

**Vysoké učení technické v Brně**  
Brno University of Technology

**Fakulta informačních technologií**

Faculty of Information Technology

## Sbírání dat z akcelerometru(ů), pro určení pozice kvadrokoptéry

SEN - Inteligentní senzory

Autor práce

Ladislav Šulák

Login

xsulak04

Brno 11/2016

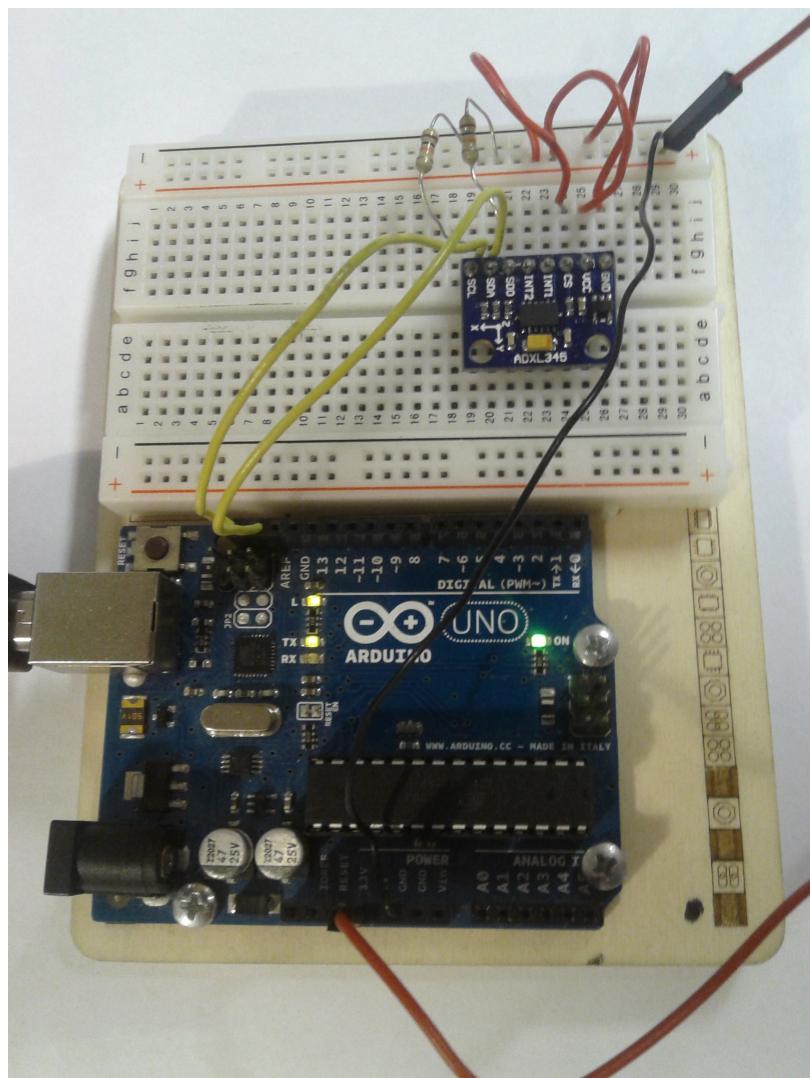
# Úvod

Cieľom bolo vytvoriť riešenie, ktorý bude získavať dátá z akcelerometru pripojeného na platformu Arduino, pričom vo výsledku sa bude zaznamenávať zmena polohy a rýchlosť podľa pohybu kvadrokoptéry, ktorá obsahuje tento obvod.

Akcelerometer, ktorý bol v tejto práci používaný, vracia hodnoty pre všetky 3 osi, má vysoké rozlíšenie (13 bitov) a dokáže merať až do  $\pm 16$  g. Dátá prenáša buď cez rozhranie SPI alebo I<sup>2</sup>C. Obsahuje radu užitočných funkcií, ako napríklad monitorovanie aktivity či neaktivity, detekcia jednoduchých alebo dvojitéh úderov (angl. *single/double tap*), volného pádu a ďalších.

## Návrh a implementácia riešenia

Platforma Arduino komunikuje s počítačom pomocou USB káblu; prepojenie s akcelerometrom je zobrazené na obrázku 1.



Obr. 1: Ukážka zapojenia obvodu. Na obrázku vidieť akcelerometer zapojený do Arduina.

## Nastavenie akcelerometra

Akcelerometer využívame pri rýchlosťi prenosu dát rovnému 9600 baudov cez rozhranie  $I^2C$ . Rozhranie SPI býva spravidla rýchlejšie, no práca s ním je komplikovanejšia. Pre naše účely postačovalo rozhranie  $I^2C$ , medzi jednotlivými skupinami dát boli aj tak vkládané oneskorenia, ako sa odporúča. Čo sa týka rozsahu hodnôt, pracujeme so 16 g, pretože tak je senzor menej citlivý na jemné pohyby. Najmenší rozsah 2 g sa používa pre oveľa citlivejšie zariadenia než je kvadrokoptéra, no pre iné účely, alebo pre prípadné experimentovanie je to veľmi jednoduché zmeniť.

## Softwarová časť riešenia

Návrh aplikácie nevyužíva prístup objektovo-orientovaného programovania. Prvá časť bola vyvíjaná v prostredí Arduino IDE pre Linux, napísaná v jazyku C. Táto hlavná časť, pre samotné Arduino, na začiatku spracúva a normalizuje dátá pomocou počítania posunutí, (angl. *offsets*). Je vhodné, aby počas tejto fázy bol akcelerometer vo vodorovnej polohe a bez pohybu. Hodnota posunutia pre každú os sa vypočíta ako priemer 50 po sebe idúcich hodnôt vynásobený -1. Ten sa následne bude vždy pripočítavať k výstupným hodnotám z akcelerometra. Na základe toho sa ďalej počíta zmena rýchlosťi, pohybu a vzdialenosťi. Dátá sa na výstup generujú približne 2x za sekundu a na každý výsledok je použitých 10 po sebe idúcich hodnôt z akcelerometru. Dátá z Arduina sa posielajú do sériového rozhrania, v našom prípade sa používal operačný systém Ubuntu 16.04 LTS so 64 bitovou architektúrou a komunikačný port sa nachádzal na `/dev/ttyACM0`.

## Spracovanie výstupu

Pre získanie dát zo sériového portu bol implementovaný skript v jazyku Python (konkrétnie verzia 3.5.2), ktorý do výstupného súboru pridáva nejaké dodatočné informácie, ako napríklad čas jednotlivých meraní. Bolo to z dôvodu toho, že nebolo triviálne obdržať dátá zo sériovej komunikácie a priamo ich vypísať napríklad do súboru. Táto funkcia je implementovaná v Arduino IDE pre jazyk Java. Novšia verzia vývojového prostredia Arduino IDE naviac neposkytuje možnosť pracovať len cez príkazový riadok tak, aby bolo možné vykonávať túto činnosť iba zvolením správneho parametru. Ďalšími možnosťami bolo použitie knižníc tretích strán, ktoré dokážu skompilovať a nahrať program do samotného Arduina, avšak ukázalo sa, že mnohé z nich nie sú veľmi udržiavané a kompatibilné s dnešnými technológiami. Alternatívnym riešením by bolo napríklad vypísanie obsahu komunikačného portu pomocou nástrojov prostredia Linux, napríklad *tail* alebo *screen*. Od tejto možnosti sa upustilo, pretože implementovaný skript v jazyku Python je elegantnejším riešením a výstup je tiež v prehľadnejšej podobe, krátká ukážka obsahujúca aj kalibráciu zariadenia:

```
[2016-11-28 05:50:43] ===== Serial Monitor Starts =====
[2016-11-28 05:50:45] Calibration...
[2016-11-28 05:50:50] Axis offsets [mG]:
[2016-11-28 05:50:50] X: 17.27462577 Y: -46.70414352 Z: 858.58233642
[2016-11-28 05:50:58] Speed [m/s]: 0.03 | Distance [m]: 0.00 | Motion vector: (0.01, -0.01, 0.02)
[2016-11-28 05:50:58] Speed [m/s]: 0.16 | Distance [m]: 0.02 | Motion vector: (-0.05, -0.01, -0.01)
[2016-11-28 05:50:59] Speed [m/s]: 0.49 | Distance [m]: 0.05 | Motion vector: (-0.33, 0.07, -0.33)
[2016-11-28 05:50:59] Speed [m/s]: 0.17 | Distance [m]: 0.02 | Motion vector: (-0.16, 0.06, 0.02)
[2016-11-28 05:51:00] Speed [m/s]: 0.09 | Distance [m]: 0.01 | Motion vector: (-0.03, 0.05, 0.04)
[2016-11-28 05:51:00] Speed [m/s]: 0.21 | Distance [m]: 0.02 | Motion vector: (-0.17, 0.07, -0.08)
[2016-11-28 05:51:01] Speed [m/s]: 0.18 | Distance [m]: 0.02 | Motion vector: (-0.14, 0.07, -0.00)
[2016-11-28 05:51:01] Speed [m/s]: 0.15 | Distance [m]: 0.02 | Motion vector: (-0.13, 0.04, -0.04)
[2016-11-28 05:51:02] Speed [m/s]: 0.06 | Distance [m]: 0.01 | Motion vector: (-0.02, 0.04, 0.04)
[2016-11-28 05:51:05] Speed [m/s]: 0.25 | Distance [m]: 0.02 | Motion vector: (-0.23, 0.07, -0.01)
[2016-11-28 05:51:05] Speed [m/s]: 0.06 | Distance [m]: 0.01 | Motion vector: (-0.05, 0.02, 0.02)
[2016-11-28 05:51:08] Speed [m/s]: 0.01 | Distance [m]: 0.00 | Motion vector: (-0.00, -0.00, 0.01)
[2016-11-28 05:51:08] ===== Serial Monitor Ends =====
```

## Použitie

Pre úspešné používanie je potrebné mať nainštalované:

- Arduino IDE<sup>1</sup> (verzia použitá v tejto práci bola 1.6.13)
- Python 3 a knižnicu *pySerial*<sup>2</sup>, dostupnú aj z tzv. *Python Package Index* s verziou 3 (*pip3*)

Po hardwarovej stránke boli použité:

- Arduino UNO<sup>3</sup>
- Akcelerometer ADXL345<sup>4</sup>
- Dva rezistory (10 kOhm), niekoľko vodičov a USB kábel typu AB

Riešenie sa skladá z 2 častí:

- *mainArduino/mainArduino.ino* - kód vykonávajúci spracúvanie dát z akcelerometru, vrátane počítania požadovaných štatistik, hlavne zmien v rýchlosťi a polohe. Tento kód sa nahrá do Arduina a to priebežne vracia výsledky do sériového portu.
- *run\_serialMonitor.py* - získava dáta zo sériového portu a ukladá ich do výstupného súboru spolu s nejakými ďalšími informáciami, ako sú napríklad časové stopy.
- *acceleratorData.log* - ukážka výstupu dát zo sériového portu.

## Záver

Riešenie, ktoré sa podarilo v tejto práci navrhnúť a implementovať je kompatibilné s OS Linux a platformou Arduino UNO s akcelerometrom ADXL345. Výstupom je textový súbor obsahujúci informácie o zmene rýchlosťi, polohy a prejdenej vzdialenosťi vrámci posledného časového úseku. Kvôli veľkým nepresnostiam výpočtov na samotnom akcelerometri a nemožnosti testovať dané metriky na reálnej kvadrokoptére nebolo možné overiť, či výsledné hodnoty zodpovedajú reálnym aj pri použití vyššej rýchlosťi a dlhšej dráhy pohybu.

Pri zadávaní a konzultovaní projektu bol návrh realizovať numerické výpočty pre určenie zmeny polohy a rýchlosťi na základe dvoch akcelerometrov. Po prieskume a experimentovaní sa ukázalo, že je to zbytočné, pomocou jedného akcelerometru sme schopní získať tieto dátu, aj keď s určitými nepresnosťami. Tie sú spôsobené tým, že máme k dispozícii iba určitú diskrétnu podmnožinu z hodnôt, ktoré nameria akcelerometer, čo by sa väčším počtom akcelerometrov úplne nevyriešilo. V praxi sa zvykne akcelerometer kombinovať ešte s gyroskopom, prípadne s GPS modulom.

Čo sa týka ďalšieho rozšírenia, okrem prepojenia obvodu s gyroskopom sa ponúka ešte výpočet rotácie, prípadne detekcia volného pádu alebo jednoduchého, či dvojitého úderu. Posledné dve by ale pravdepodobne neposkytly žiadny relevantný prínos. Taktiež sa nevyužíva detekcia aktivity a inaktivity pohybu. V prípade, že sa akcelerometer nepohybuje,

<sup>1</sup><https://www.arduino.cc/en/Main/Software>

<sup>2</sup><http://pythonhosted.org/pyserial>

<sup>3</sup><http://www.arduino.org/products/boards/arduino-uno>

<sup>4</sup><http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf>

hodnoty nie sú úplne nulové. To je spôsobené rušením a tým, že je to veľmi citlivé zariadenie reagujúce aj na veľmi malé podnety z prostredia. Nastavením určitej prahovej hodnoty uchovanej v registroch akcelerometra je môžné dosiahnuť pri jej neprekročení to, že bude senzor v šetrnom móde, ktorý neodoberá veľa energie. Do normálneho módu sa prepne pomocou prerušenia.