

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor SOČ: 18. Informatika

Ergonomie práce na počítači

Lukáš Forst

Kraj: Plzeňský

Domažlice 2016

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor SOČ: 18. Informatika

Ergonomie práce na počítači Computer ergonomics

Autor: Lukáš Forst
Škola: Gymnázium J.Š.Baara, Domažlice,
Pivovarská 323
Kraj: Plzeňský

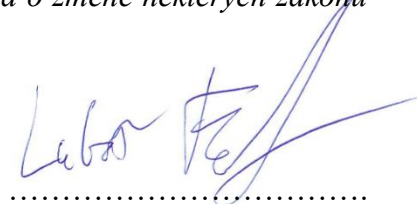
Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval samostatně a použil jsem pouze podklad (literaturu, projekty, SW atd.) uvedené v seznamu vloženém v práci SOČ.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupnění této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V*Mrákově*....., dne *9. 3. 2016*.....


.....

Poděkování

Děkuji Ing. Josefu Lípovi za původní myšlenku a za věcné připomínky během tvorby práce.

Děkuji Alžbětě Johánkové za korekturu celé práce.

Anotace

Práce se zabývá vývojem a popisem fungování přístroje na detekci nesprávného držení těla při práci na počítači. Celý projekt má dvě části - softwarovou a hardwarovou. Pro realizaci hardwarové části projektu byla použita open-source platforma Arduino. Software, který se stará o detekci, jsem si programoval sám a dělí se na dvě části – řídicí kód pro Arduino a C# aplikaci pro stolní počítač.

Výsledkem celé práce je funkční zařízení, které ve spojení s programem dokáže detekovat krčení uživatele v reálném čase, nebo zaznamenávat data o sezení v závislosti na čase.

Klíčová slova: Arduino; zdravé sezení; ergonomie práce u počítače

Annotation

The study deals with development and functional description of a device detecting bad posture when working at a computer. The project consists of software and hardware part. Hardware part of the project was realized using open-source platform Arduino. The software, which detects the posture, was written by me and it has two parts – Arduino control code and C# application for desktop computer.

The result of the work is a functional device which together with the program can detect bad posture of a person in real time or record the posture data over time.

Key words: Arduino; healthy sitting; computer ergonomics

Obsah

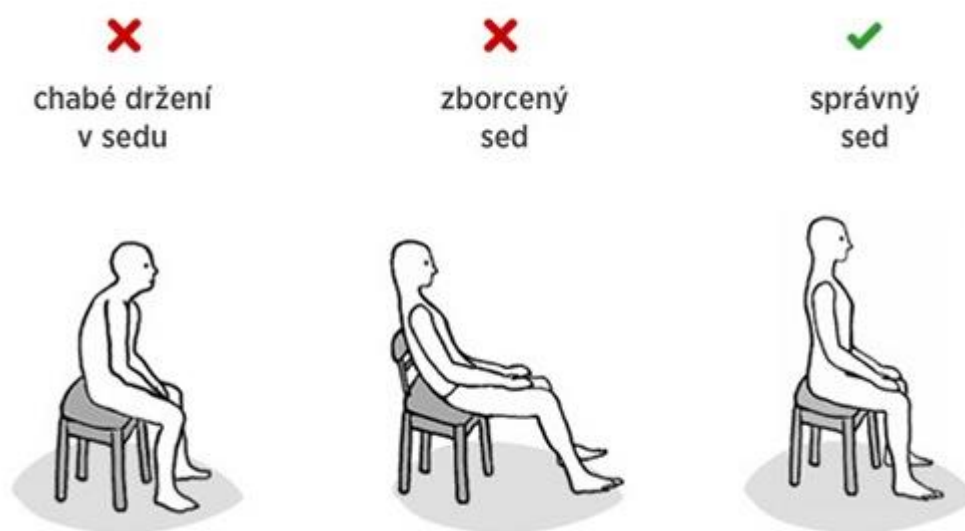
1. Úvod.....	6
2. Zařízení – hardware	7
2.1 Arduino Nano	7
2.2 MPU-6050	8
2.3 Verze 1.0.....	8
2.4 Verze 2.0.....	10
2.5 Verze 3.0 (budoucnost).....	11
3. Software	12
3.1 Řídící program Arduina (jazyk Wiring)	12
3.1.1 Verze 1.0 řídicího programu.....	12
3.1.2 Verze 2.0 řídicího programu.....	13
3.2 Program arduino_ext.exe (jazyk C#).....	13
3.3 Program data_miner.exe	16
3.4 Program simplified_checker.exe	16
4. Vyhodnocení uložených dat.....	17
5. Průzkum potencionálního trhu.....	19
6. Závěr	20
7. Dodatky.....	21
7.1 Data z průzkumu.....	21
7.2 Seznam užitečných odkazů.....	22
8. Seznam obrázků.....	23
9. Seznam zdrojů a užití literatury.....	23

1. Úvod

Život 21. století se pomalu, ale jistě přesouvá na pracovní židli u počítače. S tím se pojí některá zdravotní rizika, konkrétně poškození očí a deformace páteře nesprávným držením těla při sezení u PC. Vzhledem k tomu, že bolesti zad postihují většinu populace (zdroj: [5]) jsem se rozhodl, že se budu věnovat právě jim. Zaměřil jsem se na to, jak se tělo chová při dlouhodobějším sezením u počítače.

Cílem mojí práce bylo tedy přijít na způsob, jakým zjistit, že uživatel počítače nesedí správně a následně pak napsat program pro počítač, který bude v reálném čase vyhodnocovat data generovaná senzory a v případě detekce nesprávného sezení uživatele na tuto skutečnost upozorní.

Na začátek jsem potřeboval zjistit, jak se lidé vlastně při sezení na židli krčí. Z hledání nakonec vyšly dva hlavní způsoby nesprávného sedu na židli, a to „chabé držení v sedu“ a „zborcený sed“.



obr. 1 – Různé druhy sezení na židli, zdroj: http://i.idnes.cz/11/053/cl5/MBB3b50f0_sezeni.jpg

2. Zařízení – hardware

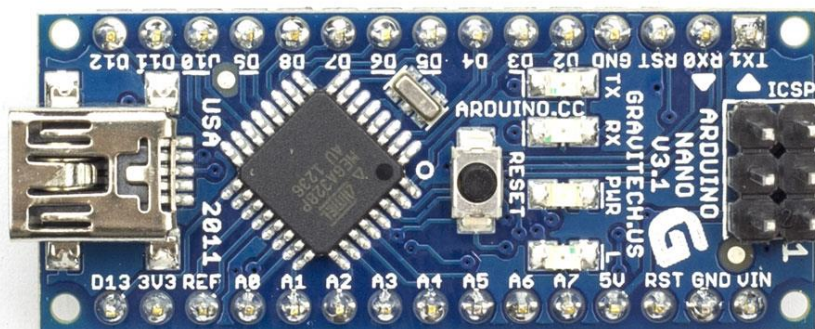
K sestavení hardwarové části projektu jsem použil platformu Arduino. Velkou výhodou této open-source platformy je nákupní cena dílů, která se pohybuje na zahraničních aukčních portálech v řádech desítek korun. Další výhodou je nesmírné množství různých návodů na internetu, které jsem mohl použít.

Aktuální verze zařízení (v. 2.0), která je v provozu (ještě ale není finální), se skládá ze senzorů MPU-6050, Arduina Nano a drátové sběrnice. Vše je propojeno a připevněno na triko, takže si uživatel nemusí lepit senzory přímo na tělo.

Protože existuje více druhů nesprávného sezení na židli a pouze jeden správný, musel jsem použít dva senzory MPU-6050. Dal jsem si za cíl detekovat dva nejčastější styly špatného sezení na židli, a to již výše zmíněné „chabé držení v sedu“ a „zborcený sed“. Jeden senzor je umístěn na šíji na krku a druhý v kříži na zádech.

2.1 Arduino Nano

Arduino je open-source platforma pro snadný návrh a vývoj elektronických programovatelných zařízení. Arduino Nano je minimalizovaná vývojová deska s mikrokontrolérem AVR ATmega328.



obr. 2 – Arduino Nano, zdroj: https://www.arduino.cc/en/uploads/Main/ArduinoNanoFront_3_lg.jpg

2.2 MPU-6050

MPU-6050 obsahuje akcelerometr a gyroskop na jednom čipu. Dá se použít jako tři nebo šestiosé gyro, já používám primárně pouze jednu osu náklonu. Tento senzor jsem použil kvůli jeho nízké ceně, která se pohybuje v přepočtu kolem 50 Kč a také kvůli již existujícím knihovnám, které jsou volně dostupné na internetu.

Díky knihovnám jsem schopen číst data velice rychle a nemusím je získávat přímo z adresované části paměti přímo na senzoru. Knihovny mi umožňují převádět surová data např. na úhly nebo na úroveň natočení (yaw, pitch, roll)¹. Já jsem se však rozhodl použít surová data, protože jsem nechtěl, aby se senzory při měření zahřívaly (knihovny aktivují DMP² v čipu, což způsobuje, že se čipy zahřívají) a znepríjemňovaly tím nošení zařízení.

Senzory jsou zapojeny v jedné SDA – SCL lince, která slouží pro komunikaci pomocí knihovny pro sběrnici I²C.



obr. 3 – MPU-6050, zdroj: <http://playground.arduino.cc/uploads/Main/mpu-6050.jpg>

2.3 Verze 1.0

Na první verzi hardwaru jsem použil Arduino Mega 2560 Rev. 3, a to hlavně kvůli velké paměti (nemusel jsem mít tak dobře optimalizovaný kód) a mnoha různým portům, takže jsem mohl zvážit, jaká sestava senzorů bude nejlepší a jaké bude konečné zapojení všech modulů.

Pro detekci zakřivení zad uživatele jsem se rozhodl použít dva senzory MPU-6050. V první verzi byl také přítomen slot pro SD kartu, na kterou se zapisovala data, která byla po připojení k PC a k příslušnému programu *arduino_ext.exe*³ vyhodnocena v programu

¹ Více o tomto tématu na - https://en.wikipedia.org/wiki/Aircraft_principal_axes

² Digital Motion Processor – provádí výpočty s hodnotami senzorů, je umístěn přímo na čipu MPU-6050

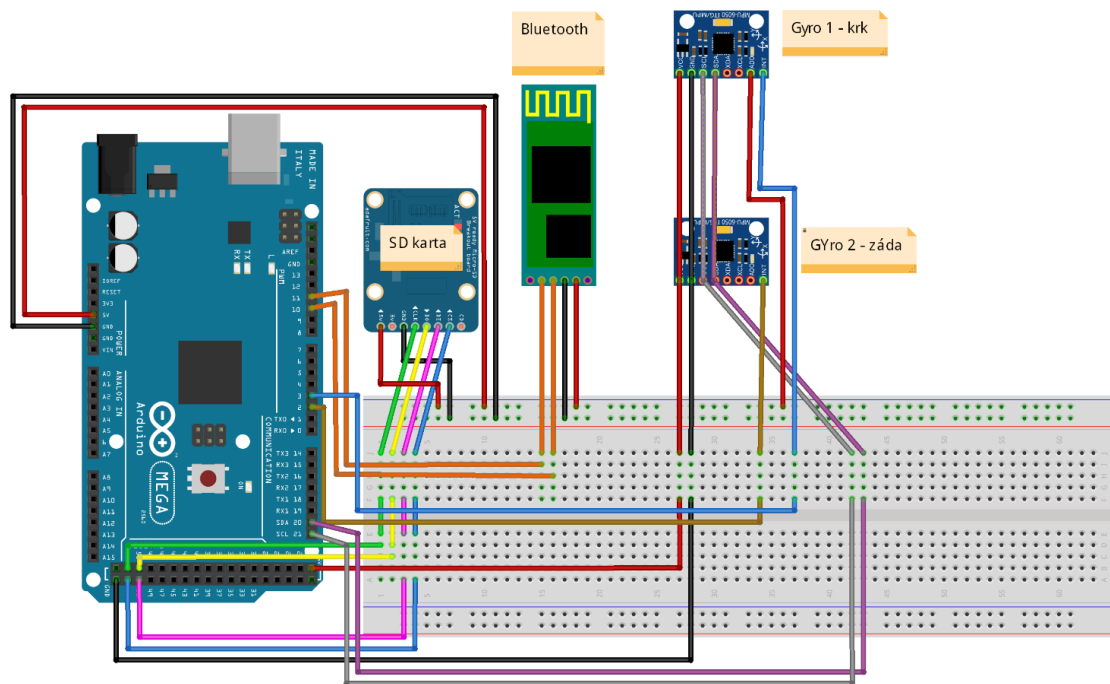
³ Mnou napsaný program pro extrakci dat, více o něm v kapitole Software.

Microsoft Excel. Poslední součástí byl bluetooth modul JY-MCU, který sloužil pro bezdrátovou komunikaci s PC nebo s mobilním telefonem.

První verze hardwaru vyžadovala přímý kontakt s tělem uživatele – senzory byly nalepeny přímo na tělo, což se ukázalo vysoce nepraktické a nepříjemné, protože pájená místa řezala do kůže a trvalo dlouho, než se senzory povedlo dát přesně na místa, kde by měly být. Dalším problémem byly dráty, které vedly ze sensorů. Na tělo se také musela připevnit celá sběrnice, která šla z drátů ven. Ta byla bohužel příliš velká na to, aby to bylo pohodlné. Tyto problémy mě donutily uvažovat trochu jinak a navrhnout verzi 2.0.



obr. 4 – umístění sensorů na tělo, v. 1.0

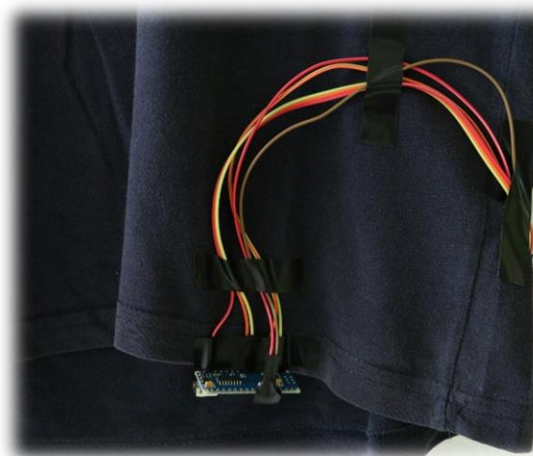


fritzing

obr. 5 – schéma zapojení verze 1.0 v open-source programu Fritzing

2.4 Verze 2.0

Současná verze, jejíž vývoj je dokončen, je verze 2.0. U té došlo k výraznému zmenšení celého zapojení a k minimalizaci drátů vedoucích od těla do počítače. Došlo zde ke kompletnímu přepracování návrhu a použití jiných součástek. Verze 2.0 již neobsahuje bluetooth, který byl vyřazen, protože použití vysokofrekvenčního zářiče přímo na těle by mohlo mít zdraví škodlivé následky. Taktéž již neobsahuje slot pro SD kartu, protože záznam dat je řešen přes stále běžící program v počítači. Negativem tohoto kroku je, že zařízení musí být stále připojeno k počítači, což je řešeno přes univerzální Mini USB kabel, který propojuje Arduino Nano a počítač. Protože jsem použil Arduino Nano, musel jsem optimalizovat kód původně napsaný pro Arduino Mega, které má větší paměť a jinak umístěné různé porty. Díky použití surových dat z MPU-6050 hned od začátku jsem nemusel nijak zasahovat do algoritmu vyhodnocování měření, protože nebylo nutné použít dodatečné knihovny, které zabírají v paměti hodně místa. Senzory se již nemusí lepit přímo na tělo, ale jsou připevněny na triko, díky čemuž se celé zařízení stává snadno použitelné a odpadá nepříjemnosti s umístěním senzorů a nepohodlnosti celého zařízení.



obr. 6.1 a 6.2 – verze 2.0 již napojená na triko

2.5 Verze 3.0 (budoucnost)

Toto zařízení je zatím ve fázi vývoje, kdy už mám vše předpřipravené a navrhnuté, pouze čekám na to, až si budu moci objednat některé ne úplně typické součástky, které jsou hůře k sehnání. Cílem této verze je dostat se až na hranici možné minimalizace. Pak by se zařízení mělo stát v podstatě neobtěžující „wearables“⁴.

Toho chci dosáhnout použitím, mimo jiné, elektro-vodivé nitě, která poslouží místo drátů. Do trika bude celá síť drátů všita a nebude uživatelsky přístupná, tím se celé zařízení razantně zmenší a zlehčí, protože dráty jsou na aktuálních verzích to největší a nejméně pohodlnější. Triko bude nahrazeno velice přiléhavým funkčním prádlem, které bude senzorům poskytovat stabilitu a umístění senzorů bude díky němu stálé (nyní se kvůli pohybům trika mění). Sensory zůstanou stejné, jen se místo připájení nožiček pro dráty připele nit do oček. Arduino Nano zůstane také, protože se lepší alternativa nenabízí, ale bude všito do lemování na spodní hraně trička, což mu poskytne alespoň minimální ochranu a také ho zakryje, takže bude vidět pouze mini USB port. Ten je nutností, protože pro bezdrátový přenos bych potřeboval vysokofrekvenční zářič, který by mohl při takhle malé vzdálenosti od těla způsobit zdravotní komplikace.

⁴ wearables – nositelná elektronika, miniaturizovaná elektronická zařízení

3. Software

Existují dva druhy software, které jsem musel vytvořit, abych mohl data zachytit a následně v počítači vyhodnotit. První běží v počítači a stará se o vyhodnocení dat. Druhý je řídicí kód pro Arduino, který obstarává jejich záznam.

3.1 Řídicí program Arduina (jazyk Wiring)

V prvotních fázích projektu byl vyvíjen pouze jeden řídicí kód pro Arduino Mega. Když jsem ale přešel na verzi 2.0, musel jsem přepracovat celý kód tak, aby vyhovoval i Arduinu Nano. Pro mikrokontroléry Arduino jsem programoval v jazyce Wiring (derivát C++). Jako programovací prostředí posloužilo Visual Studio 2015 Community Edition s doplňkem Visual Micro.

3.1.1 Verze 1.0 řídicího programu

Hlavní funkcí řídicího programu verze 1.0 bylo získávat data ze senzorů a zapisovat je na SD kartu. Zároveň také program dokázal komunikovat s počítačem a vykonávat příkazy zaslané přes sériový port. Příklady příkazů, na které Arduino Mega reaguje (vysílá data přes ComPort⁵ do počítače):

- *stav* – vyšle strukturovaná data aktuálního natočení senzorů
- *log1* nebo *log2* – vyšle uložená data pro každý senzor zvlášť
- *zapis* – zapíše data ze senzorů jako další hodnotu na SD kartu
- *getx* – příkaz používaný programem *arduino_ext.exe* pro vyhodnocování náklonu v reálném čase, vyšle pouze hodnotu pro náklon v jedné ose z obou senzorů
- *console* – používáno programem *data_miner.exe* pro zápis všech dat do PC, vyšle strukturovaná data náklonu senzorů
- *wholelog* – používáno programem *arduino_ext.exe* pro získání kompletního logu z SD karty pro oba dva senzory najednou
- *setit* – nastaví hodnoty, kdy člověk sedí rovně; kalibrace senzorů
- *krceni* – podle hodnot, kdy uživatel sedí rovně, vyhodnotí, zda sedí rovně i teď.

⁵ComPort - počítačový sériový port (například USB)

3.1.2 Verze 2.0 řídicího programu

Vzhledem k tomu, že v. 2.0 nepoužívá SD kartu a ani bluetooth modul, jsem byl nucen napsat pomocný program běžící v počítači, který data ze senzorů zaznamenává přímo do PC (*data_miner.exe*). Řídicí kód Arduina je funkčně podobný tomu pro verzi 1.0, pouze již nezapisuje data na SD kartu, ale posílá je rovnou do počítače. Opět reaguje na zasláné příkazy přes konzoli.

- *stav* – vyšle strukturovaná data aktuální polohy senzorů
- *getx* – příkaz používaný programem *arduino_ext.exe* pro vyhodnocování náklonu v reálném čase, vyšle pouze hodnotu pro náklon v jedné ose z obou senzorů
- *console* – používáno programem *data_miner.exe* pro zápis všech dat do PC, vyšle strukturovaná data náklonu senzorů
- *setit* – nastaví hodnoty, kdy člověk sedí rovně; kalibrace senzorů

K odstranění SD karty došlo z toho důvodu, že Arduino Nano má paměť pouze 30 720 bytů a celý kód z verze 1.0 by se do něj nevešel, protože zabírá 34 640 bytů paměti (celková kapacita paměti Arduina Mega je 253 952 bytů). Do těchto čísel jsou započítány i použité knihovny. Pokud bych nepracoval se surovými daty z MPU-6050, dostal bych se na vyšší hodnoty, musel bych totiž použít další knihovny.

3.2 Program *arduino_ext.exe* (jazyk C#)

Hlavní aplikace je naprogramována v jazyce C#. Snažil jsem se ji napsat univerzálně tak, aby mohla bez problémů fungovat na jakémkoliv počítači s MS Windows. Toho jsem docílil využitím komponent a možností jazyka C#, jako jsou např.

- Save dialogy – třída v C# využívaná pro nalezení cesty k souborům; uživatel může pomocí GUI⁶ najít cestu, kam chce daný soubor uložit
- dočasné složky – pro dočasné uložení dat program používá proprietární složky poskytované systémem MS Windows
- seznam dostupných sériových portů – uživatel zvolí ten, ke kterému je připojeno Arduino

Aplikace umožňuje volbu mezi dvěma verzemi hardwaru a podle zvolené verze Arduina si sama vypíná nebo zapíná některé své součásti. Pokud tedy uživatel zvolí verzi 2.0, nebude mu automaticky zpřístupněna možnost získání dat z SD karty, protože tu totiž verze 2.0 neobsahuje.

⁶ GUI – *Graphical User Interface* – Grafické Uživatelské Rozhraní, slouží k interakci s uživatelem

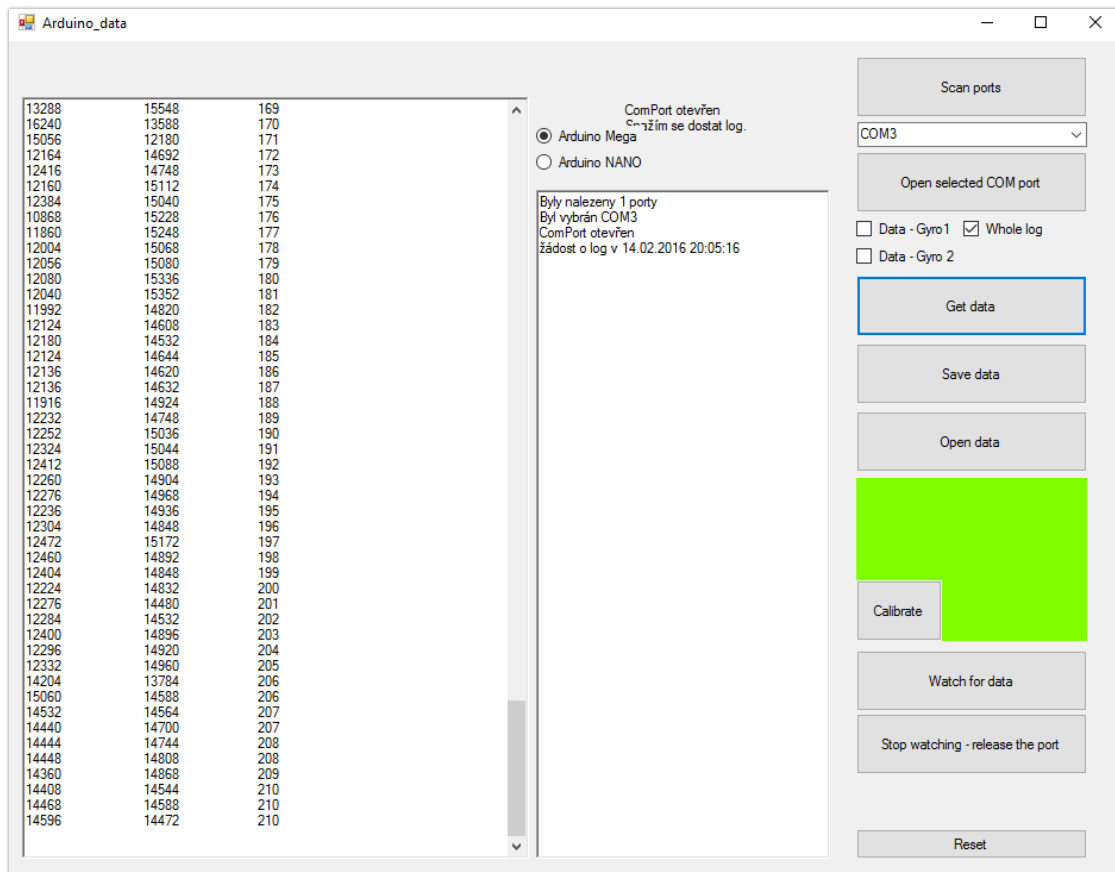
Program slouží nejen pro extrahování dat z SD karty ve verzi 1.0, ale i pro vyhodnocování náklonu senzorů v reálném čase. Tato možnost je přístupná v obou hardwarových verzích. Program využívá příkazů popsaných v předchozích kapitolách. Vzhledem k tomu, že získává velké množství dat z SD karty a zároveň by měl tyto data zobrazovat v uživatelském prostředí, musel jsem celou práci programu rozdělit mezi několik vláken, která mezi sebou musí spolupracovat. Primární vlákno má na starost GUI a interakci s uživatelem. Do programu jsem implementoval ještě další dvě vlákna. První je spuštěno ve chvíli, kdy uživatel chce dostat data z SD karty, spouští ho tlačítko „Get data“. Toto vlákno zašle přes sériový port příkaz *wholelog* a čeká na příchozí data, ta následně předává primárnímu vláknu, které je uživateli zobrazuje v reálném čase tak, jak proudí do PC.

Zde nastal můj největší problém. Nevěděl jsem totiž, jak mám data předat primárnímu vláknu, protože kvůli omezení v .NETu jiné než primární vlákno nemůže zasahovat do uživatelského prostředí. Nemohl jsem tudíž zobrazit přijatá data z jiného vlákna v primárním vlákne, a tím pádem k nim neměl uživatel přístup. Vyřešit se to podařilo až pomocí dalších pomocných tříd C#. V aktuální verzi programu už vše funguje bez jediného problému.

Třetí vlákno slouží k hlídání náklonu v reálném čase, spouští se tlačítkem „Watch for data“ a pravidelně posílá do Arduina příkaz *getx*. Přijatá data následně vyhodnocuje, a pokud se uživatel hrbí, čtverec v pravém dolním rohu, který je normálně zelený, se zbarví rudě. Po narovnání uživatele se opět zbarví do zelena.

Indikaci barvou jsem zvolil díky její srozumitelnosti a aktuálně slouží pouze jako ukázka toho, že program zvládá vyhodnocovat data v reálném čase. Oba dva senzory jsou vyhodnocovány najednou. Můžu je sice nechat vyhodnocovat zvlášť, ale to je zbytečné, protože mě v podstatě zajímají jen dva stavy, které nám říkají, jestli se uživatel hrbí nebo ne.

Před samotným použitím módu hlídání se senzory musí nakalibrovat. To probíhá po stisknutí tlačítka „Calibrate“. Uživatel se musí posadit rovně a hodnota ze senzorů je zapsána do proměnných v programu. Pomocí těchto proměnných jsou pak vyhodnocovány proudící data z Arduina.

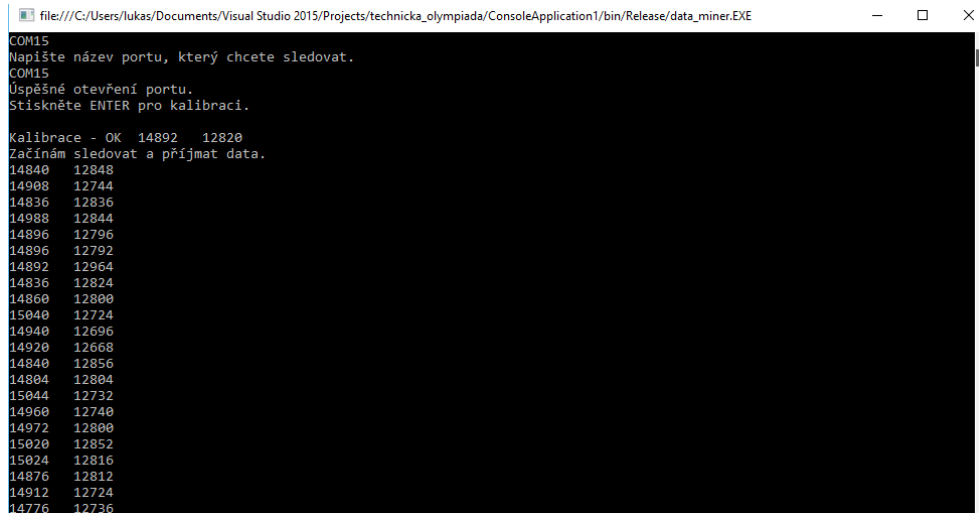


obr. 7 – GUI programu arduino_ext.exe

- *Scan ports* – zjistí dostupné sériové porty a vypíše jejich seznam do připraveného seznamu, ze kterého uživatel zvolí ten, na kterém je připojeno Arduino
- *Open selected COM port* – otevře se vybraný port pro komunikaci
- *Data Gyro 1 / Data Gyro 2 / Whole log* – uživatel vybere data, která chce z SD karty vyextrahovat
- *Get data* – spustí nové vlákno, které přenáší data z SD karty do PC a zobrazuje je v GUI
- *Save data* – vyvolá Save dialog, pomocí kterého uživatel vybere cestu a uloží data, cesta se navíc zapíše do proměnné, ze které ji může program znovu vyvolat
- *Open data* – otevře uložená data v programu MS Excel (pokud není program nainstalován, otevře je v bloku); cesta se volit nemusí, vyvolá se z proměnné
- *Calibrate* – kalibrace senzorů pro mód hlídání
- *Watch for data* – spustí nové vlákno, které hlídá, zda se uživatel nehrbí
- *Stop watching - release port* – vypne mód hlídání a zavře port
- *Reset* – restartuje kompletně celé GUI, zavře port a vyčistí proměnné pro data

3.3 Program data_miner.exe

Jedná se o konzolovou aplikaci pro sběr dat z Nano-verze. Při spuštění programu stačí nakalibrovat senzory a vybrat port pro přenos dat. Dál už aplikace sama sbírá a zároveň zapisuje data do textového souboru na disku počítače. Data jsou samozřejmě dobře formátována, takže je lze nechat vyhodnotit úplně stejně jako data z SD karty.

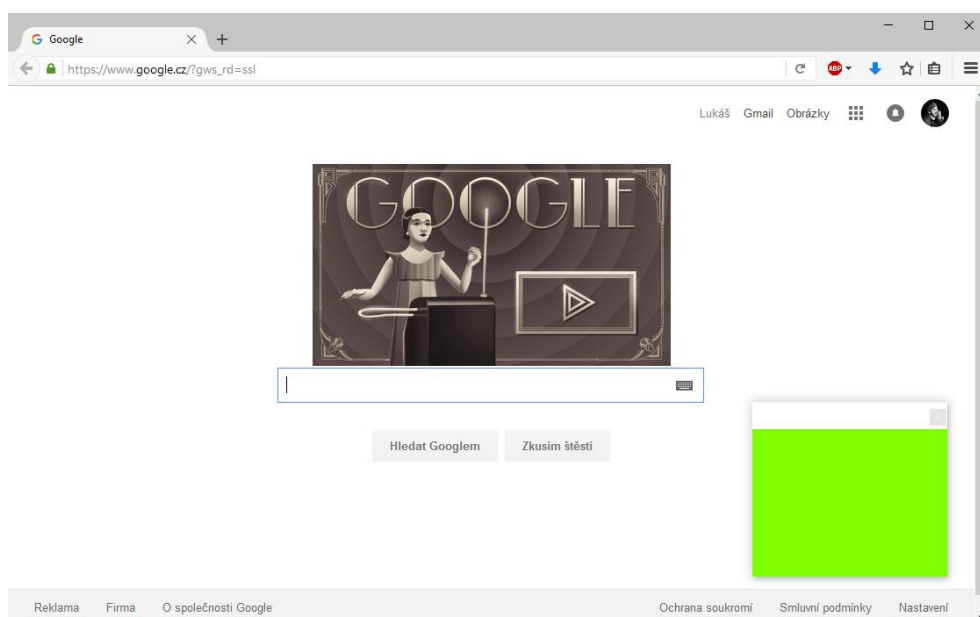


```
file:///C:/Users/lukas/Documents/Visual Studio 2015/Projects/technicka_olympiada/ConsoleApplication1/bin/Release/data_miner.EXE
COM15
Napište název portu, který chcete sledovat.
COM15
Úspěšné otevření portu.
Stiskněte ENTER pro kalibraci.
Kalibrace - OK 14892 12820
Začínám sledovat a přijímat data.
14840 12848
14908 12744
14836 12836
14988 12844
14896 12796
14896 12792
14892 12964
14836 12824
14860 12800
15040 12724
14940 12696
14920 12668
14840 12856
14804 12804
15044 12732
14960 12740
14972 12800
15020 12852
15024 12816
14876 12812
14912 12724
14776 12736
```

obr. 8 – konzolový výstup programu data_miner.exe

3.4 Program simplified_checker.exe

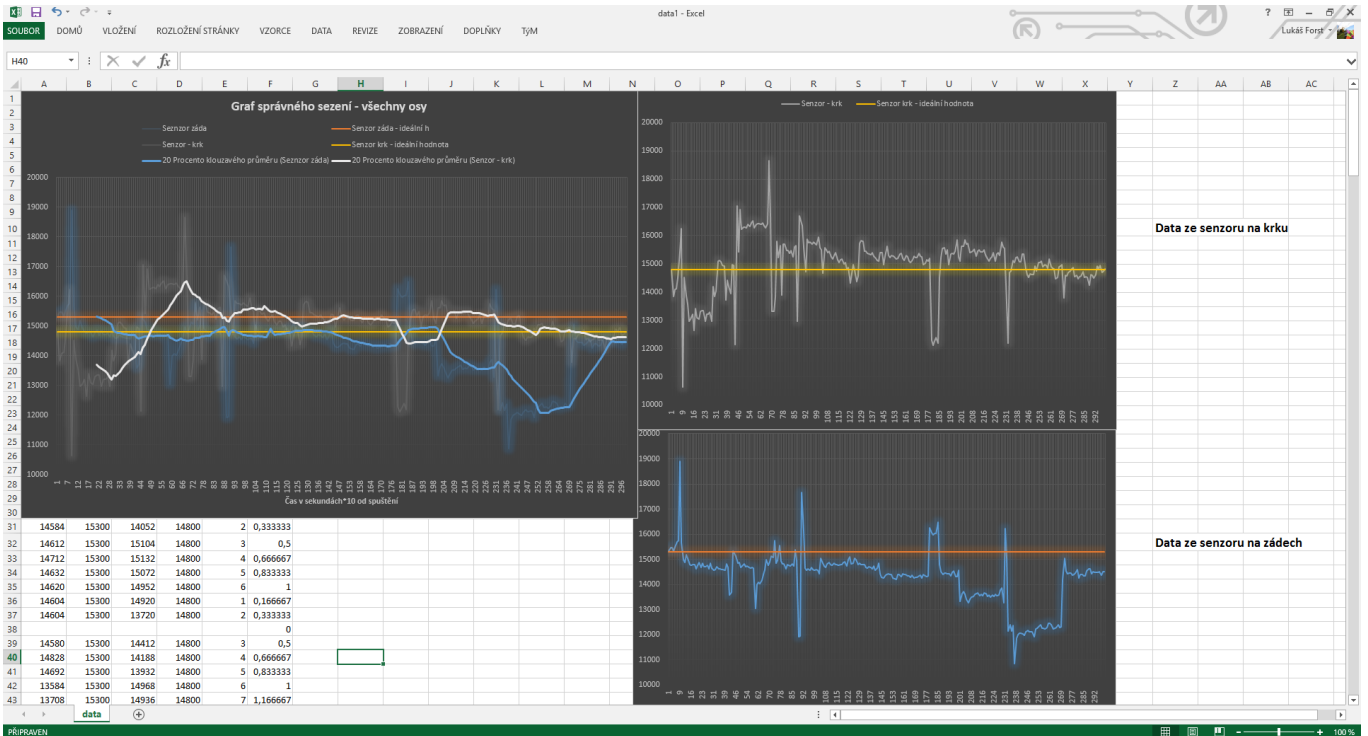
Aplikace určená ke každodennímu využití. Primárně slouží k indikaci správného držení těla, ale zároveň zaznamenává data k pozdějšímu zpracování. Jeho minimalistické GUI je nastaveno tak, že je stále pro uživatele viditelné, a to i když pracuje v jiném programu.



obr. 9 – GUI programu simplified_checker.exe při překrytí prohlížeče Firefox

4. Vyhodnocení uložených dat

Uložená data v počítači se vyhodnocují vizuálně v programu MS Excel, do kterého je nainportuje program *arduino_ext.exe*.

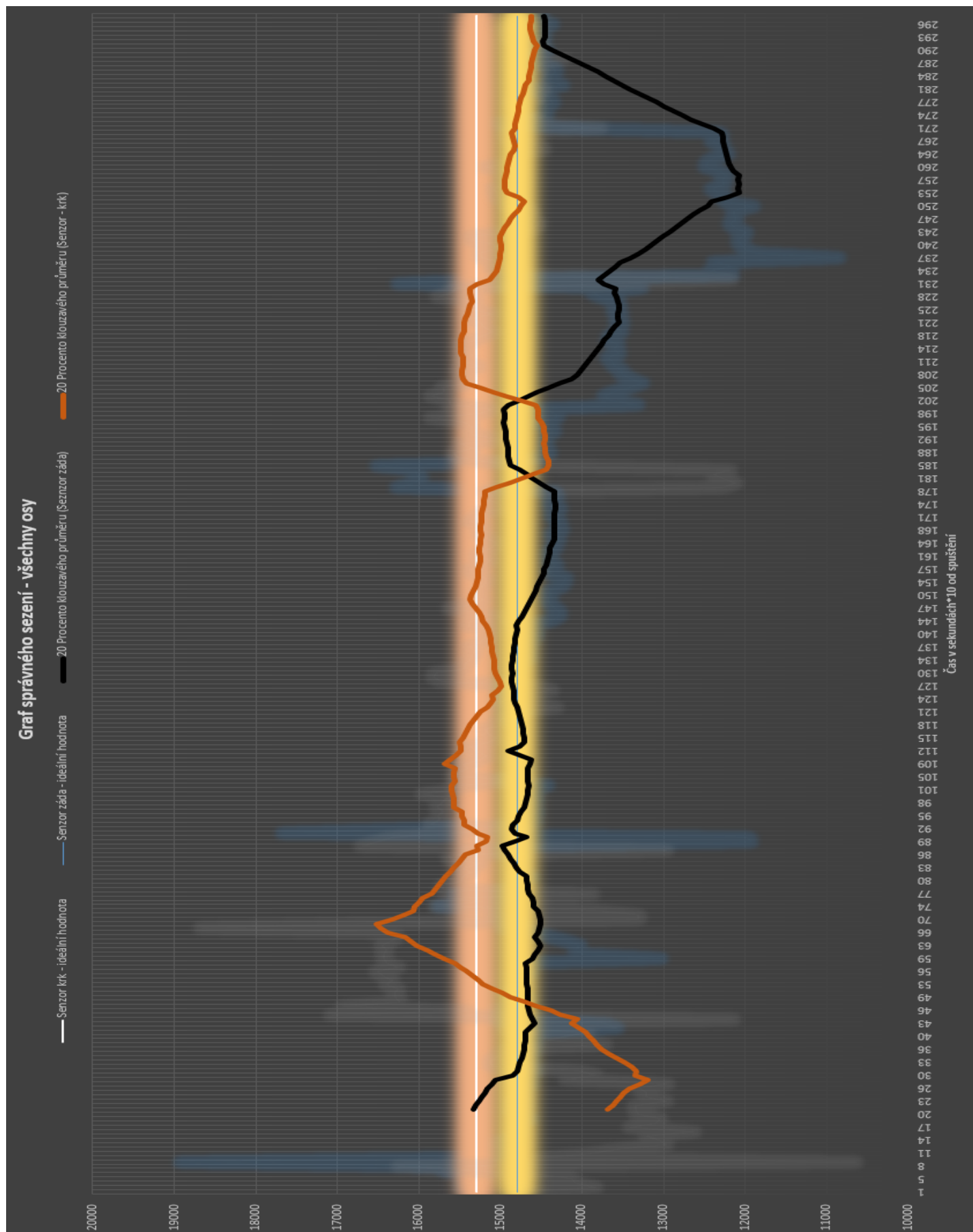


obr. 10 – ukázka vyhodnocení dat v MS Excel

Data jsou formátována tak, že stačí pouze označit oblast dat a nechat vytvořit graf. Z grafu lze pak okamžitě vidět, kdy se uživatel krčil a kdy ne.

Na přiloženém grafu lze vidět, jak vypadá vyhodnocení (obr. 9). Na grafu jsou čtyři křivky, z toho dvě konstantní. Konstantní křivky znázorňují ideální polohu senzorů (tyto ideální hodnoty jsou zaznamenány při kalibraci, kdy uživatel sedí rovně) a oblast, kde by se data měla vyskytovat (surová data jsou lehce nestabilní, testováním a zdokonalováním byla určena odchylka +200;-200). Další dvě křivky znázorňují klouzavý průměr reálných hodnot. Díky použití klouzavého průměru jsem odstranil špičky (uživatel se pravděpodobně vrtěl na židli nebo si přisedával).

Pokud se klouzavý průměr nachází v oblasti s ideálními hodnotami, uživatel seděl naprosto správně, pokud ale nikoliv, můžeme s jistotou říct, že byl buďto v chabém držení v sedu (pokud je senzor na krku mimo), nebo ve zborceném sedu (senzor na zádech je mimo).



obr. 11 – výsledný graf polohy senzorů

5. Průzkum potencionálního trhu

Během práce na samotném zařízení jsem provedl průzkum potencionálního trhu na internetu. Ptal jsem se na to, kolik hodin tráví lidé sezením u počítače; zda si myslí, že se při práci na něm hrbí, a jestli by uvítali možnost použít zařízení, které zjistí, že nesedí správně. Průzkum jsem provedl na serveru Survio (<http://www.survio.com/cs/>) a použil volně dostupnou verzi zdarma – byl jsem limitován počtem respondentů (100 unikátních přístupů), tento limit jsem naplnil během devíti hodin.

Použil jsem dvě rozdílné skupiny respondentů a výsledky jsem následně zanesl do grafů společně.

1. skupina – přátelé na sociální síti Facebook – vzorek běžných uživatelů PC
2. skupina – komunita „Android (CZ/SK)“ na sociální síti Google Plus⁷ – pokročilejší uživatelé počítače; vysoký počet lidí, kteří se prací na PC žíví

Z průzkumu vyplynulo, že převážná část respondentů tráví u počítače více než 3 hodiny denně (**65%**). Co mě osobně překvapilo, bylo to, že **79%** respondentů si uvědomuje, že se při práci na počítači hrbí. **56%** odpovídajících by uvítalo možnost použít zařízení, které detekuje, že nesedí správně a upozorní je na to.

Výsledky celého průzkumu jsou graficky znázorněny na konci celé práce.

⁷ Skupinu lze nalézt na adrese - <https://plus.google.com/communities/104404104334356453927>

6. Závěr

V průběhu práce jsem se musel vypořádat s mnoha problémy, jako např.:

- zapojení dvou zařízení do linky SDA / SCL – nedostatek návodů, popisů zapojení na internetu
- synchronizace práce Arduina a PC; Arduino bylo na některé věci příliš pomalé, musel jsem proto pečlivě volit časové intervaly požadavků od počítače, aby Arduino stíhalo tyto požadavky vykonávat
- počáteční kalibrace senzorů
- interpretace výstupu ze senzorů
- paralelní programování – synchronizace dvou a více vláken programu a předávání dat navzájem mezi sebou

Cíle práce se ale podařilo naplnit a výsledkem je plně funkční zařízení, které je schopno v reálném čase schopno detekovat nesprávné držení těla při práci u počítače. Prototyp zařízení, který jsem vytvořil, lze již bez problému použít jako zdravotní pomůcka. Buď jako prevenci, nebo jako podpůrné zařízení při léčbě, kdy pacient musí být ve vzpřímené poloze. Zařízení lze také použít pro sledování jednotlivých částí zad a to pouhým přidáním senzorů na místa, která chceme sledovat.

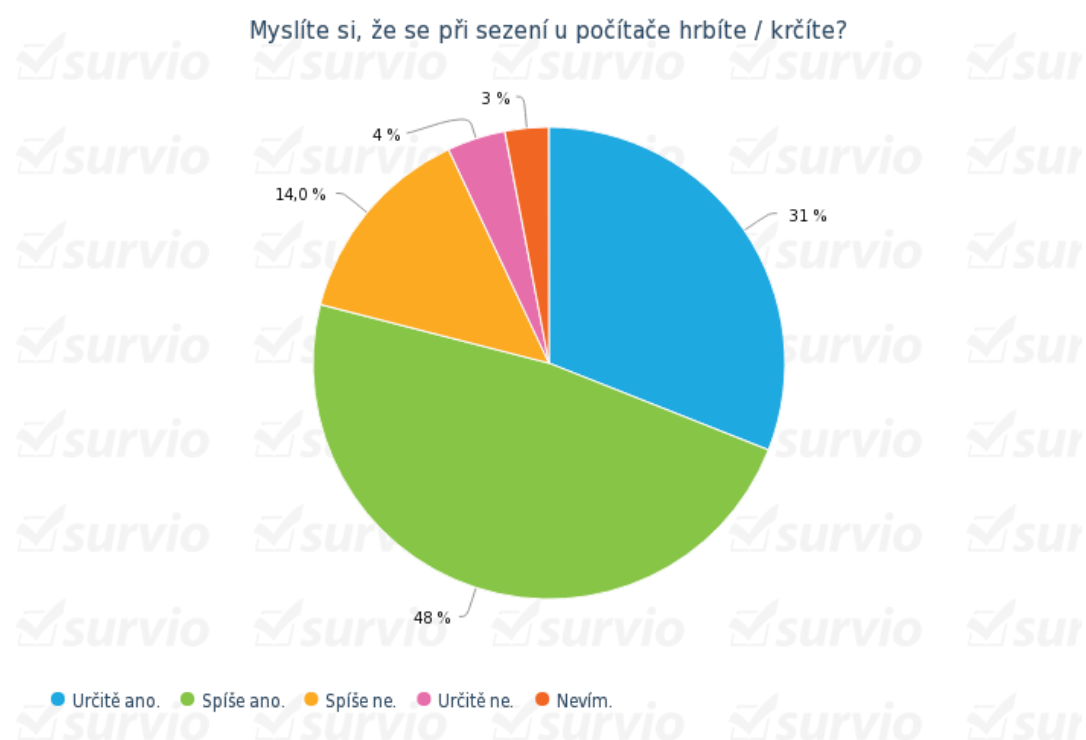
Zdrojové kódy jsou mým vlastním know-how a v tuto chvíli je nehodlám zveřejňovat, proto také nejsou přílohou této práce. Po skončení veškerého vývoje zařízení kódy a s nimi i všechny technické nákresy zveřejním pod licencí open-source na webu GitHub. Hodlám také napsat podrobný návod na výrobu celého zařízení a následně ho zveřejnit na diyhacking.com nebo na nějakém jiném, ať už českém nebo zahraničním serveru.

7. Dodatky

7.1 Data z průzkumu

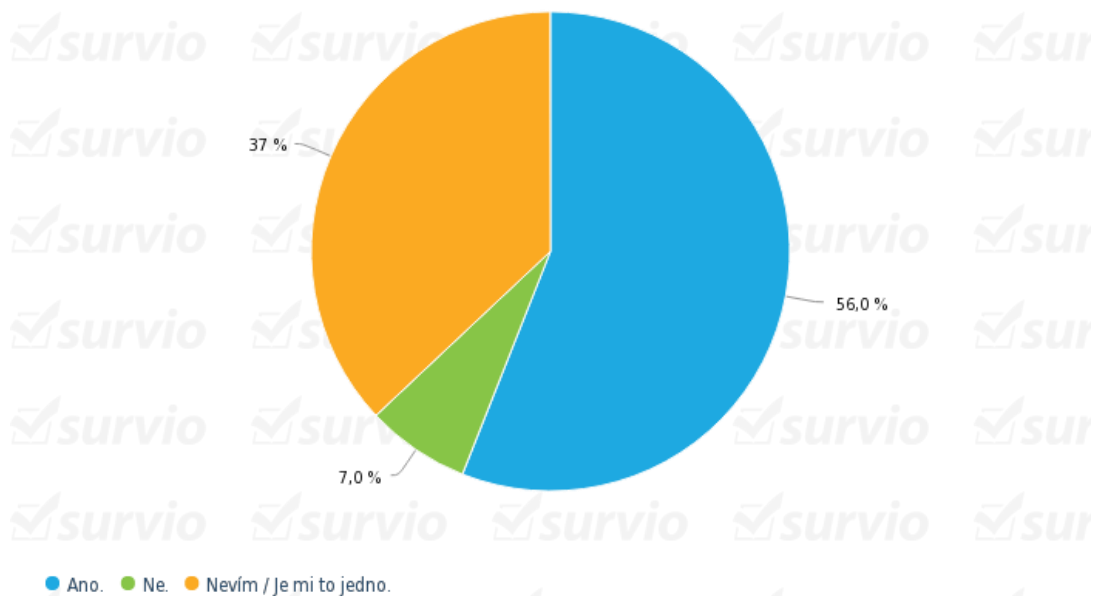


obr. 12 – data z průzkumu, zdroj: <http://www.surveio.com/cs/>



obr. 13 – data z průzkumu, zdroj: <http://www.surveio.com/cs/>

Uvítali byste možnost použít zařízení, které zjistí, že nesedíte správně?



obr. 14 – data z průzkumu, zdroj: <http://www.surveio.com/cs/>

7.2 Seznam užitečných odkazů

- <http://www.arduino8.cz/> - návody na sestavování a zapojování modulů
- <https://www.arduino.cc/> - dokumentace k hardwaru
- <http://diyhacking.com/> - návody na ovládání modulů
- <https://github.com/> - zdroj knihoven pro snadnější ovládání modulů
- <http://www.instructables.com/> - podrobné návody na využití Arduina
- <http://stackoverflow.com/> - fórum s tipy a řešeními, využíval jsem pro C#
- <https://msdn.microsoft.com/en-us/library/windows/desktop/> - dokumentace k jazyku C#
- <https://github.com/jrowberg/i2cdevlib> - repositář knihoven I²C, dokumentace
- <http://www.surveio.com/cs/> - platforma využitá pro průzkum
- <http://playground.arduino.cc> – dokumentace k modulům
- <https://cs.wikipedia.org/wiki/I%C2%B2C> – informace o funkcionalitě I²C sběrnice

8. Seznam obrázků

- 1) různé druhy sezení na židli, zdroj: http://i.idnes.cz/11/053/cl5/MBB3b50f0_sezeni.jpg
- 2) ArduinoNano, zdroj:
https://www.arduino.cc/en/uploads/Main/ArduinoNanoFront_3_1g.jpg
- 3) MPU-6050, zdroj: <http://playground.arduino.cc/uploads/Main/mpu-6050.jpg>
- 4) umístění senzorů na tělo, v. 1.0, zdroj: archiv autora
- 5) schéma zapojení verze 1.0 v open-source programu Fritzing, zdroj: archiv autora
- 6) verze 2.0
 - 6.1. verze 2.0 napojena na triko, zdroj: archiv autora
 - 6.2. verze 2.0 napojena na triko, detail Arduino Nano, zdroj: archiv autora
- 7) GUI programu arduino_ext.exe, zdroj: archiv autora
- 8) konzolový výstup programu data_miner.exe, zdroj: archiv autora
- 9) GUI programu simplified_checker.exe při překrytí prohlížeče, zdroj: archiv autora
- 10) ukázka vyhodnocení dat v MS Excel, zdroj: archiv autora
- 11) výsledný graf polohy senzorů, zdroj: archiv autora
- 12) data z průzkumu, zdroj: <http://www.survio.com/cs/>
- 13) data z průzkumu, zdroj: <http://www.survio.com/cs/>
- 14) data z průzkumu, zdroj: <http://www.survio.com/cs/>

9. Seznam zdrojů a užití literatury

- [1] MPU-6050 Accelerometer + Gyro. *Arduino Playground*. [online]. [cit. 2016-02-28]. Dostupné z: <http://playground.arduino.cc/Main/MPU-6050>
- [2] I2C device library collection for AVR/Arduino or other C++-based MCUs. *GitHub*. [online]. [cit. 2016-02-28]. Dostupné z: <https://github.com/jrowberg/i2cdevlib>
- [3] Arduino Nano User Manual. *Arduino*. [online]. [cit. 2016-02-28]. Dostupné z: <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>
- [4] Vlákna v C# – 7. díl. *Programujte.com*. [online]. [cit. 2016-02-28]. Dostupné z: <http://programujte.com/clanek/2008091400-vlakna-v-c-7-dil/>
- [5] Bolesti zad trápí většinu populace. *Česká průmyslová zdravotní pojišťovna*. [online]. [cit. 2016-03-01]. Dostupné z: <http://www.cpzp.cz/clanek/1148-0-Bolesti-zad-trapi-vetsinu-populace.html>