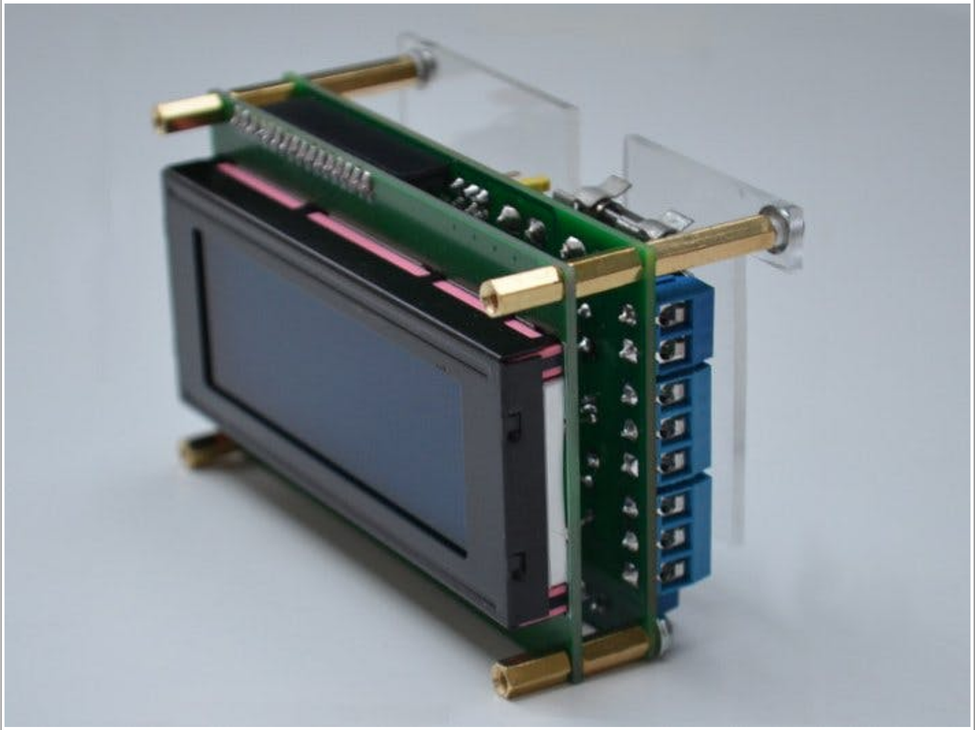
Project page on Hackster.io Published November 21, 2022 © CC BY-NC

Mini serial console with three different serial ports

This is my "poor man's HID" that I want in my projects to enter data and display messages.



Story

It is a simple serial console that can be used for data display and data input. It can handle three (different) serial interfaces at the same time: 3.3V TTL for single-board computers and microcontrollers, via traditional RS-232 and serial over USB.

I. Hardware

1. Technical data

- **Supply voltage:** 5 V DC
- **Supply current:** approx. 30 mA
- **Mechanical size:** 100 x 60 x 50 mm
- **Insulation class:** Class 0
- **IP protection:** IP 00
- **Mass of cover:** termoplastic (PC), only top cover
- **Display size:** 20x4 characters
- **Displayed character set:** ASCII 32-127
- **Connection:** 3.3V TTL, RS-232 and via USB serial ports
- **Serial port receive buffer size:** 255 byte

2. General description

This is a Raspberry Pi Pico-based device. The display is 20x4 character-sized liquid crystal display, which displays 20x4 size area of the larger size virtual screen. This area is a can be moved with buttons. The default size of the virtual screen is 80x25 characters, this value can be set in the program.

The circuit has three serial ports, depending on the setting, all three can be used for data and system message transmission, or both at the same time. The #0 serial port is via USB port, #1 is a 3.3V TTL port, #2 is a conventional RS-232 serial port. Serial #1 can be connected to a 5V TTL serial port with an external level shifter circuit.

Default port speeds: 115200, 9600 and 9600 bit/s, these values can be set in the program.

The console has four operation modes, the first three for display only can be used, these are pre-programmed, the fourth is reserved for unique solutions. With jumpers between modes we can choose.

All connections are screw terminal blocks.

3. How it works?

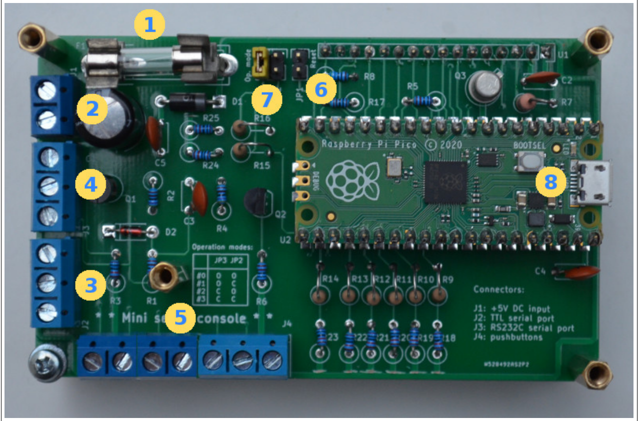
The supply voltage goes to the buffer and filter capacitors and other circuits through the fast-acting fuse F1. The function of the antiparallel diode is protection against reverse polarity supply voltage. If the USB port is in use, the device receives power from it, in which case the power input can only be used as a (low-load) output.

All peripherals are connected to the microcontroller. The signal from serial port #1 goes directly to the GPIO port through the level adapter of port #2. The push buttons and mode selection jumpers (JP2-3) are connected to the port through a protective resistor, in the open state a pull-up resistor ensures a stable high level.

The Hitachi HD44780 compatible display works in eight-bit write-only mode, the data and control lines directly connected to the GPIO ports. The program can turn the display backlight on and off. The contrast of the display depends on the resistor R7, its value may depend on the type of LCD module.

The device is operated by the program, so its detailed operation can be found in the **Software** section.

4. Manuals and connectors



- **1:** Fuse F1 - power supply
- **2:** J1 connector - power input
- **3:** J2 connector - TTL serial port
- **4:** J3 connector - RS-232 serial port
- **5:** J4 connector - push buttons
- **6:** JP1 jumper - RESET
- **7:** JP2-3 jumpers - mode selector
- **8:** USB connector

5. Schematic and PCB drawings

You can download wiring diagrams as part of the complete documentation or in separate PDF, SVG and KiCAD formats from the developer/manufacturer's website. The Gerber files needed for production are included in the package.

6. Other drawing

You can download mechanical drawing of polycarbonate cover as part of the complete documentation or in separate PDF, SVG and LibreCAD formats from the developer/manufacturer's website.

7. Terms of use

Hardware documentation can be modified and/or redistributed under the Creativ Commons 4.0 Attribution Non-Commercial (CC-BY-NC-4.0) License.

8. Downloadable documentation

The complete documentation of the hardware in the.tar.gz format compressed file can be downloaded from the manufacturer's website or Github.

II. Software

1. General description

The program that operates this device was created on the Arduino IDE development system. This application must first be prepared to work with the Raspberry Pi Pico, for which instructions can be found online (for example here).

2. Setup

Before installing the program, you need to set these prepocessor macros:

```
// enable 8 bit mode of the LCD
#define LCD_8BIT
// enable Serial #0 port
#define COM_USB
// enable Serial #1 port
#define COM_TTL
// enable Serial #2 port
#define COM_RS232C
// enable console messages on Serial #0 port
// #define COM_USB_MESSAGES
// enable console messages on Serial #1 port
// #define COM_TTL_MESSAGES
// enable console messages on Serial #2 port
// #define COM_RS232C_MESSAGES
```

and constants:

```
// settings
// LCD backlight off time after last button press
const int     lcd_bloffinterval = 60000;
// horizontal size of display
const byte    lcd_xsize         = 20;
//vertical size of display
const byte    lcd_ysize         = 4;
// horizontal size of virtual screen
const byte    virtscreenxsize   = 80;
// vertical size of virtual screen
const byte    virtscreenysize   = 25;
// speed of the USB serial port
const int     com_speed[3]      = {115200, 9600, 9600};
```

3. How it works?

At start-up, the program initializes serial ports and if consol messages is enabled sends status message. After that, it sets the GPIO ports, clears the virtual screen and initializes the display. Sets its own operating mode according to the jumpers (JP2-3). The display shows brief information about the device.

The program then continuously monitors all serial ports, the state of the buttons, and the state of the mode selection jumpers. If data is received on one of the ports, it is copied to the virtual screen. The LCD display shows a part of the virtual screen as a window. This 'window' can be moved with the buttons. The size of the virtual screen and the ability to move the 'window' depend on the selected mode.

The program turns off the backlight after 60 s of inactivity, then turns it back on when data is received or a button is pressed.

The program replaces the incoming TAB character (0x09) with SPACE (0x20), blinks the backlight in response to the BEL (0x07) character, and starts a new page to the FF (0x0B) in mode #2.

4. Terms of use

This program is free software: you can redistribute it and/or modify it under the terms of the European Union Public License 1.2 version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

5. Downloadable software package

The package can be downloaded from the manufacturer's website or Github.

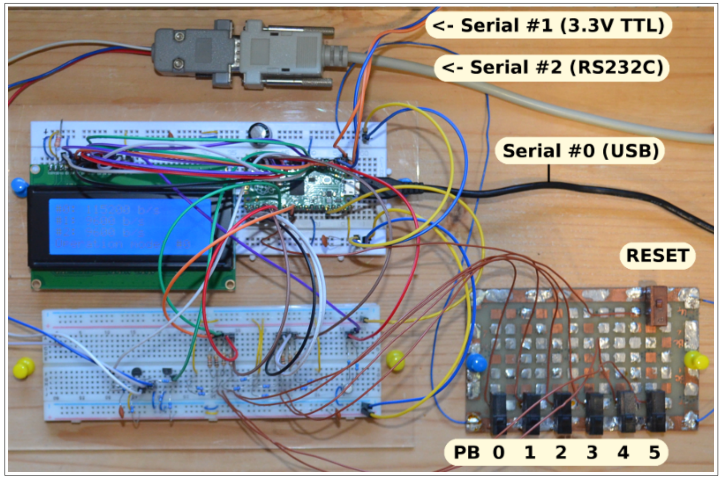
III. How to build this device

1. On breadboards

The wiring of the device is included in the attachment.

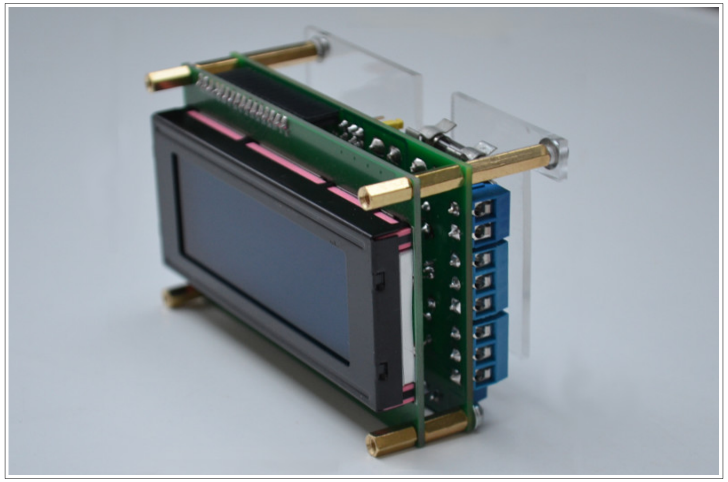
For the prototype, I also used a soldering test panel, on which the microswitches were placed.

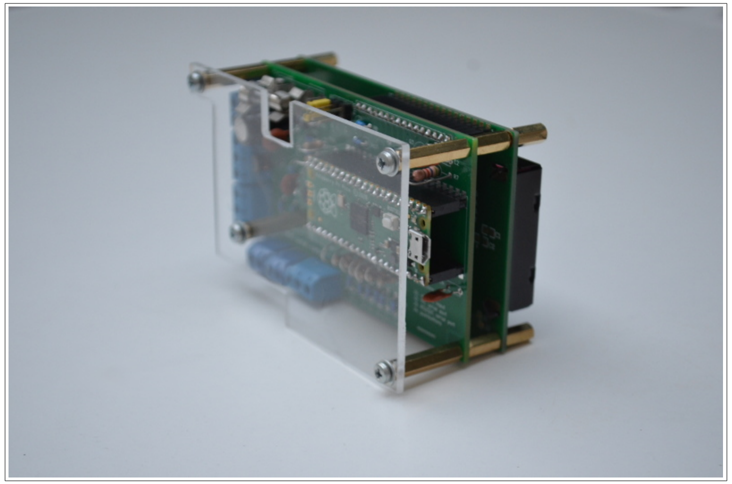
The circuit received 5 V DC power during the test via the USB cable.



2. On printed circuit board

I made the panels again this time at PCBWay.





3. Operation check

I did this activity using two computers. The Serial #0 and Serial #1 ports of the circuit were connected to the computer running Debian GNU/Linux, and the Serial #2 port was connected to the FreeDOS computer. Console messages is enabled in USB port.

The linux computer accessed the console of the circuit via /dev/ttyACM0, the first input via /dev/ttyUSB0, the DOS computer accessed the second input via COM1. (If the second input is also connected to the Linux machine, it will be available on /dev/ttySx.)

For testing, I used terminal programs, the Telix clone called Minicom on the linux machine, and the original Telix (from the early 90s) on the DOS machine. The port and communication parameters must be set on the terminal programs and it is advisable to turn on the local echo and line feed at the end of the line.

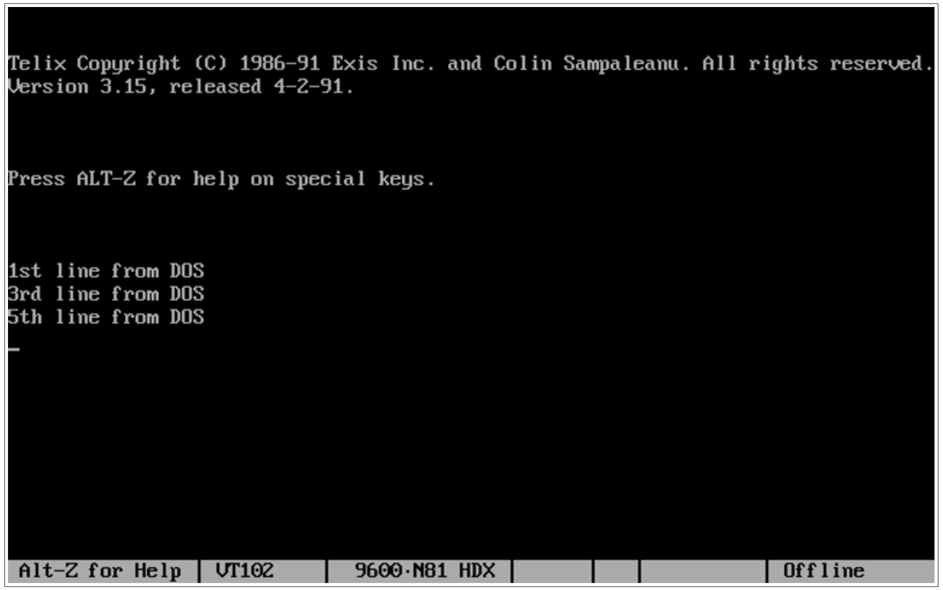
Example of starting minicom with command line parameters:

```
$ minicom --baudrate 115200 --device /dev/ttyACM0
```

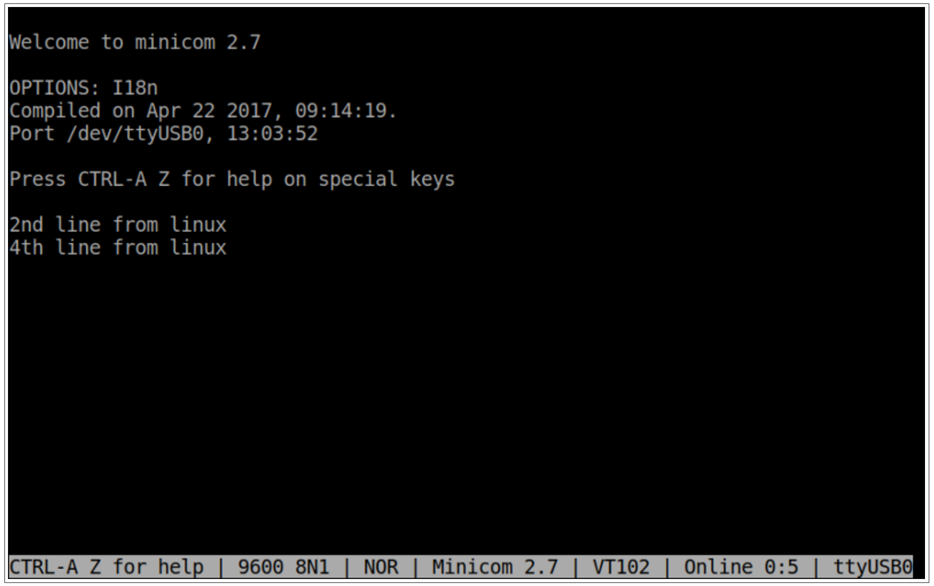
I alternately sent one line to the circuit with the terminal programs and checked correct displaying and scrolling. During the data transfer, the built-in green LED of the Pico lights up.

The system messages of the serial console can be viewed on the terminal program connected to the USB port.

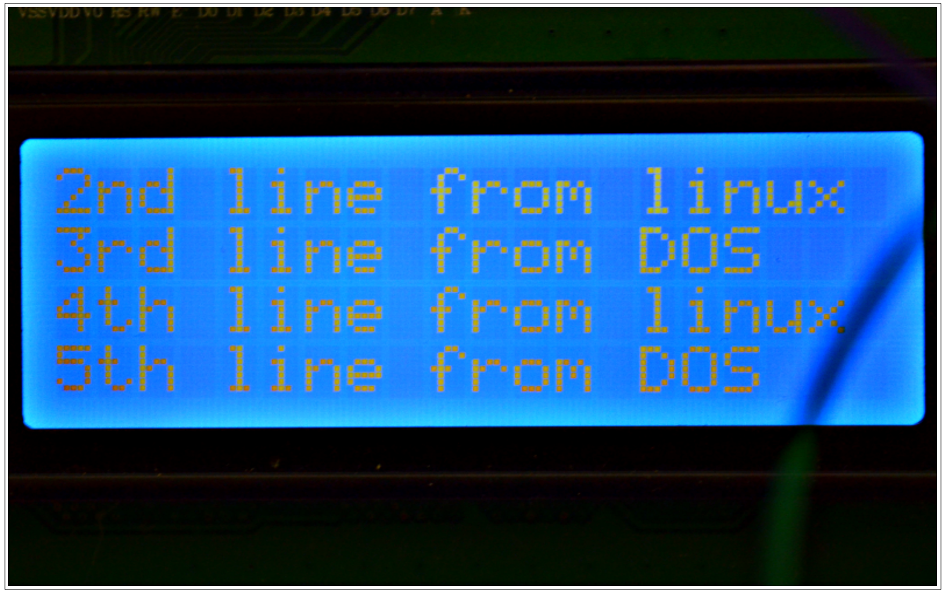
It is advisable to test the device in operation modes #1...#3.



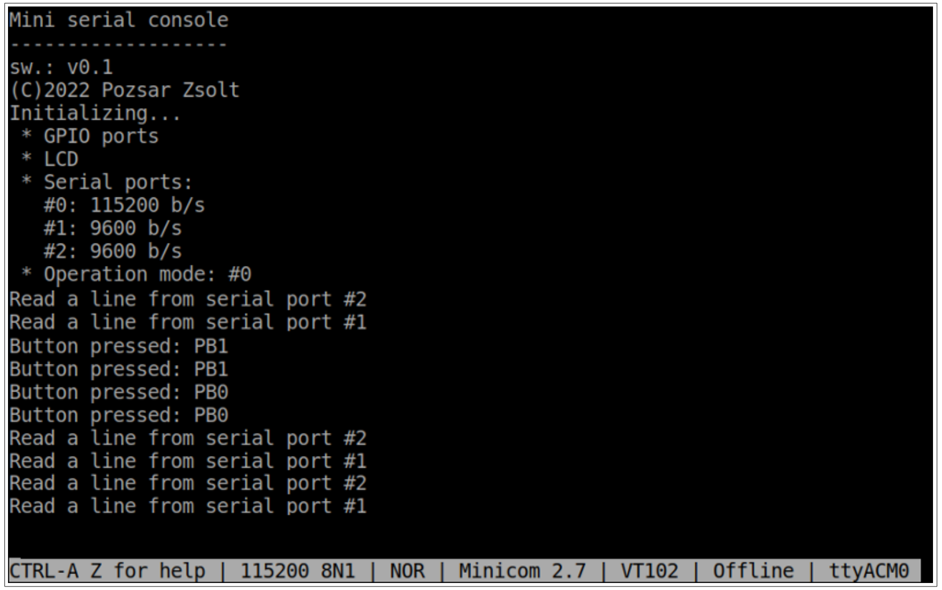
```
Telix Copyright (C) 1986-91 Exis Inc. and Colin Sampaleanu. All rights reserved.
Version 3.15, released 4-2-91.


Press ALT-Z for help on special keys.



1st line from DOS
3rd line from DOS
5th line from DOS
_










 Alt-Z for Help | VT102     | 9600·N81 HDX |       |  |          | Offline
```



```
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Apr 22 2017, 09:14:19.
Port /dev/ttyUSB0, 13:03:52

Press CTRL-A Z for help on special keys

2nd line from linux
4th line from linux













CTRL-A Z for help | 9600 8N1 | NOR | Minicom 2.7 | VT102 | Online 0:5 | ttyUSB0
```

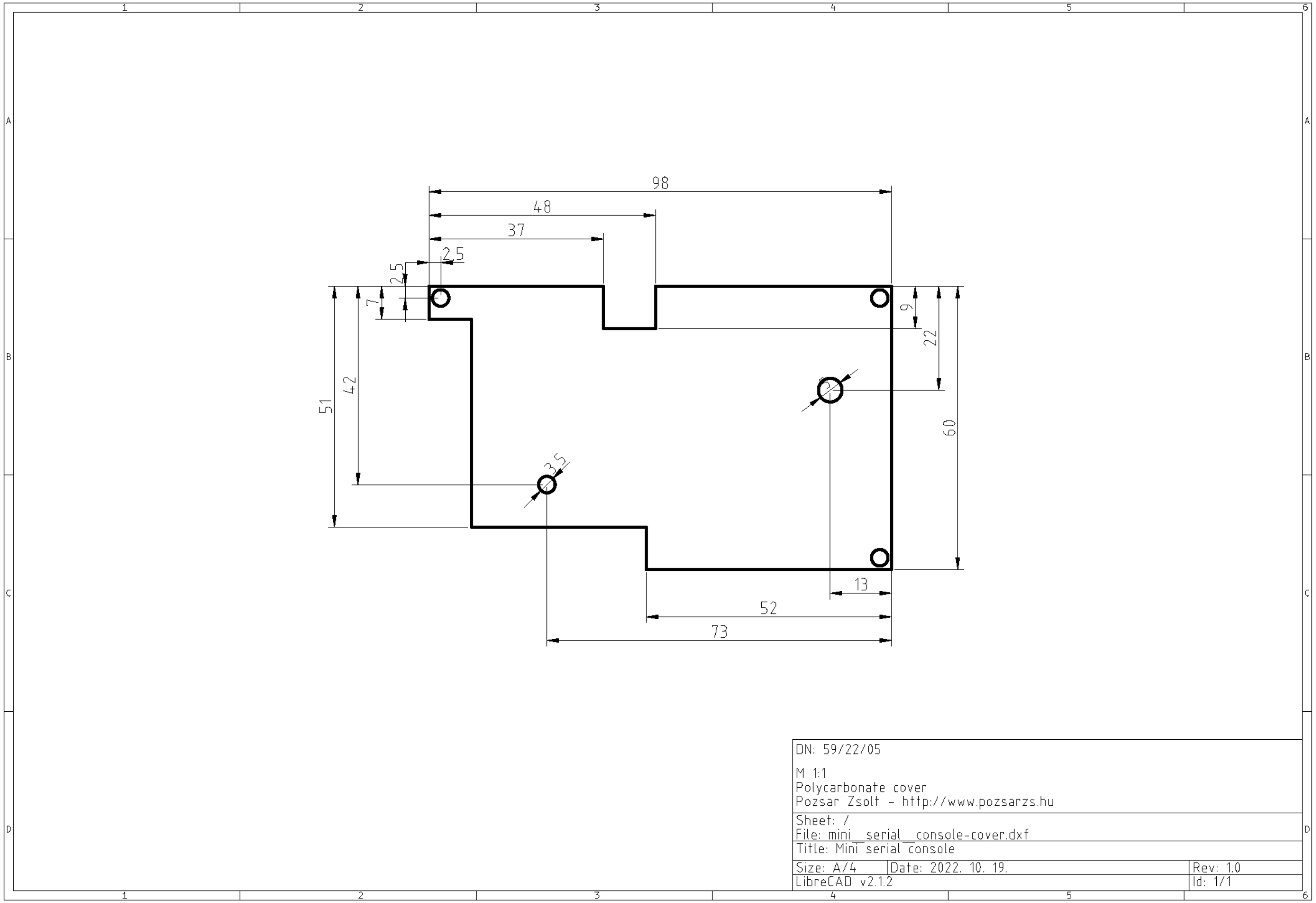


```
2nd line from linux
3rd line from DOS
4th line from linux
5th line from DOS
```



```
Mini serial console
-------------------
sw.: v0.1
(C)2022 Pozsar Zsolt
Initializing...
 * GPIO ports
 * LCD
 * Serial ports:
   #0: 115200 b/s
   #1: 9600 b/s
   #2: 9600 b/s
 * Operation mode: #0
Read a line from serial port #2
Read a line from serial port #1
Button pressed: PB1
Button pressed: PB1
Button pressed: PB0
Button pressed: PB0
Read a line from serial port #2
Read a line from serial port #1
Read a line from serial port #2
Read a line from serial port #1


CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyACM0
```

Custom parts and enclosures



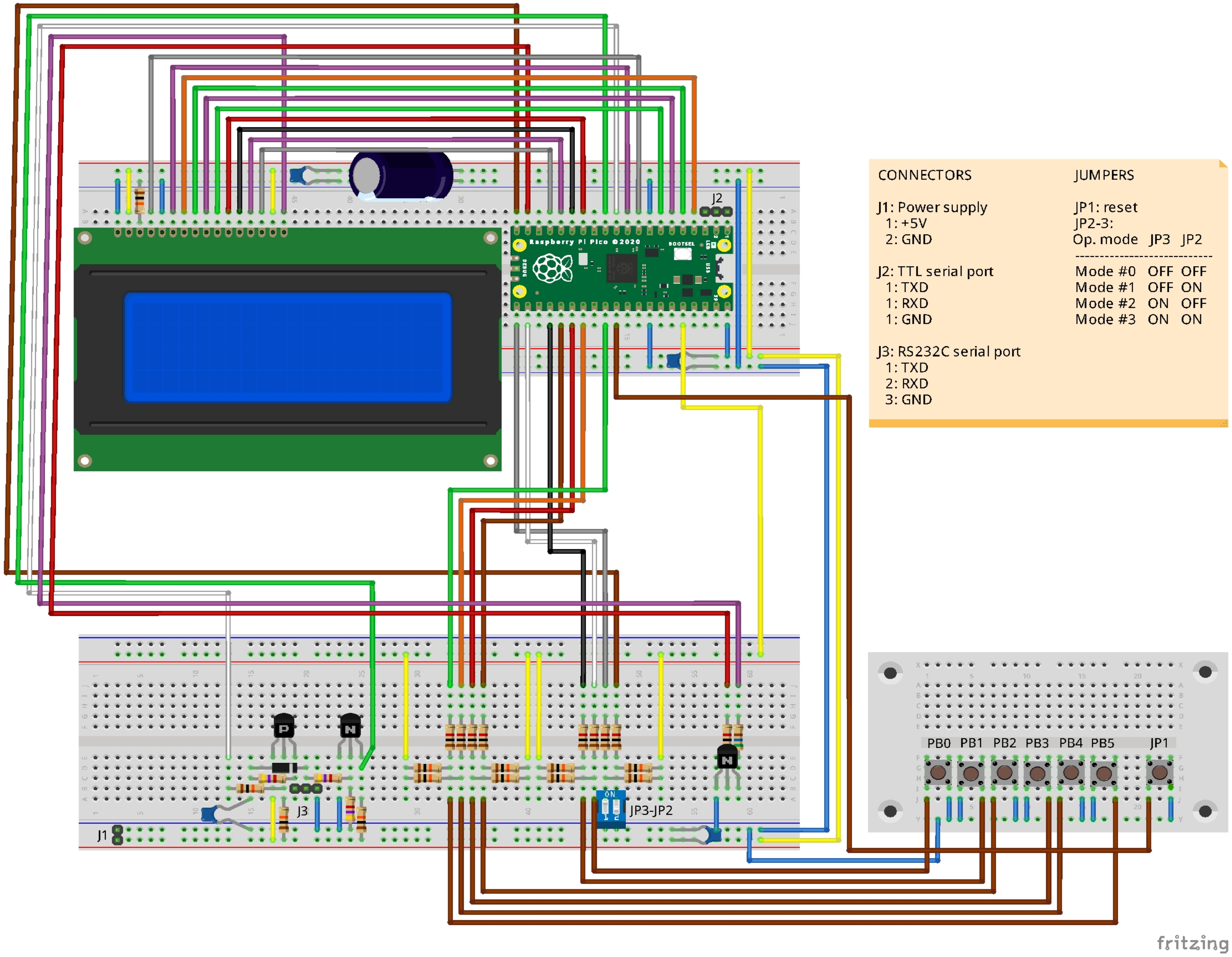
DN: 59/22/05

M 1:1
Polycarbonate cover
Pozsar Zsolt - http://www.pozsarzs.hu

Sheet: /
File: mini__serial__console-cover.dxf
Title: Mini serial console

Size: A/4	Date: 2022. 10. 19.	Rev: 1.0
LibreCAD v2.1.2		Id: 1/1

Schematics



CONNECTORS

J1: Power supply
  1: +5V
  2: GND

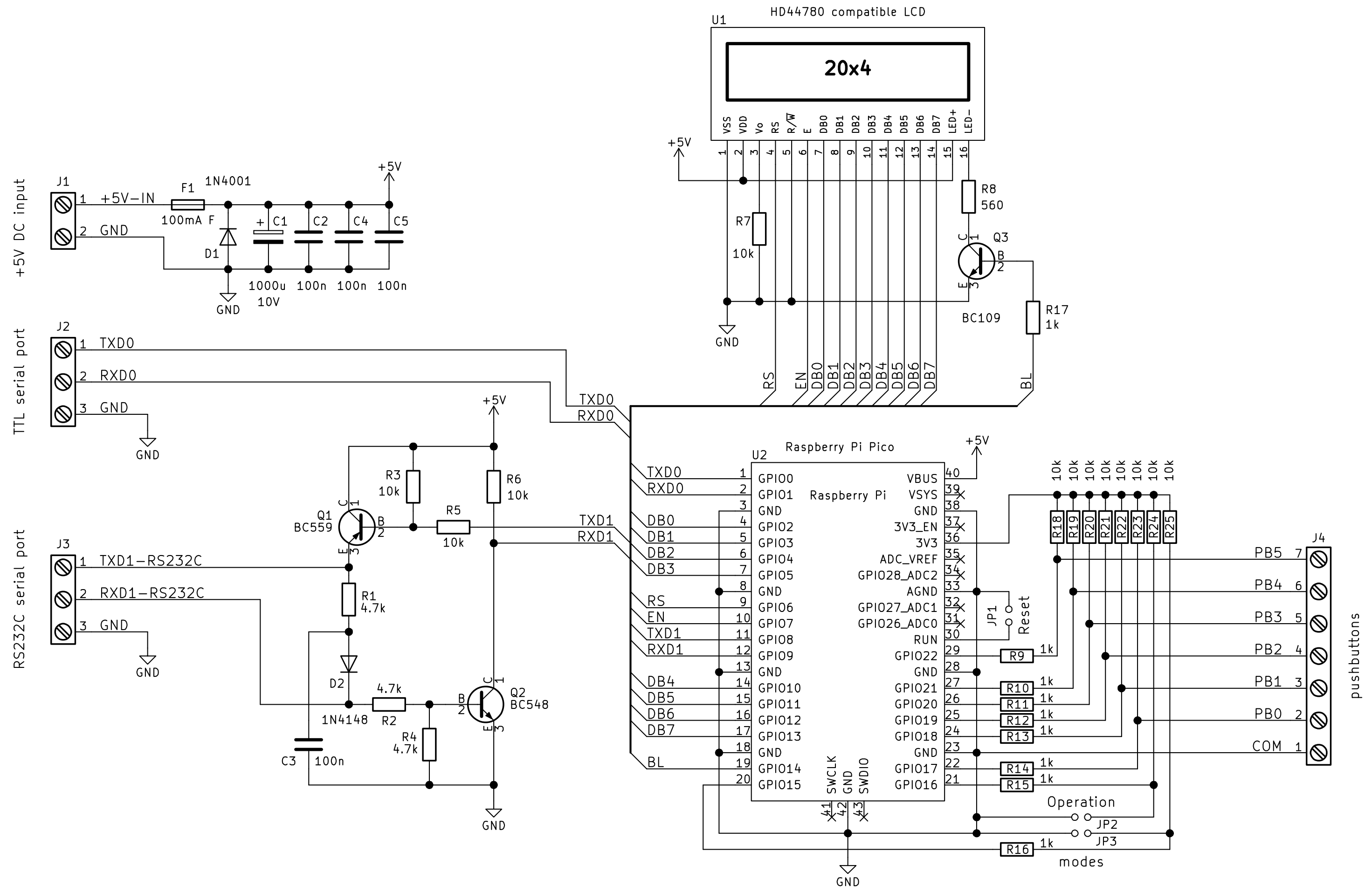
J2: TTL serial port
  1: TXD
  1: RXD
  1: GND

J3: RS232C serial port
  1: TXD
  2: RXD
  3: GND

JUMPERS

JP1: reset
JP2-3:

Op. mode	JP3	JP2
Mode #0	OFF	OFF
Mode #1	OFF	ON
Mode #2	ON	OFF
Mode #3	ON	ON


U1
HD44780 compatible LCD
20x4
VSS VDD Vo RS R/W E DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7 LED+ LED-
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
+5V
R7
10k
GND
R8
560
Q3
BC109
R17
1k
RS EN DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7 BL
+5V DC input
J1
1 +5V-IN
2 GND
F1
100mA F
1N4001
D1
C1 1000u 10V
C2 100n
C4 100n
C5 100n
GND
TTL serial port
J2
1 TXD0
2 RXD0
3 GND
TXD0
RXD0
RS232C serial port
J3
1 TXD1-RS232C
2 RXD1-RS232C
3 GND
Q1
BC559
R3
10k
R6
10k
R5
10k
TXD1
RXD1
R1
4.7k
D2
1N4148
R2
4.7k
R4
4.7k
C3
100n
Q2
BC548
GND
U2
Raspberry Pi Pico
Raspberry Pi
GPIO0 GPIO1 GND GPIO2 GPIO3 GPIO4 GPIO5 GND GPIO6 GPIO7 GPIO8 GPIO9 GND GPIO10 GPIO11 GPIO12 GPIO13 GND GPIO14 GPIO15
VBUS VSYS GND 3V3_EN 3V3 ADC_VREF GPIO28_ADC2 AGND GPIO27_ADC1 GPIO26_ADC0 RUN GPIO22 GND GPIO21 GPIO20 GPIO19 GPIO18 GND GPIO17 GPIO16
SWCLK GND SWDIO
TXD0 RXD0 DB0 DB1 DB2 DB3 RS EN TXD1 RXD1 DB4 DB5 DB6 DB7 BL
JP1
Reset
R9 R10 R11 R12 R13 R14 R15 R16
1k
R18 R19 R20 R21 R22 R23 R24 R25
10k
J4
PB5 7
PB4 6
PB3 5
PB2 4
PB1 3
PB0 2
COM 1
pushbuttons
Operation
JP2
JP3
modes
GND

Operation modes:

	JP3	JP2
Mode #0	O	O
Mode #1	O	C
Mode #2	C	O
Mode #3	C	C

DN: 59/22/02

http://www.pozsarzs.hu
Pozsar Zsolt

Sheet: /
File: mini_serial_console.sch

Title: Mini serial console

Size: A4	Date: 2022-09-30	**Rev: 1.0**
KiCad E.D.A. kicad 4.0.5+dfsg1-4+deb9u1		Id: 1/1

Code

Software

https://github.com/pozsarzs/mini_serial_console-sw

Hardware

https://github.com/pozsarzs/mini_serial_console-hw

Credits

Zsolt Pozsár

10 projects • 1 follower
I am not professional programmer, I specialize in electronics and electrical installation.
Contact