

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

VIZUALIZACE INFORMACÍ V INTELIGENTNÍ DOMÁCNOSTI

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN SAKIN

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

VIZUALIZACE INFORMACÍ V INTELIGENTNÍ DOMÁCNOSTI

VISUALIZATION OF INFORMATION IN INTELLIGENT HOME

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN SAKIN

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PAVOL KORČEK

BRNO 2015

Abstrakt

Tato práce popisuje návrh a implementaci grafické aplikace zobrazující stav inteligentní domácnosti. Aplikace běží přímo na adaptéru, který komunikuje se senzory a akčními členy. Její grafický výstup je možné zobrazit přes konektor HDMI nebo vzdáleně přes X-Server. Aplikace je implementována v jazyce C++ a využitím grafických knihoven Qt. Cílem je jednoduše a přehledně zobrazit důležité informace o domácnosti i v případě výpadku LAN sítě.

Abstract

This thesis describes the design and implementation of a graphical application that shows the status of intelligent home. Application run directly on the adapter that communicates with sensors and actuators. Its graphical output can be displayed via the HDMI connector or remotely over X-Server. The application is implemented in C++ and using the Qt graphics libraries. The goal is simply and clearly to display important information about the household even in case of failure of the LAN.

Klíčová slova

inteligentní domácnost, grafické rozhraní, internet věcí, Qt

Keywords

intelligent home, graphic user interface, internet of things, Qt

Citace

Martin Sakin: Vizualizace informací v inteligentní domácnosti, bakalářská práce, Brno, FIT VUT v Brně, 2015

Vizualizace informací v inteligentní domácnosti

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Pavola Korčeka a Ing. Tomáše Novotného.

.....
Martin Sakin
18. května 2015

Poděkování

Děkuji vedoucímu práce Ing. Pavolu Korčekovi a Ing. Tomáši Novotnému za odbornou pomoc a čas, který mi věnovali.

© Martin Sakin, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	2
2	Inteligentní domácnost	3
2.1	Možnosti ovládání	3
2.2	Popis datových zdrojů	5
2.3	Existující řešení	5
2.4	Systém vyvíjený na FIT	7
3	Návrh	9
3.1	Meziprocesové komunikace	9
3.2	Grafické zobrazení	11
3.2.1	Možnosti grafického výstupu	12
3.2.2	Návrh obrazovek	13
3.3	Úspora energie	15
3.4	Komunikace s monitorem	15
3.5	Cílové zařízení	16
3.6	Knihovna Qt	17
4	Implementace	19
4.1	Popis grafické části	19
4.2	Popis objektového návrhu	21
4.3	Software na adaptéru	23
5	Testování	25
5.1	Uživatelské	25
5.2	Dlouhodobý běh	28
5.3	Zátěž na CPU	28
6	Závěr	30
6.1	Návrhy na rozšíření	30
A	Obsah CD	32
B	Ukázky UI	33

Kapitola 1

Úvod

Představte si, že večer sedíte doma u televize a náhle si vzpomene, že vám bylo minulou noc zima a zase jste si zapomněl zapnout topení v ložnici. Ale jste zničen po dlouhém dni, pohodlně usazení a nemáte sílu jít do druhého patra jen kvůli zapnutí topení. Nebo chcete vědět, jak je zrovna u vás na chatě, kam se chystáte. Jediné, na co se zmůžete, je natáhnout se pro ovladač. A tato činnost vám naprosto stačí, pokud máte zařízenou tzn. Inteligentní domácnost. Přímo na televizi nebo telefonu se můžete podívat kolik stupňů je v různých pokojích v domě a pár kliky nastavit požadovanou teplotu. Pokud máte zájem o ještě větší komfort, můžete systém nechat dělat spoustu věcí za vás. Například automatické udržování teploty v závislosti na teplotě okolí, času a informaci, zda jste vůbec doma. Můžete tak v klidu odejít do práce a dům za vás ztlumí topení, vypne světla, hudbu, a jiné zařízení. Před odjezdem z práce domů si můžete z mobilu nebo počítače opět vše zapnout. Inteligentní domácnost vám tedy sníží následné náklady a umožní komfortnější život.

Tato práce se zaměřuje na zobrazování a ovládání domácnosti přímo z adaptéru, který sbírá data ze senzorů (viz dále). Grafický přehled domácnosti je možné zobrazit na televizi nebo monitoru s HDMI. Za účelem úspory je grafická aplikace uspána v případě dlouhodobé nečinnosti, nebo pokud HDMI kabel není vůbec připojen. Výhodou tohoto řešení je, že pro zjištění aktuálních hodnot ze senzorů stačí samotný senzor, adaptér a zobrazovací zařízení. Tudiž máte stále možnost kontroly a ovládání i případě výpadku sítě. Součástí této práce je implementace aplikace *Ada Vis* v jazyce C++ s využitím grafické knihovny Qt pro embedded Linux.

V následujícím textu, v kapitole 2, je detailně popsána definice inteligentní domácnosti, informace o již existujících systémech, popis systému vyvíjený na FIT VUT v Brně a souhrnné zasvěcení do tématu inteligentních domácností. V kapitole 3 „Návrh“ jsou shrnuty informace, technologie a možnosti nástrojů důležité k realizaci aplikace. A to od uživatelsky neviditelných činností, jako meziprocesová komunikace a získávání dat, až po grafický návrh aplikace. V další kapitole 4 jsou pak popsány jednotlivé implementační prvky a oživení aplikace. V předposlední kapitole 5 je popsán průběh a výsledky základního uživatelského testování. V závěrečné kapitole 6 jsou informace o celkovém zhodnocení výsledku práce a zamýšlení nad další možnostmi vývoje.

Kapitola 2

Inteligentní domácnost

Pojem „Inteligentní domácnost“ znamená komplex různých systémů, které vzájemně spolupracují tak, aby pobyt v ní byl pohodlný, bezpečný a přitom energeticky úsporný. Zahrnuje integraci technologií do domácnosti a správu jejich prvků. Obecnější pojem „inteligentní dům“ většinou zahrnuje i správnou stavbu celé budovy. Každý výrobce/dodavatel se zaměřuje a nabízí něco jiného. Mají ale společný základ - centrální systém, který pomocí senzorů a aktivních prvků umožňuje automatizovat provoz domu. Celý tento systém je ovládán pomocí intuitivní obrazovky na libovolném nosiči (od smartphone přes notebook až po TV). Tato automatizovaná domácnost může spravovat například:

- tepelnou techniku - regulace různých zdrojů tepla,
- zabezpečení - kamerový systém, požární čidla, vypínání spotřebičů, simulace přítomnosti obyvatel domu,
- osvětlení - předdefinování světelných scén z různého množství světel různé intenzity (režimy jako sledování filmu, práce, atd.),
- zábavní elektroniku - multimediální centrum, rádio, prohlížení fotek z NASu,
- spotřebiče - spouštění kávovaru, pračky, sledování obsahu lednice, centrální vysavač, domácí pekárna,
- další speciální zařízení - žaluzie.

Data pro automatizované ovládání mohou být sbírána ze senzorů měřící elektrické i neelektrické veličiny, nebo z vlastní meteostanice umístěné na střeše.

Pořizovací náklady takto komplexního systému jsou dosud stále vysoké, ale z dlouhodobého hlediska se návratnost projeví po několika letech. Výhodou je, že není nutné pořídit vše najednou. Na začátku je hlavní investovat především do „inteligentní“ elektroinstalace, čímž je myšlena centrální jednotka a u nových domů i kabeláž. Další rozšiřující moduly (senzory, aktéry) lze do systému zapojovat postupně a tím neustále zvyšovat možnosti kontroly nad svým domem.

2.1 Možnosti ovládání

Zde je seznam možných vstupních zařízení, které je možné použít pro ovládání grafické aplikace. U každé položky jsou připsány jejich výhody a nevýhody.

Ovladač

- Výhody: uživatelé jsou na něj zvyklí z ovládání televize.
- Nevýhody: příliš mnoho kliků k provedení požadované akce, v běžné domácnosti je ovladačů již dost.

Myš

- Výhody: přímé kliknutí na požadovanou položku, nejlevnější zařízení kdekoli dostupné, bez nastavování.
- Nevýhody: ne vždy je k dispozici vhodná podložka.

Klávesnice

- Výhody: použití klávesových zkratk, nejrychlejší možnost, bez nastavování, možnosti specializovaných klávesnic s multimediálními tlačítky.
- Nevýhody: běžná klávesnice příliš velká, může překážet.

Klávesnice s touchpadem

- Výhody: kombinace klávesnice a touchpadu nahrazující myš, kompaktní rozměry, bezdrátové, malá spotřeba, již využívané u některých SMART TV a HTPC.
- Nevýhody: stále velké zařízení pro pohodlné ovládání.

Kinect

- Výhody: zábava, netřeba mít ovládací zařízení fyzicky v ruce.
- Nevýhody: složité uživatelské nastavování, ne příliš přesné.

Hlas

- Výhody: není potřeba dalšího zařízení.
- Nevýhody: potřeba tichého prostředí, složitá implementace, do jisté míry nespolehlivé.

Dotyková obrazovka

- Výhody: jednoduché přímé ovládání, práce jednodušší než s myší.
- Nevýhody: přizpůsobení grafiky aplikace (větší tlačítka), neřešitelné u velkých televizí.

Smartphone s bluetooth

- Výhody: komfortní ovládání.
- Nevýhody: potřeba nabitého mobilu, potřeba mít nainstalovanou a zapnutou aplikaci.

2.2 Popis datových zdrojů

Adaptér, který sbírá data ze senzorů a odesílá je na vzdálený server, se musí připojit přímo do sítě s přístupem na internet, což otevírá velké množství možností dalšího využití. Hlavní důvod tohoto připojení je pro komunikaci se serverem, který ukládá naměřené hodnoty a zpřístupňuje je pomocí mobilní či webové aplikace nebo i jinými způsoby. Na serveru mohou být také uložena schémata chování, která ovládají akční členy podle informací ze senzorů, podle časového scénáře nebo v ideálním případě kombinací obou variant. V případě společného serveru může mít uživatel dokonce více adaptérů z různých lokalit pod jednotnou správou. Možné zdroje informací na adaptéru tedy jsou:

- Aktuální hodnoty ze senzorů procházející adaptérem (meziprocesová komunikace),
- Data ze serveru s databází starších hodnot (lokálního i vzdáleného adaptéru),
- Data z jiných serverů (např. pro předpověď počasí).

2.3 Existující řešení

Společností zabývajících se inteligentní domácností existuje již mnoho. Každá se ale zaměřuje trochu jiným směrem. Zákazník by si tedy měl nejdříve uvědomit, co vlastně od systému očekává. Systémy mohou být tvořeny například pro:

- Firemní budovy a továrny - pro zvýšení efektivity,
- Hotely - pro jednodušší správu,
- Rodinné domy - pro snížení nákladů,
- Konferenční místnosti - pro komfortní ovládání,
- Byty, atd.

Na českém trhu patří mezi neznámější firma ELKO EP, s.r.o., která se zabývá všemi potřebnými aspekty a to od vývoje až po výrobu a instalaci. Ostatní společnosti jsou obvykle partnerem nějaké zahraniční firmy a vyvíjí jen některé subsystemy. Společným prvkem všech systému je modularita a možnost ovládání ze smartphonu.

System iNELS

Firma ELKO EP, s.r.o. nabízí systém iNELS - smart home solutions, který poskytuje jednoduchou správu interiéru i exteriéru. Například v interiéru je možné nastavit automatickou regulaci teploty, která je regulovaná podle vnitřních senzorů i venkovní meteostanice, a vypočítání přibližné spotřeby a ceny. Dále je možná regulace osvětlení, ovládání kuchyňských spotřebičů, sledování bezpečnostních kamer, automatické zavlažování, ovládání rolet podle počasí, správa multimediálního centra a jiné. Ovládání domácnosti lze realizovat přes RF klíčenku, smartphone, dálkový ovladač FR Pilot, dotykovou jednotku RF Touch nebo SmartTV (obr. 2.1). Většina zařízení je ovládána PLC, což je relativně malý průmyslový počítač používaný pro automatizaci procesů v reálném čase. PLC slouží především pro řízení strojů a výrobních linek v továrnách a je mnohem spolehlivější než běžný počítač. Systém iNELS již reálně běží na mnoha místech a to od malých bytů přes rodinné domy až po velké komerční prostory (například hotely nebo i tovární budovy).[4]



Obrázek 2.1: Ukázka systému iNELS na SmartTV.

System Lonoxe Miniserver

Společnost Loxone Electronics se zaměřuje se na vývoj a výrobu řešení pro ovládání domácnosti. Jejich hlavním cílem je poskytovat automatizaci domácnosti, která je výkonná a zároveň cenově dostupná pro každého. Specializují se především na vývoj a návrhy. Komunikaci se zákazníky přenechávají svým obchodním partnerům. Srdce domácí automatizace tvoří centrála Loxone Miniserver ve dvou typech: optimální pro nové domy a optimální pro hotové domy. Verze pro nové domy zahrnuje natažení potřebné kabeláže mezi moduly a miniserverem, ale ve výsledku nikde nepřekáží kabely a jednotka je vhodně schována. Verze pro již hotové domy pracuje bezdrátově. Centrála je vždy nastavena dle požadavků zákazníka a údajně je možné k ní připojit téměř jakákoliv chytřejší elektronické zařízení. Od osvětlení, stínění, multimédií, alarmu až po automatizovanou regulaci vytápění a řízení energií. Loxone je jedinečný tím, že svůj software nabízí k hardwaru zdarma včetně pravidelných aktualizací. Ovládání domácnosti je možné ze smartphonu, tabletu a webového prohlížeče.[9]

System AMX

AMX je hlavní divizí americké společnosti Harman, která navrhuje, vyrábí a prodává širokou škálu infotainmentu a audio řešení pro automobilové, spotřební a profesionální trhy. Divize AMX se zaměřuje na integraci AV řešení pro IT svět. Nejčastěji je možné na tento systém narazit v konferenčních místnostech, učebnách, velitelských centrech a hotelích. Poskytuje jednoduché ovládání osvětlení, žaluzií, bezpečnostního systému, vytápění, závěsů, bazénu, domácího kina, biometrického přístupu, domácích spotřebičů a dalšího vybavení. Pro ovládání se využívá primárně dotykových ovládacích panelů a RF ovladačů přímo od firmy AMX, a nebo aplikace ve smartphonu.[1] Pro český trh tento systém lokalizuje a poskytuje firma Insight Home, a.s.

System Eaton xComfort

Firma Eaton Elektrotechnika s.r.o. nabízí rozšířený modulární systém, který umožňuje ovládat světla, topení, žaluzie a rolety, zásuvky a kamery. Pomocí senzorů a obohacených aktorů je možné i sledovat spotřebu plynu, vody a elektra. Adaptér (RF Smart Manager) ovládající všechna zařízení, může sloužit i jako domácí server. Pokud chce ale uživatel mít dostupné ovládání domácnosti i mimo lokální síť, musí mít buď veřejnou IP adresu, nebo využít cloudových služeb firmy. Při instalaci není potřeba rozvádět po domě novou kabeláž, jelikož vše funguje bezdrátově přes RF¹. Příklad: pro vzdálené ovládání vypínače světel je možné vyměnit běžný vypínač za drahý RF vypínač, nebo nechat svůj původní a za něj připojit spínací aktor. Aktor je tak napájen přímo ze sítě a není nutné řešit baterie. Vypínač tak dále může plnit svou funkci, plus pomocí aktoru je možné jej dálkově ovládat a měřit spotřebu. Výhodou tohoto řešení je, že je možné vypínač běžně používat i v případě poruchy systému.

Grafická aplikace pro Android OS či iOS nabízí velké množství způsobů ovládání a sledování, což ale má za následek, že je aplikace zbytečně složitá. Zobrazení pro SmartTV je vyřešeno přes integrovaný webový prohlížeč. Není tak potřeba řešit rozdílné parametry různých výrobců TV, ale ovládání webové aplikace dálkovým ovladačem není moc pohodlné.[2]

2.4 System vyvíjený na FIT

Na fakultě FIT-VUT v Brně se inteligentní domácností zabývá výzkumná skupina IoT (Internet of Things). Aby bylo možné systém neustále rozvíjet, je architektura navržena jako vícevrstvá stavebnice. Výzkumná skupina se zabývá vývojem všech vrstev systému (obr. 2.2):

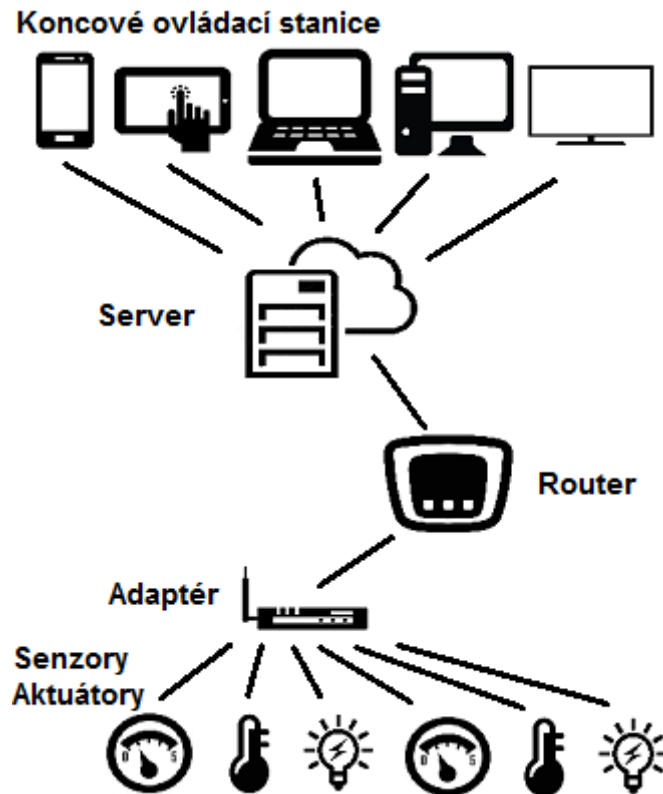
- Android aplikace - zobrazování informací a nastavování,
- Server - uchovávání dat,
- Adaptér - sběr dat z bezdrátových prvků a odesílání na server,
- Bezdrátové prvky - senzory a aktuátory.

Nejnižší vrstvu architektury tvoří bezdrátové prvky dvojího typu:

- Senzory - snímají nějakou veličinu a odesílají hodnoty adaptéru.
- Aktuátor - prvek, který nějakým způsobem zasahuje do prostředí (např. spínání zásuvky).

Oba tyto typy prvků komunikují bezdrátově pomocí protokolu MiWi (od společnosti Microchip) s adaptérem. Pro spárování prvku k adaptéru je potřeba s prvkem zatřepat, tím přejde do režimu párování a zadat číselný kód. Po spárování prvky pracují převážně v režimu spánku, čímž se sníží spotřeba a zvýší životnost baterie. Adaptér komunikující s prvky je připojen do běžného domácího routeru s připojením k internetu. Touto cestou adaptér komunikuje se vzdáleným serverem, na který posílá informace ze senzorů nebo přijímá příkazy pro aktuátory. Server si ukládá naměřené hodnoty, zajišťuje správu účtu a adaptérů. K serveru se připojují koncové ovládací stanice - smartphone, tablet, notebook, pc, smartTV atd.[8]

¹Radio frequency - rádiové vlny



Obrázek 2.2: Architektura systému pro inteligentní domácnost.

Nejsilnější stránkou tohoto systému je, že není nutná žádná fyzická instalace, tudíž si jej uživatel může zprovoznit sám i bez technických znalostí. Postup na zprovoznění je následující:

1. Vybalení všech zařízení z krabice.
2. Stažení aplikace BeeeOn, zaregistrování a přihlášení uživatele.
3. Zapojení adaptéru do elektrické sítě a domácí LAN sítě (s připojením na internet).
4. Přidání adaptéru (naskenování QR kódu).
5. Oživení senzoru vytažením proužku u baterky.
6. Spuštění párování v mobilní aplikaci a zatřepání se senzorem (v případě aktuátoru stisknout tlačítko).
7. Vlastní pojmenování zařízení.

Celý systém dostal finální název *BeeeOn*.

Kapitola 3

Návrh

Z důvodu složitosti celého systému jsou zde shrnuty potřebné informace pro implementaci. I když nejsou všechny následující technologie využity, bylo nutné zjistit jejich možnosti pro výběr té nevhodnější.

3.1 Meziprocesové komunikace

Aplikace pro sběr dat ze senzorů a přeposílání na server je primárním programem na adaptéru. Aby byl celý systém stabilnější, je vhodné, aby jakékoliv další úkoly běžely jako samostatné programy. Proto grafická interpretace aktuálních hodnot ze senzorů je řešena jako samostatná aplikace. Toto modulární řešení systému zvyšuje stabilitu, ale vyvstává problém, jak donutit aplikace spolu komunikovat. Meziprocesovou komunikaci (anglicky *Inter-Process Communication* - IPC) můžeme rozdělit na základní techniky: sdílená paměť, synchronizace, vzdálené volání procedur, zasílání zpráv.

Sdílená paměť

Sdílená paměť (anglicky *shared memory*) je část operační paměti, která slouží pro více subjektů (procesy, procesory, atd.). Příkladem může být výměna dat mezi programy běžícími současně. Proces vytvoří segment sdílené paměti pomocí funkce `shmget()`. Po vytvoření může být segment připojen k procesu pomocí `shmat()`. Jakmile je proces připojen, je možné do segmentu zapisovat nebo číst. Segment je popsán řídicí strukturou s jedinečným identifikátorem (`shmid`), který ukazuje na fyzickou oblast v paměti. Je-li povolen zápis pro více než jeden proces, může být použit semafor, aby se zabránilo kolizím a nesrovnalostem. Příliš nízkourovňové programování, pro větší projekty nevhodné.

Synchronizace

Synchronizace - koordinace dvou nebo více dějů současně. Tato metoda se převážně využívá u synchronizace dat mezi vlákny v rámci jedné aplikace. Vlákna v různých procesech mohou vzájemně komunikovat přes synchronizační objekty umístěné ve vláknem řízené sdílené paměti i přes to, že vlákna jsou pro sebe neviditelné navzájem. Možné způsoby synchronizace vláken: vzájemné vyloučení zámky (`mutex`), semaforey, monitory.

Vzájemným vyloučením zámky se dá docílit toho, že pouze jedno vlákno v jednu chvíli provádí kritickou část kódu. Za kritickou část kódu je obvykle označena ta část kódu, kde se

provádí čtení nebo zápis do proměnných více vláknů. Funkce zámku mutex jsou: inicializace, zničení, zamknutí, odemčení.

Semafor je synchronizační primitivum obsahující celočíselný čítač, který slouží jako počítadlo volných prostředků. Poskytuje dvě atomické operace: down (lock) a up (unlock). Operace „lock“ dekrementuje čítač, je-li čítač menší nebo roven nule, proces se zablokuje a přidá do fronty čekajících. Operace „unlock“ inkrementuje čítač, který když je větší jak nula vybere jeden proces z fronty čekajících a odblokuje jej.

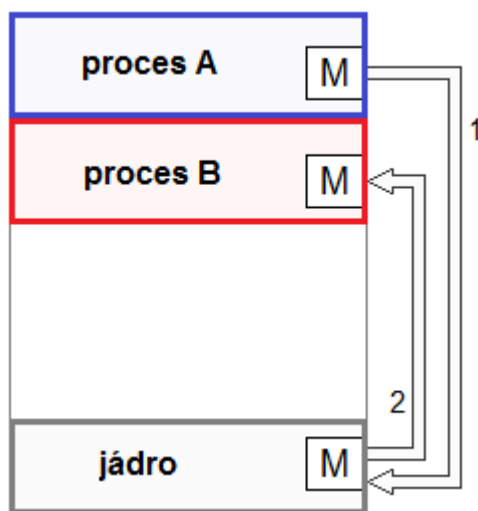
Monitor je synchronizační vysokoúrovňové primitivum, které zapouzdřuje data a pouze jeden proces může provádět operace nad chráněnými daty. Možné operace: enter, leave, wait, notify. Implementace je možná i pomocí semaforů.[10]

Vzdálené volání procedur

Vzdálené volání procedur (anglicky *remote procedure call - RPC*) je silná technika pro stavbu distribuované aplikace na bázi klient-server. Technologie umožňuje programu volání procedur, která jsou uložena jinde než volající program. Tyto procesy mohou být na stejných nebo i různých systémech síťově zapojených. Klient zavolá proceduru, která odešle požadavek na server a čeká. Proces je blokován, dokud neobdrží odpověď, nebo vyprší časový limit. Server vykonává rutinu a pošle klientovi odpověď. Proces dále pokračuje. Výhodou je možnost využití výpočetního výkonu vzdálených serverů. Nevýhodou je potřeba síťového připojení a nelze využívat globálních proměnných.

Zasílání zpráv

Je to metoda závislá na platformě. V případě jádra Linux je to nejpoužívanější metoda. Zasílání zpráv pracuje na dvou základních operacích: `send()` a `receive()`. Čtení a zápis zpráv může být blokující či neblokující a adresování přímé či nepřímé. Názorná ukázka je na obrázku 3.1. Podrobnější popis je na následujících konkrétních příkladech.



Obrázek 3.1: Ukázka posláni zprávy z procesu A přes jádro procesu B.

D-Bus (Desktop Bus) je sběrniceový systém, který používá metodu zasílání zpráv pro vzájemnou komunikaci procesů. Je implementován jako daemon a je možné jej spustit ve

více instancích. Umožňuje programům nabízet své služby ostatním aplikacím. V případě systémového D-Bus daemona je tento systém využit například pro detekci nového hardwarového zařízení. Architektonicky se D-Bus skládá ze tří vrstev:

- Knihovna *libdbus*, která slouží k propojení dvou aplikací a výměnu zpráv,
- Daemon pro zasílání zpráv, ke kterému se mohou připojovat aplikace. Zasílání zpráv je možné od jedné aplikace k žádné nebo více aplikacím,
- Obalové knihovny nebo vazby na aplikační framework.

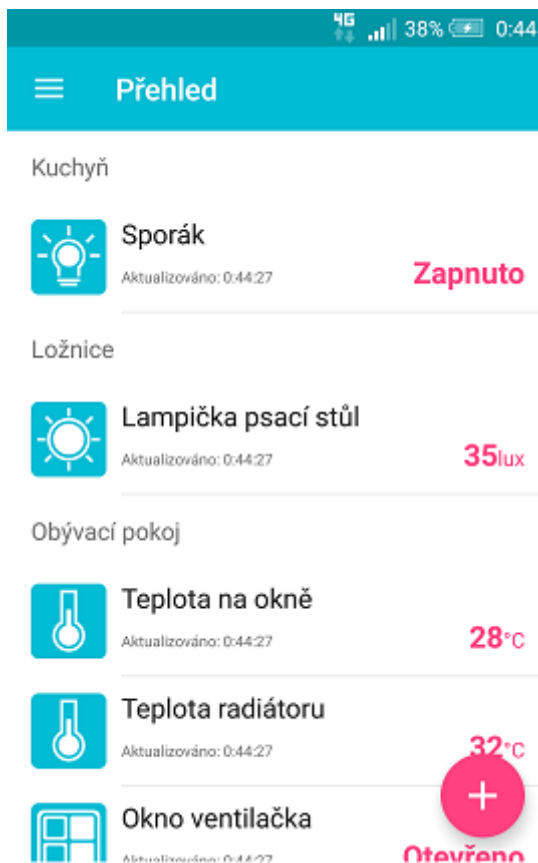
Zprávy jsou zasílány konkrétnímu objektu, nikoliv celé aplikaci. Proto musí být aplikace využívající D-Bus objektová. Každému objektu se přiřadí jméno ve tvaru souborového systému.[5] Původně bylo v plánu využívat právě D-Bus, ale kvůli nejednotným, stále se vyvíjejícím knihovnám se od této možnosti ustoupilo.

Pojmenovaná roura vychází a rozšiřuje koncept Unixových rour a umožňuje komunikaci nepříbuzných procesů. Práce s pojmenovanou rourou je stejná jako s běžným souborem. Na začátku komunikace musí být vytvořena jako „speciální soubor“, po ukončení je nutné ji smazat. Dále stačí využít standartních funkcí pro práci se souborem: `open()`, `read()`, `write()`, `close()`. Touto cestou nyní *AdaVis* získává data ze senzorů.

Mosquitto je tzv. *message broker*. Open source pro komunikaci zařízení typu „Internet of Things“ založený na protokolu MQ Telemetry Transport. Poskytuje jednoduchý způsob posílání zpráv pomocí modelu publikování a odběru. Je vhodný pro komunikaci „stroj stroji“ u zařízeních s nízkou spotřebou, jako třeba senzory, mikroprocesory či mobilní zařízení.[11] Práce s tímto nástrojem je snadnější alternativa k D-Bus.

3.2 Grafické zobrazení

Existuje mnoho způsobů jak data zobrazovat. Od zobrazení „holých“ dat pouze v terminálu až po grafické rozhraní mnoha stylů na různých zařízeních. My se snažíme uživateli co nejvíce zjednodušit a zpříjemnit práci se systémem. Toho lze docílit tím, že se snažíme udržet na všech zařízeních podobný styl zobrazení a ovládání. Proto tato grafická aplikace vychází z existujícího návrhu webového rozhraní a aplikace pro OS Android (obr. 3.2). Ovšem je rozdíl mezi zobrazením na smartphonu a velkou televizí. Cílem je vytvořit jednoduché, přehledné, intuitivní prostředí, které uživatele neodradí od jeho používání. Paradoxně by ale nemělo být příliš jednoduché, jinak aplikace uživatele nezaujme nebo jej po chvíli používání přestane bavit. V aplikaci *AdaVis* bude mít uživatel možnost si nahrát vlastní půdorys domu a libovolně přiřadit senzory jednotlivým pokojům. Pro ovládání jsem zvolil podporu pro myš a klávesnici s multimediálními tlačítky.



Obrázek 3.2: Ukázka mobilní aplikace BeeeOn pro OS Android.

3.2.1 Možnosti grafického výstupu

Pro komunikaci uživatele s centrálním systémem je zapotřebí nějaká koncová ovládací stanice. Pro ovládání inteligentní domácnosti se dnes běžně používají:

Smartphone, tablet

- Výhody: ovládání dotykem, malé zařízení, obvykle hned po ruce.
- Nevýhody: malá velikost displeje.

Notebook, PC

- Výhody: velké rozlišení, přesné ovládání.
- Nevýhody: uživatel jej nemusí mít vždy u sebe.

SmartTV

- Výhody: možnost náhledu stavu celého domu.
- Nevýhody: nemobilní, nejednotná funkčnost.

Všechna tato zařízení mají jednu společnou nevýhodu: systém inteligentní domácnosti využívá dostupné domácí LAN síť pro propojení adaptéru se serverem, takže dostupnost

správy domácnosti závisí na stabilitě LAN sítě. V případě požadavku ovládat systém mimo domácí síť, má na dostupnost informací vliv i stabilita internetového připojení.

Proto tato práce otevírá další možnost, a to přístup k sensorům a aktorům i v případě výpadku LAN sítě. Grafická aplikace běžící přímo na adaptéru umožňuje zobrazovat a ovládat některá zařízení v omezené míře. Nevýhodou tohoto řešení je nutnost být fyzicky poblíž adaptéru, který je připojen k monitoru nebo TV. Možnosti ovládání byly shrnuty již výše. Zobrazování grafické aplikace je možné na monitoru nebo TV obsahující některý z konektorů:

- **VGA** - analogový grafický výstup (používán u CRT, LCD monitorů a projektorů), snaha odstoupit od tohoto rozhraní.
- **DVI** - digitální grafický výstup (používán u LCD monitorů, monitory a novější zobrazovací zařízení).
- **HDMI** - rozhraní pro přenos nekomprimovaného obrazového a zvukového signálu ve vysoké kvalitě v digitálním formátu.
- **DisplayPort** - totéž jako HDMI a navíc obsahující ochranu šifrováním AES, je zpětně kompatibilní s konektory DVI a HDMI.

Jelikož náš adaptér (viz. 3.5) disponuje pouze HDMI, bude se tato práce dále zaměřovat pouze na toto rozhraní.

3.2.2 Návrh obrazovky

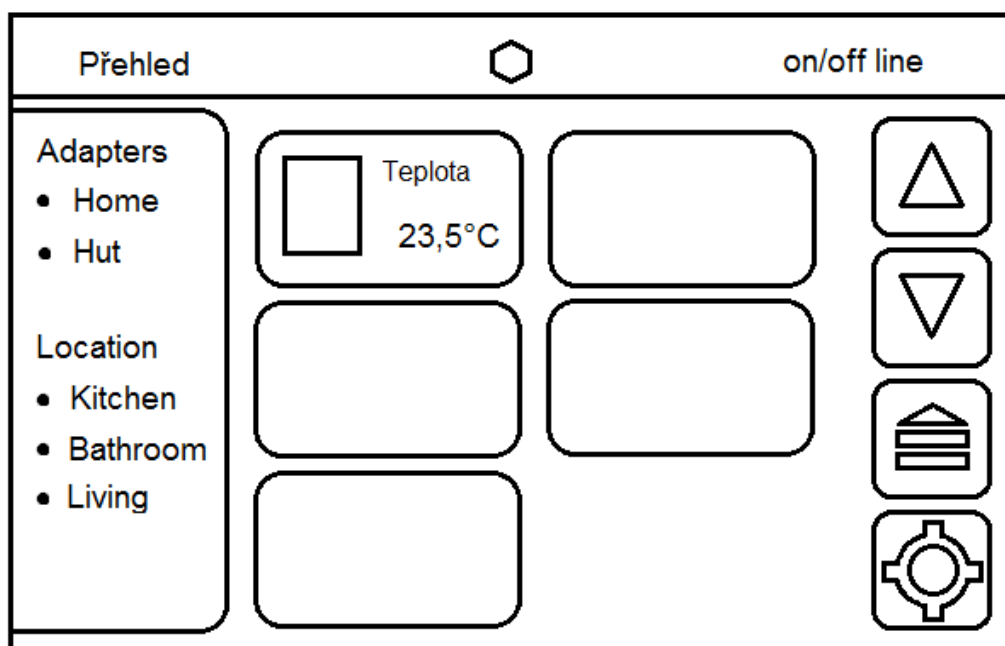
Před začátkem implementace byla potřeba provést návrh uživatelského rozhraní. Požadavkem bylo zobrazit všechna data, která senzory a aktory poskytují, tj. například teplotu, vlhkost, stav spínače, tlak, atd. Myšlenkou každé pro-uživatelské aplikace je intuitivní a jednoduché rozhraní. Aplikace *AdaVis* je převážně informačního typu, proto nesmí uživatele obtěžovat dialogovými okny, které je potřeba potvrdit. Na rozdíl od mobilní aplikace je na monitoru více prostoru pro přehlednější zobrazení. I tak je potřeba počítat s variantou, že si uživatel může připojit monitor běžné velikosti, nebo velkou televizi. Vykreslení tedy musí být dynamické a rozměry zadané v procentech. Ovšem minimální podporované rozlišení bude 800x600 pixelů.

Hlavní náhled vychází z návrhu webové aplikace, ostatní je vlastní nápad s kombinací existujících konkurenčních řešení. Nahoře informační panel pevné výšky pro jednodušší orientaci. Napravo ovládací panel s jednotným stylem tlačítek. Vnitřek, obalen těmito panely, bude proměnný dle typu zvoleného náhledu.

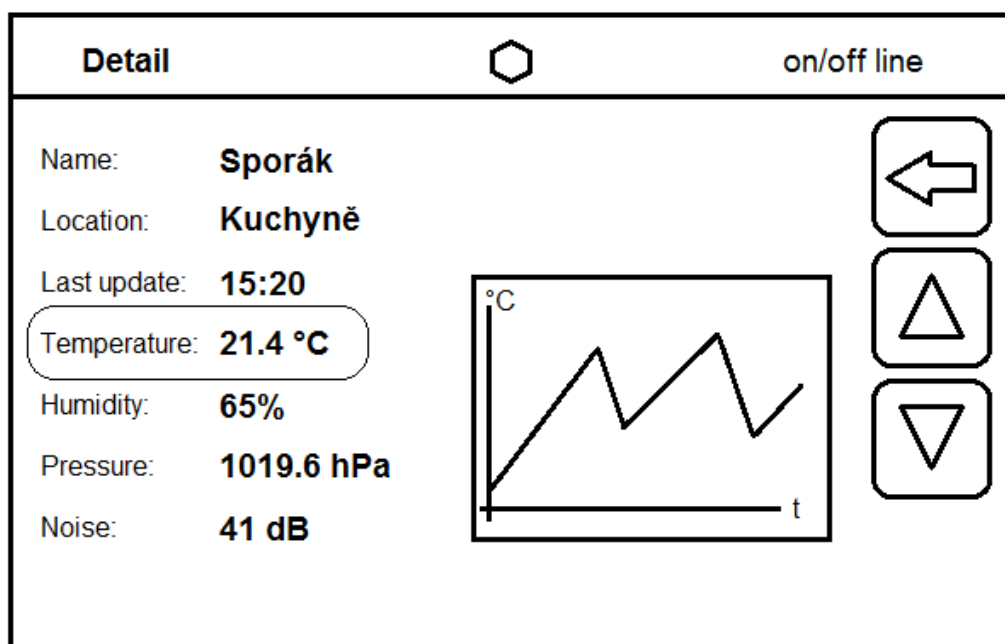
Na obrázku 3.3 je možné vidět rozložení prvků. Tlačítka budou velká pro jednodušší trefování myší. Levý panel zobrazí uživateli adaptéry a místnosti na vybraném adaptéru. Uprostřed budou boxy s většinou informací o senzoru. Kliknutím na box se zobrazí detail s náhledem na staré hodnoty ve formě grafu a další a podrobné informace (obr. 3.4). Zde bude možné provádět i nějaké základní ovládání senzoru/aktuátoru.

Další náhled (obr. 3.5) bude ve formě ikonky na půdorysu domu/bytu. Uživatel si bude moci nahrát vlastní plánek nebo vybrat z předchystaných. Ikony bude možné pomocí myši přesouvat libovolně po plánu a budou zobrazovat jeden typ zvolené hodnoty. Přepínání mezi typy hodnot bude možné pomocí pravého panelu.

Barevnou paletou a stylem se pokusím přiblížit k existující mobilní aplikaci (obr. 3.2). Když bude chtít uživatel zobrazit data z jiného adaptéru, ale nebude možné je získat,

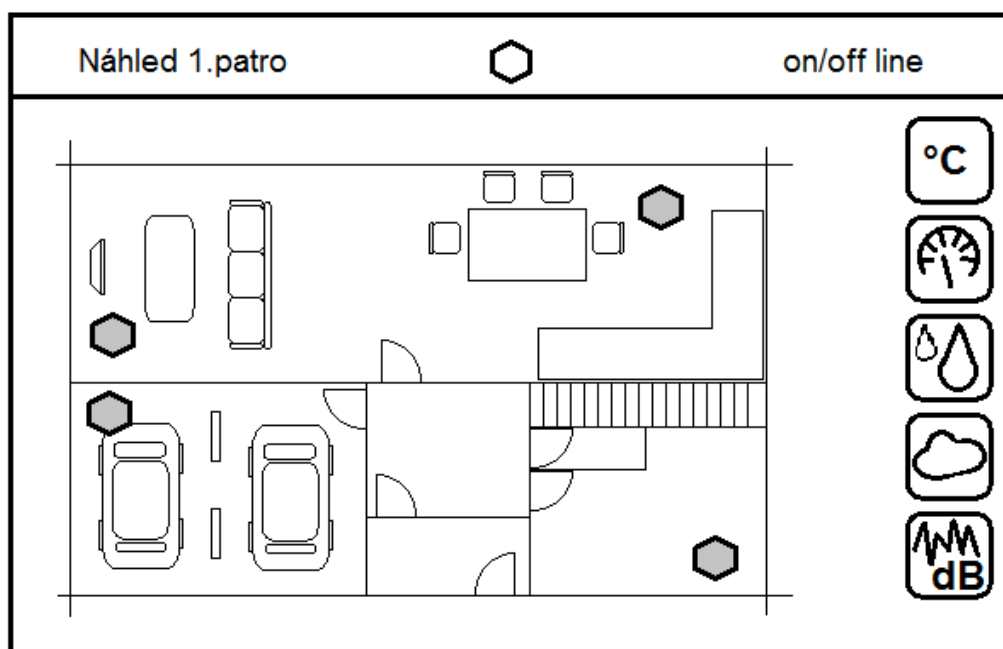


Obrázek 3.3: Návrh úvodního náhledu.



Obrázek 3.4: Detailní výpis informací o senzoru.

vypíšou se poslední dostupná data šedě nebo se pošle dotaz na server a podle stáří dat se zobrazí normálně nebo šedě. Notifikace od serveru se budou zobrazovat na libovolném náhledu v průhledném rámečku na krátký časový interval.



Obrázek 3.5: Návrh plochy s půdorysem a možností Drag&Drop.

3.3 Úspora energie

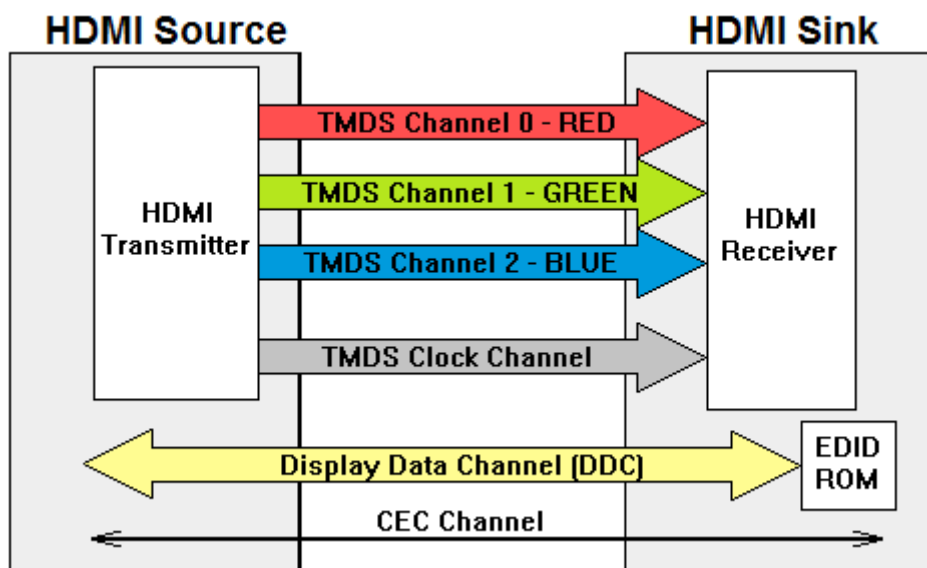
Za předpokladu, že běžný uživatel nesedí celý den jen u televize, aby sledoval stav domácnosti, je poněkud neefektivní, kdyby navrhovaná grafická aplikace stále běžela. Aby aplikace nemusela stále sledovat změnu dat ze senzorů a posílat obraz na výstup, je vhodné aplikaci uspat. To může být řešeno stopováním času od poslední uživatelské akce a po určitém časovém intervalu přejít do režimu spánku. Opětovné probuzení aplikace lze provést některým ze vstupních zařízení. Asi nejjednodušší možností je pouhé pohnutí myši. V tomto případě ale nelze aplikaci úplně vypnout. Druhou variantou může být využití zpětného kanálu HDMI, který bude aplikaci informovat, zda je sledována, či nikoliv. Může ale existovat i situace, kdy si uživatel přeje mít grafický výstup stále aktivní. Proto tato funkce musí být nastavitelná.

3.4 Komunikace s monitorem

Rozhraní HDMI nabízí komunikační kanál **DDC** (Display Data Channel), který umožňuje komunikaci mezi monitorem a grafickou kartou na dvoudrátové sběrnici I^2C , čímž může monitor sdělit, jaké režimy zobrazení podporuje a další informace pro korektní zobrazení. Tyto informace jsou uloženy v datové struktuře zvané EDID (Extended Display Identification Data), která popisuje své schopnosti zobrazení. EDID obsahuje jméno výrobce, výrobní číslo, typ monitoru, typ filtru, časování, velikost displeje, atd. a je uloženo v zobrazovacím zařízení v paměti typu EEPROM.[7] Díky tomu uživatel nemusí ručně nastavovat rozlišení monitoru. Bohužel jen málokteré monitory poskytují informaci, zdali je monitor zapnutý či vypnutý. Pro Linux existuje nástroj „ddccontrol -p“, ale tento program je použitelný v případě, že monitor umožňuje ovládání po kabelu HDMI. Navíc podpora

pro ddecontrol je od roku 2006 bez aktivity. Proto tuto možnost v praxi nemohu použít.

Některá zařízení (např. RaspberryPi) mají integrovanou podporu CEC (Consumer Electronics Control) přes HDMI i USB porty. **HDMI-CEC** umožňuje počítači ovládat až 10 zařízení CEC připojené přes HDMI pomocí jednoho ze svých dálkových ovladačů (např. TV, set-top box, atd.). V ideálním případě je tak možné přes HDMI zapnout či vypnout TV, upravovat hlasitost, nebo i ovládat další domácí zařízení jako třeba DVD nebo Blu-ray.[3] Přehledný náhled kanálů je na obrázku 3.6. Hardware Olimex Lime s procesorem A20 je podstatě schopen tento kanál používat, ale doposud nebyl implementován žádný ovladač.



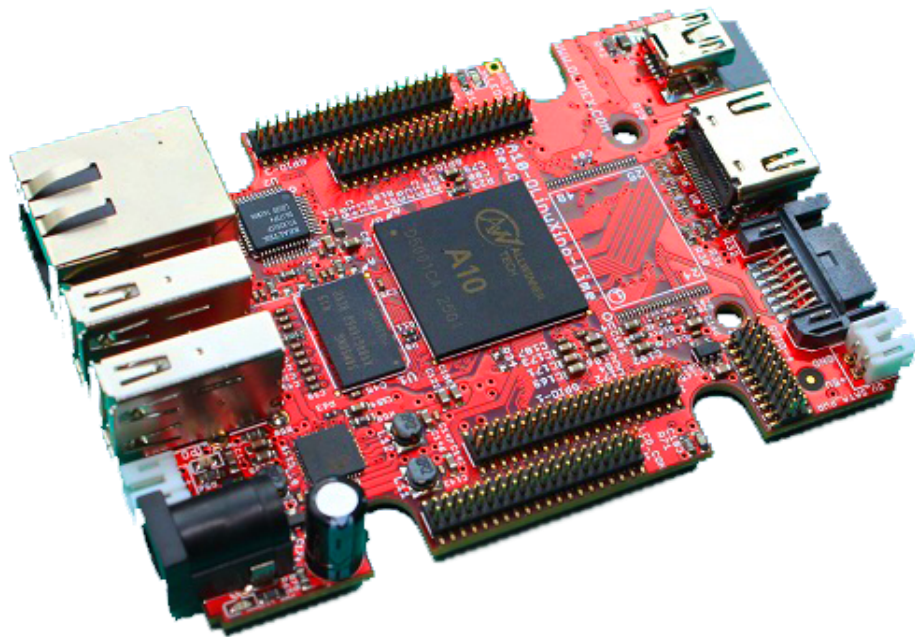
Obrázek 3.6: Přehled kanálů v konektoru HDMI¹.

3.5 Cílové zařízení

Adaptér, na kterém běží aplikace *Ada Vis*, je jednodeskový počítač o velikosti platební karty. Konkrétně se jedná o minipočítač A10-OLinuXino-LIME od firmy Olimex Ltd (obr. 3.7). Má v sobě armový procesor Allwinner A10 1GHz Cortex-A8 ARMv7 a 512MB DDR3 RAM.² Je považován za vestavěné zařízení určené pro předem definovanou činnost, kterou si uživatelé ani moc neuvědomují. Proto je při návrhu potřeba počítat s relativně omezenými zdroji a možnostmi, tj. obvykle menší velikost paměti RAM a úložné kapacity a také menší rychlosti procesoru než na stolním počítači. Proto je do něj nahrán tzv. zabudovaný Linux (anglicky *embedded Linux*), který je pro tato zařízení přizpůsoben. Příkladem embedded Linux je například Debian for ARM, Ubuntu Mobile, Android a jiné.

¹Obrázek převzat z webu: <https://www.audioauthority.com/images/misc/hdmichart.gif>.

²<https://www.olimex.com/wiki/A10-OLinuXino-LIME>



Obrázek 3.7: Jednodeskový počítač A10-OLinuXino-LIME využitý jako adaptér.

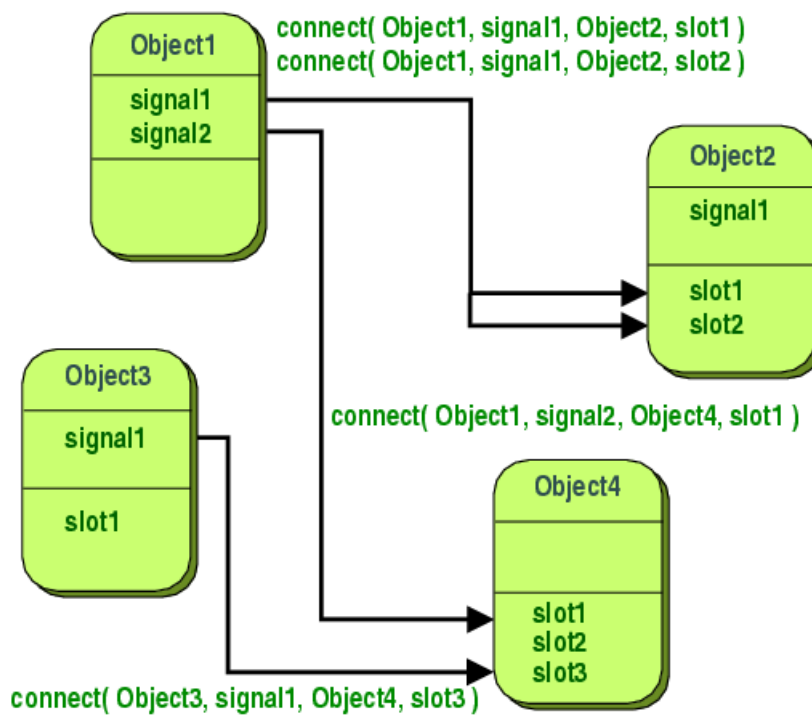
3.6 Knihovna Qt

Qt je multiplatformní knihovna pro C++, Python a další programovací jazyky usnadňující tvorbu grafického uživatelského rozhraní na různých desktopových i mobilních platformách. Knihovna nabízí i rozhraní pro práci s vlákny, sítí, databází, atd. Nástroj *Qt Designer* nabízí možnost vytvářet grafické prostředí metodou WYSIWYG³, které je jednoduché na používání, ale grafické prvky se nedají plně nastavit. Druhá možnost je programovat v kombinaci s jazykem QML. Třetí možností, kterou jsem si zvolil, je pracovat s prvky jako s běžným objektem na úrovni programovacího jazyka (v mém případě C++).

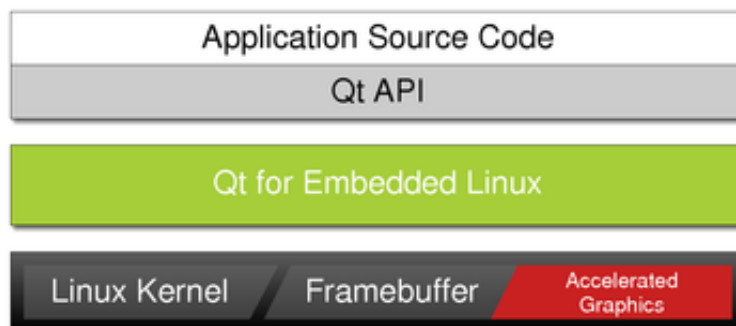
Důležitou předností této knihovny je práce se signály a sloty pro komunikaci mezi objekty (obr. 3.8). Jde tedy o událostně řízené programování. Události se převádí na signály a handlers signálů jsou sloty. Příklad: tlačítko při stisku vyvolá a pošle signál na slot jiného objektu, který provede nějakou svoji předdefinovanou úlohu.

Pro instalaci aplikace na vestavěné zařízení je potřeba si zkompileovat verzi pro embedded Linux (obr. 3.9). Protože vestavěné zařízení obvykle nemá dostatek paměti pro objemnou knihovnu Qt, a někdy ani výkon pro kompilaci, využívá se tzv. křížové kompilace (anglicky *cross compilation*). Ta umožňuje vytvořit spustitelný kód pro jinou platformu, než na které je kód kompilován a výslednou aplikaci již pouze nakopírovat.

³z angličtiny „What You See Is What You Get“ - Co vidíš, to dostaneš.



Obrázek 3.8: Schéma znázorňující práci se signály a sloty v Qt.



Obrázek 3.9: Qt pro embedded Linux.

Kapitola 4

Implementace

Tato kapitola popisuje implementační detaily aplikace *Adavis*, její zprovoznění na embeddid Linux a připojení k systému *BeeOn*. Během implementace grafické části byla snaha se držet principu KISS¹, tj. zajistit jednoduché používání bez návodu. Aplikace byla vyvíjena na linuxové distribuci Xubuntu 14.04 a pak nahrána a oživena na embeddid Debian GNU/Linux 7.0. Implementace byla provedena v jazyce C++ s frameworkem Qt 5.4 ve vývojovém prostředí QtCreatoru.

4.1 Popis grafické části

Rozhraní se drží jednotné palety barev a to kombinace azurové (modré) a růžové, stejně jako v ostatních aplikacích pro *BeeOn*.

Horní panel (obr. 4.1) je neměnný a zobrazuje: slovní popis, logo, stav připojení a aktuální čas. Slovní popis (pojmenování náhledu) slouží pro rychlou orientaci. Logo uprostřed je vlastně tlačítko a umožňuje návrat do výchozího náhledu. Indikace „ONLINE“ nebo „OFFLINE“ informuje o stavu připojení adaptéru k serveru. V případě stavu *offline* aplikace *AdaApp* ukládá data do své dočasné paměti a domácnost je nedostupná z mobilních zařízení. To se ale netýká této aplikace, která stále zobrazuje aktuální data.



Obrázek 4.1: Horní panel aplikace *AdaVis*.

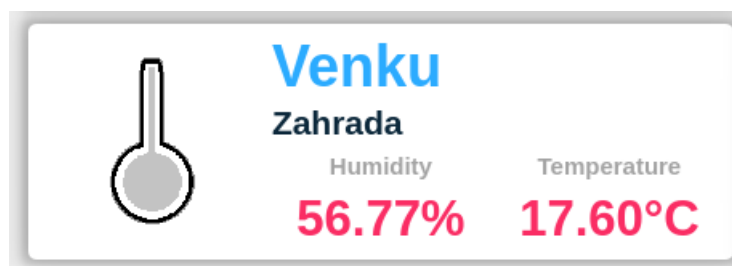
V pravém panelu (obr. 4.2) se vždy zobrazuje seznam velkých ovládacích tlačítek. Velikost tlačítek je takto nastavena pro jednoduché a rychlé trefování v případě používání myši. Jejich funkce se podle náhledu mění, ale logika obrázků zůstává stejná. V případě rozšíření možností práce s aplikací se ale budou muset zmenšit.

¹Keep It Simple, Stupid



Obrázek 4.2: Ovládací tlačítka aplikace AdaVis.

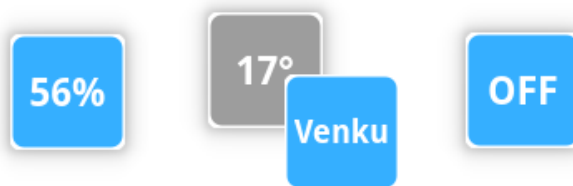
Výchozí náhled uprostřed zobrazuje boxy dostupných senzorů a aktorů se všemi jejich informacemi. Maximální počet boxů je 8 z důvodu zobrazení všech informací a přehledné čitelnosti. Pořadí boxů v této verzi odpovídá pořadí v jakém byli zařízení spárovány. Pro zobrazení dalších zařízení je nutno rolovat. Každý box (obr. 4.3) obsahuje ikonku pro zorientování, o jaký typ se jedná, vlastní pojmenování, zařazení do místnosti. Pak velkým písmem hodnoty, které zařízení sleduje (teplota, vlhkost, stav on/off, atd.). Nad hodnoty jsou popisky, o jaký typ hodnoty se jedná, které je v nastavení možné vypnout. Dříve na tomto místě bylo datum a čas poslední změny, ale tato informace je na lokálním zařízení bezvýznamná (na mobilní aplikaci už ale užitečná je). Nabízí se, aby byl box klikatelný pro zobrazení více informací, jenže vše je již zobrazeno. Tohle by byla cesta například pro vykreslení grafu, ale protože adaptér si neukládá staré záznamy a komunikace se serverem je zatím jednosměrná, není to pro *AdaVis* realizovatelné.



Obrázek 4.3: Náhled informací senzoru v boxu na aplikaci AdaVis.

Levý panel zobrazuje počet spárovaných zařízení a nabízí přepínání mezi adaptéry pro případ, že by uživatel vlastnil více adaptérů (např. doma, v kanceláři, na chatě). Dále seznam místností, který funguje i jako filtr pro zobrazení pouze zařízení, které se v dané místnosti nacházejí.

Přepnutím do vizuálního náhledu je možné vidět všechna zařízení současně ve formě ikonky (obr. 4.4) na půdorysu domu/bytu. V nastavení je možné si nahrát vlastní půdorys a něm si libovolně rozmístit ikony senzorů a aktorů. Kliknutím na ikonu se dočasně zobrazí název zařízení a je možné s ním libovolně přesouvat (Drag&Drop), přičemž se po dobu stisku myši zanechává zašedlá ikona na původním místě. V pravém panelu lze tlačítka přepínat požadovaný typ zobrazovaných hodnot na všech ikonkách.



Obrázek 4.4: Ikonky s hodnoty v aplikaci AdaVis.

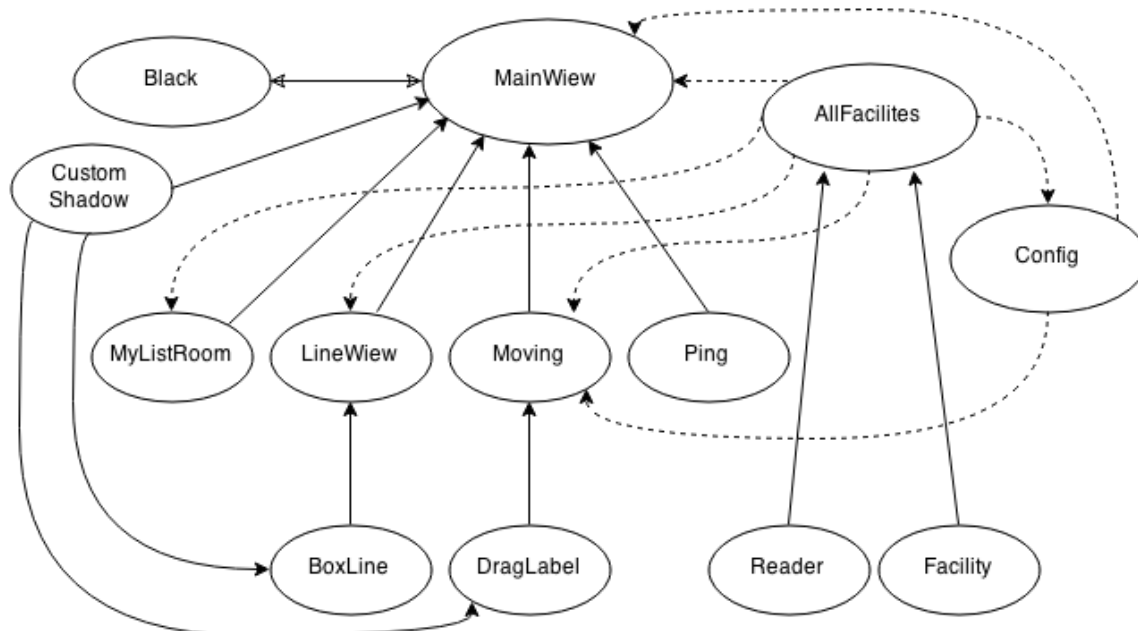
Tlačítkem s ikonkou řádků se zobrazí zjednodušený pohled na všechna zařízení. Tento pohled se snaží zobrazit pokud možno všechna spárovaná zařízení. Zobrazovány jsou pouze názvy senzorů a všechny jejich dostupné hodnoty. V případě malého množství zařízení jsou boxy roztaženy po celé šířce. S narůstajícím počtem se boxy zúží a začnou se tvořit sloupce. V případě vícero typů hodnot bude nějaká zanedbána.

V nastavení je zatím možné nahrát a vybrat si vlastní mapu (půdorys), povolit roztažení nahraného obrázku, nastavení času pro usnutí a vypnutí.

V případě nečinnosti uživatele (není detekován pohyb myši) obrazovka zčerná a pozastaví se proces čtení dat, testování spojení a aplikace se uspí. K opětovnému probuzení stačí libovolný pohyb myši.

4.2 Popis objektového návrhu

V této části je popis nejdůležitějších tříd. Podrobný popis všech tříd a funkcí je k dispozici v dokumentaci. Vazby mezi třídami jsou znázorněny na obrázku 4.5. Z důvodu přehlednosti a velkého množství metod u některých tříd, jsou třídy zobrazeny pouze zjednodušeným diagramem. Plné čáry značí zahrnutí do objektu ve směru šipky. Přerušované čáry značí vypůjčení ukazatele na instance návrhového vzoru Singleton.[6]



Obrázek 4.5: Přehled tříd aplikace AdaVis.

- **MainWiev**
Hlavní widget zobrazující celé grafické okno. Třída dědí grafické prvky operačního systému z Qt knihovny QMainWindow a je rodičem pro ostatní třídy rozšiřující grafickou část. Sama vykresluje výchozí náhled s boxy a pravý panel tlačítek.
- **Facility**
Univerzální struktura se všemi informacemi o každém zařízení s inicializačními hodnotami.
- **AllFacilites**
Zpracovává a poskytuje informace o všech zařízeních a jejich obsluhu s globálním přístupem. Data jsou uložena ve vektoru typu Facility. Globální přístup je založen na návrhovém vzoru Singleton.
- **Config**
Zajišťuje inicializační načtení a průběžné ukládání uživatelských nastavení do konfiguračního souboru v případě změn. Globální přístup ke všem informacím je zajištěn návrhem Singleton.
- **Reader**
Čtení zpráv od aplikace AdaApp pomocí meziprocesové komunikace (pojmenovaná roura a Mosquito). Běží ve odděleném vláknu a v případě přijetí nové zprávy pošle signál na slot singletonu AllFacilites, který hodnoty naparsuje, uloží a vyšle požadavek na překreslení hodnot, která se zobrazují.
- **CustomShadow**
Rozšíření pro widgety. Transformuje vzhled objektu a přikreslí do jeho okolí stín. Výsledkem je iluze prostorovosti.
- **Moving**
Plocha reagující na prokliky myši s půdorysem obsluhující ikonky s hodnoty - *DragLabel*. Prostor na kterém je možné používat gesto Drag&Drop.
- **DragLabel**
Ikonky zobrazující požadované hodnoty senzorů reagující na události myši. Číselná hodnota je zde transformována na pixmapový obrázek, se kterým je možné pohybovat.
- **MyListRoom**
Panel zobrazující seznam místností v hlavním náhledu. Názvy místností jsou tlačítka fungující jako filtr pro zobrazení zvolené místnosti.
- **LineWiev**
Widget snažící se zobrazit všechna zařízení najednou ve formě řádkových boxů. Zobrazeny jsou zde pouze názvy senzorů a jejich hodnoty. V případě většího počtu začne vytvářet sloupce řádků.
- **Ping**
Testuje stav připojení na server. V budoucnu bude pravděpodobně nahrazeno zprávou od AdaApp.
- **Black**
Prázdný widget, který je zobrazen místo widgetu *MainWiev* v případě nečinnosti.

Definice různých hodnot (konstant) a barev lze najít na jednom místě v knihovně **constants.h**, což umožňuje jednoduché a rychlejší nastavování pro další vývoj na aplikaci. Nevýhodou je znovugenerování všech objektových souborů s touto knihovnou. Pomocné utility jsou v modulu *libfunc*.

4.3 Software na adaptéru

Na adaptéru poběží tři aplikace. Každá má na starost jinou službu, ale pro správnou funkčnost spolu musí meziprocesorově komunikovat. Jedna zajišťuje správu dat a propojení s cloudem, druhá pro aktualizaci adaptéru a řešení oprav systému a třetí pro vizualizaci dat na HDMI (obr. 4.6). První dvě již běží na produkční verzi.

- **Aplikace adaptéru - AdaApp**

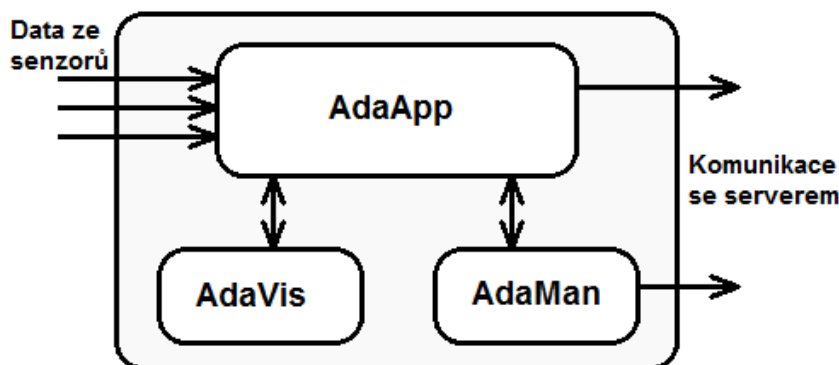
Hlavní aplikace adaptéru, která komunikuje s cloudem. Přeposílá data ze senzorů na server a přijímá informace o aktuátorech.

- **Správa adaptéru - AdaMan**

Aplikace pro aktualizaci adaptéru. Daemon, který komunikuje se speciální aplikací na serveru a dotazuje se na změny.

- **Vizualizace - AdaVis**

Zobrazuje data na výstup HDMI.



Obrázek 4.6: Aplikace běžící na adaptéru.

Část aplikace *AdaApp*, pojmenovaná jako „Distributor“, poskytuje pro aplikaci *AdaVis* aktuální data ze senzorů přes pojmenovanou rouru. Data přichází ve formátu CSV jako `typ;hodnota` a vše je od sebe odděleno středníkem.

Příklad zprávy obsahující informace o aktuálním stavu jednoho senzoru:

```
time;1429378637;eid;11246;state;data;fw_version;2.02;protocol_version;1.0;
dev_version;1;tt_version;1428580014;valid;yes;battery;2000;rssi;81;pairs;1
type;0x01;offset;0;value;62.00
type;0x0A;offset;0;value;21.50
```

Tato zpráva informuje, že senzor naměřil teplotu 21,5°C a vlhkost 62%. Pro zobrazení dat v *AdaVis* aplikaci potřebuje ze zprávy hlavně typy:

- **time**: časová značka aktualizace dat a identifikace další zprávy,
- **euid**: jednoznačné identifikační číslo zařízení, pro přiřazení hodnoty ke konkrétnímu senzoru/aktuátoru,
- **valid**: potvrzení, že jsou data správného formátu,
- **battery**: v budoucnu bude uživatele informovat o stavu baterie senzoru v náhledu detailu na zařízení ve formě „koláčové“ ikonky s procenty,
- **type**: hexadecimální hodnota typu hodnoty dat (0x01, 0x0A),
- **offset**: index hodnoty pro případ, že zařízení nabízí více hodnot stejného typu (např. vnitřní a venkovní teplotní senzor),
- **value**: samotná hodnota definovaná typem **type**.

Každá zpráva je v aplikaci napársována a uložena do dočasné struktury. Pokud je alespoň jedna hodnota validní jsou změny uloženy do globálního vektoru odkud jsou přístupné modulům, které o data požádají. V případě nevalidních dat, či nejasné nebo neúplné zprávy je zpráva zahozena.

Kapitola 5

Testování

Aplikace je cílena pro běžné uživatele, proto bylo nutné grafické rozhraní otestovat. Tato kapitola popisuje způsoby testování výsledné aplikace *Adavis*, jejich průběh a výsledky. Testování bylo provedeno na úrovni uživatelské použitelnosti, online dotazníku a dlouhodobého běhu se simulovaným provozem. Některé znalosti byly získány z přednášky předmětu Testování a dynamická analýza.[13]

5.1 Uživatelské

Testování grafického uživatelského rozhraní proběhlo dvěma způsoby: formou přímého ovládní a formou vyplnění online dotazníku. V případě přímého ovládní bylo provedeno jednoduché **testování použitelnosti** (anglicky *Usability testing*). V rámci možností a času byly použity některé metody testování použitelnosti z přednášky Usability Testing na Michigan Technological University.[12]

- poskytnutí minima informací,
- zadávání jednoduchých úkonů,
- sledování činnosti uživatele,
- sledování očí, myši, prstů,
- ústní dotazník po aktivitě.

Aplikaci odzkoušelo 5 osob bez znalosti systémů inteligentní domácnosti, různého vzdělání, různé věkové generace. Každá osoba dostala 30 sekund na rychlé seznámení s prostředím, kde bylo k dispozici 10 virtuálních senzorů. Pak byly zadávány jednoduché úkoly:

- zjištění teploty ze senzoru, který je viditelný ve výchozím náhledu,
- zjištění vlhkosti ze senzoru, který je viditelný až po zarolování dolů,
- zjištění v jakém stavu se nachází aktuátor,
- zobrazení všech zařízení v jedné místnosti,
- přepnutí na náhled s půdorysem domu,
- rozmístění ikon na půdorysu podle jejich názvů,

- změna nastavení obrázku půdorysu se znovupřizpůsobením ikon,
- nahrání a nastavení vlastního obrázku jako půdorys.

Během těchto úkonů byl uživatel sledován dvěma pozorovateli, proto nebyla potřeba kamerového záznamu. Jelikož je aplikace primárně informačního charakteru a nemá mnoho náhledů, výsledky dopadly dle očekávání kladně. Uživatelé během prvních 30 sekund pochopili ovládání a následující úkoly plnili během krátké chvíle.

V ústním dotazníku se zmínili, že se jim velice zalíbilo Drag&Drop s ikonky na půdorysu, ale rádi by na ikonkách měli možnost náhledu vícero hodnot současně. Vytýkáno bylo, že chybí grafická odezva na klik (zda bylo tlačítko stisknuto či nikoliv). Mužům pak byla trochu nepříjemná růžová barva hodnot.

Sledováním myši bylo zjištěno, že si uživatelé ve „volné chvíli“ rádi bezcílně pohybují myší nad objekty, které se nějakým způsobem mění (například pozadí tlačítka). Tímto grafickým oživením se může uživateli aplikace více zalíbit a tím ji bude chtít využívat i opakovaně.

Další zpětná vazba od uživatelů byla získána pomocí **online dotazníku** vytvořeném ve formuláři od Google.com¹, který byl šířen přes sociální síť. Dotazník vyplnilo 27 osob ve věku od 16 do 42 let. Respondenti své technické dovednosti v oblasti IT odhadli ve 22% na základní úrovni, 30% jako pokročilí a ve 48% jako experti. Začátek odpovědního formuláře obsahoval snímky obrazovky s aplikací a video ukazující základní funkcionalitu. Dále následovali otázky s hodnocením od 1 (výborné) až 5 (nedostačující) a otázky se slovní odpovědí.

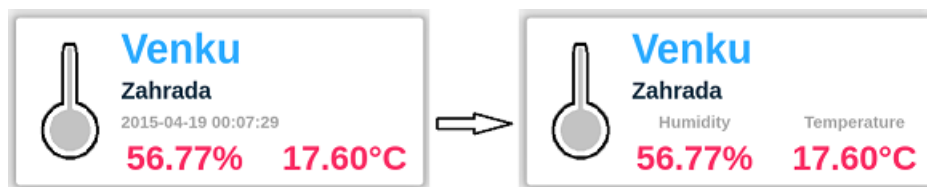
Otázky a jejich průměrné oznamkování:

- Celkový dojem z aplikace: 1,78
- Ohodnocení barevného schéma: 1,85
- Rozložení prvků na obrazovce: 1,70
- Přehlednost aplikace: 1,88
- Intuitivnost ovládání: 2,00

V otázce velikosti tlačítek pravého panelu (obr. 4.2) odpovědělo 15% respondentů, že jsou tlačítka zbytečně velká. Ostatním se zdála optimální. Jejich rozměry ale stejně budou v budoucnu redukována podle počtu rozšíření funkcí a náhledů aplikace.

V nepovinné otázce „*Co byste vylepšili na vzhledu?*“ hodnotící přispěli zajímavými návrhy na změny. Jedním z požadavků bylo vytvoření názornějších ikoněk na tlačítkách, s čímž souhlasím, jelikož aktuální obrázky byly vytvořeny především pro demonstraci. Mnoha lidem přišla zbytečná informace o poslední změně hodnot a některým nebylo hned jasné, co které hodnoty znamenají. Proto bylo datum s časem zaměněno za popisky hodnot (obr. 5.1).

¹https://www.google.com/intl/cs_CZ/forms/about/



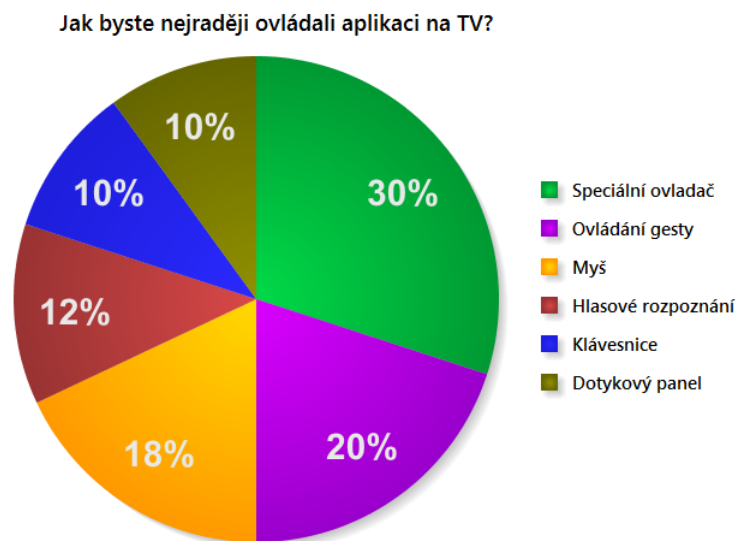
Obrázek 5.1: Přidání popisku pro hodnoty.

Špatně byla vnímána zelená barva nápisu ONLINE na modrém podkladu. Barva byla změněna na bílou, jako ostatní nadpisy. Užitečná poznámka byla k nepoměru ovládacích prvků na levé a pravé straně. Tento fakt bude zohledněn v dalších úpravách. Zajímavý nápad byl provádět změnu barvy textu hodnoty podle hodnoty. Zde je ale problém s definováním zlomových hodnot pro změnu barvy. Jedna z odpovědí obsahovala i dotaz, zdali je k zobrazení nějaký graf. Tato možnost byla navržena, ale v době psaní této práce ještě nebylo vyřešeno získávání starých dat ze serveru.

Na otázku „*Jaké informace byste chtěli primárně zobrazovat?*“ odpovídali respondenti dosti neurčitě. Vesměs by chtěli přehled všeho, což již nabízí výchozí náhled. Jeden z návrhů byl přidat možnost volby do nastavení. Zajímavá myšlenka byla seřadit zařízení podle neobvyklých stavů. Například, když je aktuátor zapnutý v čas, kdy nemá být, nebo když se nějaká hodnota vychyluje od běžných hodnot z minulosti.

V otázce „*Jak byste nejraději ovládali aplikaci na TV?*“ byly předchystany odpovědi na obvyklé ovladače a byla možnost zvolit z více možností a nebo přidat vlastní. Jako nejžádanější se ukázal speciální ovladač, který by mohl mít plochý joystick a pár rychlých tlačítek.

Výsledky jsou vykresleny ve formě grafu na obrázku 5.2.



Obrázek 5.2: Graf s žádaným ovládáním aplikace na TV.

V možnosti další libovolných připomínek bylo navrženo přidat více jazykových lokalizací, nastavení vlastní výchozí plochy, nastavení automatizovaného spínání a regulace, upravit tlačítka, atd. Některé tyto body budou aplikovány, ostatní obstarává (nebo bude) pouze aplikace pro mobilní zařízení.

Závěrem byla otázka, zda by využívali možnosti sledování domácnosti na TV. 75% respondentů odpovědělo kladně, někteří pod podmínkou úprav a 25% osob by používalo pouze mobilní aplikaci, jelikož nevlastní TV nebo o inteligentní domácnost nejeví zájem.

Ukázkové video a všechny odpovědi z dotazníku jsou k dispozici na příloženém CD.

5.2 Dlouhodobý běh

Pro testování stability aplikace jsem použil automatizovaný nástroj *Auto Mouse Click by MurGuu*². V tomto nástroji lze napsat scénář jak a kdy se má myš pohybovat a klikat. Scénář jsem se pokusil napsat s co nejvíce možnostmi, např.: opakované klikání jednoho tlačítka, přepínání mezi dvěma náhledy, klikání na hrany tlačítek i obrazovky, aj. Současně s tím musela aplikace zpracovávat každé 2 sekundy nová data, generovaná náhodně mým skriptem, od 10 virtuálních senzorů.

Testování probíhalo na Xubuntu 14.04 na virtualizovaném stroji s rozlišením 1920x1080 pixelů. První běh ukázal, že aplikace nestíhá přepínání náhledů s rozumnou odezvou na klik. Tento problém byl hned vyřešen snížením režie na vykreslování. Ve druhém běhu se již neprojevily žádné závažné problémy.

Během testování nenastal žádný neočekávaný stav, který by způsobil pád aplikace, ani v případě klikání bez senzorů. Asi jediným nedostatkem bylo vyjimečné krátkodobé zamrznutí.

5.3 Zátěž na CPU

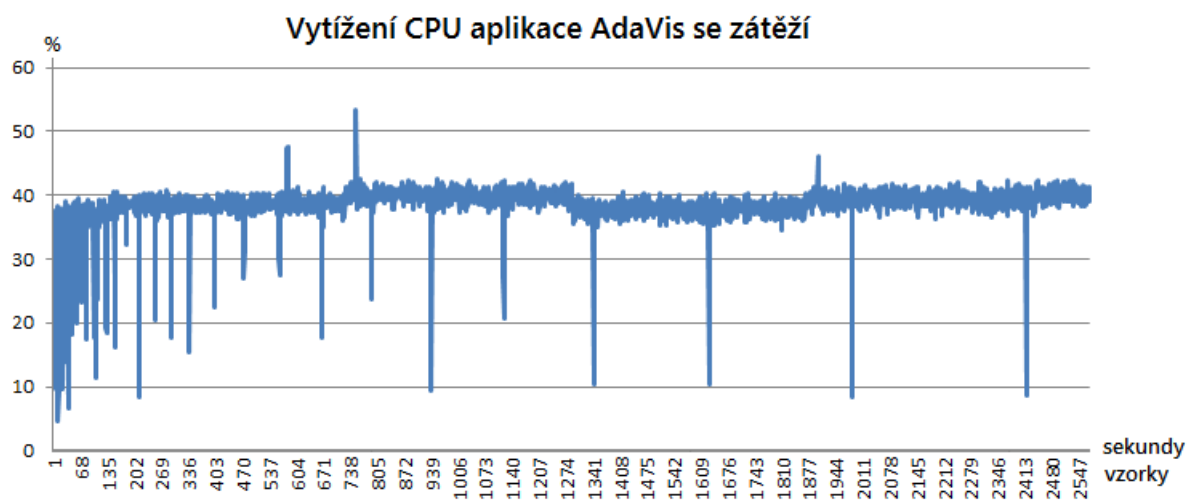
Důležitým aspektem je také informace, jak moc aplikace *AdaVis* zatěžuje embedded Linux. Náročná aplikace může způsobovat přehřívání procesoru, blokovat jiné důležitější aplikace a obecně může snižovat spolehlivost, což je u vestavěných zařízení nežádoucí stav. Pro toto měření jsem nenašel lepší nástroj než nástroj **top**, který je ovšem naprosto postačující. Pro záznam hodnoty byl použit příkaz:

```
top -d 1 | grep AdaVis > /tmp/top.log
```

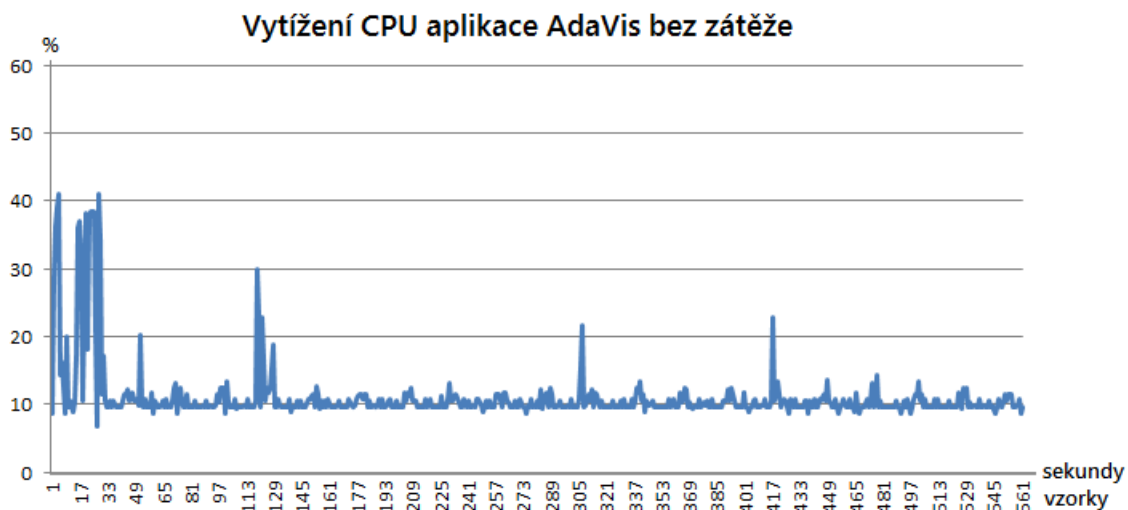
Hodnoty v procentech byly zaznamenávány každou sekundu. První běh byl spuštěn s pravidelnou aktualizací všech 10 virtuálních senzorů současně každé 2 sekundy. Měření běželo více jak 40 min a výsledkem byla **průměrná zátěž 38,51%**. Průběh měření je k dispozici na grafu 5.3.

Druhé měření bylo provedeno bez aktualizace senzorů. Aplikace tedy nemusela zpracovávat žádné vstupní data, ani překreslovat grafické části. Přesto výsledkem měření po době 10 minut byla **průměrná zátěž 11,10%**. Průběh měření je k nahlédnutí na grafu 5.4.

²<http://www.murguu.com/>



Obrázek 5.3: Vytížení CPU aplikací AdaVis se zpracováváním vstupních dat.



Obrázek 5.4: Vytížení CPU aplikací AdaVis bez zpracování vstupních dat.

Kapitola 6

Závěr

Cílem této bakalářské práce bylo navrhnout a implementovat grafické rozhraní na embedded Linux s výstupem na HDMI pro inteligentní domácnost. Návrh byl implementován a úspěšně otestován ve virtuálním prostředí. Výsledná aplikace splňuje základní funkcionalitu, tj. správně zobrazuje aktuální data ze senzorů a aktuátorů.

Aplikaci *Adavis* se podařilo zprovoznit na minipočítači s embedded Linux společně s aplikací *AdaApp*, která posílala zprávy z virtuálních senzorů. Zprovoznění na adaptéru ukázalo, že grafické zobrazení na monitoru připojeným přes HDMI se oproti zobrazení ve vývojovému prostředí liší stylem písma, které je méně „líbivé“. Tento nedostatek nebyl nějak řešen, ale pravděpodobně by stačilo najít a použít Qt knihovnu zabývající se tímto vykreslováním nebo najít jiný styl písma. Implementace detailního náhledu na informace o senzoru byla v době psaní této práce pozastavena, jelikož nebyl dořešen způsob komunikace se serverem pro získání těchto dat. Samotná grafická aplikace totiž nemůže z důvodu zabezpečení komunikace kontaktovat přímo server. Proto bude využita některá z aplikací, které se serverem komunikují zabezpečeným kanálem.

6.1 Návrhy na rozšíření

V mém plánu je dále pokračovat a vylepšovat aplikaci až na úroveň produkční verze. Kromě ladění, optimalizace a přizpůsobování se měnám během vývoje celého systému *BeeOn* se dá aplikace ještě více obohatit o další elementy.

Hlavním bodem je zprovoznit zpětnou komunikaci od aplikace *AdaVis* k *AdaApp* pro poslání žádosti o přepnutí aktuátoru. Samotný adaptér se tím pak stane „koncovým zařízením“ pro uživatelské ovládání. Touto cestou by bylo možné poslat i žádost na získání starých hodnot vybraného senzoru pro vykreslení grafu v detailním náhledu.

V nastavení by určitě měla přibýt možnost nastavení jazykové lokalizace, minimálně čeština a angličtina. Z dotazníku vyplynulo, že některým lidem se nezamlouvalo barevné schéma, což je dosti subjektivní problém, který by ale mohl být vyřešen přidáním výběru z několika barevných stylů. Užitečnou částí by bylo zobrazení předpovědi počasí, na základě něhož by si mohl uživatel přednastavit teploty. Myšlenou na závěr je autonomní zobrazování v případě, že nebude detekováno žádné ovládací zařízení. Ovládací panely by se skryly a zobrazil by se pouze náhled všech zařízení, nebo v případě velkého množství pouze některé s pomalým rolováním seznamu.

Literatura

- [1] AMX by Harman: About AMX by Harman [online]. 2015 [cit. 2014-12-21].
Dostupné z: <http://www.amx.com/aboutamx/factsheet.asp>
- [2] Eaton Elektrotechnika s.r.o.: O systému [online]. 2015 [cit. 2015-04-29].
Dostupné z: <http://www.xcomfort.cz/o-systemu/>
- [3] eLinux.org: CEC (Consumer Electronics Control) over HDMI. 2013 [cit. 2015-04-26].
Dostupné z:
[http://elinux.org/CEC_\(Consumer_Electronics_Control\)_over_HDMI](http://elinux.org/CEC_(Consumer_Electronics_Control)_over_HDMI)
- [4] ELKO EP, s.r.o.: O nás [online]. 2014 [cit. 2014-12-21].
Dostupné z: <http://www.inels.cz/cs/o-nas>
- [5] Freedesktop.org: D-Bus Tutorial [online]. [cit. 2015-01-15].
Dostupné z: <http://dbus.freedesktop.org/doc/dbus-tutorial.html>
- [6] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: *Návrh programů pomocí vzorů, Stavební kameny objektivě orientovaných programů*. Praha: Grada, 2003, ISBN 80-247-0302-5.
- [7] HDMI.org: Learning center - Knowledge Base [online]. [cit. 2014-12-14].
Dostupné z: <http://www.hdmi.org/learningcenter/kb.aspx>
- [8] Korček, P.: Smarthome architecture [online]. 2013 [cit. 2014-12-21].
Dostupné z:
https://merlin.fit.vutbr.cz/wiki-iot/index.php/Smarthome_architecture
- [9] Loxone.com: O Společnosti [online]. [cit. 2014-12-21].
Dostupné z: <http://www.loxone.com/cscz/o-spolecnosti/o-nas.html>
- [10] Marshall, D.: UNIX System Calls and Subroutines using C, The Cardiff School Of Computer Science and Informatics. 1999 [cit. 2015-02-02].
Dostupné z: <http://www.cs.cf.ac.uk/Dave/C>
- [11] Mqtt community: Mosquitto - definition [online]. 2015, [cit. 2015-05-02].
Dostupné z: <http://mosquitto.org/>
- [12] Pastel, R.: CS4760: Human-Computer Interactions - Usability Testing [online], Michigan Technological University. 2014, [cit. 2015-05-09].
Dostupné z: <http://cs4760.csl.mtu.edu/2014/lectures/usability-testing/>
- [13] Smrčka, A.: Testování aplikací využívající síť, testování grafického uživatelského rozhraní, Fakulta informačních technologií, VUT v Brně. 2014 [cit. 2015-05-07].

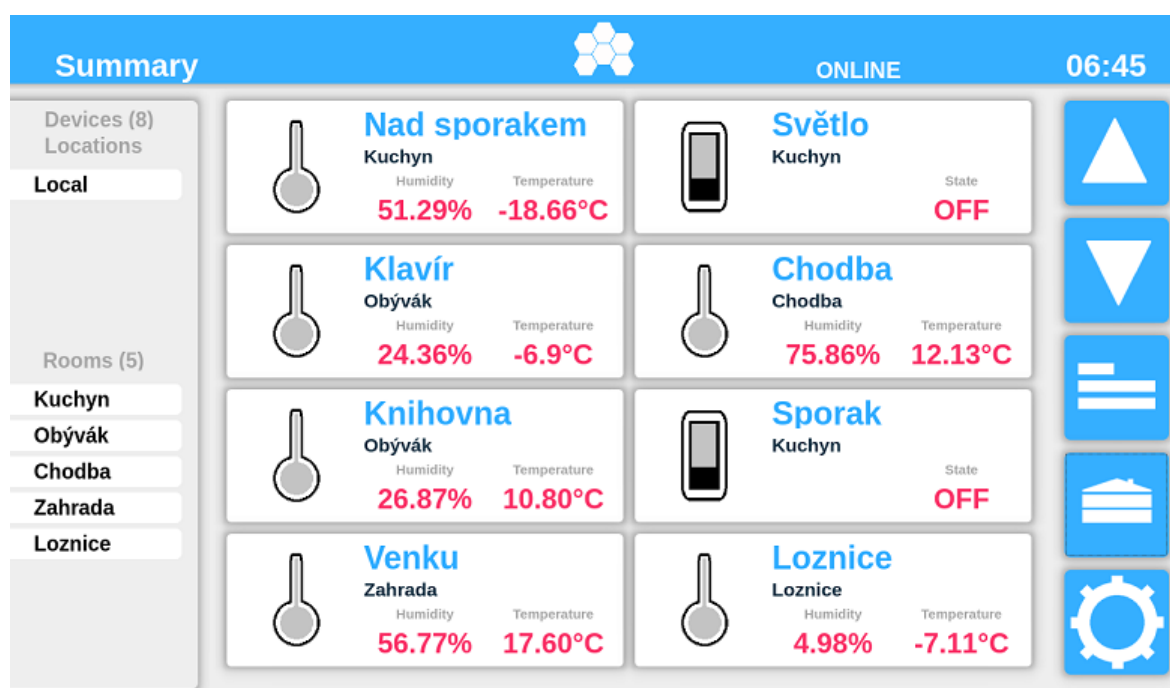
Dodatek A

Obsah CD

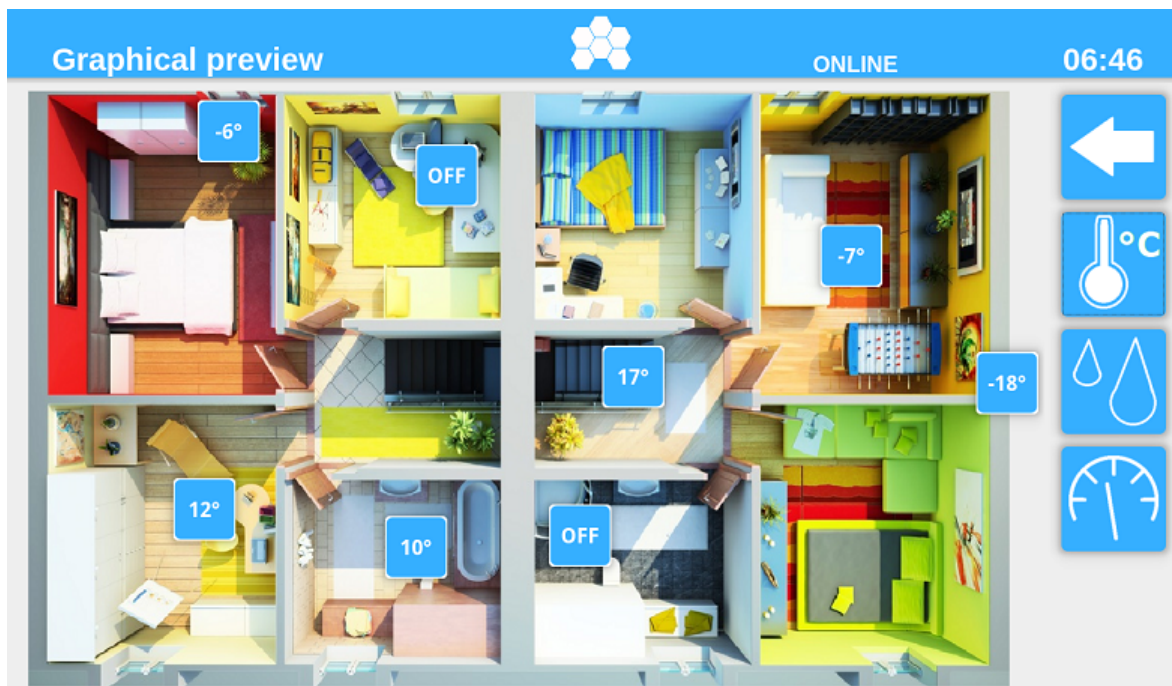
- `README`
Manuál pro instalaci a spuštění.
- `src/`
Adresář obsahující zdrojové kódy včetně souboru Makefile (viz. README)
- `doc/`
Automaticky vygenerovaná dokumentace nástrojem Doxygen.
- `projekt.pdf`
Tato bakalářská práce ve formátu PDF.
- `latex/`
Zdrojové kódy tohoto dokumentu ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.
- `dotaznik/`
Formulář v HTML a odpovědi v XLS.
- `media/`
Videoprezentace a screenshoty výsledné aplikace.

Dodatek B

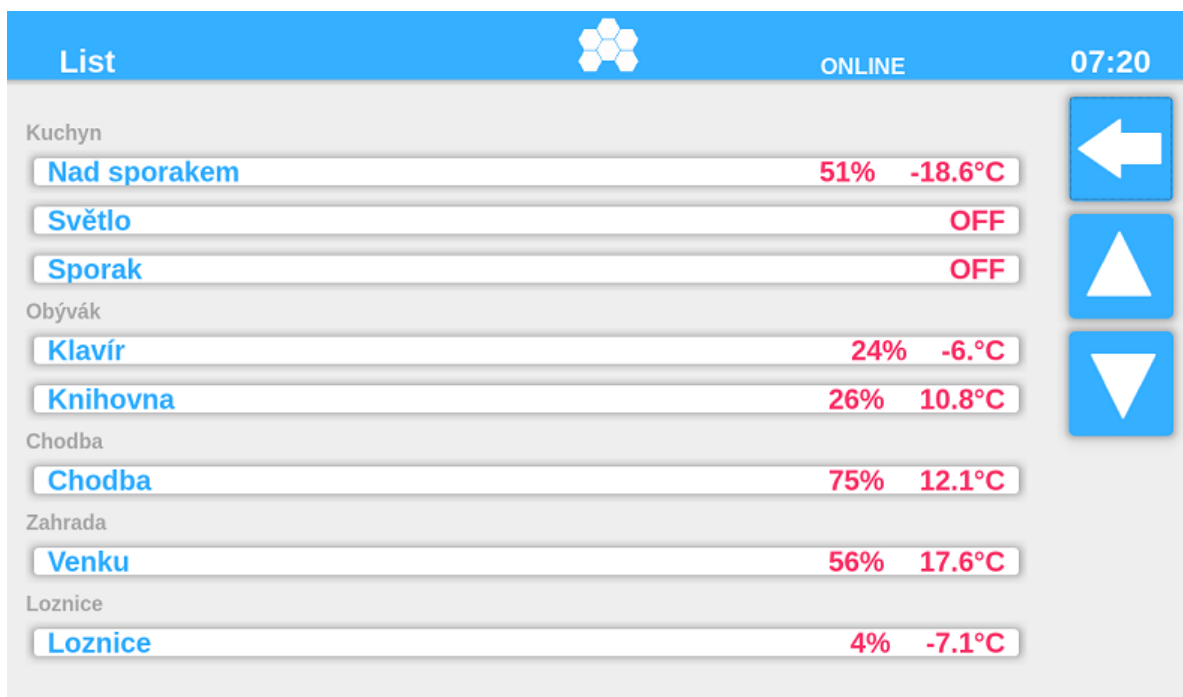
Ukázky UI



Obrázek B.1: Hlavní náhled aplikace AdaVis.



Obrázek B.2: Náhled na půdorys¹bytu/domu v aplikaci AdaVis s možností Drag&Drop.



Obrázek B.3: Řádkový náhled v aplikaci AdaVis.

¹Obrázek na pozadí náhledu převzat z webu: <http://www.domytvarozna.cz/domy.html>