

Wszystko czego nie chcecie wiedzieć o elektronicznych maszynach do pisania

Ale i tak wam powiemy

**Krzysiek Mazur, Mariusz Różycki, 1 kwietnia 2025 r.
z pomocą dr. inż. Piotra Nowakowskiego**

Poznajcie Erikę



Poznajcie Optime



PRZED OTWARCIEM
OBUDOWY
ODECZYŃ
OD SIECI !!!

OPTIMA

SP 20

MODE

CODE



Motywacja

- Mariusz miał pomysł...
- ...żeby pisać więcej *rzeczy*...
- ...ale bez rozpraszaczy na komputerze...
- ...i nie długopisem, bo to nie średniowiecze



"Y" zamienione
miejscami z "Z"

Polskie znaki na
osobnych klawiszach...

...a do tego nie da się
ich pisać z shiftem

„A po co ci to?”

Krzysztof Mazur, 28 grudnia 2024 r.

„A może dalekopis!”

Krzysztof Mazur, 28 grudnia 2024 r., 5 minut później



Trochę duży...



PRZED OTWARCIEM
URZĄDZENIA
ODECZYNIĆ
OD SIECI !!!

OPTIMA

SP 20

Polskie znaki na osobnych klawiszach...

"Y" zamienione miejscami z "Z"

...a do tego nie da się ich pisać z shiftem

Można podpiąć kabelki!



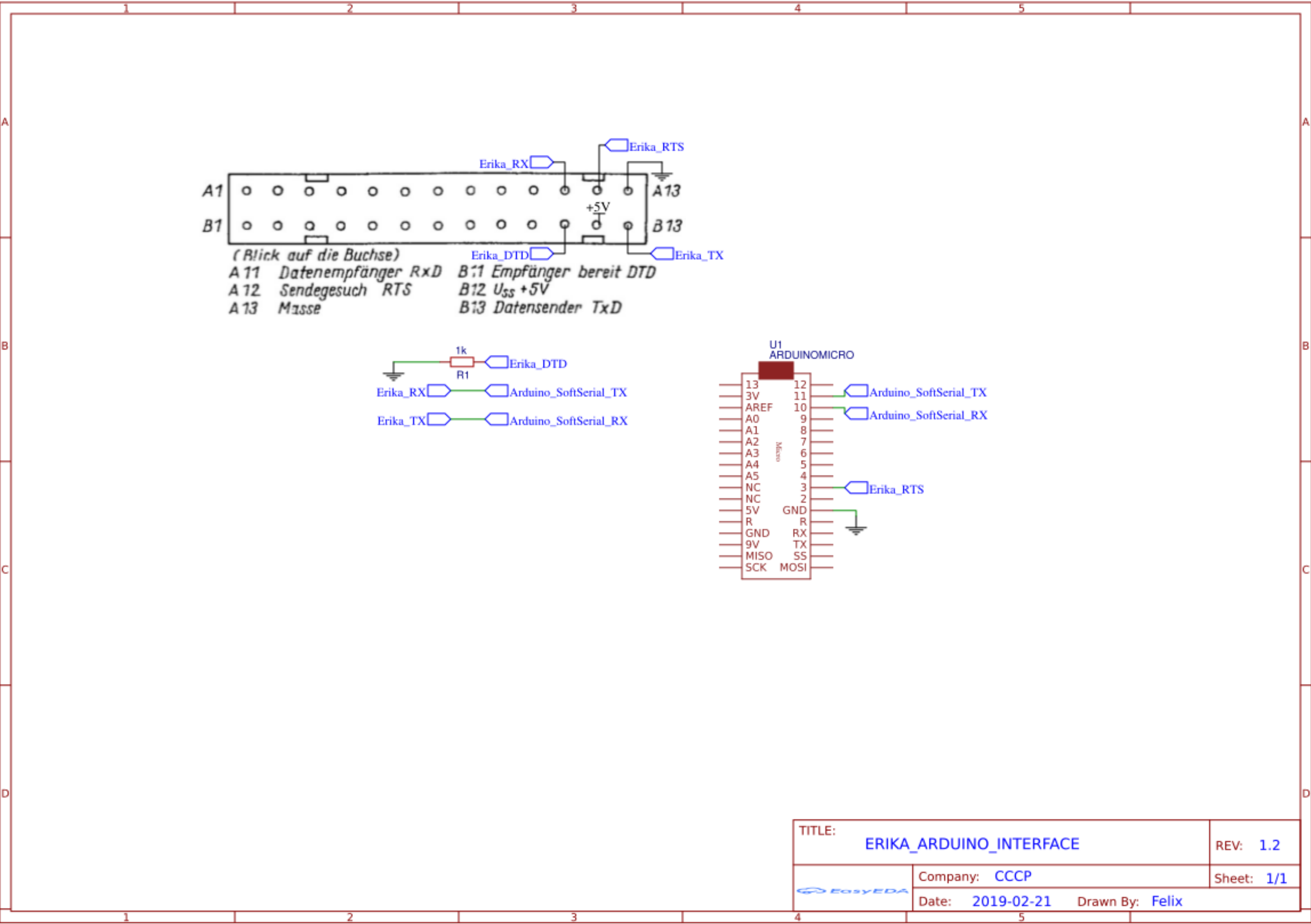


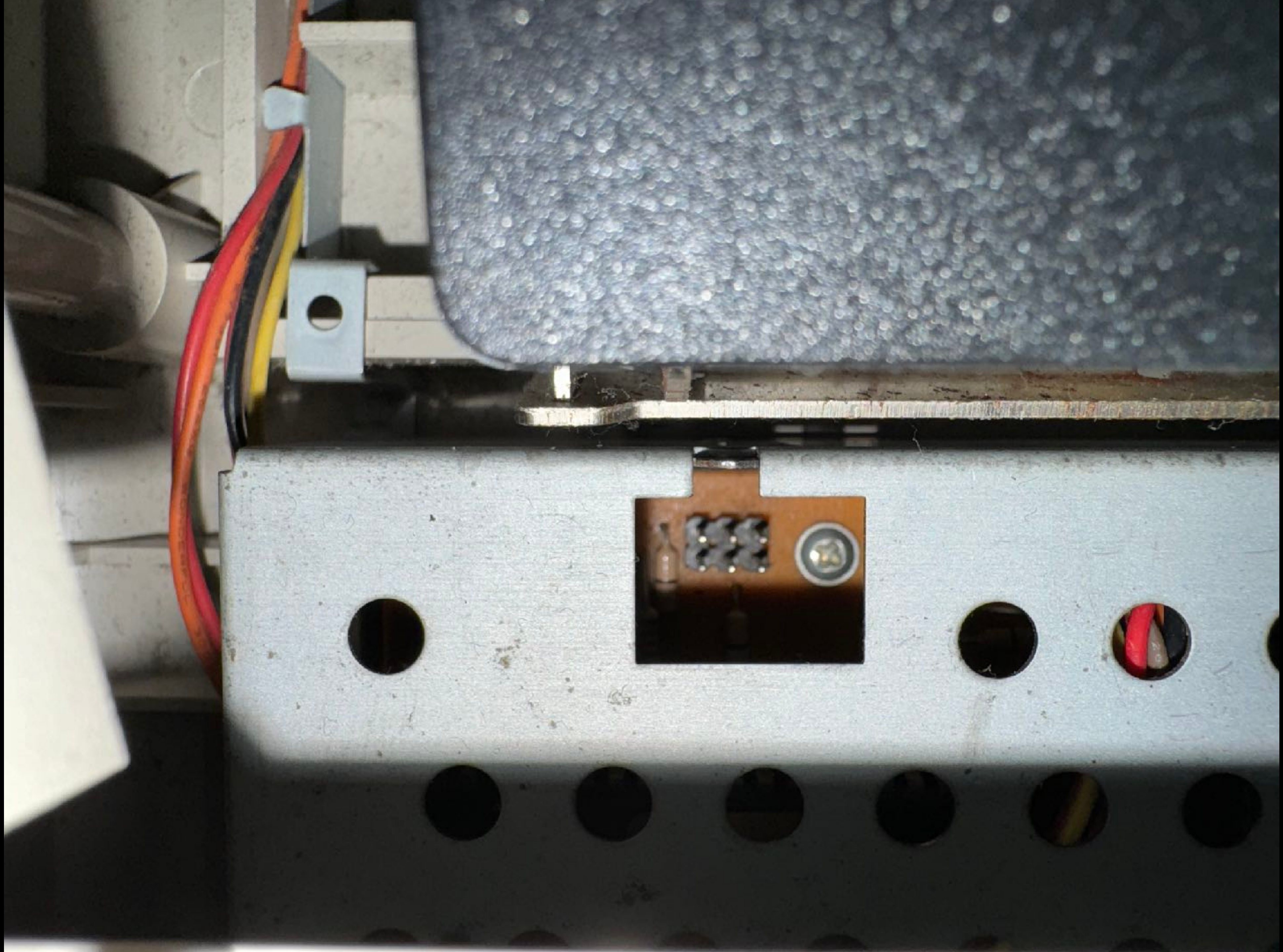
To może klawiaturę?!

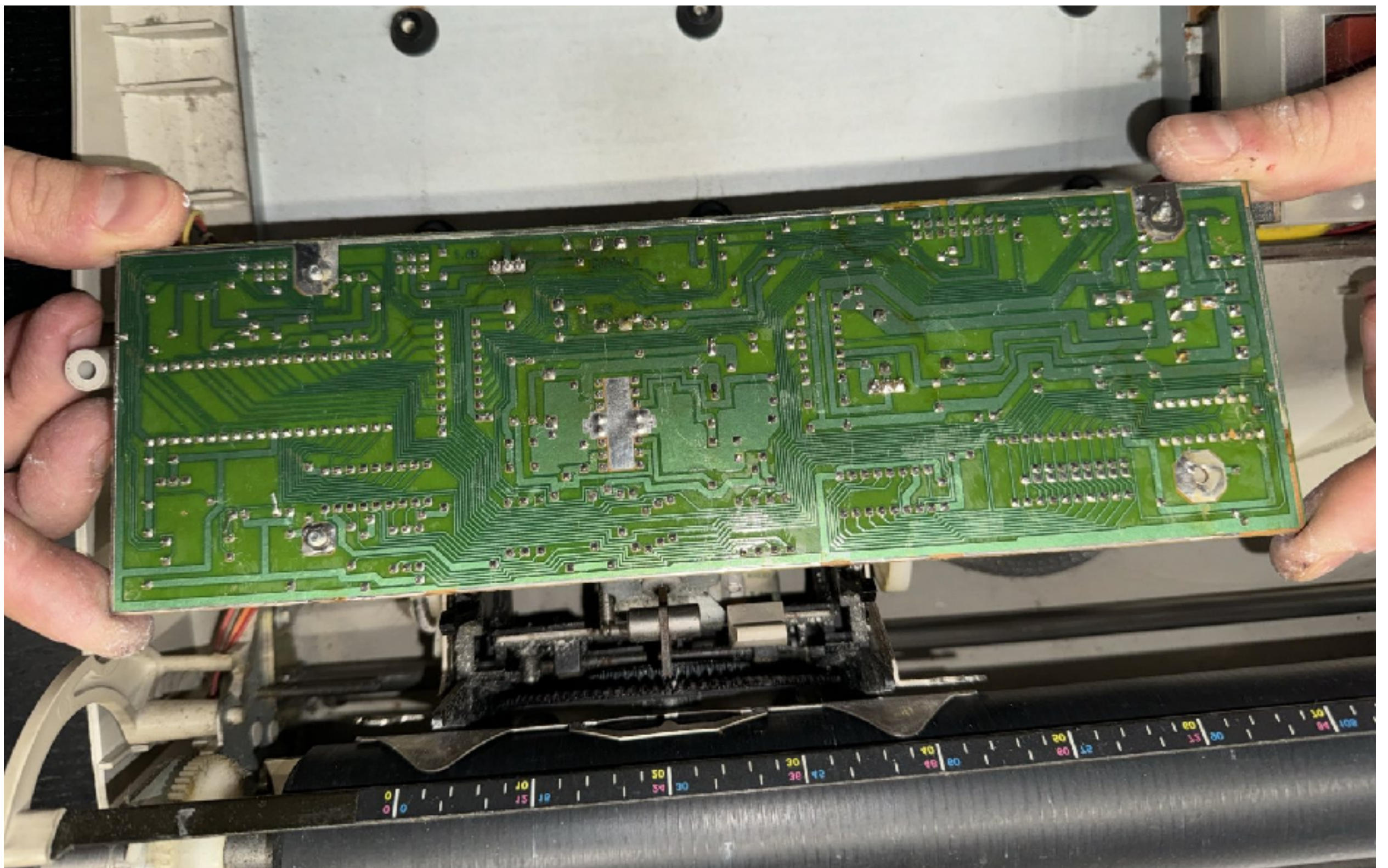
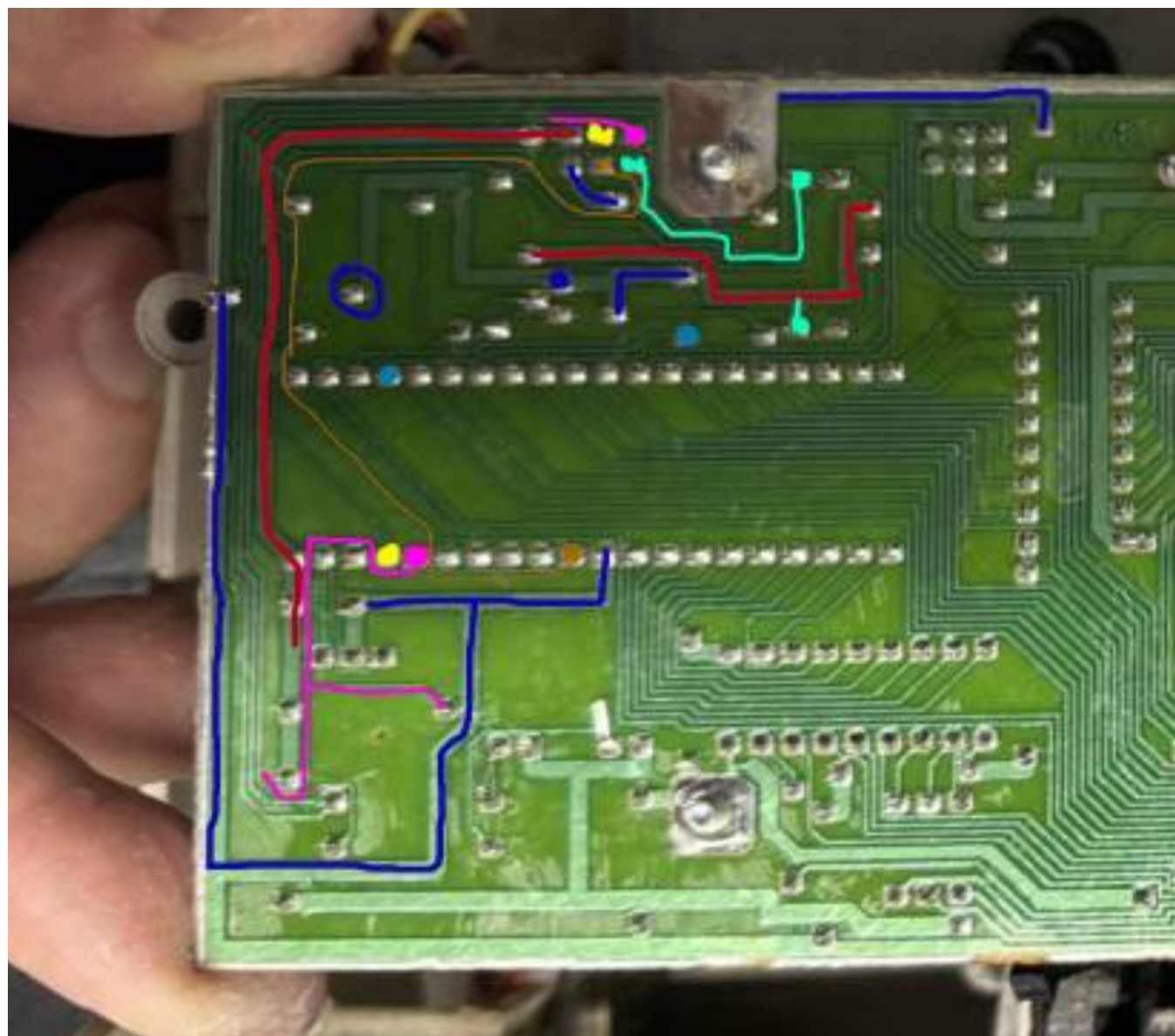
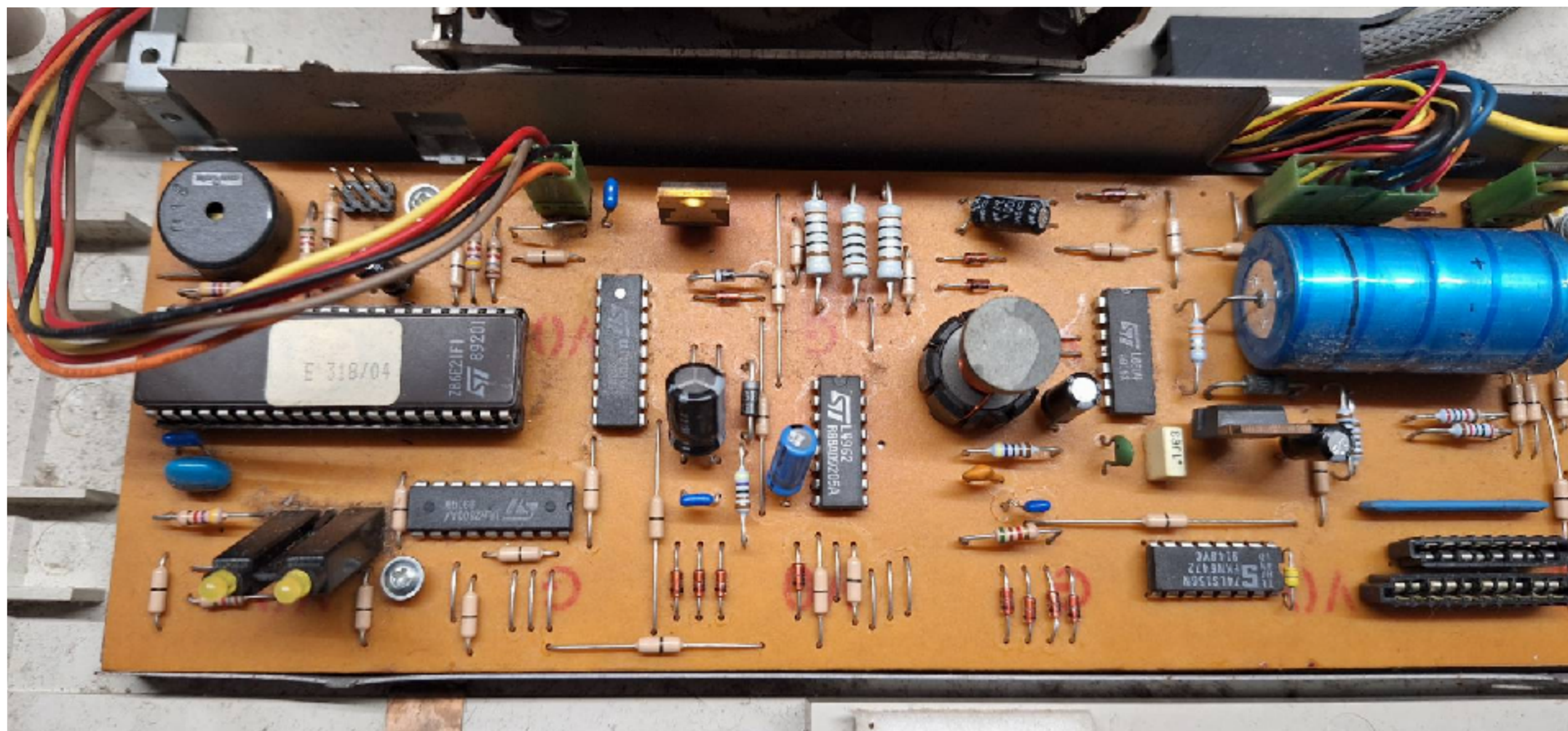


Co daje nam te kabelki?





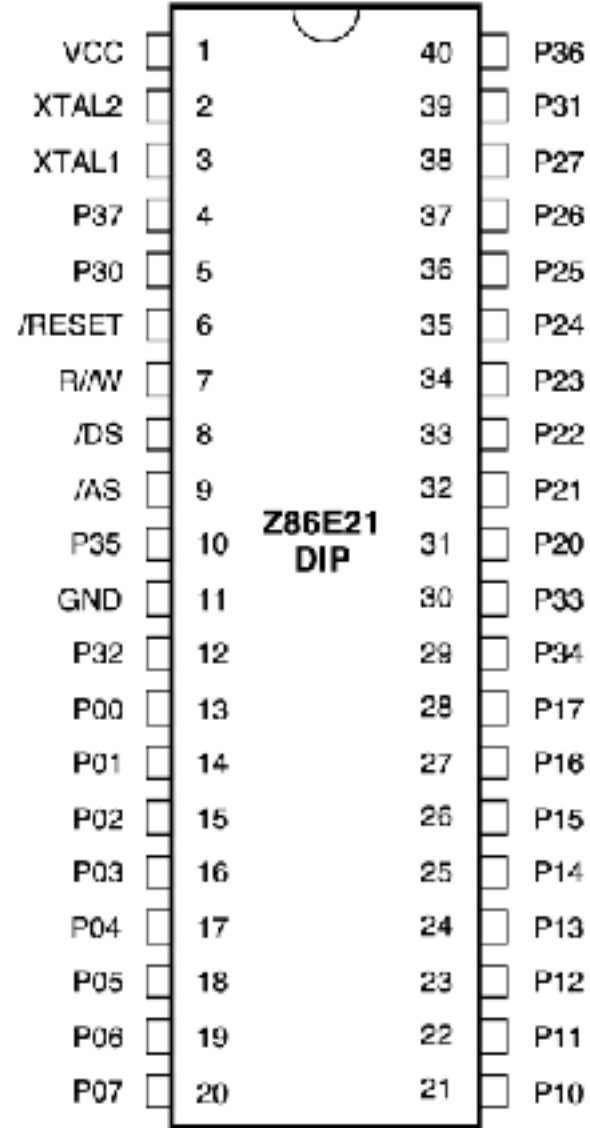




Piny Z8

- VCC
- GND
- P23 (pulled high)
- P30
- P35
- P37

PIN DESCRIPTION Standard Mode



40-Lead DIP Pin Assignments

40-Lead DIP Pin Identification

Pin #	Symbol	Function	Direction	Pin #	Symbol	Function	Direction
1	V _{CC}	Power Supply	Input	11	GND	Ground, GND	Input
2	XTAL2	Crystal, Oscillator Clock	Output	12	P32	Port 3 pin 2	Input
3	XTAL1	Crystal, Oscillator Clock	Input	13-20	P00-P07	Port 0 pin 0, 1, 2, 3, 4, 5, 6, 7	In/Output
4	P37	Port 3 pin 7	Output	21-28	P10-P17	Port 1 pin 0, 1, 2, 3, 4, 5, 6, 7	In/Output
5	P30	Port 3 pin 0	Input	29	P34	Port 3 pin 4	Output
6	/RESET	Reset	Input	30	P33	Port 3 pin 3	Input
7	R/W	Read/Write	Output	31-38	P20-P27	Port 2 pin 0, 1, 2, 3, 4, 5, 6, 7	In/Output
8	/DS	Data Strobe	Output	39	P31	Port 3 pin 1	Input
9	/AS	Address Strobe	Output	40	P36	Port 3 pin 6	Output
10	P35	Port 3 pin 5	Output				

Serial Input/Output

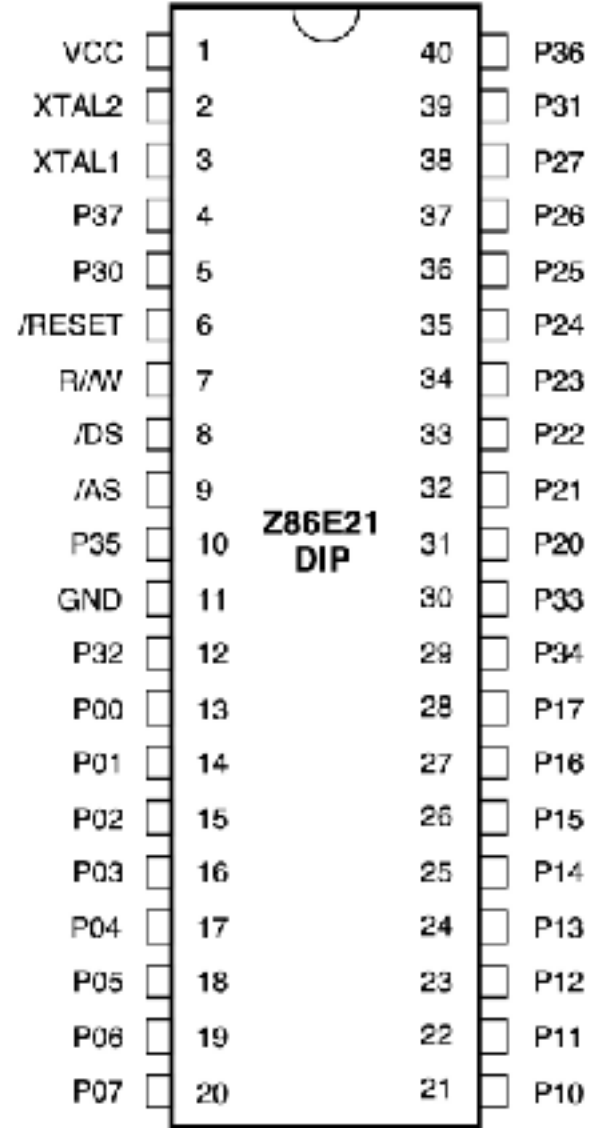
UART Introduction

Some Z8[®] CPU microcontrollers contain an on-board full-duplex Universal Asynchronous Receiver/Transmitter (UART) for data communications. The UART consists of a serial input/output (SIO) Register located at address F0h, and its associated control logic (see [Figure 105](#)). The SIO is actually two registers, the receiver buffer and the transmitter buffer, which are used in conjunction with Counter/Timer T0 and Port 3 I/O lines P30 (input) and P37 (output). Counter/Timer T0 provides the clock input for control of the data rates.

Piny Z8

- VCC
- GND
- P23 (pulled high)
- P30 -> RX
- P35
- P37 -> TX

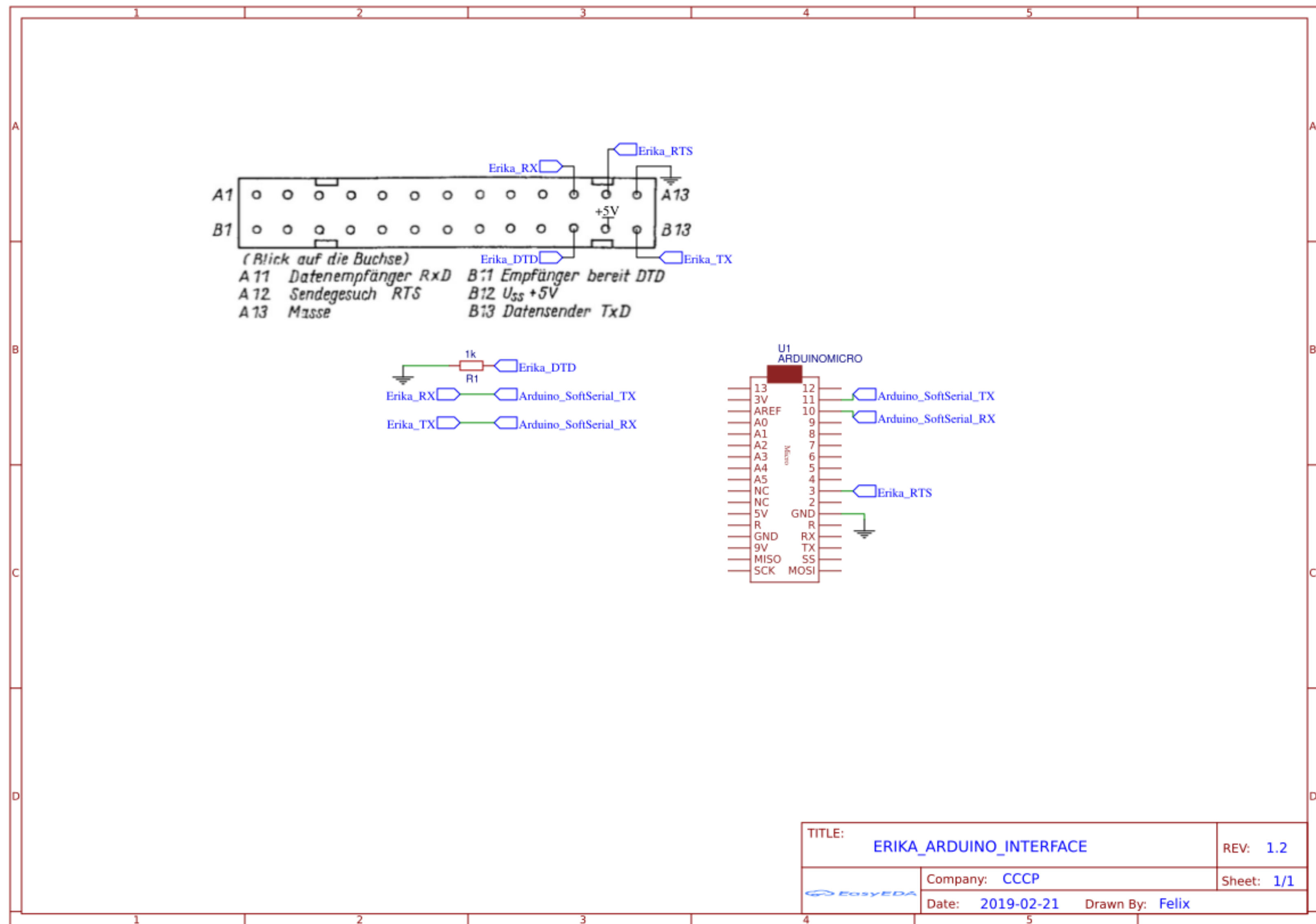
PIN DESCRIPTION Standard Mode



40-Lead DIP Pin Assignments

40-Lead DIP Pin Identification

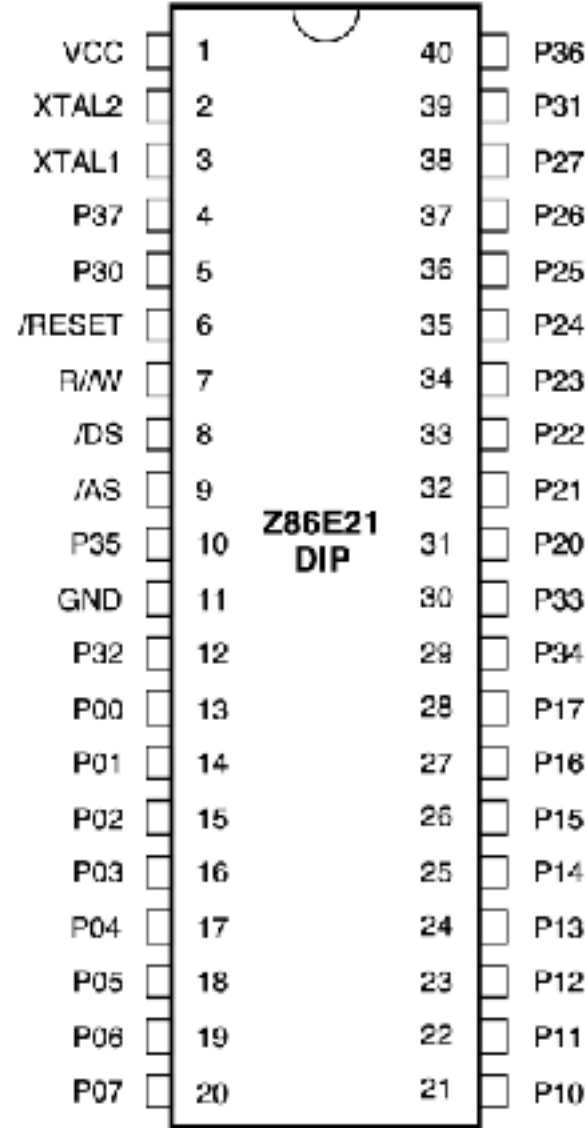
Pin #	Symbol	Function	Direction	Pin #	Symbol	Function	Direction
1	V _{CC}	Power Supply	Input	11	GND	Ground, GND	Input
2	XTAL2	Crystal, Oscillator Clock	Output	12	P32	Port 3 pin 2	Input
3	XTAL1	Crystal, Oscillator Clock	Input	13-20	P00-P07	Port 0 pin 0, 1, 2, 3, 4, 5, 6, 7	In/Output
4	P37	Port 3 pin 7	Output	21-28	P10-P17	Port 1 pin 0, 1, 2, 3, 4, 5, 6, 7	In/Output
5	P30	Port 3 pin 0	Input	29	P34	Port 3 pin 4	Output
6	/RESET	Reset	Input	30	P33	Port 3 pin 3	Input
7	R/W	Read/Write	Output	31-38	P20-P27	Port 2 pin 0, 1, 2, 3, 4, 5, 6, 7	In/Output
8	/DS	Data Strobe	Output	39	P31	Port 3 pin 1	Input
9	/AS	Address Strobe	Output	40	P36	Port 3 pin 6	Output
10	P35	Port 3 pin 5	Output				



Piny Z8

- VCC
- GND
- P23 (pulled high) -> DTD
- P30 -> RX
- P35 -> RTS
- P37 -> TX

PIN DESCRIPTION Standard Mode



40-Lead DIP Pin Assignments

40-Lead DIP Pin Identification

Pin #	Symbol	Function	Direction	Pin #	Symbol	Function	Direction
1	V _{CC}	Power Supply	Input	11	GND	Ground, GND	Input
2	XTAL2	Crystal, Oscillator Clock	Output	12	P32	Port 3 pin 2	Input
3	XTAL1	Crystal, Oscillator Clock	Input	13-20	P00-P07	Port 0 pin 0, 1, 2, 3, 4, 5, 6, 7	In/Output
4	P37	Port 3 pin 7	Output	21-28	P10-P17	Port 1 pin 0, 1, 2, 3, 4, 5, 6, 7	In/Output
5	P30	Port 3 pin 0	Input	29	P34	Port 3 pin 4	Output
6	/RESET	Reset	Input	30	P33	Port 3 pin 3	Input
7	R/W	Read/Write	Output	31-38	P20-P27	Port 2 pin 0, 1, 2, 3, 4, 5, 6, 7	In/Output
8	/DS	Data Strobe	Output	39	P31	Port 3 pin 1	Input
9	/AS	Address Strobe	Output	40	P36	Port 3 pin 6	Output
10	P35	Port 3 pin 5	Output				

VCC

TX

RX

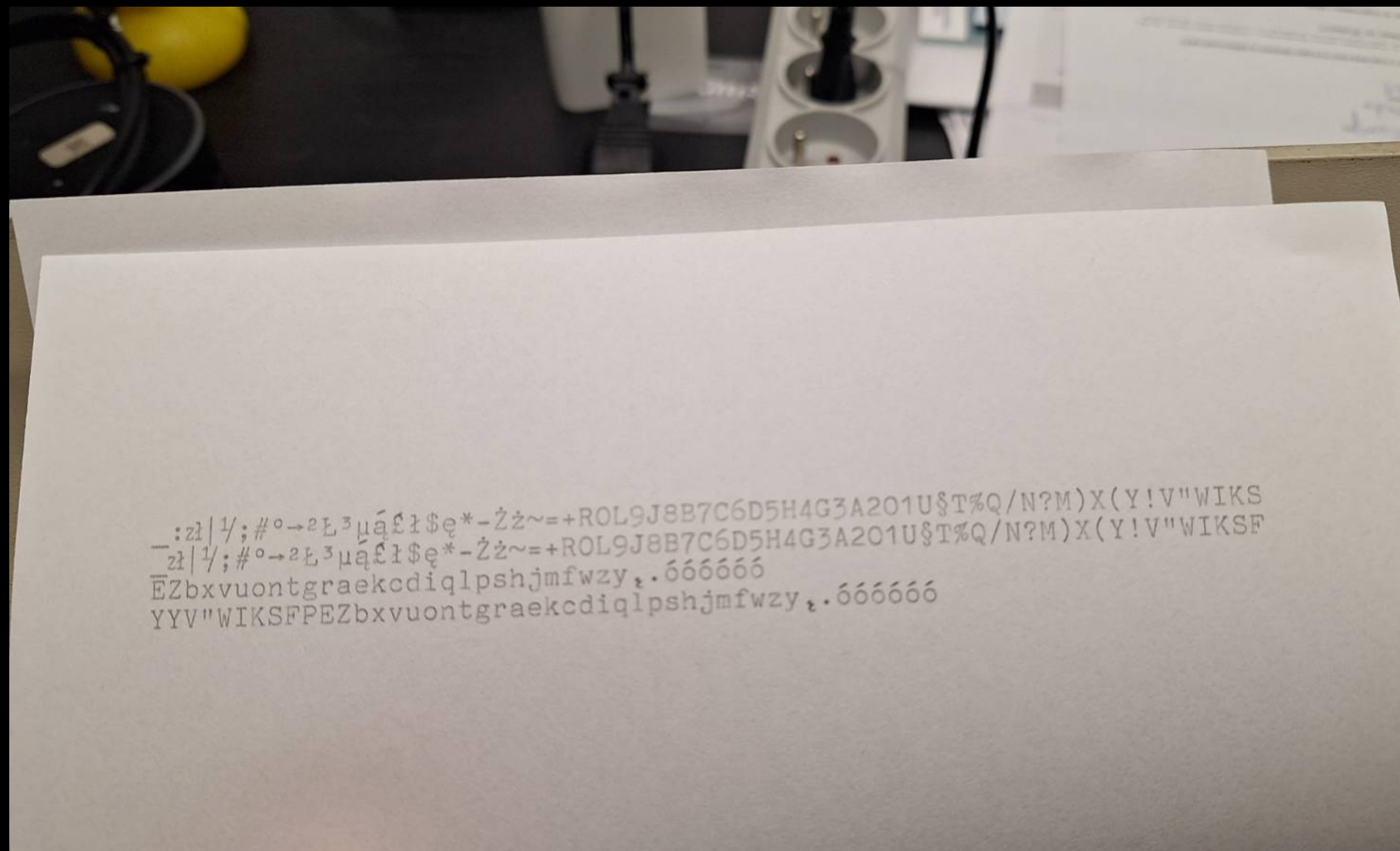
GND

RTS

DTD

:zł|1/;#°→2Ł³μā£ł\$ē*-Żż~+=ROL9J8B7C6D5H4G3A2O1U\$T%Q/N?M)X(Y!V"WIKS
—zł|1/;#°→2Ł³μā£ł\$ē*-Żż~+=ROL9J8B7C6D5H4G3A2O1U\$T%Q/N?M)X(Y!V"WIKSF
EZbxvuontgraekcdiqlpshjmfwzy, .óóóóóó
YYV"WIKSFPEZbxvuontgraekcdiqlpshjmfwzy, .óóóóóó





Kodowanie znaków wg Eriki

- 1-100 to znaki na rozetce, w takiej kolejności jak na rozetce
- 101-105 to ó, ś, ń, ć, ź
- 106-112 wydają się nie być używane
- 113-175 to kody sterujące
 - <https://hc-ddr.hucki.net/wiki/doku.php/z9001/erweiterungen/s3004#steuer codes>
 - W tym spacja, nowa linia, włączanie/wyłączanie klawiatury, tabulatory, korekcja, zmiana interlinii czy gęstości znaków

Próba pierwsza

Użycie klawiatury Eriki przeciwko niej

Próba „przeprogramowania” klawiatury

Pierwsze kroki

- Komunikacja z Eriką jest dwustronna: drukuje otrzymane znaki, ale też wysyła kody znaków pisanych na klawiaturze
- Klawiaturę można wyłączyć: kody naciskanych klawiszy nadal trafiają na Tx, ale symbole nie są drukowane. Drukowane są za to znaki otrzymane na Rx
- Używając odpowiednio przygotowanego pico możemy „przeprogramować” wbudowaną klawiaturę maszyny
- Zamiana miejscami „Y” i „Z” była prosta, gorzej było z implementacją ALTa

Próba „przeprogramowania” klawiatury

Użycie CODE jako ALT

- Klawisz CODE jest modyfikatorem i działa trochę jak ALT
- Przy normalnym działaniu wypisuje niebieskie symbole



Próba „przeprogramowania” klawiatury

Użycie CODE jako ALT

- Klawisz CODE jest modyfikatorem i działa trochę jak ALT
- Przy normalnym działaniu wypisuje niebieskie symbole
- Na UART trafia informacja, że klawisz został naciśnięty razem z CODE
- Ale nie łączy się z SHIFTem...



Próba „przeprogramowania” klawiatury

Użycie klawisza po prawej stronie spacji jako ALT

- Przy normalnym działaniu służy do przesuwania karetki w górę strony
- Zaletą jest to, że jest w tym samym miejscu co prawy ALT
- Kiedy jest trzymany, jest regularnie wysyłany na UART
- W momencie naciśnięcia innego klawisza przestaje być wysyłany
- Napisanie „ść” w „część” wymagałoby naciśnięcia „ALT”+S, puszczenia obu i naciśnięcia „ALT”+C



Próba druga

Klawiatura na USB, sterownik w Ruście

Sterownik w Ruście

Staroafrykańskie słowo oznaczające „nie umiem programować w C”

- Istnieją crate'y pozwalające zaprogramować Pico jako urządzenie HID
 - <https://github.com/rust-embedded-community/usb-device>
 - <https://github.com/embassy-rs/embassy/tree/main/embassy-usb>
- Ale my potrzebujemy użyć Pico jako hosta

Sterownik w Rustcie

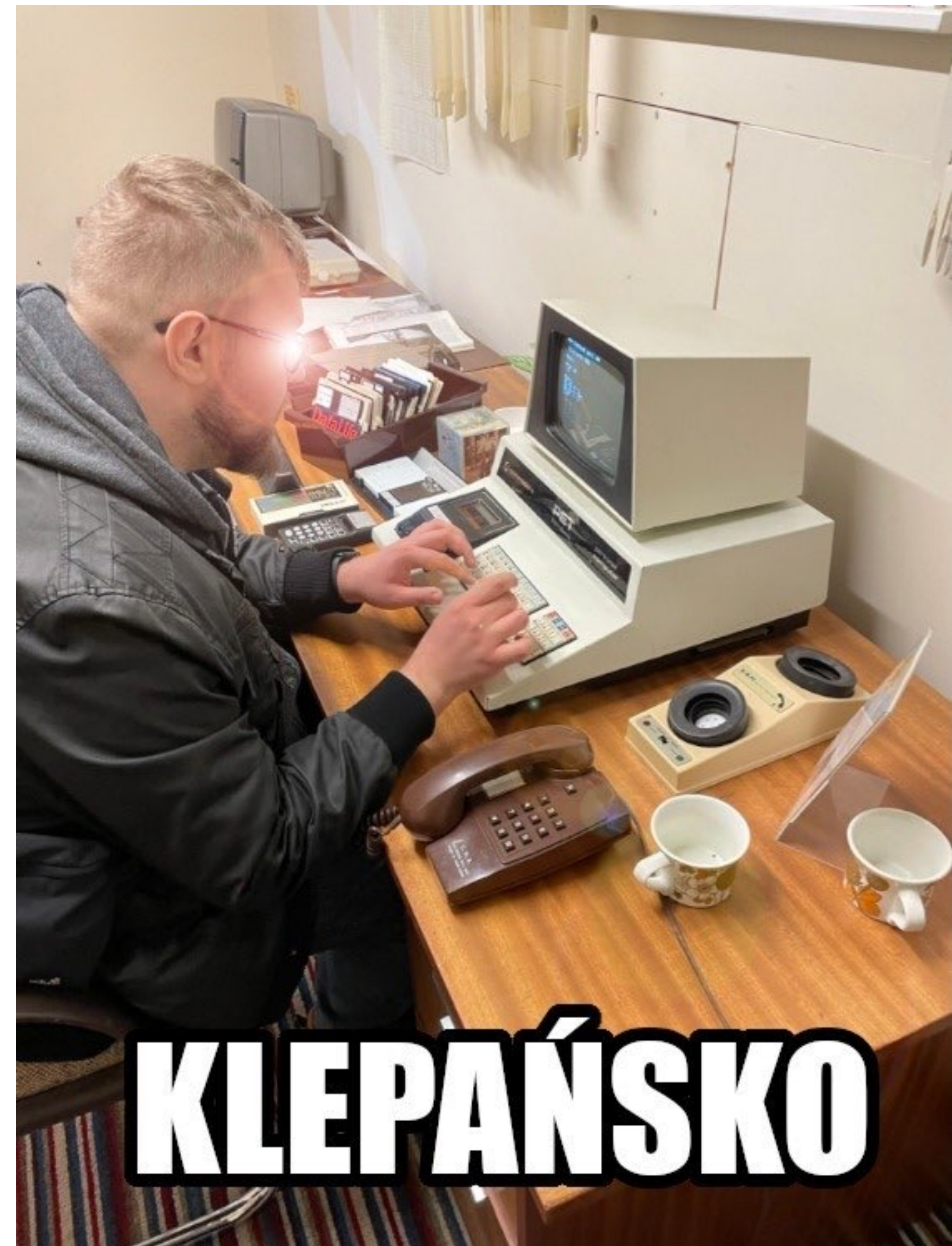
Cotton USB Host

- <https://crates.io/crates/cotton-usb-host>
- Wspiera jedynie USB 1.1, ale to nie problem dla klawiatur
- Wydaje się wspierać jedynie używanie wbudowanego portu micro USB
- Mielіśmy inne plany użycia tego portu

Próba trzecia

Klawiatura na USB

Sterownik w Circuitpythonie



Sterownik w Circuitpythonie

Wsparcie dla klawiatury

- `usb_host` pozwala obsłużyć port USB podpięty do wybranych pinów GPIO
- Dane z podpiętej klawiatury trafiają na `stdin`
- Klawiatura działa out of the box na dowolnych pinach do USB!

```
import usb_host
import board

port = usb_host.Port(board.GP16, board.GP17)

print("Type something")
something = input()
print("You wrote:", something)
```

Sterownik w Circuitpythonie

No ale my jesteśmy trochę "ąę"

- Domyślnie używany jest układ klawiatury en_US
- Wbudowane wsparcie dla keymap pozwala zdefiniować własny układ klawiatury, a więc dodać wsparcie dla polskich znaków
- Pierwszy prototyp powstał w jeden wieczór
- Co może pójść nie tak?

Sterownik w Circuitpythonie

Co natychmiast poszło nie tak

- Keymapa może zawierać jedynie 1-bajtowe wartości
- To ogranicza nas do ASCII, ale od czego jest ISO-8859-2
- Dokumentacja określa, że wartości powinny być między 32 a 126 włącznie
- Prawdopodobnie dlatego, że stdin oczekuje na wejściu UTF-8
- Ale ustawiliśmy wartości >126 , wczytujemy dane binarnie i działa

```
port = usb_host.Port(board.GP16, board.GP17)
uart = busio.UART(tx=board.GP4, rx=board.GP5, baudrate=1200)
```

```
ERIKA_MAP = {
    0xB2: [15], # a
    0xE6: [104], # ć
    0xEA: [20], # ę
    # ...
}
```

```
keymap = array.array("B")
for i in range(384):
    keymap.append(0)
```

```
keymap[256 + 0x04] = 0xB2 # a
keymap[256 + 0x06] = 0xE6 # ć
keymap[256 + 0x08] = 0xEA # ę
# ...
usb_host.set_user_keymap(keymap)
```

```
buf0 = array.array("B")
buf0.append(0)
while True:
    something = sys.stdin.readinto(buf0)
    print(f"Key pressed: {hex(buf0[0])}, erika code: {ERIKA_MAP[buf0[0]]}")
    for code in ERIKA_MAP[buf0[0]]:
        count = uart.write(bytearray(code))
```

Sterownik w Circuitpythonie

Co jeszcze poszło nie tak

- Keymapa składa się z 384 elementów
- Czyli 128 dla klawiszy bez modyfikatorów, 128 dla klawiszy z SHIFTem i 128 dla klawiszy z ALTem
- Ani jednej wartości dla klawisza wciśniętego z SHIFTem i ALTem na raz...
- Czyli mamy ten sam problem co z klawiszem CODE

Sterownik w Circuitpythonie

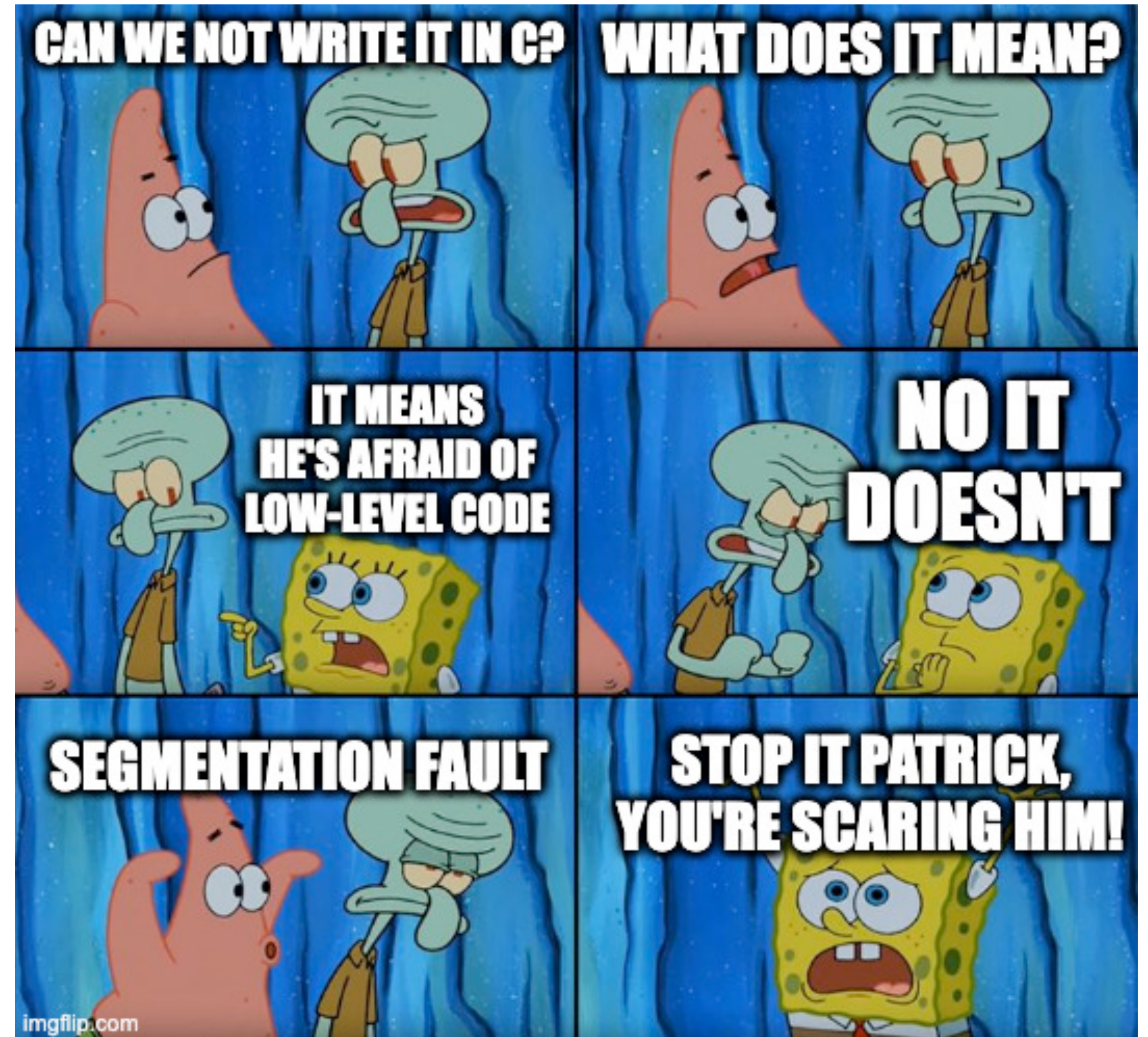
Ale to nie wszystko...

- Circuitpython dla Pico W nie wspiera usb_host
- Oczywiście możemy użyć Circuitpythona dla zwykłego Pico, ale wtedy nie mamy wsparcia dla WiFi i Bluetooth
- Ostatecznie plany wykorzystania WiFi i Bluetooth i tak zeszły na dalszy plan, ale po co pozbawiać się opcji?

Próba czwarta

Klawiatura na USB

Sterownik w C



Sterownik w C

TinyUSB

- Wspiera tryb guest i tryb host
- A nawet oba na raz
- Działa na Pico poprzez driver Pico-PIO-USB
- Nie tylko na wbudowanym porcie USB
- Przykład `host_hid_to_device_cdc` to w zasadzie to czego potrzebujemy

***SURRIPERE
ET MUTARE***

Jak to działa?

Od klawiatury do maszyny

- Z klawiatury otrzymujemy keycody, razem z modyfikatorami
- Mapujemy je na kody ISO-8859-2
- Z nich produkujemy kody Eriki
- Kody Eriki wysyłamy po UART na maszynę

Jak to działa?

Po co to ASCII po drodze?

- Wbudowanym portem USB możemy podpiąć się do komputera
- Po tej stronie Pico zachowuje się jak urządzenie USB
- Przyjmuje tekst w ASCII przez port szeregowy
- Wypisuje go na maszynę
- Wystarczy stworzyć plik tekstowy enkodowany ISO-8859-2
- `<inwokacja.txt >/dev/cu.usbmodem21371`

Jak to działa?

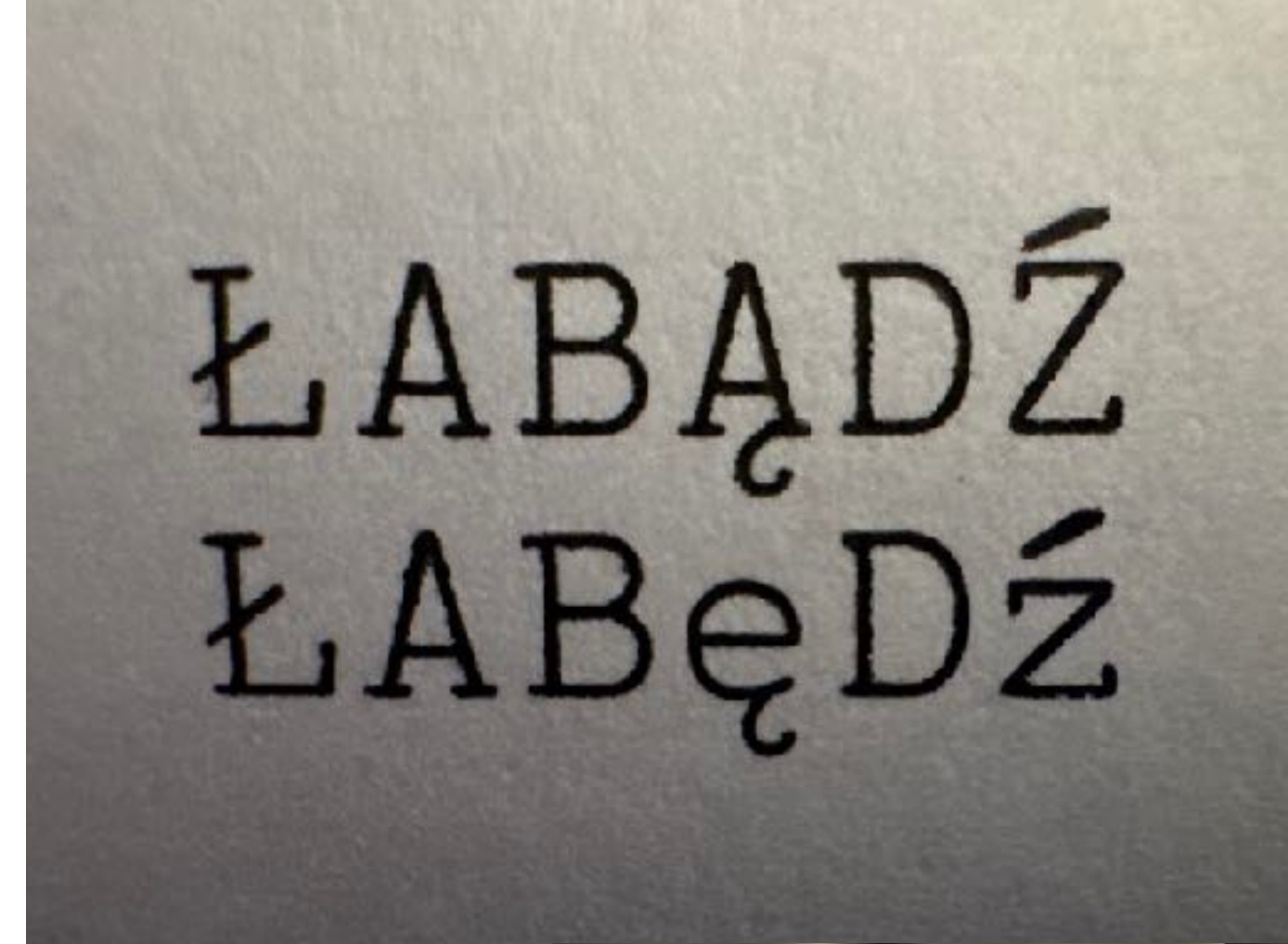
Korekta

- Maszyna wspiera korektę, zapamiętuje ostatnie 20 znaków
- Ale ignoruje znaki przesłane przez UART
- Korektę (i pamięć) trzeba zaimplementować samodzielnie
- Kody 0x8B i 0x8C (Zeichen löschen OFF/ON) dez/aktywują taśmę korektora
- Kody 0x8D i 0x8E (Rückwärtsdruck OFF/ON) dez/aktywują pisanie wstecz

Jak to działa?

Caps Lock

- Maszyna posiada Shift Lock (który jest irytujący)
- Tak czy siak nie działa przez kody sterujące
- Dostajemy informację o wciśniętym Caps Locku z TinyUSB
- Zapamiętujemy jego stan na Pico
- Przy aktywnym Caps Locku zamieniamy małe litery na wielkie
- Ustawiamy lampkę na klawiaturze przez `tuh_hid_set_report`
 - Przy okazji na sztywno włączamy Num Lock



Jak to działa?

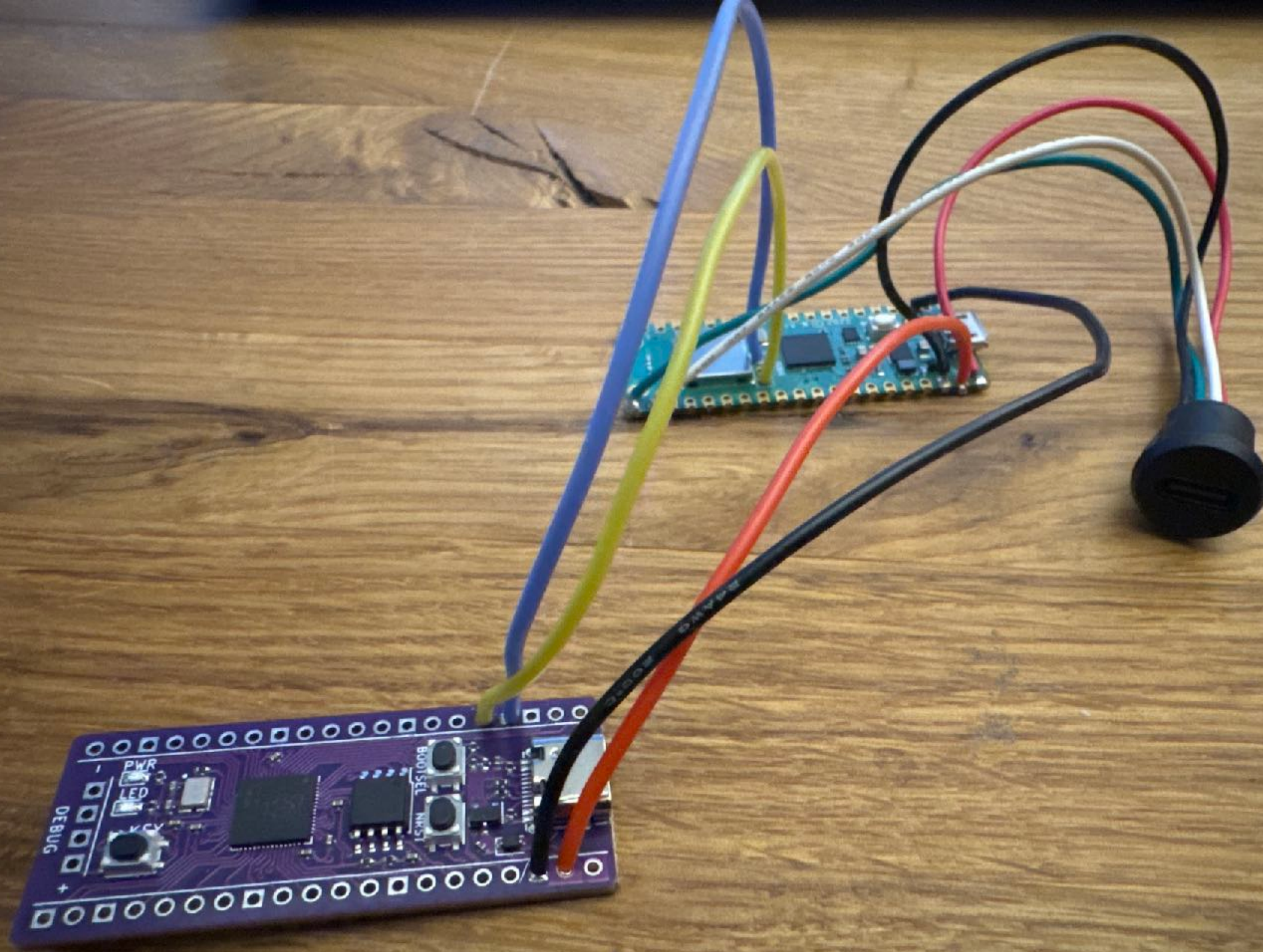
Linuksowy terminal na maszynie

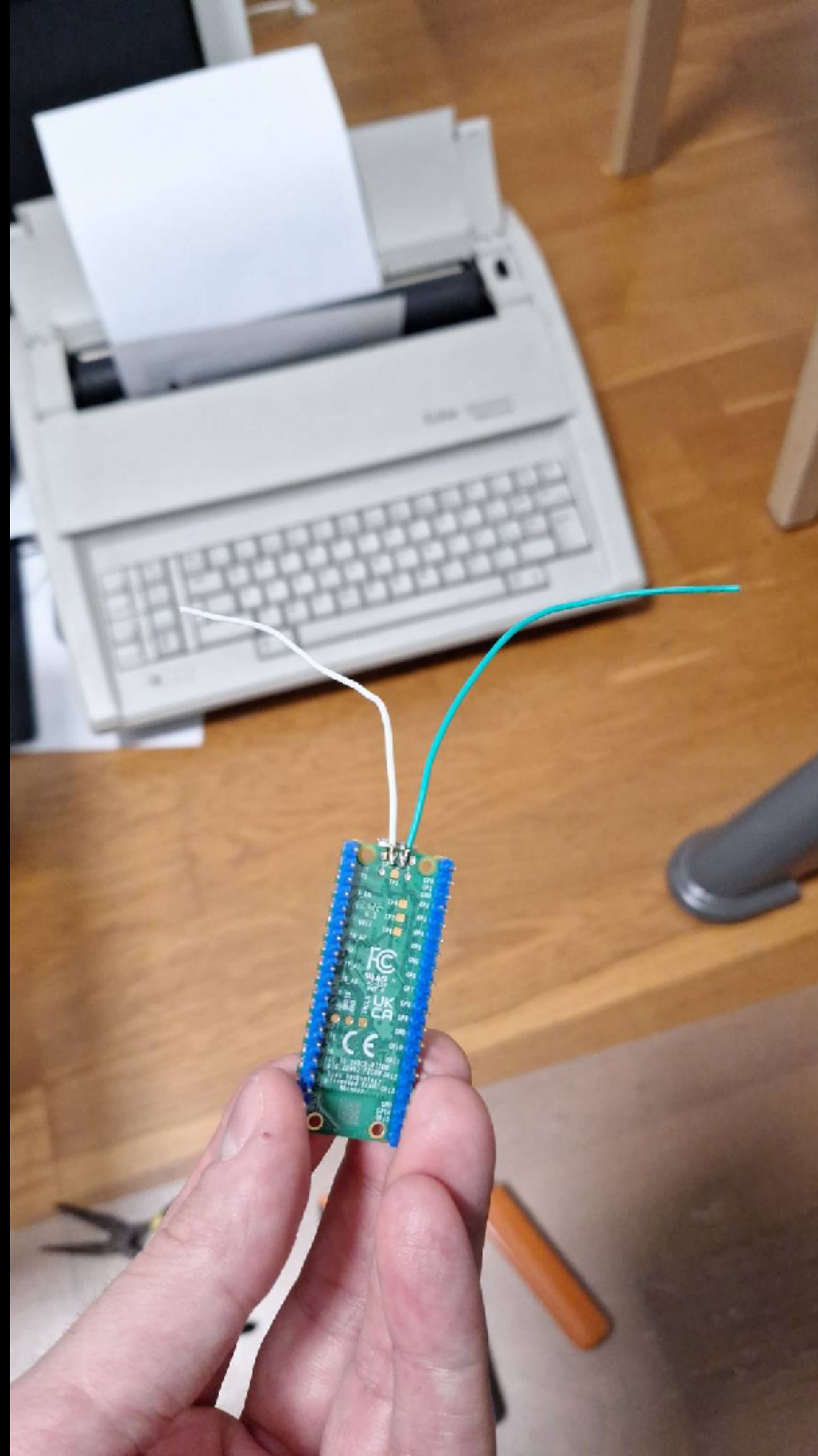
- Przy użyciu udev rules na sztywno przypisujemy urządzeniu o wybranym Vendor ID i Product ID wybrany numer TTY
- Getty uruchamia na tym TTY nową sesję terminala kiedy podłączymy urządzenie
- Drukowanie tekstu przesłanego przez serial port już działa, zostaje przesyłać w drugą stronę informacje o wpisanym przez użytkownika tekście
- Terminal zwraca wpisane przez użytkownika znaki, więc wyłączamy ich drukowanie na maszynie, żeby uniknąć podwajania znaków i pokazywania haseł
- Musimy też obsługiwać (ignorować) escape code'y i wysyłać sygnały (np. Ctrl+C)
- Podpinamy karton papieru *komputerowego* i... działa!

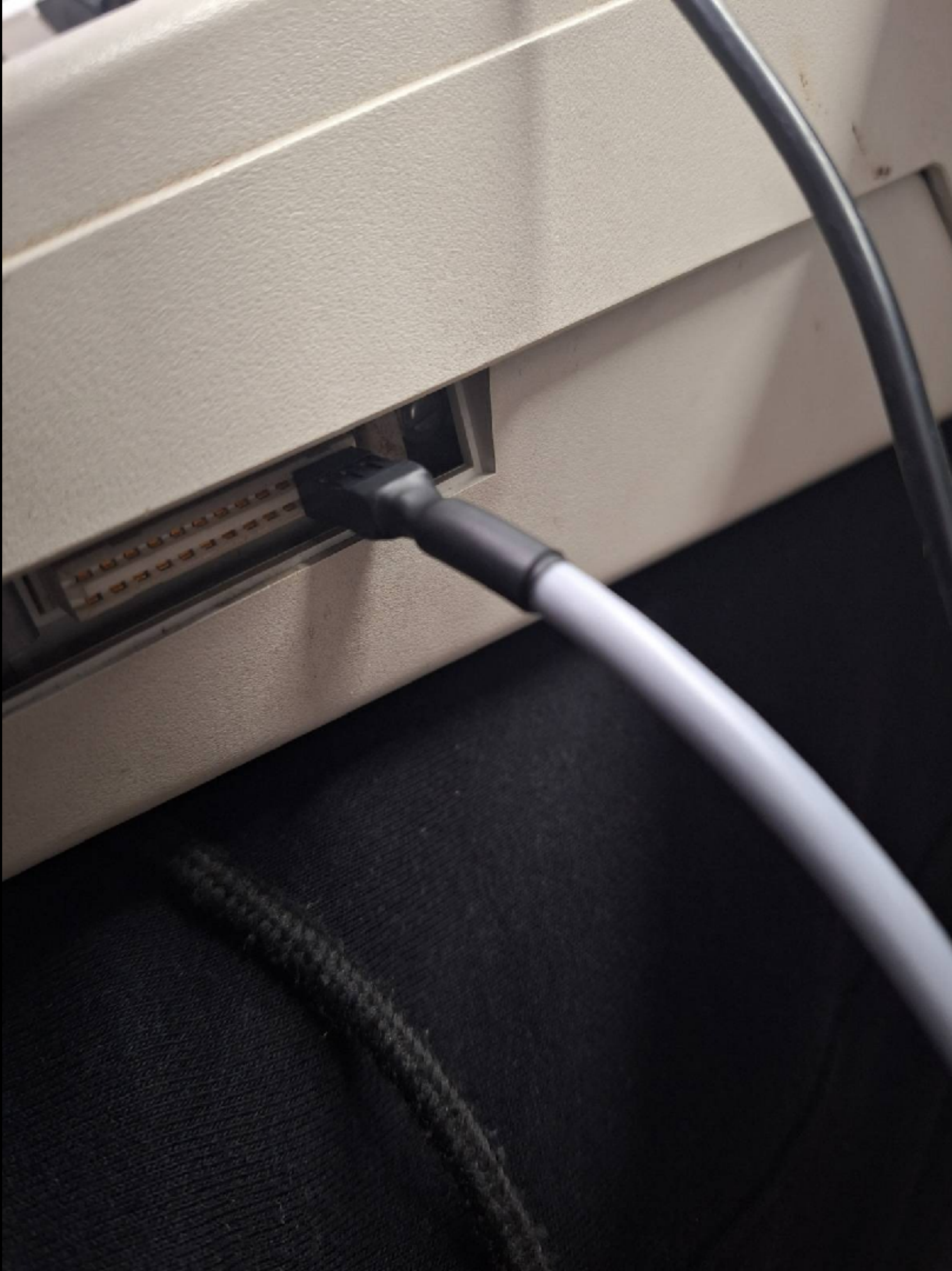
**Nie samym
softwarem żyje
człowiek**

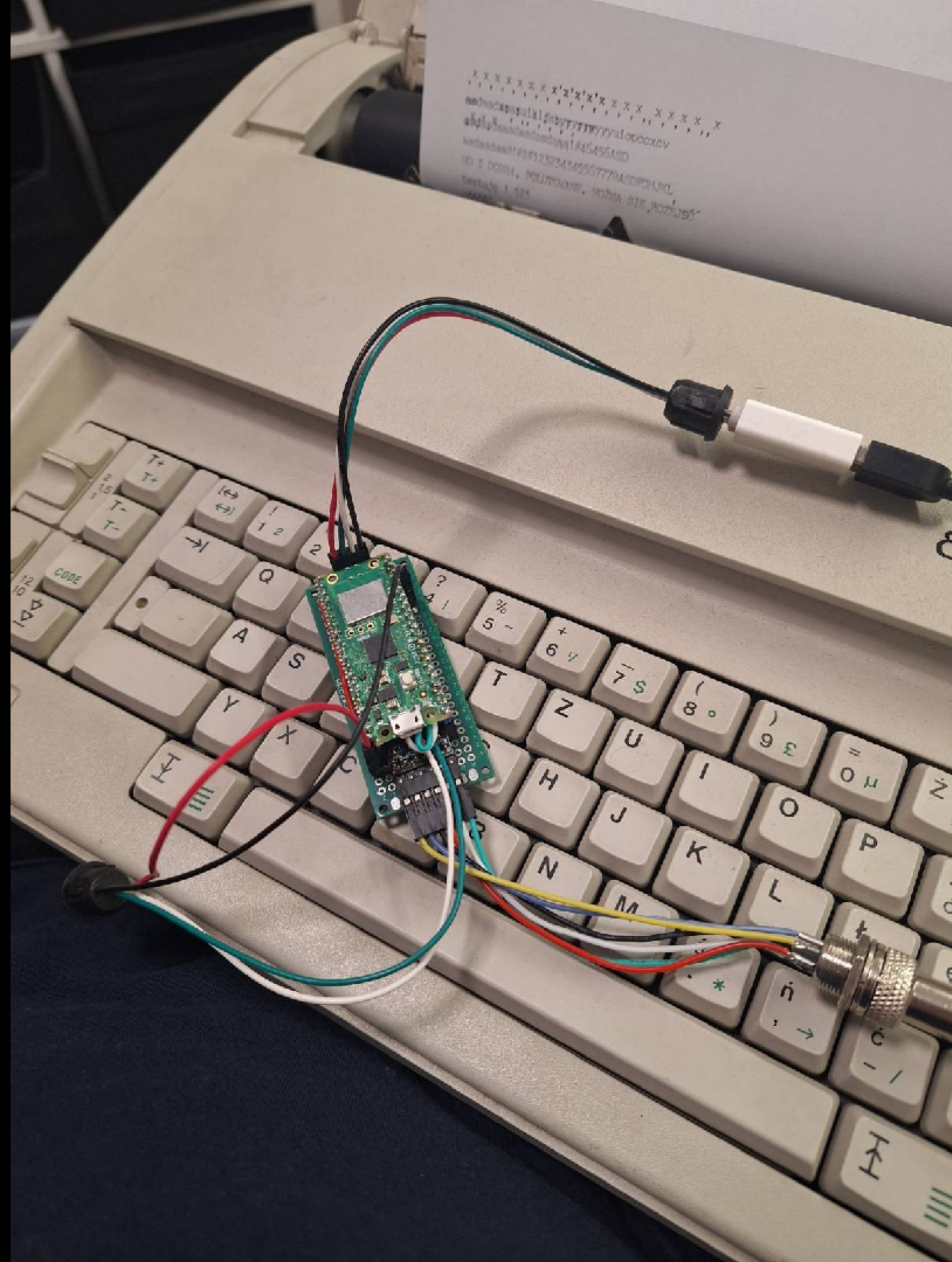
**Zamykamy to w ładną
paczuszkę**













Live Demo

Co dalej?

Przyszłość projektu

- W tym momencie projekt pewnie trafi na jakiś czas na półkę
- Optymalizacja drukowania, np. poprzez wykorzystanie druku wstecz
- Rozszerzenie funkcjonalności korekty, np. zapamiętanie całej strony
- Możliwość drukowania przez WiFi
- Możliwość zapamiętywania wpisanego tekstu na komputerze
- Rozważaliśmy fizyczne "ucięcie" wbudowanej klawiatury w celu zmniejszenia rozmiaru maszyny



 krzmaz



 mrozycki/ttpi



 mrozycki

Dziękujemy za uwagę

Pytania?