สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การเชื่อมต่อสัญญาณดนตรีแบบดิจิตอลกับกลอง MIDI DRUMS

นางสาว ณัฐวดี สินซื่อสัตย์กุล
นาย ชนินทร์ ใบม่วง
นางสาว แทน คงชูญาติ

เลขหมู่.....
เลขทะเบียน 119235
วัน,เดือน,ปี - 6 ธ.ค. 2554

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อสัญญาณดนตรีแบบดิจิตอลกับกลอง MIDI DRUMS

นาย ชนินทร์	ใบม่วง	50010303
นางสาว ณัฐวดี	สินซื่อสัตย์กุล	50010505
นางสาว แทน	คงชูญาติ	50010595

อาจารย์ที่ปรึกษา
รศ.ดร.สุรพันธุ์ เอื้อไพบูลย์

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์บัณฑิต
สาขาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MIDI DRUMS

Mr. Chanin Baimuang

Miss Nattawadee Sinsusatkul

Miss Tan Kongchooyart

THIS THESIS IS SUBMITTED IN PARTAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN ELECTRONICS ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงงาน MIDI DRUMS

จัดทำโดย

นาย ชนินทร์ ใบม่วง เลขประจำตัว 50010303

นางสาวณัฐวดี สินซื่อสัตย์กุล เลขประจำตัว 50010505

นางสาวแทน คงชูญาติ เลขประจำตัว 50010595

อาจารย์ที่ปรึกษา

รศ.ดร. สุรพันธุ์ เอื้อไพบูลย์

รายงานฉบับนี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ.....อาจารย์ที่ปรึกษา

(รศ.ดร. สุรพันธุ์ เอื้อไพบูลย์)

วันที่ 21/3/54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อสัญญาณดนตรีแบบดิจิตอลกับกลอง

นาย ชนินทร์ ใบม่วง 50010303
นางสาว ณัฐวดี สินซื่อสัตย์กุล 50010505
นางสาว แทน คงชูญาติ 50010595
รศ.ดร. สุรพันธุ์ เอื้อไพบูลย์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2553

บทคัดย่อ

ปริญญาฉบับนิพนธ์นี้ เป็นการนำเสนอการออกแบบการเชื่อมต่อสัญญาณแบบดิจิตอลกับกลอง โดยใช้เซนเซอร์รับความสั่นสะเทือนที่กลองส่งมายังไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์จะทำหน้าที่บันทึกค่า และแปลงรหัสเป็นสัญญาณดนตรีแบบดิจิตอล และส่งไปยังเครื่องซินธิไซเซอร์เพื่อถอดรหัสออกเป็นเสียงดนตรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MIDI DRUMS

Mr. Chanin	**Baimuang**	**ID.50010303**
Miss Nathawadee	**Sinsusatkul**	**ID.50010505**
Miss Tan	**Kongchooyart**	**ID.50010595**
Assist. Prof. Dr. Surapan	**Airphaiboon**	**Advisor**

Education Year 2010

Abstract

In this thesis, the design and construction of MIDI Drums are presented. The controller records signal received from drums vibration sensor, before change it to MIDI code and send it to synthesizer. The synthesizer will decode and put through music sound.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานนี้ ได้รับความช่วยเหลืออย่างดียิ่งจาก ผศ.ดร.สุรพันธุ์ เอื้อไพบูลย์ ซึ่งเป็นอาจารย์ปรึกษาโครงงาน ที่ได้ให้คำแนะนำ และให้ข้อคิดเห็นในการทำโครงงาน จึงขอขอบพระคุณอาจารย์ที่กรุณาสละเวลา พร้อมทั้งให้คำแนะนำที่เป็นประโยชน์อย่างยิ่ง

ขอขอบพระคุณบิดา-มารดาของผู้จัดทำ ซึ่งได้ให้กำลังใจและสนับสนุนการทำโครงงานอย่างดีมาก ผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณเป็นอย่างสูง คุณค่าและประโยชน์อันพึงมีในรายงานฉบับนี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

.....

(นาย ชนินทร์ ใบม่วง)

.....

(นางสาว ณัฐวดี สินซื่อสัตย์กุล)

.....

(นางสาว แทน คงชูญาติ)

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	i
ABSTRACT	ii
กิตติกรรมประกาศ	iii
สารบัญ	iv
สารบัญรูปภาพและตาราง	vi
บทที่ 1 บทนำ	
1.1 ความเป็นมาของโครงงาน	1
1.2 วัตถุประสงค์	1
1.3 ขั้นตอนการทำงาน	1
1.4 ขอบเขตของโครงงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 ปัญหาที่เกิดขึ้นและการแก้ไข	2
บทที่ 2 ทฤษฎีและหลักการทำงาน	
2.1 หลักการทำงานของเซนเซอร์เปียโซ	3-5
2.2 สวิทช์ปุ่มกด	5-7
2.3 การทำงานของระบบ MIDI	7-9
2.4 ไมโครคอนโทรลเลอร์	9
2.5 กลองชุด	10
2.6 SD Card	10-13
2.7 SPI Interface	14-15
บทที่ 3 หลักการทำงานและการออกแบบ	
3.1 แนวทางการออกแบบ	16
3.1.1 จุดประสงค์ก่อนออกแบบ	16
3.1.2 โครงสร้างของฮาร์ดแวร์	16
3.1.2.1) ส่วนวงจรรับแรงสั่นสะเทือน	16
3.1.2.2) ส่วนวงจรไมโครคอนโทรลเลอร์	17
3.1.2.3) ส่วนวงจรบันทึกข้อมูล	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ขั้นตอนการออกแบบ 17
3.2.1 กลองและส่วนประกอบของกลอง 18
3.2.2 วงจรกลองมิดิ 19
3.2.2.1 วงจรภาคส่ง 19-20
3.2.2.2 ภาครับ 20-21
3.2.3 Flow Chart การทำงานของโปรแกรมที่ได้ออกแบบ 22-23
3.2.4 การบันทึกข้อมูลไปยัง SD Card 24-25

บทที่ 4 การทดลองและผลการทดลอง
4.1 ส่วนของเซนเซอร์เปียโซ 26
4.1.1.1 เปรียบเทียบขนาดสัญญาณที่ได้รับจากวัสดุที่ใช้เคาะชนิดต่างๆ 26-30
4.1.1.2 เปรียบเทียบสัญญาณที่ได้รับการเคาะเมื่อมีพลาสติกหุ้ม 31
4.1.1.3 เปรียบเทียบขนาดสัญญาณที่ได้รับจากระยะทางที่ใช้เคาะ 32-33
4.1.1.4 เปรียบเทียบขนาดสัญญาณที่ได้รับการเคาะบนพื้นผิวที่แตกต่างกัน 33-35
4.2 การทดลองการอ่านและเขียนข้อมูลลงบน SD Card 36
4.2.1 การทดลองอ่านและเขียนข้อมูลลงบน SD Card 37-40
4.2.2 การทดลองการส่ง MIDI ไปบันทึกลงบน SD Card โดย ตรวจจับสัญญาณจาก Piezo เพียงตัวเดียว 41-44
4.2.3 การทดลองบันทึกค่าลงบน SD Card โดย ทำการตรวจจับสัญญาณอนาลอกจากเปียโซเซนเซอร์หลายตัว 45-50

บทที่ 5 สรุปผลการทดลอง 51
เอกสารอ้างอิง 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป		หน้า
รูปที่ 2.1	ปรากฏการณ์เพียโซอิเล็กตริกในผลึกควอทซ์	4
รูปที่ 2.2	เพียโซอิเล็กตริกเซนเซอร์เมื่อมีขั้วอิเล็กโทรดบนวัสดุที่มีความเป็นผลึก	5
รูปที่ 2.3	แสดงสวิตซ์ปุ่มกด	5
รูปที่ 2.4	แสดงโครงสร้างและการทำงานของสวิทซ์ปุ่มกดปกติเปิด	6
รูปที่ 2.5	แสดงโครงสร้างและการทำงานของสวิทซ์ปุ่มกดปกติปิด	6
รูปที่ 2.6	แสดงโครงสร้างและการทำงานของสวิทซ์กดปกติเปิดและปกติปิดอยู่รวมกัน	7
รูปที่ 2.7	SD Card	11
รูปที่ 2.8	การต่อวงจรระหว่างขาของ SD Card กับ Arduino	12
รูปที่ 2.9	การแสดงหมายเลขขาของ SD Card	13
รูปที่ 2.10	แสดงถึงการ Interface ระหว่าง Master และ Slave	15
รูปที่ 2.11	แสดงการทำงานแบบ Chip Select	15
รูปที่ 3.1	Block Diagram การทำงานทั้งหมด	16
รูปที่ 3.2.1.1	ส่วนประกอบของกลอง	18
รูปที่ 3.2.1.2	กลองไฟฟ้า	18
รูปที่ 3.2.2	วงจรกลองมิดิ	19
รูปที่ 3.2.2.2	Pad ที่ใช้สำหรับตีกลองฝังด้วยแผ่นเปียโซ	19
รูปที่ 3.2.2.3	Foot Switch	20
รูปที่ 3.2.3	Flow Chart	22-23
รูปที่ 3.2.4	Voltage Divider	24
รูปที่ 3.2.5	การเชื่อมต่อระหว่าง SD Card กับ Arduino	25
รูปที่ 3.2.6	ภาพกลอง MIDI แบบเสร็จสมบูรณ์	25
รูปที่ 3.2.7	ภาพเครื่องซินธิไซเซอร์	25
รูปที่ 4.1	สัญญาณที่ใช้นิ้วในการเคาะด้วยความแรงมาก	26
รูปที่ 4.2	สัญญาณที่ใช้นิ้วเคาะด้วยความแรงปานกลาง	27
รูปที่ 4.3	สัญญาณที่ใช้นิ้วเคาะด้วยความแรงน้อย	27
รูปที่ 4.4	สัญญาณที่ใช้เหล็กปลายแหลมเคาะด้วยความแรงมาก	28
รูปที่ 4.5	สัญญาณที่ใช้เหล็กปลายแหลมเคาะด้วยความแรงปานกลาง	28
รูปที่ 4.6	สัญญาณที่ใช้เหล็กปลายแหลมเคาะด้วยความแรงน้อย	29
รูปที่ 4.7	สัญญาณที่ใช้ไม้กลองเคาะด้วยความแรงมาก	29
รูปที่ 4.8	สัญญาณที่ใช้ไม้กลองเคาะด้วยความแรงปานกลาง	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.9	สัญญาณที่ใช้ไม้กลองเคาะด้วยความแรงน้อย	30
รูปที่ 4.10	สัญญาณที่ใช้ไม้กลองเคาะด้วยความแรงมากและหุ้มด้วยพลาสติก	31
รูปที่ 4.11	สัญญาณที่ใช้ไม้กลองเคาะด้วยความแรงน้อยและหุ้มด้วยพลาสติก	31
รูปที่ 4.12	สัญญาณที่เคาะห่างจากเปียโซเป็นระยะทาง 1 ซ.ม.	32
รูปที่ 4.13	สัญญาณที่เคาะห่างจากเปียโซเป็นระยะทาง 4 ซ.ม.	32
รูปที่ 4.14	สัญญาณที่เคาะห่างจากเปียโซเป็นระยะทาง 7 ซ.ม.	33
รูปที่ 4.15	สัญญาณที่ใช้นิ้วเคาะโดยเปียโซวางอยู่บนโต๊ะ	33
รูปที่ 4.16	สัญญาณที่ใช้ไม้กลองเคาะโดยเปียโซวางอยู่บนโต๊ะ	34
รูปที่ 4.17	สัญญาณที่ใช้ไม้กลองเคาะโดยเปียโซวางอยู่บนซิลิโคน	34
รูปที่ 4.18	สัญญาณที่ใช้ไม้กลองเคาะโดยเปียโซวางอยู่บนแผ่น CD	35
รูปที่ 4.19	กราฟสัญญาณ MIDI	35
รูปที่ 4.2	การต่อ SD Card กับ Arduino	36
รูปที่ 4.2.1	ผลการทดลองตอนที่ 1	40
รูปที่ 4.2.2	ผลการทดลองตอนที่ 2	44
รูปที่ 4.2.3	ผลการทดลองตอนที่ 3	50

สารบัญตาราง

หน้า

ตารางที่ 1 การเชื่อมต่อกับ SD Card แบบ SD Mode และ แบบ SPI Mode 13

ตารางที่ 2 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์และ SD Card 24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1) ความเป็นมาของโครงงาน

โลกปัจจุบันเทคโนโลยีได้มีการพัฒนาอย่างต่อเนื่องเพื่อตอบสนองต่อความต้องการของมนุษย์ที่มีอยู่อย่างไม่สิ้นสุดในทุกๆด้าน รวมไปถึงการตอบสนองทางด้านอารมณ์และความรู้สึก ซึ่งถ้าพูดด้านอารมณ์และความรู้สึกของมนุษย์แล้วสิ่งที่จะขาดไม่ได้ คือ เสียงดนตรี ซึ่งเกิดจากเครื่องดนตรีหลากหลายประเภท เสียงดนตรีก็มีหลากหลายเสียงเช่นกันซึ่งถูกกำหนดด้วยเสียงตัวโน๊ต ได้แก่ โด เร มี ฟา ซอล ลา ที สำหรับโครงการนี้เป็นโครงการที่จะช่วยเพิ่มลูกเล่นให้กับการเล่นเครื่องดนตรีระหว่างกลองและคีย์บอร์ด คือ เมื่อกลองถูกตีก็จะเกิดเสียงที่คีย์บอร์ดเช่นกัน โดยผ่านทางตัวเชื่อมต่อสัญญาณดิจิตอลระหว่างเครื่องดนตรีหรือที่เรียกว่า มิดิ (MIDI) ซึ่งก็คือสิ่งที่ถูกทำขึ้นในโครงการนี้ เพื่อเพิ่มลูกเล่นให้กับการเล่นดนตรีและความบันเทิงให้แก่ผู้รับฟัง ซึ่งจะทำให้ผู้รับฟังได้รับความสนุกสนานมากยิ่งขึ้นไป

1.2) วัตถุประสงค์

1.2.1) เพื่อนำเอา Sensor วัดความสั่นสะเทือน(เปียโซ) มาประยุกต์ใช้ทำเป็นกลองไฟฟ้า

1.2.2) เพื่อศึกษาการเชื่อมต่ออุปกรณ์ดนตรีแบบดิจิตอล

1.2.3) เพื่อศึกษาการทำงานของ Microcontroller MEGA1280 และภาษา Arduino

1.2.4) เพื่อศึกษาการทำงานของ SD Card

1.3) ขั้นตอนการทำงาน

1.3.1) ศึกษาหลักการทำงานและปรากฏการณ์ของ Piezoelectric sensor

1.3.2) ศึกษาวงจร Piezoelectric sensor เพื่อนำมาใช้ในส่วนของวงจรภาคส่งสัญญาณ

1.3.3) ศึกษาโปรแกรมที่ใช้ในการควบคุมการทำงานของ MIDI drum

1.3.4) เขียนโปรแกรมภาครับและควบคุมการทำงาน

1.3.5) ศึกษาหลักการทำงานของ SD Card

1.3.6) ทดลองโปรแกรมในการควมคุมการทำงาน

1.3.7) นำส่วนของวงจรและโปรแกรมมาทดลองภาพรวมทั้งหมด

1.3.8) ทำการแก้ไขให้ตรงตามจุดประสงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4) ขอบเขตของโครงงาน

ขอบเขตของโครงงานนี้แบ่งการทำงานเป็น 2 ส่วนคือในส่วนแรกจะเป็นวงจรภาคส่ง ซึ่งในส่วนนี้จะทำหน้าที่ตรวจจับการสั่นสะเทือนที่เกิดขึ้นจากการเคาะกลอง หรือ ตรวจจับการเหยียบ pedal แล้วเปลี่ยนเป็นความต่างศักย์ ค่าสัญญาณที่ได้นี้จะเข้าไปในส่วนที่สองต่อไปคือ ส่วนรับสัญญาณเพื่อประมวลผล โดยจะรับสัญญาณจากเซนเซอร์และสวิตซ์ นำไปแปรผลเป็นความดังเบาในการเคาะ ซึ่งค่าความดังเบาที่ได้นี้จะให้ผลเป็นความเร็วจากนั้นในส่วนไมโครคอนโทรลเลอร์จะส่งข้อมูลออกเป็นสัญญาณเสียงของกลองและความดังเบาของกลอง ผ่านระบบ MIDI ของเปียโนไฟฟ้า หรือ ซินธิไซเซอร์

1.5)ประโยชน์ที่คาดว่าจะได้รับ

1. ทราบถึงหลักการทำงานของกลองไฟฟ้า
2. พัฒนาทักษะในการเขียนโปรแกรม
3. เรียนรู้หลักการของปรากฏการณ์เปียโซอิเล็กตริก
4. ได้รับความรู้และทักษะในการทำงาน
5. สามารถนำไปใช้งานจริงได้
6. เรียนรู้หลักการทำงานของ SD Card
7. ได้รู้จักการทำงานร่วมกันและสร้างสามัคคีในหมู่คณะ

1.6) ปัญหาที่เกิดขึ้นและการแก้ไข

1. โปรแกรมที่ใช้งานเป็นโปรแกรมที่มีความเฉพาะตัว ทำให้ต้องใช้เวลาในการศึกษามากพอสมควร การแก้ไขโดยหาข้อมูลเพิ่มเติมและศึกษาโปรแกรมพื้นฐานเพื่อเป็นตัวอย่างที่สามารถใช้ประยุกต์ในโปรแกรมที่จะเขียนต่อไป
2. ความไม่ระมัดระวังในการทำงาน ทำให้เกิดข้อผิดพลาดเมื่อทำการทดลอง จึงต้องมีการตรวจสอบวงจรให้แน่ชัดอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

ในโครงการนี้นั้นการเชื่อมต่อสัญญาณดิจิตอลระหว่างเครื่องดนตรี คือ การนำเปียโซเซนเซอร์มาใช้ร่วมกับหน่วยควบคุมขนาดไมโคร หรือไมโครคอนโทรลเลอร์ ซึ่งเซนเซอร์เปียโซจะทำหน้าที่ในการสร้างสัญญาณเมื่อกลองถูกตี และส่งสัญญาณที่ได้ไปในรูปของสัญญาณไฟฟ้าขนาดต่างๆกัน ขึ้นอยู่กับแรงที่มากระทำบนกลองและตำแหน่งของตัวกลองนั้นๆ ในชุดกลอง ให้กับหน่วยควบคุมขนาดไมโครเพื่อทำการประมวลผลจากสัญญาณที่ได้รับและส่งสัญญาณที่ถูกประมวลผลแล้วให้กับคีย์บอร์ด และตัวคีย์บอร์ดก็จะให้กำเนิดเสียงที่แตกต่างกันไปขึ้นอยู่กับสัญญาณที่ได้รับมา

2.1) หลักการทำงานของเซนเซอร์เปียโซ

สารเปียโซอิเล็กทริกเซรามิกคือ สารเซรามิกชนิดหนึ่งมีสมบัติเปียโซอิเล็กทริก (piezoelectric effect) ซึ่งสมบัติเปียโซอิเล็กทริกนี้ถูกค้นพบครั้งแรกโดย Pierre Curie และ Jacques Curie เมื่อปี ค.ศ. 1880 ในการทดลองทางฟิสิกส์ของผลึกบางชนิด เช่น แร่ควอทซ์, Rochelle salt, ซิงก์เบรน (zincblende) และทัวมาลีน (tourmaline) ฯลฯ แต่เนื่องจากผลึกแบบเดียว (single crystal) ยังมีความเสถียรต่ำต่ออุณหภูมิ จนกระทั่งปี ค.ศ. 1940 จึงได้มีการพัฒนาสารเปีย โซอิเล็กทริกเซรามิก (Piezoelectric ceramics) ขึ้น ซึ่งมีข้อดีกว่าสารเปียโซอิเล็กทริกแบบเดียว คือ

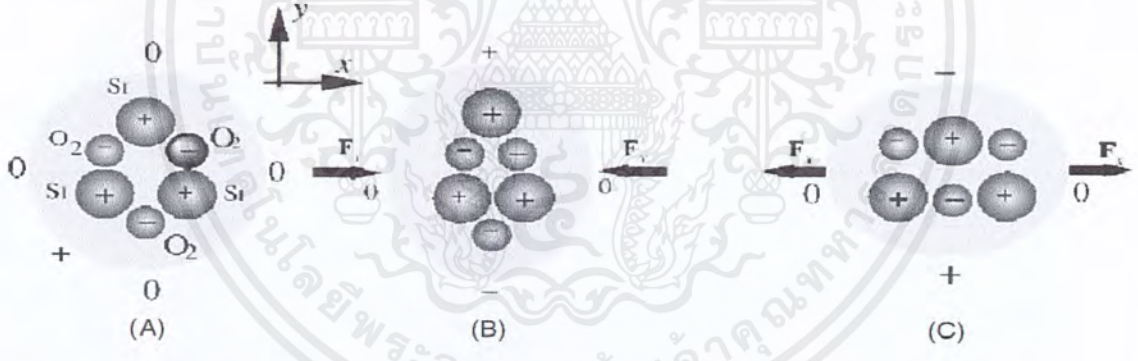
- ประสิทธิภาพการเปลี่ยนรูปพลังงาน Electromachanical สูง
- สามารถตกแต่งรูปทรงได้ตามต้องการ โดย Mach inability
- สามารถปรับคุณสมบัติด้วยการปรับปรุงส่วนประกอบของเนื้อสาร (High degree of freedom in characteristics design)
- มีเสถียรภาพสูง
- สามารถนำไปผลิตเป็นอุตสาหกรรมจำนวนมากๆ ได้

ทำให้มีการพัฒนาสารเปียโซอิเล็กทริกเซรามิกไปประยุกต์ใช้งานทางด้านอัลตร้าโซนิกส์ทรานสดิวเซอร์ (Ultrasonic transducer) อย่างมากมาย เช่นเซนเซอร์ตรวจหาปลา(fishfinders), เครื่องทำความสะอาดด้วยคลื่นอัลตร้าโซนิกส์, เครื่องเชื่อมพลาสติกด้วยเครื่องอัลตร้าโซนิกส์ ฯลฯ สำหรับการพัฒนาสารเปียโซอิเล็กทริกเซรามิกไปประยุกต์ใช้งานการวิเคราะห์สมบัติทางด้านไฟฟ้าของสารนับเป็นสิ่งสำคัญเพื่อให้สารโซเปียอิเล็กทริกเซรามิกถูกนำไปใช้งานได้อย่างเต็มประสิทธิภาพ และวิธีการอธิบายสมบัติทางไฟฟ้าของสารที่นิยม ก็คือการใช้วงจรไฟฟ้าสมมูลแบบ Butterworth Van Dyke หรือ BVD ซึ่งมีสมบัติทางไฟฟ้าที่ใกล้เคียงกับสารเปียโซอิเล็กทริกเซรามิกในช่วงความถี่เรโซแนนซ์โดยวงจร BVD สามารถคำนวณหาได้จากข้อมูลอิมพิแดนซ์ของสารเปียโซอิเล็กทริกในช่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่เรโซแนนซ์ (resonance frequency) นั่นเองเพื่อให้นักวิจัยและพัฒนาสารเปียโซอิเล็กทริกเซรามิกได้เข้าใจถึงพื้นฐานของสมบัติทางไฟฟ้าของสารเปียโซอิเล็กทริกเซรามิกก่อนนำไปประยุกต์ใช้งาน งานวิจัยนี้นำเสนอวิธีการหาวงจร BVD ของสารเปียโซอิเล็กทริกเซรามิกชนิดเลดเซอร์โคเนตไททาเนต (Lead-Zerconate-Titanate) หรือ PZT ซึ่งผลิตด้วยกระบวนการทางเซรามิกแบบดั้งเดิม ลักษณะเป็นแผ่นกลม ผิวหน้าด้านบนและด้านล่างของสารถูกเคลือบด้วยโลหะเงิน มีรูปแบบการสั่นแบบ thickness mode หรือรูปแบบการสั่นในทิศทางขนานกับความหนาของสาร PZTเมื่อสารถูกกระตุ้นด้วยแรงดันไฟฟ้ากระแสสลับปรากฏการณ์เพียโซอิเล็กตริก (Piezoelectric effect)

ปรากฏการณ์เพียโซอิเล็กตริกเป็นกระบวนการที่เกี่ยวข้องกับประจุไฟฟ้าโดยความเป็นผลึกของวัสดุเมื่อเกิดความเครียดที่วัสดุนั้นๆ ซึ่งมักพบในวัสดุประเภทควอทซ์(สูตรทางเคมี SiO2) และวัสดุโพล(poled) ซึ่งเป็นวัสดุที่สร้างขึ้นโดยทำให้มีสมบัติความเป็นโพลาไรซ์ ที่พบอยู่ทั่วไปคือเซรามิกและโพลีเมอร์บางชนิด ตัวอย่างเช่น โพลีไวนีลีดีน ฟลูออไรด์ และอาจกล่าวได้ว่าวัสดุเพียโซอิเล็กตริกมีสมบัติเป็นเฟอโรอิเล็กตริกซึ่งมีความคล้ายคลึงกับสมบัติแม่เหล็กที่เป็นแบบเฟอโรแมกเนติก โดยคำว่า piezo มาจากภาษากรีกว่า piezen ซึ่งหมายถึงการกดและปรากฏการณ์เพียโซอิเล็กตริกในผลึกควอทซ์ถูกค้นพบโดยพี่น้องตระกูล Curie ในปี 1880 แต่ยังไม่แน่ชัดนักจนกระทั่งในปี 1917 ศาสตราจารย์ P.Langevin ทำการตัดผลึกควอทซ์ในระนาบ x ในการตรวจสอบคลื่นเสียงภายในน้ำ ซึ่งงานวิจัยดังกล่าว ได้ถูกพัฒนาไปยังเรื่องของโซนา(Sonar)

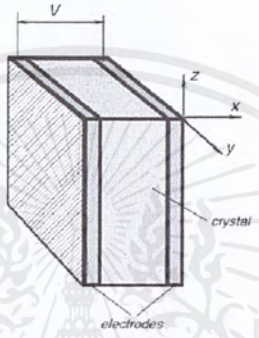


รูปที่ 2.1 ปรากฏการณ์เพียโซอิเล็กตริกในผลึกควอทซ์

รูปแบบของเพียโซอิเล็กตริกทำให้ชัดเจนยิ่งขึ้นโดย A. Meissner โดยเขาทำการสร้างโมเดลที่เรียกว่า Helix ขึ้น ดังรูป 1A ซึ่งแทนด้วยอะตอมของซิลิกอนและอะตอมของไฮโดรเจน โดยอะตอมจะเรียงตัวแบบสลับที่กันในโมเดลของ Helix ในระนาบของแกน x, y และ z จากรูปเป็นการมองรูปแบบโมเดลจากแกน z ลงบนระนาบ xy ในขณะสภาวะปกติ เมื่อถูกกระตุ้นด้วยแรงภายนอก Fx ดังแสดงในรูป 1B จะพบว่าอะตอมมีการเลื่อนตำแหน่งในบริเวณที่ถูกกดโดยแรง Fx ทำให้อะตอมของซิลิกอนหนึ่ง อะตอมเลื่อนไปอยู่ด้านบนและอะตอมของออกซิเจนเลื่อนมาอยู่ด้านล่างทำให้เกิดสภาวะการมีขั้วขึ้น และเมื่อออกแรงดึงด้วยขนาด Fx ทำให้เกิดสภาพขั้วในทิศทางที่กลับกันอันเนื่องมาจากอะตอมของออกซิเจนสองอะตอมอยู่ด้านบน และอะตอมของซิลิกอนสองอะตอมเลื่อนมาอยู่ด้านล่าง ดังรูป 1C การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

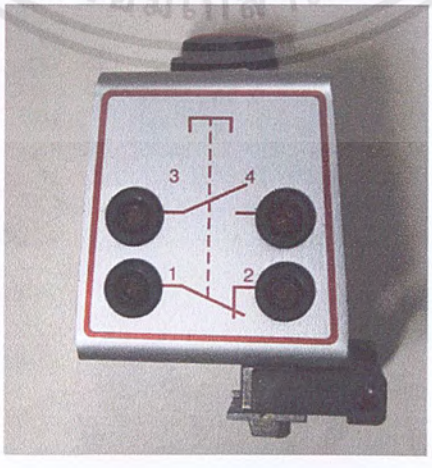
เกิดประจุดังกล่าวจะเกิดความเป็นขั้วในทิศทางของแกน y ซึ่งโมเดลนี้แสดงให้เห็นถึงกลไกการเกิดประจุ ไฟฟ้าของวัสดุที่มีความเป็นผลึกเมื่อถูกบีบหรืออัดด้วยแรงค่าหนึ่งซึ่งจะคล้ายคลึงกับการเกิดปรากฏการณ์ของไพโรอิเล็กตริกซึ่งการทำให้เกิดประจุไฟฟ้าโดยหลักการของเพียโซอิเล็กทริกนั้นทำได้โดยการทำขั้วไฟฟ้าไว้ บริเวณด้านข้างของวัสดุเพียโซทั้งสองด้านดังแสดงในรูปที่ 2.2 แสดงถึงเพียโซอิเล็กตริกเซนเซอร์ซึ่งมี ความเป็นตัวเก็บประจุไฟฟ้าที่มีค่าไดอิเล็กตริกของวัสดุซึ่งกล่าวได้คือเป็นผลึกเพียโซอิเล็กตริก เมื่อเกิด สภาวะความเป็นขั้วไฟฟ้าขึ้นจะเกิดความต่างศักย์ไฟฟ้า V ระหว่างขั้วทั้งสองของตัวเก็บประจุ



รูปที่ 2.2 เพียโซอิเล็กตริกเซนเซอร์เมื่อมีขั้วอิเล็กโทรดบนวัสดุที่มีความเป็นผลึก

2.2) สวิทซ์ปุ่มกด (push button switch)

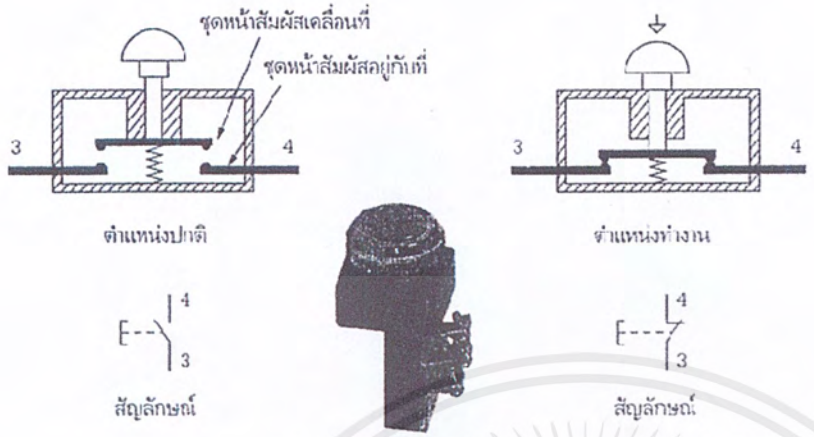
ปุ่มหน้าสัมผัส (contact) ทำจากตัวนำไฟฟ้า เช่น ทองแดง อะลูมิเนียม โครงสร้างภายนอกเป็นฉนวน เช่น พลาสติกมีก้านกระทุ้ง โดยมีแรงภายนอกมากระทำ เช่น มือกด เมื่อปล่อยมือจะกลับสู่ตำแหน่งปกติ มี 3 ชนิด คือ สวิทซ์ปุ่มกดปกติเปิด สวิตซ์ปุ่มกดปกติปิด และสวิตซ์ปุ่มกดปกติเปิดและปกติปิดอยู่ในตัวเดียวกัน ดังแสดงในรูปที่ 2.3



รูปที่ 2.3 แสดงสวิตซ์ปุ่มกด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

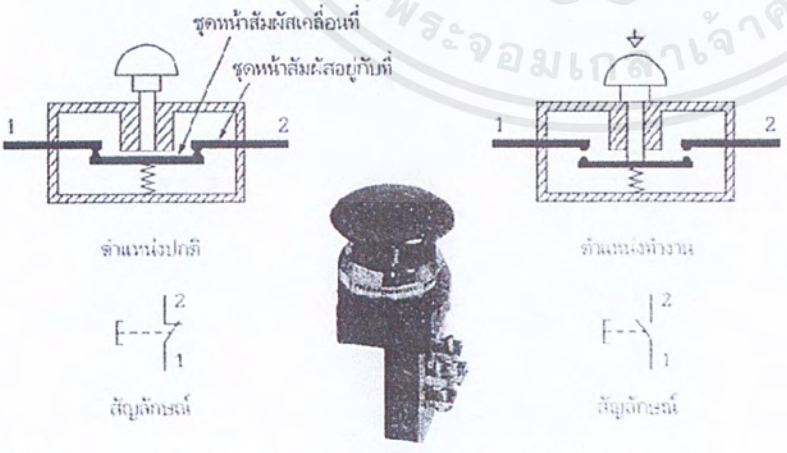
ก. สวิทซ์ปุ่มกดปกติเปิด มีโครงสร้างและการทำงาน ดังแสดงในภาพที่ 2.4



รูปที่ 2.4 แสดงโครงสร้างและการทำงานของสวิทซ์ปุ่มกดปกติเปิด

<u>หลักการทำงาน</u> : สวิตซ์ปุ่มกดปกติเปิด ในตำแหน่งปกติกระแสไฟฟ้าไม่สามารถไหลผ่านได้เพราะปุ่มหน้าสัมผัสไม่ต่อถึงกัน เมื่อถูกกดทำให้ปุ่มหน้าสัมผัสแตะถึงกัน และทำให้กระแสไฟฟ้าไหลผ่านไปได้ เมื่อปล่อยมือ แรงสปริงจะดันให้ปุ่มสัมผัสกลับสู่ตำแหน่งปกติ ใช้เมื่อเริ่มทำงาน

ข. สวิทซ์ปุ่มกดปกติปิด มีโครงสร้างและการทำงาน ดังแสดงในภาพที่ 2.5

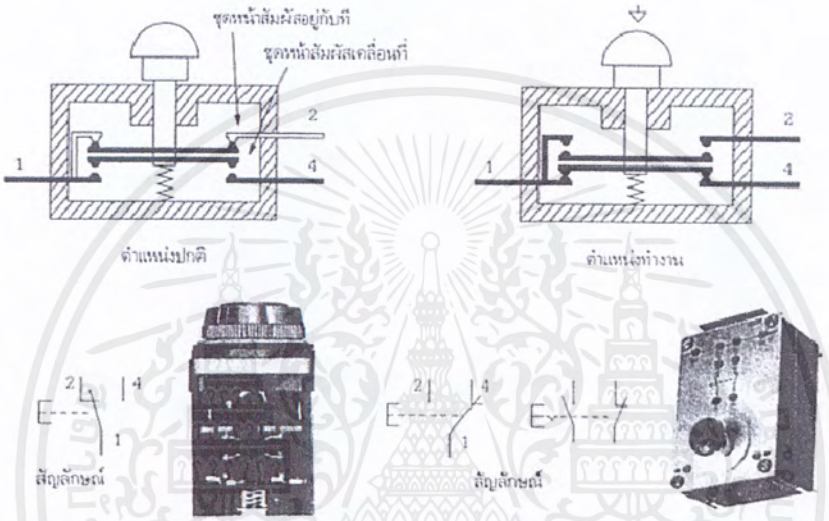


รูปที่ 2.5 แสดงโครงสร้างและการทำงานของสวิทซ์ปุ่มกดปกติปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงาน : สวิตซ์ปุ่มกดปกติปิด ในตำแหน่งปกติกระแสไฟฟ้าสามารถไหลผ่านได้ เพราะปุ่มหน้าสัมผัสต่อถึงกัน เมื่อถูกกดทำให้ปุ่มหน้าสัมผัสออกจากกัน กระแสไฟฟ้าไม่สามารถไหลผ่านไปได้ เมื่อปล่อยมือ แรงสปริงจะดันให้ปุ่มสัมผัสกลับสู่ตำแหน่งปกติ ใช้สำหรับหยุดทำงาน

ค. สวิตซ์ปุ่มกดปกติเปิดและปกติปิดอยู่ในตัวเดียวกัน เหมาะสำหรับใช้ในการป้องกันการทำงานของอุปกรณ์ทำงาน เช่น ขดลวด รีเลย์ หรือวาล์ว ไม่ให้ซ้ำซ้อนกัน คือ ถ้าตัวใดตัวหนึ่งทำงาน จะป้องกันไม่ให้อีกตัวทำงาน ดังแสดงในภาพที่ 2.6



รูปที่ 2.6 แสดงโครงสร้างและการทำงานของสวิทซ์ปุ่มกดปกติเปิดและปกติปิดอยู่รวมกัน

หลักการทำงาน : เป็นการรวมเอาสวิตซ์ปุ่มกดปกติเปิดและปกติปิดมารวมอยู่ในตัวเดียวกัน เมื่อถูกกดจะทำงานสลับกัน คือ ปกติเปิดเป็นปกติปิด ปกติปิดมาเป็นปกติเปิด ใช้สำหรับป้องกันวงจรที่ไม่ต้องการให้ทำงานพร้อมกัน

2.3) การทำงานของระบบ MIDI

การรับ-ส่งข้อมูลของระบบ MIDI มีความสัมพันธ์กับ 2 สิ่งคือ PortและChannelโดยมีความหมายคือ Port หมายถึงประตูสำหรับผ่านเข้าออกของข้อมูลในระบบ MIDI การเดินทางของข้อมูลต้องผ่านประตูของอุปกรณ์ เช่นทางออกจากอุปกรณ์หลัก ไปสู่ทางเข้าของอุปกรณ์รอง ในเบื้องต้นมีมาตรฐานว่าอุปกรณ์เครื่องหนึ่งต้องมี 1 พอร์ต ตัวอย่างเช่นเครื่องซินธิไซเซอร์มีประตูให้ข้อมูลเข้าเพียงช่องเดียวเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในความหมายที่กล่าวมาคือ อุปกรณ์หลักสามารถส่งข้อมูลให้อุปกรณ์รองได้เพียงช่องทางเดียวเท่านั้น แต่จากการพัฒนาการทางด้านอุปกรณ์ทำให้มีการผลิตอุปกรณ์ที่มีทางออกได้หลายประตู ตัวอย่าง เช่น วงจรเสียง (Sound Card) หรือวงจรต่อเชื่อมสัญญาณ (Interface Card) ที่มีช่องออกได้หลายช่อง (Port 1, Port 2, Port 3, etc) กรณีอย่างนี้สามารถส่งข้อมูลไปสู่อุปกรณ์หลายเครื่องให้ทำหน้าที่แตกต่างกันได้

Channel หมายถึงช่องเฉพาะสำหรับใช้ส่งข้อมูลคำสั่งต่าง ๆ ในระบบ MIDI ในวงจรเสียงและวงจรต่อเชื่อมสัญญาณที่เป็นมาตรฐานมีช่องกำหนดไว้ 16 ช่อง มาตรฐานนี้ใช้กับอุปกรณ์ดนตรีทุกชนิด ดังนั้นจะพบว่าซินธิไซเซอร์ 1 เครื่องก็จะรับส่งได้ 16 ช่อง วงจรเสียงก็รับส่งได้ 16 ช่อง ด้วยการรับส่งข้อมูลดนตรี 16 ช่องนี้ ทำให้อนุมานได้ว่าในการรับ-ส่งข้อมูล ณ เวลาหนึ่งสามารถรับ-ส่งเสียงเครื่องดนตรีต่างกันได้ 16 ชนิดอุปกรณ์ที่มีมากกว่า 1 พอร์ตย่อมส่งข้อมูลได้มากกว่า 16 ช่อง แต่ต้องใช้อุปกรณ์รับข้อมูลมากกว่า 1 เครื่อง

มิดิ หรือ มาตรฐานการประสานเครื่องดนตรีแบบดิจิทัล (อังกฤษ: Music Instrument Digital Interface: MIDI) เป็นโพรโทคอลมาตรฐานที่คิดค้นขึ้นเมื่อปี พ.ศ. 2525 โดยเป็นระบบการติดต่อสื่อสารทางดนตรี ของอุปกรณ์อิเล็กทรอนิกส์ทางดนตรี เช่น คอมพิวเตอร์ ซินธิไซเซอร์ ซีเควนเซอร์ ซาวด์โมดูล แซมเพลอร์ ซึ่งใช้สัญญาณไฟฟ้าแบบดิจิตอล ในการส่งข้อมูลระหว่างอุปกรณ์ที่เชื่อมต่อ โดยจะมีความหมายเป็นโน้ตดนตรี และค่าการควบคุมลักษณะเสียงต่างๆ

มาตรฐานเกี่ยวกับ MIDI อันแรกมีชื่อเรียกว่า The General MIDI System Level 1 หรือเรียกกันทั่วไปว่า มาตรฐาน GM มีข้อกำหนดให้คลังเสียงต้องมีมาตรฐานเดียวกัน ดังนี้

- เสียงที่ดังพร้อมกัน ณ เวลาหนึ่ง ไม่น้อยกว่า 24 เสียง (24 voice polyphony)
- ทุกเสียงต้องตอบสนองแรงกด (Velocity)
- โน้ต Middle C อยู่ที่ตำแหน่ง MIDI Note 60
- ต้องมีมาตรฐาน 16 ช่อง (Channel)
- เสียงเครื่องประกอบจังหวะ (Percussion) ต้องอยู่ที่ช่อง 10 (Channel 10)
- เสียงกลองต้องตรงตามกำหนด MIDI note numbers
- ต้องมีเสียงดนตรีมาตรฐาน 128 เสียงตรงตามตารางเสียงมาตรฐาน
- แต่ละช่องต้องตอบสนอง Continuous Controllers (CC) เหล่านี้ 1 (mod wheel), 7 (volume),10 (pan), 11 (expression) และ 123 (all notes off)
- ค่า registered parameter numbers (ค่า MIDI Continuous Controllers) ต้องตอบสนองการใช้งาน Pitch-bend และ การปรับเสียง

ระบบมิดิ คือการใช้ภาษาหรือข้อตกลงร่วมกันในอันที่จะให้อุปกรณ์หลายชิ้นหรือต่างชนิดกันสามารถสื่อสารข้อมูลกันได้ แฟ้มข้อมูลมิดิ เป็นส่วนของการเตรียมข้อมูลที่จะให้เกิดงานที่เกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับเสียงขึ้นมา เช่นดนตรี แต่ข้อมูลมิดิ ไม่ได้เป็นอุปกรณ์ให้กำเนิดเสียง ดังนั้นการมีเฉพาะข้อมูลมิดิแล้วเรียกเข้ามาในเครื่องคอมพิวเตอร์ จึงไม่สามารถได้ยินเสียงได้ ในกระบวนการสื่อสารครบถ้วนของระบบมิดิ

เครื่องคอมพิวเตอร์ยังต้องการวงจรเสียง (Sound Card) ซึ่งจะเป็นแหล่งเสียงที่จะนำมาใช้ หรือเครื่องดนตรีดิจิตอลที่มีอินเทอร์เฟสเป็นตัวเชื่อม การรับ-ส่งข้อมูลระหว่างอุปกรณ์ที่กล่าวมา อาศัยสายมิดิ ระบบการต่อเชื่อมโดยทั่วไปมี 3 แบบคือ MIDI IN, MIDI Out และ MIDI Thru

2.4) ไมโครคอนโทรลเลอร์

บอร์ด ET-EASY MEGA1280 บอร์ด AVR ที่ รองรับการเขียนโปรแกรมด้วยภาษา C++ ของ Arduino คุณสมบัติของบอร์ด ET-EASY MEGA1280 (DUINO MEGA)

- ใช้ ATMEGA1280 เป็น MCU ประจำบอร์ด, RUN ความถี่ 16MHz, 100 PIN TQFP
- 128KBYTE FLASH (4KBYTE สงวนไว้สำหรับ BOOTLOADER), 8KBYTE SRAM / 4KBYTE EEPROM, โดยตัว MCU ติดตั้ง BOOTLOADER ของ Arduino Mega ไว้ให้แล้ว
- รองรับการพัฒนาโปรแกรมด้วยภาษา C++ ของ Arduino ในแบบ Arduino Mega ได้เต็ม 100%
- ขั้วต่อ USB MINI ใช้ USB BRIDGE ของ FTDI เบอร์ FT232RL พร้อมอุปกรณ์ OVER CURRENT PROTECTION สำหรับใช้ติดต่อสื่อสาร และ DOWNLOAD CODE ที่เขียนจากเครื่องคอมพิวเตอร์ มายังตัวบอร์ด, พร้อม JUMPER สำหรับปรับใช้งานบอร์ดเป็นการ PROGRAM BOOTLOADER ให้กับ MCU จาก PORT USB ในบอร์ดได้เองไม่ต้องใช้เครื่องโปรแกรม AVR ISP จากภายนอก
- 54 PIN DIGITAL I/O (5V TTL LOGIC) โดยมี 14 PIN สามารถทำการโปรแกรมหน้าที่เป็น PWM ได้
- 16 PIN ANALOG INPUT (เป็น A TO D ขนาด 10 BIT 16 ช่อง)
- 4 PORT UART (เป็น HARDWARE SERIAL PORT) แบบ 5V TTL LOGIC
- ขั้ว 10 PIN (HEADER IDE) ขนาด 8 BIT, DIGITAL I/O (D22 - D29) สำหรับเชื่อมต่อกับ LCD แบบตัวอักษร ของทาง อีทีที ได้ เช่น ET-CONV 10 TO LCD, ET-CONV SPI TO LCD และบอร์ด I/O แบบต่างๆ ของทาง อีทีที
- ขนาดของ PCB บอร์ด และตำแหน่งต่างๆ ของ PIN CONNECTOR จะตรงกับมาตรฐานบอร์ด Ardino Mega ได้ทั้งหมด
- POWER SUPPLY 7-22V ใช้ได้ทั้งแบบ AC หรือ DC โดยเลือกใช้ REGULATE แบบ SWITCHING LM2575-5 ลดปัญหาเรื่องของความร้อนเมื่อมีการใช้ไฟกระแสสูงๆ สามารถใช้แหล่งจ่ายไฟจาก PORT USB ก็ได้ ในกรณีใช้กระแสไม่เกิน 500 mA โดยมีวงจรเลือกใช้แหล่งจ่ายอัตโนมัติให้ โดยจะตัดการใช้ไฟเลี้ยงจาก USB โดยอัตโนมัติ เมื่อมีการต่อแหล่งจ่ายไฟจากภายนอกให้บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5) กลองชุด

กลองชุด เป็นเครื่องดนตรีประเภทตีกระทบ ประกอบด้วยกลองหลายใบ และฉาบ โดยใช้ผู้เล่นคนเดียว ถือไม้ตีกลองและฉาบทั้งสองมือ และใช้เท้าเหยียบกระเดื่อง เพื่อตีกลองใหญ่ และ Cymbals กลองชุดเป็นที่นิยมใช้กับงานดนตรีเกือบทุกประเภท

ส่วนประกอบเครื่องดนตรีในกลองชุด ประกอบด้วย

กลองเล็ก หรือ สะแนร์ดรัม (Snare drum) ประกอบด้วยแผงลวดขึงรัดผ่านผิวหน้ากลองด้านล่าง เพื่อให้เกิดเสียงกรอบ ๆ ดังแต๊ก ๆ ตัวกลองทำด้วยไม้หรือโลหะ และสามารถรัดให้หนังตึงด้วยขอบไม้ด้านบนและล่าง สามารถปลดสายสะแนร์เพื่อให้เกิดเสียงทุ้มดังตุ้มตุ้มได้ และตีกลองเล็กด้วยไม้ นิยมใช้กลองชนิดนี้ทั้งในวงดุริยางค์และวงดนตรี

กลองทอม (Tom-tom drum) หรือ เทเนอร์ดรัม (Tenor drum) มีขนาดใหญ่กว่าสะแนร์ดรัม เป็นกลองชนิดที่สร้างขึ้นโดยไม่ใช้สายสะแนร์ โดยทั่วไปบรรเลงในหมวดกลอง ใช้หัวไม้หุ้มสักหลาด

กลองใหญ่ หรือ กลองเบส (Bass drum) เป็นกลองที่มีขนาดใหญ่ที่สุด ประกอบด้วยตัวกลองที่ทำด้วยไม้และมีหนังกลองทั้งสองด้าน เสียงที่เกิดจากการตีกลองใหญ่จะไม่ตรงกับระดับเสียงที่กำหนดไว้ทางตัวโน้ต ตีด้วยไม้ที่มีสักหลาดหุ้ม ชนิดที่มีหัวที่ปลายทั้งสองข้าง ใช้เพื่อทำเสียงรัว

กลองทิมปานี (หรือกลองเค็ทเทิ้ลดรัม) เป็นกลองที่มีลักษณะเป็นหม้อกระทะ ซึ่งมีหน้าหนังกลองหุ้มทับอยู่ด้านบน เป็นกลองชนิดเดียวที่ขึ้นเสียงแล้วได้ระดับเสียงที่แน่นอน เมื่อคลายหรือขันหน้ากลองโดยไม่ว่าจะใช้วิธีขันสกรูหรือเหยียบเพดดัล (ที่เหยียบ) ก็ได้ ไม้ที่ใช้ตีมีการหุ้มนวมตรงหัวไม้ตี ตีได้ทั้งเป็นจังหวะและรัว ฉาบ หรือ เชมเบล (Cymbal) มีอยู่2ชนิดคือฉาบที่ใช้กับกลองชุดและฉาบที่ใช้เดิน

ไฮแฮท คือ ฉาบสองใบเหมือนกับฉาบในวงดุริยางค์ แต่มีขนาดเล็กกว่า โดยทั่วไปนิยมใช้ขนาดเส้นผ่าศูนย์กลาง 14-15 นิ้ว ฉาบทั้งสองใบนี้ไม่ใช้เชือกหนังร้อยสำหรับถือ เพราะมีขาตั้งรองรับใบที่หนึ่งใส่ลงบนขาตั้งโดยให้ด้านนูนอยู่ด้านล่าง จะมีแผ่นโลหะและสักหลาดรองรับ อีกใบหนึ่งใส่ลงบนขอตั้งโดยให้ด้านนูนอยู่ด้านบน มีที่ไขติดอยู่กับแกนของขาตั้ง โดยกะระยะให้ห่างกันพอประมาณ เพื่อไม่ให้ฉาบทั้งสองใบชิดติดกัน ช่วงล่างสุดมีกระเดื่องเหมือนกับกลองใหญ่สำหรับเหยียบให้ฉาบทั้งคู่กระทบ กัน ไฮแฮทมีหน้าที่คอยขัดจังหวะหรือช่วยหนุนกลองเล็ก เน้นจังหวะขัดให้กระชับยิ่งขึ้น

2.6) SD Card

คำว่า SD card หรือ SD ย่อมาจาก Secure Digital เป็นสื่อเป็บข้อมูลที่ได้รับความนิยมมาก ถึงแม้ว่ามันจะไม่ค่อยจะ secure แบบชื่อก็ตาม สำหรับ SD Card นั้นก็คือ nand flash memory NAND Flash ถูกพัฒนาขึ้นมาเพื่อเป็นหน่วยเก็บข้อมูลขนาดกลาง สำหรับกล้องถ่ายรูปดิจิตอล, โทรศัพท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ PDA คุณลักษณะของ NAND Flash ROM คือ มีราคาถูก เก็บข้อมูลได้มาก และมีอายุการใช้งานได้นานกว่า NOR Flash ROM แบบเดิมถึง 10 เท่า

ในส่วนของข้อด้อยของ NAND Flash ROM คือ มีความเร็วในการจัดการข้อมูลช้ากว่า เพราะเป็นโครงสร้างการจัดการแบบ execute in place (XIP) กล่าวคือ กระบวนการทำงานของเครื่องที่ใช้ NAND Flash ROM จำเป็นจะต้องอ่านข้อมูลจาก NAND Flash ROM ก่อน แล้วค่อยเรียกขึ้นไปทำงาน คล้ายกับการทำงานของฮาร์ดดิสในเครื่องพีซีเดสค์ทอป และ main memory ดังนั้น ส่วนของ main memory ที่ใช้ NAND Flash ROM จำเป็นจะต้องมีขนาดค่อนข้างใหญ่ เพื่อใช้สำรองระบบการจัดการข้อมูลแบบนี้ จึงทำให้พื้นที่ของส่วนความจำเพื่อใช้เก็บโปรแกรมและข้อมูลอื่นๆ จะเหลือน้อยลง

สำหรับข้อดีของ SD Card ก็คือ ไม่มีส่วนของการเคลื่อนไหวของกลไกต่างๆ ใน card จึงสามารถทำตกได้โดยไม่เสียหาย และไม่ต้องอาศัยพลังงานใดๆ ไปหล่อเลี้ยงเอาไว้ ไม่ว่าจะเป็น SD Card, Mini SD หรือ Micro SD มันก็คือหลักการเดียวกัน การทำงานเหมือนกันแต่จะต่างกันที่ขนาดรูปร่าง เพื่อให้สอดคล้องกับพื้นที่ของอุปกรณ์ที่จะใช้งานร่วม

ความจุในขณะนี้ card ส่วนมากจะมีทำขนาดรองรับมากถึง 2 GB กันแล้วแต่เรื่องของความจุของขนาด Card นั้น ไม่มีผลใดๆ ต่อการทำงานภายนอกจากได้พื้นที่การใส่ข้อมูลเพิ่มขึ้น ในส่วนเรื่องของความเร็วไม่ว่า Card จะมีพื้นที่ความจุเล็กหรือใหญ่นั้นไม่มีผลต่อ speed การทำงาน Card และ Card reader ส่วนใหญ่นั้นรองรับการทำงานที่ version 1.1 ในปัจจุบันมี Version ที่ 2.0 ออกมาแล้วเพียงแต่เครื่องในแบบ Windows Mobile ยังไม่รองรับ SD 2.0 เท่านั้นเอง

ความเร็วSD Card ในปัจจุบันมีการพัฒนาเรื่อง Speed ความเร็วของข้อมูล (SD Card แบบ Hi Speed) โดยชื่อของ Card พวก Hi Speed เหล่านี้ก็ตั้งกันแล้วแต่จะเรียกไม่ว่าจะเป็น Ultra, 133x, High Speed ซึ่งไม่มีมาตราฐานใดๆ ออกมารับรอง แล้วแต่ทางบริษัทของผู้จำหน่ายจะทำตลาด แล้วก็ตั้งชื่อให้ดูเหมือนว่าทำงานเร็วและแตกต่างกว่า SD Card แบบธรรมดา

ดังนั้น คงไม่สามารถบอกได้ว่า Ultra จะเร็วกว่า High Speed หรือไม่ แต่หากเปรียบเทียบ SD Card แบบ ธรรมดากับพวก SD Card ความเร็วสูงนั้นมันทำงานต่างกันอย่างเห็นได้ชัด เช่น หากการโอนถ่ายข้อมูลด้วย card ธรรมดาอาจจะใช้เวลา สัก 80 วินาที แต่พอมาใช้ SD card แบบความเร็วสูง การโอนถ่ายข้อมูลในขนาดความจุเดียวกันก็จะเหลือที่ประมาณ 30 วินาที

รูปที่ 2.7 SD Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

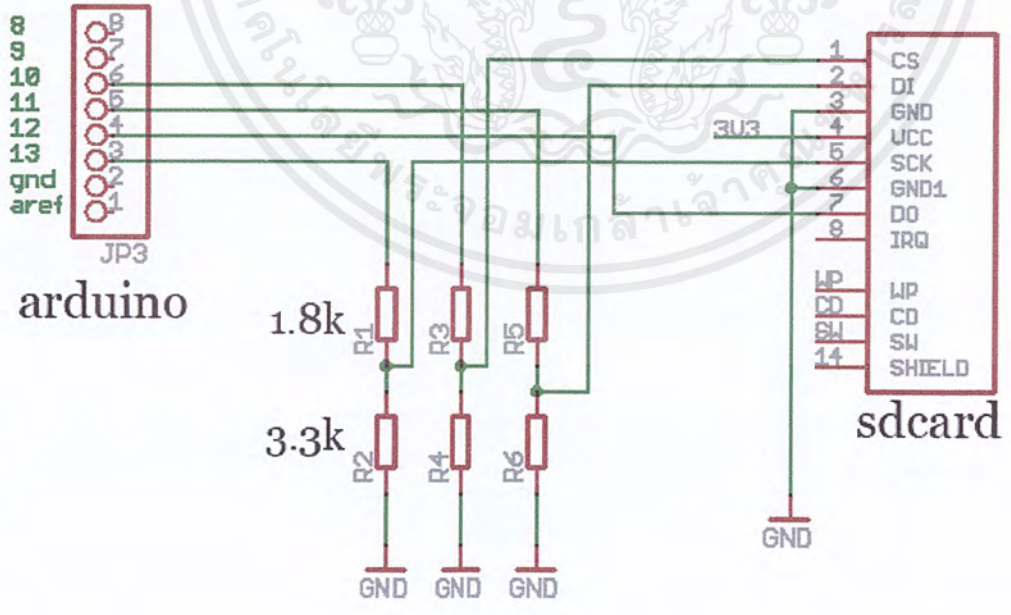
ข้อดีของ card ที่ทำงานด้วยความเร็วสูงก็คือ การแสดงผลต่างๆ ก็จะทำเร็วขึ้น การบันทึกข้อมูลก็เร็วขึ้น การโอนถ่ายข้อมูลก็เร็วขึ้น SD Card จะกินพลังงานไฟจากเครื่องก็ในช่วงที่มีการโอนถ่ายหรือเรียกใช้ข้อมูลภายใน card ดังนั้น ข้อดีอีกอย่างสำหรับ Card ที่ความเร็วสูงก็คือ พอเราโอนถ่ายข้อมูลด้วยเวลาอันสั้นเครื่องก็จะพลังงานไปกับการอ่านข้อมูลสั้นไปด้วยทำให้ประหยัดไฟมากขึ้น

การทำงานของ SD Card

ภายใน SD card จะมี microcontroller ขนาดเล็กคล้ายกับการทำงานของ CPU ทำหน้าที่คอยดูแลและติดต่อกับอุปกรณ์ต่างๆ ที่ใช้ SD Card ในขณะนั้น เช่น PDA Phone หรือ โทรศัพท์มือถือ โดย microcontroller จะทำหน้าที่จัดการเรื่องข้อมูลต่างๆ ที่ถูกอ่านและถูกเขียนของใน Flash โดย CPU หรือ microcontroller จะทำงานเฉพาะในช่วงที่มีการใช้งานข้อมูลใน card เท่านั้น

การทำงานของ SD Card จะเร็วหรือช้านั้นก็ขึ้นอยู่กับความเร็วของ CPU ในเครื่อง PDA Phone หรืออุปกรณ์ต่างๆ ด้วย เช่น เมื่อมีการใช้งาน SD Card บน โทรศัพท์ ตัวโทรศัพท์จะส่งข้อมูลแจ้งไปยัง SD card ว่า microcontroller นั้น สามารถรองรับความเร็วสูงสุดที่เท่าไร ก็ให้ใช้ความเร็วนั้น เพราะฉะนั้นหาก clock ของโทรศัพท์ยิ่งเร็ว การทำงานของ card ก็จะเร็วตามไปด้วย โดยข้อมูลจะถูกส่งในแต่ละวินาทีได้รวดเร็วขึ้น

ขาของ SD Card



รูปที่ 2.7 การต่อวงจรระหว่างขาของ SD Card กับ Arduino

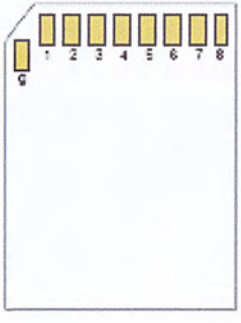
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดขาสัญญาณของ SD Card

การเชื่อมต่อกับ SD Card สามารถทำได้สองแบบก็คือ การเชื่อมต่อแบบ SD Mode และการเชื่อมต่อแบบ SPI Mode ดังรายละเอียดตามตารางต่อไปนี้

Pin	SD Mode			SPI Mode		
	Name	Type	Description	Name	Type	Description
1	CD/DAT3	I/O/PP	Card detection / Connector data line 3	CS	I	Chip selection in low status
2	CMD	PP	Command/Response line	DI	I	Data input
3	Vss1	S	GND	VSS	S	GND
4	Vdd	S	Power supply	VDD	S	Power supply
5	CLK	I	Clock	SCLK	I	Clock
6	Vss2	S	GND	VSS2	S	GND
7	DAT0	I/O/PP	Connector data line 0	DO	O/PP	Data output
8	DAT1	I/O/PP	Connector data line 1	RSV		
9	DAT2	I/O/PP	Connector data line 2	RSV		

ตารางที่ 1 การเชื่อมต่อกับ SD Card แบบ SD Mode และแบบ SPI Mode



รูปที่ 2.8 การแสดงหมายเลขขาของ SD Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7) SPI Interface

การสื่อสารแบบ SPI (Serial Peripheral Interface) จัดเป็นการสื่อสารอนุกรมแบบ Synchronous รูปแบบหนึ่งที่มีการออกแบบขึ้นมาเพื่อใช้งานกับระบบไมโครคอนโทรลเลอร์ต่างๆมาเป็นระยะเวลานานแล้ว โดยไมโครคอนโทรลเลอร์จะใช้ SPI เพื่อทำการติดต่อสื่อสารกับอุปกรณ์พ่วงต่างๆ รวมทั้งระหว่างไมโครคอนโทรลเลอร์กับไมโครคอนโทรลเลอร์ด้วยกัน ซึ่งมีการนำไปใช้งานในลักษณะต่างๆได้หลายอย่างเช่น ใช้ในการถ่ายเทข้อมูลระหว่างอุปกรณ์ต่างๆ การเชื่อมต่อ SPI จะต้องมี Master Device ตัวหนึ่งเสมอ ซึ่งโดยปกติจะเป็นไมโครคอนโทรลเลอร์เองที่เป็น Master Device ซึ่งจะเป็นตัวควบคุมอุปกรณ์พ่วงต่างๆ ลักษณะการทำงานของการสื่อสารอนุกรมแบบ Synchronous นั้น จะมีการแยกสัญญาณข้อมูล และสัญญาณนาฬิกา ออกจากกันโดยเด็ดขาด โดยมีสัญญาณที่ใช้ในการเชื่อมต่อ จำนวน 3 กลุ่มคือ

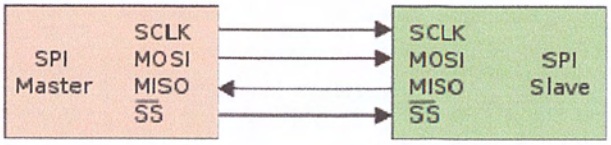
สัญญาณข้อมูล (Data) อาจมีจำนวน 1 สัญญาณหรือ 2 สัญญาณก็ได้ ขึ้นอยู่กับการออกแบบ เช่น ถ้าเป็นอุปกรณ์ที่ทำหน้าที่รับข้อมูลเพียงอย่างเดียวก็จะมีสัญญาณ Data Input เพียงเส้นเดียว แต่ถ้าเป็นอุปกรณ์ที่ทำหน้าที่ทั้งรับและส่งข้อมูลด้วยก็จะต้องมีสัญญาณ Data Input (SDI หรือ MISO) และ Data Output (SDO หรือ MOSI) แยกออกจากกันอย่างละ 1 สัญญาณ ซึ่งคือ Master In Slave Out (MISO) - Slave line สำหรับส่งข้อมูลไปยังตัวมาสเตอร์ และMaster Out Slave In (MOSI) - Master line สำหรับส่งข้อมูลไปยังอุปกรณ์พ่วง

สัญญาณนาฬิกา (Clock) หรือ Serial Clock (SCK) ซึ่งสัญญาณนี้จะมีความสำคัญมากคือจะใช้สำหรับทำหน้าที่ ซิงโครไนซ์กันระหว่างตัวรับและตัวส่งในการรับส่งข้อมูล โดยปรกติสัญญาณนี้จะถูกสร้างขึ้นโดยอุปกรณ์ที่ทำหน้าที่เป็น Master และอุปกรณ์ตัวที่ทำหน้าที่เป็น Slave นั้นจะทำการอ่านสัญญาณ Data ให้สอดคล้องสัมพันธ์กัน กับสัญญาณนาฬิกาที่ส่งมาจากตัว Master

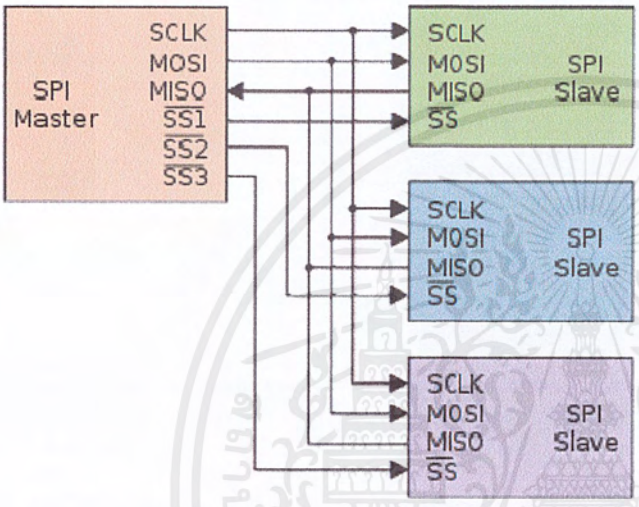
สัญญาณควบคุม (SS) หรือ Slave Select pin เมื่อเป็น Master สัญญาณนี้จะเป็น Output ใช้สำหรับเลือกการทำงานของอุปกรณ์ Slave ที่ต้องการจะติดต่อด้วย แต่ในอุปกรณ์ที่ทำหน้าที่เป็น Slave สัญญาณนี้จะเป็นตัวเลือก การทำงานให้กับอุปกรณ์ที่เป็นตัว Slave เพื่อให้เริ่มรับส่งข้อมูลกัน หรือเรียกได้อีกอย่างว่า SS เป็นขาที่อยู่บนอุปกรณ์แต่ละตัว ที่ตัวมาสเตอร์จะสามารถใช้เพื่อ enable และ disable อุปกรณ์นั้นๆได้อย่างเฉพาะเจาะจง เมื่ออุปกรณ์มี Slave Select pin เป็น low อุปกรณ์จะสามารถทำการติดต่อสื่อสารกับตัวมาสเตอร์ได้ และเมื่อเป็น high อุปกรณ์จะไม่สามารถติดต่อสื่อสารกับตัวมาสเตอร์ได้ ด้วยเหตุนี้เองเราจึงมาสามารถมีอุปกรณ์ SPI หลายตัว ที่สามารถใช้ MISO, MOSI, and CLK lines ร่วมกัน

ได้SPI คือการ interface อีกแบบหนึ่ง ย่อมาจาก Serial Peripheral Interface การ interface ของมัน จะมีขา MISO MasterIn SlaveOut MOSI MasterOut SlaveIn สองขานี้คือขาที่ส่งข้อมูลเข้าหากัน Clock สัญญาณนาฬิกาในการอ้างอิง SS หรือ Slave select เป็นขาที่ไว้ใช้เลือกอุปกรณ์ slave

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 แสดงถึงการ Interface ระหว่าง Master และ Slave



รูปที่ 2.10 แสดงการทำงานแบบ Chip Select

ส่วนการติดต่อการทำงานส่วนใหญ่จะต้องอ้างถึง datasheet เพราะการส่งข้อมูลจะขึ้นอยู่กับอุปกรณ์ว่ามันมีการติดต่อทีละกี่บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการทำงาน

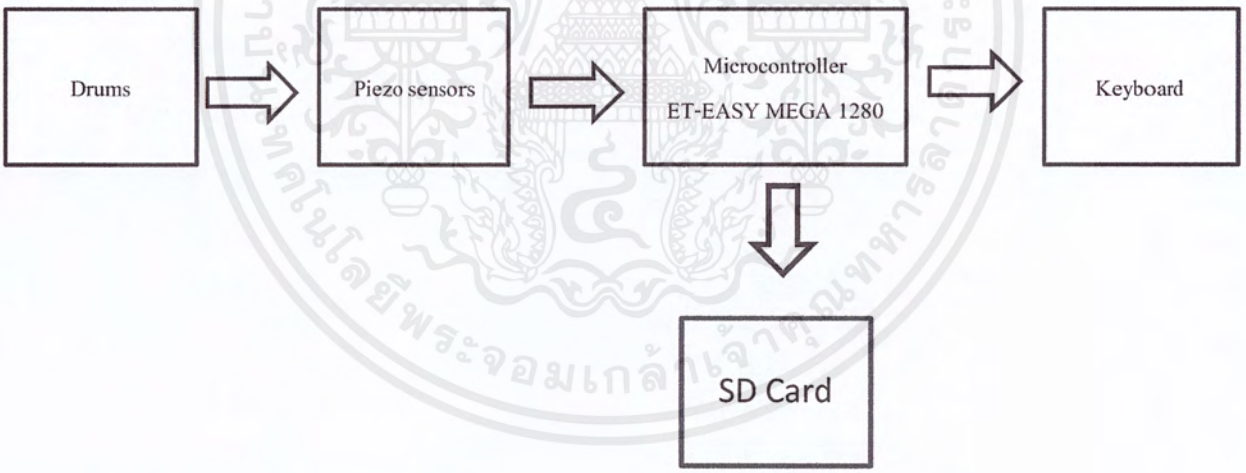
3.1) แนวทางการออกแบบ

3.1.1) จุดประสงค์ก่อนออกแบบ

1. ไมโครคอนโทรลเลอร์สามารถรับสัญญาณจากเซนเซอร์และสวิตซ์ได้
2. ควบคุมให้สามารถวัดสัญญาณระหว่างตัวรับกับจุดที่เคาะได้
3. นำสัญญาณที่รับได้ส่งเข้าไปยังไมโครคอนโทรลเลอร์ แล้วประมวลผลเป็นสัญญาณดิจิตอลตามรหัสมิดิที่ได้ตั้งไว้

3.1.2) โครงสร้างของฮาร์ดแวร์ประกอบด้วย 3 ส่วนใหญ่ ๆ คือ

- ส่วนวงจรรับแรงสั่นสะเทือน
- ส่วนวงจรไมโครคอนโทรลเลอร์
- ส่วนวงจรบันทึกข้อมูล



รูปที่ 3.1 Block Digram การทำงานทั้งหมด

3.1.2.1) ส่วนวงจรรับแรงสั่นสะเทือน

ในส่วนของวงจรรับแรงสั่นสะเทือนจะทำหน้าที่รับแรงกระทำที่เกิดจากการตีกลอง และแรงสั่นสะเทือนที่เกิดจากการตีกลองนั้นจะส่งแรงไปกระทำยังเซนเซอร์เปียโซ เมื่อเซนเซอร์เปียโซได้รับแรงสั่นสะเทือนจะเกิดสัญญาณไฟฟ้าขึ้นและส่งสัญญาณนั้นไปยังไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.1.2.2) ส่วนวงจรไมโครคอนโทรลเลอร์

ในส่วนของวงจรไมโครคอนโทรลเลอร์นี้จะทำหน้าที่ในการประมวลผลสัญญาณที่ได้รับจากเซนเซอร์เปียโซ และแปลงสัญญาณไฟฟ้าที่ได้รับนี้เป็นสัญญาณดิจิตอล และส่งสัญญาณที่ทำการแปลงเรียบร้อยแล้วไปยังพอร์ตมิดิอิน(MIDI IN) ที่ต่ออยู่ที่คีย์บอร์ดผ่านสายเคเบิล

3.1.2.3) ส่วนวงจรบันทึกข้อมูล

ไมโครคอนโทรลเลอร์จะทำหน้าที่บันทึกข้อมูลการตีกลองในแต่ละครั้งเป็นสัญญาณดิจิตอลเป็นรหัส MIDI 3 ไบท์ โดนข้อมูลนี้จะถูกเขียนลงในเอสดีการ์ด

3.2) ขั้นตอนการออกแบบ

ใช้หลักการ Piezoelectric Effect โดยใช้ปรากฏการณ์ที่แรงเชิลกลทำกับวัสดุทำให้เกิดศักดาไฟฟ้า

เมื่อมีการเคาะหรือตีเกิดขึ้น จะเกิดแรงสั่นสะเทือนไปยังแผ่นเปียโซ เกิดเป็นศักดาไฟฟ้า โดยที่ปลายทาง

จะมีตัวรับสัญญาณอยู่

เมื่อมีสัญญาณอินพุทจากเปียโซเข้ามายังขาอนาลอกอินพุทของ Arduino Board ไมโครคอนโทรลเลอร์จะวัดค่าสัญญาณ โดยความดังเบา(Dynamic) ของการเคาะจะให้ผลเป็นความเร็ว (Velocity)โดยจะใช้เฉพาะคลื่นลูกแรกเท่านั้น แล้วทำการเริ่มนับเวลาจาก Threshold ที่ได้กำหนดไว้ จนกระทั่งที่เวลาThreshold/2 จะเป็นเวลา t โดยเวลา t จะใช้เป็นค่า Velocity เพื่อที่จะให้ทราบถึง Dynamic ในการเคาะเปียโซ เมื่อเราเคาะเปียโซเบา จะพบว่า t จะมีค่าน้อย และเคาะเปียโซแรง พบว่า t มีค่ามาก

ในส่วนของภาคส่งที่เป็นสวิตซ์จะคล้ายคลึงกันกับเปียโซ เพียงแต่สัญญาณจะเป็นดิจิตอลเข้ามายังขาดิจิตอลของArduino Board ไมโครคอนโทรลเลอร์จะทำการวัดค่าสัญญาณ เมื่อมีการกด(Push) หรือปล่อย (Release) ของสวิตซ์

จากสัญญาณที่วัดได้ จะถูกเปลี่ยนให้เป็นสัญญาณดิจิตอลโดยส่งเป็น MIDI protocol ไปยังซินธิไซเซอร์เพื่อเล่นเป็นโน๊ตดนตรี และส่งรหัส MIDI นี้ไปเขียนลงบนเอสดีการ์ด

119235

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1) กลองและส่วนประกอบของกลอง

รูปที่ 3.2.1.1 ส่วนประกอบของกลอง

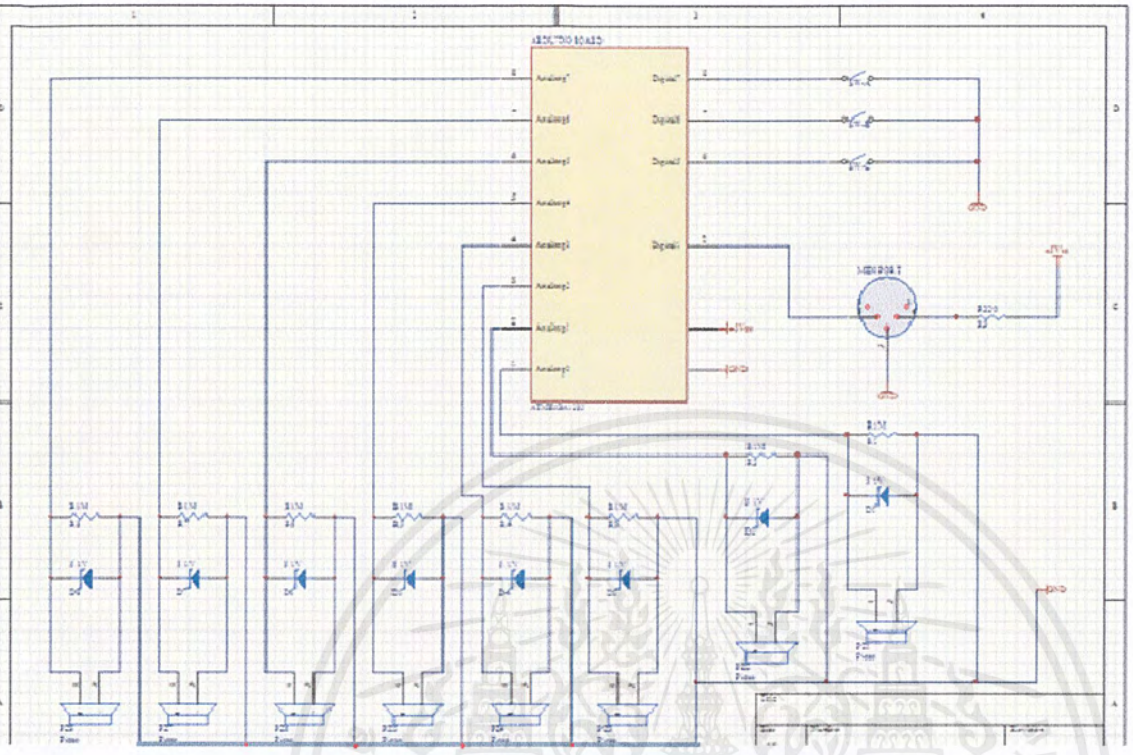
รูปที่ 3.2.1.2 กลองไฟฟ้า

กลองชุด ประกอบด้วย (1) Bass drum (2) Floor tom (3) Snare (4) Toms (5) Hi-hat (6) Crash cymbal และ Ride cymbal

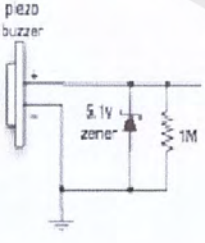
1. กลองใหญ่ (Bass Drum) กลองใหญ่จะติดอยู่กับพื้น กลองใหญ่ไม่ใช้ไม้ถือสำหรับตี แต่จะใช้กระเดื่อง (Pedal) สำหรับเท้าข้างขวาจะใช้เหยียบลงไปบนกระเดื่อง สำหรับ (2) Floor tom (3) Snare (4) Toms (6) Crash cymbal และ Ride cymbal จะต้องใช้ไม้ตี สำหรับ(5) Hi-hat จะประกอบด้วยฉาบสองใบ มีขาตั้งรองรับ ใบที่หนึ่งใส่ลงบนขาตั้งโดยให้ด้านนูนอยู่ด้านล่าง อีกใบหนึ่งใส่ลงบนขอตั้งโดยให้ด้านนูนอยู่ด้านบน มีที่ไขติดอยู่กับแกนของขาตั้ง โดยกะระยะให้ห่างกันพอประมาณ เพื่อไม่ให้ฉาบทั้งสองใบชิดติดกัน ช่วงล่างสุดมีกระเดื่องเหมือนกับกลองใหญ่สำหรับเหยียบให้ฉาบทั้งคู่กระทบกัน ดังนั้นสถานะของ Hi-hat เมื่อไม่ได้มีการเหยียบที่ pedal จะมีระยะห่างระหว่าง Hi-hat เรียกว่า Hi-hat open เมื่อมีการเหยียบที่ pedal Hi-hat จะชิดกันเรียกว่า Hi-hat close ดังนั้นในการออกแบบในส่วนของกลองใหญ่ (Bass Drum) เราจะใช้เป็น Switch แทน Pedal ในส่วนของ(2) Floor tom (3) Snare (4) Toms (6) Crash cymbal และ Ride cymbal จะใช้เป็น Sensor เชิงกลเพื่อรับแรงสั่นสะเทือนจากการเคาะหรือการตีกลอง ในส่วนของ Hi-hat จะต้องมีความพิเศษคือ จะต้องใช้ทั้งSwitchแทนPedal เพื่อแสดงสถานะ Hi-hat open และ Hi-hat close และใช้ Sensor เชิงกลเพื่อรับแรงสั่นสะเทือนจากการเคาะหรือการตีกลองด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2.2 วงจรกลองมิดิ



3.2.2.1) วงจรภาคส่ง



รูปที่ 3.2.2.2 Pad ที่ใช้สำหรับตีกลองฝังด้วยแผ่นเปียโซ

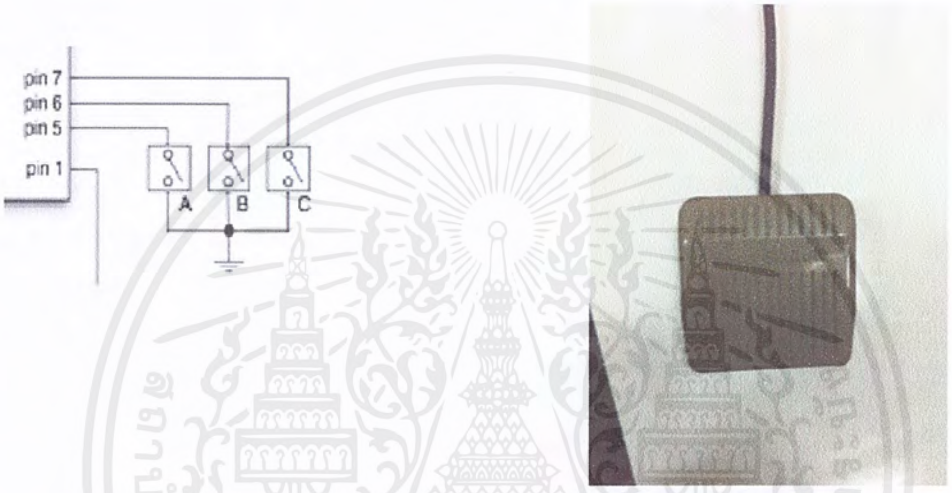
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่เป็นเปียโซอิเล็กทริกเซนเซอร์

ประกอบด้วยแผ่นเปียโซอิเล็กตริก ซีเนอร์ไดโอด 5.1V และตัวต้านทาน 1M ohm เมื่อแผ่นเปียโซที่ใส่ไว้ใน Pad ที่ไว้สำหรับฝึกตีกลอง (ดังรูป) ถูกเคาะ จะทำให้เกิดค่าความต่างศักย์ตกคร่อมที่ตัวต้านทาน ซีเนอร์ไดโอดจะทำหน้าที่ควบคุมแรงดันไฟฟ้าให้คงที่ เพื่อควบคุมแรงดันที่ออกจากเปียโซ ไม่ให้มีค่าเกิน 5.1V เพื่อไม่ให้สร้างความเสียหายแก่ Arduino Boardได้

ค่าสัญญาณที่เกิดขึ้นจะเป็นสัญญาณอนาลอก เข้าสู่ Analog Pin ของ Arduino Board ต่อไป

ส่วนที่เป็นสวิตซ์



รูปที่ 3.2.2.3 Foot Switch

ในส่วนที่ใช้สวิตซ์เป็นอินพุท จะให้สัญญาณเป็นโลจิก High และ Low โดยภายในArduino Board จะต้องมีการต่อ internal pull up ที่ Pin ที่ใช้งานด้วย ค่าสัญญาณที่เกิดขึ้นจะเป็นดิจิตอล เข้าสู่ Digital Pin ของ Arduino Board ต่อไป โดยในที่นี้จะเป็นสวิตซ์ประเภท Foot Switch หรือสวิตซ์ที่ใช้เท้าเหยียบ (ดังรูป)
ซึ่งจะให้เสียงเป็น Bass Drum ของกลองชุด

3.2.2.2) ภาครับ

หลังจากมีการรับค่าสัญญาณเข้ามาจะมีการประมวลผล โดยไมโครคอนโทรลเลอร์จะประมวลผลสัญญาณที่ได้ทั้งแบบอนาลอกและดิจิตอลให้อยู่ในสัญญาณดิจิตอลที่เป็นรูปแบบของสัญญาณมิดิ โดยมีขั้นตอนการทำงานของโปรแกรม คือ เริ่มต้นจากการกำหนดค่าพารามิเตอร์ต่างๆ ได้แก่ Channel ของ Drum เพื่อส่งรหัสแชนแนลของกลอง , Note เพื่อส่งรหัสของเสียงกลองแต่ละชิ้น ,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin สำหรับกำหนดขาที่จะใช้เป็น Input ของ Arduino Board และ LED Pin สำหรับตรวจสอบค่าสถานะของ MIDI OUT จากนั้นทำการกำหนดค่า Threshold ของสัญญาณที่ได้จากแผ่นเปียโซอิเล็กทริก เพื่อเป็นการรีเซตค่าจึงต้องกำหนด swichState และ currentSwitchState เป็น Low เนื่องจากปกติแล้วโหมดการทำงานของ DigitalPin จะเป็นเอาท์พุท จึงต้องมีการกำหนด pinMODE ให้เป็นอินพุท และต้องมีการ pull up ให้ switchPin เป็น HIGH ตั้งค่า baud rate ให้เป็นของ MIDI คือ 31250

ในส่วนของ Hi-hat เราจะใช้ SwitchA และPiezoA เริ่มต้นทำการอ่านค่าสถานะของ SwitchA มาเก็บไว้ที่ currentSwitchState แล้วเช็คค่า currentSwitchState ว่ามีสถานะเป็น HIGH หรือ LOW เมื่อยังไม่มีการเหยียบ pedal จะได้ค่าเป็น currentSwitchState เป็น LOW ให้อ่านค่าจาก PiezoA ถ้าค่าที่ได้มีค่ามากกว่าเท่ากับค่าเริ่มต้น(Threshold)ที่ได้ตั้งไว้ให้เริ่มนับเวลา (ให้ t=0) นับ t เพิ่มไปจนกระทั่งอ่านค่าได้มากกว่าเท่ากับ Threshold /2 จึงหยุด เวลาที่ได้นี้จะให้ผลเป็น Velocity โดย MCU จะทำการแสดงผลเป็น noteOn ด้วยโน๊ต hihatopen (มีเสียงโน๊ต hihatopenดังขึ้น) มี delayไปเป็นเวลา t แล้วจึงตามด้วย noteoff คือไม่มีเสียง

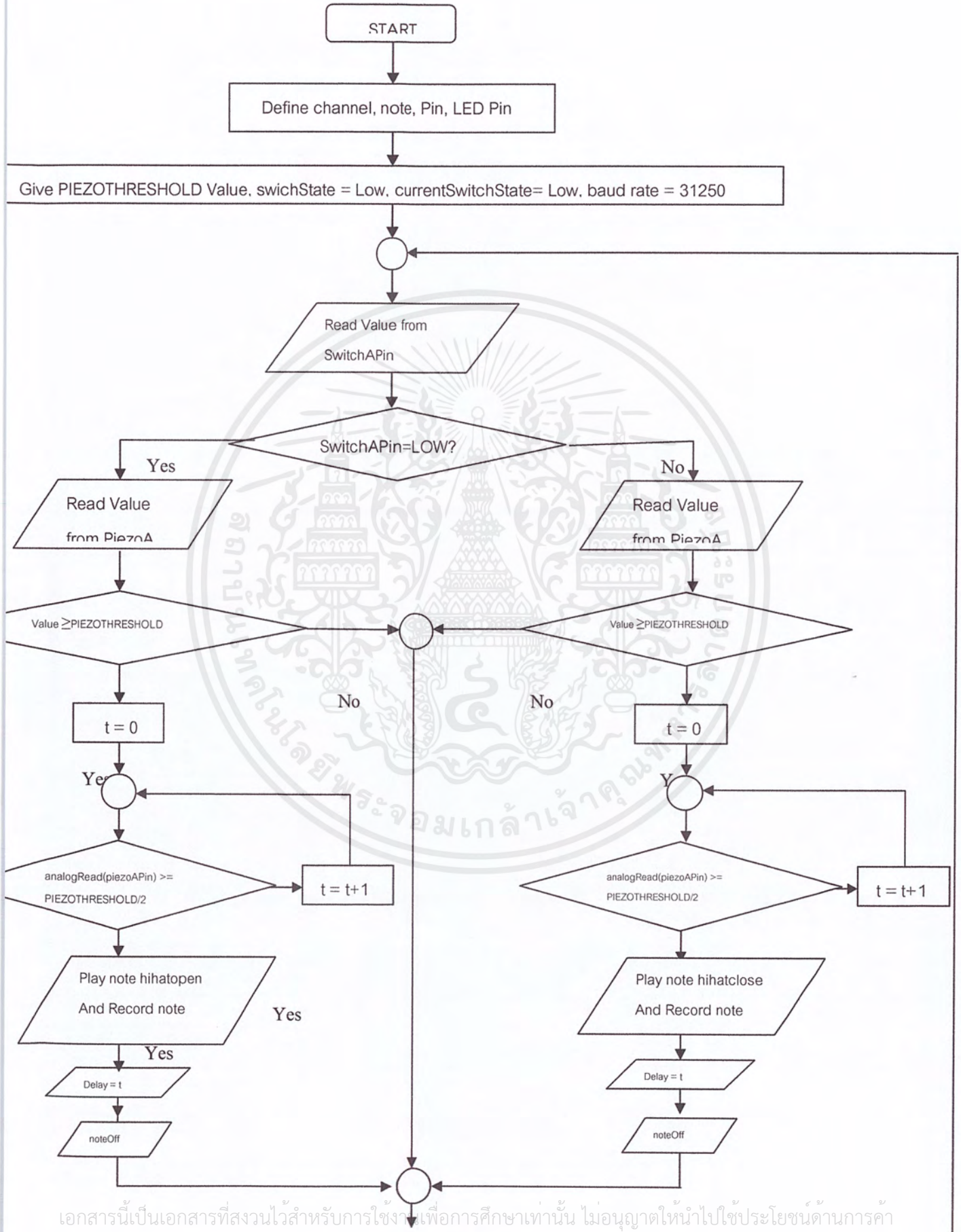
เมื่อมีการเหยียบ pedal จะได้ค่าเป็น currentSwitchState เป็น HIGH ให้อ่านค่าจาก PiezoA ถ้าค่าที่ได้มีค่ามากกว่าเท่ากับค่าเริ่มต้น(Threshold)ที่ได้ตั้งไว้ให้เริ่มนับเวลา (ให้ t=0) นับ t เพิ่มไปจนกระทั่งอ่านค่าได้มากกว่าเท่ากับ Threshold /2 จึงหยุด เวลาที่ได้นี้จะให้ผลเป็น Velocity โดย MCU จะทำการแสดงผลเป็น noteOn ด้วยโน๊ต hihatclose (มีเสียงโน๊ต hihatcloseดังขึ้น) มี delayไปเป็นเวลา t แล้วจึงตามด้วย noteoff คือเสียงเงียบ

ในส่วนของเปียโซอิเล็กทริกเซนเซอร์ (ส่วนของ(2) Floor tom (3) Snare (4) Toms (6) Crash cymbal และ Ride cymbal) จะทำการอ่านค่าสัญญาณที่เกิดขึ้นถ้าค่าที่ได้มีค่ามากกว่าเท่ากับค่าเริ่มต้น(Threshold)ที่ได้ตั้งไว้ให้เริ่มนับเวลา (ให้ t=0) นับ t เพิ่มไปจนกระทั่งอ่านค่าได้มากกว่าเท่ากับ Threshold /2 จึงหยุด เวลาที่ได้นี้จะให้ผลเป็น Velocity โดย MCU จะทำการแสดงผลเป็น noteOn ด้วยโน๊ตกลอง(มีเสียงโน๊ตกลอง ดังขึ้น แล้วแต่การตั้งเสียงเปียโซนั้นๆว่าเมื่อเปียโซตัวนั้นถูกเคาะจะให้มีเสียงอะไรอาจเป็น (2) Floor tom (3) Snare (4) Toms (6) Crash cymbal และ Ride cymbal เป็นต้น) มี delayไปเป็นเวลา t แล้วจึงตามด้วย noteoff คือเสียงเงียบ

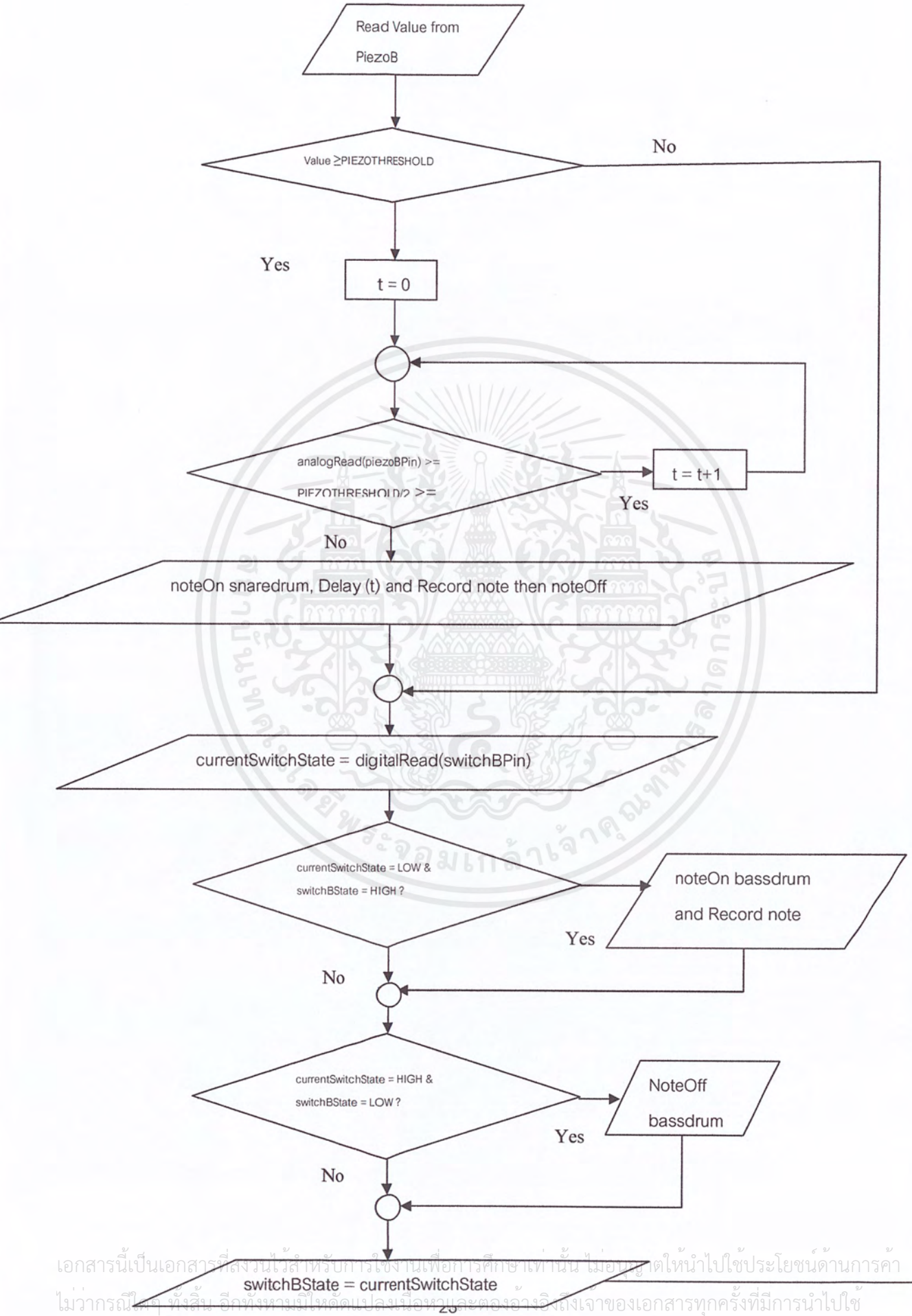
ส่วนที่เป็น Bass drum จะเป็นSwitch เริ่มต้นจากการอ่านค่าของ SwitchB มาเก็บไว้ที่ currentSwitchState แล้วเช็คค่า SwitchA ที่เกิดขึ้น เมื่อมีการเหยียบ pedal ค่า currentSwitchState จะเป็น LOW และ SwitchAจะมีค่าเป็น HIGH จะทำคำสั่ง noteOn bassdrum และ delay เป็นเวลา t เมื่อไม่มีการเหยียบ pedal ค่า currentSwitchState จะเป็น HIGH และ SwitchAจะมีค่าเป็น LOW จะทำคำสั่ง noteOff bassdrum หลังจากนั้นจะทำการวนซ้ำกลับไปที่ส่วนของ hihat ส่วนของเปียโซ และส่วนของสวิทซ์ วน loop ไปเรื่อยๆ โดยสามารถเขียนเป็นลำดับการทำงานได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3) Flow Chart การทำงานของโปรแกรมที่ได้ออกแบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

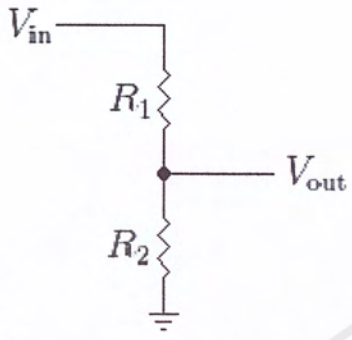


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4) การบันทึกข้อมูลไปยัง SD Card

เนื่องจาก SD Card ต้องการไฟเลี้ยงเพียงแค่ 3.3v เท่านั้น จึงต้องใช้ตัวต้านทานในการแบ่งแรงดันระดับ 5v ให้เป็น 3.3v เสียก่อน

Vin = 5 V , R1 = 1.8k , R2 = 3.3k



Vout = R2 / R1 + R2

= 3.3k / 1.8k + 3.3k

= 16.5k / 5.1k

Vout = 3.235 V

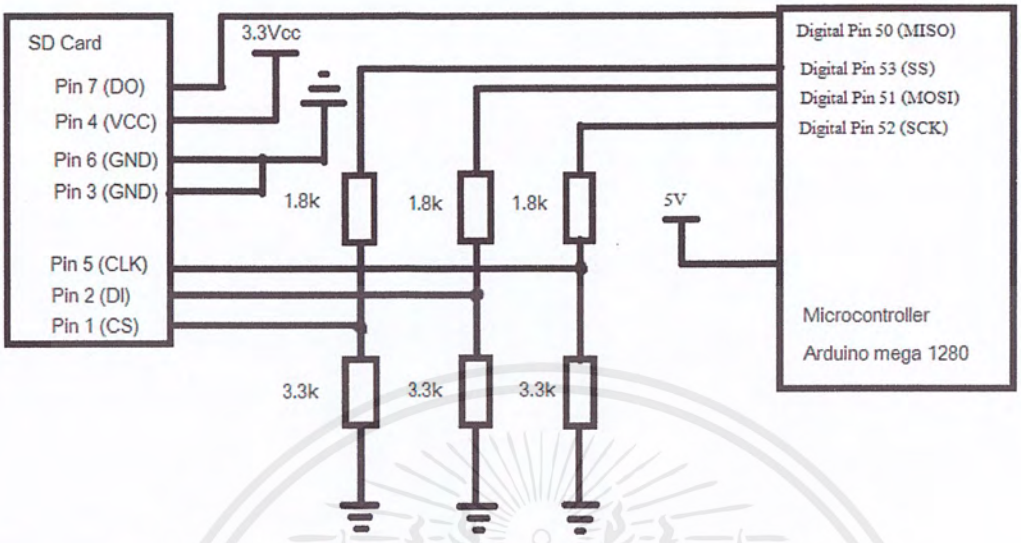
รูปที่ 3.2.4 Voltage Divider

สำหรับในส่วนของการเชื่อมต่อระหว่างบอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 1280 กับ SD Card จะมีการเชื่อมต่อดังนี้

Arduino	SD Card
+3.3v	Pin 4 (VCC)
Gnd	Pins 3 & 6 (GND)
Digital Pin 52 (SCK)	Pin 5 (CLK)
Digital Pin 50 (MISO)	Pin 7 (DO)
Digital Pin 51 (MOSI)	Pin 2 (DI)
Digital Pin 53 (SS)	Pin 1 (CS)

ตารางที่ 2 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์และเอสดีการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2.5 การเชื่อมต่อระหว่าง SD Card กับ Arduino

รูปที่ 3.2.6 ภาพกลอง MIDI แบบเสร็จสมบูรณ์

รูปที่ 3.2.7 ภาพเครื่องซินธิไซเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

ในบทที่ 4 เป็นการทดลอง และแสดงผลการในรูปแบบต่างๆ โดยเป็นการทดลองเกี่ยวกับเซนเซอร์เปียโซที่ถูกกระทำในสถานการณ์แตกต่างกัน สามารถแสดงได้ดังนี้

ส่วนของเซนเซอร์เปียโซ

1.1 เปรียบเทียบขนาดสัญญาณที่ได้รับจากวัสดุที่ใช้เคาะชนิดต่างๆ

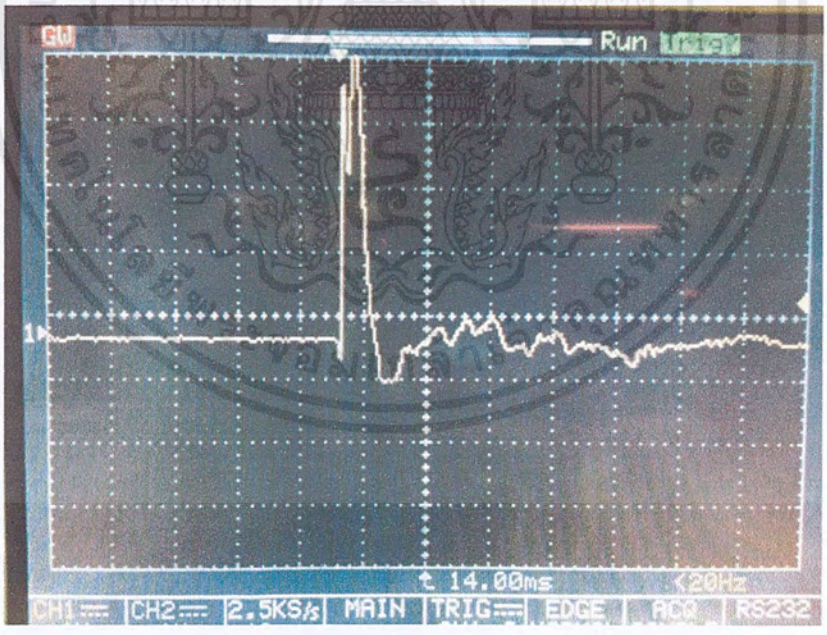
1.2 เปรียบเทียบสัญญาณที่ได้รับการเคาะเมื่อมีพลาสติกหุ้ม

1.3 เปรียบเทียบขนาดสัญญาณที่ได้รับจากระยะทางที่ใช้เคาะ

1.4 เปรียบเทียบขนาดสัญญาณที่ได้รับการเคาะบนพื้นผิวที่แตกต่างกัน

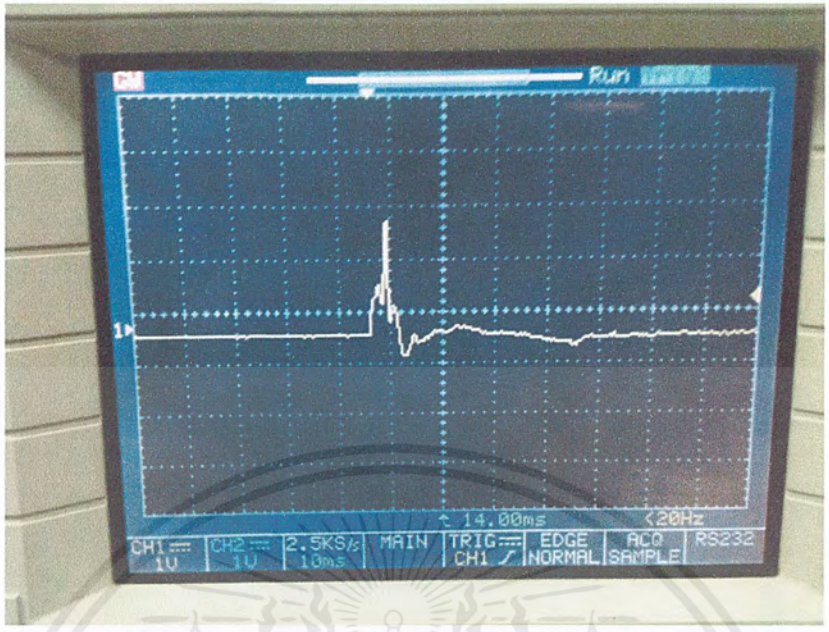
4.1) ส่วนของเซนเซอร์เปียโซ

4.1.1.1) เปรียบเทียบขนาดสัญญาณที่ได้รับจากวัสดุที่ใช้เคาะชนิดต่างๆ

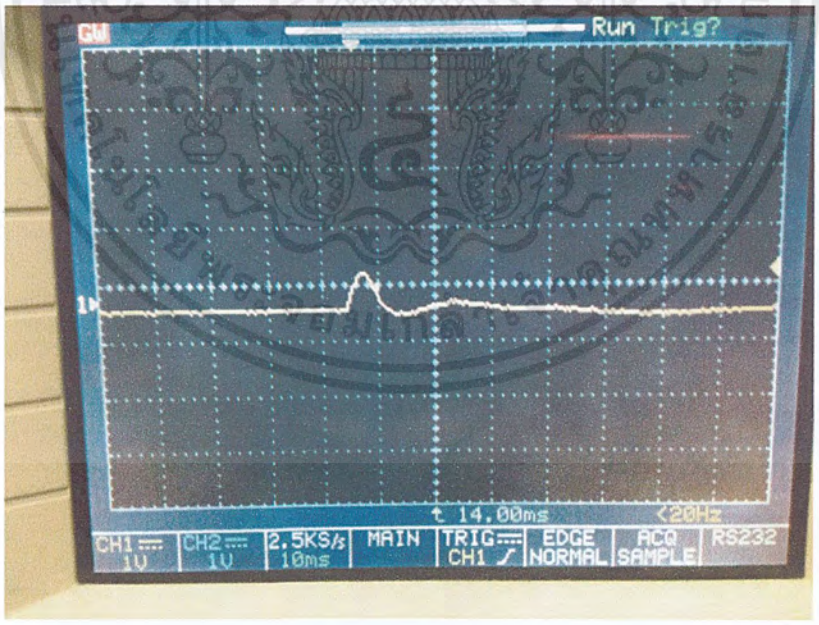


รูปที่ 4.1 สัญญาณที่ใช้นิ้วในการเคาะด้วยความแรงมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

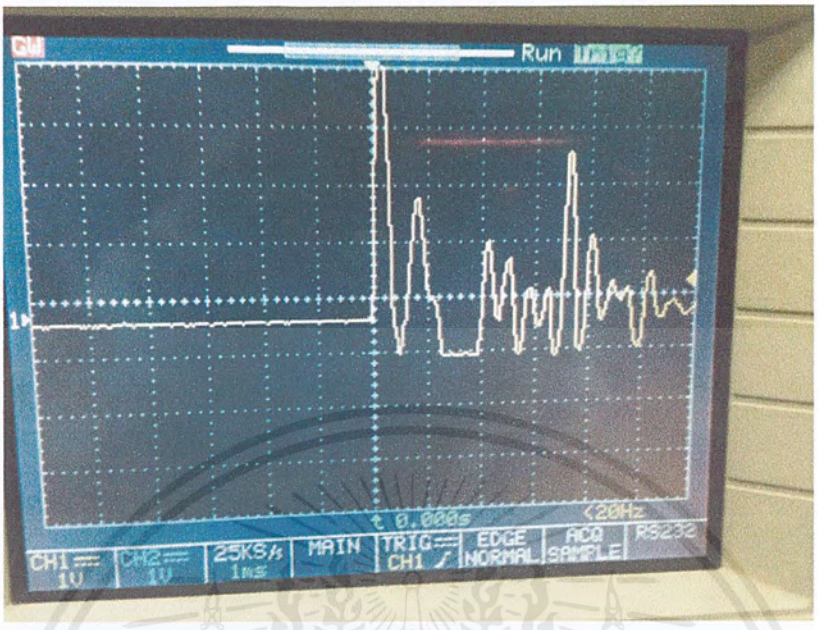


รูปที่ 4.2 สัญญาณที่ใช้นิ้วเคาะด้วยความแรงปานกลาง



รูปที่ 4.3 สัญญาณที่ใช้นิ้วเคาะด้วยความแรงน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 สัญญาณที่ใช้เหล็กปลายแหลมเคาะด้วยความแรงมาก

\

รูปที่ 4.5 สัญญาณที่ใช้เหล็กปลายแหลมเคาะด้วยความแรงปานกลาง

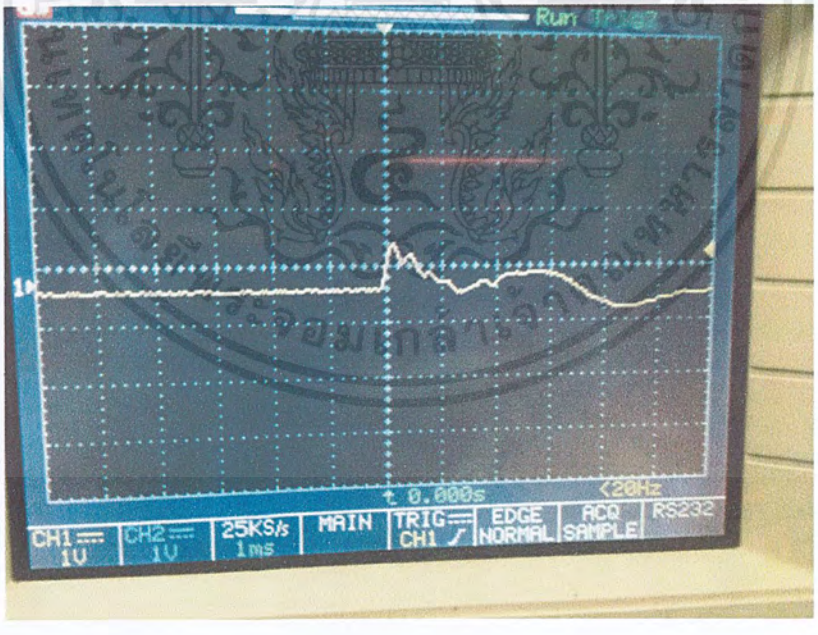
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.6 สัญญาณที่ใช้เหล็กปลายแหลมเคาะด้วยความแรงน้อย

รูปที่ 4.7 สัญญาณที่ใช้ไม้กลองเคาะด้วยความแรงมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

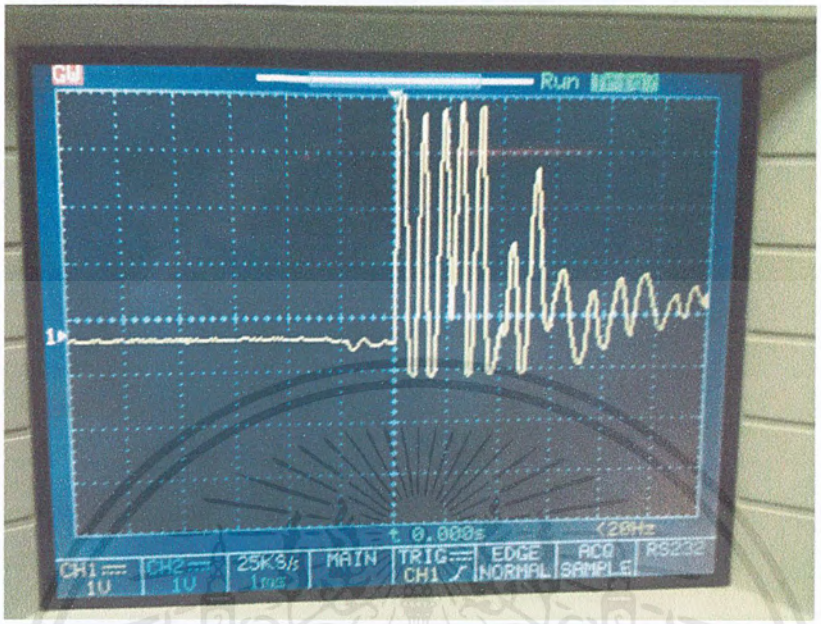
รูปที่ 4.8 สัญญาณที่ใช้ไม้กลองเคาะด้วยความแรงปานกลาง



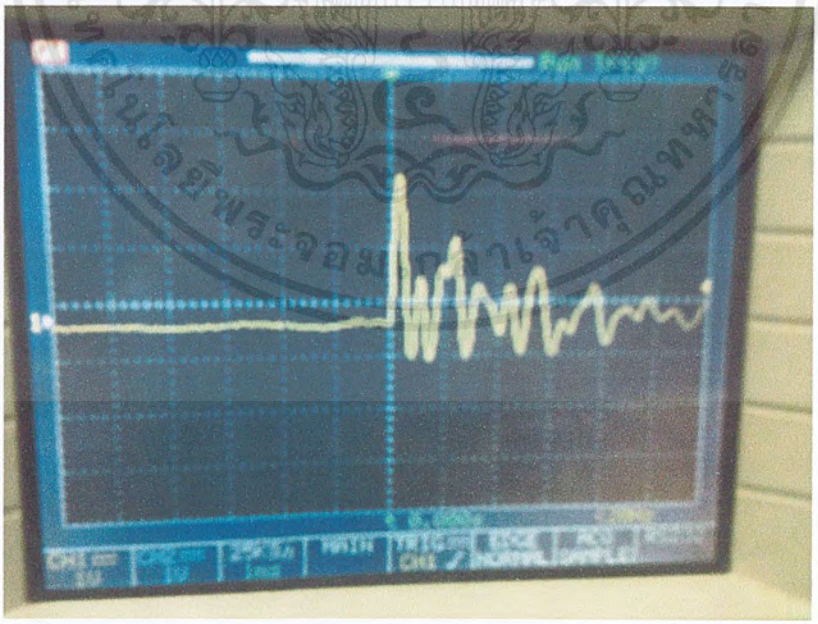
รูปที่ 4.9 สัญญาณที่ใช้ไม้กลองเคาะด้วยความแรงน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.2) เปรียบเทียบสัญญาณที่ได้รับการเคาะเมื่อมีพลาสติกหุ้ม



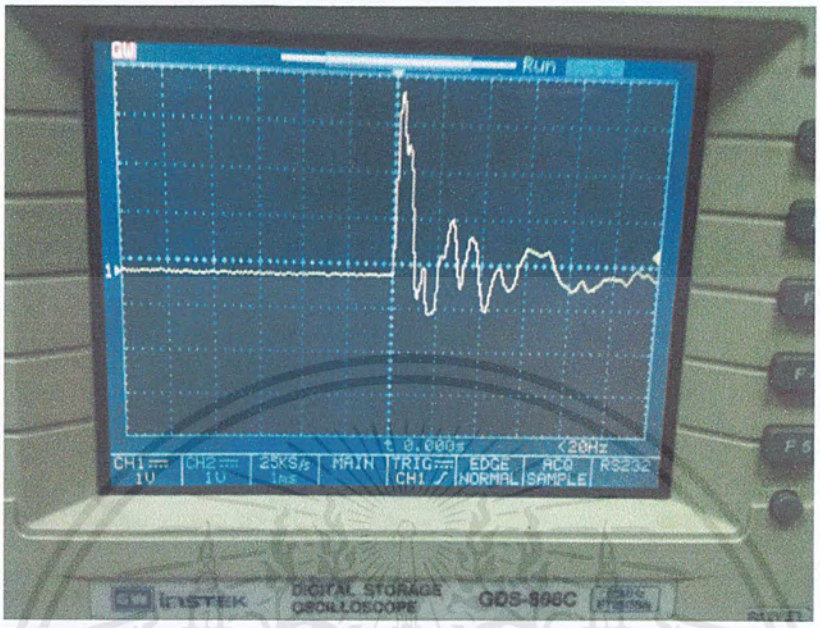
รูปที่ 4.10 สัญญาณที่ใช้ไม้กลองเคาะด้วยความแรงมากและหุ้มด้วยพลาสติก



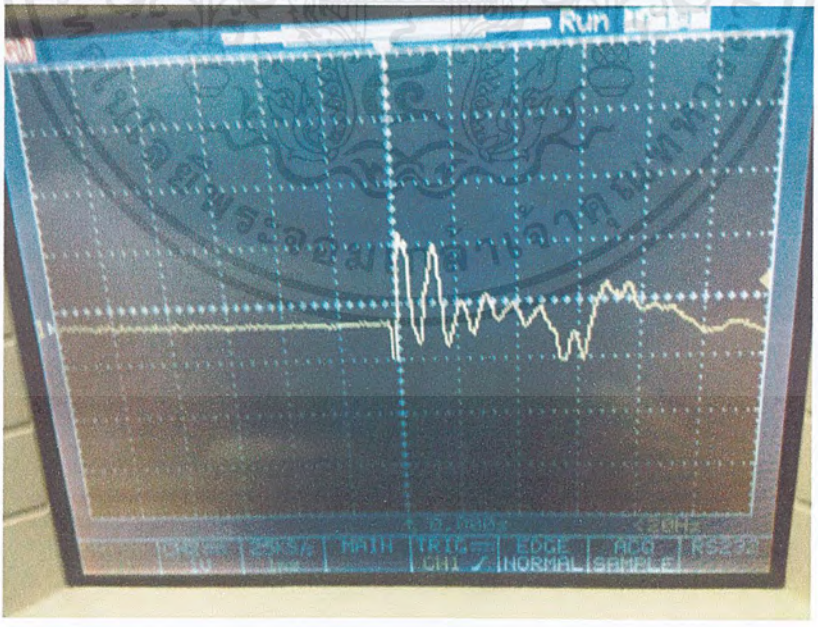
รูปที่ 4.11 สัญญาณที่ใช้ไม้กลองเคาะด้วยความแรงน้อยและหุ้มด้วยพลาสติก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.3) เปรียบเทียบขนาดสัญญาณที่ได้รับจากระยะทางที่ใช้เคาะ



รูปที่ 4.12 สัญญาณที่เคาะห่างจากเปียโซเป็นระยะทาง 1 ซ.ม.



รูปที่ 4.13 สัญญาณที่เคาะห่างจากเปียโซเป็นระยะทาง 4 ซ.ม.

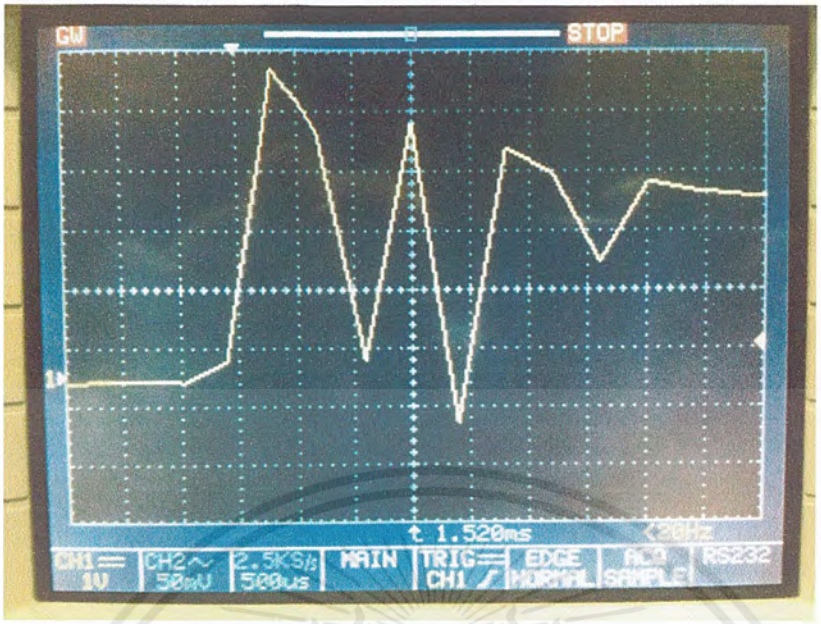
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.14 สัญญาณที่เคาะห่างจากเปียโซเป็นระยะทาง 7 ซ.ม.

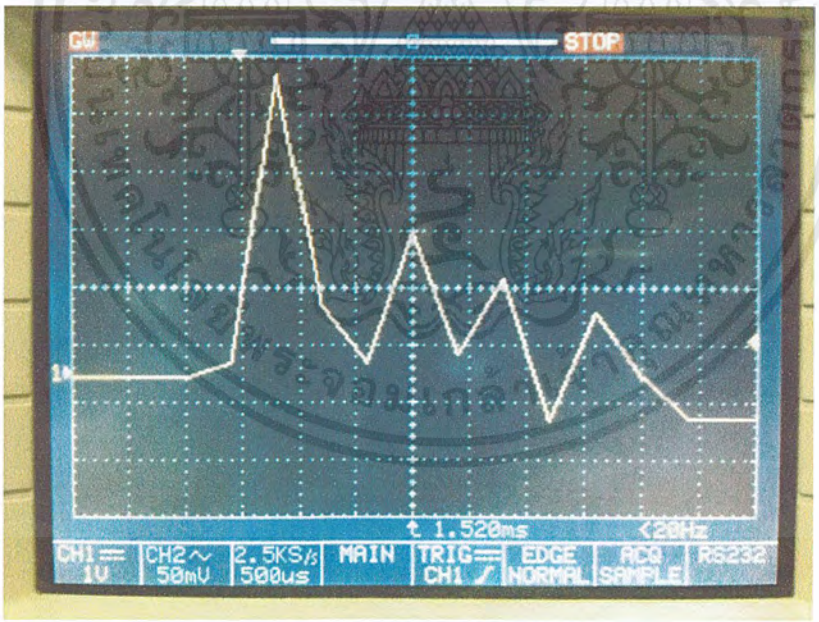
4.1.1.4) เปรียบเทียบขนาดสัญญาณที่ได้รับการเคาะบนพื้นผิวที่แตกต่างกัน

รูปที่ 4.15 สัญญาณที่ใช้นิ้วเคาะโดยเปียโซวางอยู่บนโต๊ะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

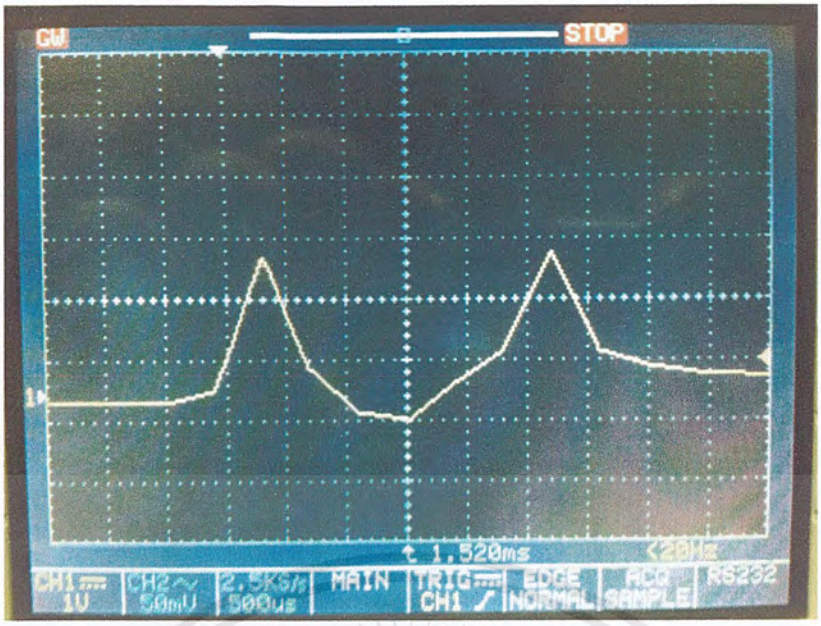


รูปที่ 4.16 สัญญาณที่ใช้ไม้กลองเคาะโดยเปียโซวางอยู่บนโต๊ะ

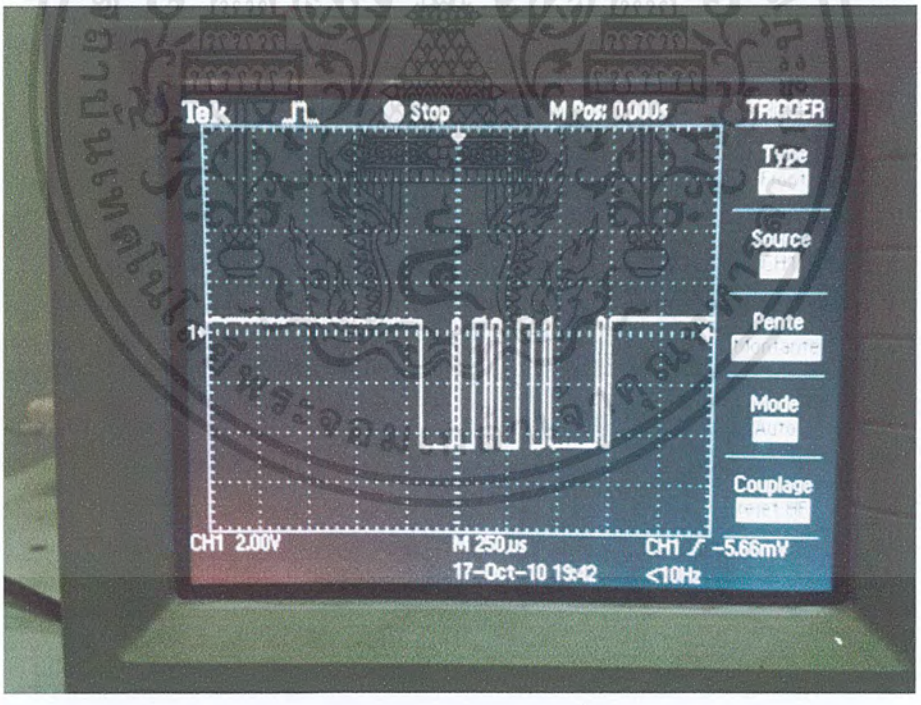


รูปที่ 4.17 สัญญาณที่ใช้ไม้กลองเคาะโดยเปียโซวางอยู่บนซิลิโคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 สัญญาณที่ใช้ไม้กลองเคาะโดยเปียโซวางอยู่บนแผ่น CD



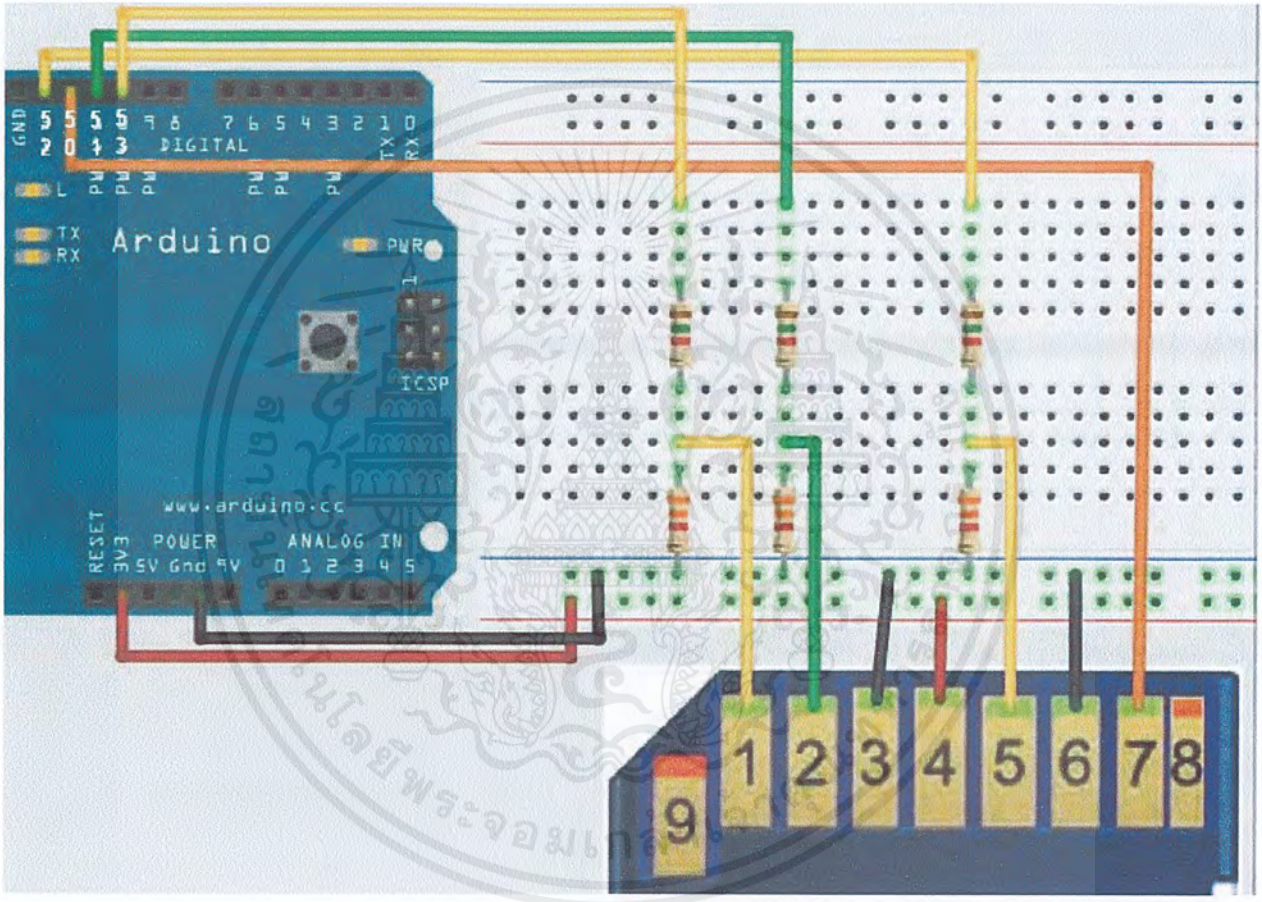
รูปที่ 4.19 กราฟสัญญาณ MIDI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2) การทดลองการอ่านและเขียนข้อมูลลงบน SD Card

SD Card เป็น Memoryที่มีขนาดเล็ก ราคาย่อมเยา และยังสามารถพกพาได้สะดวก อีกทั้ง Arduino สามารถสื่อสารผ่าน SPI Interface ได้โดยง่าย จึงเหมาะสมที่จะนำมาใช้ในการบันทึกข้อมูล

เนื่องจาก SD Card ต้องการไฟเลี้ยงเพียงแค่ 3.3v เท่านั้น จึงต้องใช้ตัวต้านทานในการแบ่งแรงดันระดับ 5v ให้เป็น 3.3v ต่อไปนี้เป็นรูปวงจรที่ใช้ในการอ่านและเขียนข้อมูลลงบน SD Card โดยใช้ไมโครคอนโทรลเลอร์ Arduino Mega 1280



รูปที่ 4.2 การต่อ SD Card กับ Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1) การทดลองอ่านและเขียนข้อมูลลงบน SD Card

โปรแกรมต่อไปนี้เป็นโปรแกรมที่ใช้ในการอ่านและเขียนข้อมูลลงบน SD Card โดยต้องทำการติดตั้ง SdFat.h and SdFatUtil.h libraries ลงใน libraries folder ของ Arduino และต้องทำการ format SD Card ให้เป็น FAT เสียก่อน

```
#include <SdFat.h>
#include <SdFatUtil.h>
Sd2Card card;
SdVolume volume;
SdFile root;
SdFile file;
// store error strings in flash to save RAM
#define error(s) error_P(PSTR(s))
void error_P(const char* str) {
PgmPrint("error: ");
SerialPrintln_P(str);
if (card.errorCode()) {
PgmPrint("SD error: ");
Serial.print(card.errorCode(), HEX);
Serial.print(',');
Serial.println(card.errorData(), HEX);
}
while(1);
}
// Write a Carriage Return and Line Feed to the file
void writeCRLF(SdFile& f) {
f.write((uint8_t*)"\r\n", 2);
}
// Write an unsigned number to file
void writeNumber(SdFile& f, uint32_t n) {
uint8_t buf[10];
uint8_t i = 0;
do {
i++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
buf[sizeof(buf) - i] = n%10 + '0';
n /= 10;
} while (n);
f.write(&buf[sizeof(buf) - i], i);
}
// Write a string to file
void writeString(SdFile& f, char *str) {
uint8_t n;
for (n = 0; str[n]; n++);
f.write((uint8_t *)str, n);
}
void setup() {
Serial.begin(9600);
Serial.println();
Serial.println("Type any character to start");
while (!Serial.available());
// initialize the SD card at SPI_HALF_SPEED to avoid bus errors with breadboards.
// Use SPI_FULL_SPEED for better performance if your card an take it.
if (!card.init(SPI_HALF_SPEED)) error("card.init failed");
// initialize a FAT volume
if (!volume.init(&card)) error("volume.init failed");
// open the root directory
if (!root.openRoot(&volume)) error("openRoot failed");
// create a new file
char name[] = "TESTFILE.TXT";
file.open(&root, name, O_CREAT | O_EXCL | O_WRITE);
// Put todays date and time here
file.timestamp(2, 2010, 12, 25, 12, 34, 56);
// write 10 lines to the file
for (uint8_t i = 0; i < 10; i++) {
writeString(file, "Line: ");
writeNumber(file, i);
writeString(file, " Write test.");
writeCRLF(file);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
// close file and force write of all data to the SD card
file.close();
Serial.println("File Created");
// open a file
if (file.open(&root, name, O_READ)) {
Serial.println(name);
}
else{
error("file.open failed");
}
Serial.println();
int16_t character;
while ((character = file.read()) > 0) Serial.print((char)character);
Serial.println("\nDone");
}
void loop() { }
```

เมื่อ Run โปรแกรม และเปิด Serial monitor ขึ้นมา พิมพ์ตัวอักษรใดๆแล้วกด Send จากนั้น โปรแกรมจะเขียนข้อมูลไปยัง SD card แล้วทำการอ่านชื่อไฟล์และข้อมูลออกมาผ่าน serial monitor โดยได้ผลออกมาดังนี้

ผลการทดลอง 4.2.1

```
Type any character to start
File Created
TESTFILE.TXT
Line: 0 Write test.
Line: 1 Write test.
Line: 2 Write test.
Line: 3 Write test.
Line: 4 Write test.
Line: 5 Write test.
Line: 6 Write test.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Line: 7 Write test.

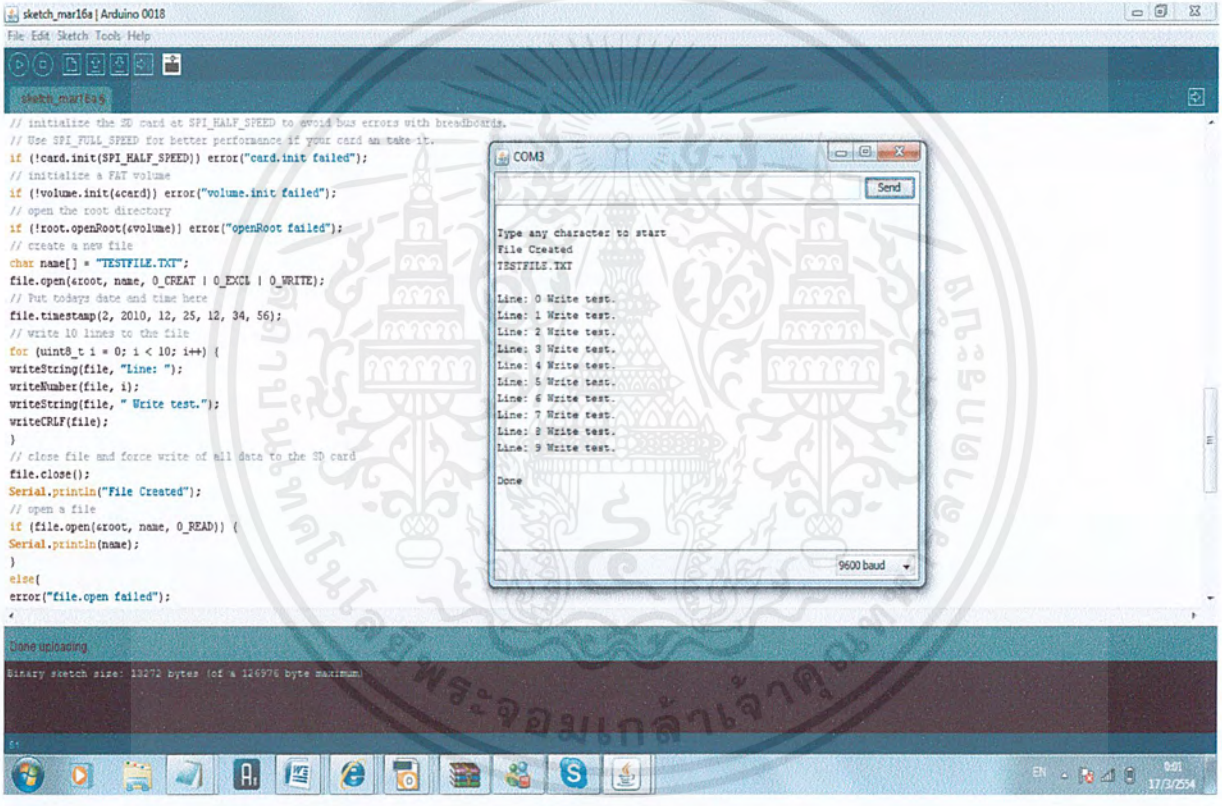
Line: 8 Write test.

Line: 9 Write test.

Done

จากนั้นเมื่อโปรแกรมทำงานเสร็จสมบูรณ์ นำ Card ออกจาก SD card connector แล้วนำมาเปิดไฟล์ จะพบชื่อไฟล์ TESTFILE.TXT เมื่อทำการเปิด TESTFILE.TXT ได้ผลการทดลองเป็นดังนี้

ผลการทดลองตอนที่ 1



รูปที่ 4.2.1 ผลการทดลองตอนที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2) การทดลองการส่ง MIDI ไปบันทึกลงบน SD Card โดยตรวจจับสัญญาณจาก Piezo เพียงตัวเดียว

โปรแกรมด้านล่างนี้เป็นโปรแกรมการบันทึกลงบน SD Card เป็นรหัส MIDI โดยหากมีสัญญาณอนาลอกเข้ามาจะทำการจะทำการบันทึกข้อมูล โดยใช้เปียโซเซนเซอร์หนึ่งตัว

```
#include <SdFat.h>
#include <SdFatUtil.h>
Sd2Card card;
SdVolume volume;
SdFile root;
SdFile file;
#define drumchan            16
#define note_hihatopen      44
#define switchAPin          7
#define piezoAPin           0
#define switchDPin          3
#define ledPin              13
#define PIEZOTHRESHOLD 100
int switchAState = LOW;
int j=0;
int count=0;
int RecordSW = LOW;
int currentSwitchState = LOW;
int val,t;
void setup() {
 pinMode(switchAPin, INPUT);
 pinMode(switchDPin, INPUT);
 digitalWrite(switchAPin, HIGH);  // turn on internal pullup
 pinMode(ledPin, OUTPUT);
 Serial.begin(31250);   // set MIDI baud rate
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
card.init(SPI_HALF_SPEED);
 // initialize a FAT volume
 volume.init(&card);
 // open the root directory
 root.openRoot(&volume);
 // create a new file
}
// Write a Carriage Return and Line Feed to the file
void writeCRLF(SdFile& f) {
f.write((uint8_t*)"\r\n", 2);
}
// Write an unsigned number to file
void writeNumber(SdFile& f, uint32_t n) {
uint8_t buf[10];
uint8_t i = 0;
do {
i++;
buf[sizeof(buf) - i] = n%10 + '0';
n /= 10;
} while (n);
f.write(&buf[sizeof(buf) - i], i);
}
// Write a string to file
void writeString(SdFile& f, char *str) {
uint8_t n;
for (n = 0; str[n]; n++);
f.write((uint8_t *)str, n);
}
 void loop() {
// deal with first piezo, this is kind of a hack
 val = analogRead(piezoAPin);
 if( val >= PIEZOTHRESHOLD ) {
   t=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

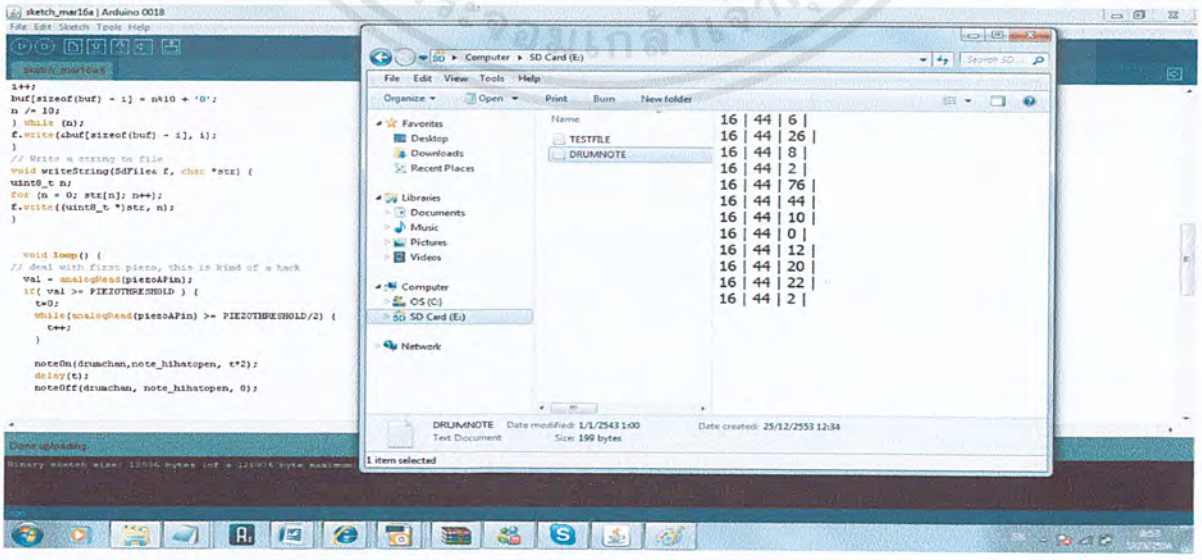
```
 while(analogRead(piezoAPin) >= PIEZOTHRESHOLD/2) {
     t++;
    }
    noteOn(drumchan,note_hihatopen, t*2);
    delay(t);
    noteOff(drumchan, note_hihatopen, 0);
    }
}
// Send a MIDI note-on message.  Like pressing a piano key
// channel ranges from 0-15
void noteOn(byte channel, byte note, byte velocity) {
  midiMsg( (0x90 | channel), note, velocity);
  char name[] = "DRUMNOTE.TXT";
  file.open(&root, name, O_CREAT | O_APPEND | O_WRITE);
  // Put todays date and time here
  file.timestamp(2, 2010, 12, 25, 12, 34, 56);
  // write MIDI CODE to the file
  writeNumber(file, channel);
  writeString(file, " | ");
  writeNumber(file, note);
  writeString(file, " | ");
  writeNumber(file, velocity);
  writeString(file, " | ");
  writeCRLF(file);
  // close file and force write of all data to the SD card
  file.close();
}
// Send a MIDI note-off message.  Like releasing a piano key
void noteOff(byte channel, byte note, byte velocity) {
  midiMsg( (0x80 | channel), note, velocity);
}
// Send a general MIDI message
void midiMsg(byte cmd, byte data1, byte data2) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
digitalWrite(ledPin,HIGH);  // indicate we're sending MIDI data
  Serial.print(cmd, BYTE);
  Serial.print(data1, BYTE);
  Serial.print(data2, BYTE);
  digitalWrite(ledPin,LOW);
}
```

จากโปรแกรมเบื้องต้น การทำงานจะเริ่มต้นทำการอ่านค่าจาก piezoAPin เมื่อค่าของ piezoAPinมีค่ามากกว่าค่าเริ่มต้น จะเริ่มนับเวลา (t=0) ทำการอ่านค่าไปเรื่อยๆ ในขณะที่ค่ายังมากกว่าครึ่งหนึ่งของค่าเริ่มต้นให้ทำการนับเวลาต่อไป(เพิ่มเวลา t++)จนกระทั่งมีค่าน้อยกว่าครึ่งหนึ่งของค่าเริ่มต้น จะไปทำคำสั่ง noteOn ซึ่งจะส่ง MIDI massage ออกไป ซึ่ง MIDI massage จะประกอบด้วย Channel ของชนิดเครื่องดนตรี , โน๊ตดนตรี , และความเร็วซึ่งให้ผลเป็นความแรงของการตี จากนั้นจะทำการ สร้างไฟล์ชื่อ DRUMNOTE.TXT ขึ้นมา ทำการเปิดไฟล์ โดยใช้คำสั่ง .open(&root, name, O_CREAT | O_APPEND | O_WRITE) โดยคำสั่ง O_APPEND เป็นการเขียนที่จะไม่เขียนทับข้อมูลเดิม หรือเป็นการเขียนต่อท้ายข้อมูลล่าสุดลงไป file.timestamp ใช้เพื่อบันทึกเวลาและวันที่ จากนั้น MIDI massage ซึ่งประกอบด้วย Channel, โน๊ตดนตรี , และความเร็วจะถูกส่งไปเก็บข้อมูลลงในไฟล์ชื่อ DRUMNOTE.TXT ที่ถูกสร้างขึ้น writeCRLF เป็นคำสั่งที่ใช้เพื่อขึ้นบรรทัดใหม่ จากนั้นจะเป็นการปิดไฟล์ และเนื่องจากเราใช้คำสั่ง O_APPEND เมื่อมีสัญญาณเข้ามาใหม่ การบันทึกข้อมูลจึงไม่ทับกับข้อมูลที่มีอยู่เดิม มี Delay เท่ากับ t จากนั้นจะทำคำสั่ง noteOff ซึ่งมี velocity = 0 นั่นคือไม่มีการตีกลองไม่มีสัญญาณเข้ามา จะทำให้เสียงเงียบ เมื่อนำ SD Card ที่ได้บันทึกข้อมูล ไปทำการเปิดไฟล์ในคอมพิวเตอร์ จะได้ข้อมูลดังนี้

ผลการทดลองตอนที่ 2



รูปที่ 4.2.2 ผลการทดลองตอนที่ 2

ลิขสิทธิ์เป็นของมหาวิทยาลัยเชียงใหม่ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3) การทดลองบันทึกค่าลงบน SD Card โดยทำการตรวจจับสัญญาณอนาลอกจากเปียโซเซนเซอร์หลายตัว

โปรแกรมการทดลองบันทึกค่าลงบน SD Card โดยทำการจับสัญญาณจากเปียโซเซนเซอร์หลายตัว

```
#include <SdFat.h>
#include <SdFatUtil.h>
Sd2Card card;
SdVolume volume;
SdFile root;
SdFile file;
#define drumchan              16
#define note_bassdrum         35
#define note_snaredrum        38
#define note_lowtom           45
#define note_hightom          50
#define note_ridecymbal       51
#define switchAPin            1
#define piezoAPin             0
#define piezoBPin             1
#define piezoCPin             2
#define piezoDPin             3
#define ledPin                13
#define PIEZOTHRESHOLD 100
int switchAState = LOW;
int count=0;
int currentSwitchState = LOW;
int val,t;
void setup() {
 pinMode(switchAPin, INPUT);
 digitalWrite(switchAPin, HIGH);  // turn on internal pullup
 pinMode(ledPin, OUTPUT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 Serial.begin(31250);   // set MIDI baud rate
   card.init(SPI_HALF_SPEED);
   // initialize a FAT volume
   volume.init(&card);
   // open the root directory
   root.openRoot(&volume);
   // create a new file
 }

 // Write a Carriage Return and Line Feed to the file
 void writeCRLF(SdFile& f) {
 f.write((uint8_t*)"\r\n", 2);
 }
 // Write an unsigned number to file
 void writeNumber(SdFile& f, uint32_t n) {
 uint8_t buf[10];
 uint8_t I = 0;
 do {
 i++;
 buf[sizeof(buf) – i] = n%10 + '0';
 n /= 10;
 } while (n);
 f.write(&buf[sizeof(buf) – i], i);
 }
 // Write a string to file
 void writeString(SdFile& f, char *str) {
 uint8_t n;
 for (n = 0; str[n]; n++);
 f.write((uint8_t *)str, n);
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void loop() {
// deal with first piezo, this is kind of a hack
  val = analogRead(piezoAPin);
  if( val >= PIEZOTHRESHOLD ) {
    t=0;
    while(analogRead(piezoAPin) >= PIEZOTHRESHOLD/2) {
      t++;
    }
    noteOn(drumchan,note_snaredrum, t*2);
    delay(t);
    noteOff(drumchan,note_snaredrum, 0);
    }
  // deal with second piezos, this is kind of a hack
  val = analogRead(piezoBPin);
  if( val >= PIEZOTHRESHOLD ) {
    t=0;
    while(analogRead(piezoBPin) >= PIEZOTHRESHOLD/2) {
      t++;
    }
    noteOn(drumchan,note_ridecymbal, t*2);
    delay(t);
    noteOff(drumchan,note_ridecymbal,0);
  }
// deal with third piezos, this is kind of a hack
  val = analogRead(piezoCPin);
  if( val >= PIEZOTHRESHOLD ) {
    t=0;
    while(analogRead(piezoCPin) >= PIEZOTHRESHOLD/2) {
      t++;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 noteOn(drumchan,note_lowtom, t*2);
    delay(t);
    noteOff(drumchan,note_lowtom,0);
  }

 // deal with fourth piezos, this is kind of a hack
  val = analogRead(piezoDPin);
  if( val >= PIEZOTHRESHOLD ) {
    t=0;
    while(analogRead(piezoDPin) >= PIEZOTHRESHOLD/2) {
      t++;
    }
    noteOn(drumchan,note_hightom, t*2);
    delay(t);
    noteOff(drumchan,note_hightom,0);
  }
  // deal with switchA
  currentSwitchState = digitalRead(switchAPin);
  if( currentSwitchState == LOW && switchAState == HIGH ) // push
    noteOn(drumchan,  note_bassdrum, 100);
  if( currentSwitchState == HIGH && switchAState == LOW ) // release
    noteOff(drumchan, note_bassdrum, 0);
  switchAState = currentSwitchState;
}
// Send a MIDI note-on message.  Like pressing a piano key
// channel ranges from 0-15
void noteOn(byte channel, byte note, byte velocity) {
  midiMsg( (0x90 | channel), note, velocity);
  char name[] = "DRUMNOTE.TXT";
  file.open(&root, name, O_CREAT | O_APPEND | O_WRITE);
  // Put todays date and time here
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

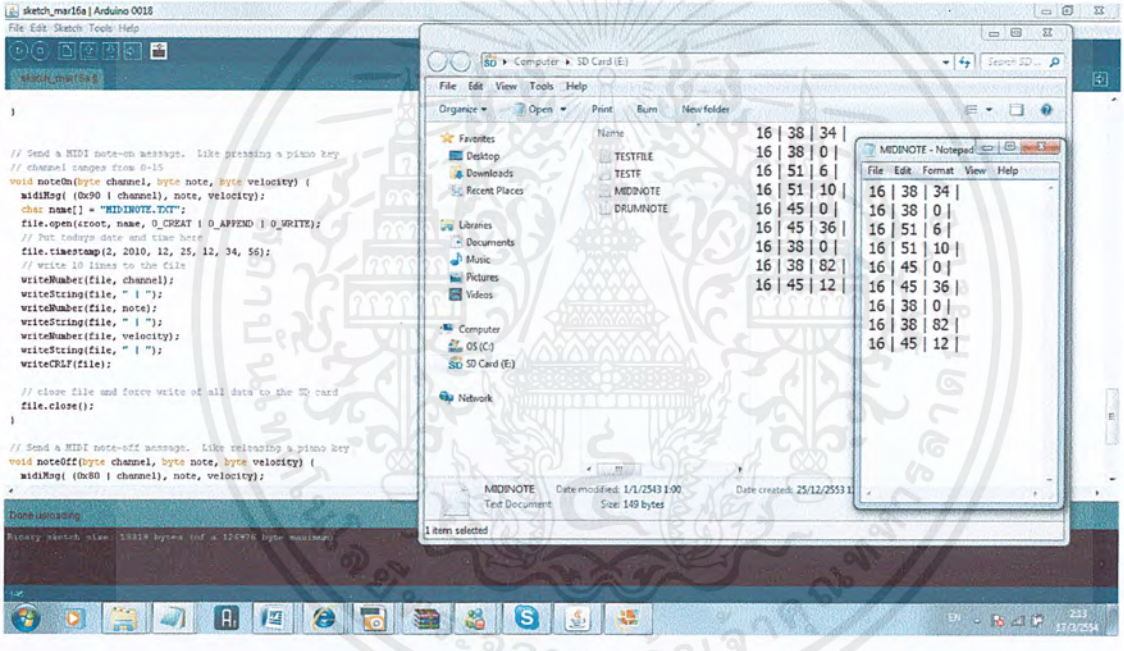
```
 file.timestamp(2, 2010, 12, 25, 12, 34, 56);
  // write MIDI Code to the file
  writeNumber(file, channel);
  writeString(file, " | ");
  writeNumber(file, note);
  writeString(file, " | ");
  writeNumber(file, velocity);
  writeString(file, " | ");
  writeCRLF(file);
  // close file and force write of all data to the SD card
  file.close();
}
// Send a MIDI note-off message.  Like releasing a piano key
void noteOff(byte channel, byte note, byte velocity) {
  midiMsg( (0x80 | channel), note, velocity);
}
// Send a general MIDI message
void midiMsg(byte cmd, byte data1, byte data2) {
  digitalWrite(ledPin,HIGH);  // indicate we're sending MIDI data
  Serial.print(cmd, BYTE);
  Serial.print(data1, BYTE);
  Serial.print(data2, BYTE);
  digitalWrite(ledPin,LOW);
}
```

จากโปรแกรมเบื่องต้น การทำงานจะเริ่มต้นทำการอ่านค่าจาก piezoAPin เมื่อค่าของ piezoAPinมีค่ามากกว่าค่าเริ่มต้น จะเริ่มนับเวลา (t=0) ทำการอ่านค่าไปเรื่อยๆ ในขณะที่ค่ายังมากกว่าครึ่งหนึ่งของค่าเริ่มต้นให้ทำการนับเวลาต่อไป(เพิ่มเวลา t++)จนกระทั่งมีค่าน้อยกว่าครึ่งหนึ่งของค่าเริ่มต้น จะไปทำคำสั่ง noteOn ซึ่งจะส่ง MIDI massage ออกไป ซึ่ง MIDI massage จะประกอบด้วย Channel ของชนิดเครื่องดนตรี , โน๊ตดนตรี , และความเร็วซึ่งให้ผลเป็นความแรงของการตี จากนั้นจะทำการ สร้างไฟล์ชื่อ DRUMNOTE.TXT ขึ้นมา ทำการเปิดไฟล์ โดยใช้คำสั่ง .open(&root, name, O_CREAT | O_APPEND | O_WRITE) โดยคำสั่ง O_APPEND เป็นการเขียนที่จะไม่เขียนทับข้อมูลเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือเป็นการเขียนต่อท้ายข้อมูลล่าสุดลงไป file.timestamp ใช้เพื่อบันทึกเวลาและวันที่ จากนั้น MIDI massage ซึ่งประกอบด้วย Channel, โน๊ตดนตรี , และความเร็วจะถูกส่งไปเก็บข้อมูลลงในไฟล์ชื่อ DRUMNOTE.TXT ที่ถูกสร้างขึ้นในตอนแรก writeCRLF เป็นคำสั่งที่ใช้เพื่อขึ้นบรรทัดใหม่ จากนั้นจะเป็นการปิดไฟล์ และเนื่องจากเราใช้คำสั่ง O_APPEND เมื่อมีสัญญาณเข้ามาใหม่ การบันทึกข้อมูลจึงไม่ทับกับข้อมูลที่มีอยู่เดิม มี Delay เท่ากับ t จากนั้นจะทำคำสั่ง noteOff ซึ่งมี velocity = 0 นั่นคือไม่มีการตีกลอง ไม่มีสัญญาณเข้ามา จึงไม่มีความดังเสียง จากนั้นโปรแกรมจะทำการวน loop เพื่อตรวจจับสัญญาณอนาลอกที่ได้รับจากเปียโซเซนเซอร์ตัวอื่นๆต่อไป จนกว่าจะมีสัญญาณเข้ามาอีกครั้ง การทำคำสั่ง noteOn แล้วจึงมีการบันทึกค่าอีกครั้ง เมื่อนำ SD Card ที่ได้บันทึกข้อมูล ไปทำการเปิดไฟล์ในคอมพิวเตอร์ จะได้ข้อมูลดังนี้

<u>ผลการทดลองตอนที่ 3</u>



รูปที่ 4.2.3 ผลการทดลองตอนที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลอง

รายงานฉบับนี้เป็นการศึกษาค้นคว้าเรื่องการทำงานของเปียโซ นำมาประยุกต์ใช้กับกลอง และเรียนรู้โปรแกรม Arduino ผ่านระบบควบคุม Microcontroler MAGA1280 เป็น MIDI code และทำการบันทึกค่าข้อมูลประวัติการตีกลองลง SD Card

ในการทดลองตอนที่ 4.2.1 เป็นการศึกษาสัญญาณไฟฟ้าที่ได้รับเมื่อมีการกระทำเชิงกลกับเปียโซ ในสภาวะที่ต่างๆกัน ได้แก่ วัสดุที่ใช้เคาะ, วัสดุที่ใช้รองแผ่นเปียโซ , ระยะในการเคาะเซนเซอร์หลายๆระยะ รวมทั้งศึกษาความสัมพันธ์ของvelocity กับความแรงในการเคาะที่เซนเซอร์

จากการทดลองตอนที่ 4.2.2 และ 4.2.3 จะเห็นได้ว่าข้อมูลที่ได้จะถูกจัดเก็บในรูปของรหัส MIDI ซึ่งรหัส MIDI ที่ใช้ เป็นมิดิมาตรฐาน มีทั้งหมด 3 byte สำหรับbyteแรกจะเป็นChannel ซึ่งบอกถึงชนิดของเครื่องดนตรี ในที่นี้กลองมี Channel 16 ต่อมาเป็นโน๊ตดนตรี โดยโน๊ตของกลองก็คือส่วนประกอบต่างๆของกลองซึ่งจะมีรหัสแตกต่างกันออกไป และสุดท้ายคือความเร็วของการตี ซึ่งให้ผลเป็นความดังเบาของการตี เหตุนี้เราจึงสามารถกำหนดค่าความดังเบาได้ ทำให้เกิดความเสมือนจริงในการเล่นมากขึ้น

SD Card เป็น memory ที่สะดวกในการบันทึกข้อมูล และ Arduino สามารถสื่อสารผ่าน SPI interface ได้ จึงได้มีการพัฒนาส่วนของโปรแกรมเพื่อเอื้ออำนวยความสะดวกต่อผู้ใช้งาน โดยสามารถเล่นกลองไฟฟ้าแล้วนำข้อมูลที่ถูกบันทึกลงSD Cardโดยไมโครคอนโทรลเลอร์ นำไปเขียนเป็นโน๊ตกลอง บันทึกการใช้งาน หรือประโยชน์อื่นๆได้

โครงงานนี้มีระยะเวลาในการศึกษาค้นคว้าตลอดปีการศึกษา ซึ่งในภาคเรียนที่1 ได้ศึกษาค้นคว้าเกี่ยวกับเปียโซ เรียนรู้การใช้โปรแกรม Arduino โดยสามารถเขียนแปลงสัญญาณที่ได้รับมาจากเซนเซอร์การสั่นสะเทือนเปียโซเป็นสัญญาณ MIDI code ส่งไปยัง MIDI synthesizer ออกเป็นเสียงตามต้องการได้

ในภาคเรียนที่ 2 ศึกษาเกี่ยวกับ SPI interface ศึกษาการเขียนและอ่านข้อมูลในเอสดีการ์ด โดยนำมาประยุกต์ใช้กับการบันทึกข้อมูล เมื่อมีการจับสัญญาณจากเปียโซเซนเซอร์ ทำให้สามารถตีกลองและบันทึกได้ไปพร้อมๆกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] เว็บไซต์ http://www.arduino.cc

[2] เว็บไซต์ http://www.etteam.com

[3] ETT, "เรียนรู้เข้าใจใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino"

[4] รศ.ดร.เติมพงษ์ เพ็ชรกูล, "เอกสารประกอบการเรียน Sensors and Transducers"

[5] Michael McRoberts, Beginning Arduino

[6] M.McRoberts, Arduino Starter Manual, Earthshine Design

[7] เว็บไซต์ http://msdn.microsoft.com/en-us/library/dd798679(v=vs.85).aspx

[8] เว็บไซต์ http://www.cs.uccs.edu/~cs525/midi/midi.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้