

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Aplikace na bázi akcelerometru MMA8451Q

IMP - Mikroprocesorové a vestavěné systémy

Obsah

1	Úvod	2
2	Vývojový kit FRDM-KL27Z	2
2.1	Deska	2
2.2	Akcelerometr	2
2.3	MMA8451Q	3
3	Program	3
3.1	Komunikace	3
3.2	Uživatelská interakce	4
4	Módy	5
4.1	Signalizace volného pádu kitu	5
4.2	Detekce úhlu náklonu kitu	6
4.3	Krokoměr	7
4.4	Odhad okamžité rychlosti pohybu kitu	9

1 Úvod

Tento projekt zkoumá problematiku práce s akcelerometrem (zde reprezentovaný obvodem MMA8451Q) pomocí vývojové desky FRDM-KL27Z. Program je možno spustit ve více módech činnosti (detekce pádu, natočení, kroků a rychloměr).

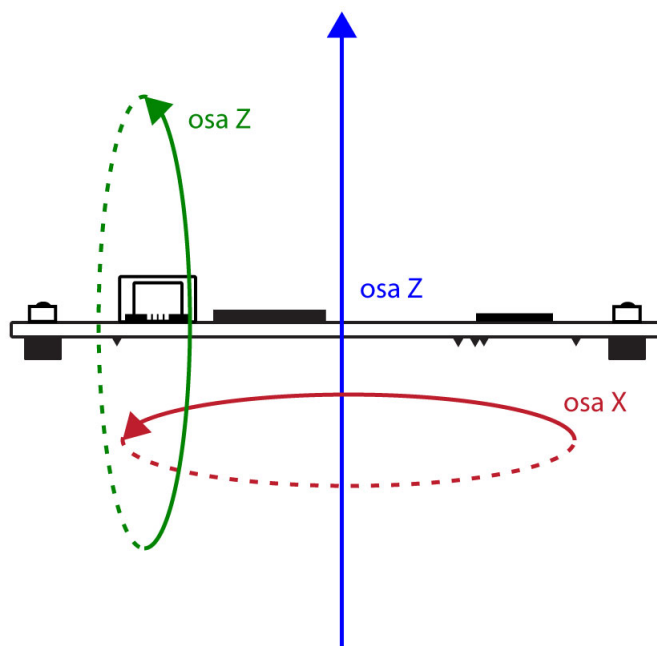
2 Vývojový kit FRDM-KL27Z

2.1 Deska

Srdcem desky je mikrokontrolér typu ARM s jádrem řady ARM Cortex-M4. Samotná deska má zabudovaný programátor, stejně tak jako mnoho dalších obvodů a obslužných prvků, které v našem případě nepoužijeme. Nicméně pro náš program je důležitý výše zmíněný akcelerometr, stejně tak jako dvě tlačítka (označena SW3 a SW1), kdy první slouží jako reset našeho programu a druhé jakožto volič módu. V neposlední řadě je užita RGB LED pro signalizaci zvoleného módu.

2.2 Akcelerometr

Jak již z názvu vypovídá, akcelerometr měří akceleraci (dále zrychlení) v prostoru. Osazený prvek MMA8451Q měří onu veličinu v trojrozměrném prostoru. Osy jsou označeny jako X, Y a Z. Více lze vidět na obrázku níže.



Po vyjasnění si os pohybu (vektorů zrychlení) je nutno tuto veličinu nějak matematicky znázornit. U akcelerometrů existují dva směry:

- vztažení vzhledem k gravitačnímu zrychlení (obecně udávaná konstanta $g_0 = 9.81m/s$, vzhledem poloze na planetě se liší na druhém desetinném místě)
- použití m/s

V neposlední řadě je nutno zavést pojem *klidová poloha*, tj. poloha, kdy na akcelerometr nepůsobí žádné uživatelské zrychlení, nýbrž pouze zrychlení planety v ose Z.

$$\vec{v} = \{a_x, a_y, a_z\} = \{0, 0, 1\} \quad [g] \quad (1)$$

2.3 MMA8451Q

Jedná se integrovaný obvod, realizující funkci akcelerometru. Disponuje až 14-bit vzorkem se snímkovací frekvencí až 800Hz. Stejně tak jako lze nastavit snímkovací frekvenci, je totéž možno udělat s rozlišením. Vzhledem k tomu, že patří do první skupiny akcelerometrů, udává přesnost v jednotkách g. V naše případě zvolíme rozsah $\pm 4g$.

Akcelerometr umožňuje nastavit ještě dva jiné rozsahy; $\pm 2g$ a $\pm 8g$. Námi zvolená prostřední hodnota rozlišení je kompromisem mezi rozsahem ($8g$ umožňuje měření značného zrychlení, ale nabízí pouze 1024 vzorků/g, což by mohl být problém u identifikace polohy) a přesností ($2g$ nabízí výbornou přesnost při 4096 vzorcích/g, avšak pro krokoměry by nemusel rozsah stačit a některé kroky by mohly být zanedbány, či chybně interpretovány).

Při zvoleném rozsahu a 14-bit módu dosahuje akcelerometr těchto parametrů.

- rozlišení - 2048 vzorků/g
- rozsah - $\pm 4g$, tj. ± 8192 vzorků

Programové nastavení akcelerometru probíhá následovně:

```
1 IMP_ComSend(ACCEL_XYZ_DATA_CFG, 0x01);
```

Do registru XYZ_DATA_CFG nastavíme na FS0 a FS1 - pozice od LSB, 0 a 1 - hodnotu 2 v binárním zápisu, čímž zajistíme zvolený rozsah. Více v dokumentaci k akcelerometru.

```
1 IMP_ComSend(ACCEL_CTRL_REG1, 0x0d);
```

Dále nastavíme registr ACCEL_CTRL_REG1, kde je možno pracovat se snímkovací frekvencí, potlačením šumu a rychlostí čtení. Více v dokumentaci k akcelerometru.

3 Program

Pro implementaci byl použit programovací jazyk C a jako vývojové rozhraní MCUXpresso. Program běží jako vestavná aplikace na kontroléru řady ARM.

3.1 Komunikace

Pro komunikaci s čipem je použita sběrnice I2C. Úkolem toho textu není čtenáři podrobně vysvětlovat implementaci sběrnice, nicméně zmíním následující: Jedná se synchronní a obousměrnou přenosovou linku, s více pány (MASTER) a sluhy (SLAVE). Je realizována dvěma vodiči (SCL - hodiny a SDA - data). Pracuje se 7-bitovou adresou, umožňuje tedy připojit až 128 zařízení.

Hlavní smička komunikace je realizováno ve funkci:

```
1 void IMP_ComInit(void);
```

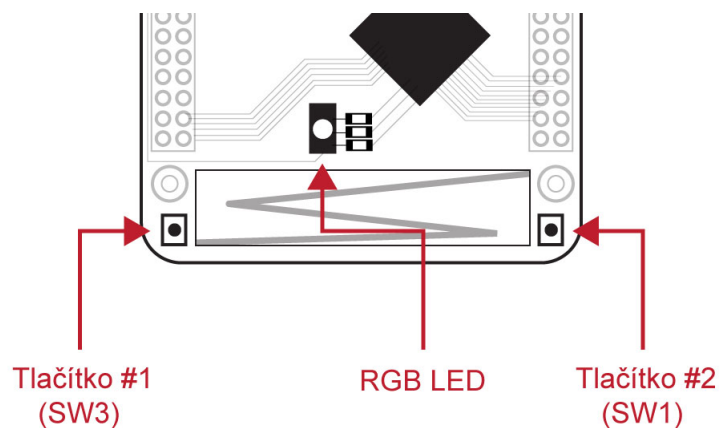
Zde je vytvořen ukazatel na zařízení (akcelerometr se nachází na adrese 0x1D), stejně tak jako je nakonfigurováno a jsou z něj čteny data, která jsou pomocí funkce

```
1 void IMP_BindAccData(int16_t x, int16_t y, int16_t z);
```

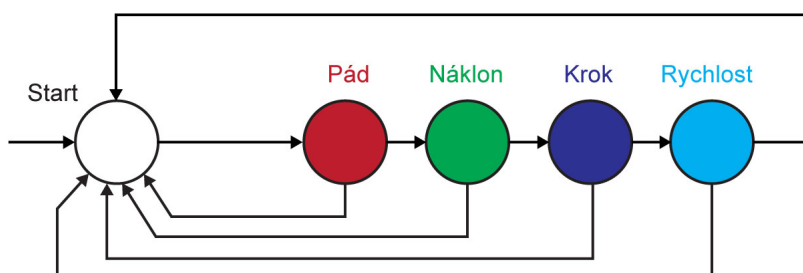
přenesena do části programu IMP_utils, kde probíhá jejich další přesměrování.

3.2 Uživatelská interakce

Uživatel ovládá desku pomocí tlačítek SW1 a SW3 a informace o vybraném módu je mu signalizováno RGB LED. Jednotlivé módy programu vychází ze zadání. V další podsekcích se podrobně probereme její princip, funkci a implementaci. Možné rozšíření aplikace je zobrazovaná data nepřenášet zpět do konzole vyvojového studia, nicméně užít segmentového či dokonce LED displaye.



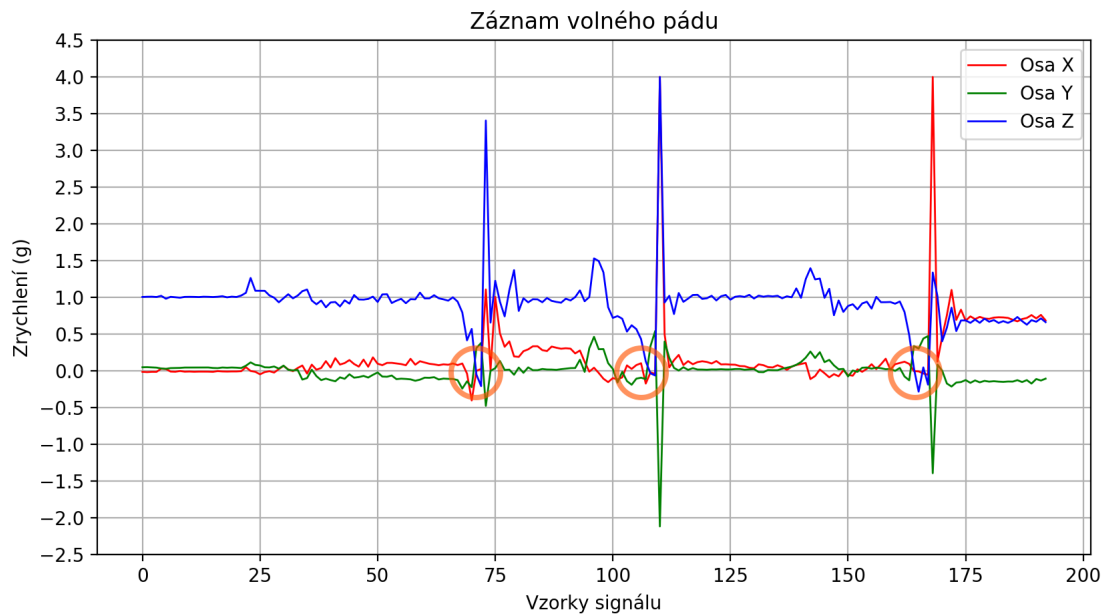
Volba módu je znázorněna konečným stavovým automatem. Barva stavu odpovídá barvě zobrazené na RGB LED při zvoleném stavu. Akcelerometr i při *vypnutí* stále pracuje a čítá.



4 Módy

4.1 Signalizace volného pádu kitu

První z módů je signalizace volného pádu. Princip funkce si vysvětlíme na grafu níže.



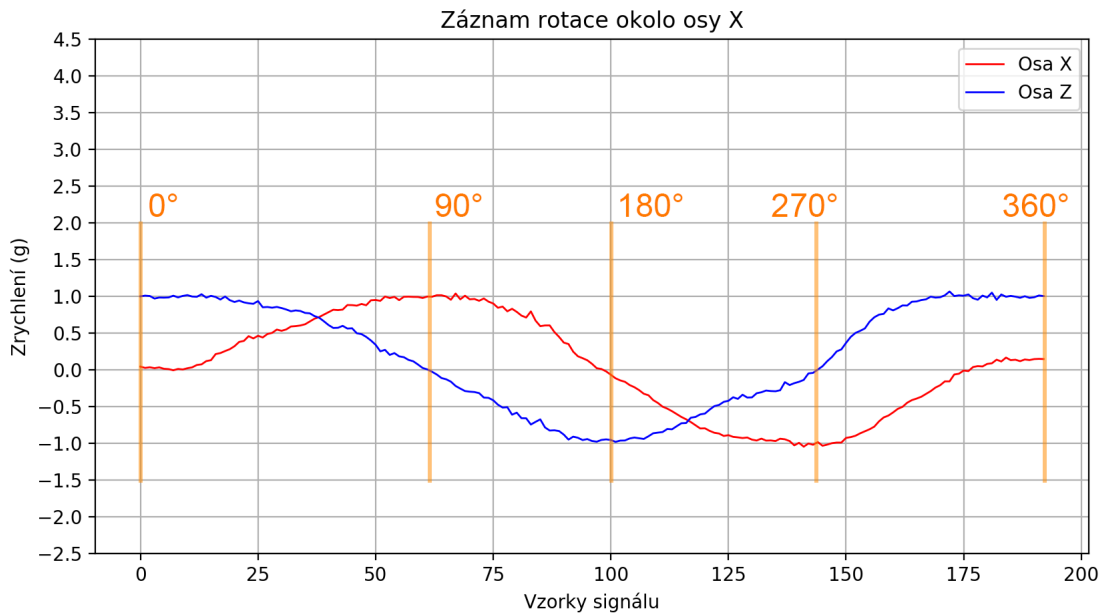
V klidovém stavu akcelerometr neregistruje žádné zrychlení (výjimka je osa Z, zemská gravitace). Fyzikálně je dokázáno, že pokud zrychlení na ose Z klesne po určitou mez - v konjunkci s ostatními osami - jedná se o volný pád (grafu vyznačeno oranžově). V tento moment se minimalizuje zrychlení vůči zemské gravitaci, což je spolehlivý **signál pádu**. Zde je důležité ukázat na důležitost všech tří os, jelikož zařízení nemusí padat tzv. "na placku", ale může být různě natočené. Dále je nutno uvažovat rotační pohyb - např. padající, nerovnoměrně vyvážené těleso bude mít tendenci během letu měnit orientaci na všech třech osách).

Výrobce předepisuje práh volného pádu v intervalu $< \pm 0.1g, \pm 0.5g >$. Experimentálně byla vyčtena hodnota $\pm 0.2g$ jakožto ideální. V programu je následující funkčnost realizována funkcí:

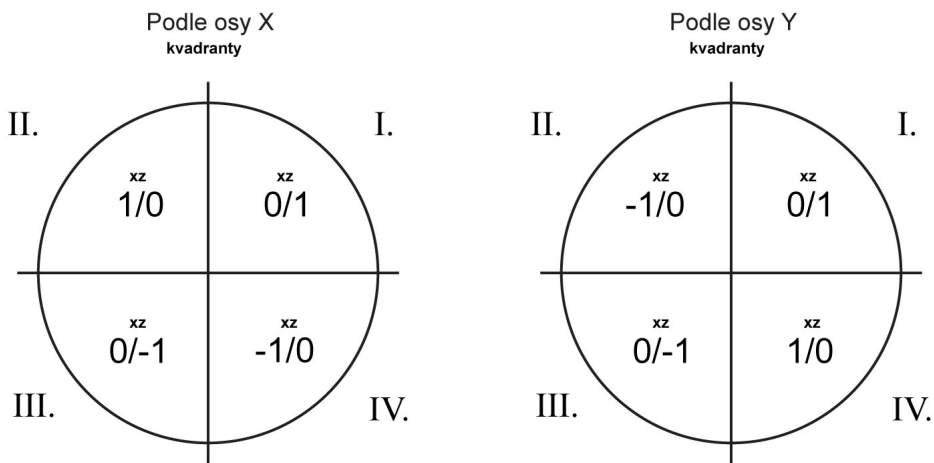
```
1 void IMP_Freefall(int16_t x, int16_t y, int16_t z);
```

4.2 Detekce úhlu náklonu kitu

Náklon desky je odečítán a zobrazován na osách X a Y, jelikož osa Z slouží jako reference pro kvadrant natočení ostatních os. V grafu je znázorněno postupné otočení o 360°. Jsou zde krásně vidět pohyby po jednotlivých osách.



Po zjištění těchto "bodů zlomu" již nebrání vynesení na jednotkou kružnici a následní triviální přepočítání v programu.

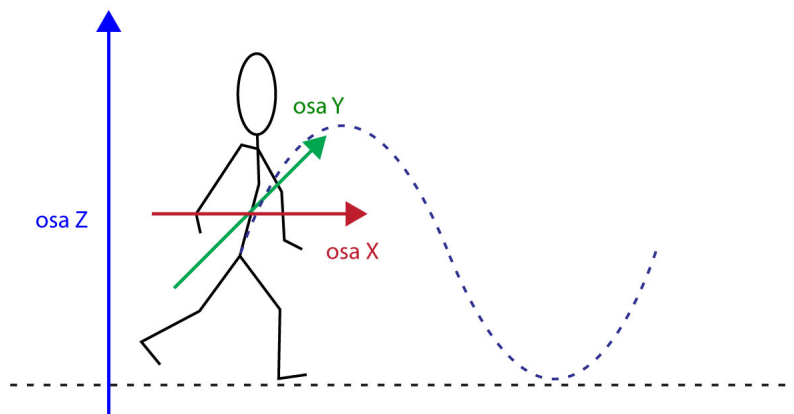


Ten je realizován následnou funkcí.

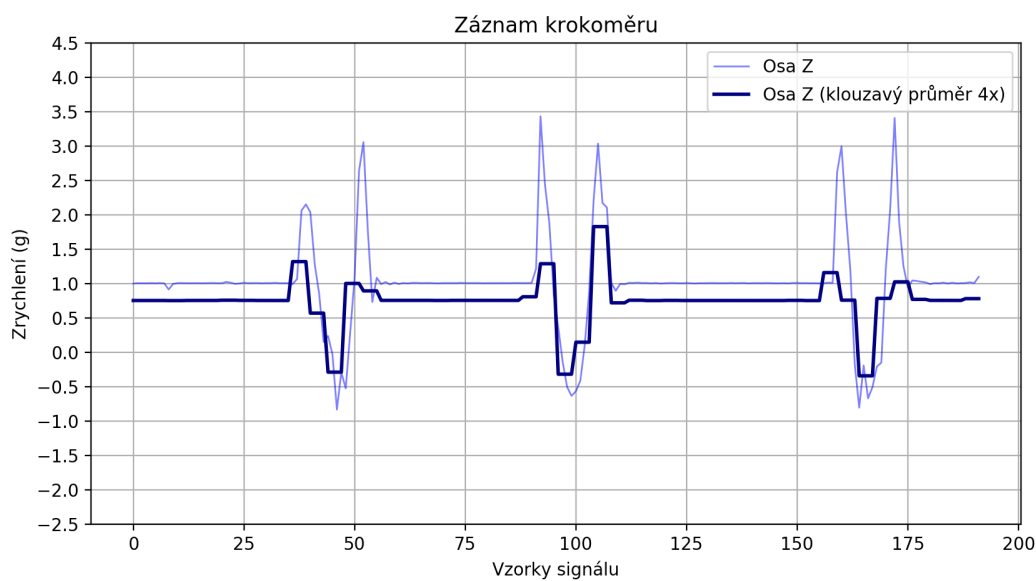
```
1 void IMP_Tilt(int16_t x, int16_t y, int16_t z);
```

4.3 Krokoměr

Tento příklad již není tak triviální jako předchozí dva. Prvně je nutno definovat osy při pohybu.



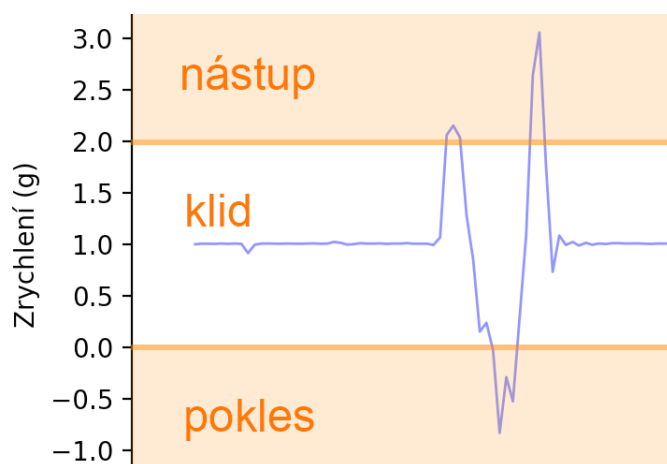
Uvažujeme člověka v pohybu, jenž se při kroku *"houpavě"* pohybuje nahoru a dolů po ose Z. Jde stále stejným směrem, tedy osa X je statická a osa Y svírá s vektorem pohybu příliš velký úhel na zásadní ovlivnění výsledků. Proto ji pro jednoduchost zanedbejme.



V ideální případě by chůze měla být harmonickým signálem, kdy každý krok je reprezentovatelný jednou periodou. Nicméně v praxi tomu tak není. Každý krok za sebou nechá zákmit, jenž je potřeba odfiltrovat nebo alespoň znatelně zmenšit. Po pokusech s subsamplingem byla tato metoda zavržena z důvodu možných falešných a nepřesných detekcí. Zvažována byla také metoda vyhlazení tzv. *klouzavým průměrem*. Nicméně přínosy metody nebyly shledány dostatečné, navíc značně vzrostou nároky na paměť.

Nakonec byl následující postup. Krok byl rozdělen na tři úseky:

- **nástup** - ACC_PEDOMETER_HIGH_THRESHOLD - cca $2g = 4000$
- **pokles** - ACC_PEDOMETER_LOW_THRESHOLD - $0g = 0$
- **klidová poloha** - mezi těmito dvěma hodnotami



Funkce postupně registruje příchozí odečtené hodnoty z osy Z a porovnává je se stavem krokoměru (ACC_PEDOMETER_STILL, ACC_RISING_UP, ACC_RISING_DOWN). Tímto je odstraněna i ona necnost falešného "nástupu" dalšího kroku. Stavový automat čeká na "sestup", nicméně je začat nový krok, čímž je současný zahozen.

Implementace se nachází ve funkci.

```
1 void IMP_Pedometer(int16_t z);
```

4.4 Odhad okamžité rychlosti pohybu kitu

Z fyziky víme, že zrychlení je změna rychlosti objektu za čas. Derivací rychlosti získáme zrychlení, nicméně my potřebujeme opačný výpočet. Proto zrychlení zintegrujeme. Tato operace je však výpočetně značně náročná, proto použijeme numerických metod. Využijeme následující vzorec:

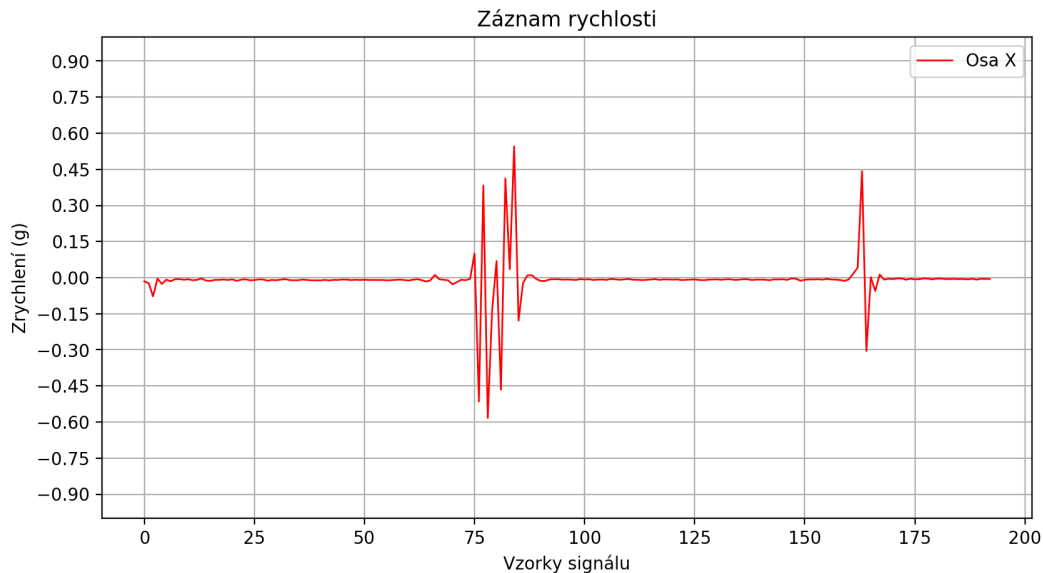
$$v_{cur} = v_{prev} + a_{prev} + \frac{a_{cur} - a_{prev}}{2} * T \quad (2)$$

Výše uvedený vzorec vychází v výpočtu rychlost, který ovšem bere zrychlení jako konstatní, což v našem případě nemusí být pravda.

$$v = v_0 + a * T \quad (3)$$

Tedy místo a dosadí integrál všech jeho částí na intervalu a numericky spočteme.

$$v(t) = \int_{t_0}^{t_1} a \cdot dt \quad (4)$$



Jelikož jsme se omezili pouze na osu X, jakožto zdroj zrychlení pro odečet rychlosti, nemusíme se zabývat normováním vzhledem k gravitačnímu zrychlení, jelikož na akcelerometr v tomto směru nepůsobí. Implementace se opírá o kalibrační rutinu, jenž v klidové poloze normalizuje zrychlení k nule. Komplementární funkčnost je ještě zaručena pomocí filtru nízkých frekvencí.

```
1 /* If below certain threshold, keep as zero */
2 if (cur_acceleration < ACC_VELOCITY_IDLE_THRESHOLD && cur_acceleration > -
    ACC_VELOCITY_IDLE_THRESHOLD) {
3     cur_velocity = 0;
4 }
```

Zajímavá je také funkce, která obchází nemožnost tisku čísel v řadové čárce. Postupným dělením odsekává desetinná místa.

```
1 void IMP_PrintFloat(int16_t data);
```

Nakonec jako obvykle se implementace celé funkčnosti nachází v samostatné funkci.

```
1 void IMP_Speed(int16_t x);
```

Reference

- [1] *Práce s osami akcelerometru* [online]. [cit. 28. prosince 2018]. Dostupné na: <https://physics.stackexchange.com/questions/41653/>.
- [2] *Výpočet rychlosti* [online]. [cit. 28. prosince 2018]. Dostupné na: <https://robotics.stackexchange.com/questions/9988/>.
- [3] *Výpočet rychlosti ze zrychlení* [online]. [cit. 28. prosince 2018]. Dostupné na: <https://physics.stackexchange.com/questions/153159/>.
- [4] *Základní informace o výpočtu rychlosti z akcelerometru* [online]. [cit. 28. prosince 2018]. Dostupné na: <https://physics.stackexchange.com/questions/153159/calculate-speed-from-accelerometer>.
- [5] *Základní princip krokoměru* [online]. [cit. 28. prosince 2018]. Dostupné na: <https://www.analog.com/en/analog-dialogue/articles/pedometer-design-3-axis-digital-acceler.html>.