

OSTRAVSKÁ UNIVERZITA V OSTRAVĚ

Přírodovědecká fakulta

Katedra informatiky a počítačů

Sběr dat z robotických zařízení

Diplomová práce

Autor práce: Bc. Jan Menšík

Vedoucí práce: RNDr. Martin Kotyrba, Ph.D.

2017

UNIVERSITY OF OSTRAVA
Faculty of Science
Department of Informatics and Computers

Data collecting from robotic devices

DIPLOMA THESIS

Author: Bc. Jan Menšík
Supervisor: RNDr. Martin Kotyrba, Ph.D.

2017

OSTRAVSKÁ UNIVERZITA V OSTRAVĚ

Přírodovědecká fakulta

Akademický rok: 2015/2016

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan MENŠÍK**
Osobní číslo: **P15097**
Studijní program: **N1801 Informatika**
Studijní obor: **Informační systémy**
Název tématu: **Sběr dat z robotických zařízení**
Zadávací katedra: **Katedra informatiky a počítačů**

Z á s a d y p r o v y p r a c o v á n í :

Cílem diplomové práce je navrhnout a vytvořit jednotný software pro sběr dat z různorodých snímačů, která jsou připojitelná k robotickému zařízení s cílem následného zpracování.

1. Úvod do robotiky a programovatelných robotů
2. Sběr dat s možností následné manipulace
3. Návrh aplikace pro sběr dat z různorodých snímačů
4. Reálné ověření funkčnosti aplikace
5. Vytvoření metodiky pro následné zpracování uložených dat
6. Závěry a doporučení

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná**

Seznam odborné literatury:

1. Patricia Pulliam Phillips and Cathy A. Stawarski, Data Collection: Planning for and Collecting All Types of Data 1st Edition, ISBN-13: 978-0787987183, ISBN-10: 0787987182.
2. LeJOS EV3. LeJOS, Java for Lego Mindstorms / EV3LeJOS, Java for Lego Mindstorms / EV3 [online]. 1997-2009 [cit. 2014-03-29]. Dostupné z: <http://www.lejos.org/ev3.php>
3. THE LEGO GROUP. Uživatelská příručka Lego MindStorms EV3 [online]. 2013 [cit. 2014-03-29]. Dostupné z: <http://www.lego.com/cs-cz/mindstorms/downloads/user-guides/cs/>
4. Luis Cruz, Humanoid Robot Nao: Developing Behaviours for Soccer Humanoid Robots, LAP Lambert Academic Publishing, 2013.

Vedoucí diplomové práce:

RNDr. Martin Kotyrba, Ph.D.

Katedra informatiky a počítačů

Datum zadání diplomové práce:

9. listopadu 2015

Termín odevzdání diplomové práce:

25. dubna 2017



RNDr. Martin Kotyrba, Ph.D.
vedoucí diplomové práce



doc. RNDr. PaedDr. Hashim Habiballa, Ph.D., Ph.D.
vedoucí katedry

V Ostravě dne 9. listopadu 2015

ČESTNÉ PROHLÁŠENÍ

Já, níže podepsaný/á student/ka, tímto čestně prohlašuji, že text mnou odevzdané závěrečné práce v písemné podobě i na CD nosiči je totožný s textem závěrečné práce vloženým v databázi DIPL2.

Ostrava dne

.....

podpis studenta/ky

ABSTRAKT

Cílem diplomové práce je navrhnout a vytvořit software a univerzální postup pro sběr dat z různých snímačů, která jsou připojitelná k robotickému zařízení Lego Mindstorm EV3 s cílem následného zpracování. Teoretická část je zaměřena na seznámení s výrobkem firmy Lego. Dále následují podrobnější informace o způsobu řešení, podle kterého bude program vypracován. Jsou zde také řešeny konstrukční možnosti robota, jednotlivé vlastnosti senzorů, které má k dispozici a jejich vlastnosti. Text této práce je rozdělen na dvě části. První část představuje seznámení s EV3 a použitými nástroji které byly využity při tvorbě aplikace pro sběr dat, a konstrukci robota. V další, praktické části je popsáno řešení software aplikace pro sběr dat.

Klíčová slova:

(robot, LEGO Mindstorm, zpracování dat, LeJOS, C#, EV3, EV3 API, JSON, XML, AJAX, WebForms, WPF, Singleton, Factory Method, Microsoft Visual Studio)

ABSTRACT

The aim of this thesis is to design and create a software and universal procedure to collect data from various sensors that are connectable to a robotic device Lego Mindstorm to subsequent processing. The theoretical part is focused on familiarization with Lego products. Further below for detailed information on how solution according to which the program will be drawn up. There are also handled the design possibilities of the robot, the individual characteristics of sensors that possession of their properties. The text of this work is divided into two parts. The first part introduces EV3 and the tools used to create the data acquisition application and the robot construction. In the next, practical part is described a software application for data collecting.

Keywords:

(robot, LEGO Mindstorm, data processing, LeJOS, C#, EV3, EV3 API, JSON, XML, AJAX, WebForms, WPF, Singleton, Factory Method, Microsoft Visual Studio)

Poděkování

Autor děkuje RNDr. Martinu Kotyrbovi, Ph.D. za odborné vedení při tvorbě diplomové práce. Rád bych také poděkoval RNDr. Jaroslavu Žáčkovi, Ph.D. za jeho náměty a čas, který mi věnoval při konzultacích a mentoringu.

Prohlašuji, že předložená práce je mým původním autorským dílem, které jsem vypracoval/a samostatně. Veškerou literaturu a další zdroje, z nichž jsem při zpracování čerpal/a, v práci řádně cituji a jsou uvedeny v seznamu použité literatury.

V Ostravě dne

.....

(podpis)

OBSAH

ÚVOD.....	9
CÍLE PRÁCE.....	10
2 POPIS AKTUÁLNÍCH NÁSTROJŮ	11
2.1 LEGO EV3.....	11
2.2 LeJOS	15
2.3 LEGO.Ev3 API	17
2.4 .NET	17
2.5 MYSQL.....	22
2.6 JSON	24
2.7 JavaScript	24
2.8 AJAX.....	26
2.9 JQuery	27
2.10 Singleton.....	28
2.11 Factory Method	28
2.12 HTTP.....	29
3 ANALÝZA SOUČASNÉHO STAVU	31
3.1 LEGO MINDSTORMS EV3 PROGRAMMING SOFTWARE	31
3.2 LEGO MINDSTORMS EV3 Teacher Edition	32
3.3 LEGO MINDSTORMS Education EV3	32
3.4 CodePlex LEGO MINDSTORMS EV3.....	33
3.5 PRemote EV3 and NXT	33
3.6 Závěr analýzy současného stavu	34
4 NÁVRH APLIKACE.....	36
5 POUŽITÍ NÁVRHOVÝCH VZORŮ V NAVRŽENÉM SOFTWARE	40
6 IMPLEMENTACE.....	42
6.1 Klientská část	44
6.2 Serverová část – univerzální možnost zpracování dat	45
7 TESTOVÁNÍ.....	51
8 POUŽITÍ APLIKACE	53
ZÁVĚR	55
SEZNAM OBRÁZKŮ	58
SEZNAM PŘÍLOH.....	60

ÚVOD

Hlavním záměrem práce je vytvoření softwaru, který umožňuje sběr dat ze senzorů umístěné na robotickém zařízení a to takovým způsobem, který umožní řízení sběru dat, monitorování sběru dat a přípravu těchto dat k dalšímu zpracování přímo z prostředí programu. Pro diplomovou práci byl vybrán Lego robot EV3, které byl pro účely této práce zapůjčen z katedry informatiky a počítačů. Sběr dat probíhal prostřednictvím Bluetooth komunikace.

V dnešní době je zařízení Lego MindStorm EV3 velice rozšířený typ modelu, sloužícího nejen k simulacím a výuce i mnoha dalším modelovým akvizicím. Jádrem tohoto zařízení je programovatelná kostka, která je výpočetním mozkiem a lze k ní připojit nepřeborné množství snímačů a následně data vhodnou formou zpracovávat. Studie práce ukázala, že žádný software podporující sběr dat není schopen data načítat a ukládat v reálném čase, proto byla tato funkcionality hlavním bodem navrženého softwaru a postupu pro reálné zpracování těchto pořízených dat.

CÍLE PRÁCE

Cílem diplomové práce je navrhnout a vytvořit software a univerzální postup pro sběr dat z různorodých snímačů, která jsou připojitelná k robotickému zařízení Lego Mindstorm s cílem následného zpracování. Teoretická část je zaměřena na seznámení s výrobkem firmy Lego. Dále následují podrobnější informace o způsobu řešení, podle kterého bude program vypracován. Jsou zde také řešeny konstrukční možnosti robota, jednotlivé vlastnosti senzorů, které má k dispozici a jejich vlastnosti. Praktická část je rozdělena na dvě části. První řeší komunikaci aplikace pro sběr dat s robotem, konstrukci robota a použité nástroje. V druhé praktické části je popsána samotná implementace software a metodické kroky pro sběr dat pro jejich následnou manipulaci.

2 POPIS AKTUÁLNÍCH NÁSTROJŮ

2.1 LEGO EV3

Ev3 je zkratka pro třetí generaci Lego Mindstorms robotů a také typ robota využít pro tuto práci. Je to následovník druhé generace Lego Mindstorms NXT 2.0. Lego Mindstorms EV3 obsahuje vše proto, aby byl robot schopný chodit, mluvit a pohybovat se takovým způsobem, jakým si přejeme. Tato edice byla vydána roku 2013. Existuje mnoho soutěží používající tuto sadu robotů například First Lego League a World robot Olympiád.

Hlavním rozdílem mezi jeho předchůdcem jsou technologické pokroky v programovatelné kostce (slouží jako řídicí centrum a napájecí stanice robota). Hlavní procesor pro NXT byl ARM7 mikrořadič, zatímco EV3 má více výkonný procesor ARM9 s 16GB pamětí běžící na Linuxu. Programovací kostku Novinkou oproti NXT je také USB konektor a mikro SD slot.



Obr. 1: Programovatelná kostka, převzato z ¹

Ev3 má sadu 31313 (tato sada je pro domácí použití), která je dodávána s návody na stavění dalších 17 robotických modelů jako jsou robot humanoid, střílející škorpión, vysokozdvizný vozík, závodní kamion a mnoho dalších. Využívá program Mindstorm k napsání kódu pomocí bloků místo řádků. Kód může být také naprogramovaný na

¹ <http://botbench.com/blog/wp-content/uploads/2013/01/ev3brick.jpg>

aktuálním robotovi a poté uložen. Využívá se také dálkové ovládání. Senzory taktéž prošli vylepšením. Nyní jsou schopny odesílat nové hodnoty do EV3 rychlostí až 1000 záznamů za jednu sekundu, což je několikrát rychlejší než NXT senzory. Senzor barev nově může snímat 7 druhů barev místo 6 a to včetně odstranění světla v pozadí.

EV3 sada 31313 obsahuje:

- Programovatelnou kostku,
- 2 velké motory,
- střední motor,
- senzor dotyku,
- senzor barev,
- infračervený senzor, dálkové ovládání,
- USB kabel a jiné druhy kabelů,
- 585 dílčích prvků k sestavení LEGO robota.

Další sada Lego Mindstorms EV3 s názvem Education Core Set (45544) je především využívána k technickému využití. V tomto setu je navíc obsaženo více motorů, gyroskopický senzor a ultrazvukový senzor. Nabízí vzdělávací podporu, počítačovou výuku a komplexní e-learning program.

EV3 taktéž není příliš náročný na systémové požadavky Vašeho počítače, vyžaduje pouze procesor 2.0 GHZ nebo vyšší paměť, což některé počítače čistě pro pracovní účely nemusí splňovat. V porovnání s předchozími verzemi Lego Mindstorms je EV3 nejvýkonnějším modelem, je vybaven všemi potřebnými konektory pro připojení k ostatním elektronickým spotřebičům (klasický USB port, port pro připojení Apple produktů). EV3 model má taktéž schopnost připojit díly ze starších modelů Lego Mindstorms, čímž představuje velmi užitečnou jednotku.

Technické parametry EV3

- Operační systém – LINUX
- Řadič ARM9 300 MHz
- Paměť Flash – 16 MB
- Paměť RAM – 64 MB
- Rozlišení displeje kostky – 178 x 128 / černobílý
- Komunikace USB 2.0 s hostitelským počítačem – až 480 Mbit/s
- Komunikace USB 1.1 s hostitelským zařízením – až 12 Mbit/s
- Karta microSD – podporuje formát SDHC, verze 2.0, max. 32 GB
- Porty motorů a senzorů
- Konektory – RJ12
- Podpora funkce Auto ID
- Napájení – 6 baterií AA

Senzory EV3

Senzory se k EV3 připojují pomocí kabelu s konektory RJ12 a slouží k digitálnímu zpracování prostředí. Jsou součástí stavebnice, kterou firma Lego dodává spolu se součástmi na výrobu robota a je také základní sadou elektronických senzorů. Každý senzor má specifickou vlastnost a princip, na kterém pracují. Při testování a programování programu pro sběr dat bylo důležité počítat s každým možným připojením senzoru. K sadě EV3 je možné dokoupit několik doplňujících senzorů. Při návrhu aplikace bylo nutné nejdříve zohlednit jednotlivé způsoby snímání informací každého ze senzorů otestováním správnosti návratových hodnot. Nyní si uvedeme některé senzory.

Color Sensor

Color Sensor je digitální senzor, který dokáže detekovat barvu nebo intenzitu světla prostupujícího malým okénkem na čelní straně senzoru. Tento senzor lze používat ve třech různých režimech: Color Mode (barevný režim), Reflected Light Intensity Mode (režim intenzity odraženého světla) a Ambient Light Intensity Mode (režim intenzity okolního světla).



Obr. 2: Color Sensor, převzato z ²

Touch Sensor

Touch Sensor je analogový senzor, který umí detekovat, když je stisknuto červené tlačítko senzoru a následně uvolněno. Touch Sensor lze programovat pomocí tří stavů: stisknuto, uvolněno nebo náraz (stisknutí a uvolnění).



Obr. 3: Touch Sensor, převzato z ³

Infrared Sensor

Infrared Sensor je digitální senzor, který umí detekovat infračervené světlo odrážené od pevných objektů. Umí také detekovat infračervené signály vysílané z Remote Infrared Beacon (Dálkové ovládání). Infrared Sensor lze používat ve třech

² www.legorobotik.ch/EV3Autonom/bilder/ev3Colorstrip.png

³ [sh-s7-live-s.legocdn.com/is/image/LEGO/45507?\\$PDPDefault\\$](http://sh-s7-live-s.legocdn.com/is/image/LEGO/45507?$PDPDefault$)

různých režimech: Proximity Mode (přiblížení), Beacon Mode (maják) a Remote Mode (vzdálený).



Obr. 4: Infrared Sensor, převzato z ⁴

Temperature Sensor

Temperature Sensor je digitální teplotní senzor kalibrovaný na měření :

- stupně Celsia (-20 - 120),
- stupně Fahrenheita (-4 - 248).



Obr. 5: Temperature Sensor, převzato z ⁵

2.2 LeJOS

LeJOS je knihovna, která má nahradit firmware pro LEGO Mindstorms programovatelnou kostku. Jedná se o soubor software podporující původní systém NXT a EV3 obsahují Java virtuální stroj, který umožňuje, aby roboti byli naprogramováni v programovacím jazyce

⁴ [sh-s7-live-s.legocdn.com/is/image/LEGO/45509?\\$PDPDefault\\$](https://sh-s7-live-s.legocdn.com/is/image/LEGO/45509?$PDPDefault$)

⁵ www.eduxe.cz/resize/domain/prezentace/files/les/9749.jpg?w=253&h=190

Java. Nabízí programování v standardním jazyku Java například pomocí IDE Netbeans, nebo Eclipse. Podporuje většinu z tříd `java.lang`, `java.util` and `java.io` a taktéž výborně dokumentované "Robotics API".

LeJOS poskytuje podporu pro přístup k portům robota. Tento přístup umožňuje manipulovat se senzory a motory robota. Díky objektově orientované struktuře Javy, vývojáři jsou schopni ukryt před uživatelem programovací rozhraní (interface), ve kterých jsou implementovány detaily senzorů a pohonů. Tento krok umožní vývojářům dále pokračovat na vysoké úrovni abstrakce. Jedná se tedy o náhradní firmware pro programovatelné kostky společnosti Lego Mindstorms, který v současné době podporuje lego RCX , leJOS NXJ a taktéž podporuje kostky NXT a EV3. LeJOS poskytuje rozsáhlé knihovny tříd, které podporují různé zajímavé funkce, jako je například navigace a chování robota [5].

Podobně jako je tomu u platformy Java i leJOS EV3 je jen softwarová platforma, jenž běží na hardwaru založenou platformou, ale především nad kostkou Lego EV3.

Platforma LeJOS má dvě složky:

- LeJOS JVM ,
- NXJ Application Programming Interface (API).

LeJOS JVM je malý Java Virtual Machine (virtuální stroj) napsaný pro cihly NXT v roce 2006. JVM je napsán v kódu C, jenž je na platformě nezávislý styl. Tato nezávislost umožňuje JVM snadno portovat i na další zařízení. Využívajíc cihly Lego NXT, Nintendo Game Boy Advance a nyní cihla Lego EV3.

LeJOS umožňuje Bluetooth bezdrátovou komunikaci přes sériový port protokolu, stejně jako komunikaci přes USB připojení. Dále představuje podporu pro vytváření pole, rekurze, synchronizace, výjimek, dynamických datových struktur a standardních datových typů [6].

Nebyla využita knihovna LeJOS, protože je třeba přinstalovat původní firmware pomocí mikro SD karty, která přináší další omezení. Například karta sama může způsobovat nekompatibilitu, musí mít minimální velikost 2GB a být naformátovaná na souborový systém FAT32 a verze LeJOS na kartě se musí shodovat s verzí knihovny pro LeJOS. V tom případě by musel každý uživatel programu pro sběr dat mít nainstalovaný LeJOS v každém EV3.

2.3 LEGO.Ev3 API

Je druhá nejvíce využívaná knihovna pro programovatelnou kostku LEGO Mindstorms, kterou vytvořil Brian Peek, Clint Rutkas, Dan Fernandez, avšak využívá C# jako programovací jazyk. Obsahuje knihovnu pro Windows software (WinForms, WPF, Console, etc.). Je zapotřebí zde nainstalovat aktuální Bricxcc verzi pro spuštění API nebo si můžeme použít ev3dev software, který si můžeme sami přepsat. Pro vývoj v EV3 API musíme mít nainstalovaný Microsoft IDE Visual Studio, která běží na operačním systému Windows, nejlépe poslední aktualizovaná verze. Tato knihovna je přímo vytvořena pro použití WPF (Windows Presentation Foundation) což je knihovna tříd pro tvorbu grafického rozhraní, více se o WPF dovíte v následujících kapitolách.

2.4 .NET

Visual Studio

Microsoft Visual Studio je vývojové prostředí od Microsoftu. Může být použito pro vývoj konzolových aplikací a aplikací s grafickým rozhraním spolu s aplikacemi Windows Forms, WPF aplikací, webovými stránkami, webovými aplikacemi a webovými službami jak ve strojovém kódu, tak v řízeném kódu na platformách Microsoft Windows, apod. [8].

VisualStudio obsahuje program na úpravu kódu podporující IntelliSense⁶ a refaktoring. Následující integrované tools obsahují návrhář formulářů pro tvorbu GUI aplikací, návrhář webu, tříd a databázových schémat. Možnost přidávat rozšíření, což vylepšuje funkčnost

⁶ IntelliSense je obecný termín pro funkce jako jsou, seznamy členů, informace o parametrech, dokončování slov.

na téměř každé úrovni a to od doplnění podpory pro vytváření verzí po nové nástroje jako jsou editory a vizuální návrhář pro doménově speciální jazyky nebo pomůcky pro další hlediska návrhu programu. Zabudovaný debugger je opravdu užitečný, pracuje jak na úrovni kódu, tak na úrovni stroje. VS (Visual Studio) podporuje jazyky prostřednictvím modulů, což umožňuje, aby editor a debugger spolupracoval s jakýmkoli programovacím jazykem (mezi ty vestavěné patří C/C++, VB.NET, C#). Další programovací jazyky mohou být přidány jazykovými službami, které musí být nainstalovány zvlášť [11].

Visual Studio nevnučuje žádný konkrétní programovací jazyk nebo nástroj, místo toho je mu možno přidat další rozšíření a funkčnosti. Veškerá funkcionalita je zapouzdřena do balíčku. Když je nainstalována, je dostupná jako služba. IDE poskytuje tři služby:

- SVSSolution (umožňuje očíslovat projekty a sestavy),
- SVSUIShell (poskytuje rozdělování na okna a UI funkce - panely, nástrojové lišty a okna nástrojů),
- SVSShell (má na starosti registraci balíčků VSPackage).

Podpora programovacích jazyků je přidána rozšířením pojmenovaným Language Servis. Jazyková služba nastavuje různá rozhraní, která může implementace VSPackage implementovat pro přidání podpory různé funkčnosti. Funkčnost, která může být tímto způsobem přidána, zahrnuje zvýraznění syntaxe, doplňování příkazů, zvýrazňování párů závorek, tipy parametrů informací, seznamy členů a chybové značky pro kompilaci na pozadí.

Visual Studio, jako každé jiné IDE, obsahuje editor kódu, který podporuje zvýraznění syntaxe a automatické dokončování za použití IntelliSense nejen pro proměnné, funkce a metody, ale také konstrukce jako cykly a dotazy. Visual Studio umožňuje kompilaci na pozadí. Během psaní kódu jej Visual Studio na pozadí kompiluje, aby poskytlo informace o syntaktických a kompilačních chybách, které jsou podtrženy červenou vlnovkou. Varování jsou podtržena zelenou vlnovkou. Kompilace na pozadí negeneruje spustitelný kód, protože používá jiný kompilátor než ten, který generuje spustitelný kód. Debugger pracuje jak se spravovaným kódem, tak se strojovým kódem. Povoluje nastavování

breakpointů a watch (sledují hodnoty proměnných během procesu). Designer pomáhá s vývojem aplikací.

Programovací jazyk C#

C Sharp je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft s platformou (.NET Framework). Microsoft založil C# na jazycích C++ a Java (a je tedy nepřímým potomkem jazyka C, ze kterého čerpá syntaxi) [7].

C# lze využít k tvorbě databázových programů, formulářových aplikací ve Windows, webových aplikací a stránek, webových služeb atd.

C# je jednoduchý, moderní, mnohoúčelový a objektově orientovaný programovací jazyk. Jazyk a jeho implementace poskytuje podporu pro principy softwarového inženýrství, jakými jsou například hlídání hranic polí, automatizovaný garbage collector pro detekci neinicializovaných proměnných. Důležité jsou také jeho vlastnosti jako:

- robustnost,
- trvanlivost,
- programátorská produktivita.

Přenositelnost zdrojového kódu je velmi důležitá, obzvláště pro ty programátory, kteří jsou obeznámeni s C a C++. C Sharp je navržen pro psaní aplikací jak pro zařízení se sofistikovanými operačními systémy, tak pro zařízení s omezenými možnostmi.

V C# neexistuje vícenásobná dědičnost – to znamená, že každá třída může být potomkem pouze jedné třídy. Toto rozhodnutí bylo přijato, aby se předešlo komplikacím a přílišné složitosti, která je spojena s vícenásobnou dědičností. Třída ale může implementovat libovolný počet rozhraní.

Neexistují žádné globální proměnné a metody, všechny musí být deklarovány uvnitř tříd. Náhradou za globální proměnné a metody jsou statické metody a proměnné veřejných tříd [9].

V objektově orientovaném programování se z důvodu dodržení principu zapouzdření často používá vzor, kdy k datovým atributům třídy lze zvenčí přistupovat pouze nepřímo, a to pomocí dvou metod: metody `get` (accessor) a metody `set` (mutator). Místo toho lze definovat tzv. property, která stále funguje jako datový atribut, ale uvnitř obsahuje prostor pro definici obou těchto metod.

C# je typově bezpečnější než C++. Jediné předdefinované implicitní konverze jsou takové, které jsou považovány za bezpečné, a taktéž nepotřebuje žádné před-deklarování na začátku třídy.

Všechny hodnotové datové typy jsou na rozdíl od odkazových typů alokované na zásobníku a to z výkonnostních důvodů. Hodnotové datové typy můžeme rozdělit do tří částí:

- primitivní datové typy (celočíslné datové typy),
- struktury (uživatelsky definované datové typy),
- výpočtové typy (množina předem definovaných hodnot).

WPF

S vydáním .NET 3.0, Microsoft uvolnil druhé, paralelní rozhraní API pro vykreslování grafických uživatelských rozhraní: Windows Presentation Foundation (WPF) založený na rozhraní DirectX, spolu s GUI deklarativní jazyk s názvem XAML .

Windows Presentation Foundation je grafický (GUI) podsystém Microsoft .NET frameworku. Poskytuje platformu pro psaní bohatých klientských aplikací pro desktopy, laptopy a tablety. Je viděn spíše jako náhrada za starší a složitější C++ na bázi Microsoft Foundation Class Library. Hodí se při psaní dotykových aplikací.

Windows Form aplikace je event-driven aplikace podporovaná Microsoft .NET Frameworkem, což znamená, že na rozdíl od dávkového programu je řízena událostmi a často tráví většinu času než uživatel klepne na tlačítko nebo vyplní pole.

Všechny vizuální prvky u formulářů jsou odvozeny od Toolboxu, jenž je přítomen v případě grafického návrhu u souboru s příponou `aspx`. To přidává funkčnost prvku uživatelského rozhraní, jako je umístění, velikost, barva, písmo, text, stejně jako společné akce, jako kliknout a přetáhnout / drop. Třída má také podporu dokovací funkce, která umístí daný element na pozici v rámci svého rodiče. Microsoft Active Accessibility podpora ve třídě také pomáhá uživatelům používat WPF lépe [10].

WPF je postaven na stávajícím systému Windows API a některé ovládací prvky pouze obalují základní součásti systému Windows. Některé z metod umožňuje přímý přístup k Win32 zpětných volání, které jsou k dispozici v platformách Windows.

ASP.NET

ASP.NET je open source webový framework pro vytváření moderních webových aplikací a služeb společně s .NET. ASP.NET vytváří webové stránky založené na HTML5, CSS a JavaScriptu. Jako první bylo vydáno v roce 2002 první verze .NET Frameworku. ASP.NET je vybudován na Common Language Runtime (CLR), dovoluje programátorům psát ASP.NET kód s využitím jakéhokoli podporovaného jazyka .NET [13].

Microsoft představil kód na pozadí modelu, který umožňuje že statické texty zůstanou na stránce ASPX, zatímco dynamický kód zůstává v `aspx.vb` nebo `aspx.cs` nebo `aspx.fs` souboru (v závislosti na použitém programovacím jazyce).

Stav aplikace je držen v kolekci sdílených uživatelem definovaných proměnných, které jsou inicializovány a nastaveny po startu aplikace a jsou dostupné po délku existence jejich poslední instance. ASP.NET podporuje tři režimy dostupnosti pro proměnné straně serveru:

- In-process mód – Proměnné jsou udržovány po celou dobu procesu. Jedná se o nejrychlejší způsob. Po ukončení procesu jsou proměnné ztraceny.
- State server mód – ASP.NET spouští samostatnou službu v prostředí systému Windows, která udržuje stavy proměnných. Vzhledem k tomu, že správa proměnných se děje mimo proces ASP.NET, je State server

mód pomalejší než In-process, ale na rozdíl umí držet hodnoty proměnných i po ukončení procesu ASP.NET.

- SQL Server mód – Tento mód ukládá hodnoty v databázi. Hlavní výhodou tohoto módu je, že dovoluje aplikacím přenést zátěž ze správy proměnných na databázový server. Toto je nejpomalejší způsob uchovávání stavu proměnných v ASP.NET.

ASP.NET se zaměřuje hlavně na výkon, oproti ostatním interpretově založeným technologiím (ASP), a to tím, že kompiluje kód na straně serveru poprvé, kdy je využit do jednoho nebo více DLL souborů na straně serveru. Tyto DLL soubory obsahuje Microsoft intermediate language runtime (MSIL) pro běh mimo common language runtime. Tato vlastnost nabízí zvýšení výkonu nad skriptovací jazyky jako Python a JSP. Tato kompilace se děje automaticky když je poprvé dotázána webová stránka, což znamená, že vývojář nemusí provádět separátní kompilování pro stránky [12].

Výhody ASP.NET jsou :

- oproti ASP je aplikace kompilována již při vývoji a běží rychleji,
- ovládací prvky definované uživatelem lze použít jako šablony,
- redukuje duplicitní kód,
- velký výběr knihoven a ovládacích prvků,
- podporuje mnoho programovacích jazyků,
- podporuje mnoho různých OS i webových serverů (IIS-Windows, Apache-Windows, Linux s open source implementací .NETu Monem),
- generuje HTML, XHTML a JavaScript.

2.5 MYSQL

Oficiálně se vyslovuje „My S-Q-L“. Je to open-source relační databázový management systém (RDBMS). Vývoj projektu MySQL byl založen na volné dostupnosti kódu za podmínek GNU General Public License.

Nyní je MySQL ve vlastnictví společnosti Oracle Corporation. MySQL je jedním z hlavních prvků LAMP což je open-source webový aplikační balíček. LAMP je zkratka pro „Linux, Apache, Perl/PHP/Python“. Aplikace, které používají MySQL databázi jsou například: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal. Také je využíván v mnoha známých rozsáhlých webových stránkách, jako je Facebook, Twitter, Flickr, YouTube, dokonce i Google (i když ne pro vyhledávání) [14].

MySQL je napsán v jazyce C a C++. Je to SQL parser, funguje na mnoho systémových platformách, včetně macOS, Microsoft Windows, iOS, Oracle Solaris, UnixWare, Symbian a Linuxu. MySQL je hodně známý a dobře hodnocen v průměrném užívání, Byl také testován aby byl „rychlý, stabilní, a pravý víceuživatelský více vláknový SQL databázový server“.

Byly zmíněny některé vlastnosti MySQL, zde jsou výhody a nevýhody.

Výhody

- Je to rychlý, nenáročný, hlavně spolehlivý databázový server.
- Je zdarma, po zakoupení licence máte přístup ke zdrojovému kódu.
- Je multiplatformní a snadno přenositelný.
- Server je možno využít i pro jiné aplikace v různých programovacích jazycích.

Nevýhody

- MySQL nepodporuje složitější programátorské konstrukce a nemá dostatečný výkon v náročných webových aplikacích.
- V náročných aplikacích jsou používány konkurenční databáze (Oracle, PostgreSQL).

V mém případě využívám MySQL databázi ve verzi 5.7.9 k ukládání hodnot na „vzdáleném“ serveru.

V databázi se ukládá záznam jako ID zařízení, název portu, čas pořízení záznamu a naměřená hodnota. Kontrola a správa databáze probíhá přes phpMyAdmin verze 4.5.2. Celá databáze běží na webovém serveru Apache/2.4.17. To vše se nachází v balíčku WAMPSEVER, což je balíček/prostředí pro vývoj webových aplikací ve Windows. Obsahuje Apache2, PHP a MySQL databázi společně s PhpMyAdmin, který pomáhá jednoduše spravovat databázi MySQL.

2.6 JSON

JSON, neboli JavaScript Object Notation, je formát pro výměnu dat nezávislý na počítačové platformě. Zpočátku se využíval obecný značkovací jazyk XML, tedy formát ke čtení výstupu, avšak tento způsob výstupu stále se vyvíjející době již nevyhovoval. XML se nezabývá vzhledem konečného výstupu a je třeba použít speciální software konkrétní firmy pro zasílání dokumentů v konkrétním tvaru (DOC, XLS, ODT).

Vstupem do tohoto formátu jsou libovolné datové struktury, které jsou při výstupu snadno čitelné textové řetězce. Tyto datové struktury (textový řetězec, čísla, boolean, hodnota null, pole, objekt) se vkládají explicitně do zápisu kódu. S ostatními datovými typy takto pracovat již nemůžeme. Využívá notaci javascriptu. Můžeme konvertovat jakýkoliv javascriptový objekt na JSON objekt a poslat jej na server, kde se přečte. Lze to i opačně, můžeme konvertovat data v JSON formátu zpětně na javascriptový objekt. I když tento formát může ve výstupu zobrazit pouze textový řetězec, lze snadněji odeslat data na server a používá se jako formát dat, který může být využíván jakýkoliv programovacím jazykem z rodiny jazyků C (např. Java, C++, C, Python, PHP), [15].

2.7 JavaScript

JavaScript je multiplatformní objektový programovací jazyk používaný k vytváření interaktivních internetových stránek. V dnešní době je velmi využíván, často se vkládá explicitně do HTML kódu stránky, kde lze vytvářet interaktivní objekty (animace, tlačítka, různé efekty). Pro spuštění JavaScriptu na webovém prohlížeči je předpoklad mít jeho podporu, která je integrovaná přímo do všech významných webových prohlížečů včetně Exploreru, Firefox, Safari, a taktéž mít povolené Java scripty ve výchozím nastavení.

Notace JavaScriptu patří do skupiny programovacích jazyků rodiny jazyka C, ale od těchto jazyků se výrazně liší jeho sémantikou. Obsahuje objektový model, který poskytuje dynamickou dědičnost, tedy zděděná funkcionalita se může lišit pro jednotlivé objekty. Tomuto přístupu se říká dědičnost prototypová. I když v JavaScriptu nefunguje hierarchické dědění jak je to např. u Javy, můžeme tuto situaci úspěšně nasimulovat vytvořením dalších prototypů. V JavaScriptu není třeba deklarace jednotlivých proměnných, tříd, dokonce i metod, samostatné funkce mohou být vlastností objektů. Není třeba dbát důraz nad modifikátorem přístupu (zdali jsou veřejně přístupné či nikoliv), není třeba implementace interface a pracuje s asociovanými hodnotami [16].

JavaScript obsahuje standardní knihovny objektů, polí, matematických a datových funkcí, a základní soubor jazykových prvků jako jsou operátory, řídicí struktury a STATEMENTS. Tyto základní soubory mohou být rozšířeny o přídavné objekty, jako jsou např.:

- Client-side JavaScript (rozšíření základních jazyků pro podporu objektů a ovládání prohlížeče a jeho DOM⁷)
- Server-side JavaScript (rozšíření základních jazyků pro podporu objektů důležité pro běžící JavaScript na serveru)

Program v JavaScriptu se obvykle spouští na straně klienta, na rozdíl od jiných jazyků (PHP, ASP), které se spouští na straně serveru. Spoléhá na run-time prostředí, aby se objekty a metody mohli stát interaktivními při spuštění webové stránky, reagují rychleji na požadavky klienta a aplikace je mnohem citlivější. JavaScript se velmi často vyskytuje i samostatně mezi HTML kódem pro skriptování požadovaných funkcí / objektů a DOM.

JavaScript zpracovává zprávy z fronty jeden po druhém. Po zavedení nové zprávy, volá funkce spojené s touto zprávou, která vytváří rámec zásobníku volání. Zásobník volání zmenšuje a roste exponenciálně na základě potřeb funkce. Po ukončení funkce, když je zásobník prázdný, JavaScript pokračuje na další zprávu ve frontě. Tento jev se nazývá

⁷ DOM – Document Object model, který reprezentuje XML nebo HTML dokument a umožňuje přizpůsobování obsahu, struktury, nebo styly dokumentu a jeho částí.

smyčka události, protože každá zpráva je zcela zpracovaná před další volanou zprávou. Nicméně, souběžnosti modelu jazyku popisuje princip smyčky pro pořádání akcí, jako non-blokování: Vstupní program / výstup se provádí pomocí událostí a funkce zpětného volání. Např., že JavaScript může zpracovávat klepnutí myši při čekání na databázový dotaz vrátet informace.

2.8 AJAX

AJAX je zkratkou pro **A**synchronous **J**avaScript and **X**ML a představuje širokou skupinu technologií pro vytváření interaktivních webových stránek využívající různé webové technologie (knihovny v JavaScriptu – např. JQuery) na straně klienta a vytváří tak synchronní běh stránky / aplikace [17].

Proč používat AJAX:

- update webové stránky bez opětovného načtení celé stránky,
- žádost o data ze serveru – po načtení stránky,
- zaslání dat na server – tato akce se provádí na pozadí,
- získání dat ze serveru – po načtení stránky,
- nezávislý na softwaru webového serveru.

Tato technologie využívá:

- HTML a CSS pro prezentaci dat společně s DOM a JavaScriptem pro zobrazení dynamického kontextu,
- XMLHttpRequest objekt pro asynchronní výměnu dat s webovým serverem.

Hlavní výhodou užití AJAXu je asynchronní běh ze strany klienta, kdy klient zašle žádost na server, který se zpracovává na pozadí. Tudiž není nutné neustále znovu obnovovat a překreslovat webovou stránku či aplikaci, ale zobrazí se pouze obnovená aktualizovaná část. Další výhodou je využití JavaScriptu, kde programátor není ničím omezen a může tak vytvářet vše, co potřebuje [17].

Nevýhodou se stává takový prohlížeč, který nepodporuje JavaScript nebo XMLHttpRequest, anebo jej uživatel úmyslně blokuje. Uživatel tak nebude moci správně použít webové stránky, které jsou závislé na AJAXu. Další nevýhodou jsou jednodušší zařízení, jako jsou např. mobilní telefony, nepodporující tyto technologie a tak není možné využití AJAXu. Jak správně vrátit funkčnost do těchto aplikací / webových stránek je použití jiných technologií než je JavaScript.

2.9 JQuery

JQuery je malá, rychlá JavaScriptová knihovna, která vytváří průchod a manipulaci, stejně jako HTML, a zpracovává události a animace. Drží se hesla „napiš méně, udělej více“, lze tak rychle a snadno dosáhnout požadovaných výsledků. Díky této knihovně, která je velmi všestranná, rychle rozšiřitelná a jednoduchá, se stal JavaScript velmi populární [18].

JQuery má spoustu společného z JavaScriptem, jako jsou volání AJAXu a manipulace s DOM. Využívá princip nevtrivního JavaScriptu, tedy odděluje „chování“ od struktury HTML. JQuery knihovna obsahuje:

- manipulaci s HTML / DOM,
- možnost úpravy CSS,
- události,
- efekty a animace,
- AJAX,
- utility,
- JavaScriptové plugíny.

JQuery knihovna se dá jednoduše stáhnout (je dostupná zdarma) nebo se na ni můžeme odkazovat přes webové stránky Google, kde je tato knihovna nahrána. Je to velké plus pro všechny uživatele webového prohlížeče Google, jelikož JQuery knihovna je již načtena v cache prohlížeče. Kromě Google je knihovna dostupná již na všech významných webových prohlížečích [18].

2.10 Singleton

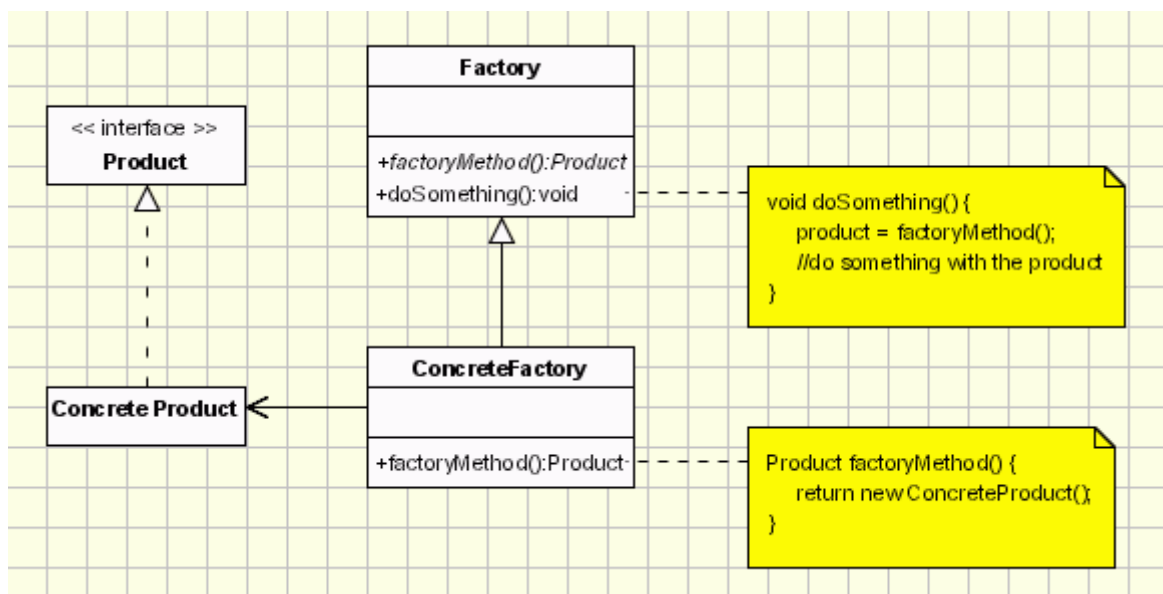
Singleton neboli jedináček je návrhový vzor pro zajištění existence pouze jediné instance dané třídy a poskytne k ní globální přístupový bod. Singleton je jednou z nejpoužívanějších návrhových vzorů pro svou účelnost. Hlavní motivací použití Singleton je při sdílení instancí mezi několika objekty, kde potřebujeme, aby se objekty pohybovali pouze ve vymezeném prostředí a zajistit tak jedinečnost daného objektu. Je mnoho způsobů, jak vytvořit tento návrhový vzor, avšak se musí zvolit taková implementace, která pokrývá všechny problémy aplikace a není příliš složitá [20].

Implementace vzoru Singleton je velmi jednoduchá, stačí navrhnout jedinou třídu s privátní konstruktorem a statickou metodou `getInstance()`. Jedinou nevýhodou tohoto návrhového vzoru je globální přístupový kód. Tento problém se dá eliminovat, pokud je Singleton navrhnout jako globální funkce či služba [20].

2.11 Factory Method

Factory Method je definovaný virtuální interface / metoda po vytváření objektů, u kterých není třeba definovat přesnou třídu a nechá podtřídy rozhodnout, která třída se má vytvářet instanci. Tento interface závisí na dědičnosti, jelikož schopnost vytvářet objekty je ponechán na podtřídách. Factory Method se používají především u konstantních tříd⁸ a poskytuje jeden z nejlepších způsobů, jak vytvořit objekt [20].

⁸ Konstantní třída – u těchto tříd nelze změnit jejich stav, pokud chceme změnit jediný parametr, je třeba vytvořit instanci novou.



Obr. 6: Factory method, převzato z ⁹

2.12 HTTP

HTTP (**H**yper**T**ext **T**ransfer **P**rotokol) je internetový protokol pro komunikaci mezi vzdáleným serverem a internetovým prohlížečem WWW, a přenáší objekty libovolného typu (webové stránky, obrázky, odkazy...). Původně tento protokol byl pouze pro přenos hypertextových odkazů. HTTP není nijak chráněn před vnějším prostředím. K tomuto slouží protokol HTTPS, který řeší problémy šifrováním dat mezi klientem a serverem [21].

HTTP funguje na principu dotaz → odpověď, kde se nejčastěji dotazuje klient a odpověď přichází ze strany serveru (programu). Klient se dotazuje většinou na webovou stránku, připojí se na server a požádá jej o URL kód stránky. Server tuto žádost zpracuje a pošle zpět jako odpověď. Klient tak má možnost si stránku přečíst, jakmile od serveru obdrží http hlavičku a text stránky v HTML [21].

Tomuto protokolu se také říká „bez-stavový“, jelikož každý příkaz je spuštěn nezávisle bez jakékoliv znalosti příkazů. Tento nedostatek je řešen v dalších technologiích běžné pro dnešní internetové rozhraní (Java, JavaScript, cookies, ...) [21].

V HTTP lze jednoduše vytvořit cestu, lze jednoduše číst, editovat a modifikovat data pomocí REST architektury. REST (Representational State Transfer) je rozhraní určené

⁹ <http://www.oodeesign.com/factory-method-pattern.html>

pro distribuované prostředí zaměřené na data. Není standardem, ale využívá určitá omezení, která jsou nutná dodržovat [21].

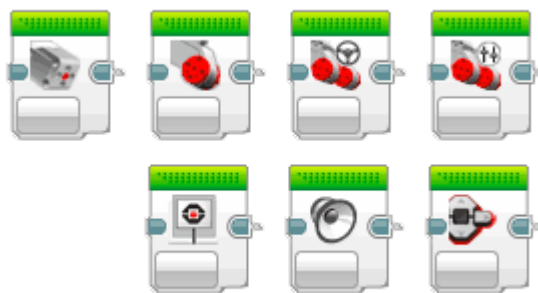
Další možností jak získat mediální obsah z webové stránky je dotazovací protokol POST. Tento protokol se obvykle odesílá prostřednictvím formuláře HTML a výsledkem je změna na serveru. POST je vhodný použít, pokud chceme bezpečně předat citlivé údaje, jako jsou např. hesla apod. Příkladem nevýhody může být, že tato metoda nedovoluje zaslání dat, které přesahuje kapacitu 20 MB při využití serveru PHP.

3 ANALÝZA SOUČASNÉHO STAVU

V úvodní části bych se chtěl zaměřit na aplikaci jako takovou, a jaké aplikace máme k dispozici. Porovnal jsem nejznámější aplikace pro sběr dat z LEGO EV3 a zjistil jsem jejich hlavní nedostatek.

3.1 LEGO MINDSTORMS EV3 PROGRAMMING SOFTWARE

Jedná se o aplikaci sloužící k programování pohybů robota. Aplikace využívá k programování robota ikony, které demonstrují části algoritmu. Aplikace má velmi jednoduché ovládání, které zvládne kdokoli.

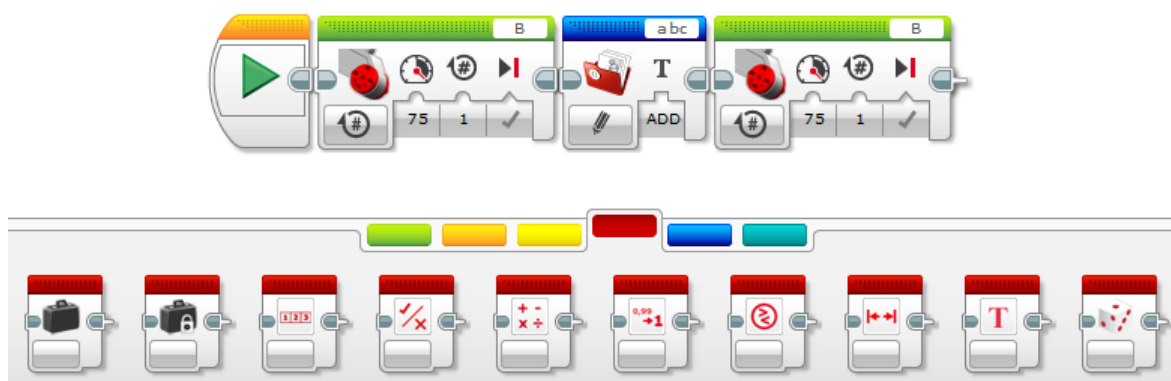


Obr. 7: Náhled ikon, převzato z ¹⁰

Aplikace má mnoho funkcí, díky kterým rozpohybujete robota, umí také ukládat data do textového dokumentu. Ale tím práce s daty končí. Aplikace je určena především pro programování pohybů robota. Můžeme ji také nalézt pod názvem LEGO EV3 Home Edition.

¹⁰ [https://lc-www-live-s.legocdn.com/r/www/r/mindstorms/-](https://lc-www-live-s.legocdn.com/r/www/r/mindstorms/-media/franchises/mindstorms%202014/learn%20to%20program/learntoprogram_action_blocks_landscape.png?l.r2=-1454760377)

[media/franchises/mindstorms%202014/learn%20to%20program/learntoprogram_action_blocks_landscape.png?l.r2=-1454760377](https://lc-www-live-s.legocdn.com/r/www/r/mindstorms/-media/franchises/mindstorms%202014/learn%20to%20program/learntoprogram_action_blocks_landscape.png?l.r2=-1454760377)



Obr. 8: Náhled Aplikace EV3 Programming Software

3.2 LEGO MINDSTORMS EV3 Teacher Edition

Jako jeho předchůdce je EV3 Teacher Edition také pouze v cizojazyčných verzích, Teacher Edition navíc podporu výuky pro učitele. Zobrazení aktuálních hodnot na senzorech. Aplikace obsahuje mnoho videí a tutoriálů, které mohou pomoci pochopit ovládání a vytváření algoritmů.

Jedná se o rozšíření aplikace LEGO MINDSTORMS EV3 PROGRAMMING SOFTWARE. Také ukládá data lokálně a soubory pro uložení hodnot senzoru ukládá ve formátu RDF¹¹.

3.3 LEGO MINDSTORMS Education EV3

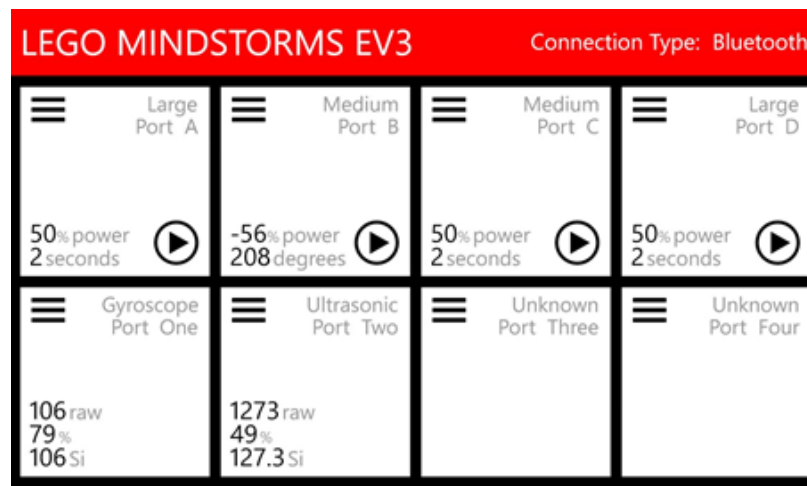
Aplikace je určena jak pro žáky základních škol, tak i pro žáky středních škol, jedná se o „ořezanou“ verzi Teacher Edition, která je ze všech verzí nejobsáhlejší. Nejdůležitější vlastnosti aplikace jsou tyto:

- ukládat data ve formě RDF,
- ovládat robota pomocí USB, Wi-Fi, Bluetooth,
- zobrazovat naměřené hodnoty v grafu,
- ukládat data pouze lokálně.

¹¹ RDF (Resource Description Framework) poskytuje jednoduchý model pro popis zdrojů na webu.

3.4 CodePlex LEGO MINDSTORMS EV3

Aplikace od tvůrců LEGO.EV3 API. Pomocí aplikace je možno se připojit k robotovi EV3 pomocí USB, Bluetooth, nebo Wi-Fi. Aplikace slouží pro kontrolu a čtení dat ze senzorů robota.



Obr. 9: CodePlex náhled aplikace, převzato z ¹²

Tato aplikace umí jak zobrazovat data z robota, tak jej také ovládat. Aplikace umí komunikovat pomocí Wi-Fi, Bluetooth a USB, ovšem neumí nalézt správný COM port a uživatel si jej musí nalézt sám. Aplikace umí dočasně uložit nastavení senzorů.

3.5 PFremote EV3 and NXT

PFremote je aplikace cílící na majitelé iPadu od společnosti Apple, aplikace se prodává za necelých pět eur a obsahuje mnoho funkcí. Podporuje EV3 i NXT. S aplikací můžete:

- ovládat motory robota,
- umí precizně regulovat a kontrolovat rychlosti motorů,
- zvládá nahrávání a opětovné přehrání akcí robota,
- obsahuje uživatelsky pozměnitelné GUI,
- ukládá nastavení aplikace,
- zaznamenává pohyby robota do souboru RoboRemote.rbt.

¹² [https://f.ch9.ms/wlwimages/ae054c0b4d7b402ab1239e6800c0220f/image \[2\]-182.png](https://f.ch9.ms/wlwimages/ae054c0b4d7b402ab1239e6800c0220f/image [2]-182.png)



Obr. 10:PFremote náhled aplikace, převzato z ¹³

3.6 Závěr analýzy současného stavu

Žádná z těchto aplikací neobsahuje možnost odeslání měřených hodnot online. Proto se tato práce zaměřuje hlavně na odesílání dat a možnosti jejich následného zpracování.

Při tvorbě aplikace bylo nutné zohlednit všechny možné technologie připojení, kterými se vyznačuje Brick EV3. Programovací kostka nabízí komunikaci pomocí Bluetooth, USB a také Wi-Fi. Protože klient již komunikuje pomocí HTTP protokolu s webovou aplikací a není jisté, zda pomocí Wi-Fi, či pomocí UTP kabelu a zda by toto připojení nenarušovalo plynulost přenášení dat z robota. Bylo rozhodnuto pro implementaci komunikace pomocí Bluetooth mezi robotem a klientskou aplikací.

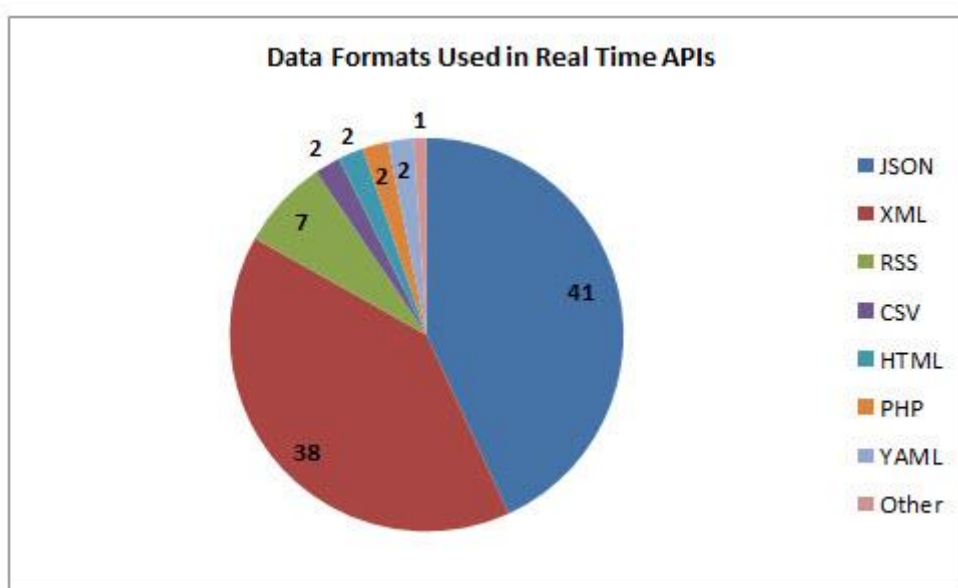
Pro zachování původního firmware, bylo rozhodnuto pro aplikaci v programovacím jazyce C# s využitím knihovny LEGO.EV3 API od Briana Peeka, Clinta Rutkase, Dana Fernandez. Aplikace by měla mít následující vlastnosti.

¹³ <http://media.148apps.com/screenshots/921365042/us-ipad-1-pfremote-ev3-remote-for-power-functions.jpeg>

- Aplikace by se měla jednoduše připojovat k robotovi pomocí příslušného COM portu, viz Obr. 31.
- Aplikace by měla být schopná získávat informace o stavech senzorů a motorů v reálném čase.
- Aplikace by měla zpracovat získaná data a odesílat je do místa určení.
- Aplikace by měla být schopná exportovat získaná data ve formátu JSON.
- Uživatel by měl být schopen měnit senzory pro zaznamenávání dat z robota.

4 NÁVRH APLIKACE

Jak již bylo zmíněno, vývoj aplikace by měl probíhat v jazyce C#, jelikož se nemusí přeinstalovat firmware, což je jistě pro uživatele jednodušší varianta, která by uživatele jinak zatěžovala. Idea tohoto řešení je přenést data z robota EV3 na libovolný server, který je schopen zpracovat získaná data. K tomu bylo potřeba určit formát odesílaných dat. K tomu se nabízí hned tři různé značkovací jazyky XML, JSON, YAML. XML je značkovací jazyk vyvinut konsorciem W3C.



Obr. 11: Datové formáty pro přenos v reálném čase, převzato z ¹⁴

Jak můžete v grafu vidět, nejpoužívanější jsou XML a JSON, proto byly porovnány tyto jazyky. Oba JSON a XML mohou být použity pro příjem dat z webového serveru. JSON je stejně jako XML samo-popisný. Oba jazyky jsou hierarchické, jsou používány v mnoha programovacích jazycích, mohou být načteny pomocí XMLHttpRequest. XML musí být analyzován pomocí XML parseru. JSON může být analyzován pomocí standardní funkce JavaScriptu.

¹⁴ <https://www.programmableweb.com/wp-content/data-formats23.jpg>

JSON není jako XML protože:

- nepoužívá tagy,
- je kratší,
- je rychlejší pro čtení a zápis,
- může používat pole.

JSON je tedy lepší, protože jeho analýza je mnohem jednodušší než XML. Při zpracovávání dokumentů musíme u XML udělat tři kroky:

- načíst XML dokument,
- použít XML DOM k procházení dokumentu,
- extrahovat hodnoty a uložit je v proměnných.

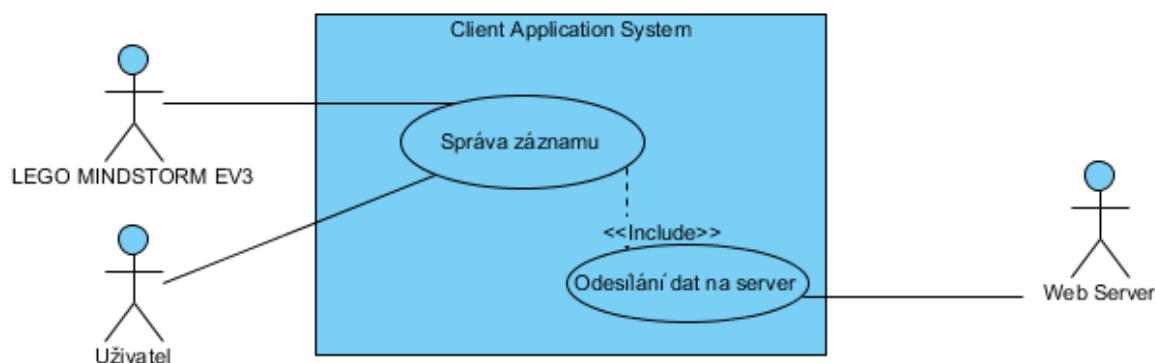
Oproti tomu u JSON jazyka:

- Načteme řetězec JSON,
- Parsujeme řetězec JSON.

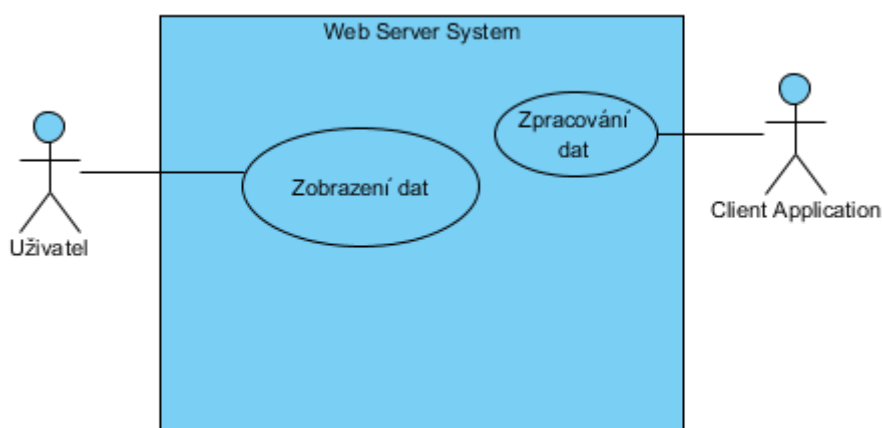
Z těchto důvodů byl vybrán jazyk JSON, který je výhodnější při přenosu dat.

Dále bylo nutné vzít v úvahu aplikační server pro zobrazení nashromážděných dat a jejich parsování, výběr tohoto serveru je již pouze na uživateli klientské aplikace. Pro prezentaci byl vytvořen server v ASP.NET, ASP.NET byl vybrán z důvodů výkonnosti například oproti PHP ale i ten přichází v potaz. PHP byl původně zamýšlen propojit s databázovým serverem MySQL. Nejlepší volbou v ASP.NET je WPF ASP.NET, který umožňuje vytváření dynamických webových stránek pomocí „chyt’ a pust’“ a pomocí modelu řízeným událostmi. Návrh designu a mnoho komponent umožňují rychle vytvářet sofistikované a výkonné servery řízené UI s přístupem k datům. Data se ukládají na databázový server MySQL, který podporuje skripty, je zdarma a má vysokou podporu komunity.

Zde můžete vidět návrh systému pomocí UML Use Case diagramu neboli digram případu užití, pomocí aplikace Visual Paradigm. Jelikož systém je komplikovaný a skládá se ze dvou aplikací.



Obr. 12: Use Case klientské aplikace



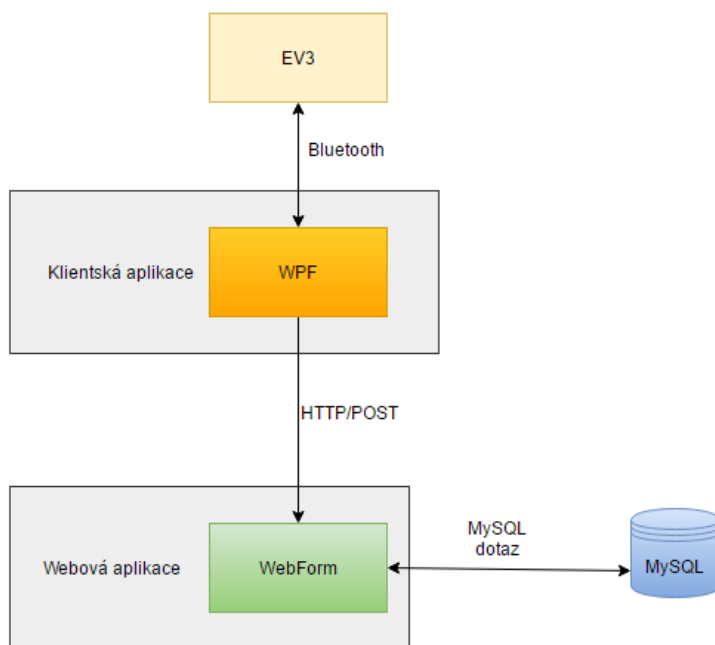
Obr. 13: Use Case Web Server

Uživatel by měl mít přístup jak do klientské desktopové aplikace, tak i do webové aplikace.

Klientská aplikace slouží pro správu záznamů, což implementuje komunikaci robota s aplikací a jedinečná identifikace robota, zobrazení hodnot senzorů a také výběr dat pro záznam a cíl určení. Odesílání dat funguje na metodě POST protokolu HTTP.

Webová aplikace na straně serveru je zodpovědná za parsování dat JSON získaných z protokolu HTTP. Následně jsou data odesílány v podobě MySQL dotazu na databázový

server. Aplikace dále nabízí zobrazení dat z databáze podle ID robota. Klientská část mé aplikace je vytvořená v programovacím jazyce C# a konkrétně byla použita šablona pro WPF aplikaci. Aplikace běží na operačním systému Microsoft Windows ve verzi 7 a novější. Tato klientská aplikace slouží především pro správu záznamu dat z robota, zobrazení dat v reálném čase. Aplikace nabízí připojení k robotovi LEGO MINDSTORM EV3 pomocí Bluetooth a umožňuje výběr cíle odesílání dat v textové formě, konkrétně v jazyce JSON. Aplikace nabízí odeslání dat buďto do souboru nebo na server a to pomocí protokolu HTTP. Server tvoří WebForm aplikace z prostředí ASP.NET, která zajišťuje parsování dat jazyka JSON do databáze MySQL. Také slouží pro zobrazení dat zpětně z databáze a to pomocí grafu a tabulky. MySQL databáze je určena převážně k ukládání dat ze záznamu z robota. Jeden takový záznam je tvořen unikátním kódem, který je vytvořen při každém připojení klientské aplikace k robotovi. Také se zde nachází ID záznamu, čas pořízení záznamu, hodnota na snímači a samozřejmě název samotného snímače. Pro lepší představu, níže můžete vidět diagram architektury.



Obr. 14: Diagram architektury

Tento model byl vytvořen z důvodu požadavku na aplikaci, pro zobrazení hodnot online.

5 POUŽITÍ NÁVRHOVÝCH VZORŮ V NAVRŽENÉM SOFTWARE

V práci byly použity vybrané návrhové vzory. Aby se aplikace nestala nepřehlednou, bylo využito pouze vzorů Tovární metoda a Singletone. Vzor Tovární metoda byl implementován do klientské aplikace. Tovární metodu volá konstruktor třídy, která má zajistit zpracování streamu¹⁵ dat a odeslání na server, nebo případné uložení do souboru. Tuto továrnu bylo nutné vytvořit pro možnost přidat další metody pro ukládání dat. Pro tento vzor byla vytvořena třída SenderFactory.cs (viz. Pascal notace). V C# se nepíše senderFactory, ale SenderFactory. Třída se stará o výběr konstruktoru vybraného typu odesílání dat.

```
public SendType createType(bool type) {  
    SendType typ = null;  
  
    if (type == true) {  
        typ = new SaveToHttp();  
    } else {  
        typ = new SaveToFile();  
    }  
    return typ;  
}
```

Obr. 15: Třída SenderFactory

Na předcházejícím obrázku je očividné, že třída počítá jen se dvěma hodnotami, protože datovým typem je bool. Toto omezení lze změnit přidáním další třídy, změnou datového typu, přidáním podmínky pro konstruktor na základě dalšího přepínače umístěného v GUI.

```
abstract public class SendType  
{  
    private string address;  
    private string text;  
    abstract public void send();  
  
    public void setAddress(string address)  
    {  
        this.adress = adress;  
    }  
    public string getAddress()  
    {  
        return this.adress;  
    }  
  
    public string getText()  
    {  
        return this.text;  
    }  
    public void setText(string text)  
    {  
        this.text = text;  
    }  
}
```

Obrázek 16: Třída SendType

Obě třídy SaveToFile a SaveToHttp dědí z abstraktní třídy SendType. Třída SendType předurčuje proměnné, SET & GET metody a metodu send(), která říká, jak budeme odesílat data.

Továrna je v tomto případě namístě, protože je to vytvářecí návrhový vzor, který si umí poradit s problémem vytváření objektů. Návrhový vzor Továrna je také specifický tím že, je specifikován v interface a implementován v podtřídách. Raději přepisujeme třídu,

¹⁵ Stream: nepřetržitý davový tok.

než voláme nový konstruktor.

```
public class SaveToHttp : SendType
{
    public override void send() {
        try
        {
            System.Net.HttpWebRequest request = (System.Net.HttpWebRequest)System.Net.WebRequest.Create(getAdress());

            request.Method = "POST";
            request.ContentType = "application/json";
            request.ContentLength = getText().Length;

            using (var streamWriter = new StreamWriter(request.GetRequestStream()))
            {
                streamWriter.Write(getText());
                streamWriter.Close();

                var httpResponse = (HttpWebResponse)request.GetResponse();
                using (var streamReader = new StreamReader(httpResponse.GetResponseStream()))
                {
                    var result = streamReader.ReadToEnd();
                }
            }
        }
        catch (Exception ex){
            Debug.WriteLine("Chyba při odesílání na adresu : "+ex);
        }
    }
}
```

Obr. 17: Příklad třídy SaveTo

Tento obrázek je třída SaveToHttp.cs, která odesílá předaný *String* na požadovanou adresu HTTP pomocí metody POST. Adresu získá pomocí metody *getAdress()* a *String* získá pomocí *getText()*.

Další důležitý a všeobecně známý návrhový vzor použitý v práci je Singleton. Tento vzor se využívá v navázání spojení mezi ASP.NET aplikací a MySQL databází. Více informací o tomto vzoru naleznete v kapitole 2.10.

6 IMPLEMENTACE

Jak již bylo zmíněno, aplikace by měla být schopna odesílat data online. Proto je popis kvůli složitosti a přehlednosti řešení rozdělen do tří kroků, Příprava prostředí, Klientská část, Serverová část.

Příprava prostředí

Nyní předpokládám, že model architektury aplikace je již vytvořený a zbývá implementovat danou ideu. Pro práci je potřeba:

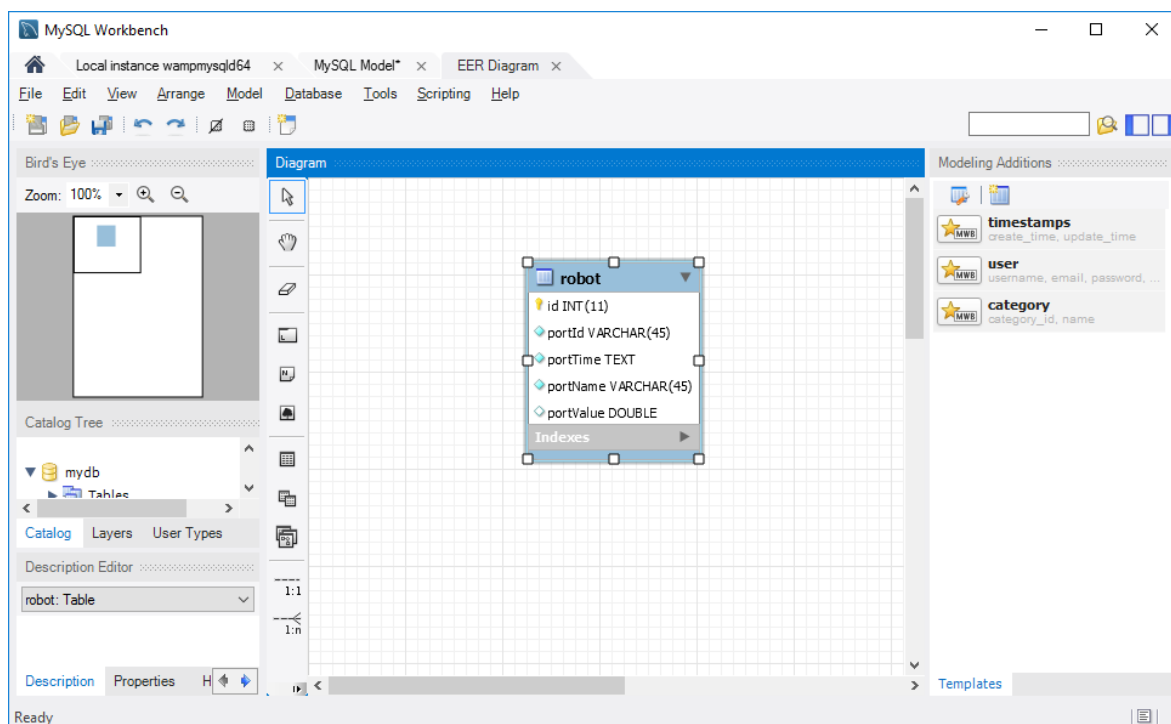
- vývojové prostředí (IDE), se kterým můžete vytvářet moderní aplikace v prostředí Microsoft Windows (Visual Studio Community 2015 nebo novější),
- přístup k internetu,
- MySQL databázi MySQL 5.7.14 a PhpMyAdmin 4.6.4, nebo novější.

Použité vývojové prostředí Visual Studio Community 2015 ve verzi 14.0.25123 Update 2. Spolu s vývojovým prostředím se vážou i instalované komponenty v programu Visual Studio. Nejdůležitější instalované části aplikace jsou tyto:

- C#,
- ASP.NET Web Framework and Tools,
- Javascript Project/Service 2.0,
- NuGet Package Manager.

MySQL databázi a PhpMyAdmin byl nainstalován v rámci instalace aplikace WampServer ve verzi 3.0.6 64 bit x64, tato aplikace nabízí balíček pro vývoj webových aplikací, obsahuje také Apache verze 2.4.23 a PHP 5.6.25/7.0.10.

V PhpMyAdmin si musíme předem nastavit vlastní přihlašovací heslo a vytvořit tabulku pro ukládání dat do databáze např. „robot“ a tabulku „ev3“. K vytvoření tabulky a pro přehlednost nám nejlépe poslouží aplikace MySQL Workbench, ve které se připojíme k naší databázi MySQL a vytvoříme si jednoduchou tabulku.



Obr. 18: MySQL Workbench

Výstupní hodnota ze senzorů je vždy číselná hodnota, proto přidáme sloupec „*portValue*“ a nastavíme jej jako double, kvůli senzorům, které vracejí desetinné číslo. Název portu budeme ukládat pod sloupec „*PortName*“ jako *varchar(45)*, čas záznamu pod název „*portTime*“ jako text a ID portu pod název „*portId*“ jako *varchar(45)*, který bude sloužit pro ukládání identifikačního jedinečného kódu zařízení. Nakonec přidáme *ID* jako *int(11)* a primární klíč, nastavíme u něj hodnotu na *AUTO_INCREMENT*.

	#	Název	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Další
<input type="checkbox"/>	1	id	int(11)			Ne	Žádná	AUTO_INCREMENT
<input type="checkbox"/>	2	portId	varchar(45)	utf8_general_ci		Ne	Žádná	
<input type="checkbox"/>	3	portTime	text	utf8_general_ci		Ne	Žádná	
<input type="checkbox"/>	4	portName	varchar(45)	utf8_general_ci		Ne	Žádná	
<input type="checkbox"/>	5	portValue	double			Ano	NULL	

Obr. 19: PhpMyAdmin tabulka

6.1 Klientská část

Pro klientskou část aplikace si jako první vytvořím nový projekt WPF Application. Pro komunikaci s robotem LEGO MINDSTORM EV3 je potřeba nastavit reference na API knihovnu, která zprostředkovává komunikaci mezi robotem a vytvářenou aplikací. Knihovnu nalezneme při použití NuGet pod názvem Lego.Ev3.Desktop a přidáme ji do projektu jako referenci. Je nutné také říci, že .NET API pro EV3 obsahuje celkem tři knihovny, které jsou pojmenovány:

- Lego.Ev3.Desktop,
- Lego.Ev3.Phone,
- Lego.Ev3WinRT.

Pomocí referencí můžeme vytvořit objekt kostky a začít komunikovat tak s EV3 programovací kostkou. Nicméně, musíme projít instanci objektu, který implementuje ICommunication rozhraní ke kostce konstruktoru, který určí, jak bude knihovna mluvit s programovací kostkou. Můžeme toho dosáhnout díky využití následujících variant. **Desktop** vyžaduje Bluetooth (vyžaduje COM port během párování dvou zařízení) nebo USB nebo Internet (vyžaduje vložení IP adresy do programovací kostky). **Phone** vyžaduje Bluetooth nebo Internet. **WinRT** vyžaduje Bluetooth nebo USB port nebo Internet.

K vytvoření nové kostky napíšeme kód uvedený níže ke komunikaci přes Bluetooth.

```
BluetoothCommunication b = new BluetoothCommunication(cPorts.SelectedItem.ToString());  
brick = new Brick(b);  
brick.BrickChanged += OnBrickChanged;  
await brick.ConnectAsync();
```

Obr. 20: Bluetooth komunikace

Programovací kostka obsahuje BrickChanged událost, přes kterou můžeme dostávat události při každé změně na programovací kostce. Příkladem může být změna hodnoty senzorů.

Na programovací kostce taktéž nalezneme objekty Portů. Tyto mapy fyzických portů programovací kostky EV3 obsahují hodnoty, v jakém režimu jsou v určitém momentu spuštěny apod. K těmto událostem se můžete připojit, viz. Obr. 21.

```

void OnBrickChanged(object sender, BrickChangedEventArgs e)
{
    BlockOne.Text = e.Ports[InputPort.One].SIValue.ToString();
    BlockTwo.Text = e.Ports[InputPort.Two].SIValue.ToString();
    BlockThree.Text = e.Ports[InputPort.Three].SIValue.ToString();
    BlockFour.Text = e.Ports[InputPort.Four].SIValue.ToString();
    BlockA.Text = e.Ports[InputPort.A].SIValue.ToString();
}

```

Obr. 21: Události EV3

Pro nalezení správného komunikačního COM portu je nutno nalézt Sériový port připojeného zařízení EV3, který se nachází pod názvem BTHENUM. Pod tímto názvem se nachází všechna připojení od EV3. Dále bylo nutné vytvořit způsob pro odesílání dat. Pro více způsobů odesílání dat se výborně hodí návrhový vzor Tovární metoda, která se stará o výběr metody pro odesílání data na základě informace z GUI, který je předán v podobě parametrů viz kapitola o návrhových vzorech. Pro správné vytvoření záznamu o stavu kostky EV3 byla vytvořena třída Value, která představuje parametry získané při sběru dat ze senzorů robota. Tyto hodnoty jsou dále odesílány k dalšímu zpracování.

```

class Value
{
    public string portId;
    public string portName;
    public string portTime;
    public string portValue;
}

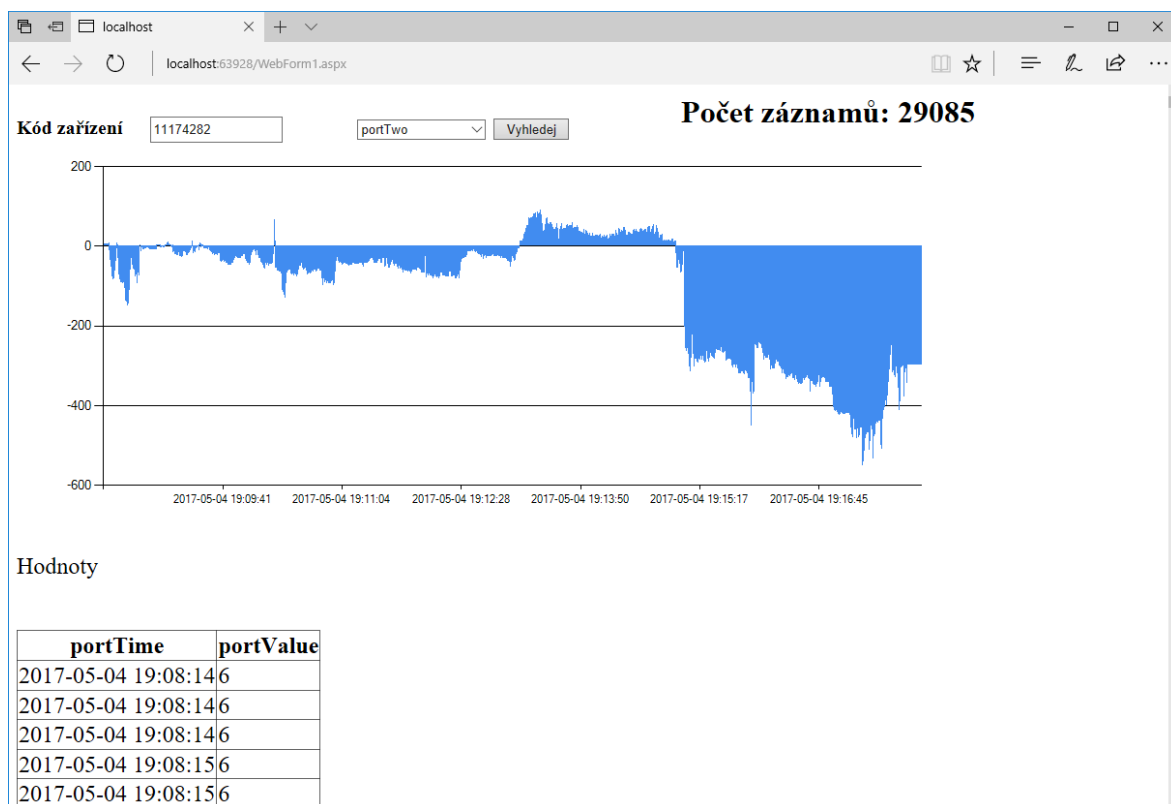
```

Obr. 22: Třída Value

Kvůli zvýšené zátěži při odesílání dat pro každý port neboli senzor zvlášť, se odesílají data ze všech portů najednou.

6.2 Serverová část – univerzální možnost zpracování dat

Tuto část práce je vytvořena pro prezentaci dat online. Zkušený uživatel ovšem může použít jiný server pro záznam. Původně bylo zamýšleno nad použitím PHP pro aplikaci na serveru, ale toto rozhodnutí bylo vyvráceno tím, že PHP by patrně nezvládalo tak velký nápor dat a souběžný parsing.



Obr. 23: Náhled serverové aplikace

Aplikace je založená na ASP.NET Web Forms. Web Forms využívají soubor s příponou .aspx, který integruje programátorův kód a design grafického rozhraní. Jelikož klientská aplikace odesílá data ve formě např.:

```
{'data':[{ "portId": "63927133", "portName": "PortThree", "portTime": "2017-03-31
22:09:22", "portValue": "75,4"}, { "portId": "63927133", "portName": "PortFour", "portTime":
"2017-03-31 22:09:22", "portValue": "204,9"}]}
```

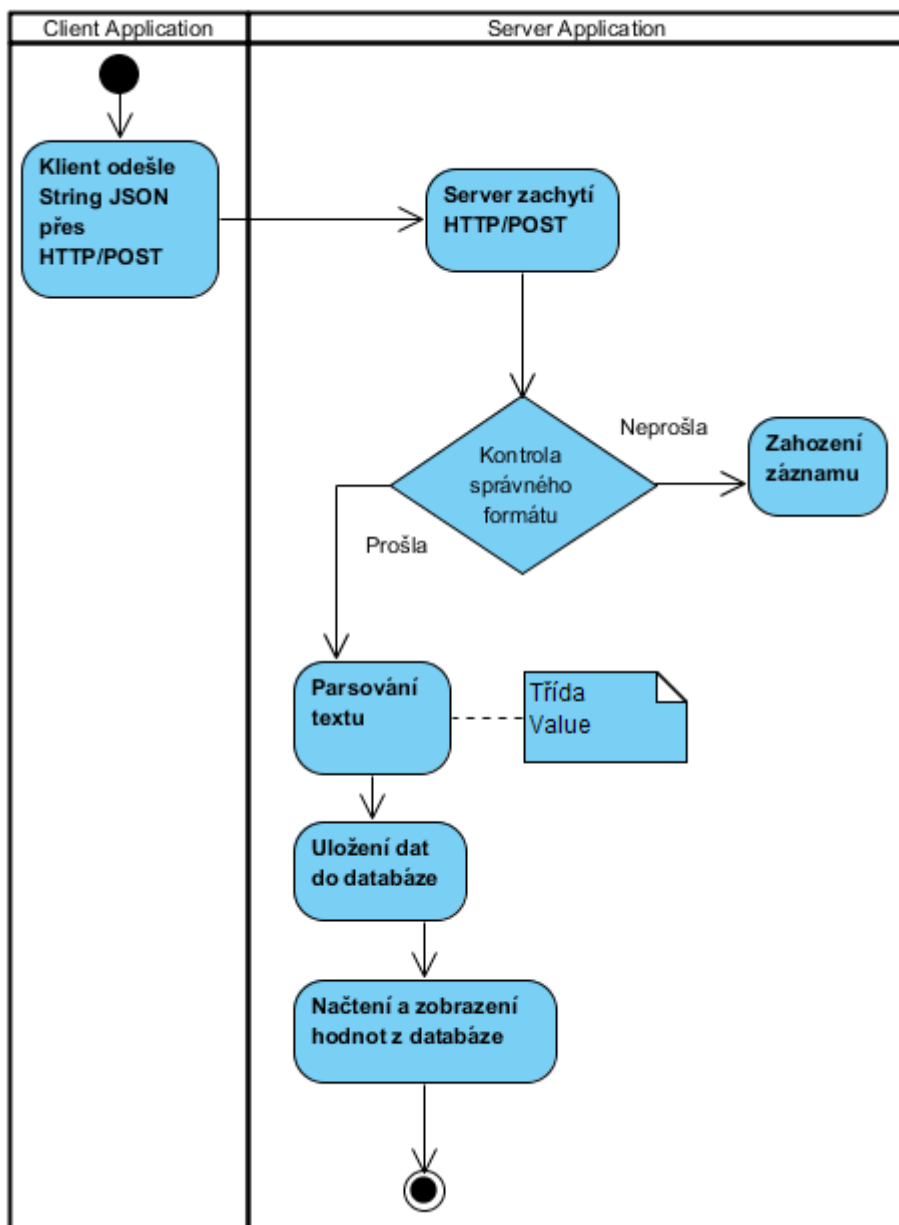
Musí se tyto hodnoty parsovat do listu hodnot upravené třídy Value, který byl vysvětlen v klientské části. Samotné parsování si programátor může naprogramovat sám, nebo může využít jednu z uvedených knihoven pod tímto odkazem¹⁶. V případě zvolení první možnosti, musí metoda procházet *String* a vyhledávat data za pomoci speciálních znaků, typických pro jazyk JSON.

```
{
    public List<Item> data;
    public class Item
    {
        public string portId;
        public string portName;
        public string portTime;
        public double portValue;
    }
}
```

Obrázek 24: Třída Value

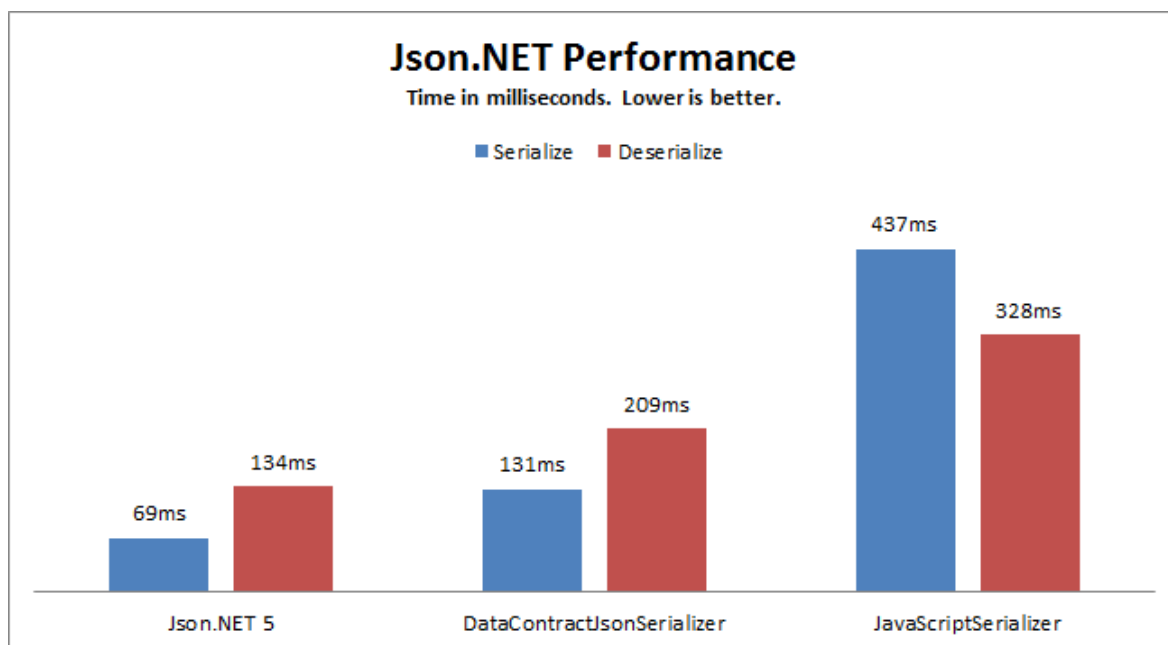
¹⁶ <http://www.json.org/json-cz.html>

Pro samotné čtení dat z http požadavku bylo využito `System.IO` pro `StreamReader` a pro parsování jednotlivých řádků je nutno přidat referenci na knihovnu *Newtonsoft.Json*, která je vhodná pro práci s formátem JSON z hlediska výkonu.



Obr. 25: Proces zpracování dat

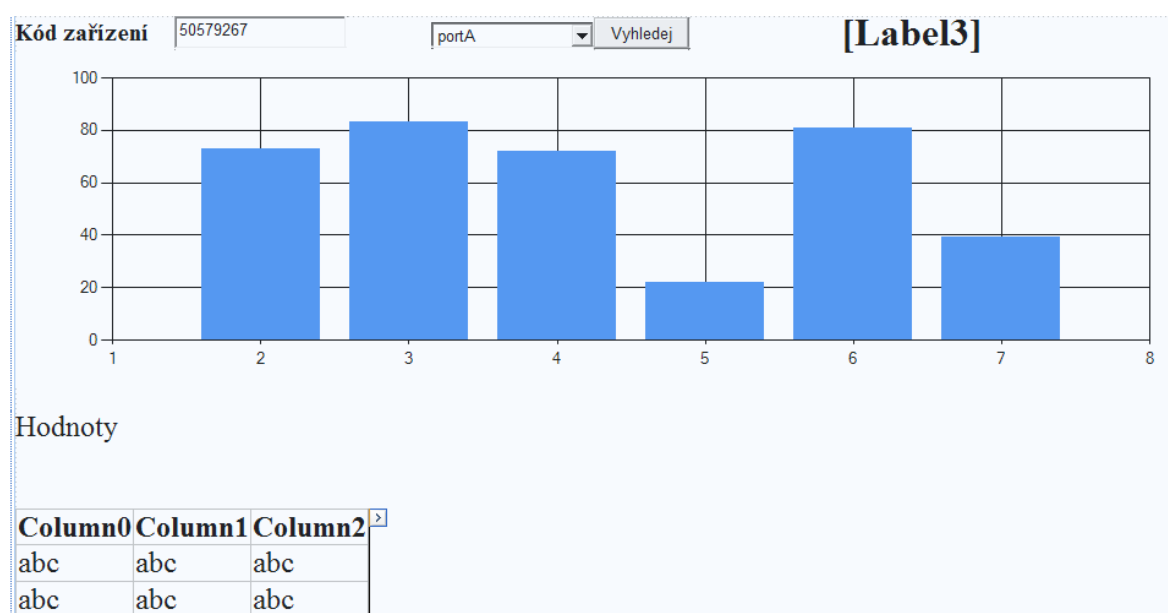
Pro proces zpracování dat byla navržena metodika, viz Obr. 25, která také byla využita při implementaci.



Obr. 26: Newtonsoft Json.NET 5, převzato z ¹⁷

Dále v této práci je využit Singleton návrhový vzor, který využívá k navázání spojení s databází MySQL referenci na MySQL.Data což je .NET knihovna pro komunikaci s MySQL serverem.

K návrhu GIU slouží sekce Design u souboru .aspx, kde se zobrazuje graf, tabulka hodnot, aktuální počet záznamů v databázi a výběrová pole pro zobrazení požadovaných hodnot.

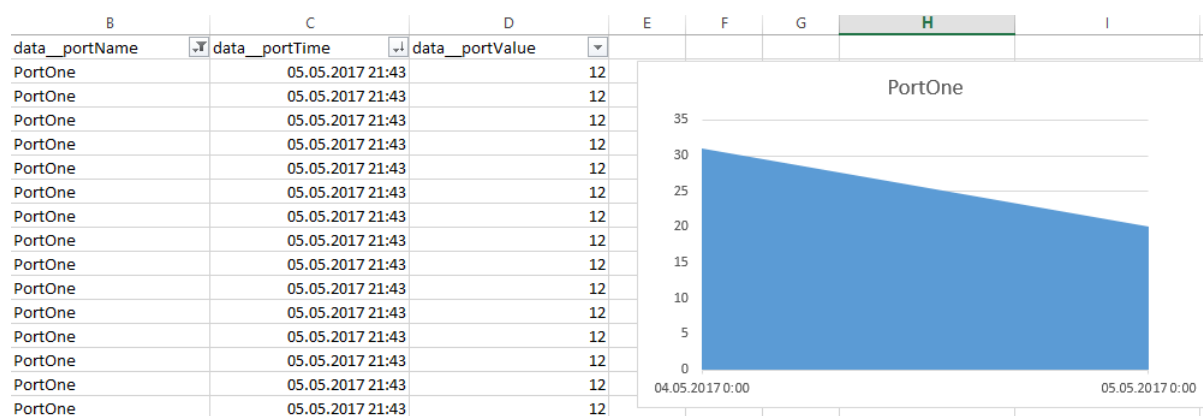


Obr. 27: Náhled GUI Web Aplikace

¹⁷ <http://www.newtonsoft.com/content/images/jsonperformance.png>

Jak můžete vidět na obrázku výše, uživatel musí nejprve zadat kód zařízení, který mu byl dříve zobrazen v klientské aplikaci, dále si uživatel může vybrat port, jehož hodnoty chce zobrazit a následně stiskne tlačítko Vyhledej. Následně se uživateli zobrazí graf hodnot a tabulka, která tyto hodnoty reprezentuje. V pozadí je spuštěn JavaScript, který s využitím jQuery aktualizuje hodnotu v komponentě Label3, kterou můžete vidět na obrázku výše.

Data JSON ze souboru Log.txt lze zobrazit také v programu Microsoft Excel, viz následující obrázek.



Obr. 28: Výstup v programu Excel

Pro periodické čtení záznamů z databáze bylo využito asynchronního JavaScriptu a XML pod názvem AJAX.

```

$(document).ready(function () {
    getRecordAjax();

    function getRecordAjax(){
        $.ajax({
            type: "POST",
            url: "WebForm1.aspx/ajaxDBNumber",
            contentType: "application/json; charset=utf-8",
            dataType: "json",
            success: function (data) {
                $("#Label3").text("Počet záznamů: " + data.d);
            },
            error: function () { }
        });
        return false;
    }

    setTimeout(function () {
        getRecordAjax();
    }, 15000);
});

```

Obr. 29:AJAX

Požadavek na zobrazení počtu záznamů by byl uživatelsky nepřívětivý a muselo by znovu dojít k načítání celé stránky, proto AJAX řeší tento problém sám. Zprvu bylo trochu složité implementovat toto rozšíření pro aplikaci, bylo nutné hledat informace na různých fórech a webu stackoverflow.com, kde si programátoři pomáhají navzájem, avšak málo které řešení je vhodné pro konkrétní implementaci. Aktualizace zmíněné komponenty pro zobrazení záznamu byla nastavena na patnáct sekund, hlavně z důvodu testování.

7 TESTOVÁNÍ

V první řadě se muselo ověřit, zda klientská aplikace opravdu zobrazuje data, převzatá ze senzorů robota. To se realizovalo pomocí jednoduchého testu na kontrolu vzdálenosti se Infrared senzorem, který vrací přibližné hodnoty v centimetrech. Testoval se také například Color Sensor, který měří intenzitu světla odráženého zpět z lampy vyzařující červené světlo. Senzor používá škálu od 0 (velmi tmavá) do 100 (velmi světlá). Další připojitelná zařízení k robotovi jsou motory, které vrací úhel natočení od počátku záznamu, takže po jedné otáčce vrací hodnotu 360, po dvou vrací 720, nebo může vracet také mínusové hodnoty po otáčení opačným směrem. Díky tomuto způsobu záznamu hodnot lze hodnoty jednoduše vyvést do grafu.

Testování naměřených dat a funkčnosti se prováděla za pomoci Unit testů. Nejprve se musí vytvořit Unit test k testovanému modulu, dále bylo nutné nainstalovat NUnit 3.6.1 balíček pomocí příkazu `Install-Package NUnit`, v konzoli `Package Manager`. Zapotřebí bylo nainstalovat mockovací framework pro nastavení předpokládaného chování na základě vstupu a díky tomu můžeme regulérně testovat aplikační kód vybrané testované metody např.: *Metoda()*.

Toto testování bylo využito například u testování metody pro vytvoření výstupu v jazyce JSON. Kdy se muselo správně sestavit data, tak aby odpovídala správnému zápisu a slučovala záznamy z více senzorů.

```
[TestMethod]
public void TestMethod1()
{
    Class1 recordMethod = new Class1();
    Value portOne = new Value();
    Value portTwo = new Value();
    portOne.portId = "10";
    portOne.portName = "portOne";
    portOne.portTime = "10.10.2015";
    portOne.portValue = "10001";

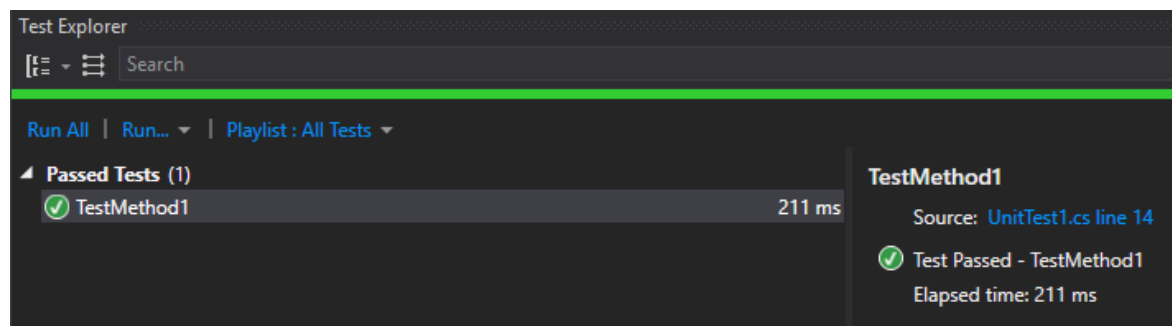
    portTwo.portId = "10";
    portTwo.portName = "portOne";
    portTwo.portTime = "10.10.2015";
    portTwo.portValue = "10001";

    String result = recordMethod.recording(portOne, portTwo);
    String optimalResult = "{"data":[{"portId":"10","portName":"portOne","portTime":"10.10.2015","portValue":"10001"}]}";

    NUnit.Framework.Assert.AreEqual(optimalResult, result);
    NUnit.Framework.Assert.AreEqual(null, null);
}
```

Obr. 30: Unit test

Do metody nahrajeme požadovaná data a porovnááme je s očekávaným výsledkem. Výsledek Unit testu může vypadat takto:



Obr. 31: Výsledek Unit testu

Testy byly vytvářeny hlavně v případech, kdy se kontrolovala funkčnost metod, ve kterých se využívají knihovny, abychom si byli jistí výstupem z daných metod.

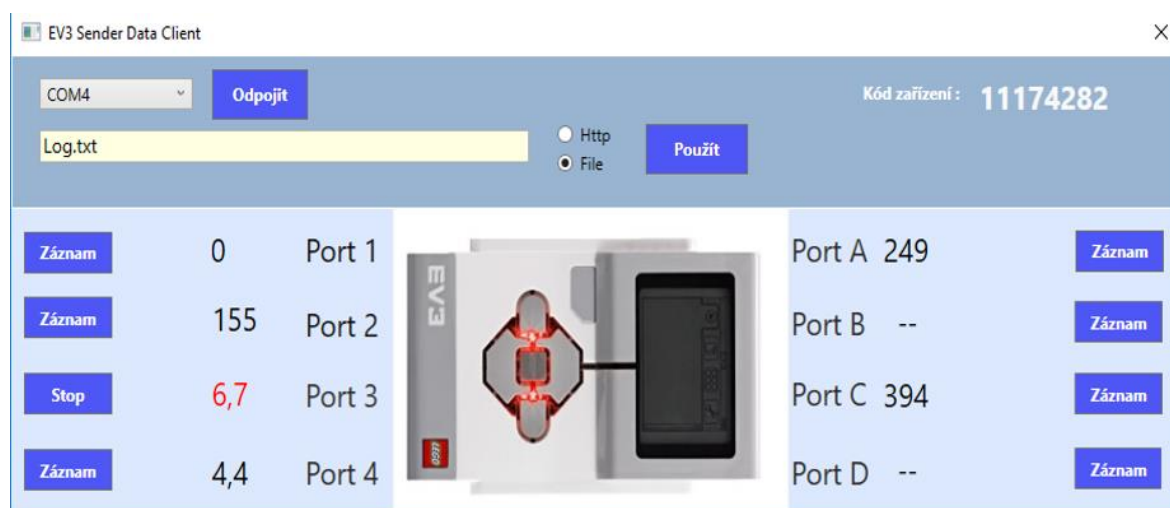
Realizovat performance testy na takovéto aplikaci nemá smysl (má to smysl až když se musí dodržovat SLA¹⁸, kde je výslovně napsáno, že aplikace má odpovídat daným požadavkům.

¹⁸ SLA - Service Level Agreement

8 POUŽITÍ APLIKACE

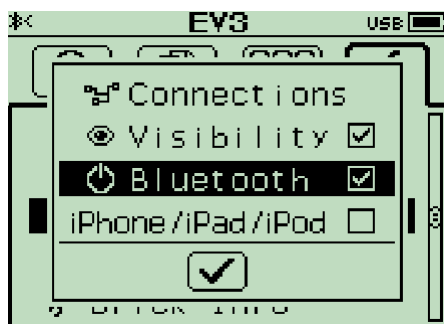
Požadavky

- Pro správnou funkčnost spuštění klientské aplikace, je třeba mít osobní počítač, laptop, tablet nebo jiné zařízení s operačním systémem Windows 7, nebo novější verze.
- Je potřeba mít Bluetooth adaptér.
- Pro zasilání dat přes HTTP protokol je potřeba mít .NET Core a MySQL databázi na lokálním serveru, nebo přístup k internetu se spuštěnou webovou aplikací.



Obr. 32: Klientská aplikace GUI

Abychom se mohli připojit k robotovi, musíme nejprve robota spárovat a naším počítačem pomocí Bluetooth. Nejprve zkontrolujeme, zda má robot zpřístupněnou Bluetooth komunikaci. Komunikaci naleznete v záložce Nastavení/Connections, kterou můžete vidět na následujícím obrázku. Hodnota -- , kterou můžete vidět na Obr. 32, znamená chybějící číselná hodnota a taky, že do daného portu není připojen žádný senzor.



Obr. 33: EV3 Nastavení Bluetooth, převzato z ¹⁹

Nyní, když je robot připraven, stačí robota „spárovat“ přes komunikační Bluetooth kanál s Vaším zařízením. Pokud „spárování“ proběhlo v pořádku můžeme spustit klientskou aplikaci.

Po spuštění klientské aplikace se zobrazí GUI, které můžete vidět na obrázku č.18 viz výše. V levém horním rohu se nachází drop-down list s výpisem prvků, kde každý prvek značí odchozí komunikační COM²⁰ port. Dále stačí stisknout tlačítko Připojit a robot LEGO Mindstorm EV3 ohlásí připojení zvukovým oznámením.

Dále jako uživatel můžete vybrat cílové umístění dat, která se budou nahrávat z robota. Cílové umístění zadáte do předem vyplněného pole, které se nachází pod výběrem COM portu. Adresa může být adresa jakéhokoli webového serveru který je schopen přijmout a zpracovat datový tok v HTTP protokolu.

Webová aplikace přiložená v této práci je samozřejmě k dispozici jako základní aplikace pro zobrazení zaznamenávaných dat. Tato aplikace slouží pro umístění na server, nebo lokální server. Pro nasazení webové aplikace na server je nutné, nejen kopírování souborů aplikace z jednoho místa na server, ale je možné že bude nutná změna nastavení souboru Webconfig, která musí být cílovém prostředí odlišná. Při použití vlastní databáze pod vlastním názvem, je nutné přepsat databázový připojovací řetězec, který se nachází pod názvem s touto proměnnou `CONNECTION_STRING = (@"Server=localhost;Database=ev3;Uid=root;Pwd=root;";)`, který se nachází ve třídě MySQLUtil. GUI je zobrazen výše na Obr. 23, kde se nachází pole pro zadání kódu zařízení a výběr daného portu robota, na kterém je/byl připojen měřený senzor. Po kliknutí na tlačítko vyhledej, se zobrazí hledaná data.

¹⁹ https://www.mathworks.com/help/supportpkg/legomindstormsev3io/ug/bt_3.png

²⁰ COM port : Rozhraní sériového portu, který slouží pro připojení vnějších zařízení.

ZÁVĚR

Při funkčním testování prošlo 100% testů a aplikace je otestovaná připravena k použití. Největším problémem při implementaci daného řešení byla práce s JSON a jeho správné sestavení/parsování. Díky využití Unit testů se problémy vyřešily daleko rychleji a předešlo se nechtěným chybám. Také se vyskytl problém, který se nemohl dlouho vyřešit a to při možnosti použití AJAX pro změny prvků uživatelského prostředí na webové aplikaci .aspx. Pro chod aplikace zdarma bych vyměnil WPF server za PHP, který je sice pomalejší, ale za to je jednodušší manipulace s daty v MySQL a JSON. Možnosti vylepšení se zde nachází hned několik, například implementovat výstup v XML formátu, nebo ovládání robota přímo z klientské aplikace k tomu napomáhá použitý vzor Tovární metoda, nebo dokonce z aplikace na serveru. Dále je možno přidat další druhy připojení k robotovi přes USB a Wi-Fi. Zdrojové kódy jsou volně přístupné zájemcům a prezentovány na serveru GitHub na této adrese ²¹. Zdrojové kódy, dokumentace a výsledná aplikace budou také využity k výuce v předmětu KIP/APLUI.

²¹ <https://github.com/MensikJan/DataCollecting>

SEZNAM POUŽITÉ LITERATURY

1. **Patricia Pulliam Phillips** and **Cathy A. Stawarski**, Data Collection: Planning for and Collecting All Types of Data 1st Edition, ISBN-13: 978-0787987183, ISBN-10: 0787987182. Dostupné z: <https://goo.gl/EZJsyV>
2. **LeJOS EV3**. LeJOS, Java for Lego Mindstorms / EV3LeJOS, Java for Lego Mindstorms / EV3 [online]. 1997-2009 [cit. 2016-08-29]. Dostupné z: <http://www.lejos.org/ev3.php>
3. **THE LEGO GROUP**. Uživatelská příručka Lego MindStorms EV3 [online]. 2013 [cit. 2016-08-29]. Dostupné z: <http://www.lego.com/cs-cz/mindstorms/downloads/user-guides/cs/>
4. **Luis Cruz**, Humanoid Robot Nao: Developing Behaviours for Soccer Humanoid Robots, LAP Lambert Academic Publishing, 2013. Dostupné z: <https://goo.gl/gVK1Vi>
5. **OLIVEIRA, Gerardo**, et al. Environment Mapping using the Lego MindstormsNXT and leJOS NXJ [online].[s.l.]:[s.n.], 2007]. Dostupné z : <http://paginas.fe.up.pt/ei04085/psite/LejosPaper.pdf>
6. **W. LEW, Michael; B. HORTON, Thomas; S. SHERRIFF, Mark**. Using LEGO MINDSTORMS NXT and LEJOS in an Advanced Software Engineering Course [online].[s.l.]:[s.n.], 2009 [cit. 2017-02-1].
7. **C Sharp**. In: What's New for Visual C# in Visual Studio 2012 [online]. Microsoft Developer Network, [cit. 2013-08-02] Dostupné z: <https://msdn.microsoft.com/en-us/library/hh156499.aspx>
8. **Microsoft Visual Studio**. Visual Studio 2005 SDK. Visual Studio Development Environment Model [online]. Microsoft, [cit. 2008-01-01] In: Dostupné z: [https://msdn.microsoft.com/en-us/library/bb165114\(VS.80\).aspx](https://msdn.microsoft.com/en-us/library/bb165114(VS.80).aspx)
9. Microsoft. C# Programming Guide [online]. [cit. 2017-05-02]. Dostupné z: [https://msdn.microsoft.com/en-us/library/0yw3tz5k\(VS.80\).aspx](https://msdn.microsoft.com/en-us/library/0yw3tz5k(VS.80).aspx)
10. Microsoft. Msdn.microsoft.com [online]. [cit. 2017-05-02]. Dostupné z: [https://msdn.microsoft.com/en-us/library/dd30h2yb\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/dd30h2yb(v=vs.110).aspx)

11. **Microsoft**. VisualStudio.com [online]. [cit. 2017-05-02]. Dostupné z: <https://www.visualstudio.com/cs/?rr=https%3A%2F%2Fwww.google.cz%2F>
12. **Microsoft**. Asp.net [online]. [cit. 2017-05-02]. Dostupné z: <https://www.asp.net/>
13. **ITNETWORK**, Úvod do ASP.NET. [online]. [cit. 2017-05-02]. Dostupné z: <https://www.itnetwork.cz/csharp/asp-net/tutorial-uvod-do-asp-dot-net>
14. **Oracle**. MySQL [online]. [cit. 2017-05-02]. Dostupné z: <https://dev.mysql.com/doc/relnotes/mysql/5.7/en/>
15. **Json.org**. JSON [online]. [cit. 2017-05-02]. Dostupné z: <http://www.json.org/json-cz.html>
16. **Dušan Janovský**. Jak psát web [online]. [cit. 2017-05-02]. Dostupné z: <https://www.jakpsatweb.cz/javascript/javascript-uvod.html>
17. **Segue Technologies Inc.**. AJAX. [online]. [cit. 2017-05-02]. Dostupné z: <http://www.seguetech.com/ajax-technology/>
18. **JQuery Foundation**. JQuery.com [online]. [cit. 2017-05-02]. Dostupné z: <http://api.jquery.com>
19. **Vojtěch Hordejčuk**. Návrhové vzory. [online]. [cit. 2017-05-02]. Dostupné z: <http://voho.eu/wiki/navrhovy-vzor/>
20. **Fielding, et al**. Hypertext Transfer Protocol. Network Working Group [online]. [cit. 2017-05-02]. Dostupné z: <https://tools.ietf.org/html/rfc2616>

SEZNAM OBRÁZKŮ

Obr. 1: Programovatelná kostka, převzato z	11
Obr. 2: Color Sensor, převzato z	14
Obr. 3: Touch Sensor, převzato z	14
Obr. 4: Infrared Sensor, převzato z	15
Obr. 5: Temperature Sensor, převzato z	15
Obr. 6: Factory method, převzato z	29
Obr. 7: Náhled ikon, převzato z	31
Obr. 8: Náhled Aplikace EV3 Programming Software	32
Obr. 9: CodePlex náhled aplikace, převzato z	33
Obr. 10: PFremote náhled aplikace, převzato z	34
Obr. 11: Datové formáty pro přenos v reálném čase, převzato z	36
Obr. 12: Use Case klientské aplikace	38
Obr. 13: Use Case Web Server	38
Obr. 14: Diagram architektury	39
Obr. 15: Třída SenderFactory	40
Obr. 16: Třída SendType	40
Obr. 17: Příklad třídy SaveTo	41
Obr. 18: MySQL Workbench	43
Obr. 19: PhpMyAdmin tabulka	43
Obr. 20: Bluetooth komunikace	44
Obr. 21: Události EV3	45
Obr. 22: Třída Value	45
Obr. 23: Náhled serverové aplikace	46
Obr. 24: Třída Value	46
Obr. 25: Proces zpracování dat	47
Obr. 26: Newtonsoft Json.NET 5, převzato z	48
Obr. 27: Náhled GUI Web Aplikace	48
Obr. 28: Výstup v programu Excel	49
Obr. 29: AJAX	50
Obr. 30: Unit test	51

Obr. 31: Výsledek Unit testu	52
Obr. 32: Klientská aplikace GUI	53
Obr. 33: EV3 Nastavení Bluetooth, převzato z	54

SEZNAM PŘÍLOH

Přílohou této práce je CD obsahující zdrojové kódy klientské aplikace a webové aplikace v souboru Zdrojové_kódy.zip, samotná klientská aplikace se zde nachází pod názvem EV3_Sender_Data_Client.exe a knihovnu tříd webové aplikace lze nalézt ve složce pod názvem EV3_Sender_Data_Server. Disk obsahuje také kopii této práce ve formátu PDF s názvem Jan_Menšík.pdf.