



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

UŽIVATELSKÉ ROZHRANÍ PRO SYSTÉM ANDROID PRO OVLÁDÁNÍ PLICNÍHO SIMULÁTORU

USER INTERFACE FOR ANDROID SYSTEM TO CONTROL LUNG SIMULATOR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARKÉTA VRÁNOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ SEKORA

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav biomedicínského inženýrství

Bakalářská práce

bakalářský studijní obor

Biomedicínská technika a bioinformatika

Studentka: Markéta Vránová

ID: 147524

Ročník: 3

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Uživatelské rozhraní pro systém Android pro ovládání plicního simulátoru

POKYNY PRO VYPRACOVÁNÍ:

1) Seznamte se s modulem mechanického plicního simulátoru. 2) Navrhněte implementaci pro bezdrátové ovládání řídicí jednotky simulátoru realizované na bázi LabVIEW pomocí uživatelského rozhraní pro mobilní zařízení se systémem Android. Rozhraní musí umožňovat pozorování a aktivní změnu parametrů dýchání mechanického plicního simulátoru. 3) Realizujte uživatelské rozhraní na obou stranách, navrhněte protokol bezdrátového připojení. 4) Správnost navrženého řešení otestujte. Téma bude řešeno ve spolupráci s Fachhochschule Technikum Wien v rámci mezinárodního projektu ALVEOPIC. Studentka se bude podílet na řešení dílčích částí projektu dle zadání konzultanta.

DOPORUČENÁ LITERATURA:

[1] BITTER, Rick, Taqi MOHIUDDIN a Matt NAWROCKI. LabView advanced programming techniques. 2nd ed. Boca Raton: CRC Press, c2007, 499 s. ISBN 08-493-3325-3.

[2] SILVA, Vladimír. Advanced Android 4 games. Berkeley: Apress, c2012, xx, 280s. ISBN 978-1-4302-4059-4.

Termín zadání: 10.2.2014

Termín odevzdání: 30.5.2014

Vedoucí práce: Ing. Jiří Sekora

Konzultanti bakalářské práce: Matthias Frohner, M.Sc.

prof. Ing. Ivo Provazník, Ph.D.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce se zabývá vytvořením dálkového ovládní plicního simulátoru sestaveného v prostředí LabVIEW. Simulátor je připojen k počítači a řízen pomocí aplikace LabVIEW. Komunikace probíhá mezi LabVIEW a systémem Android pomocí internetu, konkrétně přes službu webservis. Funkce plicního simulátoru je v práci popsána a práce dále uvádí různá řešení, jakým způsobem lze model vzdáleně ovládat. Výběr nejlepšího řešení je odůvodněn a toto řešení je detailněji popsáno. Je diskutován i způsob zprovoznění služby webservis pro komunikaci na straně aplikace LabVIEW v počítači.

KLÍČOVÁ SLOVA

Simulátor plic, dálkové ovládní, LabVIEW, systém Android, služba webservis, sdílené proměnné

ABSTRACT

This project deals with the creation of the control system of the lung simulator build in program LabVIEW. The Simulator is connected to the computer and controlled by using the LabVIEW application. The communication between LabVIEW and Android system is provided with the Internet, with the Web Service. The lung simulator is described in the project and the project includes also various solutions, how the simulator can be controlled. The selection of the best solution is justified by, and this solution is described in more detail. The connection of the application in LabVIEW to the Web Service is created and described.

KEYWORDS

Lung simulator, user interface controlling the simulator, LabVIEW, Android system, web service, shared variables

VRÁNOVÁ, M. *Uživatelské rozhraní pro systém Android pro ovládání plicního simulátoru*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav biomedicínského inženýrství, 2014. 37 s. Bakalářská práce. Vedoucí práce: ing. Jiří Sekora.

PROHLÁŠENÍ

Prohlašuji, že svoji bakalářskou práci na téma Uživatelské rozhraní pro systém Android pro ovládání plicního simulátoru, jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne 30. 5. 2014.....

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce ing. Jiřímu Sekorovi za ochotnou pomoc a dobré připomínky a vedení mé bakalářské práce.

V Brně dne 30. 5. 2014.....

.....

(podpis autora)

OBSAH

Úvod	8
1 Modul mechanického plicního simulátoru	9
1.1 Konstrukce	10
1.2 Komunikace a řízení	11
2 Propojení labview se systémem android	13
2.1 Zobrazování výstupních hodnot proměnných v programu LabVIEW pomocí systému Android	13
2.1.1 Řešení v programu LabVIEW	13
2.1.2 Řešení v systému Android	14
2.2 Aktivní změna nastavení proměnných v programu LabVIEW pomocí systému Android	16
2.2.1 Řešení v LabVIEW	16
2.2.2 Řešení v systému Android	16
3 Aplikace pro sledování a řízení plicního simulátoru	17
3.1 Popis zjednodušené aplikace	18
3.1.1 Čelní panel	18
3.1.2 Blokové schéma	20
4 Služba webservis	24
4.1 Popis služby webservis	24
4.2 Konfigurace služby webservis	24
4.3 Strana serveru	28
4.3.1 Proměnné v módu WRITE	28
4.3.2 Proměnné v módu READ	29
4.3.3 Propojení sdílených proměnných se službou webservis.....	30
4.4 Strana klienta	31
Závěr	33
Literatura	34
Přílohy	36

SEZNAM OBRÁZKŮ

Obrázek 1.1 Mechanický plicní simulátor i-Lung 2.0 (převzato z [14]).	9
Obrázek 1.2 i-Lung 2.0, kovový rám přístroje (převzato z [14]).	10
Obrázek 2.1 Ikona aplikace Data Dashboard (převzato z [5]).	14
Obrázek 2.2 Ukázka aplikace Data Dashboard (převzato z [5]).	15
Obrázek 2.3 Ikona aplikace AndroView Free + LabView VI a AndroView PRO + LabView VI (převzato z [1], [2]).	15
Obrázek 3.1 Sinusový a lichoběžníkový vstupní signál a nastavení jejich parametrů ...	18
Obrázek 3.2 Ukázka tří různých vstupních signálů, sinusového, trojúhelníkového a lichoběžníkového	19
Obrázek 3.3 Ukázka z blokového schématu - volba vstupního signálu	21
Obrázek 3.4 Ukázka z blokového schématu - generování sinusového signálu	21
Obrázek 3.5 Generování trojúhelníkového a lichoběžníkového signálu, první část	21
Obrázek 3.6 Generování trojúhelníkového a lichoběžníkového signálu, druhá část	22
Obrázek 3.7 Hlídaní mezí tlaku pomocí proměnných Hysteresis a Threshold	22
Obrázek 3.8 Přepočítání signálů z generátoru (napětí [mV]) na průtok([l/min])	23
Obrázek 3.9 Nastavení času pro průběh smyčky a trvání průběhu aplikace	23
Obrázek 4.1 Konfigurace služby webservis	25
Obrázek 4.2 Výběr zdrojových souborů VI pro službu webservis a nastavení formátu výstupu této služby pro konkrétní soubor	26
Obrázek 4.3 Výběr souborů pro posílání a příjem dat pomocí služby webservis	27
Obrázek 4.4 Výřezy blokového schématu - sdílené proměnné v módu WRITE	29
Obrázek 4.5 Výřezy blokového schématu - sdílené proměnné v módu READ	29
Obrázek 4.6 Zapojení sdílených proměnných pro příjem (nahore) a export dat (dole).	30
Obrázek 4.7 Připojené proměnné nebo indikátoru k mřížce zvané	31
Obrázek 4.8 Zobrazení posílaných dat pro proměnnou pressure v internetovém prohlížeči.	32

ÚVOD

Plicní simulátor je zařízení, do kterého je možné vkládat prasečí plíce získané z jatek nebo použít latexový vak, který opravdovou plíci nahrazuje. Simulátor umožňuje udržovat plíci ve stejných podmínkách jako v organismu, vytváří volitelně nastavitelné podmínky jako je například tlak nebo teplota. Simulátor je připojen k počítači a ovládání je vytvořeno v programu LabVIEW. V čelním panelu lze měnit nastavení simulátoru a jsou tam také zobrazovány výstupy simulátoru zpravidla jako graf. [14]

Úkolem této práce bylo vytvořit uživatelské rozhraní pro bezdrátové ovládání řídicí jednotky simulátoru pomocí mobilního zařízení. Bezdrátová komunikace musí umožňovat pozorování a změnu parametrů plicního simulátoru.

Řešení bezdrátové komunikace lze rozdělit na dvě úlohy. Vytvořenou aplikaci VI je potřeba upravit tak, aby poskytovala data ze simulátoru ve formě kompatibilní pro systém Android. Pro druhou úlohu tedy pro ovládání simulátoru ze systému Android musí být tato aplikace VI přizpůsobena příjmu dat.

Cílem této práce je vyhledat různá řešení možné komunikace jak na straně LabVIEW, tak na straně vzdáleného operačního systému, a vybrat nejlepší z těchto řešení. Dále toto řešení realizovat, testovat jeho funkčnost a posoudit jeho možné uplatnění pro integraci do ovládání plicního simulátoru.

1 MODUL MECHANICKÉHO PLICNÍHO SIMULÁTORU

Simulátor plic i-Lung 2.0 je pasivní model plic, u kterého můžeme aktivně měnit chování plíce. Simulátor je konstruován ke spojení s LabVIEW, kde lze měnit parametry simulátoru a sledovat důsledky na chod plíce. Přístroj umožňuje simulaci na latexovém vaku, který se svými vlastnostmi podobá plíci, nebo na skutečných plicích prasete domácího, získávaných z jatek. Primárně je simulátor určen pro modelování pohybu nanočástic při respiraci.



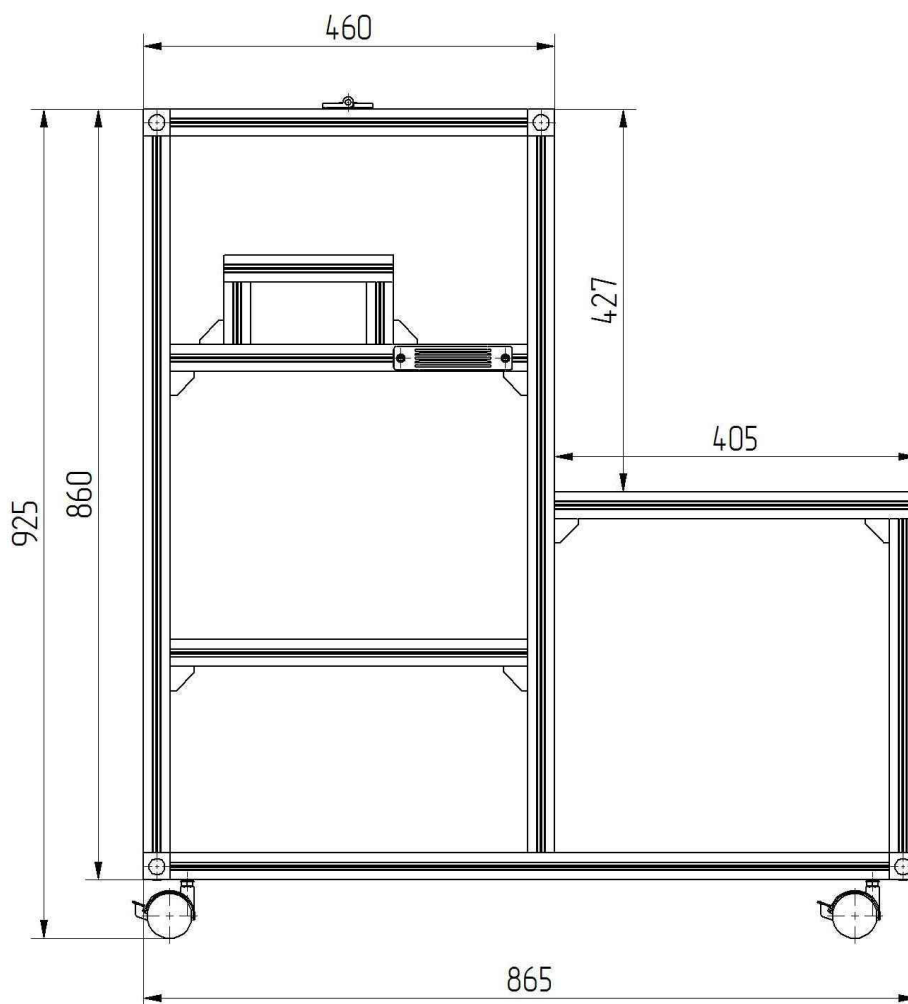
Obrázek 1.1 Mechanický plicní simulátor i-Lung 2.0 (převzato z [14]).

Plíce nebo latexový vak jsou uloženy v průhledné části přístroje (viz Obr. 2.1), která je vyrobena z plastového materiálu. Tato komora je přes ventily v jejím dně spojena s měchem a vakuovou pumpou. Pumpa vytváří stálý negativní tlak v komoře, díky kterému je situace v komoře blízká situaci v hrudníku. Trvalý negativní tlak

nahrazuje adhezi lidské plíce přes pohrudnici k vnitřnímu povrchu hrudníku. Měch, napojený přes rotační šroub k motoru, reguluje svým pohybem tlak uvnitř komory. Tlak se mění stejně jako při fyziologickém procesu dýchání. Zvýšení podtlaku v komoře (vně plíce) simuluje nádech, pokles podtlaku odpovídá výdechu. Požadovaná dechová křivka je realizována změnami podtlaku v komoře, v níž je plíce umístěna. [14]

1.1 Konstrukce

Přístroj je zkonstruován na kovovém rámu šířky 865 mm a výšky 925 mm. Hloubka rámu je 400 mm. V jednotlivých částech rámu jsou umístěny všechny komponenty přístroje. [14]



Obrázek 1.2 i-Lung 2.0, kovový rám přístroje (převzato z [14]).

V levé části přístroje je uložena elektronika, tedy napájecí zdroj a řídicí jednotka systému. Od části s komorou a od jejích vývodů je elektronika oddělena, nemůže tak dojít k žádnému vniku tekutin (např. krve) do elektronické části.

V pravé dolní části se nachází mechanický modul respiračního systému s vývody z komory, která leží nad ním. Do komory se vkládá plíce nebo latexový vak. Na obrázku je v rámu konstrukce volné místo pro tuto komoru. Komora je vyrobena z průhledného polymethylmethakrylátu s tloušťkou 15 mm, takže se dá pohyb plíce sledovat zvenčí. Rozměry komory jsou 300x300x420 mm, pracovní objem je 38 litrů. Na horní straně komory je odnímatelné víko upevněné šrouby, což zajišťuje dobrý přístup do komory. Ve dně se nachází dva ventily, kterými je komora napojena na vakuové čerpadlo a měch. Kromě ventilů se na spodní straně nachází také konektor pro připojení senzorů. [14]

1.2 Komunikace a řízení

Řídicí jednotku plicního simulátoru tvoří jednodeskový počítač (SBC – Single Board Computer). Výhodou jednodeskových počítačů je menší počet komponent a konektorů (vše je na jedné desce) a tím se zmenšuje výskyt poruch, spotřeba energie, rozměry počítače a jeho váha je také menší oproti klasickým počítačům.

Pro manuální řízení a zadávání dat je k jednodeskovému počítači buď připojen display s klávesnicí, nebo je řízení prováděno z jiného počítače. Tento počítač je připojen přes síť Ethernet nebo po sériové sběrnici RS232. Jednodeskový počítač obsahuje procesor ARM9 sloužící k vykonávání řídicího programu. Tento procesor je spojen s real-time procesorem CortexM3 přes sériové periferní rozhraní (SPI). ARM9 dostává vstupní data z počítače a přeposílá je procesoru CortexM3, který řídí motor a spínač vakuové pumpy. Kromě toho CortexM3 sbírá všechna data ze senzorů a ze spínačů krajních poloh měchu. Podle vstupních signálů z těchto senzorů a spínačů posílá CortexM3 signály ovladači motoru a pumpě a také data do ARM9 pro další zpracování. Tato data mohou být zobrazena v číselné podobě nebo v grafu. [14]

Přístroj obsahuje senzory průtoku, teploty, tlaku, vlhkosti a UV záření. Všechny senzory jsou napojeny na procesor CortexM3, využívají RS 485 komunikační protokol a jsou nízkonapěťové (5 nebo 12 V). Měch a motor přístroje jsou opatřeny dvěma

mezními spínači, které jsou napojeny na ovladač motoru. Když měch dosáhne spodní nebo horní hranice, motor se automaticky zastaví a zabrání tak možnému poškození motoru nebo měchu. [14]

2 PROPOJENÍ LABVIEW SE SYSTÉMEM ANDROID

Cílem této práce je najít způsob, jak může počítač s LabVIEW komunikovat s přístrojem s operačním systémem Android (např. tablet nebo mobilní telefon) a tuto možnost realizovat. Od této komunikace jsou požadovány dvě dílčí schopnosti: na přístroji se systémem Android požadujeme možnost sledovat okamžité výstupní hodnoty modelu, zobrazující se v čelním panelu LabVIEW; a možnost pomocí tohoto přístroje, připojeného přes internet k počítači s LabVIEW, měnit nastavení modelu, jež se za normálních podmínek nastavuje v čelním panelu aplikace v LabVIEW.

2.1 Zobrazování výstupních hodnot proměnných v programu LabVIEW pomocí systému Android

Z pohledu realizace této komunikace je jako první potřeba zajistit zobrazování výstupů a případně vstupního nastavení programu LabVIEW na přístroji s operačním systémem Android. Jde tedy o jednostranný přenos dat z počítače (z aplikace LabVIEW) na systém Android. Data budou v systému Android aktualizována s určitou frekvencí, kterou lze libovolně nastavit.

2.1.1 Řešení v programu LabVIEW

V blokovém diagramu je potřeba vytvořit sdílené proměnné (typu Network-published Shared Variables) pro ty vstupy a výstupy, které budou zobrazeny v systému Android. Tyto sdílené proměnné se připojí místo lokálních proměnných nebo navíc ke stávajícím proměnným do blokového diagramu. Druhou variantu volíme, pokud je požadována možnost měnit nastavení jak v aplikaci v LabVIEW, tak v systému Android. V takovém případě je na spoj proměnných nutné vložit prvek *logické nebo*, aby bylo možné program ovládat pomocí obou proměnných. Nebo lze lokální proměnné změnit v indikátor, takže zobrazují aktuální hodnoty v aplikaci LabVIEW nastavené pomocí služby webservis. Sdílené proměnné je nutné vytvořit z toho důvodu, že LabVIEW a LabVIEW webservis nemají stejný prostor pro ukládání dat. Přenos dat mezi paměťmi je zajištěn právě pomocí sdílených proměnných. [3], [8]

LabVIEW webservis umožňuje komunikaci s vytvořenými aplikacemi VI (VI je označení pro Virtual Instrument, dále jen VI). V aplikaci v LabVIEW je třeba zapnout Web Server, nakonfigurovat aplikaci Web Application Server (dále WAS) v internetovém prohlížeči. Je třeba povolit WAS a nastavit port (např. 8080). Lze zde také nastavovat přístupová práva. Před spuštěním služby webservis je nutné přidat VI, která budou službou využívána. Pro každé z nich je třeba zvolit RESTful web method VI, jako výstup XML soubor a v záložce URL Mappings vybrat http metodu: GET (pro výstupy z LabVIEW), POST (pro vstupní nastavení), případně PUT nebo DELETE.

Když do internetového prohlížeče zadáme adresu následujícího tvaru: `localhost:čísloportu/jmenowebservisu/vybrane.vi`, zobrazí se XML soubor s potřebnými hodnotami. Přes internetový prohlížeč je také možno měnit nastavení LabVIEW. Pak se v prohlížeči žádný XML soubor nezobrazí, protože nebyla žádná data službou webservis přijata, data byla poslána do aplikace LabVIEW. [8]

Vizuálně se dají posílaná data prohlížet například v aplikaci Data Dashboard, pokud je zařízení s touto aplikací ve stejné síti jako počítač s LabVIEW. [8], [15]

2.1.2 Řešení v systému Android

V tabletu nebo mobilním telefonu je potřeba zajistit zobrazování výstupů z aplikace LabVIEW, což může být graf, číselná hodnota, signalizace, číselný ukazatel, případně text. Existuje několik vytvořených dostupných aplikací, které to umožňují.

Aplikace od firmy National Instruments (stejně jako LabVIEW) se jmenuje **Data Dashboard**. Umožňuje vytvořit vzdálený pohled na LabVIEW, zobrazuje sdílené proměnné přenesené po síti pomocí služby webservis. Je možno zobrazit 1, 2, 4 nebo 6 ukazatelů. Ikona aplikace Data Dashboard je vidět na Obrázek 2.1, vzhled spuštěné aplikace ukazuje Obrázek 2.2. [5], [7], [8], [15]



Obrázek 2.1 Ikona aplikace Data Dashboard (převzato z [5]).



Obrázek 2.2 Ukázka aplikace Data Dashboard (převzato z [5]).

Jinou možností by bylo použít aplikace **AndroView Free + LabView VI** nebo **AndroView PRO**, jejichž ikony jsou na Obrázek 2.3. Jedná se o aplikace typu peer-to-peer, kdy jsou párována dvě zařízení propojená technologií Bluetooth. Protože Bluetooth má dosah v řádech metrů, není tato aplikace zcela vhodná pro řešení daného problému. [1], [2]



Obrázek 2.3 Ikona aplikace AndroView Free + LabView VI a AndroView PRO + LabView VI (převzato z [1], [2]).

Jinou aplikací, která není prvotně určena k zobrazování dat z aplikace LabVIEW, je **LabVIEW-Android Speed Test**. Jedná se spíše o pomocnou aplikaci, která slouží ke zjišťování rychlosti spojení smartphonu s SCCT (Smartphone & Cross-platform Communication Toolkit) knihovnou a zobrazuje, kolik kanálů a jaké rozsahy hodnot může získat. Pomocí SCCT lze data přenášet kontinuálně v reálném čase z aplikace LabVIEW do smartphonu nebo tabletu. [12]

Nejjednodušší aplikací pro daný účel je aplikace Data Dashboard, která má ale uzavřený kód a nejde proto využít pro implementaci do již hotové aplikace v Androidu. Tato aplikace byla sice vyvinuta na podobné účely, jako je zadání této práce, avšak není možné využít její dílčí části, je třeba ji převzít jako hotový celek. Další možnost – aplikace LabVIEW-Android Speed Test je prvotně určena k jinému použití. Pro dané účely by sice byla použitelná, komunikace pomocí Bluetooth ale není vyhovující z důvodu malého dosahu této technologie.

2.2 Aktivní změna nastavení proměnných v programu LabVIEW pomocí systému Android

Druhým krokem je umožnění změny nastavení programu přes systém Android. V tomto systému musí být vidět aktuální nastavení programu (stejně jako na čelním panelu). Jedná se zde o oboustranný přenos dat. Z počítače s aplikací LabVIEW do systému Android se přenáší informace o aktuálním nastavení programu (aby se zohlednily změny, pokud by bylo nastavení změněno v počítači), a změny nastavení zadané v systému Android se přenáší do počítače s LabVIEW. Změníme-li nastavení na kterékoli z obou stran, druhá strana tuto změnu registruje a zobrazí aktuální nastavení. Tato nastavení se mohou aktualizovat buď v pravidelných předem zvolených časových odstupech (např. 3 s), nebo pouze v případě, že se nastavení změní. Podle zvoleného typu aktualizace je nutné přizpůsobit aplikaci v systému Android. Posílání dat pouze v případě změny bude zřejmě výhodnější, protože se omezí objem posílaných dat. V LabVIEW bude nutné doplnit prvek, který bude změny nastavení rozpoznávat.

2.2.1 Řešení v LabVIEW

Řešení bude vypadat stejně jako u monitorování změn v LabVIEW (viz kapitola 3.1), pouze bude potřeba změnit nastavení sdílených proměnných (mód READ a WRITE) tak, aby se hodnoty z XML souboru přenesly do aplikace v LabVIEW. Detailní popis provedení je uveden v kapitole 4.3. [8]

2.2.2 Řešení v systému Android

První z již uvedených aplikací **Data Dashboard** umožňuje oboustrannou komunikaci, tedy i ovládání LabVIEW z tabletu nebo smartphonu. Pro zobrazení dat z aplikace LabVIEW slouží tlačítko Pull, pro změnu dat v LabVIEW podle nastavení v systému Android slouží tlačítko Call. Po stisknutí těchto tlačítek se aplikace Data Dashboard propojí pomocí služby webservis s aplikací LabVIEW na počítači. Přenos dat není kontinuální, probíhá pouze po stisku některého tlačítka. Aplikace Data Dashboard může pro jednoduché použití fungovat i bez sdílených proměnných v LabVIEW. Pokud jsou již vytvořené, není problém je a aplikaci v LabVIEW ponechat. [5], [7], [8]

3 APLIKACE PRO SLEDOVÁNÍ A ŘÍZENÍ PLICNÍHO SIMULÁTORU

V rámci projektu Alveopic byl sestavena aplikace „Lung Simulator“ pro sledování parametrů a řízení plicního simulátoru. Pro nastavení a odladění služby webservis byla použita zjednodušená verze programu Lung Simulator, její popis je uveden v části 4.1. Program je odladěn v prostředí LabVIEW na počítači typu PC a přenesen do jednotky Compactrio, která přímo řídí elektroniku simulátoru. Všechny aplikace jsou vytvářeny a ukládány ve verzi 2010 Fall prostředí LabVIEW. Jednotlivé verze LabVIEW nejsou mezi sebou zcela kompatibilní, proto je nutné spouštět aplikace pouze v uvedené verzi.

Služba webservis byla propojena se zjednodušenou aplikací LabVIEW, která vychází z aplikace Lung Simulator pro řízení a sledování plicního simulátoru. Výchozí aplikace snímá z plicního simulátoru hodnoty různých parametrů, 1000 vzorků za sekundu. Počet vzorků za sekundu lze nastavit v aplikaci LabVIEW i na jinou hodnotu. Použitá obnovovací frekvence 1 kS/s je v tomto případě dostačující.

Aplikace pro sledování a řízení plicního simulátoru má prvky pro ovládání simulátoru (například nastavení dechové křivky) a prvky, které zobrazují aktuální parametry simulátoru případně jejich průběh v grafu. Mezi takovéto parametry patří tlak v nádobě plíce [mBar], průtok plicí a požadovaný průtok [l/min], pozice měchu [mm] a celkový čas od spuštění aplikace [s]. Požadujeme zobrazení a možnost pozorování těchto parametrů s výjimkou času. V systému Android požadujeme možnost ovládání proměnných například frekvence dechové křivky, hystereze nebo teploty.

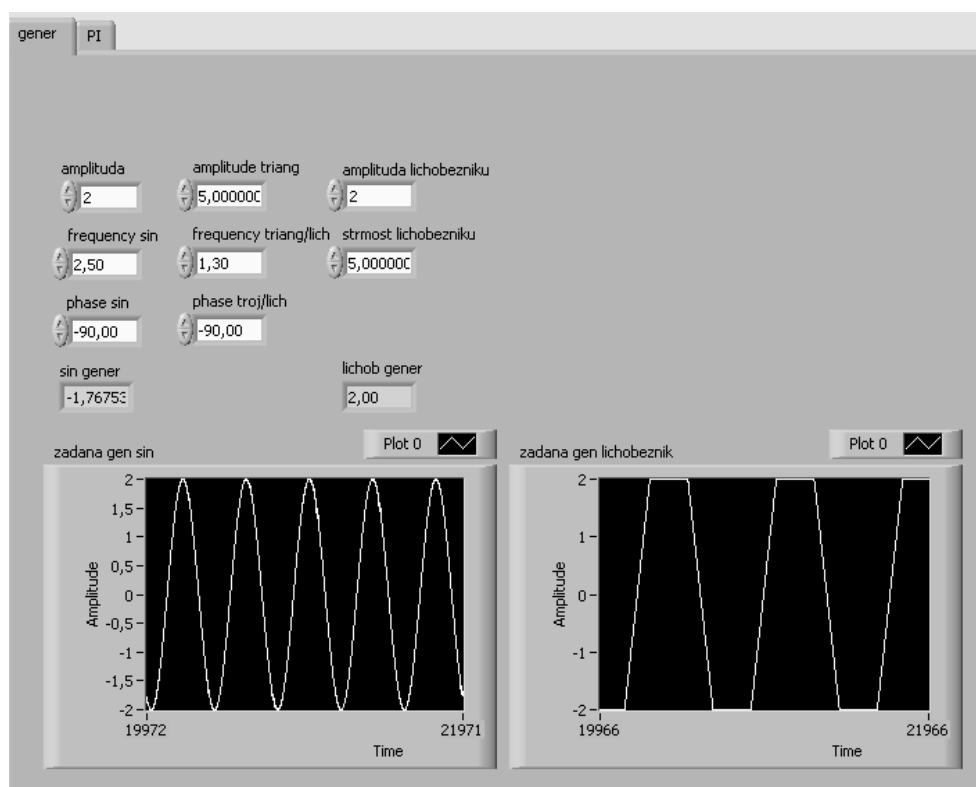
Zjednodušená aplikace neslouží pro sledování a ovládání plicního simulátoru. Je to simulace řídicí aplikace k plicnímu simulátoru, která je samostatně funkční bez propojení s plicním simulátorem. Vychází z původní aplikace, ale proměnné ze simulátoru jsou nahrazeny generovanými signály. Zvolit lze tvar dechové křivky a z ní jsou matematicky přepočítány ostatní veličiny, které by měly být v simulátoru měřeny. Nastavení některých parametrů je zachováno (například hystereze). Možnosti nastavování a zobrazované veličiny jsou popsány v dalších odstavcích.

3.1 Popis zjednodušené aplikace

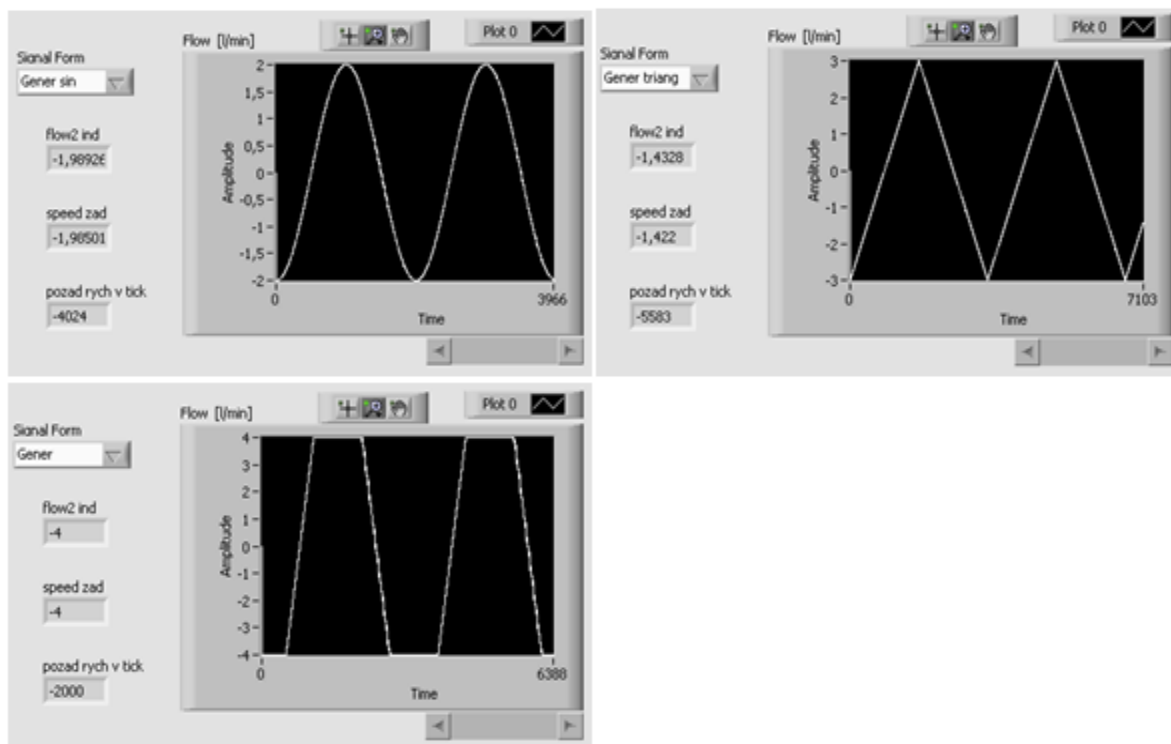
3.1.1 Čelní panel

Čelní panel aplikace obsahuje prvky pro ovládání (nastavení parametrů simulujících signálů), a prvky zobrazující stav plicního simulátoru (v tomto případě jsou hodnoty proměnných popisující stav simulátoru pouze přepočty vstupního simulujícího signálu).

Mezi ovládací prvky patří volba požadované formy vstupního signálu (což je dechová křivka). Na výběr jsou signály sinusový, trojúhelníkový a lichoběžníkový. Pro každý signál je možné nastavit amplitudu a frekvenci, případně počáteční fázi. Signály jsou ukázány v grafech v pravé části čelního panelu, vedle nastavení amplitud a frekvencí, ukázka je na Obrázek 3.1. Všechny tři možné tvary dechové křivky jsou zobrazeny na Obrázek 3.2. Lichoběžníkový signál je tvořen z trojúhelníkového signálu oříznutím špičky trojúhelníku. Strmost lichoběžníku udává výšku (amplitudu) původního trojúhelníkového signálu, amplituda pak udává výšku, ve které bude špička trojúhelníku uříznuta a bude zde vodorovná část lichoběžníku.



Obrázek 3.1 Sinusový a lichoběžníkový vstupní signál a nastavení jejich parametrů



Obrázek 3.2 Ukázka tří různých vstupních signálů, sinusového, trojúhelníkového a lichoběžníkového

Na Obrázek 3.2 jsou zobrazeny tři varianty vstupního signálu. Zobrazený sinusový signál má amplitudu 2 l/min (na grafu průtoku, který je uveden v jednotkách l/min) a frekvenci 0,5 Hz. Trojúhelníkový signál má amplitudu 3 l/min (opět graf průtoku) a frekvenci 0,3 Hz, u lichoběžníku je amplituda nastavena na 4 l/min a strmost je 11 l/min.

Aplikace obsahuje dvě tlačítka *STOP*, jedno pro pohyb měchu a rozpínání plíce, a druhé pro zastavení celé aplikace.

Simulátor lze ručně ovládat pomocí tlačítka pro vypnutí pumpy, vhánějící vzduch do plíce, tlačítka otevření ventilu nádoby, kde se plíce nachází, a tlačítka pro zastavení motoru (který mění pozici měchu). Ke každému tlačítku je v čelním panelu příslušná indikace, která značí zapnutý nebo vypnutý stav.

Proměnné *Hysteresis* a *Threshold* – nastavení horní a dolní meze tlaku, při poklesu hodnoty tlaku pod nastavenou hodnotu hystereze se sepne pumpa a obnoví výchozí hodnotu tlaku v komoře. Překročení daných mezí tlaku je indikováno rozsvícením kontrolky u příslušné proměnné (*Hysteresis* nebo *Threshold*).

Tlačítkem *real/simulace* lze přepínat mezi tím, zda jsou zobrazované hodnoty změřené na plicním simulátoru, nebo vypočtené z parametrů vstupních signálů. U

zjednodušené aplikace nemá toto tlačítko příliš smysl, neboť plicní simulátor není s aplikací propojen.

Číselná hodnota *maxtime* je nastavení času, po jehož uběhnutí se průběh aplikace automaticky zastaví, lze nastavit libovolně.

Obrázek čelního panelu je v přílohách na konci práce.

Prvky, které zobrazují stav plicního simulátoru, jsou 4 grafy. Zobrazují tlak v nádobě, kde je uložena plíce (graf s názvem *pressure*), průtok vzduchu plicí (*flow*) a požadovaný průtok (*flow* požadována), a pozici měchu (*position*). Funkce měchu je popsána v kapitole 2. Aktuální číselná hodnota průtoku se ukazuje v indikátoru vedle příslušného grafu (*flow*).

Vedle grafů se v proměnné *speed_zad* zobrazuje žádaná rychlost motoru (počet otáček za sekundu) a druhý indikátor udává rychlost přepočtenou na tiky.

Pro kontrolu jsou v horní části čelního panelu umístěny dva indikátory zobrazující čas trvání dvou smyček *while*. Doba trvání jednoho průběhu smyčky je předem známá a měla by se s touto hodnotou shodovat.

3.1.2 Blokové schéma

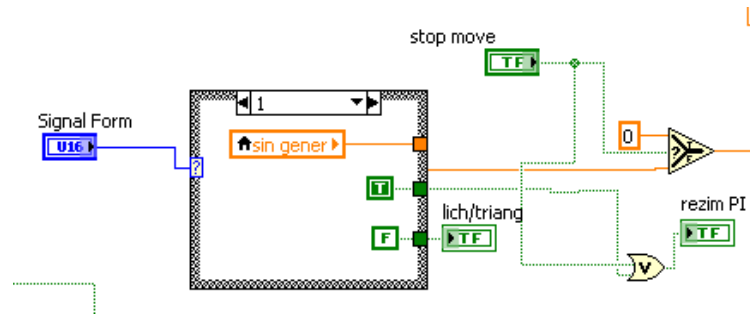
Blokový diagram je rozdělen do tří svislých polí. První pole (vlevo) je určeno pro inicializaci, nastavuje se v něm počáteční tlak v komoře, pozice měchu, teplota vzduchu a podobně. V druhém poli jsou umístěny dvě smyčky *while*, zde běží vlastní program, a třetí pole slouží pro uvedení simulátoru do klidového režimu po zastavení běhu programu. Je opět třeba nastavit tlak a teplotu v komoře.

V prostřední části jsou dvě smyčky, smyčka v horní části slouží pro čtení a vyhodnocování naměřených dat (v případě připojeného plicního simulátoru) a také pro simulaci signálu (v tomto případě, kdy aplikace není propojena se simulátorem). Smyčka v dolní části slouží jako PI regulátor pro řízení dechové křivky, ve zjednodušené (pouze simulační) aplikaci se tato smyčka nevyužívá. Důvodem, proč jsou tu dvě oddělené smyčky a nejsou všechny bloky v jedné, je ten, že každá smyčka může mít jinou periodu a vzorkovací frekvenci a jejich průběh je nezávislý.

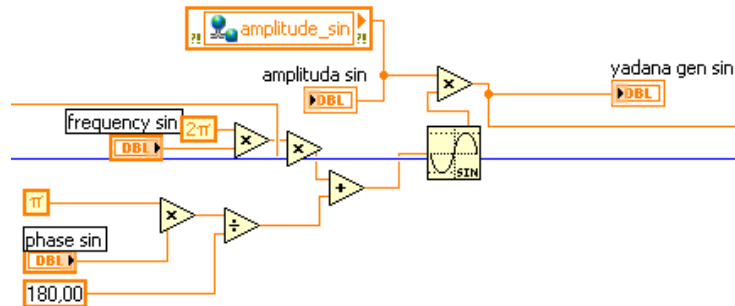
V horní smyčce se generují signály, probíhá zde řízení plíce (nastavování nejrůznějších hodnot, například nastavení prahu tlaku v komoře) a zpracování přijatých signálů (přepočet hodnot, například z rychlosti). Požadované hodnoty jsou použity pro další řízení, například požadovaná rychlost otáček motoru. Části blokového schématu pro řešení dílčích úkolů programu jsou uvedeny v následujících odstavcích.

Volba formy vstupního signálu je zobrazena na Obrázek 3.3, ve struktuře *Case* jsou zahrnuty všechny možnosti formy signálu, obrázek ukazuje nastavení pro sinusový

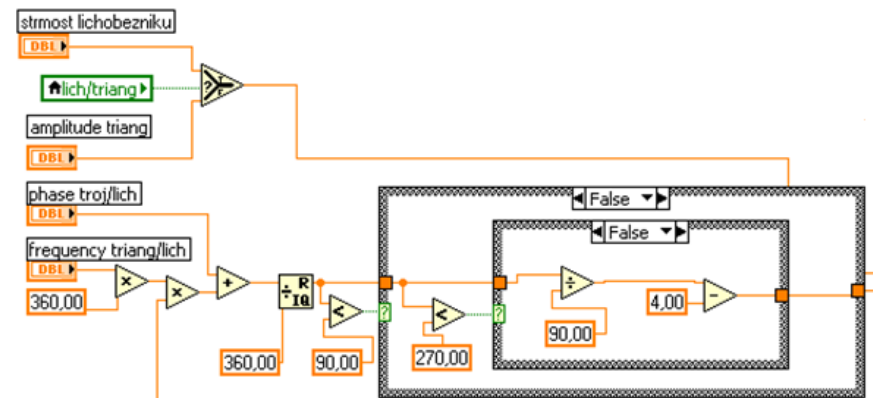
vstupní signál. Generování vstupního sinusového signálu je na Obrázek 3.4, generování trojúhelníkového a lichoběžníkového signálu je na Obrázek 3.5 a Obrázek 3.6.



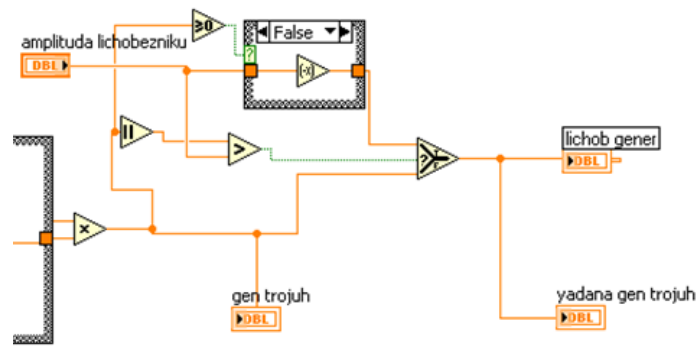
Obrázek 3.3 Ukázka z blokového schématu - volba vstupního signálu



Obrázek 3.4 Ukázka z blokového schématu - generování sinusového signálu

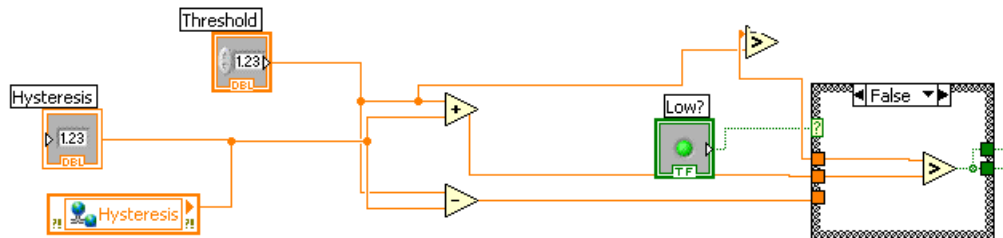


Obrázek 3.5 Generování trojúhelníkového a lichoběžníkového signálu, první část



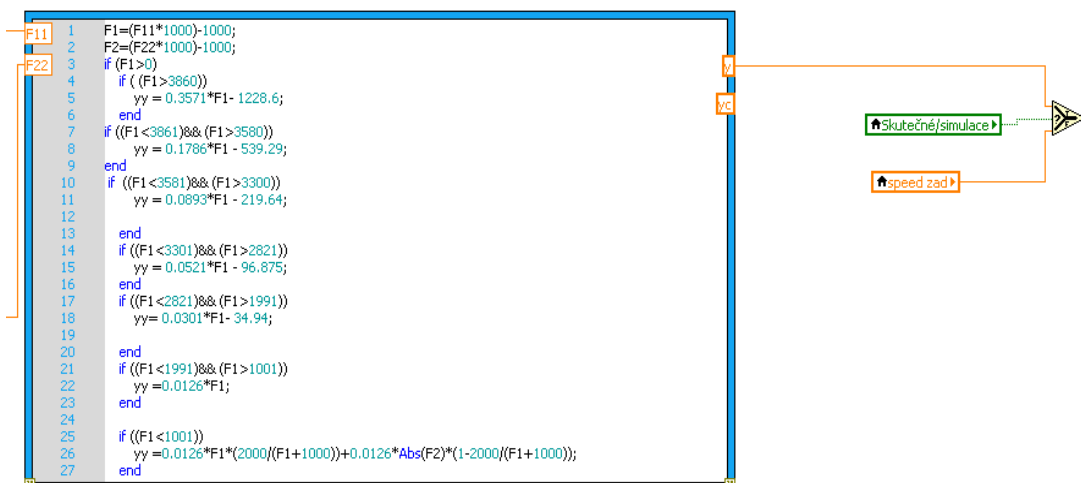
Obrázek 3.6 Generování trojúhelníkového a lichoběžníkového signálu, druhá část

Tlak v komoře se musí pohybovat mezi předem určenými mezemi, horní mez je daná proměnnou *Threshold*, dolní mez určuje proměnná *Hysteresis*. Blokové schéma hlídání překročení mezí tlaku v komoře je uvedeno na Obrázek 3.7.



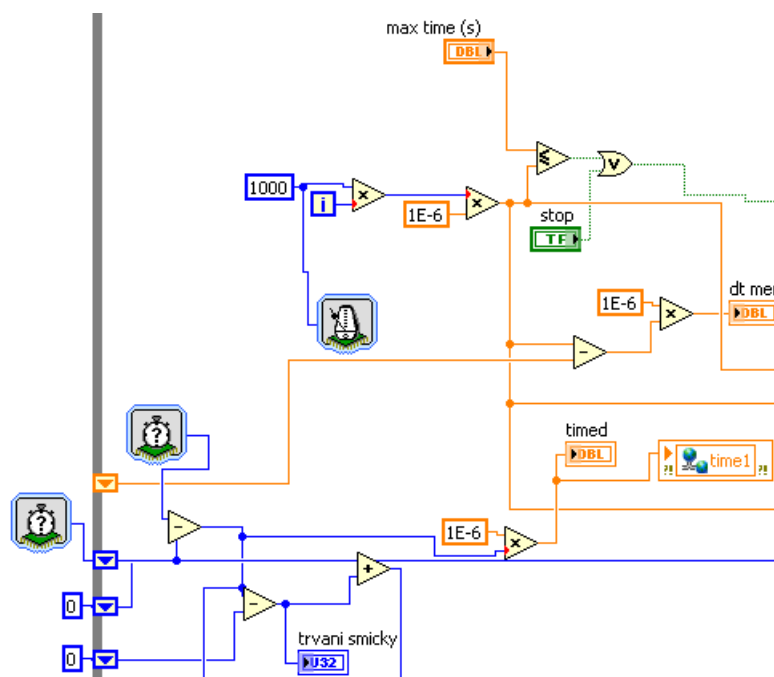
Obrázek 3.7 Hlídání mezi tlaku pomocí proměnných *Hysteresis* a *Threshold*

V původní aplikaci Lung Simulator jsou parametry měřeny buďto na plicním simulátoru, nebo jsou přepočítávány ze vstupního signálu. Pro přepínání slouží tlačítko *real/simulace*, který bylo popsáno výše. Přepočet signálu z generátoru reprezentujícího napětí (mV) na průtok (l/min) je na Obrázek 3.8.



Obrázek 3.8 Přepočítání signálů z generátoru (napětí [mV]) na průtok([l/min])

Nastavení opakovací frekvence smyčky je patrné na Obrázek 3.9, celkový čas od spuštění aplikace a trvání jednoho průběhu smyčky je na tomtéž obrázku.



Obrázek 3.9 Nastavení času pro průběh smyčky a trvání průběhu aplikace

4 SLUŽBA WEBSERVIS

4.1 Popis služby webservis

Služba webservis v LabVIEW používá strukturu takzvaného RESTful webservisu (Representation State Transfer - REST). Tento typ webservisu poskytuje zjednodušený protokol přístupný široké škále HTTP klientů a nepožaduje přesnou syntaxi.

Služba webservis podporuje následující datové typy: string, path, boolean, numeric (kromě komplexních čísel), enumeration (používá 8, 16 a 32 bitů). Dále array, cluster, digital/analog waveform ale pouze pro výstupy XML a JSON.

Aby byla služba webservis dostupná, vytváří LabVIEW soubor typu .lwvs. Poté, co jsou nastaveny všechny požadované parametry služby webservis a stávající aplikace je zpřístupněna službě webservis, vzniká zip soubor původního souboru s příponou .lwvs, do kterého jsou uloženy všechny důležité nastavení týkající se služby webservis. Po provázání aplikace se službou webservis může být soubor typu .lwvs přesunut na jiný cílový webserver, který může být spuštěn na libovolném počítači. Tam se v prostředí LabVIEW soubor typu .lwvs dekomprimuje a nainstaluje komponenty služby webservis do samostatného adresáře na cílovém počítači. [12]

4.2 Konfigurace služby webservis

V možnostech projektu vytvořeném v LabVIEW je aplikace webservis. Po výběru této možnosti program automaticky otevře v okně internetového prohlížeče stránku pro konfiguraci aplikace webservis. Nastavení web serveru je rozděleno na část System Web Server a Application Web Server (viz obrázek Obrázek 4.1). Obě části umožňují nastavení síťového portu. Port pro System Web Server slouží pro konfiguraci webservisu, v tomto případě port 3580 a konfigurace je dostupná na adrese:

localhost:3580/WebServerConfigExtension;component/

pro případ, že se okno konfigurace neotevře automaticky. Aplikaci Web Server (nastavení Application Web Server) používá LabVIEW pro komunikaci s vnějším světem. Port v Application Web Server určuje, na jakém síťovém portu poběží služba

webservis. Port 8080 je nastaven jako výchozí, lze jej změnit dle potřeby, například v souladu s nastavením pro Firewall.

Port 8080 je alternativa k portu 80, port 80 se používá pro HTTP nebo pro servery jako Apache či Microsoft IIS (Microsoft Application Server). Port 8080 je používán ve stejných případech. [13], [16]

Síťový port je šestnáctibitové číslo od 1 do 65 535, které se používá v počítačových sítích pro komunikaci pomocí protokolů TCP a UDP k odlišení síťové služby v rámci počítače. Číslo portu se někdy objevuje v URL adresách webových stránek. V protokolech UDP a TCP je v hlavičce uvedeno číslo zdrojového a cílového portu. Protokol UDP posílá pakety v přesném pořadí, může dojít ke ztrátám některých paketů, hodí se pro kontinuální přenos dat. Protokol TCP je spolehlivější, každý paket je očíslován a nedochází k jejich ztrátám. Mohou ale přicházet v přeházeném pořadí. Služba webservis využívá protokol TCP. [4], [6], [11], [12], [17]

Aplikaci webservis je potřeba aktivovat (zvolit možnost Apply, viz Obrázek 4.1). Na obrázku není tlačítko Apply dostupné, při změně nastavení již bude možné tlačítko Apply použít.



Obrázek 4.1 Konfigurace služby webservis

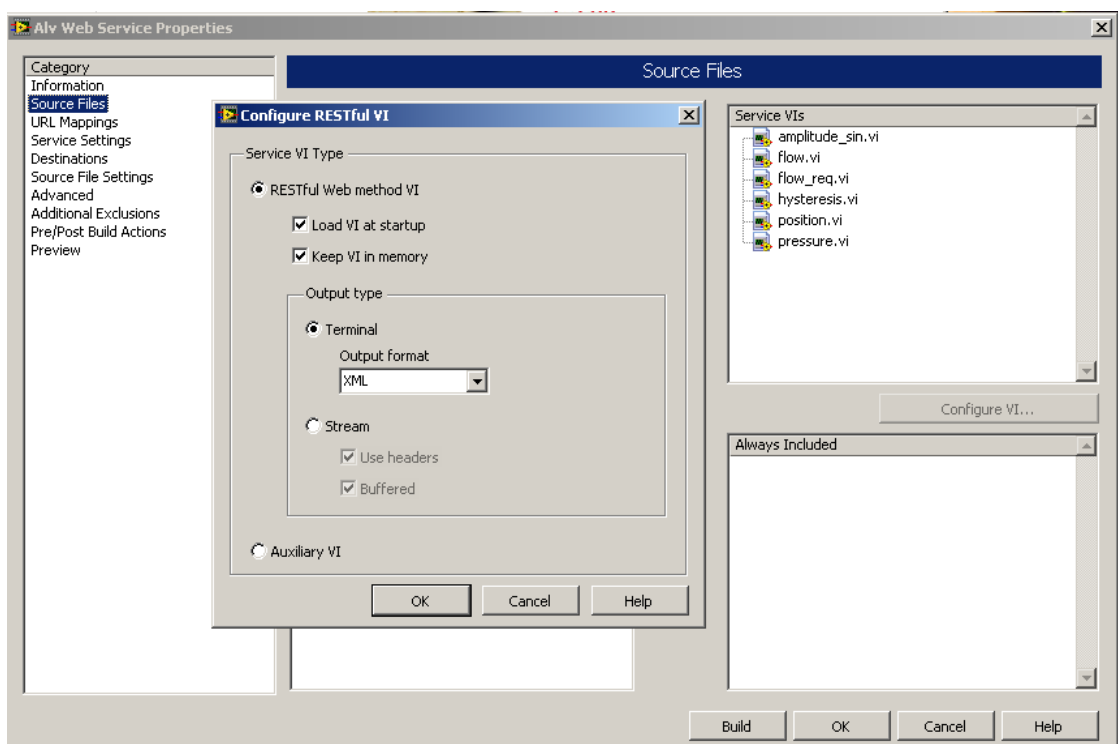
Dále lze nastavit parametry zabezpečení, přidělit různým uživatelům práva pouze na prohlížení, nebo i na změnu nastavení služby webservis a podobně. Oprávnění a nové uživatele zde lze vytvářet a definovat. [8]

Poté je nutné restartovat program LabVIEW a celý počítač. Pak znovu otevřeme program LabVIEW. V projektu, do kterého bude uloženo nastavení služby webservis, v

záložce BuildSpecification spustíme novou službu webservis typu RESTful. Ve vlastnostech této služby webservis je třeba nastavit následující:

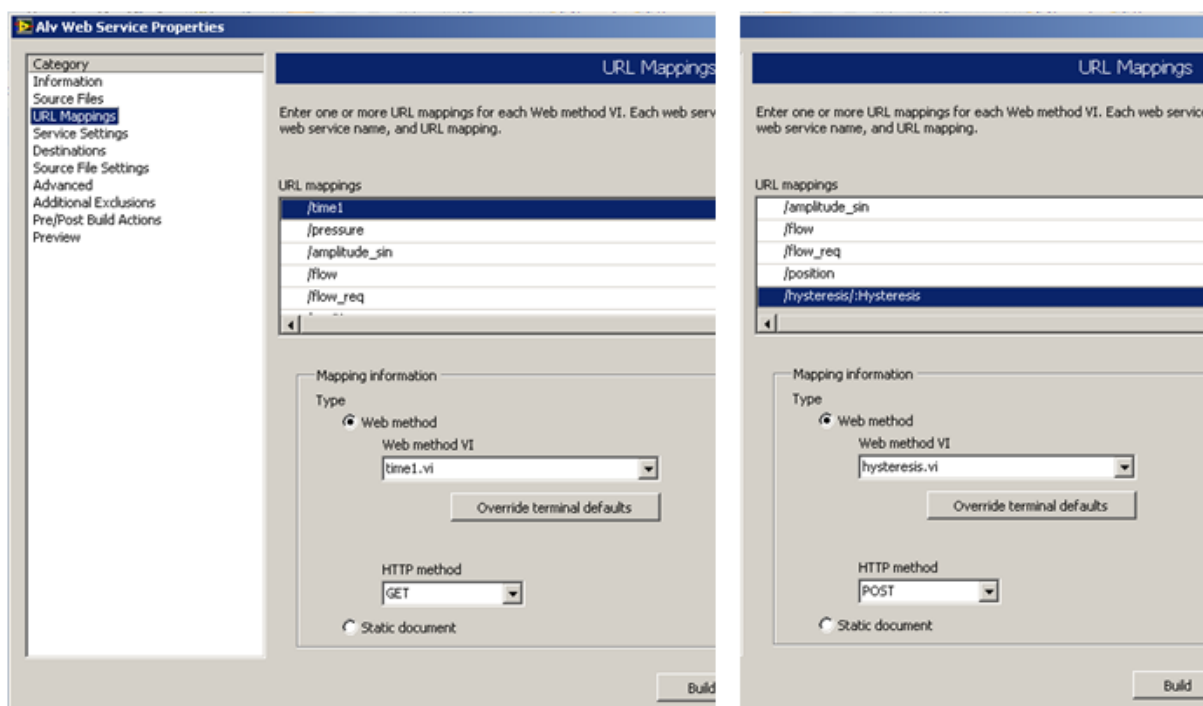
V záložce Information je nutné vyplnit SpecificationName (takto se bude servis jmenovat a toto jméno se zobrazí v projektu) a ServiceName (pod tímto jménem se bude soubor označovat v internetovém prohlížeči).

V záložce Source Files zvolíme soubory s koncovkou .vi, pomocí kterých budou data z prostředí LabVIEW posílány službou webservis k zobrazení v internetovém prohlížeči či v aplikaci na jiném zařízení, nebo naopak z jiného zařízení či z internetového prohlížeče do aplikace v LabVIEW. Vybrané soubory jsou v sekci Service Vis (viz Obrázek 4.2). Při výběru jednotlivých VI se vždy otevře okno konfigurace (na obrázku uprostřed), kde je třeba znovu vybrat metodu RESTful a formát výstupu webservis. Ve všech zvolených VI je nastavený formát XML, lze však zvolit také formáty HTML, JSON nebo text. XML bylo zvoleno z toho důvodu, že to je obecný značkovací jazyk, neobsahuje předdefinované značky, ale je nutné si vlastní značky vytvořit. Díky tomu je možné zakomponovat prvky, které by například v HTML nemohly být, protože tam nejsou definované. Takto se zapisují přenášené hodnoty proměnných a jejich jména z LabVIEW. [9]



Obrázek 4.2 Výběr zdrojových souborů VI pro službu webservis a nastavení formátu výstupu této služby pro konkrétní soubor

V záložce URL Mapping je pro každou sdílenou proměnnou (SharedVariables) nastaven soubor VI, ze kterého budou hodnoty sdílených proměnných posílány pomocí služby webservis (viz Obrázek 4.3). Níže je nastavení http metody, v nabídce jsou následující možnosti: GET, POST, PUT, DELETE. Metoda GET je použita u proměnných, jejichž hodnoty má webservis přenášet z LabVIEW (například time 1 na Obrázek 4.3). Pro proměnné, u kterých požadujeme měnit jejich hodnoty v internetovém prohlížeči a pomocí webservis toto nastavení přenášet do LabVIEW, je použita metoda POST. V této záložce je možné nastavit pro jednotlivé VI povolení pouze k prohlížení nebo i pro změnu dat, podle toho, jaká práva byla dříve vytvořena při konfiguraci v internetovém prohlížeči.



Obrázek 4.3 Výběr souborů pro posílání a příjem dat pomocí služby webservis

V dalších záložkách lze nastavit místo uložení souboru služby webservis (soubor .lvws), měnit přístupová hesla pro konfiguraci služby webservis, nastavit spuštění některých programů před nebo po spuštění služby webservis a podobně.

4.3 Strana serveru

Fungování služby webservis a přenos dat je spojen se sdílenými proměnnými (SharedVariables typu Netowk-published) v LabVIEW. V hlavním programu, který zobrazuje výstupy z plicního simulátoru a umožňuje řízení tohoto modelu, jsou sdílené proměnné v blokovém diagramu připojeny k proměnným, jejichž hodnoty chceme pomocí služby webservis přenášet nebo měnit. Sdílené proměnné se připojí místo lokálních proměnných nebo navíc ke stávajícím proměnným do blokového diagramu, detailněji je to popsáno v kapitole 2.1. Sdílené proměnné je třeba vytvořit z toho důvodu, že LabVIEW a LabVIEW Web Service nemají stejný prostor pro ukládání dat. Přenos dat mezi paměťmi je zajištěn právě pomocí sdílených proměnných. [3], [8]

Ke každé proměnné, kterou chceme pomocí služby webservis přenášet, je vytvořen program, který obsahuje sdílenou proměnnou stejnou jako v hlavním programu. Rozdíl je pouze v nastavení sdílené proměnné, aby si tyto dvě stejné proměnné mohly předávat data mezi sebou a tím pádem mezi dvěma programy, musí být jedna nastavena na mód WRITE – na té straně, kde budeme data měnit, a druhá na mód READ která změněná data akceptuje. Data se přenáší ve směru z proměnné módu WRITE do proměnné módu READ.

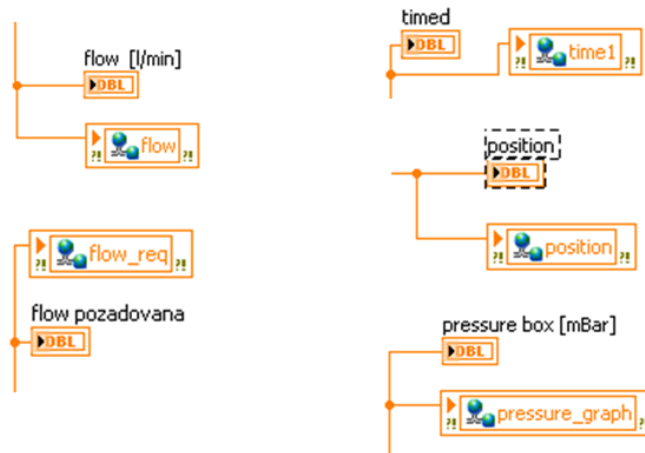
Služba webservis umožňuje dva druhy přenosu dat. Pro sledování dat mimo LabVIEW se přenáší data z LabVIEW a zobrazují se v internetovém prohlížeči, v tomto případě je v hlavním programu sdílená proměnná nastavená na mód WRITE a v pomocném programu je proměnná typu READ. Pro změnu nastavení v LabVIEW přes službu webservis je nastavení opačné, v hlavním programu je sdílená proměnná v módu READ a v pomocném programu v módu WRITE.

V následujících odstavcích jsou zobrazeny a popsány výřezy blokového schématu ukazující zapojení sdílených proměnných v hlavním programu jak v módu WRITE, tak v módu READ.

4.3.1 Proměnné v módu WRITE

Proměnné, které byly ve službě webservis nakonfigurovány k zobrazení jsou na Obrázek 4.4. Patří mezi ně průtok (flow), požadovaný průtok (flow pozadovana), celkový čas od spuštění aplikace (timed), pozice měchu (position) a tlak v komoře (pressure box). Ke všem proměnným je připojena odpovídající sdílená proměnná (její název je napsán uprostřed bloku sdílené proměnné). Jelikož u těchto parametrů požadujeme přenos dat z aplikace LabVIEW pomocí služby webservis, jsou sdílené

proměnné nastaveny do módu WRITE (pro zapisování) a tok dat ve formátu double, znázorněný oranžovým spojem vede do těchto proměnných.

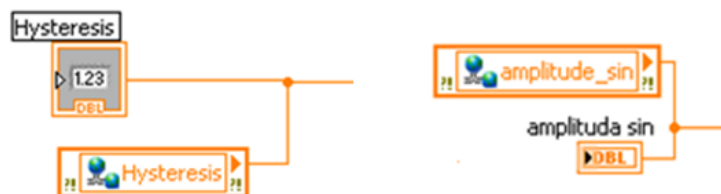


Obrázek 4.4 Výřezy blokového schématu - sdílené proměnné v módu WRITE

Výstupy (flow, timed, aj.) jsou v hlavním programu nadále ponechány jako indikátory, zobrazují tedy aktuální hodnoty. Kromě toho lze v internetovém prohlížeči zobrazit hodnoty přenášené pomocí služby webservis.

4.3.2 Proměnné v módu READ

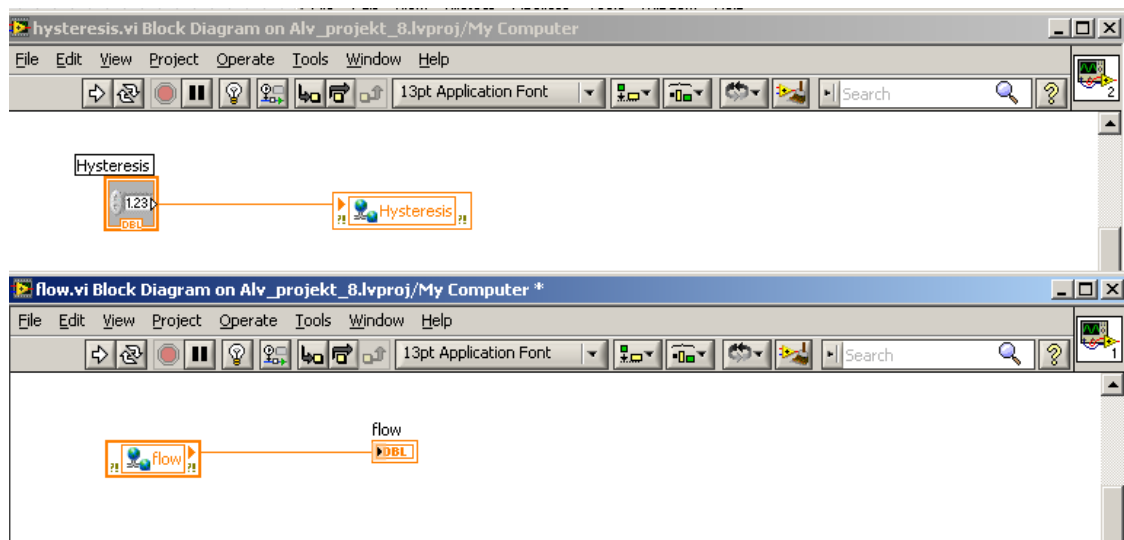
Proměnné, jejichž nastavení lze provádět v systému Android nebo z internetového prohlížeče, jsou na Obrázek 4.5. Byly zvoleny proměnné *Hysteresis* (spodní práh tlaku v komoře) a *Amplitude_sin* (amplituda vstupního sinusového signálu) pro možnost nastavování mimo aplikaci LabVIEW a přenos nastavení službou webservis. Ke každé z proměnných je připojena sdílená proměnná v módu READ. Z této proměnné jsou data načítána v aplikaci LabVIEW a to znázorňuje i spoj boků, který zobrazuje tok dat ve formátu double ze sdílených proměnných. Původní proměnné jsou převedeny na indikátor a pouze zobrazují aktuální nastavení.



Obrázek 4.5 Výřezy blokového schématu - sdílené proměnné v módu READ

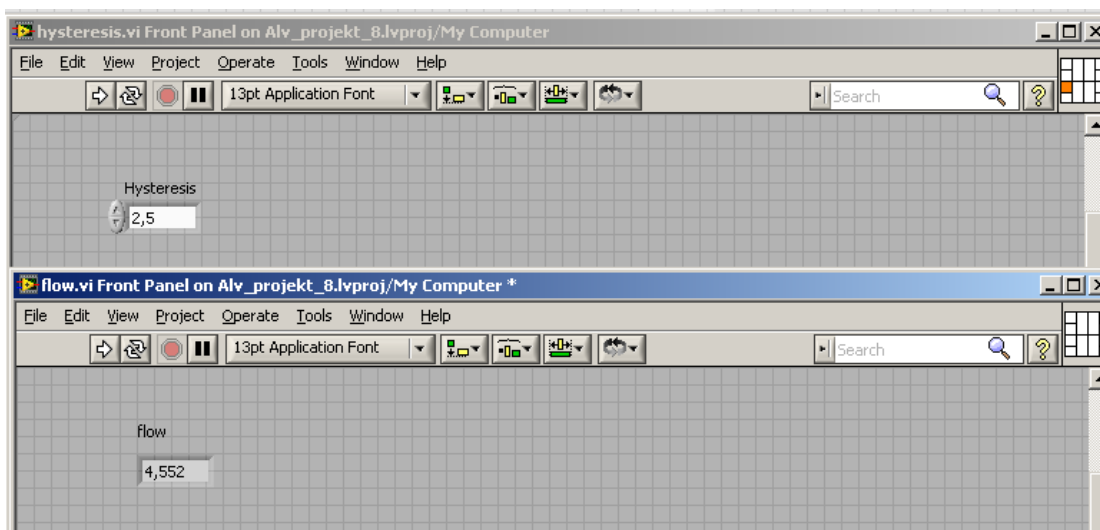
4.3.3 Propojení sdílených proměnných se službou webservis

V každém pomocném programu se nachází sdílená proměnná a indikátor nebo proměnná *Control*. U nastavování hodnot pomocí služby webservis je hodnota přenášena z proměnné *Control* do sdílené proměnné. Pro sledování hodnot je přenášena hodnota ze sdílené proměnné do indikátoru. Rozdíl v zapojení je patrný na Obrázek 4.6 v okně nahoře je varianta s proměnnou *Control*, ve spodním okně varianta s indikátorem.



Obrázek 4.6 Zapojení sdílených proměnných pro příjem (nahore) a export dat (dole)

V čelním panelu pomocných programů je indikátor nebo proměnná *Control* připojena na takzvaný *Connector pane*, což je mřížka v pravém horním rohu každého okna (viz Obrázek 4.7). Do pravé části se připojují proměnné *Control* pro tok dat ze služby webservis do tohoto pomocného programu, do levé části se připojují indikátory, ze kterých jsou data exportována do služby webservis.



Obrázek 4.7 Připojené proměnné (nahore) nebo indikátoru (dole) k mřížce zvané *connector pane* (v pravém horním rohu)

4.4 Strana klienta

Data posílaná z LabVIEW pomocí služby webservis si lze prohlédnout v internetovém prohlížeči na adrese formátu:

http://localhost:číslo-portu/jméno-webservisu/název-proměnné

pro přístup z počítače, ve kterém služba běží. Adresa může mít tvar například

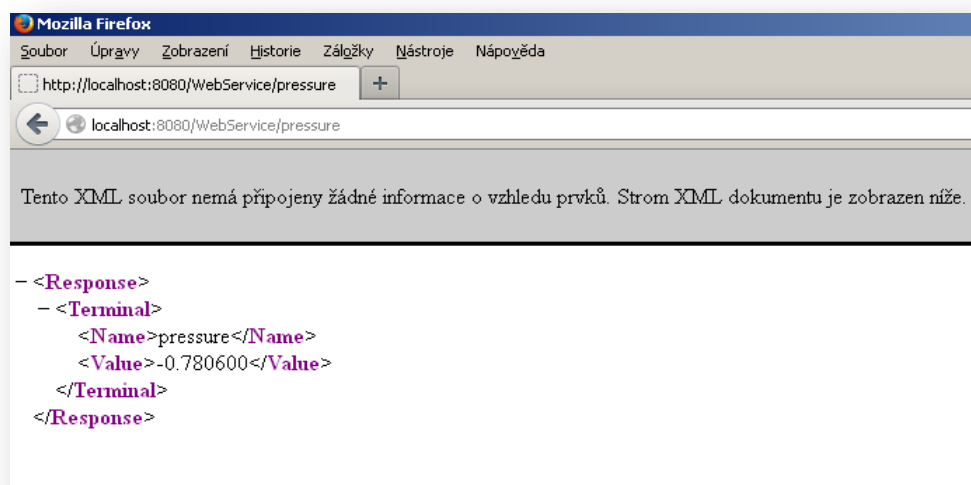
http://localhost:8080/WebService/pressure

pro konkrétní webservis a proměnnou pressure. Pro zobrazení dat na jiném počítači je v adrese nutno localhost nahradit IP adresou počítače, na němž služba webservis spuštěna.

Formát dat byl nastaven ve vlastnostech služby webservis, byl zvolen formát XML (viz kapitola 4.2).

Internetový prohlížeč zobrazuje data posílaná službou webservis. Na Obrázek 4.8 je vidět formát přenesených dat (XML). Prohlížeč během testování služby zobrazuje aktuální hodnotu zvolené proměnné (řádek Value), jméno zobrazované proměnné (Name), zapsáno vše ve formátu XML. Řádek Response značí, že byla přijatá nějaká

data, Terminal určuje použití terminálu. Při každém načtení této stránky se zobrazí poslední (aktuální) hodnota.



Obrázek 4.8 Zobrazení posílaných dat pro proměnnou pressure v internetovém prohlížeči

Při změně nastavení ze vzdáleného zařízení pomocí služby webservis se v internetovém prohlížeči žádná data nezobrazují, zobrazí se pouze formulace: `<Response/>`, která poukazuje na to, že data byla odeslána.

Aplikace pro systém Android, která bude se službou webservis spojena, bude načítat data přenášená z LabVIEW službou webservis s určitou frekvencí. Pro každý nový údaj je třeba, aby aplikace aktualizovala posílané hodnoty, které jsou na Obrázek 4.8. Koncové uživatelské prostředí pro zobrazování proměnných a nastavování parametrů musí být sestaveno tak, aby v daných intervalech provádělo znovunačítání aktuální hodnoty.

ZÁVĚR

V první části této práce je popsán plicní simulátor, pro který má být ovládání pomocí systém Android vytvořeno. Jsou definovány požadavky na výměnu dat. Ve druhé části jsou uvedeny různé možnosti pro komunikaci systému Android s aplikací v prostředí LabVIEW. Nejprve je popsáno řešení pouze pro sledování výstupů čelního panelu v systému Android a poté i pro vzdálené ovládání simulátoru. Řešení je rozděleno na dvě části. První částí je úprava aplikace VI v LabVIEW a dostupné aplikace pro monitorování, druhou částí je návrh možné kontroly simulátoru. Po diskuzi výhod a nevýhod každého řešení bylo zvoleno nejlepší z nich. V LabVIEW byly vytvořeny sdílené proměnné a pomocí služby webservis bude aplikace VI komunikovat s aplikací v systému Android.

V rámci bakalářské práce bylo testováno vzdálené sledování a řízení plicního simulátoru vybraným řešením. Monitorování a nastavování proměnných z aplikace probíhá v kompatibilním formátu XML, který prezentuje každou definovanou proměnnou zvlášť. Tento formát umožňuje díky své kompatibilitě mezi různými operačními systémy implementaci do aplikace v systému Android. Část služby, která zprostředkovává monitorování, funguje bez problémů, část služby, která umožňuje i aktivní změnu hodnot, vykazuje na různých počítačích problémy s jejím nastavením. Toto může být závislé na nastavení firewallu daného počítače, protože u aktivního nastavení přijímá služba po daném portu data, která mohou být zahazována. Nastavení je proto třeba pro danou hostitelskou stanici s prostředím LabVIEW odladit. Služba a program byly odladěny pro LabVIEW ve verzi 2010 Fall.

Bakalářská práce na téma Uživatelské rozhraní pro systém Android pro ovládání plicního simulátoru je součástí řešení projektu ALVEOPIC – Advanced Lung research for VETERINARY medicine Of Particlesfor Inhalation - Cooperation, projektové číslo OPPTS AT-ČR M00250. Hlavním řešitelem projektu je FH TechnikumWien, spoluřešitelem VUT v Brně. Výsledek práce nabízí a popisuje možnost přenosu dat mezi prostředím LabVIEW a klienty a to nezávisle na klientském operačním systému. Navržené řešení je tak použitelné pro implementaci do aplikace vyvinuté v rámci projektu Alveopic.

LITERATURA

- [1] AndroidView Free + LabView VI. [online]. 2012 [cit. 2014-01-06]. Dostupné z: <https://play.google.com/store/apps/details?id=com.heightdev.androidviewbluetooth>
- [2] AndroidView PRO + LabView VI. [online]. 2012 [cit. 2014-01-06]. Dostupné z: <https://play.google.com/store/apps/details?id=com.heightdev.androidviewbluetoothpro>
- [3] BITTER, Rick, Taqi MOHIUDDIN a Matt NAWROCKI. LabView advanced programming techniques. 2nd ed. BocaRaton: CRC Press, c2007, 499 s. ISBN 08-493-3325-3
- [4] CRAFT, Nix. Difference between UDP and TCP. www.cyberciti.biz/ [online]. 15.5.2007. [cit. 2014-05-27]. Dostupné z: <http://www.cyberciti.biz/faq/key-differences-between-tcp-and-udp-protocols/>
- [5] Data Dashboardfor LabVIEW. *Data Dashboard for LabVIEW* [online]. 2012 [cit. 2014-01-06]. Dostupné z: <https://play.google.com/store/apps/details?id=com.ni.labview.SharedVariableViewer>
- [6] Diskuzní fórum National Instruments . [online]. 2014. vyd. [cit. 2014-05-27]. Dostupné z: <http://forums.ni.com/>
- [7] GRANT HEIMBACH. *Data Dashboard for LabVIEW app communicating with a LabVIEW Web Service* [online]. 2013, 22. 4. 2013 [cit. 2014-01-06]. Dostupné z: <http://www.youtube.com/watch?v=4ZblAhW422U>
- [8] GRANT HEIMBACH. *How to create a LabVIEW Web Service* [online]. 2013, 1. 3. 2013 [cit. 2014-01-06]. Dostupné z: <http://www.youtube.com/watch?v=FCujba8euWc>
- [9] KOSEK, Jiří. Seriál o XML. [Kosek.cz](http://www.kosek.cz) [online]. 2001. vyd. [cit. 2014-05-27]. Dostupné z: <http://www.kosek.cz/clanky/swn-xml/index.html>
- [10] LabVIEW-Android Speed Test. [online]. 2013 [cit. 2014-01-06]. Dostupné z: <https://play.google.com/store/apps/details?id=scct.demo.performance.test>
- [11] National Instruments. [online]. 2014. vyd. [cit. 2014-05-27]. Dostupné z: <http://www.ni.com/>
- [12] NI LabVIEW Help Resources. [online]. 2012. vyd. [cit. 2014-05-27]. Dostupné z: <http://www.ni.com/gettingstarted/labviewbasics/onlinehelp.htm>
- [13] Port Authority Database. Gibson Research Corporation [online]. 2014. vyd. [cit. 2014-05-27]. Dostupné z: https://www.grc.com/port_80.htm
- [14] RWETC. S.R.O. *I-Lung 2.0 simulator: Documentation*. Jihlava, 2013

- [15] SILVA, Vladimír. Advanced Android 4 games. Berkeley: Apress, c2012, xx, 280s. ISBN 978-1-4302-4059-4
- [16] SOLDÁT, Marek. Slabikář XML. Interval.cz [online]. 09. 05. 2002. [cit. 2014-05-27]. Dostupné z:<http://interval.cz/clanky/slabikar-xml-odkazy/>
- [17] TCP vs UDP. diffen.com. Kate T., Pooja Sehgal a kol. [online]. Diffen LLC, n.d. Web. 27 May 2014. vyd. [cit. 2014-05-27]. Dostupné z: http://www.diffen.com/difference/TCP_vs_UDP

PŘÍLOHY

A	Čelní panel zjednodušené aplikace v LabVIEW.....	37
----------	---	-----------

Příloha A Čelní panel zjednodušené aplikace v LabVIEW

