแกรมเบลนเดอร์ โปรแกรมเบลนเดอร์จะรับค่าจากไมโครคอนโทรเลอร์มาเพื่อนำไปคำนวณหาทิศทางการเคลื่อนที่ต่อ คำนวณการล้มของพิน และคะแนนที่เกิดขึ้น พร้อมนำเสนอผ่านหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Title	Virtual Strike Bowl
Student	Mr.Theerawat Luewong
	Mr. Pubase Krammart
Student ID.	48070127
	48070146
Degree	Bachelor of Science
Programme	Information Technology
Year	2008
Advisor	Assoc.Prof. Dr.Nopporn Chotikakamthorn
Co-Advisor	Mr. Natapon Pantuwong

ABSTRACT

Bowling is one of the most popular games among both children and adults but it required plenty of spaces to install the system especially the huge bowling lanes. Given that problem, we have come with the virtual real bowling game which is smaller and removable function added. Virtual strike bowl is the imitation of the virtual reality of bowling game by developing 6 meters bowling lane for the users to have really throw in the bowling ball. Around the developed lane, there are censorships and lasers in order to receive the data and pass them to microcontroller. When the thrown ball has been move to the end-point of the lane, all the data would be sent to program blender. After that would be the imitation of the remaining length of bowling lane, bowling pins and bowling ball using program blender. Program blender would receive the data from microcontroller and compute them the result of moving direction, the tumble of the pins and scores the users are supposed to have and show them through the monitors.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการนี้ ได้ดำเนินการภายใต้ความช่วยเหลือและความอนุเคราะห์จากบุคคลต่าง ๆ ดังนี้

ขอขอบพระคุณ บิดามารดา ผู้ให้กำเนิด อบรม เลี้ยงดู สั่งสอน และให้โอกาสทางการศึกษา ทั้งยังคอยเป็นที่ปรึกษาและกำลังใจที่ดีเสมอมา

ขอขอบพระคุณ อาจารย์ที่ปรึกษา รศ.ดร. นพพร โชติกกำธร และอาจารย์ที่ปรึกษาร่วม อาจารย์ณฐพล พันธุวงศ์ ได้กรุณาให้ทั้งความช่วยเหลือ คำแนะนำ และคำชี้แนะข้อผิดพลาดต่าง ๆ เป็นอย่างดีเสมอมาตลอดการดำเนินงาน

ขอขอบพระคุณ คณาจารย์ทุกท่านที่มิได้เอ่ยนาม ณ ที่นี้ ที่ได้ประสิทธิ์ประสาทวิชาความรู้ อันพึงเป็นประโยชน์ในการดำเนินงาน ทั้งในสมัยอนุบาล จนถึง อุดมศึกษา

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ทุกคนที่คอยให้ทั้งกำลังใจ ความรู้ คำแนะนำและ ข้อเสนอแนะ ที่ดีเสมอมา

ธีรวัฒน์ ลือวงษ์

ภูเบศ กร้ำมาตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ.....I

กิตติกรรมประกาศ.....II

สารบัญ..... III

สารบัญตาราง..... V

สารบัญรูป..... VI

บทที่ 1 บทนำ

1.1 ความเป็นมา.....1

1.2 วัตถุประสงค์.....1

1.3 ขอบเขตของโครงงาน.....2

1.4 ขั้นตอนการดำเนินงาน.....2

1.5 ประโยชน์ที่ได้รับ.....2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 โปรแกรมเบลนเดอร์.....3

2.1.1 จุดเด่นของโปรแกรมเบลนเดอร์.....3

2.2 การนำโปรแกรมเบลนเดอร์มาใช้ในการสร้างโมเดล.....4

2.2.1 การสร้างโมเดล.....4

2.2.2 สร้างส่วนของเกมส์ โดยใช้ Game Engine ของโปรแกรมเบลนเดอร์.... 9

2.3 ทฤษฎีการทำงานของเซนเซอร์.....11

2.3.1 คุณสมบัติทั่วไปของเซนเซอร์.....11

2.3.2 ประเภทของเซนเซอร์.....12

2.3.3 ลักษณะการทำงานของตัวต้านทานปรับค่าได้..... 12

2.3.4 ลักษณะการทำงานของเซ็นเซอร์แสงและอินฟราเรด.....13

2.3.5 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล..... 14

2.3.6 ข้อบ่งเฉพาะของการแปลงสัญญาณ A/D (A/D SPECIFICATIONS).... 15

2.4 อุปกรณ์ที่ใช้ในการสร้างโบว์ลิ่งเสมือนจริง.....15

2.4.1 ไมโครคอนโทรลเลอร์ (CP-JR ARM7 USB-LPC2148)..... 15

2.4.2 คุณสมบัติของบอร์ด.....17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.4.3 โครงสร้างบอร์ด CP-JR ARM7 USB-LPC2148 / EXP 19
2.4.4 การเชื่อมต่อกับ Character LCD.....22
2.4.5 ขั้วต่อ JTAG ARM 23
2.4.6 การ Download Hex file ให้กับ MCU ของบอร์ด.....25
2.5 เลเซอร์ (Laser).....28
2.6 LDR ขนาด 10 มิลลิเมตร.....30
2.7 VR ปรับค่าแบบเกือกม้า.....31

บทที่ 3 การวิเคราะห์และออกแบบระบบ
3.1 ภาพรวมของระบบ.....33
3.2 ผังงาน (Flowchart).....34
3.2.1 ผู้เล่นโยนลูกโบว์ลิ่งผ่าน Sensor.....34
3.2.2 ไมโครคอนโทรลเลอร์มาคำนวณหาพิกัดจุด.....34
3.2.3 ส่งพิกัดจุดที่ได้ให้โปรแกรม.....35
3.2.4 นำค่าพิกัดจุดที่ได้มาสร้างจำลองการเคลื่อนที่.....36
3.2.5 ตรวจเช็คว่าพินล้มกี่พิน.....36
3.2.6 นำจำนวนพินที่ล้มไปแสดงผล.....37
3.3 โครงสร้างและการออกแบบเลนส์.....37
3.4 การรับค่า และตรวจจับการเคลื่อนที่ของลูกโบว์ลิ่ง.....39

บทที่ 4 การพัฒนาระบบ
4.1 การพัฒนาโปรแกรมโดยใช้ไมโครคอนโทรลเลอร์.....41
4.1.1 ไมโครคอนโทรลเลอร์ (ARM 7) 41
4.2 การพัฒนาโปรแกรมโดยใช้เบลนเดอร์ (Blender).....47
4.2.1 การสร้างโมเดล.....47
4.2.2 การรับค่าจากไมโครคอนโทรลเลอร์เพื่อใช้งานSerial Port.....49
4.2.3 การกำหนดการรับค่าอินพุทและการแสดงเอ้าท์พุทของเกมส์.....53
4.3 ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ.....55
4.3.1 ฮาร์ดแวร์.....55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

4.3.2 ซอฟต์แวร์..... 58
4.4 อธิบายการติดตั้งและการทำงานของโปรแกรม..... 59

บทที่ 5 สรุปการพัฒนาโครงงาน

5.1 สรุปโครงงาน.....63
5.2 ประโยชน์ที่ได้รับจากการพัฒนาโครงงาน.....63
5.3 ปัญหาที่พบในการพัฒนาโครงงานและแนวทางแก้ไข..... 63
5.4 แนวทางการพัฒนาในอนาคต..... 65

บรรณานุกรม.....66
ภาคผนวก.....67
ประวัติผู้แต่ง.....75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 รูปแสดงการขยายก้นผิวของพิน..... 4
รูปที่ 2.2 รูปแสดงส่วนของพินในมุมฟรอนวิว.....5
รูปที่ 2.3 รูปแสดงส่วนของพินในขั้นตอนการตกแต่ง.....6
รูปที่ 2.4 รูปแสดงส่วนของลูกโบว์ลิ่ง.....7
รูปที่ 2.5 รูปแสดงส่วนเลนโบว์ลิ่ง.....8
รูปที่ 2.6 รูปแสดงส่วนของท่อเลนโบว์ลิ่ง.....9
รูปที่ 2.7 รูปแสดงส่วนของการกำหนดวัตถุที่เราสร้างมีมวล.....10
รูปที่ 2.8 ลักษณะโครงสร้างของบอร์ด CP-JR ARM7 USB-LPC2148 / EXP..... 19
รูปที่ 2.9 ลักษณะของบอร์ด CP-JR ARM7 USB-LPC2148 EXP..... 20
รูปที่ 2.10 แสดงการเชื่อมต่อ Character LCD.....22
รูปที่ 2.11 ขั้วต่อ Port I/O ต่างๆของบอร์ด.....22
รูปที่ 2.12 ขั้วต่อแบบ IDE 10 Pin จำนวน 5 ชุด ชุดละ 8 บิต 23
รูปที่ 2.13 ขั้วต่อ RS232.....24
รูปที่ 2.14 ขั้วต่อ Power Input (+5V) 24
รูปที่ 2.15 ภาพแสดงการจัดขั้ว.....25
รูปที่ 2.16 ภาพแสดงการสั่ง Run โปรแกรม.....25
รูปที่ 2.17 ภาพแสดงข้อความเตือน.....26
รูปที่ 2.18 ภาพแสดงรายละเอียด Part ID และ Boot Loader ID..... 27
รูปที่ 2.19 แสดง Status Progress.....27
รูปที่ 2.20 แสดง Status “File Upload Successfully Completed”.....28
รูปที่ 2.21 ภาพแสดงแถบสีต่าง ๆ.....28
รูปที่ 2.22 ภาพแสดงการเปล่งแสงจากหลอดไฟ.....29
รูปที่ 2.23 ภาพแสดงทิศทางแสงที่แน่นอน.....29
รูปที่ 2.24 โครงสร้างของ LDR..... 30
รูปที่ 2.25 ตัวอย่างวงจรเปลี่ยนสัญญาณแสงเป็นสัญญาณ.....30
รูปที่ 2.26 ภาพแสดงตัวต้านทานปรับค่าได้..... 31
รูปที่ 3.1 ภาพโดยรวมของระบบ..... 33
รูปที่ 3.2 แสดงขั้นตอนการโยนลูกโบว์ลิ่ง.....34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 3.3 แสดงขั้นตอนการเคลื่อนที่ของลูกโบว์ลิ่งผ่านจุดเช็คทั้งสามจุด.....34
รูปที่ 3.4 แสดงขั้นตอนการส่งค่าตัวแปร.....35
รูปที่ 3.5 แสดงการนำค่าพิกัดจุดไปจำลองการเคลื่อนที่.....36
รูปที่ 3.6 แสดงการเช็คว่าพินล้มกี่พิน.....36
รูปที่ 3.7 แสดงจำนวนพินที่ล้ม.....37
รูปที่ 3.8 เลนโบว์ลิ่งจากโปรแกรม Blender.....37
รูปที่ 3.9 เลนโบว์ลิ่งจริง.....38
รูปที่ 3.10 ลักษณะของเลนโบว์ลิ่ง.....39
รูปที่ 4.1 รูปขั้นตอนการทำงานของโปรแกรมเบลนเดอร์.....47
รูปที่ 4.2 รูปโมเดลพิน.....47
รูปที่ 4.3 รูปโมเดลลูกโบว์ลิ่ง.....48
รูปที่ 4.4 รูปโมเดลเลนโบว์ลิ่ง.....48
รูปที่ 4.5 รูปแสดงการใส่ Logic และปุ่มควบคุมการทำงานต่างๆให้กับแต่ละส่วนของเกมส์.....53
รูปที่ 4.6 รูปแสดงการใส่ Logic ให้กับพินโบว์ลิ่ง.....54
รูปที่ 4.7 เป็นการแสดงหน้าจอของเกมส์.....54
รูปที่ 4.8 แผงวงจรควบคุมการทำงานของ Sensors.....55
รูปที่ 4.9 ตัวเซนเซอร์ และ เลเซอร์.....56
รูปที่ 4.10 ภาพการนำเลเซอร์มาประกอบติดกับเลนโบว์ลิ่ง.....56
รูปที่ 4.11 ภาพการนำเซนเซอร์มาประกอบติดกับเลนโบว์ลิ่งส่วนปลายเลน.....57
รูปที่ 4.12 ภาพของบอร์ด CP-JR ARM7 USB-LPC2148 EXP ที่ใช้ในระบบ.....57
รูปที่ 4.13 ภาพของสายแพขนาด 10 พินที่ใช้ในระบบ.....58
รูปที่ 4.14 ภาพแสดงการเชื่อมต่ออุปกรณ์.....59
รูปที่ 4.15 ภาพแสดงการเคลื่อนที่ของลูกโบว์ลิ่งเมื่อผู้ใช้เริ่มโยน(1).....60
รูปที่ 4.16 ภาพแสดงการเคลื่อนที่ของลูกโบว์ลิ่งเมื่อผู้ใช้เริ่มโยน(2).....60
รูปที่ 4.17 ภาพแสดงการจำลองพินและเลนส์โบว์ลิ่งจากโปรแกรมเบลนเดอร์.....61
รูปที่ 4.18 ภาพจำลองจากโปรแกรมเมื่อลูกโบว์ลิ่งเคลื่อนที่ผ่านเซนเซอร์ตัวสุดท้าย.....61
รูปที่ 4.19 ภาพจำลองจากโปรแกรมที่เมื่อลูกโบว์ลิ่งเคลื่อนที่ชนพิน(1).....62
รูปที่ 4.20 ภาพจำลองจากโปรแกรมที่เมื่อลูกโบว์ลิ่งเคลื่อนที่ชนพิน(2).....62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

กีฬาโบว์ลิ่งเป็นกีฬาที่ไม่ค่อยได้รับความนิยมอย่างแพร่หลายเพราะมีสถานที่ให้เล่นน้อย การเล่นแต่ละครั้งต้องมีค่าใช้จ่ายสูง ทางทีมผู้พัฒนาจึงคิดระบบที่จะช่วยส่งเสริมให้กีฬาโบว์ลิ่งได้รับความนิยมอย่างแพร่หลายมากขึ้น โดยในปัจจุบัน เทคโนโลยีทางด้าน Microcontroller และ 3D เป็น เทคนิคที่ได้รับความนิยมและเข้ามามีบทบาทในชีวิตประจำวันอย่างเห็นได้ชัด

จากแนวคิดข้างต้นนี้ จึงได้มีการนำเอาเทคโนโลยีเหล่านี้ ผสานกับความต้องการที่ต้องการให้ความสะดวกสบาย และความบันเทิงไว้ในเครื่องเครื่องเดียว และยังมีไว้เพื่อเป็นกิจกรรมเล่นร่วมกันภายในครอบครัว จึงได้มีการคิดพัฒนาจำลองเกมส์โบว์ลิ่งเสมือนจริง ที่จะสามารถตอบสนองความต้องการดังกล่าวได้ โดยใช้เทคโนโลยีของ เซนเซอร์ในการตรวจจับการเคลื่อนที่ร่วมกับการประมวลผลของภาพจากคอมพิวเตอร์ โดยจะเป็นการจำลองในส่วนของพินโบว์ลิ่ง และจะมีการใช้เลนส์โบว์ลิ่งที่มีขนาดของความกว้างเท่าของจริงแต่มีความยาวที่สั้นลงจากโปรแกรมเบลนเดอร์เป็นแบบ “Mixed Reality” เพื่อให้สามารถนำไปติดตั้งในสถานที่ต่าง ๆ ได้ ซึ่งถือว่าเป็นอีกหนึ่งเทคโนโลยีที่น่าสนใจ

1.2 วัตถุประสงค์

1. เพื่อนำเทคโนโลยีไปใช้พัฒนาเกมกีฬาให้สามารถเล่นได้ง่ายขึ้นผู้คนจะได้หันไปเล่นกีฬากันมากขึ้น
2. เพื่อให้กีฬาโบว์ลิ่งเป็นกีฬาที่ได้รับความนิยมและ แพร่หลายเพราะต่อไปผู้ที่อยากเล่นโบว์ลิ่งไม่จำเป็นที่จะต้องไปเล่นเฉพาะในห้างสรรพสินค้าเท่านั้นสามารถเล่นได้ทุกที่ ที่ได้นำระบบนี้ไปใช้
3. เพื่อใช้สร้างความบันเทิงและใช้เป็นเครื่องมือเพื่อสร้างกิจกรรมที่ส่งเสริมความสามัคคีและสร้างความอบอุ่นภายในครอบครัว
4. เพื่อช่วยให้คนที่ติดเกมหรือชอบเล่นเกม ได้ออกกำลังกายไประหว่างเล่นเกมเพราะระบบนี้ก็ จัดได้ว่าเป็นการนำกีฬามาผนวกกับเกม ส่งผลให้ผู้เล่นมีสุขภาพอนามัย ที่ดีขึ้น ไม่ต้องนั่งเล่นเกมอยู่กับที่เหมือนเคย
5. เพื่อประหยัดค่าใช้จ่ายในการเล่นโบว์ลิ่งเพราะปัจจุบันถ้าใครจะกีฬาประเภทนี้ก็จะต้องไปตามห้างต่างๆและจะเสียค่าใช้จ่ายในการเช่าเลนโบว์ลิ่งและรองเท้าเฉพาะของกีฬานี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

ระบบสามารถจำลองเลนโบว์ลิ่งจากเลนโบว์ลิ่งจริงๆขนาด 6 เมตรให้มีขนาด 22 เมตรและเมื่อผู้เล่นทำการโยนลูกโบว์ลิ่งไปบนเลนโบว์ลิ่งจริงที่มีขนาดความยาว 6 เมตร ระบบก็สามารถที่จะทำการ Simulate การเคลื่อนที่ของลูกโบว์ลิ่งให้เป็น 22 เมตรจากระยะที่ลูกโบว์ลิ่งวิ่งจริงๆ เพียงแค่ 6 เมตร ซึ่งหลังจากได้ตำแหน่งของลูกโบว์ลิ่ง ณ ตำแหน่งที่ 22 เมตรแล้ว ก็จะทราบว่าลูกโบว์ลิ่งจะกระทบพิน ณ ตำแหน่งไหน ทิศทางใด ระบบก็จะสร้างแบบจำลองการเคลื่อนที่ของลูกโบว์ลิ่ง และการกระทบพินของลูกโบว์ลิ่งออกมาเป็น 3D โดยคำนวณการล้มของพินและการเคลื่อนที่ของลูกโบว์ลิ่งโดยใช้หลักการทางฟิสิกส์

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาการทำงานของเซนเซอร์ที่เหมาะสมกับการตรวจจับการเคลื่อนที่ของลูกโบว์ลิ่ง
2. ศึกษาการทำงานของ API ที่จะใช้ติดต่อเพื่อจะรับค่าจาก Microcontroller
3. ทำระบบเซนเซอร์ตรวจจับการเคลื่อนที่ของลูกโบว์ลิ่ง
4. Simulate เส้นทางการเคลื่อนที่ของลูกโบว์ลิ่งโดยใช้โปรแกรมเบลนเดอร์หาเส้นทางที่เหลือจากการเคลื่อนที่ - เมตร
5. หาพิกัดที่ลูกโบว์ลิ่งกระทบพิน ว่ากระทบพิน ณ ตำแหน่งไหน
6. พัฒนาตัวระบบที่จะนำมาใช้กับเกมโดยสร้าง กติกา, Logicของเกม, การบันทึกผลคะแนน และ การแสดงผลทางหน้าจอต่างๆของผู้เล่น
7. ทดสอบการทำงานของระบบ และทำการแก้ไขข้อบกพร่องที่เกิดขึ้น
8. สรุปและรายงานผล

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ระบบสามารถนำมาใช้ได้เสมือนไปเล่นตามเลนโบว์ลิ่งจริงๆ
2. เด็กและเยาวชนที่ติดเกมหันมาสนใจเกมนี้และเมื่อได้เล่นก็ได้ออกกำลังกายไปด้วย ทำให้ร่างกายแข็งแรงและผ่อนคลายทำให้มีสุขภาพที่ดีทั้งร่างกายและจิตใจ
3. เป็นแนวทางในการศึกษาเพื่อการพัฒนาต่อไป อาจเป็นในเชิงพาณิชย์ หรือเกมส์เอ็นจิน
4. จะได้รับความรู้เกี่ยวกับการใช้งานโปรแกรมเบลนเดอร์
5. จะได้รับความรู้เกี่ยวกันการต่อและการใช้งานไมโครคอนโทรเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 โปรแกรม เบลนเดอร์

โปรแกรมเบลนเดอร์(Blender) เป็นโปรแกรมที่สร้างงาน 3 มิติที่มีความสามารถในการสร้างโมเดลรูปทรงต่าง ๆ ทั้งยังกำหนดพื้นผิดหรือลวดลายให้กับวัตถุได้สามารถจัดแสงกำหนดมุมมองสร้างชิ้นงานให้เป็นแอนิเมชันพร้อมทั้งใส่เอ็ฟเฟ็กต์สร้างความเหมือนจริงและชวนติดตามจนกระทั่งประมวลผลงานทั้งหมดออกมาเป็นงาน 3 มิติที่สมบูรณ์แบบ นอกจากนั้นยังเป็นโปรแกรมที่รองรับการสร้างเกมส์เนื่องจากมีเครื่องมือที่ช่วยในการทำโมเดลสำหรับเกมส์และการทำงานอื่นๆที่เกี่ยวข้องกับเกมส์

2.1.1 จุดเด่นของโปรแกรมเบลนเดอร์

โปรแกรมเบลนเดอร์โปรแกรมโอเพ่นซอส ที่สามารถดาวน์โหลดมาใช้ได้ฟรีและยังมีความสามารถทัดเทียมกับโปรแกรมสร้างงาน 3 มิติโปรแกรมอื่นๆโดยจุดเด่นที่น่าสนใจของโปรแกรมเบลนเดอร์ มีดังนี้

- เป็นโปรแกรมที่ใช้ทรัพยากรระบบและพื้นที่ในการติดตั้งโปรแกรมน้อย
- มีความสามารถในการสร้างงานได้หลายรูปแบบ เช่น การสร้างการ์ตูนแอนิเมชัน งานดีไซน์ งานสถาปัตยกรรมและตกแต่งภายใน การสร้างสเปเชียลเอฟเฟ็กต์ และสร้างเกมเป็นต้น
- รองรับการนำเข้าไฟล์จากโปรแกรมอื่นๆ เช่น Wavefront OBJ, Wings 3D, 3 ดีเอสแมกซ์, LightWave3D, COLLADA และอื่นๆ
- สามารถเขียนโปรแกรมเสริมการทำงานได้ด้วยภาษา ไพทอนสคริป
- มีเกมเอนจินในตัว
- มีเครื่องมือสำหรับใช้ตัดต่อและตบแต่งวีดีโอในตัว
- ทำงานได้หลายแพลตฟอร์มเช่น วินโดว์, แมกซ์, ลินุกซ์ และอื่นๆ
- สามารถระบายสีบนพื้นผิว 3มิติได้ทันที
- เป็นโปรแกรมที่มีกลุ่มผู้ใช้งานมากกว่า 250,000 คนทั่วโลก
- ราคา 0 บาท และยังสามารถนำไปเขียนลงซีดีขายได้โดยไม่ผิดกฎหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมเบลนเดอร์สามารถใช้เป็นโปรแกรมพื้นฐานสำหรับผู้ที่ต้องศึกษาการสร้างงาน 3 มิติ รวมทั้งบริษัทขนาดเล็กและขนาดกลาง ก็สามารถนำไปใช้สร้างชิ้นงานตามวัตถุประสงค์ของบริษัทได้เช่นกัน เนื่องจากไม่ต้องจ่ายงบประมาณในการซื้อโปรแกรมมาใช้นั่นเอง

2.2 การนำโปรแกรมเบลนเดอร์มาใช้ในการสร้างโมเดล

2.2.1 การสร้างโมเดล

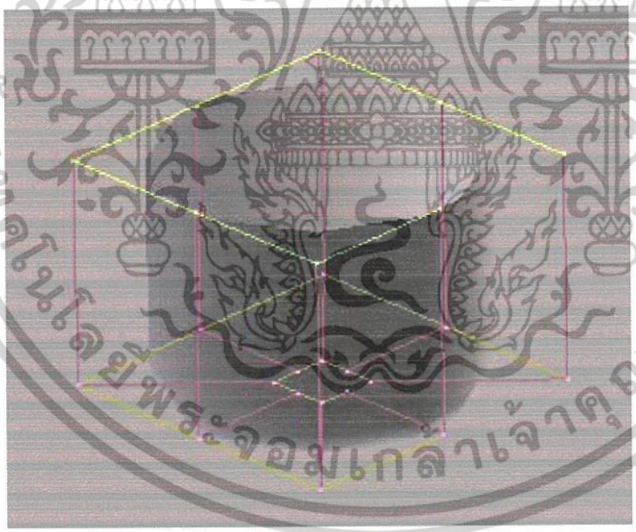
2.2.1.1 การสร้างพินโบว์ลิ่ง

โดยมีขั้นตอนการสร้างดังนี้

- สร้างก้นพินโบว์ลิ่ง

เราจะสร้างพินจากพื้นก้นพินด้านล่างให้เป็นฐานก่อนจึงจะขึ้นโมเดลไปถึงส่วนหัวของพินได้ โดยเริ่มต้นจากการเลือกเส้น NURBS Circle จากนั้นสร้างพื้นผิวที่ก้นพินด้วยวิธีดังนี้

1. ขยายพื้นผิวก้นพินโดยกดคีย์<E>สร้างพื้นผิวก่อนเพื่อสร้างพื้นผิวแล้วกด<S> ขยายออกมาจากนั้นคลิกเมาส์ปุ่มซ้ายเพื่อกำหนดพื้นผิว



รูปที่ 2.1 รูปแสดงการขยายก้นผิวของพิน

2. เราจะปรับให้อยู่ในมุมองเป็นเปอร์สเปกทีฟเพื่อดูพื้นผิวในอีกมุมมองซึ่งเราจะเห็นการเปลี่ยนแปลงในการ Extrude ได้ชันเจน จากนั้นเลือกจุดควบคุมภายนอกทุกจุดของเส้น NURBS Circle แล้วกดแป้น <E> ดึงพื้นผิวขึ้นมาใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แกน Z โดยกดแป้น <Z> จะปรากฏเส้นสีฟ้าแนวตั้งในแนวแกน Z เมื่อได้ความสูงของพื้นผิดที่ต้องการแล้วให้คลิกเมาส์ปุ่มซ้ายเพื่อกำหนดพื้นผิว

- สร้างตัวและคอพินโบว์ลิ่ง

เราจะสร้างส่วนของตัวพินซึ่งนอกจากจะสร้างในมุมมองเป็นเปอร์สเปกทีฟ แล้วเรายังสามารถสร้างในมุมฟรอนวิวได้เช่นกันและมุมมองนี้จะช่วยให้เราเห็นในส่วนของพินที่เราจะสร้างได้ชัดเจน โดยมีวิธีการสร้างดังนี้

1. สร้างตัวขวดโดยกดคีย์ <E> ดึงพื้นผิวขึ้นมาในแกน Z โดยกดปุ่มแป้น <Z> จะปรากฏเส้นสีฟ้าแนวตั้งในแนวแกน Z
2. ย่อปลายตัวขวดด้วยการกดคีย์ <S> แล้วลากเมาส์เพื่อย่อ จากนั้นคลิกเมาส์ซ้าย 1 ครั้งเพื่อกำหนดขนาดของตัวพิน
3. สร้างส่วนคอของพินโดยกดคีย์ <E> ดึงพื้นผิวขึ้นมาในแกน Z โดยกดปุ่ม <Z> จะปรากฏเส้นสีฟ้าแนวตั้งในแนวแกน Z

รูปที่ 2.2 รูปแสดงส่วนของพินในมุมฟรอนวิว

- สร้างส่วนหัวพินโบว์ลิ่ง

โดยมีขั้นตอนการสร้างดังนี้

1. สร้างส่วนหัวของพินโดยกดคีย์ <E> ดึงพื้นผิวขึ้นมาในแกน Z โดยกดแป้น <Z> แล้วย่อส่วนหัวขอพินหุบเข้ามาด้วยการกด <S> แล้วลากเมาส์เพื่อย่อ
2. จากนั้นให้ Extrude ปากขวดขึ้นมา 2 ครั้งต่อกัน แล้วกดแป้น <S> ย่อส่วนหัวของขวดบีบเข้ามาจนถึงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เก็บรายละเอียด

1. จากขั้นตอนที่ผ่านมา เราก็จะได้รูปทรงพินที่สร้างจากเส้น NURBS Curve และเมื่อเราเลื่อนมุมมองมาที่มุมมองมาที่ก้นพิน จะพบว่าส่วนก้นของพินนั้นมีช่องว่างที่พื้นผิวไม่ติดกัน เนื่องจากเราสร้างเส้นวงกลมแล้ว Extrude ออกมา จึงทำให้มีรูหรือช่องว่างบริเวณก้นพิน วิธีแก้ไขคือ ต้องรวมจุดรอบวงกลมนั้นให้กลายเป็นจุดเดียวกัน โดยเลือกจุดใดจุดหนึ่งในวงกลม แล้วกดคีย์ <Shift+R> หรือเลือก คำสั่ง Select>Control Point Row เพื่อเลือกจุดใดในแถวเดียวกันกับจุดที่เราเลือก

รูปที่ 2.3 รูปแสดงส่วนของพินในขั้นตอนการตกแต่ง

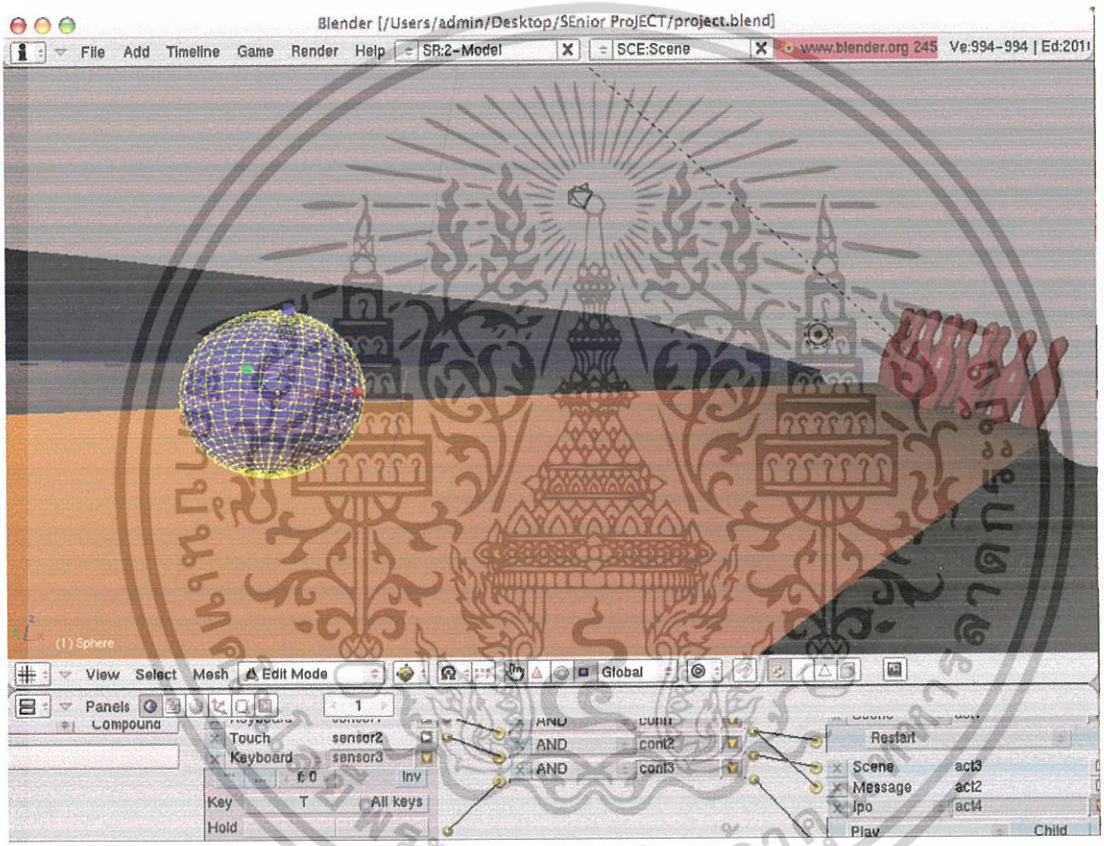
2. ต่อมาให้เรากดคีย์ <S> แล้วกดคีย์ <0> ที่ Numpad เพื่อย่อทุกจุดเข้ามารวมกันให้เป็นจุดเดียวกัน จะทำให้เราได้ผิวล่างที่ติดกันเป็นก้นขวดที่ไม่มีช่องว่างใดๆ จากนั้นกดคีย์ <Enter>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.2 การสร้างลูกโบว์ลิ่ง

โดยมีขั้นตอนการสร้างดังนี้

1. กดเลือก Add > Mesh > UVSpher แล้วจะมี Pop-up ให้ใส่เกี่ยวกับ Properties ขึ้นมา โดยกำหนด Segments = 42 , Ring = 32 และ Raduis = 2 แล้วกด Ok.
2. ทำการ Scaling ให้ได้ขนาดที่เหมาะสมกับขนาดของพินโบว์ลิ่ง
3. ใส่ Surface ให้กับลูกโบว์ลิ่งโดยเลือก Add New Surface แล้วเลือกสีให้เหมาะสม



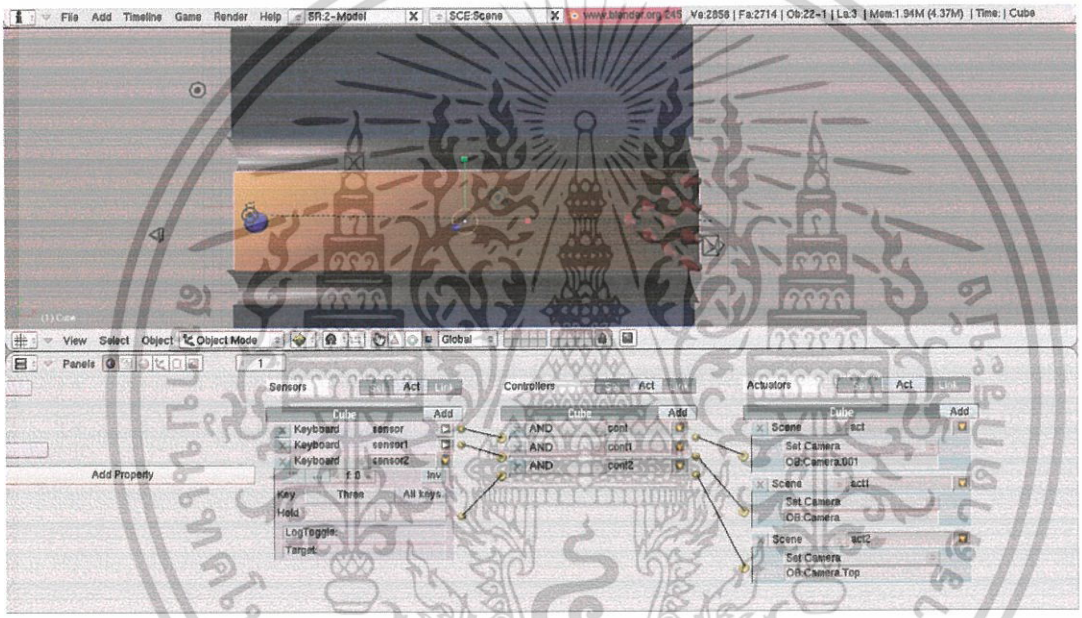
รูปที่ 2.4 รูปแสดงส่วนของลูกโบว์ลิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.3 การสร้างเลนโบว์ลิ่ง

โดยมีขั้นตอนการสร้าง ดังนี้

1. กดเลือก Add > Mesh > Cube ก็จะได้รูปลูกบาศก์มา 1 รูป
2. ทำการ Scaling ลูกบาศก์ให้มีลักษณะเป็นแท่งคล้ายกับเลนโบว์ลิ่ง เมื่อได้ขนาดที่ต้องการแล้วก็ทำการ Copy เลนโบว์ลิ่งขึ้นมาอีกสองอัน ไว้เป็นบริเวณลานด้านข้างโบว์ลิ่งโดยทำการเลือก Object > Duplicated หรือ กดแป้น <Shift+D>
3. ก็จะได้เลนโบว์ลิ่งขึ้นมาก็นำมาจัดตำแหน่งให้ได้ตำแหน่งที่มีลักษณะเหมือนลานโบว์ลิ่ง



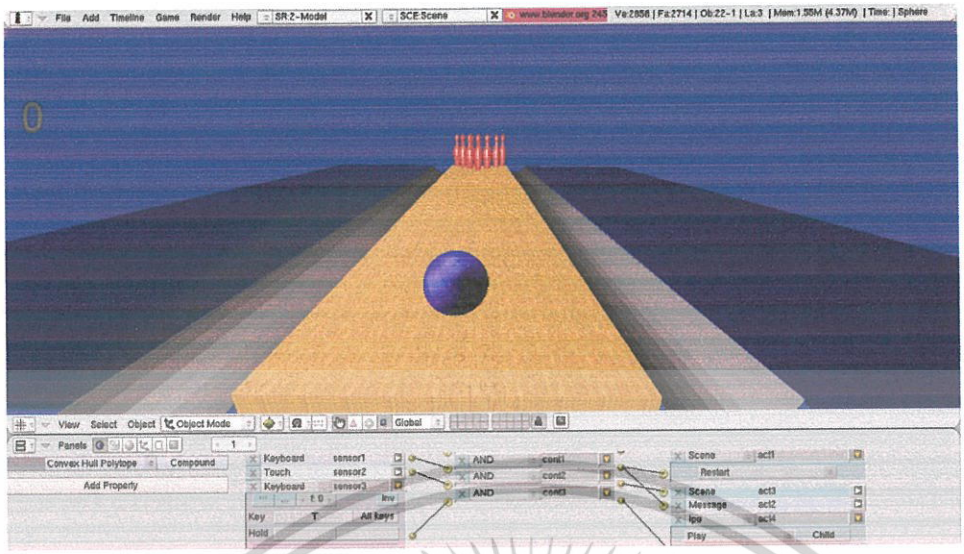
รูปที่ 2.5 รูปแสดงส่วนเลนโบว์ลิ่ง

2.2.1.4 การสร้างท่อของเลนโบว์ลิ่ง

โดยมีขั้นตอนการสร้างดังนี้

1. สร้างท่อของเลนโบว์ลิ่งโดยใช้ Add > Curves > NURBS Curve แล้วปรับระดับความโค้งและขนาดของเส้น Curve ให้ได้ขนาดเท่ากับท่อ
2. ทำการ Extrude ลากเส้น Curve ให้มีขนาดยาวเท่ากับขนาดของเลนโบว์ลิ่งที่สร้างไว้
3. สุดท้ายก็จะได้เลนโบว์ลิ่งที่มีองค์ประกอบด้านวัตถุครบถ้วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 รูปแสดงส่วนของท่อเลนโบว์ลิ่ง

2.2.2 การสร้างส่วนของเกมส์โดยใช้ Game Engine ของโปรแกรมเบลนเดอร์

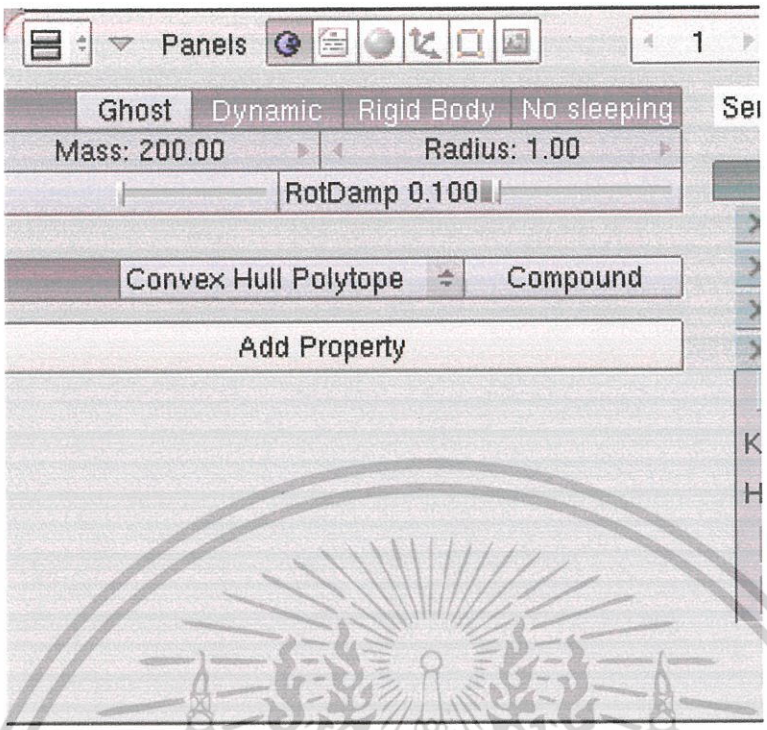
Blender Game Engine คือ สิ่งที่ช่วยสร้างเกม เปรียบเสมือนเครื่องมือช่วยเหลือให้การทำงาน มีความสะดวกและรวดเร็วมากขึ้น ตัวอย่างเช่น Engine ที่ถูกสร้างขึ้นเพื่อช่วยในการเรนเดอร์ เกมของคุณให้มีความสวยงามมากขึ้น หรือ จัดการด้านกราฟิก ด้านอุปกรณ์ต่อพ่วง และด้านเสียง หรืออาจมีส่วนประกอบเพิ่มขึ้นมาเพื่อการจัดการด้านกายภาพ การสร้างพื้นฉาก การตรวจจับการชนและอื่น ๆ อีกมากมาย

2.2.2.1 การกำหนดมวลให้วัตถุที่สร้างขึ้น

เราจะได้นำไปใช้ในการสร้างเกมโดยใช้หลักการทางฟิสิกส์มาคำนวณ ซึ่งถูกจัดเตรียมไว้อยู่แล้วใน Blender Game Engine ซึ่งวัตถุใน Virtual Strike Bowl ที่จะต้องมีการกำหนดมวลให้คือ พินโบว์ลิ่ง และลูกโบว์ลิ่ง กำหนดได้ดังนี้

1. คลิกขวาเลือกวัตถุที่ต้องการจะกำหนดมวลให้ (ในที่นี้ได้แก่ พินโบว์ลิ่ง และ ลูกโบว์ลิ่ง)
2. เสร็จแล้วกดเลือกหน้าต่างไปที่ Game Logic แล้วกดเลือก Actor แล้วกำหนดมวลให้กับวัตถุตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 รูปแสดงส่วนของการกำหนดวัตถุที่เราสร้างมีมวล

2.2.2.2. การกำหนด อินพุต และแอคชั่น ที่เกิดขึ้น

ค่าที่รับเข้าไปจะได้จากการประมวลผลของไมโครคอนโทรลเลอร์โดยจะรับค่าผ่านทาง Python Scripts และเมื่อไมโครคอนโทรลเลอร์ส่งค่าเข้าไปค่าเข้าไป ก็จะต้องมาทำการเซ็ตให้ลูกโบว์ลิ่งเกิดการเคลื่อนที่ตามเส้นทางที่ได้จากการเคลื่อนของลูกโบว์ลิ่งที่ได้จากการคำนวณในไมโครคอนโทรลเลอร์ โดยเราจะสามารถกำหนดเส้นทางได้ด้วยการใช้ IPO Curve ในการกำหนดเส้นทางให้กับวัตถุ โดยมีการระบุพิกัดจุดแค่เพียง 3 จุดเท่านั้น ลูกโบว์ลิ่งในเบลนเดอร์ ก็จะวิ่งผ่านจุด 3 จุดดังกล่าว ไปกระทบพิน ซึ่งการกระทบพินก็จะเป็นไปตามกฎทางฟิสิกส์ของการกระแทกกันของวัตถุ ซึ่งจะดูจากทิศทางของวัตถุ ความเร็วของวัตถุที่เข้ามากระทบ และจะแสดงผลออกมาว่าพินไหนล้มบ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ทฤษฎีการทำงานของเซนเซอร์

ในการทำงานของอุปกรณ์เซ็นเซอร์ใช้ในการควบคุมการป้อนกลับภายใน และค่าจากภายนอก มนุษย์และสัตว์ต่างมีสัญชาตญาณการรับรู้เหมือนกันอย่างเช่น เมื่อตื่นนอน ก่อนที่จะลืมตาขึ้นนั้นจะรู้สึกและรับรู้ถึงแขนและขาคุณอยู่อย่างไร เพราะเส้นประสาทจะส่งสัญญาณไปยังสมอง และเช่นเดียวกันกับการเกร็งและคลายกล้ามเนื้อก็จะมีสัญญาณที่ส่งไปยังสมองเช่นกัน

คล้ายกับหุ่นยนต์ที่แขนและข้อต่อจะมีการเคลื่อนไหว เซ็นเซอร์จำพวก Potentiometer ที่ทำการถอดรหัสและแปลงสัญญาณไปให้ส่วนควบคุมได้รับรู้ว่ามีการเคลื่อนไหวอย่างไร นอกจากนี้ในขณะที่มนุษย์และสัตว์นั้นรับรู้ได้ทางประสาทสัมผัสทั้ง 5หุ่นยนต์ก็สามารถรับรู้ได้เช่นกัน เซ็นเซอร์ถูกออกแบบมาให้มีการทำงานเลียนแบบการรับรู้ของมนุษย์ และนอกจากนั้นยังสามารถรับรู้ได้มากกว่ามนุษย์อีกด้วย

2.3.1 คุณสมบัติทั่วไปของเซนเซอร์

ราคา เป็นสำคัญในการพิจารณาเลือกใช้ให้เหมาะสมกับอุปกรณ์ที่ต้องการ

ขนาด ขึ้นอยู่กับความสามารถของเซ็นเซอร์ ขนาดมีความสำคัญมาก อย่างเช่นอุปกรณ์ที่มีพื้นที่ขนาดเล็ก จำเป็นต้องใช้เซ็นเซอร์ที่มีขนาดเล็กตามไปด้วย

น้ำหนัก จะต้องคำนึงถึงอุปกรณ์ที่มีการเคลื่อนที่ได้ เพราะน้ำหนักจะมีผลเป็นการเพิ่มแรงเฉื่อยของการเคลื่อนที่

ชนิดของเอาท์พุท มีความสำคัญในการพิจารณาเอาท์พุทที่จะนำไปใช้งานในรูปแบบต่างๆ กัน อย่างเช่น ดิจิตอลจะใช้เป็นเอาท์พุทใช้งานกับไมโครโพรเซสเซอรี

การเชื่อมต่อ จะต้องคำนึงถึงการเชื่อมต่อตัวอุปกรณ์และเซ็นเซอร์ เช่น วงจรหรือสายสัญญาณต่างๆ

ความละเอียด เป็นการพิจารณาค่าเอาท์พุทของเซ็นเซอร์ว่าให้ความละเอียดได้มากเพียงใด เช่น เอาท์พุท 4บิต ให้ความละเอียดได้ 16ระดับ

ความอ่อนไหว เป็นอัตราการเปลี่ยนแปลงของเอาท์พุทที่มีผลมาจากการเปลี่ยนแปลงของอินพุทว่ามีผลมากน้อยเพียงใด

ความสัมพันธ์แบบเส้นตรง เป็นความสัมพันธ์ในการแปรผันตรงแบบเชิงเส้นระหว่างอินพุทและเอาท์พุท เมื่ออินพุทเข้าไปเท่าไรค่าก็จะเปลี่ยนเป็นเอาท์พุทออกมาในอัตราที่เท่าเดิมเสมอ

ช่วง เป็นช่วงที่แตกต่างกันระหว่างเอาท์พุทที่น้อยที่สุดและมากที่สุดที่เซ็นเซอร์สามารถจะรับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตอบสนอง เป็นความรวดเร็วในการตรวจจับและแสดงค่าความต่างของเอาท์พุทเมื่อมีการเปลี่ยนแปลงของอินพุท

ความถี่ในการตอบสนอง ความสามารถในการรับความถี่ของสัญญาณอินพุท เนื่องจากบางสภาวะแวดล้อมที่เราต้องการตรวจจับ เช่น เสียง หรือคลื่นวิทยุจะต้องใช้ความถี่ในการจำแนกและตรวจจับ

ความน่าเชื่อถือ คือ อัตราความถูกต้องของการทำงานต่อจำนวนครั้งที่ใช้งาน

ความแม่นยำ คือ ความถูกต้องของค่าเอาท์พุทที่ใกล้เคียงกับค่าที่คาดการณ์เอาไว้

การทำงานซ้ำ ความสามารถในการทำงานอย่างเดิมหลายครั้งและสามารถทำงานได้อย่างถูกต้องตามคุณสมบัติ

2.3.2 ประเภทของเซนเซอร์

เซนเซอร์วัดตำแหน่ง (Position sensors)

- Potentiometer
- Encoder

เซนเซอร์วัดความเร็ว (Velocity sensors)

- Encoders

เซนเซอร์วัดแสงและอินฟาเรด(Light and Infrared sensors)

เซนเซอร์ประมาณระยะ (Proximity sensors)

- Optical proximity sensors

2.3.3. ลักษณะการทำงานของตัวต้านทานปรับค่าได้ (Position sensors)

โพเทนชันมิตเตอร์ จะแปลงค่าตำแหน่งให้กลายเป็นการเปลี่ยนแปลงของค่าความต่างศักย์ โดยมาจากค่าความต้านทานที่เปลี่ยนไป เช่น การเปลี่ยนแปลงค่าความต้านทานแคบก็จะเกิดจากการเปลี่ยนแปลงตำแหน่งในช่วงแคบเช่นกัน ซึ่งค่าความต้านทานก่อนและหลังที่เกิดการเปลี่ยนแปลงจะถูกกำหนดโดยช่วงทั้งหมดของตัวต้านทานปรับค่าได้ โดยที่ความสามารถของโพเทนชันมิตเตอร์ จะทำงานแบบการแบ่งกระแสที่ไหลผ่านตัวมันเอง ทำให้สามารถคำนวณผลลัพธ์ได้จากความสัมพันธ์ของความต้านทานคือ

$$V_{out} = V_{cc}*(R_1/R) \qquad (2.1)$$

โพเทนชันมิตเตอร์ มีทั้งแบบหมุนและแบบแถบเลื่อนขึ้นอยู่กับการใช้งานว่ามีการเคลื่อนไหวในแบบใด โดยในแบบหมุนได้นั้นมีแบบที่เป็นหมุนได้หลายรอบ ทำให้สามารถใช้ในอุปกรณ์ที่มีการเคลื่อนแบบหมุนวนไหลรอบได้

โพเทนชันมิตเตอร์ มีทั้งแบบ wire wound และแบบ thin film deposit ซึ่งก็ทำจากพลาสติก นั้นค่าความต้านทานที่ต่างกันจะอยู่ที่พื้นผิวหน้าสัมผัส โดยคุณลักษณะพิเศษของแบบ thin film deposit คือการให้ค่าเอาท์พุทที่ต่อเนื่องและมีสัญญาณรบกวนต่ำ ซึ่งก็เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลให้สามารถนำไปใช้งานทำการหาอัตราความต่างของเอาท์พุท differentiate output ซึ่งจะสามารถนำไปหาอัตราเร็วต่อไป ซึ่งเอาท์พุทของแบบ wire wound เป็นแบบไม่ต่อเนื่องทำให้ไม่สามารถนำไปใช้งานแบบหาอัตราความต่างเอาท์พุทได้

โดยทั่วไปแล้วโพเทนชันมิตเตอร์ จะใช้ในการหาการป้อนกลับภายในเป็นลำดับเพื่อหาตำแหน่งปัจจุบันของแขนและจุดเชื่อมต่อของอุปกรณ์โพเทนชันมิตเตอร์ นั้นสามารถใช้งานตามลำพังหรืออาจใช้งานร่วมกับตัวเซ็นเซอร์อื่นๆ ได้เช่น ตัวถอดรหัส ในกรณีนี้ ตัวถอดรหัสจะรายงานค่าตำแหน่งในปัจจุบันของแขนและจุดเชื่อมต่อของอุปกรณ์ดังเช่นรายงานเมื่อตำแหน่งเริ่มต้นทำงาน ผลที่ได้จากการใช้งานร่วมกันหลายเซ็นเซอร์สิ่งที่ต้องการคือค่าของเอาท์พุทที่น้อยแต่มีความชัดเจนและแม่นยำสูง

2.3.4 ลักษณะการทำงานของเซนเซอร์แสงและอินฟราเรด (Light and Infrared sensors)

เซ็นเซอร์ชนิดนี้จะทำงานโดยความเข้มของแสงที่ตกกระทบตัวมันเองจะเป็นตัวเปลี่ยนค่าความต้านทานทางไฟฟ้า ถ้าความเข้มแสงน้อยหรือต่ำจนเป็นศูนย์ค่าความต้านจะมีค่ามากที่สุด แต่เมื่อมีความเข้มแสงตกกระทบจะทำให้ค่าความต้านทางไฟฟ้าต่ำลงทำให้เกิดกระแสไหลผ่านได้มาก ผลที่ได้คือ ค่าความต่างศักย์ที่ตกคร่อมจะมีค่าน้อย เซ็นเซอร์ชนิดนี้มีราคาไม่แพงและยังมีประโยชน์มากมายในการนำมาประยุกต์ใช้งาน โดยสามารถนำมาสร้างเป็นตัวตรวจจับสัญญาณโดยแสงได้ และอุปกรณ์อื่นๆ อีกมากมาย ทั้งยังสามารถนำไปใช้งานในเซ็นเซอร์วัดพื้นผิวได้ด้วย

ตัวเซ็นเซอร์แสงที่นิยมใช้กันนั้นคือ โฟโต้ทรานซิสเตอร์ ซึ่งจะทำงานเมื่อมีแสงที่มีความเข้มถึงขั้นมากระทบจะทำให้โฟโต้ทรานซิสเตอร์เปิด และเมื่อไม่มีแสงมากระทบหรือความเข้มน้อยกว่าที่ระบุก็จะทำให้อยู่ในสถานะปิดโดยปกติแล้วโฟโต้ทรานซิสเตอร์จะทำงานโดยใช้แหล่งกำเนิดแสงจำพวก LED

การประยุกต์ใช้งานโดยอาจมีการนำเซ็นเซอร์มาวางเรียงกันเพื่อใช้วัดระยะการเปลี่ยนตำแหน่งไปของตัวกำเนิดแสง ซึ่งจะนำไปใช้งานในการวัดค่าความผิดปกติหรือระยะการเคลื่อนที่น้อยๆ ของตัวเครื่องมือต่างๆ

เซ็นเซอร์แสงมีผลกระทบจากช่วงความถี่ของแสงปกติ เซ็นเซอร์อินฟราเรดก็จะได้รับผลกระทบจากช่วงของความถี่อินฟราเรดเช่นกัน เนื่องจากอินฟราเรดไม่สามารถมองเห็นได้โดยตา ทำให้ไม่เกิดการรำคาญหากต้องใช้เป็นส่วนในอุปกรณ์สักอย่างที่มีการรับส่งแสงกัน เช่นว่า อุปกรณ์อย่างหนึ่งที่ต้องการแสงในการวัดระยะทางที่ไกลเพื่อนำมาสร้างเป็นระบบนำทาง ในที่นี้จะนิยมใช้อินฟราเรดมากกว่า เพราะว่าไม่เป็นที่สนใจและไม่รบกวนผู้อื่น หรือหลบเลี่ยงจากการก่อกวนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่ใช้อินฟราเรด เช่น รีโมตทีวี ซึ่งเป็นอุปกรณ์ที่มีประโยชน์มากในการควบคุมอุปกรณ์ทำงานต่างๆ ได้จากระยะไกล

2.3.5 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

- **Analog Computer สัญญาณอนาลอก** คือ สัญญาณข้อมูลแบบต่อเนื่อง (Continuouse Data) มีขนาดของสัญญาณไม่คงที่ การเปลี่ยนแปลงขนาดของสัญญาณแบบค่อยเป็นค่อยไปแปรผันตามเวลา เป็นสัญญาณที่มนุษย์สามารถสัมผัสได้ เช่น แรงดันของน้ำ

- **Digital Computer สัญญาณดิจิตอล** คือ สัญญาณข้อมูลแบบไม่ต่อเนื่อง (Discrete Data) มีขนาดของสัญญาณคงที่ การเปลี่ยนแปลงขนาดของสัญญาณเป็นแบบทันที ทันใด ไม่แปรผันตามเวลา เป็นสัญญาณที่มนุษย์ไม่สามารถสัมผัสได้ เช่น สัญญาณไฟฟ้า

- **Analog to Digital Converter (A/D)** ทำหน้าที่แปลงสัญญาณข้อมูลที่มนุษย์รับรู้ สัมผัสได้ เป็นข้อมูลทางไฟฟ้า เพื่อป้อนเข้าสู่การประมวลผล จึงเป็นขบวนการหนึ่งของการรับข้อมูล (Input Unit) เป็นกระบวนการอีเลคโทรนิคส์ที่สัญญาแปรผันต่อเนื่อง (analog) ได้รับการแปลงให้เป็นสัญญาณดิจิตอลโดยไม่มีการลบข้อมูลสำคัญผลลัพธ์ของ ADC มีลักษณะตรงข้าม คือ กำหนดระดับหรือสถานะ ตัวเลขของสถานะมักจะเป็นการยกกำลังของ 2คือ 2, 4, 8, 16เป็นต้น สัญญาณดิจิตอลพื้นฐานมี 2สถานะและเรียกว่า binary ตัวเลขทั้งหมดสามารถแสดงในรูปของไบนารี ในฐานะข้อความของ หนึ่งและศูนย์

วงจรที่ใช้ในการแปลงสัญญาณอนาลอกเป็นดิจิตอลมีมากมายหลายชนิด โดยทั่วไปแล้ววงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล (A/D converters) มีใช้งานอยู่ประมาณ 7ชนิดคือ

1. Paralled Comparator, Simultaneous, หรือ Flash A/D converter
2. Single – Ramp หรือ Single – Slope A/D converter
3. Dual – Slope A/D converter
4. Charge balance A/D converter
5. A/D converters using Counters and D/A converters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Tracking A/D converters

7. Successive – Approximation A/D converters

- **Counting Converter**เป็นการแปลงสัญญาณอนาล็อก เป็นสัญญาณดิจิตอล โดยใช้อัลกอริทึม การนับค่าเพิ่มขึ้นเรื่อยๆ แล้วนำผลที่ได้จากการนับไปเปรียบเทียบกับค่าที่ต้องการที่ตั้งไว้ การแปลงสัญญาณอนาล็อก เป็นสัญญาณดิจิตอล มีประโยชน์มากในการควบคุมอุปกรณ์สวิตชิ่ง ซึ่งมีลักษณะการแปลงสัญญาณได้หลายวิธี แต่ละวิธีจะมีอัลกอริทึม ความรวดเร็วในการทำงาน และการใช้อุปกรณ์ฮาร์ดแวร์ต่างกันด้วย

2.3.6 ข้อบ่งเฉพาะของการแปลงสัญญาณ A/D (A/D SPECIFICATIONS)

ข้อบ่งเฉพาะจะบอกถึงขีดความสามารถของ converter โดยทั่วไปแล้วจะมีอยู่หลายคำ เช่น ความแม่นยำ,ความเที่ยงตรง และความเที่ยงตรงเป็นเส้นตรง ซึ่งค่าเหล่านี้จะขึ้นอยู่กับลักษณะของแต่ละวงจร แต่มีข้อบ่งเฉพาะอีกข้อหนึ่งที่ไม่ขึ้นอยู่กับลักษณะของวงจรคือ ค่าผิดพลาดระหว่างค่าจริงของสัญญาณอะนาลอก กับค่าของดิจิตอลที่ใช้แทนค่า (ค่าของ Output ของ A/D converter) ซึ่งเรียกว่า Quantizing error จะมีค่าอยู่ประมาณ 2/1+digit ต่ำสุด (LSB) ของการแปลงสัญญาณซึ่งก็เป็นการบ่งถึงความแม่นยำได้อีกทางหนึ่งด้วยค่าพารามิเตอร์ที่สำคัญอีกตัวหนึ่งสำหรับ A/D converter คือ conversion time หรือค่าเวลาสำหรับการแปลงสัญญาณ ซึ่งมีช่วงเวลาอยู่ประมาณ -10 9วินาที ถึง 3-10วินาที ซึ่งขึ้นอยู่กับชนิดของconverterและจำนวน bit

2.4 อุปกรณ์ที่ใช้ในการสร้างโบว์ลิ่งเสมือนจริง

2.4.1 ไมโครคอนโทรลเลอร์ (CP-JR ARM7 USB-LPC2148)

CP-JR ARM7 USB-LPC2148 เป็นบอร์ดไมโครคอนโทรเลอร์ในตระกูล ARM7TDMI-S Core ซึ่งเลือกใช้ไมโครคอนโทรเลอร์ 16/32-Bit ขนาด 64 Pin แบบใช้พลังงานต่ำเป็น MCU ประจำบอร์ด ซึ่งบอร์ดนี้เลือกใช้ MCU เบอร์ LPC2148 ของ Philips โดยการออกแบบโครงสร้างของบอร์ดนั้นจะเน้นเรื่องของการจัดวางอุปกรณ์พื้นฐานที่จำเป็นต่อการ ศึกษา และ ทดลอง ขั้นพื้นฐาน อย่างเป็นสัดส่วน และเป็นอิสระต่อกันเช่น LED แสดงสถานะของ Output Logic และ Push Button Switch สำหรับสร้างสัญญาณ Logic เพื่อทดสอบการทำงานของ Input หรือ Volume ปรับค่าแรงดัน เพื่อใช้ทดสอบการทำงานของ A/D เป็นต้น โดยวงจรพื้นฐานต่างๆเหล่านี้จะถูกออกแบบและจัดวางเป็นจุดเชื่อมต่อแบบอิสระไว้ โดยยังไม่ได้เชื่อมต่อสัญญาณเข้ากับขาสัญญาณใดๆของ CPU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้ผู้ใช้สามารถปรับเลือกใช้งานวงจรประกอบการทดลองที่มีอยู่ในบอร์ดให้สอดคล้องกับทรัพยากรภายในตัว CPU ได้อย่างเหมาะสมและเป็นอิสระ โดยไม่ยึดติดกับวงจรใดวงจรหนึ่งเพียงอย่างเดียว ตัวอย่างเช่น ขาสัญญาณ GPIO0.8 ของ LPC2148 สามารถเลือกกำหนด การทำงานได้มากถึง 4 แบบ คือ ใช้งานเป็น Input/Output, TXD1, PWM และ ADC ดังนั้นเมื่อผู้ใช้ต้องการทดลองการใช้งานขา GPIO0.8 ในโหมดใดก็สามารถเลือกต่อสัญญาณจาก GPIO0.8 ไปยังวงจรสนับสนุนที่เหมาะสมได้ตามต้องการเช่น

- เมื่อต้องการทดลองการใช้งาน GPUIO0.8 เป็น Input ก็สามารถเลือกต่อสัญญาณจาก GPIO0.8 ไปยังวงจรของ Push Button Switch เพื่อทดลองการทำงานกับ Input
- เมื่อต้องการทดลองการใช้งาน GPUIO0.8 เป็น Output ก็สามารถเลือกต่อสัญญาณจากGPIO0.8 ไปยังวงจรของ LED เพื่อดูสถานะของการทำงานของ Output
- เมื่อต้องการทดลองการใช้งาน GPUIO0.8 เป็น ADC ก็สามารถเลือกต่อสัญญาณจาก GPIO0.8 ไปยังวงจรของ Volume ปรับค่าแรงดัน เพื่อทดสอบอ่านค่าแรงดัง Analog
- เมื่อต้องการทดลองการใช้งาน GPUIO0.8 เป็น TXD1 ก็สามารถเลือกต่อสัญญาณจากGPIO0.8 ไปยังวงจรของ Line Driver ของ RS232 เพื่อทดลองการส่งข้อมูลแบบ RS232

ซึ่งจะเห็นได้ว่า ผู้ใช้สามารถเลือก กำหนด และ ปรับเปลี่ยน การใช้งานขาสัญญาณต่างๆของ CPU ได้อย่างครบถ้วนทุกฟังก์ชั่นการทำงาน ได้โดยอิสระอย่างแท้จริง โดยโครงสร้างของบอร์ดจะนำ MCU มาจัดวงจรร่วมกับอุปกรณ์พื้นฐานที่จำเป็นและจัดขาออกมาให้ใช้งานภายนอกผ่านขั้วต่อ Connector แบบ IDE 10 Pin ซึ่งการจัดเรียงขาสัญญาณจะทำการจัดเรียงอย่างเป็นระเบียบ ตามมาตรฐานของ อีทีที เพื่อให้สามารถต่อใช้งานร่วมกับอุปกรณ์สนับสนุนต่างๆของ อีทีที ได้โดยสะดวก ตัวบอร์ดใช้ไฟ +5VDC จากภายนอก หรือ USB Port (ในกรณีใช้กระแสไม่เกิน 100mA ขณะยังไม่ Enumerate USB และไม่เกิน500mA ขณะ Enumerate USB แล้ว) ซึ่งภายในบอร์ดจะมีวงจร Regulate ขนาด 3.3V/1A จัดเตรียมไว้ให้เรียบร้อยแล้ว และภายในบอร์ดจะมี Connector RS232(UART0) จำนวน 1 Port สำหรับทำการ Download Hex File หรือใช้งานในการสื่อสาร RS232 ในโปรแกรม Application ที่เขียนขึ้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

2.4.2 คุณสมบัติของบอร์ด

1. ใช้ MCU ตระกูล ARM7TDMI-S เบอร์ LPC2148 ของ Philips ซึ่งเป็น MCU ขนาด 16/32-Bit

2. ใช้ Crystal 12.00 MHz โดย MCU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 60 MHz เมื่อใช้งานร่วมกับ Phase-Locked Loop (PLL) ภายในตัว MCU เอง

3. รองรับการโปรแกรมแบบ In-System Programming (ISP) และ In-Application Programming(IAP) ผ่านทาง On-Chip Boot-Loader Software ทางพอร์ต UART0 (RS232)

4. Power Supply ใช้แรงดันไฟฟ้า +5VDC โดยใช้ขั้วต่อแบบ CPA-2PIN จากภายนอก หรือ ใช้พลังงานจาก USB Port ได้ (ในกรณีใช้กระแสไม่เกิน 500mA)

5. ภายใน MCU มีหน่วยความจำโปรแกรมแบบ Flash ขนาด 512KB, Static RAM ขนาด 40KB

6. มีวงจร USB มาตรฐาน 2.0 แบบ Full Speed ภายในตัว (USB Function มี 32 End Point)

7. จำนวน GPIO สูงสุดถึง 47 I/O Pins สามารถเชื่อมต่อกับระบบ I/O ที่เป็นสัญญาณ 5V ได้ ซึ่งขาสัญญาณ GPIO จะมีการใช้งานร่วมกันของ Function อื่นๆอีกดังนี้

- วงจรสื่อสารอนุกรมแบบ SPI จำนวน 2 ช่อง และ วงจรสื่อสารอนุกรมแบบ I2C จำนวน 2 ช่อง
- วงจร ADC ขนาด 10 Bit จำนวน 14 ชุด และ วงจร DAC ขนาด 10 Bit จำนวน 1 ชุด
- วงจร UART แบบ Full-Duplex จำนวน 2 ช่อง คือ UART-0 มาตรฐาน 4 Pin ETT เป็นสัญญาณระดับ RS232 Level และ UART-1 เป็นสัญญาณระดับ TTL Level
- Timer 32-bit จำนวน 2 ช่อง (4 Input Capture / 4 Output Compare), 6-Channel PWM
- Output, Watchdog Timer และ Real Time Clock

8. มีวงจรเชื่อมต่อกับ Character LCD โดยใช้วงจรการเชื่อมต่อแบบ 4 บิต จาก GPIO1[25..31]พร้อมวงจรปรับความสว่างหน้าจอ

9. มีวงจรเชื่อมต่อกับ JTAG ARM ขนาด 20 Pin มาตรฐาน เพื่อทำการ Debug แบบ Real Time ได้

10. มีวงจรทดลองขั้นพื้นฐานสำหรับสนับสนุนการใช้งานและทดลองเรียนรู้ ขั้นพื้นฐานอย่างครบถ้วน จัดเตรียมไว้ภายในบอร์ด (ติดตั้งไว้เฉพาะรุ่น CP-JR ARM7 USB-LPC2148 EXP) ซึ่ง ได้แก่

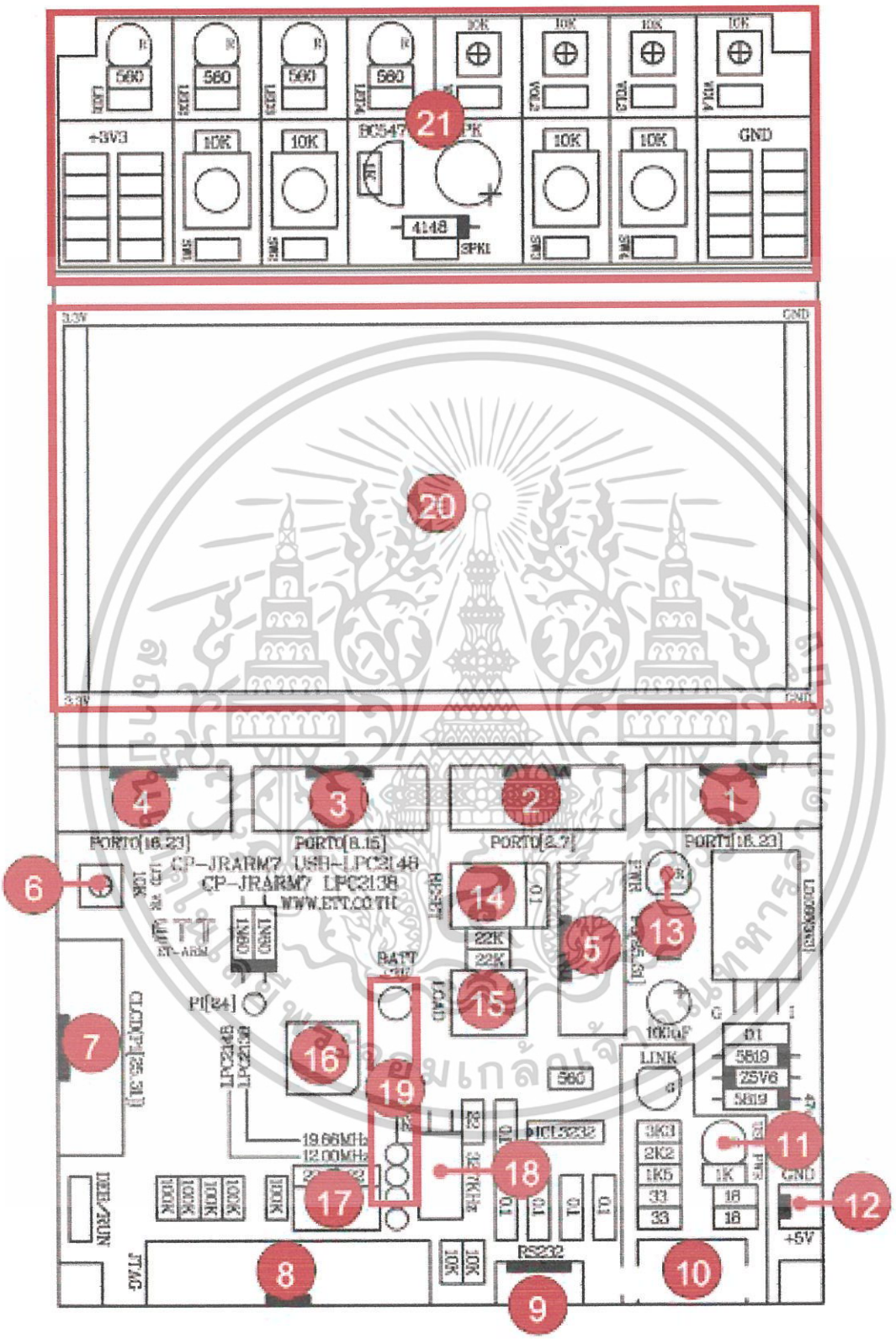
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
06042

- LED Output แบบ Sink Current สำหรับแสดงสถานะของ Output จำนวน 4 ชุด
- Push Button Switch แบบ Active Logic “0” สำหรับทดสอบ Input Logic จำนวน 4 ชุดVolume ปรับค่าแรงดัน 0-3.3V สำหรับทดสอบการทำงานของ ADC จำนวน 4 ชุด
- ชุดกำเนิดสัญญาณเสียง Mini Speaker สำหรับทดสอบการเสียงแบบต่างๆ จำนวน 1 ชุด
- แผงต่อวงจร Project Board รุ่น AD-100 ขนาด 360 จุด สำหรับเป็นพื้นที่ต่อทดลองวงจรขนาดเล็กๆ เพื่อใช้งานร่วมกับ CPU ได้อย่างอิสระ
- จุดต่อแหล่งจ่ายไฟ +3.3V และ GND สำหรับเชื่อมต่อไปยังวงจรภายนอกอื่นๆ

11. ทนอุณหภูมิใช้งานระหว่าง -40 to +85°C

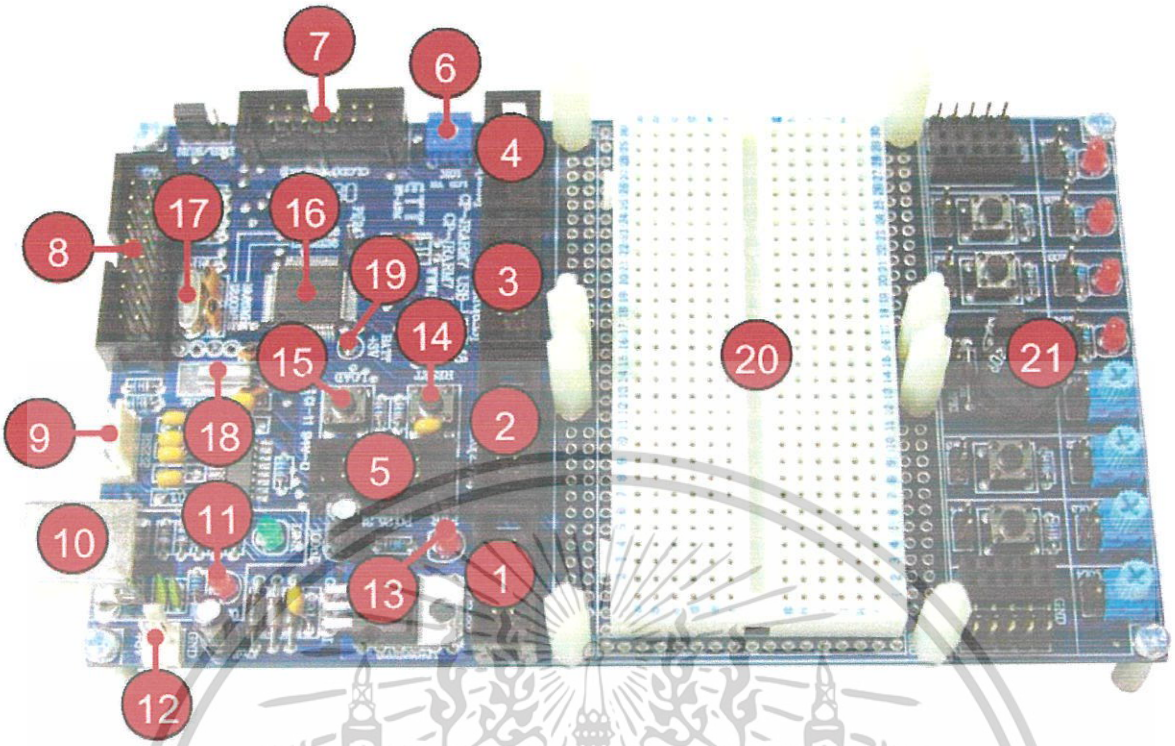
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 โครงสร้างบอร์ด CP-JR ARM7 USB-LPC2148 / EXP



รูปที่ 2.8 ลักษณะโครงสร้างของบอร์ด CP-JR ARM7 USB-LPC2148 / EXP (www.ett.co.th)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ลักษณะของบอร์ด CP-JR ARM7 USB-LPC2148 EXP (www.ett.co.th)

2.4.3.1 รายละเอียดของแต่ละหมายเลขบนบอร์ด CP-JR ARM7 USB-LPC2148 EXP

- **หมายเลข 1 คือ** ขั้วต่อ Port1[16..23] จำนวน 8 บิต
- **หมายเลข 2 คือ** ขั้วต่อ Port0[2..7] จำนวน 6 บิต
- **หมายเลข 3 คือ** ขั้วต่อ Port0[8..15] จำนวน 8 บิต
- **หมายเลข 4 คือ** ขั้วต่อ Port0[16..23] จำนวน 8 บิต
- **หมายเลข 5 คือ** ขั้วต่อ Port0[25..31] จำนวน 7 บิต
- **หมายเลข 6 คือ** ตัวต้านทานสำหรับปรับค่าความสว่าง (Contrast) ของหน้าจอ LCD
- **หมายเลข 7 คือ** ขั้วต่อ Character LCD โดยใช้สัญญาณ Port1[25..31] ในการเชื่อมต่อ
- **หมายเลข 8 คือ** ขั้วต่อ JTAG โดยใช้สัญญาณ Port1[26..31] และ Reset ของ CPU
- **หมายเลข 9 คือ** ขั้วต่อ RS232 สำหรับใช้งาน และ Download Hex File ให้ CPU
- **หมายเลข 10 คือ** ขั้วต่อ USB สำหรับเชื่อมต่อกับ USB Hub รุ่น 2.0
- **หมายเลข 11 คือ** LED แสดงสถานะ Power จาก USB และ สถานของการเชื่อมต่อกับ USB
- **หมายเลข 12 คือ** ขั้วต่อ Power ขนาด +5VDC และ GND เพื่อจ่ายให้กับบอร์ด
- **หมายเลข 13 คือ** LED แสดงสถานะของแหล่งจ่ายไฟ Power ของบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• **หมายเลข 14 คือ** Switch RESET สำหรับสั่ง Reset การทำงานของ CPU

• **หมายเลข 15 คือ** Switch LOAD ใช้ร่วมกับ Switch RESET เพื่อ Download Hex ให้ CPU

• **หมายเลข 16 คือ** CPU เบอร์ LPC2148 ของ Philips ซึ่งเป็น CPU ประจำบอร์ด

• **หมายเลข 17 คือ** Crystal 12.00 MHz สำหรับป้อนให้เป็นสัญญาณนาฬิกาของ LPC2148

• **หมายเลข 18 คือ** Crystal 32.768 KHz สำหรับ Real Time Clock (RTC) ในตัวของ LPC2148

• **หมายเลข 19 คือ** จุดเชื่อมต่อ ลังถ่าน Battery ขนาด +3V (อยู่ด้านใต้บอร์ด) สำหรับต่อให้กับRTC เพื่อเก็บรักษาค่าเวลาของ RTC ในขณะที่ไม่ได้จ่ายไฟเลี้ยงให้กับบอร์ด

• **หมายเลข 20 คือ** แผง Project Board รุ่น AD-100 ขนาด 360 จุด สำหรับต่อวงจร (มีติดตั้งไว้เฉพาะในรุ่น CP-JR ARM7 USB-LPC2148 EXP)

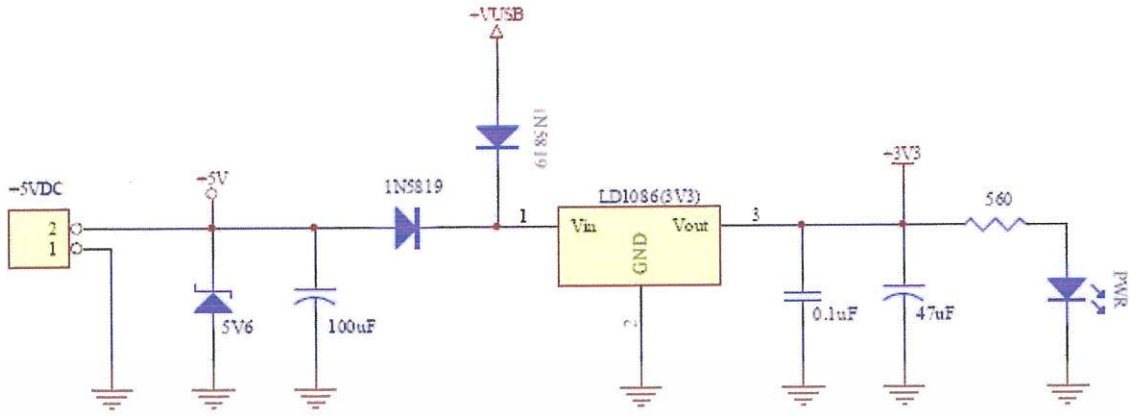
• **หมายเลข 21** คือ ส่วนของวงจร I/O พื้นฐาน สำหรับใช้ทดสอบการทำงานของ Function ต่างๆของ CPU (มีติดตั้งไว้เฉพาะในรุ่น CP-JR ARM7 USB-LPC2148 EXP) โดยมีรายละเอียดดังนี้คือ

- LED สำหรับแสดงผลการทำงานของ Output แบบ Sink Current มีทั้งหมด 4 ชุด
- Push Button Switch สำหรับกำเนิด Logic เพื่อทดสอบการทำงาน Input มีทั้งหมด 4 ชุด
- Volume สำหรับปรับค่าแรงดัน 0-3.3V เพื่อใช้ทดสอบการทำงานของ A/D มีทั้งหมด 4 ชุด
- Mini Speaker สำหรับใช้กำเนิดเสียง เช่น Beep จำนวน 1 ชุด จุดต่อแหล่งจ่ายไฟ +3.3V และ GND

2.4.3.2 วงจรแหล่งจ่ายไฟ

วงจรแหล่งจ่ายไฟสามารถใช้งานได้กับไฟ DC ขนาด +5V ได้ทันที โดยวงจรภาคแหล่งจ่ายไฟในส่วนที่เป็นวงจร Regulate ขนาด 3.3V นั้นจะจ่ายให้กับ CPU และวงจร I/O ของบอร์ดทั้งหมด ยกเว้น LCD แบบ Character ซึ่งจะใช้แหล่งจ่ายไฟขนาด +5VDC จากขั้ว Terminal ขนาด 2 Pin ที่จ่ายให้กับบอร์ดในกรณีที่ไม่ได้ใช้งาน Character LCD ผู้ใช้สามารถใช้แหล่งจ่ายจาก USB Port เพียงอย่างเดียวได้แต่ต้องไม่นำอุปกรณ์อื่นๆภายนอกมาเชื่อมต่อกับบอร์ด โดย USB Port สามารถจ่ายพลังงานให้กับบอร์ดได้ไม่เกิน 100mA ถ้ายังไม่ทำการ Enumerate USB Function และสามารถจ่ายกระแสให้บอร์ดได้สูงสุดไม่เกิน 500mA หลังจากทำการ Enumerate USB Function เรียบร้อยแล้ว

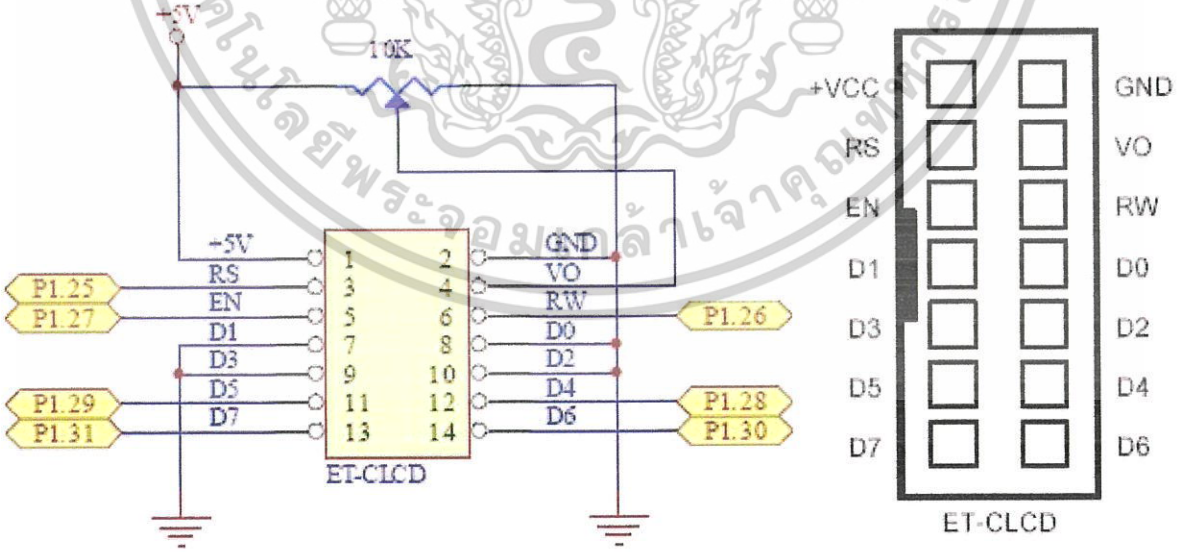
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงการเชื่อมต่อ Character LCD

2.4.4 การเชื่อมต่อกับ Character LCD

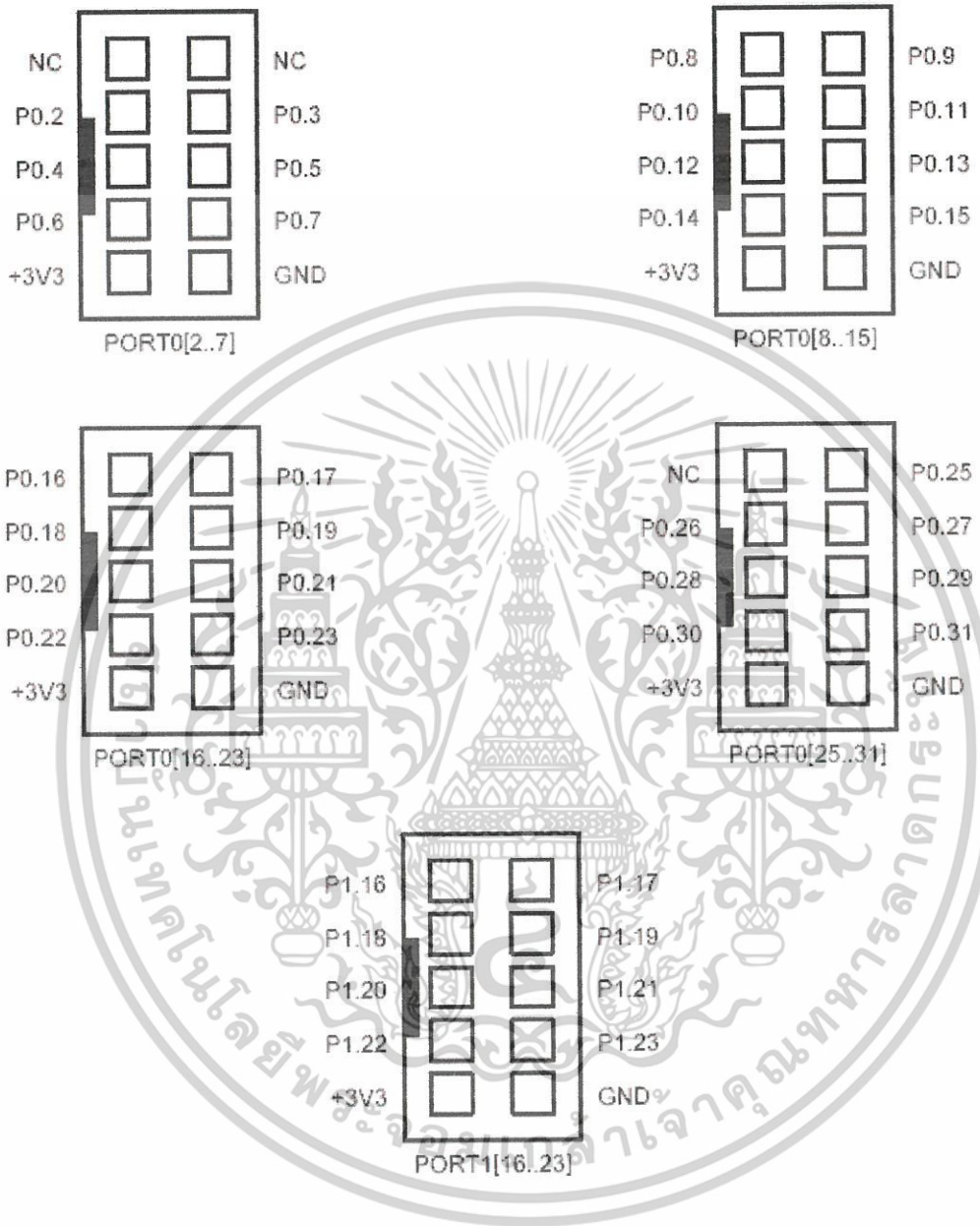
สำหรับการเชื่อมต่อ LCD นั้นจะสามารถใช้ได้กับ LCD แบบ Character LCD เท่านั้น โดยเชื่อมต่อแบบ 4 บิต โดยสัญญาณที่ใช้เชื่อมต่อกับ LCD จะเป็นสัญญาณจาก GPIO1.25 ถึง GPIO1.31 จำนวน 7บิต ซึ่งจะมีสัญญาณบางบิตถูกต่อไปยังขั้วต่อ JTAG ด้วย โดยในการเชื่อมต่อสายสัญญาณจากขั้วต่อของพอร์ต LCD ไปยังจอแสดงผล LCD นั้น ให้ยึดชื่อสัญญาณเป็นจุดอ้างอิง โดยให้ต่อสัญญาณที่มีชื่อตรงกันเข้าด้วยกันให้ครบทั้ง 14 เส้น ดังรูป



รูปที่ 2.11 ขั้วต่อ Port I/O ต่างๆของบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับขั้วต่อ Port I/O ของ CPU นั้น จะจัดเรียงออกมารอไว้ยังขั้วต่อแบบ IDE 10 Pin จำนวน 5 ชุด ชุดละ 8 บิต ดังนี้

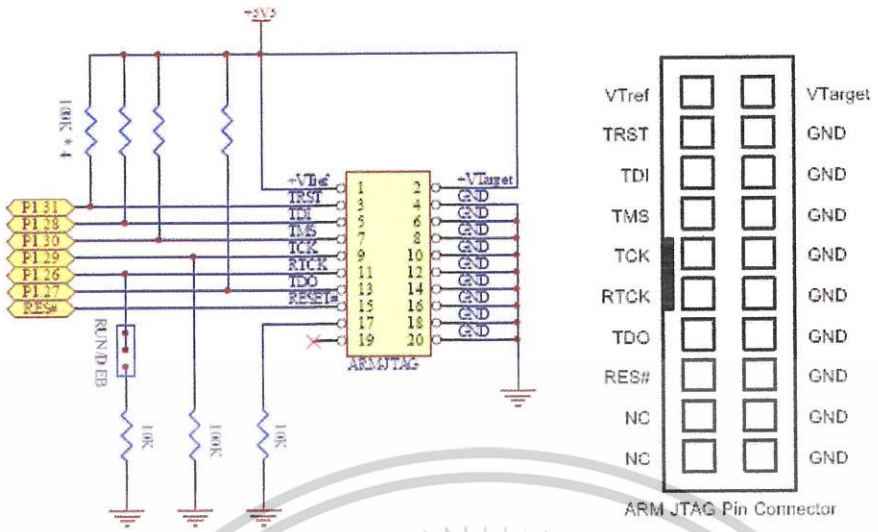


รูปที่ 2.12 ขั้วต่อแบบ IDE 10 Pin จำนวน 5 ชุด ชุดละ 8 บิต

2.4.5 ขั้วต่อ JTAG ARM

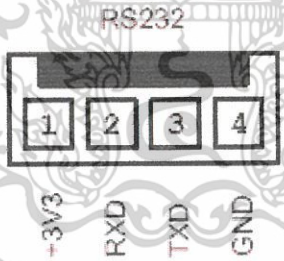
JTAG หรือ JTAG ARM จะเป็น Connector แบบ IDE 20 Pin สำหรับ Interface กับ JTAG Debugger โดยมีการจัดวงจรและสัญญาณตามมาตรฐานของ JTAG ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 ขั้วต่อ RS232

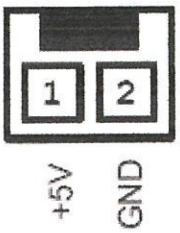
พอร์ต RS232 เป็นสัญญาณ RS232 ซึ่งผ่านวงจรแปลงระดับสัญญาณ MAX232 เรียบร้อยแล้วสามารถใช้เชื่อมต่อกับสัญญาณ RS232 เพื่อรับส่งข้อมูล นอกจากนี้แล้วยังสามารถใช้งาน ร่วมกับ Switch LOAD และ Switch RESET เพื่อทำการ Download Hex File ให้กับ CPU ได้ด้วย



รูปที่ 2.14 ขั้วต่อ Power Input (+5V)

เป็นจุดต่อไฟเลี้ยงบอร์ด +5V และ GND ซึ่งใช้สำหรับเป็นแหล่งจ่ายไฟให้กับบอร์ด โดยต้องป้อนเป็นไฟกระแสตรง DC ขนาด +5V เท่านั้น โดยมีการจัดขั้วดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 ภาพแสดงการจัดขั้ว

2.4.6 การ Download Hex file ให้กับ MCU ของบอร์ด

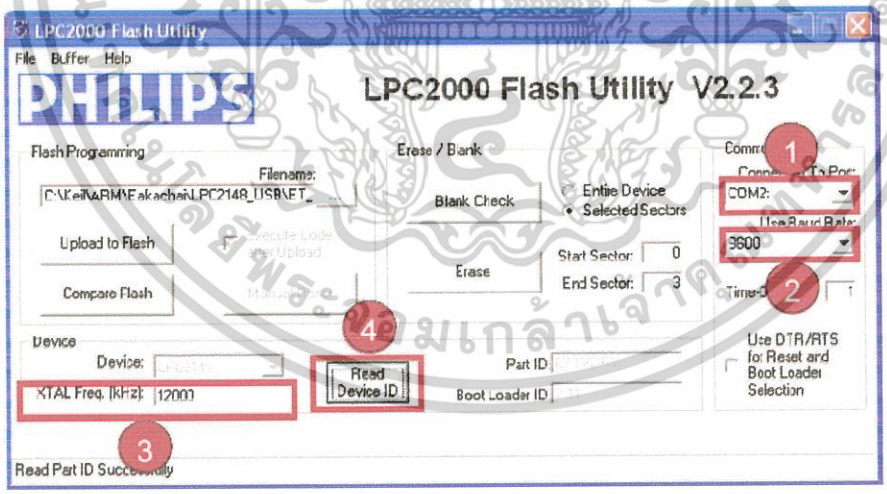
การ Download Hex File ให้กับหน่วยความจำ Flash ของ MCU ในบอร์ดนั้น จะใช้โปรแกรมชื่อ LPC2000 Flash Utility ของ Philips ซึ่งจะติดต่อกับ MCU ผ่าน Serial Port ของคอมพิวเตอร์ PC

ขั้นตอนการ Download HEX File ให้กับ MCU

1. ต่อสายสัญญาณ RS232 ระหว่างพอร์ตสื่อสารอนุกรม RS232 ของ PC และบอร์ด (RS232)

2. จ่ายไฟเลี้ยงวงจร +5VDC ให้กับบอร์ด ทางขั้วต่อ CPA ขนาด 2 Pin (หรือใช้การผ่าน USB Port ได้ถ้า USB Hub สามารถจ่ายพลังงานได้) ซึ่งจะสังเกตเห็น LED สีแดง (PWR) ติดสว่างให้เห็น

3. สั่ง Run โปรแกรม LPC2000 Flash Utility ของ Philips ซึ่งจะได้ผลดังรูป



รูปที่ 2.16 ภาพแสดงการสั่ง Run โปรแกรม

4. เริ่มต้นกำหนดค่าตัวเลือกต่างๆให้กับโปรแกรมตามต้องการ ซึ่งในกรณีนี้ใช้กับ LPC2148 ของบอร์ด CP-JR ARM7 USB-LPC2148 ของ อีทีที ให้เลือกกำหนดค่าต่างๆให้โปรแกรมดังนี้

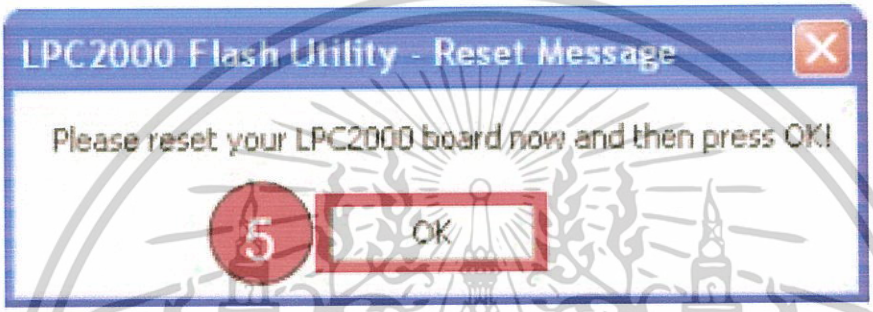
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) เลือก COM Port ให้ตรงกับหมายเลข COM Port ที่ใช้งานจริง (ในตัวอย่างใช้ COM1)

2) ตั้งค่า Baud Rate อยู่ที่ระหว่าง 4800 - 38400 ซึ่งเป็นค่าที่ทดสอบแล้วใช้ได้โดยไม่เกิดปัญหา หรือใช้ค่าความเร็วมาตรฐานคือ 9600

3) กำหนดค่าคริสตอล ออสซิลเลเตอร์ ให้ตรงกับที่ใช้ในจริงภายในบอร์ด โดยกำหนดให้มีหน่วยเป็น KHz และห้ามใส่ค่าเกิน 5 หลัก ในที่นี้ใช้ค่า 12.000MHz ซึ่งเท่ากับ 12000

4) คลิกเมาส์ที่ปุ่มคำสั่ง Read Device ID เพื่อติดต่อกับ CPU ซึ่งจะมีข้อความขึ้นมาเตือนให้เข้าสู่ Boot Mode ดังแสดงในรูป



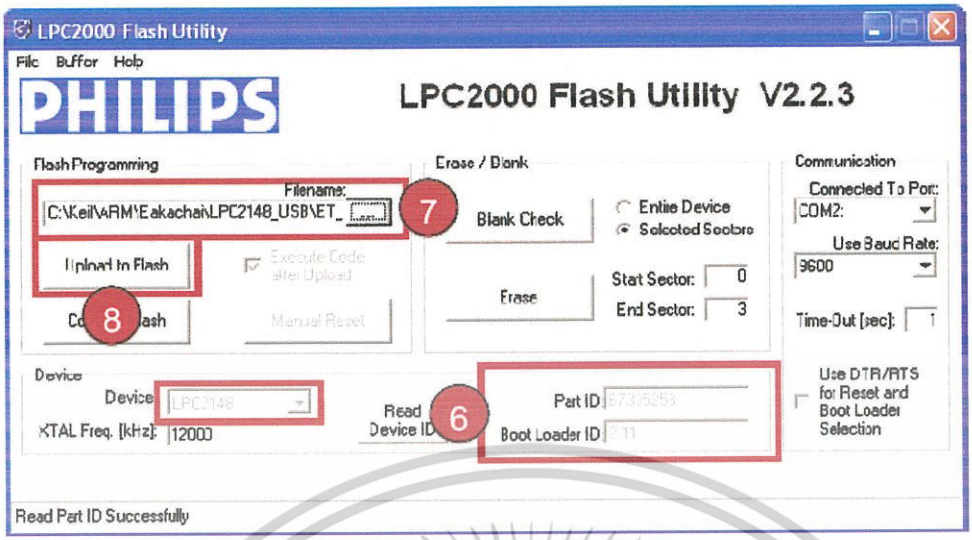
รูปที่ 2.17 ภาพแสดงข้อความเตือน

5) ให้กดสวิตช์ RESET และ LOAD ที่บอร์ด CP-JR ARM7 USB-LPC2148 เพื่อทำการ Reset ให้ MCU ทำงานใน Boot Loader ตามขั้นตอนดังต่อไปนี้

- กดสวิตช์ LOAD ค้างไว้
- กดสวิตช์ RESET โดยที่สวิตช์ LOAD ยังกดค้างอยู่
- ปล่อยสวิตช์ RESET โดยที่สวิตช์ LOAD ยังกดค้างอยู่
- ปล่อยสวิตช์ LOAD เป็นลำดับสุดท้าย เสร็จแล้วจึงคลิกเมาส์ที่ “OK”

6) เมื่อติดต่อกับ CPU ได้ จะปรากฏรายละเอียด Part ID และ Boot Loader ID ดังรูป

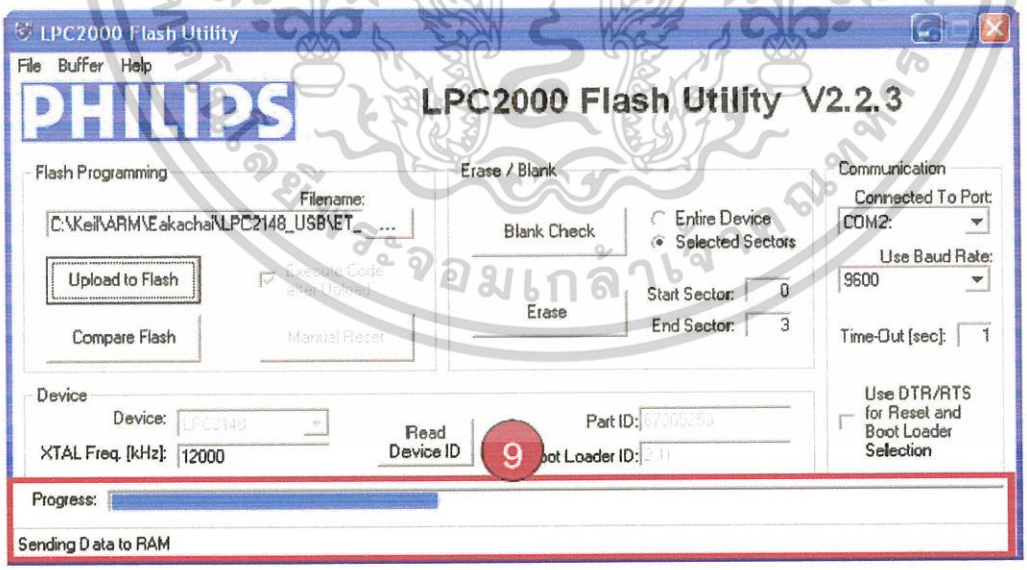
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 ภาพแสดงรายละเอียด Part ID และ Boot Loader ID

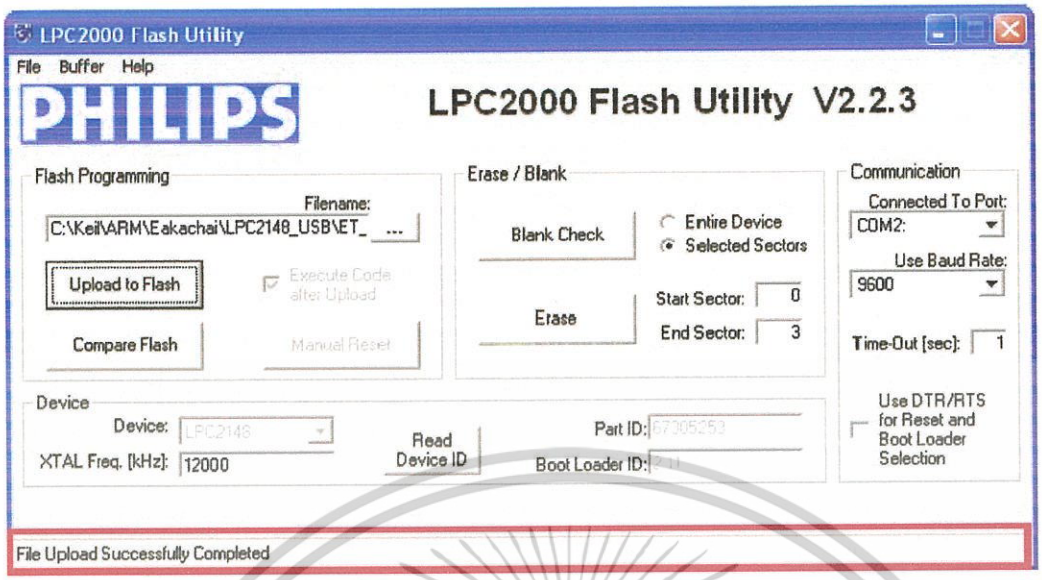
7) ให้ทำการเลือกกำหนด HEX File ที่จะทำการสั่งโปรแกรม

8) ให้ทำการคลิกเมาส์ที่ "Upload to Flash" ซึ่งโปรแกรม LPC2000 จะเริ่มต้นทำการ Download ข้อมูลให้กับ MCU ทันที โดยสังเกตที่ Status "Uploading to LPC2000 RAM and Copying to Flash Memory" ดังรูป โดยในขั้นตอนนี้ให้รอจนกว่าการทำงานของโปรแกรมจะเสร็จสมบูรณ์ ซึ่งให้สังเกตที่ Status "File Upload Successfully Completed"



รูปที่ 2.19 แสดง Status Progress

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 แสดง Status "File Upload Successfully Completed"

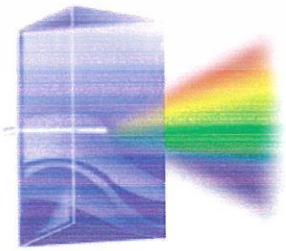
9) เมื่อการทำงานของโปรแกรมเสร็จเรียบร้อยแล้ว ให้กดสวิตช์ Reset ที่บอร์ด ซึ่ง MCU จะเริ่มต้นทำงานตามโปรแกรมที่สั่ง Download ให้ทันที

2.5 เลเซอร์ (Laser)

ลักษณะเด่นของแสเลเซอร์ที่แตกต่างจากแสงทั่วไป มีดังต่อไปนี้

เป็นแสงสีเดียว (Monochromaticity)

เป็นสมบัติสำคัญประการหนึ่งของเลเซอร์ ซึ่งหมายถึงแสงเลเซอร์มีความยาวคลื่นเพียงค่าเดียว ก่อนอื่นเราจะพิจารณาแสงที่ปล่อยจากต้นกำเนิดแสง โดยทั่วไปที่เราพบเห็นในชีวิตประจำวัน เช่น หลอดไฟฟ้า และดวงอาทิตย์ แสงจากหลอดไฟฟ้าและดวงอาทิตย์จะเป็นแสงสีขาว ถ้าให้แสงสีขาวนี้ผ่านปริซึม จะเห็นแถบสีต่างๆ เรียงกันอย่างต่อเนื่องจากสีม่วงถึงสีแดง เรียกว่า แถบสเปกตรัมของแสง



รูปที่ 2.21 ภาพแสดงแถบสีต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีของแสงเลเซอร์ เช่นเลเซอร์ฮีเลียม-นีออน เมื่อให้แสงสีแดงของเลเซอร์ฮีเลียม-นีออนผ่านปริซึม จะไม่มีการแยกเป็นหลายเส้น แต่ยังคงมีเพียง 1 เส้นที่มีความยาวคลื่น 632.8 นาโนเมตร

1. **มีความพร้อมเพรียงของคลื่น (Coherence)**

ต้นกำเนิดแสงธรรมดาโดยทั่วไป เช่น หลอดไฟฟ้า และหลอดฟลูออเรสเซนต์ จะให้แสงที่มีหลายสี หรือหลายความยาวคลื่น ยิ่งไปกว่านั้นแต่ละคลื่นยังเคลื่อนที่ต่างๆ กันไม่แน่นอนดังรูปด้านล่าง

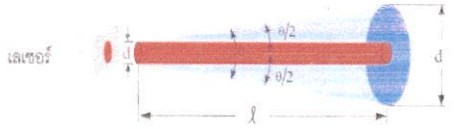
รูปที่ 2.22 ภาพแสดงการปล่งแสงจากหลอดไฟ

เราสามารถอธิบายการปล่อยแสงจากหลอดไฟฟ้าได้ดังนี้ ไส้หลอดไฟฟ้าที่เปล่งแสงประกอบด้วยอะตอมที่เล็กจำนวนมาก โดยแต่ละอะตอมจะทำหน้าที่เป็นต้นกำเนิดแสง ดังนั้นแต่ละอะตอมก็ปล่อยแสงออกมาอย่างอิสระซึ่งกันและกัน แสงที่ถูกปล่อยออกมาจากหลอดไฟฟ้าจึงมีเฟสและความยาวคลื่นต่างๆ กัน ยิ่งกว่านั้นแต่ละคลื่นที่ถูกปล่อยออกมามีทิศทางไม่แน่นอน หรือเป็น Random แสงจากแหล่งต้นกำเนิดแสงธรรมดาโดยทั่วไปจะเรียกว่า แสงอินโคฮีเรนต์ (incoherence light)

อย่างไรก็ตาม ต้นกำเนิดของแสงเลเซอร์นอกจากจะให้แสงสีเดียว ทุกๆ คลื่นของแสงเลเซอร์จะมีเฟสเดียวกันหมด ดังนั้นแสงเลเซอร์จึงเรียกว่า แสงโคฮีเรนต์ (coherence light)

2. **มีทิศทางที่แน่นอน (Directionality)**

สมบัติที่เด่นชัดของแสงเลเซอร์ คือลำแสงของเลเซอร์ไม่กระจายออกเหมือนแสงจากไฟฉาย หรือต้นกำเนิดแสงอื่นๆ แสงที่ออกจากเลเซอร์มีการบานออกค่อนข้างน้อย (divergence) กล่าวกันว่าทุกๆ ระยะทาง 1 เมตรที่ลำแสงเลเซอร์เคลื่อนที่ เส้นผ่านศูนย์กลางของลำแสงจะโตขึ้น 1 มิลลิเมตร



รูปที่ 2.23 ภาพแสดงทิศทางแสงที่แน่นอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

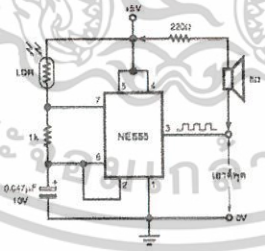
3. มีความเข้มหรือมีความสว่างสูงมาก (Brightness)

แสงเลเซอร์มีลักษณะเด่นไม่ซ้ำแหล่งกำเนิดแสงชนิดอื่นในเชิงความเข้มสูง และเมื่อลำแสงตกกระทบวัตถุ ก็เกิดความระยิบระยับของลำแสงขึ้น (Laser Spackle) โดยเฉพาะเมื่อวัตถุนั้นมีความหยาบ หรือแม้แต่ในบรรยากาศที่มีฝุ่นละอองหรือควันซึ่งเป็นอนุภาคแขวนลอยอยู่อย่าง random ทั้งนี้เนื่องจากแสงเลเซอร์เกิดการสะท้อนแบบไม่มีทิศทางกับอนุภาค หรือผิวหยาบของวัตถุ และเกิดการแทรกสอดของลำแสง ทำให้เกิดความระยิบระยับขึ้น จึงเป็นมิติของการมองเห็นโดยใช้ Laser displays

แสงเลเซอร์กำลังต่ำๆ เช่นเลเซอร์ฮีเลียม-นีออน ขนาด 1 mW ก็มีความเข้มแสงสูงกว่าแสงพระอาทิตย์ ฉะนั้นถ้าฉายเข้าตามนุษย์โดยตรงแล้ว จะเป็นอันตรายต่อนัยน์ตาถึงตาบอดได้

2.6 LDR ขนาด 10มิลลิเมตร

รูปที่ 2.24 โครงสร้างของ LDR



รูปที่ 2.25 ตัวอย่างวงจรเปลี่ยนสัญญาณแสงเป็นสัญญาณ

นอกจากวงจรเครื่องวัดแสง ซึ่งเป็นที่รู้จักกันดีในการประยุกต์ LDR ให้ใช้งานแบบทุกช่วงการเปลี่ยนแปลงแล้ว ยังสามารถดัดแปลงไปใช้ในวงจรอื่นๆ อีก เช่น วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล เพื่อเชื่อมต่อส่วนที่เป็น วงจรอนาล็อก ให้ส่งสัญญาณผ่านเข้าไปทำงานในวงจรดิจิตอลได้ ดังเช่น รูปที่ 9 เป็นวงจรแปลงระดับความเข้มแสง ซึ่งเป็นสัญญาณ อนาล็อกให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกมาเป็นจำนวนลูกคลื่นสี่เหลี่ยม ยิ่งความเข้มแสงมากเท่าไหร่ จำนวนลูกคลื่น สี่เหลี่ยมก็จะยิ่งออกมามากเท่านั้น วงจรนี้ ใช้ไอซี 555 ความถี่ของคลื่นที่ออกมาจะได้ประมาณ 22kHz ถ้าเอาไปรับแสงใกล้ๆ หลอดไฟขนาด 60 วัตต์ แต่จะ เหลือเพียงประมาณ 1Hz ในที่มืด ถ้าเอาลำโพงอนุกรมกับตัวต้านทาน 220 Ω ไปต่อเข้ากับขา 3 และ ไฟบวกก็จะได้ยินเสียง สูงๆต่ำๆ ตามความเข้มของแสง

2.7 VR ปรับค่าแบบเกือกม้า

ตัวต้านทานปรับค่าได้

รูปที่ 2.26 ภาพแสดงตัวต้านทานปรับค่าได้

ส่วนในตัวต้านทานปรับค่าได้จะมีอยู่หลายประเภท เช่น แบบหมุน ซึ่งมีทั้งแบบธรรมดาที่หมุนได้รอบเดียวที่ใช้กันทั่วไป และแบบ พิเศษที่หมุนได้หลายรอบ (Trimmer Potentiometer) และแบบสไลด์ ซึ่งมีรูปร่างและโครงสร้างของบาง เพราะว่า เมื่อมีกระแส IC ไหลปผ่าน R4 ไปยังขา C ของ Q1 จะเกิดแรงดันตกคร่อม R4 ทำให้แรงดันจากแบตเตอรี่ไปยังขา C ลดลง จะคงเหลือ 4.7 โวลต์ที่นี้เราจะรู้ค่า R4 ได้จากกฎของโอห์ม

$$R = E / I = VR4 / I \qquad (2.2)$$

เมื่อ E คือ แรงดันที่ตกคร่อมตัวต้านทาน R4 เราจะรู้ได้โดยถ้าเราต้องการให้แรงดันที่จุด 2 เท่ากับ 4.7 โวลต์ เพราะฉะนั้นแรงดันที่ตก R4 (VR4) ; VR4 = 9 - 4.7 = 4.3 โวลต์ แล้วค่ากระแส (I) ล่ะ ในที่นี้เราจะมีค่าสมมติให้ I = Ic = 1มิลลิแอมป์ (0.001 แอมป์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

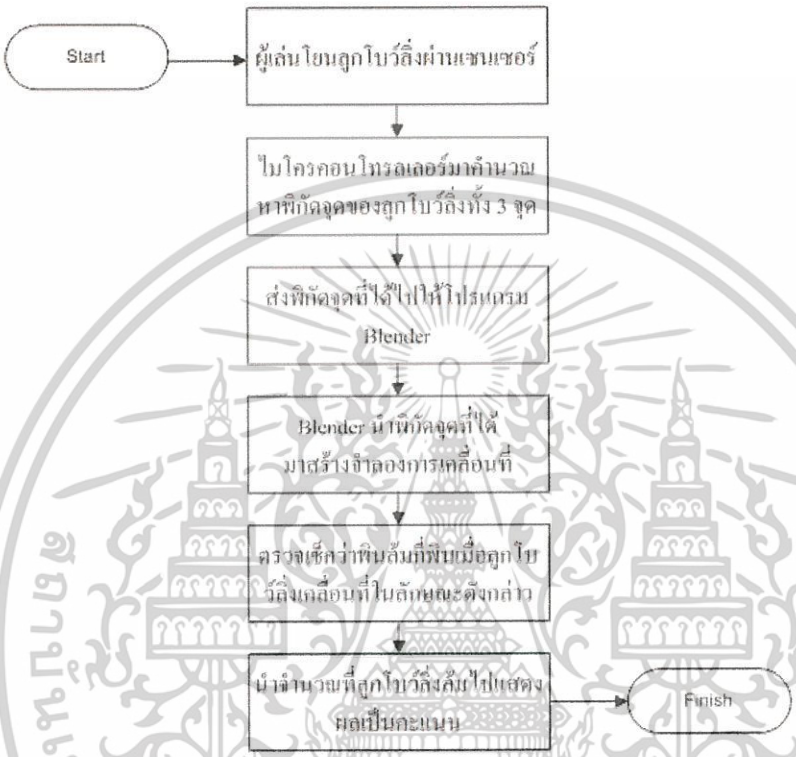
เพราะฉะนั้นเราจะคำนวณ R4 ได้ R = VR4 / I = 4.3 / 0.001 = 4300 โอห์ม = 4.3 กิโลโอห์ม R1 , R2 จะทำหน้าที่เป็นตัวจัดแรงดัน เพื่อให้ทรานซิสเตอร์ทำงานในสภาวะที่เหมาะสม และถ้าคุณใช้มิเตอร์วัดแรงดันที่ตกคร่อม R4 (VR4) ได้ คุณก็สามารถคำนวณกระแส IC ได้เช่นกัน เพราะกระแสที่ไหลผ่าน R 4 มีค่า เท่ากับ I C โดยใช้กฎโอห์ม

$$I = E / R \qquad (2.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 ภาพรวมของระบบ



รูปที่ 3.1 ภาพโดยรวมของระบบ

จากรูป 3.1 เมื่อผู้เล่นทำการโยนลูกโบว์ลิ่งออกไปบนเลนโบว์ลิ่งระบบจะมีการตรวจจับการเคลื่อนที่ของลูกโบว์ลิ่งนั้น โดยมีเซนเซอร์ในการตรวจจับ และจะนำมาหาพิกัดจุดที่ลูกโบว์ลิ่งผ่าน ซึ่งจะได้ออกมาทั้งหมด 3 จุดในระยะ 3 เมตรแรก ระบบก็จะส่งค่า 3 จุดนี้ผ่าน Serial-port ไปให้โปรแกรม Blender คำนวณหาการเคลื่อนที่ของลูกโบว์ลิ่งที่เหลือ จาก 3 พิกัดจุดใน 3 เมตรแรก โดยการกำหนดจุดการเคลื่อนที่ให้กับลูกโบว์ลิ่ง เมื่อได้พิกัดจุดที่ลูกโบว์ลิ่งจะวิ่งแล้ว ก็จะให้ลูกโบว์ลิ่งไปกระทบพิน โปรแกรม Blender ก็จะใช้ในส่วนของ Game Engine คำนวณทางฟิสิกส์ออกมาว่า ถ้าลูกโบว์ลิ่งชนพินในลักษณะดังกล่าวพินจะล้มกี่พิน เมื่อลูกโบว์ลิ่งชนพินแล้วก็จะมาตรวจเช็คว่าพินล้มกี่พิน ก็จะนำจำนวณที่พินล้มนั้นส่งแสดงผลออกมาเป็นแต้มที่ได้ในการโยนในครั้งนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ผังงาน (Flowchart)

3.2.1 ผู้เล่นโยนลูกโบว์ลิ่งผ่าน Sensor

รูปที่ 3.2 แสดงขั้นตอนการโยนลูกโบว์ลิ่ง

จากรูป 3.2 เริ่มต้นโดยผู้เล่นโยนลูกโบว์ลิ่งไปบนเลนโบว์ลิ่งที่เตรียมไว้ยาว 3 เมตร แล้วลูกโบว์ลิ่งก็เคลื่อนที่ผ่านเซนเซอร์ที่ติดไว้บนเลนโบว์ลิ่ง (แนวแกน x 5 ตัวและแนวแกน y 3 ตัว)

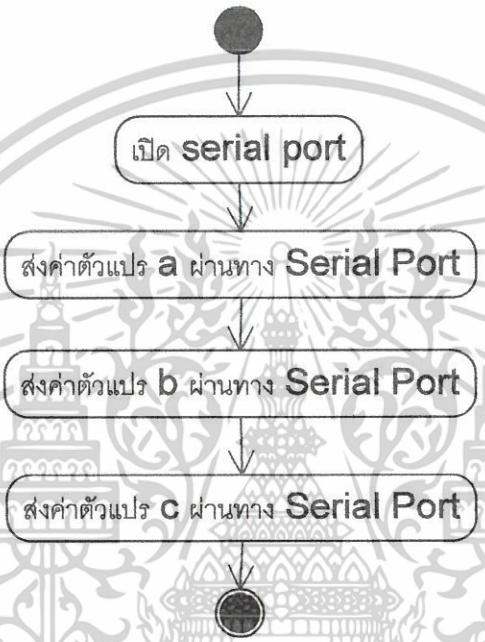
3.2.2 ไมโครคอนโทรลเลอร์มาคำนวณหาพิกัดจุด

รูปที่ 3.3 แสดงขั้นตอนการเคลื่อนที่ของลูกโบว์ลิ่งผ่านจุดเช็คทั้งสามจุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 3.3 เมื่อลูกโบว์ลิ่งผ่านจุดเช็คที่ 1 ให้นำค่าในแนวแกน y มาเก็บไว้ในตัวแปร value แล้วมาเช็คค่าว่าบิตไหนในตัวแปร value มีค่าเป็น 1 ก็จะแสดงว่าลูกโบว์ลิ่งผ่านเซ็นเซอร์ตัวนั้น เมื่อทราบแล้วว่าบิตไหน ก็ให้นำตำแหน่งบิตนั้นไปเก็บไว้ในตัวแปร a เช่น เมื่อทราบว่าลูกโบว์ลิ่งผ่านเซนเซอร์ตัวที่ 2 ก็จะกำหนดให้ a= 2 ซึ่งจะทำเช่นเดียวกันกับจุดเช็คที่ 2 และ 3 ใส่ตัวแปร b และ c ตามลำดับ ท้ายที่สุดก็จะได้ตัวแปร a, b และ c เก็บค่าของพิกัดจุด ณ ตำแหน่งจุดเช็คดังกล่าวทั้ง 3 จุด

3.2.3 ส่งพิกัดจุดที่ได้ให้โปรแกรม Blender

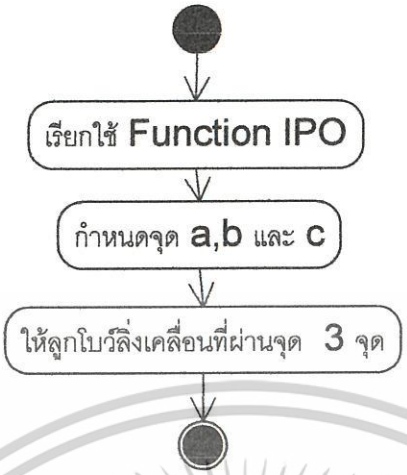


รูปที่ 3.4 แสดงขั้นตอนการส่งค่าตัวแปร

จากรูป 3.4 ไมโครคอนโทรลเลอร์จะทำการติดต่อกับคอมพิวเตอร์ผ่านทาง Serial port โดยใช้ Python Scripts ในการรับค่า โดยจะส่งค่าในตัวแปร a , b และ c ตามลำดับ โดยจะส่งผ่าน Serial port ไปยังโปรแกรม Blender

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

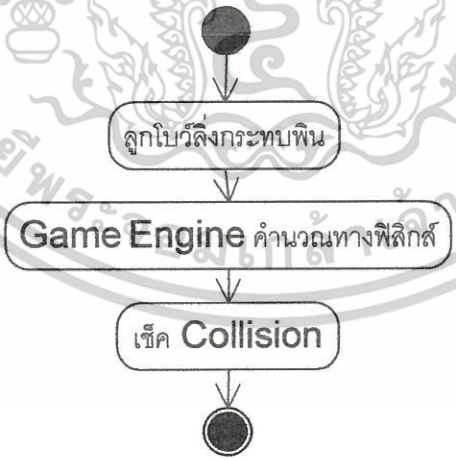
3.2.4 นำค่าพิกัดจุดที่ได้มาสร้างจำลองการเคลื่อนที่



รูปที่ 3.5 แสดงการนำค่าพิกัดจุดไปจำลองการเคลื่อนที่

จากรูป 3.5 นำค่าที่ได้ไปให้โปรแกรม Blender สร้างจำลองการเคลื่อนที่โดยใช้ Function IPO ซึ่งเป็นฟังก์ชั่นที่ไว้กำหนดแนวเส้นทางการเคลื่อนที่ของวัตถุ โดยจะต้องมีการกำหนดจุดไว้ที่จะให้ลูกโบว์ลิ่งวิ่งผ่าน ซึ่งจุดดังกล่าวจะได้จากพิกัดจุดที่ส่งมาจากไมโครคอนโทรลเลอร์ และเมื่อได้กำหนดจุดไว้แล้ว ลูกโบว์ลิ่งก็จะเคลื่อนที่ผ่านจุดดังกล่าว เป็นเส้นทางการเคลื่อนที่ลูกโบว์ลิ่ง

3.2.5 ตรวจเช็คว่าพินล้มกี่พิน



รูปที่ 3.6 แสดงการเช็คว่าพินล้มกี่พิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

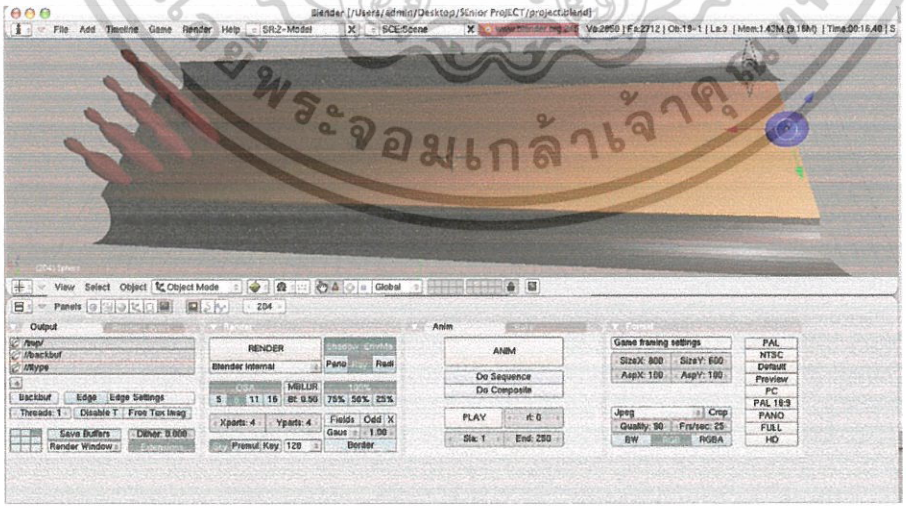
จากรูป 3.6 เมื่อสร้างจำลองขึ้นมาเป็น 3D แล้ว โปรแกรม Blender ในส่วนของ Game Engine จะมาคำนวณทางฟิสิกส์ว่าโมเดลที่เป็นลูกโบว์ลิ่ง กับโมเดลที่เป็นพินโบว์ลิ่ง เมื่อมากระทบกันในรูปแบบใดๆ แล้วพินจะล้มในรูปแบบใด แล้วมาเช็คว่าพินโดนกระทบบ้าง

3.2.6 นำจำนวนพินที่ล้มไปแสดงผล

รูปที่ 3.7 แสดงจำนวนพินที่ล้ม

จากรูปที่ 3.7 เมื่อเช็คได้ว่าพินล้มกี่พิน ก็จะนำมารวมกันใส่ในตัวแปร แล้วส่งค่าไปให้ Scene : Score แสดงผลออกมา

3.3 โครงสร้างและการออกแบบเลน



รูปที่ 3.8 เลนโบว์ลิ่งจากโปรแกรม Blender

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.8 เป็นเลนโบว์ลิ่งที่ได้สร้างจากโปรแกรม Blender จะเป็นส่วนการ simulation ต่อจากเลนโบว์ลิ่งจริงที่มีความยาว 3 เมตร เลนโบว์ลิ่งจากโปรแกรม Blender นี้จะ simulation ต่อไปอีก 15 เมตรจนกระทั่งลูกโบว์ลิ่งชนพิน

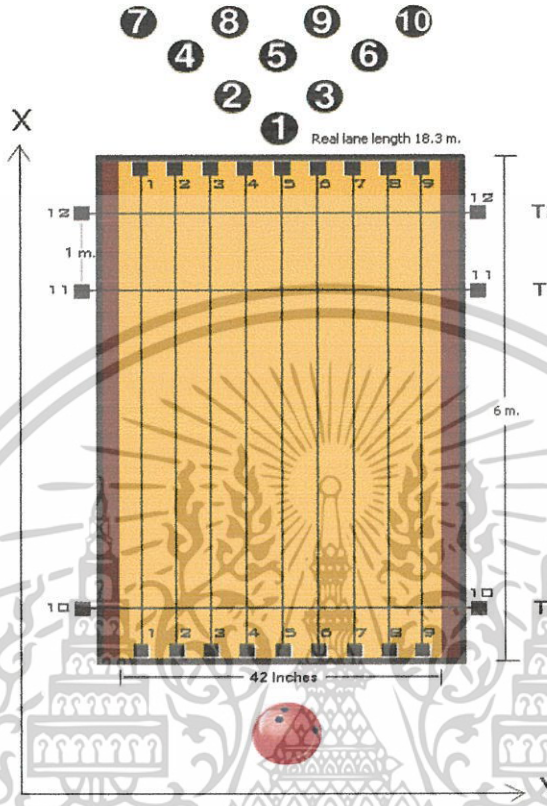
รูปที่ 3.9 เลนโบว์ลิ่งจริง

จากรูปที่ 3.9 เป็นเลนโบว์ลิ่งจริงที่ได้สร้างขึ้นมา ซึ่งจะยกสูงจากพื้น 20 ซ.ม. ความกว้างทั้งหมด มีขนาด 75 ซ.ม. โดยเลนมีความกว้าง 50 ซ.ม. และท่อสองข้างมีขนาดข้างละ 12.5 ซ.ม. เลนความยาวทั้งหมดมีขนาด 340 ซ.ม. โดยมีขนาดความยาวเลน 310 ซ.ม.และพื้นที่ก่อนถึงเลน 17.5 ซ.ม.และท่อด้านหลังอีก 12.5 ซ.ม.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การรับค่า และตรวจจับการเคลื่อนที่ของลูกโบว์ลิ่ง

จะใช้เซนเซอร์ทั้งหมด 8 ตัว แบ่งออกเป็น ตัวที่ 1-5 จะตรวจจับการเคลื่อนที่แนวแกน Y และ เซ็นเซอร์ตัวที่ 6 - 8 จะเป็นตัว Milestone จะคอยเช็คเมื่อวัตถุวิ่งผ่าน ซึ่งมีขั้นตอนดังนี้



รูปที่ 3.10 ลักษณะของเลนโบว์ลิ่ง

1. เมื่อผู้เล่นเริ่มโยนลูกโบว์ลิ่ง ลูกโบว์ลิ่งก็จะวิ่งผ่านตำแหน่งที่เป็น Milestone ตัวที่ 1,2 และ 3 ตามลำดับ ก็ให้นำค่าบิตจากแกน y ส่งไปคำนวณ
2. เช็คค่าในแนวแกน y ว่าบิตไหนมีค่าเป็น 1 ซึ่งนั่นแสดงว่าลูกโบว์ลิ่งผ่านเซนเซอร์ตัวนั้น ก็จะให้กำหนดค่าให้กับตัวแปร โดยตัวแปร a จะเก็บค่าของแนวแกน x ณ จุด Milestone ที่ 1 ส่วนตัวแปร b และ c ก็จะเก็บค่าในแนวแกน x ณ ตำแหน่ง Milestone จุดที่ 2 และ 3 ตามลำดับ
3. ณ ขั้นตอนนี้ ใน Microcontroller จะมีค่าตัวแปรทั้งหมด 3 ตัว คือ a, b และ c Microcontroller ก็จะส่งค่าของตัวแปรทั้ง 3 ตัวให้กับโปรแกรม Blender ในการประมวลผลต่อ โดยผ่านทาง Serial-port โดยผ่านทาง Python Scripts

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อโปรแกรม Blender ได้ค่าตัวแปร a, b และ c แล้วก็จะทราบว่าพิกัดนั้นเป็นพิกัดของลูกโบว์ลิ่งที่เคลื่อนที่ในระยะ 6 เมตรแรก ก็จะนำพิกัดจุดดังกล่าวไปกำหนดการเคลื่อนที่ของลูกโบว์ลิ่งใน 22 เมตรที่เหลือ ประมวลผลออกมาเป็น 3 มิติโดยใช้ IPO Curves เป็นตัวกำหนดพิกัดจุดการเคลื่อนที่ของลูกโบว์ลิ่ง
5. การใช้ IPO Curve มาช่วยในการปรับแต่งการเคลื่อนที่ของลูกโบว์ลิ่ง โดยใช้หน้าต่างการทำงานของ IPO Curve ซึ่งจะมีลักษณะเป็นเส้นกราฟหลายสี ซึ่งจะเป็นเส้นกราฟที่แสดงผลเส้นทางการเคลื่อนที่ของลูกโบว์ลิ่ง ซึ่งการเรียกใช้หน้าต่างการทำงานนี้โดยทำการ Split หน้าจอแล้วเลือก Window Type เป็น IPO Curve Editor ซึ่งการใช้งานจะต้องมีการกำหนดจุด Keyframe เพื่อเป็นจุดกำหนดการเคลื่อนที่และกำหนดว่า ณ จุด Keyframe นั้นวัตถุจะเคลื่อนที่อยู่ ณ ตำแหน่งใด และสำหรับระบบนี้เราจะทราบการเคลื่อนที่ของลูกโบว์ลิ่งทั้งหมดเพียงแค่ 3 จุด และต้องการทราบอีก 1 จุด จึงนำทั้ง 4 จุดนั้นมาสร้างเป็น Keyframe กำหนดการเคลื่อนที่ ซึ่ง ณ 3 จุดแรกที่เราทราบพิกัดจุดแล้ว เราจะต้องกำหนดจุด (x,y,z) ครบทั้ง 3 จุด ส่วนจุดที่ 4 จุดสุดท้ายนั้น เราจะทราบเพียงแค่ 2 แกนคือ x กับ z ส่วนแกน y เรายังไม่ทราบก็ไม่ต้องกำหนด แล้วค่อยให้โปรแกรมสร้างให้จากแนวการเคลื่อนที่ของลูกโบว์ลิ่งจาก 3 จุดแรก
6. แต่เนื่องจากการเคลื่อนที่ของลูกโบว์ลิ่งแต่ละครั้งที่ผู้เล่นทำการโยนจะไม่เหมือนกัน จึงต้องมีการสร้างจุด Keyframe โดยใช้ Python Script โดยใช้ ตัวแปรในการกำหนด Keyframe ซึ่งตัวแปรนี้จะได้มาจากการรับค่ามาจาก Microcontroller ซึ่งสร้างได้ดังนี้

```
# Create an IPO of type Object named 'Recorded IPO'.
        blenderObject = Blender.Object.Get( objectName )
        ipo = Blender.Ipo.New( 'Object', 'Recorded IPO' )
        blenderObject.setIpo( ipo )
# Set the curves for the X, Y, and Z axis and 0 is a number of keyframe
        GameLogic.rec[1].addBezier( ( 0, X ) )
        GameLogic.rec[1].update()
        GameLogic.rec[2].addBezier( ( 0, Y ) )
        GameLogic.rec[2].update()
        GameLogic.rec[3].addBezier( ( 0, Z ) )
        GameLogic.rec[3].update()
```

7. ทำการแสดงผลการเคลื่อนที่ออกมาในรูปของ 3 มิติ และใช้ Game Engine มาคำนวณทางฟิสิกส์ว่าถ้าลูกโบว์ลิ่งมากระทบพินในลักษณะดังกล่าวพินจะล้มกี่พิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การพัฒนาระบบ

4.1 การพัฒนาโปรแกรมโดยใช้ไมโครคอนโทรลเลอร์

4.1.1 ไมโครคอนโทรลเลอร์ (ARM 7)

ในระบบโบว์ลิ่งเสมือนจริงในส่วนของไมโครคอนโทรลเลอร์ (ARM 7) มีโค้ด ดังนี้

```
#include <LPC214x.h>
#include <stdio.h>
#define CR 0x0D

long int mile1, mile2, mile3;
unsigned long int value;
unsigned int a=1;
unsigned int b=1;
unsigned int c=1;

void init (void)
{
        PLL0CFG &= 0xE0;
        PLL0CFG |= 0x04;
        PLL0CFG &= 0x9F;
        PLL0CFG |= 0x20;

        PLL0CON &= 0xFC;

        PLL0CON = 0x01;
        PLL0FEED = 0xAA;
        PLL0FEED = 0x55;
        while (!(PLL0STAT & 0x00000400));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        PLL0CON |= 0x02;
        PLL0FEED = 0xAA;
        PLL0FEED = 0x55;

        VPBDIV &= 0xFC;
        VPBDIV |= 0x02;

        MAMCR = 0x00;
        MAMTIM = 0x03;
        MAMCR = 0x02;
}
void uart0_init(unsigned int baudrate)
{
        unsigned short u0dl;
        u0dl = 30000000/(16*baudrate);
        PINSEL0 |= 0x00000005;
        U0LCR = 0x83;
        U0DLL = u0dl & 0xFF;
        U0DLM = (u0dl>>8);
        U0LCR &= 0x7F;
}
int putchar (int ch)
{
        if (ch == '\n')
        {
                while (!(U0LSR & 0x20));
                U0THR = CR;
        }
        while (!(U0LSR & 0x20));
        return (U0THR =ch);
}
int getchar (void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        while (!(U0LSR & 0x01));
        return (U0RBR);
}

void isr_uart0 (void) __irq
{
        char msg;
        if(((msg = U0IIR)&0x01) == 0)
        {
                switch(msg & 0x0E)
                {
                        case 0x04 : while (!(U0LSR & 0x20));
                                U0THR = U0RBR;
                                break;
                        case 0x02: break;
                        default: break;
                }
        }
        VICVectAddr = 0;
}

int main (void)
{
        init();
        uart0_init(9600);
        U0IER = 3;
        VICVectAddr0 = (unsigned)isr_uart0;
        VICVectCntl0 = 0x20 | 6;
        VICIntEnable = 1<<6;

        PINSEL0 = 0x00000005;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PINSEL1 = 0x00000000;
IODIR0 = 0x00000000;

while(1)
{
        while(a){
                if((IOPIN0 & 0x00000200)==0)    // If ball is pass milestone 1
                {
                        if((IOPIN0 & 0x00600000)==0){// Between 5 to 4
                                printf("8\n");
                        }else if((IOPIN0 & 0x00300000)==0){// Between 4 to 3
                                printf("6\n") ;
                        }else if((IOPIN0 & 0x00180000)==0){// Between 3 to 2
                                printf("4\n") ;
                        }else if((IOPIN0 & 0x000C0000)==0){// Between 2 to 1
                                printf("2\n") ;
                        }else if((IOPIN0 & 0x00040000)==0){// Sensor 1
                                printf("1\n");
                        }else if((IOPIN0 & 0x00080000)==0){// Between 2
                                printf("3\n") ;
                        }else if((IOPIN0 & 0x00100000)==0){// Between 3
                                printf("5\n") ;
                        }else if((IOPIN0 & 0x00200000)==0){ // Between 4
                                printf("7\n") ;
                        }else if((IOPIN0 & 0x00400000)==0){ // Between 5
                                printf("9\n") ;
                        }else{
                                printf("10");
                        }a=0;}
        }
        while(b){
                if((IOPIN0 & 0x00000400)==0)  // If ball is pass milestone 2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        if((IOPIN0 & 0x00600000)==0){// Between 5 to 4
            printf("8\n") ;
        }else if((IOPIN0 & 0x00300000)==0){// Between 4 to 3
            printf("6\n") ;
        }else if((IOPIN0 & 0x00180000)==0){// Between 3 to 2
            printf("4\n") ;
        }else if((IOPIN0 & 0x000C0000)==0){// Between 2 to 1
            printf("2\n") ;
        }else if((IOPIN0 & 0x00040000)==0){// Sensor  1
            printf("1\n");
        }else if((IOPIN0 & 0x00080000)==0){// Between 2
            printf("3\n") ;
        }else if((IOPIN0 & 0x00100000)==0){// Between 3
            printf("5\n") ;
        }else if((IOPIN0 & 0x00200000)==0){// Between 4
            printf("7\n") ;
        }else if((IOPIN0 & 0x00400000)==0){// Between 5
            printf("9\n") ;
        }else{
            printf("10");
        } b=0;
    }
}
while(c){
    if((IOPIN0 & 0x00000800)==0) // If ball is pass milestone 3
    {
        if((IOPIN0 & 0x00600000)==0){// Between 5 to 4
            printf("8\n") ;
        }else if((IOPIN0 & 0x00300000)==0){// Between 4 to 3
            printf("6\n") ;
        }else if((IOPIN0 & 0x00180000)==0){// Between 3 to 2
```

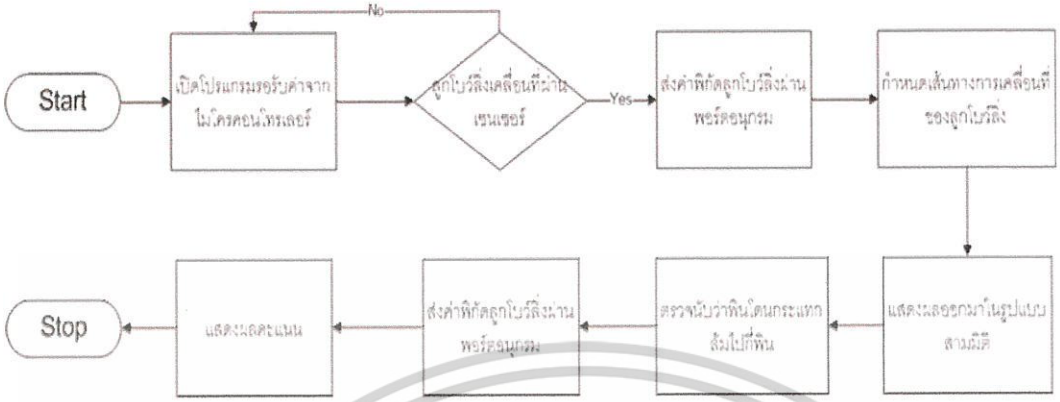
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                printf("4\n") ;
            }else if((IOPIN0 & 0x000C0000)==0){// Between 2 to 1
                printf("2\n") ;
            }else if((IOPIN0 & 0x00040000)==0){// Sensor  1
                printf("1\n");
            }else if((IOPIN0 & 0x00080000)==0){// Between 2
                printf("3\n") ;
            }else if((IOPIN0 & 0x00100000)==0){// Between 3
                printf("5\n") ;
            }else if((IOPIN0 & 0x00200000)==0){// Between 4
                printf("7\n") ;
            }else if((IOPIN0 & 0x00400000)==0){// Between 5
                printf("9\n") ;
            }else{
                printf("10");
            } c=0;
        }
    } a=1,  b=1, c=1;
    }
}
```

จาก โค้ดดังกล่าว function init() จะเป็นฟังก์ชั่นที่ใช้ในการกำหนดค่าเริ่มต้นในการส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์และคอมพิวเตอร์ผ่านพอร์ตอนุกรม หลังจากกำหนดค่าเริ่มต้นสำหรับการส่งข้อมูลเรียบร้อยแล้ว ก็จะเข้าสู่ function main() ซึ่งเป็นฟังก์ชั่นของการทำงานหลัก เริ่มจากกำหนดพอร์ต 0 ให้เป็นพอร์ตอเนกประสงค์เพื่อทำหน้าที่ในการรับอินพุทและส่งเอาท์พุท หลังจากนั้นจะต้องมากำหนดว่าพินใดในพอร์ต 0 ทำหน้าที่เป็นอินพุท ซึ่งจากโค๊ดดังกล่าวกำหนดให้ทุกพินทำหน้าที่เป็นอินพุท แต่พินที่มีการรับค่าจริงๆ เพียงแค่พินที่ 9-11 และพินที่ 18-22 โดยพินที่ 9-11 จะทำหน้าที่เป็นไมล์สโตนคือทำการเช็คว่าลูกโบว์ลิ่งวิ่งผ่านมาหรือไม่ และพินที่ 18-22 จะเป็นตัวบอกว่าลูกโบว์ลิ่งผ่าน ณ พิกัดไหน ซึ่งจะทำการเช็คค่าของพินดังกล่าวโดยจะทำการวนลูปเช็คค่าตลอดเวลา เมื่อไหร่ที่ลูกโบว์ลิ่งผ่านจุดเช็คตัวที่ 1 นั่นคือพินที่ 9 จะมีค่าเป็น 0 ก็จะมาดูว่าลูกโบว์ลิ่งผ่านพิกัดจุดไหน คือจะดูว่าพินที่ 18-22 พินไหนที่มีค่าเป็น 0 ก็ให้ส่งค่าตัวเลขของพินนั้นผ่านพอร์ตอนุกรม และลูกโบว์ลิ่งก็จะเคลื่อนที่ผ่านจุดเช็คตัวที่ 2 และ 3 ตามลำดับ ก็จะเช็คเช่นเดียวกัน ก็จะเป็นเช่นนี้ไปตลอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การพัฒนาโปรแกรมโดยใช้ เบลนเดอร์ (Blender)

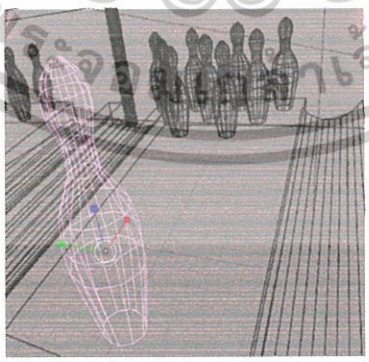


รูปที่ 4.1 รูปขั้นตอนการทำงานของโปรแกรมเบลนเดอร์

จากรูป 4.1 เป็นรูปขั้นตอนการทำงานของโปรแกรมเบลนเดอร์เริ่มต้นการทำงานของโปรแกรมโดยการเริ่มเปิดโปรแกรมที่ทำการสร้างโมเดลไว้เสร็จแล้ว เพื่อรอทำการรับค่าจากไมโครคอนโทรลเลอร์ เมื่อลูกโบว์ลิ่งเคลื่อนที่ผ่านเซนเซอร์ไมโครคอนโทรลเลอร์ก็จะส่งค่าพิกัดของลูกโบว์ลิ่งผ่านพอร์ตอนุกรม หลังจากโปรแกรมเบลนเดอร์ได้รับค่าจากไมโครคอนโทรลเลอร์แล้วก็จะนำมาทำการกำหนดเส้นทางการเคลื่อนที่ของลูกโบว์ลิ่ง แล้วนำมาแสดงผลในรูปแบบ 3 มิติซึ่งการแสดงผลดังกล่าว ก็จะมาทำการตรวจเช็คว่าพินไหนที่โดนกระแทกล้มบ้าง แล้วรวมรวมเป็นคะแนนแสดงผลออกมาทางหน้าจอ

4.2.1 การสร้างโมเดล

4.2.1.1 การสร้างโมเดลพินโบว์ลิ่ง

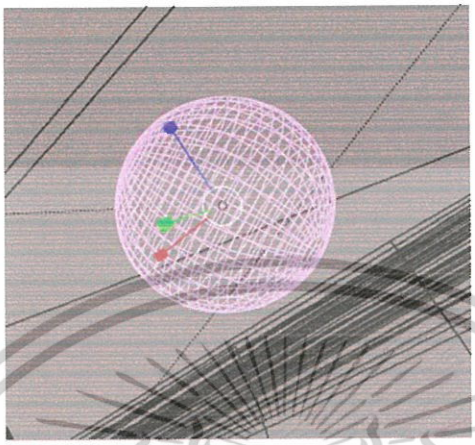


รูปที่ 4.2 รูปโมเดลพิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 4.2 เป็นรูปที่แสดงการสร้างพินโบว์ลิ่งโดยใช้ Curve >> NURBS Curve ซึ่งรายละเอียดของ Curve >> NURBS Curve ในการสร้างพินจะแสดงอยู่ในบทที่ 2 หน้า 4

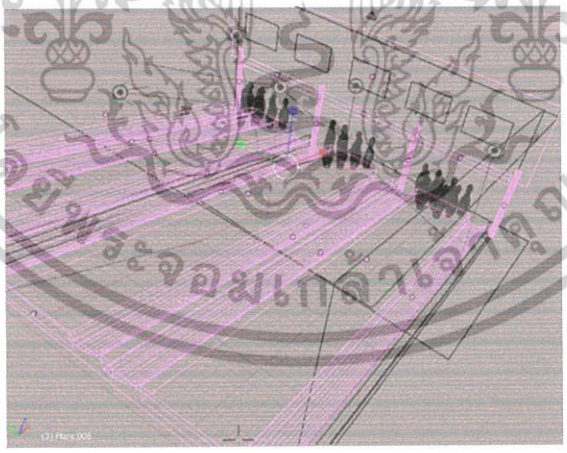
4.2.1.2 การสร้างโมเดลลูกโบว์ลิ่ง



รูปที่ 4.3 รูปโมเดลลูกโบว์ลิ่ง

จากรูป 4.3 เป็นรูปที่แสดงกาสร้างลูกโบว์ลิ่ง โดยใช้ Mesh >> UVSphere ซึ่งรายละเอียดของ Mesh >> UVSphere ในการสร้างลูกโบว์ลิ่งจะแสดงอยู่ในบทที่ 2 หน้า 7

4.2.1.3 การสร้างโมเดลเลนโบว์ลิ่ง



รูปที่ 4.4 รูปโมเดลเลนโบว์ลิ่ง

จากรูป 4.4 เป็นรูปที่แสดงการสร้างเลนโบว์ลิ่ง โดยใช้ Mesh รูปทรงต่างๆ ประกอบกันจนเป็นเลนโบว์ลิ่งซึ่งรายละเอียดของ Mesh ในการสร้างเลนโบว์ลิ่ง จะแสดงอยู่ในบทที่ 2 หน้า 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การรับค่าจากไมโครคอนโทรลเลอร์เพื่อใช้งานSerial Port

ในระบบโบว์ลิ่งเสมือนจริงจะมีส่วนที่ใช้ python scripts มีโค้ด ดังนี้

```
#Get Value from Serial port
        ser = serial.Serial(0)
        if ser.isOpen():
                ser.close()
                ser.open()
                value1 = int(ser.readline())
                value2 = int(ser.readline())
                value3 = int(ser.readline())
                print value1,value2,value3
                ser.close()
        #Simulation with IPO
        cont = GameLogic.getCurrentController()
        gameObject = GameLogic.getCurrentController().getOwner()
        position = gameObject.getPosition()

        gameObject = GameLogic.getCurrentController().getOwner()
        objectName = gameObject.getName()[2:]

        # Create an IPO of type Object named 'Recorded IPO'.
        blenderObject = Blender.Object.Get( objectName )
        ipo = Blender.Ipo.New( 'Object', 'Recorded IPO' )
        blenderObject.setIpo( ipo )

        # Get the position of gameObject as IPO curves.
        locx = ipo.addCurve( 'LocX' )
        locy = ipo.addCurve( 'LocY' )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
locz = ipo.addCurve( 'LocZ' )


# game Object per frame.

GameLogic.rec = [gameObject, locx, locy, locz]


#check Vertex

if value1==1 :  first=-4

elif value1==2 : first=-3

elif value1==3 : first=-2

elif value1==4 : first=-1

elif value1==5 : first=0

elif value1==6 : first=1

elif value1==7 : first=2

elif value1==8 : first=3

elif value1==9 : first=4

else :  first=5

if value2==1 : second=-4

elif value2==2 : second=-3

elif value2==3 : second=-2

elif value2==4 : second=-1

elif value2==5 : second=0

elif value2==6 : second=1

elif value2==7 : second=2

elif value2==8 : second=3

elif value2==9 : second=4

else : second=5

if value3==1 : third=-4

elif value3==2 : third=-3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
elif value3==3 : third=-2
elif value3==4 : third=-1
elif value3==5 : third=0
elif value3==6 : third=1
elif value3==7 : third=2
elif value3==8 : third=3
elif value3==9 : third=4
else : third=5

# Only try to update the curves if the variable has been initialised.
if hasattr( GameLogic, 'rec' ):

# Set the curves for the X, Y, and Z axis.
GameLogic.rec[1].addBezier( ( 0, -27 ) )
GameLogic.rec[1].update()
GameLogic.rec[2].addBezier( ( 0, first ) )
GameLogic.rec[2].update()
GameLogic.rec[3].addBezier( ( 0, 3.75 ) )
GameLogic.rec[3].update()
GameLogic.rec[1].addBezier( ( 20, -15 ) )
GameLogic.rec[1].update()
GameLogic.rec[2].addBezier( ( 20, second ) )
GameLogic.rec[2].update()
GameLogic.rec[3].addBezier( ( 20, 3.75 ) )
GameLogic.rec[3].update()
GameLogic.rec[1].addBezier( ( 40, 0 ) )
GameLogic.rec[1].update()
GameLogic.rec[2].addBezier( ( 40, third ) )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GameLogic.rec[2].update()
GameLogic.rec[3].addBezier( ( 40, 3.75 ) )
GameLogic.rec[3].update()
GameLogic.rec[1].addBezier( ( 100, 40 ) )
GameLogic.rec[1].update()
#GameLogic.rec[2].addBezier( ( 100, y4 ) )
#GameLogic.rec[2].update()
GameLogic.rec[3].addBezier( ( 100, 3.75 ) )
GameLogic.rec[3].update()
cont = GameLogic.getCurrentController()
motion1 = cont.getActuator("act4")
motion2 = cont.getActuator("act6")
GameLogic.addActiveActuator(motion1,1)
GameLogic.addActiveActuator(motion2,1)
else :
        print "Can't connect to serial port"
```

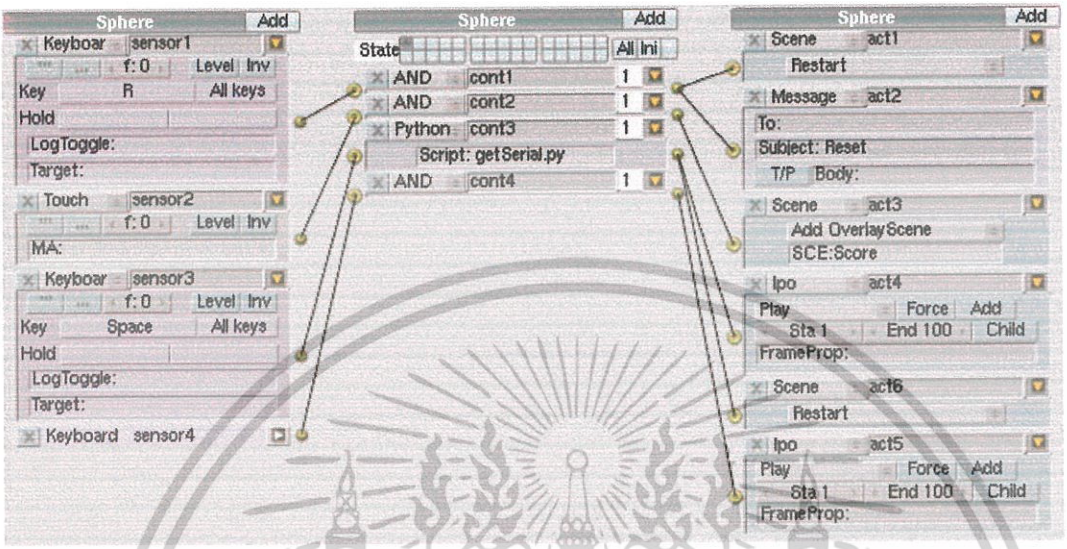
จากโค้ดดังกล่าวเป็นการใช้ python scripts โดยใช้ Pyserial ทำการรับค่าค่ามาเป็นสตริงจากไมโครคอนโทรเลอร์ เมื่อได้ค่ามาก็จะนำค่านั้นไปใช้ในการกำหนดเส้นทางการเคลื่อนที่ของลูกโบว์ลิ่งโดยใช้ฟังก์ชั่น IPO Curve มาทำการกำหนดการเคลื่อนที่ โดยเริ่มจากทำการกำหนดคีย์เฟรมของการเคลื่อนที่และกำหนดว่า ณ คีย์เฟรมดังกล่าวลูกโบว์ลิ่งอยู่ ณ ตำแหน่งไหน (x,y,z) ซึ่งทำการกำหนดอย่างนี้จนครบ 3 จุดตามที่ได้รับค่ามาจากไมโครคอนโทรลเลอร์ ส่วนจุดที่ 4 จุดสุดท้ายที่ลูกโบว์ลิ่งจะกระทบพิน ก็ให้ IPO Curve ทำการคำนวณให้โดยใช้หลักการทางฟิสิกส์ ก็จะได้มาว่าลูกโบว์ลิ่งกระทบพิน ณ ตำแหน่งไหน และหลังจากนั้นก็เป็นส่วนของการตรวจเช็คว่าพินใดโดยกระทบบ้าง แล้วแสดงผลมาเป็นคะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การกำหนดการรับค่าอินพุทและการแสดงเอ้าท์พุทของเกมส์

การใส่ Logic และปุ่มควบคุมการทำงานต่างๆให้กับแต่ละส่วนของเกมส์

4.2.3.1 การใส่ Logic ให้กับลูกโบว์ลิ่ง



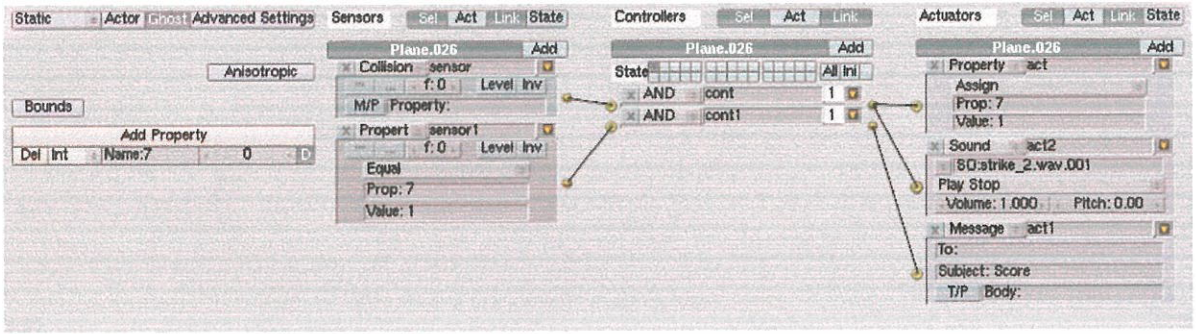
รูปที่ 4.5 รูปแสดงการใส่ Logic และปุ่มควบคุมการทำงานต่างๆให้กับแต่ละส่วนของเกมส์

จากรูป 4.5 แสดงรายละเอียดได้ดังนี้

1. Sensor 1 - กดปุ่ม R เพื่อ Restart Scene คือ เพื่อทำการเริ่มเกมส์ใหม่
2. Sensor 2 – เมื่อลูกโบว์ลิ่งสัมผัสกับวัตถุอื่นให้ทำการ Add OverlayScene นั่นคือ การ Add Scene ส่วนของ Score เข้ามาแสดงหน้าต่างเดียวกัน
3. Sensor 3 – เมื่อกดปุ่ม Space Bar ให้ทำการ Run Script : getSerial.py ซึ่งจะทำการรับค่าจาก Serial Port แล้วกำหนด IPO Curve แล้วให้ทำการ Restart Scene อีกครั้งเพื่อทำการ Reset ค่า IPO Curve ใหม่
4. Sensor 4 – เมื่อกดปุ่ม Enter ก็ให้ลูกโบว์ลิ่งเคลื่อนที่ตามเส้น IPO Curve ที่ได้กำหนดไว้ในข้อ c.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

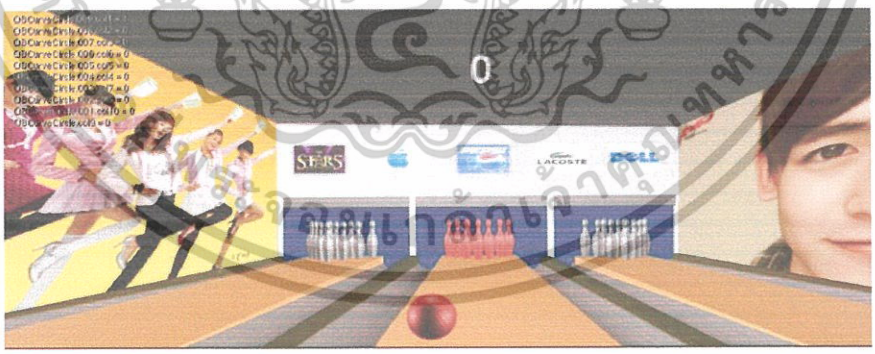
4.2.3.2 การใส่ Logic ให้กับพินโบว์ลิ่ง



รูปที่ 4.6 รูปแสดงการใส่ Logic ให้กับพินโบว์ลิ่ง

จากรูป 4.6 แสดงรายละเอียดได้ดังนี้

1. Add Property ขึ้นมาเพื่อทำการรับค่าเมื่อพินโบว์ลิ่งมีการกระทบเกิดขึ้น
2. Sensor – เมื่อพินโบว์ลิ่งมีการกระทบเกิดขึ้นให้ทำการกำหนดค่าให้กับ Property ที่สร้างขึ้นในข้อ a. ให้เท่ากับ 1
3. Sensor 1 – เมื่อค่า Property ในข้อ a. มีค่าเท่ากับ 1 นั่นคือแสดงว่าพินโดนกระแทกให้ทำการส่งเมสเสจไปยังหน้า Score เพื่อแสดงผลคะแนน
4. ทำแบบนี้กับทุกพินจนครบทั้ง 10 พิน



รูปที่ 4.7 เป็นการแสดงหน้าจอของเกมส์

จากรูป 4.7 เป็นหน้าจอที่แสดงเมื่อมีการเปิดโปรแกรมเบลนเดอร์เรียบร้อยแล้วหน้าจอนี้จะมีการทำงานโดยลูกโบว์ลิ่งจะเคลื่อนที่เมื่อได้รับค่าจาก Serial Port แล้วหาพิกัดจนกระทั่งชนพิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ

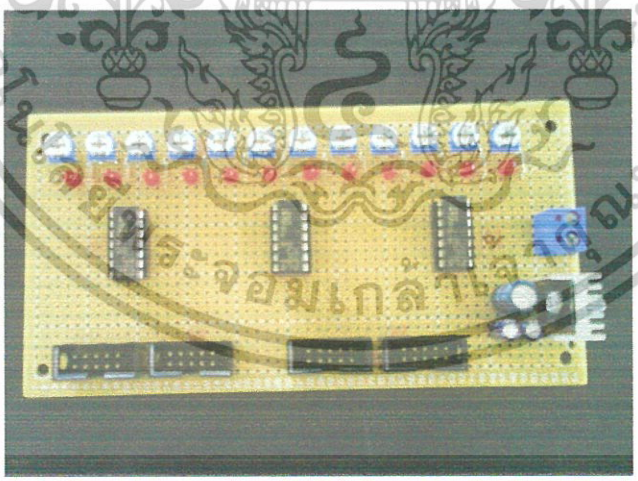
การพัฒนาโครงงานนี้ได้ใช้ฮาร์ดแวร์ ซอฟต์แวร์และเครื่องมือในการพัฒนาดังนี้

4.3.1 ฮาร์ดแวร์

อุปกรณ์ด้านอิเล็กทรอนิกส์

1.	VR เกือกม้า 10 K แนวนอน	12 ตัว
2.	ซ๊อกเก็ต IC 14M	4 ตัว
3.	LM 339	4 ตัว
4.	LDR 10 mm.	12 ตัว
5.	R1/4W5% 10 K	15 ตัว
6.	เลเซอร์ตัวเล็ก	12 ตัว
7.	ARM 7 LPC2148	1 ตัว
8.	สายแพร 10 P	10 ฟุต
9.	สายไฟ	20 เมตร
10.	แบตเตอรี่	12 V
11.	บอร์ด CP-JR ARM7 USB-LPC2148 EXP	1 ชุด

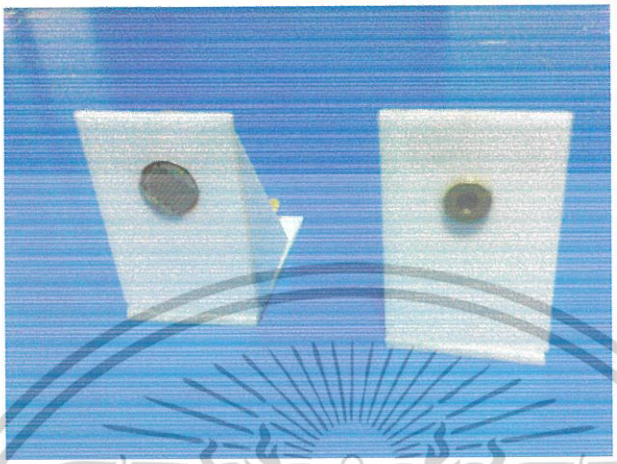
ภาพอุปกรณ์ด้านอิเล็กทรอนิกส์



รูปที่ 4.8 แผงวงจรควบคุมการทำงานของ Sensors

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.8 แผงวงจรควบคุมการทำงานของ Sensors นี้จะทำหน้าที่แสดงผลโดยใช้หลอด LED ในการเปล่งแสงออกมาโดยรับค่าจากเซนเซอร์ที่ติดอยู่บนเลนโบว์ลิ่ง



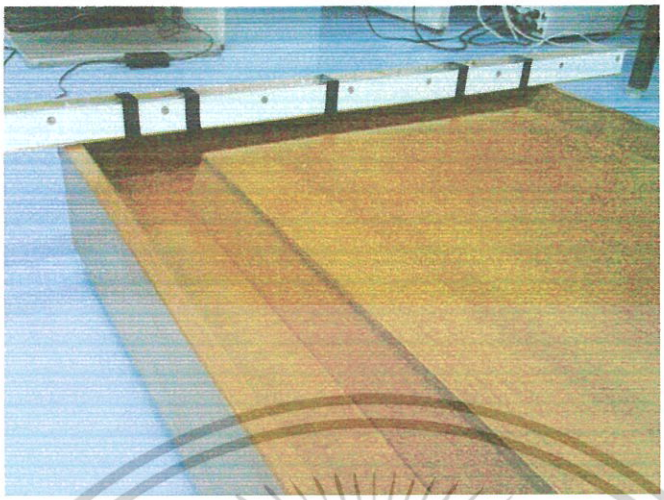
รูปที่ 4.9 ตัวเซนเซอร์ และ เลเซอร์

จากรูปที่ 4.9 จะเป็นลักษณะของตัวเซนเซอร์ และ เลเซอร์ที่ใช้ติดตั้งอยู่บนเลนโบว์ลิ่งใช้ในการจับตำแหน่งของลูกโบว์ลิ่งเพื่อนำไปคำนวณหาเส้นทางการเคลื่อนที่ต่อ

รูปที่ 4.10 ภาพการนำเลเซอร์มาประกอบติดกับเลนโบว์ลิ่ง

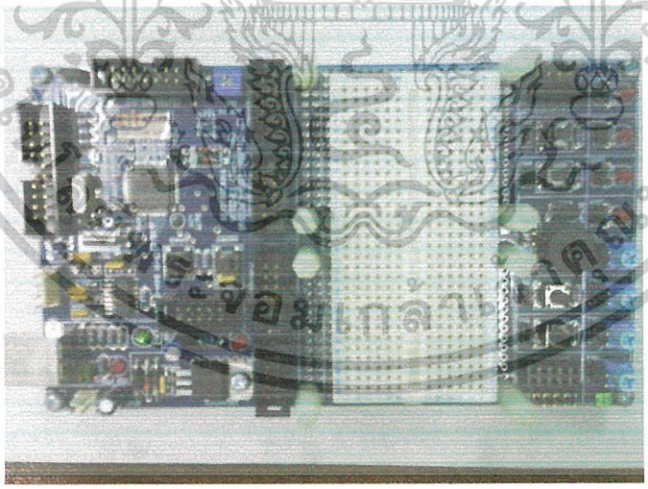
จากรูปที่ 4.10 เป็นการนำตัวเลเซอร์มาติดตั้งเข้ากับเลนโบว์ลิ่งโดยติดตั้งที่ตำแหน่งเริ่มต้นของเลนโบว์ลิ่งทั้งหมด 5 ตัวในแนวแกน X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 ภาพการนำเซนเซอร์มาประกอบติดกับเลนโบว์ลิ่งส่วนปลายเลน

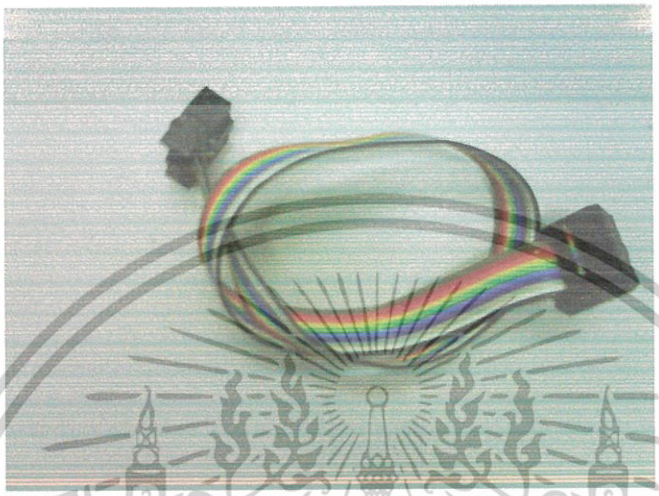
จากรูปที่ 4.11 เป็นการนำตัวเซนเซอร์มาประกอบติดเข้ากับเลนโบว์ลิ่งโดยติดตั้งที่ตำแหน่งส่วนปลายของเลนเพื่อเป็นตัวรับแสงจากเลเซอร์ที่ยิงตรงมาจากส่วนต้นทางของเลน



รูปที่ 4.12 ภาพของบอร์ด CP-JR ARM7 USB-LPC2148 EXP ที่ใช้ในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.12 เป็นภาพของบอร์ด CP-JR ARM7 USB-LPC2148 EXP ที่ใช้ในระบบ บอร์ดนี้จะมีหน้าที่ในการรับค่าที่ได้จากตัวเซนเซอร์ที่ติดอยู่กับเลนโบว์ลิ่งและจะส่งค่านั้นเข้าไปคำนวณต่อในโปรแกรม Blender เพื่อทำการคำนวณหาเส้นทางต่อจนกระทั่งลูกโบว์ลิ่งชนพิน



รูปที่ 4.13 ภาพของสายแพขนาด 10 พินที่ใช้ในระบบ

จากรูปที่ 4.13 เป็นภาพของสายแพพินขนาด 10 พินที่ใช้เชื่อมต่อเซนเซอร์ที่ติดอยู่กับเลนโบว์ลิ่งเข้ากับตัวไมโครคอนโทรเลอร์และเชื่อมจากตัวไมโครคอนโทรเลอร์เข้าตัวบอร์ด ARM 7

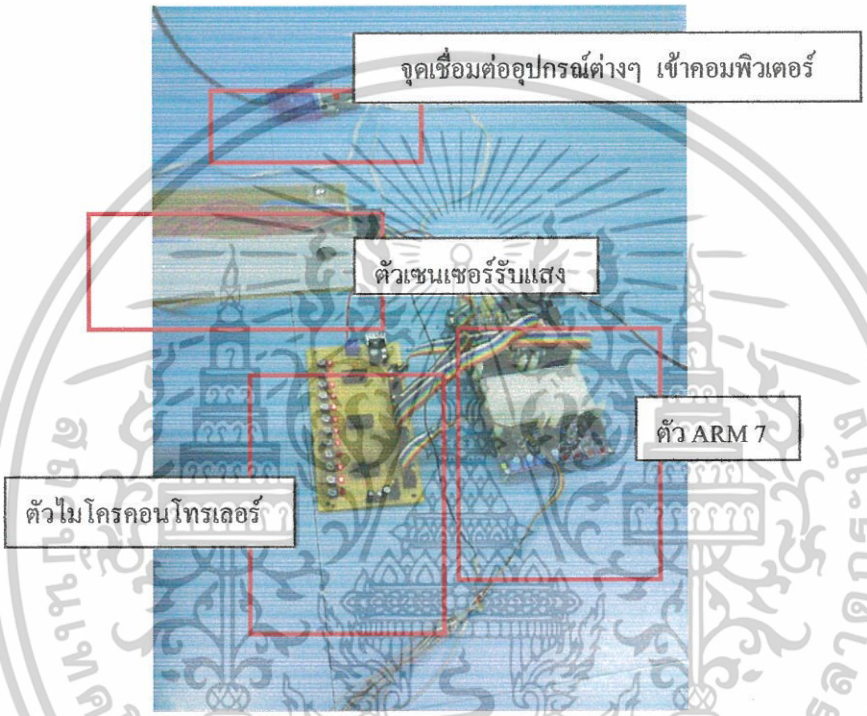
4.3.2 ซอฟต์แวร์

1. Blender
2. C
3. Microsoft Office 2007
4. Visio 2007
5. Keil uVision3
6. Philips LPC2000 Flash Utility V2.2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 อธิบายการติดตั้งและการทำงานของโปรแกรม

1. เริ่มจากการต่ออุปกรณ์ต่าง ๆ เข้ากับเครื่องคอมพิวเตอร์ โดยเริ่มจากเชื่อมตัวเซนเซอร์เข้ากับตัวไมโครคอนโทรเลอร์เพื่อรับค่า Input โดยมีสายแพรขนาด 10 พินเป็นตัวเชื่อม เมื่อเรียบร้อยแล้วก็ทำการเชื่อมตัวตัวไมโครคอนโทรเลอร์ออกเป็นค่า Output ไปยังตัวบอร์ด Arm 7 และสุดท้ายเชื่อมตัว Arm เข้ากับคอมพิวเตอร์โดยเชื่อมผ่านพอร์ต serial เพื่อนำค่าไปคำนวณต่อในโปรแกรมเบรนเดอร์ ดังรูปที่ 4.14 จะแสดงการเชื่อมต่ออุปกรณ์ต่างๆเข้าด้วยกัน



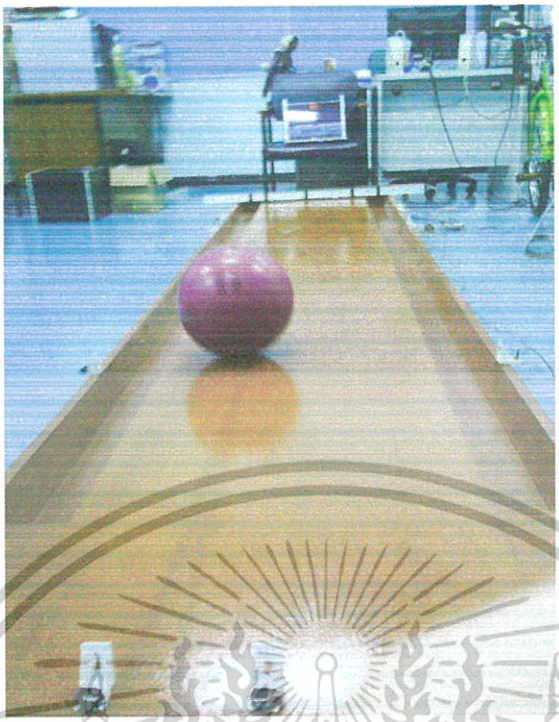
รูปที่ 4.14 ภาพแสดงการเชื่อมต่ออุปกรณ์

2. ทำการกำหนดค่าอุปกรณ์ที่เชื่อมต่อให้เป็นค่าเริ่มต้นทั้งหมด

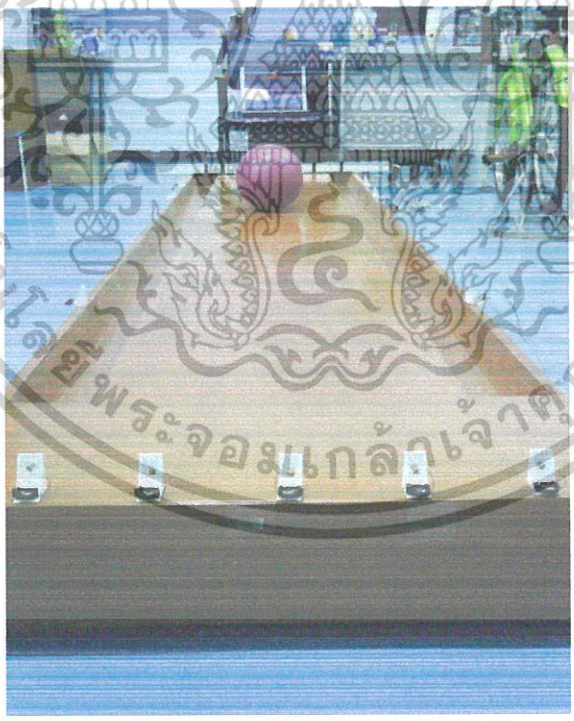
3. เปิดโปรแกรมเบลนเดอร์ เพื่อการจำลองภาพต่อจากเลนโบว์ลิ่งเมื่อลูกโบว์ลิ่ง เคลื่อนที่ไปสุดปลายเลน โดยเครื่องคอมพิวเตอร์จะวางไว้ปลายเลนโบว์ลิ่ง

4. โปรแกรมพร้อมใช้งาน ผู้ใช้งานสามารถเริ่มโยนลูกโบว์ลิ่ง โดยที่จากรูป 4.15 และ รูปที่ 4.16 จะสังเกตว่าที่รอบเลนโบว์ลิ่งจะมีเซนเซอร์ติดอยู่ทั้งหมด 8 จุดโดยจะมีหน้าที่ในการรับค่าจากเลเซอร์เพื่อไปประมวลผลต่อในโปรแกรมเบลนเดอร์โดยในขณะที่ลูกโบว์ลิ่งยังเคลื่อนที่ไม่ผ่านเซนเซอร์ตัวสุดท้ายที่ติดอยู่ปลายเลนส์ หน้าจอคอมพิวเตอร์จะแสดงผลการจำลองพิน และเลนส์โบว์ลิ่งดังรูปที่ 4.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

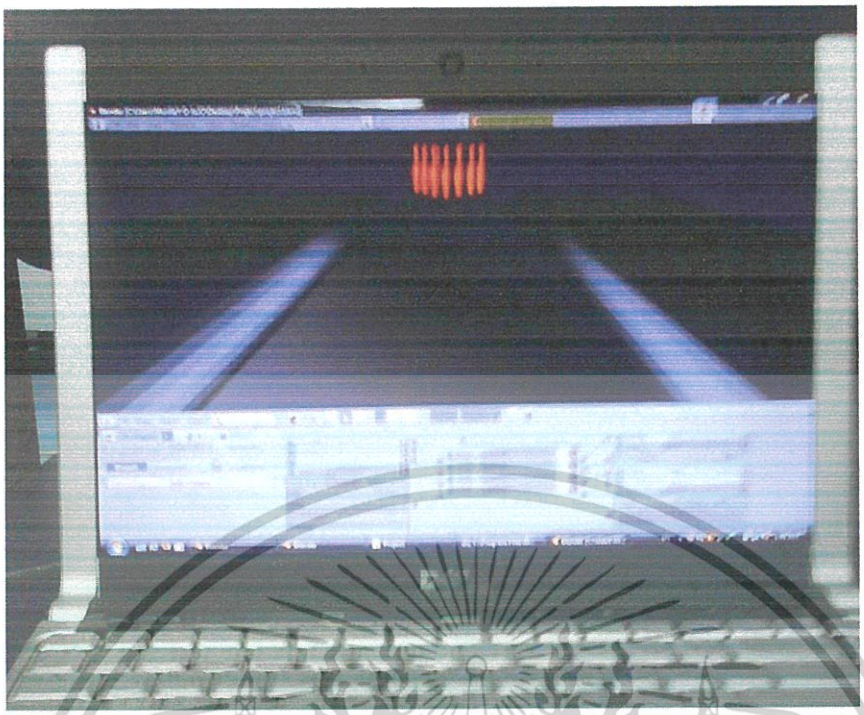


รูปที่ 4.15 ภาพแสดงการเคลื่อนที่ของลูกโบว์ลิ่งเมื่อผู้ใช้เริ่มโยน(1)



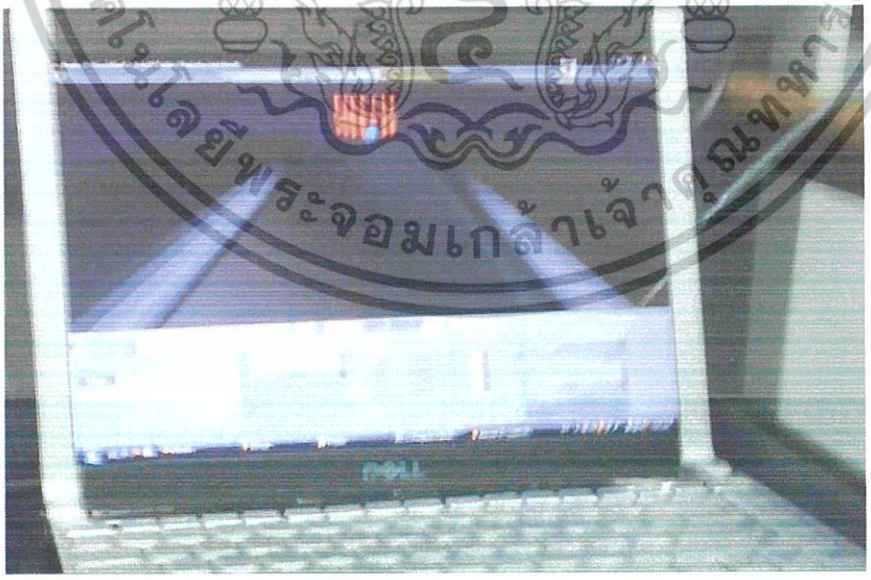
รูปที่ 4.16 ภาพแสดงการเคลื่อนที่ของลูกโบว์ลิ่งเมื่อผู้ใช้เริ่มโยน(2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



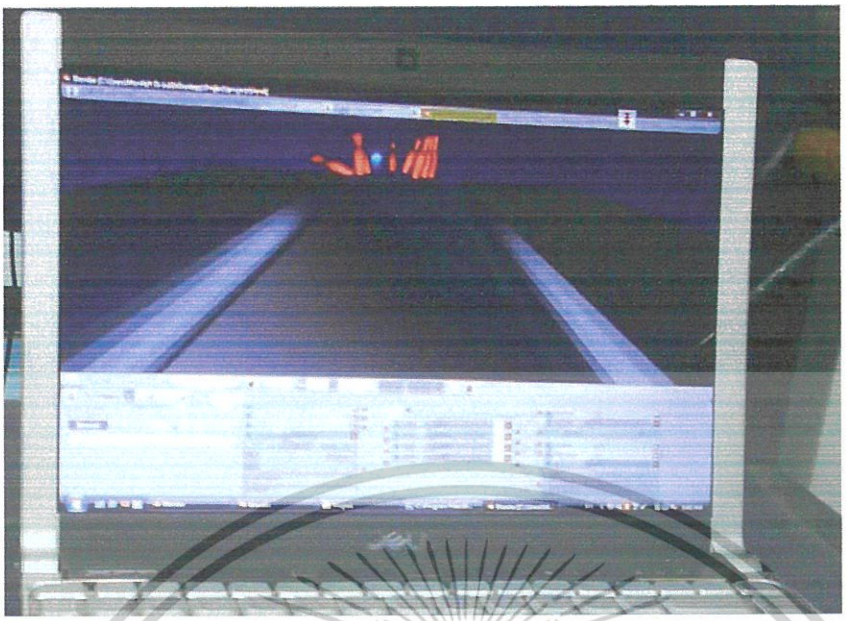
รูปที่ 4.17 ภาพแสดงการจำลองพินและเลนส์โบว์ลิ่งจากโปรแกรมเบลนเดอร์

5. เมื่อลูกโบว์ลิ่งเคลื่อนที่ผ่านเซนเซอร์ตัวสุดท้ายที่อยู่ปลายเลนส์ จากนั้นจะส่งค่าการเคลื่อนที่ต่อให้กับ โปรแกรมเบลนเดอร์ทำการประมวลผลการเคลื่อนที่จนกระทั่งลูกโบว์ลิ่งเคลื่อนที่ชนพินล้ม ดังรูป 4.18

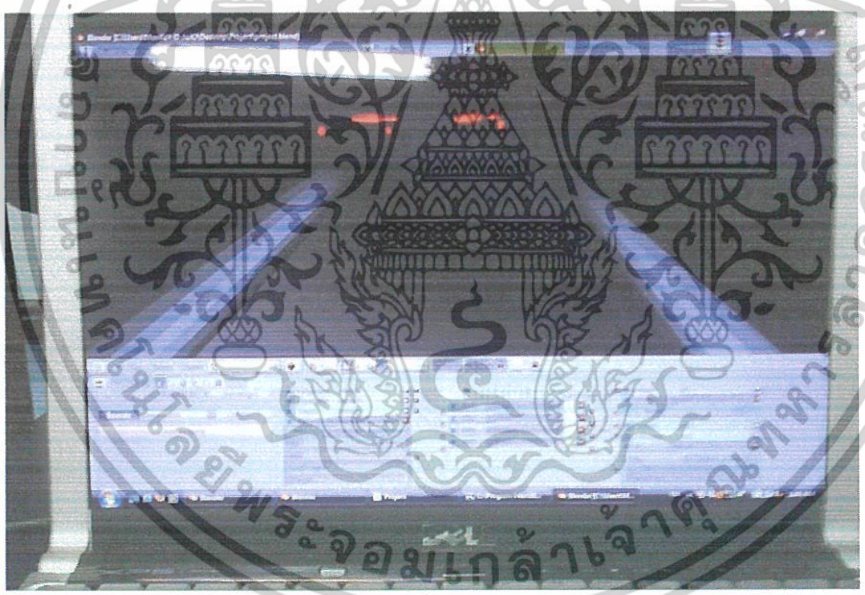


รูปที่ 4.18 ภาพจำลองจากโปรแกรมเมื่อลูกโบว์ลิ่งเคลื่อนที่ผ่านเซนเซอร์ตัวสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 ภาพจำลองจากโปรแกรมที่เมื่อลูกโบว์ลิ่งเคลื่อนที่ชนพิน(1)



รูปที่ 4.20 ภาพจำลองจากโปรแกรมที่เมื่อลูกโบว์ลิ่งเคลื่อนที่ชนพิน(2)

6. เมื่อพินล้มโปรแกรมจะคำนวณแต้มที่ได้พร้อมแสดงผลให้ทราบ จากนั้นหากต้องการเล่นใหม่ ต้องทำการตั้งค่าใหม่ให้กับอุปกรณ์ผ่านปุ่ม reset ดังรูป 4.19 และ 4.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปการพัฒนาโครงการ

5.1 สรุปโครงงาน

ผลการดำเนินงานพัฒนาระบบโบว์ลิ่งเสมือนจริง สามารถสรุปได้ดังนี้

ผู้พัฒนาได้ทำการศึกษาข้อมูลเกี่ยวกับการเคลื่อนที่ของลูกโบว์ลิ่งเพื่อวิเคราะห์ว่า สมควรใช้อุปกรณ์ใดในการจับภาพเคลื่อนไหว จากนั้นได้ทำการออกแบบและสร้างฮาร์ดแวร์ที่ใช้ในระบบว่าควรมีลักษณะใด ความกว้าง ความยาว และตำแหน่งในการติดเซนเซอร์ เพื่อรับค่าลูกโบว์ลิ่งที่เคลื่อนที่บนเลน แล้วนำค่านั้นไปประมวลผลหาทิศทางการเคลื่อนที่ จากนั้นทำการต่อเข้าสู่ส่วนของโปรแกรมเบลนเดอร์ที่ได้ทำการสร้างพิน เลนจำลองเรียบร้อยแล้ว เมื่อรวมจะนำค่าที่ได้จากการโยนมาใช้ในการคำนวณว่าพินใดจะล้มอย่างไรและมีแต้มเท่าไร

5.2 ประโยชน์ที่ได้รับจากการพัฒนาโครงงาน

1. ได้เรียนรู้เกี่ยวกับโปรแกรมเบลนเดอร์ (Blender) ที่นำมาใช้พัฒนาระบบ
2. ได้เรียนรู้เกี่ยวกับไมโครคอนโทลเลอร์ที่นำมาใช้พัฒนาระบบ
3. ได้เรียนรู้เรื่องการออกแบบเครื่องมือหรืออุปกรณ์ให้เหมาะสมกับความต้องการและสามารถนำไปใช้งานได้จริง
4. ได้เรียนรู้ที่จะแก้ปัญหาที่เกิดขึ้นจากฮาร์ดแวร์และซอฟต์แวร์ขั้นที่สูงมากขึ้น
5. ได้เรียนรู้เกี่ยวกับเรื่องราวของกีฬาโบว์ลิ่ง เช่นกติกา,ความกว้างของเลน,ความยาว, จำนวนพิน,และการคิดแต้มของกีฬาชนิดนี้
6. ได้เรียนรู้การทำงานเป็นทีม
7. ได้เรียนรู้ว่าการวางแผนกับการทำงานจริงแตกต่างกันซึ่งในการทำงานเราจะต้องมีการปรับเปลี่ยนแผนนั้น

5.3 ปัญหาที่พบในการพัฒนาโครงงานและแนวทางในการแก้ไข

- **ปัญหาที่พบ** ตัวไมโครคอนโทรลเลอร์ (CP-JR ARM7 USB-LPC2148) จะต้องใช้พอร์ต serial เนื่องจากในคอมพิวเตอร์โน๊ตบุ๊คไม่มี พอร์ต ชนิดนี้ จึงต้องใช้ต้องใช้อุปกรณ์แปลงจาก usb to serial ซึ่งเป็นอุปกรณ์ชนิดนี้ลง driver ได้ยากมากใน windows vista

  แนวทางในการแก้ปัญหา ได้เปลี่ยนจากการใช้คอมพิวเตอร์โน๊ตบุ๊คมาใช้ PC ในการรันโปรแกรมลงไมโครคอนโทรลเลอร์ (CP-JR ARM7 USB-LPC2148) แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ปัญหาที่พบ** หลังจากการทำอุปกรณ์ในวิชาโครงงาน 1 ที่ผ่านมามีปัญหาในส่วนของเลเซอร์และเซนเซอร์ที่ไม่ตรงกัน ทำให้ทางกลุ่มนำอุปกรณ์เหล่านั้นมาปรับปรุงใหม่เพื่อที่จะทำให้ทั้งตัวรับและตัวส่งสัญญาณตรงกันมากขึ้นเพื่อเมื่อนำไปติดตั้งกับตัวเลนโบว์ลิ่งแล้วจะไม่เกิดปัญหาตัวรับตัวส่งไม่ตรงกัน
  แนวทางในการแก้ปัญหา ตัวเลเซอร์และเซนเซอร์แต่ละตัวจะแยกออกจากกันเป็นอิสระติดตั้งอยู่บนเลนโบว์ลิ่งทั้ง 8 จุด

- **ปัญหาที่พบ** ในช่วงแรกของการพัฒนาผู้จัดทำได้จับการเคลื่อนที่ของลูกโบว์ลิ่งด้วยกล้อง แต่ความละเอียด และข้อมูลที่ได้ไม่เพียงพอต่อการนำมาประมวลผล
  แนวทางในการแก้ไข ได้เปลี่ยนวิธีในการรับค่าโดยเลือกใช้เซนเซอร์ในการจับภาพเพื่อให้เกิดความละเอียดของข้อมูลที่จะนำมาประมวลผลหาความเร็วและทิศทางการเคลื่อนที่ โดยใช้ เซนเซอร์ทั้งหมด 8 ตัว

- **ปัญหาที่พบ** ยังขาดความรู้ในเรื่องโปรแกรมเบลนเดอร์ (Blender) และการเขียนภาษาซีเพื่อรันลงไมโครคอนโทรลเลอร์ (CP-JR ARM7 USB-LPC2148) ทำให้การดำเนินการทำโปรเจ็คเป็นไปอย่างไม่ค่อยราบรื่นนัก
  แนวทางในการแก้ไข ให้เวลากับการศึกษามากขึ้นและเข้าหาผู้ที่มีความรู้เพื่อช่วยแนะนำในส่วนที่ขาดหายไป

- **ปัญหาที่พบ** การสร้างเลนโบว์ลิ่งที่เดิมทีจะสร้างตามความกว้างจริงของเลนมาตรฐานและมีความยาว ขนาด 6 เมตร ไม่สามารถสร้างได้จริงตามที่วางแผนไว้
  แนวทางในการแก้ไข นำเลนโบว์ลิ่งของเดิมที่มีอยู่แล้วซึ่งมีขนาดเล็กกว่าที่ได้วางแผนไว้มาปรับใช้ในโปรเจ็ค

- **ปัญหาที่พบ** การนำเลนโบว์ลิ่งของเดิมที่มีอยู่แล้วมาใช้ทำให้อุปกรณ์ที่ได้สร้างไว้แล้วในตอนโครงงานที่ 1 บางส่วนใช้งานไม่ได้
  แนวทางในการแก้ไข ปรับเปลี่ยนอุปกรณ์เดิมที่มีอยู่ให้ใช้ได้กับเลนโบว์ลิ่ง

- **ปัญหาที่พบ** ราคาของอุปกรณ์ที่ใช้ในการพัฒนามีราคาแพง
  แนวทางในการแก้ไข เลือกซื้ออุปกรณ์มือสองที่คุณภาพดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 แนวทางในการพัฒนาในอนาคต

ผู้พัฒนาได้พัฒนาให้เกิดเป็นโบว์ลิ่งเสมือนจริงแล้วในส่วนโครงสร้างหลัก ๆ เช่นมีรางโบว์ลิ่งที่มีการติดตั้งตำแหน่งเซนเซอร์ต่าง ๆ มีการใช้โปรแกรมเบลนเดอร์ในการทำให้เกิดเป็นการเคลื่อนที่ต่อส่วนที่ลูกโบว์ลิ่งจะกระทบพินและล้มลง ผู้พัฒนามีความคาดหวังว่า โบว์ลิ่งเสมือนจริงนี้ควรมีการพัฒนาต่อให้สามารถที่จะใช้งานได้เหมือนของจริงคือโยนได้บนเลนส์โบว์ลิ่งที่สร้างขึ้นมาตามขนาดความกว้างของจริง มีเสียงลูกโบว์ลิ่งกระทบพินเพื่อให้สมจริงมากขึ้น อาจมีการเพิ่มในส่วนของระดับการเล่นว่าสามารถที่จะเลือกโยนแบบยาก หรือแบบง่าย สำหรับมือสมัครเล่นที่ฝึกเล่นโบว์ลิ่ง ในลักษณะเกมส์เอ็นจิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

WWW.ETT.CO.TH

SiPA (2549).blender **คู่มือสร้างงาน 3D แบบครบวงจร.**กรุงเทพฯ

โอภาส ศิริครรชิตถาวร (2549).**เรียนรู้และพัฒนาไมโครคอนโทรลเลอร์ ARM7 LPC2148 ด้วยภาษาซี**. กรุงเทพฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ประวัติ

กีฬาโบว์ลิ่ง (Bowling) เริ่มเข้ามาแพร่หลายในประเทศไทยตั้งแต่เมื่อใดไม่มีหลักฐานที่แน่นอน แต่ประมาณ 10 ปีที่ผ่านมา กีฬาโบว์ลิ่งได้ปรากฏฐานที่มั่นคงโดยมีสถานที่และอุปกรณ์การเล่นถูกต้องตามมาตรฐานสากล บางคนมีความเข้าใจกันว่ากีฬาโบว์ลิ่งเป็นเกมกีฬาใหม่ชนิดหนึ่ง แต่ความจริงแล้วกีฬาโบว์ลิ่งเป็นกีฬาที่ชาวโลกได้เล่นกันมาประมาณ 5,000 ปี ก่อนคริสตศักราช ในสมัยนั้นชาวอียิปต์ได้มีการเล่นกีฬาชนิดหนึ่ง ซึ่งคล้ายกับการเล่นโบว์ลิ่ง โดยใช้ไม้เป็นแก่นและใช้ลูกหินกลมๆ กลิ้งไปทอยแก่นให้ล้ม หลักฐานยืนยันในข้อนี้ปรากฏอยู่ในรูปภาพในหลุมฝังศพเด็กชาวอียิปต์ในสมัยนั้น ชาวโบลิเนเซียน (Bolinasian) มีการเล่นที่เรียกว่า อูลาไมก้า วิธีการเล่นนี้ผู้เล่นจะต้องกลิ้งก้อนหินไปให้ถูกเป้า ซึ่งตั้งห่างออกไป 60 ฟุต เป็นที่น่าแปลกอย่างหนึ่ง กล่าวคือ มีระยะจากเส้นฟาวล์ถึงพินหนึ่งของโบว์ลิ่งในปัจจุบัน

เมื่อประมาณต้นสมัยคริสตศักราช ในประเทศสหพันธ์สาธารณรัฐเยอรมนี การเล่นกีฬาชนิดหนึ่งเรียกว่า เคเกล (Kegel) ซึ่งคล้ายกับการเล่นโบว์ลิ่งและเคเกลนี้เองที่เป็นรากศัพท์ของคำว่าเคเกลอรี่ (Kegler) ซึ่งหมายถึงนักโบว์ลิ่งบรรดานักบวชในเยอรมันถือว่ากีฬาเคเกลนี้เป็นส่วนหนึ่งของพิธีกรรมในศาสนาต่อมาในสมัยคริสตศักราชที่ 15 Martin Luther ผู้ตั้งนิกายโปรเตสแตนต์เป็นนักเล่นเคเกลคนสำคัญคนหนึ่ง ได้วางกฎเกณฑ์การเล่นเคเกล โดยกำหนดให้มี 1 พิน ทำให้เคเกลได้แพร่หลายไปยังประเทศต่างๆ ในยุโรป ชาวอังกฤษจึงนำเอาไปดัดแปลงเล่นแบบสนามหญ้าเรียกว่า โบว์ลิ่งสนาม และเมื่อชาวดัตช์อพยพเข้ามาอยู่ในอเมริกา ที่นิวอัมสเตอร์ดัมก็ได้นำเอากีฬาเคเกลนี้เข้าไปเล่นด้วย

ในปี พ.ศ. 2166 ได้มีการสร้างสถานโบว์ลิ่งขึ้นเป็นครั้งแรก นับว่าเป็นการปฏิวัติจากกีฬากลางแจ้งมาเป็นกีฬาในร่ม และเพิ่มพินขึ้นอีกหนึ่งพินเป็น 10 พินเรียกว่า โบว์ลิ่ง โบว์ลิ่งได้แพร่หลายออกไปอย่างรวดเร็วที่อเมริกาและประเทศอื่นๆ ทั่วโลก เฉพาะในสหรัฐอเมริกามีนักโบว์ลิ่งกว่า 30 ล้านคน มีการจัดตั้งสมาคมให้ชื่อว่าอเมริกันโบว์ลิ่งคองเกรส ซึ่งเป็นที่รู้จักกันมาในนาม เอ.บี.ซี. (A.B.C.) สมาคมนี้เป็นที่ยอมรับนับถือกันอย่างกว้างขวางว่าเป็นผู้วางรากฐานและกฎเกณฑ์ในการเล่นโบว์ลิ่ง ตลอดจนการแข่งขันโบว์ลิ่งประจำปี เพื่อหาผู้ชนะเลิศซึ่งมีผู้เข้าร่วมการแข่งขันปีละจำนวนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กติกาโบว์ลิ่ง

เกมการเล่น – คำจำกัดความ (game – definition)

เกมการเล่นโบว์ลิ่งสิบพินใน หนึ่งเกมจะถูกแบ่งออกเป็น 10 เฟรม ผู้เล่นแต่ละคนสามารถโยนลูกโบว์ลิ่ง ได้ 2 ครั้งในแต่ละ เฟรม ตั้งแต่เฟรมที่ 1 ถึง 9 ยกเว้นในกรณีสไต้รค์จะโยนได้เพียงลูกเดียว ในเฟรมที่ 10 ผู้เล่นสามารถโยนลูกโบว์ลิ่งได้ 3 ครั้งบนเลนเดียวกันถ้าผู้เล่นคนนั้นสามารถทำสไต้รค์หรือแปร์ได้ ผู้เล่นจะต้องโยนในเฟรมของตัวเองให้เสร็จเรียบร้อยตามลำดับ

เกมการเล่น – การบันทึกคะแนน (game – how scored)

จำนวนพินที่ล้มโดยการโยนลูกแรกในแต่ละเฟรม จะต้องถูกเขียนลงในช่องสี่เหลี่ยมเล็กที่อยู่มุมซ้ายด้านบนของตารางเฟรมนั้น และจำนวนพินล้มโดยการโยนของแต่ละเฟรม จะต้องถูกเขียนลงในช่องสี่เหลี่ยมเล็กที่อยู่มุมขวาด้านบนของตารางเฟรมนั้น ถ้าในการโยนครั้งทีสอง ไม่มีพินใดล้มเลย ให้ทำ เครื่องหมาย - ลงในช่องสี่เหลี่ยมเล็กการนับคะแนนในแต่ละเฟรมจะต้องถูกบันทึกลงในที่ว่างด้านล่างของเฟรมนั้น

1		2		3		4		5		6		7		8		9		10	

สไตรค์ (strike)

การทำสไต้รค์คือการโยนลูกแรกในแต่ละเฟรม และสามารถ ทำให้พินที่ตั้งอยู่ในตำแหน่งที่ถูกต้องทั้งสิบพินล้มลงทั้งหมด ให้ทำเครื่องหมาย x ลงในช่องสี่เหลี่ยมเล็กมุมซ้ายด้านบนของเฟรมนั้น การทำสไตร์คจะได้ สิบคะแนน รวมกับจำนวนพินล้มที่ทำได้ในการโยนลูกอีกสองครั้งของผู้เล่นคนนั้น สไตร้ค 2 ตัวติดกัน (double) เมื่อผู้เล่นสามารถทำสไต้รค์ได้สองครั้งติดกัน คะแนนของสไตร้คตัวแรกจะเป็น 20 คะแนน รวมกับจำนวนพินล้มที่ทำได้ในลูกถัดจากสไต้รค์ตัวที่สองสไตร์ค 3 ตัว ติดกัน (triple) เมื่อผู้เล่นสามารถทำสไตร้คได้สามครั้งติดกัน คะแนนของสไต้รค์ตัวแรกจะเป็น 30 คะแนน การจะทำคะแนนเต็ม 300 คะแนน ผู้เล่นจะต้องทำสไตร้คติดต่อกัน 12 ครั้งในเกมนั้น

สแปร์ (spare)

ผู้เล่นใดสามารถทำให้พินที่เหลือจากการโยนครั้งแรก ล้มลงในการโยนครั้งที่สองในเฟรมนั้น ให้ถือว่าผู้เล่นนั้นทำสแปร์ได้ ให้ทำเครื่องหมาย/ ลงในช่องสี่เหลี่ยมเล็กมุมขวาด้านบนของเฟรมที่ทำสแปร์ได้ คะแนนของเฟรมที่ทำสแปร์ได้จะเป็นลิบคะแนนรวมกันจำนวนพินที่ล้มลงในลูกถัดไปของผู้เล่นนั้นโอเพ่นเฟรม (open frame) การที่ผู้เล่นไม่สามารถทำให้พินทั้งสิบล้มลงได้หมดในการโยนสองครั้งในแต่ละเฟรม ยกเว้นสปลิด ให้เรียกเฟรมนั้นว่า โอเพ่นเฟรม

สปลิท(split)

สปลิท คือ การตั้งของพินที่เหลืออยู่หลังจากที่ได้โยนลูกแรกไปแล้ว โดยที่ไม่มีพินหนึ่งตั้งอยู่และ อย่างน้อยมีพินหนึ่งล้มลงระหว่างพินอย่างน้อยสองพินที่ตั้งเหลืออยู่ เช่น 7 – 9 หรือ 3 –

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10 อย่างน้อยต้องมีพินล้มลงข้างหน้า ของสองหรือสามพินที่ตั้งเหลืออยู่ เช่น 5-6 เครื่องหมายที่ใช้แทนสปลิท คือ 0

รูปแบบของการเล่น (style of play)

การเล่นในเกมหนึ่งให้ใช้เล่นในสองเลนที่อยู่คู่กัน โดยผู้เล่นที่เล่นในประเภททีม ทีม 3 คน คู่ หรือเดียวจะต้องโยนเรียงตามลำดับของผู้เล่นในแต่ละเฟรม ผู้เล่นจะเปลี่ยนเลนโยนในเฟรมต่อไป ผู้เล่นแต่ละคนจะได้โยนทั้งหมดห้าเฟรมในแต่ละเลนที่อยู่คู่กันพินล้มถูกกติกา (legal pinfall) ลูกโบว์ลิ่งจะถือว่าถูกปล่อยออกไปอย่างถูกต้องเมื่อลูกโบว์ลิ่งพ้นจากตัวผู้เล่นและข้ามเส้นฟาวล์เข้าไปในเขตเลนการเล่น การโยนลูกทุกครั้งจะถูกนับคะแนน ยกเว้นแต่ในกรณีที่ตัดสินว่าเป็นลูกผิดกติกา (dead ball) ลูกโบว์ลิ่งจะต้องถูกปล่อยออกไปโดยใช้มือเท่านั้น และจะใช้เครื่องใด ๆ มาช่วยไม่ได้ ไม่ว่าจะอยู่ภายในลูกโบว์ลิ่งหรือติดอยู่กับลูกโบว์ลิ่ง และไม่ว่าจะหลุดออกจากลูกโบว์ลิ่งในขณะปล่อยออกไป หรือเป็นส่วนที่เคลื่อนไหวไปกับลูกระหว่างการปล่อยลูกออกไป *ผู้เล่นสามารถที่จะใช้อุปกรณ์เสริมในการช่วยยึดจับลูกโบว์ลิ่งและช่วยในการโยนได้ถ้าอุปกรณ์อยู่ในมือ หรือแยกส่วนออกจากลูกโบว์ลิ่ง

พินที่จะนับเป็นคะแนนให้กับผู้เล่นจากการโยนที่ถูกต้องมีดังนี้

พินที่ล้มหรือพินที่หลุดพ้นออกจากผิวเลนไม่ว่าจะเกิดจากลูกโบว์ลิ่งหรือจากพินอื่น พินที่ล้มลงหรือพินที่หลุดพ้นออกจากผิดเลนโดยพินที่กระดอนจากด้านข้างและกำแพงปะทะลูกโบว์ลิ่งด้านหลัง พินที่ล้มลงหรือพินที่หลุดพ้นออกจากผิดเลนโดยพินที่กระดอนจากเครื่องกวาดพินที่ลงมาอยู่ในบริเวณที่ตั้งพินก่อนที่จะทำการกวาดพิน พินที่พิงกับขอบรางหรือผนังด้านข้าง พินที่ล้มหรือพินที่พิงกับขอบรางหรือผนังด้านข้างจะต้องถูกทำให้พ้นไปจากเลนก่อนที่จะทำการโยนลูกต่อไป

พินล้มที่ผิดกติกา (illegal pinfall) ลูกที่โยนแต่จะไม่นับคะแนน แต่ถือว่าเป็นการโยน ดังนี้

1. ลูกโบว์ลิ่งหลุดพ้นจากผิดเลนก่อนที่จะถูกพินใดพินหนึ่ง
2. ลูกโบว์ลิ่งกระดอนกลับจากกำแพงปะทะลูกด้านหลัง
3. พินที่ล้มลงโดยพินที่ไปโดนร่างกาย แขน หรือขาที่ตั้งพินและกระดอนกลับ
4. พินที่ตั้งอยู่และล้มลงโดยไปถูกกับอุปกรณ์เครื่องตั้งพิน
5. พินที่ถูกทำให้ล้มลงโดยพินที่ล้มไปแล้วในขณะที่เครื่องกวาดพินกำลังกวาดพิน
6. พินที่ล้มลงโดยกรรกระทำของเจ้าหน้าที่ตั้งพิน
7. ผู้เล่นทำฟาวล์
8. ผู้เล่นโยนลูกโบว์ลิ่งในขณะที่มีพินล้มอยู่ บนเลนหรือในราง และลูกโบว์ลิ่งไปโดนพินดังกล่าว จนทำให้ลูกโบว์ลิ่งหลุดพ้นออกจากผิวเลน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. หากเกิดกรณีที่พินล้มโดยผิดกติกา และผู้เล่นได้สิทธิ์โยนใหม่ในเฟรมนั้น ให้นำพินที่ล้มโดยผิดกติกาทั้งหมดมาตั้งไว้ในจุดที่พินนั้นๆ ตั้งอยู่ดังเดิม
10. พิน- ตั้งไม่ถูกต้อง (pins – improperly set)
11. ในกรณีที่ผู้เล่นได้โยนลูกโบว์ลิ่งหลังจาก การตั้งพินทั้งหมดหรือตั้งพินเพื่อเก็บสแปร์ แล้วพบว่ามีพินใดพินหนึ่งหรือมากกว่าตั้งไม่ถูกจุด แต่จำนวนพินครบ ให้นับว่าเป็นการโยนที่ถูกต้องและนับเป็นคะแนน ผู้เล่นจะต้องเป็นผู้รับผิดชอบในการตรวจดูว่าพินถูกตั้งในตำแหน่งที่ถูกต้อง หากพินใดตั้งไม่ถูกต้อง ผู้เล่นจะต้องแจ้งต่อผู้รับผิดชอบให้ดำเนินการตั้งพินให้ถูกต้อง หากไม่มีการแจ้งจะถือว่าพินที่ถูกตั้งผิดจุดนั้นได้รับการยอมรับจากผู้เล่น และพินที่เหลืออยู่จากการโยนจะต้องถูกตั้งอยู่ในจุดที่ผิด และไม่สามารถที่จะขอตั้งในจุดที่ถูกต้องได้
12. พิน – กระดอน (pins – rebound)
13. พินที่กระดอนกลับมาตั้งอยู่ จะถือว่า เป็นพินที่ตั้งเหลืออยู่ไม่อนุญาตให้คะแนน (pins-may not be conceded)
14. ไม่อนุญาตการยกพิน และให้นับคะแนนพินที่ล้มลงหรือเคลื่อนออกจากพื้นผิดเลนในการโยนที่ถูกต้องเท่านั้น
15. พิน – การเปลี่ยนพิน (pins – replacement)
16. ในกรณีที่มีพินหนึ่งพินใดแตกหัก เสียหายอย่างมากในระหว่างเกม จะต้องหาพินที่มีรูปแบบ น้ำหนักและสภาพใกล้เคียงกับพินที่เหลืออยู่ในชุดให้มากที่สุดเท่าที่จะเป็นไปได้โดยทันที เจ้าหน้าที่ดำเนินการแข่งขันจะทำหน้าที่เป็นผู้ตัดสินเกี่ยวกับการเปลี่ยนพิน ในทุกกรณี

ลูกไม่นับคะแนน (dead ball)

1. การโยนที่จะไม่นับคะแนนและไม่นับว่าเป็นการโยน มีดังนี้
2. หลังจากการโยนลูกไปแล้ว (ก่อนการโยนลูกต่อไป) ได้มีการแจ้งว่าพินขาดหายไปจากที่ตั้ง เจ้าหน้าที่พินได้เข้ามามีส่วนเกี่ยวข้องกับพินที่ตั้งอยู่ก่อนที่ลูกโบว์ลิ่งจะถูกพิน
3. เจ้าหน้าที่ตั้งพินได้เข้ามาย้ายหรือเกี่ยวข้องกับพินที่ล้มอยู่ก่อนที่พินดังกล่าวจะหยุดเคลื่อนไหว เมื่อผู้เล่นโยนผิดเลนหรือผิดลำดับ หรือมีผู้เล่นคนใดคนหนึ่งในทีมทั้งสองทีมทีโยนอยู่เลนคู่กันขึ้นโยนผิดเลย
4. ในกรณีที่ผู้เล่นถูกทำให้เสียสมาธิหรือรบกวนทางกายภาพจากผู้เล่นอื่น ผู้ชมหรือสิ่งเคลื่อนไหวใดใดหรือโดยเจ้าหน้าที่ตั้งพิน ในขณะที่โยนลูกออกไป และก่อนที่การโยนจะถือว่าสมบูรณ์ ผู้เล่นจะสามารถที่จะเลือกว่าจะยอมรับผลการโยนลูกนั้น หรือว่าให้ถือเป็นลูกไม่นับคะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ในกรณีที่มีพินใดพินหนึ่งเคลื่อนที่หรือล้มลงหลังจากผู้เล่นได้ปล่อยลูกออกไปแล้วแต่ลูกยังไม่ถึงพิน
6. หากมีการแจ้งว่าการโยนเป็นลูกไม่นับคะแนน การโยนลูกนั้นจะไม่นับว่าเป็นคะแนน และจะต้องทำการโยนใหม่ในเลนที่ถูกต้อง
7. การโยนผิดเลน (bowling on wrong lane) ดังที่ได้กล่าวอ้างในกติกา 116 หากมีผู้เล่นคนหนึ่งคนใดโยนผิดเลน ลูกนั้นจะไม่นับเป็นคะแนน และจำต้องทำการโยนใหม่ในเลนที่ถูกต้อง

ผู้เล่นคนใดโยนผิดเล่น

ผู้เล่นคนหนึ่งจากทั้งสองทีมขึ้นโยนผิดเลน ในกรณีที่มีผู้เล่นมากกว่าหนึ่งคนในทีมเดียวกันโยนลูกผิดเลนให้ถือว่าเป็นการโยนที่ถูกต้องและให้เล่นเกมนั้นจนจบโดยไม่มีการเปลี่ยนแปลง และเกมต่อไปให้เริ่มในเลนที่ถูกต้อง

คำจำกัดความของฟาวล์ (definition of a foul)

เมื่อมีส่วนหนึ่งส่วนใดของร่างกายของผู้เล่นล่วงล้ำหรือเลยเข้าไปในเส้นฟาวล์เข้าไปโดนกับส่วนอื่นใดของเลน อุปกรณ์หรือสิ่งก่อสร้างของอาคาร ในระหว่างที่โยนหรือหลังจากการโยนอย่างถูกต้องการโยนลูกนั้นให้ถือว่าเป็นการโยน จนกระทั่งผู้เล่นคนเดิมหรือผู้เล่นอื่นขึ้นไปยืนอยู่บนแอพโพรชในตำแหน่งที่จะทำการโยนลูกต่อไป

การฟาวล์โดยจงใจ (deliberate foul)

ในกรณีที่ผู้เล่นคนใดทำฟาวล์โดยจงใจเพื่อประโยชน์ของตนเอง ให้ผู้เล่นนั้นได้รับคะแนนศูนย์ในเฟรมนั้นและไม่อนุญาตให้โยนลูกต่อไปในเฟรมนั้น

การฟาวล์ที่นับเป็นการโยน (foul count as ball delivered)

ในกรณีเมื่อเกิดฟาวล์ขึ้นให้นับลูกที่โดยผู้เล่นนั้น แต่ไม่ให้นับคะแนนพินที่ล้มลงในการโยนลูกนั้น หากเป็นการโยนลูกแรกในเฟรมนั้น ให้ตั้งพินที่ล้มทั้งหมดขึ้นใหม่ และให้ผู้เล่นโยนในลูกต่อไป การฟาวล์ที่เห็นโดยชัดเจน (apparent foul) ในกรณีที่ไฟฟาวล์ไม่ขึ้นและเจ้าหน้าที่จับฟาวล์ ไม่แจ้งว่าเป็นการฟาวล์ แต่ว่าให้ถือว่าเป็นการฟาวล์ หากว่ามีผู้หนึ่งผู้ใดดังนี้เห็นเหตุการณ์และทำการประท้วงหรือโต้แย้ง

1. หัวหน้าทีมของทั้งสองทีม หรือผู้เล่นอย่างน้อยหนึ่งคนคัดค้าน
2. เจ้าหน้าที่บันทึกคะแนน
3. เจ้าหน้าที่จัดการแข่งขัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอุทธรณ์ฟาวล์ (foul-appeal)

ไม่อนุญาตให้มีการอุทธรณ์ที่เกี่ยวกับการฟาวล์ใด ๆ นอกจาก

1. ในกรณีที่พิสูจน์ได้ว่าเครื่องจับฟาวล์ทำงานไม่ถูกต้อง
2. ในกรณีที่หลักฐานที่มีน้ำหนักพอที่จะเชื่อได้ว่าผู้เล่นไม่ได้ทำฟาวล์

ลูกชั่วคราว (provisional ball)

เมื่อมีการประท้วงเกี่ยวกับการทำฟาวล์หรือ การล้มของพินว่าถูกต้องหรือไม่ และเจ้าที่ที่ควบคุมการแข่งขันไม่สามารถตัดสินได้ในทันที ให้ผู้เล่นนั้นโยนลูกชั่วคราวหนึ่งลูก หรือเฟรมชั่วคราวหนึ่งเฟรม

1.ในกรณีที่มีข้อขัดแย้งเกิดขึ้นในการโยนลูกแรกของเฟรมใด หรือการโยนลูกที่สองของเฟรมสิบในกรณีที่สามรถทำสไต้รค์ได้

1.1ในกรณีที่มีข้อขัดแย้งและไม่สามารถที่จะตัดสินใจได้ว่าเป็นการฟาวล์หรือไม่ ให้ผู้เล่นโยนในเฟรมนั้น จนครบและให้ผู้เล่นนั้นโยนลูกชั่วคราวหนึ่งลูกโดยตั้งพินให้ครบตามจำนวนพินที่มีอยู่ในลูกนั้น

1.2ในกรณีที่มีข้อขัดแย้งเกี่ยวกับพินล้มว่าถูกต้องหรือไม่ ให้ผู้เล่นโยนจนจบเฟรมนั้นแล้วให้โยนลูกชั่วคราวหนึ่งลูกโดยให้ตั้งพินให้ถูกต้องครบตามจำนวนพินที่มีอยู่หากพินไม่ได้ล้มลง

1.3 ในกรณีที่มีข้อขัดแย้งว่าจะเป็นการโยนที่นับคะแนนหรือไม่ ให้ผู้เล่นโดยจนจบเฟรมนั้น และให้ผู้เล่นนั้นโยนชั่วคราวขึ้นมาหนึ่งเฟรม

2.ในกรณีที่มีข้อขัดแย้งในการเก็บสแปร์ หรือในการโยนลูกที่สามในเฟรมสิงไม่จำเป็นต้องโยนลูกชั่วคราวนอกจากมีการโต้แย้งว่าจะเป็นลูกไม่นับคะแนนหรือไม่ ให้มีการโยนลูกชั่วคราวโดยให้มีการตั้งพินให้ครบก่อนที่จะมีโต้แย้ง

ลูกโบว์ลิ่ง – การเปลี่ยนแปลงผิดลูกโบว์ลิ่ง (bowling ball-altering surface)

1. ห้ามให้มีการปรับเปลี่ยนพื้นผิวลูกโบว์ลิ่งในระหว่างการแข่งขันที่ได้รับการรับรองโดยสหพันธ์ฯ การแข่งขันให้รวมไปถึงเกมทั้งหมดในกลุ่มนั้น รวมถึงการซ้อมก่อนการเริ่มแข่งขันของกลุ่มนั้น หากมีการฝ่าฝืนในกติกาข้อ 211 หรือ 321 ให้นำลูกโบว์ลิ่งลูกนั้นออกจากเกมการแข่งขันของกลุ่มนั้น
2. ห้ามทำให้แอพโพรชเสียสภาพ (approaches must not be defaced)
3. ห้ามมิให้ผู้ใดติดเครื่องหมายหรือสิ่งอื่นใดไว้ที่บริเวณแอพโพรช ซึ่งอาจจะทำให้เกิดเป็นภัยทำให้เสียรูป หรือทำให้ผู้อื่นไม่สามารถใช้แอพโพรชได้ในสภาพปกติ ห้ามมิให้ใช้วัตถุ เช่น แป้งลื่น หินขัด ยางไม้ พื้นรองเท้าหรือส้นรองเท้าทำจากยางอ่อน ซึ่งอาจจะขัดและเปลี่ยนสภาพของแอพโพรชให้ผิดไปจากสภาพปกติความผิดพลาดใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล นายธีรวัฒน์ ลือวงษ์

วัน เดือน ปีเกิด 6 กันยายน 2529

ที่อยู่ 25/2 ซอยโรงเหล็ก ถนนตะกั่วทุ่ง ตำบลตลาดใหญ่ อำเภอเมือง จังหวัดภูเก็ต 83000

ประวัติการศึกษา 2551 คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อ-นามสกุล นายภูเบศ กร้ำมาตร

วัน เดือน ปีเกิด 18 ธันวาคม 2529

ที่อยู่ 158/18 หมู่บ้านพูนสินธานี 1 ถนนราษฎร์พัฒนา แขวงคลองสองต้นนุ่น เขตลาดกระบัง กรุงเทพฯ 10520

ประวัติการศึกษา 2551 คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้