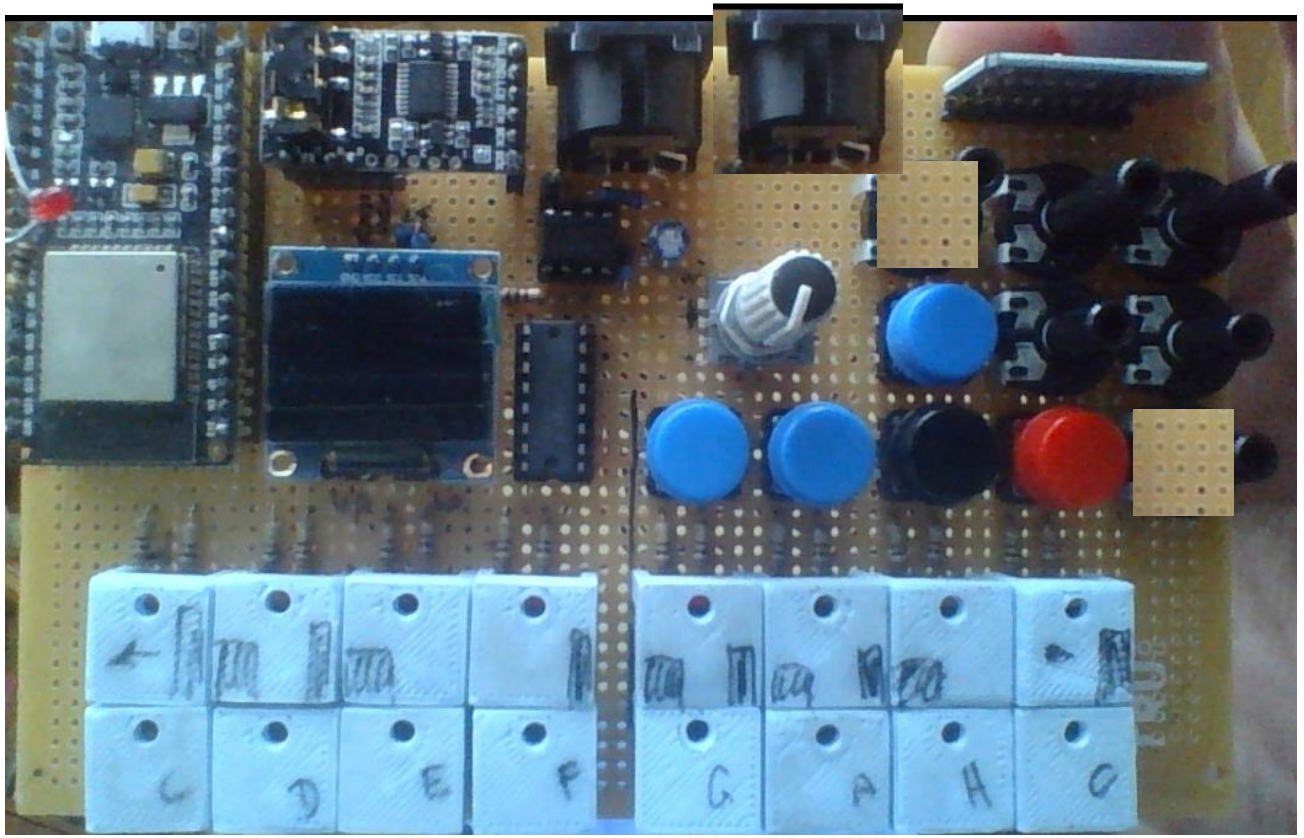
POLCA ESP32 Step Sequencer

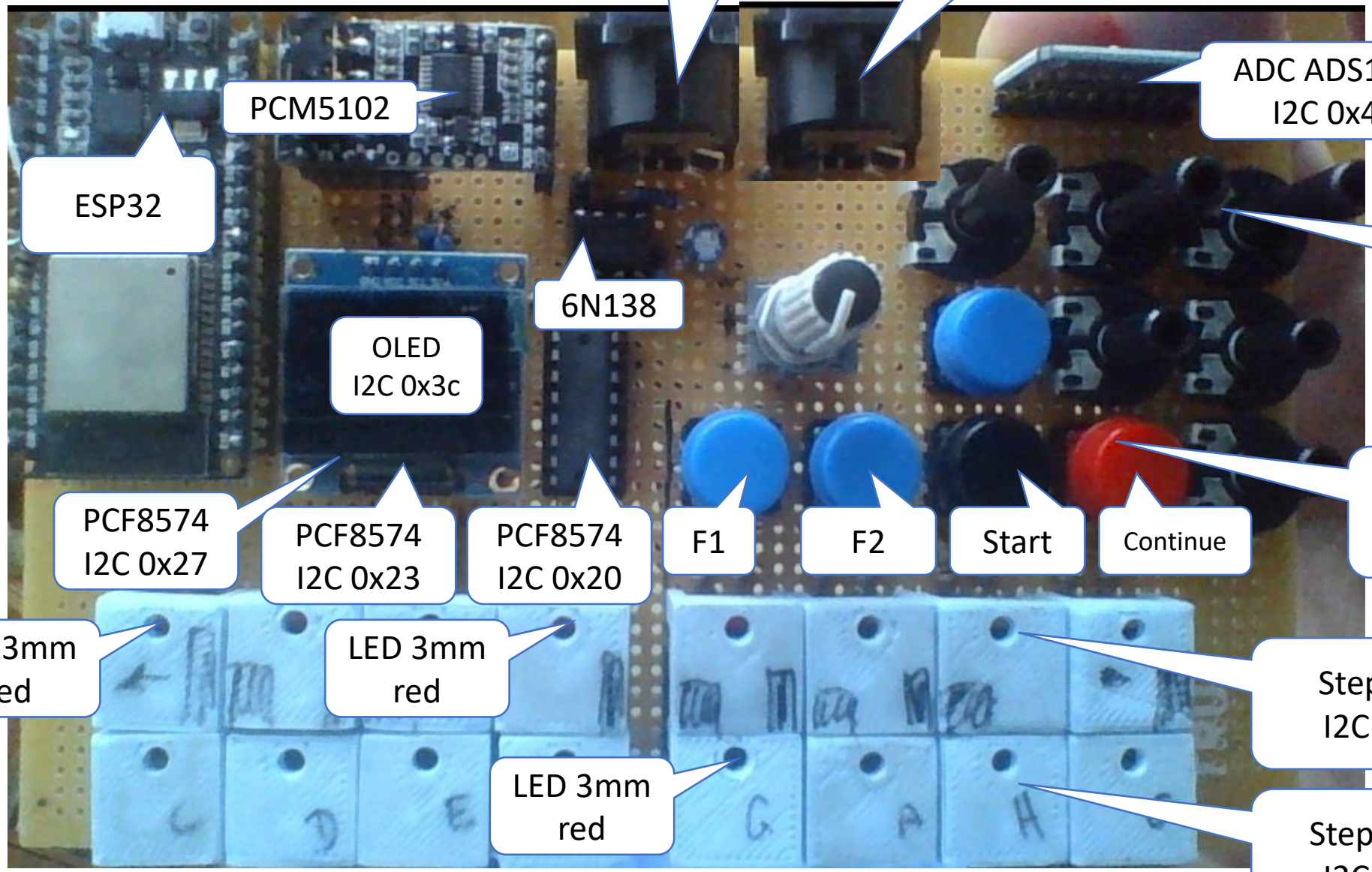
Polca

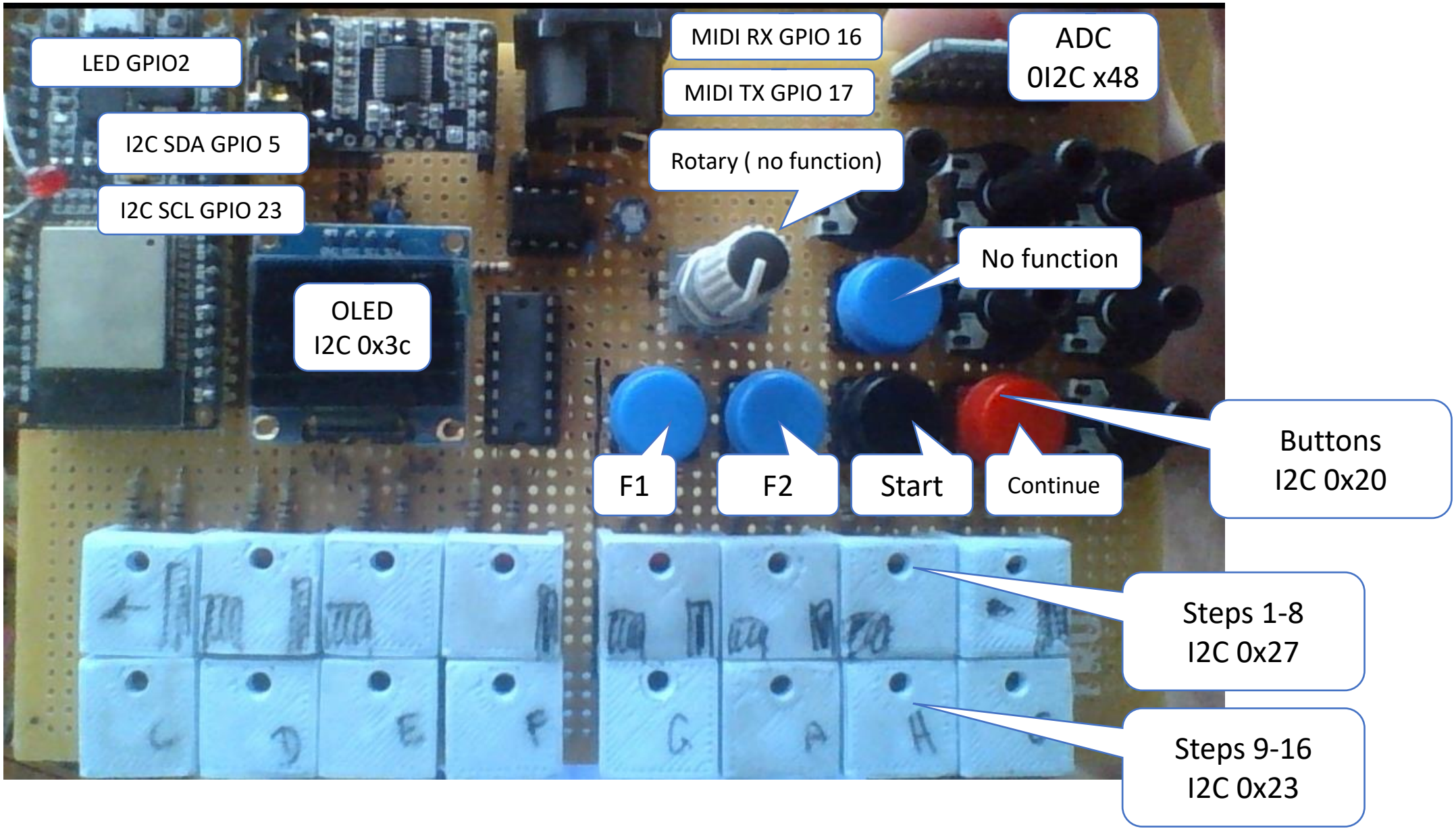
Soundengine based on EDP32 and the Code from Marcel Licence

Step-Sequencer made by Erich Heinemann

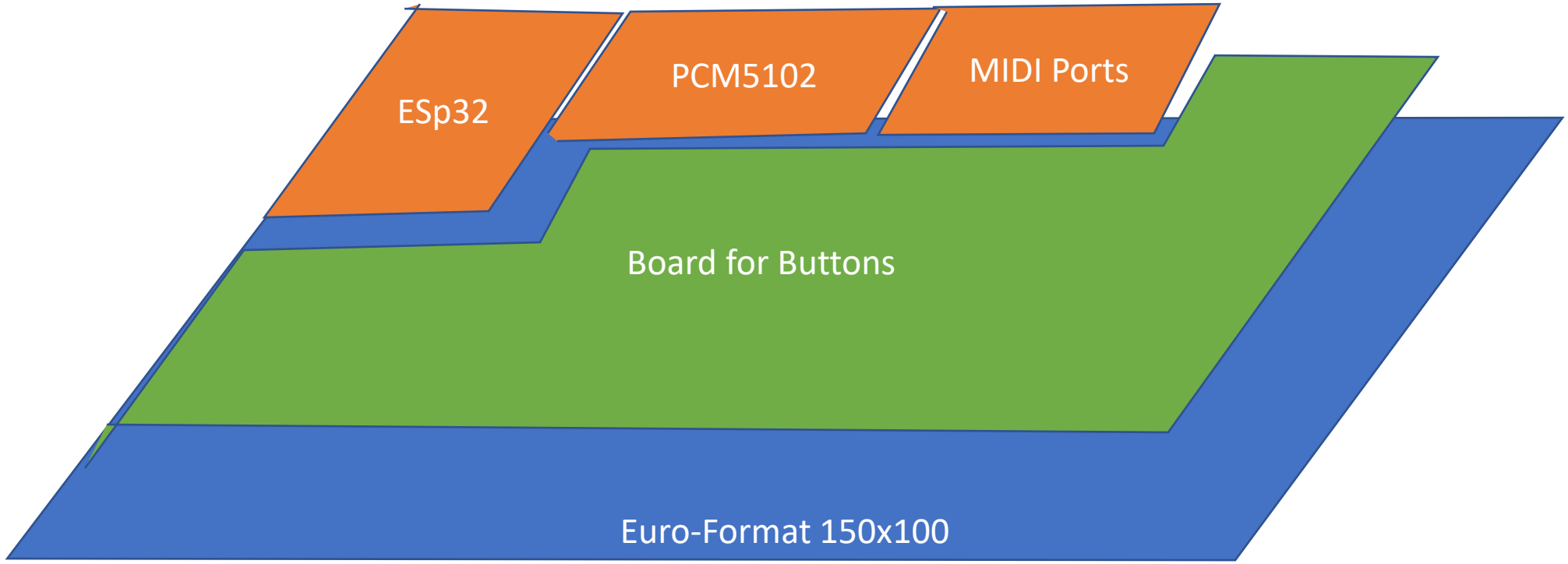
LAB-Board




MIDI In-Port
MIDI Out-Port
PCM5102
ADC ADS1115
I2C 0x48
ESP32
Trim-Pots
10kOhm
6N138
OLED
I2C 0x3c
PCF8574
I2C 0x27
PCF8574
I2C 0x23
PCF8574
I2C 0x20
F1
F2
Start
Continue
Buttons
I2C 0x20
LED 3mm
red
LED 3mm
red
Steps 1-8
I2C 0x27
LED 3mm
red
Steps 9-16
I2C 0x23


LED GPIO2
I2C SDA GPIO 5
I2C SCL GPIO 23
OLED
I2C 0x3c
MIDI RX GPIO 16
MIDI TX GPIO 17
Rotary (no function)
ADC
0I2C x48
No function
F1
F2
Start
Continue
Buttons
I2C 0x20
Steps 1-8
I2C 0x27
Steps 9-16
I2C 0x23

In the first LAB-Version, the Buttons are on the Euro-Board and that way, the buttons are to low, MID-Ports and the EP32 are the highest circuits on the board and define the top height level of the box.
On any Korg Volca, the Buttons and Pots are on the same level. The only way to get to this point with simple Boards is by adding an extra board for the buttons and pots.
Another way would be to mount the big things like the ESP32, PCM5102 and MIDI-Ports underneeth the Board – upside down. But this is more like a nightmare to do this on a simple throughhole stripboard.



ESP32 Dual-Core

Core1 (Default)

- I2S
  - Sample-Player for Drums
- Sequencer
- MIDI

Core0

- I2C
  - PCF8574
  - Rotary Encoder
  - Buttons
  - Display
  - Keyboard
- Menu
- Display-Management
- UI-Refresh

Control-Buttons
P0 = Func1
P1=GlobalFunc
P2=Start/Stop
P3=Continue
P4= *not implemented*
P5-P7 - Encoder

SCL
SDA

PCF8574 No.3

Steps 1-8

PCF8574 No.1

Steps 9-16

PCF8574 No.2

SCL GPIO23
SDA GPIO 5
GND
3.3V

p0
p7

S17

LED

100Ohm

S1

1 StepButton

Next 15 Steps
are the same

16Bit I2C ADC+PGA
ADS1115

pot0
pot1
pot2
pot3

I2S_BCLK_PIN GPIO25
I2S LRCK / WCLK GPIO27
I2S_DOUT_PIN GPIO26
GND
5V

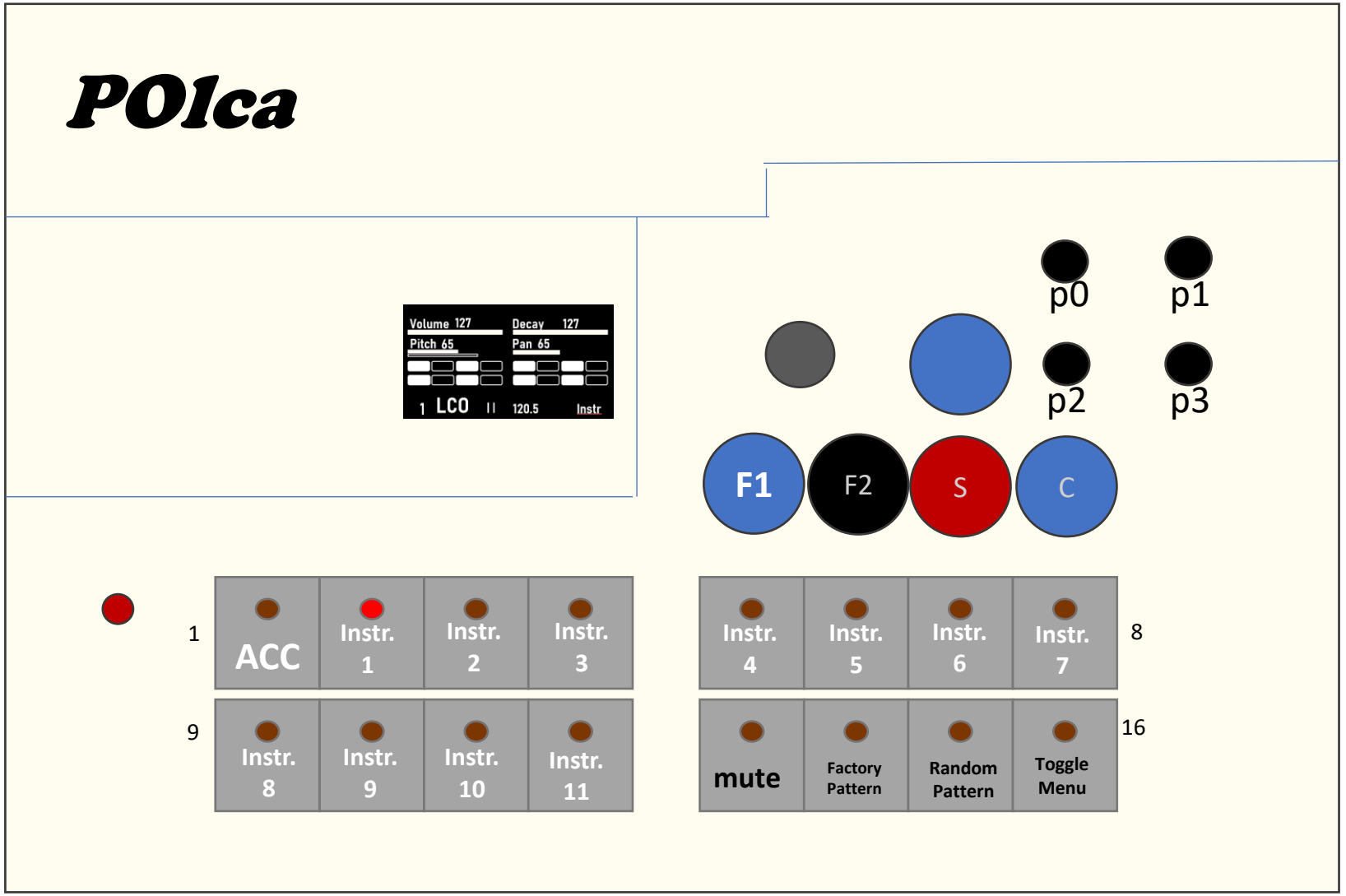
Bridges!

GPIO 17
Midi TX
GPIO 16
Midi RX
3.3V
10k
1k
RX2
GND

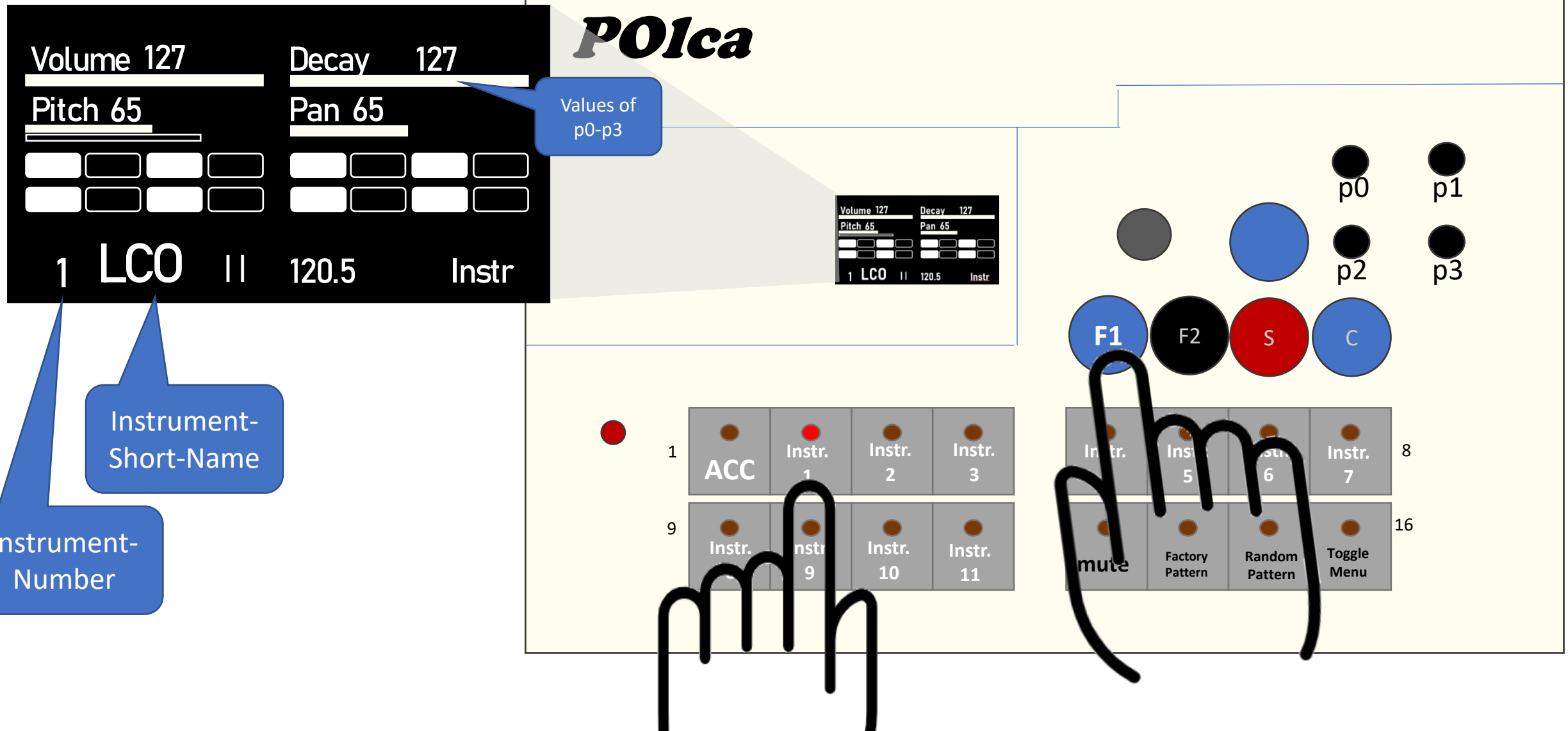
6N138
2x 110
or use one
220 resistor
MIDI

POlca Userguide

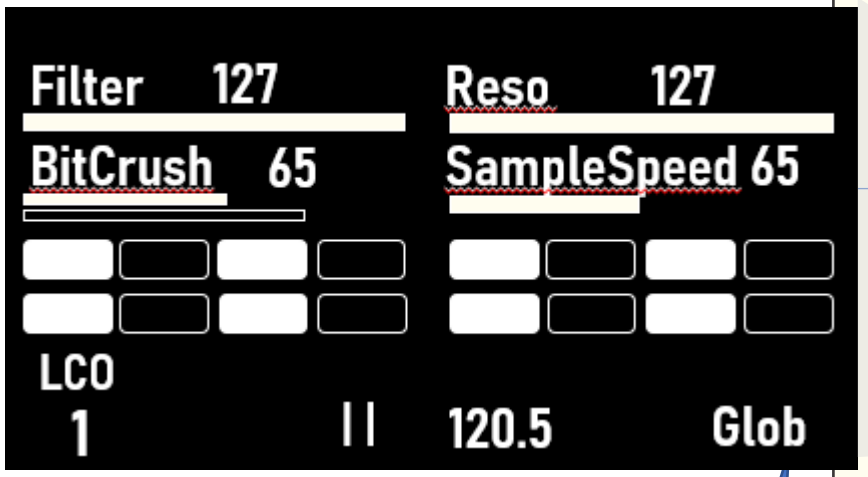
- 29-Jul 2021
  E.Heinemann



Press F1 and then add pressing a Step-Button to switch the Instrument or to mute the Instrument (Step 13), load factory (Step 14) or random (Step15) Patterns or toggle the Sound-Menu (16)
This feature works in Instrument-Menu and Sound-Menu !



Press F2 and then add pressing one of the first 8 Step-Button 1 - 8 to switch menu
This should work in all menus.
Button 16 provides a feature to toggle through all menus

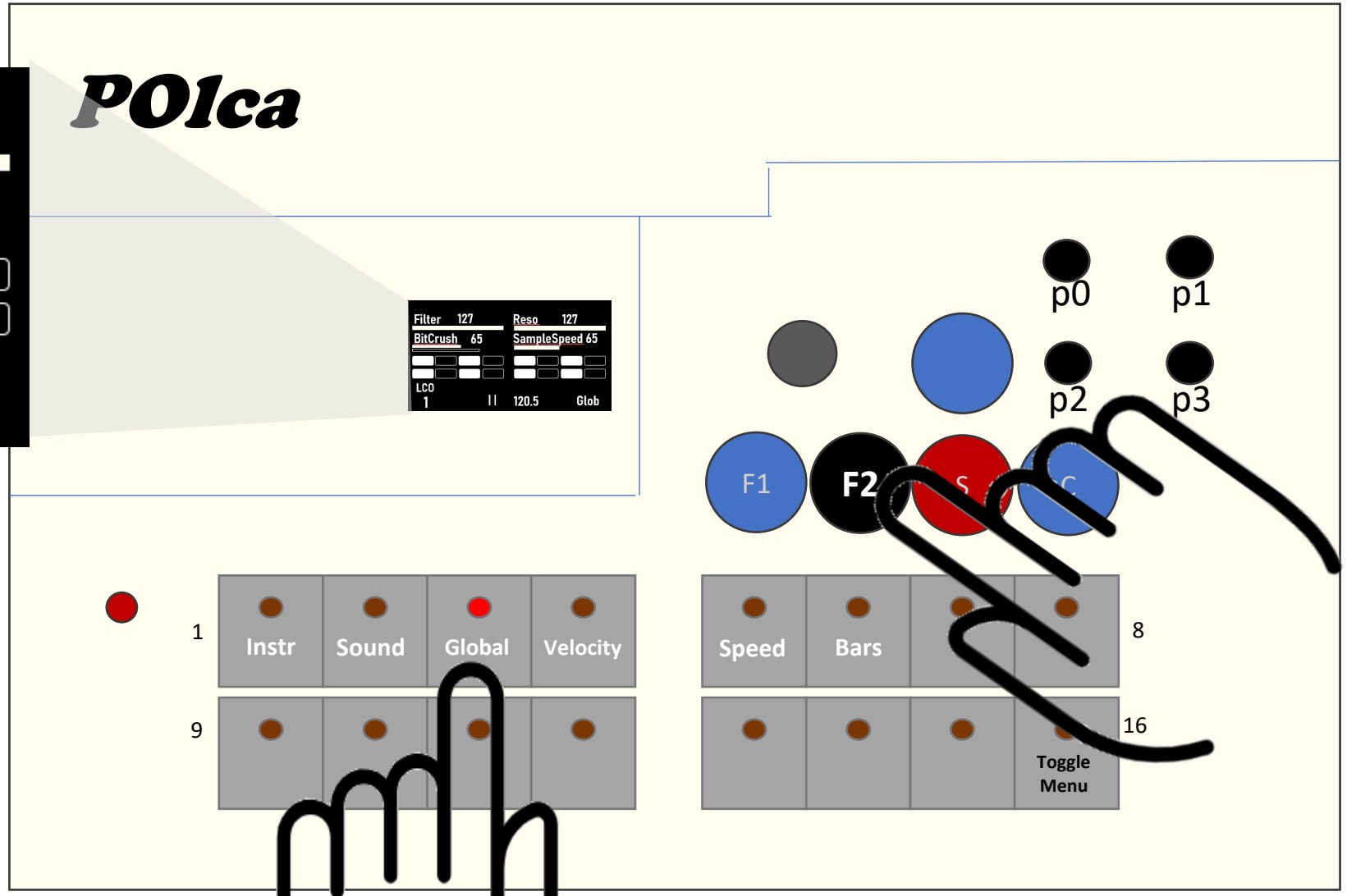


The Global Effects provide a BiQuad-Filter with FilterFrequency and Resonance.
The Filter acts as a Lowpass if the values are lower than 65. If the values are higher than 65 it acts as a Highpass.

If the Bitcrusher-Value is higher than 0, Bitcrusher reduces the Bitrate. Very high values could cause total loss of the signal!

The Samplespeed modifies the PCM-Speed of the global Sampleplayer. All Sounds are affected.

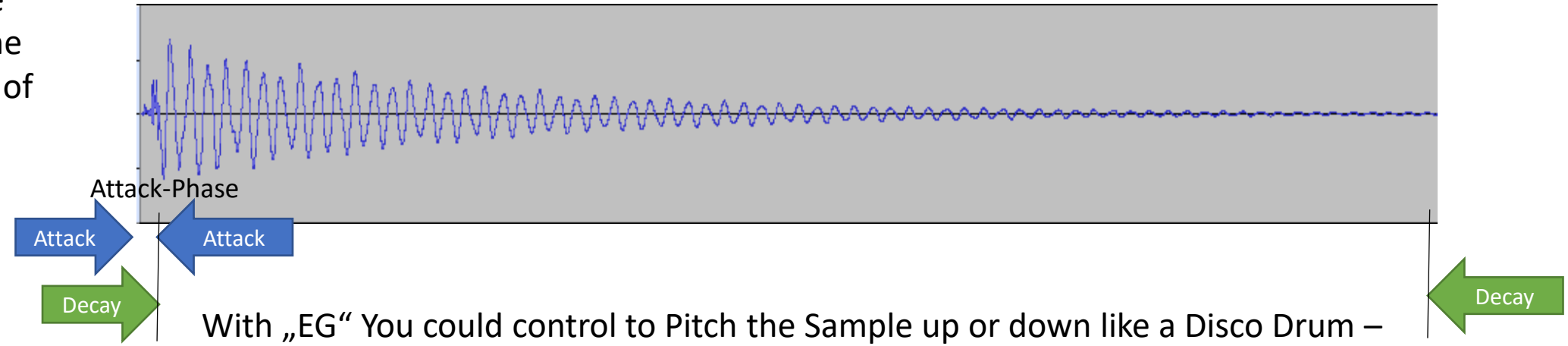
Menu-Short-Name



In the Sound-Menu, the pots manage the MIDI-Note-Number and MIDI-Channel (1-16) to trigger external MIDI-Gear.

Attack is a simple feature to cut off the start of a Sample.
If the attack is at the value „0“ the sample will be played starting at Byte 0
If the Attack is at the value 100, the 16Bit-Sample will be played starting at Sample 100 – Byte 200.

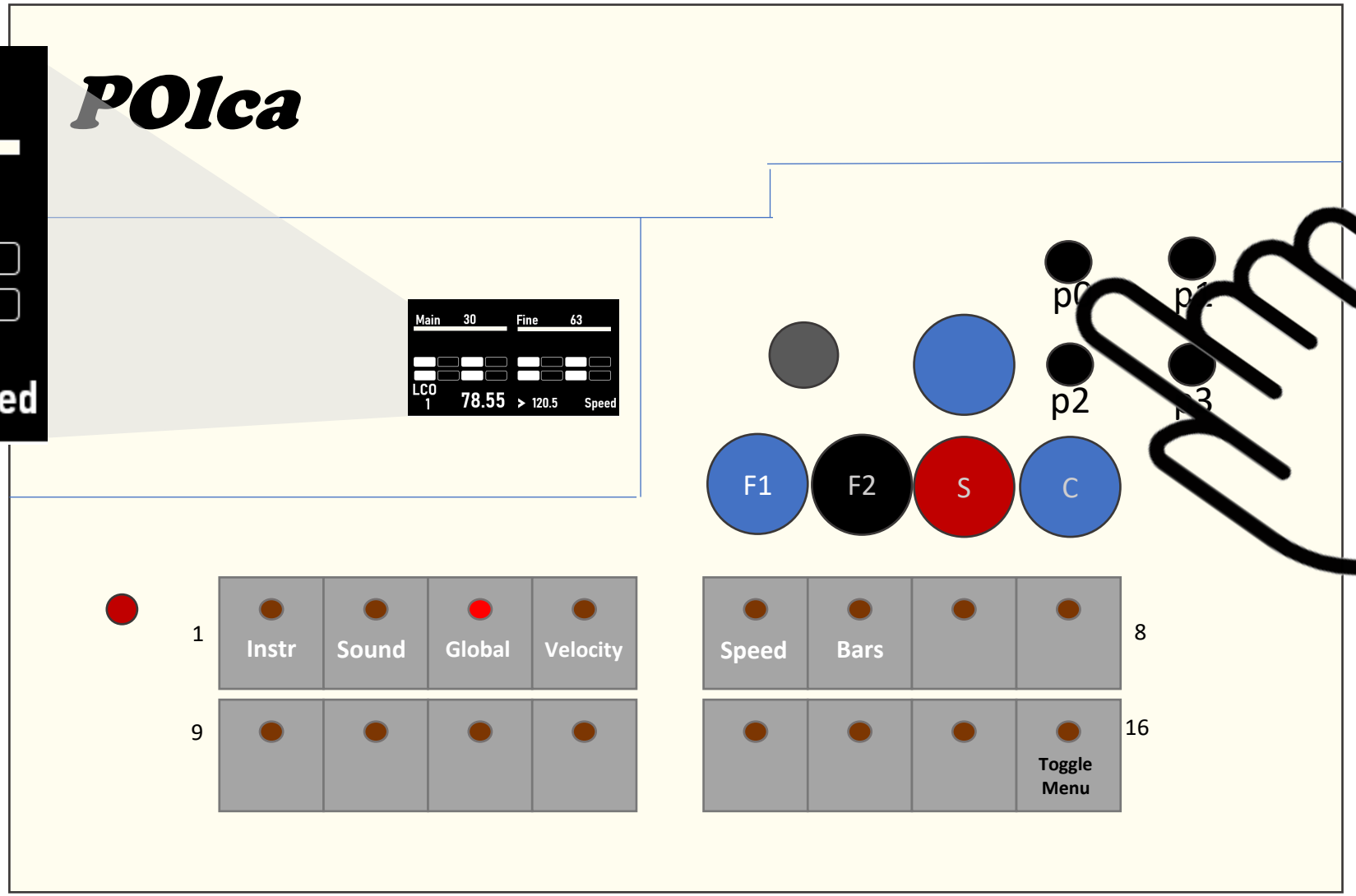
That way, You are able to change the first milliseconds of the sound.

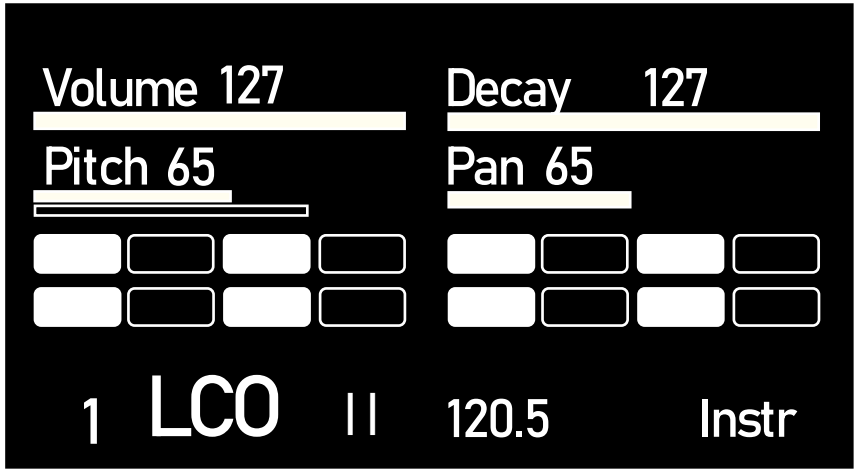


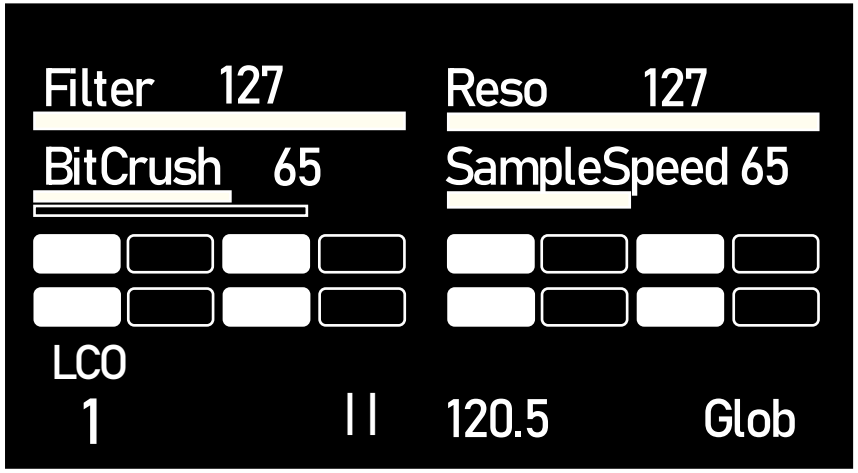
With „EG“ You could control to Pitch the Sample up or down like a Disco Drum – PIU versus PUI-Sounds…
Values of 64 does not modify the sound and the sample will be played with the same Samplerate/Speed.

In the Speed-Menu, manage the Speed via Potentiometer p0 and p1.

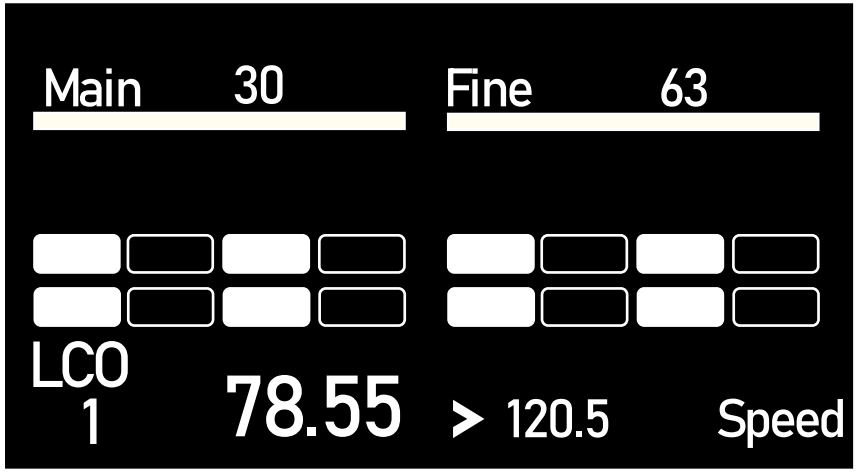
p0 works in 2BPM Steps, p1 provides access to 0.05 BPM-Steps




Volume 127
Decay 127
Pitch 65
Pan 65
1 LCO II 120.5 Instr


Filter 127
Reso 127
BitCrush 65
SampleSpeed 65
LCO
1 II 120.5 Glob


NoteNum 40
Channel 11
Attack 65
EG 65
1 LCO II 120.5 Sound


Main 30
Fine 63
LCO
1 78.55 > 120.5 Speed

Github ESP32 Collection

- https://github.com/RuudErmers/NTS1_ESP32_Development/tree/main/ESP32AudioA1S/WaveSynth

- https://github.com/bwhitman/alles