

Integracija odprtokodnih rešitev za avtomatizirano daljinsko upravljanje in nadzor zgradb

Nejc Šmid¹, Iztok Humar²

¹Mediainteractive, Beleharjeva 2, 4208 Šenčur, Slovenija

²Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana, Slovenija

E-pošta: iztok.humar@fe.uni-lj.si

Povzetek. V prispevku predstavimo izdelavo celovitega in cenovno ugodnega sistema za avtomatizirano daljinsko upravljanje in nadzor zgradb, ki integrira prodorne odprtokodne projekte s tega področja. Takšen pristop omogoča uporabo uveljavljenih protokolov za nadzor zgradb, krmiljenje že razvitih naprav za upravljanje, pa tudi uporabo namenske strojne opreme, ki je danes dostopna na trgu. Odprtokodni sistem podpira žično in brezžično komunikacijo, prek katere centralni nadzorni in upravljalški del komunicira s senzorji in aktuatorji. Uporabnik lahko sistem krmili prek spletnega ali posebej prirejenih vmesnikov za sodobne mobilne naprave, omogočeno pa je tudi upravljanje na podlagi vnaprej pripravljenih scenarijev. Na konkretnem primeru prikažemo merjenje temperature, vlage in osvetljenosti ter upravljanje grelnih teles, razsvetljave in rolojev ob sočasnem zajemanju zunanjih podatkov o vremenski napovedi. Poleg nizke cene sta dodatni prednosti takšnega sistema majhna poraba energije in odprtost za nadaljnje nadgradnje.

Ključne besede: avtomatizirano daljinsko upravljanje in nadzor, zgradbe, odprtokodne rešitve

Integration of open source solutions for automated remote management and control of buildings

The paper presents a complete and affordable open-source system for automated remote management and control of buildings. The developed system enables using the well-adopted protocols for control of buildings and integration of devices and hardware already developed and available on the market. The system supports using the wire and wireless communication between the central control and management on one side and sensors and actuators on the other. The user can control the system by using either the web interface or interfaces specifically developed for modern mobile devices. Scenarios-based management is also enabled. An example is given of measuring the temperature, humidity and illumination as well as controlling the heating elements, lighting and shutters at simultaneous capturing the weather data. Besides the low price, low power consumption and openness to further improvements another two advantages of the system are.

1 UVOD

Avtomatizirano daljinsko upravljanje in nadzor postajata pomembna dejavnika pri ustvarjanju prijaznejšega in varnejšega bivalnega, delovnega in industrijskega okolja ter zmanjševanju rabe energije in stroškov.

Področje avtomatiziranega daljinskega upravljanja in nadzora zgradb danes spada v domeno velikih poslovnih ali nadstandardnih bivalnih objektov, kar je posledica relativno visokih stroškov začetne investicije. Razlog za to je v poslovnem modelu podjetij, ki se ukvarjajo z

izdelavo strojne in programske opreme za krmiljenje in nadzor zgradb. Zaprti komunikacijski protokoli (angl. proprietary protocols) ter namensko naravnana in posledično draga strojna oprema kažejo na poslovni model, usmerjen v zaprt trg. Širšo uporabo preprečujejo tudi razpršeni standardi, ki se osredinjajo zgolj na posamezne podsisteme (krmiljenje, avdio/video, televizija, varnost itd.) in ne na celovit sistem.

Našteti razlogi zavirajo bolj množično razširjenost tovrstnih sistemov in omejujejo rast celotne panoge. Zaradi omejene konkurence so cene celovitih sistemov za avtomatizirano daljinsko upravljanje in nadzor zgradb relativno visoke, cenovno pa še posebej izstopajo uporabniški vmesniki (zaslonski na dotik) in prehodi med različnimi protokoli za krmiljenje in nadzor zgradb ter uveljavljenimi telekomunikacijskimi protokoli (TCP/IP), ki prevzemajo vlogo komunikacijske hrbtnice tudi v primeru sodobnih sistemov za daljinsko upravljanje in nadzor zgradb.

Zaradi prodora novih tehnologij in hitrega razvoja odprtokodnih rešitev je zdaj mogoče sistem za avtomatizirano daljinsko upravljanje in nadzor zgradb zgraditi po bistveno nižji ceni. Poleg nižjih stroškov začetne investicije omogoča pristop, prikazan v nadaljevanju tega članka, tudi odprtost za nadaljnje nadgradnje in majhno porabo energije. V zadnjem času smo namreč priča razmahu pametnih telefonov, dlančnikov, miniaturnih računalnikov in drugih cenovno ugodnih namenskih izdelkov, ki so povezani v omrežje

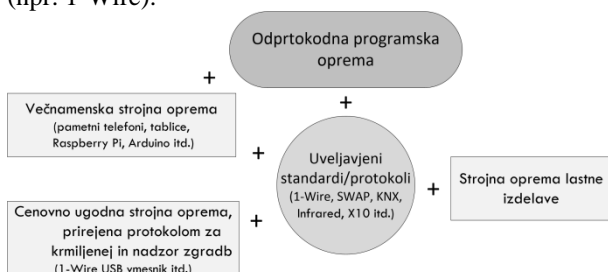
IP in prevzemajo vlogo uporabniških vmesnikov tudi pri sistemih za daljinsko upravljanje in nadzor zgradb, neodvisno od lokacije uporabnika.

V nasprotju z zaprtimi in cenovno težko dostopnimi rešitvami za avtomatizirano daljinsko upravljanje se v zadnjem času razvijajo prosto dostopne odprtokodne rešitve, ki omogočajo integracijo že obstoječih tehnologij in izvedbo cenениh ter inovativnih sistemskih pristopov tako na ravni programske kot strojne opreme. Cilj tovrstnih pristopov je ponuditi možnosti povezovanja najrazličnejših električnih naprav oziroma podsistemov na celovit centralni sistem z najboljšim mogočim izkoristkom posamezne naprave oziroma podsistema. Zaradi proste dostopnosti odprtokodnih rešitev se povečuje njihova razširjenost med uporabniki, kar zagotavlja hitrejši razvoj panoge.

2 KONCEPT ODPRTOKODNEGA SISTEMA

Odprtokodni sistemi za avtomatizirano daljinsko upravljanje in nadzor zgradb se v osnovi osredinjajo na izdelavo programske opreme, ki tvori upravljalno jedro sistema. Praviloma gre za brezplačno programsko opremo, katere izvorna koda je javno dostopna. Centralno programsko jedro omogoča povezovanje z različnimi že uveljavljenimi protokoli za krmiljenje in nadzor zgradb (npr. 1-Wire, KNX, Z-Wave) ter drugimi komunikacijskimi protokoli (npr. TCP/IP, HTTP). Povezovalni vidik omogoča izbiro različnih protokolov in s tem prilagajanje sistema glede na potrebe.

Odprtost odprtokodne programske opreme prinaša možnosti za uporabo palete izvajalnih okolij in tudi podporo že uveljavljeni strojni opremi. Tovrstna strojna oprema je zaradi svoje razširjenosti zanesljiva, cenovno ugodna, ponuja pa tudi široke možnosti v luči uporabe. Njen dober primer so vse bolj razširjeni pametni telefoni in dlančniki z velikimi zasloni na dotik, medtem ko so za delovanje centralnega sistema in različnih podsistemov primerni majhni, cenovno ugodni in energetske učinkoviti miniaturni računalniki in odprtokodne mikrokrmilne platforme (npr. Arduino). Obstaja tudi možnost izdelave lastne strojne opreme v kombinaciji z že izdelanimi čipi, ki komunicirajo z obstoječimi protokoli za krmiljenje in nadzor zgradb (npr. 1-Wire).



Slika 1: Koncept odprtokodnega sistema za avtomatizirano upravljanje zgradb

Pri svojem delu smo si zadali cilj, da postavimo sistem, ki bo zadostil naslednjim zahtevam:

- nizka cena (koncept odprtokodnega sistema),

- širok nabor funkcionalnosti,
- možnost integracije uveljavljenih protokolov za krmiljenje in nadzor zgradb,
- uporaba klasičnih naprav za upravljanje in že uveljavljene strojne opreme (pametni telefoni, tablice, Arduino itd.),
- majhna energijska poraba sistema,
- uporaba žične in brezžične komunikacije,
- možnost nadaljnega razvoja in nadgradnje.

3 KOMPONENTE ODPRTOKODNEGA SISTEMA

V tem delu predstavimo komponente – odprtokodne gradnike – za izvedeno rešitev odprtokodnega sistema za avtomatizirano daljinsko upravljanje in nadzor zgradb.

3.1 OpenRemote

Osrednji del sistema temelji na odprtokodnem projektu OpenRemote [2]. Njegove glavne prednosti pred drugimi podobnimi sistemi so: dobro razvito spletno orodje za konfiguracijo naprav in izdelavo uporabniškega vmesnika, podpora širokemu krogu protokolov ter dobra dokumentacija, podprta z obsežnim forumom. Projekt OpenRemote je dobitnik nagrade Bossie 2012 [4] za eno najboljših odprtokodnih programskih oprem.

Cilj projekta je zagotoviti odprtokodno programsko opremo za gradnjo osrednjega nadzornega in upravljalnega sistema, ki omogoča povezovanje z obstoječimi sistemi in protokoli. Sistem ni omejen na namenske naprave specifičnih izdelovalcev, saj predvideva za svoje delovanje uporabo različne že obstoječe strojne opreme, kot so osebni računalniki, pametni telefoni, miniaturni računalniki in drugo.

Sestavljajo ga štirje glavni programski sklopi, katerih integracija v skupen upravljalno-nadzorni sistem je prikazana na sliki 4:

- Controller (centralna logična in povezovalna enota),
- Designer (spletno orodje za konfiguracijo naprav in izdelavo uporabniškega vmesnika),
- Beehive (baza s podatki o uporabniških nastavitvah),
- Control Panel (nadzorna plošča).

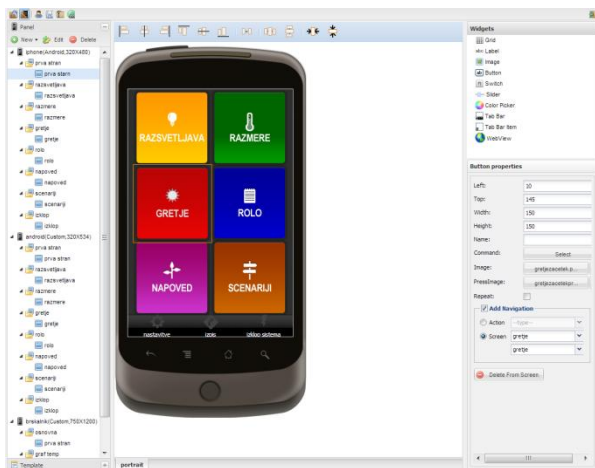
Controller je samostojna centralna upravljalna in nadzorna enota v sistemu OpenRemote. Zasnovana je na platformi Java (J2EE) in programskem jeziku C, zato potrebujemo za njegovo delovanje programskega okolja Java 6. Za izvajanje Controllerja skrbi spletni strežnik Apache Tomcat.

Kot osrednja komponenta opravlja Controller več ključnih nalog:

- Zagotavlja ustrezno zgradbo uporabniškega vmesnika za že izdelani Android in iOS aplikaciji ter aplikacijo, namenjeno spletnim brskalnikom. Zgradbo uporabniškega vmesnika določajo ustrezne datoteke XML.

- V postopku sinhronizacije pridobi vse nastavitve s strani baze Beehive (nastavitve, ki so nastavljene v okolju Designer).
- Sprejema ukaze iz uporabniških vmesnikov ter jih posreduje upravljanim sistemom. Podprt je širok nabor protokolov (1-Wire, KNX, X10, Z-Wave, Infrared, EnOcean, Insteon, TCP/IP idr.). Mogoča je integracija lastnih protokolov. Omogoča komunikacijo med sistemi oz. protokoli, ki v osnovi med seboj niso združljivi.
- Skrbi za ažuriranje podatkov na uporabniškem vmesniku s strani različnih sistemov, protokolov, naprav;
- Zagotavlja celovit REST [11] strežnik, ki prek HTTP/REST/XML in HTTP/REST/JASON API-jev skrbi za komunikacijo med drugimi komponentami sistema OpenRemote ali drugimi spletnimi aplikacijami.
- Omogoča zaporedno izvedbo združenih ukazov (angl. macro) za izvajanje pravil oz. scenarijev. Izvajanje pravil omogoča integracija projekta Drools [12], ki je deklarativno, na pravilih temelječe kodno okolje v programskem jeziku Java. Skrbi za ustrezen odziv sistema na pogoje, določene s pravili.

Designer je inovativna Java aplikacija, locirana v oblaku, ki omogoča dve ločeni upravljalni okolji. Prvo je namenjeno izdelavi uporabniškega vmesnika za že izdelane aplikacije za pametne telefone in tablice z operacijskim sistemom Android in iOS (iPhone, iPod, iPad) ali za spletne brskalnike.



Slika 2: Designer – oblikovanje uporabniškega vmesnika

Drugo okolje je namenjeno nastavitvi parametrov za različne protokole (protokoli za krmiljenje in nadzor zgradb, drugi komunikacijski protokoli) in drugim nastavitvam sistema OpenRemote. Pod druge nastavitve štejemo nastavitve različnih pravil (Drools), združenih ukazov (macros), nastavitve Controllerja in nekatere bolj splošne nastavitve protokolov. Ker je Designer lociran v oblaku, je – po predhodni prijavi – dostopen kar prek spletnega brskalnika.

Beehive je centralna podatkovna baza, locirana v oblaku, ki shranjuje vse uporabniške nastavitve: informacije o uporabniškem računu, šablone za oblikovanje, nastavitve naprav, protokolov in Controllerja. Tako kot Controller zagotavlja tudi Beehive baza HTTP/REST/XML in HTTP/REST/JASON API, ki skrbi za komunikacijo z drugimi komponentami sistema OpenRemote ali drugimi spletnimi aplikacijami. Zasnovana je na SQL (angl. Structured Query Language) in uporablja odprtokodni sistem za upravljanje podatkovnih baz MySQL.

Control panel je uporabniško upravljalno okolje sistema OpenRemote, ki se lahko izvaja na spletni aplikaciji iOS, Android ali HTML/JS. Odjemalec potrebuje za gradnjo kontrolne plošče ustrezne podatke o obliki in nastavitvah, ki jih pridobi od Controllerja, komunikacijo med njima pa zagotavlja HTTP/REST/XML programski vmesnik oz. API. Ustrezno obliko in nastavitve kontrolne plošče izdelamo s pomočjo začetnega nastavitvenega okolja.

Opozoriti je treba, da je varnostni vidik v sistemu OpenRemote trenutno še zelo slabo razvit in je glavna pomanjkljivost sistema. Na voljo sta zgolj varnostna mehanizma, ki ju omogoča strežnik Apache Tomcat. Prvi mehanizem zagotavlja zaščito pri dostopanju do kontrolnih plošč v odjemalcu. Omogoča zaščito kontrolne plošče z uporabniškimi gesli, ki jih uvedemo z ustrezno definicijo uporabnikov na strani strežnika Apache, in definicijo določenih omejitev na strani Controllerja. Drug mehanizem zagotavlja šifriranje podatkov med Controllerjem in odjemalcem. Šifriranje zagotavlja protokol SSL/TLS v izvedbi JSSE, ki je implementiran v strežnik Apache in skrbi za varnost na ravni seje. Za ustrezno delovanje potrebujemo samopodpisani certifikat (angl. self-signed certificate), ki skrbi za overjanje strežnika in določanje parametrov znotraj strežnika Apache. Omeniti velja še varnostno nastavitve v uporabniškem vmesniku Designerja, ki onemogoči komunikacijo med bazo Beehive in Controllerjem ter tako prepreči vnos lažnih nastavitvev v Controller. Navedeni varnostni mehanizmi niso zadostni in za zdaj pomenijo resna varnostna tveganja sistema OpenRemote.

3.2 PanStamp

PanStamp [6], [7] je odprtokodni projekt, namenjen brezžični kontroli in merjenju. Jedro projekta so miniaturne plošče Arduino z izjemno nizko porabo energije, brezžično komunikacijo in paleto vhodno-izhodnih priključkov. Njihova vloga je zelo dobrodošla pri avtomatizaciji zgradb, merjenju porabe energije, gradnji vremenskih postaj, krmiljenju robotov itd.

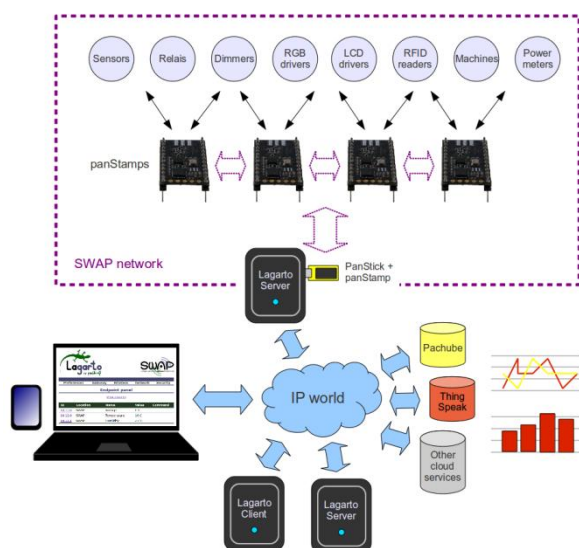
Celoten projekt je zasnovan tako, da je primeren tako za začetnike kot tudi resne razvijalce. Poleg brezžičnih modulov panStamp so na voljo tudi fizični vmesniki panStick (USB/SWAP vmesnik) in druge namenske plošče za merjenje temperature, tlaka, vlage, krmiljenje

relejev, LED diod itd. Za izvajanje preproste brezžične komunikacije uporablja panStamp lasten paketno naravnani komunikacijski protokol SWAP (angl. Simple Wireless Abstract Protocol) [8], ki skrbi za prenos vsebine registrov in drugih polj, potrebnih za komuniciranje in varnost.

Poleg plačljive strojne opreme je na voljo tudi brezplačna odprtokodna programska oprema. Lagarto predstavlja programsko opremo, ki omogoča povezavo omrežja SWAP (brezžičnih modulov panStamp) navzven v IP omrežje. Gre za nezahtevno programersko rešitev, ki lahko teče na miniaturnih računalnikih z majhno porabo. Lagarto omogoča integracijo najrazličnejših tehnologij ter gradnjo kompleksnih sistemov. Spisan je v programskem jeziku Python in sestavljen iz dveh ločenih entitet:

- strežnik Lagarto – omogoča povezavo med omrežjem SWAP in zunanjim IP omrežjem;
- odjemalec Lagarto – končna točka, ki strežniku pošilja ukaze, iz njega sprejema podatke ali pa podatke pošilja drugim aplikacijam.

Slika 3 prikazuje shemo sistema panStamp. Z več brezžičnimi enotami panStamp tvorimo omrežje SWAP. Na enote so priključeni različni senzorji, releji, motorji itd. Most med omrežjem SWAP in prostranim omrežjem (oblakom) na ravni strojne opreme tvori panStick v kombinaciji z brezžičnim modulom panStamp. Strežnik Lagarto, ki je programski prehod in upravljalno središče, komunicira z drugimi namenskimi strežniki, odjemalci oz. drugimi storitvami v oblaku. Projekt panStamp je s svojim spletnim uporabniškim vmesnikom lahko že sam po sebi zaključen sistem.



Slika 3: Celovita sistemska zgradba projekta panStamp

Brezžični moduli panStamp so veliki $17,1 \times 30,5$ mm. Vsebujejo mikroprocesor Atmel Atmega328P (8 MHz) v kombinaciji z radijsko enoto TI CC1101 (frekvenci 868 in 915 MHz). Za svoje delovanje

potrebuje napajanje okrog 3 VDC, poraba znaša $1 \mu\text{A}$ v stanju globokega spanja in 2,5 mA med oddajanjem.

3.3 1-Wire

1-Wire [9] je protokol, ki omogoča cenovno ugodno žično komunikacijo med osebnim računalnikom in paleto čipov, ki delujejo kot senzorji ali aktuatorji (merjenje temperature, vlage, stikala, identifikacijski čipi, A/D pretvorniki itd.). V osnovi gre za komunikacijo med neenakovrednima napravama, in sicer med nadrejeno napravo (kamor spada tudi 1-Wire prehod v kombinaciji s programsko opremo, na primer One Wire File System – OWFS) in podrejenimi napravami (različni čipi 1-Wire). Komunikacija uporablja nivoje TTL, je izmenično dvosmerna in poteka po dveh žičnih linijah. Napajalna napetost sistema je med 2,8 V in 6 V. Komunikacijo sproži in nadzoruje nadrejena naprava.

3.4 OWFS

OWFS [10] je skupek odprtokodnih programov, ki omogočajo preprost in učinkovit nadzor in upravljanje naprav 1-Wire in se izvaja na operacijskem sistemu Linux. Podprt je širok nabor čipov 1-Wire. OwsServer, ki je hrbtenica sistema OWFS, drugim procesom zagotavlja dostop do vodila 1-Wire. Za fizično povezavo z vodilom 1-Wire se ponavadi uporabijo serijska ali USB vrata, drugi procesi pa se na strežnik povezujejo preko omrežnega vtiča (TCP vrata). Komunikacija med strežnikom in drugimi procesi poteka prek omrežnega komunikacijskega protokola OwsServer (temelječega na TCP/IP) z arhitekturo odjemalec–strežnik.

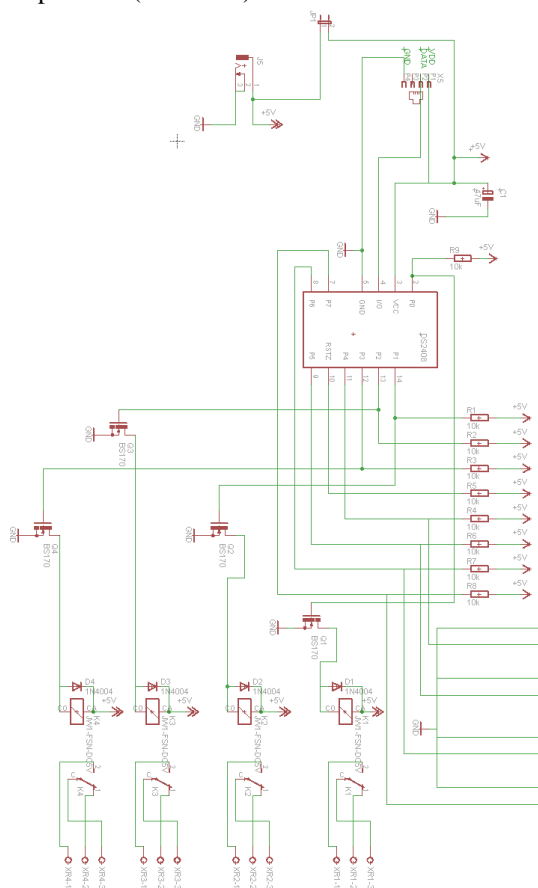
4 PRAKTIČNA IZVEDBA SISTEMA

Z integracijo zgoraj predstavljenih odprtokodnih rešitev smo izdelali sistem za avtomatizirano daljinsko upravljanje in nadzor zgradb.

Kot upravljalno in nadzorno središče sistema smo uporabili sistem OpenRemote. Za izvajanje Controllerja skrbi spletni strežnik Apache Tomcat, na katerem se poleg Controllerja izvajata še dve pomembni komponenti: Drools Engine, ki skrbi za izvajanje pravil in scenarijev avtomatizacije, in RRD4J, ki skrbi za shranjevanje podatkov in njihovo grafično prikazovanje (grafi časovne odvisnosti temperature in vlage). Zagotoviti je treba podporo ustreznim protokolom za branje senzorjev in krmiljenje aktuatorjev. Za žično upravljanje smo izbrali protokol 1-Wire, brezžično pa smo zagotovili z moduli panStamp. Komunikacijo med Controllerjem in omrežji 1-Wire in SWAP je treba izvajati prek posrednikov: OwsServer (OWFS, 1-Wire) in Lagarto (panStamp, SWAP).

Komunikacija med Controllerjem in OwsServerjem se izvaja na način odjemalec–strežnik, kjer je OwsServer odjemalec že integriran v sam Controller. Uporablja se lastni omrežni komunikacijski protokol OwsServer (angl.

Owserver network protocol). Na fizični ravni smo vodilo 1-Wire povezali na strežnik Owserver prek nadrejene enote 1-Wire, ki predstavlja fizični prehod. Uporabili smo prehod USB/1-Wire DS9490R. Z vidika optimizacije stroškov smo izdelali lastne senzorje in aktuatorje. Čipe 1-Wire smo uporabili za izdelavo strojne opreme relejev, stikal in senzorjev (DS2408) – slika 4, osvetljenosti (DS2438Z+ in fotodiode SFH203P), vlage (DS2438Z+ in HIH-4000) in temperature (DS18S20).

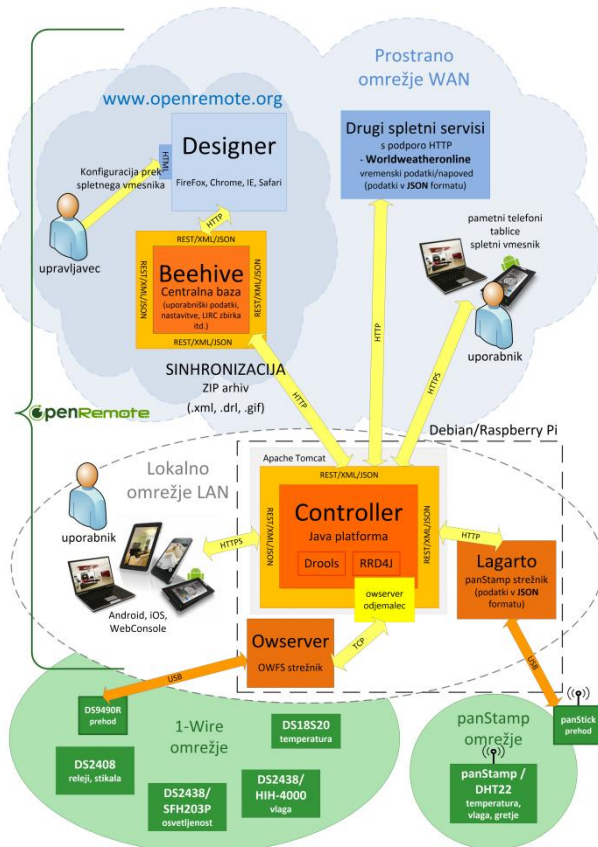


Slika 4: Shema enote 1-Wire z releji in stikali

Žal podpora za sistem panStamp v sistemu OpenRemote v času praktične implementacije še ni bila razvita (v času pisanja tega prispevka je že na voljo, op. a.), zato je komunikacija med Controllerjem in strežnikom Lagarto implementirana preko HTTP zahtev (JSON format), kar ilustrira možnost integracije zunanjih sistemov v OpenRemote. Omrežje SWAP tvorita dva brezžična modula panStamp. Za povezavo strežnika Lagarto z brezžičnim omrežjem panStamp (SWAP) smo uporabili fizični prehod, imenovan panStick. En modul v kombinaciji s panStick je prehod med strežnikom Lagart-SWAP in omrežjem SWAP, drugi pa v kombinaciji z lastno izdelanim tiskanim vezjem krmili gretje in s senzorja (DHT22) odčitava temperaturo in vlago.

Za pridobivanje zunanjih vremenskih prametrov in napovedi smo sistem povezali s spletno storitvijo

Worldweatheronline, ki podatke zajema prek HTTP zahtev v podatkovnem formatu JSON.



Slika 5: Shema zgradbe v praksi izvedenega sistema

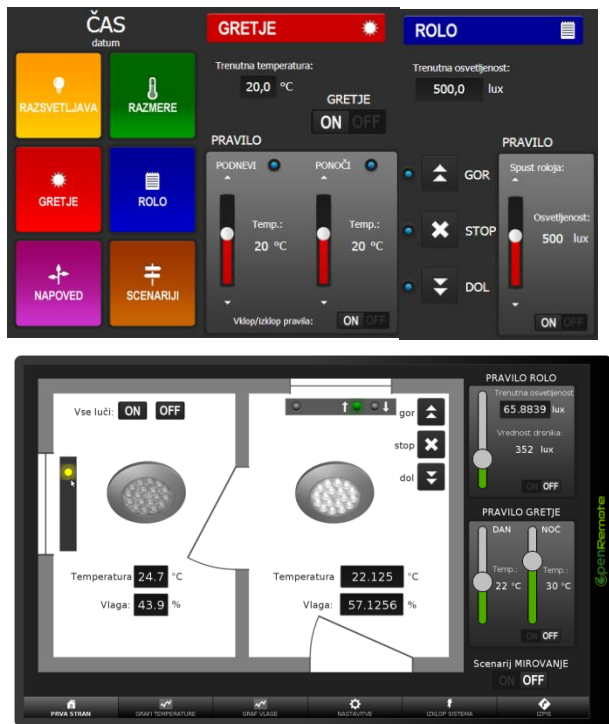
Na podlagi izdelanih senzorjev in aktuatorjev je nastal sistem s široko funkcionalnostjo: omogoča upravljanje razsvetljave, grelnih teles ter rolojev, merjenje temperature, vlage in osvetljenosti ter pridobiva zunanje vremenske podatke in napovedi. Za avtomatsko odzivanje sistema so bili izdelani dve pravili in scenarij (v okviru Drools), ki omogočajo avtomatsko prižiganje grelnega telesa glede na nastavljeno mejno vrednost temperature (nastavljivost z drsnikom) ter spuščanje in dviganje roloja glede na stopnjo osvetljenosti (nastavljivost z drsnikom). Če je nastavljena vrednost prekoračena, se rolo spusti. Scenarij ob njegovem vklopu hkrati izvede več nastavitvev.

Za nadzor in upravljanje sistema so bili izdelani trije uporabniški vmesniki (spletni vmesnik ter vmesnika za Android in iOS). Zasnova je ločena na dva tipa. Prvi je prilagojen spletnemu brskalniku, drugi pa pametnim telefonom z zaslonom na dotik (Android in iOS). Videz izdelanih uporabniških vmesnikov prikazuje slika 6.

5 SKLEP

Izdelan sistem je funkcijsko bogat, omogoča pa tudi avtomatiziran vidik delovanja. Uporabniški vmesniki zagotavljajo preprosto upravljanje in nadzor nad sistemom. Sistem odlikuje nizka cena, ki znaša za

predstavljen sistem približno 180 evrov. Vanjo je všteto vezje s štirimi releji in štirimi detektorji stikal, senzor temperature, vlage, osvetljenosti, vezje za upravljanje gretja in branje temperature ter vlage prek modula panStamp, fizični prehodi in drugi elektronski elementi.



Slika 6: Uporabniški vmesnik za Android in brskalnik

Pomemben vidik je tudi izjemno majhna energijska poraba celotnega sistema, ki znaša okrog 5 W (uporaba izvajane okolja Rasperry Pi [13]).

Sistem odlikujeta odlična možnost za razširljivost, nadgradnje ter možnost dodajanja velikega števila različnih naprav. Pri tem imamo v mislih predvsem sistem OpenRemote, medtem ko sta podsistema 1-Wire in panStamp nekoliko omejena. Glavna omejitev omrežja 1-Wire je število naprav na vodilu, saj vse uporabljajo isto vodilo. Omejitev glede največje oddaljenosti naprav na žičnem vodilu je približno 500 metrov. Do podobnih zaključkov lahko pridemo tudi pri sistemu panStamp, kjer je glavni omejitveni parameter število zahtev, ki jih prejema modem. Doseg brezžičnih modulov je okrog 100 metrov vidne razdalje (odvisno od tipa uporabljene antene). Z ustrezno postavitvijo in konfiguracijo sistemov 1-Wire in panStamp lahko dosežemo priklop več deset naprav na posamezni podsistem, kar po oceni zadošča za upravljanje manjših in srednje velikih zgradb.

Praktični preizkusi so pokazali, da je sistem dobro odziven. Odziv sistema na pritisk fizične tipke oziroma virtualnega stikala na uporabniškem vmesniku je zadovoljivo hiter in nemoteč (do pol sekunde). Tudi sinhronizacija stanj stikal na različnih uporabniških vmesnikih je zadovoljivo hitra.

Uporaba grafičnega prikaza podatkov prek integriranega orodja RRD4J [14] se je izkazala kot problematična, prikazovanje prek uporabniškega vmesnika namreč za brskalnik ni bilo mogoče. Znova velja opozoriti na pomanjkljivo varnost samega sistema. Na podlagi izkušenj bi za produkcijske implementacije za zdaj uporabo sicer privlačnega izvajalnega okolja Raspberry Pi (Debian weezy-armel) odsvetovali, saj se je izkazal za problematičnega s stališča procesorske moči in zanesljivosti delovanja. Priporočamo uporabo zmogljivejših naprav z operacijskim sistemom Debian Squeeze.

Za konec naj poudarimo, da ima sistem velike možnosti za nadaljnji razvoj.

LITERATURA

- [1] Neje Šmid, Odprtokodne rešitve za avtomatizirano upravljanje in nadzor zgradb, diplomsko delo, Fakulteta za elektrotehniko, Ljubljana, marec 2013.
- [2] OpenRemote, Open Source Automation Platform, <http://openremote.org/display/HOME/OpenRemote> (30.3. 2013).
- [3] OpenRemote, Open Source for Internet of Things, <http://www.openremote.com/> (30.3. 2013).
- [4] InfoWorld, Bossie Awards 2012: Now for something completely different, september 2012, <http://www.infoworld.com/slideshow/65168/bossie-awards-2012-now-something-completely-different-202541#slide2> (30.3. 2013).
- [5] OpenRemote forum, Beehive accountservice, security and certificates, maj 2012, <http://www.openremote.org/display/forums/Beehive+AccountService%2C+Security+and+Certificates> (30.3. 2013).
- [6] PanStamp, Introduction, <http://www.panstamp.com/home> (30.3. 2013).
- [7] PanStamp Documentation wiki and source code, panStamp, <https://code.google.com/p/panstamp/wiki/panStamp> (30.3. 2013).
- [8] PanStamp Documentation wiki and source code, SWAP <https://code.google.com/p/panstamp/wiki/SWAP> (30.3. 2013).
- [9] Maxim Integrated, 1-Wire Devices, <http://www.maximintegrated.com/products/1-wire/> (30.3. 2013).
- [10] OWFS, OWFS 1-Wire File System, <http://owfs.org/> (30.3. 2013).
- [11] Wikipedia, Representational state transfer, http://en.wikipedia.org/wiki/Representational_state_transfer (30.3. 2013).
- [12] Drools, Documentation, maj 2012, http://docs.jboss.org/drools/release/5.4.0.Final/drools-expert-docs/html_single/index.html (30.3. 2013).
- [13] Raspberry Pi, Quick start guide, <http://www.raspberrypi.org/quick-start-guide> (30.3. 2013).
- [14] RRD4J, Summary, <https://code.google.com/p/rrd4j/> (30.3. 2013).

Neje Šmid je diplomiral leta 2013 na Fakulteti za elektrotehniko v Ljubljani. Področje njegovega predhodnega strokovnega dela je zajemalo sisteme za avtomatizacijo ter daljinsko upravljanje in nadzor zgradb, trenutno pa deluje na področju razvoja multimedijskih platform, namenjenih e-učenju, e-zdravju, e-knjižnici in digitalnemu oglaševanju.

Iztok Humar je izredni profesor na Fakulteti za elektrotehniko Univerze v Ljubljani. Njegovo raziskovalno področje vključuje komunikacijske tehnologije za upravljanje in nadzor.