

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



Inteligentní senzory - projekt
Krokomer

1 Úvod

Zadanie toho projektu je vytvoriť jednoduchý krokomer na ľubovoľnej platforme s využitím akcelerometra.

2 Použité technológie

2.1 NHduino Uno

Ako jadro krokomeru posluží Nhduino Uno. Táto čínska verzia arduina disponuje rovnakým čipom ako originálne arduino uno. Ponúka tak dostatočný výkon za rozumnú cenu.

Špecifikácia a parametre:

1. mikročip - ATmega328P,
2. kryštál - 16MHz,
3. počet digitálnych I/O pinov - 14,
4. počet analógových vstupov - 6.

2.2 Akcelerometer

Použitý senzor pre meranie zrýchlenia je akcelerometer LSM303D od spoločnosti Pololu. Tento senzor kombinuje trojosí akcelerometer a trojosí magnetometer. Je preto vhodný pre použitie aj ako presný kompas. Poskytuje až 6 rôznych údajov o natočení alebo magnetickom poli, ktoré sú dostupné cez IIC alebo SPI rozhranie. Tento malý senzor obsahuje aj regulátor napätia, je tu teda možnosť ho napájať v rozmedzí od 2,5 do 5,5 V. Aj napriek jeho malým rozmerom dokáže merať zrýchlenie v rozmedzí $\pm 16 \text{ g}$, a magnetickú indukciu v rozmedzí $\pm 12 \text{ gauss}$.

2.3 Liquid crystal

Pre zobrazovanie počtu krokov je použitý 4-riadkový liquid crystal displej. Pre zjednodušenie kabeláže je na displeji pripojené IIC rozhranie ktorý pôvodný počet 16 pinov redukuje na 4 a to: VIN, GND, SDA a SCL.

2.4 Power banka

Power banka posluží ako zdroj napájania pre všetky komponenty. Jej výstup 5 voltov je ideálny pre Nhduino. 1500MAh úplne postačí na prevádzku krokometra.

3 Zapojenie periférií

Keďže arduino poskytuje aj 5V výstup aj 3.3V výstup. Jeden sa použije na napájanie displeja a druhý pre akcelerometer. Displej komunikuje pomocou IIC zbernice, knižnica ktorá poskytuje rôzne funkcie používa pin A4 ako SCL a pin A5 ako SDA, ďalej ho pripojíme na 5V výstup a zem.

Akcelerometer pripojíme na 3.3V výstup, zem, SCL a SDA piny na arduino. Arduino IDE poskytuje knižnicu aj pre použitý akcelerometer takže nastavovanie a inicializácia pinov nie je potrebná. Pre prepojenie použijeme nepájivé pole.

4 Návrh

Existujú minimálne 3 algoritmy ktorými sa dá zisťovať počet krokov z údajov poskytnutými akcelerometrom.

Prvým z nich a zároveň najjednoduchším je Pytagorasin teorém pre výpočet veľkosti akceleračného vektora. Podľa vzorca sa vypočíta vektor, ak presiahne určitú hodnotu, dá sa považovať za krok. Tento algoritmus je jeden z najpoužívanejších.

Ďalším je analýza signálu pomocou Fourierovej transformácie. Táto možnosť však vyžaduje veľké množstvo testovacích dát a zároveň sa implementácia takéhoto algoritmu nemusí vojsť do malej pamäte NHduina.

Najideálnejším spôsobom na detekciu krokov by bola neurónová sieť. Pre túto variantu by nám postačila neurónová sieť o troch vrstvách. Mala by 3 vstupy ktoré by predstavovali súradnice x,y,z dostupné z akcelerometra a jeden výstup ktorý by nadobúdval hodnoty 0 alebo 1, podľa toho či nastal krok. Táto varianta algoritmu taktiež nie je vôbec triviálna. Je potrebné mať veľké množstvo testovacích dát, na to aby sa sieť dostatočne dobre naučila čo je krok. Ďalším problémom môže byť náročnosť na veľkosť programu, aj v tomto prípade sa neuronová sieť len s ťažkosťami vojde do pamäte.

5 Implementácia

Medzi použité knižnice patria `LSM303.h` pre akcelerometer a `LiquidCrystal_I2C.h` pre displej. Výstupy akcelerometra dostupné pomocou knižnice sú získavané spojením dvoch 8-bitových registrov akcelerometra pre každú os. Tieto hodnoty sú používané pre získanie hodnôt gravitačného zrýchlenia. Dáta poskytnuté akcelerometrom sú 16-bitové hodnoty. Pre použitie akcelerometra v rozmedzí $\pm 2 \text{ g}$ je potrebné vydeliť túto hodnotu číslom 16. Použijeme teda bitový posun do prava o 4 bity. Pre príklad načítame číslo z registra osi -16144, bitovým posunom získame číslo -1009 čo odpovedá gravitačnému zrýchleniu 1.009 g .

Ako prvé v programe prebehne inicializácia kompasu a displeja. Ďalej kalibrácia akcelerometra kde sa získa 10 hodnôt z polohy akcelerometra a urobí sa ich priemer. Zabezpečí to určenie si počiatočnej polohy od ktorej bude výpočet kroku prebiehať. Následne sa vypočíta veľkosť vektora podľa vzorca:

$$v = \sqrt{x^2 + y^2 + z^2}$$

ak veľkosť vektora prekročí hodnotu $0,21 \text{ g}$ bude sa takéto zrýchlenie počítat za 1 krok. Nasleduje čakanie 500 mili sekúnd aby nedošlo hneď k výpočtu ďalšieho kroku. Hodnota $0,21 \text{ g}$ bola vybraná experimentálne viacerými pokusmi ak by bola hodnota menšia, ako krok by sa počítal aj najmenší pohyb akcelerometra, naopak ak by bola hodnota príliš vysoká kroky by sa vôbec nedetekovali, nakoľko by bol potrebný veľmi rýchly pohyb akcelerometra. Zároveň po každom kroku prebieha kalibrácia aby sa predošlo nekonečnému započítavaniu krokov pri natočení senzoru.

6 Testy

Testovanie aplikácie prebiehalo meraním reálneho počtu krokov, meraním pomocou komerčnej aplikácie na android - *Pedometer step counter* a meraním pomocou implementovaného krokometra. Ako je vidieť z údajov v tabuľke, počty reálnych krokov a počty krokov nameraním krokometra sa príliš nelýšia. Pri väčšej vzdialenosti je však počet nameraných krokov väčší ako počet reálnych krokov. Je toho tak aj u komerčného krokometra.

Tabuľka 1: Testy

Reálny počet krokov	Pedometer aplikácia	Moja aplikácia
15	15	15
50	50	51
100	101	103

7 Záver

Implementácia krokometra sa podarila, a namerané výsledky sa približne zhodujú aj s komerčným krokometrom. U implementovaného krokometra však nastáva aj problém súvisiaci s použitým algoritmom kedy sa za krok počíta aj otočenie krokometra okolo niektorej z jeho osí. Riešením tohoto problému by bol výber iného algoritmu, ktorého implementácia je však pomerne zložitá.