

การควบคุม MIDI บนเครื่องดนตรี Theremin

Control of MIDI for Theremin

ปริญญา สถาพรเจริญยิ่ง

พลัช สุขเจริญยิ่งยง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

ปริญญาโท การศึกษา 2557

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุม MIDI บนเครื่องดนตรี Theremin

Theremin MIDI Controller

ผู้จัดทำ

- | | | | |
|--------------|----------------|--------------|----------|
| 1. นาย ปฎิญา | สถาพรเจริญยิ่ง | รหัสนักศึกษา | 54010752 |
| 2. นาย พลัช | สุขเจริญยิ่งยง | รหัสนักศึกษา | 54010891 |



(อาจารย์ สรยุทธ กลมกล่อม) อาจารย์ที่ปรึกษา

การควบคุม MIDI บนเครื่องดนตรี Theremin

นายปฎิญา สถาพรเจริญยิ่ง รหัสนักศึกษา 54010752

นายพลัซ สุขเจริญยิ่ง รหัสนักศึกษา 54010891

อาจารย์ สรยุทธ กลมกล่อม อาจารย์ที่ปรึกษา

ปีการศึกษา 2557

บทคัดย่อ

MIDI Controller คือเครื่องดนตรีสำหรับเชื่อมต่อคอมพิวเตอร์ที่ใช้สำหรับควบคุมโปรแกรมต่างๆ ที่รองรับ MIDI Protocol ส่วนมาก ทำอยู่ในรูปแบบ knob slider และ keyboard แต่มีรูปแบบหนึ่งที่ยังไม่เป็นที่แพร่หลาย คือ Antenna ซึ่งมีเครื่องดนตรีประเภทนี้อยู่แล้ว เรียกว่า Theremin ซึ่งมีราคาสูง และทุก Prototype ที่ออกแบบมาไม่มีความต่อเนื่องของเสียงสูงเสียงต่ำ ซึ่งเป็นเสน่ห์ของเครื่องดนตรีนี้ จึงได้ออกแบบ MIDI Controller ชนิดนี้ในรูปแบบใหม่ขึ้นมา โดยใช้ Microcontroller เป็นตัวกลางรับไฟฟ้า กระแสตรง จากวงจร Analog ที่เชื่อมต่อกับเสาอากาศ และ ถูกออกแบบขึ้นมาเฉพาะกิจ แล้วนำค่าที่ได้รับ มาประมวลผล ส่ง MIDI เข้าคอมพิวเตอร์ Microcontroller จะทำหน้าที่ในการ Sampling ค่าที่ได้รับ จากวงจร แบ่งช่วงค่าที่ได้รับให้ผู้เล่นควบคุมเสียงสูงต่ำได้ง่ายขึ้น และ คำนวณค่าที่ใช้สำหรับส่ง MIDI Messages

Control of MIDI for Theremin

Mr. Patinya Sathaporncharoenying 54010752

Mr. Parat Sukcharoenyingyong 54010891

Asst Sorayut klomklomm Advisor

Academic Year 2014

Abstract

Most of MIDI Controllers, digital instruments, used for controlling MIDI supported program, generally are built in with knobs, sliders or keyboard but there is also a fascinating alternative media which controlled without physical contact. The antenna can be used based on Theremin. Common Theremins are high cost and they send out MIDI note without smooth pitch bend which is an attractive point of Theremin. Therefore, I designed a new low-cost Theremin MIDI controller that could produce continuous pitch message to computer. Development board is brought to be an interface between specifically designed Analog Circuit connected with the antenna and computer. The microcontroller handles sampling process, value quantization for simplifying pitch control and compatible value calculation for MIDI Message.

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงไปด้วยดีเนื่องจากได้รับความช่วยเหลือแนะนำเป็นอย่างดีจากอาจารย์ที่ปรึกษา คือ อาจารย์ สรยุทธ กลมกล่อม ในการแนะนำ ชี้แนะ ให้ข้อเสนอแนะ ตรวจสอบ ข้อบกพร่องของการทำโครงการ และติดตามความก้าวหน้าในการดำเนินการโครงการ กลุ่มผู้จัดทำโครงการรู้สึกซาบซึ้งในความกรุณาของอาจารย์เป็นอย่างยิ่ง และขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ท้ายที่สุดกลุ่มผู้จัดทำโครงการหวังว่าโครงการฉบับนี้จะเป็นประโยชน์แก่ผู้สนใจไม่มากก็น้อย สำหรับข้อบกพร่องต่างๆที่อาจจะเกิดขึ้นนั้น กลุ่มผู้จัดทำโครงการขอน้อมรับผิดทุกประการ และยินดีที่จะรับฟังคำแนะนำจากทุกท่านที่ได้เข้ามาศึกษา เพื่อเป็นประโยชน์ในการพัฒนาโครงการต่อไป

นายปริญญญา สถาพรเจริญยิ่ง

นายพลัช สุขเจริญยิ่งยง

สารบัญ

หน้า

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	IV
สารบัญตาราง	V
สารบัญรูป	VII

บทที่ 1 บทนำ

1.1 ความเป็นมาของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 วิธีการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 Theremin.....	3
2.2 MIDI.....	7
2.3 Arduino : Teensy 3.1.....	11
2.4 Exponential to Linear Converting.....	16

สารบัญ(ต่อ)

หน้า

บทที่ 3 การออกแบบและพัฒนา

3.1 การทำงานโดยรวมของโครงการ18

3.2 ขั้นตอนการดำเนินงาน.....24

บทที่ 4 การทดลอง

4.1 วงจรกำเนิดSquare Wave27

4.2 วงจร Frequency to Voltage28

4.3 Teensy Board29

บทที่ 5 สรุปผลการทดลอง

5.1 สิ่งที่ได้34

5.2 ปัญหาอุปสรรคแนวทางแก้ไข.....35

5.3 แนวทางการพัฒนาต่อ36

บรรณานุกรม38

สารบัญตาราง

ตาราง	หน้า
2.2.1 Part of Note Messages Table	7
2.2.2 Part of Control Change Messages Table.....	8-9
2.4.1 ตารางแปลงกราฟที่เป็น Exponential ไปเป็น linear.....	16
4.1.4 ตารางที่ค่าเปลี่ยนแปลง.....	29
4.1.5 ตารางค่า K.....	31
4.1.6 ตารางเปรียบเทียบระยะห่างระหว่างมือกับAntennaกับค่าที่ใช้ในการส่ง CC Value.....	31
4.1.7 ตารางเปรียบเทียบระยะห่างระหว่างมือกับAntennaกับค่าที่ใช้ในการส่ง Pitch bend.....	33

สารบัญรูป

รูป	หน้า
2.1.1 Analog Theremin with Oscilloscope.....	3
2.1.2 Leon Theremin	4
2.1.3 Block diagram ของ Theremin	4
2.1.4 ส่วนของ Circuit Theremin (Pitch Oscillator).....	5
2.1.5 การต่อ capacitor แบบขนาน	5
2.1.6 LM2907/LM2917 Frequency to Voltage Converter	6
2.1.7 วงจรกับกราฟความสัมพันธ์ระหว่างแรงดันไฟฟ้ากับความถี่คลื่น	6
2.2.2 ภาพเปรียบเทียบ ความถี่ คีย์บอร์ด ตัวโน้ต และ MIDI number	8
2.2.3 MIDI Mapping.....	9
2.2.4 MIDI Mapping & MIDI Controller	10
2.2.5 ตัวอย่าง DAW Software	10
2.2.6 ตัวอย่าง VST Instrument	11
2.3.1 ลักษณะ Teensy 3.1.....	12
2.3.2 Teensy 3.1 กับ Input_Pullup	15
2.3.3 ตำแหน่งบน Teensy 3.1.....	15
2.4.1 Exponential to Linear Converting.....	16
3.1.1 วงจรกำเนิด Square Wave.....	18
3.1.2 วงจร Frequency to Voltage.....	19
3.1.3 Theremin.....	20
3.1.4 Flowchart การทำงานโดยรวมของ Board	21
3.2.1 Arduino Website	25
4.1.1 วงจรกำเนิด Square Wave.....	27
4.1.2 วงจร Frequency to Voltage.....	29
4.1.3 Teensy Board.....	28
4.1.8 ทดลองเล่นMIDI Controller	33
5.1.1 MIDI Theremin.....	34

สารบัญรูป(ต่อ)

รูป	หน้า
5.3.2 ตัวอย่างที่การใช้ใน MIDI Library	36
5.3.3 ตัวอย่างการต่อลำโพง	37

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

ในยุคปัจจุบันเครื่องดนตรีมีหลายชนิดมากมาย ไม่ว่าจะเป็น กีตาร์ กลอง เบส เปียโน เป็นต้น แต่ส่วนใหญ่จะสามารถเชื่อมต่อกับคอมพิวเตอร์โดยผ่าน MIDI Protocol ซึ่งมีเครื่องดนตรีชนิดหนึ่งไม่เป็นที่แพร่หลายในการเชื่อมต่อกับ MIDI นั่นก็คือ Theremin จึงได้ออกแบบ MIDI Controller ชนิดนี้ในรูปแบบใหม่ขึ้นมา โดยใช้ Microcontroller เป็นตัวกลางรับไฟฟ้ากระแสตรง จากวงจร Analog ที่เชื่อมต่อกับเสาอากาศ และ ถูกออกแบบขึ้นมาเฉพาะกิจ แล้วนำค่าที่ได้รับมาประมวลผล ส่ง MIDI เข้าคอมพิวเตอร์ Microcontroller จะทำหน้าที่ในการ Sampling ค่าที่ได้รับจากวงจร แบ่งช่วงค่าที่ได้รับให้ผู้เล่นควบคุมเสียงสูงต่ำได้ง่ายขึ้น และ คำนวณค่าที่ใช้สำหรับส่ง MIDI Messages เพื่อเพิ่มประสิทธิภาพในการเล่นดนตรีสมัยนี้ให้มีความน่าสนใจมากยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงการ

- 1).สร้าง MIDI Controller ที่ใช้หลักการของ Theremin
- 2).พัฒนาให้ MIDI Theremin ให้ดีกว่าที่มีอยู่แล้ว
- 3).สร้าง Feature ที่น่าสนใจและแตกต่างจาก MIDI Controller ทั่วไป

1.3 ขอบเขตของโครงการ

- 1).เรียนรู้และศึกษา MIDI Protocol
- 2).เรียนรู้และศึกษาการใช้งาน Arduino Library และ Teensy Library ที่เกี่ยวข้องกับการใช้พัฒนาโปรแกรม
- 3).ใช้บอร์ด Teensy 3.1 และ ส่วนของวงจร Theremin
- 4).ตรวจสอบผลลัพธ์จากโปรแกรม Digital audio workstation ทั่วไป

1.4 วิธีการดำเนินการ

- 1).ศึกษาการทำงานของ Theremin และ ศึกษาวงจรต่างๆที่ใช้ในการทำงาน
- 2).ศึกษาการทำงานของ MIDI Controller และ MIDI Protocol ต่างๆ
- 3).ศึกษาการใช้โปรแกรมที่ใช้โปรแกรมบอร์ด Teensy 3.1
- 4).ออกแบบการเชื่อมต่อของเครื่องดนตรีเข้ากับคอมพิวเตอร์และบอร์ด ว่ามี Input และ Output ใดบ้าง
- 5).ออกแบบวงจรในส่วนของ ปุ่มกับสวิทช์ และ วงจร Theremin
- 6).ออกแบบการทำงานบนบอร์ด
- 7).เตรียมอุปกรณ์ และ ติดตั้งโปรแกรมที่ต้องใช้งาน
- 8).สร้างวงจร Prototype บน Breadboard ทดลองผลและบันทึก
- 9).พัฒนาโปรแกรม และ ทดลองผล กับวงจร
- 10). บัดกรีวงจรลงบอร์ดแล้ว ประกอบเป็น Package

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1).ได้เรียนรู้เข้าใจเกี่ยวกับวงจรอิเล็กทรอนิกส์
- 2).ได้พัฒนาทักษะการวิเคราะห์ บันทึกค่า และ คำนวณ
- 3).ได้พัฒนาทักษะในการเขียนโปรแกรม
- 4).เข้าใจการทำงานของ Development Board ที่เลือกใช้
- 5).ได้ MIDI Controller ที่ใช้งานได้จริง

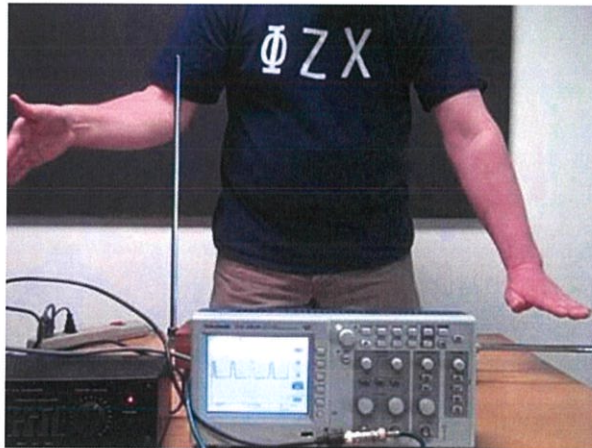
บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 Theremin

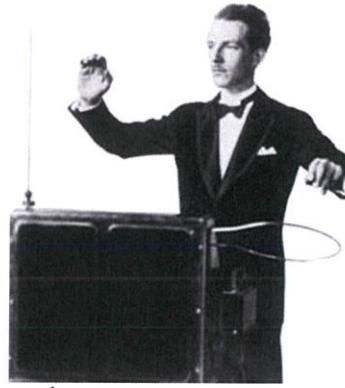
2.1.1 ลักษณะ

จัดเป็นเครื่องดนตรีประเภท Continuous Pitch (มีความต่อเนื่องของความถี่เสียง) ซึ่งผู้เล่นจะต้องประมาณระยะตัวโน้ตด้วยตนเอง ดังเช่น Violin Cello ฯลฯ



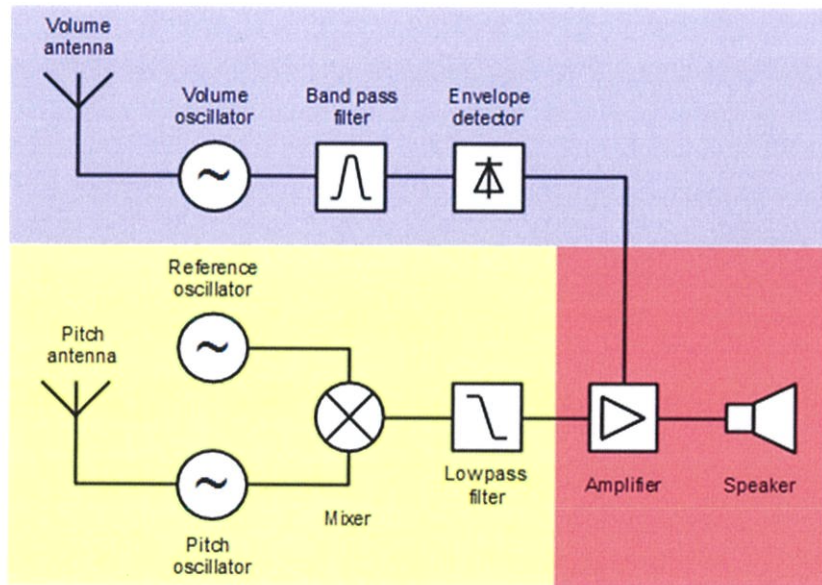
รูปที่ 2.1.1 Analog Theremin with Oscilloscope

Theremin อาศัยหลักการของคลื่นวิทยุ โดยผู้เล่นสามารถเล่นโดยไม่ต้องสัมผัสถูกเครื่องดนตรี วิธีการเล่นก็คือ Theremin ก็คือการใช้มือสองมือเคลื่อนที่ไปมาระหว่างเสาอากาศสองต้น ระยะห่างระหว่างมือกับเสาอากาศต้นแรกใช้เป็นการกำหนดความถี่ (Theremin สำหรับบางตัวที่มีราคาสูงจะใช้โวลุ่มหมุนแทน และใช้มือกำหนด pitch เท่านั้น)

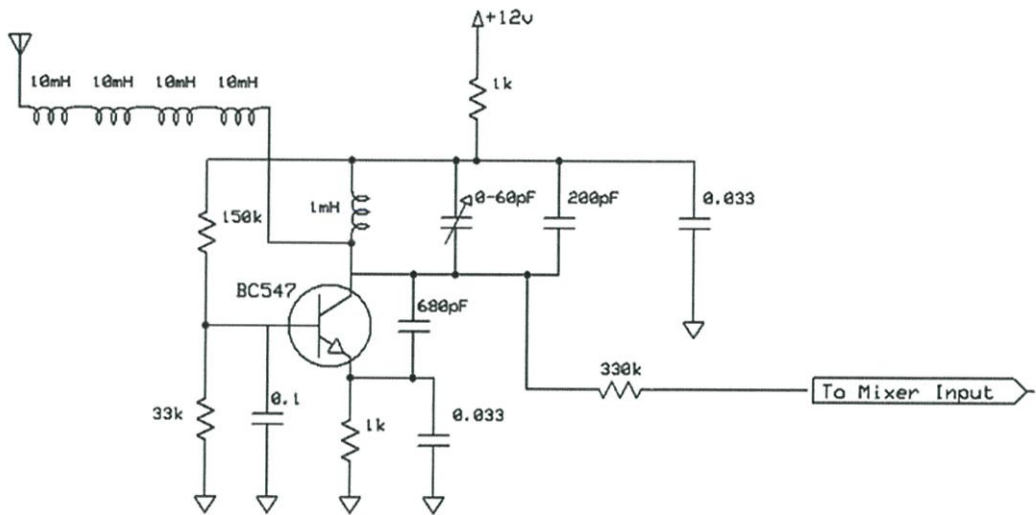


รูปที่ 2.1.2 Leon Theremin

2.1.2 หลักการทำงาน



รูปที่ 2.1.3 Block Diagram ของ Theremin

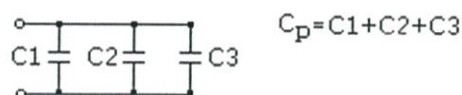


รูปที่ 2.1.4 ส่วนของ Circuit Theremin (Pitch Oscillator)

หลักการของเครื่องคือการทำให้กำเนิดเสียงโดย ทฤษฎี Heterodyne* ตัวเครื่องดนตรีจะมี oscillator (ตัวสร้างคลื่นความถี่) สองตัว ตัวหนึ่งผลิตคลื่นความถี่แบบคงที่ อีกตัวหนึ่งจะปรับความถี่ขึ้นกับระยะห่างระหว่างมือผู้เล่นและเสาอากาศ โดยมือผู้เล่นทำตัวเป็น กราวด์เพลทที่ทำหน้าที่เหมือนมี variable capacitor จากนั้นความถี่ที่แตกต่างกันของ oscillator สองตัวจะทำให้เกิดระดับของความถี่เสียงซึ่งจะผ่านแอมป์ไฟลู่ลำโพงอีกที

*Heterodyne คือ เทคนิคการประมวลผลสัญญาณวิทยุ

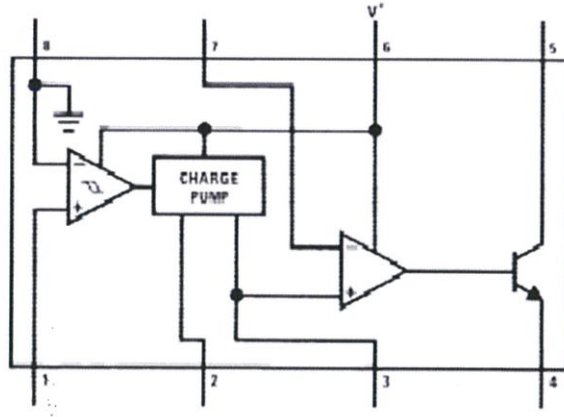
การทำ variable capacitor ในแบบนี้ จะต้องทำให้กระแสที่ไหลในวงจร มีค่าต่ำ และ มีความถี่ของสัญญาณสูง เนื่องจาก ค่า capacitor แปรผกผันกับความถี่ แต่เมื่อผู้เล่นทำตัวเป็น กราวด์เพลท จะสามารถปรับค่าของ capacitor ได้ต่ำกว่า 1pF เท่านั้น และ เป็นการเชื่อมต่อแบบขนาน



รูปที่ 2.1.5 การต่อ capacitor แบบขนาน

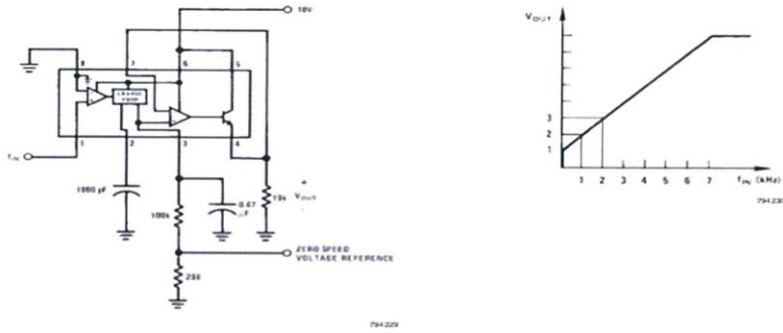
Frequency to Voltage Converter

ตัวแปลงความถี่เป็นค่าแรงดันไฟฟ้า โดยมีลักษณะดังภาพด้านล่าง

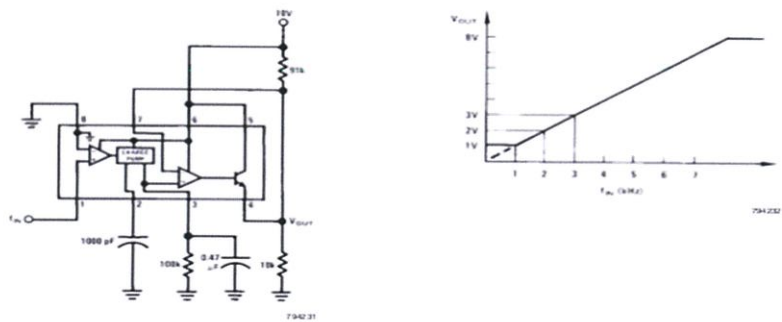


รูปที่ 2.1.6 LM2907/LM2917 Frequency to Voltage Converter

Changing the Output Voltage for an Input Frequency of Zero



Changing Tachometer Gain Curve or Clamping the Minimum Output Voltage



รูปที่ 2.1.7 วงจรกับกราฟความสัมพันธ์ระหว่างแรงดันไฟฟ้ากับความถี่คลื่น

จากวงจร เมื่อความถี่สูงขึ้นแรงดันไฟฟ้าจะเพิ่มขึ้นด้วยตามกราฟ

2.2 MIDI

2.2.1 Midi Messages

ส่วนที่ใช้แบ่งเป็น 2 อย่างคือ

1.Note Message

STATUS BYTE		DATA BYTES	
1st Byte Value	Function	2nd Byte	3rd Byte
Binary Hex Dec			
10000000 = 80 = 128	Chan 1 Note off	Note Number (0-127)	Note Velocity (0-127)
10000001 = 81 = 129	Chan 2 Note off	Note Number (0-127)	Note Velocity (0-127)
10000010 = 82 = 130	Chan 3 Note off	Note Number (0-127)	Note Velocity (0-127)
10010000 = 90 = 144	Chan 1 Note on	Note Number (0-127)	Note Velocity (0-127)
10010001 = 91 = 145	Chan 2 Note on	Note Number (0-127)	Note Velocity (0-127)
10010010 = 92 = 146	Chan 3 Note on	Note Number (0-127)	Note Velocity (0-127)
11100000 = E0 = 224	Chan 1 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11100001 = E1 = 225	Chan 2 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)
11100010 = E2 = 226	Chan 3 Pitch Bend Change	Pitch Bender LSB (0-127)	Pitch Bender MSB (0-127)

ตาราง 2.2.1 Part of Note Messages Table

Note On/Off มีไว้สำหรับเล่นตัวโน้ตและหยุดการเล่นตัวโน้ต โดยมี Byte ที่กำหนด ตัวโน้ต และ Byte ที่กำหนดน้ำหนักที่เล่น

เครื่องดนตรีที่มีการเปลี่ยนแปลงตัวโน้ตแบบต่อเนื่อง ไม่มีการ Quantize ให้เป็นตัวโน้ต แบบ Chromatic Scale อย่างเช่น Piano



รูป 2.2.2 ภาพเปรียบเทียบ ความถี่ คีย์บอร์ด ตัวโน้ต และ MIDI number

MIDI Pitch bend Change ออกแบบมาสำหรับการเบนความถี่ของของตัวโน้ต โดย Value ประกอบด้วย LSB 1 Byte (0-127) MSB 1 Byte (0-127) รวมกันจะได้ความละเอียดของการ bend note เป็น $128 \times 128 = 16384$ (0-16383) Pitch bend จะสามารถปรับค่าระยะ Semi note ที่ต้องการจะ bend ได้ใน Software

2.2.2 Control change Messages

Table 2: Expanded Status Bytes List

STATUS BYTE		DATA BYTES	
1st Byte Value	Function	2nd Byte	3rd Byte
Binary Hex Dec			
10110000= B0= 176	Chan 1 Control/Mode Change	see Table 3	see Table 3
10110001= B1= 177	Chan 2 Control/Mode Change	see Table 3	see Table 3
10110010= B2= 178	Chan 3 Control/Mode Change	see Table 3	see Table 3
10110011= B3= 179	Chan 4 Control/Mode Change	see Table 3	see Table 3
10110100= B4= 180	Chan 5 Control/Mode Change	see Table 3	see Table 3
10110101= B5= 181	Chan 6 Control/Mode Change	see Table 3	see Table 3
10110110= B6= 182	Chan 7 Control/Mode Change	see Table 3	see Table 3
10110111= B7= 183	Chan 8 Control/Mode Change	see Table 3	see Table 3
10111000= B8= 184	Chan 9 Control/Mode Change	see Table 3	see Table 3
10111001= B9= 185	Chan 10 Control/Mode Change	see Table 3	see Table 3

10111010= BA= 186	Chan 11 Control/Mode Change	see Table 3	see Table 3
10111011= BB= 187	Chan 12 Control/Mode Change	see Table 3	see Table 3
10111100= BC= 188	Chan 13 Control/Mode Change	see Table 3	see Table 3
10111101= BD= 189	Chan 14 Control/Mode Change	see Table 3	see Table 3
10111110= BE= 190	Chan 15 Control/Mode Change	see Table 3	see Table 3
10111111= BF= 191	Chan 16 Control/Mode Change	see Table 3	see Table 3

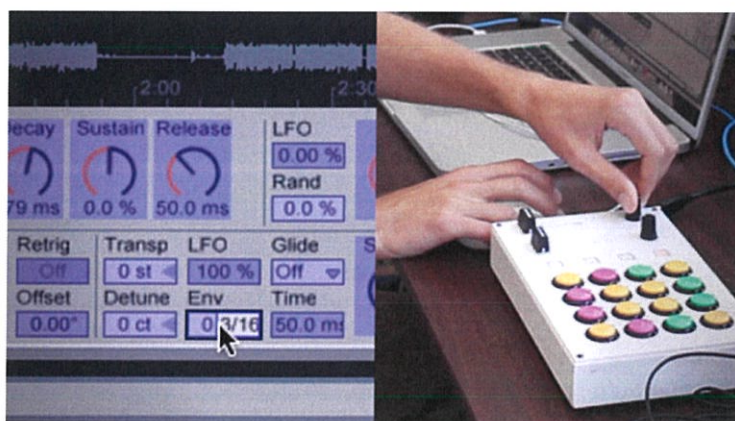
ตาราง 2.2.2 Part of Control change Messages Table

Control change มีไว้สำหรับปรับค่าต่างๆใน software โดย User จะต้อง Map ตัว MIDI Controller กับ Parameter ในโปรแกรม

Channel	Note/Control	Path	Name	Min	Max
1	CC 10	Vocals	Select Track		
1	CC 28	Vocals Mixer	A-Return	-inf dB	0.0 dB
1	CC 74	Lead Grain Delay	Feedback	0.0 %	95 %
1	CC 82	Vocals Mixer	Track Volume	-inf dB	6.0 dB
1	CC 83	Lead Mixer	Track Volume	-inf dB	6.0 dB
1	Note C3	Vocals Mixer	Speaker On		
1	Note C4	Vocals Mixer	Solo/Cue		
1	Note D4	Drums Mixer	Solo/Cue		
1	Note E3	Lead Mixer	Speaker On		

รูป 2.2.3 MIDI Mapping

จากภาพ ใน Row ที่สอง หมายถึง CC# 28 ของ Channel 1 ถูก map กับ A-Return (ใช้ในการส่งเสียงของ Track ไปที่ A-Return) ในส่วนของ Mixer ใน Track Vocal โดยสามารถปรับค่าได้ตั้งแต่ -inf dB ถึง 0.0 dB



รูป 2.2.4 MIDI Mapping & MIDI Controller

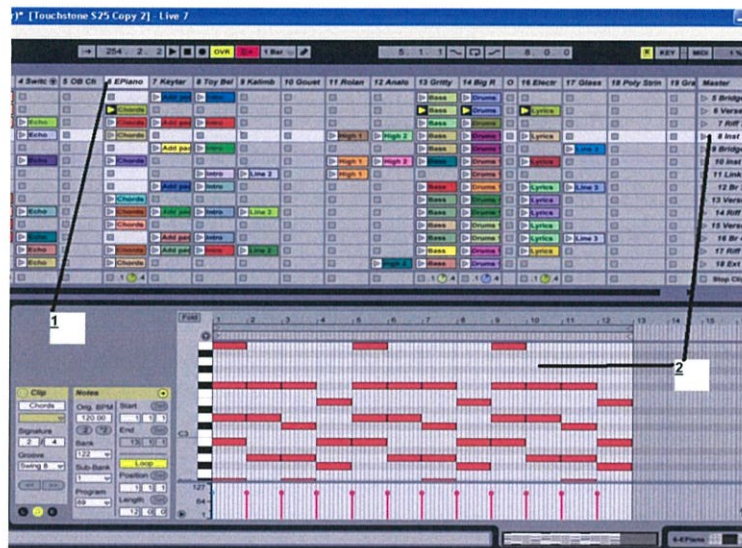
จากภาพเป็นการเปิดโหมด MIDI Mapping ในโปรแกรม Ableton จะเห็นปุ่มที่สามารถ Map ได้ เป็นสีน้ำเงิน ซึ่งขณะนี้ กำลัง Map knob ที่กำลังหมุนอยู่ กับ Env (Envelope level เป็น

parameter ควบคุม ผลของ Envelope เป็น percentage) เมื่อมีการหมุน knob หลังจากปิด MIDI Mapping จะทำให้ค่า Env เปลี่ยนแปลงตามการหมุน

2.2.3 MIDI compatible software

- DAW (Digital Audio Workstation)

- โปรแกรมสำหรับสร้างเสียงไม่ว่าจะเป็นดนตรีหรือเสียงเอฟเฟค



รูป 2.2.5 ตัวอย่าง DAW Software

- VST (Virtual Studio Technology)

เป็นจำพวกโปรแกรมสังเคราะห์เสียงขึ้นมา หรือ เป็นพวกเอฟเฟค ซึ่งมักออกแบบมาเพื่อจำลอง Hardware ที่มีจริง อาจอยู่ในรูปแบบของ Standalone หรือ Plugin สำหรับเชื่อมต่อกับโปรแกรมอื่นๆเช่น DAW



รูป 2.2.6 ตัวอย่าง VST Instrument

2.3 Arduino : teensy 3.1

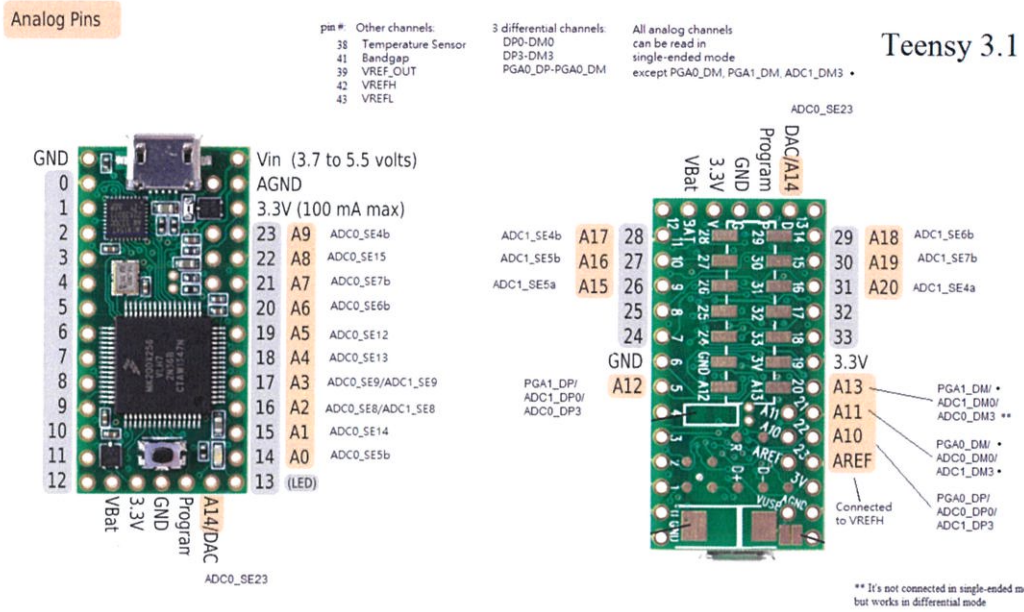
2.3.1 Arduino คือ

Arduino คือเครื่องมือที่จะทำให้คอมพิวเตอร์สามารถรับสัญญาณจากภายนอกและส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพมากกว่าใช้เครื่องพีซีตั้งโต๊ะ ตัวบอร์ดออกแบบจากไมโครคอมพิวเตอร์ชิพเดียว, และมีโปรแกรมพัฒนาสำหรับเขียนโปรแกรมให้บอร์ดทำงาน Arduino สามารถประยุกต์ทำเครื่องใช้อัจฉริยะ รับสัญญาณจากสวิทช์ หรือ เซนเซอร์, และควบคุม หลอดไฟ, มอเตอร์, หรืออุปกรณ์อื่นๆ และ Arduino เป็นได้ทั้งแบบทำงานอิสระ หรือทำงานติดต่อกับโปรแกรมที่ทำงานบนเครื่องพีซี

บอร์ด Arduino ใช้พัฒนาการใช้งาน Microcontroller ในตระกูล AVR ที่กำลังได้รับความนิยมอย่างสูงทั่วโลก เพราะว่าเป็น Open Source ท่านสามารถดัดแปลง ไปใช้งานได้ทั้ง Hardware และ software ได้ทันที ภาษาที่ใช้กับบอร์ดนี้จะเป็นลักษณะของ C/C+ โดยจัดให้มี Libraries ต่างๆ ให้พร้อมให้เรียกใช้งานได้ทันทีมากมาย ครอบคลุมการติดต่อกับ I/O ต่างๆได้กว้างมาก การใช้งานก็ง่ายเพียงแต่เสียบสาย USB ติดตั้งโปรแกรมที่ให้มาด้วยท่านก็พร้อมที่จะก้าวเข้าสู่โลกของการประยุกต์ใช้ Micro Controller ได้ทันที

2.3.2 Arduino : Teensy 3.1 คือ

Teensy 3.1 คือบอร์ดไมโครคอนโทรลเลอร์ตระกูล ARM Cortex M4 96MHz



รูป 2.3.1 ลักษณะ Teensy 3.1

1. มีส่วนรับ Analog input ซึ่งมี ADC แปลงไฟสูงสุด 5V เป็น Digital 10bit (0-1023)
2. ส่วนของ Digital input สามารถรับค่า Voltage เป็น High (สูงกว่า 3V) Low (ต่ำกว่า 3V)
3. MicroUSB ในการเชื่อมต่อสำหรับการโปรแกรม และรองรับการรับส่ง MIDI
4. สามารถโปรแกรมด้วย Arduino IDE เมื่อลง Teensyduino ซึ่งเป็น add-on ของโปรแกรม และใช้ C Programming

2.3.3 Arduino & Teensy Libraries

2.3.3.1 Bouncing

`Bounce myButton = Bounce(pin, milliseconds);`

สร้าง Bounce object myButton, โดยใช้ "pin" เป็น พารามิเตอร์ตัวแรก และตัวที่สองใช้หน่วยการทำให้ Bouncing คือ milliseconds ซึ่งอาจสร้างหลายๆ Bounce Object เพื่อสำหรับ Pin ที่ลักษณะการทำงานแตกต่างกันออกไป.

`myButton.update();`

การอ่านและอัปเดตสถานะของข้อมูล จะทำได้โดยใช้ฟังก์ชัน update() โดยควรตรวจสอบอัปเดตอยู่เสมอ เพื่อให้ข้อมูลเกี่ยวกับ Pin นั้นมีความถูกต้อง แล้ว return ค่าจริงออกมา

```
myButton.fallingEdge();
```

ตรวจสอบสถานะจากระดับสูงไปต่ำโดยเช็คที่ระหว่าง Pin กับ Ground

```
myButton.risingEdge();
```

ตรวจสอบสถานะจากระดับต่ำไปสูงโดยเช็คที่ระหว่าง Pin กับ Ground

2.3.3.2 USB MIDI

การส่งตัวโน้ต ผ่าน USB

```
usbMIDI.sendNoteOn(note, velocity, channel)
```

```
usbMIDI.sendNoteOff(note, velocity, channel)
```

- note มีตัวโน้ต 128 ตัว ตั้งแต่ 0-127
- velocity น้ำหนัก 128 ระดับ 0-127
- channel มี 16 channels 1-16

```
usbMIDI.sendControlChange(control, value, channel)
```

- control cc# 128 numbers 0-127
- value ค่าปรับอยู่ 128 ระดับ 0-127

```
usbMIDI.sendPitchBend(value, channel)
```

- value ในการ bend Pitch 16384 ระดับ 0-16383
- channel มี 16 channels 1-16

2.3.3.3 I/O

Input vs Output: pinMode()

เมื่อไหร่ที่จะเริ่มการทำงาน pin จะทำงานใน input mode โดยสามารถควบคุม pin ด้วยฟังก์ชัน pinMode()

```
pinMode(PIN_D6, OUTPUT);
```

โดย parameter ตัวแรกจะกำหนดค่าของ pin ส่วนตัวที่สองจะบอกว่าเป็น Input หรือ Output

Output: digitalWrite()

ถ้า pin ได้ถูกกำหนดเป็นส่วนของ output จะสามารถทำให้ pin มีแรงดันเป็น HIGH หรือ LOW

ได้ โดยเรียกฟังก์ชัน digitalWrite(). HIGH = แรงดันสูง LOW = แรงดันต่ำ

```
digitalWrite(PIN_D6, LOW); // LED on
```

```
digitalWrite(PIN_D6, HIGH); // LED off
```

Input: digitalRead()

ถ้า pin ได้ถูกกำหนดเป็นส่วนของ input จะสามารถอ่านค่าของ pin ได้ฟังก์ชัน digitalRead() โดยถ้าค่าที่ returns ออกมาเท่ากับ 0 แสดงว่า pin มีสถานะที่ LOW และถ้า returns ออกมาได้เท่ากับ 1 แสดงว่า pin มีสถานะที่ HIGH

```
if (digitalRead(PIN_C2)) {
  // do this if C2 is high
} else {
  // do this if C2 is low
}
```

Input With Pullup

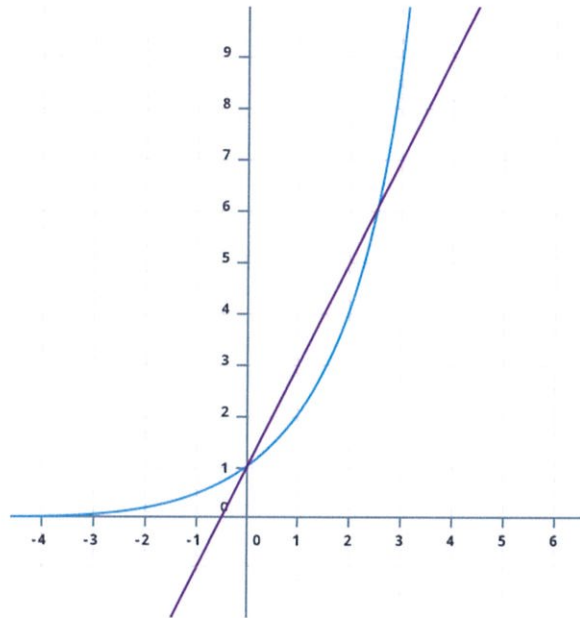
ในทุกๆ pin จะมี ตัวต้านทานชื่อว่า pullup ซึ่งมันจะทำงานต่อเมื่อ pin เป็น input โดยจะใช้ฟังก์ชัน pinMode() ร่วมกับ INPUT_PULLUP.

```
pinMode(PIN_D7, INPUT_PULLUP);
```

ตัว pullup นี้ จะมีประโยชน์เมื่อเชื่อมต่อกับปุ่มกดกับ pin ลง ground แต่ถ้าไม่ได้กดปุ่มมันก็จะไม่การเชื่อมต่อเลย ซึ่งจะเห็นว่าค่าความดันไฟฟ้าที่ pin เป็น High ตอนไม่ได้เชื่อมต่อกับอะไร

การทำงานนั้นอาจจะล่าช้าเล็กน้อยตอนตั้งค่า pinMode() การกำหนดค่า INPUT_PULLUP mode และ digitalWrite() จะไม่สามารถเชื่อมต่อกับ pin ที่มีสถานะเป็น HIGH อยู่ได้ ตัว pullup นี้จะเพิ่ม voltage ให้สูงขึ้น ซึ่งขึ้นอยู่กับค่าความจุไฟฟ้าในวงจร รวมไปถึงค่าความจุไฟฟ้าของ pin และ breadboards หรือ wires

2.4 Exponential to Linear Converting



รูป 2.4.1 Exponential to Linear Converting

LINEAR EQUATION				EXPONENTIAL EQUATION		
x	$y=5x+1$	y	Change in y-value	$Y=5^x$	Y	Change in y-value
-1	$5(-1)+1$	-4		5^{-1}	1/5	
0	$5(0)+1$	1	+5 ("plus 5")	5^0	1	x5 ("times 5")
1	$5(1)+1$	6	+5	5^1	5	x5
2	$5(2)+1$	11	+5	5^2	25	x5
3	$5(3)+1$	16	+5	5^3	625	x5

ตาราง 2.4.1 ตารางแปลงกราฟที่เป็น Exponential ไปเป็น linear

เราสามารถใช้ Logarithm ในแปลงกราฟที่เป็น Exponential ไปเป็น linear

$$y = m(\log(Y)/\log(x))+k$$

โดยที่ Y คือ ค่าในแกน y ของสมการ Exponential

x คือ ตัวยกกำลัง ของสมการ Exponential

m คือค่าความชันของสมการ Linear ใหม่

k คือค่าคงที่ของสมการ Linear ใหม่

บทที่ 3

การออกแบบและพัฒนา

3.1 การทำงานโดยรวมของโครงการ

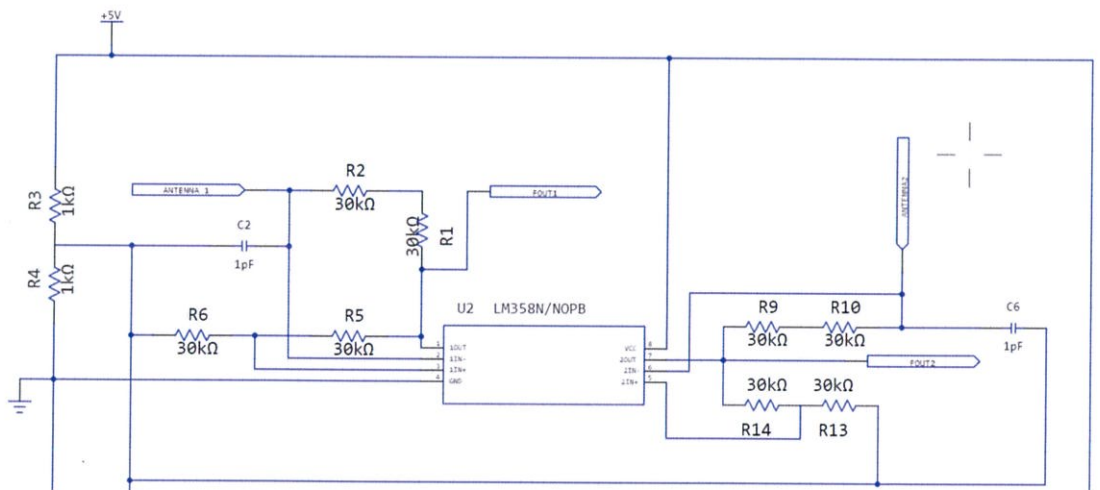
โครงการการควบคุม MIDI บนเครื่องดนตรี Theremin นี้ ได้แบ่งส่วนของการศึกษาออกเป็นสามส่วน คือ

3.1.1 ส่วนของการรับอินพุท

3.1.1.1 วงจร theremin

1 วงจรกำเนิด Square Wave

ใช้ LM358N 1 ตัว (Op-amp 2 ตัว) ต่อ 2 Oscillator

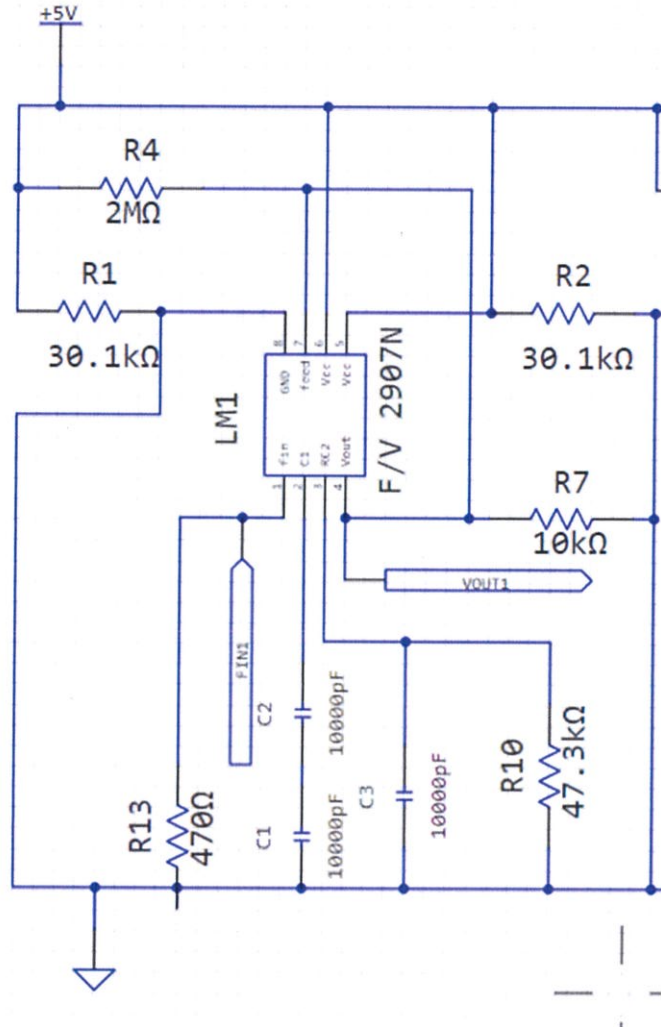


รูป 3.1.1 วงจรกำเนิด Square Wave

มี Antenna เชื่อมต่อที่ ANTENNA Input เพื่อรับการเหนี่ยวนำ

FOUT Output สำหรับส่งออกสัญญาณ ไปยังอีกวงจร

2 วงจร Frequency to Voltage



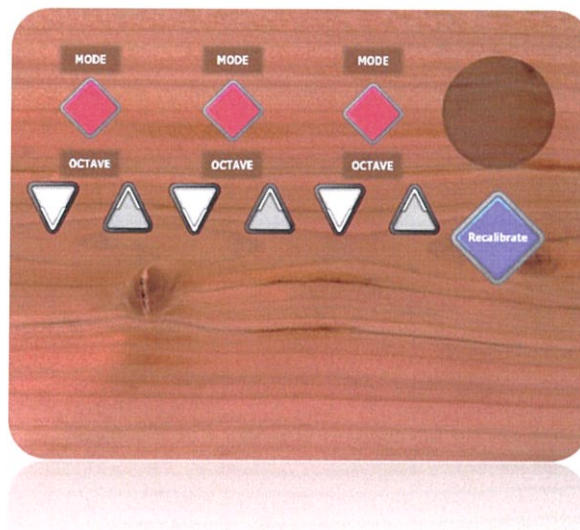
รูป 3.1.2 วงจร Frequency to Voltage

FIN Input จะเชื่อมกับ FOUT จากวงจรกำเนิด Square wave

VOUT Output สำหรับเชื่อมต่อกับ Teensy Analog Input pin

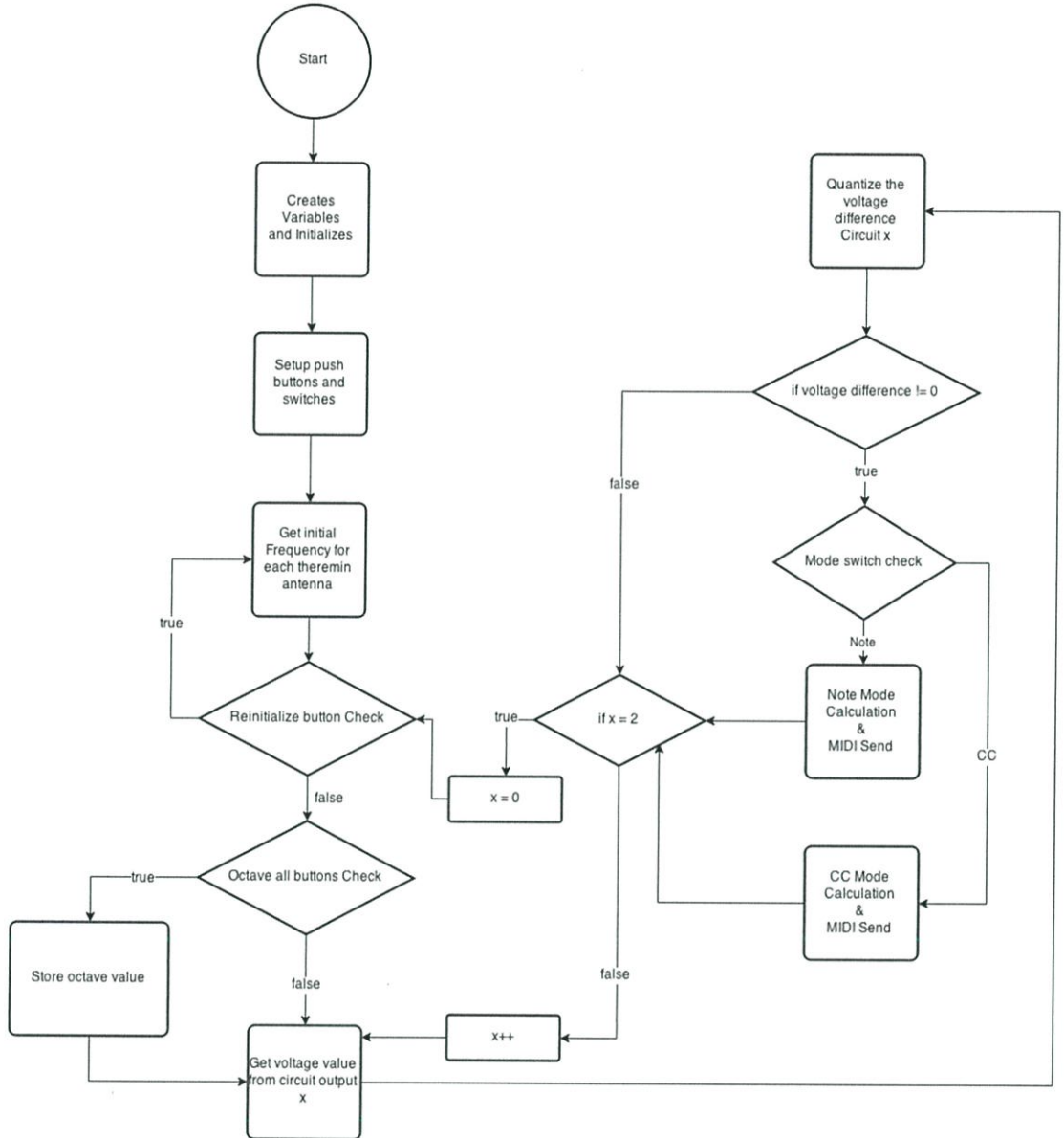
3.1.1.2 Teensy Development Board

1. ส่วนที่รับกระแสไฟฟ้าจากวงจร เป็น analog input มี Theremin 3 ตัว (0V-5V)
2. สวิตช์เปลี่ยน Mode ของ Theremin 3 อัน (เชื่อมกับ Ground) และสวิตช์สำหรับเปิดปิดเครื่อง
3. Push buttons 2 อัน 3 คู่ (6 อัน) สำหรับ 3 Theremin ไว้เพิ่ม octave ในการเล่นโหมด (เชื่อมกับ Ground)



รูปที่ 3.1.3 Theremin

3.1.2 ส่วนของการประมวลผล



รูปที่ 3.1.4 Flowchart การทำงานโดยรวมของ Board

Teensy จะทำงานในส่วนของ function setup ก่อน 1 ครั้งจึงจะเข้า main loop ซึ่งเป็น infinite loop โดย setup() คือตั้งค่าเริ่มต้นต่างๆ สำหรับบอร์ด

○ ส่วน setup()

Set I/O pin ทั้งหมด สวิตช์ ปุ่มกด

Variable initialization กำหนดค่าเริ่มต้นของตัวแปร

- ค่าเริ่มต้นของตัวแปรที่ใช้จับเวลา
- ค่าเริ่มต้นของแรงดันแรกที่ใช้ในการหาผลต่างของแรงดัน
เรียกใช้ function INITVoltpoint() เพื่อหา ค่าต่ำสุดที่เป็นไปได้

○ ส่วน loop()

- เช็คสวิตช์เปิดปิด และ ปุ่มกด Octave
 1. ปิด =>เรียก function INITVoltpoint() / เปิด => ดำเนินโปรแกรมต่อ
 2. มีการกดปุ่ม octave => บันทึกค่า octave ใหม่
- รับค่า Voltage มา ด้วย function getVolt(pin)
- Quantize ผลต่างแรงดัน
- เช็ค สวิตช์ Mode เพื่อเข้า Note mode หรือ CC Mode
- แปลงค่าสำหรับ MIDI Note แล้วส่ง (NoteProcess(pin,diffVolt))
- แปลงค่าสำหรับ MIDI CC แล้วส่ง (CCProcess(pin,diffVolt))

1. Analog Read and ADC Conversion (ส่วนรับกระแสไฟฟ้าจากวงจร)

ใช้ ADC แปลง Analog 0-5 V เป็น 0-1023 integer getVolt(pin) ใช้ analogRead(pin) เพื่อรับค่าแรงดันบน pin นั้นโดยเมื่อเรียก getVolt(pin) 1 ครั้งจะเก็บมา 1 sample เมื่อครบ 750 Samples แล้ว เฉลี่ยค่า แล้วใช้ค่าเฉลี่ยนี้ เป็นค่าแรงดันที่ได้รับ INITVoltpoint() ใช้ getVolt(pin) ในการหาค่าแรงดันเริ่มต้น โดย เรียก getVolt(pin) 2250 ครั้ง ซึ่งจะได้ค่ามา $2250/750 = 3$ ค่า แล้วใช้ค่าที่น้อยที่สุดเป็นแรงดันเริ่มต้นของเฉพาะ Antenna

2. Voltage Difference Quantization

$\text{diffVolt} = (x/y) * (\text{แรงดันใหม่} - \text{แรงดันเก่า})$

$\text{Quantized value} = \text{diffVolt} - (\log_3(\text{diffVolt}) - 3)^3$

เป็น code : $\text{diffVolt} = (15.0/16.0) * (\text{Voltout}[\text{pin}] - \text{INITVoltpoint}[\text{pin}]);$

$\text{diffVolt} = \text{pow}((\log(\text{diffVolt})/\log(3)) - 3.0, 3.0);$

3. คำนวณค่า Control change value หรือ คำนวณค่า Note number ขึ้นอยู่กับ Mode และ octave ปัจจุบัน ที่เลือก

3.1 แปลงค่าที่ได้รับไปอยู่ใน Range ที่สำหรับส่งข้อมูล MIDI

3.1.1 สำหรับ MIDI CC : CC value จะเท่ากับ $(\text{diffVolt} - 50) * 128/300$
และจะมีการส่งหาค่าต่างจากเดิมเกิน 10

3.1.2 สำหรับ MIDI Note : Pitch bend value จะเท่ากับ

$(\text{diffVolt} - 50.00) * (16384.00/300.00)$ และจะมีการส่งหาค่าต่างจากค่าเดิม
ตั้งแต่ 100 ขึ้นไป

3.2 คำนวณค่าที่ใช้ในการส่ง MIDI Note (Octave offset)

$\text{Firstnote} = 36 + (12 * \text{octave})$

4. ส่ง ผ่านทาง USB Cable

`usbMIDI.sendNoteOn(firstnote[pin], 110, pin+1)`

3.1.3 ส่วนของการแสดงผล

MIDI Message เข้าคอมพิวเตอร์ผ่านทาง USB และเข้าเครื่องดนตรีผ่านสาย MIDI Cable

โดย MIDI Message ส่วนที่สำคัญ ประกอบด้วย

- Control Change ประกอบด้วย Control, Value และ Channel
- Note ประกอบด้วย Note, Velocity, Channel, Pitch bend

3.2 ขั้นตอนการดำเนินงาน

3.2.1 ศึกษาและทดลองวงจร Theremin

- ศึกษาและค้นคว้าข้อมูลเกี่ยวกับ วงจร Theremin โดยคำนึงถึง Input output ที่เหมาะสมกับบอร์ดที่เลือกใช้ และอุปกรณ์ที่หาได้

3.2.2 ศึกษาการทำงานของ MIDI Controller และ MIDI Protocol ต่างๆ

- ศึกษาเกี่ยวกับการรับส่ง message ของ MIDI
- ศึกษาการทำงานของ MIDI Controller ของคนอื่น

3.2.3 ศึกษาการโปรแกรม Teensy 3.1

- ทำการศึกษาขั้นตอนการทำงานของ Library ของ Board Teensy โดยศึกษาจากโค้ดโปรแกรมตัวอย่างจาก Tutorial กับ Project คนอื่น และ Library ที่เกี่ยวข้องกับการทำงาน

3.2.4 ออกแบบการเชื่อมต่อของเครื่องดนตรีเข้ากับคอมพิวเตอร์และบอร์ด ว่ามี Input และ Output ใดบ้าง

3.2.5 ออกแบบวงจรในส่วนของ ปุ่มกับสวิทช์ และ วงจร Theremin

- นำความรู้ที่ได้จากศึกษามาออกแบบวงจร เป็น Schematic
- ออกแบบการบัดกรีบนบอร์ดไปปลา

3.2.6 ออกแบบการทำงานบนบอร์ด

- ออกแบบ Flowchart การทำงาน

3.2.7 เตรียมอุปกรณ์ที่ต้องใช้งาน และ ติดตั้งโปรแกรม

- ยืมและซื้ออุปกรณ์ที่ต้องการ
- ติดตั้งโปรแกรมที่ใช้

3.2.8 สร้างวงจร Prototype บน Breadboard ทดลองผลและบันทึก

- ต่อวงจรตามที่ได้ออกแบบ
- นำวงจรมาทดลอง โดยทดลองความสัมพันธ์ของ การเล่น Theremin และ output ที่ได้ กับ Oscilloscope

3.2.9 พัฒนาโปรแกรม และ ทดลองผล กับวงจร



รูปที่ 3.2.1 Arduino Website

- พัฒนาโปรแกรมเป็นส่วนๆ และทดลองทีละส่วน

- `getVolt(int pin)` : รับแรงดันที่ได้จากวงจร
- `getNITVolt()` : จับความถี่เริ่มต้นของทุกเสาส (Calibrate)
- `CCProcess(int pin,double diffVolt)` : แปลงค่าสำหรับ CC
- `NoteProcess(int pin,double diffFreq)` : แปลงค่าสำหรับ Note
- ส่วน main loop : ตรวจสอบสวิช ปุ่มกด และ คำนวณช่วงต่างของความถี่จับได้กับความถี่แรกเริ่ม และ Quantization

- ทดลองกับวงจรที่สร้างขึ้นมาก และ บันทึกผล

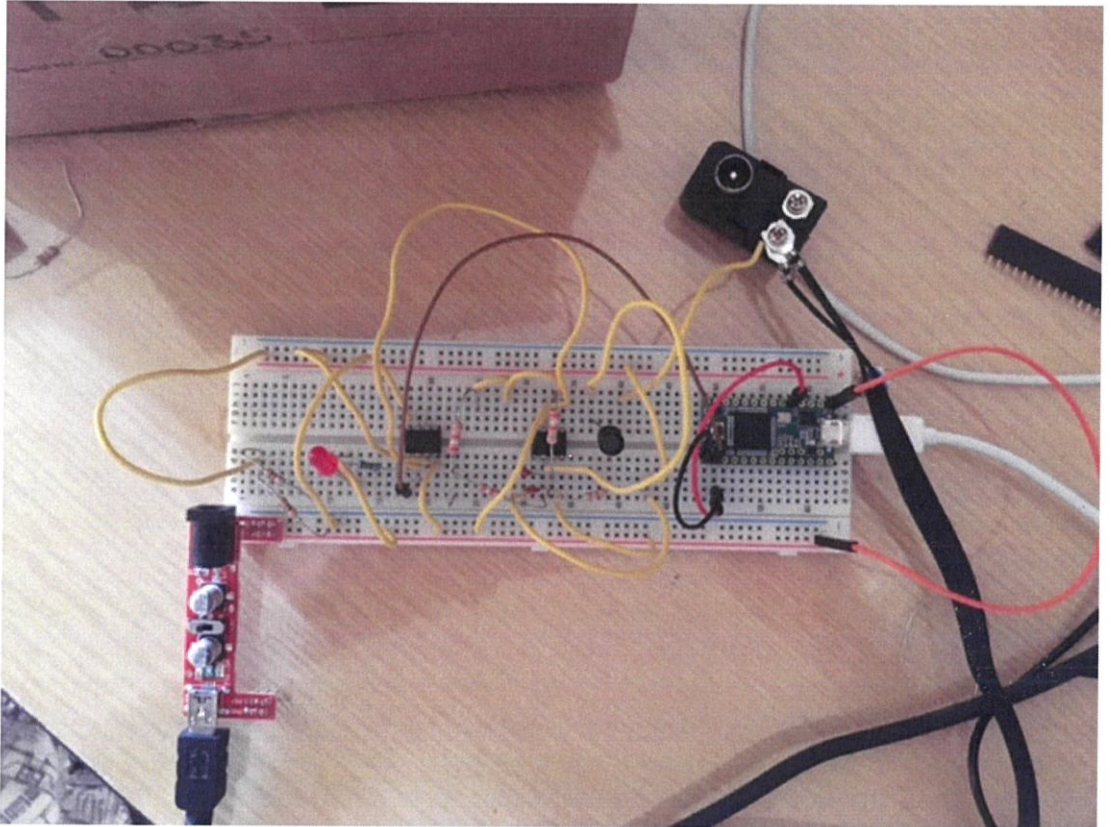
3.2.10 บัดกรีวงจรลงบอร์ดแล้ว ประกอบเป็น Package

บทที่ 4

การทดลองและผลลัพธ์

4.1 วงจรกำเนิด Square Wave

ใช้ LM358N 1 ตัว ต่อ 2 Oscillator

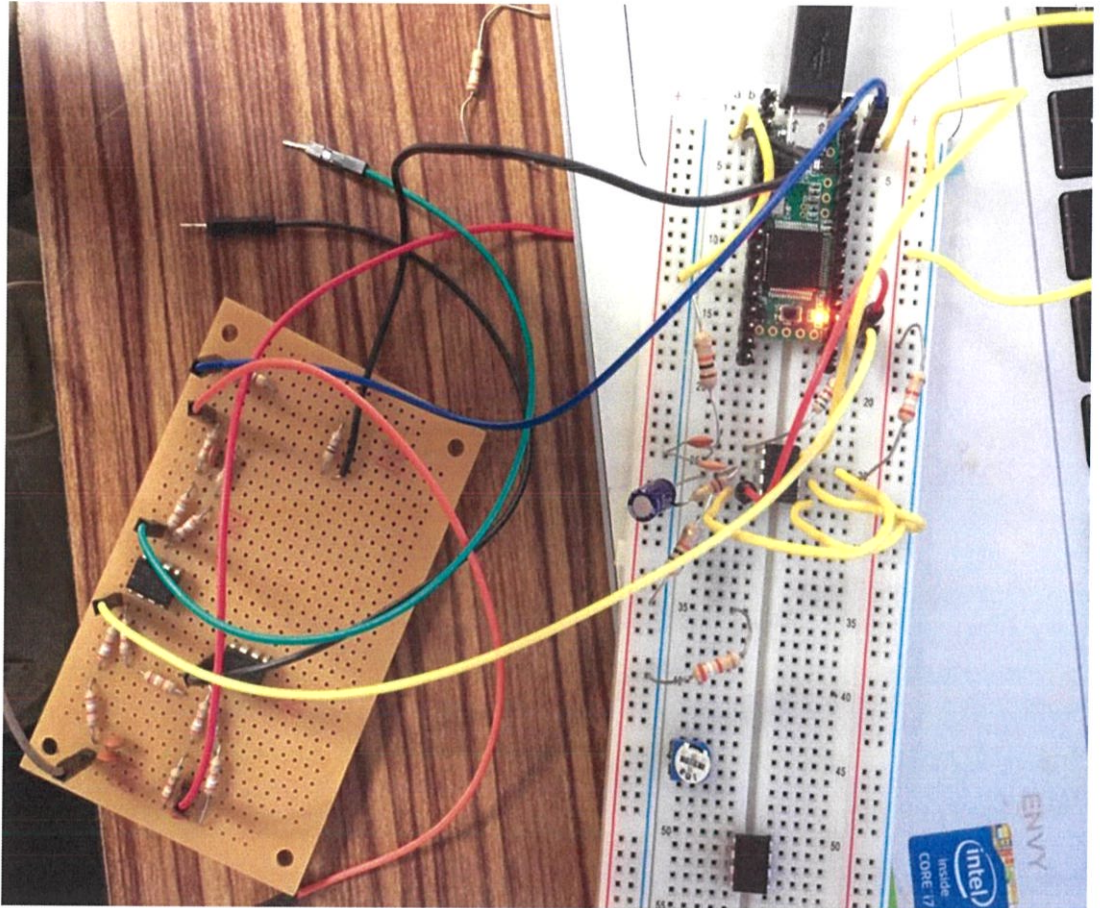


รูป 4.1.1 วงจรกำเนิด Square Wave

ได้ความถี่เริ่มต้นประมาณ 100000Hz ถึง 50000 Hz ขึ้นอยู่กับสภาพแวดล้อม และ เสอาอากาศ
เมื่อเอามือไปเล่นกับเสอาอากาศ ความถี่ลดลงต่ำสุดอยู่ในช่วง 25000-60000 Hz ตามความถี่เริ่มต้น

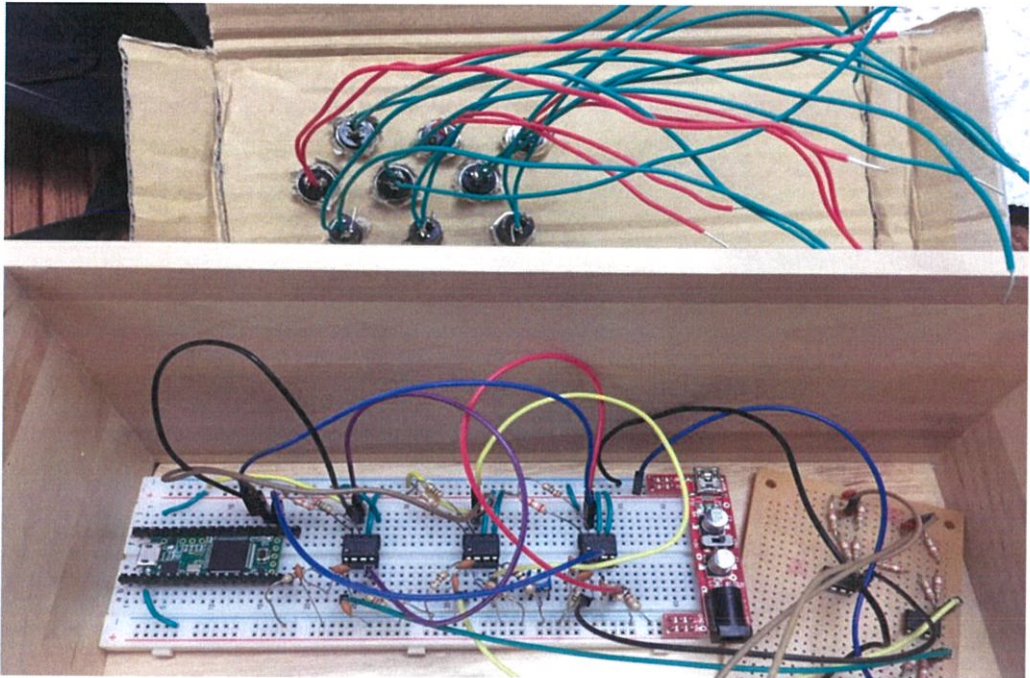
4.2 วงจร Frequency to Voltage

ถ้าหากใช้ความถี่เริ่มต้นประมาณ 55000 Hz จะได้ค่าต่ำสุดประมาณ 30000Hz จะแปลงได้เป็นประมาณ 2V-5V



รูป 4.1.2 วงจร Frequency to Voltage

4.3 Teensy Board



รูป 4.1.3 Teensy Board

ADC ที่ (0V => 0 5V => 1023)

4.3.1 พัฒนาและทดลอง function อ่านค่า Voltage บนบอร์ด และ function กำหนดค่าเริ่มต้นสำหรับแต่ละ Theremin ได้ Freqout และ INITfreqpoint มีค่าเริ่มต้นประมาณ 0.4V-2V และสูงสุดที่ 5V return ได้ 82 -1023

ค่าเริ่มต้น ขึ้นอยู่กับสภาพแวดล้อม และ เสอาอากาศ

สภาพแวดล้อม	เสอาอากาศ	ผลลัพธ์ (0-1023)
แคบ	เคสเครื่องดนตรีที่เป็นเหล็ก	232-357
แคบ	แผ่น foil ขนาด 28.5x17.5cm	13-223
โล่ง	แผ่น foil ขนาด 28.5x17.5cm	472-1023

ตาราง 4.1.4 ตารางที่ค่าเปลี่ยนแปลง

4.3.2 ทดลอง Quantize ค่าโดยใช้

$$4.3.2.1 \text{ diffFreq} = (x / y) * (\text{Freqout}[\text{pin}] - \text{INITfreqpoint}[\text{pin}])$$

โดย x/y เป็นเศษส่วน ในการลดค่าแบบ linear

$$X = 14 \quad y = 16$$

ค่าที่ได้ดีขึ้น แต่ยังมี การเปลี่ยนแปลงต่อระยะห่างสูงเมื่อ มีอยู่ใกล้ Antenna

$$4.3.2.2 \text{ diffFreq} = (x / y) * (\text{Freqout}[\text{pin}] - \text{INITfreqpoint}[\text{pin}]) - (\log_z((x / y) * \text{Freqout}[\text{pin}] - \text{INITfreqpoint}[\text{pin}]) - d) ^ p$$

$$\text{โดยใช้ } x = 15 \quad y = 16 \quad z=3 \quad p=3 \quad d=3$$

ผลออกมาเป็นกราฟตัวโน้ตต่อ linear มากขึ้นมาก

Code แบบย่อ: `diffFreq =(15.0/16.0) * (Freqout[pin]-INITfreqpoint[pin]);`

`diffFreq -= pow((log(diffFreq)/log(3)) - 3.0 , 3.0);`

$$4.3.1.3 \text{ diffFreq} = (x / y) * (\log(\text{Freqout}[\text{pin}] - \text{INITfreqpoint}[\text{pin}]) / \log z)$$

ผลออกมาเป็นกราฟตัวโน้ตต่อ linear มากขึ้น

4.3.3 ทดลองและพัฒนา MIDI CC Process function

แปลงค่า ให้อยู่ใน Range ที่เหมาะสมสำหรับ MIDI CC Value (diffMax คือค่าต่างกันมากที่สุดของ voltage)

$$\text{value} = (\text{diffFreq} - 50) * (128.00/\text{diffMax})$$

- diffMax=400

ได้ค่าละเอียดเมื่อเทียบกับมือค่าเปลี่ยนแปลงอย่างเสถียร แต่เมื่อเปลี่ยนสภาพแวดล้อม บางครั้งไม่สามารถใช้ได้ เพราะ value ที่ใช้ในการส่งไม่พอ (ไม่ถึง 127)

- diffMax = 300

ค่าที่ได้ละเอียดน้อยลง และ เกิน 127 แต่ได้จำกัดไว้ที่ 127 หากเกิน จึงไม่มีปัญหา

-ทดลองค่า k เพื่อกำหนด Gap ความต่าง ที่ทำให้มีการส่ง MIDI CC Message

$$\text{if}(\text{pren}[\text{pin}]-\text{value} \geq k \parallel \text{pren}[\text{pin}]-\text{value} \leq -k)$$

`usbMIDI.sendControlChange(controller+pin, n0, channel);`

ค่า k	CC MIDI Message Value
0	ไม่คงที่ (+-2)
1	คงที่ขึ้น (+-2)
2	คงที่มาก
3	คงที่มาก แต่ดูไม่ต่อเนื่อง

ตาราง 4.1.5 ตารางค่า K

ระยะห่าง (cm)	ค่า CC (0-127)
12	0
11	0
10	1-3
9	10-12
8	22-24
7	40-42
6	60-63
5	70-72
4	81-83
3	92-97
2	111-114
1	123-127
0.5	127

ตาราง 4.1.6 ตารางเปรียบเทียบระยะห่างระหว่างมือกับ Antenna กับค่าที่ใช้ในการส่ง CC Value

4.3.4 ทดลองและพัฒนา MIDI Note Process function

-ทดลองหาค่า Threshold ที่ไม่ทำให้สภาพแวดล้อม มีผลได้โดยง่าย

- ค่า diffFreq > 30

มีโอกาศที่ค่าจะเกิน 30 บางครั้ง เมื่อสภาพแวดล้อมมีผล ทำให้เครื่องดนตรีเล่นไปเอง

- ค่า diffFreq > 50

เป็นค่าที่ใช้ได้เหมาะสม

- แปลงค่า ให้อยู่ใน Range ที่เหมาะสมสำหรับ MIDI Pitch Bend

$$n0 = ((diffFreq - 50.00) * (16384.00 / 300.00));$$

-ทดลองค่า k เพื่อกำหนด Gap ความต่าง ที่ทำให้มีการส่ง MIDI Message

Code แบบย่อ : `if(!(pren[pin]-n0>=Gap || pren[pin]-n0<=-Gap))`

`firstnote[pin]=36+(12*octave[pin]);`

`usbMIDI.sendNoteOn(firstnote[pin], 110, pin+1);`

`usbMIDI.sendPitchBend(n0, pin+1);`

- ตั้ง Gap ไว้ที่ 50

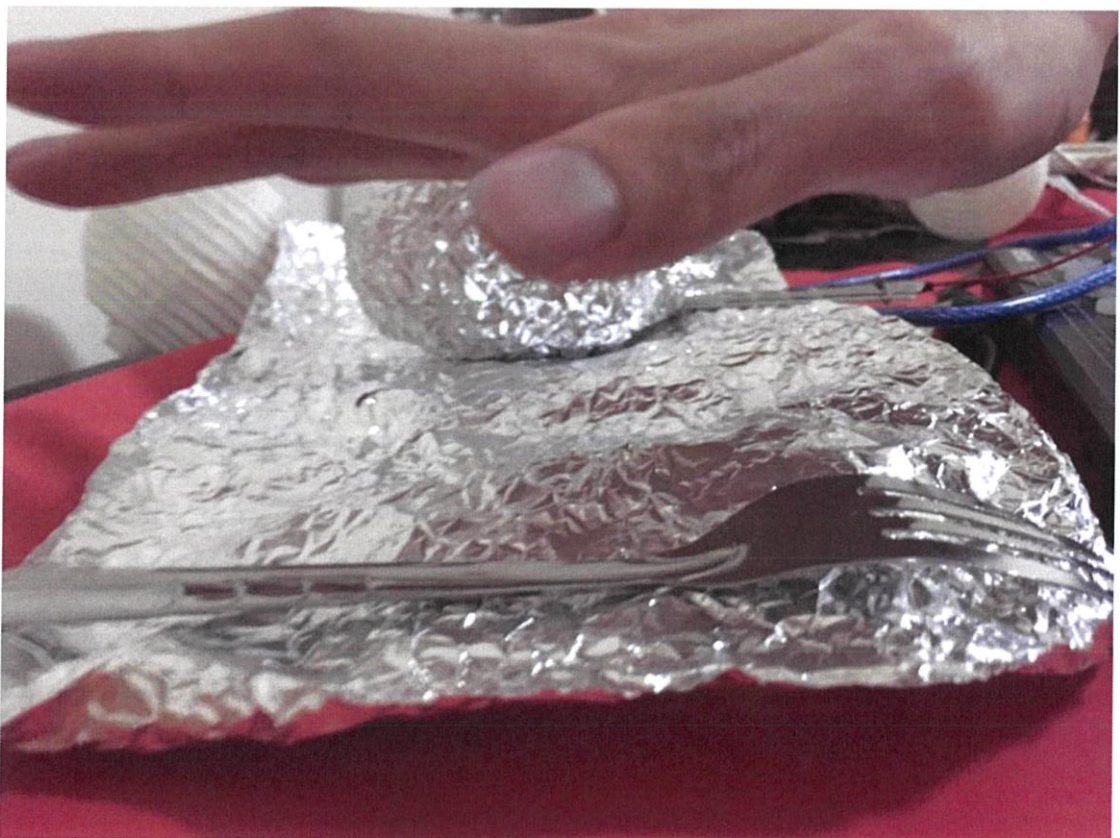
ผล ไม่เสถียรมากพอ ค่า Pitch bend เปลี่ยนแปลงเมื่อไม่มีการขยับของมือ

- ตั้ง Gap ไว้ที่ 100

ผลเสถียรมากพอ ค่า Pitch bend เปลี่ยนแปลงน้อยลงเมื่อไม่มีการขยับของมือแต่ทำให้มีความใกล้เคียง Analog น้อยลง

ระยะห่าง (cm)	ค่า Pitch bend (0-16383)
12	0-120
11	100-300
10	1240-1450
9	2100-2350
8	3410-3600
7	4650-5260
6	6450-7660
5	8510-8720
4	10000-10300
3	11700-12000
2	13500-13800
1	15700-16100
0.5	16383

ตาราง 4.1.7 ตารางเปรียบเทียบระยะห่างระหว่างมือกับ Antenna กับค่าที่ใช้ในการส่ง Pitch bend

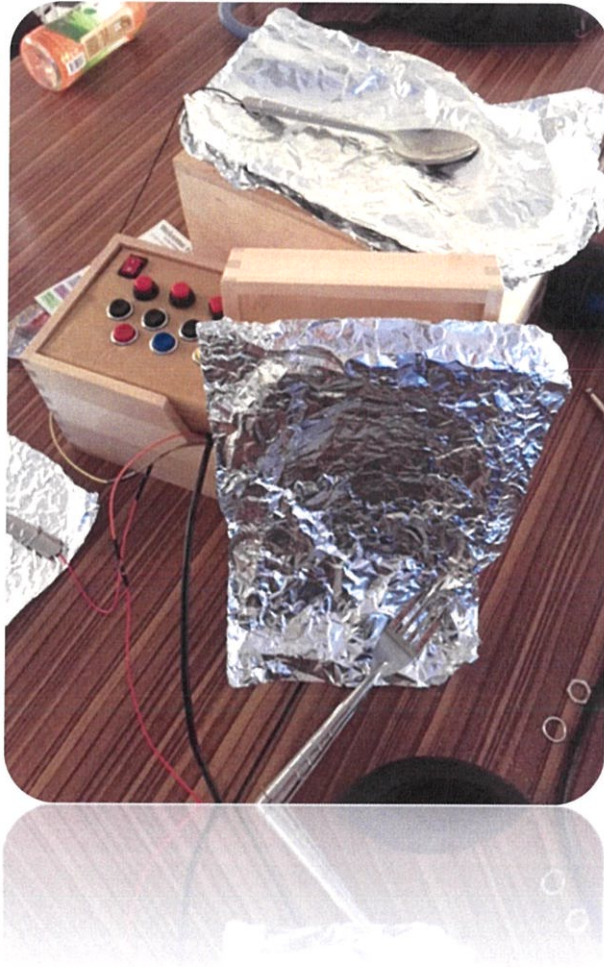


รูป 4.1.8 ทดลองเล่น MIDI Controller

บทที่ 5

สรุปผลการทดลอง

5.1 สิ่งที่ได้



รูป 5.1.1 MIDI Theremin

5.1.1 ได้ MIDI Controller ที่สามารถควบคุม MIDI ในโปรแกรมที่รองรับได้

5.1.2 การเล่นได้ความรู้สึกใกล้เคียง Real-time และ เป็น Pitch ที่ต่อเนื่อง

5.1.3 มี feature ในการเปลี่ยนแปลง octave และ การเปลี่ยน Mode

5.2 ปัญหาอุปสรรคแนวทางแก้ไข

5.2.1 ความเร็วของ Teensy Board ไม่เพียงพอในการทำทุก Process เมื่อใช้ Board ในการตรวจความถี่

5.2.1.1 เปลี่ยนแปลง Algorithm เพิ่ม priority ให้การตรวจความถี่

5.2.1.2 เพิ่มเต็มวงจร เปลี่ยนความถี่เป็นความต่างศักย์ แล้วใช้ค่าความต่างศักย์ แทน

5.2.2 Voltage output ที่ได้จากวงจรไม่เสถียร

5.2.2.1 เพิ่ม Gap ที่เป็น Threshold จำกัดการเปลี่ยนแปลงของ MIDI Messages

5.2.2.2 เฉลี่ยค่า Voltage input ที่ได้ ด้วยการ Sampling โดยต้องคำนึงถึง ความเป็น Real-time ของการเล่นด้วย

5.2.3 กราฟระหว่าง Voltage input กับ ค่า Pitch bend เป็น exponential ทำให้การเล่น MIDI Controller นี๊ยาก

5.2.3.1 ทดลองและบันทึกค่า แล้วออกแบบสมการคณิตศาสตร์ สำหรับการแปลง สมการ Exponential ให้ใกล้เคียง Linear

5.3 แนวทางการพัฒนาต่อ

5.3.1 ผลลัพธ์ที่ได้จากวงจรมีค่าไม่เสถียรพอ

5.3.1.1 แก้ด้วยการออกแบบวงจรใหม่ หรือ เพิ่มเต็มวงจร

- เพิ่มเต็ม Inductor

5.3.1.2 ลดจำนวน Antenna แล้วแก้ไขโปรแกรมให้มีการเฉลี่ยของ Voltage input มากขึ้น

5.3.1.3 เปลี่ยน Antenna เป็นอย่างอื่น

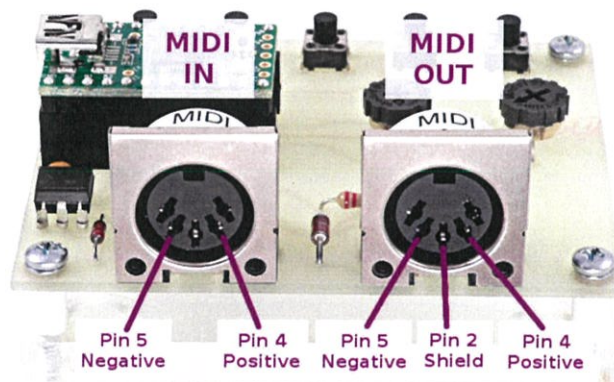
5.3.1.4 แก้โปรแกรมให้ Gap ที่เป็น Threshold ในการ จำกัดการเปลี่ยนแปลง MIDI กว้างขึ้น แต่จะทำให้ค่าดูต่อน้อยลงเมื่อมีการเปลี่ยนแปลง

5.3.2 เพิ่มเต็มส่วน MIDI I/O สำหรับเชื่อมต่อกับเครื่องดนตรีอื่น

5.3.2.1 สร้างวงจรเพิ่มเติม ที่มี MIDI Female 5pin connector เป็น port เชื่อมต่อกับ MIDI Cable

5.3.2.2 เขียนโปรแกรมโดยใช้ Library ของ Teensy ชื่อ MIDI

https://www.pjrc.com/teensy/td_libs_MIDI.html

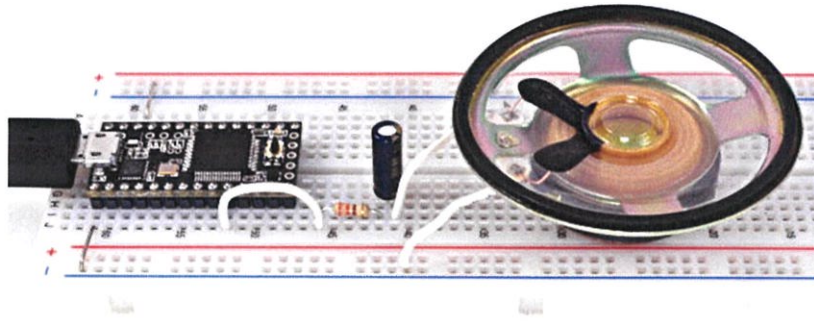


รูป 5.3.2 ตัวอย่างที่การใช้ใน MIDI Library นี้

5.3.3 สร้างให้เป็นเครื่องดนตรีที่กำเนิดเสียงในตัวหรือ Synthesize คลื่นขึ้นมา

5.3.3.1 เพิ่มเต็มให้ Board generate เคลื่อนออกมา

https://www.pjrc.com/teensy/td_pulse.html



รูป 5.3.3 ตัวอย่างการต่อลำโพง

5.3.4 พัฒนาให้ใช้ได้ดีขึ้นเมื่อเปลี่ยนสภาพแวดล้อม

5.3.4.1 คิดการคำนวณ เพื่อแก้ไขค่าคงที่ที่ใช้ในการคำนวณ Note และ MIDI โดยนำค่าเริ่มต้นมาคำนวณ(INITVoltpoint)

บรรณานุกรม

Arduino

Available : www.arduino.cc

PJRC

Available : <http://pjrc.com/teensy/>

Computer Science Department New Mexico State University

Available : <http://www.cs.nmsu.edu/~rth/EMTheremin.pdf>

Visionlearning

Available : <http://www.visionlearning.com/en/library/Math-in-Science/62/Exponential-Equations-in-Science-I/206>

MIDI

Available : http://www.midi.org/aboutmidi/tut_protocol.ph