



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název: Výukový p ípravek s tlakovými senzory
Student: Pavel Bezstarosti
Vedoucí: Ing. Miroslav Skrbek, Ph.D.
Studijní program: Informatika
Studijní obor: Informa ní technologie
Katedra: Katedra po íta ových systém
Platnost zadání: Do konce letního semestru 2017/18

Pokyny pro vypracování

Prove te řešerši IOT technologií a protokol , které dovolí p es TCP/IP protokol zp ístupnit po síti data z výukového p ípravku s tlakovými senzory. P ípravek poskytuje data o poloze, p ípadn pohybu, b emene na ploše na základ ty tlakových senzor . Výukový p ípravek navrhnete a realizujete, opat ete procesorem a komunika ním WiFi modulem a s využitím zvolených modul a knihoven implementujte vhodné IOT technologie a protokoly. Funk nost ov te v demo aplikaci. Rozsah práce up esnete po dohod s vedoucím práce.

Seznam odborné literatury

Dodá vedoucí práce.

prof. Ing. Róbert Lórencz, CSc.
vedoucí katedry

prof. Ing. Pavel Tvrdlík, CSc.
d kan

V Praze dne 13. prosince 2016

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA INFORMAČNÍCH TECHNOLOGIÍ



Bakalářská práce

Výukový přípravek s tlakovými senzory

Pavel Bezstarostí

Vedoucí práce: Ing. Miroslav Skrbek, Ph.D.

12. května 2017

Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce Ing. Miroslavu Skrbkovi, Ph.D. za konzultace a průběžné rady. Dále bych chtěl poděkovat rodině za motivaci a podporu v průběhu studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 12. května 2017

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2017 Pavel Bezstarosti. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Bezstarosti, Pavel. *Výukový přípravek s tlakovými senzory*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Tato bakalářská práce je zaměřena na výrobu výukového přípravku pro práci s IoT technologiemi. Přípravek umožňuje studentům si vyzkoušet napojení předpřipravených programů, nebo vlastních programu na protokoly MQTT a HTTP. Pomocí těchto protokolů je možné přípravek také vzdáleně ovládat. Pro tyto účely je přípravek dále osazen jednoduchými senzory v podobě potenciometrů, tlakových snímačů, tlačítek a fotorezistoru. Aktory zde zastupují LED diody. Tyto komponenty umožňují interakci s uživatelem. Důležitou součástí je dotyková plocha sestavená z tlakových senzorů, která umožňuje demonstraci využití více senzorů pro výpočet umístění břemene na ploše.

Výpočetní část přípravku je postavena na výukové platformě Raspberry Pi, které byl přidán rozšiřující modul obsahující zmíněné senzory a A/D převodník, který v Raspberry Pi chybí. Pro zjednodušení přístupu k datům ze senzorů byl napsán ovladač pro rozšiřující modul ve formě modulu do jádra systému, který zpřístupní data ve formě souborů. To umožňuje čtení pomocí libovolného programovacího či skriptovacího jazyka bez potřebných znalostí senzorů.

Výsledky a použité technologie v této práci jsou popsány v následujícím textu.

Klíčová slova IoT, Raspberry Pi, MQTT, Tlakové senzory, Vzdálené ovládání

Abstract

This bachelor thesis is focused on the production of educational kit for work with IoT technologies. The kit allows students to try to connect pre-programmed programs or custom programs to MQTT and HTTP protocols. Using these protocols, kit can also be remotely controlled. For this purpose, the product is further fitted with simple sensors in the form of potentiometers, pressure sensors, buttons and photoresistor. The actors are represented by LEDs. These components allows interaction with the user. An important part is a touch pad built up of pressure sensors that demonstrates the use of multiple sensors to calculate the load location on the surface.

The computational part of the product is built on the Raspberry Pi learning platform, which has been added the expansion module containing the sensors and the A/D converter missing on Raspberry Pi. To simplify access to sensor data, an extension module driver has been written in the form of a kernel module, which makes data available in the form of files. This allows reading with any programming or scripting language without the necessary knowledge of the sensors.

The results and technologies used in this work are described in the following text.

Keywords IoT, Raspberry Pi, MQTT, Force sensors, Remote control

Obsah

Úvod	1
1 Cíle práce	3
1.1 Rešerše IoT protokolů	3
1.2 Návrh a výroba přípravku	3
2 Rešerše Aplikačních IoT Protokolů	5
2.1 HTTP	5
2.2 MQTT	6
2.3 XMPP	8
2.4 CoAP	8
3 Rešerše Přenosových IoT Protokolů	11
3.1 Bluetooth LE	11
3.2 Wi-Fi	12
3.3 Mobilní připojení	12
3.4 Sigfox	13
4 Rešerše HW	15
4.1 Silové senzory	15
4.2 A/D Převodník	16
4.3 I ² C Komunikace	16
4.4 Raspberry Pi	20
5 Návrh	21
5.1 Ovladač a komunikace	21
5.2 Konstrukce	23
5.3 Elektronika	25
5.4 Výpočet polohy	29

6 Realizace	31
6.1 Rozšiřující modul	31
6.2 AD7997	32
6.3 Ovladač	33
6.4 Java MQTT Klient	35
6.5 Simulace kurzoru	35
6.6 Mosquitto	36
6.7 Webové skripty	36
6.8 NodeRed	36
6.9 Přímý přístup k datům	37
6.10 Konstrukce	38
7 Testování	41
7.1 Rozšiřující modul	41
7.2 Dotyková plocha	41
7.3 Touchpad	42
7.4 Ovladač	42
7.5 Programová výbava	42
Závěr	45
Literatura	47
A Seznam použitých zkratk	49
B Zapojení	51
B.1 Rozšiřující modul	51
B.2 Touchpad	51
C Kompilace	53
C.1 Ovladač	53
C.2 Simulace kurzoru	53
C.3 MQTT Java Klient	53
D Použití	55
D.1 Síťové připojení	55
D.2 Ovladač	55
D.3 Webové skripty	56
D.4 Program pro simulaci kurzoru	57
D.5 NodeRed	57
D.6 MQTT Java Klient	57
E Obsah přiloženého CD	61

Seznam obrázků

0.1	Ukázka práce studenta s přípravkem	2
3.1	Ukázka sigfox modemu	14
4.1	FSR senzor	15
4.2	Příklad zapojení sběrnice I ² C	17
4.3	Ukázka komunikace na I ² C	18
5.1	Návrh propojení programových komponent	22
5.2	Ukázka scriptu v NodeRed prostředí	22
5.3	Návrh komponent	23
5.4	Návrh použití senzoru	24
5.5	Návrh rámečku touchpadu	24
5.6	Blokový návrh zapojení	25
5.7	Napěťový dělič	26
5.8	Zapojení tlačítka	27
5.9	Komunikace s touchpadem	28
5.10	Zpracování polohy	30
6.1	Fotografie Přípravku	31
6.2	Rozšiřující modul	32
6.3	Render úchytu dotekové plochy	39
6.4	Render rámečku pro touchpad	40
6.5	Převodník sběrnic pro touchpad	40
7.1	Testování NodeRed	42

Seznam tabulek

3.1	Ukázka předdefinovaných služeb Bluetooth LE	12
3.2	Ukázka kombinace signálů pro jednotlivé stavy DTMF	13
4.1	Adresy na I ² C sběrnici se speciálním významem	18
6.1	Použité piny	32
6.2	I/O přístup k ovladači	34
6.3	Přehled webových skriptů	37
B.1	Použité piny	51
B.2	Zapojení AD7997	51
D.1	I/O přístup k ovladači	56
D.2	Přehled webových skriptů	57

Úvod

Cílem této práce je rešerše IoT protokolů a vytvoření výukového IoT přípravku s tlakovými senzory pro předmět MI-PVS na ČVUT FIT.

Internet of things neboli IoT jsou zařízení podporující komunikaci mezi ostatními IoT zařízeními nebo počítači skrze internet. Zařízení, která měla komunikační schopnosti, existují již dlouhou dobu, ale až v roce 2012 bylo definováno označení IoT [1].

Veškerá komunikace musí dodržovat předepsaný formát neboli protokol. Protokol je definice pravidel, která umožňují výměnu zpráv mezi zařízeními. U IoT je nutné rozlišit přenosové a aplikační protokoly. Aplikační protokoly určují tvar zpráv na již existujícím přenosovém mediu, zatímco protokoly přenosových medií určují technologii a formát přenášení zprávy.

Přípravek umožňuje studentům vyzkoušet si práci s IoT zařízením. K dispozici mají programové vybavení pro komunikaci přes MQTT a HTTP, kde si mohou vyzkoušet napojení vlastních programů na reálném IoT zařízení. Přípravek obsahuje krom tlakových senzorů, pomocí kterých je možné vypočítat polohu zdroje síly působící na měřenou plochu, také senzory a aktory pro interakci s uživatelem.

Tato práce pojednává o rešerši IoT protokolů, návrhu a výrobě výukového přípravku postaveného na výukové platformě Raspberry Pi. Během práce bylo nutné naprogramovat nástroje pro práci s přípravkem. Rozšířit systém Raspberry Pi o programy potřebné pro IoT. Navrhnout rozšiřující modul pro Raspberry Pi, který umožňoval čtení ze senzorů a navrhnout díly, které poté byly vytisknuty na 3D tiskárně. V závěru práce byla otestována funkčnost jednotlivých komponent.



Obrázek 0.1: Ukázka práce studenta s přípravkem

Cíle práce

1.1 Rešerše IoT protokolů

Kolem nás se nachází mnoho zařízení, která nám zjednodušují každodenní život. V posledním desetiletí začala většina těchto zařízení mezi sebou komunikovat. To přináší velké množství výhod, ale má to i své stinné stránky v podobě nedostatečně implementovaných bezpečnostních funkcí.

Zařízení musí dodržovat předem určené standardy neboli protokoly, aby se mohla spojit s jiným zařízením. V první části této práce se nachází stručná rešerše IoT protokolů. Protokoly jsou zde dělené na aplikační a přenosové. Přenosové protokoly definují medium, po kterém jsou data přenášena. Aplikační protokoly definují přenos dat na již existujících datových spojkách.

1.2 Návrh a výroba přípravku

Druhá část práce se zabývá návrhem a výrobou přípravku pro předmět MI-PVS vyučovaný Fakultě informačních technologií na Českém vysokém učení technickém v Praze. Součástí přípravku bude soubor senzorů a aktorů, které umožňují interakci s okolním světem. Dále bude umožňovat demonstraci využití více tlakových senzorů pro výpočet zdroje síly na dotykovou plochu.

Přípravek umožní studentům vyzkoušet si předpřipravené nebo své vlastní programy ve spojení s IoT protokoly na reálném zařízení.

Rešerše Aplikačních IoT protokolů

2.1 HTTP

HTTP neboli Hyper Text Transfer Protocol je protokol pro výměnu hypertextových dokumentů. Je postaven na TCP portu 80. Jedná se o jeden z nejvíce používaných protokolů na internetu. Ve verzích 1.0 a 1.1 byl převážně textově orientovaný a od verze 2.0 [2] je plně binární pro rychlejší zpracování. Protokol disponuje dotazovacími metodami:

- **GET** - Požadavek na poslání konkrétního objektu ze serveru.
- **HEAD** - Podobná metoda jako GET, ale poskytne pouze metadata o požadovaném objektu.
- **POST** - Metoda sloužící pro přenos od uživatele k serveru. Následně se chová stejně jako metoda GET. Využívá se například u posílání dat z webformu, aby se neposílaly jako parametry adresy.
- **PUT** - Slouží k nahrání souboru na server.
- **TRACE** - Slouží ke zjištění modifikace hlavičky po cestě, například při průchodu Proxy serveru.
- **DELETE** - Metoda sloužící ke smazání objektů na serveru. Moc se nepoužívá.

Jedná se o nenáročný, jednoduchý protokol pro výměnu dat, a proto je možné ho využívat i k předávání dat z IoT. Například: máme teploměr, který je připojen pomocí Wi-Fi do internetu a ke svému předávání používá HTTP. Teploměr každých 10 minut změří teplotu a pošle pomocí *GET* požadavků teplotu jako parametr cesty. Na straně serveru je možné pro zpracování zvolit

již libovolný jazyk např. PHP, NodeJs. Pro čtení dat ze serveru se musí teplo-
měr aktivně dotázat, což postačí pro načtení konfigurace, ale je nevhodné pro
okamžité ovládání.

2.1.1 Zabezpečení

Protokol obsahuje dva mechanismy ověření.

- **Basic** - Základní ověření, kde se uživatelské jméno i heslo posílá v ne-
zašifrované či hašované formě.
- **Digest** - Namísto od základního ověření se přenášená data hašují s uni-
kátním číslem, které dodá server. Hašovací funkcí je MD5, která je již
zastaralá a prolomená na generování kolizí.

HTTP protokol je možné obalit do TLS nebo SSL vrstvy, také označované
jako HTTPS komunikujícím na TCP portu 443. V HTTPS je možné bezpečně
používat *Basic* ověření z důvodu šifrování celého přenosu.

2.2 MQTT

MQTT neboli Message Queue Telemetry Transport je nenáročný protokol pro
okamžitý a spolehlivý sběr dat ze vzdálených senzorů. Protokol MQTT vznikl
v IBM již v roce 2001 [3]. Je postavený na modelu publikace a odebírání
příspěvků. Každý příspěvek má při publikaci předmět, který vypadá jako vir-
tuální cesta a ostatní zařízení si mohou zažádat o odběr.

Pro provoz MQTT protokolu je nutné provozovat tzv. Broker, neboli soft-
ware, který se chová jako prostředník při komunikaci vzdálených zařízení. Pro-
tokol má malé datové nároky a je postaven na TCP/IP. Existuje i varianta
MQTT-SN, která nevyžaduje TCP jako přenosový protokol. Pro vytvoření
komplexní sítě je možné propojit více Brokerů.

Protokol disponuje následujícími metodami:

- **Connect** - Slouží k navázání spojení se serverem.
- **Disconnect** - Slouží k ukončení spojení se serverem.
- **Subscribe** - Slouží k žádosti o odebírání předmětů.
- **UnSubscribe** - Slouží k žádosti o zrušení odebírání předmětů.
- **Publish** - Slouží k publikování příspěvku.

2.2.1 Odebírání příspěvků

Odebírání nemusí vždy být konkrétní, ale může využívat tzv. wildcard, což umožňuje odebírání více cest, kterým odpovídá požadavek odběru. Symbol `#` slouží k odběru všech potomků předmětů a používá se na konci předmětu viz. `/dum/mistnosti/kuchyne/#` pro odebírání všech příspěvků z místností kuchyně např. příspěvku `/dum/mistnosti/kuchyne/teplomer/hodnota`. Dále je možné použít symbol `+` pro náhradu části předmětu viz. `/dum/mistnosti/+/teplomer/hodnota` pro odběr teplot ze všech místností.

2.2.2 Zabezpečení

Protokol MQTT podporuje přihlašování pomocí uživatelského jména a hesla. Po přihlášení je možné na Brokeru nastavit použití ACL pro nastavení přístupu k jednotlivým předmětům. Je možné určit, jaké uživatelské jméno a právo, kde publikovat a co odebrat. Konkrétní nastavení záleží na implementaci Brokerů.

Komunikace mezi zařízením a Brokerem není šifrovaná, proto je vhodné zvláště při komunikaci přes cizí síť - jako je internet - celou komunikaci šifrovat. Je možné využít SSL nebo TLS vrstvy pro šifrování při přenosu.

2.2.3 Quality of Service

QoS slouží pro nastavení spolehlivosti doručení zprávy. Při nespolehlivém přenosovém médiu programátor nemusí řešit doručení zprávy a může vše přenechat MQTT protokolu. QoS má tři stupně nastavení. Režie protokolu stoupá s třídou QoS.

- **QoS 0** - Doručení příspěvku není garantované.
Vhodné pro informační periodické příspěvky.
- **QoS 1** - Doručení příspěvku je garantované minimálně jednou.
Vhodné pro důležitější příspěvky, kde nevadí opakování - např. teplota.
- **QoS 2** - Doručení příspěvku je garantované právě jednou.
Vhodné pro důležité příspěvky.

2.2.4 Trvalý příspěvek

Každý příspěvek je možný označit *retain* značkou. Příspěvek zůstane na Brokeru, dokud není přepsán dalším příspěvkem o stejném předmětu.

Trvalý příspěvek je vhodné použít, pokud u odběratelů není zaručeno, že poslouchají, když je zpráva publikována, a mohla by zaniknout. Při použití je příspěvek poslán odběratelům ihned po žádosti o odběr, ne až s její další publikací.

2.2.5 Závěť

Jedná se o speciální typ příspěvků, který je zveřejněn při odpojení zařízení nebo při poruše spojení. O zveřejnění se stará Broker.

Závěť při kombinaci s trvalými příspěvky může sloužit k nastavení stavu zařízení při odpojení či ztrátě spojení.

Například: `/dum/mistnosti/kuchyn/teplomer/stav -> offline`

2.3 XMPP

XMPP neboli Extensible Messaging and Presence Protocol je protokol pro decentralizovanou výměnu zpráv a metadat. Je postavený na TCP/IP spojení a pro serializaci zpráv používá XML formát. Původní účel byl výměna zpráv mezi uživateli, ale dá se použít i pro komunikaci se vzdálenými zařízeními. Protokol je relativně nenáročný, ale pro IoT, kde je nutné šetřit s paměťovými a komunikačními zdroji, je nevhodný na použití. Existují zjednodušené verze pro použití na limitovaných zdrojích, například XMPP-IoT.

Na rozdíl od MQTT všechna zařízení nebo uživatelé mají své unikátní ID, které je sestaveno z uživatelského jména a domény serveru, kde jsou registrovaní **uživatel@server.cz**, připomíná e-mailovou adresu. Doména serveru v ID je povinná z důvodu decentralizace.

2.3.1 Decentralizace

XMPP servery jsou decentralizované a s ostatními servery komunikují pouze v případě nutnosti výměny dat. Pro komunikaci se využívá TCP spojení na portu 5269. V případě nutnosti posílání zprávy od uživatele *pepa@a.com* uživateli *franta@b.com* putuje zpráva následovně.

- **1** - Zpráva je poslána z XMPP klienta uživatele Pepa na server a.com
- **2** - Zpráva je poslána ze serveru a.com na server b.com
- **3** - Zpráva je poslána ze serveru b.com uživateli Franta

Při výpadku serverů není nijak narušena komunikace mezi uživateli ostatních domén.

2.4 CoAP

CoAP neboli Constrained Application Protocol je protokol pro výměnu zpráv mezi zařízeními. Pro výměnu zpráv používá UDP nebo jakákoliv jiná data diagramové orientované spojení jako SMS zprávy. Protokol je strukturou podobný HTTP, ale je méně náročný[4]. Vhodné, když v zařízení není implementován TCP/IP zásobník.

Při nutnosti komunikace dvou zařízení komunikace probíhá na přímo bez prostředníka. Zařízení se jeví jako server a vzdáleně zařízení z něho čte či do něho zapisuje. To může způsobovat problémy, pokud je v cestě NAT.

- **GET** - Slouží k získání dat identifikované pomocí URI ze zařízení.
- **PUT** - Slouží k nahrání dat identifikované pomocí URI na zařízení.
- **POST** - Slouží k získání dat identifikované pomocí URI ze zařízení, ale dodatečná data posílá v hlavičce dotazu nikoliv v URI.
- **DELETE** - Slouží ke smazání dat identifikované pomocí URI na zařízení.

2.4.1 QoS

Způsob doručování zpráv je zde jednodušší než u MQTT, protože jsou přítomné pouze dva stavy. Zpráva může být pouze „Potvrzovaná“ nebo „Nepotvrzovaná“.

2.4.2 Zabezpečení

Z důvodu absence TCP není možné použít SSL či TLS jako u jiných protokolů. Proto je nutné pro zabezpečení komunikace použít DTLS, které poskytuje srovnatelné zabezpečení u UDP jako TLS u TCP.

Rešerše Přenosových IoT protokolů

3.1 Bluetooth LE

Bluetooth LE neboli Low Energy je technologie pro vytvoření osobní bezdrátové sítě pro zařízení s limitovaným zdrojem energie. Není zpětně kompatibilní s jinými bluetooth standardy, ale využívá stejnou frekvenci. To umožňuje použití jedné antény pro Bluetooth i Bluetooth LE. Vzniklo z technologie Wibree vyvíjeného společnostmi Nokia a Nordic Semiconductor [5].

Díky podpoře v nových mobilních telefonech je vhodné pro přenos dat z nositelné elektroniky. Dosah i rychlost jsou limitovány z důvodu energetické úspory. Zařízení zpravidla komunikují v periodických intervalech a většinu doby jsou v energeticky úsporném režimu. To umožňuje provozovat zařízení napájené z baterií po několik let [6].

3.1.1 Přenos dat

Periferie je označena jako GATT server a přistupuje k ní GATT klient, což může být například mobilní telefon. Komunikace probíhá v předem dohodnutých intervalech. Klient vždy zahajuje komunikaci vůči serveru.

Data jsou organizována do logických skupin neboli služeb. Služby jsou identifikovány 128bitovým klíčem a každá služba definuje určitý okruh dat neboli "charakteristiky". Charakteristika popisuje přístupový bod dat. Mnoho služeb je předdefinováno a standardizováno pro jednotný přístup k datům (viz. tabulka 3.1).

Jméno služby	Typ služby	UID
Monitor Baterie	org.bluetooth.service.battery_service	0x180F
Hladina Glukózy	org.bluetooth.service.glucose	0x1808
Krevní tlak	org.bluetooth.service.blood_pressure	0x1810
Okamžitý Alarm	org.bluetooth.service.immediate_alert	0x1802
...

Tabulka 3.1: Ukázka předdefinovaných služeb Bluetooth LE [7]

3.2 Wi-Fi

Zkratka Wi-Fi slučuje několik standardů definovaných v IEEE 802.11 pro přenos dat v rámci bezdrátové lokální sítě. Rychlost a dosah závisí na konkrétním standardu. Je vhodná pro připojení přenosných zařízení jako telefon či notebook, pro přenos dat z domácí přípojky k poskytovateli internetu.

Je nevhodná pro přenos dat u zařízení s limitovanou zásobou energie nebo u zařízení, kde je nepraktické periodické dobíjení. Tam, kde nejsme limitováni energií, je její použití výhodné, zvláště díky již existujícímu pokrytí. Zařízení jsou připojena v síti a mohou komunikovat navzájem nebo posílat data směrem do internetu. Pro přenos dat je nutné použít aplikační protokol, například protokoly popsané v předchozí kapitole.

3.2.1 Zabezpečení

Přenos po Wi-Fi je možné zabezpečit již na první ISO/OSI vrstvě, ale jedná se o zabezpečení přenosu od zařízení k AP. Je vhodné použít zabezpečení i ve vyšší vrstvě jako DTLS či TLS.

3.3 Mobilní připojení

Pro přenos dat ze vzdálených míst, která je nemožné připojit k internetu, je využití telefonního připojení téměř jediná možnost. Výhoda je, že signál pokrývá téměř vše. Nevýhoda je, že za každou SIM kartu je nutné platit.

3.3.0.1 Telefonní Hovor

Některá zařízení si vystačí s přenosem dat pomocí modulace zvukového signálu do telefonního hovoru. Jedna z možností je využití DTMF neboli dual tone multi frequency signaling. Jedná se o poměrně starou, ale ověřenou technologii, která je k nalezení u komunikace mezi telefonem a ústřednou pro navázání spojení nebo u zabezpečovacích stanic.

Signál je generován pomocí dvou sinusových signálů o dané frekvenci. Můžeme rozlišit 16 diskrétních stavů (viz. tabulka 3.2).

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Tabulka 3.2: Ukázka kombinace signálů pro jednotlivé stavy DTMF [8]

3.3.0.2 SMS

SMS slouží k posílání krátkých textových zpráv bez háčeků a čárek do délky 160 znaků. Tato komunikace je možná využít pro jednoduché ovládání zařízení nebo sběr dat ze senzorů. Výhodné je, že potenciální ovladač mají všichni vždy u sebe.

CoAP podporuje pro přenos libovolné datadiagramové orientované spojení. Proto je možné využít tento aplikační protokol pro přenos dat pomocí SMS [9].

3.3.0.3 Datová komunikace

V případě datového spojení je to podobné jako u připojení k internetu pomocí Ethernetu nebo Wi-Fi. Je nutné využít aplikačního protokolu pro přenos dat. Využívá se především u zařízení, která se často pohybují. Vhodné pro sledování aut nebo vzdálených senzorů s častou periodickou komunikací či větším datovým provozem, například fotopast.

3.4 Sigfox

Sigfox je technologie pro posílání velmi malého objemu dat ze vzdálených senzorů. Využívá nelicencovaného pásma 868 MHz. V České republice síť buduje T-Mobile s cílem pokrýt celé území [10].

Každý přijímač je vybaven unikátním identifikátorem, který slouží jako adresa.

3.4.1 Přenos Dat

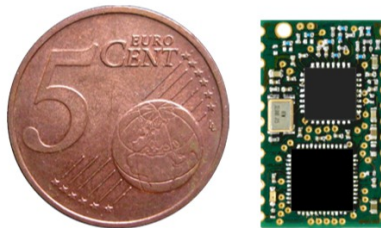
Technologie umožňuje přenos velice malých objemů dat o maximální velikosti 12 bajtů. Za den je možné poslat maximálně 140 zpráv. Pro odesílání dat z jednoduchých senzorů je kapacita i množství dostatečné.

Technologie původně zamýšlela jednocestnou výměnu zpráv, ale možnost přijímání zpráv byl přidána. Velikost přijímané zprávy je maximálně 8 bajtů a za den mohou být přijaté maximálně 4 zprávy. Problém s přijímáním je, že vysílač neví, kdy je přijímač připraven a poslouchá.

3. REŠERŠE PŘENOSOVÝCH IoT PROTOKOLŮ

Rychlost komunikace je 100 b/s při délce odesílání 2 - 5 sekund, záleží na délce zprávy. Data jsou při přenosu podepsána a hašovaná, tím je zaručen původ a neporušenost zprávy.

K datům je přístupováno přes REST API vůči serverům operátora [11].



Obrázek 3.1: Ukázka sigfox modemu. (Obrázek převzat zdroj [12])

Rešerše HW

4.1 Silové senzory

Působením síly na senzor se mění jeho vnitřní elektrické vlastnosti, které je možné měřit. U FSR (Force Sensitive Resistor) se mění vnitřní odpor, který je možné použitím napětového děliče převést na poměr napětí. Signál je dále vhodné dle potřeb zesílit a poté přechíst A/D převodníkem.

Rozsah senzoru je závislý na jeho provedení a je možné ho nalézt v dokumentaci senzorů.

Pro převod vnitřního odporu na sílu působící na senzor je nutné znát převodní vztah, který je dán výrazem či převodní tabulkou, která je dodaná výrobcem v datasheetu.



Obrázek 4.1: FSR senzor (Obrázek převzat zdroj [13])

4.2 A/D Převodník

A/D převodník slouží k převodu spojité (analogové) hodnoty na hodnotu diskrétní (digitální), odtud jméno A/D převodník. Digitální procesory nedokáží s analogovou hodnotou počítat přímo, proto je nutné vstupní signál nejprve vhodně zpracovat, což může být zesílení a filtrace, a poté převést na diskrétní hodnotu.

Proces vzorkování se skládá z rozdělení časové osy na stejně dlouhé úseky a v každém úseku se změří jeden vzorek. Vzorkovací frekvence rozhoduje o kvalitě převedeného signálu, udává se v (k)Hz. Pro zaznamenání signálu o maximální frekvenci f je dle Shannonova teoremu nutné použít vzorkovací frekvenci více než $2f$ [14]. Signál by měl být před zpracováním zbaven pomocí dolnoproputního filtru o frekvence větší než $f/2$ pro zamezení aliasingu. Hudební CD má signál o maximální frekvenci 22 kHz, které už jsou pro lidské ucho neslyšitelné, proto je vzorkovací frekvence 44.1 kHz.

Proces kvantování převádí amplitudu v rámci jednoho vzorku na diskrétní hodnotu. Rozsah hodnot udává rozlišení převodu. Např. 8bitové rozlišení udává 256 různých stupňů pro změřenou hodnotu.

4.2.1 A/D Převodník s aproximačním registrem

Funguje na principu porovnávání měřené hodnoty s D/A převodníkem pomocí komparátoru. Bity D/A převodníku jsou nastavovány postupně od bitu s nejvyšší vahou. Pokud po nastavení bitu je napětí na D/A převodníku vyšší než na vstupu, bit se resetuje na 0, a pokračuje se dalším bitem. Tento způsob je rychlejší než u čítacího a doba převodu je konstantní. Je to nejčastější typ A/D převodníku u víceúčelových procesorů. Výhodou tohoto typu je jeho jednoduchost.

4.2.2 Paralelní A/D Převodník

Funguje na principu paralelně zapojených operačních zesilovačů, kde na jednom vstupu mají měřené napětí, a na druhém vstupu má každý svou referenční hodnotu vytvořenou přes skupinu sériových odporů. Výhodou tohoto převodníku je jeho rychlost, ale nevýhodou jeho náročnost při vyšším rozlišení.

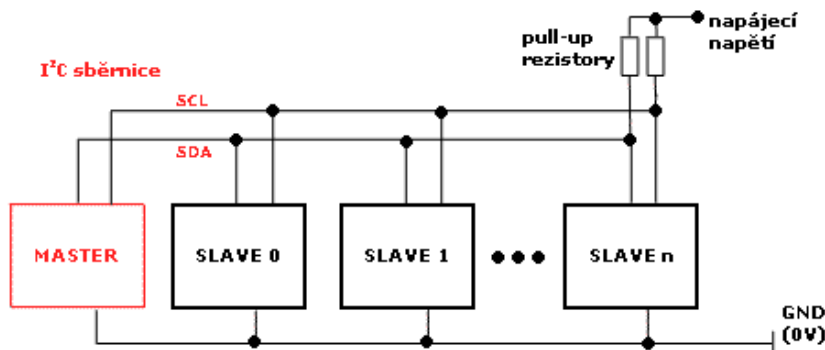
4.3 I²C Komunikace

I²C neboli Internal-Integrated-Circuit je sběrnice vytvořena v 80. letech firmou Philips pro jednoduché propojení integrovaných obvodů v rámci jednoho zařízení. I²C je sběrnice typu multi-Master a zároveň multi-Slave, proto se hodí pro komunikaci s integrovanými obvody typu RTC, EEPROM, A/D převodníky, nenáročné LCD displeje, Akcelerometry, kterých je v rámci jednoho zařízení více, a mohou se kombinovat.

Sběrnice se skládá ze dvou vodičů, které jsou v klidovém stavu v logické hodnotě 1 pomocí *Pull-Up* rezistoru. Vodič serial data (SDA) pro datový signál a serial clock (SCL) pro hodinový signál (viz. obrázek 4.2). Rychlost sběrnice je v základu 100 kHz, ale může být 400 kHz nebo 1 MHz. Na rychlosti závisí i výběr použitých *Pull-Up* odporů, např. pro 100 kHz se používají odpory o hodnotě 4k7, neboli 4700 Ω pro vyšší rychlosti odpor klesá kvůli kompenzaci kapacity vodičů.

Standardní adresace je 7bitová při teoretickém maximálním počtu zařízení 127 nebo v rozšířené 10bitové adresaci 1024. V praxi je zařízení méně, protože možnost nastavit celou adresu pomocí 7 nebo 10 pinů je na integrovaných obvodech nemožné kvůli nedostatku pinů. Proto se používá k nastavení do 3 pinů, takové zařízení má od výrobce pevně daný prefix adresy a piny nastavují poslední bity adresy. V tomto případě je možno použít maximálně 8 stejných čipů, aby nevznikla adresní kolize na sběrnici. U některých integrovaných obvodů je možné kvůli nízkému počtu pinů nastavit adresu zápisem do paměti senzoru. Pro 10bitovou adresaci se používají speciální adresy, na kterých poslouchají 10bitové adresované integrované obvody. Další rezervované adresy (viz. tabulka 4.1).

Sběrnice I²C se používá jako doplněk jiných sběrnic pro jednoduchou konfiguraci. Je možné ho nalézt v Display Data Channel (DDC)[15] u HDMI či DVI, nebo u PCI(e) v modifikaci SMBus.



Obrázek 4.2: Příklad zapojení sběrnice I²C (Obrázek převzat zdroj [16])

4.3.1 Vyhledávání zařízení

Při standardní 7bitové adresaci je maximálně 127 adres, na kterých mohou poslouchat slave jednotky, proto je možné vyzkoušet všechny adresy postupně v krátkém čase. Master pošle start bit a adresu, se kterou chce komunikovat. Pokud se na sběrnici objeví ACK, tak ví, že adresa je platná. Tento princip se nedá použít u sběrnic s delší adresací. Například sběrnice 1-Wire od Dallas, která má 64bitovou adresu [18].

Slave adresa	R/W Bit	Význam
0000 000	0	Broadcast
0000 000	1	Start bajt
0000 001	x	CBUS adresa
0000 010	x	Rezervováno
0000 011	x	Rezervováno
0000 1xx	x	HS-Mode
1111 1xx	x	Rezervováno
1111 0xx	x	10bitové adresování

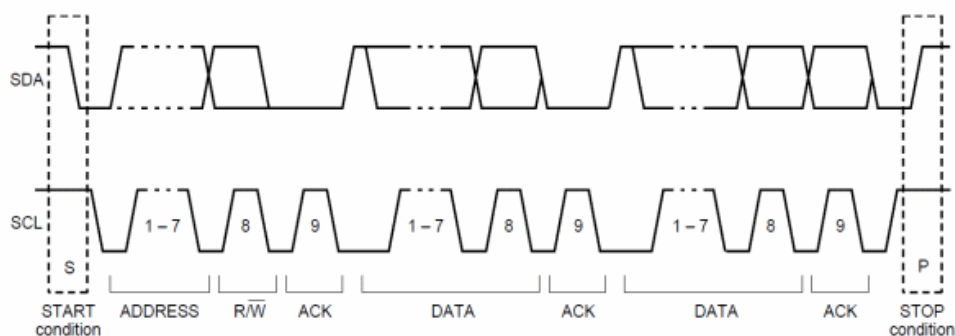
Tabulka 4.1: Adresy na I²C sběrnici se speciálním významem [17]

4.3.2 Arbitrace

Za normálního stavu na sběrnici komunikuje vždy jen jeden Master. Pokud je aktivních více Masterů, tak arbitrační procedura rozhodne o vítězi. Arbitrace probíhá na SDA vodiči. Master zapisuje svá data na sběrnici a přitom sleduje stav sběrnice. Pokud ze sběrnice přečte jiná data, prohrál a musí ukončit komunikaci.

4.3.3 Ukázka komunikace

Komunikaci vždy zahajuje Master start bitem a adresou. Rozlišuje, zda se bude číst nebo zapisovat. To řídí R/W bit, který se posílá za adresou (viz. obrázek 4.3). U čtení i u zápisu je možné přistupovat pouze k jednomu registru nebo k více po sobě adresovaným registrům.

Obrázek 4.3: Ukázka komunikace na I²C (Obrázek převzat zdroj [17])

4.3.3.1 Zápis

1. Master pošle na sběrnici start bit
2. Master pošle na sběrnici 7bitovou adresu
3. Master pošle na sběrnici R/W bit - v našem případě 0
4. Pokud na oslovené adrese někdo poslouchá, nastaví ACK bit
5. Master pošle na sběrnici 8bitovou adresu registru, od které bude zapisovat
6. Slave pošle na sběrnici ACK bit
7. Master pošle na sběrnici 8 bitů, které chce zapsat do zvoleného registru
8. Slave pošle na sběrnici ACK bit
9. Pokud jsou další data k zápisu, zpět na krok 7. (Slave automaticky inkrementuje adresu zápisu)
10. Master pošle na sběrnici stop bit a ukončí komunikaci

4.3.3.2 Čtení

1. Master pošle na sběrnici start bit
2. Master pošle na sběrnici 7bitovou adresu
3. Master pošle na sběrnici R/W bit - v našem případě 1
4. Pokud na oslovené adrese někdo poslouchá, nastaví ACK bit
5. Master pošle na sběrnici 8bitovou adresu registru, od kterého bude číst
6. Slave pošle na sběrnici ACK bit
7. Slave pošle na sběrnici 8 bitů ze čtené adresy
8. Slave pošle na sběrnici ACK bit
9. Pokud jsou další data ke čtení, zpět na krok 7. (Slave automaticky inkrementuje adresu čtení)
10. Master pošle na sběrnici stop bit a ukončí komunikaci

4.4 Raspberry Pi

Raspberry Pi je malý počítač o rozměrech platební karty, vyvinutý Raspberry Pi Foundation primárně pro výukové účely, s cílem zkvalitnit výuku informatiky na školách. Díky své nízké ceně a vysoké univerzálnosti našel umístění v mnoho projektech od multimediálních center po senzory připojené do IoT. Pro jeho nízkou spotřebu bylo napojeno přes USB kabel.

4.4.1 Operační systém

Primární operační systémem uzpůsobený pro chod na Raspberry Pi je Raspbian, který vychází z linuxové distribuce Debian. Existuje mnoho dalších kompatibilních operačních systémů, většina z nich je postavena na linuxovém jádře, ale naleznou se i výjimky jako Windows 10 IoT nebo FreeBSD.

4.4.2 Verze

Původní Raspberry Pi se vyrábělo ve dvou verzích - A a B. Největší rozdíly mezi verzemi byly absence síťového konektoru, více USB portů a menší RAM. Později vyšly verze A+ a B+, které obsahovaly rozšířený 40pinový konektor pro rozšíření. U verze Raspberry Pi 2 byla posílena RAM a přidán vícejádrový procesor Cortex-A7 při 900MHz. U poslední verze Raspberry Pi 3 bylo přidáno Wi-Fi a Bluetooth rozhraní zabudované přímo na desce spolu se silnějším procesorem. Byly vydány i zjednodušené verze Raspberry Pi Zero a Raspberry Pi ZeroW, které vycházely z původní verze za cenu kolem 5 USD.

4.4.3 Programování

Jelikož byl projekt primárně určen pro výuku informatiky, je v Raspbianu možné programovat v mnoha programovacích či skriptovacích jazycích. Pro lidi, co nemají zkušenosti s normálním programováním, je tu jazyk Scratch, který využívá spojování grafických bloků. Pokročilejší uživatelé mohou využít Pythonu, Javy, C, C++, Bash, PHP, Perl...

4.4.4 Rozšíření HW

Rozšíření se provádějí přes konektory osazené na desce. Je možné připojit kameru do CSI (Camera serial interface) konektorů, nebo displej do DPI (Display parallel interface) konektoru. Na 40pinovém konektoru se nachází krom GPIO (General purpose input output) také výstupy sběrnic I²C, UART a SPI, které slouží pro připojení rozšiřujících desek.

Návrh

Účelem přípravků je demonstrace a využití IoT zařízení a senzorů k nim připojených na předmětu MI-PSV FIT ČVUT. Vedoucí práce dal požadavek vytvořit plochu, která bude schopna detekovat pozici břemene na ní umístěné. Pro srovnání bude přípravek obsahovat i klasický touchpad z notebooku.

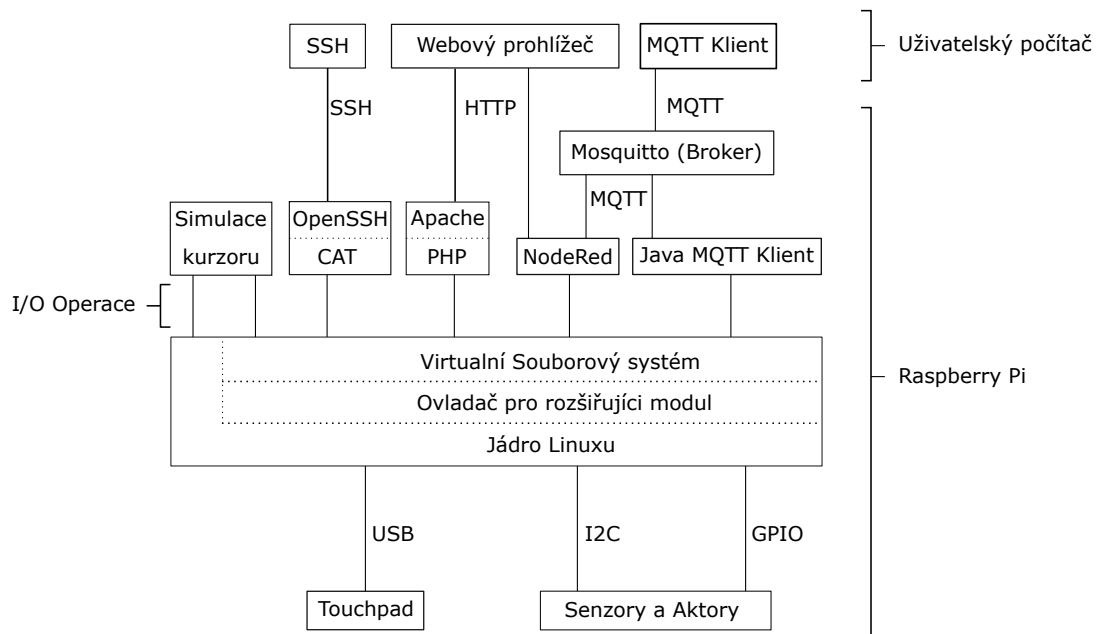
Přípravek bude dále osazen dalšími senzory a aktory, které umožní získávat data z okolí.

- Dotyková plocha měřící na základě tlakových senzorů
- Touchpad z notebooku
- 2× potenciometr
- 2× LED
- 2× Tlačítko
- Fotorezistor
- Konektor pro budoucí rozšíření

Pro snadné spojení s okolím bude vybaven Wi-Fi modulem, který umožní vzdálenou správu i sběr dat z přípravku.

5.1 Ovladač a komunikace

Pro jednodušší a univerzálnější používání přípravků je čtení dat ze senzorů a jejich odesílání rozděleno do více vrstev.



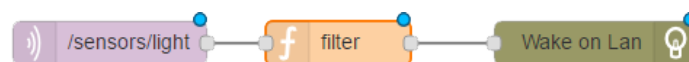
Obrázek 5.1: Návrh propojení programových komponent

5.1.1 Komunikace

Přenos dat k uživateli bude využíván IoT protokol MQTT. O čtení dat z ovladače se stará Java MQTT Klient, který odesílá data do MQTT brokeru, ke kterému musí být uživatel připojen.

Tato metoda vyžadovala nestandardní programovou výbavu na uživatelské straně, proto může být občas nevhodná. Mezi další možnosti patří předpřipravené webové skripty nebo SSH a pomocí programu *cat* je možné vyčíst konkrétní hodnoty.

Softwarová výbava bude obsahovat *NodeRed* prostředí, které umožňuje jednoduché skriptování pomocí spojování diagramu ve webovém prohlížeči. Podporuje MQTT protokol i přístup k souborům.



Obrázek 5.2: Ukázka skriptu v NodeRed prostředí.

5.1.2 Ovladač

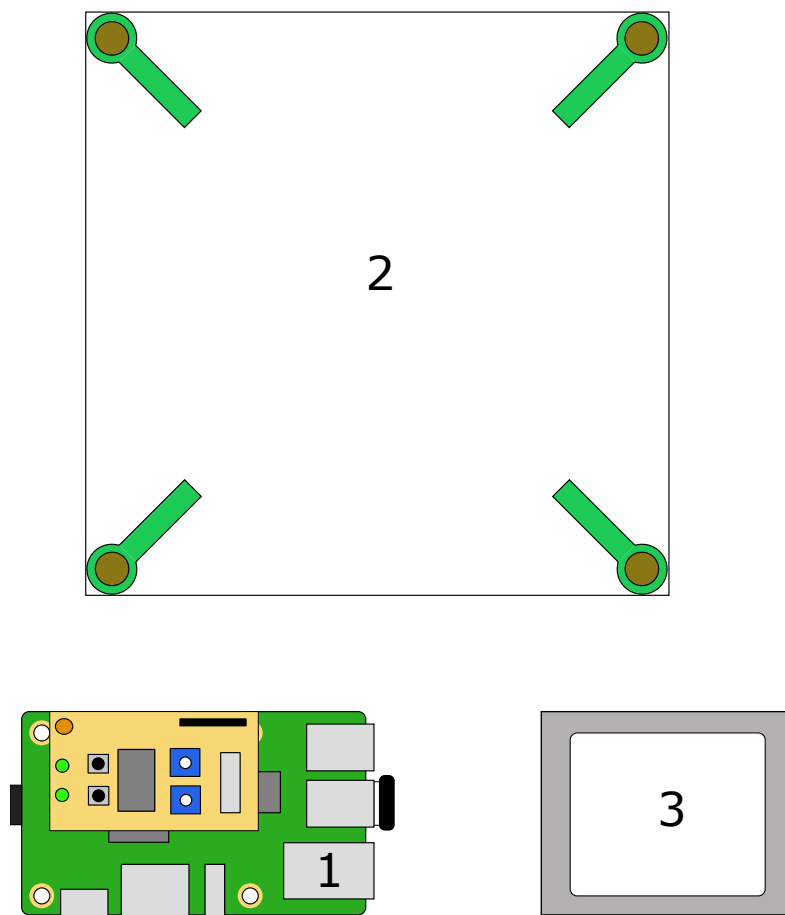
Komunikace se senzory a aktory bude probíhat na úrovni jádra operačního systému a uživateli jsou přístupné konkrétní funkce v podobě virtuálních souborů, ze kterých může číst či zapisovat. Tímto odpadá nutnost složitého na-

stavování senzorů a sběrnice. V případě, že uživatel potřebuje přímý přístup k senzorům či aktorům, postačí vyjmout modul z jádra operačního systému. Uživatel může využít přístup na nižší úrovni.

Výhoda použití ovladače, který ke komunikaci využívá virtuální souborový systém, je možnost použití libovolného programovacího či skriptovacího jazyka. Jediné, co musí podporovat, jsou IO operace, které podporuje téměř každý jazyk.

5.2 Konstrukce

Přípravek bude sestaven na dřevěné desce. Konstrukce musí být uzpůsobena pro snadné přenášení a zvýšenou odolnost, proto bude vše přivrtáno.

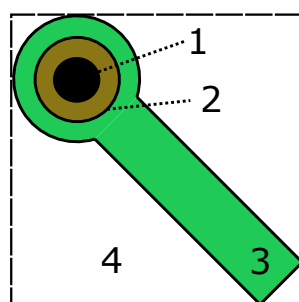


Obrázek 5.3: Návrh komponent. 1 Raspberry Pi + rozšiřující modul, 2 Měřicí dotyková plocha, 3 Kontrolní touchpad

5.2.1 Dotyková plocha

Jeden z hlavních požadavků přípravku je lokace předmětů na ploše pomocí tlakových senzorů. Plocha bude vytvořena z plexiskla ve tvaru čtverce. Pro detekci objektů na ploše se využijí rozdílně působící síly v rozích plochy na tlakové senzory, které se nachází mezi plochou a dřevěnou deskou.

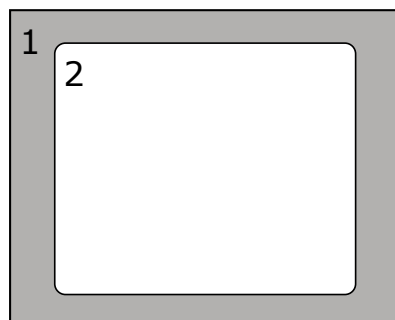
Vedoucí práce dodal tlakové senzory FSR-402. Po otestování bylo zjištěno, že je nutné mezi dotykovou plochu a senzory použít distanční podložku. Ta musí ideálně pokrýt co největší plochu na měřicí části senzorů, ale nesmí ji přesahovat. V takovém případě tlakový senzor dává nepoužitelné výsledky.



Obrázek 5.4: Návrh použití senzoru. 1 Distanční podložka, 2 Měřicí plocha senzorů, 3 Tlakový senzor, 4 Plexisklová deska

5.2.2 Touchpad

Jeden z dalších požadavků vedoucího práce bylo začlenění touchpadu z notebooku Asus EEE. Pro upevnění na podkladovou desku je nutné navrhnout rámeček, ve kterém bude touchpad zapuštěn. Rámeček se následně uchytí k desce pomocí šroubů.

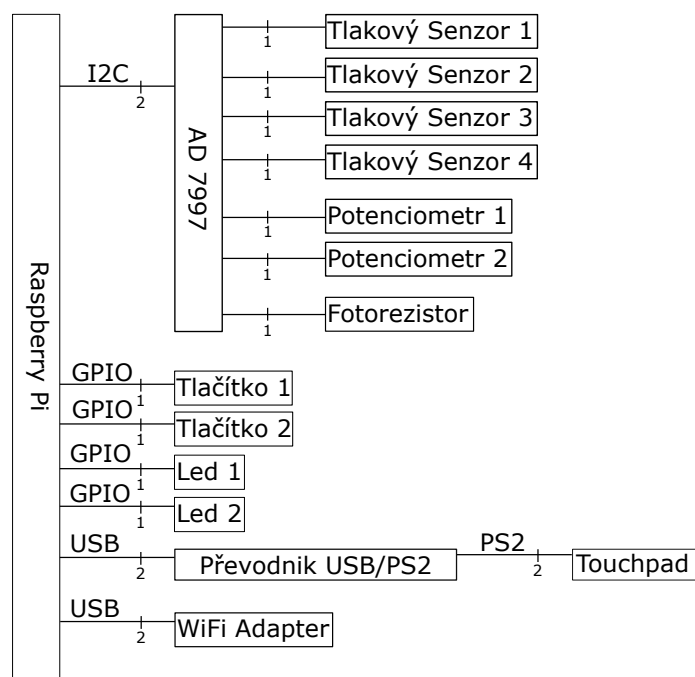


Obrázek 5.5: Návrh rámečku touchpadu. 1 Rámeček, 2 Touchpad

5.2.3 Raspberry Pi

Pro zpracování dat a komunikaci bylo zvoleno Raspberry Pi s Wi-Fi adaptérem. Sensory a A/D převodník budou umístěny na rozšiřujícím modulu, který bude umístěn na GPIO patici Raspberry Pi. Samotné Raspberry Pi bude uchyceno pomocí distančních sloupků.

5.3 Elektronika



Obrázek 5.6: Blokový návrh zapojení

5.3.1 Měření odporu

Činnost všech použitých senzorů je založena na změně vnitřního odporu. Pro měření odporů je nutné sestavit tzv. napěťový dělič, který vznikne přidáním referenčního odporu a aplikováním napětí (viz. obrázek 5.7). Napětí na V_{OUT} se vypočítá pomocí vzorce:

$$V_{\text{OUT}} = \frac{R_1}{R_1 + R_2} \cdot V_{\text{CC}} \quad (5.1)$$

kde V_{CC} je vstupní napětí

V_{OUT} je výstupní napětí

R_1 je hodnota odporu R_1

R_2 je hodnota odporu R_2

Výsledné napětí je již možné změřit pomocí A/D převodníku a následně spočítat odpor senzorů. Nahradíme senzor za R_2 , pak úpravou vzorce dostaneme, že odpor senzoru je

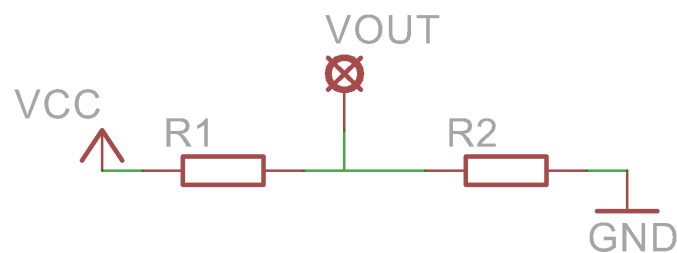
$$R_{\text{SENSOR}} = \left(\frac{V_{\text{cc}}}{V_{\text{out}}} - 1 \right) \cdot R_1 \quad (5.2)$$

kde V_{CC} je vstupní napětí

V_{OUT} je výstupní napětí

R_1 je hodnota odporu R_1

R_{SENSOR} je odpor senzoru



Obrázek 5.7: Napěťový dělič

Hodnota referenčního odporu závisí na rozsahu vnitřního odporu senzorů, rozsahu a přesnosti A/D převodníku.

Vhodný referenční odpor pro tlakový senzor FSR-402 je dle produktové dokumentace od 3k - 10k Ω [13]. Zda zvolený odpor 4k7 Ω je vhodný můžeme otestovat i počtetně.

$$V_{\text{MIN}} = V_{\text{CC}} \cdot \frac{4700}{4700 + 30000} \doteq 0.135 \cdot V_{\text{CC}} \quad (5.3)$$

Rozsah vnitřního odporu senzorů	400 - 30k Ω
Rozlišení A/D převodníku	10-bitů

$$V_{\text{MAX}} = V_{\text{CC}} \cdot \frac{4700}{4700 + 400} \doteq 0.92 \cdot V_{\text{CC}} \quad (5.4)$$

$$\left\lfloor (0.92 - 0.135) \cdot 2^{10} \right\rfloor = 803 \quad (5.5)$$

kde V_{CC} je vstupní napětí

V_{MIN} je minimální naměřené napětí

V_{MAX} je maximální naměřené napětí

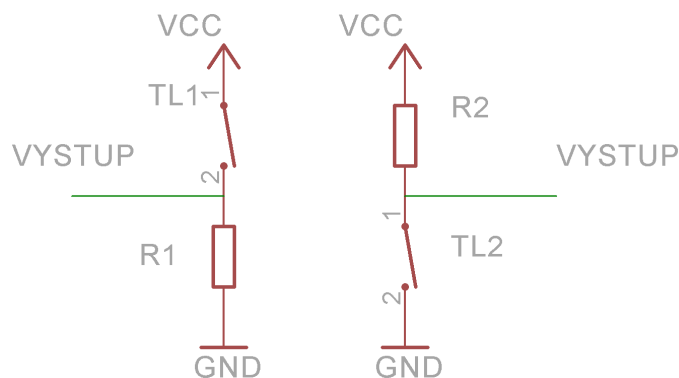
D je počet diskretních stavů

Z výpočtu je zřejmé, že při použití odporu 4k7 Ω můžeme zvolený rozsah senzorů ohodnotit 803 diskretními stavy.

Pokud by rozlišovací schopnost nebyla dostatečná, je nutné signál předem zesílit či posunout.

5.3.2 Tlačítko

Pro připojení tlačítka je nutné použít rezistor, který zamezí tzv. *floating* stavu. Rezistor udržuje klidovou hodnotu tlačítka v logické 0 při *Pull-Down* nebo 1 při *Pull-Up* (viz. obrázek 5.8). Hodnota rezistoru je v rozmezí 10 - 100 k Ω .



Obrázek 5.8: Zapojení tlačítka. Levé *Pull-Down*, Pravé *Pull-Up*

Toto schéma neřeší *glitch* neboli zákmit logické hodnoty při změně stavu. Zákmit může způsobit vícenásobné přerušení nebo volání funkce. Přidáním kondenzátoru paralelně k tlačítku je možné zákmit eliminovat na HW úrovni. Na SW úrovni je možné zákmit eliminovat pomocí filtrů.

5.3.3 LED Dioda

Pro zapojení LED diody je nutné připojit do série odpor, který omezuje množství protékaného proudu. Velikost odporu je možné vypočítat následujícím vztahem:

$$R = \frac{V_{CC} - V_{LED}}{I_{LED}} \quad (5.6)$$

kde V_{CC} je vstupní napětí

R je výsledný odpor

V_{LED} je provozní napětí LED diody

I_{LED} je proud LED diodou při provozním napětím

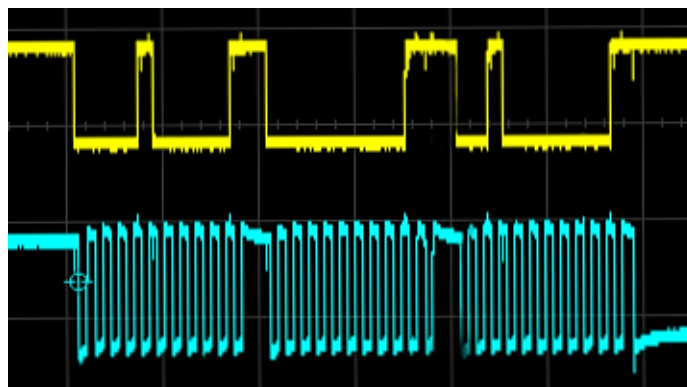
5.3.4 Fotorezistor

Zapojení fotorezistoru je identické jako zapojení tlakovému senzoru. S rostoucím množstvím dopadajícího světla se snižuje vnitřní odpor.

5.3.5 Touchpad

Zvolený touchpad disponuje sběrnici PS2, která ke komunikaci využívá 2 vodiče, data a hodiny. Raspberry Pi ale tímto řadičem nedisponuje, proto je nutné pomocí GPIO pinů napsat softwarový řadič nebo použít externí převodník.

Pro usnadnění práce je použit hotový převodník mezi PS2 a USB sběrnici.



Obrázek 5.9: Komunikace s touchpadem. Žlutá - Data, Modrá - Hodinový signál

5.3.6 Vzdálené připojení

Přípravek si po spuštění vytvoří vlastní Wi-Fi síť, která zajistí jednoduché a lehce nastavitelné spojení s počítačem studenta či vyučujícího. Další možnost je využití ethernetového připojení.

5.3.7 Napájení

Raspberry Pi B+ má relativně nízkou spotřebu energie. Pro napájení postačí adaptér s micro USB konektorem, který je schopný dodat 1.8A při 5V[19].

5.4 Výpočet polohy

Pro výpočet polohy je měření rozděleno na levou a pravou skupinu (viz. obrázek 5.10). V obou částech jsou změřené hodnoty ze senzorů. Tlakové senzory mají v námi používaném rozmezí charakteristiku podobnou s lineární funkcí. Následně je spočten poměr působícího tlaku na senzory umístěné horizontálně a vertikálně vůči sobě v obou skupinách. Z poměrů nám vznikne bod vůči třem senzorům.

Po získání bodů z obou skupin je vypočtena poloha na základě poměru působících sil mezi skupinami.

$$P_{1X} = \frac{S_3}{S_3 + S_4} \quad (5.7)$$

$$P_{1Y} = \frac{S_3}{S_1 + S_3} \quad (5.8)$$

$$P_{2X} = \frac{S_1}{S_1 + S_2} \quad (5.9)$$

$$P_{2Y} = \frac{S_4}{S_2 + S_4} \quad (5.10)$$

$$P_{3X} = \frac{S_3}{S_2 + S_3} \cdot P_{1X} + \frac{S_2}{S_2 + S_3} \cdot P_{2X} \quad (5.11)$$

$$P_{3Y} = \frac{S_3}{S_2 + S_3} \cdot P_{1Y} + \frac{S_2}{S_2 + S_3} \cdot P_{2Y} \quad (5.12)$$

kde S_1, S_2, S_3, S_4 jsou tlakové senzory

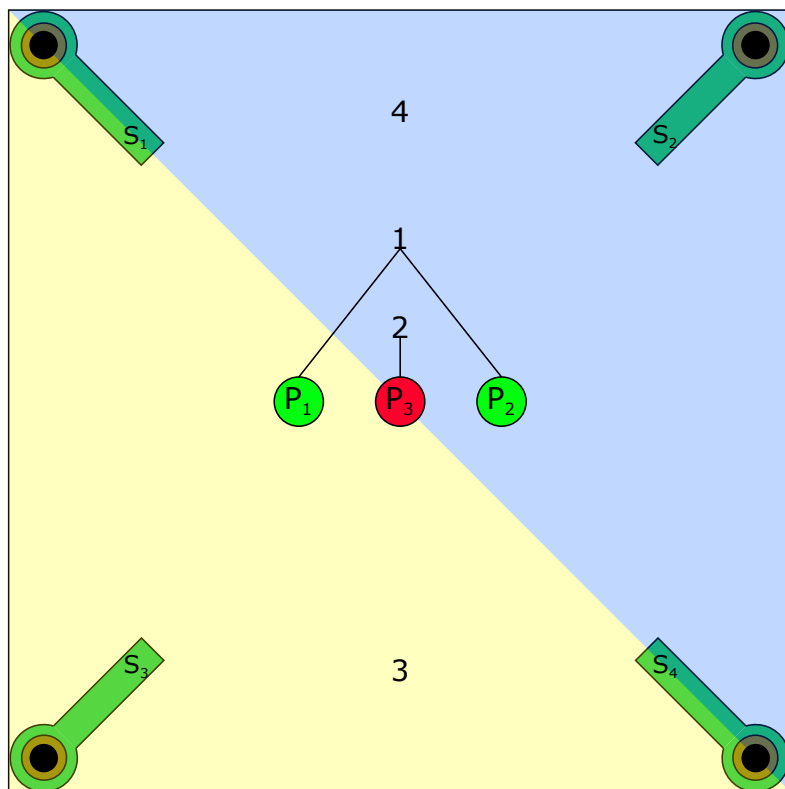
P_{1X}, P_{1Y} jsou X,Y vypočteného bodu pro levou skupinu

P_{2X}, P_{2Y} jsou X,Y vypočteného bodu pro pravou skupinu

P_{3X}, P_{3Y} jsou X,Y vypočteného bodu dotyku

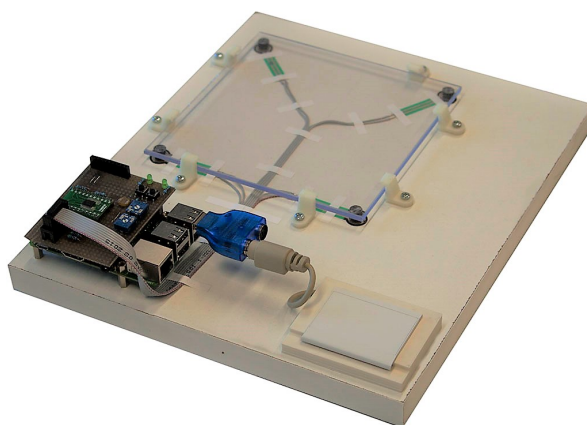
5. NÁVRH

Výsledná pozice P_3 je v rozmezí 0 - 1, kde 1 odpovídá šířce desky.



Obrázek 5.10: Zpracování polohy. 1 - Změřené body dotyku, 2 - Bod dotyku, 3 - Levá skupina, 4 - Pravá skupina

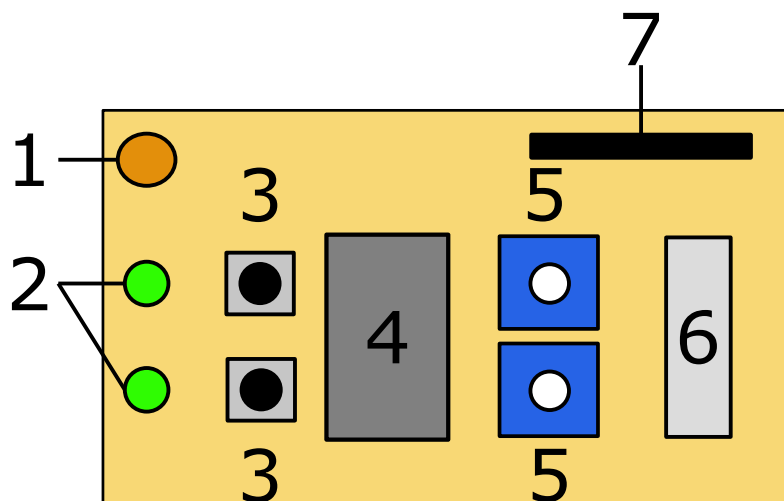
Realizace



Obrázek 6.1: Fotografie Přípravku

6.1 Rozšiřující modul

Rozšiřující modul obsahuje všechny potřebné komponenty pro připojení tlakových senzorů z dotykové plochy. Dále obsahuje senzory a aktory, které byli zmíněny v návrhu (viz. obrázek 6.2). Byl vyroben pomocí univerzálního plošného spoje. Použitý A/D převodník nelze napájet přímo na univerzální plošný spoj protože není v DIP pouzdře, ale v TSSOP. Problém byl vyřešen pomocí koupené převodní patice z TSSOP do DIP.



Obrázek 6.2: Rozšiřující modul. 1 - Foto Rezistor, 2 - LED diody, 3 - Tlačítka, 4 - A/D převodník, 5 - Potenciometry, 6 - Konektor pro tlakové senzory, 7 - Rozšiřující konektor

Rozšiřující modul je připojen pomocí 40pinové patice na Raspberry Pi (viz. tabulka B.1).

Funkce	Název pinu	Číslo pinu
I ² C SDA	SDA1	3
I ² C SCL	SCL1	5
LED 1	GPIO12	32
LED 2	GPIO20	38
Tlačítko 1	GPIO13	33
Tlačítko 2	GPIO19	35

Tabulka 6.1: Použité piny

6.2 AD7997

A/D převodníku AD7997 podporuje průběžné čtení vstupů v předem nastavené sekvenci nebo na vyžádání pomocí I²C příkazu. Přípravek používá převod na vyžádání, při kterém se dotazuje A/D převodníku každých 200 ms na hodnoty vstupů.

Adresní registr je rozdělen na dvě části, kde spodní 4 bity určují nastavený registr a horní 4 bity zahajují A/D převod.

Postup pro převod a čtení vstupů je následující:

- **1.** Zahájení převodu pomocí zápisu hodnoty $80_{\text{HEX}} + (\text{Číslo kanálu} \ll 4)$ do adresního registru A/D převodníku.

- **2.** Vyčíst data z *CONVERSION RESULT REGISTER*. Registr je 16bitový proto je nutné přečíst 2 bajty. Vrchní část registru je poslána jako první.

V nejvyšším bitu se nachází **Alert Flag**, který znázorňuje překročení nastavených limitů. Další 3 bity určují zdrojový kanál. Zbýlých 12 bitů určuje změřenou hodnotu kanálu.

$$V_{in_x} = \frac{((D_{\text{Horní}} \& F_{\text{HEX}}) \ll 8) \mid D_{\text{Dolní}}}{4096} \cdot V_{cc} \quad (6.1)$$

kde V_{cc} je vstupní napětí

V_{in_x} je napětí na kanálu x

$D_{\text{Horní}}$ je horních 8 bitů přečteného registru

$D_{\text{Dolní}}$ je dolních 8 bitů přečteného registru

\mid je bitový součet

$\&$ je bitový součin

\ll je bitový posun do leva

6.3 Ovladač

Pro jednoduchou komunikaci se senzory byl implementován modul linuxového jádra, který se stará o čtení ze senzorů a přístup k aktorům. Data z A/D převodníku jsou obnovována každých 200 ms a jsou přístupná skrze I/O operace z libovolného jazyka či nástroje. Ovladač umožňuje přístup k zpracovaným i nezpracovaným datům (viz. tabulka 6.2).

Cesta vůči /sys/class/	Význam	Zápis	Čtení
mi-pvs/adc/ad2	Tlakový senzor 1		0-1023
mi-pvs/adc/ad5	Tlakový senzor 2		0-1023
mi-pvs/adc/ad3	Tlakový senzor 3		0-1023
mi-pvs/adc/ad4	Tlakový senzor 4		0-1023
mi-pvs/adc/ad0	Potenciometr 1		0-1023
mi-pvs/adc/ad1	Potenciometr 2		0-1023
mi-pvs/adc/ad6	Foto Senzor		0-1023
mi-pvs/touchpad/pressed	Detekce dotyku		0,1
mi-pvs/touchpad/position	Pozice dotyku		0-500 0-500
mi-pvs/touchpad/calibration	Kalibrační vstup	1,2	
mi-pvs/io/led0	LED 1	0,1	0,1
mi-pvs/io/led1	LED 2	0,1	0,1
mi-pvs/io/bt0	Tlačítko 1	0,1	0,1
mi-pvs/io/bt1	Tlačítko 2	0,1	0,1

Tabulka 6.2: I/O přístup k ovladači

Kalibrace dotykové plochy probíhá přes zápis do *calibration* souboru. Hodnoty 1 a 2 určují kroky kalibrace. Více o kalibraci v kapitole 7.2.

6.3.1 Aritmetické operace s desetinným číslem

V kernelu by neměly být používány matematické operace s pohyblivou desetinnou čárkou [20]. Proto bylo nutné pro výpočty použít pevnou řádovou čárku vytvořenou z datového typu *uint32_t* a bitového posunu.

$$\frac{(R \ll P) \cdot A}{A + B} \gg P \quad (6.2)$$

kde A, B jsou poměřované proměnné

R je rozsah pro poměr

P je velikost bitového posunu

\gg je bitový posun do prava

\ll je bitový posun do leva

Tento výraz vypočítá poměr v předem zvoleném rozsahu z proměnných A a B při přesnosti 2^{-P} . Hodnota P musí být volena rozumně, aby při výpočtu nedošlo k přetečení proměnné, tím ke špatnému výsledku. Toto je možné částečně vyřešit použitím datového typu s větším rozsahem např. *uint64_t*.

6.4 Java MQTT Klient

Pro přístup k datům ze senzorů byl naprogramovaný jednoduchý MQTT klient v jazyce Java. Klient je postaven na knihovně MQTT Client od FuseSource. Nastavuje se pomocí XML konfiguračního souboru, kde je možné nastavit IO operace a jejich návaznost na MQTT příspěvky.

Klient umožňuje tři typy úkolů. Periodické odesílání obsahu souboru s předem zvoleným předmětem. Průběžné čtení ze souboru po řádcích, kde každá řádka se odesílá jako samostatná zpráva s předem zvoleným předmětem. Naslouchání zpráv na zvolené předměty. Tyto zprávy jsou ukládány do předem zvolených souborů. Pomocí těchto funkcí je možné nejen číst data ze senzorů, ale i posílat hodnoty aktorům pomocí MQTT protokolu.

Pro odesílání dat ze senzorů je nutné nastavit "přispívání" souborů z adresáře `/sys/class/mi-pvs`. Návod na použití a popsání konfiguračního souboru je možné najít v příloze.

Listing 6.1: Ukázka konfiguračního souboru

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <configuration>
3   <charset>UTF8</charset>
4   <port>1883</port>
5   <address>localhost</address>
6   <username>pi</username>
7   <password></password>
8   <publish>
9     <interval>500</interval>
10    <path>/sys/class/mi-pvs/ad2</path>
11    <topic>/sensors/presure/1</topic>
12    <name>Tlakovy senzor 1</name>
13  </publish>
14  ... //Konfiguracni soubor byl zkracen , cely soubor je
      mozne najit v prilozce
15  <subscribe>
16    <path>/sys/class/mi-pvs/io/led1</path>
17    <topic>/actor/led/2</topic>
18    <name>Led 2</name>
19  </subscribe>
20 </configuration>

```

6.5 Simulace kurzoru

Dodaný touchpad vrací data v podobě relativních pohybů. Pro zpracování dat byl napsán jednoduchý program, který je nezávislý na grafickém prostředí. Vytváří virtuální plochu umožňující převod relativních pohybů na absolutní souřadnice a stav tlačítek. Pro změnu velikosti plochy je nutné změnit zdrojový soubor.

Hodnoty jsou posílány na standardní výstup programu. Pro napojení na Java MQTT klient a prostředí NodeRed je nutné výstup přesměrovat do "pojmenované roury", která se chová jako soubor ze kterého jmenované programy mohou číst.

Program čte data ze systémového souboru `/dev/input/mice` po 3 bajtech. První dva bajty určují posun v osách X a Y. Třetí bajt určuje stisklá tlačítka.

6.6 Mosquitto

Image Raspberry Pi byl rozšířen o Open Source software Mosquitto. Jedná se o nenáročný plnohodnotný MQTT Broker pro linux. Broker byl nainstalován z oficiálních katalogů Raspbianu pomocí balíčkovacího nástroje **APT** příkazem **apt-get install mosquitto**. Balíčkovací nástroj se postaral o vytvoření potřebného uživatele pro démona a nastavil automatické spouštění při startu systému.

Broker byl ponechán v základním nastavení, kde nevyžaduje žádné přihlášení. Při normálním nasazení je nutné z bezpečnostních důvodů nastavit uživatelské účty a k nim příslušná oprávnění. Mosquitto podporuje externí moduly pro přihlášení, tím je možné doprogramovat adaptér pro libovolný ověřovací mechanismus uživatelů. Dále je nutné nastavit šifrované spojení pomocí TLS vrstvy.

Data ze senzorů jsou posílány do MQTT Brokeru pomocí Java MQTT Klient a NodeRed.

6.7 Webové skripty

Další možnost přístupu k datům je přes předpřipravené skripty. Ty umožňují jednoduchý přístup k datům bez nutnosti instalovat na uživatelský počítač MQTT klienta. Skripty je dále možné využít pro čtení dat z vlastního programu, protože data jsou přístupná ve formátu JSON. Pro správnou funkčnost skriptů je nutné mít zaveden ovladač pro rozšířený modul.

Skripty jsou napsány v jazyku PHP a JavaScript. Jako webový server byl použit Apache. Webový server i PHP prostředí byly nainstalovány ze standardního katalogu Raspbianu pomocí nástroje **APT**. Webový server byl nastaven na naslouchání na portu 80 a soubory pro webové stránky očekává v adresáři `/var/www`.

6.8 NodeRed

Image Raspberry Pi byl rozšířen o skriptovací prostředí NodeRed, které umožňuje jednoduché napojení na souborový systém i MQTT Broker. Skriptování probíhá pomocí propojování funkčních bloků. Je možné použít vlastní funkční

Adresa	Význam
http://192.168.3.1/index.html	Aktualizovaný přehled hodnot
http://192.168.3.1/touchpad.html	Grafické znázornění zdroje síly na dotykovou plochu
http://192.168.3.1/data.php	Data ze senzorů ve formátu JSON
http://192.168.3.1/dataTouchpad.php	Polohová data z dotykové plochy ve formátu JSON

Tabulka 6.3: Přehled webových skriptů

bloky napsané v jazyce JavaScript. Přístup k prostředí je možný pomocí webového prohlížeče na adrese <http://192.168.3.1:1880>.

Pro čtení dat ze senzorů je nutné použít I/O operace na soubory ve složce `/sys/class/mi-pvs/`. Data mohou být následně zpracována a odeslána do MQTT Brokeru nebo zobrazena na webové stránce.

NodeRed bylo staženo z webu <http://nodered.org>. Pokyny pro spuštění je možné najít v příloze.

6.9 Přímý přístup k datům

Pro přímý přístup je nutné nejprve ovladač vyjmout z jádra operačního systému pomocí příkazu `rmmmod`. Toto uvolní alokované zdroje a prostředky používané ovladačem.

Další možnost je uvolnění pouze A/D převodníku přes zápis adresy A/D převodníku do souboru `/sys/bus/i2c/devices/i2c-1/delete_device`

6.9.1 i2c-tools

Jedná se o skupinu nástrojů, která umožňuje řízení sběrnice I²C z příkazové řádky.

- **i2cdetect** - Detekuje zařízení na I²C sběrnici.

```
pi@raspberrypi ~ $ i2cdetect -y 1
0 1 2 3 4 5 6 7 8 9 a b c d e f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- 21 -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

70: -- -- -- -- -- -- --

Zde je možné vidět, že nástroj našel A/D převodník na adrese 0x21. Pokud je zařízení přiřazeno ovladači na místě adresy je *UU*.

- **i2cdump** - Prozkoumá registry I²C zařízení. Tento program je nutné pouštět s rozvahou, může zmást zařízení, kde zápis do adresního registru vyvolá akci. Zápis do horních 4 bitů adresy u AD7997 nastartuje převod zvoleného kanálu.
- **i2cset** - Nastaví registr v zařízení.
- **i2cget** - Přečte registr v zařízení.

6.9.2 Přístup k GPIO ovladači přes I/O

GPIO ovladač umožňuje ovládat GPIO piny skrz virtuální souborový systém. Pomocí souboru `/sys/class/gpio/export` je nutné exportovat vybraný pin, aby ho bylo možné ovládat. Exportovaný pin vytvoří virtuální adresář, který obsahuje soubory potřebné k ovládání pinu. Pomocí zápisu do souboru **direction** je možné nastavit vstupní, či výstupní režim pinu. Hodnotu pinu je možné číst, či nastavit přes soubor **value**.

6.9.3 Přístup k I²C ovladači přes I/O operace

K I²C ovladači je možno přistupovat z programovacích jazyků, které umožňují I/O operace a dokáží pracovat s funkcí **ioctl**. Vstupní bod ovladače je v souboru `/dev/i2c-(číslo sběrnice)`.

6.9.4 Knihovna Pi4J

Jedná se o knihovnu umožňující programovacímu jazyku Java nízko úrovněvý přístup k perifériím. Programy využívající tuto knihovnu musí být spuštěny pod ROOT účtem.

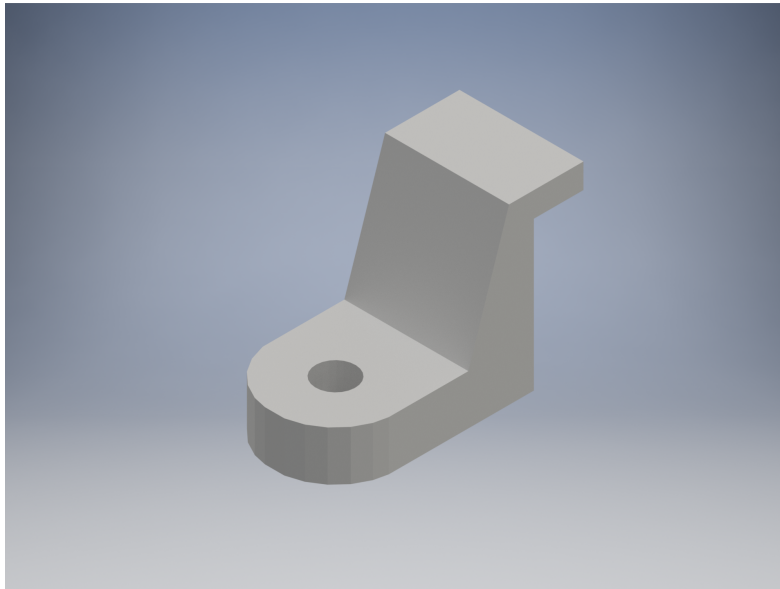
6.10 Konstrukce

6.10.1 Dotyková plocha

Dotyková plocha byla vyrobena z průhledné plexisklové desky o rozměrech 15 × 15 cm.

Pro zajištění desky byl navrhnut úchyt o rozměrech 10 × 13 × 20 mm (viz. obrázek 6.3). Na každé hraně jsou dva úchyty rozmístěny 5 cm od krajů. Aby nedocházelo k dotyku mezi deskou a bočními úchyty, což by znehodnotilo výsledky měření, byla mezi úchyty a plochou nechána mezera o 0.5 mm.

Úchyty byly navrženy v programu Autodesk Inventor 2016 a vytisknuty na 3D tiskárně.



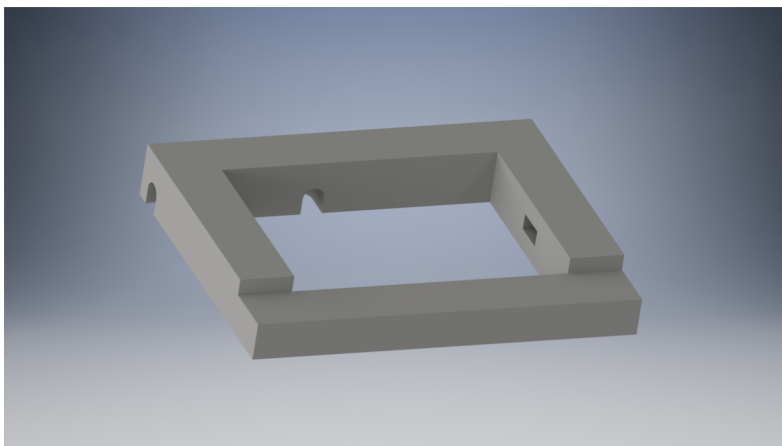
Obrázek 6.3: Render úchytu dotekové plochy

Do každého rohu desky byly nalepeny gumové distanční podložky pro ideální přenos tlaku na tlakové senzory. Senzory byly uchyceny pomocí lepicí pásky k dřevěné podkladové desce.

6.10.2 Touchpad

Touchpad je zasazen do rámečku (viz. obrázek 6.4) a přichycen pomocí lepidla. Kabel je vyveden předem připravenou drážkou v horní části rámečku. Model rámečku byl navrhnout v programu Autodesk Inventor 2016 a následně vytisknut na 3D tiskárně.

Šrouby pro uchycení jsou vedeny skrz podkladovou desku. Tím je docíleno estetičtějšího vzhledu.



Obrázek 6.4: Render rámečku pro touchpad

Pro připojení touchpadu k Raspberry Pi byl zakoupen převodník mezi sběrnicemi (viz. obrázek 6.5).



Obrázek 6.5: Převodník sběrnic pro touchpad

Pro zjednodušení čtení dat touchpadu byl vytvořen jednoduchý program. Program čte údaje o relativních pohybech na touchpadu a simuluje pohyb kurzoru. Na standardní výstup se vypisují X,Y souřadnice a stav tlačítek.

Testování

7.1 Rozšiřující modul

7.1.1 AD7997

Pomocí trimru použitého jako napěťový dělič mezi 3.3 V a zemí byl otestován celý rozsah A/D převodníku. Data byla vyčítána pomocí ovladače pro rozšířený modul i pomocí nástroje *i2c-tools*.

7.1.2 Fotorezistor

Fotorezistor byl otestován v rozdílných světelných podmínkách od úplné tmy po přímé osvětlení halogenovou lampou. Naměřené hodnoty byli v rozsahu 56 až 1011. To pokrývá 93% rozsahu A/D převodníku.

7.1.3 Potenciometr

Při nastavování obou potenciometrů byli naměřeny hodnoty z celého rozsahu A/D převodníku. Změna naměřené hodnoty na potenciometru byla přímo úměrná změně nastavení potenciometrů.

7.2 Dotyková plocha

Po otestování byly zjištěny nesrovnalosti v charakteristikách tlakových senzorů. Tento problém je možné vyřešit kalibrační procedurou před používání přípravku.

- **1.** Změřit působení sil na senzory v klidovém stavu. Nastavit offset pro každý senzor, aby v klidovém stavu dával nulové hodnoty.
- **2.** Změřit působení sil na senzory při zátěži minimálně 400g působící na střed desky. Nastavit převodní funkci pro každý senzor, aby všechny senzory dávaly stejnou hodnotu.

7. TESTOVÁNÍ

Kalibrace je implementovaná v ovladači.

7.3 Touchpad

Pro správnou funkci převodníku nejsou nutné externí ovladače. Po připojení touchpadu s redukcí k Raspberry Pi bylo zařízení rozpoznáno. Čtení dat z touchpadu pomocí vytvořeného programu pro simulaci kurzoru fungovalo úspěšně.

7.4 Ovladač

Po zavedení ovladače do linuxového jádra byla automaticky vytvořena souborová struktura ve virtuálním souborovém systému. Pomocí programu *cat* bylo možné přečíst hodnoty senzorů i pozici doteku na dotykové ploše po provedení kalibrační procedury. Dále bylo otestováno ovládání LED diod zápisem do souboru.

7.5 Programová výbava

7.5.1 Mosquitto

MQTT Broker Mosquitto byl ponechán v základním nastavením, kde poslouchá na TCP portu 1883. V tomto nastavení se může přihlásit libovolný klient.

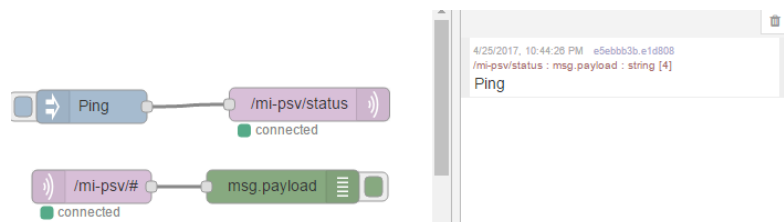
Broker byl otestován pomocí programů:

- **mosquitto_sub** - Slouží pro odebrání určitého předmětu.
- **mosquitto_pub** - Slouží pro publikování příspěvku.

Na Raspberry Pi bylo zapnuto odebrání příspěvků a pomocí externího PC byl odeslán příspěvek, který Raspberry Pi přijal.

7.5.2 NodeRed

Po nainstalování byla otestována základní funkčnost I/O bloků. Dále byla potřeba otestovat napojení na MQTT Broker.



Obrázek 7.1: Testování NodeRed

Na ukázce (viz. obrázek 7.1) je vidět úspěšné publikování i přijetí příspěvku.

7.5.3 MQTT Klient

MQTT klient byl nastaven na periodické čtení a následnou publikaci dat z potenciometrů. Zároveň byl nastaven pro poslech a zápis do souboru na předměty použité při publikaci. Data byla úspěšně zapsaná do určených souborů.

Závěr

Cílem zadání této bakalářské práce bylo vytvoření výukového přípravku s tlakovými senzory podporující IoT na předmět MI-PVS vyučovaném na FIT ČVUT Praha.

Na počátku práce byla provedena rešerše IoT protokolů a otestování tlakových senzorů pro měření tlaku na dotykovou plochu. Dále byl vyvinut rozšiřující modul pro Raspberry Pi, který umožňoval čtení analogových dat ze senzorů. Funkčnost rozšiřujícího modulu byla otestována.

Byla navržena konstrukce přípravku, pro kterou bylo nutné vymodelování a následné vytisknutí dílů na 3D tiskárně.

K interakci se senzory byl naprogramován ovladač ve formě modulu do jádra linuxového systému. Ovladač umožňuje přístup k datům přes I/O operace a výpočet polohy působené síly na dotykovou plochu. Výukový přípravek byl rozšířen o touchpad z přenosného počítače.

Pro demonstraci IoT byl naprogramován MQTT klient v jazyce Java, který umožňuje napojení MQTT protokol na souborový systém. Komunikace s přípravkem využívá WiFi nebo Ethernetové spojení.

V poslední řadě byl systém rozšířen o skripty a skriptovací prostředí NodeRed, které umožňují rychlý a jednoduchý přístup k datům. Přípravek byl pak jako celek otestován a jeho funkčnost byla potvrzena.

Literatura

- [1] Telecommunication Standardization Sector (ITU-T): *Internet of Things Global Standards Initiative*. [cit. 2017-03-26]. Dostupné z: <http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx>
- [2] Internet Engineering Task Force: *Hypertext Transfer Protocol Version 2 (HTTP/2)*. [cit. 2017-03-26]. Dostupné z: <https://tools.ietf.org/html/rfc7540>
- [3] O'Connell, B.; Stanford-Clark, A.: Using MQ Telemetry Transport with WebSphere Business Integration Message Broker. *IBM WebSphere [online]*, Srpen 2005, [cit. 2017-03-26]. Dostupné z: https://www.ibm.com/developerworks/websphere/library/techarticles/0508_oconnell/0508_oconnell.html
- [4] Jaffey, T.: MQTT and CoAP, IoT Protocols. *Eclipse Newsletter [online]*, Unor 2014, [cit. 2017-03-26]. Dostupné z: https://eclipse.org/community/eclipse_newsletter/2014/february/article2.php
- [5] Poole, I.: Wibree / Bluetooth Low Energy. *Radio-Electronics [online]*, [cit. 2017-03-26]. Dostupné z: <http://www.radio-electronics.com/info/wireless/bluetooth/bluetooth-low-energy-wibree.php>
- [6] BLUETOOTH VS. BLUETOOTH LOW ENERGY: WHAT'S THE DIFFERENCE? *Link Labs Blog [online]*, Listopad 2015, [cit. 2017-03-26]. Dostupné z: <https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy>
- [7] Bluetooth: *GATT Services*. [cit. 2017-03-26]. Dostupné z: <https://www.bluetooth.com/specifications/gatt/services>
- [8] Řehák, J.: DTMF - fámy a skutečnost. *vyvoj.hw.cz [online]*, Prosinec 1998, [cit. 2017-03-26]. Dostupné z: <http://vyvoj.hw.cz/teorie-a-praxe/dokumentace/dtmf-famy-a-skutecnost.html>

- [9] K. Kuladinithi, E.: Transport of CoAP over SMS. *IETF [online]*, Unor 2017, [cit. 2017-03-26]. Dostupné z: <https://core-wg.github.io/sms/>
- [10] SIGFOX: NOVÁ BEZDRÁTOVÁ SÍŤ PRO „INTERNET VĚCÍ“ V ČESKÉ REPUBLICE. *T-PRESS [online]*, Kveten 2015, [cit. 2017-03-26]. Dostupné z: <https://www.t-press.cz/cs/tiskove-materialy/tiskove-zpravy-t-mobile/sigfox-nova-bezdratova-sit-pro-internet-veci-v-ceske-republice.html>
- [11] Simple Cell: *Simple Cell FAQ*. [cit. 2017-03-26]. Dostupné z: <https://www.simplecell.eu/faq/>
- [12] Sigfox: *ARM-P8-SIGFOX*. [cit. 2017-03-26]. Dostupné z: <https://partners.sigfox.com/products/arm-p8-sigfox>
- [13] InterLink Electronics: *FSR 402 Data Sheet*. [cit. 2017-03-26]. Dostupné z: <http://www.trossenrobotics.com/productdocs/2010-10-26-DataSheet-FSR402-Layout2.pdf>
- [14] University of California, Berkeley: *The Nyquist-Shannon Sampling Theorem*. [cit. 2017-03-26]. Dostupné z: <https://www.raspberrypi.org/trademark-rules/>
- [15] *Understanding EDID - Extended Display Identification Data*. [cit. 2017-03-26]. Dostupné z: <http://www.extron.com/company/article.aspx?id=uedid>
- [16] Olejár, M.: Stručný popis sběrnice I2C a její praktické využití k připojení externí eeprom 24LC256 k mikrokontroléru PIC16F877. *vyvoj.hw.cz [online]*, kveten 2000, [cit. 2017-03-26]. Dostupné z: <http://vyvoj.hw.cz/navrh-obvodu/strucny-popis-sbernice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-eeeprom-24lc256>
- [17] *I2C Bus Specification*. [cit. 2017-03-26]. Dostupné z: <http://i2c.info/i2c-bus-specification>
- [18] Maxim Integrated: *Programmable Resolution 1-Wire Digital Thermometer*. [cit. 2017-03-26]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [19] Raspberry Pi Foundation: *Raspberry FAQs*. [cit. 2017-03-26]. Dostupné z: <https://www.raspberrypi.org/help/faqs/>
- [20] Torvalds, L.: Re: How to use floating point in a module? *Mail Communication [online]*, Březen 2003, [cit. 2017-03-26]. Dostupné z: http://yarchive.net/comp/linux/kernel_fp.html

Seznam použitých zkratek

ACL Access control List

CVUT České vysoké učení technické

DIP Dual in-line package

DTLS Datagram Transport Layer Security

FIT Fakulta informačních technologií

GPIO General-purpose input/output

IoT Internet of Things

QoS Quality of Services

SSL Secure Sockets Layer

TLS Transport Layer Security

TCP Transmission Control Protocol

UDP User Datagram Protocol

USD United States dollar

Zapojení

B.1 Rozšiřující modul

Rozšiřující modul je připojen pomocí 40pinového konektoru Raspberry Pi.

Funkce	Název pinu	Číslo pinu
I ² C SDA	SDA1	3
I ² C SCL	SCL1	5
LED 1	GPIO12	32
LED 2	GPIO20	38
Tlačítko 1	GPIO13	33
Tlačítko 2	GPIO19	35

Tabulka B.1: Použité piny

Kromě použitých pinů v tabulce modul využívá 3v3 a GND piny.

Číslo kanálu	Připojené zařízení
0	Potenciometr 1
1	Potenciometr 2
2	Tlakový senzor 1
3	Tlakový senzor 3
4	Tlakový senzor 4
5	Tlakový senzor 2
6	Foto Senzor

Tabulka B.2: Zapojení AD7997

B.2 Touchpad

Pro připojení touchpadu je nutné využít předpřipravený PS2 USB převodník.

Kompilace

C.1 Ovladač

Pro kompilaci ovladače je vhodné použít předpřipravený **Makefile** s parametrem *build*. Parametr *clean* vyčistí složku a zanechá zdrojové soubory.

```
obj-m+=mi-psv-driver.o
```

```
all:
```

```
make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD) modules
```

```
clean:
```

```
make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD) clean
```

Pro kompilaci je nutné mít hlavičkové soubory použitého linuxového jádra.

C.2 Simulace kurzoru

Program lze zkompilevat pomocí **GCC** kompilátoru. Je přiložen Makefile pro jednodušší práci.

```
build:
```

```
gcc mouse.c -o mouse
```

C.3 MQTT Java Klient

Program je dodán ve formě exportovaného NetBeans projektu. Potřebná MQTT knihovna je dodána ve složce lib.

Použití

D.1 Síťové připojení

Uživatelské jméno - pi
Heslo - raspberry

D.1.1 WiFi

Po startu systému Raspberry Pi vytvoří bezdrátovou WiFi síť *RaspAP*. Síť není zabezpečená. DHCP server je nakonfigurován pro rozdělování adres v síti 192.168.3.0/24. Raspberry Pi má adresu 192.168.3.1.

D.1.2 Ethernet

Po připojení síťového kabelu si Raspberry Pi vyžádá adresu od DHCP serveru. Na zařízení je nastaven NAT mezi ethernetem a WiFi adaptérem, který umožňuje přístup na internet WiFi klientům.

D.2 Ovladač

Ovladač je nutné zavést ručně pomocí příkazu **insmod**. Dále je nutné přiřadit AD7997 ovladači pomocí zápisu řetězce *AD7997 0x21* do souboru */sys/class/i2c-adapter/i2c-1/new_device*.

Je přiložen Makefile soubor, který zavedení modulu zjednodušuje. Parametr *insmod* zavede modul, přiřadí AD7997 ovladači a upraví oprávnění. Module je možné vyjmout použitím parametru *rmmod*.

```
insmod:  
insmod mi-psv-driver.ko  
echo AD7997 0x21 > /sys/class/i2c-adapter/i2c-1/new_device  
chmod +rw /sys/class/mi-pvs/io/bt0
```

D. Použití

```
chmod +rw /sys/class/mi-pvs/io/bt1
chmod +rw /sys/class/mi-pvs/touchpad/calibration

rmmod:
rmmod mi-psv-driver
echo 0x21 > /sys/class/i2c-adapter/i2c-1/delete_device
```

Cesta vůči /sys/class/	Význam	Zápis	Čtení
mi-pvs/adc/ad2	Tlakový senzor 1		0-1023
mi-pvs/adc/ad5	Tlakový senzor 2		0-1023
mi-pvs/adc/ad3	Tlakový senzor 3		0-1023
mi-pvs/adc/ad4	Tlakový senzor 4		0-1023
mi-pvs/adc/ad0	Potenciometr 1		0-1023
mi-pvs/adc/ad1	Potenciometr 2		0-1023
mi-pvs/adc/ad6	Foto Senzor		0-1023
mi-pvs/touchpad/pressed	Detekce dotyku		0,1
mi-pvs/touchpad/position	Pozice dotyku		0-500 0-500
mi-pvs/touchpad/calibration	Kalibrační vstup	1,2	
mi-pvs/io/led0	LED 1	0,1	0,1
mi-pvs/io/led1	LED 2	0,1	0,1
mi-pvs/io/bt0	Tlačítko 1	0,1	0,1
mi-pvs/io/bt1	Tlačítko 2	0,1	0,1

Tabulka D.1: I/O přístup k ovladači

D.2.1 Kalibrace dotekové plochy

Kalibrace probíhá přes `/sys/class/mi-pvs/touchpad/calibration` soubor.

- **1. Krok** - Odstranit veškeré předměty z dotekové plochy. Zkontrolovat zda se plocha nedotýká úchytů. Zapsat hodnotu **1**.
- **2. Krok** - Aplikovat sílu minimálně 3 N do středu dotekové plochy. Zapsat hodnotu **2**.

Bez provedení kalibrační procedury ovladač neumožňuje čtení dat z dotekové plochy.

D.3 Webové skripty

Pro funkčnost skriptů je nutné zavedení ovladače.

Adresa	Význam
http://192.168.3.1/index.html	Aktualizovaný přehled hodnot
http://192.168.3.1/touchpad.html	Grafické znázornění zdroje síly na dotykovou plochu
http://192.168.3.1/data.php	Data ze senzorů ve formátu JSON
http://192.168.3.1/dataTouchpad.php	Polohová data z dotykové plochy ve formátu JSON

Tabulka D.2: Přehled webových skriptů

D.4 Program pro simulaci kurzoru

Každý řádek standardního výstupu programu obsahuje pět čísel oddělených mezerou.

```
140 423 0 1 0
(X pozice kurzoru)(Y pozice kurzoru)
(Levé tlačítko) (Prostřední tlačítko) (Pravé tlačítko)
```

Při každé změně polohy je vygenerován nový řádek.

Pro napojení na Java MQTT Klient nebo na NodeRed je nutné přesměrovat standardní výstup do "pojmenované roury", ze které jde pak už lépe číst. Rouru je možné vytvořit příkazem **mkfifo**.

Ukázka přesměrování soubory do "pojmenované roury"s jejím vytvořením.

```
mkfifo vystup
./mouse > vystup
```

D.5 NodeRed

Prostředí NodeRed je nutné spustit ručně pomocí příkazu **node-red-pi**. Po startu je možné prostředí otevřít pomocí libovolného webového prohlížeče na adrese <http://192.168.3.1:1880>. Dokumentace prostředí je na adrese <https://nodered.org/docs/>.

D.6 MQTT Java Klient

Pro spuštění programu je nutné definovat cestu ke konfiguračnímu souboru pomocí argumentu.

Konfigurační soubor se skládá z následujících částí:

XML Konfigurační soubor

charset	Použité kódování
address	Adresa MQTT Brokeru
username	Uživatelské jméno pro přihlášení k MQTT Brokeru
password	Heslo pro přihlášení k MQTT Brokeru
publish	Konfigurace pro publikování příspěvku ze souboru
interval	Interval čtení souboru v ms, 0 pro řádkové čtení
path	Cesta k souboru
topic	Předmět příspěvku
name	Identifikace použitá při ladících hláškách
subscribe	Konfigurace pro ukládání příspěvku do souboru
path	Cesta k souboru
topic	Předmět příspěvku
name	Identifikace použitá při ladících hláškách

Listing D.1: Použitý konfiguračního souboru

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <configuration>
3   <charset>UTF8</charset>
4   <port>1883</port>
5   <address>localhost</address>
6   <username>pi</username>
7   <password></password>
8   <publish>
9     <interval>500</interval>
10    <path>/sys/class/mi-pvs/ad2</path>
11    <topic>/sensors/presure/1</topic>
12    <name>Tlakovy senzor 1</name>
13  </publish>
14  <publish>
15    <interval>500</interval>
16    <path>/sys/class/mi-pvs/ad5</path>
17    <topic>/sensors/presure/2</topic>
18    <name>Tlakovy senzor 2</name>
19  </publish>
20  <publish>
21    <interval>500</interval>
22    <path>/sys/class/mi-pvs/ad3</path>
23    <topic>/sensors/presure/3</topic>
24    <name>Tlakovy senzor 3</name>
25  </publish>
26  <publish>
27    <interval>500</interval>
28    <path>/sys/class/mi-pvs/ad4</path>
29    <topic>/sensors/presure/4</topic>
30    <name>Tlakovy senzor 4</name>
31  </publish>
32  <publish>
33    <interval>500</interval>

```

```

34     <path>/sys/class/mi-pvs/adc/ad0</path>
35     <topic>/sensors/pot/1</topic>
36     <name>Potenciometr 1</name>
37 </publish>
38 <publish>
39     <interval>500</interval>
40     <path>/sys/class/mi-pvs/adc/ad1</path>
41     <topic>/sensors/pot/2</topic>
42     <name>Potenciometr 2</name>
43 </publish>
44 <publish>
45     <interval>500</interval>
46     <path>/sys/class/mi-pvs/adc/ad6</path>
47     <topic>/sensors/light</topic>
48     <name>Fotorezistor</name>
49 </publish>
50 <publish>
51     <interval>500</interval>
52     <path>/sys/class/mi-pvs/io/bt0</path>
53     <topic>/sensors/button/1</topic>
54     <name>Tlacitko 1</name>
55 </publish>
56 <publish>
57     <interval>500</interval>
58     <path>/sys/class/mi-pvs/io/bt1</path>
59     <topic>/sensors/button/2</topic>
60     <name>Tlacitko 2</name>
61 </publish>
62 <subscribe>
63     <path>/sys/class/mi-pvs/io/led0</path>
64     <topic>/actor/led/1</topic>
65     <name>Led 1</name>
66 </subscribe>
67 <subscribe>
68     <path>/sys/class/mi-pvs/io/led1</path>
69     <topic>/actor/led/2</topic>
70     <name>Led 2</name>
71 </subscribe>
72 </configuration>

```


Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
impl	
mqttClient.....	adresář se spustitelnou formou Java MQTT Klientu
module.....	adresář s Ovladačem rozšiřujícího modulu
mouse.....	adresář se spustitelnou formou Simulace kurzoru
imageRPi.....	soubor se zálohou image pro Raspberry Pi
src	
models.....	adresář s 3D modely
impl	
mqttClient..	adresář se zdrojovými kódy pro Java MQTT Klientu
module.....	adresář se zdrojovými kódy pro Ovladač rozšiřujícího modulu
mouse.....	adresář se zdrojovými kódy pro Simulaci kurzoru
www.....	adresář se zdrojovými kódy pro Webové skripty
thesis	
thesis.....	zdrojová forma práce ve formátu L ^A T _E X
images.....	použité obrázky pro práci
data.....	doplňkové soubory pro práci
text.....	text práce
thesis.pdf.....	text práce ve formátu PDF