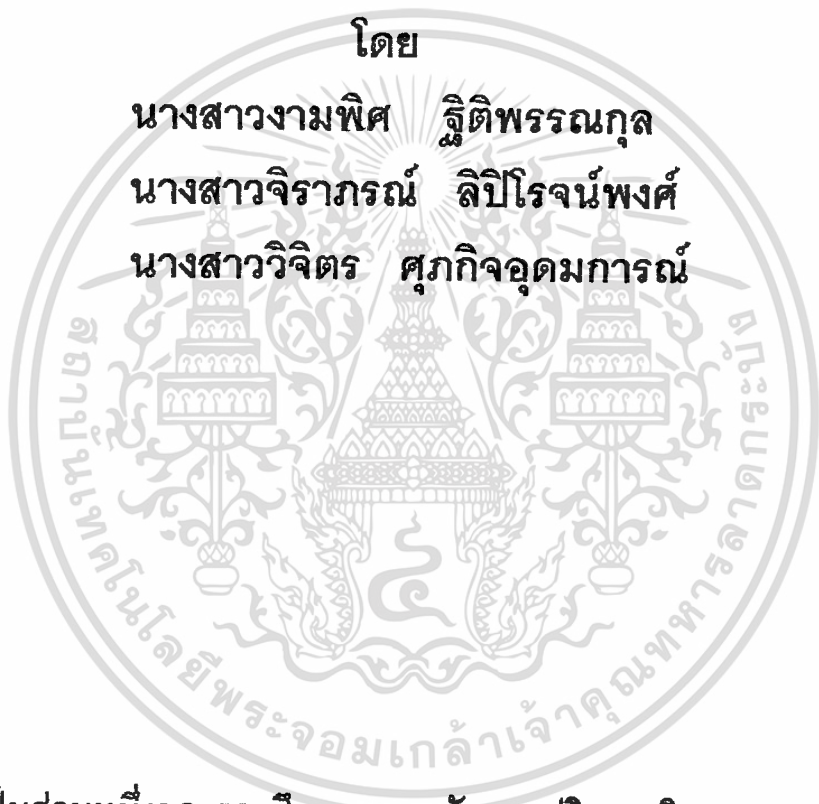


การใช้ประยุกต์ใช้คอมพิวเตอร์ และระบบมิดีเพื่อดนตรีไทย
(MIDI TRANSLATION)

โดย

นางสาวงามพิศ ฐิติพรรณกุล
นางสาวจิราภรณ์ ลิปิโรจน์พงศ์
นางสาววิจิตร ศุภกิจอุดมการณ์



ปฏิญานีพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
ภาควิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง
ปีการศึกษา 2536

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปฏิทินพันธปีการศึกษา 2536

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การประยุกต์ใช้ประยุกต์ใช้ระบบคอมพิวเตอร์ และมีดีเพื่อดนตรีไทย
(MIDI TRANSLATION)

ผู้จัดทำ

- 1) นางสาวงามพิศ ฐิติพรรณกุล 33100045
- 2) นางสาวจิราภรณ์ ลิปิโรจน์พงศ์ 33100059
- 3) นางสาววิจิตร ศุภกิจอุดมการณ์ 33100341


ดร. รัตติกร วรากุลศิริพันธ์ อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีของระบบมิดี	3
บทที่ 3 การออกแบบและการสร้าง	12
3.1) ฮาร์ดแวร์	13
3.1.1) ฮาร์ดแวร์ในส่วนภาครับสัญญาณ	13
3.1.2) ฮาร์ดแวร์ส่วนมิดีคาร์ต	19
3.2) ซอฟต์แวร์	
3.2.1) ซอฟต์แวร์การทำงานของโครงงาน	23
3.2.2) ซอฟต์แวร์ส่วนที่ทำงานร่วมกับฮาร์ดแวร์	25
บทที่ 4 การทดลองและผลการทดลอง	27
4.1) ปัญหาที่เกิดขึ้น	28
บทที่ 6 สรุปและวิจารณ์	30
กิจกรรมประกาศ	31
หนังสืออ้างอิง	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้ระบบคอมพิวเตอร์ และมีดีเพื่อดนตรีไทย

(MIDI TRANSLATION)

งามพิศ ฐิติพรรณกุล 33100045

จิราภรณ์ ลิปิโรจน์พงศ์ 33100059

วิจิตร ศุภกิจอุดมการณ์ 33100341

อาจารย์ที่ปรึกษา

ดร. รัตติกร วรากุลศิริพันธ์

ปีการศึกษา 2537

บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้ นำเสนอการนำคอมพิวเตอร์มาช่วยงานทางด้านดนตรี โดยการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับเครื่องดนตรี ซึ่งโครงการนี้จะทำการเชื่อมต่อกับเครื่องดนตรีไทยมีวัตถุประสงค์ เพื่อการบันทึกการเล่นดนตรีไว้กับเครื่องไมโครคอมพิวเตอร์ ในโครงการนี้ได้ใช้เครื่องดนตรีไทยประเภทระนาดซึ่งเป็นเครื่องดนตรีไทยที่จัดอยู่ในกลุ่มเครื่องตี การใช้เครื่องดนตรีในกลุ่มนี้จะง่ายในการตรวจจับสัญญาณการตี เพื่อนำมาวิเคราะห์สัญญาณได้ใกล้เคียงกับการเล่นจริงได้มากขึ้น ส่วนสำคัญของโครงการนี้ ก็คือการทำให้ระนาดนี้เสมือนเป็นเครื่องดนตรีอิเล็กทรอนิกส์ที่จะสามารถสื่อสารกับเครื่องดนตรีอิเล็กทรอนิกส์อื่นๆ ได้ โดยในการที่จะสื่อสารกับเครื่องดนตรีอื่นๆได้นั้น จะต้องใช้ระบบมีดีมาเป็นสื่อกลางในการสื่อสาร ในการทำให้เครื่องดนตรีไทยเป็นเครื่องดนตรีอิเล็กทรอนิกส์นี้ จะได้มีการกล่าวอย่างละเอียดต่อไป ในปฏิญานิพนธ์ฉบับนี้

ABSTRACT

In this thesis illustrate how to use the computer with Thai-instrument by linking. The objective for this project is storing the data from the Thai instrument to the microcomputer. In this project we use 'RANARD', Thai instrument type of striking. Since it easy to detect the striking signal. The importion part is 'How to do RANARD become the electronic instrument for communicate with the other electronic instrument. The communication will be success, we have to use the midi system to be the medium. The detail, How to do the Thai instrument to be the electronic instrument, will show in the following.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ปัจจุบันมีการนำไมโครคอมพิวเตอร์ มาใช้กันมากขึ้นในงานเกือบทุกๆแขนง หรือในชีวิตประจำวันก็ตาม และรวมไปถึงระบบมิตีซึ่งเป็นภาษาคอมพิวเตอร์ตัวหนึ่ง ที่มีไว้สำหรับการติดต่อระหว่างเครื่องดนตรีด้วยกัน หรือการติดต่อระหว่างเครื่องดนตรีกับคอมพิวเตอร์ในเรื่องของเครื่องดนตรีนั้นโครงการนี้ได้้นำเครื่องดนตรีไทยมาใช้

ในอดีตมาจะมีการใช้ดนตรีกับเครื่องไมโครคอมพิวเตอร์นี้ ซึ่งส่วนมากจะใช้กับเครื่องดนตรีอิเล็กทรอนิกส์แทบทั้งสิ้น เช่น เครื่องดนตรีประเภทคีย์บอร์ด ในปัจจุบันเครื่องดนตรีอิเล็กทรอนิกส์จะมีช่องในการติดต่อกับคอมพิวเตอร์ได้เลย โดยการผ่านทางช่องสัญญาณมิตี หรือจะมีในเครื่องดนตรีอื่น ๆ อีก ซึ่งในส่วนนี้ได้มีผู้สนใจและพัฒนาไปกันอย่างมากแล้ว (รายละเอียดเกี่ยวกับเรื่องมิตี จะได้มีการอธิบายในเรื่องทฤษฎีของมิตีในส่วนตัวต่อไป)

โครงการนี้ได้มีการคิดนำเอาเครื่อง ไมโครคอมพิวเตอร์มาใช้กับเครื่องดนตรีไทยกันบ้าง- แต่จำเป็นต้องมีการวิจัยและค้นคว้า เพื่อให้เครื่องดนตรีไทยสามารถสื่อสารกับเครื่องดนตรีอื่น ๆ ในโลกนี้ได้โดยใช้ภาษามิตีเป็นภาษามาตรฐานสื่อกลางในการสื่อสาร และในทางตรงกันข้ามเราจำเป็นต้องสร้างระบบคอมพิวเตอร์ที่สร้างไว้เพื่อดนตรีไทย แล้วเครื่องดนตรีทั่วโลกก็จะสามารถติดต่อกับระบบดนตรีไทยได้ โดยผ่านภาษาสื่อกลางที่เรียกว่า "มิตี" นั้นนั่นเองโดยเรายังคงศิลปะอันละเอียดอ่อนของเครื่องดนตรีนี้ให้มากที่สุดด้วย โดยโครงการนี้จะใช้เครื่องดนตรีไทยประเภทระนาด ซึ่งเป็นเครื่องดนตรีที่ไม่ใช่เครื่องดนตรีไฟฟ้า (acuostic) ซึ่งวัตถุประสงค์หลักของโครงการนี้จะเป็นการทำให้เครื่องดนตรีไทยนี้เป็นเครื่องดนตรีไฟฟ้า ที่มีระบบมิตีในตัวเองได้ หลักการและวิธีการต่าง ๆ ได้นำเสนอต่อไปดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์

ประโยชน์ที่ได้รับจากระบบคอมพิวเตอร์ดนตรีในปัจจุบัน

- ทำให้ดนตรีง่ายต่อการเรียนรู้และเข้าใจ เนื่องมาจากการที่มีดังกล่าวมาใช้
- ทำให้ผู้ศึกษาด้านดนตรีสามารถได้ยินเสียงดนตรี และเห็นโน้ตได้ก่อนที่จะต้องเล่นเครื่องดนตรีดังกล่าวเป็น
- เกิดโปรแกรมการสร้างโน้ตทางดนตรี ซึ่งง่ายต่อการทำมากเมื่อเปรียบเทียบกับการเล่นด้วยมือ และการถอดจากสื่อที่ใช้เก็บ เช่น เทป ซีดี
- การจัดเก็บดนตรีในรูปของสื่อทางคอมพิวเตอร์ ทำให้สามารถนำดนตรีไปประยุกต์ใช้งานในสาขาอื่น ๆ ได้อย่างกว้างขวาง

ประโยชน์ที่จะได้รับจากระบบคอมพิวเตอร์สำหรับดนตรีไทย

- สามารถทำให้เครื่องดนตรีไทยสื่อสารกับเครื่องดนตรีสากลทั่วโลกได้
- เครื่องดนตรีทั่วโลกสามารถประพาดตัวเป็นเครื่องดนตรีไทย เพื่อเข้าร่วมเล่นกับเครื่องดนตรีไทยได้
- เกิดระบบคอมพิวเตอร์ดนตรีไทย ที่เครื่องดนตรีสากลทั่วโลกสามารถนำระบบโปรแกรมหาดังกล่าว ไปใช้กับเครื่องดนตรีในประเทศของตนได้ และเป็นการเผยแพร่ดนตรีไทยออกไปทั่วโลก
- ทำให้เกิดการจัดเก็บดนตรีไทยในรูปแบบคอมพิวเตอร์ ซึ่งคนไทยรุ่นหลังสามารถนำมาศึกษาใหม่ได้ทุกเมื่อ เพราะทุกขบวนการในการเล่นดนตรีขณะนั้น ๆ จะถูกบันทึกไว้ทุกขั้นตอน ซึ่งไม่เป็นเพียงแค่การบันทึกเสียงธรรมดา ๆ เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

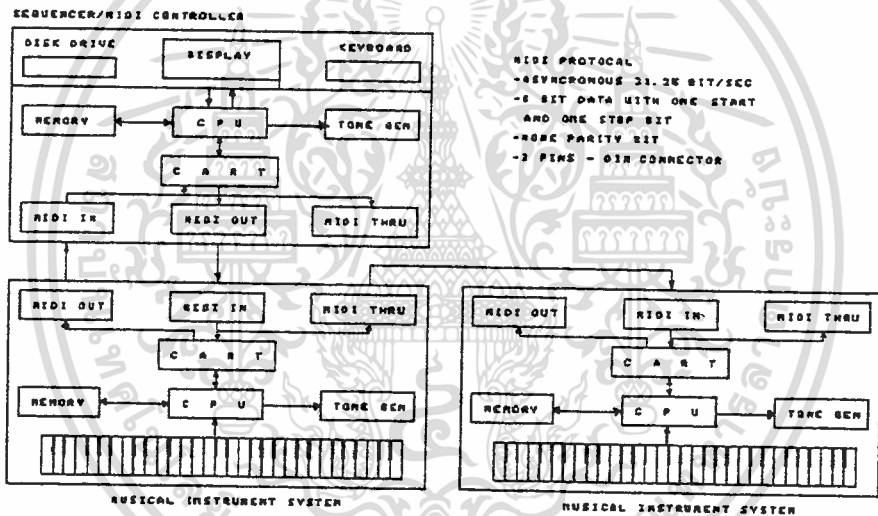
บทที่ 2

ทฤษฎีของระบบมิดี

หลักการ และ ข้อกำหนดในระบบมิดี

หลักการ

มิดี (MIDI) ย่อมาจาก Musical Instrument Digital Interface เป็นระบบมาตรฐานสำหรับการสื่อสารข้อมูลระหว่างเครื่องดนตรี โดยจะมีการสื่อสารกัน แบบอซิงโครนัส (asynchronous) ซึ่งเขียนแผนภาพแสดงการทำงานได้ดังนี้



รูปที่ 2 แผนภาพแสดงข้อกำหนดในการเชื่อมต่อ

จากรูป อุปกรณ์ควบคุม (controlling instrument) เช่น คอมพิวเตอร์ สามารถควบคุมการทำงานของเครื่องดนตรีได้ โดยคอมพิวเตอร์จะส่งรหัสมิติไปยังคีย์บอร์ดตัวที่ 1 และใช้ออปโตไดโอด (optoisolator) เป็นตัวคัปปลิงสัญญาณ ส่วน UART จะเปลี่ยนข้อมูลแบบขนานให้เป็นแบบอนุกรม (และเปลี่ยนจากแบบอนุกรมให้เป็นแบบขนานด้วย)

ข้อมูลที่ได้อาจจะเข้ามายัง ส่วนควบคุมภายในเครื่องดนตรี (micro controller) เพื่อตีความ แล้วส่งให้คีย์บอร์ดทำงานตามรหัสมิติที่ได้รับ ข้อมูลจากออปโตไดโอดส่วนหนึ่งจะไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านไปยังบัฟเฟอร์แล้วส่งออกไปเรียกว่า มิดีทรู (MIDI thru) ซึ่งก็คือการลอกแบบข้อมูลที่ได้รับมานั่นเอง ข้อมูลส่วนนี้จะถูกส่งไปที่คีย์บอร์ด 2 จะเห็นได้ว่า คีย์บอร์ดตัวแรกกับตัวที่สองจะติดต่อกันได้ โดยคีย์บอร์ดตัวแรกจะควบคุมคีย์บอร์ดตัวที่สอง ซึ่งอาศัยข้อมูลที่ได้มาจากคอมพิวเตอร์ ถ้าไม่อาศัยข้อมูลจากคอมพิวเตอร์ เมื่อมีการเล่น คีย์บอร์ด 1 (ใช้คนเล่น) ส่วนควบคุมภายในคีย์บอร์ด 1 จะส่งรหัสมิติออกไป และคอมพิวเตอร์จะได้รับข้อมูลนี้ แล้วนำมาทำขบวนการต่างๆ ได้ เช่น เก็บข้อมูลนี้ไว้ในดิสก์ ในลักษณะนี้จะคล้ายกับเทปบันทึกเสียงแต่แทนที่จะเก็บเป็นสัญญาณเสียง คอมพิวเตอร์จะเก็บในรูปของรหัสมิติ และเมื่อต้องการเล่นซ้ำ

จะเห็นว่าเมื่อเครื่องดนตรีสามารถสื่อสารกันภาษาได้มาตรฐานเดียวกัน แล้วเราสามารถนำไปประยุกต์ใช้งานได้อย่างกว้างขวาง เช่น

1. ควบคุมเครื่องดนตรีได้ทีละหลายๆ เครื่อง เนื่องจากเครื่องดนตรีทุกเครื่องสื่อสารด้วยภาษาเดียวกัน
2. สามารถส่งรหัสมิติไปควบคุมกลองอิเล็กทรอนิกส์ (Drum machine setups) เพื่อให้มีจังหวะตรงกับเครื่องดนตรีอื่นๆ (synchronized) ได้
3. อุปกรณ์แต่งเสียง (sound effect) ต่างๆ สามารถควบคุมได้ด้วยรหัสมิติ
4. ใช้ในการสร้างเสียง (voice quality) ของเครื่องดนตรีประเภท ซินธิไซเซอร์- (voice editing software)
5. ข้อมูลจากเครื่องดนตรีเครื่องหนึ่งอาจนำไปใช้กับอีกเครื่องได้ (Data transfer)
6. ถ้าใช้การเก็บรหัสมิติ แทนที่จะเก็บเป็นสัญญาณเสียง เราสามารถเล่นซ้ำได้ดังที่กล่าวไปแล้ว ซึ่งจะคล้ายกับการบันทึกเสียง (Tepeless recording)
7. ประโยชน์จากการสื่อสาร เช่นสามารถส่งข้อความมิติผ่านโมเด็ม (modem) ได้ ดังนั้นแม้จะอยู่กันคนละประเทศนักดนตรี ก็ยังสามารถส่งดนตรีของเขาไปที่ห้องอัดเสียง ได้ทางโทรศัพท์โดยไม่มีความเพี้ยนของเสียงดนตรีและเนื่องจากการส่งเป็นบีต

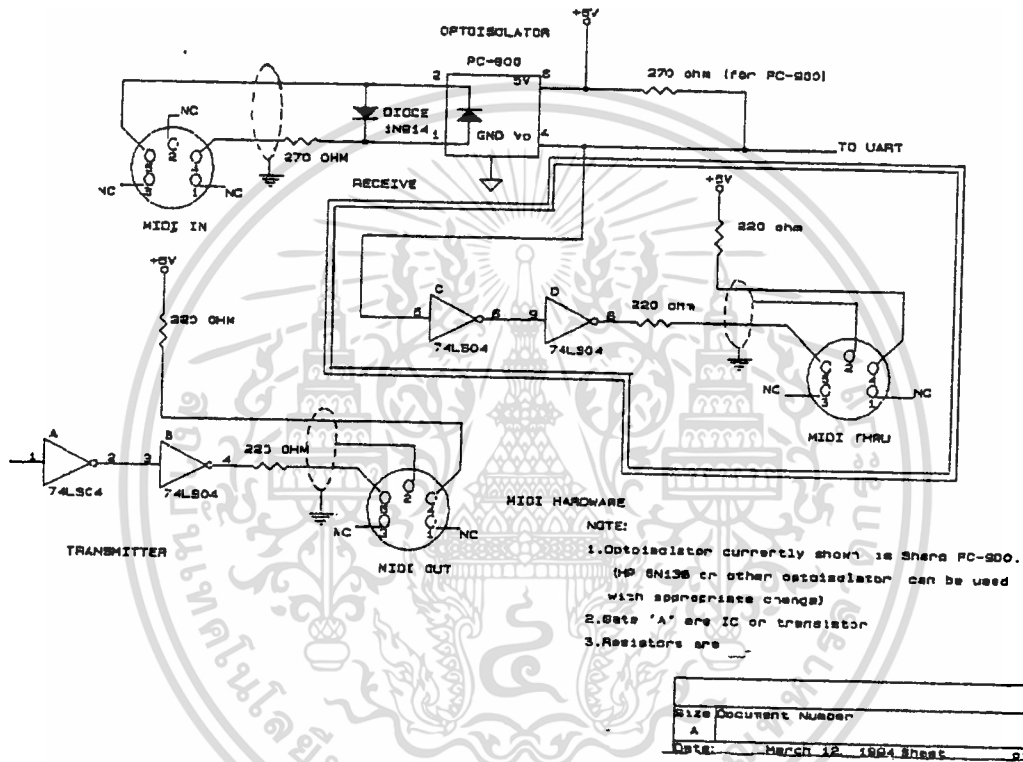
สำหรับเครื่องดนตรีที่ไม่ใช้ไฟฟ้า (acoustic) ก็สามารถใช่วงกับเครื่องดนตรีอื่นด้วยระบบมิติได้ ถ้าเราสามารถทำให้มันส่งข้อมูลออกไปตามข้อกำหนดของระบบมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อกำหนดของระบบมิดี

- ข้อกำหนดโดยทั่วไป

มิดีเป็นมาตรฐานที่จะให้ ซินธิไซเซอร์ (synthesizer) ,ซีควเอนเซอร์ (sequencer) คอมพิวเตอร์, อุปกรณ์สร้างจังหวะ ฯลฯ สามารถติดต่อสื่อสารกันได้



รูปที่ 1 แสดงรูปแบบการสื่อสารของเครื่องดนตรี

อุปกรณ์ที่ใช้ระบบมิดี มักจะรับ-ส่งข้อความในระบบมิดีได้ โดยทางด้านรับจะรับข้อความ (message) ที่เป็นรหัสมิดีแล้วทำงานตามความหมายของข้อความนั้น อุปกรณ์ทางฮาร์ดแวร์ที่สำคัญประกอบด้วย ออฟดีไดโวลิตเซอร์, Universal Asynchronous Receiver Transmitter (UART) ส่วนทางด้านส่งก็จะส่งรหัสมิดีออกไป โดยมี UART และตัวขับสัญญาณ (line driver) เป็นองค์ประกอบสำคัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฮาร์ดแวร์

สื่อสารแบบอนาล็อก โดยใช้ความเร็วในการ รับ-ส่งที่ 31.25 kBaud มีบิตเริ่มและบิตหยุด (start bit, stop bit) อย่างละ 1 บิต, บิตข้อมูลมี 8 บิต รวมทั้งหมด 10 บิต ต่อข้อมูล 1 ไบต์ ซึ่งใช้เวลาส่ง 320 ไมโครวินาที มีวงจรเป็นดังรูป MIDI HARDWARE

จากรูป ประแสในรูป = 1.5 mA โดยที่ โวลจิก 0 จะมีกระแสไหล ทางด้านรับควรมีออฟได้อะไหล่เตอร์เป็นตัวคัปปลิงสัญญาณ ซึ่งเป็นชนิดทำงานที่ความเร็วสูง เช่น Sharp PC-900 เวลาในการสวิตช์ ควรน้อยกว่า 2 ไมโครวินาที

- อุปกรณ์สำหรับเชื่อมต่อ (connector)

เป็นแบบ DIN 5 ขา (แจ๊คกลม) ตัวเมีย ขาที่ใช้คือขา 5,4 ส่วนขา 2 ต่อกราวนด์ - นอกจากนั้นไม่ใช่ ปลั๊กไอวีเฉยๆ

ความยาวของสาย (MIDI cable) ไม่เกิน 15 เมตร หรือ 50 ฟุต และควรมีการพันเกลียว (twist) เพื่อลดสัญญาณรบกวน

ชื่อของคอนเนคเตอร์ MIDI IN (ชื่อข้อมูล), MIDI OUT (ข้อมูลออก) และ MIDI THRU ซึ่งเป็นข้อมูลขาออกที่ได้มาจากการก๊อปปี้ข้อมูลมีด้นอื่น โดยตรง

- รูปแบบของข้อมูล (data format)

แต่ละข้อความในระบบมีดี มีหลายไบต์โดยประกอบด้วยไบต์สถานะ (status byte) - ตามด้วยไบต์ข้อมูล (data byte) อีก 1 หรือ 2 ไบต์ ยกเว้นข้อความเกี่ยวกับเวลา (real-time message) และข้อความพิเศษ (exclusive message) ซึ่งจะกล่าวถึงภายหลัง

ประเภทของข้อความ จะแบ่งออกเป็น 2 ประเภทใหญ่ๆ คือ แชนแนล กับ ระบบ

- ข้อความเกี่ยวกับแชนแนล

ในไบต์สถานะจะใช้ 4 บิต เป็นตัวกำหนดแชนแนลของไบต์สถานะ (สามารถกำหนดได้ 16 แชนแนล) ทางด้านรับจะตอบสนองเฉพาะไบต์สถานะ ที่มีแชนแนลตรงกับแชนแนลของตัวเองเท่านั้น - ถ้าไม่ตรงกันจะไม่สนใจ

ข้อความเกี่ยวกับแชนแนล มี 2 ประเภท

Voice - ใช้ควบคุมเสียง ข้อความเกี่ยวกับเรื่องเสียง จะถูกส่งผ่านทางแชนแนลเสียง (voice channel)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
Mode - ใช้กำหนดลักษณะการตอบสนองต่อข้อความ เกี่ยวกับเรื่องเสียงของเครื่องดนตรี
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะส่งมาทาง แชนแนลพื้นฐาน (Basic channel) ของเครื่องดนตรี

- ข้อความเกี่ยวกับระบบ

ข้อความเกี่ยวกับระบบจะไม่มีกำหนดแชนแนล ดังนั้นไม่ว่าด้านรับจะเป็นแชนแนลไหนก็จะตอบสนองต่อข้อความเกี่ยวกับระบบเสมอ

ข้อความเกี่ยวกับระบบมี 3 แบบ

1. ข้อความทั่วไป (common message) เป็นข้อความสำหรับทุกอุปกรณ์ในระบบ
2. ข้อความเกี่ยวกับเวลา จะมีแต่ไบต์สถานะเท่านั้น ไม่มีไบต์ข้อมูล สำหรับข้อความประเภทนี้จะส่งเมื่อไหร่ก็ได้ แม้แต่ส่งคั่นระหว่างไบต์ของข้อความอื่น
3. ข้อความพิเศษ (exclusive message) มีหลายไบต์ข้อมูล และจะจบด้วยไบต์ที่เรียกว่า End Of Exclusive (EOX) หรือไบต์สถานะอื่น ๆ เป็นข้อความที่ใช้เฉพาะกับอุปกรณ์ของผู้ผลิตที่กำหนดไว้ในรหัสประจำเครื่องของแต่ละบริษัทผู้ผลิตหรือ Identification Codes (ID codes) เท่านั้น ถ้าอุปกรณ์ทางด้านรับมีรหัสประจำเครื่องไม่ตรงกับข้อความพิเศษที่ได้มันจะไม่สนใจไบต์ข้อมูลที่ตามมา ผู้ผลิตแต่ละรายสามารถกำหนดรูปแบบ ความหมายข้อมูลของตนเองได้ โดยส่งเป็นข้อความพิเศษ ที่มีรหัสประจำเครื่องของตน (ซึ่งเป็นรหัสเดี่ยว

- ประเภทของข้อมูล (DATA TYPES)

ไบต์สถานะ มี 8 บิต โดยมีบิตที่มีนัยสำคัญสูงสุด (MSB) ถูกเซ็ท (เป็น 1) ไบต์สถานะจะเป็นตัวบอกถึงประเภทของข้อความ และบอกให้รู้ว่าไบต์ข้อมูลที่ตามมาจะเอาไปใช้เพื่ออะไร

สถานะที่กำลังทำงาน (running status) สำหรับข้อความเกี่ยวกับเสียง และโหมดเท่านั้น เมื่อด้านรับได้รับไบต์สถานะ และทำขบวนการต่าง ๆ ด้านรับจะยังคงอยู่ในสถานะนั้นจนกว่าจะได้รับไบต์สถานะใหม่ที่ไม่เหมือนเดิม ดังนั้น ถ้ามีการส่งไบต์สถานะเดิมเข้ามาอีกครั้ง อาจทำเมื่อต้องการแก้ไขไบต์ข้อมูลที่ผิดโดยส่งใหม่ ภายหลังสถานะที่กำลังทำงาน

สถานะที่ไม่สามารถทำงานได้ (Unimplement Status) ทางด้านรับจะไม่สนใจไบต์สถานะที่มันไม่สามารถทำงานตามได้ และจะไม่สนใจไบต์ข้อมูลที่ตามมา

สถานะที่ไม่มีการกำหนดความหมาย (Undefined Status) จะไม่มีการใช้ไบต์สถานะที่ไม่มี ความหมาย ควรระวังในขณะที่เปิดเครื่อง หรือปิดเครื่อง อาจส่งข้อมูลผิดพลาดได้

ไบต์ข้อมูล จะถูกส่งตามหลังไบต์สถานะ โดยมีไบต์ข้อมูลประมาณ 1-2 ไบต์ (ยกเว้นข้อความเกี่ยวกับเวลา) ไบต์ข้อมูลมี 8 บิต โดยบิตที่มีนัยสำคัญต่ำสุดถูกรีเซ็ท (เป็น 0) จำนวนและขอบเขต (ค่าที่มีได้) ของไบต์ข้อมูล จะเป็นไปตามที่ไบต์สถานะกำหนด ในแต่ละไบต์สถานะจะส่งไบต์ข้อมูลจำนวนที่ถูกต้องตามไปเสมอ การตอบสนองต่อข้อความ จะมีได้ก็ต่อเมื่อด้านรับได้รับไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหากมีการนำไปใช้

ไปที่ข้อมูลตามต้องการ . แล้วทางด้านรับจะไม่สนใจกับไปที่ข้อมูลที่ตามหลังไปที่สถานะที่มันไม่รู้จัก
-โหมดต่างๆ ในแต่ละแชนแนล (channel Mode)

ซินธิไซเซอร์จะประกอบด้วย ส่วนที่สร้างเสียง (sound generator) เสียงที่เกิดขึ้นเรียกว่า voice การกำหนดเสียง เป็นขบวนการเกี่ยวกับข้อมูลพวกการเล่นโน้ต (note on) กับการเล่นโน้ต (note off) จากคีย์บอร์ดซึ่งเสียงที่มีโน้ตต่างๆ จะถูกเล่นได้ในจังหวะที่ถูกต้อง

ข้อความเกี่ยวกับโหมด (mode message) ใช้กำหนดลักษณะการสร้างเสียงของเครื่องดนตรีประเภทซินธิไซเซอร์ ได้แก่ Omni (On/Off), Poly และ Mono

Mono off (Poly on) ซินธิไซเซอร์ที่อยู่ในโหมดนี้สามารถสร้างเสียงได้พร้อม ๆ กันทีละหลายๆ เสียง (Poly phonic) ซึ่งทั้งนี้จะขึ้นอยู่กับความสามารถในการสร้างเสียง ของตัวซินธิไซเซอร์ด้วย

สำหรับ Omni ถ้า Omni on ตัวรับจะสามารถรับข้อความเกี่ยวกับเสียงได้ จากทุกแชนแนลเสียงโดยไม่มีความต่างกันของแชนแนล ถ้า Omni off ด้านรับจะรับเฉพาะข้อความเกี่ยวกับเสียงที่มีแชนแนลตรงกับของตนเท่านั้น

ทางด้านรับที่ถูกกำหนดเป็นแชนแนลพื้นฐานจะมีลักษณะการทำงานได้ 4 โหมดดังนี้

โหมด 1. Omni On , Poly

โหมด 2. Omni On , Mono

โหมด 3. Omni off , Poly

โหมด 4. Omni off , Mono

ทางด้านส่งและรับจะทำงานได้ที่ละ 1 โหมดเท่านั้นในเวลาเดียวกัน และตามปกติจะส่งและรับในโหมดเดียวกัน

ด้านรับจะรับรู้ข้อความเกี่ยวกับ โหมดทางแชนแนลพื้นฐานเท่านั้น ส่วนข้อความเกี่ยวกับเสียงอาจได้รับทางแชนแนลพื้นฐานหรือ แชนแนลอื่นๆ ก็ได้ซึ่งเรียกรวมกันว่าแชนแนลเสียง

โดยทั่วไปมักกำหนดให้แชนแนลแรก (0) เป็นแชนแนลพื้นฐาน

-ความหมายของข้อความต่างๆ ในระบบมีดังนี้

1. ข้อความที่เกี่ยวกับเสียงในแต่ละแชนแนล (channel voice message)

การเล่นโน้ต (note on)

เป็นข้อความที่บอกให้รู้ถึงการเล่นโน้ต (เช่นการเอานิ้วกดไปที่คีย์บอร์ด) มี 3 ไบท์
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1. ไบท์สถานะคือ 1001 nnnn (ฐานสี่) โดยที่ nnnn คือหมายเลขแชนแนล (0-15)
2. ไบท์ข้อมูลแรกจะบอกให้รู้ว่าโน้ตไหนที่ถูกกด (note number) มีค่าได้ตั้งแต่ 0-127 โดย 0 จะเป็นโน้ตที่ต่ำที่สุด 127 เป็นโน้ตที่สูงที่สุด แต่โดยทั่วไปเครื่องดนตรีจะรับรู้โน้ตได้น้อยกว่า เช่น ซินธิไซเซอร์ที่มี 5 ออกเตฟ จะรับรู้หมายเลขโน้ตได้ตั้งแต่ 36-96 หมายเลขโน้ตของโดกลาง (middle C) คือ 60

การเลิกเล่นโน้ต (note off)

เป็นข้อความที่บอกให้รู้ถึงการเลิกสร้างเสียง (เช่นเกิดขึ้นในขณะที่กดนิ้วขึ้นจากคีย์) มี 3 ไบท์

1. ไบท์สถานะ คือ 1000nnnn (ฐานสอง) โดยที่ nnnn คือ หมายเลขแชนแนล
2. ไบท์ข้อมูลแรกบอกให้รู้ว่าโน้ตตัวไหนที่จะให้เลิกสร้างเสียง (key number)
3. ความเร็วที่ปล่อยคีย์ (key off release velocity) เป็นไบท์ข้อมูลที่บอกถึง ความเร็วในการปล่อยคีย์ ในกรณีที่อุปกรณ์ไม่มีความเร็วของคีย์จะให้ไบท์นี้มีค่าเป็น 64

เมื่อเครื่องดนตรีได้รับข้อความเกี่ยวกับการเล่นโน้ต มันจะผลิตเสียงค้างอยู่อย่างนั้น จนกว่าจะได้รับข้อความเกี่ยวกับการเลิกเล่นโน้ตของโน้ตตัวนั้น มันถึงจะหยุดทำเสียง ดังนั้นเมื่อมีการส่งข้อความเกี่ยวกับการเล่นโน้ตออกไปแล้ว จะส่งข้อความเกี่ยวกับการเลิกเล่นโน้ตตามมาด้วยทุกครั้ง มิฉะนั้นจะเกิดเสียงค้าง

การกำหนดความดันของคีย์แบบโพลีโฟนิก (Polyphonic Key pressure)

คีย์บอร์ดในระบบมีคีย์บางเครื่องไม่เพียงแต่ตอบสนองความเร็วของคีย์ได้เท่านั้นยังตอบสนองน้ำหนัก(ความดัน) ของนิ้วที่กดลงไปอีกหลังจากที่กดคีย์ลงไปสุดแล้ว มี 3 ไบท์

1. ไบท์สถานะ คือ 1010nnnn
2. หมายเลขของโน้ต
3. ความดันของคีย์ (key pressure value) มีค่าตั้งแต่ 0-127 โดย 0 หมายถึง ไม่มีความดันเลข และ 127 จะมีความดันสูงสุด

ประโยชน์ของโพลีโฟนิกคีย์เพรชเชอร์ คือ จะให้ความรู้สึกของเครื่องเป่า เช่น ทรัมเป็ต-เมื่อตั้งเสียงของซินธิไซเซอร์เป็นเสียงทรัมเป็ต การกดคีย์ลงไปอีกหลังจากสุดคีย์แล้วจะมีลักษณะเหมือนการที่เราเป่าลมเข้าไปในทรัมเป็ตอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดความดันรวม (Overall Pressure)

ต่างกับโพลีโฟนิกคีย์เพรชเซอร์ตรง ที่ไม่สามารถบอกความดันแยกกันของแต่ละคีย์ที่ถูกกดได้แก่ จะบอกเป็นความดันเฉลี่ยของทุก ๆ คีย์ในเวลานั้น ๆ มีอยู่ด้วยกัน 2 ไบท์

1. ไบท์ข้อมูลคือ 1101nnnn (nnnn คือ หมายเลขแชนแนล)
2. ความดันของแชนแนล (channel pressure value) มีค่าตั้งแต่ 0-127 โดยที่ 0 จะไม่มี ความดัน และ 127 มี ความดันมากที่สุด

ใช้ในการปรับแต่งเสียง สำหรับข้อมูลเกี่ยวกับการควบคุมในปัจจุบัน ยังไม่มีมาตรฐานที่แน่นอนระหว่างแต่ละบริษัทผู้ผลิตมี 3 ไบท์

1. ไบท์สถานะ คือ 1011nnnn
2. หมายเลขตัวควบคุม (controller number) มีค่าตั้งแต่ 0-127 ดูความหมายได้จาก MIDI Implementation Sheet ของแต่ละบริษัทผู้ผลิต
3. ค่าที่จะให้แก่ตัวควบคุม (controller value)

- การเอื้อนเสียง (Pitch Bend)

เป็นลักษณะของการเอื้อนเสียงจากสูงไปต่ำ หรือจากต่ำไปสูงโดยที่ไม่ต้องเปลี่ยนตำแหน่งของคีย์ที่กด การเอื้อนเสียงมี 2 ไบท์

1. ไบท์สถานะคือ 1110nnnn
2. ค่าที่จะให้เอื้อน (pitch bend value)

ข้อมูลเกี่ยวกับโปรแกรมเสียง (Program Change) หรือบางที่เรียกว่า Program - Select จะบอกให้รู้จักการใช้โปรแกรมเสียงต่าง ๆ ในเครื่องดนตรี มี 2 ไบท์

1. ไบท์สถานะ คือ 1100nnnn
2. โปรแกรมเสียงที่ใช้ (selected program number) มีค่าตั้งแต่ 0-127

ข้อมูลเกี่ยวกับลักษณะการทำงานในแต่แชนแนล (Channel Mode Messages)

เป็นข้อความเกี่ยวกับโหมดในการทำงาน เป็นส่วนหนึ่งของข้อความเกี่ยวกับ การควบคุม การควบคุมคีย์บอร์ดแบบ โคลด/รีโมด (Local/Remote Keyboard Control) ใช้ในการ พ่วงเครื่องดนตรีเข้าด้วยกันโดยควบคุมจากเครื่องเดียว

- การรีเซ็ตระบบ (System Reset)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า เป็นข้อความที่บอกให้อุปกรณ์ระบบมีค้ดทุกชั้นทำการรีเซ็ตตัวเอง (กลับไปอยู่ในสภาวะแรกหลังไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากเปิดไฟเข้าเครื่อง)

-สัญญาณนาฬิกา (Timer Clock)

ในการกำหนดจังหวะจะมีการส่งสัญญาณนาฬิกาออกไปด้วยอัตรา 24 ลูก ต่อโน้ตตัวดำหนึ่งตัว (ส่งออกไปอย่างต่อเนื่อง) เพื่อให้เป็นจังหวะอ้างอิงในระบบควาใช้สัญญาณนาฬิกาเดียวกัน (อุปกรณ์ส่งสัญญาณนาฬิกาเพียงตัวเดียว)

เป็นข้อความที่บอกให้ซีควเอนเซอร์ และกลองอิเล็กทรอนิกส์กลับไปจุดเริ่มต้นของเพลงแล้ว เริ่มเล่นเพลงในทันทีที่ได้รับสัญญาณนาฬิกา นั่นคือจะบอกให้แต่ละอุปกรณ์เริ่มเล่นพร้อมๆ กับ การหยุด (stop)

ข้อความนี้จะบอกให้หยุดการกระทำทุกอย่าง ที่มีผลจากการกำหนดจังหวะ (timing-sensitive)

การเริ่มเล่นต่อ (Continue Start)

บอกให้อุปกรณ์ทำการเล่นต่อหลังตำแหน่งเพลงที่ถูกหยุดครั้งสุดท้าย

การตรวจสอบสภาวะการทำงาน (Active Sensing)

เป็นข้อความที่ใช้แก้ปัญหาตัวส่งกับตัวรับถูกตัดขาดออกจากกัน (เช่นสายหลุด) ซึ่งจะก่อให้เกิดความผิดพลาด เช่น ส่งข้อความให้เล่นโน้ตออกไปแล้วเกิดสายหลุด เมื่อทางด้านส่งส่งข้อความให้เลิกเล่นโน้ตตามไป ก็จะไปไม่ถึงด้านรับ ทำให้เกิดเสียงค้างที่เครื่องดนตรีด้านรับปัญหาจะเกิดความเสียหายมากในการแสดงสด

- แอคทีฟเซ็นซิง จะทำงานโดยการส่งข้อความที่เรียกว่า แอคทีฟเซ็นซิง ออกไปเรื่อยๆ เมื่อด้านรับได้รับ แอคทีฟเซ็นซิง แล้ว ถ้าภายในเวลาสั้น ๆ ที่กำหนดไม่ได้รับข้อความนี้เข้ามาอีกมันจะหยุดการสร้างเสียง

ที่กล่าวมาแล้วทั้งหมดเป็นความหมายของข้อความในระบบมีดีต่างๆ แต่ โดยทั่วไปเครื่องดนตรีแต่ละชิ้น ไม่สามารถตอบสนองต่อข้อความเหล่านี้ทุกข้อความ ซึ่งความสามารถในการตอบสนองต่อข้อความต่างๆ ของแต่ละอุปกรณ์ จะหาได้ใน มีดี อิมพลีเม้นท์ชีต (Midi - Implementation Sheet) ของอุปกรณ์นั้น ๆ สำหรับโครงการนี้ไม่ได้ใช้ข้อความทั้งหมดของที่กล่าวไปนั้น คงใช้แต่เพียงบางส่วน เนื่องจากเครื่องดนตรีประเภทระนาดนั้น จะไม่เหมือนกับเครื่องดนตรีสากลในบางอย่าง เช่น ในเรื่องของการเอื้อนเสียง ซึ่งในการตีระนาด แต่ละครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงพาณิชย์ด้านการค้า ไม่จำเป็นต้องมีการเอื้อนเสียง แต่ทักกล่าวไว้เพื่อเป็นการบอกพื้นฐานของระบบมีดี ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการสร้าง

รายละเอียดเกี่ยวกับโครงงาน

โครงงานนี้จะทำได้โดยการรับสัญญาณการตีระนาด โดยจะใช้ตัวเพียโซ (piezo) ซึ่งเป็นลำโพงอิเล็กทรอนิกส์ชนิดหนึ่งเป็นตัวรับสัญญาณ แล้วนำไปผ่านวงจรไฟฟ้าชุดหนึ่ง (รายละเอียดเกี่ยวกับวงจรชุดนี้จะได้กล่าวถึงต่อไป) ซึ่งใช้ในการอ่านค่าสัญญาณที่ได้รับ แล้วทำการปรับแต่งสัญญาณเพื่อนำไปให้เครื่องไมโครคอมพิวเตอร์ทำการวิเคราะห์ จากนั้นจึงนำมาแปลงเป็นรูปแบบของภาษามิตีมาตรฐาน เพื่อที่จะได้ใช้ระบบมิตีสำหรับการติดต่อสื่อสารกับเครื่องดนตรีอิเล็กทรอนิกส์ที่มีระบบมิตีในตัวเอง เช่น คีย์บอร์ด, กลองไฟฟ้าได้ การเชื่อมต่อระหว่างวงจรผ่านไปยังไมโครคอมพิวเตอร์นั้น จะใช้ผ่านทางพอร์ตเครื่องพิมพ์ อันนี้เพื่อประโยชน์ในการเชื่อมต่อกับเครื่องไมโครคอมพิวเตอร์เครื่องอื่นๆ ได้อย่างสะดวกและง่ายมากขึ้น ส่วนการเชื่อมต่อในคอมพิวเตอร์สำหรับการสื่อสารทางมิตีจะใช้การ์ดเสียบภายในเครื่อง ดังนั้นการรับสัญญาณจะผ่านทางพอร์ตเครื่องพิมพ์ แล้วนำสัญญาณจากพอร์ตเครื่องพิมพ์มาทำการแปลงเป็นภาษามิตีมาตรฐาน แล้วจึงส่งสัญญาณที่แปลงแล้วนั้นไปยังมิตีการ์ด เพื่อทำการสื่อสารกับคีย์บอร์ดในการกำเนิดเสียงตามระดับความดังค่อยและตัวโน้ต ซึ่งได้จากการวิเคราะห์แล้ว สำหรับการบันทึกข้อมูลนั้นจะทำเมื่อมีสัญญาณผ่านพอร์ตเครื่องพิมพ์เข้ามา และได้มีการใช้ซอฟต์แวร์วิเคราะห์ แล้วทำเป็นภาษามิตีมาตรฐานแล้วจึงเริ่มทำการบันทึกข้อมูลนั้น จะทำเมื่อมีสัญญาณผ่านพอร์ตเครื่องพิมพ์เข้ามา และได้มีการใช้ซอฟต์แวร์วิเคราะห์ แล้วทำเป็นภาษามิตีมาตรฐานแล้วเริ่มทำการบันทึกข้อมูล

ในการบันทึกข้อมูลนั้น ได้มีการใช้ฐานเวลาให้กับระบบ ซึ่งทำโดยการใช้วงจรรวมเบอร์ 8253 ซึ่งเป็น programmable interval timer มาทำเป็นฐานเวลา

โดยสรุปแล้ว การรับค่าสัญญาณ input ผ่านไมโครคอมพิวเตอร์นั้น การวิเคราะห์ระดับความดัง รวมทั้งการตรวจจับการตีจากระนาดต้องได้ค่าที่ถูกต้องแม่นยำ ดังนั้น ทางซอฟต์แวร์และฮาร์ดแวร์ ต้องมีประสิทธิภาพที่สุด มิเช่นนั้น สัญญาณที่ได้และข้อมูลที่นำมาบันทึก จะเป็นข้อมูลที่อาจจะผิดพลาดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1) ฮาร์ดแวร์

3.1.1) ฮาร์ดแวร์ในส่วนภาครับสัญญาณ

รายละเอียดในส่วนภาครับสัญญาณ input

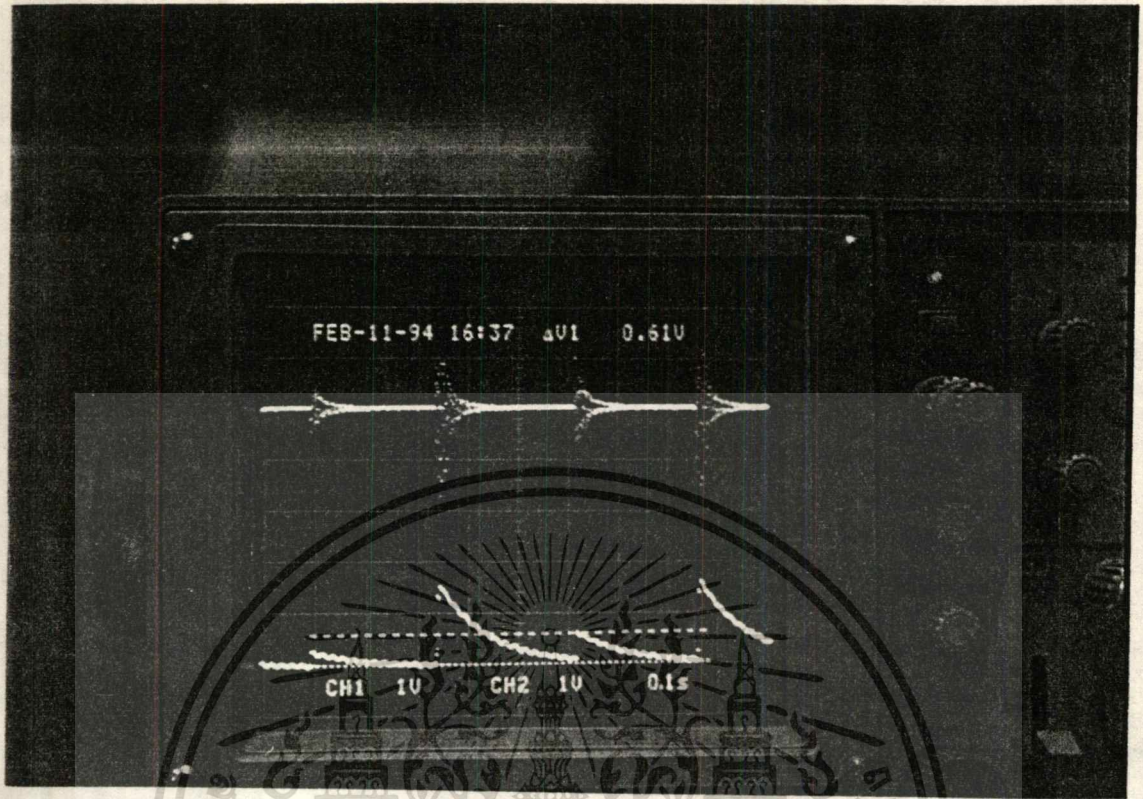
จากรูป เป็นส่วน HARDWARE ซึ่งจะมีการทำงานที่สามารถแบ่งการทำงานได้ 3 ภาคคือ

1. ภาคตรวจจับสัญญาณจากระนาด (Detect Signal)
2. ภาคการแปลงสัญญาณจากอะนาลอกเป็นดิจิตอล เพื่อเข้าคอมพิวเตอร์ (Analog to Digital)
3. ภาคการเชื่อมต่อกับคอมพิวเตอร์ ผ่านทางพริ้นเตอร์พอร์ต (Interface via printer port)

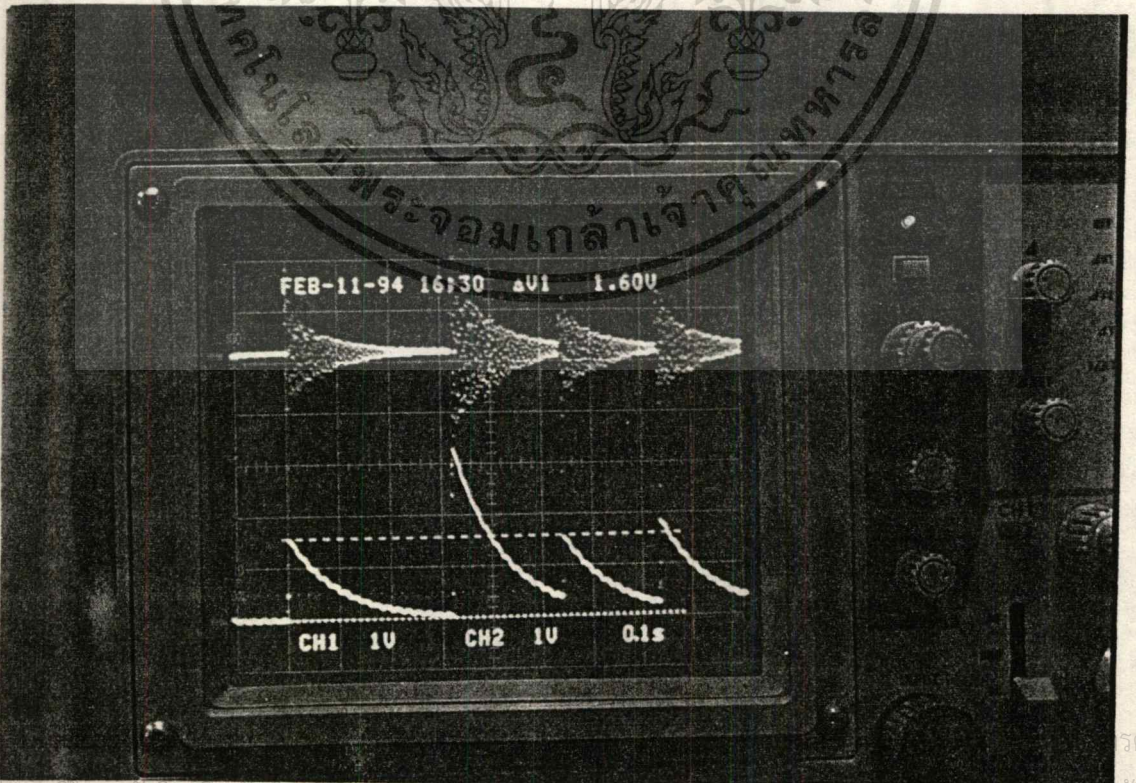
รายละเอียดของแต่ละภาคเป็นดังนี้

1. ภาคการตรวจจับสัญญาณจากระนาด (Detect Signal) ทำหน้าที่ในการตรวจจับสัญญาณจากระนาด หรือให้ได้ขนาดแรงดันสูงสุดในการตีระนาดแต่ละตัว โดยใช้ตัวเซ็นเซอร์เพียร์โซ (piezo) ในการจับสัญญาณจากระนาด คุณสมบัติของเพียร์โซ (piezo) คือ จะรับแรงสั่นสะเทือนแล้วเปลี่ยนเป็นสัญญาณไฟฟ้า เมื่อมีการตีระนาด จะทำให้เพียร์โซสั่น แล้วเกิดเป็นสัญญาณไฟฟ้าขึ้น ซึ่งสัญญาณไฟฟ้าที่ได้จะมีขนาดแรงดันสูงพอสมควร แต่กระแสที่เกิดขึ้นจะน้อยมาก จึงต้องมีตัวช่วยขยายกระแส จากนั้น ก็นำมาผ่านวงจรพีคดีเทค (peak detect) เพื่อหาขนาดของแรงดันสูงสุดในการตีแต่ละครั้ง เอาท์พุทที่ได้จะเป็นอินพุทในส่วนต่อไป

รูปหน้าถัดไปจะแสดง สัญญาณอินพุทในระดับการตีค่าต่างๆ



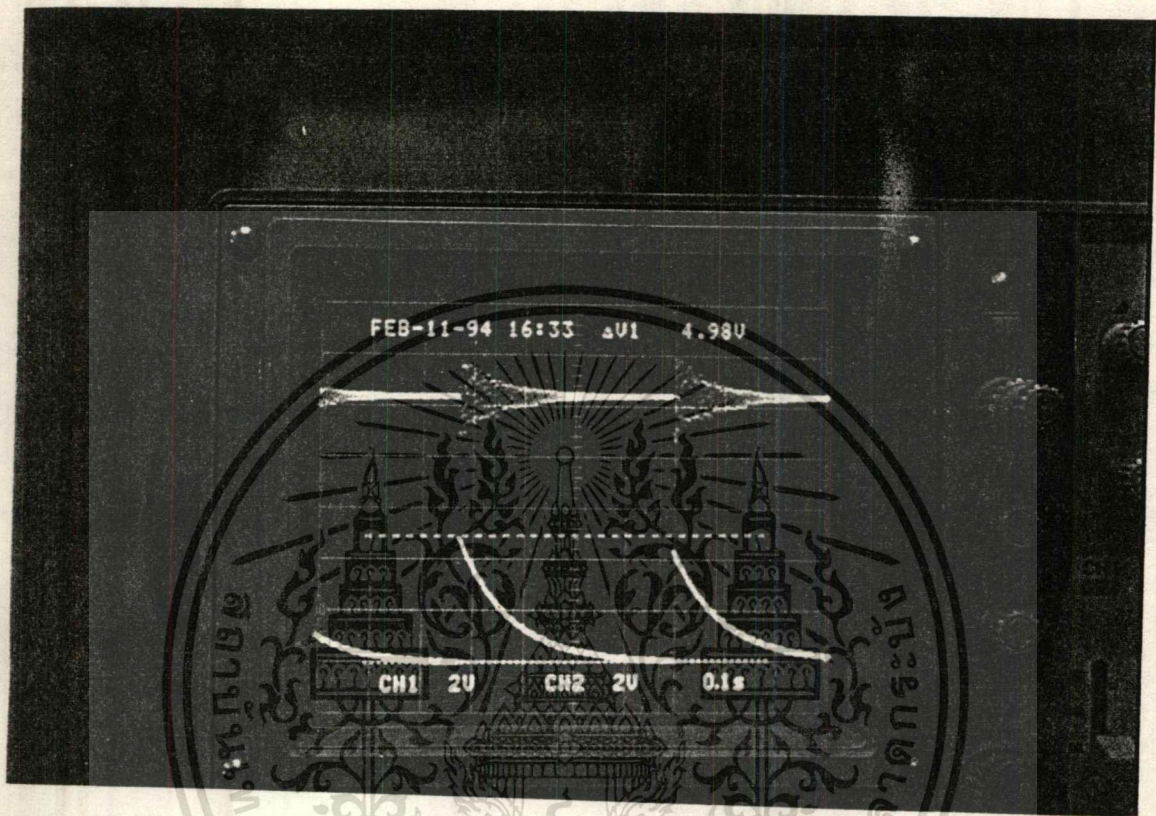
สัญญาณจากการตีระนาดขนาดเบา



สัญญาณจากการตีระนาดขนาดปานกลาง

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงแหล่งที่มาของเอกสารในทุกครั้งที่มีการนำแบบใช้



สัญญาอนุญาตการตีระนาดขนาดใหญ่
พระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

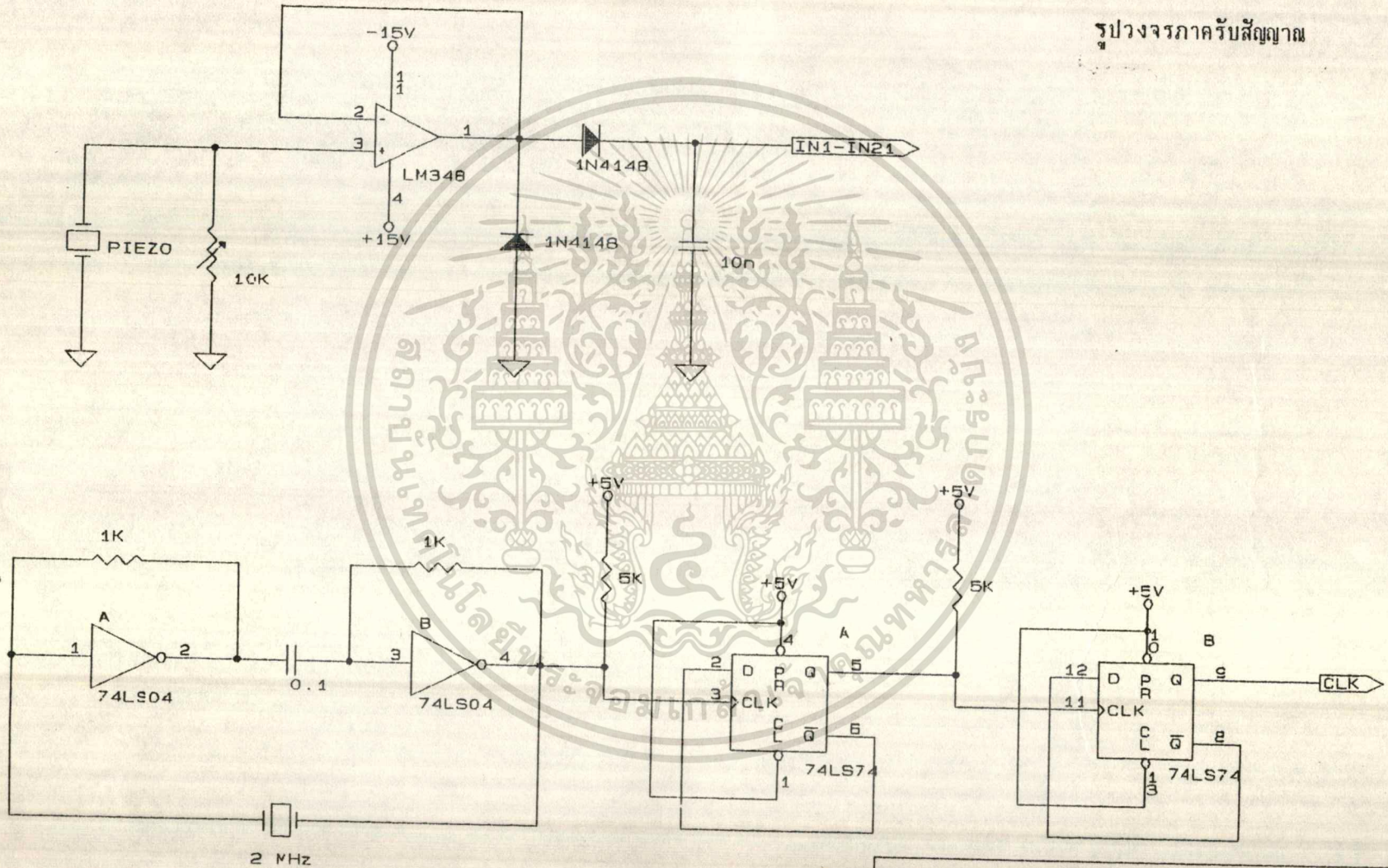
2. ภาคการแปลงสัญญาณจากอะนาล็อกเป็นดิจิทัล (Analog to Digital) ทำหน้าที่แปลงสัญญาณที่ตรวจจับได้จากกระนาบ ซึ่งเป็นทั้งหมด 21-22 สัญญาณ . ถ้าต้องใช้ A/D ถึง 21-22 ตัว ก็จะเป็นการสิ้นเปลืองมาก จึงมีส่วนของมัลติเพล็กซ์สวิตช์ หรืออะนาล็อกสวิตช์ หรือเลือกสัญญาณอินพุท ที่จะเข้า A/D เพียงตัวเดียว โดยการควบคุมสวิตช์

3. ภาคการเชื่อมต่อกับคอมพิวเตอร์ (Interface via printer port) ทำหน้าที่เป็นตัวเชื่อมต่อระหว่างส่วนของฮาร์ดแวร์ กับคอมพิวเตอร์ โดยผ่านทางพริ้นเตอร์พอร์ต เหตุที่เลือกใช้พริ้นเตอร์ ก็เพื่อความสะดวกในการเชื่อมต่อ และสามารถเคลื่อนย้ายไปต่อกับคอมพิวเตอร์เครื่องใดก็ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปวงจรภาครับสัญญาณ



MIDI TRANSLATION		
Title		
Detect Part and Clock Generate		
Size	Document Number	REV
A	1	

การใช้งาน พรีนเตอร์ พอร์ต (printer port)

พรีนเตอร์ พอร์ต เป็นพอร์ตแบบขนาน ซึ่งประกอบด้วยสัญญาณต่างๆ ทั้งหมด 25 เส้นสัญญาณต่างๆ นี้จะถูกใช้เป็นอินพุต และเอาต์พุตต่างๆ กัน สัญญาณต่างๆ เหล่านี้มาจากการจองแอดเดรส พอร์ต. อินพุต-เอาต์พุต พอร์ต 3 พอร์ต สำหรับพอร์ตอะแดปเตอร์เครื่องพิมพ์จะจองพอร์ตเบอร์ 378H-37AH ส่วนพอร์ตโมโนโครมใช้พอร์ตหมายเลข 3BCH-3BEH แต่ละพอร์ต มีหน้าที่ต่างๆ กัน คือ

-378 เป็นพอร์ตซึ่งสามารถอ่าน และเขียนข้อมูลได้ 8 บิต แต่ในทางปฏิบัติเราจะสามารถเขียนข้อมูลได้อย่างเดียว เพราะข้อมูลเอาต์พุตของพอร์ตนี้ จะถูก แลชชิ่ง โดย 74LS374 บนการ์ด ทำให้ข้อมูลที่เอาต์พุตออกไปถูกคงค่าไว้ เมื่อเราอ่านข้อมูลเข้ามา ก็จะได้ข้อมูลเดิมที่เอาต์พุตออกไป ซึ่งจะใช้ในการตรวจสอบความถูกต้องในการส่งข้อมูลออกไปนั่นเอง

-379 เป็นพอร์ตที่ใช้อ่านสถานะของเครื่องพิมพ์ ซึ่งจะสามารถอ่านข้อมูลได้เพียงบิตด้านที่ไบท์สูง บิต D0-D2 จะใช้ไม่ได้ ฮาสัญญาณที่พอร์ตนี้ มีดังนี้

1. BUSY (ขาที่ 11) เป็นสัญญาณที่ บิต D7 ซึ่งจะถูก กลับค่า (invert) ก่อนที่จะเข้าคอมพิวเตอร์
2. ACKING (ขาที่ 10) เป็นสัญญาณที่ออกจากบิตข้อมูล D6
3. PE (ขาที่ 12) เป็นสัญญาณที่ออกจากบิตข้อมูล D5
4. SLCT OUT (ขาที่ 13) เป็นสัญญาณที่ออกจากบิตข้อมูล D4
5. ERROR (ขาที่ 32) เป็นสัญญาณที่ออกจากบิตข้อมูล D3

ส่วนสัญญาณจาก D0-D2 จะไม่ถูกนำมาพิจารณา จะเห็นว่าสัญญาณที่เข้ามานี้ จะมีบางบิตที่ถูกกลับค่า ก่อนที่เครื่องจะทำการอ่านสถานะ ดังนั้นในการนำไปใช้งานเราอาจจะใช้ฮาร์ดแวร์ช่วยโดยการต่อ อินเวอร์เตอร์ (invertor) ก่อนที่จะเข้าเครื่องคอมพิวเตอร์ เพื่อความสะดวกในการตรวจสอบการทำงาน

-37A เป็นพอร์ตที่เป็นได้ทั้งอินพุต และเอาต์พุต แต่จะอ่านและเขียนสถานะได้ 5 บิตทางต่ำเท่านั้น โดย บิต D5-D7 จะไม่ถูกใช้งาน ส่วนสัญญาณที่ D4 ซึ่งเมื่อรวมกับสัญญาณที่ขา -ACKING จะได้เป็นสัญญาณอินเทอร์รัพท์ ของพรีนเตอร์ (IRQ7) ซึ่งบิตนี้จะนำมาใช้ทั่วไปไม่ได้ ดังนั้น สัญญาณที่เราสามารถใช้อ่าน-เขียนได้มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
1. SLCT IN (ขา 36) เป็นสัญญาณที่ขา D3 จะถูก invert ก่อนที่จะถูกส่งออกจากคอมพิวเตอร์ ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการเขียนข้อมูล และถูกกลับค่า ในช่วงการอ่านค่าข้อมูลด้วย

2. INIT (input prime) (ขา 31) เป็นสัญญาณที่ขา D2
3. AUTO FEEDKT (ขาที่ 14) เป็นสัญญาณที่ขา D1 ถูกกลับค่าเช่นกัน
4. STORE (ขาที่ 1) เป็นสัญญาณที่ขา D0 ถูกกลับค่าเช่นกัน

สำหรับโครงงานนี้เราจะเชื่อมต่อระหว่างเครื่องดนตรีกับคอมพิวเตอร์ โดยผ่านทาง พรีนเตอร์-พอร์ต โดยการนำสัญญาณจากพอร์ต ทั้ง 3 พอร์ตมาใช้ควบคุมฮาร์ดแวร์ของระบบ ในส่วนของการรับอินพุต จาก ADC0809 ซึ่งจะต้องใช้สัญญาณอินพุต ทั้งหมด 8 บิต แต่เนื่องจากพรีนเตอร์ที่ใช้เป็นอินพุตใด ไม่มีพอร์ตที่สามารถรับอินพุตได้ 8 บิตพร้อมกัน เราจึงใช้บิตข้อมูลจาก พอร์ต 379 มา 5 บิต (D3-D7) และจาก พอร์ต 37A 3 บิต (D0-D2) แล้วจึงนำมารวมกันโดยซอฟต์แวร์ ส่วนสัญญาณเอาต์พุต ที่ได้มาจากพอร์ต 378 จะถูกใช้ในการทริก(trig) ADC และใช้ในการเลือกขาสัญญาณจากกระนาบ เพื่อเข้าคอมพิวเตอร์

3.1.2) ฮาร์ดแวร์ในส่วนมิดีการ์ด

หลักการทํางานที่สำคัญมีดี การ์ด (MIDI CARD) อยู่ที่ IC Z80B-SIO ซึ่งจะทำหน้าที่เป็นอินพุต-เอาต์พุต แบบอนุกรม (SERIAL INPUT OUTPUT) โดยเราจะทำการโปรแกรมให้ SIO ทํางานในโหมด อินเทอร์รัพท์ (Interrupt Mode) ซึ่งมีวงจรตามภาพหน้าถัดไป

Z80-SIO จะรับความถี่ในการมอด บนคริสตัล (CRYSTAL) ขนาด 2 MHz ซึ่งต้องร่วมอยู่กับ IC 74LS00 ในการส่งข้อมูลจะมี IC 74LS04 ทำหน้าที่เป็น บัฟเฟอร์ (BUFFER) โดยจะมีตัวต้านทาน (RESISTOR) ขนาด 220 โอห์ม ซึ่งจะแมทซ์ (Match) กับ ค่าอิมพีแดนซ์ (Impedance) ของมิดีอินพอร์ต (ค่ามาตรฐานของมิดี)

ในการอ้างพอร์ท(port) ของ IBM ถึง Z80B-SIO เนื่องจากมันมีประสิทธิภาพที่เหนือกว่า CHANNEL A (แต่ในโครงงานนี้ยังไม่ได้ใช้) และ ใช้หมายเลขของ คอนโทรลพอร์ท - (CONTROL PORT) คือ 2B1H และ ดาต้าพอร์ท (DATA PORT) คือ 2B0H

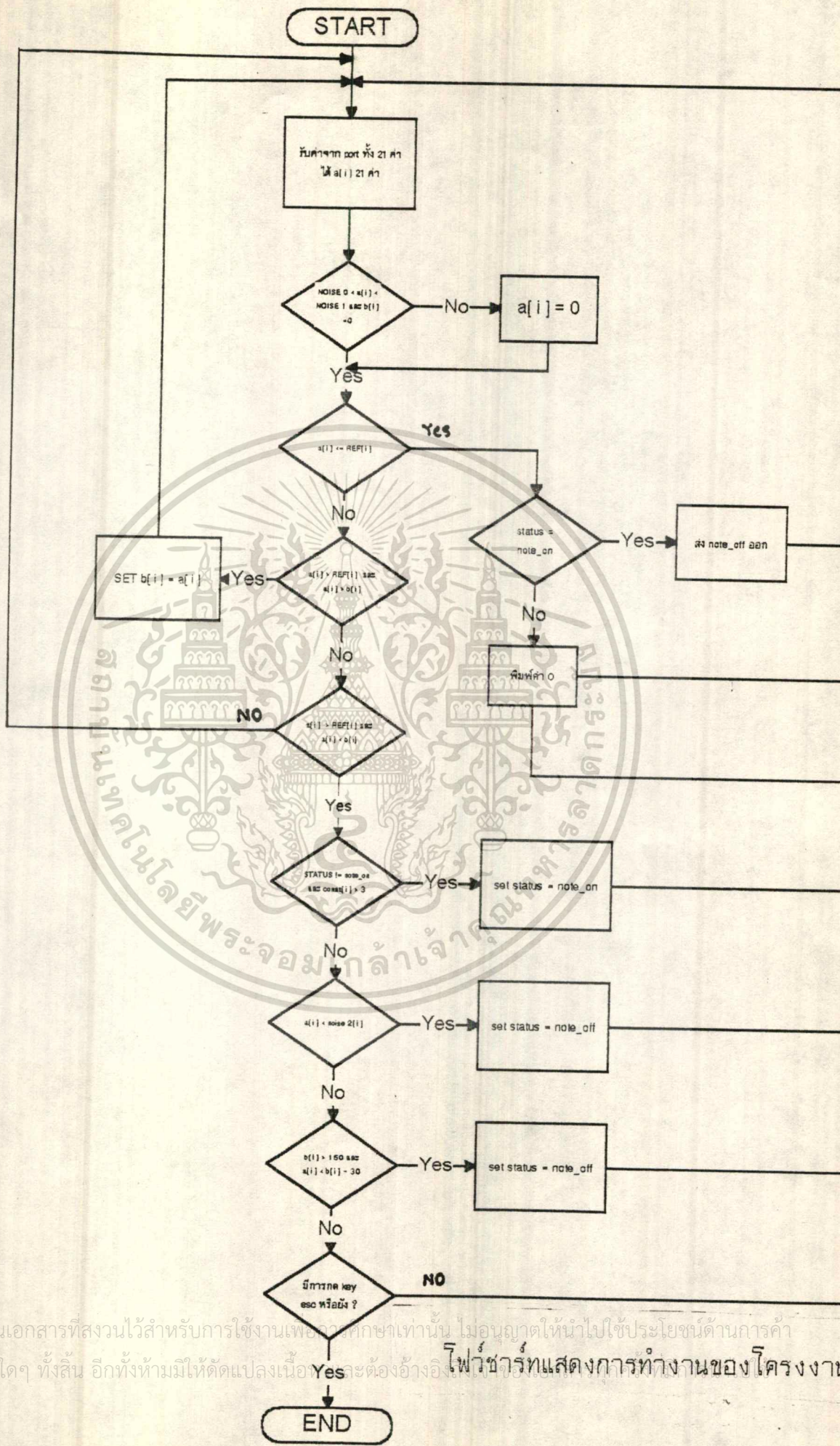
Z80B-SIO จะถูกโปรแกรมให้รับส่งข้อมูลด้วย BAUD RATE FACTOR = 64 ซึ่งจะได้ความถี่ในการมอดดูเลขขึ้น = $2 \text{ MHz} / 64 = 31.25 \text{ Kbaud}$ พอดี และมีขนาดข้อมูล 8 Bit พร้อมบิต-เริ่ม/บิตหยุด (STOP/START BIT) อย่างละ 1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ส่วนประกอบสำคัญอีกส่วนหนึ่ง คือ 8253 ซึ่งเป็น PROGRAMMABLE INTERVAL TIMER ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะนำมาใช้ในการทำ ฐานเวลา ให้กับระบบสำหรับตัว 8253 เองจะมี วงจรนับ(COUNTER) อยู่ 3 ตัวอิสระ คือ CO, C1 และ C2 ที่ใช้ใน มิตี์ คาร์ดนี้ เราจะใช้ CO และ C1 โดยเราจะกำหนดให้ C1 ทำงานในโหมด 3 (MODE 3) ผลิตคลื่นสี่เหลี่ยม (SQUARE WAVE GENERATOR) ซึ่งในโหมดนี้ เราจะใช้ในการนำมารักษาความถี่ลงมาจากความถี่ อินพุต ที่ป้อนให้ที่ขา C1 คือ- 2 MHz ลงมาเหลือ 1 MHz โดยทำการหาร 2 นั้นเองจากนั้นเราจะนำความถี่ที่หารแล้วจาก - C1 มาป้อนเป็น คล็อก(clock) ให้กับ CO ซึ่งที่ CO นี้ ก็จะทำงานใน โหมด 3 เช่นกันแต่ค่าที่ใช้ในการหารความถี่ตรงนี้ จะเป็นค่าที่ อินพุตมาจาก ผู้ใช้ (user) เพื่อใช้กำหนดให้เป็นฐานเวลาของความเร็วย้ำของจังหวะ(tempo) จากนั้นนำค่าความถี่จาก CO ที่ขา OUT 0 ไปทริกที่- IORQ 3 ส่วน Z80B-SIO ที่ขา อินเทอร์รัพท์ (INT) จะต่อกับ IORQ2 เพื่ออินเทอร์รัพท์ซึ่งจะมีการทำงานเมื่อมีการรับส่งข้อมูลจากเครื่องดนตรี

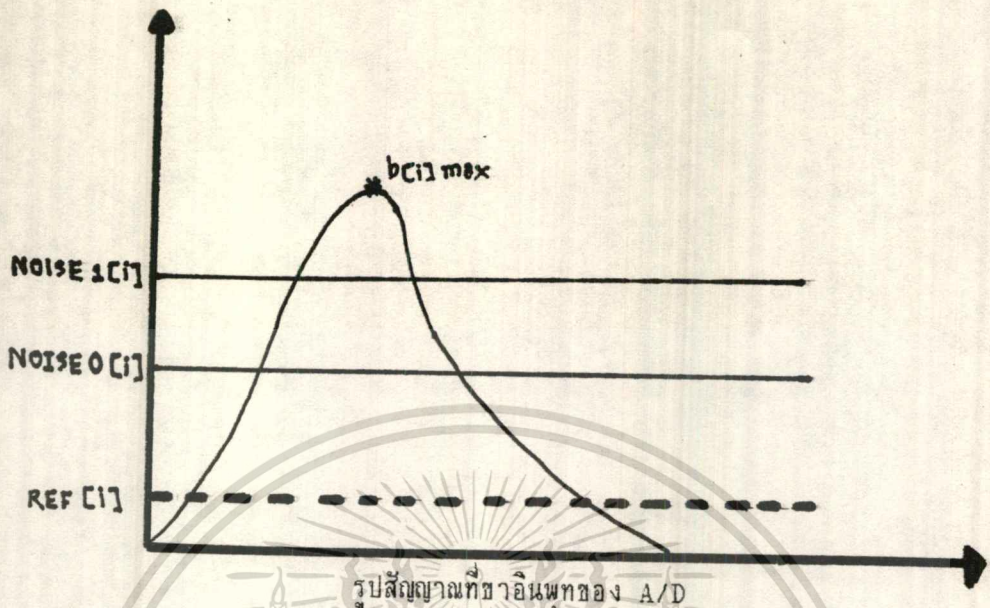
คอนเนคเตอร์ (Connector) ในส่วนที่ติดต่อกับอุปกรณ์ มิตี์ จะใช้แจ็คดิน ตามมาตรฐาน มิตี์ แต่ในส่วนที่ติดต่อกัน เครื่อง IBM จะใช้ DB CONNECTOR 9 ขา แทนเนื่องจากยังอยู่ในช่วงการศึกษาและพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อเรื่องและต้องอ้างอิงไว้ที่ [เว็บไซต์ที่แสดงการทำงานของโครงการ](#)

3.2) ซอฟต์แวร์



- NOISE0[i] - NOISE1[i] : เป็นค่าของสัญญาณรบกวนของวงจรส่วน hardware
- REF[i] : เป็นระดับอ้างอิงที่ A/D จะสามารถทำการชั่งค่าสัญญาณได้อย่างถูกต้อง
- status[i] : เป็นตัวแปรที่กำหนดสภาพระดับการปิด เปิดตัวไนต์
- NOISE2[i] : เป็นระดับการปิดตัวไนต์

3.2.1) ซอฟต์แวร์การทำงานของโครงงาน

1. รับค่าจากพอร์ตเครื่องพิมพ์เข้ามาพร้อมกัน 21 ค่า
2. นำค่าที่รับเข้ามา มาเปรียบเทียบว่า อยู่ในช่วงระหว่าง NOISE0[i] ถึง NOISE1[i] หรือไม่ ถ้าอยู่ในช่วงนี้ แสดงว่ายังไม่มีค่าเข้ามา จะกำหนดให้ a[i] มีค่าเท่ากับ 0 ถ้า a[i] ไม่อยู่ในช่วงนี้ แสดงว่ามีค่าเข้ามาแล้ว a[i] ก็มีค่าคงเดิมที่รับเข้ามา
3. นำค่า a[i] ที่ได้มาทำการเปรียบเทียบ โดยพิจารณาออกเป็น 3 กรณีด้วยกัน คือ
 - 3.1 ถ้า a[i] มีค่าน้อยกว่าค่า REF[i] จากกราฟ การที่ค่า a[i] <= REF[i] นี้ มี 2 ช่วงคือ

*ช่วงที่กราฟเริ่มขึ้น (เพิ่งมีการตีเข้ามา) ช่วงนี้โปรแกรม จะทำเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า การกำหนดค่าให้ a[i] เท่ากับ 0
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

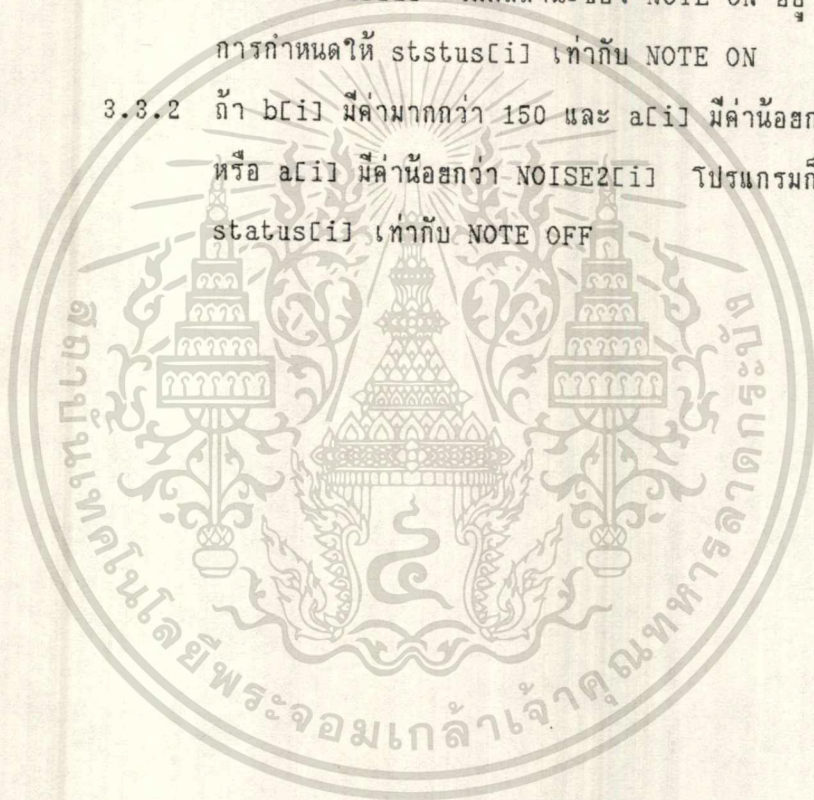
*ช่วงกลางของกราฟ โปรแกรมจะทำการส่งสัญญาณการปิดโน้ต (NOTE OFF) ไปหยุดการส่งโน้ต

3.2 ถ้า $a[i]$ มีค่ามากกว่า $REF[i]$ และ มากกว่า $b[i]$ ด้วย โปรแกรมก็จะกำหนดให้ $b[i]$ เท่ากับ $a[i]$

3.3 ถ้า $a[i]$ มีค่ามากกว่า $REF[i]$ และน้อยกว่า $b[i]$ จะแยกพิจารณาได้เป็น 2 กรณีด้วยกันคือ

3.3.1 ถ้า $status[i]$ ไม่มีสถานะของ NOTE ON อยู่ โปรแกรมก็จะทำการกำหนดให้ $status[i]$ เท่ากับ NOTE ON

3.3.2 ถ้า $b[i]$ มีค่ามากกว่า 150 และ $a[i]$ มีค่าน้อยกว่า $(b[i]-30)$ หรือ $a[i]$ มีค่าน้อยกว่า $NOISE2[i]$ โปรแกรมก็จะกำหนดให้ $status[i]$ เท่ากับ NOTE OFF



3.2.2) ซอฟต์แวร์ ส่วนที่ทำงานร่วมกับ ฮาร์ดแวร์

ซอฟต์แวร์ ของโครงการนี้จะทำงานประสานกับ ฮาร์ดแวร์ หลายๆ ส่วนในระบบซึ่งได้แก่ - คีย์บอร์ด ซึ่งใช้ในการติดต่อกับระบบทั้งในขณะเล่นเพลงและขณะ SETUP PARAMETER- ต่างๆ ของเพลง รวมทั้งการ โทลด์, เซฟ และ แก้ไข (EDIT)

-จอภาพ จะใช้การแสดงผลข้อมูลต่างๆ ของระบบและที่สำคัญ ก็คือการแสดงผลเวลาหรือ หมายเลขบีท(Beat) ของระบบ ซึ่งจะต้องใช้วีซีดีแอม (FILL)ลงในหน่วยความจำหน้าจอ (CRT MEMORY)

-INT 1Ch ไทม์เมอร์ตีค อินเทอร์รัพท์ (TIMER TICK INTERRUPT)ตัวนี้จะทำงานขณะที่ 8253 ตีค 8259 เพื่อขอการเพิ่มเวลาในระบบของ IBM/PC ซึ่งเราจะใช้โปรแกรมย่อย (ROUTINE) นี้ในการส่ง แอ็คทีฟ เซนส์ (ACTIVE SENSE) ลงไปในอินเทอร์รัพท์นี้

-MIDI CARD 8253 INTERRUPT จะเป็นฐานเวลาสำหรับเพลงซึ่งจะถูก SET ให้มีขนาด 24 CLOCK/โน้ตตัวค่า ผ่านทาง IORQ2

-MIDI CARD -SIO INTERRUPT จะ INTERRUPT ระบบเมื่อมีข้อมูลส่งเข้ามาในระบบ ผ่านทาง IORQ3

ซอฟต์แวร์ ที่ทำงานเหล่านี้จะทำงานร่วมกับ โปรแกรมหลัก (MAIN PROGRAM) ในขณะเล่น เพลง, บันทึกเพลง และการ SETUP ค่าต่างๆ สำหรับ คีย์บอร์ด และมอนิเตอร์ เป็นการเขียน โปรแกรมติดต่อกับ ระบบผ่าน ไบออส (BIOS) และ DOS INTERRUPT ธรรมดา ซึ่งจะไม่ขอ อธิบายในที่นี้

การทดลองและผลการทดลอง

1) ส่วนการทำงาน

ได้ทำการทดลองโดยการติดตัวรับสัญญาณไว้ได้ขนาด แล้วนำค่ามาวิเคราะห์โดยใช้ไมโครคอมพิวเตอร์ เพื่อทำการแปลงเป็นรหัสมิติ เพื่อที่จะสามารถติดต่อกับเครื่องดนตรีหรือเครื่องดนตรีอื่น ๆ ได้ นอกจากนี้ยังได้ทดลองการอัดเพลงเก็บไว้ในหน่วยความจำ โดยการเก็บเป็นไฟล์ แต่ละไฟล์ก็สามารถบรรจุเพลงแต่ละเพลงเอาไว้ เพื่อจะสามารถเรียกข้อมูลที่เก็บไว้ออกมาเพื่อทำการเล่นใหม่ได้

2) ส่วนการแสดงผลและการใช้งาน

ได้ทำให้การใช้งานง่ายขึ้น โดยผ่านการแสดงหน้าจอเพื่อรับการเลือกใช้คำสั่งต่าง ๆ ไว้ นอกจากนี้ ยังได้แสดงรูปร่างระนาบไว้ด้วย เพื่อสามารถดูการตีว่าตีที่รางใด (เมื่อต้องการเรียกข้อมูลออกมา) และยังได้แสดงหน้าจอที่แสดงระดับความดังของสัญญาณไว้ด้วย

ผลการทดลอง

- เมื่อมีการตีระนาด โปรแกรมก็จะสามารถวิเคราะห์ระดับสัญญาณเสียง และส่งไปที่คีย์บอร์ดได้ แต่เนื่องจาก ข้อบกพร่องของ hardware ซึ่งทำให้เกิด noise ขึ้น รวมทั้งความไม่แน่นอนของสัญญาณที่ได้จากการตี ทำให้เกิดปัญหาหลายๆ อย่าง คือ

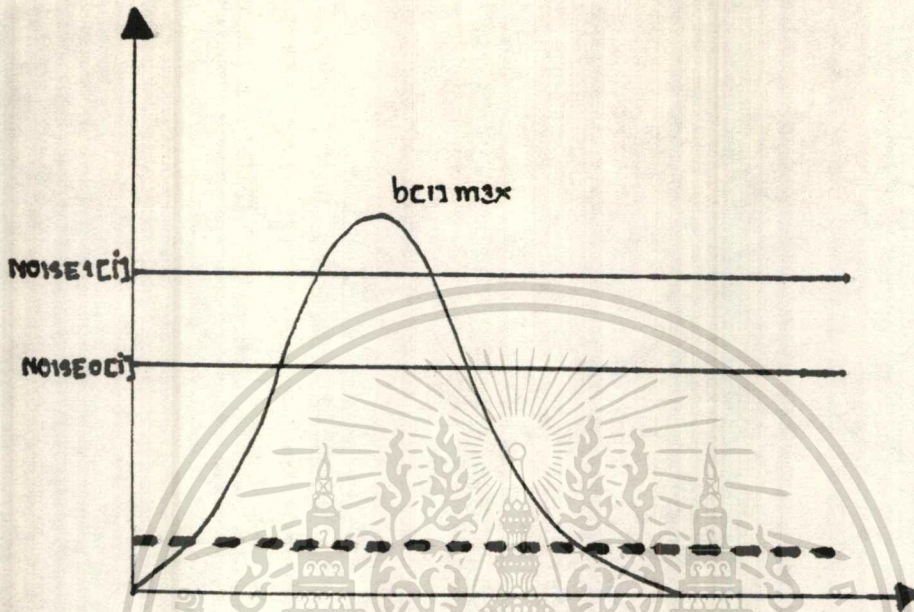
1. โปรแกรมไม่สามารถตรวจจับสัญญาณได้ในทุก ๆ ครั้ง ที่เรากำลังตี
2. เมื่อตีเพียงครั้งเดียว จะทำให้เกิดตัวอื่นๆ เกิดเสียงตามมาด้วย เพราะเกิดจากความสั่นสะเทือนของรางระนาด

ปัญหา 2 ข้อข้างต้นดังกล่าว เราพอที่จะทราบถึงสาเหตุซึ่งได้พยายามแก้ไขแล้ว แต่ไม่สามารถแก้ปัญหาทั้งหมดในขณะเดียวกันได้ จึงพยายามแก้ไขให้เกิดปัญหาน้อยที่สุด

จากผลการทดลอง พบว่าสามารถทำให้ระนาดนี้ติดต่อกับเครื่องดนตรีอื่นๆ ได้ โดยการใช้งานผ่านระบบมิติ ในส่วนการรับค่าจากการตีเพื่อนำไปวิเคราะห์ข้อมูลนั้น ปัญหาหลายประการที่ทำให้ยากในการที่จะรับข้อมูลได้ละเอียดครบถ้วน และเหมือนจริงมากที่สุด ซึ่งได้วิเคราะห์ปัญหาไว้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่เกิดขึ้นทางซอฟต์แวร์



รูปสัญญาณที่ขาอินพุทของ A/D

- NOISE0[i] - NOISE1[i] : เป็นค่าของสัญญาณรบกวนของวงจรส่วน hardware
- REF[i] : เป็นระดับอ้างอิงที่ A/D จะสามารถทำการชั่งค่าสัญญาณได้อย่างถูกต้อง
- status[i] : เป็นตัวแปรที่กำหนดสภาวะระดับการปิด เปิดตัวไนต์
- NOISE2[i] : เป็นระดับการปิดตัวไนต์

ข้อผิดพลาดที่เกิดขึ้นจากการทำงานของซอฟต์แวร์

1. การสั่นสะเทือนของรางขนาดทำให้ลำโพงเพียโซ (piezo) ตรวจจับสัญญาณได้ทุกตัว จึงเหมือนกับว่ามีการตีระนาดทุกตัวพร้อมกัน

การแก้ไข โดยการตั้งค่า REFERENCE ไว้ที่ค่าสูงๆ

ผลเสีย - เมื่อตีเบาๆ โปรแกรมจะไม่สามารถตรวจจับค่าได้ เพราะค่าที่ได้จากการตีเบาๆ บางครั้ง จะมีค่าต่ำกว่า REFERENCE ผลที่เกิดตามมา คือ โปรแกรมจะตรวจจับการตีระนาดไม่ได้ทุกครั้ง

2. ความไว ในการรับค่าจากการสุ่มสัญญาณ (SAMPLING) (ความไวหมายถึงความสามารถที่จะตรวจจับสัญญาณได้ทุกค่าและทุกครั้งที่มีการตี)

ไม่มีการนับใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุแบบส่งเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแก้ไข โดยการตั้งค่า REFERENCE ไว้ที่ค่าต่ำๆ และกำหนดระดับสัญญาณ NOTE-OFF ไว้ที่ค่าสูงๆ

ผลเสีย สัญญาณที่ขาอินพุทของวงจรแปลงสัญญาณอนาล็อกไปเป็นดิจิทัล ยังไม่หมดไปโปรแกรมจะทำการตรวจจับได้อีก ผลที่เกิดในข้อนี้จะเป็นสาเหตุของปัญหาต่อไป

3. การเกิดมากกว่า 1 เสียงในตัวเดียวกัน กรณีที่ตีระนาดตัวนั้นเพียงเดียว ซึ่งมีสาเหตุมาจากการตั้งระดับของการปิดโน้ต (NOTE-OFF) สูงเกินไป

การแก้ไข โดยการตั้งระดับการปิดโน้ตที่ค่าต่ำๆ คือ โปรแกรมจะทำการสุ่มสัญญาณจนอินพุทที่เข้าสู่ A/D เป็น 0

ผลเสีย จะเกิดการค้างของระดับเสียงเป็นเวลานาน เพราะบางครั้งสัญญาณที่ได้ จะมีช่วงขาลงของกราฟที่ชันมาก

จากที่กล่าวมาแล้วทั้งหมด จะเห็นว่า เป็นภาระยากมากที่โปรแกรมจะทำการตรวจจับค่าได้อย่างถูกต้องแน่นอน ดังนั้นโปรแกรมในส่วนนี้จึงยังมีข้อจำกัดอยู่

บทที่ 6
สรุปและวิจารณ์

การใช้ไมโครคอมพิวเตอร์มาประยุกต์ใช้กับระบบมิดี้ สามารถใช้งานได้ดี แต่ต้องคำนึงถึงเรื่องของความเร็วของเครื่องไมโครคอมพิวเตอร์ด้วย สำหรับการนำมาประยุกต์ใช้กับเครื่องดนตรีไทยก็สามารถใช้ร่วมกันได้อย่างดี แต่จะเกิดมีปัญหากำในส่วนของวงจรรักษาบริบทสัญญาณ และโปรแกรมในการวิเคราะห์ที่ไม่ดีจริงๆ จากโครงงานนี้พบว่า สามารถทำให้ระนาดซึ่งเป็นเครื่องดนตรีไทย สามารถติดต่อกับเครื่องดนตรีสากลได้จริง โดยผ่านระบบมิดี้ แต่จะใช้งานได้เป็นอย่างดี ต้องได้รับการพัฒนาต่อไป

แนวทางในการพัฒนาต่อไป

- 1) พัฒนาฮาร์ดแวร์ โดยใช้ฮาร์ดแวร์ที่สามารถรับสัญญาณการตีได้แม่นยำ และค่าถูกต้องมากขึ้น
- 2) ทางซอฟต์แวร์ ใช้ซอฟต์แวร์ที่มีความเร็วสูง เพื่อสามารถวิเคราะห์ค่าได้ถูกต้อง และใกล้เคียงความจริงมากที่สุด และเป็นซอฟต์แวร์ที่สามารถครอบคลุมถึงการใช้งานจริงๆ ได้ดี

ซึ่งจากโครงงานนี้ ถ้าจะพัฒนาต่อไปอาจจะนำมาประยุกต์ใช้กับเครื่องดนตรีไทย ที่เป็นเครื่องตีชนิดอื่น ซึ่งการทดลองจะใกล้เคียงกับโครงงานนี้ เพื่อประโยชน์ในการติดต่อกันได้ของเครื่องดนตรีเหล่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กติการมประกาศ

โครงการนี้คงไม่สามารถดำเนินมาถึงขั้นนี้ได้ ถ้าปราศจากการสนับสนุนจากบุคคลที่มีพระ
คุณยิ่งเหล่านี้

1. คุณอุทัย เกียรติวิกรัย ผู้ช่วยทำให้เกิดโครงการนี้
2. ดร. รัตติกกร วรากรศิริพันธ์ อาจารย์ที่ปรึกษา
3. ห้องสมุดคณะวิศวกรรมศาสตร์
4. พี่ชุ้น ให้ข้อมูลและให้สัมภาษณ์บอร์ด, อาจารย์คณชัย สุขเจริญผล เอื้อเพื่ออุปกรณ์ทดสอบ
พีไอ, พีโทน ช่วยวิเคราะห์วงจร, พี่มัม เอื้อเพื่อคอมพิวเตอร์, คุณเอ๋, คุณโอ เพื่อน
ร่วมวิจัยโครงการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. CRAIG ANDERSON, "MIDI FOR MUSICIANS," AMSCO PUBLICATION 105P.1986
2. JAY KUBICKY "A MIDI PROJECT", 199P, BYTE JUNE 1985 VOL 11
3. ชานินทร์ แก้วคำสรวงศ์ กับ ทินกร ตึก "การอินเทอร์เฟส IBM PC " 63P.
4. JAMES W. COFFRON "Z80 APPLICATION" 239P.1983
5. คู่มือ TURBO C++
6. IBM PC/XT TECHNICAL REFERENCE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <string.h>
#include <alloc.h>
#include <bios.h>
#include <time.h>
#include <dir.h>
#include <dos.h>
#include <process.h>
#include <stdlib.h>
#include <mem.h>

#define max_menu 5
#define ACK 'A'
#define NACK 'N'

/* DEFINE NUMBER PORT */
#define OPP 0x378
#define INPH 0x379
#define INPL 0x37a
#define J 21
#define control_port 0x2b1
#define data_port 0x2b0
#define note_on 0x90
#define note_off 0x80
#define Active_sense 0xFE

```

เอกสารนี้เป็นทรัพย์สินของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะในรูปแบบใดก็ตาม หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define MINIMUM 30
#define MIN 60
#define MIDDLE 90
#define MAXIMUM 120
#define MAX1 70 /* REFERENCE VALUE FOR NOTE ON */
#define TWO 50
#define ONE 29

```

```

int i=0;
char code[211]={52,53,55,57,59,61,63,64,66,68,70,72,74,75,77,79,81,82,84,86,88};
unsigned char far *head_mem,*tail_mem,*head;
unsigned char far *ptr;
unsigned int main_tick,main_timer;
char menu_text[max_menu][10]={"File","Translate","Record","Play","Tempo"};
char menu_xpos[max_menu]={4,11,23,34,43};
char far *vga;
int funckey,candle;
char *data_temp_node,*data_temp;
int tempo_rate=120;
char string[20];

```

```

void far interrupt ( *SAVE_IRQ3)(void);
void far interrupt ( *Save_Int1C)(void);
void init_sio(void);
void program_8253(void);

```

```

void midiout(unsigned char midi_data);
void quit(void);
void far *getmem(unsigned long size);
void interrupt active_sense(void);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void startup(void);
void setmem(void *,unsigned,char);
void play_(void);
void tempo(void);
void show_window(int x1,int y1,int x2,int y2,char *ptr);
int play(void);
void playback(void);
int forward(void);
int reverse(void);
void record(void);
int func(int ascii);
void init_menu_command(void);
void main_menu(void);
void draw_main_menu(void);
void translate(void);
void menu(void);
void box(int r,int c,int k,int m);
void check_status(void);

```

```

typedef struct{
    int max;
    int pos;
    int x_pos;
    int y_pos;
    int width;
    char command[8][20];

```

```

}pull;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pull p_menu[max_menu];

```



```
char readfunc(void){
    char ch;

    ch=getch();
    if(!ch){
        ch=getch();
        funckey=1;
    }
    else
        funckey=0;
    return(ch);
}
```

```
void init_menu_command(void){
    p_menu[0].max=2;
    p_menu[0].pos=0;
    p_menu[0].x_pos=0;
    p_menu[0].y_pos=1;
    p_menu[0].width=10;
    strcpy(p_menu[0].command[0],"Load");
    strcpy(p_menu[0].command[1],"Quit");

    p_menu[1].max=0;
```

```
p_menu[2].max=2;
p_menu[2].pos=0;
p_menu[2].x_pos=20;
p_menu[2].y_pos=1;
```



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
p_menu[2].width=10;
strcpy(p_menu[2].command[0],"Record");
strcpy(p_menu[2].command[1],"Playback");
```

```
p_menu[3].max=0;
```

```
p_menu[4].max=0;
```

```
}
```

```
void load(void)
```

```
{ char *save_scr=malloc(400);
```

```
unsigned int size;
```

```
FILE *fp;
```

```
do { show_window(20,3,50,7,save_scr);
```

```
gotoxy(24,6);
```

```
printf("File name = ");
```

```
gets(string);
```

```
fp = fopen(string,"rb");
```

```
if(fp==NULL)
```

```
{ gotoxy(24,7);
```

```
printf("Can't open file.");
```

```
}
```

```
} while(fp==NULL);
```

```
puttext(20+1,3+1,50+1,7+1,save_scr);
```

```
free(save_scr);
```

```
fseek(fp,0,SEEK_END);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

size=ftell(fp);
tail_mem=head_mem+size;
fread(head_mem,size,1,fp);
fclose(fp);
}

```

```

void bk_ground(int n,int type){

```

```

    int i;

```

```

    unsigned char attr;

```

```

    if(type==0)

```

```

        attr=0x70;

```

```

    else

```

```

        attr=0xF4;

```

```

    for(i=0;i<6;i++){

```

```

        *(vga+3*160+8*2+n*14+i*2+1)=attr;

```

```

    }

```

```

}

```

```

void play_(void){

```

```

    #define REVERSE 0

```

```

    #define PLAY 1

```

```

    #define FORWARD 2

```

```

    #define STOP 3

```

```

    int mode=STOP;

```

```

    int old_mode=STOP;

```

```

    int end=0;

```

```

    char ch;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int x_pos=10,y_pos=3;

int i;

*(vga+y_pos*160+x_pos*2)=17;
*(vga+y_pos*160+(x_pos+1)*2)=17;
*(vga+(y_pos-1)*160+(x_pos+1)*2)=27;

*(vga+y_pos*160+(x_pos+8)*2)=16;
*(vga+(y_pos-1)*160+(x_pos+8)*2)=24;

*(vga+y_pos*160+(x_pos+14)*2)=16;
*(vga+y_pos*160+(x_pos+15)*2)=16;
*(vga+(y_pos-1)*160+(x_pos+15)*2)=26;

*(vga+y_pos*160+(x_pos+20)*2)='S';
*(vga+y_pos*160+(x_pos+21)*2)='T';
*(vga+y_pos*160+(x_pos+22)*2)='O';
*(vga+y_pos*160+(x_pos+23)*2)='P';
*(vga+(y_pos-1)*160+(x_pos+22)*2)=25;

```

```

for(i=0;i<4;i++)

```

```

    bk_ground(i,0);

```

```

do{

```

```

    if(kbhit()){

```

```

        ch=readfunc();

```

เอกสารนี้เป็นเอกสารลับ (secret) ได้รับความเห็นชอบจากคณะกรรมการบริหารงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม้ if(ch==27)end=1; ห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

```

```

else{
gotoxy(20,23);
printf("                ");
switch(ch){
case 75:old_mode=mode;
mode=REVERSE;
bk_ground(old_mode,0);
bk_ground(mode,1);
break;

case 77:old_mode=mode;
mode=FORWARD;
bk_ground(old_mode,0);
bk_ground(mode,1);
break;

case 72:old_mode=mode;
mode=PLAY;
bk_ground(old_mode,0);
bk_ground(mode,1);
break;

case 80:old_mode=mode;
mode=STOP;
bk_ground(old_mode,0);
break;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch(mode){
    case PLAY :if(!play())
    { old_mode = mode;
      bk_ground(old_mode,0);
      mode = STOP;
    }
    break;
    case REVERSE :if(!reverse())
    { old_mode = mode;
      bk_ground(old_mode,0);
      mode = STOP;
    }
    break;
    case FORWARD :if(!forward())
    { old_mode = mode;
      bk_ground(old_mode,0);
      mode = STOP;
    }
    break;
    case STOP :break;
}
}while(!end);

textbackground(1);

window(2,3,79,9);

clrscr();

window(1,1,80,25);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าในรูปแบบใด ๆ ก็ตามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void tempo(void)
```

```
{ char ch[4]=" ",i,j,x1,x2,y1,y2;
```

```

void *ptr;

x1=38;
x2=38+14;
y1=2;
y2=4;

ptr=(void*)malloc(1024);
show_window(x1,y1,x2,y2,ptr);

gotoxy(x1+3,y1+2);

printf("Tempo = ");

for(i=0; i<3 && ch[i]!='\n'; )
    { ch[i]=getch();
if(ch[i]>='0' && ch[i]<='9')
    { putchar(ch[i]);
    i++;
    }
    }

tempo_rate = atoi(ch);
puttext(x1+1,y1+1,x2+1,y2+1,ptr);

free(ptr);

program_8253();
}

void show_window(int x1,int y1,int x2,int y2,char *ptr)
{ int i,j;

gettext(x1+1,y1+1,x2+1,y2+1,ptr);
for(i=x1;i<=x2;i++)

for(j=y1;j<=y2;j++)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ภายใต้กฎหมายที่ออกมามีผลตั้งแต่วันที่ ๑๖ ตุลาคม ๒๕๖๓ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใด ทั้งสิ้น อีกทั้งห้ามแก้ไขหรือดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*(vga+i*2+j*160)=0;
    for(i=x1;i<=x2;i++){
        *(vga+i*2+y1*160)='_';
        *(vga+i*2+y2*160)='_';
    }
    for(i=y1;i<=y2;i++){
        *(vga+x1*2+i*160)='|';
        *(vga+x2*2+i*160)='|';
    }
    *(vga+y1*160+x1*2)='┌';
    *(vga+y1*160+x2*2)='┐';
    *(vga+y2*160+x1*2)='└';
    *(vga+y2*160+x2*2)='┘';
    for(i=x1;i<=x2;i++)
        for(j=y1;j<=y2;j++)
*(vga+i*2+j*160+1)=0x70;
}

```

```

char pull_down(pull *menu){
    char ch;
    void *ptr;
    int x1,x2,y1,y2;
    int i,j;

```

```

    if(menu->width==0)
        return 13;

```

```

    x1=menu->x_pos;
    x2=menu->x_pos+menu->width+1;
    y1=menu->y_pos;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

y2=y1+menu->max+1;

ptr=(void*)malloc(1024);

gettext(x1+1,y1+1,x2+1,y2+1,ptr);

for(i=x1;i<=x2;i++)
    for(j=y1;j<=y2;j++)
        *(vga+i*2+j*160)=0;

for(i=x1;i<=x2;i++){
    *(vga+i*2+y1*160)='_';
    *(vga+i*2+y2*160)='_';
}

for(i=y1;i<=y2;i++){
    *(vga+x1*2+i*160)='|';
    *(vga+x2*2+i*160)='|';
}

*(vga+y1*160+x1*2)='┌';
*(vga+y1*160+x2*2)='┐';
*(vga+y2*160+x1*2)='└';
*(vga+y2*160+x2*2)='┘';

for(i=0;i<menu->max;i++)
    { gotoxy(x1+3,y1+2+i);

printf("%s",menu->command[i]);

}

for(i=x1;i<=x2;i++)
    for(j=y1;j<=y2;j++)
*(vga+i*2+j*160+1)=0x70;

do{
    for(i=x1+1;i<x2;i++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*(vga+(y1+menu->pos+1)*160+i*2+1)=0x07;

    ch=readfunc();

    for(i=x1+1;i<x2;i++)

        *(vga+(y1+menu->pos+1)*160+i*2+1)=0x70;

    if(funckey)

        switch(ch){

case 72:if(menu->pos>0)menu->pos--;

else menu->pos=menu->max-1;

break;

case 80:if(menu->pos<menu->max-1)menu->pos++;

else menu->pos=0;

break;

}

}while((((funckey)&&((ch==75)||((ch==77))))&&((((!funckey)

&&((ch==27)||((ch==13))))));

if(menu->width)

    ( puttext(x1+1,y1+1,x2+1,y2+1,ptr);

    free(ptr);

}

return(ch);

}

void ranad_on(int no){

    int i;

    for(i=2;i<3;i++){

        *(vga+160*10+17+i*160+(no-1)*6)=0x40;

        *(vga+160*10+19+i*160+(no-1)*6)=0x40;

    }

}

void ranad_off(int no){

    int i;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

void ranad_off(int no) { ปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int i;
```

```

for(i=2;i<3;i++){
    *(vga+160*10+17+i*160+(no-1)*6)=0x70;
    *(vga+160*10+19+i*160+(no-1)*6)=0x70;
}
}

```

```

void main_menu(void){

```

```

    int i,j;

```

```

    int menu_pos=0;

```

```

    int pull_act=0;

```

```

    int end=0;

```

```

    char ch;

```

```

    draw_main_menu();

```

```

do{

```

```

    for(i=0;i<strlen(menu_text[menu_pos])+2;i++)

```

```

        *(vga+(menu_xpos[menu_pos]-2)*2+i*2+1)=0x0F;

```

```

    if((!pull_act)|| (p_menu[menu_pos].max==0))

```

```

        ch=readfunc();

```

```

    else

```

```

        ch=pull_down(&p_menu[menu_pos]);

```

```

    for(i=0;i<strlen(menu_text[menu_pos])+2;i++)

```

```

        *(vga+(menu_xpos[menu_pos]-2)*2+i*2+1)=0x70;

```

```

    if(funckey){

```

```

        switch(ch){

```

```

            case 75:if(menu_pos>0)menu_pos--;

```

```

            else menu_pos=max_menu-1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

break;

case 77:if(menu_pos<max_menu-1)menu_pos++;

else menu_pos=0;

break;

}

}

else{

switch(ch){

case 13:if(p_menu[menu_pos].max==0){

// Execute Command

pull_act=0;

switch(menu_pos){

case 1:translate();

draw_main_menu();

break;

case 3:play_();

draw_main_menu();

break;

case 4:tempo();break;

}

}

else{

if(!pull_act)pull_act=1;

else{

//Execute Menu Command

switch(menu_pos){

case 0:switch(p_menu[menu_pos].pos){

case 0:load();break;

case 1:save();break;

case 2:new_();break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    case 1:end=1;break;
}
break;
case 2:switch(p_menu[menu_pos].pos){
    case 0:record();
draw_main_menu();
break;
    case 1:playback();
draw_main_menu();
break;
}
break;
}
pull_act=0;
}
break;
case 27:if(pull_act)pull_act=0;
else end=1;
break;
}
}
}while(!end);
}

```

```

void draw_main_menu(void)

```

```

{ int i,j;
for(j=1;j<25;j++)
for(i=0;i<80;i++){
*(vga+j*160+i*2+1)=0x1F;
*(vga+j*160+i*2)=0;

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

    for(i=2;i<24;i++){
*(vga+i*160)='|';
*(vga+i*160+158)='|';
}

    for(i=1;i<79;i++){
*(vga+i*2+160)='=';
*(vga+i*2+3840)='=';
}

    for(i=0;i<80;i++)
*(vga+i*2)=0;

        for(i=0;i<max_menu;i++){
gotoxy(menu_xpos[i],1);
printf("%s",menu_text[i]);
}

        for(i=0;i<80;i++)
*(vga+i*2+1)=0x70;

*(vga+160)='|';
*(vga+318)='|';
*(vga+3840)='|';
*(vga+3998)='|';

gotoxy(30,2);
textcolor(4);
textbackground(2);
cprintf(" MIDI Translation ");
for(i=0;i<7;i++){
for(j=0;j<strlen(ranad[i]);j++){
*(vga+160*9+16+j*2+i*160)=ranad[i][j];
*(vga+160*9+17+j*2+i*160)=0x70;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ควรนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้.

```
}  
}
```

```
void interrupt inc_timer(void) {
```

```
    unsigned char port21h;
```

```
    port21h=inportb(0x21)&0xF7;
```

```
    outportb(0x21,0xFF);
```

```
    main_tick++;
```

```
    SAVE_IRQ3();
```

```
    asm sti;
```

```
    inportb(0x21);
```

```
    outportb(0x21,port21h);
```

```
}
```

```
void init_sio(void){
```

```
    asm cli;
```

```
    asm nop;
```

```
    outportb(0x2b1,0x18); asm nop;
```

```
    outportb(0x2b1,0x04); asm nop;
```

```
    outportb(0x2b1,0xc4); asm nop;
```

```
    outportb(0x2b1,0x03); asm nop;
```

```
    outportb(0x2b1,0xc1); asm nop;
```

```
    outportb(0x2b1,0x05); asm nop;
```

```
    outportb(0x2b1,0x68); asm nop;
```

```
    outportb(0x2b1,0x01); asm nop;
```

```
    outportb(0x2b1,0x00); asm nop;
```

```
    asm sti;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์และบุคลากรที่ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

```
void midiout(unsigned char midi_data)
```

```
{
```

```
    char chk_buffer;
```

```
    do{
```

```
        outportb(control_port,0);
```

```
        chk_buffer=inportb(control_port)&0x04;
```

```
    } while (chk_buffer==0 );
```

```
    outportb(data_port,midi_data);
```

```
}
```

```
void program_8253(void){
```

```
    unsigned long int divider,div1,div2;
```

```
    divider=(unsigned int)41666u/tempo_rate*60;
```

```
    div1=divider%256;
```

```
    div2=(unsigned char)(divider/256);
```

```
    outportb(0x2B3,0x76);
```

```
    outportb(0x2A3,2);
```

```
    outportb(0x2A3,0);
```

```
    outportb(0x2B3,0x36);
```

```
    outportb(0x2A2,div1);
```

```
    outportb(0x2A2,div2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

```

void interrupt active_sense(void){
    int active_count=0;

    active_count++;
    if(active_count==5){
        midiout(Active_sense);
        active_count=0;
    }
    Save_Int1C();
}

void far *getmem(unsigned long size) {
    if((ptr=(char far *)farcalloc(size,sizeof(char)))!=NULL){
        printf("Not enough memory !\n");
        printf("Program Abort\n");
        exit(0);
    }
    head_mem = tail_mem = ptr;
    return(ptr);
}

```

```

void startup(){
    unsigned char port21h;

```

```

    program_8253();

```

```

    init_sio();

```

```

    Save_Int1C=getvect(0x1C);

```

```

    SAVE_IRQ3=getvect(0x0B);

```

```

setvect(0x0B,inc_timer);

setvect(0x1C,active_sense);

port21h=inportb(0x21);

port21h&=0xF7;

outportb(0x21,port21h);

}

int size;

void record()
{ char *save_scr=malloc(200);
  FILE *fp;

  do { show_window(20,3,50,7,save_scr);
    gotoxy(24,6);
    printf("File name = ");
    gets(string);
    fp = fopen(string,"wb");
    if(fp==NULL)
      { gotoxy(24,7);
        printf("Can't open file.");
      }

    } while(fp==NULL);

  puttext(20+1,3+1,50+1,7+1,save_scr);

  free(save_scr);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
translate();
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(tail_mem != head_mem)

```

```

        { size = tail_mem - head_mem;
fwrite(head_mem,size,1,fp);
        }
        fclose(fp);
        tail_mem = head_mem;
}

```

```

void playback(void) {
    unsigned int main_time,time_tick;
    unsigned char data,data1,data2;
    char far *i;
    FILE *fp;

// fp = fopen("Sampling.dat","rb");
// ptr = head_mem;
// fread(ptr,size,1,fp);
// fclose(fp);

    if(tail_mem==head_mem)
        return;

    main_tick = 0;
    for (i=head_mem;i<tail_mem;i=i+3) {
        data1 = *ptr; ptr++;
        data2 = *ptr; ptr++;
        do { main_time = main_tick;
            time_tick = ((data1<<8) | data2);
            if (main_time>time_tick)
                { printf("TIME ERROR");
                }
        } while(1);
        return;
    }
}

```



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    } while ( !( (((main_time & 0xff00)>>8)==data1)&&((main_time & 0x00ff)==d
        do { midiout(*ptr);

ptr++;i++;

data = *ptr;

} while ( data != 0xFF);

ptr++;

if (func(27))

{ head = ptr;

return;

}

}

return;

}

int forward(void) {

unsigned char data1,data2,data3;

unsigned int main_time;

main_time = main_tick;

// gotoxy(10,8);

// printf("ptr0 = %4d,tail_mem = %4d \n",ptr,tail_mem);

if (ptr == tail_mem)

{ gotoxy(20,23);

printf("This is the end of this song \n");

return 0;

}

}

do {

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

delay(48);

main_time++;

```

```

ptr++;

}while( !((func(80))||(ptr==tail_mem)) );

if (ptr==tail_mem) {
gotoxy(20,23);
printf("This is the end of this song \n");
return 0;
}

do { data1 = *ptr; ptr--;
data2 = *ptr; ptr--;
data3 = *ptr; ptr=ptr+3;
}

while ( !((data1==0xFF)&&((data2&0x80)==0)&&((data3&0x80)==0)) );
if (ptr==tail_mem){
gotoxy(20,23);
printf("This is the end of this song");
return 0;
}
else { head = ptr;
main_tick = 0;
}

return 0;
}

```

```

int reverse(void) {
unsigned char data1,data2,data3;
unsigned int main_time;

main_time = main_tick;

```

เอกสารนี้เป็นเอกสารที่สาม ใช้สำหรับนักเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (ptr == head_mem)
        { gotoxy(20,23);
printf("This is the start of this song");
main_tick = 0;
return 0;
    }
do {
delay(48);
main_time--;
ptr--;
gotoxy(70,10);
textcolor(4);
cprintf("%d",main_time);
    }while( !(((func(80))!:(ptr==head_mem)) ) );
if (ptr==head_mem) {
head = ptr;
main_tick=0;
}
else { do { data1 = *ptr; ptr--;
data2 = *ptr; ptr--;
data3 = *ptr; ptr=ptr+3;
    } while ( !(((data1==0xFF)&&((data2&0x80)==0)&&((data3&0x80)==0)) ) );
head = ptr; main_tick = 0;
}
return 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int func(int ascii){
```

```

    if (kbhit())
    {
        if (getch()==asci)
            return(1);
        else return(0);
    }
    else return(0);
}

int play(void) {
    unsigned int main_time;
    unsigned char data,data1,data2;
    char far *i;
    ptr = head ;
    main_tick = 0;
    for (i=head;i<tail_mem;i=i+3) {
        data1 = *ptr; ptr++;
        data2 = *ptr; ptr++;
        do main_time = main_tick;
            while ( !(((main_time & 0xff00)>>8)==data1)
                &&((main_time & 0x00ff)==data2)) ) ;
        do { midiout(*ptr);
            ptr++;i++;
            data = *ptr;
        }
        while ( data != 0xFF);
        ptr++;
        if (func(27))
            { head = ptr;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะในรูปแบบใดก็ตาม หากมีให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

}

// printf("\n Already play");
return 0;
}

int a[J],b[J];
int status[J];
void translate(void)
{
    int count[21];
    int REF[21] = {2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,5,5,5,2};
    int NOISE0[21]={15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,17,17,17};
    int NOISE1[21]={40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40};
    int NOISE2[21] = {40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,38};
    unsigned char z0=0,z1=0;
    int x0=0,x1=0;
    clrscr(); nosound();

    /* CLEAR ARRAY */
    menu();
    for (i=0;i<J;i++)
{ a[i] = 0;
  b[i] = 0;
  status[i] = 0;
  count[i]=0;
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการแก้ไขหรือดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* LOOP TAKE VALUE */
do
{
    /* first take value */
for (i=0;i<J;i++)
{
z0 = 0x80+i;
z1 = 0x40+i;
outportb(OPP,i);
outportb(OPP,z0);
outportb(OPP,z1);
delay(2);
x0 = inportb(INPH) & 0xf8;
x1 = inportb(INPL) & 0x07;
a[i] = x0 | x1;
if((a[i]>NOISE0[i])&&(a[i]<NOISE1[i])&&(b[i]==0))
    a[i]=0;
}

/* loop compare */
for(i=0;i<J;i++)
{
    if (a[i]<=REF[i])
    {
if (status[i] == NOTE_ON)
    { status[i] = NOTE_OFF;

    check_status();
ranad_off(i);
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
// nosound();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

b[i] = 0;

textcolor(4);

gotoxy(68,4+i);

printf("%d ",b[i]);

    if((b[i]<0) && (b[i]>10))

b[i] = MINIMUM;

    if((b[i]<0) && (b[i]>10))

b[i] = MIN;

    if((b[i]<0) && (b[i]>10))

b[i] = MIDDLE;

    if((b[i]<0) && (b[i]>10))

b[i] = MAXIMUM;
}

    else if ((a[i] > REF[i]) && (a[i] > b[i]))
{
b[i] = a[i];
}

    else if ((a[i] > REF[i]) && (a[i] < b[i]))
{

```

```

if((status[i] != NOTE_ON))

```

```

{

```

```

    status[i] = NOTE_ON;

```

```

    check_status();

```

```

    ranad_on(i);

```

```

    if((b[i]<40) && (b[i]>0))

```

ไม่ว่า b[i] = MINIMUM; ห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if((b[i]<100) && (b[i]>39))

```

```

b[i] = MIN;

```

```

        if((b[i]<150) && (b[i]>99))
b[i] -= MIDDLE;
        if((b[i]<255) && (b[i]>149))
b[i] = MAXIMUM;
        textcolor(4);
        gotoxy(68,4+i);
        cprintf("%d ",b[i]);
        //      sound(200+i*100);
        //delay(50);
    }

```

```

else if( (a[i] < NOISE2[i]) )
{
status[i] = NOTE_OFF;
check_status();
ranad_off(i);
    //  nosound();
b[i] = 0;
textcolor(4);
gotoxy(68,4+i);
cprintf("%d ",b[i]);
}

```

```

}

```

```

}

```

```

}while(!(func(27)));

```

```

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void box(int r,int c,int k, int m)

```

```

{

```

```

int i;

gotoxy(c,r); printf("%c",0xc9);
for(i=c+1;i<m;i++) {gotoxy(i,r); printf("%c",0xcd);}
gotoxy(m,r); printf("%c",0xbb);
for(i = r+1;i<k;i++) { gotoxy(m,i); printf("%c",0xba);}
gotoxy(m,k); printf("%c",0xbc);
for(i = m-1;i>c;i--) { gotoxy(i,k); printf("%c",0xcd);}
gotoxy(c,k); printf("%c",0xc8);
for(i = k-1;i>r;i--) { gotoxy(c,i); printf("%c",0xba);}

}

void menu(void)
{
int i=0;
int NC211 = {52,53,55,57,59,61,63,64,66,68,70,72,74,75,77,79,81,82,84,86,88,90,92,94,96,98,100};
clrscr();
textbackground(11);
clrscr();
box(1,6,25,75);
textcolor(4);
gotoxy(9,2);
cprintf("NO\n");
textcolor(5);
gotoxy(16,2);
cprintf("NOTE CODE\n");
textcolor(6);
gotoxy(30,2);
cprintf("A/D START ON\n");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
textcolor(8);
gotoxy(48,2);
cprintf("VELOCITY RATIO\n");
textcolor(4);
gotoxy(66,2);
cprintf("INPUT\n");
```

```
for(i=0;i<5;i++)
```

```
{
textcolor(4);
gotoxy(9,4+i);
cprintf("%d",i);
gotoxy(35,4+i);
cprintf("32");
gotoxy(55,4+i);
cprintf("80");
gotoxy(18,4+i);
cprintf("%d",N[i]);
}
```

```
for(i=5;i<9;i++)
```

```
{
textcolor(5);
gotoxy(9,4+i);
cprintf("%d",i);
gotoxy(35,4+i);
cprintf("32");
gotoxy(55,4+i);
cprintf("80");
gotoxy(18,4+i);
cprintf("%d",N[i]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    for(i=9;i<13;i++)
    {
textcolor(8);

    gotoxy(9,4+i);
    cprintf("%d",i);
    gotoxy(35,4+i);
    cprintf("32");
    gotoxy(55,4+i);
    cprintf("80");
    gotoxy(18,4+i);
    cprintf("%d",N[i]);
    }

    for(i=13;i<17;i++)
    {
textcolor(9);

    gotoxy(9,4+i);
    cprintf("%d",i);
    gotoxy(35,4+i);
    cprintf("32");
    gotoxy(55,4+i);
    cprintf("80");
    gotoxy(18,4+i);
    cprintf("%d",N[i]);
    }

    for(i=17;i<21;i++)
    {
textcolor(12);
    gotoxy(9,4+i);
    cprintf("%d",i);

```



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    cprintf("%d",i);

```

```

gotoxy(35,4+i);
cprintf("32");
gotoxy(55,4+i);
cprintf("80");
gotoxy(18,4+i);
cprintf("%d",N[i]);
    }
}

int note[J] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
int volume[J] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
void check_status(void) {
    unsigned int main_timer=0;
    unsigned char Htime,Ltime;

    if (main_tick != main_timer)
        { main_timer = main_tick;
Htime = (main_timer & 0xff00)>>8;
setmem(ptr,1,Htime); ptr++;
Ltime = (main_timer & 0x00ff);
setmem(ptr,1,Ltime); ptr++;
        }

    else ptr = ptr-1;

    switch (status[i]) {
        case NOTE_ON:
setmem(ptr,1,0x90); ptr++;
        midiout(0x90);

setmem(ptr,1,code[i]); ptr++; note[i]=code[i];

```

```
midout(code[i]);  
setmem(ptr,1,b[i]); ptr++; volume[i]=b[i];  
midout(b[i]);  
setmem(ptr,1,0xFF); ptr++;  
break;
```

```
case NOTE_OFF:  
setmem(ptr,1,0x80); ptr++;  
midout(0x80);  
setmem(ptr,1,note[i]);ptr++;  
midout(note[i]);  
setmem(ptr,1,volume[i]); ptr++;  
midout(volume[i]);  
setmem(ptr,1,0xFF); ptr++;  
tail_mem = ptr;  
break;  
}
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้