

# Arduino digitální multimetr

... aneb jak to dopadne, když softwarový architekt navrhuje hardware...

Petr Stehlík  
[www.pstehlik.cz](http://www.pstehlik.cz)

# Osnova

- Idea a motivace
- Návrh a plán
- Teorie a implementace
- Problémy a chyby

# USB CHARGER Doctor



# CHARGER Doctor

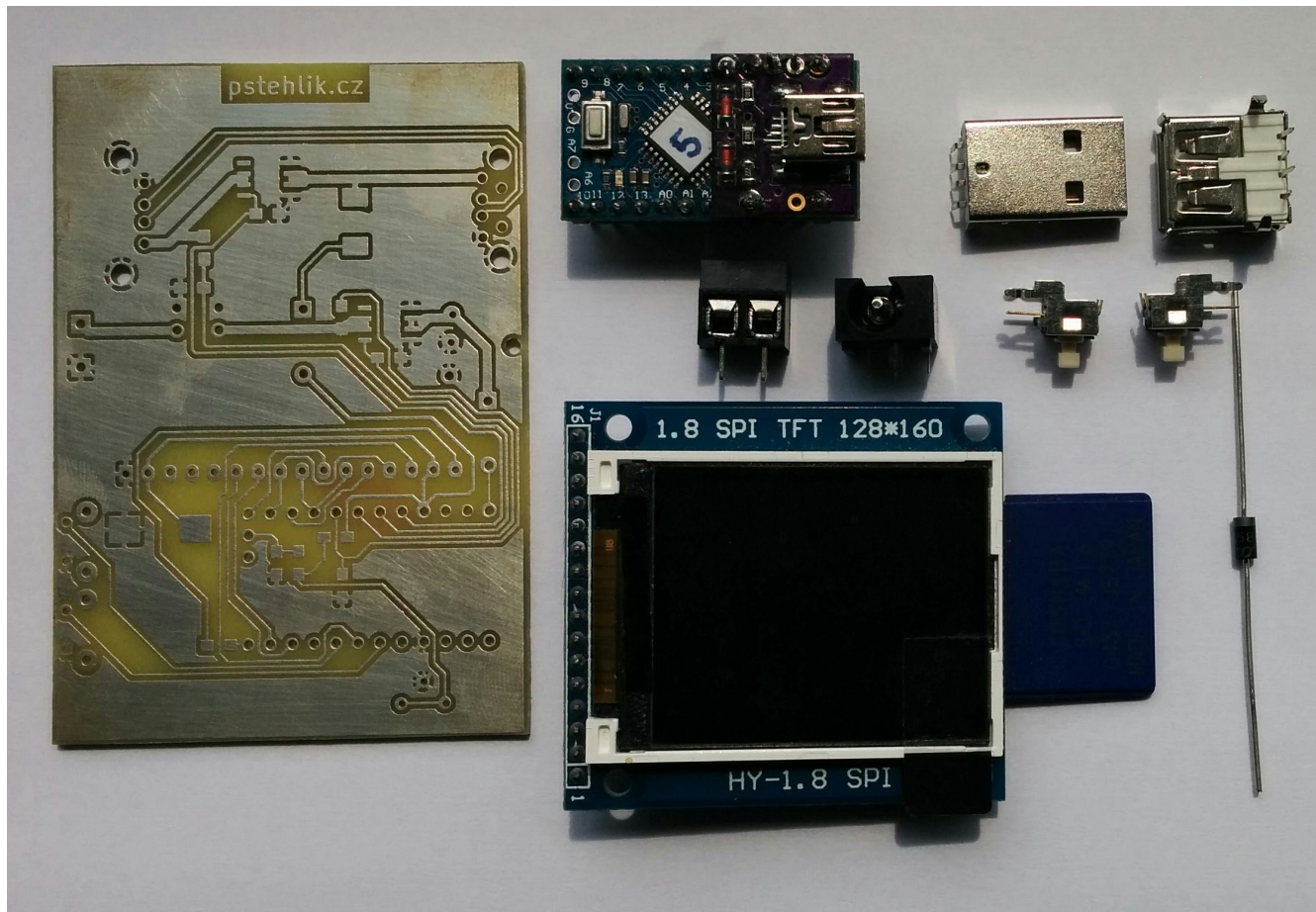
- Vlastnosti: měří napětí a proud na průchozím USB portu
- Výhody: rychle zjistíme, co se při USB napájení děje
- Nevýhody: nečitelné, nepřesné (jen 3 číslice), nešikovný konektor, a hlavně ten časovač přepínání mezi V a A!

<http://joysfera.blogspot.cz/2014/08/bezdratova-nabijecka-telefonu-pan.html>

# Můj „~~Doctor~~“ Arduino Multimetr

- musí měřit procházející proud 0 – 3A
- mohl by měřit i aktuální USB napětí
- nemusel by být omezený pouze na USB vstup
- mohl by měřit na dvou stupech naráz (dvoukanálový)
- mohl by zobrazovat měření v grafu (jako osciloskop)
- mohl by počítat „spotřebu“ v mAh
- super kdyby tak posílal naměřená data do PC
- měl by se napájet z měřeného obvodu, pokud možno

# Rozložený Arduino Multimetr



# Složení Arduino Multimetru

- Arduino Pro Mini
- 1,8“ TFT LCD
- MAX 471 = převodník proudu na napětí
- P-channel MOSFET pro přerušení vybíjení
- TL-431A = zdroj referenčního napětí 2,48 V
- ochranné diody, pojistka a pasivní bižuterie
- konektory a tlačítka

# Pro Mini (Arduino kompatibilní)

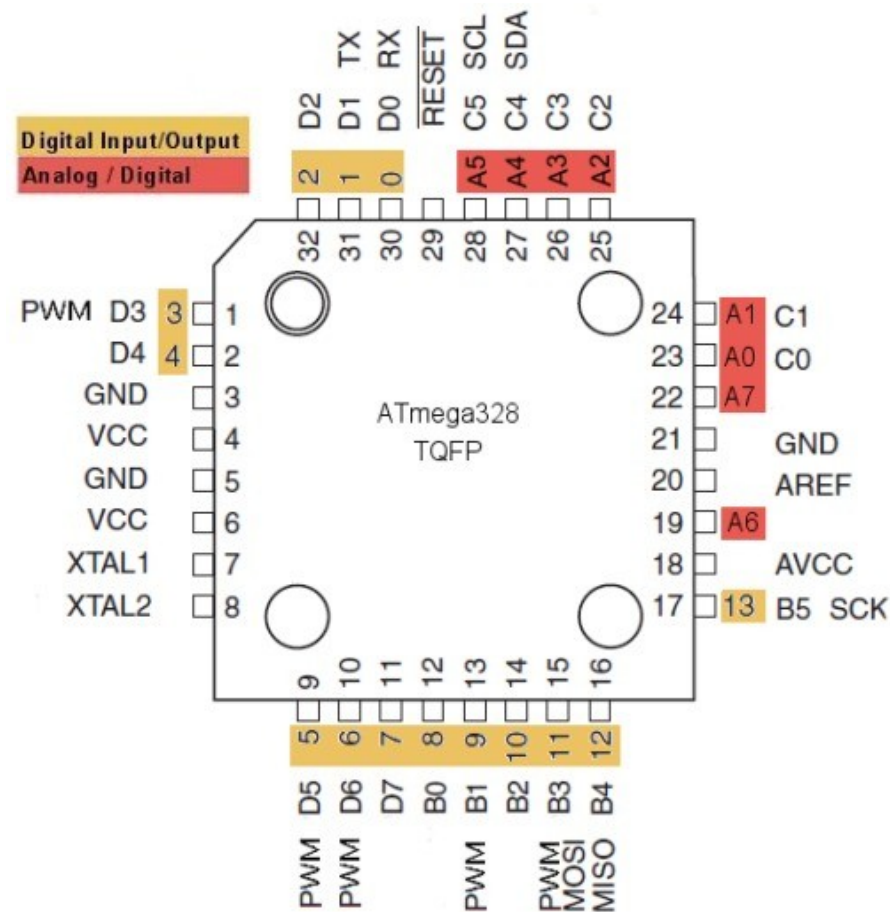




# Arduino ATMEGA 328p

- v klasickém Arduino UNO i oblíbeném Pro Mini
- 6 analogových pinů (Pro Mini navíc A6, A7)
- 10bitové rozlišení (hodnoty 0 – 1023)
- tři různá referenční napětí (VCC, interní 1,1 V, ext. AREF)
- jeden ADC a multiplexer k němu
- rychlost převodu (125 kHz, víc => méně přesné)
- převod jednorázově nebo napořád (free-running)
- případné přerušení na konci převodu (pro round-robin)

# srdce Pro Mini: ATMEGA328



# Příklad čtení z ADC

```
void adc_init(void)
{
    ADCSRA |= ((1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0)); // 16 MHz / 128 = 125 kHz
    ADMUX |= (1<<REFS0); // Voltage reference from Avcc (5v)
    ADCSRA |= (1<<ADEN); // Turn on ADC
    ADCSRA |= (1<<ADSC); // Initial conversion
}

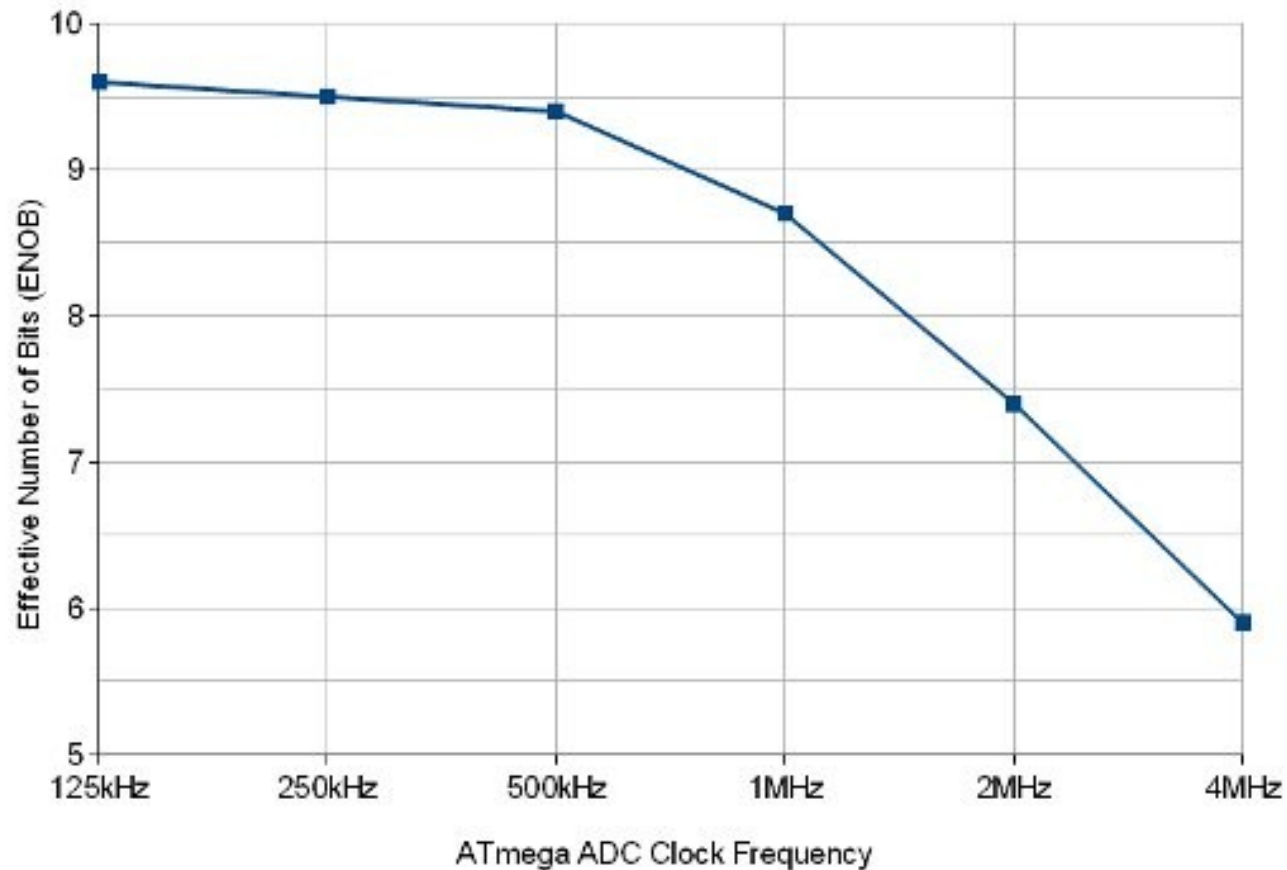
uint16_t read_adc(uint8_t channel)
{
    ADMUX &= 0xF0;
    ADMUX |= channel; // defines the new ADC channel to be read
    ADCSRA |= (1<<ADSC); // Starts a new conversion
    while(ADCSRA & (1<<ADSC)); // Waits until the conversion is done
    return ADCW; // Returns the ADC value of the chosen channel
}
```

Anebo prostě použijeme Arduino API: `analogRead(channel)`

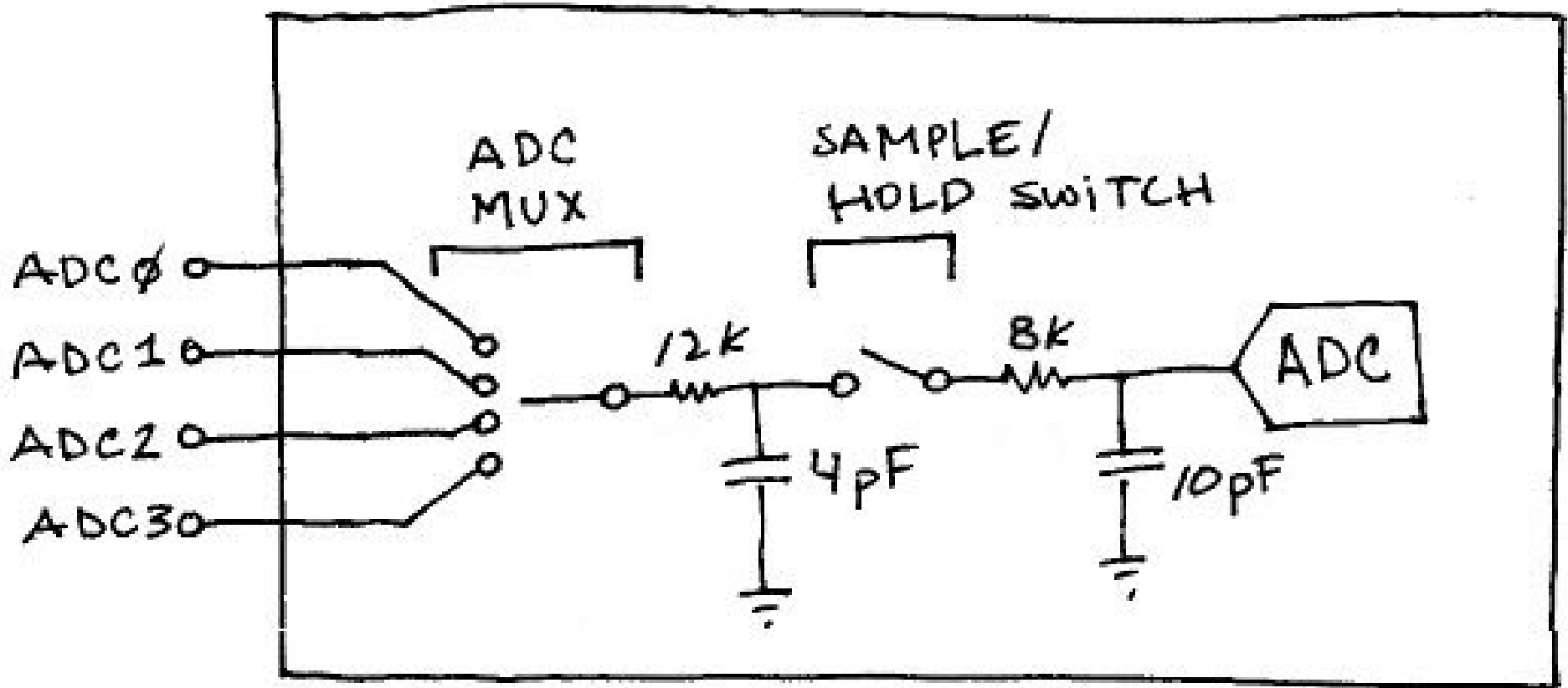
# Vlastnosti a chytáky ADC

- první převod trvá 25 cyklů, další 13 cyklů 125 kHz hodin (tj. 104 μs)
- přechod z kanálu na kanál ve free-running módu vyžaduje zahodit dvě měření (jinak čteme minulý kanál)
- změna kanálu v jednorázovém měření **prý** nevyžaduje žádné zahazování
- vypínejte digitální buffer u analogových vstupů (registr DIDR)
- čtěte nejdřív ADCL a pak ADCH, nebo prostě čtěte ADCW
- ADLAR registr zarovná hodnoty doleva (pro ≤ 8bit stačí číst ADCH)
- možnost uspat CPU – ADC Noise Reduction mode
- naměřenou hodnotu dělit 1023 nebo 1024?  
 **$\text{analogRead} = V_{in} \times 1024 / V_{ref} \rightarrow V_{in} = \text{analogRead}() / 1024 \times V_{ref}$**

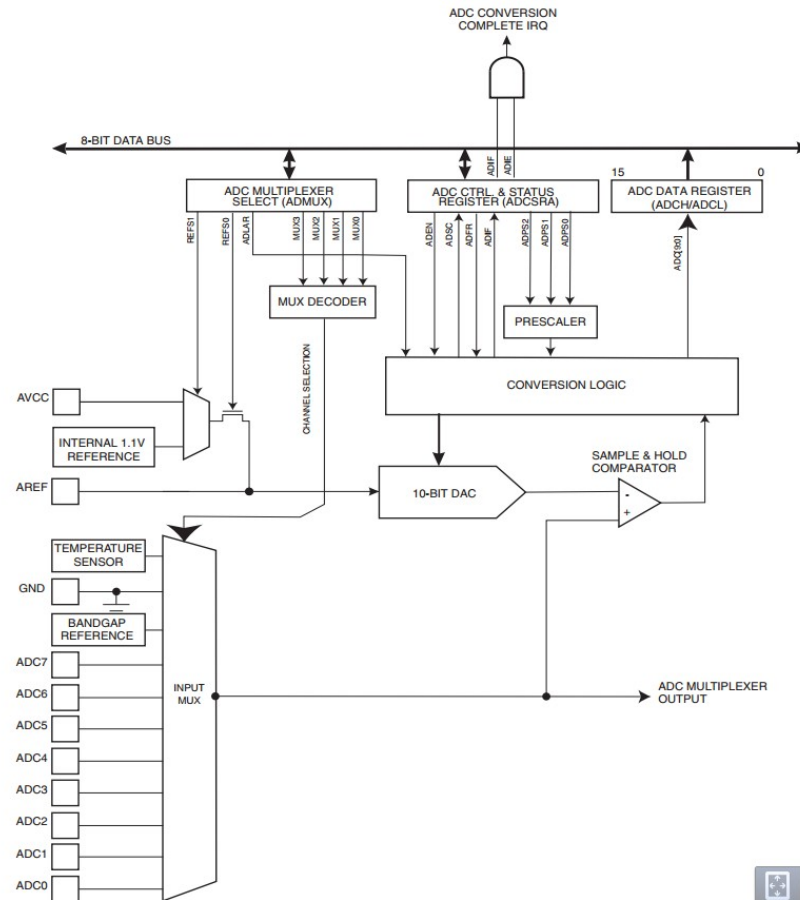
# Přesnost závislostí rychlosti převodu



# Ilustrace vnitřního zapojení ADC



# Skutečné vnitřní zapojení ADC



# Jak funguje ADC doopravdy

- předchozí ručně malovaný obrázek je z <http://www.openmusiclabs.com/learning/digital/atmega-adc/>
- kvůli kapacitě a odporu použijte zdroj s impedancí menší než 10 kOhm
- pěkný popis skutečné funkčnosti (successive approximation) je na <http://apcmag.com/arduino-analog-to-digital-converter-how-it-works.htm/>
- definitivní zdroj informací o ADC je kapitola 24 v datasheetu: [http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P\\_datasheet\\_Complete.pdf](http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf)
- „When the bandgap reference voltage is used as input to the ADC, it will take a certain time for the voltage to stabilize. If not stabilized, the first value read after the first conversion may be wrong.“
- „The first ADC conversion result after switching reference voltage source may be inaccurate, and the user is advised to discard this result.“



# Ještě jeden ADC trik

- Oversampling and Decimation

<http://www.atmel.com/images/doc8003.pdf>

- zvyšuje bitovou přesnost pomocí vícenásobného převedení vstupní hodnoty s trochou šumu
- pokud přirozený šum (tepelný, z CPU, ze zdroje atd.) v signálu není, můžeme dokonce přidat vlastní, uměle generovaný (z PWM výstupu)
- 4 vzorky => 11 bitů, 16 vzorků => 12 bitů,  $4^n \dots$

# TFT LCD 1,8"

- krásné rozlišení 128 x 160
- 18bitová barevná hloubka (262144 barev)
- připojen přes SPI rozhraní
- toleruje TTL úrovně na datových pinech
- nabízí slot pro SD kartu
- podporován knihovnou Adafruit ST7735
- levný a dostupný (Ebay: 88 Kč)

# TFT LCD na Ebay



**1.8" Serial SPI TFT LCD Module Display PCB Adapter Power IC SD Socket 128X160 S2**

Item condition: **New** ✉️ | [f](#) | [t](#) | [p](#) | [Add to watch list](#)

Quantity:  More than 10 available / 11 sold

Price: **US \$3.59**  
Approximately 88.38 CZK [Buy It Now](#)

[Add to cart](#)

6 watching [Add to watch list](#)  
[Add to collection](#)

<b>Free</b> Shipping	<b>Experienced</b> Seller	<b>New</b> Condition
-------------------------	------------------------------	-------------------------

Shipping: **FREE** Standard Int'l Shipping | [See details](#)  
International items may be subject to customs processing and additional charges. ⓘ  
Item location: **Shen Zhen, China**  
Ships to: **Worldwide** [See exclusions](#)

Delivery: **Estimated between Mon. Nov. 16 and Tue. Dec. 8**

**Top-rated seller**  
**service2worldwide** (53980) ⭐  
98,9% Positive feedback

- ✓ Consistently receives highest buyers' ratings
- ✓ Ships items quickly
- ✓ Has earned a track record of excellent service

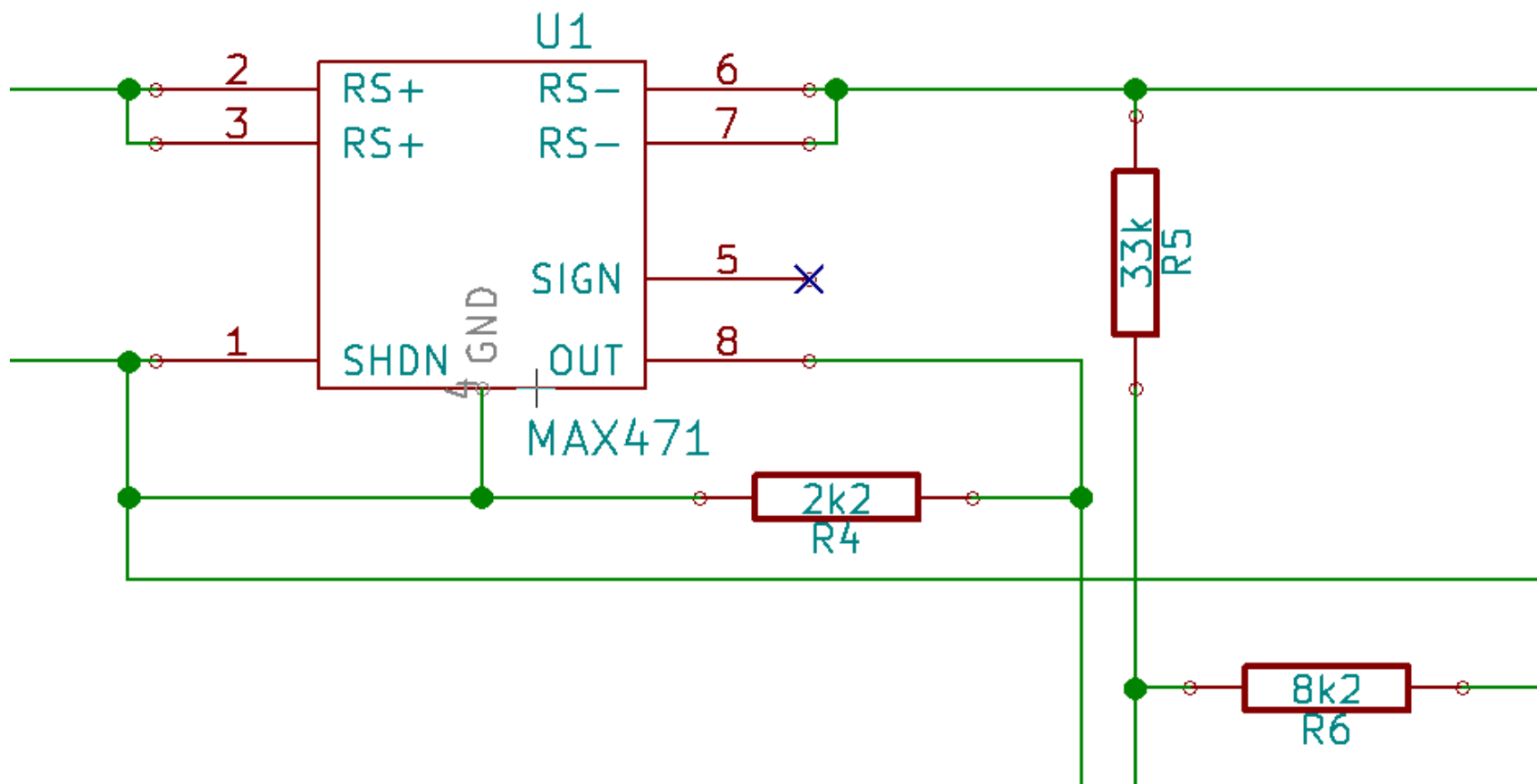
[Follow this seller](#)

Visit store: [service2worldwide](#)  
[See other items](#)

# MAX 471

- úkolem je převést proud na napětí
- ve skutečnosti převede proud 1 A na 0,5 mA :-)
- měřicí rezistor má jen 35mΩ
- funguje v napájecím rozsahu 3 - 36 V
- rozumně přesný (2%) a úsporný (< 100uA)
- výběhový typ – výrobce říká „nepoužívat“
- levný a dostupný (AliExpress: 10 ks za 150 Kč)
- podobné IO: MAX4072, LT1495, ZXCT1021 atd.

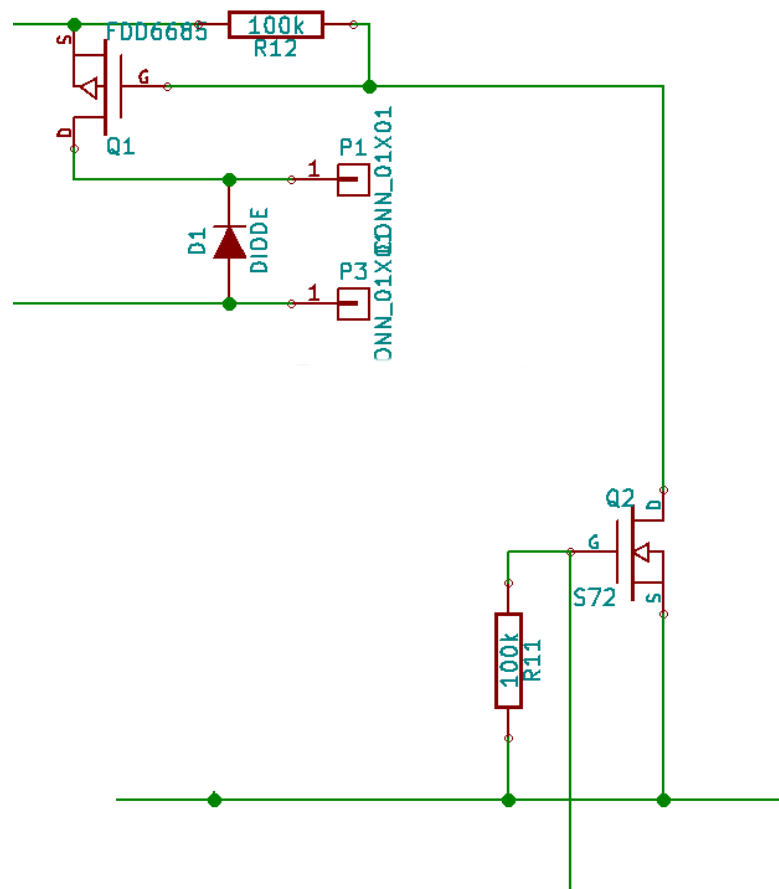
# Detail zapojení MAX 471



# P-channel MOSFET

- Fairchild Semiconductor FDD6685
- úkolem je volitelně vypnout výstup „jack“ kanálu
- má smysl při řízeném vybíjení článků
- $R_{DS(ON)}$  pouhých 30 m $\Omega$  (při 4,5 V mezi G a S)
- Gate Threshold Voltage typicky už 1,8 V

# Detail zapojení MOSFETu

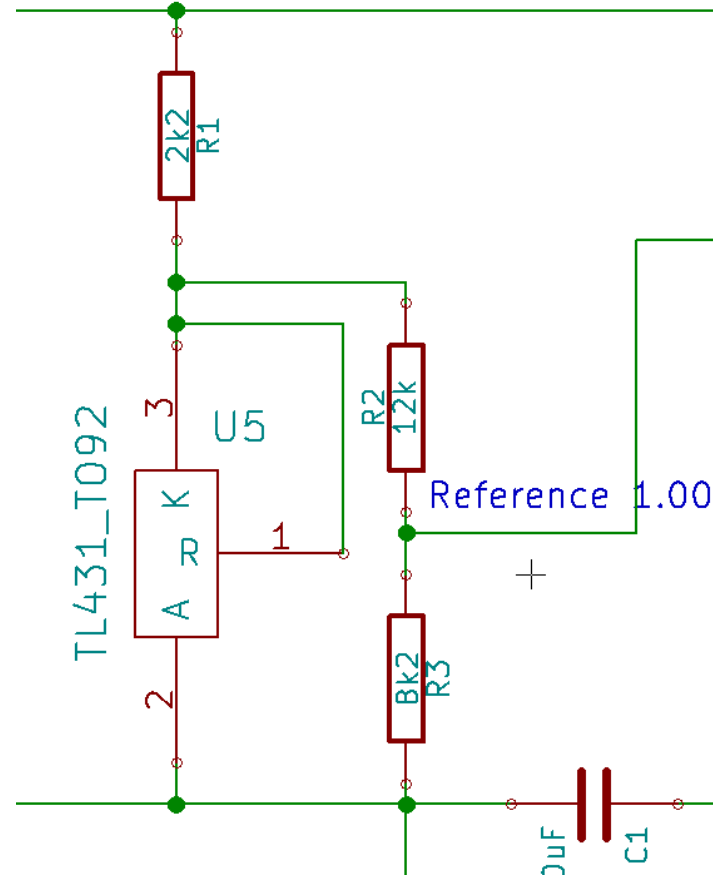


# TL431A

- úkolem je generovat stabilní referenční napětí
- recyklován z mrtvého ATX zdroje
- možné lepší náhrady – AD780 nebo MCP1525
- měl by být připojen na AREF, ale není...



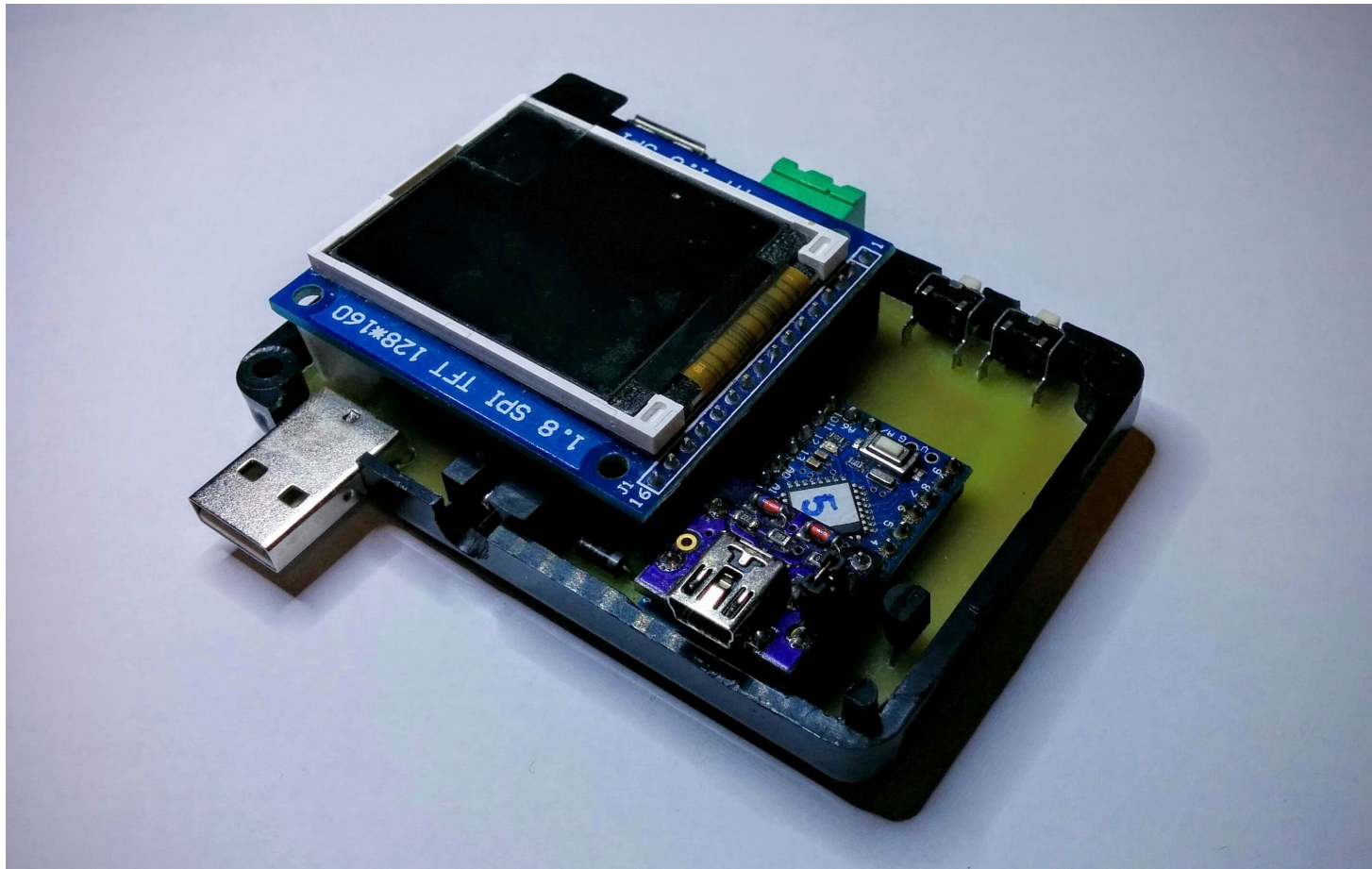
# Detail zapojení TL431



# Mechanická konstrukce

- když chybí 3D tiskárna, každý vynález musí začít výběrem krabičky - zde použita KM-22
- plošný spoj přesně na míru krabičky, displej jako sendvič ve slotu, výškově přesně vychází
- jak vyřezat prostupy pro displej, konektory a tlačítka?

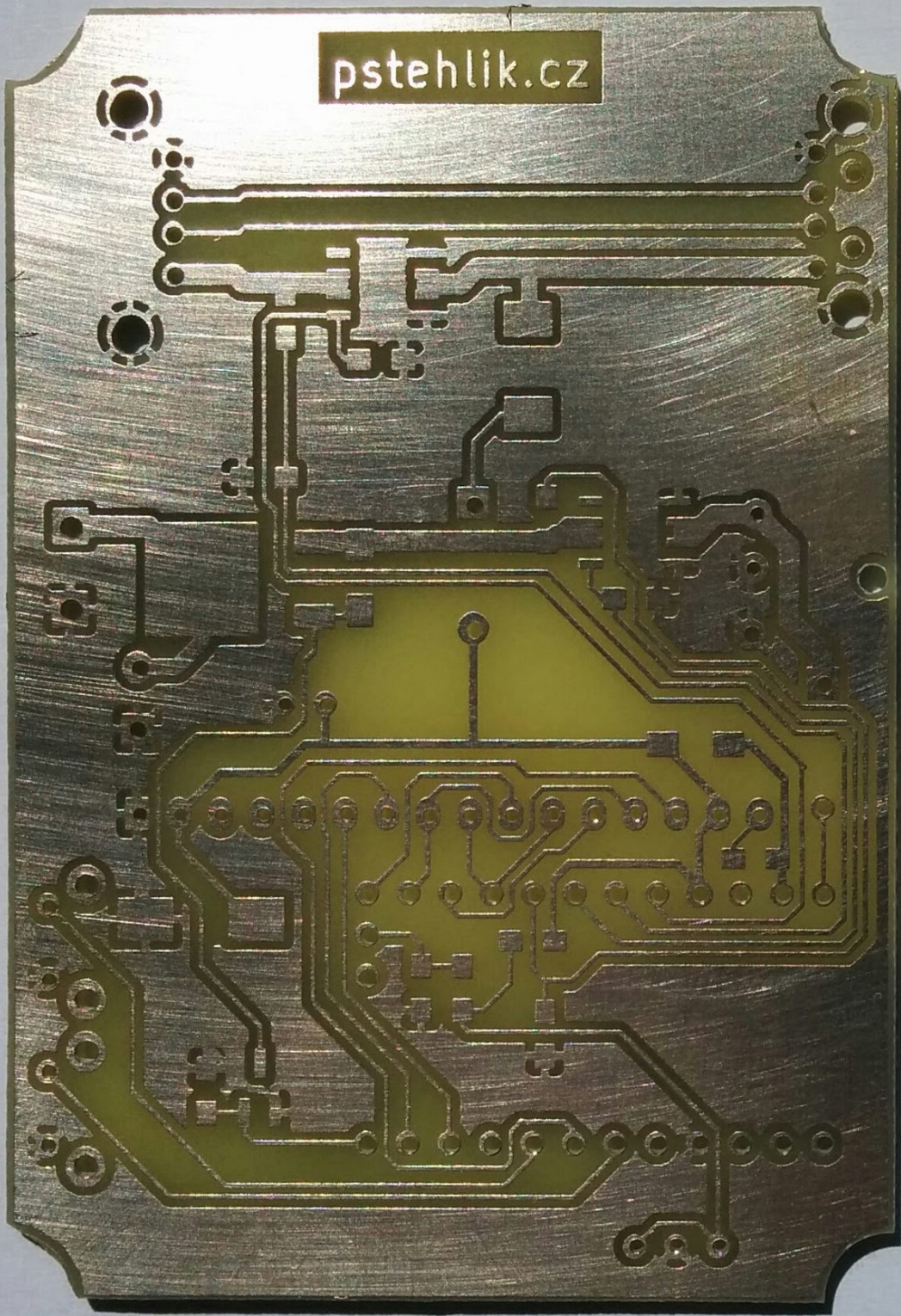
# Fotka otevřeného multimetru



# Plošný spoj

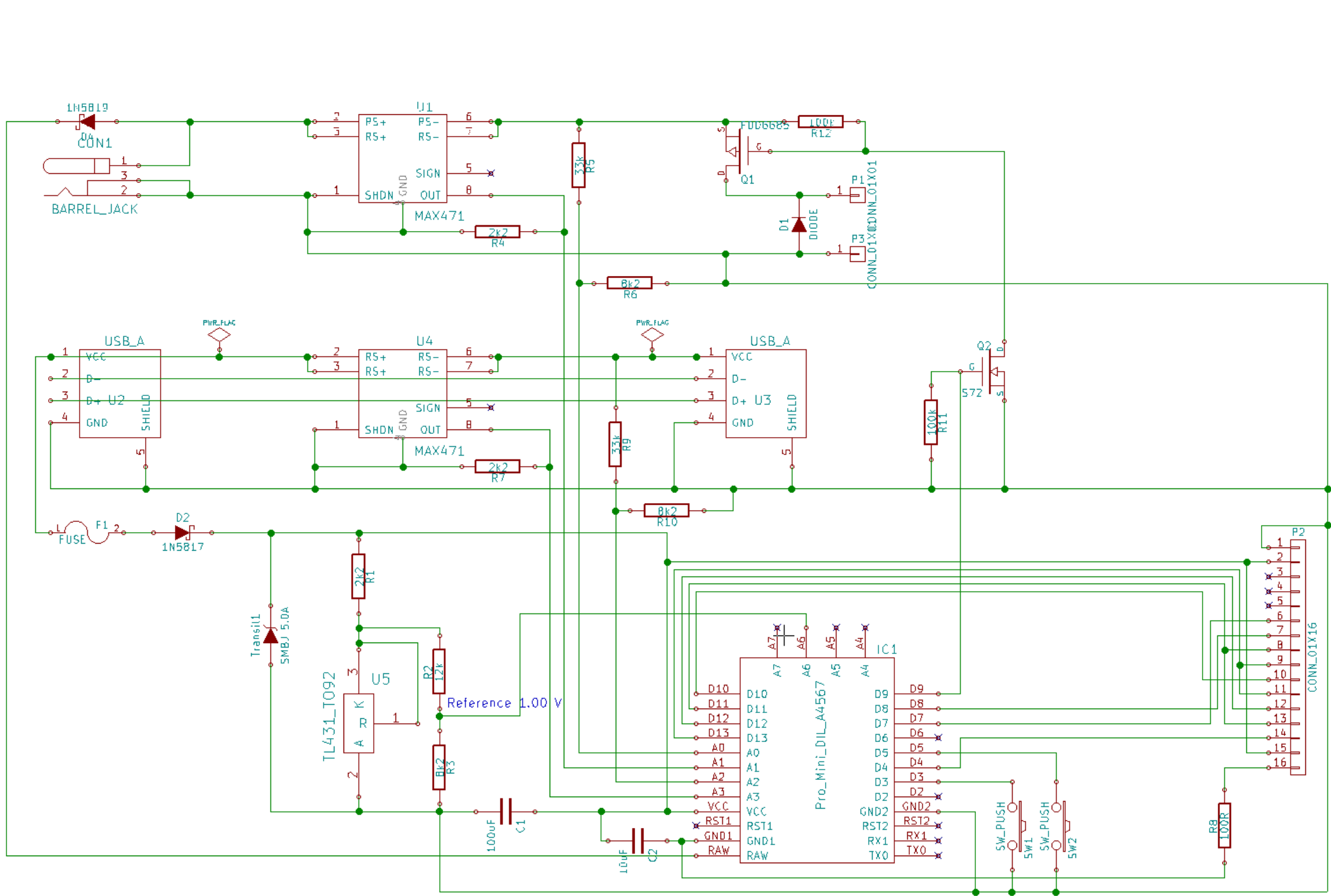
- v čem navrhovat – KiCad či gEDA (oba GPL)
- dimenzovat na zamýšlené proudy – 35/70  $\mu\text{m}$
- jednostranný či oboustranný? Bez prokovů?
- vyrobit doma či na zakázku (a kde)?  
<http://joysfera.blogspot.cz/2014/10/vyroba-plosnych-spoju-doma-i-venku.html>

pstehlik.cz



# Princip činnosti a použité triky

- dva automaticky přepínané rozsahy:  $< 1 \text{ A} / 5,5 \text{ V}$  a  $\geq 1 \text{ A} / 5,5 \text{ V}$
- rezistorové děliče napětí 1:5 (33k / 8k2) pro měření vstupních napětí až do 25 V
- externí TL-431 reference 1,00 V použita na přesné změření napětí interní reference (cca 1,1 V)
- interní reference použita při měření v nižším rozsahu
- VCC použito jako reference při měření ve vyšším rozsahu
- proud 1 A přeměněn na 1,1 V rezistorem 2k2 – díky tomu je 1 A právě plné rozlišení ADC (1023+1) → přesnost 1 mA
- Oversampling pro až 12bitovou přesnost (0,25 mA)



# Chyby v zapojení

- ochranná dioda u USB nemůže být typu Schottky kvůli velkému závěrnému proudu (ovlivňuje měření)
- MOSFET není plně saturovaný při napětí menším než 3,0 V a pak ovlivňuje měření svým proudem
- nepoužití AREF vstupu byla chyba (která se dá ještě napravit – A6 a AREF spolu v TQFP sousedí)



# Software Arduino Multimetru

- měří napětí na analog. vstupech A0-A3, A6 a interní ref. napětí
- hodnoty se snaží držet v celočíselných proměnných
- vypočítává napájecí napětí z poměru k interní referenci
- vypočítává změřené hodnoty podle napětí na A6 (1,00 V)
- vypisuje spočítané hodnoty na displej (a měl by kreslit graf)
- je připraven ukládat naměřené hodnoty do round-robin bufferu
- může zapisovat data na SD paměťovou kartu
- reaguje na dvě tlačítka (v přerušení)
- <https://github.com/joysfera/arduino-multimeter>

# Chyby a nedodělky v software

- ~~má dělit 1024 (tedy >> 10), dělí pořád 1023~~
- nevyužívá ADC Noise Reduction (neuspí CPU)
- nevypne MOSFET při příliš nízkém napětí
- nekreslí grafy
- nemá menu
- nekomunikuje s PC
- ... a ještě by se něco našlo :)

# Chyby v samotném návrhu

- nepoužít AREF znamená číst navíc A6 - zdržuje
- měřit oproti měnícímu se VCC je šílenost
- měřit proud a napětí na jednom ADC je hloupost nešikovné
- není kompatibilní s USB 3.0+ (napětí až 20 V)
- tajemné chyby ADC multiplexeru v ATMEGA

# Děkuji za pozornost

... a rád odpovím na vaše dotazy

[www.pstehlik.cz](http://www.pstehlik.cz)