

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

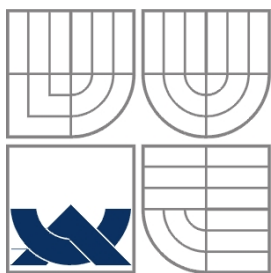
APLIKACE PRO ANDROID SMARTPHONE A TABLET
NA OVLÁDÁNÍ INELS RF CONTROL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

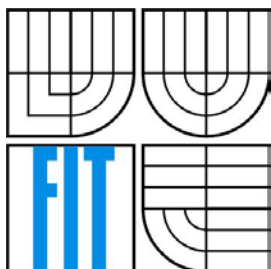
AUTOR PRÁCE
AUTHOR

Bc. JAKUB ŠREIBR

BRNO 201



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

APLIKACE PRO ANDROID SMARTPHONE A TABLET NA OVLÁDÁNÍ INELS RF CONTROL

APPLICATION FOR ANDROID SMARTPHONE AND TABLET FOR CONTROL OF THE INELS RF
UNIT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAKUB ŠREIBR

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing., Dipl.-Ing.,
MARTIN DRAHANSKÝ, Ph.D.

BRNO 2013

Abstrakt

První část práce pojednává o problematice inteligentních elektroinstalací. Popisuje jejich typy a výhody a nevýhody jednotlivých typů. Detailně se zabývá bezdrátovým systémem iNELS RF Control firmy ELKO EP s.r.o. Popisuje princip komunikace a vybrané prvky tohoto systému. Pojednává také o platformě Android a jejích možnostech. Další část tvoří analýza požadavků a návrh aplikace pro ovládání systému iNELS RF Control. V návrhu je uvedena použitá architektura, popis komunikace, model databáze a modelování logiky aplikace. Čtvrtá část práce popisuje implementaci aplikace včetně použitých technologií, vývojových nástrojů a systémových komponent. Obsahuje popis částí výsledné aplikace. Pátá část se zabývá testováním aplikace. Obsahuje použité metody, nástroje a techniky. V závěru práce je uvedeno zhodnocení práce a dosažených cílů.

Abstract

The first part deals with the issue of intelligent electrical installations. It describes the types and their advantages and disadvantages. It deals in detail with the wireless system iNELS RF Control, which was developed by ELKO EP s.r.o. It describes the principles of communication and selected elements of the system. It also discusses the Android platform and its capabilities. Another part consists of requirements analysis and application design for control of the system iNELS RF Control. Application design consists of used architecture model, description of communication, database model and application logic's model. The fourth part describes the implementation of applications, including the technology, development tools and system components. It contains a description of the final application parts. The fifth section deals with testing of the application. It contains methods, tools and techniques. The conclusion contains appreciation of the work and achievements.

Klíčová slova

Inteligentní dům, inteligentní elektroinstalace, ELKO EP, s.r.o., iNELS RF Control, Android aplikace, mobilní aplikace, smartphone, tablet, eLAN-RF

Keywords

Intelligent house, intelligent electrical installation, ELKO EP, s.r.o., iNELS RF Control, Android application, mobile application, smartphone, tablet, eLAN-RF

Citace

Šreibr Jakub: Aplikace pro Android smartphone a tablet na ovládání iNELS RF Control, diplomová práce, Brno, FIT VUT v Brně, 2013

Aplikace pro Android smartphone a tablet na ovládání iNELS RF Control

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením doc. Ing., Dipl.-Ing., Martina Drahanského, Ph.D. Další informace mi poskytli zaměstnanci firmy ELKO EP s.r.o. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jakub Šreibr
29.5.2013

Poděkování

Chtěl bych poděkovat vedoucímu práce doc. Ing., Dipl.-Ing., Martinu Drahanskému, Ph.D., za poskytování odborných rad. Dále pak zaměstnancům a vedení firmy ELKO EP s.r.o., zvláště pak Jaromíru Příkladovi, za poskytnutí potřebných materiálů a informací.

© Jakub Šreibr, 2013

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
1.1	Inteligentní elektroinstalace.....	3
2	Popis problematiky	6
2.1	ELKO EP, s.r.o.	6
2.2	iNELS RF Control	7
2.3	Smartphony a tablety	16
2.4	Platforma Android	17
3	Analýza a návrh aplikace	23
3.1	Přehled požadavků.....	23
3.2	Architektura	24
3.3	Komunikace.....	25
3.4	Data.....	27
3.5	GUI	28
3.6	Logika aplikace.....	29
4	Implementace	33
4.1	Vývoj aplikace pro Android	33
4.2	Výsledné řešení.....	37
5	Testování.....	43
5.1	ACRA	43
5.2	Metoda white-box	43
5.3	Testování ochranného intervalu.....	43
6	Závěr	45

1 Úvod

Automatizace a vestavěné systémy jsou dnes již běžnou součástí lidského života. Není tedy divu, že tento trend neminul ani naše domácnosti. Prakticky každý dům v dnešní době obsahuje nějaké inteligentní zařízení (např. automatickou pračku, PC, lednici, mikrovlnnou troubu atd.). Z důvodu zvyšování pohodlí a potřeb obyvatel je míra integrace inteligentních zařízení stále vyšší.

Zájem o zvyšování funkcionality domácností se datuje do šedesátých let, kdy se jednalo spíše o záležitost domácích kutilů. V roce 1984 se však tento zájem přenesl i do komerční sféry a americká asociace NAHB (National Association of Home Builders) založila skupinu „*Smart House*“, která měla na starosti zahrnutí potřebných technologií do návrhů nových budov. Od osmdesátých let se začala výroba elektroniky a elektrického vybavení zaměřovat na vývoj digitálních systémů a komponent namísto analogových. Z hlediska vybavení budov byl důležitý zejména přechod z elektromechanického spínání na digitální, nahrazení koaxiálních a kroucených dvojlinek optickými vlákny a vývoj nových komunikačních sítí (internet, ISDN) a koncových zařízení (video telefony). V devadesátých letech přestal být koncept tzv. chytrého domu pouze záležitostí několika nadšenců a vstoupil do podvědomí celé společnosti. Televize BBC odvysílala sérii dokumentů nazvanou *Dream House*, ve které šlo o pozorování rodiny žijící v experimentálním chytrém domě po dobu šesti měsíců. Diváci tak získali představu o tom, co vlastně tento pojem znamená. Stále však nebylo jisté, jestli by lidé díky své opatrnosti přivítali takové technologie i ve svých domácnostech [1].

Za posledních deset let se velmi změnil způsob života a s ním i trendy vývoje našich domácností. Společnost stárne a v roce 2020 bude pětina Evropské populace starší než 65 let. Zvyšuje se tedy poptávka po domácích zdravotních zařízeních, aby mohli obyvatelé zůstat nezávislí na pomoci ostatních a samostatně žít ve svých domech, jak nejdéle to jen půjde [2]. Velmi příznivým faktem pro vývoj chytrých domů je také zvyšující se poptávka po různých smartphonech, tabletech apod. Tato přenosná zařízení disponující dotykovou obrazovkou jsou totiž ideální pro centrální ovládání celé domácnosti. V roce 2015 by měl průměrný počet obrazovek vzrůst až na 3,6 na osobu, což je oproti roku 2000, kdy tento počet dosahoval pouze 1,7, opravdu velký skok. V současné době mnoho firem pracuje na vývoji nových aplikací, operačních systémů, telekomunikačních sítí a různých jiných technologií, které ženu vývoj inteligentních domácností kupředu [3].

Původně pojem chytrý dům apod. představoval budovu s nějakými automatizovanými systémy, jakými jsou například osvětlení a vytápění [4]. V dnešní době značí budovu vybavenou počítačovou komunikační technikou, která reaguje na potřeby obyvatel s cílem zvýšit jejich komfort, pohodlí, snížit spotřebu energií, poskytnout jim bezpečí a zábavu pomocí řízení všech technologií v domě a jejich interakcí s vnějším světem [5]. Zahrnuje tedy i systémy pro zábavu, zabezpečení, sdílení dat a zdravotní péči.

Z pohledu uživatele je velmi důležité mít možnost tyto systémy monitorovat a ovládat kdekoli na světě. K tomuto účelu se nejlépe hodí různá mobilní zařízení (PDA, smartphony, tablety apod.). Náplní této práce je vývoj aplikace sloužící pro správu inteligentní elektroinstalace pro mobilní zařízení běžící na platformě Android. Kompletní problematika je shrnuta ve druhé kapitole, následující část potom obsahuje návrh celé aplikace. Třetí kapitola je věnována výsledné implementaci. Poslední část je zaměřena na testování. V závěru této práce jsou shrnuty dosažené výsledky a možnosti vylepšení a rozšíření aplikace.

1.1 Inteligentní elektroinstalace

Velmi podstatnou součástí chytrého domu je systém inteligentní elektroinstalace. Ta umožňuje správu celého domu od ovládání světel, rolet, spotřebičů, regulace topení a klimatizace až po zabezpečení [6].

Klasická elektroinstalace poskytuje pouze omezené možnosti správy. Ovladače jsou napevno spojeny s konkrétními zařízeními a jejich funkci nelze po instalaci změnit, aniž by to nevyžadovalo položení nové kabeláže. Takovéto okruhy tvoří vlastně samostatné celky, které mezi sebou nijak nekomunikují. Téměř chybí možnost nějaké centralizované správy a monitoringu elektroinstalace z jednoho místa.

V inteligentní elektroinstalaci lze nakonfigurovat, který ovladač bude ovládat konkrétní zařízení bez nutnosti zásahů do její fyzické struktury. Zařízení může být ovládáno z několika míst přesně podle potřeby uživatele. Pokud se ovládání stane nevyhovujícím, lze ho jednoduše změnit přeprogramováním elektroinstalace. Ovládací prvky jsou oproti klasickým vypínačům daleko komfortnější. Využívají se panely s tlačítky, která nezabírají tolik místa, různé dálkové ovladače a dotykové panely. Na stisk jednoho tlačítka se dá nastavit hned několik akcí, což umožňuje rychlé nastavení optimální konfigurace podle potřeby.

Další velká výhoda spočívá v možnosti správy a monitorování celého systému z jednoho místa (např. z PC, tabletu nebo chytrého telefonu). Tato zařízení poskytují i možnost vzdáleného ovládání domu prostřednictvím internetu, takže uživatel se může kdekoliv na světě připojit ke svému domovnímu systému a nastavit potřebné parametry.

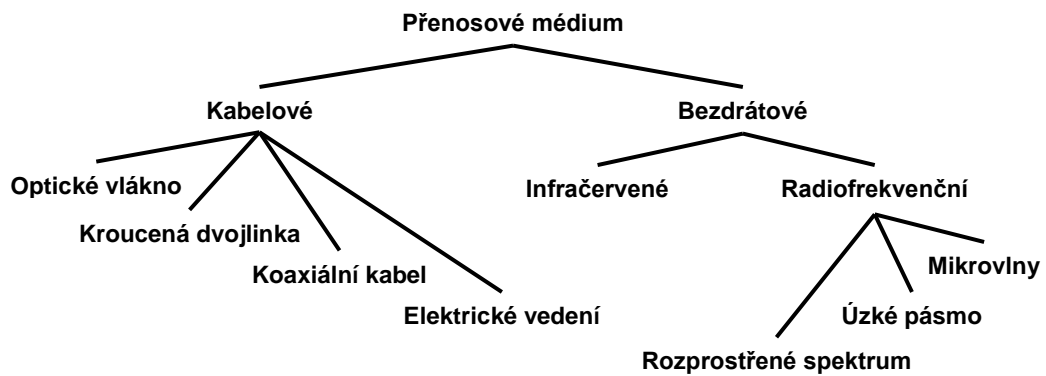
Výše uvedené výhody jsou ty nejpodstatnější, ale jejich výčet tím nekončí. Inteligentní instalace umožňuje propojování s jinými systémy, nastavování akcí podle času, dopředné plánování činností nebo třeba simulaci přítomnosti osob v prázdném domě [5].

Základními jednotkami každé inteligentní elektroinstalace jsou:

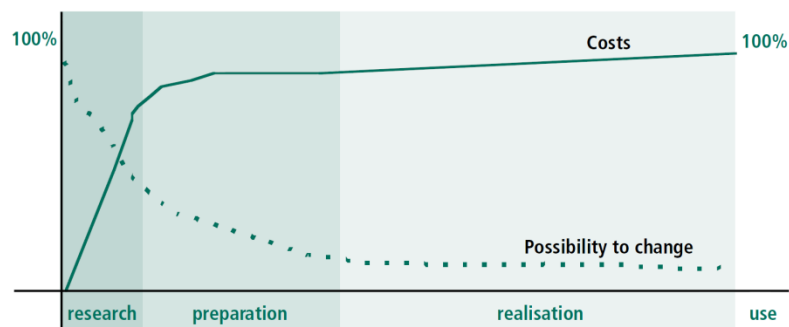
- **Vysílače** - Různé ovládací prvky nebo senzory (vypínače, dálkové ovladače, detektory pohybu, teplotní čidla apod.), které reagují na podněty z okolí (stisk, změnu teploty), a na základě nich zasílají signály aktorům.
- **Aktory** – Akční členy, které přijímají signály od vysílačů a na jejich základě provádějí předdefinované úkoly (rozsvítí světlo, vypnou televizi, spustí alarm, ztlumí topení apod.). Mezi základní typy aktorů patří spínací, stmívací, roletové a teplotní.

1.1.1 Typy inteligentních elektroinstalací

Inteligentní elektroinstalace se podle přenosového média rozlišují na dva základní typy: sběrníkovou a bezdrátovou (obrázek 1.1). Sběrníkové elektroinstalace, jak již z názvu vyplývá, využívají pro přenos informací sběrnici, což je speciální nízkonapěťový kabel, přes který se přenáší signál prostřednictvím určitého protokolu. Bývají většinou implementovány do nově postavených budov a musí se zahrnout do již existujícího stavebního procesu. Toto je práce pro speciální konzultanty, kteří jsou za implementaci inteligentních technologií zodpovědní. Je velmi důležité správně navrhnout celý systém podle požadavků už na samém začátku stavebního procesu, protože v jeho průběhu možnost změn klesá a naopak stoupá cena výsledného řešení, což je patrné z grafu na obrázku 1.2 [7].



Obrázek 1.1 Rozdělení přenosových médií [7].



Obrázek 1.2 Cena a možnost změny inteligentní elektroinstalace v jednotlivých fázích stavebního procesu [7].

Typy sběrníkové elektroinstalace se instalují především do nových budov, jak již bylo řečeno. Jejich výhodou je rychlost, spolehlivost a bezpečnost přenosu. Velkým problémem je ovšem integrace do již postavených domů nebo změna infrastruktury v pozdějších fázích stavebního procesu. Tyto překážky eliminuje systém bezdrátové elektroinstalace, který je daleko flexibilnější, a pro jeho instalaci nejsou potřeba žádné zásadní úpravy domu. Dříve se pro bezdrátový přenos používaly protokoly a sítě speciálně navržené pro tento účel. V dnešní době se využívají i jiné přenosové technologie, které nejsou primárně určeny pro tento účel. Můžeme je rozdělit do tří základních tříd:

- **Tradiční RF sítě pro chytré domy** – Jedná se o sítě s nízkofrekvenčním přenosem a pouze základními modulačními technikami, které poskytují většinou velmi malou šířku pásma (pouze pár kbit/s nebo méně). Příkladem protokolů patřících do této skupiny, které využívají radiofrekvenční přenos, jsou X10 (běžící na 200MHz) a KNX (běžící na 868MHz).
- **Širokopásmové RF protokoly** – Protokoly, které byly primárně navrženy pro počítačové sítě. Poskytují velkou šířku pásma (stovky Mbits/s) a většinou používají frekvence 2,4 GHz a 5 GHz. Nejběžnějším standardem je IEEE 802.11 a/b/g.
- **Sítě s nízkým výkonem** – Tyto sítě byly navrženy pro mobilní zařízení, kde byl důležitý nízký odběr energie (Bluetooth). Pro bezdrátové elektroinstalace je velmi vhodná síť Zigbee [7].

Pro některé podsystémy inteligentní elektroinstalace je však bezdrátový přenos nevhodný. Jedná se především o přenos videa a ostatní datové přenosy, kde je vyžadována velká šířka pásma [2]. Nevýhodami bezdrátové elektroinstalace jsou:

- **Spolehlivost** – Bezdrátový přenos má větší chybovost při komunikaci z důvodu okolního rušení, chyb softwaru a vyšší složitosti komunikace. Komunikace po sběrnici je daleko spolehlivější, protože se jedná o přímé propojení pomocí kabelu.
- **Rychlost** – Bezdrátová komunikace je dnes nejméně desetkrát až dvacetkrát pomalejší než přenos po kabelu. Důvodem je opět vyšší režie při komunikaci a nutnost vyrovnávání se se ztrátami paketů. Pro ovládání klasických zařízení jako jsou světla, rolety a topení to není až takový problém, ale pokud je potřeba přenášet například video z bezpečnostních kamer, je v tomto ohledu sběrnice systém nenahraditelný.
- **Bezpečnost** – Ze sběrnice zabudované ve zdi je únik dat prakticky nemožný na rozdíl od bezdrátových sítí, kde se musí řešit šifrování přenosu, což přináší do celé komunikace další režii a zpomalení. Navíc po prolomení šifrování jsou informace útočnickovi přístupné kdekoli v dosahu signálu.

Bezdrátovou elektroinstalaci je tedy dobré používat pouze v již postavených domech, kde by zabudování sběrnice systému vyšlo draž, nebo jen jako doplněk klasické kabeláže. V případě, že dům ještě není postavený, je zatím stále lepší nechat do něj při stavbě zabudovat sběrnice elektroinstalaci. Vývoj bezdrátových technologií jde ale stále kupředu a je tedy možné, že v budoucnu bude situace jiná [5].

2 Popis problematiky

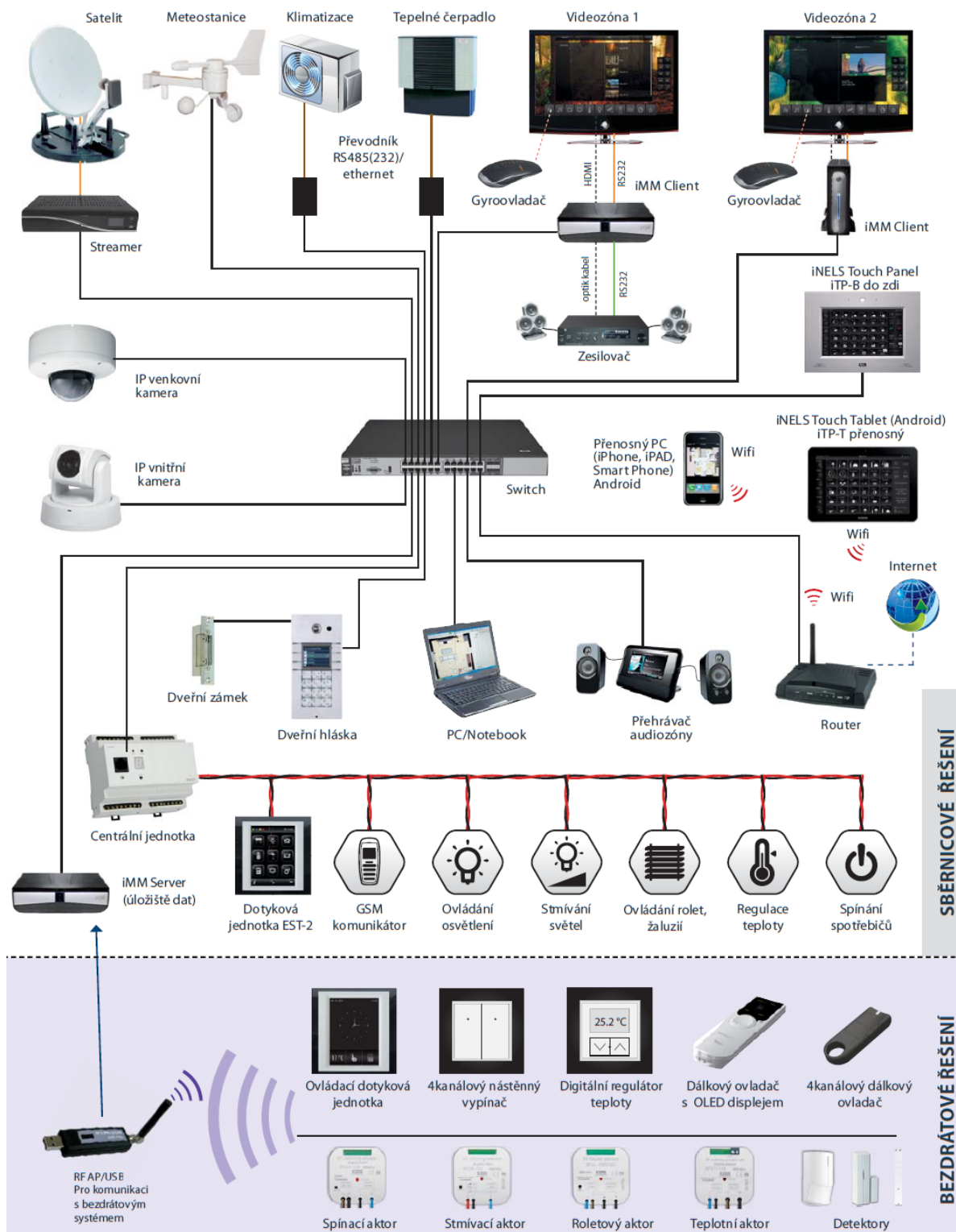
Cílem práce je vytvořit aplikaci pro chytrý telefon a tablet, prostřednictvím které bude možné monitorovat a ovládat systém bezdrátové elektroinstalace *iNELS RF Control* od firmy *ELKO EP, s.r.o.* [14]. Tato klientská aplikace bude komunikovat s jednotkou *eLAN-RF* [10], která bude pracovat jako server. *eLAN-RF* slouží jako převodník povelů z klasické LAN TCP/IP sítě na RF signál, který zasílá bezdrátovým jednotkám v systému. Díky tomu bude možné na smartphonu nebo tabletu vyčítat stavy jednotlivých zařízení a tyto stavy měnit. Režim scén bude umožňovat pomocí stisku jednoho tlačítka nastavit více vybraných zařízení do předem zvolených stavů (např. při scéně „film“ se ztlumí osvětlení a zatáhnou žaluzie).

2.1 ELKO EP, s.r.o.

Firma *ELKO EP, s.r.o.* se již sedm let zabývá vývojem chytrých domů a je v tomto ohledu v ČR zřejmě nejdál. V roce 2005 uvedla na trh svůj první systém inteligentní elektroinstalace s označením *iNELS* [10], který od té doby prošel značným rozvojem a dočkal se několika rozšíření. Dnes je jeho plné označení *iNELS SMART HOME SOLUTIONS* [10] (obrázek 2.1) a slučuje dohromady systémy *iNELS BUS System* a *iNELS RF Control* [10]. Zkušenosti firmy *ELKO EP, s.r.o.* ukazují, že jejich systém dokáže ušetřit až 40% celkových nákladů na energie v domácnosti. Podíl na tom má zejména efektivní správa vytápění, osvětlení a domácích spotřebičů, která je centralizovaná a přístupná prakticky odkudkoliv [8][9][10][11].

iNELS BUS System využívá pro propojení prvků a jejich vzájemnou komunikaci dvouvodičovou sběrnici. Její maximální rozsah je 6 x 550 m a je možné k ní připojit celkem až 192 digitálních a analogových jednotek. Systém je možné rozšířit o multimediální nastavbu s označením *iMM* [10]. Toto rozšíření obsahuje datové úložiště, na které lze ukládat fotografie, hudbu, videa a přistupovat k nim pomocí televize nebo jiného zobrazovacího zařízení. *iMM* také poskytuje přístup k internetu, televizi, přehrávání optických médií a monitorování objektů prostřednictvím kamerového systému. Vše je možné ovládat pomocí televize, smartphonu, tabletu nebo dalších pro tento účel navržených zařízení [9].

Bezdrátové řešení s označením *iNELS RF Control* je mladší a zatím ne tolik vyspělé. Umožňuje spravovat spínací, stmívací, roletové a teplotní aktory. Pro ovládání jsou k dispozici různé nástěnné vypínače a specializované ovladače. Aplikace pro chytrý telefon a tablet by tedy mohla být velkým přínosem a zvýšit oblíbenost tohoto systému u zákazníků.



Obrázek 2.1 Systém iNELS SMART HOME SOLUTION [10].

2.2 iNELS RF Control

iNELS RF Control je bezdrátový autonomní systém inteligentní elektroinstalace, který sestává z vysílačů a aktorů. Každý prvek systému má přidělenou jedinečnou adresu, na základě které probíhá

komunikace. Využívá se pro ni RF přenos na kmitočtu 868 MHz. Prvky mezi sebou komunikují přímo nebo prostřednictvím různých zařízení v režimech routerů a repeaterů.

Systém umožňuje spínání spotřebičů, stmívání světel, nastavování světelných scén a ovládání žaluzií a vytápění. Vše je možné řídit manuálně či automaticky pomocí různých ovladačů a senzorů. Díky detektorům pohybu a otevření oken nebo dveří je možné nastavit reakci systému při nedovoleném vniknutí do objektu [11].

2.2.1 Vysílače

Ovladače nebo senzory, které zasílají RF signál aktorům na základě určitého podnětu. Aktory potom signál dekódují a provedou podle něj naprogramovanou akci. Firma ELKO EP s.r.o. nabízí pro svůj systém bezdrátové komunikace širokou řadu ovládacích prvků. Nejdůležitější z nich jsou uvedeny dále v textu.

Celý systém iNELS RF Control je kompatibilní se zabezpečovacím systémem *OASiS* firmy *Jablotron*. Do elektroinstalace lze tedy zakomponovat i detektory této firmy pro zjištění pohybu osob, otevření okna nebo dveří. Detektory komunikují s aktory pomocí protokolu *OASiS*.

2.2.2 RF Touch

Jedná se o ovládací dotykovou jednotku, která umožňuje centrální ovládání všech RF aktorů z jednoho místa. *RF Touch* disponuje 3,5 palcovým barevným dotykovým displejem. Ovládání je zprostředkováno pomocí jednoduchého a intuitivního GUI, ve kterém lze snadno přidávat nové teplotní, spínací, stmívací i žaluziové aktory. Následně je možné těmto zařízeními vysílat povely pomocí RF signálu a na základě zpětné reakce monitorovat jejich stav. Velmi užitečnou funkcí je nastavování týdenních programů pro regulaci topení [13].

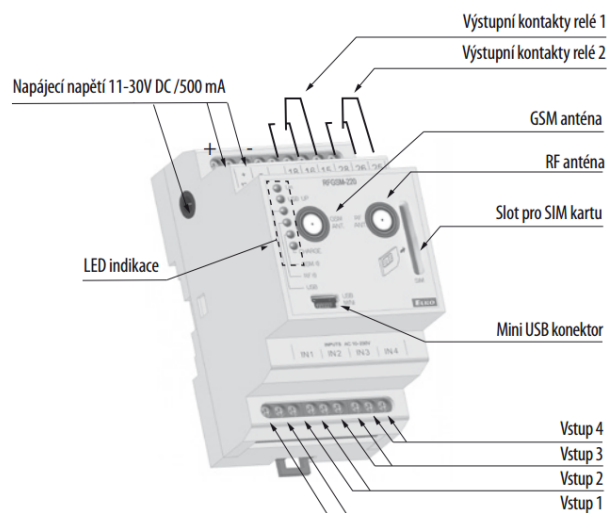


Obrázek 2.2 Jednotka RF Touch [14].

2.2.3 GSM komunikátor RFGSM-220M

Tato jednotka slouží pro bezdrátové ovládání aktorů elektroinstalace pomocí GSM sítě. Dále je díky ní možné ovládat zařízení, která jsou připojena na reléových výstupech. Stav bezdrátových aktorů a jednotek připojených na napěťových vstupech jsou indikovány pomocí zasílání SMS zpráv nebo prozváněním na mobilní telefon. Na stejném principu funguje i ovládání.

Konfigurace probíhá pomocí příloženého softwaru a umožňuje přidat až 30 telefonních čísel. GSM vysílač pracuje ve čtyřech frekvenčních pásmech (850, 900, 1800 a 1900 MHz). Obsahuje 2 výstupy s přepínacími kontakty a 4 oddělené vstupy. Může obsluhovat až 10 spínacích RF aktorů a sledovat stav až 10 bezdrátových detektorů a teplotních senzorů. Díky Li-Ion baterii se komunikátor vyrovná i s výpadky sítě [14].

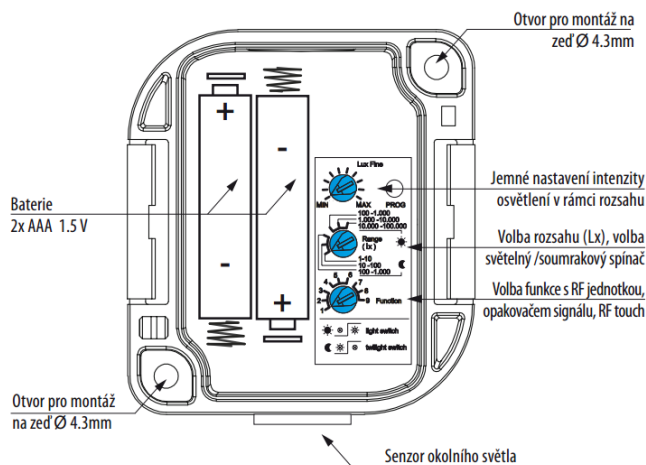


Obrázek 2.3 GSM komunikátor RFGSM-220M

2.2.4 Soumrakový světelný spínač RFSOU-1

RFSOU-1 je vysílač, jehož součástí je vestavěný světelný senzor. Díky tomuto senzoru se dá spínač nastavit do dvou režimů:

- **Soumrakový spínač** – Režim pro spínání při poklesu intenzity okolního světla a rozeptání při zvýšení intenzity. Režim je vhodné použít pro spínání večerního osvětlení.
- **Světelný spínač** – V tomto režimu zařízení spíná při zvýšení intenzity okolního světla a rozeptává při jejím snížení. Může být použit pro zatemňování žaluzií nebo aktivaci solárních panelů [14].

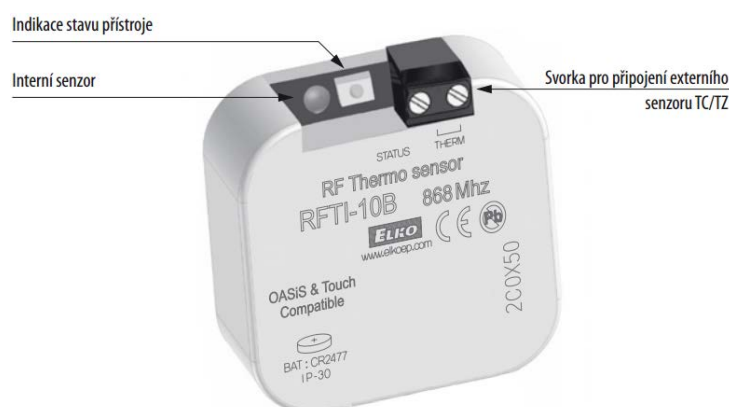


Obrázek 2.4 Soumrakový světelný spínač RFSOU-1 [14].

2.2.5 Teplotní senzor RFTI-10B

Tento vysílač slouží jako snímač teploty a údaje o naměřené teplotě zasílá v pravidelných intervalech do jednotky RF Touch. Jednotka na základě naměřených teplot a nastaveného topného programu může vysílat signály pro sepnutí nebo rozeptnutí topných zařízení. RFTI-10B obsahuje interní teplotní senzor, ale lze k němu připojit i externí senzor a snímat s ním například venkovní teploty.

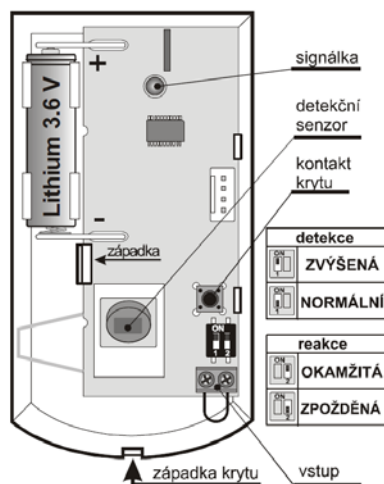
Snímání teploty probíhá pomocí NTC termistoru, což je termistor s negativním teplotním koeficientem. Pro měření se využívá můstková výchylková metoda a hodnota měnicího se odporu termistoru, který při zahřívání klesá [14].



Obrázek 2.5 Teplotní senzor RFTI-10B [14].

2.2.6 JA-80P bezdrátový PIR detektor pohybu osob

Jedná se o zařízení firmy Jablotron, které slouží pro detekci pohybu osob v interiérech budov. Detekční charakteristiku lze změnit použitím jiné čočky. Alternativní čočky umožňují například eliminovat pohyb drobných zvířat po podlaze díky snímání pouze prostoru nad podlahou, nebo vytvořit detekční stěnu. Lze nastavit citlivost snímání a tím omezit možnost vzniku falešných poplachů [15].



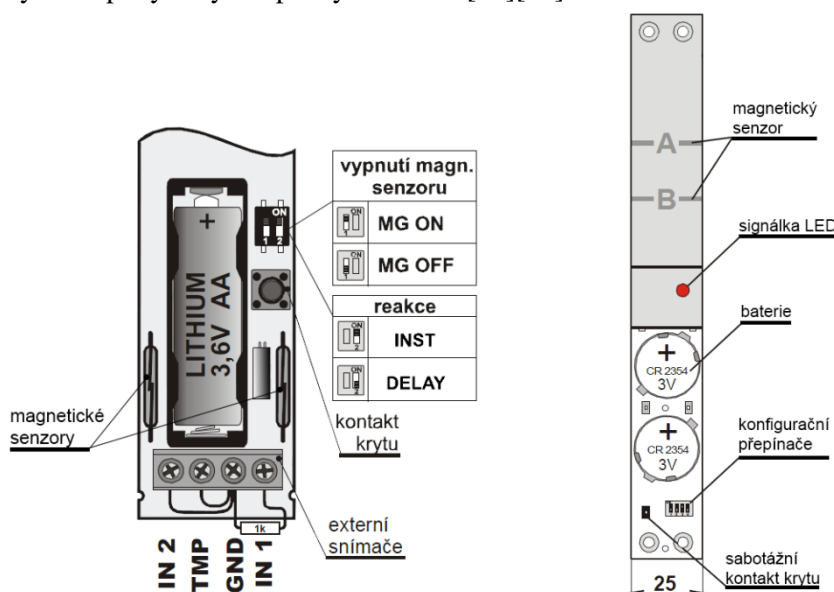
Obrázek 2.6 JA-80P bezdrátový PIR detektor pohybu osob [15].

2.2.7 Dveřní a okenní detektory JA-81M a JA-82M

Obě zařízení jsou výrobky firmy Jablotron a slouží pro detekci otevření (okna nebo dveří). Jsou to magnetické detektory, takže reagují na oddálení magnetu od senzoru.

JA-81M disponuje vstupy pro připojení dalších snímačů, jako jsou detektory otevření dveří a různé jiné senzory se spínacími kontakty (například záplavový detektor). Může být využit i k detekci manipulace s předokenní roletou, pokud je tato roleta vybavena rohatkou.

Detektor *JA-82M* je používán především k detekci otevření okna. Díky jeho tvaru je možné ho umístit do mezery mezi pohyblivým a pevným rámem [16][17].



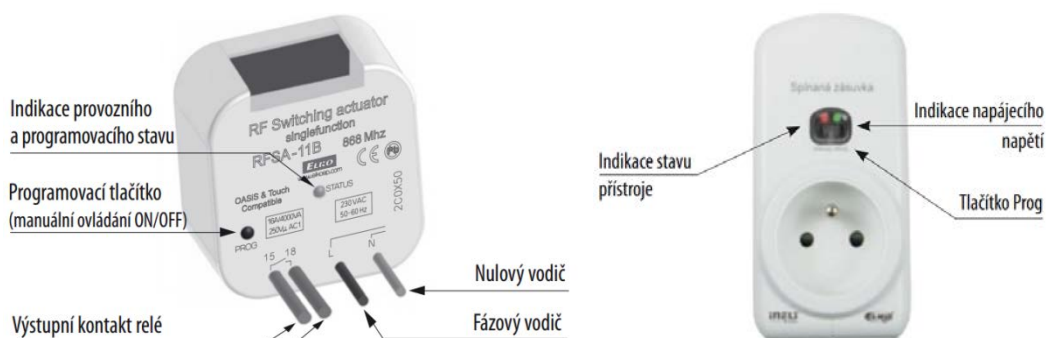
Obrázek 2.7 a) Detektor *JA-81M* [16] . b) Detektor *JA-82M* [17].

2.2.8 Aktory

Pro systém iNELS RF Control nabízí firma ELKO EP, s.r.o. celou řadu různých aktorů. Dají se rozdělit do pěti hlavních kategorií: spínací, stmívací, roletové, teplotní a analogové.

2.2.9 Spínací aktory

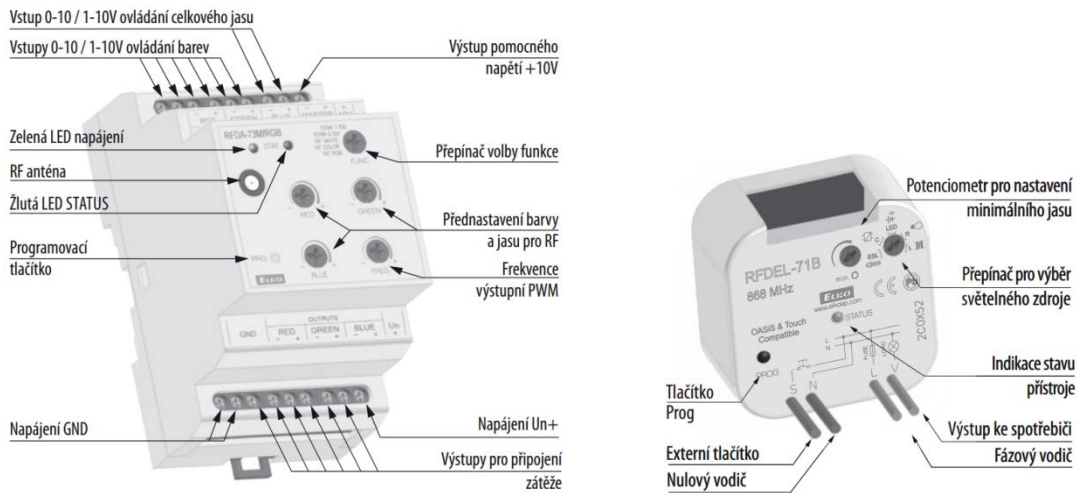
Přijímače, které slouží ke spínání různých elektrospotřebičů, jako jsou zásuvky a osvětlení, se nazývají spínací aktory. Mimo klasické spínání/vypínání disponují i dalšími funkcemi, jako je impulsní relé, tlačítko, zpožděný rozběh nebo návrat s možností nastavení na 2s až 60 min. Jako zástupce této kategorie je možno uvést spínanou zásuvku *RFSC*, která je vhodná pro ovládání domácích spotřebičů. Dále pak aktor *RFSA*, který se používá pro spínání spotřebičů bez kabelu do zásuvky (osvětlení, garážová vrata atd.). Pro náročná prostředí je potom určený aktor typu *RFUS* díky zvýšenému IP 65 krytí [14].



Obrázek 2.8 a) Jednotka *RFSA-61B* [14]. b) Jednotka *RFSC-61* [14].

2.2.10 Stmívací aktory

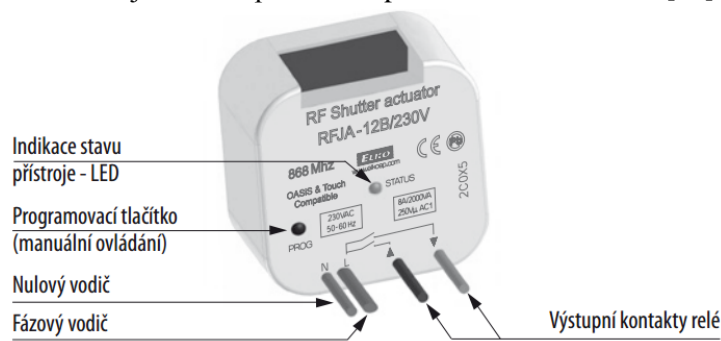
Umožňují plynule regulovat intenzitu osvětlení žárovek, halogenových svítidel, zářivek se stmívatelným předřadníkem (*RFDEL-71B*), LED pásků (*RFDA-73M*), RGB pásků apod. Disponují funkcemi pro pomalý náběh a vypínání osvětlení (v rozmezí 2s – 30 min). Umožňují vytváření tzv. světelných scén, což je nastavení různých zdrojů osvětlení na předem nastavené hodnoty [14].



Obrázek 2.9 a) Jednotka RFDA-73M [14]. b) Jednotka RFDEL-71B [14].

2.2.11 Roletové aktory

Tato zařízení slouží pro ovládání žaluzií, markýz, rolet, garážových vrat a podobných věcí. Disponují časovou funkcí, díky které si aktor zapamatuje dobu zatahování a vytahování rolet. Mohou být ovládány až 32 kanály, kde kanál představuje jedno tlačítko na ovládací jednotce. Aktor se vyrábí ve dvou verzích. První verze s označením *RFJA-12B/230V* je napájena střídavým napětím 230V a používá se pro ovládání klasických rolet a markýz. Verze *RFJA-12B/24VDC* je napájena stejnosměrným napětím 24V a je vhodná především pro meziokenní žaluzie [14].

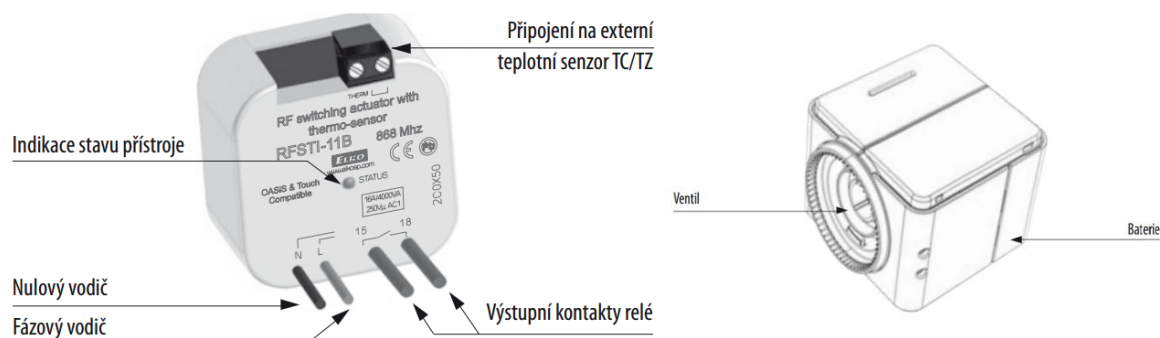


Obrázek 2.10 Jednotka RFJA [14].

2.2.12 Teplotní aktory

Teplotní aktory slouží k ovládání topných zařízení, jako jsou kotle, radiátory, podlahové topení, klimatizace a další. Jsou vybaveny teplotními senzory, které měří aktuální teplotu a údaje o ní zasílají řídicí jednotce, kde je nastavený určitý topný program. Jednotka potom vyhodnotí, zda je potřeba zvýšit nebo snížit teplotu a informaci odešle zpět aktoru, který provede požadovanou operaci.

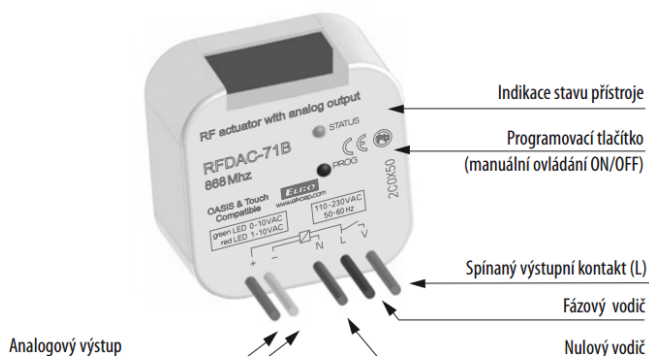
Existuje více typů těchto aktorů. Patří mezi ně spínací *RFSTI-11*, který spíná a rozepíná topné okruhy a dále termohlavice *RFATV-1*, která ovládá ventily topných těles. Je možné k nim připojit i externí teplotní senzory pro získávání teploty z jiných míst [14].



Obrázek 2.11 a) Jednotka RFSTI-11 [14]. b) Termohlavice RFATV-1 [14].

2.2.13 Analogové aktory

Aktory určené pro regulaci zařízení, která je nutno ovládat spojitým analogovým signálem 1 – 10V. Využívají se především k řízení výkonových stmívačů, zářivek se stmívatelným předřadníkem a pro ovládání jiných regulátorů jako jsou termohlavice, ventily topení, větrání atd. Po sepnutí aktoru je na jeho výstupu nastaveno poslední známé napětí. Firma ELKO EP, s.r.o. má v nabídce zatím jediné zařízení tohoto typu, které nese označení *RFDAC-71B* [14].



Obrázek 2.12 Jednotka RFDAC-71B [14].

2.2.14 Komunikace

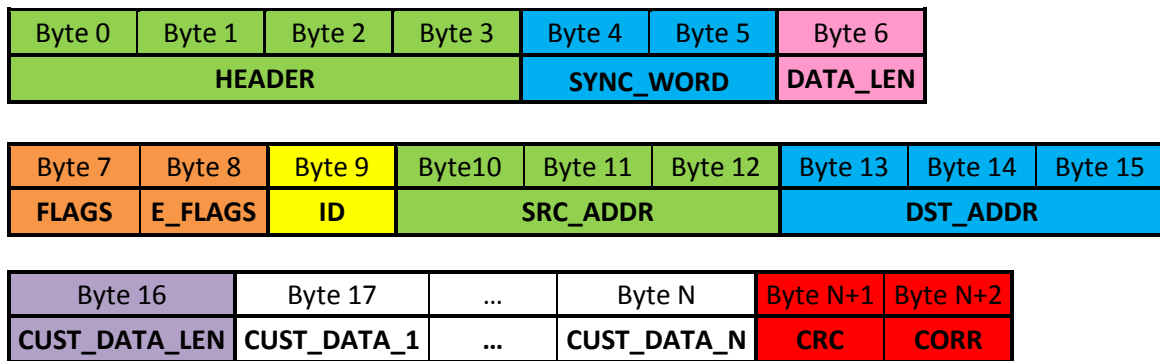
Komunikace v iNELS RF Control probíhá prostřednictvím radiofrekvenčního přenosu na frekvenci 868 MHz. Vysílače zasílají pakety s povely a aktory na ně odpovídají informací o změně svého stavu. Na obrázku 2.13 je vidět struktura paketu, kterou používá pro komunikaci mezi jednotkami protokol *Elko*. První čtyři byty jsou hlavičkové a tvoří je hodnoty 0xAA. Za nimi následuje synchronizační slovo, které má hodnotu 0xE1C0. V šestém bytu je zaslána celková délka datové části paketu.

Datová část je tvořena dvěma příznakovými byty. Význam jednotlivých bitů prvního z nich (FLAGS) je následující:

- **Bit 0** – Určuje, zda je v paketu příznakový byte E_FLAGS.
- **Bit 1** – Určuje, zda je v paketu zdrojová adresa SRC_ADDR.
- **Bit 2** – Určuje, zda je v paketu cílová adresa DST_ADDR.

- **Bit 3** – Určuje, zda je v paketu byte CUST_DATA_LEN s délkou dat jednotky.
- **Bit 4** – Určuje, zda je v paketu je samoopravný kód CORR.
- **Bit 5** – Určuje, zda je v paketu pole ID.
- **Bit 6** – Určuje, zda se jedná o potvrzovací paket od aktoru (ACK).
- **Bit 7** – Paritní bit (výsledek operace xor bitů 0 - 6).

Byte E_FLAGS je nepovinný význam jeho bitů není stanoven (pouze poslední bit musí být paritní). Pole ID obsahuje identifikaci paketu. Podle něj se dá rozeznat, jestli se jedná o novou zprávu nebo o opakování dříve zasláné zprávy. Za ID následuje šest adresových bitů, kde první tři obsahují adresu zaslajícího prvku (SRC_ADDR) a další tři adresu cílového prvku (DST_ADDR). V bytech 17 – N jsou data, která jsou specifická pro každou jednotku. Při směru komunikace vysílač – aktor obsahují povely, které má aktor vykonat. Při opačném směru komunikace (zpráva typu ACK) je v nich uveden stav jednotky. V CRC bytu je uveden kontrolní součet celého paketu, což zajišťuje integritu přenášených dat. Poslední pole paketu je CORR a obsahuje data potřebná pro korekci chyb ve zprávě.



Obrázek 2.13 Paket protokolu Elko.

2.2.15 eLAN-RF-003

eLAN-RF-003 by mohl být zařazen do kategorie vysílačů. Jedná se o zařízení, které převádí povely z LAN TCP/IP sítě na RF signál a umožňuje tak ovládání aktorů bezdrátové elektroinstalace prostřednictvím klasické LAN sítě. Komunikace může probíhat v obou směrech, takže ovládací rozhraní má zpětnou vazbu o stavu aktorů. Konfigurace a správa zařízení může probíhat pomocí IMM serveru (centrální úložiště dat multimediální nastavby systému sběrnice elektroinstalace iNELS) a díky tomu je možné ovládat prvky bezdrátové elektroinstalace prostřednictvím sběrnice systému. Dalšími možnostmi správy a konfigurace je webové rozhraní a software eLAN-RF-003 Configurator.

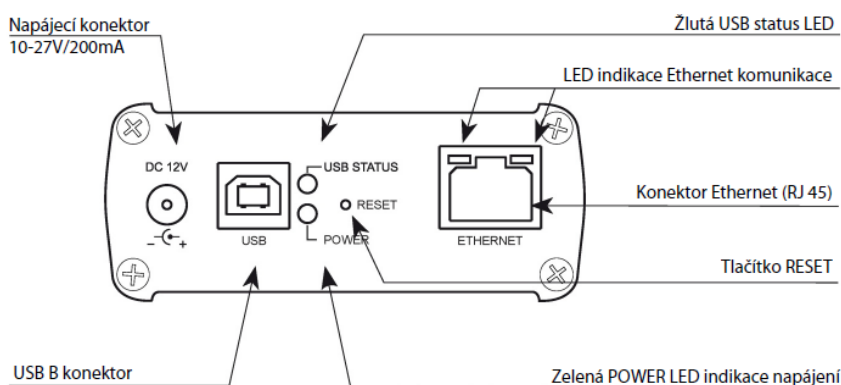
Jednotka má dvě základní funkce:

- **RF Router** – Slouží k prodloužení vzdálenosti a ovládání RF aktorů pomocí propojení dvou jednotek eLAN-RF síťovým kabelem s koncovkami typu RJ45. První jednotka v režimu MASTER přijímá RF povely. Příchozí povely přeloží a odešle příkaz po ethernetovém rozhraní. Druhá jednotka v režimu SLAVE přijímá příkazy ze sítě Ethernet, překládá je a zasílá RF aktorům. Pokud přijde zpět od aktoru zpráva, jednotka v režimu SLAVE ji odešle zpět jednotce v režimu MASTER.
- **RF Repeater** – Používá se samostatně nebo v kombinaci s RF Routerem. Při samostatném použití přijímá RF signály z ovladačů a přeposílá je dále RF aktorům. Stejným způsobem zasílá i zprávy od aktorů zpět vysílačům. Při propojení s RF routerem přes ethernetové

rozhraní přijímá a přeposílá RF signály a zároveň překládá i povely z Ethernet sítě a odesílá je RF aktorům.

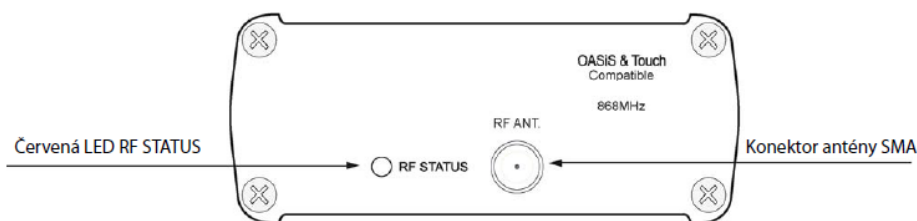
Pokud jsou obě funkce deaktivované a zařízení je v základním nastavení, poslouchá pouze povely ze sítě LAN a všechny ze sítě RF ignoruje.

V jednotce je naimplementovaný vlastní web server, který slouží pro komunikaci s PC prostřednictvím webového prohlížeče. V tomto rozhraní je možné přidávat plánky pokojů a do nich pak ikony pro různé spotřebiče, ke kterým se přiřazují příslušné akce. Po uložení konfigurace lze prostřednictvím ovládacího panelu ve webovém rozhraní zobrazit plánek příslušného pokoje a pomocí „klikání“ na ikony zasílat povely do eLAN-RF. Všechny tyto informace se ukládají přímo do interní paměti jednotky, kterou je vybavena, a webové rozhraní je potom pouze načítá [18].



Obrázek 2.14 Přední panel eLAN-RF-003 [18].

- **Napájecí konektor** – Vstup pro připojení napájecího adaptéru.
- **Žlutá USB status LED** – Indikuje aktivitu USB adaptéru.
- **LED indikace Ethernet komunikace** – Indikuje aktivitu komunikace a provozní stav na ethernetovém rozhraní
- **Konektor Ethernet** – Konektor pro připojení do sítě LAN (typ konektoru RJ 45)
- **Tlačítko RESET** – Tlačítko restart nebo pro obnovení továrního nastavení
- **Zelená POWER LED indikace napájení** – Indikuje napájecí napětí.
- **USB B konektor** – Slouží pro servisní účely nebo pro napájení zařízení.



Obrázek 2.15 Zadní panel eLAN-RF-003 [18].

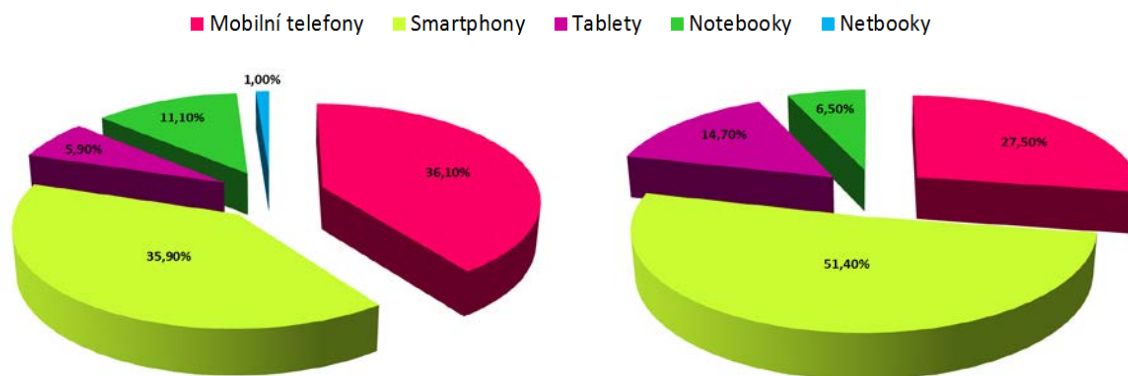
- **Červená LED RF STATUS** – Indikuje aktivitu RF komunikace.
- **Konektor antény SMA** – Slouží pro připojení RF antény.

2.3 Smartphony a tablety

Pojmem *smartphone* se rozumí telefon s otevřeným operačním systémem a pokročilými funkcemi. V dnešní době standardně disponuje několikapalcovým dotykovým displejem (3'' - 5'') s vysokým rozlišením a výkonným procesorem. Běžnými součástmi bývá Wi-Fi modul, GPS přijímač, fotoaparát a různé senzory (pohybové, světelné atd.).

Tablet je mobilní zařízení s velkým dotykovým displejem, jehož úhlopříčka je ovšem oproti smartphonům daleko větší (7'' - 12''). Jejich vybavení bývá podobné jako u chytrých telefonů, ale mívají vyšší výkon.

Díky nízké ceně, vysokému výkonu a mobilitě jejich popularita stále roste. Z výzkumu firmy *Canalys* (obrázek 2.16) vyplývá, že za rok 2012 činily celosvětové dodávky smartphonů 35,9% ze všech mobilních zařízení. Dodávky tabletů tvořily pouze necelých 6%, což je ovšem dáno tím, že přišly na trh mnohem později. Podle odhadů by měla v příštích letech tato čísla stále stoupat a v roce 2016 by mohly dodávky smartphonů a tabletů tvořit dohromady přes 65% [19]. Tato skutečnost je velice příznivá pro vývoj aplikací určených pro tato zařízení. Další velkou výhodou je otevřený operační systém, který poskytuje vývojářům velké množství různých knihoven pro využívání systémových komponent.



Obrázek 2.16 Celosvětové dodávky mobilních zařízení v roce a) 2012 b) 2013 [19].

2.3.1 Obrazovky

Obrazovky mobilních zařízení využívají v současnosti dvě odlišné technologie. Je to technologie LCD (liquid crystal displays), založená na polarizaci světla a technologie LED (light-emitting diodes), která používá pro zobrazování matici LED diod. Do mobilních zařízení jsou instalovány displeje, které jsou založeny na dvou konkrétních implementacích těchto technologií. TFT (thin film transistor) displej je založený na metodě LCD a jeho výhodou je dlouhá výdrž. AMOLED (active-matrix organic LED) displej používá LED technologii, nepotřebuje podsvícení a jeho výhodou je dobrý kontrast a čitelnost za špatných světelných podmínek [23].

2.3.2 Způsoby ovládání

Mobilní telefony dříve poskytovaly ovládání pouze pomocí hardwarové klávesnice. Smartphony v dnešní době od tohoto řešení ustupují a naprostá většina z nich má ovládání řešené pomocí dotykové obrazovky. Existují tři základní metody pro zjištění pozice, kde došlo ke kontaktu s obrazovkou:

- **Rezistivní** – Na skleněném povrchu displeje se nacházejí dvě rezistivní vrstvy. Je-li na displej vyvinut v nějakém místě tlak, vrstvy se navzájem dotknou a na základě změny odporu se vypočítají souřadnice, kde k dotyku došlo. Tato metoda je nízkonákladová, ale displej propustí pouze 70% světla.
- **Kapacitní** – Na povrchu displeje je vrstva pod napětím. Po přiložení prstu nebo jiného vodivého materiálu dojde k odebrání určitého množství elektrického náboje a tím se změní kapacitance. Po jejím změření se zjistí místo, kde došlo ke kontaktu. V dnešní době patří tento typ displejů mezi nejpoužívanější. Výhodou je vysoká propustnost světla (až 90%), ale jeho přesnost je menší než u rezistivních displejů.
- **Povrchová akustická vlna** – Displeje využívající tuto metodu vysílají a přijímají akustické vlnění. Když se prst dotkne obrazovky, vlnění je jím částečně absorbováno a na základě jeho měření je vypočítáno místo kontaktu. Tato metoda je nejpřesnější, ale hodí se více pro zařízení s velkými obrazovkami jako je například bankomat.

Vedle dotykových obrazovek existují ještě alternativní způsoby ovládání:

- **D-pad** – Všesměrový joystick.
- **Trackball** – Směrovací zařízení využívající kuličku podobně jako počítačová myš.
- **Trackpad** – Využívá speciální povrch pro snímání pohybu prstu [23].

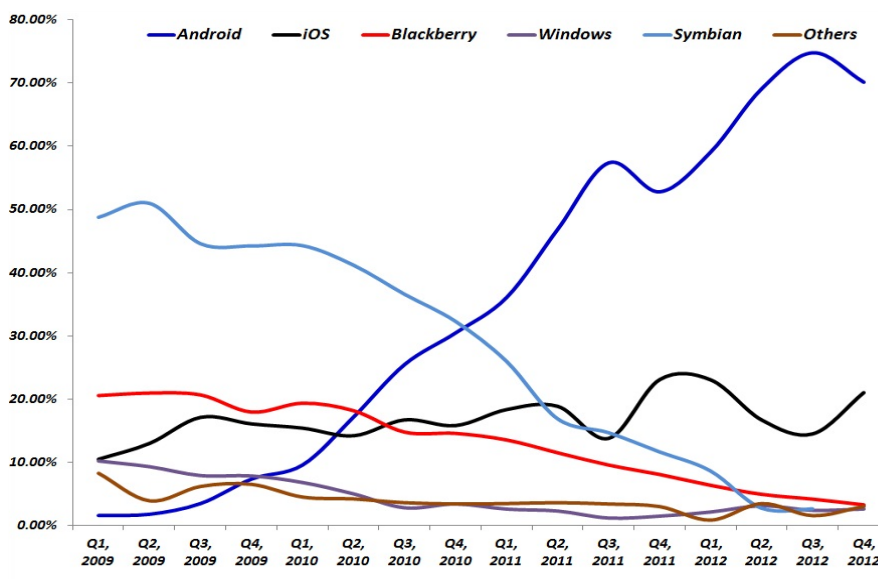
2.4 Platforma Android

Andy Rubin, tvůrce a bývalý ředitel vývoje systému Android ho označil za „*první opravdu otevřenou a rozsáhlou platformu pro mobilní zařízení*“ [20]. Tento systém byl původně vyvíjen malou firmou Android Inc., kterou finančně podporovala a nakonec v roce 2005 skoupila společnost Google, která tím potvrdila dlouhodobé spekulace o svém zájmu na vývoji mobilních zařízení. Android byl představen v listopadu roku 2007 společně s oznámením o vzniku OHA (Open Handset Alliance) OHA je seskupení společností jako jsou *HTC, Qualcomm, Motorola, NVIDIA* a další. Jejich cílem je spolupracovat na vývoji otevřených standardů pro mobilní zařízení. I když jsou zdrojové kódy Androidu vyvíjeny primárně společností Google, členové OHA mají možnost do nich také zasahovat.

První verze této platformy s označením 1.0 byla vydána v říjnu roku 2008 pod open-source licencí Apache 2.0 [21]. Open-source licence a rozsáhlé možnosti využití systému představují obrovskou příležitost pro vývojáře. Volná dostupnost zdrojových kódů systému jim může pomoci v pochopení problematiky, pokud není některá jeho část dostatečně zdokumentovaná. Díky tomu si také vývojáři mohou přizpůsobit systém podle svých potřeb a nahradit nativní komponenty svými [23]. Takto modifikované uživatelské verze jsou nazývány módy, firmwary nebo romy. V současné době patří mezi nejznámější a nejlépe vybavenou verzi Cyanogen. Mnoho členů OHA vytvořilo své vlastní verze Androidu s modifikovanými uživatelskými rozhraními. Takto vzniklo například rozhraní Sense od společnosti HTC nebo MOTOBLUR společnosti Motorola.

Z výhod open-source licence a možnosti modifikace systému těží i výrobci mobilních zařízení. Android je možné po jeho přizpůsobení různým výpočetním výkonům integrovat do zařízení všech cenových kategorií. Tato platforma tedy není omezena pouze pro drahé high-end přístroje, ale objevuje se i na hůře vybavených zařízeních a tím pádem je dostupná širokému spektru lidí. Systémy jako jsou například iOS, Windows Mobile a Palm Pre určitým způsobem upřednostňují nativní aplikace před aplikacemi třetích stran. Android v tomto ohledu představuje výhodu v tom, že aplikace třetích stran mají stejnou prioritu jako nativní, využívají stejné API a stejné běhové prostředí [22][26].

Všechna výše uvedená fakta ovlivňují pozici Androidu na trhu. Z grafu na obrázku 2.17 je vidět strmý procentuální růst jeho podílu na celosvětovém trhu s mobilními platformami. V současné době ovládá okolo 70%.



Obrázek 2.17 Podíl mobilních platforem na celosvětovém trhu [24].

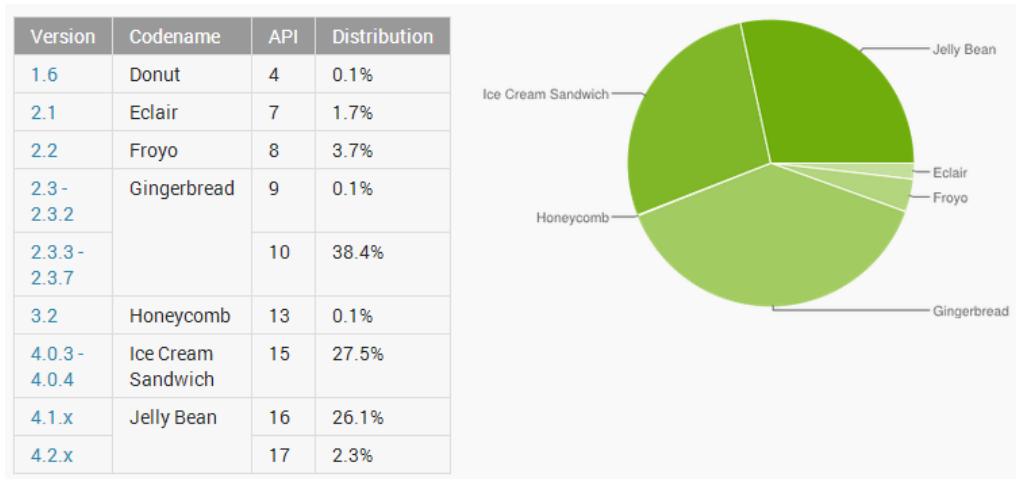
2.4.1 Verze

Během svého vývoje se Android dočkal celé řady verzí. Každá z nich přidala do systému novou funkcionalitu. Verze 1.5 s označením *Cupcake* umožnila vývojářům přidávat do jejich aplikací nativní knihovny. To přináší výhodu zejména v situacích, kdy je od aplikace požadován vysoký výkon a je třeba dbát na její efektivitu. Další verze 1.6 (*Donut*) přinesla podporu pro obrazovky s různým rozlišením. Android 2.0 přidala podporu pro multi-dotykové displeje. Ve své době velmi rozšířená verze 2.2 (*Froyo*) přinesla vylepšení virtuálního stroje *Dalvik*, který má na starosti spuštění a běh aplikací. Jednalo se o přidání JIT (just-in-time) kompilace, která dokáže zrychlit běh aplikace v závislosti na dané situaci až pětkrát. *Dalvik* byl vylepšen i verzí *Gingerbread* (2.3) přidáním nového garbage collectoru.

Další vydaná verze byla vyhrazena pouze pro tablety a nesla označení *Honeycomb* (3.0). Obsahovala výrazné změny v API. *Honeycomb* 3.1 přidal podporu obrazovek tabletů s vysokým rozlišením a možnost jejího rozdělávání. Dále byla zahrnuta podpora USB periférií, jako jsou klávesnice, myši a joysticky.

Velmi přínosná byla verze 4.0 (*Ice Cream Sandwich*), která vznikla jako výsledek spojení *Honeycombu* a *Gingerbreadu* a byla určena jak pro tablety, tak pro smartphony. Přidala mnoho vylepšení do uživatelského rozhraní a vestavěných aplikací jako jsou e-mailový klient nebo například internetový vyhledávač. Díky spojení s *Honeycombem* se i smartphony dočkaly podpory obrazovek s vysokým rozlišením. Dnes nejaktuálnější verze 4.2 (*Jelly Bean*) byla vydána na podzim roku 2012 a přinesla možnost tvoření uživatelských účtů, která je ovšem omezená pouze pro tablety, a dále snížila požadavky na paměť RAM.

Díky nesjednocené podpoře pro mobilní telefony nemají ovšem dnes všechny nejnovější verzi, ale jsou v četné míře zastoupené i starší (obrázek 2.18). Toto je problém především pro vývojáře, protože aplikace určené pro novější verzi systému jsou nekompatibilní se staršími verzemi. Proto je velmi důležité na začátku vývoje zvolit adekvátní verzi, pro kterou bude aplikace určena.



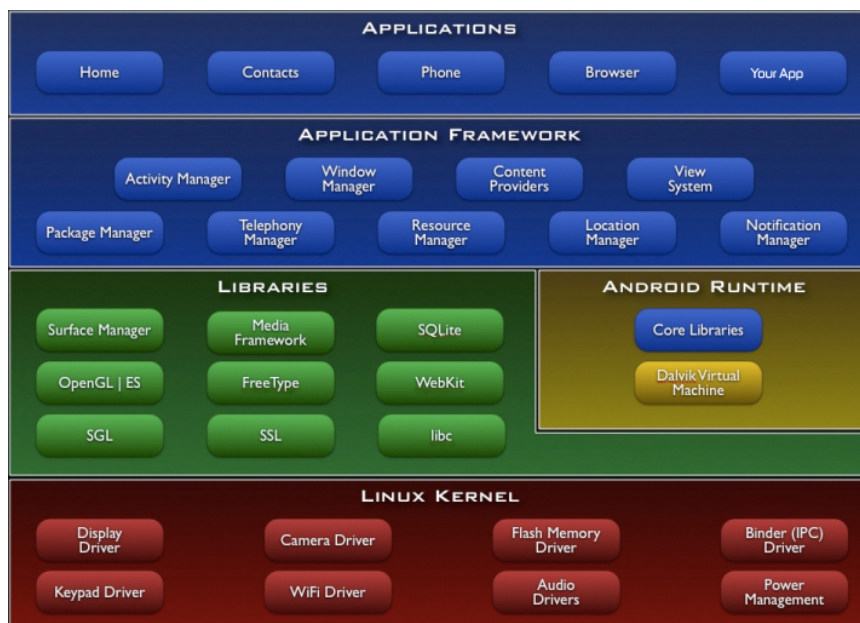
Obrázek 2.18 Současné zastoupení verzí platformy Android na mobilních zařízeních [25].

2.4.2 Architektura

Android je operační systém, který je vystavěný na Linuxovém jádře, ale nejedná se přímo o linuxovou distribuci, protože některé standardní Linuxové komponenty jako X-windows nebo knihovny GNU C nejsou Androidem podporovány. Vývoj aplikací pro tuto platformu prostřednictvím Java frameworku, který ovšem na rozdíl od Javy neobsahuje všechny standardní knihovny (AWT, Swing). Některé z nich byly nahrazeny vlastními knihovnami Androidu [23].

Architektura Androidu (obrázek 2.19), jejíž jednotlivé části jsou popsány v následujících kapitolách, má několik charakteristických rysů:

- **Aplikační framework** – Poskytuje široké spektrum rozhraní pro podporu vývoje aplikací. Také poskytuje možnost využívání a nahrazení nativních komponent komponentami třetích stran.
- **Virtuální stroj Dalvik** – Má na starosti spouštění a běh aplikací.
- **Grafické knihovny** – Zajišťují podporu pro 2D a 3D grafiku.
- **Podpora médií** – Android poskytuje podporu opravdu široké škály různých mediálních formátů, mezi které patří například Ogg Vorbis, MP3, MPEG-4, H.264 a PNG.
- **Přístup k periferiím** – Mobilní zařízení bývají vybavena mnoha periferiemi, jako jsou kamera, GPS, různé pohybové a světelné senzory, kompas, akcelerometr apod. Ke všem těmto periferiím lze přistupovat a za pomoci knihoven je využívat ve vyvíjených aplikacích.



Obrázek 2.19 Architektura platformy Android [26].

2.4.3 Jádro systému

Linuxové jádro je základ celé architektury platformy Android. Stará se o operace jako jsou přístup k perifériím, procesní management, síťová komunikace, a správa paměti a napájení. Poskytuje základní drivery pro hardwarové komponenty a tvoří tak hardwarovou abstraktní vrstvu systému (*HAL*). Tato vrstva je důležitá kvůli různorodosti mobilních zařízení, které běží pod Androidem. Poskytuje jednotný přístup k jejich hardwaru, který se liší v závislosti na typu a výrobci. Tím je zajištěna i ochrana pro vývojáře systému a aplikací [21][26].

2.4.4 Knihovny

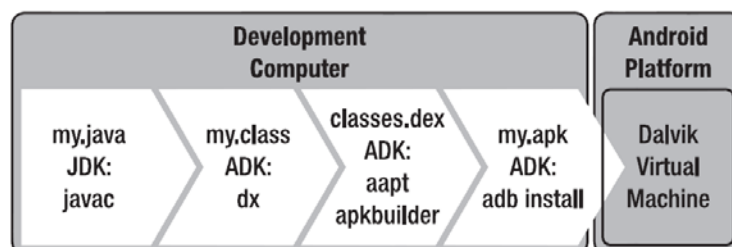
Přímo nad *HAL* jádrem běží nativní C/C++ knihovny, které zprostředkovávají služby jádra aplikacím a Android runtime. Jsou také zodpovědné za složité procesy, jako je grafické renderování, přehrávání audia nebo přístup do databáze. API knihoven je obaleno Java třídami, takže jsou přístupné vývojářům aplikací [26].

2.4.5 Runtime

Android Runtime je vrstva, která obsahuje aplikační virtuální stroj *Dalvik*. Ten byl vytvořen roku 2005 a jeho vznik byl inicializován potřebou virtuálního stroje s open-source licencí, protože zdrojové kódy Javy jsou volně šiřitelné, zatímco *JVM* nikoliv. Jedná se o open-source implementaci Java SE virtuálního stroje s registrově orientovanou architekturou. *Dalvik* je navržen tak, aby poskytoval vysoký výkon na zdrojově omezených zařízeních, jako jsou smartphony nebo tablety, a dbal na úsporu energie. Snaží se minimalizovat problémy spojené s provozem ve výpočetně omezeném prostředí a maximalizovat výhody, které poskytuje programování v jazyce Java.

Při překladu aplikace je kód nejprve zkompileován pomocí standardního Java kompilátoru. Výsledný bytekód ovšem *Dalvik* nedokáže spustit. Proto je potřeba pro spuštění aplikace provést ještě další úpravy vygenerovaného bytekódu. Celá cesta vedoucí k vytvoření spustitelné aplikace je vidět na obrázku 2.20. Tento proces se skládá z následujících kroků:

- Kód vytvořený ve vývojovém prostředí je zkompileován pomocí Java kompilátoru ze standartního JDK. Výsledkem je klasický Java bytekód.
- V další fázi jsou zkompileované třídy převedeny pomocí programu *dx* na soubory s příponou *.dex*, které jsou kompatibilní s Dalvikem.
- Vytvořené soubory jsou potom podepsány a zazipovány dohromady jako *.apk* balík (Android application package) pomocí utilit *aapt* a *apkbuilder*.
- Výsledný aplikační balík je připraven pro spuštění na zařízení s operačním systémem Android. Jejich instalace probíhá pomocí nástroje *adb* (Android Debug Bridge) zadáním příkazu *install* [21].



Obrázek 2.20 Vytvoření spustitelného aplikačního balíku [21].

2.4.6 Aplikační framework

Aplikační framework zabaluje systémové knihovny a runtime do jednoho celku. Stará se o správu aplikací a poskytuje jim přístup k systémovým knihovnám a operacím. Framework taky obsahuje řadu aplikačních rozhraní pro programování uživatelských rozhraní, upozornění, služeb běžících na pozadí atd. Všechny nativní systémové aplikace poskytované Androidem, jako je například mail klient, webový prohlížeč nebo přehrávač multimedií, jsou napsané s využitím těchto aplikačních rozhraní.

Klasické aplikace s uživatelským rozhraním nebo služby běžící na pozadí mohou poskytovat své komponenty jiným aplikacím. Příkladem může být aplikace fotoaparátu, která umožňuje pořídit snímek a pak na něj aplikovat různé efekty. Komponenta, která zajišťuje zachycení fotografie, může být potom využita třeba aplikací pro fotogalerii. Tento systém velice pomáhá vývojářům, kteří nemusejí implementovat všechny funkce sami, ale mohou k tomu využívat již vytvořené aplikace [26].

2.4.7 SDK

SDK (Software Development Kit) je sada umožňující vývoj aplikací pro platformu Android. Skládá se z nástrojů, dokumentace, tutoriálů a vzorových aplikací, které pomáhají pochopit problematiku a použití některých komponent a konstrukcí. Jeho velmi důležitou součástí jsou také Java knihovny potřebné pro vývoj aplikací. Obsahem těchto knihoven jsou rozhraní aplikačního frameworku. SDK je možné nainstalovat na všechny hlavní desktopové operační systémy.

SDK obsahuje tyto nástroje:

- **DDMS** – Dalvik Debug Monitor Server, který zprostředkovává logcat (slouží pro zobrazování ladících informací), snímání obrazovky, simulaci přichozích hovorů a textových zpráv informace o procesech, rádiovém přijímači, síti, vláknech a paměti.

- **Hierarchy Viewer** – Slouží k ladění a optimalizaci uživatelského rozhraní. Poskytuje vizuální reprezentaci hierarchie layoutů v aplikaci.
- **Emulátor** zařízení sloužící pro emulaci zařízení s platformou android na desktopovém systému je založen na QEMU (open-source virtuální stroj pro simulaci různých hardwarových platforem).
- **Monkey** – Běží v emulátoru nebo na připojeném zařízení a generuje pseudo-náhodné sekvence uživatelských událostí, jako jsou například dotyky a gesta, a sekvence systémových událostí.
- **ProGuard** – Zmenšuje a optimalizuje kód odstraněním nepoužívaných částí. Dále provádí obfuskaci kódu pomocí přejmenování tříd, proměnných a metod pomocí názvů s nejasnou sémantikou.
- **Systrace** – Umožňuje analýzu běhu aplikace v kontextu systémových procesů. Pomáhá diagnostikovat výkonnostní problémy.
- **sqlite3** – Umožňuje přistupovat k databázovým souborům aplikace a zobrazovat strukturu databáze [28].
- **ADB** - Příkazový řádek pro komunikaci s připojeným mobilním zařízením.
- **Překladačové skripty** - Skripty pro překlad a sestavení výsledného aplikačního balíku [26].

3 Analýza a návrh aplikace

3.1 Přehled požadavků

V této kapitole jsou shrnuty všechny požadavky na aplikaci. Požadavky jsou rozděleny na funkční, které popisují funkce aplikace, a systémové a výkonostní, které specifikují požadavky na systém a výkon aplikace.

3.1.1 Funkční požadavky

Aplikace musí poskytovat následující funkce:

- **Správa místností** – Možnost přidávání, editace a mazání místností, které slouží pro uspořádání aktorů aplikaci.
- **Správa aktorů** – Možnost přidávání, editace a správy aktorů. Aktory bude možné přiřazovat k místnostem nebo do scén.
- **Správa scén** – Možnost přidávání, editace a mazání scén, které slouží pro hromadné nastavení stavů více aktorů najednou.
- **Zobrazení stavů aktorů** – Zobrazení aktuálních stavů aktorů v místnosti.
- **Změna stavů aktorů** – Možnost nastavování stavů aktorů v místnosti.
- **Ovládání scén** – Možnost zapnutí scény (nastavení více aktorů na předem nastavené hodnoty pomocí).
- **Informace o jednotce eLAN-RF** – Zobrazení informací o verzi hardwaru a firmwaru.
- **Vícejazyčnost** – Možnost nastavení jazyku aplikace.
- **Autentizace pro eLAN-RF** – Přihlašování k jednotce eLAN-RF.

3.1.2 Systémové a výkonostní požadavky

Požadavky na systém a výkon aplikace jsou následující:

- **Platforma Android 2.3** – Aplikace poběží na tabletech a smartphonech s operačním systémem Android verze 2.3 a vyšší.
- **API verze 10** – Aplikace bude podporovat API verze 10 a vyšší.
- **Rychlé ovládání aktorů** – Mezi změnou stavu aktoru v aplikaci a jeho skutečnou změnou musí být co nejmenší prodleva.
- **Omezení síťové komunikace** – Komunikace s jednotkami elektroinstalace nesmí probíhat příliš často, aby nedošlo k zahlcení pásma a také kvůli úspoře napájení jednotek (některé jednotky mohou být napájeny pomocí baterie).
- **Nízká spotřeba** – Vzhledem k tomu, že aplikace poběží na zařízeních napájených pomocí akumulátorů, musí být navržena efektivně, aby neodebírala příliš mnoho energie.
- **Nízké zatížení procesoru** – Kvůli omezené hardware konfiguraci mobilních zařízení a požadavku na nízkou spotřebu musí být aplikace výpočetně nenáročná.

3.2 Architektura

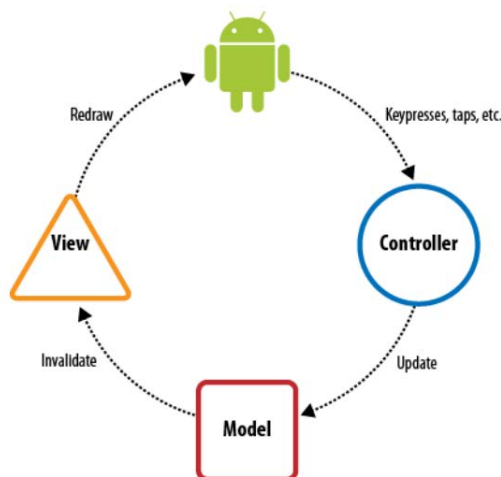
Aplikace pro ovládání inteligentní elektroinstalace bude vycházet z architektury MVC (Model-View-Controller) stejně jako většina aplikací vyvíjených pro Android.

3.2.1 MVC

Architektura MVC (Model-View-Controller) je organizována do tří vrstev. Tyto vrstvy od sebe oddělují uživatelské rozhraní, řídicí logiku aplikace a data. Díky tomuto způsobu organizace je zdrojový kód dobře strukturovaný a modulární. Změna v jedné vrstvě má minimální dopad na ostatní vrstvy.

Celý koncept architektury MVC je vidět na obrázku 3.1. Její vrstvy jsou:

- **Model** – Vrstva, která definuje základní funkcionalitu aplikace. Obsahuje struktury dat a funkce pro práci s těmito daty. Příkladem může být aplikace hudebního přehrávače. Model potom obsahuje databázi hudebních souborů a funkci pro přehrávání hudby. Jeden Model může být využíván více aplikacemi.
- **View** – Tato vrstva se stará o vizualizaci Modelu. Je zodpovědná za renderování grafického uživatelského rozhraní. GUI je implementováno jako strom podtříd třídy View. Kořen této stromové struktury tvoří okno aplikace. Vykreslování probíhá postupným procházením této struktury. Prvky z nižších úrovní překrývají prvky z vyšších úrovní. V případě aplikace pro hudební přehrávač může jeden View obsahovat informace o skladbě a další ovládací prvky přehrávače.
- **Controller** – Část, která reaguje na události, jako jsou stisk klávesy, dotyk obrazovky prstem, příchozí volání atd. Tyto události se ukládají do fronty požadavků, odkud jsou odebírány frameworkem a následně odeslány. Pokud například uživatel stiskne klávesu na telefonu, vygeneruje se KeyEvent, který se přidá do fronty událostí. Tato událost je po vyjmutí z fronty předána jako parametr metodě dispatchKeyEvent. V Controlleru může být nadefinovaný handler, který tuto událost zachytí a provede nadefinovanou akci [29].

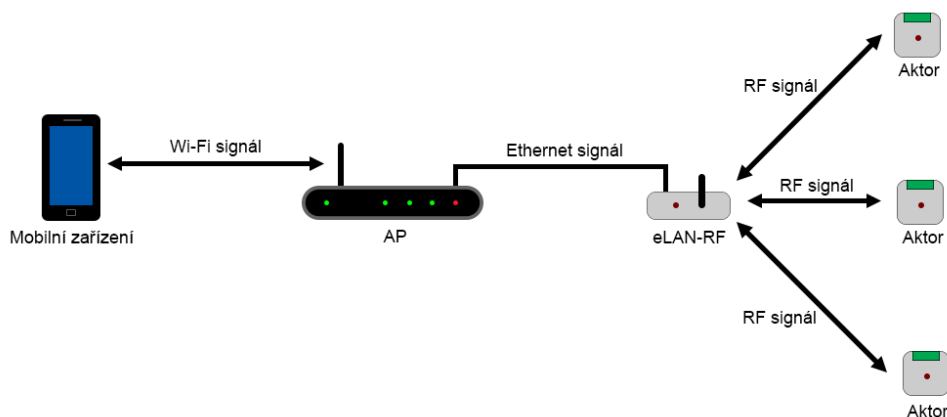


Obrázek 3.1 Architektura typu MVC [29].

3.3 Komunikace

Pro ovládání bezdrátové elektroinstalace se bude využívat princip klient-server. V roli klienta bude aplikace běžící na mobilním zařízení, která bude obsahovat řídicí logiku a potřebná data. Aplikace se bude připojovat na server a zasílat mu podle potřeby pakety s dotazy a povely. V roli serveru bude zařízení eLAN-RF, které převádí povely z LAN TCP/IP sítě na RF signál. eLAN-RF bude přijímat povely od mobilního zařízení, provádět jejich transformaci na pakety podle protokolu Elko a ty pak pomocí rádio-frekvenčního signálu zasílat aktorům v elektroinstalaci. Aktory na základě příchozího povelu provedou definovanou akci (změní svůj stav) a odešlou odpověď zpátky jednotce eLAN-RF. Ta provede transformaci příchozích dat z RF signálu na data pro LAN TCP/IP síť a zašle je zpět mobilnímu zařízení. Aplikace běžící na mobilním zařízení přijme příchozí data a zpracuje je dle potřeby.

Celý tento komunikační model je uveden na obrázku 3.2. Vzhledem k tomu, že smartphony a tablety lze připojit k síti LAN pouze pomocí Wi-Fi modulu, bylo potřeba do schématu zakomponovat přístupový bod (AP), který vytváří Wi-Fi síť a pomocí LAN kabelu je připojen k jednotce eLAN-RF. Mobilní zařízení potom zasílá pakety prostřednictvím Wi-Fi sítě a AP je přeposílá jednotce eLAN-RF.



Obrázek 3.2 Schéma komunikace.

3.3.1 Protokol NippNET

Komunikace mezi aplikací na mobilním zařízení a jednotkou eLAN-RF bude probíhat pomocí protokolu TCP. Komunikační pakety musí dodržovat formu, kterou předepisuje protokol *NippNET* vyvinutý společností ELKO EP, s.r.o. Pakety tohoto protokolu mají následující strukturu (obrázek 3.3)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
HEADER				ID1	ID2	DATA_LEN	FLAGS	TYPE

Obrázek 3.3 Struktura paketu protokolu NippNET.

- **Hlavička** – Je tvořena čtyřmi prvními byty paketu (HEADER). Slouží pro rozpoznání paketů protokolu NippNET. Byty mají hodnoty postupně 0xFF, 0xFA, 0xFF a 0xFF.
- **Identifikace paketu** – Dva identifikační byty (ID1, ID2) slouží pro identifikaci paketu a musí se náhodně nebo s posloupností měnit pro každý odeslaný paket.

- **Délka paketu** – V šestém bytu (DATA_LEN) je uvedena celková délka paketu.
- **Příznaky** – Pole s označením FLAGS obsahuje příznakové bity. Význam bitů je následující:
 - **Bit 0** – Nastaven na 1, pokud paket obsahuje přihlašovací údaje administrátora.
 - **Bit 1** – Nastaven na 1, pokud paket obsahuje přihlašovací údaje uživatele.
 - **Bit 7** – Nastaven na 1, pokud je požadována odpověď na odeslaný paket.
 Význam ostatních bitů není definován a jsou nastaveny na hodnotu 0.
- **Typ** – V osmém bytu paketu je uvedena hodnota značící jeho typ. Protokol podporuje celkem 17 těchto typů. Vyvíjená aplikace pro ovládání iNELS RF Control bude ovšem používat pouze následující z nich:
 - **Test Packet** – Značí ho hodnota 0x00 a slouží pro otestování komunikace s jednotkou eLAN-RF. Neobsahuje žádná dodatečná data, ale pouze prvních devět bytů. Pokud se jednotka nachází na adrese, na kterou je tento paket zaslán, odpoví pomocí zprávy Test Packet ACK.
 - **Test Packet ACK** – Označen hodnotou 0x01. Je zaslán jednotkou eLAN-RF jako odpověď na Test Packet. V jeho datové části (obrázek 3.4) je obsažen typ jednotky eLAN-RF (DEVICE_ID), verze firmwaru (FW_VERSION) a verze hardwaru (HW_VERSION).

Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14
DEVICE_ID		FW_VERSION			HW_VERSION

Obrázek 3.4 Datová část paketu Test packet ACK.

- **RFIO Quick Controll** – Pro jeho označení slouží hodnota 0x04 a používá se pro zasilání povelů aktorům (směr klient-server). Na pozici devátého bytu je uvedena délka uživatelského jména a hesla (LOGIN_LEN). V následující posloupnosti bytů, která má délku danou dvojnásobkem hodnoty v LOGIN_LEN, je uvedeno přihlašovací jméno a heslo uživatele. Za nimi se nachází tři byty vyhrazené pro adresu příslušného aktoru (RFIO_ADDR) a jeden byte, který obsahuje délku dat specifických pro jednotku (DATA_LEN). Tato data jsou uvedena dále v paketu a jejich význam a délka je pro každý aktor jiná.

Byte 9	Byte 10...Byte N	Byte N...Byte 2N	Byte 2N+1	Byte 2N+2	Byte 2N+3	Byte 2N+4
LOGIN_LEN	LOGIN	PASSWORD	RFIO_ADDR			DATA_LEN

Obrázek 3.5 Datová část paketu RFIO Quick Controll.

- **RFIO Quick Controll ACK** – Odpověď na RFIO Quick Controll označená hodnotou 0x05. Devátý byte paketu (ERROR_CODE) obsahuje kód chybového stavu, v desátém (RF_STATUS) je uvedeno, zda jednotka odpověděla nebo ne. Hodnota v bytu 11 (DATA_LEN) označuje délku dat specifických pro jednotku. Tato data jsou uvedena dále v paketu a jejich délka a význam jsou proměnné podle jednotky, která paket zaslala. Pokaždé ale obsahují aktuální stav jednotky.

Byte 9	Byte 10	Byte 11
ERROR_CODE	RF_STATUS	DATA_LEN

Obrázek 3.6 Datová část paketu RFIO Quick Controll ACK.

- **Data** – Od devátého bytu dále obsahuje paket datové byty, jejichž význam závisí na typu paketu. Maximální délka těchto dat je však 244 bytů.

Vzhledem k tomu, že komunikace mezi klientem a serverem může být odposlouchávána, musí být komunikace šifrovaná. Jednotlivé pakety se tedy zakódují pomocí speciální matice hodnot a pro uložení údajů potřebných k dešifrování se využijí poslední dva byty paketu.

3.3.2 Autentizace u eLAN-RF

Aby mohla aplikace komunikovat s jednotkou eLAN-RF, musí být nejprve provedena autentizace uživatele. Jednotce budou zaslány autentizační údaje (login a heslo) prostřednictvím RFIO Quick Controll s adresou prvku nastavenou na hodnotu 0x000000 a délkou dat 0. Po zaslání tohoto paketu mohou nastat 3 případy:

- **Přijde ACK s chybovým kódem 0x00** – Autentizace byla úspěšná. Uživatel je přihlášen.
- **Přijde ACK s chybovým kódem 0x01** – Neplatné uživatelské jméno nebo heslo. Je potřeba změnit uživatelské údaje a provést autentizaci znovu.
- **Nepřijde odpověď** – Neplatná IP adresa nebo port. Je potřeba změnit IP nebo port a znovu se autentizovat.

Pokud autentizace proběhne v pořádku, jednotce eLAN-RF bude zaslán Test Packet a z odpovědi na něj se získá verze firmwaru a hardwaru. Po takto provedeném úspěšném přihlášení je možné komunikovat s jednotkou eLAN-RF pomocí zadaných autentizačních údajů.

3.3.3 Monitorování stavu aktorů

Pro monitorování stavu jednotlivých aktorů v systému je možné využít pakety typu RFIO Quick Controll. Paket musí obsahovat jméno a heslo získané autentizačním procesem, adresu aktoru a data pro definování testovacího paketu. Na testovací paket aktor pouze odpovídá paketem obsahujícím jeho stav, který zůstává neměnný.

3.3.4 Ovládání aktorů

Ovládání aktorů bude stejně jako monitorování prováděno pomocí zaslání RFIO Quick Controll paketu. Paket bude obsahovat adresu konkrétního prvku, jméno a heslo uživatele a data pro definování požadované akce. Tato data se liší podle typu aktoru, protože každý typ má různou sadu funkcí. Aktor na základě tohoto paketu změní svůj stav a odešle jako odpověď RFIO Quick Controll ACK se svým aktuálním stavem, což je důležité kvůli nutné aktualizaci jeho stavu v aplikaci na mobilním zařízení.

3.4 Data

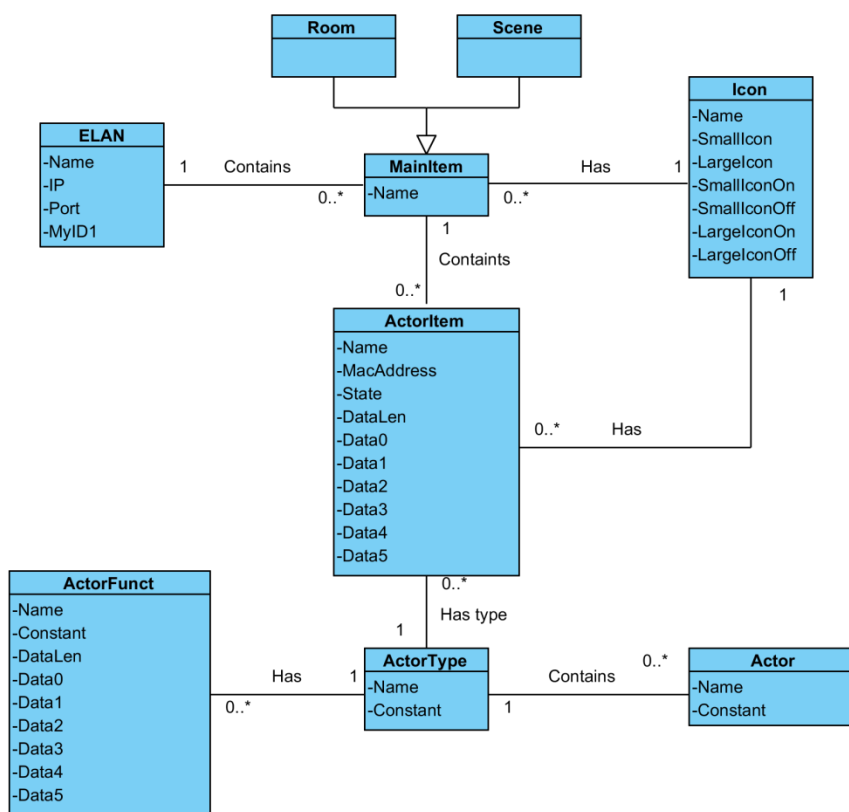
Aplikace musí poskytovat rozhraní pro správu místností, scén a jednotek, jak vyplývá ze specifikace funkčních požadavků. Z toho důvodu je potřeba informace o těchto entitách ukládat. K tomuto účelu bude nejvhodnější použít databázi.

Model databáze je uvedený na obrázku 3.7. Jeho hlavní třída ELAN obsahuje informace o jednotce eLAN-RF. Aplikace bude od začátku obsahovat jeden objekt této třídy. Po úspěšném

přihlášení k jednotce se do tohoto objektu doplní informace o IP adrese a portu. Objekt třídy ELAN může obsahovat libovolné množství objektů třídy MainItem. Ta má dvě specializace, kterými jsou pokoj (třída Room) a scéna (třída Scene). Objekty těchto tříd mají přidělený název a objekt třídy Icon, ve které jsou uvedeny cesty k potřebným ikonám. Hlavní položka může obsahovat libovolné množství objektů třídy ActorItem, která reprezentuje jednotlivé aktory ve scéně nebo pokoji. Každá jednotka má přidělený název, ikonu, adresu aktoru, jeho stav a pak délku dat pro paket RFIO Quick Controll a předem nastavené hodnoty datových bytů, které se budou využívat při nastavování scény.

Každá jednotka má přidělenou kategorii aktoru (třída ActorType), který reprezentuje. K tomuto typu budou v databázi uloženy funkce (ActorFunct), které definují, jaké povely lze jednotce zasílat. U jednotlivých objektů třídy ActorType pak budou v databázi uloženy objekty třídy Actor reprezentující konkrétní zařízení.

Díky této databázi bude možné spravovat jednotky v aplikaci. Pouhým přidáním nových záznamů do databáze bude také možné přidávat do aplikace nové aktory, jejich kategorie a rozšiřovat množinu funkcí.



Obrázek 3.7 Konceptuální diagram tříd pro databázi aplikace.

3.5 GUI

Aplikace bude mít jednotné GUI jak pro smartphony, tak pro tablety. Rozhraní bude muset být přizpůsobeno kapacitním displejům. Prostřednictvím GUI bude možné provádět přihlašování, ovládání aktorů a bude zobrazovat stav aktorů. Pro lepší orientaci v aplikaci budou aktory rozděleny do pokojů a scén. Scény se budou od pokojů lišit tím, že po otevření scény zašle aplikace pakety všem aktorům, které scéna obsahuje, s definovaným nastavením stavů. Takovéto chování umožní komfortní nastavování více aktorů najednou pomocí stisku jediného tlačítka. Grafické uživatelské

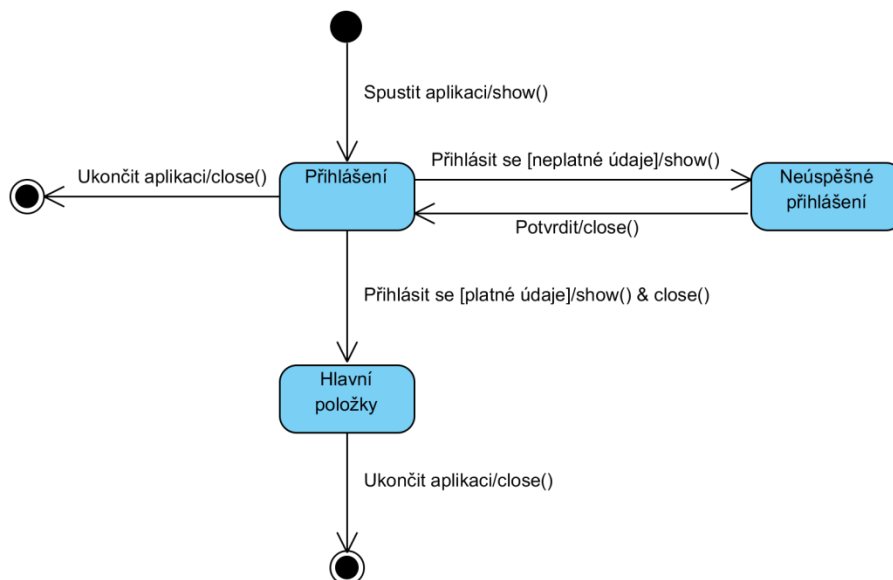
rozhraní navíc musí umožňovat správu pokojů, scén a aktorů. Tím je myšleno jejich přidávání, odebírání a editace.

3.6 Logika aplikace

V této kapitole je popsána logika aplikace pomocí stavových diagramů. Pro modelování byly vybrány její nejdůležitější části a pro přehlednost jsou rozděleny do čtyř podkapitol.

3.6.1 Přihlášení

Po startu aplikace bude potřeba provést přihlášení k jednotce eLAN-RF (obrázek 3.8). Přihlašovací údaje budou zadány do formuláře a potom zaslány jednotce. Pokud bude přihlášení úspěšné, provede se uložení přihlašovacích informací do databáze a aplikace přejde do stavu zobrazení hlavních položek, jako jsou scény a pokoje. Při neúspěšném přihlášení dojde k zobrazení chybového hlášení a po jeho potvrzení přejde aplikace zpět do stavu pro přihlášení. Ze stavů s přihlášením a hlavními položkami lze také přejít do konečného stavu a tím aplikaci ukončit.



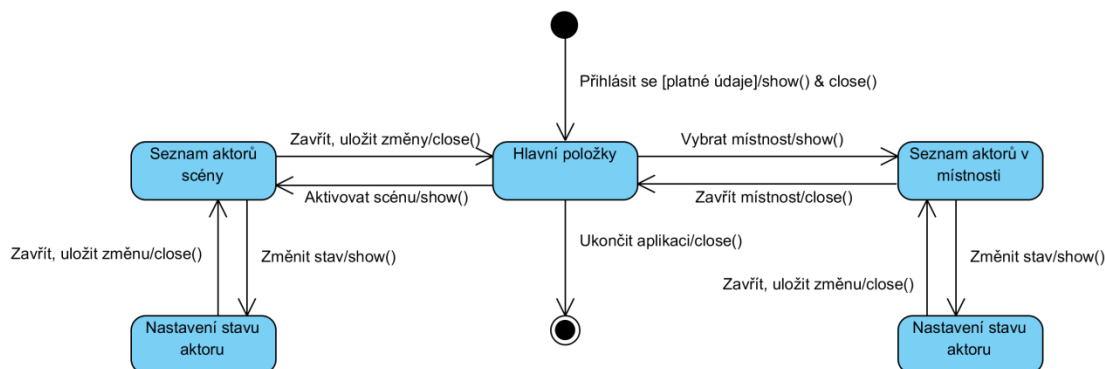
Obrázek 3.8 Diagram stavů pro přihlášení do aplikace.

3.6.2 Ovládání aktorů

Po přihlášení do aplikace se provede zobrazení výčtu hlavních položek, ze kterého je možné přejít do stavů pro ovládání aktorů (obrázek 3.9). Po otevření položky pokoje dojde k zobrazení všech aktorů přiřazených k této místnosti. Dále budou pomocí komunikace s jednotkou eLAN-RF načteny a následně zobrazeny jejich stavy. Pomocí výběru některého z nich přejde aplikace do nastavení stavu aktoru. V tomto nastavení je možné provést potřebné změny a ty následně uložit.

Po otevření položky se scénou dojde k zobrazení aktorů příslušejících scéně. Dále budou zaslány povely jednotce eLAN-RF pro nastavení stavů aktorů podle nadefinovaných hodnot. Pokud nastavení proběhne úspěšně, stav uvedený u zobrazených aktorů bude odpovídat stavu, který byl požadován. Po kliknutí na položku aktoru přejde aplikace do stavu jeho nastavení. V tomto nastavení je možné provést požadované změny a následně je uložit. Tímto způsobem bude možné změnit

nastavení všech aktorů uvedených v seznamu a následně uložit nastavení celé scény. Po příštím zapnutí scény se aktory nastaví na hodnoty, které byly naposledy uloženy.

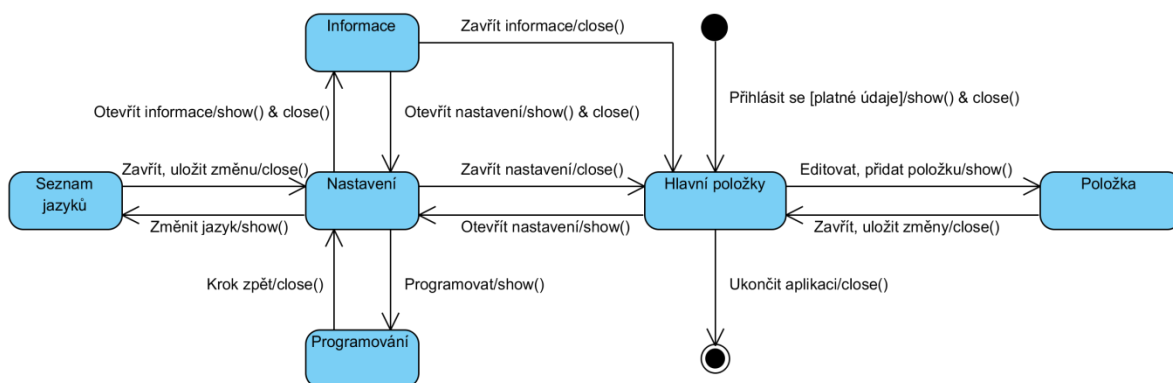


Obrázek 3.9 Diagram stavů pro ovládání aktorů.

3.6.3 Nastavení

Z hlavního rozdělení položek bude možné přejít do stavu pro úpravu a přidání těchto položek a také do menu s nastavením aplikace (obrázek 3.10). Ve stavu pro přidávání nebo editaci položek bude možné zadat název položky a přiřadit k ní ikonu. Ze stavu nastavení aplikace je možné provést přechod do následujících stavů:

- **Informace** – Stav sloužící pro zobrazení informací o jednotce eLAN-RF. Z tohoto stavu lze přejít do stavu s hlavním rozdělením položek nebo do nastavení.
- **Seznam jazyků** – V tomto stavu jsou zobrazeny všechny dostupné jazykové verze aplikace. Po vybrání jazyka se změní popisky v celé aplikaci.
- **Programování** – Z tohoto stavu je možné provádět programování aplikace (správa jednotek, pokojů a scén). Je možné navrátit se z něj zpět do nastavení. Podrobnější specifikace tohoto stavu je uvedena v následující kapitole.



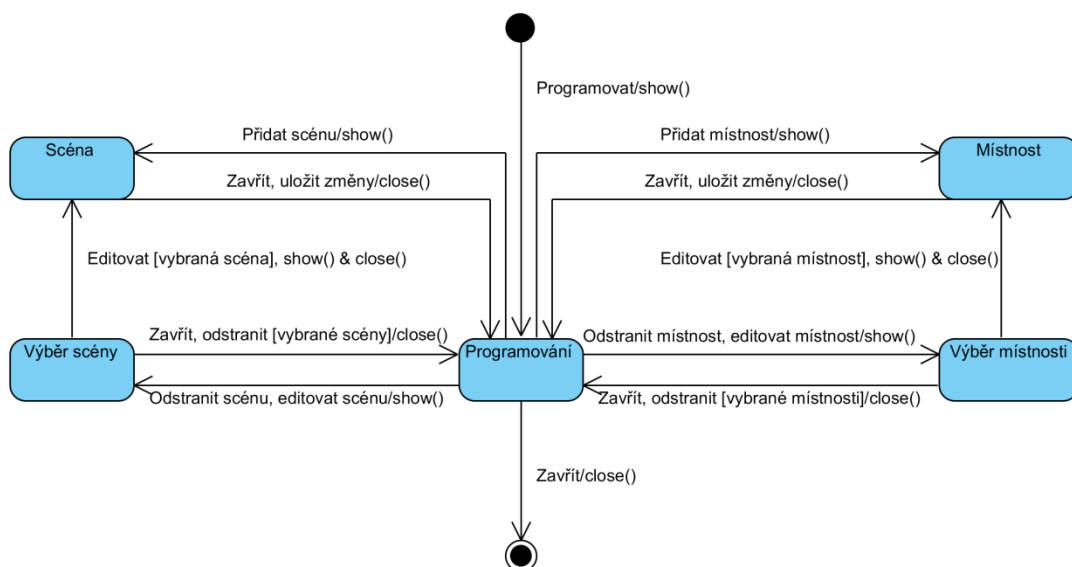
Obrázek 3.10 Diagram stavů pro nastavení aplikace.

3.6.4 Programování

Tato část aplikace bude umožňovat správu jednotek, jako jsou pokoje, scény a aktory. Do stavu programování lze přejít ze stavu pro nastavení aplikace. V diagramu na obrázku 3.11 jsou uvedeny přechody stavů pro správu scén a místností. Při volbě možnosti přidání místnosti, přejde aplikace do

stavu, ve kterém je možné zvolit její název a ikonu. Po uložení změn je tato místnost přidána a aplikace přejde zpět do stavu programování.

Pokud bude zvolena možnost editace nebo odstranění místnosti, aplikace přejde do stavu se seznamem místností. Pak je možné vybrat více místností a následně je odstranit nebo vybrat pouze jednu a přejít do stavu pro její editaci. Analogicky stejný systém slouží i pro správu scén.

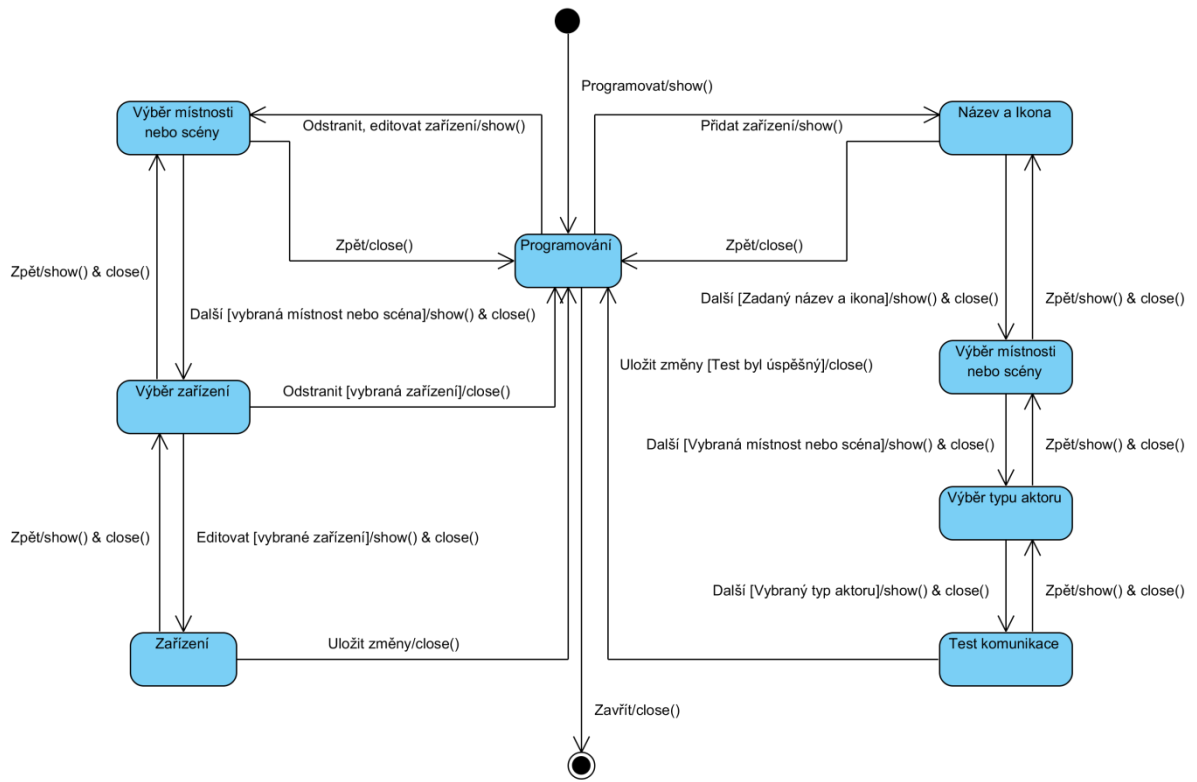


Obrázek 3.11 Diagram stavů pro správu scén a místností.

Správa zařízení je uvedena na obrázku 3.12. Pro přidání nového zařízení slouží následující sekvence stavů:

- **Název a ikona** – Stav sloužící pro zvolení názvu zařízení a výběr ikony.
- **Výběr místnosti nebo scén** – Slouží pro výběr místnosti nebo scén, ke které bude zařízení přiřazeno.
- **Výběr typu aktoru** – V tomto stavu se vybere typ přidávaného zařízení.
- **Test komunikace** – Stav sloužící pro zadání fyzické adresy zařízení a následné otestování komunikace. Pokud test proběhne v pořádku, zařízení je přidáno do databáze.

Pro odebrání nebo editaci zařízení přejde aplikace do stavu s výběrem místnosti nebo scén. Po výběru některé položky je proveden přechod na výběr zařízení. V tomto stavu je možné vybrat více zařízení, provést jejich odstranění a přejít do stavu programování nebo vybrat pouze jedno zařízení a přejít do stavu editace. V editaci je možné měnit jeho název, ikonu, místnost a adresu aktoru. Po uložení aplikace přejde zpět do stavu programování.



Obrázek 3.12 Diagram stavů pro správu zařízení.

4 Implementace

Tato část popisuje implementaci aplikace pro platformu Android. Implementace vychází z požadavků získaných analýzou zadání a z návrhu aplikace. V kapitole 4.1 jsou popsány technologie vývoje pro Android, použité nástroje a komponenty. Kapitola 4.2 se věnuje vlastní implementaci jednotlivých částí aplikace s využitím uvedených komponent a technologií.

4.1 Vývoj aplikace pro Android

Vývoj aplikací pro platformu Android probíhá pomocí modifikovaného programovacího jazyka Java. Jako vývojové prostředí se tedy dá použít jakékoliv IDE pro Javu, které má plugin pro vývoj Android aplikací. Pro vývoj této aplikace bylo použito prostředí *IntelliJ IDEA Community Edition*. Nezbytností je také Android SDK, který obsahuje nástroje potřebné pro překlad, ladění a testování aplikace.

4.1.1 Java

Jedná se o objektově orientovaný univerzální jazyk. Jeho koncept vychází z jazyků C a C++, ale má jinou organizaci. Zahrnuje některé idey z C, C++ a dalších jazyků. Díky tomu, že se jedná o jazyk se silnou typovou kontrolou, chyby vzniklé při vývoji lze přesně rozlišit na ty, které jsou detekovány už při kompilaci programu, a ty, které se objeví až při jeho běhu. V čase kompilace dochází k překladu programu na bytekód, který je nezávislý na platformě. Díky tomu jsou aplikace naprogramované v Javě přenositelné mezi platformami.

Java je jazyk s vysokou úrovní abstrakce, takže detaily, jako je strojová reprezentace kódu, nejsou při vývoji přístupné. Na rozdíl od C a C++ má automatickou správu paměti, takže ze strany programátora nedochází k chybám kvůli špatnému alokování a uvolňování paměti. O správu paměti se stará garbage collector. Jazyk také neumožňuje žádné nebezpečné konstrukce (přístup do pole bez kontroly indexu), které by mohly způsobit nezvyklé chování programu [30].

4.1.2 IntelliJ IDEA

IntelliJ IDEA je vývojové prostředí firmy *JetBrains*. Umožňuje vývoj projektů v různých jazycích, jako Java, Groovy nebo například PHP. Mezi její velké výhody patří real-time kontrola kódu, pokročilé možnosti refaktoringu a inteligentní napovídání kódu. Plugin pro vývoj aplikací na Android zahrnuje intuitivní grafický designer obrazovek, který obsahuje všechny základní grafické komponenty.

IntelliJ IDEA je k dispozici ve dvou verzích *Community Edition* a *Ultimate Edition*. Pro vývoj aplikace byla použita verze *Community Edition*, která je dostupná zdarma pod licencí Apache 2.0.

4.1.3 Aktivita

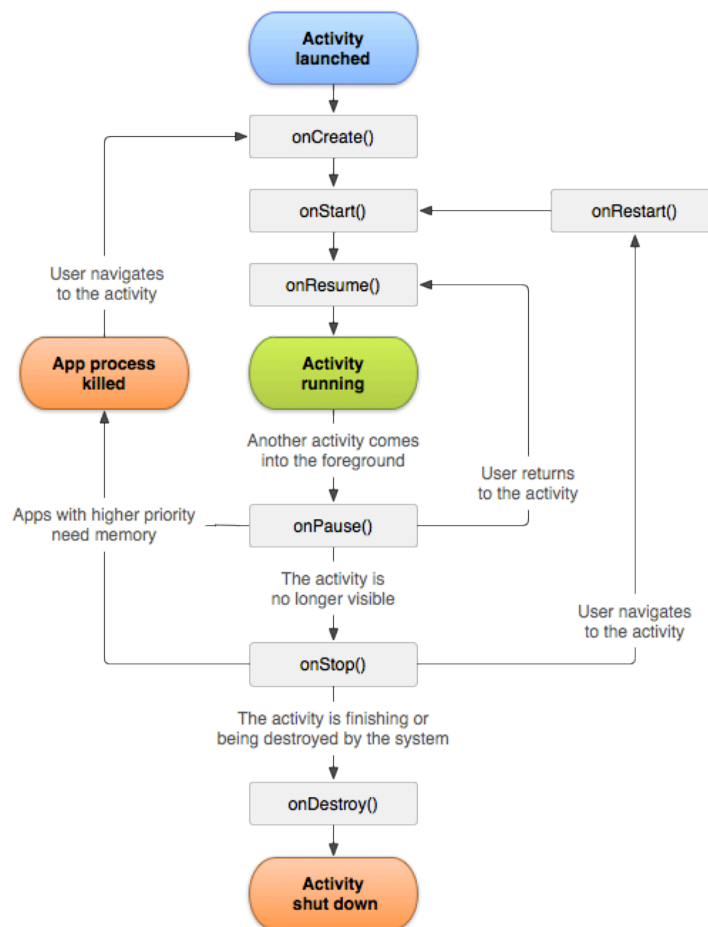
Základními komponentami všech Android aplikací jsou *aktivity*. Aktivita je komponenta uživatelského rozhraní, která většinou reprezentuje jednu obrazovku aplikace [30]. Grafický vzhled

aktivity je určen načteným layoutem, který obsahuje XML s definicí rozložení grafických prvků. Aktivity mají většinou formu full-screen okna, ale mohou být například i součástí okna jiné aktivity.

Aktivity jsou uspořádány v zásobníku. Po startu je aktivita umístěna na vrchol tohoto zásobníku a zobrazena na popředí displeje. Aktivita, která byla na vrcholu před tím, se posune v zásobníku o jednu úroveň níže a zůstane na pozadí. Aktivita se může se nacházet ve třech stavech:

- **Běžící aktivita** – V tomto stavu se aktivita nachází na vrcholu zásobníku, je zobrazená na popředí a uživatel s ní může interagovat.
- **Dočasně pozastavená aktivita** – Aktivita je stále vidět na obrazovce, ale nejde s ní interagovat, protože je nad ní transparentní aktivita, dialog nebo zamykací obrazovka zařízení. Dočasně zastavená aktivita stále existuje a může kdykoliv přejít do běžícího stavu. Zároveň ovšem může být kdykoliv zrušena systémem například kvůli nedostatku paměti.
- **Zastavená aktivita** – Do tohoto stavu se aktivita dostane, pokud jí překryje jiná aktivita a zaujme její pozici na vrcholu zásobníku. To se může stát například při přechodu na domácí obrazovku zařízení nebo při příchozím telefonním hovoru. Systém může kdykoliv takovouto aktivitu zrušit a odstranit z paměti, pokud je jí nedostatek [26].

Celý životní cyklus aktivity je vidět na obrázku 4.1. Ovály značí stavy aktivity a v šedých obdélnících jsou uvedeny názvy metod, které se volají, když aktivita přechází mezi stavy. Přetížení těchto metod lze využít pro vykonání akcí při přechodech aktivity mezi stavy.



Obrázek 4.1 Životní cyklus aktivity [32].

Životní cyklus aktivity obsahuje tři základní smyčky:

- **Celý cyklus** – Je to smyčka mezi voláním metody *onCreate()* a *onDestroy()*.
- **Viditelná aktivita** – Nachází se mezi voláním metod *onStart()* a *onStop()*. Aktivita je viditelná na obrazovce, ale nemusí být na popředí.
- **Aktivita na popředí** – Smyčka mezi voláním metody *onResume()* a *onPause()*. V této části životního cyklu je aktivita viditelná na popředí obrazovky a může interagovat s uživatelem. K volání těchto metod může docházet poměrně často (např. když se zařízení uspí), takže by neměly obsahovat výpočetně náročné akce.

4.1.4 Intenty

Intent je obecně definován jako „záměr“ vykonat nějakou práci. Je to v základu pasivní datová struktura, ve které je definován popis akce, která se má vykonat. Vedle definice akce mohou intenty obsahovat i další dodatečná data. Způsobů jejich použití je více:

- Broadcastové zaslání zprávy.
- Spuštění služby.
- Spuštění aktivity.
- Zobrazení webové stránky nebo seznamu kontaktů.
- Vytočení telefonního čísla nebo přijetí příchozího hovoru.

Intenty mohou být iniciovány buď aplikací nebo systémem (upozornění aplikace na příchod nové SMS) [31]. Existují dva typy intentů:

- **Explicitní** – Pro tyto intenty jsou jasně definovány komponenty, které na ně reagují. Intent potom nemusí obsahovat dodatečné informace o komponentě.
- **Implicitní** – Není pro ně specifikovaná žádná komponenta. Musí v sobě nést informace pro systém, aby věděl, kterou z dostupných komponent má pro daný intent zavolat [33].

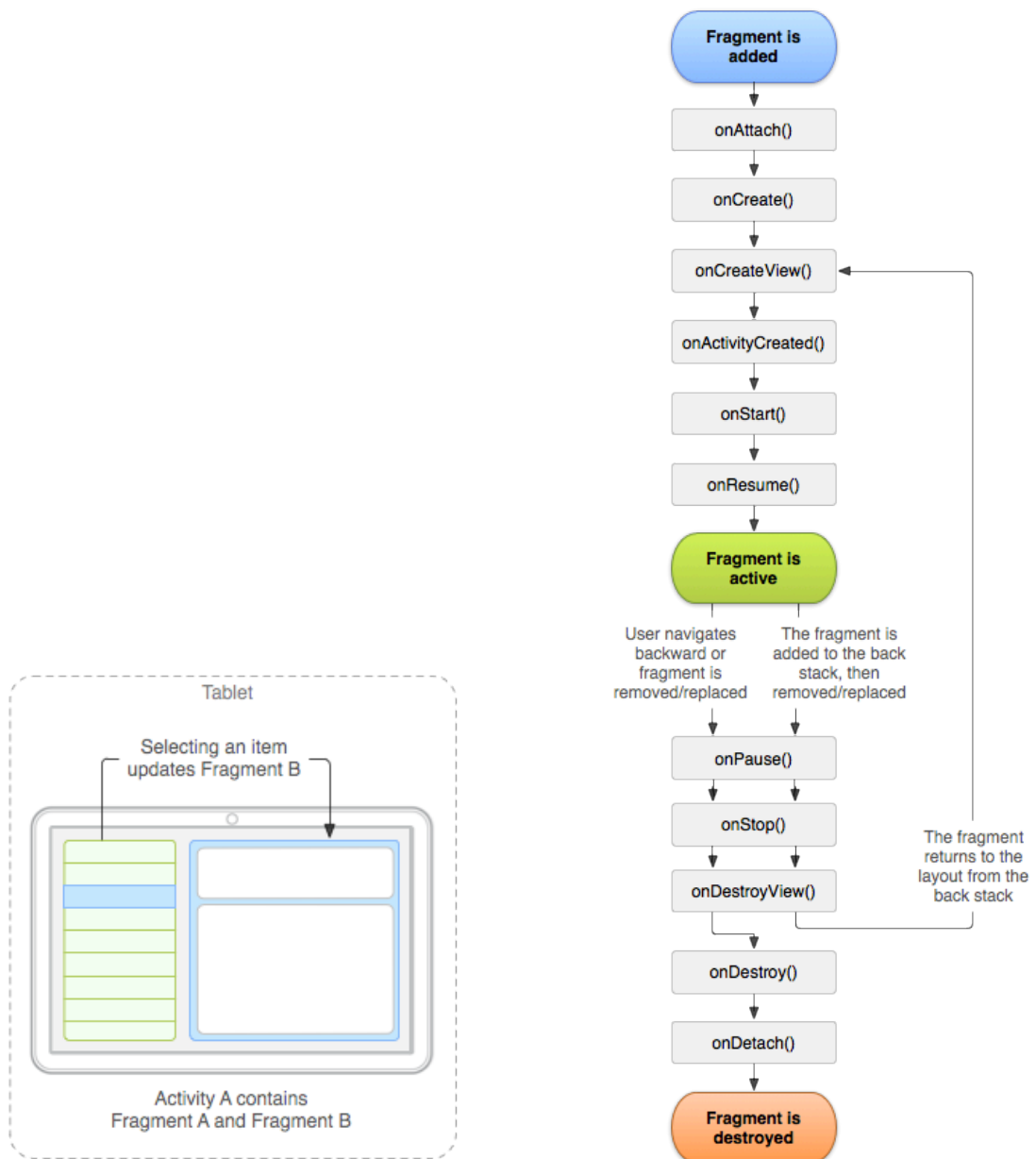
4.1.5 Fragmenty

Někdy je zapotřebí rozdělit obrazovku mobilního zařízení na více částí a definovat chování každé z nich zvlášť. Pro tuto situaci jsou v Androidu dostupné komponenty, které se jmenují fragmenty. Fragment je modulární část aktivity, která má svůj vlastní životní cyklus podobně jako aktivita. Fragmenty přijímají vstupní události a je možné je vytvářet a odstraňovat za běhu aktivity. Příklad aktivity s fragmenty je uveden na obrázku 4.2a, kde je zobrazena aktivita s dvěma fragmenty. V prvním z nich je seznam položek. Po vybrání položky se aktualizuje obsah druhého fragmentu.

Životní cyklus fragmentu je ovlivňován stavem hostitelské aktivity. Pokud je aktivita dočasně pozastavena, tak také všechny fragmenty, které obsahuje, jsou dočasně pozastaveny. Životní cyklus fragmentu je na obrázku 4.2b. Podobně jako aktivita přechází fragment do jednotlivých stavů. Mezi těmito přechody dochází k volání funkcí, které jsou označeny šedými obdélníky. Do těchto metod je možné naimplementovat vlastní akce. Důležitými metodami jsou:

- **onCreate()** – Metoda je zavolána při vytvoření fragmentu. Využívá se pro inicializaci proměnných, které existují po celou dobu životnosti fragmentu.

- **onCreateView()** – Tato metoda je zavolána, když je poprvé vykresleno uživatelské rozhraní fragmentu.
- **onPause()** – Systém zavolá tuto metodu pro indikaci, že uživatel opouští fragment.



Obrázek 4.2 a) Aktivita s fragmenty. b) Životní cyklus fragmentu [34].

4.1.6 ViewPager

ViewPager je layout, který umožňuje uživateli přecházet mezi jeho stránkami pomocí přetažení prstu po obrazovce z jedné strany na druhou. Stránky tohoto layoutu jsou generované pomocí PagerAdapteru. Jako stránky ViewPageru je možné použít fragmenty. Díky tomu je možné obsluhovat životní cyklus každé jeho stránky. Tato komponenta je obsažena v knihovně android-support-v4.jar [35].

4.1.7 AsyncTask

Android verze 3.0 a vyšší nepodporují síťovou komunikaci běžící v hlavním vlákne kvůli předcházení zablokování aplikace. Síťovou komunikaci je tedy potřeba provádět na pozadí aplikace. Pro tyto účely slouží komponenta AsyncTask, která provádí operace na pozadí a informuje o jejich stavu hlavní vlákno. Při použití AsyncTasku není potřeba ošetřovat volání a ukončování vláken a problémy s nimi spojené (uzamykání proměnných apod.). AsyncTask se skládá ze čtyř metod:

- **onPreExecute()** – Zavolá se před spuštěním běhu akcí na pozadí.
- **doInBackground()** – Provádí akce na pozadí aplikace.
- **onProgressUpdate()** – Informuje hlavní vlákno o průběhu akcí spuštěných na pozadí.
- **onPostExecute()** – Zavolá se po dokončení akcí běžících na pozadí.

4.1.8 SQLite

Pro mobilní zařízení je důležité využít pro ukládání dat efektivní mechanismus z důvodu jejich omezených paměťových kapacit. Pro vytváření a práci s databází na platformě Android je proto použita SQLite databáze. Jedná se o odlehčený typ relační databáze, který zajišťuje bezpečné a efektivní ukládání dat.

4.1.9 GUI

Tvorba GUI pro Android probíhá pomocí jazyka XML. Android obsahuje v základu komponenty pro definici uspořádání prvků na obrazovce (layouty) a různé jiné komponenty, jako jsou tlačítka, komponenty pro obrázky, textová pole, checkboxy, progressbary a další. Grafické uživatelské rozhraní je načítáno aktivitami, které ho zobrazí a definují funkce jeho jednotlivých prvků.

4.2 Výsledné řešení

Na základě návrhu aplikace byla provedena její implementace pro zařízení běžící na platformě Android. Podle vymodelovaných stavových diagramů byly vytvořeny jednotlivé obrazovky a logika přepínání mezi nimi. GUI je jednotné pro smartphone i tablet. V následujících kapitolách je uvedeno, jakým způsobem byly jednotlivé části naimplementovány a jaké komponenty k tomu byly zapotřebí.

4.2.1 Přihlašování a hlavní obrazovka

Po spuštění aplikace se jako první zobrazí přihlašovací aktivita (obrázek 4.3a). Pro přihlášení k jednotce eLAN-RF je nutné vyplnit jeho IP adresu, port a uživatelské jméno a heslo. Po stisku tlačítka pro přihlášení se pomocí objektu třídy AsyncTask vytvoří TCP spojení s jednotkou eLAN-RF. Pokud jsou adresa nebo port zadány špatně, spojení se nevytvoří a zobrazí se chybové hlášení.

V případě, že se spojení korektně vytvoří, odešle se RFIO Quick Control paket se zadanými údaji a adresou prvku nastavenou na hodnotu 0x000000. Objekt třídy AsyncTask potom čeká 5 sekund v neblokujícím čekání na odpověď od jednotky. Tento objekt je v průběhu tohoto čekacího cyklu čtyřikrát uspán, aby zbytečně nezatěžoval procesor. Pokud odpověď nepřijde, zobrazí se chybové hlášení. Jestliže ACK přijde, aplikace rozparsuje příchozí paket a z chybového kódu zjistí stav přihlášení. V případě, že bylo neúspěšné, je potřeba opravit přihlašovací údaje a přihlásit se znovu. Pokud proběhlo úspěšně, je zaslán Test Packet a z odpovědi na něj se získá verze hardwaru a

firmwaru, které se uloží do databáze. Potom je zobrazena hlavní aktivita aplikace s položkami scén a pokojů (obrázek 4.3b).

Na hlavní obrazovce jsou pokoje a scény uspořádány do matice 2x5 prvků. Scény jsou označeny piktogramem „S“. Celý tento vzhled je umístěn ve fragmentu a fragmenty ve ViewPageru, takže pokud je obrazovek s položkami více než jedna, lze mezi nimi přepínat pomocí přejetí prstu po obrazovce z jedné strany na druhou. Po otevření položky je zobrazeno menu ovládání aktorů, které je popsáno dále. Pokud se nad některou z položek podrží prst déle jak 2 sekundy, otevře se aktivita pro editaci dané položky (obrázek 4.3c).

V této aktivitě je možné změnit název položky a její ikonu. Ikony jsou uspořádány do matice 4x4 a jsou umístěny ve ViewPageru prostřednictvím fragmentů. Pokud jsou změny položky uloženy, změní se záznam v databázi a ViewPager na hlavní obrazovce se aktualizuje.



Obrázek 4.3 a) Přihlašovací obrazovka. b) Hlavní obrazovka. c) Editace položky.

4.2.2 Ovládání aktorů a scén

Po otevření položky pokoje na hlavní obrazovce se otevře aktivita se seznamem aktorů (obrázek 4.4b). Tato aktivita se ze začátku překryje transparentní aktivitou, která aktualizuje stavy jednotek (obrázek 4.4a). Tato aktualizací aktivita odesílá RFIO Controll pakety jednotlivým jednotkám pro zjištění jejich stavu. Pro vytvoření spojení a zaslání paketů používá AsyncTask s neblokujícím čekáním. Po odeslání paketu jednotce musí aktivita čekat sekundu na odpověď. Pokud odpověď přijde, aktivita aktualizuje stav jednotky. Po vypršení intervalu nebo po přijetí odpovědi zašle tato aktivita paket další jednotce a opět čeká sekundu na odpověď. Intervaly čekání jsou nevyhnutelné, protože v příchozích paketech není obsažena adresa aktoru, který ji zaslal. Pokud by tedy nebyl použitý ochranný interval a aktualizací aktivita by zaslala pakety jednotkám hned po sobě, nešlo by rozlišit, která odpověď patří ke které jednotce.

Po aktualizaci jednotek lze jejich stavy měnit kliknutím na položku v seznamu (zapnout/vypnout). Toto probíhá opět pomocí třídy AsyncTask a RFIO Controll paketů. Pro nastavení jednotky RGB slouží speciální aktivita (obrázek 4.4c). Disponuje velkým barevným terčem pro zvolení barvy RGB pásku a posuvníkem pro nastavení intenzity. Konfigurace se odešle jednotce pomocí kulatého tlačítka uprostřed palety barev.

Ovládání scén používá stejné obrazovky. Na začátku je ovšem místo aktualizace stavu jednotek provedeno zaslání konfiguračních paketů aktorům podle dříve nastavených hodnot. Díky tomu se

ihned po otevření scény aktory nastaví do požadovaných stavů. Následně je možné scénu upravovat tak, jak je popsáno výše pro aktory v pokoji. Pokud si uživatel přeje nastavit aktuální konfiguraci aktorů jako scénu, provede tak pomocí spodního tlačítka „Uložit“, které u seznamu jednotek scény nahrazuje tlačítko „Zpět do menu“, které je vidět na obrázku 4.4b. Při následném spuštění scény se aktory nastaví na naposledy uložené hodnoty.



Obrázek 4.4 a) Aktualizace stavů jednotek. b) Seznam jednotek. c) Nastavení RGB pásku.

4.2.3 Nastavení aplikace

Po stisknutí tlačítka pro nastavení se zobrazí aktivita s menu nastavení (obrázek 4.5a). Toto menu obsahuje položky pro změnu jazyka a pro programování aplikace. Při výběru změny jazyka je zobrazen seznam všech dostupných jazykových verzí aplikace (obrázek 4.5b). Po výběru některé z nich je změněn jazyk popisků v celé aplikaci.

Pomocí otevření záložky „Informace“ v liště záložek, která se nachází v horní části aktivity, jsou zobrazeny informace o připojené jednotce eLAN-RF (obrázek 4.5c), které byly získány při přihlášení do aplikace pomocí Test Packetu. Jedná se o informace o verzi hardwaru a firmwaru jednotky.

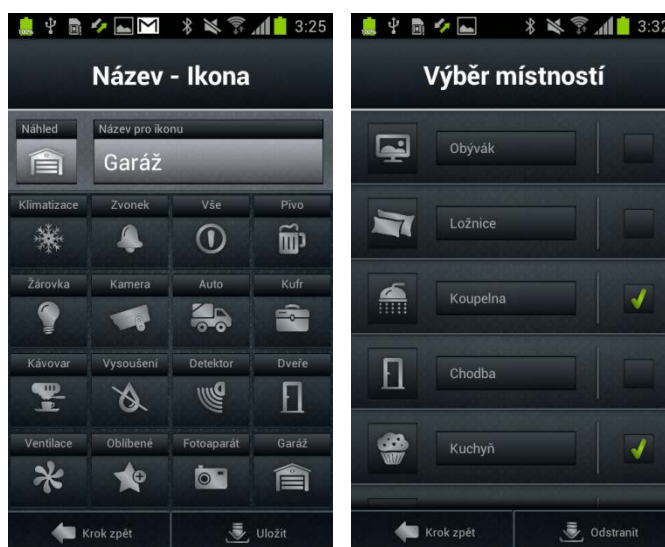


Obrázek 4.5 a) Menu nastavení. b) Výběr jazyka. c) Informace o eLAN-RF.

4.2.4 Správa místností a scén

Po výběru možnosti programování v hlavní nabídce nastavení je možné přejít na obrazovky sloužící pro správu pokojů a scén. Pro editaci a přidávání pokoje nebo scény slouží aktivita uvedená na obrázku 4.6a. Poskytuje možnost nastavení názvu a ikony položky. Ikony jsou uspořádány ve fragmentech v matici 4x4 a dají se posouvat pomocí ViewPageru.

Pro odstranění místností slouží druhá obrazovka (obrázek 4.6b). Pomocí checkboxů je možné vybrat místnosti, které mají být odstraněny. Po potvrzení se provede jejich odebrání z databáze včetně všech příslušných aktorů. Pro odebrání scén slouží stejná obrazovka lišící se pouze názvem.



Obrázek 4.6 a) Editace místnosti nebo scény. b) Odstranění místnosti.

4.2.5 Správa zařízení

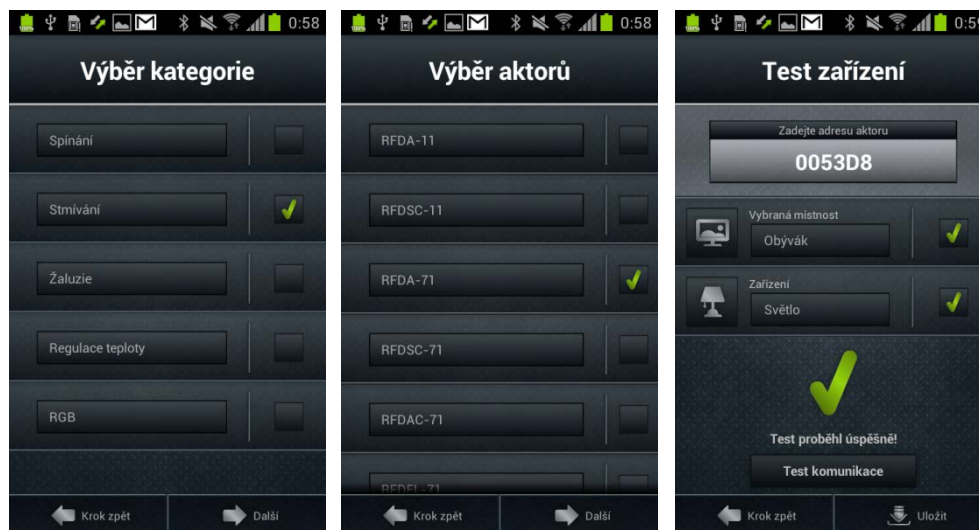
Z hlavního nastavení je možné prostřednictvím položky „Programování“ přejít na správu zařízení. Pro přidávání zařízení slouží sekvence aktivit, jejichž obrazovky jsou vidět na obrázcích 4.7 a 4.8. Při přidávání zařízení do aplikace je nejdřív spuštěna aktivita pro zadání názvu zařízení a výběr ikony, kde jsou ikony tradičně uloženy ve ViewPageru. Následuje aktivita pro výběr pokoje nebo scény, ke které bude zařízení přiřazeno.



Obrázek 4.7 a) Nastavení názvu a ikony zařízení. **b)** Volba místnosti nebo scény, do které bude aktor zařazen.

Aby byla aplikace schopná se zařízením komunikovat a znala jeho dostupné funkce, je potřeba přiřadit tomuto zařízení typ. Pro tento účel slouží dvě obrazovky (obrázek 4.8a, b). V první z nich je zvolena kategorie aktoru (spínací, stmívací, žaluziový, teplotní a RGB). Po výběru kategorie je potřeba zvolit konkrétní typ aktoru.

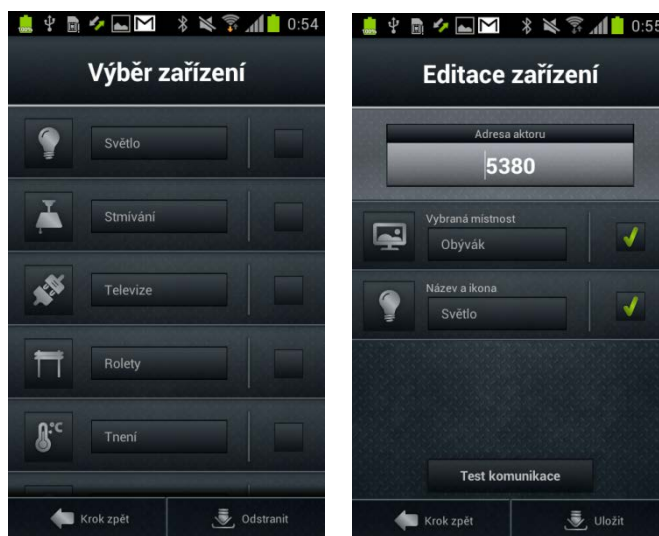
Po nastavení všech potřebných informací prostřednictvím výše uvedených obrazovek je potřeba otestovat komunikaci se zařízením. K tomuto účelu slouží aktivita na obrázku 4.8c. Do textového pole se zadá adresa prvku a stiskem tlačítka pro testování je aktoru zaslán testovací RFIO Quick Control paket. Pokud nepřijde odpověď, je tento paket zaslán znovu. Maximálně však 5x za sebou. Pokud přijde odpověď od aktoru, zařízení je uloženo do databáze. V opačném případě vyskočí při pokusu o uložení chybové hlášení.



Obrázek 4.8 a) Výběr kategorie aktoru. **b)** Výběr typu aktoru. **c)** Test komunikace.

Mimo přidávání nových zařízení do aplikace je taky možné upravit nebo odebrat zařízení, která se v ní již nacházejí. Pro tyto účely slouží aktivity, které se nacházejí na obrázku 4.9. První z nich slouží pro vybrání zařízení pomocí checkboxů a jejich následné odebrání z databáze.

Druhá je určená pro editaci zařízení. Je možné v ní změnit adresu aktoru a pomocí výběru některé z položek i název, ikonu a přiřazenou místnost. Tyto úpravy jsou pak prováděny pomocí aktivit pro výběr místnosti a změnu názvu a ikony. Tyto aktivity byly uvedeny v textu již dříve.



Obrázek 4.9 a) Odstranění zařízení. b) Editace zařízení.

5 Testování

Testování celé aplikace bylo prováděno průběžně po celou dobu vývoje a probíhalo na zařízeních Samsung Galaxy S2 a Samsung Galaxy Tab. Pro testování byly použity nástroje z monitorovacího serveru DDMS a dále služba ACRA.

5.1 ACRA

ACRA (Application Crash Report for Android) je služba, která zachytává výjimky v testované aplikaci a ošetřuje padání aplikace. O všech těchto událostech zasílá informace na definovanou e-mailovou adresu. Vývojář má díky tomu informace o tom, z jakého důvodu je aplikace nestabilní a může odstranit řadu chyb.

Velkou výhodou ACRA je volná licence Apache 2.0, takže tato služba je zdarma. Pro její použití stačí přidat do aplikace speciální knihovnu a do podtřídy třídy Application přidat řetězec, který se zobrazí na displeji při pádu aplikace, a e-mailovou adresu, na kterou mají být zasílána hlášení o chybách. V testovacím zařízení potom musí být správně nakonfigurovaný e-mailový klient.

5.2 Metoda white-box

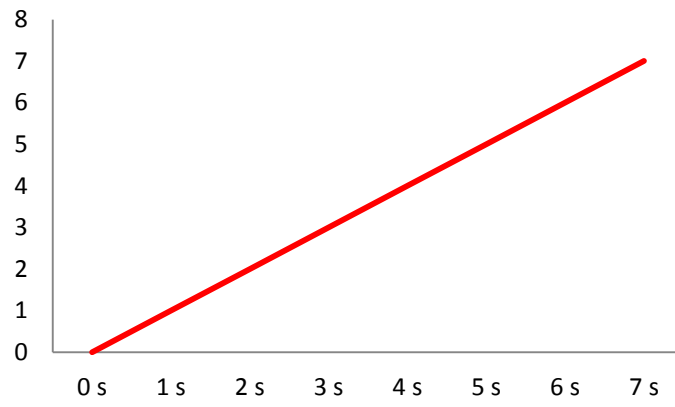
Při testování aplikace byla uplatněna metoda white-box. Jedná se o metodu, kdy tester zná strukturu aplikace a ví, jak byla naimplementována. Při tomto testování byly sledovány životní cykly aktivit a správné ukončování vláken a akcí běžících na pozadí. Dále byla pomocí nástroje SQLite Database Viewer zkontrolována struktura vytvořené databáze. Při těchto testech bylo opraveno velké množství chyb. Jako velmi užitečný nástroj se při testování ukázala konzole Logcat.

5.3 Testování ochranného intervalu

Jak již bylo uvedeno dříve, při aktualizaci stavů jednotek je potřeba používat ochranný interval. Hodnota tohoto intervalu se v průběhu vývoje několikrát měnila. Velmi záleželo na použitých aktorech, protože u některých byla doba odeslání odpovědi na správu delší než u jiných. Kratší interval znamená rychlejší načítání stavů, delší zase větší přesnost. Z hlediska aplikace je důležitější přesnost, aby měl uživatel korektní informace o stavu jednotek v domě. Hodnota tedy byla nakonec nastavena na 1 sekundu, protože při této délce nebyly zaznamenány chyby při načítání stavů.

Na obrázku 5.1 je uvedena doba načítání stavů v závislosti na počtu jednotek v systému. Tato závislost je lineární, takže přidání každé jednotky prodlouží dobu načítání o 1s. Pro malý počet aktorů v místnosti (standartní situace), je doba načítání únosná.

Tomuto problému by značně pomohla úprava komunikačního protokolu. Kdyby příchozí pakety od jednotek obsahovaly jejich adresy, bylo by načítání stavů téměř okamžité.



Obrázek 5.1 *Závislost doby aktualizace stavů na počtu jednotek.*

6 Závěr

Bezdrátové elektroinstalace zatím nedosahují kvalit sběrnicových, ale jejich vývoj jde neustále kupředu. Poskytují nejlepší řešení v domech, kde by zavedení sběrnicové elektroinstalace znamenalo vysoké náklady z důvodu nutnosti zabudování kabelů. Aplikace pro tablety a chytré telefony jenom přispěje ke zdokonalení systému iNELS RF Control. Přinese do bezdrátové elektroinstalace pohodlí a možnost ovládat dům centrálně z jednoho místa. Uživatel již nebude potřebovat množství ovladačů, protože bude moci svoji instalaci komfortně ovládat prostřednictvím telefonu. V současné době, kdy trh mobilních zařízení začínají ovládat smartphony, by mohla taková aplikace získat u uživatelů velkou oblibu.

V této práci byla uvedena problematika týkající se inteligentních elektroinstalací a zvláště do detailu pak byl popsán systém bezdrátové elektroinstalace s označením iNELS RF Control firmy ELKO EP, s.r.o. včetně popisu dostupných jednotek a možností komunikace. Pro dobré pochopení všech technik a programovacích principů zde bylo uvedeno i velké množství informací o platformě Android. Zároveň byl proveden i průzkum trhu, aby se zjistilo, zda může být aplikace pro mobilní zařízení, jako jsou smartphony a tablety, úspěšným projektem.

Byla provedena analýza požadavků na aplikaci pro ovládání bezdrátové elektroinstalace a na základě ní vznik návrh. Pro lepší pochopení návrhu bylo použito modelování aplikace. Část implementace popisuje některé systémové komponenty, které byly při vývoji použity a dále podrobně specifikuje výslednou aplikaci a princip její funkce. Část testování se věnuje použitým testovacím nástrojům a technikám. Dále popisuje možnost budoucího vylepšení aplikace.

Aplikace pro chytrý telefon a tablet s operačním systémem Android, která je produktem této práce, umožňuje monitorovat a ovládat jednotky v bezdrátové elektroinstalaci iNELS RF Control. Dále je pomocí ní možné nastavovat různé scény. Díky zvolenému návrhu databáze je aplikace velmi modulární. Bude do ní možné bez problémů přidávat nové jednotky s novými funkcemi bez nutnosti její programové změny. Tímto byly splněny všechny cíle stanovené v zadání práce.

Společnost ELKO EP, s.r.o má v plánu dále pracovat na vývoji této aplikace a systému komunikace s bezdrátovou elektroinstalací prostřednictvím jednotky eLAN-RF. V této době vyvíjí jednotku eLAN-RF-Wi-003, která bude moci komunikovat s mobilním zařízením pomocí wi-fi sítě. Přístupový bod (AP), který se používal pro směrování Wi-Fi signálu na LAN signál už tedy nebude potřeba. Velký krok dopředu by mohly přinést také úpravy komunikačního protokolu. Díky tomu by bylo možné aplikaci značně urychlit.

Chytré domy a inteligentní elektroinstalace už nejsou jenom hudbou budoucnosti, jako to bývalo kdysi. Tyto pojmy se čím dál více dostávají do povědomí všech lidí a aplikace, jejímž vývojem se zabývá tato práce, může zase posunout toto odvětví o krok dále.

Literatura

- [1] HARPER, Richard. *Inside the smart home*. Editor Richard Harper. London: Springer, c2003 [cit. 2013-05-13], 264 s. ISBN 18-523-3688-9.
- [2] BERLO, Ad. Smart home technology: Have older people paved the way? *Gerontechnology* [online]. 2002, roč. 2, č. 1, s. 77-87 [cit. 2013-05-12]. DOI: <http://dx.doi.org/10.4017/gt.2002.02.01.010.00>. Dostupné z: <http://gerontechnology.info/index.php/journal/article/view/gt.2002.02.01.010.00/94>
- [3] LITTLE, Arthur D. Catching the Smart Home Opportunity: Room for Growth for Telecom Operators. In: *Arthur D. Little* [online]. 2012 [cit. 2013-05-12]. Dostupné z: http://www.adlittle.com/downloads/tx_adlreports/ADL_TIME_2012_Smart_Home_Opportunity.pdf
- [4] BOUCHARD, Kevin, Bruno BOUCHARD a Abdenour BOUZOUANE. *Guidelines to efficient smart home design for rapid AI prototyping* [online]. 2012 [cit. 2013-05-13]. DOI: 10.1145/2413097.2413134. Dostupné z: http://dl.acm.org/ft_gateway.cfm?id=2413134&ftid=1324494&dwn=1&CFID=215447090&CFTOKEN=77889514
- [5] VALEŠ, Miroslav. *Inteligentní dům*. Brno: ERA group spol. s r. o., 2006 [cit. 2013-05-13]. ISBN 80-7366-062-8. s. 1-29.
- [6] ELKO EP, s.r.o. INELS: *Váš dům Vás má rád* [online]. 2007 [cit. 2013-05-13]. Dostupné z: http://www.elkoep.cz/downloads/promotion_materials/iNELS_pro_laiky_02.pdf
- [7] ROE, Patrick R. W. *Towards an inclusive future* [online]. Brusel: COST, 2007 [cit. 2013-05-14]. ISBN 92-898-0027. s. 110-124. Dostupné z: http://www.tiresias.org/cost219ter/inclusive_future/inclusive_future_book.pdf
- [8] Inteligentní domy: Současnost, ne budoucnost. *Právo: příloha Bydlení*. 23.2.2013, s. 15. Dostupné z: <http://www.inels.cz/ke-stazeni/napsali-o-nas?subaction=download&did=124>
- [9] Novinky v inteligentních elektroinstalacích. *Sdělovací technika: Telekomunikace-elektronika-multimedia*. 28.2.2011, roč. 2011, č. 3, s. 40. Dostupné z: <http://www.floowie.com/cs/cti/sdelovaci-technika032011/#/strana/40/>
- [10] ELKO EP, s.r.o. *Přehled sortimentu: RELÉ / iNELS / iNELS RF Control / LOGUS 90 / ETI*. 1. vyd. Holešov, 2012. Dostupné z: http://www.elkoep.cz/downloads/promotion_materials/Prehled_sortimentu.pdf
- [11] INELS: Jednou nohou doma. *Právo: Inzerce*. 12.2.2013, s. 1-2. Dostupné z: <http://www.inels.cz/ke-stazeni/napsali-o-nas?subaction=download&did=124>
- [12] RF - Control - systém bezdrátového ovládání. *A-Z ELEKTRO* [online]. 2011, roč. 2, květen/červen, s. 15 [cit. 2013-05-21]. Dostupné z: http://www.azcasopis.cz/casopis/az_casopis_3_2011/index.html
- [13] Soutěž o Zlatý Amper 2011. *Automa: Časopis pro automatizační techniku* [online]. 2011, č. 6, s. 56-58 [cit. 2013-05-21]. Dostupné z: <http://www.odbornecasopisy.cz/res/pdf/43763.pdf>

- [14] ELKO EP, s.r.o. *Bezdrátová inteligentní elektroinstalace*. 1. vyd. Holešov, 2013. Dostupné z:
http://www.elkoep.cz/downloads/promotion_materials/Katalog_Inels_RF.pdf
- [15] JABLOTRON ALARMS A.S. *JA-80P bezdrátový PIR detektor pohybu osob*. Jablonec nad Nisou. Dostupné z:
<http://www.jablotron.com/DownloadHandler.aspx?method=GetFileDownload&fileID=209&DontParse=true>
- [16] JABLOTRON ALARMS A.S. *JA-81M bezdrátový detektor otevření a univerzální vysílač*. Jablonec nad Nisou. Dostupné z:
<http://www.jablotron.com/DownloadHandler.aspx?method=GetFileDownload&fileID=1082&DontParse=true>
- [17] JABLOTRON ALARMS A.S. *JA-82M bezdrátový detektor otevření - magnetický*. Jablonec nad Nisou. Dostupné z:
<http://www.jablotron.com/DownloadHandler.aspx?method=GetFileDownload&fileID=218&DontParse=true>
- [18] ELKO EP, s.r.o. *Návod eLAN-RF-003*. Holešov, 2012. Dostupné z:
http://www.elkoep.cz/fileadmin/produkty/Elko/RfX/eLAN-RF-003/CS/Manual_eLAN-RF-003_cz.pdf
- [19] Mobile device market to reach 2.6 billion units by 2016: Smart phones and tablet PCs will continue to impact shipments of other phones and notebook PCs. In: *Canalyst* [online]. 22.2.2013 [cit. 2013-05-22]. Dostupné z:
<http://www.canalys.com/newsroom/mobile-device-market-reach-26-billion-units-2016>
- [20] RUBIN, Andy. Where's my Gphone?. In: *Google: Official Blog* [online]. 5.11.2007 [cit. 2013-05-23]. Dostupné z:
<http://googleblog.blogspot.cz/2007/11/wheres-my-gphone.html>
- [21] JORDAN, Lucas a Pieter GREYLING. *Practical Android projects*. New York: Distributed by Springer Science Business Media, 2011, 404 s. ISBN 14-302-3243-9.
- [22] MEIER, Reto. *Professional Android 2 application development*. Indianapolis: Wiley, c2010, 543 s. Wrox programmer to programmer. ISBN 978-0-470-56552-0.
- [23] STEELE, James a Nelson TO. *The Android developer's cookbook: building applications with the Android SDK*. Upper Saddle River, NJ: Addison-Wesley, 2011, 339 s. Developer's library. ISBN 978-0321741233.
- [24] Q4, 2012 Smartphone Market shares- IDC [graf]. In: *Ronnie05's Blog* [online]. 15.2.2013 [cit. 2013-05-23]. Dostupné z:
<http://ronnie05.files.wordpress.com/2013/02/smartphone-oss-ii.jpg>
- [25] Platform Versions [graf]. In: *Android Developers: Dashboards* [online]. 1.5.2013 [cit. 2013-05-24]. Dostupné z:
<http://chart.googleapis.com/chart?chs=500x250&cht=p&chf=bg%2Cs%2C00000000&chd=t%3A1.8%2C3.7%2C38.5%2C0.1%2C27.5%2C28.4&chco=c4df9b%2C6fad0c&chl=Eclair%7CFroyo%7CGingerbread%7CHoneycomb%7CIce%20Cream%20Sandwich%7CJelly%20Bean>
- [26] ZECHNER, Mario. *Beginning Android 4 games development*. Berkeley: Apress, 2011, 677s. ISBN 978-1-4302-3987-1.

- [27] Android Architecture: The Key Concepts of Android OS [obrázek]. In: *Android-App-Market.com* [online]. 17.2.2012 [cit. 2013-05-24]. Dostupné z: <http://www.android-app-market.com/wp-content/uploads/2012/02/Android-architecture.jpg>
- [28] Tools Help. GOOGLE. *Android Developers* [online]. c2012 [cit. 2013-05-25]. Dostupné z: <http://developer.android.com/tools/help/index.html>
- [29] *Programming Android*. 1. ed. Sebastopol: O'Reilly, 2011, 482 s. ISBN 978-1-449-38969-7.
- [30] Java Language Specification. ORACLE. *ORACLE* [online]. [cit. 2013-05-28]. Dostupné z: <http://docs.oracle.com/javase/specs/jls/se7/html/jls-1.html>
- [31] KOMATINENI, Satya, Dave MACLEAN a Eric FRANCHOMME. *Pro Android 4*. New York: Distributed to the book trade worldwide by Springer Science Business Media, 2012, 991 s. ISBN 978-143-0239-307.
- [32] Activity. In: *Android Developers* [online]. 2013 [cit. 2013-05-28]. Dostupné z: <http://developer.android.com/reference/android/app/Activity.html>
- [33] Intent. In: *Android Developers* [online]. 2013 [cit. 2013-05-28]. Dostupné z: <http://developer.android.com/reference/android/content/Intent.html>
- [34] Fragments. In: *Android Developers* [online]. 2013 [cit. 2013-05-28]. Dostupné z: <http://developer.android.com/guide/components/fragments.html>
- [35] ViewPager. In: *Android Developers* [online]. 2013 [cit. 2013-05-28]. Dostupné z: <http://developer.android.com/reference/android/support/v4/view/ViewPager.html>
- [36] AsyncTask. In: *Android Developers* [online]. 2013 [cit. 2013-05-28]. Dostupné z: <http://developer.android.com/reference/android/os/AsyncTask.html>

Obsah přiloženého CD

./Technicka_zprava.pdf

./Technicka_zprava.docx

Součástí přiloženého CD nejsou zdrojové kódy ani přeložená aplikace, protože podléhají autorským právům společnosti ELKO EP, s.r.o.