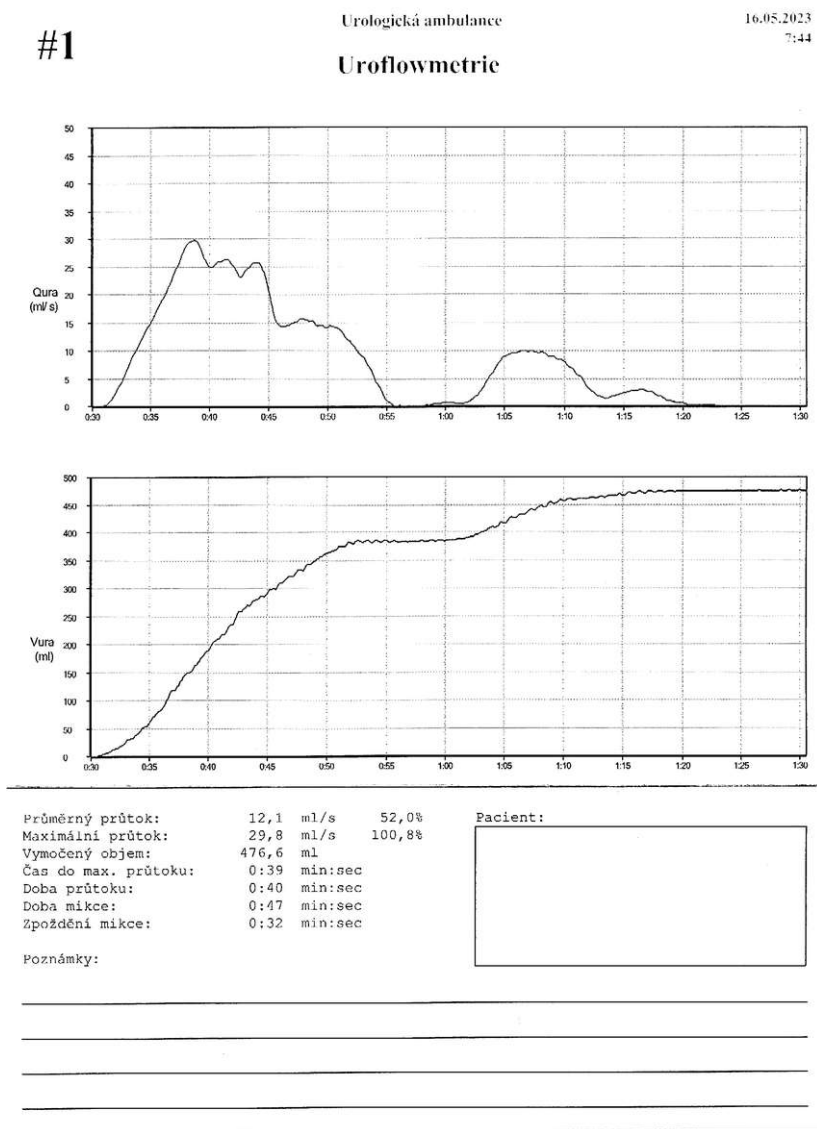


# Příběh urologické váhy a hořké zklamání z ní

PhDr. Mgr. Jeroným Klimeš, Ph.D. 2023-10-19

Jak jde psycholog životem, tak napřed řeší problémy batolící, školkové ap. O těch pravda toho moc nenapíše, ale pak se dostane na vysokou školu a tam začne postupně psát o partnerských problémech, napřed samozřejmě probere seznamování, pak partnerské dohadování, nakonec rozcházení. Průběžně přibudou problémy v zaměstnání, když tu zcela nečekaně přijde dlouho ohlašovaná krize středního věku, pak začne psát o "life span psychology" a to celé posléze završí úvahami o stárnutí a umírání. Tak jsem i já nakonec skončil u prostaty. (Kdybych byla ženou, skončila bych u inkontinence.)

Můj urolog mě proto seznámil s urologickou váhou, uroflowmetrií a uroflowgrafem. Zní to složitě (urea - moč, flow - tečení, metrie - měření), proto tomu soukromě říkám chcací graf. Díky jeho urologické váze ze mě tak vypadlo toto:



Uromic for Windows © 2017 by MEDKONSULT medical technology s.r.o. - www.mmtsistemas.com



Jak to funguje a obrázky najdete ve vyhledávači pod heslem "uroflowmeter". Návod na použití je jednoduchý. Zmáčknete tlačítko. Váha se probudí. Pokud jí to psychický stav dovolí, tak si pípne, a vy můžete začít močit do nádoby. Když domočíte, měření se samo ukončí po 30 sekundách nečinnosti (bez změny hmotnosti), popř. měření ukončuje manipulace s nádobou (prudká změna hmotnosti).

Další grafy na toto téma:

[https://www.researchgate.net/figure/Uroflow-curve-patterns-a-Bell-shaped-normal-b-Tower-shaped-suggestive-of-overactive\\_fig2\\_317316271](https://www.researchgate.net/figure/Uroflow-curve-patterns-a-Bell-shaped-normal-b-Tower-shaped-suggestive-of-overactive_fig2_317316271)

Pan doktor si postěžoval, že ta váha se prodává za 50.000 nekřesťanských korun českých a navíc ta firma je prodávala s tzv. kurvítky. Co to znamená. Stejná váha se prodávala též za 3000 Kč, ale byla na jedno použití. Pacient se jí měřil měsíc, doktor si stáhl jeho data a od té doby byla váha nepoužitelná, zablokovaná. Prostě

bylo v ní naprogramované tzv. kurvítko, aby se po jednom použití už nedala použít a šla do popelnice, do šrotu. Další pacient si musel koupit váhu novou, opět za 3000 Kč.

Podíval jsem se na tu váhu a graf a říkám mu, že to nemůže být nic až tak složitého. Na grafu vidím kumulativní křivku, tedy časosběrná křivka nárůstu hmotnosti a před ní distributivní graf průtoku, což je přibližně vždy rozdíl stávající a předcházející hodnoty. Tak jsem mu slíbil - nezávazně, že se na to podívám.

Technický popis mého bádání je popsán na konci souboru. Teď tento složitý proces můžeme v našem příběhu přeskočit a povíme si, co se stalo poté, co už jsem tu váhu měl hotovou a zprovozněnou.

## **Hořké zklamání z urologické váhy, aneb Jak mizí peníze v našem zdravotnickém systému**

Ukazoval jsem tuto váhu ještě jiným lékařům, včetně dalších urologů, a tak jsem se dočkal studené sprchy. Vcelku jedna z prvních otázek byla: "A má to certifikaci?" "Samozřejmě, že nemá, ale zkalibrovat si to můžete sami." A tak se mi dostalo poučení, jak to v našem zdravotnickém systému fičí.

Výrobní cena takové váhy je, jak vidíte, i při kusové výrobě cca 2000 Kč a dokáže ji postavit běžný psycholog. Stejná váha, která také nemá kurvítko, ale která nějakou obskurní cestou dostane zdravotní certifikaci, se pak může prodávat za 60.000 Kč.

### **To ale není vše...**

Kdo ji chce prodat, musí navíc nějakými penězi či úplatkem přemluvit vedení nemocnice, aby si tu váhu koupili právě od něho. Takže obyčejná časosběrná váha pak vyjde české občany na 100.000 Kč.

### **To ale stále není vše...**

Pokud lékař chce vykázat pojišťovně uroflowmetrii, pak musí uvést číslo certifikované váhy. Jinak uroflowmetrii nebude mít od pojišťovny proplacenou.

### **To je důvod, proč si musí pacient sám koupit koupit váhu s kurvítkem na jedno použití - aby doktor dostal zaplacenou (jednorázová váha totiž číslo pro pojišťovnu má).**

Doporučuji tedy, abyste se s doktorem domluvili, že mu raději poplatek od pojišťovny zaplatíte sami. Určitě vás to vyjde levněji, než si kupit váhu na jedno použití.

### **To ale stále není vše...**

Váha se musí opakovaně certifikovat a kalibrovat, a tak vzniká další kontinuální penězotok, který vytahuje peníze z kapes nemocných lidí. Proto nejsou sestry, proto je péče na LDN tak šílená, jaká je, proto v psychiatrických léčebnách na 40 opravdových bláznů připadá 4 osoby sloužícího personálu. Na vlastní oči jsem viděl tu hrůzu, jak vypadá pavlón pro mentální retardace a jak tam žijí Otíci z Vesničky mé střediskové.

Já jsem tuto váhu vyráběl s představou, že lékařům usnadním živnost. Odpovědí mi bylo - **úkollem této váhy není ušetřit, ale naopak více utratit. Cílem není, aby co nejvíc doktorů mohlo levně využívat uroflowmetrii, ale cílem je vytáhnout ze zdravotnického systému co nejvíce peněz.** Žijeme v 21. století v ČR, kde egoismus a sebestřednost je nejvyšší hodnotou a kde lidi nestárnou. Pokud nějaký Pražák dostane Alzheimeru, tak je coby odhalená náplava odlifován 200 km za Prahu, aby ho tam nikdo nemohl navštívit. Je totiž vědecky dokázáno, že [http://klimes.mysteria.cz/clanky/komentare/vyznamne\\_osobnosti\\_ujezd.htm#seniori](http://klimes.mysteria.cz/clanky/komentare/vyznamne_osobnosti_ujezd.htm#seniori)

skutečný Pražák dostat Alzheimeru prostě nemůže...</a>

Poslanecká sněmovna je plná obstarožních páprďů, kteří za nedlouhý čas po ukončení své poslanecké štače půjdou rovnou na zmíněné LDNky a už teď mají problémy s prostatou. Nicméně tito nic nedělají proto, aby si usnadnili a připravili své vlastní umírání, které je čeká max do 25 let.

Moje generace, co žila z ideálů Sametové revoluce, má dnes posledních pár let na to, aby si připravila svou poslední štaci. Zde vidíte, jak na to jde od lesa. Mají představu, že když budou mít hodně peněz a konexe, tak jim to garantuje kvalitní péči. Tak to je mohu ubezpečit, že negarantuje. Dnes jsem byl v nemocnici za známým, kde na vedlejší posteli chroptěl devadesátiletý stařec - ročník 1934, prý inženýr. Tak to mohu mým vrstevníkům garantovat, že v roce 2057 až budeme 90letí na LDN, tak nebudeme mít ani konexe, ani peníze. A pokud to půjde dobře, tak na cedulce budeme mít napsáno:

*"Dnes vám zachroptěl PhDr. Mgr. Jeroným Klimeš, Ph.D., pop-psycholog, co vyrobil urologickou váhu."*

Povím vám, že toto prožžení byla opravdu studená sprcha.

**Každopádně pro všechny lékaře i pacienty. Pokud chcete využívat data uroflowmetrie, byť vám ji nikdo neproplatí, tak tady máte návod, jak si takovou jednoduchou váhu vyrobit na koleni za pár korun.**

## Jen co jsem se tak porůznu dozvěděl od různých lidí (ústní informace)

	Necertifikovaná varianta/ výrobní cena	Certifikovaná varianta
Uroflowváha na jedno použití	2 000 Kč	4 000 Kč
Úplně stejná uroflowváha bez kurvítka	2 000 Kč	60 000 Kč
Defibrilátor	tipuji podobnou částku	100 000 Kč
Úplně stejný defibrilátor jen používaný v helikoptěře	tipuji podobnou částku	300 000 Kč
Hliníkový žebřík „pro hasiče“	10 000 Kč	30 000 Kč

Příklad od zubaře. V ordinaci mají sterilizátor, je opakovaně kontrolován a certifikován co dva roky. Paraziti si vybírají asi 2000 Kč za tuto letmou zkoušku. Ale teď přišel příkaz, že musejí funkční sterilizátor vyhodit a koupit nový, protože je už 20 let starý. To má logiku, jako kdyby EU dalo nařízení, že každý občan musí vyhodit přístroje, pokud jsou starší než 20 let.

Rozdíl těch uvedených cen představují platy darmožroutů, parazitů. To jsou lidé, co nic nevytvářejí, jen parazitují na zdravotním pojištění a lékařích, zubařích, nebo na hasičích, autoopravcích ap. Stát nechává vysávat certifikáty všechny okolo, ale když potřebuje peníze, tak raději zruší důchody. Divíte se, proč to takto je? Důvod je prostý.

### Proč žádná vláda s certifikáty nic neudělá?

Česká republika se skládá z různých mafií. Nejúspěšnější mafiáni byli vždy okolo ODS, ale pozor nebyla to nikdy jedna homogenní skupina - dopravci, stavaři, zemědělci (ti jsou dnes v ANO), druhá liga byla v ČSSD (kolotočáři ap.), ale tím výčet nekončí máme tu čínskou, ruskou, vietnamskou, drogovou mafii, lichváře, co žijí lidí na sociálních dávkách (mimo jiné i tzv. „spotřebitelské úvěry) atd. Všechny tyto mafie, když je blahobyť, tak mezi sebou dodržují jakýsi pakt o neútočení. Vzácně prosáknou do médií nekomentované zprávy, že ten který kontrolní úřad podusil nějakou velkou firmu - to je způsob, jak se přenastavují pravidla. Například předpokládám, že takto kradmo se vytratil či marginalizoval Homecredit. Vietnamští stánkaři zmizeli za dekádu z ulic, aniž o tom byla jakákoli veřejná debata. EU zlikvidovala uloz.to ap.

#### Dva reálné příklady

Jeden čínský student, velmi úspěšným studentem gymnázia, jednoho dne zmizel a už nikdy do školy nepřišel. Učitelka se po něm sháněla. Když byla na policii, tak se jí zeptali: „A to byl Číňan?“ Když řekla, že ano, tak od toho okamžitě dali ruce pryč. Tak dál pátrala na sociálních sítích, když jednou tak šla po ulici, tak k ní někdo přistoupil a byl to ten student. Řekl jí: „Paní učitelko, nemějte o mě starost. Jsem v pořádku.“ Co ta věta znamenala a zda byla pravdivá si můžeme jen domýšlet... Takhle v rámci jedné třídy či školy jsou do sebe prorostlé mafie ze všech možných koutů světa.

Druhý příklad mi vyprávěl klient. Jeden poslanec, který se angažoval proti kouření či v nějaké té ropné kauze, jel po dálnici a najednou se ocitl mezi třemi kamióny - před sebou, za sebou a jeden vlevo. Společně ho přitlačili ke svodidlům. Z jednoho vystoupil chlapík a prý mu povídal: „Pane poslanče, mějte rozum. Máte rodinu. Představte si, co by se stalo, kdybych to příště neubrzdl...“ A osamělý pan poslanec měl rozum a pochopil, že má rodinu. Zcela na rovinu já bych taky pochopil.

Česká republika totiž není svrchovaný stát, to je jen hřiště, kde se odehrávají techtle mechtle nejruznějších mafií. A pokud zrovna nejste součástí české majority, tak se vás ČR vůbec nezastane.

V psychologii to popisuje teorie jestřábi a holubice. Jestřábi odrbávají holubice (obyčejné občany), ale když se porvou mezi sebou, tak to pro ně bývá dost fatální - viz kauza Rath. Proto mafiánské skupiny v Čechách se nerady vzájemně konfrontují, napřed se vždy vyzkouší, jestli by nešlo ještě víc odrbávat obyčejné občany. Tedy ani naší vládě se nechce jít do války s certifikačními agenturami, protože by to kromě jiného znamenalo minimálně složitá jednání, ne-li válku s Evropskou Uhnijí.

Vláda Fialy je neschopná a slabá, proto raději zruší důchody, než aby zredukovala tyto nesmyslné poplatky za certifikace a musela jít do sporu s jinými mafiemi této shnilé republiky. Viz příběh poslance výše.

Když jsem byl zastupitelem v Líbeznici, tak jsem například pochopil, jak se systematicky ožebračuje venkov. Všem obcím se seberou prakticky všechny peníze, co se vyberou z daní, a ty se pak vrací od Prahy do periférií devótním starostům. Kdo má svou hlavu jako Kozina, tak pro svou obec nesežene ani korunu.

<a

href=http://klimes.mysteria.cz/clanky/komentare/jeronym\_klimes\_manipulace\_politiku\_s\_prispevkem\_obcim.

pdf target=\_blank>Jak ministerstvo a poslanci zdárně dělají z lidí blbce</a> Ale dokud bude lidi více bavit Stardance než princip fungování certifikátů či zákon o státním rozpočtu, tak v Čechách lépe nebude. A lidi bude více bavit Stardance, dokud nebudou trpět hladem. Proto politologické teorie říkají, že demokracie nefunguje za extrémní bídy a za blahobytu. Podmínkou nutnou vzniku a udržení demokracie je mimo jiné i mírná bída - čili svět konce Rakouska Uherska. Lidé musejí cítit, že se jim boj o demokracii projevuje na jejich životním standardu. Pokud mají pocit, že politika je o ničem, tak se na ni vykašlou a pustí si Stardance. Proto dnes se hroučí demokracie po celém světě, nejen u nás, neb blahobyt (vymizení hladu) začíná být globální normou.

Hodně hluboký vhled do fungování našeho státu mi dal i tento článek o korupci <a href=http://www.klimes.us/mojeprace/korupce/korupce2004.pdf target=\_blank>Korupce</a>. Vyšel v jednom <a href=http://knihy.klimes.us target=\_blank>sborníku</a>. Ten jsem si pozorně přečetl a byl jsem naprosto konsternován, že se v něm například otevřeně psalo, že podle typu ekonomických dat je možné najisto konstatovat, že Česká republika patří mezi státy řízené tzv. centrální korupcí, tedy korupcí jdoucí od vlády. To je vcelku to, co jsem popsal výše.

Vidíte, že jak jdu životem, tak vidím takových politických špinavostí stále více a více, a jsem z toho TAK otráven, až to není hezké. I když s tím my občané na okraji nemůžeme nic dělat, měli bychom to znát (proto toto zde píš). **Prostě až bude nějaký politický panák zase skuhrat, že ve státní pokladně či ve zdravotnictví nejsou peníze, tak je dobré se ho zeptat: „A proč nezakrouhnete mafie okolo certifikátů?“ Jen na jedné urologické váze ušetříte jářku 100.000 Kč. To je plat zdravotní sestry skoro na tři měsíce.**

Tím končí můj příběh urologické váhy a následuje technický popis.

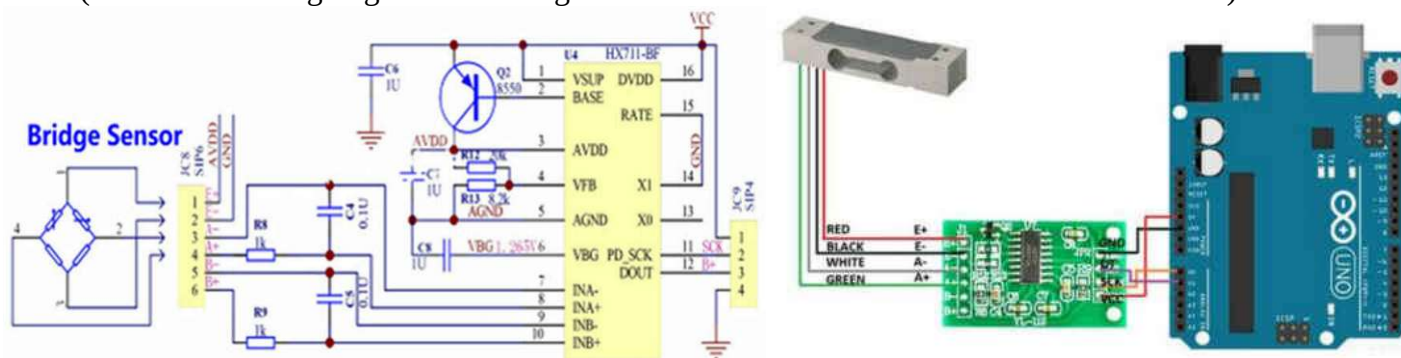
## Technický popis, jak si vyrobit urologickou váhu za 2000 Kč

Na eBay jsem si koupil:

A) **Raspberry PI** (do 1000 Kč zaleží na typu) a nainstaloval jsem do něj headless system, tzn. nejsou v něm XWindows, ale jen naskočí terminál bez přihlašování. Samo se to připojuje na mou wifinu, takže to mohu editovat přes SSH či FTP. To je možná zbytečný luxus, stačilo by Arduino se zápisem na flashdisk, ale v Raspberry se dá programovat v Pythonu a všechny konverze jdou udělat rovnou v něm.

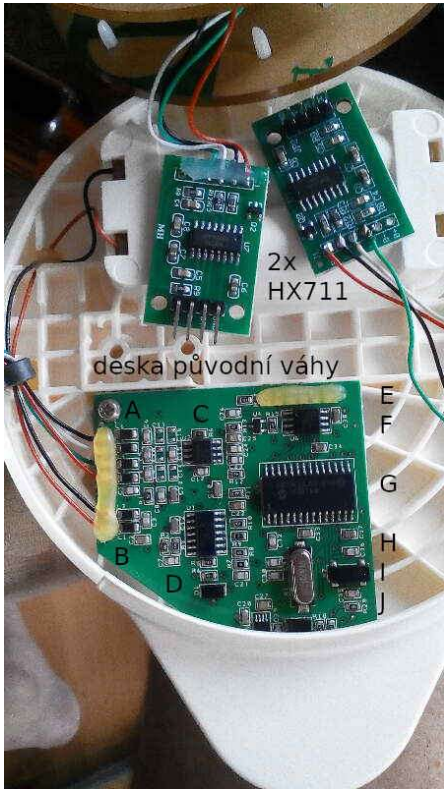
B) Chip **HX711 (30 Kč)**

(DIY **HX711** Weighing Pressure Weight Sensor 24 bit AD Converter Module for Arduino)



HX711 je ta zelená destička. Vlevo je její schéma. Vpravo je vázící senzor, čidlo - součást váhy, HX711 a Arduino, ale já jsem použil Raspberry. Arduino počká, zejména proto, že u toho starého mám asi odbouchlé vstupní piny. Asi jen ty analogové, takže digitální snad ještě fungují. Musím to vyzkoušet, nějak se mi do toho nechce.

C) Z původní váhy jsem využil vázící čidlo, viz obrázek výše. Kromě toho jsem zkoušel ještě jiné senzory, co jsem vykuchal z vyhozených vah. Fungovaly všechny, ale měly různou kalibraci. K tomu se ještě dostaneme.



Tady jsem si při vyrábění vyfotil funkční zapojení senzoru na HX711. Horní dva jsou dvě váhy - jedna z ebay, druhá ze tříděného odpadu. Obě fungovaly s tímto zapojením. Někdy ale jsou návody s jiným zapojením.

Pod těmito čipy je funkční zapojení původní váhy. Není to celá deska, zbytek je skryt pod tlačítkem. Napíšu, jak tomu rozumím, ale neručím za to (ani jsem to moc nestudoval):

A) Připojení senzoru váhy k čipu HX711 (D)

B) Napájení 2 tužkové baterie.

C) Nevím

E a F) Komunikační rozhraní k čipu G

H) Krystal na taktování G a kolem něho jsou obvody vyrábějící systémový čas a datum

G) Je vlastní "počítač", něco jako Arduino

J a níž pod tlačítkem) je mikrospínač a USB rozhraní.

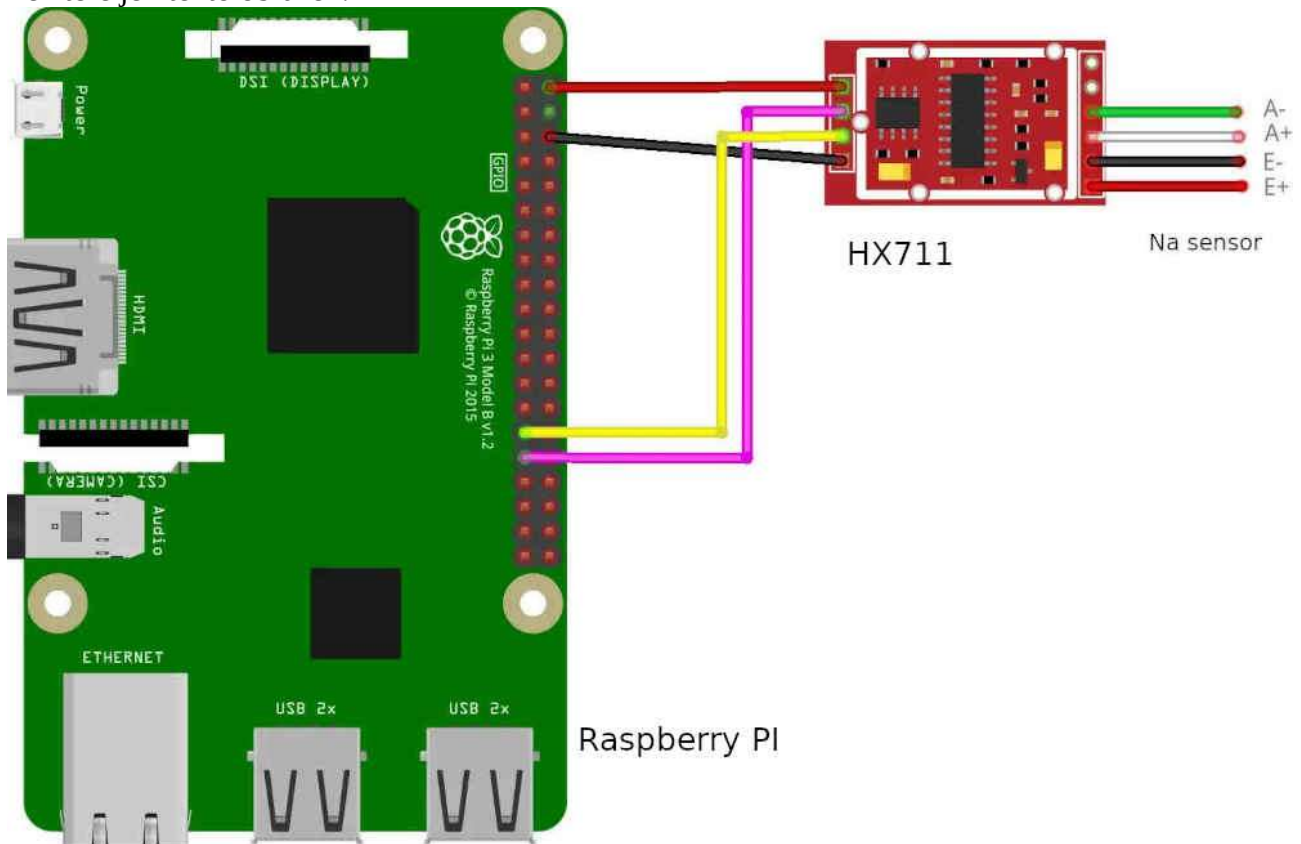
Váha přes AD převodník HX711 změří váhu, ukládá si čas a naměřenou hodnotu. Zpracování dělá až program v PC. Celé to řídí mikročip G. Programátoři do toho lezou přes E. Uživatel přes USB (J)

## Jak to prodrátovat?

Z desítek stránek byla nejlepší:

<https://tutorials-raspberrypi.com/digital-raspberry-pi-scale-weight-sensor-hx711/>

ze které je i tento obrázek:



Vypadá to nakonec takto:



Připojil jsem to na monitor, klávesnici a nechal naběhnout do terminálu.

Pak jsem nainstaloval hx711py-master.zip z uvedené stránky, přesněji řečeno upravil jsem soubor example.py. Každopádně až Vám to poběží, tak je dobré si projít vlastní knihovnu hx711.py, která vlastně čte hodnoty z čidla. Tento soubor by měl být ve stejném adresáři jako Váš program.

### Kalibrace



Protože váha v zásadě fungovala, musel jsem ji nakalibrovat. Ze skříně jsem sundal laboratorní váhu se závažími a postupně jsem je tam přidával s proměnnou v souboru exaple.py nastavenou na 1:

referenceUnit = 1 # asi na 8. řádce

Tupým odhadem jsem nastavil modus/median :

**číslo, co se objevilo závaží v gramech jednotek na gram**

19350	50	387
11590	30	386

7720                      20                      386  
 3830                      10                      383

referenceUnit = 386 # asi na 8. řádce, nova hodnota - modus měření

Soubor example.py jsem předělal, aby ukládal hodnoty do souboru s časovým razítkem, ukončoval práci, když nikdo v okolí 30s nemočí nebo pokud mu někdo sebere jeho nádobu a další desítky detailů.

### Kalibrace originálního čidla

Pak jsem vyměnil vážící čidla - použil jsem ten z originální váhy. Tam jsem udělal novou kalibraci, ale aby to nebyl tupý odhad, tak jsem raději použil metodu total stupido, totiž metodu nejmenších čtverců:

1	jmeno_souboru_vzorku: vzorek_14.dat								
2								koeficient	suma čtvercu
3								1020	586796292
4	číslo, které generuje váha	rozdíl dvou po sobě jdoucích hmotností	časové razítko (sekundy a nanosekundy)	rozdíl dvou po sobě jdoucích časů	přírůstek děleno prodleva	závaží na váze v gramech	koeficient krát hmotnost	u stabilizovaných vzorků čtverec rozdílu naměřené a odhadnuté hmotnosti.	
5	B9	=B9-B8	D9	=D9-D8	=C9/E9	G9	=G9*\$H\$3	=IF(ABS(C9)<300;(H9-B9)^2;"")	
6	Merení	Hmotnost	Prirustek	Cas	Prodleva	Prutok	zavaží	odhad	rozdil
69	63	91708,44444	-7	1691351578	0,768648863	-9,10688916	90	91800	8382
70	64	91709,44444	1	1691351578	0,768468142	1,30129012	90	91800	8200
71	65	91666,44444	-43	1691351579	0,768513918	-55,9521422	90	91800	17837
72	66	91146,44444	-520	1691351580	0,768330097	-676,792438	90	91800	
73	67	91658,44444	512	1691351581	0,77285099	662,4821685	90	91800	
74	68	91616,44444	-42	1691351582	0,786863804	-53,3764545	90	91800	33693
75	69	91571,44444	-45	1691351582	0,768903017	-58,5249362	90	91800	52238
76	70	93628,44444	2057	1691351583	0,768851042	2675,42071	90	91800	
77	71	102380,4444	8752	1691351584	0,78687501	11122,47802	100	102000	
78	72	102226,4444	-154	1691351585	0,773288965	-199,149357	100	102000	51277
79	73	102146,4444	-80	1691351585	0,787176132	-101,629098	100	102000	21446
80	74	102054,4444	-92	1691351586	0,768699884	-119,682599	100	102000	2964
81	75	102018,4444	-36	1691351587	0,768681049	-46,8334689	100	102000	340

Koeficient, jak vidíte vyšel 1020, tzn. váha měří přibližně v miligramech. Tento koeficient nastavujete tak, aby sousední suma čtverců (suma sloupce "rozdíl") byla co nejmenší (proto se to jmenuje metoda nejmenších čtverců). Dá se to dělat Solverem (menu Calcu/Excelu), ale já to dělal ručně. Nepřesnost vůči laboratorním závažím byla cca 2g/100g. Je to třeba dělat měřit dvakrát - jednou při přikládání závaží, podruhé při ubírání.



Z původní váhy zbyl stojánek s čidlem, ze kterého možná půjde 6-8 žilný kabel. Ten povede čidlo (4 dráty), tlačítko (1-2 dráty), bzučák (1-2 dráty, pokud budou mít společnou zem s čidlem). Zatím je tam jen drát od čidla.

### Hlavní jsou data

Co nejvíce dat je dobré zkonvertovat a vyexportovat přímo při měření. Času je tam dost, dat je málo, počítač se fláká, je třeba ho zaměstnat.

Merení	Hmotnost	Prirustek	Cas	Prodleva	Prutok
1	0.04	0.04	0.00	0.38	0.11

Merení	Hmotnost	Prirustek	Cas	Prodleva	Prutok
2	0.03	-0.02	0.78	0.78	-0.02
3	-0.03	-0.05	1.55	0.78	-0.07
4	0.09	0.12	2.33	0.78	0.16

Vzorce jsou vysvětleny u kalibrace. Průtok se počítá jako přírůstek hmotnosti ku přírůstku času a samozřejmě záporný průtok je jen chyba měření.

Konečná hmotnost je celkové množství moči v mililitrech.

Lehce zjistíme maximální průtok (v Python: `max(array)`) a jemu odpovídající čas do max průtoku.

Trošku složitější je zjistit konec močení a tím i celkový čas močení. To je otázka definice, kterou urologové používají.

## Export do PDF coronat opus

Výsledek je třeba exportovat nejprve do obrázků grafů a z grafů udělat html stránku a tu exportovat do PDF. To vše se dělá buď ručně v Calcu/Excelu, nebo automaticky v Python. Vyzkoušel jsem obojí.

Python načítá csv data pomocí knihovny Pandas a přes tu se vyrobí grafy v knihovně Matplotlib. To vše fungovalo elegantně.

Nevyřešený problém t.č. mám jen, jak importovat na Raspberry knihovnu Weasyprint na převod HTML do PDF. Na PC funguje bez problémů, ale do Raspberry se mi to nedaří nahrát. PDFKit se mi tam nechce instalovat, protože to má 600 MB. Takže z Raspberry dnes leze datový soubor CSV, HTML soubor a grafy. PDF dělám až na PC. V zásadě nic zásadního. V kyselině nic kyselého...

## Drobnosti - bzučák

Existují dva typy bzučáků. Jedny, ty aktivní, stačí připojit na napětí, a pak si samy začnou bzučet. S těmi je nejméně programování, ale nemůžete si nastavit třeba melodii - pípají jenom jeden tón. Druhá možnost je mít pasivní piezobzučák, do kterého ale musíte posílat napětí s určitou frekvencí. Můžete si takto naprogramovat i dvojhlasy. S tím jsem se vyblbnul u Arduina, kde to hrálo: "Kdyby byla Morava to, co je Slezsko" a při tom to zalávalo skleník. Po nedlouhé úvaze jsem ale zavrhl myšlenku, že by to při močení hrálo melodii: Nešťastný šafářův dvoreček, či Zořali briezu. Už ju vezú, už ju vezú...". Potlačil jsem touhy po kreativě a připojil fádni aktivní bzučák.

Zapojení opět vcelku primitivní - Ground a jeden GPIO pinů. Programová část v příloze.

## Drobnosti - tlačítko

Uživatelské rozhraní jsem rozšířil o tlačítko pro technické hračky a případné inteligenty. Rozeznává krátký stisk/přepnutí, dvojklik a dlouhý stisk - reset. Na krátký stisk je možno doprogramovat cokoli, třeba report počtu odebraných vzorků (počet pípnutí = počet vzorků). Dvojklik spouští odběr vzorku.

Zapojení vcelku jednoduché na ground a nějaký GPIO pin se připojí zkratující tlačítko a k tomu se napíše krátká procedura, která je níže v citovaném programu.

## Hardwarové datum a čas

Raspberry využívá falešný čas a když je připojeno na Internet, tak se synchronizuje s nějakým časovým serverem. Pro poloautomatické odebrání vzorků je třeba počítat s tím, že váha bude offline, ergo nebude mít žádný přístup k Internetu a času. K tomu bylo třeba koupit časový modul, který se připojí k Raspberry a pomocí baterie a krystalu si udržuje aktuální čas. To není moc triviální záležitost, vyžaduje to hodně systémového nastavování, proto podrobněji.

Koupil jsem si časový modul s popiskou: HZ8563, který se ale často označuje jako pcf8563.

Ten má 6 pinů, ale stačí zapojit jen tyto čtyři. Já jsem je zapojil pomocí starého USB kabelu takto:

### Drát USB kabelu

Červený drát

Bílý drát

Zelený drát

Černý drát

### Pin na HZ8563

VCC

Ground

SDA

SCL

### Pin 5 na Raspberry

pin 5

pin 34

pin 2 SDA

pin 3 SCL

Jestli jsem to dobře pochopil na VCC může být 3V i 5V. Stejně tak můžete použít jakýkoli Ground, co vám na Raspberry ještě zbývá.

### Jak poznat, že mám časový modul správně připojený?

Linux standardně používá dva tři programy, kterými se ovládá datum: `date`, `datetimectl`, `hwclock`. `Date` je univerzální příkaz, který vrací a nastavuje softwarový systémový čas. `Datetimectl` nastavuje a získává čas v



systemech se systemd, což je případ i mého Raspberry 3 Pi. *Hwclock* je právě program pro řízení hardwarového času.

```
sudo hwclock -r --verbose
hwclock from util-linux 2.33.1
System Time: 1697618589.605689
Trying to open: /dev/rtc0
Trying to open: /dev/rtc
Trying to open: /dev/misc/rtc
No usable clock interface found.
hwclock: Cannot access the Hardware Clock via any known method.
```

Takže je třeba nainstalovat tyto programy:

```
sudo apt-get install i2c-tools # popř. ještě python-smbus
```

Spustit konfiguraci Raspberry a povolit I2C:

```
sudo raspi-config
Choose Interfacing Options -> I2C -> Yes.
```

Restartovat Raspberry

```
sudo shutdown -r now
```

Po restartování vyzkoušíme

```
sudo i2cdetect -y 1
 0 1 2 3 4 5 6 7 8 9 a b c d e f
00:  -----
10:  -----
20:  -----
30:  -----
40:  -----
50: -- 51 -----
60:  -----
70:  -----
```

Když vytáhnu konektor z časového modulu, tak to na řádce 50 nehlásí nic. S modulem se objeví toto číslo 51. To mi říká, že můj modul HZ8563 je živý a je správně přidrátován. Jiné moduly hlásí jiné číslo, ale princip je stejný. Mělo by se zde objevovat nějaké číslo po připojení modulu.

Teď je třeba přidat popř. upravit řádku v následujícím souboru:

```
sudo nano /boot/config.txt
dtoverlay=i2c-rtc,param=pcf8563
popř. dtoverlay=i2c-rtc,pcf8563
```

I tento parametr je jiný, pokud máte jiný modul.

```
dtoverlay=i2c-rtc,ds1307
dtoverlay=i2c-rtc,pcf8523
dtoverlay=i2c-rtc,ds3231
```

Po restartu se změnila i zmíněná řádka:

```
sudo i2cdetect -y 1
50: -- UU -----
```

Změnila se i odpověď tohoto programu, byť stále nefunguje, jak má:

```
sudo hwclock -r --verbose
hwclock from util-linux 2.33.1
System Time: 1697628263.254094
Trying to open: /dev/rtc0
Using the rtc interface to the clock.
Assuming hardware clock is kept in UTC time.
Waiting for clock tick...
ioctl(4, RTC_UIE_ON, 0): Invalid argument
Waiting in loop for time from /dev/rtc0 to change
```

```
hwclock: ioctl(RTC_RD_TIME) to /dev/rtc0 to read the time failed: Invalid argument
...synchronization failed
```

Zkusím odinstalovat falešný čas:

```
sudo apt -y remove fake-hwclock
sudo update-rc.d -f fake-hwclock remove
```

Bloudění na Internetu dává tuto radu, která odstranila chybu *Invalid argument ioctl..*:

```
sudo hwclock --systohc -D --noadjfile --utc
```

Tímto začaly fungovat hardwarové hodiny a bylo je možné nastavit:

```
sudo hwclock --set --date "06/05/13 23:00:00"
sudo hwclock --show
2023-10-19 15:49:30.717383+02:00 -0.369063 seconds
sudo shutdown -r now
```

Po restartu zkusíme:

```
sudo hwclock --show
2023-10-19 15:54:31.314873+02:00 -0.761500 seconds
```

Ovšem příkaz *date* ukazoval špatný čas.

Je třeba vypnout toto nastavení v následujícím souboru:

```
sudo nano /lib/udev/hwclock-set
# if [ -e /run/systemd/system ] ; then
#   exit 0
# fi
# toto je třeba zakomentovat
```

A správný čas se nastavil příkazem:

```
timedatectl set-time "2023-10-18 13:48:24"
```

Od té doby Raspberry čte datum a čas z časového modulu a stále vrací čas o minutu posunutý oproti oficiálnímu času. Z toho vím, že časový modul funguje správně.

Možná znáte vtip: Které hodiny jsou lepší? Ty, co stojí, nebo ty co jdou o minutu napřed? Odpověď zněla: Ty, co stojí, protože ty dvakrát denně ukazují správný čas.

Pár odkazů:

```
https://forums.raspberrypi.com//viewtopic.php?p=364303#p364303
https://pimylifeup.com/raspberry-pi-rtc/
```

## Další plány

Výzvu zvanou urologická váha nyní považuji za sebe za prakticky dokončenou. Vše ostatní je otázkou jen leštění softwaru. To jsou sice hodiny práce, ale to hlavní je funkční a již v tomto stavu je to použitelné pro diagnostiku.

Pak je celá řada možností, jak se vypořádat s Raspberry a zpracováním dat, třeba:

A) Předělat Raspberry na samostatný počítač s XWindows, který bude mít vlastní wifi access point nebo který se sám může připojit k místní wifině a přes ní posílat výsledky kam libo.

B) Nechat Raspberry headless (tzn. bez windows), jen přidělat samostatný access point na wifinu a přes něj si stahovat data - ať v polotovaru, nebo kompletní výstupy.

C) Předělat to na Arduino a zálohovat data na flash disk. Ty zpracovávat na PC. Komandovat ho přes příkazy v souborech na FlashDisku. Je to opičárna, ale funguje to. Řešení přes FlashDisk má výhodu v možnosti kódování dat, aby nebyly problémy s Evropskou Uhnijí, GPRS a jinými nesmysly.

### **Přelepit štítek na váze a prodat to dr. Nešporovi jako blicí váhu**

Alkoholik si napřed připraví váhu. Načež si vezme prášek Antabus, a pak je rád, že má už připravenou blicí váhu, která kontinuálně zapisuje okamžitý průtok nestlačitelných kapalin jeho ezofágem.

### **Vykašlat se na urologickou váhu a celé to předělat na seismograf**

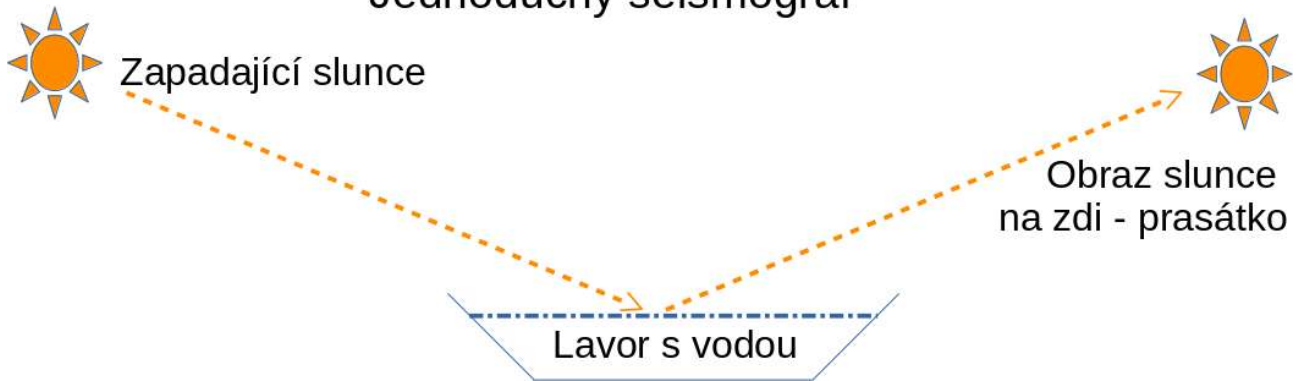
To by znamenalo vyhodit AD převodník HX711 a přímo číst surová data z čidla, asi nejlépe přes Arduino

a s nějakým datovým razítkem je logovat na FlashDisk.

Na váhu je ale třeba přes tyč dát závaží (s tlumičem, kdyby to chtělo pérovat).

Každopádně nejjednodušší dost citlivý seismometr je lavor s vodou, na který svítí zapadající slunce. To na protější stěnu dělá prasátko, které se občas jen tak samo od sebe a úplně zadarmo rozverně zachvěje.

### Jednoduchý seismograf



[http://klimes.mysteria.cz/clanky/ostatni/priklady\\_fyzika/fyzika.htm#seismograf](http://klimes.mysteria.cz/clanky/ostatni/priklady_fyzika/fyzika.htm#seismograf)

Mám zkušenost, že když do takového lavoru s vodou nešlápne omylem žena v bačkorech s bambulí, tak se jí při pohledu na neposedné prasátko rozzáří oči. Takže za pokus to stojí.

To jsem však odbočil od urologické váhy až přespříliš.

# Měření č. 1

Motto: Močení k lásce schůdeček,  
s pozdravem Běda Hudeček

Hlavička

Soubor vzorku: vzorek\_54.dat

Jméno a příjmení: PhDr. Mgr. Jeroným Klimeš, Ph.D.

Rodné číslo: 670101/0110

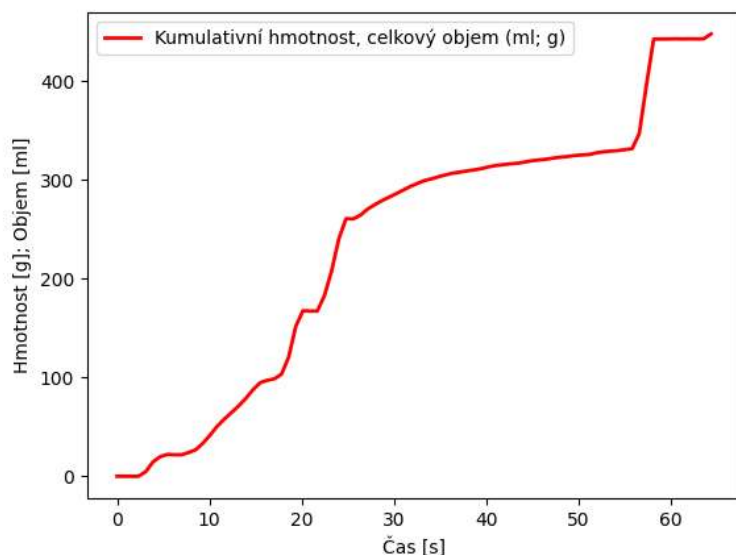
Pojišťovna: VZP

Kód pojišťovny: 111

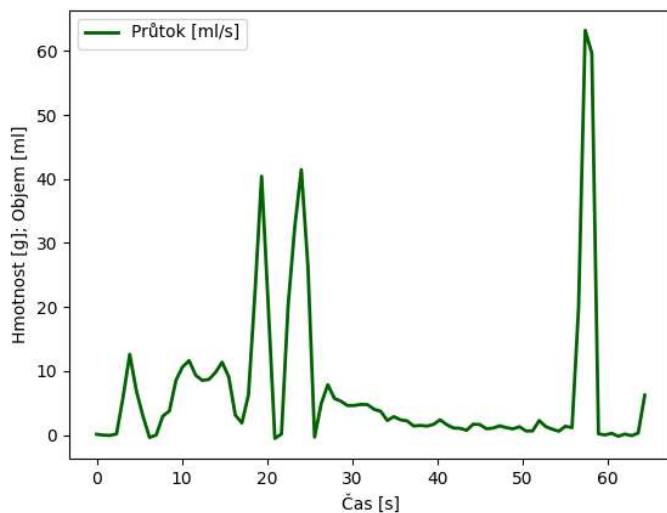
Datum a čas odběru vzorku: 2023-10-19 11:59:01

Výsledky - data a grafy

## Kumulativní graf



## Distribuční graf - průtok



## Data

Mereni	Hmotnost	Prirustek	Cas	Prodleva	Prutok
1	0.04	0.04	0.00	0.38	0.11
2	0.03	-0.02	0.78	0.78	-0.02
3	-0.03	-0.05	1.55	0.78	-0.07
4	0.09	0.12	2.33	0.78	0.16
5	4.71	4.62	3.11	0.78	5.94
6	14.52	9.80	3.89	0.78	12.61
7	19.88	5.37	4.66	0.78	6.90
8	22.16	2.27	5.44	0.78	2.93
9	21.85	-0.30	6.21	0.77	-0.39
10	21.84	-0.02	6.99	0.78	-0.02
11	24.12	2.28	7.77	0.78	2.93
12	27.03	2.91	8.54	0.78	3.75
13	33.69	6.66	9.32	0.78	8.57
14	41.89	8.20	10.09	0.77	10.64
15	50.85	8.96	10.86	0.77	11.61
16	58.05	7.20	11.63	0.77	9.33
17	64.63	6.59	12.41	0.77	8.54
18	71.38	6.74	13.18	0.78	8.67
19	78.98	7.61	13.96	0.78	9.78
20	87.82	8.84	14.74	0.78	11.37
21	94.87	7.05	15.52	0.78	9.07
22	97.26	2.39	16.29	0.77	3.09
23	98.71	1.45	17.06	0.77	1.88
24	103.55	4.84	17.83	0.77	6.28
25	120.81	17.26	18.60	0.77	22.37
26	152.02	31.21	19.37	0.77	40.43
27	167.78	15.76	20.15	0.78	20.25
28	167.36	-0.42	20.93	0.78	-0.55
29	167.48	0.12	21.71	0.78	0.16
30	183.23	15.74	22.49	0.78	20.23
31	208.23	25.01	23.26	0.77	32.39
32	240.53	32.30	24.04	0.78	41.48
33	261.03	20.49	24.82	0.78	26.32
34	260.76	-0.27	25.60	0.78	-0.35
35	264.58	3.82	26.37	0.78	4.90
36	270.69	6.12	27.15	0.78	7.86
37	275.10	4.40	27.93	0.77	5.71
38	279.16	4.06	28.70	0.77	5.27
39	282.71	3.55	29.47	0.77	4.59
40	286.30	3.59	30.25	0.78	4.61
41	290.01	3.71	31.03	0.78	4.77
42	293.70	3.69	31.80	0.78	4.75
43	296.83	3.12	32.58	0.78	4.01
44	299.71	2.88	33.36	0.78	3.70
45	301.46	1.75	34.13	0.77	2.27
46	303.67	2.21	34.91	0.77	2.86
47	305.50	1.83	35.68	0.77	2.38
48	307.22	1.72	36.46	0.78	2.20
49	308.31	1.09	37.23	0.78	1.40
50	309.47	1.16	38.01	0.78	1.49
51	310.55	1.08	38.79	0.78	1.39
52	311.83	1.29	39.57	0.78	1.66
53	313.66	1.83	40.34	0.77	2.37
54	314.92	1.25	41.11	0.77	1.62
55	315.77	0.85	41.89	0.78	1.09
56	316.58	0.81	42.67	0.78	1.05
57	317.15	0.57	43.45	0.78	0.73
58	318.46	1.31	44.22	0.78	1.68
59	319.72	1.26	45.00	0.77	1.63
60	320.48	0.76	45.77	0.78	0.98
61	321.31	0.83	46.55	0.78	1.07
62	322.42	1.11	47.33	0.78	1.42
63	323.31	0.89	48.11	0.78	1.15
64	324.05	0.74	48.89	0.78	0.95
65	325.04	0.99	49.66	0.77	1.28
66	325.52	0.48	50.43	0.77	0.63
67	326.01	0.49	51.20	0.77	0.63
68	327.75	1.74	51.97	0.77	2.26
69	328.76	1.01	52.74	0.77	1.31
70	329.45	0.69	53.51	0.77	0.90
71	329.91	0.45	54.29	0.77	0.59
72	330.96	1.05	55.06	0.77	1.36
73	331.84	0.88	55.83	0.77	1.14
74	347.20	15.36	56.60	0.77	19.90
75	396.41	49.22	57.38	0.78	63.23
76	442.82	46.41	58.16	0.78	59.64
77	442.96	0.14	58.94	0.78	0.18
78	442.97	0.00	59.71	0.78	0.00
79	443.17	0.20	60.49	0.77	0.26
80	443.01	-0.16	61.26	0.78	-0.20
81	443.13	0.12	62.04	0.78	0.15
82	443.04	-0.09	62.82	0.78	-0.11
83	443.27	0.23	63.60	0.78	0.29
84	448.11	4.85	64.38	0.78	6.23

Konec ukázky výstupu

## Program v Python k 2023-10-21

Z hlediska hardware plně funkční program, z uživatelského, softwarového hlediska je samozřejmě nevládný.

Implementuje HW datum, čidlo váhy a jeho kalibraci, ovládání tlačítkem a signalizaci aktivním bzučákem, export výsledku do csv souboru, převod do html a pdf souboru s kumulativní a distribuční křivkou.

```
#!/usr/bin/python3
#-*- coding: utf-8 -*-

#sed -rn '/[0-9]*t[^t]*t[^t]*t[^t]*p' vzorek_3.dat # najde mereni
#sed -rn 's/([0-9]*)t([^\t]*)t([^\t]*)t([^\t]*)^4/p' vzorek_3.dat vybere pozadovany sloupec

from active_buzzer import active_buzzer
global active_buzzer_flag

# pip3 install --break-system-packages weasyprint
# pip3 install --break-system-packages pandas
# pip3 install --break-system-packages matplotlib

import pandas
import datetime
import time
# import RPi.GPIO as GPIO # pouze v raspberry
import sys
import re
import shutil
import os
import socket # zjistovani systemu
#sudo apt-get install libopenjp2-7
# pak jsem musel preinstalovat pillow - smazat, znova nainstalovat
#import string

pin_button = 25
pin_buzzer = 17
step = 0.05 # sekundy je cyklus
limit_meziklik=3 # tři kroky, tedy priblizne 0,2 sekundy
limit_dlouhostisk=20 # cca 5 sekund
mlceni=False

debug=True
debug=False

#Set warnings off (optional)

prepinac = False
dvojklik = False
last_state = False
prvoklik = False
meziklik = 0
dlouhostisk = 0

soubor_vzorku_na_zpracovani="vzorek_54.dat"
jmeno_prijmeni="PhDr. Mgr. Jeroným Klimeš, Ph.D."
rodne_cislo="670101/0110"
pojistovna="VZP"
kod_pojistovny="111"
datum_odberu_vzorku=datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')

def nacteni_ulozeni_vzorku():
    import RPi.GPIO as GPIO # pouze v raspberry
    prodleva=0.2 # prodleva mezi merenimi - bohuzel neumi to mene nez 1 s
    counter_dojezd=20 # pocita to ## mereni, kdy se nic nedeje, pak to cele ukonci. V ostre verzi to bude minimalne 30 sekund
    try:
        soubor_cislo_vzorku = open("cislo_vzorku.dat", "r")
    except IOError:
        soubor_cislo_vzorku = open("cislo_vzorku.dat", "w")
        soubor_cislo_vzorku.write("1")
        soubor_cislo_vzorku.close()
    soubor_cislo_vzorku.close()

    soubor_cislo_vzorku = open("cislo_vzorku.dat", "r")
    try:
        cislo_vzorku=int(soubor_cislo_vzorku.read().strip())
    except ValueError:
        soubor_cislo_vzorku = open("cislo_vzorku.dat", "w")
        soubor_cislo_vzorku.write("1")
        soubor_cislo_vzorku.close()
        cislo_vzorku=1

jmeno_souboru_vzorku="vzorek_"+str(cislo_vzorku)+".dat"
print("Cislo vzorku je %d" % cislo_vzorku)
```

```

soubor_cislo_vzorku = open("cislo_vzorku.dat", "w")
soubor_cislo_vzorku.write(str(cislo_vzorku+1))
soubor_cislo_vzorku.close()

EMULATE_HX711=False

referenceUnit = 386 # funkcni vaha
referenceUnit = 1036 # jednorazova urologicka vaha
referenceUnit = 999 # jednorazova urologicka vaha druhy odhad
referenceUnit = 1020

if not EMULATE_HX711:
    import RPi.GPIO as GPIO
    from hx711 import HX711
else:
    from emulated_hx711 import HX711

def cleanAndExit():
    print("Cleaning...")

    if not EMULATE_HX711:
        GPIO.cleanup()

    print("Bye!")
    sys.exit()

hx = HX711(5, 6)

# I've found out that, for some reason, the order of the bytes is not always the same between versions of python, numpy and the hx711 itself.
# Still need to figure out why does it change.
# If you're experiencing super random values, change these values to MSB or LSB until to get more stable values.
# There is some code below to debug and log the order of the bits and the bytes.
# The first parameter is the order in which the bytes are used to build the "long" value.
# The second paramter is the order of the bits inside each byte.
# According to the HX711 Datasheet, the second parameter is MSB so you shouldn't need to modify it.
hx.set_reading_format("MSB", "MSB")
# hx.set_reading_format("LSB", "LSB") # timto problem nebyl

# HOW TO CALCULATE THE REFERENCE UNIT
# To set the reference unit to 1. Put 1kg on your sensor or anything you have and know exactly how much it weights.
# In this case, 92 is 1 gram because, with 1 as a reference unit I got numbers near 0 without any weight
# and I got numbers around 184000 when I added 2kg. So, according to the rule of thirds:
# If 2000 grams is 184000 then 1000 grams is 184000 / 2000 = 92.
#hx.set_reference_unit(113)
hx.set_reference_unit(referenceUnit)

hx.reset()

hx.tare()

print("Tare done! Add weight now...")
print

max_prutok=0
mereni=1
stara_hmotnost=0
stary_h_prirustek=0
stary_c_prirustek=0
counter_prirustek=0
soubor_vzorku = open(jmeno_souboru_vzorku, "w")
print("Jmeno souboru: %s" % jmeno_souboru_vzorku)
soubor_vzorku.write("Soubor vzorku:\t%s\n" % jmeno_souboru_vzorku)
soubor_vzorku.write("Jméno a příjmení: "+jmeno_prijmeni+"\n")
soubor_vzorku.write("Rodné číslo: "+rodne_cislo+"\n")
soubor_vzorku.write("Pojišťovna: "+pojistovna+"\n")
soubor_vzorku.write("Kód pojišťovny: "+kod_pojistovny+"\n")
soubor_vzorku.write("Datum a čas odběru vzorku: "+datum_odberu_vzorku+"\n")
# print("Datum a čas odběru vzorku: %s\n" % str(datetime.datetime.now().isoformat()))
# soubor_vzorku.write("datum_cas_nyni:\t%s\n\n" % str(datetime.datetime.now().isoformat()))
print("")
print("Nepresnost mereni je cca 2%")
print("")
print("Mereni\tHmotnost\tPrirustek\tCas\tProdleva\tPrutok")
soubor_vzorku.write("\nMereni\tHmotnost\tPrirustek\tCas\tProdleva\tPrutok\n")

# to use both channels, you'll need to tare them both
hx.tare_A() #bylo zapoznamkovany
hx.tare_B() #bylo zapoznamkovany, ale u jednorazove urologicke vahy musi byt obe zapnute

#while True:
stary_cas=datetime.datetime.now().timestamp()
while counter_prirustek<counter_dojezd:

```

```

try:
    # These three lines are usefull to debug wether to use MSB or LSB in the reading formats
    # for the first parameter of "hx.set_reading_format("LSB", "MSB)".
    # Comment the two lines "val = hx.get_weight(5)" and "print val" and uncomment these three lines to see what it prints.

    # np_arr8_string = hx.get_np_arr8_string()
    # binary_string = hx.get_binary_string()
    # print binary_string + " " + np_arr8_string

    # Prints the weight. Comment if you're debbuging the MSB and LSB issue.

    buzzer_od=5
    buzzer_do=9
    if (mereni > buzzer_od and mereni < buzzer_do): active_buzzer("on")
    if (mereni > (buzzer_do-1)): active_buzzer("off")

    hmotnost = hx.get_weight(5)
    h_prirustek=hmotnost-stara_hmotnost
    if h_prirustek >max_prutok:
        max_prutok=h_prirustek
    cas=datetime.datetime.now().timestamp() # je to float s nanosekundami
    c_prirustek=cas-stary_cas
    prutok=h_prirustek/c_prirustek
    print("%s\t%s\t%s\t%s\t%s\t%s\t%s" % (mereni, round(hmotnost,3), round(h_prirustek,3), round(cas,3), round(c_prirustek,3),
round(prutok,3)))
    if (abs(h_prirustek)<1 and abs(stary_h_prirustek)<1):
        counter_prirustek=counter_prirustek+1
    else:
        counter_prirustek=0
    if (abs(h_prirustek)>100):
        counter_prirustek=1000
    print("\nManipulace s nadobou: ", str(h_prirustek))
#     counter_prirustek=5
    soubor_vzorku.write(str(mereni)+"\t"+str(round(hmotnost,3))+"\t"+str(round(h_prirustek,3))+"\t"+str(round(cas,3))+"\t"+str(round(c_prirustek,3))+"\t"+str(round(prutok,3))+"\n")

    mereni=mereni+1
    stara_hmotnost=hmotnost
    stary_cas=cas
    stary_h_prirustek=h_prirustek
    hx.power_down()
    hx.power_up()
    time.sleep(prodleva)
except (KeyboardInterrupt, SystemExit):
    active_buzzer("off")
    cleanAndExit()

    # To get weight from both channels (if you have load cells hooked up
    # to both channel A and B), do something like this
    #val_A = hx.get_weight_A(5)
    #val_B = hx.get_weight_B(5)
    #print "A: %s B: %s" % ( val_A, val_B )
active_buzzer("off")
print("")
print("Maximalni h_prirustek: %s ml" % max_prutok)
soubor_vzorku.write("\nmaximalni_h_prirustek:\t%s\n" % str(max_prutok))
soubor_vzorku.write("\nsoubor_ulopen:\t%s\n" % jmeno_souboru_vzorku)
if debug: print("Debug: %d" % counter_prirustek)
soubor_vzorku.close()
print("Soubor %s byl ulozen." % jmeno_souboru_vzorku)
active_buzzer(3) # pipne a konec
active_buzzer("off")

def pipac(pocet):
    global mlceni
    if (mlceni==False):
        for i in range(0,pocet):
            GPIO.output(pin_buzzer,GPIO.HIGH)
            time.sleep(0.1)
            GPIO.output(pin_buzzer,GPIO.LOW)
            time.sleep(0.1)

def tlacitko():
    print ("Tlacitko", 1-button_state)

def prepinatko(stav):
    if (stav):
        print ("    Prepinac 1")
    else:
        print ("    Prepinac 0")

def dvojklikatko(stav):

```



```

print("Dvojklik - nacteni vzorku")
nacteni_ulozeni_vzorku();
#global mlceni
#if (stav):
    #print("        Dvojklik A - mlceti zlato")
    #mlceni=True
#else:
    #print("        Dvojklik B - pipati srebro")
    #mlceni=False

def dlouhostiskatko():
    print("Dlouhý stisk - konec")
    quit()
    #help_i_need_somebody()

if (socket.gethostname()=="raspberrypi"):
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(pin_button,GPIO.IN,pull_up_down=GPIO.PUD_UP)
    GPIO.setup(pin_buzzer,GPIO.OUT)
    while True:
        button_state = GPIO.input(pin_button)
        if button_state!=last_state:
            if (debug):
                print("Tlacitko", 1-button_state, "Meziklik",meziklik, "Prvoklik", prvoklik)
            else:
                tlacitko()
        if button_state == 0: # tlacitko stlacene
            if (last_state!=button_state): # doslo ke zmene
                if (prvoklik==True and meziklik<=limit_meziklik): # tak je to dvojklik
                    pipac(2)
                    if dvojklik==True:
                        dvojklik=False
                        dvojklikatko(False)
                    else:
                        dvojklik=True
                        dvojklikatko(True)
                    prvoklik=False
                    dlouhostisk=0
                elif (meziklik>limit_meziklik): # pri stisku nulujeme citace
                    prvoklik=True # Toto je prvoklik, ubehla doba pro dvojklik
                    meziklik=0
                else: # nacitani dlouhostisku a vyhodnoceni.
                    dlouhostisk=dlouhostisk+1
                    if dlouhostisk>limit_dlouhostisk:
                        pipac(3)
                        meziklik=0
                        dlouhostisk=0
                        prvoklik=False
                        dlouhostiskatko()
                    else: # tlacitko je pustene
                        if (prvoklik==True and meziklik>limit_meziklik): # vyprsel cas dvojklik,
                            # takze je to obycejny staceni/prepnuti
                            pipac(1)
                            if prepina==False:
                                prepina=True
                                prvoklik=False
                                prepinaatko(True)
                                dlouhostisk=0
                            elif prepina==True:
                                prepina=False
                                prvoklik=False
                                prepinaatko(False)
                                dlouhostisk=0
                                meziklik=meziklik+1 # pocitame prubehy
                                last_state=button_state
                                time.sleep(step)

def zpracovani_vzorku():
    import weasyprint
    zdroj= r'./template.htm'
    zdroj= os.path.abspath(zdroj)
    vystup_htm = r'./vystup_odpad.htm'

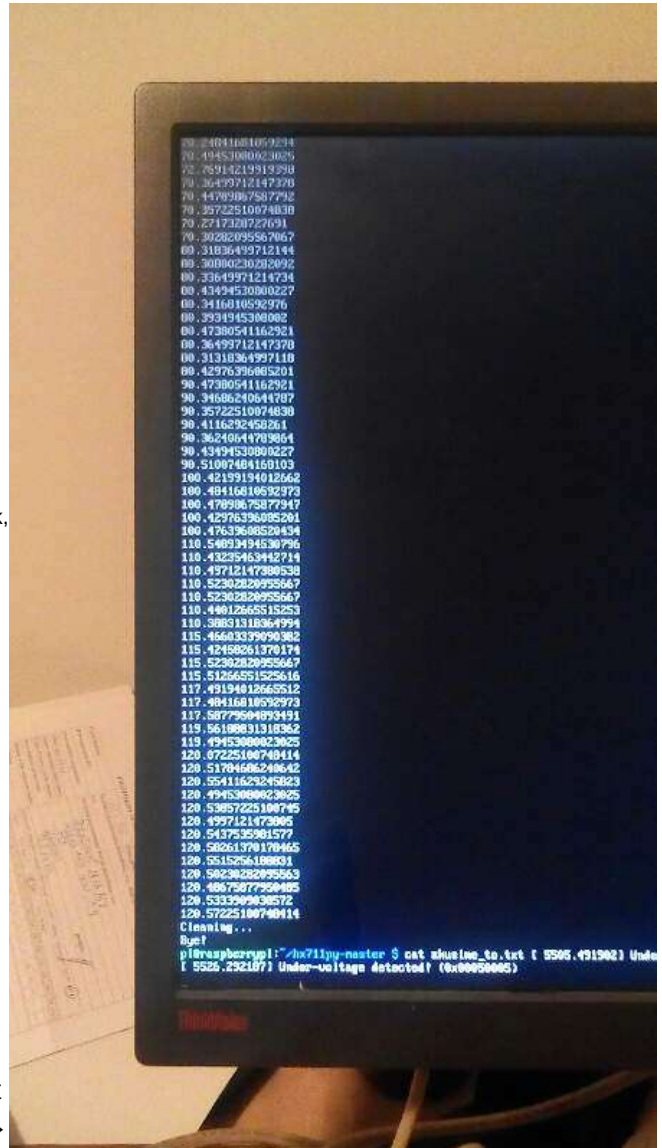
```

## Výpis surových dat na monitor z Raspberry →

```

vystup_htm = os.getcwd()+"/"+vystup_htm
vystup_pdf = r'./vystup_odpad.pdf'
vystup_pdf = os.getcwd()+"/"+vystup_pdf
shutil.copyfile(zdroj, vystup_htm)
vystup= open(vystup_htm, "a")
zdroj_vzorek=open(soubor_vzorku_na_zpracovani, "r")

```



```

for i in range(6):
    vystup.write("<p>"+zdroj_vzorek.readline()+"</p>\n")
    vystup.write("#####\n")
<h2>Výsledky - data a grafy</h2>
<p><b>Kumulativní graf</b></p>
<img width=40% border=0 src=kumulativni.png>
<p><b>Distribuční graf</b></p>
<img width=40% border=0 src=distribucni.png>
<h2>Výsledky - data</h2>\n
#####)
zdroj_vzorek.close()
#nactena_mereni=pandas.read_csv("vzorek_22.dat", skiprows=4, delimiter="\t", infer_datetime_format=False, header=1 )
nactena_mereni=pandas.read_csv(soubor_vzorku_na_zpracovani, skiprows=8, delimiter="\t", header=0, skip_blank_lines=False )
#df = pd.read_csv(<filepath>, skip_blank_lines=False)
prazdna_radka = nactena_mereni.loc[nactena_mereni.isnull().all(1)]
if len(prazdna_radka) > 0:
    cislo_prvni_prazdne_radky = prazdna_radka.index[0]
    nactena_mereni = nactena_mereni[:cislo_prvni_prazdne_radky]
    nactena_mereni = nactena_mereni[:-1] # odrizne pripadnou manipulaci s nadobou
    nactena_mereni["Hmotnost"]=pandas.to_numeric(nactena_mereni["Hmotnost"])
    nactena_mereni["Cas"]=nactena_mereni["Cas"]-nactena_mereni["Cas"].loc[0]
    vystup.write(nactena_mereni.round(2).to_html(index=False))
    print(nactena_mereni.round(2))

vystup.write("</body>\n</html>")
vystup.close()

#matice=pandas.array(nactena_mereni)
#print(nactena_mereni.loc(3,5))
print()
#print(nactena_mereni["Cas"])
#print(nactena_mereni["Hmotnost"])
import matplotlib.pyplot as plt
#plt.plot(nactena_mereni["Cas"]-nactena_mereni["Cas"].loc[0], nactena_mereni["Hmotnost"], color='red',linewidth=2.0,label="Kumulativní
hmotnost, celkový objem (ml; g)")
plt.plot(nactena_mereni["Cas"]-nactena_mereni["Cas"].loc[0], nactena_mereni["Hmotnost"], color='red',linewidth=2.0,label="Kumulativní
hmotnost, celkový objem (ml; g)")
plt.ylabel("Hmotnost [g]; Objem [ml]")
plt.xlabel("Čas [s]" )

#plt.plot(nactena_mereni["Mereni"], label="Měření")
#plt.plot(nactena_mereni["Hmotnost"], label="Hmotnost, objem")
plt.legend()
plt.savefig("kumulativni.png")
plt.show()
plt.close()

plt.plot(nactena_mereni["Cas"]-nactena_mereni["Cas"].loc[0], nactena_mereni["Prutok"], linewidth=2.0,color='darkgreen',label="Průtok
[ml/s]")
plt.ylabel("Hmotnost [g]; Objem [ml]")
plt.xlabel("Čas [s]" )

plt.legend()
plt.savefig("distribucni.png")
plt.show()
plt.close()

weasyprint.HTML(vystup_htm).write_pdf(vystup_pdf) # funguje
if (socket.gethostname()=="T430" or socket.gethostname()!="raspberrypi" ):
    zpracovani_vzorku(); quit()

```

# Program v Python k 2023-08-11

```
#!/usr/bin/python3
#-*- coding: utf-8 -*-

#sed -rn '/[0-9]*\t[^\\t]*\t[^\\t]*\t[^\\t]*\t/p' vzorek_3.dat # najde mereni
#sed -rn 's/([0-9]*)\t([\\t]*)\t([\\t]*)\t([\\t]*)/4/p' vzorek_3.dat vybere pozadovany sloupec
import time
import sys
import datetime
import re
import shutil
import os
import pandas
#sudo apt-get install libopenjp2-7
# pak jsem musel preinstalovat pillow - smazat, znova nainstalovat
#import string
debug=False
debug=True

def nacteni_ulozeni_vzorku():
    prodleva=0.2 # prodleva mezi merenimi - bohuzel neumi to mene nez 1 s
    counter_dojezd=20 # pocita to ## mereni, kdy se nic nedeje, pak to cele ukonci. V ostre verzi to bude
    minimalne 30 sekund
    try:
        soubor_cislo_vzorku = open("cislo_vzorku.dat", "r")
    except IOError:
        soubor_cislo_vzorku = open("cislo_vzorku.dat", "w")
        soubor_cislo_vzorku.write("1")
        soubor_cislo_vzorku.close()
    soubor_cislo_vzorku.close()

    soubor_cislo_vzorku = open("cislo_vzorku.dat", "r")
    try:
        cislo_vzorku=int(soubor_cislo_vzorku.read().strip())
    except ValueError:
        soubor_cislo_vzorku = open("cislo_vzorku.dat", "w")
        soubor_cislo_vzorku.write("1")
        soubor_cislo_vzorku.close()
        cislo_vzorku=1

    jmeno_souboru_vzorku="vzorek_"+str(cislo_vzorku)+".dat"
    print("Cislo vzorku je %d" % cislo_vzorku)

    soubor_cislo_vzorku = open("cislo_vzorku.dat", "w")
    soubor_cislo_vzorku.write(str(cislo_vzorku+1))
    soubor_cislo_vzorku.close()

EMULATE_HX711=False

referenceUnit = 386 # funkčni váha
referenceUnit = 1036 # jednorazová urologická váha váha
referenceUnit = 999 # jednorazová urologická váha druhý odhad
referenceUnit = 1020

if not EMULATE_HX711:
    import RPi.GPIO as GPIO
    from hx711 import HX711
else:
    from emulated_hx711 import HX711

def cleanAndExit():
    print("Cleaning...")

    if not EMULATE_HX711:
        GPIO.cleanup()

    print("Bye!")
    sys.exit()

hx = HX711(5, 6)

# I've found out that, for some reason, the order of the bytes is not always the same between versions of
python, numpy and the hx711 itself.
# Still need to figure out why does it change.
# If you're experiencing super random values, change these values to MSB or LSB until to get more stable
values.
# There is some code below to debug and log the order of the bits and the bytes.
# The first parameter is the order in which the bytes are used to build the "long" value.
# The second paramter is the order of the bits inside each byte.
```