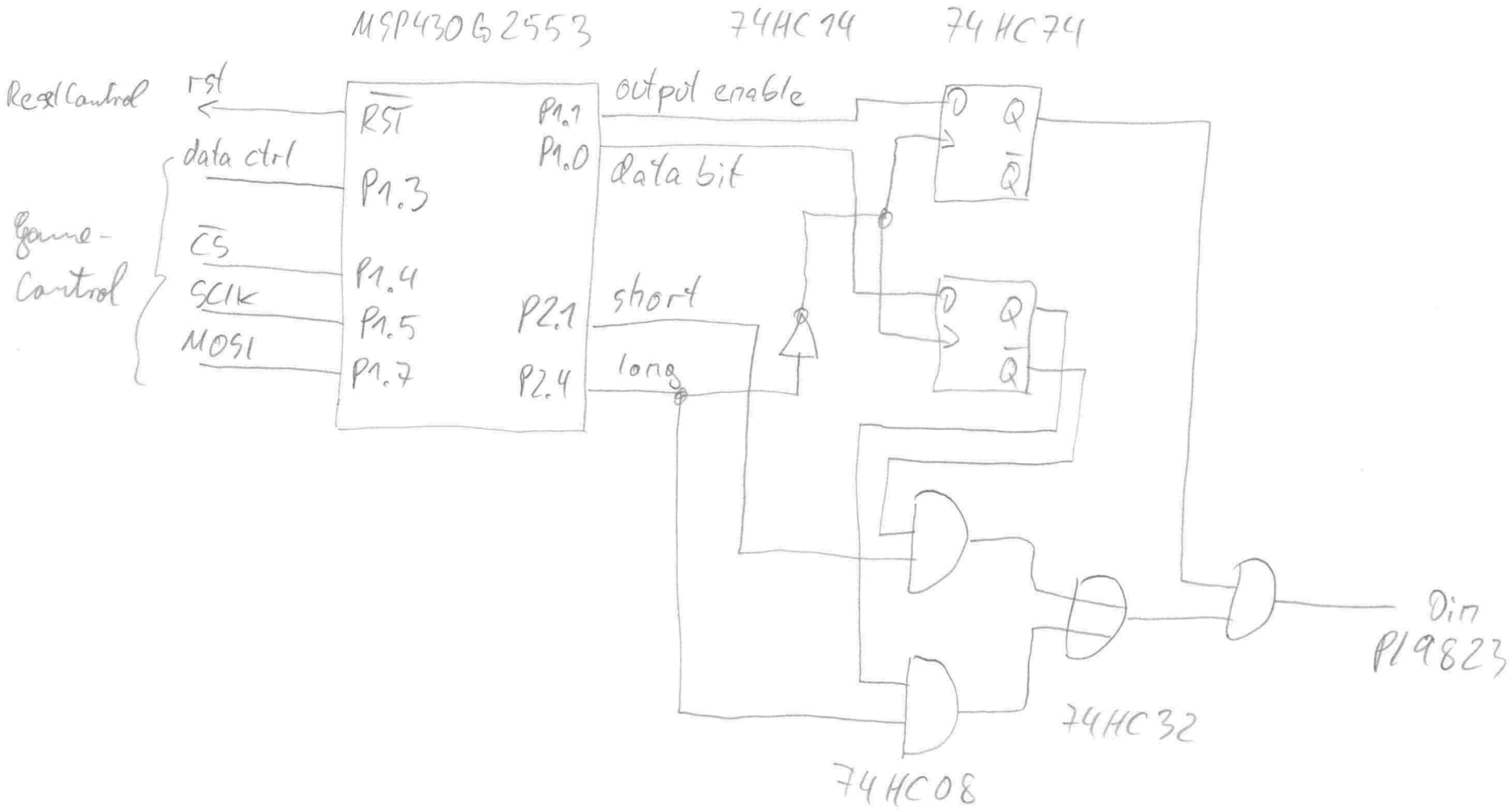
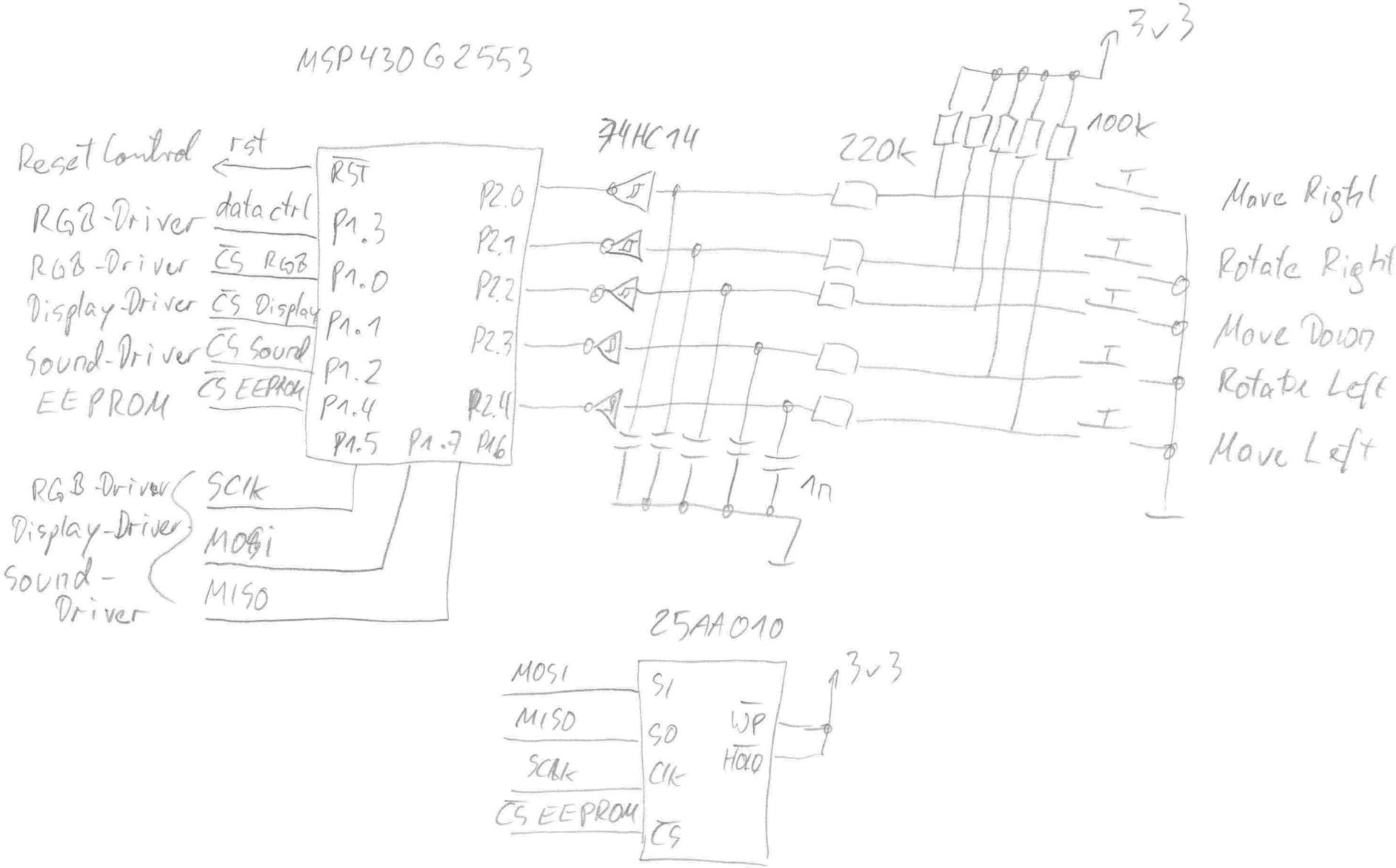
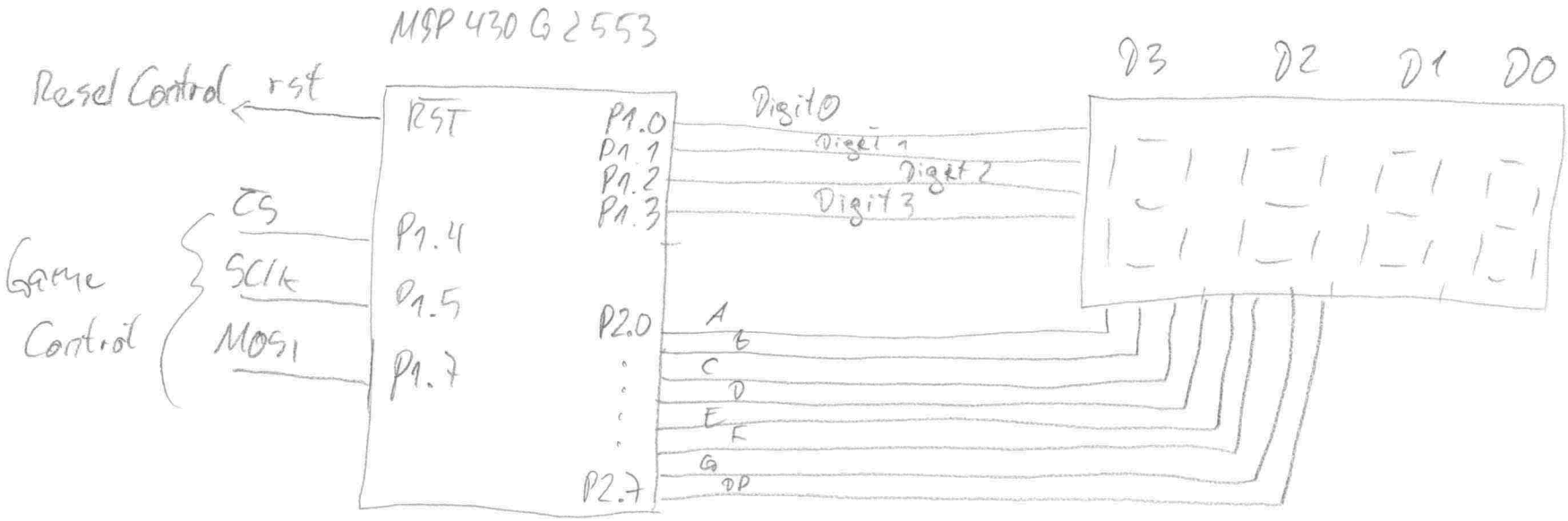
RGB-Driver


MSP430G2553
74HC14
74HC74
Reset Control
rst
RST
P1.7
output enable
data ctrl
P1.3
P1.0
data bit
Game-Control
CS
P1.4
SCLK
P1.5
short
P2.1
MOSI
P1.7
P2.4
long
D
Q
Q̄
Din
P19823
74HC32
74HC08

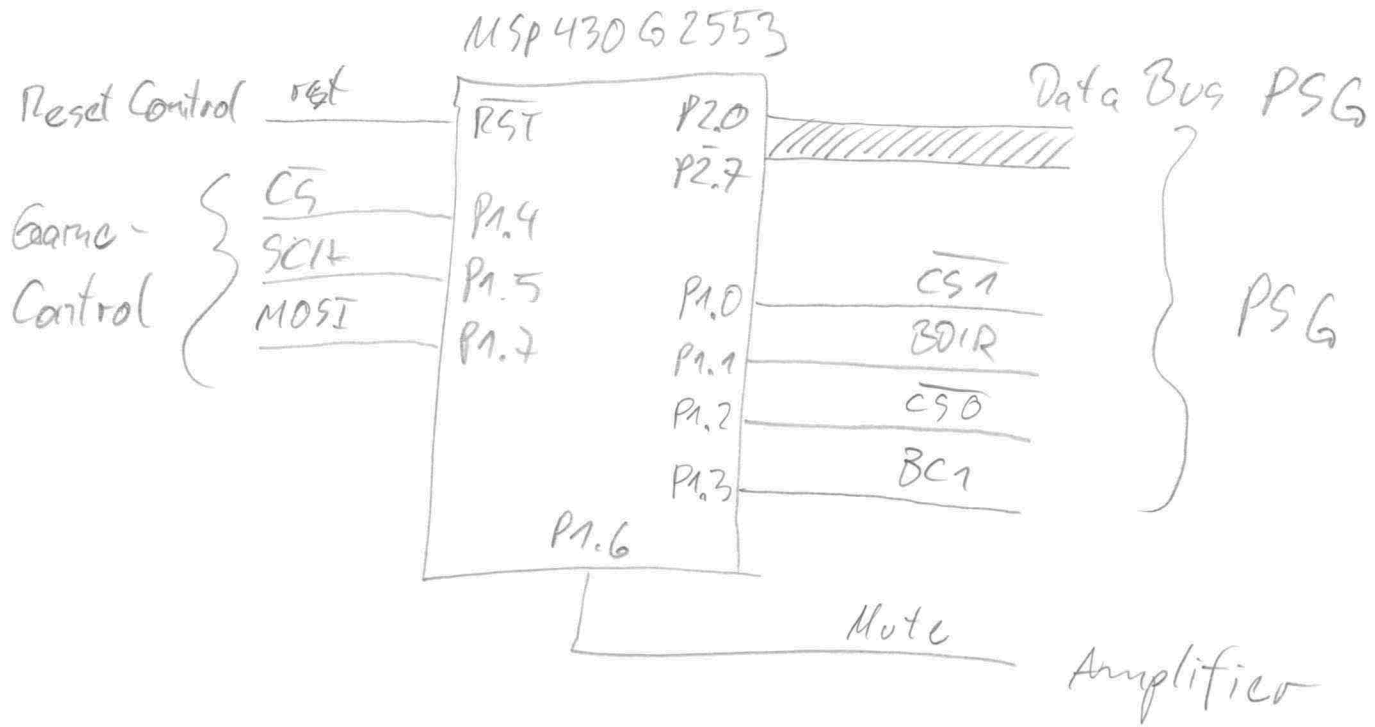
Game-Control



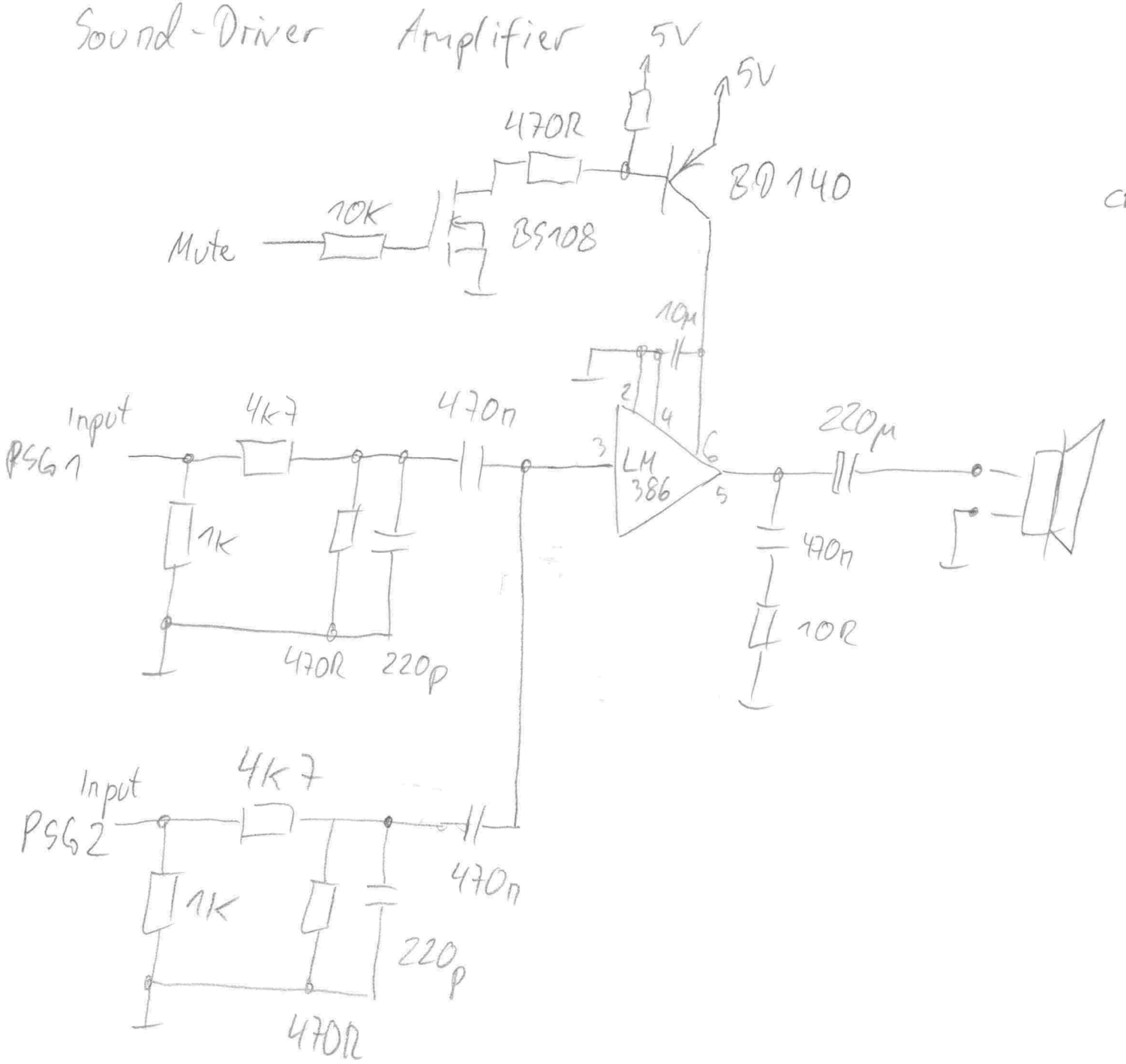
Display-Driver



Sound-Driver

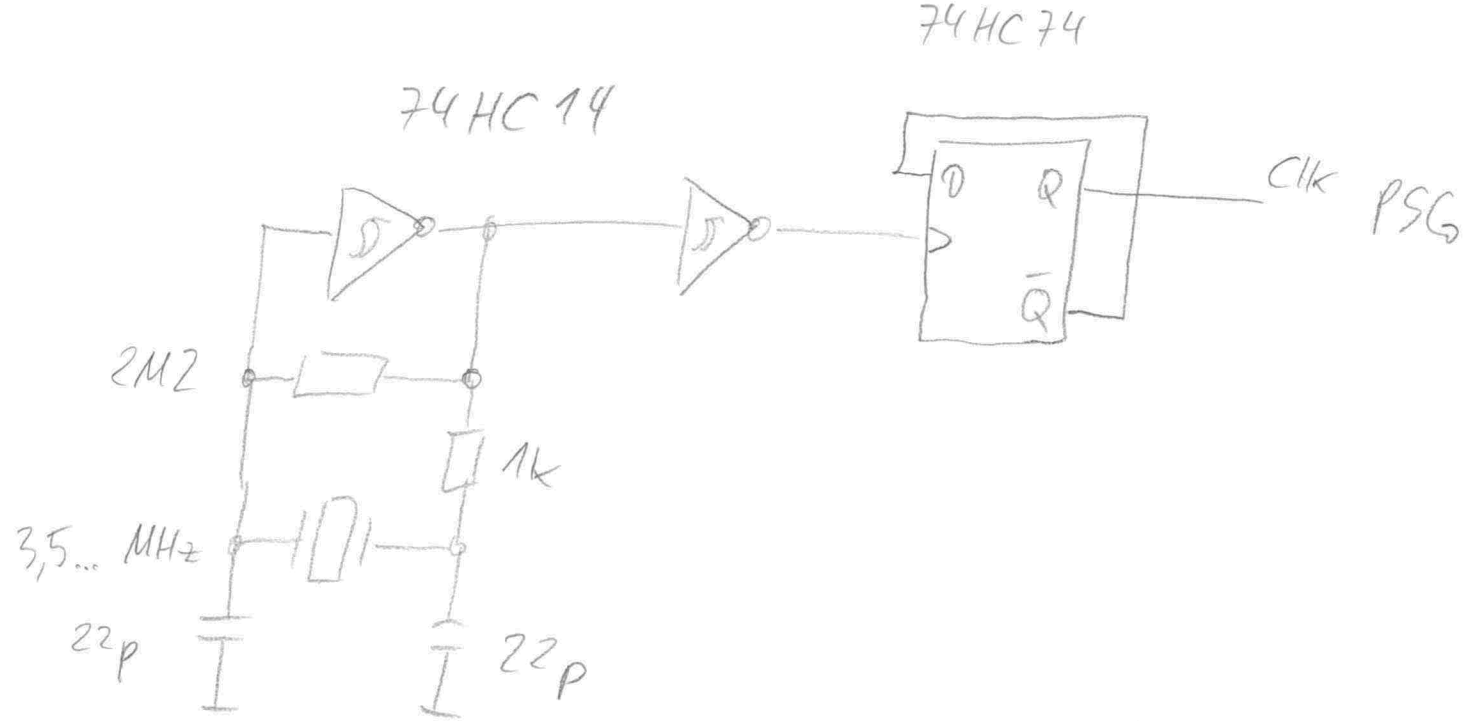


Sound-Driver Amplifier



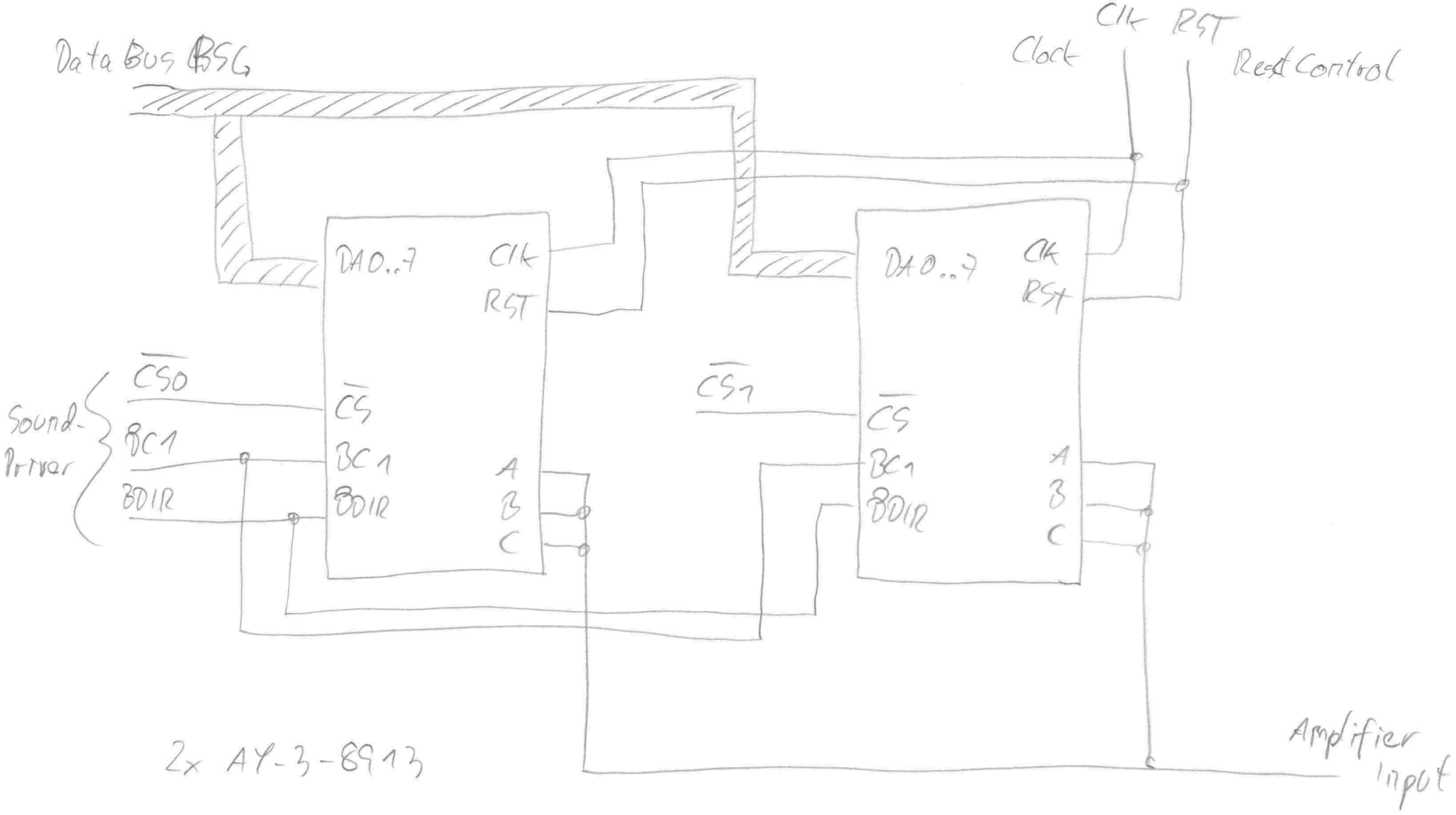
cap. datasheet
AY-3-8910, cp. 4.3

Sound-Driver Clock

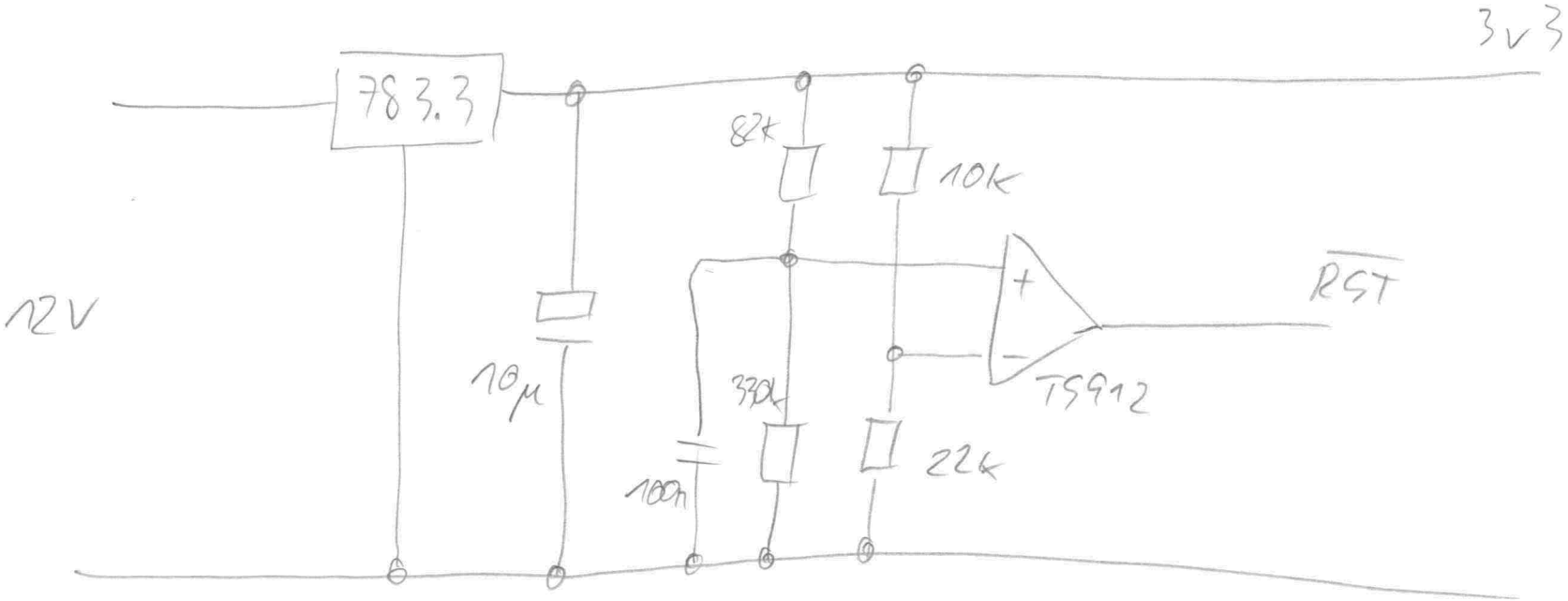


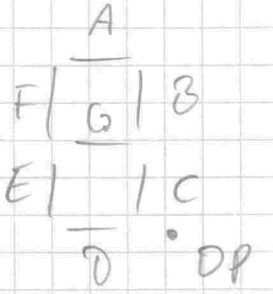
cmp. "The Art of Electronics"
3. Edition
p. 447, fig 7.38 D

Sound-Driver PSG



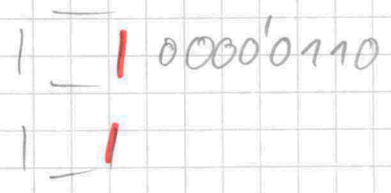
Reset Control

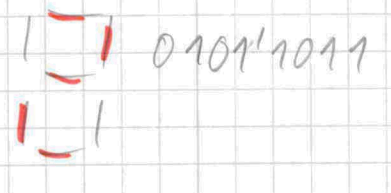


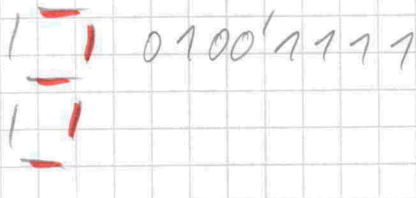


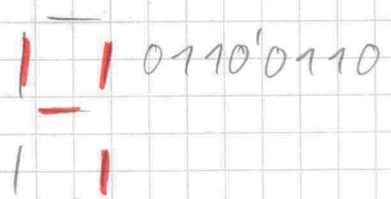
A: P2.0
B: P2.1
C: P2.2
D: P2.3
E: P2.4
F: P2.5
G: P2.6
DP: P2.7

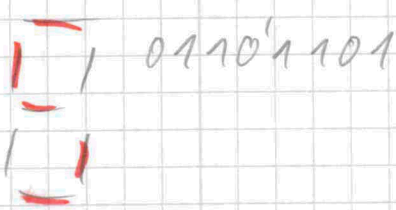
1er P1.0
10er P1.1
100er P1.2
1000er P1.3

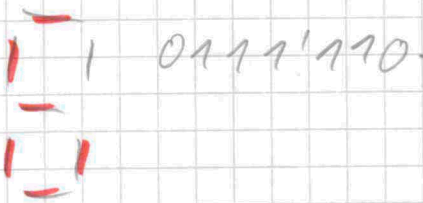


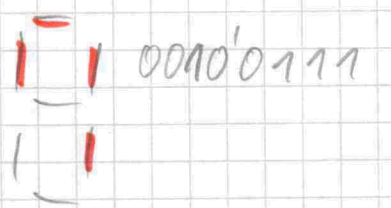


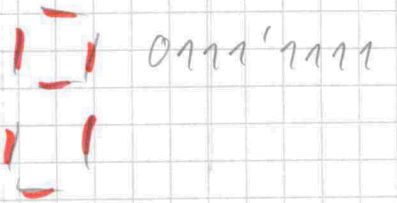


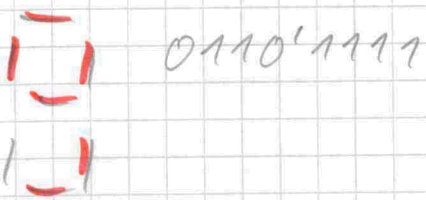


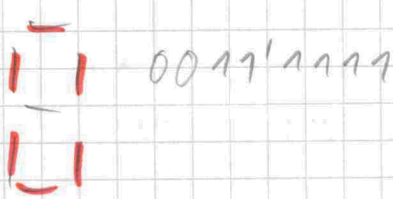


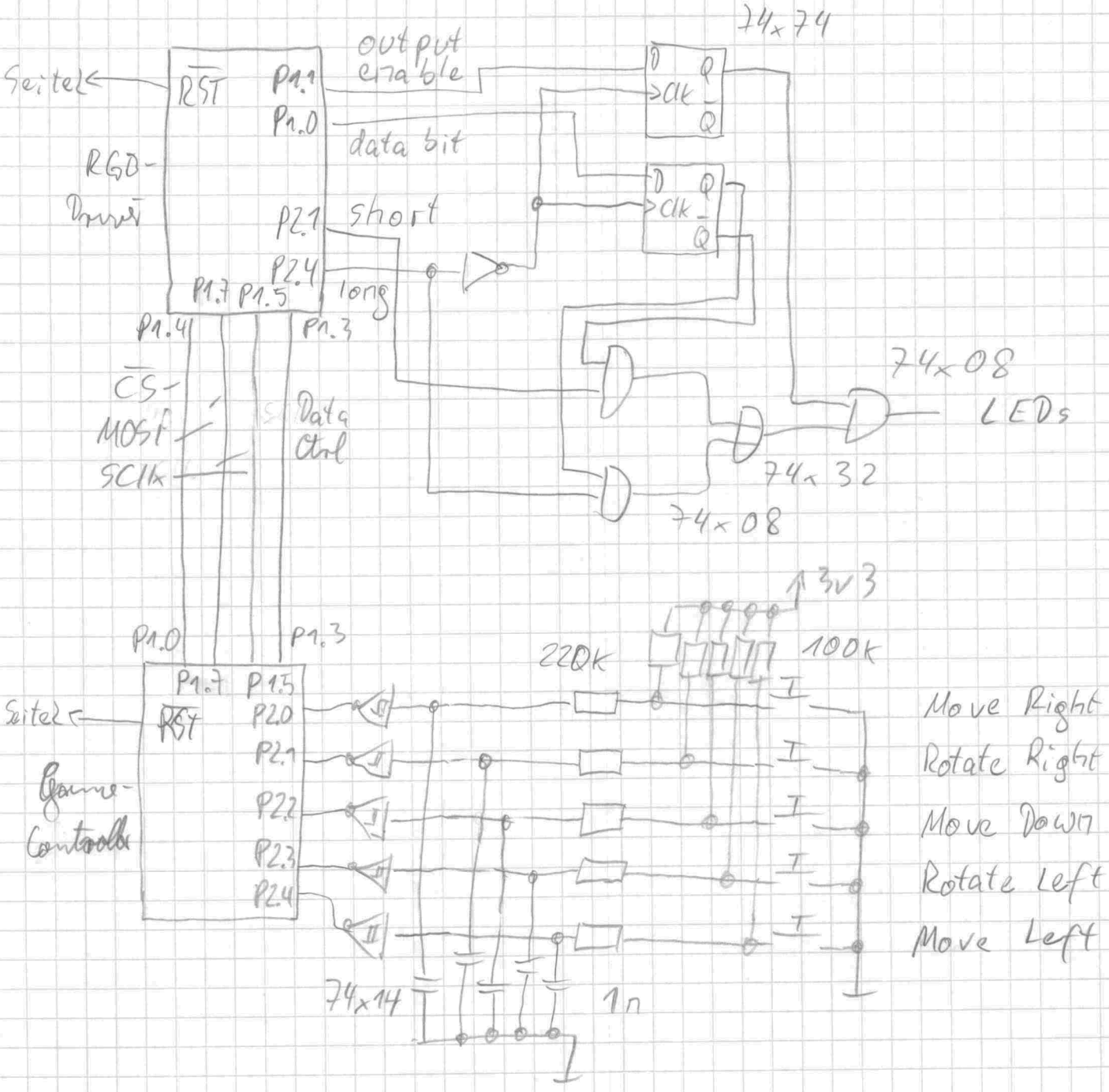


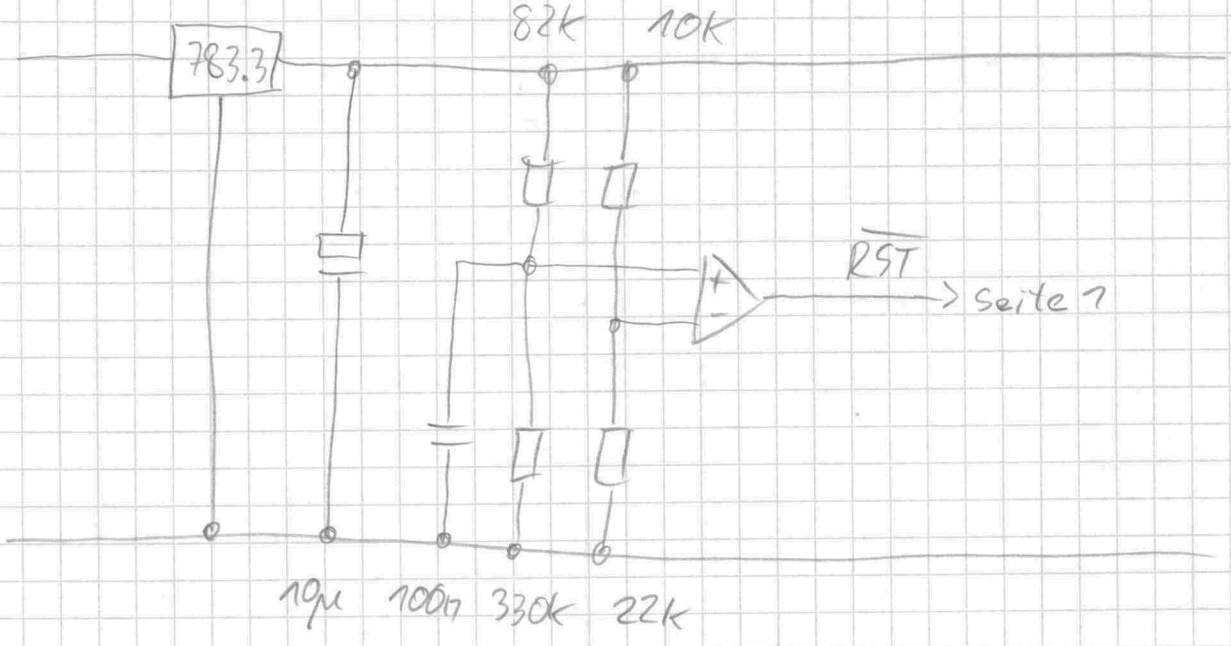


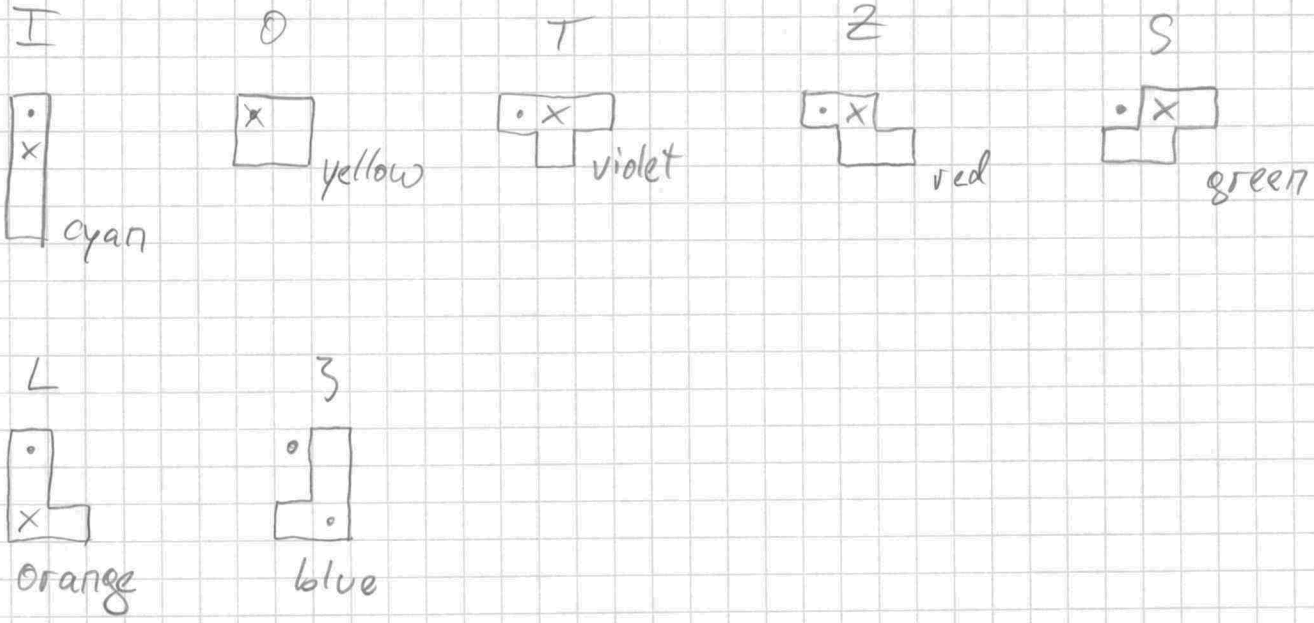






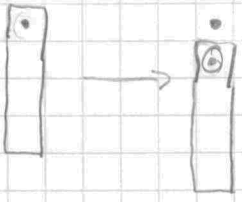

74x74
output enable
data bit
short
long
Seitel
RST
P1.1
P1.0
RGD-
P2.1
P2.4
P1.7
P1.5
P1.4
P1.3
CS-
MOSI
SCLK
Data Ctrl
74x08
LEDs
74x32
74x08
3v3
220k
100k
P1.0
P1.3
Seitel
RST
P2.0
P2.1
P2.2
P2.3
P2.4
Game-Controller
Move Right
Rotate Right
Move Down
Rotate Left
Move Left
74x14
1n


82k
10k
783.3
RST
Seite 7
10µ
100n
330k
22k


I
O
T
Z
S
cyan
yellow
violet
red
green
L
J
orange
blue

draw: 0,0 ; 0,1 ; 0,2 ; 0,3

· Origin
⊙ new Origin



move down, 0° + 180°

set: 0,4 offset: 0,1

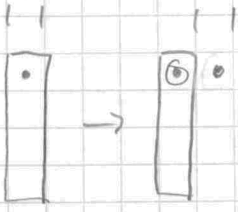
reset: 0,0

move down, 90° + 270°

set: 0,1 ; 1,1 ; 2,1 ; 3,1

reset: 0,0 ; 1,0 ; 2,0 ; 3,0

offset: 0,1



move left 0° + 180°

set: -1,0 ; -1,1 ; -1,2 ; -1,3

reset: 0,0 ; 0,1 ; 0,2 ; 0,3

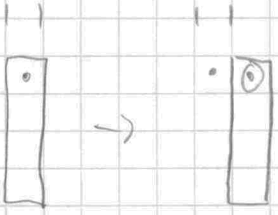
offset: -1,0

move left 90° + 270°

set: -1,0

reset: 3,0

offset: -1,0



move right, 0° + 180°

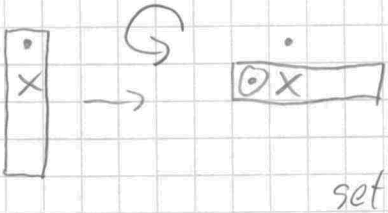
set: 1,0 ; 1,1 ; 1,2 ; 1,3
reset: 0,0 ; 0,1 ; 0,2 ; 0,3
offset: 1,0

move right, 90° + 270°

set: 4,0
reset: 0,0
offset: 1,0

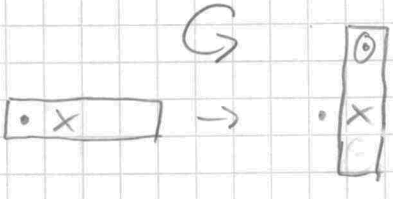
---

x Pivot



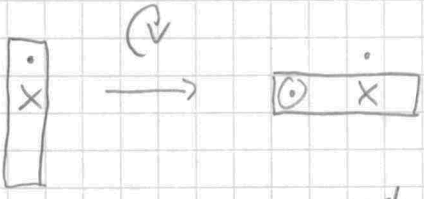
rotate left, 0° + 180°

set: -1,1 ; 1,1 ; 2,1
reset: 0,0 ; 0,2 ; 0,3
offset: -1,1



rotate left, 90° + 270°

set: 1,-2 ; 1,-1 ; 1,1
reset: 0,0 ; 2,0 ; 3,0
offset: 1,+2

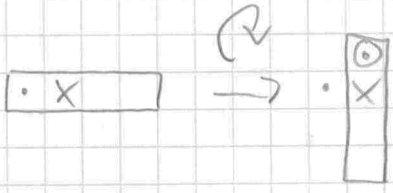


rotate right, 0° + 180°

set: -2, 1 ; -1, 1 ; 1, 1

reset: 0, 0 ; 0, 2 ; 0, 3

offset: -2, 1



rotate right, 90° + 270°

set: 1, -1 ; 1, 1 ; 1, 2

reset: 0, 0 ; 2, 0 ; 3, 0

offset: 1, -1

draw: 0,0 ; 0,1 ; 1,0 ; 1,1

---

move down 0° + 90° + 180° + 270°

set: ~~0,0~~ ; ~~1,0~~ ; 0,2 ; 1,2

reset: 0,0 ; 1,0

offset: 0,1

---

move left 0° + 90° + 180° + 270°

set: -1,0 ; -1,1

reset: 1,0 ; 1,1

offset: -1,0

---

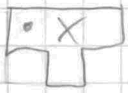
move right

set: 2,0 ; 2,1

reset: 0,0 ; 0,1

offset: 1,0

no rotation



draw: 0,0 ; 1,0 ; 2,0 ; 1,1

move down, 0°

set: 0,1 ; 2,1 ; 1,2

reset: 0,0 ; 1,0 ; 2,0

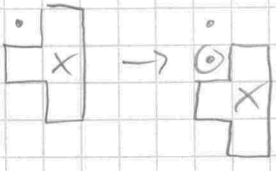
offset: 0,1

move down, 180°

set: 0,2 ; 1,2 ; 2,2

reset: 1,0 ; 0,1 ; 2,1

offset: 0,1

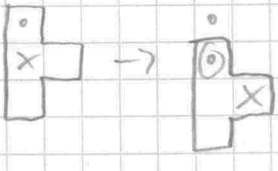


move down, 90° ✓

set: 0,2 ; 1,3

reset: 1,0 ; 0,1

offset: 0,1



move down, 270° ✓

set: 0,3 ; 1,2

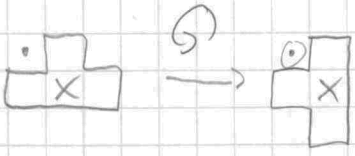
reset: 0,0 ; 1,1

offset: 0,1

rotate left 0°

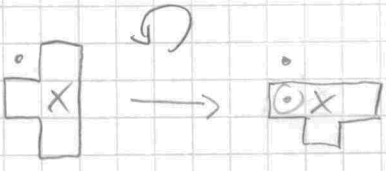
set: 1, -1
reset: 0, 0
offset: 1, -1

rotate left 180°



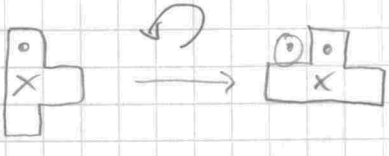
set: 1, 2
reset: 2, 1
offset: 0, 0

rotate left 90°



set: 2, 1
reset: 1, 0
offset: 0, 1

rotate left 270°



set: -1, 1
reset: 0, 2
offset: -1, 0

move left, 0°

set: −1,0; 0,1

reset: 1,1; 2,0

offset: −1,0

move left, 270°

set: −1,0; −1,1; −1,2

reset 0,0; 0,2; 1,1

offset: −1,0

move left, 180°

set: −1,1; 0,0

reset: 1,0; 2,1

offset: −1,0

move left, 90°

set: 0,0; −1,1; 0,2

reset: 1,0; 1,1; 1,2

offset: −1,0

move right, 0°

set: 3,0 ; 2,1
reset: 0,0 ; 1,1
offset: 1,0

move right, 90°

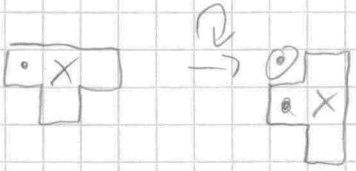
set: 2,0 ; 2,1 ; 2,2
reset: 0,1 ; 1,0 ; 1,2
offset: 1,0

move right, 180°

set: 3,1 ; 2,0
reset: 0,0 ; 0,1 ; 1,0
offset: 1,0

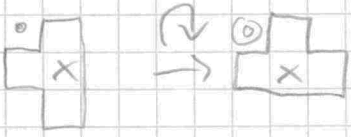
move right, 270°

set: 1,0 ; 1,2 , 2,1
reset: 0,0 ; 0,1 ; 0,2
offset: 1,0



rotate right, 0°

ref: 1, -1 ✓
resd: 2, 0
offred: 0, -1

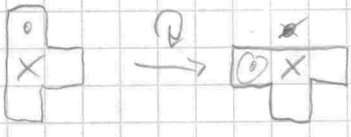


rotate right, 90°

ref: 2, 1
resd: 1, 2 ✓
offset: 0, 0

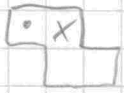
rotate right, 180°

set: 1, 2
reset: 0, 1 ✓
offred: 1, 0



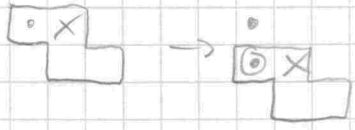
rotate right, 270°

set: -1, 1
reset: 0, 0 ✓
offset: -1, 1



draw

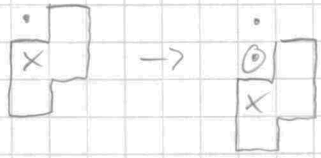
0,0 ; 1,0 ; 1,1 ; 2,1



down, 0°, 180°

set: 0,1 ; 1,2 ; 2,2 offset:

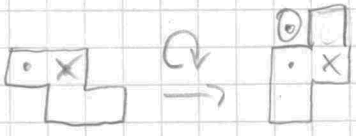
reset: 0,0 ; 1,0 , 2,1 0,1



down, 90°, 270°

set: 1,2 ; 0,3 offset: 0,1

reset: 1,0 ; 0,1



+ left

rotate right, 0°, 180°

set: 1,-1 ; 0,1

reset: 2,1 ; 1,1

offset: 0,-1

+ left

rotate right, 90°, 270°

set: 0,0 ; 2,1

reset: 0,2 ; 0,1

offset: 0,0

move right, 0°, 180°

set: 3,1 ; 2,0
reset: 0,0 ; 1,1
offset: 1,0

move right, 90°, 270°

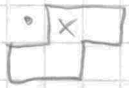
set: 2,0 ; 2,1 ; 1,2
reset 0,1 ; 0,2 ; 1,0
offset 1,0

move left, 0°, 180°

set: -1,0 ; 0,1
reset: 1,0 ; 2,1
offset: -1,0

move left, 90°, 270°

set: -1,1 ; -1,2 ; 0,0
reset: 1,0 ; 1,1 ; 0,2
offset: -1,0



draw

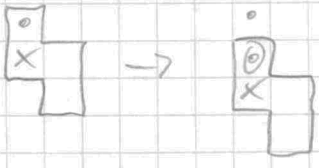
0,1 ; 1,1 ; 1,0 ; 2,0

down 0°, 180°

set: 0,2 ; 1,2 ; 2,1

reset: 0,1 ; 1,0 ; 2,0

offset: 0,1

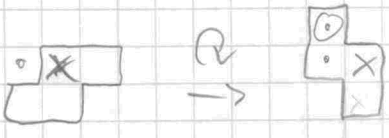


down, 90°, 270°

set: 0,2 ; 1,3

reset: 0,0 ; 1,1

offset: 0,1



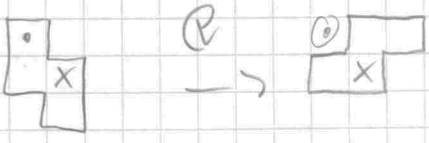
+left

rotate right 0°, 180°

set: 0,0 ; 0,-1

reset: 0,1 ; 2,0

offset: 0,-1



+left

rotate right 90°, 270°

set: 1,0 ; 2,0

reset: 0,0 ; 1,2

offset: 0,0

move right, 0°, 180°

set: 2,1 ; 3,0

reset: 1,0 ; 0,1

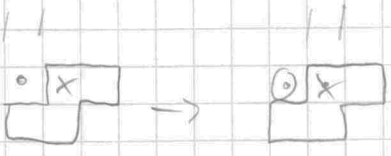
offset: 1, 0

move right, 90°, 270°

set: 1,0 ; 2, 1 ; 2, 2

reset: 0,0 ; 0, 1 ; 1, 2

offset: 1, 0



move left, 0°, 180°

set: 0,0 ; -1, 1

reset: 2,0 ; 1, 1

offset: -1, 0

move left, 90°, 270°

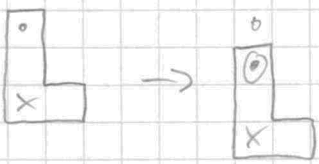
set: -1, 0 ; -1, 1 ; 0, 2

reset 0, 0 ; 1, 1 ; 1, 2

offset: -1, 0

draw

0,0 ; 0,1 ; 0,2 ; 1,2

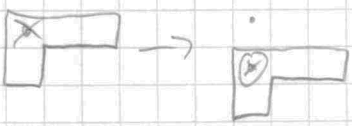


down 0°

set: 0,3 ; 1,3

red: 0,0 ; 1,2

offset: 0,1

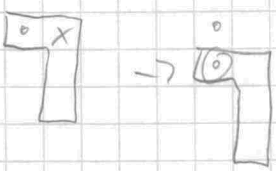


down 90°

set: 1,1 ; 2,1 ; 0,2

red: 0,0 ; 1,0 ; 2,0

offset: 0,1



down 180°

set: 0,1 ; 1,3

red: 0,0 ; 1,0

offset 0,1

down 270°

set: 0,2 ; 1,2 ; 2,2

red: 2,0 ; 0,1 ; 1,1

offset: 0,1

move right, 0°
set: 1,0; 1,1; 2,2
reset: 0,0; 0,1; 0,2
offset: 1,0

move right, 270°
set: 3,0; 3,1
reset: 0,1; 2,0
offset: 1,0

move right, 180°
set: 2,0; 2,1; 2,2
reset: 0,0; 1,1; 1,2
offset: 1,0

move right, 90°
set: 1,1; 3,0
reset: 0,0; 0,1
offset: 1,0

move left, 0°

set: -1,0 ; -1,1 ; -1,2

reset: 0,0 ; 0,1 ; 1,2

offset: -1,0

move left, 270°

set: -1,1 ; 1,0

reset: 2,0 ; 2,1

offset: -1,0

move left, 180°

set: 0,1 ; 0,2 ; -1,0

reset: 1,0 ; 1,1 ; 1,2

offset: -1,0

move left, 90°

set: -1,0 ; -1,1

reset: 0,1 ; 2,0

offset: -1,0

rotate left, 0°

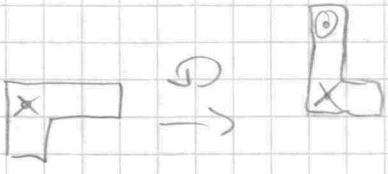
set: -1, 2 ; -2, 2
reset: 0, 0 ; 1, 2
offset: -2, 1

rotate left, 270°

set: 2, 2 ; 2, 3
reset: 0, 1 ; 2, 0
offset: 1, 1

rotate left, 180°

set: 2, 0 ; 3, 0
reset: 0, 0 ; 1, 2
offset 1, 0



rotate left, 90°

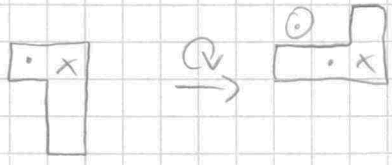
set: 0, -1 ; 0, -2
reset: 0, 1 ; 2, 0
offset: 0, -2

rotate right, 0°

set: 0,3 ; 2,2
reset: 0,0 ; 0,1;
offset: 0,2

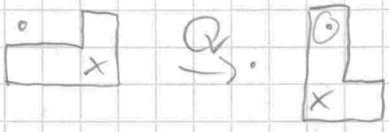
rotate right, 90°

set: -1,0 ; 0,2
reset: 1,0 ; 2,0
offset: -1,0



rotate right, 180°

set: -1,0 ; 1,-1
reset: 1,1 ; 1,2
offset: -1,-1



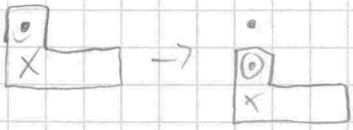
rotate right, 270°

set: 2,-1 ; 3,1
reset: 0,1 , 1,1
offset: 2,-1

draw
0,2 ; 1,0 ; 1,1 ; 1,2

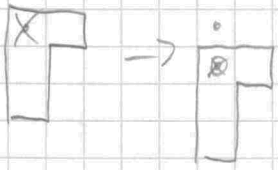
down 0°

set: 0,3 ; 1,3
reset: 0,2 ; 1,0
offset: 0,1



down , 90°

set: 0,2 ; 1,2 ; 2,2
reset: 0,0 ; 1,1 ; 2,1
offset: 0,1

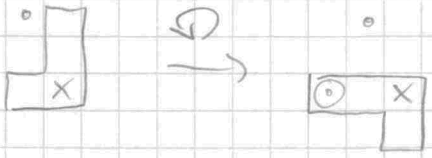


down 180°

set: 0,3 ; 1,1
reset: 0,0 ; 1,0
offset: 0,1

down 270°

set: 0,1 ; 1,1 ; 2,2
reset: 0,0 ; 1,0 ; 2,0
offset: 0,1

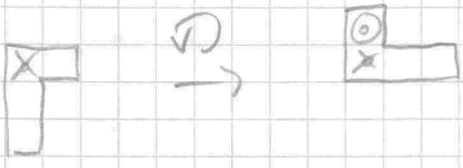


rotate left, 0°

set: 1, 3 ; -1, 2
reset: -1, 0 ; 1, 1
offset: -1, 2

rotate left, 270°

set: 2, 2 ; 3, 0
reset: 0, 0 ; 1, 0
offset: 2, 0



rotate left, 180°

set: 2, 0 ; 0, -1
reset: 0, 1 ; 0, 2
offset: 0, -1

rotate left, 90°

set: -1, 1 ; 0, -1
reset: 1, 1 ; 2, 1
offset: -1, -1

move right, 0°

set: 2,0 ; 2,1 ; 2,2

reset: 1,0 ; 1,1 ; 0,2

offset: 1,0

move right, 270°

set: 3,0 ; 3,1

reset: 0,0 ; 2,1

offset: 1,0

move right, 180°

set: 1,1 ; 1,2 ; 2,0

reset: 0,0 ; 0,1 ; 0,2

offset: 1,0

move right, 90°

set: 1,0 ; 3,1

reset: 0,0 ; 0,1

offset: 1,0

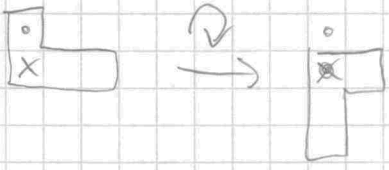
move left, 0°
set: 0,0; 0,1; -1,2
reset: 1,0; 1,1; 1,2
offset: -1,0

move left, 270°
set: -1,0; 1,1
reset: 2,0; 2,1
offset: -1,0

move left, 180°
set: -1,0; -1,1; -1,2
reset: 1,0; 0,1; 0,2
offset: -1,0

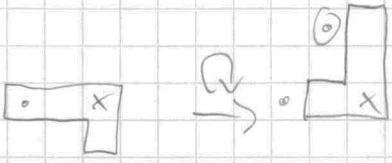
move left, 90°
set: -1,0; -1,1
reset: 0,0; 2,1
offset: -1,0

rotate right, 0°
ref: 2,2 ; 3,2
rref: 1,0 ; 0,2
offset: 1,1

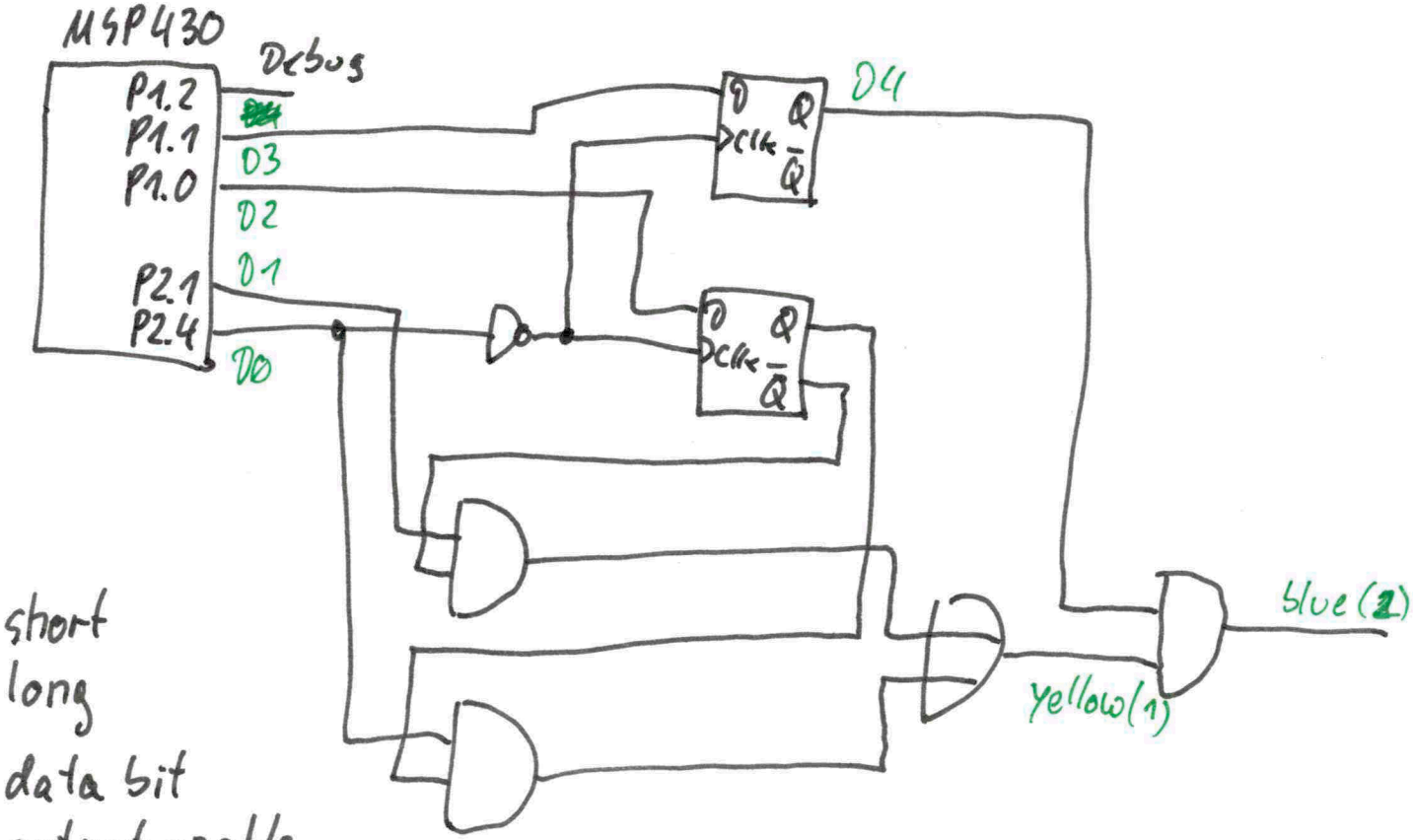


rotate right, 90°
ref: 0,2 ; 0,3
rref: 0,0 ; 2,1
offset: 0,1

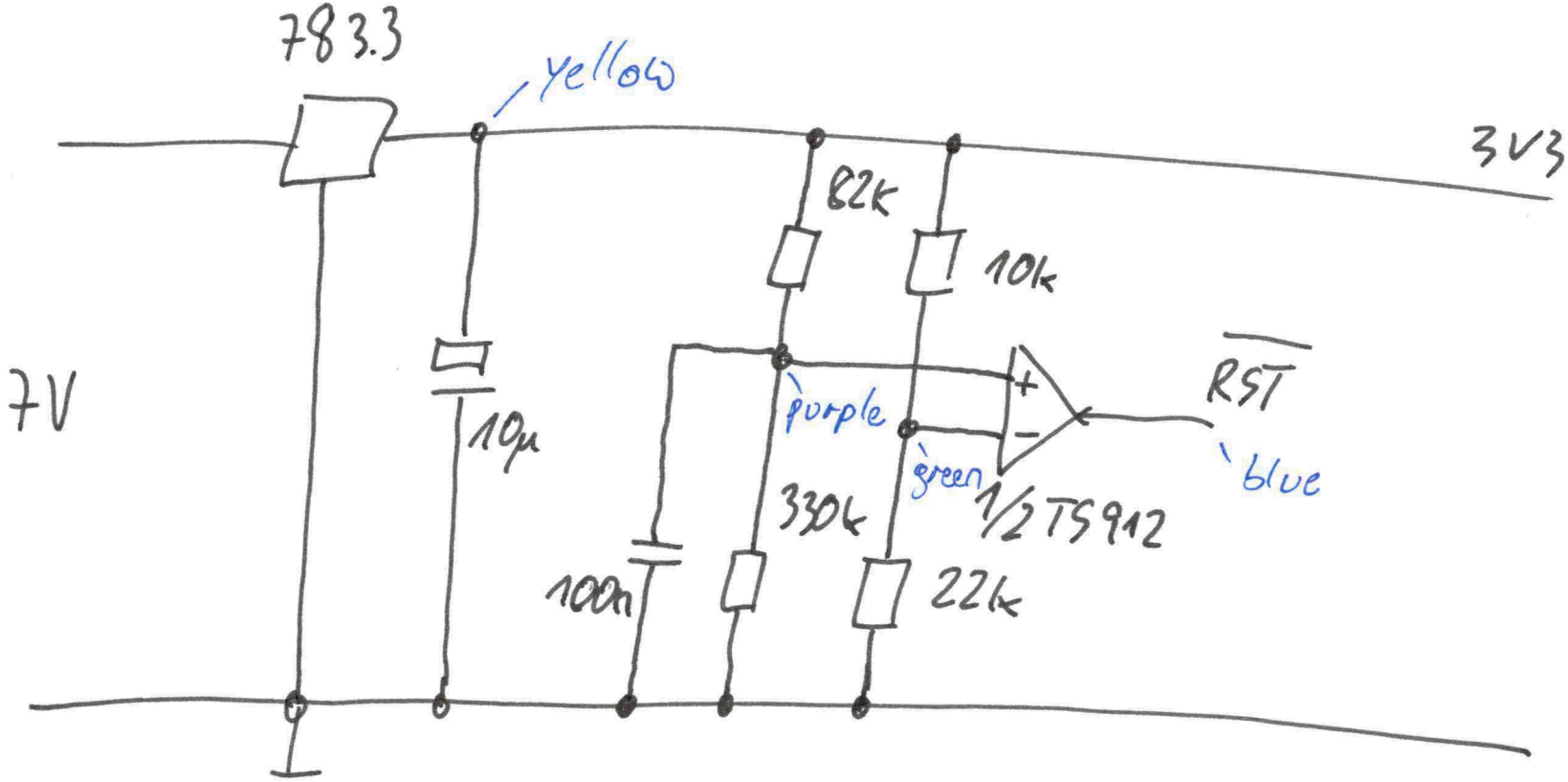
rotate right, 180°
ref: -1,0 ; -2,0
rref: 1,0 ; 0,2
offset: -2,0

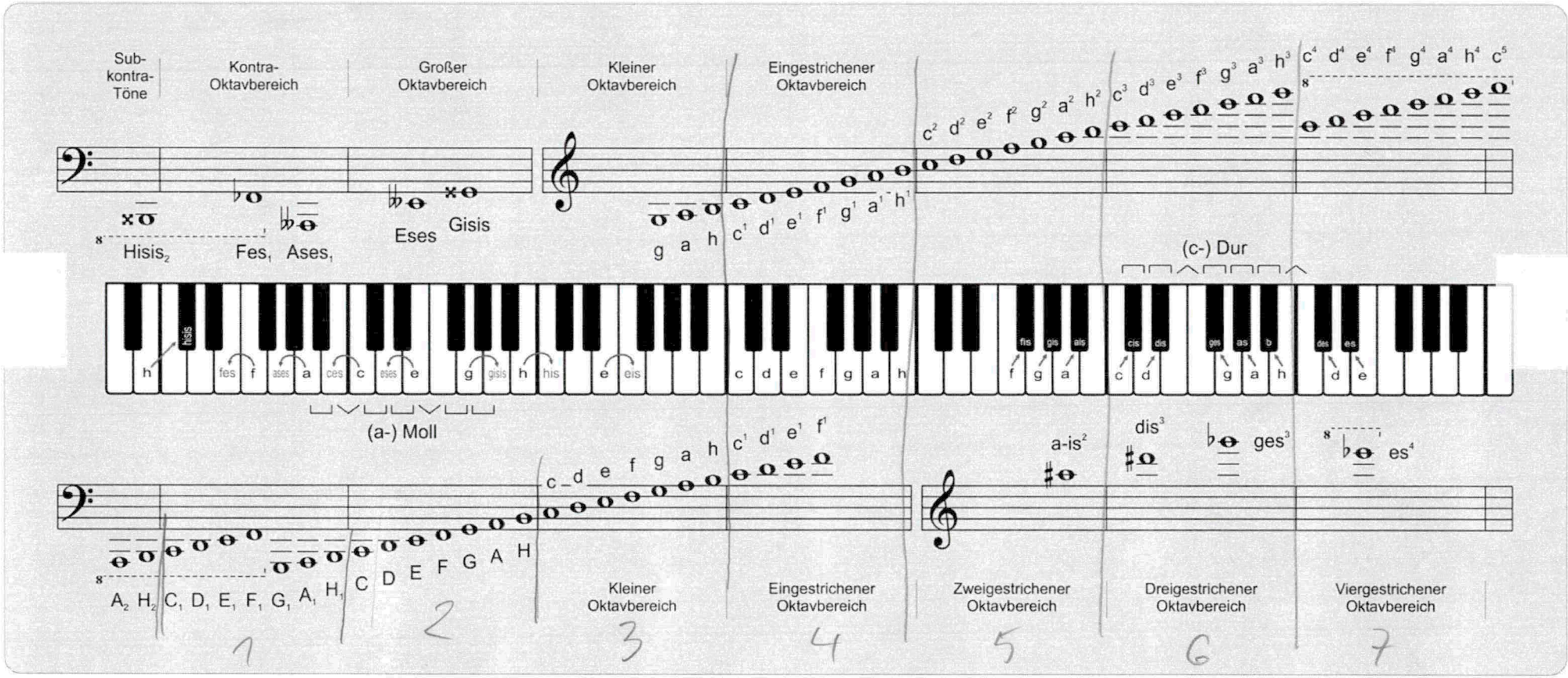


rotate right, 270°
ref: 2,-1 ; 2,-2
rref: 0,0 ; 2,1
offset: 1,-2



P2.1 short
P2.2 long
P1.0 data bit
P1.1 output enable


7833
yellow
3V3
82k
10k
7V
10µ
purple
green
RST
blue
330k
1/2 TS912
100n
22k



Klaviernotation

Weitere Einzelheiten

CellarDoor85 (Robert Aehnelt) - Eigenes Werk

Piano - Notation

File: Klaviernotation.png

Erstellt: 30. September 2011

Hochgeladen: 22. Januar 2022

Fig. 3 PSG REGISTER ARRAY

REGISTER \ BIT		B7	B6	B5	B4	B3	B2	B1	B0
R0	Channel A Tone Period	8-BIT Fine Tune A							
R1						4-BIT Coarse Tune A			
R2	Channel B Tone Period	8-BIT Fine Tune B							
R3						4-BIT Coarse Tune B			
R4	Channel C Tone Period	8-BIT Fine Tune C							
R5						4-BIT Coarse Tune C			
R6	Noise Period				5-BIT Period Control				
R7	Enable	IN/OUT		Noise			Tone		
		IOB	IOA	C	B	A	C	B	A
R10	Channel A Amplitude				M	L3	L2	L1	L0
R11	Channel B Amplitude				M	L3	L2	L1	L0
R12	Channel C Amplitude				M	L3	L2	L1	L0
R13	Envelope Period	8-BIT Fine Tune E							
R14		8-BIT Coarse Tune E							
R15	Envelope Shape/Cycle					CONT.	ATT.	ALT.	HOLD
R16	I/O Port A Data Store	8-BIT PARALLEL I/O on Port A							
R17	I/O Port B Data Store	8-BIT PARALLEL I/O Port B							

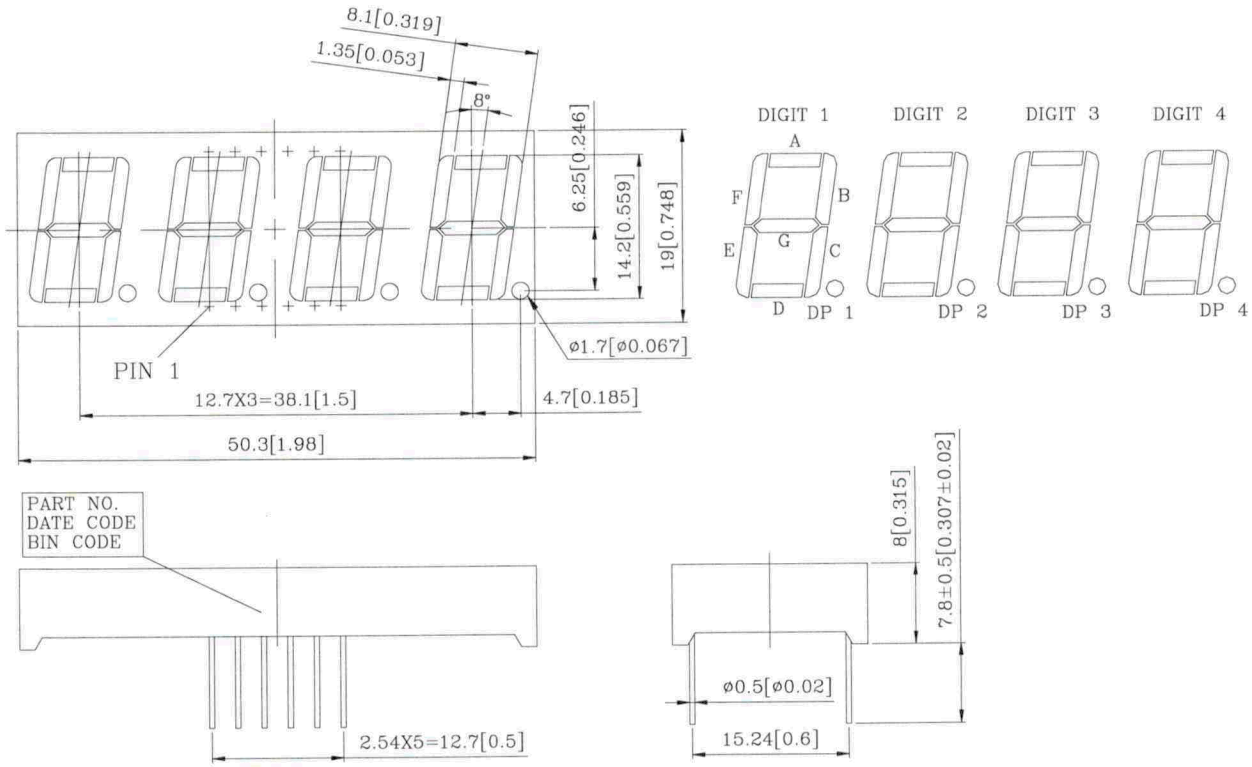
↑
octal numbers!

NOTE	OCTAVE	IDEAL FREQUENCY	ACTUAL FREQUENCY	12-BIT REGISTER VALUE IN OCTAL			
C	1	32.703	32.698	6	5	3	5
C#	1	34.648	34.653	6	2	3	4
D	1	36.708	36.712	5	7	4	7
D#	1	38.891	38.895	5	4	7	4
E	1	41.203	41.201	5	2	3	3
F	1	43.654	43.662	5	0	0	2
F#	1	46.249	46.243	4	5	6	3
G	1	48.999	48.997	4	3	5	3
G#	1	51.913	51.908	4	1	5	3
A	1	55.000	54.995	3	7	6	2
A#	1	58.270	58.261	3	6	0	0
B	1	61.735	61.733	3	4	2	4
C	2	65.406	65.416	3	2	5	6
C#	2	69.296	69.307	3	1	1	6
D	2	73.416	73.399	2	7	6	4
D#	2	77.782	77.789	2	6	3	6
E	2	82.406	82.432	2	5	1	5
F	2	87.308	87.323	2	4	0	1
F#	2	92.498	92.523	2	2	7	1
G	2	97.998	98.037	2	1	6	5
G#	2	103.826	103.863	2	0	6	5
A	2	110.000	109.991	1	7	7	1
A#	2	116.540	116.522	1	7	0	0
B	2	123.470	123.467	1	6	1	2
C	3	130.812	130.831	1	5	2	7
C#	3	138.592	138.613	1	4	4	7
D	3	146.832	146.799	1	3	7	2
D#	3	155.564	155.578	1	3	1	7
E	3	164.812	164.743	1	2	4	7
F	3	174.616	174.510	1	2	0	1
F#	3	184.996	184.894	1	1	3	5
G	3	195.996	195.903	1	0	7	3
G#	3	207.652	207.534	1	0	3	3
A	3	220.000	220.198	0	7	7	4
A#	3	233.080	233.043	0	7	4	0
B	3	246.940	246.933	0	7	0	5
C	4	261.624	261.357	0	6	5	4
C#	4	277.184	276.883	0	6	2	4
D	4	293.664	293.598	0	5	7	5
D#	4	311.128	310.724	0	5	5	0
E	4	329.624	329.973	0	5	2	3
F	4	349.232	349.565	0	5	0	0
F#	4	369.992	370.400	0	4	5	6
G	4	391.992	392.494	0	4	3	5
G#	4	415.304	415.839	0	4	1	5
A	4	440.000	440.397	0	3	7	6
A#	4	466.160	466.087	0	3	6	0
B	4	493.880	494.959	0	3	4	2

NOTE	OCTAVE	IDEAL FREQUENCY	ACTUAL FREQUENCY	12-BIT REGISTER VALUE IN OCTAL			
C	5	523.248	522.714	0	3	2	6
C#	5	554.368	553.766	0	3	1	2
D	5	587.328	588.741	0	2	7	6
D#	5	622.256	621.449	0	2	6	4
E	5	659.248	658.005	0	2	5	2
F	5	698.464	699.130	0	2	4	0
F#	5	739.984	740.800	0	2	2	7
G	5	783.984	782.243	0	2	1	7
G#	5	830.608	828.598	0	2	0	7
A	5	880.000	880.794	0	1	7	7
A#	5	932.320	932.173	0	1	7	0
B	5	987.760	989.918	0	1	6	1
C	6	1046.496	1045.428	0	1	5	3
C#	6	1108.736	1107.532	0	1	4	5
D	6	1174.656	1177.482	0	1	3	7
D#	6	1244.512	1242.898	0	1	3	2
E	6	1318.496	1316.009	0	1	2	5
F	6	1396.928	1398.260	0	1	2	0
F#	6	1479.968	1471.852	0	1	1	4
G	6	1567.968	1575.504	0	1	0	7
G#	6	1661.216	1669.564	0	1	0	3
A	6	1760.000	1747.825	0	1	0	0
A#	6	1864.640	1864.346	0	0	7	4
B	6	1975.520	1962.470	0	0	7	1
C	7	2092.992	2110.581	0	0	6	5
C#	7	2217.472	2237.216	0	0	6	2
D	7	2349.312	2330.433	0	0	6	0
D#	7	2489.024	2485.795	0	0	5	5
E	7	2636.992	2663.352	0	0	5	2
F	7	2793.856	2796.520	0	0	5	0
F#	7	2959.936	2943.705	0	0	4	6
G	7	3135.936	3107.244	0	0	4	4
G#	7	3322.432	3290.023	0	0	4	2
A	7	3520.000	3495.649	0	0	4	0
A#	7	3729.280	3728.693	0	0	3	6
B	7	3951.040	3995.028	0	0	3	4
C	8	4185.984	4142.992	0	0	3	3
C#	8	4434.944	4474.431	0	0	3	1
D	8	4698.624	4660.866	0	0	3	0
D#	8	4978.048	5084.581	0	0	2	6
E	8	5273.984	5326.704	0	0	2	5
F	8	5587.712	5593.039	0	0	2	4
F#	8	5919.872	5887.410	0	0	2	3
G	8	6271.872	6214.488	0	0	2	2
G#	8	6644.864	6580.046	0	0	2	1
A	8	7040.000	6991.299	0	0	2	0
A#	8	7458.560	7457.385	0	0	1	7
B	8	7902.080	7990.056	0	0	1	6

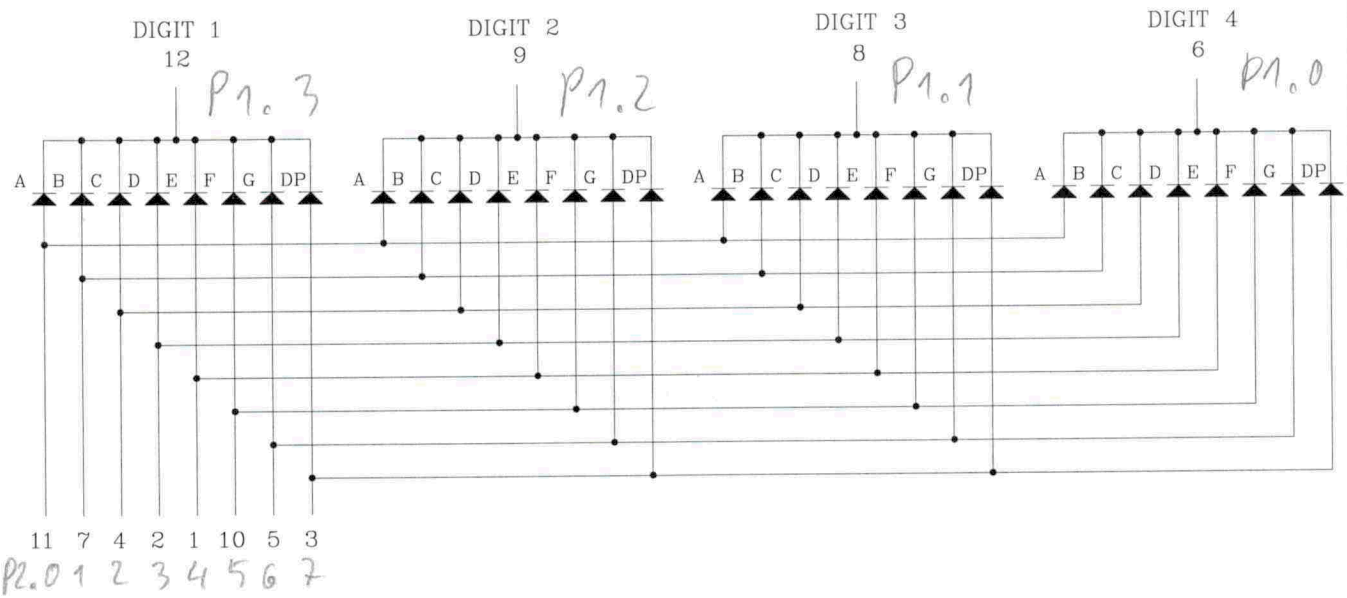
Fig. 23 EQUAL TEMPERED CHROMATIC SCALE (f_{CLOCK}=1.78977MHz)

PACKAGE DIMENSIONS



NOTES: All dimensions are in millimeters. Tolerances are± 0.25 mm (0.01") unless otherwise noted.

INTERNAL CIRCUIT DIAGRAM



Programmable Sound Generator

FEATURES

- Full Software Control of Sound Generation
- Interfaces to Most 8-Bit and 16-Bit Microprocessors
- Three Independently Programmed Analog Outputs
- Two 8-Bit General Purpose I/O Ports (AY-3-8910)
- One 8-Bit General Purpose I/O Port (AY-3-8912)
- Single +5 Volt Supply

DESCRIPTION

The AY-3-8910/8912/8913 Programmable Sound Generator (PSG) is a LSI Circuit which can produce a wide variety of complex sounds under software control. The AY-3-8910/8912/8913 is manufactured in the General Instrument N-Channel Ion Implant Process. Operation requires a single +5V power supply, a TTL compatible clock, and a microprocessor controller such as the General Instrument 16-bit CP1610 or one of the PIC1650 series of 8-bit microcomputers.

The PSG is easily interfaced to any bus oriented system. Its flexibility makes it useful in applications such as music synthesis, sound effects generation, audible alarms, tone signalling and FSK modems. The analog sound outputs can each provide 4 bits of logarithmic digital to analog conversion, greatly enhancing the dynamic range of the sounds produced.

In order to perform sound effects while allowing the processor to continue its other tasks, the PSG can continue to produce sound after the initial commands have been given by the control processor. The fact that realistic sound production often involves more than one effect is satisfied by the three independently controllable channels available in the PSG.

All of the circuit control signals are digital in nature and intended to be provided directly by a microprocessor/microcomputer. This means that one PSG can produce the full range of required sounds with no change in external circuitry. Since the frequency response of the PSG ranges from sub-audible at its lowest frequency to post-audible at its highest frequency, there are few sounds which are beyond reproduction with only the simplest electrical connections.

Since most applications of a microprocessor/PSG system would also require interfacing between the outside world and the microprocessor, this facility has been designed into the PSG. The AY-3-8910 has two general purpose 8-bit I/O ports and is supplied in a 40 lead package: the AY-3-8912 has one port and 28 leads: the AY-3-8913 has no ports and 24 leads.

PIN FUNCTIONS

DA7--DA0 (input/output/high impedance): pins 30--37 (AY-3-8910)
Data/Address 7--0: pins 21--28 (AY-3-8912)
pins 4--11 (AY-3-8913)

These 8 lines comprise the 8-bit bidirectional bus used by the microprocessor to send both data and addresses to the PSG and to receive data from the PSG. In the data mode, DA7--DA0 correspond to Register Array bits B7--B0. In the address mode, DA3--DA0 select the register number ($0--17_8$) and a DA7--DA4 in conjunction with address inputs $\overline{A9}$ and A8 for the high order address (chip select).

A8 (input): pin 25 (AY-3-8910)
pin 17 (AY-3-8912)
pin 23 (AY-3-8913)

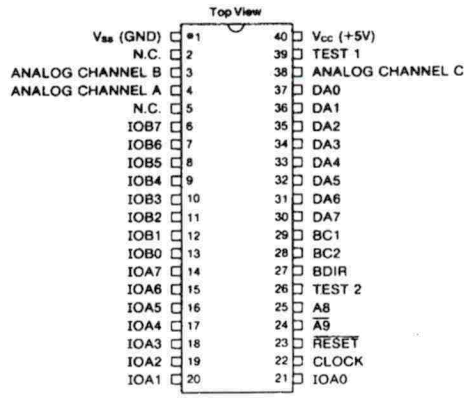
$\overline{A9}$ (input): pin 24 (AY-3-8910)
pin 22 (AY-3-8913)
(not provided on AY-3-8912)

$\overline{\text{Address 9}}$, Address 8

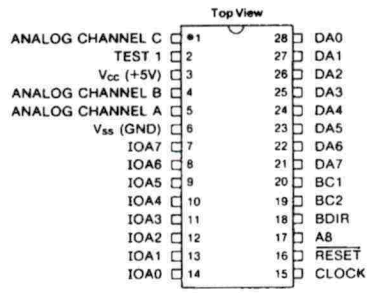
These "extra" address bits are made available to enable the positioning of the PSG (assigning a 16 word memory space) in a total 1,024 word memory area rather than in a 256 word memory area as defined by address bits DA7--DA0 alone. If the memory size does not require the use of these extra address lines they may be left unconnected as each is provided with either an on-chip pull down ($\overline{A9}$) or pull-up (A8) resistor. In "noisy" environments, however, it is recommended that $\overline{A9}$ and A8 be tied to an external ground and +5V, respectively, if they are not to be used.

PIN CONFIGURATIONS

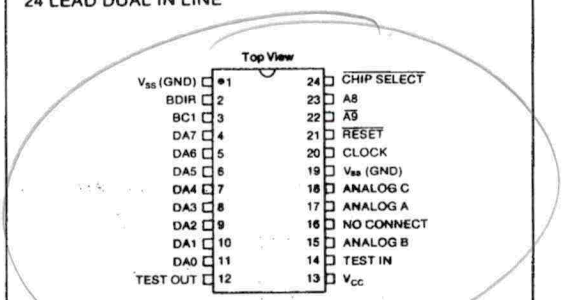
40 LEAD DUAL IN LINE
AY-3-8910



28 LEAD DUAL IN LINE
AY-3-8912



24 LEAD DUAL IN LINE



9

DA0-7 : P2.0-7

~~A8~~ -> ~~P1.0~~

BC1 -> P1.3

BDIR -> P1.1

P1.2 $\overline{CS0}$

P1.0 $\overline{CS1}$

~~$\overline{CS}$~~ -> ~~P1.3~~

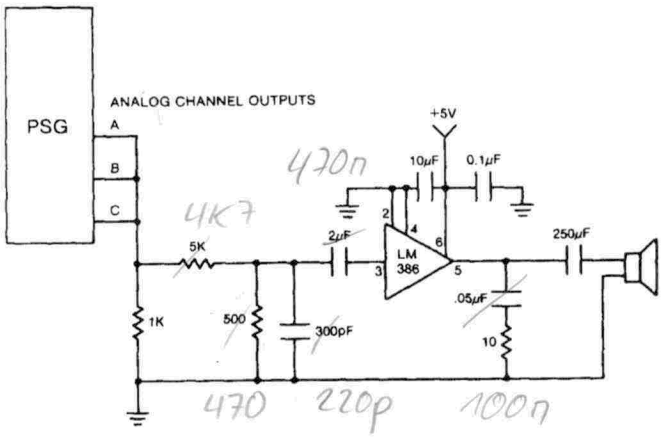
$\overline{RST}$ -> P1.2

4.3 Audio Output Interface

Fig. 16 illustrates the audio output connections to a commercially available LM386 audio amplifier. It shows channels A, B, and C summed together to enable complex waveforms to becomposed and amplified through a single external amplifier. These channels may be individually amplified through separate channels for more exotic sound systems.

Each output channel is individually controlled by separateamplitude registers (R10, R11, R12) and an enable register (R7) in the PSG.

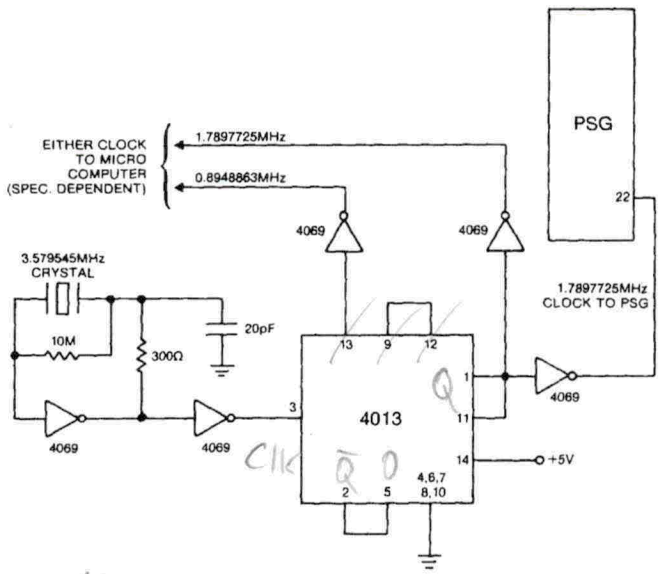
Fig. 16 AUDIO OUTPUT INTERFACE



Clock Generation

4 . 2 An economical solution to providing a system clock is shown in Fig. 15. It consists of a 3.579545MHz standard color burst crystal, a CD4089 CMOS inverter, and a CD4013 to divide the color burst frequency in half. The clock produced for the PSG runs at a 1.7897725MHz rate. Depending on the microcomputer used, its clock should be selected within its specified value.

Fig. 15 CLOCK GENERATION



4069 → 74HC04

4013 → 74HC74

Theme A - Tetris

http://www.gamemusicthemes.com

Arranged by Gori Fater