Floppy Drive Orchestra

University of Colorado Boulder

Independent Study

Jeffery Lim

Jeffery.Lim@colorado.edu

Under supervision of Dr. Shalom Ruben

Contents

Chapter 1

Introduction

Chapter 2

Floppy Drive Characteristics

In order to properly proceed to develop a floppy drive orchestra, a single floppy drive must be understood in order to understand the capabilities and restrictions that the floppy drives presents.

2.1 Floppy Pinout

The floppy pinout is shown in Figure 3.2. The bottom row, or the odd valued pins, are all grounded, where the top pins, or the even valued pins, are live. The actual names of the active pins are shown in Figure 3.3

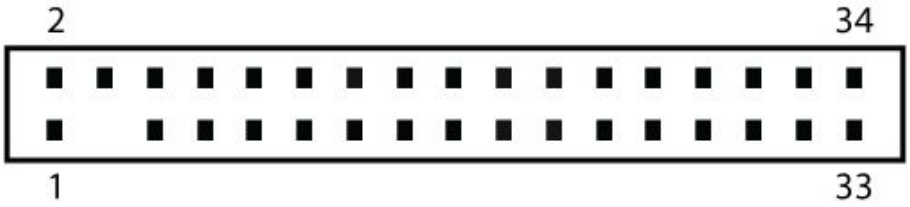

Figure 2.1: Floppy Drive Pinout

Pin	Name	Dir	Description
2	/REDWC	→	Density Select
4	n/c		Reserved
6	n/c		Reserved
8	/INDEX	←	Index
10	/MOTEA	→	Motor Enable A
12	/DRVSB	→	Drive Sel B
14	/DRVSA	→	Drive Sel A
16	/MOTEB	→	Motor Enable B
18	/DIR	→	Direction
20	/STEP	→	Step
22	/WDATE	→	Write Data
24	/WGATE	→	Floppy Write Enable
26	/TRK00	←	Track 0
28	/WPT	←	Write Protect
30	/RDATA	←	Read Data
32	/SIDE1	→	Head Select
34	/DSKCHG	←	Disk Change/Ready

Figure 2.2: Floppy Drive Pin Names

The only necessary pins are pin number 12 or 14, 18, and 20. Pins 12 and 14 are the drive select B and A. These are the drive enable pins. Some drives are B drives and others are A, so in order to test them, either one of them will need to be sleected.

Pin 18 is a direction pin. The direction is what determins the direction of the motor drive. When the pin is grounded, the drive heads away from the pins, and when the pin is high, the drive returns.

Pin 20 is a step pin. This pin drives the stepper motor, so every time the pin goes high, the stepper motor will go forward one tick.

2.2 Stepper Motor Movement

The floppy drive motor head has a physical limit to how far it can go. To determine this value, a 1 Hz square wave is sent through to the floppy drive's direction pin (pin 18). The number of ticks is manually recorded, and the step pin is inverted, to count back.

The total

2.3 Stepper Motor Bandwidth

2.4 Power

Chapter 3

Wiring Diagram

3.1 Power

The floppy drive takes a mini Molex cable as seen in Figure 3.1. A mini Molex cable has a 5V line as well as 12V. Modern floppy drives used to use the 12V line for the drive motor, however, more modern floppy drives only use the 5V to drive the entire drive.

The power consumption of the drive when idling is an average of 50 mA and when the motor is active, it pulls 400 mA.

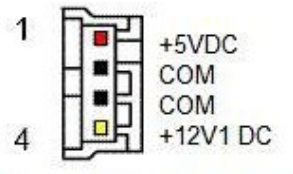

Figure 3.1: Floppy Drive Power Pinout

For each floppy drive added to the system, the total power draw increases by 400 mA, so for 4 floppy drives, the total power consumption from the floppy drive system would be around 1600 mA, or 1.6 A.

3.2 Arduino Wiring

Chapter 4

MIDI Files

MIDI, or Musical Instrument Digital Interface, is a standard that has its own protocols, interface, and connectors. It allows a single file to contain multiple tracks for several instruments in its own channel. This allows a MIDI file to play several instruments at once, up to a maximum of 16 instruments. This advantage of the MIDI file allows multiple floppy drives to be played at once, like how a MIDI file would send data to several instruments.

4.1 MIDI Notes

The MIDI interface has notes that are mapped to specific piano keys at their respected frequencies.

Note	Octave										
	-1	0	1	2	3	4	5	6	7	8	9
C	0	12	24	36	48	**60**	72	84	96	108	120
C#	1	13	25	37	49	**61**	73	85	97	109	121
D	2	14	26	38	50	**62**	74	86	98	110	122
D#	3	15	27	39	51	**63**	75	87	99	111	123
E	4	16	28	40	52	**64**	76	88	100	112	124
F	5	17	29	41	53	**65**	77	89	101	113	125
F#	6	18	30	42	54	**66**	78	90	102	114	126
G	7	19	31	43	55	**67**	79	91	103	115	127
G#	8	20	32	44	56	**68**	80	92	104	116	
A	9	21	33	45	57	**69**	81	93	105	117	
A#	10	22	34	46	58	**70**	82	94	106	118	
B	11	23	35	47	59	**71**	83	95	107	119	

Figure 4.1: MIDI Note Number to Key

4.2 MIDI Messages

MIDI Message				
Status			Data	Data
Command (4 bits)		Channel(4 bits)	Note (8 bits)	Velocity (8 bits)
Note On	1001	nnnn	0xxxxxxx	0vvvvvvv
Note Off	1000	nnnn	0xxxxxxx	0vvvvvvv

Chapter 5

Software

5.1 MIDI Player

5.2 MIDI to Serial Driver

5.3 Arduino

Chapter 6

Arduino

```
1
2
3  //Freq = 1/(RESOLUTION * Tick Count)
4  static int noteLUT[127];
5
6  void setup()  {
7
8    //Init parameters
9    int i,j;
```