Synth- DIY

Skills & Knowledge

Electronics, Hardware Shopping
Soldering
Oscilloscope

C/C++
Arduino Plattform, Library APIs
IDE: Visual Studio + Visualmicro + Arduino IDE
Git SCM
UML for Documentation

Eagle, Fritzing for PCB

CAD for Frontpanel (Laser Cutting)
Inkscape for Design of Panel
Case made of wood

License, Patents
Marketing: Webdesign, SEO, Social Media

1. Case + Power
2a. Frontpanel and Mounting
    Print- Layer with CAD, Inkscape, GIMP
2b. Wiring and Connectivity
    MIDI -> Sync, Master- Keyboard
    CV/Gate -> Korg monotron Mod
    Wifi + BT -> OSCTouch, Laptop, Wiimote
3a. Controller, analog
    Potis, Switches, Joystick, Ribbon
3b. Controller and MCU
    Display + Encoder, Trellis, Button Array
4. Arpeggiator/ Step- Sequencer with Arduino
5. Klangerzeugung
5a. Digital: Auduino (Granular Synthesizer)
5b. Analog: MFOS AlienScreamer
5c. x0xb0x
5d. Sampling/ Drums with Raspberry Pi
    PyGame, Drum Kit Kit + MCP3008 (ADC)
5e. Pure Data on Raspberry Pi
5f. DAW: LMMS, Audacity, GM, SF2, VST
6. Wiimote (BT Controller), TouchOSC (Wifi)
7. external Devices
7a. BassStation (MIDI- Keyboard, analog Synth)
7b. Jimi Box (Doepfer), MCV4, R2M
7c. Korg SQ10
7d. Yamaha QY100 (Sequencer, GM)
8. PC (GM, SF2, DAW, VST): Ableton Live

Tracks (vertical), Voices

8 Tracks

Lead Synth, Melody
2nd Voice, Melody
Strings, Harmony
Strings 2, Harmony
Bass
Drumkit
    Percussion, FX
    Cymbals
    Tom-Toms
    Snare, Clap
    Bassdrum

8
16
32 Beats Patterns

Sequence, Patterns (horizontal, time)
Song Parts

8 Parts

Intro
Fill
Main, Refrain
Fill, Break
Main 2, Variation
Fill, Break
Ending

Controller

Mimugloves.com
(Imogen Heap)

Wiimote

Kinect

Android Smartphones
- TouchOSC
- Sensors to Serial Protocoll via Wifi

MIDI-(Master-Keyboard)-Controller
- M-Audio
-

MIDI - In
CLK
Sequencer
Step
ARP
Noise
OSC 1
OSC 2
FM
PWM
ext
Mixer
VCF
VCA
(output)
FX
Mix-Out
IN
MG
AM
LFO 1
EG 2
EG 1
LFO 2
CV - In
Gate

Clock intern / MIDI
Tempo
Portamento / Glide

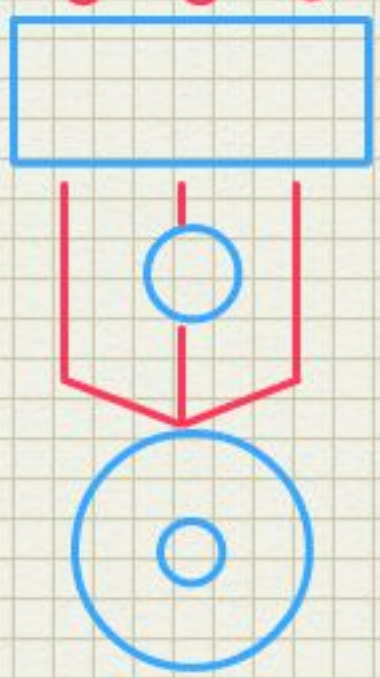
- pitch
- Level
  Shape

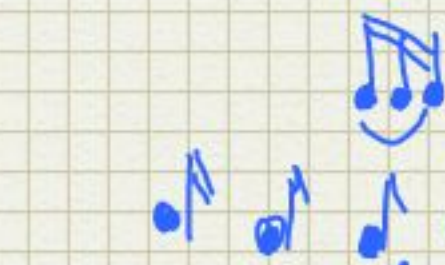
- Cutoff Freq.
- Resonance / Peak

- Rate
  Sync intern | off | MIDI CLK
  Shape
- Intensity

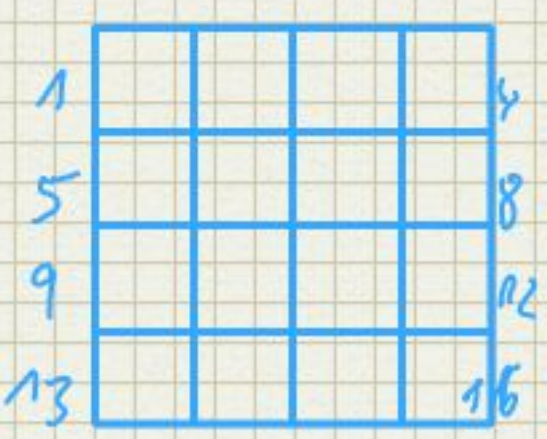
- Mod. Intensity
  Mod. Source: LFO | EG
- ADSR
  AR

Joy
X
Y
Rib
Mod
US
Dist
ACC.
X
Y
Z
Wii mote
MIDI Btn. Trellis
CV
Gate





Quintenzirkel
Kadenz
Intervalle



PiTFT

<Arduino>

<FBTFT>

Arduino — USB — Raspberry Pi — WLAN

ADC

MUX 16 Ch

analog

Joystick
Potis
UltraSonic
LDR
...

Shift Register

MIDI

PWM

GPIO

Audio

DAC

CV
Gate/
Trigger

Korg monotron Mod

Cam

openFrameworks
OpenCV

BT

Wiimote

<PD>

DAW — Samples

Wine
Rebirth | VST

OSC

TouchOSC

Lemur
(iPhone)

Breadboard

MFOS analog
Synthesizer
(Modules)

Poly 800 Step-Seq. | Arpeggiator | 808

<Neopixels>

<Potis>

<7 seg Disp>

<Encoder>

<select>

<Joystick>

Pitch

<Pushbuttons>

Pause
Step

Shift

<Trellis>

<select>

Audvino, Groovesizer RED

MIDI

Granular Synthese

Korg monotron

Controller & Sequencer

Step

Arpeggiator

Pattern

Joystick

Ribbon

Pots

MUX

LED-Buttons

Nebulophone

MIDI

CV/Gate, DAC

7-Seg./Alphanum. Display

Encoder (w. LED)

Trellis

Neopixel (32 RGB-LED)

Drum Kit Kit

Raspberry Pi, GPIO, ADC (8 Inputs)

Samples: BBox

PD, OSC, Wiimote, TouchOSC

Pygame + FBTFT

Laptop

- USB, WLAN
- Pure Data
  Serial 2 MIDI
  DAW, VST

Alien Screamer

Raspberry Pi (Linux)

Pure Data : OSC, TouchOSC (Android),
Wiimote + OSCulator | GlovePie

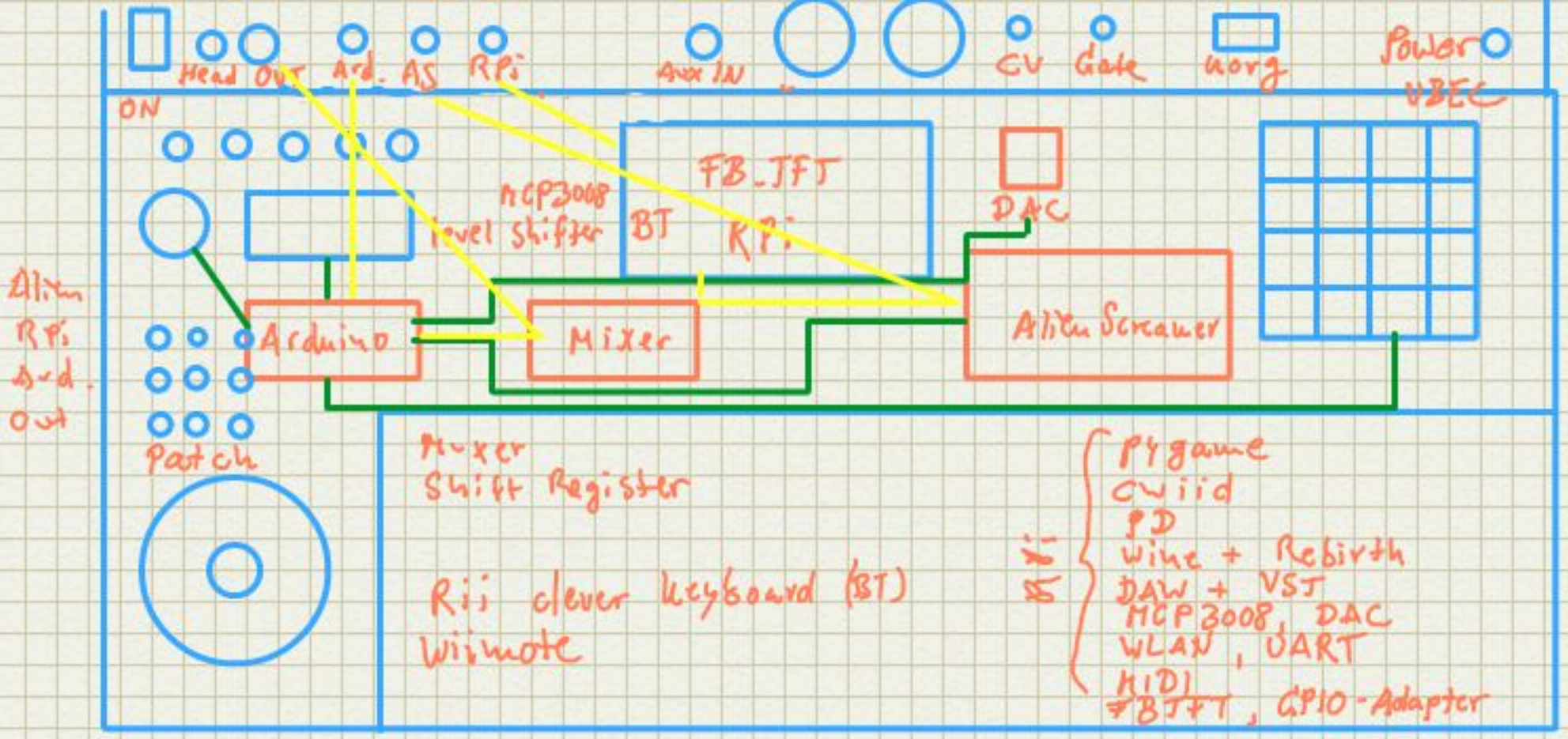
DAW
VST-plugins : Sampler | Drumkit
Propellerheads Rebirth + Wine
Audacity (Multitrack Recording)

Pygame (GUI for FBTFT)
FBTFT + Rii Clever (QWERTZ) + BT
WLAN + VNC + Laptop | Handy
HDMI + USB-Tastatur/Maus

Hardware:

GPIO + ADC (MCP MUX) } + Sensors (Joystick, Ribbon, Buttons, etc.)

USB
BT
WLAN



Keyboard

CV | Gate

Seq Arp

EG

Env 1
Env 2

MG

LFO

Aux
Noise
VCO2
VCO1

HP
LP

VCO — Mix — VCF — VCA — FX ——— Audio Signal

Chorus
Hall
Echo
Dist

FM

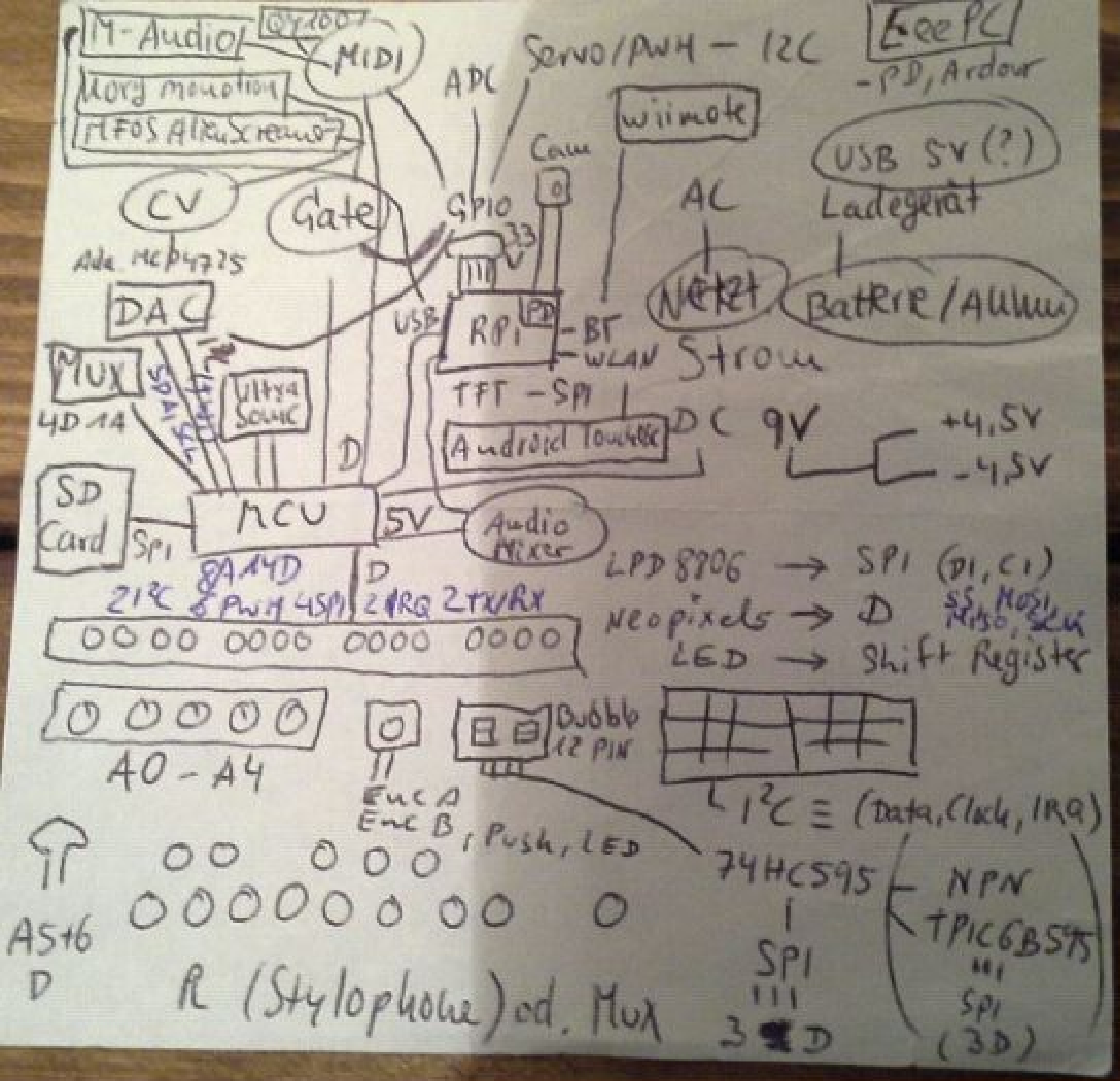
M

AM

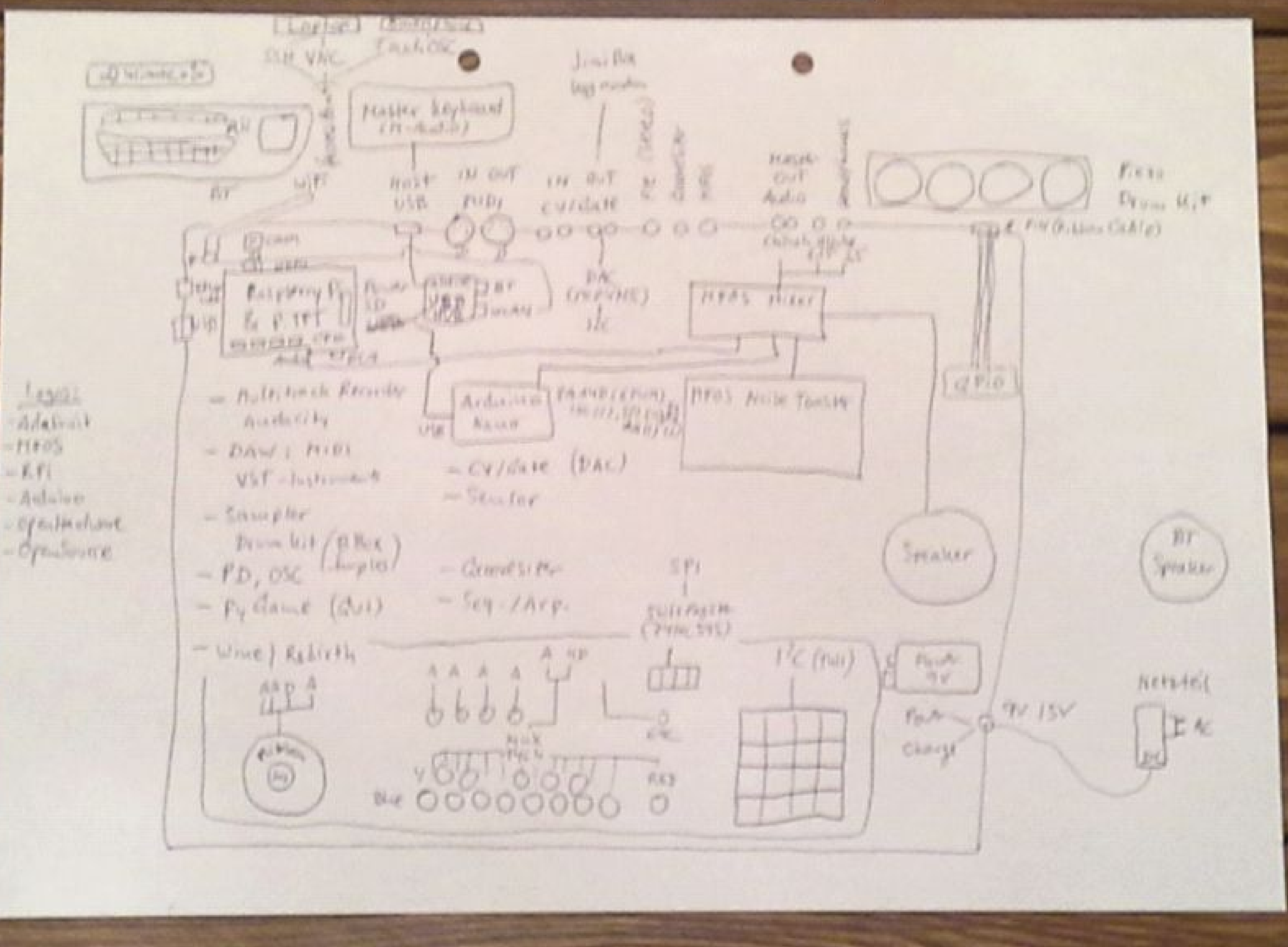
Freq Mod

Freq

Level



M-Audio
QY100
MIDI
Korg monotron
MFOS AlienScreamer
Servo/PWM - I2C
ADC
Eee PC
- PD, Ardour
wiimote
USB 5V (?)
Cam
CV
Gate
GPIO
AC
Ladegerät
Ada MCP4725
DAC
USB
RPi
PD
- BT
- WLAN
Netz
Batterie/Akku
Strom
MUX
4D 14
SPI SCL
Ultra Sonic
TFT - SPI
Android Toolkit
DC 9V
+4,5V
-4,5V
SD Card
SPI
MCU
5V
D
Audio Mixer
8A 14D
2I2C 8PWM 4SPI 2IRQ 2TX/RX
LPD 8806 → SPI (DI, CI)
Neopixels → D
SS, MOSI, MISO, SCK
LED → Shift Register
0000 0000 0000 0000
00000
A0 - A4
Bubble 12 PIN
Enc A
Enc B, Push, LED
I2C ≡ (Data, Clock, IRQ)
74HC595
NPN
TPIC6B595
SPI
(3D)
A5+6
D
R (Stylophone) od. Mux
3 D



Master Keyboard
(M-Audio)
BT
WiFi
HOST USB
IN OUT MIDI
IN OUT CV/Gate
OUT Audio
Piezo Drum Kit
DAC
(MCP4725)
I2C
MFOS Mixer
GPIO
Arduino
MFOS Noise Toaster
Legend
- Adafruit
- MFOS
- RPi
- Arduino
- OpenSource
- Multitrack Recorder Audacity
- DAW + MIDI VST-Instruments
- CV/Gate (DAC)
- Sampler Drum Kit
- PD, OSC
- PyGame (GUI)
- Sequencer
- Seq. / Arp.
- Wine / Rebirth
SPI
Shift Register (74HC595)
I2C
Power 9V
9V 15V
Speaker
BT Speaker
AC

Nano 1,85 x 4,32 Uno 6,86 x 5,33

Touch OSC WLAN 85,6 x 56 mm

WLAN + SSH + VNC

Arduino 5V 7-12V 40mA (pro PIN)

8A 14D 6PWM

Tx, Rx

I²C (TWI)

SPI 4

Raspberry 93 x 63,5 mm 5V 700mA (=3,5W)

2,8", 320 x 240px

Pi TFT 11,8 x 8,6 SPI, GPIO #25,24 Switches #23,22 #21,18

17 GPIO
2 UART
2 I²C
5 SPI
1 PWM
6 D

GPIO 3,3V, 50mA (für alle PINS)

MIDI Tx Rx

Wiimote BT

BT + Keyboard

USB Host M-Audio 5V, ca. 300mA bei 1A Netzteil (-700mA Raspi)

ADC SPI MCP3008

MIDI IN: Clock TX RX

CV / Gate out → DAC I²C MCP4725

4.7 k Potis 10 k Potis

Arduino / Groovesizer

Synthe Sequencer 5A 5D

100 k Potis 1 M

FBTFT

I²C, SPI, wiringPi, ...

Bluetooth, Cwiid.py

BBox Samples

Pygame

PD

LMMS, Ardour, VST

Audacity

MFOS Alien Screamer 9,4 x 5,08 9V DC

MFOS 3Ch-Amp (Mixer) 6,35 x 3,3 9V DC

Potis, Switches MS244 MS500, Buchsen 3,5; 6,3; M5 5mm M6 M6 M7 M9 M10

6mm; Button 10mm (Exp-Tech)

Lötösenstecker

144 LED/m Neopixels 32 D 22,2 cm Ø5mm 5V, 1,92A 26,6 cm 5V, 60mA per LED

60 LED/m LPD 8806 16-LED SPI 2D (D, C)

Touleiter mit LED-Buttons Ø 16mm (MUX) CD74HC4067 4D 1A

Shift-key

Joystick

Ribbon Controller

2A 1D A 3,81 cm Ø ... 65,6 mm (Ø außen) Ø innen ...

HT16K33

alphanum 4. digit 0,54" Display I²C (14 Seg.) 50,2 x 21,5 51 x 28 (PCB)

Encoder 2D 6mm

RGB LED 15mm Button

Drehschalter

Adafruit Trellis LED-Array, Buttons HT16K33 I²C 60 x 60mm

Piezo Drum kit 4-6A → ADC MCP3008

Shift-Register SN74HC595N SIPO

16 Pole 74HC165 PISO

LED - Button

Bohrung
Ø 16

Taster
Schwarz / rot

Ø 7
b = 10

Kippschalter
MS 244

Ø 5
b = 5x8

MS 500

Ø 6
b = 8 x 13

Poti
Alps 10k
(PCB)

Ø 6
b = 10 x 13

Poti
Reichelt 4.7k

Ø 10 x 9
b = 21
Shaft 6 mm

Poti
Adafruit 10k

Ø 7
b = 17
Shaft Stern

Encoder
LED
Exp-Tech

Ø 8
b = 13 x 15

Encoder
Reichelt

Ø 9,5
b = 16,5
x 21
D-Shaft
Ø 6,5

ohne Gewinde

Button
silber
mit Schraube
für 6mm Shaft

Reichelt
Ø 12

Button
Schwarz
für Sternform

Exp-Tech
Ø 10

Button
transparent
Ø 15 - 16

Knob
Ø 25 mm

MIDI - Port
Metall
Einbau

Ø 18 x 17
außen Ring 23 mm

Kunststoff
PCB

b = 20 x 20

Audio Buchsen
3,5 mm

Ø 6

6,25 mm

Ø 9
b = 16
t ≈ 24

Strom
DC

Ø 8

Korg monotron
6-PIN-Buchse

Joystick
Adafruit

Ø 20 (Bohrung)
b = 27 (Joy, Plastik)

C = 38 (PCB)
d = 30 (Lochabstand)

Joystick
Sainsmart

Ø 17
b = 26

C = 34 x 26
d = 26 x 19

Drehschalter (3 od. 6 Stellungen)

10 mm Bohrung

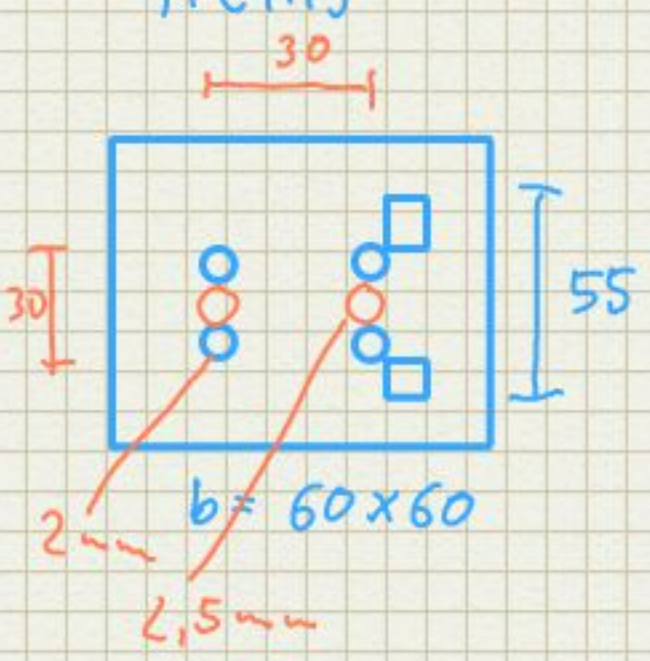
Ø 26

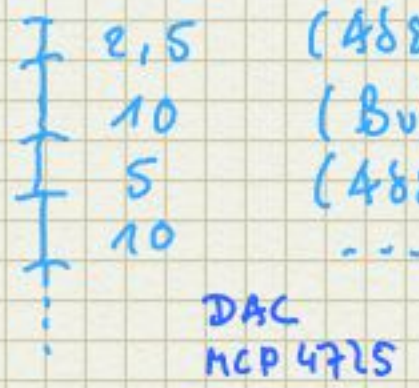
Shaft 6 mm

Mounting holes of PCBs

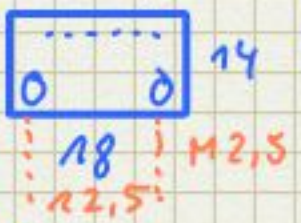
Befestigungs-Material M3 Gewinde

Trellis



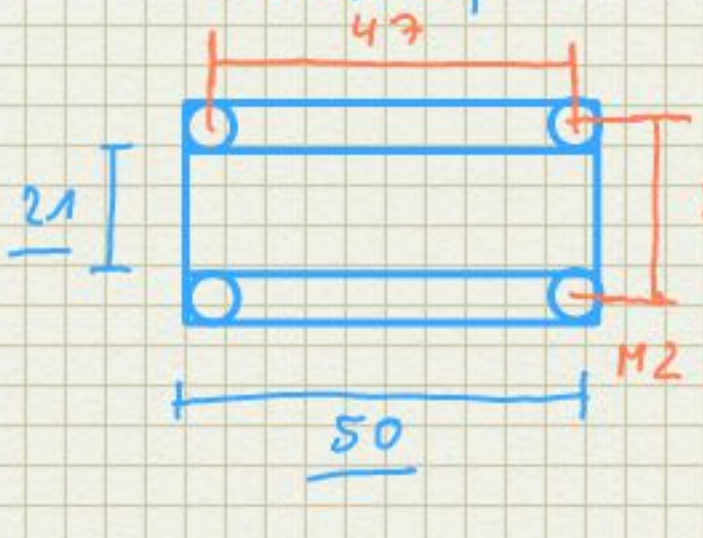


DAC MCP 4725

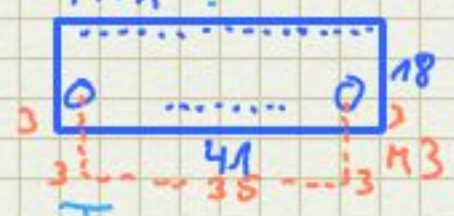


Joystick Sainsmart

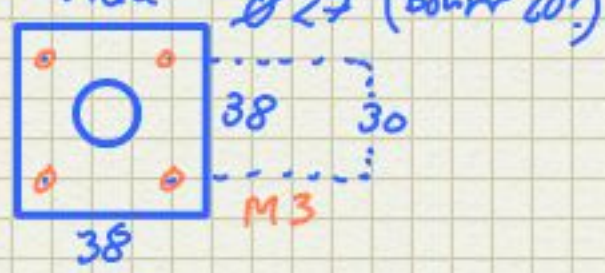
alphanumeric Display

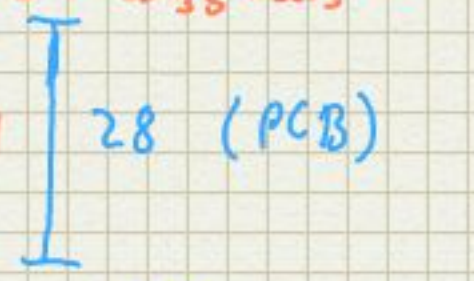


MUX

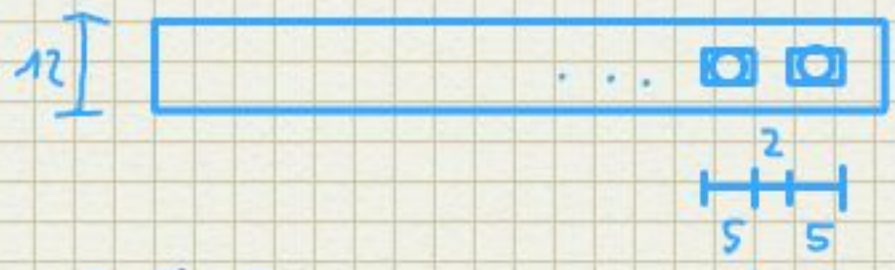


Adafruit Ada

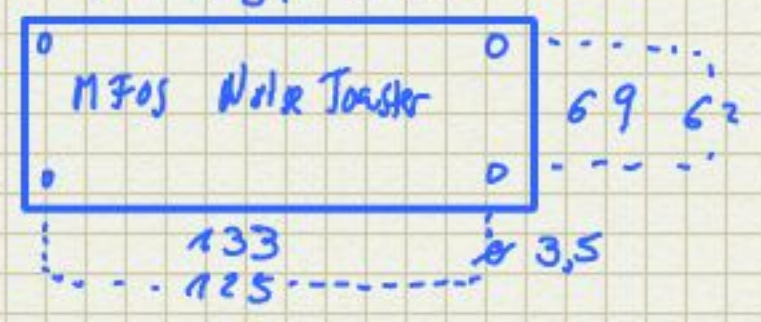




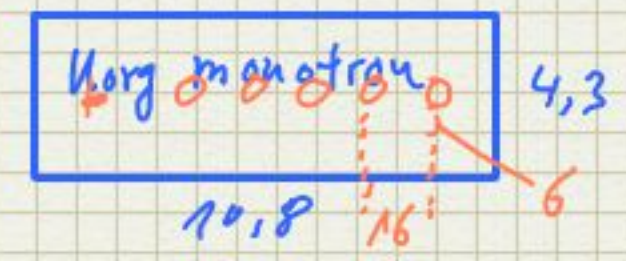
Neopixels 144 /m



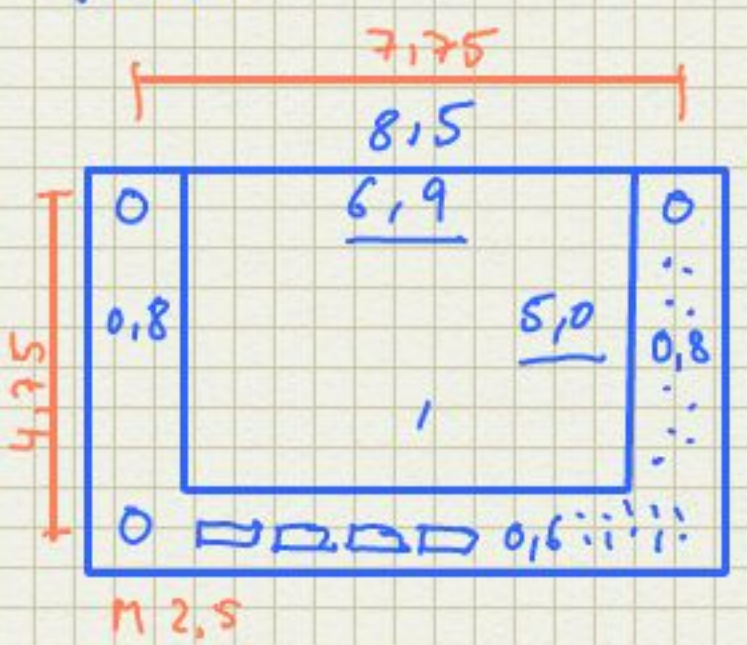
MOS 3Ch.-Amp.



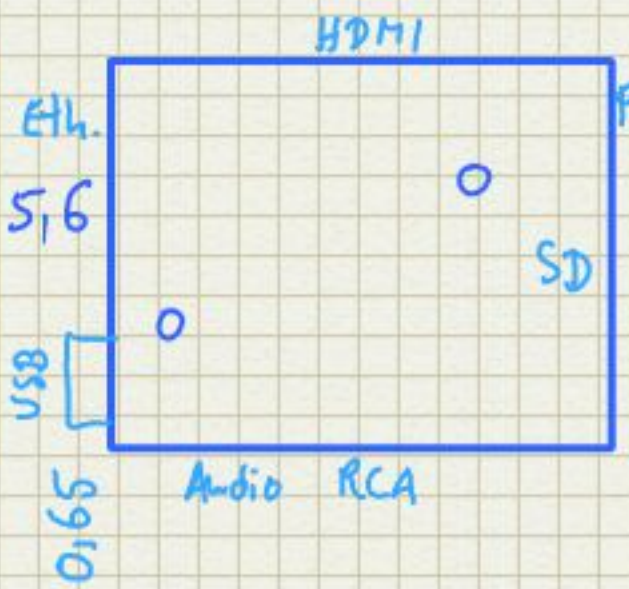
Korg monotron



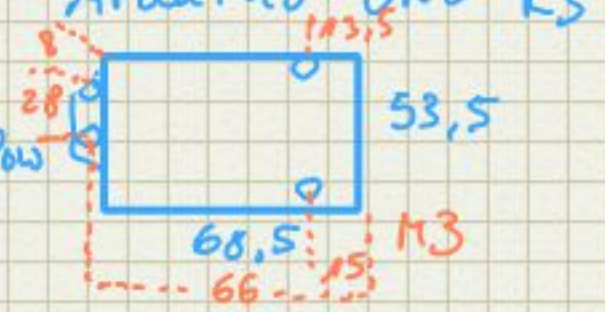
PiTFT



Raspberry Pi

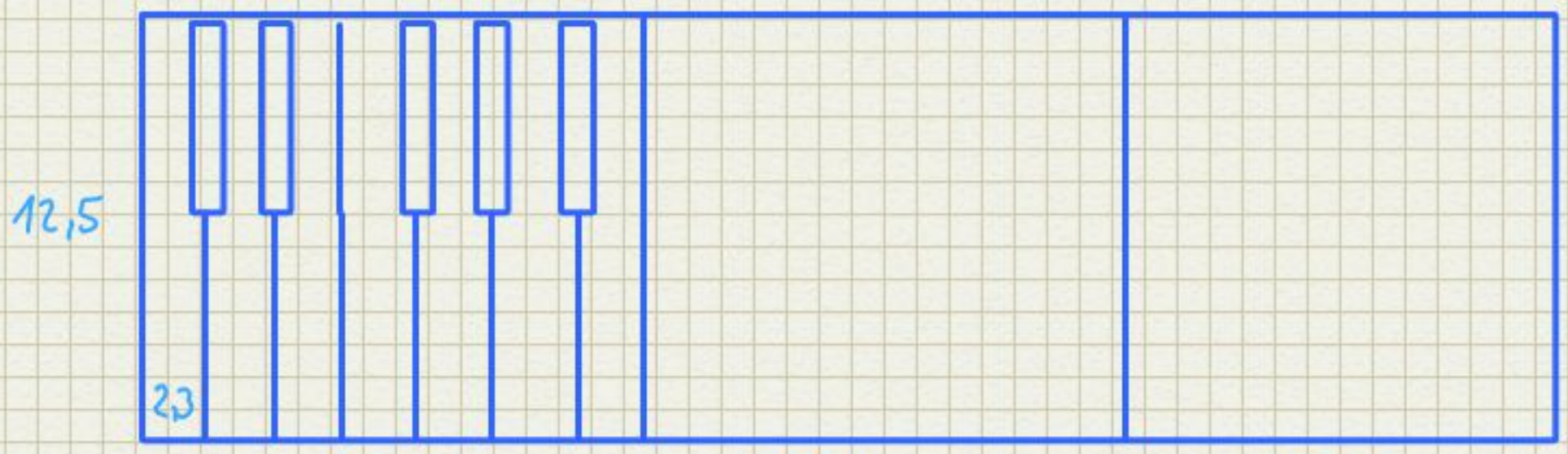


Arduino UNO R3

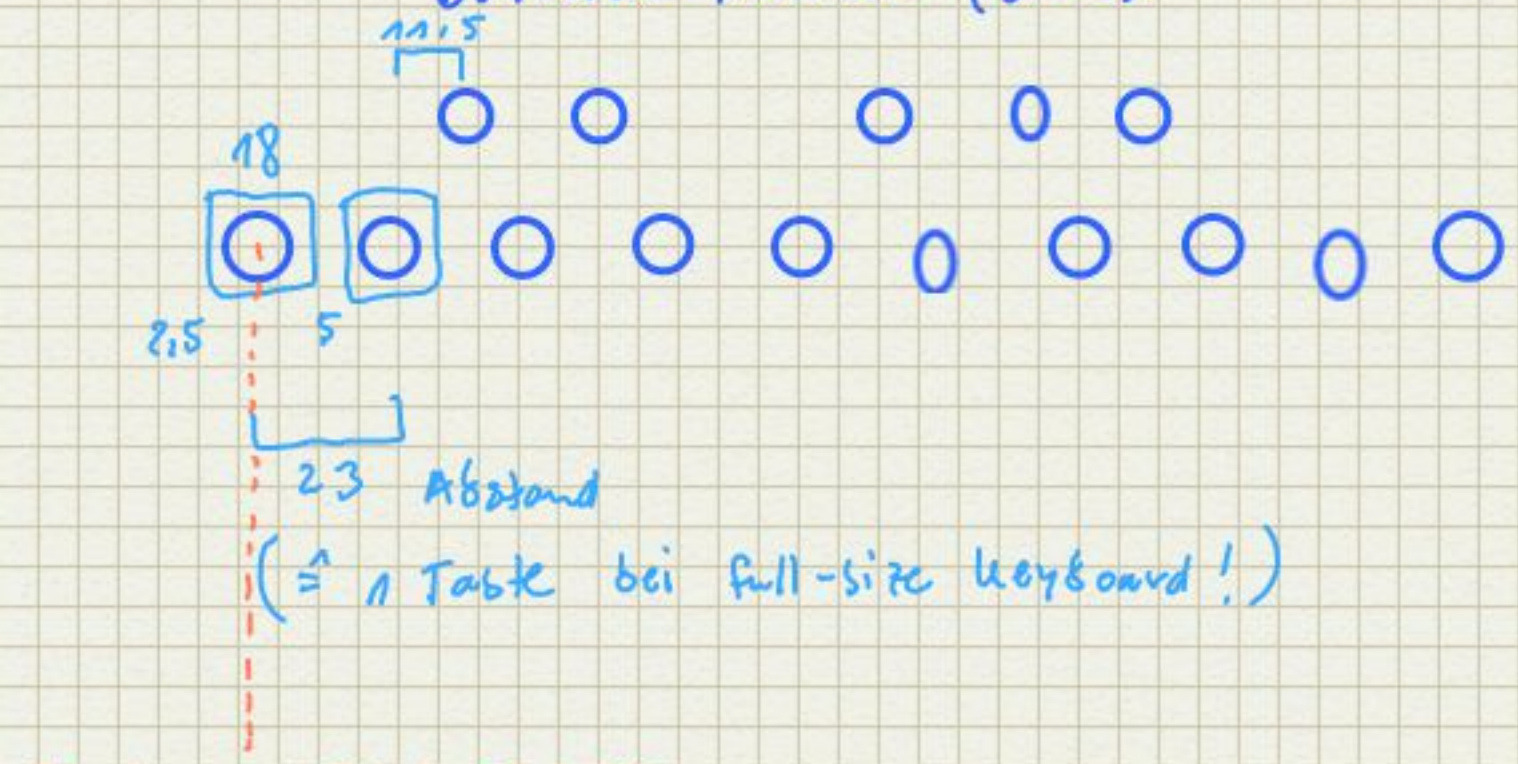


Arduino Nano V3

Tastatur / Keyboard (full-size)

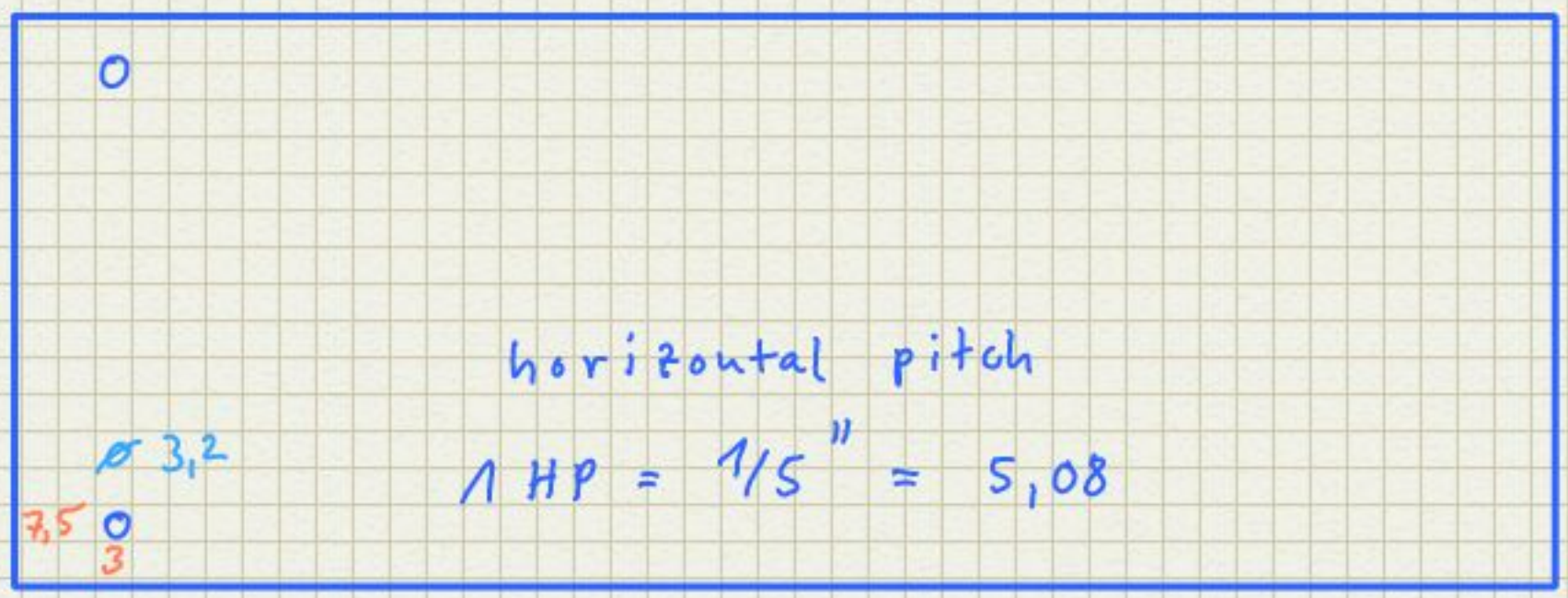


BUTTON-ARRAY (Muxer für Switches, Shift-Register für LEDs)



Döpfer (Case / Front - Panel / Rack - Mount)

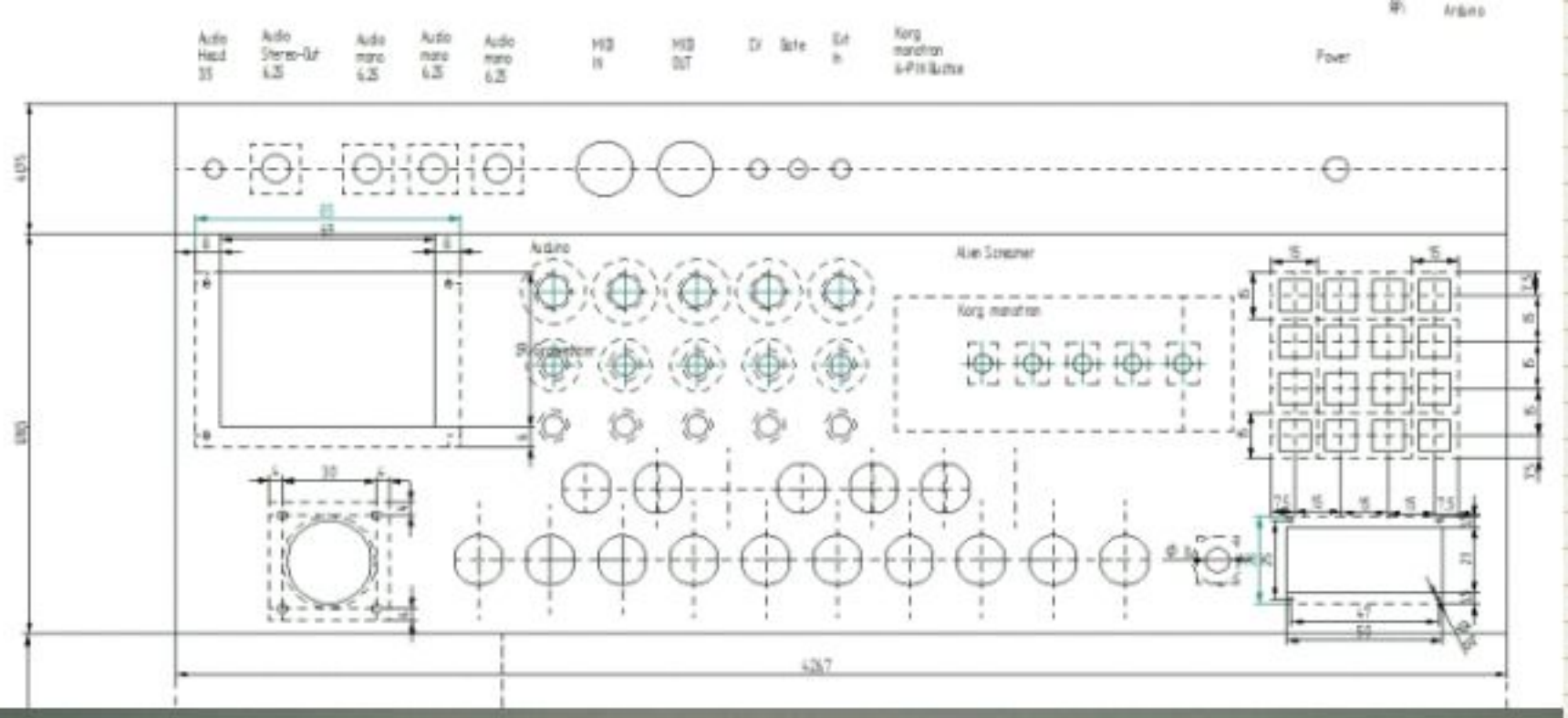
vgl. JimiBox (Schneiders Laden)

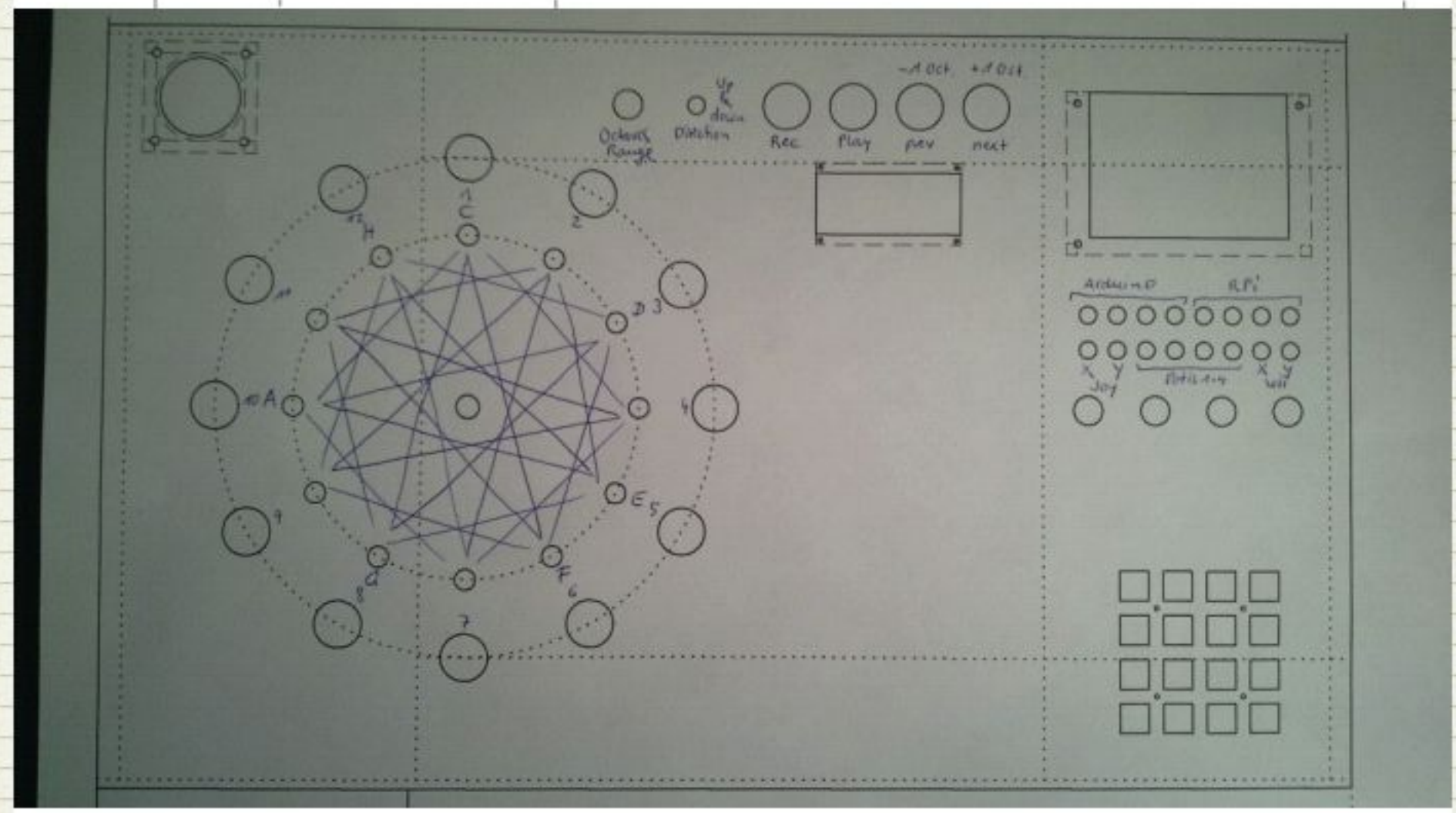


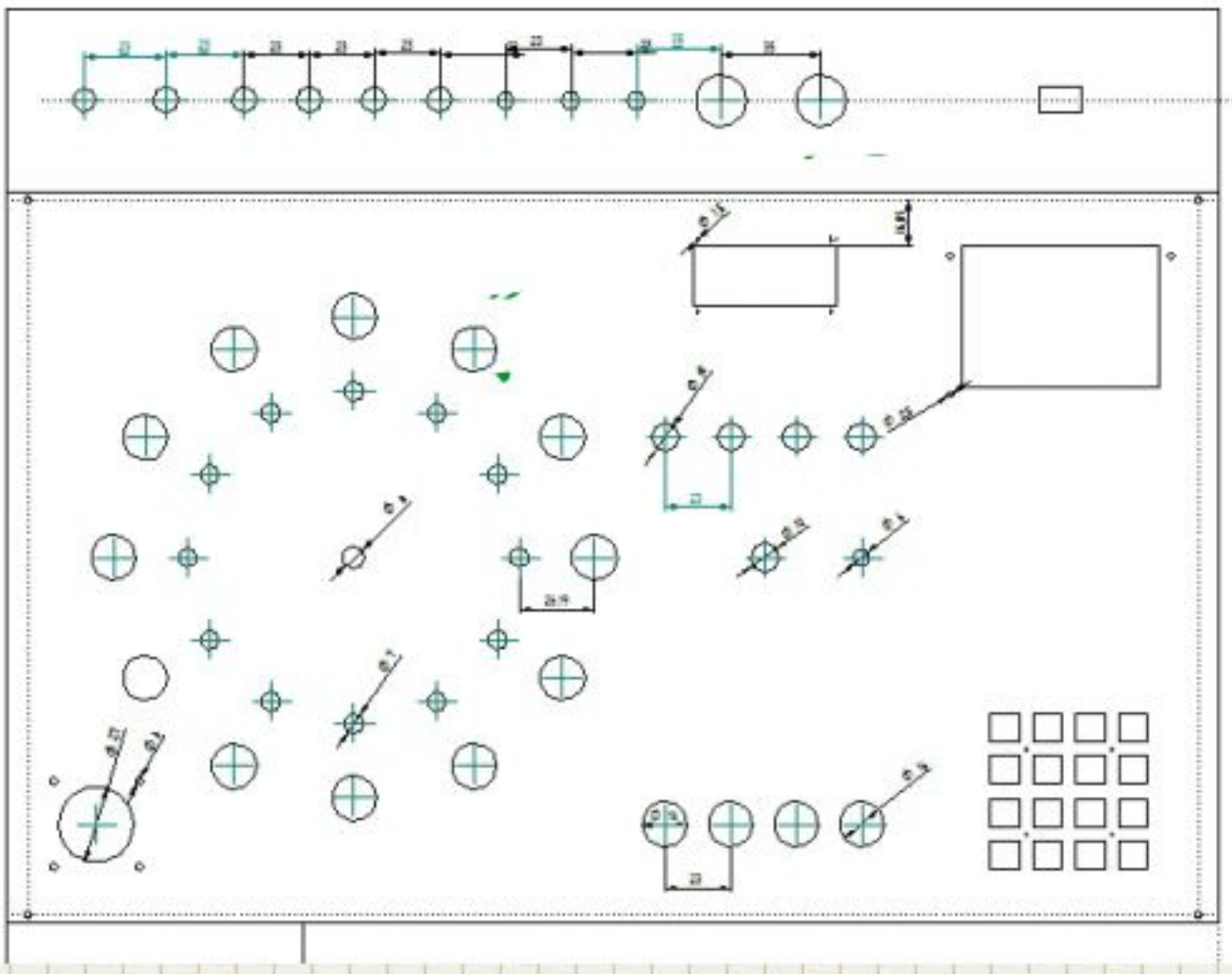
426,72

(= 84 HP)

19" - Rack


Audio Head 35
Audio Stereo-Out 6.35
Audio mono 6.35
Audio mono 6.35
Audio mono 6.35
MIDI IN
MIDI OUT
DI
Gate
Git In
Korg monotron 4-Pin Buchse
Power
Audio
Alien Screamer
Korg monotron
Sequenzer


Octave Range
Pitch
up
down
Rec
Play
prev
next
-1 Oct
+1 Oct
Arduino
RPi
Joy
Joy
1
2
3
4
5
6
7
8
A
E



I²C 7 bit Addresses

Connectivity

Arduino 8A 14D

- Neopixels 1D
+ Rotary Encoder (3A) 2D
+ alphanum Display I²C 0x70
+ Trellis 0x70 - 0x77 I²C

8 Addr
8 Addr

analog → ⊕ DAC 2 Addr (A0) I²C
digital → ⊕ MUX 16 In 1A 4D
+ MIDI Tx/Rx
+ Audio PWM 1D
~ Arduino 4.7k 5A
- Groovesizer 10k 5A
- " Buttons 5D
- Joystick 2A 1D

Raspberry Pi 17D (?)

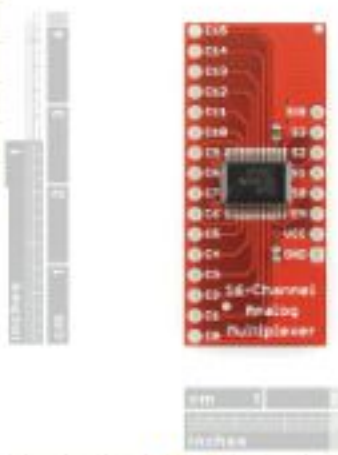
- LPD 8806 SPI
- PiTFT SPI 4D
#F 23 22 27 18
1110000 - 1110111

⊖ MCP3008 (ADC) SPI 8 In
⊕ Shift-Register 3D data latch clock
74hc595 x2
Σ 16 out

Arduino	A0 - A5	D0	1	2	3	4	5	6	7	8	9	10	11	12	13
		Rx	Tx		IRQ		~	~			~	~ SS	~ MOSI	MISO	SCK
	A4 (SDA) A5 (SCL)														PWM
	I²C/TWI				I²C				SPI			Tx	Rx		
Raspberry Pi	P1-01 bis 26			2	3										27
		0	1	4	7	8	9	10	11	14	15	17	18	21	23, 24

UART	Rx	GPIO 15
	Tx	GPIO 14
I²C	SDA	2
	SCL	3
SPI	MOSI	10
	MISO	9
	SCLK	11
	CE 0	8
	CE 1	7

?



LED 3 PWM
2 D

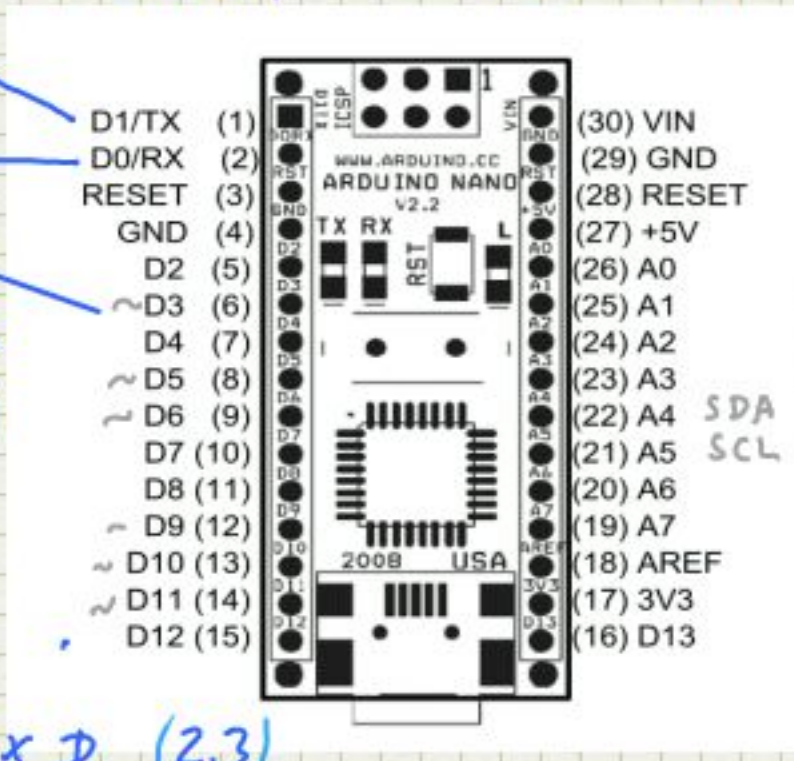
I²C

(A6,A7) A/D (MUX)

3 D
4 D
1 A (A2)

I²C/TWI
(A4) SDA
(A5) SCL
(A3) INT

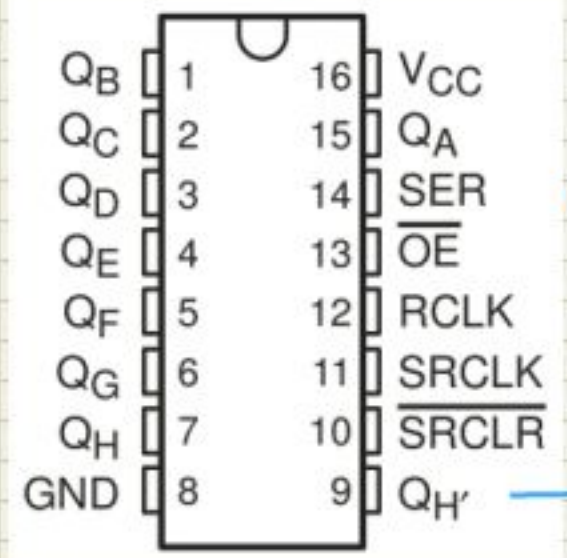
Out - MIDI — Tx (1)
In - MIDI — Rx (0)



Arduino 1 PWM
CV out — DAC-I²C
Gate out — D (13)
CV In — A ( )
Gate In — D ( )

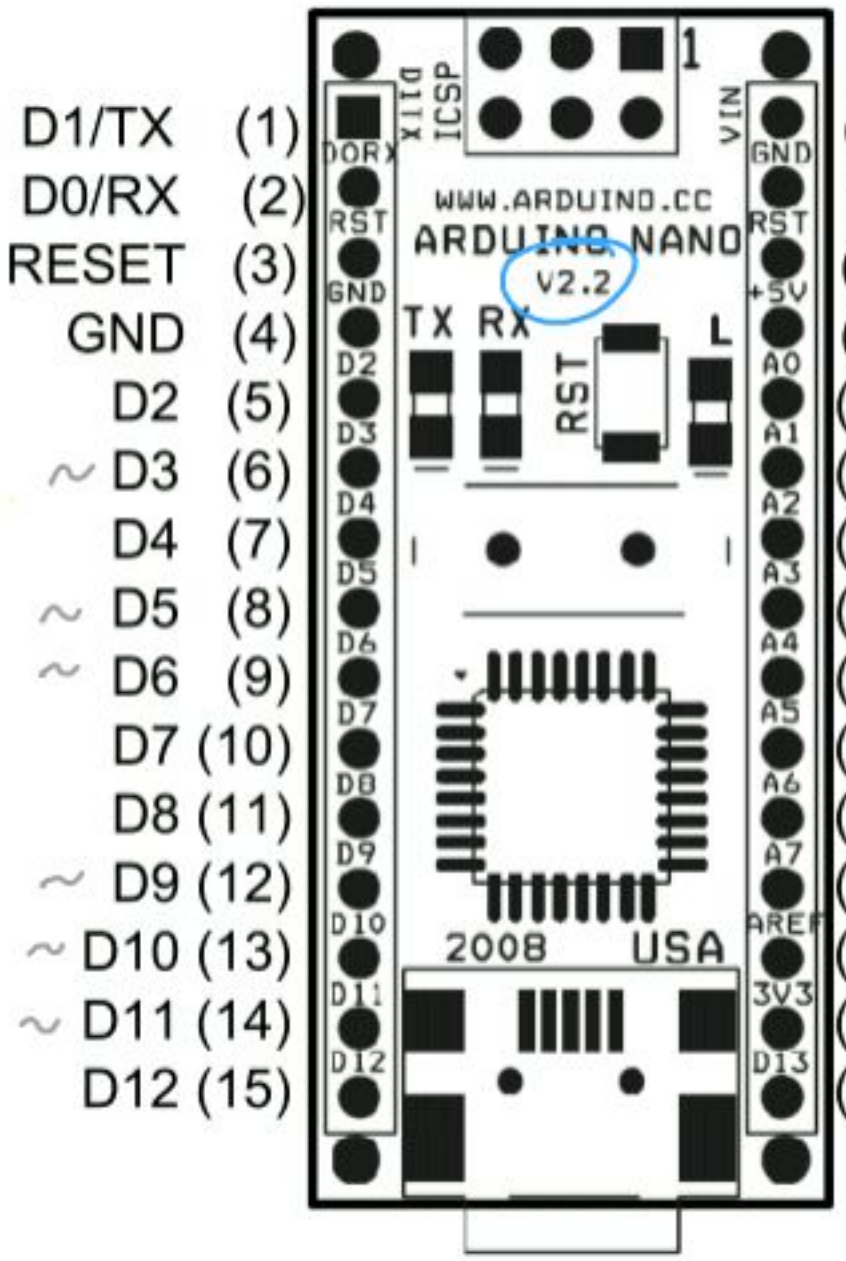
TWI/
I²C: DAC 0x60
Display 0x70
Trellis 0x72

Rotary Encoder - 2x D (2,3)
Multiplexer — 4x D + 1x A (4,5,6,7) (A2)
Shift Register — 3x D (8,9,10)
74hc595



5V
data
GND
latch clock
5V
SER of next IC

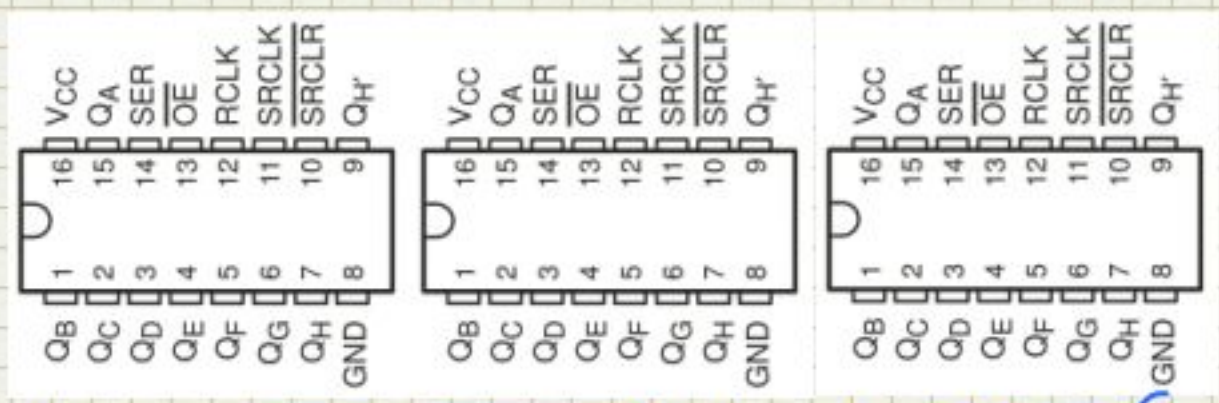
Shift Register 74HC595



shift SER
Audwino PWM
shift SCLK
RGB-LED R
RGB-LED G
shift RCLK
Enc A
Enc B
Enc Push
RGB-LED B
MIDI-In

SIG MUX-1
SIG MUX-2
X Joy
Y Joy
SDA I²C
SCL
DAC 0x60
DSP. 0x70
Trellis 0x72
MIDI-OUT
LED

(Enc Push)
Joy Push
Gate



Shift registers 3X

Button-LEDs 14

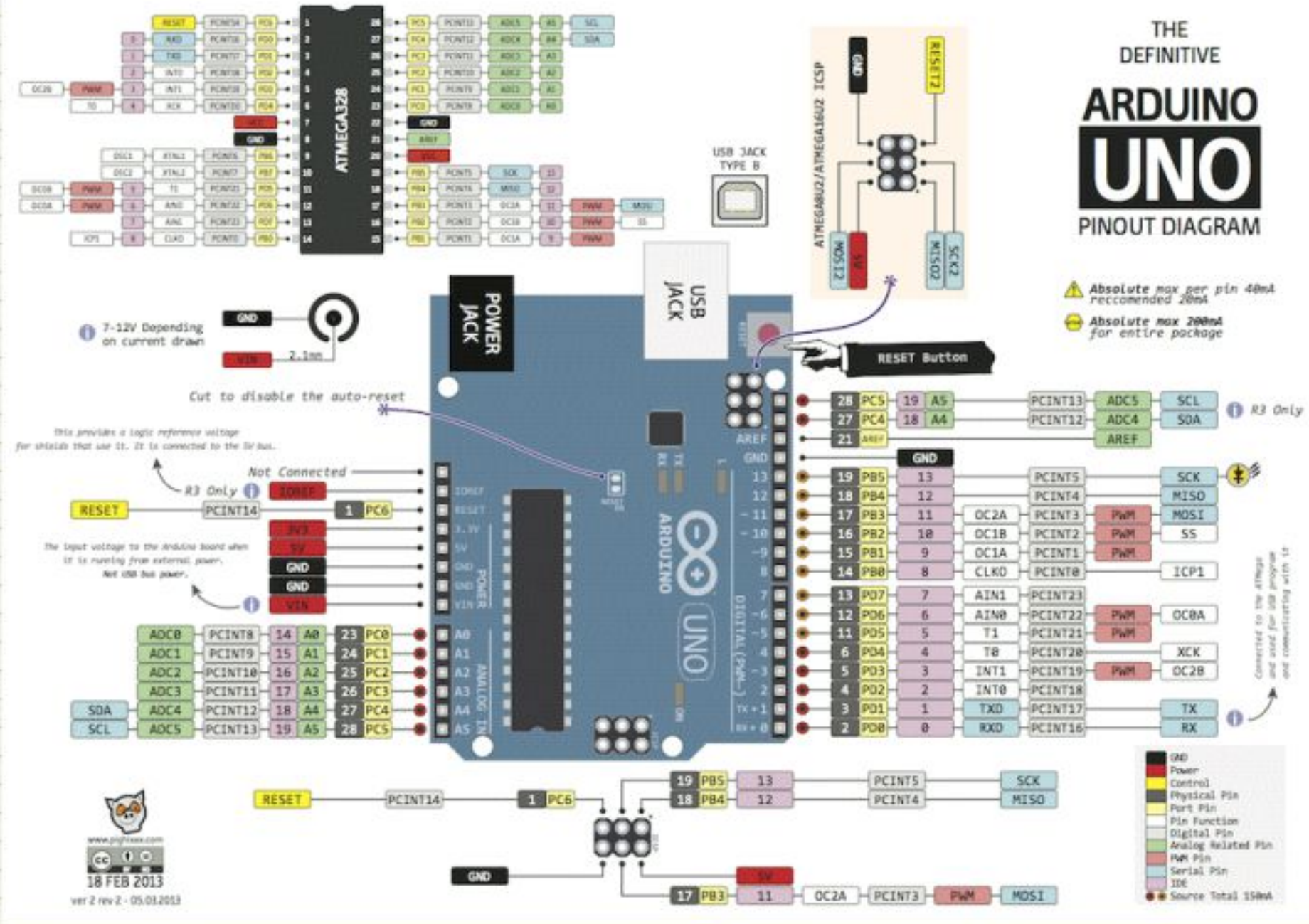
? Gate

MUX 1
MUX 2

MUX 2X

Button Array 16

Potis 16


THE
DEFINITIVE
ARDUINO
UNO
PINOUT DIAGRAM
ATMEGA328
ATMEGA8U2/ATMEGA16U2 ICSP
USB JACK
TYPE B
POWER
JACK
USB
JACK
RESET Button
Cut to disable the auto-reset
7-12V Depending
on current drawn
2.1mm
Absolute max per pin 40mA
reccomended 20mA
Absolute max 200mA
for entire package
R3 Only
Not Connected
This provides a logic reference voltage
for shields that use it. It is connected to the 5V bus.
The input voltage to the Arduino board when
it is running from external power.
Not USB bus power.
Connected to the ATMega
and used for USB program
and communicating with it
ARDUINO
UNO
DIGITAL (PWM~)
ANALOG IN
POWER
GND
Power
Control
Physical Pin
Port Pin
Pin Function
Digital Pin
Analog Related Pin
PWM Pin
Serial Pin
IDE
Source Total 150mA
18 FEB 2013
ver 2 rev 2 - 05.03.2013

SPI (18)(22)(23)(27)
(8)(9)(10)(11)

MIDI In — (15)

MIDI Out — (14)

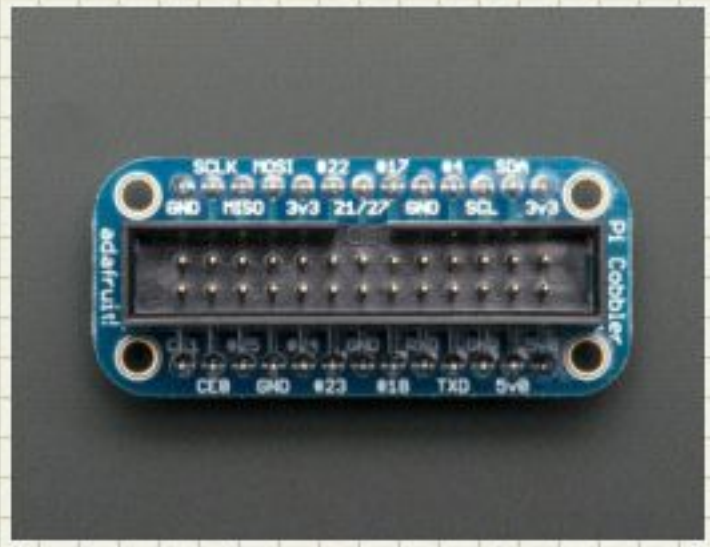
8 Potis

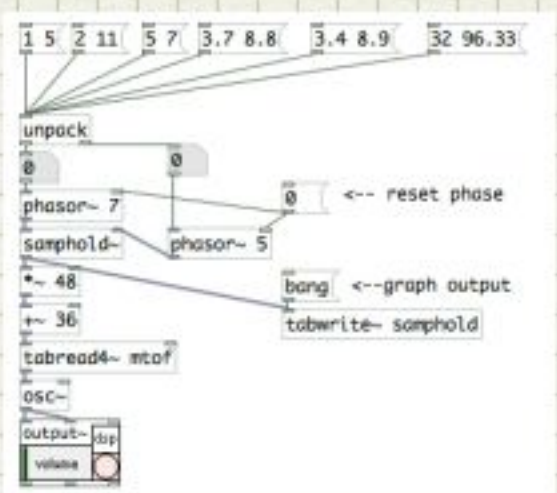
\

MCP3008 — (ADC)

DAC — (MCP4725)

Figure 8-27. Das wiringPi Pin-Schema (basierend auf dem Rev 2 Layout)





Elektronische Module

Arduino:
- Dreh- Encoder
- alphanum Display
- Trellis
- Mux <-- Button- Array
- Shift Register --> LEDs Button- Array
- Tx/ Rx <-- MIDI
- USB --> Raspberry Pi (Serial.Beginn)

- 5 Potis 4.7k (Auduino)
- 5 Potis 10k + 5 Buttons (Groovesizer)
- A1+2 <-- Joystick --> D1
- PWM --> Audio Out --> MFOS Amplifier

- DAC --> Korg monotron | MFOS Alien Screamer

Raspberry Pi:
- MIDI <-- Keyboard ()
- Creative Soundblaster
- BT <-- Wiimote | Rii
- WLAN <-- TouchOSC

- analog in MCP3008 + 8 Drehpotis
- PD, libpd (headless)
- Pygame (MIDI, Wav)
- DAW + VST
- Audacity

- Sample Library
- Waveform Library
- Soundfonts sf2

1) Sensor Controller
4) Auduino | Groovesizer
2) MIDI 2 CV
3) Sequencer : 303 | TR - Mode
   Sequencer : Step (Poly 800)
   Arpeggiator
   Ableton Op 100 { Transpose
   { Intro, Refrain, ... (Patterns)

Wii , TouchOSC

Drumkits , Samples , Strings

PD

Arduino Programmer

Recording

Power:
- 5V Raspberry Pi (3.3V logic level) --> USB --> Arduino
- USB | 5V (Pin 27) | 7 - 12V (Pin 30) Arduino (5V logic level)

- 9V MFOS Alien Screamer
- 3V Korg monotron

Analog Sequencer (8 Steps)

MFOS Noise Toaster
MFOS Module

Doepfer Module

Sequencer

Arpeggiator (Sequential Prophet 6)
Step (Poly800)
analog SQ1, Mini Baby 6
TR
Pattern Intro, ..., end
Song-Mode

1) MIDI-Clock In/Out
2) Note On, off, Velocity, Channel, CC
   ‖ Grid, Quantization, smalest Step/Timecode ‖
   ‖ Datenstruktur ‖
3) LUT für Tonleitern (Scales)
   Akkorde / Arpeggios
   (Obertöne, Quintenzirkel)
4) Swing
5) Conversion MIDI-Note to CV/Frequency
   CV-In to MIDI-Note-Number
6) Conversion Bpm to Time

Arduino Bedienmenü für Controll-Data:

0 - Einstellungen; u.a. MIDI, CV, OUT
1 - Arpeggiator
2 - Step-Sequencer
3 - TR-Mode (Looping, Live)
4 - Pattern Mode
5 - Song Mode: left shift + Note → Transpose
right shift + Note → Part | Pattern
6 - Arduino

Voices:

a) (MFOS Alien Screamer)
b) Korg monotron Delay
c) PD: 303, ...
Drum-kit
Strings
} VST, Samples, sf2
- library
- sfx
- Audacity
d) Arduino
e) ext-In → Raspberry Pi

→ Amp

Piano: anschlagdynamisch, polyphon
Strings
Bass: analog
Acid: 303
Drums
Sampler

@ see

Korg MS-20 M (new)

Roland TB-303
Roland TR-808

Korg Poly 800

Sequential Circuits (Dave Smith) Prophet 6

Arturio MicroBrute

Novation BassStation 1 + 2 (incl. Arpeggiator)

Yamaha QY100

Kirnu (VST-plugin) ARPEGGIATOR
Band-in-A-Box (Algrithmische Komposition)
PD
Logic : Arpeggiator ?
Ableton Live : Live Loops per Instrument statt Patterns
mit automatischen Timestretching bei Samples

Plasma Sound (Android) Source Code

Open Source : ... Caustic (Android)
LMMS ... (kleinere MIDI-Tools,
MIDI-Arpeggiators, MIDI-FX...)

MIDI

31250 baud

Status Byte + Data Byte + Data2 Byte
Status Bytes are > 127 (first bit 1)
Data Bytes are < 128 (first bit 0)

Channel Voice Messages

8n kk vv Note- Off
9n kk 00 Note- Off
9n kk vv Note- On

An kk vv Aftertouch (Polyphonic Key Pressure)
Bn cc vv Control Change
Cn pp Program Change
Dn vv Aftertouch (Channel Pressure)
En ll mm Pitch Bend Change: 14 Bit Value

n: MIDI-Channel {0-15}
kk: Note- No. {48-108}
vv: Velocity {normal: 40; off: 00}
cc: Controller No. {0-119}
vv: Controller Value {0-127}

ll: Least Significant Bit (LSB)
mm: Most Significant Bit (MSB)

Channel Mode Messages (cc 120-127)

Bn 78 00 All Sound Off
Bn 7B 00 All Note Off

System Realtime Messages

F8 Timing Clock: 24 times per quarter note
FA Start
FB Continue
FC Stop
FE Active Sensing: sent every 300ms (max) to keep connection to receiver alive

vgl.
MIDI Clock
MIDI Time Code (MTC)
MIDI Time Code Quarter Frame

System Common Messages
Universal System Exclusive Message (Non Realtime)

F0 Exclusive Status
7E Non Realtime Message
...
F7 End Of Exclusive

Obertöne

Grundton	1.	2.	3.	4.	5.	6.	7.
	C	C	G	C	E	G	B	C

DUR →

Untertöne

Grundton	1.	2.	3.	4.	5.	6.	7.
	C	F	C	As	F	D	C

← Moll

Frequenz-verhältnis	1/2	1/3	1/4	1/5	1/6	1/7	1/8

Bsp. Kammerton
A (440 Hz): 220 147 110 ... [Hz]

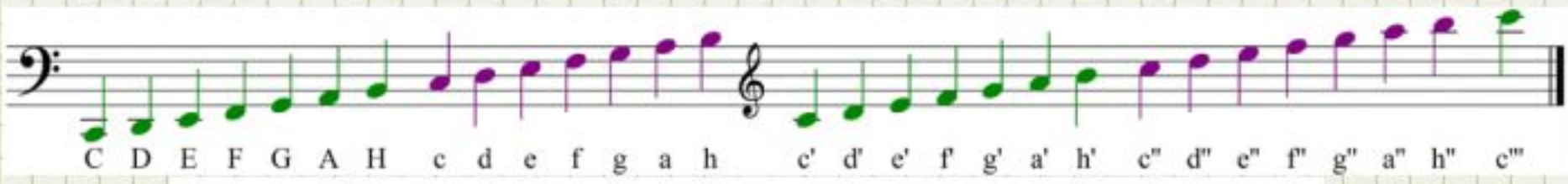
Intervalle

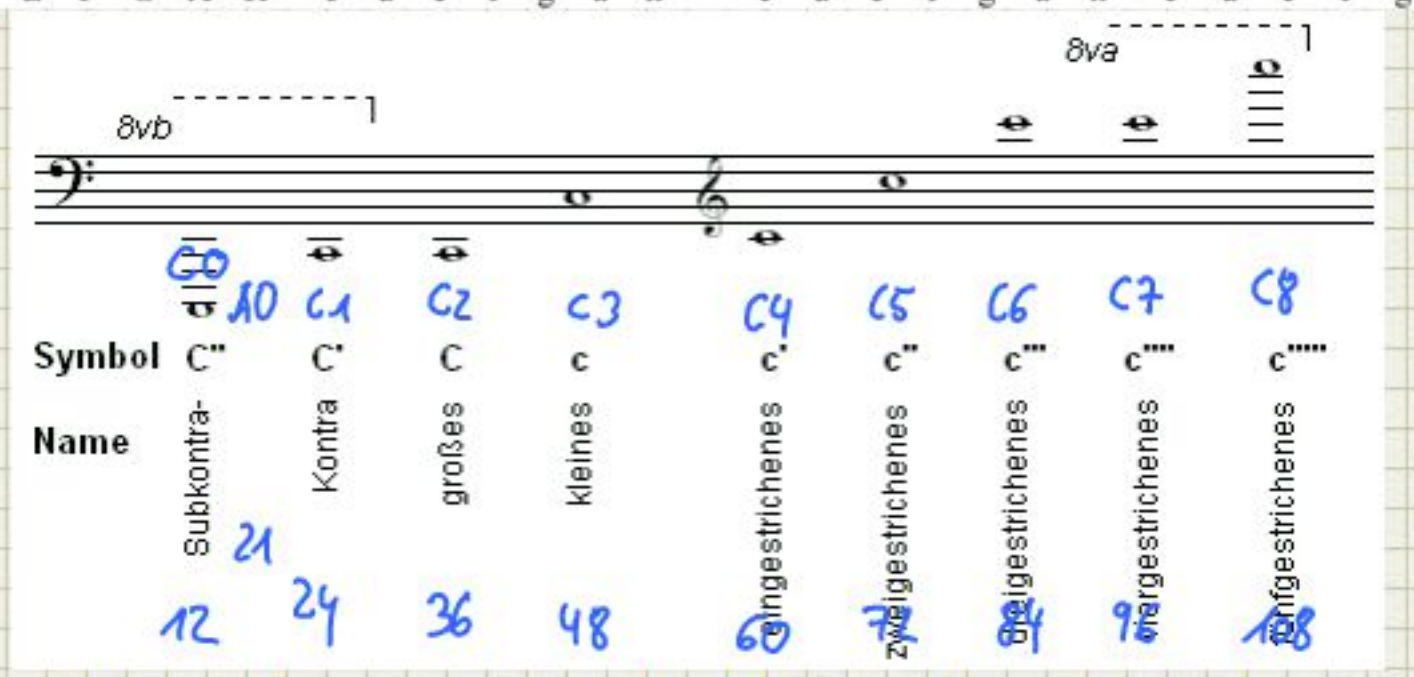
sus 4	Undezime		e'16	
sus 2	None		d'14	
	Oktave	6 C	12	
		B	11	gr. Sep.
	kl. Septime	5	10	
		A	9	gr. Sexte
	kl. Sexte	4	8	
		G	7	Quinte
#4/b5	Tritonus	3	6	
		F	5	Quarte
DUR	gr. Terz	2 E	4	
			3	kl. Terz
	gr. Sekunde	1 D	2	
			1	kl. Sek.
	Prime	0 C		

Step-gattern der Tonleiter (hier: C-Dur)

maj 7	7
	b
6	6
	b
D7	5
	b
	4
moll	3
	b
	2
	b
	1

@see Wikipedia Oktave - - > Quellenangabe





Klavier	Klavier 1. Taste	...	mittleres C'	Kammerton a'	...	Klavier 88. Taste
Note	A0		C4	A4		C8
Frequenz	27,5 Hz		261,6 Hz	440 Hz		
MIDI-Note-No.	21		60			108

Klavier 88 Tasten, Keyboard 61 Tasten, Synthe 49 Tasten
Bass-Synth 24-49

Rhythmus / Timing

MIDI Clock (Master / Slave): 24 Stps per Quarter Note

Bpm to delay / wait [ms]

$$\text{delay [ms]} = \frac{\text{Notenwert} * 240.000}{\text{Tempo [Bpm]}}$$

Bsp.: 1/16 Note @ 120 Bpm = 125 ms
1/4 @ 120 Bpm = 500 ms
1/16 @ 130 Bpm = 115,4 ms

Scales (Tonleitern)

Tonleiter der Pentatonik (5 Töne)

Ganztonleiter (Hexatonik, 6 Töne), Blues Tonleiter

Tonleitern der Heptatonik (7 Töne): DUR-, Moll-Tonleitern, Kirchentonarten

verminderte Skala (Oktatonik, 8 Töne)

Chromatische Tonleiter (12 Töne)

PD [mtof] : 8.1758 * exp (X * 0.0577623)

---

diatone (harmonische) Tonleiter

Faktor * 262 Hz * 2^Oktave

Oct. * (1V/oct.) + Note * 0.083

Oct * 12 + Note

	Faktor		Faktor [1V/Oct]	MIDI Note-No.	Frequenz [Hz]	Note
C	1			72	262	0
#		16/15	0.083	73	277	1
D	9/8		0.166	74	294	2
#		6/5	0.249	75	311	3
E	5/4		0.333	76	330	4
F	4/3		0.416	77	349	5
#			0.499	78	370	6
G	3/2		0.583	79	392	7
#		8/5	0.666	80	415	8
A	5/3		0.749	81	440	9
#		9/5	0.833	82	466	10
B	15/8		0.916	83	494	11
C	2		1	84	523	12

Sub-Kontra	C_1	C	c	c'	c''	c'''	c''''	c'''''	
C0	C1	C2	C3	C4	C5	C6	C7	C8	
12	24	36	48	60	72	84	96	108	120
Oct 1	2	3	4	5	6	7	8	9	10

Scales (Tonleitern)

chromatisch (12 Halbtöne)
Major (DUR) ionisch
reines Minor (Mol) aeolisch
Harmonic Minor
Melodic Minor
Minor Pentatonic
Major Pentatonic
Blues 1, b3, 4, b5, 5, b7
Diminished

Modes
C Ionian (= C major)
D Dorian
E Phrygian
F Lydian
G Mixolydian
A Aeolian (= A natural minor)
B Locrian

Chords (Akkorde)

Dur-Familie C C^{6} C^{add9} C^{maj7} $C^{maj7\#5}$ C^{maj7b5} C^{maj9} $C^{6/9}$ $C^{maj7/13}$ $C^{maj9\#11}$ C^{maj9b5}

Moll-Familie Cm Cm^{7} Cm^{maj7} Cm^{6} Cm^{add9} Cm^{9} $Cm^{maj7/9}$ $Cm^{6/9}$ $Cm^{7/11}$ Cm^{11}

Dom^{7}-Familie C^{7} $C^{7\#5}$ C^{7b5} C^{7sus4} C^{9} C^{7b9} $C^{7\#9}$ $C^{7b9\#5}$ $C^{7\#9\#5}$ $C^{7sus4/13}$ C^{9sus4} $C^{9\#11}$ C^{13} $C^{7b9/13}$

halbvermindert Fam. Cm^{7b5}
vermindert Fam. C^{o} C^{o7}
weitere Dreiklänge C^{+} C^{sus4} C^{sus2}

Die Akkordsymbole und ihre Bezeichnung

Beispiele in C

Akkordsymbole	Akkordbezeichnung	Alternativbezeichnung
C	Dur-Dreiklang	
Cm	Moll-Dreiklang	c, C-, Cmi, Cmin
C^7	Dominantseptakkord	
C^6	Dur-Dreiklang mit großer Sexte	
C^9	Dur-Dreiklang mit großer None	
C^o	Verminderter Akkord	Cdim, Cverm, $Cdim^7$
C^{+5}	Dur-Dreiklang mit übermäßiger Quinte	C^{5+}, C^+, $C^{5\#}$, Caug
C^{-5}	Dur-Dreiklang mit verminderter Quinte	C^{5-}, C^-
C^4	Dur-Dreiklang mit Quartvorhalt	Csus, $Csus^4$
$C^{7/4}$	Septakkord mit Quartvorhalt	C^7sus, C^7sus^4
$C^{7/6}$	Septakkord mit übergelegter Sexte	
$C^{7/9}$	Dur-Dreiklang mit kleiner Septime und großer None	C9

Akkordsymbole	Akkordbezeichnungen	Alternativbezeichnung
$C^{6/9}$	Dur-Akkord mit großer Sexte und großer None	
$Cmaj^7$	Dur-Akkord mit großer Septime	Cj^7, $C^{7\natural}$, $C_\triangle$, $C_\triangle 7$, CM, CM^7
$Cmaj^{7/9}$	Dur-Akkord mit großer Septime und großer None	$Cj^{7/9}$, Cmaj 9, $C^{7\natural/9}$ $C_\triangle 7/9$, $CM^{7/9}$
Cm^6	Moll-Akkord mit großer Sexte	
Cm^7	Moll-Akkord mit kleiner Septime	Cmi^7, $Cmin^7$
$Cm^{7/9}$	Moll-Akkord mit kleiner Septime und großer None	$Cmi^{7/9}$, $Cmin^{7/9}$, Cm^9
Cm^{7-5}	Moll-Akkord mit kleiner Septime und verminderter Quinte	$Cm^{7/5(b)}$, Cm^{7b5}
C^{7+5}	Septakkord mit übermäßiger Quinte	C^{+7}
C^{7-5}	Septakkord mit verminderte Quinte	$C^{7/5b}$
Cm^{maj7}	Moll-Akkord mit großer Septime	$Cm^{(\#)7}$, $Cm^{7\natural}$, $Cm_\triangle$, $Cmi_\triangle 7$
$C^{7/9-}$	Septakkord mit kleiner None	$C^{7/-9}$, C^{9-}, C^{9b} $C^{7/9b}$
$Cm^{7/9-}$	Moll-Akkord mit kleiner Septime und kleiner None	$Cm^{7/-9}$, $Cm^{7/9b}$


YAMAHA QY20 MUSIC SEQUENCER
MODE
ENTER
MENU
−1
NO
+1
YES
OCT DOWN
OCT UP
BASS
F#
Gb
G#
Ab
A#
Bb
C#
Db
D#
Eb
7(#9)
7(b9)
7(b13)
7(13)
7(#11)
7(#5)
7sus4
sus4
---
dim
C
C
E
F
A
B
C
D
add9
M
M9
M7
aug
6
9
7
add9
m
m9
m7
mM7
m6
INTRO
NORMAL
VARI.
FILL1
FILL2
ENDING

Hilfe seite

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B	C
	0	1	2	3	4	5	6	7	8	9	10	11	chromatische Tonleiter
		W		W	H	W		W		W	W	H	Step pattern
No. of notes	Ionian DUR (Major)	1		2		3	4		5		6		7	1	Formula
7	{0, 2, 4, 5, 7, 9, 11}	C		D		E	F		G		A		B	C	Notes
		W		H	W		W		H	W		W			
7	aeolian moll (Minor)	1		2	b3		4		5	b6		b7		1	
	{0, 2, 3, 5, 7, 8, 10}	C		D	Eb		F		G	Ab		Bb		C	
	harmonic Minor	1		2	b3		4		5	b6			7	(8)	
7	{0, 2, 3, 5, 7, 8, 11}	C		D	Eb		F		G	Ab			B	C	
		W+H			W		W			W+H				W	
	Jazz melodic Minor	1		2	b3		4		5		6		7	(8)	
7	{0, 2, 3, 5, 7, 9, 11}	C		D	Eb		F		G		A		B	C	
		W+H			W		W			W+H				W	
	Rock, Blues Minor Pentatonic	1			b3		4		5			b7			
5	{0, 3, 5, 7, 10}	C			Eb		F		G			Bb			
		W		W		W+H		W		W+H					
	Country Major Pentatonic	1		2		3			5		6				
5	{0, 2, 4, 7, 9}	C		D		E			G		A				
		W+H		W	W		H	W		W+H					
	Jazz, Rock, Blues Blues	1		b3		4	(b5)	5		b7					
	(≙ Minor Penta. + b5)	C		Eb		F	Gb	G		Bb					
6	{0, 3, 5, 6, 7, 10}														
		W	H	W	H	W	W	H							
	Jazz, Heavy Metal Diminished	1		2	b3		4	b5		b6	6		7		
8	{0, 2, 3, 5, 6, 8, 9, 11}	C		D	Eb		F	Gb		Ab	A		B		

natürliches a-moll

A		B	C	D	E	F	G	(A)
	W	H	W	W	H	W	W	
W+H								

ARRAYS and ARPEGGIOS

```
int[] scale = {0, 2, 3, 5, 6, 8, 9, 11}

for (int i=0, i < scale.length(), i++) {
    play (scale[i]);
    wait (ms);
```

Implementierung Beispiel Datenstruktur


```
int[] moll = {1,2,-3,4,5,-6,-7};
// ... andere Tonleitern und Akkorde definieren

int[] chromatic = int[14];

                  0  1  2  3  4  5  6 | 7 | 8  9  10 11 12 13 14
                  bb ba bg -  be bd -   -   c  d  e  f  g  a  b
int[] chromatic = {11, 9, 7, 0, 4, 2, 0, 0, 1, 3, 5, 6, 8, 10, 12};

int midiNote = (chromatic[moll[i]+7]-1)+(oct*12);

for (i=0; i < scale.length; i++) {
    int midiNote = ...
    play(midiNote);
    wait(del);
}
```

Quintenzirkel

Stufendreiklänge

I	II m	III m	IV	V	VI m	VII°
C	dm	em	F	G7	am	B°
T	Sp	Dp	S	D	Tp	

JVR 1 b2 b3 4 b5 b6 6 b7 8
C D EF G A BC D E
0 1 2 3 4 5 6 7 8 9 b1 1 0 1 2 3 4 5

g. Tut
1 3 5
Quart
3 5 8
1 63 b6
kl. Tut 5 8 10
1 4 6
gr. Tut

Grundstellung 1, b3, b6
1. Umkehrung (Sextakkord)
2. Umkehrung (Quartsextakkord) 1, 4, 6

Stufenvierklänge

I	II	III	IV	V	VI	VII
C maj7	Dm 7	Em 7	F maj7	G 7	Am 7	Bm 765

Vorzeichen

b S IV F d II Sp
bb B g
bbb Es c
As f
b Des

T I C a VI Tp
D 7 # Fis V
G 7 e III Dp
D h
fis A
cis E
gis
dis/es
Fis/Ges
H° VII

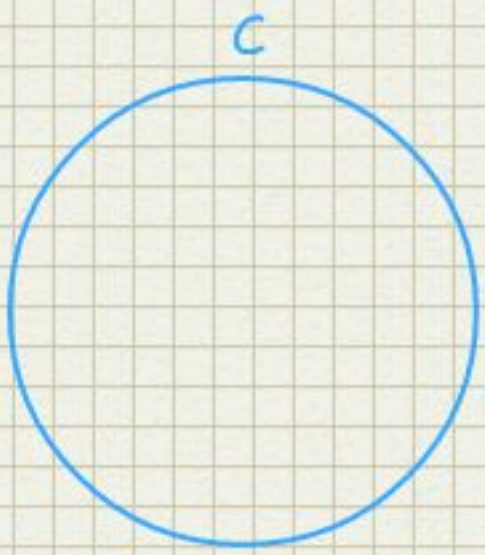
Tonart

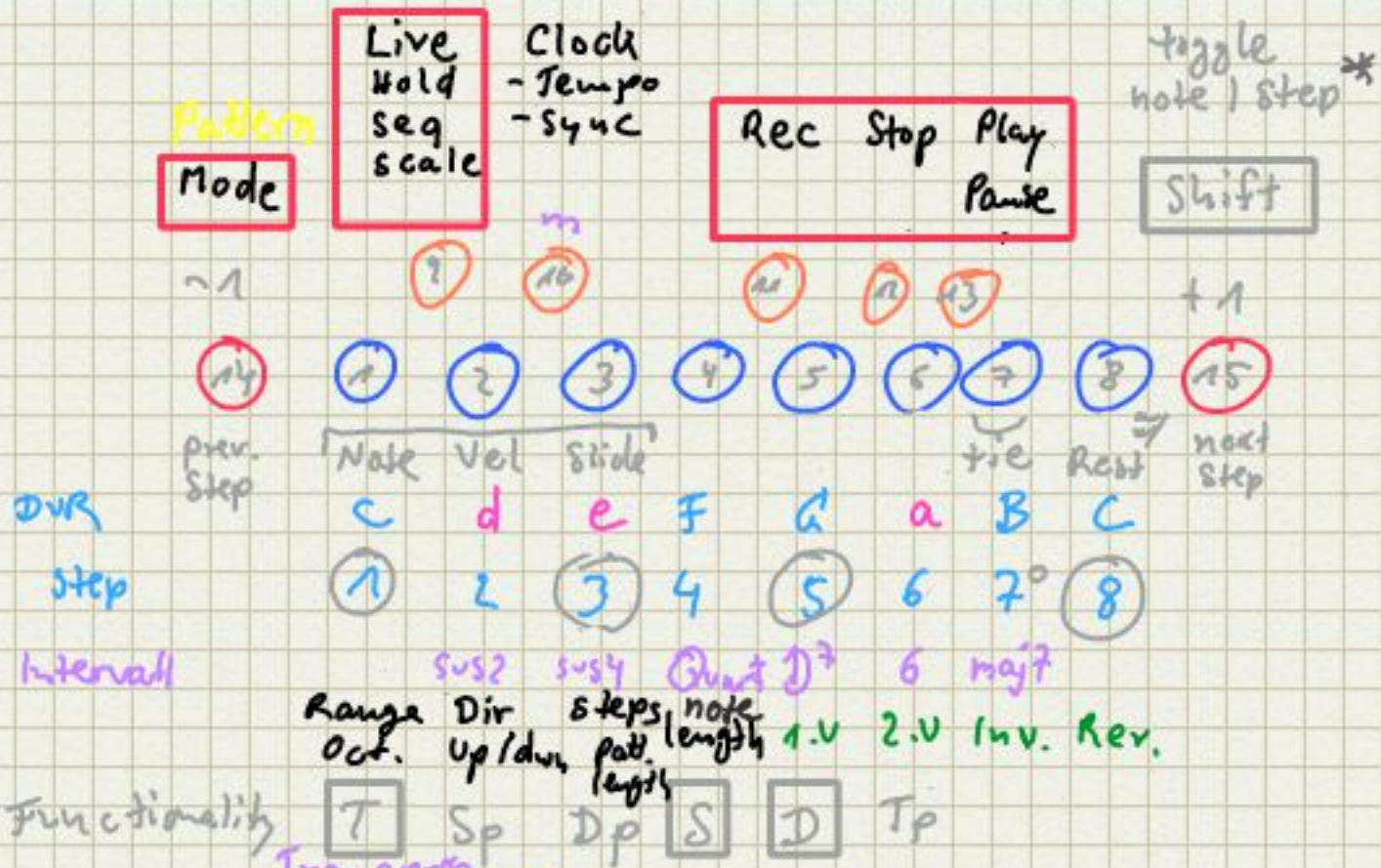
Dur | moll

Stufe

I IV V I

Kadenz: T - S - D - T


C



Speicher: Datenstruktur

8 Patterns a'

8 Beats max.

(= 32 steps je Pattern)

[Quantisierung 1/16]

je 3 Ebenen:

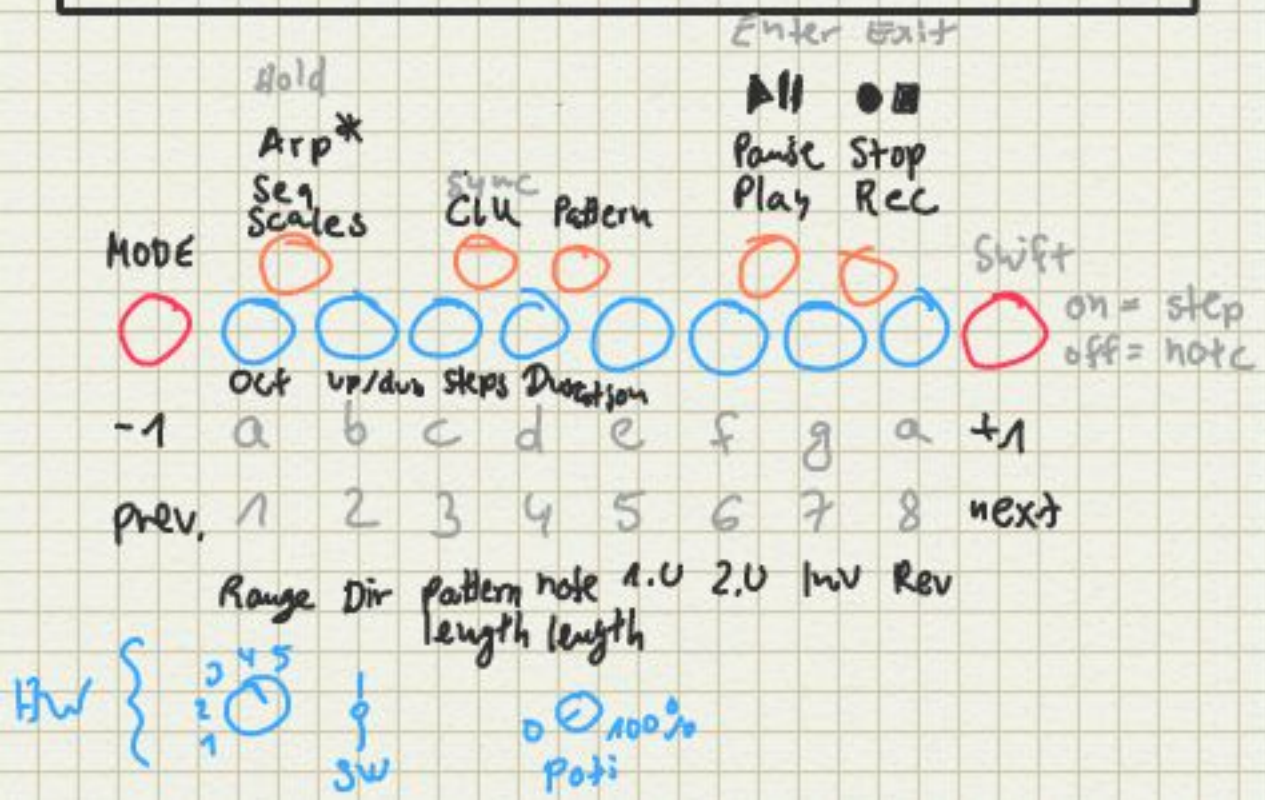
- Note | Pitch
- Velocity | Accent
- Slide | Portamento | legato

SRAM: 8 * 32 * 3 = 768 = 84 B

PROGMEM

* Toggle note / step:
Eingabe von Zahlen, wenn Button leuchtet.
Ziffern 0, 9 !

Text soll, wenn er länger als 4 Buchstaben ist, aus dem Display herauslaufen – gefolgt vom Parameter value, der dann im Display sichtbar bleibt.
Der ausgewählte button leuchtet. Der Wert wird über den Encoder eingestellt
(oder Button-Array ? oder trellis ?)



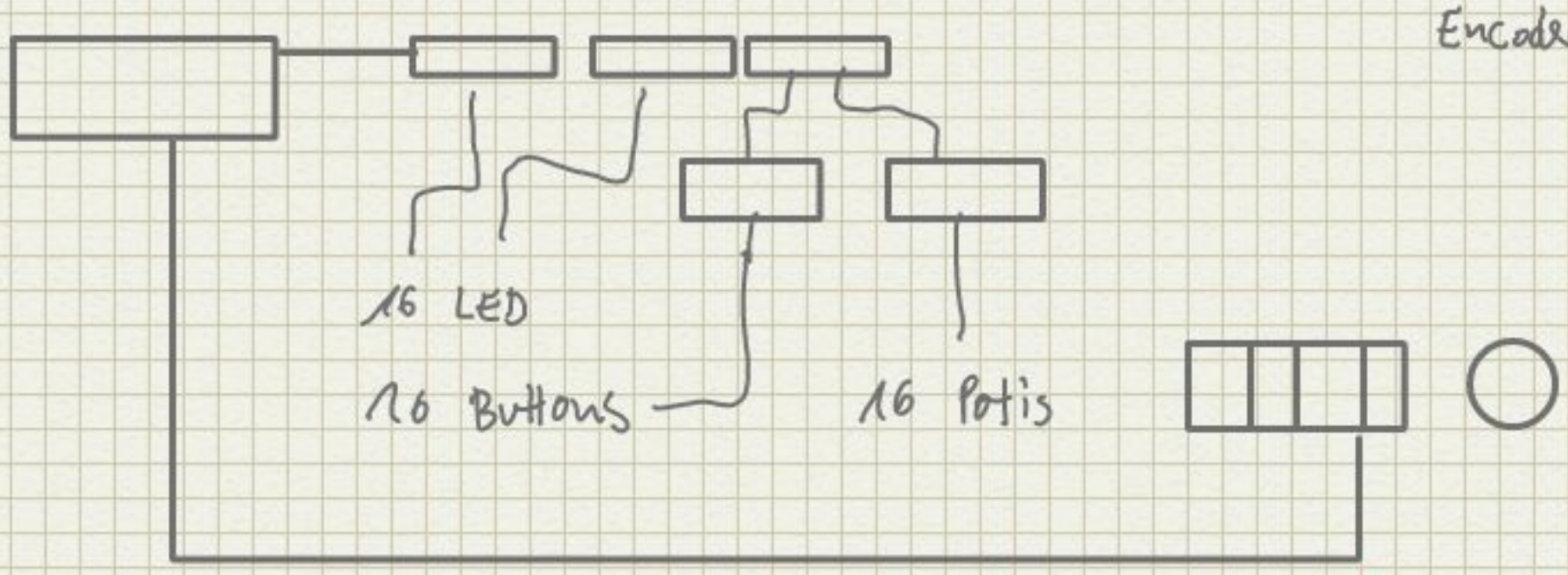
Live: MIDI-IN

* Hold: automatically chosen, if using Button-Array

Scales: Transpose

27.07.15

Arduino + 3x shift-register + 2x MUX + alpha + RGB + Taster + Encoder



PWM Audwino | PCM

PWM
PWM
PWM
} R G B LED Rotary Encoder

Lautsprecher + Amp. (intern)

Battery (UBEC) + Lade-Netzteil

Code main-loop():

MIDI.read
getButtonStates()
readEncoder()
// Menu
switch (menu)
case SET_ARPEGGIATOR:
if (button[BTN_1] == HIGH) {
param += encoder_state;
// register[BTN_MODE] = LOW;
// register[BTN_1] = LOW;
}
break;

if (encoder_state != 0)
setLED()
setRegisterPin( );
clearRegisterPin( );
ClearLED()
setButtonLED(btn)
blinkButtonLED(int btn)
setRGBLED

// Sequencer
if (mode == PLAY && !step_played) { // rainbow(from_color, to_color);
if (clock > next_step)

if (mode == STEP | PLAY

mainloop

MIDI-In
- Clk
- Note-on, Note-off
- CC

Sensoren
- Mux
- Encoder
- Trellis

soll an Ort und Stelle in Code gesetzt werden

Output LEDs
- Shift Register
- alphanum Disp.
- Encoder (RGB)
- Trellis

Output MIDI

Output CV

```
// Pseudocode

read (MIDI-IN)
  switch ...
  case clk:
    if clk== EXT
      if „next step"
        step++
    else if clk==INT
      // compare time
      if (millis() >(time + (timebase / 16)))
        step ++
    break;
  case note_on
  case note_off
  case CC
for button in Mux
  if btn_mode == PRESSED && btn_shift== NOTPRESSED
    // Debouncing Code
    // Arpegiator
  if btn_shift == PRESSED && btn_mode == NOTPR.
    // Pattern Mode: Edit Sequence Parameters
  if btn_shift== PRESSED && btn_mode = PRESSED
    // Song Mode: choose Pattern
  if btn_shift== NOTPRESSED && btn_mode== NOTP.
    // Transpose
read Encoder
read Trellis
// shift register
writeRegister();

// play note

MIDI.write( pattern[step]);
CV.set( getVoltage( pattern[step]));
if (...)
  Gate.on()
else if (...)
  Gate.off()
```

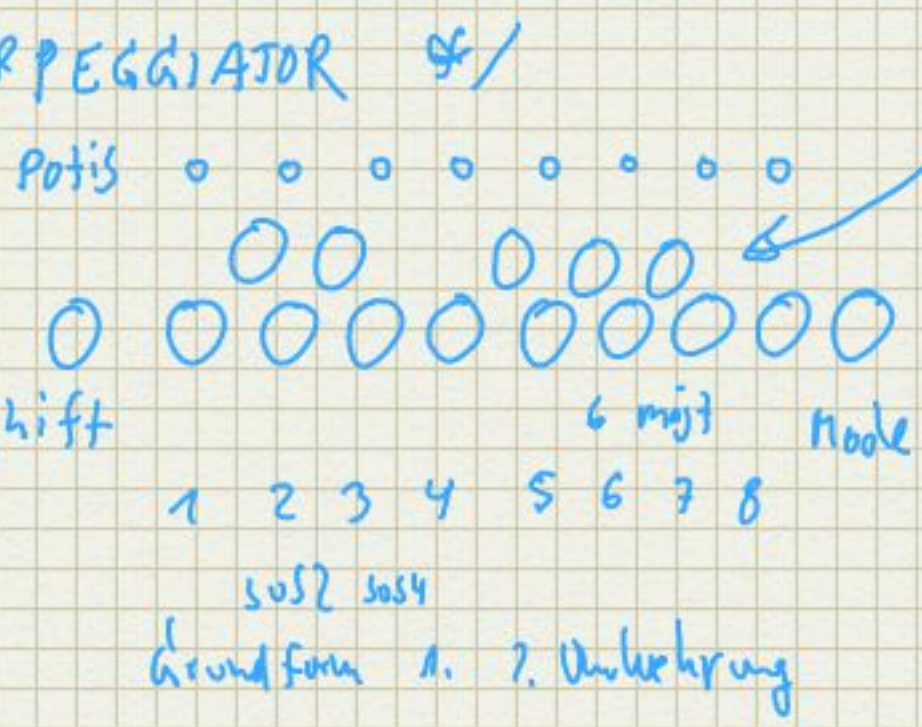
```
// global variables to store state
param: Settings // Struct
  bpm
  timebase = getTimeFromBpm();
  time
  patternNumber
  pattern[]      ; sizeof(pattern)
  step
  mode: enum{play, rec, set param}
  register[16]
        length
        bpm
        ...
```

```
/* Implementierung der Mainloop */

setup() {
    // load settings from EEPROM , init() Modules
}

loop() {
    // Input : MIDI-In
        // clock
        // note-on
        // @see MIDI-Implementation-Chart
    // Input: read Sensor Data
        // Encoder
        // Mux ← Buttons
        // Trellis
        // analog Read : Potis
    // Output : LEDs etc.
        // alphanum Display
        // shift-register → Button LED-Array
        // Trellis
    // Output : CV/Gate (DAC + Digital PIN)
    // Output : MIDI-Out
}
```

/* ARPEGGIATOR */



Mode: freehand (hold)
scales
transpose
step sequences

Parameters: octaves
steps
up/down/up+d/rnd

Input: note-NO.
velocity/Accent
speed | sync

/* menu */ alle Einstellungs-Werte speichern EEPROM
laden von

clock
  Mode
    master
    ext Sync
  Tempo [bpm]

Arpeggiator
  on | off (start | stop)
  chord
    Record | Set (from Button-Array | MIDI-In) Live mode
    Scale select (then transpose, if note is received)
  Octaves [1|2|3]
  Mode [up | down | up & down | pattern]
  Steps (= sync | reset every n steps]
  length of notes [staccato bis legato]

LED on (onReceiveMIDIIn)
- hold mode
→ alle Töne der Tonleiter leuchten lassen
- deselect erlauben

Step-Sequencer
  start | stop | record → mode < Poly 800 / 303
  // Button-Array | MIDI-In
    Note value | rest / pause / silence
    Octave -1 | +1
    Accent
    Slide | length
    next step | previous step

LED on (hold mode)

Save | Load Settings (or Presets or Demo Song)

arrays patterns[8], sequence[STEPS][3] // for note, vel, slide
↳ 8 Beats = 32 steps ⇒ Anzeige

setClock()
param.clock = getTimeFromBpm(bpm);

registers[PIN_MODE] = 1; // setLED(on)

Notizen/Kommentar: make Code faster on embedded Systems
@see C Programming for Arduino

- Hardware-Schalter für Arpeggiator verwenden (MUX 2)

use-cases

z.B. press <Mode>: Button leuchtet
press <Range>: Button blinkt, alpha zeigt Wert
- press -1 | +1
- Encoder
- press btn 1-5

press <Range> again: LEDs <Mode>, <Range> aus

Power

Netzteil

PC-Netzteil vs. DC vs. AC (Doepfer A100)

Batterie

Kapazität, Leistung, Betriebszeit

UBEC

Spannungswandler

5V 700 mAh; 9V ; 3,3V; LED a 60mAh → 6Ah

Induktions-Ladung

Arduino

LED 16 Button

16 in Trellis

32 Neopixels a 60mAh 2 Ah

1 RGB-LED in RotaryEnCoder 60mAh

MFOS 3Ch Amp. 9V

MFOS Alien Screamer 9V

Raspberry Pi 5V 700 mAh

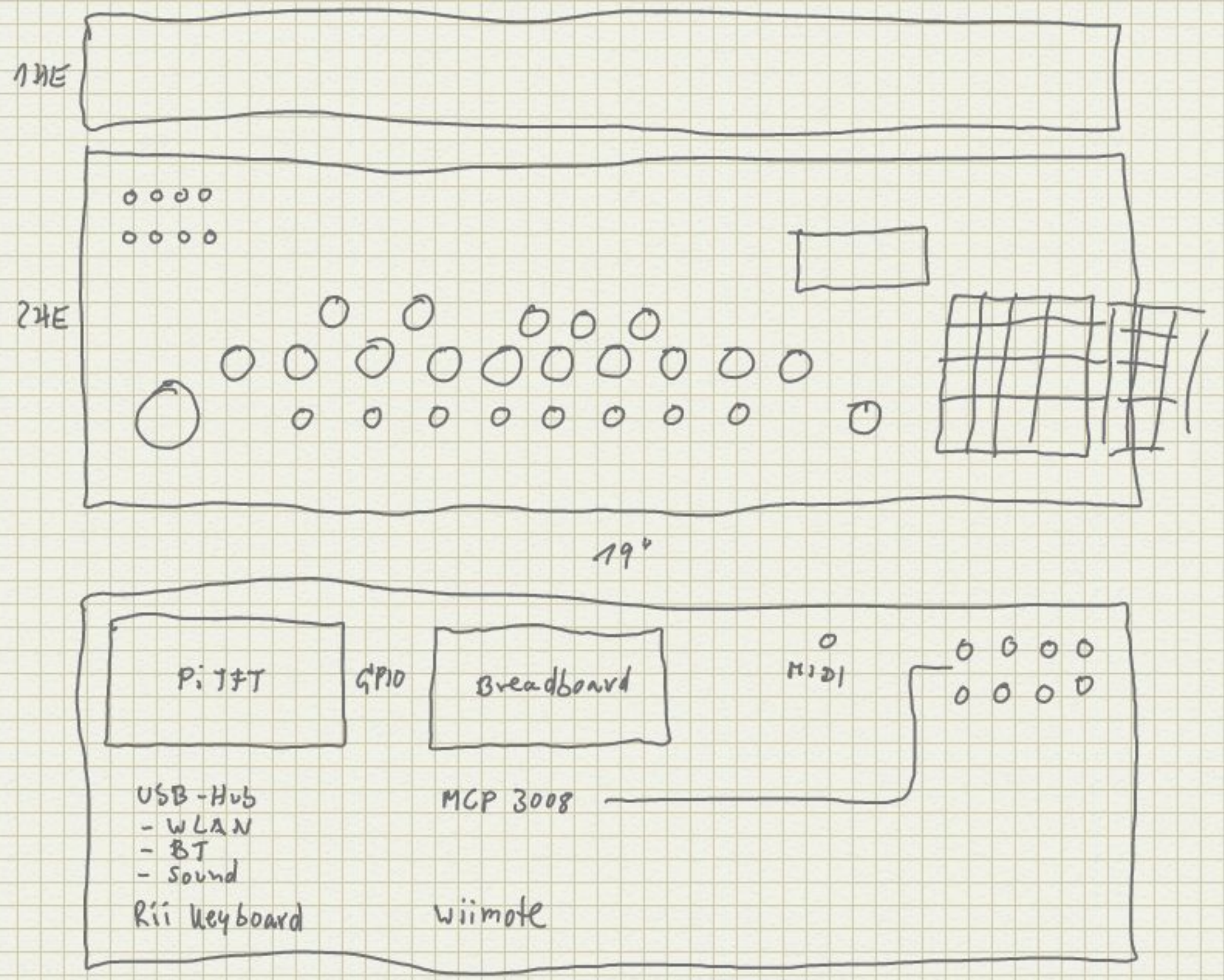
PiTFT

FRONTPANEL DESIGN
PICTOGRAMME

IoT
Physical Computing
Arduino AVR Atmel ATMEGA328p
Raspberry Pi Model B
C/C++
Python
Laser Cutting, CAD, 3D printed ...
case made of wood (zero CO2)

http:// www.floppy-infant.com

Kindermotive (Tiere aus Bilderbuch, Fotos
→ Open Source Bilder



06.08.2015

What to do next:

Implementierung des Menüs (Button Behavior)
Formel für die Sequenz
non-blocking Blinking Button-LED
Neopixels ?
Rotary Encoder (static) variables
LUT
Callback / Handle Design-Pattern für Button-Press
(Observer) @see GoF

GitHub: Adafruit, ebooks, cheat sheet
Frowtpanel Design: Colors, Patterns, Scales / Text UI, Open Badges

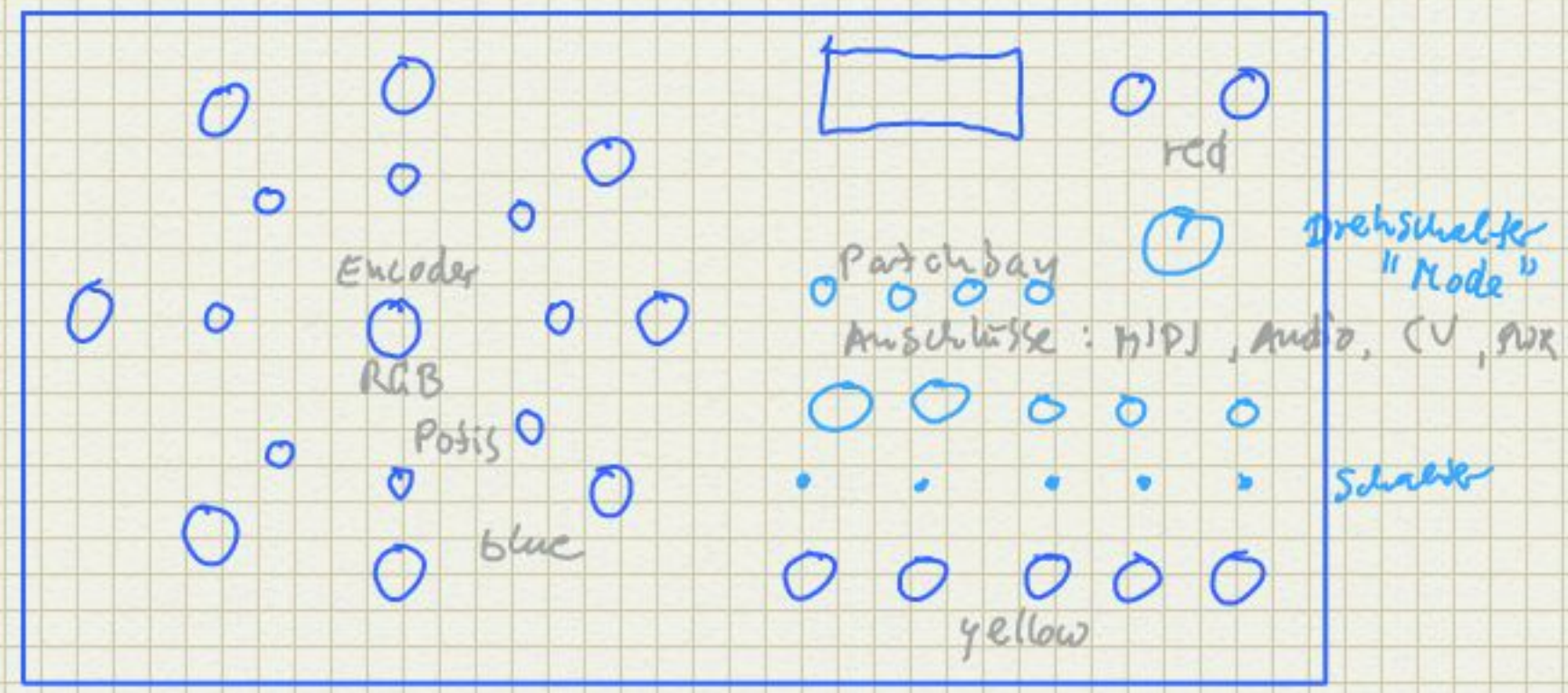
Eagle: Tutorial Adafruit
Fritzing:
Oscilloscope: Einführungen lesen, Manual, Test-Schaltung
- Adafruit PiTFT
- Soundcard & Probe
MFOS / MAKE: OpAmps, ...

Neue Ideen: – Drehschalter (>6 Stufen) als Mode-Selector

– Rundes Layout für Step-Sequencer (s. Bild)

– Mensch-Maschine-Kommunikation verbessern:
  • Live Performance Anzug mit LED, EL wire synchron zur Musik (MIDI-IF, Light-Sequencer)
  • Controller: 'Musical Glove'

– Novation-Bass Station (Tastatur!) umbauen



musicfromouterspace.com

VCO Frequency

LFO Modulation

LFO Frequency

Output

Output Level

Modulation Shape

Sync Effect

Off

MFOS Alien Screamer

Power

Off On

Music From Outer Space NOISE TOASTER

Voltage Controlled Oscillator

Frequency

Sync

Off

LFO Mod Depth

AR Mod Depth

AR Mod

Off

Output

Volume

Voltage Controlled Low Pass Filter

Input Select

VCO

Off

White Noise

Cutoff Frequency

Resonance

Mod Depth

LFO

VCA

VCA (AR mod)

Bypass

AR Envelope Generator

Attack

Release

Repeat

Manual Gate

AR

LFO

LFO Waveform Select

Rate

Power

On

Off

www.musicfromouterspace.com

CV/Gate

Gate +5V, +8V, +12V oder invertiert - 5V

S- Trigger (short circuit, connection to ground)
- - > Korg MS20, Moog

1, 2, 5 V/Oct.

8 Hz/Oct. - - > Korg MS20

- -

Doepfer MCV4

MOTU Volca
Expert Sleepers Silent Way - - > Audiocard

s.a. MFOS

BRAND NAMES

Kreative Energie
MADS1
MADS One-O-One
MADS IA
Music Analog Digital System
Musical Analog Digital Synthesizer

SUN 1
Mars 3
NERD1

Happy Sounds
Klangkulisse
Geräusch~
Hintergrundmusik
Rauschen

Lo-Fi Noisebox
ARP
SQ

Additional

MFOS Noise Toaster

- Analog Sequencer
- Ringmodulation
- Sample & Hold
- X-Modulation
- Subosc illator

ext-in
CV/Gate-Out (2x)
Patch-Bay

4 Octaves Keyboard

Stylophone (analog keyboard)

Drum-kit-kit

Ultraschall (Theremin style)

~~Laser Harp~~

Kosten

Projekte Kosten

Raspberry Pi	115 €	4x & Zubehör ≈ 500€
LED-Strips	65 €	
Boslight (LPD8806)	100 €	
Monkey Light (Arduino)	130 €	+ Neopixel 0,5m 27€
Korg monotron (3x)	105 €	
MFOS Alien Screamer	65 €	
MFOS Noise Toaster		
Synth-DIY Workstation	727	PCBs ca. 47 €
Reichelt (MFOS, Arduino, Löten)	230 €	
Exp Tech	70 €	
Exp Tech	190 €	EXP 3 45 €
Sundtek SkyTV Ultimate 4	100 €	
DIY-Cam	120 €	
Samsung Galaxy Note 8	330 €	
Zubehör	130 €	
EL-Wire	105 €	
Oscilloscope DSO203	127 €	

400 €