


ψsibox

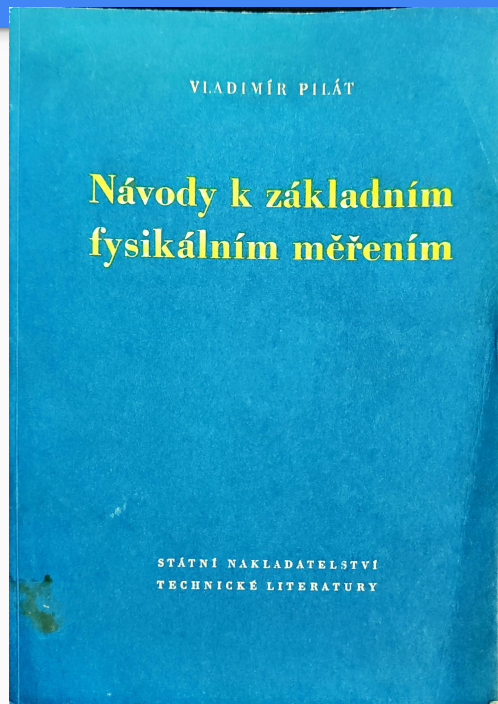
Řešení pro fyzikální experimenty

Dominik Matoulek

OpenAlt - 2.11.2024

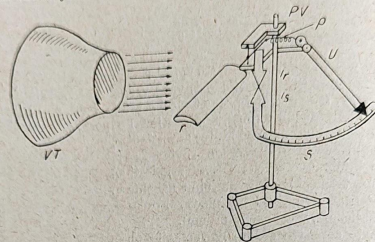


Fyzikální experiment?



Přípravné práce:

Měření obou složek F_x , F_y provádíme pomocí vážek a vzduchového tunelu (obr. 28). Zdrojem vzduchového proudu je vzduchový tunel VT . V ose plechového válce je umístěna čtyřlístá vrtule poháněná elektrickým motorem na střídavý proud. Její rychlost otáčení, a tím i rychlost vzduchového proudu, lze měnit regulací elektrického proudu reostatem. Vzniklý vzduchový proud proudí soustavou stejně dlouhých i širokých trubek, rovnoběžných s osou celého tunelu. Těmito usměrňujícími trubkami se vzduchový proud homogenizuje, t. j. proudnice jsou rovnoběžné a jejich hustota je v celém proudu stejná. Proud je laminární, bez vírů a přesvědčíme se o tom zkou-



Obr. 28. Vzdušný tunel a pérové vážky

šečem vírů. Je to malá obdélníková destička otáčivá kolem osy, kterou orientujeme rovnoběžně se směrem proudu. Destička se rotočí jen v místech, kde vzniknou víry (turbulence).

Pro vlastní měření sil používáme pérové vážky PV (obr. 28). Je to nerovnoramenná dvojzvrtná páka, na jejíž delší rameno r vsuneme tyčku s vyšetřovaným tělesem. K druhému ramenu spojenému pevně se šipkou — indexem i , — je připevněno pružné péro p , které se napíná dvojzvrtnou nerovnoramennou pákou druhého ukazatele U , až šipka vážek ukazuje proti šípce i , pevně spojené se stupnicí. Těmito vážkami lze měřit jak vodorovné síly (odpor F_x), tak i síly vertikální (vztlak F_y).

Kalibrace vážek:

1. Vážky nasaun na stativ.
2. Uvolní páčkou kloub a vážky přetáče, až budou šipky indexů vodorovné. Stupnice je pod vodorovnou rovinou vážek, které kývdají ve svislé rovině.
3. Do otvoru delšího uvolněného ramena r vsuň tyčku s kruhovou destičkou nahoru. Pak přišroubuj pákovitého ukazatele na počátek stupnice.
4. Na posuvný háček na druhém kratším ramenu vážek závěs pomocné závažky a posuň je tak, až jsou oba indexy proti sobě. Tím jsou vážky v nulové poloze připraveny ke kalibraci.
5. Na destičku vlož postupně 1, 2, 3, 5, 10 ... gramů a ramenem ukazatele páky napni vždy péro, až závažíčko vyvážíš. Poznáš to podle toho, že se obě šipky uklíní přímo proti sobě. Ukazatele přišroubuj ke stupnici. Zapiš velikost závažíčka m i velikost výchylky n .
6. Nakresli kalibrační křivku vážek. Na osu úseček nanášej výchylku n a na osu pořadnic hmotu příslušného závažíčka m .

Always has been

Ono ale už něco existuje, ne?

Pasco

- Renomovaný výrobce vzdělávacího a vědeckého vybavení, zaměřený na produkty pro školy a laboratoře.
- Senzory pro fyzikální a přírodní vědy.
- Komplexní laboratorní sady pro experimenty.
- Kompletní ekosystém přístrojů a softwaru.

Vernier

- Specializuje na výrobu digitálních měřicích přístrojů pro školní laboratoře, zejména pro oblast fyziky, chemie a biologie.
- Rozmanité senzory (např. Go Direct Acceleration Sensor, CO₂ senzory).
- Vlastní software pro analýzu dat, zaměřený na snadnou obsluhu učiteli a studenty.
- Spolehlivost a snadné použití.



Uzavřená
komerční řešení
s vysokou
cenou a omezenou
flexibilitou.



Open-source,
modulární,
levné a
snadno
přizpůsobitelné řešení.

Cíle projektu

- Otevřenost
 - Modularita
 - Dostupnost
 - Přizpůsobitelnost
-
- Opravitelnost
 - Nebojíte se ho zničit ;)



Jak na to?

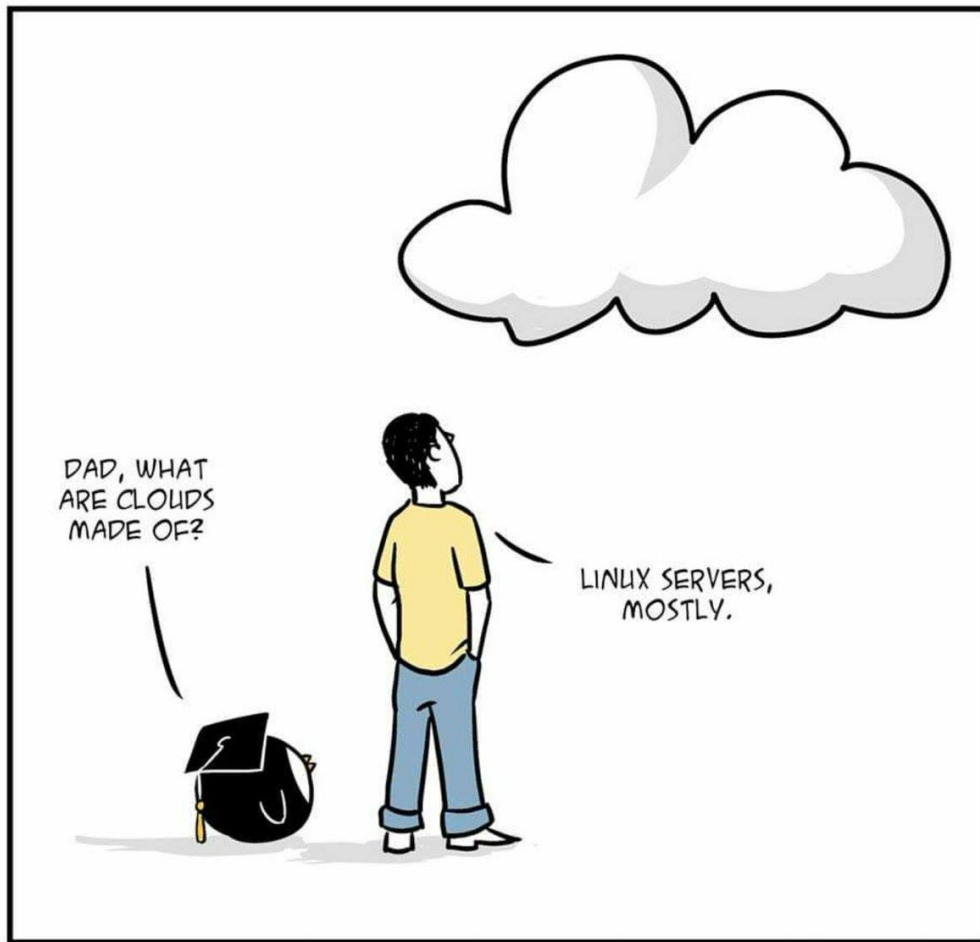
1. Udělat “krabičku” se senzory
2. Udělat “vizualizační” nástroj
3. Profit?

Krabička

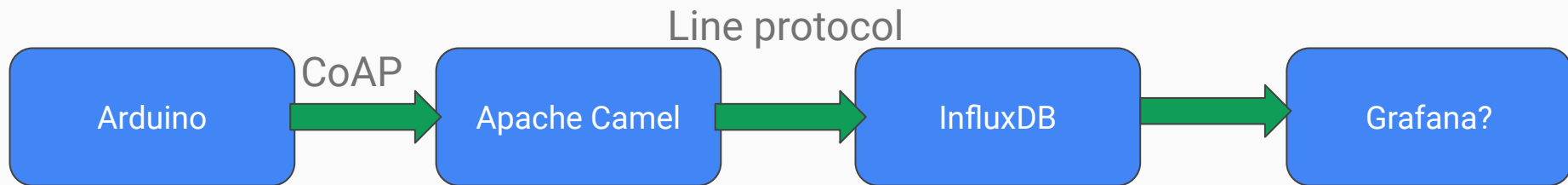
- Modulární
 - Otevřená
 - Opravitelná
-
- Arduino?

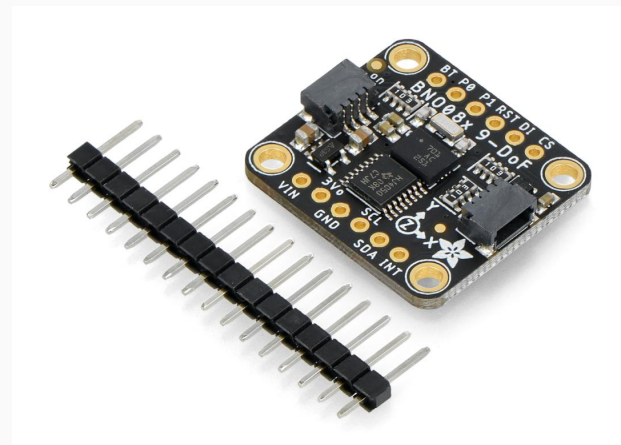
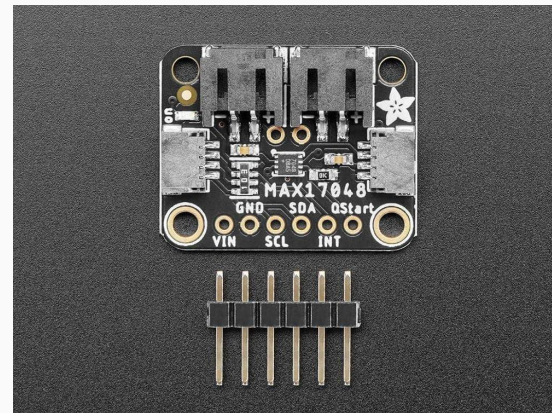
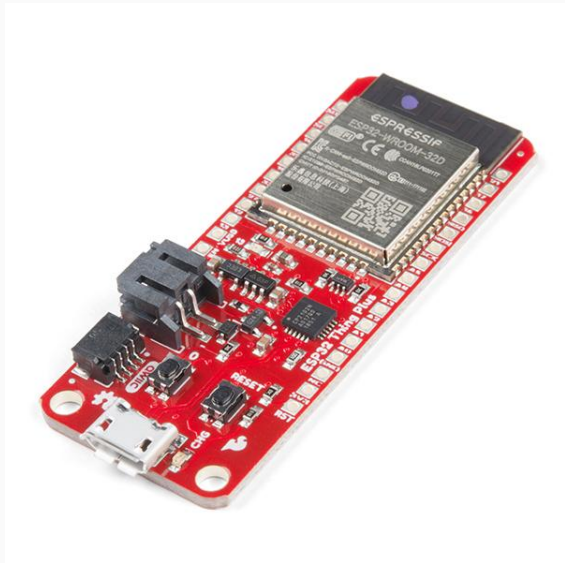
Malá odbočka: Arduino Cloud

- Platforma umožňující snadné připojení Arduino zařízení k internetu, sběr dat a jejich vizualizaci v cloudu.
- Snadná integrace s Arduino IoT projekty a vizualizace dat v jejich dashboardu.
- Data jsou přístupná odkudkoliv díky cloudové infrastruktuře.



Takže jak teda?





Střípky Arduino kódu

```
// CoAP client response callback
void callback_response(CoapPacket &packet,
  IPAddress ip, int port) {

  char p[packet.payloadlen + 1];

  memcpy(p, packet.payload, packet.payloadlen);

  p[packet.payloadlen] = NULL;

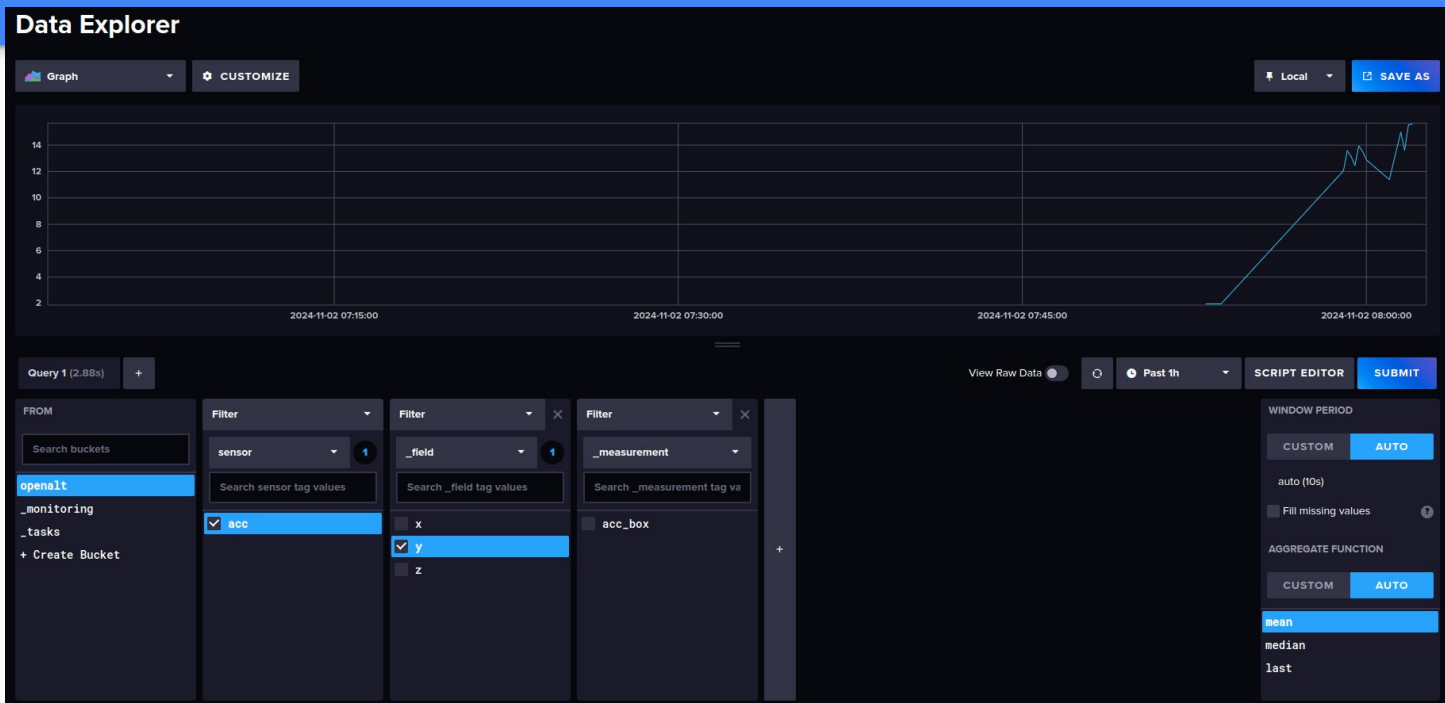
  const char* acc_message = "[" + String(accelerometer[0])
    + "," + String(accelerometer[1]) + "," +
    String(accelerometer[2]) + "].c_str();

  coap.put(server_address, 5683, "acc", acc_message);
}
```

Camel integrace (CoAP -> InfluxDB)

```
from("coap://0.0.0.0:5683/acc") .process(exchange -> {  
    // Zpracování CoAP dat vynecháno  
  
    String influxData = String.format("acc_box,sensor=%s x=%.2f,y=%.2f,z=%.2f", type, x, y, z);  
    exchange.getIn().setBody(influxData);  
})  
    .setHeader(Exchange.HTTP_METHOD, constant("POST"))  
    .setHeader(Exchange.CONTENT_TYPE, constant("text/plain"))  
    .to("http://influxdb:8086/write?db=openalt");
```

Influx?



So what?

HOW STANDARDS PROLIFERATE:

(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)

SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.



SOON:

SITUATION:
THERE ARE
15 COMPETING
STANDARDS.

Co ještě vylepšit?

- Přejít pod RTOS (ZephyrOS)
 - Přidání podpory dalších senzorů
 - Přidání dalších protokolů, třeba MQTT pro dlouhodobější experimenty
 - Podpora u něčeho jiného než Wifi
-
- Třeba nějaký manuál? Cvičebnice atd...

Otázky?

Děkuji za pozornost